

317015  
-06

농  
축  
산  
ICT  
기  
자  
재  
표  
준  
기  
술  
개  
발  
  
최  
종  
보  
고  
서

2022

농  
림  
축  
산  
식  
품  
부  
  
농  
림  
식  
품  
기  
술  
기  
획  
평  
가  
원

보안 과제( ), 일반 과제( O ) / 공개( O ), 비공개( ) 발간등록번호( O )  
첨단생산기술개발사업 2022년도 최종보고서

발간등록번호

11-1543000-004421-01

# 농축산 ICT 기자재 표준 기술 개발

2023.07.14.

주관연구기관 한국전자통신연구원

농림축산식품부  
(전문기관)농림식품기술기획평가원



## 제 출 문

농림축산식품부 장관 귀하

본 보고서를 “농축산 ICT 기자재 표준 기술 개발” (개발기간 : 2017.4. ~ 2022.12.)과제의 최종보고서로 제출합니다.

2023. 7. 14.

주관연구기관명 : 한국전자통신연구원 (대표자) 방 승 찬 (인)

주관연구책임자 : 박 주 영

국가연구개발사업의 관리 등에 관한 규정 제18조에 따라 보고서 열람에 동의 합니다.



최종보고서										보안등급 일반[√], 보안[ ]		
중앙행정기관명		농림축산식품부			사업명	사업명		첨단생산 기술개발사업				
전문기관명 (해당 시 작성)		농림식품기술기획평가원			내역사업명 (해당 시 작성)							
공고번호		제 농축 2017-74호			총괄연구개발 식별번호 (해당 시 작성)							
					연구개발과제번호							
기술 분류	국가과학기술 표준분류	LB0805	50%	LB0806	30%	LB0804	20%					
	농림식품과학기술분류	RC0102	50%	RC0202	30%	CA0301	20%					
총괄연구개발명 (해당 시 작성)		국문										
		영문										
연구개발과제명		국문			농축산 ICT 기자재 표준 기술 개발							
		영문			Development of Standards for Interoperable SmartFarm ICT Devices							
주관연구개발기관		기관명	한국전자통신연구원			사업자등록번호	314-82-04099					
		주소	(34129) 대전시 유성구 가정로 218			법인등록번호						
연구책임자		성명	박주영			직위					책임	
		연락처	직장전화			휴대전화						
			전자우편			국가연구자번호						
연구개발기간		전체			2017.04.21. - 2022.12.31 (5년 9개월)							
		단계 (해당 시 작성)	1단계			2017.04.21. - 2018.12.31.						
			2단계			2019.01.01. - 2020.12.31.						
			3단계			2021.01.01. - 2022.12.31.						
연구개발비 (단위: 천원)		정부지원 연구개발비	기관부담 연구개발비	그 외 기관 등의 지원금				합계		연구 개발 비 외 지원금		
		현금	현금	현물	현금	현물	현금	현물	현금	현물	합계	
총계		1,680,000	-	-	-	-	-	1,680,000	-	1,680,000	-	
1단계		1년차	180,000	-	-	-	-	180,000	-	180,000	-	
		2년차	300,000	-	-	-	-	300,000	-	300,000	-	
2단계		3년차	300,000	-	-	-	-	300,000	-	300,000	-	
		4년차	300,000	-	-	-	-	300,000	-	300,000	-	
3단계		5년차	300,000	-	-	-	-	300,000	-	300,000	-	
		6년차	300,000	-	-	-	-	300,000	-	300,000	-	
공동연구개발기관 등 (해당 시 작성)		기관명	책임자	직위	휴대전화	전자우편	비고					
							역할		기관유형			
공동연구개발기관												
위탁연구개발기관												
연구개발기관 외 기관												
연구개발담당자 실무담당자		성명	최문환			직위					책임	
		연락처	직장전화			휴대전화						
			전자우편			국가연구자번호						

이 최종보고서에 기재된 내용이 사실임을 확인하며, 만약 사실이 아닌 경우 관련 법령 및 규정에 따라 제재처분 등의 불이익도 감수하겠습니다.

2023년 7월 14일

연구책임자: 박 주 영

주관연구개발기관의 장: 방 승 찬 (직인)



## < 요약 문 >

사업명	첨단생산기술개발사업	총괄연구개발 식별번호 (해당 시 작성)					
내역사업명 (해당 시 작성)		연구개발과제번호		317015-6			
기술 분류	국가과학기술 표준분류	LB0805	50%	LB0806	30%	LB0804	20%
	농림식품 과학기술분류	RC0102	50%	RC0202	30%	CA0301	20%
총괄연구개발명 (해당 시 작성)	해당 없음						
연구개발과제명	농축산 ICT 기자재 표준 기술 개발						
전체 연구개발기간	2017.04.21 - 2022.12.31 (5년 9개월)						
총 연구개발비	총1,680,000천원 ( 정부지원연구개발비: 1,680,000천원, 기관부담연구개발비 : 0 천원, 지방자치단체: 0 천원, 그 외 지원금: 0 천원)						
연구개발단계	기초[ ] 응용[ ] 개발[ ] 기타(위 3가지에 해당되지 않는 경우)[ √ ]		기술성숙도 (해당 시 기재)		착수시점 기준( ) 종료시점 목표( )		
연구개발과제 유형 (해당 시 작성)	해당 없음						
연구개발과제 특성 (해당 시 작성)	해당 없음						
연구 개발 목표 및 내용	<p>○ 주관기관 : 한국전자통신연구원</p> <p>○ 최종목표 : 상호호환성 및 상호운용성 확보를 위한 농축산 ICT 기자재 인터페이스 국내/국제 표준 기술 개발</p> <p>1) 농축산 ICT 부품 및 기자재 국내단체표준 개발(누적 60종 이상)                  2) 농축산 ICT 기자재 국내단체표준 개정 (5건 이상)                  3) 국가표준(KSC)개발 (8종 이상)                  4) 스마트 온실/축산/유통 관련 국제표준 개발 (3종 이상)                  5) 스마트팜 데이터 전송/제어 국내단체표준 개발 (3종 이상)                  6) PoC (Proof of Concept) 기능 구현을 통한 표준 검증</p>						
	최종 목표						
<p>※ 기존 비표준방식의 농축산 ICT 기자재들에 대한 이중 벤더간 상호호환성 및 상호운용성 확보</p> <p>※ 신규 농축산 ICT 기자재간 상호호환성 및 상호운용성 제공</p> <p>※ 개발된 표준 기술의 국내 단체표준화 · 국가표준화 · 국제표준화</p> <p>※ 농축산 ICT 기자재는 시설원예 · 축산 · 유통 분야를 포함</p>							

○ 연구개발내용

- 1) **스마트 온실/축산/유통 관련 ICT 부품 및 기자재 국내단체표준 개발** : 농축산 ICT 기자재간 상호호환성 확보를 위한 스마트온실 생육/에너지 관련 센서 및 기자재 · 품질관련 센서 및 기자재 · 시비/방제/관리 관련 센서 및 기자재 · 시설정보 · 스마트축우/양돈용 장치 운영요구사항 및 ICT 융합 인터페이스에 대한 국내 단체 표준 개발 (누적 60종 이상)

<표1> 스마트 온실/축산/유통 관련 ICT 부품 및 기자재 대상 (안)

구 분	1세대 모델 (‘17~’18)	2세대 모델 (‘19~’20)	3세대 모델 (‘21~’23)
시설 원예	외부 기상정보/내부 환경 정보 수집 및 천창, 축창, 환기팬 등 환경 제어를 위한 기자재	생장량, 엽면적, 줄기 굵기 등 생육정보 수집 및 양액공급 등 생육 제어를 위한 기자재	난방기, 지열 등의 에너지 제어를 위한 기기자재 및 자동화 기기를 포함하는 기자재
노지	외부 기상정보 수집 기기자재	관수/관비 등을 위한 기기자재	노지용 자동화 기기
축산	축종별 환경관리를 위한 기기자재	축종별 생육관리를 위한 기기자재	축종별 자동화 기기
유통/ 소비	수·배송 및 차량 환경 센서, 저장소 환경제어기 QR/Bar code, RFID 코드체계		재배, 가공,유통 단계 품질 모니터링
ARC	ARC 센터를 통해 개발된 기자재 20종		

전체 내용

- 스마트온실, 노지, 축사 용어 정의 등
- 시설원예: 온도센서(지온, 기온 등), 습도센서, EC센서, PH센서, 풍향센서, 풍속센서, 무게 센서, CO2센서, 식물 생육 감지 센서(지상부, 근권부 등 등) 등 실내외 환경 센서류 및 천창, 축창, 커튼 등 구동기류 통신 인터페이스
- 축사: 온도센서, 습도센서, CO2센서, O2센서, 암모니아센서 등 실내외 환경 및 안전 센서류, 환풍기, 음수관리기, 양액기, 개체관리, 사료급이기, 사료빈, 사료배합기(TMR조제기) 등 사양관리기 통신 인터페이스
- ※ 우선순위 의견 수렴 및 과제담당관과의 협의를 통해 대상항목 조정 가능
- 2) **농축산 ICT 기자재 공동규격 ICT 인터페이스 국내단체표준 개정** : 기존 단체 표준으로 확정된 스마트온실 및 축산 관련 표준(농진청 주관, 센서 13종, 제어기 9종, 축산용 센서 19종)에 대한 ICT 융합 인터페이스 추가 개발 및 국내 단체 표준 개정
- ※ 2015년/2016년도에 제정된 농축산 ICT 기자재 공동규격에 대한 ICT 인터페이스 추가 및 국내 표준 개정
- 3) **국가표준(KSC)개발** : 단체표준으로 개발된 스마트온실 및 축산 관련 표준 중 국내 스마트팜 활성화를 위한 우선순위에 따른 국가 KSC 표준 개발 (8종 이상)
  - RS485 기반 스마트팜(시설/축사/노지)내 통신 프로토콜
  - 농업용 기계류(트랙터) 통신 프로토콜 국가 표준 부합화
- 4) **스마트 온실/축산/유통 관련 국제표준개발** : 농축산 ICT 기자재간 상호호환성 확보를 위한 국제표준 개발 (3종 이상)
  - ※ ITU-T Y.ISG-FR, Framework of IoT-based Smart Greenhouse 표준안 개발
  - ※ ITU-T Y.2238, Overview of Smart Farming based Network 등 후속 표준안 개발
- 5) **스마트팜 데이터 전송/제어 표준 개발** : 네트워크 인터페이스별(LoRA, LTE, OneM2M, 클라우드, WiFi, ZigBee 등)
  - RS485 기반 스마트팜(시설/축사/노지)내 통신 프로토콜
  - RF기반 스마트팜(시설/축사/노지)내 통신 프로토콜



		<ul style="list-style-type: none"> <li>○ [PoC 검증] 표준 검증을 위한 PoC S/W 개발 <ul style="list-style-type: none"> <li>- 아두이노(Arduino) 등 IoT 단말을 활용하여 IP(WiFi, LTE 등) 기반 스마트팜(시설/축사/노지)내 ICT 프로토콜 및 데이터 PoC S/W 개발 및 통신 테스트베드 운용 <ul style="list-style-type: none"> <li>. 아이피(IP)기반 스마트 온실 노드제어 프로토타입 및 큐티(Qt)기반 응용</li> <li>. 스마트온실 노드제어 및 서비스제어 프로토콜 프로토타입</li> <li>. 농업빅데이터 시각화 분석을 위한 데이터 수집기</li> </ul> </li> </ul> </li> <li>○ 기타 <ul style="list-style-type: none"> <li>- ARC 연구센터와의 협력 (ARC 연구센터와의 지속적인 연구 교류)</li> <li>- 스마트팜 ICT 표준화 관련 언론 매체를 통한 홍보 (3건)</li> <li>- 스마트팜분야 국내/국제 표준화 전략맵 발간</li> <li>- 스마트팜 ICT 표준화 기술 보급 기업기술지원 및 세미나(15건)</li> <li>- 표준화 관련 정책 제안 (1건)</li> </ul> </li> </ul>
	<p style="text-align: center;">2 단계</p>	<p style="text-align: center;">목표</p> <ul style="list-style-type: none"> <li>○ 상호호환성 및 상호운용성 확보를 위한 농축산 ICT 기자재 인터페이스 국내 및 국제 표준 기술 개발 (스마트팜 2세대 모델 대상)</li> </ul> <p style="text-align: center;">내용</p> <ul style="list-style-type: none"> <li>○ [국내 표준개발] 스마트온실 관련 ICT 부품 및 기자재 표준초안 개발, 단체 표준 과제 제안 및 표준 채택 (제안:28종, 제정:45종, 개정제안:2종, 개정:2종) <ul style="list-style-type: none"> <li>- TTA PG426 (스마트농업) 그룹을 통한 축사용어-클라우드 기반 스마트팜관제·통신 관련 국내 단체 표준 과제제안 및 국내 단체표준 승인 <ul style="list-style-type: none"> <li>. 스마트팜 용어정의-제1부 : 시설원예(1종) ('17 제안, '19 제정)</li> <li>. 스마트축사 센서 메타데이터(실내·실외 15종) ('18 제안, '19 제정)</li> <li>. 스마트온실 서비스 제어프로토콜 기본 서비스절차 ('18 제안, '19 제정)</li> <li>. 경량 제어프로토콜을 위한 IP적응계층 동작요구사항 ('18 제안, '19 제정)</li> <li>. 스마트팜 용어정의-제2부 : 축사(1종) ('19 제안, '20 제정)</li> <li>. 스마트팜관제 프레임워크-제1부: 베이스프로토콜(1종) ('19 제안, '20 제정)</li> <li>. 스마트팜관제 프레임워크-제2부: 센서 프로파일(13종) ('19 제안, '20 제정)</li> <li>. 스마트팜관제 프레임워크-제3부: 개폐기 프로파일(1종) ('19 제안, '20 제정)</li> <li>. 스마트팜 영상제어장치 (1종) ('19 제안, '20 개정)</li> <li>. 스마트온실 서비스 프레임워크 (1종) ('19 제안, '20 개정)</li> <li>. 네트워크기반 스마트파밍 개요 (1종) ('19 제안, '20 개정)</li> <li>. 스마트팜관제 프레임워크-제4부: 돈사 사양관리기(11종) ('20 제안, '21 제정)</li> </ul> </li> </ul> </li> <li>○ [국제표준화] ITU-T 스마트팜 프레임워크 국제 표준 개발을 위한, 국제표준화기구 의장단 활동과 연계하여 주도적인 국제표준 개발 (기고서:9건, 국제표준제정: 1건, 의장단:2건) <ul style="list-style-type: none"> <li>- 국제표준화기구 (ITU-T) 의장단 활동과 연계한 국제표준화 (의장단 2건) <ul style="list-style-type: none"> <li>- 1) 스마트팜 관리를 위한 참조모델, 기능 요구사항, 2) 스마트축사 프레임워크, 3) 스마트팜 유즈케이스 관련 국제 기고서 제출 (13건)</li> </ul> </li> <li>- <b>스마트팜 관제 참조모델 국제표준승인추진 (ITU-T Y.4466) (1건)</b> <ul style="list-style-type: none"> <li>※ ITU-T Y.4466:2020, "Framework of IoT-based Smart Greenhouse Service"</li> </ul> </li> <li>- ISO/IEC JTC1/JETI를 통한 국내 스마트팜 기술 기고서 제출 (1건)</li> <li>- <b>스마트팜 유즈케이스 국제 신규과제 채택 (1건, 의장단진출: 2명)</b> <ul style="list-style-type: none"> <li>※ ITU-T Y.Sup.SmartAgri-usecase, "Use cases of IoT based Smart Agriculture"</li> </ul> </li> </ul> </li> <li>○ [IPR확보] 농축산 ICT 기자재 인터페이스 국내 및 국제 표준 기술 관련 IPR확보 (특허등록:4건, 특허출원: 5건, 논문: 6건) <ul style="list-style-type: none"> <li>- 특허심사관 의견 대응을 통해 특허등록 추진 (4건) <ul style="list-style-type: none"> <li>. IoT 노드 자동 구성 시스템 및 방법 ('19)</li> </ul> </li> </ul> </li> </ul>

		<ul style="list-style-type: none"> <li>. 통합 제어 라우팅 시스템 및 그 방법 ('19)</li> <li>. 센서 교체가 가능한 범용 센서 노드 및 그 구성 방법 ('19)</li> <li>. LCP를 이용하는 스마트온실 시스템 ('20)</li> <li>- 스마트팜 빅데이터 수집, 상호운용 및 전염병 추적조사 관련 특허 출원 (5건)</li> <li>. 스마트팜 환경·생육 데이터 압축 시스템 및 그 방법 ('20)</li> <li>. 스마트팜 빅데이터 거래를 위한 데이터 무결성 보장 방법 및 장치 ('20)</li> <li>. 상호 호환성 제공을 위한 스마트팜 API 게이트웨이 시스템 및 방법 ('20)</li> <li>. 타임라인기반의 전염병 추적조사 방법 및 시스템 ('20)</li> <li>. 멀티 팩터 측정 기반 스마트팜 기자재 공간위치인식 방법 및 장치 ('20)</li> <li>- 스마트팜 표준 및 PoC 구현 기반 국내/국제 컨퍼런스 논문 제출 (6건)</li> <li>o [PoC 검증] 표준 검증을 위한 Po635C S/W 개발 <ul style="list-style-type: none"> <li>- 아두이노(Arduino) 등 IoT 단말을 활용하여 LPWA기반 스마트팜(시설/축사/노지)내 ICT 프로토콜 및 데이터 PoC S/W 개발</li> <li>. 경량형 제어프로토콜 기반 스마트온실 통합 서비스 제어 프로그램 ('19)</li> <li>. 경량형 제어프로토콜 기반 센서/구동기 제어 프로그램 ('19)</li> <li>. 아이오티(IoT) 센서 기반 스마트팜 모니터링 웹유저인터페이스(UI) ('20)</li> <li>. 아이오티(IoT) 센서기반 스마트팜 노드 프로그램 ('20)</li> <li>. 아이오티(IoT) 센서 기반 스마트팜 노드 모니터링 프로그램 ('20)</li> </ul> </li> <li>o 기타 <ul style="list-style-type: none"> <li>- 스마트팜 ICT 표준화 관련 언론 매체를 통한 홍보 (5건)</li> <li>- 스마트팜 ICT 보급을 위한 도서 공저 (1건)</li> <li>. 스마트팜 개론 (ISBN 979-11-6257-127-9)</li> <li>- 스마트팜 ICT 표준화 기술 보급을 위한 기술 교육 및 기업기술지원 (26회)</li> <li>- TTA 표준화 전략맵 분과장 수임 등 스마트팜 표준화 전략수립 (3건)</li> <li>. SSM2021-표준화전략맵, ICT융합표준프레임워크, 표준활용가이드라인</li> <li>- 클라우드 기반 스마트팜 관제 방법, ISOBUS 표준기술 등 차세대 스마트팜 표준화 기술 분석 및 전략 수립</li> </ul> </li> </ul>
3 단계	목표	<ul style="list-style-type: none"> <li>o 상호호환성 및 상호운용성 확보를 위한 농축산 ICT 기자재 인터페이스 국내 및 국제 표준 기술 개발 (스마트팜 3세대 모델 대상)</li> </ul>
	내용	<ul style="list-style-type: none"> <li>o [국내 표준개발] 스마트축사, 노지 관련 ICT 표준초안 개발, 단체표준 과제 제안 및 표준 채택 (제안:28종, 제정:45종, 개정제안:2종, 개정:2종) <ul style="list-style-type: none"> <li>- TTA PG426 (스마트농업) 그룹을 통한 노지용어·클라우드 기반 스마트팜관제·통신 관련 국내 단체 표준 과제제안 및 국내 단체표준 승인</li> <li>. 스마트팜 용어정의 - 제 3부 : 노지 및 과수원(1종) ('21 제안, '22 제정)</li> <li>. 스마트팜관제 프레임워크-제4부: 돈사 사양관리기(11종) ('20 제안, '21 제정)</li> <li>. 스마트팜관제 프레임워크-제5부: 돈사환경관리기(10종)('21제안, '22제정)</li> <li>. 스마트팜관제 프레임워크-제6부: 우사사양관리기(11종)('21제안, '22제정)</li> <li>. UHF 대역 400 MHz 무선링크 기반 스마트 온실 제어 프로토콜('21제안, '22개정)</li> </ul> </li> <li>o [국가표준 개발] 농업용 장비(트랙터) 통신 관련 국제표준의 KS 부합화 추진 <ul style="list-style-type: none"> <li>- ISO TC23/SC19 국내 전문위원회를 통해 OSI 3 및 4계층에 해당되는 농업용 통신 프로토콜에 대한 KS 부합화 추진 (2건)</li> <li>. 농림업용 트랙터 및 기계류 - 제3부: 데이터링크 계층</li> <li>. 농림업용 트랙터 및 기계류 - 제4부: 네트워크 계층</li> </ul> </li> </ul> <p>※ KS예고고시(2022.9.26.~22.11.25)이후 최종심의회를 통과, 2023년도 상반기 KS고시 예상</p>

		<ul style="list-style-type: none"> <li>○ [국제표준화] 스마트축산 서비스 프레임워크, 스마트팜 유즈케이스 관련 국제 표준화 및 IEC 63246-1, ITU-T Y.4482 국제표준 제정 (국제표준제정: 2건, 표준기고서: 13건, 의장단:4명) <ul style="list-style-type: none"> <li>- IEC TC100을 통한 <b>사용자 중심의 차량 인포테인먼트 서비스 (IEC 63246-1) 국제 표준승인 ('21.5)</b></li> <li>- ITU-T SG20을 통한 <b>IoT 기반의 스마트축산 서비스 프레임워크 (Y.4482) 국제 표준승인 ('22.8)</b> <ul style="list-style-type: none"> <li>※ IEC 63246-1 (2021), Configurable car infotainment services (CCIS) - Part 1: General</li> <li>※ ITU-T Y.4482:2022, "Framework and capabilities for smart livestock farming based on Internet of things"</li> </ul> </li> <li>- ITU-T Q2/20 의장단 활동과 연계한 IoT 기반 스마트축산 서비스를 제공하기 위한 참조모델, 기능 요구사항 국제 기고서 제출 (5건)</li> <li>- ITU-T Q2/20 의장단 활동과 연계한 국내 시설원예/축산/노지 분야 기술 사례를 조사한 국제표준 기고서 제출 (8건)</li> <li>- 국제표준의장단 수임과 연계한 주도적인 ITU-T 국제표준 개발 (4명)</li> </ul> </li> <li>○ [IPR확보] 표준개발과 연계한 국내 현장의 요구사항을 반영한 IPR확보 (등록:3건, 출원: 4건, 논문: 5건) <ul style="list-style-type: none"> <li>- 가축전염병, 데이터 수집 관련 국내특허 등록 (3건) <ul style="list-style-type: none"> <li>. 타임라인기반의 전염병 추적조사 방법 및 시스템 ('21)</li> <li>. 멀티 팩터 측정을 이용한 스마트팜 기자재 공간위치인식 방법 ('21)</li> <li>. 스마트팜 데이터셋 유효성 검증을 위한 수집 데이터 선별장치 ('22)</li> </ul> </li> <li>- 식물생육 인식, 양액 처방, 야생동물 방지 및 컨테이너팜 관련 특허 출원 (4건) <ul style="list-style-type: none"> <li>. 영상장치를 이용한 식물 생육 측정 시스템 및 그 방법('21)</li> <li>. 스마트 팜에서의 비료 사용량 처방 시스템('21)</li> <li>. 컨테이너를 활용한 조사료 재배 장치('22)</li> <li>. 야생동물 식별을 통해 학습효과 방지가 가능한 야생동물 퇴치 시스템 ('22)</li> </ul> </li> <li>- 농축산 ICT 기자재 인터페이스 표준 기술 관련 논문 투고(8건)</li> </ul> </li> <li>○ [PoC 검증] 표준 검증을 위한 PoC S/W 개발 (SW: 3건) <ul style="list-style-type: none"> <li>- 본 사업에서 개발되는 국내 표준 기술들의 구현 가능성을 판단하고 표준 개선을 위해, 클라우드 기반의 스마트팜 관제 서비스의 프로토타입 구현 (3건) <ul style="list-style-type: none"> <li>. 애플리티 기반 스마트팜 센서/구동기/제어기 노드 연동프로그램('21)</li> <li>. 애플리티 기반 스마트팜 운영노드/클라우드 연동 프로그램('21)</li> <li>. 영상 기반 생육 센서 하드웨어 및 해석 시스템('22)</li> </ul> </li> <li>※ 상기 개발한 2건의 애플리티 기반 스마트팜 장치 S/W에 대한 시험성적서 획득 (KTC, '21)</li> </ul> </li> <li>○ 기타 <ul style="list-style-type: none"> <li>- 스마트팜 ICT 표준화 관련 언론 매체를 통한 홍보 (5건)</li> <li>- 농기평 R&amp;D유망기술 (조사료 재배를 위한 컨테이너팜) 발표회 홍보(1건)</li> <li>- 스마트팜 ICT 보급을 위한 도서 공저 (1건) <ul style="list-style-type: none"> <li>. 삼고 시설원예학 스마트팜 개론 (ISBN 979-89-7187-265-9)</li> </ul> </li> <li>- 스마트팜 ICT 표준화 기술 보급을 위한 기술 교육 및 기업기술지원 (26회)</li> <li>- TTA 표준화 전략맵 분과장 수임 등 스마트팜 표준화 전략수립 (1건) <ul style="list-style-type: none"> <li>. SSM2022-표준화전략맵</li> </ul> </li> <li>- 스마트팜 기자재 정보관리, 축산 자동제어기 등 국내단체표준 개발 지원(4건)</li> </ul> </li> </ul>
--	--	---

연구개발성과	<ul style="list-style-type: none"> <li>○ 스마트팜 환경제어장비의 확장과 관리의 유연성 확보가능 <ul style="list-style-type: none"> <li>- 현재 스마트팜 환경제어기는 장애시 해당업체외 A/S가 불가능하며, 기능변경이나 확대시 유연성이 떨어지기 때문에 납품당시(설치 시)의 기능으로 계속 사용해</li> </ul> </li> </ul>
--------	--

	<p>합</p> <ul style="list-style-type: none"> <li>- 업체들의 영세성으로 제품의 연속적인 운영 및 유지보수의 어려움을 극복하기 위한 통합 A/S 센터 운영 등을 통하여 공통 기술력 확보</li> </ul> <p>○ 스마트팜 기자재 인터페이스, 통신 등 표준규격의 적용 및 가이드라인 제공</p> <ul style="list-style-type: none"> <li>- 과학적인 방법을 통해 국내 환경에 맞는 스마트팜 유형 및 환경요인에 따른 최적의 환경제어 방법과 이를 적용한 범용 하드웨어 공급 가능</li> <li>- 데이터 기반의 컨설팅 체계를 구축하여 스마트팜 농가들의 경영 안정성 증대 및 수익 향상</li> <li>- 비표준 이기종 방식에 기인한 시설 재배 장치의 비경제성 극복, 작물의 생육, 재배, 운영, 설치, 유통, 가공 등에서 표준화된 지식 베이스 구축을 통해 비효율성을 해결</li> </ul>
--	--

연구개발성과 활용계획 및 기대 효과	<p>○ 국제표준 경쟁력 강화에 활용</p> <ul style="list-style-type: none"> <li>- 농축산 ICT 분야의 고부가가치 시장 및 국제 표준 선점</li> <li>- 농축산 재해 예방을 위한 ICT 기술 확보 및 국제 표준화를 통한 국제 경쟁력 확보</li> <li>- 농진청의 ICT 융합 한국형 스마트팜 개발 및 스마트팜 융합연구단의 R&amp;D와 연계한 국제 표준특허 확보를 통한 세계 시장 진출 기반 마련</li> </ul> <p>○ 중소기업 경쟁력 강화에 활용</p> <ul style="list-style-type: none"> <li>- 축산시설/시설원에 장비간 상호 운용성을 통한 ICT기반 산업의 활성화</li> <li>- 농업 ICT 융합 제품의 표준화를 통하여 균일한 품질의 대량 생산으로 국내 시장 규모를 키우고 시장 진입 장벽을 완화하여 영세한 국내 기업의 경쟁력 증진</li> </ul> <p>○ 신규 일자리 창출, 농촌 현안 문제 해결에 활용</p> <ul style="list-style-type: none"> <li>- 스마트농업 서비스 프레임워크 및 농작물 메타데이터 표준화는 새로운 스마트팜 모델 개발이 가능해 국내 스마트팜 시장 규모를 확대를 통한 신규 사업 진출 및 일자리 창출 기여가 예상됨</li> <li>- 스마트팜 확산 시 인력절감 및 편이 농업이 가능하여, 신규 농업인의 참여 확산을 통한 일자리 창출 가능함</li> </ul>
---------------------	--

연구개발성과의 비공개여부 및 사유	해당 없음
--------------------	-------

연구개발성과의 등록·기탁 건수	특허 출원	특허 등록	단체 표준 제안	단체 표준 채택	단체 표준 개정	국가 표준 제정	국제 기고서	국제 표준 채택	S/W 구현	논문 (비SCI)	교육 지도	정책 활용	홍보 전시	기타
* (목표치)	16 (10)	5 (6)	92 (60)	92 (60)	3 (5)	25 (10)	38 (22)	3 (3)	11 (4)	16 (7)	33 (0)	1 (2)	14 (4)	38 (0)
연구시설·장비 종합정보시스템 등록 현황	구입 기관	연구시설·장비명	규격 (모델명)	수량	구입 연월일	구입가격 (천원)	구입처 (전화)	비고 (설치장소)	ZEUS 등록번호					
	-	-	-	-	-	-	-	-	-					
국문핵심어 (5개 이내)	스마트팜		ICT 기자재		상호호환성		상호운용성		표준화					
영문핵심어 (5개 이내)	SmartFarm		ICT Devices		Interoperability		interchangeability		Standardization					

# < 목 차 >

1. 연구개발과제의 개요 .....	1
2. 연구개발과제의 수행 과정 및 수행 내용 .....	4
1) 국내단체표준 기술 개발 .....	4
가) 국내 단체표준 제·개정 절차 (정보통신단체표준) .....	4
나) 국내 제·개정 단체표준 .....	6
나-1) 스마트팜 용어정의-제1부 : 시설원예(1종) ('17~ '19) .....	6
나-2) 스마트팜 용어정의-제2부 : 축사(1종) ('19~ '20) .....	15
나-3) 스마트팜 용어정의-제3부 : 노지 및 과수원(1종) ('21~ '22) .....	21
나-4) 스마트온실 서비스 프레임워크 (1종) ('19~ '20) .....	31
나-5) 네트워크기반 스마트파밍 개요 (1종) ('19~ '20) .....	40
나-6) 스마트 온실을 위한 원격 감시용 스마트 영상 장치 (1종) ('19~ '20) .....	42
나-7) 스마트 온실 센서구동기 및 제어기 간 RS485 기반 MODBUS 인터페이스(1종) ('17~ '18) ·	45
나-8) IP 기반 스마트팜(시설/축사/노지)내 통신을 위한 경량형 제어 프로토콜(1종) ('17~ '18) ·	57
나-9) 스마트온실 서비스 제어프로토콜 기본 서비스절차 ('18~ '19) .....	62
나-10) 경량 제어프로토콜을 위한 IP적응계층 동작요구사항 ('18~ '19) .....	64
나-11) UHF 대역 400 MHz 무선링크 기반 스마트 온실 제어 프로토콜('21~ '22) .....	65
나-12) 스마트온실 센서 메타데이터 (13종) ('17~ '18) .....	77
나-13) 스마트온실 구동기 메타데이터 (9종) ('17~ '18) .....	80
나-14) 스마트축사 센서 메타데이터(실내·실외·안전 15종) ('18~ '19) .....	85
나-15) 스마트팜관제 프레임워크-제1부: 베이스프로토콜(1종) ('19~ '20) .....	91
나-16) 스마트팜관제 프레임워크-제2부: 센서 프로파일(13종) ('19~ '20) .....	95
나-17) 스마트팜관제 프레임워크-제3부: 개폐기 프로파일 (1종) ('19~ '20) .....	109
나-18) 스마트팜관제 프레임워크-제4부: 돈사 사양관리기(11종) ('20~ '21) .....	110
나-19) 스마트팜관제 프레임워크-제5부: 돈사환경관리기(10종)('21~ '22) .....	122
나-20) 스마트팜관제 프레임워크-제6부: 우사사양관리기(11종)('21~ '22) .....	132
2) 국가 표준 기술 개발 .....	143
가) 국가 표준 추진 절차 .....	143
나) 국가 제·개정 표준 .....	144
나-1) [국가표준] 스마트온실 센서 메타데이터 .....	144
나-2) [국가표준] 스마트온실 구동기 메타데이터 .....	147
나-3) [국가표준] 센서/구동기 노드와 온실 통합 제어기간 RS485 모드버스인터페이스 .....	151
나-4) [국가표준] 농림업용 트랙터 및 기계류 - 제3부: 데이터링크 계층 .....	167
나-5) [국가표준] 농림업용 트랙터 및 기계류 - 제4부: 네트워크 계층 .....	187
3) 국제표준 기술 개발 .....	203
가) 국제표준 추진 절차 .....	203
가-1) ITU-T .....	203
가-2) IEC .....	205
나) 국제표준 제정 .....	206
나-1) ITU-T Y.4466 ('17~'20) .....	206

나-2) IEC 63246-1, Configurable car infotainment services (CCIS) - Part 1: General	213
나-3) ITU-T Y.4482	223
나-4) ITU-T Y.Sup.SmartAgri-usecases	230
라) 국제의장단 수입	242
라-1) [국제의장단] ITU-T Y.4466 에디터 수입	242
라-2) [국제의장단] ITU-T Y.4482 에디터 수입	243
라-3) [국제의장단] ITU-T Y.Sup.SmartAgri-usecases 에디터 수입	244
라-4) [국제의장단] IEC 프로젝트 리더 (Project Leader) 수입	245
<b>4) PoC S/W 구현</b>	<b>246</b>
가) OCF기반 스마트팜 IoT 기자재 PoC 개발	246
가-1) [SW등록] 엠큐티티(MQTT)기반 스마트팜 센서/구동기/제어기 노드 연동 프로그램 ('18)	246
가-2) [SW구현] 경량형 제어프로토콜 기반 스마트온실 통합 서비스 제어 프로그램 ('19)	249
가-3) [SW등록] 아이오티(IoT) 센서 기반 스마트팜 모니터링 S/W ('20)	274
가-4) [SW등록] 엠큐티티(MQTT)기반 스마트팜 센서/구동기/제어기 노드 연동 프로그램('21)	286
가-5) [SW등록] 엠큐티티(MQTT)기반 스마트팜 운영노드/클라우드 연동 프로그램 ('21)	294
가-6) [SW등록] 영상기반 생육 센서 하드웨어 및 해석 시스템 ('22)	310
가-7) 인공지능 기반 스마트팜 테스트베드 운용 ('19)	316
<b>5) IPR 및 학술성과</b>	<b>322</b>
가) [국내특허] 국내특허 출원 및 등록	322
가-1) [국내특허등록] 국내특허 등록 (5건)	322
가-2) [국내특허출원] 국내특허 출원	332
나) [논문] 국내외 학술대회 논문	359
나-1) 도커 기반 스마트 온실 데이터 분석 플랫폼 개발 (ICTC 2018, '18)	360
나-2) 스마트팜 장치 표준 기술 분석 및 검증 (한국통신학회, '20)	363
나-3) ITU-T SG20 스마트팜 표준화 동향 (SEP Inside, '20)	365
나-4) IoT 기반 스마트 온실 서비스 프레임워크 표준 소개 (한국통신학회, '20)	370
나-5) Implementation of SmartFarm Devices using OpenSource Software (ICACT, '21)	374
나-6) 스마트축산 표준화 동향 (JCCI2021, '21)	375
나-7) 기상 IoT 센서 데이터를 활용한 서리 예측 연구 (한국인공지능학술대회, '21)	376
나-8) Understanding IoT Climate Data based Predictive Model (ICTC2021, '21)	378
<b>6) 기타 연구활용</b>	<b>379</b>
가) 스마트 농업 관련 국내 대학 교재 공저	379
가-1) [도서저술] 「스마트팜 개론」 (구미출판사) 공저 ('20)	379
가-2) 「시설원예학」 (학문사) 공저 ('21)	386
가-3) 「스마트팜 개론」 (구미서관) 공저 ('21)	391
나) 기술지원	392
나-1) 스마트팜 단체표준 (3건) 제정을 위한 기술지원	392
나-2) [기술지원] 스마트팜 국가표준 확산을 위한 기술지원 ('20~'22)	410
다) 스마트팜 국내/국제 표준화 전략수립을 위한 표준화 전략맵 개발	421
다-1) [표준화 전략맵] 스마트팜 국내·국제 표준화 전략 개발	421
다-2) [표준화 프레임워크] 스마트팜 표준화 체계 수립 및 표준항목 도출 ('19)	446
라) 스마트팜 표준 확산을 위한 해설서 개발	451

라-1) [표준활용 가이드라인] 국내·국제 스마트팜 표준 해설 및 활용 가이드라인 ('21) .....	451
마) [홍보·전시] 스마트팜 기술 및 표준화 동향 홍보 .....	455
바) [기술분석서] 스마트팜 상호연동을 위한 IoT 기술 분석 .....	459
바-1) [상호연동을 위한 프레임워크 표준기술 연구] OCF 표준기술 분석 및 스마트팜 기자재간 상호연동성을 제공하기 위한 프레임워크 표준기술 연구('19) .....	459
바-2) [스마트팜 기자재 보안 표준기술 연구] 스마트팜에서 사용되는 IoT 기자재에 대한 보안 기술 적용방안 연구 ('19) .....	502
사) [정책활용] 스마트팜 표준용어 활성화 방안 제안 (1건) .....	560
<b>3. 연구개발과제의 수행 결과 및 목표 달성 정도 .....</b>	<b>564</b>
<b>4. 목표 미달 시 원인분석 .....</b>	<b>581</b>
<b>5. 연구개발성과의 관련 분야에 대한 기여 정도 .....</b>	<b>581</b>
<b>6. 연구개발성과의 관리 및 활용 계획 .....</b>	<b>582</b>

## 별첨 자료

별첨-1) 자체평가의견서

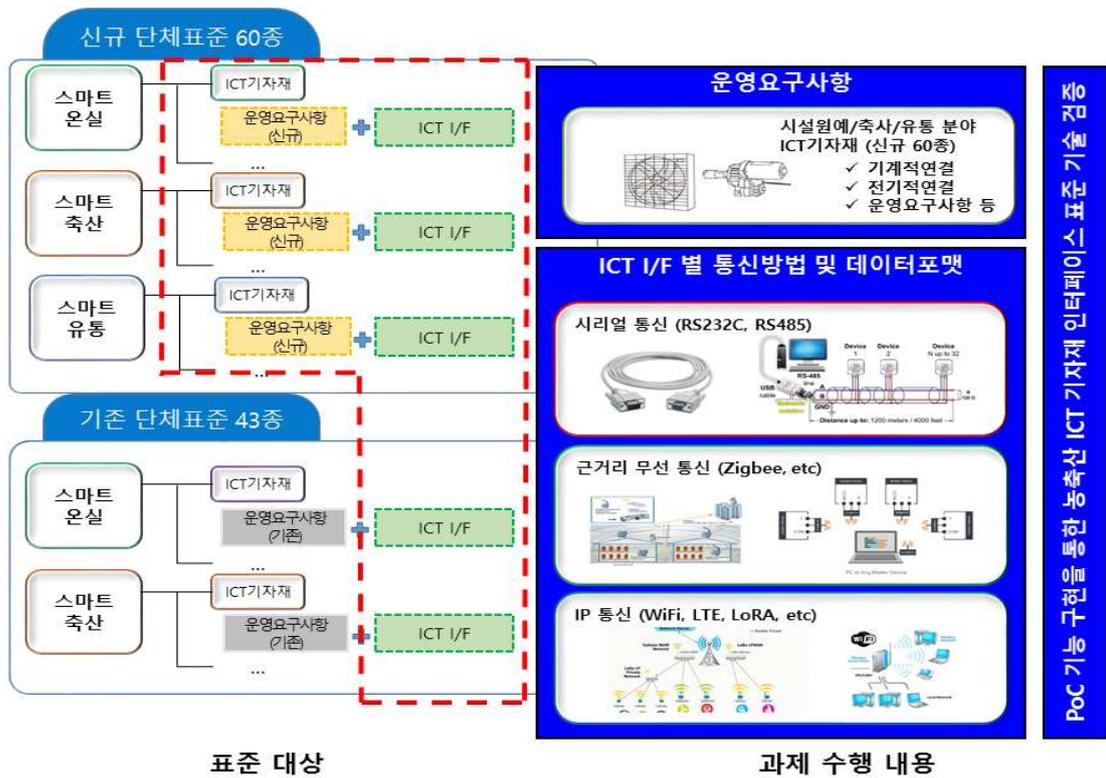
별첨-2) 연구성과 활용계획서

# 1. 연구개발과제의 개요

**상호호환성 및 상호운용성 확보를 위한  
농축산 ICT 기자재 인터페이스 국내/국제 표준 기술 개발**

- 1) 스마트 온실/축산/유통 관련 ICT 부품 및 기자재 국내단체표준 개발 (누적 60종 이상)
- 2) 농축산 ICT 기자재 공동규격 ICT 인터페이스 국내단체표준 개정 (5건 이상)
- 3) 국가표준(KSC)개발 (8종 이상)
- 4) 스마트 온실/축산/유통 관련 국제표준 개발 (3종 이상)
- 5) 스마트팜 데이터 전송/제어 국내단체표준 개발 (3종 이상)
- 6) PoC (Proof of Concept) 기능 구현을 통한 표준 검증

- ※ 기존 비표준방식의 농축산 ICT 기자재들에 대한 이중 벤더간 상호호환성 및 상호운용성 확보
- ※ 신규 농축산 ICT 기자재간 상호호환성 및 상호운용성 제공
- ※ 개발된 표준 기술의 국내 단체표준화 · 국가표준화 · 국제표준화
- ※ 농축산 ICT 기자재는 시설원예 · 축산 · 유통 분야를 포함



< 본 연구의 대상 및 범위 >

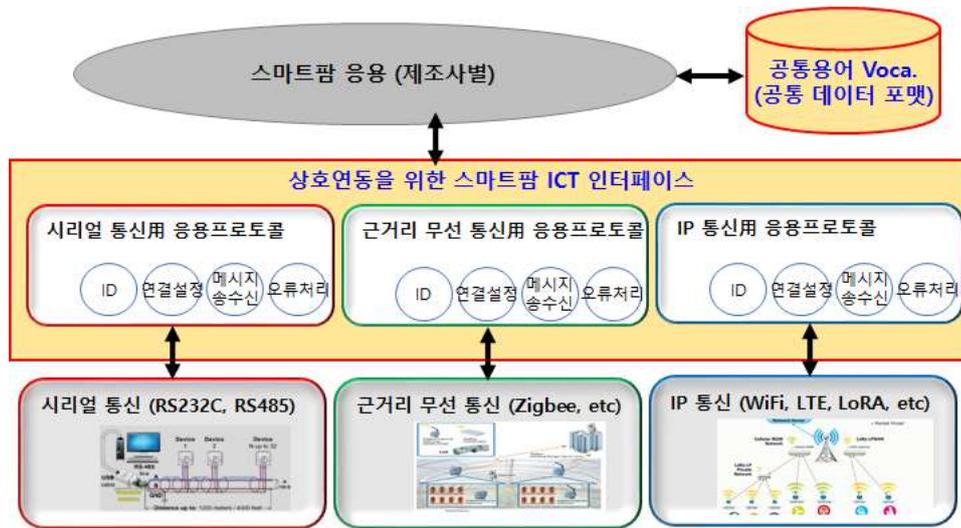
○ 핵심기술

- 1) **스마트 온실/축산/유통 관련 ICT 부품 및 기자재 국내단체표준 개발** : 농축산 ICT 기자재간 상호호환성 확보를 위한 스마트온실 생육/에너지 관련 센서 및 기자재 · 품질관련 센서 및 기자재 · 시비/방제/관리 관련 센서 및 기자재 · 시설정보 · 스마트축우/양돈용 장치 운영요구사항 및 ICT 융합 인터페이스에 대한 국내 단체 표준 개발 (누적 60종 이상)

<표1> 스마트 온실/축산/유통 관련 ICT 부품 및 기자재 대상 (안)

구 분	1세대 모델 (‘17~’18)	2세대 모델 (‘19~’20)	3세대 모델 (‘21~’23)
시설원예	외부 기상정보/내부 환경 정보 등 수집 및 천창, 측창, 환기팬 등 환경 제어를 위한 기자재	생장량, 엽면적, 줄기 굵기 등 등 생육정보 수집 및 양액공급 생육 제어를 위한 기자재	난방기, 지열 등의 에너지 제어를 위한 기자재 및 자동화 기기를 포함하는 기자재
노지	외부 기상정보 수집 기자재	관수/관비 등을 위한 기자재	노지용 자동화 기기
축산	축종별 환경관리를 위한 기자재	축종별 생육관리를 위한 기자재	축종별 자동화 기기
유통/소비	수배송 및 차량 환경 센서, 저장소 환경제어기 QR/Bar code, RFID 코드체계		재배·가공·유통 단계 품질 모니터링
ARC 기자재	ARC 센터를 통해 개발된 기자재 20종		

- 2) **농축산 ICT 기자재 공동규격 ICT 인터페이스 국내단체표준 개정** : 기존 단체 표준으로 확정된 스마트 온실 및 축산 관련 표준(농진청 주관, 센서 13종, 제어기 9종, 축산용 센서 19종)에 대한 ICT 융합 인터페이스 추가 개발 및 국내 단체 표준 개정
- 3) **국가표준(KSC)개발** : 단체표준으로 개발된 스마트온실 및 축산 관련 표준 중 국내 스마트팜 활성화를 위한 우선순위도에 따른 국가 KSC 표준 개발 (8종 이상)
- 4) **스마트 온실/축산/유통 관련 국제표준개발** : 농축산 ICT 기자재간 상호운용성 확보를 위한 국제표준 개발 (3종 이상)
  - ITU-T Y.ISG-FR, Framework of IoT-based Smart Greenhouse 표준안 개발
  - ITU-T Y.IoT-SLF, Framework and capabilities for smart livestock farming based on Internet of things 표준안 개발
  - ITU-T Y.Sup.SmartAgri-usecases, Use cases of IoT based smart agriculture 표준안 개발
  - ※ 현재, ITU-T SG20에서 개발중인 표준안들(Y.ISG-FR, Y.PSF, Y.POPS 등)의 에디터십 확보를 통해 주도적인 국제표준화 추진
- 5) **스마트팜 데이터 전송/제어 표준 개발** : 네트워크 인터페이스별(LoRa, LTE, OneM2M, 클라우드, WiFi, ZigBee 등)
  - ※ 사물인터넷 전용망 (LoRaWAN, NB-IoT, LTE-M)
  - ※ LoRaWAN(Long Range Wide Area Network) : 배터리 기반 IoT 장치들의 저전력 통신을 위해 LoRa Alliance에서 개발한 MAC 계층 기술(2015)



< 스마트팜 데이터 전송/제어 표준 대상 >

상호연동을 위한 스마트팜 ICT 인터페이스란? :

하부통신 기술별 스마트팜응용 프로그램 구현에 사용되는 공통 통신 기능요소들

- 공통 통신 기능 요소들 : ID, 연결설정, 메시지 송수신, 오류처리 기능, 메시지 형태 등
- 대표 하부 통신 기술 : 시리얼(RS485), 근거리(LoRa, ZigBee), IP(LTE, WiFi)

※ “공통된 데이터 용어 및 포맷”과 “스마트팜 ICT 인터페이스” 사용을 통해, 스마트팜 응용 프로그램은 이종 벤더와 상호연동 및 빅데이터 구축 용이

6) PoC (Proof of Concept) 기능 구현 : 개발된 표준의 무결성 검증을 위한 표준 기능 구현 및 시험

- 본 연구를 통하여 개발될 상호연동 표준기술은 산업체의 요구를 반영하여 작성하여야 하며, 국내외 표준 채택에 앞서 반드시 무결성 검증이 선행되어야 함
- 이를 위하여 본 연구에서는 상호연동을 위한 스마트팜 ICT 인터페이스와 데이터 포맷에 대한 PoC 구현을 통한 표준 적합성 검증을 수행

## 2. 연구개발과제의 수행 과정 및 수행 내용

### 1) 국내단체표준 기술 개발

#### 가) 국내 단체표준 제·개정 절차 (정보통신단체표준)

정보통신단체표준(이하 'TTA 표준'이라 함)이란 한국정보통신기술협회(TTA)를 비롯한 정보통신 분야의 단체가 그 구성원의 공동이익을 도모하고 이용자를 보호하기 위해 개발한 표준을 말한다. TTA 표준은 이해관계를 가진 개인 또는 단체가 누구든지 언제라도 표준의 제정, 개정 및 폐지를 그림 1과 같은 절차를 통해 진행할 수 있다.

개발된 표준에 대한 의견수렴은 최소 60일간의 표준 초안 공고에 대해 이루어지며 특히 표준안을 작성한 PG에서는 해당 표준초안에 대한 의견을 수집한다. 또한 기개발 표준안에 대해서는 TTA 홈페이지를 통해 배포되어 의견을 개진하고자 할 때, 개정 혹은 폐지 작업을 추진할 수 있다.

참고로 국가표준(방송통신표준)이란 통신사업자, 제조업체, 연구소, 정부 부처 등 국가 전체에 영향을 주는 표준으로서, 과학기술정보통신부 장관이 고시하는 표준을 말한다.

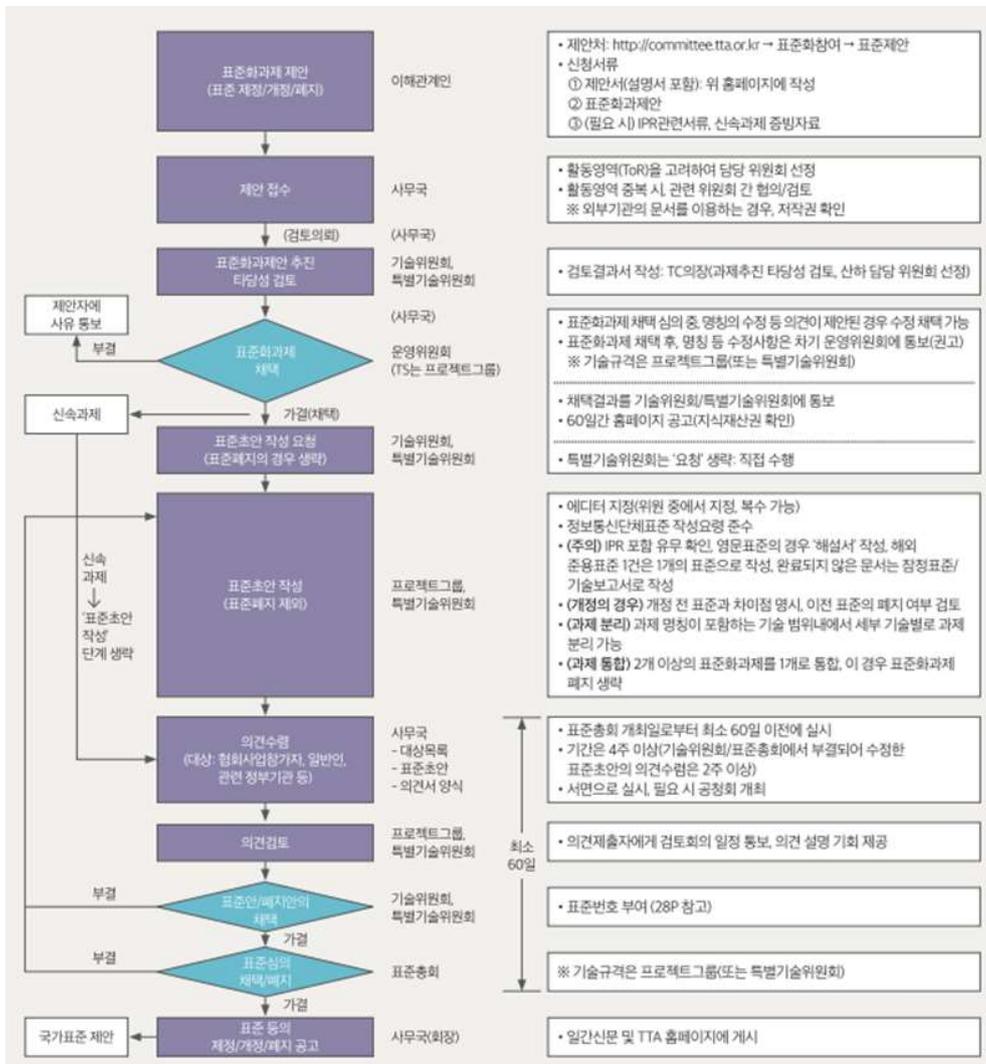


그림 5 - TTA표준 제정·개정·폐지 절차

TTA 표준은 그림 2와 같이 8개의 기술위원회(TC) 산하 54개의 프로젝트 그룹(PG)를 통하여 만들어진다. 스마트농업 분야의 경우, ICT 융합 기술위원회(TC4) 산하의 스마트 농축수산 프로젝트 그룹(PG426)에서 표준화가 진행된다.



그림 6 - TTA 표준화 위원회의 구성

## 나) 국내 제·개정 단체표준

### 나-1) 스마트팜 용어정의-제1부 : 시설원예(1종) ('17 제안, '19 제정)

스마트팜(smart farm)은 원예, 축산, 노지 등의 농업 분야에 ICT(Information and Communication Technologies)를 접목하여 원격에서 자동으로 작물이나 가축 등의 최적 생육 환경을 유지하고 관리할 수 있게 한다. 이를 통해 사용자에게 편의성을 제공할 수 있을 뿐만 아니라 빅데이터(big data) 구축을 통한 과학 영농을 가능하게 하고, 나아가 생산량과 소비량 등의 정확한 예측을 통하여 농가의 수익을 향상시킬 수 있는 기술이다.

이와 같이 농업과 ICT 분야가 융합된 스마트팜 기술이 도입되면서 각 분야에서 사용하는 용어에 대한 이해의 차이로 인한 불필요한 혼란이 가중되고 있으며, 이는 스마트팜 기술과 서비스의 확산과 보급에 걸림돌로서 작용하고 있다. 이에 정부와 산업계, 연구계 등에서 사용하는 스마트팜 관련 용어, 특히 스마트 온실이나 시설원예 분야에 대한 다양한 용어를 체계적으로 정의하여 농장주와 농장의 종사자는 물론 스마트팜 관련 장비 제조업자 및 시설업자, 서비스 제공업자의 이해를 증진시키기 위한 표준화를 추진하였으며, 2019년도 한국정보통신기술협회(TTA)의 단체표준으로서 제안하여 12월에 최종 채택될 예정이다.

스마트팜과 관련된 국내외 표준 문서뿐만 아니라 관련 부처 및 국립국어원 등 정부 및 공공기관의 발행 자료를 조사하여 스마트팜 기자재, 농업 생산 기술 및 ICT와 관련된 용어를 선별하고, 동일 용어에 대한 서로 다른 정의를 일원화하여 다양한 분야에서 공통적으로 적용할 수 있는 129개 용어에 대한 표준 정의를 제시한다. 또한 각 용어가 서로 다른 분야에서 효과적으로 사용될 수 있도록 구체적인 부연 설명과 세부 정의를 함께 제공하고 있으며 다음 표와 같다.

이 표준은 스마트팜, 특히 시설원예 및 온실 분야에 대한 표준 정의를 제시함에 그 의미가 있으며, 특히 농촌진흥청, 국립농업과학원 및 농업기술실용화재단과 협업을 통하여 다수에 걸친 용어 감수와 개선을 통하여 만들어졌다. 이 표준은 4차년도에 국가표준으로 제정 추진하여, 스마트팜 관련 국내외 표준화 작업을 함에 있어 가이드라인으로 활용할 예정이다. 또한 이를 기반으로 하여 향후 노지, 축사 분야뿐만 아니라 타 산업분야(가령, 스마트 수산 양식 분야)에 대한 표준 용어 정의를 수립함에 있어 그 기초자료로서의 활용할 수 있을 것이다.

표 1 - 스마트팜(온실분야) 용어 정의 현황

순번	용어	정의
1	1A	전자기 릴레이에서 A접점이 1극인 릴레이
2	1A1B	전자기 릴레이에서 A접점이 1극, B접점이 1극인 릴레이
3	2A2B	전자기 릴레이에서 A접점이 2극, B접점이 2극인 릴레이
4	ADC 핸들 (Analog Digital Converter Handle)	ADC 디바이스에 액세스하기 위하여 온실 통합 제어기, 센서노드, 제어 노드의 응용 프로그램에서 사용하는 변수
5	API (Application Programming Interface)	응용 프로그램이 컴퓨터 운영 체제나 데이터베이스 관리 시스템(DBMS) 등의 다른 프로그램이나 장치의 기능을 이용하기 위한 작용수단으로서, 컴퓨터 운영 체제의 기능과 그 기능을 사용하는 방법을 정의한 함수의 집합 - 비교 : C, C++, Pascal 등과 같은 언어로 만들어진 응용 프로그램에서 사용할 수 있도록 컴퓨터 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어하는 인터페이스를 말함
6	CO <sub>2</sub> 공급기 (Carbon Dioxide Generator)	온실 내 CO <sub>2</sub> (이산화탄소)를 공급하기 위한 설비
7	CO <sub>2</sub> 센서 (Carbon Dioxide Sensor)	공기 중의 CO <sub>2</sub> (이산화탄소)의 농도(μmol/mol(ppm))를 측정하는 장치
8	DVR (Digital video recorder)	영상을 하드 디스크 기반의 디지털 저장장치에 기록하는 장치(디지털 비디오 녹화기) - 비교 : 기존의 비디오 테이프 녹화기(VTR, Video Tape Recorder)를 대체하는 장치로 영상의 녹화, 재생, 전송 기능을 갖는 폐쇄 회로 텔레비전(CCTV) 시스템의 일부분임
9	EC 센서 (Electrical Conductivity sensor)	수용액 중의 전기 전도도(dS/m)를 측정하는 장치
10	EC 컨트롤러	수용액의 전기 전도도를 제어해 주는 장치

	(Electrical Conductivity controller)	
11	FDR 센서 (Frequency Domain Reflectometry sensor)	토양의 수분 함량(%)을 측정하는 장치 - 비고 1 : 전자파를 이용하여 토양의 정전 용량을 평가하고 이를 통해 토양의 수분 함량을 산정함 - 비고 2 : 토양의 수분 함량을 측정하는 또 다른 방식으로 TDR 센서(5.20 참조) 방식이 있음
12	GPIO (General Purpose Input/Output)	프로세서나 제어기(controller) 등에서 일반적인 목적으로 사용하도록 준비된 입출력 포트 - 비고 : 소프트웨어와 연동시키면서 전기적 입력을 받거나 출력으로 특정 디바이스를 제어할 수 있음
13	GPIO 핸들 (General Purpose Input/Output handle)	다용도 입출력 포트(GPIO)를 갖는 디바이스에 액세스하기 위하여 온실 통합 제어기, 센서노드, 제어 노드의 응용 프로그램에서 사용하는 변수
14	MODBUS	PLC(Programmable Logic Controller)와 함께 사용하기 위해 1978년에 개발된 직렬 통신(serial communication) 프로토콜 - 비고 : MODBUS에서는 데이터 표기를 위해 바이너리 형식의 RTU(Remote Terminal Unit) 포맷 또는 ACSII 포맷을 이용함
15	NVR (Network Video Recorder)	네트워크로 연결된 카메라나 비디오 서버의 영상 녹화, 모니터링, 이벤트 관리, 재생 등을 위한 전용 PC 서버 장치 - 비고 : 기존의 독립형 DVR(Digital Video Recorder)을 대체하는 것으로 IP(Internet Protocol) 기반 네트워크를 이용하여 디지털 영상을 전송받아 압축, 저장하므로 별도의 녹화기가 없어도 손쉽게 지정된 장소를 감시할 수 있어 가정이나 소규모 사업장에서 사용하기 적합함
16	pH 센서 (Hydrogen Exponent sensor)	수용액 중의 수소 이온 농도(pH)를 측정하는 장치
17	pH 컨트롤러 (Hydrogen Exponent Controller)	수용액 중의 수소 이온 농도를 제어해 주는 장치
18	RS232 통신 (Recommended Standard-232 communication)	데이터 단말 장치(DTE, Data Terminal Equipment)와 모뎀(modem) 또는 데이터 회선 종단 장치(DCE, Data Circuit-terminal Equipment)를 상호 접속하기 위해 EIA에서 개발한 표준 규격 - 비고 : 직렬 통신 방식 중 하나로 일반적으로 10m 정도의 범위에서 정상적인 데이터 통신을 할 수 있음
19	RS485 통신 (Recommended Standard-485 communication)	RS232, RS422 표준 규격의 확장 버전으로 EIA/TIA에서 개발한 다 대 다(N:N) 양방향 직렬 통신 표준 규격 - 비고 : 2가닥의 통신선으로 다수의 장치를 상호 연결하는 직렬 통신 방식의 하나로 온실 내에서 최대 1.2km 범위까지 신호전송이 가능하며, 주로 센서노드와 통합 제어 장치간의 송수신 방식으로 사용함
20	TDR 센서 (Time Domain Reflectometry Sensor)	토양의 수분 함량(%)을 측정하는 장치 - 비고 1 : 토양 수분의 유전율을 이용하는 방식으로 전자파가 반사되는 시간과 속도를 통해 수분 함량을 산정함 - 비고 2 : 토양의 수분 함량을 측정하는 또 다른 방식으로 FDR 센서 방식이 있음
21	WiFi (Wireless Fidelity)	'와이파이' 참조
22	ZigBee	저속, 저비용, 저전력의 양방향 단거리 무선망(WPAN, Wireless Personal Area Network) 기반의 홈네트워크나 무선 센서망에 사용되는 표준 기술 - 비고 1 : ZigBee Alliance에서 IEEE 802.15.4 물리계층(PHY, MAC) 표준 기술을 기반으로 상위 프로토콜 및 응용 프로그램을 표준화함 - 비고 2 : 세계 대부분의 지역에서 ISM(Industrial Scientific Mdedical) 대역인 2.4GHz 무선 주파수 대역에서 동작하며, 동일 주파수 대역을 사용하는 무선 랜(WLAN, Wireless Local Area Network)이나 블루투스(blueetooth) 기술과의 간섭을 피하기 위하여 868MHz(유럽), 915MHz(미국/호주) 대역도 사용됨. 변조방식은 직접 시퀀스 확산 스펙트럼(DS-SS, Direct Sequence Spread Spectrum)을 사용하며 데이터 전송 속도는 20kbps~250kbps임 - 비고 3 : 일반적으로 10m~20m 내외의 근거리 통신과 유비쿼터스 컴퓨팅을 위한 기술로서 휴대전화나 무선 랜과 같은 개념이며 전력소모를 최소화하는 대신 소량의 정보를 송수신하는 지능형 홈네트워크, 산업용 기기 자동화, 물류, 환경 모니터링, 휴먼 인터페이스, 텔레메틱스 및 군사용의 목적으로 활용

		됨
23	감우 센서 (rain detector)	비가 오는지를 감지하는 장치 - 비고 1 : 비가 올 경우에 필요에 따라 온실이나 축사의 창문을 닫게 하여 작물과 가축을 보호함 - 비고 2 : 감우 센서는 일반적으로 온실 외부의 기상대에 있는 경우가 많으며 직접 비나 눈을 맞지 않도록 하는 위험경보에 사용되거나 직접 천창 또는 측창 구동기에 신호를 보내 열거나 닫도록 함
24	강제 환풍 장치 (forced air circulating system)	창이 없는 온실 또는 축사 등의 실내 환경을 최적의 상태로 유지하기 위하여 벽체나 지붕에 환풍기를 설치하여 인위적으로 내부 온도를 조절하고 환기시키는 설비
25	개폐기 (switch)	전기 회로를 이었다 끊었다 하는 장치 - 비고 : 일반적으로 전등, 라디오, TV 등의 전기 기구를 손으로 누르거나 틀어서 작동하는 부분을 말함
26	건습구 온도계 (psychrometer)	공기의 습도를 재는 계기 - 비고 : 건구 온도계와 습구 온도계를 조합시킨 것으로 한 쪽 온도계를 젖은 헝겊 등으로 싸두면 마른 쪽의 온도계보다 그 온도가 내려가게 되며 이 온도차를 이용하여 공기의 상대습도를 측정함
27	게이트웨이 (gateway)	구조가 다른 두 개의 통신 네트워크를 연결하는 장치 - 비고 : 근거리 통신망과 같은 하나의 네트워크를 다른 네트워크와 연결할 때 그 출입구 역할을 함
28	결선 (wiring)	통신 장치나 기기, 구성 부품, 패널 또는 단자함이나 분전반 등을 상호 접속하여 회로나 회선을 구성하는 것 또는 그렇게 하기 위한 도선 - 비고 : 도선은 접속 장소, 방향, 부품, 회로의 종별에 따라 색깔이나 종류로 구별하여 사용됨
29	계전기 (relay)	전압, 전류, 전력, 주파수 등의 전기 신호를 비롯하여 온도, 빛 등 여러 가지 입력신호에 따라서 전기 회로를 열거나 닫거나 하는 구실을 하는 기기 - 비고 : 전력 시스템에 합선이나 접지 사고 등이 발생할 경우 또는 절연 파괴의 원인이나 계통의 나머지 부분에 영향을 주는 이상 상태가 생겼을 때, 이를 검출하고 차단기를 동작시켜 사고 부분을 신속히 계통에서 분리하도록 하는 일을 말하기로 함
30	관개 (irrigation)	'관수' 참조
31	관비 공급 장치 (fertigation system)	물과 비료액을 적당히 혼합하여 희석시킨 액체 상태의 비료를 작물에 공급하는 장치 - 비고 : 관수를 겸한 비료 공급이 가능하도록 보통 관수 장치에 액비 혼합 장치를 부착하여 사용함
32	관수 (irrigation)	작물 생육에 필요한 물을 주는 것 - 비고 1 : 시설 관수는 대부분 강우가 아닌 외부에서 공급된 물을 이용하므로 계획적인 관수가 가능하며, 유형으로는 고랑 관수, 지표 관수, 미스트 관수, 점적 관수, 지중 관수, 저면 급수 등이 있음 - 비고 2 : 작물의 종류와 생육 단계, 재배 시기 등을 고려하여 관수 시기를 결정해야 하며, 토양 수분의 상태에 따라 관수 시기를 결정하는 경우 대개 포장 용수량과 위조점 사이에서 관수를 해야 함
33	관수 펌프 (irrigation pump)	온실 내부로 물 등을 공급하기 위한 펌프
34	관수 밸브 (irrigation valve)	관수량을 조절해주는 장치 - 비고 : 수동식 밸브와 전자식 밸브가 있으며, 전자식 밸브 중 하나인 솔레노이드 밸브를 주로 사용함
35	광양자 센서 (light quantum sensor)	단위 면적에 대하여 단위 시간 동안 입사되는 광양자의 수(광양자속밀도, $\mu\text{mol}/\text{m}^2/\text{s}$ )를 측정하는 센서 - 비고 : 400~700nm 사이의 광양자속밀도를 '광합성 유효 광양자속밀도'(Photosynthetic Active Radiation, PAR)라고 함
36	광합성 (photosynthesis)	식물이 빛 에너지와 이산화탄소, 물을 이용하여 유기물을 합성하는 과정 - 비고 : 광합성은 유효파장의 빛이 셀수록, 이산화탄소가 많을수록 증가하지만 일정 수준이상으로 올라가면 더 이상 증가하지 않으며, 작물 생육에 피해를 줄 수 있음
37	교류 (AC, Alternating Current)	시간에 따라 크기와 방향이 주기적으로 변화하는 전류 - 비고 : 전류의 크기와 방향이 변하지 않는 전류는 직류(direct current)라 함
38	구동기	동력을 이용하여 기계 등을 동작시키는 구동 장치

	(actuator)	- 비고 : 온도, 습도와 같은 온실 내 환경 상태를 조절하기 위하여 전기, 유압 또는 기계적 신호 등에 따라 구동되는 장치
39	구동기노드 (actuator node)	구동기와 통신 모듈이 결합된 구조로서, 온실 통합 제어기의 제어 명령에 따라 구동기를 동작시키기 위한 장치
40	구역 (zone 또는 area)	농장 등에서 동일한 조건으로 생산 관리를 하는 관리 단위
41	급수 (water supply)	펌프 등을 통해 물을 대어 주는 것 또는 그 물
42	급액 (nutrient feed)	작물에 양액을 공급해 주는 일 - 비고 : 작물의 생육에 필요한 무기성분을 함유한 물을 양액이라고 함
43	기계 장치 (mechanical equipment)	기계를 설치하는 일 또는 그런 장치 - 비고 : 동력에 의하여 작업을 행하는 발전기, 전동기, 공작 기계, 작업 기계 등
44	냉난방기 (cooler and heater, air conditioner)	온실 내부의 온도 및 습도를 조절하기 위한 장치
45	노드 (node)	데이터 통신망에서 데이터를 전송하는 통로에 접속되는 하나 이상의 기능 단위로서 주로 통신망의 분기점이나 단말기의 접속점을 말함 - 비고 : 온실 통합 제어기와 디바이스(센서 또는 구동기)간 통신을 가능하게 하는 매개체
46	농도 (concentration)	용액이나 기체의 진하고 묽은 정도 - 비고 : 일반적으로 %, mol/l, N 등의 단위가 사용되며 대기 오염, 수질 오염의 분석 관계에서는 ppm, mg/l 등이 사용됨
47	다용도 입출력 포트 (GPIO, General Purpose Input/Output)	'GPIO (General Purpose Input/Output)' 참조
48	단상 (single phase)	하나의 위상을 갖는 교류로서, 하나의 교류 전압과 그에 대응하는 전류만이 있는 경우를 말함
49	단순 환경 제어 (simple environmental control)	온실을 관리하는데 있어 1~2가지 요인만 고려한 환경 제어 방식 - 비고 : 주로 온도(적정 온도 유지), 시간 주기(일정 시간 작동) 등 비교적 제어가 쉬운 요인만을 제어하는 방식
50	동력선 (power line)	전동력 기계에 전력을 공급하는 배전선
51	디바이스 (device)	어떤 특정한 목적을 위하여 구성한 기계적, 전기적, 전자적인 장치 - 비고 : 온실 내외의 환경 정보를 수집하거나 생육 환경을 제어하는 장치로 그 기능과 역할에 따라 센서 또는 구동기로 분류할 수 있음
52	로라 (LoRa, Long Range)	사물 상호간 통신(IoT)을 위한 저전력의 장거리 통신(LPWA, Low Power Wide Area) 기술
53	발아율 (germination rate)	일정한 양의 씨앗 가운데 발아할 수 있는 씨앗의 비율, 또는 뿌린 씨앗에 대하여 발아한 씨앗의 비율
54	배액 (drainage)	급액 후 작물에 흡수되지 못하고 배출되는 양액 - 비고 : 배액률(%) = (배액량/급액량) x 100
55	배지 (culture medium)	식물이나 세균, 배양 세포 등을 기르는데 필요한 영양소가 들어 있는 액체나 고체상태의 것 - 비고 1 : 식물의 뿌리를 고정 또는 지지하기 위해 사용되며 온실에서는 펄라이트, 암면, 코코섬유, 왕겨 등이 주로 이용됨 - 비고 2 : 생물은 생존·발육에 불가결한 물을 비롯하여 영양물질로서 다량요소·미량요소 등을 요구한다. 그 중 기체상으로 얻어지는 것을 제외하고는 모두 무기 또는 유기화합물로서 배지에 공급해 주어야 한다. 필요한 화합물은 독립영양·종속영양 등 영양 형식에 따라 여러 가지이다. 보통 영양원을 탄소원·질소원·무기염류·발육인자(비타민류) 등으로 나누어서 생각한다. 특히 발육인자와 관련하여 생물체에서 추출한 비교적 복잡한 조성을 가진 것을 주체로 한 경우를 천연배지(天然培地)라고 한다
56	보온재 (heat insulating material, heat insulator)	열을 잘 옮기지 않고 보온력이 풍부하여 단열재로 쓰는 재료로서 보온 또는 보냉의 목적으로 사용됨 - 비고 : 재질에 따라 유기질 보온재(저온용), 무기질 보온재(고온용)로 나눌 수 있으며 유기질에는 콜크, 톱밥, 폴리스틸렌, 연질섬유판, 펠트 등이 있고 무기질에는 석면, 암면, 글라스울, 구조토, 염기성 탄산마그네슘 등이 있음
57	복합 환경 제어 (complex environmental)	복합적인 환경 요소간의 상호 영향을 고려하거나 서로 다른 환경 요소를 변수로 하여 환경 요소의 설정치를 정하는 환경 제어 방식

	control)	- 비교 : 작물 생육에 영향을 주는 환경 요인(온도, 광도, CO <sub>2</sub> , 습도 등)을 복합적으로 제어하여 작물에 적합한 환경을 구현하는 방법
58	블루투스 (Bluetooth)	IEEE 802.15.1에서 표준화된 무선 통신 기기간 근거리, 저전력 무선 통신 표준 기술 - 비교 : 휴대폰, 노트북, 이어폰 또는 헤드폰 등의 휴대기기를 서로 연결하고 정보를 교환하는 근거리 무선 기술로서 주로 10m 내외의 초단거리에서 저전력의 무선 연결이 필요한 경우에 쓰임
59	사물인터넷 (IoT, Internet Of Things)	정보 통신 기술(ICT, Information and Communication Technologies)을 기반으로 실세계(physical world)와 가상 세계(virtual world)의 다양한 사물들을 연결하여 보다 진보된 서비스를 제공하기 위한 시범 시설 - 비교 1 : 인터넷을 기반으로 센서, 프로세서 등이 장착된 사물을 연결하여 사람과 사물, 사물과 사물간 정보를 수집하고 제어, 관리할 수 있는 지능형 기술 및 서비스 - 비교 2 : 사물 인터넷의 주요 기술로는 센싱 기술, 유무선 통신 및 네트워크 인프라 기술, 사물 인터넷 인터페이스 기술, 사물 인터넷을 통한 서비스 기술 등이 있음
60	산가 (acid value)	산성 화합물을 중화하는데 필요한 염기의 질량 - 비교 1 : 용액 속에 해리된 탄산, 유기산 등의 산 성분의 중화에 이용하는 알칼리 성분을 이에 상당하는 수산화칼륨(KOH) 양으로 나타낸 값으로 ppm으로 나타내며, 1ppm을 1도로 함 - 비교 2 : '수소 이온 농도' 참조
61	삼상 (three Phase)	세 개의 위상을 갖는 교류로서, 전압과 주파수는 같지만 위상이 서로 120°씩 다른 세 가지 전류나 전압을 한 조로 한 것 - 비교 : 교류를 사용하는 전원에는 3개의 전원용 단자(R상, S상, T상)로 구성되며, 일반적으로 발전소 등의 전기시설에서 사용됨
62	상대 생장률 (relative growth rate, RGR)	일정 기간 동안의 식물의 성장량을 중량으로 나눈 값 - 비교 : $RGR = \frac{1}{\text{개체의 건조중량}} \times \frac{\text{건조중량의 증가량}}{\text{단위시간}}$
63	상대 습도 (relative humidity)	일정 부피의 공기 속에 실제로 포함되어 있는 수증기의 양과 최대로 포함할 수 있는 수증기 양(포화 수증기 양)과의 비율(%)로서 일반적으로 습도라고 함
64	생육 (growth)	생물이 나서 길러지는 것 - 비교 : 생물이 발아하여 성숙할 때까지 달하는 과정으로 광합성 작용, 호흡 작용, 세포의 분열과 성장, 증산 작용 등을 통해 크기와 무게가 증가함
65	생육 모델 (growth model)	작물의 연령이나 시간적인 경과에 따른 생장 인자의 변화를 수리적 형식으로 표현한 것. 생장 모델이라고도 함 - 비교 : 작물의 생명 주기 내에서 최적의 환경 조건을 제공하기 위한 모델
66	생육 정보 (growth data)	생육 환경의 분석 및 최적 생육 환경 조성을 위해 주기적이고 지속적으로 수집되는 정보 - 비교 : 주로 초장(작물의 키), 줄기 직경, 엽장, 엽폭, 착과수, 엽면적 등이 있음
67	생체중 (fresh weight)	작물의 성장에 따른 전체 작물의 무게(g)
68	센서 (sensor)	빛, 온도, 상대 습도 등의 물리적인 양을 측정하는 소자 또는 그 조사를 갖춘 장치 - 비교 : 환경 정보를 수집하는 장치로서 온도 센서, 습도 센서, CO <sub>2</sub> 센서, 풍향 센서, 감우 센서, EC 센서 또는 pH 센서 등을 말함
69	센서노드 (sensor node)	센서와 통신 모듈이 결합된 구조로서 측정값을 제어기 등에 전달하는 장치 - 비교 : 센서의 출력 신호를 전압 또는 전류를 변환하는 모듈과 제어장치와의 유무선 데이터 송수신 기능을 갖는 것으로 무선 센서 네트워크를 구성하는 기본 요소로 센싱 기능, 전산 기능, 양방향 유무선 통신 기능과 전원장치를 포함
70	솔레노이드 밸브 (solenoid valve)	전자밸브의 한 종류로서 전자 코일의 전자력을 사용하여 자동으로 밸브를 개폐시키는 것 - 비교 : 관수 조절을 위하여 주로 사용되며, 전기가 통하면 플랜지가 올라가 밸브가 열리고, 전기가 차단되면 플랜지 무게에 의하여 자동적으로 밸브가 닫힘
71	수소 이온 농도 (pH; Potential of Hydrogen)	용액 속에 해리된 수소 이온이나 옥소늄 이온의 농도를 말하며, 일반적으로 수소 이온 농도의 지수로서 용액의 산성 또는 알칼리성의 정도를 나타냄 - 비교 1 : 25°C의 온도에서 비로 성분 등을 전혀 함유하지 않은 순수한 물의 산도(pH)를 7(중성)이라 하고 이보다 낮은 경우를 산성, 높은 경우를 알칼리성이라 함

		- 비고 2 : 측정하는 위치나 대상에 따라 공급 산도, 배액 산도 등으로 분류되며, 양분의 흡수와 밀접한 관계가 있는 것으로 대부분의 작물은 토양 재배 시 6.2~6.8의 산도, 수경 재배 시 5.5~6.2의 산도 범위에서 가장 생육이 좋음
72	스마트 영상 장치 (smart video processing device)	촬영 장치(카메라), 전송 장치, 저장장치, 재생장치 등으로 구성된 영상 장치 - 비고 1 : 온실 내부, 외부의 환경 제어 장치의 작동 상황 및 외부 침입 감지 등을 확인하거나 생육 상태, 병해충 피해 등 작물의 생육 정보 수집을 목적으로 함 - 비고 2 : 일종의 폐쇄 회로 텔레비전(CCTV, Closed Circuit TeleVision) 장치로서 최근 인터넷 프로토콜(IP, Internet Protocol) 기반의 네트워크 카메라, IP 카메라 시스템을 포함함
73	스마트 온실 (smart greenhouse)	작물의 생장·수량을 극대화하기 위하여 환경 요소(빛, 온도, 습도 등)를 최적으로 조절할 수 있으며, 이를 원격화·자동화한 온실을 말함 - 비고 : 어디서나 작물의 생육 환경을 점검하고 이를 적정하게 유지, 관리하기 위하여 정보통신기술(ICT)이 많이 활용되고 있음
74	스마트팜 (smart farm)	스마트 온실의 원격화·자동화·최적환경 제어를 과수원, 축사 등으로 확장한 개념으로 농업 전반에서 개발되고 있음. - 비고 : 농업의 생산뿐만 아니라 수확 후 가공·유통 단계까지 확장되고 있음
75	스마트팜 온실 (smart farm greenhouse)	'스마트 온실' 참조
76	습도 센서 (humidity sensor)	습도를 측정하는 장치 - 비고 : 습도는 공기 중에 수증기가 포함된 정도를 말하는 것으로 작물이나 가축의 생육에 적정 범위가 존재하며, 작물의 증산이나 병충해, 가축의 스트레스에 직접적인 영향을 미침
77	시설 원예 (controlled horticulture)	온실에서 작물의 생육에 필요한 양분, 빛을 공급·조절하여 채소, 과수, 화훼 등을 재배하는 것 - 비고 : 적극적인 환경 조절을 통해 노지 등에서 재배가 불가능한 시기 또는 노지보다 더 나은 환경 조건을 통해 고품질 원예 작물을 생산하는 것
78	식물공장 (plant factory)	외부와 독립된 시설 안에서 빛, 온도, 습도 등의 환경을 인위적으로 조성하여 농산물을 생산하는 시스템
79	액추에이터 (actuator)	'구동기' 참조
80	양액 (nutrient solution)	작물의 생육에 필요한 무기성분을 함유한 물
81	양액기	'관비 공급 장치' 참조
82	양액 시스템 (nutrient solution supply system 또는 fertigation system)	관비 공급 장치 또는 양액기와 이를 제어하는 제어가 포함된 시스템 - 비고 : '관비 공급 장치' 또는 '양액기' 참조
83	엽록소 (chlorophyll)	빛 에너지를 유기 화합물 합성을 통하여 화학 에너지로 전환시키는 녹색 색소 - 비고 1 : 광합성에 가장 중요한 요소로 빛에서 에너지를 흡수하여 이산화탄소를 탄수화물로 전환시킴. 녹색식물, 조류(藻類), 광합성 박테리아 등의 광합성을 하는 모든 생물체에서 나타남 - 비고 2 : 식물 등의 잎에 있는 엽록체(chloroplast) 속에 황색의 카로틴 및 크산토피과 공존하는 녹색의 호흡 색소 식물의 녹색 색소로 태양의 방사 에너지를 흡수하여 광합성을 행함. 구조적으로 폴리피린류에 속하며 철 대신에 마그네슘을 가짐 - 비고 3 : 육상 식물에서는 엽록소 a 및 엽록소 b, 일부 해양 식물에서는 엽록소 c, 광합성 세균에서는 박테리오 엽록소 등이 주요한 엽록소에 해당함
84	엽록체 (chloroplast)	식물 잎의 세포 안에 함유된 둥근 모양 또는 타원형의 작은 구조물로서 엽록소를 함유하는 색소체로 녹색을 띠고 있으며 탄소 동화 작용을 하여 녹말을 만드는 중요한 부분 - 비고 1 : 이중막으로 둘러싸여 있으며 투명한 스트로마와 동전을 쌓아 둔 모양의 그라나로 이루어짐 - 비고 2 : 엽육 세포에는 보통 수 십~수 백 개의 엽록체가 있으며, 엽록체에는 비타민K, 비타민 C, 리보산, 망간 등이 존재하고 누에 장 내에서 미생물의 증식을 억제하는 것으로 알려짐
85	엽면적 (leaf area)	식물체의 잎의 면적(cm <sup>2</sup> )을 나타내는 값 - 비고 : 주로 하나의 개체(주)의 전체 잎의 면적을 말함
86	엽면적 지수 (LAI, Leaf Area Index)	단위 면적당 엽면적의 비율

87	엽색 (leaf color)	식물체가 가진 잎의 색깔 - 비교 : 식물체의 영양상태나 병해충 피해에 따라 발현되므로 이를 통한 진단이 가능함
88	엽수 (number of leaves)	식물체에 달린 잎의 수
89	엽온 (leaf temperature)	식물 잎의 온도 - 비교 : 잎이 햇볕에 직접 노출되면 다량의 일사 에너지를 흡수하여 잎의 온도가 높아지면서 잎에서 증산 작용이 일어나고 기화열을 통해 잎의 온도가 내려가는 현상을 바탕으로 잎에서의 증산 작용을 파악하는 자료로 사용함
90	엽온 센서 (leaf temperature sensor)	엽온을 측정하는 센서 - 비교 : 비접촉식과 접촉식이 있음
91	엽장 (leaf length)	잎자루를 제외한 잎의 길이 - 비교 : 잎의 최대 길이로서 엽면적을 계산할 때 사용됨
92	엽폭 (leaf width)	잎의 너비 - 비교 : 잎 길이(엽장)의 생장 방향에 수직으로 제일 넓은 부위의 값으로 엽면적을 계산할 때 사용됨
93	온도 센서 (temperature sensor)	온도(°C)를 측정하는 장치 - 비교 : 온실 등의 내부 온도는 세포의 분화, 분열과 신장, 작물의 호흡량과 관련되므로 작물 재배에 중요한 요소임
94	온실 (greenhouse)	식물의 주요 생육 환경인 빛이나 온도, 습도 등을 인위적으로 조절하기 위한 외부와 격리된 공간(비교 : '스마트 온실' 참조) - 비교 1 : 동절기에 식물을 재배하거나 한대기후 지역에서 온대기후 지역에서 자생하는 식물을 재배할 수 있고, 환경 조절을 통해 개화, 결실 등을 조절할 수 있어 축성 재배와 역제 재배가 가능함 - 비교 2 : 식물의 생육 환경을 개선하기 위해 재배 공간을 대기 환경과 격리한 식물 재배 시설로서, 피복 자재의 종류에 따라 유리 온실, 플라스틱 온실로 구분하고 지붕의 형식에 따라 양지붕형, 외지붕형, 3/4형, 벨로형 온실로 구분됨. 동의 수와 형식에 따라 단동형과 연동형이 있음
95	온실 운영 시스템 (greenhouse operating system)	온실 내부 및 외부의 센서로부터 환경과 생육 정보를 모니터링하고, 온실 환경 제어 알고리즘 운영 및 제어, 수집된 데이터를 데이터베이스에 기록하는 장치 - 비교 : 데이터 베이스에 저장된 데이터를 바탕으로 영농 일지를 작성할 수 있으며, 온실 운영에 필요한 서비스 및 제어 소프트웨어를 온실 통합 관리 시스템으로부터 다운로드하여 사용할 수 있음. 또한 센서노드 및 제어 노드의 동작 주기와 장애 등을 관리함
96	온실 통합 관리 시스템 (integrated greenhouse management system)	온실 관리(관제)에 필요한 소프트웨어를 개별 온실 운영 시스템에 설치하고 외부의 데이터 서버와 연동하여 복수의 온실 운영 시스템으로 작물의 생육 정보 등을 피드백하는 시스템 - 비교 : 센서노드 및 제어 노드에 따라 필요한 소프트웨어를 온실 운영 시스템에 설치해 주는 역할을 하며, 관제 기능을 강조하는 경우에는 '온실 통합 관제 시스템(integrated greenhouse management and control system)'으로 부르기도 함
97	온실 통합 제어기 (integrated greenhouse controller 또는 integrated greenhouse gateway)	온실 운영 시스템으로부터의 명령을 제어 노드에 전달하여 온실의 각종 구동 장치를 제어하거나 센서노드로부터의 센싱 값을 온실 운영 시스템으로 전달하는 제어 장치 - 비교 : 프로토콜을 변환하는 게이트웨이 기능을 갖거나 온실 운영 시스템과 통합하여 운용할 수 있음
98	와이파이 (WiFi, Wireless Fidelity)	IEEE 802.11 기반의 무선랜(Wireless LAN) 표준기술 - 비교 : 무선 접속 장치(AP, Access Point)가 설치된 곳에서 전파나 적외선 전송 방식을 이용하여 일정 거리 안에서 무선 인터넷을 할 수 있는 근거리 통신망 기술로서 온실 외부에서 원격 제어를 통해 온실 관리를 할 때 사용함
99	유동 팬 (flow fan 또는 circulation fan)	온실 내부의 공기 순환을 통해 온도 및 습도를 조절하는 장치
100	유량 (flow rate)	유체(流體)가 단위 시간 동안 흐르는 양 - 비교 : 유체가 흐를 때 일정 면적의 단면을 통과하는 유체의 체적, 질량 또는 중량을 시간에 대한 비율로 표시한 것
101	유량 센서 (flow rate sensor)	하천, 수로 또는 관 등에 흐르는 유량(G(kgf/s) 또는 Q(m <sup>3</sup> /s))을 측정하는 센서 - 비교 : 유체(流體)의 유속은 기벽에서의 거리에 따라 상이하므로 일정 지점의 통과 전후에서 압력이나 수위의 변화를 측정하여 계산함

102	인터페이스 (interface)	서로 다른 두 시스템, 장치, 소프트웨어 등을 서로 이어주는 부분 또는 그런 접속 장치 - 비고 : 둘 이상의 구성 요소 경계에 있어 공용되는 부분으로서 기계적인 결합부에서는 커넥터를 뜻함
103	일사량 (amount of solar radiation 또는 amount of insolation)	태양의 복사 에너지가 지표에 닿는 양으로 위도에 따라 다름 - 비고 1 : 태양 광선에 직각으로 놓인 단위 면적에 단위 시간 동안 복사되는 에너지의 양( $W/m^2$ )으로 작물의 광합성 양을 추측하거나 온실이나 축사 내 온도 제어, 관수 제어 등에 활용됨 - 비고 2 : 특정 기간 동안 누적된 총 일사량을 누적 일사량이라 함
104	일사 센서 (pyranometer)	단위 시간에 대하여 단위 면적당 입사되는 일사 에너지( $W/m^2 \cdot h$ )를 측정하는 장치 - 비고 1 : 일사 센서에는 직달 일사 센서, 전천 일사 센서 등이 있으며 측정 위치에 따라 온실 내부 일사와 온실 외부 일사로 구분됨 - 비고 2 : 온실이나 축사 내 온도 제어, 관수 제어 등에 활용됨
105	작물 생육 정보 (growth and development information)	초장, 엽수, 엽면적, 엽록소 함유량이나 과실의 수, 색, 당도나 무기성분 함량, 생체중, 수확량 등의 작물 생육의 정도를 추정할 수 있는 정보
106	전기 전도도 (EC, Electric Conductivity)	물질이나 용액이 전하를 운반할 수 있는 정도 즉, 도체에 흐르는 전류의 크기를 나타내는 상수( $dS/m$ )를 말하는 것으로 온도가 상승하면 금속에서는 크기가 감소하고 반도체에서는 크기가 증가함 - 비고 1 : 토양 관개수의 전기 전도도는 작물에 대한 염류 장애를 판단하는데 매우 중요한 화학적 지표로 사용되며, 측정위치나 대상에 따라 배액의 전기 전도도, 급액의 전기 전도도 등으로 분류할 수 있음 - 비고 2 : 수용액에 해리된 이온이 많을수록 전기 전도도가 증가하며 양분 농도의 간접적인 지표로 사용됨. 식물은 일반적으로 2~4mS/cm의 범위에서 재배하는 것이 좋으나 이 이상일 경우에는 위조, 생육억제, 열과 등의 현상이 일어나기 쉬움
107	전자 밸브 (electrimagnetic valve)	'솔레노이드 밸브' 및 '관수 밸브' 참조
108	조도 센서 (illuminance sensor)	단위 시간에 대하여 단위 면적당 광속밀도( $dp/dS$ )를 측정하는 장치. 단위는 룩스(lux, lx) - 비고 : 포토레지스터(photoresistor) 또는 CdS(Cadmiumsulfide)로도 불리는 조도 센서는 빛의 밝기에 따라 저항값이 변하는 특징을 가짐
109	지습 센서 (soil-humidity sensor)	땅의 습한 정도를 측정하는 장치 - 비고 : 온실 내 토양, 배지, 양액 등의 작물의 근권부의 습도를 측정하는 장치로서 작물에 따라 적절한 지습을 유지하는 것이 중요함
110	지온 센서 (soil-temperature sensor)	근권부의 온도( $^{\circ}C$ )를 측정하는 장치 - 비고 : 온실 내 토양, 배지, 양액 등 작물의 근권부 온도를 측정하는 장치로서 지온은 작물의 뿌리 생육에 영향을 미치므로 작물에 적합한 범위를 유지해야 함
111	창 개폐기 (window switch)	천창 또는 측창이나 보온 커튼 등의 개폐에 이용하는 동력 개폐기 - 비고 : 형식이나 감속 방식에 따라 암식, 랙피니언 식, 권취식 또는 견인식 등으로 분류함
112	천창 (top window)	온실 등의 시설물 지붕이나 천장면에 설치한 환기창
113	측창 (side wall window)	온실 등의 시설물 측면에 설치한 환기창
114	커튼 (curtain)	창이나 문에 치는 휘장으로 온실 등의 시설 원예에서 빛이나 열을 차단하기 위한 목적으로 사용되며, 측창 또는 천창과 동일한 의미를 가질 때도 있음
115	토양 산도 (soil acidity)	토양의 산도(산의 세기) - 비고 : 흙이 산성이나 중성 또는 알칼리성 등의 성질을 나타내는 것을 말하며, 작물에 따라 다르나 대개 중성의 흙(pH 5.5-6.5)이 작물 재배에 가장 적합하다고 알려져 있음
116	토양 수분 장력 (soil moisture tension)	토양 입자의 표면과 토양 수분간에 작용하는 인력으로 토양 수분이 갖는 에너지는 자유 수분보다 적어지는데 이러한 흡인력을 압력 단위(kPa)로 표시한 것 - 비고 1 : 토양의 함수율과 달리 수분의 포센셜을 말하는 것으로 적절한 범위를 유지하는 것이 작물 생육에 좋음 - 비고 2 : 토양 속의 물(수분)은 토양 입자에 강하게 흡착된 것, 약하게 흡착된 것, 공극을 자유롭게 이동하는 것으로 나눌 수 있으며, 토양 입자와 물 사이에 작용하는 힘의 강도를 그에 대응하는 압력, 수주 높이, pF 등으로 표시

		하는데 이는 토양에서 그 수분을 분리하기 위한 압력의 크기로서 표현됨
117	토양 수분 장력 센서 (tensiometer)	토양 수분 장력(kPa)을 측정하는 장치 - 비교 : 토양 함수율 센서와 달리 토양 입자의 표면과 토양 수분 간에 작용하는 인력을 압력 단위로 표시한 것
118	토양 장력 센서 (tensiometer)	'토양 수분 장력 센서' 참조
119	토양 함수율 센서 (soil-moisture sensor)	단위 부피당 토양에 포함된 수분량(%)을 측정하는 장치
120	통합 제어 장치 (integrated greenhouse controller)	'온실 통합 제어기' 참조
121	팬/틸트/줌 (PTZ; Pan/Tilt/Zoom)	제어 신호를 이용하여 영상 장치(카메라 등)를 감시하고자 하는 곳으로 상, 하, 좌, 우로 이동시켜 피사체를 포괄적으로 감시 및 관찰할 수 있도록 제작된 모터, 클러치 등의 기기 - 비교 : Pan(회전), Tilt(기울기), Zoom(줌) 조정이 가능한 일체형 카메라 장치
122	폐액 (waste fluid)	더러워서 버리는 물 또는 용액, 일정 사용 목적이 끝나고 폐기된 액체
123	풍속 센서 (wind speed sensor)	바람의 세기(m/s)를 측정하는 장치 - 비교 : 온실이나 축사의 구조물 안전 관리, 환기량 예측, 온도 및 습도 관리를 위한 환기장치(천창, 측창) 조절 등의 목적에 사용됨
124	풍향 센서 (wind-direction sensor)	바람의 방향(°)을 측정하는 장치 - 비교 : 온실이나 축사의 온도, 환기장치 관리 등의 목적으로 사용됨
125	함수율 (moisture content)	함유하고 있는 수분의 비율(%) - 비교 1 : 일반적으로 식물체의 함수율은 중량 비율로, 토양이나 배지의 함수율은 용적 비율로 계산됨 - 비교 2 : 목재의 경우 섬유 포함점 이하에서는 함수율이 적을수록 강도가 증가됨
126	협대역 사물 인터넷 (NB IoT, NarrowBand-Internet Of Things)	데이터 통신에서 협대역을 이용하여 전력 소비가 적은 광역 통신을 지원하는 사물 인터넷 표준 기술 (비교 : '사물 인터넷') - 비교 : 저전력 광역(LPWA, Low Power Wide Area) 통신을 지원하는 협대역 사물 인터넷 표준으로, GSM 또는 LTE 망에서 협대역을 이용하여 수 백 kbps 이하의 데이터 전송 속도와 10km 이상의 광역 서비스를 지원하며, 수도 검침, 위치 추적용 기기 등의 목적에 사용됨
127	환경 정보 (environmental information)	체계적인 분류 기준과 코드화 작업을 통해 업무에 효율적으로 이용할 수 있도록 한, 자연 환경이나 생활환경 등에 대한 자료 - 비교 : 온실 내부 또는 외부의 센서로부터 수집된 센싱 값 정보와 구동기로부터 수집된 값 등 온실 환경과 관련된 측정 정보
128	환경 제어 (environmental control)	지상부 환경(광, 온도, 습도, CO <sub>2</sub> 가스 등) 및 지하부 환경 요인(배지 이·화학적, 양분 조성 등)을 제어하여 작물 생육에 적합한 환경을 구현하는 것
129	환풍기 (ventilator)	실내의 공기를 외부의 공기와 바꾸어 내부 공기를 환기시키기 위한 기구. 환기팬 - 비교 : 온실의 앞쪽 또는 뒤쪽에 설치하여 외부의 바람을 받아들이고 내부의 공기를 외부로 배출하여 온실 내 온도나 습도를 조절하는 장치로 일반적으로 프로펠러 모양의 팬이 달려 있음

나-2) 스마트팜 용어정의-제2부 : 축사(1종) ('19 제안, '20 제정)

스마트팜(smart farm)은 원예, 축산, 노지 등의 농업 분야에 ICT(Information and Communication Technologies)를 접목하여 원격에서 자동으로 작물이나 가축 등의 최적 생육 환경을 유지하고 관리할 수 있게 한다. 이를 통해 사용자에게 편의성을 제공할 수 있을 뿐만 아니라 빅데이터(big data) 구축을 통한 과학 영농을 가능하게 하고, 나아가 생산량과 소비량 등의 정확한 예측을 통하여 농가의 수익을 향상시킬 수 있는 기술이다. 이와 같이 농업과 ICT 분야가 융합된 스마트팜 기술이 도입되면서 각 분야에서 사용하는 용어에 대한 이해의 차이로 인한 불필요한 혼란이 가중되고 있으며, 이는 스마트팜 기술과 서비스의 확산과 보급에 걸림돌로서 작용하고 있다. 이에 이 표준은 정부와 산업계, 연구계 등에서 사용하는 스마트팜 관련 용어, 특히 스마트 축사 분야에 대한 다양한 용어를 체계적으로 정의함으로써 농장주와 농장의 종사자는 물론 스마트팜 관련 장비 제조업자 및 시설업자, 서비스 제공업자의 이해를 증진시키는 것을 목적으로 표준화를 추진하였으며, 2020년도 한국정보통신기술협회(TTA)의 단체표준으로 제안하여 12월에 최종 채택될 예정이다.

스마트팜과 관련된 국·내외 표준 문서뿐만 아니라 관련 부처 및 국립국어원 등 정부 및 공공기관의 발행 자료를 조사하여 스마트팜 기자재, 농업 생산 기술 및 ICT와 관련된 용어를 선별하였다. 이를 통하여 동일 용어에 대한 서로 다른 정의를 일원화하여 다양한 분야에서 공통적으로 적용할 수 있는 용어 정의를 제시한다. 또한 각 용어가 서로 다른 분야에서 효과적으로 사용될 수 있도록 구체적인 부연 설명과 세부 정의를 함께 제공하고 있다. 이 표준은 2019년도 추진한 스마트 온실 및 시설원예 분야에 대한 용어정의 표준으로 개발된 '스마트팜 용어 정의 - 제1부: 온실'(TTAK.KO-10.1164-part 1) 표준의 후속 패밀리 표준으로서 그 의미가 있으며, 향후 축사와 관련된 세부 영역 분야(예를 들어 양돈, 양계 또는 한우 등)에 대한 표준 용어 개발과 빅데이터, 클라우드 등 스마트팜 산업에 적용되는 전문 ICT 용어에 대한 표준을 수립함에 있어 기초자료로서 활용할 계획이다.

표 1 - 스마트팜(축산분야) 용어 정의 현황

순번	용어	정의
1	1A	전자기 릴레이에서 A접점이 1극인 릴레이
2	1A1B	전자기 릴레이에서 A접점이 1극, B접점이 1극인 릴레이
3	2A2B	전자기 릴레이에서 A접점이 2극, B접점이 2극인 릴레이
4	CO <sub>2</sub> 센서 (Carbon Dioxide Sensor)	공기 중의 CO <sub>2</sub> (이산화탄소)의 농도(μmol/mol(ppm))를 측정하는 장치
5	O <sub>2</sub> 센서 (oxygen sensor)	공기 중의 O <sub>2</sub> (산소)의 농도(μmol/mol(ppm))를 측정하는 장치
6	RFID (Radio Frequency Identification)	전파 신호를 통해 비접촉식으로 사물에 부착된 얇은 평면 형태의 태그(tag)를 식별하여 정보를 처리하는 기술 또는 그런 시스템 - 비교 1: RFID 시스템은 판독 및 해독 기능을 하는 판독기(reader)와 고유 정보를 내장한 태그(tag), 응용 소프트웨어 및 네트워크로 구성됨 - 비교 2: RFID 태그는 전원을 필요로 하는 능동형(active type)과 판독기의 전자기장에 의해 작동하는 수동형(passive type)으로 구분할 수 있음
7	RS232 통신 (Recommended Standard-232 communication)	데이터 단말 장치(DTE, Data Terminal Equipment)와 모뎀(modem) 또는 데이터 회선 종단 장치(DCE, Data Circuit-terminal Equipment)를 상호 접속하기 위해 EIA에서 개발한 표준 규격 - 비교: 직렬 통신 방식 중 하나로 일반적으로 10m 정도의 범위에서 정상적인 데이터 통신을 할 수 있음
8	RS485 통신 (Recommended Standard-485 communication)	RS232, RS422 표준 규격의 확장 버전으로 EIA/TIA에서 개발한 다 대 다(N:N) 양방향 직렬 통신 표준 규격 - 비교: 2 가닥의 통신선으로 다수의 장치를 상호 연결하는 직렬 통신 방식의 하나로 축사 내외에서 최대 1.2km 범위까지 신호전송이 가능하며, 주로 센서 노드와 통합 제어 장치간의 송수신 방식으로 사용함
9	Wi-Fi(Wireless Fidelity)	'와이파이' 참조
10	ZigBee	저속, 저비용, 저전력의 양방향 단거리 무선망(WPAN, Wireless Personal Area Network) 기반의 홈네트워크나 무선 센서망에 사용되는 표준 기술 - 비교 1 : ZigBee Alliance에서 IEEE 802.15.4 물리계층(PHY, MAC) 표준 기술을 기반으로 상위 프로토콜 및 응용 프로파일용 표준화함 - 비교 2 : 세계 대부분의 지역에서 ISM(Industrial Scientific Mmedical) 대역인 2.4GHz 무선 주파수 대역에서 동작하며, 동일 주파수 대역을 사용하는 무선 랜(WLAN, Wireless Local Area Network)이나 블루투스(blueetooth) 기술과의 간섭을 피하기 위하여 868MHz(유럽), 915MHz(미국/호주) 대역도 사용됨. 변조방식은 직접 시퀀스 확산 스펙트럼(DS-SS, Direct Sequence Spread Spectrum)을 사용하며 데이터 전송 속도는 20kbps~250kbps임 - 비교 3 : 일반적으로 10m~20m 내외의 근거리 통신과 유비쿼터스 컴퓨팅을 위한 기술로서 휴대전화나 무선 랜과 같은 개념이며 전력소모를 최소화하는

		대신 소량의 정보를 송수신하는 지능형 홈네트워크, 산업용 기기 자동화, 물류, 환경 모니터링, 휴먼 인터페이스, 텔레메틱스 및 군사용의 목적으로 활용됨
11	가스 검지관 (gas detector tube)	가스에 함유된 유해성분을 검출하기 위한 관 - 비교: 유해 성분의 농도는 검지체의 변색된 길이 또는 정도로 판단하며, 정확성이 높지는 않으나 사용이 간편하여 현장에서 많이 사용됨
12	가스 세정기 (gas scrubber)	가스에 포함된 고체 또는 액체 등의 불순물 성분을 제거하기 위한 장치
13	가스 센서 (gas sensor)	가스를 감지하기 위한 장치 - 비교: 감지하는 방식은 가스의 종류와 농도 등에 따라 다양함
14	가스 열량계 (gas calorimeter)	기체 연료와 같은 가스의 발열량을 측정하는 장치 - 비교: 일반적으로 Junkers type 열량계가 많이 사용되며, 가스를 일정한 속도로 통과시켜 태우면서 열량계에 물을 일정한 속도로 통과시켜 입구와 출구에서 물의 온도 변화에 따른 발열량을 측정함
15	가축 영양 관리 (livestock nutrition management)	가축이 건강을 유지하고 생산을 충분히 하여 가진 바의 유전능력을 발휘할 수 있도록 가축이 필요로 하는 영양소를 알맞게 공급하는 관리적 절차 - 비교: 넓은 의미로 사료관리, 품종관리, 축사 환경관리 등 가축 사육의 최적 환경을 위한 전반적인 절차를 의미함
16	간이 소음계 (simple sound level meter)	'소음계' 참조
17	감우 센서 (rain detector)	비가 오는지를 감지하는 장치 - 비교 1: 비가 올 경우에 필요에 따라 축사의 창문을 닫게 하여 가축을 보호함 - 비교 2: 감우 센서는 일반적으로 축사 외부의 기상대에 있는 경우가 많으며 직접 비나 눈을 맞지 않도록 하는 위험경보에 사용되거나 직접 천창 또는 측창 구동기에 신호를 보내 열거나 닫도록 함
18	강제 환풍 장치 (forced air circulating system)	창이 없는 축사 등의 실내 환경을 최적의 상태로 유지하기 위하여 벽체나 지붕에 환풍기를 설치하여 인위적으로 내부 온도를 조절하고 환기시키는 설비
19	개방식 우사 (loose barn 또는 loose stall)	기둥과 지붕만 설치하고 벽면은 개방하여 소를 계류시키지 않고 우사 안에서 자유롭게 이동할 수 있도록 한 우사
20	개폐기 (switch)	전기 회로를 이었다 끊었다 하는 장치 - 비교 : 일반적으로 전등, 라디오, TV 등의 전기 기구를 손으로 누르거나 틀어서 작동하는 부분을 말함
21	건습구 온도계 (psychrometer)	공기의 습도를 재는 계기 - 비교: 건구 온도계와 습구 온도계를 조합시킨 것으로 한 쪽 온도계를 젖은 헝겊 등으로 싸두면 마른 쪽의 온도계보다 그 온도가 내려가게 되며 이 온도차를 이용하여 공기의 상대습도를 측정함
22	게이트웨이 (gateway)	구조가 다른 두 개의 통신 네트워크를 연결하는 장치 - 비교 : 근거리 통신망과 같은 하나의 네트워크를 다른 네트워크와 연결할 때 그 출입구 역할을 함
23	계류식 우사 (stanchion barn)	소가 방목장 등에 있을 경우 외에는 줄이나 체인 등으로 계류시킨 상태로 넓은 범위로 움직이지 못하도록 한 우사 - 비교: 개체 관리가 용이하여 각 개체별 사료급이나 착유작업 등이 가능하므로 주로 젖소 사육에 사용되고 있음
24	공기 여과 장치 (air filtration unit)	공기 속의 오염된 물질을 여과용 천과 폴리에스테일 스폰지 등과 같은 애프터 필터를 이용하여 걸러내는 장치 - 비교: 넓은 의미에서 공기 정화 장치의 하나로 볼 수 있음
25	공기 정화 장치 (air washing system)	공기 속의 먼지나 세균 따위를 걸러 내어 깨끗한 공기를 배출하는 장치
26	공기 조화 설비 (air conditioning equipment)	공기를 깨끗하게 정화하고 자동적으로 일정한 온도와 습도로 조절하여 실내에 공급하는 시설
27	구동기 (actuator)	동력을 이용하여 기계 등을 동작시키는 구동 장치 - 비교: 온도, 습도와 같은 축사 내 환경 상태를 조절하기 위하여 전기, 유공압 또는 기계적 신호 등에 따라 구동되는 장치
28	구동기노드 (actuator node)	구동기와 통신 모듈이 결합된 구조로서, 축사의 통합 제어기(시스템)의 제어 명령에 따라 구동기를 동작시키기 위한 장치
29	기계 환기 (mechanical ventilation)	'강제 환풍 장치' 참조
30	낙뢰보호기	벼락(낙뢰) 등으로 인해 유입되는 서지(surge)의 과전압 또는 과전류로부터 센

	(lighting protector)	서나 구동기, 제어장치 등의 기기를 보호하기 위한 장치 - 비교: 보호기에는 낙뢰 외에도 강전류전선과의 접촉에 의해 유입되는 과전압을 제한하고 과전류를 차단하는 기능 또는 보호기의 발화나 변형을 방지할 수 있는 기능 등을 가지고 있음
31	노드 (node)	데이터 통신망에서 데이터를 전송하는 통로에 접속되는 하나 이상의 기능 단위로서 주로 통신망의 분기점이나 단말기의 접속점을 말함 - 비교: 축사의 통합 제어기(시스템)와 디바이스(센서 또는 구동기) 간 통신을 가능하게 하는 매개체
32	농도 (concentration)	용액이나 기체의 진하고 묽은 정도 - 비교 : 일반적으로 %, mol/l, N 등의 단위가 사용되며 대기 오염, 수질 오염의 분석 관계에서는 ppm, mg/l 등이 사용됨
33	누전 (electric leakage)	절연이 불완전하거나 시설이 손상되어 전기의 일부가 전선 밖으로 새어 나와 주변의 도체에 흐르는 현상 - 비교: 전기 설비나 오래된 전선의 절연 불량, 전선 피복의 손상 또는 습기 등이 그 원인이 됨
34	누전 센서 (electric leakage sensor)	누전을 감지하는 장치
35	누전 차단기 (electric leakage breaker)	누전 등에 의한 전원의 불평형 전류가 일정한 값을 초과하면 전원을 자동적으로 차단하는 장치 - 비교: 전류 동작형과 전압 동작형이 있으며, 저압 배선의 콘센트 회로 또는 습기나 물기가 많은 장소에서의 배선 시 누전 차단기를 설치하는 것이 안전함
36	단상 (single phase)	하나의 위상을 갖는 교류로서, 하나의 교류 전압과 그에 대응하는 전류만이 있는 경우를 말함
37	단순 환경 제어 (simple environmental control)	축사를 관리하는데 있어 1~2가지 요인만 고려한 환경 제어 방식 - 비교: 주로 온도(적정 온도 유지), 시간 주기(일정 시간 작동) 등 비교적 제어가 쉬운 요인만을 제어하는 방식
38	데이터 변환기 (data convertor)	스마트 축사에 설치된 정보통신기술(ICT) 장치(SW 포함)로부터 수집된 데이터를 전송하기 위해 일정한 형식으로 데이터를 변환하는 하드웨어 또는 소프트웨어 - 비교: 서로 다른 형태의 데이터를 생성하는 기기종의 센서 또는 구동기로부터의 데이터를 효과적으로 처리하기 위해서는 동일한 형식을 갖는 데이터로 변환될 필요가 있음
39	데이터 수집기 (data collector)	스마트 축사에 설치된 정보통신기술(ICT) 장치(SW 포함)에서 생성되는 데이터를 수집하기 위한 하드웨어 또는 소프트웨어
40	디지털 소음계 (digital sound level meter)	'소음계' 참조
41	무선 인식 전자 이표 장치 (radio frequency identification, RFID)	'RFID' 참조
42	보온재 (heat insulating material, heat insulator)	열을 잘 옮기지 않고 보온력이 풍부하여 단열재로 쓰는 재료로서 보온 또는 보냉의 목적으로 사용됨 - 비교: 재질에 따라 유기질 보온재(저온용), 무기질 보온재(고온용)로 나눌 수 있으며 유기질에는 코르크, 톱밥, 폴리스틸렌, 연질섬유판, 펠트 등이 있고 무기질에는 석면, 암면, 글라스 울, 규조토, 염기성 탄산마그네슘 등이 있음
43	복합 환경 제어 (complex environmental control)	두 가지 이상의 환경 요소간 상호 영향을 고려하거나, 서로 다른 환경 요소를 변수로 환경 조건을 정하는 환경 제어 방식 - 비교: 가축 사육에 영향을 주는 환경 요인(온도, 암모니아, 광도, CO <sub>2</sub> , 습도 등)을 복합적으로 제어하여 가축에 적합한 환경을 구현하는 방법
44	사료 빈 (feed bin)	가축의 사료를 저장하는 저장소(사일로)
45	사물인터넷 (IoT, Internet Of Things)	정보 통신 기술(ICT, Information and Communication Technologies)을 기반으로 실세계(physical world)와 가상 세계(virtual world)의 다양한 사물들을 연결하여 보다 진보된 서비스를 제공하기 위한 기반 시설 - 비교 1: 인터넷을 기반으로 센서, 프로세서 등이 장착된 사물을 연결하여 사람과 사물, 사물과 사물 간 정보를 수집하고 제어, 관리할 수 있는 지능형 기술 및 서비스 - 비교 2: 사물 인터넷의 주요 기술로는 센싱 기술, 유무선 통신 및 네트워크 인프라 기술, 사물 인터넷 인터페이스 기술, 사물 인터넷을 통한 서비스 기술 등이 있음
46	사양 관리	가축의 생산성을 향상시키기 위하여 사료의 관리나 공급 등을 관리하는 절차

	(livestock breeding management)	
47	삼상 (three Phase)	세 개의 위상을 갖는 교류로서, 전압과 주파수는 같지만 위상이 서로 120°씩 다른 세 가지 전류나 전압을 한 조로 한 것 - 비교 : 교류를 사용하는 전원에는 3개의 전원용 단자(R상, S상, T상)로 구성되며, 일반적으로 발전소 등의 전기시설에서 사용됨
48	생산 경영 관리 시스템 (production management system)	축사의 생산과 관련된 정보 데이터베이스(DB)를 기반으로 효율적인 생산과 경영 관리를 목적으로 운영되는 시스템
49	생체 정보 수집 장치 (biometrics collection device)	가축의 특성과 관련된 정보를 수집하는 장치 - 비교: 귀, 목, 발목 등 가축 체외에 센서를 부착하는 형태와 구축 체내에 센서를 삽입하는 형태로 구분
50	센서 (sensor)	빛, 온도, 상대 습도 등의 물리적인 양을 계측하는 소자 또는 그 조사를 갖춘 장치 - 비교 : 환경 정보를 수집하는 장치로서 온도 센서, 습도 센서, CO <sub>2</sub> 센서, 풍향 센서, 감우 센서, EC 센서 또는 pH 센서 등을 말함
51	센서노드 (sensor node)	센서와 통신 모듈이 결합된 구조로서 측정값을 제어기 등에 전달하는 장치 - 비교 : 센서의 출력 신호를 전압 또는 전류를 변환하는 모듈과 제어장치와의 유무선 데이터 송수신 기능을 갖는 것으로 무선 센서 네트워크를 구성하는 기본 요소로 센싱 기능, 전산 기능, 양방향 유무선 통신 기능과 전원장치를 포함
52	소음 측정기 (sound measuring apparatus)	'소음계' 참조
53	소음계(sound level meter)	소음의 크기를 측정하는 기구나 장치 - 비교 1: 소리 주파수에 대한 감도의 차이를 사람의 청각과 같게 되도록 맞은 값으로 나타내며, 단위로는 phone 또는 dB를 사용함 - 비교 2: 예비적인 방법으로 소음의 크기를 대략적인 값으로 나타내는 간이 소음계, 충격음의 크기를 측정하기 위한 임펄스 소음계, 소음의 크기를 현장에서 수치화(순시값 또는 대표값)하여 바로 확인할 수 있는 디지털 소음계 등이 있음
54	수전 설비 (passive equipment)	타인의 전기설비 등으로부터 축사 내 배전설비로 전기를 공급받기 위한 설비 - 비교: 전력회사로부터 고압의 전기를 공급받아 변환한 후 시설 내부로 공급하기 위한 것으로 제어개폐기, 보호계전기가 설치된 배전반, 차단기 등의 기기 및 전선로의 총칭
55	스마트 축사 (smart livestock barn)	가축의 사육이나 생산량을 극대화하기 위하여 환경 요소(빛, 온도, 습도 등)를 최적으로 조절할 수 있으며, 이를 원격화·자동화한 축사를 말함 - 비교: 어디서나 가축의 사육 환경을 점검하고 이를 적정하게 유지, 관리하기 위하여 센서노드나 제어 장치, 구동기 노드 등의 정보통신기술(ICT)이 많이 활용되고 있음
56	스마트 축산 (smart livestock industry)	정보통신기술(ICT)을 축산물의 생산, 사육, 가공, 유통과 소비 전반에 접속하여 사육 환경을 관리하고 생산 효율성을 높일 수 있는 산업
57	스마트팜 (smart farm)	스마트 온실의 원격화·자동화·최적환경 제어를 과수원, 축사 등으로 확장한 개념으로 농업 전반에서 개발되고 있음. - 비교 : 농업의 생산뿐만 아니라 수확 후 가공·유통 단계까지 확장되고 있음
58	습도 센서 (humidity sensor)	습도를 측정하는 장치 - 비교: 습도는 공기 중에 수증기가 포함된 정도를 말하는 것으로 가축의 생육에 적정 범위가 존재하며, 병충해, 가축의 스트레스에 직접적인 영향을 미침
59	아크 (arc)	기체 방전의 하나로, 양극과 음극에 고압의 전위차를 가할 경우 발생하는 밝은 전기 불꽃 또는 방전(아크 방전) - 비교: 전류 밀도가 크고 방전에 지속성이 있으며, 이는 전기화재의 원인이 되기도 함
60	아크 센서 (arc sensor)	아크 방전을 감지하는 장치
61	암모니아 센서 (ammonia gas sensor)	공기 중의 암모니아를 검지하거나 그 농도를 측정하는 장치
62	액추에이터 (actuator)	'구동기' 참조
63	온도 센서 (temperature sensor)	온도(°C)를 측정하는 장치 - 비교: 축사 등의 내부 온도는 가축의 사육 등과 관련되므로 가축 사육에 중요한 요소임
64	와이파이 (WiFi, Wireless Fidelity)	IEEE 802.11 기반의 무선랜(Wireless LAN) 표준기술 - 비교: 무선 접속 장치(AP, Access Point)가 설치된 곳에서 전파나 적외선 전

		송 방식을 이용하여 일정 거리 안에서 무선 인터넷을 할 수 있는 근거리 통신망 기술로서 축사 외부에서 원격 제어를 통해 축사를 관리할 때 사용함
65	유동 팬 (flow fan 또는 circulation fan)	축사 내부의 공기 순환을 통해 온도 및 습도를 조절하는 장치
66	유량 (flow rate)	유체(流體)가 단위 시간 동안 흐르는 양 - 비교 : 유체가 흐를 때 일정 면적의 단면을 통과하는 유체의 체적, 질량 또는 중량을 시간에 대한 비율로 표시한 것
67	유량 센서 (flow rate sensor)	하천, 수로 또는 관 등에 흐르는 유량(G(kgf/s) 또는 Q(m <sup>3</sup> /s))을 측정하는 센서 - 비교 : 유체(流體)의 유속은 기벽에서의 거리에 따라 상이하므로 일정 지점의 통과 전후에서 압력이나 수위의 변화를 측정하여 계산함
68	일사량 (amount of solar radiation 또는 amount of insolation)	태양의 복사 에너지가 지표에 닿는 양으로 위도에 따라 다름 - 비교 1: 태양 광선에 직각으로 놓인 단위 면적에 단위 시간 동안 복사되는 에너지의 양(W/m <sup>2</sup> )으로 축사 내 온도 제어, 관수 제어 등에 활용됨 - 비교 2: 특정 기간 동안 누적된 총 일사량을 누적 일사량이라 함
69	일사 센서 (pyranometer)	단위 시간에 대하여 단위 면적당 입사되는 일사 에너지(W/m <sup>2</sup> ·h)를 측정하는 장치 - 비교 1: 일사 센서에는 직달 일사 센서, 전천 일사 센서 등이 있으며 측정 위치에 따라 내부 일사와 외부 일사로 구분됨 - 비교 2: 축사 내 온도 제어, 관수 제어 등에 활용됨
70	일조 센서 (sunshine sensor)	일조 상태 및 시간이나 태양 복사량(solar radiation)을 측정하는 장치
71	임펄스 소음계 (impulse sound level meter)	'소음계' 참조
72	자동 급이기 (automatic feeder)	사료를 자동으로 가축에게 전달하는 장치 - 비교: 일반적으로 축사 외부에 설치된 사료 bin에서 스크류 컨베이어를 타고 축사 내부로 운반된 후 벨트 컨베이어, 스크류 컨베이어, 체인 컨베이어 또는 이동식 호퍼 피더를 통하여 자동으로 급여됨
73	적외선 센서 (infrared ray sensor, Ir sensor)	적외선을 이용해 온도, 압력, 방사선의 세기 등의 물리량이나 화학량을 감지하여 신호처리가 가능한 전기량으로 변환시키는 장치 - 비교 1: 스스로 적외선을 방사하여 빛의 차단에 따른 변화를 감지하는 능동식과 외부로부터의 적외선의 변화만을 감지하는 수동식이 있음 - 비교 2: 온도 센서나 자외선 센서보다 감도나 정확도가 높아 방법이나 화재 감시 등에 널리 사용되고 있으며, 의료용 서모그래피, 동식물의 생태 관찰 등에 응용되고 있음
74	정전 센서 (power failure sensor)	전기 공급이 끊긴 상황을 감지하는 장치
75	조도 센서 (illuminance sensor)	단위 시간에 대하여 단위 면적당 광속밀도(dp/dS)를 측정하는 장치. 단위는 룩스(lux, lx) - 비교 : 포토레지스터(photoresistor) 또는 CdS(Cadmiumsulfide)로도 불리는 조도 센서는 빛의 밝기에 따라 저항값이 변하는 특징을 가짐
76	차압 센서 (differential pressure sensor)	압력의 차이를 측정하는 장치 - 비교: 축사 내부와 외부에 설치된 차압 센서를 통해 공기의 흐름을 측정함
77	축사 통합 관리 시스템 (integrated livestock barn management system)	돈사, 우사 또는 계사 등과 같은 축사 관리(관제)에 필요한 소프트웨어를 개별 축사 운영 시스템에 설치하고 외부의 데이터 서버와 연동하여 복수의 축사 운영 시스템으로 가축의 사육 정보 등을 피드백하는 시스템 - 비교1: 센서 노드 및 제어 노드에 따라 필요한 소프트웨어를 축사 운영 시스템에 설치해 주는 역할을 하며, 관제 기능을 강조하는 경우에는 '축사 통합 관제 시스템(integrated livestock barn management and control system)'으로 부르기도 함 - 비교2: 적용되는 축종에 따라 돈사 통합 관리 시스템, 우사 통합 관리 시스템 또는 계사 통합 관리 시스템 등으로 부름
78	축사 통합 제어 시스템 (integrated livestock barn control system 또는 Integrated livestock barn gateway)	축사 운영 시스템으로부터의 명령을 제어 노드에 전달하여 축사의 각종 구동 장치를 제어하거나 센서 노드로부터의 센싱값을 축사 운영 시스템으로 전달하는 장치 - 비교 1: 프로토콜을 변환하는 게이트웨이 기능을 갖거나 축사 운영 시스템과 통합하여 운용할 수 있으며, 관리 기능을 포함하여 관제의 기능을 강조하는 경우에는 '축사 통합 관제 시스템(integrated livestock barn management and control system)'으로 부르기도 함 - 비교 2: 가축 사육 환경에 대한 정보 수집과 모니터링을 통해 자료를 분석하여 최적의 생산성을 이룰 수 있도록 각종 장비를 제어하거나 개체 정보를 기록·관리함

		- 비고 3: 적용되는 축종에 따라 돈사 통합 제어 시스템, 우사 통합 제어 시스템 또는 계사 통합 제어 시스템 등으로 부름
79	풍속 센서 (wind speed sensor)	바람의 세기(m/s)를 측정하는 장치 - 비고: 축사의 구조물 안전 관리, 환기량 예측, 온도 및 습도 관리를 위한 환기량(천창, 측창) 조절 등의 목적에 사용됨
80	풍향 센서 (wind-direction sensor)	바람의 방향(°)을 측정하는 장치 - 비고: 축사의 온도, 환기량 관리 등의 목적으로 사용됨
81	협대역 사물 인터넷 (NB IoT, NarrowBand-Internet Of Things)	데이터 통신에서 협대역을 이용하여 전력 소비가 적은 광역 통신을 지원하는 사물 인터넷 표준 기술('사물 인터넷' 비교) - 비고: 저전력 광역(LPWA, Low Power Wide Area) 통신을 지원하는 협대역 사물 인터넷 표준으로, GSM 또는 LTE 망에서 협대역을 이용하여 수 백 kbps 이하의 데이터 전송 속도와 10km 이상의 광역 서비스를 지원하며, 수도 검침, 위치 추적용 기기 등의 목적에 사용됨
82	환경 정보 (environmental information)	체계적인 분류 기준과 코드화 작업을 통해 업무에 효율적으로 이용할 수 있도록 한, 자연 환경이나 생활환경 등에 대한 자료 - 비고: 축사 내부 또는 외부의 센서로부터 수집된 센싱값 정보와 구동기로부터 수집된 값 등 축사 환경과 관련된 측정 정보
83	환경 제어 (environmental control)	지상부 환경(광, 온도, 습도, CO <sub>2</sub> 가스 등) 및 지하부 환경 요인(이·화학성, 양분 조성 등)을 제어하여 가축 사육에 적합한 환경을 구현하는 것 - 비고: 환경 제어를 위해 고려하는 환경 요소의 수와 상호 영향에 따라 단순 환경 제어와 복합 환경 제어로 구분
84	환기구 (ventilating opening)	탁한 공기를 맑은 공기로 바꾸거나 온도 조절 등을 위하여 공기의 출입이 가능하도록 만든 입배기구 - 비고: 축사 등에서 내부 공기 중의 산소량을 유지해주고 연기나 유해 가스 등을 희박하게 하기 위해 신선한 공기를 공급해 주는 역할을 함
85	환풍기 (ventilator)	실내의 공기를 외부의 공기와 바꾸어 내부 공기를 환기시키기 위한 기구. 환기팬 - 비고: 축사에 설치하여 외부의 바람을 받아들이고 내부의 공기를 외부로 배출하여 축사 내 온도나 습도를 조절하는 장치로 일반적으로 프로펠러 모양의 팬이 달려 있음

나-3) 스마트팜 용어정의-제 3부 : 노지 및 과수원(1종) ('21 제안, '22 제정)

원예, 축산, 노지 등의 농업 기술과 ICT(Information and Communication Technology)를 접목한 스마트팜은 원격에서 자동으로 작물이나 가축 등의 최적 생육 환경을 유지하고 관리할 수 있게 한다. 이를 통해 사용자에게 편의성을 제공할 수 있을 뿐만 아니라 AI, 빅데이터(big data) 기술을 활용한 정밀 농업을 가능하게 하고, 나아가 생산량과 소비량 등의 정확한 예측을 통하여 농가의 수익을 향상시킬 수 있는 기술이다. 이와 같이 농업과 ICT 분야가 융합된 스마트팜 기술이 도입되면서, 각 분야에서 사용하는 동일한 용어에 대한 이해의 차이로 인한 불필요한 혼란이 가중되고 있다. 이는 스마트팜 기술과 서비스의 확산과 보급에 걸림돌로서 작용한다.

이 표준은 정부와 산업계, 연구계 등에서 사용하는 스마트팜 관련 용어, 특히 스마트 노지와 과수 분야에 대한 다양한 용어를 체계적으로 정의한다. 이를 통해 농장주와 농장의 종사자는 물론 스마트팜 관련 장비 제조업자 및 시설업자, 서비스 제공업자의 이해를 증진시키는 것을 목적으로 한다.

표 1 - 스마트팜(축산분야) 용어 정의 현황

순번	용어	정의
1	EC 센서 (Electrical Conductivity Sensor)	수용액 중의 전기 전도도(dS/m)를 측정하는 장치
2	EC 컨트롤러(Electrical Conductivity controller)	수용액의 전기 전도도를 제어해 주는 장치
3	FDR 센서(Frequency Domain Reflectometry sensor)	토양의 수분 함량(%)을 측정하는 장치 - 비교 1: 전자기파를 이용하여 토양의 정전용량을 평가하고 이를 통해 토양의 수분함량을 산정함 - 비교 2: 토양의 수분함량을 측정하는 다른 방식으로 TDR 센서 방식이 있음
4	pH 센서(hydrogen exponent sensor)	수용액 중의 수소 이온 농도(pH)를 측정하는 장치
5	pH 컨트롤러(hydrogen exponent controller)	수용액 중의 수소 이온 농도를 제어해 주는 장치
6	RS232 통신 (recommended standard-232 communication)	데이터 단말 장치(DTE)와 모뎀 또는 데이터 회선 종단 장치(DCE)를 상호 접속하기 위해 EIA에서 개발한 표준 규격 - 비교 : 직렬 통신 방식 중 하나로 일반적으로 10m 정도의 범위에서 정상적인 데이터 통신을 할 수 있음
7	RS485 통신 (recommended standard-485 communication)	RS232, RS422 표준 규격의 확장 버전으로 EIA/TIA에서 개발한 다 대 다(N:N) 양방향 직렬 통신 표준 규격 - 비교 : 2가닥의 통신선으로 다수의 장치를 상호연결하는 직렬 통신 방식의 하나로 최대 1.2km 범위까지 신호전송이 가능하며, 주로 센서노드와 통합제어장치 간의 송수신방식으로 사용함
8	TDR 센서(time domain reflectometry sensor)	토양의 수분 함량(%)을 측정하는 장치 - 비교 1 : 토양수분의 유전율을 이용하는 방식으로 전자파가 반사되는 시간과 속도를 통해 수분함량을 산정함 - 비교 2 : 토양의 수분함량을 측정하는 다른 방식으로 FDR 센서 방식이 있음
9	가시광선(visible light)	일반적으로 380 nm에서 780 nm 사이의 파장을 갖는 광선으로 사람의 눈으로 볼 수 있는 빛 - 비교 : 파장의 길이에 따라 무지개색으로 분류하는데 적색(780nm)보다 긴 파장을 적외선, 보라색(380nm)보다 짧은 파장을 자외선으로 부름
10	감우 센서(rain detector)	비가 오는지를 감지하는 장치
11	강설량(snowfall)	일정한 기간 동안 일정한 장소에 내린 눈의 양
12	강수량(precipitation)	비, 눈, 우박, 서리, 이슬 등으로 일정 기간 동안 일정한 장소에 내린 물의 총량
13	강우량(rainfall)	비고:강설량과강우량을합하여표시한양 일정한 기간 동안 일정한 장소에 내린 비의 양
14	개폐기(switch)	전기 회로를 이었다 끊었다 하는 장치 - 비교: 일반적으로 전등, 라디오, TV 등의 전기 기구를 손으로 누르거나 틀어서 작동하는 부분을 말함
15	건습구 온도계(psychrometer)	공기의 습도를 재는 계기 - 비교: 건구 온도계와 습구 온도계를 조합시킨 것으로 한 쪽 온도계를 젖은 헝겊 등으로 싸두면 다른 쪽의 온도계보다 그 온도가 내려가게 되며 이 온도

		차를 이용하여 공기의 상대습도를 측정함 지면으로부터 이삭목까지의 길이
16	경장(stem length)	- 비교: 지면으로부터 최하단 원가지가 발생한 지점까지의 길이로서 '줄기 길이'로 순화
17	관개(irrigation)	'관수' 참조
18	관비(fertigation)	비료에 물을 섞어 관수를 겸하여 공급하는 것
19	관비 공급 장치(fertigation system)	물과 비료액을 적당히 혼합하여 희석시킨 액체 상태의 비료를 작물에 공급하는 장치 - 비교 1: 관수를 겸한 비료 공급이 가능하도록 보통 관수 장치에 액비 혼합 장치를 부착하여 사용함 - 비교 2: '양액 시스템' 비교
20	관수(irrigation)	작물 생육에 필요한 물을 주는 것 - 비교 1: 시설 관수는 대부분 강우가 아닌 외부에서 공급된 물을 이용하므로 계획적인 관수가 가능하며, 유형으로는 고랑 관수, 지표 관수, 미스트 관수, 점적 관수, 지중 관수, 저면 관수 등이 있음 - 비교 2: 작물의 종류와 생육 단계, 재배 시기 등을 고려하여 관수 시기를 결정해야 하며, 토양 수분의 상태에 따라 관수 시기를 결정하는 경우 대개 포장 용수량과 위조점 사이에서 관수를 해야 함
21	관수 밸브(irrigation valve)	관수량을 조절해주는 장치 - 비교 1: 수동식 밸브와 전자식 밸브가 있으며, 전자식 밸브 중 하나인 솔레노이드 밸브를 주로 사용함 - 비교 2: '솔레노이드 밸브' 비교
22	관수 펌프(irrigation pump)	노지에 물 등을 공급하기 위한 펌프
23	관정(tube well)	자유면 지하수 또는 피압면 지하수를 퍼 올리기 위하여 땅속을 굴삭하여 설치한 관 모양의 우물 - 비교 1: 피압면 지하수(confined an artesian water)는 암반 또는 불투수층으로 둘러싸인 지하수로서, 흔히 이 지하의 하부는 상부의 물에 의하여 가압되어 있음 - 비교 2: 자유면 지하수(free ground water)는 대기 압력과 같은 내부 압력을 가진 지하수
24	관제(monitring and control)	작물 생장에 필요한 노지의 환경을 모니터링하고 제어하는 것
25	광보상점(light compensation point)	작물에 의한 이산화탄소의 흡수량과 방출량이 같아져서 작물이 외부 공기 중에서 실질적으로 흡수하는 이산화탄소의 양이 0이 되는 광의 강도 - 비교: 작물은 각각의 광보상점 값 이상의 광량을 제공해 주어야 광합성이 가능하여 성장하기 시작하며, 그 미만의 광량일 경우 대부분 줄기만 성장하고 잎이 자리지 않는 웃자람이나 생장부진, 꽃 수가 적어지고 꽃과 과일이 떨어지는 현상 등이 발생함
26	광양자 센서(light quantum sensor)	단위 면적에 대하여 단위 시간 동안 입사되는 광양자의 수(광양자속밀도, $\mu\text{mol}/\text{m}^2/\text{s}$ )를 측정하는 센서 - 비교: 400~700 nm 사이의 광양자속밀도를 '광합성 유효 광양자속밀도'(PAR)라고 함
27	광포화점(light saturation point)	작물의 광합성 과정에서 더 강하게 빛을 비추어도 광합성량이 증가하지 않을 때의 빛의 세기 - 비교: 광도가 증가함에 따라 광합성 속도가 증가하다가 광포화점 이상에서는 광합성량이 더 이상 증가하지 않음
28	광합성(photosynthesis)	식물이 빛 에너지와 이산화탄소, 물을 이용하여 유기물을 합성하는 과정 - 비교: 광합성은 유효파장의 빛이 셀수록, 이산화탄소가 많을수록 증가하지만, 일정 수준 이상으로 올라가면 더 이상 증가하지 않으며, 작물 생육에 피해를 줄 수 있음
29	광합성 유효 방사(photosynthetically active radiation)	작물의 광합성에 유효한 파장 범위(약 400 nm~700 nm)에 들어있는 빛 - 비교 1: 작물의 생육에 영향을 주는 광선 영역으로 광합성의 에너지원으로 작용 - 비교 2: 광합성 유효 광의 측정은 이 파장 범위의 광양자 플럭스에 의해서 표현되는 것이 일반적이며 태양 방사의 경우 일사 센서로 측정되는 총 에너지 플럭스 중 광합성 유효 광이 차지하는 비율은 약 45%임
30	광합성 측정	온도, 빛, 이산화탄소, 수분의 변화량을 확인하여 작물의 광합성량을 측정하는

	장치(photosynthesis measurement unit)	장치
31	구동기(actuator)	동력을 이용하여 기계 등을 동작시키는 구동 장치 - 비교: 온도, 습도와 같은 환경 상태를 조절하기 위하여 전기, 유공압 또는 기계적 신호 등에 따라 구동되는 장치
32	구동기 노드(actuator node)	구동기와 통신 모듈이 결합된 구조로서, 노지의 통합 제어기(시스템)의 제어 명령에 따라 구동기를 동작시키기 위한 장치
33	급수(water supply)	펌프 등을 통해 물을 대어 주는 것 또는 그 물
34	급액(nutrient feed)	작물에 양액을 공급해 주는 일 - 비교: 작물의 생육에 필요한 무기성분을 함유한 물을 양액이라고 함
35	급액 제어(nutrient control)	작물의 생장이 최적의 상태를 유지할 수 있도록 급액을 관리하는 것 - 비교: 순환식과 비순환식으로 구분할 수 있으며, 제어 방식과 관계 없이 작물 생장을 최적의 상태로 유지하면서 배양액의 손실을 최소한으로 하거나 품질을 높이기 위한 것
36	기계 장치(mechanical equipment)	기계를 설비하는 일 또는 그런 장치 - 비교: 동력에 의하여 작업을 행하는 발전기, 전동기, 공작 기계, 작업 기계 등
37	기상대(meteorological sensor)	온도, 습도, 풍향, 풍속 및 기압 등의 외부 기상 현상을 측정하기 위한 장치 - 비교: 기상 현상의 관측, 조사, 연구를 위한 기관을 말하기도 하며, 관상대 또는 측후소라고 부르기도 함
38	기온 센서(temperature sensor)	공기의 덩고 찬 정도를 측정하는 센서 - 비교 : '온도 센서' 비교
39	기후 요소(climatic element)	기후를 구성하는 요소로서 기온, 강수량, 풍향, 풍속, 습도, 구름량, 증발량, 일사량, 일조 시간, 기압 등을 말함
40	낙뢰보호기(lightning protector)	벼락(낙뢰) 등으로 인해 유입되는 서지(surge)의 과전압 또는 과전류로부터 센서나 구동기, 제어장치 등의 기기를 보호하기 위한 장치 - 비교: 보호기에는 낙뢰 외에도 강전류 전선과의 접촉에 의해 유입되는 과전압을 제한하고 과전류를 차단하는 기능 또는 보호기의 발화나 변형을 방지할 수 있는 기능 등을 가지고 있음
41	노드(node)	데이터 통신망에서 데이터를 전송하는 통로에 접속되는 하나 이상의 기능 단위로서 주로 통신망의 분기점이나 단말기의 접속점을 말함 - 비교: 노지의 통합 제어기(시스템)와 디바이스(센서 또는 구동기) 간 통신을 가능하게 하는 매개체
42	노지(open field 또는 open land)	지붕 등으로 덮거나 가리지 않은 땅 - 비교: 온실 등의 시설이 아닌 바깥의 자연조건을 말하기도 함
43	노지 스마트팜(open field smart farm)	노지 경작의 생산성 향상과 이상 기후 및 병충해로부터 피해를 방지하기 위하여 재배 환경을 인위적으로 관리할 수 있는 노지 재배 시설 - 비교 : '스마트 노지' 비교
44	노지 운영 시스템(open field operating system)	노지에 설치된 센서로부터 환경과 생육 정보를 모니터링하고, 노지 환경 제어 알고리즘 운영 및 제어, 수집된 데이터를 데이터베이스에 기록하는 장치 - 비교: 데이터 베이스에 저장된 데이터를 바탕으로 영농 일지를 작성할 수 있으며, 노지 운영에 필요한 서비스 및 제어 소프트웨어를 노지 통합 관리 시스템으로부터 다운로드하여 사용할 수 있음. 또한 센서 노드 및 제어 노드의 동작 주기와 장애 등을 관리함
45	노지 통합 관리 시스템(integrated open field management system)	노지 관리(관제)에 필요한 소프트웨어를 개별 노지 운영 시스템에 설치하고 외부의 데이터 서버와 연동하여 복수의 노지 운영 시스템으로 작물의 생육 정보 등을 피드백하는 시스템 - 비교: 센서 노드 및 제어 노드에 따라 필요한 소프트웨어를 노지 운영 시스템에 설치해 주는 역할을 하며, 관제 기능을 강조하는 경우에는 '노지 통합 관제 시스템(integrated open field management and control system)'으로 부르기도 함
46	노지 통합 관제 시스템(integrated open field monitoring and control system)	'노지 통합 관리 시스템' 참조
47	노지 통합 제어기(integrated open	노지 운영 시스템으로부터의 명령을 제어 노드에 전달하여 노지의 각종 구동 장치를 제어하거나 센서 노드로부터의 센싱값을 노지 운영 시스템으로 전달하

	field controller 또는 integrated open field gateway)	는 제어 장치 - 비교: 프로토콜을 변환하는 게이트웨이 기능을 갖거나 노지 운영 시스템과 통합하여 운용할 수 있음
48	농도(concentration)	용액이나 기체의 진하고 묽은 정도 - 비교: 일반적으로 %, mol/l, N 등의 단위가 사용되며 대기 오염, 수질 오염의 분석 관계에서는 ppm, mg/l 등이 사용됨
49	단상(single phase)	하나의 위상을 갖는 교류로서, 하나의 교류 전압과 그에 대응하는 전류만이 있는 경우를 말함
50	데이터 변환기(data convertor)	스마트 노지에 설치된 정보통신기술(ICT) 장치(SW 포함)로부터 수집된 데이터를 전송하기 위해 일정한 형식으로 데이터를 변환하는 하드웨어 또는 소프트웨어 - 비교: 서로 다른 형태의 데이터를 생성하는 기기종의 센서 또는 구동기로부터의 데이터를 효과적으로 처리하기 위해서는 동일한 형식을 갖는 데이터로 변환될 필요가 있음
51	데이터 수집기(data collector)	스마트 노지에 설치된 정보통신기술(ICT) 장치(SW 포함)에서 생성되는 데이터를 수집하기 위한 하드웨어 또는 소프트웨어
52	로라(LoRa, Long Range)	사물 상호간 통신(IoT)을 위한 저전력의 장거리 통신(LPWA) 기술
53	모니터링(monitoring)	작물의 생육 상태나 환경 상태의 변화를 기록하기 위해 과학적으로 계획된 연속적 측정과 관측
54	모세 관수(capillary water)	모세관 현상으로 물의 표면장력에 의해 토양의 공극에 일시적으로 보유되어 있는 물 - 비교: '저면 관수' 참조
55	미기상(micrometeorology)	지면에 접한 대기층의 미세한 규모의 기후
56	미스트 관수(mist irrigation)	호스나 파이프관에 높은 수압이 유지되는 상태에서 미세한 분출구를 통과한 수분이 안개 상태로 사방에 분산되어 관수하는 방법 - 비교: 파종상, 육모상이나 번식상 등의 관리에 적합함
57	발아율(germination rate)	일정한 양의 씨앗 가운데 발아할 수 있는 씨앗의 비율, 또는 뿌린 씨앗에 대하여 발아한 씨앗의 비율
58	배수(drainage)	과습 상태의 노지의 물이나 불필요한 물을 자연적 또는 인공적으로 다른 곳으로 빼주어 작물 생육에 적합한 조건으로 만드는 일 - 비교: 일반적으로 자연 배수와 시설 배수로 구분하는데 자연 배수는 지형에 따라 자연적으로 이루어지는 것을 말하고, 시설 배수는 지상과 지하에 배수 시설을 설치하는 이용함
59	배액(nutrient drainage)	급액 후 작물에 흡수되지 못하고 배출되는 양액 - 비교: 배액률(%) = (배액량/급액량) x 100
60	배지(culture medium)	식물이나 세균, 배양 세포 등을 기르는데 필요한 영양소가 들어 있는 액체나 고체상태의 것 - 비교 1: 식물의 뿌리를 고정 또는 지지하기 위해 사용되며 온실에서는 펄라이트, 암면, 코코섬유, 왕겨 등이 주로 이용됨 - 비교 2: 생물은 생존·발육에 불가결한 물을 비롯하여 영양물질로서 다량요소·미량요소 등을 요구한다. 그 중 기체상으로 얻어지는 것을 제외하고는 모두 무기 또는 유기화합물로서 배지에 공급해 주어야 한다. 필요한 화합물은 독립 영양·중속영양 등 영양 형식에 따라 여러 가지이다. 보통 영양원을 탄소원·질소원·무기염류·발육인자(비타민류) 등으로 나누어서 생각한다. 특히 발육인자와 관련하여 생물체에서 추출한 비교적 복잡한 조성을 가진 것을 주체로 한 경우를 천연배지(天然培地)라고 함
61	백립중(one hundred seed weight)	콩이나 팥 등의 씨알의 크기나 중량을 알고자 할 때 백 개의 낱알을 단 총무게
62	백엽상(instrument shelter(screen))	기상 관측용 설비가 태양이나 대지의 복사 또는 비나 눈의 직접적인 영향을 받지 않도록 설치된 지붕이 달린 작은 나무상자 - 비교: 미기상 제어시스템의 일부로서 내부에는 주로 온도 센서나 습도 센서를 설치하고 측정 감도와 범위를 높이기 위해 통풍이 잘 될 수 있도록 함
63	보수력(capacity to retain water)	토양 흡인력에 따른 토양 수분 함량을 중량 백분율 또는 용량 백분율 등으로 표시한 것 - 비교: 일반적으로 토양은 흡착력 또는 모세관 장력 등의 흡인력으로 수분을 보관하며, 보수력은 수분 장력에 따라 그 값이 달라질 수 있음
64	보온재(heat insulating)	열을 잘 옮기지 않고 보온력이 풍부하여 단열재로 쓰는 재료로서 보온 또는 보

	material, heat insulator)	냉의 목적으로 사용됨 - 비교: 재질에 따라 유기질 보온재(저온용), 무기질 보온재(고온용)로 나눌 수 있으며 유기질에는 코르크, 톱밥, 폴리스틸렌, 연질섬유판, 펠트 등이 있고 무기질에는 석면, 암면, 글라스 울, 규조토, 염기성 탄산마그네슘 등이 있음
65	복사(radiation)	물체로부터 열이나 전자기파가 사방으로 방출되는 현상 - 비교 1: 단위 시간 당 방출 또는 입사되는 양을 복사량(amount of radiation)이라 함 - 비교 2: 방사와 같은 의미로 사용됨
66	복사 에너지(radiation energy)	복사되는 열이나 전자기파의 에너지 - 비교: 태양이나 전등의 빛, 난로의 열 등과 같이 복사되는 에너지는 검은 색의 물체에 잘 흡수되며 표면이 흰색 이거나 광택이 나는 물체에서는 반사되는 특성이 있음
67	비정형 데이터(unstructured data)	형태와 구조가 복잡하여 정형화되지 않은 데이터 - 비교 1: 숫자 등과 달리 동영상 파일, 오디오 파일, 사진, 보고서, 메일 본문 등과 같이 정의된 데이터 구조가 없어 내용에 대한 질의 처리(query processing)가 어렵기 때문에 데이터의 특징을 추출하여 반정형 또는 정형 데이터로 변환하는 전처리(pre-processing)가 필요함 비교 2: 비구조적 데이터 또는 다크 데이터(dark data)와 유사한 의미를 가짐 비교 3: '정형 데이터'와 비교
68	사물인터넷(IoT, Internet Of Things)	정보 통신 기술(ICT, Information and Communication Technologies)을 기반으로 실세계(physical world)와 가상 세계(virtual world)의 다양한 사물들을 연결하여 보다 진보된 서비스를 제공하기 위한 기반 시설 - 비교 1: 인터넷을 기반으로 센서, 프로세서 등이 장착된 사물을 연결하여 사람과 사물, 사물과 사물 간 정보를 수집하고 제어, 관리할 수 있는 지능형 기술 및 서비스 - 비교 2: 사물인터넷의 주요 기술로는 센싱 기술, 유무선 통신 및 네트워크 인프라 기술, 사물인터넷 인터페이스 기술, 사물인터넷을 통한 서비스 기술 등이 있음
69	산가(acid value)	산성 화합물을 중화하는 데 필요한 염기의 질량 - 비교 1: 용액 속에 해리된 탄산, 유기산 등의 산 성분의 중화에 이용하는 알칼리 성분을 이에 상당하는 수산화칼륨(KOH) 양으로 나타낸 값으로 ppm으로 나타내며, 1ppm을 1도로 함 - 비교 2: 산도라고도 하며 '수소 이온 농도' 비교
70	삼상(three phase)	세 개의 위상을 갖는 교류로서, 전압과 주파수는 같지만 위상이 서로 120°씩 다른 세 가지 전류나 전압을 한 조로 한 것 - 비교: 교류를 사용하는 전원에는 3개의 전원용 단자(R상, S상, T상)로 구성되며, 일반적으로 발전소 등의 전기시설에서 사용됨
71	상대 성장률(relative growth rate, RGR)	일정 기간 동안의 식물의 성장량을 증량으로 나눈 값 - 비교: $RGR = \frac{1}{\text{개체의 건조중량}} \times \frac{\text{건조중량의 증가량}}{\text{단위시간}}$
72	상대 습도(relative humidity)	일정 부피의 공기 속에 실제로 포함되어 있는 수증기의 양과 최대로 포함할 수 있는 수증기 양(포화 수증기 양)과의 비율(%)로서 일반적으로 습도라고 함
73	생리 장애(physiological stress)	작물이 생명을 유지하는데 있어서의 여러 가지 현상이나 기능에 이상이 발생하는 것 - 비교: '생리 장애' 비교
74	생리 장애(physiological disorder)	공기의 조성이나 온도, 영양 등 외부의 어떤 물리적, 이화학적인 환경이나 스트레스에 의하여 작물에 나타나는 비정상적 생리현상 - 비교 1: 일반적으로 영양 불균형 등에 의한 영양 생리 장애와 같이 내부 요인에 의한 것을 장애, 기상 악화나 독성 가스 등에 의한 재해 장애 등 외부 요인에 의한 것을 장애로 대별할 수 있으나 복합적인 요인에 의해 발생하게 되므로 장애와 장애를 구분하는 것이 어려움 - 비교 2: 작물의 대사과정에 연관되는 기온, 광, 습도 등의 지상부 환경과 산도, 수분, 영양, 지온 등의 지하부 환경, 경작에 이용되는 각종 자재 등이 단독 또는 복합적으로 작용하여 발생 - 비교 3: '생리 장애' 비교
75	생산 경영 관리 시스템(production management system)	노지의 생산과 관련된 정보 데이터베이스(DB)를 기반으로 효율적인 생산과 경영 관리를 목적으로 운영되는 시스템

76	생육(growth)	<p>생물이 나서 길러지는 것</p> <ul style="list-style-type: none"> <li>- 비교 1: 생물이 발아하여 성숙할 때까지 달하는 과정으로 광합성 작용, 호흡 작용, 세포의 분열과 성장, 증산 작용 등을 통해 크기와 무게가 증가함</li> <li>- 비교 2: 생장이라고도 함</li> </ul>
77	생육 모델(growth model)	<p>작물의 연령이나 시간적인 경과에 따른 생장 인자의 변화를 수리적 형식으로 표현한 것</p> <ul style="list-style-type: none"> <li>- 비교 1: 작물의 생명 주기 내에서 최적의 환경 조건을 제공하기 위한 모델</li> <li>- 비교 2: 생장 모델이라고도 함</li> </ul>
78	생육 정보(growth data)	<p>생육 환경의 분석 및 최적 생육 환경 조성을 위해 주기적이고 지속적으로 수집되는 정보</p> <ul style="list-style-type: none"> <li>- 비교: 주로 초장(작물의 키), 줄기 직경, 엽장, 엽폭, 착과수, 엽면적 등이 있음</li> </ul>
79	생체중(fresh weight)	작물의 성장에 따른 전체 작물의 무게(g)
80	센서(sensor)	<p>빛, 온도, 상대 습도 등의 물리적인 양을 측정하는 소자 또는 그 소자를 갖춘 장치</p> <ul style="list-style-type: none"> <li>- 비교: 환경 정보를 수집하는 장치로서 온도 센서, 습도 센서, CO<sub>2</sub> 센서, 풍향 센서, 감우 센서, EC 센서 또는 pH 센서 등을 말함</li> </ul>
81	센서 노드(sensor node)	<p>센서와 통신 모듈이 결합된 구조로서 측정값을 제어기 등에 전달하는 장치</p> <ul style="list-style-type: none"> <li>- 비교: 센서의 출력 신호를 전압 또는 전류를 변환하는 모듈과 제어 장치와의 유무선 데이터 송수신 기능을 갖는 것으로 무선 센서 네트워크를 구성하는 기본 요소로 센싱 기능, 전산 기능, 양방향 유무선 통신 기능과 전원장치를 포함</li> </ul>
82	솔레노이드 밸브(solenoid valve)	<p>전자밸브의 한 종류로서 전자 코일의 전자력을 사용하여 자동으로 밸브를 개폐시키는 것</p> <ul style="list-style-type: none"> <li>- 비교 1: 관수 조절을 위하여 주로 사용되며, 전기가 통하면 플랜지가 올라가 밸브가 열리고, 전기가 차단되면 플랜지 무게에 의하여 자동적으로 밸브가 닫힘</li> <li>- 비교 2: '관수 밸브' 비교</li> </ul>
83	수광량(light receive amount)	작물이나 그 잎이 받는 빛의 양으로 작물이 생육하는데 필요한 광량을 말함
84	수분 보존력	'보수력' 참조
85	수분 이용 효율(water use efficiency)	<p>관수된 수분 중 작물이 흡수하여 증발산(evapotranspiration)에 이용한 수분의 비율</p> <ul style="list-style-type: none"> <li>- 비교: 관수된 수분 중 상당한 부분이 지하로 이동하거나 지표면 유거로 손실됨</li> </ul>
86	수소 이온 농도(pH, Potential of Hydrogen)	<p>용액 속에 해리된 수소 이온이나 옥소늄 이온의 농도를 말하며, 일반적으로 수소 이온 농도의 지수로서 용액의 산성 또는 알칼리성의 정도를 나타냄</p> <ul style="list-style-type: none"> <li>- 비교 1: 25°C의 온도에서 비료 성분 등을 전혀 함유하지 않은 순수한 물의 산도(pH)를 7(중성)이라 하고 이보다 낮은 경우를 산성, 높은 경우를 알칼리성이라 함</li> <li>- 비교 2: 측정하는 위치나 대상에 따라 공급 산도, 배액 산도 등으로 분류되며, 양분의 흡수와 밀접한 관계가 있는 것으로 대부분의 작물은 토양 재배 시 6.2~6.8의 산도, 수경 재배 시 5.5~6.2의 산도 범위에서 가장 생육이 좋음</li> <li>- 비교 3 : '산가' 비교</li> </ul>
87	스마트 노지(smart open field)	<p>노지 작물의 생장·수량을 극대화하기 위하여 환경 요소(빛, 온도, 습도 등)를 최적으로 조절할 수 있으며, 이를 원격화·자동화한 노지를 말함</p> <ul style="list-style-type: none"> <li>- 비교 1: 어디서나 작물의 생육 환경을 점검하고 이를 적정하게 유지, 관리하기 위하여 센서 노드나 제어 장치, 구동기 노드 등의 정보통신기술(ICT)이 많이 활용되고 있음</li> <li>- 비교 2: '노지 스마트팜' 비교</li> </ul>
88	스마트 영상 장치(smart video processing device)	<p>촬영 장치(카메라), 전송 장치, 저장장치, 재생장치 등으로 구성된 영상 장치</p> <ul style="list-style-type: none"> <li>- 비교 1: 각종 제어 장치의 작동 상황 및 외부 침입 감지 등을 확인하거나 생육 상태, 병해충 피해 등 작물의 생육 정보 수집을 목적으로 함</li> <li>- 비교 2: 일종의 폐쇄 회로 텔레비전(CCTV) 장치로서 최근 인터넷 프로토콜(IP) 기반의 네트워크 카메라, IP 카메라 시스템을 포함함</li> </ul>
89	스마트팜(smart farm)	<p>스마트 온실의 원격화·자동화·최적환경 제어를 노지, 과수원, 축사 등으로 확장한 개념으로 농업 전반에서 개발되고 있음</p> <ul style="list-style-type: none"> <li>- 비교: 농업의 생산뿐만 아니라 수확 후 가공·유통 단계까지 확장되고 있음</li> </ul>

90	습도 센서(humidity sensor)	습도를 측정하는 장치 - 비교: 습도는 공기 중에 수증기가 포함된 정도를 말하는 것으로 작물이나 가축의 생육에 적정 범위가 존재하며, 작물의 증산이나 병충해, 가축의 스트레스에 직접적인 영향을 미침
91	액추에이터(actuator)	'구동기' 참조
92	양액(nutrient solution)	작물의 생육에 필요한 무기성분을 함유한 물
93	양액기	'관비 공급 장치' 또는 '양액 시스템' 참조
94	양액 시스템(nutrient solution supply system 또는 fertigation system)	관비 공급 장치 또는 양액기와 이를 제어하는 제어가 포함된 시스템 - 비교: '관비 공급 장치' 비교
95	열 화상 카메라(thermographic camera)	물체의 열복사를 전자적으로 측정하여 기록하는 카메라
96	엽록소(chlorophyll)	빛 에너지를 유기 화합물 합성을 통하여 화학 에너지로 전환시키는 녹색 색소 - 비교 1: 광합성에 가장 중요한 요소로 빛에서 에너지를 흡수하여 이산화탄소를 탄수화물로 전환시킴. 녹색식물, 조류(藻類), 광합성 박테리아 등의 광합성을 하는 모든 생물체에서 나타남 - 비교 2: 식물 등의 잎에 있는 엽록체(chloroplast) 속에 황색의 카로틴 및 크산토필과 공존하는 녹색의 호흡 색소 식물의 녹색 색소로 태양의 방사 에너지를 흡수하여 광합성을 행함. 구조적으로 폴리피린류에 속하며 철 대신에 마그네슘을 가짐 - 비교 3: 육상 식물에서는 엽록소 a 및 엽록소 b, 일부 해양 식물에서는 엽록소 c, 광합성 세균에서는 박테리� 엽록소 등이 주요한 엽록소에 해당함
97	엽록체(chloroplast)	식물 잎의 세포 안에 함유된 둥근 모양 또는 타원형의 작은 구조물로서 엽록소를 함유하는 색소체로 녹색을 띠고 있으며 탄소 동화 작용을 하여 녹말을 만드는 중요한 부분 - 비교 1: 이중막으로 둘러싸여 있으며 투명한 스트로마와 동전을 쌓아 둔 모양의 그라나로 이루어짐 - 비교 2: 엽육 세포에는 보통 수 십~수 백 개의 엽록체가 있으며, 엽록체에는 비타민K, 비타민 C, 리보산, 망간 등이 존재하고 누에 장 내에서 미생물의 증식을 억제하는 것으로 알려짐
98	엽면적(leaf area)	식물체의 잎의 면적(cm <sup>2</sup> )을 나타내는 값 - 비교: 주로 하나의 개체(주)의 전체 잎의 면적을 말하기도 함
99	엽면적 지수(LAI, Leaf Area Index)	어떤 군단위 면적당 엽면적의 비율
100	엽색(leaf color)	식물체가 가진 잎의 색깔 - 비교: 식물체의 영양상태나 병해충 피해에 따라 발현되므로 이를 통한 진단이 가능함
101	엽수(number of leaves)	식물체에 달린 잎의 수
102	엽온(leaf temperature)	식물 잎의 온도 - 비교: 잎이 햇볕에 직접 노출되면 다량의 일사 에너지를 흡수하여 잎의 온도가 높아지면서 잎에서 증산 작용이 일어나고 기화열을 통해 잎의 온도가 내려가는 현상을 바탕으로 잎에서의 증산 작용을 파악하는 자료로 사용함
103	엽온 센서(leaf temperature sensor)	엽온을 측정하는 센서 - 비교: 비접촉식과 접촉식이 있음
104	엽장(leaf length)	잎자루를 제외한 잎의 길이 비교: 잎의 최대 길이로서 엽면적을 계산할 때 사용됨
105	엽폭(leaf width)	잎의 너비 - 비교: 잎 길이(엽장)의 성장 방향에 수직으로 제일 넓은 부위의 값으로 엽면적을 계산할 때 사용됨
106	온도 센서(temperature sensor)	온도(°C)를 측정하는 장치 - 비교 1: 온도는 세포의 분화, 분열과 신장, 작물의 호흡량과 관련되므로 작물 재배에 중요한 요소임 - 비교 2: '기온 센서' 비교
107	원수(raw water)	용수원에서 이송되어 물리적, 화학적 처리를 거치지 않은 물
108	유량(flow rate)	유체(流體)가 단위 시간 동안 흐르는 양 - 비교: 유체가 흐를 때 일정 면적의 단면을 통과하는 유체의 체적, 질량 또는

		중량을 시간에 대한 비율로 표시한 것 하천, 수로 또는 관 등에 흐르는 유량(G(kgf/s) 또는 Q(m <sup>3</sup> /s))을 측정하는 센서
109	유량 센서(flow rate sensor)	- 비교: 유체(流體)의 유속은 기벽에서의 거리에 따라 상이하므로 일정 지점의 통과 전후에서 압력이나 수위의 변화를 측정하여 계산함
110	의사 결정 지원 시스템(decision making support system)	의사, 행동 또는 활동 방침의 결정에 필요한 정보를 제공하는 컴퓨터 기반 정보 시스템 - 비교: 어떤 미리 정해진 결정 규칙이나 시뮬레이션 모델에 따라 어떤 요구에 대한 최적의 방안을 구하기 위한 지원 시스템
111	일사량(amount of solar radiation 또는 amount of insolation)	태양의 복사 에너지가 지표에 닿는 양으로 위도에 따라 다름 - 비교 1: 태양 광선에 직각으로 놓인 단위 면적에 단위 시간 동안 복사되는 에너지의 양(W/m <sup>2</sup> )으로 작물의 광합성 양을 추측하거나 온실이나 축사 내 온도 제어, 관수 제어 등에 활용됨 - 비교 2: 특정 기간 동안 누적된 총 일사량을 누적 일사량이라 함
112	일사 센서(pyranometer)	단위 시간에 대하여 단위 면적당 입사되는 일사 에너지(W/m <sup>2</sup> ·h)를 측정하는 장치 - 비교 1: 일사 센서에는 직달 일사 센서, 전천 일사 센서 등이 있으며 측정 위치에 따라 온실 내부 일사와 온실 외부 일사로 구분됨 - 비교 2: 온실이나 축사 내 온도 제어, 관수 제어 등에 활용됨
113	일조 센서(sunshine sensor)	일조 상태 및 시간이나 태양 복사량(solar radiation)을 측정하는 장치
114	자동 기상 관측 장비(AWS, Automatic Weather Station)	기압이나 온도, 습도 등의 기상요소별로 관측 센서를 통해 측정된 값을 기상학적 물리량으로 변환하여 통신망을 통해 전송하는 장비 또는 그러한 체계
115	작물 생육 정보(growth and development information)	초장, 엽수, 엽면적, 엽록소 함유량이나 과실의 수, 색, 당도나 무기성분 함량, 생체중, 수확량 등의 작물 생육의 정도를 추정할 수 있는 정보
116	저면 관수(bottom watering, sub-irrigation)	모세 관수에 의하여 작물이 밑에서부터 물을 흡수하도록 하는 관수 방법 - 비교 1: 잦은 관수는 토양이 단단해져 작물의 성장을 저해하기 때문에 이를 방지하기 위한 관수의 한 방법으로 과채류의 수경 재배나 미세 종자를 파종한 화분 등에 유용함 - 비교 2: '모세 관수' 참조
117	적외선 센서(infrared ray sensor, Ir sensor)	적외선을 이용해 온도, 압력, 방사선의 세기 등의 물리량이나 화학량을 감지하여 신호처리가 가능한 전기량으로 변환시키는 장치 - 비교 1: 스스로 적외선을 방사하여 빛의 차단에 따른 변화를 감지하는 능동식 과 외부로부터의 적외선의 변화만을 감지하는 수동식이 있음 - 비교 2: 온도 센서나 자외선 센서보다 감도나 정확도가 높아 방법이나 화재 감시 등에 널리 사용되고 있으며, 의료용 서모그래피, 동식물의 생태 관찰 등에 응용되고 있음
118	전기 전도도(EC, Electric Conductivity)	물질이나 용액이 전하를 운반할 수 있는 정도 즉, 도체에 흐르는 전류의 크기를 나타내는 상수(dS/m)를 말하는 것으로 온도가 상승하면 금속에서는 크기가 감소하고 반도체에서는 크기가 증가함 - 비교 1: 토양 관개수의 전기 전도도는 작물에 대한 염류 장애를 판단하는데 매우 중요한 화학적 지표로 사용되며, 측정위치나 대상에 따라 배액의 전기 전도도, 급액의 전기 전도도 등으로 분류할 수 있음 - 비교 2: 수용액에 해리된 이온이 많을수록 전기 전도도가 증가하며 양분 농도의 간접적인 지표로 사용됨. 식물은 일반적으로 2~4 mS/cm의 범위에서 재배하는 것이 좋으나 이 이상일 경우에는 위조, 생육억제, 열과 등의 현상이 일어나기 쉬움
119	전자 밸브(electrimagnetic valve)	'관수 밸브' 또는 '솔레노이드 밸브' 참조
120	절대 습도(absolute humidity)	공기 1 m <sup>3</sup> 중에 포함된 수증기의 질량(g) - 비교: 현재 공기 중에 포함된 수증기의 양과 공기가 최대로 포함할 수 있는 수증기의 양(포화 수증기량)의 비(%)를 상대 습도(relative humidity)라고 하며 보통 이를 단순히 습도라고 함
121	점적 관수(drip irrigation)	가는 구멍이 뚫린 관을 지중에 묻거나 지표에 두고 물방울이 떨어지게 하거나 천천히 흘러나오도록 하여 원하는 부위에 대해서만 제한적으로 소량의 물을 지속적으로 공급하는 관수 방법 - 비교 1: 원하는 작물의 부위에 제한적으로 물을 공급함으로써 토양이 굳어지

		<p>는 것을 방지하고 표토의 유실이 거의 없으며 소량의 물로 넓은 면적을 균일하게 관수할 수 있는 장점이 있으나, 관수 장치의 관리가 번거롭고 관수에 장시간이 소요될 수 있음</p> <p>- 비교 2: 과실 나무와 포기 사이가 넓은 채소 작물에 주로 사용함</p>
122	정형 데이터(structured data)	<p>관계형 데이터베이스 시스템의 테이블과 같이 고정된 컬럼에 저장되는 데이터와 파일, 지정된 행과 열에 의해 데이터의 속성이 구별되는 스프레드 시트 형태의 데이터</p> <p>- 비교: '비정형 데이터'와 비교</p>
123	조도 센서(illuminance sensor)	<p>단위 시간에 대하여 단위 면적당 광속밀도(dp/dS)를 측정하는 장치. 단위는 룩스(lux, lx)</p> <p>- 비교: 포토레지스터(photoresistor) 또는 CdS(Cadmiumsulfide)로도 불리는 조도 센서는 빛의 밝기에 따라 저항값이 변하는 특징을 가짐</p>
124	증발산(evapotranspiration)	<p>증발과 증산</p> <p>- 비교: 토양면으로부터의 증발 및 식물체로부터의 증산을 통해서 지구표면에서 대기중으로 수증기가 이동하는 것을 말함</p>
125	지습 센서(soil-humidity sensor)	<p>땅의 습한 정도를 측정하는 장치</p> <p>- 비교: 온실 내 토양, 배지, 양액 등의 작물의 근권부의 습도를 측정하는 장치로서 작물에 따라 적절한 지습을 유지하는 것이 중요함</p>
126	지온 센서(soil-temperature sensor)	<p>근권부의 온도(°C)를 측정하는 장치</p> <p>- 비교: 온실 내 토양, 배지, 양액 등 작물의 근권부 온도를 측정하는 장치로서 지온은 작물의 뿌리 생육에 영향을 미치므로 작물에 적합한 범위를 유지해야 함</p>
127	지중 관수(soil irrigation)	<p>지중에 매설된 급수관으로부터 물이 토양 중에 스며 나오게 하여 작물의 근권이 분포되어 있는 토양 중에 직접 물을 공급하는 관수 방법</p> <p>- 비교: 지표 관수에 따른 토양 유실, 표토 경화, 토양 오염, 토양 전염성 병해 등을 방지할 수 있는 장점이 있으나, 시설비가 많이 소요되고 유지·관리가 복잡할 수 있음</p>
128	지표 관수(surface irrigation)	<p>지표로 물을 흘려 보내 물이 고루 퍼져 땅 속에 스며들게 하는 관수 방법</p>
129	초분광 카메라(hyperspectral camera)	<p>다수(일반적으로 수백 개)의 폭이 매우 미세한 인접 파장 대역에 걸쳐있는 데이터를 동시에 수집하는 센서</p> <p>- 비교: 표면 물질을 탐지, 식별, 평가하여 표적 탐지뿐만 아니라 표적 또는 지형을 분류할 수 있을 정도의 세밀한 신호 분석능력까지 갖춘</p>
130	토양 산도(soil acidity)	<p>토양의 산도(산의 세기)</p> <p>- 비교: 흙이 산성이나 중성 또는 알칼리성 등의 성질을 나타내는 것을 말하며, 작물에 따라 다르나 대개 중성의 흙(pH 5.5-6.5)이 작물 재배에 가장 적합하다고 알려져 있음</p>
131	토양 수분 장력(soil moisture tension)	<p>토양 입자의 표면과 토양 수분간에 작용하는 인력으로 토양 수분이 갖는 에너지는 자유 수분보다 적어지는데 이러한 흡인력을 압력 단위(kPa)로 표시한 것</p> <p>- 비교 1: 토양의 함수율과 달리 수분의 포센셜을 말하는 것으로 적절한 범위를 유지하는 것이 작물 생육에 좋음</p> <p>- 비교 2: 토양 속의 물(수분)은 토양 입자에 강하게 흡착된 것, 약하게 흡착된 것, 공극을 자유롭게 이동하는 것으로 나눌 수 있으며, 토양 입자와 물 사이에 작용하는 힘의 강도를 그에 대응하는 압력, 수주 높이, pF 등으로 표시하는데 이는 토양에서 그 수분을 분리하기 위한 압력의 크기로서 표현됨</p>
132	토양 수분 장력 센서(tensiometer)	<p>토양 수분 장력(kPa)을 측정하는 장치</p> <p>- 비교: 토양 함수율 센서와 달리 토양 입자의 표면과 토양 수분 간에 작용하는 인력을 압력 단위로 표시한 것</p>
133	토양 함수율 센서(soil-moisture sensor)	<p>단위 부피당 토양에 포함된 수분량(%)을 측정하는 장치</p>
134	통합 제어 장치(integrated controller)	<p>'노지 통합 제어기' 참조</p>
135	페로몬 트랩(pheromone trap)	<p>페로몬 등과 같은 성 유인 물질을 이용한 곤충 포획 장치</p> <p>- 비교: 페로몬은 곤충이 냄새로 의사를 전달하는 신호물질을 말하는 것으로,</p>

		이를 인공적으로 합성하여 트랩에 바르거나 넣어 해충을 유인하고 포살할 수 있어 주로 해충의 예찰에 많이 쓰임
136	폐액(waste fluid)	더러워서 버리는 물 또는 용액, 일정 사용 목적이 끝나고 폐기된 액체
137	풍속 센서(wind speed sensor)	바람의 세기(m/s)를 측정하는 장치 - 비교: 온실이나 축사의 구조물 안전 관리, 환기량 예측, 온도 및 습도 관리를 위한 환기창(천창, 측창) 조절 등의 목적에 사용됨
138	풍향 센서(wind-direction sensor)	바람의 방향(°)을 측정하는 장치 - 비교: 온실이나 축사의 온도, 환기창 관리 등의 목적으로 사용됨
139	함수율(moisture content)	함유하고 있는 수분의 비율(%) - 비교 1: 일반적으로 식물체의 함수율은 증량 비율로, 토양이나 배지의 함수율은 용적 비율로 계산됨 - 비교 2: 목재의 경우 섬유 포화점 이하에서는 함수율이 적을수록 강도가 증가됨
140	환경 정보(environmental information)	체계적인 분류 기준과 코드화 작업을 통해 업무에 효율적으로 이용할 수 있도록 한, 자연 환경이나 생활 환경 등에 대한 자료 - 비교: 노지의 센서로부터 수집된 센싱값 정보와 구동기로부터 수집된 값 등 환경과 관련된 측정 정보
141	환경 제어(environmental control)	지상부 환경(광, 온도, 습도, CO <sub>2</sub> 가스 등) 및 지하부 환경 요인(이·화학성, 양분 조성 등)을 제어하여 작물 생육에 적합한 환경을 구현하는 것 - 비교: 환경 제어를 위해 고려하는 환경 요소의 수와 상호 영향에 따라 단순 환경 제어와 복합 환경 제어로 구분

나-4) 스마트온실 서비스 프레임워크 (1종) ('19 제안, '20 제정)

(가) 연구 수행 개요

스마트 온실 서비스는 전통적인 농업기술과 사물인터넷(IoT, Internet of Things)과 같은 ICT(Information and communication technology) 기술이 접목된 것으로 최근 이와 관련된 기술개발과 표준화 노력이 활발하게 진행되고 있다. 이 표준은 각종 센서와 구동기, 제어기, 운영시스템과 서비스 제공 사업자 및 이용자 등으로 구성되는 스마트 온실 서비스의 요구 조건, 참조 모델, 기능 구조 및 인터페이스 등을 정의하기 위한 프레임워크를 제공한다. 이 표준은 ITU-T Y.4466(Framework of smart greenhouse service) 표준을 인용한 표준이다.

(나) 연구 주요 내용

스마트 온실 서비스는 스마트 온실에 설치된 IoT 장치(센서, 구동기 등)를 통해 정밀한 농업을 가능하게 해준다. 스마트 온실 서비스는 환경 및 작물 생육 상태에 대한 정보를 수집하고 이를 분석하여 각 작물의 최적 생육 모델을 도출한다. 스마트 온실 서비스는 이와 같은 최적의 생육 모델을 활용하여 생산성의 극대화는 물론 작물의 품질을 향상시킬 수 있으며, 이용자의 편의성을 증진시킨다. 이 표준은 스마트 온실 서비스의 프레임워크를 위한 요구 조건, 참조 모델, 기능 구조 및 인터페이스를 기술한다.

1) 개요

작물의 생산성과 품질을 개선하고 생산 기간을 조절하기 위해서는 환경 조건(습도, 온도 및 광도 등)을 제어할 수 있는 격리된 공간(isolated space)이 필요하다. 온실(greenhouse)은 이러한 격리 공간을 제공하기 위한 것으로 작물의 재배 환경 제어가 가능하도록 투명한 소재의 벽과 지붕 구조를 갖는다. 단, 온실은 작물의 성장기와 경작 공간에 수반되는 문제를 극복하기 위해 사용될 수 있다. 그러나, 전통적인 온실은 다음과 같은 단점을 갖는다.

- 보다 나은 수확을 위하여 이용자(농가)는 작물 생육 환경을 수동으로 제어해야 한다.
- 이용자는 작물 생육 환경을 지속해서 감시하고 특정 조건에 맞도록 환경을 조절해야 한다.
- 이용자의 경험 부족으로 작물의 경작에 많은 시행착오를 겪을 수 있다.

이러한 단점을 극복하기 위하여 다양한 IoT 기술을 적용할 수 있다. 수집된 수많은 센싱 데이터를 기반으로 재배하고자 하는 작물을 위한 최적의 조건에 적합할 수 있도록 작물 생육 환경이 지능적으로 조절된다. 기존의 전형적인 온실과 IoT 기술이 적용된 온실을 차별화하기 위해 스마트 온실(smart greenhouse, SG)이라는 새로운 용어가 도입되었다. SG는 안정적이고 효율적인 작물 생산을 위해 IoT 기술을 사용하는 온실을 말하는 것으로, IoT 기술에 기반하여 환경 상태와 작물의 생육 상태를 감시하고 작물의 생육 환경을 지능적이고 자동적으로 제어한다. SG를 통해 언제, 어디서든 작물을 경작할 수 있으며 이용자의 개입, 노동 비용, 에너지 소비 및 비료 등의 사용량을 최소화할 수 있다.

다음 그림은 SG의 개념도를 보이는 것으로, 다음과 같은 기능을 갖는다.

- 특정한 최적 생육 모델에 따른 작물 생육 조건의 설정 기능
- 다양한 센서를 이용한 제어기의 환경 상태 및 작물 생육 상태의 감시 기능
- 다양한 구동기를 이용한 제어기의 작물 생육 환경 조절 기능

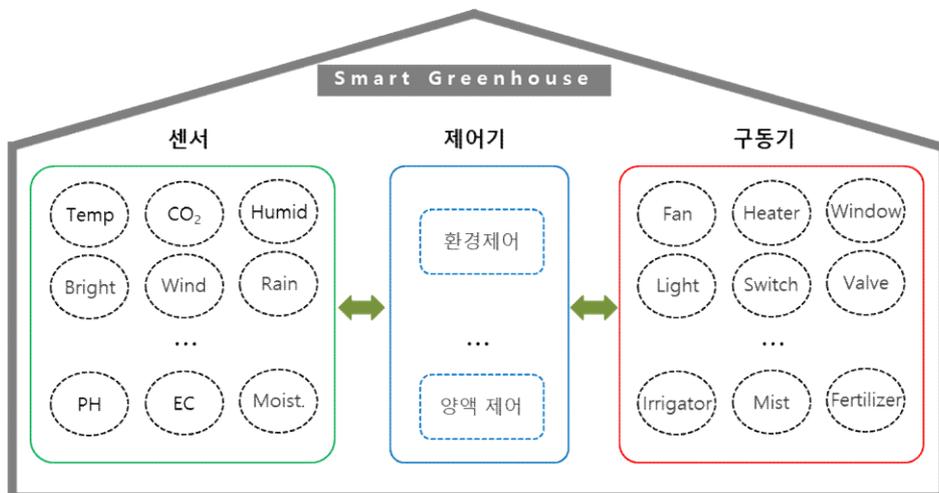


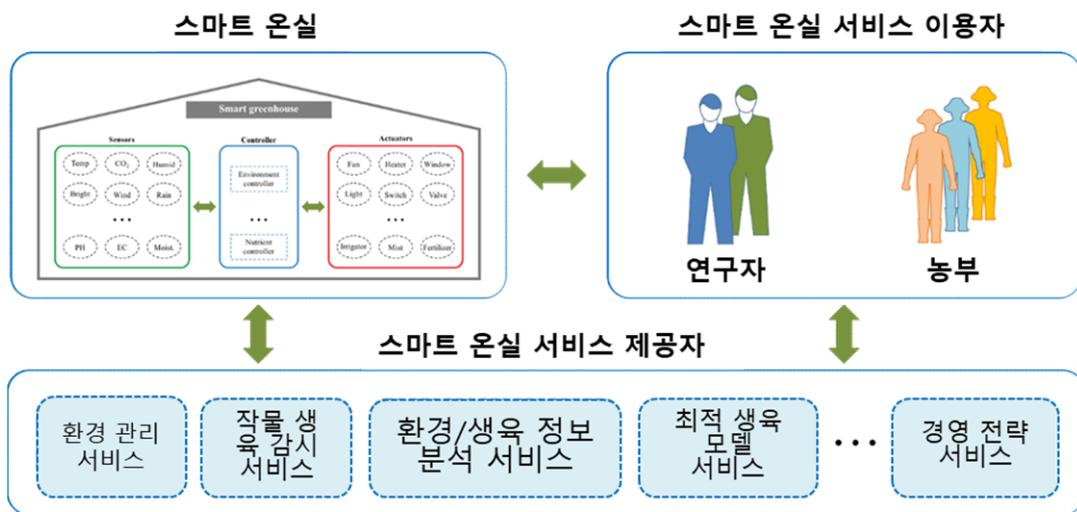
그림 1 - 스마트 온실의 개념도 [출처: ITU-T Y.4466]

다음 그림은 SG 서비스(SG service)의 개념도를 보이는 것으로, SG 서비스는 세 개의 이해관계자가 참여한다. SG는 정밀 농업을 가능하게 해주고, SG 서비스 이용자에게는 경영 전략을 제공할 수 있으며 SG 서비스 제공자는 관련된 다양한 서비스를 제공할 수 있다.

SG 서비스는 다음과 같은 기능을 갖는다.

- 센서를 활용하여 수집된 환경 및 작물 생육 정보간 관계 분석 기능
- 각 작물을 위한 최적 생육 모델의 생성 및 제공 기능
- 최적 생육 모델의 공유 기능
- 최적 생육 모델을 통한 생산성 및 품질 개선 기능
- 이용자 경험 공유를 통한 시행착오의 감소
- 자원 소비 최소화를 위한 경영 전략 수립 기능
- 생산 정보 기반의 마케팅 전략 수립 기능

SG 서비스의 궁극적인 목표는 작물의 생산성 및 품질 극대화뿐만 아니라 이용자 편의성의 증진에 있다.



Y.4466(20)\_F02

그림 2 - 스마트 온실 서비스의 개념도 [출처: ITU-T Y.4466]

## 2) SG 서비스의 요구 조건

SG는 작물의 생산성과 품질 향상을 위한 최적 생육 모델을 설정하기 위하여 하나 이상의 센서와 구동기를 포함하며, 다음과 같은 요구 조건을 갖는다.

- SG는 하나 이상의 센서와 구동기의 설치가 요구된다.
- SG는 SG 내부 및 외부의 환경 조건을 감시하는 기능이 요구된다.
- SG는 작물 생육 조건을 감시하는 기능이 요구된다.
- SG는 환경 상태 및 작물 생육 상태를 포함한 정보를 SG 서비스 이용자에게 실시간 또는 비실시간 방식으로 통보하는 기능이 요구된다.
- SG는 환경 상태 및 작물 생육 상태를 포함한 정보를 SG 서비스 제공자에게 실시간 또는 비실시간 방식으로 보고하는 기능이 요구된다.
- SG는 감시되거나 제어되는 작물 생육 조건 및 환경 조건을 특정할 수 있는 기능이 권고된다.
- SG는 특정 환경 조건을 조절할 수 있는 기능이 요구된다.
- SG는 환경 조건을 조절하기 위해 복수의 센서와 구동기를 동시에 처리할 수 있는 기능이 요구된다.
- SG는 SG 서비스 이용자의 지시에 따라 특정 구동기를 동작시킬 수 있는 기능이 요구된다.
- SG는 SG 서비스 제공자의 지시에 따라 특정 구동기를 동작시킬 수 있는 기능이 권고된다.
- SG는 제어 명령의 실행 결과를 제어 명령을 내린 SG 서비스 이용자나 SG 서비스 제공자에게 보고하는 기능이 요구된다.
- SG는 SG 서비스 제공자에게 작물 경작 정보를 제공하는 기능이 권고된다.
- SG가 자동적으로 또는 SG 서비스 제공자의 지시에 따라 운영되는 경우, SG는 작물 경작 활동과 관련된 정보를 SG 서비스 이용자에게 제공하는 기능을 갖도록 권고된다.

- SG는 작물 경작 활동 기록을 기록하는 기능이 요구된다.
- SG는 환경 상태에 대한 과거 기록 데이터를 유지하는 기능이 요구된다.
- SG는 특정 작물의 상태에 대한 과거 기록 데이터를 유지하는 기능이 권고된다.
- SG의 내구성(tolerant)을 보장하기 위하여 SG는 SG 서비스 이용자 및/또는 SG 서비스 제공자에게 SG에 설치된 장치의 상태 정보를 제공하는 기능이 권고된다.

### 3) SG 서비스 제공자 요구 조건

SG 서비스 제공자는 서비스 이용자에게 SG 서비스를 제공하는 객체로서 다음과 같은 요구 조건을 갖는다.

- SG 서비스 제공자는 작물의 생산성과 품질 극대화를 위한 최적 생육 모델을 수립하기 위해 특정 SG에 대한 환경 상태 및 작물 생육 상태 정보를 실시간 또는 비실시간 방식으로 수집하는 기능이 요구된다.
- SG 서비스 제공자는 최적 생육 모델을 수립하여 작물의 생산성과 품질을 극대화하기 위하여 어떤 SG의 작물 경작 정보를 실시간 또는 비실시간 방식으로 수집하는 기능이 요구된다.
- SG 서비스 제공자는 SG 서비스 이용자의 수익을 극대화하기 위한 경영 전략을 수립하기 위하여 SG 서비스 이용자가 소비한 자원 정보를 수집하는 기능이 권고된다.  
[비고] 소비된 자원 정보에는 전기, 물 및 비료 등의 사용 등이 포함된다.
- SG 서비스 제공자는 SG 서비스 이용자의 수익을 극대화하기 위한 경영 전략을 수립하기 위하여 SG 서비스 이용자의 생산 정보를 수집하는 기능이 권고된다.
- SG 서비스 제공자는 SG뿐만 아니라 SG 서비스 이용자로부터 수집된 정보에 기반하여 일련의 최적 생육 모델을 수립할 수 있는 기능이 요구된다.
- SG 서비스 제공자는 푸쉬(push) 또는 주문형(on-demand) 방식으로 최적 생육 모델을 SG 서비스 이용자에게 제공하는 기능이 요구된다.
- SG 서비스 제공자는 SG 서비스 이용자의 수익을 극대화하기 위하여 시장 정보를 SG 이용자에게 선택적으로 제공할 수 있어야 한다.  
[비고] 시장 정보는 SG 서비스 이용자가 사용할 수 있는 것으로, 작물 가격의 예측을 위한 정보를 말한다.
- SG 서비스 제공자는 데이터의 저장, 분석 및 활용을 위하여 외부의 데이터 센터와 상호 연동할 수 있는 기능을 갖출 것이 권고된다.

### 4) SG 서비스 이용자 요구 조건

SG 서비스 이용자는 SG 서비스 제공자의 서비스를 소비하는 객체로서 다음과 같은 요구 조건을 갖는다.

- SG 서비스 이용자는 작물의 생산성 및 품질 극대화, 자원 소비 최소화를 위하여 SG 서비스 제공자에게 자원 소비에 대한 정보 제공 기능이 권고된다.
- SG 서비스 이용자는 농산물의 생산량과 수확 시기를 조절하여 SG 서비스 이용자의 수익을 극대화하기 위하여 SG 서비스 제공자에게 생산 정보의 제공 기능이 권고된다.  
[비고] 생산 정보는 농산물의 판매 가격 등이 포함된다.
- SG 서비스 이용자는 자동적인 운영이 가능하도록 SG 서비스 제공자로부터의 최적 생육 모델 중 하나를 승인하고 적용하는 기능이 요구된다.
- SG 서비스 이용자는 SG 서비스 제공자와의 통신할 수 없는 경우 SG를 수동으로 조작할 수 있는 기능이 요구된다.

### 5) SG 서비스 참조 모델

그림 3은 ITU-T Y.4000의 IoT 참조 모델에 따른 SG 서비스의 참조 모델을 보인다. IoT 참조 모델은 응용 계층, 서비스 지원 계층 및 응용 지원 계층, 네트워크 계층 및 장치 계층과 관리 및 보안 기능으로 구성된다. IoT 참조 모델의 계층과 기능 중에서 SG 서비스 참조 모델은 SG 서비스를 위한 6가지의 기능을 추가로 정의한다. 이와 관련하여 SG 서비스 참조 모델의 구성 요소로서 다음과 같은 기능이 정의된다.

- SG 센싱(sensing) 기능
- SG 구동(actuating) 기능

- SG 제어(control) 기능
- SG 운영(operation) 기능
- SG 통합(integration) 기능
- SG 관리(management) 기능

SG 서비스 참조 모델의 각 기능은 (그림 8-1)과 같으며, 6절과 7절에서 각각 기술된 SG 서비스의 개념 및 요구 조건에 부합될 수 있도록 정의된다.

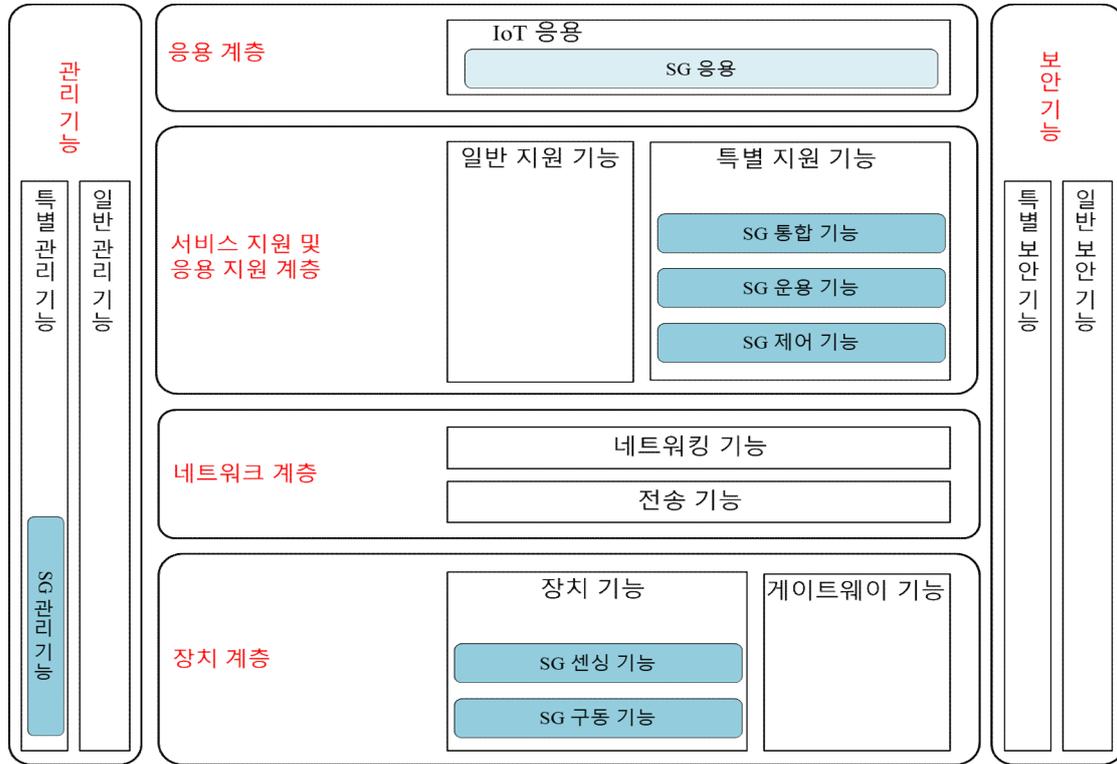


그림 3 - ITU-T Y.4000의 IoT 참조 모델에 따른 SG 서비스 참조 모델  
[출처: ITU-T Y.4466]

그림 3의 각 기능블록의 역할은 다음과 같다.

- SG 센싱 기능  
SG 센싱 기능은 SG에 설치된 SG의 환경 상태 및 작물 생육 상태 정보를 수집하고 이를 SG 제어 기능으로 전달한다. SG 센싱 기능은 또한 각 센서에 대한 정보(형태, 위치, 상세 규격 및 센싱 값의 단위 등)를 SG 관리 기능으로 전달한다. 참고로, 하나의 SG는 다양한 센서를 위한 복수의 SG 센싱 기능을 가질 수 있다.
- SG 구동 기능  
SG 구동 기능은 SG 제어 기능으로부터 수신한 제어 명령에 따라 SG에 설치된 구동기를 동작시킨다. SG 구동 기능은 각 구동기에 대한 정보(형태, 위치, 상세 규격 및 동작 상태 등)를 SG 관리 기능으로 전달한다. 참고로, 하나의 SG는 다양한 구동기를 위한 복수의 SG 구동 기능을 가질 수 있다.
- SG 제어 기능  
SG 제어 기능은 SG 운영 기능으로부터 수신한 최종적인 생육 모델 정보 또는 어느 한 구동기의 동작 상태 정보에 따라 구동기를 동작시키기 위한 구동 제어 명령을 생성한다. 제어 명령은 SG에 설치된 각 구동기의 기능과 각 구동기에 대한 이용자의 요구사항을 반영한다. SG 제어 기능은 SG 구동 기능으로 제어 명령을 전달하고 또한 SG 센싱 기능과 SG 구동 기능으로부터 수신한 환경 상태 및 작물 생육 상태 정보를 SG 운영 기능으로 전달한다. 참고로, 하나의 SG는 하나의 제어 기능을 갖는다. 그러나, SG 센싱 기능 또는 SG 구동 기능이 특정한 목적을 위해 그룹화 되는 경우에는 하나의 SG가 복수의 SG 제어 기능을 가질 수 있으며, SG 제어 기능 이용자는 농부 또는 SG 관리자가 될 수 있다.
- SG 운영 기능

SG 운영 기능은 이용자의 경험을 반영하여 특정 작물을 위한 최종적인 환경 제어 조건을 생성한다. 또한, SG 운영 기능은 SG 제어 기능으로부터 수신한 환경 상태 및 작물 생육 상태 정보를 SG 통합 기능으로 전달한다. 단, 하나의 스마트팜(smart farm, SF)은 SG 운영 기능을 가지며, SG 운영 기능은 복수의 SG를 관리할 수 있다. 농부는 일반적으로 하나의 스마트팜에서 하나 이상의 작물을 경작하는데 작물들은 다양한 생육 환경 조건을 요구하게 된다. 유사한 환경 조건을 필요로 하는 작물들은 하나의 SG에서 경작할 수 있지만 서로 다른 환경 조건을 요구하는 작물들은 그렇지 않다. 따라서 어느 하나의 SG에 대한 환경 조건은 경작하고자 하는 특정 종류의 작물을 결정하게 된다. 이에 경작하는 다양한 작물에 대한 환경 조건 및 작물 생육 조건 정보를 저장해야 할 필요가 있으며, 이는 이용자의 요구 조건에 부합하는 최종적인 최적 생육 모델을 수정하기 위한 SG 운영 기능에 있어서 중요한 부분이다.

- SG 통합 기능

SG 통합 기능은 SG 운영 기능으로부터의 환경 상태 및 작물 생육 상태 정보를 축적한다. 축적되는 정보의 범위는 SG 단위가 아닌 스마트팜(SF)에 해당한다. 하나의 SF에 대한 정보를 축적한 뒤에는 전문가 그룹(expert group) 또는 전문가 시스템(expert system)을 통해 최적 생육 모델을 생성하고 이를 저장한다. 생성된 최적 생육 모델은 또한 이용자의 추가적인 요구사항을 반영하기 위하여 다시 SG 운영 기능으로 전달된다.

이 표준에서는 농업 분야에 대한 광범위한 기술이나 지식을 가진 전문가 그룹을 표현하기 위해 '전문가 그룹'이라는 용어를 사용한다. 또한 '전문가 시스템'이란 용어를 사용하여 농업 분야에 대한 전문적인 자문이나 결정을 내릴 수 있는 컴퓨터 프로그램을 표현한다. 또한 SG 통합 기능은 각 스마트팜(SF)을 위한 최적 생육 모델을 생성할 수 있다. 일반적으로 작물은 다양한 환경을 갖는 복수의 스마트팜에서 재배되기 때문에 작물 생육에 대한 환경적인 영향을 알아야 한다. 이는 다양한 스마트팜에서 재배되는 특정 작물을 위한 환경 상태 및 작물 생육 상태 정보를 분석함으로써 얻을 수 있다. 따라서 수많은 스마트팜에서 재배되는 다양한 작물을 위한 환경 상태 및 작물 생육 상태 정보를 수집해야 한다.

- SG 관리 기능

SG 관리 기능은 하나의 스마트팜에 있는 복수의 SG에서 환경 상태 및 작물 생육 상태 정보를 관리한다. 하나의 스마트팜은 하나 이상의 SG로 구성된다. SG는 SG에 설치된 SG 센싱 기능 및 SG 구동 기능으로부터 각각 센서 및 구동기 정보를 수집한다. 또한, SG 관리 기능은 각 정보의 전달을 위하여 SG 제어 기능, SG 운영 기능 및 SG 통합 기능과 상호 작용을 한다. 이때, SG 관리 기능은 복수의 SG에 대한 정보를 유지 관리한다.

SG 서비스에는 두 개의 정보 흐름이 존재한다. 하나는 각 SG의 환경 상태 및 작물 생육 상태 정보를 전달하는 흐름이고, 다른 하나는 각 SG로 제어 명령을 전달하는 흐름이다. (그림 8-2)는 SG 서비스 참조 모델 측면에서 기능간 높은 수준의 정보 및 제어 명령 흐름을 보여준다. SG 서비스를 위한 정보 흐름의 예는 다음과 같다.

- SG 환경 및 작물 생육 정보의 생성과 소비 : 하나의 SG에서 SG 센싱 기능이 생성한 환경 상태 및 작물 생육 상태 정보는 SG 제어 기능에 의해 수집된다. 이렇게 수집된 SG 정보는 SG 운영 기능으로 전달되며, 이 정보는 다시 SG 통합 기능으로 전달된다. 마지막으로 SG 통합 기능은 전문가 그룹 또는 전문가 시스템과 같은 이용자에게 이 정보를 전달한다.
- 작물 생육 조건의 생성과 제어 명령의 실행 : 이전의 정보 흐름에 의해 제공된 정보는 각 작물을 위한 최적 생육 조건을 전달하기 위하여 전문가 그룹 또는 전문가 시스템에서 활용된다. SG 통합 기능은 전문가 그룹 또는 전문가 시스템으로부터 최적의 작물 생육 조건을 가져와서 유지한다. 그다음 SG 운영 기능으로 최적의 작물 생육 조건을 전달하고 SG 운영 기능은 SG 통합 기능에 의해 제공된 최적 생육 조건을 이용자가 추가로 수정할 수 있도록 한다. 그리고 수정된 최적 생육 조건은 SG 제어 기능으로 전달되며, 마지막으로 SG 제어 기능은 적절한 SG 구동 기능으로 일련의 제어 명령을 발행하여 구동기가 적절한 작동을 수행하도록 한다.

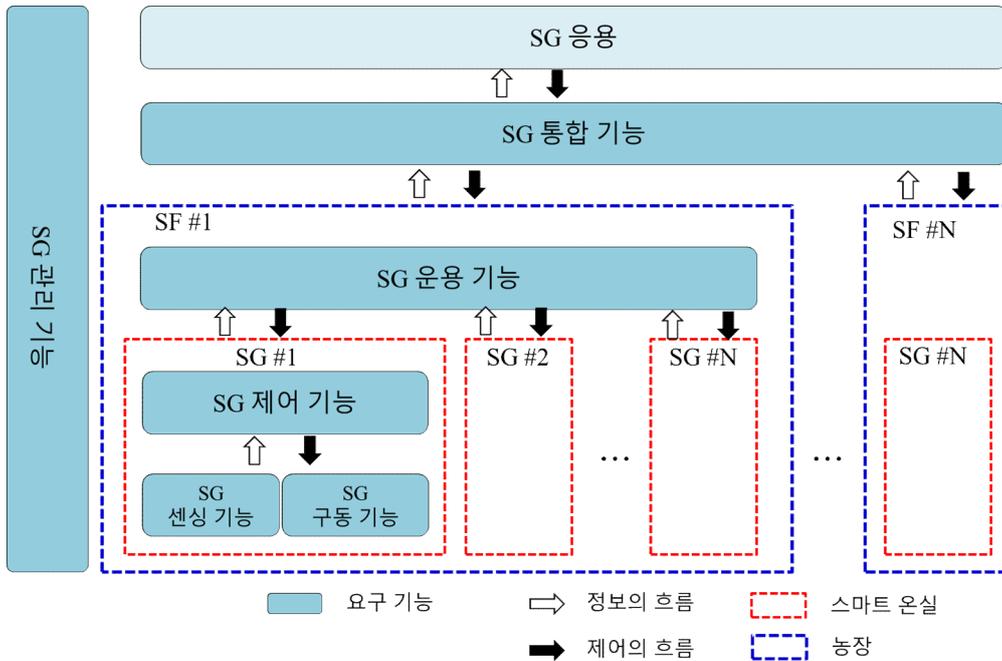


그림 4 - SG 서비스를 위한 높은 수준의 정보 및 제어 명령의 흐름 [출처: ITU-T Y.4466]

#### 6) SG 서비스 기능 구조

그림 4는 SG 서비스를 위한 기능 구조를 보인다.

- SG 센싱 기능

SGS-EnvInfoProcFE와 SGS-GrowthInfoProcFE의 기능 객체(Functional Entity, FE)로 구성되는 SG 센싱 기능은 환경 정보 및 작물 생육 정보를 수집하여 SG 제어 기능으로 전달한다. 각 FE의 세부 사항은 다음과 같다.

- SGS-EnvInfoProcFE(SG Sensing-Environment Information Processing FE)는 다양한 환경 센서(IoT 장치)로부터 환경 정보를 수집하고 이를 SGC-EnvInfoProcFE로 전달한다.  
[비고] 환경 정보는 센서의 형태에 따라 온도, 습도 또는 CO<sub>2</sub> 농도 등 다양한 포맷의 센싱 값을 갖는다.
- SGS-GrowthInfoProcFE (SG Sensing-Growth Information Processing FE)는 다양한 생육 센서(IoT 장치)로부터 작물 생육 정보를 수집하고 이를 SGC-GrowthInfoProcFE로 전달한다.  
[비고] 작물 생육 정보는 센서의 형태에 따라 생장 길이(growth length), 줄기의 지름이나 엽장(leaf length) 등 다양한 포맷의 센싱 값을 갖는다.

- SG 구동 기능

SGA-ActContCmdProcFE로 구성된 SG 구동 기능은 SG 제어 기능으로부터 제어 명령을 받아 이를 해당 구동기로 전달한다. 세부 사항은 다음과 같다.

- SGA-ActContCmdProcFE(SG Actuating-Actuator Control Command Processing FE)는 SGC-ActContCmdGenFE로부터 일련의 제어 명령뿐만 아니라 구동기의 세부 속성을 받아 이에 대응하는 구동기(관련된 IoT 장치)로 전달한다.  
[비고 1] 각 구동기의 제어 명령들은 구동기의 형태에 따라 다르다. 구동기에 대한 제어 명령에는 'open/close', 'on/off'와 'operate/stop' 등이 있다.  
[비고 2] 구동기의 속성에는 'operation time', 'operation duration', 'open position'과 'operation speed' 등이 있다.

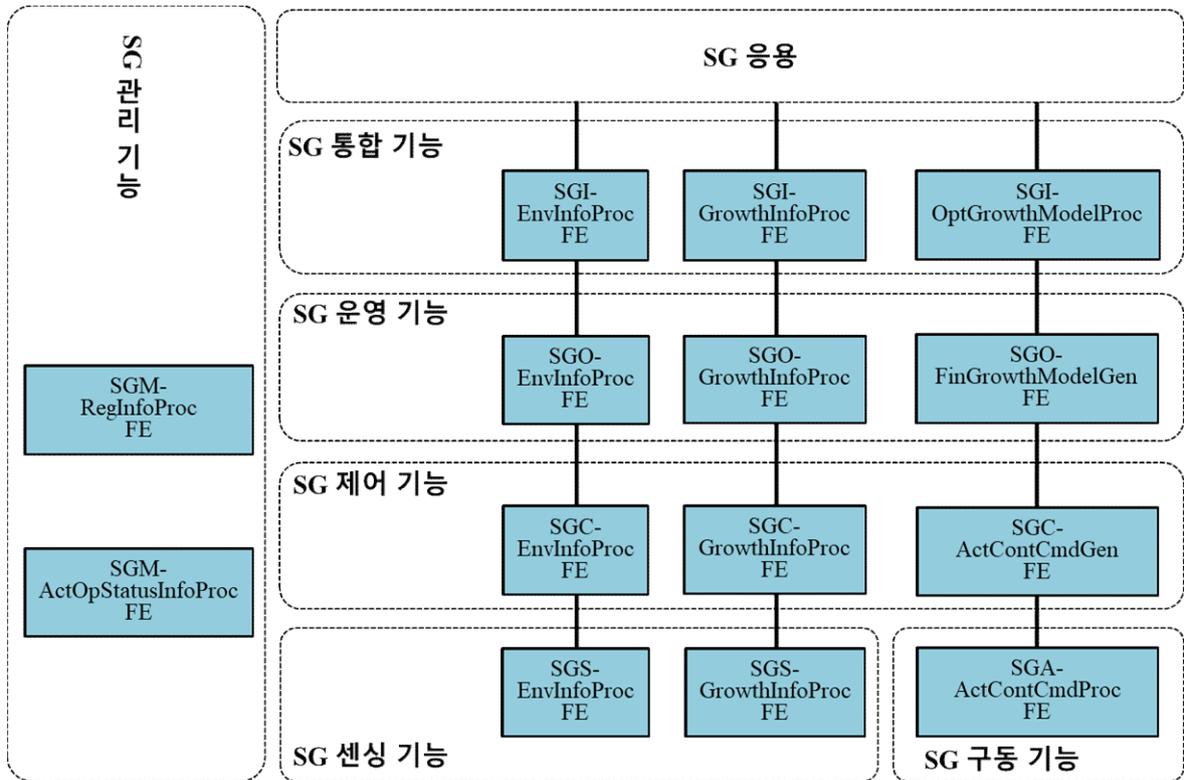


그림 5 - SG 서비스의 기능 구조 [출처: ITU-T Y.4466]

• SG 제어 기능

SG 제어 기능은 SG의 환경 상태 및 작물 생육 상태에 대한 정보를 수집하여 이를 SG 운영 기능으로 전달한다. 또한, SG 제어 기능은 SG 운영 기능으로부터 재배되는 작물을 위한 환경 조건을 수신하고 적절한 구동기에게 일련의 제어 명령을 보낸다. SG 제어 기능의 각 FE의 세부 사항은 다음과 같다.

- SGC-EnvInfoProcFE(SG Control-Environment Information Processing FE)는 SG 센싱 기능의 SGS-EnvInfoProcFE를 통해 SG의 환경 정보를 수집하고 구동기의 동작을 조정하기 위하여 수집된 정보를 SGC-ActContCmdGenFE에게 전달한다.

[비고] 환경 정보는 '특정 시간에', '주기적으로', '주문형' 또는 '이벤트 발생 시' 등으로 제공되어야 한다.

- SGC-GrowthInfoProcFE(SG Control-Growth Information Processing FE)는 SGS-GrowthInfoProcFE를 통해 SG의 작물 생육 정보를 수집하고 이를 SGO-GrowthInfoProcFE와 SGC-ActContCmdGenFE로 전달한다.

[비고] 작물 생육 정보는 '특정 시간에', '주기적으로', '주문형' 또는 '이벤트 발생 시' 등으로 제공되어야 한다.

- SGC-ActContCmdGenFE(SG Control-Actuator Control Command Generation FE)는 SGO-FinGrowthModelGenFE로부터 최종적인 생육 모델을 수집한다. 또한, 이 FE는 SGC-EnvInfoProcFE와 SGC-GrowthInfoProcFE로부터 각각 환경 상태 및 작물 생육 상태 정보를 수집하고, SG 관리 기능의 SGM-ActOpStatusInfoProcFE로부터 각 구동기의 동작 상태에 대한 정보를 수집한다. SG의 환경 조건에 따라 SGC-ActContCmdGenFE는 SGA-ActContCmdProcFE를 위하여 적절한 제어 명령을 생성한다.

[비고 1] 제어 명령은 구동기의 프로파일, 상태 및 기능에 대한 다양한 정보 항목에 따라 생성된다. 이러한 모든 정보 항목은 SG 관리 기능에 있는 SGM-ActOpStatusInfoProcFE로부터 얻을 수 있다.

[비고 2] 구동기 그룹은 최적 생육 모델에 부합되는 목표 기간 및 목표값과 관련된 제어 명령에 따라 선택된다.

[비고 3] 현재의 제어 명령이 최적 생육 모델에 부합되는 환경 조건을 생성할 수 없는 경우에는 구동기 그룹을 다시 선택해야 한다.

[비고 4] 최종적인 생육 모델에 따른 제어 명령은 SG 제어 기능이나 SG 운용 기능에 의해 만들어질 수 있다.

• SG 운영 기능

SGO-EnvInfoProcFE, SGO-GrowthInfoProcFE 및 SGO-FinGrowthModelGenFE로 구성되는 SG 운용 기능은 SG 제어 기능으로부터 환경 상태 및 작물 생육 상태 정보를 수집하고, 최적 생육 모델 생성을 위하여 이 정보를 SG 통합 기능으로 전달한다. SG 운용 기능의 각 FE에 대한 세부 사항은 다음과 같다.

- SGO-EnvInfoProcFE(SG Operation-Environment Information Processing FE)는 SG 제어 기능의 SGC-EnvInfoProcFE를 통해 SG의 환경 정보를 수집하고, 이 환경 정보를 SG 통합 기능에 있는 SGI-EnvInfoProcFE로 전달한다.

[비고] SGO-EnvInfoProcFE에 의해 수집되는 환경 정보에는 각 SG의 기상 조건이 포함된다.

- SGO-GrowthInfoProcFE(SG Operation-Growth Information Processing FE)는 SGC-GrowthInfoProcFE를 통해 각 SG의 작물 생육 정보를 수집하고 이를 SG 통합 기능에 있는 SGI-GrowthInfoProcFE와 SG 운용 기능에 있는 SGO-FinGrowthModelGenFE로 전달한다.

[비고 1] SGO-GrowthInfoProcFE에 의해 수집되는 정보에는 각 SG의 작물 생육 정보가 포함된다.

[비고 2] SGO-FinGrowthModelGenFE는 특정 작물을 위한 최종의 최적 작물 생육 모델을 생성하기 위해 작물 생육 정보를 이용한다.

- SGO-FinGrowthModelGenFE(SG Operation-Final Growth Model Generation FE)는 SG 통합 기능에 있는 SGI-OptGrowthModelProcFE로부터 일반적인 최적 생육 모델들을 수집하고 SG 제어 기능에 있는 SGS-ActContCmdGenFE를 위한 최종적인 최적 생육 모델을 생성한다. 또한, 이 FE는 최적 생육 모델을 생성하는 동안에 SG 서비스 이용자의 개입을 허용함으로써, SG 서비스 이용자의 맞춤형 최종 생육 모델을 도출하기 위하여 생성된 최종의 최적 생육 모델에 대한 SG 서비스 이용자의 추가적인 수정을 허용한다.

- SG 통합 기능

SGI-EnvInfoProcFE, SGI-GrowthInfoProcFE 및 SGI-OptGrowthModelProcFE로 구성되는 SG 통합 기능은 환경 상태 및 작물 생육 상태 정보를 축적하고 이 정보를 전문가 그룹 및 전문가 시스템에게 제공한다. 각 FE의 세부 사항은 다음과 같다.

- SGI-EnvInfoProcFE(SG Integration-Environment Information Processing FE)는 SGO-EnvInfoProcFE로부터 하나의 스마트팜에 있는 SG들의 환경 정보를 수집하고 이 정보를 SG 응용 서비스로 전달한다.

[비고] SG 응용 서비스는 전문가 그룹 및 전문가 시스템을 포함한다.

- SGI-GrowthInfoProcFE(SG Integration-Growth Information Processing FE)는 SGO-GrowthInfoProcFE로부터 하나의 스마트팜에 대한 작물 생육 정보를 수집하고 요청 시, 이 정보를 SG 응용서비스로 전달한다.

- SGI-OptGrowthModelProcFE(SG Integration-Optimal Growth Model Processing FE)는 SG 응용 서비스로부터 하나의 스마트팜을 위한 일반적인 최적 생육 모델들을 수집하고 요청 시, SGO-FinGrowthModelGenFE에게 최적 생육 모델들을 제공한다.

[비고] 일반적인 최적 생육 모델은 작물의 생애주기 측면에서 잘 알려진 최적의 작물 생육 조건을 설명한다.

- SG 관리 기능

SGM-RegInfoProcFE와 SGM-ActOpStatusInfoProcFE로 구성되는 SG 관리 기능은 하나의 스마트팜에 있는 SG들의 환경 상태 및 작물 생육 상태 정보를 유지·관리한다. 각 FE에 대한 세부 사항은 다음과 같다.

- SGM-RegInfoProcFE(SG Management-Registration Information Processing FE)는 IoT 장치(센서, 구동기 등)에 대한 정보를 기술한다. 이 FE는 SG 센싱 기능과 SG 구동 기능으로부터의 정보를 수집하고 관리한다.

[비고 1] 센서와 관련된 정보는 형태와 세부 규격, 위치, 센싱 목표 및 센싱 값의 단위 등과 관련된다.

[비고 2] 구동기와 관련된 정보는 형태와 세부 규격, 위치, 구동 목표 및 동작 속도 등과 관련된다.

- SGM-ActOpStatusInfoProcFE(SG Management-Actuator Operation Status Information Processing FE)는 SGA-ActContCmdProcFE로부터 구동기의 동작 상태 정보를 수집한다. 또한, 이 FE는 최종 생육 모델의 작물 생육 조건을 업데이트하기 위하여 수집된 정보를 SGO-FinGrowthModelGenFE에게 제공하고, 관련된 구동기를 위한 제어 명령을 생성하기 위하여 수집된 정보를 SGC-ActContCmdGenFE에게 제공한다.

## 7) SG 서비스를 위한 인터페이스

이 절에서는 그림 6에서 보는 바와 같이 서로 관련된 기능들간 참조 포인트(reference point)에 대해서 다룬다. 그림 6의 화살표는 기능들간 논리적인 데이터의 흐름을 나타낸다. 이 표준에서는 다음과 같은 인터페이스를 정의한다.

- 인터페이스 R1은 SG 통합 기능과 SG 운영 기능 사이에서 환경 상태 및 작물 생육 상태 정보의

전달뿐만 아니라 SG의 일반적인 최적 생육 모델 전달을 위해 사용된다.

- 인터페이스 R2는 SG 운영 기능과 SG 제어 기능 사이에서 환경 상태 및 작물 생육 상태 정보의 전달뿐만 아니라 SG의 최종적인 최적 생육 모델 전달을 위해 사용된다.
- 인터페이스 R3는 SG 제어 기능과 SG 센싱 기능 사이에서 SG의 환경 상태 및 작물 생육 상태 정보를 전달하기 위해 사용된다.
- 인터페이스 R4는 SG 제어 기능과 SG 구동 기능 사이에서 구동기 모듈을 동작시키기 위한 제어 명령의 전달과 (또는) 구동기 모듈의 상태 정보를 전달하기 위해 사용된다.
- 인터페이스 R5는 SG 관리 기능과 SG 센싱 기능 사이에서 형태, 제조사의 규격, 위치 및 특성 등을 포함한 정보를 갖는 IoT 센서 모듈의 등록을 위해 사용된다.
- 인터페이스 R6는 SG 관리 기능과 SG 구동 기능 사이에서 형태, 제조사의 규격, 위치 및 특성 등을 포함한 정보를 갖는 IoT 구동기 모듈의 등록을 위해 사용된다.
- 인터페이스 R7은 SG 관리 기능과 SG 제어 기능 사이에서 SG에 설치된 IoT 센서 모듈 및 IoT 구동기 모듈과 관련된 정보를 전달하기 위해 사용된다.
- 인터페이스 R8은 SG 관리 기능과 SG 운영 기능 사이에서 스마트팜에 설치된 IoT 센서 모듈 및 IoT 구동기 모듈과 관련된 정보를 전달하기 위해 사용된다.
- 인터페이스 R9은 SG 관리 기능과 SG 통합 기능 사이에서 하나 이상의 스마트팜에 설치된 IoT 센서 모듈 및 IoT 구동기 모듈과 관련된 정보를 전달하기 위해 사용된다.
- 인터페이스 R10은 SG 서비스 이용자와 SG 통합 기능 사이에서 SG의 환경 상태 및 작물 생육 상태와 관련된 정보를 전달하기 위해 사용된다.
- 인터페이스 R11은 SG 서비스 이용자와 SG 운영 기능 사이에서 SG의 환경 상태 및 작물 생육 상태와 관련된 정보를 전달하기 위해 사용된다.
- 인터페이스 R12는 SG 서비스 이용자와 SG 제어 기능 사이에서 SG의 환경 상태 및 작물 생육 상태와 관련된 정보를 전달하기 위해 사용된다.
- 인터페이스 R13은 SG 서비스 이용자와 SG 센싱 기능간 센싱 데이터를 전달하기 위해 사용된다.
- 인터페이스 R14는 SG 서비스 이용자와 SG 구동 기능 사이에서 구동기의 상태 정보를 전달하기 위해 사용된다.
- 인터페이스 R15는 SG 센싱 기능과 센서 모듈 사이에서 센싱 값을 읽어 들이기 위해 사용된다.
- 인터페이스 R16은 SG 구동 기능과 구동기 모듈 사이에서 구동기 모듈을 동작시키기 위해 사용된다.

[비고] 인터페이스 R10, R11, R12, R13, R14, R15 및 R16은 이 표준의 적용 범위에 포함되지는 않는다.

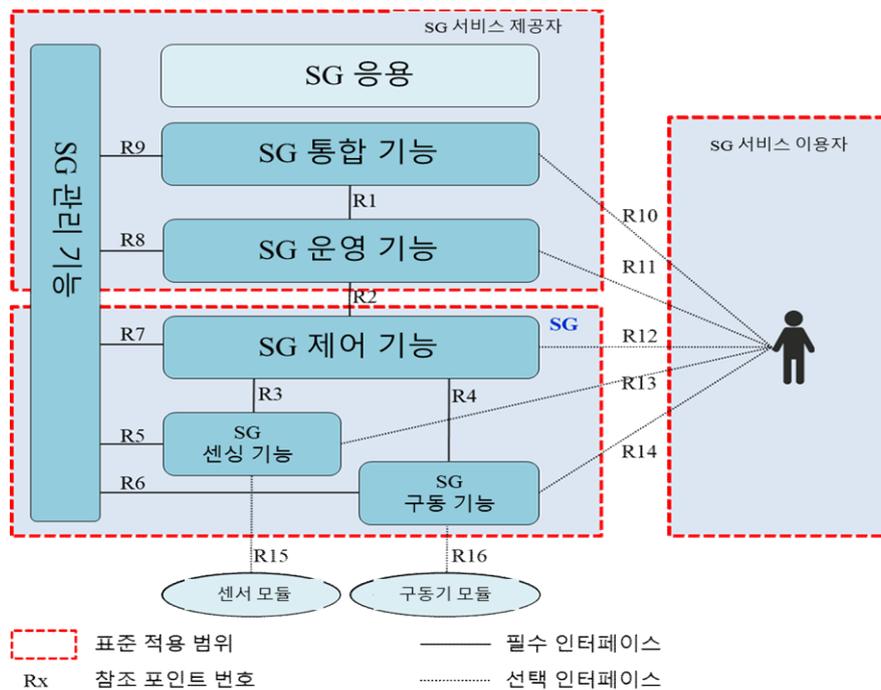


그림 6 - 관련 기능간 참조 포인트 [출처: ITU-T Y.4466]

## 8) 보안 고려사항

이 표준의 프레임워크는 보안 사항과 관련된 어떤 특정한 요구 조건을 기술하지는 않으며, 이 표준의 프레임워크를 위해 잘 알려진 기존의 보안 기능이 적용될 수 있다.

**나-5) 네트워크기반 스마트팜 개요 (1종) ('19 제안, '20 개정)**

소규모의 독립적 스마트팜을 구축하고 운영할 때는 네트워크를 통한 연결이 불필요할 수 있으나, 네트워크 기반의 보다 체계적인 모니터링과 제어를 통한 스마트팜을 관리하여 스마트팜운영의 편리성을 높이고 재배의 효과성을 높일 수 있다. 이러한 관점에서 네트워크 기반의 스마트팜에 대한 개념을 정립하는 것은 매우 중요할 것이다. 이러한 배경으로 ITU-T SG13에서 개발된 국제표준 ITU-T Rec. Y.2238은 ITU-T SG20으로 이관되어 ITU-T Rec. Y.4450/Y.2238으로 변경되었다. ITU-T Rec. Y.2238을 기반으로 한 이에 TTAE.IT-2238은 농업의 계획, 생산, 유통 및 마케팅 분야에 이르기까지 체계적인 생산 및 활용을 위한 환경조성에 관한 기본 표준 권고안으로 스마트 기본 구조로 계획단계(pre-production stage), 생산 단계(Production stage), 유통 및 마케팅 단계(post-production stage)로 규정하고 있다. ICT기반의 스마트 농업의 모형을 기반으로 하여 농산물에 IT 기술을 적용하여 생산성 향상을 목표로 하고 있으며, 그 개정본인 TTAE.IT-2238/R1은 2020년도 한국정보통신기술협회(TTA)의 개정 단체표준으로 제안하여 12월에 최종 채택되었다.

이 표준은 먼저 네트워크 기반 스마트 팜을 소개한다. 기반 스마트 팜은 기후나 작물 재배 환경, 병해 등의 다양한 문제를 해결하기 위하여 농업과 ICT가 결합된 형태로서, 네트워크 기반 스마트 팜은 부가적으로 배터리 문제와 같은 기술적 문제나 센서의 오작동 등을 해결하고 자동 제어 기능을 제공할 수 있다. 이러한 네트워크 기반 스마트 팜은 아래의 그림처럼 농업인, 서비스 사업자, 물류 업체, 시장, 소비자간 상호작용을 고려할 필요가 있다.

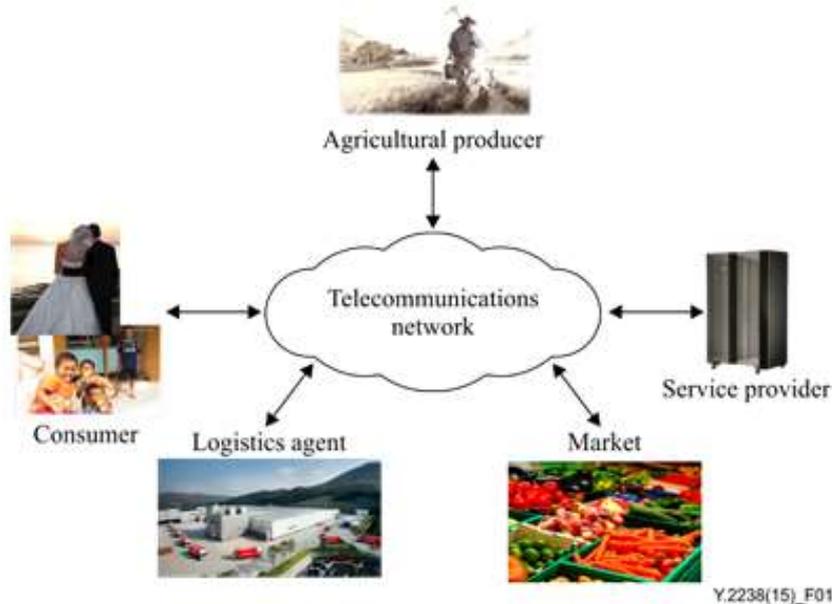


그림 1 - 네트워크 기반 스마트 팜의 개념도

전술한 네트워크 기반 스마트 팜을 구현하기 위해서 네트워크 기반 스마트 팜의 참조 모델을 이어서 기술한다. 다음 그림은 네트워크 기반 스마트 팜의 참조 모델이다. 참조 모델을 구성하는 도메인별 주요 역할은 다음과 같다.

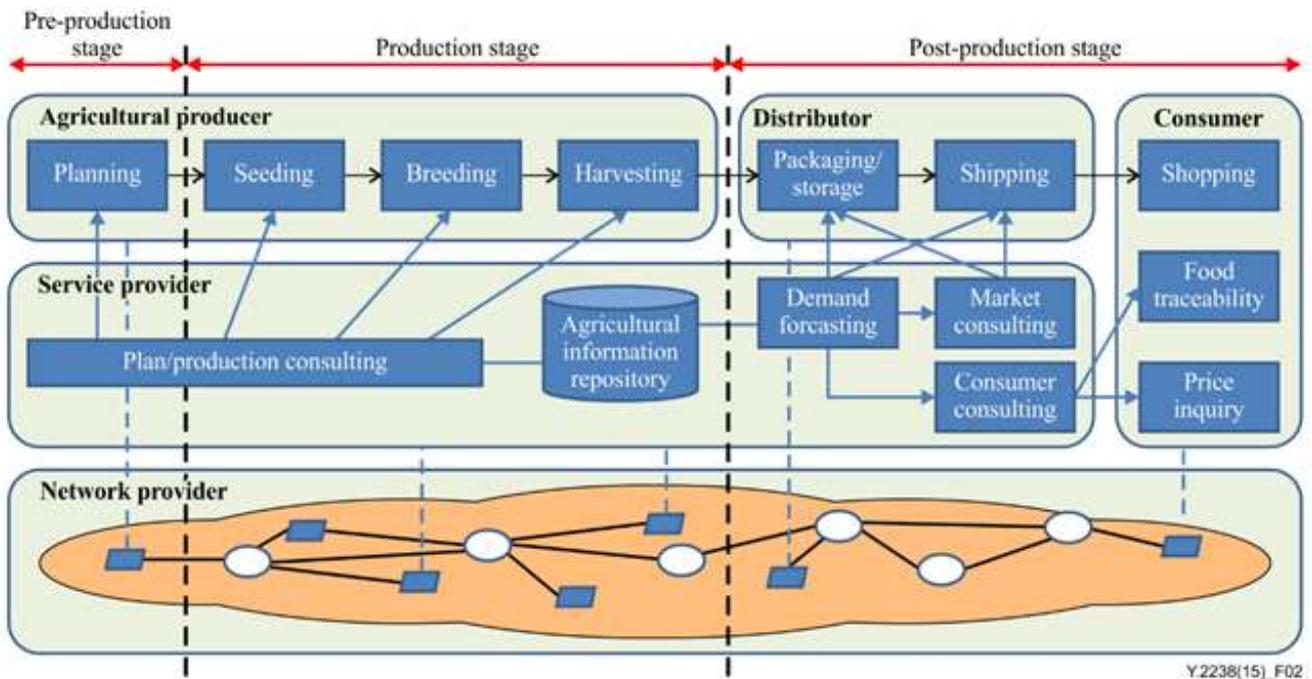


그림 2 - 네트워크 기반 스마트 파밍 참조 모델

- 소비자: 유통사나 농업인으로부터 농작물을 구매하며, 서비스 사업자에게 거래 이력 정보를 제공할 수 있다.
- 농업인: 농작물을 생산하여 유통사나 소비자에게 공급하며, 농작물 보호 서비스나 원격 스마트팜 관리 서비스 등 스마트 파밍 서비스의 이용자이다.
- 유통사: 농업인이 공급한 농작물을 유통한다.
- 서비스 사업자: 스마트 파밍 서비스를 제공한다.
- 네트워크 사업자: 스마트 파밍 관련 정보를 전송하기 위한 인프라를 제공한다.

상기 참조 모델을 기반으로 스마트 파밍을 제공하기 위해서 필요한 기능들을 서비스 관점과 네트워크 관점으로 나누어서 고려할 필요가 있다. 서비스 관점의 기능은 참조 모델에서 정의하고 있는 단계에 따라서 정의된다. 계획단계에서 농업인은 서비스 사업자의 계획 또는 재배 컨설팅을 기반으로 농작물 재배에 대한 계획을 수립한다. 생산 단계에서 농업인은 서비스 사업자의 재배 컨설팅을 기반으로 농작물을 재배 및 수확한다. 농업 정보 저장소를 통해서 상기 서비스 사업자의 컨설팅을 제공할 수 있다. 유통 및 마케팅 단계에서는 유통사가 농작물을 포장 및 저장하고 서비스 사업자의 시장 컨설팅에 기반하여 유통한다. 이후 소비자가 농작물을 구입한다.

네트워크 관점의 기능들을 살펴보면, 우선 다양한 장비간 연결 기능이 필수적이다. 연결 기능은 사람간, 사물간, 사람-사물간 이루어질 수 있다. 특히, 네이밍이나 주소지정 기능은 연결 기능에 필수적이다. 오픈 웹기반 서비스 환경 기능도 필요하다. 이는 웹을 기반으로 다양한 서비스들이 제공되어 집에 따라서 API를 통한 확장 및 연결 기능이 필요함을 의미한다. API를 통하여 다양한 서비스들이 매쉬업될 수도 있을 것이다.

상황인지 기능은 사물의 상태 변화를 인지함을 의미하며, 사용자나 환경을 인식하는 지능형 시스템을 위하여 필요한 기능이다. 다중 네트워크 기능은 하나의 네트워크만 지원하는 것이 아닌 다양한 종류의 네트워크를 지원함을 의미한다. 예를 들어서 다중 네트워크를 통한 다중 경로로 고용량의 트래픽이 나뉘어서 여러 수신자에게 전송될 수도 있다. 이는 서비스의 품질(QoS)과 결함 감내성(Fault tolerance)가 높아짐을 의미한다.

## 나-6) 스마트 온실을 위한 원격 감시용 스마트 영상 장치 (1종) ('19 제안, '20 개정)

최근 스마트팜의 보안과 작물 생육 환경 모니터링을 위해 스마트팜 내부 혹은 외부 영상 감시 장치에 대한 요구가 증가되고 있다. 특히 영상 감시 장치는 온실뿐만 아니라 축사에서도 매우 중요한 장치로 인식되는데, 이는 향후 스마트팜 분야에 AI를 접목함에 있어 영상장치가 매우 중요한 수단이기 때문이다. 이와 관련하여 지난 2016년 농진청에서는 TTA PG426을 통하여 “스마트온실을 위한 원격 감시용 스마트영상장치 (TTAK.KO-10.0945)”를 표준화한 바 있다. 하지만 그 내용은 영상 감시 장치에 대한 화소(pixel), CODEC, 프레임 등에 대한 설명 위주로 되어 있기 때문에, 스마트팜에 영상 감시 장치를 설치하고자 하는 사용자들에 스마트팜의 규모와 용도 등에 따라 어떠한 규격의 영상 장치를 선택해야 하는지에 대한 가이드를 제공하기에는 역부족이었다.

스마트팜 영상 감시 장치를 설치하고자 하는 사용자는 보안 관점에서 대상(피사체)과의 거리에 따른 최적 해상도(화소), 초점거리 등에 대한 인식이 필요하며, 작물의 생육 상태 파악 관점에서는 대상 작물의 크기와 depth, 명암 등에 대한 구체적인 지침이 필요하다. 이에 기 개발된 표준을 기반으로 하여 화소, 프레임, 조도 및 화각 등의 영상 장치 카메라의 규격과 전송 방식(압축규격 및 형식 등), 저장방법(방식, 기간, 장치 등) 및 케이블과 전원설비 등 스마트 영상 감시 장치의 설치 고려사항을 추가할 수 있도록 개정안 (TTAK.KO-10.0945/R1)을 제안하였으며 2019년 12월 최종 채택될 예정이다.

### ■ 카메라

스마트 영상 장치를 구성하는 카메라는 대상을 명확하게 식별할 수 있는 성능을 가져야 하며 이를 위해서는 화소 및 조도, 초점거리와 화각 등을 고려하여 제품을 선정해야 한다. 카메라의 화소 수가 많을수록 정밀하고 상세한 화면을 얻을 수 있으므로 피사체 또는 피사 범위의 크기나 원하는 정밀도에 따라 적절하게 선택해야 한다. 스마트 온실에서의 현장 원격 감시용 카메라의 경우 130만 화소 이상, 작물의 생육 정보 관측용의 경우에는 400만 화소 이상의 제품을 사용한다. 또한 카메라의 최저 조도는 0.5 lux 이하(메가 픽셀 이상의 화소 수를 갖는 경우에는 1.0 lux 이하)이어야 하며 주간 또는 야간 구분 없이 피사체의 식별이 가능해야 한다. 또한 영상 장치의 카메라는 촬영 음영 지역이 발생하지 않도록 적절한 위치에 설치되어야 하며, 이를 위해서는 원하는 피사체 또는 피사 범위를 모두 포함할 수 있는 카메라 렌즈의 초점 거리에 따른 화각 특성을 고려하여 적합한 제품을 선정해야 한다. 초점 거리가 짧으면 피사의 범위가 넓어지고 반대로 긴 초점 거리에서는 피사 범위가 좁아지므로 스마트 온실의 규모와 감시하고자 하는 범위(피사 범위) 또는 생육 상태 측정 대상 작물(피사체)까지의 거리, 높이 및 이미지 픽업(pickup) 장치의 크기 등을 고려할 필요가 있다. 식 1은 일반적인 초점 거리 산출 공식을 보인다.

$$\text{초점 거리}(f) = \frac{\text{피사체와의 거리}(L) \times \text{이미지 픽업 장치의 크기}(V)}{\text{피사체의 높이}(X)} \quad (\text{식 1})$$

<출처 : TTAK.KO-04.0152>

특정 작물의 생육 상태 등을 지속적으로 관측하고자 하는 경우에는 즉, 피사 범위가 좁은 경우에는 초점 거리가 길어야 하며 스마트 온실 내부 또는 외부의 전반적인 감시를 위해 넓은 피사 범위가 요구되는 경우에는 초점 거리가 짧아야 한다. 요구되는 피사 범위가 상당한 규모인 경우에는 복수의 영상 장치를 설치하거나 PTZ 기능을 갖는 영상 장치를 설치하여 음영 지역이 발생하지 않도록 해야 한다.

### ■ 하우징

하우징(housing)은 눈이나 직사광선, 온도 등의 외부 환경 영향으로부터 영상 장치의 카메라를 보호하기 위한 함체로서, 현장 여건에 따라 부식되지 않는 재질 특성을 가져야 하며 카메라와 렌즈 등의 조립에 방해가 되어서는 안된다. 특히 환경적인 영향 요인이 많은 온실 외부에 영상 장치가 설치되는 경우에는 팬이나 히터 등의 자동 온도 조절 장치가 내장된 영상 처리 장치를 사용하는 것이 바람직하며 눈이나 비, 먼지 등으로 인하여 영상 품질이 저하되는 것을 방지하기 위한 와이퍼, 세정 장치, 펌프 등의 부가 기능을 탑재할 수도 있다.

### ■ 압축 형식 및 프레임

압축 형식은 전송 방식에 따라 MPEG-4 AVC(H.264), MJPEG 또는 MPEG-H(H.265) 등을 효율적으로 지원할 수 있어야 하며, 특히 MPEG-4 AVC(H.264) 방식은 필수적으로 지원이 가능해야 한다. 압축 형식

에 따라 원격 전송을 위한 영상 데이터의 해상도(resolution)는 다음의 규격(사양)중 하나 이상을 선택하여 사용할 수 있다.

- 해상도 352 × 240
- 해상도 720 × 240
- 해상도 720 × 480

또한 압축 형식에 따라 원격 전송을 위한 영상 데이터의 초당 프레임 수(fps; frames per second, Fps)는 5, 10, 15, 30 중 하나 이상을 선택하여 사용할 수 있으며, DVR 또는 NVR 저장을 위한 프레임은 1280 × 720 @ 16Fps를 사용해야 한다.

### ■ 저장장치

스마트 온실을 위한 영상 장치는 DVR 또는 NVR과 같은 영상 데이터의 녹화, 저장 및 재생 장치가 필수적으로 포함되어야 하며, 저장된 영상 데이터의 보존 기간은 최소 30일 이상이 요구되나 이는 농가와의 협의를 통해 적절히 조정할 수 있다. 영상 데이터는 스마트 온실의 효율적인 운용을 위해 사전에 정의된 이벤트가 발생하는 즉시 저장되어야 하며, 이 외의 on-demand에 따른 이벤트의 경우에는 선택적으로 저장 여부를 결정할 수 있어야 한다.

영상 저장 장치는 설치 목적에 부합하는 화질을 지속적으로 유지할 수 있어야 하며, 경우에 따라 원격 전송을 위한 IP 포트, 외부 감지기 또는 센서 모듈과의 접속 및 연동, 영상 장치의 제어 기능 등을 포함하는 것이 바람직하다. 일반적으로 영상 데이터의 저장을 위한 최소 요구 용량은 저장 기간, 화소 수 등의 조건에 따라 산정할 수 있다. 식 2의 예시와 같이 저장되는 이미지의 크기(SSI), 영상 장치의 수(NCCTV), 초당 프레임 수(Fps), 일당 운영 시간(SH)에 따른 저장 가능 기간(CDVR)을 산출할 수 있으며, 이는 1일 24시간의 운용 시간을 고려한 것이다. 다음 표는 24시간 운용되는 영상 장치의 화소, 해상도, 운영 대수에 따라 30일의 보존 기간을 고려할 때 요구되는 저장 장치의 최소 용량의 예시를 보인다.

$$C_{DVR} = \frac{S_{HD}}{S_{SI} \times N_{CCTV} \times F_{ps} \times S_H \times 3,600} \quad (\text{식 2})$$

표 1 - 화소별, 영상 장치의 운영 대수별 저장장치의 최소 용량(예시)

구분 (화소)	해상도		이미지 크기 (Kbyte)	영상 장치 운영대수(대)별 최소 요구 용량(GByte)				
	H	V		1	5	10	15	20
300만	2,048	1,536	88	6,834.6	34,173	68,346	102,520	136,693
200만	1,920	1,080	58	4,505.2	22,526	45,053	67,579	90,105
190만	1,600	1,200	54	4,171.6	20,858	41,715	62,573	83,431
130만	1,280	1,024	37	2,847.8	14,239	28,478	42,717	56,955
120만	1,280	960	34	2,669.8	13,349	26,698	40,047	53,396
92만	1,280	720	26	2,002.4	10,012	20,023	30,035	40,047
79만	1,024	768	22	1,708.6	8,543	17,087	25,630	34,173
48만	800	600	13	1,042.8	5,214	10,429	15,643	20,858
36만	800	450	10	782.2	3,911	7,822	11,732	15,643

### ■ 설치 고려사항

스마트 온실 내·외부에 영상 장치를 설치할 때에는 각종 센서 장치, 감지기, 구동기 등의 동작에 지장을 주지 않도록 해야 하며, 시설 출입자나 종사자 등의 이동 및 업무에 방해가 되지 않아야 한다. 또한 영상 장치를 구성하는 설비간 접속을 위한 통신 케이블(꼬임케이블, 동축케이블 또는 광섬유케이블)이나 전력선 등은 관수나 급액 등으로 인한 습기의 영향이나 피복이 벗겨지는 등의 케이블 손상을 고려하여 습한 곳을 피해 배관 등을 이용한 포설이 이루어져야 하며, 낙뢰 시 유도 전류 등에 의한 피해를 최소화하기 위한 보호기, 접지 시설을 갖추어야 한다.

스마트 온실 내·외부에 대한 영상 정보는 손상이나, 파괴, 해킹, 바이러스 감염 등에 노출되지 않도록 조치하여야 한다. 또한 인터넷 회선 등을 통해 영상 정보가 외부로 전송되는 경우에는 해킹 등을 방지하기

위하여 비밀번호의 설정 또는 변경 기능을 가지고 있어야 한다.

스마트 온실 종사자의 사생활 침해 등을 방지하기 위하여 필요한 경우 온실 내·외부에 영상 장치가 설치되었음을 알리고 영상 장치의 설치 목적과 장소, 촬영 범위 및 시간, 관리책임자 연락처 등을 안내하는 것이 바람직하다. 또한 설치된 영상 장치는 접근이 허용된 관계자 외에는 인위적인 조작이 불가능하도록 조치해야 한다. 이 외에 개인 정보의 보호 조치를 위해 TTAK.KO-12.0203\_ed15 또는 TTAK.KO-12.0216\_ed15를 참고한다.

설치된 영상 장치는 온도나 습도, 먼지 등의 외부 환경적인 요인으로부터 안정적인 운영을 위해 카메라, 모니터, 하우징, DVR/NVR 및 케이블 등의 구성 설비의 동작 상태나 접속 상태, 영상 정보의 원활한 전송 상태뿐만 아니라 누수나 발열, 노후화 등에 대하여 주기적으로 점검해야 한다. 스마트 온실은 작물 재배를 위한 온도, 습도 등의 인위적인 조절이 필요한 곳으로서, 여기에 설치되는 영상 장치는 이러한 환경 조건이 운용상의 신뢰성과 효율성을 저하시키지 않도록 관리되어야 한다.

나-7) 스마트 온실 센서구동기 및 제어기 간 RS485 기반 MODBUS 인터페이스(1종) ('17 제안, '18 제정)

스마트 온실에서는 다양한 센서들을 통해 온실 내부의 환경 정보를 취득한다. 측정된 환경 정보는 해당 온실의 작물을 생육하기 위한 환경 조건을 조절하거나, 최적의 생육 알고리즘을 도출하기 위한 기반 정보로도 사용될 수 있다. 이를 위하여 각 온실의 센서 노드 및 구동기 노드, 온실 통합 제어기 간 표준화된 방식의 프로토콜이 사용되어야 서로 다른 장치 간 상호 연동이 보장된다.

이 표준이 스마트 온실에서 적용되는 범위는 그림 1과 같다.

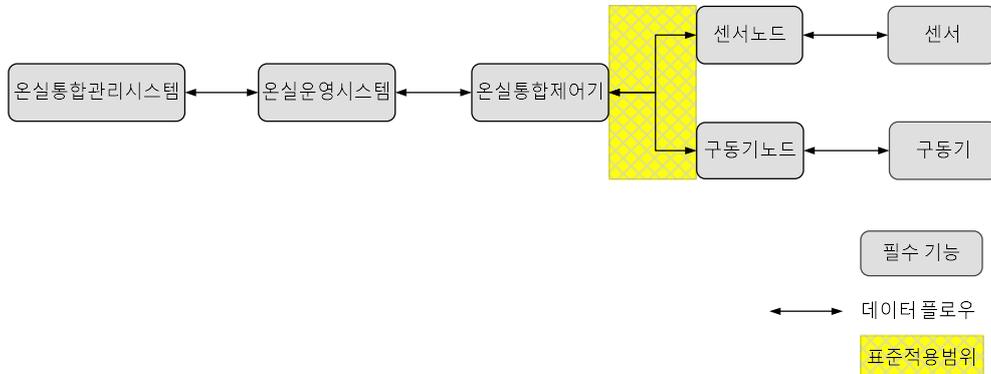


그림 1 - 센서/구동기 노드와 온실 통합 제어기 간 RS485 MODBUS 인터페이스 범위

스마트 온실에서 센서 노드와 구동기 노드가 독립적으로 존재하지 않고, 통합 노드로 존재하는 경우 이 표준의 적용 범위는 그림 2와 같다.

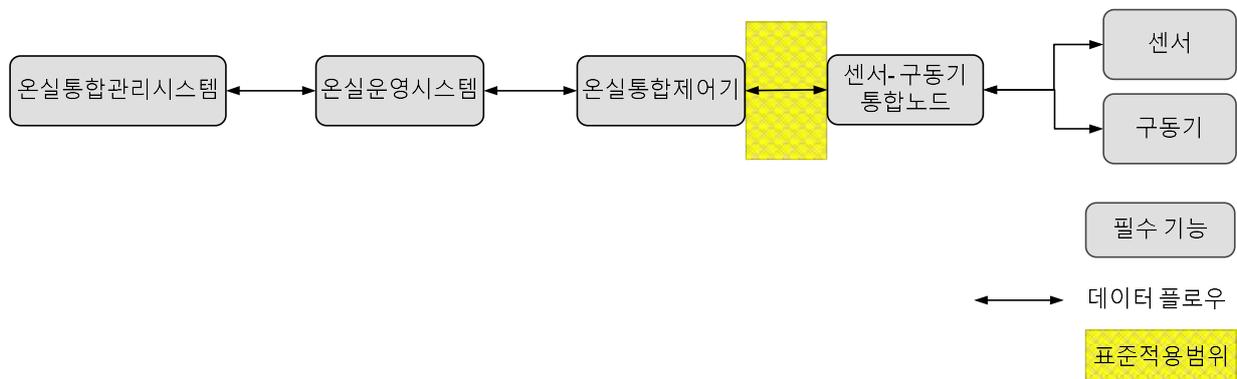


그림 2 - 센서-구동기 통합 노드와 온실 통합 제어기 간 RS485 MODBUS 인터페이스 범위

이 표준에서는 센서 노드/구동기 노드/센서-구동기 통합 노드와 온실 통합 제어기 간에 제공되는 기능과 각 기능을 위해 상호 교환되는 메시지 인터페이스, 전달되는 데이터에 대하여 기술한다. 이 표준은 스마트 온실의 센서 노드와 온실 통합 제어기 간, 구동기 노드와 온실 통합 제어기 간 RS485 통신을 하는 경우, MODBUS(MODBUS) 방식의 인터페이스를 제안함으로써 상호 연동을 제공하는 것을 목적으로 한다.

■ 센서노드, 구동기노드 및 온실통합제어기를 위한 주소 할당 (ID)

- RS485 MODBUS RTU 방식을 사용하는 장치들을 식별하기 위해서는 1바이트의 주소를 사용한다.
- RS485 MODBUS RTU 버스에 접속한 장치마다 고유한 ID 를 부여해야 한다.
- 장치마다 ID를 부여하는 방법은 표준에서 정의하지 않는다.

표 1 - ID와 RS485 MODBUS 주소 매핑

제어기, 노드 및 디바이스 ID	RS 485 MODBUS 주소
온실 통합 제어기의 ID	마스터 주소(master address)

센서 노드, 구동기 노드, 센서-구동기 통합 노드 등의 노드 ID	슬레이브 주소(slave address)
센서, 구동기 등의 디바이스 ID	레지스터 주소(register address)

■ RS485 MODBUS RTU 시리얼 통신 프로토콜 개요 및 적용 방안

MODBUS application protocol에 정의된 MODBUS 표준은 다양한 타입의 버스나 네트워크 상에 연결된 장치 사이에 클라이언트 서버 간 통신을 제공하는 OSI(Open Systems Interconnection) 모델의 7 계층에 해당하는 응용 계층의 메시징 프로토콜이다.

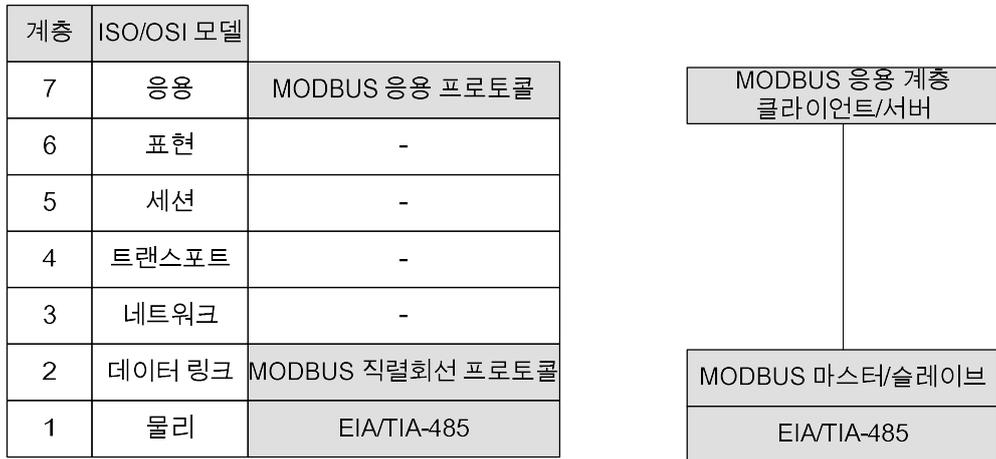


그림 3 - MODBUS 프로토콜 계층

MODBUS 응용 프로토콜은 하부의 통신 계층에 독립적인 MODBUS PDU를 정의한다. MODBUS PDU는 기능 코드(function code)와 데이터로 구성된다.

- 기능 코드: 서버가 어떤 종류의 액션을 수행해야 하는지를 표시한다. 코드 값은 1-255가 가능하나, 코드 값 128-255는 예외 응답을 위해 사용되도록 예약되어 있다. 0은 사용하지 않는다.
- 데이터: 요청이나 응답 메시지 파라미터를 기술한다. 서버가 기능 코드에 정의된 동작을 수행하기 위해 사용하는 부가적인 정보들이다. 레지스터 주소, 다루어야 할 항목의 양, 실제 데이터 바이트의 수 등을 포함할 수 있다.

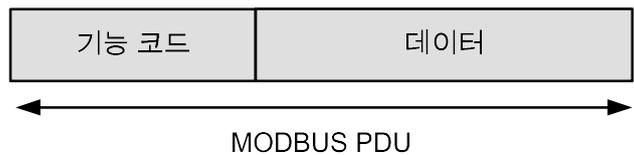


그림 4 - MODBUS PDU

데이터 인코딩 방법은 빅엔디언(big-endian) 표현법을 사용한다. 주소가 데이터 항목의 수치 값이 1바이트 이상이 전송될 때, 가장 의미 있는 바이트를 우선 전송한다. 예를 들어, 레지스터 크기가 16비트이고 그 값이 '0x1234'인 경우, 0x12부터 먼저 보내고, 0x34는 다음에 전송한다.

MODBUS over serial line에 정의된 MODBUS 직렬 회선(serial line) 프로토콜은 마스터와 하나 이상의 슬레이브 사이에 MODBUS 요청 메시지를 교환하기 위한 프로토콜이다. 하부의 물리적인 인터페이스로는 RS485 2선 인터페이스가 사용된다. MODBUS 직렬 회선 프로토콜은 마스터-슬레이브 방식을 사용한다. 이는 MODBUS 상에 오직 하나의 마스터 노드가 존재하고, 최대 247개의 슬레이브 노드가 연결되어 있다. 마스터 노드가 슬레이브 노드 중 하나에게 요청을 하고 이에 대한 응답을 처리한다. 슬레이브 노드는 마스터 노드에서의 요청 없이 데이터를 전송할 수 없고, 다른 슬레이브 노드와 통신할 수도 없다. 마스터 노드는 동시에 하나의 트랜잭션만을 발생시킨다.

MODBUS 직렬 회선 프로토콜은 유니캐스트 모드와 브로드캐스트 모드가 있으며, 이 표준에서는 유니캐스트 모드를 사용한다.

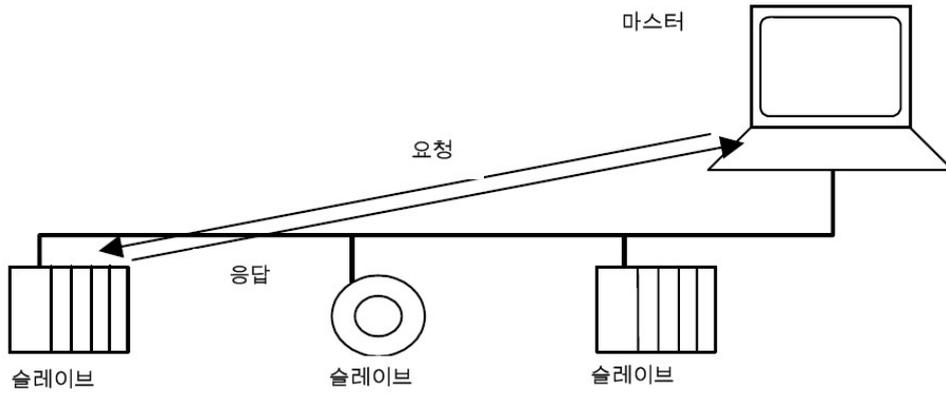


그림 5 - 마스터-슬레이브(master-slave) 방식

마스터 노드는 특정 슬레이브 노드를 지정하여 요청 메시지를 보내고, 해당 슬레이브 노드는 요청 메시지를 받아 처리한 후 이에 대한 응답 메시지를 보낸다. 즉, MODBUS 트랜잭션은 요청과 응답의 두 가지 메시지로 구성된다. 개별 슬레이브 노드 주소는 1에서 247이 가능하다. 주소 0은 모든 슬레이브 노드에 요청 메시지를 보내는 브로드캐스트 주소로 이 표준에서는 사용하지 않는다.

직렬 회선상에서 메시지 전송 모드는 원격 단말 장치(RTU) 모드와 아스키(ASCII) 모드가 있으며, 이 표준에서는 RTU 모드를 사용한다. 각 장치가 RTU 모드를 사용하여 MODBUS 직렬 회선으로 통신할 때 데이터 형식은 4비트 16진수 데이터 포맷을 사용한다. 이는 문자 밀집도를 제공함으로써 ASCII 모드보다 더 좋은 처리량(throughput)을 제공한다.

MODBUS PDU를 시리얼 라인으로 전달하기 위해서는 통신 PDU로 MODBUS 직렬 회선 PDU를 생성한다. MODBUS 직렬 회선 PDU는 MODBUS PDU의 앞에 주소 필드(address field)를 추가하고, MODBUS PDU 뒤에 순환 중복 검사(CRC) 필드를 추가함으로써 생성된다.

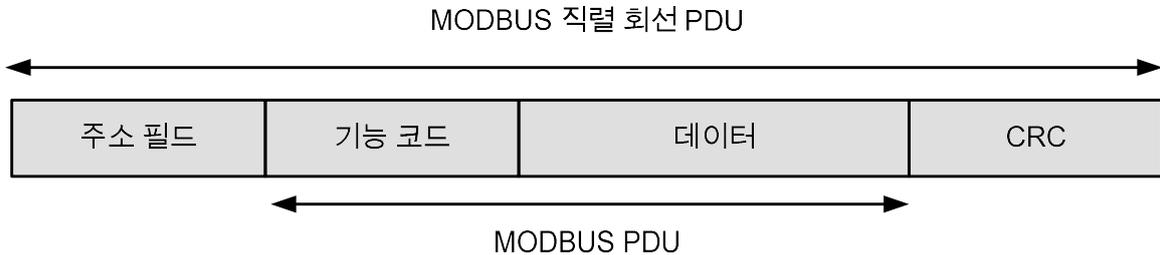


그림 6 - MODBUS 메시지 프레임과 MODBUS PDU와의 관계

MODBUS 직렬 회선 상에서 주소 필드는 슬레이브 주소만을 포함한다. 따라서 직렬 회선 상의 MODBUS RTU 메시지 프레임 구성은 다음과 같이 구성된다. MODBUS RTU 프레임의 최대 크기는 256바이트이다.

주소 필드	기능 코드	데이터	CRC
1바이트	1바이트	0~252 바이트	2바이트

그림 7 - MODBUS 메시지 프레임 구성

MODBUS 메시지는 전송 장치가 직렬 회선상에 프레임 형태로 놓임으로써 전달된다. 전체 메시지 프레임은 문자의 연속적인 스트림으로 전송되어야 하며, MODBUS 메시지 프레임의 전송 직전이나 직후에 3.5 문자(character) 이상의 시간 공백을 유지하도록 한다. 만약 두 문자 사이에 1.5 문자 이상의 공백 기간(silent interval)이 존재하면, 그 메시지 프레임은 완전하지 않은 것으로 간주하여 수신자가 버리도록 한다.

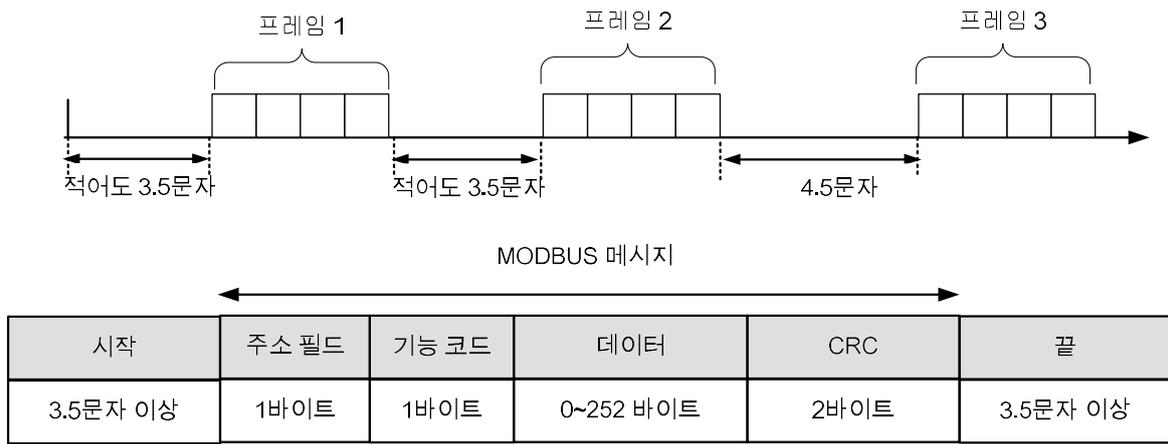


그림 8 - MODBUS 메시지 전송

■ 메시지 플로우

데이터 교환을 위한 메시지 플로우는 다음과 같다.

- 정상적인 경우: 요청(request) → 응답(response)

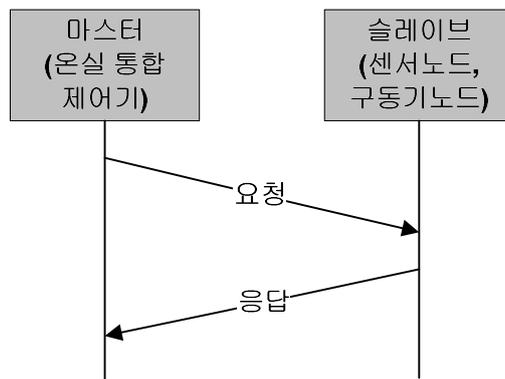


그림 9 - 정상적인 경우의 메시지 플로우

- 예외 응답의 경우: 요청(request) → 예외 응답(exception response)

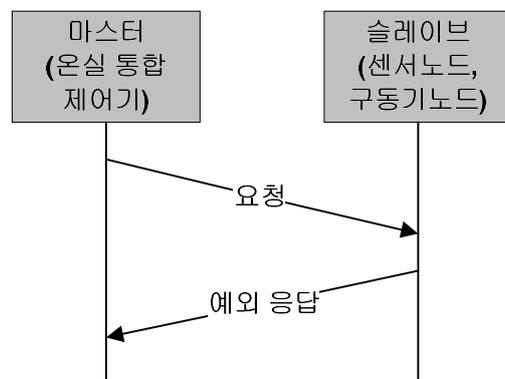


그림 10 - 예외 응답인 경우의 메시지 플로우

- 패킷 에러의 경우: 요청(request) → 시간 경과(timeout)

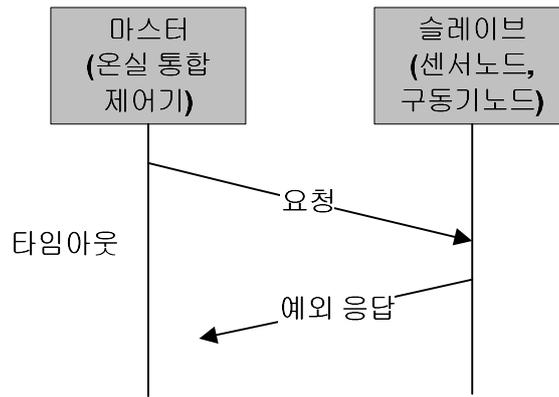


그림 11 - 패킷 에러인 경우의 메시지 플로우

■ 패킷 구조

a) 요청(request) 패킷

표 2 - 요청 패킷의 구조

슬레이브 주소	기능 코드	데이터	CRC
1바이트	1바이트	0-252바이트	2바이트

- 슬레이브 주소: 센서 노드나 구동기 노드의 주소가 해당됨. 1-247이 가능하며, 0은 브로드캐스트 주소로 되어 있으나, 이 표준에서는 브로드캐스트 주소를 사용하지 않음
- 기능 코드: 지정된 슬레이브가 어떤 동작을 수행해야 할지를 알려 주는 값이며, 마스터에서 슬레이브로 요청하는 요청값(request code)
- 데이터: 슬레이브 노드가 기능 코드에 정의된 동작을 수행하기 위해 사용하는 부가적인 정보들이다. 레지스터 주소, 다루어야 할 항목의 양, 실제 데이터 바이트의 수 등을 포함할 수 있으며, 기능 코드별로 구체적인 필드가 정해짐
- CRC: CRC-16 알고리즘 오류 체크, 수신 장치는 에러 체크를 위한 CRC 필드를 제외한 나머지 필드값으로 CRC 계산

b) 정상적인 응답(normal response) 패킷

표 3 - 정상 응답 패킷의 구조

슬레이브 주소	기능 코드	데이터	CRC
1바이트	1바이트	0-252바이트	2바이트

- 슬레이브 주소: 응답하는 센서 노드나 구동기 노드의 주소
- 기능 코드: 마스터로부터 수신한 기능 코드 값 복사
- 데이터: 슬레이브 노드가 기능 코드에 정의된 동작을 수행한 후 응답하는 데 사용하는 부가적인 정보들이다. 레지스터 주소, 실 데이터 바이트의 수 등을 포함할 수 있으며, 기능 코드별로 구체적인 필드가 정해짐
- CRC: 에러 체크 값을 표현하는 16비트 수

c) 예외 응답(exception response) 패킷

표 4 - 예외 응답 패킷의 구조

슬레이브 주소	기능 코드	에러 코드	CRC
1바이트	1바이트	N바이트	2바이트

- 슬레이브 주소: 응답하는 센서 노드나 구동기 노드의 주소
- 기능 코드: 마스터에서 수신한 기능 코드 값에 0x80 추가하여 수정
  - 마스터에서 수신한 기능 코드: 0x03 → 예외 응답의 기능 코드: 0x83

- 마스터에서 수신한 기능 코드: 0x06 → 예외 응답의 기능 코드: 0x86
- 에러 코드: 예외 발생 내역에 대한 상세 코드 값의 예
  - 0x01(illegal function): 제품에서 지원되지 않는 기능 코드
  - 0x02(illegal data address): 유효하지 않은 주소를 접근하고자 할 때
  - 0x03(illegal data value): 지원되지 않는 데이터 값으로 지정하고자 할 때
  - 0x04(slave device failure): 디바이스 문제 발생
- CRC: 에러 체크 값을 표현하는 16비트 수

■ 온실 통합 제어기와 센서 노드, 구동기 노드, 통합 노드 간 제공 기능

이 절에서는 온실 통합 제어기와 센서 노드/구동기 노드/센서-구동기 통합 노드 간에 제공되는 기능을 기술한다. 크게 센싱 정보 획득, 구동기 상태 정보 획득 등의 데이터 확보 기능(GetData)과 구동기 제어 명령 등의 데이터 지정 기능(SetData)으로 분류된다. 이를 위하여 온실 통합 제어기는 센서 노드/구동기 노드/센서-구동기 통합 노드 내에 각 센서나 구동기에 대한 정보를 알고 있다고 가정한다.

이 절에서는 위의 기능들이 MODBUS 프로토콜을 이용하여 어떻게 표현되는지를 기술한다.

a) 데이터 확보 기능(GetData)

온실 통합 제어기는 마스터로 동작하고, 센서 노드 또는 구동기 노드는 슬레이브로 동작한다. 온실 통합 제어기(마스터)는 데이터를 읽어 올 레지스터 주소와 읽어 올 해당 레지스터 수를 포함한 기능 코드 0x03(Read Holding Register) 요청 메시지를 센서 노드 또는 구동기 노드(슬레이브)에게 보낸다. 센서 노드 또는 구동기 노드(슬레이브)는 온실 통합 제어기(마스터)로, 해당 레지스터 주소의 값과 바이트 수를 포함한 응답 메시지를 온실 통합 제어기(마스터)에 전송한다.

이때 온실 통합 제어기는 해당 센서 노드에 연결된 각 센서의 레지스터 주소와 데이터 값의 크기를 사전에 알고 있어야 한다. 마찬가지로 온실 통합 제어기는 해당 구동기 노드에 연결된 각 구동기의 레지스터 주소와 데이터 값의 크기를 사전에 알고 있어야 한다.

MODBUS 프로토콜에서 레지스터 데이터 값은 2바이트를 가정하고 있다. 따라서 센서나 구동기의 데이터 값이 2바이트를 초과하는 경우, 해당 레지스터의 시작 주소부터 연속해서 해당 데이터 값을 포함하는 크기를 요청하도록 한다. 예를 들어, 만약 온도 센서 A의 값이 4바이트로 표현이 되는 경우, MODBUS 메시지는 온도 센서 A로 매핑된 레지스터 주소와 레지스터의 수로 2를 포함한 0x03(Read Holding Registers) 요청 메시지를 전송한다.

b) 요청 패킷

요청 메시지의 기능 코드로 0x03을 사용하며, 시작 주소(start address)로 시작하는 레지스터부터 시작하여 레지스터 수(quantity of registers)만큼의 레지스터 값을 읽을 것을 요청한다.

표 5 - 요청 패킷의 구조

기능 코드	시작 주소	레지스터 수
1바이트	1바이트	2바이트

- 기능 코드: 0x03(Read Holding Registers) 요청
- 시작 주소: 데이터를 읽어 올 대상 레지스터의 시작 주소, 각 센서나 구동기에 매핑되는 레지스터 시작 주소
- 레지스터 수: 데이터를 읽어 올 대상 레지스터의 수, 대상 레지스터에 표현된 데이터 값의 크기

c) 응답 패킷

기능 코드 0x03의 레지스터 읽기 요청에 대하여 해당 레지스터로부터 읽어 온 데이터 값의 바이트 수와 레지스터 데이터 값을 포함하여 응답한다.

표 6 - 응답 패킷의 구조

기능 코드	바이트 수	레지스터 값
1바이트	1바이트	2*N바이트(N: quantity of registers)

- 기능 코드: 0x03(Read Holding Registers) 요청에 대한 응답
- 바이트 수: 레지스터 값들을 구성하는 부분의 길이(2\*N), 데이터를 표현하는 완전한 바이트의 수
- 레지스터 값: 각 레지스터 값으로, 각 레지스터별 2바이트를 가정한다.

■ 레지스터 데이터 포맷

응답 패킷의 레지스터 값에 포함되는 데이터 포맷은 다음과 같다.

a) 센서 데이터 포맷

스마트 온실의 각 센서에서 센싱한 데이터 값에 대한 데이터 포맷은 TTAK.KO-10.1046에서 정의한 내용을 따르며, 이 표준에서는 이를 다음과 같이 적용한다.

- 데이터 전송 대역폭이 작은 관계로 바이너리 인코딩 방식을 사용한다.
- 각 센서에서 센싱 값(sensing value)만 전달하도록 한다.

센서에서 센싱하는 값과 관련된 속성 정보는 표 7과 같다.

표 7 - 센서의 센싱 정보 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
상태 정보	센싱 값	M	각 센서 타입별 센싱 값

표 7에서 응답 패킷의 레지스터 값에 포함되는 내용은 센싱 값만 포함하며, 센싱 값에 대한 바이너리 인코딩 포맷은 표 8과 같다.

표 8 - 센서 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
센싱 값	4	32	값 타입에 따라 다름	디바이스 타입 및 분류에 해당하는 디바이스 센싱 값 - 센싱 값이 실수(float)로 표현되는 경우, 4바이트로 표현되며 IEEE-754 표준을 따른다.	-

센서는 센서의 타입에 따라 데이터 측정 범위가 다르다. 표 9는 TTAK.KO-10.0903에 따른 각 센서 종류별 데이터 범위, 데이터 타입을 제공한다.

표 9 - 센서 종류별 데이터 범위 및 데이터 타입

번호	분류	데이터 측정 범위	데이터 단위	타입
1	온도 센서	-20 ~ 80	°C	float
2	습도 센서	0 ~ 100	%	float
3	CO <sub>2</sub> 센서	0 ~ 3 000	μmol/mol(PPM)	float
4	일사 센서	0 ~ 2 000	W/m <sup>2</sup>	float
5	풍향 센서	0 ~ 360	°(방향각)	float
6	풍속 센서	0 ~ 40	m/s	float
7	감우 센서	On/OFF	-	integer
8	광양자 센서	0 ~ 2 000	μmol/m <sup>2</sup> /s	float
9	토양 함수율 센서	0 ~ 50	% vol	float
10	토양 수분 장력 센서	0 ~ 100	kPa	float
11	EC 센서	0 ~ 10	dS/m	float
12	PH 센서	2 ~ 12	pH	float
13	지온 센서	-20 ~ 80	°C	float

b) 구동기 데이터 포맷

스마트 온실에서 각 구동기 상태에 대한 데이터 포맷은 TTAK.KO-10.1045에서 정의한 내용을 따르며, 이 표준에서는 이를 다음과 같이 적용한다.

- 데이터 전송 대역폭이 작은 관계로 바이너리 인코딩 방식을 사용한다.
- 구동기 타입을 스위치형과 개폐형으로 구분하며, 각 타입에 따른 동작 상태와 동작 시간, 동작 상태와 동작 위치 등의 속성 정보를 함께 전달하도록 한다.
- 스위치형 구동기의 동작 기간 속성 정보와 개폐형 구동기의 동작 속도 속성 정보는 적용하지 않는다.

구동기는 타입에 따라 상태 정보가 다르다. 표 10은 TTAK.KO-10.0845에 따른 각 구동기 종류별 타입 및 가능한 상태 정보를 제공한다.

표 10 - 구동기 종류별 타입 구분 및 가능한 상태 정보

번호	분류	구동기 타입	상태 정보
1	천창	개폐형	OPENING(여는 중)/CLOSING(닫는 중)/ OPEN(열림)/CLOSED(닫힘)
2	측창	개폐형	OPENING/CLOSING/OPEN/CLOSED
3	보온 덮개	개폐형	OPENING/CLOSING/OPEN/CLOSED
4	차광막	개폐형	OPENING/CLOSING/OPEN/CLOSED
5	환풍기	스위치형	ON(작동)/OFF(정지)
6	유동 팬	스위치형	ON(작동)/OFF(정지)
7	관수 모터	스위치형	ON(작동)/OFF(정지)
8	관수 밸브	스위치형	ON(작동)/OFF(정지)
9	냉난방기	스위치형	ON(작동)/OFF(정지)

스위치형 구동기와 관련된 상태 정보 및 상태 관련 속성 정보는 표 11과 같이 제공한다.

표 11 - 스위치형 구동기의 상태 정보 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
상태 정보	동작 상태 (Operation Status)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	O	동작 유지 기간 날짜와 시간 정보를 'P4Y2M2DT2H2M2S' 형태로 기술
	동작 시간 (Operation Time)	O	동작 유지 시간 시(hour), 분(minute), 초(second) 정보를 기술

응답 패킷의 레지스터 값에 포함되는 내용은 표 11의 구동기 동작 상태와 상태 관련 속성 정보 부분으로, 이 표준에서는 동작 상태와 동작 시간이 적용된다. 바이너리 인코딩 포맷은 표 12와 같다. 이때, RS485 MODBUS는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 상태 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 12 - 스위치형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 상태 (Operation Status)	2	8	부호 없는 정수형 (unsigned integer)	구동기 동작 상태 정보 - 0x00: ON (작동) - 0x01: OFF (정지)	스위치형 구동기는 ON/OFF
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 시간 (Operation Time)	4	24	부호 없는 정수형 (unsigned integer)	동작 시간 - 시(hour), 분(minute), 초(second) 순서 로 1바이트씩 기술	-
		8	-	레지스터를 구성하기 위하여 추가된 비트	-

개폐형 구동기와 관련된 상태 정보 및 상태 관련 속성 정보는 표 13과 같이 제공한다.

표 13 - 개폐형 구동기의 상태 정보 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
상태 정보	동작 상태 (Operation Status)	M	OPENING(여는 중)/CLOSING(닫는 중)/ OPEN(열림)/CLOSED(닫힘)
	동작 위치 (Operation Position)	O	현재 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	O	동작 속도

응답 패킷의 레지스터 값에 포함되는 내용은 표 13의 구동기 동작 상태와 상태 관련 속성정보 부분으로, 이 표준에서는 동작 상태와 동작 위치가 적용된다. 바이너리 인코딩 포맷은 표 14와 같다. 이때, RS485 MODBUS는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 상태 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 14 - 개폐형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 상태 (Operation Status)	2	8	부호 없는 정수형 (unsigned integer)	구동기 상태 정보 - 0x10: OPENING(열리는 중) - 0x11: CLOSING(닫히는 중) - 0x12: OPEN(열림) - 0x13: CLOSED(닫힘)	- 개폐형 구동기는 OPENING/ CLOSING/OPEN/CLOSED
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 위치 (Operation Position)	2	8	부호 없는 정수형 (unsigned integer)	동작 위치 정보, 비율로 표시 - 0 ~ 100	- 개폐형 구동기에 사용 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
		8	-	레지스터를 구성하기 위하여 추가된 비트	-

■ 데이터 지정 기능(SetData) - 단일 레지스터

온실 통합 제어기는 마스터로 동작하고, 구동기 노드는 슬레이브로 동작한다. 온실 통합 제어기(마스터)는 데이터를 쓸 레지스터 주소와 레지스터 값을 포함한 기능 코드(0x06, Write Holding Register) 요청 메시지를 구동기 노드(슬레이브)에 보낸다.

구동기 노드(슬레이브)에서는 온실 통합 제어기(마스터)로 레지스터 주소 정보와 해당 레지스터에 기록한 값을 포함한 응답 메시지를 온실 통합 제어기(마스터)에게 전송한다. 정상적인 응답 메시지는 요청 메시지를 그대로 표현하게 된다. 이때 온실 통합 제어기는 해당 구동기 노드에 연결된 각 구동기의 레지스터 주소와 레지스터에 쓰고자 하는 제어 명령 데이터 값의 크기를 사전에 알고 있어야 한다. MODBUS 프로토콜에서 레지스터 데이터 값은 2바이트를 가정하고 있다. 따라서 구동기의 제어 명령 데이터 값이 2바이트를 초과하지 않는 경우에 사용할 수 있다.

a) 요청 패킷

요청 메시지의 기능 코드로 0x06를 사용하며, 레지스터 주소로 시작하는 레지스터에 레지스터 값을 쓸(write) 것을 요청한다.

표 15 - 요청 패킷의 구조

기능 코드	레지스터 주소	레지스터 값
1바이트	2바이트	2바이트

- 기능 코드: 0x06(Write Holding Register)
- 레지스터 주소: 데이터를 기록할 대상 레지스터 주소, 각 구동기에 매핑되는 레지스터 주소
- 레지스터 값: 데이터를 기록할 대상 레지스터 값, 레지스터 값에 포함되는 데이터 포맷은 6.3.3에서 기술한다.

**b) 응답 패킷**

기능 코드 0x06의 레지스터 기록 요청에 대하여 해당 레지스터의 주소와 해당 레지스터에 기록한 데이터 값을 포함하여 응답한다.

표 16 - 응답 패킷의 구조

기능 코드	레지스터 주소	레지스터 값
1바이트	2바이트	2바이트

- 기능 코드: 0x06(Write Holding Register)
- 레지스터 주소: 데이터를 기록한 대상 레지스터 주소
- 레지스터 값: 데이터를 기록한 대상 레지스터 값

**■ 데이터 지정 기능(SetData) - 복수 레지스터**

온실 통합 제어기는 마스터로 동작하고, 구동기 노드는 슬레이브로 동작한다. 온실 통합 제어기(마스터)는 데이터를 쓸 레지스터의 시작 주소와 레지스터 수, 레지스터 데이터 값의 바이트 수, 레지스터 값을 포함한 기능 코드(0x10, Write Multiple Registers) 요청 메시지를 구동기 노드(슬레이브)에게 보낸다.

구동기 노드(슬레이브)에서는 온실 통합 제어기(마스터)로 레지스터 시작 주소 정보와 레지스터 수에 대한 정보를 포함한 응답 메시지를 온실 통합 제어기(마스터)에게 전송한다. 이때 온실 통합 제어기는 해당 구동기 노드에 연결된 각 구동기의 레지스터 주소와 레지스터에 쓰고자 하는 제어 명령 데이터 값의 크기를 사전에 알고 있어야 한다.

MODBUS 프로토콜에서 레지스터 데이터 값은 2바이트를 가정하고 있다. 따라서 이 기능은 구동기의 제어 명령 데이터 값이 2바이트를 초과하는 경우에 사용할 수 있다. 예를 들어, 만약 천창 A에 대한 제어 명령의 값이 4바이트로 표현이 되는 경우, MODBUS 메시지는 천창 A로 매핑된 레지스터 시작 주소와 레지스터의 수로 2, 제어 명령에 해당하는 레지스터 값의 바이트 수, 제어 명령 레지스터 값을 포함한 0x10(Write Multiple Registers) 요청 메시지를 전송한다.

**a) 요청 패킷**

요청 메시지의 기능 코드로 16(0x10)을 사용하며, 레지스터 시작 주소로 시작하여 레지스터 수에 해당하는 레지스터에 레지스터 값을 바이트 수만큼 쓸(write) 것을 요청한다. MODBUS 프로토콜에서 레지스터는 2바이트를 가정하고 있다. 따라서 특정 디바이스의 데이터 값이 2바이트를 초과하는 경우, 해당 레지스터의 시작 주소부터 연속해서 해당 데이터 값을 포함하는 크기를 기록하도록 한다. 또한, 특정 디바이스들을 연속한 주소로 할당하고, 각 디바이스에 대한 데이터 값의 크기를 아는 경우, 한 명령어로 두 개 이상의 디바이스에 대한 제어 명령을 전송할 수 있다.

표 17 - 요청 패킷의 구조

기능 코드	시작 주소	레지스터 수	바이트 수	레지스터 값
1바이트	2바이트	2바이트	1바이트	(2*N) 바이트 N = 레지스터 수

- 기능 코드: 16(0x10, Write Multiple Registers)
- 시작 주소: 데이터를 기록할 대상 레지스터의 시작 주소
- 레지스터 수: 대상 레지스터의 수(N)
- 바이트 수: 레지스터 데이터 값의 바이트 수
- 레지스터 값: 데이터를 기록할 대상 레지스터 데이터 값

b) 응답 패킷

기능 코드 16(0x10)의 여러 개의 레지스터에 레지스터 기록 요청에 대하여 해당 레지스터의 시작 주소와 레지스터 수를 포함하여 응답한다.

표 18 - 응답 패킷의 구조

기능 코드	시작 주소	레지스터 수
1바이트	2바이트	2바이트

- 기능 코드: 16(0x10, Write Multiple Registers)
- 시작 주소: 데이터를 기록한 대상 레지스터의 시작 주소
- 레지스터 수: 데이터를 기록한 레지스터의 수

c) 레지스터 데이터 포맷

요청 패킷이나 응답 패킷의 레지스터 값에 포함되는 데이터 포맷은 다음과 같다. 레지스터 값의 크기가 2바이트를 넘지 않는 경우, 데이터 지정 기능을 통하여 전달한다. 스마트 온실에서 각 구동기 제어 명령에 대한 데이터 포맷은 TTA.KO-10.1045에서 정의한 내용을 따르며, 이 표준에서는 다음과 같이 적용한다.

- 데이터 전송 대역폭이 작은 관계로 바이너리 인코딩 방식을 사용한다.
- 구동기 타입을 스위치형과 개폐형으로 구분하며, 각 타입에 따른 제어 명령과 동작 시간, 제어 명령과 동작 위치 등의 속성 정보를 함께 전달하도록 한다.
- 스위치형 구동기의 동작 기간 속성 정보와 개폐형 구동기의 동작 속도 속성 정보는 적용하지 않는다.

스위치형 구동기와 관련된 제어 명령 및 제어 명령 관련 속성 정보는 표 19와 같이 제공한다.

표 19 - 스위치형 구동기의 제어 명령 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
제어 정보	동작 명령 (Operation Command)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	O	동작 유지 기간 - 날짜와 시간 정보를 'P4Y2M2DT2H2M2S' 형태로 기술
	동작 시간 (Operation Time)	O	동작 유지 시간 - 시(hour), 분(minute), 초(second) 정보를 기술

응답 패킷의 레지스터 값에 포함되는 내용은 표 19의 구동기 동작 제어 명령과 제어 관련 속성 정보 부분으로, 이 표준에서는 동작 명령과 동작 시간이 적용된다. 바이너리 인코딩 포맷은 표 20과 같다. 이때, RS485 MODBUS는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 제어 명령 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 20 - 스위치형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 명령 (Operation Command)	2	8	부호 없는 정수형 (unsigned integer)	구동기 제어 명령 - 0x00: ON(작동) - 0x01: OFF(정지)	스위치형 구동기는 ON/OFF
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 시간 (Operation Time)	4	24	부호 없는 정수형 (unsigned integer)	시(hour), 분(minute), 초(second) 순서로 1바이트씩 기술	-
		8	-	레지스터를 구성하기 위하여	-

개폐형 구동기와 관련된 제어 명령 및 제어 명령 관련 속성 정보는 표 21과 같이 제공한다.

표 21 - 개폐형 구동기의 제어 명령 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
제어 정보	동작 명령 (Operation Command)	M	OPEN(열기)/CLOSE(닫기)
	동작 위치 (Operation Position)	0	원하는 동작 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	0	동작 속도

요청 패킷의 레지스터 값에 포함되는 내용은 표 21의 구동기 동작 제어 명령과 제어 관련 속성 정보 부분으로, 이 표준에서는 동작 명령과 동작 위치가 적용된다. 바이너리 인코딩 포맷은 표 20과 같다. 이때, RS485 MODBUS는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 제어 명령 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

나-8) IP 기반 스마트팜(시설/축사/노지)내 통신을 위한 경량형 제어 프로토콜(1종) ('17 제안, '18 제정)

스마트 온실은 기존 온실 기술에 ICT 기술을 접목하여 원격 혹은 자동으로 작물의 생육환경을 적정하게 유지·관리할 수 있는 시설을 의미한다. 사용자는 스마트 온실에서 자라는 작물의 생육 및 환경 정보 데이터를 추적할 수 있으며, 이를 통해 최적의 생육 환경을 조성할 수 있으며, 또한 최소한의 노동력·에너지만으로도 농산물의 생산성과 품질을 향상시킬 수 있다. 하지만 작물과 작목 방식에 따라 스마트 온실의 구성 방법을 달리하기 때문에 스마트 온실에 설치되는 장치, 통신 방식 및 소프트웨어가 제조사별 버전별로 상이하게 된다. 그러므로 개발자는 스마트 온실의 구성 방법에 따라 소프트웨어를 수정해야 할 필요가 있다. 그러므로 본 연구에서는 작물과 작목 방식에 따라 상이하게 구축되는 스마트 온실 환경에 상관없이 관제를 위한 소프트웨어 모듈의 재활용이 가능한 LCP(Lightweight Control Protocol) 프로토콜을 설계하여 국가 단체표준으로 추진하였다(2018년 12월).

LCP는 다음과 같이 특정 기술에 종속적이지 않고 지속/확장 가능성을 목적으로, 시설원예, 노지, 축사, 유통 분야에서 사용되는 다양한 네트워크 인프라(WiFi, ZigBee, RS485, LoRa 등)를 투명화 시킬 수 있는 스마트팜 공통 기본 프로토콜로, 제어응용들은 하부 네트워크 인프라 기술의 API 인터페이스를 고려하지 않고, 스마트팜 공통 데이터 포맷(용어)을 이용해 자신의 제어응용을 만든 후, 공통 스마트팜 기본 프로토콜(통신)에서 제공하는 API를 사용하여 명령을 교환할 수 있다.

환경 관리를 위해 하나 이상의 센서노드와 구동기노드들이 온실에 설치되어 있으며, 제어기는 설치된 센서노드와 구동기노드들을 이용하여 하나의 구역에 대한 환경을 제어한다. 하나의 구역은 제어가 환경 제어를 할 수 있는 영역으로써, 설치 방법에 따라 하나의 온실을 여러 구역으로 쪼개어 구성할 수도 있으며, 반대로 다수의 온실을 하나의 구역으로도 구성할 수 있다. 그렇기 때문에 본 표준에서 다루는 구역은 제어기의 관리 단위가 되는 영역이라 정의한다.



그림 1 - 하부 네트워크에 투명한 스마트팜 공통기본 프로토콜

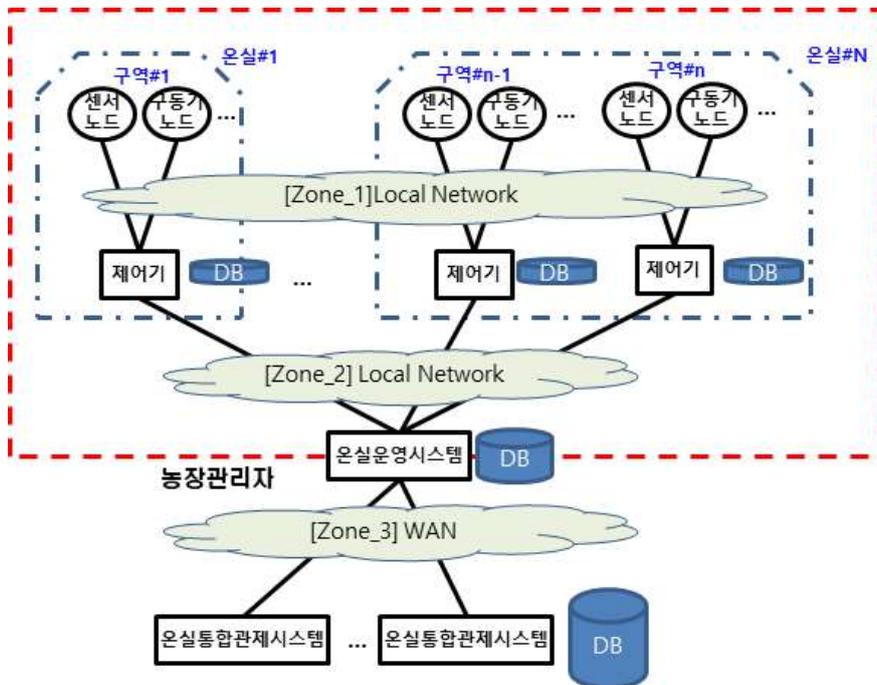


그림 2 - LCP 적용 범위

LCP에서는 스마트 온실의 특성을 고려하여 적용 가능한 네트워크 하부 기술들을 다음과 같이 구분하여 정의한다.

- Zone\_1: 센서/구동기노드와 제어기간 통신을 위한 네트워크
- Zone\_2: 온실 내 제어기와 온실 운영 시스템 간 통신을 위한 네트워크
- Zone\_3: 외부 빅데이터 서버로의 통신 위한 네트워크

Zone\_1과 Zone\_2는 농장 관리자가 다양한 하부 통신 기술(Ethernet, RS485, CAN, ZigBee 등)을 이용하여 설치할 수 있다고 간주하고, Zone\_3은 유무선 인터넷 망이 사용될 수 있다. 하지만 본 LCP에서는 특정 하부 통신 기술을 제한하지 않는다.

### ■ LCP 요구사항

LCP는 온실 환경뿐만 아니라, 노지 및 축사 분야에 적용이 가능하나, 이해를 위해 스마트온실 분야에 국한하여 설명하도록 한다. 스마트 온실은 환경 정보를 모니터링하기 위한 다수의 센서와 온실 환경을 자동 혹은 원격으로 조절하기 위한 구동기, 그리고 온실의 환경을 제어하기 위한 제어기로 구성된다. 그리고 사용자(혹은 농장 관리자)는 온실 운영 시스템을 통해 하나 이상의 온실을 관제할 수 있다.

LCP는 하나의 제어기와 다수의 센서와 구동기들이 설치되어 있는 다수의 온실 환경을 제어하는 것을 목적으로 한다. 이를 위하여 본 표준에서는 다음과 같은 프로토콜 요구 사항을 도출하고, 이들 요구 사항을 만족시킬 수 있는 LCP를 정의한다.

- LCP는 사용되는 하부 전송 기술(IP, RF 등)에 독립적으로 정의되어야 한다.
- LCP에서 사용하는 식별자(주소 체계)는 실제 온실 환경을 고려하여 정의되어야 한다.
- LCP는 업체마다 상이한 제어 방식을 수용할 수 있어야 한다.
- LCP는 센서/구동기노드, 제어기 및 온실 운영 시스템 프로그램 개발자에게 자율성을 최대한 제공할 수 있어야 한다.
- 스마트 온실 내 새로운 장비가 개발되거나 도입되더라도, 프로그램 개발자는 LCP 자체의 변경 없이도 확장할 수 있어야 한다.
- LCP는 제한된 데이터 통신 환경을 지원할 수 있어야 한다.
- LCP의 모든 기기들은 타이머를 탑재해야 한다.

### ■ LCP 동작 방법

LCP는 온실 운영 시스템, 제어기 및 센서/구동기노드 프로그램의 일부로서 동작하는 응용 계층 프로토

콜(Application Layer Protocol)이다.

센서/구동기노드 프로그램, 제어기 응용 프로그램 및 온실 운영 시스템 프로그램들은 그 목적인 바에 따라 제조사마다 각기 다른 운용 방법으로 구현된다. 대개의 경우 제조사들은 하부 전송 기술들(예: IP, LoRa, ZigBee 등)이 제공하는 API에 따라 구현하기 때문에, 하부 전송 기술이 변경될 때 전체 프로그램을 수정해야 할 필요가 있다. LCP는 이러한 문제점을 해결하기 위해, 제조사들이 별도의 프로그램 수정 없이 식별 체계와 전송 방식이 상이한 하부 전송 기술(예: IP, LoRa, ZigBee 등)들을 사용할 수 있도록 한다. 그림 3은 동일한 센서나 구동기 기능을 갖지만 네트워크 인터페이스가 다른 두 회사 제품과 제어기 응용 프로그램이 상호 연동하는 방법을 보여주고 있다. LCP를 사용할 경우, 네트워크 인터페이스(혹은 제조사)에 종속되지 않고 센서/구동기노드 프로그램과 제어기 응용 프로그램을 구현할 수 있다. 다음은 LCP의 간단한 동작 설명을 다루고 있다.

- LCP는 응용 계층 프로토콜로서 존재하며, 그 구현 방법에 따라 센서/구동기노드프로그램, 제어기 응용 프로그램 및 온실 운영 시스템 프로그램의 일부 혹은 별도로 동작한다.
- LCP는 하부 네트워크 기술과 정합하기 위하여, 정합(adaptation) 기술을 이용한다. LCP와 하부 전송 기술 간 정합을 위한 세부 메커니즘은 별도의 표준 규격에서 다룬다.

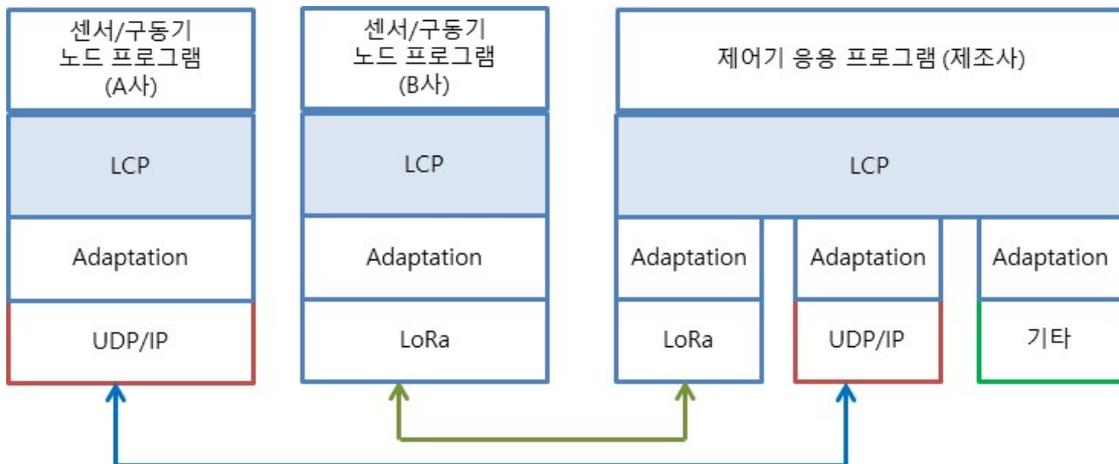


그림 3 - 하부 통신 기술에 비종속적인 LCP

### ■ LCP 프로토콜 특징

LCP는 총 4바이트의 공통헤더와 가변 메시지 바디를 갖는다. 메시지는 데이터그램 형태로 전달되며 메시지 송수신을 위한 주소 길이는 2바이트로 한정한다. 2바이트의 메시지 아이디를 정의한 이유는 스마트 팜 관련 현장을 고려하여 정의한 것인데, 가령 하나의 온실에 제어기를 포함하여 설치되어 있는 센서/구동기노드의 수가 100개라고 가정하고, 하나의 농장주가 자신이 관리하는 온실이 100개라고 하였을 때, 필요한 주소 수는 1만개이다. 또한 하나의 센서노드가 온도, 습도, 지열과도 같은 환경 정보를 동시에 취득할 수 있는 복합 센서노드처럼, 최근 하나의 노드가 다수의 동작을 할 수 있는 복합 노드가 개발되는 추세이므로, 하나의 농장에서 필요로 하는 노드 식별자 수는 위에서 산출한 것보다 작을 것으로 예상되기 때문이다.

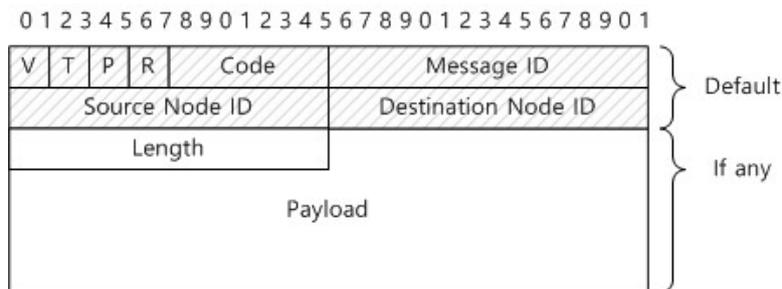


그림 4 - 하부 통신 기술에 비종속적인 LCP

메시지의 요청과 반환 값, 혹은 오류의 의미를 고정 헤더의 code 값을 변화시켜 사용할 수 있다.

요청		성공값 반환		오류값 반환	
Code	의미	Code	의미	Code	의미
0.01	GET	2.00	OK	4.00	Bad Request
0.02	POST	2.04	Changed	4.01	Unauthorized
0.03	PUT	2.05	Content	4.03	Forbidden
0.05	Registration			4.04	Not Found
0.06	Release			4.05	Method Not Allowed
0.07	Heartbeat			4.06	Not Acceptable
0.08	RESET			4.15	Unsupported Content-Format

그림 5 - LCP 프로토콜 code 값

### ■ LCP 메시지 전달 방법

LCP는 메시지 전달을 보증하지 않는 '알림'형태의 메시지와 메시지 전달여부를 보증하는 '요청-응답'형태로 메시지를 전달할 수 있다. 또한 장치의 종류, 구역 및 단말 식별자로 이루어진 2바이트의 주소체계를 조합하여, 개개의 장치는 물론 '구역전체' 혹은 '단말전체'로 메시지를 보내는 것이 가능하다.



그림 6 - LCP 주소 설정 규칙

### ■ LCP 메시지 구성 방법

LCP는 제어를 위한 메시지를 LCP 헤더가 아닌 LCP 메시지 본문에 실어 전송하는 방법을 취한다. 이 방법은 RTP(Real-Time Protocol)등에서도 많이 사용되는 방법인데, 다음 그림 7은 온도/습도/PH를 동시에 측정할 수 있는 복합 센서노드가 제어기에 자신이 측정한 센싱 값을 알리기 위해 제어 서비스 프로파일을 작성하여 비신뢰적 메시지 전달 방식으로 보낼 수 있다.

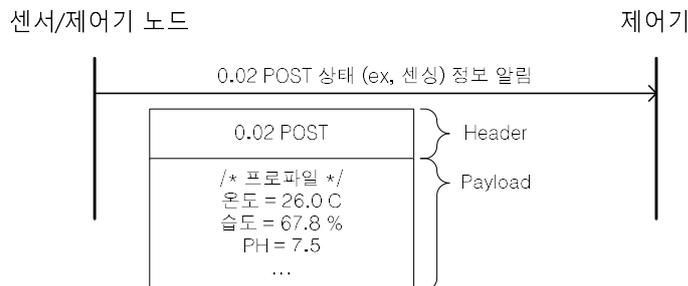


그림 7 - 측정한 센싱 값을 알리는 경우의 실시 예

만일 제어기가 센서노드에게 자신이 원하는 센싱 값을 요청하기 위해 제어 서비스 프로파일을 신뢰적 메시지 전달 방식으로 보내고자 할 경우, 메시지 전달 명령(code)을 0.01로 설정하고 자신이 원하는 센싱 값을 요청하는 절차를 보여준다. 실시 예로써, 그림 8의 경우, 비록 온도/습도/PH를 동시에 측정할 수 있는 복합 센서일지라도, 제어기가 온도와 습도 값만을 요청했기 때문에 제어기의 요청에 따른 해당 측정값만을 응답한다.

그림 7의 경우에는 ACK 혹은 응답이 필요하지는 않지만, 페이로드가 포함된 경우, LCP 공통 헤더의 T 필드 값을 '01'로 설정하고, 페이로드의 형태에 따라 P 필드 값을 '10'(페이로드가 바이너리 형태) 혹은 '11'(페이로드가 텍스트 형태)값을 설정한다.

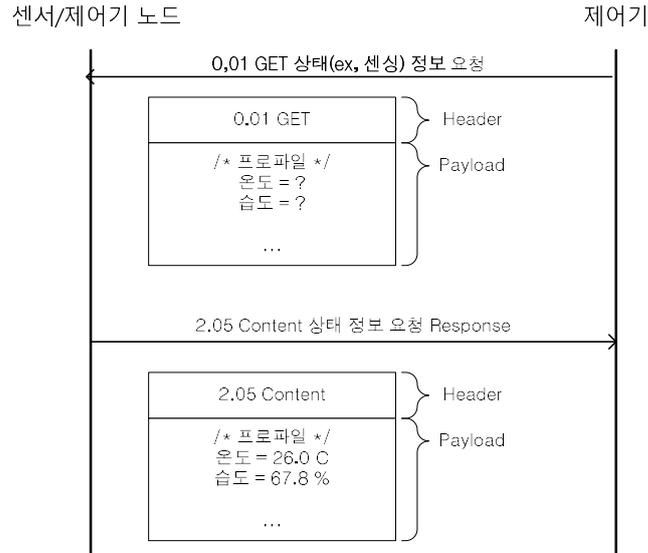


그림 8 - 요청에 의해 측정된 센싱 값을 알리는 경우의 실시 예

그림 8의 경우에서와 같이, ACK 혹은 응답이 필요하면서 페이로드가 포함된 경우, 요청 메시지는 LCP 공통헤더의 T 필드 값을 '00'로 설정하고, 페이로드의 형태에 따라 P필드 값을 '10'(페이로드가 바이너리 형태) 혹은 '11'(페이로드가 텍스트 형태)로 설정하며, 응답 메시지의 경우 LCP 공통 헤더의 T 필드 값을 '10'로 설정하고, 페이로드의 형태에 따라 P 필드 값을 '10'(페이로드가 바이너리 형태) 혹은 '11'(페이로드가 텍스트 형태)로 설정한다.

나.9) 스마트온실 서비스 제어프로토콜 기본 서비스절차 ('18 제안, '19 제정)

이 표준의 목적은 스마트온실에 설치된 제어기 노드가 다수의 센서노드, 구동기노드를 제어하기 위한 메시지 규격 및 동작 절차를 정의함으로써, 노드간 호환성 및 연동성을 확보하는 것을 목적으로 한다. 주요 내용으로 스마트온실에 설치된 제어기 노드가 센서노드, 구동기노드를 제어하기 위해 필요한 동작 요구사항과 이를 만족시키기 위한 메시지 규격, 동작 절차를 정의한다. 또한, 표준의 이해를 돕기 위해 메시지 예를 제공한다. 이 표준은 스마트온실 내 센서노드, 구동기노드를 제어하기 위한 제어메시지를 정의한다. 이 제어 메시지는 LCP표준을 이용하여 전달되며, TTAS.KO.1045, TTAS.KO.1046 메타데이터 규격에 맞춰 데이터를 기술하도록 하고 있다. 이와 관련하여, 관련 서비스를 제공하기 위해 필요한 각 노드의 동작 요구사항을 정의하였으며, 다음과 같은 서비스를 제어할 수 있도록 구성되어 있다. 여기에서 정의한 서비스들은 가장 기초적이고 기본적인 서비스로 향후 확장할 경우에도 재사용이 가능하도록 구성되었다.

LCP는 인터넷 환경에 국한되지 않는 공통된 개념을 갖는 프로토콜인 LCP는 CoAP 프로토콜의 개념을 이용하여 다양한 하위 물리 통신 환경에 적용 할 수 있도록 노드 식별자를 이용하여 메시지를 라우팅 한다. SCP 프로토콜은 액추에이터 / 센서노드와 컨트롤러 노드 사이의 스마트 온실의 센서 / 액추에이터 장치를 제어하는 데 사용됩니다. 센서노드는 장치 식별자, 센서 값, 센서 값 유형과 같은 여러 정보를 유지 관리하고 액추에이터 노드는 장치 상태를 유지하고 컨트롤러 노드로부터 제어 명령을 수신합니다. 컨트롤러는 이러한 정보에 액세스해야 하며 단일 노드에 여러 장치 (센서, 액추에이터)가 있을 수 있으므로 하나의 메시지로 여러 대상 장치에서 정보를 제어하고 검색 할 수 있어야 합니다. 또한 하나의 센서노드에 여러 센서가 부착 되더라도 원하는 값을 선택적으로 읽어야 한다. 이 인터페이스에서 SCP 프로토콜의 주요 기능은 다음과 같다.

- 센서 값 획득
- 액추에이터 상태 취득
- 액추에이터의 작동 / 작동 해제

센서 값을 취득하기 위한 기본 절차는 감지기 장치의 상태를 가져 오는 것이다. 첫째, 컨트롤러는 센서 노드에 LCP GET 메시지를 전송하고 대상 ID를 페이로드에 넣는다. 목표 ID는 조작되거나 조회될 오브젝트를 지정한다. 노드에 연결된 모든 센서의 정보를 얻으려면 필요한 경우 "SENSED / STATUS"를 대상 ID로 사용할 수 있다. 특정 센서의 측정값만 확인하는 경우 대상 ID는 "SENSED / STATUS / DeviceID = <device\_id>"와 같다. 이 요청 메시지를 받으면 센서노드는 요청에 대한 응답으로 LCP 2.00 OK 메시지를 전송하고 페이로드에는 <SensingStatus> 정보가 아래와 같이 지정된 형식으로 포함된다.

표1 - 센서 값 취득 요청에 대한 응답 메시지의 페이로드 정보

Name	Repeat	M/O	Type	
NodeID	1	M	number	
ActDevStatus	1	M	-	
DeviceID	N	1	M	number
SensingValue		1	M	number
ValueType		1	O	0: Integer, 1: float

센서 장치와 달리 컨트롤러는 창 열기, 팬 속도 등과 같은 액추에이터 장치의 현재 상태를 알아야 한다. 이를 위해 LCP GET 메시지를 사용하여 액추에이터의 상태를 가져 오는 방법을 사용한다. 제어기 노드는 액추에이터의 상태를 알아야 할 때, 목표 ID를 포함하는 페이로드로 LCP GET 메시지를 보낸다. 컨트롤러가 노드에 연결된 모든 액추에이터 장치에 대한 상태 정보를 얻으려는 경우 대상 ID는 "ACTDEV/STATUS"와 같아야 한다. 컨트롤러가 특정 장치의 상태 만 확인하려는 경우 "ACTDEV/STATUS/DeviceID=<device\_id>"를 대상 ID로 사용한다. 이 요청 메시지를 받으면 액추에이터 노드는 LCP 2.00 OK 메시지로 응답하고 페이로드에는 액추에이터 장치의 상태가 포함된다. 그러나 온실에는 개폐식 및 스위치형 두 가지 유형의 액추에이터가 있기 때문에 제어하는 방식/메시지가 약간의 차이를 가진다.

표 2 - 개폐형 구동기 제어 메시지

Name	Repeat	M/O	Type	
Nodded	1	M	number	
ActDevStatus	1	M	-	
DeviceID	N	1	M	number
OperationStatus		1	M	OPEN/OPENING/

				CLOSED/CLOSING
OperationPosition		1	0	Openness in percent
OperationSpeed		1	0	Operating speed in %/sec

표 3 - 스위치형 구동기 제어 메시지

Name	Repeat	M/O	Type
NodeID	1	M	number
ActDevStatus	1	M	-
DeviceID	N	1	number
OperationStatus		1	ON/OFF
OperationDuration		1	Duration of operation express by "PnYnMnDTnHnMnS" format
OperationTime		1	Time of operation as express by "hh:mm:ss"

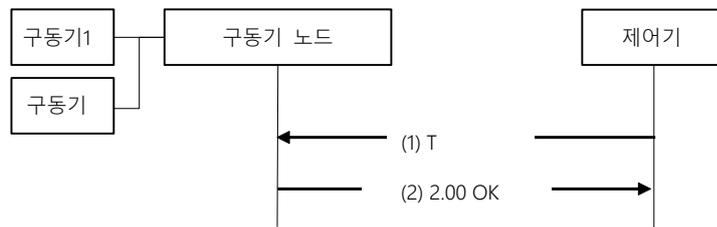


그림1 - 구동기노드와 제어기 구성의 예

위 그림은 구동기1에는 환풍기(DeviceType:5)를 구동기2에는 유동팬(DeviceType:6)를 부착한 모습을 표현한다. 이 표준안은 실제 개발자의 이해를 돕기 위해 개발된 코드를 이용해 도출된 실제 메시지를 예제로 아래와 같은 포맷으로 포함시키고 있다.

- ① Target ID에 "ACTDEV/STATUS"을 기재하여, LCP PUT 메시지를 송신한다.

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| V=1 | T=0 | P=3 | R=0 | HC=0 / LC=PUT | MessageID= 850 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Source ID=49281 | Dest ID=16513 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Length= 151 | Payload = |
+-----+-----+-----+-----+-----+-----+-----+-----+
{
  "ActDevControl": {
    "TargetID": "ACTDEV/STATUS",
    "OperationCommand": "ON",
    "OperationDuration": 0,
    "OperationTime": "01:00:00"
  }
}
+-----+-----+-----+-----+-----+-----+-----+-----+
  
```

- ② 센서노드는 LCP 2.00 OK 메시지를 제어기에게 송신한다.

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| V=1 | T=2 | P=3 | R=0 | HC=2 / LC=0 | MessageID= 850 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Source ID=16513 | Dest ID=49281 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Length= 301 | Payload = |
+-----+-----+-----+-----+-----+-----+-----+-----+
{
  "NodeID": 16513,
  "ActDevStat": [
    {
      "DeviceID": 1,
      "OperationStatus": "ON",
      "OperationDuration": 0,
    }
  ]
}
  
```

```

    "OperationTime": "01:00:00"
  },
  {
    "DeviceID": 2,
    "OperationStatus": "ON",
    "OperationDuration": 0,
    "OperationTime": "01:00:00"
  }
}
}
}
+-----+

```

**나-10) 경량 제어프로토콜을 위한 IP적응계층 동작요구사항 ('18 제안, '19 제정)**

이 표준의 목적은 경량형 제어프로토콜(LCP; TTA.KO.xxxx(2018))을 IP 통신환경에서 사용하기 위한 IP 적응 계층(IP adaptation layer)의 동작을 기술한다. 주요 내용으로 경량형 제어프로토콜(LCP)을 IP 통신 환경에서 사용하기 위한 변환계층인 IP적응계층(IPAL; IP Adaptation Layer)에 대한 설명과 동작 요구사항을 기술하고, 주요 기능들에 대한 동작 플로우 및 API를 제공한다. 아래 그림은 LCP 표준이 다루고자 하는 IP 적응 계층의 논리적 위치를 도시한다.

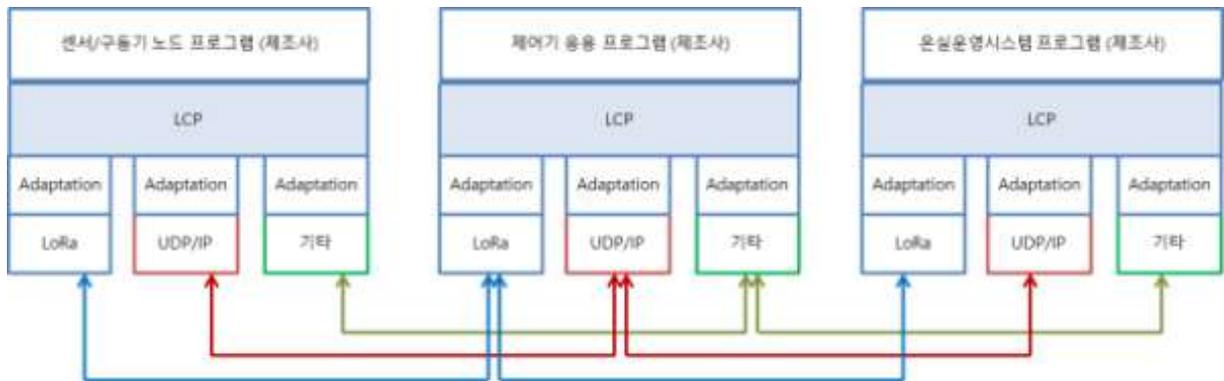


그림 1 - 서비스 제어 구조 및 LCP 표준의 범위

이 표준에서는 IPAL이 가져야 할 요구사항으로 다음과 같이 기술한다.

표 1 - IPAL 일반 요구사항

요구사항 번호	필수 여부	요구사항
GEN-REQ-01	M	IPAL은 상위 응용의 메시지를 송수신하기 위해 UDP를 이용한다.
GEN-REQ-02	O	상위 LCP 프로토콜은 IPAL의 내부 정보 및 상태에 대한 지식 없이, 미리 정해진 API를 이용해 호출한다.
GEN-REQ-03	O	하나의 LCP 노드는 복수개의 적응계층 위에 존재할 수 있다.
GEN-REQ-04	M	하나의 LCP 노드는 최소 1개 이상의 적응계층위에 탑재되어야 한다.
GEN-REQ-05	M	센서노드, 구동기노드는 하나의 제어기 노드에만 연결된다.
GEN-REQ-06	O	하나의 제어기는 복수개의 하우스를 제어할 수 있다.
GEN-REQ-07	M	LCP는 하위 물리통신계층 종류와 관계없이 동일한 동작을 수행한다. 즉, 하위 물리통신계층이 변경되더라도 동작이나 메시지의 내용에 변화가 발생하지 않는다.
GEN-REQ-08	M	IPAL 주소는 다음을 포함한다. - IP address - Port 번호 - LCP 노드 ID
GEN-REQ-09	O	하나의 제어기 노드는 복수의 AL을 통해 이종환경 장비들과 통신할 수 있다.

표 2 - IPAL 초기화 요구사항

요구사항 번호	필수 여부	요구사항
INI-REQ-01	M	상위 응용은 IPAL 초기화 및 동작설정을 위해 다음과 같은 정보를 제공해야

		한다. - 제어기의 IPAL 주소 - 노드의 IPAL 주소 - 노드의 아이디
INI-REQ-02	O	상위 응용은 IPAL이 UDP 메시지를 수신하기 위한 포트 번호를 제공한다. 만일, 명시적으로 제공되지 않을 경우 디폴트 포트 번호는 9999로 한다.
INI-REQ-03	M	IPAL은 지정된 네트워크 포트에 전달되는 UDP 메시지 수신이 가능해야 한다.

표 3 - IPAL 주소 변환 요구사항

요구사항 번호	필수 여부	요구사항
CNV-REQ-01	M	LCP 노드 아이디와 IPAL 주소간 맵핑 정보를 유지해야 한다.
CNV-REQ-02	M	LCP 노드 아이디로 목적지의 IPAL을 검색할 수 있어야 한다.
CNV-REQ-03	M	LCP 노드 아이디에 해당되는 IPAL 주소를 찾을 수 없는 경우, 에러를 리턴한다.
CNV-REQ-04	M	제어기 노드의 IPAL은 다른 노드들(센서, 구동기)로부터 메시지 수신시 맵핑 정보를 갱신하거나 신규 등록한다.

표 4 - IPAL 메시지 송수신 요구사항

요구사항 번호	필수 여부	요구사항
TRNS-REQ-01	M	IPAL은 LCP 메시지에서 DEST 노드의 아이디를 추출할 수 있어야 한다.
TRNS-REQ-02	M	다른 노드로부터 IPAL을 통해 메시지를 수신 시, LCP계층으로 메시지를 전달해야 한다.
TRNS-REQ-03	M	다른 노드의 IPAL로 메시지를 송신할 수 있어야 한다.

표 5 - IPAL 상태진단 요구사항

요구사항 번호	필수 여부	요구사항
DIAG-REQ-01	O	IPAL은 network connectivity 상태 변화가 발생하면 상위 응용계층에 알릴 수 있다.

#### 나-11) UHF 대역 400 MHz 무선링크 기반 스마트 온실 제어 프로토콜('21제안, '22개정)

##### (가) 연구 수행 개요

2019년에 TTA 단체표준으로 채택된 이후, 현장의 요구사항을 수렴하여 2021년에 개정한 이 표준은 현재까지의 유선기반 (RS485 Modbus) 스마트팜 관제 프로토콜을 탈피하여, 무선 기술을 이용하여 작업의 편의성을 증진시키고자 하는 목적을 갖는다.

스마트 온실에는 작물과 시설에 따라 다양한 형태의 물리적인 전송기술을 활용할 수 있다. 실제 농가에 설치된 스마트 온실에서 RS485, Wi-Fi, ZigBee, CAN 등의 전송 기술들이 혼용되어 사용되는 이유는, 스마트 온실을 구축함에 있어 작물의 종류, 재배 방법, 시설의 형태나 규모, 비용 등에 따라 각기 다른 요구사항이 발생하기 때문이다.

스마트 온실을 제어하기 위한 통신 선로를 유선으로 설치할 경우 안정적인 통신이 가능하지만, 낙뢰 등으로 인한 자연 피해, 영농 중 단락 등의 다양한 위험에 노출될 뿐만 아니라 이동성이 불편하다는 단점이 있다. 따라서 본 표준에서는 이러한 위험성과 불편함을 극복하기 위하여 UHF 대역의 RF 링크를 통신 선로로 활용하는 스마트 온실 제어 프로토콜을 제시하고자 한다.

본 표준에서 사용하는 UHF 대역은 업무 및 생활 무전기를 위해 할당된 비면허 산업용 주파수 대역으로, HAM용 무전기 등 다양한 무선통신장치에서 사용되곤 한다. UHF 대역의 통신의 경우, 전파의 직진성으로 인해 낮은 회절성으로 장애물이 있는 경우 통신거리가 짧아진다는 단점을 가진 고주파(가령 2.4GHz 대역) 대역과는 달리, 전파의 회절이 잘 되어 2.4GHz 대역을 사용하는 경우보다 통신거리가 길어진다는 장점을 갖는다. 그렇기 때문에, 식물이나 농기계 등 다양한 장애물이 존재할 수 있는 스마트 온실 내부의 통신 방식으로써, 비교적 먼 거리의 통신 반경과 회절성을 동시에 만족시킬 수 있는 UHF 대역의 무선 링크가 스마트 온실의 제어를 위한 무선 통신수단으로 활용 가능하다.

참고로, 국내에서 사용할 수 있는 UHF ISM 밴드는 424MHz, 447MHz 두 대역으로 나뉜다. 424MHz 대역의 경우 출력이 10dBm으로 제한되고 있으며, 채널 제어용 주파수를 포함하여 21개의 채널을 가진다. 본 표준에서 정의하는 스마트 온실 제어를 위한 프로토콜인 RF-SCP는 UHF 대역중 400MHz 주파수를 사용한다.

(나) 연구 주요 내용

1) 개요

다음 그림은 UHF 대역의 무선링크를 활용하여 스마트 온실을 제어하기 위한 통신 구성도를 보여준다.

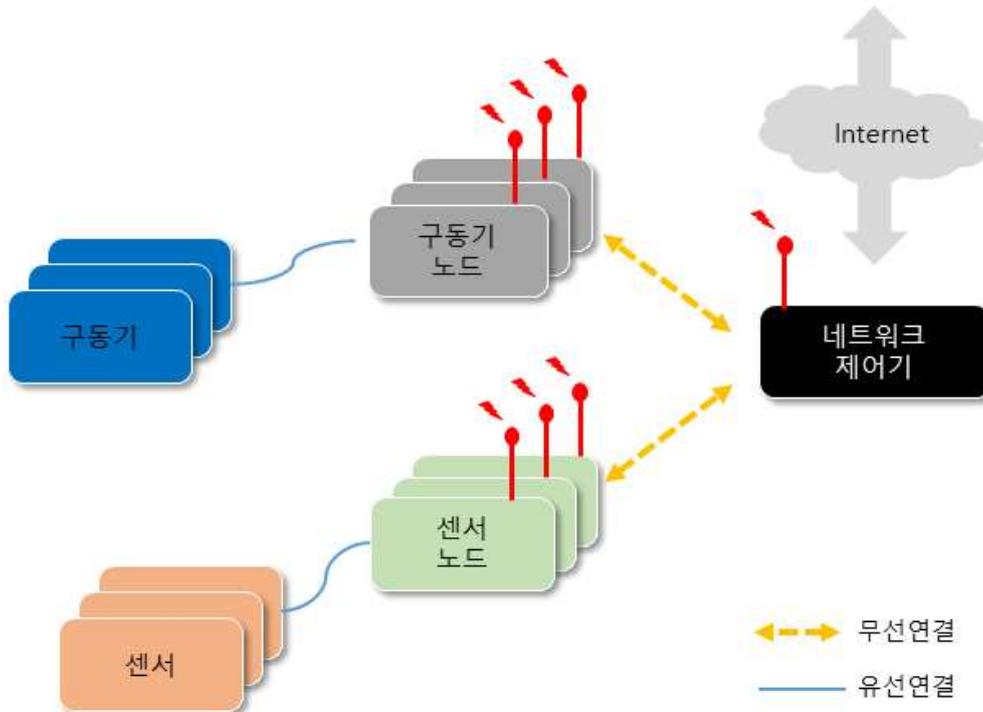


그림 1 - 무선링크 기반 스마트 온실 관리 프로토콜 기본 구성

무선링크 기반의 스마트 온실 관리 프로토콜은, 다수의 센서 노드와 구동기 노드들이 네트워크 제어기 사이에 사용된다. 네트워크 제어기는 통상적으로 온실제어기의 내부 기능으로 구성하거나 별도의 장치로 구성할 수 있다. 본 표준에서는 네트워크 제어기의 물리적인 구성 방법에 대한 제한을 두지는 않는다.

RF 통신모듈과 전원모듈로 구성되는 센서 노드는 하나 이상의 센서 모듈을 포함할 수 있다. 센서 모듈과 센서 노드사이의 H/W 연결 방법은 KS X 3266을 준용한다. RF 통신모듈과 전원모듈로 구성되는 구동기 노드는 하나 이상의 구동기 모듈(가령 모터나 릴레이 등)을 포함한다. 구동기 모듈과 구동기 노드사이의 연결 방법은 KS X 3265를 준용한다.

본 표준에서 정의하는 RF-SCP는 UHF 대역의 무선 채널을 이용하여 사용자 응용에게 연결성을 제공한다. RF-SCP는 다음 그림에서와 같이 OSI 7계층 중 네트워크와 트랜스포트 계층에 위치한다. RF-SCP는 UHF 대역의 네트워크 인터페이스(무선 미디어 및 UHF 대역 송수신카드)를 사용하여, 스마트 온실용 관리 응용 프로그램을 이용하는 제어기, 센서 노드 및 구동기 노드에게 연결성을 제공한다.

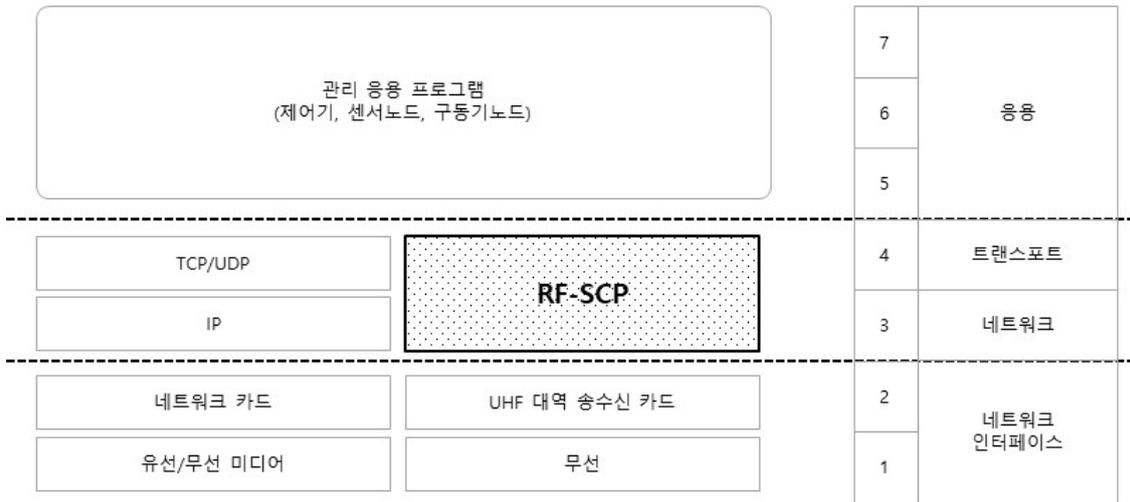


그림 2 - RF-SCP 프로토콜 계층

2) 메시지 구조

RF-SCP에서는 다음 표에서와 같이 총 14개의 메시지들이 정의된다. 이들 메시지들은 1) 센싱 정보의 알림, 2) 데이터나 제어 명령의 전송, 3) 장치의 설정을 위해 사용되는데, 모든 메시지들은 “요청-응답” 방식으로 동작한다.

표 1 - RF-SCP 메시지 종류

구분	상세구분	ID	내용
알림	주기적 알림	PNReq	▪ 주기적인 Notification
		PNRes	▪ 주기적인 Notification 응답
	이벤트 알림	ENReq	▪ 이벤트 알림
		ENRes	▪ 이벤트 알림 응답
전송	데이터	DTRReq	▪ 데이터 요청
		DTRRes	▪ 데이터 요청 응답
	제어명령	CCReq	▪ 제어명령 요청
		CCRes	▪ 제어명령 응답
	시퀀스 자동화	SARReq	▪ 시퀀스 자동화 요청
		SARRes	▪ 시퀀스 자동화 응답
설정	디바이스 특성 설정	DPSReq	▪ 디바이스 파라미터 설정 요청
		DPSRes	▪ 디바이스 파라미터 설정 응답
	디바이스 특성 값	DPVReq	▪ 디바이스 파라미터 값 요청
		DPVRes	▪ 디바이스 파라미터 값 응답

앞의 표에서 나열된 RF-SCP의 메시지들은 다음 그림과 같은 공통 메시지 구조를 갖는다.



그림 3 - RF-SCP 공통 메시지 구조

그림 3의 공통메시지의 필드들은 다음과 같은 의미를 갖는다.

- STX(4 바이트): 전송의 시작을 알리는 구분자로, 0xABCD1234를 사용한다.
- LEN(1 바이트): FCF부터 CRC까지의 길이(바이트 단위)
- FCF(1 바이트): Frame Control Field
  - . 전송(TX) 시에는 0x04 사용
  - . 수신(RX) 시에는 0x06 사용
- H\_ID(2 바이트): Handle ID이며, 랜덤값을 갖는다.
- G\_PAN(2 바이트): Global PAN ID의 하위 2B Byte 인근 PAN과의 충돌 방지 목적
- DST(2 바이트): 목적지 노드 ID
- SRC(2 바이트): 수신지 노드 ID
- Payload(N 바이트): 메시지의 데이터를 포함하며, 각각의 명령 형태에 따라 정의됨
- RSSI (2 바이트): 무선 전파세기
  - . 패킷 송신을 할 때에는 RSSI를 추가할 필요 없음
  - . 수신 시에는 무선 통신 모듈을 추가함
- CRC(1 바이트): 메시지 오류를 검출하기 위한 체크값
  - . STX 필드를 제외한 메시지 합의 하위 1 바이트
  - . 단, 수신인 경우 RSSI 필드가 추가되지만, 송신인 경우 RSSI 필드가 제외됨

### 3) 식별자

다음 그림은 RF-SCP가 탑재된 센서 노드, 구동기 노드 그리고 네트워크 제어가 동작하는 환경을 도시하였다. RF-SCP은 하나의 제어 구역을 그림에서와 같이 하나의 네트워크 제어를 통하여 제어 할 수 있는 하나 이상의 유무선 센서 노드와 구동기 노드들의 집합으로 이루어진 구역을 하나의 PAN으로 정의한다. PAN의 구성범위는 환경조절을 하고자 하는 사용자의 목적에 따라 하나의 온실 또는 다수 온실을 하나의 PAN으로 구성할 수도 있으며, 하나의 온실에 있어서 내부 경작 지역 하나를 PAN으로 구성할 수도 있다.

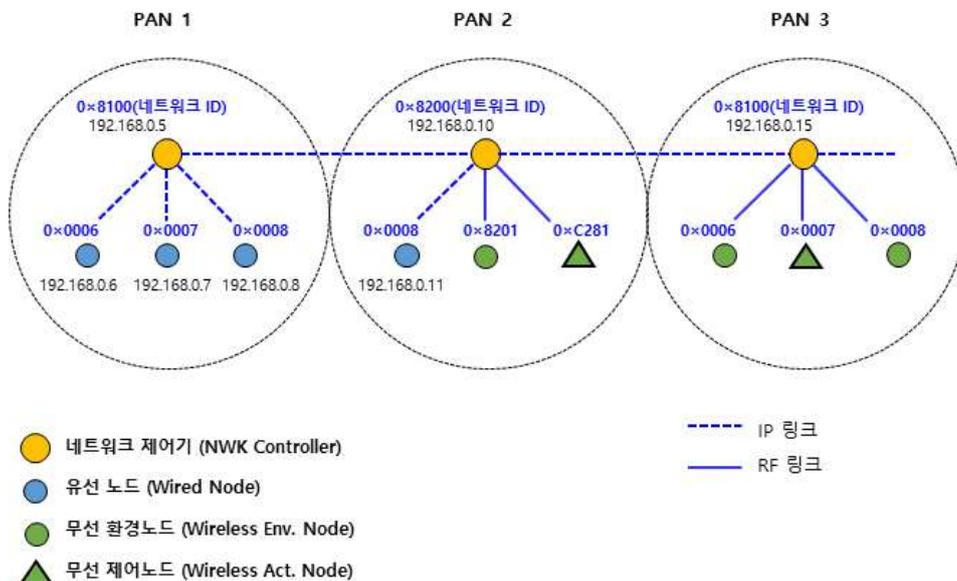


그림 4 - RF-SCP 식별체계

RF-SCP에서는 센서 및 구동기 노드와 네트워크 제어기 사이는 400MHz 대역의 무선링크를 사용하며, 네트워크 제어기 간은 IP링크를 사용한다. 본 표준에서는 네트워크 제어기 간의 통신 방식은 고려하지 않는다.

• 무선노드 ID

RF-SCP에서는 송수신 노드(센서 노드, 구동기 노드 및 네트워크 제어기)를 식별하기 위해 16비트의 노드 ID를 사용한다. 다음 그림은 무선노드 ID를 구성하기 위한 파라미터를 보여준다.

비트맵	15(1bit)	14(1bit)	13~8(6bit)	7(1bit)	6~0(7bit)
설명	무선 식별 플래그 1:무선 0:유선	전원 플래그 0:배터리 1:상시	PAN ID	장치 플래그 0:환경 1:제어	노드카운터

그림 5 - 무선노드 ID

각각의 필드는 다음과 같은 의미를 갖는다.

- 무선 식별 플래그(1비트): 무선 노드 여부를 표시하기 위한 플래그(무선: 1, 유선: 0)
- 전원 플래그(1비트): 노드의 전원 형태 식별을 위한 플래그(배터리: 0, 상시전원: 1)
- PAN ID(6비트): PAN 식별자(관리자가 임의 설정)
- 장치 플래그(1비트): 센서 혹은 구동기 노드임을 구분하기 위한 식별자(센서 노드: 0, 구동기 노드: 1)
- 노드 카운터(7비트): 동일 PAN내의 노드를 유일하게 식별하기 위한 필드(동일 PAN내 유일)

• 디바이스 ID

디바이스 ID는 센서 노드에 부착된 센서 모듈이나 구동기 노드에 부착된 구동기 모듈을 식별하기 위해 사용되는 다음 그림에서와 같이 구성된 2바이트(16비트)의 식별자이다.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
구분				종류				Reserved							

그림 6 - 디바이스 ID

디바이스 ID의 각 필드는 다음과 같은 의미를 갖는다.

- 구분(4비트): 센서 혹은 구동기를 구분하기 위한 필드
- 종류(4비트): 구분에 따른 상세 파라미터를 구분하기 위한 필드
- Reserved(8비트)

다음 표는 디바이스 ID를 환경센서와 구동기에 따라 할당하는 규칙을 제시한다.

표 2 - 디바이스 ID

	구분	종류	내용
환경센서 #1	0000(2)	0000(2)	온도
		0001(2)	습도
		0010(2)	조도
		0011(2)	지온
		0100(2)	지습
		0101(2)	CO <sub>2</sub>
		0110(2)	풍향
		0111(2)	풍속
		1000(2)	강우
		1001(2)	강우량
		1010(2)	기압
		1011(2)	상대습도

		1100(2)	일사량
		1101(2)	EC
환경센서 #2	0001(2)	0000(2)	Reserved
구동기 #1	0010(2)	0000(2)	SSR_Relay
		0001(2)	모터 일반(왕복)
		0010(2)	모터 개폐기
구동기 #2	0011(2)	0000(2)	Reserved
구동기 #3	0100(2)	0000(2)	응급 스위치
		0001(2)	토글 스위치

[비고] 가령 온도 환경센서 #1을 표시하기 위한 디바이스 ID는 환경센서 #1을 지칭하는 구분 필드값(00002)과 온도를 나타내는 종류 필드값(00002)을 포함하여, "0000 0000 0000 00012"이 된다. 마찬가지로, 습도 환경센서 #1을 표시하기 위한 디바이스 ID는 환경센서 #1을 지칭하는 구분필드값(00002)과 습도를 나타내는 종류 필드값(00012)을 포함하여, "0000 0001 0000 00012"이 된다.

#### 4) 프로토콜

- 주기적인 통보

주기적인 통보는 센서 노드 혹은 구동기 노드가 네트워크 제어기에게 주기적인 알림을 할 때 사용된다. 주기적인 통보는 PNReq와 PNRes 메시지를 통하여 수행된다. 센서 노드가 측정된 값을 주기적으로 알려 줄 때 사용되는 PNreq 메시지는, 다음 그림에서 보이는 것처럼 (14+N)바이트로 구성된다.

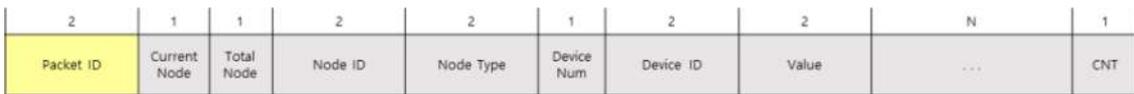


그림 7 - PNReq 메시지

PNReq 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): PNReq 메시지 식별자(0x0010)
- Current Node(1바이트): 현재 Node 번호
- Total Node(1바이트): 네트워크를 구성하는 Entry 수
- Node ID(2바이트): 개별 Node ID
- Node Type(2바이트): 개별 Node Type
- Device Num(1바이트): 디바이스별 센서 수
- Device ID(2바이트): 개별 디바이스 ID
- Device Value(2바이트): 개별 디바이스 값
- CNT(1바이트): 알림 카운터. 최초 Noti는 0으로 시작되며, 이후 1씩 증가. Overflow 시 1부터 시작. 만일 0인 경우 파라미터 싱크 수행

[비고] 모든 정보를 Node 별로 분절하여 서버에 전송한다. 이때의 Packet Header의 Time 정보는 동일하다.

성공적인 PNReq 메시지를 수신한 노드는 다음 (그림 8-2)에서 보이는 PNRes 메시지로 응답한다.

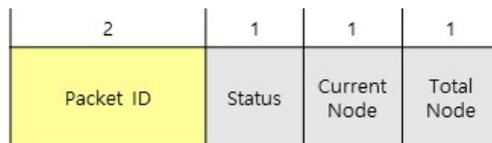


그림 8- PNRes 메시지

PNRes 메시지의 각각의 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): PNRes 메시지 식별자(0x8010)
- Status(1바이트): REQ에 대한 상태값
- Current Node(1바이트): PNReq에 사용된 Current Node 정보

- Total Node(1바이트): REQ Total Node 정보

• 이벤트 알림

주기적인 통보이외에 어떤 특정 이벤트가 발생하였을 때, 센서 노드 혹은 구동기 노드는 제어기에게 이벤트성 알림을 보낼 때 사용된다. 주기적인 통보는 ENReq와 ENRes 메시지를 통하여 이루어진다. 어떤 이벤트가 발생하였을 때 센서 노드가 알려줄 때 사용되는 ENreq 메시지는, 다음 그림에서 보이는 것처럼 (14+N)바이트로 구성된다.

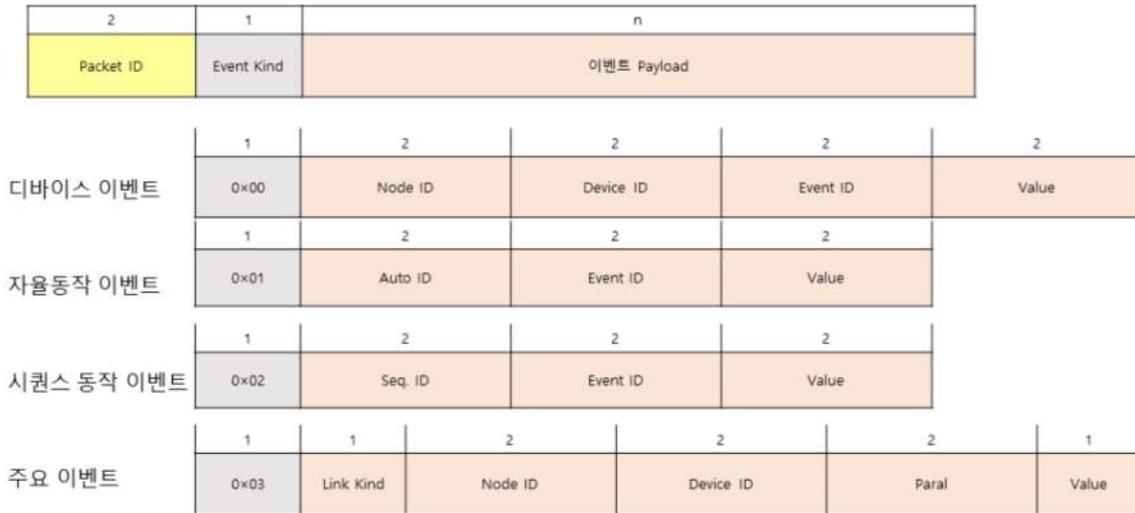


그림 9 - ENReq 메시지

ENReq 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): ENReq 메시지 식별자(0x0012)
- Event Kind(1바이트): 이벤트를 구분하는 식별자로서, 다음과 같이 4개의 이벤트를 정의한다.
  - . 디바이스 이벤트(0x00): 디바이스 동작에 관련 이벤트
  - . 자율동작 이벤트(0x01): 자율 동작 자동제어에서 발생하는 이벤트
  - . 시퀀스 동작 이벤트(0x02): 시퀀스 동작 자동화에서 발생하는 이벤트
  - . 스위치 동작 이벤트(0x03): 스위치 동작에 대한 이벤트
- Node ID(2바이트): 개별 Node ID
- Device ID(2바이트): 개별 디바이스 ID
- Event ID(2바이트): 센서별 Event ID를 나타내며 다음과 같이 6 종류의 이벤트를 정의한다.
  - . STOP(0x0000)
  - . START(0x0001)
  - . ERROR(0x0002)
  - . DEVICE\_OUTPUT(0x0003)
  - . SEND\_USER\_ALARM(0x0004)
  - . SEND\_ADMIN\_ALARM(0x0005)
- Auto ID(2바이트): 자율동작 자동화 ID를 표시하기 위한 필드로, 사용자나 관리자가 결정
- Seq. ID(2바이트): 시퀀스 동작 자동화를 식별하기 위한 필드
- Link Kind(1바이트): 링크 종류를 나타내며 다음과 같이 2개의 종류를 정의한다.
  - . 자동화 제어(0x01)
  - . 전체 제어(0x03)
- paral(2바이트): 자동화 또는 전체제어를 구분하기 위한 필드로서, 다음과 같이 2개의 값이 사용된다.
  - . 자동화 ID(자동화 ID)
  - . 전체 제어 ID(0x0000)
- Value(1바이트): Value 전송을 위한 값 필드로 다음과 같이 3가지 경우를 정의한다.
  - . STOP(0x00)

- . CW(0x01)
- . CCW(0x02)

ENReq 메시지를 수신한 노드는 다음 그림과 같은 ENRes 메시지로 응답한다.

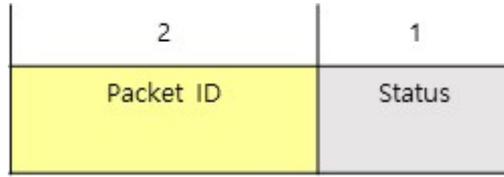


그림 10- ENRes 메시지

ENRes 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): ENReq 메시지 식별자(0x8012)
- Status(1바이트): REQ에 대한 상태값
- 데이터 요청  
센싱 데이터를 요청하기 위한 다음 그림의 DTReq 메시지는 특정 노드의 특정 디바이스에 대한 데이터를 요청하는데 사용된다.

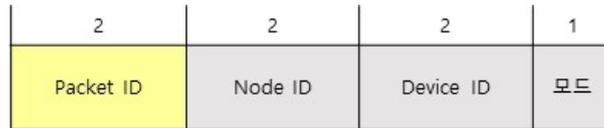


그림 11- DTReq 메시지

DTReq 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): DTReq 메시지 식별자(0x0020)
- Node ID(2바이트): 개별 Node ID
- Device ID(2바이트): 개별 Device ID. 만일 특정 노드 내 모든 디바이스 값을 요청하기 위해서는 0xFFFF를 사용한다.
- 모드(1바이트): 디바이스에게 데이터를 직접 요청 설정(1: 직접 요청, 2: 저장된 값)

DTReq 메시지를 수신한 노드는 다음 그림의 DTRes 메시지로 응답한다.

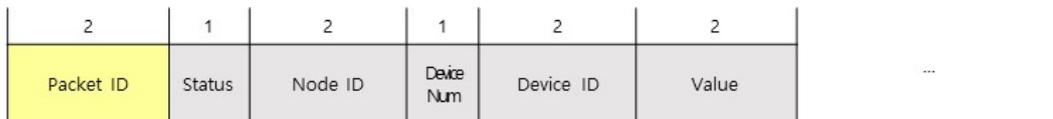


그림 12 - DTRes 메시지

DTRes 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): DTRes 메시지 식별자(0x8020)
- Status(1바이트): REQ에 대한 상태값
- Node ID(2바이트): 개별 Node ID
- Device Num(1바이트): Device 개수
- Device ID(2바이트): Device ID
- Value(2바이트): Device 값
- 제어 명령 요청/응답  
RF-SCP에서는 구동기에게 제어 명령을 하기 위하여 다음 그림의 CCReq 메시지를 사용한다.



그림 13 - CCReq 메시지

CCReq메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): CCReq 메시지 식별자(0x0021)
- Entry ID(2바이트): 개별 Entry ID
- Sensor ID(2바이트): 개별 센서 ID
- 제어명령 ID(2바이트): 개별 제어기의 제어 명령
- Para1(2바이트): 명령에 따라 사용되는 파라미터 1
- Para2(2바이트): 명령에 따라 사용되는 파라미터 2

CCReq 메시지를 수신한 노드는 다음 그림의 CCRes 메시지로 응답한다.

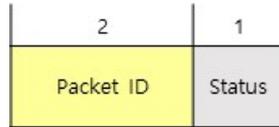


그림 14 - CCRes 메시지

각각의 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): CCRes 메시지 식별자(0x8021)
- Status(1바이트): 제어 명령 처리 결과
- 시퀀스 자동화 요청/응답  
시퀀스 자동화 명령을 하기 위하여 다음 그림의 SAREq 메시지를 사용한다.

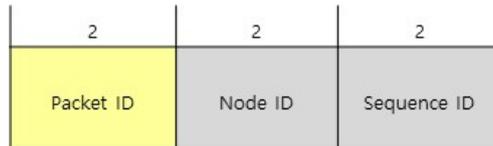


그림 15 - SAREq 메시지

SAREq메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): CCRes 메시지 식별자(0x0022)
- Entry ID(2바이트): 개별 Entry ID
- Sequence ID(2바이트): 시퀀스 자동화 ID

SAREq 메시지를 수신한 노드는 다음 그림의 SARes 메시지로 응답한다.

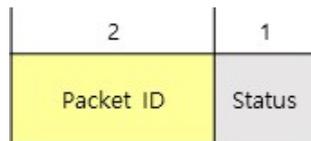


그림 16 - SARes 메시지

SARes 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): CCRes 메시지 식별자(0x8022)
- Status(1바이트): 제어 명령 처리 결과
- 디바이스 파라미터 설정 요청/응답

디바이스 파라미터 값을 설정하기 위해 다음 그림의 DPSReq 메시지를 사용한다.



그림 17 - DPSReq 메시지

DPSReq 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): DTSRes 메시지 식별자(0x0040)
- Device id(2바이트): 디바이스 ID,
- num(2바이트): 설정 파라미터 개수, multiple parameter 설정을 위해 사용되는 필드,
- Paraid(1바이트): 각각의 장치에 사용되는 파라미터 id
- Value(2바이트): 각각의 장치에 사용되는 파라미터 값
- Paraid+Value 쌍(3바이트): 다수의 파라미터가 존재할 때 사용

DPSReq 메시지를 수신한 노드는 다음 그림의 DPSRes 메시지로 응답한다.

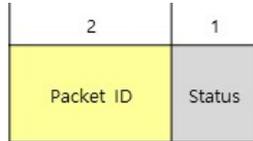


그림 18 - DPSRes 메시지

DPSRes 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): DTSRes 메시지 식별자(0x8040)
- Status(1바이트): REQ에 대한 상태값을 나타냄
- 디바이스 파라미터 값 요청/응답  
디바이스 파라미터 값을 요청을 하기 위해 다음 그림의 DPVReq 메시지를 사용한다.

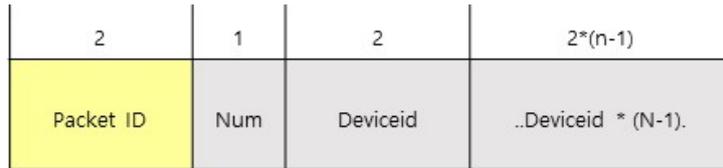


그림 19 - DPVReq 메시지

DPVReq 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): DTSRes 메시지 식별자(0x0041)
- Num(1바이트): 디바이스 개수
- Deviceid(2바이트): 디바이스 ID

DPVReq 메시지를 수신한 노드는 다음 그림의 DPVRes 메시지로 응답한다.

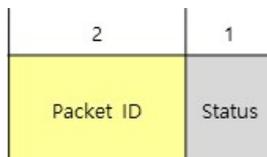


그림 20 - DPVRes 메시지

DPVRes 메시지의 각 필드는 다음과 같은 의미를 갖는다.

- Packet ID(2바이트): DPVRes 메시지 식별자( 0x8041)
- Status(1바이트): REQ에 대한 상태값

5) 프로토콜 파라미터

- 공통 Status ID  
메시지의 status 필드에 사용되는 값은 다음 표와 같다.

표 3 - RF-SCP 공통 상태 ID

	ID	내용
1	0x00	Success
2	0x01	Invalid Command
3	0x02	Invalid Parameter
4	0x03	Invalid STX
5	0x04	None Exist Entry
6	0x05	None Exist Device
7	0x06	None Exist Sensor
8	0x07	CRC 에러
9	0x08	NO RSP
10	0x09	BUSY
11	0x0A	TIMEOUT
12	0x0B	HOLD_TX
13	0x0C	BUFF FULL
14	0x0D	Same profile version
15	0xF0	NOT CONNECTED
16	0xF1	S_ERROR

- RF-SCP 센서 및 구동기 파라미터  
RF-SCP는 다양한 센서와 구동기들을 식별할 수 있도록 디바이스 ID를 정의하였다. 다음 표는 이들 디바이스 ID를 나타내기 위한 값들과 단위를 나열하고 있다.

표 4 - RF-SCP 파라미터

분류	구분 ID	종류 ID	내용	단위	데이터형태
환경1	0	0	온도 센서	℃	실수 (소수점 1자리)
		1	습도 센서	%	정수
		2	조도 센서	kW/m <sup>2</sup>	정수
		3	지온 센서	℃	실수 (소수점 1자리)
		4	지습 센서	%	정수
		5	CO <sub>2</sub> 센서	PPM	정수
		6	풍향	°	정수
		7	풍속	m/s	정수
		8	강우	-	TRUE/FALSE

		9	강우량	mm	정수
		A	기압	mmHg	정수
		B	상대 습도	%	실수(소수점 1자리)
		C	일사량 센서	kW/m <sup>2</sup>	정수
		D	EC 센서	ds/m	실수 (소수점 2자리)
제어1	2	0	SSR_Relay	On/Off	On/Off
		1	모터 일반(양복)	1단 동작	On/Off
		2	모터 개폐기	2단 동작	On/Off
조작	4	0	응급 스위치	-	On/Off
		1	토글 스위치	-	On/Off

나-12) 스마트온실 센서 메타데이터 (13종) ('17 제안, '18 제정)

스마트온실을 위한 센서 인터페이스[TTAK.KO-10.0903] 표준에서는 스마트 온실 내부의 센서 장치들을 작동시키기 위한 기계적 인터페이스와 전기적 인터페이스 및 측정 범위 등을 기술하고 있다. 온실에서 사용되는 각 센서에서 측정된 센서 데이터는 온실에 대한 환경 정보를 다루고 있으며, 측정된 환경 정보는 센서노드를 통하여 전달되어 해당 온실의 작물을 생육하기 위한 환경 조건을 조절하거나 최적의 생육 알고리즘을 도출하기 위한 기반 정보로 사용된다. 이를 위하여, 각 온실에서 측정된 환경 정보는 해당 온실에서 사용되는 이기종 센서와 상관없이 동일한 정보가 동일한 형태로 표현되고 해석되어야 한다.

따라서 본 연구에서는 스마트온실을 위한 센서 인터페이스[TTAK.KO-10.0903] 표준에서 정의한 센서별 측정된 센싱 데이터의 측정 범위에 대한 메타데이터의 데이터 엘리먼트 및 구조를 기술하는 내용을 기반으로 TTA 단체 표준을 제안하여 2018년 6월 최종 단체표준으로 승인되었다. 제안된 표준에서 정의하는 메타데이터는 다양한 인터페이스 프로토콜에서 공통으로 사용될 수 있다.

■ 센싱 디바이스 종류별 데이터 범위

센싱 디바이스는 센싱 디바이스의 타입에 따라 센싱 데이터 측정 범위가 다르다. 표 1은 각 센싱 디바이스 종류별 스마트 온실을 위한 센싱 데이터 측정 범위, 센싱 데이터 최소 측정 범위, 센싱 데이터 단위, 데이터 타입 등을 기술한다. 각 센싱 디바이스 종류별 데이터 측정 범위 및 데이터 단위는 스마트온실을 위한 센서 인터페이스 [TTAK.KO-10.0903] 표준에서 정의된 내용을 따른다.

표 1 - 센싱 디바이스 종류별 데이터 범위 및 데이터 타입

번호	분류	데이터 측정 범위	데이터 단위	타입
1	온도	-20~80	℃	xs:float
2	습도	0~100	%	xs:float
3	CO <sub>2</sub>	0~3000	ppm	xs:float
4	일사	0~2000	W/m <sup>2</sup>	xs:float
5	풍향	0~360	°(방향 각)	xs:float
6	풍속	0~40	m/s	xs:float
7	감우	on/off	-	xs:Integer
8	광양자	0~2000	umol/m <sup>2</sup> s	xs:float
9	토양함수율	0~50	% vol	xs:float
10	토양수분장력	0~100	kPa	xs:float
11	EC	0~10	dS/m	xs:float
12	PH	2~12	pH	xs:float
13	지온	-20~80	℃	xs:float

센서의 센싱 값 관련 정보는 표 2와 같다.

표 2 - 센서의 센싱 값 관련 정보

구분	분류	M/O	데이터 값
상태 정보	센싱 값 (Sensing Value)	M	각 센서 타입별 센싱 값

※ M: Mandatory, O: Optional

■ 센싱 디바이스 메타데이터

스마트 온실에서 사용되는 센싱 디바이스에 대한 메타데이터는 각 센싱 디바이스에 대한 속성 정보, 센싱 디바이스에서 보유하고 있는 센싱 상태 등의 디바이스 센싱 값 등을 포함하고 있다.

본 표준에서 정의된 메타데이터는 프로토콜에 따라 바이너리 인코딩 포맷 또는 JSON 인코딩 포맷 등 다양하게 적용될 수 있다. 따라서 메타데이터 기술 시 각 값은 바이너리 인코딩을 위한 값과 JSON 인코딩 값을 함께 기술한다.

■ 센싱 디바이스 정보

표 1의 센싱 디바이스 종류 별 센싱 디바이스 정보에 대한 메타데이터는 표 3과와 같다.

표 3 - 센싱 디바이스 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceInfo	디바이스별 클래스, 타입, 식별자 등을 포함하는 컨테이너.		
DeviceID	DeviceInfo의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:ID
DeviceClass	DeviceInfo의 엘리먼트. 디바이스의 클래스를 기술한다. 디바이스 클래스의 종류는 다음과 같다. - 0x01: Sensor(센서) - 0x02: Actuator(구동기)	M(1)	xs:NMTOKEN enumeration
DeviceType	DeviceInfo의 엘리먼트. 디바이스 클래스 별 디바이스 종류를 기술한다. 센싱 디바이스 종류는 다음과 같다. - 0x01: Temperature(온도) - 0x02: Humidity(습도) - 0x03: CO <sub>2</sub> - 0x04: Pyranometer(일사) - 0x05: WindDirection(풍향) - 0x06: WindSpeed(풍속) - 0x07: RainDetector(감우) - 0x08: Quantum(광양자) - 0x09: SoilMoisture(토양 함수율) - 0x10: Tensiometer(토양 수분 장력) - 0x11: EC - 0x12: pH - 0x13: SoilTemperature(지온)	M(1)	xs:NMTOKEN enumeration
Description	DeviceInfo의 엘리먼트. 센싱 디바이스에 대한 설명을 기술한다.	O(0-1)	xs:string
Location	DeviceInfo의 엘리먼트. 센싱 디바이스가 설치된 위치를 기술한다. 온실 내에서의 상세 위치, 온실 밖의 상세 위치 등 다양한 정보가 기술될 수 있다.	O(0-1)	xs:string
Target	DeviceInfo의 엘리먼트. 센싱 디바이스가 센싱하고자 하는 대상을 기술한다. 온실 내 대기, 배지(토양), 작물의 줄기, 잎, 과실, 뿌리 등 다양한 정보가 기술될 수 있다.	O(0-1)	xs:string
ValueUnit	DeviceInfo의 엘리먼트. 센싱 값의 단위, 디폴트는 표준에서 정한 단위, 센싱 값의 단위에 대한 디폴트 값 선택 시, 디바이스 종류별 표에서 정한 단위가 사용됨. - 0x01: Celcius - 0x02: Percentage - 0x03: ppm - 0x04: W/m <sup>2</sup> - 0x05: Degree - 0x06: m/s	O(0-1)	xs:NMTOKEN enumeration

엘리먼트	정의/의미	지원	타입
	<ul style="list-style-type: none"> <li>- 0x07: NoUnit</li> <li>- 0x08: <math>\mu\text{mol}/\text{m}^2\text{s}</math></li> <li>- 0x09: %vol.</li> <li>- 0x10: kPa</li> <li>- 0x11: dS/m</li> <li>- 0x12: pH</li> <li>- 0x13: Fahrenheit</li> </ul>		

※ M: Mandatory, O: Optional

### ■ 센싱 디바이스의 센싱 상태

센싱 디바이스의 센싱 상태 정보에 대한 메타데이터는 표 4와 같다.

표 4 - 센싱 상태 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceSensingStatus	센싱 디바이스의 센싱 값을 포함하는 컨테이너		
DeviceID	DeviceSensingStatus의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:IDREF
ValueType	DeviceSensingStatus의 엘리먼트. 값의 타입을 기술한다. 값의 타입은 다음과 같다. 디폴트 값은 float을 사용한다. - 0x00: integer - 0x01: float	O(0-1)	xs:NMTOKEN enumeration
SensingValue	DeviceSensingStatus의 엘리먼트. 값 타입(ValueType)에 따른 디바이스의 값을 기술한다. 디바이스 타입 및 디바이스 종류에 해당하는 센싱 값(Sensing Value)을 의미한다. - 센싱 값이 float로 표현되는 경우, 4 바이트로 표현되며 IEEE-754 표준을 따른다.	M(1)	타입에 따라 적용됨

※ M: Mandatory, O: Optional

나-13) 스마트온실 구동기 메타데이터 (9종) ('17 제안, '18 제정)

스마트 온실을 위한 구동기 인터페이스[TTAK.KO-10.0845]에서는 스마트 온실 내부에서 구동기 장치들을 작동시키기 위한 기계적, 전기적 연결 규칙과 작동 방식을 기술하고 있다. 온실에서 사용되는 각 센서에서 측정된 센서 데이터는 온실에 대한 환경 정보를 다루고 있으며, 측정된 환경 정보는 센서노드를 통하여 전달되어 해당 온실의 작물을 생육하기 위한 환경 조건을 조절하거나 최적의 생육 알고리즘을 도출하기 위한 기반 정보로 사용된다. 환경 조건을 조절하기 위해서는 구동기노드를 통하여 특정 구동기를 제어하는 제어 정보로 전달하여 최적의 생육 조건을 유지하고자 한다. 이를 위하여, 각 온실에서 구동기 제어 정보는 해당 온실에서 사용되는 기기종 구동기와 상관없이 동일한 정보가 동일한 형태로 표현되고 해석되어야 한다.

따라서 본 연구에서는 스마트 온실을 구동기 인터페이스[TTAK.KO-10.0845]에서 정의한 구동기 작동 방식에 대한 메타데이터의 데이터 엘리먼트 및 구조를 기술하는 내용을 기반으로 TTA 단체 표준을 제안하여 2018년 6월 최종 단체표준으로 승인되었다. 제안된 표준에서 정의하는 메타데이터는 다양한 인터페이스 프로토콜에서 공통으로 사용할 수 있다.

■ 구동기 디바이스 종류별 상태 정보

구동기 디바이스는 구동기 타입에 따라 상태 정보와 동작 제어 명령이 구분된다. 표 1은 각 구동기 디바이스 종류별 구동기 타입 및 가능한 상태 정보를 제공한다.

표 1 - 구동기 디바이스 종류 별 타입 구분 및 가능한 상태 정보

번호	분류	구동기 타입	상태 정보
1	천창	개폐형	OPENING(여는 중)/CLOSING(닫는 중)/ OPEN(열림)/CLOSED(닫힘)
2	측창	개폐형	OPENING/CLOSING/OPEN/CLOSED
3	보온재	개폐형	OPENING/CLOSING/OPEN/CLOSED
4	커튼	개폐형	OPENING/CLOSING/OPEN/CLOSED
5	환풍기	스위치형	ON(작동)/OFF(정지)
6	유동팬	스위치형	ON(작동)/OFF(정지)
7	관수 모터	스위치형	ON(작동)/OFF(정지)
8	관수 밸브	스위치형	ON(작동)/OFF(정지)
9	냉난방기	스위치형	ON(작동)/OFF(정지)

■ 스위치형 구동기의 상태 정보와 제어 명령 정보

스위치형 구동기의 상태 정보와 제어 명령 정보는 표 2와 같다.

표 2 - 스위치형 구동기의 상태 정보 및 제어 명령 정보

구분	분류	M/O	데이터 값
상태 정보	동작 상태 (Status)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	O	동작 유지 기간 날짜와 시간 정보를 기술
	동작 시간 (Operation Time)	O	동작 유지 시간 시(hour), 분(minute), 초(second) 정보를 기술
제어 정보	동작 명령 (Operation Command)	M	ON(작동)/OFF(정지)
	동작 기간	O	원하는 동작 기간

	(Operation Duration)		날짜와 시간 정보를 기술
	동작 시간 (Operation Time)	O	원하는 동작 시간 시(hour), 분(minute), 초(second) 정보를 기술

※ M: Mandatory, O: Optional

### ■ 개폐형구동기의 상태 정보와 제어 명령 정보

개폐형 구동기의 상태 정보와 제어 명령 정보는 표 3과 같다.

표 3 - 개폐형 구동기의 상태 정보 및 제어 명령 정보

구분	분류	M/O	데이터 값
상태 정보	동작 상태 (Operation Status)	M	OPENING(여는 중)/CLOSING(닫는 중) /OPEN(열림)/CLOSED(닫힘)
	동작 위치 (Operation Position)	O	현재 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	O	동작 속도
제어 정보	동작 명령 (Operation Command)	M	OPEN(열림)/CLOSE(닫힘)
	동작 위치 (Operation Position)	O	원하는 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	O	동작 속도

※ M: Mandatory, O: Optional

### ■ 구동기 디바이스 메타데이터

스마트 온실에서 사용되는 각 구동기 디바이스에 대한 메타데이터는 구동기 디바이스에 대한 속성 정보, 구동기 디바이스에서 보유하고 있는 동작 상태 정보, 구동기 디바이스의 동작 제어 명령 정보 등을 포함하고 있다. 표준에서 정의된 메타데이터는 프로토콜에 따라 바이너리 인코딩 포맷 또는 JSON 인코딩 포맷 등 다양하게 적용될 수 있다. 따라서 메타데이터 기술 시 각 값은 바이너리 인코딩을 위한 값과 JSON 인코딩 값을 함께 기술한다.

### ■ 구동기 디바이스 정보

구동기 디바이스 종류별 구동기 디바이스 정보에 대한 메타데이터는 표 4와 같다.

표 4 - 구동기 디바이스 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceInfo	디바이스별 클래스, 타입, 식별자, 위치 정보, 타깃 등을 포함하는 컨테이너.		
DeviceID	DeviceInfo의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:ID
DeviceClass	DeviceInfo의 엘리먼트. 디바이스의 클래스 기술한다. 디바이스 클래스는 다음과 같다. <sup>주1)</sup> - 0x01: Sensor(센서) - 0x02: Actuator(구동기)	M(1)	xs:NMTOKEN enumeration
DeviceType	DeviceInfo의 엘리먼트.	M(1)	xs:NMTOKEN

	<p>디바이스 클래스별 디바이스 종류를 기술한다. 구동기 디바이스 종류는 다음과 같다.<sup>주1)</sup></p> <ul style="list-style-type: none"> <li>- 0x01: UpperMotor(천창)</li> <li>- 0x02: SideMotor(측창)</li> <li>- 0x03: WarmMotor(보온재)</li> <li>- 0x04: CurtainMotor(커튼)</li> <li>- 0x05: VentilationFan(환풍기)</li> <li>- 0x06: CirculationFan(유동팬)</li> <li>- 0x07: IrrigationPump(관수 모터)</li> <li>- 0x08: IrrigationValve(관수 밸브)</li> <li>- 0x09: CoolingHeater(냉난방기)</li> </ul>		enumeration
Description	<p>DeviceInfo의 엘리먼트. 구동기 디바이스에 대한 설명을 기술한다.</p>	0(0-1)	xs:string
Location	<p>DeviceInfo의 엘리먼트. 구동기 디바이스가 설치된 위치를 기술한다. 온실 내에서의 상세 위치, 온실 밖의 상세 위치 등 다양한 정보가 기술될 수 있다. - 예: 북쪽/남쪽</p>	0(0-1)	xs:string
Target	<p>DeviceInfo의 엘리먼트. 구동기 디바이스 종류별 필요에 따라 추가적으로 기술하고자 하는 대상 정보를 기술한다. 천창과 측창의 경우, 구체적인 대상 정보 - 예: 1중창/2중창/3중창/4중창/5중창</p>	0(0-1)	xs:string
SpeedUnit	<p>DeviceInfo의 엘리먼트. 구동기 디바이스에 대한 동작 속도 값의 단위.<sup>주1)</sup> - 0x00: rpm</p>	0(0-1)	xs:NMTOKEN enumeration

※ M: Mandatory, O: Optional

주1) 바이너리 인코딩의 경우 분류 값과 JSON 인코딩의 경우 분류 값을 기술한다.

## ■ 구동기 디바이스 동작 상태

구동기 디바이스의 동작 상태 정보에 대한 메타데이터는 표 5와 같다.

표 5 - 구동기 디바이스 동작 상태 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceOperationStatus	디바이스 식별자, 동작 상태 정보, 상태 정보 관련 속성 정보 등을 포함하는 컨테이너.		
DeviceID	DeviceOperationStatus의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:IDREF
StatusValue	<p>DeviceOperationStatus의 엘리먼트. 구동기 디바이스의 동작 상태 값을 기술한다.<sup>주1)</sup></p> <ul style="list-style-type: none"> <li>o 스위치형 구동기 동작 상태 정보 <ul style="list-style-type: none"> <li>- 0x00: ON(작동)</li> <li>- 0x01: OFF(정지)</li> </ul> </li> <li>o 개폐형 구동기 동작 상태 정보 <ul style="list-style-type: none"> <li>- 0x10: OPENING(열리는 중)</li> <li>- 0x11: CLOSING(닫히는 중)</li> <li>- 0x12: OPEN(열림)</li> <li>- 0x13: CLOSED(닫힘)<sup>주2)</sup></li> </ul> </li> </ul>	M(1)	xs:NMTOKEN enumeration

OperationDuration	DeviceOperationStatus의 엘리먼트. 동작 유지 기간. 날짜와 시간 정보를 기술한다. 스위치형 구동기에 사용된다.	0(0-1)	xs:Duration
OperationTime	DeviceOperationStatus의 엘리먼트. 동작 유지 시간. 시(hour), 분(minute), 초(second) 정보를 기술한다. 스위치형 구동기에 사용된다.	0(0-1)	xs:Time
OperationPosition	DeviceOperationStatus의 엘리먼트. 원하는 동작 위치 정보, 0 ~ 100. - 0x00 : 0 %: 완전히 닫힌 상태 - 0x64 : 100 %: 완전히 열린 상태 개폐형 구동기에 사용된다.	0(0-1)	xs:integer
OperationSpeed	DeviceOperationStatus의 엘리먼트. 동작 속도, rpm을 디폴트 단위로 사용한다. 개폐형 구동기에 사용된다.	0(0-1)	xs:integer

※ M: Mandatory, O: Optional

주1) 바이너리 인코딩의 경우 분류 값과 JSON 인코딩의 경우 분류 값을 기술한다.

주2) 개폐형 구동기의 동작 상태 정보에서 'CLOSED' 상태는 'OPEN' 상태이면서 동작 위치가 '0%'인 경우와 동일하다.

## ■ 구동기 디바이스 동작 제어 명령

구동기 디바이스의 동작 제어 명령 정보에 대한 메타데이터는 표 6과 같다.

표 6 - 구동기 디바이스 동작 제어 명령 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceOperationCommand	디바이스 식별자, 제어 명령, 제어 명령 관련 속성 정보 등을 포함하는 컨테이너		
DeviceID	DeviceOperationCommand의 엘리먼트. 구동기 디바이스를 식별자를 기술한다.	M(1)	xs:IDREF
ControlCommand	DeviceOperationCommand의 엘리먼트. 구동기 디바이스에 대한 동작 제어 명령 (Operation Control Command). <sup>주1)</sup> o 스위치형 구동기 동작 제어 명령 정보 - 0x00: ON(작동) - 0x01: OFF(정지) o 개폐형 구동기 동작 제어 명령 정보 - 0x10: OPEN(열어라) - 0x11: CLOSE(닫아라)	M(1)	xs:NMTOKEN enumeration
OperationDuration	DeviceOperationCommand의 엘리먼트. 동작 기간. 날짜와 시간 정보를 기술한다. 스위치형 구동기에 사용된다.	0(0-1)	xs:Duration
OperationTime	DeviceOperationCommand의 엘리먼트. 동작 시간, 시(hour), 분(minute), 초(second) 정보를 기술한다. 스위치형 구동기에 사용된다.	0(0-1)	xs:Time
OperationPosition	DeviceOperationCommand의 엘리먼트. 동작 위치 정보, 0 ~ 100. - 0x00 : 0 %: 완전히 닫힌 상태 - 0x64 : 100 %: 완전히 열린 상태 개폐형 구동기에 사용된다.	0(0-1)	xs:integer

OperationSpeed	DeviceOperationCommand의 엘리먼트. 동작 속도, rpm을 디폴트 단위로 기술한다. 개폐형구동기에 사용된다.	0(0-1)	xs:integer
----------------	---	--------	------------

※ M: Mandatory, O: Optional

주1) 바이너리 인코딩의 경우 분류 값과 JSON 인코딩의 경우 분류 값을 기술한다.

나-14) 스마트축사 센서 메타데이터(실내·실외·안전 15종) ('18 제안, '19 제정)

온실과 마찬가지로 축사에서도 다양한 형태의 센서들이 사용된다. 일반적으로 스마트축사에서 사용되는 센서로는 내기 센서, 외기 센서 및 안전 센서로 분류할 수 있다. 2017년도 TTA PG426에서는 내기 센서 인터페이스(TTAK.KO-10.0980/R1), 외기 센서 인터페이스(TTAK.KO-10.0979/R1) 그리고 안전 센서 인터페이스(TTAK.KO-10.0981/R1)에 대한 표준을 개발한 바 있으며, 이 표준에서 정의된 각 센서의 종류는 다음 그림과 같다.

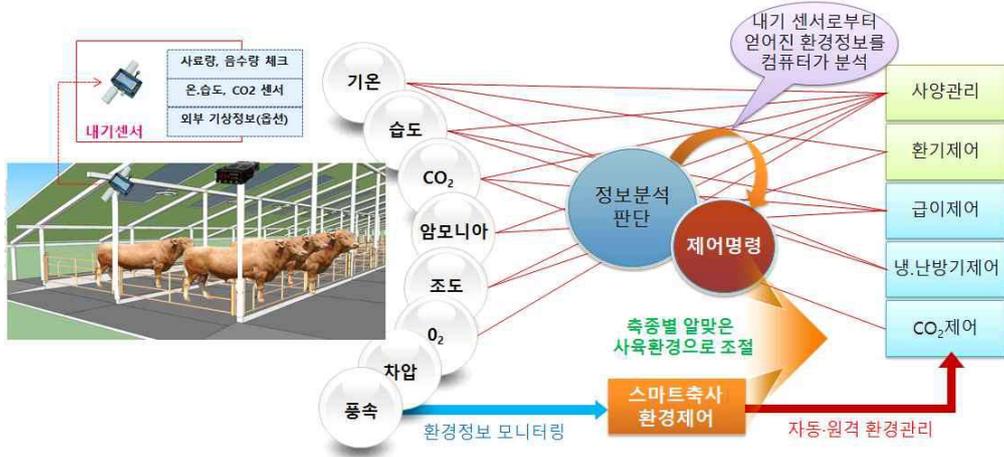


그림 1 - 스마트축사 내기 센서 (TTAK.KO-10.0980/R1)

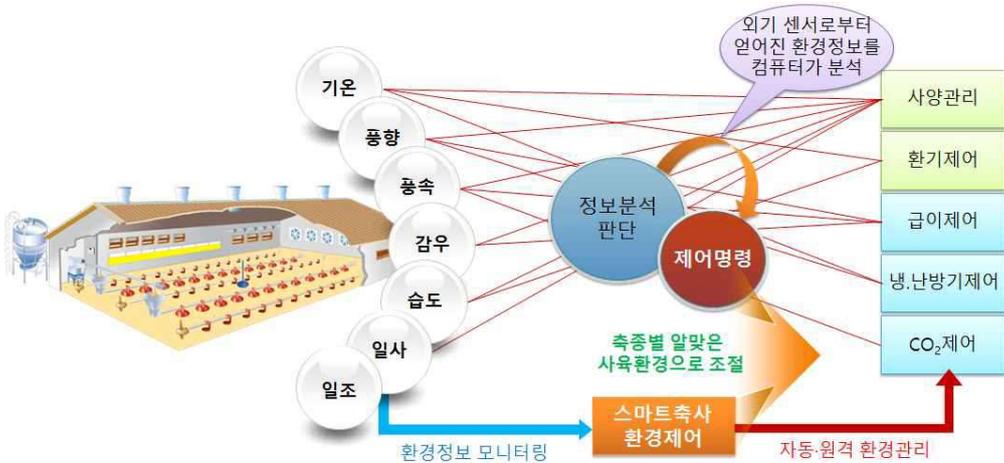


그림 2 - 스마트축사 외기 센서 (TTAK.KO-10.0979/R1)

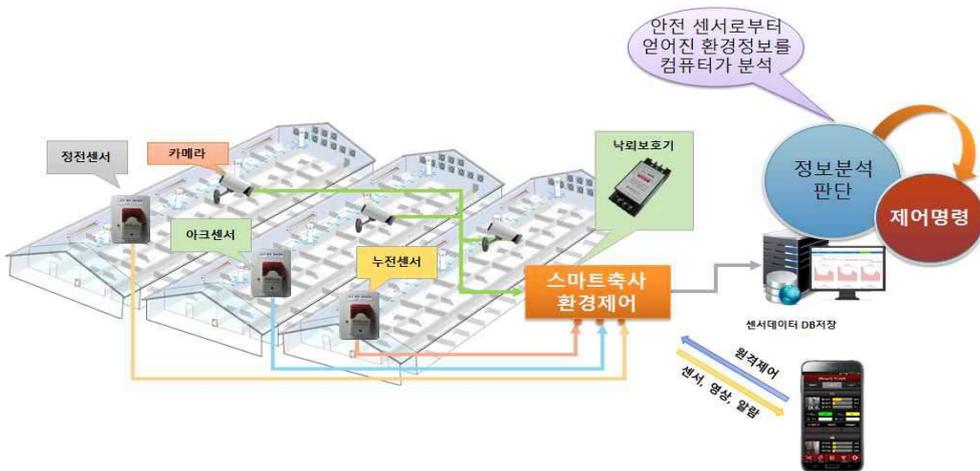


그림 3 - 스마트축사 안전 센서 (TTAK.KO-10.0981/R1)

하지만 스마트측사에서 사용되는 다양한 센서들은 제조사별로 센싱 값과 센싱 값 정보를 획득하기 위한 표현방식이 서로 상이하여 기기종 장치 간 상호 연동이 어렵기 때문에 센서를 위한 표준화된 메타데이터의 개발이 요구되었다. 이에 스마트측사에서 사용되는 내기 센서 8종에 대한 메타데이터 표준과 외기 센서 7종에 대한 메타데이터 표준을 각각 개발하고 정부 및 학계, 관련 업체의 의견수렴을 거쳐 TTA에 이를 제안하였으며 2019년 12월에 최종 채택될 예정이다. 스마트측사를 위한 안전센서 메타데이터 표준은 향후 개발 예정이다.

스마트 측사를 위한 내기 및 외기 센서 메타데이터 표준은 내기 센서(온도, 습도, CO<sub>2</sub>, 암모니아, 조도, O<sub>2</sub>, 차압, 풍속 센서 등 8종) 및 외기 센서(온도, 풍향, 풍속, 감우, 습도, 일사, 일조 센서 등 7종)별로 측정된 센싱 데이터의 측정 범위를 기반으로 하여 속성 정보, 센싱 값 및 센서 상태에 대한 속성 정보 등에 대한 메타데이터의 엘리먼트 및 구조를 정의하였으며, 이를 통해 기기종 장치간 상호 연동 및 각 장치 간 상호 연동을 위한 프로토콜 개발을 위한 데이터 표현 방식으로 활용될 수 있다.

## ■ 데이터 타입

내기 및 외기 센서 메타데이터 표준에 사용된 데이터 타입은 W3C의 XMLSchema에서 정의된 데이터 타입을 기반으로 하였으며 아래 표와 같다.

메타데이터 기술 시 컨테이너 요소는 연관된 요소를 그룹화하기 위해 정의되며, 아래 표에서 기술하는 ‘지원’ 열(column)은 해당 인스턴스의 요구 수준 및 발생 횟수를 기술한다.

요구 수준 및 발생 횟수에 대한 표기법은 다음과 같다.

- M: 필수 항목(mandatory), O: 선택 항목(optional)
- 1: 하나의 인스턴스, 0-1: 0 또는 하나의 인스턴스

표 1 - 스마트측사 내기/외기 센서 메타데이터 표준의 데이터 타입

데이터 타입	기술
xs:float	부동 소수점을 표시하는 데 사용됨. IEEE single-precision 32-bit floating point 형태를 따름.
xs:ID	식별자를 명시하는 데 사용됨.
xs:IDREF	xs:ID에 대한 참조를 명시하는 데 사용됨.
xs:integer	분수 부분 (fractional component)이 없는 수의 값을 명시하는 데 사용됨. 값의 범위로 {...-2, -1, 0, 1, 2...}의 무한 집합이 해당됨.
xs:string	문자(characters), 줄 바꿈(line feeds), 캐리지 리턴(carriage returns), 탭 문자(tab characters) 등을 포함하는 스트링 값을 명시하는 데 사용됨
xs:NMTOKEN	화이트 스페이스(white space) 대치 후의 스트링을 명시하는 데 사용됨. 줄 바꿈(line feeds), 캐리지 리턴(carriage returns), 연속적인 스페이스(space), 탭 문자(tab characters) 등은 하나의 스페이스(space)로 대치하고, 처음에 나오는 스페이스(space)들과 마지막에 나오는 스페이스(space)들은 제거한 경우를 의미함.
xs:NMTOKEN enumeration	나열 목록(enumeration) 제한을 가진 NMTOKEN 타입.

## ■ 센서 종류별 데이터의 범위 및 데이터 타입

센서는 타입에 따라 센싱 데이터 측정 범위가 다르다. 다음 표는 각각 내기 센서 및 외기 센서의 종류별 센싱 데이터 측정 범위, 센싱 데이터 단위, 데이터 타입 등을 기술한다. 각 센서 종류별 데이터 측정 범위 및 데이터 단위는 스마트 측사를 위한 내기 센서 인터페이스 (TTAK.KO-10.0980/R1) 표준 및 외기 센서 인터페이스(TTAK.KO-10.0979/R1)에서 정의된 내용을 따른다.

표 2 - 스마트측사 내기 센서의 종류별 데이터의 범위 및 데이터 타입

번호	분류	데이터 측정 범위	데이터 단위	타입
1	온도 센서	-40~60	°C	xs:float
2	습도 센서	0~100	%	xs:float
3	CO <sub>2</sub> 센서	0~10,000	ppm	xs:float
4	암모니아 센서	0~100	ppm	xs:float
5	조도 센서	0~1,000	lux	xs:float
6	O <sub>2</sub> 센서	0-25	%	xs:float
7	차압 센서	1~2	kPa	xs:float
8	풍속 센서	0.1~5	m/s	xs:float

표 3 - 스마트측사 외기 센서의 종류별 데이터의 범위 및 데이터 타입

번호	분류	데이터 측정 범위	데이터 단위	타입
1	온도 센서	-40~60	°C	xs:float
2	풍향 센서	0-360	°(방향 각)	xs:float
3	풍속 센서	0~75	m/s	xs:float
4	감우 센서	on/off	-	xs:Integer
5	습도 센서	0~100	%	xs:float
6	일사 센서	0-3000	W/m <sup>2</sup>	xs:float
7	일조 센서	0-1,440	분/일	xs:integer

센서의 센싱 값 및 센서 상태 관련 정보는 다음 표와 같다.

표 4 - 스마트측사 내기/외기 센서의 센싱 값 관련 정보

구분	분류	M/O	데이터 값
상태 정보	값(Value)	M	각 센서 타입별 센싱 값
	상태(Status)	M	각 센서의 상태(정상 및 고장여부 판단) 정보

※ M: Mandatory, O: Optional

■ 센서 메타데이터

스마트 측사에서 사용되는 내기 및 외기 센서에 대한 메타데이터는 각 센서에 대한 속성 정보, 센서에서 보유하고 있는 센싱 값 및 센서 상태 등을 포함하고 있다. 정의된 메타데이터는 프로토콜에 따라 바이너리 인코딩 포맷 또는 JSON 인코딩 포맷 등 다양하게 적용될 수 있다. 따라서 메타데이터 기술 시 각 값은 바이너리 인코딩을 위한 값과 JSON 인코딩 값을 함께 기술한다. 내기 및 외기 센서의 종류별 센서 정보에 대한 메타데이터는 각각 다음 표와 같다.

표 5 - 스마트측사 내기 센서의 센싱 디바이스 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceInfo	디바이스별 클래스, 타입, 식별자 등을 포함하는 컨테이너.		
DeviceID	DeviceInfo의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:ID
Class	DeviceInfo의 엘리먼트. 디바이스의 대 분류를 기술한다. 디바이스 대 분류의 종류는 다음과 같다. - 0x01: sensor(센서)	M(1)	xs:NMTOKEN enumeration

엘리먼트	정의/의미	지원	타입																																				
Type	DeviceInfo의 엘리먼트. 디바이스 대 분류 내 세부 종류를 기술한다. 센서 종류는 다음과 같다. - 0x01: temperature-sensor(온도 센서) - 0x02: humidity-sensor(습도 센서) - 0x03: CO <sub>2</sub> -sensor(CO <sub>2</sub> 센서) - 0x04: ammonia-gas-sensor(암모니아 센서) - 0x05: illuminance-sensor(조도 센서) - 0x06: oxygen-sensor(O <sub>2</sub> 센서) - 0x07: differential-pressure-sensor(차압 센서) - 0x08: wind-speed-sensor(풍속 센서)	M(1)	xs:NMTOKEN enumeration																																				
Model	DeviceInfo의 엘리먼트. 디바이스 모델 번호를 기술한다. 제조사가 자체 관리하는 해당 제품의 모델번호를 문자열로 표현하여, 영문과 숫자의 조합으로 기술된다.	O(0-1)	xs:string																																				
Name	DeviceInfo의 엘리먼트. 디바이스 모델명을 기술한다. 제어 기능 모듈의 UI에 단순 표출하기 위한 필드로 한글 기입이 가능하다.	O(0-1)	xs:string																																				
ValueUnit	DeviceInfo의 엘리먼트. 센싱 값의 단위, 디폴트는 표준에서 정한 단위, 센싱 값의 단위에 대한 디폴트 값 선택 시, 디바이스 종류별 이 표에서 정한 단위가 사용됨. <table border="1" data-bbox="424 1099 1015 1559"> <thead> <tr> <th>코드값</th> <th>대상</th> <th>기호</th> <th>타입</th> </tr> </thead> <tbody> <tr> <td>0x01</td> <td>온도 센서</td> <td>°C</td> <td>float</td> </tr> <tr> <td>0x02</td> <td>습도 센서</td> <td>%</td> <td>float</td> </tr> <tr> <td>0x03</td> <td>CO<sub>2</sub> 센서</td> <td>ppm</td> <td>float</td> </tr> <tr> <td>0x04</td> <td>암모니아 센서</td> <td>ppm</td> <td>float</td> </tr> <tr> <td>0x05</td> <td>조도 센서</td> <td>lux</td> <td>float</td> </tr> <tr> <td>0x06</td> <td>O<sub>2</sub> 센서</td> <td>%</td> <td>float</td> </tr> <tr> <td>0x07</td> <td>차압 센서</td> <td>kPa</td> <td>float</td> </tr> <tr> <td>0x08</td> <td>풍속 센서</td> <td>m/s</td> <td>float</td> </tr> </tbody> </table>	코드값	대상	기호	타입	0x01	온도 센서	°C	float	0x02	습도 센서	%	float	0x03	CO <sub>2</sub> 센서	ppm	float	0x04	암모니아 센서	ppm	float	0x05	조도 센서	lux	float	0x06	O <sub>2</sub> 센서	%	float	0x07	차압 센서	kPa	float	0x08	풍속 센서	m/s	float	M(1)	xs:NMTOKEN enumeration
코드값	대상	기호	타입																																				
0x01	온도 센서	°C	float																																				
0x02	습도 센서	%	float																																				
0x03	CO <sub>2</sub> 센서	ppm	float																																				
0x04	암모니아 센서	ppm	float																																				
0x05	조도 센서	lux	float																																				
0x06	O <sub>2</sub> 센서	%	float																																				
0x07	차압 센서	kPa	float																																				
0x08	풍속 센서	m/s	float																																				
ValueType	DeviceInfo의 엘리먼트. 센싱 값의 타입을 기술한다. 디폴트 값은 float을 사용한다. - 0x00: integer - 0x01: float	M(1)	xs:NMTOKEN enumeration																																				

※ M: Mandatory, O: Optional

표 6 - 스마트측사 외기 센서의 센싱 디바이스 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceInfo	디바이스별 클래스, 타입, 식별자 등을 포함하는 컨테이너.		
DeviceID	DeviceInfo의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:ID

엘리먼트	정의/의미	지원	타입																																
Class	DeviceInfo의 엘리먼트. 디바이스의 대 분류를 기술한다. 디바이스 대 분류의 종류는 다음과 같다. - 0x01: sensor(센서)	M(1)	xs:NMTOKEN enumeration																																
Type	DeviceInfo의 엘리먼트. 디바이스 대 분류 내 세부 종류를 기술한다. 센서 종류는 다음과 같다. - 0x01: temperature-sensor(온도 센서) - 0x02: wind-direction-sensor(풍향 센서) - 0x03: wind-speed-sensor(풍속 센서) - 0x04: RainDetector(감우 센서) - 0x05: humidity-sensor(습도 센서) - 0x06: pyranometer-sensor(일사 센서) - 0x07: sunshine-sensor(일조 센서)	M(1)	xs:NMTOKEN enumeration																																
Model	DeviceInfo의 엘리먼트. 디바이스 모델 번호를 기술한다. 제조사가 자체 관리하는 해당 제품의 모델번호를 문자열로 표현하여, 영문과 숫자의 조합으로 기술된다.	O(0-1)	xs:string																																
Name	DeviceInfo의 엘리먼트. 디바이스 모델명을 기술한다. 제어 기능 모듈의 UI에 단순 표출하기 위한 필드로 한글 기입이 가능하다.	O(0-1)	xs:string																																
ValueUnit	DeviceInfo의 엘리먼트. 센싱 값의 단위, 디폴트는 표준에서 정한 단위, 센싱 값의 단위에 대한 디폴트 값 선택 시, 디바이스 종류별 이 표준에서 정한 단위가 사용됨. <table border="1" data-bbox="429 1218 1018 1626"> <thead> <tr> <th>코드값</th> <th>대상</th> <th>기호</th> <th>타입</th> </tr> </thead> <tbody> <tr> <td>0x01</td> <td>온도 센서</td> <td>°C</td> <td>float</td> </tr> <tr> <td>0x02</td> <td>풍향 센서</td> <td>°</td> <td>float</td> </tr> <tr> <td>0x03</td> <td>풍속 센서</td> <td>m/s</td> <td>float</td> </tr> <tr> <td>0x04</td> <td>감우 센서</td> <td>-</td> <td>integer</td> </tr> <tr> <td>0x05</td> <td>습도 센서</td> <td>%</td> <td>float</td> </tr> <tr> <td>0x06</td> <td>일사 센서</td> <td>W/m<sup>2</sup></td> <td>float</td> </tr> <tr> <td>0x07</td> <td>일조 센서</td> <td>분/일</td> <td>integer</td> </tr> </tbody> </table>	코드값	대상	기호	타입	0x01	온도 센서	°C	float	0x02	풍향 센서	°	float	0x03	풍속 센서	m/s	float	0x04	감우 센서	-	integer	0x05	습도 센서	%	float	0x06	일사 센서	W/m <sup>2</sup>	float	0x07	일조 센서	분/일	integer	M(1)	xs:NMTOKEN enumeration
코드값	대상	기호	타입																																
0x01	온도 센서	°C	float																																
0x02	풍향 센서	°	float																																
0x03	풍속 센서	m/s	float																																
0x04	감우 센서	-	integer																																
0x05	습도 센서	%	float																																
0x06	일사 센서	W/m <sup>2</sup>	float																																
0x07	일조 센서	분/일	integer																																
ValueType	DeviceInfo의 엘리먼트. 센싱 값의 타입을 기술한다. 디폴트 값은 float을 사용한다. - 0x00: integer - 0x01: float	M(1)	xs:NMTOKEN enumeration																																

※ M: Mandatory, O: Optional

내기 및 외기 센서의 센싱 값 및 센서 상태 정보에 대한 메타데이터는 다음 표와 같다.

표 7 - 스마트측사 내기/외기 센서의 센싱 값 및 센서 상태 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceSensingStatus	센서의 센싱 값 및 센서 상태를 포함하는 컨테이너		
DeviceID	DeviceSensingStatus의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:IDREF
Value	DeviceSensingStatus의 엘리먼트. 센서의 측정 값을 기술한다. 디바이스 타입 및 디바이스 종류에 해당하는 센싱 값을 의미한다. - 센싱 값이 float로 표현되는 경우, 4 바이트로 표현되며 IEEE-754 표준을 따른다.	M(1)	타입에 따라 적용됨
Status	DeviceSensingStatus의 엘리먼트. 센서의 상태를 기술한다. 0x00: OK/READY 0x01: ERROR 0x02: BUSY 0x03: 동작 전압 이상 0x04: 동작 전류 이상 0x05: 동작 온도 이상 .... 0x101: 센서 및 소모품 교체 요망 0x102: 센서 교정 요망 0x103: 센서 점검 필요 0x900-999: 제조사 정의 에러코드 ....	M(1)	xs:NMTOKEN enumeration

※ M: Mandatory, O: Optional

나-15) 스마트팜관제 프레임워크-제1부: 베이스프로토콜(1종) ('19 제안, '20 제정)

스마트팜은 기존의 전통적 농업에 IoT 등 ICT기술을 융합하여 생산성과 효율성을 제고함으로써 고부가 가치 창출을 추구하는 첨단 농업 기술로서, 다양한 IT기기들이 활용되며, 구성요소들간 유기적인 상호동작을 통해 구현된다. 특히 센서 및 구동기 노드는 IoT 기술을 기반으로 구현되기 때문에 각 구성요소간의 정보교환을 위한 네트워킹 방식은 인터넷 프로토콜 기반으로 구성될 수 있다. 스마트팜이 인터넷을 통해 연결되면, 인터넷을 대변하고 있는 웹 기술부터 클라우드, 빅데이터, 인공지능 등의 이른바 4차산업혁명기술 등을 비롯한 인터넷 기반의 풍부한 서비스 관련 기술들이 스마트팜에 접목될 수 있게 된다, 인터넷과 연동을 통해 스마트팜 관련 다양한 인터넷 서비스가 확장될 가능성이 커짐에 따라, 웹기반 스마트팜 관리 및 제어 프레임워크의 필요성 또한 대두되게 된다.

스마트팜은 적용분야에 따라 작물을 생육하는 스마트온실, 가축을 키우는 스마트축사, 과수를 생육하는 스마트과수원 등으로 분류될 수 있다. 즉, 스마트팜은 생육하는 대상에 따라 분류될 수 있으며, 생육하는 대상의 품종에 따라 다시 세분화된다. 따라서 스마트팜은 영농의 종류 및 방식에 따라 다양한 형태의 시설 및 장치들이 구비될 수 있는데, 본 표준에서는 그 중 시설원예를 주요 대상으로 하는 스마트온실의 경우를 예로 설명한다. 스마트온실은 생육 및 환경 정보를 모니터링 하기 위한 센서 장치, 스마트팜 시설을 제어하는 구동기 장치, 각 시설에 부착된 장치들을 제어하는 제어기, 그리고 제어기와의 연동을 통해 다수의 스마트온실을 통합적으로 운영하는 온실운영시스템, 원격지의 스마트온실들을 통합하여 관리하는 온실통합관리시스템으로 구성된다.

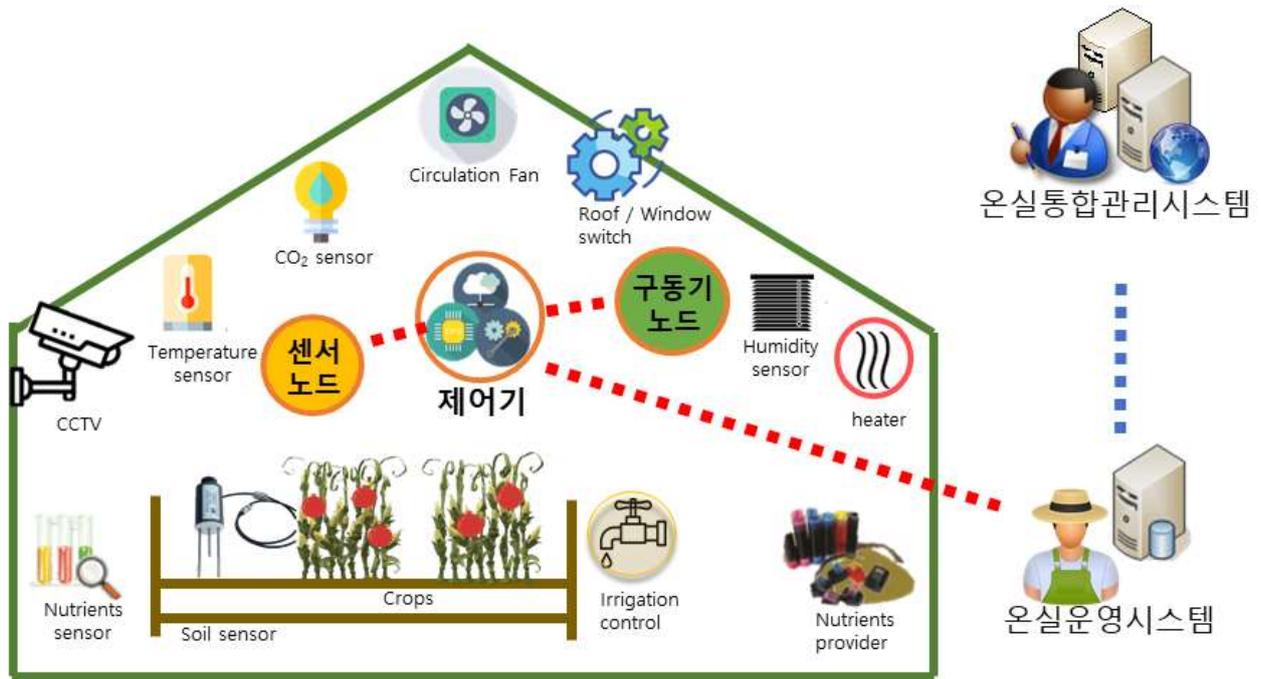


그림 1 - 스마트 온실 개념도

센서는 온도(Temperature), 습도(Humidity), 이산화탄소(CO<sub>2</sub>), 토양(Soil), 양분(Nutrient sensor) 등 다양한 센서들이 있으며, 각 센서들은 정해진 주기 또는 요청에 따라 현재의 측정값을 제어기 및 온실운영시스템으로 전달한다. 구동기는 천창/측창(Roof/Window switch), 환기(Circulation Fan), 난방(Heater), 관수(Irrigation Control), 관비(Nutrient Provider), 영상장치(CCTV) 등 스마트 온실의 환경조절 및 상태 관측을 위한 다양한 구동기들이 있으며 제어신호에 따라 고유의 동작을 수행한다. 제어기는 구동기 장치를 제어하는 신호를 전달하고, 또한 센서 및 구동기가 전송하는 데이터를 운영관리시스템으로 전달한다. 온실 운영시스템은 복수의 온실을 통합하여 관리하는데, 농가가 운영하는 각 온실의 상태를 전반적으로 관찰하고 통합적으로 제어한다. 온실통합관리시스템은 인터넷을 통해 다수 농가의 온실운영시스템과 연결되어 각 농가에서 생성하는 데이터를 수집하고, 수집된 데이터를 분석하고, 분석된 데이터를 기반으로 최적생육 알고리즘을 개발하고, 개발된 알고리즘을 활용하여 각 농가에 적합한 최적의 조건으로 온실제어가 가능하다.

한편, Open Connectivity Foundation (OCF)는 다양한 IoT 유/무선 연결 기술을 활용하여, 상호연동성을 추구하고 있으며, 표준 개발과 동시에 생태계 확대를 위한 오픈소스 프로젝트(IoTivity)도 진행하고 있다. OCF는 REST 구조 기반의 경량 프로토콜로 (예: CoAP) IoT 장치들을 연결하는데, 현재는 스마트홈 처럼 가전에 특화되어 있다.

CoAP 기반 스마트팜 관리 및 제어 프레임워크는 여러 응용을 지원하기 위해 식별 및 주소, 탐색, 리소

스 모델, CRUDN(Create, Request, Update, Delete, Notification), 장치 관리 및 보안 기능을 제공한다. 다음은 CoAP 기반 스마트팜 관리 및 제어 프레임워크가 가지는 특징 및 기능을 설명한다.

[비고] CoAP기반 스마트팜 관리 및 제어 프레임워크가 지원하는 응용의 예로는 센서/구동기/제어기 노드 관제 프로그램, 운영시스템 관리 프로그램 등을 들 수 있다.

1) 프레임워크 기능구조

CoAP 기반 스마트팜 관리 및 제어 프레임워크는 다음 그림과 같이 클라이언트-서버 기반의 구조를 가진다. 센서 및 구동기와 같이 고유의 기능을 수행하는 스마트팜 기자재는 서버로 정의될 수 있다. 각 기자재가 가지는 고유 기능은 리소스로 표현될 수 있다. 센서의 경우 온도, 습도 등과 같이 센서로부터 측정된 값이거나, 스위치와 같이 구동기를 제어하는 명령 등이 모두 리소스로 표현될 수 있다. 제어기는 다수의 센서 및 구동기로부터 값을 전달받기 때문에 다수의 장치를 내재하는 복합노드로 표현될 수 있다. 운영시스템과 같이 서버로부터 값을 수집하거나, 서버를 제어하는 명령을 보내야 하는 장치는 클라이언트로 정의될 수 있다. 클라이언트는 다수의 서버들과 연결되어 스마트팜 고유의 기능을 수행하도록 각 서버를 제어하게 된다. 이때 서버와의 연결은 http와 호환가능하도록 RESTful 오퍼레이션으로 수행된다.

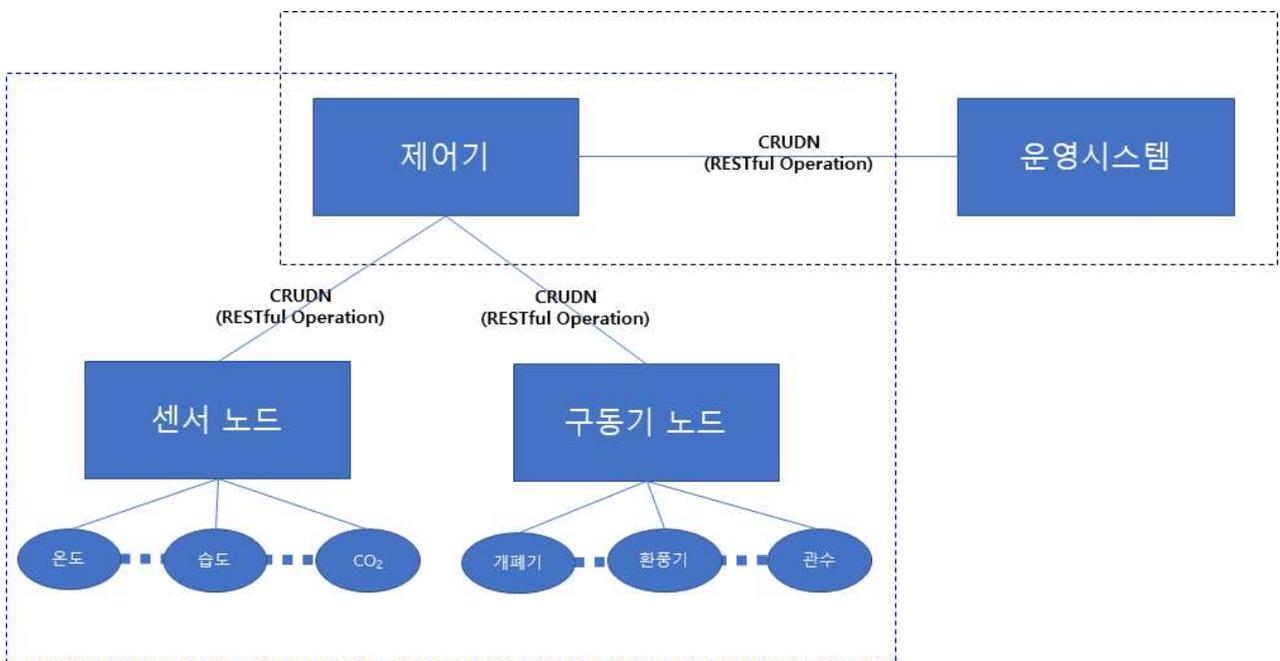


그림 2 - CoAP 기반 스마트팜 관리 및 제어 프레임워크를 위한 구조

다음 그림은 CoAP 기반 스마트팜 관리 및 제어 프레임워크 기능블록을 나타낸다. 여기에는 각 장치 및 장치의 리소스를 식별하고 데이터를 주고받을 수 있도록 하는 식별 및 주소 기능, 각 장치가 가진 특성 및 기능을 리소스로 표현할 수 있도록 모델링 하는 리소스 모델 기능, 장치간 상호 인식을 위한 탐색 기능, 다른 장치의 리소스를 활용하기 위한 CRUDN 기능, 장치의 상태 및 유지관리를 위한 장치관리 기능, 장치간 연결 및 연결된 장치간 교환하는 데이터에 대한 보안을 위한 보안 기능이 포함된다.

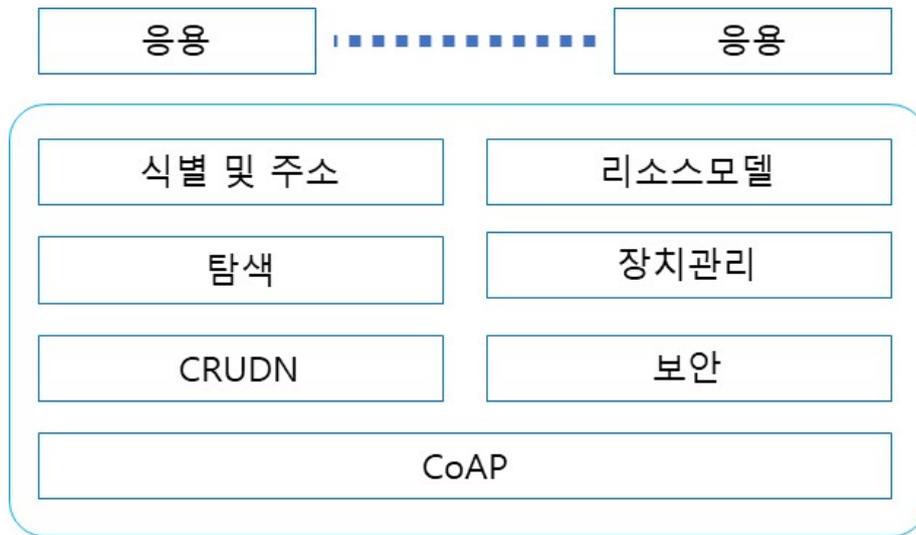


그림 3 - CoAP 기반 스마트팜 관리 및 제어 프레임워크 기능블록

## 2) 식별 및 주소

스마트팜 내의 각 장치간 효율적인 상호작용을 위해서는 각 장치 및 장치의 리소스를 구분하고, 이들에 대해 이름 및 주소를 지정하는 수단이 필요하다. 식별자(ID)는 스마트팜 도메인에서 장치 및 리소스를 명확히 식별하는 역할을 한다. 각 식별자는 수명주기동안 변경될 수 없으며 명확해야 한다.

식별자는 다음과 같이 세 가지 유형으로 구분될 수 있다.

- 영구ID: 장치에 할당되는 고유 ID로서 변경되지 않는다.
- 장치ID: 장치가 가지는 리소스의 ID로서, 장치가 On 할 때 부여되어 장치가 Off 할 때 소멸하며, 유효 장치간 상호작용에 활용된다.
- 플랫폼ID: 장치가 탑재된 플랫폼 ID로서, 단일 플랫폼(예: 제어기)에 여러 논리 장치(예: 센서 및 구동기)가 노출된 경우 각 장치에 의해 노출되는 플랫폼ID는 동일하다.

주소는 장치 및 리소스와의 상호작용을 위해 해당 장치 및 리소스에 접근하기 위한 수단을 정의한다. 주소는 어떤 통신수단을 사용하여 상호작용하는가에 따라 변경될 수 있다. 예로 주소는 URL 또는 IP주소가 될 수 있으며, 경우에 따라 (예: 특정 IP주소를 가지는 특정 장치에서의 특정 리소스에 대한 동작을 수행하는 경우와 같이) 두 가지 모두 필요할 수 있다.

해당 장치 및 리소스가 사용되는 환경에서, 주소가 식별자로 사용되거나 또는 역으로 사용될 수 있다. 예로, URL은 리소스의 식별자로도 사용될 수 있고 URI로 지정될 수도 있다. 리소스는 URI를 사용하여 식별될 수 있으며, URI가 URL인 경우 리소스는 URI를 주소로 사용할 수 있다. 경우에 따라 리소스는 URI와 다른 식별자가 필요할 수 있다. 이 경우 리소스는 속성값으로 식별자를 가질 수 있다. URI가 URL의 형태인 경우, URI는 리소스의 주소로 사용될 수 있다.

## 3) 리소스 모델

리소스 모델은 장치 및 리소스를 표현하는 방법과, CRUDN처럼 장치간 상호 운용성을 제공하는 요소를 정의한다. 리소스 모델의 개념은 기초적 제품 및 상호관계 모델링을 지원하고 컨텍스트에서 상호 운용성에 필요한 의미 정보를 획득한다. 리소스 모델을 구성하는 기본 요소는 개체, 리소스, URI (Uniform Resource Identifier), 리소스 타입, 속성, 표현, 인터페이스, 콜렉션 및 링크이다. 또한 CREATE, RETRIEVE, UPDATE, DELETE 및 NOTIFY를 통해 상호운용성을 제공한다.

개체는 장치를 의미할 수 있는데, 개체를 가시적이고, 상호작용이 잘되고, 조작하기 쉽도록 인식하기 위해 리소스로 추상화한다. 리소스는 개체의 상태를 캡슐화하고 표현한다. URI는 리소스의 식별 및 주소지정을 위해 사용된다. 속성은 "key = value" 쌍이며 리소스 상태를 나타내고, 속성의 스냅샷은 리소스의 표현이 된다. 표현에 대한 특정 뷰와 해당 뷰에 적용 가능한 CRUDN이 인터페이스로 정의된다. 리소스와의 상호 작용은 표현을 포함하는 요청 및 응답으로 수행된다. 리소스 인스턴스는 리소스 타입에서 파생된다. 리소스 사이의 단방향 관계는 링크로 정의된다. 콜렉션은 속성과 링크를 가지고 있는 리소스이다.

일련의 속성을 사용하여 리소스의 상태를 정의 할 수 있다. 이 상태는 해당 리소스에 대한 응답 및 요청에서 각각 적절한 표현을 사용하여 검색 또는 업데이트될 수 있다. 장치에서의 리소스 탐색은 속성이 해당 디바이스의 리소스에 대한 설명 및 참조 값을 가지는 특정 리소스의 표현을 검색하는 것으로 정의 될

수 있다. 리소스는 URI를 가져야한다. URI는 리소스가 생성 될 때 할당되거나 사전에 정의된 리소스 타입에 의해 미리 정의 될 수있다. 다음 그림은 리소스 표현의 예시이다.

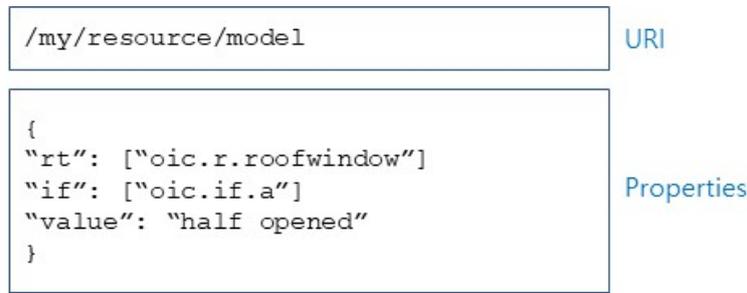


그림 4 - 리소스 표현 예시

#### 4) 탐색

장치간 상호 인식을 위한 탐색은 장치의 엔드포인트 탐색과 리소스의 탐색으로 구분될 수 있다. 장치의 엔드포인트 탐색에 있어서, 엔드포인트는 사용되는 전송 프로토콜에 따라 정해진다. 예로, IPv6와 UDP를 사용하는 CoAP에서, 엔드포인트는 IPv6 주소와 UDP 포트 번호로 식별된다. 각 장치는 CRUDN 요청 및 응답 메시지를 교환할 수 있도록 하나 이상의 엔드포인트와 연결되어야 한다. 메시지가 엔드포인트로 전송되면, 연결된 장치로 메시지가 전달된다. 리소스 탐색에 있어서, 클라이언트는 서버에 대한 적절한 정보를 찾을 수 있다. 이 정보는 리소스, 리소스 타입 또는 탐색이 필요한 다른 요소가 될 수 있다. 탐색 기능을 위해서는 해당 리소스가 탐색될 수 있는 정보를 포함해야 한다. 이 정보는 다른 정보들보다 우선적으로 찾을 수 있어야 한다.

#### 5) CRUDN

CRUDN (CREATE, RETRIEVE, UPDATE, DELETE 및 NOTIFY)은 리소스 조작을 위해 정의되며, 클라이언트가 n개의 서버의 각 리소스에 대해 수행한다. 유효한 CRUDN 작업을 수신하면 요청 대상인 리소스를 가지는 서버는 요청에 포함된 인터페이스에 따라 응답을 생성한다. 해당 자원 유형에 대한 인터페이스가 존재하지 않는 경우 Default 인터페이스를 사용한다.

CRUDN의 세부 설명은 다음과 같다.

- CREATE: 서버에 새로운 리소스 생성을 요청하는데 사용
- RETRIEVE: 리소스의 현재 상태 또는 표현을 요청하는 데 사용
- UPDATE: 리소스 정보를 부분적으로 업데이트하거나 완전히 대체하는데 사용
- DELETE: 리소스의 제거를 요청하는데 사용
- NOTIFY: 상태 변경에 대한 비동기 통지를 요청하는 데 사용

CRUDN은 모두 클라이언트로부터 트리거 되며, 요청, 처리, 응답의 세 단계로 구성된다. 또한 CRUDN은 클라이언트와 서버간 메시지로 전달되며, 세부 파라미터는 추후 정의한다.

#### 6) 장치 관리

장치관리는 장치가 가지는 리소스를 알리는 기능이다. 이를 위해 각 장치는 리소스에 대한 전체 개요 정보를 제공하는 인트로스펙션을 사용한다. 인트로스펙션 구성에 대한 세부적인 정의는 추후 정의한다.

#### 7) 보안

장치의 연결 및 접근에 대한 보안은 다음과 같이 두 가지 측면을 고려할 수 있다.

- Device Onboarding: 개봉한 새로운 기기를 사용자 도메인에 페어링한다. 예를 들어 새로 설치한 센서노드를 기존 제어기에 연결한다.
- Access Control: 장치의 리소스에 대한 요청에 대해 승인/거절 여부를 결정하거나, 각 요청에 대해 사용권한에 따라 필터링한다.

장치간 연결되는 메시지 보안 및 사용자 개인 정보 등 기타 보안 관련 사항은 추후 별도의 문서로 정의한다.

나-16) 스마트팜관제 프레임워크-제2부: 센서 프로파일(13종) ('19 제안, '20 제정)

본 연구에서는 웹기반 스마트팜 센서 프로파일 표준 개발을 위해 CoAP기반 스마트팜 관리 및 제어 프레임워크를 활용한다. CoAP기반 스마트팜 관리 및 제어 프레임워크에서 스마트팜 기자재는 클라이언트-서버 구조로 동작하며, 각 기자재가 가지는 고유의 기능은 리소스가 되며, 리소스는 리소스 모델을 통해 표현되어 외부로 노출될 수 있고, 이를 통해 해당 기자재의 기능을 이용할 수 있다. 따라서 해당 기자재가 어떤 기자재인지를 정의하는 것은 해당 장치의 리소스 모델을 구성하는 것으로 달성될 수 있다. 리소스 모델은 기자재의 특성 및 기능에 따라 자유롭게 정의될 수 있고, CoAP 기반 스마트팜 관리 및 제어 프레임워크에서 호환될 수 있는 JSON, OpenAPI2.0, YAML 등의 포맷으로 정의한다.

리소스 모델을 구성하는 예로, 센서의 경우 장치가 측정하여 생성하는 값이 온도인지 습도인지, 온도가면 섭씨인지 화씨인지, 유효 온도의 범위는 어떤지 등을 표현할 수 있다. 즉, 특정 포맷의 데이터를 GET 메소드를 사용하여 얻을 수 있게 정의한다. 구동기의 경우에도 센서의 경우처럼 구동의 상태를 얻을 수 있는 GET 메소드를 사용할 수 있고, 아울러 구동기를 제어할 수 있는 POST 메소드가 적용될 수 있다. 다음은 CoAP 기반 스마트팜 관리 및 제어프레임워크에 적용할 수 있는 센서 프로파일을 정의한다. 센서 프로파일은 온도, 습도, CO<sub>2</sub>, 일사, 풍향, 풍속, 감우, 광양자, 토양 함수율, 토양 수분 장력, EC, pH, 지온 센서의 13종 센서이며, 각 센서 프로파일은 OCF에서 추진하는 바와 같이 OpenAPI2.0 형태로 정의한다.

1) 온도 센서

온도 센서는 덩고 찬 정도를 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 "C", "F", "K" 중 하나이며, 클라이언트는 쿼리 매개 변수를 사용하여 온도의 단위를 지정할 수 있다. 요청된 단위가 없으면 기본적으로 "C"가 적용되고, 요청된 단위를 지원하지 않는 온도 센서는 에러를 반환한다.

표 1 - 온도 센서 프로파일

URI	Method	Request	Response	Description
/TemperatureResURI	GET	interface	200	센싱온도값
		unit	403	unit 에러
			404	리소스에러
	POST	interface	200	속성설정
value		403	설정에러	

온도 센서 속성은 온도 리소스 유형 (rt), 측정된 온도 데이터 값 (temperature), 온도 데이터의 단위 (unit), 온도 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 온도 데이터의 유효 범위 (range), 범위 내에서 온도의 증분 단위 (step), 온도 센서가 측정한 온도값의 +/- 오차범위를 지정하는 정밀도 (precision), 측정된 온도값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다. 예로, 유효범위(range)가 (0, 10)이고 단계(step)가 2이면 유효한 값은 (0, 2, 4, 6, 8, 10)이다. 또한, 유효범위(range)가 (0, 10)이고 정밀도(precision)가 2이면 유효한 값의 범위는 (-2 ~ 12)이다.

표 2 - 온도 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
temperature	number	M	read write	값
units	string	0	read write	단위
n	string	0	read write	이름
id	string	0	read write	식별자
range	number (min, max)	0	read write	범위
step	number	0	read write	단계
precision	number	0	read write	정밀도
if	array	0	read only	인터페이스

[비고] 센서 속성 표에서 M/O 표기는 다음을 나타낸다. M: 필수항목(mandatory), O: 선택항목(optional). 이하 동일 내용 설명 생략.

온도 센서의 질의(Request) 및 응답(Response) 메시지에는 온도 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 및 실패(403) 메시지로 구분된다.

```

{
  "rt": [
    "oic.r.temperature"
  ],
  "temperature": 0,
  "units": "C",
  "n": "string",
  "id": "string",
  "range": [
    0
  ],
  "step": 0,
  "precision": 0,
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 1 - 온도 센서 메시지(GET/POST) 예시

2) 습도 센서

습도 센서는 습한 정도를 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 백분율이다.

표 3 - 습도 센서 프로파일

URI	Method	Request	Response		Description
/HumidityResURI	GET	interface	200	센싱습도값	습도값 얻기
			404	측정에러	
	POST	interface	200	속성설정	희망 습도 설정
			403	설정에러	
			404	리소스에러	

습도 센서 속성은 습도 리소스 유형 (rt), 희망 습도 데이터 값 (desiredHumidity), 측정된 습도 데이터 값 (humidity), 습도 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 습도 값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 4 - 습도 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
desiredHumidity	number	0	read write	희망습도값
humidity	number	M	read only	측정습도값
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스

습도 센서의 질의(Request) 및 응답(Response) 메시지에는 습도 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.humidity"
  ],
  "desiredHumidity": 0,
  "humidity": 0,
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a"
  ]
}

```

그림 2 - 습도 센서 GET 메시지 예시

```

{
  "desiredHumidity": 0
}

```

그림 3 - 습도 센서 POST 메시지 예시

### 3) CO<sub>2</sub> 센서

CO<sub>2</sub> 센서는 대기에서 이산화탄소의 정도를 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 ppm이다.

표 5 - CO<sub>2</sub> 센서 프로파일

URI	Method	Request	Response	Description
/CO2ResURI	GET	interface	200	센싱 CO <sub>2</sub> 값
			403	유효범위에러
			404	리소스에러
				CO <sub>2</sub> 값 얻기

CO<sub>2</sub> 센서 속성은 CO<sub>2</sub> 리소스 유형 (rt), 측정된 CO<sub>2</sub> 데이터 값 (co2value), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), CO<sub>2</sub> 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 CO<sub>2</sub>값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 6 - CO<sub>2</sub> 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
co2value	number	M	read only	측정CO <sub>2</sub> 값
precision	number	0	read write	정밀도
n	string	0	read write	이름
range	number (min, max)	0	read write	측정값 유효 범위
step	number	0	read write	단계
id	string	0	read write	식별자
if	array	0	read only	인터페이스

CO<sub>2</sub> 센서의 질의(Request) 및 응답(Response) 메시지에는 CO<sub>2</sub> 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.co2"
  ],
  "co2value": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 4 - CO<sub>2</sub> 센서 메시지(GET) 예시

4) 일사 센서

일사 센서는 단위 면적당 입사되는 일사 에너지를 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 W/m<sup>2</sup>이다.

표 7 - 일사 센서 프로파일

URI	Method	Request	Response		Description
/pyranometerResURI	GET	interface	200	센싱일사값	일사값 얻기
			403	유효범위에러	
			404	리소스에러	

일사센서 속성은 일사 리소스 유형 (rt), 측정된 일사 데이터 값 (pyranometervalue), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), 일사 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 일사 데이터값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 8 - 일사 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	O	read only	유형
pyranometervalue	number	M	read only	측정값
precision	number	O	read write	정밀도
n	string	O	read write	이름
range	number (min, max)	O	read write	측정값 유효 범위
step	number	O	read write	단계
id	string	O	read write	식별자
if	array	O	read only	인터페이스

일사 센서의 질의(Request) 및 응답(Response) 메시지에는 일사 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.pyranometer"
  ],
  "pyranometervalue": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 5 - 일사 센서 메시지(GET) 예시

5) 풍향 센서

풍향 센서는 바람의 방향을 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 방향각(°)이다.

표 9 - 풍향 센서 프로파일

URI	Method	Request	Response		Description
/winddirectionResURI	GET	interface	200	센싱풍향값	풍향값 얻기
			403	유효범위에러	
			404	리소스에러	

풍향센서 속성은 풍향 리소스 유형 (rt), 측정된 풍향값 (direction), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), 풍향 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 풍향값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 10 - 풍향 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	O	read only	유형
direction	number	M	read only	측정값
precision	number	O	read write	정밀도
n	string	O	read write	이름
range	number (min, max)	O	read write	측정값 유효 범위
step	number	O	read write	단계
id	string	O	read write	식별자
if	array	O	read only	인터페이스

풍향 센서의 질의(Request) 및 응답(Response) 메시지에는 풍향 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.winddirection"
  ],
  "direction": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 6 - 풍향 센서 메시지(GET) 예시

6) 풍속 센서

풍속 센서는 바람의 세기를 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 초당 속도(m/s)이다.

표 11 - 풍속 센서 프로파일

URI	Method	Request	Response		Description
/windspeedResURI	GET	interface	200	센싱풍속값	풍속값 얻기
			403	유효범위에러	
			404	리소스에러	

풍속 센서 속성은 풍속 리소스 유형 (rt), 측정된 풍속값 (speed), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), 풍속 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 풍속값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 12 - 풍속 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	O	read only	유형
speed	number	M	read only	측정값
precision	number	O	read write	정밀도
n	string	O	read write	이름
range	number (min, max)	O	read write	측정값 유효 범위
step	number	O	read write	단계
id	string	O	read write	식별자
if	array	O	read only	인터페이스

풍속 센서의 질의(Request) 및 응답(Response) 메시지에는 풍속 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.windspeed"
  ],
  "speed": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 7 - 풍속 센서 메시지(GET) 예시

7) 감우 센서

감우 센서는 비가 오는지를 감지하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 on/off(boolean)이다.

표 13 - 감우 센서 프로파일

URI	Method	Request	Response		Description
/raindetectorResURI	GET	interface	200	센싱감우값	감우값 열기
			403	유효범위에러	
			404	리소스에러	

감우 센서 속성은 감우 리소스 유형 (rt), 비가 오는지 여부를 나타내는 감우값 (value), 측정된 비의 양을 나타내는 측정값 (measurement), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), 감우 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 감우값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 14 - 감우 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	O	read only	유형
value	boolean	M	read only	감우값
measurement	number	O	read only	측정값
precision	number	O	read write	정밀도
n	string	O	read write	이름
range	number (min, max)	O	read write	측정값 유효 범위
step	number	O	read write	단계
id	string	O	read write	식별자
if	array	O	read only	인터페이스

감우 센서의 질의(Request) 및 응답(Response) 메시지에는 감우 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.raindetector"
  ],
  "value": true,
  "measurement": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 8 - 감우 센서 메시지(GET) 예시

8) 광양자 센서

광양자센서는 단위 시간당, 단위 면적당 입사되는 광양자 수를 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 광양자속밀도( $\mu\text{mol}/\text{m}^2/\text{s}$ )이다.

표 15 - 광양자 센서 프로파일

URI	Method	Request	Response		Description
/quantumResURI	GET	interface	200	센싱값	광양자값 얻기
			403	유효범위에러	
			404	리소스에러	

광양자센서 속성은 광양자 리소스 유형 (rt), 측정된 광양자 데이터 값 (quantumrvalue), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), 광양자 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 광양자값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 16 - 광양자 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
quantumvalue	number	M	read only	측정값
precision	number	0	read write	정밀도
n	string	0	read write	이름
range	number (min, max)	0	read write	측정값 유효 범위
step	number	0	read write	단계
id	string	0	read write	식별자
if	array	0	read only	인터페이스

광양자 센서의 질의(Request) 및 응답(Response) 메시지에는 광양자 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.quantum"
  ],
  "quantumvalue": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 9 - 광양자 센서 메시지(GET) 예시

### 9) 토양 함수율 센서

토양 함수율 센서는 단위 부피당 토양에 포함된 수분량을 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 수분함유량(%)이다.

표 17 - 토양 함수율 센서 프로파일

URI	Method	Request	Response		Description
/soilmoistureResURI	GET	interface	200	센싱값	토양함수율 값 얻기
			403	유효범위에러	
			404	리소스에러	

토양 함수율 센서 속성은 토양 함수율 리소스 유형 (rt), 측정된 토양 함수율 데이터 값 (soilmoisturevalue), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), 토양 함수율 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 토양 함수율 값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 18 - 토양 함수율 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
soilmoisturevalue	number	M	read only	측정값
precision	number	0	read write	정밀도
n	string	0	read write	이름
range	number (min, max)	0	read write	측정값 유효 범위
step	number	0	read write	단계
id	string	0	read write	식별자
if	array	0	read only	인터페이스

토양 함수율 센서의 질의(Request) 및 응답(Response) 메시지에는 토양 함수율 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.soilmoisture"
  ],
  "soilmoisturevalue": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 10 - 토양 함수율 센서 메시지(GET) 예시

10) 토양 수분 장력 센서

토양 수분 장력 센서는 토양의 수분 장력을 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 수분장력(kPa)이다.

표 19 - 토양 수분 장력 센서 프로파일

URI	Method	Request	Response		Description
/tensiometerResURI	GET	interface	200	센싱값	토양 수분 장력 값 얻기
			403	유효범위에러	
			404	리소스에러	

토양 수분 장력 센서 속성은 토양 수분 장력 리소스 유형 (rt), 측정된 토양 수분 장력 데이터 값 (tensiometervalue), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), 토양 수분 장력 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 토양 수분 장력 값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 20 - 토양 수분 장력 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
tensiometervalue	number	M	read only	측정값
precision	number	0	read write	정밀도
n	string	0	read write	이름
range	number (min, max)	0	read write	측정값 유효 범위
step	number	0	read write	단계
id	string	0	read write	식별자
if	array	0	read only	인터페이스

토양 수분 장력 센서의 질의(Request) 및 응답(Response) 메시지에는 토양 수분 장력 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.tensiometer"
  ],
  "tensiometervalue": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 11 - 토양수분장력센서 메시지(GET) 예시

### 11) EC 센서

EC 센서는 수용액 중의 전기 전도도를 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 전기 전도도(dS/m)이다.

표 21 - EC 센서 프로파일

URI	Method	Request	Response		Description
/ECsensorResURI	GET	interface	200	센싱값	EC 값 얻기
			403	유효범위예러	
			404	리소스에러	

EC 센서 속성은 EC 리소스 유형 (rt), 측정된 EC 데이터 값 (ecvalue), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), EC 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 EC 값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 22 - EC 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	O	read only	유형
ecvalue	number	M	read only	측정값
precision	number	O	read write	정밀도
n	string	O	read write	이름
range	number (min, max)	O	read write	측정값 유효 범위
step	number	O	read write	단계
id	string	O	read write	식별자
if	array	O	read only	인터페이스

EC 센서의 질의(Request) 및 응답(Response) 메시지에는 EC 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.ec"
  ],
  "ecvalue": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 12 - EC 센서 메시지(GET) 예시

12) pH 센서

pH 센서는 수용액 중의 수소 이온 농도(산도)를 측정하는 센서로서, 현재의 측정된 값을 전송한다. 전달하는 값의 단위는 수소이온농도(pH)이다.

표 23 - pH 센서 프로파일

URI	Method	Request	Response		Description
/pHsensorResURI	GET	interface	200	센싱값	pH 값 얻기
			403	유효범위에러	
			404	리소스에러	

pH 센서 속성은 pH 리소스 유형 (rt), 측정된 pH 데이터 값 (phvalue), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), EC 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 pH 값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 24 - pH 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	O	read only	유형
phvalue	number	M	read only	측정값
precision	number	O	read write	정밀도
n	string	O	read write	이름
range	number (min, max)	O	read write	측정값 유효 범위
step	number	O	read write	단계
id	string	O	read write	식별자
if	array	O	read only	인터페이스

pH 센서의 질의(Request) 및 응답(Response) 메시지에는 pH 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```

{
  "rt": [
    "oic.r.sensor.ph"
  ],
  "phvalue": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}

```

그림 13 - pH 센서 메시지(GET) 예시

13) 지온 센서

지온 센서는 근권의 온도(°C)를 측정하는 센서로서, 토양, 배지, 양액 등의 온도를 측정하여 측정된 값을 전달한다. 전달하는 값의 단위는 섭씨(°C)이다.

표 25 - 지온 센서 프로파일

URI	Method	Request	Response		Description
/soiltemperatureResURI	GET	interface	200	센싱값	지온 값 얻기
			403	유효범위에러	
			404	리소스에러	

지온 센서 속성은 지온 리소스 유형 (rt), 측정된 지온 데이터 값 (soiltemperaturevalue), 측정값의 +/- 오차범위를 나타내는 정밀도 (precision), 지온 속성임을 나타내는 사용자 친숙한 이름 (n), 측정값의 유효 범위 (range), 범위 내에서 측정값의 증분 단위 (step), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 지온 값을 전송하고자 하는 인터페이스 (if) 등으로 정의된다.

표 26 - 지온 센서 속성

속성	값유형	M/O	접근유형	설명
rt	array	O	read only	유형
soiltemperaturevalue	number	M	read only	측정값
precision	number	O	read write	정밀도
n	string	O	read write	이름
range	number (min, max)	O	read write	측정값 유효 범위
step	number	O	read write	단계
id	string	O	read write	식별자
if	array	O	read only	인터페이스

지온 센서의 질의(Request) 및 응답(Response) 메시지에는 지온 센서 속성에 정의된 값들이 포함된다. 응답 메시지는 해당 질의에 대한 성공(200) 메시지이다.

```
{
  "rt": [
    "oic.r.sensor.soiltemperature"
  ],
  "soiltemperaturevalue": 0,
  "precision": 0,
  "n": "string",
  "range": [
    0
  ],
  "step": 0,
  "id": "string",
  "if": [
    "oic.if.baseline"
  ]
}
```

그림 14 - 지온 센서 메시지(GET) 예시

나-17) 스마트팜관제 프레임워크-제3부: 개폐기 프로파일 (1종) ('19 제안, '20 제정)

본 연구에서는 웹기반 스마트팜 구동기(창개폐기) 프로파일 표준 개발을 위해 CoAP기반 스마트팜 관리 및 제어 프레임워크를 활용한다. CoAP 기반 스마트팜 관리 및 제어 프레임워크에 적용할 수 있는 창 개폐기 프로파일을 정의한다. 창 개폐기는 스마트팜 시설내 창을 개폐하는 구동기의 하나로서, 창을 여닫는 동작을 수행하고 결과 상태를 전송한다. 창 상태 알람 설정을 통해 창 상태 알람을 활성화 또는 비활성화 한다. 창 개폐기 프로파일은 OCF에서와 추진하는 바와 같이 OpenAPI2.0 형태로 정의한다.

표 1 - 창 개폐기 프로파일

URI	Method	Request	Response	Description
/WindowSwitchgearResURI	GET	interface	200	창상태값 창 상태값 얻기
	POST	interface	200	속성설정 창 상태 알람 설정
value				

창 개폐기 속성은 창 개폐기 리소스 유형 (r), 창의 여닫은 상태 (openState), 창의 개방 지속시간 (openDuration), 창의 열림 알람 (openAlarm), 창 개폐기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 창 개폐기 상태값을 전송하고자 하는 인터페이스(if) 등으로 정의된다.

[비고] 창개폐기 속성 표에서 M/O 표기는 다음을 나타낸다.

- M: 필수항목(mandatory)
- O: 선택항목(optional)

표 2 - 창개폐기 속성

속성	값유형	M/O	접근유형	설명
rt	array	O	read only	유형
openDuration	number	O	read write	개방지속시간
openState	string	M	read only	개폐상태
openAlarm	boolean	O	read write	개폐상태알람
n	string	O	read write	이름
id	string	O	read write	식별자
if	array	O	read only	인터페이스
openAlarm	boolean	M	read write	개폐상태알람

창 개폐기의 질의(Request) 및 응답(Response) 메시지는 온도 센서 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.windowswitchgear"
  ],
  "openDuration": "string",
  "openState": "Open",
  "openAlarm": true,
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a"
  ]
}
    
```

그림 1- 창 개폐기 GET 메시지 예시

```

{
  "openAlarm": true
}
    
```

그림 2 - 창 개폐기 GET 메시지 예시

## 나-18) 스마트팜관제 프레임워크-제4부: 돈사 사양관리기(11종) ('20 제안, '21 제정)

### (가) 연구 수행 개요

본 표준은 CoAP 기반 스마트팜(Smart Farm) 관리와 제어 프레임워크 베이스 프로토콜 (TTAS.KO-10.1243-Part1)을 기반으로 스마트팜 돈사에 적용되는 사양관리기들의 프로파일을 정의한다. 이를 위해 현재 Open Connectivity Foundation (OCF)에서 추진되고 있는 내용을 참고하여 스마트팜에 적합한 프로파일을 정의한다.

이 표준은 스마트팜 돈사 사양관리 제어를 위해 다음과 같은 장비들에 대해 OpenAPI2.0 기반의 프로파일을 제공한다.

- 임신돈군사급이기, 모돈자동급이기, 개체식별기, 컴퓨터엑상급이기, 사료믹스급이기, 출하돈선별기, 사료빈관리기, 음수관리기, 체중측정기, 발정탐지기, 분만탐지기 등 11종

### (나) 연구 주요 내용

CoAP기반 스마트팜 관리 및 제어 프레임워크에서 스마트팜 기자재는 클라이언트-서버 구조로 동작하며, 각 기자재가 가지는 고유의 기능은 리소스가 되며, 리소스는 리소스 모델을 통해 표현되어 외부로 노출될 수 있고, 이를 통해 해당 기자재의 기능을 이용할 수 있다.

따라서 해당 기자재가 어떤 기자재인지를 정의하는 것은 해당 장치의 리소스 모델을 구성하는 것으로 달성될 수 있다. 리소스 모델은 기자재의 특성 및 기능에 따라 자유롭게 정의될 수 있고, CoAP 기반 스마트팜 관리 및 제어 프레임워크에서 호환될 수 있는 JSON, OpenAPI2.0, YAML 등의 포맷으로 정의한다.

리소스 모델을 구성하는 예로, 센서의 경우 장치가 측정하여 생성하는 값이 온도인지 습도인지, 온도이면 섭씨인지 화씨인지, 유효 온도의 범위는 어떤지 등을 표현할 수 있다. 즉, 특정 포맷의 데이터를 GET 메소드를 사용하여 얻을 수 있게 정의한다. 구동기의 경우에도 센서의 경우처럼 구동의 상태를 얻을 수 있는 GET 메소드를 사용할 수 있고, 아울러 구동기를 제어할 수 있는 POST 메소드가 적용될 수 있다.

다음은 CoAP 기반 스마트팜 관리 및 제어 프레임워크에 적용할 수 있는 스마트팜 돈사 사양관리 장비 프로파일을 정의한다. 사양관리 장비 프로파일은 임신돈군사급이기, 모돈자동급이기, 개체식별기, 컴퓨터엑상급이기, 사료믹스급이기, 출하돈선별기, 사료빈관리기, 음수관리기, 체중측정기, 발정탐지기, 분만탐지기의 11종 기기이며, 각 장비 프로파일은 OCF에서 추진하는 바와 같이 OpenAPI2.0 형태로 정의한다.

## ■ 임신돈 군사급이기

임신돈 군사급이기는 임신사에 사육중인 모돈에게 사료를 개체별로 공급하는 장치로서, 개체별로 급이량, 급이횟수, 급이시간 등을 조절하여 사료를 공급하고, 결과 상태를 전송한다. 또한 설정을 통해 장치의 기능을 제어한다.

URI	Method	Request	Response		Description
/GroupedSowFeeder	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정에러		

임신돈 군사급이기 속성은 임신돈 군사급이기 리소스 유형(rt), 임신돈 군사급이기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 임신돈 군사급이기 상태값을 전송하고자 하는 인터페이스(if), 각각의 임신돈 개체를 구분할 수 있는 개체ID(p\_id), 임신돈 군사급이기를 식별하기 위한 급이기ID(b\_id), 급이횟수(f\_count), 급이량(f\_amount), 급이량설정(s\_amount), 급이일시(f\_datetime), 급이량 단위(f\_unit), 급이잔량(c\_amount) 등으로 정의된다.

[비고 1] 센서 속성 표에서 M/O 표기는 다음을 나타낸다.

- M: 필수항목(mandatory)

- O: 선택항목(optional). 이하 동일 내용 생략.

[비고 2] 급이일시(f\_datetime)와 같이 날짜 및 시간에 관련한 데이터 표기는 ISO8601을 따른다. 이하 동일 내용 생략.

[비고 3] 급이량 단위(f\_unit)와 같이 무게에 관한 데이터의 단위는 "kg", "g", "lb", "oz" 중에 택일한다. 이하 동일 내용 생략.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
p_id	string	M	read only	개체ID
b_id	string	M	read only	급이기ID
f_count	number	0	read write	급이횟수
f_amount	number	M	read write	급이량
s_amount	number	0	read write	급이설정량
f_datetime	string	0	read write	급이일시
f_unit	string	0	read write	급이량 단위
c_amount	number	0	read only	급이잔량

임신돈 군사급이기의 질의(Request) 및 응답(Response) 메시지는 임신돈 군사급이기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.groupedsowfeeder"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a"
  ]
  "p_id": "string",
  "b_id": "string",
  "f_count": 0,
  "f_amount": 2.8,
  "s_amount": 3.5,
  "f_datetime": "string",
  "f_unit": "Kg",
  "c_amount": 100
}
```

```
{
  "p_id": "string",
  "b_id": "string",
  "s_amount": 3.5
}
```

### ■ 모든 자동급이기

모든 자동급이기는 분만사의 분만틀이나 임신사의 스톨에 설치하여 분만사에 사육 중인 포유모돈 및 분만대기돈에게 자동으로 사료를 급이하는 장치로서, 급이횟수, 급이량, 급이시간 등을 조절하여 사료를 공급하고, 결과상태를 전송한다. 또한 설정을 통해 장치의 기능을 제어한다.

URI	Method	Request	Response	Description	
/AutomaticSowFeeder	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정에러		

모든 자동급이기 속성은 모든 자동급이기 리소스 유형(rt), 모든 자동급이기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 모든 자동급이기 상태값을 전송하고자 하는 인터페이스(if), 모든 자동급이기를 식별하기 위한 급이기ID(b\_id), 급이횟수(f\_count), 급이량 (f\_amount), 급이량설정(s\_amount), 급이일시(f\_datetime), 급이량 단위(f\_unit), 급이잔량(c\_amount) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
b_id	string	M	read only	급이기ID
f_count	number	0	read write	급이횟수
f_amount	number	M	read write	급이량
s_amount	number	0	read write	급이설정량
f_datetime	string	0	read write	급이일시
f_unit	string	0	read write	급이량 단위
c_amount	number	0	read only	급이잔량

모든자동급이기의 질의(Request) 및 응답(Response) 메시지에는 모든 자동급이기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.automaticsowfeeder"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "b_id": "string",
  "f_count": 0,
  "f_amount": 1.0,
  "s_amount": 1.5,
  "f_datetime": "string",
  "f_unit": "kg",
  "c_amount": 20
}
```

```
{
  "b_id": "string",
  "s_amount": 1.5
}
```

### ■ 개체식별기

개체식별기는 돈사내 각 개체에 부착된 전자이표 등과 같이 개체의 식별정보를 인식하는 장치로서, 돈사내 각 개체를 구분하고, 해당정보가 필요한 장치로 전송한다.

URI	Method	Request	Response		Description
/PigIdStation	GET	interface	200	정상	장치값 얻기
			403	reading에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정어러		

개체식별기 속성은 개체식별기 리소스 유형(rt), 개체식별기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 개체식별기 장치값을 전송하고자 하는 인터페이스(if), 개체식별기가 설치된 농장을 식별하는 농장ID(barn\_id), 농장내 각 개체를 구분하는 개체ID(p\_id), 각 개체에 부착된 식별장치정보를 나타내는 개체식별자ID(tag\_id), 인식된 개체식별자가 유효한값인지를 나타내는 개체ID유효성(reading) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
----	-----	-----	------	----

rt	array	0	read onl	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
p_id	string	0	read write	개체ID
barn_id	string	0	read write	농장ID
tag_id	number	M	read only	개체식별장치ID
reading	boolean	M	read only	개체ID유효성

개체식별기기의 질의(Request) 및 응답(Response) 메시지는 개체식별기기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.pigidstation"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "p_id": "string",
  "barn_id": "string",
  "tag_id": 001001,
  "reading": true
}
```

```
{
  "p_id": "string",
  "barn_id": "string",
  "tag_id": 001001
}
```

### ■ 컴퓨터액상급이기

컴퓨터액상급이기는 키친룸에서 사료와 음수를 모두 배합하여 공급하는 장치로서 돼지의 모든 사육구간에 적용할 수 있다. 또한 전체적인 사료량을 측정하며, 사료급이량 확인이 가능하고 센서가 잔류사료를 감지하여 섭취속도를 조절하여 신선도가 떨어지거나 부패하는 것을 방지할 수 있다.

URI	Method	Request	Response	Description
/ComputerLiquidFeeder	GET	interface	200	정상
		unit	403	unit에러
			404	리소스에러
	POST	interface	200	속성설정
value		403	설정에러	

컴퓨터액상급이기 속성은 컴퓨터액상급이기 리소스 유형(rt), 컴퓨터액상급이기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 컴퓨터액상급이기 상태값을 전송하고자 하는 인터페이스(if), 각각의 급이기를 식별하기 위한 급이기ID(b\_id), 급이사료정보(f\_info), 급이횟수(f\_count), 급이량(f\_amount), 급이량설정(s\_amount), 급이일시(f\_datetime), 급이량 단위(f\_unit), 급이잔량(c\_amount) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
b_id	string	M	read only	급이기ID
f_info	string	0	read write	급이사료정보
f_count	number	0	read write	급이횟수

f_amount	number	M	read write	급이량
s_amount	number	0	read write	급이설정량
f_datetime	string	0	read write	급이일시
f_unit	string	0	read write	급이량 단위
c_amount	number	0	read only	급이잔량

컴퓨터액상급이기의 질의(Request) 및 응답(Response) 메시지에는 컴퓨터액상급이기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    oic.r.computerliquidfeeder"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "b_id": "string",
  "f_info": "string",
  "f_count": 0,
  "f_amount": 50,
  "s_amount": 100,
  "f_datetime": "string",
  "f_unit": "kg",
  "c_amount": 1000
}
```

```
{
  "b_id": "string",
  "s_amount": 100
}
```

### ■ 사료믹스급이기

사료믹스급이기는 돈방에 설치되어 건식사료와 물을 함께 급이하는 장치로서, 물과 사료를 독립적으로 공급한다.

URI	Method	Request	Response		Description
/MixFeeder	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정어러		

사료믹스급이기 속성은 사료믹스급이기 리소스 유형(rt), 사료믹스급이기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 사료믹스급이기 상태값을 전송하고자 하는 인터페이스(if), 각각의 급이기를 식별하기 위한 급이기ID(b\_id), 급이횟수(f\_count), 급이량(f\_amount), 급이량설정(s\_amount), 급이일시(f\_datetime), 급이량 단위(f\_unit), 급이잔량(c\_amount), 물공급량(w\_amount), 물공급량설정(s\_water), 물공급일시(w\_datetime), 물과 사료의 배합비율을 나타내는 급이배합율(f\_rate) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
b_id	string	M	read only	급이기ID
f_count	number	0	read write	급이횟수
f_amount	number	M	read write	급이량

w_amount	number	M	read write	물공급량
s_amount	number	0	read write	급이설정량
s_water	number	0	read write	물공급설정량
f_datetime	string	0	read write	급이일시
w_datetime	string	0	read write	물공급일시
f_unit	string	0	read write	급이량 단위
c_amount	number	0	read only	급이잔량
f_rate	string	0	read write	급이배합율

사료믹스급이기의 질의(Request) 및 응답(Response) 메시지는 사료믹스급이기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.mixfeeder"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "b_id": "string",
  "f_count": 0,
  "f_amount": 10,
  "w_amount": 10,
  "s_amount": 40,
  "s_water": 40,
  "f_datetime": "string",
  "w_datetime": "string",
  "f_unit": "kg",
  "f_rate": "1:1",
  "c_amount": 1000
}
```

```
{
  "b_id": "string",
  "s_water": 40,
  "s_amount": 40
}
```

■ 출하돈선별기

출하돈선별기는 개체별 체중을 측정하여 개체를 설정기준 체중에 따라 선별하는 장치로서, 적합개체수 설정을 통해 출하 또는 돈군 분리시 활용될 수 있다.

URI	Method	Request	Response	Description
/PigSelector	GET	interface	200 정상	장치값 얻기
		unit	403 unit에러	
			404 리소스에러	
	POST	interface	200 속성설정	속성 설정
value		403 설정에러		

출하돈선별기 속성은 출하돈선별기 리소스 유형(rt), 출하돈선별기 속성을 나타내는 사용자 친숙한 이름(n), 해당 속성 인스턴스를 구분할 수 있는 식별자(id), 출하돈선별기 장치값을 전송하고자 하는 인터페이스(if), 개체를 식별하는 개체ID(p\_id), 체중의 유효범위(range), 개체의 측정체중(p\_weight), 설정체중(s\_weight), 측정된 개체의 누적체중(c\_weight), 측정개체수(p\_count), 설정개체수(s\_count), 개체체중측정일시(p\_datetime), 체중측정단위(p\_unit) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자

if	array	0	read only	인터페이스
p_id	string	0	read write	개체ID
range	numner(min,max)	0	read write	유효체중범위
p_weight	number	M	read only	측정체중
s_weight	number	M	read write	설정체중
c_weight	number	0	read write	누적체중
p_count	number	0	read write	측정개체수
s_count	number	0	read write	설정개체수
p_datetime	string	0	read write	측정일시
p_unit	string	0	read write	측정단위

출하돈선별기기의 질의(Request) 및 응답(Response) 메시지에는 출하돈선별기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.pigselector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "p_id": "string",
  "range": [ 5, 300 ],
  "p_weight": 0,
  "s_weight": 10,
  "c_weight": 30,
  "p_count": 0,
  "s_count": 40,
  "p_datetime": "string",
  "p_unit": "kg"
}
```

```
{
  "s_weight": 10,
  "s_count": 40
}
```

### ■ 사료빈관리기

사료빈관리기는 사료빈의 하부에 전자저울(로드셀)을 설치하여 무게를 측정하는 장치로서 사료의 잔량을 확인할 수 있다.

URI	Method	Request	Response		Description
/BinManager	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정어러		

사료빈관리기 속성은 사료빈관리기 리소스 유형(rt), 사료빈관리기 속성을 나타내는 사용자 친숙한 이름(n), 해당 속성 인스턴스를 구분할 수 있는 식별자(id), 사료빈관리기 상태값을 전송하고자 하는 인터페이스(if), 사료빈ID(b\_id), 사료정보(f\_info), 사료배출횟수(f\_count), 사료배출량(f\_amount), 사료배출량설정(s\_amount), 사료배출일시(f\_datetime), 사료배출 설정일시(s\_datetime), 사료배출 단위(f\_unit), 사료잔량(c\_amount) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름

id	string	0	read write	식별자
if	array	0	read only	인터페이스
b_id	string	M	read only	사료빈ID
f_info	string	0	read write	사료정보
f_count	number	0	read write	사료배출횟수
f_amount	number	M	read write	사료배출량
s_amount	number	0	read write	사료배출설정량
f_datetime	string	0	read write	사료배출일시
s_datetime	string	0	read write	사료배출설정일시
f_unit	string	0	read write	사료배출량 단위
c_amount	number	0	read only	사료잔량

사료빈관리기의 질의(Request) 및 응답(Response) 메시지에는 사료빈관리기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.binmanager"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "b_id": "string",
  "f_info": "string",
  "f_count": 0,
  "f_amount": 50,
  "s_amount": 100,
  "f_datetime": "string",
  "s_datetime": "string",
  "f_unit": "kg",
  "c_amount": 1000
}
```

```
{
  "b_id": "string",
  "s_datetime": "string",
  "s_amount": 100
}
```

### ■ 음수관리기

음수관리기는 측사에 공급되는 음수량을 측정하는 장치로서, 공급되는 음수량을 설정된 주기에 맞추어 측정하여 측사에 음수가 정상적으로 공급되고 있는지 확인할 수 있다.

URI	Method	Request	Response		Description
/WaterManager	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정에러		

음수관리기 속성은 음수관리기 리소스 유형(rt), 음수관리기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 음수관리기 상태값을 전송하고자 하는 인터페이스(if), 음수관리기ID(w\_id), 음수공급량(w\_amount), 음수공급량측정일시(w\_datetime), 음수공급단위(w\_unit), 음수공급제어(w\_control) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자

if	array	0	read only	인터페이스
w_id	string	M	read only	음수관리기ID
w_amount	number	M	read write	음수공급량
w_datetime	string	0	read write	음수공급일시
w_unit	string	0	read write	음수공급유량단위
w_control	string	0	read write	음수공급제어

음수관리기의 질의(Request) 및 응답(Response) 메시지에는 음수관리기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.watermanager"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "w_id": "string",
  "w_amount": 50,
  "w_datetime": "string",
  "w_unit": "L",
  "w_control": "open"
}
```

```
{
  "w_id": "string",
  "w_unit": "L",
  "w_control": "close"
}
```

### ■ 체중측정기

체중측정기는 돼지의 체중을 측정하는 장치로서, 측정된 체중값을 전송한다.

URI	Method	Request	Response	Description
/WeightMeter	GET	interface	200 정상	장치값 얻기
		unit	403 unit에러	
			404 리소스에러	
	POST	interface	200 속성설정	속성 설정
value		403 설정에러		

체중측정기 속성은 체중측정기 리소스 유형(rt), 체중측정기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 체중측정기 장치값을 전송하고자 하는 인터페이스(if), 개체를 식별하는 개체ID(p\_id), 체중의 유효범위(range), 개체의 측정체중(p\_weight), 개체체중측정일시 (p\_datetime), 체중측정단위(p\_unit) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
p_id	string	0	read write	개체ID
range	numner(min,max)	0	read write	유효체중범위
p_weight	number	M	read only	측정체중
p_datetime	string	0	read write	측정일시
p_unit	string	0	read write	측정단위

체중측정기의 질의(Request) 및 응답(Response) 메시지에는 체중측정기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.weightmeter"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "p_id": "string",
  "range": [ 5, 300 ],
  "p_weight": 0,
  "p_datetime": "string",
  "p_unit": "kg"
}
```

```
{
  "range": [ 5, 300 ],
  "p_unit": "kg"
}
```

### ■ 발정탐지기

발정탐지기는 개체의 발정을 체크하는 장치로서, 응돈의 모든 개체별 접근 횟수, 체온, 활동량 등을 측정하여 탐지한다.

URI	Method	Request	Response	Description
/EstrousDetector	GET	interface	200 정상	장치값 얻기
		unit	403 unit에러	
			404 리소스에러	
	POST	interface	200 속성설정	속성 설정
value		403 설정에러		

발정탐지기 속성은 발정탐지기 리소스 유형(rt), 발정탐지기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 발정탐지기 장치값을 전송하고자 하는 인터페이스(if), 모든 개체를 식별하는 개체ID(p\_id), 응돈 등 특정 개체의 접근 횟수(a\_count), 모든 개체의 측정 체온 (p\_temperature), 개체 측정 체온의 단위(pt\_unit), 모든 개체의 움직임 횟수(pm\_count) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자

if	array	0	read only	인터페이스
p_id	string	M	read write	개체ID
a_count	number	0	read write	접근횟수
p_temperature	number	0	read only	측정체온
pt_unit	string	0	read write	체온단위
pm_count	number	0	read write	움직임 횟수

발정탐지기의 질의(Request) 및 응답(Response) 메시지에는 발정탐지기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.estrousdetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "p_id": "string",
  "a_count": 10,
  "p_temperature": 25,
  "pt_unit": "C",
  "pm_count": 100
}
```

```
{
  "a_count": 0,
  "pt_unit": "C"
}
```

### ■ 분만탐지기

분만탐지기는 모든 개체의 분만을 탐지하는 장치로서, 꼬리의 움직임, 빛, 활동량, 체온 등을 측정하여 탐지한다.

URI	Method	Request	Response	Description
/FarrowingDetector	GET	interface	200 정상	장치값 얻기
		unit	403 unit에러	
			404 리소스에러	
	POST	interface	200 속성설정	속성 설정
value		403 설정에러		

분만탐지기 속성은 분만탐지기 리소스 유형(rt), 분만탐지기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 분만탐지기 장치값을 전송하고자 하는 인터페이스(if), 개체를 식별하는 개체ID(p\_id), 꼬리 등 기기 장착 부위의 움직임 횟수(pm\_count), 빛의 세기(light), 빛 세기의 단위(l\_unit), 개체의 측정 체온(p\_temperature), 측정 체온의 단위(pt\_unit) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
p_id	string	M	read write	개체ID
pm_count	number	0	read write	움직임 횟수
light	number	0	read only	측정조도
l_unit	string	0	read write	조도단위
p_temperature	number	0	read only	측정체온
pt_unit	string	0	read write	체온단위

분만탐지기의 질의(Request) 및 응답(Response) 메시지에는 분만탐지기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.farrowingdetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "p_id": "string",
  "pm_count": 100,
  "light": 1000,
  "l_unit": "lux",
  "p_temperature": 25,
  "pt_unit": "C"
}
```

```
{
  "pm_count": 0,
  "l_unit": "lux",
  "pt_unit": "C"
}
```

**나-19) 스마트팜관제 프레임워크-제5부: 돈사환경관리기(10종)(’21제안, ’22제정)**

**(가) 연구 수행 개요**

본 표준은 CoAP 기반 스마트팜(Smart Farm) 관리와 제어 프레임워크 베이스 프로토콜 (TTAS.KO-10.1243-Part1)을 기반으로 스마트팜 돈사에 적용되는 환경관리기들의 프로파일을 정의한다. 이를 위해 현재 Open Connectivity Foundation (OCF)에서 추진되고 있는 내용을 참고하여 스마트팜에 적합한 프로파일을 정의한다.

이 표준은 스마트팜 돈사 환경 제어를 위해 다음과 같은 장비들에 대해 OpenAPI2.0 기반의 프로파일을 제공한다.

- 온습도감지기, CO<sub>2</sub>감지기, 환기팬, 보온등, CCTV, 악취감지기, 정전감지기, 누전감지기, 아크감지기, 조도감지기

**(나) 연구 주요 내용**

CoAP기반 스마트팜 관리 및 제어 프레임워크에서 스마트팜 기자재는 클라이언트-서버 구조로 동작하며, 각 기자재가 가지는 고유의 기능은 리소스가 되며, 리소스는 리소스 모델을 통해 표현되어 외부로 노출될 수 있고, 이를 통해 해당 기자재의 기능을 이용할 수 있다.

따라서 해당 기자재가 어떤 기자재인지를 정의하는 것은 해당 장치의 리소스 모델을 구성하는 것으로 달성될 수 있다. 리소스 모델은 기자재의 특성 및 기능에 따라 자유롭게 정의될 수 있고, CoAP 기반 스마트팜 관리 및 제어 프레임워크에서 호환될 수 있는 JSON, OpenAPI2.0, YAML 등의 포맷으로 정의한다.

리소스 모델을 구성하는 예로, 센서의 경우 장치가 측정하여 생성하는 값이 온도인지 습도인지, 온도이면 섭씨인지 화씨인지, 유효 온도의 범위는 어떤지 등을 표현할 수 있다. 즉, 특정 포맷의 데이터를 GET 메소드를 사용하여 얻을 수 있게 정의한다. 구동기의 경우에도 센서의 경우처럼 구동의 상태를 얻을 수 있는 GET 메소드를 사용할 수 있고, 아울러 구동기를 제어할 수 있는 POST 메소드가 적용될 수 있다.

다음은 CoAP 기반 스마트팜 관리 및 제어 프레임워크에 적용할 수 있는 스마트팜 돈사 환경관리 장비 프로파일을 정의한다. 환경관리 장비 프로파일은 온습도감지기, CO<sub>2</sub>감지기, 환기팬, 보온등, CCTV, 악취감지기, 정전감지기, 누전감지기, 아크감지기, 조도감지기의 10종 기기이며, 각 장비 프로파일은 OCF에서 추진하는 바와 같이 OpenAPI2.0 형태로 정의한다.

**■ 온습도감지기**

온습도감지기는 돈사내 온도와 습도를 측정하는 장치로서 현재의 측정된 값을 전송한다.

URI	Method	Request	Response		Description
/TemperatureHumidity	GET	interface	200	측정값	측정값 얻기
		unit	403	unit 에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정어러		

온습도감지기 속성은 온습도감지기 리소스 유형 (rt), 온습도감지기 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 온도값 및 습도값을 전송하고자 하는 인터페이스 (if), 측정된 온도 데이터 값 (temperature), 온도 데이터의 단위 (t\_unit), 설정희망 온도값 (s\_temperature), 온도 데이터의 유효 범위 (t\_range), 측정된 습도 데이터 값 (humidity), 습도 데이터의 단위 (h\_unit), 설정희망 습도값(s\_humidity), 습도 데이터의 유효 범위 (h\_range) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
temperature	number	M	read only	측정온도값
t_unit	string	0	read write	온도단위
s_temperature	number	0	read write	희망온도설정
t_range	number (min, max)	0	read write	온도유효범위

humidity	number	M	read only	측정습도값
h_unit	string	0	read write	습도단위
s_humidity	number	0	read write	희망습도설정
h_range	number (min, max)	0	read write	습도유효범위

온습도감지기의 질의(Request) 및 응답(Response) 메시지에는 온습도감지기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.temperaturehumidity"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "temperature": 0,
  "t_unit": "C",
  "s_temperature": 30,
  "t_range": [ 0, 50 ],
  "humidity": 50,
  "h_unit": "%",
  "s_humidity": 50,
  "h_range": [ 30, 60 ]
}
```

```
{
  "s_temperature": 30,
  "t_range": [ 0, 50 ],
  "s_humidity": 50,
  "h_range": [ 30, 60 ]
}
```

■ CO<sub>2</sub>감지기

CO<sub>2</sub>감지기는 돈사 내 이산화탄소의 정도를 측정하는 장치로서, 현재 측정된 CO<sub>2</sub>값을 전송한다.

URI	Method	Request	Response	Description
/CO <sub>2</sub> Detector	GET	interface	200	측정값
		unit	403	unit 에러
			404	리소스에러
	POST	interface	200	속성설정
value		403	설정어러	

CO<sub>2</sub>감지기 속성은 CO<sub>2</sub>감지기 리소스 유형 (rt), CO<sub>2</sub>감지기 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 측정된 CO<sub>2</sub> 데이터값을 전송하고자 하는 인터페이스 (if), 측정된 CO<sub>2</sub> 데이터 값 (co2), CO<sub>2</sub> 데이터 단위 (co2\_unit), 설정희망 CO<sub>2</sub>값(s\_co2), CO<sub>2</sub> 데이터의 유효 범위 (co2\_range) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
co2	number	M	read only	CO <sub>2</sub> 측정값
co2_unit	string	0	read write	CO <sub>2</sub> 단위
s_co2	number	0	read write	희망값 설정
co2_range	number (min, max)	0	read write	유효범위

CO<sub>2</sub>감지기의 질의(Request) 및 응답(Response) 메시지에는 CO<sub>2</sub>감지기 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.co2detector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "co2": 0,
  "co2_unit": "ppm",
  "s_co2": 700,
  "co2_range": [ 0, 1000 ]
}

```

```

{
  "s_co2": 7000,
  "co2_range": [ 0, 1000 ]
}

```

■ 환기팬

환기팬은 외부의 공기를 유입하여 돈사내 공기를 환기하기 위한 장치로서, 돈사내 오염된 공기를 외부로 배출할 수 있다.

URI	Method	Request	Response		Description
/VentilationFan	GET	interface	200	측정값	측정값 얻기
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
		value	403	설정어러	

환기팬 속성은 환기팬 리소스 유형 (rt), 환기팬 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 환기팬 상태값을 전송하고자 하는 인터페이스 (if), 환기팬 동작상태 (op\_mode), 환기팬 동작 방향 (direction), 환기팬 동작 단계 (speed), 환기팬 동작단계 범위 (vfs\_range) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
op_mode	string	M	read write	동작상태
direction	string	M	read write	동작방향
speed	number	0	read write	동작단계
vfs_range	number (min, max)	0	read write	동작단계범위

환기팬의 질의(Request) 및 응답(Response) 메시지에는 환기팬 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.ventilationfan"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "op_mode": "on",
  "direction": "cw",
  "speed": 3,
  "vfs_range": [ 1, 7 ]
}

```

```

{
  "op_mode": "on",
  "direction": "ccw",
  "speed": 3,
  "vfs_range": [ 1, 7 ]
}

```

■ 보온등

보온등은 돈사내 온도 및 습도를 조절하기 위한 장치로서, 현재 상태를 전송한다.

URI	Method	Request	Response		Description
/HeatLamp	GET	interface	200	상태값	상태값 얻기
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
		value	403	설정어러	

보온등 속성은 보온등 리소스 유형 (rt), 보온등 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 보온등 상태값을 전송하고자 하는 인터페이스 (if), 보온등 동작상태 (switch) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
switch	string	M	read write	동작상태

보온등의 질의(Request) 및 응답(Response) 메시지에는 보온등 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.heatlamp"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "switch": "on"
}

```

```

{
  "switch": "off"
}

```

■ CCTV

CCTV는 돈사내 환경을 관측하기 위한 장치로서, 현재 관측되는 영상을 전송한다.

URI	Method	Request	Response		Description
/CCTVcam	GET	interface	200	상태값	상태값 얻기
			404	리소스에러	

	POST	interface	200	속성설정	속성 설정
		value	403	설정예러	

CCTV 속성은 CCTV 리소스 유형 (rt), CCTV 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), CCTV 상태값을 전송하고자 하는 인터페이스 (if), CCTV 팬 상태 (pan), CCTV 틸트 상태 (tilt), CCTV 줌상태 (zoomFactor), CCTV 동작상태 (switch) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
pan	number	0	read write	팬
tilt	number	0	read write	틸트
zoomFactor	string	0	read write	줌
switch	string	M	read write	동작상태

CCTV의 질의(Request) 및 응답(Response) 메시지에는 CCTV 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.a.cctvcam"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "pan": 0.0,
  "tilt": 0.0,
  "zoomFactor": "1x",
  "switch": "on"
}
```

```
{
  "pan": 10.0,
  "tilt": -10.0,
  "zoomFactor": "4x",
  "switch": "on"
}
```

■ 악취감지기

악취감지기는 돈사내 악취를 감지하는 장치로서, 현재 측정된 값을 전송한다.

URI	Method	Request	Response	Description
/OdorDetector	GET	interface	200	상태값
			404	리소스에러
	POST	interface	200	속성설정
		value	403	설정예러
				상태값 얻기
				속성 설정

악취감지기 속성은 악취감지기 리소스 유형 (rt), 악취감지기 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 악취감지기가 측정값을 전송하고자 하는 인터페이스 (if), 측정하는 악취의 원인물 성분 (contaminanttype), 악취감지기가 측정한 오염성분의 정도 (contaminantvalue), 설정하고자 하는 악취 정도(s\_contaminantvalue), 악취 정도의 유효 범위(range) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
contaminanttype	string	0	read write	악취유형

contaminantvalue	number	M	read write	악취정도
s_contaminantvalue	number	0	read write	설정악취정도
range	number(min, max)	0	read write	악취정도범위

악취감지기의 질의(Request) 및 응답(Response) 메시지에는 악취감지기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.odordetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "contaminanttype": "CO",
  "contaminantvalue": 10,
  "s_contaminantvalue": 100,
  "range": [0,500]
}
```

```
{
  "contaminanttype": "NH3",
  "s_contaminantvalue": 100,
  "range": [0,500]
}
```

### ■ 정전감지기

정전감지기는 돈사내 전기 공급 단절을 감지하는 장치로서, 측정된 값을 전송한다.

URI	Method	Request	Response	Description
/PowerFailureDetector	GET	interface	200	상태값
		unit	403	unit 에러
			404	리소스에러
	POST	interface	200	속성설정
value		403	설정어러	

정전감지기 속성은 정전감지기 리소스 유형 (rt), 정전감지기 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 정전감지기가 측정값을 전송하고자 하는 인터페이스 (if), 정전감지기가 측정하는 전원 상태 (powerstate), 정전감지기가 알람을 하는 시간간격 (alarminterval), 알람간격 단위(interval\_unit) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
powerstate	string	M	read only	전원상태
alarminterval	number	0	read write	알람간격
interval_unit	string	0	read write	알람간격단위

정전감지기의 질의(Request) 및 응답(Response) 메시지에는 정전감지기 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.powerfailuredetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "powerstate": "on",
  "alarminterval": 10,
  "interval_unit": "sec"
}

```

```

{
  "alarminterval": 5,
  "interval_unit": "min"
}

```

■ 누전감지기

누전감지기는 돈사 시설의 누전을 감지하는 장치로서, 측정된 값을 전송한다.

URI	Method	Request	Response	Description
/ElectricLeakageDetector	GET	interface	200	상태값
		unit	403	unit 에러
			404	리소스에러
	POST	interface	200	속성설정
value		403	설정에러	

누전감지기 속성은 누전감지기 리소스 유형 (rt), 누전감지기 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 누전감지기가 측정값을 전송하고자 하는 인터페이스 (if), 누전감지기가 측정하는 누전상태 (leakagestate), 누전감지기가 알람을 하는 시간간격 (alarminterval), 알람간격 단위(interval\_unit) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
leakagestate	string	M	read only	누전상태
alarminterval	number	0	read write	알람간격
interval_unit	string	0	read write	알람간격단위

누전감지기의 질의(Request) 및 응답(Response) 메시지는 누전감지기 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.electricleakagedetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "leakagestate": "normal",
  "alarminterval": 10,
  "interval_unit": "sec"
}

```

```

{
  "alarminterval": 5,
  "interval_unit": "min"
}

```

■ 아크감지기

아크감지기는 돈사 시설의 아크 방전을 감지하는 장치로서, 측정된 값을 전송한다.

URI	Method	Request	Response	Description
/ArcDetector	GET	interface	200	상태값
		unit	403	unit 에러
			404	리소스에러
	POST	interface	200	속성설정
value		403	설정어러	

아크감지기 속성은 아크감지기 리소스 유형 (rt), 아크감지기 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 아크감지기가 측정값을 전송하고자 하는 인터페이스 (if), 아크감지기가 측정하는 아크발생상태 (arcstate), 아크감지기가 알람을 하는 시간간격 (alarminterval), 알람간격 단위(interval\_unit) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
arcstate	string	M	read only	아크상태
alarminterval	number	0	read write	알람간격
interval_unit	string	0	read write	알람간격단위

아크감지기의 질의(Request) 및 응답(Response) 메시지에는 아크감지기 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.arcdetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "arcstate": "normal",
  "alarminterval": 10,
  "interval_unit": "sec"
}

```

```

{
  "alarminterval": 1,
  "interval_unit": "min"
}

```

■ 조도감지기

조도감지기는 돈사내 조도를 감지하는 장치로서, 측정된 값을 전송한다.

URI	Method	Request	Response	Description
/LightDetector	GET	interface	200	상태값
		unit	403	unit 에러
			404	리소스에러
	POST	interface	200	속성설정
value		403	설정어러	

조도감지기 속성은 조도감지기 리소스 유형 (rt), 조도감지기 속성임을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 조도감지기가 측정값을 전송하고자 하는 인터페이스 (if), 조도감지기가 측정하는 조도값 (light), 설정 조도값 (s\_light), 조도 단위 (l\_unit), 조도유효범위 (l\_range) 등으로 정의된다.

속성	값유형	M/O	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
light	number	M	read only	측정조도
s_light	number	0	read write	설정조도
l_unit	string	0	read write	조도단위
l_range	number(min, max)	0	read write	조도유효범위

조도감지기의 질의(Request) 및 응답(Response) 메시지에는 조도감지기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.lightdetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "light": 450,
  "s_light": 300,
  "l_unit": "lux",
  "l_range": [ 100, 1000 ]
}
```

```
{
  "s_light": 450,
  "l_unit": "lux",
  "l_range": [ 100, 1000 ]
}
```

나-20) 스마트팜관제 프레임워크-제6부: 우사사양관리기(11종)(’21제안, ’22제정)

(가) 연구 수행 개요

본 표준은 CoAP 기반 스마트팜(Smart Farm) 관리와 제어 프레임워크 베이스 프로토콜 (TTAS.KO-10.1243-Part1)을 기반으로 스마트팜 우사에 적용되는 사양관리기들의 프로파일을 정의한다. 이를 위해 현재 Open Connectivity Foundation (OCF)에서 추진되고 있는 내용을 참고하여 스마트팜에 적합한 프로파일을 정의한다.

(나) 연구 주요 내용

CoAP 기반 스마트팜 관리 및 제어 프레임워크에서 스마트팜 기자재는 클라이언트-서버 구조로 동작하며, 각 기자재가 가지는 고유의 기능은 리소스가 되며, 리소스는 리소스 모델을 통해 표현되어 외부로 노출될 수 있고, 이를 통해 해당 기자재의 기능을 이용할 수 있다. 따라서 해당 기자재가 어떤 기자재인지를 정의하는 것은 해당 장치의 리소스 모델을 구성하는 것으로 달성될 수 있다. 리소스 모델은 기자재의 특성 및 기능에 따라 자유롭게 정의될 수 있고, CoAP 기반 스마트팜 관리 및 제어 프레임워크에서 호환될 수 있는 JSON, OpenAPI2.0, YAML 등의 포맷으로 정의한다.

리소스 모델을 구성하는 예로, 센서의 경우 장치가 측정하여 생성하는 값이 온도인지 습도인지, 온도이면 섭씨인지 화씨인지, 유효 온도의 범위는 어떤지 등을 표현할 수 있다. 즉, 특정 포맷의 데이터를 GET 메소드를 사용하여 얻을 수 있게 정의한다. 구동기의 경우에도 센서의 경우처럼 구동의 상태를 얻을 수 있는 GET 메소드를 사용할 수 있고, 아울러 구동기를 제어할 수 있는 POST 메소드가 적용될 수 있다. 다음은 CoAP 기반 스마트팜 관리 및 제어 프레임워크에 적용할 수 있는 스마트팜 우사 사양관리 장비 프로파일을 정의한다. 사양관리 장비 프로파일은 로봇착유기, 자동급이기, 체중측정기, 자동포유기, 분만감지기, 발정감지기, TMR자동급이기, TMR배합기, 사료빈관리기, 음수관리기 의 10종 기기이며, 각 장비 프로파일은 OCF에서 추진하는 바와 같이 OpenAPI2.0 형태로 정의한다.

■ 로봇착유기

로봇착유기는 착유실에 설치하여 젖소의 유두를 탐지하여 착유(진공펌프), 탈락, 집유, 세척 등의 전 과정을 자동으로 수행하는 장치로서, 착유량, 착유시간, 착유횟수 등 착유와 관련된 측정값을 전송한다.

통합자원식별자	요청방법	요청	응답	설명
/MilkingMachine	GET	interface	200 정상	장치값 얻기
		unit	403 unit에러	
			404 리소스에러	
	POST	interface	200 속성설정	속성 설정
value		403 설정에러		

로봇착유기 속성은 로봇착유기 리소스 유형(rt), 로봇착유기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 로봇착유기 장치값을 전송하고자 하는 인터페이스(if), 개체를 식별하는 개체ID(o\_id), 개체 체중의 유효범위(o\_range), 개체의 측정체중 정밀도(o\_precision), 개체의 측정체중(o\_weight), 개체의 체중측정 일시(o\_datetime), 개체의 체중측정 단위(o\_unit), 개체의 사료섭취량(o\_feed), 착유시 사료섭취량의 단위(o\_funit), 착유시작시간(m\_stime), 착유종료시간(m\_etime), 착유지속시간(m\_duration), 착유지속시간단위(m\_dunit), 착유량(m\_volume), 착유량단위(m\_vunit), 착유 횟수(m\_count), 전도도(m\_ec), 착유온도(m\_temp), 착유온도단위(m\_tunit), 착유성분(m\_ingredient) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
o_id	string	0	read write	개체ID
o_range	number(min,max)	0	read write	개체체중유효범위
o_precision	number	0	read write	체중정밀도
o_weight	number	M	read only	측정체중
o_datetime	string	0	read write	체중측정일시

o_unit	string	0	read write	체중측정단위
o_feed	number	0	read write	착유시사료섭취량
o_funit	string	0	read write	사료량단위
m_stime	string	0	read write	착유시작시간
m_etime	string	0	read write	착유종료시간
m_duration	number	M	read write	착유지속시간
m_dunit	string	0	read write	착유지속시간단위
m_volume	number	M	read write	착유량
m_vunit	string	0	read write	착유량단위
m_count	number	M	read only	착유회수
m_ec	number	0	read only	착유전도도
m_temp	number	0	read only	착유온도
m_tunit	string	0	read write	착유온도단위
m_ingredient	string	0	read write	착유온도성분

로봇착유기의 질의(Request) 및 응답(Response) 메시지에는 로봇착유기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.milkingmachine"
  ],
  "n": "0",
  "id": "string",
  "if": [
    "oic.if.a"
  ],
  "p_id": "string",
  "o_range": [ 100, 1000 ],
  "o_precision": 1,
  "o_weight": 500,
  "o_datetime": "string",
  "o_unit": "Kg",
  "o_feed": 1,
  "o_funit": "Kg",
  "m_stime": "string",
  "m_etime": "string",
  "m_duration": 30,
  "m_dunit": "min",
  "m_volume": 15,
  "m_vunit": "Kg",
  "m_count": 1,
  "m_ec": 2.1,
  "m_temp": 38.1,
  "m_tunit": "C",
  "m_ingredient": "string"
}
```

```
{
  "o_id": "string",
  "o_range": [ 50, 1000 ],
  "o_precision": 0.1,
  "o_unit": "Kg",
  "o_feed": 2,
  "o_funit": "Kg",
  "m_duration": 1800,
  "m_dunit": "sec",
  "m_volume": 10,
  "m_vunit": "Kg",
  "m_count": 1,
  "m_ec": 2.1,
  "m_temp": 38.1,
  "m_tunit": "C",
  "m_ingredient": "string"
}
```

■ 자동급이기

자동급이기는 군사로 사육중인 우군(육성우, 착유우, 건유우 등)에 개체를 식별할 수 있는 기기를 통해 개체별로 사료를 자동으로 공급하는 장치로서, 급이량, 급이횟수, 급이시간 등을 조절하여 사료를 공급하고, 결과 상태를 전송한다. 또한 설정을 통해 장치의 기능을 제어한다.

통합자원식별자	요청방법	요청	응답	설명	
/AutomaticCowFeeder	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정

		value	403	설정예러	
--	--	-------	-----	------	--

자동급이기 속성은 자동급이기 리소스 유형(rt), 자동급이기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 자동급이기 상태값을 전송하고자 하는 인터페이스(if), 자동급이기를 식별하기 위한 급이기ID(b\_id), 급이횟수(f\_count), 급이량(f\_amount), 급이량설정(s\_amount), 급이일시(f\_datetime), 급이량 단위(f\_unit), 급이잔량(c\_amount) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	O	read only	유형
n	string	O	read write	이름
id	string	O	read write	식별자
if	array	O	read only	인터페이스
b_id	string	M	read only	급이기ID
f_count	number	O	read write	급이횟수
f_amount	number	M	read write	급이량
s_amount	number	O	read write	급이설정량
f_datetime	string	O	read write	급이일시
f_unit	string	O	read write	급이량 단위
c_amount	number	O	read only	급이잔량

자동급이기의 질의(Request) 및 응답(Response) 메시지에는 자동급이기 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.automaticcowfeeder"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "b_id": "string",
  "f_count": 0,
  "f_amount": 1.0,
  "s_amount": 1.5,
  "f_datetime": "string",
  "f_unit": "kg",
  "c_amount": 20
}

```

```

{
  "b_id": "string",
  "s_amount": 1.5
}

```

### ■ 체중측정기

체중측정기는 소의 체중을 측정하는 장치로서, 측정된 체중값을 전송한다.

통합자원식별자	요청방법	요청	응답	설명	
/WeightMeter	GET	interface	200	정상	장치값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정예러		

체중측정기 속성은 체중측정기 리소스 유형(rt), 체중측정기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 체중측정기 장치값을 전송하고자 하는 인터페이스(if),

개체를 식별하는 개체ID(p\_id), 체중의 유효범위(range), 개체의 측정체중(p\_weight), 개체체중측정일시(p\_datetime), 체중측정단위(p\_unit) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
p_id	string	0	read write	개체ID
range	numner(min,max)	0	read write	유효체중범위
p_weight	number	M	read only	측정체중
p_datetime	string	0	read write	측정일시
p_unit	string	0	read write	측정단위

체중측정기의 질의(Request) 및 응답(Response) 메시지에는 체중측정기 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.weightmeter"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "p_id": "string",
  "range": [ 5, 300 ],
  "p_weight": 0,
  "p_datetime": "string",
  "p_unit": "kg"
}

```

```

{
  "range": [ 5, 300 ],
  "p_unit": "kg"
}

```

### ■ 자동포유기

자동포유기는 개체의 체중과 일령에 따라 포유량과 시기를 자동으로 조절하는 장치로서, 급이시간, 급이간격 등을 측정하여 전송한다. 또한 설정을 통해 장치의 기능을 제어한다.

통합자원식별자	요청방법	요청	응답	설명	
/AutomaticCalfFeeder	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정어러		

자동포유기 속성은 자동포유기 리소스 유형(rt), 자동포유기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 자동포유기 상태값을 전송하고자 하는 인터페이스(if), 개체를 식별하는 개체ID(o\_id), 급이횟수(f\_count), 급이량(f\_amount), 급이량설정(s\_amount), 급이일시(f\_datetime), 급이량 단위(f\_unit), 우유온도(f\_temp), 우유온도의 단위(f\_tunit), 급이잔량(c\_amount) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자

if	array	0	read only	인터페이스
o_id	string	M	read only	개체식별자
f_count	number	M	read write	급이횟수
f_amount	number	M	read write	급이량
s_amount	number	0	read write	급이설정량
f_datetime	string	0	read write	급이일시
f_unit	string	0	read write	급이량 단위
f_temp	number	0	read write	우유온도
f_tunit	string	0	read write	우유온도 단위
c_amount	number	0	read only	급이잔량

자동포유기의 질의(Request) 및 응답(Response) 메시지에는 자동포유기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.automaticcalffeeder"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "o_id": "string",
  "f_count": 0,
  "f_amount": 1.0,
  "s_amount": 1.5,
  "f_datetime": "string",
  "f_unit": "L",
  "f_temp": 38.5,
  "f_tunit": "C",
  "c_amount": 20
}
```

```
{
  "o_id": "string",
  "f_count": 8
  "s_amount": 1
}
```

### ■ 분만감지기

분만감지기는 소의 분만을 탐지하는 장치로서, 꼬리의 움직임, 빛, 활동량 및 체온의 변화 등을 측정하여 탐지한다.

통합자원식별자	요청방법	요청	응답	설명	
/FarrowingDetector	GET	interface	200	정상	장치값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정에러		

분만감지기 속성은 분만감지기 리소스 유형(rt), 분만감지기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 분만감지기 장치값을 전송하고자 하는 인터페이스(if), 개체를 식별하는 개체ID(o\_id), 꼬리 등 기기 장착 부위의 움직임 횟수(o\_mcount), 빛의 세기(light), 빛 세기의 단위(l\_unit), 개체의 측정 체온(o\_temperature), 측정 체온의 단위(o\_tunit) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자

if	array	0	read only	인터페이스
o_id	string	M	read write	개체ID
o_mcount	number	0	read write	움직임 횟수
light	number	0	read only	측정조도
l_unit	string	0	read write	조도단위
o_temperature	number	0	read only	측정체온
o_tunit	string	0	read write	체온단위

분만감지기의 질의(Request) 및 응답(Response) 메시지에는 분만감지기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.farrowingdetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "o_id": "string",
  "o_mcount": 100,
  "light": 1000,
  "l_unit": "lux",
  "o_temperature": 25,
  "o_tunit": "C"
}
```

```
{
  "o_mcount": 0,
  "l_unit": "lux",
  "o_tunit": "C"
}
```

### ■ 발정감지기

발정감지기는 소의 체내·외(귀, 목, 발목 등)에 센서를 설치하여 생체정보(활동량, 반추시간, 체온 등)를 수집하는 장치로서, 개체별 활동량 등을 측정하여 전송한다.

통합자원식별자	요청방법	요청	응답	설명	
/EstrousDetector	GET	interface	200	정상	장치값 얻기
		unit	403	unit에러	
	POST	interface	200	속성설정	속성 설정
		value	403	설정에러	

발정감지기 속성은 발정감지기 리소스 유형(rt), 발정감지기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 발정감지기 장치값을 전송하고자 하는 인터페이스(if), 개체를 식별하는 개체의 ID(o\_id), 씨수소 등 특정 개체의 접근 횟수(a\_count), 개체의 측정 체온 (o\_temperature), 개체 측정 체온의 단위(o\_tunit), 개체의 움직임 횟수(o\_mcount) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
o_id	string	M	read write	개체ID
a_count	number	0	read write	접근횟수
o_temperature	number	0	read only	측정체온
o_tunit	string	0	read write	체온단위

o_mcount	number	0	read write	움직임 횟수
----------	--------	---	------------	--------

발정감지기의 질의(Request) 및 응답(Response) 메시지는 발정감지기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.estrousdetector"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "o_id": "string",
  "a_count": 10,
  "o_temperature": 25,
  "o_tunit": "C",
  "o_mcount": 100
}
```

```
{
  "a_count": 0,
  "o_tunit": "C"
}
```

### ■ TMR자동급이기

TMR자동급이기는 육성우, 번식우, 비육우 등 우방별로 사육중인 집단(우군)에 TMR사료를 자동으로 공급하는 장치로서, 급이량, 급이횟수, 급이시간 등을 조절하여 사료를 공급하고, 결과 상태를 전송한다. 또한 설정을 통해 장치의 기능을 제어한다.

통합자원식별자	요청방법	요청	응답	설명	
/AutomaticTMRFeeder	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정에러		

TMR자동급이기 속성은 TMR자동급이기 리소스 유형(rt), TMR자동급이기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), TMR자동급이기 상태값을 전송하고자 하는 인터페이스(if), 우방을 식별하기 위한 우방ID(l\_id), 급이횟수(f\_count), 급이량(f\_amount), 급이량설정(s\_amount), 급이일시(f\_datetime), 급이량 단위(f\_unit), 급이잔량(c\_amount) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	O	read only	유형
n	string	O	read write	이름
id	string	O	read write	식별자
if	array	O	read only	인터페이스
l_id	string	M	read only	우방ID
f_count	number	M	read write	급이횟수
f_amount	number	M	read write	급이량
s_amount	number	O	read write	급이설정량
f_datetime	string	O	read write	급이일시
f_unit	string	O	read write	급이량 단위
c_amount	number	O	read only	급이잔량

자동급이기의 질의(Request) 및 응답(Response) 메시지는 자동급이기 속성에 정의된 값들이 포함된다.

```

{
  "rt": [
    "oic.r.automatictmrfeeder"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "l_id": "string",
  "f_count": 0,
  "f_amount": 10,
  "s_amount": 100,
  "f_datetime": "string",
  "f_unit": "kg",
  "c_amount": 20
}

```

```

{
  "l_id": "string",
  "f_count": 2,
  "s_amount": 150
}

```

■ TMR배합기

TMR배합기는 소에 공급하고자 하는 영양소에 따라 조사료, 농부산물, 농후사료 따위를 배합 비율 프로그램에 맞춰 기계적으로 배합하는 장치로서, 배합량, 배합시간 등을 조절하여 사료를 배합하고, 결과 상태를 전송한다. 또한 설정을 통해 장치의 기능을 제어한다.

통합자원식별자	요청방법	요청	응답	설명
/TMRMixer	GET	interface	200 정상	장치상태값 얻기
		unit	403 unit에러	
			404 리소스에러	
	POST	interface	200 속성설정	속성 설정
value		403 설정에러		

TMR배합기 속성은 TMR배합기 리소스 유형(rt), TMR배합기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), TMR배합기 상태값을 전송하고자 하는 인터페이스(if), TMR급이기를 식별하기 위한 급이기ID(b\_id), 배합량(tmr\_amount), 배합량단위(tmr\_funit), 배합시간(tmr\_time), 배합시간단위(tmr\_tunit), 배합성분(tmr\_ingredient), 배합비(tmr\_ratio), 급이량(f\_amount), 급이량설정(s\_amount), 급이일시(f\_datetime), 급이량 단위(f\_unit) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	O	read only	유형
n	string	O	read write	이름
id	string	O	read write	식별자
if	array	O	read only	인터페이스
b_id	string	M	read write	급이기식별자
tmr_amount	number	M	read write	배합량
tmr_funit	string	O	read write	배합량단위
tmr_time	string	M	read write	배합시간
tmr_tunit	string	O	read write	배합시간단위
tmr_ingredient	string	O	read write	배합성분
tmr_ratio	string	O	read write	배합비
f_amount	number	M	read write	급이량
s_amount	number	O	read write	급이설정량
f_datetime	string	O	read write	급이일시
f_unit	string	O	read write	급이량 단위

자동급이기의 질의(Request) 및 응답(Response) 메시지에는 자동급이기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.tmrmixer"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ],
  "b_id": "string",
  "tmr_amount": 100,
  "tmr_funit": "Kg",
  "tmr_time": "string",
  "tmr_tunit": "Min",
  "tmr_ingredient": "string",
  "tmr_ratio": "string",
  "f_amount": 10,
  "s_amount": 100,
  "f_datetime": "string",
  "f_unit": "kg"
}
```

```
{
  "b_id": "string",
  "tmr_amount": 30,
  "tmr_funit": "Kg",
  "tmr_time": 30,
  "tmr_tunit": "Min",
  "f_amount": 30
}
```

■ 사료빈관리기

사료빈관리기는 사료빈의 하부에 전자저울(로드셀)을 설치하여 무게를 측정하는 장치로서 사료의 잔량을 확인할 수 있다.

통합자원식별자	요청방법	요청	응답	설명
/BinManager	GET	interface	200	정상
		unit	403	unit에러
			404	리소스에러
	POST	interface	200	속성설정
value		403	설정에러	

사료빈관리기 속성은 사료빈관리기 리소스 유형(rt), 사료빈관리기 속성을 나타내는 사용자 친숙한 이름(n), 해당 속성 인스턴스를 구분할 수 있는 식별자(id), 사료빈관리기 상태값을 전송하고자 하는 인터페이스(if), 사료빈ID(b\_id), 사료정보(f\_info), 사료배출횟수(f\_count), 사료배출량(f\_amount), 사료배출량설정(s\_amount), 사료배출일시(f\_datetime), 사료배출 설정일시(s\_datetime), 사료배출 단위(f\_unit), 사료잔량(c\_amount) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	O	read only	유형
n	string	O	read write	이름
id	string	O	read write	식별자
if	array	O	read only	인터페이스
b_id	string	M	read only	사료빈ID
f_info	string	O	read write	사료정보
f_count	number	O	read write	사료배출횟수
f_amount	number	M	read write	사료배출량
s_amount	number	O	read write	사료배출설정량
f_datetime	string	O	read write	사료배출일시
s_datetime	string	O	read write	사료배출설정일시

f_unit	string	0	read write	사료배출량 단위
c_amount	number	0	read only	사료잔량

사료빈관리의 질의(Request) 및 응답(Response) 메시지는 사료빈관리기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.binmanager"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "b_id": "string",
  "f_info": "string",
  "f_count": 0,
  "f_amount": 50,
  "s_amount": 100,
  "f_datetime": "string",
  "s_datetime": "string",
  "f_unit": "kg",
  "c_amount": 1000
}
```

```
{
  "b_id": "string",
  "s_datetime": "string",
  "s_amount": 100
}
```

### ■ 음수관리기

음수관리기는 축사에 공급되는 음수량을 측정하는 장치로서, 공급되는 음수량을 설정된 주기에 맞추어 측정하여 축사에 음수가 정상적으로 공급되고 있는지 확인할 수 있다.

통합자원식별자	요청방법	요청	응답	설명	
/WaterManager	GET	interface	200	정상	장치상태값 얻기
		unit	403	unit에러	
			404	리소스에러	
	POST	interface	200	속성설정	속성 설정
value		403	설정에러		

음수관리기 속성은 음수관리기 리소스 유형(rt), 음수관리기 속성을 나타내는 사용자 친숙한 이름 (n), 해당 속성 인스턴스를 구분할 수 있는 식별자 (id), 음수관리기 상태값을 전송하고자 하는 인터페이스(if), 음수관리기ID(w\_id), 음수공급량(w\_amount), 음수공급량측정일시(w\_datetime), 음수공급단위(w\_unit), 음수공급제어(w\_control) 등으로 정의된다.

속성	값유형	M(선택)/O(필수)	접근유형	설명
rt	array	0	read only	유형
n	string	0	read write	이름
id	string	0	read write	식별자
if	array	0	read only	인터페이스
w_id	string	M	read only	음수관리기ID
w_amount	number	M	read write	음수공급량
w_datetime	string	0	read write	음수공급일시
w_unit	string	0	read write	음수공급유량단위
w_control	string	0	read write	음수공급제어

음수관리기의 질의(Request) 및 응답(Response) 메시지는 음수관리기 속성에 정의된 값들이 포함된다.

```
{
  "rt": [
    "oic.r.watermanager"
  ],
  "n": "string",
  "id": "string",
  "if": [
    "oic.if.a",
    "oic.if.baseline"
  ]
  "w_id": "string",
  "w_amount": 50,
  "w_datetime": "string",
  "w_unit": "L",
  "w_control": "open"
}
```

```
{
  "w_id": "string",
  "w_unit": "L",
  "w_control": "close"
}
```

## 2) 국가 표준 기술 개발

### 가) 국가 표준 추진 절차

국가표준이란 국가 사회의 모든 분야에서 정확성, 합리성 및 국제성을 높이기 위하여 국가에서 통일적으로 준용하는 과학적·기술적 공공 기준을 말한다(국가 표준기본법 제3조 제1호). ICT 국가표준은 방송, 통신, 전파, 정보 등 정보통신기술(ICT)을 활용한 모든 분야에서 통일적으로 준용하는 공공기준 및 단위로 과학기술정보통신부 국립전파연구원에서 제정·고시하며, 국가표준 제·개정 절차는 다음과 같다.

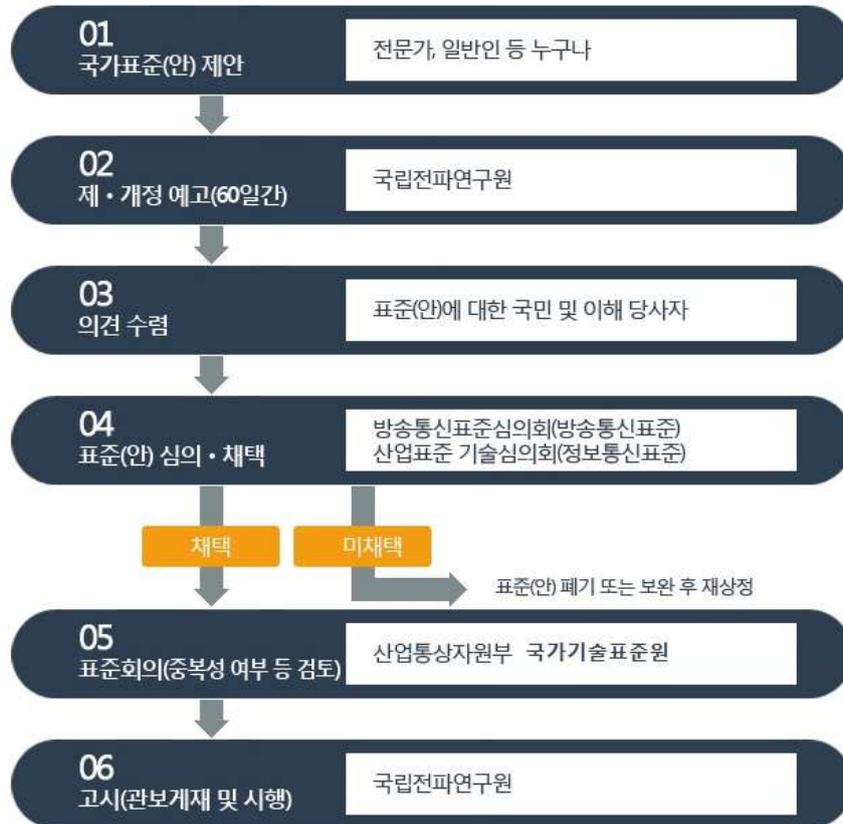


그림 99 - 국가표준(KS) 진행 절차

개발된 표준에 대한 의견수렴은 관보 게재 및 시행에 앞서 공청회 등을 통해 이루어진다. 또한 기개발된 표준안에 대해서는 개정 작업을 통해 추후 이해 관계자의 의견을 반영할 수 있다.

## 나) 국가 제·개정 표준

### 나-1) [국가표준] 스마트온실 센서 메타데이터

스마트온실을 위한 센서 인터페이스[TTAK.KO-10.0903] 표준에서는 스마트 온실 내부의 센서 장치들을 작동시키기 위한 기계적 인터페이스와 전기적 인터페이스 및 측정 범위 등을 기술하고 있다.

온실에서 사용되는 각 센서에서 측정된 센서 데이터는 온실에 대한 환경 정보를 다루고 있으며, 측정된 환경 정보는 센서노드를 통하여 전달되어 해당 온실의 작물을 생육하기 위한 환경 조건을 조절하거나 최적의 생육 알고리즘을 도출하기 위한 기반 정보로 사용된다. 이를 위하여, 각 온실에서 측정된 환경 정보는 해당 온실에서 사용되는 이기종 센서와 상관없이 동일한 정보가 동일한 형태로 표현되고 해석되어야 한다.

따라서 본 연구에서는 스마트온실을 위한 센서 인터페이스[TTAK.KO-10.0903] 표준에서 정의한 센서별 측정된 센싱 데이터의 측정 범위에 대한 메타데이터의 데이터 엘리먼트 및 구조를 기술하는 내용을 기반으로 TTA 단체 표준을 제안하여 2018년 6월 최종 단체표준으로 승인되었다. 제안된 표준에서 정의하는 메타데이터는 다양한 인터페이스 프로토콜에서 공통으로 사용될 수 있다.

#### 1) 센싱 디바이스 종류별 데이터 범위

센싱 디바이스는 센싱 디바이스의 타입에 따라 센싱 데이터 측정 범위가 다르다. 표 1은 각 센싱 디바이스 종류별 스마트 온실을 위한 센싱 데이터 측정 범위, 센싱 데이터 최소 측정 범위, 센싱 데이터 단위, 데이터 타입 등을 기술한다. 각 센싱 디바이스 종류별 데이터 측정 범위 및 데이터 단위는 스마트온실을 위한 센서 인터페이스 [TTAK.KO-10.0903] 표준에서 정의된 내용을 따른다.

표 1 - 센싱 디바이스 종류별 데이터 범위 및 데이터 타입

번호	분류	데이터 측정 범위	데이터 단위	타입
1	온도	-20~80	℃	xs:float
2	습도	0~100	%	xs:float
3	CO <sub>2</sub>	0~3000	ppm	xs:float
4	일사	0~2000	W/m <sup>2</sup>	xs:float
5	풍향	0~360	°(방향 각)	xs:float
6	풍속	0~40	m/s	xs:float
7	감우	on/off	-	xs:Integer
8	광양자	0~2000	umol/m <sup>2</sup> s	xs:float
9	토양함수율	0~50	% vol	xs:float
10	토양수분장력	0~100	kPa	xs:float
11	EC	0~10	dS/m	xs:float
12	PH	2~12	pH	xs:float
13	지온	-20~80	℃	xs:float

센서의 센싱 값 관련 정보는 표 2와 같다.

표 2 - 센서의 센싱 값 관련 정보

구분	분류	M/O	데이터 값
상태 정보	센싱 값 (Sensing Value)	M	각 센서 타입별 센싱 값

※ M: Mandatory, O: Optional

#### 2) 센싱 디바이스 메타데이터

스마트 온실에서 사용되는 센싱 디바이스에 대한 메타데이터는 각 센싱 디바이스에 대한 속성 정보, 센싱 디바이스에서 보유하고 있는 센싱 상태 등의 디바이스 센싱 값 등을 포함하고 있다.

본 표준에서 정의된 메타데이터는 프로토콜에 따라 바이너리 인코딩 포맷 또는 JSON 인코딩 포맷 등

다양하게 적용될 수 있다. 따라서 메타데이터 기술 시 각 값은 바이너리 인코딩을 위한 값과 JSON 인코딩 값을 함께 기술한다.

### 3) 센싱 디바이스 정보

표 1의 센싱 디바이스 종류 별 센싱 디바이스 정보에 대한 메타데이터는 표 3과 같다.

표 3 - 센싱 디바이스 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceInfo	디바이스별 클래스, 타입, 식별자 등을 포함하는 컨테이너.		
DeviceID	DeviceInfo의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:ID
DeviceClass	DeviceInfo의 엘리먼트. 디바이스의 클래스를 기술한다. 디바이스 클래스의 종류는 다음과 같다. - 0x01: Sensor(센서) - 0x02: Actuator(구동기)	M(1)	xs:NMTOKEN enumeration
DeviceType	DeviceInfo의 엘리먼트. 디바이스 클래스 별 디바이스 종류를 기술한다. 센싱 디바이스 종류는 다음과 같다. - 0x01: Temperature(온도) - 0x02: Humidity(습도) - 0x03: CO <sub>2</sub> - 0x04: Pyranometer(일사) - 0x05: WindDirection(풍향) - 0x06: WindSpeed(풍속) - 0x07: RainDetector(감우) - 0x08: Quantum(광양자) - 0x09: SoilMoisture(토양 함수율) - 0x10: Tensiometer(토양 수분 장력) - 0x11: EC - 0x12: pH - 0x13: SoilTemperature(지온)	M(1)	xs:NMTOKEN enumeration
Description	DeviceInfo의 엘리먼트. 센싱 디바이스에 대한 설명을 기술한다.	O(0-1)	xs:string
Location	DeviceInfo의 엘리먼트. 센싱 디바이스가 설치된 위치를 기술한다. 온실 내에서의 상세 위치, 온실 밖의 상세 위치 등 다양한 정보가 기술될 수 있다.	O(0-1)	xs:string
Target	DeviceInfo의 엘리먼트. 센싱 디바이스가 센싱하고자 하는 대상을 기술한다. 온실 내 대기, 배지(토양), 작물의 줄기, 잎, 과실, 뿌리 등 다양한 정보가 기술될 수 있다.	O(0-1)	xs:string
ValueUnit	DeviceInfo의 엘리먼트. 센싱 값의 단위, 디폴트는 표준에서 정한 단위, 센싱 값의 단위에 대한 디폴트 값 선택 시, 디바이스 종류별 표에서 정한 단위가 사용됨. - 0x01: Celcius - 0x02: Percentage - 0x03: ppm - 0x04: W/m <sup>2</sup> - 0x05: Degree - 0x06: m/s - 0x07: NoUnit - 0x08: umol/m <sup>2</sup> s - 0x09: %vol. - 0x10: kPa - 0x11: dS/m - 0x12: pH	O(0-1)	xs:NMTOKEN enumeration

엘리먼트	정의/의미	지원	타입
	- 0x13: Fahrenheit		

※ M: Mandatory, O: Optional

#### 4) 센싱 디바이스의 센싱 상태

센싱 디바이스의 센싱 상태 정보에 대한 메타데이터는 표 4와 같다.

표 4 - 센싱 상태 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceSensingStatus	센싱 디바이스의 센싱 값을 포함하는 컨테이너		
DeviceID	DeviceSensingStatus의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:IDREF
ValueType	DeviceSensingStatus의 엘리먼트. 값의 타입을 기술한다. 값의 타입은 다음과 같다. 디폴트 값은 float을 사용한다. - 0x00: integer - 0x01: float	O(0-1)	xs:NMTOKEN enumeration
SensingValue	DeviceSensingStatus의 엘리먼트. 값 타입(ValueType)에 따른 디바이스의 값을 기술한다. 디바이스 타입 및 디바이스 종류에 해당하는 센싱 값(Sensing Value)을 의미한다. - 센싱 값이 float로 표현되는 경우, 4 바이트로 표현되며 IEEE-754 표준을 따른다.	M(1)	타입에 따라 적용됨

※ M: Mandatory, O: Optional

## 나-2) [국가표준] 스마트온실 구동기 메타데이터

스마트 온실을 위한 구동기 인터페이스[TTAK.KO-10.0845]에서는 스마트 온실 내부에서 구동기 장치들을 작동시키기 위한 기계적, 전기적 연결 규칙과 작동 방식을 기술하고 있다.

온실에서 사용되는 각 센서에서 측정된 센서 데이터는 온실에 대한 환경 정보를 다루고 있으며, 측정된 환경 정보는 센서노드를 통하여 전달되어 해당 온실의 작물을 생육하기 위한 환경 조건을 조절하거나 최적의 생육 알고리즘을 도출하기 위한 기반 정보로 사용된다. 환경 조건을 조절하기 위해서는 구동기노드를 통하여 특정 구동기를 제어하는 제어 정보로 전달하여 최적의 생육 조건을 유지하고자 한다. 이를 위하여, 각 온실에서 구동기 제어 정보는 해당 온실에서 사용되는 기기종 구동기와 상관없이 동일한 정보가 동일한 형태로 표현되고 해석되어야 한다.

따라서 본 연구에서는 스마트 온실을 구동기 인터페이스[TTAK.KO-10.0845]에서 정의한 구동기 작동 방식에 대한 메타데이터의 데이터 엘리먼트 및 구조를 기술하는 내용을 기반으로 TTA 단체 표준을 제안하여 2018년 6월 최종 단체표준으로 승인되었다. 제안된 표준에서 정의하는 메타데이터는 다양한 인터페이스 프로토콜에서 공통으로 사용할 수 있다.

### 1) 구동기 디바이스 종류별 상태 정보

구동기 디바이스는 구동기 타입에 따라 상태 정보와 동작 제어 명령이 구분된다. 표 1은 각 구동기 디바이스 종류별 구동기 타입 및 가능한 상태 정보를 제공한다.

표 1 - 구동기 디바이스 종류 별 타입 구분 및 가능한 상태 정보

번호	분류	구동기 타입	상태 정보
1	천창	개폐형	OPENING(여는 중)/CLOSING(닫는 중)/ OPEN(열림)/CLOSED(닫힘)
2	측창	개폐형	OPENING/CLOSING/OPEN/CLOSED
3	보온재	개폐형	OPENING/CLOSING/OPEN/CLOSED
4	커튼	개폐형	OPENING/CLOSING/OPEN/CLOSED
5	환풍기	스위치형	ON(작동)/OFF(정지)
6	유동팬	스위치형	ON(작동)/OFF(정지)
7	관수 모터	스위치형	ON(작동)/OFF(정지)
8	관수 밸브	스위치형	ON(작동)/OFF(정지)
9	냉난방기	스위치형	ON(작동)/OFF(정지)

### 2) 스위치형 구동기의 상태 정보와 제어 명령 정보

스위치형 구동기의 상태 정보와 제어 명령 정보는 표 2와 같다.

표 2 - 스위치형 구동기의 상태 정보 및 제어 명령 정보

구분	분류	M/O	데이터 값
상태 정보	동작 상태 (Status)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	O	동작 유지 기간 날짜와 시간 정보를 기술
	동작 시간 (Operation Time)	O	동작 유지 시간 시(hour), 분(minute), 초(second) 정보를 기술
제어 정보	동작 명령 (Operation Command)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	O	원하는 동작 기간 날짜와 시간 정보를 기술
	동작 시간 (Operation Time)	O	원하는 동작 시간 시(hour), 분(minute), 초(second) 정보를 기술

※ M: Mandatory, O: Optional

### 3) 개폐형구동기의 상태 정보와 제어 명령 정보

개폐형 구동기의 상태 정보와 제어 명령 정보는 표 3과 같다.

표 3 - 개폐형구동기의 상태 정보 및 제어 명령 정보

구분	분류	M/O	데이터 값
상태 정보	동작 상태 (Operation Status)	M	OPENING(여는 중)/CLOSING(닫는 중) /OPEN(열림)/CLOSED(닫힘)
	동작 위치 (Operation Position)	O	현재 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	O	동작 속도
제어 정보	동작 명령 (Operation Command)	M	OPEN(열림)/CLOSE(닫힘)
	동작 위치 (Operation Position)	O	원하는 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	O	동작 속도

※ M: Mandatory, O: Optional

### 4) 구동기 디바이스 메타데이터

스마트 온실에서 사용되는 각 구동기 디바이스에 대한 메타데이터는 구동기 디바이스에 대한 속성 정보, 구동기 디바이스에서 보유하고 있는 동작 상태 정보, 구동기 디바이스의 동작 제어 명령 정보 등을 포함하고 있다. 표준에서 정의된 메타데이터는 프로토콜에 따라 바이너리 인코딩 포맷 또는 JSON 인코딩 포맷 등 다양하게 적용될 수 있다. 따라서 메타데이터 기술 시 각 값은 바이너리 인코딩을 위한 값과 JSON 인코딩 값을 함께 기술한다.

### 5) 구동기 디바이스 정보

구동기 디바이스 종류별 구동기 디바이스 정보에 대한 메타데이터는 표 4와 같다.

표 4 - 구동기 디바이스 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceInfo	디바이스별 클래스, 타입, 식별자, 위치 정보, 타겟 등을 포함하는 컨테이너.		
DeviceID	DeviceInfo의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:ID
DeviceClass	DeviceInfo의 엘리먼트. 디바이스의 클래스 기술한다. 디바이스 클래스는 다음과 같다.주1) - 0x01: Sensor(센서) - 0x02: Actuator(구동기)	M(1)	xs:NMTOKEN enumeration
DeviceType	DeviceInfo의 엘리먼트. 디바이스 클래스별 디바이스 종류를 기술한다. 구동기 디바이스 종류는 다음과 같다.주1) - 0x01: UpperMotor(천창) - 0x02: SideMotor(측창) - 0x03: WarmMotor(보온재)	M(1)	xs:NMTOKEN enumeration

	<ul style="list-style-type: none"> <li>- 0x04: CurtainMotor(커튼)</li> <li>- 0x05: VentilationFan(환풍기)</li> <li>- 0x06: CirculationFan(유동팬)</li> <li>- 0x07: IrrigationPump(관수 모터)</li> <li>- 0x08: IrrigationValve(관수 밸브)</li> <li>- 0x09: CoolingHeater(냉난방기)</li> </ul>		
Description	DeviceInfo의 엘리먼트. 구동기 디바이스에 대한 설명을 기술한다.	0(0-1)	xs:string
Location	DeviceInfo의 엘리먼트. 구동기 디바이스가 설치된 위치를 기술한다. 온실 내에서의 상세 위치, 온실 밖의 상세 위치 등 다양한 정보가 기술될 수 있다. - 예: 북쪽/남쪽	0(0-1)	xs:string
Target	DeviceInfo의 엘리먼트. 구동기 디바이스 종류별 필요에 따라 추가적으로 기술하고자 하는 대상 정보를 기술한다. 천창과 측창의 경우, 구체적인 대상 정보 - 예: 1중창/2중창/3중창/4중창/5중창	0(0-1)	xs:string
SpeedUnit	DeviceInfo의 엘리먼트. 구동기 디바이스에 대한 동작 속도 값의 단위.주1) - 0x00: rpm	0(0-1)	xs:NMTOKEN enumeration

※ M: Mandatory, O: Optional

주1) 바이너리 인코딩의 경우 분류 값과 JSON 인코딩의 경우 분류 값을 기술한다.

#### 6) 구동기 디바이스 동작 상태

구동기 디바이스의 동작 상태 정보에 대한 메타데이터는 표 5와 같다.

표 5 - 구동기 디바이스 동작 상태 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceOperationStatus	디바이스 식별자, 동작 상태 정보, 상태 정보 관련 속성 정보 등을 포함하는 컨테이너.		
DeviceID	DeviceOperationStatus의 엘리먼트. 디바이스를 구분하는 식별자를 기술한다.	M(1)	xs:IDREF
StatusValue	DeviceOperationStatus의 엘리먼트. 구동기 디바이스의 동작 상태 값을 기술한다.주1) o 스위치형 구동기 동작 상태 정보 - 0x00: ON(작동) - 0x01: OFF(정지) o 개폐형 구동기 동작 상태 정보 - 0x10: OPENING(열리는 중) - 0x11: CLOSING(닫히는 중) - 0x12: OPEN(열림) - 0x13: CLOSED(닫힘) <sup>주2)</sup>	M(1)	xs:NMTOKEN enumeration
OperationDuration	DeviceOperationStatus의 엘리먼트. 동작 유지 기간. 날짜와 시간 정보를 기술한다. 스위치형 구동기에 사용된다.	0(0-1)	xs:Duration
OperationTime	DeviceOperationStatus의 엘리먼트. 동작 유지 시간. 시(hour), 분(minute), 초(second) 정보를 기술한다. 스위치형 구동기에 사용된다.	0(0-1)	xs:Time
OperationPosition	DeviceOperationStatus의 엘리먼트. 원하는 동작 위치 정보, 0 ~ 100. - 0x00 : 0 %: 완전히 닫힌 상태 - 0x64 : 100 %: 완전히 열린 상태 개폐형 구동기에 사용된다.	0(0-1)	xs:integer

OperationSpeed	DeviceOperationStatus의 엘리먼트. 동작 속도, rpm을 디폴트 단위로 사용한다. 개폐형 구동기에 사용된다.	0(0-1)	xs:integer
----------------	---	--------	------------

※ M: Mandatory, O: Optional

주1) 바이너리 인코딩의 경우 분류 값과 JSON 인코딩의 경우 분류 값을 기술한다.

주2) 개폐형 구동기의 동작 상태 정보에서 'CLOSED' 상태는 'OPEN' 상태이면서 동작 위치가 '0%'인 경우와 동일하다.

### 7) 구동기 디바이스 동작 제어 명령

구동기 디바이스의 동작 제어 명령 정보에 대한 메타데이터는 표 6과 같다.

표 6 - 구동기 디바이스 동작 제어 명령 정보 메타데이터

엘리먼트	정의/의미	지원	타입
DeviceOperationCommand	디바이스 식별자, 제어 명령, 제어 명령 관련 속성 정보 등을 포함하는 컨테이너		
DeviceID	DeviceOperationCommand의 엘리먼트. 구동기 디바이스를 식별자를 기술한다.	M(1)	xs:IDREF
ControlCommand	DeviceOperationCommand의 엘리먼트. 구동기 디바이스에 대한 동작 제어 명령 (Operation Control Command).주1) o 스위치형 구동기 동작 제어 명령 정보 - 0x00: ON(작동) - 0x01: OFF(정지) o 개폐형 구동기 동작 제어 명령 정보 - 0x10: OPEN(열어라) - 0x11: CLOSE(닫아라)	M(1)	xs:NMTOKEN enumeration
OperationDuration	DeviceOperationCommand의 엘리먼트. 동작 기간. 날짜와 시간 정보를 기술한다. 스위치형 구동기에 사용된다.	O(0-1)	xs:Duration
OperationTime	DeviceOperationCommand의 엘리먼트. 동작 시간, 시(hour), 분(minute), 초(second) 정보를 기술한다. 스위치형 구동기에 사용된다.	O(0-1)	xs:Time
OperationPosition	DeviceOperationCommand의 엘리먼트. 동작 위치 정보, 0 ~ 100. - 0x00 : 0 %: 완전히 닫힌 상태 - 0x64 : 100 %: 완전히 열린 상태 개폐형 구동기에 사용된다.	O(0-1)	xs:integer
OperationSpeed	DeviceOperationCommand의 엘리먼트. 동작 속도, rpm을 디폴트 단위로 기술한다. 개폐형 구동기에 사용된다.	O(0-1)	xs:integer

※ M: Mandatory, O: Optional

주1) 바이너리 인코딩의 경우 분류 값과 JSON 인코딩의 경우 분류 값을 기술한다.

### 나-3) [국가표준] 스마트 온실 센서/구동기 노드 및 온실 통합 제어기 간 RS485 기반 모드버스 인터페이스

스마트 온실에서는 다양한 센서들을 통해 온실 내부의 환경 정보를 취득한다. 측정된 환경 정보는 해당 온실의 작물을 생육하기 위한 환경 조건을 조절하거나, 최적의 생육 알고리즘을 도출하기 위한 기반 정보로도 사용될 수 있다. 이를 위하여 각 온실의 센서 노드 및 구동기 노드, 온실 통합 제어기 간 표준화된 방식의 프로토콜이 사용되어야 서로 다른 장치 간 상호 연동이 보장된다.

이 표준이 스마트 온실에서 적용되는 범위는 그림 1과 같다.

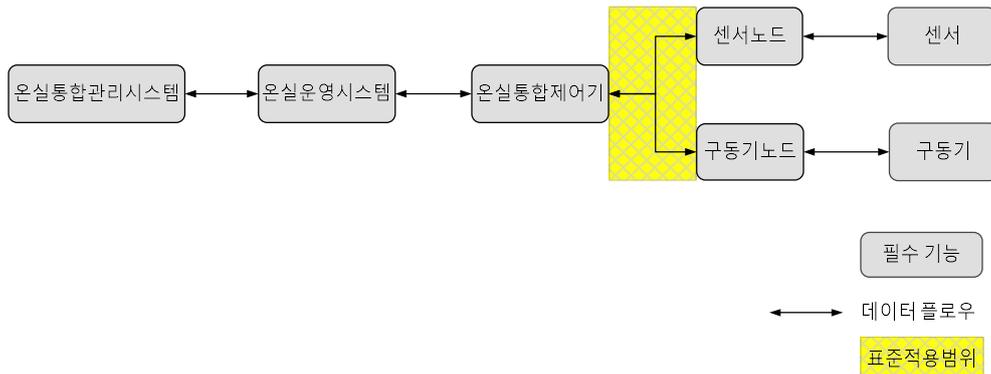


그림 1 - 센서/구동기 노드와 온실 통합 제어기 간 RS485 모드버스 인터페이스 범위

스마트 온실에서 센서 노드와 구동기 노드가 독립적으로 존재하지 않고, 통합 노드로 존재하는 경우 이 표준의 적용 범위는 그림 2와 같다.

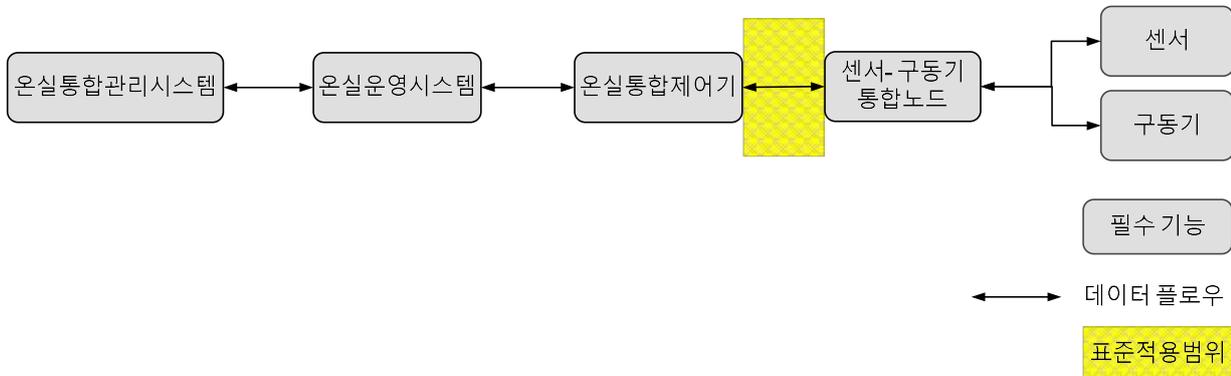


그림 2 - 센서-구동기 통합 노드와 온실 통합 제어기 간 RS485 모드버스 인터페이스 범위

이 표준에서는 센서 노드/구동기 노드/센서-구동기 통합 노드와 온실 통합 제어기 간에 제공되는 기능과 각 기능을 위해 상호 교환되는 메시지 인터페이스, 전달되는 데이터에 대하여 기술한다.

이 표준은 스마트 온실의 센서 노드와 온실 통합 제어기 간, 구동기 노드와 온실 통합 제어기 간 RS485 통신을 하는 경우, 모드버스(MODBUS) 방식의 인터페이스를 제안함으로써 상호 연동을 제공하는 것을 목적으로 한다.

#### ■ 센서노드, 구동기노드 및 온실통합제어기를 위한 주소 할당 (ID)

- RS485 Modbus RTU 방식을 사용하는 장치들을 식별하기 위해서는 1바이트의 주소를 사용한다.
- RS485 Modbus RTU 버스에 접속한 장치마다 고유한 ID 를 부여해야 한다.
- 장치마다 ID를 부여하는 방법은 표준에서 정의하지 않는다.

표 1 - ID와 RS485 모드버스 주소 매핑



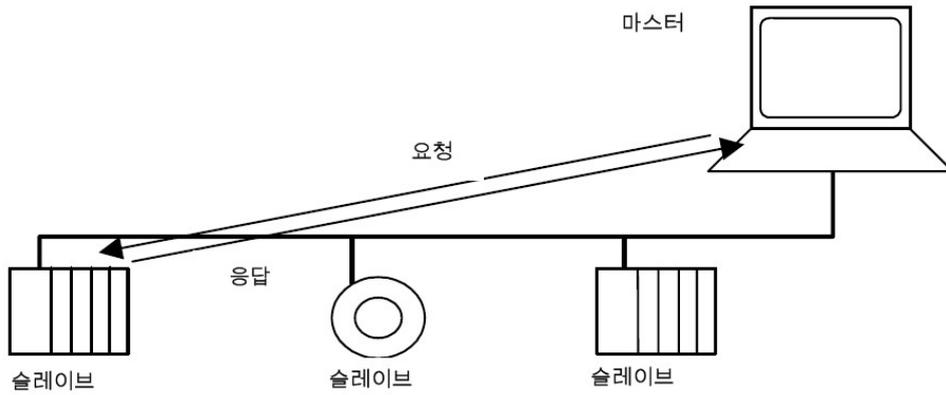


그림 5 - 마스터-슬레이브(master-slave) 방식

마스터 노드는 특정 슬레이브 노드를 지정하여 요청 메시지를 보내고, 해당 슬레이브 노드는 요청 메시지를 받아 처리한 후 이에 대한 응답 메시지를 보낸다. 즉, 모드버스 트랜잭션은 요청과 응답의 두 가지 메시지로 구성된다. 개별 슬레이브 노드 주소는 1에서 247이 가능하다. 주소 0은 모든 슬레이브 노드에 요청 메시지를 보내는 브로드캐스트 주소로 이 표준에서는 사용하지 않는다.

직렬 회선상에서 메시지 전송 모드는 원격 단말 장치(RTU) 모드와 아스키(ASCII) 모드가 있으며, 이 표준에서는 RTU 모드를 사용한다.

각 장치가 RTU 모드를 사용하여 모드버스 직렬 회선으로 통신할 때 데이터 형식은 4비트 16진수 데이터 포맷을 사용한다. 이는 문자 밀집도를 제공함으로써 ASCII 모드보다 더 좋은 처리량(throughput)을 제공한다.

모드버스 PDU를 시리얼 라인으로 전달하기 위해서는 통신 PDU로 모드버스 직렬 회선 PDU를 생성한다. 모드버스 직렬 회선 PDU는 모드버스 PDU의 앞에 주소 필드(address field)를 추가하고, 모드버스 PDU 뒤에 순환 중복 검사(CRC) 필드를 추가함으로써 생성된다.

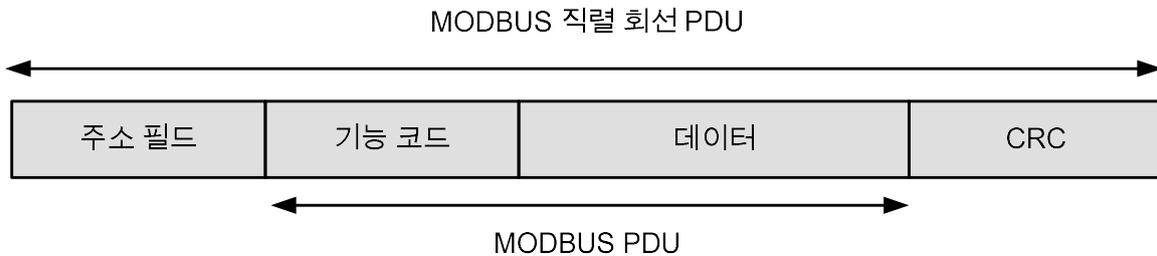


그림 6 - 모드버스 메시지 프레임과 모드버스 PDU와의 관계

모드버스 직렬 회선상에서 주소 필드는 슬레이브 주소만을 포함한다. 따라서 직렬 회선상의 모드버스 RTU 메시지 프레임 구성은 다음과 같이 구성된다. 모드버스 RTU 프레임의 최대 크기는 256바이트이다.

주소 필드	기능 코드	데이터	CRC
1바이트	1바이트	0~252 바이트	2바이트

그림 7 - 모드버스 메시지 프레임 구성

모드버스 메시지는 전송 장치가 직렬 회선상에 프레임 형태로 놓임으로써 전달된다. 전체 메시지 프레임은 문자의 연속적인 스트림으로 전송되어야 하며, 모드버스 메시지 프레임의 전송 직전이나 직후에 3.5 문자(character) 이상의 시간 공백을 유지하도록 한다. 만약 두 문자 사이에 1.5 문자 이상의 공백 기간(silent interval)이 존재하면, 그 메시지 프레임은 완전하지 않은 것으로 간주하여 수신자가 버리도록 한다.

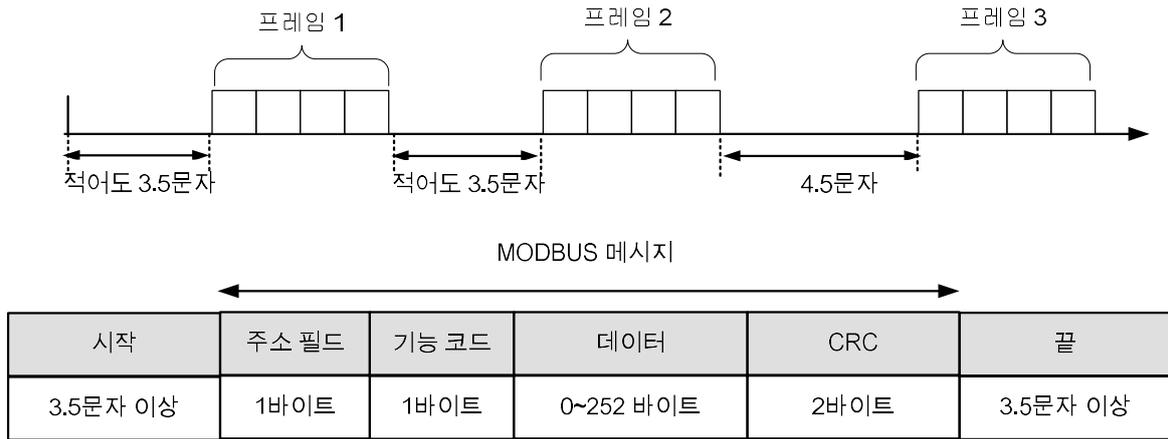


그림 8 - 모드버스 메시지 전송

■ 메시지 플로우

데이터 교환을 위한 메시지 플로우는 다음과 같다.

- 정상적인 경우: 요청(request) → 응답(response)

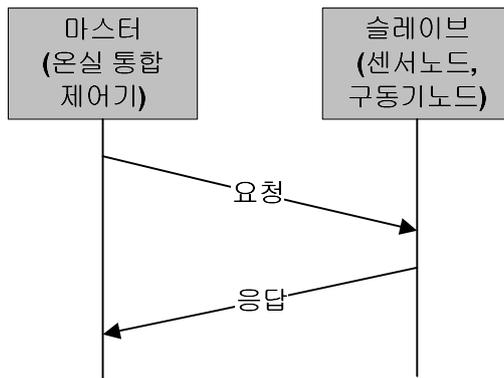


그림 9 - 정상적인 경우의 메시지 플로우

- 예외 응답의 경우: 요청(request) → 예외 응답(exception response)

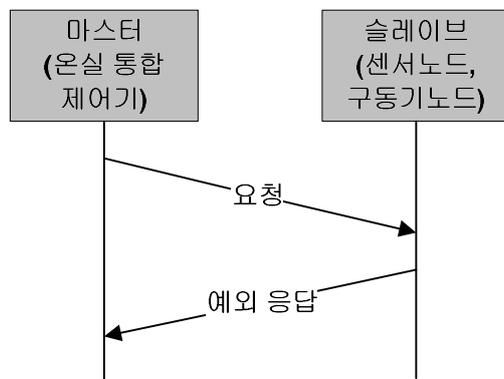


그림 10 - 예외 응답인 경우의 메시지 플로우

- 패킷 에러의 경우: 요청(request) → 시간 경과(timeout)

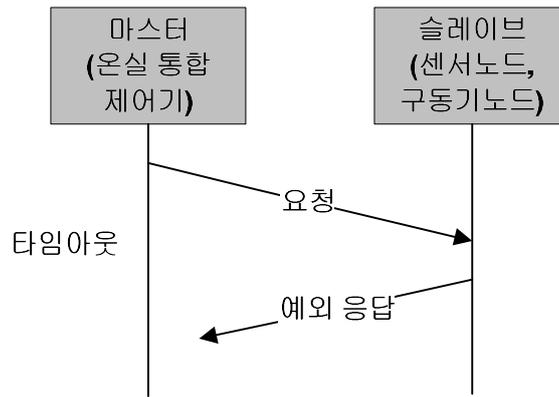


그림 11 - 패킷 에러인 경우의 메시지 플로우

## ■ 패킷 구조

### 요청(request) 패킷

표 2 - 요청 패킷의 구조

슬레이브 주소	기능 코드	데이터	CRC
1바이트	1바이트	0-252바이트	2바이트

- 슬레이브 주소: 센서 노드나 구동기 노드의 주소가 해당됨. 1-247이 가능하며, 0은 브로드캐스트 주소로 되어 있으나, 이 표준에서는 브로드캐스트 주소를 사용하지 않음.
- 기능 코드: 지정된 슬레이브가 어떤 동작을 수행해야 할지를 알려 주는 값이며, 마스터에서 슬레이브로 요청하는 요청값(request code)
- 데이터: 슬레이브 노드가 기능 코드에 정의된 동작을 수행하기 위해 사용하는 부가적인 정보들이다. 레지스터 주소, 다루어야 할 항목의 양, 실제 데이터 바이트의 수 등을 포함할 수 있으며, 기능 코드별로 구체적인 필드가 정해짐.
- CRC: CRC-16 알고리즘 오류 체크, 수신 장치는 에러 체크를 위한 CRC 필드를 제외한 나머지 필드값으로 CRC 계산

### a) 정상적인 응답(normal response) 패킷

표 3 - 정상 응답 패킷의 구조

슬레이브 주소	기능 코드	데이터	CRC
1바이트	1바이트	0-252바이트	2바이트

- 슬레이브 주소: 응답하는 센서 노드나 구동기 노드의 주소
- 기능 코드: 마스터로부터 수신한 기능 코드 값 복사
- 데이터: 슬레이브 노드가 기능 코드에 정의된 동작을 수행한 후 응답하는 데 사용하는 부가적인 정보들이다. 레지스터 주소, 실 데이터 바이트의 수 등을 포함할 수 있으며, 기능 코드별로 구체적인 필드가 정해짐.
- CRC: 에러 체크 값을 표현하는 16비트 수

### b) 예외 응답(exception response) 패킷

표 4 - 예외 응답 패킷의 구조

슬레이브 주소	기능 코드	에러 코드	CRC
1바이트	1바이트	N바이트	2바이트

- 슬레이브 주소: 응답하는 센서 노드나 구동기 노드의 주소
- 기능 코드: 마스터에서 수신한 기능 코드 값에 0x80 추가하여 수정
  - 마스터에서 수신한 기능 코드: 0x03 → 예외 응답의 기능 코드: 0x83
  - 마스터에서 수신한 기능 코드: 0x06 → 예외 응답의 기능 코드: 0x86
- 에러 코드: 예외 발생 내역에 대한 상세 코드 값의 예

- 0x01(illegal function): 제품에서 지원되지 않는 기능 코드
  - 0x02(illegal data address): 유효하지 않은 주소를 접근하고자 할 때
  - 0x03(illegal data value): 지원되지 않는 데이터 값으로 지정하고자 할 때
  - 0x04(slave device failure): 디바이스 문제 발생
- CRC: 에러 체크 값을 표현하는 16비트 수

■ 온실 통합 제어기와 센서 노드, 구동기 노드, 통합 노드 간 제공 기능

이 절에서는 온실 통합 제어기와 센서 노드/구동기 노드/센서-구동기 통합 노드 간에 제공되는 기능을 기술한다. 크게 센싱 정보 획득, 구동기 상태 정보 획득 등의 데이터 확보 기능(GetData)과 구동기 제어 명령 등의 데이터 지정 기능(SetData)으로 분류된다. 이를 위하여 온실 통합 제어기는 센서 노드/구동기 노드/센서-구동기 통합 노드 내에 각 센서나 구동기에 대한 정보를 알고 있다고 가정한다.

이 절에서는 위의 기능들이 모드버스 프로토콜을 이용하여 어떻게 표현되는지를 기술한다(5.3에서 정의된 기능 코드와 데이터 부분을 구체화해서 기술).

a) 데이터 확보 기능(GetData)

온실 통합 제어기는 마스터로 동작하고, 센서 노드 또는 구동기 노드는 슬레이브로 동작한다. 온실 통합 제어기(마스터)는 데이터를 읽어 올 레지스터 주소와 읽어 올 해당 레지스터 수를 포함한 기능 코드 0x03(Read Holding Register) 요청 메시지를 센서 노드 또는 구동기 노드(슬레이브)에게 보낸다. 센서 노드 또는 구동기 노드(슬레이브)는 온실 통합 제어기(마스터)로, 해당 레지스터 주소의 값과 바이트 수를 포함한 응답 메시지를 온실 통합 제어기(마스터)에 전송한다.

이때 온실 통합 제어기는 해당 센서 노드에 연결된 각 센서의 레지스터 주소와 데이터 값의 크기를 사전에 알고 있어야 한다. 마찬가지로 온실 통합 제어기는 해당 구동기 노드에 연결된 각 구동기의 레지스터 주소와 데이터 값의 크기를 사전에 알고 있어야 한다.

모드버스 프로토콜에서 레지스터 데이터 값은 2바이트를 가정하고 있다. 따라서 센서나 구동기의 데이터 값이 2바이트를 초과하는 경우, 해당 레지스터의 시작 주소부터 연속해서 해당 데이터 값을 포함하는 크기를 요청하도록 한다. 예를 들어, 만약 온도 센서 A의 값이 4바이트로 표현이 되는 경우, 모드버스 메시지는 온도 센서 A로 매핑된 레지스터 주소와 레지스터의 수로 2를 포함한 0x03(Read Holding Registers) 요청 메시지를 전송한다.

b) 요청 패킷

요청 메시지의 기능 코드로 0x03을 사용하며, 시작 주소(start address)로 시작하는 레지스터부터 시작하여 레지스터 수(quantity of registers)만큼의 레지스터 값을 읽을 것을 요청한다.

표 5 - 요청 패킷의 구조

기능 코드	시작 주소	레지스터 수
1바이트	1바이트	2바이트

- 기능 코드: 0x03(Read Holding Registers) 요청
- 시작 주소: 데이터를 읽어 올 대상 레지스터의 시작 주소, 각 센서나 구동기에 매핑되는 레지스터 시작 주소
- 레지스터 수: 데이터를 읽어 올 대상 레지스터의 수, 대상 레지스터에 표현된 데이터 값의 크기

c) 응답 패킷

기능 코드 0x03의 레지스터 읽기 요청에 대하여 해당 레지스터로부터 읽어 온 데이터 값의 바이트 수와 레지스터 데이터 값을 포함하여 응답한다.

표 6 - 응답 패킷의 구조

기능 코드	바이트 수	레지스터 값
1바이트	1바이트	2*N바이트(N: quantity of registers)

- 기능 코드: 0x03(Read Holding Registers) 요청에 대한 응답

- 바이트 수: 레지스터 값들을 구성하는 부분의 길이(2\*N), 데이터를 표현하는 완전한 바이트의 수
- 레지스터 값: 각 레지스터 값으로, 각 레지스터별 2바이트를 가정한다.

■ 레지스터 데이터 포맷

응답 패킷의 레지스터 값에 포함되는 데이터 포맷은 다음과 같다.

a) 센서 데이터 포맷

스마트 온실의 각 센서에서 센싱한 데이터 값에 대한 데이터 포맷은 TTAK.KO-10.1046에서 정의한 내용을 따르며, 이 표준에서는 이를 다음과 같이 적용한다.

- 데이터 전송 대역폭이 작은 관계로 바이너리 인코딩 방식을 사용한다.
- 각 센서에서 센싱 값(sensing value)만 전달하도록 한다.

센서에서 센싱하는 값과 관련된 속성 정보는 표 7과 같다.

표 7 - 센서의 센싱 정보 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
상태 정보	센싱 값	M	각 센서 타입별 센싱 값

표 7에서 응답 패킷의 레지스터 값에 포함되는 내용은 센싱 값만 포함하며, 센싱 값에 대한 바이너리 인코딩 포맷은 표 8과 같다.

표 8 - 센서 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명
센싱 값	4	32	값 타입에 따라 다름	디바이스 타입 및 분류에 해당하는 디바이스 센싱 값 - 센싱 값이 실수(float)로 표현되는 경우, 4바이트로 표현되며 IEEE-754 표준을 따른다.

센서는 센서의 타입에 따라 데이터 측정 범위가 다르다. 표 9는 TTAK.KO-10.0903에 따른 각 센서 종류별 데이터 범위, 데이터 타입을 제공한다.

표 9 - 센서 종류별 데이터 범위 및 데이터 타입

번호	분류	데이터 측정 범위	데이터 단위	타입
1	온도 센서	-20 ~ 80	°C	float
2	습도 센서	0 ~ 100	%	float
3	CO <sub>2</sub> 센서	0 ~ 3 000	µmol/mol(PPM)	float
4	일사 센서	0 ~ 2 000	W/m <sup>2</sup>	float
5	풍향 센서	0 ~ 360	°(방향각)	float
6	풍속 센서	0 ~ 40	m/s	float
7	감우 센서	On/OFF	-	integer
8	광양자 센서	0 ~ 2 000	µmol/m <sup>2</sup> /s	float
9	토양 함수율 센서	0 ~ 50	% vol.	float
10	토양 수분 장력 센서	0 ~ 100	kPa	float
11	EC 센서	0 ~ 10	dS/m	float
12	PH 센서	2 ~ 12	pH	float
13	지온 센서	-20 ~ 80	°C	float

b) 구동기 데이터 포맷

스마트 온실에서 각 구동기 상태에 대한 데이터 포맷은 TTAK.KO-10.1045에서 정의한 내용을 따르며, 이 표준에서는 이를 다음과 같이 적용한다.

- 데이터 전송 대역폭이 작은 관계로 바이너리 인코딩 방식을 사용한다.
- 구동기 타입을 스위치형과 개폐형으로 구분하며, 각 타입에 따른 동작 상태와 동작 시간, 동작 상태와 동작 위치 등의 속성 정보를 함께 전달하도록 한다.
- 스위치형 구동기의 동작 기간 속성 정보와 개폐형 구동기의 동작 속도 속성 정보는 적용하지 않는다.

구동기는 타입에 따라 상태 정보가 다르다. 표 10은 TTAK.KO-10.0845에 따른 각 구동기 종류별 타입 및 가능한 상태 정보를 제공한다.

표 10 - 구동기 종류별 타입 구분 및 가능한 상태 정보

번호	분류	구동기 타입	상태 정보
1	천창	개폐형	OPENING(여는 중)/CLOSING(닫는 중)/ OPEN(열림)/CLOSED(닫힘)
2	측창	개폐형	OPENING/CLOSING/OPEN/CLOSED
3	보온 덮개	개폐형	OPENING/CLOSING/OPEN/CLOSED
4	차광막	개폐형	OPENING/CLOSING/OPEN/CLOSED
5	환풍기	스위치형	ON(작동)/OFF(정지)
6	유동 팬	스위치형	ON(작동)/OFF(정지)
7	관수 모터	스위치형	ON(작동)/OFF(정지)
8	관수 밸브	스위치형	ON(작동)/OFF(정지)
9	냉난방기	스위치형	ON(작동)/OFF(정지)

스위치형 구동기와 관련된 상태 정보 및 상태 관련 속성 정보는 표 11과 같이 제공한다.

표 11 - 스위치형 구동기의 상태 정보 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
상태 정보	동작 상태 (Operation Status)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	O	동작 유지 기간 날짜와 시간 정보를 'P4Y2M2DT2H2M2S' 형태로 기술
	동작 시간 (Operation Time)	O	동작 유지 시간 시(hour), 분(minute), 초(second) 정보를 기술

응답 패킷의 레지스터 값에 포함되는 내용은 표 11의 구동기 동작 상태와 상태 관련 속성정보 부분으로, 이 표준에서는 동작 상태와 동작 시간이 적용된다. 바이너리 인코딩 포맷은 표 12와 같다. 이때, RS485 모드버스는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 상태 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 12 - 스위치형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 상태 (Operation Status)	2	8	부호 없는 정수형 (unsigned integer)	구동기 동작 상태 정보 - 0x00: ON (작동) - 0x01: OFF (정지)	스위치형 구동기는 ON/OFF
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 시간 (Operation Time)	4	24	부호 없는정수형 (unsigned integer)	동작 시간 - 시(hour), 분(minute), 초(second) 순 서로 1바이트씩 기술	-
		8	-	레지스터를 구성하기 위하여 추가된 비트	-

개폐형 구동기와 관련된 상태 정보 및 상태 관련 속성 정보는 표 13과 같이 제공한다.

표 13 - 개폐형 구동기의 상태 정보 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
상태 정보	동작 상태	M	OPENING(여는 중)/CLOSING(닫는 중)/

	(Operation Status)		OPEN(열림)/CLOSED(닫힘)
	동작 위치 (Operation Position)	0	현재 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	0	동작 속도

응답 패킷의 레지스터 값에 포함되는 내용은 표 13의 구동기 동작 상태와 상태 관련 속성정보 부분으로, 이 표준에서는 동작 상태와 동작 위치가 적용된다. 바이너리 인코딩 포맷은 표 14와 같다. 이때, RS485 모드버스는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 상태 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 14 - 개폐형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 상태 (Operation Status)	2	8	부호 없는 정수형 (unsigned integer)	구동기 상태 정보 - 0x10: OPENING(열리는 중) - 0x11: CLOSING(닫히는 중) - 0x12: OPEN(열림) - 0x13: CLOSED(닫힘)	- 개폐형 구동기는 OPENING/CLOSING/OPEN/CLOSED
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 위치 (Operation Position)	2	8	부호 없는 정수형 (unsigned integer)	동작 위치 정보, 비율로 표시 - 0 ~ 100	- 개폐형 구동기에 사용 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
		8	-	레지스터를 구성하기 위하여 추가된 비트	-

#### ■ 데이터 지정 기능(SetData) - 단일 레지스터

온실 통합 제어기는 마스터로 동작하고, 구동기 노드는 슬레이브로 동작한다. 온실 통합 제어기(마스터)는 데이터를 쓸 레지스터 주소와 레지스터 값을 포함한 기능 코드(0x06, Write Holding Register) 요청 메시지를 구동기 노드(슬레이브)에 보낸다.

구동기 노드(슬레이브)에서는 온실 통합 제어기(마스터)로 레지스터 주소 정보와 해당 레지스터에 기록한 값을 포함한 응답 메시지를 온실 통합 제어기(마스터)에게 전송한다. 정상적인 응답 메시지는 요청 메시지를 그대로 표현하게 된다.

이때 온실 통합 제어기는 해당 구동기 노드에 연결된 각 구동기의 레지스터 주소와 레지스터에 쓰고자 하는 제어 명령 데이터 값의 크기를 사전에 알고 있어야 한다. 모드버스 프로토콜에서 레지스터 데이터 값은 2바이트를 가정하고 있다. 따라서 구동기의 제어 명령 데이터 값이 2바이트를 초과하지 않는 경우에 사용할 수 있다.

#### a) 요청 패킷

요청 메시지의 기능 코드로 0x06를 사용하며, 레지스터 주소로 시작하는 레지스터에 레지스터 값을 쓸(write) 것을 요청한다.

표 15 - 요청 패킷의 구조

기능 코드	레지스터 주소	레지스터 값
1바이트	2바이트	2바이트

- 기능 코드: 0x06(Write Holding Register)
- 레지스터 주소: 데이터를 기록할 대상 레지스터 주소, 각 구동기에 매핑되는 레지스터 주소
- 레지스터 값: 데이터를 기록할 대상 레지스터 값, 레지스터 값에 포함되는 데이터 포맷은 6.3.3에서 기술한다.

**b) 응답 패킷**

기능 코드 0x06의 레지스터 기록 요청에 대하여 해당 레지스터의 주소와 해당 레지스터에 기록한 데이터 값을 포함하여 응답한다.

표 16 - 응답 패킷의 구조

기능 코드	레지스터 주소	레지스터 값
1바이트	2바이트	2바이트

- 기능 코드: 0x06(Write Holding Register)
- 레지스터 주소: 데이터를 기록한 대상 레지스터 주소
- 레지스터 값: 데이터를 기록한 대상 레지스터 값

**■ 데이터 지정 기능(SetData) - 복수 레지스터**

온실 통합 제어기는 마스터로 동작하고, 구동기 노드는 슬레이브로 동작한다. 온실 통합 제어기(마스터)는 데이터를 쓸 레지스터의 시작 주소와 레지스터 수, 레지스터 데이터 값의 바이트 수, 레지스터 값을 포함한 기능 코드(0x10, Write Multiple Registers) 요청 메시지를 구동기 노드(슬레이브)에게 보낸다.

구동기 노드(슬레이브)에서는 온실 통합 제어기(마스터)로 레지스터 시작 주소 정보와 레지스터 수에 대한 정보를 포함한 응답 메시지를 온실 통합 제어기(마스터)에게 전송한다. 이때 온실 통합 제어기는 해당 구동기 노드에 연결된 각 구동기의 레지스터 주소와 레지스터에 쓰고자 하는 제어 명령 데이터 값의 크기를 사전에 알고 있어야 한다.

모드버스 프로토콜에서 레지스터 데이터 값은 2바이트를 가정하고 있다. 따라서 이 기능은 구동기의 제어 명령 데이터 값이 2바이트를 초과하는 경우에 사용할 수 있다. 예를 들어, 만약 천창 A에 대한 제어 명령의 값이 4바이트로 표현이 되는 경우, 모드버스 메시지는 천창 A로 매핑된 레지스터 시작 주소와 레지스터의 수로 2, 제어 명령에 해당하는 레지스터 값의 바이트 수, 제어 명령 레지스터 값을 포함한 0x10(Write Multiple Registers) 요청 메시지를 전송한다.

**a) 요청 패킷**

요청 메시지의 기능 코드로 16(0x10)을 사용하며, 레지스터 시작 주소로 시작하여 레지스터 수에 해당하는 레지스터에 레지스터 값을 바이트 수만큼 쓸(write) 것을 요청한다. 모드버스 프로토콜에서 레지스터는 2바이트를 가정하고 있다. 따라서 특정 디바이스의 데이터 값이 2바이트를 초과하는 경우, 해당 레지스터의 시작 주소부터 연속해서 해당 데이터 값을 포함하는 크기를 기록하도록 한다. 또한, 특정 디바이스들을 연속한 주소로 할당하고, 각 디바이스에 대한 데이터 값의 크기를 아는 경우, 한 명령어로 두 개 이상의 디바이스에 대한 제어 명령을 전송할 수 있다.

표 17 - 요청 패킷의 구조

기능 코드	시작 주소	레지스터 수	바이트 수	레지스터 값
1바이트	2바이트	2바이트	1바이트	(2*N) 바이트 N = 레지스터 수

- 기능 코드: 16(0x10, Write Multiple Registers)
- 시작 주소: 데이터를 기록할 대상 레지스터의 시작 주소
- 레지스터 수: 대상 레지스터의 수(N)
- 바이트 수: 레지스터 데이터 값의 바이트 수
- 레지스터 값: 데이터를 기록할 대상 레지스터 데이터 값

**b) 응답 패킷**

기능 코드 16(0x10)의 여러 개의 레지스터에 레지스터 기록 요청에 대하여 해당 레지스터의 시작 주소와 레지스터 수를 포함하여 응답한다.

표 18 - 응답 패킷의 구조

기능 코드	시작 주소	레지스터 수
1바이트	2바이트	2바이트

- 기능 코드: 16(0x10, Write Multiple Registers)
- 시작 주소: 데이터를 기록한 대상 레지스터의 시작 주소
- 레지스터 수: 데이터를 기록한 레지스터의 수

c) 레지스터 데이터 포맷

요청 패킷이나 응답 패킷의 레지스터 값에 포함되는 데이터 포맷은 다음과 같다. 레지스터 값의 크기가 2바이트를 넘지 않는 경우, 데이터 지정 기능을 통하여 전달한다. 스마트 온실에서 각 구동기 제어 명령에 대한 데이터 포맷은 TTA.KO-10.1045에서 정의한 내용을 따르며, 이 표준에서는 다음과 같이 적용한다.

- 데이터 전송 대역폭이 작은 관계로 바이너리 인코딩 방식을 사용한다.
- 구동기 타입을 스위치형과 개폐형으로 구분하며, 각 타입에 따른 제어 명령과 동작 시간, 제어 명령과 동작 위치 등의 속성 정보를 함께 전달하도록 한다.
- 스위치형 구동기의 동작 기간 속성 정보와 개폐형 구동기의 동작 속도 속성 정보는 적용하지 않는다.

스위치형 구동기와 관련된 제어 명령 및 제어 명령 관련 속성 정보는 표 19와 같이 제공한다.

표 19 - 스위치형 구동기의 제어 명령 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
제어 정보	동작 명령 (Operation Command)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	O	동작 유지 기간 날짜와 시간 정보를 'P4Y2M2DT2H2M2S' 형태로 기술
	동작 시간 (Operation Time)	O	동작 유지 시간 시(hour), 분(minute), 초(second) 정보를 기술

응답 패킷의 레지스터 값에 포함되는 내용은 표 19의 구동기 동작 제어 명령과 제어 관련 속성 정보 부분으로, 이 표준에서는 동작 명령과 동작 시간이 적용된다. 바이너리 인코딩 포맷은 표 20과 같다. 이때, RS485 모드버스는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 제어 명령 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 20 - 스위치형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 명령 (Operation Command)	2	8	부호 없는 정수형 (unsigned integer)	구동기 제어 명령 - 0x00: ON(작동) - 0x01: OFF(정지)	스위치형 구동기는 ON/OFF
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 시간 (Operation Time)	4	24	부호 없는 정수형 (unsigned integer)	동작 시간 시(hour), 분(minute), 초(second) 순서로 1바이트씩 기술	-
		8	-	레지스터를 구성하기 위하여 추가된 비트	-

개폐형 구동기와 관련된 제어 명령 및 제어 명령 관련 속성 정보는 표 21과 같이 제공한다.

표 21 - 개폐형 구동기의 제어 명령 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
----	----	------------	-------

제어 정보	동작 명령 (Operation Command)	M	OPEN(열기)/CLOSE(닫기)
	동작 위치 (Operation Position)	0	원하는 동작 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	0	동작 속도

요청 패킷의 레지스터 값에 포함되는 내용은 표 21의 구동기 동작 제어 명령과 제어 관련 속성 정보 부분으로, 이 표준에서는 동작 명령과 동작 위치가 적용된다. 바이너리 인코딩 포맷은 표 22와 같다. 이때, RS485 모드버스는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 제어 명령 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 22 - 구동기 종류별 타입 구분 및 가능한 상태 정보

번호	분류	구동기 타입	상태 정보
1	천창	개폐형	OPENING(여는 중)/CLOSING(닫는 중)/ OPEN(열림)/CLOSED(닫힘)
2	측창	개폐형	OPENING/CLOSING/OPEN/CLOSED
3	보온 덮개	개폐형	OPENING/CLOSING/OPEN/CLOSED
4	차광막	개폐형	OPENING/CLOSING/OPEN/CLOSED
5	환풍기	스위치형	ON(작동)/OFF(정지)
6	유동 팬	스위치형	ON(작동)/OFF(정지)
7	관수 모터	스위치형	ON(작동)/OFF(정지)
8	관수 밸브	스위치형	ON(작동)/OFF(정지)
9	냉난방기	스위치형	ON(작동)/OFF(정지)

스위치형 구동기와 관련된 상태 정보 및 상태 관련 속성 정보는 표 23과 같이 제공한다.

표 23 - 스위치형 구동기의 상태 정보 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
상태 정보	동작 상태 (Operation Status)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	0	동작 유지 기간 날짜와 시간 정보를 'P4Y2M2DT2H2M2S' 형태로 기술
	동작 시간 (Operation Time)	0	동작 유지 시간 시(hour), 분(minute), 초(second) 정보를 기술

응답 패킷의 레지스터 값에 포함되는 내용은 표 23의 구동기 동작 상태와 상태 관련 속성정보 부분으로, 이 표준에서는 동작 상태와 동작 시간이 적용된다. 바이너리 인코딩 포맷은 표 24와 같다. 이때, RS485 모드버스는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 상태 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 24 - 스위치형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 상태 (Operation Status)	2	8	부호 없는 정수형 (unsigned integer)	구동기 동작 상태 정보 - 0x00: ON (작동) - 0x01: OFF (정지)	스위치형 구동기는 ON/OFF
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 시간 (Operation Time)	4	24	부호 없는정수형 (unsigned integer)	동작 시간 - 시(hour), 분(minute), 초(second) 순 서로 1바이트씩 기술	-
		8	-	레지스터를 구성하기 위하여 추가된 비트	-

개폐형 구동기와 관련된 상태 정보 및 상태 관련 속성 정보는 표 25와 같이 제공한다.

표 25 - 개폐형 구동기의 상태 정보 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
상태 정보	동작 상태 (Operation Status)	M	OPENING(여는 중)/CLOSING(닫는 중)/ OPEN(열림)/CLOSED(닫힘)
	동작 위치 (Operation Position)	O	현재 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	O	동작 속도

응답 패킷의 레지스터 값에 포함되는 내용은 표 25의 구동기 동작 상태와 상태 관련 속성정보 부분으로, 이 표준에서는 동작 상태와 동작 위치가 적용된다. 바이너리 인코딩 포맷은 표 26과 같다. 이때, RS485 모드버스는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 상태 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 26 - 개폐형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 상태 (Operation Status)	2	8	부호 없는 정수형 (unsigned integer)	구동기 상태 정보 - 0x10: OPENING(열리는 중) - 0x11: CLOSING(닫히는 중) - 0x12: OPEN(열림) - 0x13: CLOSED(닫힘)	- 개폐형 구동기는 OPENING/CLOSING/OPEN/CLOSED
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 위치 (Operation Position)	2	8	부호 없는 정수형 (unsigned integer)	동작 위치 정보, 비율로 표시 - 0 ~ 100	- 개폐형 구동기에 사용 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
		8	-	레지스터를 구성하기 위하여 추가된 비트	-

#### ■ 데이터 지정 기능(SetData) - 단일 레지스터

온실 통합 제어기는 마스터로 동작하고, 구동기 노드는 슬레이브로 동작한다. 온실 통합 제어기(마스터)는 데이터를 쓸 레지스터 주소와 레지스터 값을 포함한 기능 코드(0x06, Write Holding Register) 요청 메시지를 구동기 노드(슬레이브)에 보낸다.

구동기 노드(슬레이브)에서는 온실 통합 제어기(마스터)로 레지스터 주소 정보와 해당 레지스터에 기록한 값을 포함한 응답 메시지를 온실 통합 제어기(마스터)에게 전송한다. 정상적인 응답 메시지는 요청 메시지를 그대로 표현하게 된다. 이때 온실 통합 제어기는 해당 구동기 노드에 연결된 각 구동기의 레지스터 주소와 레지스터에 쓰고자 하는 제어 명령 데이터 값의 크기를 사전에 알고 있어야 한다.

모드버스 프로토콜에서 레지스터 데이터 값은 2바이트를 가정하고 있다. 따라서 구동기의 제어 명령 데이터 값이 2바이트를 초과하지 않는 경우에 사용할 수 있다. 구동기의 제어 명령 데이터 값이 2바이트를 초과하는 경우에는 6.3을 활용한다. 각 구동기 제어 명령에 대한 데이터 포맷은 6.3.3을 따른다.

##### a) 요청 패킷

요청 메시지의 기능 코드로 0x06를 사용하며, 레지스터 주소로 시작하는 레지스터에 레지스터 값을 쓸(write) 것을 요청한다.

표 27 - 요청 패킷의 구조

기능 코드	레지스터 주소	레지스터 값
1바이트	2바이트	2바이트

- 기능 코드: 0x06(Write Holding Register)
- 레지스터 주소: 데이터를 기록할 대상 레지스터 주소, 각 구동기에 매핑되는 레지스터 주소
- 레지스터 값: 데이터를 기록할 대상 레지스터 값, 레지스터 값에 포함되는 데이터 포맷은 6.3.3에서 기술한다.

#### b) 응답 패킷

기능 코드 0x06의 레지스터 기록 요청에 대하여 해당 레지스터의 주소와 해당 레지스터에 기록한 데이터 값을 포함하여 응답한다.

표 28 - 응답 패킷의 구조

기능 코드	레지스터 주소	레지스터 값
1바이트	2바이트	2바이트

- 기능 코드: 0x06(Write Holding Register)
- 레지스터 주소: 데이터를 기록한 대상 레지스터 주소
- 레지스터 값: 데이터를 기록한 대상 레지스터 값

#### ■ 데이터 지정 기능(SetData) - 복수 레지스터

온실 통합 제어기는 마스터로 동작하고, 구동기 노드는 슬레이브로 동작한다. 온실 통합 제어기(마스터)는 데이터를 쓸 레지스터의 시작 주소와 레지스터 수, 레지스터 데이터 값의 바이트 수, 레지스터 값을 포함한 기능 코드(0x10, Write Multiple Registers) 요청 메시지를 구동기 노드(슬레이브)에게 보낸다.

구동기 노드(슬레이브)에서는 온실 통합 제어기(마스터)로 레지스터 시작 주소 정보와 레지스터 수에 대한 정보를 포함한 응답 메시지를 온실 통합 제어기(마스터)에게 전송한다. 이때 온실 통합 제어기는 해당 구동기 노드에 연결된 각 구동기의 레지스터 주소와 레지스터에 쓰고자 하는 제어 명령 데이터 값의 크기를 사전에 알고 있어야 한다.

모드버스 프로토콜에서 레지스터 데이터 값은 2바이트를 가정하고 있다. 따라서 이 기능은 구동기의 제어 명령 데이터 값이 2바이트를 초과하는 경우에 사용할 수 있다. 구동기의 제어 명령 데이터 값이 2바이트를 초과하지 않는 경우에는 6.2를 활용한다. 예를 들어, 만약 천창 A에 대한 제어 명령의 값이 4바이트로 표현이 되는 경우, 모드버스 메시지는 천창 A로 매핑된 레지스터 시작 주소와 레지스터의 수로 2, 제어 명령에 해당하는 레지스터 값의 바이트 수, 제어 명령 레지스터 값을 포함한 0x10(Write Multiple Registers) 요청 메시지를 전송한다. 구동기를 위한 제어 명령에 대한 데이터 포맷은 6.3.3을 따른다.

#### a) 요청 패킷

요청 메시지의 기능 코드로 16(0x10)을 사용하며, 레지스터 시작 주소로 시작하여 레지스터 수에 해당하는 레지스터에 레지스터 값을 바이트 수만큼 쓸(write) 것을 요청한다.

모드버스 프로토콜에서 레지스터는 2바이트를 가정하고 있다. 따라서 특정 디바이스의 데이터 값이 2바이트를 초과하는 경우, 해당 레지스터의 시작 주소부터 연속해서 해당 데이터 값을 포함하는 크기를 기록하도록 한다. 또한, 특정 디바이스들을 연속한 주소로 할당하고, 각 디바이스에 대한 데이터 값의 크기를 아는 경우, 한 명령어로 두 개 이상의 디바이스에 대한 제어 명령을 전송할 수 있다.

표 29 - 요청 패킷의 구조

기능 코드	시작 주소	레지스터 수	바이트 수	레지스터 값
1바이트	2바이트	2바이트	1바이트	(2*N) 바이트 N = 레지스터 수

- 기능 코드: 16(0x10, Write Multiple Registers)
- 시작 주소: 데이터를 기록할 대상 레지스터의 시작 주소

- 레지스터 수: 대상 레지스터의 수(N)
- 바이트 수: 레지스터 데이터 값의 바이트 수
- 레지스터 값: 데이터를 기록할 대상 레지스터 데이터 값, 응답 패킷의 레지스터 값에 포함되는 데이터 포맷은 6.3.3에서 기술한다.

**b) 응답 패킷**

기능 코드 16(0x10)의 여러 개의 레지스터에 레지스터 기록 요청에 대하여 해당 레지스터의 시작 주소와 레지스터 수를 포함하여 응답한다.

표 30 - 응답 패킷의 구조

기능 코드	시작 주소	레지스터 수
1바이트	2바이트	2바이트

- 기능 코드: 16(0x10, Write Multiple Registers)
- 시작 주소: 데이터를 기록한 대상 레지스터의 시작 주소
- 레지스터 수: 데이터를 기록한 레지스터의 수

**c) 레지스터 데이터 포맷**

요청 패킷이나 응답 패킷의 레지스터 값에 포함되는 데이터 포맷은 다음과 같다. 레지스터 값의 크기가 2바이트를 넘지 않는 경우, 6.2의 데이터 지정 기능을 통하여 전달한다.

스마트 온실에서 각 구동기 제어 명령에 대한 데이터 포맷은 TTAK.KO-10.1045에서 정의한 내용을 따르며, 이 표준에서는 다음과 같이 적용한다.

- 데이터 전송 대역폭이 작은 관계로 바이너리 인코딩 방식을 사용한다.
- 구동기 타입을 스위치형과 개폐형으로 구분하며, 각 타입에 따른 제어 명령과 동작 시간, 제어 명령과 동작 위치 등의 속성 정보를 함께 전달하도록 한다.
- 스위치형 구동기의 동작 기간 속성 정보와 개폐형 구동기의 동작 속도 속성 정보는 적용하지 않는다.

스위치형 구동기와 관련된 제어 명령 및 제어 명령 관련 속성 정보는 표 31과 같이 제공한다.

표 31 - 스위치형 구동기의 제어 명령 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
제어 정보	동작 명령 (Operation Command)	M	ON(작동)/OFF(정지)
	동작 기간 (Operation Duration)	O	동작 유지 기간 날짜와 시간 정보를 'P4Y2M2DT2H2M2S' 형태로 기술
	동작 시간 (Operation Time)	O	동작 유지 시간 시(hour), 분(minute), 초(second) 정보를 기술

응답 패킷의 레지스터 값에 포함되는 내용은 표 31의 구동기 동작 제어 명령과 제어 관련 속성 정보 부분으로, 이 표준에서는 동작 명령과 동작 시간이 적용된다. 바이너리 인코딩 포맷은 표 32와 같다. 이때, RS485 모드버스는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 제어 명령 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

표 32 - 스위치형 구동기 레지스터 값 표현 바이너리 포맷

필드	바이트	비트	타입	설명	비고
동작 명령 (Operation Command)	2	8	부호 없는 정수형 (unsigned integer)	구동기 제어 명령 - 0x00: ON(작동) - 0x01: OFF(정지)	스위치형 구동기는 ON/OFF
		8	-	레지스터를 구성하기 위하여 추가된 비트	-
동작 시간	4	24	부호 없는	동작 시간	-

(Operation Time)		정수형 (unsigned integer)	시(hour), 분(minute), 초(second) 순서로 1바이트씩 기술	
	8	-	레지스터를 구성하기 위하여 추가된 비트	-

개폐형 구동기와 관련된 제어 명령 및 제어 명령 관련 속성 정보는 표 33과 같이 제공한다.

표 33 - 개폐형 구동기의 제어 명령 및 관련 속성 정보

구분	분류	필수/선택(M/O)	데이터 값
제어 정보	동작 명령 (Operation Command)	M	OPEN(열기)/CLOSE(닫기)
	동작 위치 (Operation Position)	O	원하는 동작 위치 정보, 0 ~ 100 - 0 %는 완전히 닫힌 상태 - 100 %는 완전히 열린 상태
	동작 속도 (Operation Speed)	O	동작 속도

요청 패킷의 레지스터 값에 포함되는 내용은 표 33의 구동기 동작 동작 제어 명령과 제어 관련 속성 정보 부분으로, 이 표준에서는 동작 명령과 동작 위치가 적용된다. 바이너리 인코딩 포맷은 표 32와 같다. 이때, RS485 모드버스는 레지스터 단위로 이루어지고, 레지스터는 2바이트로 구성되므로, 동작 제어 명령 및 그 속성 정보는 레지스터 단위로 구성된다. 인코딩시 빅엔디언(big-endian) 방식으로 표현한다.

## 나-4) [국가표준] 농업용 트랙터 및 기계류 - 직렬 제어 및 통신 데이터 네트워크 - 제3부: 데이터링크 계층

### (가) 연구 수행 개요

ISO/TC23 SC19에서 개발된 ISO 11783 시리즈는 총 14편으로 구성되어 있으며, 농업용 기계류의 전자 정보를 위해 널리 통용되는 표준이다. 국내에서는 ISO TC23 국내위원회를 통해, 1부와 2부를 국가표준으로 부합화한 바 있다.

널리 통용되는 국제 표준임에도 불구하고, 통신분야에 관련된 내용들이 기술됨에 따라, 제 3부이후로는 아직까지 부합화 시도가 없었기 때문에, 본 과제를 통해 제3부 (데이터링크 계층)와 제4부(네트워크 계층)에 대한 국가표준 부합화를 시도하였다. 현재는 ISO TC23 SC19 국내위원회를 통해 회람중에 있으며, 2022년도에 KS로 채택될 예정이다.

### (나) 연구 주요 내용

데이터 링크 계층은 물리적 링크를 통해 안정적인 데이터 전송을 가능케 한다. 이 계층은 기존 CAN 데이터프레임을 전송하기 위한 필수 동기화, 시퀀스 제어, 오류 제어 및 흐름 제어로 이루어진다. 흐름 제어는 일관된 메시지 프레임 형식을 통해 이루어진다.

메시지 프레임 형식은 CAN 요구사항을 준수해야 한다. 이 표준 전반에 걸쳐 참조하는 CAN 규격은 ISO 11898-1에 기술되어 있다. CAN 문서의 정보-라우팅-관련 부분에는 제어 기능 주소가 사용되지 않는다고 명시되어 있다. 이것은 일부 CAN 응용에는 해당되지만, ISO 11783에는 해당되지 않는다. ISO 11783 네트워크 정의에는 여러 제어 기능들이 동일한 CAN 식별자 필드를 사용하는 것을 방지하기 위해 제어 기능 주소를 사용할 것이 요구된다. 추가적인 많은 요구 사항들이 CAN에는 정의되어 있지 않지만, ISO 11783에 존재한다.

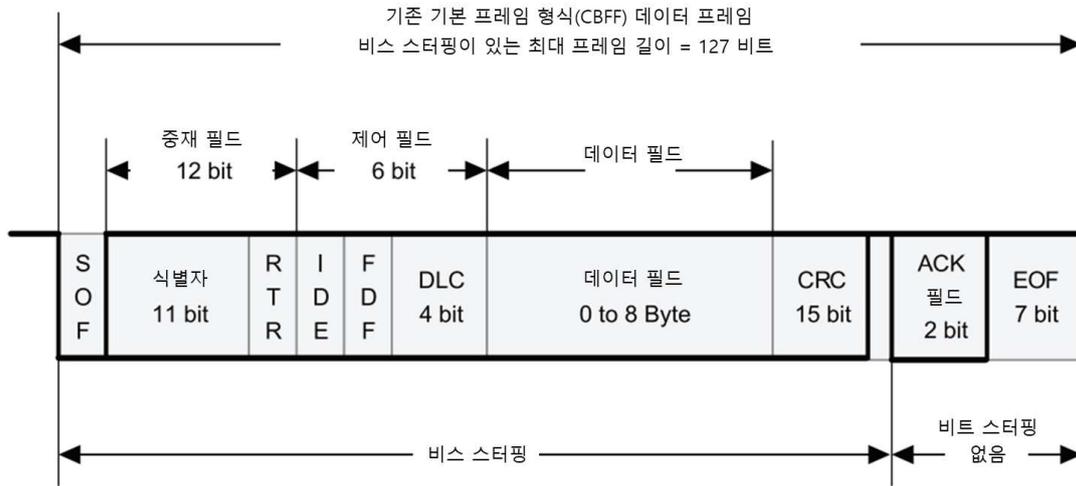
ISO 11898-1은 CBFF (Classical Basic Frame Format, 기존 기본 프레임 형식)와 CEFF라는 두 가지 기존 프레임 형식을 정의한다. ISO 11898-1 호환성은 서로 다른 형식을 인식할 수 있는 특정 비트 코딩을 사용하여 잠재적으로 단일 네트워크에 서로 다른 형식을 갖는 메시지가 존재할 수 있다는 것을 암시한다. 그러나 ISO 11783은 CEFF를 사용하는 표준화된 통신에 대한 전반적인 전략만을 정의한다. 모든 CBFF 메시지는 이 표준에 정의된 규칙에 따라 독점적으로 사용되어야 하며, 모든 FD 프레임 형식은 ISO 11783 네트워크에서 사용되지 않는다.

따라서 ISO 11783 컨트롤러는 CEFF를 사용해야 한다. 이 표준을 따르는 CBFF 메시지일 경우에만, 네트워크에 공존할 수 있다.

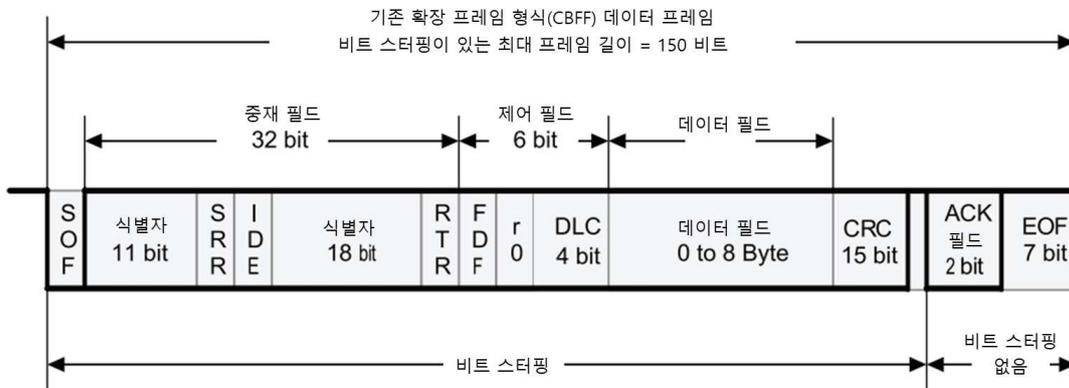
그림 1에서와 같이, 기존 CAN 데이터프레임은 다른 비트 필드로 구분 해석된다. CBFF와 CEFF 메시지간 중재와 제어를 위한 필드 비트 수와 구분 분석은 상이하다. CBFF 메시지는 그림 1 a)에서와 같이 중재 필드에 11 개의 식별자 비트를 포함하는 반면, CEFF 메시지의 중재 필드는 그림 1b)에서와 같이 29 개의 식별자 비트를 포함한다. ISO 11783은 CAN 메시지 프레임 형식의 중재 필드에 식별자 비트를 추가로 정의한다. 이들 정의는 표 1에 나열되었다.

다음 그림에 표시된 CEFF 메시지는 단일 PDU (프로토콜 데이터 단위)를 포함한다. PDU는 응용 계층에서 제공하는 정보를 따르도록 다음과 같이 7개의 사전에 정의된 필드로 구성된다.

- 우선 순위
- 확장 데이터 페이지 (EDP, Extended Data Page)
- 데이터 페이지 (DP, Data Page)
- PDU 형식 (PF, PDU Format)
- 수신 주소 (DA, Destination Address), 그룹 확장 (GE, Group Extension) 또는 독점일 수 있는 특정 PDU (PS, PDU Specific)
- 송신지 주소 (SA, Source Address)
- 데이터



a) 기존 기본 프레임 형식 (CBFF)



b) 기존 확장 프레임 형식 (CEFF)

그림 1 - 기존 CAN 데이터프레임

필드들은 하나의 기존 CAN 데이터프레임으로 패키징된 후 물리적 미디어를 통해 다른 네트워크 컨트롤러로 전송된다. ISO 11783이 지원하는 OSI 모델의 계층은 그림 2에 제시된다. 일부 매개변수 그룹 정의는 정보를 전송하기 위하여 하나 이상의 기존 CAN 데이터프레임을 요구할 수 있다.

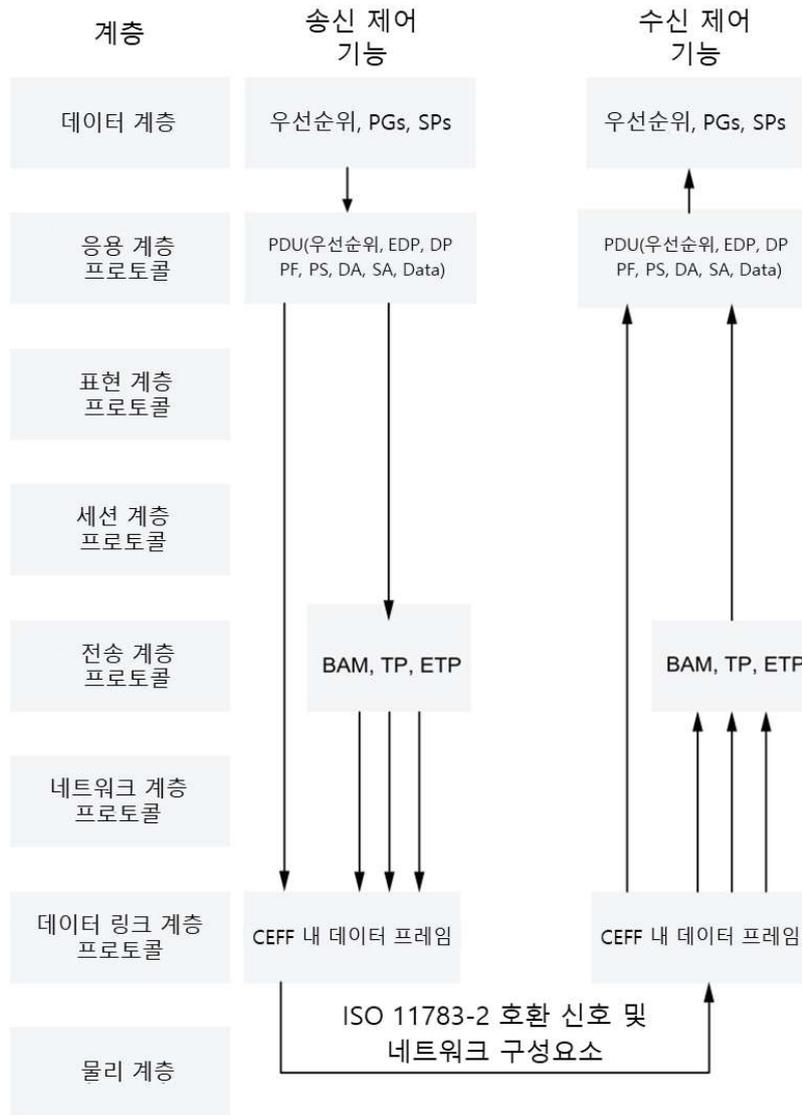


그림 2 - ISO 11783에 따른 OSI 모델 적용

다음의 표 1은 1) CAN에 대한 29 비트 식별자, 2) ISO 11783에 대한 29비트 식별자, 3) CAN에 대한 11비트 식별자에 대한 중재 필드와 제어 필드를 보여주며, 4) ISO 11783 네트워크 상에서의 11비트 식별자 사용을 보여준다.

ISO 11783에서 CAN 데이터프레임 데이터 필드는 바이트 1에서 8로 기술된다. 바이트 1의 MSB(최상위 비트)인 비트 8은 데이터 길이 코드 (DLC)에 가장 가깝게 전송되는 첫 번째 비트이다. 바이트 8의 LSB(최하위 비트)인 비트 1은 전송할 마지막 데이터 비트이며, CRC (Cyclic Redundancy Check, 순환중복검사) 필드에 가장 가깝다. 확장 데이터 페이지 (EDP)가 1이고 데이터 페이지 (DP)가 1이면, CAN 프레임은 ISO 15765-2 형식의 프레임으로 식별된다. ISO 15765-2는 CAN을 통한 진단 통신(DoCAN)을 정의한다. 따라서, 이 특정 CAN 프레임 형식은 ISO 11783을 따르지 않고 ISO 15765-2를 따라 처리해야 한다.

표 1 - CAN 중재 필드와 제어 필드에 대한 ISO 11783 매핑

비트 번호	29비트 식별자		11 비트 식별자	
	CAN	ISO 11783	CAN	ISO 11783b
1	SOF	SOFa	SOF	SOFa
2	ID28	P3	ID28	P3
3	ID27	P2	ID27	P2
4	ID26	P1	ID26	P1
5	ID25	EDP	ID25	ID8a
6	ID24	DP	ID24	ID7a
7	ID23	PF8	ID23	ID6a
8	ID22	PF7	ID22	ID5a

9	ID21	PF6	ID21	ID4a
10	ID20	PF5	ID20	ID3a
11	ID19	PF4	ID19	ID2a
12	ID18	PF3	ID18	ID1a
13	SRR (r)	SRRa	RTR (x)	RTRa (d)
14	IDE (r)	IDEa	IDE (d)	IDEa
15	ID17	PF2	FDf (d)	FDfa
16	ID16	PF1	DLC4	DLC4
17	ID15	PS8	DLC3	DLC3
18	ID14	PS7	DLC2	DLC2
19	ID13	PS6	DLC1	DLC1
20	ID12	PS5		
21	ID11	PS4		
22	ID10	PS3		
23	ID9	PS2		
24	ID8	PS1		
25	ID7	SA8		
26	ID6	SA7		
27	ID5	SA6		
28	ID4	SA5		
29	ID3	SA4		
30	ID2	SA3		
31	ID1	SA2		
32	ID0	SA1		
33	RTR (x)	RTRa (d)		
34	FDf (x)	FDfa (d)		
35	r0 (d)	r0a		
36	DLC4	DLC4		
37	DLC3	DLC3		
38	DLC2	DLC2		
39	DLC1	DLC1		
SOF	프레임비트의 시작시작	EDP	ISO 11783에 따른 확장 데이터 페이지	
ID##	식별자 비트 번호 (#)	SA#	ISO 11783에 따른 송신지 주소 비트	
SRR	원격 요청 대체	DP	ISO 11783에 따른 데이터 페이지	
RTR	원격 전송 요청 비트	PF#	ISO 11783에 따른 PDU 형식 비트	
IDE	Identifier Extension bit 식별자 확장 비트	PS#	ISO 11783에 따른 PDU 특정 비트	
FDf	FD 형식 지시자	(d)	우성 비트	
r#	CAN 예약 비트 번호 (#)	(r)	열성 비트	
DLC#	데이터 길이 코드 비트 번호 (#)	(x)	메시지에 따른 비트 상태	
P#	ISO 11783에 따른 우선권 비트 번호 (#)			
a	ISO 11783에서 변경되지 않은 CAN-정의 비트.			
b	독점 11비트 식별자의 필수 형식.			

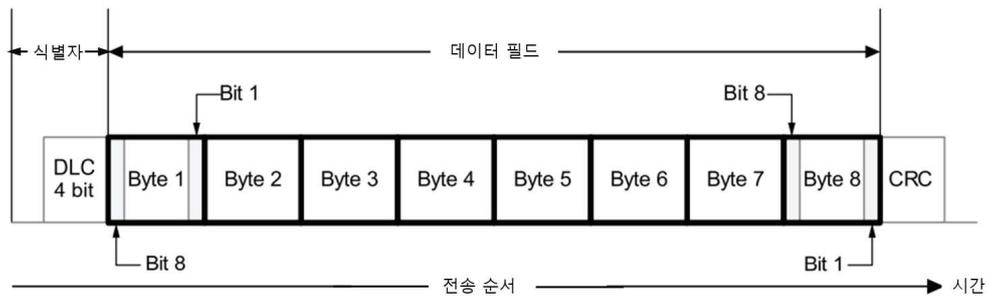


그림 3 - 기존 CAN 데이터 필드

1) 매개변수 그룹 번호 (PGN)

기존 CAN 데이터프레임의 데이터 필드에서, 매개변수 그룹을 식별할 필요가 있을 때에는 24 비트로 표현된다. 이 24 비트의 값은 최하위 바이트 (LSB)가 가장 먼저 전송된다. 즉, 최상위 바이트 (MSB)가 세 번째로, 중간 바이트가 두 번째로, LSB가 첫 번째 순으로 전송된다. 24 비트의 PGN은, 0으로 설정된 6비트, 확장 데이터 페이지 비트, 데이터 페이지 비트, PDU 형식 필드 (8 비트) 및 PDU 특정 필드 (8 비트)로 구성된다.

비트 필드를 PGN으로 변환하는 방법은 다음과 같다. 우선, PGN의 6개 MSB를 0으로 설정한다. 다음, 10개 비트에 확장 데이터 페이지 비트, 데이터 페이지 비트 및 PDU 형식 필드를 복사한다. 마지막으로, LSB는 만일 PF 값이 240 (F016) 미만이면 0으로 설정하고, 그렇지 않으면 PS 필드의 값으로 설정한다.

표 2 - 매개변수 그룹 번호 (PGN) 예

PGN 구성 성분					PGN		할당 가능한 PG 수	누적 PG 수	ISO 또는 제조사 할당	
PGN (MSB) CAN 데이터 프레임에서 세번째로 보내는 바이트 1		PGN CAN 데이터 프레임에서 두번째로 보내는 바이트 2		PGN(LSB) CAN 데이터 프레임에서 첫번째로 보내는 바이트 3		Dec10				Hex16
비트 8-3	EDP 비트2	DP 비트1	PF 비트8-1	PS 비트8-1						
0	0	0	0	0	0	00000016	239	239	ISO	
0	0	0	238	0	60,928	00EE0016				
0	0	0	239	0	61,184	00EF0016	1	240	MF	
0	0	0	240	0	61,440	00F00016			ISO	
							3,840			
0	0	0	254	255	65,279	00FEFF16		4,080		
0	0	0	255	0	65,280	00FF0016				
							256		MF	
0	0	0	255	255	65,535	00FFFF16		4,336		
0	0	1	0	0	65,536	01000016				
0	0	1	238	0	126,464	01EE0016	239		ISO	
0	0	1	239	0	126,720	01EF0016	240	4,576	MF	
0	0	1	240	0	126,976	01F00016				
							4,096		ISO	
0	0	1	255	255	131,071	01FFFF16		8,672		

2) ISO 11783에서의 ISO 11898-1 CBFF 메시지 지원

ISO 11783 네트워크상의 컨트롤러는 CBFF (11 비트 식별자) 메시지 형식을 지원할 수 있다. 비록 이들은 ISO 11783 메시지 구조와 호환되지는 않지만, 두 형식이 공존하기 위한 최소 수준의 정의가 제공된다. 이 최소 정의를 통해, 이 형식을 사용하는 컨트롤러들이 다른 컨트롤러들에게 영향을 끼치지 않게 한다. CBFF 메시지는 독점으로 정의된다. 11 비트 식별자 필드는 다음과 같이 구문 분석된다. 3개의 최상위 비트는 우선 순위 비트로 사용된다. 8개의 최하위 비트는 PDU의 SA를 식별한다.

하나의 기본 프레임과 하나의 확장 프레임으로 구성된 두 개의 메시지가 동시에 버스에 접근할 경우 잘못된 버스 중재가 발생할 수 있다. 확장 프레임 메시지보다 기본 프레임 메시지에서 소스 주소 (SA)의 상대적 우선 순위가 높다. 11 비트 식별자 (기본 프레임)를 갖는 메시지는, 29 비트 식별자 (확장 프레임) 메시지의 확장 데이터 페이지 비트, 데이터 페이지 비트 및 PDU 형식보다 높은 우선 순위를 나타내는 SA를 가질 수 있다. 올바른 버스 중재를 위해서는 3개의 우선 순위 비트를 사용해야 한다.

3) 프로토콜 데이터 단위 (PDU)

■ 일반

응용 및/또는 네트워크 계층은 프로토콜 데이터 단위에 따라 일련의 정보를 제공한다. 프로토콜 데이터 유닛은 전송하고자 하는 각각의 기존 CAN 데이터에 필수적인 정보를 구성하기 위한 프레임워크를 제공한다. ISO 11783 네트워크의 프로토콜 데이터 단위 (PDU)는 다음과 같이 5.1.2에 나열된 7개의 필드로 구성된다. 이들 필드는 하나 이상의 기존 CAN 데이터프레임으로 패키징되고, 물리적 매체를 통해 다른 네트워크 컨트롤러로 전송된다. 기존 CAN 데이터프레임에는 하나의 PDU만 존재한다.

기존 CAN 데이터프레임 필드의 어떤 비트들은 PDU 정의에서 제외된다. 왜냐하면, 이들은 전적으로 CAN 규격에 의해 제어되고, 데이터링크 계층 위의 모든 OSI 계층에 표시되지 않기 때문이다. 여기에는 SOF, SRR, IDE, RTR, FDF, CRC, ACK 및 EOF 필드와, 제어 필드의 일부가 포함된다. 이들은 CAN 프로토콜 규격에 따라 정의되며 ISO 11783에 의해 수정되지 않은 상태로 유지된다.

	우선순위,	EDP	DP	PF	PS	SA	데이터
비트 수	...3...	...1...	...1...	...8...	...8...	...8...	...64...

그림 3 - PDU 필드

### ■ 우선 순위 (P)

우선 순위 비트는 단지 버스 전송을 위한 메시지 대기 시간을 최적화하는데 사용된다. 수신 컨트롤러는 이들 비트들을 전역적으로 마스킹해야 한다 (무시됨). 모든 메시지의 우선 순위는, 가장 높은 0에서 가장 낮은 7까지 설정할 수 있다. 모든 제어 지향 메시지에 대한 기본값은 3이다. 기타 다른 모든 정보, 독점, 요청 및 NACK 메시지에 대한 기본값은 6이다. 이를 통해 새로운 PGN 값을 할당하거나 버스 트래픽이 변경될 때 우선 순위를 높이거나 낮출 수 있다. 권장 우선 순위는 응용 계층 표준에 추가될 때마다 각각의 PGN에 할당된다. 하지만, 제조업체가 필요할 때 네트워크를 조정할 수 있도록 우선 순위 필드는 재프로그래밍 될 수 있어야 한다.

### ■ 확장 데이터 페이지 (EDP)

확장 데이터 페이지 (EDP) 비트는 데이터 페이지 비트와 함께, 기존 CAN 데이터프레임의 CAN 식별자 구조를 결정하기 위해 사용된다. 전송할 때 모든 ISO 11783 메시지는 확장 데이터 페이지 비트를 0으로 설정해야 한다. 향후 새로운 정의를 통해 PDU 형식 필드의 확장, 새로운 PDU 형식의 정의, 우선 순위 필드의 확장 또는 주소 공간의 증가가 가능하다.

### ■ 데이터 페이지 (DP)

기존 CAN 데이터프레임의 CAN 식별자 구조를 결정하기 위해 데이터 페이지 (DP) 비트는 EDP 비트와 함께 사용된다. EDP를 0으로 설정하고, DP 비트는 PGN 설명을 위한 페이지 0과 페이지 1중 하나를 선택한다.

### ■ PDU 형식 (PF)

PDU 형식(PF)은 PDU 형식을 결정하는 8 비트 필드이며, 기존 CAN 데이터 필드에서 사용되는 PGN을 결정하기 위해 사용되는 필드이다. PGN은 명령, 데이터, 일부 요청, 승인 또는 부정 승인을 식별하거나 표시하는데 사용될 뿐만 아니라, 정보 전달을 위해 사용되는 하나 이상의 기존 CAN 데이터프레임에 대한 정보를 식별하거나 표시하는데 사용된다. 만일 8 데이터 바이트에 들어갈 수 있는 것보다 정보가 클 경우, 다중-패킷 메시지를 사용하여 보낸다. 만일 데이터 바이트가 8개 이하인 경우, 단일 기존 CAN 데이터프레임을 사용한다. PGN은 하나 이상의 매개변수를 나타낼 수 있다. 여기서 매개변수란 분당 엔진 회전 수와 같은 데이터이다. 하나의 매개변수에 PGN 레이블을 사용할 수 있지만, 여러 매개변수를 그룹화하여 8 바이트의 데이터 필드 모두를 사용하는 것이 권장된다.

두 개의 독점 PGN 정의를 통해 PDU1와 PDU2 형식 모두를 사용할 수 있다. 독점 정보의 해석은 제조업체에 따라 다르다.

### ■ PDU 특정 (PS)

PDU 특정(PS) 필드는 PDU 형식에 따라 정의가 달라지는 8비트 필드이며, DA 또는 GE 필드 여부를 결정한다. DA 필드는 메시지가 전송되는 특정 주소를 정의한다. 다른 주소를 갖는 컨트롤러는 이 메시지를 무시해야 한다. 전역 대상 주소 (255)일 경우, 모든 컨트롤러는 메시지의 수신자인 것처럼 수신하고 응답해야 한다.

PF 필드의 최하위 4 개 비트와 함께 GE 필드는 데이터 페이지 당 4096 개의 매개변수 그룹을 제공한다. 이것은 GE 형식 PDU (PDU2)를 통해서만 사용할 수 있다. 또한, 목적지-특정 형식 PDU (PDU1 형식)에서만 사용할 수 있도록 각 데이터 페이지에 240개의 매개변수 그룹이 제공된다. 현재 사용 가능한 두 개의 데이터 페이지를 사용하여 총 8,672개의 매개변수 그룹을 정의할 수 있다.

### ■ 송신지 주소 (SA)

송신지 주소 (SA) 필드의 길이는 8 비트이다. 네트워크에는 주어진 송신지 주소를 갖는 제어 기능이 하나만 존재해야 한다. 주소 관리 및 할당, SA 복제 방지 절차에 대해서는 ISO 11783-5를 참조한다.

### ■ 데이터 필드

#### a) 0에서 8 바이트까지의 데이터

주어진 매개변수 그룹을 표현하기 위해 8 바이트 이하의 데이터가 필요한 경우, 기존 CAN 데이터프레임의 8 바이트 데이터를 모두 사용할 수 있다. 향후 확장 가능성이 있는 모든 PGN 할당에 대해 8 바이트를 할당하거나 예약하는 것이 권고된다. 이것은 매개변수를 쉽게 추가하고 데이터 필드의 일부만 정의하는 이전 개정판과의 비호환성을 피할 수 있는 수단을 제공한다. 만일 PGN과 연관된

데이터의 바이트 수가 지정된다면 변경이 불가하다 (원래 그렇게 정의되지 않는 한, 다중-패킷이 될 수도 없다). CAN 데이터 길이 코드 (DLC)가 8 바이트 이하일 때, 정의된 매개변수 그룹 "데이터 길이"값으로 설정된다. 그렇지 않고 PG 데이터 길이가 9 바이트 이상이면, CAN DLC가 8로 설정된다. 예를 들어, REQUEST PGN 59,904는 PG 데이터 길이가 3 바이트이기 때문에 CAN DLC가 3으로 설정된다. 개별 그룹 기능은 CAN 식별자가 항상 동일하기 때문에 동일한 데이터 필드 길이를 사용해야 하는 반면, 기존 CAN 데이터 필드는 특정 그룹 하위 기능을 전달하는데 사용된다. 이러한 그룹 기능들은 기존 CAN 데이터 필드를 기반으로 하는 다양한 해석이 필요하다.

b) 8 바이트보다 큰 데이터

주어진 매개변수 그룹을 표현하기 위해 8개 이상의 데이터 바이트가 필요한 경우, 이러한 데이터의 통신은 여러 기존 CAN 데이터 프레임을 통해 수행된다. 다중-패킷이라는 용어는 이러한 유형의 매개변수 그룹을 설명하는 데 사용된다. 특정 인스턴스에서, 전송할 데이터 바이트가 9 개 미만이고 다중-패킷 가능으로 정의된 매개변수 그룹은, DLC를 8로 설정하여 단일 기존 CAN 데이터프레임으로 전송한다. 특정 매개변수 그룹에 전송할 데이터 바이트가 9개 이상인 경우 전송 프로토콜 기능 중 하나가 사용된다. 전송 기능 연결 관리 기능은 다중 패킷 매개변수 그룹의 통신을 설정하고 종료하는 데 사용된다. 전송 프로토콜의 데이터 전송 기능은 "패킷화된" 데이터를 포함하는 일련의 기존 CAN 데이터 프레임(패킷)에서 데이터 자체를 통신하는 데 사용된다. 또한 전송 프로토콜 기능은 대상별 전송을 위한 흐름 제어와 핸드셰이킹 기능을 제공한다.

특정 다중 패킷 응답과 관련된 모든 기존 CAN 데이터 프레임은 DLC 값으로 8을 가져야 한다. 사용하지 않는 모든 데이터 바이트는 "사용할 수 없음"으로 설정된다. 패킷당 바이트 수는 고정되어 있지만, ISO 11783은 가변 및/또는 고정된 수의 패킷이 있는 다중 패킷 메시지를 정의한다. 활성 진단 코드에 대한 PGN은 다양한 패킷 수를 갖는 다중-패킷 메시지의 예이다. 다중-패킷으로 정의된 매개변수 그룹은 전송할 데이터 바이트 수가 8개를 초과하는 경우에만 전송 프로토콜을 사용한다.

4) 프로토콜 데이터 단위 (PDU) 형식

다음 그림에서와 같이 사용 가능한 PDU 형식은 PDU1 (PS = DA)과 PDU2 (PS = GE)로 정의된다. PDU1은 특정 대상 주소를 갖는 기존 CAN 데이터프레임을 허용하며 (제어 기능). PDU2는 특정 대상 주소를 갖지 않는 기존 CAN 데이터프레임의 통신만을 허용한다. 더 많은 가능한 매개변수 그룹 번호 조합을 제공하는 동시에 목적지별 통신을 제공하기 위해, 두 개의 개별적인 PDU 형식이 생성된다. 독점 매개변수 그룹 정의를 할당하여 두 PDU 형식이 독점적 통신에 사용될 수 있다. 독점 통신을 위한 표준화된 방법을 통해 식별자 사용에서 발생할 수 있는 충돌을 방지할 수 있다. PDU1과 PDU2 형식이 모두 사용될 수 있도록 독점 매개변수 그룹 번호 정의가 정의된다. 독점 정보에 대한 해석은 제조업체마다 다르다.

	우선순위,	EDP,	DP,	PF,	PS(DA),	SA,	데이터
비트 수	...3...	...1...	...1...	...8...	...8...	...8...	...64...

a) PDU1

	우선순위,	EDP,	DP,	PF,	PS(GE),	SA,	데이터
비트 수	...3...	...1...	...1...	...8...	...8...	...8...	...64...

b) PDU2

그림 4 - 가용 PDU 형식

■ PDU1 형식

PDU1 형식은 특정 또는 전역 대상으로 보낼 해당 매개변수 그룹을 제공한다. PS 필드는 DA를 포함한다. PDU1 형식 메시지는 요청하지 않은 메시지로 요청하거나 보낼 수 있다. PDU1 형식 메시지는 PF 필드에 의해 결정된다. 해당 필드의 값이 0 ~ 239 일 때 메시지는 PDU1 형식이다. PDU1 메시지의 형식은 다음 그림에 도시되어 있다.

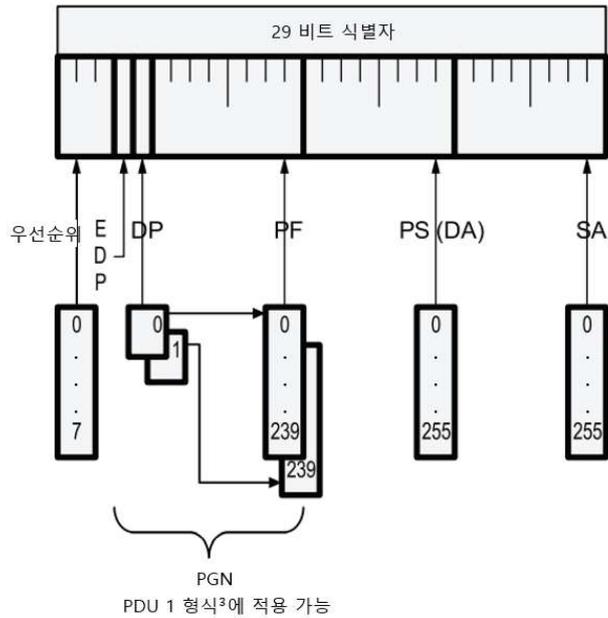


그림 6 - PDU1 형식

대상 (PDU1)과 최소 지연 시간이 필요한 매개변수 그룹은 PF = 0에서 시작하여 x (또는 x1)로 증가한다. 지연 시간이 중요하지 않은 대상을 요구하는 매개변수 그룹은 PF = 239에서 시작하여 x (또는 x1)로 감소한다. 239 (확장 데이터 페이지 비트 = 0 및 데이터 페이지 비트 = 0)와 동일한 PF 가 독점 사용을 위해 할당된다. 이 경우 PS 필드는 목적지 주소이다 (5.4.6 참조). 독점 A의 PGN은 61,184이다.

■ PDU2 형식

PDU2 형식은 매개변수 그룹을 전역 메시지로써 통산하는 데만 사용할 수 있다. PDU2 형식 메시지는 요청하지 않은 메시지로 요청하거나 보낼 수 있다. PGN이 할당될 때 PDU2 형식의 선택은 PGN이 특정 목적지로 향할 수 있는 것을 방지한다. PS 필드는 GE를 포함한다.

PDU2 형식 메시지는 PF 값이 240 ~ 255인 것으로 정의된다. PDU2 메시지 형식은 다음 그림과 같다.

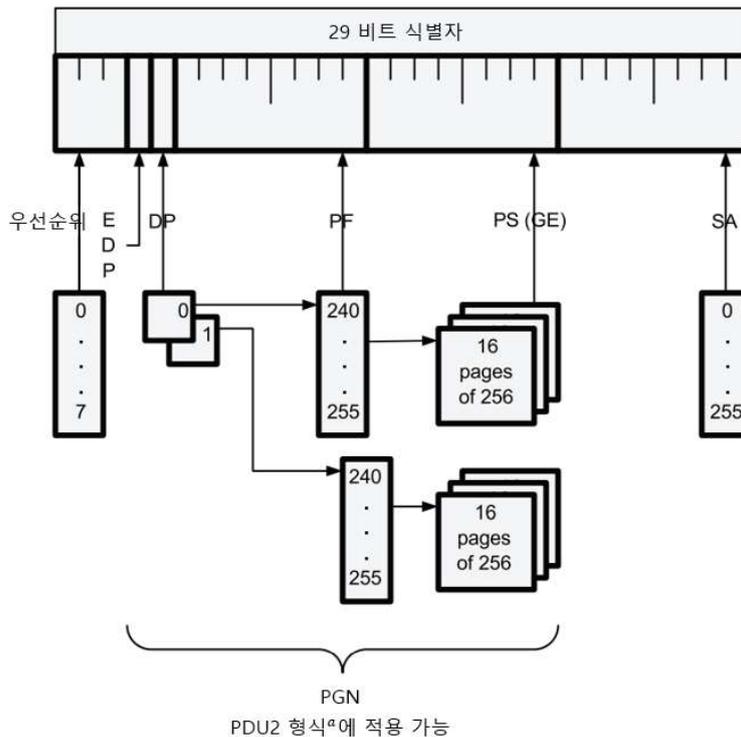


그림 7 - PDU2 형식

빠른 업데이트 속도 (일반적으로 100ms 미만)로 보내는 메시지의 PGN은 PF = 240에서 시작하여 y (또는 y1)로 증가한다. 요청만 받거나, 변경 사항이 있을 때 전송되거나, 느린 업데이트 속도 (일반적으로 100ms 이상)로 전송되는 메시지의 PGN은 PF = 254에서 시작하여 y (또는 y1)로 감소한다. 255 (확장 데이터 페이지 비트 = 0 및 데이터 페이지 비트 = 0)를 갖는 PF가 독점 사용을 위해 할당된다. PS 필드는 각 제조업체에서 정의하고 사용하도록 남겨둔다 (5.4.6절 참조). 독점 B에 대한 PGN은 65,280 ~ 65,535 범위를 포함한다.

## 5) 메시지 유형

현재 지원되는 메시지 유형에는 다음과 같이 5가지가 있다.

- 명령
- 요청
- 브로드캐스트/응답
- 확인
- 그룹 기능

특정 메시지 유형은 할당된 PGN에 의해 인식된다. RTR 비트 (원격 프레임의 CAN 프로토콜에 정의됨)는 열성 상태 (논리 1)에서 사용되지 않는다. 따라서, ISO 11783 네트워크에서는 원격 전송 요청(RTR = 1)을 사용할 수 없다.

RTR은, 데이터 바이트없이 네트워크에서 단지 CAN 식별자를 알림으로써, 특정 CAN 객체를 "요청"하기 위한 것이다. ISO 11783은 CAN 식별자의 일부를 메시지 우선 순위나 소스 주소로 사용하기 때문에, 이 메커니즘은 충돌이 발생할 수 있다. 장치는 다른 장치의 SA를 사용하여 메시지를 보낼 수 없다. ISO 11783에는 별도의 요청 메커니즘이 정의되어 있다. 5.4.3을 참조한다.

기존 CAN 데이터프레임의 데이터 필드에 나타나는 다중-바이트 매개변수는 LSB에 먼저 배치되어야 한다. 해당되는 경우 예외가 표시된다 (예, ASCII 데이터). 만일 2 바이트 매개변수가 기존 CAN 데이터프레임의 7 및 8번째 바이트에 위치하는 경우, CAN 데이터 프레임에서 LSB는 7번째 바이트에 배치되고 MSB는 8번째 바이트에 위치된다.

### ■ 명령

명령 메시지 유형은 소스로부터 특정 혹은 전역 대상을 명령하는 매개변수 그룹들을 분류한다. 이 명령 메시지 유형을 수신한 대상은 특정 작업을 수행해야 한다. PDU1 (PS = DA)와 PDU2 형식 (PS = GE) 메시지 모두 명령에 사용할 수 있다. 명령 모드의 예로는 전송 제어, 주소 요청 및 토크/속도 제어기가 있다.

### ■ 요청

PGN으로 식별되는 요청 메시지 유형은, 전역 또는 특정 대상으로부터 정보를 요청하는 기능을 제공한다. 하나의 대상을 특정한 요청을 대상-특정 요청이라 한다. 아래 정보는 요청 PGN 매개변수 그룹에 PGN을 할당한다. 정보는 ISO 11783의 매개변수 그룹 정의와 동일한 형식이다. 다음은 PGN을 전역 또는 특정 대상으로 보낼지를 결정하기위한 일반적인 동작 규칙이다.

- a) 전역 주소로 요청이 전송되면, 전역 주소로 응답이 전송된다. NACK는 전역 요청에 대한 응답으로 사용되지 않는다.
- b) 특정 주소로 요청이 전송되면, 특정 주소로 응답이 전송된다. PGN을 지원하지 않는 경우, NACK가 필요하다. 데이터 길이가 8 바이트 이상인 경우, 특정 주소에 대한 응답을 위해 전송 프로토콜 RTS / CTS를 사용한다.
- c) 주기적인 브로드캐스트 또는 요청하지 않은 메시지의 경우, PDU1 또는 PDU2 형식 PGN을 전역 또는 특정 대상 주소로 보낼 수 있다.
- d) 위에서 볼 수 있듯이, 이러한 규칙에 대한 예외가 존재한다. PGN이 정의된 해당 문서의 절에 명시되어 있다.

특정 PGN 값이 지원되지 않음을 나타내는 NACK인 경우에도 지정된 대상(전역 아님)에서 항상 응답이 필요하다. 특정 PGN이 제어 기능에 의해 지원되지 않는 경우, 전역 요청에 대해 NACK로 응답하지 않는다.

요청 PGN은 특정 매개변수 그룹이 지원되는지 확인하기 위해 특정 대상 주소로 전달 될 수 있다 (즉, "요청 대상 주소가 특정 그룹을 전송하는가?"). 요청에 대한 응답은 PGN의 지원 여부를 결정한다.

지원하는 경우, 수신 제어 기능은 요청된 정보를 전송해야 한다. 만일 승인 PGN이 적절한 경우, 제어 바이트는 0 또는 2 또는 3으로 설정된다. 지원되지 않는 경우, 수신 제어 기능은 부정적인 승인을 위해 제어 바이트가 1로 설정된 승인 PGN을 전송한다. ISO 11783 PDU 형식 및 매개변수 그룹의 나머지 부분은 적절하게 채워져야 한다. 이 방법을 통해, 제어 기능이 PG (수신시)에 작용하는지에 대한 여부를 확인할 수는 없다.

만일 제어 기능이 응답 제한 시간 내에 요청에 대한 응답(PG 또는 ACKNOWLEDGEMENT)을 수신하지 못한다면, 제어 기능은 동일한 요청을 재전송하거나 재시도 할 수 있다. 특정 요청에 대한 재시도 횟수는 2 회로 제한된다. 즉, 요청은 총 3 회까지 발생한다. 만일 제어 기능이 두 번째 재시도 후에도 요청에 대한 응답 (PG 또는 ACKNOWLEDGEMENT)을 받지 못한다면, 제어 기능은 동일한 정보에 대한 추가 요청 시도를 포기해야 한다. 그렇지 않으면 제어 기능은 동일한 정보를 요청하기 전에 오랜 시간 (몇 초가 아닌 몇 분) 동안 대기해야 한다.

#### ■ 브로드캐스트/응답

브로드캐스트/응답 메시지 유형은 다음 두 가지 중 하나일 수 있다. 하나는 제어 기능에서 요청되지 않은 정보의 브로드캐스트이며, 다른 하나는 명령 또는 요청에 대한 응답이다.

#### ■ 승인

두 가지 방식의 승인을 사용할 수 있다. 첫 번째 방식은 CAN 프로토콜에 의해 제공된다. 이것은 메시지가 적어도 하나의 컨트롤러에 의해 수신되었음을 확인하는 "프레임-내" 승인으로 구성된다. 또한 CAN 오류 프레임이 없을 경우 메시지들이 추가로 승인된다. 이들의 부재는 전원이 공급되고 연결되어 있는 다른 모든 컨트롤러가 메시지를 올바르게 수신했음을 승인한다.

두 번째 방식은 응용 계층에서 제공하는 특정 명령 또는 요청에 대한 "일반 브로드캐스트" 또는 "ACK" 또는 "NACK" 에 대한 응답이다. 승인 (Acknowledgment) 매개변수 그룹의 정의는 아래와 같다. 일부 매개변수 그룹에 필요한 승인 유형은 응용 계층에서 정의된다.

그룹 기능 매개변수 그룹 (5.6.6 참조)의 경우, 그룹 기능 값 매개변수를 사용하면 제어 기능이 인식되고있는 특정 그룹 기능을 식별 할 수 있다. 그룹 기능 값은 각 그룹 기능 PG에 고유하다. 그룹 기능 값은 0 ~ 250 범위만 사용하는 것을 권장한다.

승인의 각 형태는 승인이 전달되는 요청의 발신자의 소스 주소가 포함된 주소 승인 바이트를 포함한다. SAE J1939와의 최대 호환성을 위해 주소 승인 필드는 PS 필드와 중복된다. 바이트 5의 매개변수는 승인된 승인 (Address Acknowledged), 부정 승인된 주소(Address Negative Acknowledgement), 접근 거부된 주소(Address Access Denied) 및 사용중 주소(Address Busy)이다.

#### ■ 그룹 기능

그룹 기능 메시지 유형은 기능 그룹에 사용된다. 독점 기능, 다중-패킷 전송 기능 및 네트워크 관리 기능 (ISO 11783-5) 가상 터미널 (ISO 11783-6), 작업 컨트롤러 (ISO 11783-10), 파일 서버 (ISO 11783-13) 및 시퀀스 컨트롤러 (ISO 11783-14) 그룹 기능 메시지를 사용한다.

기능의 각 그룹은 할당된 PGN에 의해 인식된다. 함수 자체는 데이터 구조(일반적으로 데이터 필드의 첫 번째 바이트) 내에서 정의된다. 독점 그룹 기능은 서로 다른 제조업체 간의 CAN 식별자 사용 충돌을 없애는 방식으로, 독점 메시지를 전송하는 수단을 제공한다. 또한 필요시, 사용하고자 하는 독점 메시지를 수신하고 구별하는 수단을 제공한다. 이 문서에 정의된 메시지가 충분하지 않을 경우, 그룹 기능은 자체적인 요청, ACK 와/또는 NACK 메커니즘을 제공할 수도 있다.

PGN 59,904를 사용하는 요청은 그룹 기능 메시지 유형의 특정 매개변수 그룹이 지원되는지 여부를 결정하는데 사용할 수 있다. 지원하는 경우, 수신 제어 기능은 승인 (Positive Acknowledgement)의 경우 0, 접근 거부 (Access Denied)의 경우 2 또는 응답 불가(Cannot Respond)의 경우 3과 같은 제어 바이트를 사용하여 승인 PGN(Acknowledgment PGN)을 전송한다. 지원하지 않을 경우, 수신 제어 기능은 부정 승인 (Negative Acknowledgement)에 대해 제어 바이트를 1로 설정한 승인 (Acknowledgment) PGN을 전송한다. ISO 11783에 지정된 PF와 매개변수 그룹의 나머지 부분은 적절하게 채워져야 한다. 이 단락의 정의에 따라 "지원되지 않음"이라는 문구는 PG가 전송되지 않음을 의미하며, 이 방법을 사용하여 수신한 PG를 통해 제어 기능이 작동하는지 여부를 판별할 수 없다.

독점 (Proprietary) A, A2, B 메시지의 데이터 길이는 제조사별로 설정할 수 있다. 따라서 두 제조사가 동일한 GE 값을 사용할 수 있고 다른 데이터 길이의 코드를 가질 수 있다. 이 정보의 수신 제어 기능은 둘을 구별해야 한다. 이 메시지의 데이터 필드가 사용되는 방법은, 독점 메시지의 사용과 마찬가지로 제조업체의 선택에 달려 있다. 정상 작동 중에는 독점 메시지 사용을 최소화해야 한다. 여기서 CF 당 Proprietary A, A2, B의 합계는 네트워크 용량의 2 %를 초과해서는 안된다.

#### ■ 요청2 (Request2)

요청2(Request2) PG에는 수신 제어 기능이 전송(Transfer) PGN 51,712를 사용하는지 여부를 지정하는 추가 기능이 있다. 수신 제어 기능이 전송(Transfer) PGN을 사용하도록 지정함으로써, 수신 제어 기능이 보고하는 모든 제어 기능에 대한 데이터 세트를 보고하는 기능을 제공한다. 이것은 PGN 59,904 (전송 PGN용으로 적절하게 지정됨)과 보고되어야 하는 제어 기능들의 데이터 세트에 대해 요청된 동일한 PGN을 수신한 경우에 대한 수신 제어 기능의 일반적인 보고를 포함한다. 예를 들어, 전송 사용 (Use Transfer) PGN 매개변수가 yes이면 응답은 요청된 PGN 관련 모든 알려진 데이터를 포함해야 한다. 전송 사용 (Use Transfer) PGN이 0일 경우, 요청2(Request2) PGN의 효과는 요청(Request) PGN (59,904)을 사용한 것과 같다. 전송 사용(Use Transfer) PGN이 0일 때 요청2(Request2)에 대한 응답은 전송(Transfer) PGN을 사용하여 전송되지 않으며, 요청(Request) PGN(즉, PGN 59,904)을 사용하여 요청했을 때와 정확히 동일하게 전송된다. 아래 정보는 PGN을 요청2(Request2) 매개변수 그룹에 할당한다.

요청2(Request2)와 전송(Transfer) PGN은 주어진 제어 기능이 PGN과 두 개 이상의 제어 기능에 대한 데이터를 보고하는 경우에 필요하다.

#### ■ 전송(Transfer)

전송(Transfer) PGN은 요청2(Request2)에 대한 응답으로 주어진 PGN에 대한 여러 데이터 세트를 보고하는 메커니즘을 제공한다. 주어진 PGN에 대한 이러한 다중 데이터 세트는 각 데이터 세트에 길이가 있고 ISO 11783-5 NAME의 4바이트 레이블이 지정되어야 한다. NAME의 4 바이트는 각 제어 기능을 식별한다. 요청에 응답하는 제어 기능은 이 응답의 첫 번째 데이터 세트와 동일한 정보를 PGN 59,904에서 보고해야 한다. 제어 기능에 단 하나의 데이터 세트만 있을 경우, 전송(Transfer) PGN을 사용하는 하나의 데이터 세트로 응답해야 한다.

요청2(Request2)와 전송(Transfer) PGN은 주어진 제어 기능이 PGN과 둘 이상의 제어 기능에 대한 데이터를 보고하는 임무를 수행하는 경우에 유용하다. 예로서는 구현 식별, 구성 요소 ID 및 소프트웨어 식별과 같은 PGN이 포함된다. 아래 정보는 PGN을 Transfer 매개변수 그룹에 할당한다.

#### 6) 메시지 우선 순위

CAN 데이터프레임의 우선 순위는 ISO 11898-1를 따른다. CAN 식별자 필드 내의 값은 메시지 우선 순위를 결정한다. 낮은 값 (모두 0으로 설정된 29 비트 식별자)이 가장 높은 우선 순위이고, 가장 큰 CAN 식별자 (모두 1로 설정된 29 비트 식별자)가 가장 낮은 우선 순위이다. 할당 방법은 5.9에 제시된 지침에 따라 응용 계층에서 식별된다.

#### 7) 버스 접근

버스가 비어 있으면 모든 컨트롤러가 기존 CAN 데이터프레임 전송을 시작할 수 있다. 두 개 이상의 컨트롤러가 동시에 프레임을 전송하고자 할 때 발생하는 버스 액세스 충돌은, CAN 데이터프레임 식별자를 사용하는 경합-기반 중재를 통해 해결된다. 중재 메커니즘은 정보나 시간이 손실되지 않도록 한다. 우선 순위가 가장 높은 프레임을 가진 송신 컨트롤러가 버스 접근권을 얻는다.

#### 8) 경합-기반 중재

중재 동안 모든 송신 컨트롤러는 전송된 비트의 레벨을 버스에서 관측되는 레벨과 비교한다. 만일 이들 레벨이 동일하다면, 컨트롤러는 계속 전송할 수 있다. 만일 열성 수준이 전송되고 지배적 수준이 관측된다면, 해당 컨트롤러는 중재를 잃고 다른 비트의 전송없이 철회한다. 만일 지배적 수준이 전송되고 열성 수준이 관측된다면, 해당 컨트롤러는 비트 오류를 감지한다.

#### 9) 오류 검출

오류 감지를 위해 다음 방법이 제공된다.

- 송신 컨트롤러는 전송한 비트 레벨을 버스에서 검출된 비트 레벨과 비교한다
- 비트 순환 중복 검사 (CRC)
- 스테프 폭이 5 인 가변 비트 스테핑
- 프레임 형식 확인

## 10) SA와 PGN 할당 절차

### ■ 일반

사용 가능한 프로토콜 데이터 단위으로써 PDU1과 PDU2의 두 가지 형식이 제공된다. 매개변수 그룹은 PDU1나 PDU2 형식을 사용하도록 특별히 지정된다. 형식이 지정되면 해당 매개변수 그룹에 다른 형식을 사용할 수 없다. PDU1 형식은 매개변수 그룹을 특정 대상으로 보낼 필요가 있을 때마다 사용된다. 매개변수 그룹의 할당은 다음 특성을 사용해야 한다.

- 우선 순위
- 업데이트 속도
- 다른 네트워크 제어 기능에 대한 패킷 데이터의 중요성
- 매개변수 그룹과 관련된 데이터의 길이

이 할당 프로세스를 돕기 위해 새로운 SA 또는 PGN 할당을 요청할 때 요청 양식을 사용할 수 있다. 우선 순위 열은 각 PGN에 대한 기본 우선 순위 값을 할당하는 데 사용된다. 우선 순위 필드는, 필요에 따라 OEM (Original Equipment Manufacturer)이 네트워크 튜닝을 할 수 있도록, 각 PGN 값에 대해 프로그래밍할 수 있다. 모든 PGN을 요청할 수 있지만, 이미 주기적으로 브로드캐스트되는 메시지에 대한 요청은 권장하지 않는다.

여러 특정 제어 기능들 중 하나를 직접 제어(명령)하기 위한 매개변수인 경우에만, 메시지에 목적지가 필요한 PGN이 할당된다. 그렇지 않으면, 제어 기능이 메시지 내의 매개변수에 액세스할 수 있도록, 목적지가 없는 PGN이 선택된다.

기본 SA는 메시지 우선 순위, 업데이트 속도 또는 중요도를 고려하지 않고 선형 방식으로 할당된다. PGN은 PGN 및 SA 요청 양식에 제공된 기준에 따라 표 7의 다양한 섹션에 선형으로 할당된다. 만일 반복률이 10Hz 이상인 경우, 다중-패킷 메시지가 허용되지 않는다.

### ■ 주소 할당 조건

ISO 11783에서 할당되지 않은 주소의 수는 제한되어 있기 때문에, 새로운 주소 할당은 효율적으로 이루어져야 한다. 시스템 내에 할당 가능한 최대 주소 수는 256을 초과할 수 없다. 따라서 주소 정의에 대한 추가는 트랙터 또는 기계 내에서 특정 기능을 제공하는 장치로 제한되어야 한다. 특정 기능의 예로는 엔진, 변속기, 브레이크 및 연료 시스템에 대해 현재 정의된 주소가 있다. 이 문서에서 새로운 주소 할당을 위해 제안된 기능은 현재 정의된 주소와 유사한 범위를 가져야 하며, 대부분의 ISO 11783 사용자에게 유용해야 한다. ISO 11783 컨트롤러는 ISO 11783-5를 준수하여 자체-주소-구성을 지원해야 한다.

### ■ 매개변수 그룹 할당 조건

ISO 11783에서 할당되지 않은 가용한 매개변수 그룹의 수는, 임업 또는 농업용으로 제안될 수 있는 수를 고려할 때, 제한적이다. 많은 수의 매개변수 그룹에 대한 필요성은 ISO 11783에 통합된 기능에 의해 완화된다. ISO 11783에는 세 가지 기본 통신 방법이 있기 때문에, 각 방법들을 적절하게 사용한다면 가용한 매개변수 그룹을 효과적으로 사용할 수 있다.

- PDU1 형식 (PS = DA, 목적지 별 통신 허용)
- PDU2 형식 (PS = GE)
- 두 개의 사전 정의된 독점 PGN을 사용하는 독점 통신

이러한 각 방법에는 적절한 용도가 있다. 동일한 메시지가 하나 또는 다른 대상으로 보낼 수 있는 경우 대상별 매개변수 그룹이 필요하다. 엔진에 보내는 토크 제어 메시지는 ISO 11783에 정의되어 있다. 엔진이 두 개 이상인 경우 이 메시지를 원하는 엔진으로만 보내야 하기 때문에, 대상별 매개변수 그룹이 필요하고 할당된다.

PDU2 형식 통신은, 단일 또는 여러 소스에서 단일 대상으로 보낸 메시지와 단일 또는 다중 소스에서 여러 대상으로 보내는 메시지들을 포함하여 여러 상황에 적용된다. 메시지를 둘 다가 아닌 하나 또는 다른 목적지로 전송되어야 하는 경우 PDU2 통신을 사용할 수 없다.

독점 통신은 독점 PGN을 사용하여 제공된다. 비목적지 독점 통신과 목적지 특정 독점 통신에 대해 서로 다른 PGN이 할당된다. 이것은 두 가지 대체 기능을 허용한다.

- 특정 소스는 사용자가 원하는 대로 식별된 PS 필드를 사용하여 PDU2 형식 (비목적지 특정)으로 독점 메시지를 송신

- 진단 도구가 가능한 컨트롤러 그룹 중 특정 대상으로 통신을 지시하는 상황을 허용하는 PDU1 형식(대상별)의 사용

독점 통신은 다음 두 가지 상황에서 유용하다.

- 표준화된 통신이 불필요한 경우
- 독점 정보 전달이 중요할 때

단일 제조업체가 구축한 컨트롤러간의 통신은 대부분 표준화를 필요로 하지 않는다. 전달되는 정보는 일반적으로 네트워크의 다른 컨트롤러에 유용하지 않기 때문에, 이러한 상황에서 독점 매개변수 그룹이 사용될 수 있다.

매개변수 그룹 할당을 고려할 때 독점 및 PDU2 형식 통신 방법이 고려되어야 한다. 만일 독점 정보가 전달되는 경우 또는 전달되는 정보가 일반적인 관심사가 아닌 경우, 독점 방법을 사용해야 한다. 만일 정보가 일반적인 관심사이고 특정 제어 기능에 대한 메시지의 방향을 요구하지 않는 경우, PDU2 형식 할당이 필요하다. 마지막으로, 정보가 일반적인 관심사이지만 하나 또는 다른 제어 기능에 대한 지시가 필요한 경우, 대상별 주소 지정이 필요하며 대상 PDU1 형식 매개변수 그룹 할당을 찾아야 한다.

### ■ 데이터 필드 정의

CAN 기반 시스템으로 메시지 오버 헤드를 최소화하려면 데이터 필드(모두 8 바이트)를 최대한 사용해야 한다. 시간이 매우 중요한 메시지의 경우를 제외하고 관련 매개변수는 8 바이트 데이터 필드를 채울 수 있도록 그룹화되어야 한다. 이 원칙을 따르면 향후 할당을 위해 PGN을 절약할 수 있다. 드물게 사용되는 데이터 필드를 초래하는 매개변수 그룹의 정의를 허용하기 위해선 강력한 정당성이 필요하다.

### 11) 전송 프로토콜 기능

전송 프로토콜 기능은 전송 프로토콜 기능이 메시지의 "패킷화"와 재조립, 그리고 연결 관리라는 두 가지 주요 기능으로 세분된다는 인식과 함께 데이터 링크 계층의 일부로 설명된다. 여기서 발신 제어 기능이라는 용어는 송신 요청 (Request to Send) 메시지를 전송하는 제어 기능에 해당하고, 수신 제어 기능은 송신 가능 (Clear to Send) 메시지를 전송하는 제어 기능에 해당한다.

전송 프로토콜의 기능에는 두 가지 종류가 있는데, 메시지 길이가 8 바이트 이상 1785 바이트 이하인 경우에는 전송 프로토콜 기능을 사용하고, 메시지 길이가 1785 바이트 이상인 경우에는 확장 전송 프로토콜 기능을 필요로 한다. 전송 프로토콜 기능은 SAE와 조화를 이루지만 확장 전송 프로토콜 기능은 그렇지 않다.

### ■ “패킷화”와 재조립

길이가 8바이트보다 큰 메시지는 기존 단일 CAN 데이터프레임에 비해 너무 크기 때문에 여러 개의 작은 패킷으로 분할되고, 이러한 패킷들은 별도의 기존 CAN 데이터 프레임들로 전송된다. 수신 제어 기능에서 개별 메시지 프레임들은 수신된 패킷의 순차 번호 대로 수신되고 재조립된다.

#### a) 메시지 패킷

기존 CAN 데이터프레임은 8 바이트 데이터 필드를 포함한다. 대용량 메시지를 구성하는 개별 패킷들이 정확하게 재조립되기 위해서는 개별적으로 식별 가능해야 하기 때문에, 데이터 필드의 첫 번째 바이트를 패킷의 시퀀스 번호로 정의한다.

개별 메시지 패킷에는 1에서 255까지의 시퀀스 번호가 할당된다. 전송 프로토콜을 사용할 때의 최대 메시지 크기는 255 패킷 × 7 바이트/패킷 = 1,785 바이트이다.

#### b) 시퀀스 번호

시퀀스 번호는 "패킷화" 하는 동안 네트워크에 보내기 위한 메시지들에게 할당되며, 이들 패킷들을 수신하여 원래의 메시지로 재조립할 때 사용된다.

시퀀스 번호는 1부터 시작하여 전체 메시지가 "패킷화"되어 보낼 때까지 개별 패킷에 순차적으로 할당된다. 패킷은 패킷 1부터 오름차순으로 순차적으로 전송된다.

#### c) “패킷화”

다중-패킷 메시지는 데이터가 단일 CAN 데이터프레임의 데이터 필드에 맞지 않는 메시지로 정의된다 (즉, 데이터 필드가 8 바이트보다 큰 메시지).

이 프로토콜은 9 바이트 이상의 문자열이 있는 매개변수 그룹을 커다란 메시지로 간주된다. 첫 번째 데이터 전송 패킷에는 시퀀스 번호 1과 문자열의 처음 7바이트가 포함된다. 두 번째 7바이트는 시퀀스 번호 2와 함께 다른 데이터프레임에 배치되고, 세 번째 바이트는 시퀀스 번호 3과 함께 배치된다. 이러한 방법으로 원본 메시지의 모든 바이트가 ISO 11783를 따르는 기존 CAN 데이터프레임에 실어 전송할 때까지 계속된다.

각 데이터 전송 패킷 (전송 시퀀스의 마지막 패킷 제외)에는 원본 대용량 메시지의 7바이트가 포함된다. 최종 패킷은, 패킷의 시퀀스 번호와 매개변수 그룹과 관련된 최소 1바이트의 데이터가 포함되는, 8바이트의 데이터 필드로 구성된다. 나머지 사용되지 않은 바이트는 FF16로 설정된다.

다중-패킷 브로드캐스트 메시지의 패킷간 시간은 10ms ~ 200ms이다 (5.13.3 참조). 특정 목적지로 향하는 다중-패킷 메시지의 경우, 전송 제어 기능은 최소 시간 요구 사항 없이 200ms의 패킷간 최대 시간(CTS가 둘 이상을 허용하는 경우)을 유지한다. 수신 제어 기능은 데이터를 포함하는 패킷이 모두 동일한 식별자를 가지고 있음을 인식해야 한다.

#### d) 재조합

데이터 패킷은 순차적으로 수신된다. 다중-패킷 메시지의 각 데이터 패킷은, 시퀀스 번호 순서대로 단일 바이트 문자열로 재조립된다. 이 바이트 문자열은 대형 메시지를 담당하는 응용 프로그램 객체로 전달된다.

### ■ 전송 프로토콜-연결 관리

연결 관리는 대상별 전송을 위한 제어 기능 간의 가상 연결 열기, 사용 및 종료와 관련된다. ISO 11783을 따르는 환경에서 가상 연결이란 단일 PGN으로 기술되는 대용량 메시지를 전송하기 위한 두 제어 기능의 임시 연결로 간주된다. 만일 연결이 일-대-다인 경우 흐름 제어나 폐쇄가 제공되지 않는다.

#### a) 다중-패킷 브로드캐스트

다중-패킷 메시지는 대상이 한정되지 않을 수 있다. 즉, 브로드캐스트 메시지일 수 있다. 다중-패킷 메시지를 브로드캐스트하기 위해 제어 기능은 먼저 BAM (Broadcast Announce Message)을 전송한다. 전역 목적지 주소로 전송되는 이 메시지는 큰 메시지를 구성되는데, 이것은 네트워크의 제어 기능에 경고를 위함이다. BAM 메시지에는 브로드캐스트할 대용량 메시지의 PGN, 크기 및 분절된 패킷 수를 포함한다. 브로드캐스트 데이터에 관련이 있는 제어 기능은 메시지를 수신하고 재조립하기 위해 필요한 리소스를 할당해야 한다. 그런 다음 데이터전송 PGN (PGN = 60,160)을 사용하여 관련 데이터를 보낸다.

#### b) 연결 초기화

제어 기능이 DA에 전송 요청(Request to Send) 메시지를 보낼 때 연결이 시작된다. 전송 요청 메시지는 전체 메시지의 크기 (바이트), 전송할 개별 메시지의 수, 하나의 CTS에 대한 응답으로 보낼 수 있는 최대 패킷 수와 전송중인 메시지의 PGN이 포함된다.

전송 요청 메시지를 수신할 경우, 제어 기능은 연결의 수락이나 거부를 선택할 수 있다. 연결 수락을 위해서 수신 제어 기능은 송신 가능(Clear to Send) 메시지를 전송한다. 송신 가능메시지에는 수락 가능한 메시지 패킷 수와 예상되는 첫 번째 패킷의 시퀀스 번호가 포함된다. 수신 제어 기능은 전달을 수락하는 패킷 수를 처리하기에 충분한 자원이 있음을 보장해야 한다. 새로 열린 연결의 경우, 패킷의 시퀀스 번호는 1이다.

연결을 거부하기 위해서 제어 기능은 연결 중단(Connection Abort) 메시지로 응답한다. 리소스, 메모리 부족 등이 원인이 될 수 있지만 연결은 어떤 이유든 거부할 수 있다.

발신 제어 기능이 수신 제어 기능 (즉, CTS 제어 기능)으로부터 해당 CTS를 수신하면, 발신 제어 기능 (즉, RTS 제어 기능)에 대한 연결이 설정된 것으로 간주된다. RTS에 대한 응답으로 CTS 메시지를 성공적으로 보낸 경우 수신 제어 기능에 대한 연결이 설정된 것으로 간주된다. 이러한 정의는 연결을 종료하기 위한 연결 중단(Connection Abort)을 보낼 시기를 결정하는 데 사용된다. 수신 제어 기능은 RTS 메시지를 보고 연결을 설정하지 않기로 결정한 경우 연결 중단(Connection Abort)을 보내야한다. 이는 송신 제어 기능이 시간 초과를 위한 대기 없이도 새 연결을 시도할 수 있게 한다.

#### c) 데이터 전달

연결에 대한 송신 제어기능이 송신 가능(Clear to Send)메시지를 수신한 후 데이터 전송이 시작된다. 데이터 전송이 BAM의 결과로 이루어진 경우는 예외이다. 이 경우 송신 가능(Clear to Send)메시지가 사용되지 않는다. 데이터 전송 패킷(Data Transfer)은 데이터 전송 PGN(5.10.5 참조)을 사용하지만, 이 패킷에 포함된 데이터 바이트는 전송 요청(Request to Send) 또는 브로드캐스트 알림(Broadcast Announce) 메시지의 6~8바이트를 통해 공지된 PGN에 적용된다. 데이터 필드의 첫 번째 바이트에는 패킷의 시퀀스 번호가 포함된다.

목적지-특정 메시지의 경우, 수신 제어 기능은 제어 기능 간의 흐름 제어를 조정하는 역할을 한다. 수신 제어 기능이 연결이 열려 있는 동안 일시적으로 데이터 흐름을 중지하려면, 보낼 패킷 수를 0으로 설정하여 송신 가능(Clear to Send)메시지를 사용해야 한다. 흐름이 몇 초 동안 중지되어야 하는 경우, 수신 제어 기능은 송신 제어 기능 연결이 끊어지지 않도록 Th s (0.5 초) 마다 송신 가능(Clear to Send) 메시지를 반복적으로 전송해야 한다. 나머지 비트 필드는 모두 1로 설정된다 ( "Don't care").

#### d) 연결 종료

오류가 없는 경우 두 가지 연결 종료 사례가 존재한다: 첫 번째는 전역 대상에 대한 것이고, 두 번째는 특정 대상에 관한 것이다. 전역 대상의 경우, 데이터 자체를 수신하는 것 이상으로 연결 종료 작업이 수행되지 않는다. 목적지-특정 전송의 경우와 메시지 스트림의 마지막 패킷을 수신하는 경우에, 수신 제어 기능은 메시지의 발신 제어 기능에게 메시지 종료(End of Message) 확인을 전송한다. 이것은 연결이 수신 제어 기능에 의해 닫힌 것으로 간주된다는 발신 제어 기능에 대한 신호이다. 메시지 종료(End of Message) ACK 종료는, 다른 제어 기능의 추후 사용을 위해, 연결을 해제하는데 필요하다.

전역 대상 메시지의 경우, 수신 제어 기능은 연결 중단(Connection Abort) 메시지를 사용할 수 없다. 목적지-특정 전송의 경우, 발신 또는 수신 제어 기능은 언제든지 연결 중단을 사용하여 연결을 종료할 수 있다. 예를 들어, 수신 제어 기능이 메시지 처리에 사용할 수 있는 리소스가 없다고 결정하면 연결을 중단할 수 있다. 이 때 중단 이유 2와 함께 연결 중단 메시지를 생성한다. 이 메시지를 받게 되면, 이미 전달된 모든 메시지 패킷들이 폐기된다. 두 제어 기능들 중 하나가 실패하면 연결이 종료될 수도 있다.

### ■ 전송 프로토콜 - 연결 관리 메시지(TP.CM)

이 유형의 메시지는 연결의 시작과 종료, 그리고 흐름 제어를 위해 용된다. 전송 프로토콜은 다음과 같은 5개의 전송 프로토콜 연결 관리 메시지를 제공한다. 이들은 연결 모드 송신 요청(Connection Mode Request to Send), 연결 모드 송신 가능 (Connection Mode Clear to Send), 메시지 종료 확인 (End of Message Acknowledgement), 연결 중단(Connection Abort) 및 브로드캐스트 알림 메시지(Broadcast Announce Message) 이다. 이 메시지의 형식은 아래 전송 프로토콜-연결 관리에 대한 매개변수 그룹 정의에 나와 있다.

#### a) 전송 프로토콜 연결 모드 송신 요청 (TP.CM\_RTS)

TP.CM\_RTS 메시지는 네트워크의 다른 제어 기능이 가상 연결을 열기를 원한다는 것을 제어 기능에게 알린다. TP.CM\_RTS는, SA 필드가 발신 제어 기능의 필드로 설정되고, DA가 대형 메시지의 의도된 수신 제어 기능으로 설정되고, 전송되는 PGN에 따라 적절하게 나머지 필드가 설정된 메시지이다.

이 메시지의 바이트 5를 사용하면 발신 제어 기능이 송신 가능(Clear to Send)메시지에 지정된 수신 제어 기능의 패킷 수를 제한할 수 있다(그림 B.4 및 B.5 참조). 수신 제어 기능이 이 제한을 준수한다면, 발신 제어 기능이 어떤 이유로든 수신 제어 기능이 수신하지 못한 패킷을 항상 재전송할 수 있다. 동일한 PGN에 대해 동일한 SA에서 여러 RTS가 수신되는 경우, 가장 최근의 RTS가 실행되고 이전의 RTS는 폐기된다. 이 특정 경우에는 중단된 RTS에 대해 중단 메시지가 전송되지 않는다. TP.CM\_RTS는 발신 제어 기능에 의해서만 전송된다.

#### b) 전송 프로토콜 연결 모드 송신 가능(TP.CM\_CTS)

TP.CM\_CTS 메시지는 전송 요청(Request To Send) 메시지에 응답하는데 사용된다. 이것은 일정량의 대용량 메시지 데이터에 대한 준비가 되었음을 피어 제어 기능에게 알린다. 보내기 위해 지운 대용량 메시지 데이터의 양은 발신 제어 기능 TP.CM\_RTS 메시지의 바이트 4와 바이트 5에 있는 두 값 중 작은 값보다 크지 않아야 한다. 연결이 이미 설정된 후, 여러 CTS가 수신되면 연결이 중단된다. 발신 제어 기능이 연결을 중단하면 표 8의 중단 이유 4와 함께 연결 중단(Connection Abort) 메시지를 전송해야 한다. 수신 제어 기능은 이전 CTS의 마지막 데이터 패킷을 수신하거나 시간 초과될 때까지 다음 CTS를 보내지 않는다. 시간 초과인 경우, 수신 제어 기능은 연결 중단을 보낼지 아니면 CTS를 보낼지 선택할 수 있다. 데이터 전송이 오류와 함께 발생하는 경우 다음과 같은 경우가 있다.

- 누락되거나 오류가 있는 패킷을 감지하고 마지막 패킷을 성공적으로 수신하면, 수신 제어 기능은 누락된 패킷부터 재전송을 요청하는 CTS를 보낸다.
- 마지막 패킷을 포함하여 누락된 패킷은 T1 시간 초과로 이어진다.

CTS가 데이터 패킷의 재전송을 요청하는데 사용되는 경우 2 개 이상의 재전송 요청을 사용하지 않는 것을 권장한다. 만일 이 제한에 도달하면, 중단 이유 5와 함께 연결 중단이 전송된다. 연결이 설정되지 않은 상태에서 CTS를 수신하면 무시한다. CTS는 흐름을 제어할 뿐만 아니라 해당 CTS 패킷 번호 이전의 모든 데이터 패킷에 대한 올바른 수신을 확인한다. 따라서 이전 CTS에 대한 정보가 손상된 경우,

다음 순차 패킷을 보내기 전에 손상된 정보에 대한 CTS를 보내야 한다. 이러한 요구 사항 때문에 대용량 메시지 전송의 발신 제어 기능은, 전송하도록 지워진 마지막 패킷 집합 내에서 패킷 재전송을 보장하는 방법으로, TP.CM\_RTS 메시지의 바이트 5를 사용할 수 있다. TP.CM\_CTS는 수신 제어 기능에 의해서만 전송된다.

c) 전송 프로토콜 메시지 종료 확인 (TP.CM\_EndofMsgACK)

TP.CM\_EndofMsgACK 메시지는 큰 메시지의 수신 제어 기능이 발신 제어 기능으로 보내어, 전체 메시지가 올바르게 수신되고 재조립되었음을 나타낸다. 수신 제어 기능은 TP.CM\_EndOfMsgACK을 즉시 전송하지 않으므로써, 세션의 마지막 데이터를 전송한 후에도 연결을 열린 상태로 유지할 수 있다. 이것은 수신 제어 기능이 필요한 경우 패킷을 재전송하도록 한다. 최종 데이터 전송 이전에 발신 제어 기능에 의해 메시지 끝 승인을 수신하면 발신 제어 기능은 이를 무시한다. 메시지 종료 확인(End of Message Acknowledgement)은 송신 제어 기능에게 대용량 메시지의 전송이 올바르게 수신되고 조립되었음을 알리기 위해 사용된다. TP.CM\_EndOfMsgACK은 수신 제어 기능에 의해서만 전송된다.

d) 전송 프로토콜 연결 중단 (TP.Conn\_Abort)

TP.Conn\_Abort 메시지는 가상 연결에 관련된 제어 기능에서 메시지 전송을 완료하지 않고 연결을 닫거나 연결이 초기화되는 것을 방지하는 위해 사용된다. 연결 모드 송신 요청(Connection Mode Request To Send)메시지를 수신하면, 제어 기능은 이 연결이 필요한 메시지를 처리할 수 있는 충분한 리소스가 있는지를 결정해야 한다. 예를 들어, 제어 기능이 시스템 힙에서 메모리를 가져와야 하고 전체 메시지를 수락할 만큼 충분하지 않을 경우, 또는 제어 기능이 다른 작업에 점유되어 있어 큰 메시지를 처리하기 위한 프로세서 주기를 사용하지 못하는 경우이다. 이러한 경우 연결이 설정되지 않은 경우에도 연결 중단(Connection Abort) 메시지를 보낼 수 있다. 이는 원래 제어 기능이 타임아웃이 발생할 때까지 기다릴 필요 없이 다른 가상 연결을 시도할 수 있도록 하기 위함이다. 어떤 이유로든 발신 또는 수신 제어 기능이 시간 초과를 포함하여 데이터 전송을 완료하기 전에 연결을 종료하기로 결정하면 적절한 연결 중단 이유와 함께 연결 중단 (Connection Abort) 메시지를 보낸다.

발신 제어 기능 (즉, RTS 제어 기능)은 CAN 프로토콜 제어 기능에 따라, 연결 중단(Connection Abort) 메시지 수신 후 즉시 전송을 중지해야 한다. 이것이 가능하지 않은 경우, 데이터 패킷 전송을 중지하는 프로세스는 32 데이터 패킷을 초과하지 않고 50ms를 초과하지 않아야 한다. 연결 중단(Connection Abort) 메시지를 보내거나 받은 후 수신한 모든 관련 데이터 패킷은 무시된다. TP.Conn\_Abort는 발신 제어 기능 또는 수신 제어 기능에 의해 전송된다.

e) 브로드캐스트 알림 메시지 (TP.CM\_BAM)

TP.CM\_BAM은 네트워크의 모든 제어 기능에게 대용량 메시지가 곧 브로드캐스트될 것임을 알리는데 사용된다. 이 것은 매개변수 그룹과 전송할 바이트 수를 정의한다. TP.CM\_BAM 메시지 전송 후, "패킷화된" 브로드캐스트 데이터를 포함하는 데이터 전송(Data Transport) 메시지가 전송된다. TP.CM\_BAM은 발신 제어 기능에 의해서만 전송된다.

■ 전송 프로토콜 - 데이터 전송 (Data Transfer) 메시지 (TP.DT)

TP.DT 메시지는 매개변수 그룹과 관련된 데이터를 전달하는데 사용된다. TP.DT 메시지는 다중-패킷 메시지 전송에 사용되는 개별 패킷이다. 예를 들어, 큰 메시지를 전달하기 위해 5 개의 패킷으로 분할한 경우 5개의 TP.DT 메시지가 존재한다. 이 메시지의 형식은 다음 매개변수 그룹 정의에 나와 있다. TP.DT는 발신 제어 기능에 의해서만 전송된다.

■ 전송 프로토콜 연결 조건

제어 기능이 다른 세션을 처리 할 수 없는 경우, 다른 제어 기능이 시도하는 연결 시작을 거부해야 한다. 동일한 SA에서 기존 세션과 동일한 목적지로의 다른 PGN에 대한 RTS도 거부된다. 두 경우 모두 표 8에서 중단 이유 1과 함께 연결 중단(Connection Abort) 을 전송하여 새로 요청한 세션을 거부해야 한다. 이렇게 하면 연결을 원하는 장치가 시간 초과가 될 때까지 기다릴 필요없이 새 연결로 이동할 수 있다.

a) 제어 기능이 지원해야 하는 연결의 수와 유형

네트워크의 각 제어 기능은 한 번에 지정된 대상과 하나의 목적지별 전송 프로토콜 연결 전송을 시작할 수 있다. TP.DT에는 전송중인 데이터의 PGN이 아닌 SA 및 DA만 포함되기 때문이다. 확장 전송 프로토콜 세션은 전송 프로토콜 연결과 병행할 수 있다. 주어진 시간에 대해 발신 제어 기능은 하나의 다중 패킷 BAM(즉, 전역 대상)만 보낼 수 있다. 이는 TP.DT에 실제 PGN이나 연결 식별자가 포함되어 있지 않기 때문이다. 그러나 수신 제어 기능은 서로 다른 발신 제어 기능(즉, 소스 주소)에서 여러 다중-패킷 메시지를 수신할 수 있음을 인지해야 한다.

한편 제어 기능은 동일한 SA에서 하나의 RTS/CTS 세션과 하나의 BAM 세션을 동시에 지원할 수 있어야 한다. 따라서 수신 제어 기능은 두 전송 프로토콜 메시지의 목적지 주소를 사용하여 적절하게 분리된 상태를 유지해야 한다. 전송 프로토콜 메시지 중 하나는 전역이고 다른 하나는 특정 DA이다. DA는 TP.DT가 실제 PGN이나 연결 식별자를 포함하지 않기 때문에, 둘 사이의 차이점을 구별하는 데 사용되어야 한다. 제어 기능이 여러 개의 동시 전송 프로토콜 세션(RTS/CTS 및/또는 BAM)을 지원할 수 있는지 여부에 관계없이 동일한 SA에서 다른 DA를 갖는 TP.DT/ETP.DT 메시지가 구별되는 것을 보장해야 한다. 수신 제어 기능은 DA와 SA를 사용하여 메시지에 대한 데이터를 올바르게 유지해야 한다. 병렬 TP/ETP 세션의 송신기는 수신기가 일정 기간 동안 하나 혹은 두 개의 세션을 모두 보류할 수 있음을 인지해야 한다 (5.10.3.4.1 참조). 그렇기 때문에 어떤 세션이 가장 먼저 완료될 것인지 예측할 수 없다.

b) 의도된 전송 프로토콜의 사용

전송 프로토콜은 9개 이상의 데이터 바이트를 사용하여 PGN을 전송하도록 설계되었다. 특정 인스턴스에서 전송할 9개 미만의 데이터 바이트를 가질 수 있는 다중-패킷으로 정의된 PGN은 DLC가 8로 설정된 단일 클래식 CAN 데이터 프레임으로 전송된다.

c) 동시 PGN 수신

특정 매개변수 그룹은 8바이트 이하일 때, 비-전송-프로토콜 형식으로 보낼 수 있고, 8 바이트보다 크면 전송 프로토콜 형식으로 보낼 수도 있다. 동일한 PG의 이 두 가지 형태가 동시에 전송될 수 있다.

12) 확장 전송 프로토콜 기능

확장 전송 프로토콜 기능은 전송 프로토콜 기능과 유사한 방식으로 데이터 링크 계층의 일부로 설명된다. 이 하위의 절들은 확장 전송 프로토콜의 고유 특성에 중점을 둔다. 이 하위절들에 기술된 방법은 확장 전송 프로토콜 메시지로 구현된다 (예: TP.CM\_CTS에 대한 참조는 ETP.CM\_CTS 메시지로 구현되는 요구사항을 정의한다).

길이가 1,785 바이트보다 큰 메시지는 너무 크기 때문에 전송 프로토콜 세션에 맞지 않다. 따라서 이러한 패킷은 여러 개의 작은 패킷으로 쪼개지고, 수신 제어 기능은 이러한 패킷을, 시퀀스 번호 순서대로 메시지를 재조립할 수 있도록 하는 프레임 연결 관리와 함께, 별도의 기존 CAN 데이터프레임으로 전송한다.

■ 메시지 패킷

확장 전송 프로토콜을 사용하여 단일 연결로 보낼 수 있는 최대 패킷 수는 확장 데이터 패킷 오프셋 (3 바이트)에 의해 제한된다. 확장 전송 프로토콜의 최대 메시지 크기는  $(224 - 1) \text{ 패킷} \times 7 \text{ 바이트 /패킷} = 117,440,505$  바이트이다.

■ 확장 전송 프로토콜-연결 관리

제어 기능이 1,785 바이트의 전송 프로토콜 길이보다 긴 메시지를 보내고자 할 때, 전송 프로토콜의 확장 기능이 필요하다. 확장 전송 프로토콜은 확장 메시지 연결 요청 및 응답에 5개의 제어 바이트를 사용한다. 단일 연결을 통해 보낼 수 있는 확장 메시지 전송 프로토콜의 최대 바이트 수는 117,440,505 바이트이다.

발신 제어 기능은 확장 메시지 요청(ETP.CM\_RTS)을 수신 제어 기능에게 보낸다. 수신 제어 기능은 ETP.CM\_CTS를 발신 제어 기능이 보낼 수 있는 패킷 수(0에서 255 사이)와 전송 프로토콜 데이터 전달 메시지의 첫 번째 세션에 보낼 첫번째 패킷의 패킷 번호와 함께 발신 제어 기능으로 다시 보낸다. 전송할 패킷 수를 0으로 설정하면 수신 및 발신 제어 기능 간의 연결이 열린 상태로 유지된다.

그런 다음 발신 제어 기능은 확장 데이터 패킷 오프셋 (ETP.CM\_DPO)을 전송한 다음, 필요한 수의 데이터 패킷과 함께 확장 전송 프로토콜 데이터 전송 (Extended Transport Protocol Data Transfer) 메시지를 보낸다. 그런 다음, 다음 확장 데이터 오프셋 메시지와 필요한 수의 확장 전송 프로토콜 데이터 전송 메시지를 반복하기에 앞서, 수신 제어 기능으로부터의 다음 ETP.CM\_CTS 메시지를 기다린다. 실제 확장 메시지의 각 패킷 절차는 확장 전송 프로토콜 데이터 전송 메시지의 시퀀스 번호에 추가된 마지막 수신 확장 데이터 패킷의 오프셋과 동일하다. 또한 확장 전송 프로토콜의 타이밍은 전송 프로토콜과 동일하다.

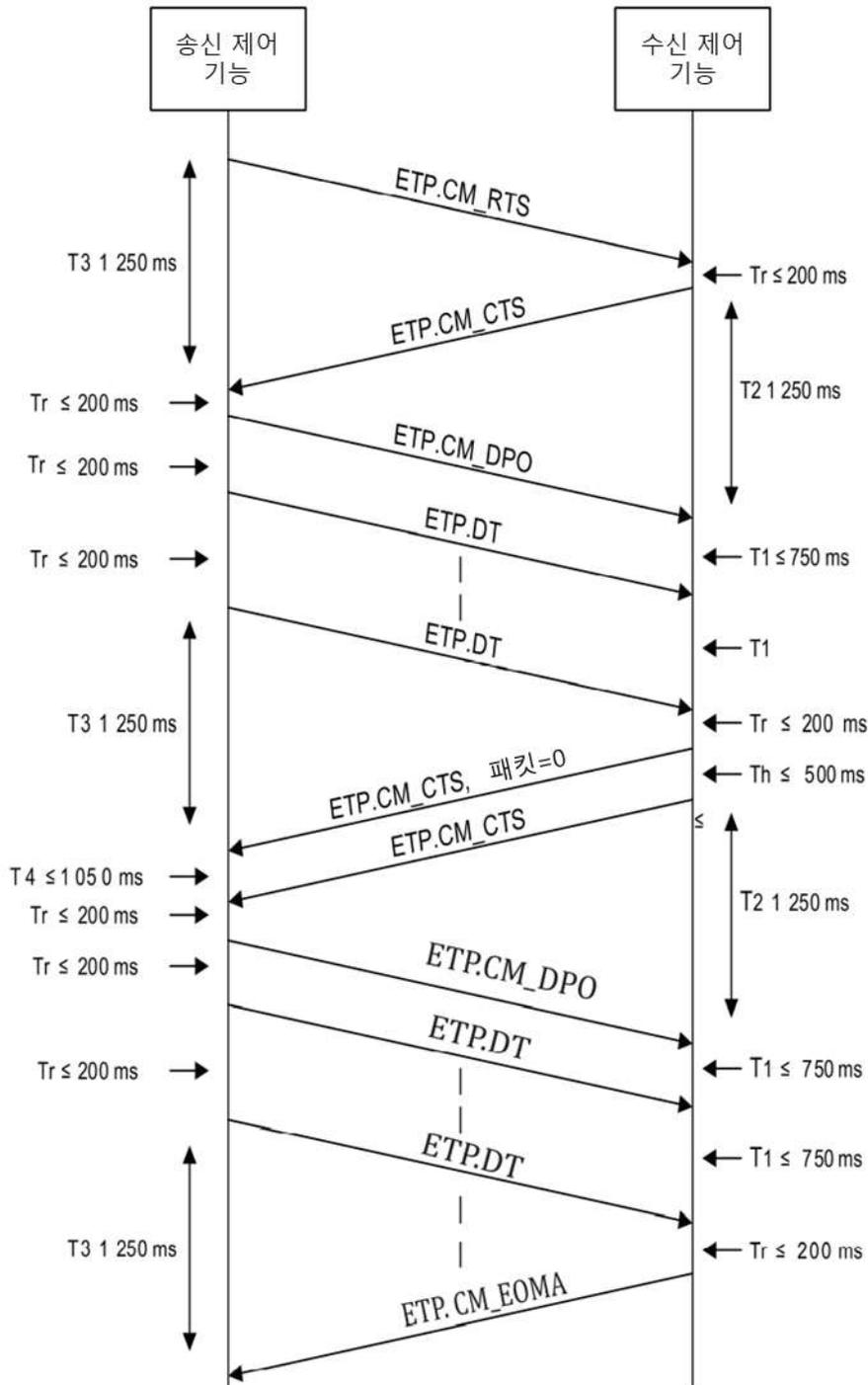


그림 8 - 확장된 전송 메시지 시퀀스

#### ■ 확장 전송 프로토콜 - 연결 관리 메시지 (ETP.CM)

확장 전송 프로토콜 연결 관리 메시지는 전송 프로토콜에 사용되는 것과 다른 PGN을 사용한다.

##### a) 확장 연결 모드 송신 요청 (ETP.CM\_RTS)

ETP.CM\_RTS 메시지는 네트워크의 다른 제어 기능이 확장 전송 프로토콜을 위한 가상 연결을 열려고 함을 제어 기능에게 알려준다. 이 메시지는 발신 제어 기능과 동일한 SA 필드가 있는 메시지로 구성되며, DA는 대용량 확장 메시지의 보내고자 하는 수신 제어 기능의 DA로 설정되고, 나머지 필드는 전송되는 PGN에 적절하게 설정된다.

##### b) 확장 연결 모드 송신 가능(ETP.CM\_CTS)

ETP.CM\_CTS 메시지는 확장 송신 요청(Extended Request to Send)메시지에 대한 응답으로 사용된다.

일정량의 대용량 확장 메시지 데이터에 대한 준비가 되었음을 피어 제어 기능에 알린다. 전송할 패킷의 수를 0으로 설정하여 제어 기능 간의 연결을 유지할 수 있는데, 이 경우, ETP.CM\_CTS "0"은 ETP.CM\_DPO 메시지에 의해 확인되지 않는다.

**c) 연결 모드 패킷 오프셋 (ETP.CM\_DPO)**

ETP.CM\_DPO는 하나의 CTS 데이터 그룹 내에서 전송될 패킷들의 번호가 시작되는 오프셋을 설정한다. 실제 확장 메시지 시퀀스 번호 = (ETP.DT 시퀀스 번호 + ETP.CM\_DPO 데이터 패킷 오프셋).

**d) 확장 메시지 종료 확인 (ETP.CM\_EOMA)**

ETP.CM\_EOMA 메시지는 대용량 메시지의 수신 제어 기능으로부터 발신 제어 기능에게 전달됨으로써, 전체 확장 메시지가 올바르게 수신되었고 재조립되었음을 나타낸다.

**e) 확장 연결 중단 (ETP.Conn\_Abort)**

ETP.Conn\_Abort 메시지는 가상 연결과 관련된 제어 기능들 중 어느 하나가, 메시지 전송을 완료하지 않고 연결을 종료하기 위해 사용한다.

확장 연결 모드 송신 요청(Extended Connection Mode Request to Send)메시지를 수신하면, 제어 기능은 이 연결이 필요로 하는 메시지를 처리하기 위한 가용 리소스가 충분한지를 판단해야 한다. 예를 들어, 제어 기능이 시스템 힙에서 메모리를 가져와야 하고, 전체 메시지를 수락할 만큼 충분하지 않을 경우, 또는 제어 기능이 다른 작업에 점유되어 있어 큰 메시지를 처리하는 프로세서 주기를 사용하지 못하는 경우이다. 이러한 경우 제어 기능은, 송신할 패킷 수를 0으로 응답함으로써 발신 제어 기능과 수신 제어 기능 사이의 연결을 열린 상태로 유지하거나, 확장 연결 중단 (Extended Connection Abort) 메시지를 보내 연결을 닫을 수 있다.

어떠한 이유로든 발신 또는 수신 제어 기능이, 시간 초과를 포함하여, 데이터 전송 완료 이전에 연결을 종료하고자 한다면, 적절한 연결 중단 이유와 함께 확장 전송 중단 (Extended Connection Abort) 메시지를 보낸다. 표 9를 참조한다. 만일 표 9 에 연결 중단 이유가 없다면 코드 254를 사용한다.

**■ 확장 전송 프로토콜 - 데이터 전송 메시지 (ETP.DT)**

ETP.DT 메시지는 매개변수 그룹과 관련된 데이터 통신에 사용된다. ETP.DT 메시지는 다중-패킷 메시지 전송을 위한 개별 패킷이다. 예를 들어, 큰 메시지를 전달하기 위해 300 개의 패킷으로 분할한다면, 300 개의 ETP.DT 메시지가 존재한다. 확장 전송 프로토콜 데이터 전송 메시지는 전송 프로토콜에 사용된 것과 다른 PGN을 사용한다.

**■ 확장 전송 프로토콜 - 연결 제약**

병렬 TP/ETP 세션의 송신기는 수신기가 일정 기간동안 하나 혹은 두 세션 모두를 보류할 수 있다는 것을 인지할 수 있어야 한다. 그렇기 때문에 어떤 세션이 가장 먼저 완료될 것인지 예측할 수 없다.

13) PDU 처리 요구사항

PDU 처리를 위해선 구체적인 절차가 필요하다. 컨트롤러는 데이터 링크가 100% 활용되고 있을 때, 메시지 손실을 방지할 수 있을 만큼 충분히 빠르게 데이터 링크 메시지를 처리할 수 있어야 한다. 이것은 또한 활용률이 낮은 상황에서 연속 메시지가 있을 때, 각 컨트롤러가 연속적 특성으로 인해 메시지를 잃어버리지 않을 만큼 충분히 빠르게 메시지를 처리할 수 있어야 함을 의미한다. 메시지를 충분히 빠르게 처리한다고 해서 메시지가 즉시 생성되어야 하는 것은 아니지만, 새 메시지가 이전 메시지를 초과하지 않아야 한다.

14) 어플리케이션 노트

**■ 높은 데이터 속도**

빠른 속도로 업데이트되어야 하고 지연 시간을 엄격히 준수해야 한다는 요구사항을 갖는 데이터는, 가능할 경우, 하드웨어 기반의 메시지 필터링 사용을 허용해야 한다.

**■ 요청 스케줄링**

요청이 전송되기 전에 요청한 정보가 수신된다면 요청 스케줄링을 취소해야 한다. 즉, 요청 스케줄링의 50ms 이전에 정보를 수신한다면, 요청을 보내지 않아야 한다. 브로드캐스트가 권장되는 매개변수 그룹들을 요청해서는 안되며, 권장 브로드캐스트 시간이 특별한 경우를 초과하는 경우, 예외가 발생할 수 있다.

**■ 컨트롤러 응답 시간 및 타임아웃 기본값**

응답을 해야 하는 경우, 모든 컨트롤러는 0.20초(Tr) 이내에 응답해야 한다. 응답을 기다리는 모든컨트롤러는 최소 1.25초(T3)를 기다린 후 포기 또는 재시도해야 한다. 이러한 시간은, 버스 액세스 또는 브리지를 통한 메시지 전달로 인한 지연으로 인하여 원치 않는 타임아웃이 발생하지 않도록 한다. 필요할 경우 특정 애플리케이션에 다른 시간 값을 사용할 수 있다. 예를 들어, 고속 제어 메시지의 경우 20ms 응답이 예상될 수 있다. 더 빠른 응답을 위해선 버퍼링된 메시지를 재정렬해야 할 수도 있다. 최소 응답 시간에는 제한이 없다.

특정 목적지로 향하는 다중 패킷 메시지의 패킷 간격은 0ms에서 200ms이다. 이는 연속된 메시지가 발생할 수 있고 동일한 식별자가 포함될 수 있음을 의미한다. CTS 메커니즘은 패킷 사이의 주어진 시간 간격을 보장하는데 사용할 수 있다(흐름 제어). 다중 패킷 브로드캐스트 메시지에서 필요로 하는 패킷 간격은 10ms ~ 200ms이다. 10ms의 최소 시간은 수신 제어 기능이 CAN 하드웨어로부터 메시지를 가져올 수 있는 시간을 갖도록 한다. 수신 제어 기능은 750ms의 타임아웃(즉, T1)을 사용해야 하는데, 이것은 200ms의 최대 전송 간격보다 더 큰 타임아웃을 제공한다.

#### ■ 필수 응답

요청자를 포함하여, 요청된 파라미터 그룹을 사용하는 모든 제어 기능의 전역 요청에 대한 응답이 필요하다. 전역 요청에 대한 승인은 허용하지 않는다.

요청(예: "주소 요청")을 위해 전역 DA(255)를 사용하는 제어 기능은, 데이터를 요청한 경우 자체적으로 응답을 보내야 한다. 이것은 모든 제어 기능이 응답할 것으로 예상되기 때문에 요구사항이다. 만일 요청을 한 제어 기능이 응답하지 않으면, 다른 네트워크 제어 기능들은 요청된 정보에 대한 잘못된 결론을 내릴 수 있다.

#### ■ 특정 또는 전역 목적지로의 PGN 전송

대부분의 경우, 주기적으로 브로드캐스트되는 PGN을 전역 대상으로 보내는 것이 바람직하다.

#### ■ CTS의 추천 패킷 수

정상적인 구현 작업 중에는, 전송가능한 CTS 당 패킷 수를 16개 이하로 설정하는 것을 권장한다.

## 나-5) [국가표준] 농림업용 트랙터 및 기계류 - 직렬 제어 및 통신 데이터 네트워크 - 제4부: 네트워크 계층

### 1) 네트워크 상호 연결 장치 (NIU)

네트워크에 여러 세그먼트가 있을 때 NIU는 한 세그먼트에서 다른 세그먼트로 메시지를 보내기 위한 수단을 제공한다. 이 장치는 세그먼트당 하나씩 있는 둘 이상의 포트 간에 개별 메시지 프레임을 전송한다.

#### ■ 메시지-전송 작업

유형에 따라, NIU는 다음 메시지-전송 작업 중 하나 이상을 수행할 수 있다.

- 포워딩 (forwarding) (6.1);
- 필터링 (filtering) (6.2);
- 주소 변환 (address translation) (6.3);
- 재패키징 (repackaging) (6.4).

또한 주어진 애플리케이션에 대한 NIU의 적합성을 결정하기 위한 세 가지 주요 성능 기준이 있다.

- 초당 전달되는 최대 메시지 수: 평균 또는 최대 버스 부하로 인해 이 수를 초과하면, 메시지가 손실될 수 있다.
- 초당 필터링되는 최대 메시지 수: 데이터베이스의 항목 수로 인해 이 수를 초과하면, 메시지가 과도하게 지연될 수 있다.
- 최대 전송 지연: 이것은 메시지가 하나의 CF로부터 전송되어 다른 버스 세그먼트의 다른 CF가 수신하는데 발생하는 최악의 대기 시간을 결정하는데 사용된다.

#### ■ 데이터베이스 관리

NIU는 또한 브리지(bridge)와 데이터베이스 관리(6.6)를 지원하여, 상호 연결 장치 자체 내에서 내부 데이터베이스에 대한 접근과 구성을 가능하게 한다.

예 - 브리지가 두 개의 미디어 세그먼트와 각각의 메시지 트래픽을 분리하지만, 상호 연결 장치로 통신이 가능한 네트워크에서는, 주소 공간과 식별자 측면에서 단일 네트워크로 간주한다.

#### ■ 기타 네트워크 계층 기능

네트워크 상호 연결 장치는, 공급자가 제공하거나 네트워크 구성에 따라, ISO 11783의 이 부분에 정의된 기능 이외의 다른 기능들을 수행할 수 있다. ISO 11783-1은 이러한 다른 기능들의 예를 제공한다.

### 2) 네트워크 계층의 역할

네트워크 계층의 주요 역할은 세그먼트 간의 메시지 전송을 관리하는 것이다. 네트워크 계층은, 필요한 기능에 따라, 다음과 같은 서비스를 제공할 수 있는 다양한 유형의 네트워크 상호 연결 장치를 포함한다.

- 메시지를 전달하는 중계기;
- 메시지를 필터링하고 메시지-필터 데이터베이스를 관리하는 브리지;
- 주소 변환을 사용하여 네트워크 세그먼트가 네트워크의 다른 부분에 단일 CF로 표시되도록 하는 라우터;
- CF가 보다 더 쉽게 송수신 및 해석할 수 있도록, 매개변수를 다른 메시지로 재패키징하는 게이트웨이;
- 트랙터 또는 자주식 작업기의 작업기와 트랙터 버스를 연결하는 특수 네트워크 연결 장치인 트랙터 ECU (ISO 11783-9 참조).

이러한 메시지-전송 기능과 함께 네트워크 계층은 NIU(ISO 11783-1) 내의 데이터베이스에 대한 접근을 제공하고 데이터베이스 구성을 허용한다.

다음 그림은 직렬 제어 및 통신 데이터 NIU를 사용하는 농업 및 임업의 일반적인 네트워크 토폴로지이다. ISO 11783-2에 하나의 작업기당 최대 노드 수가 지정되어 있으며, 세그먼트의 최대 표면상의 CF 수는 ISO 11783-5에 명시된 주소 지침에 따른다.

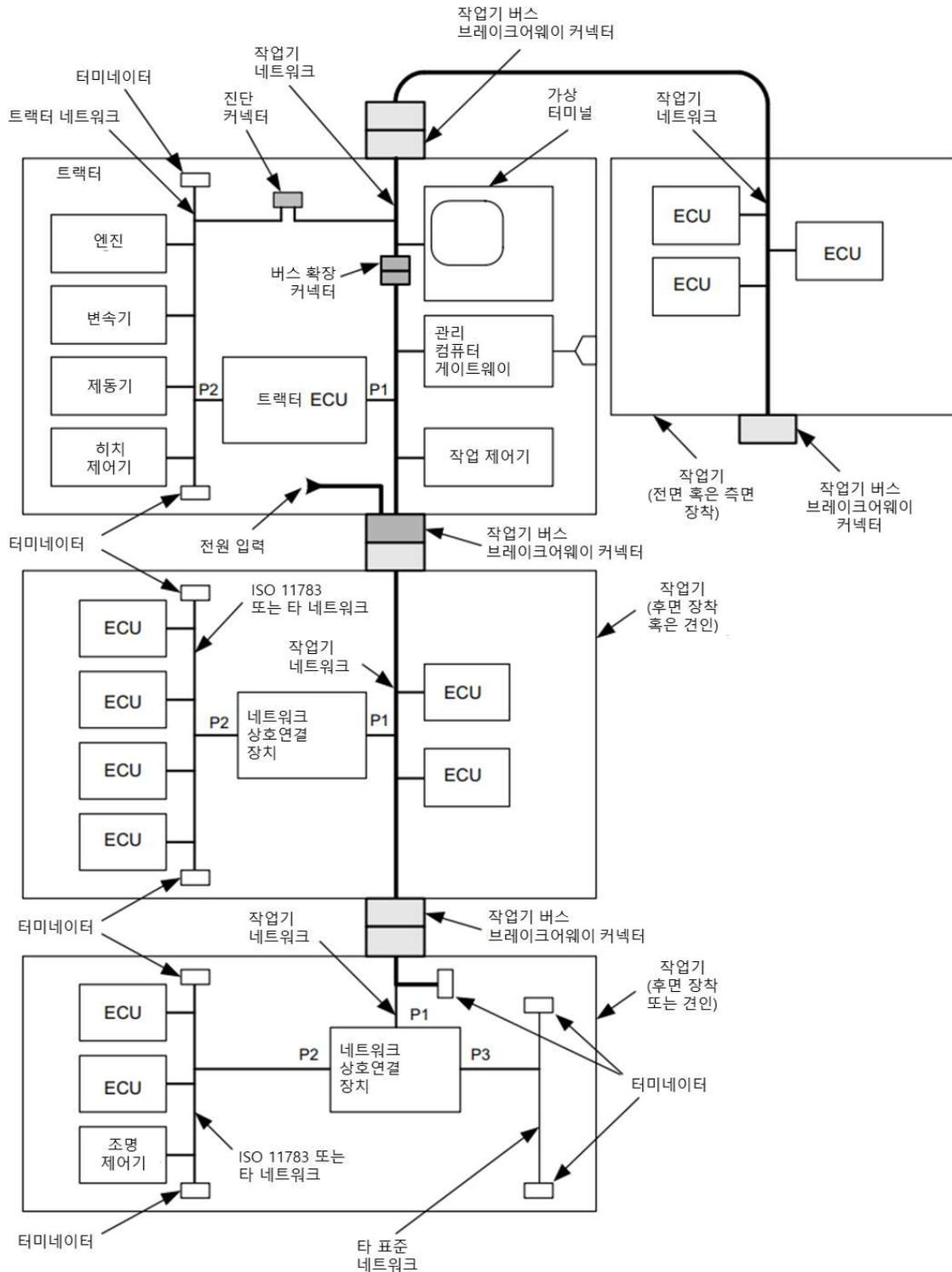


그림 1 - 전형적인 ISO 11783 네트워크

### 3) 네트워크 상호 연결 장치 (NIU)

다음 사항이 일반 요구사항으로 적용된다.

- a) NIU는 필터링과 전달 속도를 보장해야 한다.

- b) 최대 통과-지연 값을 초과해서는 안된다.
- c) 프레임이 한 포트에서 수신되고 다른 포트에 전송되는 순서는, 과도한 지연을 피하기 위하여 주어진 우선 순위를 따른다.
- d) 네트워크 상호 연결 장치는, 우선 순위가 높은 메시지를 우선 전달한 후 우선 순위가 낮은 메시지를 전달해야 한다.
- e) 주어진 우선 순위에 따라 수신한 메시지의 순서대로 동일하게 전달한다.
- f) 간단한 선입-선출 (FIFO) 메시지 대기열은 사용하지 않는다.

다음 사항이 일반 권장 사항으로 적용된다.

- a) NIU는 필터 데이터베이스를 읽고 수정할 수 있는 기능을 제공해야 한다.
- b) NIU는 브리지, 라우터 또는 게이트웨이 관리와 관련된 메시지의 포워딩, 필터링, 주소 변환 및 재패키징 구성에 대한 표준 접근을 제공하여 데이터베이스 관리를 지원해야 한다.
- c) NIU는, 작동 중일 때, 네트워크의 모든 CF에 투명해야 한다.

트랙터의 트랙터 네트워크와 작업기 부분 사이에 특별한 유형의 NIU인 트랙터 ECU가 있어야 하며, 트랙터 네트워크 부분을 격리하고 보호해야 한다. 게이트웨이와 유사하게 트랙터 ECU는 작업기 네트워크의 다른 CF에 트랙터를 나타낸다.

#### ■ 네트워크 토폴로지

시스템 네트워크 토폴로지 (6.7)는 CF 사이에 하나의 경로만 있도록 구성되어야 한다.

#### ■ 네트워크 주소 지정

네트워크의 데이터 링크 계층 (ISO 11783-3)은 256 개의 송신지 주소를 제공한다. 네트워크에서 허용되는 CF의 이론상 주소 수는 null 과 전역 주소를 사용하지 않을 경우 254이다. 각 ECU의 버스에 대한 전기적 부하는, 네트워크에 연결된 노드의 수를 제한할 수 있다(ISO 11783-2).

#### ■ 네트워크 상호 연결 장치 기능

##### a) 포워딩

NIU는 두 개 이상의 포트 (각 네트워크 세그먼트당 하나의 포트)간에 개별 메시지 프레임을 전송한다. 한 포트에서 수신되고 다른 포트에서 전송된 프레임의 순서는 주어진 우선 순위 수준을 따른다. NIU는 더 높은 우선순위의 대기열에 있는 모든 메시지를 더 낮은 우선순위의 메시지보다 먼저 전달해야 한다. 그렇지 않으면 특정 포트에 전달되는 모든 메시지가 지나치게 지연될 수 있으며, 이 요구 사항을 충족하기 위해 단순 FIFO 대기열을 사용해서는 안된다.

NIU가 동일한 주소 공간의 다른 세그먼트로 메시지를 전달할 때 (7.1 리피터 및 7.2 브리지) 메시지 발신지의 주소와 동일한 주소를 사용한다. 일반적으로 장치는 원래 메시지를 보낸 세그먼트로 메시지를 재전송하지 않으며, 주소 또한 지정된 ISO 11783 네트워크에서 고유하므로 일반적으로 중재 문제를 일으키지 않는다.

유일한 예외는, 다른 CF가 동일한 주소를 동시에 요청하는 경우, 주소 요청 메시지가 세그먼트로 전달되는 경우이다. 이러한 낮은 가능성의 상황에서 NIU는 CAN 프로토콜 칩 내에서 자동 재전송 절차를 중지해야 한다. 그렇지 않으면 NIU가 여러 번의 충돌을 감지하여 "버스 오프(bus off)" 상태가 되고, NIU가 버스 오프 상태를 복구할 수 있을 때까지 다른 메시지가 전달되는 것을 막는다.

NIU가 단순 리피터나 브리지 역할을 하는 경우, NIU는 주소를 요청하기 전에(즉, 주소 변환을 수행하지 않음) 한 세그먼트에서 다른 세그먼트로 메시지 전달을 시작할 수 있다.

##### b) 필터링

필터링 기능의 경우, 전송 프로토콜, 확장 전송 프로토콜, 고속 패킷 또는 기타 메시지 패킷 메커니즘으로 전송된 메시지는 포함된 메시지의 매개변수 그룹 번호(PGN)에 따라 처리된다. 포함된 메시지의 PGN이 필터에 대해 정의된 경우, 프로토콜 처리 메시지는 정의된 필터에 따라 처리되어야 한다.

##### - 블록 모드 (Block mode)

블록 필터 모드 (0)에서, NIU는 기본적으로 모든 메시지를 전달한다(7.2). 버스 사용률 (트래픽)은 각 버스 세그먼트에서 더 높을 수 있지만, 허용 가능한 한계 내에 있으면 메시지 필터링 알고리즘이 존재하지 않는다. NIU 내의 필터 데이터베이스는 전달(차단)되지 않아야 하는 메시지에 대한 식별자 항목 (PGN 값)을 포함 할 수 있다. 이것은 주어진 세그먼트에서 전체 버스 트래픽을 줄이는데 사용할 수 있으며, ISO 11783을 준수하는 브리지의 기본 작동 모드이다. 필터 데이터베이스 항목은 일반적으로 조립 또는 초기

구성 중에 만들어지며 비휘발성 메모리에 유지된다.

**- 패스 모드 (Pass mode)**

패스 필터 모드 (1)는 기본적으로 메시지가 전달되지 않도록 NIU를 설정한다(7.2). 그런 다음, 메시지를 전달하려면 해당 메시지에 대한 특정 식별자 (PGN 값)가 있는 항목이 존재해야 한다. 이 모드는 하부 네트워크에 연결되어 특정한 기능을 수행하는 NIU의 포트에 가장 적합하다. 이것은 CF와 전체 네트워크의 기능에 대한 사전 지식이 필요하거나, 또는 CF가 필터 데이터베이스에 항목을 추가할 수 있어야 한다. 이 경우, NIU가 대규모 필터 데이터베이스를 수용하기 위해선 더 많은 메모리와 처리 능력이 필요할 수 있다. 또한 데이터베이스 내의 일부 항목은 영구적이어야 하므로(즉, 항상 존재하도록 구성되어야 함) 해당 메시지가 항상 전체 네트워크를 통해 전달된다. 전형적인 적용 분야는 네트워크 관리, 진단 및 전역 요청의 경우이다.

**c) 주소 변환**

NIU는 특정 메시지에 대한 주소 변환을 제공하여(7.3), 세그먼트의 특정 기능의 주소 (예, 조명)를 알지 못하더라도, 단일 주소를 사용하여 특정 세그먼트(예, 작업기)를 지칭할 수 있다. 이를 위해, “조회” 테이블을 통해 해당 송신지 주소 또는 목적지 주소를 식별하기 위한 주소 변환 데이터베이스가 있어야 한다. NIU는 이 서비스를 제공하기 전에 유효한 요청 주소가 있어야 한다.

**d) 메시지 재패키징**

NIU는 메시지를 한 세그먼트에서 다른 세그먼트로 전송할 때, 메시지를 다시 패키징할 수 있다(7.4). 이것은, 메시지당 유용한 매개변수의 양을 향상시키고 특정 CF가 수신하는 다른 메시지의 수를 줄임으로써, 잠재적으로 버스 트래픽을 감소시킨다. 메시지를 재패키징하는 방법을 정하려면, 메시지 재패키징 데이터베이스 또는 처리 루틴이 있어야 한다.

**■ 네트워크 메시지**

**a) 네트워크 메시지 PGN**

네트워크 메시지에 대한 PGN은 다음 표에 나와 있다.

표 1 - 네트워크 메시지

매개변수 그룹 이름	네트워크
정의	NIU 매개변수와 데이터베이스에 액세스하는데 사용.
전송 반복률	사용자 요구 사항에 따라 초당 5회를 초과할 수 없음
데이터 길이	가변
확장 데이터 페이지	0
데이터 페이지	0
PDU 형식	237
PDU-특정 필드	목적지 주소
기본 우선순위	6
PGN	60,672 (00ED0016)

네트워크 메시지는 다음을 위한 수단을 제공한다.

- 데이터베이스 접근 및 구성
- 포트 주소 접근
- 네트워크 상호 연결 장치 내의 상태 및 통계로의 접근
- 네트워크 세그먼트 간 연결의 열고 닫기

특정 대상에 대한 요청이나 명령이 있을 때(즉, 전역적이지 않음), 특정 기능 코드가 지원되지 않거나 수행할 수 없다는 확인 응답만 있더라도 응답이 필요하다. CF는 요청이나 명령을 보낸 후 또 다른 요청이나 명령을 보내기 앞서, 응답 또는 "응답 없음" 시간 초과를 기다려야 한다(ISO 11783-3 참조). 다중-패킷 PGN의 경우, 단일 요청의 결과로 여러 CAN 데이터 프레임이 발생할 수 있다.

8 바이트 미만의 가변 길이 메시지의 경우 사용되지 않는 바이트를 FF16으로 채운다. 만일 8 바이트 이상이 필요한 경우 전송 프로토콜을 사용해야 한다 (ISO 11783-3 참조).

b) 메시지 기능

네트워크 메시지 데이터 필드의 첫 번째 바이트는 메시지 수신자가 수행해야 하는 기능을 식별하는 데 사용한다. 기능 및 해당 코드가 표 2에 나열되어 있다.

표 2 - 메시지 기능 요약

기능 코드	정의	방향
010	필터 데이터베이스 사본 요청	CF 에서 NIU
110	필터 데이터베이스 사본 요청에 대한 응답	NIU 에서 CF
210	필터 데이터베이스에 항목 추가	CF 에서 NIU
310	필터 데이터베이스에서 항목 삭제	CF 에서 NIU
410	필터 데이터베이스에서 항목 지우기	CF 에서 NIU
510	폐지, 사용하지 않음	해당 없음
610	필터 데이터베이스 항목 만들기	CF 에서 NIU
710	NAME 정규화된 필터 데이터베이스 항목 추가 요청	CF 에서 NIU
810 - 6310	예약	해당 없음
6410	송신지 주소 목록 요청	CF 에서 NIU
6510	송신지 주소 목록 요청에 대한 응답	NIU 에서 CF
6610	송신지 주소 및 NAME 목록 요청	CF 에서 NIU
6710	송신지 주소 목록 및 NAME 요청에 대한 응답	NIU 에서 CF
6810 - 12710	예약	해당 없음
12810	NIU 일반 매개변수 요청	CF 에서 NIU
12910	NIU 일반 매개변수 요청에 대한 응답	NIU 에서 CF
13010	일반 통계 매개변수 재설정 명령	CF 에서 NIU
13110	NIU-특정 매개변수 요청	CF 에서 NIU
13210	NIU-특정 매개변수 요청에 대한 응답	NIU 에서 CF
13310	특정 통계 매개변수를 재설정하는 명령	CF 에서 NIU
13410 - 19110	예약	해당 없음
19210	연결 열기 요청	CF 에서 NIU
19310	연결 열기 요청에 대한 응답	NIU 에서 CF
19410	연결 종료 요청	CF 에서 NIU
19510	연결 종료 요청에 대한 응답	NIU 에서 CF
19610 - 25510	예약	

c) 포트 번호

포트 번호는 번호가 매겨진 각 포트에 대한 니블(nibble)로 표시된다. 포트 번호 0 (로컬)은, 연결 포트 번호를 알 수 없을 때, CF가 NIU와 메시지를 주고받을 수 있도록 하는데 사용된다. 메시지는 메시지를 수신하는 "로컬" 포트에 보내진다.

포트 번호 15 (전역)는 CF가, NIU의 포트 수를 알 필요없이, NIU로 메시지를 보내는 것을 용이하게 하게 위해 사용된다. 메시지 기능에 필요한 경우, 네트워크 메시지의 두 번째 바이트는 포트 간 메시지 방향을 식별하는데 사용된다. 바이트의 하위 니블 (비트 3-0)은 "To"포트를 식별하고 상위 니블 (비트 7-4)은 "From"포트를 식별한다. 포트-쌍 (from/to) 내의 포트 번호 중 하나가 전역으로 설정되면, NIU로부터 각 포트-쌍에 대한 응답이 여러 번 보내질 수 있다.

d) 데이터베이스 관리

장치의 매개변수(상태 및 통계)와 네트워크 토폴로지를 포함하여, 네트워크 상호 연결 장치 내의 다양한 데이터베이스에 접근하고 구성하기 위한 표준 방법이 마련되어야 한다. 관련된 모든 기능은 전력 손실의 경우에도 데이터 값이 유지되도록 비휘발성 메모리를 사용해야 한다. 이는 정적 필터 데이터베이스를 유지해야 하는 경우 특히 중요하다.

e) 네트워크 상호 연결 장치 메시지 필터 데이터베이스의 구성

- 메시지 필터 데이터베이스 구성 방법

다음과 같은 방법으로 메시지 필터 데이터베이스를 구성할 수 있다.

- a) 공급자는 NIU에 고정 필터 데이터베이스를 제공한다. 브리지 설계를 통해 OEM은 제조시 필터 데이터베이스를 미리 구성할 수 있다. 이것은 존재하는 CF와 메시지를 포함하여 전체 네트워크에

대한 사전 지식을 필요로 하며, 시간이 경과됨에 따라 서비스 중인 NIU의 재구성 없이 네트워크에 추가나 변경이 불가하다.

- b) 브리지는 서비스 절차의 일부로, 진단 도구를 사용하여 네트워크를 통해 구성된다.
- c) NIU는 네트워크의 모든 CF에 의해 언제든지 구성된다. 데이터베이스 수정을 위한 별도의 보안 절차가 필요할 수 있으며, 재구성을 위한 접근 제한은 애플리케이션에 따라 다르다.

N.MFDB\_Create\_Entry 데이터베이스 관리 기능을 사용하여 생성된 항목에 대한 각 필터 데이터베이스 항목에는 NAME이 추가되어야 한다(표 5 참조). NAME은 항목을 배치한 CF를 의미하며 동일한 NAME으로만 항목을 제거할 수 있다. 이것은 CF가 충돌되는 요청을 입력하는 것을 방지하지는 않지만, 예기치 않은 항목 삭제를 방지한다. 그럼에도 불구하고 진단 도구를 사용할 때, 이 요구사항을 무시할 수 있는 조항이 있어야 한다.

각 필터 데이터베이스 항목은 필터링을 위해 PGN을 식별하고, 이를 통과 혹은 차단 (6.2.2 및 6.2.3)으로 표시한다. 또한 식별해야 할 포트-쌍 (방향)은 특정 하위 네트워크로의 트래픽을 제한하는 동시에 전달되는 특정 메시지를 허용하기 위해 필요하다.

포트 번호 쌍 (from/to) 중 하나가 15 (전역)로 설정되면, 모든 포트-쌍에 대한 NIU의 다중 응답 때문에, NIU의 포트 번호를 알 수 없는 경우라도 메시지를 보낼 수 있다. NIU는 로컬(0) 또는 전역(15) 포트 식별 방법을 사용하여 필터 데이터베이스를 구성 할 수 있어야 한다.

#### - 필터 모드

필터 데이터베이스에 나열된 PGN의 필터 모드는 표 4에 정의된 필터 모드 바이트 값으로 식별된다.

#### - 메시지 필터 데이터베이스 구성을 위한 메시지

##### • 필터 데이터베이스 사본 요청 (N.MFDB\_Request)

N.MFDB\_Request를 사용하면 CF가 NIU에 의해 전달될 메시지 필터 데이터베이스 사본을 요청할 수 있다.

##### • 필터 데이터베이스 사본 요청에 대한 응답 (N.MFDB\_Response)

필터 데이터베이스 사본 요청에 대한 NIU 응답 메시지는 필터링된 PGN와 해당 필터 모드 항목을 포함한다.

##### • 필터 데이터베이스 항목 추가 명령 (N.MFDB\_Add)

NIU 필터 데이터베이스에 하나 이상의 항목을 추가하기 위한 메시지이다. "To" 포트가 전역으로 설정되어 있으면, 필터 데이터베이스에 적절한 항목을 만들어 각 포트-쌍에서 원하는 작업을 수행해야 한다. 필터 모드는 이 명령에 포함되어 있지 않으며 또한 해당 포트-쌍에 대한 데이터베이스는 삭제와 재생성 없이 변경이 불가하기 때문에, 이 메시지를 사용하는 모든 CF는 항목을 만들기 전에 관련 필터 데이터베이스의 필터 모드를 사전에 알고 있어야 한다. 명령 승인은 승인 메시지(PGN 59392)와 함께 제공된다.

##### • 필터 데이터베이스 항목 삭제 명령 (N.MFDB\_Delete)

NIU 필터 데이터베이스에서 하나 이상의 항목을 삭제하기 위한 명령 메시지이다. 명령 승인은 승인 메시지 (PGN 59392)와 함께 제공된다.

##### • 필터 데이터베이스 지우기 명령 (N.MFDB\_Clear)

이것은 지정된 포트-쌍 및 방향에 대해 하나 이상의 필터 데이터베이스를 지우는 명령 메시지이다. 명령 승인은 승인 메시지 (PGN 59392)와 함께 제공된다.

##### • 필터 데이터베이스 항목 생성 명령 (N.MFDB\_Create\_Entry)

이것은 필터 데이터베이스에 하나 이상의 항목을 생성하는 명령 메시지이다. "To" 포트가 전역으로 설정된 경우, "From" 포트를 포함하는 각 포트-쌍마다 필터 데이터베이스에 여러 항목이 만들어질 수 있다. 이 명령에 포함된 필터 모드는 새로운 항목이 차단을 위한 것인지 통과를 위한 것인지를 나타낸다. 메시지에 포함된 필터 모드는 포트-쌍에 대한 기존 필터 모드를 변경할 수 없다. N.MFDB\_Create\_Entry 명령이 지워지지 않은 데이터베이스에 요청된 경우, 이 명령의 요청자에게 오류를 반환하는 것이 권고된다. N.MFDB\_Add 명령 사용으로 충분하며(Set\_Mode 명령의 폐기와 함께) 단일 데이터베이스 레코드에서 다른 모드로 항목을 생성하도록 요청할 가능성을 제거해야 한다. 명령 승인은 승인 메시지(PGN 59392)와 함께 제공된다.

##### • NAME Qualified 필터 데이터베이스 항목 추가 요청 명령 (N.MFDBNQ\_Add)

이 명령은 메시지의 PGN과 소스의 NAME을 기반으로 메시지를 필터링하는 데 사용된다.

공통 주소 공간 또는 별도의 주소 공간에 적용 할 수 있다. 메시지가 주소 공간이 다른 세그먼트로부터 전달될 때, 개방 연결이 설정되지 않은 경우 (6.9 참조) NIU는 자체 주소를 송신지

주소로 사용하여 메시지를 전달해야 한다. 전송 CF와 개방 연결 (6.9.5.1)이 설정된 경우, 가상 CF의 SA가 전달된 메시지에 사용된다.

전역 주소로 지정된 PDU2 메시지와 PDU1 메시지는 브리지를 통해 필터링될 수 있다. 이것은 CF 또는 서비스 도구는 PGN과 소스 ID간 일치를 기반으로 메시지를 전달하도록 NIU에 요청한다. 소스의 64-비트 NAME과 64-비트 검증 마스크를 지정하여 요청자가 ID를 설정한다. 그런 다음 NIU는 이 ID를 송신지 주소/NAME 테이블의 송신지 주소와 연결한다.

전달된 메시지에는 선택적으로 "데이터 속도 감소"가 적용될 수 있다. 이것은 필터를 설정하는 CF에게 원래 네트워크의 업데이트 속도에 관계없이 메시지에 대한 최대 업데이트 속도를 지정하는 기능을 제공한다. 전달된 메시지의 경우 가장 최근 데이터가 사용된다.

이 명령 메시지는 NAME 필드와 PGN에 의해 규정된 필터 데이터베이스 항목을 생성한다. 최대 전송 속도는 선택적으로 지정할 수 있다.

추가 바이트 t, p, d 및 n는 다중 항목에 대해 필요한 만큼 반복된다.

예제 표의 메시지는 트랙터 ECU (SA = 240) 내의 필터 데이터베이스 항목을 얻기 위해 전송된다. 특정-목적지 요청은 오프-보드(off-board) 진단 도구 (SA = 249)에서 시작되는데, 이 요청은 작업기로(포트 1에서 포트 2로) 갈 때 필터링되는 PGN의 목록만을 위한 것이다. 특정-목적지 응답은 차단된 유일한 메시지 (필터 모드 = 0)가 엔진 구성 (00FEE316)임을 나타낸다.

표 3 -메시지 필터 데이터베이스 접근 예제

기능	식별자						데이터				
	PRI	EDP	DP	PF	DA	SA	제어 코드	포트 쌍	필터 모드	PGN	예약 바이트
N.MFDB_Request	110	0	0	237	240	249	010	1216	FF16	FFFFFF16	FFFF16
N.MFDB_Response	110	0	0	237	249	240	110	1216	0016	00FEE316	FFFF16

■ 네트워크 토폴로지 정보

ISO 11783의 이 부분을 따르는 모든 네트워크 상호 연결 장치는 네트워크의 다른 CF에 투명해야 한다. 데이터베이스를 적절하게 설정하려면 네트워크 토폴로지에 대한 지식이 필요할 수 있다. 6.7.2에 정의된 메시지는 이 누락된 정보를 얻을 수 있는 기능을 제공한다.

데이터 바이트의 하위 니블(nibble)에 포함된 포트 번호는 NIU 포트와 관련된 주소를 식별하는 데 사용된다. 네트워크에 두 개 이상의 NIU가 있는 경우, 송신지 주소가 있는 포트만 식별할 수 있다. 특정 송신지 주소는 원격 버스 세그먼트에 위치할 수 있으며, 각 NIU의 응답은 주어진 송신지 주소를 포함하는 로컬 버스 세그먼트를 결정하기 위해 비교되어야 한다. 장치는 우선 주소-요청을 수행한 다음, 각 포트와 관련된 송신지 주소 목록을 구성해야 한다.

■ 네트워크 토폴로지 메시지

a) 송신지 주소 목록 요청 (N.NT\_Request)

이 메시지는 새로운 설계에는 권장되지 않는다. CF는 N.NT\_Request를 사용하여 NIU 포트의 송신지 주소 목록을 요청할 수 있다. 이 네트워크 토폴로지 메시지 요청은, 포트-쌍이 동일한 주소 공간에 존재할 때, 요청자가 NIU와 관련된 포트의 물리적 레이아웃 또는 토폴로지를 결정하고자 할 때 사용할 수 있다. 브리지의 경우 모든 주소 요청이 NIU의 양쪽에 나타나므로, 요청자는 이 메시지 요청을 사용하여 네트워크의 어떤 세그먼트에 어떤 포트가 있는지 확인할 수 있다.

b) 송신지 주소 목록 요청에 대한 응답 (N.NT\_Response)

이 메시지는 새로운 설계에는 권장되지 않는다. 송신지 주소 목록 요청에 대한 NIU 응답 메시지는 송신지 주소 항목을 포함한다. 이 네트워크 토폴로지 메시지 응답은, 포트-쌍이 동일한 주소 공간에 존재하는 경우, 요청자가 NIU에 대한 포트의 물리적 레이아웃 또는 토폴로지를 결정하려고 할 때 사용될 수 있다. 브리지의 경우 모든 주소 요청이 NIU의 양쪽에 나타나므로, NIU는 이 메시지 응답을 사용하여 어떤 포트가 네트워크의 어떤 세그먼트에 있는지 보고할 수 있다.

c) 송신지 주소 및 NAME 목록 요청 (N.NTX\_Request)

송신지 주소 및 NAME 목록 요청 메시지를 통해 CF는 NIU 포트에서 송신지 주소 및 관련 NAME 목록을 요청할 수 있다. 이 네트워크 토폴로지 메시지는, 요청중인 네트워크 세그먼트 토폴로지가 요청자와 동일한 주소 공간의 일부가 아닐 때 사용된다. 즉, NIU는 이 포트-쌍에게 라우터 역할을 한다. 따라서 요청자는 목적지 네트워크 세그먼트의 주소 요구사항을 볼 수 없다. 전역 포트 번호에 대한

요청이 허용되지 않는다.

**d) 송신지 주소 및 NAME 목록 요청에 대한 응답 (N.NTX\_Response)**

송신지 주소와 NAME 목록 요청에 대한 NIU 응답 메시지에는, NIU가 요청한 주소와 NAME을 포함하여 정의된 포트와 연관된 네트워크 세그먼트의 송신지 주소와 NAME의 모든 항목이 포함된다.

**■ 네트워크 상호연결 장치 매개변수**

NIU 매개변수 (상태 및 통계)에 접근하기 위한 두 가지 메시지 기능 집합이 있다. 이러한 매개변수는 표 6에 정의되어 있다. NIU에 적용할 수 있는 일반 매개변수는 일반 매개변수 메시지로 접근되고, 주어진 NIU 포트-쌍에 적용할 수 있는 매개변수는 특정 매개변수 메시지로 접근된다. 유효한 값의 범위는 ISO 11783-7에 따르며, "사용할 수 없음"을 나타내기 위한 모든 비트 값은 "1"로 설정되고 각 매개변수 번호의 길이는 1 바이트이다.

응답 목록 내의 매개변수는 요청 목록의 매개변수 식별자와 동일한 순서이다 (모든 매개변수와 매개변수 식별자의 길이가 고정되어 있으므로 구분 기호가 필요하지 않음). 주어진 주소로 다른 요청을 보내기 전에 요청 메시지에 대한 응답을 받아야 하지만, 이 요청은 여러 매개변수에 대한 것일 수 있다. 항상 매개변수 0에 대해 요청하는 것이 권장된다. 그러면 전체 목록이 숫자 순서로 반환된다.

표 4 - NIU 매개변수 (상태 및 통계)

초기화 가능	매개변수 식별자	바이트	정의
	0	N/A	모든 매개변수를 숫자 순서로 요청하는데 사용(응답 메시지에 사용되지 않음)
-	1	2	버퍼 크기(바이트)
-	2	2	최대 필터 데이터베이스 크기(바이트)
-	3	2	필터 데이터베이스 항목 수
-	4	2	초당 수신되는 최대 메시지 수
-	5	2	초당 전달되는 최대 메시지 수
-	6	2	초당 필터링된 최대 메시지 수
-	7	2	최대 전송지연 시간(ms)
Yes	8	2	평균 전송지연 시간(ms)
Yes	9	2	버퍼 오버플로로 인해 손실된 메시지 수
Yes	10	2	초과 전송 지연 시간이 있는 메시지 수
Yes	11	2	초당 수신된 평균 메시지 수
Yes	12	2	초당 전달된 평균 메시지 수
Yes	13	2	초당 필터링된 평균 메시지 수
-	14 <sup>a</sup>	4	마지막 전원 켜기 재설정 이후 가동 시간(s)
-	15	1	포트 수
-	16	1	네트워크 상호 연결 장치 유형
	17-255	N/A	ISO의 향후 할당을 위한 예약

a 매개변수 식별자 "14"는 ISO 11783-4의 이전 버전에서 변경되었으며 이 매개변수는 이제 SAE J1939-31[4]를 준용한다.

**a) 상태 및 통계 매개변수 정의**

상태 및 통계 매개변수 정의는 데이터가 N.GP\_Response 또는 N.SP\_Response와 같은 NIU 매개변수 응답으로 보고될 때 사용된다. 데이터의 길이가 1 바이트보다 큰 매개변수는 ISO 11783-7에 설명된 ISO 11783 바이트 순서 규칙을 준수해야 한다. 매개변수 정의에서 달리 명시적으로 언급되지 않는 한, 매개변수 범위에 대한 정의는 ISO 11783-7에 설명된 매개변수 범위 규칙을 따라야 한다. 이러한 매개변수 중 일부는 주어진 NIU, 특정 포트-쌍, 또는 둘 모두에 적용할 수 있다.

- **버퍼 크기 (매개변수 1)**  
이 매개변수는 NIU 버퍼 크기 (바이트)를 보고한다.
- **최대 필터 데이터베이스 크기 (매개변수 2)**  
이 매개변수는 필터 데이터베이스에 사용 가능한 최대 크기를 보고한다.
- **필터 데이터베이스 항목 수 (매개변수 3)**  
이 매개변수는 NIU 필터 데이터베이스의 항목 수를 보고한다.
- **초당 수신되는 최대 메시지 (매개변수 4)**  
이 매개변수는 초당 NIU가 수신 가능한 최대 메시지 수를 보고한다.

- **초당 전달되는 최대 메시지 (매개변수 5)**  
이 매개변수는 초당 NIU가 포워딩할 수 있는 최대 메시지 수를 보고한다.
- **초당 필터링되는 최대 메시지 (매개변수 6)**  
이 매개변수는 초당 NIU가 필터링 할 수 있는 최대 메시지 수를 보고한다.
- **최대 이동 지연 시간 (매개변수 7)**  
이 매개변수는 메시지가 NIU를 통해 이동될 때 걸리는 최대 전송 지연 시간에 관한 성능 기능을 보고한다.
- **평균 이동 지연 시간 (매개변수 8)**  
이 매개변수는 해당 데이터가 마지막으로 재설정된 이후, 메시지가 NIU를 통해 이동될 때의 실제 전송 지연 시간 평균을 보고한다.
- **버퍼 오버플로로 인한 손실 메시지 수 (매개변수 9)**  
이 매개변수는 해당 데이터가 마지막으로 재설정된 이후, 버퍼 오버 플로우로 인해 NIU에서 손실된 메시지의 수를 보고한다.
- **초과 전송 지연 시간을 갖는 메시지 수 (매개변수 10)**  
이 매개변수는 해당 데이터가 마지막으로 재설정된 이후, 전송 지연이 초과된 메시지 수를 보고한다.
- **초당 수신된 평균 메시지 (매개변수 11)**  
이 매개변수는 해당 데이터가 마지막으로 재설정된 이후, 초당 수신된 평균 메시지 수를 보고한다.
- **초당 포워딩되는 평균 메시지 (매개변수 12)**  
이 매개변수는 해당 데이터가 마지막으로 재설정된 이후, 초당 전달된 평균 메시지 수를 보고한다.
- **초당 필터링된 평균 메시지 (매개변수 13)**  
이 매개변수는 해당 데이터가 마지막으로 재설정된 이후, 초당 필터링된 평균 메시지 수를 보고한다.
- **마지막 전원 리셋 이후 가동 시간 (매개변수 14)**  
이 매개변수는 장치의 마지막 전원 재설정 이후, NIU의 총 가동 시간을 보고한다.
- **포트 수 (매개변수 15)**  
이 매개변수는 NIU의 포트 수를 보고한다. "사용할 수 없음"이나 매개변수 데이터가 의미 없으므로 예약된 F16 를 제외한 모든 값 상태는 이 매개변수에 대한 유효한 데이터에 사용된다.
- **네트워크 상호 연결 장치 유형 (매개변수 16)**  
이 매개변수는 네트워크 상호 연결 장치 유형의 선언을 보고한다. NIU는 각 포트-쌍에 대해 다른 NIU 유형을 가리킬 있다. 네트워크 상호 연결 장치 유형 (Network Interconnection Unit Type) 값 할당은 표 7에 나와 있다.

#### b) 일반 매개변수 메시지 (GP)

##### - NIU 매개변수 요청 (N.GP\_Request)

CF는 N.GP\_Request를 이용하여 한 개 이상의 NIU 매개변수를 요청할 수 있다. 매개변수 0에 대한 요청은 매개변수의 전체 목록을 반환한다. 전송 프로토콜은 메시지 길이가 8바이트보다 클 때마다 사용된다. 7개 미만의 매개변수가 요청되면, 사용되지 않은 데이터 바이트는 FF16으로 설정한다. 매개변수 식별자는 숫자 오름차순으로 나열되어야 한다.

##### - NIU 매개변수 요청에 대한 응답 (N.GP\_Response)

요청 NIU 매개변수에 대한 NIU 응답에는 한 개 이상의 매개변수가 포함된다. 매개변수 0에 대한 요청에 대한 응답에는 모든 NIU 매개변수가 숫자 순서로 포함된다. 반환된 매개변수 목록이 예상보다 짧으면, NIU가 목록에 추가된 정보를 갖고 있지 않고 마지막으로 알려진 매개변수로 중지되었음을 나타낸다. 메시지 길이가 8바이트보다 클 때마다 전송 프로토콜을 사용한다. 메시지 길이가 8바이트 미만인 경우 사용하지 않은 데이터 바이트는 FF16으로 설정한다.

##### - 통계 매개변수 재설정 명령 (N.GP\_Reset\_Statistics)

이것은 재설정 가능한 모든 통계 매개변수를 지우는 명령 메시지이며, 표 4에 "예"로 표시되어 있다. 명령에 대한 승인은 승인 메시지(Acknowledgement Message) (PGN 59392)와 함께 제공된다.

#### c) 특정 매개변수 (SP)

##### - NIU-특정 매개변수 요청 (N.SP\_Request)

CF는 N.SP\_Request를 이용하여 하나 이상의 NIU 매개변수를 요청할 수 있다. 매개변수 0에 대한 요청은 매개변수의 전체 목록을 반환한다. 전송 프로토콜은 메시지 길이가 8바이트보다 클 때마다 사용한다. 6개 미만의 매개변수가 요청되면, 사용되지 않은 데이터 바이트는 FF16으로 설정한다. 매개변수 식별자는 오름차순으로 나열된다.

##### - NIU-특정 매개변수 요청에 대한 응답 (N.SP\_Response)

요청 NIU-특정 매개변수 메시지에 대한 NIU 응답에는 하나 이상의 매개변수가 포함된다. 매개변수 0의 요청에 대한 응답에는 모든 NIU 매개변수들이 숫자 순서로 포함된다. 예상보다 반환된 매개변수 목록이 짧으면, NIU가 목록에 추가된 정보를 갖고 있지 않고 마지막으로 알려진 매개변수로 중지되었음을 나타낸다. 전송 프로토콜은 메시지 길이가 8바이트보다 클 때마다 사용된다. 메시지 길이가 8바이트 미만인 경우, 사용되지 않은 데이터 바이트는 FF16으로 설정한다.

**- 통계 매개변수 재설정 명령 (N.SP\_Reset\_Statistics)**

이것은 재설정 가능한 모든 특정 통계 매개변수를 지우기 위한 명령 메시지이며, 표 4에 "예"로 표시된다. 명령의 승인은 승인 메시지 (PGN 59392)와 함께 제공된다.

**4) 연결 설정 기반 특정-목적지 메시지 포워딩**

그림 1의 네트워크는 게이트웨이 또는 라우터가 될 수 있는 NIU를 사용하여, 작업기 하부 네트워크를 ISO 11783 작업기 네트워크에 연결하는 네트워크 토폴로지를 보여준다. 각 네트워크 세그먼트(21)는 작업기 네트워크와 분리된 고유한 주소 공간을 갖는다. 이러한 세그먼트에는 기본 네트워크 세그먼트에서 CF와의 통신량이 적은 센서, 액추에이터 및 저용량 컨트롤러로 구성된 다수의 CF가 포함될 수 있다.

일반적인 라우터 (7.3 참조)는 한 네트워크 세그먼트의 CF에서 다른 주소 공간을 가진 다른 네트워크 세그먼트의 CF로 메시지를 전달하거나 라우팅하는 기능이 있다. 라우터의 주소 변환 데이터베이스는 메시지를 전달할 주소를 결정하는데 사용된다. 이 기능을 통해 서비스 도구는 단일 연결에서 여러 네트워크 세그먼트에 대한 진단을 수행할 수 있다. 소프트웨어나 캘리브레이션을 다운로드하는 것도 이러한 기능을 통해 보다 수월해질 수도 있다. 추가적인 예는, 다중 네트워크 세그먼트와 주요 작업기 네트워크의 일부인 VT (가상 터미널)간 통신이 필요한 CF가 있는 작업기이다.

주소 변환에 사용하는 대신, 다른 형태의 라우터는, 네트워크의 "검색" 토폴로지와 NAME에 표시된 대로 CF를 동작시키는 특정 기능을 기반으로 연결하기 위해, CF에 의해 동적으로 구성될 수 있다. 이 구현을 통해 연결이 이루어지면 NIU에서 메시지 전송을 처리할 수 있다. 통신에 참여하는 CF는, 연결 프로세스의 초기 토폴로지 검색과 연결 프로세스 개시 이후의, 단일 또는 다중-패킷 특정-대상 메시지에 대한 기존 소프트웨어 구현을 변경하지 않고 수행한다.

**■ 연결 설정**

그림 2는 네트워크 세그먼트 2의 CF C와 통신해야 하는 네트워크 세그먼트 1의 CF A에 대한 메시지 흐름도를 보여준다. CF A는 먼저 NIU B로 향하는 네트워크 메시지의 N.NTX 요청 기능을 사용하여 네트워크 2에 연결된 CF의 이름을 결정한다. NIU B는 N.NTX 응답 기능과 네트워크 2의 NAME과 주소 목록으로 응답한다. CF A가 이 목록에서 대상 CF C가 네트워크 2에 있다고 판단하면, NIU B에게 연결을 설정할 것을 요청한다. 즉, CF 간의 연결을 연다. NIU B는 메시지 포워딩 기능에 대한 연결 정보를 기록한 다음, CF C와 동일한 NAME을 사용하여 CF A의 네트워크에게 C^에 대한 주소를 요청함으로써, 네트워크 1에 가상 C 또는 C^를 생성한다. 요청된 주소는 네트워크 세그먼트 2에서 CF C가 요청한 주소와 반드시 같을 필요는 없지만, 자체 구성 가능한 주소 범위에서 네트워크 1에서 사용할 수 있는 주소이다.

NIU는 또한 CF A의 동일한 NAME과 사용 가능한 주소를 사용하여, 네트워크 2에게 A^에 대한 주소를 요청함으로써 네트워크 2의 가상 A^를 만든다. CF A가 네트워크에서 C^에 대한 성공적인 주소 요청을 수신하면, 네트워크 세그먼트에서 요청된 다른 주소로 보내는 것처럼 특정-목적지 메시지를 C^로 보내기 시작할 수 있다. NIU B는 목적지 주소가 C^ 인 메시지를 수신하면 해당 가상 CF 주소에 대한 메시지 포워딩 정보를 결정한다. 그리고 나서 A^의 주소를 메시지 헤더의 송신지 주소로 하고 CF C를 목적지 주소로 하여 네트워크 2에서 메시지를 재전송한다.

CF C는 A^로부터 메시지를 수신하고, 이 메시지에 대한 적절한 방식으로 응답한다. 여기에는 보안 접근 권한, 하드웨어가 설계한 작업 및/또는 필수 메시지로써 응답, 승인 또는 NACK (부정 승인)가 포함된다. 연결이 다른 CF에 의해 열려 있지 않은 경우, 연결을 설정한 CF에 의해 연결이 닫힐 수도 있다. 동일한 연결이 단일 CF에 의해 여러 번 열린 경우 단일 연결로 간주되므로 단일 닫기 명령으로 닫을 수 있다.

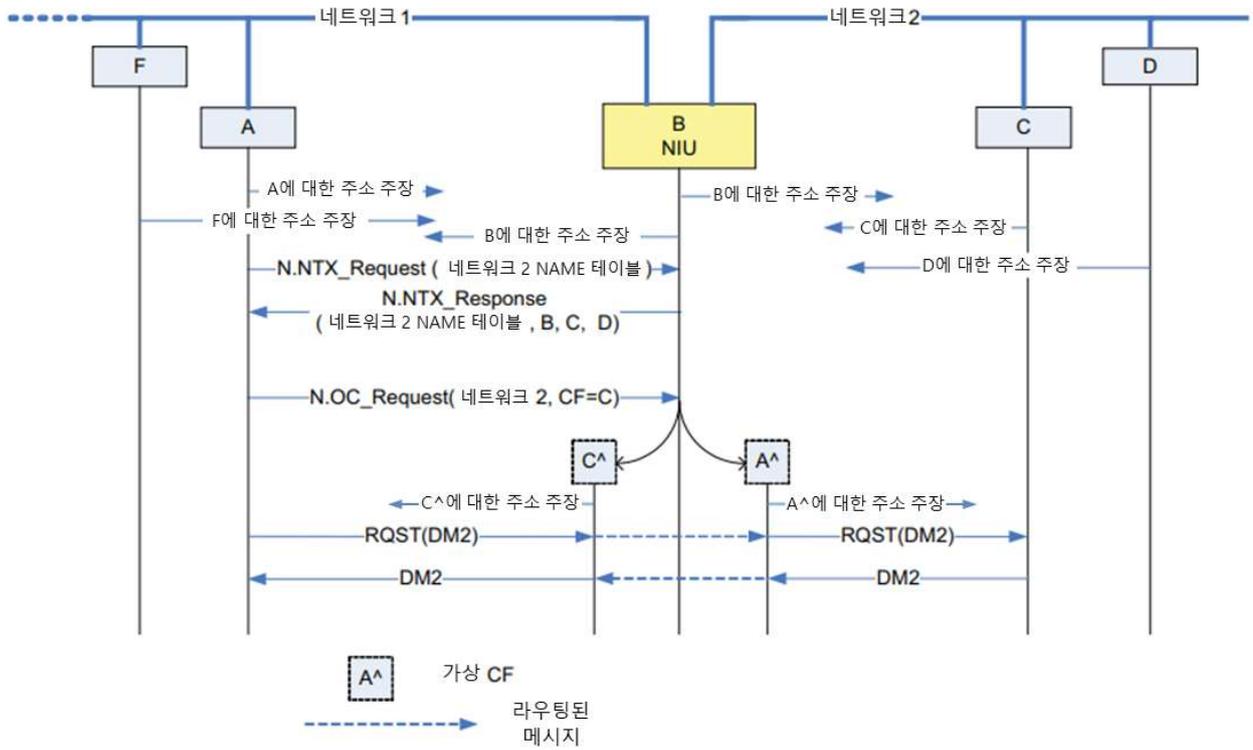


그림 2 - 메시지 전송을 위한 NIU 연결 설정

■ 다중 연결

그림 3은 네트워크 1의 CF A를 보여준다. 그림 2에서와 같이 먼저 CF C에 연결한 다음, 네트워크 2의 두 번째 CF D에 연결한다. A^는 네트워크 2에 이미 존재하므로, 연결하는 데 사용된다. D^의 주소는 네트워크 1에서도 요청된다. 이 경우, 첫 번째 연결이 닫히지 않았기 때문에, A^는 C^와 D^ 모두에게 메시지를 포워딩한다.

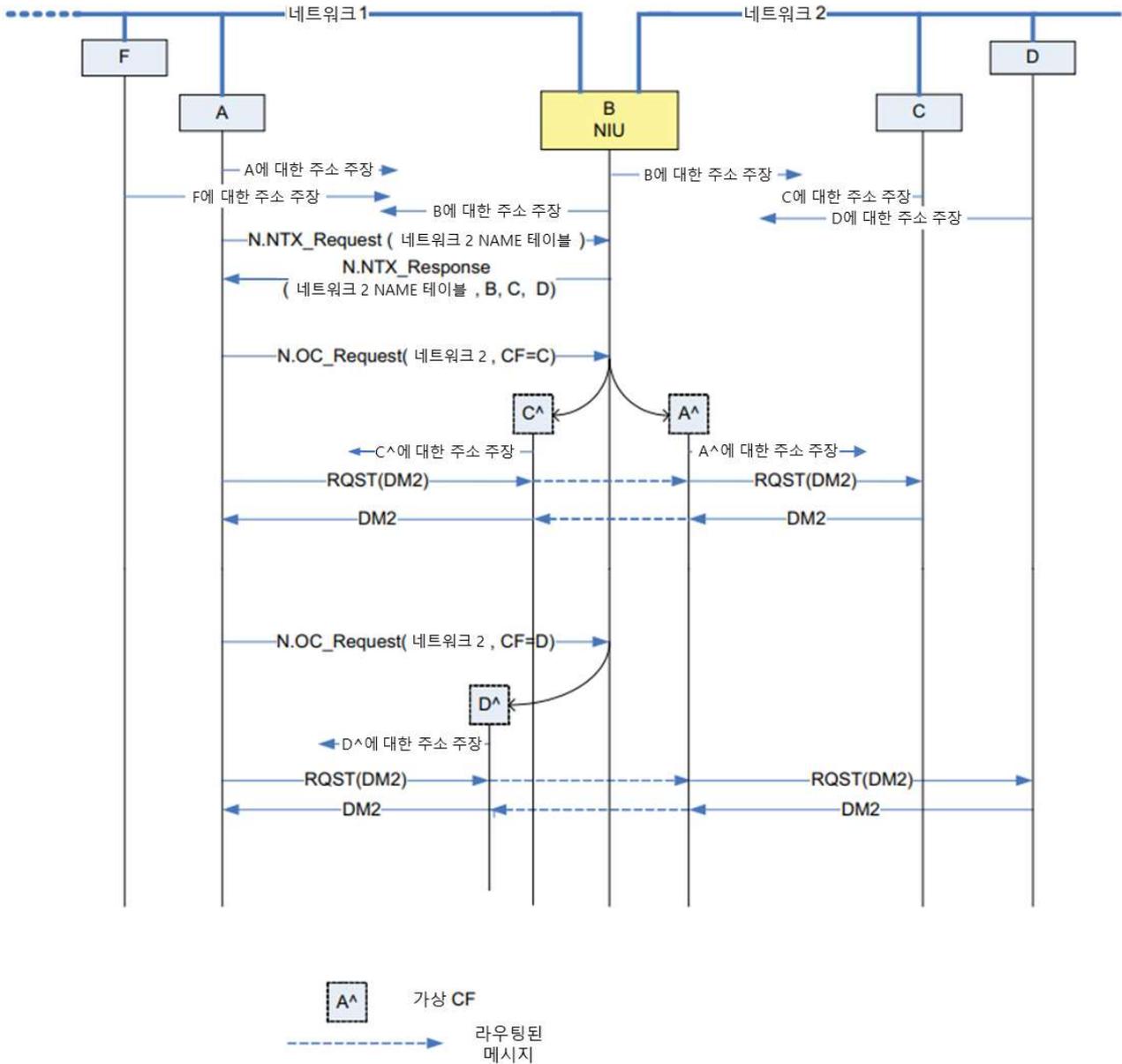


그림 3 - 네트워크 2의 두번째 CF로 NIU 연결 설정

그림 4에 표시된 또 다른 예는, 네트워크 2의 동일한 CF C에 연결하는 네트워크 1의 두 번째 CF F이다. CF F는 C^에 메시지를 보내지만, CF C에 등록된 개방 연결이 없기 때문에, ISO 11783-3에 정의된 대로 접근 거절 (Access Denied) 제어 바이트로 승인한다. CF F는 CF C와의 연결을 열도록 요청해야 한다. 그런 다음 F^는 네트워크 2에 설정되고, 네트워크 1의 C^는 재사용된다. 그런 다음 CF F는 네트워크 2의 CF C로 전달되는 메시지를 C^에 보낼 수 있다.

만일 네트워크 세그먼트에서 주소가 이미 요청된 경우, 연결을 추가로 열기 위한 추가 요청이 필요하지 않다.

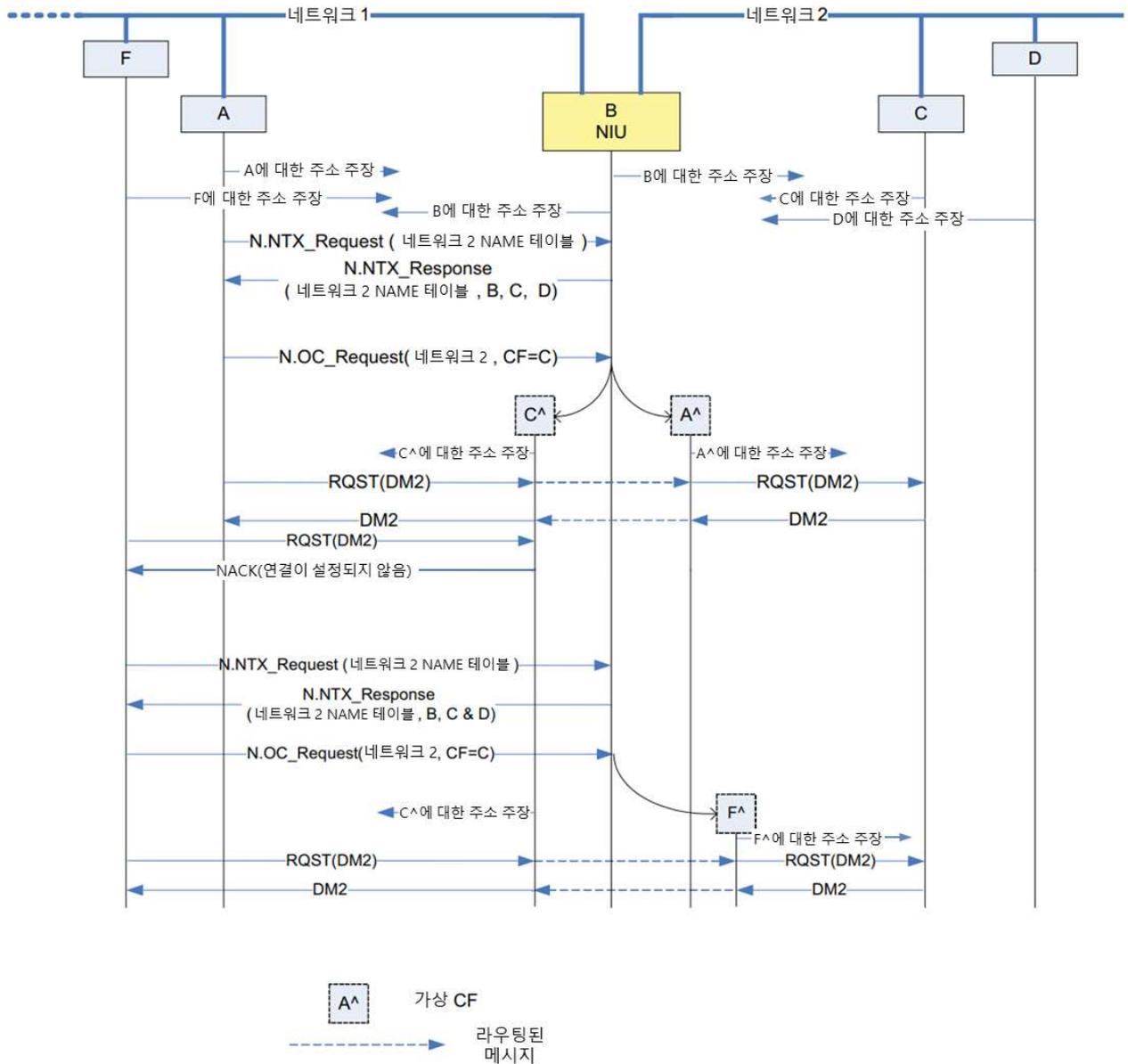


그림 4 - 네트워크 1의 두번째 CF로 NIU 연결 설정

위 예의 절차는, 전체 네트워크의 가장 먼 부분을 찾을 때까지, 네트워크 세그먼트 2의 다른 NIU 포트에 대한 제어 기능을 학습하기 위해 반복될 수 있다. 그림 5의 예를 참조한다.

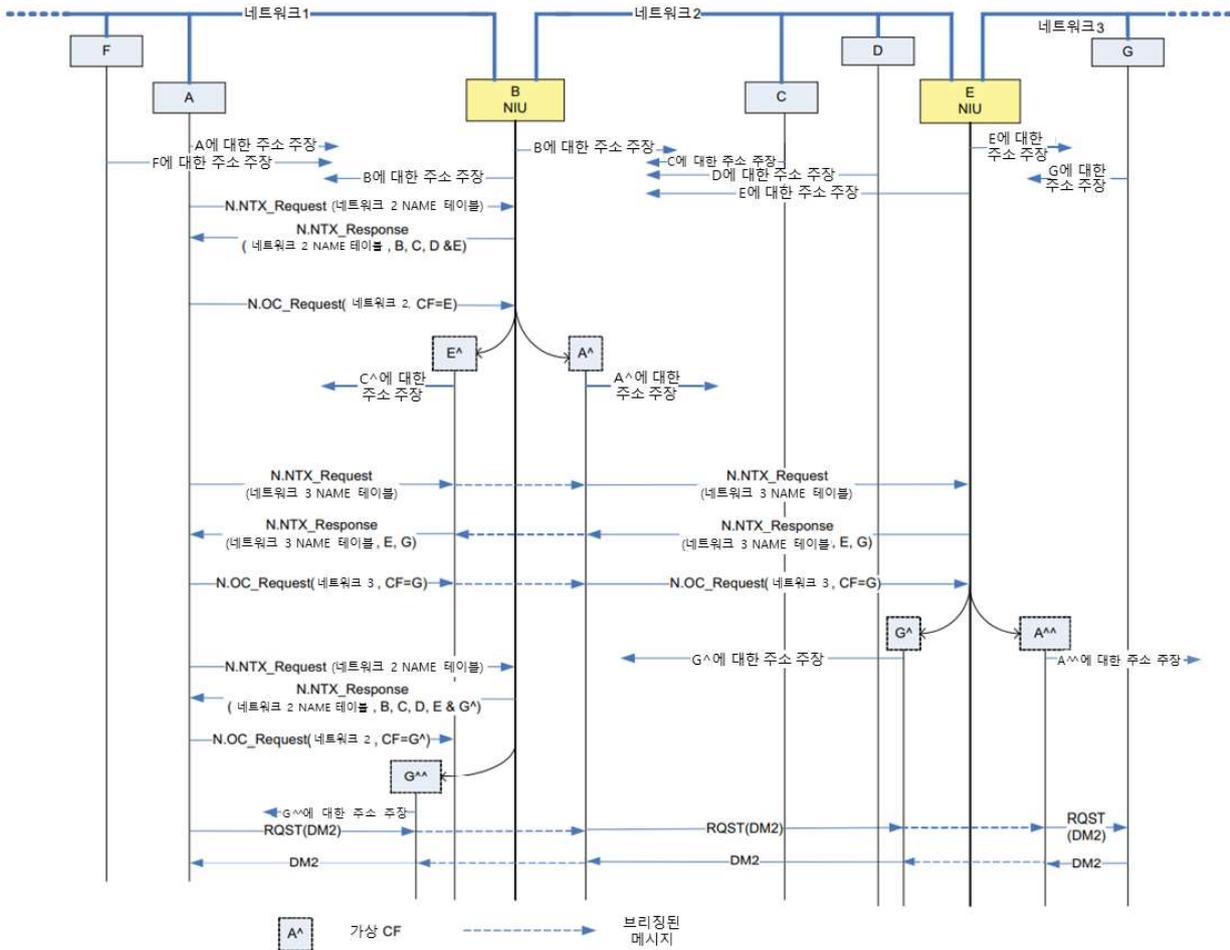


그림 5 - 다중-세그먼트 NIU 연결 설정

## ■ 연결 요구 사항

다음 요구 사항은 두 네트워크 세그먼트 간의 연결을 설정하는 NIU에 의해 완료되어야 한다.

- CF간에 연결이 열리면 현재 전원주기 동안 또는 연결을 연 원래 CF에 의해 연결이 닫힐 때까지, 두 CF간에 이러한 방식으로 메시지를 계속 보낼 수 있다.
- 전송 프로토콜이 필요한 메시지도 보낼 수 있으며, NIU는 전송 세션 타이밍 요구 사항을 따르길 위해 충분히 작은 지연으로 재전송해야 한다.
- 한 네트워크의 CF와 다른 네트워크 세그먼트의 CF사이에 원하는 수만큼의 가상 연결을 설정할 수 있으며, 이것은 연결을 유지하는 NIU의 기능과 네트워크 세그먼트당 총 253 개의 CF 주소를 갖는 일반적인 제약에 의해서만 제한된다. 사용할 수 있는 자체-구성 가능한 주소는 128 ~ 247이다.
- 원래 CF와 동일한 NAME을 갖는 가상 CF는 네트워크에서 표시되는 송신지 주소가 같거나 다를 수 있다.
- 연결 개방에 관련된 CF가 어떤 이유로든 주소를 변경하는 경우, NIU는 이 주소 변경을 추적하고 연결 정보를 업데이트한다.
- 연결 개방에 관련된 CF가 NAME의 일부를 변경하면, 우선 개방된 연결을 모두 닫아야 한다. 연결 개방에 관련된 CF가 개방된 연결을 먼저 닫지 않고 NAME의 일부를 변경하면 예측할 수 없는 NIU의 동작이 발생할 수 있다. 새로운 이름과 주소를 사용하여 새로운 연결이 설정된다.
- 연결 개방의 일부인 CF가 어떤 이유로든 주소 요청 메시지를 재전송하는 경우, NIU는 적절한 주소와 함께 가상 CF의 하부 네트워크에 대한 주소 요청도 보내야 한다.
- 연결 개방의 일부인 CF가 주소를 요청할 수 없고 CANNOT 요청을 보내는 경우, NIU는 가상 연결을 닫고 가상 CF의 하부 네트워크에 CANNOT 요청을 보낸다.
- 네트워크 세그먼트에는 CF의 가상 표현이 하나만 존재 해야 한다. 이것은 여러 연결에 대한

연결을 가질 수 있다.

- j) 개방된 연결을 닫아도 네트워크에서 가상 CF가 제거되지 않는다. 가상 CF는 해당 가상 CF와 관련된 모든 연결이 닫힐 때까지 연결된 상태를 유지한다.
- k) NIU가 다른 네트워크 세그먼트 (포트-쌍)에 대한 NAME과 주소 테이블을 보고할 때, 해당 네트워크 세그먼트에 대한 자체 NAME과 주소 또한 보고해야 한다.

가상 CF는 네트워크의 다른 CF에게 다른 CF로 인식되기 때문에, 실제 CF는 연결을 열지 않고도 통신을 설정할 수 있다. NIU는 해당 가상 CF에 대해 개방된 연결에서, 현재 구성되지 않은 CF로부터 가상 CF가 수신한 메시지에 대해 부정 승인(NACK)해야 한다. 승인 메시지는 주소 승인(ADD\_NACK) 매개변수에 가상 CF의 주소를 포함해야 한다. 제어 바이트는 "접근 거부"(2)로 설정되어야 한다.

## ■ 연결 메시지

### a) 다른 네트워크 세그먼트에 대한 연결 개방 요청 (N.OC\_Request)

연결을 설정하기 위한 NIU 요청 메시지에는, 목적지 네트워크 세그먼트의 포트-쌍 ID와 목적지 네트워크 세그먼트에서 원하는 CF의 NAME이 포함된다. 이 메시지는 요청하는 CF에서 NIU로 전송된다. 이 형식의 메시지는 전송 프로토콜을 사용한다.

요청하는 CF는 네트워크 메시지의 "요청 송신지 주소 및 NAME 목록"(6.7.2.3) 기능을 사용하여 목적지 네트워크 세그먼트에서 CF의 ID를 얻고 원하는 CF를 선택해야 한다. 연결 개방 요청이 현재 활성화된 연결과 정확히 일치하는 경우, 평소와 같이 응답이 전송되며 연결 상태는 변경되지 않는다.

### b) 다른 네트워크 세그먼트에 대한 연결 종료 요청 (N.CC\_Request)

연결을 종료하기 위한 NIU 요청 메시지에는 목적지 네트워크 세그먼트의 포트-쌍 ID와 목적지 네트워크 세그먼트에서 원하는 CF의 NAME이 포함된다. 이 메시지는 요청하는 CF에서 NIU로 전송된다. 요청자는 원래 연결을 연 CF이어야 하며, 이 형식의 메시지는 전송 프로토콜을 사용한다. 요청하는 CF는 연결을 개방한 CF이어야 하며, 연결을 개방하는데 사용한 목적지 네트워크 세그먼트의 CF에 대해 동일한 NAME을 사용해야 한다.

### c) 연결 시작 요청에 대한 응답 (N.OC\_Response)

연결을 설정하기 위한 NIU 응답 메시지에는 성공 또는 실패 표시와 해당되는 경우 실패 이유가 포함된다. 이 메시지는 NIU에서 요청하는 CF로 전송된다. 바이트 3은 성공 또는 실패를 나타내고 바이트 4는 실패 이유를 나타낸다.

### d) 연결 종료 요청에 대한 응답 (N.CC\_Response)

연결을 닫기 위한 NIU 응답 메시지에는 성공 또는 실패 표시와 해당되는 경우 실패 이유가 포함된다. 이 메시지는 NIU에서 요청하는 CF로 전송된다. 바이트 3은 성공 또는 실패를 나타내고 바이트 4는 실패 이유를 나타낸다.

## 5) 네트워크 상호 연결 장치 유형

### ■ 리피터

리피터는 기본적으로 네트워크의 모든 CF에 투명하다. 리피터는 하위-비트-시간 간격과 안티-루프백/록아웃(anti-loopback/lockout)으로 한 세그먼트에서 다른 세그먼트로 신호를 재생성하여, 동일한 데이터 속도로 동작하는 버스 세그먼트 간에 메시지를 전달한다. 리피터는 메시지를 필터링할 수 없으므로 모든 메시지를 전달한다. 하위-비트-시간 지연만으로 이 작업을 수행하기 때문에, 최대 전송 지연은 비트 시간의 10% 미만이어야 한다(250 kbits/s에서 400ns). 이것은 비트-단위 중재가 리피터 전체에서 올바르게 발생하도록 하면서도, 여전히 합리적인 전파 지연(케이블 거리)을 허용한다.

오류 격리가 제공되는 경우, 리피터는 세그먼트의 버스 오류가 감지될 때 하나 이상의 송신기를 비활성화 할 수 있다.

### ■ 브리지

브리지는 기본적으로 네트워크의 모든 CF에 투명하다. 브리지는 버스 세그먼트간에 메시지를 전달하고 필터링한다. 이러한 작업을 수행할 때 메시지도 저장한다. 필터링을 통해 네트워크의 각 세그먼트에 존재하는 버스 트래픽의 양을 효과적으로 줄일 수 있다. 브리지는 애플리케이션에 따라 수신하는 메시지의 일부 또는 전부를 필터링하거나 필터링을 전혀 하지 않을 수 있다.

브리지를 통한 약간의 전송 지연이 있으며 최대 값은 애플리케이션에 따라 다르다. 그러나 ISO 11783의 이 부분에서 권장하는 최대 전송 지연은 10ms이다.

## ■ 라우터

주소 요청 메시지는 라우터를 통과하지 않는다. 일단 작동하면, 라우터는 기본적으로 네트워크의 모든 CF에 투명해야 한다.

### a) 주소 변환

전달 및 필터 기능 외에도 라우터는 한 포트 (버스 세그먼트)에서 다른 포트 (버스 세그먼트)로 주소를 다시 매핑 할 수 있다.

### b) 라우터 관리

필수는 아니지만, 전달, 필터 및 주소-변환 데이터베이스를 구성하기 위한 표준 접근을 제공하려면 데이터 관리 기능이 지원되어야 한다. 포워딩과 필터링 이외에도, 라우터는 어느 한 포트나 버스 세그먼트에서 다른 포트나 버스 세그먼트로 주소를 재매핑하거나 메시지를 라우팅하므로 하위 시스템이 네트워크의 다른 부분에게 단일 주소로 나타날 수 있다. 메시지 필터 데이터베이스는 일반적으로 통과 모드(1)로 구성되며 특정 항목이 있는 메시지를 제외한 모든 메시지가 차단된다. CF는 시스템의 다른 개별 CF (주소)에 대한 특별한 지식을 필요로 하지 않기 때문에 CF 개발을 단순화할 수 있다. 주소 요청 메시지는 라우터를 통과 할 수 없으므로 주소 변환 맵을 제공하기 위한 "조회" 테이블이 있어야 하며, 약간의 변환과 포워딩 지연이 있다.

## ■ 게이트웨이

전술한 라우터의 수행 작업에 추가하여, 게이트웨이가 수행하는 주요 작업은 메시지 재패키징이다. 이것은 하부시스템의 다른 개별 CF (주소)에 대한 특정 지식을 필요로 하지 않는 CF의 개발을 단순화할 수 있다. 일단 작동하면, 게이트웨이는 기본적으로 네트워크의 모든 CF에 투명해야 한다.

### a) 메시지 재패키징

게이트웨이는 하나 이상의 메시지에서 "새" 메시지로 매개변수를 다시 패키징할 수 있다. 이를 통해 여러 CF의 매개변수를 그룹화하여 다른 CF가 송수신과 해석을 좀 더 쉽게 할 수 있다.

### b) 게이트웨이 관리

필수는 아니지만 전달, 필터, 주소-변환 및 메시지-재패키징 데이터베이스 구성에 대한 표준 접근을 제공하려면 데이터베이스 기능이 지원되어야 한다. 라우터가 기본적으로 수행하는 작업이외에도, 게이트웨이는 하나 이상의 메시지에서 "새" 메시지로 매개변수를 다시 패키징 할 수 있다. 이를 통해 여러 CF의 매개변수를 그룹화하여 다른 CF가 송수신과 해석을 좀더 쉽게 할 수 있다. 라우터와 마찬가지로 CF는 시스템의 다른 개별 CF (주소)에 대한 특별한 지식이 필요하지 않으므로 CF 개발을 단순화할 수 있다.

마찬가지로 메시지 필터 데이터베이스는 일반적으로 통과 모드(1)로 구성되며, 특정 항목이 있는 메시지를 제외한 모든 메시지가 차단된다. 게이트웨이를 통한 약간의 변환, 재패키징 및 포워딩 지연이 있으며, 재패키징을 위한 데이터베이스를 포함하는 메시지-구축 기능이 필요하다.

## ■ 트랙터 ECU

트랙터 ECU는 트랙터나 자주식 작업기에 있는 트랙터 버스와 작업기 버스를 연결하는데 사용되는 특수 NIU이다.

### 3) 국제표준 기술 개발

#### 가) 국제표준 추진 절차

##### 가-1) ITU-T

본 연구에서 국제표준화를 추진하고 있는 ITU는 ISO, IEC와 함께 3대 공적 표준화 기구중의 하나이다. ITU는 유럽 20개국이 전신서비스의 국가간 상호접속을 용이하도록 체결한 국제전신협약(International Telegraph Convention)을 바탕으로 1865년 5월 17일 설립된 국제전신연합(International Telegraph Union)을 모체로 하여 설립되었다.

현재는 UN 산하 전문기구인 ITU는 3개의 분과로 구성되어 운영되는데, 유·무선 통신, 전파, 방송, 위성 주파수 등에 대한 규칙(Regulation)은 ITU-R 및 표준(Recommendation)의 개발·보급은 ITU-T 그리고 국제적인 협력을 담당하는 ITU-D가 바로 그것이다.

ITU-T에서의 표준개발을 위해 국가간 참여한 정책 사항을 표준화하는 TAP(Traditional Approval Process) 방식과 기술 문서를 표준화하기 위한 AAP(Alternative Approval Process) 방식을 병행하여 운영한다.

TAP 방식보다 표준 제정기간이 상대적으로 짧은 AAP 방식은 통상 3년 정도의 국제표준 제정 기간을 가능 한다. ITU-T에서의 전통적인 표준화는 표준화 과제 제안으로부터 시작된다. 신규 표준화 과제에 대한 SG(Study Group)회의에서의 승인과 회원국의 승인에 따라 편집자(Editor)가 선정되는데, AAP 방식에서는 1년에 1번 있는 SG 회의에서만 신규 과제를 제안하는 것이 아니라 WP(Working Party)회의에서도 신규 과제를 제안할 수 있다.

신규 과제가 된 표준화 과제는 1년에 2~3회 개최되는 회의(SG회의와 WP회의)를 통하여, 지속적인 개정 작업이 이루어진다. ITU-T에서는 개정을 위해 문서화된 개정 의견을 제안할 것을 요청하는데, 이를 ITU-T에서는 contribution 문서라고 일컫는다. Editor는 회의에 제안되는 contribution 문서에 따라 개발하고 있는 권고안을 수정한 후, 이를 매 회의 마지막에 회람한다.

SG 혹은 WP 회의를 통해, 개발되는 문서가 표준으로 인정할 만큼 안정화가 되었다면, TAP 방식에서는 약 9개월에 걸친 협의 기간을 가진 후 ITU-T 표준(혹은 권고안, Recommendation)으로 발간한다. AAP 방식에서는 9개월의 협의 기간 대신, 4주의 Last call 기간을 두어 정보통신 분야와 같이 빠른 표준화가 필요로 하는 표준 개발을 가능하게 한다.

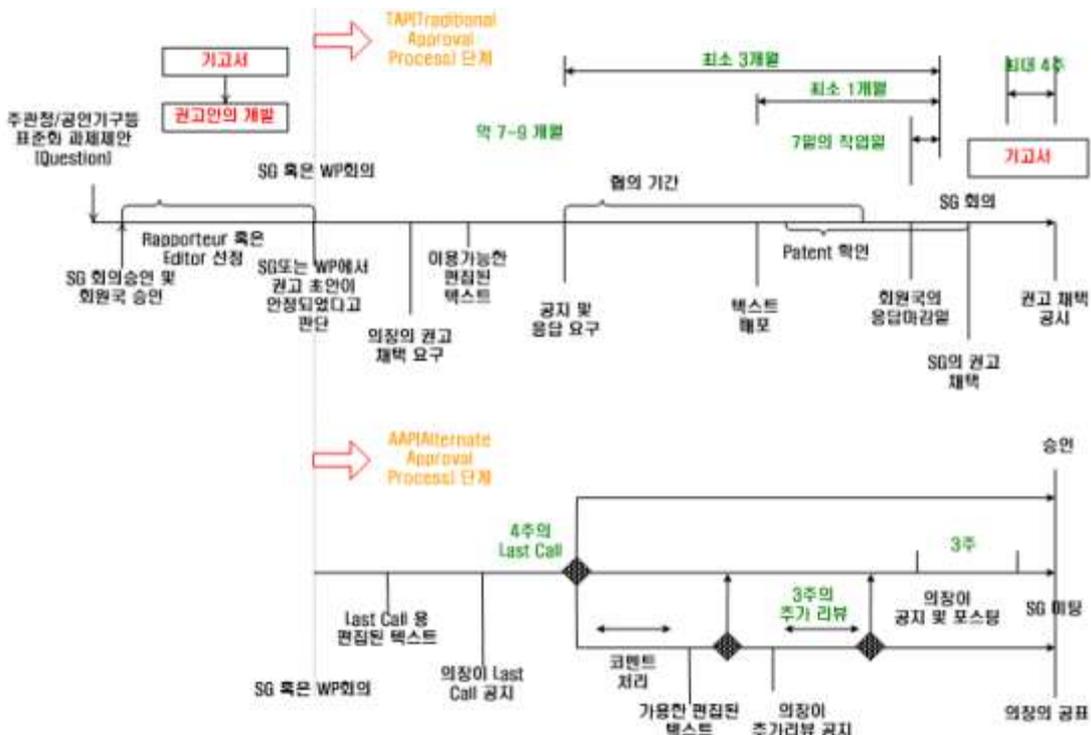


그림 100 - ITU-T 표준화 절차

2017년부터 2021년까지의 회기에 있어, ITU-T는 총 11개의 연구반을 구성하여 운영하고 있다. 이중 SG20은 IoT와 스마트시티에 대한 표준을 개발하는 연구반으로서 세부 내용으로 스마트팜 관련 표준화를

추진하고 있다. 그렇기 때문에 본 사업에서는 ITU-T SG20 연구반의 Q2 작업반과 Q4 작업반에서 각각 Y.IoT-SLF(축산분야)와 Y.ISG-FR(시설원예분야) 국제표준화를 추진하고 있다.



그림 101 - ITU-T Study Group 구성 (2017~2021)

## 가-2) IEC

ISO, ITU-T와 함께 IEC (International Electrotechnical Commission, 국제전기표준회의)는 UN 산하의 국제 표준화기구로서 전기 및 전자기술분야에서의 표준화와 관련된 모든 문제 및 관련사항에 관한 국제협력을 촉진하고 이에 따라 국제적 의사 소통을 도모하려는 집단이다. IEC의 표준화 절차는 문서의 수준에 따라 다음과 같이 진행된다.

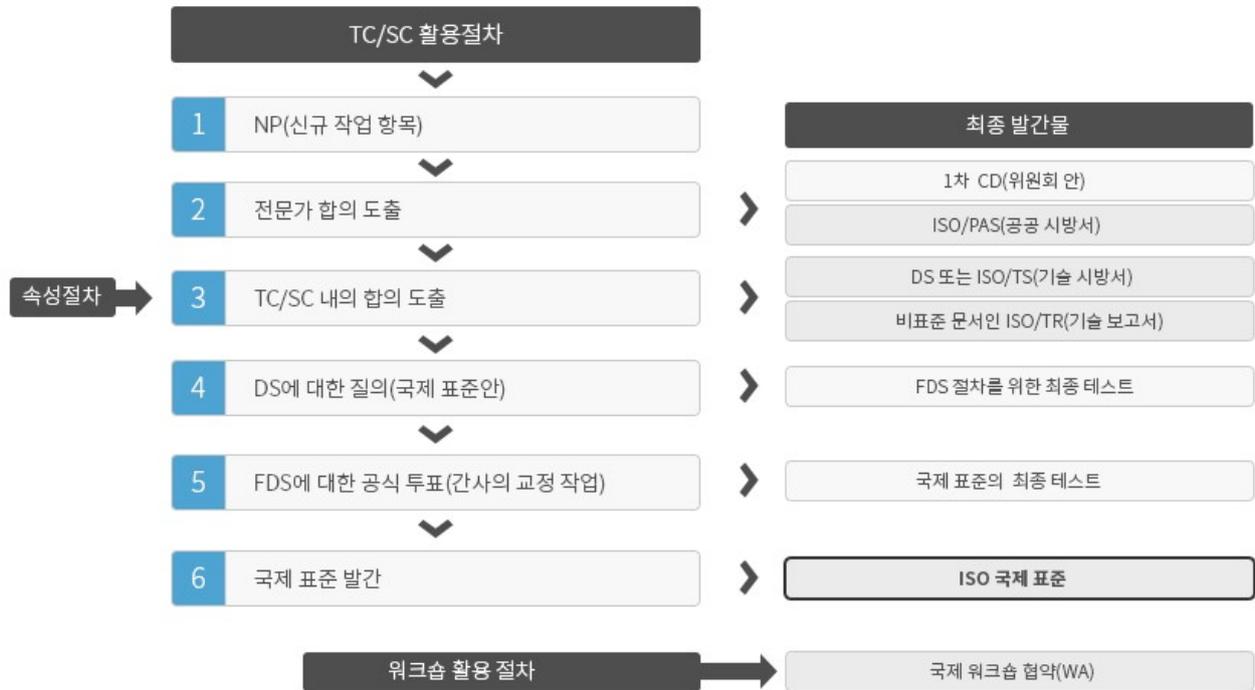


그림 1 - IEC의 표준화 절차

IEC에서 정의하는 문서의 수준이란 다음과 같다.

- NP/NWIP: 신규 작업 항목(New Work Item Proposal)을 관련된 기술위원회나 분과위원회에서 회의를 통해, 또는 3개월간 서면 투표를 통해 승인
- WD: 국제표준 의 작업 초안(Working Draft)를 작성하는 단계
- CD: 위원회 안(Committee Draft)에 대한 해당 위원회 회원 간의 합의가 이루어져야 하는 단계
- DIS/CDV: 위원회 단계를 완성하여 작성된 국제표준안(Draft International Standard) - IEC의 경우는 투표를 위한 위원회 안(Committee Draft for Vote) - 을 중앙사무국 관장 하에 투표를 실시하는 단계로 ISO의 경우는 3개월, IEC의 경우는 5개월 투표 기간을 가짐
- FDIS: 최종 국제표준안(Final Draft International Standard)을 가지고 중앙사무국에서 2개월 투표 회람을 실시하는 단계
- IS: 해당 위원회의 간사와 최종 감수를 거쳐 중앙 사무국에서 국제표준을 출판하는 단계

나) 국제표준 제정

나-1) ITU-T Y.4466 ('17~'20)

(가) 제정 경과

표 1 - ITU-T Y.4466 표준 제정 경과

순번	기고서 제목	제출일
1	ITU-T Y.ISG-fr : Proposed texts for scope, terms, definitions and conventions	2017-07-17
2	Y.ISG-fr: Proposed texts for terms definition	2017-09-04
3	Y.ISG-fr: Proposed text for information and control flow of smart greenhouse	2017-09-04
4	Y.ISG-fr - Proposed modification on funtional requirements	2018-01-15
5	Y.ISG-fr-detail description of functional element in referen...	2018-01-18
6	Y.ISG-fr-Proposed text for detail description of SG service framework	2018-01-18
7	Y.ISG-fr-Proposed text for reference architecture	2018-05-06
8	Y.ISG-fr - Proposed modification of functional requirements	2018-05-07
9	Y.ISG-fr: Proposed updated text of clause 8	2018-12-07
10	Y.ISG-fr: Proposed updated text of clause 9	2018-12-07
11	Y.ISG-fr: proposed modifications on clause 10	2019-07-22
12	Y.ISG-fr: proposed modifications on Appendix II	2019-07-22
13	Y.ISG-fr: proposed modifications on clause 9	2019-07-22
14	Y.ISG-fr: proposed modifications on clause 8	2019-07-22
15	Y.ISG-fr: proposed modifications on clause 7	2019-07-22
16	Y.ISG-fr: proposed new terms and editorial changes from clause 3 to clause 5	2019-07-22
17	Y.ISG-fr: proposed modifications on clause 6	2019-07-22
18	Y.ISG-FR: Proposed overall modifications	2019-09-23
19	Y.ISG-fr : Proposed text for consent	2019-10-31

(나) 주요 내용

ITU-T Y.ISG-fr은 작물을 키우기 위해 사용되는 다양한 센서와 구동기로 이루어진 인공공간을 스마트 온실이라고 정의하고, 이러한 환경을 통해 농민 혹은 기업들에게 새로운 서비스를 제공할 수 있는 모델을 정의한다.

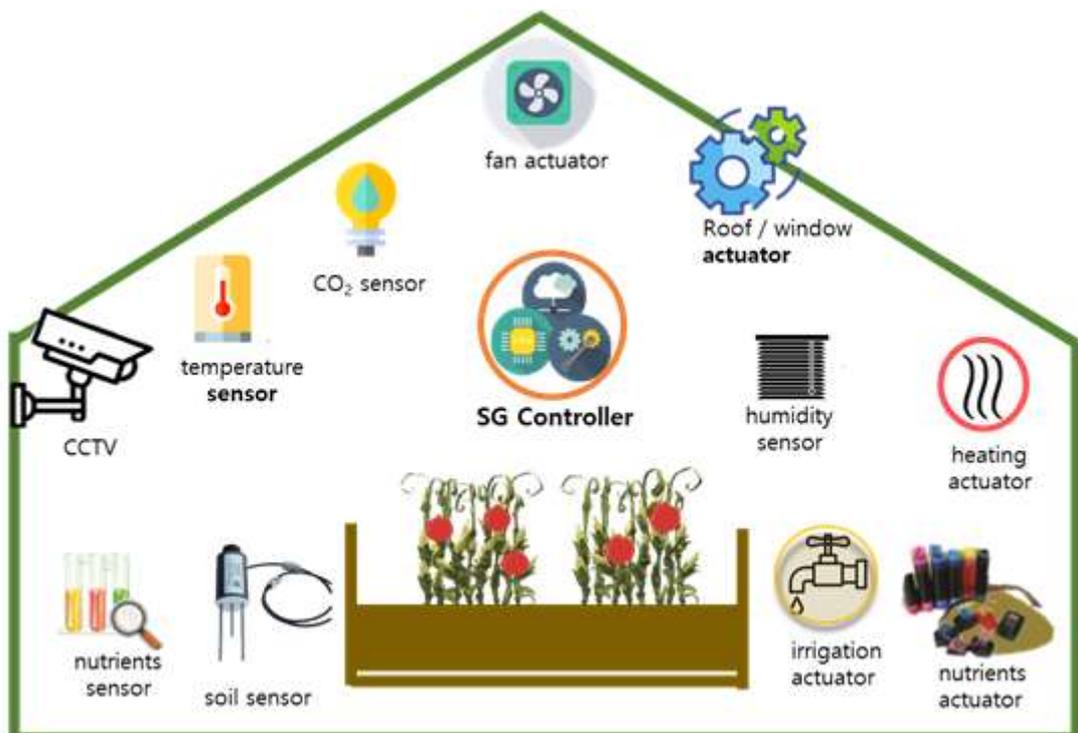


그림 1 - ITU-T Y.ISG-fr에서 고려하는 스마트온실 기본 개념도

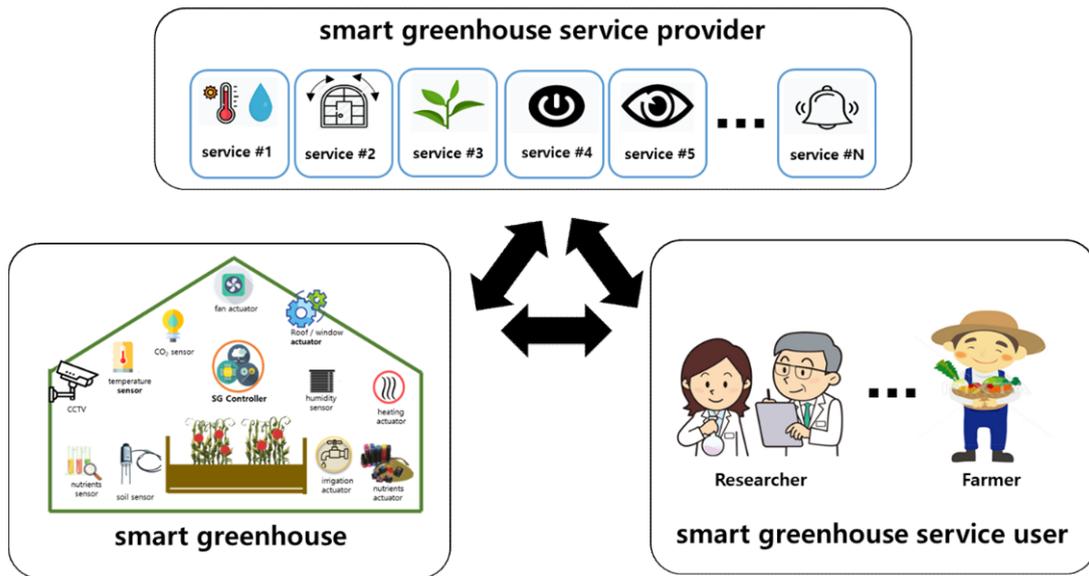


그림 2 - ITU-T Y.ISG-fr에서 정의하는 스마트온실 서비스 개념도

이 표준은 온실 환경에서 작물을 재배하기 위한 최적의 조건을 제공하고 유지하는 스마트 온실 서비스를 위한 참조 아키텍처를 기술하며, 스마트 온실 서비스의 프레임워크에서 다루는 범위는 다음과 같은 내용을 포함한다.

- 스마트 온실 서비스 개요
- 스마트 온실 서비스를 위한 참조 아키텍처
- 스마트 온실 서비스 인터페이스
- 스마트 온실 서비스 이용 사례(부록)

당해연도에서는 IoT와 스마트팜 서비스의 연관성을 보다 구체적으로 기술함으로써, 스마트팜이 스마트 공장, 스마트차량 등과 같이 IoT를 이용하는 다른 산업과 유연하게 연동할 수 있도록 하였다. 스마트온실 서비스는 다음 그림과 같으며, SG 응용 계층, 서비스 및 응용 지원 계층, 디바이스 및 리소스 계층으로 구성되며, 스마트온실 서비스 프레임워크는 SG 응용 계층과 디바이스 및 리소스 계층 사이에 존재한다. 스마트온실 서비스 프레임워크는 다음과 같은 5개의 기능 요소로 구성된다.

- SG 센싱 기능은 온실의 환경 상태 및 생육 상태에 대한 정보를 모니터링 및 측정하여 SG 제어 기능으로 전달한다.
- SG 구동 기능은 구동기의 작동 상태를 모니터링하고 SG 제어 기능에서 전달되는 구동기 제어 명령에 따라 구동기가 동작하도록 구동기에게 전달한다.
- SG 제어 기능은 SG 운영 기능에서 제공되는 최종 환경 제어 조건에 따라 관련 구동기 제어 명령을 생성하며, 이는 온실에 설치된 각 구동기의 능력과 각 구동기에 대한 농부의 요구 사항을 반영합니다. SG 제어 기능은 SG 센싱 기능에서 SG 제어 기능으로 전달되는 환경 및 생육 정보도 제공한다.
- SG 운영 기능은 온실의 특정 작물에 대한 최종 환경 제어 조건을 생성하며, 이는 선택된 최적 생육 모델의 농가 경험을 반영한다. SG 운영 기능은 SG 제어 기능에서 통합 SG 관리 기능으로 제공되는 환경 및 생육 정보도 제공한다.
- 통합 SG 관리 기능은 SG 운영 기능에서 제공되는 각 작물의 환경 및 생육 정보를 축적하여 농업 전문가 그룹 또는 전문가 시스템에 의한 일반적인 최적 환경 모델을 창출하고 저장하는 인터페이스를 제공한다. 일반적인 최적 환경모델은 농부의 요구사항에 따라 SG 운영 기능에게 제공한다.

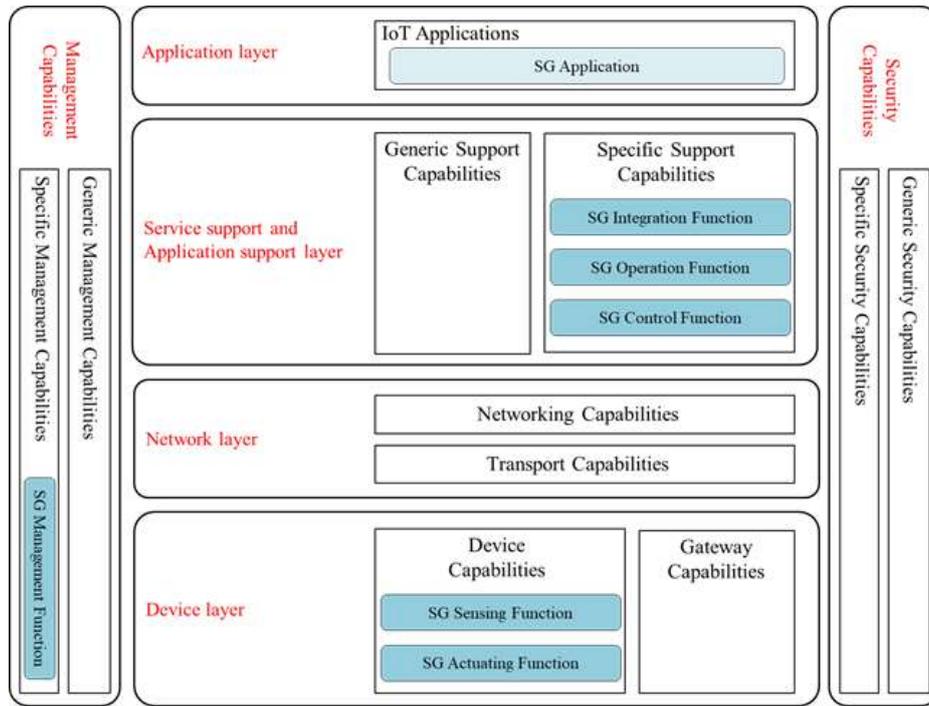


그림 3 - IoT 기반의 스마트팜 기능 구조

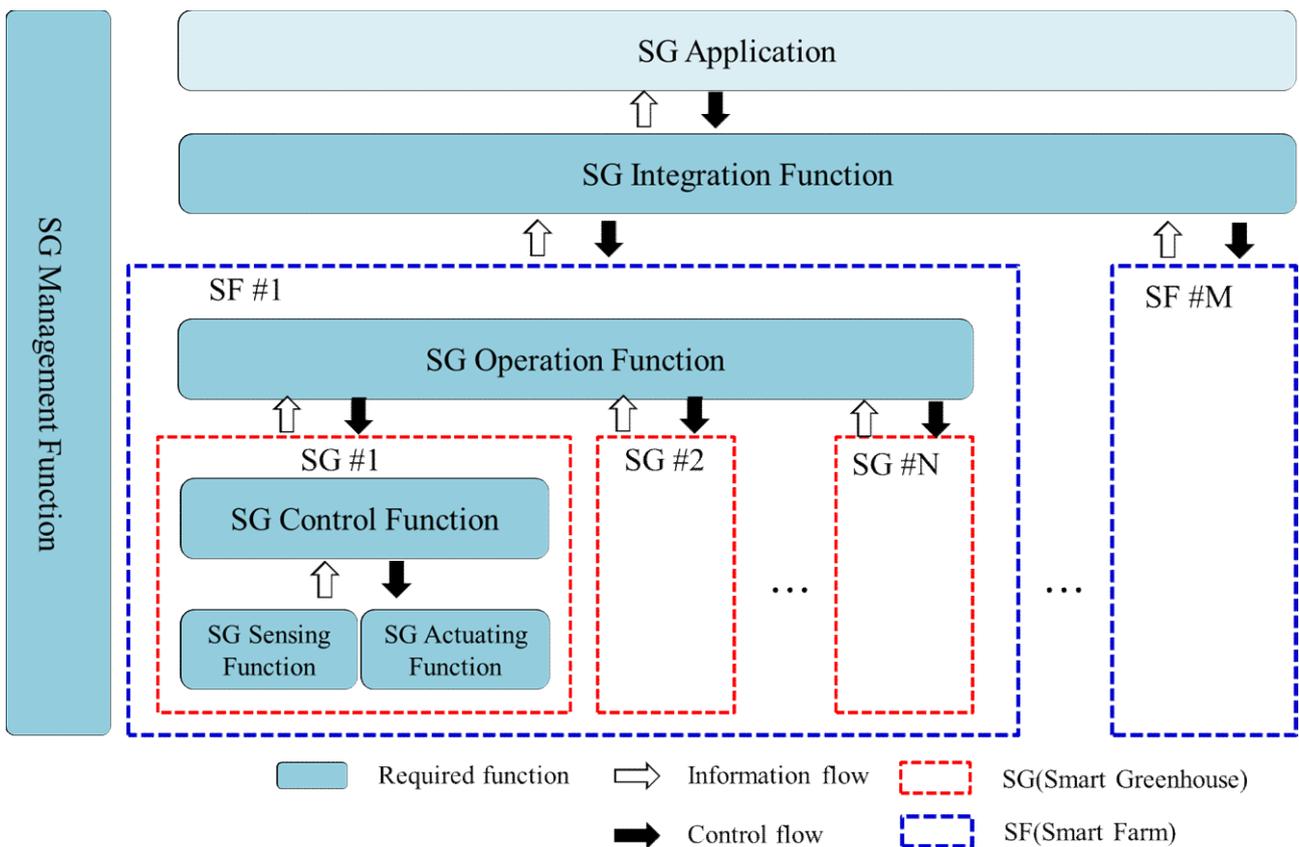


그림 4 - SG 서비스의 정보 흐름

상기 그림은 스마트온실에서의 정보 흐름을 나타낸 그림이다.

- SG 환경 정보 및 작물 성장 정보의 생산 및 소비 : SG 감지 기능에 의해 생성된 SG의 환경 상태 및 작물 성장 상태는 SG 제어 기능에 의해 수집된다. SG에 관한 수집된 정보는 SG 작동 기능으로 전달된 이후, SG 통합 기능으로 전송된다. 마지막으로 SG 통합 기능은 농업 전문가 그룹 또는 전문

가 시스템과 같은 정보를 사용자에게 제공한다.

- 작물 재배 조건 생성 및 구동기 제어 명령 실행 : 이전 정보 흐름에서 제공 한 정보는 농업 전문가 그룹과 전문가 시스템에서 각 작물에 대한 최적의 성장 조건을 도출하는 데 사용된다. SG 통합 기능은 농업 전문가 그룹 전문가 시스템에서 최적의 작물 성장 조건을 검색하고 유지한다. 그런 다음 SG 작동 기능에 최적의 작물 성장 조건을 제공한다. SG 작동 기능은 사용자가 SG 통합 기능에 의해 제공되는 최적의 성장 조건을 추가로 수정할 수 있게 한다. 그런 다음 수정된 최적 성장 조건을 SG 제어 기능으로 전달한다. 마지막으로 SG 제어 기능은 구동기 제어 명령을 적절한 SG 작동 기능으로 제공함으로써 구동기가 올바르게 작동하도록 한다.

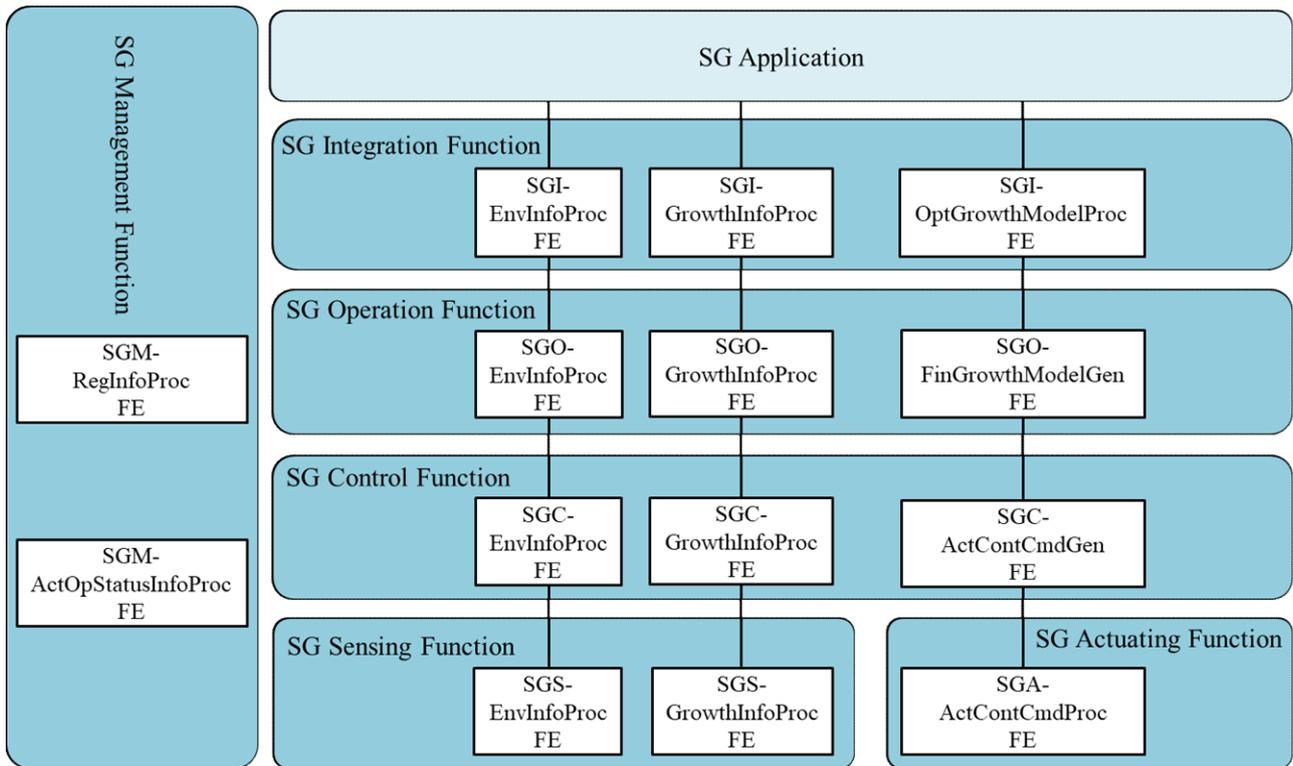


그림 5 - SG 서비스 기능 구조

- SG 센싱 기능  
SGS-EnvInfoProcFE와 SGS-GrowthInfoProcFE로 구성된 SG 센싱 기능은 SG 및 작물 재배 정보에 대한 환경 정보를 수집하고 SG 제어 기능에 센싱정보를 제공하는데, 각 FE의 세부 사항은 다음과 같다.
  - SGS-EnvInfoProcFE (SGS-Environmental Information Processing FE)는 다양한 유형의 환경 센서 (IoT 장치)에서 SG의 환경 정보를 수집한 이후 SGC-EnvInfoProcFE에 정보를 제공한다.
  - SGS-GrowthInfoProcFE (SGS-Growth Information Processing FE)는 다양한 유형의 성장 센서 (IoT 장치)에서 SG로 작물의 작물 성장 정보를 수집 한 다음이 FE는 수집 된 작물 성장 정보를 SGC-GrowthInfoProcFE에 제공한다.
- SG 구동 기능  
SGA-ActContCmdProcFE로 구성된 SG 구동 기능은 SG 제어 기능에서 구동기 제어 명령을 수신 한 후 해당 온실에 설치된 구동기로 전달한다. 각 FE의 세부 내용은 다음과 같다.
  - SGA-ActContCmdProcFE (SGA- 액추에이터 제어 명령 처리 FE)는 SGC-ActContCmdGenFE로부터 특정 액추에이터 속성뿐만 아니라 제어 명령을 수신하여 해당 구동기(관련 IoT 장치)에 전달한다.
- SG 제어 기능  
SG 제어 기능은 SG의 환경 정보와 온실의 작물 성장 정보를 수집하여 SG 운영 기능에 제공한다. 또한, SG 제어 기능은 SG 작동 기능으로부터 온실에서 재배 된 작물에 대한 환경 조건을 수신하고 온실에서 작물의 작물 재배 환경을 유지하기 위해 액추에이터에 제어 명령을 제공한다. SG 제어 기능에 대한 각 FE의 자세한 역할은 다음과 같다.

- SGC-EnvInfoProcFE (SGC-Environment Information Processing FE)는 SG Sensing 기능의 SGS-EnvInfoProcFE를 통해 SG의 환경 정보를 수집 한 다음이 FE는 SGC-ActContCmdGenFE에 정보를 제공하여 액추에이터 작동을 조정한다.
- SGC-GrowthInfoProcFE (SGC- 성장 정보 처리 FE)는 SGS-GrowthInfoProcFE를 통해 SG에서 작물 성장 정보를 수집 한 다음 SGO-GrowthInfoProcFE 및 SGC-ActContCmdGenFE에 정보를 제공한다. 이때 SGC-GrowthInfoProcFE는 '특정 시간에', '정기적으로', '주문형' 또는 '사건이 있을 때' 등과 같이 작물 성장 정보를 수집하기 위한 요청 방법을 지정한다.

- SG 운영 기능

SGO-EnvInfoProcFE, SGO-GrowthInfoProcFE 및 SGO-FinGrowthModelGenFE로 구성된 SG 운영 기능은 SG 제어 기능에서 온실의 환경 정보 및 작물 성장 정보를 수집 한 다음 SG 통합 기능에 정보를 제공하여 각 작물에 대한 최적의 성장 모델을 생성한다. SG 운영 기능에서 각 FE의 세부 사항은 다음과 같다.

- SGO-EnvInfoProcFE (SGO-Environment Information Processing FE)는 SG Control 기능에서 SGC-EnvInfoProcFE를 통해 SG의 환경 정보를 수집 한 다음 SG 통합 기능의 SGI-EnvInfoProcFE에 환경 정보를 제공한다.
- SGO-GrowthInfoProcFE (SGO-Growth Information Processing FE)는 SGC-GrowthInfoProcFE를 통해 각 SG의 작물 성장 정보를 수집 한 다음이 FE는 SG 통합 기능의 SGI-GrowthInfoProcFE와 SG 작동 기능의 SGO-FinGrowthModelGenFE에 작물 성장 정보를 전달한다.
- SGO-FinGrowthModelGenFE (SGO-Final Growth Model Generation FE)는 SG 통합 기능의 SGI-OptGrowthModelProcFE에서 일반 최적 성장 모델을 수집 한 다음 SG 제어 기능의 SGS-ActContCmdGenFE에 대한 최종 최적 성장 모델을 생성하고, SG 서비스 사용자의 개입을 허용하여 최종 최적 성장 모델을 생성한다. 이 FE를 통해 SG 서비스 사용자가 생성된 최종 최적 성장 모델을 추가로 수정하여 SG 서비스 사용자 맞춤형 최종 최적 성장 모델을 생성할 수 있다.

- SG 통합 기능

SGI-EnvInfoProcFE, SGI-GrowthInfoProcFE 및 SGI-OptGrowthModelProcFE로 구성된 SG 통합 기능은 환경 정보 및 작물 재배 정보를 추적 한 다음 농업 전문가 및 농업 전문가 시스템에 정보를 제공한다. 각 FE의 세부 사항은 다음과 같다.

- SGI-EnvInfoProcFE (SGI-Environment Information Processing FE)는 SGO-EnvInfoProcFE에서 SF로 SG의 환경 정보를 수집하고 필요시 SG 응용 프로그램에 제공한다.
- SGI-GrowthInfoProcFE (SGI-Growth Information Processing FE)는 SGO-GrowthInfoProcFE에서 SF에 대한 작물 성장 정보를 수집하고이 정보는 요청시 SG 응용 프로그램에 제공한다.
- SGI-OptGrowthModelProcFE (SGI- 최적 성장 모델 처리 FE)는 SG 응용 프로그램에서 SF로 경작된 각 작물에 대한 일반적인 최적 성장 모델을 수집 한 다음, 필요에 따라 SGO-FinGrowthModelGenFE에 최적의 성장 모델을 제공한다.

- SG 관리 기능

SGM-RegInfoProcFE 및 SGM-ActOpStatusInfoProcFE로 구성된 SG 관리 기능은 SF에서 SG의 환경 및 작물 성장 정보를 관리하고 유지한다. 각 FE의 세부 내용은 다음과 같다.

- SGM-RegInfoProcFE (SGM 등록 정보 처리 FE)는 IoT 장치 (센서, 액추에이터)의 메타 데이터 정보를 제시한다. 이 FE는 SG 감지 기능 및 SG 작동 기능에서 메타 데이터 정보를 수집하고 관리한다.
- SGM-ActOpStatusInfoProcFE (SGM-Actuator 작동 상태 정보 처리 FE)는 SGA-ActContCmdProcFE에서 액추에이터의 작동 상태 정보를 수집하고 액추에이터의 작동 상태를 모니터링한다. 이 FE는 최종 성장 모델에서 성장 조건을 업데이트하기 위해 수집된 정보를 SGO-FinGrowthModelGenFE에 제공하고 관련 액추에이터에서 적절한 다음 제어 명령을 생성하기 위해 수집된 정보를 SGC-ActContCmdGenFE에 제공한다.

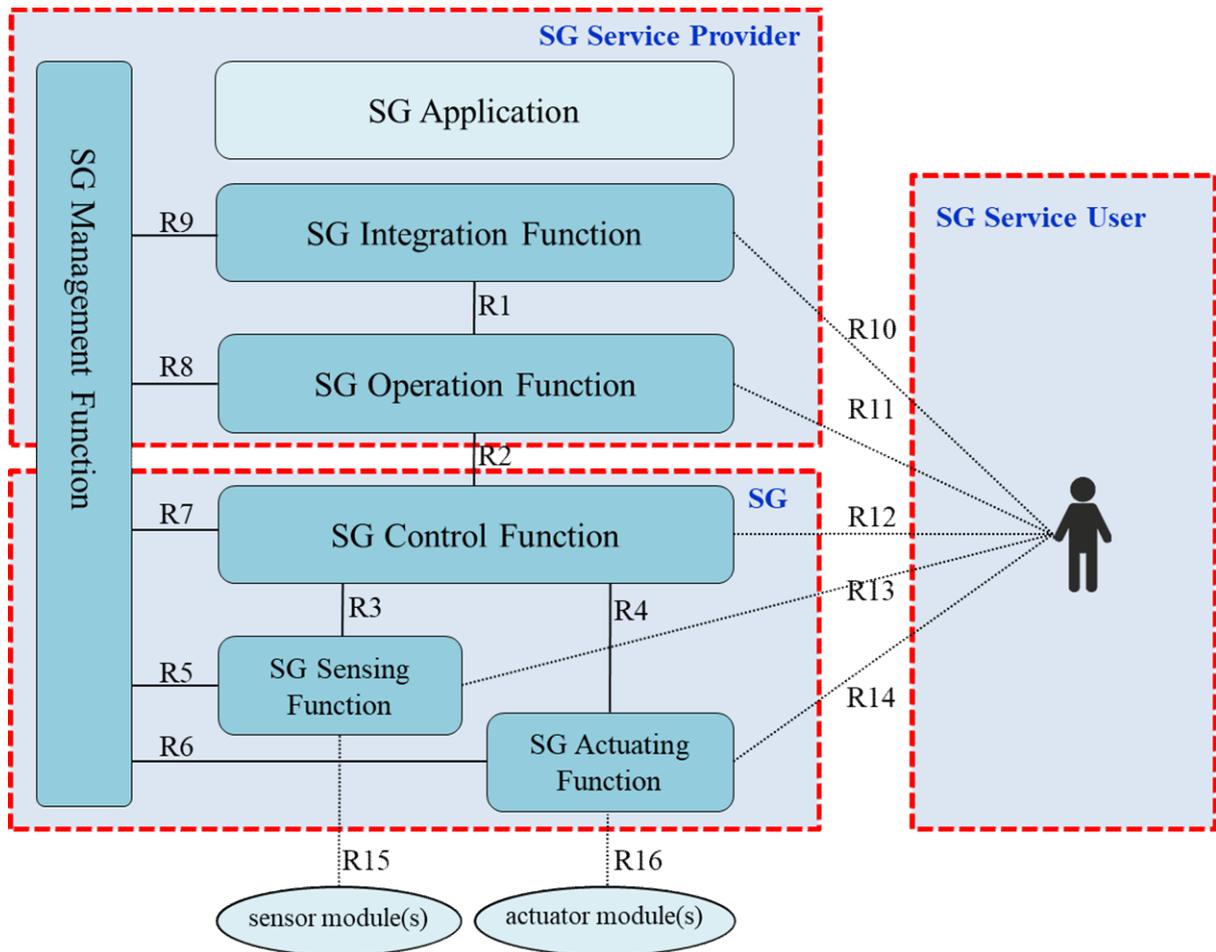


그림 6 - SG 서비스 기능 구조

끝으로 본 표준기술에서는 앞에서 기술하였던 6개의 기능들이 어떠한 관계하에서 그 기능들을 제공할 수 있는지를 정의하기 위해 다음과 같이 R1에서 R16까지 총 16개의 인터페이스를 정의하였다.

각각의 인터페이스들은 다음과 같은 역할을 수행한다.

- 인터페이스 R1은 SG 통합 기능과 SG 작동 기능 사이에서 사용되어 SG의 환경, 작물 성장 및 각 작물의 일반적인 최적 성장 모델 정보에 대한 정보를 제공한다.
- 인터페이스 R2는 SG 작동 기능과 SG 제어 기능 사이에서 사용되어 SG 환경에 대한 정보, 작물의 성장 및 작물의 최종 성장 모델 정보를 제공한다.
- 인터페이스 R3는 SG 제어 기능과 SG 감지 기능 사이에서 사용되어 SG 환경 및 작물 성장에 대한 정보를 제공한다.
- 인터페이스 R4는 SG 제어 기능과 SG 작동 기능 사이에서 사용되어 액추에이터 (들) 및 특정 액추에이터의 상태를 제어하기위한 명령을 전달한다.
- 인터페이스 R5는 SG 관리 기능과 SG 감지 기능 사이에서 사용되어 장치 유형, 제조업체 설명, 장치 위치, 감지 장치의 특성을 포함한 정보로 IoT 감지 장치를 등록한다.
- 인터페이스 R6은 SG 관리 기능과 SG 작동 기능 사이에서 사용되어 IoT 작동 장치를 장치 유형, 제조업체 설명, 장치 위치, 작동 장치 특성 등의 정보로 등록한다.
- 인터페이스 R7은 SG 관리 기능과 SG 제어 기능 사이에서 사용되어 SG에 설치된 IoT 감지 장치 및 IoT 작동 장치의 정보를 제공한다.
- 인터페이스 R8은 SG 관리 기능과 SG 작동 기능 사이에 사용되어 SF에 설치된 IoT 감지 장치 및 IoT 작동 장치의 정보를 제공한다.
- 인터페이스 R9는 SG 관리 기능과 SG 통합 기능 사이에서 사용되어 하나 이상의 SF에 설치된 IoT 감지 장치 및 IoT 작동 장치의 정보를 제공한다.
- 인터페이스 R10은 SG 서비스 사용자와 SG 통합 기능 사이에서 사용되어 SG 환경 및 작물 재배에 대한 정보를 제공한다.
- 인터페이스 R11은 SG 서비스 사용자와 SG 작동 기능 사이에서 사용되어 SG 환경 및 작물 성장에 대한 정보를 제공한다.

- 인터페이스 R12는 SG 서비스 사용자와 SG 제어 기능 사이에서 사용되어 SG 환경 및 작물 성장에 대한 정보를 제공한다.
- 인터페이스 R13은 SG 서비스 사용자와 SG 감지 기능 사이에서 사용되어 감지 데이터를 전달한다.
- 인터페이스 R14는 SG 서비스 사용자와 SG 작동 기능 사이에서 사용되어 액추에이터 상태를 제공한다.
- 인터페이스 R15는 SG 감지 기능과 센서 모듈 사이에서 전류 또는 전압 형태의 감지 값을 판독하는데 사용됩니다.
- 인터페이스 R16은 SG 작동 기능과 액추에이터 모듈 사이에서 사용되어 전류 또는 전압의 형태로 작동 모듈을 제어한다.

## 나-2) IEC 63246-1, Configurable car infotainment services (CCIS) - Part 1: General (IEC 63246-1)

### (가) 제정 경과

IEC (International Electrotechnical Commission)은 전기전자와 관련한 국제표준을 개발하는 UN 산하의 표준화 기구로, ISO 및 ITU-T와 함께 권위있는 국제표준화 기구로 인식되고 있다. 현재 172개국이 IEC에 가입을 하고 있으며, 표준화 전문 분야에 따라, 산하에 총 98개의 TC (Technical Committee)를 두고 있다. 이 중, TC100은 오디오, 비디오 및 멀티미디어 시스템 및 장비 분야의 국제 표준을 개발하고 있다. TC100에서 개발하는 표준에는 주로 멀티미디어 장치에 관한 성능 사양, 소비자 및 전문 장비의 측정 방법, 시스템에서의 적용 및 다른 시스템 또는 장비와의 상호 운용성이 포함된다. TC100에서 간주하는 멀티미디어는 모든 형태의 오디오, 비디오, 그래픽, 데이터 및 통신의 통합이며, 통합에는 이러한 정보의 생산, 저장, 처리, 전송, 표시 및 재생이 포함된다. TC100은 산하에 11개의 TA (Technical Area)를 두고 표준을 개발하고 있다. 이 중 TA17은 차량 운전자 및/또는 승객이 사용하는 오디오, 비디오 및 멀티미디어 시스템 및 장비의 모든 측면을 다루는 국제 표준을 개발하고 있다.

Technical Area	
TA 1	Terminals for audio, video and data services and contents
TA 2	Colour measurement and management
TA 4	Digital system interfaces and protocols
TA 5	Cable networks for television signals, sound signals and interactive services
TA 6	Storage media, storage data structures, storage systems and equipment
TA 10	Multimedia e-publishing and e-book technologies
TA 15	Wireless Power Transfer
TA 16	Active Assisted Living (AAL), wearable electronic devices and technologies, accessibility and user interfaces
TA 17	Multimedia systems and equipment for vehicles
TA 18	Multimedia home systems and applications for end-user networks
TA 19	Environmental and energy aspects for multimedia systems and equipment
TA 20	Analogue and digital audio

그림 1 - IEC TC100 구성도

최근 트랙터, 농업용 드론 등과 같은 농기계류의 발전과 5G 등과 같은 통신 기술의 발달로 인해, 기존 단독형 농기계류가 점차 연결되고(connected) 사용자 편의성을 높이는 UI(User Interface)를 추구하고 있다.

본 표준은 2017년 5월 싱가포르 회의에서 처음으로 개념을 소개한 이후, 농기계에만 국한되지 않고 일반적인 환경(일반용 차량)까지 포함할 것이 요청되어 일반 차량까지 그 범위를 확대하여 2021년 5월에 IEC 국제표준으로 승인되었다.

단계	문서명	제출일
PWI		2018-11-16
PNW	100/3188/NP	2018-11-30
ACD	100/3235/RVN	2019-04-05
CD	100/3266/CD	2019-06-14
ACDV	100/3337/CC	2019-11-08
CCDV	100/3414/CDV	2020-05-15
APUB	100/3538/RVC	2020-11-28

(나) 주요 내용  
1) 시스템 모델

CCIS 서비스 또는 시스템은 다음 그림과 같이 CCIS 마스터의 도움으로 차량 내 다양한 CCIS 장치 및 CCIS 프로필을 쉽게 관리 및 제어할 수 있는 통신 인터페이스를 CCIS 사용자에게 제공한다. CCIS 시스템은 차량 내에서 내장 플랫폼으로 또는 소프트웨어 업그레이드를 통해 제공된다.

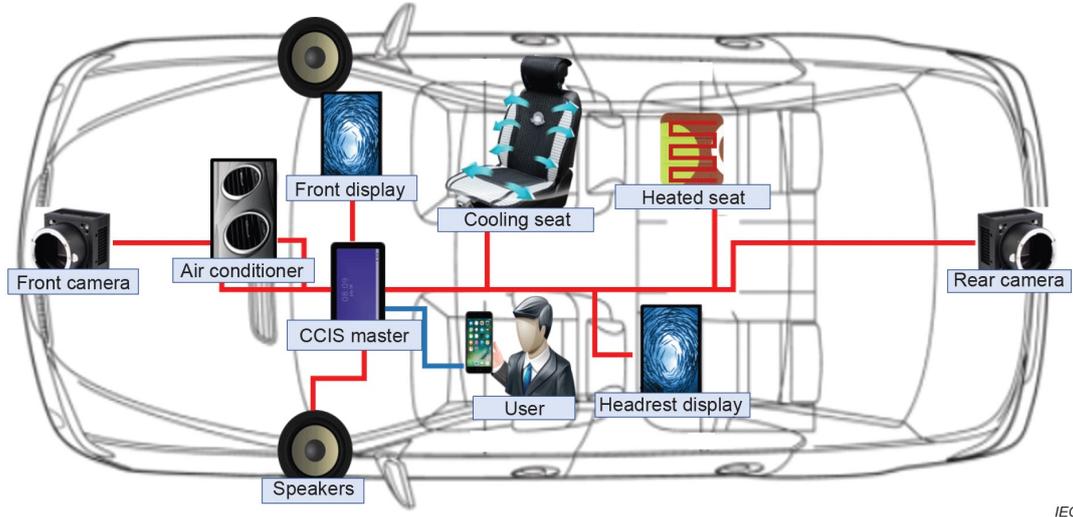


그림 2 - CCIS 환경

다음 그림은 CCIS의 시스템 모델을 보여준다. CCIS 시스템은 CCIS 마스터를 차량 내의 다양한 CCIS 장치에 연결하여 CCIS 장치를 관리하고 제어한다. CCIS 사용자는 CCIS 마스터와의 통신 인터페이스를 통해 CCIS 서비스를 사용할 수 있으며, CCIS 사용자는 CCIS 장치를 제어하거나 장치에 포함된 CCIS 콘텐츠를 사용할 수 있다. 이를 위해 CCIS 마스터는 장치 상태 및 가용성과 같은 CCIS 장치 및 CCIS 프로필을 관리해야 하며 CCIS 사용자는 CCIS 마스터와의 적절한 등록 및 인증 프로세스를 통해 CCIS 장치에 액세스할 수 있다. CCIS 사용자는 자동차 소유자, 임시 소유자, 개인 클라이언트 및 공공 클라이언트로 분류할 수 있다.

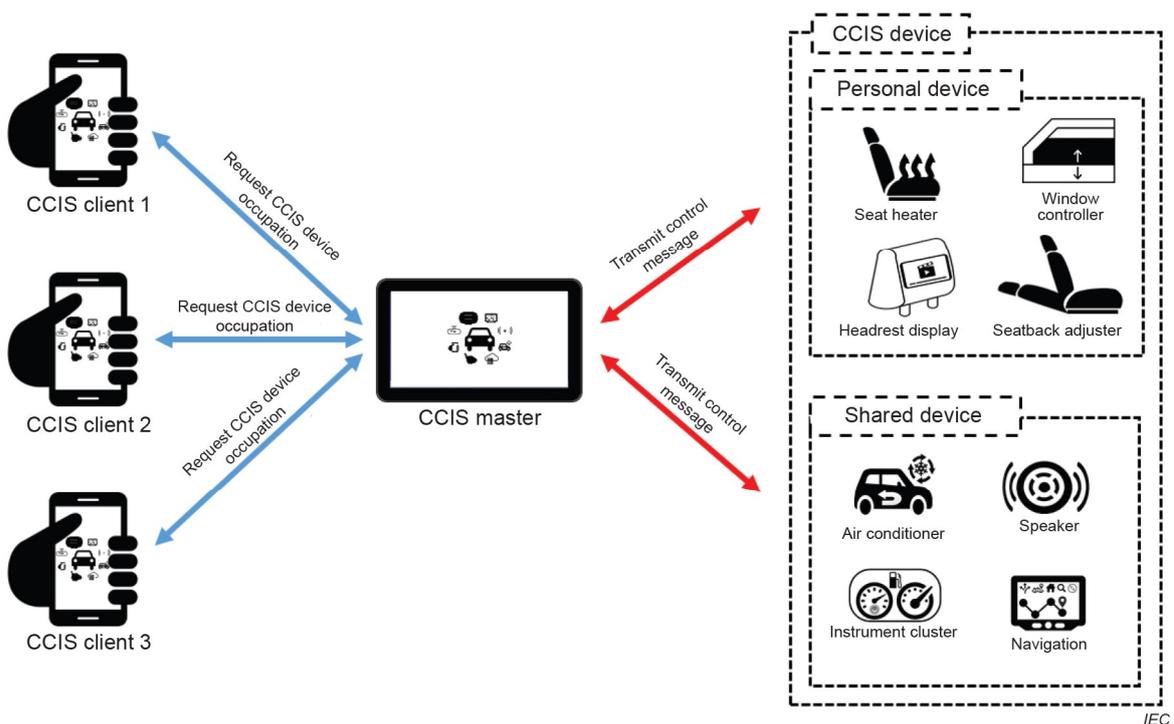


그림 3 - CCIS 시스템 모델

2) CCIS 사용자 및 서비스 플로우

2.1) CCIS 사용자 유형

차량에는 다양한 스마트 기기와 CCIS 프로파일들이 존재할 수 있다. CCIS 사용자 유형에 따라 일부 장치는 허용될 수 있지만, 나머지 장치는 CCIS 서비스로 사용이 제한될 수 있다. 이와 관련하여 CCIS 사용자의 유형은 다음 표와 같이 4가지 경우로 분류된다.

표 1 - CCIS 사용자 유형

분류	장기 사용	단기 사용
소유주 (인증 불필요)	차량 소유주	임시 소유주
클라이언트 (인증 필요)	개인 클라이언트	공공 클라이언트

위 표에서와 같이 CCIS 사용자는 소유 및 사용 기간에 따라 자동차 소유자, 임시 소유자, 개인 고객, 공공 고객으로 분류된다.

- 자동차 소유자: CCIS 마스터와 함께 자동차를 소유하고 CCIS 기능 및 서비스에 대한 전반적인 권한을 가진 사용자이다. 자동차의 소유권은 장기이다.
- 임시 소유자: 자동차 소유자로부터 일시적으로 소유권을 넘겨 받은 사용자(예: 카셰어링 또는 렌탈 서비스 사용자). 임시 소유자에게 부여되는 CCIS 서비스 사용에 대한 특정 수준의 권한은 자동차 소유자가 미리 지정할 수 있다.
- 개인 고객: 소유권 없이 장기적으로 CCIS 서비스를 이용할 수 있는 사용자(예: 자동차 소유자의 가족). 개인 고객에게 제공되는 CCIS 서비스 사용에 대한 특정 수준의 권한은 자동차 소유자가 미리 지정할 수 있다.
- 공용 클라이언트: 아직 CCIS 마스터에 의해 인증되지 않은 사용자(예: 게스트). 공개 클라이언트는 CCIS 마스터를 통해 등록 및 인증 절차를 수행해야 한다.

CCIS 장치는 다음과 같이 기능에 따라 공유되거나 공유되지 않을 수 있다.

- CCIS 기기 공유 허용(공유 기기): 차량에서 한 명 이상의 사용자가 공유하는 기기(예: 에어컨 및 스피커)
- CCIS 기기 공유 불가(개인 기기): 특정 사용자 유형 전용 기기(예: 헤드레스트 디스플레이 및 열선 시트)

2.2) 차량 소유주에 대한 CCIS 시나리오 및 서비스 흐름

다음 그림은 자동차 소유자가 자신의 CCIS 장치와 CCIS 콘텐츠를 사용하는 서비스 모델을 보여준다.

앨리스는 회사의 영업 부서에서 일하고 있다. 그녀의 직업 특성상 그녀는 차 안에서 많은 시간을 보낸다. 따라서 그녀는 도로 차량 사고에 대한 이벤트 비디오 데이터 레코더(EVDR, IEC 63005 시리즈 참조), 스마트 미러, 카메라, 헤드레스트 디스플레이 등과 같은 다양한 장치를 차에 장착했으며, 또한 그녀는 이러한 장치들을 효율적으로 관리하기 위해 CCIS 마스터를 설치했다.

어느 날 앨리스는 의뢰인과 계약을 하기로 약속을 잡았다. 그녀는 회의를 위해 차에 오른다. 차에 타자마자 스마트폰이 자동으로 CCIS 마스터에 연결되고, 사용 가능한 CCIS 기기 목록과 CCIS 마스터에게 CCIS 콘텐츠 목록을 묻는다. 앨리스는 스마트폰에 저장된 자신이 좋아하는 음악 파일을 재생하고 싶기 때문에, CCIS 마스터는 사용 가능한 CCIS 장치의 사용 가능한 리소스 목록과 CCIS 콘텐츠를 앨리스의 스마트폰으로 보낸다. 앨리스가 운전하는 동안 음악 파일을 재생한다.

앨리스가 미팅 장소에 도착한 후, CCIS 마스터와 연결을 끊고 회의실로 향했다.

이 서비스 모델에는 보안 문제가 있을 수 있다. 등록 및 권한이 부여되지 않은 외부 사용자는 CCIS 마

스터에 접근을 시도할 수 있으며 외부 사용자는 차량 소유자의 권한으로 CCIS 장치를 추가하거나 제거할 수 있다. 따라서 CCIS 마스터에 등록 및 승인이 자동차 소유자에게 제공되어야 한다.

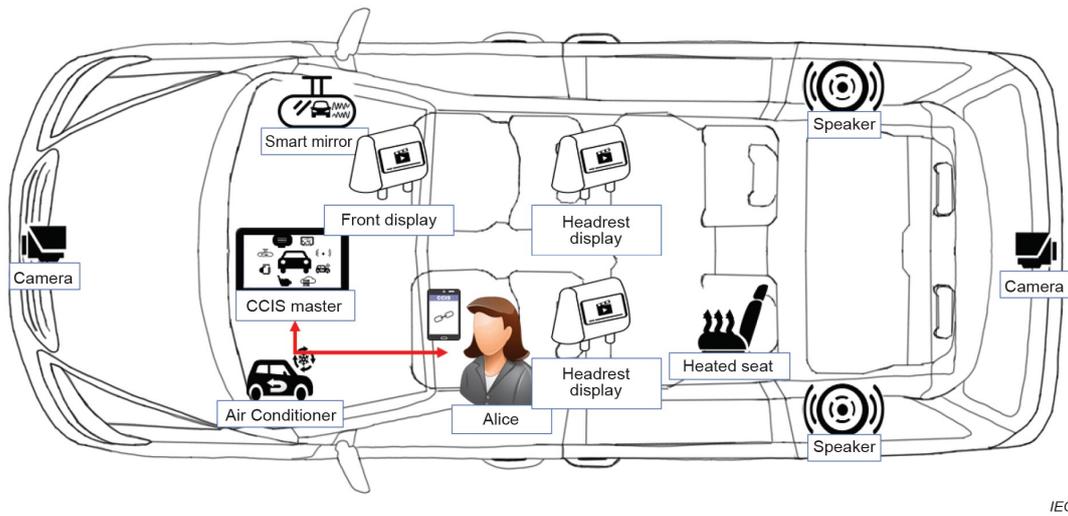


그림 4 - 차량 소유주를 위한 CCIS 모델

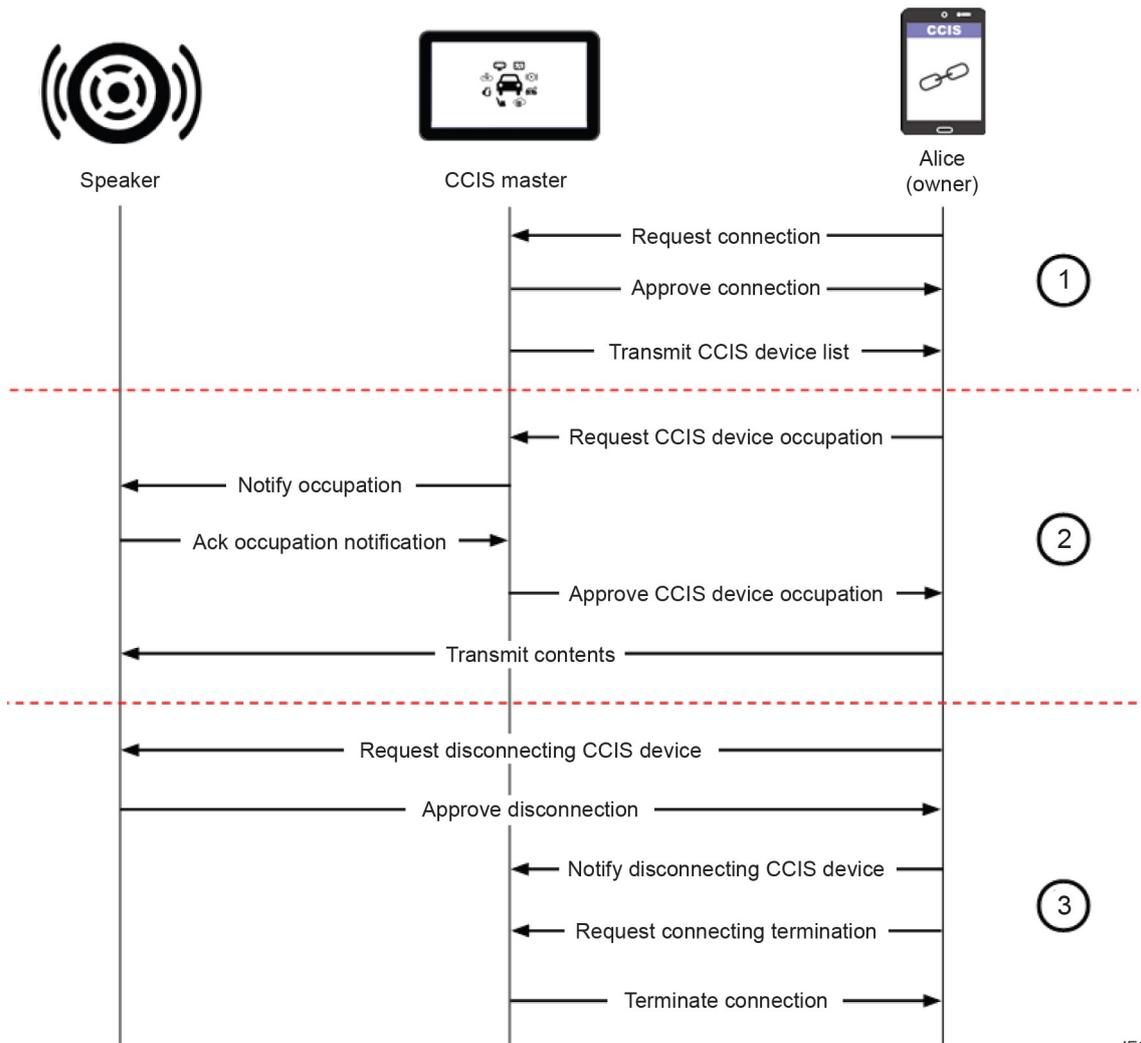


그림 5 - 차량 소유주 관련 서비스 흐름

상기 그림은 위의 서비스에 대한 서비스 흐름을 보여준다.

- a) 앨리스가 차량에 있을 때 그녀의 스마트 장치는 CCIS 마스터에 연결되고 CCIS 장치 목록을 검색한다.
- b) 앨리스는 CCIS 장치와 CCIS 콘텐츠 목록에서 스피커를 선택하고 CCIS 마스터에게 점유 요청 메시지를 보낸다. 요청 시 CCIS 마스터는 스피커에게 자신의 자원이 앨리스에게 점유되었음을 알리고, 스피커는 점유에 대한 응답으로 연결이 준비되었음을 알려준다. CCIS 마스터가 앨리스에게 장치 점유 승인을 보낸 후, 앨리스는 스피커를 통해 스마트 기기에 저장된 음악 파일을 재생한다.
- c) 앨리스는 스피커 점유를 해제하고 CCIS 마스터와의 연결을 종료한다. 그러면 CCIS 마스터는 스피커의 상태를 사용 가능으로 변경한다.

### 2.3) 임시 소유주에 대한 CCIS 시나리오 및 서비스 흐름

다음 그림은 임시 소유주가 CCIS 장치 스피커를 사용하는 서비스 모델을 보여준다.

휴가로 여행을 떠나는 밥은 편안하게 여행할 수 있도록 차를 빌린다. 차를 선택한 밥은 임시 소유자로 등록 절차를 거쳐야 차를 빌릴 수 있다. 자동차 소유자의 권한을 가진 렌트카 서비스 관리자가 CCIS 마스터의 모드를 임시 소유자 모드로 변경한다. 모드를 변경한 후 밥은 임시 소유자 등록 요청을 CCIS 마스터에게 보낸다. 그 후 서비스 관리자는 기간을 설정하고 CCIS 장치 및 CCIS 콘텐츠에 선택적으로 액세스할 수 있으며 요청을 승인한다. 승인된 CCIS 마스터는 밥을 임시 소유자로 등록한다.

임시 소유주의 경우 추가 절차 및 인증이 필요하다. 예를 들어, 임시 소유주가 소유주 권한을 가질 수 있는 기간을 정의하고 액세스할 수 있는 CCIS 장치 및 CCIS 콘텐츠를 분류해야 한다.

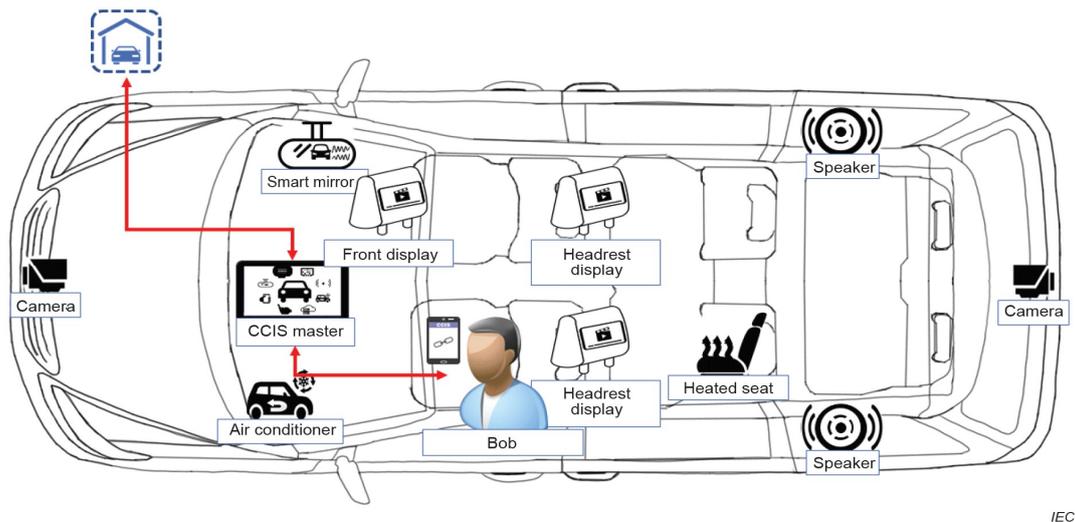


그림 6 - 차량 임시 소유주를 위한 CCIS 모델

다음 그림은 위의 서비스에 대한 서비스 흐름을 보여준다.

첫 번째 단계는 렌트카 회사(관리자)의 CCIS 마스터 사용자 모드 구성을 보여준다. CCIS 마스터가 제공하는 CCIS 장치 및 CCIS 콘텐츠 기능은 해당 결제에 따라 달라질 수 있다. 먼저 관리자는 CCIS 마스터에게 사용자 모드를 수정하도록 요청한다. 그런 다음 CCIS 마스터는 사용 가능한 CCIS 장치 및 CCIS 콘텐츠 목록을 관리자에게 제공한다. 관리자는 액세스 가능한 CCIS 장치 액세스 목록을 CCIS 마스터에 할당한다. 할당된 리소스 액세스 목록을 사용하여 CCIS 마스터는 사용자 모드를 변경할 수 있다. CCIS 마스터가 사용자 모드를 성공적으로 변경하면 CCIS 마스터가 사용자 모드가 변경되었음을 확인한다.

두 번째 단계는 밥이 임대 기간 동안 CCIS 마스터의 적절한 사용자가 되는 방법을 보여준다. 밥은 CCIS 마스터에 연결을 시도한다. 밥은 등록된 사용자가 아니기 때문에 CCIS 마스터는 밥의 참여를 렌터카 회사에 문의한다. 렌터카 회사가 밥의 참여를 승인하면 CCIS 마스터는 밥에게 이제 임시 소유자의 권리가 있음을 알린다.

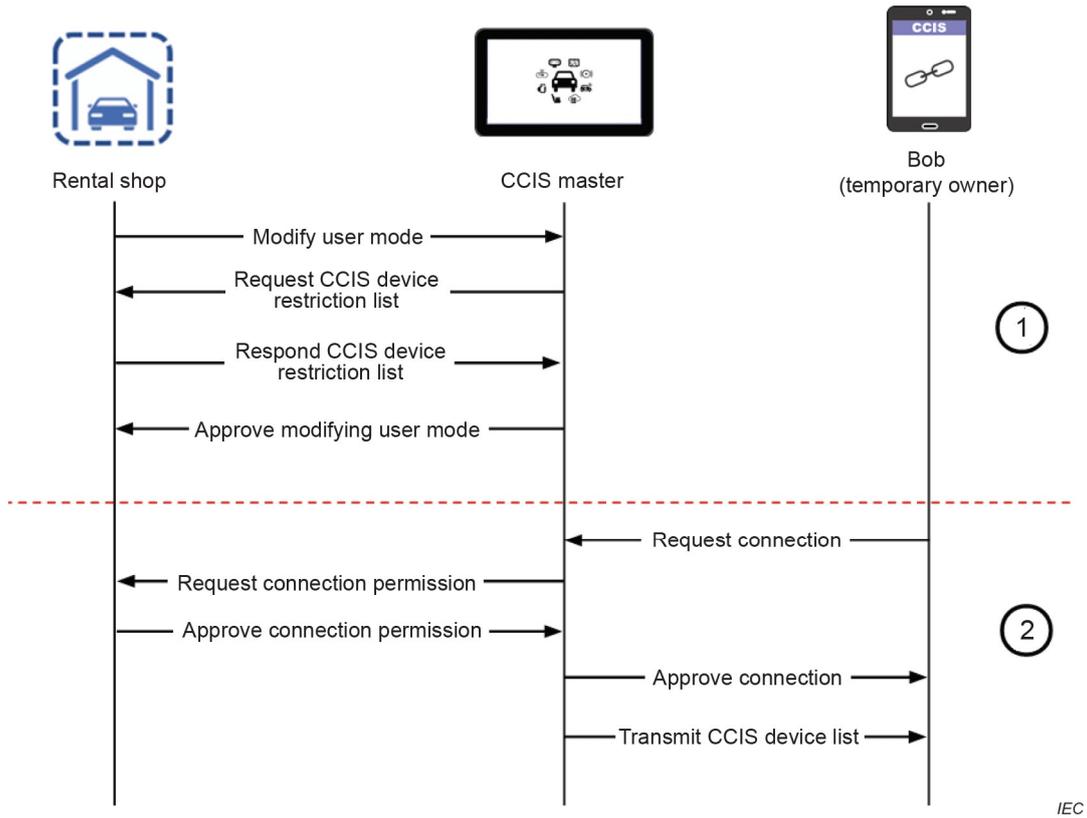


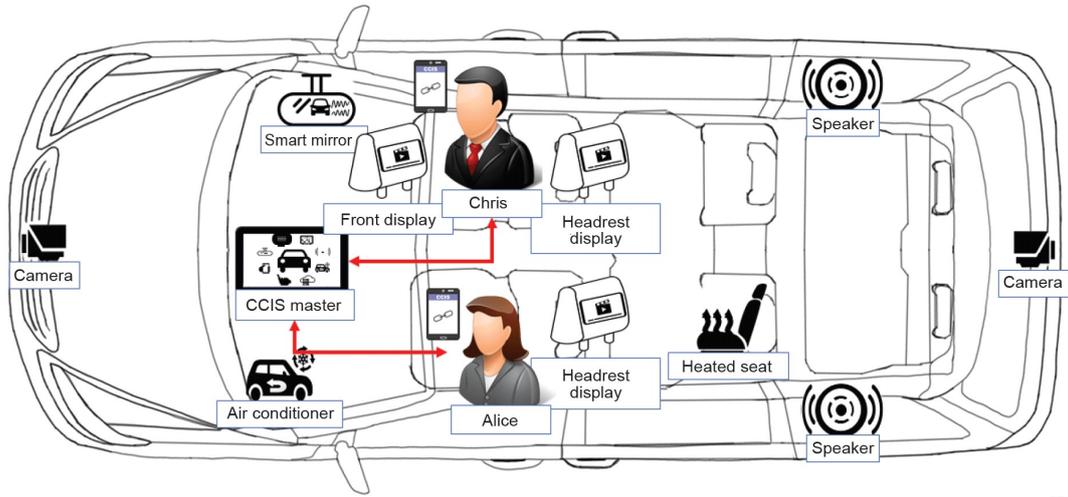
그림 7 - 차량 임시 소유주 관련 서비스 흐름

#### 2.4) 개인 클라이언트에 대한 CCIS 시나리오 및 서비스 흐름

다음 그림은 사용자가 CCIS 장치를 사용하기 위해 개인 클라이언트로 CCIS 마스터를 등록한 서비스 모델을 보여준다.

앨리스는 주말에 동생 크리스와 함께 영화관에 가기 위해 차에 탔다. 크리스는 앨리스의 좌석 앞에 있는 머리 받침 디스플레이를 사용하여 그가 보고 싶은 영화 목록을 보여주려고 한다. 이미 개인 클라이언트로 등록된 크리스의 스마트폰은 CCIS 마스터로부터 자동으로 CCIS 장치 목록을 수신하고 크리스는 그녀의 좌석 앞 헤드레스트 디스플레이를 선택한다. 영화 선택이 끝나면 크리스는 스피커를 사용하여 스마트폰에 저장된 음악을 재생하려고 한다. 크리스는 CCIS 마스터에게 스피커 사용을 요청하고, CCIS 마스터는 앨리스에게 공유 CCIS 장치인 스피커 사용 허가를 요청한다. 앨리스의 허락을 받은 크리스는 스마트폰에 저장된 음악을 스피커를 통해 재생한다.

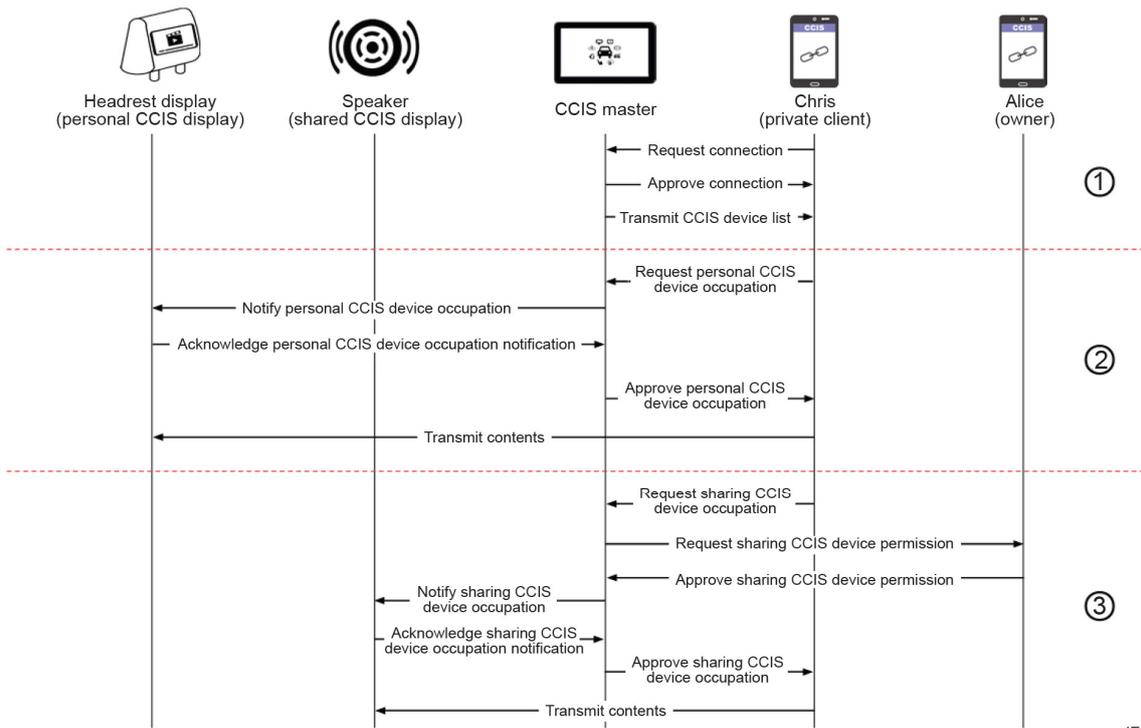
이 서비스 모델에는 보안 문제가 있을 수 있다. 공인 클라이언트 또는 외부 사용자와 같은 권한이 없는 클라이언트는 CCIS 장치에 액세스하기 위해 개인 클라이언트의 권한을 시도할 수 있기 때문에 개인 클라이언트에 대한 해당 등록 및 인증 절차가 제공되어야 한다.



IEC

그림 8 - 개인 클라이언트를 위한 CCIS 모델

다음 그림은 개인 클라이언트에 대한 서비스 예와 관련된 서비스 흐름을 보여준다.



IEC

그림 9 - 개인 클라이언트 관련 서비스 흐름

첫 번째 단계는 개인 클라이언트의 부착 절차를 보여준다. 크리스가 앨리스의 차에 탈 때 크리스의 스마트 장치는 CCIS 마스터에 연결되고 CCIS 마스터는 크리스의 스마트 장치에 CCIS 장치 목록을 제공한다.

두 번째 단계는 특정 클라이언트 전용 개인 장치를 부착하는 절차를 보여준다. 크리스는 장치 목록에 포함되어 있으며 허가 없이 액세스할 수 없는 전면 디스플레이를 선택한다. 선택한 장치를 점유하기 위해 크리스의 장치는 크리스를 위해 선택된 전면 디스플레이를 점유할 수 있는 CCIS 마스터 권한을 요구한다. 이 요구에 따라 CCIS 마스터는 전면 디스플레이에 크리스가 차지할 것임을 알린다. 전면 디스플레이는 응답으로 점유를 확인하고 크리스에게 연결할 수 있다. 크리스의 요청에 대한 CCIS 마스터의 승인 후 크리스의 스마트 장치와 전면 디스플레이는 다양한 유형의 데이터 콘텐츠를 교환하기 위한 연결을 설정한다.

세 번째 단계는 하나 이상의 클라이언트가 공유할 수 있는 공유 장치의 연결 절차를 보여준다. 크리스는 CCIS 마스터에게 공유 장치인 스피커 점유 권한을 요청한다. 요청을 받은 CCIS 마스터는 앨리스(스피커

소유자)에게 요청한 스피커를 크리스와 공유하도록 요청한다. 앨리스의 승인을 받아 CCIS 마스터는 스피커에게 크리스가 점유하고 있음을 알린다. 스피커는 응답으로 점유를 확인하고 크리스와 연결할 준비가 된다.

## 2.5) 공공 클라이언트에 대한 CCIS 시나리오 및 서비스 흐름

다음 그림은 공용 클라이언트가 CCIS 장치를 사용하는 서비스 모델을 보여준다.

앨리스는 고객 데이비드를 데리러 그의 회사로 향하고 있다. 데이비드는 CCIS 마스터에게 연결을 요청했고 앨리스는 이를 승인한다. 데이비드는 차량 내부의 온도를 제어하기 위해 에어컨을 점유해 달라는 메시지를 보낸다. CCIS 마스터는 앨리스에게 공용 장치 권한의 승인을 요청하고 앨리스는 이를 승인한다. 그 후 데이비드는 회의와 관련된 자료를 보기 위해 CCIS 마스터에게 헤드레스트 디스플레이 점유 메시지를 요청한다. 헤드레스트 디스플레이를 점유한 데이비드는 헤드레스트 디스플레이를 통해 스마트폰에 저장된 자료를 본다.

보안을 위해 공용 클라이언트는 CCIS 장치를 사용하기 위해 자동차 소유자의 승인을 받아야 한다.

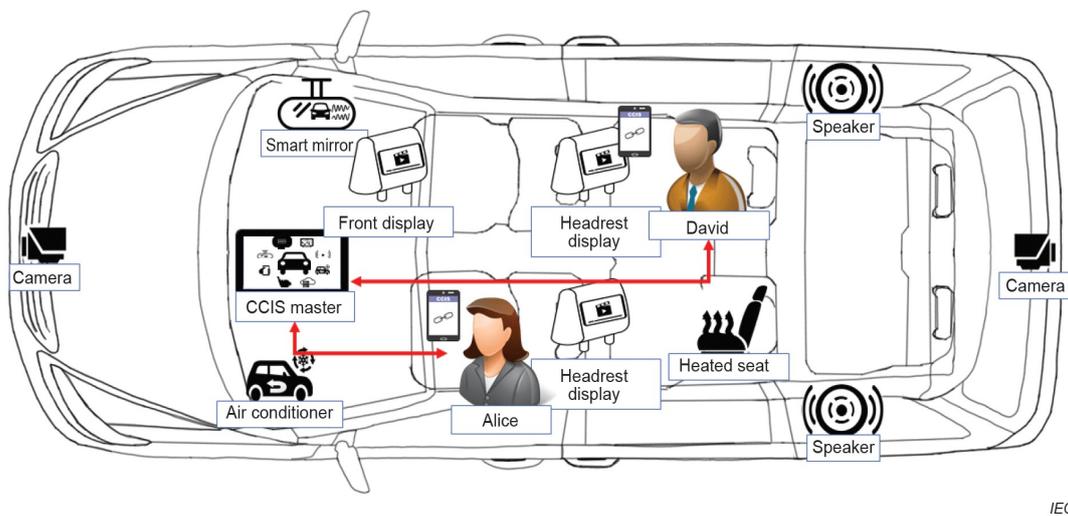


그림 10 - 공공 클라이언트를 위한 CCIS 모델

다음 그림은 위 서비스 예와 관련된 서비스 흐름을 보여준다.

첫 번째 단계는 클라이언트 등록 절차를 보여준다. 등록된 클라이언트가 아닌 데이비드는 앨리스의 CCIS 장치를 사용하기 위해 CCIS 마스터 권한을 요구한다. CCIS 마스터는 데이비드가 등록된 클라이언트가 아니라는 것을 알게 되므로 데이비드에 대한 앨리스의 액세스 권한이 필요하다. 이 경우 앨리스는 데이비드에게 다음 두 가지 액세스 권한 중 하나를 부여할 수 있다.

- 클라이언트가 개인 클라이언트가 되고 CCIS 마스터에 등록되는 개인 클라이언트 권한. 그러면 개인 클라이언트는 별도의 인증 절차 없이 CCIS 마스터에 연결할 수 있다.
- 클라이언트가 공개 클라이언트가 되고 CCIS 마스터로부터 임시 키를 받는 공개 사용자 권한. 공개 사용자는 일정 기간 개인 클라이언트와 동일한 장치 액세스 권한을 갖는다.

앨리스가 데이비드에게 적절한 액세스 권한을 부여하면 CCIS 마스터는 앨리스가 소유한 장치 목록을 제공하여 데이비드의 요구를 확인한다.

두 번째 단계는 현재 다른 사람이 소유한 장치를 사용하는 절차를 보여준다. 데이비드가 현재 앨리스가 소유하고 있는 에어컨을 사용하려고 하자 데이비드는 CCIS 마스터에게 에어컨 사용 권한을 요청한다. 데이비드의 요청을 받은 CCIS 마스터는 앨리스에게 데이비드와 장치(에어컨)를 공유할 수 있는 권한을 요청한다. 앨리스의 허락하에 CCIS 마스터는 에어컨에 데이비드가 차지할 것이라고 알린다. 이에 대한 응답으로 에어컨은 확인 후 데이비드가 연결할 준비를 한다.

세 번째 단계는 개인 기기 사용 절차를 보여준다. 데이비드가 기기 목록에 개인 기기로 등록된 헤드레스트 디스플레이를 사용하려고 할 때, 데이비드는 CCIS 마스터에게 헤드레스트 디스플레이에 대한 액세스

권한을 요청한다. 요청을 받으면 CCIS 마스터는 데이비드가 사용하기를 원한다고 헤드레스트 디스플레이에 알린다. 데이비드의 액세스가 허용되면 헤드레스트 디스플레이는 데이비드의 액세스 권한이 확인되었으며 데이비드가 연결할 준비가 되었음을 CCIS 마스터에게 알린다. CCIS 마스터는 데이비드에게 헤드레스트 디스플레이를 사용할 수 있음을 알린다.

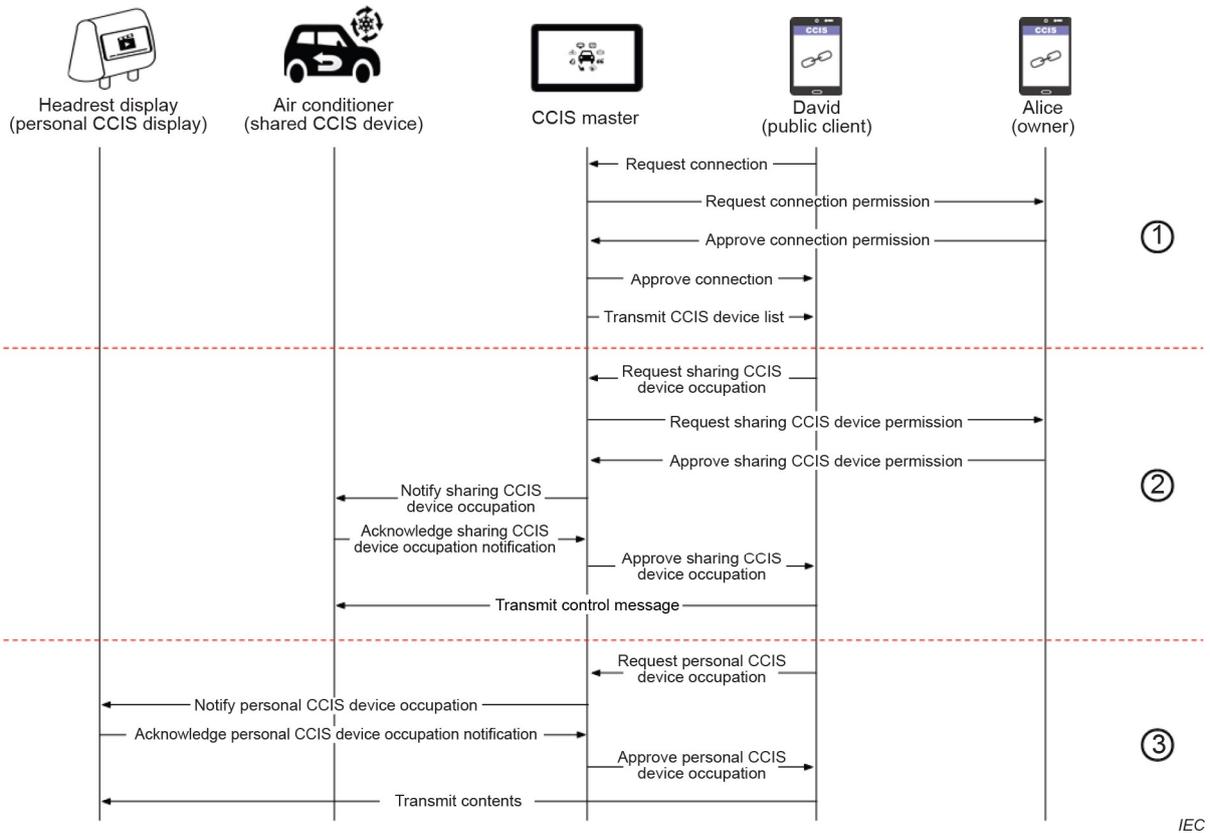


그림 11 - 공공 클라이언트 관련 서비스 흐름

### 3) 보안

CCIS에 대한 몇 가지 보안 고려 사항을 고려할 수 있다. 차량 외부의 인증되지 않은 사용자는 그림 11과 같이 CCIS 마스터에 연결을 시도할 수 있다. 예를 들어 인증되지 않은 외부 사용자가 CCIS 마스터에 액세스하여 스피커의 소리를 높이거나 임의로 경로를 변경할 수 있는데, 이는 자동차 사고로 이어질 수 있다.

이러한 비정상적인 접근을 방지하기 위해 CCIS 마스터는 자동차 소유자가 CCIS 마스터에 등록할 때 자동차 소유자의 장치 ID(예: MAC 주소)를 저장해야 한다. 또한, CCIS 마스터는 자동차 소유자가 변경될 때 기존 장치 ID를 제거하여 하나의 자동차 소유자 장치만 추적할 수 있다. 이것은 임시 소유자에게는 적용되지 않는다. 적절한 보안 솔루션을 통해 외부 클라이언트에 대한 이러한 무단 액세스를 방지하기 위해 CCIS가 제공되어야 한다.

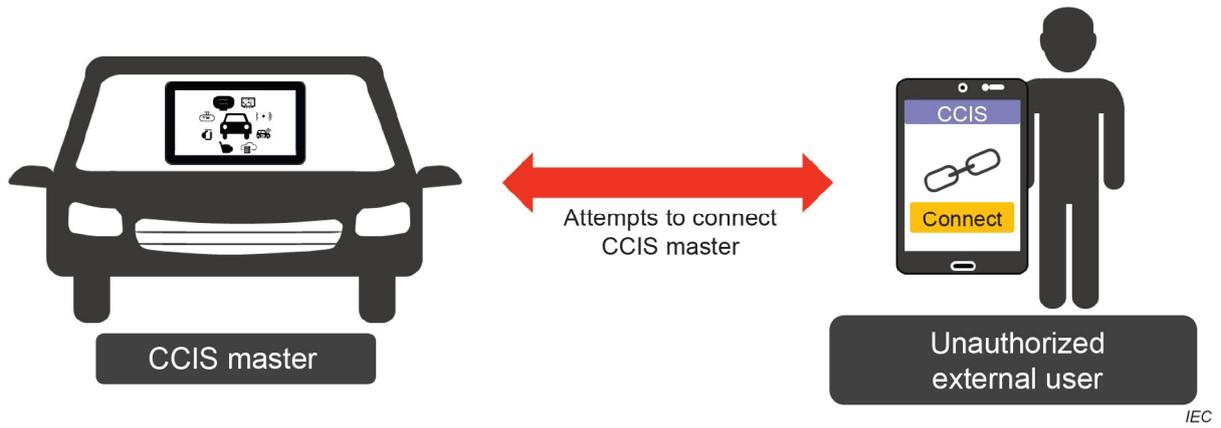


그림 12 - 비인가된 외부 사용자로부터의 비정상적인 접근

다음 그림은 공개 클라이언트가 CCIS 마스터가 지정한 시스템 설정을 조정하려고 시도하는 무단 클라이언트에 의한 보안 공격의 또 다른 예를 보여준다. 각 클라이언트는 권한 수준에 따라 CCIS 장치를 제어할 수 있어야 한다. 고객은 자동차 소유자의 고유한 기능을 수행해서는 안 된다.

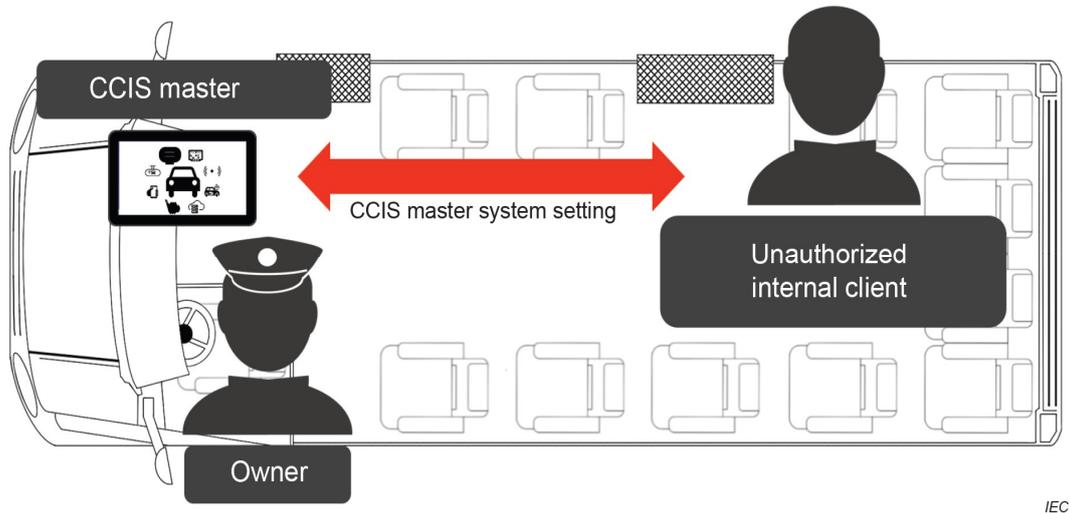


그림 13 - 내부 사용자로부터의 비인가된 제어 시도

CCIS 서비스는 보안을 강화하기 위하여 CCIS 장치와 각 클라이언트에 대한 권한 수준을 정의해야 한다. 각 클라이언트는 권한 수준에 따라 장치에만 액세스할 수 있다. 클라이언트가 자신의 허가 수준에 맞지 않는 장치에 접근을 시도하는 경우, CCIS 마스터는 허가 수준을 확인하고 필요한 경우 자동차 소유자의 동의를 따라 접근을 허용할 수 있다.

나-3) ITU-T Y.4482

(가) 제정 경과

표 1 - ITU-T Y.4482 표준 제정 경과

순번	기고서 제목	제출일
1	Y.IoT-SLF; Proposed modification of scope and structure	2020-04-02
2	Y.IoT-SLF: Proposed modification on SLF model and associated community view in clause 7	2020-07-16
3	Y.IoT-SLF: Proposed requirements and security consideration of SLF service	2020-07-16
4	Y.IoT-SLF: Proposed modifications on Clause 7, "SLF model"	2021-02-22
5	Y.IoT-SLF - Proposed modifications on clause 8	2021-06-28
6	Y.IoT-SLF - Proposed modifications on clause 8 and 9	2021-10-11
7	Contribution on ITU-T Y.SLF "Framework and capabilities for smart livestock farming based on Internet of things" based on the revision of all text (yellow marks)	2021-12-14
8	Proposed modification on draft Recommendation ITU-T Y.IoT-SLF, "Requirements and framework for smart livestock farming based on the Internet of things"	2022-04-07
9	Proposed modification on draft Recommendation ITU-T Y.IoT-SLF, "Requirements and framework for smart livestock farming based on the Internet of things" for consent	2022-07-18

(나) 주요 내용

가축 상품은 복잡한 이해관계자와 물류를 포함하여 국내 또는 국제적인 시장에서 판매될 수 있어야 한다. 다음 그림은 스마트 축산 농장의 3개층 개념 모델을 보이는 것으로 최상위 레벨의 커뮤니티 뷰는 공급자와 농가, 가공업체, 도·소매업자 및 소비자 등과 같은 주요 이해관계자를 대표하는 것으로서 이들 이해관계자의 주요 활동과 의무를 설명한다. 이들은 지역이나 국가, 국제적인 법률과 요구조건을 적용받는 ICT 변화에 동시에 대응할 수 있어야 한다. 또한 식품의 품질 표준과 지속가능성을 준수하여 차별화된 시장을 확보하고 제품의 가치를 높일 수 있다. 중간계층에는 커뮤니티의 운영을 지원하기 위해 제공되는 서비스에 대한 것을 다룬다. 각 서비스 클래스는 하나 이상의 이해관계자를 위한 결과를 달성하기 위해 연결된 서로 다른 프로세스를 갖는 하위 서비스가 있다. 마지막으로 최하위 계층에서는 IoT 엔진의 기능과 축산의 가치사슬을 따라 감시와 제어 목표를 달성할 수 있는 기술적인 장비와 인프라에 대한 목록을 제공한다.

1) 커뮤니티 뷰

개념 모델의 첫 번째 레벨은 축산 가치 사슬로 대표되는 주요 이해관계자를 설명하는 것으로 각 이해관계자에 대한 세부 내용은 다음과 같다.

공급자는 도구나 장비, 생산품, 제품, 원자재, 물류 및 유형서비스를 제공할 책임을 가지며 공급자의 예는 다음과 같다.

- 사료, 의약품, 비료, 제초제, 살충제, 종자, 물, 기계 및 장비, 작업 및 전기를 공급하는 투입물 공급업체
- 공정을 분석하고 농가 또는 생산자에게 필요한 지식과 전문 서비스를 제공하는 등 컨설턴트 역할을 하는 전문가로서 가축의 건강, 영양, 사육 및 농업과 관련된 수의사, 동물원, 공학자, 인공지능 기술자, 농업경제학자 등
- 금융 공급자는 대출이나 보험, 회계, 재무계획, 금융거래 등과 같이 자금을 관리하는 광범위한 비즈니스를 포함하는 금융 산업 관련 분야

농부는 생산단위 및 시설과 관련된 모든 측면을 포함하여 생산단계에 대한 책임을 갖는다. 대부분의 시간을 생산과정 전반에 걸쳐 동물 또는 동물과 관련된 문제를 다루는 이해관계자에 해당하며, 낙농업, 육류 생산업, 수산양식업 또는 알생산업 등이 있다.

가공업자는 생산된 제품을 처리하는 것으로서 육류 생산과 관련해서는 일반적으로 도축장으로 대표되지만 다른 측면에서 이는 식품과 관련된 산업이 될 수 있다. 농산물이 언제, 어떻게 생산되었는지에 대한 정보와 가공 단계에 대한 종합적인 정보를 받을 수 있기 때문에 최종적인 제품의 추적을 위한 중요한 역할을 한다. 농가의 생산품을 받아 최종 제품으로 변환하는 것으로서 포장 및 보관 역시 여기에 해당한다. 또한 살아있는 가축이나 완제품의 운송과 생산 이력 및 추적·플 위한 원산지, 투입물, 품질 및 제품의 목적지에 대한 기록과 정보를 관리하여 생산에서 최종 소비 또는 폐기까지의 유통과정을 추적할 수 있어야 한다.

유통/판매업자는 시장에 공급하기 위한 제품의 유통과 판매에 대한 책임을 갖는 이해관계자로서 소비자나 서비스 판매자, 농산물의 가공에서 유통까지 완제품의 다단계 추적을 담당하는 물류 추적 에이전트 등이 있다.

소비자는 가치사슬의 최종 단계로서 제품의 품질과 추적 가능성, 가격 등과 같은 문제에 가장 많은 관심을 갖는 이해관계자이다.

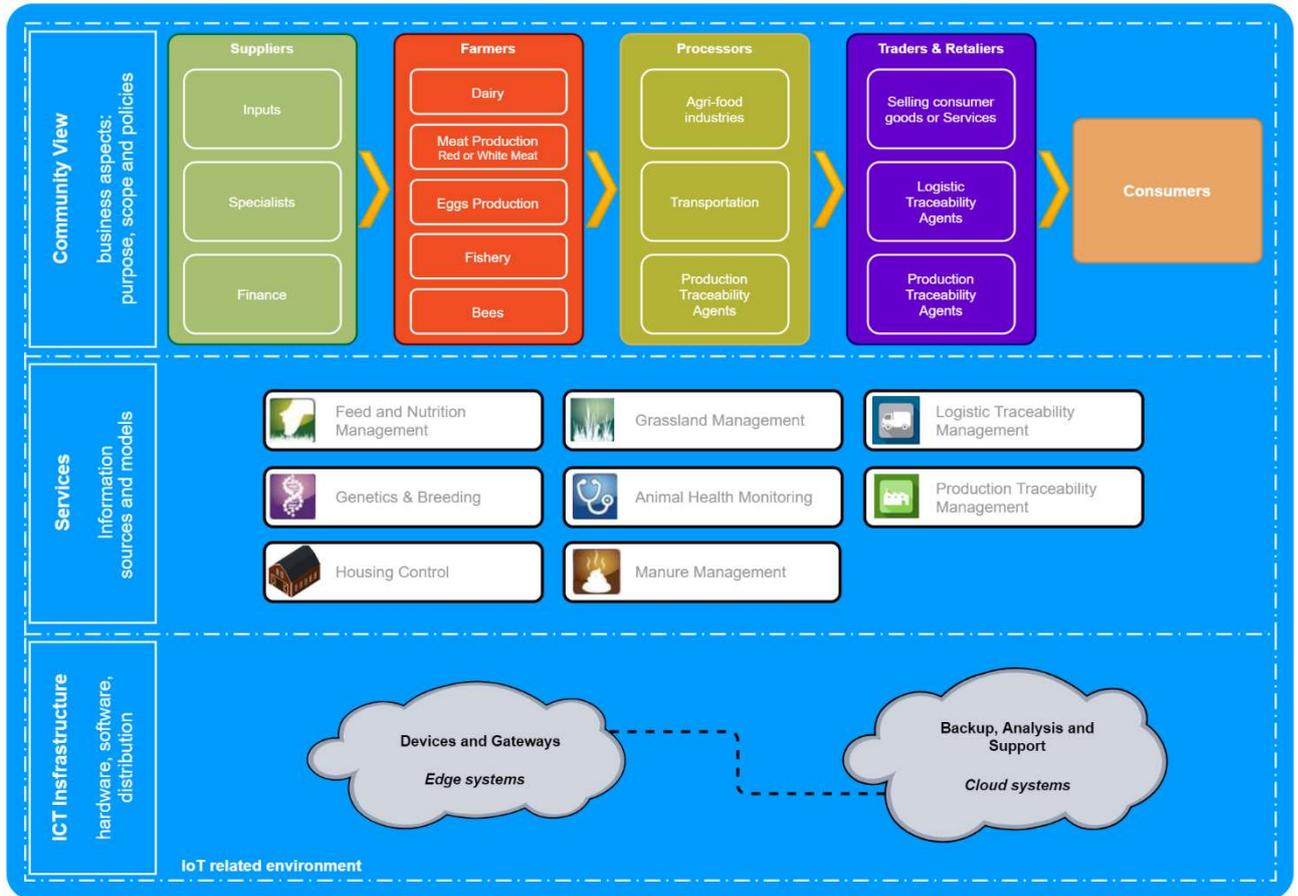


그림 1 - SLF를 위한 3단계 개념 모델

## 2) SLF 서비스

### 2.1) 사료 및 영양 관리

생산 단계 중에 사육 단계는 사료 전환율과 에너지 소비를 기반으로 한 동출의 최적 체중 증가를 포함한다. 이를 위해 구축된 IoT 장치 또는 시스템의 통합은 개별 가축의 ID, 시간 경과에 따른 체중 및 사료 소비(사료 전환율 확보를 위한), 물 섭취 및 이동(에너지 소비 추정을 위한)과 같은 매개변수를 연결할 수 있어야 한다. 전체 가치사슬에 따른 주요 정보 중에 하나는 가축의 식별에 있다. 농장에서 사육되는 가축의 등록이나 가축에서 파생된 모든 제품에 부착된 상태로 유지되기 때문이다. 적절한 장치나 안테나로 가축을 쉽고 즉각적으로 식별할 수 있는 모든 종류의 전자 식별 방식(내부, 외부 또는 피하)이 사용될 수 있다.

### 2.2) 유전과 번식

가축의 유전과 번식은 유전적 특성의 최대화를 위한 과학기술에 해당한다. 따라서 가축의 유전과 번식 서비스는 농장의 동물이 세대를 거듭하여 지속적으로 개선될 수 있도록 하는 것이 중요하다. 또한 가축 생산의 집종도, 제어 가능성 및 균일성을 높이는 동시에 부정적인 환경 영향을 줄이는 역할을 수행한다. 이를 위해 다양한 IoT 센서를 사용하여 가축 및 가축 성장과 건강 정보를 식별하고 이러한 정보의 예로서는 사료, 체중, 질병, 예방접종 등이 포함된다.

### 2.3) 축사 환경 관리

생산단계 및 운송과정의 또 다른 중요 이슈는 가축에 대한 노출환경에 대한 것이다. 에너지 소비가 신진대사를 통해 체온을 조절하거나 극단적으로는 생존 자체를 위해 증가하기 때문에 생산성과 밀접한 관련이 있다. 이를 위해 환경 관리 시스템은 IoT 장비를 사용하나 온도, 습도, 복사열과 빛 등을 모니터링하고, 구동기를 사용하여 적절한 환경을 제어하는 것을 말한다.

#### 2.4) 초지 관리

초지 관리 서비스는 가축 사육을 위한 적절한 토지를 제공한다. FAO에 따르면 전 세계의 거주 가능한 육지 표면의 약 40%가 방목 시스템으로 덮여 있다고 한다. 이는 세계 농경지의 약 77%가 가축 사육을 위해 사용된다는 것을 의미한다. 여기에는 방목지와 사료생산을 위한 경작지가 포함되고 있다. 전 세계 초지는 정치적, 사회적, 경제적, 문화적인 환경에 따라 서로 다른 방목 관리 시스템을 갖게 되는데 가축 방목은 10억 명 이상의 사람들에게 생계를 제공하는 주요한 토지 이용 유형에 해당한다. 초지 관리 시스템은 센서와 드론을 활용하여 대기와 토양 상태를 모니터링하고 구동기를 사용하여 환경제어와 동물에 대한 영향을 제어하게 된다.

#### 2.5) 가축 건강 관리

가축의 건강 관리 서비스는 생산성의 저하 또는 인간의 건강에 대한 잠재적 위협과 관련이 있다. 이러한 측면에서 중요한 매개변수를 감지하고 통신할 수 있는 센서를 사용하여 적절한 시간 간격으로 서버나 클라우드에 제공할 수 있어야 한다. 예방접종이나 검사, 병해충, 질병 등 건강 관련 기록과 같이 중요한 정보는 안전을 위해 관리 시스템과 가축의 전자식별 장치에 등록되어 있어야 한다.

동물 복지는 동물의 건강 및 생산성과 관련된 중요한 문제이기도 하며 주로 동물의 생명존중과 관련된다. 이는 동물에게 지각이 있으며 그들의 안녕과 고통을 배려해야 한다는 믿음에 바탕을 두고 있다. 이 문제는 기상조건을 제어하고 건강을 모니터링하는 것과 상호 관련되어 있으며 GPS 모니터링 또는 이미지 인식 등을 통해 비정상적인 행동을 감지할 수 있는 시스템을 선택해야 한다.

#### 2.6) 분뇨 관리

분뇨 관리 서비스는 가축의 분뇨를 지속가능한 방식으로 수집하고 저장, 처리, 활용하는 것을 말한다. 다양한 보관시설을 활용하여 액체, 슬러리 또는 고체 형태로 발생하는 분뇨를 관리해야 하며 수질오염이나 수용불가능한 수준의 영양농축을 일으키지 않고 토양을 비옥하게 하기 위해 들판에 배포된다.

가축 분뇨는 황화수소, 메탄, 암모니아 및 이산화탄소 등 4가지 주요 독성 가스를 포함하여 여러 가지 가스를 생성한다. 이를 위해 분뇨 관리 시스템은 IoT 장치로 가축 분뇨를 모니터링하고 구동기를 통해 환경에 미치는 영향을 제어할 수 있어야 한다.

#### 2.7) 물류 추적 관리

물류 적성이란 제품이 최종 소비자에게 전달될 때까지 제품이 생산되는 과정과 물류, 유통 등 원산지과 다양한 단계를 식별할 수 있는 가능성을 의미한다. 명확하게 정의된 물류 이력 관리 서비스는 가축 또는 제품의 위치에 대한 정보를 서비스 이용자에게 실시간으로 제공할 수 있어야 하며 이를 위하여 위치 기반 서비스(LBS) 기능을 갖는 IoT 장치를 사용하여 가축이나 제품의 운송을 모니터링 해야 한다. 여기서 말하는 물류 이력 관리는 운송 측면에서의 생산 이력 관리와는 다른 개념이다.

#### 2.8) 생산품 추적 관리

최적의 작업 공정을 유지하기 위해 전체 가치사슬에 걸쳐 정확한 타이밍에 적절한 양의 입력을 제공하는 것이 중요하다. 빠르고 효과적인 회수를 위해 조직 간 이동 시 가치사슬을 통한 제품 단위 경로를 추적할 수 있어야 한다. 이러한 의미에서 바코드, QR 코드, RFID 태그 및 기타 추적 매체를 이용할 수 있으며, 가축 또는 부패하기 쉬운 제품의 운송은 추적성을 유지하기 위해 기후 변수 관련 기록을 추가하여 11.3 절을 준수해야 한다.

### 3) ICT 인프라

IoT 엣지 컴퓨팅과 관련하여 컴퓨팅, 스토리지 및 분석과 같이 IoT가 제공하는 일부 기능은 IoT 데이터 소스에 밀접하게 관련되어 진화하고 있다. IoT 장치 또는 게이트웨이에는 IoT 플랫폼 또는 IoT 어플리케이션 서버의 컴퓨팅, 스토리지 및 AI 지원이 필요할 수 있다. EC(Edge Computing)를 지원하기 위한 IoT 요구사항은 컨텍스트(예, 어플리케이션 및 네트워크 기술)에 따라 달라질 수 있다. EC는 식별된 요구사항과 관련하여 유연성(성능, 배포 모델 등)을 제공하기 위한 것으로 주어진 상황에서 어떤 서비스 및 기능 요구 사항이 필요한지 평가하는 것이 중요하다. 엣지 시스템은 사료 및 영양 관리, 축사 관리 등 다양한 SLF 서비스를 가능하게 한다.

#### 4) SLF 서비스를 위한 상위 레벨 요구조건

##### 4.1) SLF 서비스를 위한 공통 요구조건

###### ■ 식별

SLF 서비스에서 사육과 가공 및 추적을 위해 가축과 제품을 개별적으로 식별하는 것이 중요하며, 다음과 같은 요구조건을 충족해야 한다.

- 각 가축의 개별 또는 그룹 식별
- 먹이, 입력 식별
- 사육 장소의 식별
- 가축 제품 식별
- 각 가축 또는 가축그룹의 식별정보 관리
- 적절한 요청에 대한 식별 정보의 제공

■ **센싱 데이터**

SLF 가치사슬의 여러 단계와 절차에 따라 데이터를 평가하기 위해서는 ICT, 자동화 시스템, 무선 주파수 식별 및 IoT용 무선 센서 네트워크를 기반으로 하는 정보의 흐름을 구축해야 한다. 그런 측면에서 아래 표는 축산의 가치사슬에 따라 측정되거나 기록되어야 하는 다양한 데이터 유형을 포함하고 있다.

- 운송 상태 모니터링을 위한 차량 및 제품, 하중 데이터, 주변 환경 데이터 측정
- 추적을 위한 제품 데이터 측정
- 가축 건강 모니터링을 위한 개별 또는 그룹 가축 데이터 측정
- 생산성 모니터링을 위한 개별 또는 그룹 가축 데이터 측정
- 기후 모니터링 및 오염 모니터링을 위한 시설 상태 측정

모니터링 특성	측정 대상	센싱 파라미터
운송 모니터링	차량/제품/하중	온도 상대습도 풍속 진동 균형 공기압 무게 (부하 용량)
	주변 환경	온도 상대습도 빛 소음
제품 추적성	생산품	식별 번호 시설 식별 번호
가축 추적	가축 (개별 또는 그룹)	식별 번호 위치 수량 (작은 동물의 군집 또는 그룹)
가축 건강	가축 (개별 또는 그룹)	생리학적 매개변수(내부, 표면온도, 심박수, 호흡수, 움직임, 혈압, 코티솔 수치, 혈액 및 반추위 pH 등)
		이미지, 소리 인식(비정상적인 행동, 질병, 싸움, 사망 및 활동 감지) 백신과 검사 등록 품종 또는 품종의 식별 발정 식별 수정 기록
생산성	가축 (개별 또는 그룹)	무게 먹이 섭취량 물 섭취량 움직임
기후 조건 제어	시설	온도 상대습도 환기율 방사 빛 가스농도, 먼지
오염 제어	환경	공기 질 (가스 및 먼지 배출)
		수질 토양 오염 소음 생성

■ 네트워크 인프라

- (편재성) 네트워크 인프라는 항상 연결되어 있어야 하며 언제나 최신의 데이터를 유지해야 한다. 또한 서로 다른 장치와 상호운용성을 가져야 하며 안정적이고 실시간적인 유연한 메커니즘을 제공해야 한다.
- (무엇이든 연결) 다양한 유비쿼터스 네트워킹 통신방식을 지원하기 위해 사람-사람 통신, 사람-기기 통신, 기기-기가 통신을 제공하고 RFID와 같은 태그 기반의 장치 및 센서의 지원과 장치에 대한 식별, 이름 및 주소 지정 기능을 제공해야 한다.
- (개방형 웹 기반 서비스 환경) 서비스 환경은 대화형, 협업형 및 사용자 정의 기능을 사용하여 유비쿼터스 네트워킹 서비스 및 어플리케이션을 제공하는 풍부한 사용자 경험과 새로운 비즈니스 기회를 제공할 수 있어야 한다. 이를 위해 개방형 웹 기반의 서비스 환경을 기반으로 해야 하며, 기존의 통신 서비스와의 호환성, 동적인 상호작용을 통해 API 및 웹서비스를 제공해야 한다.
- (상황 인식) 상황에 대한 인식은 객체 상태의 변화를 감지하는 능력을 말하는 것으로 지능형 시스템과 연계하여 환경과 사용자의 상태를 인식하고 상황에 따라 최상의 서비스를 제공할 수 있어야 한다. 또한 어플리케이션이 시스템 및 다른 어플리케이션과 원활하고 투명한 상호작용을 할 수 있어야 한다.
- (다중 네트워킹) 전송 계층은 유니캐스트/멀티캐스트, 다중 homing과 다중 경로를 동시에 지원하기 위한 다중 네트워킹 기능이 요구된다. 상당한 양의 통신량과 수신기의 수로 인하여 유비쿼터스 네트워킹은 리소스 효율성을 위한 멀티캐스트 전송 기능을 가져야 한다. 멀티 homing은 다양한 고정/이동 액세스 기술을 포함한 여러 네트워크 인터페이스를 사용하여 장치가 항상 최상의 연결성을 유지할 수 있어야 한다.
- (상호 연결 네트워크를 통한 종단 간 연결성) 모든 객체는 NGN이나 기타 IP 기반 네트워크, 방송네트워크, 모바일/무선 네트워크 및 PSTN/ISDN과 상호 연결된 이종 네트워크를 통해 액세스가 가능해야 한다. IPv6는 개체에 전역적으로 고유한 주소를 제공하기 위한 좋은 방식으로 볼 수 있다.
- (용이한 유지보수) 네트워크 유지 관리는 네트워킹을 수행하고 유지하는데 요구되는 모든 문제와 절차를 다룬다. 이러한 작업은 구조화된 작업 또는 인터럽트 기반의 작업으로 간주될 수 있다. 이를 위해 네트워킹 문제에 빠르고 접근 가능한 문제해결을 제공해야 하며, 하드웨어 및 소프트웨어의 설치와 구성을 지원하고, 네트워크 성능 및 QoS 개선을 위한 자체 모니터링 기능과 각종 위협으로부터의 안전한 네트워크를 제공하기 위한 기능을 가져야 한다.

■ 클라우드 인프라 요구조건

- (스토리지 및 컴퓨팅) 클라우드 스토리지는 광범위한 온라인 공간에서 디지털 데이터를 저장하는 절차를 말하는 것으로, 일반적으로 호스팅 회사나 기관에서 이를 담당한다. 이를 위해 승인된 당사자가 다양한 장치를 통해 클라우드의 데이터에 접근할 수 있어야 하며, 클라우드 컴퓨팅을 사용하여 지정된 프로젝트를 수행하고, 항상 연결된 상태를 보장하기 위해 엣지 컴퓨팅 기능(옵션)을 제공할 수 있어야 한다.
- (고급 데이터 분석) 진보된 스마트 축산 시스템은 첨단 의사결정 지원 시스템과 실시간 평가를 통해 예측과 행동분석 기능에 대처하기 위해 기계학습과 인공지능(AI) 기술을 사용할 수 있어야 한다. 또한 이기종 형식의 대용량 데이터에서 핵심적인 내용을 추출하기 위해 빅데이터 분석 기술을 적용하는 것을 권장한다.
- (데이터 품질) 데이터 분석을 위해서는 다양한 소스의 데이터를 통합하는 것이 중요하다. 이 과정에서 오류 또는 중복으로 인한 데이터 품질 문제가 발생할 수 있는데, 아래 그림에서 보는 것과 같이 데이터 품질을 보장하기 위해 원 데이터에 대한 일련의 작업을 활성화 하는 것이 좋으며 대용량 데이터를 실시간으로 처리할 것을 권장한다.

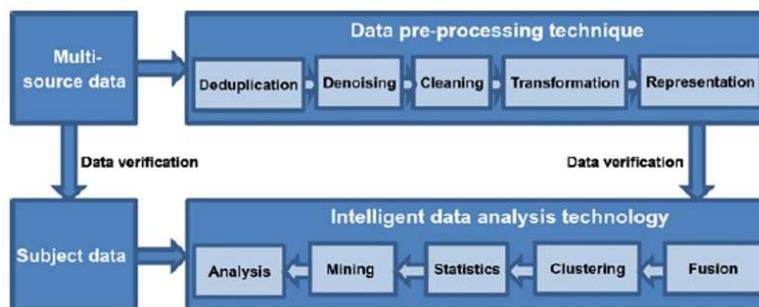


그림 2 - 빅데이터의 지능형 처리를 위한 흐름도

4.2) SLF 서비스를 위한 특정 요구조건

## ■ 사료 및 영양 관리

가축의 최적 체중 증가를 위해 IoT 장치와 사료 및 영양 관리 시스템의 통합이 필요하다. 이를 통해 각 가축의 사료 및 물, 영양 섭취량을 모니터링하고 가축의 정보와 이벤트 기록 기능이 요구된다.

## ■ 유전 및 번식

가축의 지속적인 개선을 보장하기 위해 가축과 관련된 성장과 건강 정보가 제공되어야 한다. 이를 위한 장치는 내부 및 표면 온도, 심박수, 호흡수, 움직임, 혈압, 코티솔 수치, 혈액 및 반추위 pH 등과 같은 생리적인 매개변수를 감지할 수 있어야 한다. 또한 비정상적인 행동과 질병, 싸움, 사망 및 활동 감지를 위해 가축의 이미지와 소리를 인식할 수 있는 시스템을 갖추는 것이 좋다. 이 외에도 백신이나 검사 등록, 품종식별, 발정상태와 수정이력의 기록이 요구된다.

## ■ 축사 환경 관리

효과적인 축사 환경 관리를 위해 축사 내부와 외부의 환경 상태를 감지하고 이를 인위적으로 조절할 수 있어야 한다. 또한 환경오염을 최소화 하기 위한 분뇨 관리 또한 지원되어야 한다.

## ■ 초지 관리

초지의 대기상태나 토양상태를 관리하기 위해 센서, 드론, 관개설비와 같은 IoT 장치를 적용할 수 있다. 이를 위해서 실외의 대기상태를 모니터링하고 습도 및 토양의 영양분과 같은 토양조건을 제어하고 이미지나 센서, 관개시설을 통한 풀의 성장을 모니터링하고 제어할 수 있어야 한다.

## ■ 가축 건강 모니터링 및 복지

생산성의 저하, 가축 질병 또는 인간 건강에 대한 위협요소를 방지하기 위해 IoT 장치를 이용하여 가축의 건강을 지속적으로 모니터링해야 한다. 가축 건강상태, 사육조건, 영양 섭취량 등에 대한 모니터링과 기록, 관리 기능이 필요하다. 또한 가축의 혈통과 같은 필수 정보 관리 기능이 지원되어야 하며 외부의 요청에 따라 이와 같은 건강 관련 기록을 제공할 수도 있어야 한다. 가축의 비정상적인 행동을 감지하고 농장주 또는 서비스 제공자에 가축 건강에 관한 모든 정보에 실시간으로 접근할 수 있어야 한다.

## ■ 분뇨 처리

액체, 슬러리 또는 고체 형태의 다양한 독성 가스와 같이 처리되지 않은 가축 분뇨로 인한 위험 영향을 방지하기 위해서 IoT 장치를 이용하여 가축의 분뇨를 모니터링해야 한다. 가스나 먼지 배출에 따른 공기, 물, 토양의 질을 실시간으로 모니터링할 수 있어야 하며 축사에서 발생하는 소음을 감지하고 제어할 수 있어야 한다.

## ■ 물류 추적 관리

가치사슬을 통해 제품 단위로 경로를 추적할 수 있는 수단을 제공하기 위해 위치 기반 서비스(LBS) 기능을 갖는 IoT 장치를 사용할 수 있으며, 이를 통해 가축이나 제품의 운송과정을 모니터링 할 수 있다. 이를 위해 제품의 원산지과 품질에 대한 기록, 운송 상태 모니터링과 운송 환경 제어, 생산, 가공 및 유통에 대한 다단계 추적기능, 시장 및 유통정보 기록 기능을 요구한다.

## ■ 생산품 추적 관리

제품의 원산지를 식별하기 위해 IoT 장치를 사용하여 제품 단위 경로를 추적할 수 있어야 한다. 제품의 유통기간 동안 고유한 식별이 필요하며 제품의 위조방지를 위한 수단이 제공되어야 한다.

### 5) SLF 서비스를 위한 참조 모델

SLF 서비스를 위한 참조 모델은 생산, 가공 및 운송 단계간 상호 연결과 모든 가축 생산 유형과 관련될 수 있는 정보의 흐름을 고려하여 가치사슬에 대한 일반적인 순서를 정의한다. 아래 그림에서 실선으로 표시된 링크는 단계간 또는 객체간 데이터의 흐름을 보이는 것으로 디지털 또는 물리적인 방식에 의한 정보의 교환을 나타낸다. 점선으로 표시된 링크는 프로세스 그룹, 장치, 인터넷 및 클라우드 인프라간 상호 연결을 나타내는 것으로 주로 광대역 링크에서 지원하는 기계적인 통신을 의미한다.

#### 5.1) 생산 전 단계

(계획) 계획단계는 생산 이전의 단계에 해당하며 실질적인 생산을 시작하기 전의 초기단계를 말한다. 이 단계에서는 다양한 필수품을 제공하기 위한 활동, 인력, 자재 및 생산능력에 따라 자원이 할당된다. 농장 운영에는 제품의 생산방법을 둘러싼 모든 세부사항이 포함된다. 여기에는 토지, 건물, 장비, 공급품 및 프로세스는 물론 비즈니스에 영향을 미치는 법률과 규정 역시 포함된다. 이 단계에서는 IoT 요구사항을 제시하고 이를 준수할 수 있도록 시스템 기능을 계획/설계한다.

#### 5.2) 생산 단계

- (번식) 번식은 생산 단계의 첫 번째 단계로서 가축간 번식이나 인공수정 또는 배아 이식 등을 통한 개체의 증가를 말하며 언제나 바람직한 유전형질의 개선을 모색해야 한다. 이 단계는 생산 전 단계의

계획 단계에서 제시된 계획에 적합해야 하며 다음 단계인 먹이공급 단계와 상호작용될 수 있어야 한다. 이를 위해 개별 가축의 식별, 유전과 번식, 가축 건강 모니터링 및 복지를 위한 IoT 요구사항을 준수해야 한다.

- (먹이공급) 이 단계는 생산 단계에서 경제적인 영향이 가장 큰 단계로 최종적인 제품의 품질과 효율성을 결정한다. 농부나 서비스 제공자 또는 관리시스템을 통해 실시간으로 액세스할 수 있어야 하며, 마찬가지로 생산 전 단계의 계획 단계에서 제시된 계획에 적합해야 하고 생산 단계의 번식 단계와 상호작용되어야 한다. 이를 위해 개별 가축의 식별, 사료 및 영양 관리, 축사 환경 제어, 초지 관리, 가축 건강 및 복지, 분뇨 처리를 위한 IoT 요구사항을 따라야 한다.
- (동물 제품) 이 단계에서는 제품의 수집이나 추출작업이 수행된다. 가축을 죽일 필요가 없이 얻을 수 있는 모든 제품과 관련된 것으로 유제품, 계란, 양모, 양봉 등이 있다. 이 단계에서 가장 중요한 IoT 요구사항은 가축의 식별과 제품의 처리 식별이며, 이 외에도 가축의 건강 모니터링과 복지, 생산 이력 관리 기능이 있다. 도축 단계와 마찬가지로 생산 단계의 마지막 단계로 간주할 수 있으며 이 단계 이후는 생산 후 단계가 시작된다.
- (도축) ‘동물 제품’ 단계에 대한 대안으로서 도축은 생산 단계의 마지막 단계가 된다. 음식이나 모피, 실트와 같은 특정 제품을 얻기 위해 가축을 도살하는 것으로 정의된다. 이 단계에서 필요한 IoT 요구사항은 개별 가축의 식별, 축사 환경 제어, 가축의 건강 모니터링과 복지, 생산 이력 관리 기능 등이 있다.

### 5.3) 생산 후 단계

#### ■ 포장 및 보관

동물성 제품을 얻은 후에는 식품 안전 표준과 관련 규정에 따라 적절하게 포장, 보관되어야 한다. 이를 위한 IoT 요구사항으로는 물류 이력 관리, 생산 이력 관리 기능이 있다.

#### ■ 운송, 물류 및 추적

물류는 가축의 가치사슬의 상당 부분에서 존재한다. 생산 중 정확한 시간에 필요한 양의 투입물을 제공하거나 최적의 작업 흐름을 유지할 수 있도록 해야 한다. 가축을 운송하는 과정에서 다양한 형태의 스트레스로 인해 제품 품질의 저하가 야기될 수 있으며 심각한 경우 사망에 이를 수도 있다. 이를 위해 서비스 제공자는 운송 및 가축의 상태에 대한 모든 정보를 실시간으로 접근할 수 있어야 하며, IoT 장비는 개별 가축 식별, 시설 환경 제어, 가축 건강 모니터링 및 복지, 물류 이력 및 생산 이력 관리 기능을 요구한다.

#### ■ 판매

판매 단계 동안에 제품은 최종 소비자 또는 소매업체가 구매할 때까지 일정기간 보관되어야 한다. 이를 위해 물류 이력과 생산 이력이 관리되어야 한다.

#### ■ 식품 추적

추적성 또는 제품의 추적은 각 단계간 이동 시 가치사슬에 의해 지정된 제품 단위의 경로를 추적하는 기능이다. 식품의 도매공급, 제조 또는 수입에 종사하는 식품사업은 안전하지 않은 제품에 대한 회수 시스템을 갖추어야 한다. 이 단계에서 필요한 IoT 요구사항은 물류 이력 및 생산 이력 관리 기능이다.

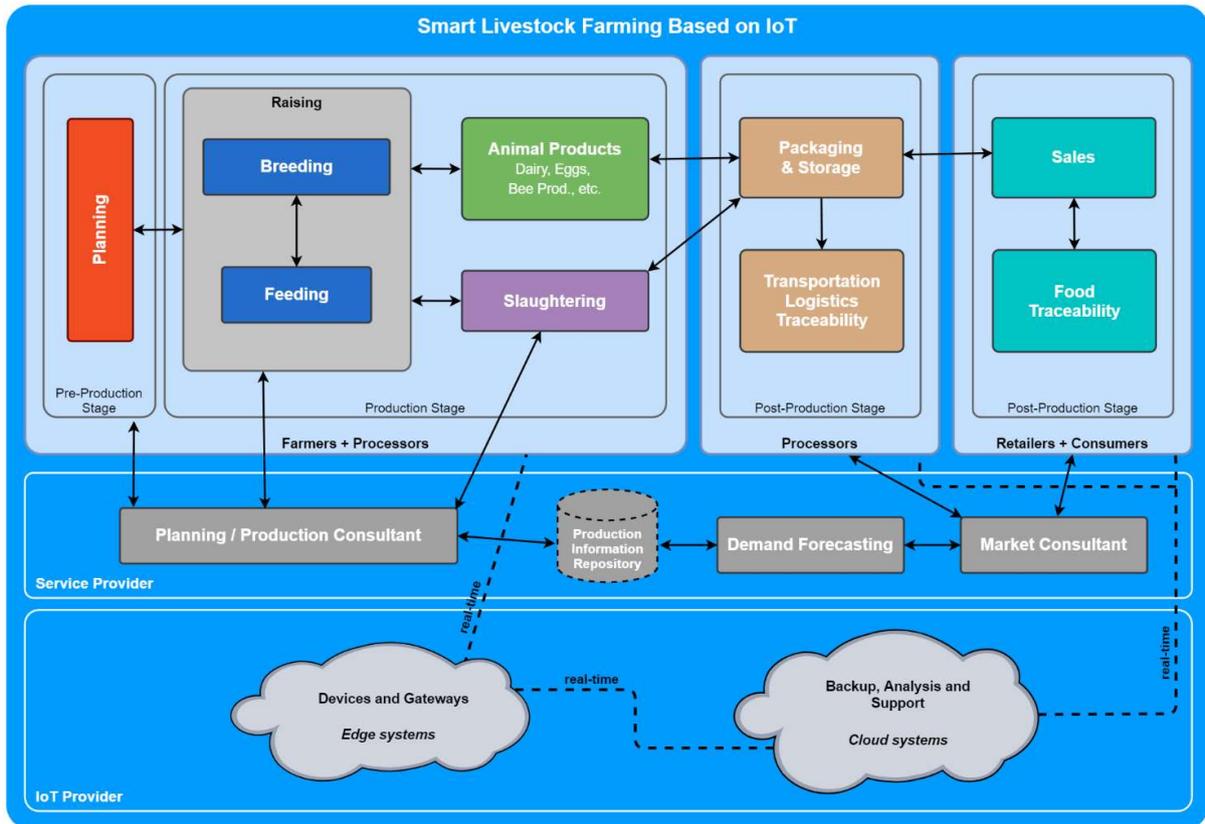


그림 3 - IoT 기반 SLF 서비스를 위한 참조 모델

나-4) ITU-T Y.Sup.SmartAgri-usecases

(가) 제정 경과

ITU-T SG20의 Q2에서는 스마트 농업을 위한 다양한 유즈케이스를 제공하기 위한 Supplement 문서 (Y.Sup.SmartAgri-usecases; Use cases of IoT based smart agriculture)를 개발하고 있다.

기후 변화, 물 부족, 급속한 도시화, 경작지 감소, 인구 고령화로 인해 전 세계적으로 식량 수급 문제가 예상된다. 스마트 농업은 농작물과 가축의 생육에 대한 정확한 데이터와 환경정보를 바탕으로 언제 어디서나 농작물과 가축의 생육환경을 확인하고 적은 노동력, 에너지, 영양분이 투입되더라도 농산물의 질과 양을 크게 향상시킬 수 있는 농업을 의미한다.

일반적으로 스마트 농업은 IoT, 빅데이터, AI, 자동화 시스템, 로봇 기술을 이용하여 원예(비닐 및 유리 온실), 노지, 축사 등의 시설물에 원격 또는 자동으로 농작물 및 가축의 성장 환경을 유지·관리할 수 있다. 스마트 농업에 적용되는 IoT 장치가 많기 때문에 기기간 상호운용성이 매우 중요하다. 따라서 기존 스마트 농업 기술과 향후 기술에 대한 조사가 필요하며, 조사 결과를 바탕으로 의미 있는 표준화 작업 항목을 도출할 수 있다.

이를 위해 2020년 7월 ITU-T SG20 회의를 통해 스마트 농업을 위한 유즈케이스 개발을 위한 신규 과제가 제안/채택되었으며, 당해연도 본 연구과제를 통해 Y.Sup.SmartAgri-usecases 표준화를 추진하기 위해 2회의 회의(virtual)에 참여하여 총 2건의 기고활동을 진행하였고 그 주요 내용은 다음과 같다.

표 1 - ITU-T Y.Sup.SmartAgri-usecases 개발 경과

순번	기고서 제목	제출일
1	Y.SmartAgri-usecases: Proposed modification on use case of smart greenhouse in clause 8.1	2020-10-20
2	Y.Sup.SmartAgri-usecases: Proposed new usecase "IoT cloud based smart agriculture"	2021-06-28
3	Proposed items to be discussed for Y.Sup.SmartAgri-usecases	2021-10-11
4	Y.Sup.SmartAgri-usecases: Proposal for a new use case "Connected agricultural machinery-Tractor"	2022-07-28
5	Proposal for a new use case "UAV-assisted pest control"	2022-07-28
6	Y.Sup.SmartAgri-usecases: Proposal for a new use case "Connected agricultural	2022-07-28

	machinery-Tractor”	
7	Y.Sup.SmartAgri-usecases: Proposal for a new use case “Automated dairy farm-IoT based milking system”	2022-07-28
8	Y.Sup.SmartAgri-usecases: Proposal for updated texts from caluse 8.4 to 8.7	2022-10-13

(나) 주요 내용

스마트 농업을 위한 유즈케이스는 다음 표와 같이 분류할 수 있다.

표 2 - ITU-T Y.Sup.SmartAgri-usecases - 분야별 유즈케이스 카테고리

ID 카테고리	유즈케이스 카테고리	유즈케이스
1	스마트 온실	- 스마트 온실 환경 조건의 자동 감시 및 제어 - 클라우드 기술 기반의 스마트 농업
2	스마트 노지	- 스마트 노지의 관수 관리 - 무인기를 활용한 병해충 방지 - 커넥티드 농작업기-트랙터
3	수경재배	- 모듈식 컨테이너를 활용한 수직 식물 농장
4	스마트 축사	- 클라우드 기술 기반의 스마트 농업 - 자동화된 낙농장(축사) - 스마트 축사의 가축 건강 관리
5	데이터 서비스	- 스마트 농업 데이터 서비스

1) 스마트 온실 환경 조건의 자동 감시 및 제어

본 유즈케이스는 스마트 온실에서 온도, 습도 및 토양 수분 수준을 자동으로 모니터링하고 작물 성장을 위한 최적의 조건으로 조정하는 방법을 설명한다.

작물의 성장률, 수확 시기, 생산량은 환경적 요인에 크게 좌우되기 때문에 많은 국가에서 온실을 도입하여 작물 생육 조건을 외부 환경 영향과 분리하고 있지만 대부분의 온실에서는 작물 생육 환경을 수동으로 제어하고 있다. 이에 환경 조건의 자동 모니터링 및 제어는 환경 상태를 모니터링하는 센서, 환경 조건을 조정하는 액추에이터, 최적의 환경에서 작물이 재배될 수 있도록 액추에이터를 활성화하는 컨트롤러와 같은 IoT 장치를 사용한다.

세계 인구증가, 고령화, 기후변화, 경작지 감소, 도시화에 따른 노동력 감소 등으로 ICT와 농업기술이 결합된 스마트 농업의 중요성이 강조되고 있다. 스마트 농업 분야 중 하나인 스마트 온실은 온도, 습도, 일사량 등 외부 환경에 상대적으로 독립적이며, 다음 그림과 같이 스마트 온실에 설치된 각종 센서와 구동기를 이용하여 최적의 작물 재배를 위한 환경 조건을 모니터링하고 조정한다. 환경 조건을 자동으로 모니터링하고 조정하기 위해 스마트 온실은 각 센서에서 환경 상태 정보(측정값)를 수집한 다음 적절한 구동기를 활성화하여 작물 성장 환경 조건을 조정한다. 이때 환경 상태, 제어 명령 및 실행 결과에 대한 정보를 전송하기 위해서는 적절한 통신 링크가 필요하며, 데이터 양이 많지 않기 때문에 광대역 네트워크를 갖출 필요는 없다.

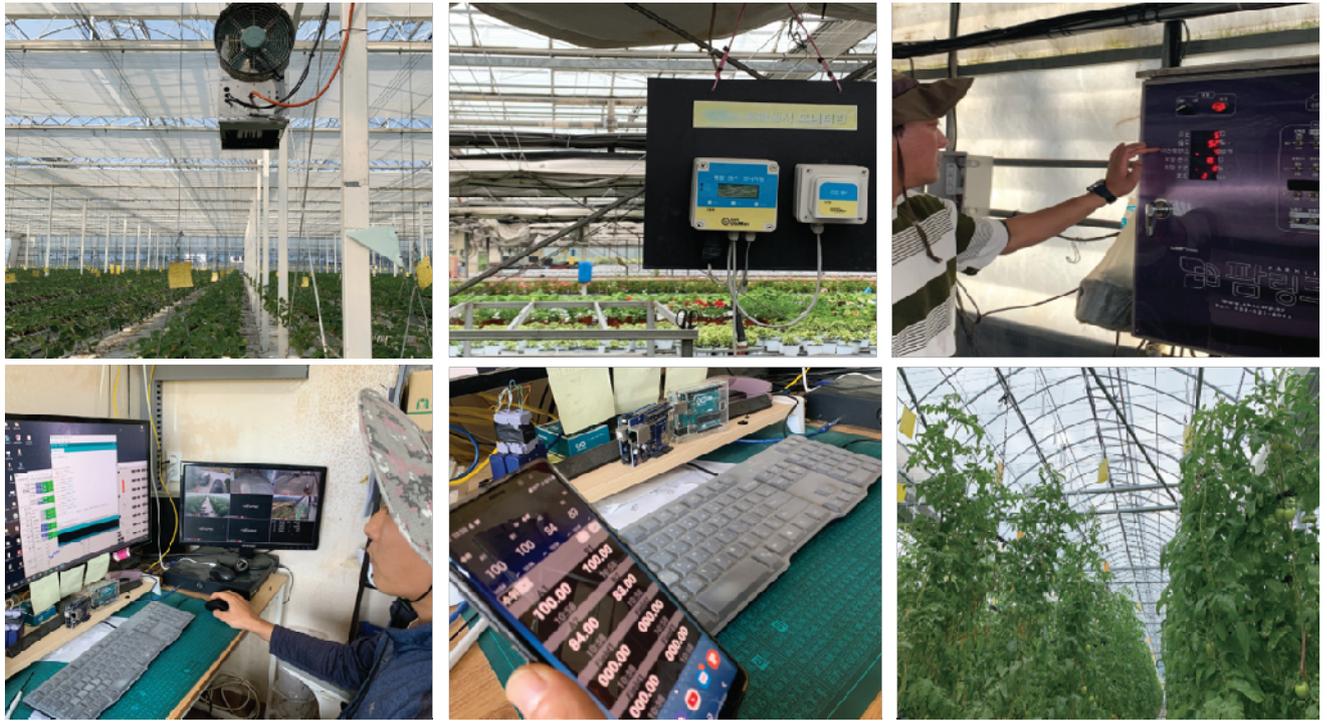


그림 1 - 자동 감시 및 제어 설비 유즈케이스

스마트 온실에 설치된 센서는 환경 조건을 측정하여 스마트 온실 관리 시스템에 정보를 전송하고, 스마트 온실 관리 시스템은 환경 조건을 분석하고 적절한 구동기를 활성화하여 작물 성장 환경을 최적으로 조정한다. 마지막으로 스마트 온실 관리 시스템은 구동기의 동작 실행 결과를 확인한다.

아래 그림은 이 유즈케이스를 위한 처리 과정을 다이어그램으로 나타낸다. 각 센서는 관리 시스템에 센서 장치의 유형과 측정 값 등을 포함한 관련 정보를 제공하고 관리 시스템은 온도나 습도, 환기조정 등의 제어 정보를 이용하여 구동기를 동작시킨다. 동작을 완료한 구동기는 작업결과 및 에너지 소비량 등의 정보를 포함한 실행 결과를 관리시스템에 제공한다.

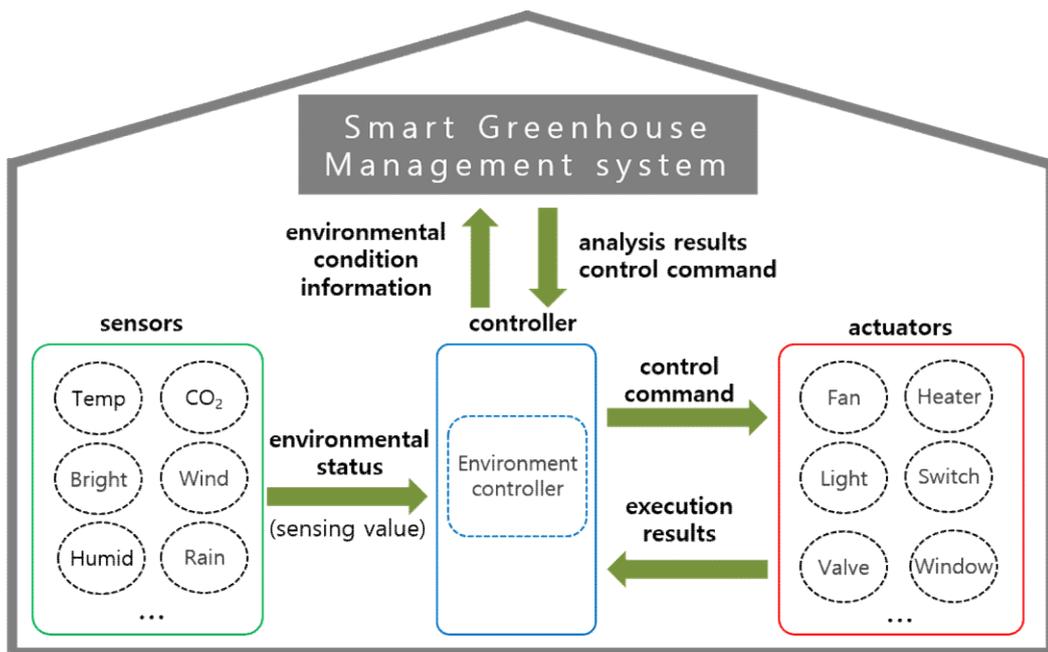


그림 2 - 환경 조건의 자동 감시 및 제어를 위한 처리 과정의 흐름도

스마트 온실의 제어기는 하나 이상의 센서에서 측정된 정보를 기반으로 하여 환경 조건을 조정하기 위

해 하나 이상의 구동기를 활성화할 수 있어야 한다. 관리 시스템은 구동기의 동작과 센서의 측정 정보에 관한 제반 정보를 수집한다. 스마트 온실에서 환경 조건을 모니터링하고 제어하기 위해서는 최소한의 유무선 통신링크를 갖추어야 한다. 또한 스마트 온실을 위한 사용자 인터페이스는 ICT 기술에 익숙하지 않은 사람들을 위하여 충분한 접근성을 제공해야 한다.

센서, 구동기 및 제어기의 모든 데이터는 장치 유형이나 제조업체와 관계없이 일반적인 형식으로 저장할 수 있어야 하며, 축적된 데이터(빅데이터)는 정밀농업과 농가의 경영 및 마케팅 전략, 작물의 최적 성장 모델을 수립하는 기초 자료로 활용된다. 환경 조건의 모니터링 및 제어를 위한 데이터는 상당한 기간 동안 저장되어야 한다.

수집된 데이터가 많을수록 보다 정확한 통계 및 분석이 가능하며 이는 생산성 향상과 직결된다. 그러나 저장 데이터의 양이 방대해지면 저장 공간의 유지 및 관리 비용이 수반된다. 따라서 이를 고려하여 적절한 데이터 저장 및 관리 전략을 수립할 필요가 있다.

미래의 기술 발전은 보다 정확하고 효율적인 데이터 수집을 가능하게 하지만 과거에 수집된 데이터와의 호환성이 보장되어야 한다.

센서 및 액추에이터의 상태를 원격으로 모니터링해야 하며, 이러한 장치는 장애가 발생하면 즉시 교체해야 하므로 원격으로 장치를 관리해야 한다.

스마트 온실을 악의적인 사용자로부터 안전하게 보호하기 위해서는 원격 제어를 위한 데이터뿐만 아니라 통신 링크도 보호되어야 한다.

## 2) 모듈식 컨테이너를 활용한 수직 식물 농장

물 부족, 사막화, 기후 변화 등 지구 온난화로 인해 농업에 사용할 수 있는 토양의 크기가 줄어들고 있으며, 증가하는 식량 수요의 압력에 대처하는 방법 중 하나로서 수직 식물 재배 방식이 있다. 수직 식물 재배 방식은 지정된 실내공간에서 다층 선반을 이용하여 식물을 재배하는 농법으로, 토양의 비옥도에 의존하지 않고 기후의 영향을 받지 않으며, 병해충 등 외부요인의 개입 없이 친환경적인 재배가 가능하여 부족한 공간으로 대량 재배가 가능하다. 또한 농촌에서 생산되어 도시로 배송될 때 배출되는 이산화탄소를 줄일 수 있다는 장점도 있다.

수직 식물 재배를 가능하게 하려면 별도의 건물이 필요하지만 별도의 건물이 아닌 모듈식 컨테이너 박스를 사용하면 도시에서도 수직 식물 재배를 쉽게 구현하고 설치할 수 있다. 다만, 수직형 식물 재배 시설의 초기 설치 비용은 옥외형 시설보다 훨씬 높기 때문에 수직형 식물 재배 시설의 설치 및 유지 관리 비용을 감당할 수 없는 국가에서는 어려울 수 있음을 고려해야 한다.

이 유즈케이스는 모듈식 컨테이너 상자와 빅데이터 센터를 사용하는 수직 식물 농장을 포함할 수 있다. 수직 식물 농장에는 여러 개의 센서와 구동기가 설치되어 있으며 제어기에 의해 제어된다. 한국에서는 컨테이너 박스를 활용한 수직 식물 농장을 건설하여 도시 농업을 가능하게 하는 노력이 진행되고 있으며, 다음 그림과 같이 각 컨테이너 상자에는 센서, 구동기 및 제어기가 설치되어 있다.



그림 3 - 모듈식 컨테이너 박스를 이용한 수직 농장 사례

수직 식물 농장에서 농업 데이터를 획득하고 원격으로 제어하기 위해서는 적절한 통신 링크가 필요하지만, 데이터 양이 많지 않기 때문에 광대역 네트워크를 고려할 필요는 없다.

수직 식물 농장의 센서는 빛, 온도, 습도, 이산화탄소 등과 같은 재배 조건을 모니터링하고, 구동기는 제어기의 명령에 따라 재배 조건을 조정한다. 빅데이터 센터는 농산물의 생산성과 품질을 극대화하기 위한 계획을 수립하기 위해 재배 조건 및 현황에 대한 정보를 수집한다. 수직 식물 농장의 광도, 온도, 습도 및 이산화탄소 등의 센싱 정보는 센서로부터 제어기로 제공된다. 센서로부터 수신된 센싱 정보를 기반으로 제어기는 적절한 구동기에 조명 켜기, 온도 올리기, 물 분사 등과 같은 일련의 작업을 요청한다. 이 센싱 정보와 그에 따른 제어 정보는 빅데이터 센터로 전송되어 수직 식물 농장 관리 계획을 수립한다. 수직 식물 농장은 빅 데이터 센터에서 수립한 계획에 따라 관리되며, 관리 계획을 수립하기 위해 수직 식물 농장의 재배 조건을 제어하는 제어기는 환경 및 제어 조건에 대한 정보를 제공해야 한다. 환경 조건 정보는 각 센서에서 생성된 센싱 정보이고, 제어 조건 정보는 각 구동기에서 수행하는 활동 정보에 해당한다.

수직 식물 농장의 환경을 감지하는 센서, 식물 재배를 위한 인공 환경을 조정하는 구동기, 감지 정보에 따라 구동기를 활성화하는 제어기, 내부 이벤트를 모니터링하는 CCTV, 빅 데이터 서버에 정보를 전달하는 IoT 게이트웨이가 고려되어야 한다. 지리적인 고려사항은 없다.

수직 식물 농장과 빅 데이터 센터 간의 통신, 센서/구동기와 제어기 간의 통신은 유무선 통신 링크를 기반으로 한다. 증가하는 식량 수요와 기후 변화의 압력에 대응하기 위해 IoT, 빅 데이터, 분석 및 클라우드 컴퓨팅을 통한 데이터 관리 및 활용이 매우 중요하다. IoT 장치는 데이터 수집을 가능하게 하고 수직 식물 농장에 설치된 센서는 토양, 대기 및 빛에 대한 정보를 실시간으로 생성할 수 있다. 이와 같은 정보는 다양한 목적을 위해 클라우드의 식물 생육 정보와 같은 다른 정보(생장 예측, 에너지 소비 최소화 등)와 함께 수집되어야 한다.

빅 데이터 센터가 수직 식물 농장을 위한 최적의 계획을 수립하려면 충분히 검증된 데이터가 매우 중요하지만, 검증된 데이터는 단기간에 획득할 수 없으므로 오랜 경험을 통해 데이터를 관리할 수 있어야 한다. 센서와 액추에이터는 습도가 높은 환경에 노출되기 때문에 상태를 원격으로 모니터링하고 필요한 경우 즉시 교체할 수 있도록 해야 한다.

### 3) 클라우드 기술 기반의 스마트 농업

클라우드 기술에 기반한 스마트 농업은 환경 및 작물 생육 정보를 수집 및 분석하여 스마트 온실 또는 스마트 축사의 경작 조건을 관리하는 것을 말한다. 스마트 농업에 필요한 데이터의 종류와 양은 물론, 생산성과 편의성을 높이기 위한 스마트 농업 서비스에 대한 사용자의 요구도 급격히 증가하고 있으며, 또한 아래 그림과 같이 독일에서는 '인더스트리 4.0'의 모범 사례를 참고하여 '농업 4.0'을 통한 농업의 디지털화를 추진하고 있다. 농업 4.0의 주요 기술에는 드론, 센서, IoT 기술, 로봇 공학, 빅 데이터 및 AI가 포함되며, 이러한 기술을 사용하여 정밀 농업과 자원 관리 프로세스의 자동화 및 기계화와 같은 다양한 비즈니스 모델이 생성되고 있다. 클라우드 지원 스마트 농업은 다음과 같은 기능을 통해 생산성을 개선하기 위한 방대한 데이터를 쉽게 분석할 수 있으나, 이러한 데이터를 분석하기 위한 개별 시스템을 소유하는 데 상당한 비용이 들기 때문에 클라우드를 활용하여 분석을 위한 비용을 절감할 수 있다.

- a) 작물 또는 가축에 대한 최적의 성장 모델 제공
- b) 정교한 의사결정 지원
- c) 이웃 농장 간의 데이터 공유

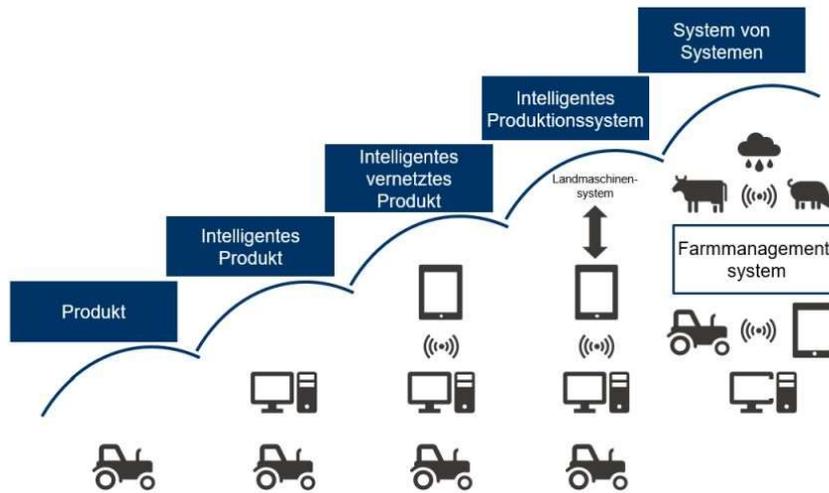


그림 4 - 농업 4.0의 농업 관리 시스템의 진화 방향

빅데이터와 AI 기술을 활용한 클라우드 기반 스마트 농업은 다양한 분야에서 활용될 수 있다. 환경, 생육 및 농업에 관한 모든 의미 있는 정보는 스마트 온실 또는 스마트 축사에서 생성되며, 이 정보는 실시간 또는 비실시간으로 클라우드로 전송된다. 그런 다음 클라우드는 빅데이터와 AI 기술을 사용하여 이를 분석하고 분석결과를 토대로 최적 생육 모델, 농장 관리 전략 수립 등 사용자에게 필요한 최적의 서비스를 제공한다.

다음은 클라우드 지원 스마트 농업 솔루션의 사례를 보인다.

[미국] 미국의 한 기업은 환경 정보와 관련된 빅데이터를 분석해 인공지능 기반 파종, 비료, 농약 살포 의사결정 지원 시스템을 개발하고 있다. 효율적인 데이터 분석을 위해 클라우드 컴퓨팅 사업도 강화하고 있다.

[이스라엘] 이스라엘은 작물 생육 상태와 수분 함유량 등을 자동으로 측정하여 작물 생산량을 예측하는 기술을 개발했으며, 솔루션 제품으로 클라우드를 통해 토양 상태를 측정하고 분석할 수 있다. 또한 사용자는 언제 어디서나 자신의 데이터에 액세스할 수 있다.

[일본] 일본 기업이 제공하는 농업 ICT 클라우드 서비스는 M2M 기술을 기반으로 한 스마트 온실 모니터링 서비스이다. 다양한 센서와 터미널의 네트워킹이 이 기술의 핵심으로서 센서를 통해 수집 및 축적된 현장 환경 데이터를 활용하여 생산량 및 생산 시간 예측의 정확도를 높일 수가 있다. 일본의 또 다른 회사의 솔루션 제품은 농업 생산 및 시설 제어에 대한 관리 서비스를 제공하는 것으로 클라우드는 매일의 농장의 작업 기록, 날씨 정보, 이미지 정보를 수집하여 외부 날씨, 지리 정보 등 부가 정보를 사용자에게 제공한다.

스마트 온실이나 축사의 내부 장치(센서, 구동기, 제어기)와 외부 장치(클라우드) 간에 환경, 생육, 제어 및 실행과 관련된 정보를 교환하려면 적절한 통신 링크가 필요하다. 특히, 이미지나 비디오 정보를 전송하기 위해 광대역 네트워크가 필요할 수 있다. 또한 네트워크 장애로 인해 외부 클라우드와의 통신 연결에 장애가 발생하면 작물이나 가축의 생육 환경을 제때 제어하기 어려울 수 있기 때문에 이를 극복하기 위해서는 클라우드 플랫폼과 엣지 노드 사이에 적절한 기능이 분산되어 있는 엣지-클라우드 기능이 있어야 한다. 엣지 노드의 기본 기능은 실시간 처리를 제공하는 데 필요한 컴퓨팅 성능을 확보해야 한다.

클라우드 기반 스마트 농업의 기본적인 시나리오는 다음과 같다.

- (환경 제어) 스마트 온실이나 축사에 설치된 센서는 환경 조건을 모니터링하고 측정된 정보(데이터)를 외부 클라우드로 전송한다. 클라우드는 수집된 데이터를 분석하여 빅데이터와 AI 기술을 활용한 환경 관리 계획을 수립하고, 이를 바탕으로 농작물이나 가축의 생육환경을 조정한다.
- (최적 생육 모델 생성) 클라우드는 축적된 데이터를 바탕으로 최적의 생육 모델을 수립하여 이에 기반한 정확한 영농법을 제시한다.

온도, 습도 및 이산화탄소와 같은 모든 센싱 데이터는 센서에서 생성되고 클라우드에서 수집된다. 그런 다음 클라우드는 수집된 데이터를 분석하여 온도, 습도 및 토양 영양과 같은 재배 환경을 조정하기 위한 일련의 적절한 조치를 생성한다. 서비스 사용자라면 누구나 특정 팜의 상태를 조회하거나, 환경 조건을 변경하거나, 특정 팜에 설치된 장치를 관리할 수 있어야 한다.

스마트 온실이나 스마트 축사에 설치된 각각의 센서는 센싱 데이터를 주기적으로 클라우드에 전달하고 클라우드는 센서에서 수집된 데이터를 분석하고 적절한 제어 명령을 생성하여 작물이나 가축의 생육에 적합한 환경을 조정한다. 또한 클라우드는 빅데이터 또는 AI 기술을 사용하여 작물 생산량을 예측할 수도 있다.

클라우드 지원 스마트 농업은 한 농장 또는 농장 그룹이 소유할 수 있으며, 각 농장의 장치(센서, 구동기, 제어기)는 클라우드와 직접 연결된다. 지리적인 고려사항은 없다.

클라우드는 유무선 통신 링크를 이용하여 다양한 서비스를 제공할 수 있다. 센서와 각 농장에서는 협대역 링크를 사용할 수 있지만 대용량 데이터 수집 및 영상 데이터 분석이 요구되는 경우에는 농장과 클라우드 간에 광대역 링크가 필요할 수 있다. 클라우드의 사용자 인터페이스는 ICT 기술에 익숙하지 않은 사용자에게 충분한 접근성을 제공할 수 있어야 한다.

센서, 구동기 및 제어기의 모든 데이터는 장치 유형이나 제조업체에 관계 없이 공통 형식으로 클라우드에 저장되어야 하며, 정확한 데이터 분석을 위해서는 환경 정보 및 생육 정보는 물론 경영 정보와 관련된 모든 데이터가 표준화된 방식으로 표현되어야 한다. 축적된 데이터(빅데이터)는 정밀농업을 위해 분석되어 농작물이나 가축의 최적 생육 모델과 관리 및 마케팅 전략 수립을 위한 기초 데이터로서 활용된다. 환경 조건의 모니터링 및 제어를 위한 데이터는 상당한 기간 동안 클라우드 저장소에 저장되어야 하며, 수집된 데이터가 많을수록 보다 정확한 통계 및 분석이 가능하고 이는 생산성 향상과 직결됩니다. 그러나 저장 데이터의 양이 방대해지면 저장 공간의 유지 및 관리 비용이 수반되므로 이를 고려하여 적절한 데이터 저장 및 관리 전략을 수립할 필요가 있다. 미래의 기술 발전은 보다 정확하고 효율적인 데이터 수집을 가능하게 하지만 과거에 수집된 데이터와의 호환성이 보장되어야 한다.

센서, 구동기 및 제어기의 상태를 원격으로 모니터링 해야 하며, 이러한 장치는 장애가 발생하면 즉시 교체해야 하므로 원격으로 장치를 관리할 수 있어야 한다.

#### 4) 스마트 노지의 관수 관리

스마트 노지의 관수 관리를 위한 유즈케이스로서, 농장에서의 과도한 물의 유입은 토양에 함유된 영양분의 유출 위험을 높이는 반면, 물 부족현상은 작물 수확량에 치명적인 영향을 미칠 수 있다. 이와 같이 물의 움직임을 제어하는 관수 관리 시스템을 통해 적절한 물의 공급량을 유지하면서 토양의 영양분을 관리할 수 있다.

#### 5) 무인기를 활용한 병해충 방지

무인기를 활용한 병해충 방지를 위한 유즈케이스로서, 무인항공기(UAV)는 해충 방제 작업에 활용될 수 있다. 카메라가 장착된 UAV는 해충의 영향을 받고 있지 않는 노지 상공을 비행하면서 작물의 고유 스펙트럼 신호를 기록하고 병해충의 발생 여부를 판단하여 이에 기반한 방제 시스템을 가동할 수 있다.

#### 6) 커넥티드 농작업기-트랙터

자동화된 트랙터와 같은 커넥티드 농작업기에 대한 유즈케이스로서, 농작업기는 IoT 기술의 도움으로 새로운 연결 서비스를 제공할 수 있다. 여기에는 지능형 차량 관리, 현장 데이터 사용을 위한 소프트웨어 및 펌웨어의 무선 업데이트에 이르기까지 다양하게 이용될 수 있다. 농업 효율성을 높여 농부가 기계를 보다 효과적으로 사용할 수 있게 하고 육체 노동의 필요성을 줄이며 필요할 때 장비를 운용할 수 있도록 한다.

#### 7) 자동화된 낙농장(축사)

자동화된 낙농장을 위한 유즈케이스로서, 젖소에게 먹이를 공급하고 착유를 하는 것은 자동화된 낙농장의 매우 중요한 목표 중에 하나이다. 최적의 우유 생산을 위해 낙농장 소유주는 항상 젖소가 잘 먹을 수 있도록 해야 하며, 자동 급이기는 젖소의 급이 상태를 모니터링하고 동시에 다른 사료를 혼합할 수 있다. 센서에서 생성된 데이터 분석에 따라 사료의 성분 비율과 공급이 자동으로 제어되며, 완전 자동화된 착유 시스템에는 착유실, 젖꼭지 위치 시스템, 젖꼭지 컵 부착 시스템 및 착유 시스템 등이 있다.

## 8) 스마트 축사의 가축 건강 관리

가축을 건강하게 관리하고 질병에 걸린 가축을 다른 가축과 분리하는 것은 축산업의 중요한 요소이다. 그러나 이 두 가지 모두 가축이 많은 대규모 농장에서는 어려운 작업이 될 수 있다. IoT 및 AI 기술에 기반한 가축 건강 관리 기술은 농부가 가축의 행동과 상태를 추적하고 건강을 모니터링 하며 잠재적인 질병이 감지되면 신속하게 경고함으로써 동물을 안전하게 보호하는 데 도움이 될 수 있다.

2000년대 한국에서 많은 구제역(FMD) 사례가 발생했다. 구제역은 소, 돼지, 양 등 발굽이 두 개인 동물에서 발생하는 바이러스성 급성 가축 전염병으로, 빠르게 전파되는 가장 위험한 A급 바이러스로 알려져 있다. 구제역은 감염된 동물의 콧물, 타액 또는 대변과의 접촉에 의해 직접적으로, 또는 차량 및 야생 동물에 의해 간접적으로 전염된다. 특히 호흡기를 통해 전염되기 때문에 한 마리만 감염되더라도 다른 모든 가축에게 빠르게 퍼지는 전염성이 높은 바이러스다. 국내에서는 2000년 2,216마리의 도살을 시작으로 2011년까지 약 3,479,962마리가 도살되었으며, 총 비용은 2,7383억 원, 미화 23억 달러에 달한다. 도살이란 전염된 가축으로부터 전염병이 퍼지는 것을 방지하기 위해 일정 반경 내에서 가축을 도축하는 것을 말한다.

공기 중 호흡기 감염인 구제역을 예방하는 가장 좋은 방법은 감염되지 않은 가축에게 신속하게 백신을 접종하는 것이다. 또한 농가의 경우 농가 출입 전후 철저한 소독을 실시하고, 헛간 출입 시 헛간 전용 신발을 신어야 하며, 일반인의 경우 농장에 진입하기 위해 소독시설을 통과할 때 적극적으로 협조하는 것이 중요하다. 그러나 그보다 더 중요한 것은 구제역을 예방할 수 있도록 환경을 통제하거나 이상상황을 선제적으로 파악하는데 있다.

이 유즈케이스는 IoT 기반의 데이터 수집 및 분석을 통해 이상상황을 식별하는 방법에 대한 실용적인 예를 제공하기 위한 것으로서, IoT 및 AI 지원 구제역 예방 시스템을 설치하는 초기 비용은 매우 높지만 가축의 손실을 줄일 수 있다는 점에서 유익하다는 장점이 있다.

이 유즈케이스는 축적된 데이터 세트를 분석하여 축산업에 종사하는 농부를 위한 가축 건강 관리 계획을 수립하기 위해 고유한 데이터 세트가 있는 축사 및 AI, 생명 공학 및 수의사와 클라우드를 포함한다.

국내에서는 돼지의 상태를 실시간으로 관리하고 다음과 같은 방식으로 수의사와의 상담을 할 수 있는 스마트 동물 건강관리 서비스가 개발되었으며, 2018년 자체 데이터 분석 결과 월간 항생제 사용량이 최대 83% 감소하고 전체 관리 비용이 43% 감소했으며 사망률도 30% 감소하여 생산성이 향상된 것으로 나타났다.

- 돼지의 혈액을 직접 채취하여 32가지 변수에 따라 면역상태, 면역활성화능력, 면역항체형성으로 분류하여 농가상태를 평가
- 농장에 센서를 설치하여 농장 상태를 관리
- 농가의 온도가 과도하게 상승하거나 일정 수준 이상의 가스가 발생하면 경보음이 울려 농가의 대응을 지원

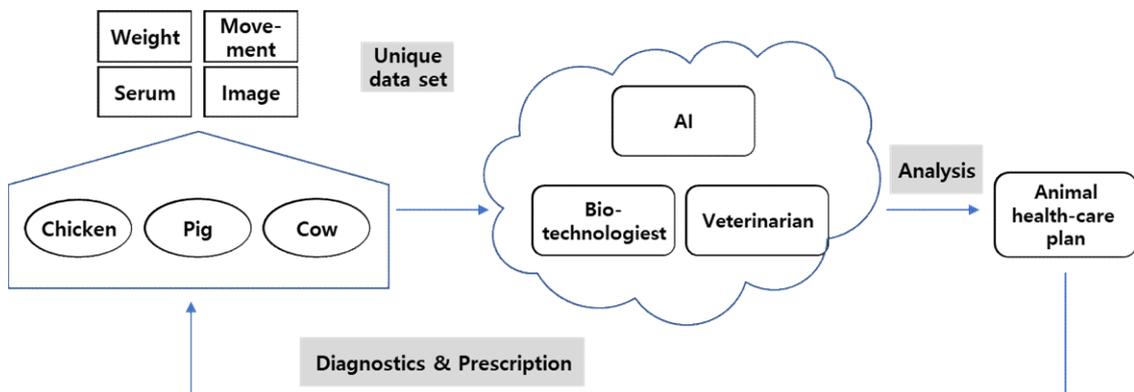


그림 5 - 가축 건강 관리 서비스의 객체와 각각의 역할

가축뿐만 아니라 축사에서도 센싱 데이터 및 영상 데이터를 얻기 위해서는 각 가축을 위한 적절한 통신 링크와 식별 방법이 필요하다. 데이터 양이 많기 때문에 현대역 네트워크는 적합하지 않다.

가축의 성장과 관련하여 수집된 다종 다양한 데이터를 분석하여 가축의 건강상태를 파악하고 적절한 처방을 농부에게 제공한다. 가축 건강 관리 서비스는 IoT, 데이터 및 AI 기술을 기반으로 하며, 데이터를 기

반으로 지속적인 가축 건강상태를 관리·분석하고 인공지능과 수의학적 전문성을 결합해 기존보다 나은 환경을 조성한다. 이는 다음과 같이 세 단계로 구성된다.

- (1단계) 가축에 대한 종합자료(체중, 움직임, 혈액 등) 수집
- (2단계) 인공지능, 생명공학, 수의학적 지식을 종합적으로 활용하여 가축의 건강을 측정하고, 인공지능으로 1차 소견을 도출한 후 최적의 건강관리 계획 도출
- (3단계) 수의사의 데이터 분석 결과를 바탕으로 정밀 진단 및 처방

축산에 대한 종합적인 정보(체중, 움직임, 혈액 등)는 가축 의료 서비스를 위한 클라우드로 제공되고 가축에 대한 처방 정보는 클라우드에서 농부에게 제공된다.

축사에는 가축의 상태를 파악할 수 있는 IoT 센서와 영상을 획득할 수 있는 카메라가 설치되어야 한다. 가축 건강 관리 서비스를 위한 클라우드는 가축의 상태 정보를 수집하고 다른 정보 제공자로부터 가축 질병 이력 정보를 얻을 수 있어야 한다. 수집된 정보를 분석하기 위해서는 전문가 그룹이 있는 AI 엔진이 필요하다. 외부 클라우드가 가축의 상태를 실시간으로 수집하기 위해서는 클라우드와 다수의 IoT 센서, 카메라가 설치된 축사 사이에 광대역 통신 인프라가 제공되어야 한다.

증가하는 식량 수요와 기후 변화의 압력에 대응하기 위해 IoT, 빅 데이터, 분석 및 클라우드 컴퓨팅을 통한 데이터 관리 및 활용이 매우 중요하다. 빅데이터 센터가 축사 및 가축의 데이터를 축적한다고 생각하면 충분한 검증된 데이터가 매우 중요하지만, 검증된 데이터는 단기간에 획득할 수 없으므로 오랜 경험을 통해 데이터를 관리하는 것이 더 중요할 것이다.

센서와 구동기는 높은 습도와 유독 가스 환경에 노출되기 때문에 상태를 원격으로 모니터링하고 필요한 경우 즉시 교체될 수 있어야 한다.

다양한 데이터는 가축의 건강 관리를 처방하는 데 사용되므로 안전하게 취급해야 한다.

## 9) 스마트 농업 데이터 서비스

스마트 농업에는 농산물의 생산단계, 유통단계, 소비단계가 있다. 각 단계에서 환경, 작물이나 가축의 생육상태, 가공 및 판매, 가격에 대한 방대한 데이터를 생성한다. 스마트 농업의 데이터 서비스는 데이터를 수집, 분석, 재생산한 후 결과 데이터를 서비스 형태로 사용자에게 제공하는 것을 말한다.

스마트 농업이 보편화되고 사용자의 요구가 다양해짐에 따라 더 많은 유형의 데이터가 방대하게 생성되고 있다. 이에 따라 스마트 농업은 클라우드, 빅데이터, AI 기술이 접목된 데이터 서비스 기반 산업으로 진화하고 있다.

스마트 농업을 위한 데이터에는 다음과 같은 정보가 포함된다.

- 생산 데이터 : 관리(재배작물의 종류, 농장시설의 종류 및 규모), 환경(온도, 습도), 방제(방제명령과 실행결과), 생육(수확, 병해충, 병해충) 등의 정보
- 유통 데이터 : 이력(재배이력, 판매이력), 도매(실시간가격, 거래량) 등의 정보
- 소비 데이터 : POS 정보, 매출(거래량, 소비동향), 인증(친환경인증) 등의 정보

다음 그림과 같이 일본에서는 'Society 5.0'이라 불리는 스마트 농업을 위한 데이터 서비스 연구를 지원하고 있다. 연구는 다음과 같은 목표를 가지고 있습니다

- 진단 : WebGIS 시스템은 작물이나 농지의 다양한 정보(재배상태, 적정 수확시기 등)를 진단하고 그 결과를 사용자의 기기에 사용하기 쉬운 형태로 제공
- 물 관리 시스템: 물 관리 시스템은 사용자의 장치를 사용하여 급배수를 자동으로 관리(물 관리를 위한 노동력 약 95% 절약)
- 스마트 기계 개발: 스마트 농업 기계는 적절한 시비량을 조정(생산 비용 절감, 작업 효율성 향상)
- 데이터 공유: 데이터 플랫폼은 토양 상태, 수온, 농산물 품질 등의 데이터를 이용자에게 제공하고 부가 가치 서비스 제공

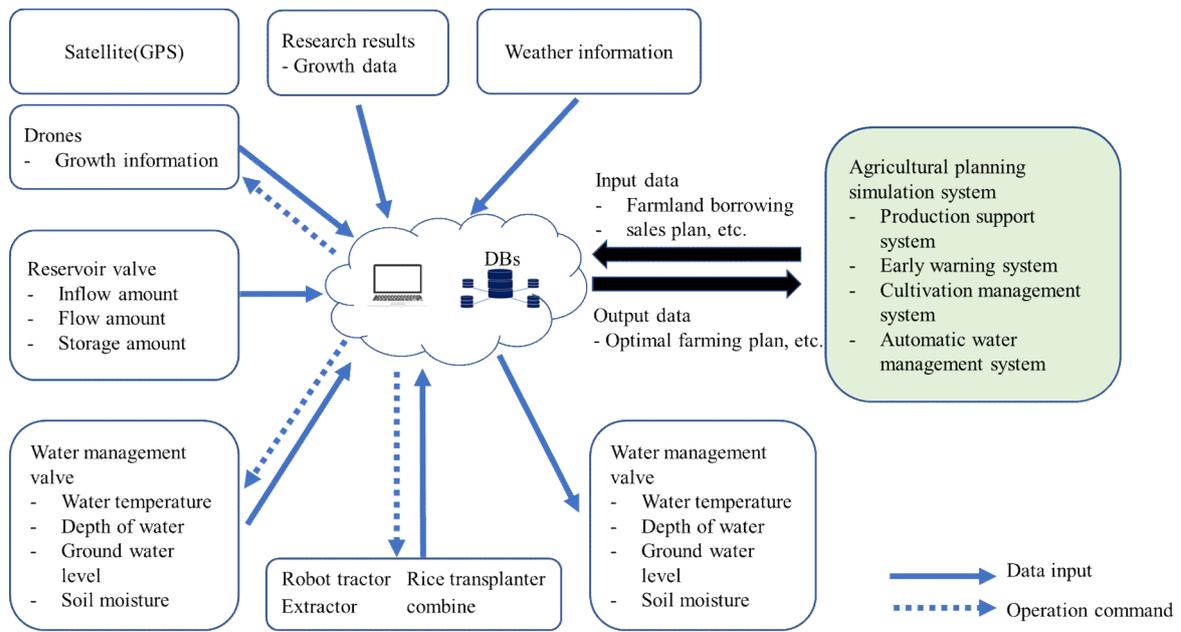


그림 6 - 스마트 농업에 의한 Society 5.0의 개념

스마트 농업 데이터 서비스는 스마트 농업의 전 단계에서 모든 데이터를 수집, 분석 및 재생산하여 다음과 같은 기능을 통해 생산성을 향상시킨다.

- 정확한 생산 계획 수립
- 농산물 공급 및 소비 관리
- 해충 및 질병 통제
- 효과적인 의사결정 지원 시스템 제공
- 스마트 농업 시스템의 운영 및 장애 상태 관리
- 인건비 절감 및 재배에 필요한 재료비 절감

또한 친환경 인증, 판매 이력 등의 제품 품질 정보를 제공하는 스마트 농업 데이터 서비스로 소비자가 고품질의 농산물을 구매할 수 있도록 한다.

스마트 농업에서 데이터를 생산, 분석, 소비하는 주체는 다음과 같다.

- 데이터 생산자: 농가(스마트 온실, 스마트 축사), 기업(농산물 가공업자, 유통업자 및 판매자) 및 데이터 서비스 제공자(데이터 재생산자)
- 데이터 분석기: 데이터 서비스 제공자
- 데이터 이용자 : 농가 및 기업, 재생산된 데이터의 최종 소비자, 농산물의 최종 소비자, 정부기관 및 연구기관 등

데이터 생산자가 생성한 데이터는 데이터 분석기로 전송된다. 데이터 서비스 제공자로서 데이터 분석기는 데이터 사용자의 요구사항에 따라 특정 서비스를 제공하기 위해 전송된 데이터를 분석하고 데이터를 재생산한다. 데이터 사용자는 데이터 분석기에서 제공한 데이터를 자신의 목적을 위해 활용한다. 데이터 서비스 제공자는 공공 데이터(날씨 정보, 지도 정보, 통계 정보 등)를 활용하여 보다 유용한 서비스를 제공하고 관련 데이터를 다른 서비스 제공자와 공유할 수 있다.

다음은 스마트 농업 데이터 서비스의 세계적인 사례를 보인다.

[일본] 일본 기업은 AI, IoT 및 빅데이터 기술을 기반으로 수확 시기 조정, 수확량 예측, 재배 기록 관리, 원격 장치 관리를 위한 데이터 서비스를 제공한다. 또한 블록체인 기술을 활용한 고품질 식품안전 서비스를 개발하고 있다.

[포르투갈] 한 포르투갈 회사는 실시간 관리형 클라우드 플랫폼을 통해 다양한 제조업체의 다양한 컨트롤러를 통합할 수 있는 플러그 앤 플레이 솔루션을 제공한다. 이 솔루션을 이용하면 실시간으로 스마트 농업 시스템에 접속하고 클라우드에 연결된 모든 사업자의 정보 데이터를 동시에 관리할 수 있다.

[미국] 미국의 한 회사는 비료 및 살충제의 투입 최적화를 포함한 데이터 서비스를 제공한다. 이 회사의

성장 현황은 미국 농장의 70% 이상을 차지할 것으로 예상된다.

[대한민국] 2021년에는 인공지능 기술을 활용한 데이터 플랫폼을 개발해 스마트 온실에서 환경 및 생육 데이터를 분석해 최적의 재배 조건 정보를 농민에게 제공하고 있다. 이 플랫폼의 주요 목적은 다음과 같습니다.

- 데이터를 활용한 농업 분야 진단
- AI 기술을 활용하여 적합한 작물 유형 추천 및 정확한 재배 방법 지원
- 유통 및 판매 활동의 편의성과 생산성 향상
- 데이터 기반 디지털 농업 시스템 추진
- 빅데이터 기술을 생산, 유통/판매, 소비 단계에 적용

스마트 농업 데이터 서비스는 스마트 온실이나 축사에 설치된 기기간 데이터 교환을 위한 통신 링크를 필요로 한다. 또한, 스마트 온실이나 축사와 데이터 서비스 제공자의 외부 클라우드 시스템 간의 연결을 위해 광대역 네트워크가 필요할 수 있다. 모든 데이터는 장치 유형이나 제조업체에 관계없이 공통 형식으로 클라우드에 저장되어야 하며, 정확한 데이터 서비스를 위해서는 스마트 농업과 관련된 모든 데이터가 표준화된 방식으로 표현되어야 한다. 또한 데이터 상호 운용성을 위한 체계적인 데이터 모델을 구축하는 것도 고려할 수 있다.

스마트 농업 데이터 서비스의 기본적인 시나리오는 다음과 같다.

- (데이터 집계) 스마트 온실이나 축사에 설치된 센서, 구동기, 제어기 및 운영 시스템에서 생성되는 모든 데이터는 데이터 서비스 제공자에게 주기적으로 전송된다. 농산물의 가공, 유통, 판매와 관련된 데이터도 데이터 서비스 제공자가 수집한다.
- (데이터 분석 및 재생산) 데이터 서비스 제공자는 전송된 데이터를 외부의 공개 데이터(날씨, 지도, 통계정보 등)와 함께 AI와 빅데이터 기술을 활용하여 분석하고 새로운 형태의 데이터를 재생산한다.
- (데이터 배포) 재생산된 데이터는 데이터 사용자에게 서비스의 형태로 제공된다. 데이터 사용자는 농부, 농산물 최종 소비자, 정부 기관 또는 연구 기관 등이 될 수 있다.
- (데이터 소비) 데이터 이용자는 생산성 향상(농가, 기업), 스마트 농업 관련 정책 수립, 스마트 농업 신기술 개발 등 자신의 목적을 위해 데이터를 활용한다. 또한 이러한 데이터를 통해 최종 소비자는 고품질 농산물을 선택할 수 있다.

아래 그림은 스마트 농업에서 데이터 서비스의 흐름도를 보여준다.

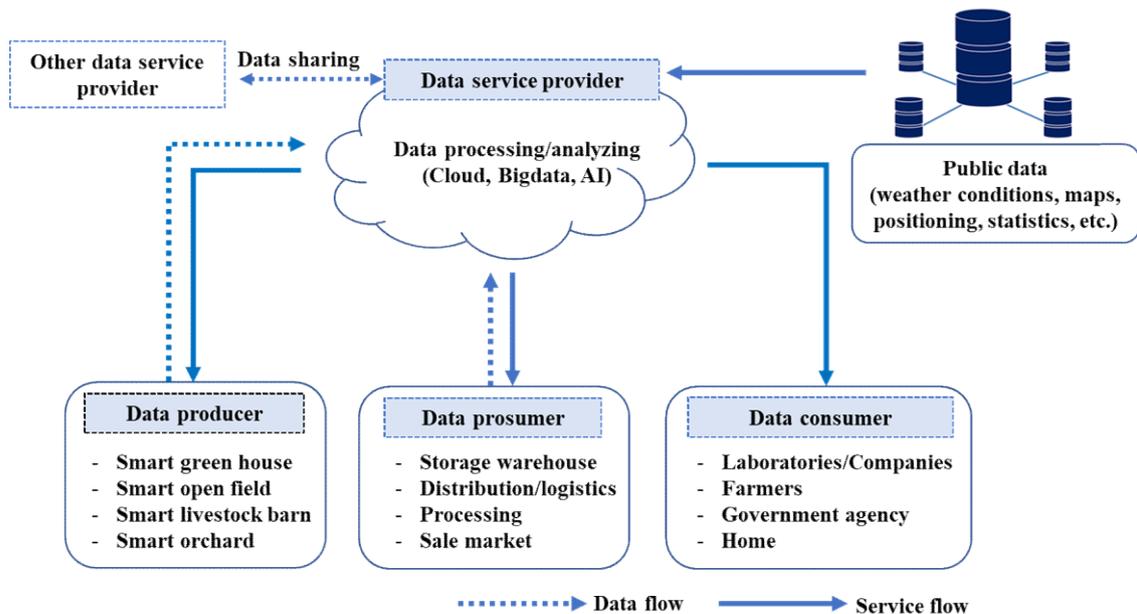


그림 7 - 스마트 농업의 데이터/데이터 서비스를 위한 프로세스 흐름도

스마트 온실이나 축사에서 생산된 환경 데이터, 생육 데이터, 농장 경영 데이터 등의 정보는 데이터 서비스 제공자에게 전송된다. 데이터 서비스 제공자는 전송된 데이터를 분석한 후 분석된 데이터를 데이터 서비스 이용자(농가, 정부기관, 연구기관 등)에게 서비스의 형태로 제공한다. 농산물의 가공, 유통, 판매와

관련된 기업들도 데이터 서비스 제공자에게 데이터를 전송한다.

스마트 온실이나 축사에 설치된 센서, 구동기, 제어기 등의 장치는 클라우드 시스템을 가진 데이터 서비스 제공자와 직접 연결된다. 개별 농가에도 데이터 분석 시스템을 설치할 수 있지만, AI와 빅데이터 기술을 활용해 대용량 데이터를 실시간으로 분석하기 위해서는 외부 클라우드 시스템을 활용하는 것이 경제적이다. 모든 데이터 서비스 사용자는 웹 또는 모바일 장치를 사용하여 항상 정보를 조회할 수 있어야 한다. 데이터 서비스 제공자의 클라우드 시스템은 유무선 통신 링크를 이용하여 다양한 서비스를 지원할 수 있다. 농장에 설치된 센서는 협대역 링크를 사용할 수 있지만 대용량 데이터 수집 및 영상 데이터 분석을 위해 농장과 데이터 서비스 제공자의 클라우드 시스템 간에 광대역 링크가 필요할 수 있다.

스마트 농업을 위한 데이터 서비스 제공자의 클라우드 시스템의 사용자 인터페이스는 ICT 기술에 익숙하지 않은 사용자에게 충분한 접근성을 제공할 수 있어야 한다.

센서, 구동기 및 제어기의 모든 데이터는 장치 유형이나 제조업체에 관계없이 공통된 형식으로 데이터 서비스 공급자에게 전달 및 저장되어야 한다. 정확한 데이터 분석을 위해서는 필요한 데이터 서비스와 관련된 모든 데이터가 표준화된 방식으로 표현되어야 한다.

축적된 데이터(빅데이터)를 분석하여 생산관리, 판매, 유통관리 등 다양한 서비스를 제공한다. 이렇게 분석된 데이터는 농작물이나 가축의 최적의 생육 모델과 관리 및 마케팅 전략을 수립하기 위한 기초 데이터로서 활용된다. 다양한 데이터 서비스를 가능하게 하기 위해서는 모든 데이터가 클라우드 시스템의 스토리지에 상당 기간 동안 저장되어야 한다. 보다 정확한 통계와 분석이 필요하므로 농장의 생산성과 편의성을 높이고 데이터 서비스 이용자에게 보다 다양한 서비스를 제공할 수 있다. 그러나 저장된 데이터의 양이 방대해지면 저장 공간의 유지 및 관리 비용이 수반되므로 이를 고려하여 적절한 데이터 저장 및 관리 전략을 수립할 필요가 있다. 미래의 기술 발전은 보다 정확하고 효율적인 데이터 수집을 가능하게 하지만 과거에 수집된 데이터와의 호환성이 보장되어야 한다.

스마트 농업을 위한 데이터를 생산하는 모든 장치는 원격으로 관리되어야 한다. 데이터 서비스 제공자 및 서비스 사용자는 장치의 작동 상태를 조회하고 고장난 장치를 즉시 수리 또는 교체할 수 있어야 한다.

스마트 농업용 데이터를 악의적인 사용자로부터 안전하게 보호하기 위해서는 원격 제어용 데이터뿐만 아니라 통신 링크도 보호되어야 한다.

라) 국제의장단 수입

라-1) [국제의장단] ITU-T Y.4466 에디터 수입

본 연구과제를 통해 스마트 온실 서비스를 위한 프레임워크 표준(Y.4466; Framework of Smart Greenhouse Services)을 국내 업체들의 기술을 국제 표준에 반영하기 위해 아래와 같이 국제의장단(에디터)을 수입하여 국내 현장 요구사항을 최대한 반영하였다.

**[2017-2020] : [SG20] : [Q4/20]**

**[Declared patent(s)] - [Publication]**

*Work item:* [Y.4466 \(ex Y.ISG-fr\)](#)

*Status:* [Approved on 2020-01-13 \[Issued from previous study period\]](#)

*Approval process:* [AAP](#)

*Type of work item:* [Recommendation](#)

*Version:* [New](#)

*Provisional name:* [Y.ISG-fr](#)

*Equivalent number:* -

*Timing:* -

*Liaison:* [ITU-T Q.1/13, IEEE](#)

*Supporting members:* -

*Subject/title:* [Framework of smart greenhouse service](#)

*Summary:* [A smart greenhouse service enables precision farming with help of IoT devices \(such as sensors and actuators\) installed in a smart greenhouse. A smart greenhouse service collects information about both environment and crop-growth status, and then analyses the information to produce an optimal growth model for each crop. With the optimal growth model, a smart greenhouse service can maximize agricultural productivity and improve crop quality. In addition, it can enhance user convenience. To describe a smart greenhouse service framework, this Recommendation specifies requirements, a reference model, a functional architecture and interfaces for a smart greenhouse service.](#)

*Comment:* -

*Base text(s):* [\[SG20-TD1555/GEN \(2019-11\) !\[\]\(57c8280bf191cdc854ec78639599d448\_img.jpg\)](#)

*Contact(s):* [Munhwan Choi, Editor](#)  
[Juyoung Park, Editor](#)

*ITU-T A.5 reference(s):* -

 [\[Submit new A.5 reference !\[\]\(b048bed441d9f6fe58ca8daaced3fa1f\_img.jpg\)](#)  
[See guidelines for creating & submitting ITU-T A.5 justifications](#)

*First registration in the WP:* [2015-11-12 15:20:31](#)

*Last update:* [2020-01-27 17:37:39](#)

라-2) [국제의장단] ITU-T Y.4482 에디터 수임

Y.4466 표준은 스마트 축사에 대한 프레임워크 및 기능 요구사항(Y.IoT-SLF: Framework and Capabilities for Smart Livestock Farming based on Internet of Things)을 개발 중인 브라질에게 훌륭한 참고 자료로 활용될 수 있기 때문에 브라질의 요청에 따라 Y.IoT-SLF 표준 개발에 직접적인 참여를 요청받았으며 이에 Y.4466 표준 개발 경험을 토대로 2020년 이후 Y.IoT-SLF 표준에 대한 공동 에디터십(editorship)을 가지고 협업을 진행하고 있다.

**[2017-2020] : [SG20] : [Q2/20]**

**[Declared patent(s)]**

Work item:	Y.IoT-SLF
Status:	[Carried to next study period]
Approval process:	AAP
Type of work item:	Recommendation
Version:	New
Provisional name:	Y.IoT-SLF
Equivalent number:	-
Timing:	-
Liaison:	ISO; ETSI; AIOTI WG06; GSMA, 3GPP; TTA; ANSI; BSI; IoT Forum; FAO; AEF; ATF
Supporting members:	Brazil, Senegal, Guinea, Cote d'Ivoire, Togo, Gambia, Nigeria, Republic of Congo and ETRI.
Subject/title:	Framework and capabilities for smart livestock farming based on Internet of things
Summary:	Smart Livestock Farming (SLF) is a concept that refers to modern Information and Communication Technologies (ICT) applied into the livestock value chains. It has the potential to deliver a more productive and sustainable production by integrating processes of the Precision Livestock Farming (PLF), Management Information Systems (MIS), stockbreeding automation and robotics to provide a better decision making or more effective exploitation operations and management. The use of Internet of Things (IoT) technologies in the SLF aims at providing a full coverage of the processes by collecting and transmitting data from the entire agroecosystem. That means SLF can establish contact with each participant of a livestock chain, bringing and collecting information about their processes, increasing the possibilities for control and improvement on the efficiency of their tasks. This Recommendation provides an overview of SLF based on IoT, as well as proposes a reference model for the SLF services, specifies SLF services requirements to address the challenges for smart and connected livestock value chains. It is applicable to these chains as a whole, regardless of species or rearing techniques.
Comment:	-
Base text(s):	[SG20-TD2472/GEN (2021-10) 
Contact(s):	<p>Munhwan Choi, Editor</p> <p>Carlos Eduardo Cugnasca, Editor</p> <p>Gustavo Marques Mostiaco, Editor</p> <p>Juyoung Park, Editor</p> <p>Ricarda Carolina Rende, Editor</p> <p>Sayyadi Sani, Editor</p> <p>João Alexandre Moncaio Zanon, Editor</p>
ITU-T A.5 reference(s):	-
	 <p>[Submit new A.5 reference </p> <p>See guidelines for creating &amp; submitting ITU-T A.5 justifications</p>
First registration in the WP:	2018-05-25 09:26:39
Last update:	2021-10-26 08:59:45

라-3) [국제의장단] ITU-T Y.Sup.SmartAgri-usecases 에디터 수입

스마트 농업을 위한 다양한 유즈케이스의 개발 필요성에 따라 2020년 7월 ITU-T SG20 회의를 통해 스마트 농업 유즈케이스를 위한 신규 표준화 과제를 제안하였으며, 회의 결과에 따라 이후 Y.Sup.SmartAgri-usecases (Use cases of IoT based smart agriculture) 표준 개발을 추진하고 있으며, Y.IoT-SLF의 협업 과정에서 거둔 괄목할 성과에 기반하여 이 표준 역시 브라질과 공동 데이터섭을 가지고 표준화를 진행하고 있다.

**[2017-2020] : [SG20] : [Q2/20]**

**[Declared patent(s)]**

<i>Work item:</i>	Y.Sup.SmartAgri-usecases
<i>Status:</i>	[Carried to next study period]
<i>Approval process:</i>	Agreement
<i>Type of work item:</i>	Supplement
<i>Version:</i>	New
<i>Provisional name:</i>	Y.Sup.SmartAgri-usecases
<i>Equivalent number:</i>	-
<i>Timing:</i>	-
<i>Liaison:</i>	ITU-T SG13, ISO TC23, ISO/IEC JTC1/SC41, W3C
<i>Supporting members:</i>	ETRI, Korea (Republic of), Brazil, EPUSP, Universidad Tecnológica Nacional (Argentina)
<i>Subject/title:</i>	Use cases of IoT based smart agriculture
<i>Summary:</i>	<p>Due to climate change, water shortage, rapid urbanization, reduced cropland, and aging population, food supply and demand problems are expected globally. Smart agriculture is based on accurate data on the growth and environmental information of crops and livestock, and checks the growth environment of crops and livestock anytime, anywhere, and prescribes them in a timely manner, even if less labor, energy, and nutrients are introduced than before. It means agriculture that can greatly improve quality and quantity of products. In general, smart agriculture can remotely or automatically perform the maintenance and management of the growth environment of crops and livestock by using IoT, big data, AI, automation systems, and robot technologies to facility horticulture (vinyl and glass greenhouses), houses, and orchards. Because there are a number of IoT devices applied to smart agriculture, it is very important to know their interaction for interoperability. In this reason, it is necessary to survey existing smart agriculture technologies as well as forthcoming and then based on the survey results we can deduce meaningful standardization work items. This new work item will survey use cases related to smart agriculture in the perspective of, but not limited to: 1) smart greenhouse, 2) smart open field, 3) smart hydroponics, 4) smart livestock barn, and 5) smart agriculture data service.</p>
<i>Comment:</i>	-
<i>Base text(s):</i>	[SG20-TD2516/GEN (2021-10) 
<i>Contact(s):</i>	<p>Munhwan Choi, Editor</p> <p>Gustavo Marques Mostaco, Editor</p> <p>Juyoung Park, Editor</p> <p>Rodrigo Santana dos Santos, Editor</p>
<i>ITU-T A.5 reference(s):</i>	<p> [Submit new A.5 reference </p> <p>See guidelines for creating &amp; submitting ITU-T A.5 justifications</p>
<i>First registration in the WP:</i>	2020-07-20 17:29:22
<i>Last update:</i>	2021-10-29 16:19:15

라-4) [국제의장단] IEC 프로젝트 리더 (Project Leader) 수입

IEC의 표준개발은 프로젝트 단위로 이루어진다. 본 사업을 통해 개발완료된 IEC 63246-1 프로젝트에 대한 책임은 프로젝트 리더(Project Leader)가 담당하며, 이는 ITU-T에서의 표준 개발 에디터와 동등한 직책으로 간주할 수 있다.

The screenshot shows the ITU-T project page for TA 17. The main title is "TA 17 Multimedia systems and equipment for vehicles". Below the title are navigation tabs: "Scope", "Structure", "Projects / Publications" (selected), "Meetings", and "Collaboration Platform". Under "Projects / Publications", there is a link for "Project: IEC 63246-1:2021 ED1". A "Detail" section follows, containing a table with project information.

Committee	Working Groups	Project Leader	Current Status	Frcst Pub Date	Stability Date
TA 17	<a href="#">PT 63246</a>	Mr Juyoung Park	PPUB	2021-09	2024

#### 4) PoC S/W 구현

##### 가) OCF기반 스마트팜 IoT 기자재 PoC 개발

##### 가-1) [SW등록] 엠큐티티(MQTT)기반 스마트팜 센서/구동기/제어기 노드 연동 프로그램 ('18)

표준적합성, 안정성 등을 확보하기 위해 자체적으로 프로토콜 라이브러리 프로토타입을 파이썬을 이용하여 개발하였다. 아래 그림은 PoC SW의 동작화면으로 복수개의 구동기, 센서가 LCP/SCP로 동작하고 제어되는 모습을 보여준다.

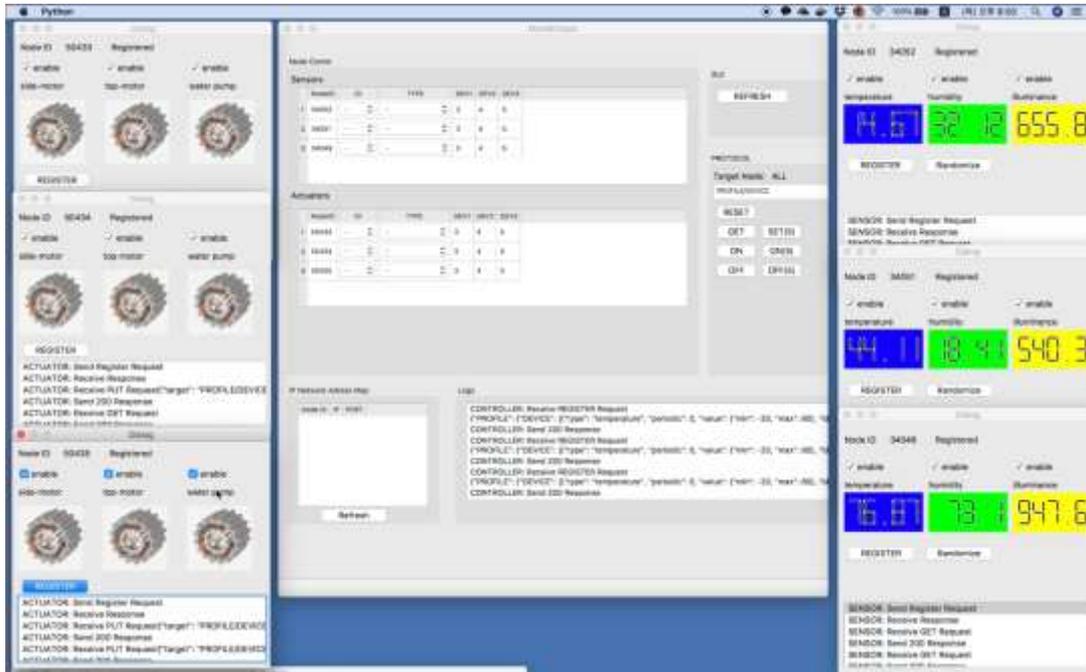


그림 120 - 센서와 구동기 운용 프로그램 GUI

하나의 센서노드에는 복수개의 센서들이 부착될 수 있고, 각각의 디바이스(센서)들은 구분되어 제어될 수 있다. 중간에 위치한 컨트롤러 노드(제어기)는 하위에 부착된 모든 디바이스의 상태를 조회하거나 동작을 제어할 수 있도록 하였다. 또한, 메시지의 송수신 내역을 추적할 수 있도록 하였다. 합체를 개발하기에 앞서 솔리드웍스를 이용한 3D 모델링 작업을 선행하여, 다음과 같이 모델을 완성하였다.

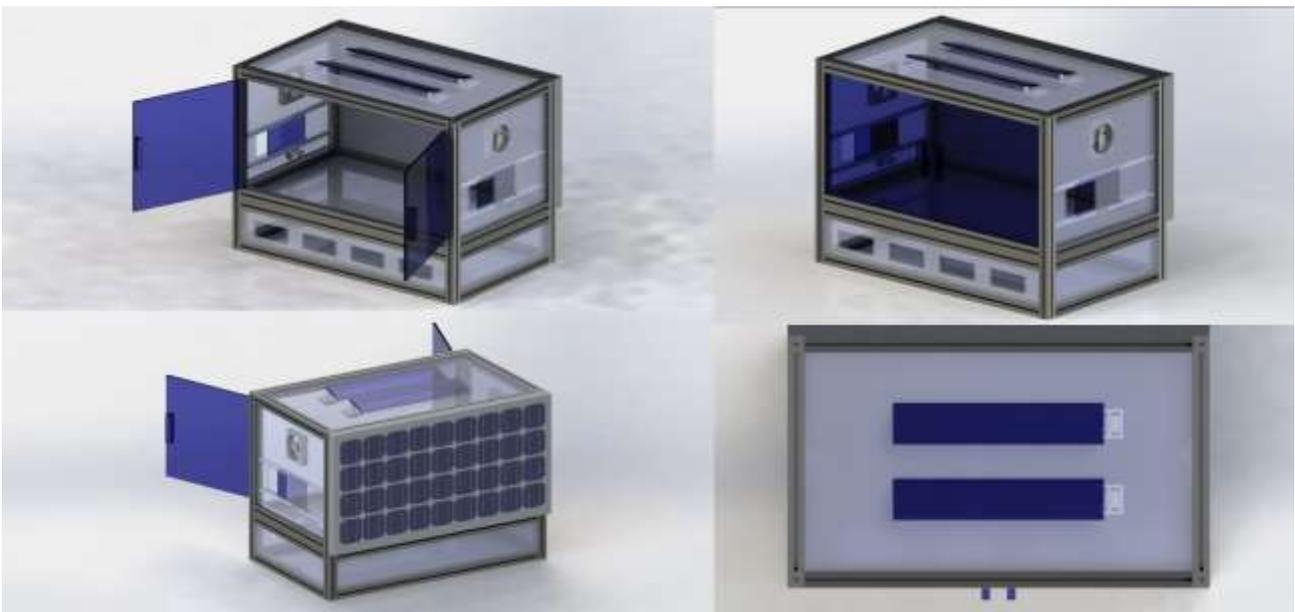


그림 121 - PoC 검증을 위한 스마트온실 기능 시험장치

이렇게 설계된 합체를 기반으로 아래와 같은 시작품을 제작하였으며, 내부에 라즈베리파이, 아두이노 등

을 탑재하고, 내부적으로 SCP와 LCP, IPAL을 접목하여 원활히 동작하는 모형을 제작할 수 있었다. 이 합체를 기반으로 프로토콜 탑재, 동작 제어 시험 등을 통해 표준적합성을 확보할 수 있고, 표준의 안정성을 더욱 높일 수 있는 기반을 마련하였다고 볼 수 있다.

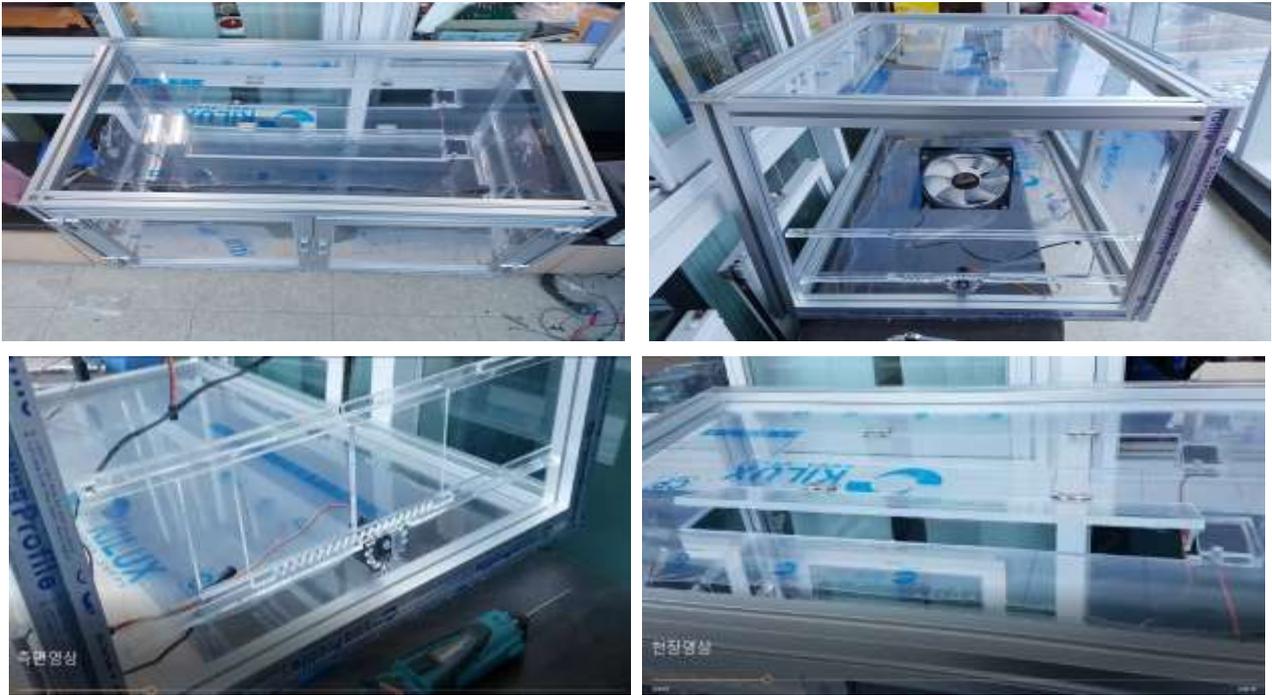


그림 122 - PoC 검증을 위한 스마트온실 기능 시험장치

현재 지원되는 네트워크 인터페이스는 RS485/Modbus, WiFi, ZigBee, LoRa 로, 아직은 안정성 향상을 위해 추가 작업이 필요한 상태이다. 그리고 각종 센서들로부터 수집된 데이터들은 내부 시계열 데이터베이스에 축적하였고, 시각화 프로그램을 이용하여 데이터를 분석할 수 있도록 하였다.

향후 이를 확장하여, 디지털트윈(Digital Twin)의 기능까지 포함시킬 계획이다. 즉, 시설 모형에서 발생하는 일이 직접 사이버 상에 있는 웹 모델에 반영이 되고, 그 역의 경우까지 가능하도록 할 수 있다. 이 과정에서 개발된 PoC 프로그램들은 모두 한국저작권 위원회에 등록되었다.

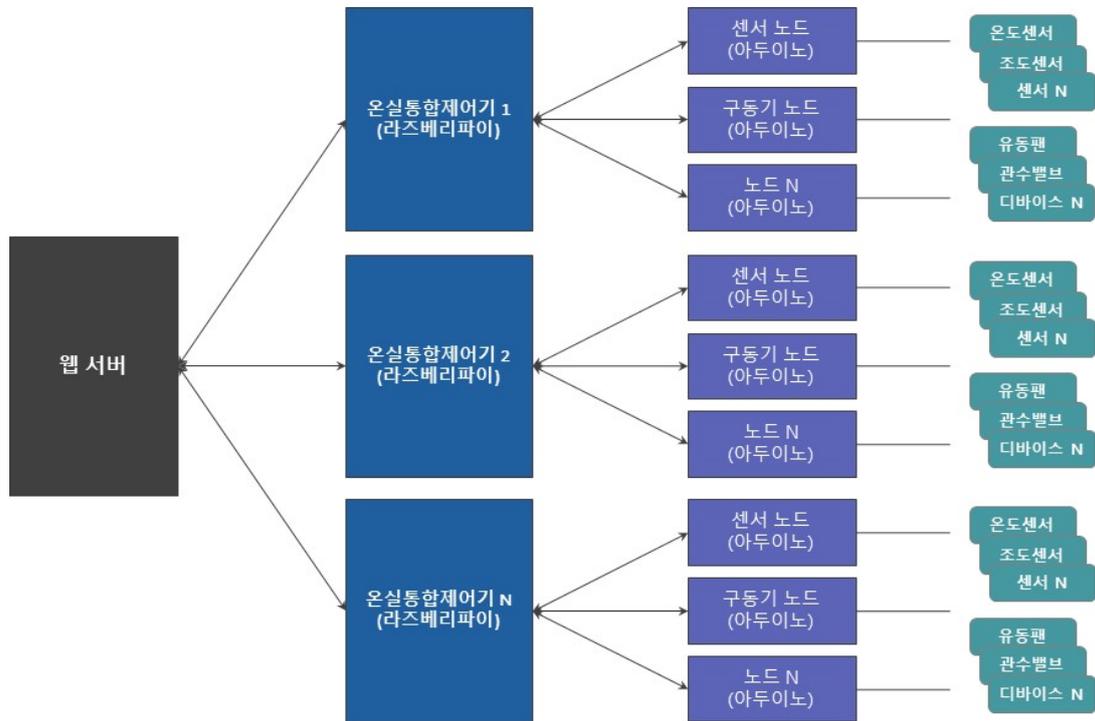


그림 123 - 수집된 데이터 시각화 결과



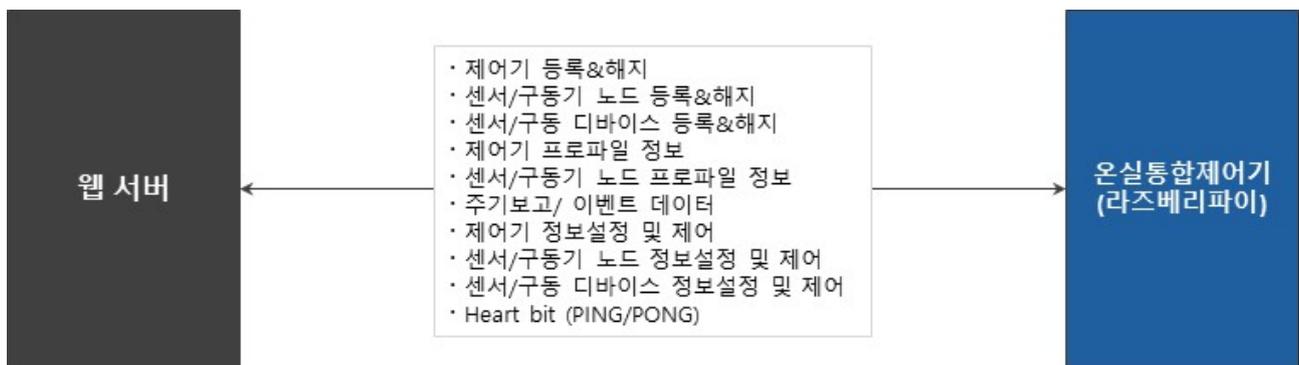
가-2) [SW구현] 경량형 제어프로토콜 기반 스마트온실 통합 서비스 제어 프로그램 ('19)

본 과제에서 추진중인 표준을 검증하기 위하여, 다음과 같이 온실에 설치되는 제어기, 센서/구동기노드 그리고 센서와 구동기들의 집합 환경을 구성하였다.

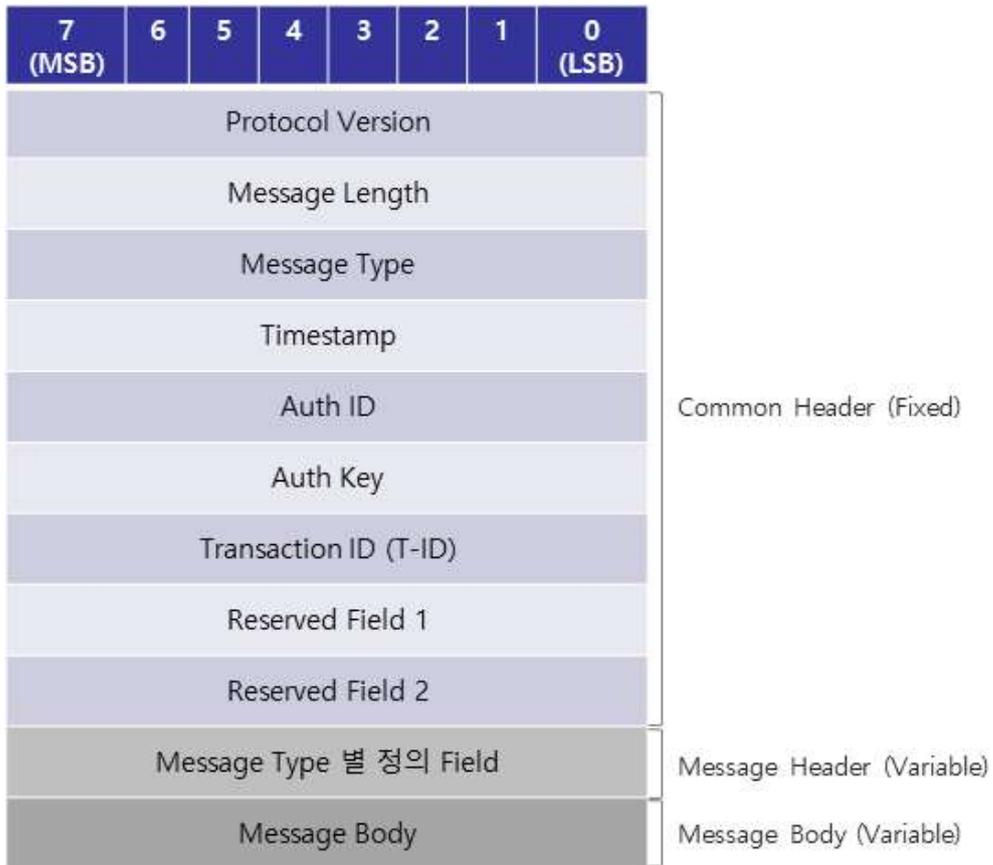


웹서버는 온실에 설치되어 있는 센서/구동기들의 동작 현황이나 생육환경에 대한 데이터를 수집하기 위한 빅데이터 서버로 간주하고 설계하였다.

웹서버와 통합제어기간 통신 내용은 다음과 같다.



웹서버와 통합제어기간 통신프로토콜은 표준으로 정의되어 있지 않지만, 본 PoC 동작 구현 검증을 위해 다음과 같은 통신 메시지 포맷을 이용하여 통신한다.



각 필드의 길이와 데이터 타입 및 사용방법을 아래 표로 정리하였다.

필드 이름	길이 (byte)	자료형	설명
ProtocolVersion	1	Char	프로토콜 버전으로 1~ 255 까지의 범위로 설정한다. 현재는 1 로 설정한다.
Message Length	2	Short	전체 메시지 길이를 나타낸다. (MessageHeader Length + Message Body Length)
Message Type	1	Char	메시지타입(등록/주기보고/제어/이벤트 등)
Timestamp	4	Integer	UTC(1970년 1월1일부터 경과된 시간 - 초 단위)
AuthID	16	Char	통합제어기의 Serial Number MAC Address 값을 사용한다.
Auth Key	16	Char	메시지 인증 키 등록 절차 시 서버에서 할당한다.
TransactionID	4	Integer	서버와 통합제어기간의 트랜잭션을 구분하는 값으로써 요청하는 곳에서 할당하고 응답하는 곳에서 해당 T-ID 값을 그대로 반환한다. ※ 통합제어기 : 0 ~ 99,999 ※ 서버 : 0 ~ 99,999 이외의 값
ReservedField	1	Char	예약 필드(현재 사용하지 않음 - 0x00 으로 설정한다)
ReservedField	1	Char	예약 필드(현재 사용하지 않음 - 0x00 으로 설정한다)
MessageHeader	var	-	MessageType 별 필드 정의 값 (가변 길이)
Message Body	var	-	MessageType 별 실제 Payload 영역 (최대 2,048 bytes/JSON)

통합 서버와 온실통합제어기간 통신은 다음과 같은 기능들을 수행하도록 설계하였다.

Operation	Type Name	Value (Hexa)	Direction	Description
통합 제어기 등록/해지	Controller Registration Request	0x01	C --> S	통합제어기 등록 요청
	ControllerRegistration Response	0x02	S --> C	통합제어기등록 응답
	ControllerDeregistrationRequest	0x03	C --> S	통합제어기해지 요청
	ControllerDeregistration Response	0x04	S --> C	통합제어기해지 응답
센서 노드 등록/해지	Sensor Node Registration Request	0x05	C --> S	센서노드 등록 요청
	Sensor NodeRegistration Response	0x06	S --> C	센서노드 등록 응답
	Sensor NodeDeregistrationRequest	0x07	C --> S	센서노드 해지 요청
	Sensor NodeDeregistration Response	0x08	S --> C	센서노드 해지 응답
구동기 노드 등록/해지	Actuator Node Registration Request	0x09	C --> S	구동기노드 등록 요청
	Actuator NodeRegistration Response	0x0A	S --> C	구동기노드 등록 응답
	Actuator NodeDeregistrationRequest	0x0B	C --> S	구동기노드 해지 요청
	Actuator NodeDeregistration Response	0x0C	S --> C	구동기노드 해지 응답
디바이스 등록/해지	Device Registration Request	0x0D	C --> S	디바이스 등록 요청
	DeviceRegistration Response	0x0E	S --> C	디바이스 등록 응답
	DeviceDeregistrationRequest	0x0F	C --> S	디바이스 해지 요청
	DeviceDeregistration Response	0x10	S --> C	디바이스 해지 응답
프로파일 (Profile) 정보	Profile Request	0x11	C --> S	
	Profile Response	0x12	S --> C	
주기보고	Data Delivery Request	0x13	C --> S	
	Data Delivery Response	0x14	S --> C	
제어	Control Request	0x15	S --> C	
	Control Response	0x16	C --> S	
제어결과	Control Result Request	0x17	C --> S	
	Control Result Response	0x18	S --> C	
Heartbeat (PING/PONG)	Heartbeat Request	0x19	C --> S	
	Heartbeat Response	0x1A	S --> C	
주기보고 (멀티 데이터)	Multi Data Delivery Request	0x1B	C --> S	센서/구동기노드 별 디바이스 데이터를 한번에 전송할 때 사용됨
	Multi Data Delivery Response	0x1C	S --> C	

• Controller Registration Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader (Auth ID 필드에 통합제어기의 MAC 값을 설정한다)				
Name	16	Char	M	이름 (controller_1,controller_2, ...)

• Controller Registration Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader (Auth Key 필드에 서버가 할당한 인증키 값이 설정된다)				
Controller ID	16	Char	M	통합제어기 ID
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Controller Deregistration Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader (Auth ID 필드에 통합제어기의 MAC 값을 설정한다)				
Controller ID	16	Char	M	통합제어기 ID

• Controller Deregistration Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader (Auth Key 필드에 서버가 할당한 인증키 값이 설정된다)				
Controller ID	16	Char	M	통합제어기 ID
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Sensor Node Registration Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
Name	16	Char	M	이름 (sensornode_1, sensornode_2,...)
Description	16	Char	O	현재는 통합제어기와의 통신기술 표시 - wifi, lora, zigbee, rs485

• Sensor Node Registration Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
Sensor Node ID	16	Char	M	센서노드ID
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Sensor Node Deregistration Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
Sensor Node ID	16	Char	M	센서노드ID

• Sensor Node Deregistration Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				

ControllerID	16	Char	M	통합제어기ID
Sensor Node ID	16	Char	M	센서노드ID
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Actuator Node Registration Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
Name	16	Char	M	이름 (actnode_1, actnode_2, ...)
Description	16	Char	O	현재는 통합제어기와의 통신기술 표시 - wifi, lora, zigbee, rs485

• Actuator Node Registration Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
Actuator Node ID	16	Char	M	구동기노드ID
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Actuator Node Deregistration Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
Actuator Node ID	16	Char	M	구동기노드ID

• Actuator Node Deregistration Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
Actuator Node ID	16	Char	M	구동기노드ID
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Device Registration Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID

NodeID	16	Char	M	센서 또는 구동기노드 ID
DeviceType	2	Short	M	디바이스 타입 (8. Device Type표 참조)
DeviceNo	1	Char	M	디바이스 번호 (1 ~ 255) 같은 타입의 디바이스가 여러 개일 때 구분하기 위함
Description	16	Char	0	여러 개의 창 개폐기를 구분하기 위해 표시한다. "up1" - 천창 1 "up2" - 천창 2 "left" -왼쪽 측창 "right" - 오른쪽 측창

• Device Registration Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
NodeID	16	Char	M	센서 또는 구동기노드 ID
DeviceID	16	Char	M	디바이스 ID
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Device Deregistration Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
NodeID	16	Char	M	센서 또는 구동기노드 ID
DeviceID	16	Char	M	디바이스 ID

• Device Deregistration Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID
NodeID	16	Char	M	센서 또는 구동기노드 ID
DeviceID	16	Char	M	디바이스 ID
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Profile Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
Node ID	16	Char	M	Node ID (0x00 : 통합제어기의 Profile, Other : 해당 Node의 Profile)

• Profile Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
Node ID	16	Char	M	Node ID (0x00 : 통합제어기의 Profile, Other : 해당 Node의 Profile)
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)
HeartbeatPeriod	2	Short	M	Heartbeat주기 (minute) ※ NodeProfile일 경우에는 0 이다.
ReportingPeriod	2	Short	M	보고 주기(minute)

• Data Delivery Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
NodeID	16	Char	M	Node ID (0x00 : 통합제어기의 데이터)
Device ID	16	Char	M	DeviceID (0x00: 통합제어기 또는 노드의 데이터)
Report Type	1	Char	M	Report Type - 0x01 : collectdata (periodic data) - 0x02: event data (현재는 사용안함)
Message Body	Variable	Variable	M	Data (최대 2,048 bytes/JSON)

• Data Delivery Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
NodeID	16	Char	M	Node ID (0x00 : 통합제어기의 데이터)
Device ID	16	Char	M	DeviceID (0x00: 통합제어기 또는 노드의 데이터)
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Device Control Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
NodeID	16	Char	M	Node ID (0x00 : 통합제어기의 데이터)

Device ID	16	Char	M	DeviceID (0x00: 통합제어기 또는 노드의 데이터)
ControlType	1	Char	M	Control Type (9. ControlType 표 참조)
Message Body	Variable	Variable	0	Data (최대 2,048 bytes/JSON)

• Device Control Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
NodeID	16	Char	M	Node ID (0x00 : 통합제어기의 데이터)
Device ID	16	Char	M	DeviceID (0x00: 통합제어기 또는 노드의 데이터)
ControlType	1	Char	M	Control Type
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Device Control Result Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
NodeID	16	Char	M	Node ID (0x00 : 통합제어기의 데이터)
Device ID	16	Char	M	DeviceID (0x00: 통합제어기 또는 노드의 데이터)
ControlType	1	Char	M	Control Type
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)
Message Body	Variable	Variable	0	Data (최대 2,048 bytes/JSON)

• Device Control Result Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
NodeID	16	Char	M	Node ID (0x00 : 통합제어기의 데이터)
Device ID	16	Char	M	DeviceID (0x00: 통합제어기 또는 노드의 데이터)
ControlType	1	Char	M	Control Type
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)
Message Body	Variable	Variable	0	Data (최대 2,048 bytes/JSON)

• Device Control Result Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
Controller ID	16	Char	M	통합제어기 ID
NodeID	16	Char	M	Node ID (0x00 : 통합제어기의 데이터)
Device ID	16	Char	M	DeviceID (0x00: 통합제어기 또는 노드의 데이터)
ControlType	1	Char	M	Control Type
ResultCode	1	Char	M	ResultCode (0x00 : Success, Other : Failure Reason)

• Heartbeat Request

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID

•Heartbeat Response

Message Element	길이 (byte)	자료형	M/O	Description
CommonHeader				
ControllerID	16	Char	M	통합제어기ID

모든 응답 메시지에 포함되는 Result Code를 정의한다.

구분	값 (Value)	Description
	0x00	ERROR_OK
MESSAGE	0x10	ERROR_MESSAGE_LENGTH
	0x11	ERROR_MESSAGE_TYPE
	0x12	ERROR_MESSAGE_AUTH_ID
	0x13	ERROR_MESSAGE_AUTH_KEY
	0x14	ERROR_MESSAGE_T_ID
	0x15	ERROR_MESSAGE_CONTROLLER_ID
	0x16	ERROR_MESSAGE_NODE_ID
	0x17	ERROR_MESSAGE_DEVICE_ID
	0x18	ERROR_MESSAGE_REPORT_TYPE
	0x19	ERROR_MESSAGE_DEVICE_TYPE
SERVER Side	0x1A	ERROR_MESSAGE_CONTROL_TYPE
	0x31	ERROR_SERVER_AUTH_ID_NOT_FOUND (ControllerID)
	0x32	ERROR_SERVER_AUTH_KEY_NOT_FOUND
	0x33	ERROR_SERVER_NODE_ID_NOT_FOUND
	0x34	ERROR_SERVER_DEVICE_ID_NOT_FOUND
	0x35	ERROR_SERVER_DATABASE
	0x36	ERROR_SERVER_INTERNAL
CLIENT	0x41	ERROR_CLIENT_AUTH_ID_NOT_FOUND (Controller ID)

Side	0x42	ERROR_SERVER_AUTH_KEY_NOT_FOUND
	0x43	ERROR_CLIENT_NODE_ID_NOT_FOUND
	0x44	ERROR_CLIENT_DEVICE_ID_NOT_FOUND
	0x45	ERROR_CLIENT_DATABASE
	0x46	ERROR_CLIENT_INTERNAL

센서 또는 구동기 디바이스의 타입을 정의한다.

Type	값 (Value)	Description	Comment
air_temperature_sensor	0x0001	온도	한 개의 디바이스가 여러 가지의 측정을 할 수 있으므로 마스크 연산이 가능하도록 값을 정의함
air_humidity_sensor	0x0002	습도	
co2_sensor	0x0004	이산화탄소	
illuminance_sensor	0x0008	조도	
soil_moisture_sensor	0x0010	토양습수율	
ec_sensor	0x0020	EC	
ph_sensor	0x0040	PH	
soil_temperature_sensor	0x0080	지온	
insolation_sensor	0x0100	일사	
wind_direction_sensor	0x0200	풍향	
wind_speed_sensor	0x0400	풍속	
fan_actuator	0x1000	유동팬	
light_led_actuator	0x2000	채광용LED	
watering_valve_actuator	0x3000	관수밸브	
humidifier_actuator	0x4000	가습기	
window_actuator	0x5000	창 개폐기	
light_shielding_film_actuator	0x6000	차광막	

Device Control (Result) 메시지에 사용되는 Control Type 필드에 대하여 정의한다.

Type	값 (Value)	Description
RESET	0x01	통합제어기 또는 센서/구동기노드를 리셋
TURN OFF	0x02	통합제어기 또는 센서/구동기노드를OFF
STATUS CHECK	0x03	상태 요청 (주기보고 데이터를 전송)
PROFILE UPDATE	0x04	통합제어기 또는 센서/구동기노드의 프로파일을 업데이트 (ProfileRequest/Response 수행)
ACTUATOR ON	0x05	구동기 ON
ACTUATOR OFF	0x06	구동기 OFF
ACTUATOR OPEN	0x07	구동기 OPEN (모터형으로 구동되는 디바이스 제어)
ACTUATOR CLOSE	0x08	구동기 CLOSE(모터형으로 구동되는 디바이스 제어)

Message Body (Device Control Request)

ControlType	Message Body(JSON)	Comment
0x07 (ACTUATOR OPEN)	{"position": "70"}	0 ~100
0x08 (ACTUATOR CLOSE)	{"position": "30"}	0 ~100

Message Body (Data Delivery Request) - 1

구분	Message Body(JSON)	Comment
통합제어기	{ "uptime":"2683740", // 시스템 가동시간 - 초 단위 "battery-level":"100" // 배터리 레벨 (라즈베리파이 - 100) }	
센서노드	{ "uptime":"2683740", // 시스템 가동시간 - 초 단위 "battery-level":"100", // 배터리 레벨 (아두이노 - 100) "radio-strength":"7" // 0~10 단계 (Wi-Fi, LoRa, Zigbee 무선 강도) }	
구동기노드	{ "uptime":"2683740", // 시스템 가동시간 - 초 단위 "battery-level":"100", // 배터리 레벨 (아두이노 - 100) "radio-strength":"7" // 0~10 단계 (Wi-Fi, LoRa, Zigbee 무선 강도) }	
센서 디바이스	[ {"id":"xxxx","air_temperature":"20.5", "unit":"celcius"}, {"id":"xxxx","air_humidity":"30", "unit":"percent"}, {"id":"xxxx","co2":"620", "unit":"ppm"}, {"id":"xxxx","illuminance":"10", "unit":"lx"}, {"id":"xxxx","soil_moisture":"28.5", "unit":"percent"}, {"id":"xxxx","ec":"28.5", "unit":"ds/m"}, {"id":"xxxx","ph":"3", "unit":"ph"}, {"id":"xxxx","soil_temperature":"28.5", "unit":"celcius"}, {"id":"xxxx","insolation":"20", "unit":"lx"}, {"id":"xxxx","wind_direction":"45", "unit":"angle"}, {"id":"xxxx","wind_speed":"1.2", "unit":"m/s"} ]	디바이스 타입에 따라 개별 메시지로 전송
구동기 디바이스	[ {"id":"xxxx","fan":"ON","desc":"left"}, {"id":"xxxx","fan":"OFF","desc":"right"}, {"id":"xxxx","light_led":"ON","desc":""}, {"id":"xxxx","watering_valve":"OFF","desc":""}, {"id":"xxxx","humidifier":"ON","desc":""}, {"id":"xxxx","window":"ON","desc":"up1"}, {"id":"xxxx","window":"ON","desc":"up2"}, {"id":"xxxx","window":"ON","desc":"left"}, {"id":"xxxx","window":"ON","desc":"right"}, {"id":"xxxx","ligh"ON","desc":""} ]	개별 메시지로 전송

측정 데이터 단위

센서 종류	Description	Comment
온도	섭씨 (celcius)	
습도	percent	
이산화탄소	ppm	
조도	lx	
토양함수율	percent	
EC	ds/cm	
PH	ph (0 ~ 14)	
지온	celcius	
일사	lx	
풍향	angle (0,45,90,135,180,225,270,315)	0 - 북쪽
풍속	m/s	

본 프로토콜 검증을 위한 PoC 구현에서, 경량형 제어프로토콜 기반 스마트온실 통합 서비스 제어 프로그램을 구현하였으며, 이를 위한 개발환경 구성 및 설치 과정은 다음과 같다.

---

◎ 서버 기본 설치 환경

1. http 2.4 이상
  - mod\_rewrite 필수
2. php 5.4 이상 (7.0이상 권장)
  - curl, mongodb 모듈 필수

[HTTPD]

```
apt-get install apache2
```

[PHP]

```
apt-get install php libapache2-mod-php php-xml php-gd
```

```
apt-get install php-mongodb
```

```
apt-get install php7.0-curl
```

◎ Docker-compose로 기본 framework 설치

1. ICT-Greenhouse/Docker/docker-compose.yml 파일을 설치할 서버에 업로드
2. 볼륨 설정 및 소유자 설정

```
[root@localhost ~]# mkdir -p /srv/grafana
[root@localhost ~]# mkdir -p /srv/influxdb
[root@localhost ~]# mkdir -p /srv/mongo
[root@localhost ~]# chown -R 472:472 /srv/grafana
[root@localhost ~]#
```

3. 업로드한 위치에서 docker-compose up -d 명령어 실행

```
[root@localhost ~]# docker-compose up -d
Creating network "ict-greenhouse" with the default driver
Pulling influxdb (influxdb:latest)...
latest: Pulling from library/influxdb
741437d97401: Pull complete
34d8874714d7: Pull complete
0a108aa26679: Pull complete
980899fb615c: Pull complete
331d7c452a94: Pull complete
966d95f585c3: Pull complete
ce84a9b54514: Pull complete
2da79ec2329d: Pull complete
Digest: sha256:73a55bc3d9e6a8d77baa5a8930baecf5991669ee60444799d21d3c529e1dcff
Status: Downloaded newer image for influxdb:latest
Pulling mongo (mongo:latest)...
latest: Pulling from library/mongo
7b722c1070cd: Pull complete
5fbf74db61f1: Pull complete
ed41cb72e5c9: Pull complete
7ea47a67709e: Pull complete
778aebef6b26: Pull complete
```

```
[root@localhost ~]# docker-compose up -d
Creating network "ict-greenhouse" with the default driver
Pulling influxdb (influxdb:latest)...
latest: Pulling from library/influxdb
741437d97401: Pull complete
34d8874714d7: Pull complete
0a108aa26679: Pull complete
980899fb615c: Pull complete
331d7c452a94: Pull complete
966d95f585c3: Pull complete
ce84a9b54514: Pull complete
2da79ec2329d: Pull complete
Digest: sha256:73a55bc3d9e6a8d777baa5a8930baecf5991669ee60444799d21d3e529e1dcf
Status: Downloaded newer image for influxdb:latest
Pulling mongo (mongo:latest)...
latest: Pulling from library/mongo
7b722c1070cd: Pull complete
5fb74db61f1: Pull complete
ed41cb72e5c9: Pull complete
7ea47a67709e: Pull complete
778a6be6fb26: Pull complete
```

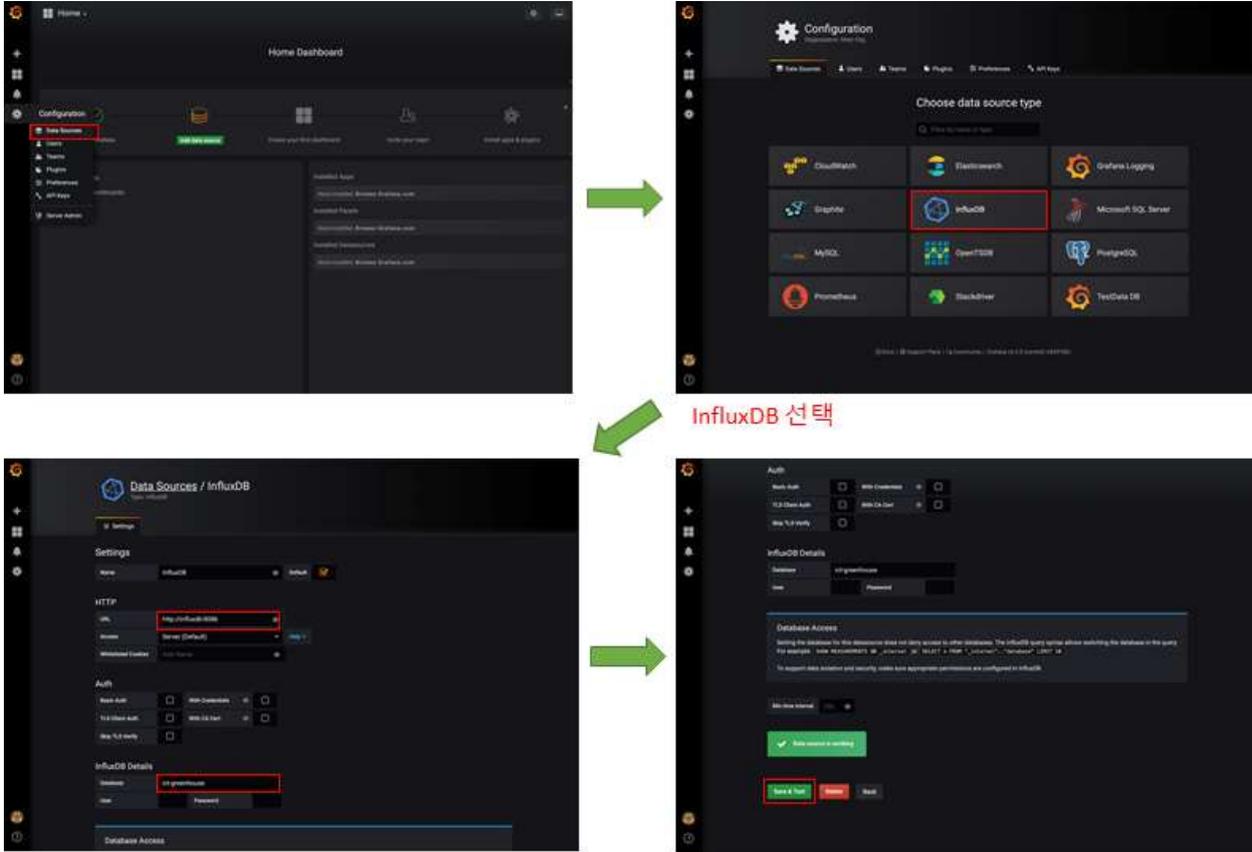
◎ InfluxDB 설정

1. Ict\_greenhouse 데이터베이스 생성
2. 다음 명령어 실행 `curl -XPOST 'http://localhost:8086/query' --data-urlencode 'q=CREATE DATABASE "ict_greenhouse"'`

```
[root@localhost ~]# curl -XPOST 'http://localhost:8086/query' --data-urlencode 'q=CREATE DATABASE "ict_greenhouse"'
{"results":[{"statement_id":0}]}
[root@localhost ~]#
```

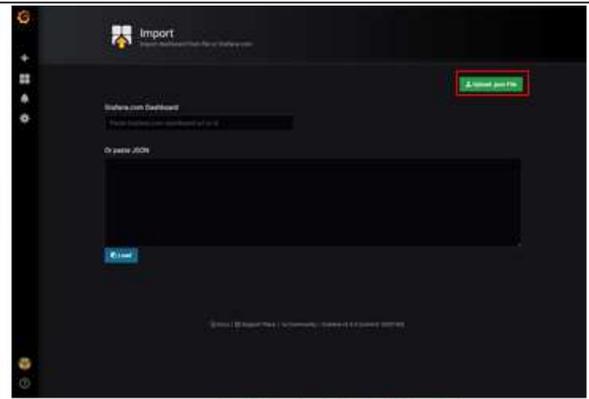
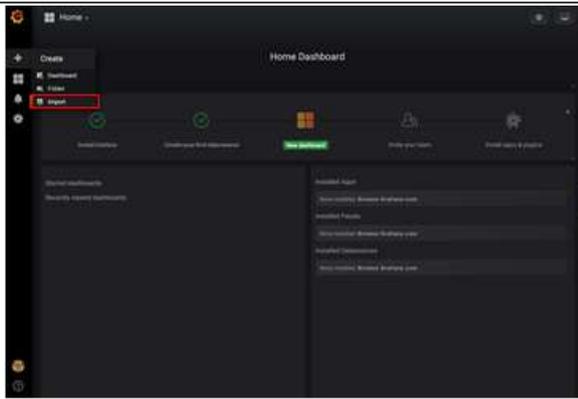
◎ Grafana 설정

1. `http://localhost:3000` 접속 후 Data Sources 설정

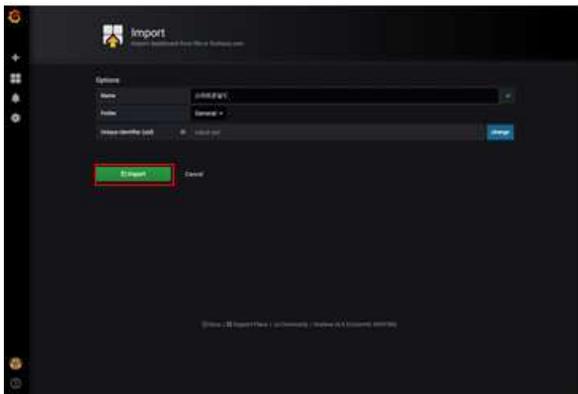


URL: <http://influxdb:8086> Database: `ict_greenhouse`

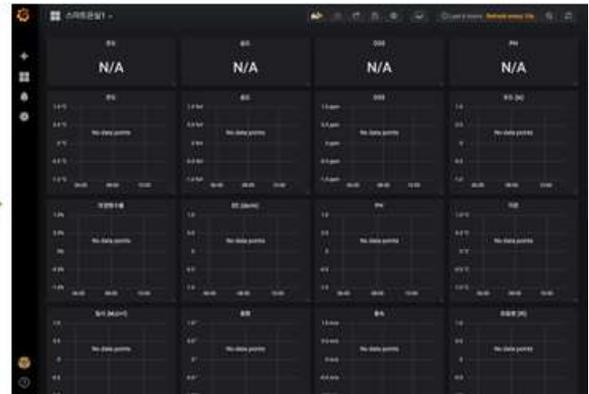
2. Dashboard 설정 (Import: 스마트온실1-1550540937430.json, 스마트온실2-1550541466960.json)



Upload .json File 선택하여 해당 파일 업로드

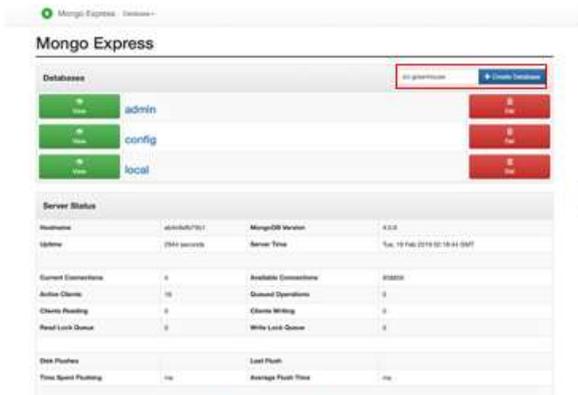


Import 선택하여 저장

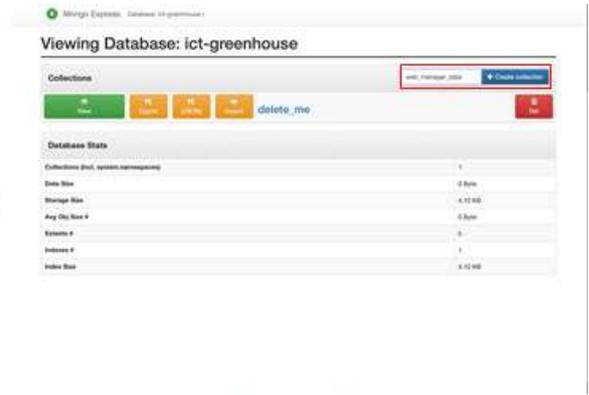


© Mongo-DB 설정

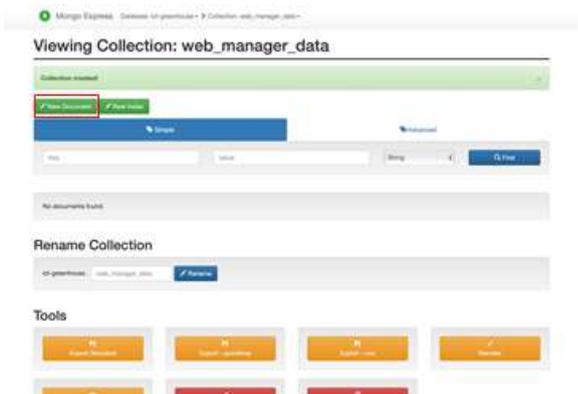
1. http://localhost:8081 접속 후 ict\_greenhouse 데이터베이스, web\_manager\_data 콜렉션, 기본 문서 생성



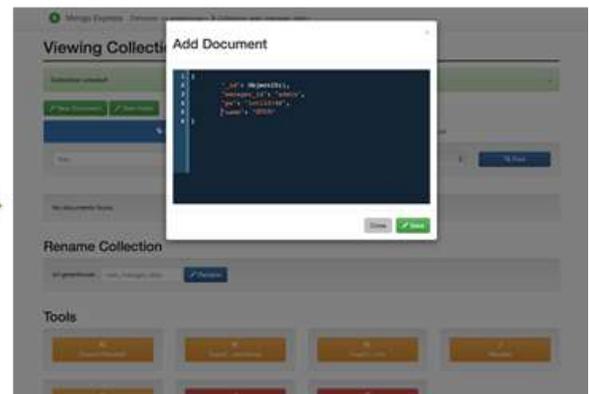
ict\_greenhouse 입력 후 생성



web\_manager\_data 입력 후 생성



New Document 선택



이미지와 같이 입력 후 저장

◎ Socket Server 설치

1. ICT-Greenhouse/Socket-Server/ 하위의 파일을 설치할 서버에 업로드
2. config.php 설정
3. 서버 ip와 포트 입력 (방화벽에서 포트 열려있는지 확인)
4. influxdb, mongodb의 설정값을 변경하지 않았다면 변경할 필요 없음

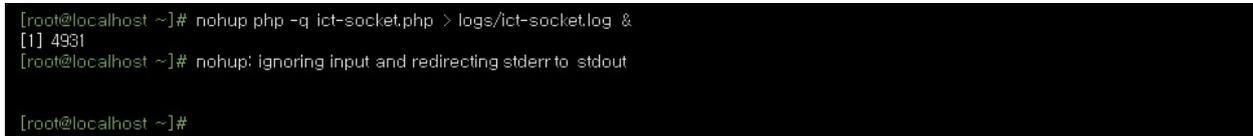


```
config.php (~/ICT-Greenhouse/Socket-Server) - gedit
<?php
defined('IG_ENV') OR exit('No direct script access allowed');

return (object) array(

    'server' => (object) array(
        'ip' => '',
        'port' => 55000,
        'timeout' => NULL
    ),
    'mongodb' => (object) array(
        'host' => 'mongodb://localhost',
        'port' => 27017,
        'username' => '',
        'password' => '',
        'database' => 'ict_greenhouse'
    ),
    'influxdb' => (object) array(
        'host' => 'http://localhost',
        'port' => 8006,
        'username' => '',
        'password' => '',
        'database' => 'ict_greenhouse',
    )
);
```

5. nohup 명령어로 ict-socket.php 실행

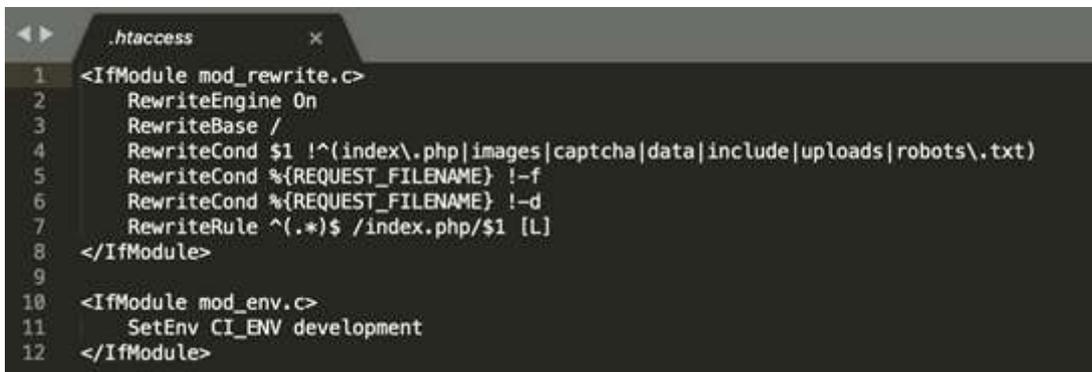


```
[root@localhost ~]# nohup php -q ict-socket.php > logs/ict-socket.log &
[1] 4931
[root@localhost ~]# nohup: ignoring input and redirecting stderr to stdout

[root@localhost ~]#
```

◎ Web 관리자 설치

1. ICT-Greenhouse/Web/ 하위의 파일을 설치할 서버에 업로드
2. .htaccess 설정



```
.htaccess
1 <IfModule mod_rewrite.c>
2 RewriteEngine On
3 RewriteBase /
4 RewriteCond $1 !^(index\.php|images|captcha|data|include|uploads|robots\.txt)
5 RewriteCond %{REQUEST_FILENAME} !-f
6 RewriteCond %{REQUEST_FILENAME} !-d
7 RewriteRule ^(.*)$ /index.php/$1 [L]
8 </IfModule>
9
10 <IfModule mod_env.c>
11 SetEnv CI_ENV development
12 </IfModule>
```

- mod\_rewrite가 설치 되어있어야 함
- 설치위치가 웹 루트 디렉토리일 경우 위 7번 라인을 RewriteRule ^(.\*)\$ /index.php/\$1 [L] 로 수정
- 특정 디렉토리에 설치했을 경우 RewriteRule ^(.\*)\$ /설치디렉토리/index.php/\$1 [L] 로 수정
- 배포환경(에러 메시지를 출력 안함)으로 변경하려면 development -> production 로 변경
- 설치 중 정상 동작 하지 않을 경우 development로 변경 하여 에러 메시지 확인 바람

◎ 전역 변수 설정 (/설치디렉토리/application/config/constants.php)

1. 설치디렉토리 입력 (95번 라인) /var/www/html/test/ 에 설치하였다면 /test/ 만 입력
2. Socket-server 설치한 서버 ip와 port 입력 (96,97번 라인)
3. Grafana 접속 주소 설정 (100라인 부터 http://localhost:3000, 기본 framework 설정을 변경하지 않았다면 변경 불필요)

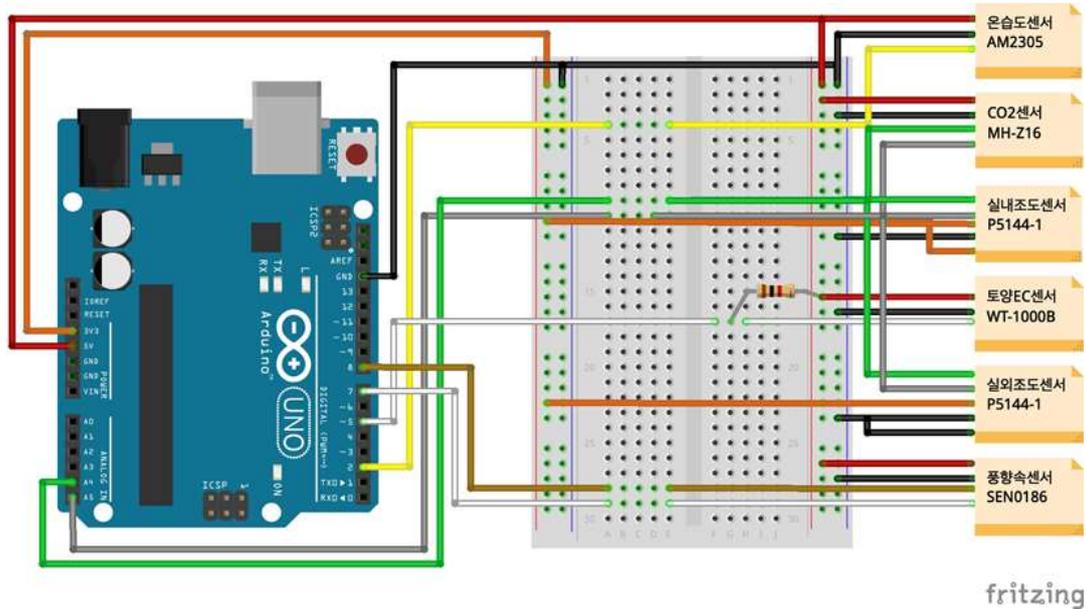
```
constants.php
87
88
89 IG Customized constants
90
91
92 Application customized constants
93
94
95 defined('ROOT_PATH') OR define('ROOT_PATH', '/');
96 defined('IGSOCK_IP') OR define('IGSOCK_IP', '');
97 defined('IGSOCK_PORT') OR define('IGSOCK_PORT', 65000);
98
99 // Dashboard
100 defined('GRAFANA_DASHBOARD1_TEMPERATURE_LATEST') OR define('GRAFANA_DASHBOARD1_TEMPERATURE_LATEST', 'http://localhost:3000/d-solo/HxdVNuymk/
    seumateuonsil1?refresh=10s&orgId=1&panelId=24');
101 defined('GRAFANA_DASHBOARD1_HUMIDITY_LATEST') OR define('GRAFANA_DASHBOARD1_HUMIDITY_LATEST', 'http://localhost:3000/d-solo/HxdVNuymk/
    seumateuonsil1?refresh=10s&orgId=1&panelId=26');
102 defined('GRAFANA_DASHBOARD1_CO2_LATEST') OR define('GRAFANA_DASHBOARD1_CO2_LATEST', 'http://localhost:3000/d-solo/HxdVNuymk/
    seumateuonsil1?refresh=10s&orgId=1&panelId=28');
103 defined('GRAFANA_DASHBOARD1_PH_LATEST') OR define('GRAFANA_DASHBOARD1_PH_LATEST', 'http://localhost:3000/d-solo/HxdVNuymk/
    seumateuonsil1?refresh=10s&orgId=1&panelId=30');
104
105 // Sensor1
106 defined('GRAFANA_SENSOR1_TEMPERATURE') OR define('GRAFANA_SENSOR1_TEMPERATURE', 'http://localhost:3000/d-solo/HxdVNuymk/
    seumateuonsil1?refresh=10s&orgId=1&panelId=2');
107 defined('GRAFANA_SENSOR1_HUMIDITY') OR define('GRAFANA_SENSOR1_HUMIDITY', 'http://localhost:3000/d-solo/HxdVNuymk/
```

◎ DB 설정 (/설치디렉토리/application/config/mongo\_db.php)

1. Mongo-db 환경 설정
2. 기본 framework 설정을 변경하지 않았다면 변경 불필요

```
119 /**
120  * Group name for active connection.
121  * If empty, will be activated group #default.
122  */
123 $config['mongo_db']['active_config_group'] = 'default';
124
125 /**
126  * Connection settings for #default group.
127  */
128 $config['mongo_db']['default'] = [
129     'settings' => [
130         'auth' => FALSE,
131         'debug' => TRUE,
132         'return_as' => 'array',
133         'auto_reset_query' => TRUE
134     ],
135
136     'connection_string' => '',
137
138     'connection' => [
139         'host' => 'localhost',
140         'port' => '27017',
141         'user_name' => '',
142         'user_password' => '',
143         'db_name' => 'ict_greenhouse',
144         'db_options' => []
145     ],
146
147     'driver' => []
148 ];
149
```

경량형 제어프로토콜 기반의 스마트온실 센서노드를 구현하기 위하여 다음과 같은 아두이노 기반의 PoC 하드웨어를 설계하였다. 센서들의 모델과 아두이노 보드와의 결선방식은 다음 그림과 같다.

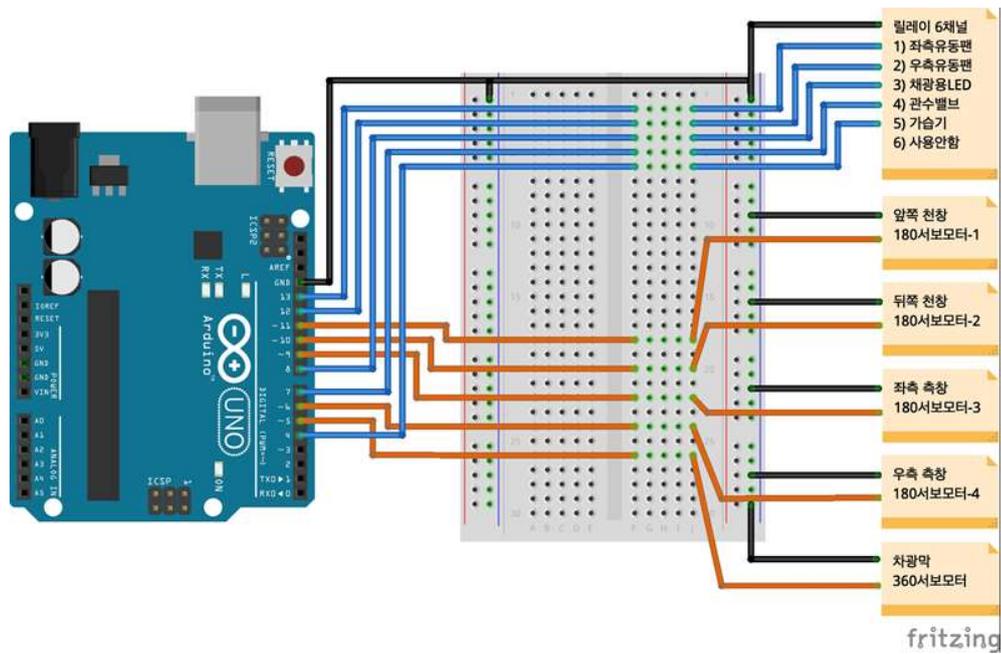


기기의 동작은 다음의 규칙을 따른다.

1. 모든 센서 디바이스의 VCC공급전원은아두이노에서공급됨
2. CO2센서,실내조도센서,실외조도센서의 SDA는 녹색(A4)에 공통으로 연결됨
3. CO2센서,실내조도센서,실외조도센서의 SCL은 회색(A5)에 공통으로 연결됨
4. 실내조도센서의 ADDR은 3.3V에 연결됨
5. 토양EC센서의 TX통신선은1K저항이 5V쪽에 풀업으로연결됨
6. 실외조도센서의 ADDR은 GND에 연결됨
7. 풍향속센서의 TX는 흰색(D7), RX는 갈색(D8)에 연결됨

구동기 또한 아두이노를 통해 구현되었으며, 구동을 위한 장치는 1) 릴레이6채널 (좌측유동팬, 우측유동팬, 채광용LED, 관수밸브, 가습기, 기타), 2) 앞쪽 천창, 3) 뒤쪽 천창, 4) 좌측 측창, 5) 우측 천창 및 6) 차광막을 예물레이트 하였다.

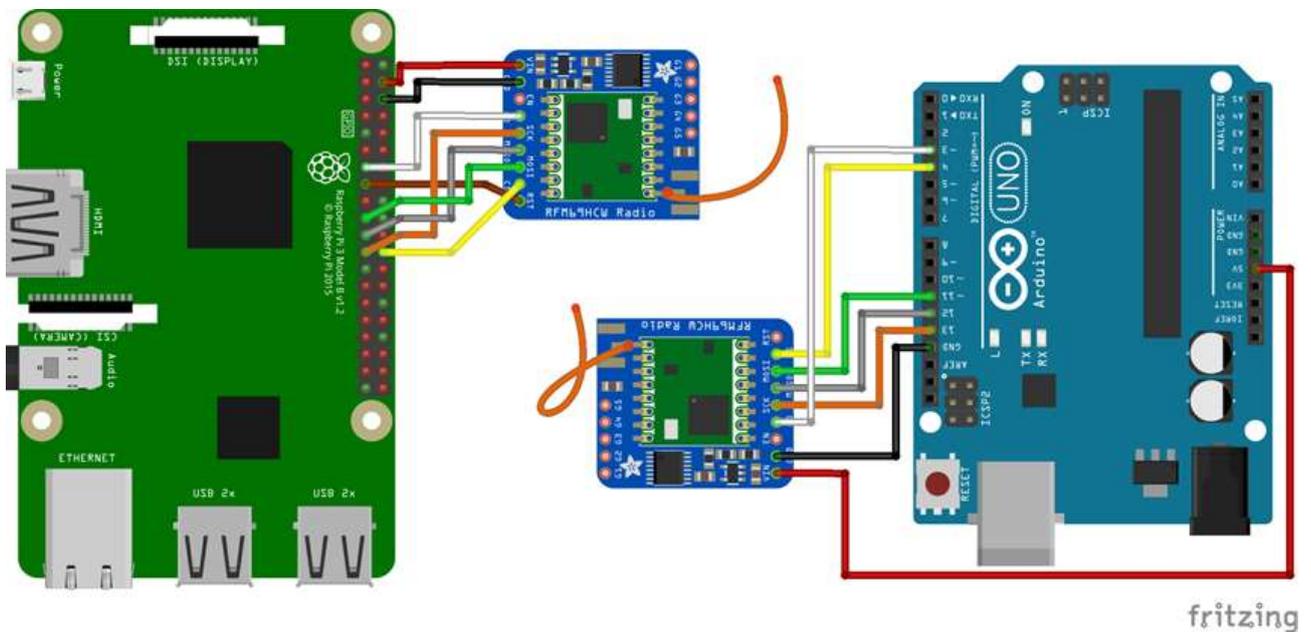
실제 스마트온실에 사용되는 구동기는 아래 하드웨어에서와 같이, 제어 신호가 릴레이와 연결되어 전원을 공급할 수 있을 것으로 가정한다.



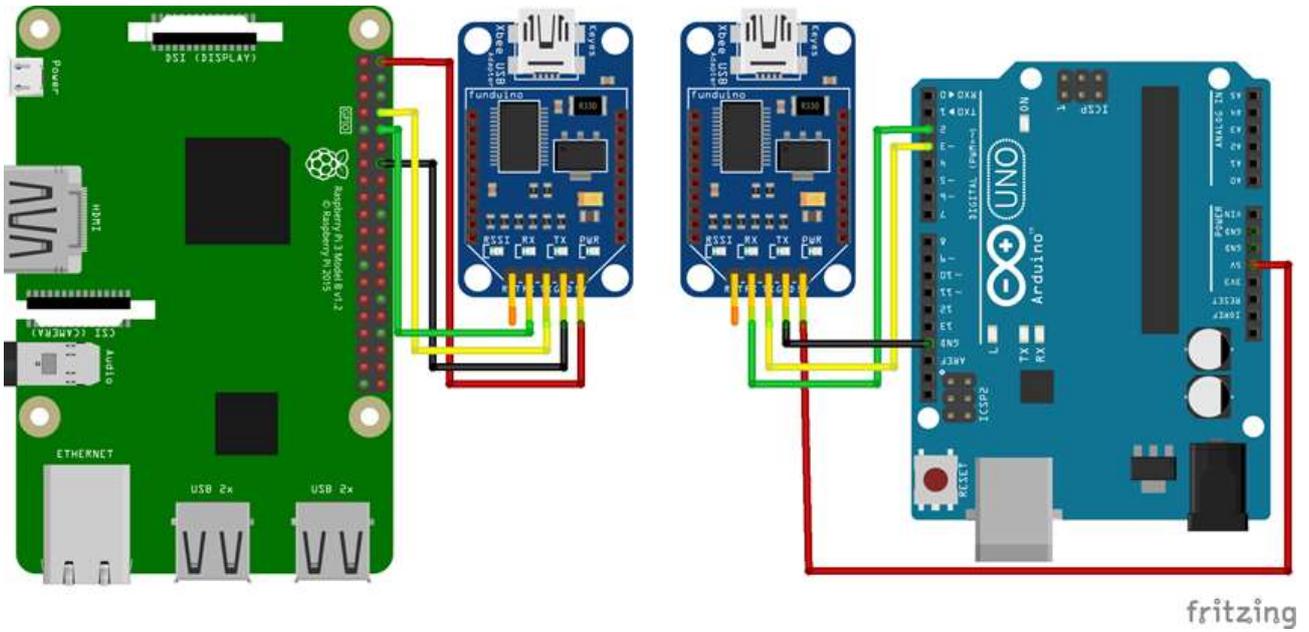
본 구현의 구동기는 다음의 규칙을 따른다.

1. 모든 구동기 디바이스의 VCC 전원공급은외부 별도전원을사용해야함
2. 외부 전원입력되는 GND는 아두이노의GND와 연결되어야 함

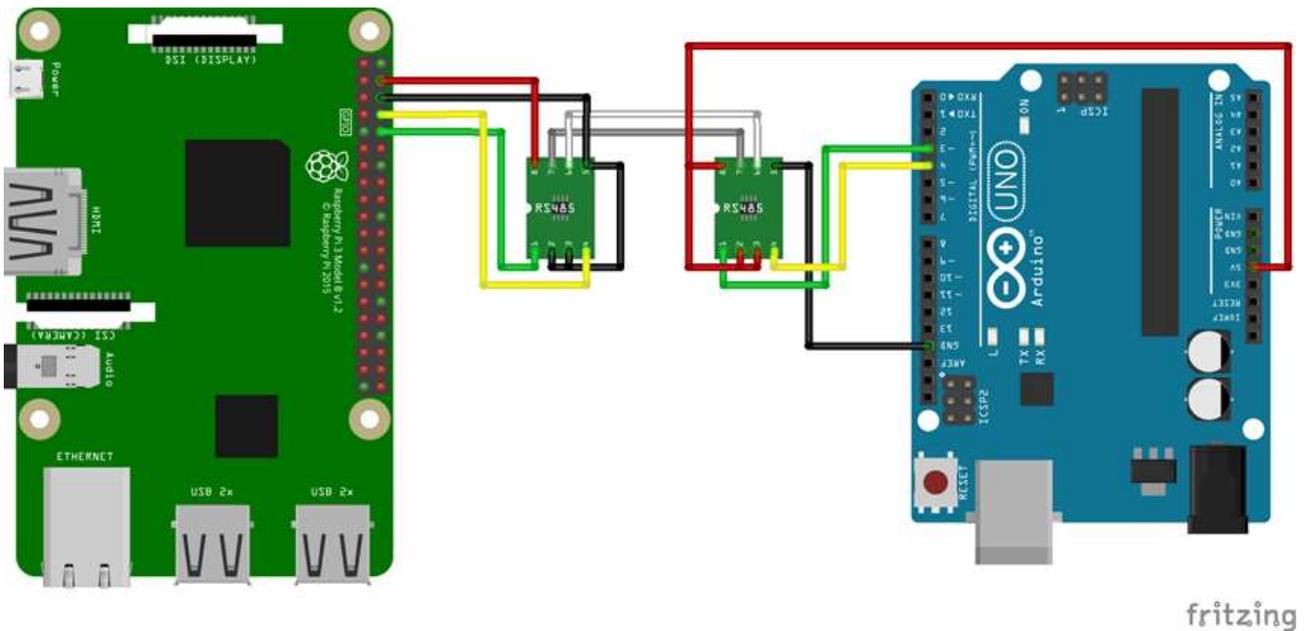
본 프로토콜 검증을 위한 PoC 구현에서, 제어기(라즈베리 보드)와 센서/구동기노드(아두이노 보드)간 LoRa 기반의 통신을 위한 환경을 구축하였다. 이를 위한 LoRa 모듈 연결 방식은 다음과 같다.



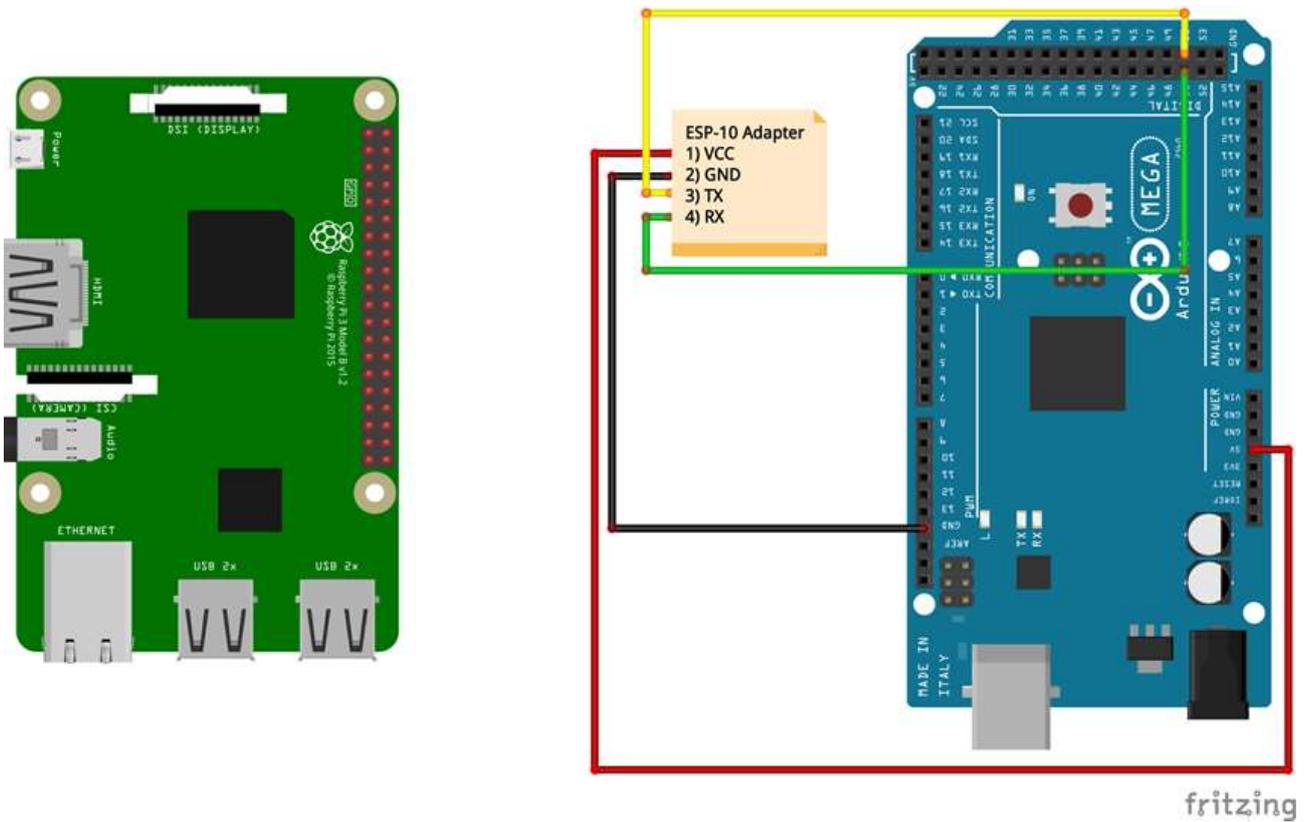
본 프로토콜 검증을 위한 PoC 구현에서, 제어기(라즈베리 보드)와 센서/구동기노드(아두이노 보드)간 Xbee 기반의 통신을 위한 환경을 구축하였다. 이를 위한 Xbee 모듈 연결 방식은 다음과 같다.



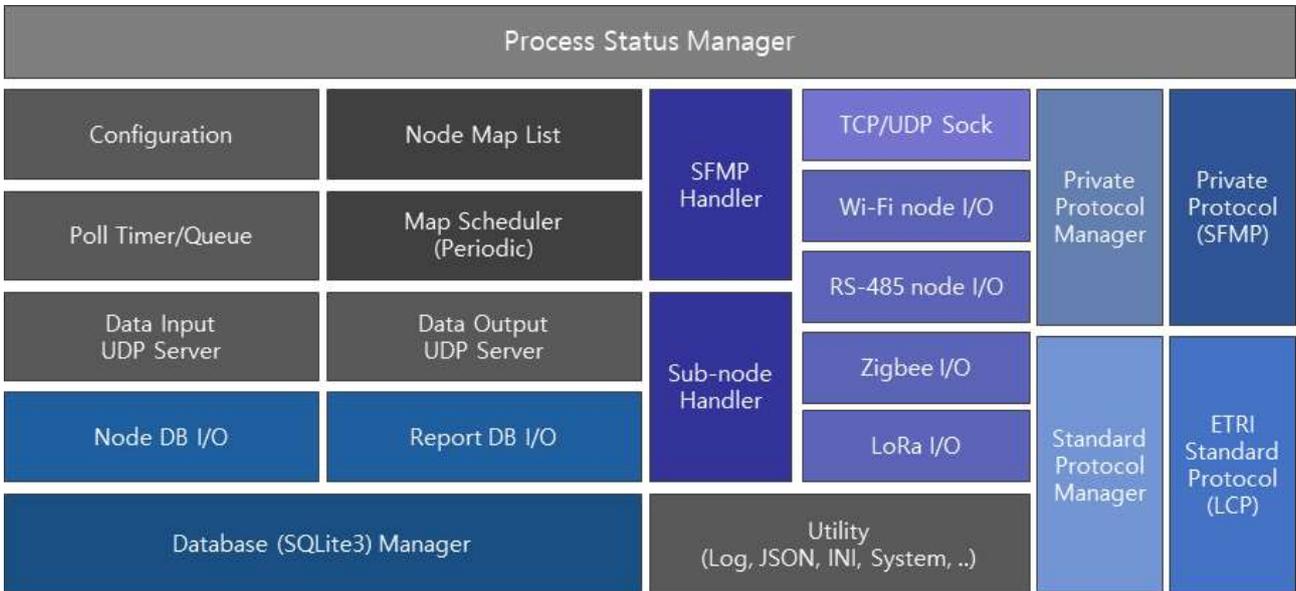
본 프로토콜 검증을 위한 PoC 구현에서, 제어기(라즈베리 보드)와 센서/구동기노드(아두이노 보드)간 RS485 기반의 통신을 위한 환경을 구축하였다. 이를 위한 RS485 모듈 연결 방식은 다음과 같다.



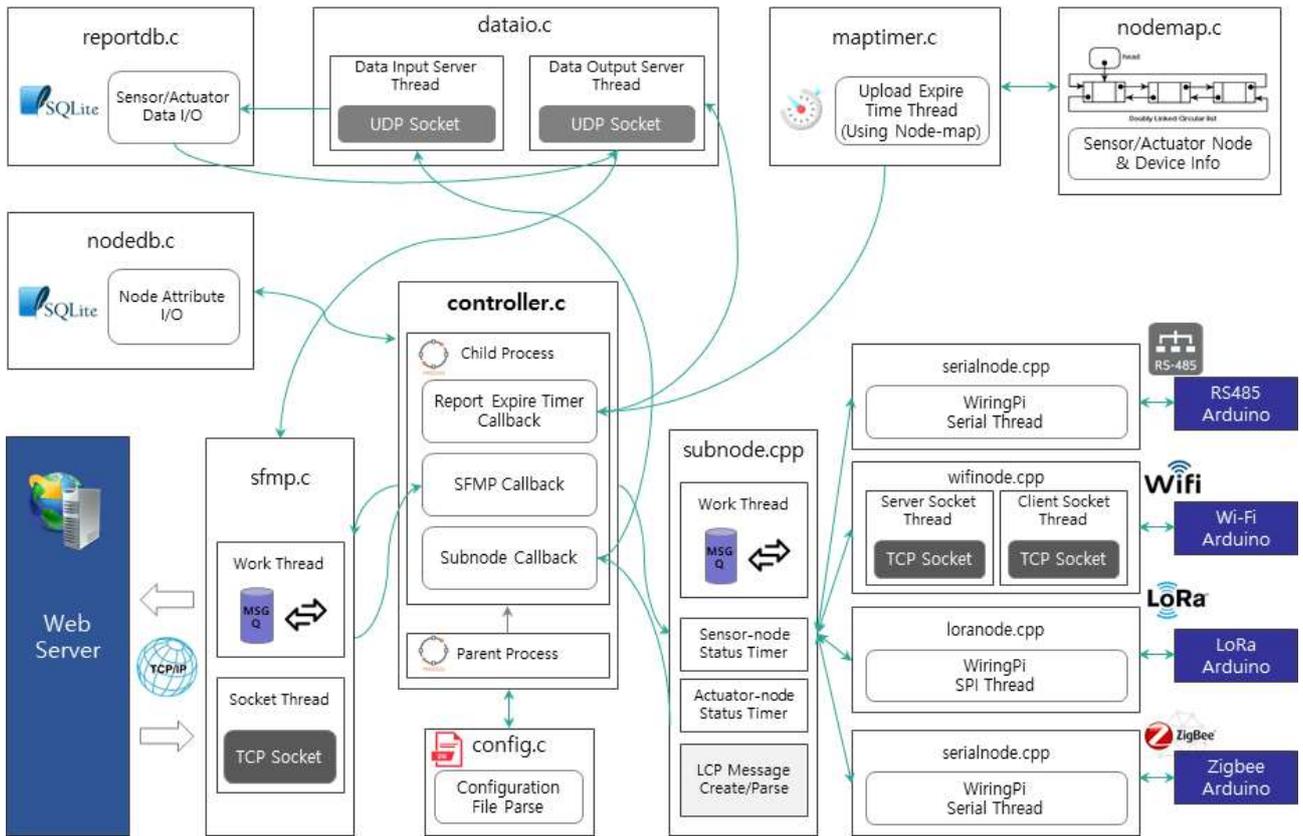
본 프로토콜 검증을 위한 PoC 구현에서, 제어기(라즈베리 보드)와 센서/구동기노드(아두이노 보드)간 WiFi기반의 통신을 위한 환경을 구축하였다. 이를 위한 WiFi모듈 연결 방식은 다음과 같다.



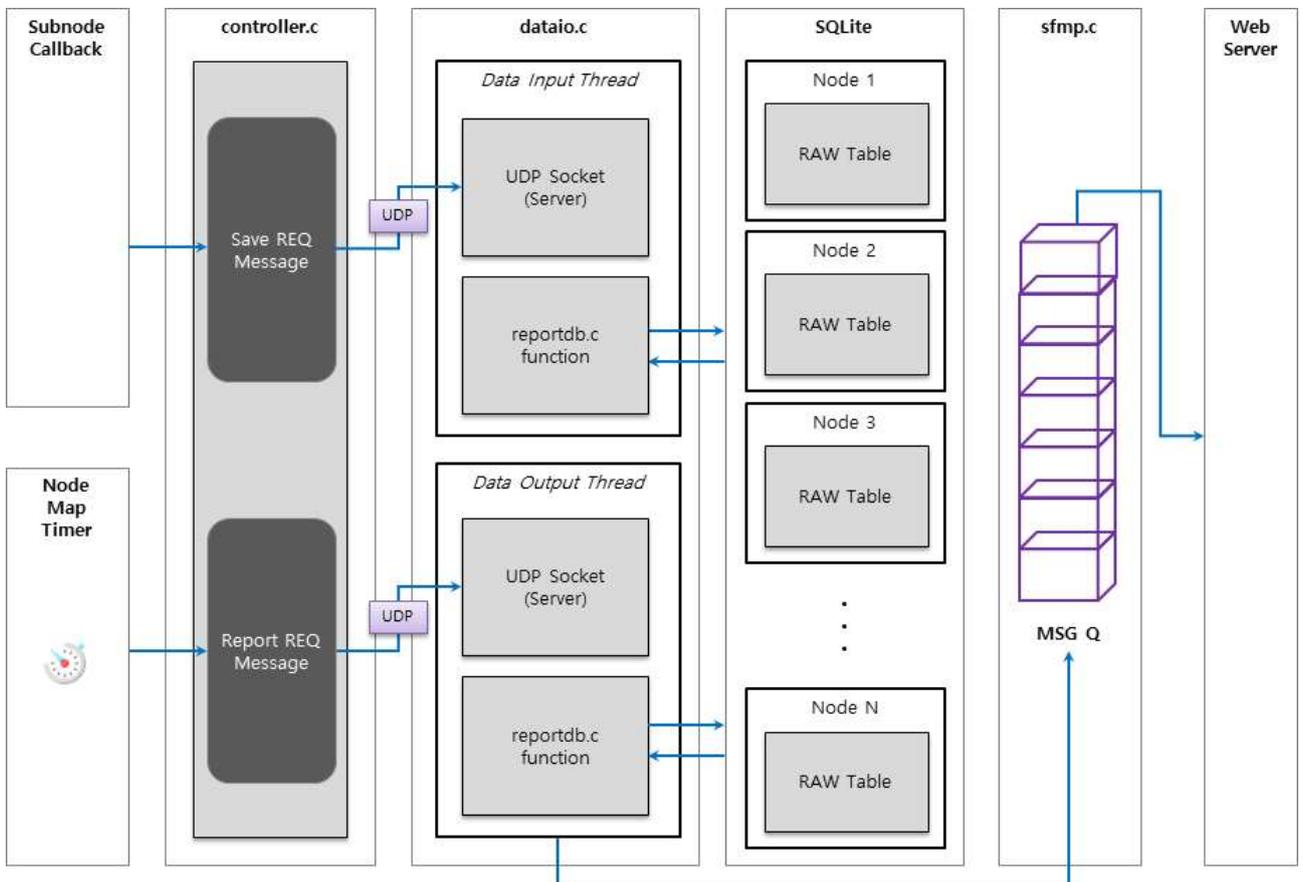
본 프로토콜 검증을 위한 PoC 구현에서, 스마트 온실을 위한 통합제어기 노드는 라즈베리 파이에 구현하였으며, 이를 위한 통합제어기 노드를 구성하는 소프트웨어 모듈 블록은 다음 그림과 같다.



본 프로토콜 검증을 위한 PoC 구현에서, 통합제어기 노드를 라즈베리파이 보드에 구현하였으며, 메인 모듈간 상호 동작은 다음 그림과 같다.



본 프로토콜 검증을 위한 PoC 구현에서, 통합제어기 노드를 라즈베리파이 보드에 구현하였으며, 데이터 저장절차는 다음 그림과 같다.



본 프로토콜 검증을 위한 PoC 구현에서, 통합제어기 노드를 라즈베리파이 보드에 구현하였으며, 통합제어기기의 데이터 저장을 위한 데이터베이스 스키마에서 테이블 정보는 다음과 같다.

테이블 이름	설명
CONTROLLER_INFO_TB	컨트롤러 정보 (프로파일 정보 포함 )
SUB_NODE_LIST_TB	센서 /구동기노드 정보
DEVICE_LIST_TB	센서 /구동 디바이스 정보
NODEINFO_STATUS_TB	노드 데이터베이스의 상태 정보
SUB_NODE_REPORT_LATEST_RAW_TB	센서 /구동기노드의 최근 주기보고 데이터
DEVICE_REPORT_LATEST_RAW_TB	센서 /구동 디바이스의 최근 주기보고 데이터

본 프로토콜 검증을 위한 PoC 구현에서, 통합제어기 노드를 라즈베리파이 보드에 구현하였으며, 통합제어기의 데이터 저장을 위한 데이터베이스 스키마에서 각 테이블의 컬럼 정보는 다음과 같다.

테이블 이름	컬럼 이름	자료형	설명
CONTROLLER_INFO_TB	id	varchar (16)	컨트롤러 ID
	name	varchar (16)	컨트롤러 이름
	heartbeat_period	integer	Heartbeat 주기 (minute)
	report_period	integer	보고 주기 (minute)
	lcp_id	integer	ETRI LCP 에서의 컨트롤러 ID
SUB_NODE_LIST_TB	id	varchar (16)	노드 ID
	name	varchar (16)	노드 이름
	node_type	integer	노드 타입 (0 - sensor, 1 - actuator)
	nw_type	integer	0x01 - wifi, 0x02 - lora, 0x04 - zigbee, 0x08 - rs485
	device_cnt	integer	노드에 연결된 센서 또는 구동 디바이스 개수
	report_period	integer	보고 주기 (minute)
	lcp_id	integer	ETRI LCP 에서의 컨트롤러 ID
DEVICE_LIST_TB	node_id	varchar (16)	디바이스가 연결된 노드 ID
	device_id	varchar (16)	디바이스 ID
	device_type	integer	디바이스 타입
	device_no	integer	디바이스 번호
	description	varchar (16)	설명 (창 개폐기 구분 )
	lcp_id	integer	ETRI LCP 에서의 디바이스 ID
NODEINFO_STATUS_TB	status	integer	0 - NODEINFO_STATUS_NOT_EXIST 1 - NODEINFO_STATUS_CREATING 2 - NODEINFO_STATUS_COMPLETE
SUB_NODE_REPORT_LATEST_RAW_TB	id	varchar (16)	노드 ID
	uptime	varchar (16)	시스템 가동 시간
	batt-level	varchar (1)	배터리 레벨
	rsssi	varchar (1)	무선 강도 ( Wi-Fi, LoRa, Zigbee)
	updatetime	varchar (20)	갱신 시간
DEVICE_REPORT_LATEST_RAW_TB	node_id	varchar (16)	디바이스가 연결된 노드 ID
	device_id	varchar (16)	디바이스 ID
	value	varchar (16)	데이터 값
	updatetime	varchar (20)	갱신 시간

본 프로토콜 검증을 위한 PoC 구현에서, 통합제어기 노드를 라즈베리파이 보드에 구현하였으며, 이를 위한 개발환경 구성 및 설치 과정은 다음과 같다.

◎ 개발환경 구성

1. SQLite 설치
  - apt-get install sqlite3 libsqlite3-dev
2. WiringPi 라이브러리 설치
  - <https://tutorials-raspberrypi.com/installing-wiringpi-and-pin-allocation/>
3. WiFi 무선 네트워크
  - Raspberry Pi 3의 경우 기본적으로 탑재되어 있음

◎ 소스 컴파일 및 설정

1. 특정 디렉터리에 소스 파일을 복사
2. common.h 파일의 PROG\_BIN\_HOME\_DIR\_NAME 수정 -> 실행파일인 “controller” 경로 설정
3. 소스 root 디렉터리에서 \$ make
4. config.ini 파일 수정

◎ 자동실행 설정 (루트 권한)

1. \$ sudo su
2. /etc/rc.local 파일에 다음 내용 추가 (경로는 소스 디렉터리에 맞게 수정 필요)  
/home/pi/smartfarm\_controller/controller &

◎ DB(SQLite) 및 로그 파일

1. DB 디렉터리 : 소스 root 디렉터리에 “db” 디렉터리가 생성되어 저장됨 (\*.s3db)
2. 로그 디렉터리 : 소스 root 디렉터리에 “log” 디렉터리가 생성되어 저장됨 (\*.txt)

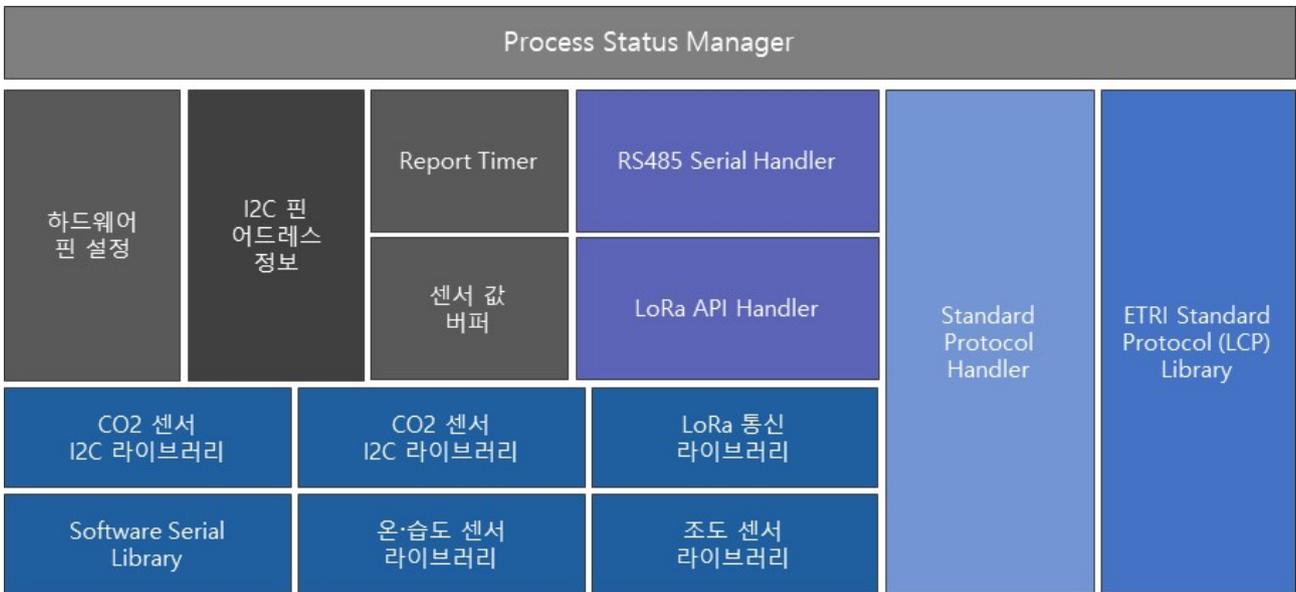
◎ Raspberry Pi 접속 정보

Wi-Fi SSID : Rasp-AP와 Rasp-AP1234 (실제 설치 환경에 따라 다름)  
보안 비밀번호 : raspberry

◎ SSH 또는 FTP 접속 정보

방법 1 : Raspberry Pi에 해당 SSID로 연결된 상태에서 192.168.0.1 호스트 주소로 접근 (실제 설치 환경에 따라 다름)  
방법 2 : Raspberry Pi의 유선 로컬 네트워크에 할당된 호스트 주소로 접근

본 프로토콜 검증을 위한 PoC 구현에서, 센서 및 구동기노드를 아두이노 보드에 구현하였으며, 이를 위한 센서노드의 모듈 블록 다이어그램은 다음과 같다.



본 프로토콜 검증을 위한 PoC 구현에서, 센서 및 구동기노드를 아두이노 보드에 구현하였으며, 이를 위한 구동기노드의 모듈 블록 다이어그램은 다음과 같다.



본 프로토콜 검증을 위한 PoC 구현에서, 센서 및 구동기노드를 아두이노 보드에 구현하였으며, 이를 위한 센서 및 구동기노드 소프트웨어의 소스 및 라이브러리 구성은 다음과 같다.

구분	파일 이름	설명	비고
Arduino Source	smartfarm_sensor.ino	각종 센서 연동 구현 RS485, LoRa 통신 모듈 연동 구현 LCP 프로토콜을 통한 통신기능 구현 주기적으로 센싱된 값을 제어기로 전달하도록 처리	
	smartfarm_actuator.ino	각종 구동기 연동 구현 (ON/OFF 제어 ,모터제어 ) Zigbee, WiFi 통신 모듈 연동 구현 LCP 프로토콜을 통한 통신기능 구현 제어기의 요청에 따라 구동기의 상태값을 전달하도록 처리 제어기의 명령 수신시 구동기를 제어하도록 처리	
Library	ETRI_Smartfarm_LCP.zip	LCP 프로토콜 라이브러리	
	ITEADLIB_Arduino_WeeESP8266-master.zip	ESP-01 WiFi 통신 모듈 라이브러리	
	RadioHead-master.zip	RFM9x LoRa 통신 모듈 라이브러리	
	Adafruit_Sensor-master.zip	AM2305 온습도 센서 라이브러리	
	DHT-sensor-library-master.zip	AM2305 온습도 센서 라이브러리	
	BH1750FVI.zip	P5144-1 조도 센서 라이브러리	
	NDIRZ16-master.zip	MH-Z16 CO2 센서 라이브러리	
	UART_Bridge-master.zip	MH-Z16 CO2 센서 라이브러리	

본 프로토콜 검증을 위한 PoC 구현에서, 센서 및 구동기노드를 아두이노 보드에 구현하였으며, 이를 위한 개발환경 구성 및 설치 과정은 다음과 같다.

```
smartfarm_sensor_v1.0 | 아두이노 1.8.5
smartfarm_sensor_v1.0 §
#include <DeviceID.h>
#include <jsmn.h>
#include <json-maker.h>
#include <Lcp.h>
#include <LcpMsgOperation.h>
#include <LcpNodeControl.h>
#include <LcpNodeInfo.h>
#include <Util.h>

#define CONTROLLER_ID      1000
#define NODE_ID_SENSOR    100
#define SENSOR_CNT        11

//통신 모듈 선택(업로드 전에 해당보드에 맞춰 둘 중 하나만 선택할것)
#define RS485              0
#define LORA                1

//라이브러리
#include <SPI.h>
```

가-3) [SW등록] 아이오티(IoT) 센서 기반 스마트팜 모니터링 S/W ('20)

본 연구와 관련하여 다음과 같이 3건의 소프트웨어 등록이 이루어졌다.

프로그램명	등록일	등록번호
아이오티(IoT) 센서 기반 스마트팜 모니터링 웹유저인터페이스(UI)	2020.07.14.	C-2020-023223
아이오티(IoT) 센서기반 스마트팜 노드 프로그램	2020.07.14.	C-2020-023222
아이오티(IoT) 센서 기반 스마트팜 노드 모니터링 프로그램	2020.07.14.	C-2020-023220

스마트팜에 설치되는 IoT 장치들은 빈번한 급수와 높은 온도·습도 등의 농작물 양육 환경 등으로 인해 전달되는 상당량의 데이터에 오류와 누락이 발생한다. 현재의 스마트팜 관련 장치 제조사들은 이러한 문제점을 해결하기 위하여 자신만의 노하우로 자신만을 위한 제품을 개발하고 있는 실정(자급자족형)이며, 국내 80%에 달하는 비닐하우스의 경우, RS485 시리얼 케이블을 통한 제어 환경에 머물고 있다. 그러나 최근 발전해가는 IoT 기술과 보조를 맞추어, 스마트팜용 IoT 장비 제조사들이 자신만을 위한 제품이 아니라 개방형 제품을 제조한다면 규모의 경제를 만들 수 있다.

현재 IoT 관련 사실 표준화기구인, OneM2M, OCF 등지에서는 IoT 장비간 상호 연동을 위한 표준을 개발 중이지만 그 주요 대상은 스마트제조, 스마트홈 및 스마트헬스에 국한되어 있다. 미래의 스마트팜에서는 현재의 RS485 시리얼 통신 방식으로는 그 적용 제한이 있을 것으로 예상되는 바, 해외 표준화 동향에 따라, 즉, OneM2M 및 OCF에서 개발되는 표준에 대응할 수 있는 스마트팜 기술을 확보와 함께 PoC를 통한 기술 검증을 병행하는 것이 매우 중요하다. 따라서, 일반 스마트팜 제조사 혹은 설치 업체들이 쉽게 IoT 장치(센서 및 구동기)를 이용하고 운영 관리할 수 있도록 하는 기술적인 해법이 요구되고 있다.

본 연구는 OCF 표준 IoTivity 아키텍처를 기반으로 국제적으로 규격화된 스마트팜 기술에 대한 개념검증을 위한 스마트팜 IoT 기자재 테스트베드를 개발하는 것을 목적으로 한다. 본 테스트베드에서 구현되는 IoTivity 기반 스마트팜 제어 플랫폼은 농작물 재배를 하기 위한 스마트팜 환경에 적용함으로써 실제 서비스가 개시되기 이전에 해당 서비스에 대한 평가 및 기능 검증 수행이 가능하기 때문에 많은 활용이 가능할 것으로 전망된다.

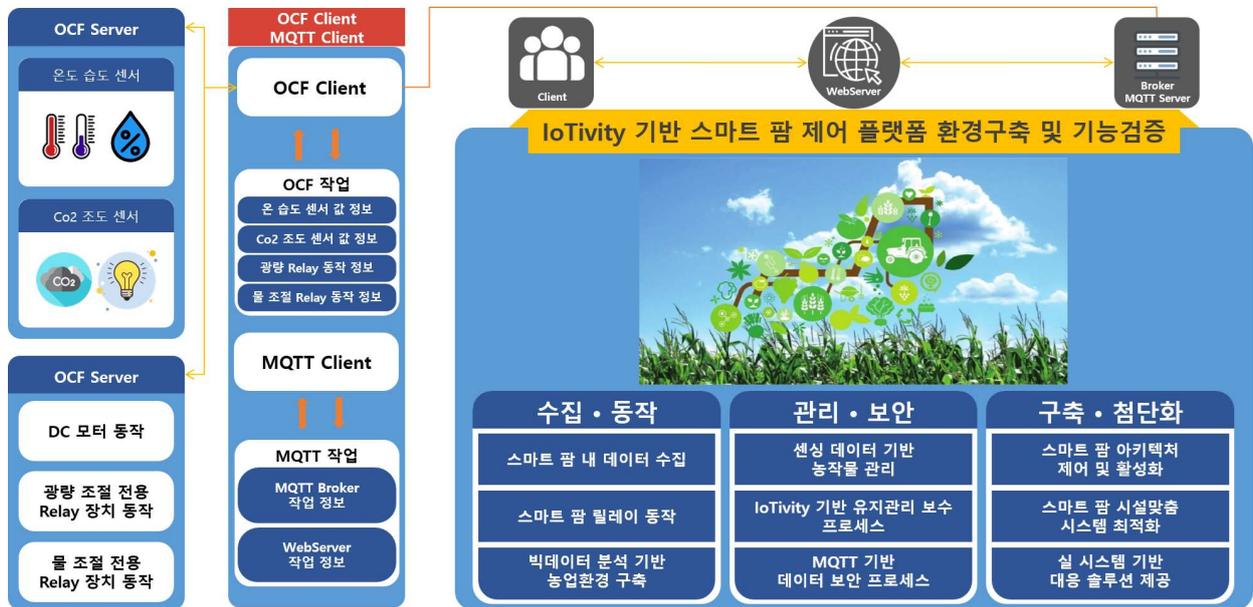


그림1 - IoTivity 기반 스마트팜 제어 플랫폼

(가) 요구사항

사용자 기능 요구사항은 다음과 같다.

- OCF Server는 Sensor의 값을 받아오며 Relay 기능을 수행해야 한다.
- OCF Client는 OCF Server를 통제할 수 있는 명령과 값을 받아오는 역할을 수행해야 한다.
- MQTT Server 역할을 하는 Broker는 MQTT Client와 OCF Client 역할을 수행하는 장치에 통제할 수 있는 명령과 값을 받아 오는 역할을 수행해야 한다.
- MQTTServer 역할을 하는 Broker는 MQTT Client와 WebServer 역할을 수행하는 장치에 통제할 수

있는 명령과 값을 받아 오는 역할을 수행해야 한다.

- WebServer 장치에서는 사용자(Client)의 명령과 결과를 보여주며 MQTT 통제할 수 있는 명령과 값을 받아오는 역할을 수행해야 한다.

성능 효율성(Performance Efficiency) 관련 사용자 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 스마트팜 내 데이터 전송 기능을 제공해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 DB 검색 질의에 대해 빠른 시간 내에 결과를 화면에 보여줘야 한다.

사용성(Usability) 관련 사용자 비기능 요구사항은 다음과 같다.

- 사용자가 지식 데이터를 쉽게 접근 및 등록 할수 있는 인터페이스를 제공해야 한다.

유지보수성(Maintainability) 관련 사용자 비기능 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 주기적으로 모든 자동화된 단위 테스트를 실행해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 다른 시스템에서 연동하여 실행 가능하도록 모듈화하여 제공해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 외부 시스템과 연동 인터페이스를 제공해야 한다.

보안성(Security) 관련 사용자 비기능 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 유효한 사용자만 해당 페이지를 접근할 수 있어야 한다.

신뢰성(Reliability) 관련 사용자 비기능 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 100개의 정보를 추출하면 90개는 사용자 키워드에 부합해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 100번의 통화 연결을 시도하면 95번은 성공해야 한다.

시스템 기능 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 사용자 반응과 스마트팜 구조를 반영하여 상호작용 정확도를 확립해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템의 OCF Server에서는 온도, 습도, 조도, Co2 센서의 값을 인식해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템의 OCF Server에서는 Relay를 통해 물 공급 및 광량 조절 기능을 수행해야 한다.
- MQTT Server는 MQTT Client의 CRUD+OBSERVE 요청을 통해 각 센서의 값을 전송해야 한다.
- Web Server는 MQTT Server에 Subscribe를 하여 MQTT Client 역할을 수행해야 한다.
- Web Server는 Client에게 OCF Client/Server, MQTT Client/Server의 정보를 제공해야 한다.

성능 효율성(Performance Efficiency) 관련 시스템 비기능 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 스마트팜 센서 값 데이터를 80% 이상의 정확도로 수집해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 DB 검색 질의중 80% 이상이 1초 이내에 결과를 화면에 보여줘야 한다.

사용성(Usability) 관련 시스템 비기능 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 모든 메뉴에 쉽게 접근하고 이동할 수 있도록 구성된 사용자 웹 인터페이스를 제공해야 한다.

유지보수성(Maintainability) 관련 시스템 비기능 요구사항은 다음과 같다.

- OCF Client 및 Server는 기능별로 모듈화 하여 컴포넌트 형태로 제공해야한다.
- MQTT Client 및 Server는 기능별로 모듈화 하여 컴포넌트 형태로 제공해야한다.

사용자 인터페이스 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템에서는 대쉬보드를 통해 사용자 정보를 제공해야 한다.

시스템 인터페이스 요구사항은 다음과 같다.

- 스마트팜 제어 플랫폼 시스템은 IoTivity 기반 서비스간 공개 데이터 연계를 위한 Periodic 및 On-demand 정보 수집과 DB 연동을 위한 TCP/IP 소켓 인터페이스를 제공해야 한다.

소프트웨어 인터페이스 요구사항은 다음과 같다.

- 스마트팜 제어 플랫폼 시스템은 IoTivity 기반 데이터 센서 값 정보 전달을 위한 COAP, MQTT, HTTP 인터페이스를 제공해야 한다.

하드웨어 인터페이스 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 동작 및 센서데이터 연동을 위한 유 무선 Ethernet 인터페이스를 제공해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 센서 간의 연동을 위한 범용적 하드웨어 인터페이스를 제공해야 한다.

통신인터페이스 요구사항은 다음과 같다.

- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 MQTT Client로부터 MQTT Server로 전송되는 데이터는 MQTT 프로토콜을 제공해야 한다.
- IoTivity 기반 스마트팜 제어 플랫폼 시스템은 OCF Client로부터 OCF Server로 전송되는 데이터는 IoTivity 기반 프로토콜을 제공해야 한다.

#### (나) 시스템 개요

IoTivity의 제어를 위한 환경을 구축하기 위해 각 연계과정을 위한 구성을 만들고 전체를 관장하는 프로그램에 의해 동작한다. 전체 구성의 기본 요소는 제어부의 MQTT (Server/ Client), OCF (Server/Client), 온 습도 센서 등으로 구성하였다. 각 부분의 구성은 다음 구조와 같다.

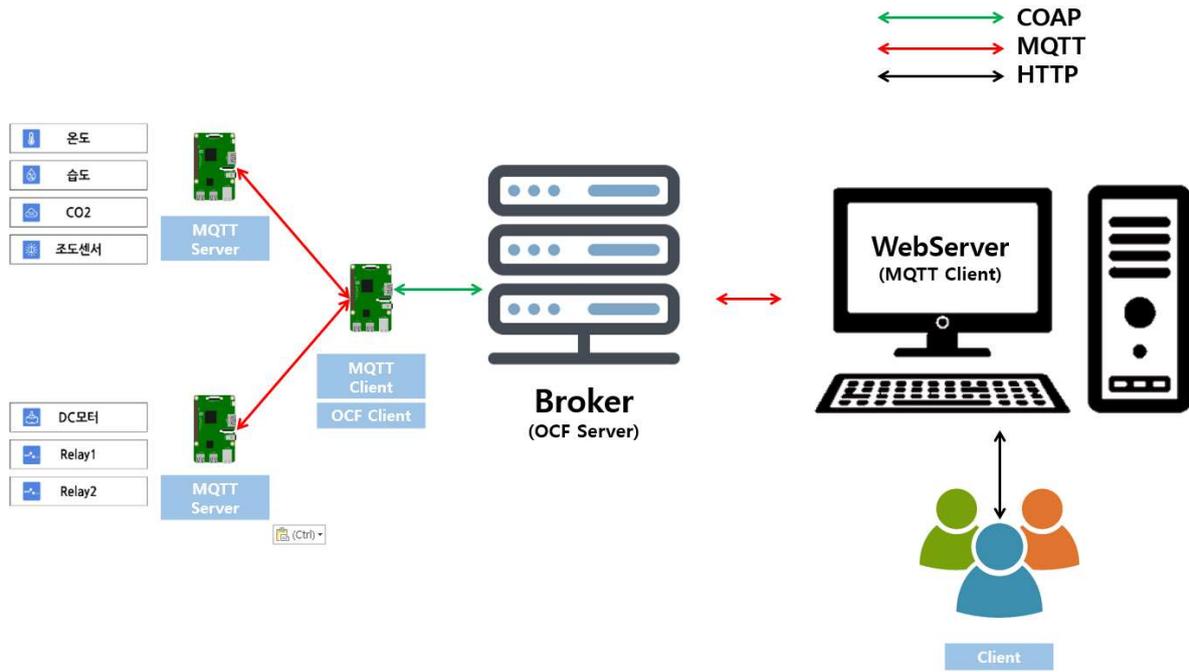


그림 2 - 전체 시스템 구성

각 구성의 기능은 다음과 같다.

1. MQTT Server(라즈베리파이)

동작을 수행 및 센서값을 저장하기 위한 장치로써 MQTT에 의해 실행된다.

- 설정 및 상태 표시를 위해 SPI에 연결되어있는 아래의 장치를 제어한다.
- 센서를 사용하여 현재 상태의 온습도 및 CO2, 조도를 읽는다.
- MQTT Client를 통해 Relay를 통제한다.

2. MQTT Client(라즈베리파이)

사용자가 시스템을 사용하기 위해 명령을 제공한다.

- MQTT Server 제어 하에 상태 표시를 하고 설정을 위한 수단을 제공한다.
- MQTT Server에 DC모터 및 Relay 작업을 통제한다.

3. 센서

센서를 통해 실험 상태에 관한 변인 및 각종 정보를 수집한다.

- 조도, CO<sub>2</sub>, 온도, 습도를 측정하여 통신을 통해 MQTT Server로 전송한다.
- 실험 상태의 측정을 위해 실험장 내부에 센서를 달았다.

4. OCF Client(라즈베리파이)

센서로부터 수집된 데이터를 전달하기 위해 OCF 규격으로 통신을 한다.

- MQTT Server 측에서 전달받은 값들을 전달받아 데이터 값을 저장한다.
- 전달 받은 값들을 Broker 서버인 OCF Server에 전달한다.

5. OCF Server(Broker)

OCF Client로 전달받은 데이터 값을 전달받아, MQTT Client(Web Server)에 전달한다.

- OCF Client로 전달받은 데이터 값을 OCF 규격으로 통신한다.
- 저장된 데이터 값들을 MQTT Client로 전달한다.

## 6. MQTT Client(Web Server)

OCF Server로부터 전달된 데이터를 Client에 제공한다.

- OCF Server 측에서 전달받은 데이터 값을 전달받아 저장한다.
- 전달받은 값들을 Client에게 직접적으로 제공하며, 쉽게 모니터링 및 조작을 할 수 있는 수단으로 사용한다.

스마트팜 제어를 위한 OCF 시스템은 다음과 같은 구조의 인터페이스 사용하여 실행된다.

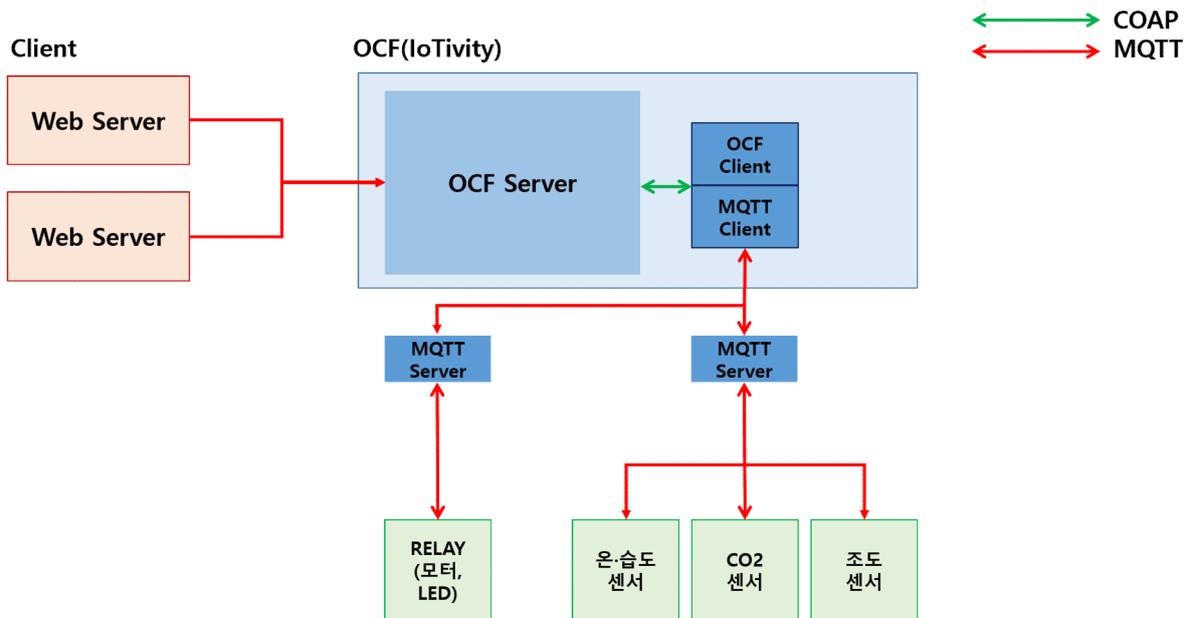


그림 3 - 시스템 제어 구성

사용자 UI는 Web Server의 기능을 활용하여 MQTT 통신으로 제어 명령에 따라 UI 화면이 표출되고. 시스템 상태는 MQTT 출력의 데이터에 의해 그리고 각종 시스템 설정은 입출력 메카니즘에 의해 입력된다. UI에 입력된 설정값은 MQTT 통신을 통해 OCF Server에 전달되어 관리된다. 온도와 습도는 센서 모듈의 통신에 의해 수신되고 이 수신된 스마트팜의 환경 데이터에 따라 정해진 알고리즘에 따라 물 공급 모터 및 LED 조명을 제어하여 온습도 관리가 행해진다

### (다) 개발환경

#### 1) 하드웨어

하드웨어는 라즈베리파이 기반위에 제어 프로그램의 실행에 의해 전체가 동작해야 한다. 라즈베리파이 재단의 Model 3 B+를 사용하였다. 본 하드웨어를 기반으로 OCF 통신, MQTT 통신과 포트를 사용하여 외부 입출력 장치를 제어한다.

본 시스템에서 필수적인 모듈의 기능은 다음과 같다.

- GPIO : 센서와 라즈베리파이 간의 통신을 단말 내부의 GPIO를 사용하여 SPI 통신을 제공하고 모터 제어 신호를 출력한다. 내부의 메모리를 이용하여 프로그램 실행 코드를 실행하고 상태제어 한다.
- PWR IN : 안드로이드 5핀 충전 케이블을 통해 본 단말의 전력을 공급한다. 이를 통한 전력 공급으로 GPIO에 VCC를 제공해준다.

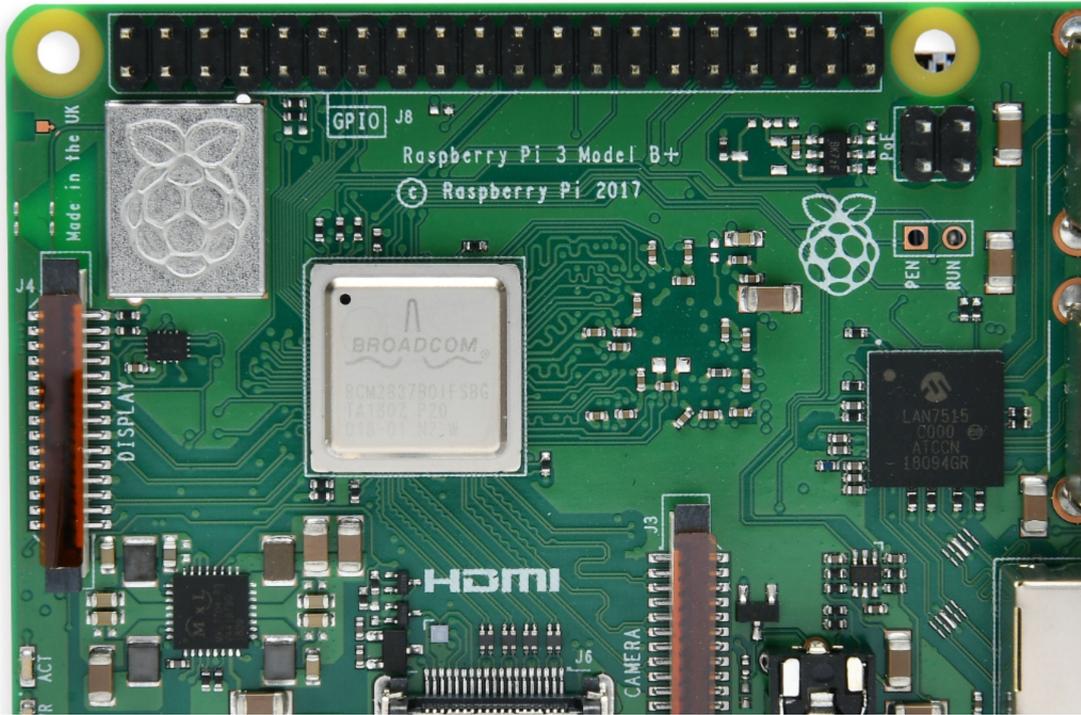


그림 4 - UART 출력에 485 통신 칩을 사용

## 2) 소프트웨어 개발 도구

사용한 라즈베리파이는 ST-LINK를 사용하여 MCU 내부에 FLASH에 코드를 쓴다. 또한 이를 통해 디버깅을 제공한다. C언어 기반의 제어 프로그램은 IDE로 IAR Embebed Workbench ARM을 사용하였다.

### ■ GPIO 단자와 세부 기능 명세서

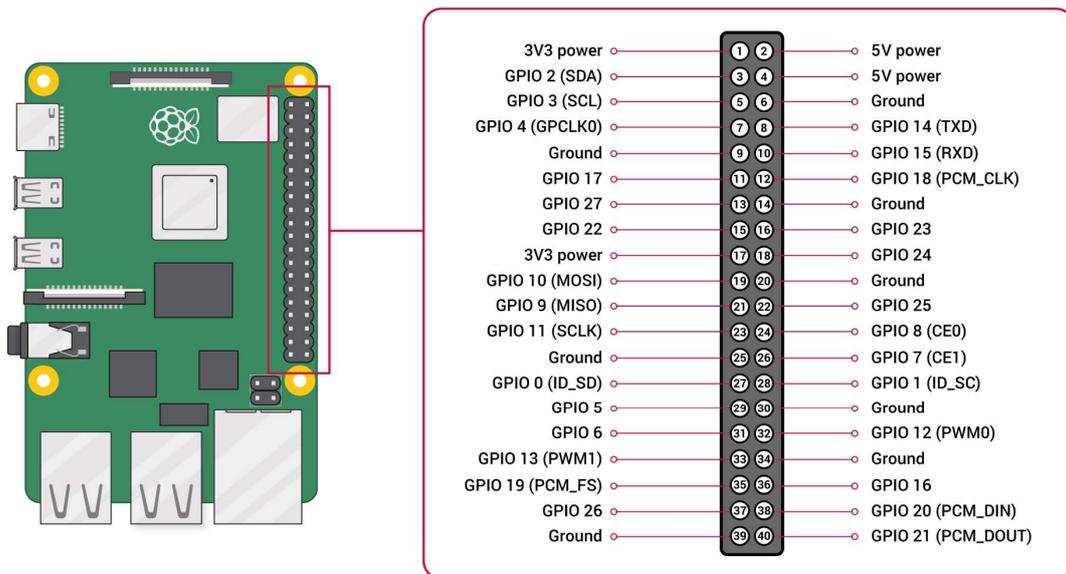


그림 5 - GPIO 단자 모습

대표 센서로 Relay는 그림과 같이 VC, GND, GPIO21 신호를 사용한다.

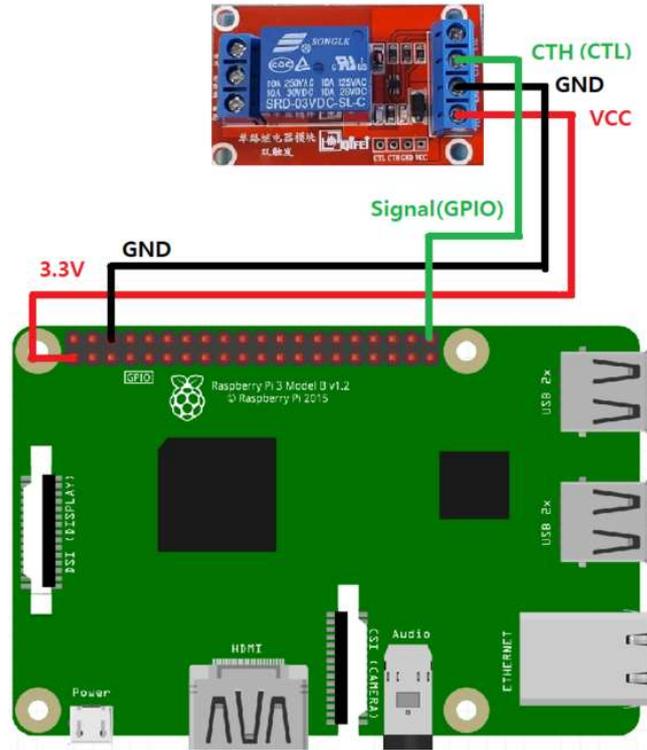


그림 6 - GPIO와 Relay 모듈 연결 방법

■ 프로그램 개발 IDE

DHT22 센서는 공개된 라이브러리를 사용하여 센서를 작동시켜야 한다. 그래서 맨 처음 센서를 작동하기 전 미리 라이브러리를 다운로드 받아야 한다. 맨 처음 라즈베리파이의 터미널 창을 열어서 다음 명령어를 입력한다.

- 
1. cd ..
  2. git clone https://github.com/adafruit/Adafruit\_Python\_DHT.git
  3. cd Adafruit\_Python\_DHT
  4. sudo apt-get update
  5. sudo apt-get install build-essential python-dev python-openssl
  6. sudo python setup.py install
-

```

pi@raspberrypi:~$ git clone https://github.com/adafruit/Adafruit_Python_DHT.git
Cloning into 'Adafruit_Python_DHT'...
remote: Enumerating objects: 325, done.
remote: Total 325 (delta 0), reused 0 (delta 0), pack-reused 325
Receiving objects: 100% (325/325), 98.35 KiB | 0 bytes/s, done.
Resolving deltas: 100% (176/176), done.
Checking connectivity... done.
pi@raspberrypi:~$ cd Adafruit_Python_DHT/
pi@raspberrypi:~/Adafruit_Python_DHT$ sudo apt-get update
Get:1 http://archive.raspberrypi.org jessie InRelease [22.9 kB]
Get:2 http://mirrordirector.raspbian.org jessie InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org jessie/main armhf Packages [170 kB]
Hit http://mirrordirector.raspbian.org jessie/main armhf Packages
Hit http://mirrordirector.raspbian.org jessie/contrib armhf Packages
Hit http://mirrordirector.raspbian.org jessie/non-free armhf Packages
Hit http://mirrordirector.raspbian.org jessie/rpi armhf Packages
Get:4 http://archive.raspberrypi.org jessie/ui armhf Packages [58.9 kB]
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
Fetched 267 kB in 33s (8,055 B/s)
Reading package lists... Done
pi@raspberrypi:~/Adafruit_Python_DHT$ sudo apt-get install build-essential python-dev python-openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version.
The following extra packages will be installed:
  libpython-dev libpython2.7-dev python-cffi python-cryptography python-ply python-pycparser python2.7-dev
Suggested packages:
  python-cryptography-doc python-cryptography-vectors python-openssl-doc python-openssl-dbg python-ply-doc
The following NEW packages will be installed:
  python-openssl
Setting up python-openssl (0.14-1) ...
pi@raspberrypi:~/Adafruit_Python_DHT$ sudo python setup.py install
running install
running bdist_egg

```

그림 7 - 라즈베리파이에서의 DHT22 라이브러리 설치 방법

개발 도구 내 컴파일러, 링커, 라이브러리 그리고 디버깅 방법을 제공한다. 파일을 생성하고 사용하는 센서별로 앞과 동일하게 라이브러리를 선정하면 코드 내에서 호출할 수 있다. 관련된 라이브러리 헤더가 설정되고 'Thonny'를 작동하면 코드를 다음처럼 작성할 수 있다.

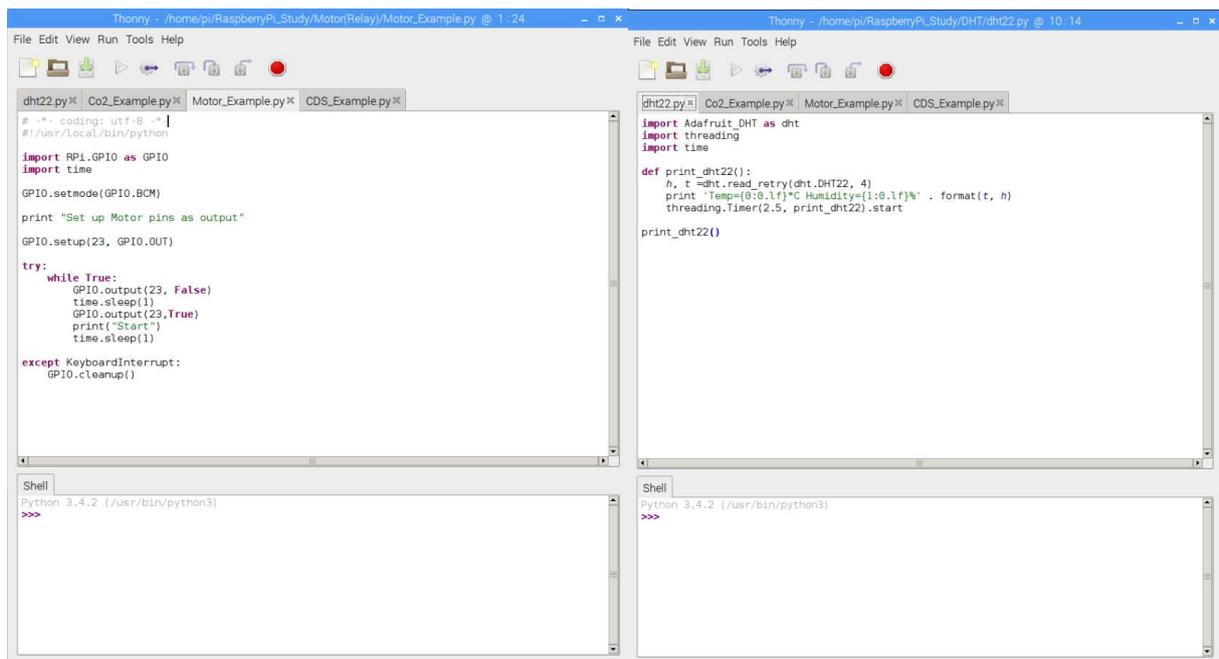


그림 8 - 센서 작동을 할 수 있도록 하는 Python 코드 작성 전용 Thonny IDE 실행화면

라즈베리파이에서 센서의 값을 받기 위해 다음과 같은 Thonny IDE를 설치 후 실행되는 모습이다. 라즈베리파이 안에서 디버깅을 완료한 후 센서값을 받도록 설정 후, OCF 통신을 진행해야 한다.

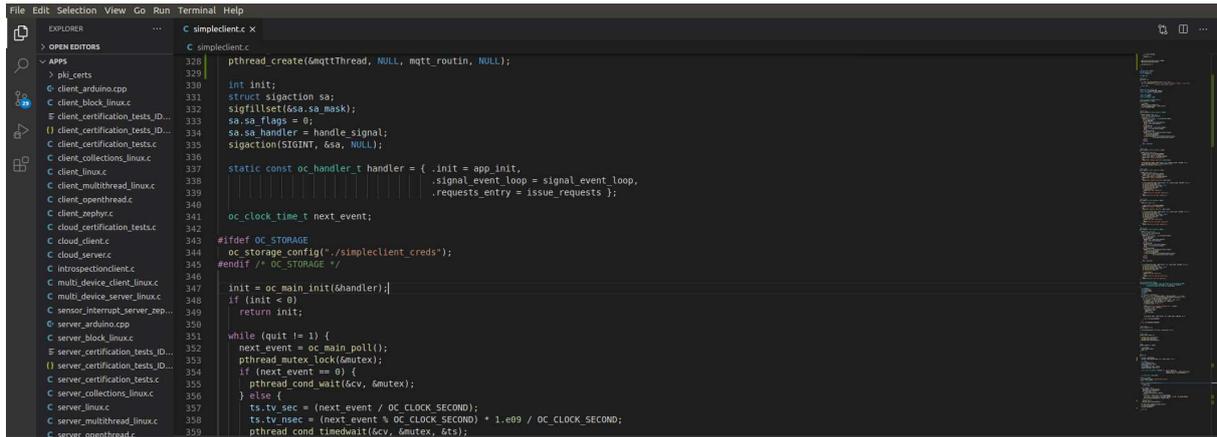


그림 9 - OCF 통신을 위한 IoTivity 작업을 진행하고 있는 Visual Studio Code 실행화면

본 프로그램은 OCF 통신을 위한 작업 환경으로, 라즈베리파이와 외부의 컴퓨터와 연결하여 Visual Studio Code를 활용하여 직접 프로그래밍을 진행하고 있는 환경이다. 이에 근거하여, 라즈베리파이 단에서 작업을 진행하지 않고 라즈베리파이가 외부 SSH 연결을 허용하기에 작업의 용이성과, 이에 따른 확장성도 보장할 수 있다.

(라) 소프트웨어 개발

1) 소프트웨어 구성

스마트팜 제어를 위한 소프트웨어 구성은 다양한 모듈을 연결하고 이에 대응하는 프로그램 코드가 있어야 한다. 프로그램으로 제어 해야 할 대상은 다음과 같다.

- 온습도 센서, 조도센서, CO<sub>2</sub> 센서로부터 해당 환경 값을 읽는다.
- Web Server를 사용하여 사용자 인터페이스를 제공한다.
- 각종 기기를 제어하기 위해 전기 장치인 컨트롤러를 라즈베리파이가 제어한다. 중요 수단으로는 릴레이를 제어를 통해 각종 기기를 조작한다.
- 기타 통신을 통해 사용자 편의성을 높인다.

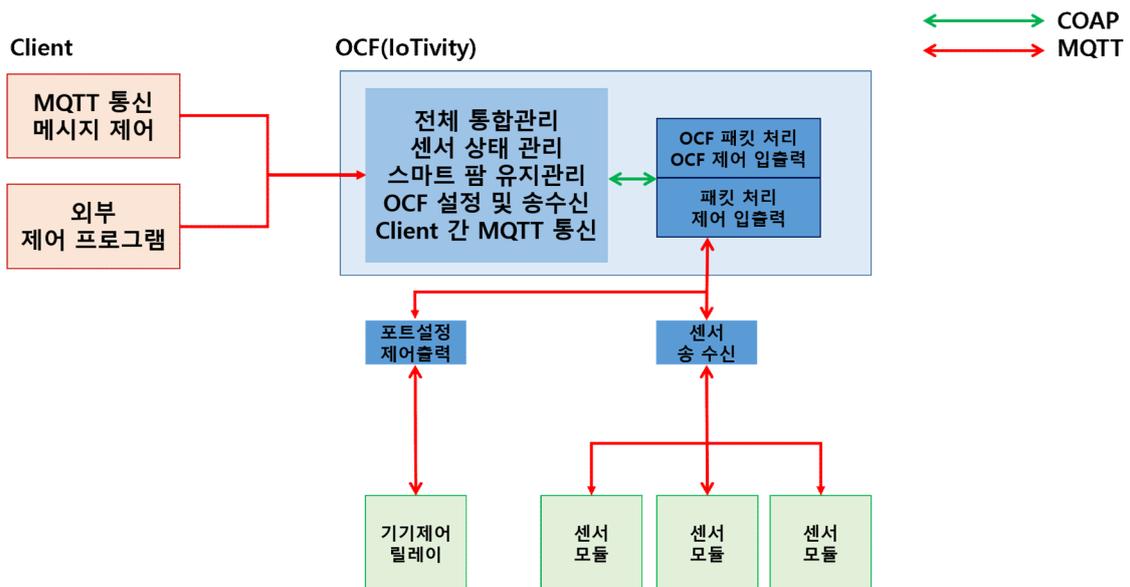


그림 10 - 제어 소프트웨어 구성

2) 소프트웨어 기능

중요 기능은 Client에게 스마트팜을 컨트롤 할 수 있는 UI 제공, 센서 읽기와 Relay 동작 통제를 OCF

규격에 따라 맞추어 통신하기이다.

이와 같은 기능을 구현하기 다음과 같은 기능적 프로그램이 필요하다.

- Client UI 기능 : Client 전용 UI는 웹을 기반으로 하는 메커니즘을 제공한다. 따라서 운영체제의 제한 없이 웹 표준에 맞는 브라우저를 통하여 접근할 수 있다. 웹 페이지에 있는 기능들을 통해 스마트팜 내의 환경을 모니터링 및 명령을 수행할 수 있도록 지원한다.
- 센서 노드 : 센서 모듈의 현재 센서값을 읽기 위해 SPI 통신을 사용한다. 프로그램으로 우선 SPI 모듈을 설정하고, 센서에 특정 포맷의 형태로 데이터 보내면 센서가 반응하여 수집된 값을 MQTT Client로 보내온다. 이 수신 포맷을 분석하여 센서값을 인지하고 변수에 넣어 제어에 활용한다.
- Relay 액추에이터 제어 : 만약 원하는 온도보다 높게 상승하면 Web Server를 활용하여 온도를 낮춘다. 액추에이터 제어는 라즈베리파이의 포트의 디지털 출력을 활용한다.
- OCF 통신 : Operator 역할인 OCF Client는 Controller Node 역할인 OCF Server와 통신을 하며 Publish와 Subscribe 작업을 병행하며 스마트팜 통제에 관한 데이터를 교환한다.적으로 연결되어 동작한다. 프로그램 내에서는 우선 변수에 값을 저장하고 처리해야 한다.

### 3) 소프트웨어 설계

CF 제어 프로그램을 구현하기 데이터 수집, 온도의 상황에 맞는 제어, UI에 의한 목표값 및 환경 값 설정 등이 필요하다. 이 모든 요소가 유기적으로 연결되어 동작한다. 프로그램 내에서는 우선 변수에 값을 저장하고 처리해야 한다.

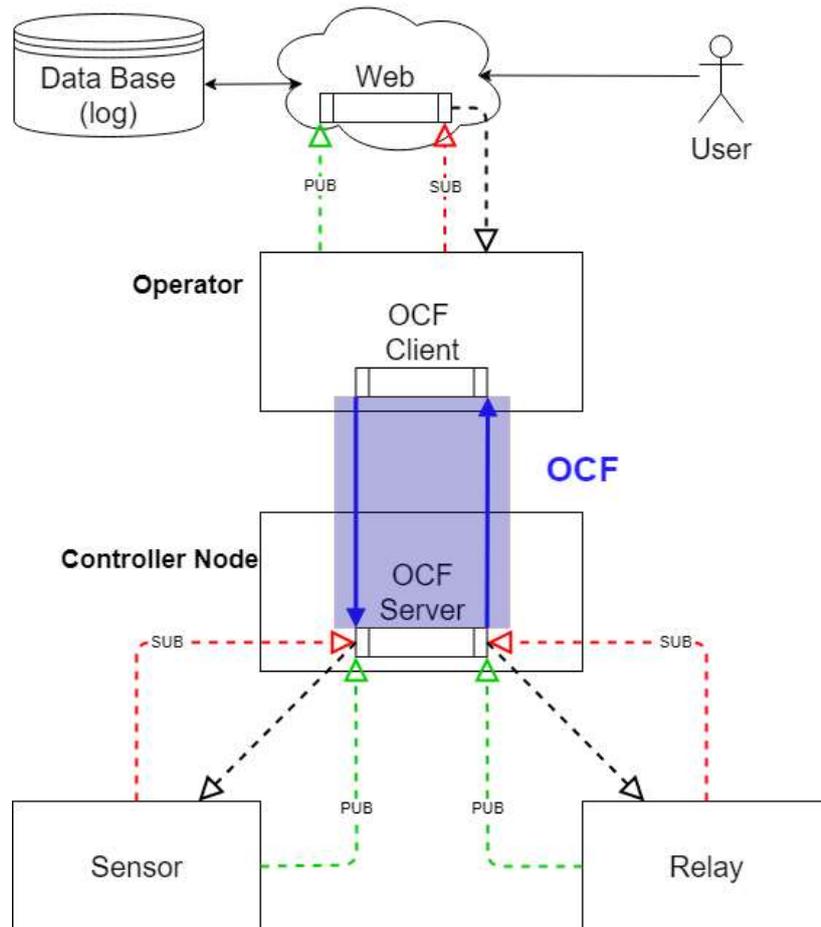


그림 11 - OCF 규격에 따른 스마트팜 통신 다이어그램

### 4) 외부 장치 통신 연결 및 호출

HTML를 사용하는 장치뿐만 아니라 다양한 웹에서 동작으로 하기 위해 컨트롤러와 단말간의 통신을 설정하고 필요한 데이터를 공유한다. 이를 위해 우선 서버-클라이언트 개념을 도입한다.

#### ■ 웹을 통한 사용자 접근

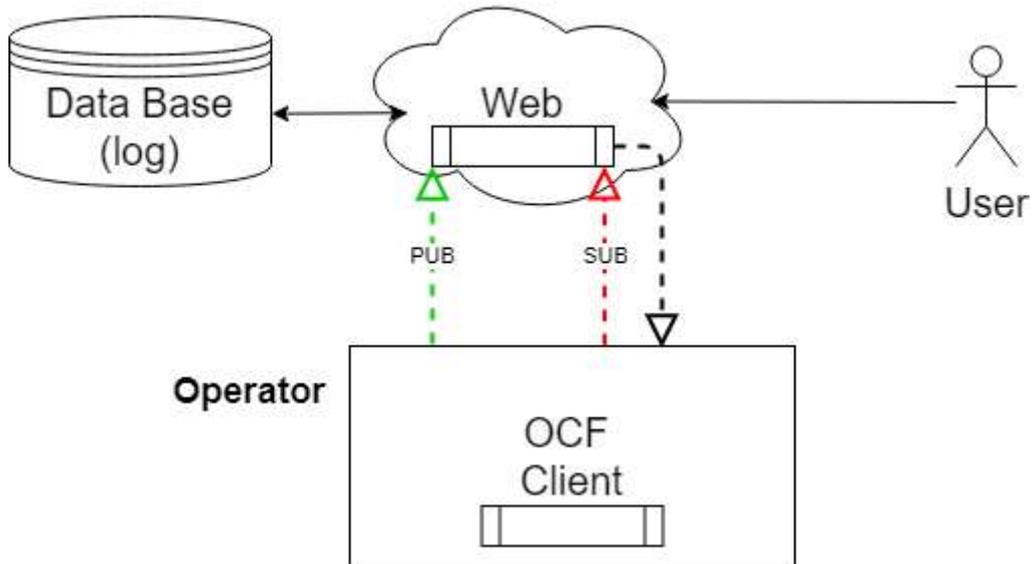


그림 12 - Web Server와 관련된 통신 다이어그램

- User : 이 프로그램에 대한 전반적인 모니터링 및 각종 기능을 제어할 권한을 가지고 있다.
- Web : Web을 보여 주는 컴퓨터 사용자 요청 큐(Queue)에 명령을 넣어 보관한다.
- DataBase : 데이터베이스에서는 모든 입출력 log들이 기록되면서 동시에 데이터를 보관한다.
- OCF Client : OCF Client는 계속 Web에 있는 사용자 요청 큐를 Subscribe를 하며, 동시에 명령이 들어온다면, 이를 가져와 OCF Client에 있는 사용자 요청 큐에 명령을 넣어 보관한다. 또한, OCF Client에 있는 사용자 요청 큐에 대한 정보를 Web 쪽으로 Publish를 해주어 데이터를 전송한다.

클라이언트의 프로그램의 양상이나 표현 방식 등은 차후의 통신 기기의 성향에 따라 달라지므로 OS나 개발환경에 맞게 프로그램 한다. 따라서 서버 입장에서는 정해진 패턴의 대응을 하면 유연성이 떨어지므로 서버는 자기가 가지는 데이터를 제공하는 서버 역할을 한다. 그리고 클라이언트에서 오는 명령 수행을 한다. 클라이언트에서 원하는 기능 구현을 위해 UI를 설계하고 클라이언트가 주체가 되어 서버에게 필요한 데이터를 요구한다. 서버는 데이터를 전송함으로써 전체 시스템을 구성한다.

■ □ OCF 통신 설정

다른 기기 간 접속을 가정하고 우선 OCF 규격의 인식이 필요하다. 소켓 프로그램처럼 CONNECT 절차를 두어 기기를 서버에서 인식하는 절차를 거친다. OCF Client와 Server는 서버에 등록된 기기에 대해 CONNECT 이후 데이터 전송을 시작한다.

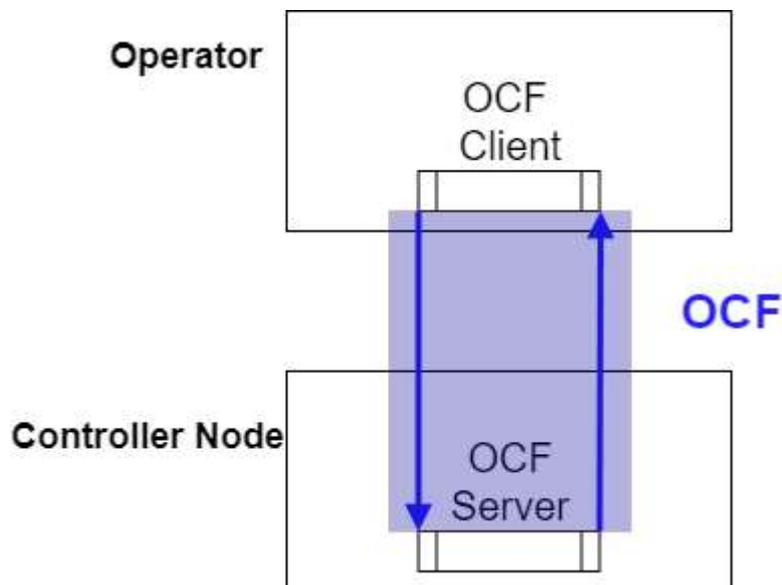


그림 13 - CONNECT

OCF Client가 CONNECT 요청을 하면 OCF Server는 기기를 등록하고 응답에 대한 대기한다. OCF Server에 있는 큐에서 이 패킷을 응답을 통해 Client측으로 보낼 데이터가 있다면 그 정보를 가지고 데이터 통신을 시작한다.

■ OCF Server와 Sensor, Relay 간 통신 설정

다른 기기 간 접속을 가정하고 우선 MQTT 규격의 인식이 필요하다. 소켓 프로그램처럼 CONNECT 절차를 두어 기기를 서버에서 인식하는 절차를 거친다. Controller Node는 MQTT Server/Client의 임무를 수행하고, 각 Sensor 측과 Relay 측은 MQTT Client의 임무를 수행하여 MQTT를 통해 통신한다.

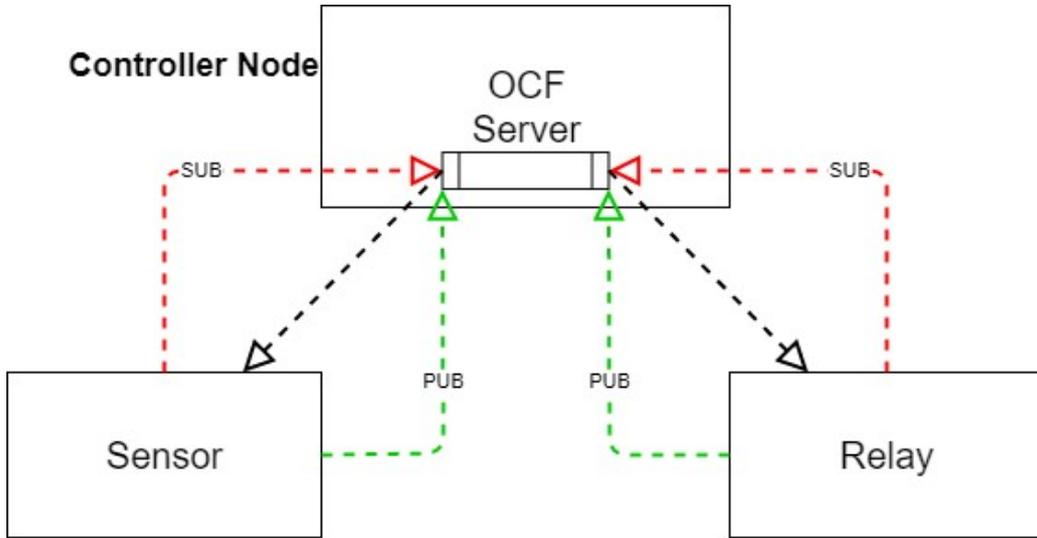


그림 14 - OCF 서버 사용자 요청 Queue

OCF Server에 있는 사용자 요청 큐에는 항상 Sensor와 Relay측이 Subscribe를 하고 있다. 그러다가 큐 안에 요청이 들어올 때 그 요청에 알맞은 곳으로 이동한다. 모터 동작 및 제어를 하고자 하는 요청이 들어올 때 Relay 측으로, Sensor 값의 측정을 하고자 하는 요청이 들어올 때는 Sensor 측으로 요청을 한다. 끝으로 각 단말에 들어온 요청을 수행하고 데이터를 추출 해내면 이 데이터를 OCF Server에 있는 사용자 요청 큐에 Publish를 해준다.

#### 가-4) [SW등록] 엠큐티티(MQTT)기반 스마트팜 센서/구동기/제어기 노드 연동 프로그램('21)

##### (가) 연구 수행 개요

온도, 습도, 조도 등의 스마트팜 센서 노드 및 개폐기 등의 스마트팜 구동기 노드와 이들을 제어하는 스마트팜 제어기 노드를 구현하여 시험함으로써 OCF 표준기술 기능 검증에 활용하였다. 스마트팜 (시설원예) 분야에서 활용되는 센서 노드, 구동기 노드, 제어기 노드의 특성 및 기능을 정의하는 프로파일을 구성하고 이를 통해 신규 센서 및 구동기 노드의 추가가 가능함을 확인하였다. 이를 위하여 사전 정의된 프로파일에 따라 스마트팜에서 활용되는 센서 및 구동기 노드 그리고 제어기 노드의 특성 및 기능을 구현하고 검증하였다.

\*) 본 구현 S/W에 대한 시험을 위해, 아래와 같은 동작 시나리오를 작성하여 시험성적서 발급 담당자가 참여한 환경에서 동작 여부 일치를 확인하는 과정을 거쳐 시험 성적서 발급받음  
([http://www.ktc.re.kr/web\\_united/customerservice/customerservice.asp?pagen=1527](http://www.ktc.re.kr/web_united/customerservice/customerservice.asp?pagen=1527))

##### (나) 연구 주요 내용

MQTT기반 스마트팜 센서/구동기/제어기노드를 소프트웨어로 구현하여 테스트베드를 구성하였다.

스마트팜 센서 노드 및 구동기 노드는 제어기 노드와 MQTT로 통신하면서 제어기 노드에 상태정보를 전송하고, 제어기 노드로부터 제어정보를 수신하여 동작하였다.

스마트팜 제어기 노드는 별도로 구현된 스마트팜 운영노드와 OCF기반의 CoAP통신을 수행하여 동작하였다.

#### ■ 센서 노드

##### ①. pimoroni 설치 방법

# 제조사 제공 라이브러리 설치

```
curl https://openconnectivity.github.io/Sample-Raspberry-Pi-Code/pi-boards/install.sh | bash
```

# 설치 완료 후 생성파일들은 다음을 참고.

# ~/Sample-Raspberry-Pi-Code, ~/Pimoroni, raspi-gpio (Lib) 등등

# Pimoroni Enviro pHAT 보드 동작 확인

```
vi ~codes/all.py << ~/Pimoroni/envirophat/example/all.py
```

# 예제 소스 참고

```
pi@raspberrypi:~/codes $ python all.py
|-- Enviro pHAT Monitoring --
Temp: 35.76c
Pressure: 1005.81hPa
Altitude: 62.11m
Light: 630
RGB: 87, 99, 81
Heading: 292.31
Magnetometer: 499 -1413 -2908
Accelerometer: 0.03g -0.0g 1.06g
Analog: 0: 0.516, 1: 0.537, 2: 0.528, 3: 0.531
```

# Tip. pimoroni 하드웨어 설치의 경우 솔더링 필요. 라즈베리파이에 그냥 올려놓아서 연결 안됨. 접촉 불량.

# 피모로니 보드를 장착한 상태에서 라즈베리파이에 케이스를 씌우면 센서값이 다소 부정확해짐을 감안해야 하였다. 동작중인 라즈베리파이 보드 자체 온도에 영향을 받게 됨.

# 실제로 라즈베리파이 케이스를 씌운 경우 상온의 온도가 섭씨 41도가 넘게 측정됨.

```
pi@raspberrypi: ~  
pi@raspberrypi:~/Pimoroni/envirophat/examples $ cp all.py ~  
> ^C  
pi@raspberrypi:~/Pimoroni/envirophat/examples $ cp all.py ~  
pi@raspberrypi:~/Pimoroni/envirophat/examples $ cd  
pi@raspberrypi:~ $ ls  
all.py  Documents  Music  Pictures  Public  Templates  
Desktop Downloads +old  Pimoroni  Sample-Raspberry-Pi-Code  Videos  
pi@raspberrypi:~ $ ./all.py  
^Cpi@raspberrypi:~ $ oring ---  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $ a  
pi@raspberrypi:~ $ ./all.py  
█-- Enviro pHAT Monitoring ---  
Temp: 41.69c  
Pressure: 1006.65hPa  
Altitude: 55.06m  
Light: 1066  
RGB: 79, 101, 86  
Heading: 346.65  
Magnetometer: 1368 -124 -4218  
Accelerometer: 0.02g -0.06g 1.13g  
Analog: 0: 0.495, 1: 0.504, 2: 0.501, 3: 0.522
```

# 케이스를 씌우지 않더라도 섭씨 37도 이상 측정됨. 동작중인 라즈베리파이 보드의 영향.

```
pi@raspberrypi: ~/codes  
Project-Scripts  all.py  mqtt  pimoroni_sensor2.py  
pi@raspberrypi:~/codes $ python --version  
Python 2.7.16  
pi@raspberrypi:~/codes $ python3 --version  
Python 3.7.3  
pi@raspberrypi:~/codes $ vi all.py  
pi@raspberrypi:~/codes $ ./all.py  
^Cpi@raspberrypi:~/codes $ ---  
pi@raspberrypi:~/codes $  
pi@raspberrypi:~/codes $  
pi@raspberrypi:~/codes $  
pi@raspberrypi:~/codes $ ./all.py  
█-- Enviro pHAT Monitoring ---  
Temp: 36.88c  
Pressure: 1006.10hPa  
Altitude: 59.63m  
Light: 779  
RGB: 86, 99, 81  
Heading: 326.6  
Magnetometer: 1350 -845 -3230  
Accelerometer: 0.01g -0.03g 1.06g  
Analog: 0: 0.504, 1: 0.519, 2: 0.531, 3: 0.531
```

# 위의 경우 상온의 온도는 25도 이하.

# DHT11 센서로 측정해보면 섭씨 23도 정도로 측정됨.

```

pi@raspberrypi: ~
Using username "pi".
pi@192.168.2.103's password:
Linux raspberrypi 5.10.11-v7+ #1399 SMP Thu Jan 28 12:06:05 GMT 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar 30 13:48:16 2021 from 192.168.2.101

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ ls
2019-11-06-143937_656x416_screenshot.png  Documents  Music  Templates
Desktop  Downloads  Pictures  Videos
dht11  iot-lite  Public
dht11.py  MagPi  temp humid.py
pi@raspberrypi:~ $ ./temp humid.py
2021-03-30 13:53:06.321450 @1.123, #2, Temperature = 23.0*C Humidity = 20.0%

```

# mqtt를 통해 pimoroni 센싱값을 전송(발행)하는 코드 작성  
vi ~/codes/pimoroni\_sensor.py << ~/cdes/main.py # 소스코드 및 설정파일 참고

# DHT11 온습도 센서를 사용하는 경우 다음과 같이 Adafruit\_DHT 라이브러리 설치..  
pip install Adafruit\_DHT # python3을 사용하는 경우.. pip3 install  
Adafruit\_DHT  
# DHT11 (파란색) 온습도 센서와 DHT22 (하얀색) 센서의 차이는 측정값의 정확도..  
# DHT22가 신형으로, 소수점 1자리까지 측정.. 예로, DHT11이 24도라면 DHT22은 24.3도.

---

## ②. paho 설치 방법

---

# 파호 파이썬 라이브러리 설치  
pip install paho-mqtt #python3을 쓰는 경우에는.. pip3 install  
paho-mqtt

# 적당한 이름으로 디렉토리 생성. 여기서는 ~/codes 생성  
mkdir ~/codes  
cd ~/codes

# 간단한 테스트 파일 작성  
vi sub.py # 예제 소스 참고  
vi pub.py # 예제 소스 참고

# 파호 라이브러리를 활용한 MQTT 동작 확인  
python sub.py << from another terminal #1.  
python sub.py << from another terminal #1.  
python pub.py

---

---

# 센서노드 작성코드 동작확인

```
pi@raspberrypi:~/codes $ python3 pimoroni_sensor.py
Connected ! result code = 0
{"humidity": 62.00211137236013, "temperature": 36.04162243590772, "co2": 1005.8248399152995, "brightness": 640, "timestamp": 1613873622, "datetime": "2021-02-21 11:13:42.574580"}
ocf/sensor/pimoroni/1 Published
```

# 예시: “운영PC” 노드에서 센싱값 전송 확인

# 센싱값이 다른 이유는 시간차를 두고 subsensor.py를 실행했기 때문.. 시간값 확인..

```
(base) osboxes@ocf:~/iot-lite/mqtt-test$ python3 subsensor.py
Connected with result cod 0
ocf/sensor/pimoroni/1

ocf/sensor/pimoroni/1  b'{"humidity": 62.269044179492575, "temperature": 36.11581859468536, "co2": 1005.7929684831668, "brightness": 643, "timestamp": 1613874342, "datetime": "2021-02-21 11:25:42.779254"}'
```

# Tip. 파호를 통해 다양한 방법으로 Pub/Sub을 구현할 수 있음.

python pubm.py # 멀티플 퍼블리시..

# Tip. 루프 코드 없이 지정된 시간 주기로 반복하고 싶을 때..

watch -n 1500 python pub.py

---

## ■ 구동기

### ①. relay 설치 방법

---

# RPi.GPIO 라이브러리 :

# 파이썬에서 라즈베리파이의 gpio핀을 제어할 수 있도록 해주는 라이브러리. 라즈비안에 기본 설치 되어있다.

# GPIO 기반 일반 릴레이 모듈을 사용하므로, 별도의 라이브러리 설치는 필요 없음.

# PIN 번호만 확인하면 됨

# 라즈베리파이 <> 릴레이 편연결

# 전원 (5V) > 2번

# 접지 (GND) > 6번

# 신호 (GPIO27) > 13번

---

### ②. paho 설치 방법

---

# MQTT프로토콜을 활용하기 위해 파호 파이썬 라이브러리 설치

pip install paho-mqtt #python3을 쓰는 경우에는.. pip3 install paho-mqtt

# 적당한 이름으로 작업 디렉토리 생성 # 여기서는 ~/codes 생성

mkdir ~/codes/

cd ~/codes

vi relay.py # 소스코드 및 설정파일 참고

# mqtt 테스트 코드 작성

---

---

```
# 구동기는 pub과 sub 둘 다 필요
# pub : 릴레이 모듈의 상태 값을 전송(발행)
# sub : 릴레이 모듈의 제어 명령을 구독.
vi sub.py    # 예제 소스 참고      # 센서노드에서 작성한 코드 재활용
vi pub.py    # 예제 소스 참고      # 센서노드에서 작성한 코드 재활용
```

```
# 파호 라이브러리를 활용한 MQTT 동작 확인
python sub.py:      python pub.py # 예제 소스 참고
```

```
# 구동기 노드 작성코드 확인 # 릴레이 모듈 동작 확인
python3 relay.py      # python이 아니라 python3으로 실행. # 버전2는 에러.
```

```
pi@raspberrypi:~/codes $ python3 relay.py
relay.py:82: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings
(False) to disable warnings.
  GPIO.setup(RELAY_GPIO_PIN, GPIO.OUT)
Connected ! result code = 0
-->> Origin state is {"value": false}
{"value": true} -->> TRUE state test pass
{"value": false}-->> FALSE state test pass
```

```
# 구동기 노드 응답 전송 확인
# 예시: "운영PC" 노드에서 센싱값 전송 확인
```

```
(base) osboxes@ocf:~/iot-lite/mqtt-test$ python3 subactuator.py
Connected with result code 0
ocf/response/actuator/#

ocf/response/actuator/waterdispensor    b'{"value": false}'
ocf/response/actuator/waterdispensor    b'{"value": true}'
ocf/response/actuator/waterdispensor    b'{"value": false}'
```

---

## ■ 제어기

### ①. iot-lite (server) 설치 방법

<https://openconnectivity.github.io/IoTivity-Lite-setup/#development-flow>

```
# IoTivity lite 버전 설치.      # IoTivity classic은 2.2.1(?) 버전을 끝으로 개발이 멈춤.
curl https://openconnectivity.github.io/IoTivity-Lite-setup/install.sh | bash
```

```
# IoTivity lite 동작 테스트      (디바이스빌더 활용)      # IoTivity lite 제공 기본 예제 파일
활용
```

```
# IoTivity lite 디렉토리로 이동
cd ~/iot-lite
```

```
# example.json 편집. # 소스코드 및 설정파일 참고
# example.json에 해당 장치가 어떤 리소스를 가지고 있고, 이 리소스를 활용하기 위한 방법들이
```

---

---

명세 되고, 이후 과정에서 이 json파일(example.json)을 기초로 코드 및 인트로스펙션 파일들 생성.

# 즉, 해당 장치에 대한 “프로파일”이 정의된 json 파일.

# 즉, json파일은 해당 장치가 어떤 데이터를 소비하고 어떤 데이터를 생성하는지 명세하게 됨.

# 여기서는 “바이너리스위치” 리소스를 가지는 구동기를 예제로 프로파일을 정의.

# 간단히 “path”, “rt”, “if” 리소스 정도만 가지는 예제.

# 리소스 활용방법이 되는 CRUDN 중에서 사실상 R과 U, 그리고 N까지만 지원.

./edit\_input.sh # 소스코드 및 설정파일 참고

>> ctrl+x # 가상 센서를 활용하는 예제이므로 일단 그대로 활용. 수정없이 종료.

# 코드(device\_builder\_server.c), 명세파일(introspection), 보안파일(pki\_cert.zip) 등을 생성하는 스크립트 실행.

# ~/iot-lite/device\_output/code/simpleserver.c, server\_introspection.dat.h, server\_introspection.dat 등을 활용

./gen.sh # 소스코드 및 설정파일 참고

>> DeviceBuilder\_IotivityLiteServer.sh # 소스코드 및 설정파일 참고

# gen.sh를 통해 생성된 device\_builder\_server.c 코드 편집

./edit\_code.sh # 소스코드 및 설정파일 참고

>> ctrl+x # 가상 센서를 활용하는 예제이므로 일단 그대로 활용. 수정없이 종료.

# 추후에 구동기 및 센서 노드 데이터 송수신을 위한 MQTT 및 cJSON 관련 함수 추가

# 컴파일을 통해 플랫폼에 맞는 실행 바이너리 파일 빌드.

# 여기서는 “cd iotivity-lite/port/linux/” 에서 빌드. # 리눅스

./build.sh

# MQTT 적용 버전의 경우에는 Iot-lite 빌드에서 MQTT 라이브러리 필요.

# 기타 참고사항에서 [paho, cJSON 소스코드로 설치] 참고.

# Tip. example.json을 그대로 쓰는 기본버전의 경우는 해당 디렉토리에서 make 만 해도 가능

# 라즈베리파이인경우 보통 다음의 위치에서 동작.

cd ~/iot-lite/iotivity-lite/port/linux:

make

# simpleserver, simpleclient, onboarding\_tool 등으로 간단한 테스트..

# device\_builder\_server 시작

./run.sh

# 온보딩 프로세스는 서버보다는 클라이언트에서 수행하는 것이 바람직

# device\_builder\_server 종료

pkill device\_builder\_server

# 다른 방법으로도 사용이 가능

ctrl+Z # 실행중인 프로세스에 정지 신호 보내기

[#]+ Stopped # 백그라운드 프로세스 전환

---

---

```
jobs          # 백그라운드 프로세스 확인
kill -9 %#    # 백그라운드 프로세스 종료
```

---

## ②. mosquito 설치 방법

---

```
# MQTT 브로커 실행을 위한 바이너리 설치

# 모스키토 소프트웨어 패키지 설치
sudo apt install mosquitto

# 기본 설정 파일 확인
sudo vi /etc/mosquitto/mosquitto.conf      # 특별히 수정사항 없음.

# Tip. 모스키토는 클라이언트도 제공하였다. 이 테스트베드는 파호를 활용하므로 필요 없음.
# 모스키토 클라이언트 설치는 필요 없음.
# 대신에 paho 라이브러리를 사용.
#모스키토 클라이언트 설치
sudo apt install mosquitto-clients
# 클라이언트 구독 테스트 (단, 브로커는 미리 설치해야 함)
mosquitto_sub -h 127.0.0.1 -t 'topic/test'
# 클라이언트 발행 테스트
mosquitto_pub -h 127.0.0.1 -t 'topic/test' -m "Hello Mosquitto!"
```

---

## ③. paho 설치 방법

---

```
# 파호 파이썬 라이브러리 설치
pip install paho-mqtt          #python3을 쓰는 경우에는..      pip3 install
paho-mqtt

# 적당한 이름으로 디렉토리 생성. 여기서는 iot-lite 아래 생성
mkdir ~/iot-lite/mqtt-test:    cd ~/iot-lite/mqtt-test

# 간단한 테스트 파일 작성
vi sub.py    # 예제 소스 참고
vi pub.py    # 예제 소스 참고

# 파호 라이브러리를 활용한 MQTT 동작 확인
# sub 실행
python sub.py

# 다른 터미널에서 pub 실행
python pub.py

# Tip. 파호를 통해 다양한 방법으로 Pub/Sub을 구현할 수 있음.
python pubm.py          # 멀티플 퍼블리시..
```

---

---

# Tip. 루프 코드 없이 지정된 시간 주기로 반복하고 싶을 때..

```
watch -n 1500 python pub.py
```

---

## 가-5) [SW등록] 엠큐티티(MQTT)기반 스마트팜 운영노드/클라우드 연동 프로그램 ('21)

### (가) 연구 수행 개요

스마트팜 노드의 상태를 웹UI를 통해 확인 및 제어할 수 있도록 클라우드 기반의 기능을 구현하여 시험함으로써 표준기술검증에 활용하였다. 이를 위해 클라우드 기반의 스마트팜 노드들에 대해서는 모니터링 및 제어 그리고 데이터 수집을 통해 클라우드 기반 스마트팜 기능 검증 및 데이터 수집이 가능함을 확인하였다. 센서 및 구동기 등 스마트팜 노드에 대해서는 상태를 원격에서 모니터링하고, 특정 기능에 대한 제어 명령을 송신하는 기능을 구현하여 테스트베드에 적용하여 기능검증을 시험하였다.

\*) 본 구현 S/W에 대한 시험을 위해, 아래와 같은 동작 시나리오를 작성하여 시험성적서 발급 담당자가 참여한 환경에서 동작 여부 일치 여부를 확인하는 과정을 거쳐 시험 성적서 발급받음

([http://www.ktc.re.kr/web\\_united/customerservice/customerservice.asp?pagen=1527](http://www.ktc.re.kr/web_united/customerservice/customerservice.asp?pagen=1527))

### (나) 연구 주요 내용

MQTT기반 스마트팜 운영노드 및 웹&DB노드를 소프트웨어로 구현하여 테스트베드를 구성하였다.

스마트팜 운영노드는 웹&DB노드와 MQTT로 통신함으로써, 스마트팜의 상태정보를 클라우드로 전송하고, 클라우드로부터의 제어정보를 수신하였다. 스마트팜 운영노드는 OCF기반의 CoAP통신을 통해 제어기 노드와 통신함으로써 센서 노드 및 구동기 노드를 제어하였다. 웹서버 및 DB서버가 탑재되는 노드는 클라우드에 탑재되어 구동될 수 있다.

## ■ 운영PC

### ①. iot-lite (client) 설치 방법 및 테스트

---

<https://openconnectivity.github.io/IoTivity-Lite-setup/#development-flow>

# IoTivity lite 버전 설치. # 제어기 노드와 동일

```
curl https://openconnectivity.github.io/IoTivity-Lite-setup/install.sh | bash
```

# IoTivity lite 동작 테스트 # 제어기 노드의 서버는 디바이스빌더 활용.

# MQTT 적용 버전의 경우에는 Iot-lite 빌드에서 MQTT 라이브러리 필요.

# 기타 참고사항에서 [paho, cJSON 소스코드로 설치] 참고.

# Tip. example.json을 그대로 쓰는 기본버전의 경우는 해당 디렉토리에서 make 만 해도 가능

# 라즈베리파이인경우 보통 다음의 위치에서 동작.

```
cd ~/iot-lite/iotivity-lite/port/linux:
```

```
make
```

# simpleserver, simpleclient, onboarding\_tool 등으로 간단한 테스트..

# simpleclient.c 내용 수정

```
cd ~/iot-lite.org/iotivity-lite/apps
```

```
vi simpleclient.c # 소스코드 및 설정파일 참고 # MQTT 및 cJSON 라이브러리 추가
```

---

simpleclient.c 수정 내용

---

```

/* simpleclient.c 내용 수정
 * 주요수정내용(simpleclient.c)
 * 구동기 및 센서 노드 데이터 송수신을 위한 MQTT 및 cJSON 관련 함수 추가*/

static struct _smmc_state;
static int app_init(void);
static void post_waterdispensor(oc_client_response_t *data);
static void get_waterdispensor(oc_client_response_t *data);
static oc_event_callback_retval_t stop_observe(void *data)
static void get_smmc(oc_client_response_t *data);
void delivered(void *context, MQTTClient_deliveryToken dt);
int msgarrvd(void *context, char *topicName, int topicLen, MQTTClient_message
*message); //topicName, message데이터
void connlost(void *context, char *cause);
void *mqtt_routine(void *arg);
static oc_discovery_flags_t discovery(const char *anchor,
                                     const char *uri,
                                     oc_string_array_t types,
                                     oc_interface_mask_t iface_mask,
                                     oc_endpoint_t *endpoint,
                                     oc_resource_properties_t bm,
                                     void *user_data);

static void issue_requests(void);
static void signal_event_loop(void);
void handle_signal(int signal);

```

---

컴파일 및 실행은 다음과 같이 수행한다.

---

```

# make 파일 수정
cd ~/iot-lite/iotivity-lite/port/linux
vi Makefile          # 소스코드 및 설정파일 참고
# 주요수정내용 (Makefile):
# MQTT 및 cJSON 라이브러리 관련 컴파일 옵션 추가
simpleclient: libiotivity-lite-client.a $(ROOT_DIR)/apps/simpleclient.c
    @mkdir -p $$@_creds
    ${CC} -o $$@ ../../apps/simpleclient.c libiotivity-lite-client.a -DOC_CLIENT
${CFLAGS} ${LIBS} -lpaho-mqtt3c -lcjson -Wno-unused-parameter

# simpleclient 컴파일
cd ~/iot-lite/iotivity-lite/port/linux
make
osboxes@ocf:~/iot-lite/iotivity-lite/port/linux$ make
gcc -o simpleclient ../../apps/simpleclient.c libiotivity-lite-client.a -DOC_CLIENT -fPIC
-fno-asynchronous-unwind-tables -fno-omit-frame-pointer -ffreestanding -Os

```

---

---

```
-fno-stack-protector -ffunction-sections -fdata-sections -fno-reorder-functions
-fno-defer-pop -fno-strict-overflow -I./ -I../include/ -I../ -std=gnu99 -Wall -Wextra
-Werror -pedantic -Wl,--gc-sections -I../deps/mbedtls/include -D__OC_RANDOM
-DOC_PKI -DOC_DYNAMIC_ALLOCATION -DOC_IDD_API -DOC_SECURITY -lm -pthread
-lrt -lpaho-mqtt3c -lcjson -Wno-unused-parameter
```

```
gcc -o server_certification_tests ../../apps/server_certification_tests.c
libiotivity-lite-client-server.a -DOC_CLIENT -DOC_SERVER -fPIC
-fno-asynchronous-unwind-tables -fno-omit-frame-pointer -ffreestanding -Os
-fno-stack-protector -ffunction-sections -fdata-sections -fno-reorder-functions
-fno-defer-pop -fno-strict-overflow -I./ -I../include/ -I../ -std=gnu99 -Wall -Wextra
-Werror -pedantic -Wl,--gc-sections -I../deps/mbedtls/include -D__OC_RANDOM
-DOC_PKI -DOC_DYNAMIC_ALLOCATION -DOC_IDD_API -DOC_SECURITY -I../api/cloud
-lm -pthread -lrt
```

```
gcc -o smart_home_server_linux ../../apps/smart_home_server_linux.c
libiotivity-lite-server.a -DOC_SERVER -fPIC -fno-asynchronous-unwind-tables
-fno-omit-frame-pointer -ffreestanding -Os -fno-stack-protector -ffunction-sections
-fdata-sections -fno-reorder-functions -fno-defer-pop -fno-strict-overflow -I./
-I../include/ -I../ -std=gnu99 -Wall -Wextra -Werror -pedantic -Wl,--gc-sections
-I../deps/mbedtls/include -D__OC_RANDOM -DOC_PKI -DOC_DYNAMIC_ALLOCATION
-DOC_IDD_API -DOC_SECURITY -lm -pthread -lrt
```

```
gcc -o client_certification_tests ../../apps/client_certification_tests.c
libiotivity-lite-client-server.a -DOC_CLIENT -DOC_SERVER -fPIC
-fno-asynchronous-unwind-tables -fno-omit-frame-pointer -ffreestanding -Os
-fno-stack-protector -ffunction-sections -fdata-sections -fno-reorder-functions
-fno-defer-pop -fno-strict-overflow -I./ -I../include/ -I../ -std=gnu99 -Wall -Wextra
-Werror -pedantic -Wl,--gc-sections -I../deps/mbedtls/include -D__OC_RANDOM
-DOC_PKI -DOC_DYNAMIC_ALLOCATION -DOC_IDD_API -DOC_SECURITY -lm -pthread
-lrt
```

```
gcc -o onboarding_tool ../../onboarding_tool/obtmain.c libiotivity-lite-client.a
-DOC_CLIENT -fPIC -fno-asynchronous-unwind-tables -fno-omit-frame-pointer
-ffreestanding -Os -fno-stack-protector -ffunction-sections -fdata-sections
-fno-reorder-functions -fno-defer-pop -fno-strict-overflow -I./ -I../include/ -I../
-std=gnu99 -Wall -Wextra -Werror -pedantic -Wl,--gc-sections
-I../deps/mbedtls/include -D__OC_RANDOM -DOC_PKI -DOC_DYNAMIC_ALLOCATION
-DOC_IDD_API -DOC_SECURITY -lm -pthread -lrt
```

```
make --directory=../../deps/gtest/make
```

```
make[1]: Entering directory '/home/osboxes/iot-lite/iotivity-lite/deps/gtest/make'
```

```
make[1]: Nothing to be done for 'all'.
```

```
make[1]: Leaving directory '/home/osboxes/iot-lite/iotivity-lite/deps/gtest/make'
```

```
# simpleclient 실행
```

```
cd ~/iot-lite/iotivity-lite/port/linux
```

```
(base) osboxes@ocf:~/iot-lite/iotivity-lite/port/linux$ ./simpleclient
Resource /smmc hosted at endpoints:
coaps://[fe80:0000:0000:0000:241b:81a5:5fff:5c0d]:48458
Resource /waterdispensor hosted at endpoints:
coaps://[fe80:0000:0000:0000:241b:81a5:5fff:5c0d]:54601
Subscribing to topic
```

```
# discovery가 수행되어 사용가능한 리소스들이 표시됨 # 예제는 smmc, waterdispensor 표시
```

```
# 온보딩 프로세스는 서버보다는 클라이언트에서 수행하는 것이 바람직 # 여기서는 운영PC
```

```
# 온보딩 프로세스는 서버 또는 클라이언트가 시작하고 수행
```

---

---

```
# 둘 다 시작되고 수행되도 상관없음.
```

```
# simpleclient 종료 # 이 방법 보다는 아래 방법이 더 확실하였다.  
pkill simpleclient
```

---

## ②. paho 설치 방법 및 테스트

---

```
# 파호 파이썬 라이브러리 설치  
pip install paho-mqtt #python3을 쓰는 경우에는.. pip3 install  
paho-mqtt
```

```
# 적당한 이름으로 디렉토리 생성. 여기서는 iot-lite 아래 생성 # 제어기 노드와 동일  
mkdir ~/iot-lite/mqtt-test: cd ~/iot-lite/mqtt-test
```

```
# 간단한 테스트 파일 작성  
vi sub.py # 예제 소스 참고  
vi pub.py # 예제 소스 참고
```

```
# 파호 라이브러리를 활용한 MQTT 동작 확인  
# sub 실행  
python sub.py
```

```
# 다른 터미널에서 pub 실행  
python pub.py
```

```
# Tip. 파호를 통해 다양한 방법으로 Pub/Sub을 구현할 수 있음.  
python pubm.py # 멀티플 퍼블리시..
```

```
# Tip. 루프 코드 없이 지정된 시간 주기로 반복하고 싶을 때..  
watch -n 1500 python pub.py
```

---

## ■ 웹&DB 서버

### ①. Web server (nginx) 설치 및 시험 방법

---

```
# 웹서버 (nginx) 소프트웨어 패키지 설치  
sudo apt install nginx
```

```
# 버전 확인  
nginx -v
```

```
# 기본 환경설정 파일  
sudo vi /etc/nginx/sites-available/default
```

```
# 메인 페이지 확인  
sudo vi /var/www/html/index.nginx-debian.html
```

---

---

```
# nginx 동작 확인
http://192.168.0.120 # 브라우저를 통해 확인
curl http://localhost # 또는 터미널에서 확인

# 프론트엔드 nginx, 백엔드 노드제이에스 설정을 위해 nginx 설정 수정.
sudo vi /etc/nginx/sites-available/example.conf          ## 소스파일 및 설정파일 참고

# 설정파일 링크
cd /etc/nginx/sites-enabled
ln -s /etc/nginx/sites-available/example.conf example.conf

# Tip. nginx는 시작할 때 다음 디렉토리에 있는 설정을 반영.
# /etc/nginx/nginx.conf          # 기본 설정 파일이며, 아래 두 디렉토리도 여기에 명시
# /etc/nginx/sites-enabled/*     # 모든 파일
# /etc/nginx/conf.d/*           # 모든 파일

# nginx 설정 테스트 # 테스트 실패일 경우 설정파일 수정      # 성공할 때까지..
sudo nginx -t

# nginx 재시작
sudo nginx -s reload

# Tip. 다음 명령들도 각각 nginx 재시작 가능
sudo /etc/init.d/nginx restart
sudo service nginx reload

-----

# NodeJS는 별도의 항목으로 다시 다루어짐. 여기는 웹서비스와 같이 고려되는 부분만 설명.
# Node.JS 설치.
sudo apt install nodejs

# Tip. 아래와 같은 메시지가 나오면 레거시 버전도 마저 설치
The program 'node' is currently not installed. You can install it by typing:
sudo apt install nodejs-legacy

# nodejs 레거시 버전 설치
sudo apt install nodejs

# nodejs 버전 확인
nodejs -v
# Tip. 레거시 버전도 같이 설치되면 다음 명령도 가능.
# node -v
```

---

---

```

> v4.2.6      # 신버전으로 업그레이드 필요, nodejs 설명 참고 # 일단 진행

# nodejs 패키지 매니지먼트 설치 및 버전 확인
sudo apt install npm
npm -v
> v3.5.2      # 신버전으로 업그레이드 필요, nodejs 설명 참고 # 일단 진행

# example.conf 동작 확인 테스트용 웹애플리케이션 작성
# 위치는 nodejs가 실행될 수 있는 아무 적당한 곳에..
vi app.js      # 예제 소스 참고

# 테스트용 웹애플리케이션 실행
nodejs app.js

# 동작 확인
# hosts 파일에 example.com을 반영한 노드에서 브라우저로 확인      # vscode 활용 추천
http://example.com

# nodejs를 실행한 노드의 다른 터미널에서 확인
curl http://127.0.0.1:3000
# Tip. 다음도 동일한 효과
curl http://localhost:3000

# 현재의 nginx 설정으로는 (example.conf) 다음과 같이 접근 불가.
http://192.168.0.120:3000
# 이게 되려면, nginx 설정의 (example.conf) "server_name"에 example.com 대신 ip주소를
넣든지,
# 아니면 server { ... } 를 하나 더 구성하든지 해야 하였다.
# 익스프레스를 쓰면 상관 없는듯. (app.js / nodejs_tutorial.js 비교 필요.. 일단 스킵.. 추후
분석..)

# Tip. 단말에서 example.com을 찾기 위해 hosts 파일에 다음 항목을 추가
# 서버 주소를 찾는 과정은 DNS보다 hosts 파일이 우선권이 높기 때문.
192.168.0.120example.com

# 윈도우의 경우 다음의 위치 참고.
C:\Windows/System32/drivers/etc/hosts
# 리눅스나 맥의 경우 다음의 위치를 참고.
/etc/hosts

#Tip. vscode를 사용하는 경우, vscode가 원격서버의 웹 결과를 로컬로 리다이렉션.
# 웹서버에 vscode로 원격접속하여 nodejs app.js를 실행하는 경우 다음과 같은 메시지 팝업.
Your service running on port 3000 is available. [See all forwarded
ports](command:~remote.forwardedPorts.focus)
# 이 경우 로컬의 브라우저에서 다음과 같이 접속해도 동일한 결과를 얻을 수 있음.

```

---

---

http://localhost:3000

②. DB server (MongoDB)

# 몽고디비 소프트웨어 패키지 설치 (우분투 기준)

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -  
sudo apt update
```

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu  
xenial/mongodb-org/4.2 multiverse" | sudo tee  
/etc/apt/sources.list.d/mongodb-org-4.2.list
```

```
sudo apt install -y mongodb-org
```

###

# 라즈베리파이의 경우 위와같이 번거롭게 안해도.. # 간단하게 다음과 같이..

# 몽고디비 소프트웨어 패키지 설치 (라즈비안 기준)

```
% sudo apt-get install mongodb-server # 설치
```

```
% sudo systemctl status mongodb # 서비스 상태 확인
```

```
% sudo systemctl enable mongodb # 서비스 등록
```

```
% netstat -antp # 서비스 포트확인 # 몽고디비 서비스 default port는 27017
```

```
% sudo mongo # 몽고 쉘 실행
```

```
% sudo mongo --version # 몽고 버전 확인
```

```
>> MongoDB shell version : 2.4.14 .. >> 버전이 너무 낮음.. 2.6이상은  
되야..몽고커넥션..
```

```
>> 4.45 버전 소스코드 컴파일 설치..
```

# 몽고디비 소스컴파일 설치시 다음 참고..

```
% sudo apt-get install libcurl4-openssl-dev
```

```
% python3 -m pip install -r etc/pip/compile-requirements.txt
```

```
% python3 buildscripts/scons.py install-mongod --disable-warnings-as-errors
```

# 몽고디비 외부 접근 허용 설정..

```
% sudo vi /etc/mongodb.conf # 설정편집
```

```
bind_ip = 127.0.0.1 주석처리
```

```
bind_ip = 0.0.0.0 추가
```

```
sudo systemctl restart mongodb # 서비스재시작
```

# 아래와 같은 에러 발생시

```
'Server at localhost:27017 reports maximum wire version 0, but this version of the  
Node.js Driver requires at least 2 (MongoDB 2.6)'
```

# pymongo 버전확인 후 다음 버전 설치

```
% pip install pymongo==3.4.0
```

###

# 몽고디비 시작

```
% sudo service mongod start
```

---

```
# 몽고디비 수행 상태 확인
% sudo service mongod status

# 버전 확인
% mongo --version

# 셸에서 몽고디비 실행 # 몽고디비 CLI
% mongo

# 데이터베이스 목록 확인 (데이터베이스 목록 출력)
>> show dbs

# 현재 사용중인 데이터베이스 확인
>> db

# 데이터베이스 선택 (데이터베이스 목록에 없다면 데이터베이스 생성)
>>use smart-farm

# 선택된 데이터베이스 세부정보 확인
>>db.stats()

# 컬렉션 (테이블) 확인
>>show collections # show tables 명령도 가능

# 컬렉션 (테이블) 삭제
>>db.test_collection.drop()

# 테스트 데이터 입력 # test_collection 컬렉션이 없으면 생성하고 해당 데이터 입력
>> db.test_collection.insert({"test_name":"ys", "test_numebr":1234})
>> db.test_collection.insert({"test_name":"sh", "test_numebr":4321})

# test_collection 컬렉션 문서 개수 확인 # 데이터가 몇 개인지 확인
>> db.test_collection.count()

# test_collection 컬렉션 문서(데이터) 확인 # 데이터 세부 내용 확인
>> db.test_collection.find()
>> db.test_collection.find().pretty() # 이쁘게 (가독성있게) 출력

# test_collection 컬렉션 문서(데이터) 검색 # test_name = ys 인 데이터 출력
>> db.test_collection.find({"test_name":"ys"})

# 테스트 입력 데이터 확인
>>db.test_collection.insert({"test_name":"ys", "test_numebr":1234})
```

---

---

```
# 현재 사용중인 db 삭제
>>db.dropDatabase()

# 종료
>>exit

# 도움말
>>help

# Tip. 개발자 선호에 따라 선택사항이 달라져도 크게 문제 되지 않음.
다른 버전 설치      4.2 -> 4.4
패키지 툴 선택      apt -> apt-get # apt와 apt-get은 호불호가 있음. 본인 취향에 따름.
시스템 툴 선택      sudo service mongod start
                   or sudo systemctl start mongod
```

---

### ③. mosquito 설치 및 시험 방법

---

```
# MQTT 브로커 실행을 위한 바이너리 설치 # 설치방법은 제어기 노드와 동일

# 모스키토 소프트웨어 패키지 설치
% sudo apt install mosquitto

# 기본 설정 파일 확인
% sudo vi /etc/mosquitto/mosquitto.conf

# Tip. 모스키토는 클라이언트도 제공하였다. 이 테스트베드는 파호를 활용하므로 필요 없음.
# 모스키토 클라이언트
% sudo apt install mosquitto-clients

# 클라이언트 구독 테스트
% mosquitto_sub -h 127.0.0.1 -t 'topic/test' # 브로커는 미리 설치..

# 클라이언트 발행 테스트
% mosquitto_pub -h 127.0.0.1 -t 'topic/test' -m "Hello Mosquitto!"
```

---

### ④. paho 설치 및 시험 방법

---

```
# 파호 파이썬 라이브러리 설치
% pip install paho-mqtt
#python3을 쓰는 경우에는.. pip3 install paho-mqtt

# 간단한 테스트 파일 작성 # 위치는 python이 실행되는 아무 곳이면 됨.
% vi pub.py # 예제소스 참고
```

---

---

```
% vi sub.py # 예제소스 참고

# MQTT 프로토콜 동작 테스트
% python sub.py          # sub 실행
% python pub.py          # 다른 터미널에서 pub 실행

# Tip. 파호를 통해 다양한 방법으로 Pub/Sub을 구현할 수 있음.
% python pubm.py         # 멀티플 퍼블리시..

# Tip. 루프 코드 없이 지정된 시간 주기로 반복하고 싶을 때..
% watch -n 1500 python pub.py
```

---

#### ⑤. nodejs, npm, express 설치 및 시험 방법

---

# 이미 웹서버를 설치했는데 이 부분이 필요한 이유는 여러가지가 있지만, 특히 서비스 안정성과 보안성 확보에 대한 측면이 강하였다.

# nginx는 일종의 프록시 웹서비스를 제공하고, 실제 서비스는 뒷단의 nodejs, express 등이 제공

```
# 노드제이에스 런타임 패키지 설치 및 버전 확인
# 이미 웹서버 설치 단계에서 수행했다면 스킵
% sudo apt install nodejs ; nodejs -v

# nodejs가 (4점대) 구버전이라면 의존성 문제를 회피하기 위해 삭제하고 최신 버전 설치..
# 최신버전은 PPA(personal package archive)를 통해 설치 가능
# 구버전일 경우 express, nodemon 등과의 의존성 문제 발생 가능..
% node -v
> v4.2.6 # 설치된 nodejs 버전, 현재 15.x.x가 최신 버전
% sudo apt remove nodejs nodejs-legacy

# npm도 버전업을 위해 삭제
% sudo apt remove npm

# 2021년 3월 현재 가장 최신 버전은 15점대 버전
# LTS 버전인 14점대 버전으로 설치.
% cd ~ # PPA추가 스크립트 코드를 다운로드 받기 위해 적당한 위치로 이동

# 다른 버전을 설치하려면 버전 번호 변경하여 다운로드
% curl -sL https://deb.nodesource.com/setup_14.x -o nodesource14_setup.sh

# PPA 추가 및 업데이트
% sudo bash nodesource_setup.sh

# 상기 셸스크립트가 권고하는 명령대로 설치..
```

---

---

```
% sudo apt-get install -y nodejs

# 버전 확인
% node -v
> v14.16.0 # 설치된 nodejs 버전

# PPA를 통해 Node.js를 설치하면 npm도 함께 설치됨.
% npm -v
> 6.14.11 # 설치된 npm 버전

# npm을 제대로 동작하기 위해 다음 패키지도 설치
% sudo apt-get install build-essential

# nodejs 동작 테스트를 위한 디렉토리 생성
# 동작확인을 위한 테스트이므로 적당한 이름으로 프로젝트 폴더 생성
% mkdir express
% cd express

# nodejs 동작 확인을 위한 간단한 테스트 코드 작성 # http.createServer() 함수 사용
% vi app.js # 소스코드 및 설정파일 참고

# nodejs http 테스트 코드 실행
% node app.js

# 브라우저를 통해 결과확인 # vscode 사용하는 경우 local에서도 가능
http://localhost:3000

# 터미널의 경우
% curl http://localhost:3000

# express 설치 # 이 명령을 통해 node_modules 폴더가 생성
% npm i express
# 이미 express가 설치되어 있으면 에러발생..
# 기존 모듈을 사용하거나 또는 지우고 새로 설치..

# express 테스트를 위한 코드 생성 # express() 함수 사용
% vi app2ex.js # 소스코드 및 설정파일 참고

# nodejs express 테스트 코드 실행
% node app2ex.js

# 브라우저를 통해 결과확인 # vscode 사용하는 경우 local에서도 가능
http://localhost:3000
```

---

---

```
# 터미널의 경우
% curl http://localhost:3000

# HTML 및 CSS 연동 테스트..
# 테스트를 위한 html 및 css 파일 작성
# 각각 다른 URL로 요청이 들어올 때 웹서비스가 제공하는 html이 다름
% vi index.html      # / 리소스를 요청하는 경우      # 소스코드 및 설정파일 참고
% vi about.html     # /about 리소스 요청의 경우      # 소스코드 및 설정파일 참고
% vi 404.html       # 요청한 리소스가 없는 경우      # 소스코드 및 설정파일 참고
% mkdir public
% vi public/index.css
# index.html 표현식 정의
# 소스코드 및 설정파일 참고

# express를 활용한 웹서비스 테스트를 위한 코드 생성 # express() 함수 사용
% vi app3html.js      # 소스코드 및 설정파일 참고

# nodejs express 테스트 코드 실행
% node app3html.js

# 브라우저를 통해 결과확인 # vscode 사용하는 경우 local에서도 가능
http://localhost:3000      # index.html 출력 / index.css 적용
http://localhost:3000/about # about.html 출력
http://localhost:3000/sdfjsdlkf # 404.html 출력
http://localhost:3000/abou # 404.html 출력

# nodemon를 활용하여 코드 수정시 웹서비스 자동 재기동 테스트..
# nodejs 테스트 프로젝트 초기화 (이 명령의 결과로 package.json 생성)
% npm init
# 간단히 테스트만 하는데, 세부사항을 넣기 귀찮을 때..
% npm init --y

# 생성된 package.json 파일 확인
% vi package.json      # 소스코드 및 설정파일 확인

# dependencies 항목에 express 확인
"dependencies": {
  "express": "^4.17.1"
}

# nodemon 설치
# package.json에 반영하기 위해서 --save-dev 옵션
npm i nodemon --save-dev
```

---

---

```
# devDependencies 항목에 nodemon 확인
"devDependencies": {
  "nodemon": "^2.0.7"

# scripts 항목에 실행코드 확인
"scripts": {
  "dev": "nodemon app3html.js"
},

# nodejs express 테스트 앱 실행
npm run dev          # scripts 항목에 정의된 실행코드 실행

# 테스트 코드 실행 확인
> nodemon app3html.js
```

# 이후 테스트 코드가(app3html.js) 수정될 때 마다 알아서 웹서비스를 재기동.

```
> nodemon app3html.js

[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app3html.js`
The server is listening on port 3000
[nodemon] restarting due to changes...
[nodemon] starting `node app3html.js`
The server is listening on port 3000
```

```
# Tip. npm 라이브러리 설치 (라이브러리 의존성 검사를 위해 --save 옵션)
# --save 옵션을 적용하면 pacakage.json의 dependencies 항목에 반영됨
# --save-dev 옵션을 적용하면 pacakage.json의 devDependency 항목에 반영됨
# 둘의 차이는 웹서비스용, 즉 서비스 제공에 필요한 라이브러리 모듈과 단지 개발과정에 필요한 라이브러리 모듈을 구분
```

```
# scripts 항목에는 실행코드를 지정
# package.json 파일에도 주석은 가능하지만, 웬만하면 주석을 달지 않기를 권고
# 주석은 한줄 주석 (//), 여러줄 주석 (/* */), (/** */) 등이 사용 가능
```

```
# Tip. node_modules 확인   # 설치된 라이브러리 모듈이 무엇지 확인
npm list
# 다음 명령들도 참고
npm ll
npm la
```

```
# Tip. npm 프로젝트 확인
npm show
```

---

```
# Tip. 라이브러리 모듈을 재설치 하고자 할 때
rm -rf node_modules
npm cache clean
npm i

-----

# express, npm 관련 기타 참고 사항

# Tip. express-generator를 활용한 웹서버 테스트
# express 및 express-generator 설치
# -g 는 npm install 에서 글로벌로 사용하겠다는 의미. 즉 어떤 경로에서든 사용가능함을 의미.
# /usr/bin 등의 모듈을 건드리므로 sudo로 실행
sudo npm install -g express
sudo npm install -g express-generator

# Tip. 특정 프로젝트가 아닌 글로벌 패키지로 설치를 위해서는 -g 옵션 사용
sudo npm install -g express >> /usr/local/lib/express@4.17.1
sudo npm install -g express-generator >> /usr/local/lib/express-generator@4.16.1

# Tip. 로컬이 아닌 전역 모듈로 설치 되었으면 삭제도 -g 옵션 필요
sudo npm uninstall -g express
sudo npm uninstall -g express-generator

-----

# express 프로젝트 생성을 통한 웹서비스 동작 테스트..
# 단지 이렇게도 웹서비스 제공이 가능함을 보여주는 예제.

# express 프로젝트 생성 및 이동
express ex-mqtt
cd ex-mqtt

# 의존성 모듈 설치
npm install

# 웹어플리케이션 실행
npm start

# 브라우저로 동작 확인
http://localhost:3000

-----
```

---

---

```
# 몽고디비 연결성 테스트.. # 몽고디비 설치는 DB 설치 단계 확인..
```

```
# 몽고디비 연결에 필요한 라이브러리 모듈 설치
```

```
% npm install express body-parser dotenv mongoose --save
```

```
# 이미 설치된 라이브러리 모듈이 있으면 에러.
```

```
# 해당 라이브러리 모듈을 빼고 설치 # 또는 지우고 설치
```

```
# express 테스트중에 이미 express를 설치했으므로 express를 제외한 나머지 설치
```

```
# 프로젝트 최상위 폴더에 환경파일 생성
```

```
% vi .env
```

```
PORT=4500
```

```
MONGO_URI=mongodb://localhost/sfarm
```

```
# 몽고디비 연결을 위한 실행코드 작성
```

```
% vi app4mongo.js # 소스코드 및 설정파일 참고
```

```
▷ 테스트베드 시험을 위한 express-mqtt 테스트..
```

```
# 기존 작성된 package.json을 활용하여 동작 테스트
```

```
# 적당한 위치에 프로젝트 폴더 생성
```

```
% mkdir express-mqtt
```

```
% cd express-mqtt
```

```
# express-mqtt/package.json 편집
```

```
% vi package.json # 소스코드 및 설정파일 참고
```

```
# 웹애플리케이션을 위한 express-mqtt 노드JS 모듈 생성
```

```
% npm i # 또는 npm install
```

```
# 웹애플리케이션을 위한 디렉토리 생성
```

```
% mkdir src
```

```
# 웹애플리케이션을 위한 nodejs 모듈 편집
```

```
% vi index.js # 소스코드 및 설정파일 참고
```

```
% vi app.js # 소스코드 및 설정파일 참고
```

```
% vi socketIO.js # 소스코드 및 설정파일 참고
```

```
% vi DB.js # 소스코드 및 설정파일 참고
```

```
% vi mqttClient # 소스코드 및 설정파일 참고
```

```
# 센서 및 구동기 노드 데이터 송수신을 위한 스크립트 편집
```

```
% vi public/script/script.js # 소스코드 및 설정파일 참고
```

```
# nodejs 테스트 프로젝트 시작 및 웹 브라우저를 통해 웹애플리케이션 동작 확인
```

---

---

npm run dev

```
> express-mqtt@1.0.0 dev /home/osboxes/iot-lite.org/express-mqtt
> nodemon --delay 3 .

[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node .`
(node:23704) Warning: Accessing non-existent property 'MongoError' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
Express Server Running on localhost:5001
✔MQTT Connected
✔Subscribing AA
✔MongoDB Connected
```

---

가-6) [SW등록] 영상기반 생육 센서 하드웨어 및 해석 시스템 ('22)

(가) 연구 수행 개요

기존에 식물의 생육 과정을 측정하기 위한 방법으로는 실측과 영상을 이용한 측정 방법이 존재한다. 실측 방법은 사람이 수동으로 식물의 길이나 무게를 재는 방법을 의미하며, 영상을 이용한 측정 방법은 비연속적인 식물의 생육 상태를 고가의 피노타이핑 기술을 적용하거나 컴퓨터 비전 기술을 이용하여 상태를 측정하는 것이었다.

하지만 식물의 생육의 경우 외부의 영향(빛, 수분, 영양분 등)에 즉각적인 반응을 하지 않기 때문에, 종래 기술들은 어느 한 시점에서 식물의 생육 상태를 판단하기에 적합하지만, 식물의 생육 과정을 연속성 있게 관측하기에는 이런 특이 사항과 외부 영향에 대한 고려가 부족하다는 한계에 직면하곤 한다.

(나) 연구 주요 내용

이러한 문제점을 해결하기 위하여 '식물의 생육 측정'을 위한 다음과 같은 시스템을 설계하였다. 식물 생육상태의 영상을 획득하기 위한 생육 센서)는 일반적으로 컴퓨팅을 하기위한 기본적인 프로세서, 메모리, 저장장치 이외에도, 영상을 획득하기 위한 카메라 모듈, 피사체와의 거리를 측정하거나 조도를 측정하기 위한 하나 이상의 센서 모듈, 각종 상태를 알려주거나 동작 명령을 내리기 위한 상태 알림 및 입력 모듈, 위치를 감지하기 위한 모듈 및 네트워크에 접속하기 위한 유무선 통신 인터페이스로 구성된다.

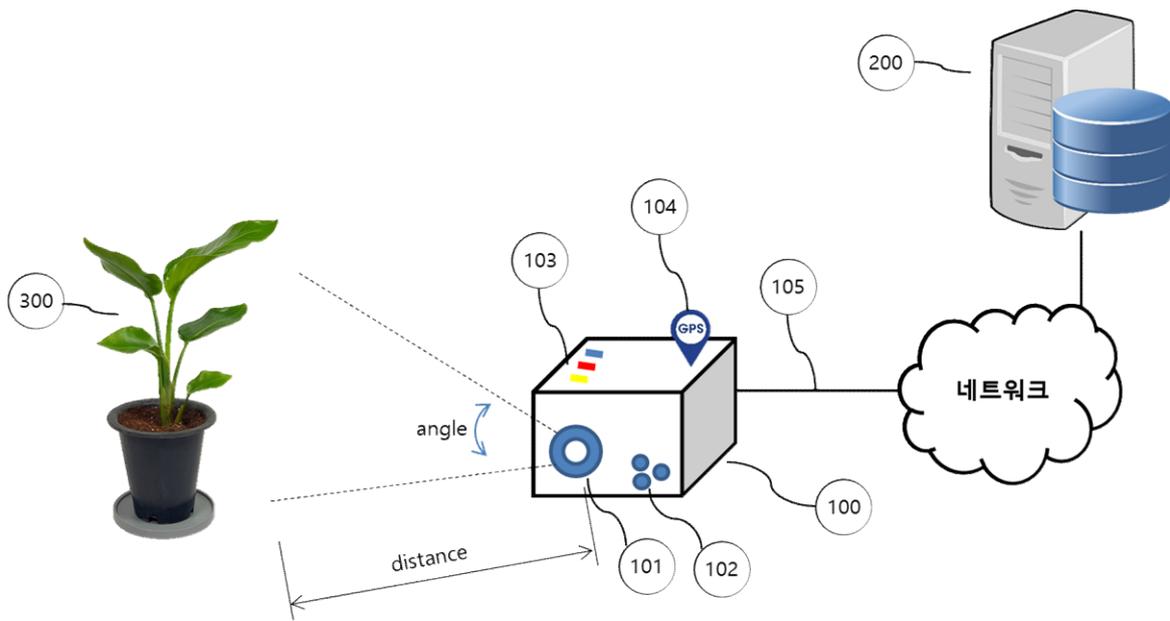


그림 1 - 영상장치를 이용한 식물 생육 센서 시스템 구성도

■ 해석 시스템 시제품

해석 시스템 구현을 위해 사용한 H/W 및 S/W 패키지는 다음과 같다.

◎ H/W 사양

- cpu : Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz
- 코어/쓰레드 : 6/12
- 메인보드 : TUF GAMING B460M-PLUS
- 메모리(GB) : 32
- HD : WD4T01
- gpu : GP107 [GeForce GTX 1050 Ti]

◎ S/W 패키지

- 1.1.1.1.1. - os : Ubuntu 20.04.2 LTS

- 
- 1.1.1.1.2. - cuda 10.2 / cudnn-11.0
  - 1.1.1.1.3. - anaconda : conda 4.7.12
  - 1.1.1.1.4. - python : Python 3.7.4
  - 1.1.1.1.5. - python package : requirements.txt
  - 1.1.1.1.6. - 학습 관련 python 주요 package

- . numpy
  - . pandas
  - . scikit-learn
  - . torch==1.7.1+ cu110 torchvision==0.8.2+ cu110 torchaudio==0.7.2
  - . mmcv-full
- 

해석 시스템의 실행은 다음과 같이 학습하기, 개체 검출하기, 동일 개체 찾기 등의 순으로 이루어진다.

#### 1) 학습하기

1.1.1.1.7. 이미지와 annotation을 이용하여 학습을 수행한다.

1.1.1.1.8.

---

```
# 실행방법
% python training.py
```

---

이 때 학습을 위한 학습 대상 이미지를 저장하는 공간은 다음과 같다.

- 학습 이미지 : 디렉토리(image > train)내 84개
- 검증 이미지 : 디렉토리(image > val)내 21개

학습을 수행할 경우, 디렉토리(work\_dir)에 학습된 모델 epoch\_36.pth를 생성하며, segmentation : 79.5% 정확도(IoU=0.50:0.95, area = all)를 보이는 것을 확인하였다.

추후 검출을 위해 모델을 복사 : work\_dir에서 model로 복사하도록 한다.

#### 2) 개체 검출하기

1.1.1.1.9. 1)에서 학습된 모델을 사용하여 개체(있)을 검출한다.

1.1.1.1.10.

---

```
# 실행방법
% python inference.py
```

---

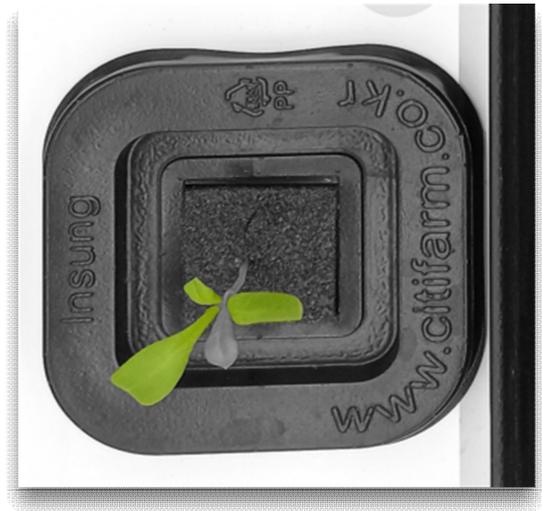
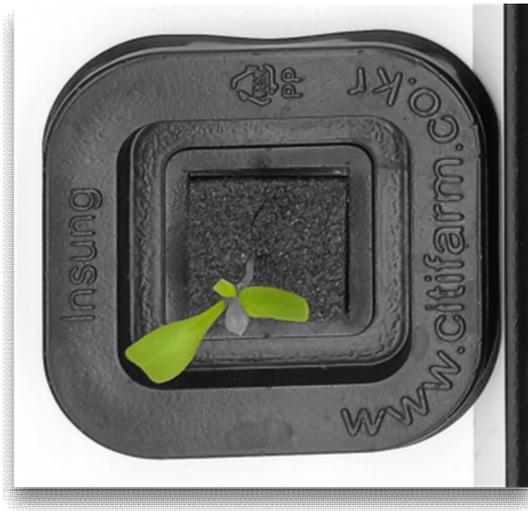
1.1.1.1.11.

1.1.1.1.12.

1.1.1.1.13. 검출대상 이미지는 동일작물에 대하여 이미지가 저장된 디렉토리 (image > target)내의 21개 이미지를 사용하며, 검출된 결과는 디렉토리(image > detected)내 21개가 저장됨을 확인하였다.

1.1.1.1.14. 이 때 흑백은 검출되지 않은 부분, color로 나오는 부분은 검출된 부분이다.

1.1.1.1.15.



1.1.1.1.15.1.

1.1.1.1.15.2. 검출된 객체의 segmentation 이외의 다른 정보들을 얻기 위해 csv형태로 저장하였다.

1.1.1.1.15.3. 실행 예) result > result\_bbox\_segmentation.csv

```

shoot_dt,obj_id,sub_obj_seq,lt_x,lt_y,rb_x,rb_y,threshold,center_x,center_y,
20211007090024,C49_L04_001,0,904,914,960,960,0,932.0,937.0,56,46,1471
20211007090024,C49_L04_001,1,969,922,1114,985,0,1041.5,953.5,145,63,6661
20211007090024,C49_L04_001,2,789,955,955,1114,0,872.0,1034.5,166,159,9040
20211007140024,C49_L04_001,0,897,911,956,960,0,926.5,935.5,59,49,1654
20211007140024,C49_L04_001,1,781,960,954,1134,0,867.5,1047.0,173,174,10157
20211007140024,C49_L04_001,2,965,936,1081,995,0,1023.0,965.5,116,59,5556
20211007050024,C49_L04_001,0,973,930,1093,988,0,1033.0,959.0,120,58,5630
20211007050024,C49_L04_001,1,786,973,938,1116,0,862.0,1044.5,152,143,9709
20211007050024,C49_L04_001,2,912,940,969,982,0,940.5,961.0,57,42,741
20211007050024,C49_L04_001,3,913,952,951,980,0,932.0,966.0,38,28,545
20211012142049,C49_L04_001,0,588,694,767,876,0,677.5,785.0,179,182,17017
20211012142049,C49_L04_001,1,814,652,913,816,0,863.5,734.0,99,164,9585
20211012142049,C49_L04_001,2,822,933,957,1247,0,889.5,1090.0,135,314,29595
20211007060024,C49_L04_001,0,970,926,1110,988,0,1040.0,957.0,140,62,6461
20211007060024,C49_L04_001,1,787,969,941,1112,0,864.0,1040.5,154,143,9236
20211007060024,C49_L04_001,2,910,946,948,971,0,929.0,958.5,38,25,590
20211007040024,C49_L04_001,0,778,983,938,1125,0,858.0,1054.0,160,142,10445
20211007040024,C49_L04_001,1,968,942,1078,1001,0,1023.0,971.5,110,59,5055
20211007040024,C49_L04_001,2,910,935,976,988,0,943.0,961.5,66,53,1014
20211006150024,C49_L04_001,0,802,947,954,1089,0,878.0,1018.0,152,142,8559
20211006150024,C49_L04_001,1,970,931,1075,988,0,1022.5,959.5,105,57,4452
20211006150024,C49_L04_001,2,934,921,969,951,0,951.5,936.0,35,30,713
20211007150024,C49_L04_001,0,895,915,954,961,0,924.5,938.0,59,46,1539
20211007150024,C49_L04_001,1,782,963,952,1133,0,867.0,1048.0,170,170,10337
20211007150024,C49_L04_001,2,962,939,1077,997,0,1019.5,968.0,115,58,5466
20211007110024,C49_L04_001,0,966,930,1104,990,0,1035.0,960.0,138,60,6419

```

- (a) shoot\_dt : 촬영년월일시분초
- (b) obj\_id : 작물ID
- (c) sub\_obj\_seq : 잎 개체 순번
- (d) lt\_x : bbox 좌상단 x좌표
- (e) lt\_y : bbox 좌상단 y좌표
- (f) rb\_x : bbox 우하단 x좌표
- (g) rb\_y : bbox 우하단 y좌표
- (h) threshold : threshold
- (i) center\_x : bbox 중심 x좌표
- (j) center\_y : bbox 중심 y좌표
- (k) b\_width : bbox 너비
- (l) b\_height : bbox 높이
- (m) area\_pix : 면적(단위 pixel)

#### 1.1.1.1.15.4.

#### 3) 동일 개체(잎) 찾기

1.1.1.1.16. 사용자가 선택한 잎을 대상으로 하여 가장 오래전에 찍은 사진으로부터 시간 순으로 다음 이미지를 찾아 동일 개체를 찾고 면적 정보를 저장한다.

```

obj_id,shoot_dt,sub_obj_seq,lt_x,lt_y,rb_x,rb_y,area_pix,inc_area,inc_area_rate,time
C49_L04_001,2021-10-06 14:00:24,0,803,947,954,1088,8420,,,
C49_L04_001,2021-10-06 15:00:24,0,802,947,954,1089,8559,139,101.65083135391923,1.0
C49_L04_001,2021-10-06 16:00:24,0,800,950,954,1093,8956,397,104.6383923355532,1.0
C49_L04_001,2021-10-06 17:00:24,0,801,954,956,1095,9230,274,103.05940151853505,1.0
C49_L04_001,2021-10-07 04:00:24,0,778,983,938,1125,10445,1215,113.16359696641388,11.0
C49_L04_001,2021-10-07 05:00:24,1,786,973,938,1116,9709,-736,92.95356629966491,1.0
C49_L04_001,2021-10-07 06:00:24,1,787,969,941,1112,9236,-473,95.12823153774848,1.0
C49_L04_001,2021-10-07 07:00:24,2,790,953,954,1108,9101,-135,98.53832828064097,1.0
C49_L04_001,2021-10-07 08:00:24,2,789,955,955,1111,8820,-281,96.91242720580156,1.0
C49_L04_001,2021-10-07 09:00:24,2,789,955,955,1114,9040,220,102.49433106575965,1.0
C49_L04_001,2021-10-07 10:00:24,2,789,955,954,1115,9284,244,102.69911504424778,1.0

```

#### 1.1.1.1.17.

#### 1.1.1.1.19. # 실행방법

#### 1.1.1.1.20. % python tracking.py

#### 1.1.1.1.18.

#### 1.1.1.1.21.

1.1.1.1.21.1.

1.1.1.1.21.2. 프로그램을 실행하고 피사체에 대해 제시된 숫자(검출된 잎의 개수 만큼 나옴)를 선택한다.

- ex) 0, 1, 2 중 선택 > 2

1.1.1.1.21.3.

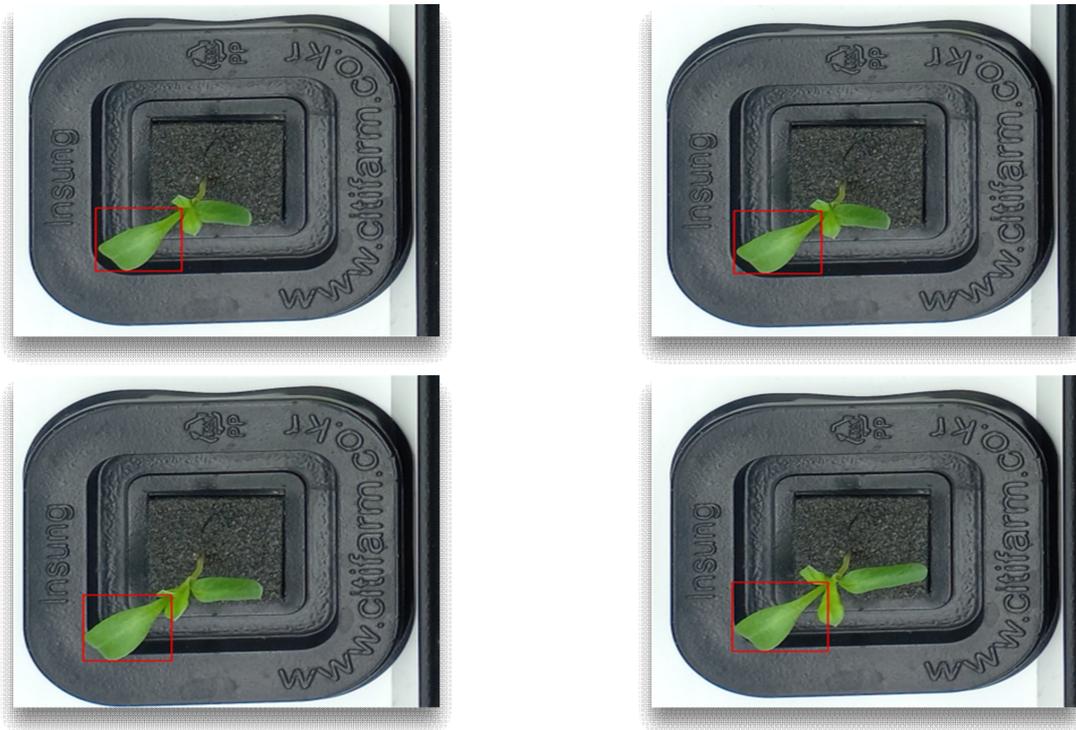
1.1.1.1.21.4. 대상 파일은 2)의 결과인 result > result\_bbox\_segmentation.csv 파일 사용한다.

1.1.1.1.22. 이를 수행한 결과는 동일한 개체(잎)을 찾아 시각화한 이미지가 출력된다.

1.1.1.1.23. - image > tracking내의 18개 이미지

- 카메라의 위치가 변동되고 시간차가 큰 이미지 3개 제외

- ex)



1.1.1.1.23.1. 면적의 추이에 대한 이미지 이외의 여러 정보들을 csv형태로 저장한다.

1.1.1.1.23.2. 실행 예) result > result\_area.csv

1.1.1.1.23.3.

1.1.1.1.23.4.

. : 시간)

```

obj_id,b_shoot_dt,b_sub_obj_seq,shoot_dt,sub_obj_seq,inc_area,inc_area_rate,time_in_hour
C49_L04_001,2021-10-06 14:00:24,0,2021-10-06 15:00:24,0,139,101.65083135391923,
C49_L04_001,2021-10-06 15:00:24,0,2021-10-06 16:00:24,0,397,104.6383923355532,1
C49_L04_001,2021-10-06 16:00:24,0,2021-10-06 17:00:24,0,274,103.05940151853505,
C49_L04_001,2021-10-06 17:00:24,0,2021-10-07 04:00:24,0,1215,113.1635969664138,
C49_L04_001,2021-10-07 04:00:24,0,2021-10-07 05:00:24,1,-736,92.95356629966491,
C49_L04_001,2021-10-07 05:00:24,1,2021-10-07 06:00:24,1,-473,95.12823153774848,
C49_L04_001,2021-10-07 06:00:24,1,2021-10-07 07:00:24,2,-135,98.53832828064097,
C49_L04_001,2021-10-07 07:00:24,2,2021-10-07 08:00:24,2,-281,96.91242720580156,
1.1.1.1.24. C49_L04_001,2021-10-07 08:00:24,2,2021-10-07 09:00:24,2,220,102.49433106575965,
C49_L04_001,2021-10-07 09:00:24,2,2021-10-07 10:00:24,2,244,102.69911504424778,
C49_L04_001,2021-10-07 10:00:24,2,2021-10-07 11:00:24,2,99,101.06635071090047,1

```

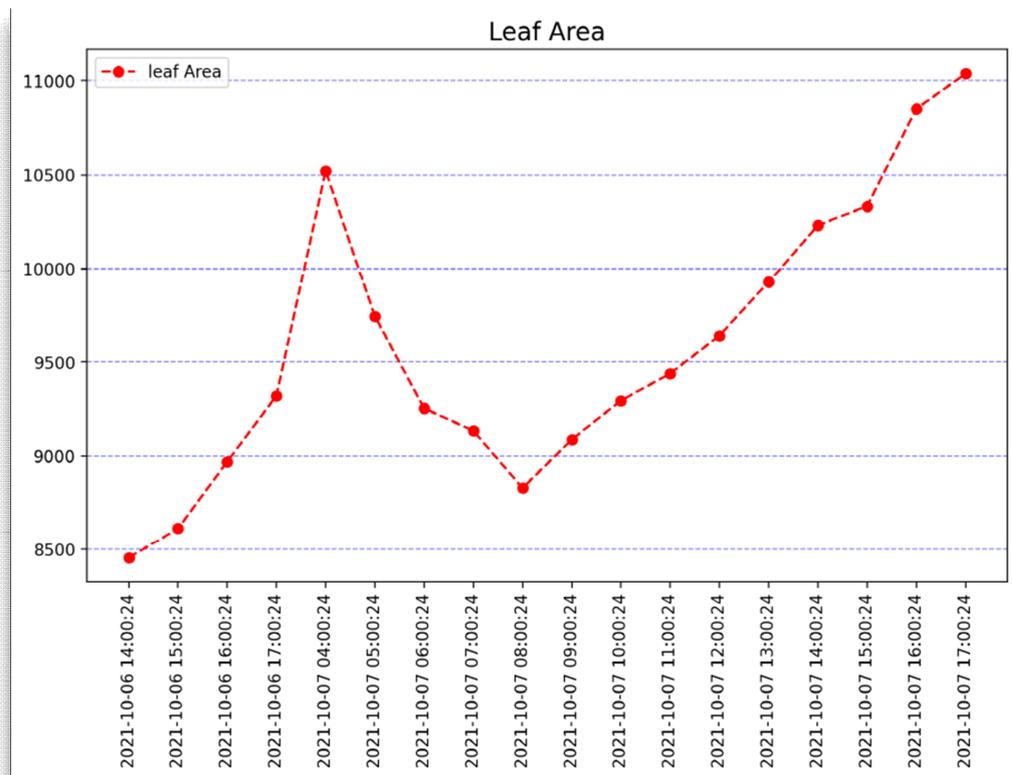
1.1.1.1.25.

- (a) obj\_id : 작물ID
- (b) shoot\_dt : 촬영일시
- (c) sub\_obj\_seq : 잎 개체 순번
- (d) lt\_x : bbox 좌상단 x좌표 .
- (e) lt\_y : bbox 좌상단 y좌표
- (f) rb\_x : bbox 우하단 x좌표
- (g) rb\_y : bbox 우하단 y좌표
- (h) area\_pix : 면적(단위 pixel)
- (i) inc\_area : 이전사진 대비 면적증감값
- (j) inc\_area\_rate : 이전사진 대비 면적증감율
- (k) time\_in\_hour : 이전사진에서 경과된 시간(단위

1.1.1.1.26.

1.1.1.1.27.

1.1.1.1.28.



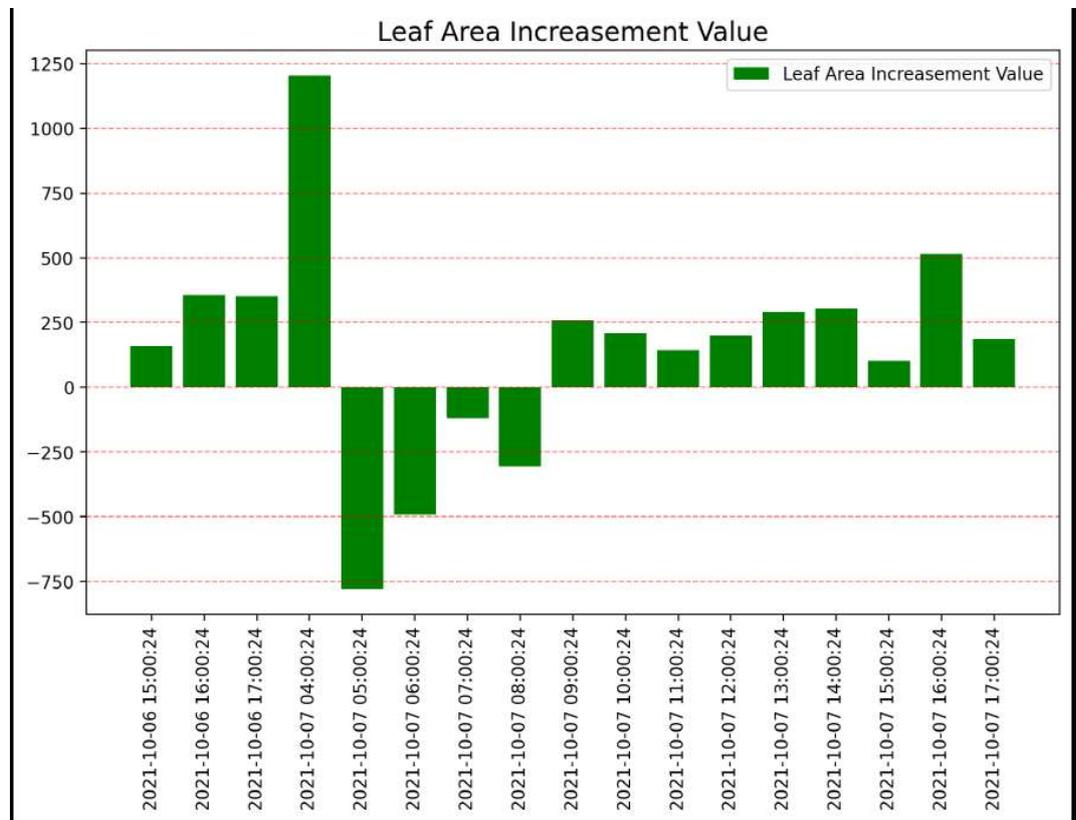
1.1.1.1.28.1.

면적증감의 추이를 csv파일로 저장하여 면적 증감 추이에 대한 시각화를 수행할 경우 다음과 같은 결과를 얻을 수 있다.

1.1.1.1.28.2.

1.1.1.1.28.3. 실행 예) result > result\_inc\_dec.csv

- (a) obj\_id : 개체ID
- (b) b\_shoot\_dt : 이전 사진의 촬영일시
- (c) b\_sub\_obj\_seq : 이전 사진의 개체(잎) 순번
- (d) shoot\_dt : 촬영일시 .
- (e) sub\_obj\_seq : 개체(잎) 순번
- (f) inc\_area : 면적 증가값
- (g) inc\_area\_rate : 면적 증가율
- (h) time\_in\_hour : 이전사진에서의 경과 시간(단위 : 시간)



1.1.1.1.28.4.

### 가-7) 인공지능 기반 스마트팜 테스트베드 운용 ('19)

최근 스마트팜을 통해 획득되는 작물 생육 및 환경에 관한 데이터는 작물 수확예측을 위한 자료로 활발히 활용되는 추세이다. 인공지능 기술이 발달함에 따라, 스마트팜을 통해 획득한 데이터를 활용하여 작물 수확예측을 위한 기술이 선보임에 따라, 점차 관련 기술 수집 및 조사가 필요해지고 있다. 국내 원예와 관련한 다수 기관(농진청, 농정원 및 학교)에서는 비닐하우스 등과 같이 인공적으로 작물의 성장 환경 조절이 가능한 공간에서의 농업을 통하여, 식물의 생육과 환경에 관련한 많은 데이터를 확보하고 있다. 이와 같은 이유로 이들 작물 생육과 환경에 관련한 데이터의 확보는 스마트팜 농가의 생산성 향상을 위해 매우 필요하다.

본 연구에서는 확보한 작물 생육과 환경에 관한 데이터를 토대로 텐서플로 등과 같이 공개형 AI 알고리즘을 활용하여, 작물 수확예측 시뮬레이션이 가능한지에 대한 PoC를 수행할 수 있도록, 작물의 생육 및 환경 데이터를 수집하고 분석할 수 있는 테스트베드를 개발하는 것을 목적으로 한다. 본 연구에서는 작물의 효율성과 경제성을 고려하여 인공지능 기반 스마트팜 테스트베드의 대상 작물로서 병풀과 아이스플랜트를 대상 작물로 선정하였고, 이를 위한 테스트베드 육종 계획을 구성하였다.

#### (가) 병풀(*Centella asiatica*) 광질 및 배양액 농도에 따른 데이터 분석

##### ■ 재료 및 방법

- 식물재료
  - 병풀의 모주로부터 2~3마디를 자른 뒤 발근시킨 후 3주 육묘하여 사용
  - 묘를 스펀지에 끼워 수경 재배시스템에 정식
  - 육묘조건 : Hoagland(MB cell, Korea) / EC : 1.5dS·m<sup>-1</sup> / 광 : White LED
- 실험구 처리조건
  - 배양액 종류 및 농도 : Hoagland(MB cell, Korea) / EC: 1.0, 2.0 및 3.0dS·m<sup>-1</sup>
  - 수경재배법 : 박막수경(NFT, Nutrient film technique)
  - 광원 : LED ①Red90 : Blue10, ②Red70 : Blue30, ③Red50 : Blue50
  - 광도 : 150μmol·m<sup>-2</sup>·s<sup>-1</sup>
  - 온도 / 습도 : 25±1°C, 65±5%

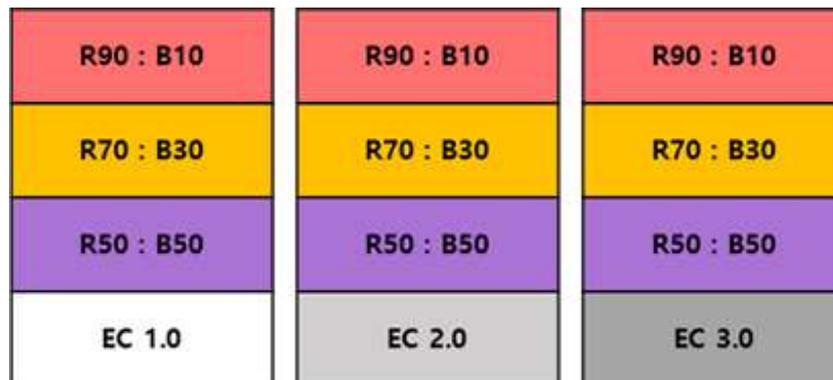


그림 1 - 병풀의 재배단 모식도

##### ■ 데이터 수집 및 분석

- 생육조사
  - 1차 : 처리 전 지상부 길이 측정
  - 2차 ~ 5차 : 처리 1주차부터 4주차(육묘 4 ~ 7주차)까지 매주 측정
  - 조사항목 : 초장, 엽면적, 엽수, 지상부 및 지하부의 생체중과 건물중\*
    - \* 성분분석과 향산화능 분석을 위하여 동결건조 후 건물중 측정
  - 모든 처리구의 3개체씩 측정
- 성분 함량분석 : 아시아티코사이드(asiaticoside) 측정
- 향산화능 분석 : DPPH 자유 라디칼 소거활성 측정, SOD 활성 측정

- 유전자 분석 : 아시아티코사이드(asiaticoside) 생합성 관련 유전자 발현 분석
- 통계분석
  - SPSS Statistics(ver. 24.0, SPSS Inc.,Chicago, IL, USA)를 이용하여 수행
  - one-way ANOVA 분석 후 p<0.05일 때 사후 분석으로 Duncan's multiple range test하여 유의성 검증

표 1 - 병풀 식물 생육조사 연구수행 일정

parameter \ WAP*	1	2	3	4**	5	6	7
육묘	○	○	○				
엽장, 엽수, FW, DW				○	○	○	○
아시아티코사이드				○			○
항산화물질				○			○
유전자 분석				○			○

\*Week after planting(9/15)

\*\*initiation of light and EC treatments (4, 5, 6, 7주차 생육조사)

### Effects of light intensity on growth and accumulation of triterpenoids in three accessions of Asiatic pennywort (*Centella asiatica* (L.) Urb.)

○ 목적  
- *C. asiatica*의 토양 재배 시 적정 광도 구명

○ 방법  
가. 재배방식/재배기간: 토양 이식/90일  
나. 온도/생육조건: 24-27°C, 70-85% 습도  
다. 광조건: sunlight (631 μmol·m<sup>-2</sup>·s<sup>-1</sup>), 50% (925 μmol·m<sup>-2</sup>·s<sup>-1</sup>), 및 20% (633 μmol·m<sup>-2</sup>·s<sup>-1</sup>) 광 투과율의 점은 원 밑

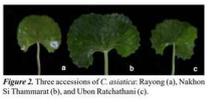


Figure 2. Three accessions of *C. asiatica*: Rayong (a), Nakhon Si Thammarat (b), and Ubon Ratchathani (c).

Accession	PPFD (μmol m <sup>-2</sup> s <sup>-1</sup> )	Leaf area (cm <sup>2</sup> )	Petiole length (cm)	Fresh weight (g/m <sup>2</sup> )	Dry weight (g/m <sup>2</sup> )
Nakhon Si	933	11.21 b	2.94 d	2.94 d	2.94 d
	362	13.87 d	5.75 b	1.45 b	2.13 b
	93	16.72 f	11.15 c	0.83 a	100.52 ab
Rayong	933	6.79 a	7.23 b	2.02 c	380.94 f
	362	9.96 b	9.89 c	1.22 ab	168.19 b
	93	12.87 cd	14.7 d	0.69 a	82.0 a
Ubon Ratchathani	933	7.45 a	3.96 c	3.96 c	326.96 e
	362	10.71 b	9.36 c	1.66 bc	221.14 d
	93	14.36 d	10.16 bc	0.82 a	104.15 b

Accessions	PPFD (μmol m <sup>-2</sup> s <sup>-1</sup> )	Asiaticoside (μg/m <sup>2</sup> )	Madecassic acid (μg/m <sup>2</sup> )	Asiatic acid (μg/m <sup>2</sup> )	Madecassic acid (μg/m <sup>2</sup> )	Triterpenoids acid (μg/m <sup>2</sup> )
Nakhon Si	933	2.77 d	2.73 d	0.343 c	0.656 f	28.77 f
	362	2.36 c	2.64 c	0.380 c	0.581 e	13.28 d
	93	1.83 b	1.71 b	0.154 a	0.505 c	4.56 b
Thammarat	933	2.88 d	2.82 d	0.200 bc	0.688 f	19.67 e
	362	1.26 a	1.07 a	0.735 d	0.772 d	6.47 c
	93	1.21 a	1.21 a	0.295 a	0.447 b	2.55 a
Ubon	933	3.50 d	3.30 d	0.343 c	0.647 e	38.82 g
	362	2.82 d	2.26 c	0.173 ab	0.615 de	12.69 d
	93	2.04 b	1.63 b	0.126 a	0.427 ab	4.75 b

(Journal of Food, Agriculture & Environment, 2011)

### Growth and centelloside production in hydroponically established medicinal plant *Centella asiatica* (L.)

○ 목적  
- *C. asiatica* 재배를 위한 수경 생산기술을 표준화하고 활용하는 것을 목표로 함.  
- 수경 재배 기반 기술은 토양에서 재배된 식물과 비교하여 제어된 환경에서 생육되므로 재현 가능성이 높은 생산 시스템 원료의 대량 생산을 수월을 예측이 가능하고 품질 개선이 보다 용이하다는 장점이 있음.

○ 방법  
가. 재배방식/재배기간: 70일 (10일마다 약액 교체)  
나. 온도/생육조건: 20°C/EC 1.5 dS/m, pH 6.2, Hogland and Amon salt solution

Culture age (days)	Fresh wt. of whole plant (g)	Dry wt. of whole plant (g)	Growth index <sup>a</sup>	Centelloside content in leaves (mg g <sup>-1</sup> dry wt.)			
				Madecassic acid	Asiaticoside	Madecassic acid	Asiatic acid
7	0.843 ± 0.254	0.4327 ± 0.106	16.23 ± 3.09	3.5	0.3	46.2	4.4
14	1.02 ± 0.444	0.4927 ± 0.119	40.42 ± 13.99	0.7	0.2	30.8	2.4
21	1.319 ± 0.802	0.641 ± 0.159	74.94 ± 13.70	0.3	0.2	23	4.1
28	1.73 ± 1.077	0.911 ± 0.162	131.54 ± 14.0	0.2	0.7	17	3.4
35	1.849 ± 1.021	0.7409 ± 0.178	132.51 ± 33.92	0.1	1.0	28.8	5.0
42	1.96 ± 0.828	0.7969 ± 0.246	156.39 ± 36.9	11.0	1.2	36.6	6.3
49	2.115 ± 1.128	0.5227 ± 0.124	122.54 ± 16.45	0.2	0.6	18	3.3
56	2.46 ± 1.33	0.7366 ± 0.122	105.08 ± 11.08	0.4	0.2	6.2	1.5
63	1.382 ± 0.181	0.2841 ± 0.103	98.48 ± 16.54	0.6	0.2	6.2	1.8
70	0.847 ± 0.521	0.1943 ± 0.007	51.64 ± 4.28	0.4	0.1	4.2	1.0

Analysis of variance (ANOVA) using REED, replicates=10, treatments=10 for the three parameters studied

Sources of variation	Degree of freedom	Mean sum-of-squares		
		Fresh weight	Dry weight	Growth index
Replications	9	0.041	0.041	290.564
Treatments	9	0.057	0.127	4152.664
Error	81	0.155	0.018	239.354

(Industrial Crops and Products, 2012)

### Growth and centelloside production in hydroponically established medicinal plant *Centella asiatica* (L.)

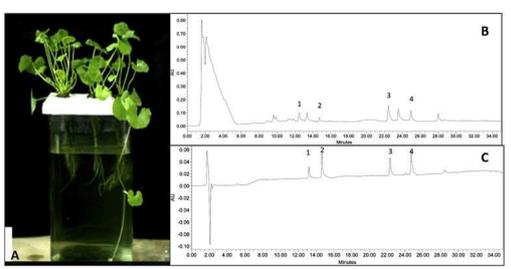


Figure 1. Cultivation of *C. asiatica* in hydroponics culture. (A) 42 days old hydroponic culture. (B) HPLC chromatogram of leaf extract of a 42 days old culture. (C) HPLC chromatogram of reference compounds: madecassic acid (1), asiaticoside (2), madecassic acid (3) and asiatic acid (4).

(Industrial Crops and Products, 2012)

### 병풀 재배의 보도자료

상처치료제 원료 '병풀' 온실 재배로 생산성 높여  
- 노지 재배보다 수량 4배... 물은 두둑에 직접 줘야 효과 -

□ 농촌진흥청(청장 김경규)은 화장품과 건강기능식품의 원료로 활용되며, 소득 작물로 떠오른 '병풀'의 새 재배 기술을 제시했다.

○ 병풀은 말기처분 기는종기(포복종)를 이용해 증식하며 2개월 정도면 자라는 잎과 지하줄을 2-3번까지 수확할 수 있다

○ 병풀은 노지와 비닐 온실에서 재배해 수량을 비교한 결과, 온실에서 재배했을 때 잎 수량은 1㎡당 1,056.1g 노지(247.2g)보다 4.27배 많았다.

○ 온실에서는 스프링클러(살수기)보다 두둑 고랑에 물이 차 있도록 저면관수(바닥 물대기) 했을 때 수량은 34%, 향 성분인 플라비놀과 플라보노이드는 각각 80%, 69% 높았다. 병풀의 유효성분인 아시아티코사이드도 82% 증가했다.

- 병풀은 제주도 저지대 해안가 또는 물가 주변에서 자생하는데, 저면 관수를 하면 자생지와 비슷한 습한 환경이 유지 수량이 높고 유효성분 함량도 높아지는 것으로 판단된다.

○ 병풀을 재배할 때 두둑은 20cm 정도로 높여주고, 고랑에 물은 1-2cm 잠기도록 대는 것이 좋다. 또한 비닐 등 덮지 않는 것이 증식에 유리하다.

(농촌진흥청)

### 식물공장 내 병풀 재배 시 적정 수경재배 방법 및 배양액 농도

○ 목적  
- 연중 안정재배를 위해 식물공장 내에서 적정 수경재배방법 및 배양액 농도를 제시하고자 함

○ 방법  
- 재배방법  
\* 병풀의 모주로부터 2~3마디를 자른 뒤, 발근된 모 사용  
\* 발근된 모를 파충충 스펀지에 끼워 담액수경  
- 재배조건: 온도 25±1°C, 습도 65±5%  
- 광원광도: 형광등, 120µmol·m<sup>-2</sup>·s<sup>-1</sup>  
- 수경재배: 담액식(Deep Flow Technique)  
- 배양액 및 농도: 아마자키배양액, 2.0dS·m<sup>-1</sup>

가. 병풀 재배방법 및 EC 농도에 따른 생육특성

재배 방법 (A)	EC 농도 (B)	생체중(g/부)	신로중(g/부)	질면적 (cm <sup>2</sup> )	SPAD value
1.0	4.3±0.12 b	0.7±0.02 b	0.5±0.03 b	0.1±0.01 b	56.3±3.82 b
1.5	5.0±0.25 b	0.9±0.02 b	0.6±0.02 b	0.1±0.01 b	60.0±4.50 b
2.0	6.7±0.23 a	1.4±0.12 a	0.8±0.07 a	0.2±0.02 a	60.4±3.11 a
2.5	6.3±0.31 a	1.5±0.23 a	0.8±0.07 a	0.2±0.02 a	65.4±5.65 a
3.0	2.0±0.21 b	1.3±0.20 c	0.3±0.07 c	0.2±0.02 a	38.3±4.62 c
3.5	3.9±0.23 b	1.8±0.16 bc	0.5±0.10 b	0.2±0.04 a	45.2±7.47 b
4.0	5.3±0.56 a	2.8±0.27 a	0.7±0.17 a	0.3±0.08 a	63.8±5.42 a
4.5	4.1±0.30 a	2.5±0.43 ab	0.6±0.09 ab	0.2±0.04 a	78.8±5.02 ab
5.0	4.1±0.30 a	2.5±0.43 ab	0.6±0.09 ab	0.2±0.04 a	44.1±4.31 a

다. 재배방법 및 EC 농도에 따른 병풀 기능성 물질 함량

총 폴리페놀 함량 (농촌진흥청, 2019)

총 플라보노이드 함량 (농촌진흥청, 2019)

### 식물공장 재배 병풀의 기능성 향상을 위한 수확 전 고농도 양액처리 효과

○ 목적  
- 일대형 식물생산시스템 내 담액수경재배 시 물리적 환경조건을 이용한 품질 향상 기술 개발

○ 방법  
- 처리시기: 수확 전 약 7일간  
- 양액종류: 아마자키배양액  
- 처리방법: 배양액 농도 3.0dS·m<sup>-1</sup>  
- 재배방법: 담액 수경재배  
- 병풀 정식 후 배양액 농도 2.0dS·m<sup>-1</sup>으로 재배하여 생산량 확보, 수확 7일 전 고농도 양액(3.0dS·m<sup>-1</sup>)을 공급

가. 고농도 양액처리에 따른 병풀 생육특성

배양액 농도 (dS·m <sup>-1</sup> )	생체중(g/부)	신로중(g/부)	질면적 (cm <sup>2</sup> )	SPAD value
2.0	5.6±0.36 a	0.7±0.04 a	72.7±2.69 a	60.3±4.2
3.0	5.2±0.19 ab	0.7±0.02 a	67.6±3.15 a	61.3±1.4
4.0	4.7±0.10 bc	0.6±0.02 a	54.2±3.15 b	62.9±1.1
5.0	4.4±0.17 c	0.6±0.01 a	51.2±1.16 b	57.6±3.3

나. 고농도 양액처리에 따른 병풀 기능성 물질 함량

총 폴리페놀 함량 (농촌진흥청, 2019)

총 플라보노이드 함량 (농촌진흥청, 2019)

### 비닐하우스 재배 병풀의 최적 이용부위, 성분추출 용매

○ 일반성분은 분 6.18% 조단백 27.14%, 조지방 14.11%, 탄수화물 44.86%, 회분 5.67%

Moisture	Crude Protein	Crude Fat	Crude Ash	Carbohydrate <sup>1)</sup>
6.18±0.37	27.14±0.16	14.11±0.19	5.67±0.13	44.86±1.03

○ 부위별 성분 및 생리활성소 일 > 줄기 > 잎자루 순으로 높으며, 악리성분인 Asiaticoside는 잎에만 3.44% 함유 되어 있음

구분	Total Polyphenol(%)		DPPH (Ascorbic acid mg /g d.b.)		Asiaticoside(%)
	잎	줄기	잎	줄기	
잎	1.42	1.40	31.02	3.44±0.01	
잎자루	0.16	0.16	7.23	No Detection	
줄기	0.87	0.35	24.02	No Detection	

○ 추출 용매별 항산화 성분은 50%에탄올 > 물 > 70%에탄올 > 100%에탄올 순으로 높았음

구분	100%에탄올	70%에탄올	50%에탄올	물
Total Polyphenol	0.80	1.50	1.98	1.88
Total Flavonoid	0.81	1.03	1.27	1.05

(농촌진흥청)

### 비닐하우스 재배 병풀의 생육 및 수량, 성분함량

○ 병풀 비닐하우스 재배조건  
- 재배환경: 흑색비닐피복, 두둑높이 20cm, 재식거리 30cm\*15cm  
- 재배방법: 4월 하순 병풀 근경을 잘라서 식재 후 3개월 간 재배 (7월 일 수확)  
- 관수방법: 스프링클러 이용 1주간격

○ 비닐하우스 병풀의 일 수량은 1,056.1g/m<sup>2</sup>으로 노지재배 247.2g/m<sup>2</sup> 보다 4.27배 많음

재배조건	엽장 (cm)	엽폭 (cm)	일수 (g/m <sup>2</sup> )	일무게 (g/m <sup>2</sup> )
비닐하우스	2.9	5.2	3,203	1,056.1
노지	2.1	3.7	1,592	247.2

(농촌진흥청)

그림 2 - 병풀 관련자료 수집분석

(나) Iceplant(Mesembryanthemum crystallinum) 광질 및 배양액 농도에 따른 데이터 분석

■ 재료 및 방법

- 식물재료
  - Iceplant 종자를 발아시켜 5주 육묘 후 사용
  - 육묘를 스펀지에 끼워 수경 재배시스템에 정식
  - 육묘조건 : Hoagland(MB cell, Korea) / EC : 1.5dS·m<sup>-1</sup> / 광 : White LED
- 실험구 처리조건
  - 배양액종류 및 농도 : Hoagland(MB cell, Korea) / EC : 1.5, 3.0 및 5.0dS·m<sup>-1</sup>
  - 수경재배법 : NFT(박막수경)
  - 광원 : LED ①Red90 : Blue10, ②Red70 : Blue30, ③Red50 : Blue50
  - 광도 : 150µmol·m<sup>-2</sup>·s<sup>-1</sup>
  - 온도 / 습도 : 25±1°C, 65±5%



그림 3 - 아이스플랜트의 재배단 모식도

■ 데이터 수집 및 분석

- 생육조사
  - 1차 : 처리 전 지상부 길이 측정
  - 2차 ~ 5차 : 처리 1주차부터 4주차(육묘 6~ 9주차)까지 매주 측정
  - 조사항목 : 초장, 엽면적, 엽수, 지상부 및 지하부의 생체중과 건물중\*
    - \* 성분분석과 항산화능 분석을 위하여 동결건조 후 건물중 측정
- 성분분석 : 피니톨(Pinitol), 이노시톨(Inositol) 측정
- 항산화능 분석 : DPPH 자유 라디칼 소거활성 측정, SOD 활성 측정
- 유전자 분석 : 피니톨(Pinitol)과 이노시톨(Inositol) 생합성 관련 유전자 발현 분석
- 통계분석
  - SPSS Statistics(ver. 24.0, SPSS Inc.,Chicago, IL, USA)를 이용하여 수행
  - one-way ANOVA 분석 후  $p < 0.05$ 일 때 사후 분석으로 Duncan's multiple range test하여 유의성 검증

표 2 - 아이스플랜트 생육연구수행 일정

parameter \ WAP*	1	2	3	4	5**	6	7	8	9
육묘	○	○	○	○	○				
엽장, 엽수, FW, DW						○	○	○	○
피니톨, 이노시톨						○			○
항산화물질						○			○
유전자 분석						○			○

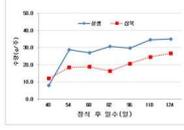
\*Week after planting

\*\*initiation of light and EC treatments (6, 7, 8, 9주차 생육조사)

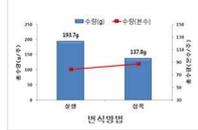
### 인공광 식물공장내에서 아이스 플랜트의 실생묘와 삼목묘의 생산성 비교

- 인공광 식물공장 내에서 아이스 플랜트의 실생묘와 삼목묘를 정식하여 재배 할 경우 정식 후 수확까지 전 40일까지 생육에서 실생묘에서 앞섰었다
- 정식 후 40일째부터 실생묘와 삼목묘 모두 수확이 가능하였으나, 조기 수확된 삼목묘에서 다소 많았음
- 정식 후 124일까지 전체 수확할 가운데 생치율에서 실생묘는 주당 193.7g, 삼목묘는 137.8g으로 실생묘에서 40.5% 높수되었으나, 전체 수확에서 주당 개수(분지수)는 실생묘가 78.6개, 삼목묘 87.0개로 실생묘가 10.7% 적었음

#### ○ 정식 후 일수에 따른 수량성 비교



#### ○ 총수량 비교



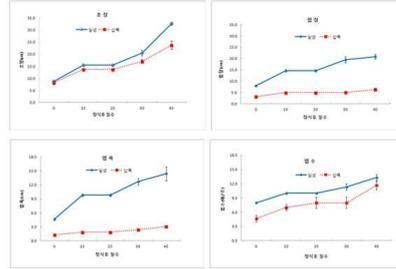
(농촌진흥청, 2013)

### 인공광 식물공장내에서 아이스 플랜트의 실생묘와 삼목묘의 생산성 비교

#### ○ 재배 개요

수경방식	배양액조성	조사항원	환경관리	재배기간
배지경 (플랜트)	내일원드 원배양구소 상투표준액	LED 155.2umol (F5B2-W1혼합)	- 온도 : 20°C - 광기 : 12h	131.30 ~ 6.6

#### ○ 초기생육 변화



(농촌진흥청)

### 인공광 이용형 Common Ice Plant 식물공장의 실용적 설계

#### ○ 적정 광도 수준 및 필요 전력 추정

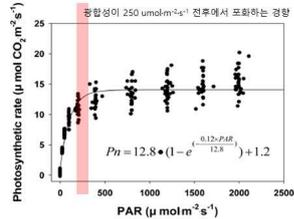


Fig. 1. Photosynthetic rate curve of common ice plant at 1,000µmol·mol<sup>-1</sup> of CO<sub>2</sub> concentration, 50-65% of relative humidity, and 25°C of leaf temperature.

이상적인 광합성량: 12.8µmolCO<sub>2</sub>m<sup>-2</sup>s<sup>-1</sup>  
 광포화점: 456µmol·m<sup>-2</sup>·s<sup>-1</sup>  
 광보상점 : 0µmol·m<sup>-2</sup>·s<sup>-1</sup>  
 → 인공광을 이용한 아이스플랜트 재배 시 적정광도: 120~200µmol·m<sup>-2</sup>·s<sup>-1</sup>

따라서, 형광등을 이용하는 완전제어형 식물공장에서 common ice plant 재배를 위한 최저 광도는 120µmol·m<sup>-2</sup>·s<sup>-1</sup>(8,880lx)이상이 되어야 한다.

(Protected Horticulture and Plant Factory, 2014)

### 아이스플랜트 기능성분 강화를 위한 전용 양액 및 환경조건 구명

#### ○ NaCl 처리 시기 구명

가. 재배방식/재배기간 : 박막수경/30일

나. LED광원 : Red 2: Blue 1: White 1, 150±10umol·m<sup>-2</sup>·s<sup>-1</sup>

다. 광주기(명기/암기): 12/12

라. 온도/상대습도: 20°C/EC 1.5 dS/m, pH 6.0, 일본원시처방액

마. NaCl 처리 시기: 수확 10일전, 15일전, 20일전에 80mM NaCl 첨가 후 생육 30일째 생육조사

표 1. NaCl 처리시기에 따른 생육 특성

처리시기	엽장 (cm)	엽수 (개)	생체중 (g/주)	건물중 (g/주)	건물률 (%)
수확 20일전	12.3a	11.8a	43.8a	3.6a	8.7
수확 15일전	10.5b	12.1a	37.7a	1.6a	4.4
수확 10일전	9.4c	10.8a	26.6b	1.3b	5.1
대조구	8.2d	10.9a	20.6b	1.2b	5.7

DMBRT(5%)

※ 수확 20, 15, 10일전 처리 : 처리시기별~수확일(9월 13일)까지 NaCl 처리

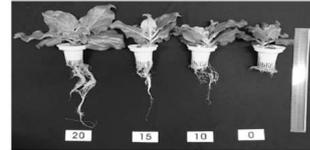


그림 2. 처리시기에 따른 식물체 변화

(농촌진흥청)

### 아이스플랜트 기능성분 강화를 위한 전용 양액 및 환경조건 구명

#### ○ NaCl 처리 시기 구명

가. 재배방식/재배기간 : 박막수경/30일

나. LED광원 : Red 2: Blue 1: White 1, 150±10umol·m<sup>-2</sup>·s<sup>-1</sup>

다. 광주기(명기/암기): 12/12

라. 온도/상대습도: 20°C/EC 1.5 dS/m, pH 6.0, 일본원시처방액

마. NaCl 처리 시기: 수확 10일전, 15일전, 20일전에 80mM NaCl 첨가 후 생육 30일째 생육조사

표 2. NaCl 처리시기별 식물체 무기성분 함량

처리시기	T-N (%)	P <sub>2</sub> O <sub>5</sub> (%)	K <sub>2</sub> O (%)	CaO (%)	MgO (%)	Na <sub>2</sub> O (%)
수확 20일 전	4.5	1.0	7.8	0.7	0.4	16.8
수확 15일 전	4.5	1.1	8.3	0.8	0.5	13.8
수확 10일 전	4.6	1.1	8.9	1.1	0.6	9.5
대조구	4.9	1.1	9.0	1.5	0.6	2.6

※ 수확 20, 15, 10일전 처리 : 처리시기별~수확일(9월 13일)까지 NaCl 처리

(농촌진흥청, 2018)

### 신선 아이스플랜트 생산증대를 위한 적정 NaCl 함량

#### ○ NaCl 처리 농도 구명

가. 재배방법 : 일본원시표준액 + NaCl 120mM ~ 200mM 첨가

나. 재배방식/재배기간 : DFT(박막수경)/3개월

다. LED광원 : Red 2: Blue 1: White 1, 100±10umol·m<sup>-2</sup>·s<sup>-1</sup>

라. 광주기(명기/암기): 12/12

마. 온도/상대습도: 19°C/EC 1.5 dS/m, pH 6.0

바. 주의사항

- 기본양액에 NaCl 240mM 이상을 첨가했을 때 오히려 생육과 수량이 감소하였음
- 아이스플랜트 3개월이상 재배하면 품질이 떨어지므로 3개월 주기로 정식하는 것이 유리함
- 아이스플랜트의 생육은 Accoris에 비해 DFT, NFT 방식으로 재배하는 것이 우수함

가. NaCl 농도에 따른 생육 특성

농도 (mM)	엽장 (cm)	엽수 (개)	생체중 (g/주)	건물중 (g/주)
대조구	10.5b	7.8abc	86.4b	5.2bc
80	11.6ab	6.4bc	150.0b	7.3ab
120	11.8ab	4.1d	369.9a	8.0ab
160	12.9a	8.1ab	345.1a	8.7a
200	12.1ab	6.2c	343.6a	8.6a
240	13.2a	8.5a	194.0b	7.8ab
320	12.0ab	9.3a	128.9b	3.5c

<sup>a</sup> Mean separation within columns by Duncan's multiple range test at P=0.05

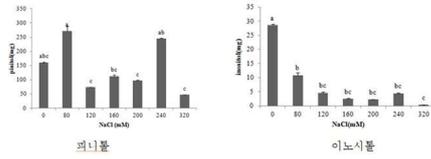
(농촌진흥청)

### 신선 아이스플랜트 생산증대를 위한 적정 NaCl 함량

나. NaCl 농도에 따른 식물체 변화



다. NaCl 농도별 식물체 기능성분 함량



(농촌진흥청, 2018)

### 아이스플랜트에 적합한 수경재배 방식 구명

○ 수경재배 방식 구명

가. 재배방법 : 일본원시표준액 + NaCl 160mM 첨가

나. 재배방식/재배기간 : DFT(담액수경), NFT(박막수경) 및 Aeroponics(분무경)

다. LED광원 : Red 2:Blue 1, 100±10umolm<sup>-2</sup>s<sup>-1</sup>

라. 광주기(광기/암기) : 12/12

마. 온도/상대습도 : 19°C/EC 1.5 dS/m, pH 6.0

재배방식	잎장 (cm)	잎수 (매)	생체중 (g/주)	건물중 (g/주)
DFT	12.9b	8.1b	345.1a	8.7a
NFT	17.0a	11.3a	303.9a	6.9a
Aeroponics	10.1c	9.2b	71.0b	3.1b

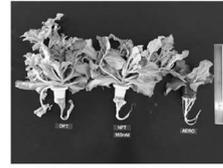


그림 4. NaCl 농도에 따른 식물체 변화

(농촌진흥청)

그림 4 - 아이스플랜트 관련자료 수집분석

## 5) IPR 및 학술성과

### 가) [국내특허] 국내특허 출원 및 등록

#### 가-1) [국내특허등록] 국내특허 등록 (5건)

본 사업의 연구 수행결과를 통해 다음과 같은 5건의 출원 특허에 대한 특허를 등록 결정받은 바 있다.

특허명	출원일	특허등록결정일
IoT 노드 자동 구성 시스템 및 방법	2017-11-28	2019-07-26
통합 제어 라우팅 시스템 및 그 방법	2018-07-09	2019-08-22
센서 교체가 가능한 범용 센서 노드 및 그 구성 방법	2018-11-28	2019-09-24
LCP를 이용하는 스마트온실 시스템	2018-11-28	2020-01-14
스마트팜 데이터셋 유효성 검증을 위한 수집 데이터 선별 장치 및 그 방법	2020-12-11	2022-10-18

(가) IoT 노드 자동 구성 시스템 및 방법 (출원:'17 , 등록:'19)

본 발명은 IoT 환경에서 센서, 구동기 등 노드(node)의 관리 및 데이터 획득과정을 용이하게 하는 IoT 노드 자동 구성 시스템 및 방법에 관한 것이다. 종래 기술에 따르면, 센서, 구동기 등 제품 별로 데이터가 저장되는 메모리의 주소(어드레스, address) 정보를 기초로 제어 응용(SW)을 설계하여 이용한다. 이러한 제어 응용을 이용하는 경우, 기정의된 제조사별 레지스터 주소를 기반으로 하여 각 노드의 데이터 획득을 위한 요청/응답이 이루어지게 되는데, 새로운 노드가 장착되는 등 노드의 변경이 발생한 경우, 제어 응용 자체를 새로 코딩하여야만 이용 가능하여 변경된 환경에 대한 적응성이 떨어지는 문제점이 있다.

본 발명은 이러한 문제점을 해결하기 위하여 제안된 것으로, 인스톨된 IoT 노드에 대한 제조사별 식별자와 레지스터의 매핑 테이블을 자동 구성하고, 식별자 기반의 요청을 어드레스 기반의 요청으로 자동 변환하고 그에 대한 응답을 수신함으로써, IoT 노드의 추가 등 변경이 일어난 경우에도 어플리케이션 자체의 수정이 불필요하며 범용성을 확보하는 것이 가능한 IoT 노드 자동 구성 시스템 및 방법을 제공하는 데 그 목적이 있다.

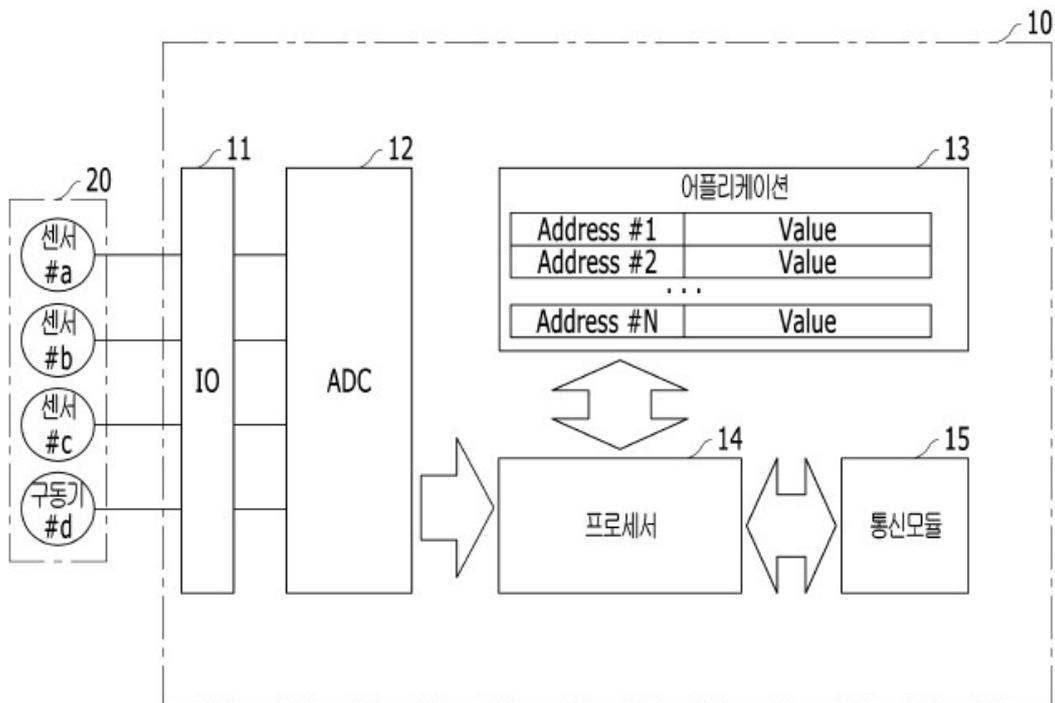


그림 181 - IoT 노드 자동 구성 시스템 및 방법 개념도

(나) 통합 제어 라우팅 시스템 및 그 방법 (출원:'18 , 등록:'19)

본 발명은 그룹화된 센서 및 단말의 그룹 제어를 위한 라우팅 시스템 및 그 방법에 관한 것이다. 종래 기술에 따르면, 사용자는 그룹화된 단말 제어를 위해 식별 체계 또는 주소 체계를 알고 있어야 하는데, 이러한 식별 체계 또는 주소 체계는 사용자에게 익숙하지 않은 문자 체계로서, 사용 편의성이 떨어지는 문제점이 있다. 또한, 종래 기술에 따른 무선 단말의 그룹화된 제어 수단은 라우터 접속 반경을 벗어난 그룹 내 단말을 제어할 수 없는 한계점이 있다. 본 발명은 이러한 문제점을 해결하기 위하여 제안된 것으로, 목적 그룹 지정에 있어 사용자가 기존 언어를 사용할 수 있고, 라우터 접속 반경을 벗어난 단말에 대한 제어가 가능한 통합 제어 라우팅 시스템을 제공할 수 있다.

본 발명에 따른 통합 제어 라우팅 시스템은 제어 대상 그룹 정보를 전송하는 제어단말과, 제어 대상 그룹 정보를 수신하고, 테이블 내에서 그룹 정보와 매칭되는 대표 좌표 정보를 추출하여 이를 전송하는 서버 및 대표 좌표 정보를 수신하여, 접속 반경 내 단말 중 메시지 전달 경로가 최단 거리가 되는 제1 단말을 조회하고, 제1 단말로 메시지를 전송하는 라우터를 포함하는 것을 특징으로 한다.

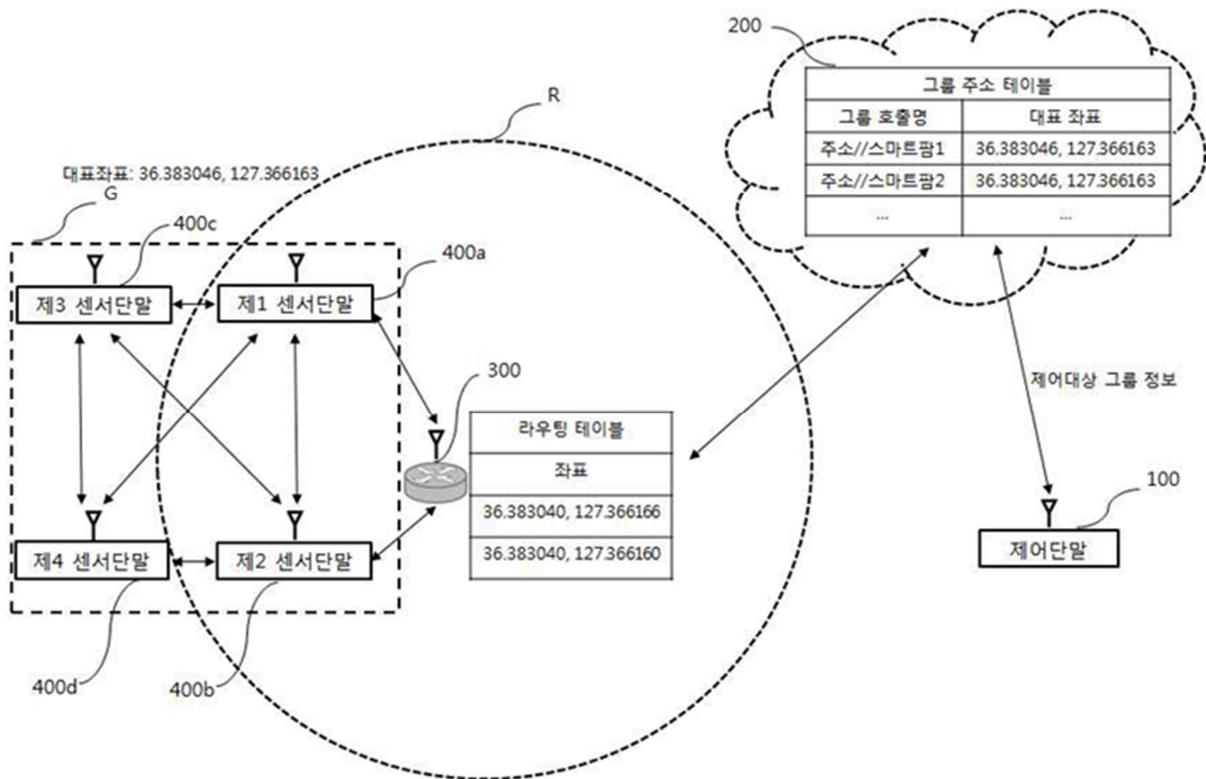


그림 182 - 위치기반의 스마트팜 통합 제어 라우팅 시스템 개념도

(다) 센서 교체가 가능한 범용 센서노드 및 그 구성 방법 (출원: '18, 등록: '19)

스마트온실은 기존 온실 기술에 ICT 기술을 접목하여, 원격 또는 자동으로 작물의 생육 환경을 적정하게 유지하고 관리할 수 있는 농장을 의미한다. 종래 기술에 따르면, 스마트 온실에 설치되는 센서노드는 센서 소자를 ADC, CPU 및 통신 모듈과 일체형으로 패키지로 구성하게 되어, 타 부품들에 비해 교체 시기가 빠른 센서 소자의 수명이 다함에 따라, 일체형 패키지 전체를 교환하여야 하는 문제점이 있다. 본 발명은 이러한 문제점을 해결하기 위하여 제안된 것으로, 사용자의 니즈에 따라 교체형으로 설치되는 센서 모듈 및 센서 변환값을 이용하여 맞춤형 센서노드로서 활용하는 것이 가능한 범용 센서노드 및 그 구성 방법을 제공하는데 그 목적이 있다.

본 발명에 따른 범용 센서노드는 센서 소자로부터 입력값을 획득하는 수신부와, 입력값을 이용하여 변환값을 획득하는 프로그램이 저장된 메모리 및 프로그램을 실행시키는 프로세서를 포함하고, 프로세서는 센서 타입에 따른 변환 규칙을 이용하여 변환값을 획득하는 것을 특징으로 한다. 본 발명에 따른 센서 교체가 가능한 범용 센서노드 구성 방법은 공통 센서노드 하드웨어의 I/O 포트에 센서 모듈을 접속하는 단계와, 센서 모듈로부터 센싱값을 획득하는 단계 및 센싱값에 대해 변환 규칙을 적용하여 변환값을 출력하는 단계를 포함하는 것을 특징으로 한다

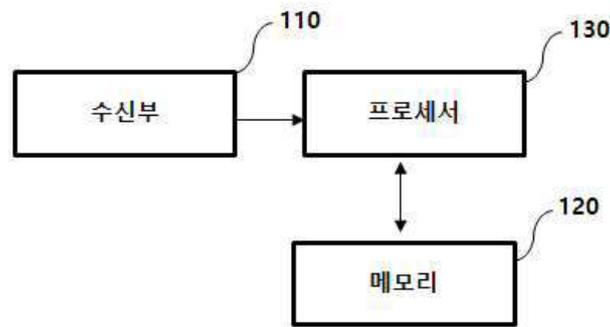


그림 183 - 센서 교체가 가능한 범용 센서노드 및 그 구성 방법 대표 구성도

(라) LCP를 이용하는 스마트온실 시스템 (출원:'18 , 등록:'20)

본 발명은 스마트온실을 관제하기 위한 프로토콜에 관한 것으로, 스마트온실이라 함은 기존 온실 기술에 ICT 기술을 접목하여, 원격 또는 자동으로 작물의 생육 환경을 적정하게 유지하고 관리할 수 있는 농장을 의미한다. 그런데, 현재 우리나라의 스마트온실은 대부분 설치 방법, 재배 작물 등의 종류에 따라 설치하는 장치, 통신 방식 및 소프트웨어가 제조사 별 또는 버전 별로 상이하므로, 개발자는 스마트온실의 구성 방법에 따라 소프트웨어를 수정하여야 하는 문제점을 내재하고 있다. 이러한 문제점을 해결하기 위해 본 발명에서는 작물과 작목 방법에 따라 다른 형태로 구축되는 스마트온실 환경에 범용적으로 적용되며, 관제를 위한 소프트웨어 모듈의 재활용이 가능한 LCP(Lightweight Control Protocol)를 이용하는 스마트온실 시스템을 제공하는 것을 목적으로 한다.

본 발명에 따른 LCP를 이용하는 스마트온실 시스템은 온실 내부에 설치된 센서 및 구동기 디바이스의 노드와 노드의 상위 기기로서, 명령을 노드로 전달하여 주는 제어기와 제어기의 상위 기기로서, 환경 및 작물 생육을 모니터링하고, 온실 환경제어 알고리즘을 운영하는 온실운영시스템을 포함하고, 노드, 제어기 및 온실운영시스템 프로그램의 일부로서 동작하는 응용 계층 프로토콜인 LCP(Lightweight Control Protocol)를 이용하여, 식별 체계와 전송 방식이 다른 하부 전송 기술의 사용을 지원하는 것을 특징으로 한다. 본 발명의 실시 예에 따르면, 작물 및 작목 방법에 따라 다른 형태로 구축되는 스마트온실에 범용적으로 적용되어, 관제 메시지를 교환하는 것이 가능한 LCP를 제공함으로써, 업체마다 다른 제어 방식의 수용이 가능하고, 스마트온실 내 새로운 장비의 개발 또는 도입에도 LCP 자체의 변경 없이 확장하는 것이 가능한 효과가 있다.

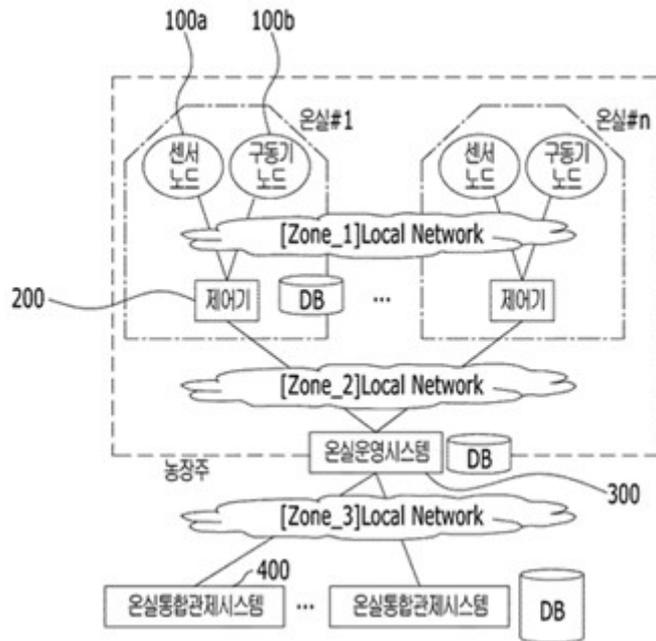


그림 184 - LCP를 이용하는 스마트온실 시스템 대표도

다음 그림은 LCP가 하부 통신기술에 비종속적이라는 특징을 보여주고 있다. 가령 A사, B사 두 회사의 제품은 동일한 센서나 구동기 기능을 가지나, 네트워크 인터페이스가 다를 수도 있다. 본 발명의 실시예에 따른 LCP를 사용할 경우, 네트워크 인터페이스(또는 제조사)에 종속되지 않고 센서/구동기 노드 프로그램과 제어기 응용 프로그램을 구현하는 것이 가능하다. 이를 위해 본 발명에서는 식별자 정의, 노드 및 제어기의 등록(Registration) 메커니즘, 프로토콜 엔터티(노드 및 제어기)의 디스커버리 메커니즘, 연결의 안정성 확보를 위한 하트비트(Heartbeat) 메커니즘, 알림 및 요청에 따른 정보 전송 메커니즘, 노드 및 제어기의 등록 해제(Deregistration) 메커니즘, 기기 초기화를 위한 리셋(RESET) 메커니즘을 포함한다.

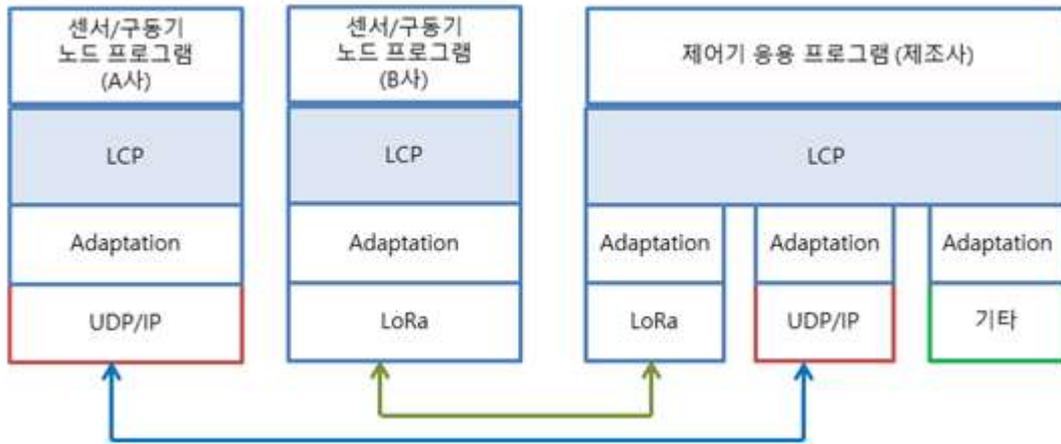


그림 185 - LCP 프로토콜 스택

다음 그림은 LCP의 상태 천이도이다. 모든 통신 개체(노드 및 제어기)는 상위 기기에 성공적인 등록이 이루어진 이후 CLOSED 상태에서 통신이 가능한 CONNECTED 상태로 천이한다. Heartbeat나 응답을 요청하는 메시지를 전송한 이후 응답 타이머가 만료될 경우, 해당 타이머 값에 해당되는 동작을 수행한다. Heartbeat 메커니즘에 의해 기기 간 연결이 단절되었을 경우 CLOSED 상태로 천이하게 된다. 만일 응답을 요청하는 메시지를 수신하였을 경우, 해당 메시지의 페이로드에 포함된 명령을 처리하여 해당되는 응답을 한다(PROCESSING ACTION).

LCP 프로토콜을 이용한 관제 방법은 컴퓨터 시스템에서 구현되거나, 또는 기록매체에 기록될 수 있다. 컴퓨터 시스템은 적어도 하나 이상의 프로세서와, 메모리와, 사용자 입력 장치와, 데이터 통신 버스와, 사용자 출력 장치와, 저장소를 포함할 수 있다. LCP 프로토콜을 이용한 관제 방법은 컴퓨터에서 실행 가능한 방법으로 구현될 수 있는데, 본 발명의 실시예에 따른 LCP 프로토콜을 이용한 관제 방법이 컴퓨터 장치에서 수행될 때, 컴퓨터로 판독 가능한 명령어들이 본 발명에 따른 관제 방법을 수행할 수 있다.

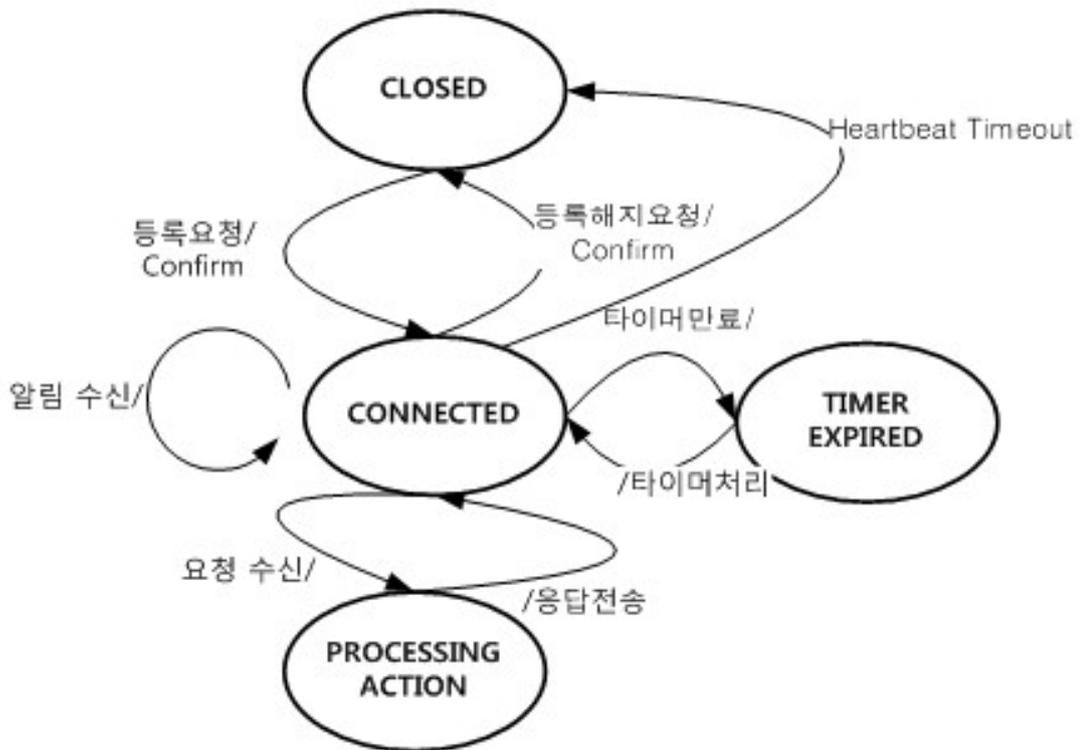


그림 6 - LCP 프로토콜 동작 상태 천이도

(마) 스마트팜 데이터셋 유효성 검증을 위한 수집 데이터 선별 장치 및 그 방법 (출원: '20, 등록: '22)

스마트팜에서 생성되고 수집될 수 있는 데이터는 온실, 축사, 노지 등 스마트팜의 종류에 따라 다양하겠지만, 스마트 온실을 기준으로 (물론 다른 유형의 스마트팜에도 적용이 가능하므로 스마트 온실에 한정되지는 않는다) 크게 시설, 환경, 생육, 경영의 4가지 분야로 분류 될 수 있다.

시설 데이터는 스마트온실과 같이 스마트팜 시설의 기초정보를 위한 데이터로서, 농가 정보, 시설 정보, 담당자 정보 등이 포함된다. 시설 데이터는 스마트팜 관리를 위한 목적으로 주로 사용되므로 한번 생성된 데이터는 잘 변경되지 않는 특성이 있다.

환경 데이터는 시설 내외부의 센서와 구동기 등의 스마트팜 기자재들로부터 생성되는 정보들로서, 정해진 주기에 따라 자동으로 측정되는 특성이 있다.

생육 데이터는 작물의 생장에 따라 측정되어 생성되는 정보로서, 작물의 종류에 따라 매우 다양한 특성이 있다. 예로서, 동일한 스마트온실에서 재배되는 시설원에 작물이라 하더라도 토마토, 파프리카, 딸기 등 작물의 종류에 따라 생육 데이터의 종류는 매우 다양하다. 또한 일부에서는 인공지능 기술을 도입하여 생육 데이터 수집을 자동화 하는 시도가 있지만, 아직까지 생육 데이터는 주로 숙련된 인력에 의한 수작업을 통해 측정이 이루어 지고 있다.

경영 데이터는 스마트팜 경영에 소요되는 비용 및 수익과 관련된 정보로서, 생산 비용, 수확 정보, 판매 정보 등이 포함 될 수 있다. 경영 정보는 농가의 수익성을 드러내기 때문에 통상 잘 공개되지 않는 특성이 있으며, 경매 등의 시장 정보 등을 통해 추정하는 경우가 많다.

스마트팜을 운영하면서 생성되고 관리될 수 있는 데이터는 매우 다양하게 존재 한다. 이들 중 시설 데이터 및 경영 데이터는 통상 데이터 생성의 주기가 불분명하기 때문에 별도의 데이터 수집 장치를 쓰기 보다 수동으로 입력하는 경우가 많으며, 이로 인해 적용되는 데이터 선별 방법 또한 전처리 선별 방법 보다 후처리 선별 방법을 적용하는 것이 보다 바람직하다.

생육 데이터의 경우에도 일부에서 인공지능 기술을 이용한 자동 데이터 수집을 시도하고 있지만 아직까지는 수동으로 확인하여 입력하는 경우가 일반적이므로, 측정하는 과정에서 자연스레 인력에 의해 일차적으로 선별된 데이터가 입력되기 때문에 후처리 선별 방법이 더 적합하다고 할 수 있다.

다만 환경 데이터의 경우에는, 스마트온실의 보급이 보편화 되었고, 이로 인해 온도, 습도, CO2, 일사, 풍향, 풍속, 감우, 광양자, 토양 함수율, 토양 수분 장력, EC, pH, 지온 등을 측정하는 센서와 천창, 측창, 보온 덮개, 차광막, 환풍기, 유동 팬, 관수 모터, 관수 밸브, 냉난방기 등의 구동기를 포함한 다양한 스마트팜 기자재들이 생산 및 보급되어 활용되고 있기 때문에 매우 다양한 스마트팜 데이터들이 지속적으로 생성되고 있다.

일반적으로 빅데이터 활용을 위한 데이터 수집 절차는 다음 그림과 같다. 처음 데이터가 생성되면 데이터 선별 방식에 따라 데이터 선별 처리 여부가 결정된다. 후처리 선별 방식의 경우라면 데이터의 선별없이 다음 단계로 데이터가 전달되게 되고, 전처리 선별 방식의 경우라면 데이터 선별 과정을 거치게 된다.

데이터 선별 과정은 데이터의 유형, 데이터의 활용 목적 등 다양한 조건을 통해 이루어 지게 된다. 데이터의 유형이 적합하지 않거나, 중복된 데이터이거나, 데이터의 수집 또는 활용 목적에 부합되지 않는 데이터는 데이터 폐기 과정을 통해 버려지게 되고, 데이터 수집 목적에 따라 데이터로의 가치가 있는 데이터는 데이터 분석 등을 위해 데이터 전달 과정을 통해 다음 단계로 전달된다.

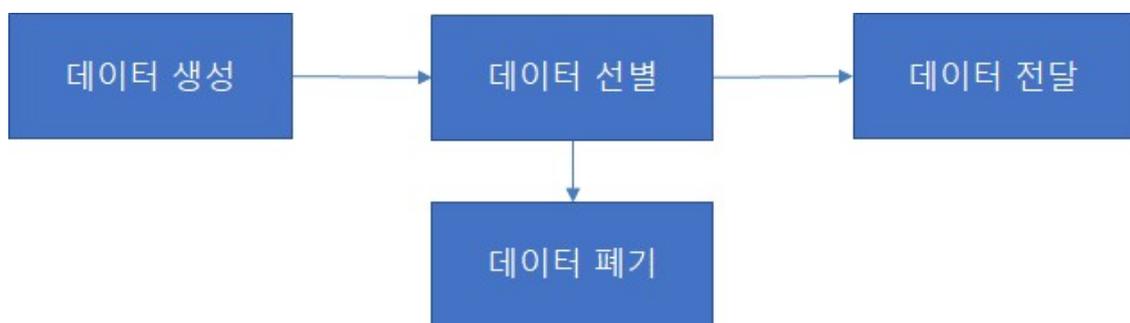


그림 7 -데이터 수집 절차 예시

다음 그림은 스마트팜 데이터 수집 절차에 대한 예시이다. 스마트팜은 온실, 축사, 노지 등 적용 분야에 따라 다양한 분야로 구분될 수 있으며, 여기에서도 다시 생육 대상에 따라 다양한 종류로 구분될 수 있다. 각 분야 및 생육 대상에 따라 기자재들이 달리 적용될 수 있지만, 스마트팜 데이터의 생성은 센서 및 구동기 등을 포함하는 스마트팜 기자재에서 이루어진다. 즉, 스마트팜 기자재가 어떤 유형의 기자재인지에 상

관없이 스마트팜 기자재는 스마트팜 데이터를 생성 할 수 있는 여건을 갖추고 있다고 할 수 있다.

통상 온실, 축사, 노지 등에 설치되는 스마트팜 기자재는 최초 설치시 기자재의 종류에 따라 어떤 데이터를, 어떤 주기로, 어떤 단위로, 언제까지 등의 데이터 생성을 위한 기준을 설정하게 된다. 이후 설정을 마친 스마트팜 기자재는 가동되는 순간부터 주기적으로 데이터를 생성하게 된다.

스마트팜 데이터를 생성에 있어, 센서 및 구동기 모듈은 해당 종류의 환경 및 상태 데이터를 지속적으로 측정하게 된다. 측정된 값은 해당 센서 및 구동기 모듈이 연결된 센서 및 구동기 노드로 보내지게 된다. 통상 하나의 센서 노드 또는 구동기 노드에 다수의 센서 모듈 또는 구동기 모듈이 속하게 되는데, 이는 스마트팜 시설의 구성에 따라 다르다. 센서 및 구동기 모듈이 측정하여 보낸 아날로그 값은 센서 및 구동기 노드가 수치화하여 데이터 선별의 대상이 되는 디지털 데이터로 생성하게 된다.

센서 및 구동기 노드가 생성한 데이터는 전처리 선별 방법에 따라 데이터 선별 과정을 거치게 된다. 데이터 선별 과정은 데이터의 목적에 따라 달라지기 때문에 유효한 데이터를 선별하기 위해서는 무엇보다 데이터의 활용 목적이 명확해야 한다. 즉 필요한 데이터가 무엇인지에 대한 정의가 있어야 하고, 데이터의 유효 범위 또한 명시되어야 한다. 아울러 데이터의 생성 시기 또한 명시되어야 한다. 이는 추후 데이터 분석을 위해 다른 종류의 데이터와의 비교를 위해서는 필수적인 요소라고 할 수 있다. 예로써, 작물의 최적 생육 알고리즘을 도출하기 위한 데이터를 얻기 위해서는 작물의 생육 데이터만 있어서는 안되고, 해당 작물이 자라는 기간 동안 유지 및 관리되는 시설의 환경데이터가 함께 필요하기 때문이다. 통상 최적 생육 알고리즘을 도출하기 위해서는 작물의 생육 데이터와 시설의 환경 데이터는 다시 세분화 되며, 세분화된 각 항목들에 대해 각각의 인과관계가 분석되고 난 후에, 각 항목들이 다르게 적용되는 환경을 인위적으로 구성하고 추후 작물의 생육 결과와 비교하여 최적 알고리즘을 도출하게 된다. 즉 결과적으로 작물의 생육 데이터와 시설의 환경 데이터가 가지는 역학관계를 분석해야 최적 생육 알고리즘이 도출 될 수 있다.

데이터 선별 과정을 통해 센서 및 구동기 노드가 생성한 데이터 중 유효하지 않다고 판단된 데이터는 버려지게 되고, 유효한 데이터로 판단한 데이터들은 데이터 분석 등에 활용되기 위해 연결된 운영 노드로 전달하게 된다. 필요한 경우 해당 데이터의 종류, 생성되는 데이터 및 폐기되는 데이터의 수량 등과 같은 데이터 선별 처리 과정에 대한 이력 정보가 추후 해당 장치의 성능 측정 등을 위해 별도로 보관 될 수 있다.

스마트 온실, 스마트 축사, 스마트 노지 등의 단위로 선별된 데이터들은 농가 내의 통합된 데이터 보관 장소인 운영 노드로 보내진다. 운영 노드는 각 제어 노드에서 수신되는 데이터를 모아 취합하게 된다.

취합된 데이터는 저장되어 농가 내 데이터로서 데이터 분석에 활용 될 수 있고, 인터넷을 통해 클라우드의 관리 시스템으로 전달 될 수도 있다. 관리 시스템으로 전달될 데이터는 관리 시스템이 제공하는 서비스의 목적에 따라 데이터 선별 과정을 통해 관리시스템으로 보낼 데이터인지 아닌지를 판단한다.

관리 시스템 또한 각 농가의 운영노드로부터 수신된 데이터를 취합하고, 취합된 데이터의 선별 과정을 통해 데이터 분석에 유효한 데이터를 추출하여 활용하게 된다.

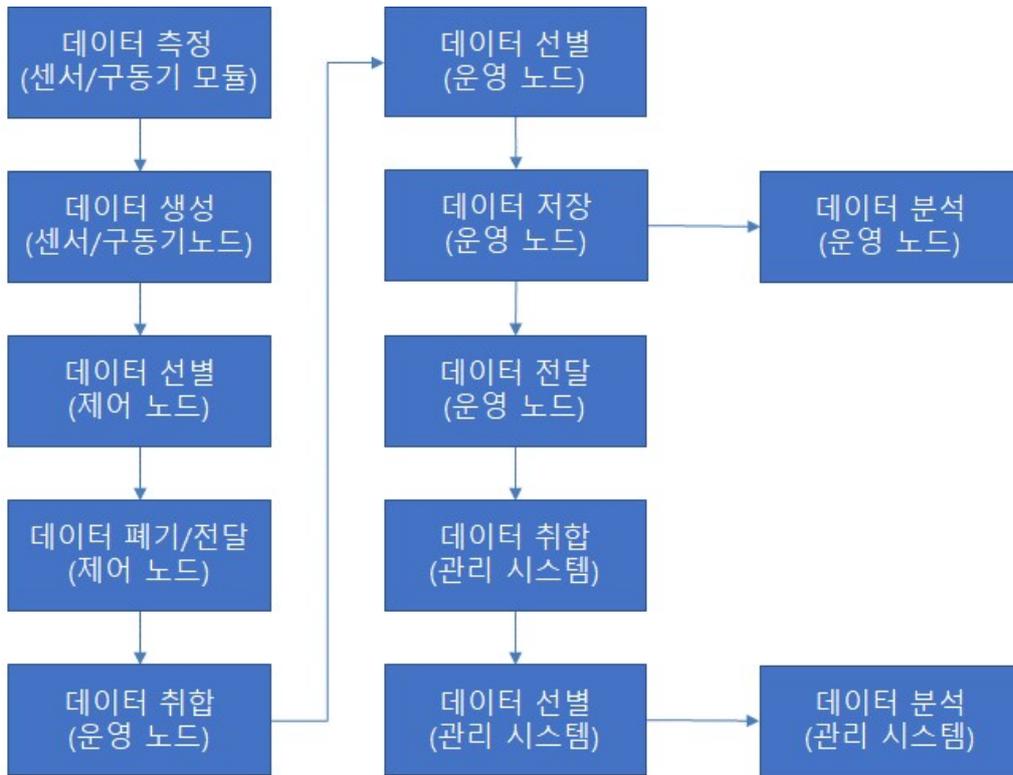


그림 8 - 스마트팜 데이터 수집 절차 예시

다음 그림은 본 발명에 따른 스마트팜 데이터 선별 장치를 위한 기능블록이다. 스마트팜 데이터 선별 장치는 룰셋 기반으로 동작하는 소프트웨어로 구성되며, 데이터 선별 과정을 수행하는 제어 노드, 운영 노드, 관리 시스템 등에 탑재될 수 있다.

스마트팜 데이터 선별 장치는 다음의 기능 블록을 포함한다.

데이터 입력을 위한 입력인터페이스 및 데이터 출력을 위한 출력 인터페이스는 스마트팜 데이터 선별 장치에 입출력 되는 인터페이스를 담당한다.

룰셋 관리는 입력되는 각 데이터에 대해 데이터 선별을 위한 판단의 기준이 되는 조건과 해당 조건에 부합하는 데이터에 대한 처리 규칙을 규정하는 룰셋을 추가하고, 검색하고, 수정하고, 삭제하는 등의 연산을 담당한다. 룰셋은 데이터를 선별하기 위한 규칙들의 집합으로서, 어떤 데이터를 선별할 것인지는 어떤 룰셋을 적용할 것인가에 달려있다. 결국 운영자가 룰셋을 어떻게 정의하는가에 따라 데이터 선별 기준이 달라지게 된다. 데이터를 선별하기 위한 규칙들의 예로서는 다음과 같은 규칙들이 있을 수 있다.

- 데이터가 적정 시간에 수신되었는가? 또는 데이터 선별 장치에 등록된, 기대 주기내에 수신되었는가? 설정된 센싱 주기보다 느리거나 빠르지는 않은가?
- 데이터가 설정된 유효 범위 내에 있는가?
- 데이터가 기준 시간 내에서 측정오차범위를 벗어나는가?
- 다른 연관 데이터와 비교하여 유효한가? 이중화된 센서의 측정값이 차이가 나는 경우가 포함된다.

사용자 인터페이스는 룰셋에 대한 연산을 하기 위한 인터페이스를 담당한다.

데이터 캐싱은 입력되는 데이터에 대한 임시 버퍼 역할을 담당한다. 이를 통해 입력되는 데이터 상태에 대한 연산이 가능해진다. 예를 들어, 온도센서#1의 직전 10개 데이터의 평균값을 룰셋에 적용할 수 있다.

룰셋 적용은 입력되는 데이터에 대해 정의된 룰셋을 적용함으로써 유효한 데이터를 선별하는 프로세스를 담당한다.

이력 관리는 입력 데이터 및 선별된 데이터에 대한 이력 보관을 담당한다. 예를 들어 특정 센서 모듈의 최근 1일 동안 입력 데이터 및 폐기된 데이터의 개수 등을 보관할 수 있다. 데이터 폐기율이 이상적으로 높거나 낮을 경우 운영자는 해당 장치의 상태를 점검 해 볼 수 있다.

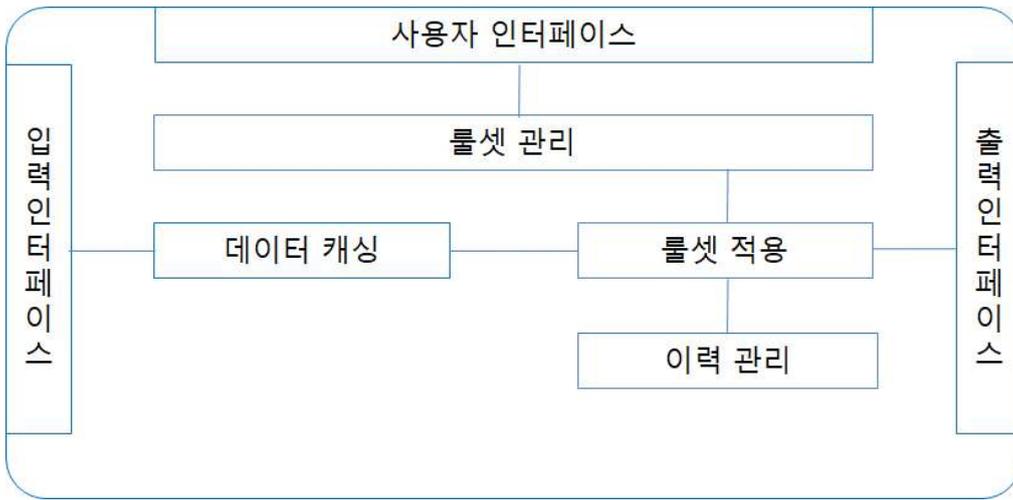


그림 9 - 스마트팜 데이터 선별 기능 블록

가-2) [국내특허출원] 국내특허 출원

당해 연도에서는 스마트팜에서 생성되는 환경 및 생육에 대한 빅데이터 처리 기법, 최적 생육환경 알고리즘과 수반되는 데이터를 신뢰적으로 관리할 수 있는 방법 등 다음과 같이 3건 국내 특허들이 출원되었다. (2019.11.28.)

특허명	출원일	특허등록결정일
IoT 디바이스 인터페이스 시스템 및 그 방법	2018-11-28	
스마트팜 게이트웨이	2020-02-06	
스마트팜 관리 방법 및 장치	2020-02-25	
스마트팜 환경 및 생육 데이터 저장 시스템	2020-05-11	
타임라인 기반의 전염병 추적조사 시스템 및 그 방법	2020-07-29	
스마트팜 기자재 공간 위치 인식 장치 및 그 방법	2020-08-31	
스마트 팜에서의 비료 사용량 처방 시스템	2021-01-29	
스마트팜 데이터 전송 장치 및 그 방법	2021-10-21	
영상장치를 이용한 식물 생육 측정 시스템 및 그 방법	2021-12-07	
컨테이너를 활용한 조사료 재배 장치 및 그 시스템	2022-10-14	
야생동물 식별을 통해 학습효과 방지가 가능한 야생동물 퇴치 시스템 및 그 방법	2022-10-14	

(가) [국내특허출원] IoT 디바이스 인터페이스 시스템 및 그 방법 (출원: '18)

종래 기술에 따르면, 노드의 센서/구동기 디바이스가 변경될 때마다, 제어기 운영자는 디바이스-레지스터 주소 맵을 숙지하여, 제어 응용프로그램을 변경하거나 내부적으로 유지하고 있는 변환 테이블을 수정하여야 하므로, 적응성이 떨어지는 문제점이 있다.

또한, 종래 기술에 따르면 제어기와 노드 사이에 교환되는 메시지는 노드로부터 제공되는 기능이 다양해짐에 따라, 모든 노드의 능력을 모두 포함하는 구조를 제공하여야 하는데, 이에 따라 대역폭이 낭비되고 속도가 저하되는 문제점이 있다.

따라서 본 발명에서는 IoT 디바이스 속성 정보에 대한 자동 구성(Auto Configuration)을 수행하여 범용적인 동작을 지원하고, 데이터 획득 및 기록 과정 중에 해당 노드의 디바이스에 맞는 최적화된 데이터만을 상호 전송하도록 함으로써, 대역폭 절약과 빠른 전송 속도 구현이 가능한 IoT 디바이스 인터페이스 시스템 및 그 방법을 제공하는데 그 목적이 있다.

본 발명에 따른 IoT 디바이스 인터페이스 시스템은 IoT 디바이스에 대한 데이터 읽기 또는 쓰기 요청을 수신하는 수신부와, IoT 디바이스에 대한 속성 정보를 확보하고, 데이터 읽기 또는 쓰기 요청 시 데이터를 송수신하는 프로그램이 저장된 메모리 및 프로그램을 실행시키는 프로세서를 포함하고, 프로세서는 노드 내 IoT 디바이스의 환경설정 정보 및 환경설정 정보의 길이 정보를 이용하여 IoT 디바이스에 대한 속성 정보를 자동 구성하고, IoT 디바이스의 능력 정보를 포함하는 메시지 구조를 이용하여 데이터 획득 및 기록 과정을 수행하는 것을 특징으로 한다.

본 발명에 따른 IoT 디바이스 인터페이스 방법은 노드 내 IoT 디바이스의 속성 정보를 자동 구성하는 단계와, 자동 구성된 속성 정보를 이용하고, 노드 내 IoT 디바이스의 능력 정보를 포함하는 교환 메시지 구조를 이용하여, 데이터 획득 및 기록 과정을 수행하는 단계를 포함하는 것을 특징으로 한다.

본 발명의 실시 예에 따르면, 온실통합제어기는 초기에 또는 일정 주기로 디바이스-레지스터 맵 정보와 맵 정보의 길이 등의 환경설정 정보를 가져오고, 변경된 내용을 반영하여 자동으로 확보하는 IoT 디바이스 속성 정보 자동 구성을 수행함으로써, 센서/구동기노드 제조사별 구현 방법이나 노드에 연결된 센서/구동기 디바이스와 상관없이 범용적으로 동작하는 것이 가능한 효과가 있다.

본 발명의 실시 예에 따르면, IoT 디바이스 속성 정보 자동 구성(Auto Configuration) 과정 중에 노드 내 디바이스에 대한 레지스터 주소, 데이터 길이, 데이터 속성 정보를 확보하고, 데이터 획득과 데이터 기록 과정 중에 해당 노드의 디바이스에 맞는 최적화된 데이터만 상호 전송함으로써, 대역폭 절약과 전송 속도 향상 등 최적화된 통신 수행이 가능한 효과가 있다.

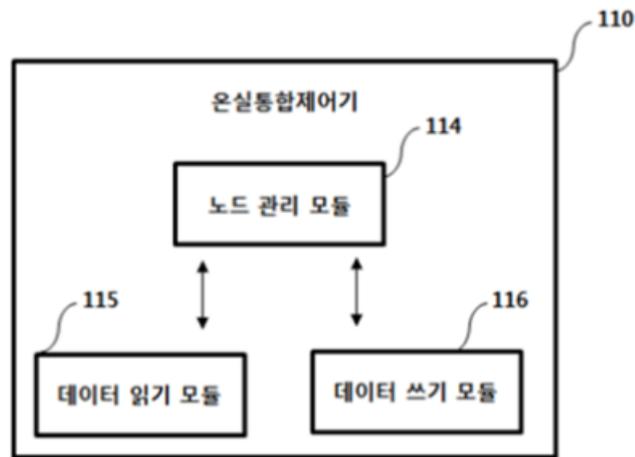


그림 1 - 온실통합제어기의 구성을 나타내는 블록도

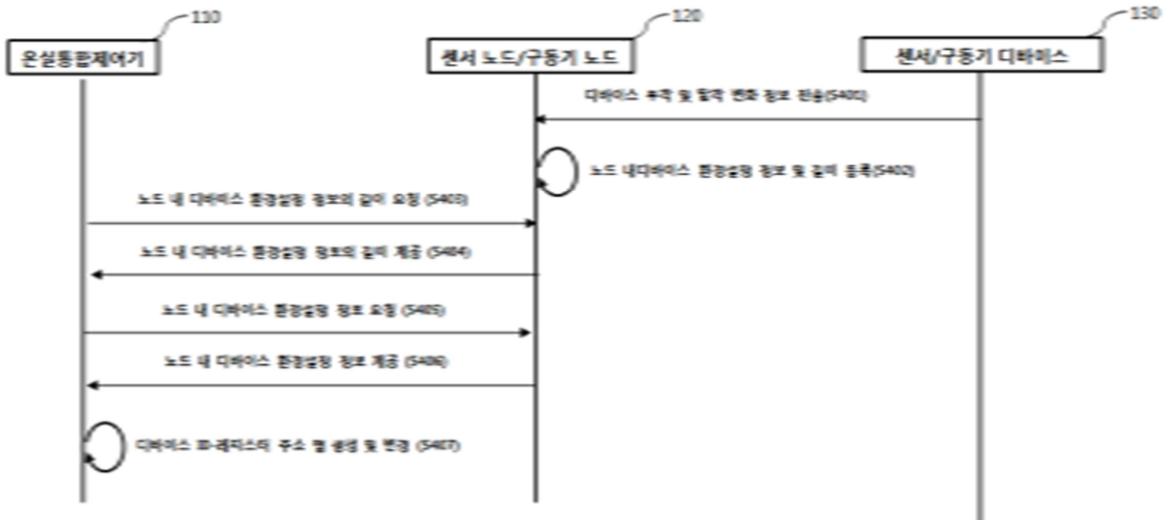


그림 2 - 환경설정 정보 관리 과정을 도시하는 시퀀스 다이어그램

(나) [국내특허출원] 스마트팜 게이트웨이 (출원: '20)

본 발명은 스마트팜에 활용되는 각종 기자재들간 상호 호환성을 제공함으로써, 농장의 환경에 맞는 구동기 및 센서 등의 기자재를 자유롭게 선택하여 활용하게 하기 위함이다. 스마트팜은 기존의 농업에 IoT 등 ICT기술을 융합하여 생산성과 효율성을 제고함으로써 고부가가치 창출을 추구하는 첨단 농업 기술로서, 다양한 IT기기들이 활용된다.

스마트팜은 통상 생육 및 환경 정보를 모니터링 하기 위한 센서 장치, 스마트팜 시설을 제어하는 구동기, 각 시설에 부착된 장치들을 제어하는 제어기, 그리고 이들을 통합적으로 관리하는 게이트웨이 등으로 구성된다. 게이트웨이의 경우 활용 규모나 방식에 따라 관리시스템, 관리PC, 관리서버 등 다양한 형태로 구비될 수 있다. 최근의 스마트팜은 클라우드, 인공지능, 빅데이터 기술 등이 접목되어 원격에서 데이터를 보관하고 이를 분석하여 보다 정확한 의사결정을 지원하도록 진화하고 있다.

그림 3은 본 발명에 따른 스마트팜의 구성도이다.

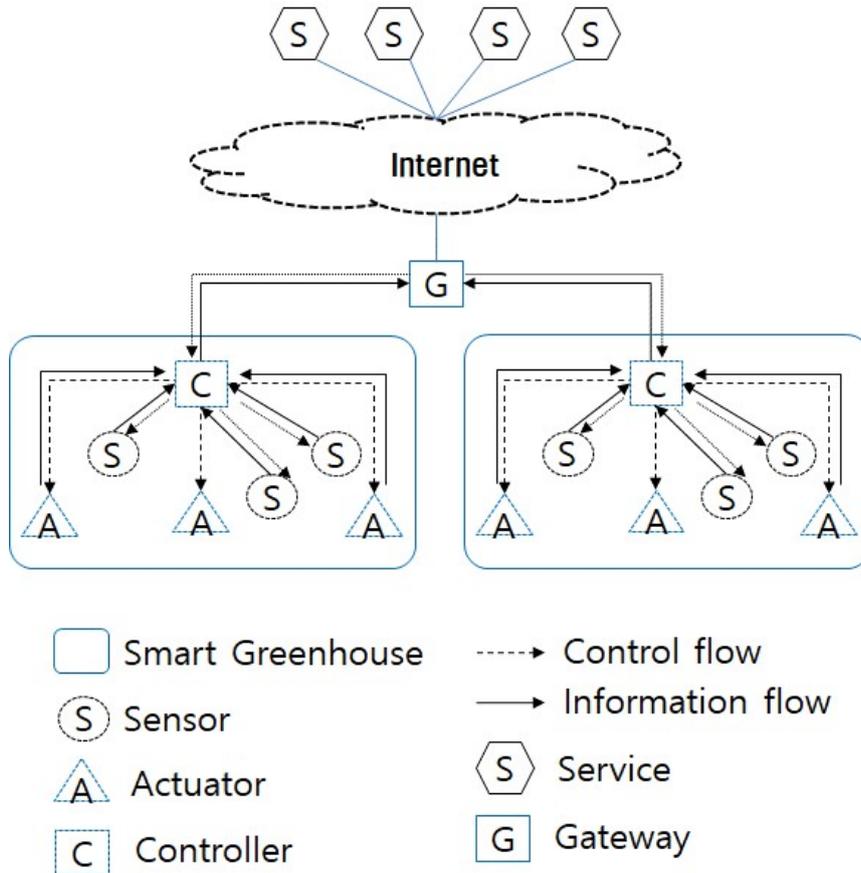


그림 3 - 스마트팜 구성도

스마트팜에는 스마트온실, 스마트축사, 스마트과수원 등 영농의 종류 및 방식에 따라 다양한 형태의 시설 및 장치들이 구비될 수 있는데, 그 중 스마트온실의 경우를 예로 설명한다. 스마트 온실에는 전술한 바와 같이 센서 및 구동기 장치들과 환경 데이터를 수신하고 제어 명령을 전달하는 제어기가 구비된다. 센서의 종류는 온도, 습도, 조도, CO<sub>2</sub>, 강우, 일사, 풍향, 풍속 등 다양한 센서들이 있으며, 각 센서들은 정해진 주기 또는 요청에 따라 현재의 측정값을 전달한다. 구동기는 환기, 난방, 차광, 보온, 관수, 관비 등을 위한 다양한 장치들이 있으며 제어 신호에 따라 정해진 동작을 수행한다. 제어기는 구동기 장치를 제어하는 신호를 전달하고, 또한 센서가 전송하는 데이터를 게이트웨이로 전달한다.

각 스마트온실의 제어기는 스마트팜을 관리하는 스마트팜 게이트웨이에 연결되며, 스마트팜 게이트웨이는 농장내 각 시설들을 통합 관리할 수 있는 인터페이스를 제공한다. 또한 클라우드와 연결되어 원격지의 스마트팜 서비스와 연결되는 인터페이스를 제공할 수 있다. 스마트팜 서비스의 예로는, 스마트온실에서 생성하는 생육 및 환경 정보를 분석하여 최적생육모델을 도출하는 스마트팜 데이터 분석 서비스, 스마트 온실내 작물의 생육상태를 영상으로 전송 받아 원격에서 분석하여 작물의 생육 상태나 병해충 식별을 해주는 원격 진단 서비스 등이 될 수 있다. 물론 네트워크를 통해 외부의 서비스를 이용하지 않고, 농가 내에 자가 시설을 갖추고 스마트팜 서비스를 자체적으로 구성할 수도 있다.

한편, 스마트팜 시설을 구비하고 스마트팜 서비스를 활용하기 위해서는 각 장치 간 호환성이 담보되어야

하는데, 통상 기존 구축된 시설에 새로운 장치를 구비할 때 생산 업체간 규격이 다르면 장치가 호환이 되지 않는다. 이 경우 장치 상호간 호환성을 제공하기 위해 컨버터를 사용한다. 다음 그림은 컨버터를 이용하여 장치간 상호 호환성을 제공하는 예시이다. 그림에서와 같이 컨버터를 사용하여 상호 호환성을 제공하는 경우 농장에서 활용하는 장치가 증가할수록 필요한 컨버터도 비례하여 증가하게 되고, 장치 생산업체 또한 다수의 컨버터에 대한 부담이 있다.

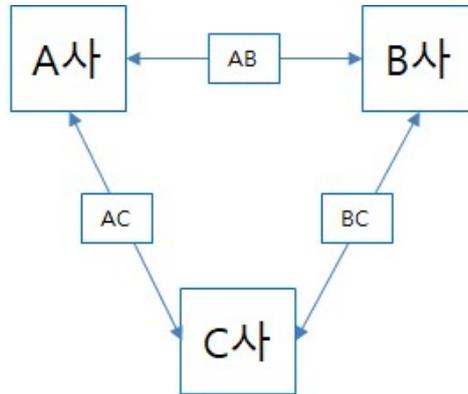


그림 4 - 컨버터를 이용한 상호호환성 제공 예시

다음 그림은 본 발명에 따른, 스마트팜 게이트웨이를 이용한 상호 호환성 제공에 대한 예시이다. 스마트팜 게이트웨이는 게이트웨이 인터페이스 계층에 표준 규격 인터페이스를 탑재하게 되고, 각 생산업체의 장치는 스마트팜 게이트웨이의 표준 규격 인터페이스를 준용하여 장치를 생산한다. 스마트팜 게이트웨이와 스마트팜 장치들간의 상호 데이터 및 신호 교환은 게이트웨이 인터페이스를 통해 이루어진다. 게이트웨이 인터페이스에 표준 규격 인터페이스는 스마트팜 장치의 종류 및 기능에 따라 다양하게 존재할 수 있는데, 장치의 기능 범위를 나타내는 리소스 모델링 관련으로는 RAML, YAML, JSON, RESTapi 등이 될 수 있고, 데이터 메시지 전달을 위한 규격으로는 CoAP, MQTT 등이 될 수 있다.

이는 전술한 컨버터를 사용하는 방법보다 진보한 방법으로서, 농장의 경우에는 표준 규격을 준용하는 스마트팜 게이트웨이만 구비하면 다양한 생산 업체의 장치를 활용할 수 있고, 각 생산업체는 다양한 타사의 규격을 일일이 맞추지 않고 스마트팜 게이트웨이의 표준 규격에 대한 인터페이스만 제공하면 된다.

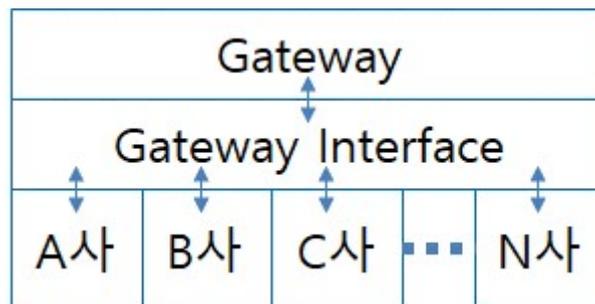


그림 5 - 스마트팜 게이트웨이를 이용한 상호호환성 제공 예시

(다) [국내특허출원] 스마트팜 관리 방법 및 장치 (출원: '20)

본 발명은 전술한 종래기술의 문제점을 해결하기 위하여, 스마트팜의 센서에서 수집되는 측정값과 스마트팜의 구동기를 제어하기 위한 제어 명령어에 대한 분산 원장을 구성함으로써 데이터의 무결성을 보장하고 확인할 수 있는 방법을 제공하고자 한다.

스마트팜은 창 개폐등 관리 동작을 위한 하나 이상의 구동기와 온도나 습도 등의 정보를 얻기 위한 하나 이상의 센서 그리고 상기 구동기와 센서를 제어하기 위한 제어를 포함한다. 또한 스마트팜 관리 서비스를 제공하는 사업자는 스마트팜의 효율적인 관리를 위해 클라우드 기반 스마트팜 관리 서비스를 제공할 수 있다. 이 경우, 하나 이상의 스마트팜이 공용망인 인터넷 또는 개별로 구성된 사설망을 통해서 클라우드 서버와 연결될 수 있다. 스마트팜의 제어기는 클라우드 서버와 스마트팜에 설치된 센서와 구동기를 연결해주는 허브 역할을 수행한다. 이때, 제어기가 센서로부터의 측정값을 클라우드 서버로 전송하거나 클라우드 서버가 제어기로 구동기를 조작하기 위한 제어 명령을 전송하는 각각의 데이터 전송을 본 특허에서는 거래(transaction)이라고 명명한다.

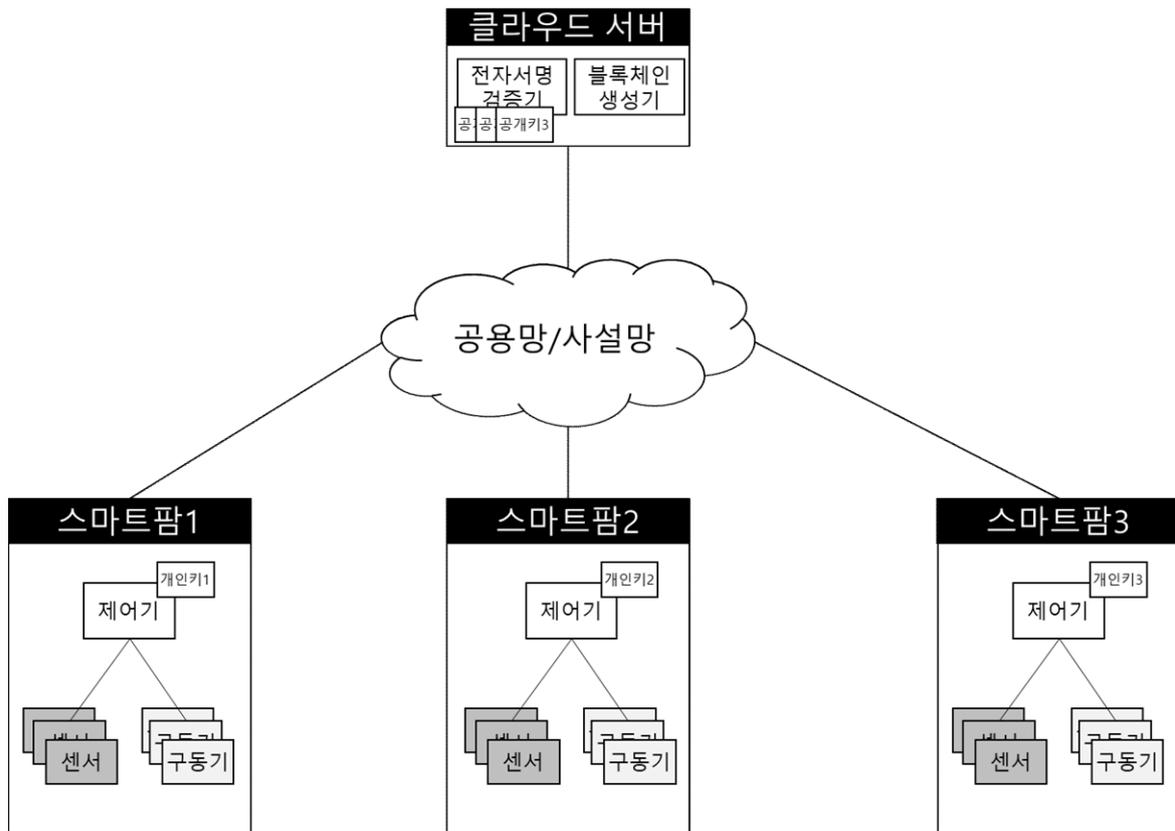


그림 6 - 클라우드 기반 스마트팜 관리 시스템 개요도 클라우드 기반 스마트팜 관리 시스템 개요도

본 발명에 따른 제어기는 크게 두가지의 동작을 수행한다.

첫 번째는 센서로부터 수신한 측정값에 대한 처리이다. 도2와 같이 센서가 제어기로 측정값을 전달하면, 제어기는 상기 측정값을 자신의 개인키로 전자서명하고 전자서명된 측정값을 클라우드 서버로 전송한다. 이후 클라우드 서버는 전자서명 검증기에 저장된 상기 제어기의 공개키를 이용하여 수신한 측정값이 상기 제어기가 보낸 측정값이 맞는지 확인한다. 상기 과정을 위한 제어기의 공개키 배포는 다양한 방법으로 이루어질 수 있으며, 본 특허에서는 다루지 않는다. 또한, 개인키를 이용한 전자서명과 공개키를 이용한 전자서명의 검증 방법은 본 특허의 범위가 아니며 일반적으로 사용하고 있는 방법이므로 본 특허에서 다루지 않는다. 이후 클라우드 서버는 수신한 측정값에 대한 무결성을 보장하기 위해서 측정값에 대한 블록체인을 생성한다. 이때 블록체인의 생성 방법이나, 블록체인을 구성하는 블록의 형태는 본 발명에서 다루지 않지만 SHA-256 등 해시 함수를 사용하는 어떠한 블록체인 기법을 사용해도 무방하다. 또한 상기 블록은 상기 클라우드 서버가 생성하며 상기 블록체인이 서비스 사업자 내부에서 관리하는 사적(private) 블록체인이기에 블록의 생성이 높은 난이도를 가지지 않을 것은 자명하다. 다만, 측정값에 대한 블록체인을 구성하는 블록은 제어기로부터 수신한 측정값을 포함해야 한다. 추가적으로 상기 블록은 측정값을 수신한 시간과 측정값을 송신한 제어기의 ID를 포함할 수 있다. 일반적인 블록체인 기술에 따르면, 클라우드 서버가 구성하는 측정값에 대한 블록체인을 구성하는 블록들은 링크드 리스트 형태로서

연결되어 있기 때문에 특정 측정값을 조작하는 것이 불가능하다. 따라서 측정값의 블록체인화는 보고된 측정값의 의도적 또는 비의도적 수정을 방지할 수 있다.

(라) [국내특허출원] 스마트팜 환경 및 생육 데이터 저장 시스템 (출원: '20)

스마트팜 빅데이터를 구축함에 있어서 필요로 하는 데이터는 크게 온실 내부의 습도, 온도, PH 등 작물을 재배하는데 필요로 하는 환경데이터와 작물의 크기, 작기 등을 기술하는 생육데이터로 나뉘게 된다.

본 발명에서는 그림 7에서와 같이 스마트팜에서 재배하는 작물에 대한 생육 데이터와 재배환경에 관한 비가공데이터 (Raw 데이터)를 압축한 후, 압축된 데이터를 저장하고, 저장된 압축 데이터를 복원하여 제공할 수 있는 방법과 그 시스템을 다룬다.

스마트팜에서 발생하는 데이터는 크게 작물을 키우기 위한 환경 데이터, 작물의 성장상태 및 작물에 비료등을 공급해 주는 환경 및 생육 데이터로 구분된다. 일반적으로 스마트팜 환경 데이터는 스마트팜에 설치된 환경센서를 통해 주기적(1분, 혹은 10분 단위로) 수집된다. 이와 반해 작물의 생육 데이터는 별도의 센서가 설치되지 않는 한 관리자가 수기 혹은 별도의 방법으로 측정하곤 한다.

이들 환경 및 생육 데이터는 과거의 성공 혹은 실패 사례를 통해 농민들이 보다 나은 영농을 할 수 있는 필수 데이터로 활용이 된다. 이들 데이터들은 최근 빅데이터로 구성되어 과학적인 영농을 가능하게 한다. 환경 센서로부터 수집되는 데이터는 비교적 매우 잦은 주기로 수집된다. 이렇듯 잦은 빈도로 환경 정보를 수집하는 이유는, 혹시라도 스마트팜 내부에 환경 변화가 발생하였을 때,스마트팜 내부에 설치된 자동화 기기를 동작시켜 이를 해결하기 위함이다. 또한 이러한 데이터들은 작물을 키움에 있어 매우 중요한 데이터로 활용이 되기 때문에, 기록으로 관리해야 할 필요가 있어 빅데이터로 구축된다.

하지만 이러한 환경 데이터들은 실시간으로 감지됨에 따라 많은 경우에 있어서 중복되는 데이터로 발생을 하게 되기 때문에, 보다 효과적인 빅데이터로 구축 및 관리 방법이 필요하다. 즉, 다양한 환경센서로부터 (혹은 생육 관련 센서로부터) 발생하는 비가공데이터 (Raw 데이터)를 가공 데이터 (압축데이터) 형태로 빅데이터를 구축하고 관리하는 것이 매우 중요하다. 따라서 본 발명에서는 스마트팜의 환경과 식물의 생육 데이터를 “스마트팜 데이터”로 지칭하여 기술하도록 한다.

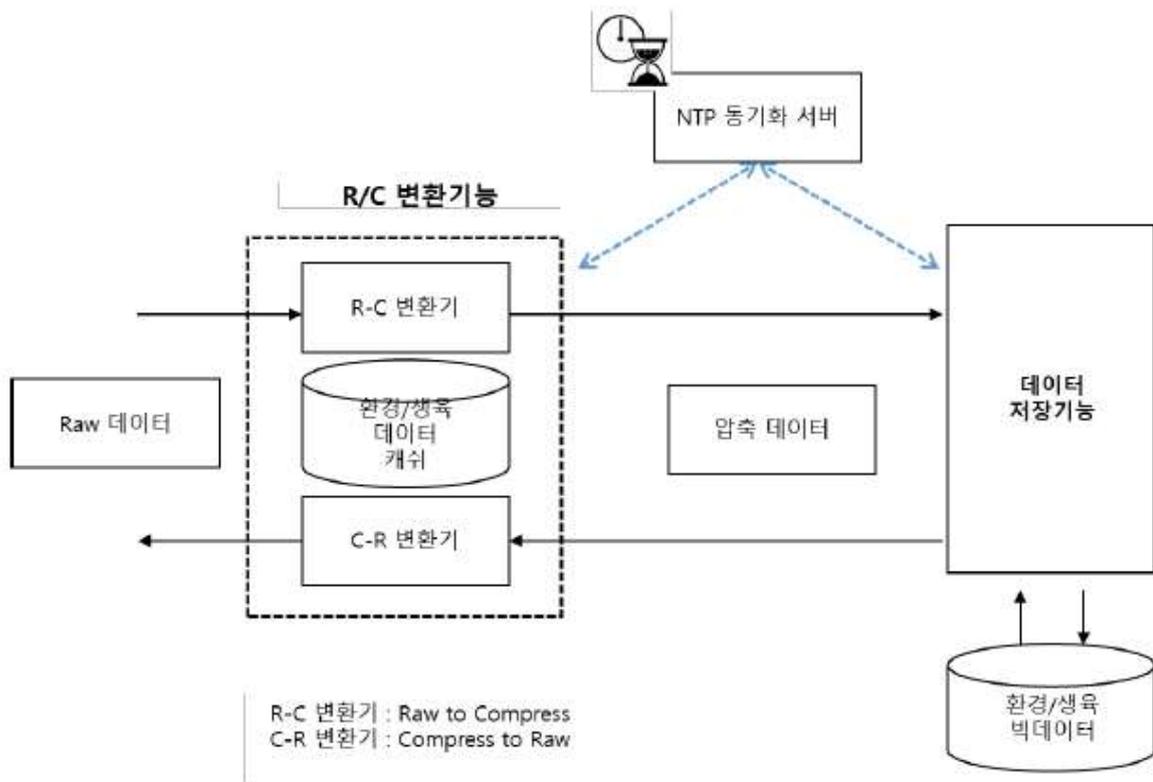


그림 7 - 스마트팜 생육/환경 데이터 압축하는 방법 (대표도)

(마) [국내특허출원] 타임라인 기반의 전염병 추적조사 시스템 및 그 방법 (출원:'20)

본 사업에서는 구제역, 조류독감 등과 같은 감염병(혹은 전염병)에 있어 사람이 주요한 매개체로 동작할 때, 스마트폰에 저장된 사람의 위치 경로와 시간 정보를 토대로 선제적으로 감염 위험 여부를 파악하여 전파할 수 있는 방법 및 시스템에 관한 특허를 출원하였다.

본 발명과 관련한 종래의 기술은 가축 전염병 발생 시 가축 전염병의 전파 경로를 추적하고 그 전파 경로 상에 있는 축산인, 수의사 등 관련 종사자, 가축 이동 및 사료차량, 사료 및 배설물 등을 신속히 소독 및 방역을 함으로써 더 이상의 전파를 방지하고 피해를 줄이고자 하는 것에 관한 것이다. 기존 기술을 좀 더 구체적으로 살펴보면, 축산 농가의 축산인, 축산차량, 방문자, 방문차량, 공무원 등이 소유한 스마트폰과 같은 사용자 단말기 정보와 축산 농가, 축산 관련기관, 관련회사, 시설 등을 메인 서버에 등록하고, 상기 등록받은 축산인, 차량, 방문자, 방문차량 등이 상기 축산 농가, 축산 관련기관, 관련회사, 시설 등을 이동할 때 상기 인원, 차량이 소유한 사용자 단말기의 이동 경로 상의 위치 정보 및 이동 시간 정보를 수집하고, 상기 메인 서버에 구축된 DB를 이용하여 축산인과 차량, 방문자, 방문차량 등에 대한 이동 경로를 시간대별로 파악함으로써 가축 전염병 발생 시 방역 인원 및 지역을 신속히 파악하여 소독 및 방역을 효과적으로 함으로써 전염병 확산을 방지하고 피해를 최소한으로 줄이도록 구성된 것이다. 하지만, 종래의 기술은 중앙 집중 서버(메인 서버)를 통해 등록한 대상으로부터 수집한 이동 경로와 이동 시간 정보를 토대로 전염병에 대한 판단을 하기 때문에, 중앙 집중 서버의 부하가 커질 수 있으며, 가령 축산인을 방문한 지인으로부터 전염되는 경우에서와 같이, 비축산인을 매개체로 하는 전염 경로를 파악하기 어렵다는 문제점이 있다.

이러한 문제점을 해결하기 위해, 본 발명에서는 전염병 감염체의 이동 경로와 시간 정보를 분산 데이터(P2P, 빅데이터, 클라우드 등)에 구축한 후 주기적인 전염병에 대한 알림을 제공하여, 전염 위험성에 대한 판단 여부를 중앙 집중 서버가 아닌 개인의 스마트폰으로 판단하고 이상 징후 발생시 개인이 자신의 이동 경로와 시간 정보의 등록이 가능하기 때문에, 감염 경로에 대한 파악을 실시간으로 제공할 수 있는 방법을 제시한다. 특히 본 발명에서는 어느 특정 전용 서버를 통해 감염여부를 판단하는 것이 아니기 때문에, 기존의 발명에서와 같이 축산 감염 전용 서버를 통한 축산 전염병 감시만을 제공하는 것이 아니라, COVID-19와 같이 사람을 대상으로 하는 전염병 추적 시스템으로도 활용이 가능한 기술이다.

다음 그림은 본 발명에서 기술하고자 하는 하고자 하는 타임라인 기반의 전염병 추적 조사 시스템의 개요이다.

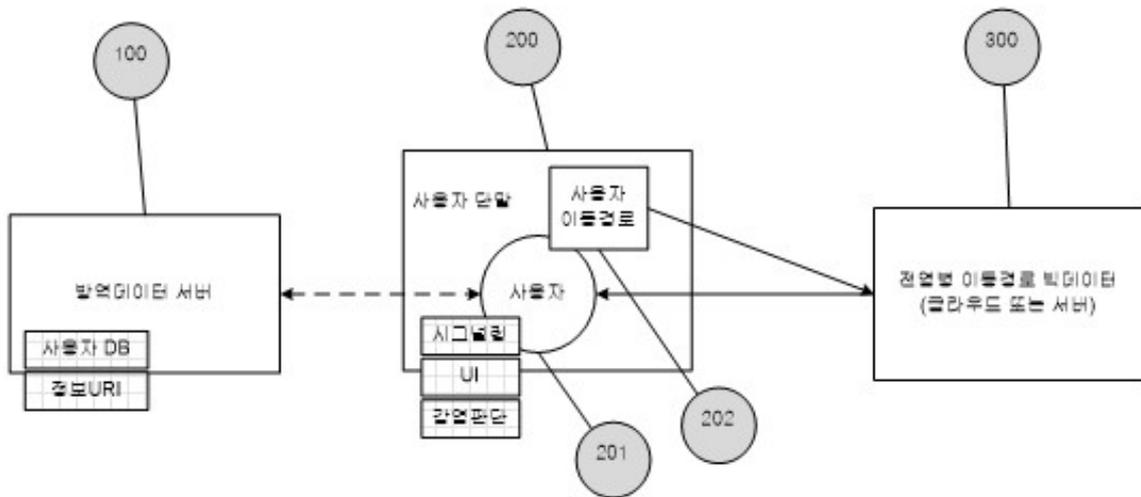


그림 8 - 타임라인 기반의 전염병 추적조사 방법 및 시스템

본 발명에서 고려하는 시스템은 방역데이터 서버(100), 사용자 단말(200), 및 전염병 이동경로 빅데이터 서버(300)로 구성된다. 이들 시스템들간에는 다양한 유무선 통신 망을 이용하여 상호 통신이 가능하다.

방역데이터 서버(100)은 사용자 DB와 정보 URI를 관리한다. 사용자 DB에는 감염체 정보를 수집하라는 알림을 보내기 위한 대상들을 저장한다. 즉, 사용자 DB는 사용자 단말(200)에게 메시지 송신이 가능한 주소 DB이다. 방역데이터 서버(100)는 단말(200)들이 제공하는 전염병 이동경로 정보를 전염병 이동경로 빅데이터 서버(300)에 저장한 후, 주기적으로 혹은 신규 감염체 발생 이벤트가 발생할 경우, 등록되어 있는 모든 사용자 단말(200)에게 감염체 정보를 수집하라는 알림을 보내는 역할을 수행한다. 이때 사용자 단말(200)들이 전염병 이동경로 데이터로의 접근을 안내하기 위한 주소를 정보 URI에 보관한다.

사용자 단말(200)에는 사용자 응용(201)과 사용자 이동경로 정보(202)가 탑재된다. 사용자 응용(201)은 방역데이터서버(100)과 전염병 이동경로 빅데이터 서버(300)과 제어 정보를 교환하기 위한 시그널링 기능과 단말을 사용하는 사람에게 정보를 표출하기 위한 UI, 감염 위험성을 판단하기 위한 계산 모듈로 구성

된다. 사용자 이동경로 정보(202)는 사용자 단말의 위치 정보와 시간 정보를 저장한다.

전염병 이동경로 빅데이터(300)은 실시 방법에 따라 다양한 형태(가령, 전용 DB서버, 클라우드, 혹은 P2P 등)로 구현 될 수 있다. 전염병 이동경로 빅데이터(300)은 사용자 단말(200)에게 축적되어 있는 전염병 이동경로를 제공하거나, 방역데이터 서버(100)의 요청에 따라 사용자 단말(200)의 사용자 이동경로(202)를 제공받아 전염병 이동경로에 축적하는 역할을 한다. 이 때 전염병 이동경로를 축적하는 기간은, 전염병의 잠복기간의 특성에 따라 정한다. 전염병 이동경로 빅데이터의 크기를 한정하기 위해 전염병의 잠복기간에 따라 조정한다. (실시예에서, 전염병 잠복기간의 2배로 한다)

다음 그림은 본 발명의 동작에 대한 이해를 높이기 위하여 감염체에 따른 전염 경로 예시를 보였다. 다음 그림에서 9명의 사용자 (A~I)는 축산 차량에 설치된 네비게이션, 사용자가 휴대하고 있는 스마트폰, 스마트패드 등 위치 경로와 시간 정보를 수집할 수 있는 모든 장치를 의미한다. 다음 그림의 왼쪽 그림에서는 사용자들을 2차원 평면으로 나열하였을 때, 전염 경로에 대한 표현이며, 다음 그림의 오른쪽 그림은 사용자들간의 전염 인과 관계를 트리형태로 나타낸 그림이다.

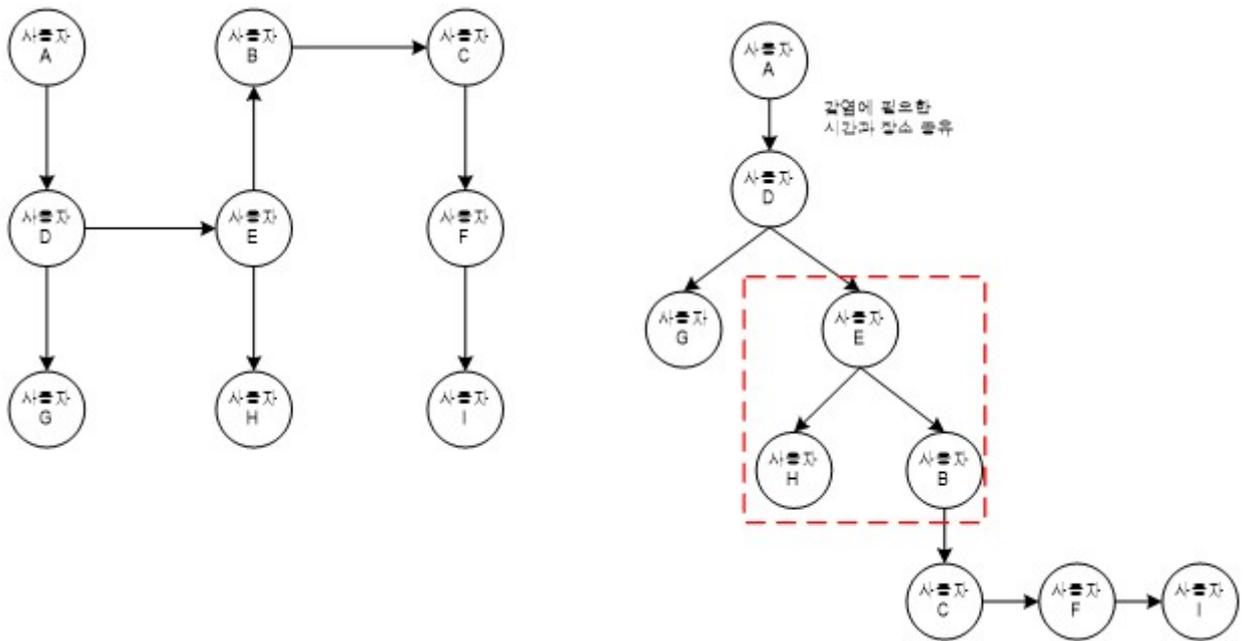


그림 9 - 본 발명에서 고려하는 전염 경로 예시

위 그림의 오른쪽 점선으로 표시한 사용자 E, 사용자 H, 사용자 B의 감염 원인을 아래 그림에서와 같은 시나리오로 구성하였다. 사용자 E는 사용자 H와 사용자 B에게 Loc 3 지점에서 감염을 시킨 경우이다. 이 때, Loc1에서 사용자 E와 사용자 B의 동선이 겹쳤지만, 감염가능시간을 벗어났기 때문에, Loc1에서 사용자 E는 사용자 B에게 감염시키질 못하였다.

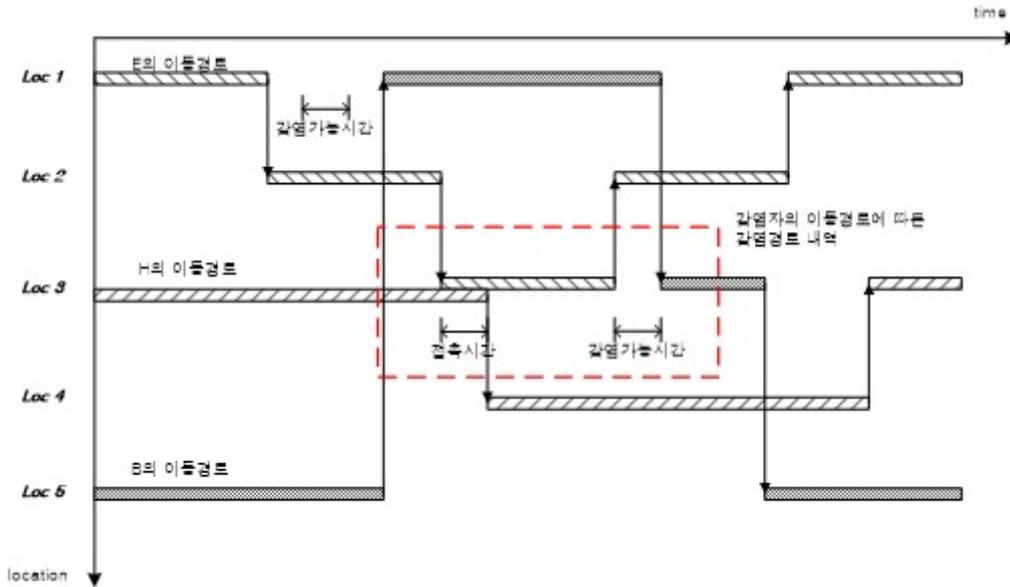


그림 10 - 본 발명에서 고려하는 전염 경로에 대한 해석 예시

이때 전염경로에 대한 해석은 전염병의 잠복기간에 따라 조절이 가능하다. 실시 예에 따라 잠복기간이 1주일 경우, 전염 경로에 대한 판단은 잠복기간(1주일)을 포함할 수 있도록 충분한 기간을 고려한다. 이러한 시나리오를 감안하였을 때, 사용자 E의 경로가 사용자 H와 사용자 B에게 실시간으로 통보될 수 있다면 전염 방지에 큰 도움이 될 수 있다.

본 발명은 기존 가축 방역시스템에서 검역 하고 있는 항목들에 추가하여, 축산인은 물론 이동경로를 추적할 수 있는 어떠한 단말을 소지한 비축산인들 또한 실시간으로 방역 시스템에 참여를 할 수 있다. 따라서 더 많은 데이터 축적을 통해 방역 시스템의 완성도를 높일 수 있을 뿐만 아니라, 감염체에 대한 실시간 이동경로 제공을 통해 더 빠른 대비를 할 수 있어, 가축전염병으로 인한 손실을 크게 줄 일수 있다는 이점이 있다. 또한 본 발명은 가축 방역뿐만 아니라, COVID-19와 같은 전염병에도 실시가 가능하여 대규모 전염병에 손쉽게 대응할 수 있다는 이점이 있다.

(바) [국내특허출원] 스마트팜 기자재 공간 위치 인식 장치 및 그 방법 (출원: '20)

센서 및 구동기 같은 스마트팜 기자재의 경우 현재는 제어기와 유선으로 통신하는 노드들이 많지만, 무선으로 통신하는 노드들이 점차 많아지고 있다. 유선의 경우 배선에 따른 작업 효율 저하의 원인이 될 수 있고, 소규모의 스마트팜 시설에서는 유선과 무선의 차이가 크지 않더라도, 다수의 시설로 이루어진 대규모 스마트팜 온실이나 또는 과수원같은 노지농업 분야에서는 무선으로 스마트팜 기자재를 설치하는 것이 유선으로 설치하는 것에 비해 상대적으로 유리하기 때문이다.

특히 근래 많이 도입되고 있는 드론 같은 실시간 이동형 기자재들은 무선연결이 필수적이며, 향후 미래 지향적 기자재를 활용하기 위해서는 무선통신이 점차 보편화 될 것으로 전망된다. 한편 IoT기술은 급속한 발전에 따라 점차 생활 속의 편리성을 더해주고 있다. 바코드, QR코드, RFID, NFC 등은 인터넷 및 스마트폰 보급이 보편화 됨에 따라 이제는 누구나 쉽게 이 기술을 사용한 정보 매체를 생성할 수 있게 되었다.

이와 같은 매체를 활용하여 누구나 연락처 같은 자기가 원하는 정보를 쉽게 타인과 교환할 수 있게 되었다. 심지어 NFC등을 활용하면 NFC등을 지원하는 IT기기간 설정까지도 쉽게 조작할 수 있는 수준까지 발전하였다.

본 기술은 스마트팜에 활용되는 각종 기자재의 설치 및 배치를 손쉽게 하고, 더 나아가 기자재의 이동 설치를 손쉽게 함으로써 기자재의 재배치를 통해 재사용성을 높이고자 함이다. 멀티 팩터 측정이란 스마트팜 기자재의 설치를 다양한 방법으로 수행한다는 의미이다. 구체적으로는, 스마트팜 기자재의 설치에 있어, 기자재 자체의 정보와 기자재가 설치되는 위치의 정보를 분리하여 두 가지 정보요소를 함께 처리하는 방식으로 수행한다. 정보요소를 처리하는 방식은 근래 많이 활용되고 있는 바코드, QR코드, RFID, NFC 등을 활용한다.

각 스마트 온실은 생육하는 작물(crops)의 종류에 따라 각기 다른 종류의 스마트팜 기자재가 사용되고, 스마트팜 기자재가 배치되는 위치에 따라 동일 종류라 하더라도 복수의 기자재가 사용된다. 일례로, 온도 및 습도 센서의 경우 온실내 위치, 크기 및 성장점등 작물의 상태에 따른 위치 등 측정하고자 하는 목표에 따라 다수의 온도 센서가 설치될 수 있으며, 정밀한 값을 얻기 위해 동일한 위치에 다수의 온도센서를 배치할 수도 있다. 다음 그림의 예시와 같이, 온실내 대기온도를 관찰을 위해 온실의 상단에 온도센서가 설치될 수 있으며, 센싱 값의 안정적 획득을 위해 이중화 하여 설치될 수 있다. 온실의 중단에는 작물의 생장에 적합한 온도와 습도를 측정하기 위해 온도센서 및 습도센서를 설치할 수 있으며, 이는 작물의 생장에 따라 설치위치가 변경될 수 있다. 온실의 하단에는 작물의 근권 환경을 측정하기 위한 온도센서 및 습도 센서가 설치될 수 있다.

장치ID	위치ID	설치시간	GPS좌표	등록단말	비고
센서1	위치1	Timestamp1	x1, y1, z1	Smartphone1	
센서2	위치2	Timestamp2	x2, y2, z2	Smartphone1	
센서3	위치3	Timestamp3	x3, y3, z3	Smartphone1	
센서4	위치4	Timestamp4	x4, y4, z4	Smartphone1	
센서5	위치5	Timestamp5	x5, y5, z5	Smartphone1	
센서6	위치6	Timestamp6	x6, y6, z6	Smartphone1	
센서7	위치7	Timestamp7	x7, y7, z1	TabletPC2	
센서8	위치8	Timestamp8	x8, y8, z8	TabletPC2	
센서9	위치9	Timestamp9	x9, y9, z9	TabletPC2	
센서10	위치10	Timestamp10	x10, y10, z10	TabletPC2	
센서11	위치11	Timestamp11	x11, y11, z11	TabletPC2	
센서12	위치12	Timestamp12	x12, y12, z12	Smartphone3	
센서13	위치13	Timestamp13	x13, y13, z13	Smartphone3	
...	...	...	...	...	
센서N	위치N	TimestampN	xN, yN, zN	등록단말N	

그림 11 - 스마트팜 기자재 배치정보 예시

(사) [국내특허출원] 스마트 팜에서의 비료 사용량 처방 시스템 (출원: '21)

본 사업에서는 효과적인 작물의 성장을 위한 비료 표준사용량 처방 정보를 손쉽게 활용하는 방법 및 시스템에 관한 특허를 출원하였다.

본 발명과 관련한 종래의 기술을 살펴보면, 스마트팜에서 작물을 효과적으로 재배하기 위해 적정량의 비료를 사용하는 것이 중요하다. 관련하여 농업진흥청은 2021년 현재 흙토람 서비스를 제공하고 있다. 흙토람 서비스를 통하여 작물의 품종과 재배환경인 토양 정보를 토대로 비료 표준사용량을 알려주는 비료사용 처방서를 처방을 받을 수 있다.

농업인은 재배지의 토양을 농업기술센터에 제출하여 처방서를 받은 후 농업인이 처방서에 맞춰서 비료의 양을 조절해야 한다. 토양 정보를 얻기 위한 시설이 각 농가에 없으므로 토양을 제출하는 것은 불가결하지만, 처방서 발급과 이후 적용하는 방법이 수동적이라는 점은 개선의 여지가 있다.

수동적 처방서 활용에서 자동 처방서 활용으로 개선하기 위하여 스마트팜의 제어기와 농진청 서버간 연동을 필수적이며, 본 발명에서는 스마트팜의 제어기와 비료 표준사용량 처방 정보 제공 서버와의 연동 방법을 제시한다.

다음 그림은 비료처방 서버와 스마트팜 제어기의 연동 개요도이다.

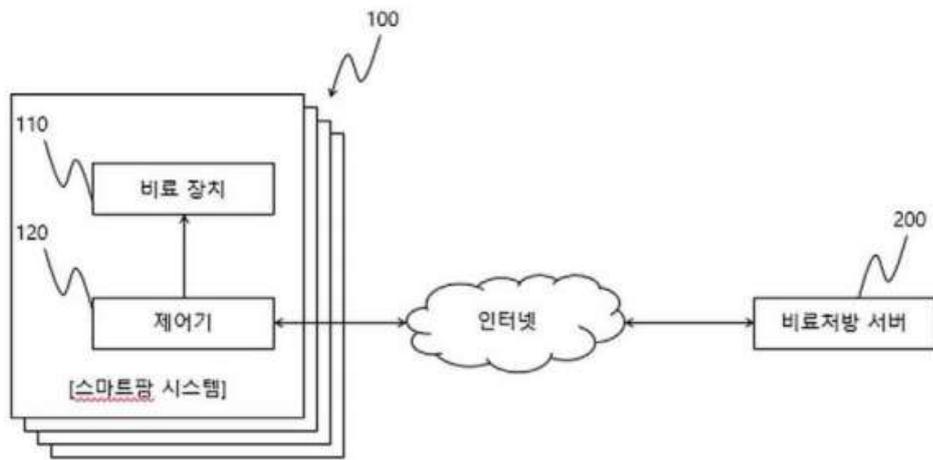


그림 12 - 비료처방 서버와 스마트팜 제어기의 연동 개요도

본 발명에서 스마트팜은 네트워크로 연결된 제어기와 상기 제어기로 제어될 수 있는 비료 장치를 포함한다. 상기 비료 장치는 비료 살포 장치, 비료 제조기, 및 비료자동투입기 등일 수 있으며, 비료 장치는 제어기의 제어에 따라서 비료의 구성 및 용량을 제어한다. 스마트팜이 설치된 재배지의 토양을 재배할 작물 정보와 함께 제출하면 상기 토양을 분석하여 상기 작물을 재배하기 위해서 가장 적절한 구성의 비료 처방서가 작성된다. 이때, 상기 토양 및 상기 작물 정보를 제출할 제출처는 현재 지역 농업기술센터이나 본 발명에서 이로 한정하지 않는다. 또한, 상기 처방서를 요청할 때 스마트팜에서 사용할 비료의 종류를 제공할 수 있다. 이는 비종에 맞춰서 처방서가 발급될 수 있기 때문이다. 상기 비료 처방서는 현재 지역 농업기술센터에서 발급하고 있으나 본 발명에 이로 한정하지 않으며, 본 발명에서는 도1의 비료처방 서버가 스마트팜으로 처방서를 발급한다. 처방서는 스마트팜의 제어기로 직접 발급된다. 이때, 상기 처방서는 제어기가 처방서를 해석할 수 있도록 표준화된 방법으로 작성되어야 한다.

본 발명에서는 상기 처방서가 특정한 언어나 형식으로 작성되도록 한정하지 않는다. 또한, 상기 처방서는 스마트팜의 제어기가 비료 장치를 제어토록 처방된 것이며, 스마트팜을 보유한 농업인이 볼 수 있도록 현재와 같은 형태로 별도로 처방을 받을 수 있다. 스마트팜의 제어기는 상기 처방서가 상기 비료처방 서버로부터 도착하면, 이를 해석하여 비료 장치를 제어한다. 상기 비료 장치에는 사용할 작물의 비료를 보유하고 있다. 작물의 비료를 미리 보유하는 것은 작물마다 사용하는 비료가 일반화되어 있으며, 상기 처방서가 사용할 비종에 따라서 비료의 사용량을 처방할 수 있기에 농업계에서는 일반적이다. 비료의 보유는 탱크의 형태로 보유할 수 있다. 비료 장치는 제어기의 제어에 따라서 비료의 양을 조절하여 사용할 수 있다.

다음 그림은 본 발명에서의 제어기의 동작 순서도이다.

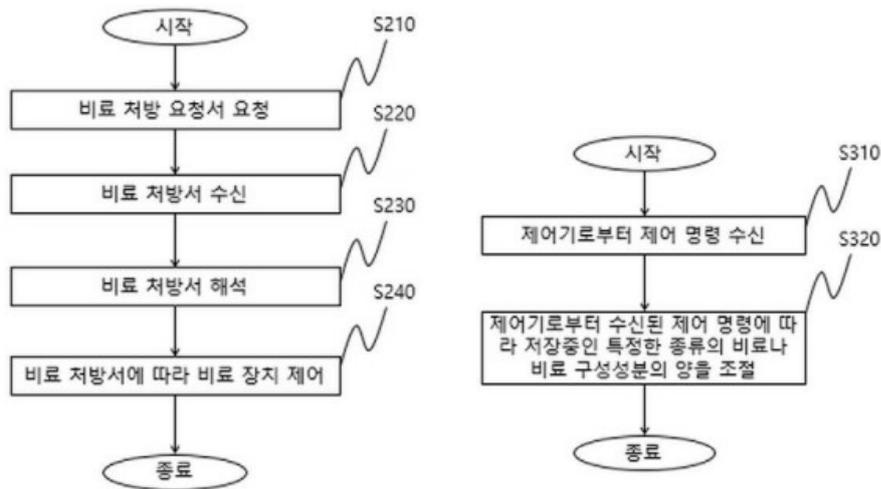


그림 13 - 비료처방 서버와 스마트팜 제어기의 동작 순서도

본 발명에 따른 제어기는 크게 세 가지의 동작을 수행한다.

첫 번째는 비료처방 서버로부터 처방서를 수신하는 것이다. 이를 위해서 제어기는 네트워크로 비료처방 서버와 연결되어 있어야 한다. 처방서를 수신하는 방법은 본 발명에서 한정하지 않으나, 이메일과 같은 형태로 수신할 수 있고 비료처방 서버에서 직접 제어기에 접속하여 데이터 형태로 전송할 수 있다. 상기 직접 접속 방법은 상기 비료처방 서버와 상기 제어기가 어떠한 통신 프로토콜을 통해서 이루어질 수 있다. 예컨대 제어기가 HTTP 웹서버이고 비료처방 서버가 제어기에 할당된 URL을 통해서 접속할 수 있다. 또는 이러한 서비스를 위해서 별도로 만들어진 통신 프로토콜일 수 있다.

두 번째는 비료처방 서버로부터 수신한 처방서를 해석하는 것이다. 상기 처방서에는 처방서 신청 시 제공한 특정 비료의 사용 정보가 적혀있을 수 있다. 또한 상기 처방서에는 비료의 구성 정보가 있을 수 있다. 상기 구성 정보는 질소, 인산, 칼리로 구성될 수 있으나 그로 한정하지 않는다. 또한, 상기 처방서는 밑거름 비료에 대한 처방과 웃거름 비료에 대한 처방을 포함할 수 있다. 상기 제어기가 처방서 해석을 위해서 상기 처방서는 제어기가 처방서를 해석할 수 있도록 표준화된 방법으로 작성되어야 한다. 본 발명에서는 상기 처방서가 특정한 언어나 형식으로 작성되도록 한정하지 않으나, 제어기와 비료처방 서버간에는 사전에 처방서 작성 및 해석에 대한 규칙이 약속되어야 한다.

마지막은 비료 장치의 제어이다. 제어기는 비료 장치에게 특정 비료를 특정한 양만큼만 사용하도록 제어할 수 있다. 또한, 상기 처방서가 특정 비료의 양이 아닌 비료의 구성 정보로 온 경우 제어기는 비료의 구성 정보에 따라서 구성 성분의 양을 조절할 수 있다. 상기 비료의 구성 성분별로 양을 조절할 경우, 비료 장치는 특정 비료가 아닌 비료의 구성 성분들을 포함하는 것으로 가정한다.

(아) [국내특허출원] 스마트팜 데이터 전송 장치 및 그 방법 (출원: '21)

스마트팜은 기존의 전통적 농업에 IoT 등 ICT기술을 융합하여 생산성과 효율성을 제고함으로써 고부가가치 창출을 추구하는 첨단 농업 기술로서, 다양한 IT기기가 활용된다. 스마트팜은 적용분야에 따라 작물을 생육하는 스마트온실, 가축을 키우는 스마트축사, 과수를 생육하는 스마트과수원 등으로 분류될 수 있다. 즉, 스마트팜은 생육하는 대상에 따라 분류될 수 있으며, 생육하는 대상의 품종에 따라 다시 세분화된다. 이렇듯 스마트팜에는 영농의 종류 및 방식에 따라 다양한 형태의 시설 및 장치들이 구비될 수 있으며, 스마트팜 데이터는 각각의 경우에서 매우 다양한 종류로 생성된다.

다음 그림은 통상적인 스마트 온실의 구성 예시이다.

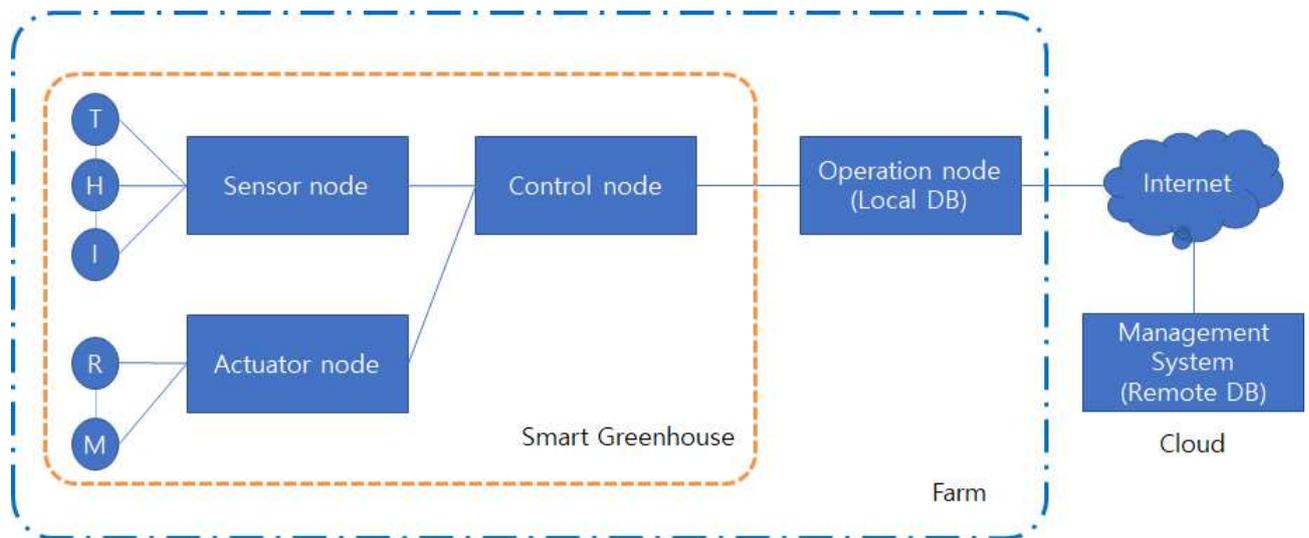


그림 14 - 스마트팜 구성 예시

그림과 같이, 각 스마트 온실에는 센서 모듈(T,H,I 등) 및 구동기 모듈(R, M 등) 들과 각각 연결된 센서 노드(Sensor node) 및 구동기 노드(Actuator node)와 이들로부터 온실의 환경 및 기자재의 상태 데이터를 수신하고 이들에 대한 제어 명령을 전달하는 제어 노드(Control node)가 구비된다. 센서의 종류는 온도(Temperature), 습도(Humidity), 조도(Illuminance) 등 측정하고자 하는 데이터에 따라 다양하게 구비될 수 있으며, 구동기 종류 또한 릴레이(Relay), 모터(Motor) 등 고유의 동작에 따라 다양하게 구비될 수 있다.

이렇게 구성되는 스마트 온실은 하나의 농가에 다수가 구비될 수 있으며, 각 스마트온실마다 구비된 제어 노드를 통해 농가의 전체 운영을 담당하는 운영 노드(Operation node)에 연결된다. 각 제어 노드는 구동기 노드를 제어하는 신호를 전달하고, 또한 센서 노드 및 구동기 노드가 전송하는 데이터를 운영 노드로 전달한다.

운영 노드는 복수의 온실을 통합하여 관리함으로써 농가가 운영하는 각 온실의 상태를 전반적으로 관찰하고 통합적으로 제어하는 운영 시스템처럼 사용되기 때문에, 농가에서 생성되는 스마트팜 데이터는 운영 노드에서 취합되고 관리될 수 있다.

각 농가의 운영 노드는 클라우드의 관리 시스템과 연결될 수 있으며, 이 경우 관리시스템은 다수의 원격지 농가의 운영상태를 파악하고 분석하여 최적의 운영 조건을 찾아 다시 각 농가의 운영 노드를 통해 피드백 할 수 있게 된다. 따라서 각 농가로부터 수집되는 데이터들은 클라우드 관리 시스템에서 취합되고 관리되게 된다.

이를 통해 클라우드는 다양한 스마트팜 서비스를 제공할 수 있는데, 스마트팜 서비스의 예로는 스마트온실에서 생성하는 생육 및 환경 정보를 분석하여 최적생육모델을 도출하는 스마트팜 데이터 분석 서비스, 스마트 온실내 작물의 생육상태를 영상으로 전송받아 원격에서 분석하여 작물의 생육 상태나 병해충 식별을 해주는 원격 진단 서비스 등이 될 수 있다.

물론 이런 서비스들은 농가 내의 운영 노드에서 자체 데이터를 가지고 수행할 수도 있지만, 클라우드에서 처리되는 데이터의 양 및 분석 결과와는 차이가 있을 수밖에 없다.

■ 스마트팜 데이터의 종류 및 특성

스마트팜에서 생성되고 수집될 수 있는 데이터는 온실, 축사, 노지 등 스마트팜의 종류에 따라

다양하겠지만, 크게 시설, 환경, 생육, 경영의 4가지 분야로 분류될 수 있다.

시설 데이터는 스마트온실과 같이 스마트팜 시설의 기초정보를 위한 데이터로서, 농가 정보, 시설 정보, 담당자 정보 등이 포함된다. 시설 데이터는 스마트팜 관리를 위한 목적으로 주로 사용되므로 한번 생성된 데이터는 잘 변경되지 않는 특성이 있다.

환경 데이터는 시설 내외부의 센서와 구동기 등의 스마트팜 기자재들로부터 생성되는 정보들로서, 정해진 주기에 따라 자동으로 측정되는 특성이 있다.

생육 데이터는 작물의 생장에 따라 측정되어 생성되는 정보로서, 작물의 종류에 따라 매우 다양한 특성이 있다. 예로서, 동일한 스마트온실에서 재배되는 시설원에 작물이라 하더라도 토마토, 파프리카, 딸기 등 작물의 종류에 따라 생육 데이터의 종류는 매우 다양하다. 또한 일부에서는 인공지능 기술을 도입하여 생육 데이터 수집을 자동화하는 시도가 있으며, 아직까지는 장치에 의한 자동측정과 숙련된 인력에 의한 수작업을 통한 측정이 병행되어 이루어지고 있지만, AI기술의 발전으로 인해 향후에는 생육데이터의 장치에 의한 자동 생성이 보편화 될 것으로 전망되고 있다.

경영 데이터는 스마트팜 경영에 소요되는 비용 및 수익과 관련된 정보로서, 생산 비용, 수확 정보, 판매 정보 등이 포함될 수 있다. 경영 정보는 농가의 수익성을 드러내기 때문에 통상 잘 공개되지 않는 특성이 있으며, 경매 등의 시장 정보 등을 통해 추정하는 경우가 많다.

결국 스마트팜 데이터에서 실시간 전송이 필요한 것은 환경 및 생육 데이터이며, 이들은 센서 및 구동기 등의 장치에 의해 지속적으로 측정 및 생성되고, 생성된 데이터는 사람의 확인이 없이 사전에 정의된 방법에 따라 사전에 정의된 주소로 자동으로 전송된다.

여기서 외부로 전달되는 데이터는 결국 통신회선을 활용하게 되는데, 각 데이터 특성 및 종류와 무관하게 데이터 송수신 채널을 사용하게 된다. 결국 송수신할 데이터의 특성과 데이터를 실어나를 통신채널의 특성이 전혀 연관되지 않게 동작하게 된다.

결국 데이터 송수신에 대한 비용은 전적으로 데이터 통신회선의 특성에 의존될 수 밖에 없다. 또한 다양한 통신회선을 구비한다 하더라도, 각 통신회선에 어느 데이터를 보낼지는 물리적으로 구분할 수 밖에 없으므로 매우 비효율적으로 운영될 수 밖에 없다.

#### ■ 데이터 특성을 반영한 스마트팜 데이터 전송

다음 그림은 스마트팜 데이터 전송장치의 기능요소를 나타낸다. 본 발명에 따른 스마트팜 데이터 전송장치는 사용자 인터페이스부, 데이터 프로파일 정의부, 데이터 입력부, 데이터 분류부, 데이터 처리부, 이력관리부, 데이터 출력부의 기능요소로 구성된다.

사용자 인터페이스부는 스마트팜 데이터 특성을 정의하는 프로파일을 정의하고, 해당 프로파일에 따라 스마트팜 데이터 전송장치가 동작할 수 있도록 장치를 조작하고, 스마트팜 데이터 전송장치의 상태 및 동작 결과를 확인 할 수 있는 사용자 인터페이스 기능을 수행한다. 사용자 인터페이스부의 구현은 통상 원격접속이 가능하도록 웹기반으로 구성되는 것이 바람직하다.

데이터 프로파일 정의부는 스마트팜 데이터 전송의 기준이 되는 데이터 프로파일을 정의하고, 정의된 프로파일에 따라 스마트팜 데이터 전송장치가 동작할 수 있도록 데이터 프로파일을 관리하는 기능을 수행한다. 각 데이터 프로파일에는 각 데이터의 특성을 구분할 수 있도록 각 데이터 항목 구성과 각 데이터 항목 구성에 따라 어떤 방법으로 데이터를 전송할 것인지에 대한 데이터 전송 규칙이 포함된다.

데이터 입력부는 스마트팜 데이터의 송수신 과정에서 농가의 내부 장치에서 스마트팜 데이터 전송장치로 전달되는 데이터를 수신하는 기능을 수행한다. 데이터 입력부는 농가에 구비된 장치에 따라 다양하게 통신 인터페이스를 구비하는 것이 바람직하다. 다만 스마트팜 농가 내부의 데이터 처리 기능을 담당하는 특성상 주로 근거리 통신에 활용되는 인터페이스를 구비하는 것이 바람직하다. 통상 스마트팜에 활용되는 장치들은 데이터를 전송하기 위해 통신 모듈이 필수적인데, 각 장치의 특성을 고려하여 통신 모듈 또한 다양하게 존재한다. 일반 농가의 스마트온실 등과 같이 실내 시설 기반의 스마트팜에서는 경제성 및 호환성 또는 다른 이유로 유선 기반의 단순 통신 모듈을 활용하는 것이 유리하지만, 대규모 스마트온실 및 스마트 노지 등과 같이 유선 보다 무선 인프라가 더 적합한 시설에는 다양한 무선 통신 모듈을 갖추고 무선 기반의 데이터 관리를 하는 것이 유리하다. 스마트팜 장치가 어떤 통신 모듈을 사용하든, 데이터 입력부는 해당 장치의 데이터를 수신할 수 있도록 해당 장치의 통신 인터페이스를 구비하는 것이 바람직하다.

데이터 분류부는 사전에 정의된 데이터 프로파일에 따라 데이터 입력부로부터 전달된 데이터를 분류하는 기능을 수행한다. 분류된 데이터에는 마킹, 태깅 등 스마트팜 데이터 전송장치가 인식할 수 있는 방법으로 각 데이터 프로파일이 표시된다.

데이터 처리부는 데이터 분류부에서 분류된 데이터를 프로파일에 따른 별도의 채널로 전달하는 기능을 수행한다. 분류된 데이터는 각 데이터 프로파일에 따라 각기 다른 방식으로 처리하게 되는데, 이는 프로파일에 따라 각각 별도의 데이터 전송 채널이 생길 수 있음을 의미한다. 이는 각각 다른 데이터의

특성을 반영하기 위함이다. 예로써 온실내 대기온도 데이터는 지연이 1시간 이내이고, 축사내 CO2 및 약취데이터는 10분 이내로 설정된 데이터 프로파일을 적용할 수 있다.

이력 관리부는 데이터 처리부에서 처리되는 결과를 보관하는 기능을 수행한다. 추후 데이터 전송 및 장치의 동작에 대한 이력 분석을 통해 더 정교한 데이터 프로파일 생성이 가능하고, 다양한 밴드의 장비를 운영하는 경우 개별 장비의 오작동 판단을 위한 근거로 활용된다. 예로써 데이터 프로파일이 정의되지 않은 데이터가 입력될 경우, 데이터 분류부는 해당 데이터에 대해 디폴트 마킹을 하여 데이터 처리부로 보내게 되며, 데이터 처리부는 디폴트 마킹된 데이터 채널로 처리하게 된다. 이 과정이 반복되어 디폴트 데이터 처리가 많아지게 되면 새로운 데이터 프로파일 정의가 필요함을 인지할 수 있다.

데이터 출력부는 각각의 데이터 채널에 따라 스마트팜 데이터를 외부 장치로 송신하는 기능을 수행한다. 외부로 연결되는 통신회선은 최소 하나 이상 구비하게 되고, 복수로 구비하는 경우 데이터 프로파일에 따라 가장 적합한 통신회선을 통해 전송된다. 데이터 출력을 위한 통신회선은 데이터 전송의 안정성, 경제성 등을 이유로 다수의 통신회선을 이용하게 되고, 이에 따라 데이터 출력부의 구성은 다수의 통신 인터페이스로 구성될 수 있다. 다만 복수의 통신 인터페이스를 활용하는 경우 경제성 및 안정성 면에서 함께 사용할 경우 시너지가 큰 조합의 활용이 바람직하다. 통상 통신회선을 선택할 때 가장 타당한 이유는 적합한 전송 성능에 따른 합리적인 비용인 만큼, 다수의 통신 인터페이스를 활용하는 것은 큰 이득이 될 수 있다. 예로써, 근래 많이 사용되는 LPWA 기술만 하더라도 세부 구현에 따라 성능 및 비용차이가 크기 때문에 여러 기술을 복합적으로 사용하게 되면, 데이터 프로파일 정의에 의한 데이터 전송 채널 선택의 폭이 다양해질 수 있으므로, 효율적 데이터 전송 관리를 통한 비용 절감이 용이하다. 한편, 데이터 입력부와 달리 데이터 출력부는 농가의 데이터를 외부의 시스템에 전송하는 것이 주된 기능이므로, 근거리 통신망(LAN) 보다는 원거리 통신망(WAN)에 적합한 통신 인터페이스를 갖추는 것이 바람직하다.



그림 15 -스마트팜 데이터 전송장치 기능요소

다음 그림은 스마트팜 데이터 프로파일 정의를 위한 데이터 항목 구성 예시이다. 스마트팜 데이터 프로파일을 구성하기 위해서는 다양한 요소들이 활용될 수 있지만, 스마트팜내 데이터를 다루는 각 장치들이 인식할 수 있는 항목이어야 한다. 따라서 데이터 프로파일을 정의하기 위한 구성요소는 장치 주소, 측정 데이터 타입, 장치 설정 데이터, 제어기 주소, 제어기 상태 데이터, 제어 명령 데이터 등이 될 수 있으며, 이외에도 기타 필요한 구성요소를 사용자 정의 데이터로서 추가할 수 있다.

장치는 스마트팜내에서 활용되는 센서 및 구동기 등의 장치로서 각 장치는 고유의 측정 데이터를 고유의 프로토콜에 따라 제어기로 송신하게 된다. 이때 각 장치들을 구분하는 식별자는 해당 통신 프로토콜의 주소가 될 수 있다. 여기서 장치 주소는 장치의 통신채널을 식별하는 구성요소이기 때문에 사용하는 통신 프로토콜의 특성에 따라 장치 주소는 장치 ID로 대체될 수 있다.

측정 데이터 타입은 각 장치가 측정하는 값의 종류를 나타낸다. 예로, 온도 센서의 측정 데이터 타입은 온도가 되며, 온습도 센서의 측정 데이터 타입은 온도 및 습도가 된다. 즉 측정 데이터 타입은 해당 장치에서 생성하는 데이터의 종류를 나타낸다.

장치 설정 데이터는 스마트팜의 원활한 운영을 위해 장치에 특별히 설정된 값을 나타낸다. 예로 온도 센서의 경우 특정하는 온도의 유효 범위, 섭씨나 화씨와 같은 측정하는 온도의 단위, 측정하는 온도의 증분 단위 등이 될 수 있다. 즉 장치 설정 데이터는 해당 장치가 가지고 있는 고유의 특성을 반영한다.

제어기는 스마트팜내 장치들로부터 고유의 프로토콜에 따라 수신한 데이터를 스마트팜내 모든 데이터를 총괄하는 운영노드로 전달하게 된다. 제어기는 스마트팜 내 복수로 존재할 수 있으며, 이때 제어기 주소는 각 제어기를 식별하는 역할을 한다. 장치 주소와 마찬가지로 제어기 주소 또한 제어기와 운영 노드를 연결하는 통신 프로토콜의 주소가 될 수 있다. 또한 마찬가지로 제어기 주소는 제어기 ID로 대체될 수

있다.

제어기 상태 데이터는 운영의 효율성을 위해 제어기에 특별히 설정된 값을 나타낸다. 예로 특정 제어기에 연결된 장치의 우선순위 등이 될 수 있다. 해당 제어기에 특정 위치에 따라 다수의 온도 센서가 연결될 경우 먼저 확인되어야 할 위치의 온도 센서 등이 될 수 있다. 동일한 형식의 온도 데이터임에도 작물의 줄기가 체감하는 온실내 기온, 작물의 뿌리가 위치하는 배지 또는 땅의 지온, 작물의 열매가 체감하는 엽온 등은 각각 의미가 다르다.

제어 명령 데이터는 제어기가 인식하는 장치 및 제어기에 대한 제어를 위한 데이터를 나타낸다. 제어 명령 데이터에는 장치 및 제어기의 상태에 대한 조건이 포함된다. 즉 특정 장치 및 제어기의 상태가 특정 조건을 충족할 경우 실행되는 제어 명령이 될 수 있다. 예로, 특정 온도 센서의 측정 데이터 값이 특정 온도를 충족할 경우 실행되는 명령이 될 수 있다.

이 외, 스마트팜 운영에 따른 고유의 장치가 있을 경우 사용자 정의 데이터가 추가로 구성될 수 있다. 다만 해당 사용자 정의 데이터는 임의의 데이터가 아닌 스마트팜 데이터 전송장치가 인식할 수 있는 데이터 형식이어야 하며, 이에 따라 장치, 제어기, 운영 노드 등에서 생성 가능한 데이터로 정의되는 것이 바람직하다.

장치 주소	측정 데이터 타입	장치 설정 데이터	제어기 주소	제어기 상태 데이터	제어 명령 데이터	...
----------	-----------------	-----------------	-----------	------------------	-----------------	-----

그림 16 - 스마트팜 데이터 프로파일 정의를 위한 데이터 항목 구성 예시

(자) [국내특허출원] 영상장치를 이용한 식물 생육 측정 시스템 및 그 방법 (출원: '21)

기존에 식물의 생육 과정을 측정하기 위한 방법으로는 실측과 영상을 이용한 측정 방법이 존재한다. 실측 방법은 사람이 수동으로 식물의 길이나 무게를 재는 방법을 의미하며, 영상을 이용한 측정 방법은 비연속적인 식물의 생육 상태를 고가의 피노타이핑 기술을 적용하거나 컴퓨터 비전 기술을 이용하여 상태를 측정하는 것이었다.

하지만 식물의 생육의 경우 외부의 영향(빛, 수분, 영양분 등)에 즉각적인 반응을 하지 않기 때문에, 종래 기술들은 어느 한 시점에서 식물의 생육 상태를 판단하기에 적합하지만, 식물의 생육 과정을 연속성 있게 관측하기에는 이런 특이 사항과 외부 영향에 대한 고려가 부족하다는 한계에 직면하곤 한다.

본 발명은 전술한 종래기술의 문제점을 해결하기 위하여, 작물 생육 장치를 통하여 대상 작물의 생육 정보를 획득하고, 네트워크에 존재하는 시스템을 통해 작물의 생육 정보 해석하여, 사용자가 수동으로 작물의 생육 상태를 측정하지 않고서도 비-파괴적인 방법으로 연속적인 작물의 생육상태를 측정할 수 있는 방법을 제공하고자 한다.

본 발명에서 이루고자 하는 ‘식물의 생육 측정’을 위하여 다음 그림과 같은 시스템 구성도를 갖는다. 식물 생육상태의 영상을 획득하기 위한 생육 센서 (100)는 일반적으로 컴퓨팅을 하기위한 기본적인 프로세서, 메모리, 저장장치 이외에도, 영상을 획득하기 위한 카메라 모듈 (101), 피사체와의 거리를 측정하거나 조도를 측정하기 위한 하나 이상의 센서 모듈(102), 각종 상태를 알려주거나 동작 명령을 내리기 위한 상태 알림 및 입력 모듈 (103), 위치를 감지하기 위한 모듈 (104) 및 네트워크에 접속하기 위한 유무선 통신 인터페이스(105)로 구성된다. 여기서 생육센서(100)는 관측하기 위한 대상체인 피사체(300)에 대한 영상 데이터와 함께, 영상을 획득했을 당시 생육센서에 포함되어 있는 다양한 모듈로부터 획득한 정보를 생육해석시스템(200)으로 보낸다.

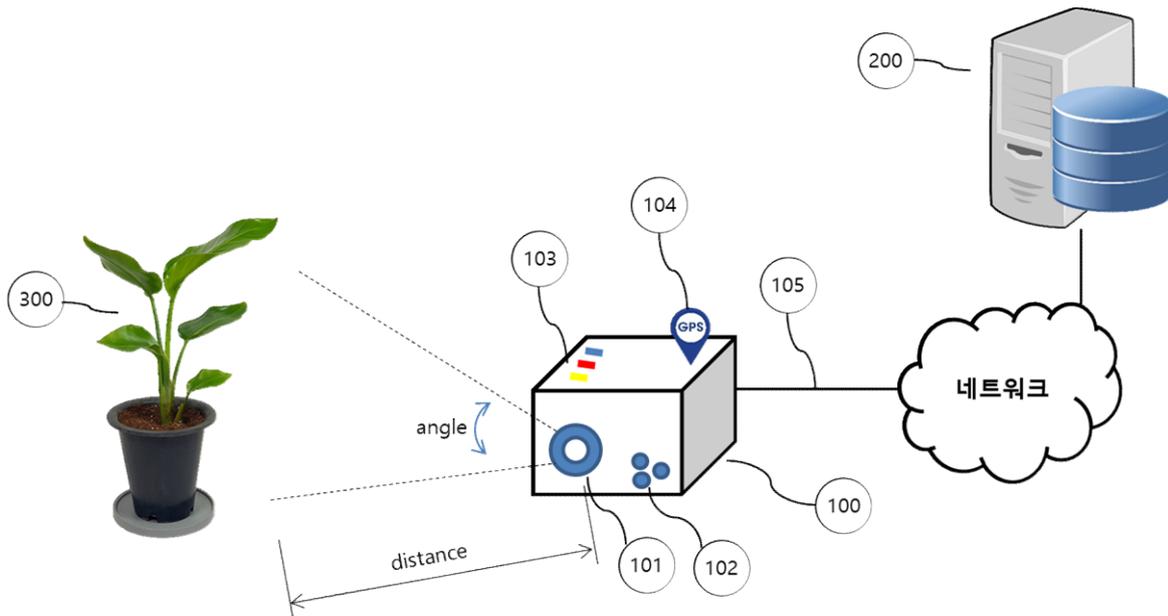


그림 17 - 영상장치를 이용한 식물 생육 센서 시스템 구성도 (대표도)

생육분석시스템은 생육 센서로부터 대상작물에 대한 데이터와 영상데이터를 획득한다(402). 이러한 데이터는 이미 확보한 학습데이터를 활용하여, 대상 작물에 대한 특이점 (위치 또는 포지션)을 특정한다(403). 이때 하나의 영상에서 특이점을 특정할 경우, 인식이 되질 않는 경우가 발생할 수가 있기 때문에 본 발명에서는 이전의 영상데이터와의 상관관계를 통해 인식을 보다 높일 수 있는 방법을 고안하였다.

이는 직전에 특이점을 특정한 이전 영상 데이터와 비교하여, 403단계에서 특정한 특이점의 개수와 위치가 동일한지를 판단한다(404). 만일 동일할 경우, 특이점에 대한 라벨링 (잎, 줄기, 꽃받침 등)과 위치정보를 파악 (또는 기록) 한다(405).

대상 작물의 특이점을 특정하였기 때문에, 대상 작물의 색상을 분석한다. 이때 조명(혹은 일사량)에 의해 색상이 변조될 수 있기 때문에, 생육 센서는 조도에 관한 정보를 영상데이터와 함께 제공한다. 생육분석시스템은 획득한 영상 정보를 조도 정보에 따라 캘리브레이션을 한 후, 대상작물에 대해 이전에 학습한 데이터를 활용하여, 색상을 특정한다(406).

앞의 경우에서와 마찬가지로 이전 특이점과의 상관관계를 비교함으로써, 이전 특이점과 동일한지를 판단한다(407). 만일 이전 영상과 동일한 색상을 가지지 않는 경우, 이는 대상 작물의 특정 부위 (예, 잎, 줄기 등)에 변화가 발생하였음을 의미하는 것이므로, 변화된 색상에 따른 질병 혹은 수분 부족 등의 진단을 내린다(408). 만일 이전 영상과 동일한 색상을 가지는 경우, 이는 특기 사항이 발생하지 않는 것이기 때문에, 다음 차수의 영상 분석을 위해 특이점 라벨링 및 색상정보를 파악 및 기록한 후 (409), 이 절차를 종료한다(410).

404단계에서 대상 작물의 특이점이 이전 영상 데이터에서 라벨링한 특이점과 동일하지 않을 경우, 이는 영상 인식의 오류가 발생하거나, 대상 작물에 변화가 발생한 경우이다.

영상인식의 오류가 발생한 경우를 극복하기 위해, 본 발명에서는 사용자가 설정한 threshold 횟수만큼, 이전의 영상데이터와의 상관관계를 비교한다.

우선 이전 데이터와 차이점을 발생되었을 경우, 이전 영상에서 기록된 특이점 위치를 중심으로 영상을 재검색한다(411). 식물의 경우, 품종마다 다소 차이가 발생하지만, 영상데이터의 획득 속도가 작물의 생장 속도보다 빠르다면, 이전 영상의 특이점의 위치와 대동소이하기 때문에, 만일 영상인식에 오류가 발생하여 이전 영상에 비해 특이점의 개수가 적거나 잘못된 위치를 판별하였을 때, 상관관계를 통해 이러한 오류를 극복할 수 있다.

만일 특이점의 위치에 관한 재검색을 성공하였을 경우(412), 404단계를 반복한다.

만일 재검색에 실패하였을 경우, 신뢰도를 높이기 위해, 사용자가 설정한 threshold 만큼의 이전 영상 데이터들과 비교를 한다(413). 즉, 현재 시간을 T라고 하면, 직전 시간을 T-1로하여 T-threshold만큼 이전의 영상데이터를 비교한다는 것을 의미한다. threshold를 초과하지 않았을 경우, 신뢰도를 높이기 위해 411단계를 반복한다. threshold를 초과했음에도 계속하여 이전 특이점과 상이할 경우, 이는 새로운 위치에 새로운 특이점이 발생하거나 없어진 것을 의미한다. 만일 일정 횟수를 반복함에도 특이점이 소멸된 것으로 분석이 되면(416), 특이점에 해당되는 대상체(예, 잎, 열매)가 사라졌음을 판단한다(418). 이와는 반대로, 새로운 특이점이 생긴 것으로 분석이되면(417), 이는 특이점에 해당되는 대상체(예, 잎, 열매)가 새로 발생하였음을 판단하고(419), 절차를 종료한다(420).

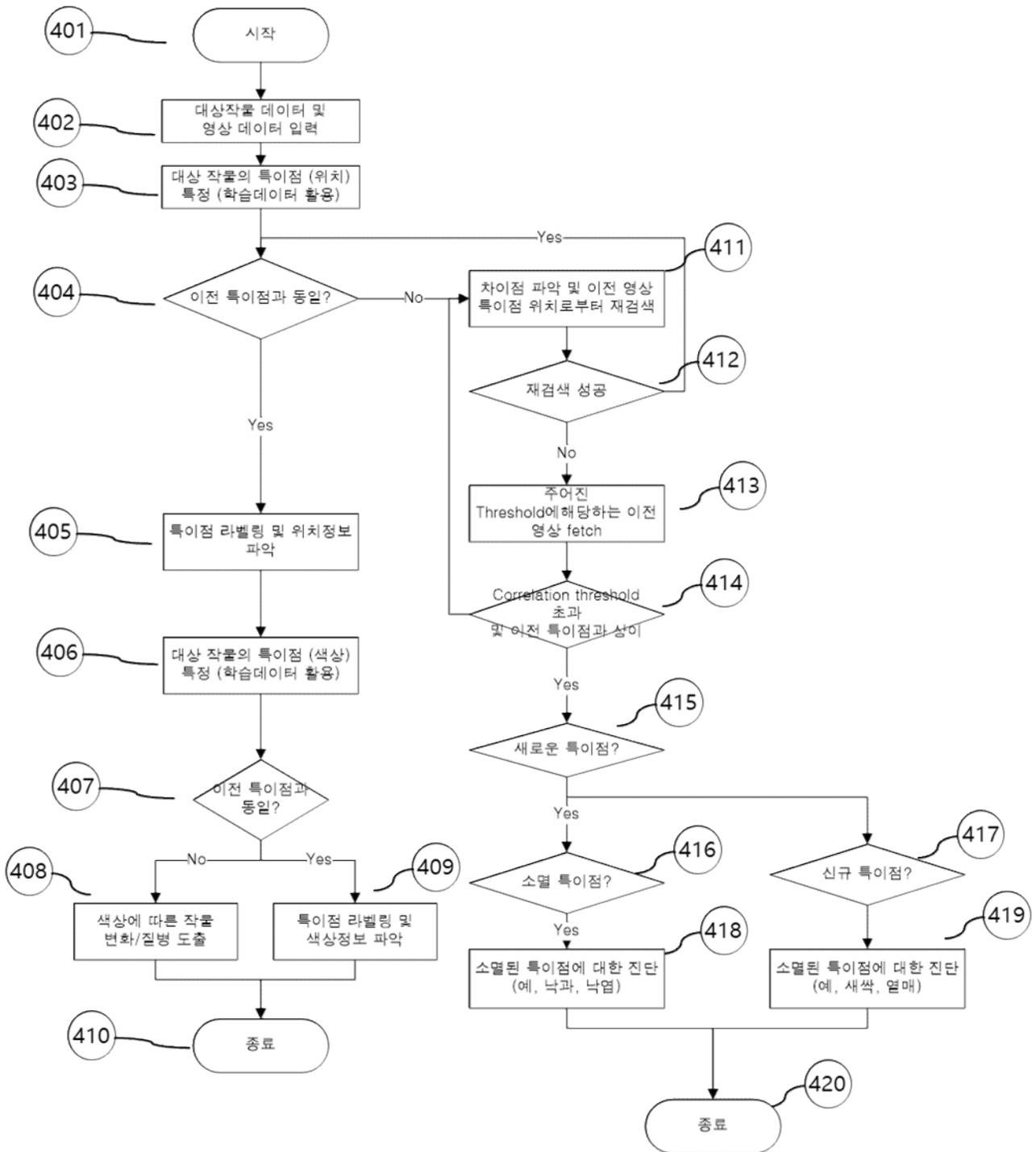


그림 18 - 상관관계 기반 생육 해석 방법

위의 과정을 통해 대상 작물의 특이점 변화를 파악할 수 있으며 이는 아래 그림의 실시예에서처럼 간단한 DB를 구축할 수 있다. 이때 시간, 특이점 위치, 특이점 판별, 해당 영상데이터 및 기타 센싱정보를 포함한다.

시간	특이점위치	특이점판별	영상데이터	기타 센싱정보
12:00	{X:1, Y:1} {X:0, Y:0}	잎 끝 줄기 끝	filename001.jpg	조도: 500Lx 거리: 1M 해상도: 200만
13:00	{X:5, Y:3} {X:2, Y:1}	잎 끝 줄기 끝	filename002.jpg	조도: 450Lx 거리: 1M 해상도: 200만
14:00	{X:8, Y:6} {X:5, Y:3}	잎 끝 줄기 끝	filename003.jpg	조도: 600Lx 거리: 1M 해상도: 200만

그림 19 - 특이점 위치 저장 데이터 구조 실시예

다음 그림은 위에서 획득한 정보를 토대로 대상 작물의 성장 상태를 해석하는 실시예이다. 아래 그림의 왼쪽에는 특이점의 위치와 특이점(잎끝=파란색, 줄기끝=빨간색)을 표시한 것이며, 이에 대한 실제 대상 작물의 성장 상태는 그림의 오른쪽과 같이 시각 12:00의 잎 형태는 501 단계가, 시각 13:00의 잎 형태는 502 단계가, 시각 14:00의 잎 형태는 503 단계가 나타난다.

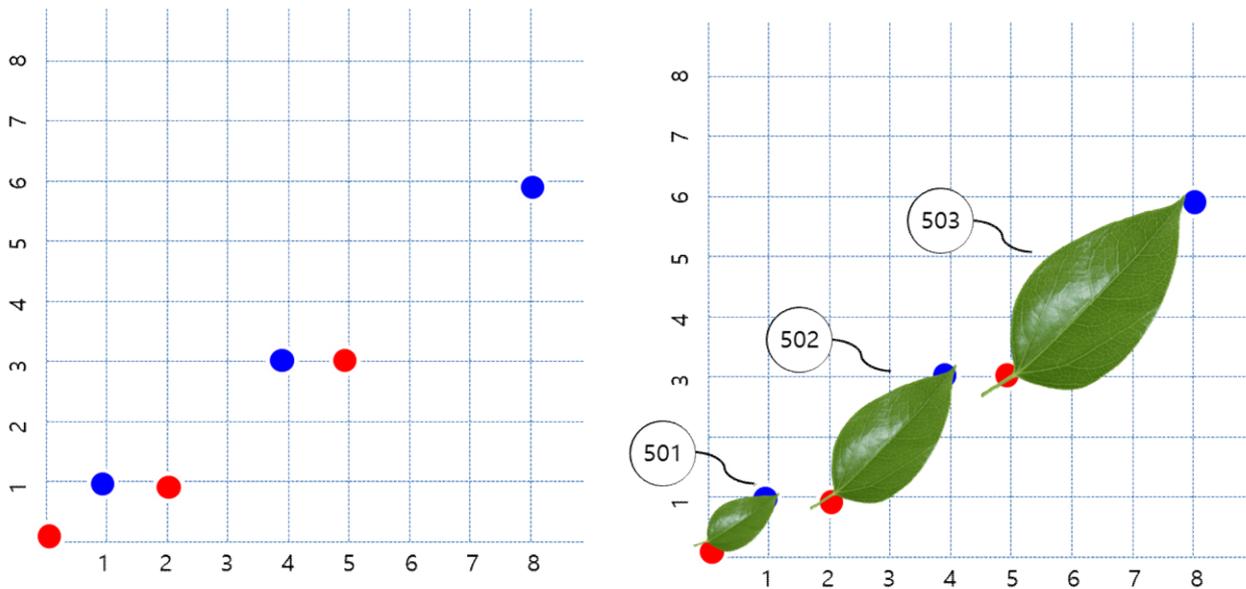


그림 20 - 특이점 위치 상관관계에 따른 작물의 성장 해석실시예

(차) [국내특허출원] 컨테이너를 활용한 조사료 재배 장치 및 그 시스템 (출원: '22)

본 발명은 전술한 종래기술의 문제점을 해결하기 위하여, 재배 장치는 축산농가에서 저렴하게 사용할 수 있도록 기존 목적에 꼭 필요하지 않은 자동화 설비를 최소화하면서도 그 효과를 얻을 수 있도록 하며, 조사료의 파종이나 수확 시 재배 장치의 오염을 최소화하도록 하고, 재배 공간을 청결하게 유지할 수 있도록 청소 등이 용이하며, 재배 장치 내의 부속품의 교체가 용이하도록 하는 것을 목적으로 한다.

또한 재배장치 수의 증가에 따라 재배 장치의 관제를 위한 장치를 추가 구축하기 위한 별도의 비용이 증가하지 않도록 하며, 사용자가 시간과 장소에 구애받지 않고도 원격에서도 재배장치를 손쉽게 모니터링하고 환경을 조절할 수 있도록 하는 것을 그 목적으로 한다.

본 발명에서 컨테이너를 활용한 조사료 재배 장치와 그 시스템을 이루고자 [도1]에서와 같은 시스템 구성도를 갖는다.

그림 21에서 재배장치(101)는 조사료를 재배하기 위해 컨테이너를 활용한 조사료 재배 장치를 의미하며, 재배서버(102)는 클라우드 기반으로 조사료 재배 환경을 감시하고 조절하기 위한 기능을 제공하는 서버이며, 재배자응용(103)은 농장주가 새싹보리등과 같은 조사료를 재배하기 위해 필요로 하는 일련의 작업을 가능하게 하는 제어 응용 프로그램이다.

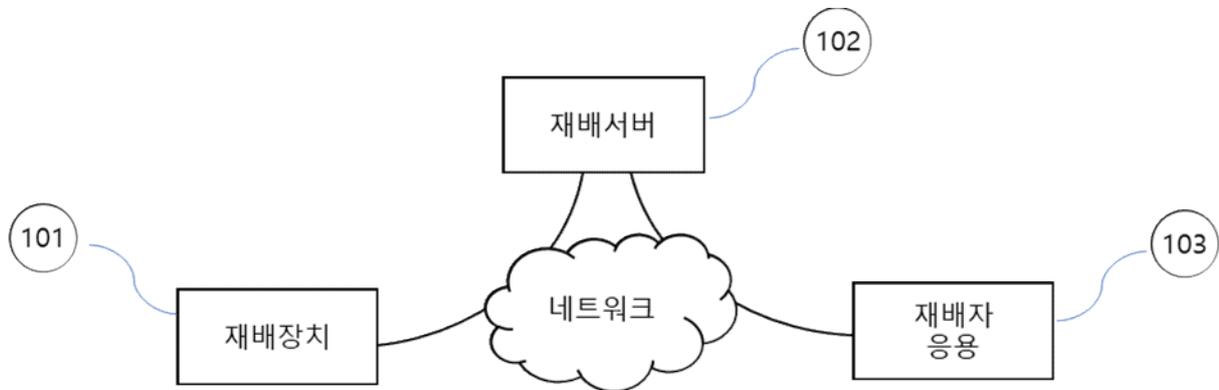


그림 21 - 컨테이너를 활용한 조사료 재배 시스템 구성도 (대표도)

그림 22는 컨테이너를 활용한 조사료 재배 장치 외부 구성도이다.

재배장치(201)는 컨테이너를 활용하여 구성하며, 외형은 조사료 파종 및 수확을 위해 상하로 개폐되는 문(202)과 재배장치 제어 패널을 조작하거나, 온냉방기, 관수펌프 등의 관리, 적재 공간 활용등을 위해 좌우로 개폐되는 문(203)으로 구성된다. 단, 상하로 개폐되는 문(202)의 개수는 문의 무게와 컨테이너의 길이에 따라 가변적이다.

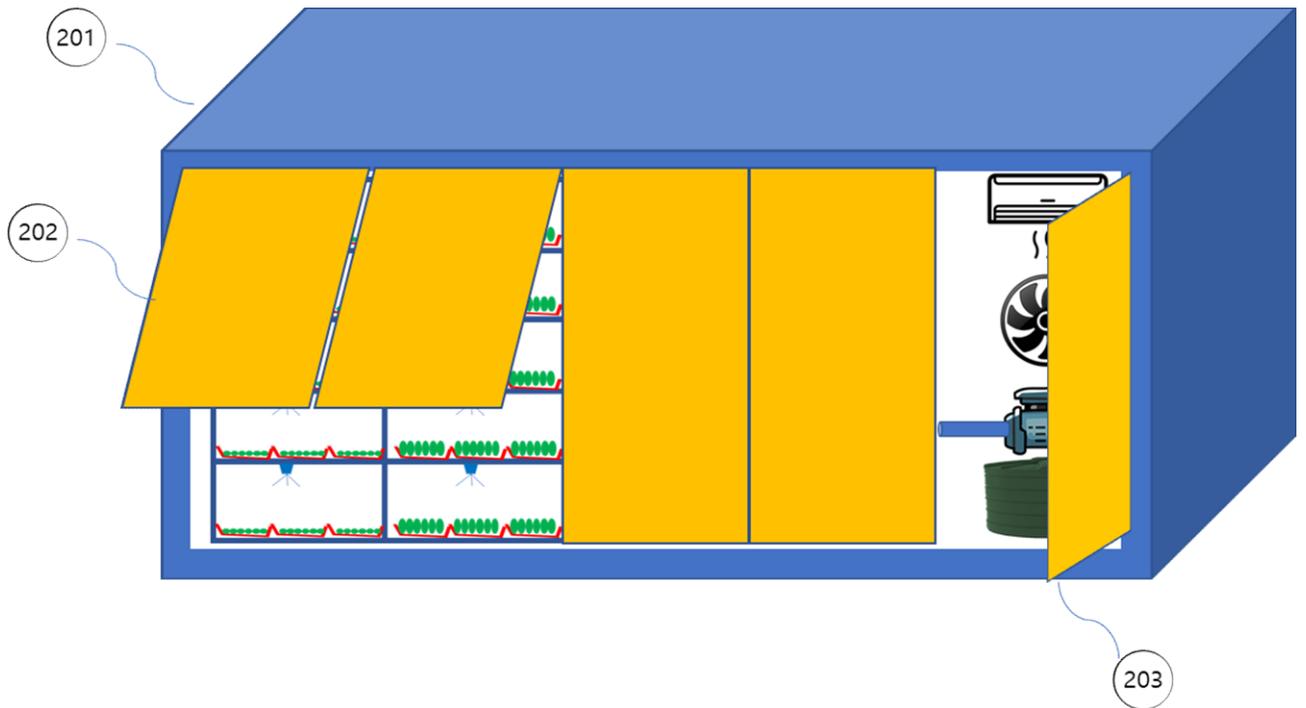


그림 22 - 컨테이너를 활용한 조사료 재배 시스템 전면도

그림 22는 상하개폐문(202)과 좌우개폐문(203)을 제거한 컨테이너를 활용한 조사료 재배 장치 내부 구성도이다. 다음 그림 23의 컨테이너를 활용한 조사료 재배 장치(301)는 파종시기가 다른 다수의 조사료를 재배하기 위해 다단으로 구성된 재배 프레임(302)을 포함한다. 이 재배 프레임은 조사료 재배 트레이(303)의 무게를 고려하여 지지할 수 있는 프레임이 포함되며, 이 재배 프레임의 길이는 컨테이너 크기에 따라 가변적이다.

재배 장치(301)내부의 조명을 위해 벽면에 LED 조명등(304)을 설치하도록 한다. LED 조명등은 재배트레이(303)과 수평이 되도록 설치하며 그 빛이 중앙에까지 미칠 수 있도록 한다.

재배 장치(301)의 양액공급을 위해 조사료의 재배를 위한 양액 또는 물을 미스트 형식의 노즐(305)을 통해 제공하도록 한다. 이때 미스트의 방출 압력은 조사료 씨앗이 흘러내려가지 않도록 조절한다.

재배 장치(301)의 대기환경을 감지하기 위해 온도, 습도, 조도 등 내부 환경 상태 감지와 양액 또는 물의 과잉 공급을 감지하기 위한 센서류 (307)가 포함된다.

재배 장치(301)의 대기환경을 조절하기 위해 내부공기의 환기를 위한 환풍팬(306)과 냉난방기(308), 공기순환팬(309) 등이 포함되며,

조사료 재배를 위한 양액 또는 물을 담기 위한 양액탱크(312)와 이를 미스트 노즐(305)까지 공급하기 위한 양액펌프(311)가 포함된다.

마지막으로 이러한 일련의 작업들을 수동으로 조작하기 위한 수동작업패널(310)이 탑재된다.

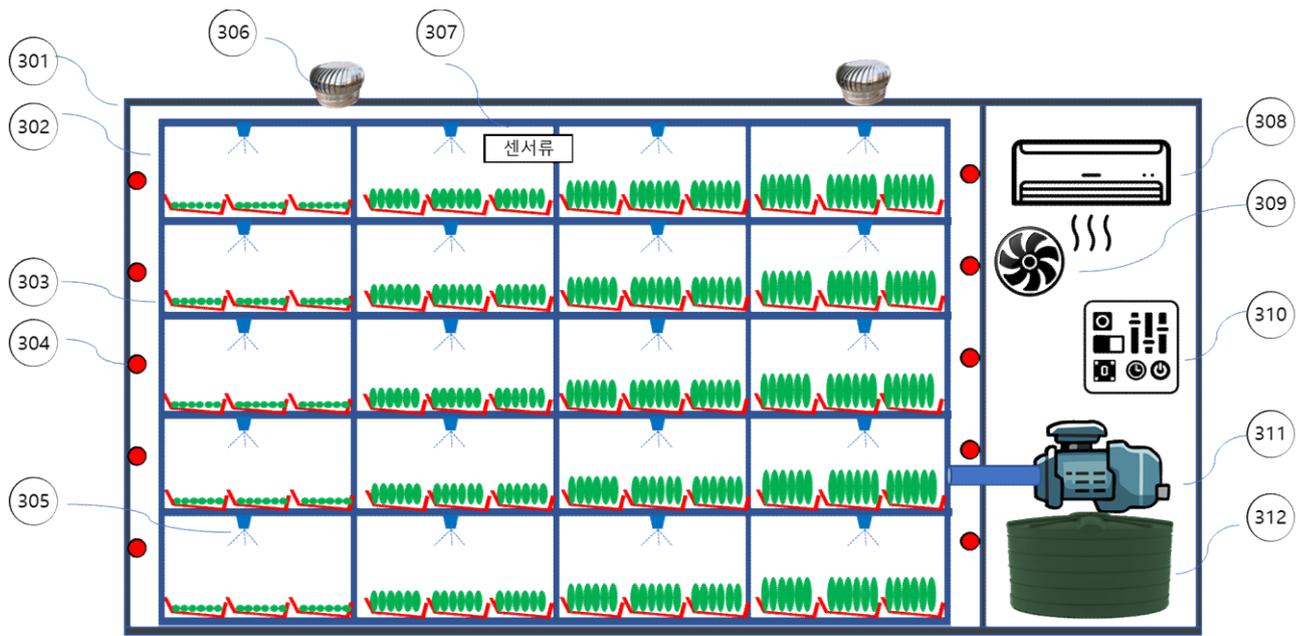


그림 23 - 컨테이너를 활용한 조사로 재배 시스템 내부도

(카) [국내특허출원] 야생동물 식별을 통해 학습효과 방지가 가능한 야생동물 퇴치 시스템 및 그 방법 ('22)

본 발명은 야생동물의 침입을 감지할 경우 이미지 기반으로 야생 동물을 식별하고 식별된 야생 동물을 퇴치하기 위한 효과적인 방법을 경험 기반으로 도출하여 빛, 소리, 고/저주파수 등의 다양한 매체를 이용하여 야생동물을 퇴치하는 것을 그 목적으로 한다.

본 발명에서 야생동물 식별을 통해 학습효과 방지가 가능한 야생동물 퇴치 시스템을 이루고자 [도1]과 같은 시스템 구성도를 갖는다. 그림 24는 야생동물 식별을 통해 학습효과 방지가 가능한 야생동물 퇴치 시스템 구성도로, 동물퇴치기 (102)와 관제 센터(105), 위치와 초점 변경이 가능한 PTZ 카메라(106)로 구성된다.

하나의 동물퇴치기(102) 장치는 야생동물을 감지할 수 있는 하나 이상의 동작 센서(103)과 야생동물을 퇴치하기 위한 하나 이상의 구동기(104)로 이루어진다. 한편 동작 센서(103)의 물리적인 특성에 따라 감지할 수 있는 범위가 한정되기 때문에, 하나의 동물퇴치기가 야생동물을 감지하고 퇴치할 수 있는 구역을 Zone(101)으로 정의한다. 이렇듯 Zone이 한정된 범위를 갖기 때문에 하나의 발을 감지하기 위해서는 다음 수의 동물퇴치기가 필요하다.

$$n(\text{동물퇴치기}) \geq (\text{경계하고자 하는 둘레})/(\text{Zone의 지름})$$

야생동물을 퇴치하기 위하여 야생동물의 침입을 감지한 동물퇴치기(102)는 침입한 야생동물 객체를 식별을 위해 관제 센터(105)에게 무선 혹은 유선 통신 채널(107)을 통하여 야생동물 침입 사실을 알리면, 관제센터(105)는 로컬 혹은 원격에 위치한 PTZ 기능이 있는 카메라(106)에게 고속 통신 회선(108)을 통해 이미지 캡처를 명령하고 캡처한 이미지를 저장하고 이를 식별한 후, 해당 동물을 퇴치할 수 있는 방법을 동물퇴치기(102)에게 지시한다.

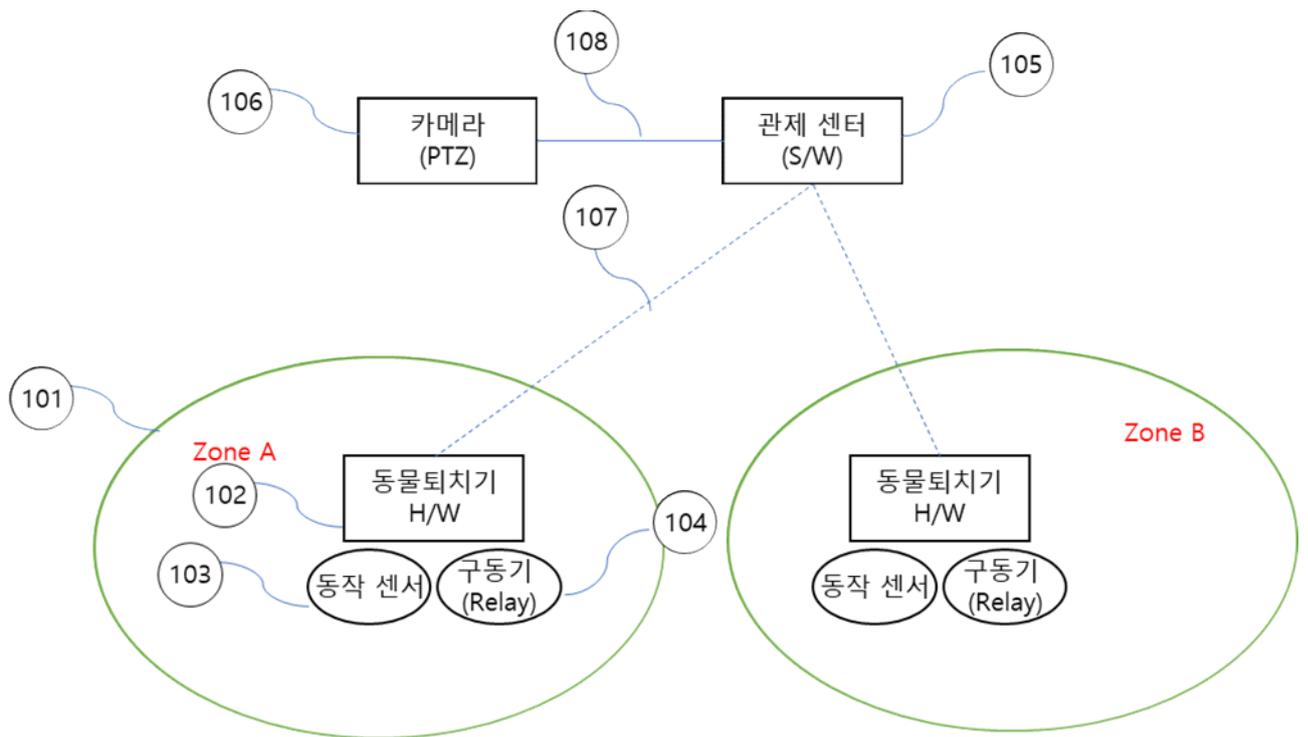


그림 24 - 야생동물 식별을 통해 학습효과 방지가 가능한 야생동물 퇴치 시스템 구성도 (대표도)

그림 25는 다수의 설치된 야생동물퇴치기(102)의 위치를 PTZ 카메라(106)에 설정하는 절차를 도시한 그림이다. 동물퇴치기의 처음 설치 혹은 관리자의 요청에 따라 동물퇴치기 위치 등록을 시작한다(201). 관제 센터는 각 동물퇴치기의 위치를 파악하기 위해, ID 별 동물퇴치기를 호출한다(202). 호출된 동물퇴치기는 설치되어 있는 LED의 섬광을 통해 자신의 위치를 관제센터에게 알린다(203). 관제센터는 PTZ 카메라를 구동하여 해당 동물퇴치기의 위치를 식별한다(204). 관제센터는 카메라에 해당 동물퇴치기의 위치를 등록한 후(205), 만일 다음번 ID가 모든 동물퇴치기를 등록했는지를 확인한 후 (206), 만일 다 되었으면 과정을 종료하고(207), 그렇지 않으면 (202) 단계로 회기한다.

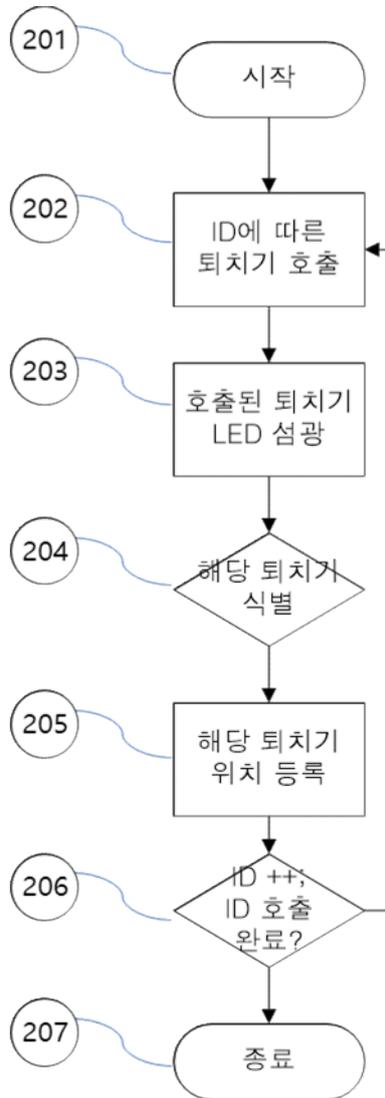


그림 25 - 야생동물 퇴치 장치 자동 등록을 위한 순서도

나) [논문] 국내외 학술대회 논문

본 사업기간을 통해 다음 표와 같이 국내 전문 학술지 또는 학술 대회를 통해 18건의 비SCI 논문을 발표 혹은 게재 하였다.

No	논문명	논문지	년도
1	A STUDY ON MACHINE LEARNING WEB SERVICE USING SPREADSHEETS	ICTC 2017	2017
2	스마트팜 기술 개발 및 표준화 동향	한국통신학회	2017
3	Implementation of docker-based smart greenhouse data analysis platfor	ICTC 2018	2018
4	LCP for Managing Smart Greenhouse	ICTC 2018	2018
5	Implementation of greenhouse service control protocol using Python on Raspberry PI	ICACT 2019	2019
6	스마트온실에서 RS485 MODBUS기반 IoT 디바이스 속성 정보 자동구성 및 데이터 송수신 최적화방안	한국통신학회	2019
7	ETRI Insight 표준화동향: 스마트팜 (12월)	ETRI Insight	2019
8	ITU-T SG20 스마트팜 표준화 동향	SEP inside	2020
9	IoT 기반 스마트온실 서비스 프레임워크 표준 소개	한국통신학회	2020
10	스마트팜 장치 표준 기술 분석 및 검증	한국통신학회	2020
11	Implementation of Smart Farm Devices using Open Source Software	ICACT 2021	2021
12	스마트축산 표준화 동향	JCCI2021	2021
13	기상 IoT 센서 데이터를 활용한 서리 예측 연구	한국인공지능학술대회	2021
14	Understanding IoT Climate Data based Predictive Model for Outdoor Smart Farm	ICTC2021	2021
15	스마트 농업을 위한 유즈케이스 표준개발 연구	한국통신학회	2022
16	데이터기반스마트팜전망	한국통신학회	2022
17	IoT 기반 스마트축산을 위한 서비스 프레임워크 표준화 연구	한국통신학회	2022
18	IoT 기반 강아지 급이기 설계 및 구현	한국멀티미디어학회	2022

다음은 위 논문들 중 선도성이 있는 일부 논문들을 선별하여 소개하도록 한다.

나-1) 도커 기반 스마트 온실 데이터 분석 플랫폼 개발 (ICTC 2018, '18)

ICT 지원 장치가 장착 된 스마트 온실은 대량의 데이터를 사용하여 최적의 성장 환경을 구축함으로써 비용을 절감하고 생산량을 늘릴 수 있었으며, EPIS는 약 5 년 동안 다양한 지역에 위치한 온실에 설치된 다양한 종류의 센서로부터 축적 된 데이터를 가지고 있으며 SOAP 기반 공개 API를 사용하여 일반 사용자에게 원시 데이터를 제공한다. 그러나 원재료의 원재료 자체가 최적의 성장 알고리즘을 얻기에 충분하지 않기 때문에 농업 전문가가 다양한 요인을 쉽게 조합하여 데이터 분석을 위한 플랫폼을 구축해야 한다. 본 논문에서는 EPIS에서 제공하는 Open API를 사용하여 자동으로 데이터를 수집 및 시각화하여 팜 환경 정보를 쉽게 분석 할 수 있는 플랫폼을 구현한 경험을 기술한다. 또한, 구축 된 플랫폼은 Docker를 사용하는 모든 시스템으로의 포팅을 용이하게 하도록 구성하였다. Docker는 시스템 가상화를 통해 다양한 환경에서 안정적인 운영이 가능한 소프트웨어 플랫폼으로 사용자가 컨테이너를 사용하여 응용 프로그램을 신속하게 빌드, 테스트 및 배포 할 수 있게 해주는 소프트웨어 플랫폼이다. Docker를 사용하면 OS 버전 및 장비의 라이브러리 버전에 관계없이 서비스 환경을 구축 할 수 있다. 따라서 우리는 서버 이식성 및 기능 개발의 편의를 위해 Docker를 사용하여 분석 플랫폼을 구축했다. InfluxDB는 시간 열이 내장 된 일종의 시계열 데이터베이스이며 HTTP REST API를 통해 데이터를 제공하므로 다른 시각화 응용 프로그램에서 쉽게 사용할 수 있다. 온실 센서에서 수집 한 환경 데이터를 시계열 정보로 InfluxDB에 저장했다. Grafana는 시계열 데이터를 효과적이고 쉽게 시각화 할 수 있는 일종의 오픈 소스 시각화 도구다. 과거에는 미리 정의 된 스키마를 사용하여 관계형 데이터베이스에 데이터를 저장하고 시각화를 위해 별도의 JavaScript 라이브러리를 사용하여 프로그램을 작성해야 했지만 Grafana는 쿼리를 작성하고 즉시 시각화 할 수 있다. Traefik는 내부 서비스를 숨기는 역방향 프록시이며 특정 트래픽을 방화벽이나 게이트웨이 뒤의 지정된 로컬 서버에 라우팅 한다. 이러한 도구들을 이용하여, 데이터의 수집 및 분석은 다음과 같이 유기적인 동작을 통해 이뤄진다.

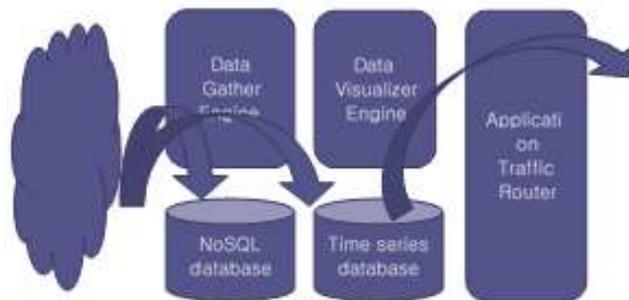


그림 1 - 도커기반 스마트온실 구성도

데이터 수집 엔진은 SOAP API를 사용하여 EPIS 서버에서 데이터를 가져 와서 데이터베이스에 저장합니다. 우리는 두 종류의 데이터베이스를 사용했다. 온실 내의 센서로부터 온 환경 데이터를 위한 정보 데이터 및 시계열 데이터베이스를 위한 NoSQL 데이터베이스. 우리는 mongoDB와 influxDB를 사용했습니다. MongoDB는 일반적인 NoSQL 데이터베이스로 별도의 스키마를 지정할 필요가 없으며 JavaScript로 쿼리를 수행하기 때문에 사용하기 쉽습니다. 또한 InfluxDB는 별도의 스키마를 구성 할 필요가 없으며 시계열 데이터를 전문적으로 관리 할 수 있는 이점이 있다. 데이터 시각화 엔진은 시계열 데이터베이스에 저장된 환경 데이터를 사용하여 그래프를 표시하며, 이를 위해 Grafana를 사용했습니다. 응용 프로그램 트래픽 라우터가 로컬 서비스 응용 프로그램을 숨기고 특정 서비스 응용 프로그램을 노출하기 때문에 traefik 역 프록시를 사용했다.

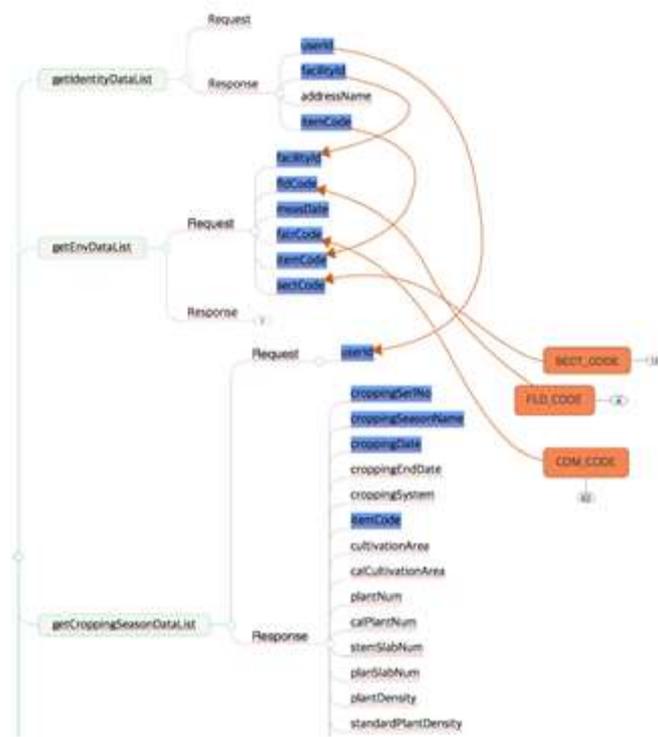


그림 2 - 데이터 수집 엔진 개념도

위 그림은 EPIS 데이터들의 연관관계를 도시화 한 것으로, 실제 데이터를 가져오기 위해서는 위 구조에 대한 파악이 선행되어야 한다.

아래는 도커 이미지를 만들기 위해 사용한 설정파일이다(Dockerfile).

```

: '3.2'
services:
  traefik:
    container_name: "traefik"
    image: library/traefik
    command: -c /dev/null --api --docker --docker.domain=traefik.u2pia.com --logLevel=DEBUG
    ports:
      - "80:80"
      - "8080:8080"
    volumes:
      - ./app/traefik/traefik.toml:/etc/traefik/traefik.toml
      - /var/run/docker.sock:/var/run/docker.sock
    networks:
      - frontend
      - traefik
    deploy:
      placement:
        constraints:
          - 'node.role == manager'

  mongodb:
    image: mongo:latest
    container_name: "mongodb"
    environment:
      - MONGO_DATA_DIR=/data/db
      - MONGO_LOG_DIR=/dev/null
    volumes:
      - ./app/mongodb/data/db:/data/db
    ports:
      - 27017:27017
    command: mongod --smallfiles --logpath=/dev/null # --quiet

  influxdb:
    image: influxdb:latest
    container_name: influxdb
    networks:
      - frontend

```

```

volumes:
  - ./app/influxdb/influxdb:/var/lib/influxdb
ports:
  - 8086:8086

grafana:
  image: grafana/grafana
  container_name: "grafana"
  ports:
    - 9001:3000
  volumes:
    - ./app/grafana:/var/lib/grafana
  networks:
    - frontend
  labels:
    - "traefik.port=3000"
    - "traefik.enable=true"
    - "traefik.docker.network=traefik"
    - "traefik.backend=grafana"
    - "traefik.frontend.rule=Host:grafana.u2pia.com"

networks:
  frontend:
    driver: overlay
  traefik:
    driver: overlay

```

이를 통해 수집된 데이터는 다음과 같이 시각화하여 분석이 용이하도록 하였다.



그림 3 - 시각화 결과

본 논문에서는 Docker를 이용한 여러 가지 마이크로 서비스를 구성하여 농업 데이터 분석 플랫폼을 구현한 경험을 기술했다. 또한 EPIS 개방형 API에서 데이터를 수집하는 방법을 설명하였다. 이 플랫폼은 여러 매개 변수를 한 눈에 파악하고 다양한 매개 변수를 조정하여 각 관계를 쉽게 분석할 수 있는 그래프를 그려 사용자가 최적의 성장 모델을 개발할 수 있도록 도와준다. 다음 단계로 최적의 생산 데이터를 인공지능으로 교육하여 최적의 성장 알고리즘과 환경 변화에 대한 자동 경고를 위한 추가 분석을 계획하고 있다. 그러나 이를 위해서는 데이터에 대한 데이터 품질 관리(DQM)가 필요하다. 정확한 모델을 만들기 위해서는 생산된 데이터의 신뢰성을 유지해야 하므로, 생육데이터 측정의 자동화, 정량화 및 일반화에 대한 연구가 추가적으로 진행될 필요가 있다.

## 나-2) 스마트팜 장치 표준 기술 분석 및 검증 (한국통신학회, '20)

사물인터넷(IoT)이 차세대 ICT산업의 핵심으로 주목 받으며 다양한 영역에 적용하려는 시도가 확대되고 있는 가운데 가장 주목받고 있는 분야 중 하나가 스마트팜이다. 스마트팜은 기존의 농업에 IoT 등 ICT 기술을 융합하여 생산성과 효율성을 제고함으로써 고부가가치 창출을 추구하는 첨단 농업 기술로서, 다양한 IT 기기들이 활용된다.

스마트팜의 일반적인 구성은 생육 및 환경 정보를 모니터링 하기 위한 센서 장치(온도, 습도, CO2, 토양, 양분 등), 스마트팜 시설을 제어하는 구동기(창개폐기, 환기, 난방, 관수, 관비 등), 각 시설에 부착된 장치들을 제어하는 제어기, 그리고 이들을 통합적으로 관리하는 운영관리시스템 등으로 구성된다. 센서 및 구동기는 센서 및 구동기 장치와 이들과 연동되어 제어기를 통해 운영시스템까지 연결되는 센서 및 구동기 노드로 세분될 수 있다. 운영관리시스템은 인터넷을 통해 클라우드와 연결되어 원격에서 스마트팜 관리도 가능한 수준으로 진화하고 있다.

현재 보급되고 있는 스마트팜 장비는 생산업체마다 제품규격이 다르기 때문에 상호 호환성이 보장되지 않는다. 이는 농가의 입장에서는 다양한 기자재를 구비하는데 문제가 된다. 즉, 특정 기능을 가진 다른 생산업체의 장비를 새로이 도입하고 싶어도 기존 장비와 호환되지 않기 때문에 할 수가 없다. 또한 생산업체의 입장에서도 호환성이 없는 경우 신규 구축이 아닌 이상 자사의 장비를 적용할 수 있는 여건이 안되기 때문에 자연스럽게 판매의 입지가 좁아지게 된다. 따라서 스마트팜 기자재의 상호호환성 확보는 사용자 및 제조사에 중요한 요구사항이 된다.

OCF (Open Connectivity Foundation)는 IoT관련 국제 표준화 기구로서 전세계의 많은 IoT기업들이 참여하고 있고, 표준개발과 동시에 오픈 소스 프로젝트도 동시에 진행함으로써, 규모뿐만 아니라 생태계 확장에서도 타 표준화 기구에 비해 비교우위에 있다. OCF는 다양한 IoT 유/무선 연결 기술을 활용하여, 상호 연동성을 이용하여 구축하고 있으며, 스마트홈, 자동차, 물류, 헬스케어 등 다양한 IoT 서비스를 개발할 수 있도록 구성되어 있다.

OCF는 IoT을 구현하는 REST 구조 기반으로, 경량의 프로토콜(coap)로 디바이스들을 연결하고, 서로 연동시킬 수 있다. 특히, OCF의 가장 큰 장점은 무상특허정책으로, OCF 회원으로 가입하면 회원사간 OCF 관련 기술들의 모든 소유 특허는 무상으로 사용 가능하다.

OCF는 리소스 모델에 의해 클라이언트-서버 모델로 동작하게 되는데, 리소스를 제공하는 장치는 서버로, 리소스를 요청하는 장치는 클라이언트로 동작한다. 스마트팜 환경에 적용되는 일례로서, 센서 및 구동기 등의 스마트팜 기자재는 서버로서, 사용자 인터페이스를 통해 스마트팜 시설을 제어할 수 있는 운영관리시스템은 클라이언트로서의 기능이 탑재될 수 있다. 이는 OCF를 지원하는 모든 기기들간 상호 호환성에 따라 스마트폰, 태블릿 등을 포함하여 TV 등과 같은 가전내 가전들까지도 연결될 수 있음을 의미한다.

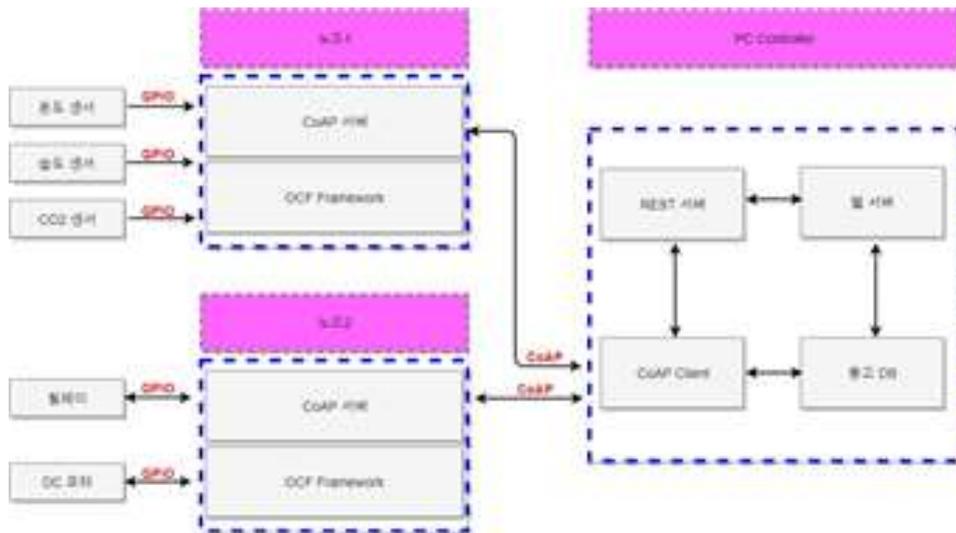


그림 4 - OCF 기반 스마트팜 시스템 PoC 구성

본 논문에서는 다양한 사물인터넷 유무선 연결기술 환경에서 사물들간 논리적인 상호연동성을 보장하는 미들웨어 표준 기술인 OCF가 제공하는 오픈소스 소프트웨어인 IoTivity를 활용한 스마트팜 장치 구현을 통해 스마트팜을 위한 표준기술 분석 및 장치 적용성을 검증하였다.

스마트팜에 설치되는 IoT 장치들은 빈번한 급수와 높은 온도·습도 등의 농작물 양육 환경 등으로 인해

전달되는 상당량의 데이터에 오류와 누락이 발생한다. 현재의 스마트팜 관련 장치 제조사들은 이러한 문제점을 해결하기 위하여 자신만의 노하우로 자신만을 위한 제품을 개발하고 있는 실정이기 때문에, 스마트팜용 IoT 장비 제조사들이 자신만을 위한 제품이 아니라 상호 호환성을 담보할 수 있는 개방형 제품을 제조한다면 규모의 경제를 만들 수 있다.

본 논문에서는 OCF에서 제공하는 오픈소스를 활용한 스마트팜 장치 구현을 통해 스마트팜 적용을 위한 PoC를 검증함으로써, 일반 스마트팜 장치 제조사 혹은 설치 업체들이 쉽게 IoT 장치를 이용하고 운영 관리할 수 있도록 하는 기술적인 가능성을 확인하였고, 향후 스마트팜 기자재간 호환성을 담보하는 개방형 스마트팜 확산에 활용될 수 있으리라 기대한다.

### 나-3) ITU-T SG20 스마트팜 표준화 동향 (SEP Inside, '20)

본 연구에서 진행 중인 국제표준화 동향을 소개하기 위해, 표준특허전문지인 SEP Inside에 관련 논문을 게재하였다.

본 논문에서는 데이터 기반의 스마트팜을 작물이나 가축의 생육 정보와 환경 정보를 수집·분석하여 기존에 투입하던 노동력·에너지·양분 등을 최소화하면서도 더 나은 생육환경을 조성하기 위하여, IoT, 빅데이터, 클라우드, 로봇/자동화, AI 등의 기술을 활용한 스마트온실, 스마트축사나 스마트과수원(노지) 등이라고 정의하고 있다. 이렇게 다양한 ICT 기술들을 활용하여 스마트팜을 구축하기 위해서는 스마트팜 ICT 기자재 간 상호호환 및 데이터 상호연동을 제공할 수 있는 표준 제공이 매우 필수적이기 때문에, 본 논문에서는 수년 전부터 국내외 표준화기구를 통해 활발히 진행되고 있는 스마트팜 분야의 표준화를 소개하고 있다. 다음에서는 본 논문에서 소개하고 있는 스마트팜 분야의 표준화 동향을 정리하도록 한다.



그림 5 - 스마트팜의 정의 및 주요 기술

국내의 경우, TTA PG426에서 스마트팜(온실과 축사) 환경관리에 사용되는 ICT 장치 간 호환성 및 연동성을 제공하기 위한 표준화를 진행하고 있으며, 이중 스마트팜(온실)에서 사용되는 센서와 구동기 인터페이스에 관한 H/W 연결 방식과 메타데이터에 대한 국가표준으로 채택시킨 바 있다.

국외의 경우, ISO와 ITU-T 등의 국제 공적 표준화 기구와 OCF, OneM2M 등의 국제 사실표준화기구에서 스마트팜 관련 표준화를 진행하고 있다. ISO/TC 23에서는 농작업 자동화기기 간 상호접속을 위한 표준화가 주로 진행되고 있으며, ITU-T SG20에서는 식물의 생육 환경과 생육상태를 관찰하고 이를 기반으로 식물의 생육환경을 조절하기 위한 시설원예와 노지의 생육환경 조절을 위한 IoT 기반의 프레임워크 표준화와 IoT 기술을 활용하여 스마트축사의 환경을 관리하기 위해 축사 내 온도, 습도, 이산화탄소, 암모니아 등 축사의 환경정보들을 효과적으로 수집하고 운영할 수 있는 기술에 대한 요구사항이 표준화되고 있다.

한편 사실표준화기구인 OCF와 oneM2M에서는 스마트팜에 대한 표준화 가능성을 열어두고는 있지만, 아직 구체적으로 시작된 표준화 항목은 없는 실정이다.

표 1 - 국내외 스마트팜 표준화 현황

구분	표준화 기구	표준화 현황	
국내	TTA PG426	스마트온실 및 축사 관련 상호호환성 표준, 유통 분야 표준화	
국제 (공식)	ISO	TC 23	자동 착유 시스템, 농작업 및 임업에 사용되는 부착형 작업기, 가축에 대한 전기전자 장비, 무인 드론 자동방제 시스템, 농업용 무선통신 시스템 분야 표준화
		TC 34	동물 복지 분야 표준화
	ITU-T	SG 13	네트워크 기반의 스마트파밍(Y.2238), 네트워크 기반의 예찰 서비스 모델 프레임워크(Y.farms), 융합서비스 기반의 농업정보 서비스 모델(Y.saic) 및 스마트파밍 교육서비스(Y.sfes) 분야 표준화
		SG 20	IoT 기반의 스마트축산 프레임워크(Y.IOT-SLF), IoT 기반의 스마트온실 프레임워크(Y.ISG-FR) 분야 표준화
국제 (사실)	OCF/oneM2M	IoT 장치간 상호운용성을 제공하기 위한 사실표준화기구로서, 현재 스마트홈, 스마트시티 및 스마트공장에 대한 표준 접촉방식과 표준 데이터 모델 분야 표준화 * 스마트팜 분야는 현재 유즈케이스만 제공하고 있는 상태	

[ITU-T SG20 국제표준화 동향]

ITU-T SG20은 2014년 부산에서 개최되었던, ITU-T WTSA(World Telecommunication Standardization Assembly)에서 주요 아젠다로 논의되었던 IoT와 스마트시티 표준화에 대응하기 위해 2016년에 생성된 그룹이다. ITU-T SG20이 신설됨에 따라, IoT 기반으로 스마트팜을 감시 및 제어하기 위한 프레임워크(Y.ISG-FR) 표준과제가 신설되었다. 또한, 스마트축산 분야에서도 IoT 기술이 매우 주요하게 주목받음에 따라, 2018년 5월에 스마트축산 분야에 대한 신규 표준화 과제(Y.IOT-SLF)가 신설되었다.

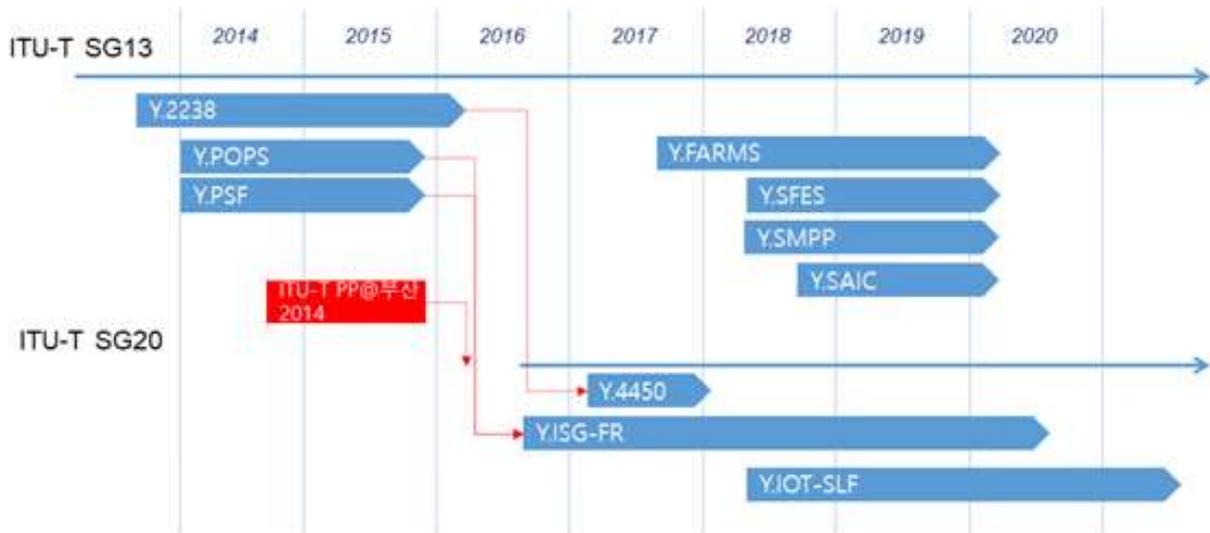


그림 6- ITU-T 스마트팜 관련 표준화 이력

ITU-T SG20에서 추진하는 스마트팜 관련 표준화 소개에 앞서, ITU-T SG13에서의 표준화를 잠시 소개하고자 한다. 이러한 이유는 ITU-T SG20이 신설되기 이전에는 스마트팜과 관련된 표준화가 ITU-T SG13에서 이루어졌었고, ITU-T SG20이 신설됨에 따라 IoT와 관련된 과제들이 이관되었기 때문이다.

ITU-T SG13에서 처음으로 시작된 스마트팜 관련 표준은 2015년에 제정된 Y.2238(Overview of Smart Farming based on networks)이다. 이 표준은 스마트팜을 통해 경작하는 방법을 생산 단계, 생산 이전 단계 및 생산 이후 단계로 구분하여 각각의 단계에 대한 개념을 소개한 표준이다. Y.2238 표준 개발과 함께 2014년부터는 IoT 기반으로 스마트팜 감시와 제어를 위한 스마트팜의 생산 단계 요구사항(Y.PSF)과 작물 생산 단계 이전에 고려되어야 할 계획, 데이터 활용 등과 관련된 요구사항(Y.POPS)에 대한 표준개발이 시작되었다.

2015년 10월, ITU-T SG20이 신설됨에 따라 IoT 기술을 활용하는 스마트팜과 관련된 표준과제들이 ITU-T SG13에서 ITU-T SG20으로 이관됐고, 이들 두 표준과제는 ITU-T SG20에서 진행된 Y.ISG-FR 과제와 통합되었다. ITU-T SG20은 스마트 시티 및 커뮤니티(SC & C)에 관한 IoT 애플리케이션에 중점을 두고 IoT 기술의 표준화 요구사항을 효과적으로 대응하고자, 다음 표에서와 같이 2개의 WP(Working Party)와 7개의 Question을 산하에 두고 있다.

표 2 - ITU-T SG20 표준화 영역

Working Party	Question	목적
WP1	Q1/20	IoT 및 SC&C와 관련된 엔드-투-엔드 연결, 네트워크, 상호운용성, 인프라 및 빅데이터 측면
	Q2/20	산업 분야별 요구사항, 기능 및 사용 사례
	Q3/20	아키텍처, 관리, 프로토콜 및 서비스 품질
	Q4/20	e/스마트 서비스, 애플리케이션 및 지원 플랫폼
WP2	Q5/20	신기술, 용어 및 정의
	Q6/20	IoT 및 SC & C의 보안, 개인정보, 신뢰 및 식별
	Q7/20	스마트 지속 가능한 도시 및 커뮤니티의 평가 및 평가

본 고에서 소개하고자 하는 Y.ISG-FR(Framework of smart greenhouse service)는 Q4/20에서 개발되었으며 2019년에 Y.4466으로 승인되었다. 또 다른 하나는 Y.IOT-SLF(Framework and Capabilities for Smart Livestock Farming based on Internet of Things)로써, 이는 2021년 제정을 목표로 Q2/20에서 개발되는 표준안이다. 다음은 각각의 표준에 대해 간략한 소개를 하고 있다.

1) ITU-T Y.4466(Framework of smart greenhouse service)

온실은 작물생산 및 품질을 개선하고 생산기간을 제어하기 위해서는 환경 조건(즉, 습도, 온도 및 광도)을 제어할 수 있는 격리공간이 필요하기 때문에 출현했다. 스마트온실은 기존 온실에 사물인터넷(IoT) 기술을 적용하여 작물의 생육환경 상태와 작물 재배 상태를 모두 관찰하고 작물 재배 환경을 자동적이고 지능적으로 제어한다. 스마트온실은 작물 재배를 위한 시공간의 자유와 인건비, 에너지 소비 및 비료 소비를 최소화할 수 있게 한다.

다음 그림은 ITU-T Y.4000에서 정의하는 IoT 참조 모델을 기반으로 만들어진 스마트팜 서비스 참조 모델이다. 기존 IoT 참조 모델에 스마트온실에서 필요로 하는 추가적인 6가지 기능을 더하여 스마트온실 서비스 참조 모델을 정의하였다. 스마트팜 서비스를 제공하기 위해 추가로 정의한 기능들은 다음과 같다.

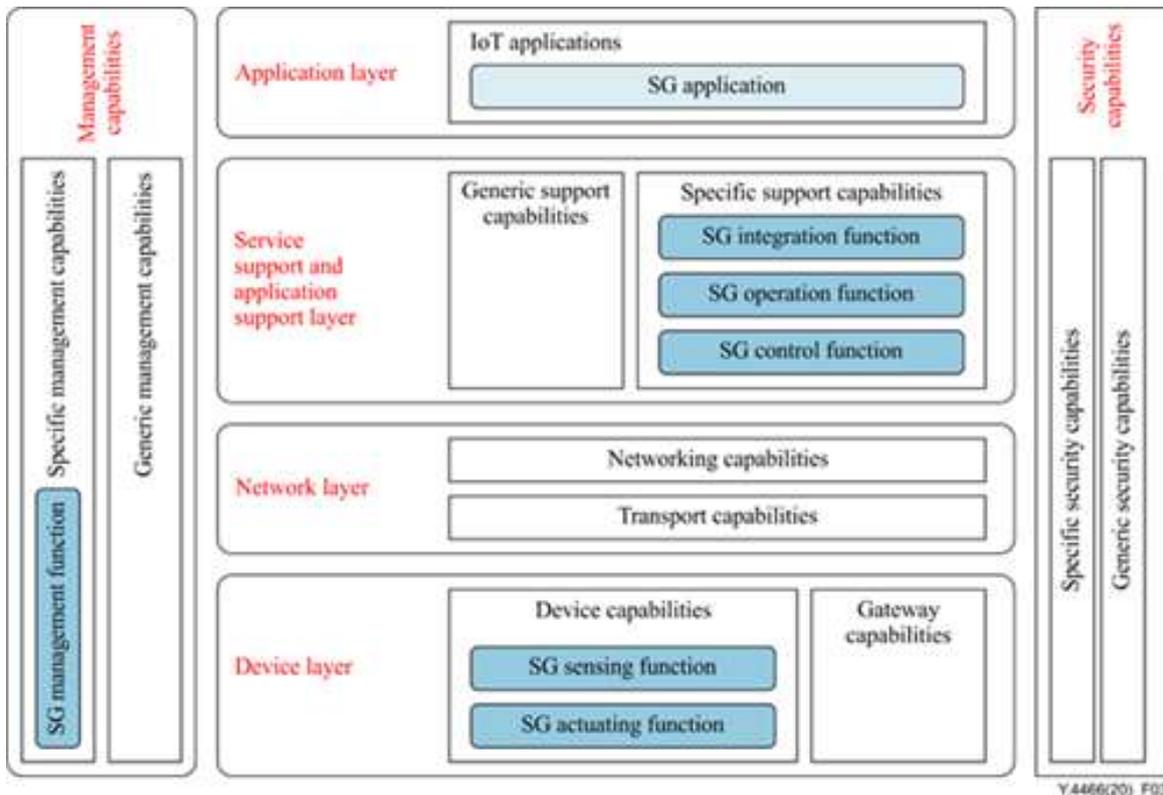


그림 7 - 스마트팜 서비스 참조 모델 (ITU-T Y.4466)

- 스마트온실 감지기능(SG sensing function)은 스마트온실에 설치된 센서를 통하여 환경과 작물의 성장 상태 정보를 수집하여 스마트온실 제어기능으로 전달하며, 각 센서에 대한 정보(유형, 위치, 설명 및 감지 값의 단위 등)를 스마트온실 관리기능에 보낸다.
- 스마트온실 작동기능(SG actuating function)은 스마트온실 제어기능으로부터 받은 제어 명령에 따라 스마트온실에 설치된 액추에이터를 동작시키며, 각 액추에이터에 대한 정보(유형, 위치 설명 및 작동 상태 등)를 스마트온실 관리기능으로 전달한다.
- 스마트온실 제어 기능(SG control function)은 스마트온실 작동기능으로부터 받은 최종 성장 모델 정보나 액추에이터의 작동 상태 정보에 따라 작동기를 동작시키기 위한 작동 제어 명령을 생성한다. 이때의 제어 명령은 스마트온실에 설치된 각 액추에이터의 기능과 각 액추에이터에 대한 추가적인 사용자 요청을 모두 포함한다.
- 스마트온실 작동 기능(SG operation function)은 사용자의 농업 경험을 반영하여 스마트온실의 특정 작물에 대한 최종 환경제어 조건을 생성한다. 스마트온실 작동기능은 또한 스마트온실 제어기능으로부터 받은 환경 및 작물 성장 상태 정보를 스마트온실 통합 기능으로 전달한다.
- 스마트온실 통합 기능(SG integration function)은 스마트온실 작동기능에서 환경 및 작물 재배 상태 정보를 축적한다. 축적하고자 하는 정보의 범위는 스마트온실 단위가 아닌 다수의 스마트온실을 포함하는 농장 혹은 작목반에 따라 정보를 축적하여 전문가 그룹 또는 전문가 시스템의 도움을 받아 최적의 성장 모델을 만들어 저장한다. 생산된 최적 성장 모델은 농민의 추가적인 요구사항을 반영하기 위해 스마트온실 운영기능으로 다시 전달한다.

이러한 일련의 기능을 수행하기 위하여 ITU-T Y.4466은 다음 그림과 같이 16개의 스마트온실 서비스 제공자와 서비스 사용자 간 인터페이스를 정의하였다. 이 중 R1에서 R9까지의 인터페이스는 Y.4466에서 상세한 설명을 제공하지만, R10에서 R16까지의 인터페이스들은 주로 구현에 관련된 사항이기 때문에 해당 표준에서 자세히 언급하지 않는다.

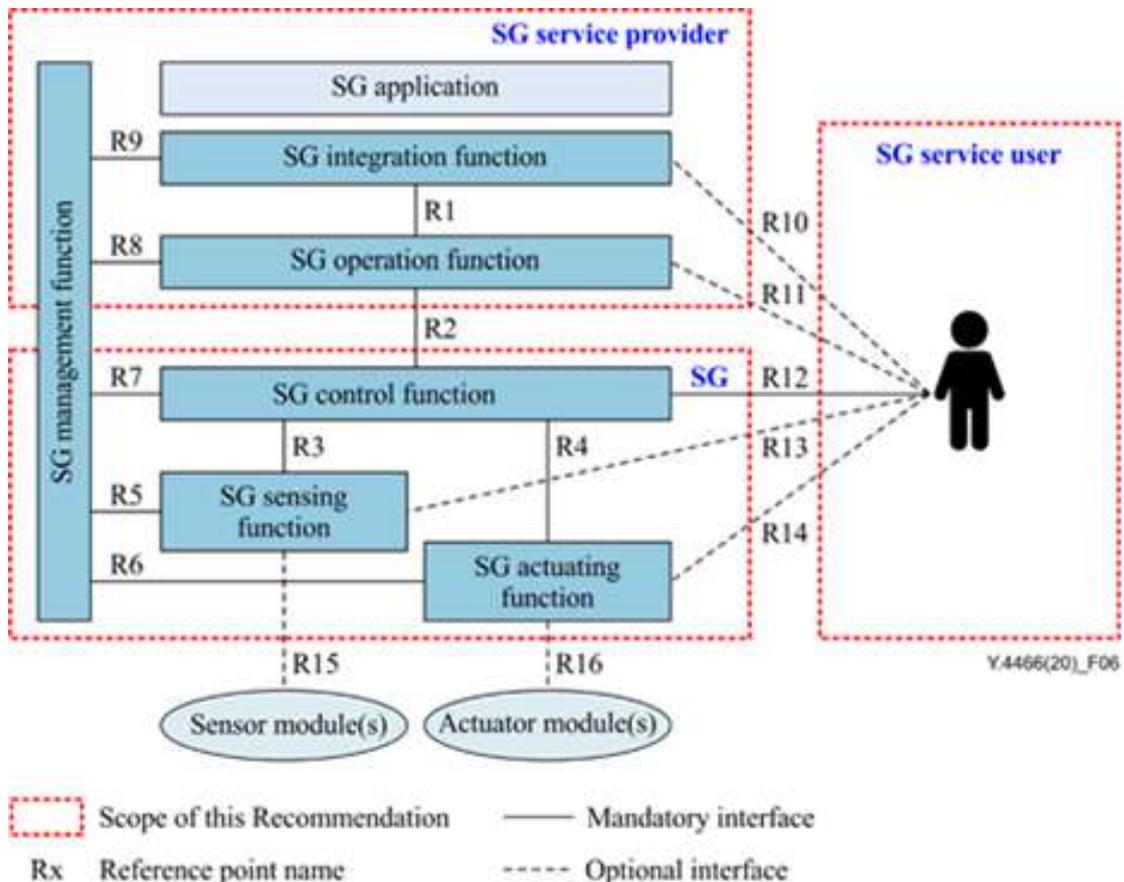


그림 8 - 스마트팜 서비스 인터페이스 (ITU-T Y.4466)

2) Y.IOT-SLF(Framework and Capabilities for Smart Livestock Farming based on Internet of Things)

스마트 축산(SLF)은 정보통신기술(ICT) 을 가축 사육 가치 사슬에 적용한 개념이다. 스마트 축산에 대한 개념이 중요한 이유는 스마트축산을 운영하는 회사는 정밀한 가축 사육 시스템, 관리정보시스템, 가축 사육 자동화 및 로봇 프로세스를 통합하여 더욱 생산적이고 지속 가능한 생산을 가능하게 할 수 있을 뿐만 아니라, 더 나은 의사결정이나 효과적인 사육과 관리를 가능하게 하기 때문이다. 이 표준 기술에서는 IoT 를 기반의 SLF에 대한 개요와 참조 모델을 제시하고, 가축 사육 가치 사슬을 표현하기 위한 서비스 요구 사항과 주요 기능들을 정의한다. 축산물은 일련의 이해 관계자들이 참여하여 생산되고 판매된다. 다음 그림은 Y.IOT-SLF에서 제시하고 있는 스마트 축산에 대한 개념을 제시하기 위한 3계층 개념모델이다.

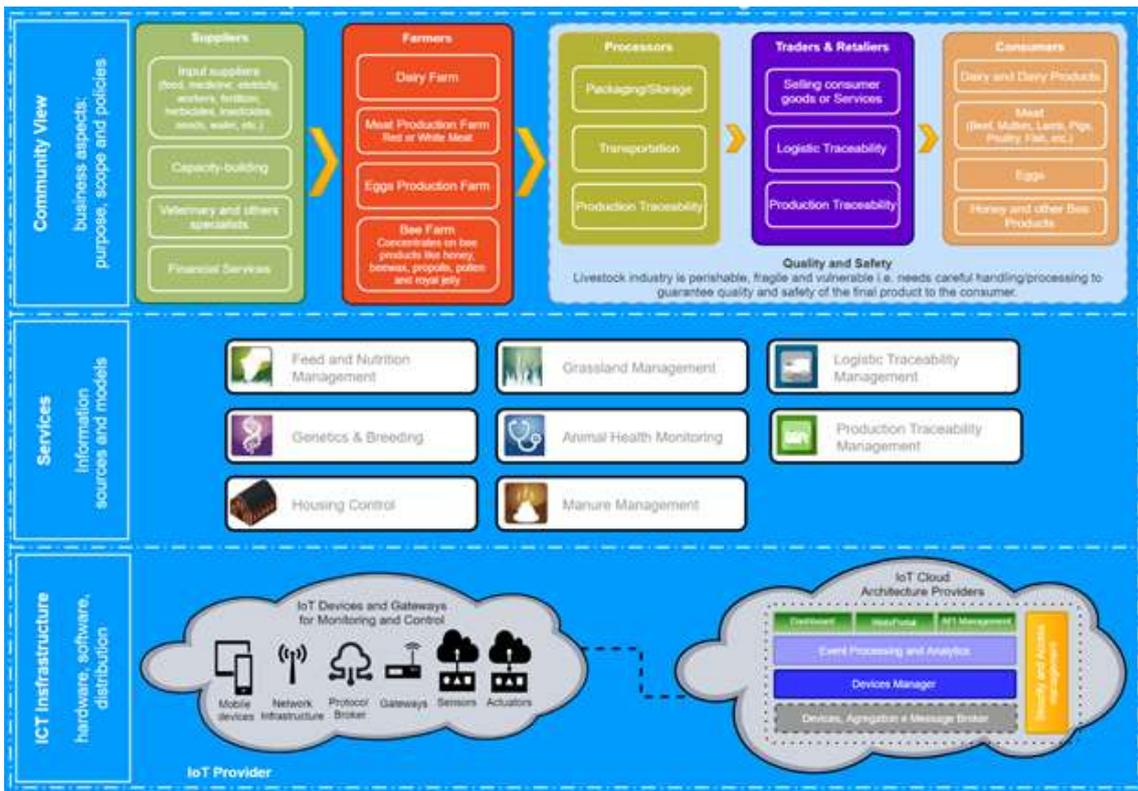


그림 9 - 스마트 축산 3계층 개념모델 (ITU-T Y.IOT-SLF)

최상위 계층에는 주요 이해 관계자를 나타내는 커뮤니티가 있다. 이 커뮤니티에는 공급 업체, 농민, 기업가, 상인과 소비자들이 포함되며, 각각에 대한 주요 역할에 대한 설명이 제시된다. 중간 계층에는 커뮤니티 운영을 지원하는 서비스들로 구성된다. 이들 서비스는 각기 다른 프로세스를 가진 여러 하위 서비스들을 포함하고 있으며 하나 이상의 이해관계자가 사용하기 위한 결과를 얻기 위해 서로 연결되어 있다. 하위 계층에는 IoT 엔진의 기능을 가능하게 하는 기술 장치 및 인프라로 구성된다.

또한 Y.IOT-SLF에서는 IoT 기반의 SLF 서비스 참조 모델을 제시하는데, 이는 ITU-T Y.2238과 유사하게 스마트축산을 다음과 같이 3단계로 구분하여 정의하고 있다.

- 생산 이전 단계는 생산을 위한 계획을 하는 단계이다. 이 단계에서는 생산 자체가 시작되기 전의 초기 단계로서, 이 단계에서는 자원 수요에 따라 다양한 자원(인력, 자재 및 생산 능력)을 적절하게 할당하는 과정이다.
- 생산 단계는 축산물을 얻기 위해 가축을 사육하고 가공하는 단계이다. 이 단계에는 가축의 번식, 사료 급이, 질병 관리, 가축 부산물 획득 및 도축 등에 관련된 과정들이 포함된다.
- 마지막으로 생산 이후 단계는 축산물을 유통하기 위한 단계이다. 이 단계에는 축산물의 포장, 유통 및 안전먹거리를 위한 생산자 추적 등의 과정들이 포함된다.

#### 나-4) IoT 기반 스마트 온실 서비스 프레임워크 표준 소개 (한국통신학회, '20)

최근 IoT(Internet of Things) 기술의 비약적인 발전과 함께 농업 분야에 IoT 기술을 접목한 스마트팜 (smartfarm) 기술개발 및 표준화를 위한 노력이 본격화 되고 있다. ITU-T Study Group(SG) 13에서는 2012년부터 스마트팜에 대한 기준모델, 서비스 및 네트워크 성능에 대한 표준화를 진행하여 2015년 Y.4450을 개발한 바 있다. 또한 스마트팜 생산 전 단계의 서비스 모델을 규정하고 있는 Y.2244, 네트워크 기반의 스마트팜 질병관리 서비스 모델을 다루는 Y.2243 등 지속적인 표준화 연구가 진행되었다.

이후 IoT 및 스마트 시티에 대한 표준화 중요성에 강조됨에 따라 2016년 설립된 SG20에서는 온실과 같은 시설원에 분야에 IoT 기능을 적용하기 위한 표준인 Y.4466이 2020년 1월 개발이 완료되었으며 스마트축산 분야에 대한 서비스 모델 표준인 Y.SLF(Smart Livestock Farming based on Internet of Things) 개발이 진행 중이다.

##### 1) 스마트온실 및 스마트온실 서비스의 개념

전통적인 온실은 수동적인 환경조건 제어, 환경상태 모니터링을 위한 지속적인 인력투입 및 경험부족에 따른 시행착오 등의 단점을 가지고 있으며, 이를 극복하기 위하여 센서, 구동기와 같은 IoT 디바이스, 제어기 및 관리시스템 등이 접목된 스마트온실이 도입되고 있다. 스마트온실은 작물의 생육환경 상태와 작물 재배 상태를 관찰하고 환경조건을 자동적이고 지능적으로 제어함으로써 작물재배를 위한 시공간적인 자유와 인건비의 절감, 에너지 등의 리소스 소비를 최소화할 수 있도록 한 것으로, 최적 생육 모델 기반의 작물 생육조건 설정, 센서 디바이스를 이용한 환경 상태 및 생육 상태 감시/제어, 구동기를 통한 생육 환경 제어 등의 기능을 갖는다.

또한 스마트온실 서비스는 정밀농업을 수행하는 스마트온실, 다양한 환경조건 및 작물 재배조건에 따른 관리 전략을 제공하는 스마트온실 서비스 이용자 및 이를 위한 다양한 서비스를 제공하는 스마트온실 서비스 제공자로 구성되며, 다음과 같은 기능을 통해 생산성의 극대화 및 이용자 편의를 제공을 도모한다.

- 센서를 통해 수집된 환경정보 및 생육정보의 상관성 분석
- 특정 작물에 대한 최적 생육 모델 도출 및 공유
- 최적 생육 모델을 통한 생산성과 품질 개선
- 이용자 경험 공유를 통한 시행착오 감소
- 자원 관리 전략 및 마케팅 전략 수립

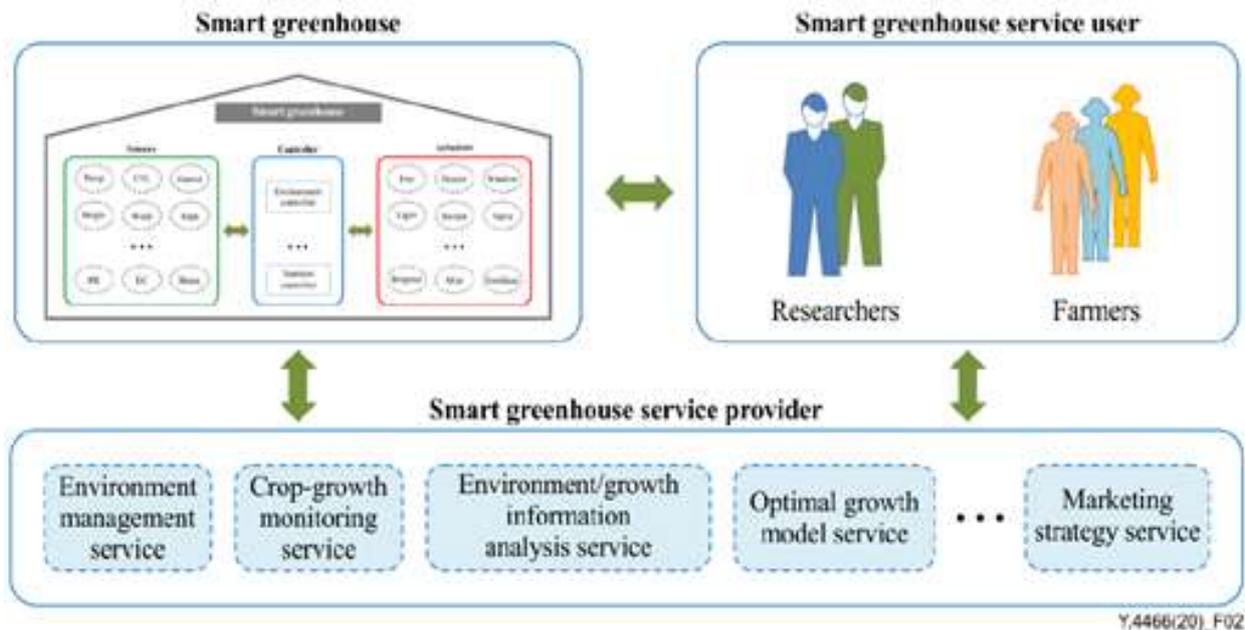


그림 10 - 스마트온실 서비스의 개념도

##### 2) 스마트온실 서비스 요구조건

스마트온실은 최적 생육 조건 설정과 생산성, 품질 개선을 위해 다양한 센서와 구동기를 필요로 하며 환경조건 및 생육조건 감시기능, 이용자와 서비스 제공자에 대한 환경 및 생육상태 정보의 실시간/비실시간

제공 및 보고기능, 환경조건 제어기능 및 제어를 위한 센서 및 구동기 동시처리기능, 이용자 요구에 따른 특정 구동기 조작기능, 제어명령에 대한 실행결과 보고기능, 작물 재배 활동 기록기능 및 환경생태 이력데이터 유지기능이 요구된다. 또한 필요에 따라서 감시와 조절이 필요한 생육조건과 환경조건 특정기능과 서비스 제공자의 지시에 따른 특정 구동기 조작기능, 서비스 제공자에 대한 작물 재배정보 제공기능과 스마트온실의 원활한 운용을 위해 센서 및 구동기, 제어기 등의 디바이스 상태를 이용자나 사업자에게 제공할 수 있는 기능 또한 요구된다.

아울러 스마트온실 서비스 제공자는 최적 생육 모델 도출을 위한 특정 스마트온실의 실시간/비실시간적인 환경 및 생육상태 정보 수집기능, 수집된 정보 기반의 최적 생육 모델 수립 및 제공기능, 이용자의 수익성 극대화를 위한 시장정보(예측 시장가격 등) 제공기능이 요구되며, 필요에 따라서는 스마트온실 관리 전략 수립을 위한 이용자의 리소스(전력, 물, 비료 등) 투입 정보나 생산 정보를 수집할 수 있는 기능을 가져야 한다. 스마트온실 이용자는 스마트온실의 자동적인 운용을 위해 서비스 제공자로부터의 최적 생육 모델 중 하나를 선택하고 이를 적용할 수 있으며, 서비스 제공자와의 통신장애 상황에 대비하여 수동적으로 스마트온실을 조작할 수 있어야 한다. 또한 이용자의 리소스 투입정보와 생산정보를 서비스 제공자에게 제공할 수 기능을 갖추어야 한다.

### 3) 스마트온실 서비스 참조모델 및 각 기능의 역할과 인터페이스

스마트온실 서비스 참조모델은 기존의 IoT 참조모델(Y.4000)을 기반으로 하고 있으며 다음과 같은 6가지 기능을 추가하였다.

- 센싱기능(sensing function) : 환경 및 생육 상태정보 수집 및 제어기능으로의 전달, 각 센서에 대한 정보(타입, 위치, 및 센싱값의 단위 등)를 관리기능으로 전달
- 구동기능(actuating function) : 제어기능으로부터의 제어명령에 따른 구동기 동작, 각 구동기에 대한 정보(타입, 위치, 운용상태 등)를 관리기능으로 전달
- 제어기능(control function) : 운용기능으로부터의 생육 모델 정보 또는 구동기의 운용상태 정보에 따른 구동기 제어명령의 생성, 제어명령을 구동기능으로 전달, 센싱기능 및 구동기능으로부터의 환경 및 생육상태 정보를 운용기능으로 전달
- 운용기능(operating function) : 이용자의 경작 경험에 기반을 둔 특정 작물에 대한 최종 환경제어 조건의 생성, 제어기능으로부터의 환경 및 생육상태 정보를 통합기능으로 전달
- 통합기능(integration function) : 운용기능으로부터의 스마트팜(스마트온실들로 구성된) 환경 및 생육상태 정보의 축적과 최적 생육 모델 도출, 도출된 최적 생육 모델을 운용기능으로 전달
- 관리기능(management function) : 환경 및 생육상태 정보의 관리, 정보전달을 위한 제어기능, 운용기능 및 통합기능과의 상호작용

이와 같은 각 기능들의 역할에 따라 스마트온실 서비스는 크게 각 스마트온실의 환경 및 생육상태 정보의 전달과 스마트온실에 대한 제어명령 전달의 두 가지 형태의 정보 흐름을 제공한다.

- 환경 및 생육상태 정보의 생성과 소비 : 센싱기능이 생성한 환경 및 생육상태 정보는 제어기능, 운용기능, 통합기능 및 이용자에게 전달
- 생육조건 생성과 제어명령의 수행 : 바로 이전의 정보흐름에 따른 정보는 각 작물에 대한 최적의 생육조건 도출을 위해 활용되며, 통합기능은 이러한 최적 생육 조건을 유지하고 관리하며 운용기능에 이를 전달, 운용기능은 전달된 최적 생육 조건을 이용자의 요구에 따라 수정하고 이를 제어기능에 전달, 제어기능은 전달된 정보에 따른 적절한 제어명령을 구동기에 전달

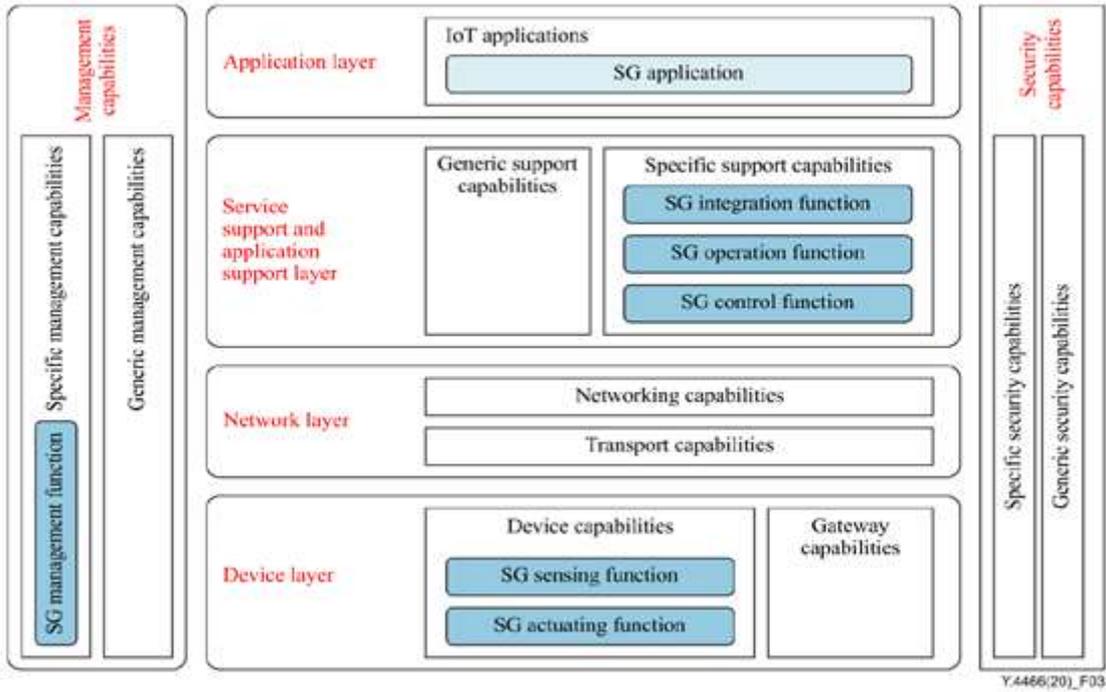


그림 11 - IoT 참조모델 기반 스마트온실 서비스 참조모델

이와 같은 일련의 기능을 효율적으로 수행하기 위해 다음 그림에서와 같이 스마트온실 서비스 제공자와 사용자간 16개의 인터페이스를 정의한다.

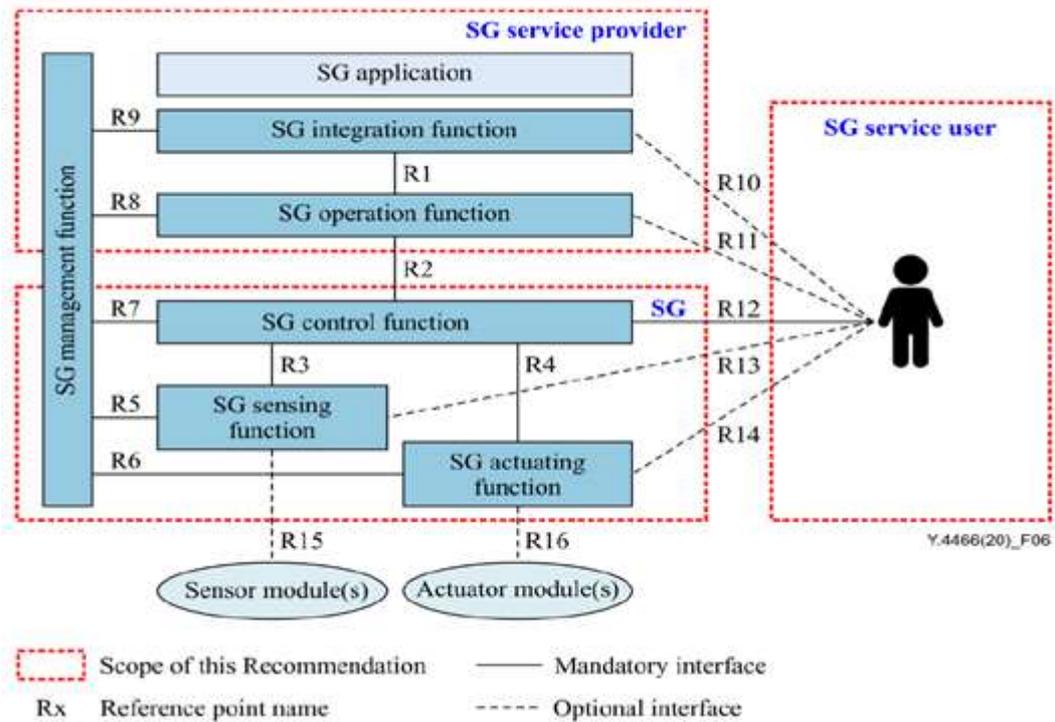


그림 12 - 각 기능간 인터페이스 포인트

최근 스마트온실과 같은 스마트팜의 중요성이 날로 증가하고 있으며 국내외에서는 관련 기술시장을 선점하기 위한 노력을 기울이고 있으나 이를 위한 표준의 부재로 제조사별 상이한 규격을 갖는 제품들이 개발되고 있어 상호호환성과 연동에 많은 문제점들이 드러나고 있다. 특히 클라우드(cloud) 기반의 스마트팜 관제의 필요성이 대두되면서 단독형 스마트온실은 물론 클라우드 기반의 스마트온실 관제 서비스에 대한 필요성이 높아지고 있다.

이에 IoT를 접목한 스마트온실 서비스에 대한 프레임워크를 다루는 ITU-T Y.4466 표준의 스마트온실 서비스 제공을 위한 구체적인 기능과 요구사항, 각각의 기능 개체간 상호 인터페이스를 소개함으로써 이중 형태를 갖는 스마트온실 서비스간 상호연동 가능성을 확대하고 향후 국내 시장은 물론 해외 시장에서의 경쟁력 향상을 위하여 활용될 것으로 기대한다.

#### 나-5) Implementation of Smart Farm Devices using Open Source Software (ICACT2021, '21)

사물인터넷(IoT)이 차세대 ICT산업의 핵심으로 주목 받으며 다양한 영역에 적용하려는 시도가 확대되고 있는 가운데 가장 주목받고 있는 분야 중 하나가 스마트팜이다. 스마트팜은 기존의 농업에 IoT 등 ICT 기술을 융합하여 생산성과 효율성을 제고함으로써 고부가가치 창출을 추구하는 첨단 농업 기술로서, 다양한 IT기기들이 활용된다.

스마트팜의 일반적인 구성은 생육 및 환경 정보를 모니터링 하기 위한 센서 장치(온도, 습도, CO<sub>2</sub>, 토양, 양분 등), 스마트팜 시설을 제어하는 구동기(창개폐기, 환기, 난방, 관수, 관비 등), 각 시설에 부착된 장치들을 제어하는 제어기, 그리고 이들을 통합적으로 관리하는 운영관리시스템 등으로 구성된다. 센서 및 구동기는 센서 및 구동기 장치와 이들과 연동되어 제어기를 통해 운영시스템까지 연결되는 센서 및 구동기 노드로 세분될 수 있다. 운영관리시스템은 인터넷을 통해 클라우드와 연결되어 원격에서 스마트팜 관리도 가능한 수준으로 진화하고 있다.

현재 보급되고 있는 스마트팜 장비는 생산업체마다 제품규격이 다르기 때문에 상호 호환성이 보장되지 않는다. 이는 농가의 입장에서는 다양한 기자재를 구비하는데 문제가 된다. 즉, 특정 기능을 가진 다른 생산업체의 장비를 새로이 도입하고 싶어도 기존 장비와 호환되지 않기 때문에 할 수가 없다. 또한 생산업체의 입장에서도 호환성이 없는 경우 신규 구축이 아닌 이상 자사의 장비를 적용할 수 있는 여건이 안되기 때문에 자연스럽게 판매의 입지가 좁아지게 된다. 따라서 스마트팜 기자재의 상호호환성 확보는 사용자 및 제조사에 매우 중요한 요구사항이 된다.

OCF (Open Connectivity Foundation)는 IoT관련 국제 표준화 기구로서 전세계의 많은 IoT기업들이 참여하고 있고, 표준개발과 동시에 오픈 소스 프로젝트도 동시에 진행함으로써, 규모뿐만 아니라 생태계 확장에서도 타 표준화 기구에 비해 비교우위에 있다.

OCF는 다양한 IoT 유/무선 연결 기술을 활용하여, 상호 연동성을 이용하여 구축하고 있으며, 스마트홈, 자동차, 물류, 헬스케어 등 다양한 IoT 서비스를 개발할 수 있도록 구성되어 있다. OCF는 IoT를 구현하는 REST 구조 기반으로, 경량의 프로토콜(coap)로 디바이스들을 연결하고, 서로 연동 시킬수 있다. 특히, OCF의 가장 큰 장점은 무상특허정책으로, OCF 회원으로 가입하면 회원사간 OCF 관련 기술들의 모든 소유 특허는 무상으로 사용 가능하다.

OCF는 리소스 모델에 의해 클라이언트-서버 모델로 동작하게 되는데, 리소스를 제공하는 장치는 서버로, 리소스를 요청하는 장치는 클라이언트로 동작한다. 스마트팜 환경에 적용되는 일례로서, 센서 및 구동기 등의 스마트팜 기자재는 서버로서, 사용자 인터페이스를 통해 스마트팜 시설을 제어할 수 있는 운영관리시스템은 클라이언트로서의 기능이 탑재될 수 있다. 이는 OCF를 지원하는 모든 기기들간 상호 호환성에 따라 스마트폰, 태블릿 등을 포함하여 TV 등과 같은 가정내 가전들까지도 연결될 수 있음을 의미한다. 마찬가지로, 스마트팜에 OCF가 적용되면, 기존에는 호환되지 않던 스마트팜 장치들이 서로 연결되어 농가의 생산성 향상에 큰 기여를 할 수 있게 된다.

본 연구에서는 OCF 기반 스마트팜 데이터 모델 표준 기술을 분석하고, OCF가 제공하는 오픈소스 소프트웨어인 IoTivity[4]를 활용한 테스트베드를 구현함으로써, IoT 장치간 상호 호환성을 담보할 수 있는 표준기술의 스마트팜 적용을 위한 개념검증을 수행하였다.

나-6) 스마트축산 표준화 동향 (JCCI2021, '21)

기존 농업 기술에 ICT 기술을 접목하여 생산성과 편의성을 크게 높일 수 스마트팜 기술에 대한 관심이 높아짐에 따라, 최근에는 그 적용 분야를 시설원에 뿐만 아니라, 노지, 축산 및 수산양식으로까지 확대를 꾀하고 있다. 본 논문에서는 스마트팜 기술의 한 분야인 스마트축산 분야에 관한 국내외 기술 동향과 표준화 기술을 소개하고 기술 발전의 방향성을 제시하기 위한 목적으로 작성하였다.

현재 국내외 국제 표준화 기구에서 스마트축산과 관련한 표준화 활동이 진행되고 있다. 국내표준화는 포럼과 KS 표준화가 진행되고 있으며, 국제표준화는 ISO와 ITU-T에서 진행되고 있다. 스마트축사와 관련한 국내 단체표준화 활동은 한국정보통신기술협회(TTA)에서 추진되고 있다. TTA (PG426)에서는 2016년부터 스마트축사를 위한 용어, 축사용 센서의 인터페이스 및 데이터 구조, 축산 분야 ICT융복합 장비의 서비스 요구사항 등에 대한 표준을 개발하고 있는데, 이중 스마트축사를 위한 센서 인터페이스와 축산 분야 ICT융복합 장비의 서비스 요구사항 등은 한국산업표준(KS)으로 제정하는 중이다. 스마트축사를 위한 센서 인터페이스에서는 축사에서 사용되고 있는 다양한 내외부 센서들의 운용환경을 제시하고 각각의 센서의 측정 범위와 허용 오차를 정의하고 있다.

국제표준화는 ISO의 ISO 65 시리즈(Agriculture)[3]와 67 시리즈(Food Technology)[4] 표준화 작업을 통하여 농장 구조물, 농업기기, 양봉, 가공식품 등 농업 전반적인 분야에 대한 기술표준을 제시하고 있으며, ITU-T의 SG20을 통해 축산과 ICT 기술이 융합된 스마트축산분야 기술표준이 진행되고 있다. 특히 한국전자통신연구원과 브라질이 공동으로 개발하고 있는 ITU-T Y.IoT-SLF “Framework and capabilities for smart livestock farming based on Internet of things” 표준초안에서는 IoT 기술과 연계한 스마트축산 산업군에 대한 소개와 함께 서비스 요구사항이 정의되고 있다.

ITU-T Y.IoT-SLF는 IoT 기반의 스마트축산에 대한 이해를 돕고자 축산 분야의 생산, 가공, 유통에 이르는 전단계에 걸친 이해 당사자 관점에서 문서를 기술하고 있다. 이 표준은 향후 세부 기능에 대한 기능 표준을 개발하기 위한 개념 표준으로서의 가치를 갖고 있다. 이 프레임워크에서는 스마트 축산의 3 계층 개념 모델을 제시한다. 가장 상위 계층에는 주요 이해 관계자가 위치하는데, 주요 이해 관계자들로는 공급 업체, 농부, 가공 업체, 상인 및 소매 업체 및 소비자 등이 있으며, 이들의 주요 활동 및 의무에 대한 설명이 제공된다. 두번째 계층에는 이들 이해당사자들에게 제공되는 서비스들이 기술된다. 각 서비스 클래스에는 하나 이상의 이해 관계자의 사용에 대한 결과를 얻기 위해 서로 연결된 프로세스가 서로 다른 여러 하위 서비스가 정의된다. 본 표준에서 IoT 기술과 접목 가능한 스마트 축산 서비스로는 식이 관리 (사료 및 영양 관리), 종자 및 사육 (유전학과 번식), 축사환경관리, 녹지 관리, 동물 건강 감시, 비료 관리(분뇨 관리), 물류 추적 관리, 제품 추적 관리 서비스들을 제시하고 있다. 맨 마지막인 세번째 계층에는 IoT, 클라우드와 같이 스마트축산의 모든 활동을 관제하기 위한 기술 장치와 인프라로써, IoT와 클라우드 기술을 포함한다.

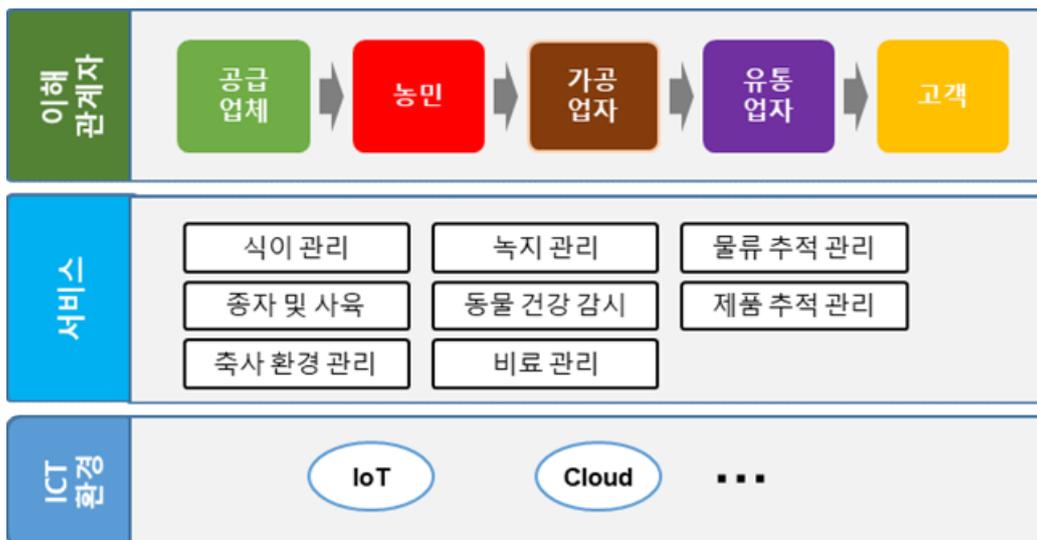


그림 13 - 스마트축산 구조 (ITU-T Y.IoT-SLF)

나-7) 기상 IoT 센서 데이터를 활용한 서리 예측 연구 (한국인공지능학술대회, '21)

최근 지구온난화 현상으로 인해 작물의 개화시기가 앞당겨지고 있고 이때 발생하는 서리는 작물 생육에 큰 영향을 미친다. 서리가 발생하는 시기를 정확히 예측할 수 있다면 농민들은 서리를 대비할 수 있고, 서리 발생으로부터 오는 손실을 막을 수 있다. 따라서, 봄, 가을철 서리로 인한 농가 피해를 최소화하기 위해 정확한 서리 발생 예측이 필요하다. 선행 연구에서는 서리 예측의 정확성을 높이기 위한 서리 발생일의 기온 위주 기상 특성을 제시하였다. 서리는 일반적으로 날씨가 맑으면서 춥고 바람이 잘 불지 않을 때 발생한다. 서리 발생 예측을 위해 도출된 기상인자는 최저기온, 평균풍속, 상대습도, 이슬점 온도이다. 서리가 발생한 날이 서리가 발생하지 않은 날에 비해 최저기온, 이슬점 온도, 평균풍속은 낮게 나타나고 상대습도는 높게 나타난다.

이 논문에서는 농업 기상 IoT 데이터를 전제로 머신러닝 알고리즘을 사용하여 서리 발생을 예측한다. 그림 14는 본 논문에서 제시한 서리 예측을 위한 전체적인 데이터 분석 단계를 나타낸다. 특징 선정 단계에서는 랜덤 포레스트 기반의 특징 중요도를 활용하여 서리 발생에 주요한 영향을 주는 변수들을 결정한다. 특징 중요도를 계산하기 위하여 랜덤 포레스트 분류기 기반의 MDI(Mean Decrease in Impurity) 모델을 활용한다. 분류 단계에서는 서리 예측 모델의 구축을 위해 분류기 알고리즘을 구현하며 최적화 및 두 가지 성능지표를 활용하여 분류기 성능을 확인한다. 이 논문에서는 2020년 1월 1일부터 5월 31일까지 경상북도 영천시 화북면의 IoT 기상 관측 장비에서 수집한 기온, 상대습도, 강수량, 초상 온도 등의 센서 데이터를 활용한다. 관측 기간 152일 중 서리가 관측된 날은 29일이고, 서리 현상이 발생하는 새벽 시간대(01:00 ~ 06:00)의 데이터를 분류 모델 학습에 사용한다.

머신러닝 모델을 학습시키기 위해서 설정한 시간대의 데이터를 요약하여 대표할 수 있는 새로운 데이터 세트를 산술 평균, 최대/최솟값, 분산 등의 기술 통계량을 활용하여 생성한다. 또한, 기존 연구를 바탕으로 서리 발생에 영향을 주는 변수로서 기온역전층과 이슬점을 추가 정의한다.

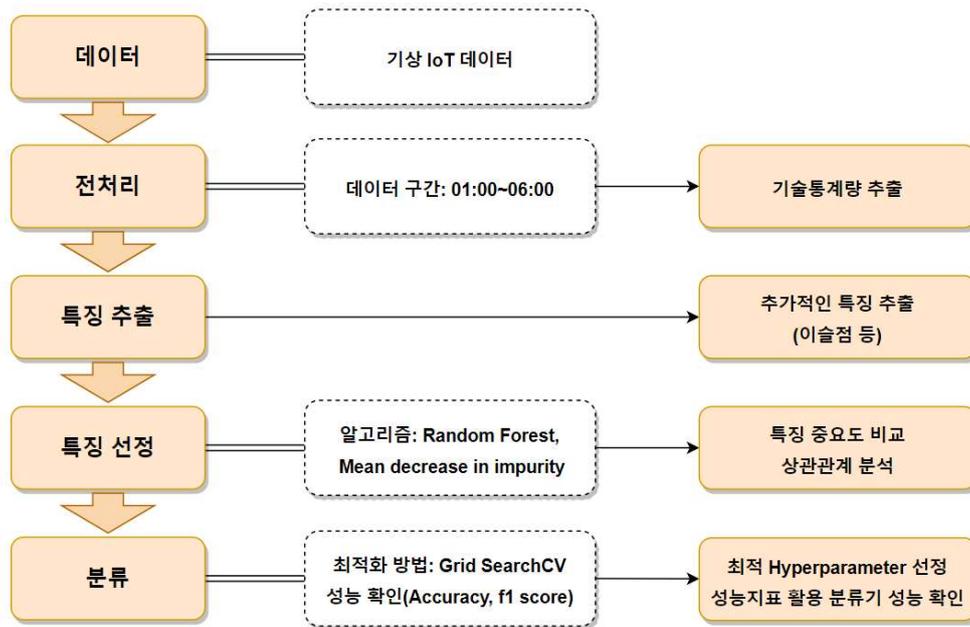


그림 14 - 데이터 분석단계

본 연구에서는 MDI 알고리즘을 이용하여 각 특징의 중요도를 계산한 뒤, 높은 중요도의 특징들을 활용하여 모델을 학습한다. MDI 중요도는 트리형 알고리즘에서 각 데이터가 split될 때 불순도의 감소분에 따라 각 특징의 중요도를 계산한다.

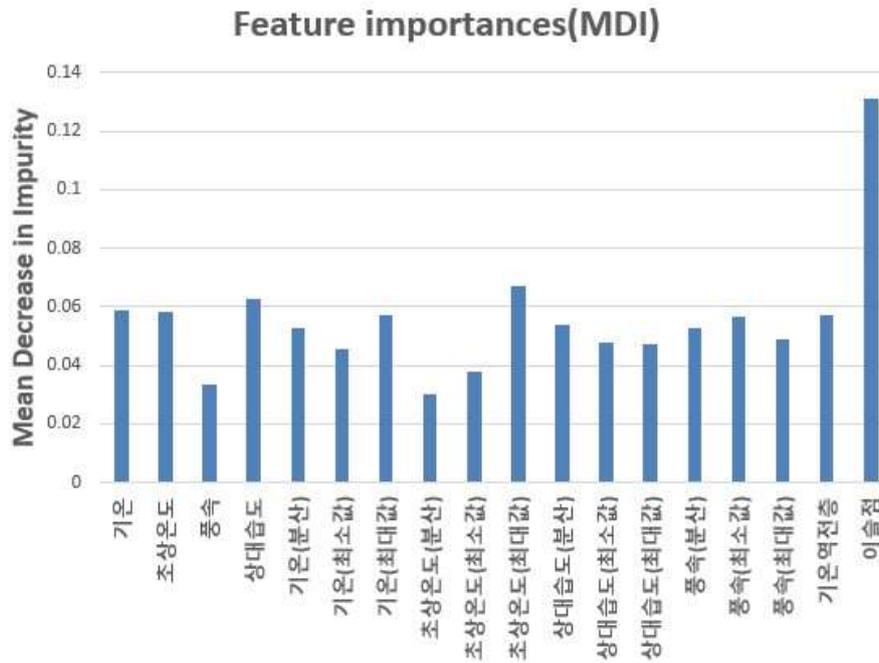


그림 15 - 특징 중요도(MDI)

그림 15는 각 특징의 MDI 중요도를 나타낸다. 추가한 기온역전층과 이슬점이 가장 높은 중요도를 가졌고, 상대습도의 최솟값과 기온의 분산 값 또한 높은 중요도를 가진다. 이 결과에 따라 기온역전층과 이슬점, 상대습도의 최솟값, 그리고 기온의 분산 값을 활용하여 모델을 학습한다.

이 연구에서는 서리 발생 예측에 필요한 중요 인자들을 도출해보고, 랜덤 포레스트 모델 최적화를 통해 서리의 발생을 예측해보았다. 전체 데이터 세트의 숫자가 크지 않았고 편향적인 데이터 세트를 활용하였지만 적절한 특징 데이터를 도출하고, 최적의 초매개변수를 설정함으로써 서리의 발생을 정확하게 예측할 수 있는 가능성을 실험을 통해 확인되었다.

## 나-8) Understanding IoT Climate Data based Predictive Model for Outdoor Smart Farm (ICTC2021, '21)

사물인터넷(IoT) 데이터 분석의 발전은 현재의 지식 발견 패러다임을 과학 영역에서 농업 영역으로 변화시키고 있으며, 최근 IoT 솔루션은 농업 분야에 새로운 지식 통찰력을 제공하고 다양한 농업 응용 프로그램을 지원하여 스마트 농업을 가능하게 하고 있다. 예를 들어 IoT 기반 농업 데이터는 서리 피해 방지, 관개 및 해충 방제와 같은 시간에 민감한 농업 서비스와 관련된 정확한 예측을 가능하게 한다. 예측 서비스, 인간 경험에 의존하는 전통 농업은 데이터 중심의 정밀 농업으로 전환할 수 있다. 이 논문에서는 IoT 기 후 기반 예측 응용 프로그램이 농업 영역에 대한 자동 의사 결정 지원 시스템을 만드는 데 도움이 되어 적절한 시간에 관련 액추에이터를 작동할 수 있음을 시사한다.

IoT 기상 관측소는 다양한 종류의 온도(3m, 1.5m, 0.3m, 0.1m 미만), 일사량, 강수량, 습도, 풍속, 풍향 등의 센서로 구성되어 있으며, 실시간 기후 정보를 수집하여 전달한다. 센서 데이터를 1분마다 수집하여 원격 데이터 서버에 시간당 평균 형식으로 요약 KMA(한국 기상청)에서 발표한 지역 기상 정보를 서버에 저장하여 해충 예측 활용 표 1은 IoT에 포함된 센서 정보 과수원의 농작물 주변에 설치된 기상 관측소 분당 수집된 데이터는 서버로 전송되어 다양한 예측 모듈 및 모니터링 서비스를 위해 서버의 데이터베이스에 저장 모든 센서로부터 수집된 기후 정보는 분 단위로 저장 시간 기반 기후 데이터는 특히 주요 기상 변수 발굴에 필수적인 예측모델 구성요소는 기후예측모델, 서리모델, 살충제 모드로 구성되다.

이 논문에서는 미기상 정보를 이용하여, 서리와 해충에 대한 예측 모델을 다음과 같이 설계하였다.

### ■ A. 서리 예측 모델

"서리"는 결빙 이슬이 있거나 증기에서 얼음으로 상이 변화하는 표면의 얼음 결정 형태를 의미한다. 서리는 특히 개화 기간 동안 꽃을 손상시켜 심각한 작물 실패를 초래할 수 있다. 복사 손실은 열 역전과 고기압 중심이 우세한 맑고 바람이 없는 밤에 발생한다. 서리가 농업에 미치는 상당한 영향으로 인해 이 현상으로 인한 피해를 해결하는 데 많은 과학적 연구가 집중되었다. 일부에서는 적절한 예방 조치를 취하면 서리 피해를 줄일 수 있으므로 서리에 유리한 조건을 발견하면 피해를 최소화하는 효과를 얻을 수 있다고 언급하였다.

### ■ B. 해충 예측 모델

곤충, 식물 병원체 및 잡초 해충은 매년 모든 잠재적 식량 생산의 40% 이상을 파괴한다. 해외 연구진들은 날씨, 작물 및 해충의 상호 관계에 주목했는데, 작물 해충도 습도에 민감하다는 결과를 얻었다. 따라서 해충예측모델은 IoT 기후데이터와 지역기상정보를 읽어 감염확률 및 발생일자를 산출하고, 예측모델에서 산출된 감염예측일은 설치현장의 ID별로 데이터베이스에 저장된다. 해충 침입이 계산되면 서버는 가입한 농부에게 알림 메시지를 보냄으로써 효과적으로 해충 방제가 가능할 것으로 기대된다.

## 6) 기타 연구활용

### 가) 스마트 농업 관련 국내 대학 교재 공저

#### 가-1) [도서저술] 「스마트팜 개론」(구미출판사) 공저 ('20)

해당연도에는 ARC 스마트팜 전문가들과 협업을 통해, 국내 대학교나 대학원에서 수업 교재로 활용이 가능한 스마트팜 개론을 저술하였다. 본 저서에서는 표준화의 개요, 스마트팜 관련 국내외 표준화 동향 내용을 수록하였다. 본 보고서에서는 본 저서의 주요 부분을 발췌하여 정리하도록 한다.

##### (가) 표준화의 개요

표준이란 편의성, 효율성 그리고 안전성을 위하여 사람 간 약속으로 통용되는 것을 의미한다. 국제표준화기구인 ISO에서는 표준을 "공통적이고 반복적인 사용을 위하여 제시된 규칙이나 지침 또는 제품의 특성이나 관련 공정 및 생산방법을 규정한 합의된 문서"라고 정의하고 있다. 즉 '표준(Standard)'이란, 합의에 따라 작성되고, 인정된 기관에 의해 승인되며, 공통적이고 반복적인 사용을 위해 제공되는 규칙, 가이드 또는 특성을 제공하는 문서를 의미한다. '표준화(Standardization)'란 표준을 만들기 위한 일련의 활동을 의미한다.

##### (나) 표준의 분류

표준은 표준의 제정 목적과 방법, 준수 대상과 의무 등 다양한 방법으로 분류되곤 한다. 그중 널리 사용되는 분류 방법은 다음 표에서처럼 사내표준, 단체표준, 국가표준, 지역표준, 국제표준 등과 같이 표준을 제정하는 기구에 따라 분류하는 방법이다.

표 1 - 준 제정 기구에 따른 표준의 분류

표준의 종류	내용	예
사내표준	<ul style="list-style-type: none"> <li>기업 내부, 공장 내부에서 적용되는 표준으로 내부 관계자의 합의를 통해서 제정되는 표준</li> </ul>	<ul style="list-style-type: none"> <li>ETRI 오픈소스 가이드라인 등</li> </ul>
단체표준	<ul style="list-style-type: none"> <li>업계, 단체, 학회 등의 내부 구성원 사이에서 적용되는 표준을 관계자의 합의를 통해 제정하는 표준</li> </ul>	<ul style="list-style-type: none"> <li>한국정보통신기술협회(ITA)의 정보통신단체표준(TTAS), ITS 표준총회에서 제정하는 ITSK 단체표준 등</li> </ul>
국가표준	<ul style="list-style-type: none"> <li>국가표준을 제정하는 기관이 한 나라의 영토 내에서 적용되는 표준을 이해관계자의 합의를 얻어 제정하는 표준</li> </ul>	<ul style="list-style-type: none"> <li>한국산업규격(KS), 일본산업규격(JIS), BS, ANSI, DIN 등</li> </ul>
지역표준	<ul style="list-style-type: none"> <li>복수의 국가 또는 지역 단체에 의해서 공통의 이익을 위해 일정한 지역 내에서 적용되는 표준</li> </ul>	<ul style="list-style-type: none"> <li>유럽 범 지역 표준기구인 CEN이 제정하는 CENELEC, 아시아태평양기구인 ASTAP 등</li> </ul>
국제표준	<ul style="list-style-type: none"> <li>다수의 국가가 각국의 이해를 회의 형식으로 조정하여 국제적으로 적용되는 표준</li> </ul>	<ul style="list-style-type: none"> <li>ISO, IEC, ITU 등</li> </ul>
대표적인 기구	<ul style="list-style-type: none"> <li>IEEE, IETF, W3C</li> </ul>	<ul style="list-style-type: none"> <li>ISO, IEC, ITU</li> </ul>

참고로 선진국의 경우는 하급 표준, 즉 사내표준이나 단체표준을 모태로 하여 상위 표준인 국가표준, 지역표준에 적용되는 것이 일반적인 데 반하여, 후진국의 경우는 상위표준을 규범으로 하여 하위표준으로 확대·발전되는 양상을 띠고 있다.

##### (다) 사실표준 및 공적표준

앞에서 소개한 표준의 분류 방법에 더하여, 표준을 제정하는 기구가 어떠한 성격을 갖는지에 따라 분류하기도 한다. 표준은 일반적으로 공인된 표준화 기관이나 생산자협회 혹은 표준 관련 학술·전문협회에서 기술적·전문적 사항을 충분히 검토한 후 이해관계인의 합의에 따라 제정 혹은 개정되는데 이를 공적표준(de Jure)이라고 한다. 이와는 달리 시장 내에서 기술적으로 우위에 있거나 혹은 시장 지배력이 큰 기업이 사용하고 있는 기술명세가 동 시장에서의 사실상의 표준으로 활용되는 예도 있는데 이를 사실표준(de Facto)이라고 한다. 표준을 사실표준과 공적표준으로 구분하는 것은 표준의 주체가 누구(시장 혹은 공적) 인지를 쉽게 파악할 수 있게 해준다는 장점도 있지만, 그보다는 사실표준인지 공적 표준인지에 따라 관련 표준화 전략 수립 방법이 달라지기 때문이다. 다음 표에서 비교한 바와 같이 일반적으로 공적표준은 책정 과정이 투명하고 표준내용이 명확하고 개방적이며 원칙적으로 단일표준을 제공하며 회원가입이 비교적 개방적이나, 표준개발 속도가 느리고 표준의 보급과 제품보급의 시간 격차(time lag)가 발생하며 기술의 무임승차(Free Rider)가 발생한다는 약점이 있을 수 있다. 이에 반해 사실표준은 책정 속도가 빠르고 표준과

제품의 보급이 동시에 이루어질 수 있어 표준화 주체자가 시장을 독점할 수 있다는 장점이 있으나, 정보의 공개가 불확실하고 회원가입이 폐쇄적이며 개정 절차가 불투명하다는 약점이 있을 수 있다.

표 133 - 사실표준과 공적표준의 차이

	사실표준	공적 표준
표준의 결정자	• 시장 메커니즘	• 표준화 기관
표준의 정통성	• 사용자에게 의해서 결정	• 표준화 기관의 권위에 의해서 결정
표준화의 동기	• 불편하기 때문	• 제품은 기능을 발휘
표준화의 관건	• 시장 도입기의 점유율(share)-유력 기업의 참여	• 표준화 기관의 강제력, 관련 기업 등 이해관계자의 참여
표준화와 제품의 순서	• 사업화에서 표준결정으로 이동	• 표준결정에서 제품화
대표적인 기구	• IEEE, IETF, W3C	• ISO, IEC, ITU

(라) 표준과 기술규정

많은 사람이 때때로 표준을 강제성이 수반되는 기술규정으로 오해하곤 한다. 본 절에서는 표준과 기술규정에 대한 차이점을 설명하고자 한다. 표준은 “제시된 상황에서 최적의 질서를 달성하기 위해 일반적이고 반복적인 용도의 규칙, 지침, 또는 행위와 그 결과에 대한 특성을 제공하는, 합의에 따라 개발되고 공인기구에 의해 승인된 문서”라고 정의된다. 일례로 ISO/IEC에서 발간하는 기술시방서(Technical Specification), 기술보고서(Technical Report), 가이드(guide), 공개 규격(Publicly Available Specification)과 같이 표준 문서들이 표준의 예이다.

반면에 기술규정(Technical Regulation)은 적용할 수 있는 행정적 규정을 포함하여 상품의 특성 또는 관련 공정 및 생산방법을 규정하는 문서이며, 그 준수 여부가 강제적인 문서이다. 기술규정은 안전·환경보호·소비자보호·국방 등과 같은 공공이익의 추구에 그 목적이 있으며, 대부분 국가가 이러한 공익의 보호를 위해 독자적인 기술규정을 두고 있다. 일반적으로 기술규정은 정부에 의해 시행·감독되며, 기술규정에 부합되지 않는 제품이 생산·유통될 경우 그 자체가 불법으로서 법에 따른 제재의 대상이 된다. 이 둘 간의 차이점을 아래 표에 정리하였다.

표 134 - 표준과 기술규정의 차이점

	표준 (standard)	기술규정 (regurgitation)
제정 목적	• 생산·유통의 효율성 제고	• 안전·국방 등 공공이익 추구
제정 과정	• 표준화 기구(공적표준) 혹은 시장요구(사실표준)에 따른 제정	• 정부 주도 제정 (예, 무선설비 기술기준)
준수 의무	• 자발적인 준수	• 강제적인 준수
행정조항	• 거의 포함하지 않음	• 광범위하게 포함

(마) 스마트팜 국내 표준화 동향

2017년도 한국농촌경제연구원의 조사에 따르면, 스마트팜의 도입에 따라 시설원예·과수·축산 분야에서 모두 영농 편이성과 단위 면적당 생산량이 크게 늘고 투입 노동시간 또한 크게 줄었다고 조사되었다. 하지만 기자재의 비표준화로 인한 낮은 호환성과 시공업체의 사후기술 지원 서비스 및 교육이 미흡하다는 사용상 애로사항들이 지적되었다.

시설원예·과수·축산 분야에 적용되는 스마트팜 장치들에 대한 표준 준수가 미흡한 이유는, 아직 업체들에 마땅한 표준이 제공되지 않는다는 문제점도 있지만, ICT 기자재뿐만 아니라 작물의 특성 및 경작 방법에 따라 달라지는 시설 형태 등을 고려해야 해서 시장으로 진출하기 위한 자금이나 시간적인 제약은 받는 제조사나 시설 설치 업체들은 표준화를 고려하지 않고 제품을 개발 및 보급한다는 현실적인 문제점이 매우 크다.

스마트팜 분야의 표준화는 현재의 스마트팜 기자재 간 상호호환성 제공을 위함도 있지만 미래의 스마트팜 기술에 대한 보다 적극적인 대응을 위해 매우 중요하다. 미래의 스마트팜은 단순 자동화로 인한 편의성 증진뿐만 아니라, 빅데이터나 인공지능 등의 첨단 기술들을 활용하여 사람보다 더 경작을 잘하는 데이터 기반의 영농이 주가 될 것이다. 시설 또한 현재와 같이 축창, 보일러 등과 같은 단순한 구동기 장치가 아니라 자동 수확기와 같이 더 고가의 전문화된 장치들이 사용될 것으로 기대된다. 그래서 더 복잡한 미래의 스마트팜 기술을 활용하기 위해서는 반드시 잘 정의된 표준들이 필요하다.

### 1) 스마트팜 포럼 동향

일반적으로 표준화를 위해서는 공인된 표준화 기구가 필요하지만, 때에 따라 상설 공인 표준화 기구가 아니라, 표준을 필요로 하는 당사자들이 포럼을 구성하여 표준을 만들 수도 있다. 국내에서는 이미 스마트팜 관련 표준화를 진행하기 위해 몇몇 표준화 포럼들이 구성된 바 있다. 그중에서 2014년도에 과기정통부의 지원을 받아 신설된 스마트 농업 ICT융복합표준포럼은 2015년부터 시설원예, 유통 및 축산 분과 운영을 통해 시설원예 인터페이스와 관련된 표준 등 가장 활발한 활동을 보인다. 또한 농촌진흥청에서는 스마트팜 관련 산업체들의 의견을 수렴하기 위해 시설원예 및 축산분야 ICT융합산업화 포럼을 구성하여 스마트 온실이나 축사를 위한 센서/구동기 공동규격을 개발한 바 있다. 이어 2017년에는 스마트팜 ICT융합표준화포럼을 새롭게 발족하여 시설원예 센서/구동기 공동규격들이 국가표준으로 제안된 바 있다.

### 2) TTA를 통한 스마트팜 관련 표준

스마트팜 분야에 대한 국내 단체표준을 제정하기 위한 목적으로 2014년도에 신설된 TTA 스마트 농업 PG(Project Group)는 2018년도 현재 28개의 재개정 표준과 20여 개의 신규 표준들에 대한 작업을 진행하고 있으며, 특히 앞에서 소개된 스마트팜 관련 포럼 표준을 국내 단체표준으로 승인하기도 하였다. 스마트 농업 PG는 시설원예, 축산, 유통 및 스마트팜 클라우드 등 다양한 주제의 표준을 다루고 있다.

#### 2.1) 온실 관제 시스템 표준

온실 관제 시스템 표준은 스마트팜과 관련되어 2012년에 승인된 최초의 국내 단체 표준이다. 이 표준은 온실에 정보통신기술을 적용할 때 필요한 요소를 정의하고, 그에 따른 기술적 요구사항과 구성 내용들을 정의하는 것을 목적으로 한다. 본 표준에서 정의하는 온실 관제 시스템은 온실에서 작물 성장 관리를 위해 다양한 정보를 수집하고, 이를 활용해 성장 환경을 조성하는 설비 기기의 제어가 가능한 시스템이다. 그림 1에서와 같이 크게 4 부분(센서노드, 구동기노드, 온실 통합제어기 및 온실 통합 관리 시스템)으로 구성되며, 그 서비스적 기능은 성장 환경의 모니터링 및 제어를 포함한다. 또한 이 표준은 온실의 장치들간 인터페이스에 사용되는 프로토콜을 정의하는 후속 표준들을 소개하고 있다.

온실 관제 시스템-Part 1은 센서노드와 온실통합제어기 사이에서, 센서노드의 센싱값을 모니터링하는 기능을 담고 있으며, 온실 관제 시스템-Part 2는: 제어노드(구동기노드)와 온실통합제어기 사이에서, 온실통합제어기가 구동기노드를 제어하는 기능을 기술한다. 온실 관제 시스템-Part 3은 온실통합제어기와 온실 운영 시스템 사이에서, 센서노드와 제어 노드의 정보를 받아 온실 통합 제어기에서 설정하는 기능과 구동기노드의 알고리즘을 이용하여 운영 및 제어하는 기능을 기술한다. 마지막으로 온실 관제 시스템-Part 4는 온실 운영 시스템과 빅데이터 서버와 같은 온실통합관리시스템 사이에서, 온실 운영 시스템에 소프트웨어를 설치하거나 외부 데이터 서버와 연동하는 기능을 가지며, 필요에 따라서 온실 운영 시스템과 연동하여 데이터를 피드백하는 기능을 갖는다.

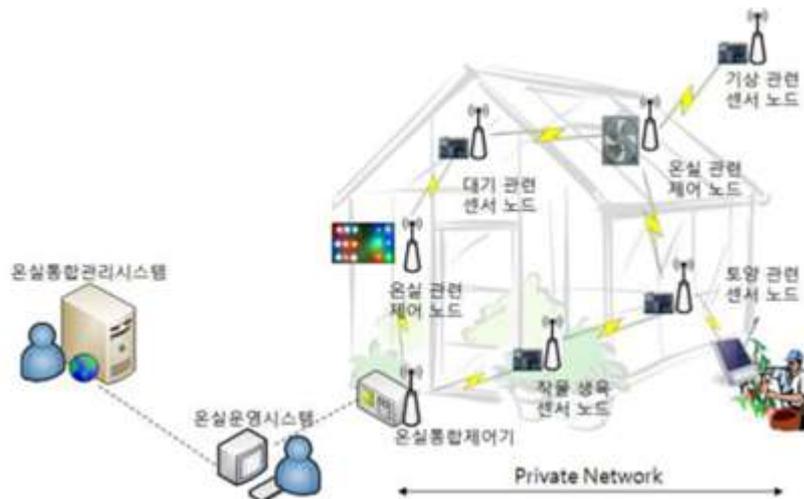


그림 212 - 온실 관제 시스템 개요

한편 위에서 언급한 온실 관제 시스템 표준 시리즈의 경우, 물리적인 장치로 의미 단위를 나누고 프로토콜을 정의하였기 때문에, 업체에 따라 다른 물리적인 형태로 시스템을 구성하는 경우 표준을 이해하기 어려움이 있었다. 그러므로 스마트 온실을 구성하는 각 기능 요소별로 인터페이스를 기술함으로써, 향후 관련 기능 요소 간 구체적인 메시지, 메타데이터 등을 좀 더 유연하게 정의할 수 있도록, 온실 각각의 구성 요소 간에 어떠한 정보가 교환되는지를 정의하는 스마트 온실 기능 요소 간 인터페이스('16) 표준이 개발되었다. 이 표준에서는 각각의 구성 요소 (센서노드, 구동기노드, 온실통합제어기, 온실 운영 시스템)에 필

요로 하는 정보의 형태를 좀 더 상세하게 정의하고 있다.

### 2.2) 시설원에 및 축산

앞에서 소개한 온실 관제 시스템 표준은 시설원에 환경에 설치된 장치들간 통신 부분만을 고려하였기 때문에, 실제 시설원에 설치된 센서나 구동기(천창, 팬 등)에 문제가 생겨 교체하기 위한 부분까지 만족시키지는 못한다. 그래서 2015년부터 시설원예와 축산분야 스마트팜 제조업체들이 농촌진흥청이 주관하는 ICT융합산업화 포럼을 중심으로 센서와 구동기에 대해 다음 그림 2와 같은 기계적, 전기적 접속 방법에 대한 공동규격을 개발한 바 있다.

기계적 연결 인터페이스 규격				전기적 연결 인터페이스 규격	
접속단자	결선형식	결선식별	단자순서	전원전압	출력신호 형태 및 범위
① 커넥터 ② 터미널단자	<2선식> ① W(백색) ② B(흑색)	① R(적색): 전원 ② Y(황색): 신호1 ③ G(녹색): 신호2 ④ B(흑색): 접지	① 전원 ② 신호1 ③ 신호2 ④ 접지	<직류> ① 5V ② 12V ③ 24V ④ 48V	<아날로그> (1) ① 0~3V ② 0~5V ③ 0~10V <아날로그> (2) ① 4~20mA ② 0~20mA
	<3선식> ① R(적색) ② Y(황색) ③ B(흑색)				
	<4선식> ① R(적색) ② Y(황색) ③ G(녹색) ④ B(흑색)				
	<교류> ① 220V 60Hz				
					<디지털> ① RS232 ② RS485 ③ CAN ④ SPI

그림 213 - 스마트 온실을 위한 센서 인터페이스 규격 조사표

### 2.3) 시설원예 센서 인터페이스

2016년 TTA 단체표준으로 제정되어 2018년 국가표준(KSC)으로 제정 예정인 본 표준은 우리나라에서 가장 널리 보급된 비닐하우스에서 가장 많이 사용되는 센서 13종(온도, 습도, CO<sub>2</sub>, 일사, 풍향, 풍속, 감우, 광양자, 토양함수율, 토양, 수분장력, EC, pH, 지온 센서 등)에 대한 접속단자·결선형식·결선식별·단자순서와 같은 기계적 인터페이스와 전원전압·출력신호 형태 등과 같은 전기적 인터페이스를 기술하고 있다. 또한 해당 센서들이 어떤 범위의 값을 측정할 수 있는지에 대한 운용 요구사항을 정의한다.

### 2.4) 시설원예 구동기 인터페이스

2015년 TTA 단체표준으로 제정되어 2018년 국가표준(KSC)으로 제정 예정인 본 표준은 구동기 9종(천창, 측창, 보온재, 커튼, 환풍기, 유동팬, 관수 모터, 관수 밸브, 냉난방기 등)에 대한 기계적·전기적 인터페이스와 구동과 관련된 운용 요구사항을 정의하고 있다.

### 2.5) 스마트 축사를 위한 내기 및 외기 센서 인터페이스

TTA 단체표준으로 제정된 본 표준은 스마트 축사에서 사용되고 있는 내기와 외기에 대한 센서 각각 8종(기온, 습도, CO<sub>2</sub>, 암모니아, 조도, O<sub>2</sub>, 차압, 풍속 센서 등) 및 7종(기온, 풍향, 풍속, 감우, 습도, 일사, 일조 센서 등)에 대한 기계적·전기적 인터페이스와 구동과 관련된 운용 요구사항을 정의하고 있다.

### 2.6) 유통

2016년도 TTA 단체표준으로 제정된 ‘농축산물 식품 메타데이터 모델링 가이드라인’ 표준은 농산물을 유통함에 있어 출처가 확실하지 않은 수많은 정보가 산재해 있다는 근본적인 문제점을 해결하기 위해, GS1 표준 메타데이터 서비스인 GS1 Source를 통해 농축산물 식품 메타데이터를 운용하는 방법을 제공하는 것을 목적으로 제정된 표준이다. 본 표준에는 제품과 지역의 식별자, 정보 제공자, 제품 데이터 레코드, 제품 원산지 정보 등과 같은 수많은 데이터를 XML 방식으로 규격화하여 표현하는 방법을 기술한다.

## 2.7) 스마트팜 클라우드

아마존, 구글의 사례에서처럼, 최근 컴퓨팅 환경이 서버 방식으로부터 클라우드 방식으로 변하고 있다. 미래의 스마트팜은 현재처럼 개별적인 시스템이 아니라 클라우드 기능을 이용한 시스템으로 변할 것으로 예측된다. 2016년도 TTA에서 개발된 “클라우드 기반 스마트팜 서비스 요구사항” 표준은 클라우드 컴퓨팅 기술을 기반으로 스마트팜을 관리, 운영하는 데 필요한 서비스의 기술적 요구사항과 구성 내용을 정의하는 것을 목적으로 하는 표준이다. 또한 본 표준은 후속 예정인 스마트팜 클라우드 세부 표준들의 참조 모델을 다음 그림에서처럼 정의하고 있다.



그림 214 - 클라우드 기반의 Farm as a Service (FaaS) 개념도

### (바) 국제표준화 동향

UN에서는 현재 74억의 세계 인구가 2050년에는 94억 명에 달할 것으로 예측한다. 이미 구글, IBM 등 글로벌 IT 기업들이 땅의 성질과 환경, 작물 상태 등을 분석한 데이터를 기반으로 종자·비료·농약 등의 분야에 인공지능(AI) 기술을 도입하고, 로봇, 드론, 사물인터넷 기술 등을 활용하는 스마트팜 시장을 선도함으로써 앞으로 다가올 전 세계 식량난에 준비하고 있는 모습을 보인다. 하지만 “총성 없는 3차대전”인 관련된 표준화는 지금까지 ITU-T 이외에는 그리 활발한 모습을 보이고 있지는 않다.

#### 1) ITU-T 표준화 동향

1865년 UN 산하 기구로 창설된 ITU는, 표준화를 담당하는 ITU-T, 무선통신을 담당하는 ITU-R, 개발도상국 지원을 위한 ITU-D 부문으로 운영된다. 이중 표준화를 담당하는 ITU-T에서는 IoT, Smart cities & communities와 관련된 표준화를 진행하고 있는 SG20 그룹과 Future networks(& cloud)와 관련한 표준화를 진행하고 있는 SG13 그룹에서 스마트팜과 관련된 표준화가 진행되고 있다.

ITU-T SG20에서 개발 중인 표준으로 강조할 표준으로는, “IoT 기반의 스마트 온실 서비스를 제공하기 위한 프레임워크 (ITU-T Y.ISG-FR)” 표준이다. 이 표준은 우리나라가 주도적으로 개발하고 있으며, 국내 업체 보유 기술을 토대로, 스마트팜 장치간 상호연동성을 제공하기 위한 요구사항과 구조를 기술하고 있다. 다음 그림에서 보이는 바와 같이, 국내 기술을 토대로 제정한 TTA 단체표준에서 다루고 있는 내용들을 포함시킬 수 있도록 기존의 개념을 확장하였다.

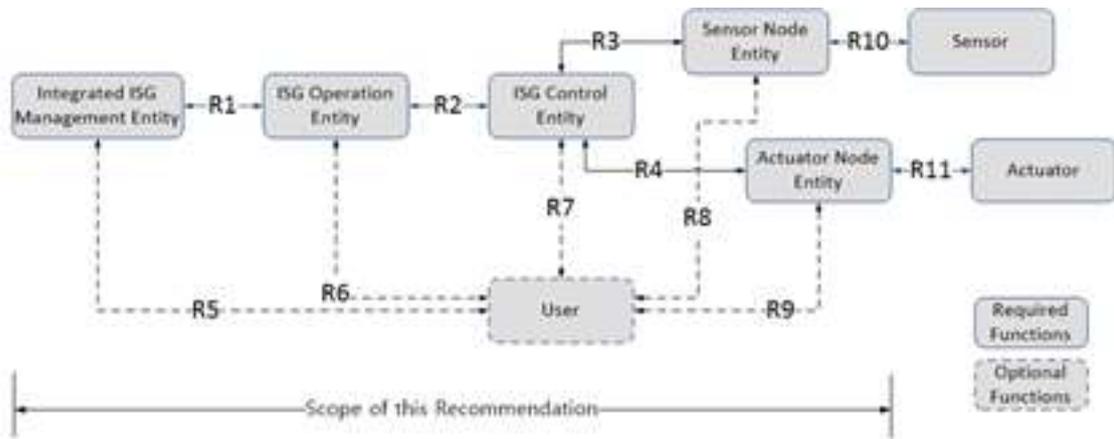


그림 215 - ITU-T Y.ISG-FR 참조 인터페이스

2) ISO 표준화 동향

1947년 창설된 UN산하 국제 표준화 기구로, IEC, ITU와 함께 세계 3대 국제 규모이다. ISO에서는 2018년 현재 249의 기술위원회(Technical Committee)를 통해 표준을 개발하고 있는데, 이 중 스마트팜 혹은 농기계와 관련된 표준을 개발하는 곳은 TC 23(Tractors and machinery for agriculture and forestry)이다. ISO TC23에서는 2007년도에 ISO 11783-Part1을 시작으로 2012년도까지 총 13편으로 발간된 ISO 11783시리즈 표준은 농림업용 트랙터와 장착식·반장착식·견인식·자주식 작업기에서 제어·통신을 위해 사용되는 직렬 데이터 네트워크, 트랙터 ECU 등이 정의된다.

3) oneM2M 표준화 동향

oneM2M은 사물통신, IoT 기술을 위한 요구사항, 아키텍처, API 사양, 보안 솔루션, 상호 운용성을 제공하기 위해 2012년 7월에 설립된 단체이다. oneM2M은 우리나라의 TTA를 비롯하여, ARIB(일본), ATIS(미국), CCSA(중국), ETSI(유럽), TTA(미국), TDSI(인도), TTC(일본) 등의 표준 개발 단체들(SDO) 및 브로드밴드 포럼, CEN, CENELEC, 글로벌플랫폼, 넥스트 제네레이션 M2M 컨소시엄, OMA 등 6개의 산업 표준 단체와 협력 관계를 갖고 있다. 또한 현재 200개 이상의 참여 파트너들과 회원들이 있으며, 여기에는 알카텔-루슨트, AT&T, BT 그룹, 어도비, 에릭슨, 도이체 텔레콤, IBM, 시스코 시스템즈, 시에라 와이어리스, 인터디지탈, 인텔, 삼성그룹, LG유플러스, 텔레포니카 등이 참여하고 있다.

특히 oneM2M은 2018년 12월 ITU-T SG20을 통하여, 자신이 만든 표준을 ITU-T 표준으로 채택시킨 바 있다. oneM2M은 스마트시티, 스마트그리드, 커넥티드카, 홈오토메이션, 스마트헬스 등 IoT 기반의 응용 서비스 프레임워크를 제공하는 표준을 개발하고자 한다. 특히 oneM2M에서는 관수시스템, 농업용 드론, 지능형 농산물 추적시스템 등과 같은 스마트팜 분야의 응용 서비스에도 관심을 갖고 있다.

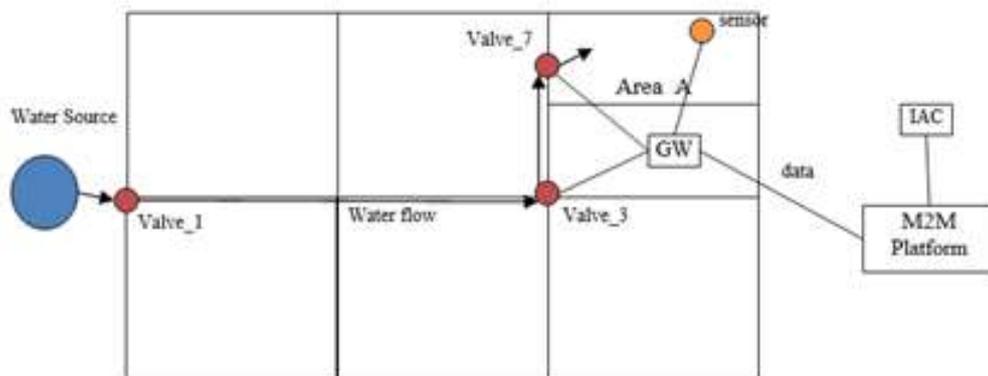


그림 216 - oneM2M의 농업용 관수 시스템

4) OCF 표준화 동향

OCF는 사물인터넷(IoT)에 관련된 장치에 대한 상호 운용성 표준을 개발하고 제품간 인증 프로그램을 제공하는 것을 목적으로 하는 산업 그룹이다. 현재 삼성 전자, 인텔, 마이크로 소프트, 퀄컴, 일렉트로룩스등을 포함 약 300개 이상의 회원사가 가입해 있다. 하지만 OCF의 표준화 분야는 스마트폰, 스마트헬스케어 등 주로 가전쪽의 IoT 응용 서비스 프레임워크 개발에 주로 집중되고 있다.

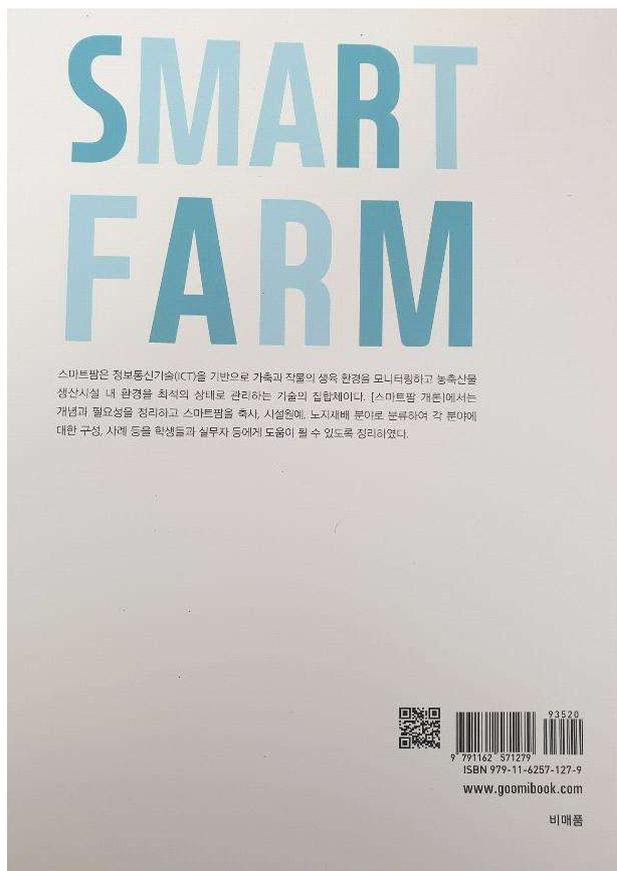
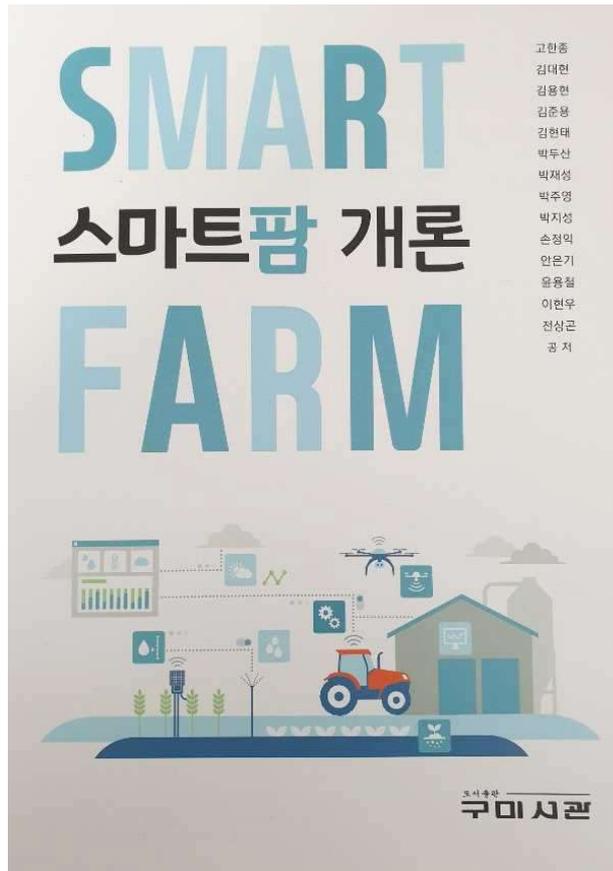


그림 217 - 해당연도 산출 저서 표지

가-2) 「시설원예학」(학문사) 공저 ('21)

국제표준화기구인 ISO에서는 표준을 "공통적이고 반복적인 사용을 위하여 제시된 규칙이나 지침, 또는 제품의 특성이나 관련 공정 및 생산방법을 함의를 통해 규정한 문서"라고 정의하고 있다.

즉 '표준(Standard)'이란, 함의를 통해 작성되고, 인정된 기관에 의해 승인되며, 공통적이고 반복적인 사용을 위해 제공되는 규칙, 가이드 또는 특성을 기술하는 문서를 의미한다. 또한 '표준화(Standardization)'란 표준을 만들기 위한 일련의 활동을 뜻하며, 생산, 소비, 유통, 서비스 등 여러 분야에 있어 능률을 증진하고 경제성을 높이기 위한 목적을 갖는다.

표준은 제정 목적과 방법, 준수 대상과 의무, 표준 제정 기구에 따라 다양한 방법으로 분류되곤 한다. 예를 들어, 단체표준, 국가표준, 국제표준 등과 같이 표준을 제정하는 기구에 따라 분류할 수도 있으며, 표준화 기구의 공인 여부에 따라, 공적표준 또는 사실표준으로 분류할 수 있다. 한편 일부에서는, 표준을 강제성이 수반되는 기술규정(Technical Regulation)으로 종종 오해하곤 한다. 기술규정이란 안전·국방 등 공공의 이익을 추구하기 위한 행정적 규정 즉, 상품의 특성, 공정 또는 성능 등을 규정하는 문서이며, 그 준수 여부가 강제적인 문서이다. 표준과 기술기준 간의 차이점을 표 4에 보였다.

표 4 - 표준과 기술규정의 차이점

	표준 (Standard)	기술규정 (Regulation)
제정목적	생산·유통의 효율성 제고	안전·국방 등 공공이익 추구
제정과정	표준화기구(공적표준) 혹은 시장요구(사실표준)에 따른 제정	정부 주도 제정 (예, 무선설비 기술기준)
준수의무	자발적인 준수	강제적인 준수
행정조항	거의 포함하지 않음	광범위하게 포함

ISO와 IEC는 체계적인 표준화를 진행하기 위해, 다음 표 5와 같은 절차를 규정하고, 각 단계에 따른 표준 문서의 품질과 최대 기한 등을 제시한다. 물론 모든 표준화 기구가 ISO와 IEC가 제시하는 표준 제정 절차를 따르는 것은 아니나 기본적으로 함의를 위한 방법은 대동소이하다.

표 5 - 표준 제정 절차 (ISO/IEC JTC1)

	관련 문서		
	국문 명칭	영문 명칭	약어
0 예비단계	예비 작업 항목	Preliminary Work Item	PWI
1 제안단계	신규 작업 항목 제안	New work item Proposal	NP
2 준비단계	작업안	Working Draft	WD
3 위원회단계	위원회안	Committee Draft	CD
4 질의안	질의안	Draft International Standard	DIS
5 승인단계	최종 국제표준안	Final Draft International Standard	FDIS
6 발간단계	국제표준	International Standard	IS
※ ISO와 IEC 표준 제정 절차는 동일함 ※ IEC에서는 DIS를 CDV(Committee Draft for Vote)라고 부름			

(가) 기기 표준화 현황

작물의 생육 환경을 정밀하게 조절하고 데이터 분석을 통해 과학적인 영농을 추구하게 됨에 따라, 다양한 ICT 기자재의 사용과 데이터의 축적·분석이 필요하게 되었다. 이러한 이유로 국내에서는 ICT 기자재와 데이터의 호환성을 제공하기 위한 표준화가 2014년부터 본격적으로 착수되었다. 다음에는 한국정보통신기술협회(TTA)를 통해 제정된 스마트팜 관련 표준들 중 일부를 간략하게 소개한다.

1) 국내 표준화

■ 스마트팜 관련 TTA 표준

TTA 스마트농업 PG(Project Group)는 ICT와 융합된 시설원예·축산·노지·유통 분야 표준화를 진행하고

있으며, 2020년도 현재 58개의 재개정 표준화 작업을 완료하였고, 11개의 신규 표준화 작업을 진행하고 있다. 다음은 TTA에서 제정한 스마트팜 관련 표준 중 일부를 간략히 설명하도록 한다.

#### ■ 온실 관제 시스템 표준

온실 관제 시스템 표준은 스마트팜과 관련되어 2012년에 승인된 최초의 국내 단체 표준이다. 이 표준은 온실에 IT 기술을 적용하기 위한 필수요소 항목을 정의하고, 수반되는 기술적 요구사항과 구성 내용을 정의하고 있다. 이 표준에서 정의하는 온실 관제 시스템이란 온실에서 작물 성장 관리를 위해 다양한 정보를 수집하고, 이를 활용해 성장 환경을 조성하는 설비 기기의 제어가 가능한 시스템으로 그 기능과 목적에 따라 4가지의 인터페이스를 정의하고 이들 각각에 대한 프로토콜을 정의하고 있다.

#### ■ 시설원에 센서 및 구동기 인터페이스 표준

앞에서 소개한 온실 관제 시스템 표준에서는 시설원에 환경에 설치된 장치 간 통신 부분만을 고려하기 때문에, 실제 시설원에 설치된 센서나 구동기(천창, 팬 등)의 기계적/전기적 호환성 문제를 다루고 있지 않다.

이러한 문제점을 해결하고자 2015년부터 시설원에 스마트팜 제조업체들의 의견을 수렴하여 시설원에 스마트팜 센서와 구동기의 인터페이스에 관한 표준을 제정한 이후, 2018년도에 KS 표준으로 이관하였다.

#### ■ RS485 Modbus 기반 센서/구동기 노드와 통합제어기간 통신 표준

스마트 온실의 센서/구동기 노드와 통합제어기간 서로 다른 통신방식을 사용하는 경우 동작을 위한 메시지 교환 방식이 다르기 때문에 기기 간 상호 연동성 확보가 어렵게 된다. 이러한 문제점을 해결하기 위해 RS485 Modbus방식을 사용하는 센서/구동기 노드와 통합 제어 기간 통신에 대한 메시지 포맷과 데이터를 정의한 표준이 개발되었다. 이 표준 또한 2018년도에 KS 표준으로 이관되었다.

#### ■ 스마트팜 클라우드

아마존, 구글의 사례에서처럼, 최근 컴퓨팅 환경이 서버 방식으로부터 클라우드 방식으로 변하고 있다. 미래의 스마트 팜은 현재처럼 개별적인 시스템이 아니라 클라우드 기능을 이용한 시스템으로 변할 것으로 예측된다. 2016년도 TTA에서 개발된 “클라우드 기반 스마트팜 서비스 요구 사항” 표준은 클라우드 컴퓨팅 기술을 기반으로 스마트 팜을 관리, 운영하는 데 필요한 서비스의 기술적 요구 사항과 구성 내용을 정의하는 것을 목적으로 하는 표준이다. 이 표준은 향후 개발될 클라우드 기반 스마트팜 제어를 위한 표준들을 위한 참조 모델을 제시한다.

#### ■ 스마트팜 관련 KS 표준

2020년 현재 스마트팜과 관련하여 KS 표준으로 제정된 것은 총 5편이다. 이들 표준은 국내 스마트팜의 실정에 따라, 센서 노드, 구동기 노드 및 통합 제어기에 대한 상호 호환성을 제공하기 위해 제정된 것으로서, 현재에는 이들 표준을 현장에 적용하기 위한 수정 작업이 진행되고 있다.

#### ■ 스마트 온실을 위한 센서 인터페이스

스마트 온실의 환경 모니터링과 제어를 위해서는 시설의 규모와 작물의 종류 등에 따라 다양한 형태의 센서와 구동기가 사용될 수 있으며, 시설의 용도와 설치자의 숙련도에 따라 센서나 구동기 등의 설치방법 또한 달라질 수 있다. 이는 특정 제조사의 제품이나 특정한 시설 방법에 의존적일 수밖에 없으므로 스마트 온실의 효율적인 유지·보수 및 관리가 어렵게 된다. 이에 이 표준은 스마트 온실을 구성하는 대표적인 센서 13종(온도, 습도, CO<sub>2</sub>, 일사, 풍향, 풍속, 광양자, 토양 함수율, 토양 수분 장력, EC, pH, 지온 등)에 대한 기계적/전기적 연결 규격과 측정 범위를 정의한다. 이때 케이블의 기본 배색은 KS C 6003[29]을 따른다. 한편, 이 표준에서는 과도한 옵션을 허용하기 때문에, 실제 현장에서 상호 호환성을 마련하기 어려운 실정이라는 지적에 따라 2020년부터는 이들 초안을 기반으로 관련 이해 당사자 간 대표 규격을 선정하여 해당 표준을 개정하려는 움직임이 일고 있다.

#### ■ 스마트 온실을 위한 구동기 인터페이스

스마트 온실을 위한 센서 인터페이스 표준 제정 목적과 마찬가지로, 스마트 온실에서 사용되는 구동기에 대한 상호 호환성을 마련하기 위해 9종(천창, 측창, 보온 덮개, 차광막, 환풍기, 유동 팬, 관수 모터, 관수 밸브, 냉난방기 등)을 선정하고 이들의 기계적/전기적 연결에 대한 세부 항목을 정의하고 있다. 이때 케이블의 기본 배색은 KS X 6003을 따른다.

이 표준 또한 과도한 옵션을 허용하기 때문에, 실제 현장에서 상호호환성을 마련하기 어려운 실정이다. 2020년부터는 이들 초안을 기반으로 관련 이해 당사자 간 대표 규격을 선정하여 해당 표준을 개정하려는 움직임이 일고 있다.

#### ■ 스마트 온실 센서/구동기 및 제어기 간 RS485 기반 모드버스(MODBUS) 인터페이스

스마트 온실의 센서/구동기 노드와 통합제어기간 서로 다른 통신방식을 사용하는 경우, 이종 제품 간 동작 수행을 위한 메시지의 이해가 불가하므로 제품 간 상호 연동성 확보가 어렵게 된다. 이러한 문제점을 해결하기 위해 이 표준에서는 RS485 모드버스를 이용하는 센서/구동기와 통합제어기간 통신을 위한 메시지 포맷과 데이터 교환 방식을 정의한다.

이 표준은 RS485 모드버스 통신 프로토콜을 수정 없이 사용하므로, 가장 중요한 부분이 데이터를 식별

하기 위한 레지스터 주소가 된다. 가령, 온실의 온도 측정값을 알고 싶을 때, RS485 모드버스 통신방식에서 사용하는 GetData라는 명령을 통해 특정 레지스터 주소에 저장된 값을 가져온다. 이때 이들 특정 레지스터 주소에 대한 정보를 레지스터 맵이라 하는데, 이 표준에서는 아직 레지스터 맵에 대한 정의가 부족하므로, 이해 관계자들의 의견을 수렴하여 레지스터 맵에 대한 보완 작업이 진행되고 있다.

## 2) 국제 표준화

스마트팜 관련 국제표준화에 대한 관심이 점차 일고는 있지만, 아직 ITU-T 이외에는 그리 활발한 모습을 보이지 않고 있다.

### ■ ITU-T 표준화

1865년 UN 산하 기구로 창설된 ITU는, 표준화를 담당하는 ITU-T, 무선통신을 담당하는 ITU-R, 개발도상국 지원을 위한 ITU-D 부문으로 운영된다. 이중 표준화를 담당하는 ITU-T에서는 IoT, Smart cities & communities와 관련된 표준화를 진행하고 있는 SG20 그룹과 Future networks(& cloud)와 관련한 표준화를 진행하고 있는 SG13 그룹에서 스마트팜과 관련된 표준화가 진행되고 있다. ITU-T SG20에서 제정한 ITU-T Rec. Y.4466[30] 국제표준은 IoT 기술을 사용하는 스마트팜에서 반드시 고려되어야 할 기능들과 인터페이스를 국내 주도로 개발한 표준이다.

### ■ ISO 표준화 동향

1947년 창설된 UN 산하 국제 표준화 기구로, 2018년 현재 249의 기술 위원회(Technical Committee)를 구성하고 있다. 이중 스마트팜 혹은 농기계와 관련된 표준을 개발하는 기술 위원회는 TC 23(Tractors and machinery for agriculture and forestry)으로, 2007년도부터 농림업용 트랙터와 장작식·반장작식·견인식·자주식 작업기간 통신을 위해 사용하는 직렬 데이터 네트워크, 트랙터 ECU 등을 정의하는 ISO 11783시리즈 표준을 제정한 바 있다.

### ■ oneM2M 표준화 동향

oneM2M은 IoT 기술을 위한 요구사항, 아키텍처, API 사양, 보안 솔루션, 상호 운용성을 제공하기 위해 2012년에 설립되었다. oneM2M은 우리나라의 TTA를 비롯하여 6개의 산업 표준 단체와 협력 관계가 있으며, 현재 200개 이상의 참여 파트너들과 회원들을 보유하고 있다. oneM2M의 주요 관심사는 스마트시티, 스마트그리드, 커넥티드카, 홈오토메이션, 스마트 헬스 등 IoT 기반의 응용 서비스 프레임워크의 표준화이다. 한편 최근에는 관수 시스템, 농업용 드론, 지능형 농산물 추적 시스템 등과 같은 스마트팜 분야의 응용 서비스에도 관심을 두고 있으며, 2018년에 ITU-T SG20을 통하여, oneM2M 표준을 ITU-T 표준으로 채택시킨 바 있다.

### ■ OCF 표준화 동향

OCF는 사물인터넷(IoT)에 관련된 장치에 대한 상호 운용성 표준을 개발하고 제품 간 인증 프로그램을 제공하는 것을 목적으로 하는 산업 그룹이다. 현재 삼성전자, 인텔, 마이크로소프트, 퀄컴, 일렉트로룩스 등을 포함 약 300개 이상의 회원사가 가입되어 있으나, 아직까지 OCF의 주요 표준화 분야는 스마트홈, 스마트 헬스케어 등 주로 가전 분야의 IoT 응용 서비스 프레임워크 개발에 집중되고 있다.

#### (나) 데이터 표준관리시스템(open API)

농촌진흥청의 분류 기준에 따르면, 제1세대 스마트팜은 시간과 장소의 제약 없는 온실의 원격 제어로 사용자 편의성 고도화를, 제2세대 스마트팜은 빅데이터 활용을 통해 생산량의 증대, 품질의 고도화 및 비용 절감을 목적으로 한다.

스마트팜에서 말하는 빅데이터란 작물의 생육상태를 정밀 관리하기 위한 복합 데이터로, 크게 환경, 제어, 생육 및 경영 데이터로 구성되며,

이들 데이터는 인공지능 기술과 결합되어 최적의 생육조건을 도출하거나, 경영 비용 절감을 위한 자료로 활용될 수 있다. 이러한 데이터들을 개인, 기업 또는 공공이 관리할 수 있는데, 현재 우리나라에서는 농촌진흥청과 농림수산물교육문화정보원이 스마트팜 빅데이터를 공공 형태로 제공하고 있다. 다음 표 6은 국내에서 제공되는 스마트팜 공공데이터의 서비스유형과 제공기관을 정리한 표이다.

표 6 - 국내에서 제공되는 스마트팜 공공데이터

데이터 구분	공공데이터명	서비스유형	제공기관
스마트팜	스마트팜 빅데이터 제공 서비스	SOAP/RESTful	농림수산물교육문화정보원
노지	스마트팜 빅데이터 제공 서비스	SOAP/RESTful	농림수산물교육문화정보원
축산	스마트팜 빅데이터 제공 서비스	SOAP/RESTful	농림수산물교육문화정보원
스마트팜	농진청 파일 데이터 서비스	파일/동영상	농촌진흥청
스마트팜	스마트팜 우수농가 데이터	공공데이터포털	농촌진흥청

농촌진흥청에서 제공하는 공공데이터는 홈페이지를 통해 다운을 받을 수 있으며, 농림수산물교육문화정보원에서 제공하고 있는 SOAP/RESTful 기반의 Open API(<https://www.smartfarmkorea.net>)를 통해 다운받을 수 있다. 다음 표 7은 농림수산물교육문화정보원의 스마트팜 OpenAPI를 사용하기 위한 절차를 정리한 표이다.

표 7 - 농림수산물교육문화정보원의 스마트팜 Open API 사용 절차

단계	단계 이름	설명
1	OPEN API신청	제공되는 '신청서' 양식에 따라, API 신청자 정보와 용도 등을 제출
2	관리자승인	활용 목적 등을 관리자가 검토하여 승인여부 결정
3	승인완료	신청자의 추가 정보 제공
4	OPEN API활용	기관으로부터 발급된 서비스키를 활용하여 API 사용
5	활용결과등록	API 서비스 개선을 위한 목적으로 활용결과 등록

일단 Open API의 사용을 위한 서비스키를 확보한 사용자는, 농림수산물교육문화정보원에서 제공하고 있는 SOAP/RESTful 기반의 Open API를 사용하여 필요한 데이터를 획득할 수 있다. 다음 표 8은 농림수산물교육문화정보원의 빅데이터에 접근하기 위한 URI이다.

표 8 - 농림수산물교육문화정보원의 스마트팜 빅데이터 URI

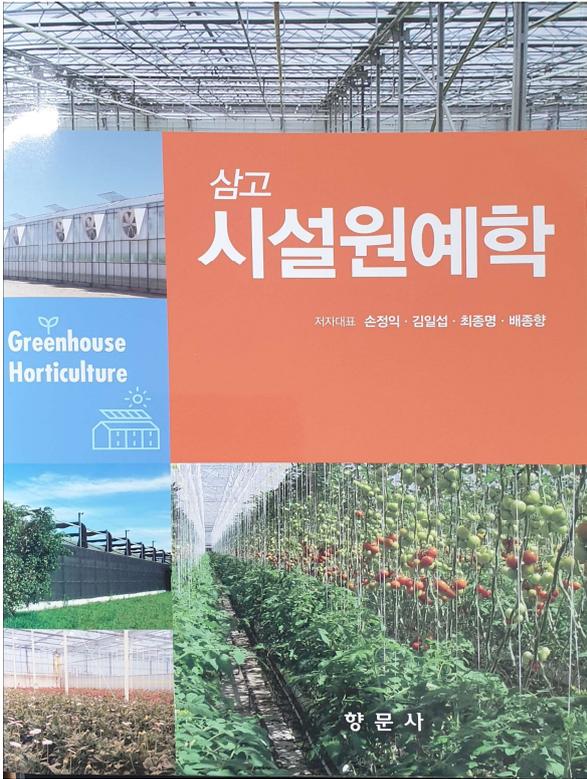
서비스 타입	URI
SOAP	<a href="http://www.smartfarmkorea.net/Agree_WS/webservices/ProvideService?wsdl">http://www.smartfarmkorea.net/Agree_WS/webservices/ProvideService?wsdl</a>
RESTful	<a href="http://www.smartfarmkorea.net/Agree_WS/webservices/ProvideRestService/">http://www.smartfarmkorea.net/Agree_WS/webservices/ProvideRestService/</a>

한편 위의 URI를 통하여 다음 표 9에 나열한 8 종류에 대한 데이터를 획득 가능하다.

표 9 - 농림수산물교육문화정보원의 스마트팜 빅데이터 종류

일련 번호	오퍼레이션명(영문)	오퍼레이션명(국문)
1	getIdentityDataList	아이덴티티정보 (농가ID, 시설ID, 지역, 품목코드 등)
2	getCroppingSeasonDataList	작기정보 (작기일자, 재배 면적, 작기 품목 등)
3	getEnvDataList	환경정보 (내부/외부/토양 환경, 양액 및 제어 정보)
4	getCultivateDataList	생육정보 (토마토/파프리카/오이/가지의 생장길이, 잎 길이/수 등)
5	getStrbCultivateDataList	생육정보 (딸기의 생장길이, 잎 길이/수 등)
6	getMumCultivateDataList	생육정보 (국화의 생장길이, 잎 길이/수 등)
7	getFruitCultivateDataList	생육정보 (참외의 생장길이, 잎 길이/수 등)
8	getManagementData	경영정보 (인건비, 양액비, 방제비, 출하량 등)

끝으로, 앞서 간략하게 소개한 농림수산물교육문화정보원에서 제공하는 OpenAPI에 대한 상세한 사용 방법은 <https://www.smartfarmkorea.net/> 을 참고한다.

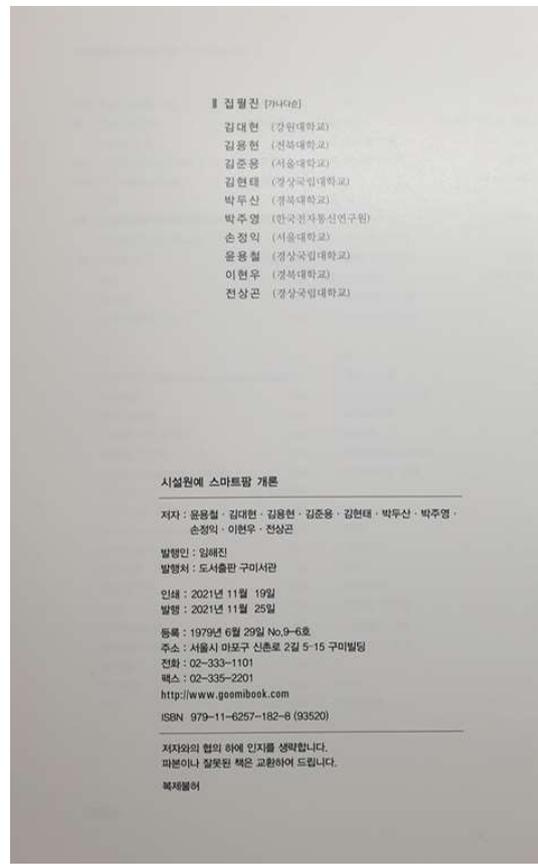
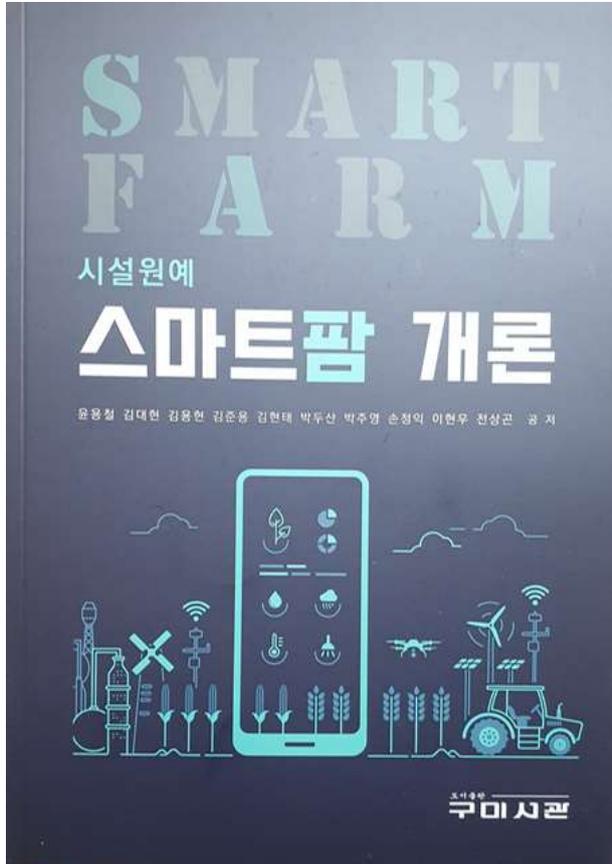


\* 저 자 약 려 \*

김무현	서울대학교 농업생명과학대학 식물생산과학부 졸업 서울대학교 대학원 농학박사 현 배재대학교 원예산업학과 교수
곽유리나	서울대학교 농업생명과학대학 식물생산과학부 졸업 도쿄대학교 대학원 농학박사 현 연일대학교 스마트원예경영 교수
김성걸	전국대학교 농속산생물과학대학 농업교과학과 졸업 서울대학교 대학원 농학박사 현 경북대학교 원예과학과 교수
김영식	고려대학교 농과대학 원예학과 졸업 도쿄대학교 대학원 농학박사 현 상명대학교 식물식품공학과 교수
김용현	서울대학교 농과대학 농공학과 졸업 서울대학교 대학원 농학박사 현 전북대학교 생물산업기계공학과 교수
김일섭	강원대학교 농업생명과학대학 원예학과 졸업 유수대학교 대학원 농학박사 현 강원대학교 원예·농업자원경제학부(원예과학 전공) 교수
김종준	고려대학교 자연과학대학 원예과학과 졸업 조지아대학교 대학원 농학박사 현 고려대학교 식물생명공학과 교수
박경섭	서울대학교 농업생명과학대학 원예학과 졸업 서울대학교 대학원 농학박사 현 북호대학교 원예과학과 교수
박준석	전국대학교 농과대학 농공학과 졸업 도쿄대학교 대학원 농학박사 졸업 현 충남대학교 원예학과 교수
박주영	충남대학교 공과대학 전기공학과 졸업 충남대학교 대학원 공학박사 현 한국전자통신연구원 책임연구원
배중항	원광대학교 농과대학 원예학과 졸업 원광대학교 대학원 농학박사 현 원광대학교 원예 산업학부(원예학 전공) 교수

가-3) 「스마트팜 개론」 (구미서관) 공저 ('21)

해당연도에는 ARC 스마트팜 전문가들과 협업을 통해, 국내 대학교 및 대학원에서 수업 교재로 활용이 가능한 스마트팜 개론을 개정하였다. 이번 도서는 2020년도에 출간한 초판에 비해, 시설원예에 국한하여 도서의 내용을 축소하였으며, 인공지능 등 최신 기술을 업데이트하였다. 본 연구진은 해당 도서의 일부로서 표준화의 개요, 스마트팜 관련 국내외 표준화 동향 내용을 수록하였다. 개정된 내용이기 때문에 본 보고서에서는 본 저서의 내용 수록을 생략하기로 한다. (ISBN 979-11-6257-182-8)



## 나) 기술지원

### 나-1) 스마트팜 단체표준 (3건) 제정을 위한 기술지원

#### (가) 연구 수행 개요

해당 표준은 시설원에 관제를 위해 사용되는 스마트팜 ICT 기자재에는 다양한 정보들이 저장되어야 한다. 이를 위해, 그간 RS486 Modbus 기반의 통신을 표준화 하기 위한 “디폴트 레지스터 맵 (default register map)” 등과 같은 방법이 국내 스마트팜 제조업체들간에 표준으로 제시된 바 있다.

하지만, 최근 유무선 통신 기술이 발달하고, 제어하는 기능이 발전함에 따라, 기존의 RS485 Modbus 기반의 레지스터 맵으로는 한계성이 예상되는바, 당해 년도에는 한국스마트팜 산업협회의 시설원에 표준위원회를 지원하여 “스마트온실 ICT 기자재 MIB 정보 체계” 시리즈 표준개발을 지원하였다.

해당 표준의 주요한 내용은 “스마트온실 ICT 기자재를 위한 MIB (Management Information Base) 정보체계”를 기술하는 것인데, MIB란 초기에는 네트워크 장비의 관리를 위한 관리 정보 (Management Information)를 체계화(Hierarchically)하기 위한 목적으로 주로 사용되었으나, 그 사용 범위가 센싱 정보의 전달이나 이들 정보의 저장과 관리를 위한 목적으로 점차 확대되고 있다. 해당 표준에서 정의하는 MIB 정보체계는 시설원예에서 사용되는 스마트팜 ICT 기자재들의 정보를 기술하고 이들 정보를 OID 형태로 활용하기 위한 방법을 설명한다.

본 시리즈 표준은, 다음과 같이 구성되었다.

- 제1부: 일반 사항
- 제2부: 온실통합제어기
- 제3부: 양액기

#### (나) 연구 주요 내용

##### 1) MIB 정보체계 개요

###### 1.1) MIB 소개

관리를 수행할 때 Manager와 Agent사이에 특정한 정보를 주고 받는 것이 통신의 기본이다. 관리되어야 할 특정한 정보 (Information)나 자원 (Resource)을 통칭하여 객체 (object)라 한다. 이런 객체들을 모아놓은 집합체를 MIB (Management Information Base)라고 한다.

시설원예의 환경을 관리하는 행위는, 관리대상인 장비들이 보유하는 MIB 중에서 특정 값을 얻어와서 그 장비의 상태를 파악하거나 그 값을 변경함을 의미한다. 상태를 파악한다는 것은 단순히 해당 MIB의 문자열이나 수치를 읽어오는 것을 의미한다. 이렇게 읽어온 값은 해당 장비와 관련된 상태 정보나 설정값을 의미한다. 값을 변경한다는 것은 해당 MIB의 문자열이나 수치를 변경하는 것을 의미하는데, 이러한 값의 변경을 통하여 해당 장비의 상태를 변경시키거나 일정한 작동을 지시 또는 수행할 수 있게 한다.

해당 표준에서 고려하는 관리 대상인 기본 장비는 센서노드, 구동노드, 온실통합제어기, 양액기 등같은 장비들을 포함한다. 하지만 필요할 경우 통신을 위해 사용되는 게이트웨이, 클라우드 관리 서버와 같은 장비들까지도 그 정보를 확장할 수 있다.

MIB를 정의하고 구성하는 프레임워크인 SMI (Structure of Management Information)는 RFC1155에 정의되어 있으며, 주요 내용은 다음과 같다.

- a) MIB의 각 객체들은 ISO와 ITU-T의 표준인 ASN.1(Abstract Syntax Notation One)을 사용해서 정의한다.
- b) 정의할 모든 객체는 반드시 이름(name), 선택스(syntax), 부호화(encoding)에 관한 정보를 포함해야 한다. 이름은 해당 객체를 식별하는 객체 식별자(object identifier), 선택스는 객체의 데이터 종류(가령, 정수인지 문자열인지), 부호화는 객체의 데이터가 전송되는 비트패턴을 의미한다.
- c) ASN.1에서 정의된 데이터 종류 중 SNMP에서 사용할 수 있는 것과 허용되는 구조가 명시되어 있어야 한다.

MIB 구조는 계층적 트리구조의 형태를 이루고 있다. MIB 내에서 특정 객체는 OID로 식별된다. OID의 형태는 일반적으로 사용자에게 친숙한 문자열이 아니라 연속된 정수 형태이다. MIB 트리는 루트를 기준으로 동일한 범주에 속하는 객체들을 분류하는 식으로 OID가 정해지고 SNMP는 최종 노드의 리프(leaf)만을 읽고 쓸 수가 있다.

다음 그림 1은 MIB는 MIB-2에서 필요로 하는 관리 항목들을 정의한 것이다. 예를 들어 System

Location에 관한 정보를 그림 1의 sysLocaion에 보관을 하고 있다면, System Location에 접근하기 위한 OID는 sysLocaion이 아니고 "1.3.6.1.2.1.1.6"이다. 또한 그리고 이 객체는 읽고 쓸 수 있으나 부모 객체인 system("1.3.6.1.2.1.1")은 읽거나 쓸 수 없다. 이는 MIB 체계에서는 반드시 최종 리프(leaf)만을 읽거나 쓸수 있기 때문이다. 만일 사용자가 sysLocation에 저장된 값 이외에도 동일 계층에 있는 "sysDescr, sysObjectID, ... ,sysServices"를 모두 읽어 오고 싶으면 "1 ~ 7" OID 모두를 지정해서 에이전트에게 물어보아야 한다. 실제로 SNMP로 관리를 하는 경우, 전술한 방법에 따른 조작으로 얻은 결과이다.

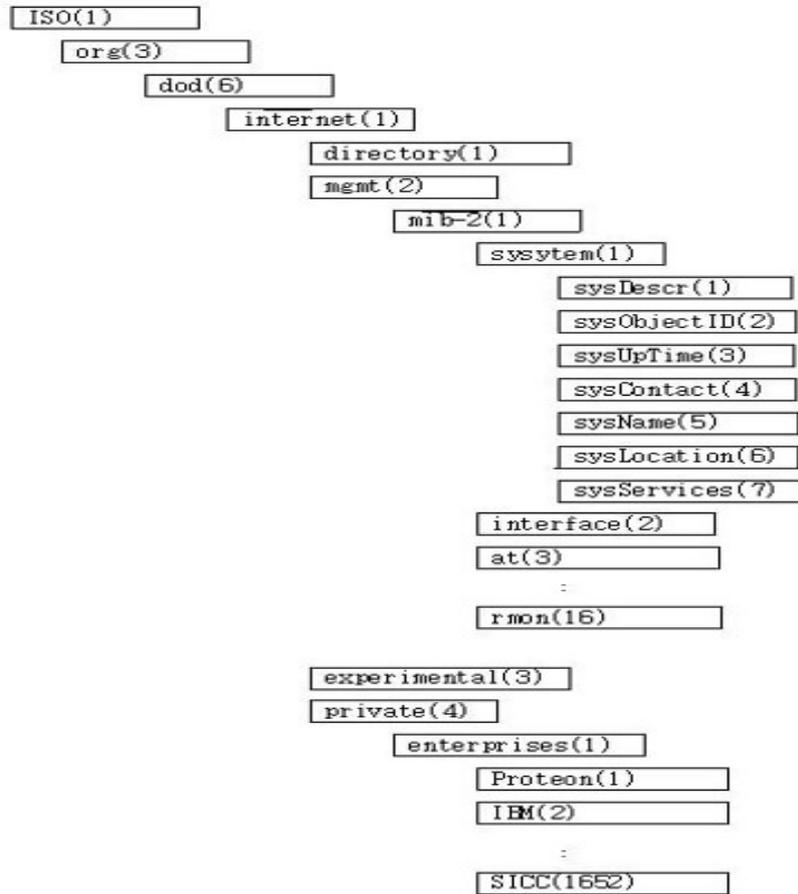


그림 1 - MIB 구조 개요 (예시)

### 1.2) MIB 계층

SNMP SMI는 "iso(1) org(3) dod(6)"의 하부-트리로 "internet(1)"을 명시하고, "internet(1)"의 하부-트리로서 directory(1), mgmt(2), experimental(3), private(4)들을 정의한다. 이 때 private(4)의 하부-트리로 enterprises(1)를 정의하고 있는데, 이 항목은 SNMP MIB에 표준으로 정의되어 있지 않는 어느 특정 회사만의 제품을 등록할 때 사용한다.

즉, 어느 시설원에 ICT 기자재를 공급하는 업체가 자신의 제품을 MIB에 등록하고자 할 때는 이 private(4) enterprises(1)에 정의할 수 있다. 물론 private(4) enterprises(1)의 하부-트리는 해당 업체에서 임의로 사용할 수 있다.

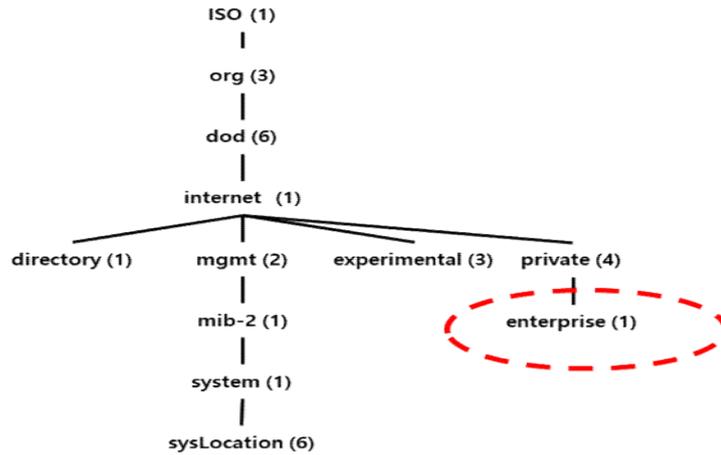


그림 2 - 사설 기업의 장치 등록을 위해 확장가능한 MIB

참고로, 글로벌하게 사용할 수 있기 위해서는 각각의 업체들을 구별하는 OID가 있어야 한다. OID는 IANA(Internet Assigned Number Authority)에서 관리하고 있기 때문에 Private Enterprises MIB를 구현하기 위해서는 IANA에서 OID를 부여 받아야 한다. 그래야 그 하위 MIB의 객체의 유일성이 보장되고 다른 업체의 사설 MIB와 구별이 가능하기 때문이다.

## 2) 시설원에 ICT 기자재를 위한 MIB 구조

### 2.1) MIB 정보 관리 대상

다음 그림은 스마트온실에 설치되어 온실관제서비스를 제공하기 위한 온실관제시스템을 도시하고 있다. 센서노드, 구동기노드, 온실통합제어기, 온실운영시스템, 온실통합관리시스템으로 구성된 온실관제시스템은 온실에서 작물 생장 모니터링 및 관리를 위해 센서노드를 통하여 다양한 정보를 수집하고, 이를 활용해 적정 생장환경을 조성하는 설비기기를 구동기노드를 통해 제어가 가능한 시스템으로 생장환경 관리, 생장환경 제어와 같은 응용 서비스 기능등을 제공한다.

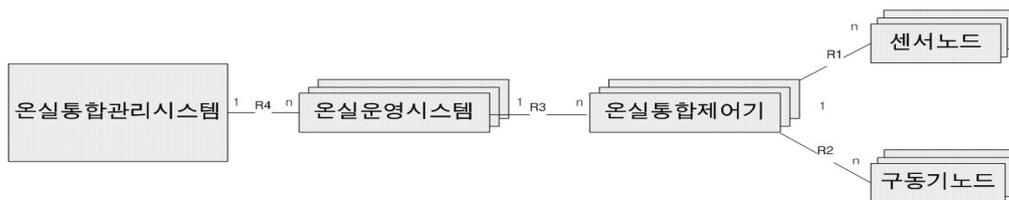


그림 3 - 온실관제시스템 개요도 (TTAK.KO-10.0934)

각 센서노드에는 다수의 센서를 포함할 수 있으며, 각 구동기노드는 다수의 구동기를 포함할 수 있다. 온실통합제어기는 센서노드로부터 정보를 수집하고 온실운영시스템으로 전달하는 역할과 온실운영시스템으로부터 받은 명령을 구동기노드로 전달하는 역할을 한다. 필요에 따라 프로토콜을 변환하는 역할도 하며 간단한 제어로직을 포함할 수도 있다. 온실운영시스템은 온실 내·외부 센서로부터 환경 및 작물 생장정보를 모니터링 할 수 있으며, 온실 환경제어 알고리즘을 통한 운영과 이를 통한 제어가 가능하다. 수집된 데이터를 데이터베이스에 기록보관하며 저장된 데이터를 바탕으로 영농일지를 작성할 수 있다. 또한, 온실운영에 필요한 서비스 및 제어 소프트웨어를 온실통합관리시스템으로부터 다운받아 사용할 수 있다. 그리고 센서노드 및 구동기노드의 동작주기 및 장애를 관리한다. 온실통합관리시스템은 다른 사이트에 위치한 온실운영시스템과 데이터서버를 연동하여 작물 생육정보를 피드백 시켜준다. 센서노드 및 구동기노드에 따라 필요한 소프트웨어를 온실운영시스템에 설치시켜 주는 역할을 한다. [TTAK.KO-10.0934] 이들 ICT 기자재들은 상호 연계하여 정보를 주고 받을 수 있어야 한다. 이를 위해 잘 정의된 MIB는 정보를 주고 받을 수 있는 프로토콜의 설계나 장치의 유지보수에 매우 효과적이다.

## 2.2) 장치 내부의 MIB 구조

다음 그림 4는 해당 표준에서 다루는 MIB의 구현 방식을 도시한 그림이다. 시설원에 ICT 기자재들은 각각의 임무를 수행하기 위한 컴퓨팅 자원 (RAM, 프로세서, 제어로직), 사용자와의 상호작용을 위한 사용자인터페이스, 다른 장치와의 통신을 위한 통신인터페이스, 그리고 장치를 관리하기 위한 MIB 등의 기능 블록을 갖는다.

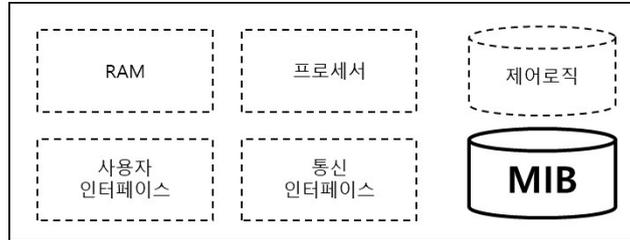


그림 4 - ICT 기자재 내부 개념도

각각의 기능 블록은 다음과 같은 역할을 담당한다.

- 제어로직 : 장치의 목적을 수행하기 위한 조건식의 관리
- 프로세서 : 장치의 기능을 실행하는 객체로서 제어로직의 연산 수행
- RAM : 프로세서 연산에 필요한 데이터의 저장
- 사용자 인터페이스 : 사용자에게 온실 관제를 위한 입출력 기능 제공
- 통신 인터페이스 : 원격 장치와의 통신 기능 제공
- MIB : 장치를 관리하기 위한 정보들을 저장

이때 RAM에 임시 저장되는 정보에는 프로세서의 연산에 필요한 데이터 이외에도, 온실의 환경 측정값과 제어 정보를 포함된다. 해당 표준에서 정의하는 MIB 정보체계는 그림 3에 도시한 시설원에 ICT 기자재들은 물론 양액기나 제어 기능이 포함된 장치들의 관리정보를 저장하는데 활용할 수 있다. 또한 시설원에 ICT 기자재들을 관리하기 위한 프로토콜 설계는 물론 데이터 수집을 위한 참조 데이터 구조 모델로 활용할 수 있다.

### 3) 온실통합제어기를 위한 MIB 정보체계

다음 그림 5는 해당 표준에서 제안하고자 하는 온실통합제어기 내부의 MIB 정보 체계이다. 해당 표준에서 다루는 온실통합제어기의 MIB는 그림 5에서 점선으로 표시한 부분에 해당된다. 온실통합제어기 MIB 정보체계를 설명하기 위한 예로써, 임의의 제조사에서 개발한 온실통합제어기는 sf\_xxx(55667)의 sf\_controller(1) 항목을 정의한다. 이 정의는 MIB 트리 내에서 {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_controller(1)}의 구조를 가지며, 이는 {1 3 6 1 4 1 55667 1} 라는 OID 형태로 표현될 수 있다. 해당 표준에서 정의하는 온실통합제어기의 MIB 정보체계는 그림 5에서와 같이 세가지 그룹으로 구성된다. 첫번째는 시스템에 관한 정보(sysInfo), 두번째는 센서노드에 관한 정보(sensorInfo), 그리고 마지막 세번째는 구동기노드에 관한 정보(actuatorInfo)이다.

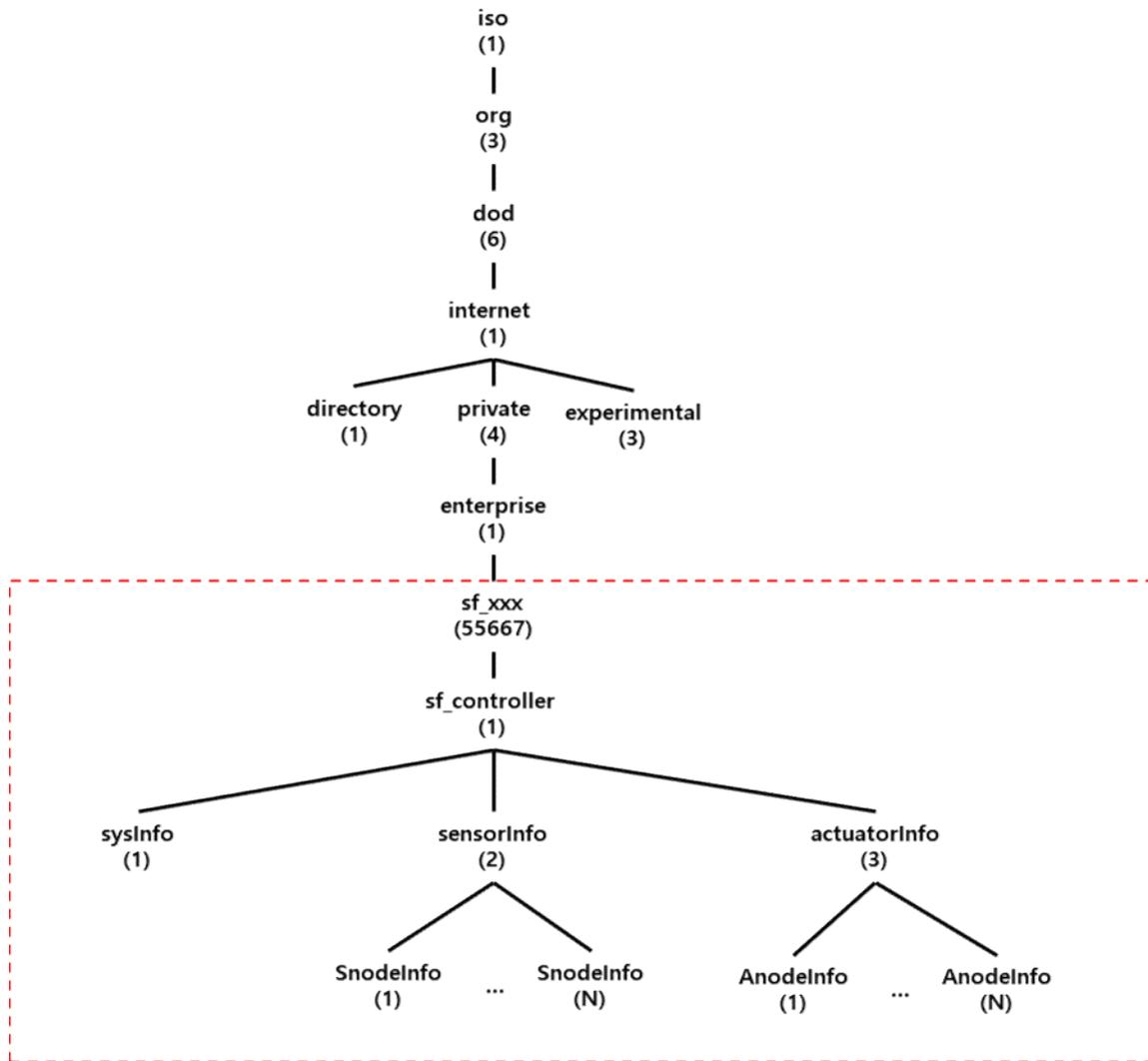


그림 5 - 온실통합제어기의 MIB 계층

### 3.1) 온실통합제어기 시스템 정보 (sysInfo)

시설원예에서 사용되는 온실통합제어기 자체에 대한 시스템 정보 구조는 다음 표 1과 같이 정의된다. 표 1은 그림 5의 9번째 계층인 sysInfo(1) 이후의 정보들에 대해 기술한 것이다. 온실통합제어기의 시스템에 관한 주요 정보로는 시스템 모델 정보, 펌웨어, 시간 정보, 네트워크 정보, 시설 위치 정보 및 오류 정보등으로 구성된다.

사용 방법의 예로서, 아래 표 1에서 가령 온실통합제어기가 설치된 시설의 위치를 알고 싶다면, {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_controller(1) sysInfo(1) GeoInfo(7) 관제시설위치(1)}에 저장된 정보를 탐색하면 된다. 이는 또한 {1 3 6 1 4 1 55667 1 1 7 1}의 형태로 표현된다.

표 1 - 온실통합제어기 시스템 정보

MIB 계층					개체 이름	값의 예
9	10	11	12	13		
1					SysInfo	
	1				ModelName	"EnvCont-100"
	2				FirmwareVer	"v1.xx_xxxx"
	3				Time	"yyyy-mm-dd-hh-mm-ss"
	4				TimeZone	"GMT+hh:mm"
	5				RebootInfo	'0' 또는 '1'
	6				NetworkLinkInfo	
		1			Network #1	
			1		RS485Modbus Adapter	"0" 또는 "1" (사용 여부)
			2		NodeID	"1~255"
		2			Network #2	

		1	IP Adapter	"0" 또는 "1" (사용 여부)
		2	NetMacInfo	"XX:XX:XX:XX:XX:XX"
		3	NetIPInfo	"xxx.xxx.xxx.xxx"
		4	GWInfo	"xxx.xxx.xxx.xxx"
		5	MaskInfo	"xxx.xxx.xxx.xxx"
	N		Network #N	"0" 또는 "1" (사용 여부)
7			GeoInfo	
		1	GeoLocation	"43°46'9"N111°15'22.2"E"
		2	Description	"강원도 철원군 산 XXX"
		3	Size	"가로 X 세로 X 높이 (m)"
		4	Type	"단동형, 연동형"
		5	RoofType	"아치형, 지붕형"
		6	SunRiseTime	"yyyy-mm-dd-hh-mm-ss"
		7	SunSetTime	"yyyy-mm-dd-hh-mm-ss"
8			SystemFaultInfo	
		1	SystemFault	
		1	PowerFault	
		1	OccurrenceTime	"yyyy-mm-dd-hh-mm-ss"
		2	HardwareFault	
		1	OccurrenceTime	"yyyy-mm-dd-hh-mm-ss"
	2		SecurityFault	
		1	NoAccessRight	
		1	OccurrenceTime	"yyyy-mm-dd-hh-mm-ss"

### 3.2) 온실통합제어기 센서노드 관리 정보 (sensorInfo)

시설원예에서 사용되는 온실통합제어기가 관리하는 센서노드에 대한 정보 구조는 다음 표 2와 같이 구성된다. 온실통합제어기가 관리하는 센서노드에 대한 정보는 크게, 센서노드에 대한 시스템 정보, 센서노드와의 통신을 위한 네트워크 정보, 센서노드의 위치 정보, 시스템 알림 정보 및 측정 정보로 구성된다. 표 2는 그림 5의 9번째 계층인 sensorInfo(2) 이후의 정보들에 대해 기술한 것이다.

사용 방법의 예로써, 어느 특정 센서노드가 설치된 위치를 알고 싶을 경우, {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_controller(1) sensorInfo(2) SensorNodeInfo#1(1) GeoInfo(10) 위치 정보 (1)}에 저장된 정보를 탐색하면 되며, 이 정보는 {1 3 6 1 4 1 55667 1 2 1 10 1}의 형태로 표현된다. 마찬가지로 가령 습도 센서가 설치된 구역을 알고 싶을 경우, {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_controller(1) sensorInfo(2) SensorNodeInfo#1(1) measurementInfo(11) 습도(2) 구역정보(4)}에 저장된 정보를 탐색하면 되며, 이 정보는 {1 3 6 1 4 1 55667 1 2 1 11 2 4}의 형태로 표현된다.

만일 어떤 특정 센서노드에서 동일한 성격의 다수 센서를 관리한다고 하면, 해당 센서에 대한 정보를 추가할 수 있다. 이를 위해 표 2에서는 온도에 관한 센서가 N개까지 설치되어 있는 경우를 예시하였다. 만일 추가적인 센서를 설치할 경우 온도센서 정보를 저장하기 위한 방법을 응용할 수 있다. 한편, 하나의 온실통합제어기가 다수의 센서노드를 포함시킬 경우 관리하고 있는 센서노드에 대한 정보를 추가할 수 있다. 예를 들어, 2개의 센서노드가 설치되어 있는 경우, 표 2에서 SensorNodeInfo #1 과 동일한 정보구조를 갖는 SensorNodeInfo #2를 추가할 수 있다.

표 2 -센서노드 관리 정보

MIB 계층						개체 이름	값의 예
9	10	11	12	13	14		
2						SensorInfo	
	1					SensorNodeInfo #1	
		1				FirmwareVer	"v1.xx_xxxx"
		2				Time	"yyyy-mm-dd-hh-mm-ss"
		3				TimeZone	"GMT+hh:mm"
		4				OrganizationCode	KSX 3267 개정안 부록 A 참조
		5				CompanyCode	KSX 3267 개정안 부록 A 참조
		6				ProductType	KSX 3267 개정안 부록 A 참조
		7				ProductCode	KSX 3267 개정안 부록 A 참조
		8				ProtocolVersion	KSX 3267 개정안 부록 A 참조
		9				NetworkLinkInfo	
			1			Network #1	

			1	RS485Modbus Adapter	"0" 또는 "1" (사용 여부)
			2	NodeID	"1~255"
		2		Network #2	
			1	IP Adapter	"0" 또는 "1" (사용 여부)
			2	NetMacInfo	"XX:XX:XX:XX:XX:XX"
			3	NetIPInfo	"xxx.xxx.xxx.xxx"
			4	GWInfo	"xxx.xxx.xxx.xxx"
			5	MaskInfo	"xxx.xxx.xxx.xxx"
		N		Network #N	
10				GeoInfo	
			1	Description	"비닐하우스 2번째 등에 있는 센서" 또는 "문앞", "중간"
11				SystemNotification	
			1	PowerNotifcation	
			1	ColdPowerOnDate	"yyyy-mm-dd-hh-mm-ss"
			2	WarmPowerOnDate	"yyyy-mm-dd-hh-mm-ss"
			3	ShutDownDate	"yyyy-mm-dd-hh-mm-ss"
			4	AbnormalStartDate	"yyyy-mm-dd-hh-mm-ss"
		2		NodeStatus	
			1	NodeStatusValue	"OK (READY), ERROR"
12				MeasurementInfo	
			1	Temperature	
			1	TemperatureSensor #1	
			1	SensingValue	"-20.0 ~ 80.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		2		TemperatureSensor #2	
			1	SensingValue	"-20.0 ~ 80.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		N		TemperatureSensor #N	
			1	SensingValue	"-20.0 ~ 80.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		2		Humidity	
			1	HumiditySensor #1	
			1	SensingValue	"0.0 ~ 100.0" %
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		3		CO2	
			1	CO2 Sensor #1	
			1	SensingValue	"0.0 ~ 3,000.0" μmol/mol(PPM)
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		4		Pyranometer	

			1		Pyranometer #1	
				1	SensingValue	"0.0 ~ 2,000.0" W/m2-h
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역 1, ... N "
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			5		6	DeviceID (Map address)
					WindDirection	"Address 값"
				1	WindDirectionSensor #1	
				1	SensingValue	"0.0 ~ 360.0" ° (방향각)
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역 1, ... N "
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
			6		6	WindSpeed
				1	WindSpeedSensor #1	
				1	SensingValue	"0.0 ~ 40.0" m/s
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역 1, ... N "
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
			7		RainDetector	
				1	RainDetector #1	
				1	SensingValue	"On/OFF"
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역 1, ... N "
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
			8		Quantum	
				1	QuantumSensor #1	
				1	SensingValue	"0.0 ~ 2,000.0" μmol/m2/s
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역 1, ... N "
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
			9		SoilMoisture	
				1	SoilMoistureSensor #1	
				1	SensingValue	"0.0 ~ 50.0" %.
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역 1, ... N "
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
			10		Tensiometer	
				1	Tensiometer #1	
				1	SensingValue	"0.0 ~ 100.0" kPa
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역 1, ... N "
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
			11		EC	
				1	ECSensor #1	
				1	SensingValue	"0.0 ~ 10.0" dS/m

			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
		12	6	DeviceID (Map address)	"Address 값"
				PH	
		1		PhSensor #1	
			1	SensingValue	"2.0 ~ 12.0" pH
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		13		SoilTemperature	
		1		TemperatureSensor #1	
			1	SensingValue	"-20.0 ~ 80.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		14		SoilMoisture	
		1		MoistureSensor #1	
			1	SensingValue	"0.0 ~ 100.0" %
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		15		DewPoint	
		1		DewPointSensor #1	
			1	SensingValue	"-80.0 ~ 20.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		16		RainGuage	
		1		RainGuage #1	
			1	SensingValue	"xxxx" mm/h
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		17		FlowRate	
		1		FlowRateSensor #1	
			1	SensingValue	"xxxx" G(kgf/s) 또는 Q(m3/s)
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역 1, ... N "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
		18		Voltage	
		1		VoltageSensor #1	
			1	SensingValue	"xxxx" V
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"

				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역 1, ... N "
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
	N				SensorNodeInfo #N	

### 3.3) 온실통합제어기 구동기노드 관리 정보 (actuatorInfo)

시설원에 분야에서 사용되는 온실통합제어기가 관리하는 구동기노드에 대한 정보 구조는 다음 표 3과 같이 구성된다. 온실통합제어기가 관리하는 구동기노드에 대한 정보는 구동기노드에 대한 시스템 정보, 구동기노드와의 통신을 위한 네트워크 정보, 구동기노드의 위치 정보, 시스템 알람 정보 및 제어 정보로 구성된다. 표 3은 그림 5의 9번째 계층인 actuatorInfo(3) 이후의 정보들에 대해 기술한 것이다.

사용 방법의 예로써, 설치되어 있는 관수밸브#1이 설치된 위치와 현재 동작 상태를 알고 싶다면, 각각 {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_controller(1) ActuatorInfo(3) ActuatorNodeInfo#1(1) GeoInfo(10) 위치 정보 (1)}와 {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_controller(1) ActuatorInfo(3) ActuatorNodeInfo#1(1) ControlInfo(12) 관수밸브(8) 관수밸브#1(1) 구역정보(6)}에 있는 정보를 검색할 수 있으며, 이는 각각 {1 3 6 1 4 1 55667 1 3 1 10 1}과 {1 3 6 1 4 1 55667 1 3 1 12 8 1 6}라는 형태로 표현된다.

만일 어떤 특정 구동기노드에서 동일한 성격의 다수 구동기를 관리할 경우, 해당 구동기에 대한 정보를 추가할 수 있다. 또한 하나의 온실통합제어기가 다수의 구동기노드를 포함시킬 경우 관리하고 있는 구동기노드에대한정보를 추가할 수 있다. 예를 들어, 2개의 구동기노드가 설치되어 있는 경우, 표 3에서 ActuatorNodeInfo #1 과 동일한 정보구조를 갖는 ActuatorNodeInfo #2를 추가할 수 있다.

표 3 - 구동기노드 관리 정보

MIB 계층						개체 이름	값의 예
9	10	11	12	13	14		
3						ActuatorInfo	
	1					ActuatorNodeInfo #1	
		1				FirmwareVer	"v1.xx_xxxx"
		2				Time	"yyyy-mm-dd-hh-mm-ss"
		3				TimeZone	"GMT+hh:mm"
		4				OrganizationCode	KSX 3267 개정안 부록 A 참조
		5				CompanyCode	KSX 3267 개정안 부록 A 참조
		6				ProductType	KSX 3267 개정안 부록 A 참조
		7				ProductCode	KSX 3267 개정안 부록 A 참조
		8				ProtocolVersion	KSX 3267 개정안 부록 A 참조
		9				NetworkLinkInfo	
			1			Network #1	
				1		RS485Modbus Adapter	"0" 또는 "1" (사용 여부)
				2		NodeID	"1~255"
			2			Network #2	
				1		IP Adapter	"0" 또는 "1" (사용 여부)
				2		NetMacInfo	"XX:XX:XX:XX:XX:XX"
				3		NetIPInfo	"xxx.xxx.xxx.xxx"
				4		GWInfo	"xxx.xxx.xxx.xxx"
				5		MaskInfo	"xxx.xxx.xxx.xxx"
			N			Network #N	
	10					GeoInfo	
		1				Description	"비닐하우스 2번째 동"
	11					SystemNotification	
		1				PowerNotifcation	
			1			ColdPowerOnDate	"yyyy-mm-dd-hh-mm-ss"
			2			WarmPowerOnDate	"yyyy-mm-dd-hh-mm-ss"
			3			ShutDownDate	"yyyy-mm-dd-hh-mm-ss"
			4			AbnormalStartDate	"yyyy-mm-dd-hh-mm-ss"
			2			NodeStatus	
			1			NodeStatusValue	"OK (READY), ERROR"
	12					ControlInfo	

		1		RoofVent	
			1	RoofVent #1	
				1	CommandTime
				2	Command
				3	OperationTime
				4	Status
				5	RemainingTime
				6	SectorDescription
				7	ActuatorStatus
				8	CommandID (Map address)
				9	StatusID (Map address)
				10	OpeningAngle
			2	SidewallVent	
				1	SidewallVent #1
				1	CommandTime
				2	Command
				3	OperationTime
				4	Status
				5	RemainingTime
				6	SectorDescription
				7	ActuatorStatus
				8	CommandID (Map address)
				9	StatusID (Map address)
				10	OpeningAngle
			3	InsulationCover	
				1	InsulationCover #1
				1	CommandTime
				2	Command
				3	OperationTime
				4	Status
				5	RemainingTime
				6	SectorDescription
				7	ActuatorStatus
				8	CommandID (Map address)
				9	StatusID (Map address)
				10	OpeningAngle
			4	ShadingScreen	
				1	ShadingScreen #1
				1	CommandTime
				2	Command
				3	OperationTime
				4	Status
				5	RemainingTime
				6	SectorDescription
				7	ActuatorStatus
				8	CommandID (Map address)
				9	StatusID (Map address)
				10	OpeningAngle
			5	Ventilator	
				1	Ventilator #1
				1	CommandTime
				2	Command
				3	OperationTime
				4	Status
				5	RemainingTime
				6	SectorDescription
				7	ActuatorStatus
				8	CommandID (Map address)
				9	StatusID (Map address)
			6	CirculationFan	
				1	CirculationFan #1
				1	CommandTime
				2	Command
				3	OperationTime

				4	Status	"On, Off"
				5	RemainingTime	"hh-mm-ss"
				6	SectorDescription	"구역 1, ... N "
				7	ActuatorStatus	"OK (READY), ERROR"
				8	CommandID (Map address)	"Address 값"
				9	StatusID (Map address)	"Address 값"
		7			IrrigationMotor	
			1		IrrigationMotor #1	
				1	CommandTime	"yyyy-mm-dd-hh-mm-ss"
				2	Command	"On, Off"
				3	OperationTime	"hh-mm-ss"
				4	Status	"On, Off"
				5	RemainingTime	"hh-mm-ss"
				6	SectorDescription	"구역 1, ... N "
				7	ActuatorStatus	"OK (READY), ERROR"
				8	CommandID (Map address)	"Address 값"
				9	StatusID (Map address)	"Address 값"
		8			IrrigationValve	
			1		IrrigationValve #1	
				1	CommandTime	"yyyy-mm-dd-hh-mm-ss"
				2	Command	"On, Off"
				3	OperationTime	"hh-mm-ss"
				4	Status	"On, Off"
				5	RemainingTime	"hh-mm-ss"
				6	SectorDescription	"구역 1, ... N "
				7	ActuatorStatus	"OK (READY), ERROR"
				8	CommandID (Map address)	"Address 값"
				9	StatusID (Map address)	"Address 값"
		9			CoolingAndHeater	
			1		CoolingAndHeater #1	
				1	CommandTime	"yyyy-mm-dd-hh-mm-ss"
				2	Command	"On, Off"
				3	OperationTime	"hh-mm-ss"
				4	Status	"On, Off"
				5	RemainingTime	"hh-mm-ss"
				6	SectorDescription	"구역 1, ... N "
				7	ActuatorStatus	"OK (READY), ERROR"
				8	CommandID (Map address)	"Address 값"
				9	StatusID (Map address)	"Address 값"
	N				ActuatorNodeInfo #N	

#### 4) 양액기 MIB 정보체계

다음 그림 6은 해당 표준에서 제안하고자 하는 양액기 내부의 MIB 정보 체계이다. 해당 표준에서 다루는 양액기의 MIB는 그림 6에서 점선으로 표시한 부분에 해당된다.

양액기 MIB 정보체계를 설명하기 위한 예로써, 임의의 제조사에서 개발한 양액기는 sf\_xxx(55667)의 sf\_NutrientSupplier (2) 항목을 정의한다. 이 정의는 MIB 트리 내에서 {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_NutrientSupplier (2)}의 구조를 가지며, 이는 {1 3 6 1 4 1 55667 2} 라는 OID 형태로 표현될 수 있다. 해당 표준에서 정의하는 양액기의 MIB 정보체계는 그림 3에서와 같이 세가지 그룹으로 구성된다. 첫번째는 양액기 시스템에 관한 정보(sysInfo), 두번째는 양액기에 속한 센서 또는 센서노드에 관한 정보(sensorInfo), 그리고 마지막 세번째는 양액기에 속한 구동기 또는 구동기노드에 관한 정보(actuatorInfo)이다.

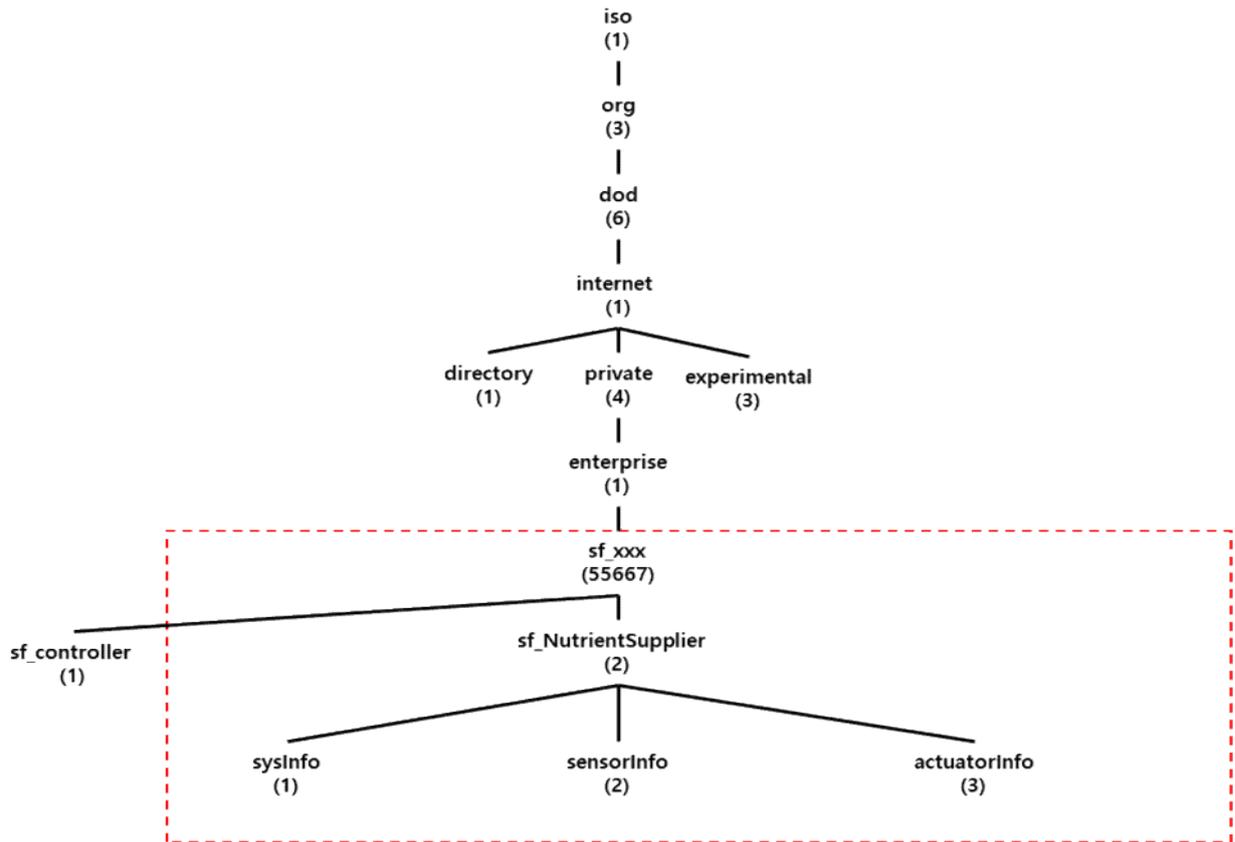


그림 6 - 양액기의 MIB 계층

#### 4.1) 양액기 시스템 정보 (sysInfo)

시설원예에서 사용되는 양액기 자체에 대한 시스템 정보 구조는 다음 표 4와 같이 정의된다. 표 4는 그림 6의 9번째 계층인 sysInfo(1) 이후의 정보들에 대해 기술한 것이다. 양액기의 시스템에 관한 주요 정보로는 시스템 모델 정보, 펌웨어, 시간 정보, 네트워크 정보, 시설 위치 정보 및 오류 정보 등으로 구성된다.

사용 방법의 예로서, 아래 표4에서 가령 양액기가 설치된 시설의 위치를 알고 싶다면, {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_NutrientSupplier (2) sysInfo(1) GeoInfo(7) 관제시설위치(1)}에 저장된 정보를 탐색하면 된다. 이는 또한 {1 3 6 1 4 1 55667 2 1 7 1}의 형태로 표현된다.

표 4 - 양액기 시스템 정보

MIB 계층					개체 이름	값의 예
9	10	11	12	13		
1					SysInfo	
	1				ModelName	"NutrientSupplier-100"
	2				FirmwareVer	"v1.xx_xxxx"
	3				Time	"yyyy-mm-dd-hh-mm-ss"
	4				TimeZone	"GMT+hh:mm"
	5				OrganizationCode	KSX 3267 준수
	6				CompanyCode	KSX 3267 준수
	7				ProductType	KSX 3267 준수
	8				ProductCode	KSX 3267 준수
	9				ProtocolVersion	KSX 3267 준수
	10				RebootInfo	'0' 또는 '1'
	11				NodeStatus	양액기상태
		1			NodeStatusValue	"OK (READY), ERROR"
		12			NetworkInfo	
		1			Network #1	
			1	1	RS485Modbus Adapter	"0" 또는 "1" (사용 여부)
					NodeID	"0"
		2			Network #2	
			2	2	IP Adapter	"0" 또는 "1" (사용 여부)

				NetMacInfo	"XX:XX:XX:XX:XX:XX"
				NetIPInfo	"xxx.xxx.xxx.xxx"
				GWInfo	"xxx.xxx.xxx.xxx"
				MaskInfo	"xxx.xxx.xxx.xxx"
		N		Network #N	"0" 또는 "1" (사용 여부)
		1	1	RS485Modbus Adapter	"1~255"
13				GeoInfo	
	1			GeoLocation	"강원도 철원군 산 XXX"
	2			Description	"단동형, 연동형"
	3			Size	"100 평"
14				SystemFaultInfo	
	1			SystemFault	
		1		PowerFault	
			1	OccurrenceTime	"yyyy-mm-dd-hh-mm-ss"
		2		HardwareFault	
			1	OccurrenceTime	"yyyy-mm-dd-hh-mm-ss"
	2			SecurityFault	
		1		NoAccessRight	
			1	OccurrenceTime	"yyyy-mm-dd-hh-mm-ss"

#### 4.2) 양액기 센서 정보 관리 (sensorInfo)

시설원예에서 사용되는 양액기가 관리하는 센서에 대한 정보 구조는 다음 표 5와 같이 구성된다. 양액기가 관리하는 센서에 대한 정보는 크게 측정정보, 측정주기, 센서의 위치 및 센서 상태 정보를 포함한다. 표 5는 그림 6의 9번째 계층인 sensorInfo(2) 이후의 정보들에 대해 기술한 것이다.

사용 방법의 예로써, 만일 설치되어 있는 첫 번째 EC 센서의 측정값을 알고 싶다면, {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_NutrientSupplier(2) sensorInfo(2) MeasurementInfo(1) EC (1) EC 센서 #1 (1) 측정값 (1)}에 저장된 정보를 탐색하면 되며, 이 정보는 {1 3 6 1 4 1 55667 1 2 2 1 1 1 1}의 형태로 표현된다. 만일 앞의 첫 번째 EC 센서의 섹터정보를 알고 싶을 경우, {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_NutrientSupplier(2) sensorInfo(2) MeasurementInfo(1) EC (1) EC 센서 #1 (1) 섹터정보 (4)}에 저장된 정보를 탐색하면 되며, 이 정보는 {1 3 6 1 4 1 55667 1 2 2 1 1 1 1}의 형태로 표현된다.

만일 동일한 성격의 다수 센서를 관리한다고 하면, 해당 센서에 대한 정보를 추가할 수 있다. 이를 위해 표 5에서는 EC에 관한 센서가 N개까지 설치되어 있는 경우를 예시하였다. 만일 추가적인 센서를 설치할 경우 온도센서 정보를 저장하기 위한 방법을 응용할 수 있다.

표 5 - 센서 정보 관리

MIB 계층					개체 이름	비고 (값의 예)
9	10	11	12	13		
2					SensorInfo	
	1				MeasurementInfo	
		1			LiquidEc	양액의 EC 정보
			1		EcSensor #1	
				1	SensingValue	"0.0 ~ 10.0" dS/m
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역1, 문앞, 온실중앙"
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
			N		EcSensor #N	
				1	SensingValue	"0.0 ~ 10.0" dS/m
				2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
				3	SensingPeriod	"hh-mm-ss"
				4	SectorDescription	"구역1, 문앞, 온실중앙"
				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"
		2			LiquidPh	양액의 PH 정보
			1		PhSensor #1	

			1	SensingValue	"2.0 ~ 12.0" pH
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
	3		6	DeviceID (Map address)	"Address 값"
				LiquidTemperature	양액의 온도
		1		TemperatureSensor #1	
			1	SensingValue	"-20.0 ~ 80.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			5	SensorStatus	"Address 값"
	4		6	DeviceID (Map address)	"Address 값"
				LiquidFlowRate	양액의 유량
		1		FlowRateSensor #1	
			1	SensingValue	"xxxx" %
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	5			TotalLiquidFlowRate	양액의 전체유량
		1		TotalFlowRateSemnsor #1	
			1	SensingValue	"xxxx" %
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	6			SoilEc	배지의 EC
		1		EcSensor #1	
			1	SensingValue	"0.0 ~ 10.0" dS/m
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	7			SoilPh	배지의 PH
		1		PhSensor #1	
			1	SensingValue	"2.0 ~ 12.0" pH
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	8			SoilTemperature	배지의 온도
		1		TemperatureSensor #1	
			1	SensingValue	"-20.0 ~ 80.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			5	SensorStatus	"Address 값"
	9		6	DeviceID (Map address)	배지의 함수율
		1		SoilMoistureSensor #1	
			1	SensingValue	"0.0 ~ 50.0" % vol.
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"

			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	10			SoilTensiometer	토양수분장력
		1		Tensiometer #1	
			1	SensingValue	"0.0 ~ 100.0" kPa
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	11			InternalTemperature	실내 온도
		1		TemperatureSensor #1	
			1	SensingValue	"-20.0 ~ 80.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1 외부 "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	12			OuterTemperature	실외 온도
		1		TemperatureSensor #1	
			1	SensingValue	"-20.0 ~ 80.0" °C
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			5	SensorStatus	"Address 값"
			6	DeviceID (Map address)	"Address 값"
	13			InternalPyranometer	실내 일사량
		1		Pyranometer #1	
			1	SensingValue	"0.0 ~ 2,000.0" W/m2
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	14			OuterPyranometer	실외 일사량
		1		Pyranometer #1	
			1	SensingValue	"0.0 ~ 2,000.0" W/m2
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1 외부 "
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	15			InternalQuantum	실내 광양자량
		1		Quantum#1	
			1	SensingValue	"0.0 ~ 2,000.0" μmol/m2/s
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1, 문앞, 온실중앙"
			5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
			6	DeviceID (Map address)	"Address 값"
	16			InternalQuantum	실외 광양자량
		1		Quantum#1	
			1	SensingValue	"0.0 ~ 2,000.0" μmol/m2/s
			2	SensingTime	"yyyy-mm-dd-hh-mm-ss"
			3	SensingPeriod	"hh-mm-ss"
			4	SectorDescription	"구역1 외부 "

				5	SensorStatus	"OK (READY), ERROR, 센서교체요망, 센서교정요망"
				6	DeviceID (Map address)	"Address 값"

#### 4.3) 양액기 구동기 정보 관리 (actuatorInfo)

시설원에 분야에서 사용되는 양액기가 관리하는 구동기에 대한 정보 구조는 다음 표 6과 같이 구성된다. 양액기가 관리하는 구동기에 대한 정보는 구동기의 상태 정보, 구동기의 섹터 정보, 시스템 알림 정보 및 각각의 구동기에 대한 제어 정보로 구성된다. 표 6은 그림 6의 9번째 계층인 actuatorInfo(3) 이후의 정보들에 대해 기술한 것이다.

사용 방법의 예로써, 설치되어 있는 관수밸브#1가 관리하는 섹터 정보와 남아 있는 동작 시간을 알고 싶다면, 각각 {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_ NutrientSupplier (2) ActuatorInfo(3) ControllInfo(1) EC(1) EC#1(1) 섹터정보 (6)}와 {iso(1) org(3) dod(6) internet(1) private(4) enterprise(1) sf\_xxx(55667) sf\_ NutrientSupplier (2) ActuatorInfo(3) ControllInfo(1) EC(1) EC#1(1) 남은동작시간(5)}에 있는 정보를 검색할 수 있으며, 이는 각각 {1 3 6 1 4 1 55667 2 3 1 1 1 6}과 {1 3 6 1 4 1 55667 2 3 1 1 1 5}라는 형태로 표현된다. 만일 어떤 특정 구동기노드에서 동일한 성격의 다수 구동기를 관리할 경우, 해당 구동기에 대한 정보를 추가할 수 있다.

표 6 - 구동기 정보 관리

MIB 계층					개체 이름	값의 예
9	10	11	12	13		
3					ActuatorInfo	
	1				ControlInfo	
		1			EC	
			1		ECValve #1	
				1	CommandTime	"yyyy-mm-dd-hh-mm-ss"
				2	Command	"Open, Close"
				3	OperationTime	"hh-mm-ss"
				4	Status	"Opening, Closing"
				5	RemainingTime	"hh-mm-ss"
				6	SectorDescription	"구역1, 문앞, 온실중앙"
				7	ActuatorStatus	"OK (READY), ERROR"
				8	CommandID (Map address)	"Address 값"
				9	StatusID (Map address)	"Address 값"
			N		ECValve #N	
				1	CommandTime	"yyyy-mm-dd-hh-mm-ss"
				2	Command	"Open, Close"
				3	OperationTime	"hh-mm-ss"
				4	Status	"Opening, Closing"
				5	RemainingTime	"hh-mm-ss"
				6	SectorDescription	"구역1, 문앞, 온실중앙"
				7	ActuatorStatus	"OK (READY), ERROR"
				8	CommandID (Map address)	"Address 값"
				9	StatusID (Map address)	"Address 값"
		2			PH	
			1		PhValve #1	
				1	CommandTime	"yyyy-mm-dd-hh-mm-ss"
				2	Command	"Open, Close"
				3	OperationTime	"hh-mm-ss"
				4	Status	"Opening, Closing"
				5	RemainingTime	"hh-mm-ss"
				6	SectorDescription	"구역1, 문앞, 온실중앙"
				7	ActuatorStatus	"OK (READY), ERROR"
				8	CommandID (Map address)	"Address 값"
				9	StatusID (Map address)	"Address 값"
		3			IrrigationMotor	
			1		IrrigationMotor #1	
				1	CommandTime	"yyyy-mm-dd-hh-mm-ss"
				2	Command	"On, Off"
				3	OperationTime	"hh-mm-ss"
				4	Status	"On, Off"
				5	RemainingTime	"hh-mm-ss"

			6	SectorDescription	"구역1, 문앞, 온실중앙"
			7	ActuatorStatus	"OK (READY), ERROR"
			8	CommandID (Map address)	"Address 값"
			9	StatusID (Map address)	"Address 값"
	4			IrrigationValve	
		1		IrrigationValve #1	
			1	CommandTime	"yyyy-mm-dd-hh-mm-ss"
			2	Command	"On, Off"
			3	OperationTime	"hh-mm-ss"
			4	Status	"On, Off"
			5	RemainingTime	"hh-mm-ss"
			6	SectorDescription	"구역1, 문앞, 온실중앙"
			7	ActuatorStatus	"OK (READY), ERROR"
			8	CommandID (Map address)	"Address 값"
			9	StatusID (Map address)	"Address 값"

나-2) [기술지원] 스마트팜 국가표준 확산을 위한 기술지원 ('20~'22)

2022년도부터 농림축산식품부에서 스마트팜 기자재간의 상호호환성 제공을 위해 국가표준 확산 사업을 적극적으로 지원한 바 있다. 확산을 꾀하는 이들 국가표준들은 본 사업을 통해 직접적 혹은 간접적으로 개발한 표준들이다. 따라서, 계획 외 연구 활동으로서, 본 사업을 통해 직접적 혹은 간접적으로 개발된 국가표준들을 확산함에 있어, 업체들의 실제 활용 가능성과 수정이 필요한 부분을 보완할 수 있도록 지원하는 역할을 수행하였다.

○ 스마트팜 국가표준 확산사업 지원

1.1.1.1.29. 스마트팜 관련 국가표준 확산을 위해 농림축산부에서 진행하고 있는 사업의 성공적인 수행을 위해, 그간 본 사업을 통해 개발하였던 KS X3267, KS X3268, KS X 3269과 농진청에서 개발한 KS X3265, KS X 3266등 총 5건에 대해 국내 기업체들의 이해도 증진과 수정 요구 사항 수렴을 위해 적극적으로 참여하였다.

1.1.1.1.30.



**스마트팜ICT기자재 국가표준 대표규격 도출 진행과정 요약**

Date	내용	비고
'20.03.27.	한국농기계공업협동조합 보조운영기관 선정 및 협약체결	
'20.04.02.	1차 전문가 회의 개최 (국가표준 5종 논의 및 호환성 제고 논의)	
'20.04.06.	현재 스마트팜 ICT기자재 시장 사용 규격 220개사 대상 전수조사 (4.6~24)	
'20.04.21.	2차 전문가 회의 개최 (조사 결과 분류 및 분석, 주요규격 도출)	
'20.05.07.	3차 전문가 회의 개최 - KS X 3265, 66 구동기 및 센서 인터페이스 주요규격 협의 및 의견수렴 - KS X 3267 등 통신분야 표준 호환성 제고 보완사항 도출 및 추가조사	
'20.05.14.	4차 전문가 회의 개최 - KS X 3265, 66 구동기 및 센서 인터페이스 주요규격 협의 및 대표규격(안) 도출 - KS X 3267 등 통신분야 표준 업계의견 수렴 및 논의	
'20.05.28.	5차 전문가 회의 개최 (KS X 3267 관련 업계의견 반영 및 대표규격(안) 도출)	
'20.06.03.	공청회 개최 (시장대표규격 산업계 의견 수렴 및 지원사업 안내)	
'20.06.09.	스마트팜ICT 기자재 국가표준 확산 지원사업 참여업체 모집 공고 시작	
'20.06.12.	컨설턴트 워크숍 개최 (공청회 산업계 의견 반영, 컨설팅 교육 및 비밀유지 서약 등)	
'20.07.09.	2차 오리엔테이션 개최, 컨설팅 및 협약체결 진행	

1.1.1.1.31.

1.1.1.1.32.



## 국가표준 적용을 위한 시설원에 구동기 및 센서 규격 조사표

한국농기계공업협동조합이 지난 3월 정부의 「스마트팜 ICT 기자재 국가표준 확산지원」 사업의 주관기관으로 선정되어 국가표준(KS)을 적용한 시제품을 개발할 예정입니다.

우리 협회는 정부연구기관과 기업 전문가 등과 함께 동사업에 참여하여 스마트팜 표준화가 효율적으로 정착될 수 있도록 주관기관과 적극 협조해나갈 계획입니다.

우선 기 제정되어 있는 시설원에 국가표준(5개)에 대해 설문조사와 공청회를 개최하여 기업의 의견을 듣고 현장 적용이 가능한 표준으로 정리하고, 참여 희망기업들을 대상으로 컨설팅을 실시하여 컨설팅 결과를 토대로 시제품을 개발 할 계획입니다. 기업들의 어려움을 고려하여 일정규격의 개발비용과 검증비용을 지원할 것입니다.

이번에 작성해 주시는 자료는 시장에 주로 사용되는 규격을 파악하고 국가표준의 제·개정에 기업참여를 유도하고 시제품 개발단계에 소중하게 활용될 것으로 생각합니다. 기업체 여러분의 적극적인 참여를 부탁드립니다.

<사업 프로세스>

시장 규격 조사 ▶ 기업대상 공청회 ▶ 국가표준 제·개정 ▶ 컨설팅 & 시제품 제작 (대상 기업 의견 수렴)

### 1. 응답회사 일반사항

가. 회 사 명 :

나. 설 립 연 도 :

다. 주 사업분야 :

라. 종 업 원 수 :

마. 농가공급실적(최근 3년 이내) :

바. 응답자: (성명/직책) \_\_\_\_\_ 연락처: \_\_\_\_\_

※ 제어시스템(PC, 통합제어기, 제어노드, 센서, 센서노드 등)의 블럭다이어그램 포함

- 1 -

그림 1 - 국가표준 적용을 위한 시설원에 구동기 및 센서 규격 조사표

구동기 인터페이스 국가표준(KSX3265)과 관련한 주요 조사 내용으로는, 업체에서 개발한 시스템에 대한 모델별로 제어 방식과 접속 단자에 대한 현황을 조사하기 위한 목적이다.



RS485기반 모드버스 인터페이스 표준(KS X 3267)은 센서 또는 구동기 노드와 제어시스템 간의 통신 규격(혹은 프로토콜)을 정의하기 때문에, 이 프로토콜을 지원하는지에 대한 여부를 조사하는 문진표를 다음과 같이 작성하였다.

RS485 MODBUS application protocol에 정의된 모드버스 표준을 사용하는가?	<input type="checkbox"/> 예 <input type="checkbox"/> 아니오 <input type="checkbox"/> 기타( )
--	--

구동기 메타데이터 규격(KS X 3268)에 관해서는, 구동기의 항목에 대해 정의된 메타데이터를 만족시키는지 여부에 대한 문진표를 아래와 같이 작성하였다.

구동기 종류	항 목	세부 내용
천창	구동기 타입	<input type="checkbox"/> 스위치형 구동기 <input type="checkbox"/> 개폐형 구동기
	상태정보 제공여부 (복수체크)	<input type="checkbox"/> 동작 상태(Operation Status)- 필수 <input type="checkbox"/> 동작 기간(Operation Duration)-옵션 <input type="checkbox"/> 동작 시간(Operation Time)-옵션
	제어정보 제공여부 (복수체크)	<input type="checkbox"/> 동작 명령(Operation Command)- 필수 <input type="checkbox"/> 동작 기간(Operation Duration)-옵션 <input type="checkbox"/> 동작 시간(Operation Time)-옵션

앞에서와 마찬가지로 센서 메타데이터 규격(KS X 3269)에 관해서는, 센서 항목에 대해 정의된 메타데이터를 만족시키는지 여부에 대한 문진표를 아래와 같이 작성하였다.

센서 종류	항 목	세부 내용	비고	
온도	데이터의 범위와 타입	데이터 측정범위		
		데이터 측정단위		
		데이터 타입	예) 실수형, 정수형	
	제공 가능한 센서 정보	DeviceID(필수)	<input type="checkbox"/> 여 <input type="checkbox"/> 부	센서구분 식별자
		DeviceClass(필수)	<input type="checkbox"/> 여 <input type="checkbox"/> 부	
		DeviceType(필수)	<input type="checkbox"/> 여 <input type="checkbox"/> 부	
		Description(옵션)	<input type="checkbox"/> 여 <input type="checkbox"/> 부	센서에 대한 설명
		Location(옵션)	<input type="checkbox"/> 여 <input type="checkbox"/> 부	센서 설치위치 기술
		Target(옵션)	<input type="checkbox"/> 여 <input type="checkbox"/> 부	센싱하고자 하는 대상
		ValueUnit(옵션)	<input type="checkbox"/> 여 <input type="checkbox"/> 부	
	제공 가능한 센싱상태 정보	DeviceID(필수)	<input type="checkbox"/> 여 <input type="checkbox"/> 부	센서구분 식별자
SensingValue(필수)		<input type="checkbox"/> 여 <input type="checkbox"/> 부	센싱 값	

본 사업에서는 이들 조사표에 담겨질 항목 선정과 항목 검토를 진행하였으며, 실제 업체들에 대한 조사는 한국농기계협동조합을 통해 이루어졌다.

#### ○ RS485 Modbus 통신 인터페이스 (KSX 3267) 개선을 위한 조사표 작성 협업

앞선 문진표에 대한 한국농기계협동조합의 조사를 통해, 표준 확산이 어려운 것으로 지적된 KSX3267의 개선작업을 위해, 본 사업에서는 한국농기계협동조합과 협업을 통해, 다음 그림과 같이 RS485 Modbus 통신 인터페이스 (KSX 3267) 개선을 위한 조사표를 공동으로 작성하였다.

## 국가표준 적용을 위한 RS485 Modbus RTU 기반 스마트팜 구동기노드 및 센서노드 규격 조사표 R1

한국농기계공업협동조합이 지난 3월 정부의 「스마트팜 IT 기자재 국가표준 확산지원」 사업의 주관기관으로 선정되어 국가표준(KS)을 적용한 시제품을 개발할 예정입니다.

우리 협회는 정부연구기관과 기업 전문가 등과 함께 동사업에 참여하여 스마트팜 표준화가 효율적으로 정착될 수 있도록 주관기관과 적극 협조해나갈 계획입니다.

기 선정되어 있는 시설원에 국가표준(KSX 3267 2018 : 스마트온실센서구동기노드 및 온실통합제어기간RS485기반모드버스 인터페이스)에 대해 국내 제조 현황을 설문조사 하여 기업의 의견을 듣고 현장 적용이 가능한 표준으로 정리 할 계획입니다.

이번에 작성해 주시는 자료는 시장에 주로 사용되는 규격을 파악하고 국가표준의 재·개정에 기업참여를 유도하고 시제품 개발단계에 소중하게 활용될 것으로 생각합니다. 기업체 여러분의 적극적인 참여를 부탁드립니다.

### 0. 응답회사 일반사항

가. 회 사 명 :

나. 설 립 연 도 :

다. 주 사업분야 :

라. 종 업 원 수 :

마. 능가공급실적(최근 3년 이내) :

바. 응답자: (성명/직책) \_\_\_\_\_ 연락처: \_\_\_\_\_

그림 2 - KSX3267 개선을 위해 공동으로 작성한 조사표

이 조사표에서는 국내 스마트팜 기자재 업체들의 RS485 Modbus RTU 프로토콜에 대한 이해도를 증가시키기 위해, RS485 Modbus RTU 프로토콜에 대한 간략한 개념과 함께, 이 프로토콜을 사용하기 위한 업체의 구현 현황을 아래와 같이 질의하였다.

1) KSX3267에서 사용하는 통신 프로토콜과 RS485 Modbus RTU 프로토콜 (<http://www.modbus.org>)과의 부합성 검토 (메시지 구조, address, function code, 에러 코드 등)

항목	질문	응답																				
ID	<p>센서노드, 구동기노드 및 통합제어기를 식별하기 위하여, 1바이트 식별자를 임의로 정할 수 있도록 제품을 수정할 수 있다. *) 예를 들어, 구현 방법에 따라 1바이트 주소를 dip switch로 설정 가능 (이는 표준 내용은 아님)</p>	<input type="checkbox"/> 가능 <input type="checkbox"/> 불가능 (                    )																				
기능 코드	<p>KSX3267과 RS485 Modbus RTU에서 정의하는 다음의 3가지 기능 코드를 지원하도록 수정할 수 있다.</p> <ul style="list-style-type: none"> <li>- Read Holding Register (0x03)</li> <li>- Write Holding Register (0x06)</li> <li>- Write Multiple Registers (0x10)</li> </ul>	<input type="checkbox"/> 가능 <input type="checkbox"/> 불가능 (                    )																				
오류 코드	<p>KSX3267과 RS485 Modbus RTU 정의에서와 같이 1) 오류를 위한 기능코드와 2) 오류에 대한 오류코드를 사용하도록 제품 수정이 가능하다.</p> <p>기능 코드: 마스터에서 수신한 기능 코드 값에 0x80 추가하여 수정</p> <ul style="list-style-type: none"> <li>- Read Holding Register Error (0x83)</li> <li>- Write Holding Register Error (0x86)</li> <li>- Write Multiple Registers Error (0x90)</li> </ul> <p>오류코드:</p> <ul style="list-style-type: none"> <li>- 0x01(illegal function): 제품에서 지원되지 않는 기능 코드</li> <li>- 0x02(illegal data address): 유효하지 않은 주소를 접근하고자 할 때</li> <li>- 0x03(illegal data value): 지원되지 않는 데이터 값으로 지정하고자 할 때</li> <li>- 0x04(slave device failure): 디바이스 문제 발생</li> </ul>	<input type="checkbox"/> 가능 <input type="checkbox"/> 불가능 (                    )																				
데이터 확보 기능 (GetData)	<p>KSX 3267에서 정의하는 RS485 Modbus RTU 패킷 포맷을 따라, 제품 수정이 가능하다.</p> <p style="text-align: center;">표 10 - 요청 패킷의 구조</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>기능 코드</th> <th>시작 주소</th> <th>레지스터 수</th> </tr> </thead> <tbody> <tr> <td>1바이트</td> <td>1바이트</td> <td>2바이트</td> </tr> </tbody> </table> <p style="text-align: center;">표 11 - 응답 패킷의 구조</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>기능 코드</th> <th>바이트 수</th> <th>레지스터 값</th> </tr> </thead> <tbody> <tr> <td>1바이트</td> <td>1바이트</td> <td>2*N바이트(N: quantity of registers)</td> </tr> </tbody> </table> <p style="text-align: center;">표 9 - 예외 응답 패킷의 구조</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>슬레이브 주소</th> <th>기능 코드</th> <th>에러 코드</th> <th>CRC</th> </tr> </thead> <tbody> <tr> <td>1바이트</td> <td>1바이트</td> <td>N바이트</td> <td>2바이트</td> </tr> </tbody> </table> <p>*) 참고 (RS485 Modbus RTU 표준)            *) 요청패킷에서 시작주소가 "1바이트"로 되어 있는 것은 "2바이트"로 수정 필요            *) 예외 응답의 경우, KSX3267에서 에러코드가 N바이트로 되어 있는 것은 오타로 보임</p>	기능 코드	시작 주소	레지스터 수	1바이트	1바이트	2바이트	기능 코드	바이트 수	레지스터 값	1바이트	1바이트	2*N바이트(N: quantity of registers)	슬레이브 주소	기능 코드	에러 코드	CRC	1바이트	1바이트	N바이트	2바이트	<input type="checkbox"/> 가능 <input type="checkbox"/> 불가능 (                    )
기능 코드	시작 주소	레지스터 수																				
1바이트	1바이트	2바이트																				
기능 코드	바이트 수	레지스터 값																				
1바이트	1바이트	2*N바이트(N: quantity of registers)																				
슬레이브 주소	기능 코드	에러 코드	CRC																			
1바이트	1바이트	N바이트	2바이트																			



항목	질문	응답																														
	<p><b>Request</b></p> <table border="1"> <tr><td>Function code</td><td>1 Byte</td><td>0x10</td></tr> <tr><td>Starting Address</td><td>2 Bytes</td><td>0x0000 to 0xFFFF</td></tr> <tr><td>Quantity of Registers</td><td>2 Bytes</td><td>0x0001 to 0x007B</td></tr> <tr><td>Byte Count</td><td>1 Byte</td><td>2 x N*</td></tr> <tr><td>Registers Value</td><td>N* x 2 Bytes</td><td>value</td></tr> </table> <p>*N = Quantity of Registers</p> <p><b>Response</b></p> <table border="1"> <tr><td>Function code</td><td>1 Byte</td><td>0x10</td></tr> <tr><td>Starting Address</td><td>2 Bytes</td><td>0x0000 to 0xFFFF</td></tr> <tr><td>Quantity of Registers</td><td>2 Bytes</td><td>1 to 123 (0x7B)</td></tr> </table> <p><b>Error</b></p> <table border="1"> <tr><td>Error code</td><td>1 Byte</td><td>0x90</td></tr> <tr><td>Exception code</td><td>1 Byte</td><td>01 or 02 or 03 or 04</td></tr> </table>	Function code	1 Byte	0x10	Starting Address	2 Bytes	0x0000 to 0xFFFF	Quantity of Registers	2 Bytes	0x0001 to 0x007B	Byte Count	1 Byte	2 x N*	Registers Value	N* x 2 Bytes	value	Function code	1 Byte	0x10	Starting Address	2 Bytes	0x0000 to 0xFFFF	Quantity of Registers	2 Bytes	1 to 123 (0x7B)	Error code	1 Byte	0x90	Exception code	1 Byte	01 or 02 or 03 or 04	
Function code	1 Byte	0x10																														
Starting Address	2 Bytes	0x0000 to 0xFFFF																														
Quantity of Registers	2 Bytes	0x0001 to 0x007B																														
Byte Count	1 Byte	2 x N*																														
Registers Value	N* x 2 Bytes	value																														
Function code	1 Byte	0x10																														
Starting Address	2 Bytes	0x0000 to 0xFFFF																														
Quantity of Registers	2 Bytes	1 to 123 (0x7B)																														
Error code	1 Byte	0x90																														
Exception code	1 Byte	01 or 02 or 03 or 04																														

2) KSX3267에서 센싱 값, 구동기 명령을 제공하기 위해 각각 4바이트의 float, 10바이트의 구조체를 사용하고 있는데, 이를 2바이트로 바꾸는 것은 불가능한지?

항목	질문	응답																																																								
MODBUS RTU에 기록되는 센싱값, 길이 변환	<p>KSX3267에서 센싱 값을 제공하기 위해 4바이트의 float를 사용하고 있는데, 이를 모두 통일된 2바이트 형태로 바꿀 수 있다.</p> <p>*) 예외 응답의 경우, KSX3267에서 에러코드가 N바이트로 되어 있는 것은 오타로 보임</p> <table border="1"> <thead> <tr> <th>번호</th> <th>분류</th> <th>현행</th> <th>수정안</th> </tr> </thead> <tbody> <tr><td>1</td><td>온도 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>2</td><td>습도 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>3</td><td>CO2센서</td><td>float</td><td>2byte word</td></tr> <tr><td>4</td><td>일사 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>5</td><td>풍향 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>6</td><td>풍속 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>7</td><td>감우 센서</td><td>integer</td><td>2byte word</td></tr> <tr><td>8</td><td>광양자 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>9</td><td>토양 함수율 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>10</td><td>토양 수분 장력 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>11</td><td>EC 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>12</td><td>PH 센서</td><td>float</td><td>2byte word</td></tr> <tr><td>13</td><td>지온 센서</td><td>float</td><td>2byte word</td></tr> </tbody> </table> <p>*) 2바이트 워드로 전달을 하더라도, 제어기에서 이를 float/string 형태로 변환할 수 있다. 다만, KSX3267 프로토콜을 기존 RS485 Modbus RTU와 align을 하기 위한 목적임</p>	번호	분류	현행	수정안	1	온도 센서	float	2byte word	2	습도 센서	float	2byte word	3	CO2센서	float	2byte word	4	일사 센서	float	2byte word	5	풍향 센서	float	2byte word	6	풍속 센서	float	2byte word	7	감우 센서	integer	2byte word	8	광양자 센서	float	2byte word	9	토양 함수율 센서	float	2byte word	10	토양 수분 장력 센서	float	2byte word	11	EC 센서	float	2byte word	12	PH 센서	float	2byte word	13	지온 센서	float	2byte word	<input type="checkbox"/> 가능 <input type="checkbox"/> 불가능 ( )
번호	분류	현행	수정안																																																							
1	온도 센서	float	2byte word																																																							
2	습도 센서	float	2byte word																																																							
3	CO2센서	float	2byte word																																																							
4	일사 센서	float	2byte word																																																							
5	풍향 센서	float	2byte word																																																							
6	풍속 센서	float	2byte word																																																							
7	감우 센서	integer	2byte word																																																							
8	광양자 센서	float	2byte word																																																							
9	토양 함수율 센서	float	2byte word																																																							
10	토양 수분 장력 센서	float	2byte word																																																							
11	EC 센서	float	2byte word																																																							
12	PH 센서	float	2byte word																																																							
13	지온 센서	float	2byte word																																																							
MODBUS RTU에 기록되는 구동기값 길이 변환	<p>KSX3267에서 구동기 명령을 제공하기 위해 10바이트의 구조체를 사용하고 있는데 통일된 2바이트 형태로 바꿀 수 있다.</p> <table border="1"> <thead> <tr> <th rowspan="2">필드</th> <th rowspan="2">설명</th> <th colspan="3">현행</th> <th colspan="3">수정안</th> </tr> <tr> <th>길이</th> <th>비트</th> <th>타입</th> <th>길이</th> <th>비트</th> <th>타입</th> </tr> </thead> <tbody> <tr> <td rowspan="2">동작 상태 (Operation Status)</td> <td>구동기 동작 상태 정보</td> <td rowspan="2">2 byte</td> <td>8</td> <td>uint</td> <td rowspan="2">1 byte</td> <td>8</td> <td>char</td> </tr> <tr> <td>레지스터를 구성하기 위하여 추가된 비트</td> <td>8</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td rowspan="2">동작 시간 (Operation Time)</td> <td>동작 시간</td> <td rowspan="2">4 byte</td> <td>24</td> <td>uint</td> <td rowspan="2">1 byte</td> <td>8</td> <td>char</td> </tr> <tr> <td>레지스터를 구성하기 위하여 추가된 비트</td> <td>8</td> <td>-</td> <td>-</td> <td>-</td> </tr> </tbody> </table>	필드	설명	현행			수정안			길이	비트	타입	길이	비트	타입	동작 상태 (Operation Status)	구동기 동작 상태 정보	2 byte	8	uint	1 byte	8	char	레지스터를 구성하기 위하여 추가된 비트	8	-	-	-	동작 시간 (Operation Time)	동작 시간	4 byte	24	uint	1 byte	8	char	레지스터를 구성하기 위하여 추가된 비트	8	-	-	-	<input type="checkbox"/> 가능 <input type="checkbox"/> 불가능 ( )																
필드	설명			현행			수정안																																																			
		길이	비트	타입	길이	비트	타입																																																			
동작 상태 (Operation Status)	구동기 동작 상태 정보	2 byte	8	uint	1 byte	8	char																																																			
	레지스터를 구성하기 위하여 추가된 비트		8	-		-	-																																																			
동작 시간 (Operation Time)	동작 시간	4 byte	24	uint	1 byte	8	char																																																			
	레지스터를 구성하기 위하여 추가된 비트		8	-		-	-																																																			

3) KSX3267에서 정의하는 22종의 센서, 구동기에 대한 ID (variable name 혹은 레지스터 맵)와 구현 가능성

- 업체에서 실제 사용하는 센서, 구동기의 종류와 이를 위한 하드웨어 구조
- 하나의 노드에 동일한 다수 센서들이 존재할 때 식별 방법 및 구현 가능성 검토

항목	질문	응답																																																																				
MODBUS RTU에 기록되는 센싱값, 길이 변환	<p>KSX3267에서 정의하는 센싱 레퍼런스값 (혹은 레지스터값)을 다음과 같은 형태로 수정할 수 있다.</p> <table border="1"> <thead> <tr> <th>번호</th> <th>분류</th> <th>주소</th> <th>길이</th> </tr> </thead> <tbody> <tr> <td rowspan="4">1</td> <td>온도1</td> <td>0</td> <td>word</td> </tr> <tr> <td>온도2</td> <td>1</td> <td>word</td> </tr> <tr> <td>온도3</td> <td>2</td> <td>word</td> </tr> <tr> <td>온도4</td> <td>3</td> <td>word</td> </tr> <tr> <td rowspan="2">2</td> <td>습도1</td> <td>4</td> <td>word</td> </tr> <tr> <td>습도2</td> <td>5</td> <td>word</td> </tr> <tr> <td>3</td> <td>CO<sub>2</sub>센서</td> <td>6</td> <td>word</td> </tr> <tr> <td>4</td> <td>일사 센서</td> <td>7</td> <td>word</td> </tr> <tr> <td>5</td> <td>풍향 센서</td> <td>8</td> <td>word</td> </tr> <tr> <td>6</td> <td>풍속 센서</td> <td>9</td> <td>word</td> </tr> <tr> <td>7</td> <td>감우 센서</td> <td>10</td> <td>word</td> </tr> <tr> <td>8</td> <td>광양자 센서</td> <td>11</td> <td>word</td> </tr> <tr> <td>9</td> <td>토양 함수율 센서</td> <td>12</td> <td>word</td> </tr> <tr> <td>10</td> <td>토양 수분 장력 센서</td> <td>13</td> <td>word</td> </tr> <tr> <td>11</td> <td>EC 센서</td> <td>14</td> <td>word</td> </tr> <tr> <td>12</td> <td>PH 센서</td> <td>15</td> <td>word</td> </tr> <tr> <td>13</td> <td>지온 센서</td> <td>16</td> <td>word</td> </tr> </tbody> </table> <p>*) 사용 방법으로는, 센서노드를 구현할 때 설치되는 센서의 종류에 따라 주소를 할당한다.</p> <p>*) 무지막지한 방법으로, 시작레지스터를 0으로 하고, 100여개의 레지스터 값을 읽어오라고 요청하면, 센서노드에 설치된 모든 센서 값을 읽을 수 있음. 값이 없는 경우는 설치되지 않은 경우로 해석 가능</p> <p>*) 참고로 이와같이 정의된 주소맵은 향후 어드레스맵의 공유방식으로 활용 가능</p>	번호	분류	주소	길이	1	온도1	0	word	온도2	1	word	온도3	2	word	온도4	3	word	2	습도1	4	word	습도2	5	word	3	CO <sub>2</sub> 센서	6	word	4	일사 센서	7	word	5	풍향 센서	8	word	6	풍속 센서	9	word	7	감우 센서	10	word	8	광양자 센서	11	word	9	토양 함수율 센서	12	word	10	토양 수분 장력 센서	13	word	11	EC 센서	14	word	12	PH 센서	15	word	13	지온 센서	16	word	<p><input type="checkbox"/> 가능 <input type="checkbox"/> 불가능 ( )</p>
	번호	분류	주소	길이																																																																		
	1	온도1	0	word																																																																		
		온도2	1	word																																																																		
		온도3	2	word																																																																		
		온도4	3	word																																																																		
	2	습도1	4	word																																																																		
		습도2	5	word																																																																		
	3	CO <sub>2</sub> 센서	6	word																																																																		
	4	일사 센서	7	word																																																																		
	5	풍향 센서	8	word																																																																		
	6	풍속 센서	9	word																																																																		
	7	감우 센서	10	word																																																																		
	8	광양자 센서	11	word																																																																		
	9	토양 함수율 센서	12	word																																																																		
10	토양 수분 장력 센서	13	word																																																																			
11	EC 센서	14	word																																																																			
12	PH 센서	15	word																																																																			
13	지온 센서	16	word																																																																			
MODBUS RTU에 기록되는 센싱값, 길이 변환	<p>KSX3267에서 정의하는 구동기 레퍼런스값 (혹은 레지스터값)을 다음과 같은 형태로 수정할 수 있다.</p> <table border="1"> <thead> <tr> <th>번호</th> <th>분류</th> <th>주소</th> <th>길이</th> </tr> </thead> <tbody> <tr> <td rowspan="4">1</td> <td>천창1</td> <td>0</td> <td>word</td> </tr> <tr> <td>천창2</td> <td>1</td> <td>word</td> </tr> <tr> <td>천창3</td> <td>2</td> <td>word</td> </tr> <tr> <td>천창4</td> <td>3</td> <td>word</td> </tr> <tr> <td rowspan="2">2</td> <td>측창1</td> <td>4</td> <td>word</td> </tr> <tr> <td>측창2</td> <td>5</td> <td>word</td> </tr> <tr> <td>3</td> <td>보온 덮개</td> <td>6</td> <td>word</td> </tr> <tr> <td>4</td> <td>차광막</td> <td>7</td> <td>word</td> </tr> <tr> <td>5</td> <td>환풍기</td> <td>8</td> <td>word</td> </tr> <tr> <td>6</td> <td>유동 팬</td> <td>9</td> <td>word</td> </tr> <tr> <td>7</td> <td>관수 모터</td> <td>10</td> <td>word</td> </tr> <tr> <td>8</td> <td>관수 밸브</td> <td>11</td> <td>word</td> </tr> <tr> <td>9</td> <td>냉난방기</td> <td>12</td> <td>word</td> </tr> </tbody> </table> <p>*) 참고로 이와같이 정의된 주소맵은 향후 어드레스맵의 공유방식으로 활용 가능</p>	번호	분류	주소	길이	1	천창1	0	word	천창2	1	word	천창3	2	word	천창4	3	word	2	측창1	4	word	측창2	5	word	3	보온 덮개	6	word	4	차광막	7	word	5	환풍기	8	word	6	유동 팬	9	word	7	관수 모터	10	word	8	관수 밸브	11	word	9	냉난방기	12	word	<p><input type="checkbox"/> 가능 <input type="checkbox"/> 불가능 ( )</p>																
	번호	분류	주소	길이																																																																		
	1	천창1	0	word																																																																		
		천창2	1	word																																																																		
		천창3	2	word																																																																		
		천창4	3	word																																																																		
	2	측창1	4	word																																																																		
		측창2	5	word																																																																		
	3	보온 덮개	6	word																																																																		
	4	차광막	7	word																																																																		
	5	환풍기	8	word																																																																		
	6	유동 팬	9	word																																																																		
	7	관수 모터	10	word																																																																		
	8	관수 밸브	11	word																																																																		
	9	냉난방기	12	word																																																																		

앞에서와 마찬가지로 실제 업체들에 대한 조사는 한국농기계협동조합을 통해 이루어졌다.

○ 센서노드와 구동기노드를 위한 디폴트 레지스터맵 제안

앞선 일련의 조사를 통해, 업체들이 RS485 Modbus 프로토콜을 적용함에 있어 가장 큰 어려움으로 지적된 것이 디폴트 레지스터 맵의 부재였던 것으로 나타났다. 이러한 문제점을 해결하기 위해, 아래 그림과 같이 센서노드와 구동기 노드에 대한 각각의 디폴트 레지스터맵을 제안하였으며, 수차례의 이해당사자 회의와 공청회 등을 통해 보완되었으며, 이 디폴트 레지스터맵을 반영한 국가표준 확산사업(건설링 사업)이 진행되게 되었다.

주소 (DEC)	의미	길이 (바이트)	주소 (DEC)	장비코드	길이 (바이트)	주소 (DEC)	값	길이 (바이트)
1	기관코드	2	101	장치구분 (장치코드)	2	201	노드상태	2
2	회사코드	2	102	장치구분 (장치코드)	2	202	센서 #1	4
3	제동타입	2				203	센서 #1 상태	2
4	제동코드	2				204	센서 #1 상태	2
5	프로토콜버전	2				205	센서 #2	4
6	연결장비수	2				206	센서 #2 상태	4
						207	센서 #2 상태	2

그림 3 - KSX3267 개선을 위한 센서노드 디폴트 레지스터맵 개념

주소 (DEC)	의미	길이 (바이트)	주소 (DEC)	장비코드	길이 (바이트)	주소 (DEC)	값	길이 (바이트)
1	기관코드	2	101	102 (스위치형)	2	201	OPID#0	2
2	회사코드	2	102	112 (개폐기형)	2	202	0 (노드상태)	2
3	제동타입	2				203	OPID#1	2
4	제동코드	2				204	201 (스위치1 상태)	2
5	프로토콜버전	2				205	100초 (스위치1)	4
6	연결장비수	2				206	남은 동작 시간, Left	4
						207	OPID#2	2
						208	개폐기 1 상태	2
						209	50초 (개폐기 1)	4
						210	남은 동작 시간, Left	4

장비 상태코드	종류
OFF/STOP/READY	0
ERROR	1
작동중 (ON/WORKING)	201 스위치형
여는중 (OPENNING)	301 개폐기형
닫는중 (CLOSING)	302 개폐기형

구동기 명령코드 대상	전체	노드
OFF/STOP	0	전체
RESET	1	노드
TIMED-ON		
특정시간(초)동안 작동하기	202	스위치형
TIMED-OPEN		
특정시간(초)동안 열기	303	개폐기형
TIMED-CLOSE		
특정시간(초)동안 닫기	304	개폐기형

주소 (DEC)	값	길이 (바이트)
501	OPID#0	2
502	0 (노드상태)	2
503	202 (스위치1 명령)	2
504	OPID#1	2
505	200초 (duration)	4
506	303 (개폐기 1 명령)	2
507	OPID#2	2
508	100 초 (Duration)	4
509		
510		

그림 4 - KSX3267 개선을 위한 구동기노드 디폴트 레지스터맵 개념

## 다) 스마트팜 국내/국제 표준화 전략수립을 위한 표준화 전략맵 개발

### 다-1) [표준화 전략맵] 스마트팜 국내·국제 표준화 전략 개발

#### ○ 추진전략

1.1.1.1.32.1. 스마트팜 기술은 4차 산업혁명의 주요한 융합기술로서 생산성 증대를 통한 농가의 소득 증대와 식량 안보를 동시에 제공할 수 있는 주요한 기술이지만, 스마트팜 관련 사업체들의 영세성과 융합 표준의 부족으로 인한 활성화가 어려운 실정이다. 세계 식량 안보에 대한 중요성이 주목받음에 따라 국내에서도 스마트팜 확산을 위한 정책을 공격적으로 펼치고 있다. 스마트팜 기술은 농축수산물의 생산성 향상과 효율적인 유통물류를 위해 정보통신기술(ICT)을 접목한 융합기술로, 해외 기술 대비 국내 스마트팜 기술 경쟁력이 비약적으로 발전됨에 따라, 해외 시장 진출에 관한 관심이 급증하고 있다. 국내 시장 활성화와 해외 시장 진출을 위한 스마트팜 관련 표준화의 중요성이 주목받음에 따라, 기술 개발과 함께 국내외 관련 표준 경쟁력 강화를 위한 지속/확산공략으로 추진한다.

본 사업을 통해 개발된 국제표준화 전략맵은, 다음 분야에 대한 필수 표준화 항목, 표준화 대상기구, 표준 및 기술 동향에 대해 분석을 수행하였다.

- 시설원예 분야 : 환경 관리 기술
- 노지과수 분야 : 자동화기기(로봇) 기술
- 축산 분야 : 환경 관리, 사양관리, 동물복지 인증 기술
- 수산양식 분야 : 환경 관리, 데이터 및 서비스 기술
- 유통·물류 분야 : 이력, 유통·물류 및 소비 관련 데이터 기술



그림 1 - 스마트팜 표준화 전략맵 대상 분야 개요도

#### ○ 표준화 목표 국가표준 적용을 위한 시설원예 구동기 및 센서 규격 조사표

1.1.1.1.32.2. 스마트팜 관련 표준화의 목표는 시설원예, 노지과수, 축산, 수산양식 분야에 대하여 ① 관제를 위한 ICT 기자재들간의 상호 호환성 확보, ② 데이터 수집과 제공, ③ 자동화 기술, ④ 방재 및 방역, ⑤ 유통과 관련한 표준화 기술들을 목표로 한다.

1.1.1.1.32.3. ITU-T, ISO와 같은 국제표준기구에서 국제 표준화를 추진하고 있으며, oneM2M, OCF, W3C, GS1과 같은 사실 표준기구에서 일부 표준을 진행 중에 있으며, 국내의 경우, TTA 스마트농업 PG(PG426), 스마트팜 표준화 포럼 등을 통해 국내 이해 당사자들의 의견을 수렴하고 관련 제품에의 표준화를 목표로 하고 있다.

1.1.1.1.32.4.

- 1.1.1.1.32.5.
- 1.1.1.1.32.6.
- 1.1.1.1.32.7.

표 1 - 스마트팜 표준화 추진 전략 개요

구분	주요내용
~ 2021년	<ul style="list-style-type: none"> <li>- 스마트온실, 노지과수, 축산, 수산양식 환경관리 프레임워크 표준 기술 확보 추진</li> <li>· ITU-T SG20, ISO TC23 등에서 관련 표준화 추진</li> <li>- 환경관리를 위한 데이터 모델링과 빅데이터 수집 프레임워크</li> <li>- 동물복지 요구사항</li> <li>- GS1 식별체계 및 GS1 블록체인</li> <li>- 로봇 통신을 위한 인터페이스에 관한 표준 개발을 착수할 목표로 추진</li> </ul>
~ 2023년	<ul style="list-style-type: none"> <li>- 스마트온실 환경관리를 위한 데이터 모델링과 빅데이터 수집 프레임워크, 동물복지 인증 인터페이스와 프레임워크에 관한 표준 기술 확보 추진</li> <li>- 클라우드 기반 스마트팜 환경관리 연동 요구사항, IoT 기반의 스마트축사 요구사항 및 환경관리 표준, 자율농기계 통신규격에 관한 표준 개발 착수</li> </ul>
~ 2025년	<ul style="list-style-type: none"> <li>- 클라우드 기반 스마트팜 환경관리 연동 요구사항</li> <li>- IoT 기반의 스마트축사 요구사항 및 환경관리</li> <li>- GS1 식별체계와 GS1 블록체인, 로봇 통신을 위한 인터페이스</li> <li>- 자율농기계 통신규격에 관한 표준 기술 확보 추진</li> </ul>

○ 표준화 기대효과

- 1.1.1.1.32.8. 국제표준 경쟁력 강화 측면에 있어서, 스마트팜 관련 기술에 대한 국제표준화 주도권 확보를 위해 ITU-T SG13 및 ITU-T SG20을 기반으로 국제표준화를 진행 중에 있다. GS1 사실표준화기구를 통한 식별체제에 대한 국제표준화에 지속 기여하고 있으며, 농업용 기계류 전반에 대한 표준화를 추진하고 있는 ISO TC23과의 협력을 통한 국제표준화 추진을 기대하고 있다.
- 1.1.1.1.32.9. 중소기업 경쟁력 강화 측면에 있어서, 스마트팜 관련 지침 제공을 통해 기업들의 시장 진입을 용이하게 하며, 표준 인증을 통해 기업 제품의 경쟁력 강화시킨다. 국민행복·안전보장 측면 측면에 있어서, 고령화와 인력감소 추세인 농촌의 현안문제 해결을 통한 농민의 생활 품질을 향상함과 동시에 작물 생산량을 극대화하여, 국내 식량 자급률 높이고자 한다.

1.1.1.1.32.10.

○ 연도별 주요 표준화 현황 및 이슈

- 1.1.1.1.32.11. 다음은 스마트팜 분야에 있어서, 관련된 주요 정책, 기술개발, 국내외 표준화에 대한 현황 및 이슈를 정리한 도표이다.

1.1.1.1.32.12.

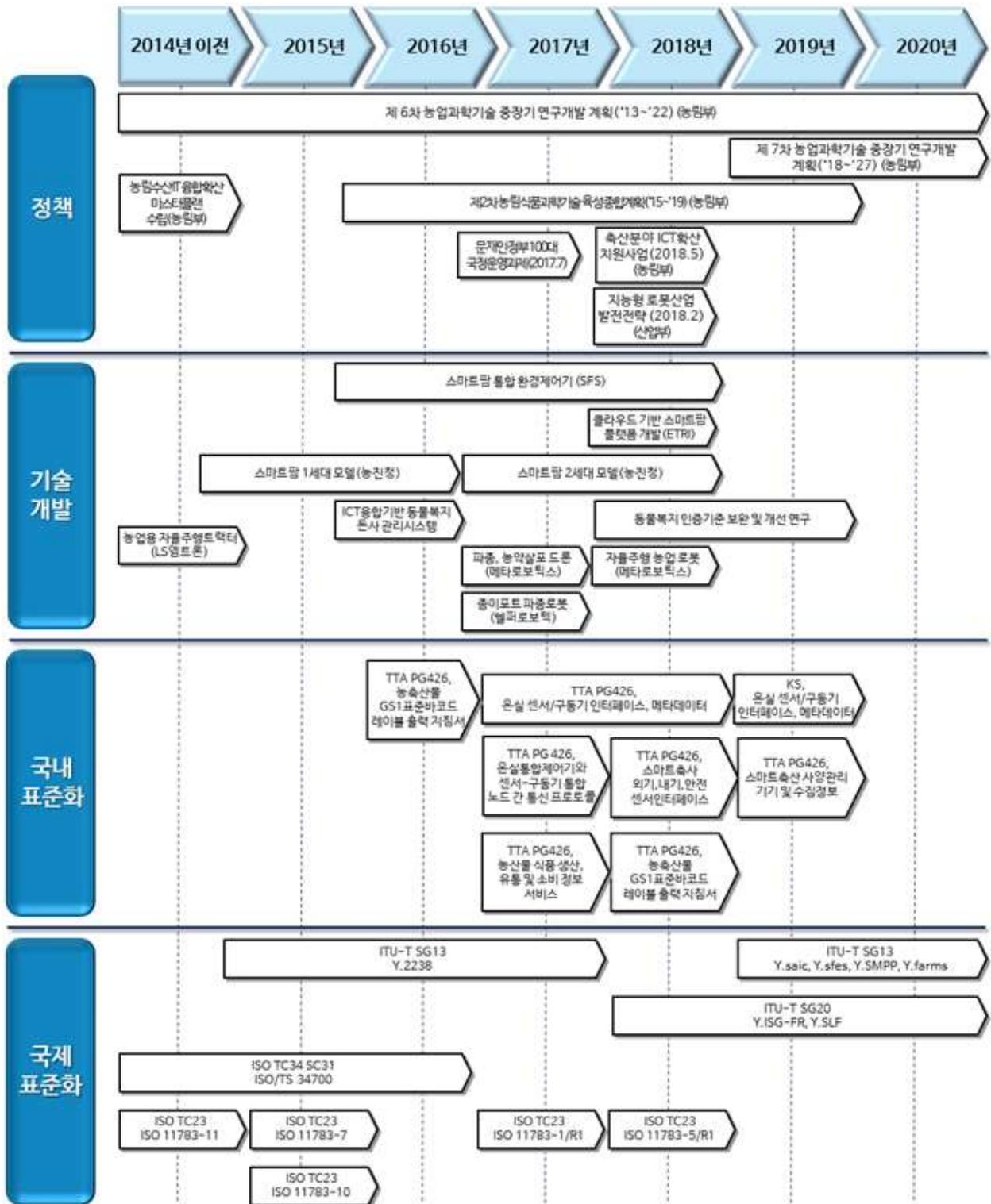


그림 2 - 연도별 주요 표준화 현황 및 이슈

1.1.1.1.32.13.

○ 표준화 항목 중분류 범위의 설정

1.1.1.1.32.14. 기존 농축수산업 분야에 ICT 기술을 적용하는 스마트팜 표준화 적용 범위가 매우 광범위할 수 있기 때문에, 국제 공적표준화기구 및 사실표준화기구에서 논의 중이거나 앞으로 논의 가능성이 높은 기술 분야와 산업적 파급효과가 높을 것으로 예상되는 적용 분야로 시설원예, 노지과수, 축산, 수산양식, 유통·물류 분야를 중분류로 선정하고, 이들 중분류 중에서 중점 기술 항목을 선정하였다.

표 2 - 스마트팜 Ver.2021 표준화 항목

표준화 항목		표준화 내용	Target SDOs	증점 항목	관련부처
시설원예	시설원예 환경 관리 표준	시설원예의 작물 생육환경을 모니터링하고 환경을 조성하기 위한 IoT 기반의 통신 및 데이터 관리 표준 - 시설원예 환경 관리를 위한 IoT 기반 기능 요소, 인터페이스, 데이터 및 프로토콜 표준 기술 - 스마트팜 환경제어를 위한 통신 기술 표준(유선, 무선, PLC 등)	ITU-T SG20, oneM2M, OCF, W3C, ISO TC23/SC18	0	과기정통부, 농림축산부
	시설원예 상호운용성 시험 및 검증 표준	시설원예 생육환경 조성을 위해 사용되는 각종 ICT 기자의 상호운용성에 대한 시험 표준 - 시설원예 환경 센서 및 구동기 시험 항목 표준 기술 - 시설원예 환경 센서 및 구동기 시험 방법 표준 기술	ITU-T SG11	X	농림축산부
	데이터 관련 표준	작물의 생육 환경, 생육 상태 등에 관한 데이터의 표현 방법과 데이터 공유를 위한 표준 - 스마트팜(시설원예) 데이터 항목 표준 기술 - 스마트팜(시설원예) 데이터 개방을 위한 클라우드·빅데이터·Open API 기반의 스마트팜 데이터 서비스 표준	ITU-T SG20	X	과기정통부, 농림축산부
	스마트팜 실내(온실, 원예) 로봇 운용 표준	스마트팜 실내 환경(온실, 원예, 축산 등)에서 필요한 로봇 작업(접목, 방제, 제초, 수확 로봇)에 대한 데이터 프레임 워크 및 통신 프로토콜, 안전성 및 성능 시험을 위한 표준 - 스마트팜 운영을 위해 사용되는 로봇과 운영시스템간의 통신 및 인터페이스 표준 기술 - 농업 로봇의 기능 요소 및 작업 상태(농자재 투입, 작업 모니터링, 생산성 등), 농업 로봇간의 데이터 통신 등에 대한 데이터 및 프로토콜 표준 기술	ITU-T SG20, JTC1 SC41	X	농식품부, 농진청, 과기정통부
	스마트팜 병해충 관리 표준	시설원예의 작물 병해충 관리를 위한 데이터 및 서비스 표준 - 시설원예 작물의 병해충 관리 요구사항, 관계 프레임워크 표준 기술	ITU-T SG13	X	과기정통부, 농림축산부
노지 과수	스마트팜 실외(과수, 발작물 등) 로봇 운용 표준	스마트팜 실외 환경(수도작, 발작물, 과수 생산)에 필요한 로봇의 농작업 데이터와 통신 프로토콜, 안전성 및 성능 시험을 위한 표준 - 운영 시스템간 통신, 인터페이스 표준 - 농작업 모니터링, 관리, 진단 및 사고 데이터 정의 표준 - 농작업 로봇간 데이터 통신 프로토콜 표준 - 자율주행 및 원격제어 지원 통신 표준 - 안전성 및 성능 시험 방법 표준	ITU-T SG20, ISO TC23	0	과기정통부, 농림축산부
	스마트팜 데이터 수집·제공 서비스 표준	스마트팜(시설원예 및 노지) 환경, 생육 및 제어와 관련된 데이터의 수집과 제공을 위한 서비스 표준 - 스마트팜(시설원예 및 노지) 생산 단계에서 수집·제공이 필요한 데이터 항목 정의 표준 - 스마트팜(시설원예 및 노지) 데이터 수집 및 활용을 위한 클라우드·빅데이터·Open API 기반의 스마트팜 데이터 서비스 표준	ITU-T SG13	X	농진청
	스마트팜 환경 관리 표준	노지의 작물 생육환경을 모니터링하고 환경을 조성하기 위한 IoT 기반의 통신 및 데이터 관리 표준 - 노지 환경 관리를 위한 IoT 기반 기능 요소, 인터페이스, 데이터 및 프로토콜 표준 기술 - 스마트팜 환경제어를 위한 통신 기술 표준(유선, 무선, PLC 등)	ITU-T SG20, oneM2M, OCF, W3C, ISO TC23/SC18	X	과기정통부, 농림축산부
	스마트팜	노지의 작물 병해충 관리를 위한 데이터 및 서비스	ITU-T SG13	X	과기정통부,

표준화 항목		표준화 내용	Target SDOs	중점 항목	관련부처
	병해충 관리 표준	표준 - 노지 작물의 병해충 관리 요구사항, 관제 프레임워크 표준 기술			농림축산부
	스마트축사 환경 관리 표준	스마트축사의 환경 및 질병 관리를 위하여 축사의 내·외부로부터 발생하는 다양한 정보를 수집, 분석, 운영 및 관리하기 위한 표준 - 축사 내·외부 환경 정보 수집 인터페이스, 데이터 분석, 관리 인터페이스 등에 대한 표준 - 동물 질병을 관리하기 위하여, 동물의 이상행동, 소리 정보, 환경정보를 수집 및 분석하기 위한 인터페이스 등의 표준	ITU-T SG20	0	과기정통부, 농림축산검역본부, 농진청
축산	축사 사양 관리 표준	스마트 축사내에서 축종별 사양관리에 필요한 기자재로부터 사양정보의 모니터링, 기자재의 제어, 서비스정보 제공을 위한 표준 - 사양관리에서 요구되는 데이터 항목 정의 표준 - 양돈, 양계, 낙동 및 한우 등에서 사용되는 급이/급수기, 생산출하, 번식 등의 장비로부터 필요한 정보를 수집하고 제어하기 위한 표준	ITU-T SG20	0	과기정통부, 농림축산검역본부, 농진청
	동물복지 인증 관리 표준	농장동물을 위한 동물복지 시스템은 데이터 수집, 연계 및 인증 등을 위한 인터페이스 표준 - 농장동물의 동물복지 관리를 위한 동물복지 농장 인증 절차, 동물복지 농장 정보 수집 및 운용 방법, 동물복지 데이터 인터페이스 등의 표준	ITU-T SG20	0	과기정통부, 농림축산검역본부
	스마트축사 데이터 수집·제공 서비스 표준(개)	축사 스마트팜 생산 단계에서의 데이터 수집·제공 방법에 대한 빅데이터 서비스 모델 표준 - 스마트축사의 생산 단계에서 수집을 필요로 하는 가축 성장, 환경 등에 대한 데이터 항목 정의 표준 - 스마트축사 데이터 수집 및 활용을 위한 클라우드·빅데이터·Open API 기반의 스마트팜 데이터 서비스 표준	ITU-T SG20	X	농정원
	상호운용성 시험 및 인증 표준	스마트축사 기자재간 상호운용성 확보를 위한 시험 요구사항, 절차 및 평가와 관련한 표준 - 스마트축사에 도입되는 센서노드, 구동기노드 등의 ICT 기자재간 상호운용성 여부를 판단하기 위한 시험 요구사항, 절차 및 평가 표준	ITU-T SG11	X	농진청, 과기정통부
	수산양식 환경 관리 표준	생산 단계에서 수집되는 양식 환경 데이터 및 제어를 위한 IoT 기반의 통신 및 데이터 관리 표준 - 양식장 내/외부 환경정보(수온, 용존산소, pH, 온도, 습도 등) 및 사육정보(먹이공급 시간, 공급량 등) 관리를 위한 인터페이스, 데이터 및 프로토콜 표준 - 양식 환경 제어(사료공급장치, 급수밸브, 산소발생장치, 히트펌프 등)를 위한 인터페이스 표준 - 양식 환경제어를 위한 통신 기술 표준(유선, 무선 등)	ITU-T SG20, ISO TC234	0	해수부, 수과원
수산 양식	수산양식 데이터 수집 및 제공 방법 등에 대한 서비스 모델 표준	종자-양성-유통 등 양식 전주기 단계에서 데이터 수집, 분석, 서비스 활용에 대한 데이터 및 API 표준 - 수산양식 데이터 정의 표준 - 수산양식 빅데이터 서비스 제공을 위한 오픈 서비스 API 표준 등 - 수산양식 디지털트윈 서비스 제공을 위한 오픈 서비스 API 표준 등 - 수산양식 에지컴퓨팅 활용을 위한 표준	GDST, ITU-T SG20	0	해수부, 수과원, 식약처
	상호운용성 시험 및 인증 표준	양식 기자재간 상호운용성 확보를 위한 시험 요구사항, 절차 및 평가와 관련한 표준 - 양식 도입되는 센서노드, 구동기노드, 제어기, 양액	ITU-T SG11	X	해수부, 수과원

표준화 항목		표준화 내용	Target SDOs	중점 항목	관련부처
		기 등의 ICT 기자재간 상호운용성 여부를 판단하기 위한 시험 요구사항, 절차 및 평가 표준 - 양식 기자재 규격 요구사항 및 시험방법 표준			
유통 물류	이력 데이터, 추적, 위조방지(블록체인) 표준	농축수산물의 신뢰성 있는 이력 데이터 수집 및 이력추적, 블록체인 기반의 위조방지 표준 - 유통 물류 이벤트 수집 및 공유/이력추적 프레임워크 표준 - 유통 물류 이벤트 위변조 방지 표준	GS1 EPCIS & CBV WG, JTC1 SC31, GDST	0	과기정통부, 산업부, 농식품부, 해수부, 식약처
	유통물류 데이터 표준	생산지부터 소비자까지의 유통물류 흐름에서 발생하는 이벤트 데이터 모델 표준 - 유통 물류 이벤트 데이터 모델 표준 - 유통 물류 제3자 인증 데이터 표준	GS1 EPCIS & CBV WG, JTC1 SC31, GDST	0	과기정통부, 산업부, 농식품부, 해수부, 식약처
	스마트 소비자 서비스 표준	온/오프라인의 생산자, 유통 물류, 소비자 간의 서비스 모델링, 등록, 검색, 접근을 위한 표준	GS1 DL & ONS	0	과기정통부, 산업부, 농식품부, 해수부
	식별 분류 체계 표준	농축수산물, 농축수산업, 물류 및 유통기업 등을 글로벌하게 식별 및 분류하기 위한 코드 체계 표준 - 농축수산물 단위별 상품 식별자 및 농축수산 관련 기업 등의 위치 및 자산 식별 표준 - 농축수산물 품종에 따른 분류 체계 표준	GS1 EPCIS & CBV WG, JTC1 SC31, GDST	X	과기정통부, 산업부, 농식품부, 해수부, 행안부, 식약처

## ○ 중점 표준화 항목 선정 이유

1.1.1.1.32.15. 스마트팜 표준화전략맵에서는 ITU-T, ISO/IEC JTC1, ISO 등 공식표준화기구와 oneM2M, OCF, W3C, GS1, GDST 등 사실표준화기구에서 논의 중이거나 앞으로 논의 가능성이 높은 기술들 중 산업적 파급효과 및 선제적 대응 가능성이 있는 항목 위주로 다음 9개의 중점 표준화 항목을 선정하였다.

- (시설원예 환경 관리 표준) ITU-T SG20에서는 식물의 생육 환경과 생육 상태를 모니터링하고 이를 기반으로 식물의 생육 환경을 조절하기 위한 시설원예와 노지의 생육환경 조절을 위한 프레임워크 표준화가 진행 중. TTA 스마트농업 PG(PG426)에서 센서 및 구동기에 대한 표준 인터페이스, 스마트팜 장치들간의 상호호환성 제공과 데이터들의 상호연동성을 제공하기 위한 스마트팜 장치들의 기능 요소, 인터페이스, 데이터 및 프로토콜 표준화가 추진되고 있으며, 국내 스마트팜 산업의 활성화와 해외 시장 진출을 위하여 국내 표준의 국제 표준화가 선결되어야하기 때문에, 이 항목을 중점 표준화 항목으로 선정
- (스마트팜 실외(과수, 발작물 등) 로봇 운용 표준) ISO TC23에서는 스마트팜 실외 로봇 운용에 관련된 시험평가(SC2), 안전 및 편의성(SC3), 방제(SC6), 관배수장비(SC18), 농업 정보통신(SC19) 표준 개발 진행 중. 특히 농기계 분야에서는 안전성과 무인자율주행에 관한 산업체의 수요가 증가하고 있고, 무인자율주행 농업기계에 대한 레벨 정의와 텔레매틱스 기술 및 이력데이터에 관한 국내 요구사항을 국제 표준화로 반영이 시급하기 때문에 이 항목을 중점 표준화 항목으로 선정
- (스마트축사 환경 관리 표준) 스마트축사 제공을 위해서는 축사의 실내외 환경 정보(온도, 습도, 이산화탄소, 암모니아, 외부기상정보 등)를 수집하고 목적에 따라 분석, 운영하는 것이 중요하지만, ITU-T SG20을 중심으로 스마트축사 서비스에 대한 표준화가 제한적으로 진행 중. 따라서, 환경 정보에 대한 효과적인 수집, 운영과 가축 질병 예방을 위한 정보 수집, 분석을 위하여 해당 인터페이스 등에 대한 표준이 시급하기에 중점 표준화 항목으로 선정
- (축사 사양 관리 표준) 양돈, 양계, 낙농 및 한우 등 축산농가의 사양관리에 필요한 급이/급수기, 생산 출하, 번식 등의 장비로부터 필요한 정보를 수집하고 제어하기 위한 프레임워크는 아직 표준이 활발히 이뤄지고 있지 않아 국내 기술을 기반으로한 다양한 통신방식, 데이터 수집 및 제어방법 등을 중심으로 하는 표준화 선점이 필요하기 때문에 이를 중점 표준화 항목으로 선정
- (동물복지 인증 관리 표준) ISO와 OIE간에 동물복지에 있어서 국제적인 기준 마련을 위해 ISO/TS 34700은 동물복지의 일반원칙, 운송, 도살, 도축, 육우, 식용우, 육용계, 유제품 생산시스템의 내용을 포함하여 표준을 진행하였고, TTA 스마트팜 PG(PG426)에서 동물복지 인증시스템과 관련된 요구사항

표준을 행하였으며, 향후 동물복지 인터페이스 표준, 세부 시스템 간의 표준과 관련하여 표준화가 추진될 예정이기 때문에 국내 기술의 국제 표준화 주도권 확보를 위해 이 항목을 중점 표준화 항목으로 선정

- (수산양식 환경 관리 표준) 수산양식 시설에서 양식 환경과 생육 상태를 모니터링하고 이를 조절하기 위해서는 양식장의 수온, 용존산소, pH, 온도, 습도 등 환경정보들을 효과적으로 수집하고 관리할 수 있는 기술과 이를 제어할 수 있는 기술이 필수적임. 아직 국제 표준화는 초기 단계이기 때문에 국내 기술을 기반으로 하는 장치들의 인터페이스, 데이터 및 프로토콜에 관한 국제 표준화 주도권 확보를 위해 본 항목을 중점 표준화 항목으로 선정
- (이력 데이터, 추적, 위조방지(블록체인) 표준) GS1에서는 농축수산물의 글로벌한 이력추적과 블록체인의 기반의 이력추적 데이터 위변조 방지를 위해서 Global Traceability 2.0 표준 패밀리와 블록체인 기반 데이터 상호호환성 표준화가 진행 중으로, 농축수산물의 생산이력 위조를 방지하기 위해서는 다양한 형태의 위변조 방지 기술들이 필수이나 표준화되지 않은 위변조 방지 기술을 사용할 경우 생산-유통물류-소비의 정보흐름이 단절될 수 있음. 먹거리 안전과 유통물류 효율성, 그리고 해외의 식품안전법 대응을 위하여 아직 국제 표준화는 초기 단계이기 때문에 국내 기술 기반으로 농산물의 이력 데이터의 위변조를 방지하기 위한 기술 표준화 추진을 통해 국내 표준의 국제 표준화가 선결을 추진하기 위해 이 항목을 중점 표준화 항목으로 선정
- (식별 분류체계, 유통물류 데이터 표준) GS1과 JTC1에서는 농축수산물의 글로벌한 유통물류 구축을 위해 필요한 상품, 박스, 팔레트, 운송수단, 농가 및 유통물류업체의 식별체계와 유통물류 데이터 표준을 위한 어휘 및 표준 포맷에 대한 표준화가 진행 중이며, Global Traceability 2.0 표준 패밀리와 블록체인 기반 데이터 상호호환성 표준화가 진행 중. 생산자부터 소비자까지의 유통물류 이력추적, 원산지 식별, 안심먹거리 서비스, 소비자 요구 정보 등을 제공하기 위해서는 농축수산물의 이력 추적이 가능한 고유 식별자, 이벤트 데이터 등 유통물류의 선진화와 빅데이터 및 AI 적용을 위해 국내 표준의 국제 표준화가 선결되어야 하나 아직 국제 표준화는 초기 단계이며, 표준화 주도권 확보를 위해 유통물류를 위한 식별 및 분류 체계, 이벤트 데이터 모델링 등을 중점 표준화 항목으로 선정
- (스마트 소비 서비스 표준) GS1에서는 농축수산물, 장소, 자산, 사람 등의 국제표준 식별자를 인덱스로 하여, 원산지, 리콜, 구매, 성분정보 등 다양한 서비스를 연결시키는 Digital Link, ONS 표준화를 진행 중. 관련 국제 표준화가 아직 시작 단계이므로 스마트 소비 서비스를 위한 서비스 모델링, 등록, 검색 및 접근 방법 등을 중점 표준화 항목으로 선정

○ 정책 현황 및 전망

표 3 - 주요 국가별 스마트팜 관련 정책 현황 및 전망

구분	주요 현황
한국	<ul style="list-style-type: none"> <li>- 해양수산부, “수산혁신 2030 계획” 발표 [2019.02.13.]</li> <li>· 지속가능한 젊은 수산업, 함께 잘사는 어촌 실현 비전 제시</li> <li>- 농식품부는 농림식품과학기술 육성 종합 계획에서 ‘ICT를 활용한 농산업 분야에 대한 첨단화 추진계획’을 수립하여 농식품 분야 미래기술 발굴 및 기술개발 투자 확대를 계획 [2019.12]</li> <li>- 농업혁신을 위한 수단으로 주목받고 있는 스마트팜의 보급 확산을 위해 ‘스마트팜 확산방안(2018)을 통해 시설원에 7,000ha, 스마트축사 5,750호의 정책목표를 제시하고 다양한 정책·제도개선을 추진 중 [2018.04]</li> <li>- 농업·농촌 및 식품산업 발전 기본계획(2018~2022)에서 걱정없이 농사짓고, 안심하고 소비하는 나라를 비전으로 제시하고 이의 달성을 위한 중점 추진과제로 산업 기반 강화, 안전한 먹거리 공급체계 구축, 농업인 소득경영 안전망 확충 등을 선정 [2018.02]</li> <li>- 산업부 “유통 산업 융합 얼라이언스” [2017.11]</li> <li>· 4차 산업혁명에 걸맞는 유통산업 혁신을 위해 5년간 170억을 투자할 계획을 발표</li> <li>· 유통산업의 혁신을 위해 AI, VR/AR, 표준화 3개의 분과로 나누어 유망 과제를 발굴을 진행 중</li> <li>- 정부는 첨단정보통신기술을 융합한 스마트농업을 지원하고 생태 친화적 미래 농업에 대한 연구개발 강화를 주요 농정공약을 제시하였으며, 문재인 정부 100대 국정과제에서 ‘지속가능한 농식품 산업 기반 조성’이라는 정책방향을 제시 [2017.07]</li> <li>· 대통령 직속 ‘4차산업혁명위원회’에서 농업 산업혁신을 위해 인공지능 기반 무인화·과학화로 친환경 정밀 농업 실현을 제시</li> <li>· 4차 산업혁명 대응 계획에서 지능화 혁신 프로젝트의 세부목표로 스마트팜 보급 확산, 농업로봇 혁신 등을 제시</li> <li>· 혁신성장 정책에서 공통 추진하는 패키지형 R&amp;D 투자 플랫폼의 우선 적용 분야로 스마트팜 분야를 선정하였으며, 현 정부 혁신성장동력 과제로 스마트팜을 후보과제로 선정하는 등 적극적인 투자확대 노력 추진</li> <li>- 5대 국정목표 중 「고르게 발전하는 지역」의 3대 국정전략에는 “사람이 돌아오는 농산어촌”이 포함 [2017.07]</li> </ul>

구분	주요 현황
	<ul style="list-style-type: none"> <li>- “사람이 돌아오는 농산어촌”의 4대 국정과제에는 ‘깨끗한 바다, 풍요로운 어장(해양수산부)’가 포함 [2017.07]</li> <li>· 깨끗한 바다, 풍요로운 어장을 조성하기 위하여 양식업의 첨단화·규모화 실시</li> <li>· 2018년 ICT 첨단양식기술 개발계획 수립, 2021년부터 스마트양식장 구축 등으로 2022년 양식 생산량 230만 톤 달성 목표</li> </ul>
미국	<ul style="list-style-type: none"> <li>- FDA는 “Smarter Food Safety 시대”의 선언을 통해, 농축수산식품의 안전성 확보를 위한 식품안전 현대화 계획 지속 [2019.4]</li> <li>· 새로운 기술과 진보된 이력추적 표준 기술을 통해, 2011년 공표된 Food Safety Modernization Act (식품안전현대화법안)을 구현</li> <li>- 4차 산업혁명에 대응하여 농무부(USDA)의 정책방향에 따라 위험관리 방향에서의 사업을 추진 [2019.02]</li> <li>· 고비용의 기반기술개발을 위한 정책이 추진 중에 있으나, 최근 민간이 농업용 로봇 분야에 진출하고 있어 공공 지출 R&amp;D는 축소</li> <li>· 자율주행 및 위치인식기술 결합 농업용 로봇에 대한 상용화는 상당히 진척되었으며, 최근 의사결정 시스템과 운송로봇에 대한 투자 진행</li> <li>- NOAA Fisheries 는 SIMP(Seafood Import Monitorng Program) 프로젝트를 통해 해외에서 수입되는 15개 어종 및 어군에 대해 이력 정보를 의무적으로 제출하게 함 [2018.01]</li> <li>- 미국은 지속되는 가뭄 및 농작물 생산 악화로 인한 기후변화에 따른 스마트 농업전략을 발표 [2017.01]</li> <li>- 미국 로보틱스 로드맵(‘16)에서는 농업분야에 대한 방향성을 제시하고 있는데, 정밀농업 및 정밀 목축업 방향을 제시 [2016.11]</li> <li>- 양식산업발전 10개년 계획, 수산양식 연구전략(‘16~’20)을 통해 지속가능한 어업을 달성과 수산양식에 대한 연구, 과학, 기술 발전을 위한 계획 추진 [2015.2]</li> <li>- 중점 투자분야에 농수산물분야 ICT 융합기술을 포함하고, 국가과학기술위원회(NSTC) 주도로 원천·상용화 기술 투자 확대 [2012.02]</li> <li>· ICT 융합 원천 기술에 대한 투자 확대(2002, 18억 달러 → 2012, 37억 달러)</li> </ul>
일본	<ul style="list-style-type: none"> <li>- 2020년까지 명목 GDP 600조 엔(円) 달성을 목표로 한 아베 정권의 2단계 성장전략 [2016.06]</li> <li>· 4차 산업혁명 시대에 대응한 일본경제의 새로운 도약을 준비하기 위해 10개 전략적 민관합동 프로젝트 추진을 선언</li> <li>- 스마트농업 확대 대안은 인공지능, 빅데이터, IoT, 로봇 등 기술의 개발 및 적용을 강조, ‘인공지능 미래농업 창조프로젝트’ 등의 시책 추진을 계획 [2016.06]</li> <li>· 궁극적으로 농가의 수익구조 개선, 신규 농업주체의 확대를 통해 일본 농업·농촌의 급속한 인구감소 및 고령화에 대응</li> <li>· 경쟁력과 지속가능성을 동시에 확보하려는 노력의 일환</li> <li>- 일본재흥전략(‘16)을 통해 ‘선제적 농림수산업 촉진과 수출 강화’를 목표로 두고 농업·농촌 경쟁력 및 지속가능성 확보를 위해 노력 [2016.05]</li> <li>- 아베 정부는 근간 국가전략특구법 개정안을 국회에 제출하고 기업 농지 소유제한을 대폭 풀어 기업형 농업프로젝트를 확대할 방침 [2016.03]</li> <li>· 전략특구지역에서 기업형 농지 소유·경작 효율성이 입증되면 기업형 농업프로젝트를 전국적으로 확대할 계획</li> <li>· 대기업들(도시바, 파나소닉, 후지쯔 등)이 반도체, 정보기술(IT) 제품을 생산했던 공장을 식물공장으로 변경하여 차세대 농업성장 사업으로 집중 개발 중</li> <li>· 수산업은 과학기술현식종합전략(‘14) 등을 통해 민관의 R&amp;D투자 확대, 제 3차 수산기본계획(‘12~’22)을 통해 수산업 회복을 위한 전략 마련</li> </ul>
유럽	<ul style="list-style-type: none"> <li>- (프랑스) 데이터기반의 영농 최적화 및 의사결정 시스템 [2019]</li> <li>- (네덜란드) 농업생산 및 유통·소비부문의 강점강화를 위한 빅데이터 분석 기반의 농업시스템 구축 연구 [2018.01]</li> <li>- (독일) 트랙터 및 무인기 활용 데이터 수집, 영상판독, 기상이변 모니터링 등을 기반으로 의사결정 지원시스템 및 최적화 투입, 병해충 예찰 관련 프로젝트 추진 [2018]</li> <li>- 디지털 시대에 대응한 2020년 이후의 공동농업정책에 대한 논의가 등장 [2017.12]</li> <li>· 디지털 기술(혁명)에 기반한 농업과학기술 연구·혁신 활동 증가는 4차 산업혁명의 여파가 농업 성장 전략에 반영하고 있다는 증거</li> <li>· 유럽집행위원회(‘16)*는 지속가능한 농업·농촌을 위한 우선순위 의제를 설정하고, 디지털 기술 및 인프라, 다분야·다국적 연구 협력을 강조</li> <li>* 유럽집행위원회(European Commission) : 유럽 통합과 관련된 조약을 수호하고 유럽연합의 행정부 역할을 담당</li> <li>- 유럽단일연구지역의 구현 및 강화를 목표로, Horizon2020 하의 실행계획(Work Programme ‘16~’17, ‘18~’20)에 적극 반영 [2016]</li> </ul>

구분	주요 현황
	<ul style="list-style-type: none"> <li>- 4차 산업혁명 대응 EU 농업 성장전략은 정보통신기술(LEIT-ICT) 등과 융복합되는 방식으로 폭넓게 반영·진개 [2016]</li> <li>- 노르웨이 등 양식 선진국들은 양식업의 첨단화·규모화, 전후방산업 육성 및 수산물 품질관리 강화 등을 통해 미래 성장 동력 발굴에 집중 [2016.10]</li> <li>- 영국, 프랑스, 독일 등은 각종 산업에 디지털 서비스를 접목한 신비즈니스 모델 창출을 통해 4차 산업혁명 대응 [2018]</li> </ul>
중국	<ul style="list-style-type: none"> <li>- 중국 국무원은 '전국 농업현대화 계획('16~'20년)'에서 농업 기술 장비와 정보화 수준 제고를 위해 IoT, 지능형설비 보급 확대 목표를 제시 [2019.12] <ul style="list-style-type: none"> <li>· 이를 위해 '20년까지 정보기술 응용 비율과 농민 인터넷 보급률을 각각 17%와 52%까지 올리고, 농촌 가정에 정보 도입을 80%이상 확대</li> </ul> </li> <li>- 중국 공업정보화부는 '중국 제조 2025' 계획에서 중점적으로 투자하는 분야로 농업기계 설비 분야에 대한 계획을 마련 [2019.01]</li> <li>- 식품 안전을 강화하기 위해 식품안전법 개정 [2015.10] <ul style="list-style-type: none"> <li>· 중국정부는 식품안전사고를 예방하기 위해 기존의 식품 위생관리조례를 강화하여 식품안전법으로 관리를 진행 중</li> <li>· 2015년 10월부터 강화된 개정안을 통해 식품의 유통이력 관리를 강화하고 있고, 점진적으로 관리방안을 보완해나갈 예정</li> </ul> </li> <li>· 중국 CFDA, GS1 CHINA, Auto-ID Labs Fudan은 장수성, 광동성을 테스트베드로 하여 식품의 생산부터 소비에 이르는 전 단계의 이력추적 및 모니터링이 가능한 체계를 구축하였으며, 향후 식품안전법의 강화 추세에 따라 17개 성으로 확대해나갈 예정</li> <li>- 공업정보화부는 농기계의 성장 방향으로 스마트화와 전체 공정의 기계화를 추진 [2015.05] <ul style="list-style-type: none"> <li>· 동시에 기술 집약형 하이테크 농업 기계의 연구 개발을 추진을 계획 중</li> </ul> </li> <li>- 농기계 분야에 대해 산업정보화부는 2020년까지 핵심 기능 부품 등의 개발 능력을 쌓고 국내 농기계 시장 점유율을 90%로 확대 [2015.05] <ul style="list-style-type: none"> <li>· 대형트랙터, 목화수확기 등 하이테크 제품 시장 점유율을 30%로 확대 목표</li> </ul> </li> <li>- 국가첨단기술연구발전계획(863 계획)에 의해 추진되는 '863 사업'을 통해 중국내 수산연구소에 연구를 지원 중에 있으며, 전 세계 첨단기술 변혁 및 경쟁에 발맞추어 중국의 첨단기술산업 발전을 촉진 [1986.03]</li> <li>- 어업혁신 현대화 계획을 통해 어업의 산업화, 첨단화, 인력양성 등을 추진 [2017.08]</li> </ul>

○ 국내 기술개발 현황 및 전망

(시설원에 환경 관리 표준) IoT 장치로부터 수집한 데이터와 AI를 접목하여 최적 생육 환경 조성 및 생산-소비를 실시간으로 연결하는 IoT 기반 스마트팜 환경관리기술 개발이 진행 중

- (농촌진흥청) 한국형 스마트팜 모델을 정의하고, 국내 업체들과의 상호연동을 위한 테스트베드를 통해 지속적인 연구를 수행하고 있으며, 한국형 스마트팜 2세대 모델의 기술 개발을 완료하고, 2019년 현장 적용을 추진 중
- (농림축산식품부) 스마트팜 ICT 기자재 국가표준의 확산을 통하여 국내 스마트팜 ICT기자재간 상호호환성을 꾀하고, 이를 통해 국산제품의 품질 향상과 유지보수의 편리성을 도모하기 위하여, 스마트팜 국가표준 확산사업을 2020년부터 추진
- (스마트팜 융합연구단) 비닐하우스 내부의 환경관리를 위한 저가형 통합제어기와 생육 센서 기술 연구 (2016~2018)
- (나래트랜드) IoT와 AI를 활용한 원격제어, 관제 기술을 개발하고, 이를 클라우드에 탑재하여 클라우드 기반 환경모니터링 및 제어시스템을 구축하여 상용화
- (유비엔) 시설원에 환경 감시와 제어를 위하여, 400MHz대역의 RF와 PoE 기술을 활용함. 또한 SaaS 기반의 스마트팜 서비스를 제공하기 위하여, 분산처리형 클라우드 스마트팜인 팜링크 개발 및 상용화
- (엘시스) 클라우드 서비스에 적용하기 위하여, oneM2M 기반의 IoT 노드 및 허브, 스마트팜 제어 등 통신 인터페이스와 물리적 가상화 중심의 기술 개발 및 상용화
- (기타) 스마트팜에서 사용되는 대부분의 센서와 제어기간 통신은 비표준화된 방식을 따르고 있지만, 온실내 일부 센서의 교체, 작물품종의 변경에 따른 센서의 추가 설치시 전체 시스템을 교체하지 않을 수 있는 규격화된 통신 시스템 개발될 것으로 전망되며, 아울러 효과적인 생산/소비 예측, 물류 및 유통 구조를 개선하기 위해 생산과 소비를 실시간으로 연결할 수 있는 통신 환경에 관한 기술 개발이 지속될 것으로 전망

(스마트팜 실외(과수, 밭작물 등) 로봇 운용 표준) 노지농업용 로봇은 트랙터, 콤팩트, 관리기 등 전통 농기계와 로봇기술의 융합을 통해 새로운 로봇 농기계 형태로 진화해 가고 있으며, 글로벌 농기계 시장에서

## 비중 확대 진행 중

- 1.1.1.1.32.16. - (국립농업과학원) 5cm이내 오차로 모 사이를 자율주행하며 잡초를 제거하는 궤도형 벼 농사 제초 로봇 개발완료. 레이저 센서만으로 과수원 자율주행 가능한 과원용 자율주행 로봇 플랫폼 개발하였으며 현재는 지능형 무인 스마트 방제기 개발 중
- 1.1.1.1.32.17. - (대동공업) 실시간 이동 측위(RTK) 기술을 활용한 자율주행 이앙기 개발 및 상용화함. 자율주행 이앙기는 직진유지, 모 간격유지, 정밀 비료 살포 기능이 탑재
- 1.1.1.1.32.18. - (LS엠트론) 5G기반의 원격제어와 무인 경작 기능 탑재한 트랙터 개발하였으며 2021년 양산이 목표
- 1.1.1.1.32.19. - (동양물산) 트랙터, 이앙기 및 기타 승용 농업기계 범용으로 사용 가능한 자동 선회 자율주행 시스템 개발완료. 현재는 KT와 자율주행 농업기계 연구 중
- 1.1.1.1.32.20. - (킨트) 농업기계로부터 수집된 데이터를 분석하고 원격제어 및 자율주행 가능한 농업기계 지능화, 무인화 서비스 플랫폼 개발하였으며 국내 농업기계 제조사들과 협업하여 트랙터용 플랫폼을 2020년 양산 완료하며 콤바인, 이앙기로 확대적용 예정. 노지 자율 작업 로봇 운영 서비스 플랫폼 개발 진행 중
- 1.1.1.1.33. **(스마트축사 환경 관리 표준)** 가축에게 최적의 사육 환경을 제공하기 위하여 축사 내 다양한 센싱을 통한 데이터 수집 및 분석을 위한 다양한 기술 개발이 진행 중이며 데이터 수집, 분석을 위한 통신 프로토콜 등의 개발 및 표준을 진행 예정
  - 1.1.1.1.33.1. - (유라이크코리아) 가축 모니터링 시스템인 ‘라이브케어’는 사물인터넷(IoT) 기반 가축 질병 모니터링 서비스로 축우에 바이오캡슐을 경구 투여해 생체 변화를 보다 정확하게 실시간으로 모니터링 가능. 수집된 데이터는 딥러닝 기술을 통해 분석되며, 구제역을 비롯한 소의 식체, 괴저성 유방염, 유행열, 일본뇌염, 폐렴 등 다양한 질병의 예방과 조기 치료가 가능하고 건강상태 외에도 발정 시기와 출산 시기까지 예측이 가능
  - 1.1.1.1.33.2. - (ETRI)에서는 ICT 기술을 활용, 스마트축사를 구축하기 위하여 축사 내·외부의 정보를 수집, 분석하여 최적의 사육환경 제공과 질병 상시 안전감시를 위한 지능형 스마트축사 요소 기술을 개발 중. 특히 디지털 트윈 개념을 활용하여 가상의 축사를 구축, 다양한 환경에 대한 시뮬레이션을 통해 최적의 사육환경을 도출하고, 가상과 실제 축사 간의 실시간 피드백을 통해 스마트축사 제공 가능하며, 카메라를 이용하여 돼지의 움직임을 실시간 모니터링 및 분석함으로써 돼지의 활동성을 바탕으로 한 건강도를 확인
- (축사 사양 관리 표준)** ERP를 활용한 축산 사양관리에서 빅데이터, 인공지능 기술을 활용한 환경관리 연계 기술개발에 초점을 맞춰 개발 및 일부 사업화를 진행중
  - (서울대) 현장중심의 환기실험과 CFD (Computational fluid dynamics) 시뮬레이션 기법을 활용하여 축산시설 내부 및 외부에서의 질병 확산 거동 및 전파 경로를 분석하고 이를 예측하는 연구를 완료 및 수행 중
  - 1.1.1.1.33.3. - ((주)이지팜) 양돈관리 프로그램인 ‘피그플랜’을 통해 웹 기반의 양돈생산경영관리 SW와 돈사 내 설치된 다양한 ICT 융합시설 장비 간의 통합 서비스를 제공하며, 양돈장의 번식돈, 육성돈을 관리하고, 방역관리를 통해 농장의 사양관리와 질병관리를 지원하며, 농장의 번식성과 함께 사료업체, 도축장 등 타 업체와 연계를 통한 사료거래내역 및 출하내역을 이용해 개별농장의 전반적인 경영상태를 분석·비교하여 해당 농가의 경영개선을 지원 중
  - (나래트랜드) 사물인터넷 및 ICT 기술을 농·축산 분야에 적용하여 원격 제어관리가 가능한 ‘반딧불이 스마트팜’을 SK텔레콤과 전국에 보급 중이며, 농장/축사/양식장 등의 시설물 및 재배작물 가축을 인공지능과 빅데이터를 이용한 원격제어 관리 운영 중
  - 1.1.1.1.33.4. - (ETRI) 디지털트윈과 AI기술을 활용하여 영상, 음성의 분석을 통한 사양관리 기술 개발 중. 특히 Triplet-eye플랫폼을 통해 딥러닝 기반의 돼지의 행동인식, 음성인식을 통한 호흡기 질병과 사양관리의 연결 등의 기술 개발을 진행 중

(동물복지 인증 관리 표준) 동물복지 인증 관리와 관련하여 농업 경영 정보 및 농어촌 정책, 동물복지를 위한 돈사 관리시스템의 기준 관련하여 연구가 진행되고 있으며, 2019년부터 관련 표준 개발을 진행 중

1.1.1.1.33.5. - (ETRI) 2018년부터 산란계농장, 양돈농장 등을 대상으로 농장동물의 동물복지 인증체계를 자동화를 위한 동물복지 인증시스템을 농림축산검역본부와 협업을 통해 개발 중

- (공주대) 2016년 'ICT융합기반 동물복지 돈사 관리시스템 개발'을 통해 해외 선진 돈사시설과 국내 시설의 비교 분석을 통한 국내 돈사환경에 적합한 자동화 관리 기준 설정 연구 진행

1.1.1.1.33.6. - (농림축산검역본부) 높은 수준의 동물복지 기준에 따라 인도적으로 동물을 사육하는 소·돼지·닭·오리농장 등에 대해 국가에서 인증하고 인증농장에서 생산되는 축산물에 '동물복지 축산농장 인증마크'를 표시하는 제도 시행 중. 2012년 산란계, 2013년 돼지를 대상으로 인증제를 도입하였고, 점차 그 대상 축종을 확대 중

1.1.1.1.33.7. - (국립축산과학원) 2018~2020년까지 '동물복지 인증기준 보완 및 개선 연구' 과제를 통해 동물복지 양돈농장 인증 농가의 시설, 환경 및 관리방법 실태조사, 동물복지 양돈 시설 및 가축관리 인증기술 개선방안 마련을 위해 연구를 진행 중

1.1.1.1.33.8. - (성우농장) 정부에서 시행하는 동물복지 인증기준이 국내현실과 맞지 않는 부분이 많아 국내현실에 적합한 동물복지 인증기준을 마련하여 사설 동물복지 인증을 준비 중

1.1.1.1.34. (수산양식 환경 관리 표준) 양식장의 무인화를 위한 기자재 개발과 양식물에 최적 양성조건을 제공하기 위해 수질센서를 이용한 데이터 수집 및 분석을 위한 기술 개발 중

1.1.1.1.34.1. - (비디수산) 국내 최초로 900MHz 대역의 SUN 무선 센서 네트워크 기술과 원거리 무선(LTE/3G)기술을 이용하여 수조 별로 수온, 용존산소량(DO), pH 센서를 설치하고 실시간으로 수조의 상태를 감시 및 양식관리 시스템. 센서 별로 상 2단계, 하 2단계 알람의 설정이 가능하고 웹과 스마트폰 앱을 통해 실시간 모니터링 및 알람기능을 제공

1.1.1.1.34.2. - (조인영어조합법인) 뱀장어RAS 양식장 수질환경 모니터링 제어시스템 구축. 국내 최초로 "제어시스템"을 상업용 양식장에 성공적으로 실현하여 운영 중인 사례로, 각 수조별로 모니터링 되는 DO에 따라서 개별적으로 산소공급/차단이 자동으로 이루어지면서, 적정산소농도를 24시간 실시간으로 운영함으로써, 리스크예방 및 생산성 향상에 기여 또한 근무하는 임직원들의 노동의 품질을 향상시키고, 복지향상에도 긍정적으로 기여한 것으로 평가

1.1.1.1.34.3. - (국립수산과학원) 2016년부터 스마트양식기술 개발을 추진하였으며, 해상 스마트양식장 플랫폼(2018년, 경남 하동)과 육상 스마트양식장 플랫폼(2019년, 경남 진해)을 구축하고 실증화 연구를 진행 중

(이력 데이터, 추적, 위조방지(블록체인) 표준) 유통 및 정보에 대한 이력추적, 위변조 방지 및 분산 원장 기술에 대해 국내기업과 대학 중심으로 개발 진행 중

1.1.1.1.34.4. - (삼성 SDS) 블록체인 플랫폼 간 연계를 지원하는 국제 무역 플랫폼 '딜리버(DELIVER)' 시스템 개발. 하이퍼레저 패브릭과 이더리움의 연계 뿐만 아니라 다른 블록체인 플랫폼과의 연계 기술도 개발 중

1.1.1.1.34.5. - (SK C&C) SKT의 사물인터넷 전용망인 LoRa를 활용해 화물의 위치정보, 온도, 습도 등 실시간 정보를 물류 관계자들에게 공유하는 서비스를 개발. 한국-상하이 구간의 컨테이너 화물을 대상으로 실험 운영 완료

1.1.1.1.34.6. - (관세청) 개인 통관 시스템에 블록체인 기술을 적용하여 2019년부터 시범 사업을 시행. 블록체인 기술을 활용해 통관 관련 정보를 관세청과 쇼핑몰, 특송업체가 공유하기 때문에 실시간 수입 신고가 가능

- (한국과학기술원) Auto-ID Labs에서 IFT(Institute of Food Technologists)와 WWF(World Wildlife Fund)가 주관하는 GDST(Global Dialogue on Seafood Traceability) 국제수산물이력추적 구축 프로젝트에 DAG(Dialogue Advsiory Group)에 참여하며, 표준화 및 시스템구축 사업을 지원

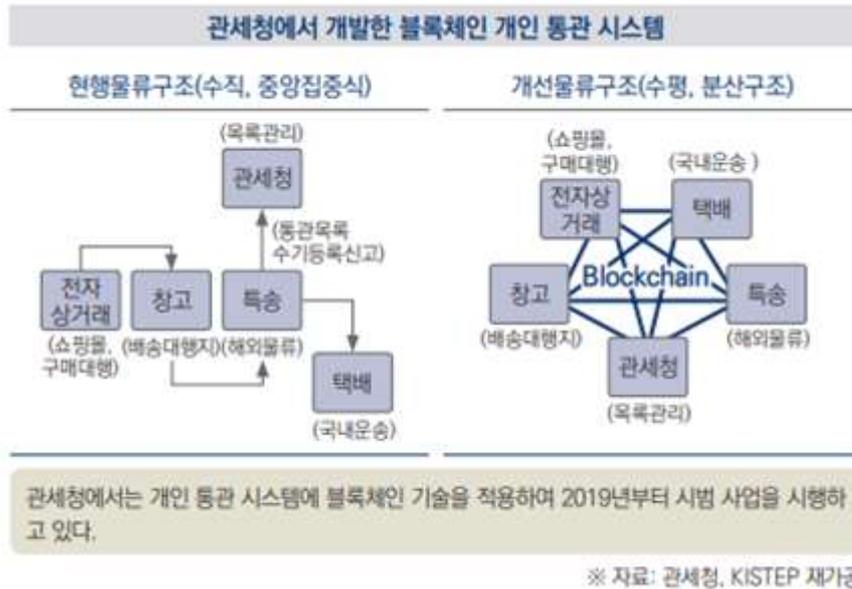


그림 3 - 블록체인기반 개인 통관 시스템

(유통물류 데이터 표준) GS1 EPCGlobal 표준 규격에 따라 생산-유통-소비에 이르는 전 생애 주기 이력 데이터를 수집 및 공유하기 위한 오픈소스 프로젝트 진행 중

- (유통물류진흥원) 유통물류산업 진흥을 위한 시장연구 및 정책개발을 추진하는 유통물류 전문기관이자 국제상품코드 표준화 기구인 GS1의 한국 대표기관으로서, GS1 국제표준식별자 및 무선인식기술(RFID/EPC) 표준을 국내에 보급 중. GS1 식별자 및 상품데이터 표준을 통해, 위해상품 판매차단 시스템, 보건의료산업, 코리안넷, 신선식품, 바코드정보제휴, 유통시장분석 정보서비스, ECR/SCM 활성화, 바코드 검증서비스를 포함하는 사업을 진행 중
- (코리안넷) 유통표준코드(바코드) 생성에서 상품정보 등록까지 원스탑으로 가능한 바코드 관리 시스템으로, 표준바코드(EAN-8/13/14, UPC-A/E)가 부착된 상품의 상세정보를 표준화시켜 데이터베이스에 등록하고, 이를 제고, 물류, 유통업체가 인터넷 및 EDI(Electronic Data Interchange)를 통해 실시간으로 활용할 수 있도록 지원하는 상품정보 전자카탈로그
- (산업통상자원부) 유통산업 융합 얼라이언스 통합 포럼을 개최하고, 유통산업의 혁신을 통한 글로벌 경쟁력 강화 지원을 위해 2018년부터 5년간 약 170억 원(2018년 34억 원)의 연구개발(R&D) 예산을 신규 투자하기로 발표. 이번 예산은 상품·구매 정보에 대한 빅데이터 구축, 인공지능 기반 개인 맞춤형 상품 추천, 가상·증강 현실(VR/AR) 쇼핑 등 미래 유통산업의 핵심 경쟁력 확보를 위한 유망 과제에 지원될 예정
- (한국과학기술원) Auto-ID Labs에서는 GS1 국제표준기구의 EPCGlobal 관련 WG에서 표준화 활동을 진행 중이며, 표준 규격에 대한 레퍼런스 오픈소스 프로젝트를 각 스택별로 OLIOT(Open Language for Internet of Things)이라는 이름으로 공개하였으며, 완주로컬푸드, 물류 프로젝트 등에서 활용 중

1.1.1.1.35. (스마트 소비 서비스 표준) 농축수산식품을 포함하는 상품, 장소, 자산, 사람 등의 식별자을 키로 하여, 원산지, 리콜, 구매, 성분정보 등 다양한 서비스를 연결시키는 표준과 기술이 개발 중

- (유통물류진흥원) 생산자, 물류 및 유통기업, 소비자등의 산업 구성원들의 식별 및 물리적 위치 식별을 위한 국내 지자체, 부서 등의 글로벌 식별자를 부여하는 프로젝트를 진행 중
- (한국과학기술원) 생산자, 물류 및 유통기업, 소비자등의 산업 구성원들의 식별 및 물리적 위치 식별을 위해 활용될 수 있는, 고도화된 주소(도로명주소, 지점번호, 사물주소)를 GS1 GLN(Global Location Number)로 매핑하며, 위치기반 서비스 및 스마트시티에 적용 중에 있으며, 상품, 장소, 자산, 사람 등의 글로벌한 식별자를 키로 하여, 다양한 서비스를 등록, 검색, 접근할 수 있는 GS1 Digital Link, ONS(Object Name Service) 기술을 개발하고 실제 산업에 적용 중

표 4 - 국내 주요 사업자 서비스 동향

사업자	주요 현황
1.1.1.1.35.1. ET	- 2020년, 디지털트윈과 AI기술을 활용하여 영상, 음성의 분석을 통한 사양관리 기술 개발 중

RI	
KT	- 2017년 2월, 'GS1 농식품 표준 정보' 플랫폼 기반으로 만든 데이터를 가공해 '농산물 이력 정보' 솔루션 제공(스페인 MWC 2017에서 소개)
긴트	- 2020년 8월, 클라우드기반 농업기계 인텔리전스 시스템 상용화
대동공업	- 2020년 2월, 6조 직진자율주행 이양기 출시 - 2019년 5월, RTK GPS 기술 활용한 직진자율주행 이양기 상용화
동양물산	- 2019년 5월, 8조 자율직진주행 이양기 상용화
스마트팜 융합연구단	- 2018년, 저가형 통합제어기와 생육 센서 기술 개발
스마트팜 융합연구단	- 2018년, 비닐하우스 내부의 환경관리를 위한 저가형 통합제어기와 생육 센서 기술 개발
1.1.1.1.35.2. 유라이코리아	- 2018년, IoT기반 바이오캡슐 개발을 통해 가축 질병 모니터링 서비스 제공 - 2019년, 말 컨디션 관리에 특화된 '패치타입' 헬스케어 디바이스
이지팜	- 2018년 4월, 완주로컬푸드에서 농식품 생산, 유통, 물류, 소비 플랫폼에 GS1국제표준 기반의 식별체계와 유통데이터, 이력추적 시스템을 구현
(주) 나래트랜드	- 2020년, IoT와 AI를 활용한 클라우드 기반 팜노트 서비스 상용화
(주) 유비엔	- 2018년, 400MHz대역 RF를 사용한 스마트팜 ICT 기자재 개발 - 2018년, SaaS기반 스마트팜 서비스를 제공하기 위한 팜링크 상용화
한국축산데이터	- 2019년 5월, 2019년 데이터 분석에 기반한 가축 건강 관리 프로그램인 '팜스플랜'을 출시. 이는 가축의 나이와 체중 등의 기본 정보에 혈액과 분변을 분석해 질병 유무와 면역력 상태를 파악 가능

#### ○ 국외 기술개발 현황 및 전망

(시설원에 환경 관리 표준) 기존 하드웨어 기반의 환경관리 제어 시스템 기술과 더불어 소프트웨어 기반의 작물별 제어시스템 기술 개발이 발전하고 있으며, IoT 기술과 클라우드 기술과 접목하여, 기존 생산/소비예측, 물류/유통구조 개선 목적뿐만 아니라 경작도구와의 데이터를 접목시키는 방향으로 발전되는 추세

- 1.1.1.1.35.3. - (독일 Bosch) 자동차 부품회사로 유명한 독일의 Bosch사도 그동안 축적된 IoT 기술을 이용하여 스마트팜 분야에 뛰어들었으며, IoT 플랫폼을 이용한 딸기 재배 솔루션 제공
- 1.1.1.1.35.4. - (이스라엘 CropX) 토양의 상태를 측정하고 그 데이터를 클라우드로 저장함으로써, 유무선 단말이 언제 어디서나 저장된 데이터에 접속할 수 있는 하드웨어와 소프트웨어 솔루션 제공
- 1.1.1.1.35.5. - (미국 FarmMobile) 농기계에 IoT 장치를 설치하고, 그 데이터를 클라우드로 저장함으로써 실시간 및 과거 경작 이력을 분석할 수 있도록 하며, 하나의 모바일 스크린을 통해 다수 농기계간의 상호 동작을 표시
- 1.1.1.1.35.6. - (네덜란드 HortiMax) 다양한 센서와 기상 정보를 이용하여 시설의 환경을 예측하고 시설 내의 온도 편차를 최적화하는 솔루션을 제공하고 있음. 또한 인공과일 센서, 적외선 온도센서, 이산화탄소 센서 등을 이용하여 보다 정확한 작물주변 환경 정보를 수집할 수 있는 기술과 사용자 설정이 가능한 소프트웨어 기반의 제어시스템을 통해 다양한 제어 옵션을 지정할 수 있는 기술을 개발
- 1.1.1.1.35.7. - (네덜란드 PRIVA) 온실 환경 제어 기술을 활용하여 각종 센서와 모니터링 장치가 유기적으로 작동하는 시스템을 개발하고 이를 활용하여 원예시설 내부의 다수의 블록을 동일 조건으로 제어 가능한 기술을 확보
- 1.1.1.1.35.8. - (일본 Fujitsu) ICT에서 농업 경영을 비약적으로 효율화시켜 언제라도 안전한 식사를 할 수 있는 환경을 실현하기 위해 농부 개개인의 경험 대신 관련 데이터의 축적과 분석, 데이터의 공유와 활용을 통한 과학적 농산물 관리 시스템인 "Akisai" 솔루션을 개발

- (일본 미도리) 입문형과 프로형으로 상용 서비스 진행 중에 있으며, 입문 버전의 경우 초기 비용이 89,000엔, 월 클라우드 사용 비용은 2,260 엔수준으로 카메라/온습도, 일사량 센서 등 최소한의 센서를 지원, 프로 버전은 노지 재배와 대규모 생산 시설에서 사용이 가능한 형태로 초기 비용이 128,000엔, 월 클라우드 사용 비용은 2,260엔 수준이며, 최대 6종류 16개까지-온습도 센서, 일사량, CO2, 토양복합 센서(수분/지온/EC) 등의 지원이 가능한 형태이나 센서를 활용한 생산량, 병해 예측 등 핵심 알고리즘의 지원은 미비
- (일본 OPTiM) AI·IoT·빅 데이터를 활용하여 “재미있고 멋있고, 스타 농업” 지원을 목표로 진행. 작업 부담의 경감 및 생산성 향상을 위해 OPTiM 정보 관리, Cloud IoT OS에 의한 시각화, 수확시기 및 수확량 예측, 농사기록지원, 원격 지원 등 노동력 최소화를 위한 서비스 제공이 주목적, 블록 체인 시스템을 통한 고품질 안심 등을 위한 기술도 개발 중
- (일본 Agri Field Manager) 드론이나 스마트 폰으로 촬영 한 농장이나 농작물의 영상 및 이미지, 인공지능을 이용하여 분석하고 효과적으로 작물의 생육 관리하는 플랫폼으로 다양한 분야의 센서 정보를 융합하여 고급 생육 분석을 제공하는 서비스를 제공하고 있으며, 세계 최초로 무인 항공기 이미지에 의한 딥 러닝 기술을 이용한 병해충 탐지기술을 개발. Agri Assistant는 자동 음성 입력에 의해 쉽게 농사 기록을 학습하여 GAP 취득에 필요한 농사 정보의 공유가 가능
- (포르투갈 FarmCloud) 축산용 클라우드 플랫폼으로 실시간 관리 솔루션 기반으로 다른 제조업체의 Climate 및 Feeding Controller와의 플러그 앤 플레이 통합 가능한 솔루션을 제공하는 것이 특징. 실시간으로 팜에 액세스하고 FarmCloud에 연결된 모든 타사 룸 컨트롤러의 모든 정보를 동시에 관리가 가능하며 별적으로 또는 그룹으로 제어 가능. 또한 FarmCloud Reports Wizard를 사용하여 대부분의 기록 데이터의 비교 및 분석이 가능하며 별도의 API 제공을 통해 다양한 소프트웨어 의 추가 개발 및 스마트 단말용 앱을 통해 서비스를 제공

(스마트팜 실외(과수, 발작물 등) 로봇 운용 표준) 자율주행, 무인 트랙터를 개발 완료하였으며 텔레매틱스를 이용한 스마트 시스템과 결합하여 영농관리 시스템과 경영 개선 서비스를 개발 중

- 1.1.1.1.35.9. - (미국, 존디어) 작물 열 감지 자율주행, 자동선회, 작업기 지능제어 등 모듈기술 상용화 하였으며 AMS(Agricultural Management Solutions) 스마트 시스템과 결합된 영농관리 시스템을 개발하여 정밀파종, 경운, 방제 등의 변량작업 기술 보급 중
- 1.1.1.1.35.10. - (영국, CNH) Case IH Concept은 완전 무인 자율주행 트랙터로써 운전석이 없음. Lidar, Radar 센서, 전후방 카메라가 장착된 컨셉 단계이며 시험 중
- 1.1.1.1.35.11. - (일본, Kubota) 소형 농기계 산업을 주도하는 쿠보다(Kubota)는 GPS 위성의 위치정보를 농기계에 보급 가능한 수준으로 제작비를 절감하는 통합형 센싱 시스템을 개발하고 있으며, 이앙기와 같은 제품에 우선적으로 장착 보급 중. KSAS(Kubota Smart Agri System)과 연동
- 1.1.1.1.35.12. - (독일, Deepfield Robotics) 이동 플랫폼에 다양한 작업 모듈을 부착하여 농작업을 수행하는 BoniRob을 개발 중
- (프랑스, Macon) 자동 가지치기 로봇, 토양데이터, 정확한 과실 데이터 저장 및 관리
- (프랑스, Nao Technologies) RTK GPS를 통해 2cm 정밀 범위에서 동작하며 비전 기술을 통해 고 정밀 제조 로봇을 개발. 와인전용 제조로봇, 채소작물 제조로봇 등 작물에 따른 전용 로봇을 개발
- 1.1.1.1.36. (스마트축사 환경 관리 표준) 축사 내 설치된 다양한 센서들로부터 정보를 수집 및 축적함으로써 축사 내 최적 환경 제공과 질병관리, 에너지 효율화 등 축사의 경영을 예측하는 기술 개발 방향으로 발전하는 추세
  - 1.1.1.1.36.1. - (네덜란드, Sparked) 센서를 소의 귀 안에 삽입, 실시간으로 상태를 측정하고 정보를 수집할 수 있는 기술을 개발하여 가축의 건강 상태를 확인하고 고기나 우유에서 비롯되는 다양한 질병을 조기에 예방 가능
  - 1.1.1.1.36.2. - (이탈리아, Zoetis) 양돈개체관리 프로그램 ‘피그와이즈(PigWise)’는 RFID 인식기와 카메라를 이용하여 돼지개체별 성장과 복지, 개체 모니터링 등을 수행 중. 프로그램의 체계는 양돈축사 내 동물행동 탐지, 스마트 사료섭취행위 모니터링, 농가 조기 알람시스템으로 구성
  - 1.1.1.1.36.3. - (덴마크, SEGES) 양돈 농가의 중앙 서버와 돈사에서 일하는 직원들의 휴대용 단말기를

연결하여 돈사에서 수집되는 다양한 데이터들을 효과적으로 저장하고 그 데이터들을 활용, 향후 경영에 대한 예측을 제시. 돼지의 짹기에서부터 임신, 분만, 이유식, 질병 치료, 비육돈 사육, 사료, 출하 등까지 양돈 경영 전반의 문제의 처리가 가능하며, 자돈, 모돈, 사료 최적화, 모돈 분석, 생산 계획, 비육돈, 자돈, 모돈 핵심 관리, 집중포인트 분석, 모돈육성 및 번식 등의 기능을 포함

- 1.1.1.1.36.4. - (오스트리아, smaXtec) 자사의 축우 모니터링 시스템은 센서가 내장된 소형기기를 젓소의 체내에 삽입하여 질병과 건강상태 등을 개체별로 모니터링하면서 데이터를 축적 및 분석하는 방식을 활용하고 있는데, 이러한 체내 측정은 통하여 젓소의 사료섭취, 움직임, 체온, pH 등 양질의 데이터 획득 가능
- 1.1.1.1.36.5. - (네덜란드, Fancom) 돈사에 소리 센서를 설치하고 이를 통해 돼지의 기침소리를 감지 및 분석하여 관리자에게 알려주는 Pig Cough Monitor 장치를 개발. 특정 돈사 및 돈방의 이상 기침 정보에 관리자의 관찰이 더해져 돼지 호흡기 질병을 초기에 대응할 수 있으며 아울러 치료효과를 높이고 항생제 사용을 줄일 수 있음
- (미국, Cargill) 아일랜드 영상인식 소프트웨어 회사 케인서스와 손잡고 암소 표정을 분석하는 프로그램을 개발. 이 프로그램은 소가 여물을 먹는 자리에 카메라를 설치하고 움직임이나 먹는 모습을 확인해 농부에게 소의 건강 상태를 실시간 제공
- (중국, 알리바바) RFID를 대신해 AI와 머신비전이라는 영상인식, 음성인식 기술을 활용해 돼지 육질이 가장 좋은 사육 환경과 도살 시점을 분석. 핵심인 머신비전은 자동화 공장이나 농장에서 사람을 대신해 카메라와 영상처리 소프트웨어로 관리가 제대로 이뤄지는지 알아내는 원리로서, 축사 천장에 달린 카메라로 돼지 몸에 새긴 번호 문신을 인식해 추적하는 방식. 이 기술은 돼지 개체 수와 새끼 돼지를 구별하는 수준이며, 향후 AI를 활용해 이보다 훨씬 정확한 분석을 제공할 계획

**(축사 사양 관리 표준)** 데이터를 중심으로 하는 축산 사양관리는 과거 경험 중심 및 비공유 중심에서 빅데이터, 인공지능 등의 기술을 활용한 서비스 지능화 방향으로 개발되고 있으며, 기술의 고도화에 따른 표준의 개발 및 확산 예상. 단순히 ICT 기술을 활용하여 원격제어를 통한 노동 편의성 제공 수준을 넘어 최종적으로는 사람의 인지 능력을 초월하고 예측되지 않는 다양한 기술의 적용을 통해 생산성의 극대화과 부작용을 최소화하는 기술개발 수준로 진화

- (이탈리아, PigWise) RFID 및 CCTV를 통해 돼지 축사 내 동물 행동탐지, 사료섭취감시, 알람 시스템을 통해 돼지의 건강, 성장 등과 관련한 문제 발생을 예측하는 긴급 알람 기능을 제공
- (벨기에, Soundtalk 시스템) 돼지 및 육계의 소리 분석을 통해 건강상태의 실시간 감시, 호흡기 질병의 예방, 환기 상태의 분석 및 제어를 통해 폐사율 감소 및 생산성 향상 서비스 제공
- (네덜란드, eYeSCAN) 8대의 카메라를 통해 돼지 활동성 및 출하 체중을 분석, 사료 급이기를 통한 사료 효율의 분석 등 노동력 절감과 생산성 향상을 위한 서비스 제공
- (일본, 후지쯔) 개발 보급하고 있는 우보시스템은 1천여농가에 보급되고 있으며, 후지쯔 클라우드서비스를 이용하여 소 발목에 만보계 장비를 채워 소의 발걸음수를 측정하고 이를 분석하여 농장주에게 이상탐지 결과를 알려주고 있음
- (스웨덴, DeLaval) 낙농목장에서 사용하는 착유용 장갑 및 앞치마부터 급이시스템, 냉각장치, 유질개선제품, 동물복지설비, 로봇착유시스템에 이르기까지, 목장의 생산효율과 수익을 증가시키면서 환경적인 영향은 감소시키고 동물 및 목장주의 더 나은 삶을 추구
- (오스트리아, smaXtec) 4년 이상 사용이 가능한 삽입 센서를 통해 젓소의 사료섭취, 움직임, 체온, pH 등 데이터 수집을 통해 빅데이터 기반으로 지역별, 규모별, 축종과 품종별로 최적의 사육관리 시스템을 제공하여 건강상 특이진단 및 치료방법에 대한 서비스 제공
- (영국, 스코틀랜드, 글래스고) 소에게 ‘디지털 목줄’을 걸어 움직임을 추적해 출산율을 높이기 위한 장치를 개발, 번식 시기가 되면 평소보다 많이 움직이는 점을 포착해 특정 소가 짹기 할 준비가 됐다고 판단되면 목장주의 휴대전화로 메시지를 보내주는 서비스 제공
- (이스라엘, 아피밀크) 낙농기술회사로써, 이 기술을 인수한 뒤 소의 질병을 감지하는 장치로 발전. 소가 음식을 먹고 되새김질하는 평균 시간을 체크해 이 시간이 줄어들면 알림
- (스코틀랜드, SRUC) 젓소가 세균 감염으로 유방염을 앓으면 우유 생산량이 줄고 죽음으로까지 이어질 수 있어 큰 비용을 초래함. 이를 막기 위해 스코틀랜드 농가들은 외양간에 열화상 카메라를 설치하고 열이 나는 소를 찾아내 치료, 소를 3D 카메라로 찍은 뒤 기존데이터를 활용해 무게와 육질 등급을 추정

1.1.1.1.37. **(동물복지 인증 관리 표준)** 동물복지 인증과 관련하여 각 국에서는 정책적으로 동물복지 인증 기준을 마련하였으나 동물복지 인증 시스템과 관련된 표준은 진행되고 있지 않으나, 동물의 5대 자유, SFS협정 등 관련하여 관련 표준의 개발 및 확산 예상

- (영국 환경식품농무부(DEFRA)) FAWC의 자문을 받아 농장동물복지 정책을 실시하기 위한 권고 규약을 제정하고, 동물의 5대 자유를 중심으로 동물복지 기준을 마련하여 “RSPCA Assurd”를 실시하여 동물복지 인증을 실시
- (독일 연방식품농업부(BMEL)) 독일의 동물복지 축산물 인증제의 경우, 축산농가, 육가공업체, 소매업체를 포함한 민간기업 및 단체 등이 공동으로 선제적인 동물복지 기준을 마련하여 주도적으로 시행
- (네덜란드 동물보호협회(Dierenbescherming)) 농장 동물 복지관련 인증제에는 동물복지 환경(방사형 사육, 유기농환경)을 구분하여 네덜란드 동물보호협회가 시행하는 친동물 육류마크 Beter Leven 인증제가 있으며, 별의 개수가 증가할수록 동물복지 환경의 수준이 높음
- (덴마크 환경식품부(MFVM)) Better Animal Welfare 마크는 3개의 하트로 동물복지 농장의 수준을 나타내며, 돼지 농장에만 운영 중이나 향후 다른 종에도 적용할 예정
- (미국 식품안전 및 검사 서비스(FIFs)) 식품안전 및 검사 서비스(FIFs, Food Safety and Inspection Service)의 승인을 받아야 유기, 자유방목, 케이지프리, 또는 무항생제 사육등과 같은 축산물 관련 인증제를 시행

1.1.1.1.38. **(수산양식 환경 관리 표준)** 2011년 기준 세계 수산물 생산량의 약 40%가 양식 수산물이며, 2030년에는 전체 수산물 생산량의 50%를 양식수산물이 차지할 것으로 예상(FAO, FISH TO 2030(Prospects for fisheries and aquaculture)). 양식업은 성장 잠재력이 가장 큰 분야중 하나로 평가되고 있으며, 세계 각국은 미래 식량 산업으로 양식업에 주목하고 투자를 강화

1.1.1.1.38.1. - (일본, NEC) 양식업 현장에 IoT기술을 활용하여 어류의 무게를 신속히 측정가능한 시스템을 개발, 2019년 이후 서비스 개시 목표

양식어장에 특수카메라를 넣어 동영상 촬영, NEC가 자체 개발한 화상인식 기술과 딥 러닝(Deep Learning) 기술을 통해 측정에 적합한 생선과 그 특징들을 자동적으로 추출하여 몸길기와 무게를 정확히 측정 가능

\* 기존 양식장에서 70마리의 몸 길이, 무게를 측정하는데 1시간이 걸렸으나, 이 기술을 활용해 555마리를 단 10분 만에 측정하는데 성공

양식장의 전체 현황을 정확히 파악함으로써 양식업자는 출하시기를 정확히 판단할 수 있으며 먹이를 주는 양도 최적화시킬 수 있어 바다 오염 등 환경 부하 감소가능

1.1.1.1.38.2. - (노르웨이, AKVA) 양식장 조명과 수온, 산소, 염도 등을 측정하는 센서시스템, 자외선 살균 시스템, 레이저 유해충 제거 시스템 등이 포함된 육상양식장 개발하고, 각종 환경 센서 및 화상 카메라, 생산제어 소프트웨어를 통해 사료 급이 자동화를 구현 중. AKVA사의 CSS 사료공급 시스템은 중앙공급 시스템의 개념으로 각종 환경센서 및 고급 카메라와 생산제어 소프트웨어가 통합되어 시간과 사료량, 사료와 증강제 비율 등을 자동으로 제어

1.1.1.1.38.3. - (아이슬란드, VAKI) 적외선이 흐르는 통로형 구조물을 수조에 넣어 구조물로 통과하는 연어의 이동속도를 고려하여 크기를 측정하는 장치 개발

1.1.1.1.38.4. - (덴마크, Oxyguard) Microsoft 사의 cloud 서비스를 활용하여, 전 세계에 약 20년간 공급하여, 설치를 해 놓은 양식장 관리 시스템 상에 축적된 데이터를 cloud로 통합하여, 데이터분석 서비스를 시행할 계획. 양식기술의 최적화뿐만 아니라, 양식장을 기준으로 전방산업(가공/유통/소비)과 후방산업(사료회사, 약품회사 등)을 통합적으로 데이터를 공유할 수 있는 시스템을 구현 추진 중

**(이력 데이터, 추적, 위조방지(블록체인) 표준)** IBM, SAP 등을 중심으로 유통의 이력데이터에 대한 기술적인 솔루션이 블록체인과 함께 개발되고 있는 추세

1.1.1.1.38.5. - (미국, IBM Food Trust) 2018년 10월 블록체인 기반 식품 추적 플랫폼을 출시. 이 플랫폼은 전세계 식품 도·소매업자, 공급자들을 대상으로 클라우드 서비스를 제공하며, 기업 블록체인 네트워크가 확장 가능한 규모로 완전히 배치된 첫 사례임

- 1.1.1.1.38.6. - (미국/유럽, IBM-머스크 TradeLens) IBM과 세계 최대 해운업체 머스크가 공동 개발한 블록체인 기반 해상 화물 운송 플랫폼인 트레이드 렌즈(TradeLens)는 실시간으로 투명하게 상품 이동 정보를 제공하고, 스마트 계약을 활용하여 페이퍼워크(Paperwork) 중심의 업무 프로세스를 대체
- 1.1.1.1.38.7. - (독일, SAP) SAP은 블록체인 기반 식품이력추적 플랫폼을 제공하며, 미국 참치 캔 제조회사인 Bumble Bee Seafoods와 함께 Ocean to Table 식품 이력추적 시스템을 구축하며, 이력 추적 생태계를 넓혀가고 있음
- 1.1.1.1.38.8. - (미국, IFT/WWF) 2017년부터 IFT(Institute of Food Technologists)와 WWF(World Wildlife Fund)는 세계 경제포럼의 권고를 받아들여, 안전한 수산물 확보와 해양생물보존을 목표를 글로벌 수산 및 유통기업과 함께 GDST(Global Dialogue on Seafood Traceability) 국제수산물 이력추적 구축 프로젝트를 수행 중

(유통물류 데이터 표준) 유통 이벤트의 분산 데이터 레파지토리 표준인 GS1 EPCIS 표준 중심으로 유통 데이터 플랫폼 구축이 진행 중이며, Oracle, SAP, KAIST 등 다양한 솔루션들이 확산 중

- 1.1.1.1.38.9. - (미국, Oracle) Oracle Pedigree and Serialization Manager(OPSM) 솔루션을 제공. 회사가 시리얼 단위의 제품을 관리하고 공급망 전체에서 시리얼 단위의 제품 데이터를 공유할 수 있으며, OPSM을 구성하는 핵심 DB기술로 GS1의 EPCIS 레파지토리를 사용 중
- 1.1.1.1.38.10. - (독일, SAP) Auto-ID Infrastructure 솔루션을 제공하며, RFID를 사용하여 공급망을 관리하고 비즈니스 프로세스를 자동화할 때 시리얼 단위의 제품에 대한 정확한 실시간 정보를 회사에 공유. Auto-ID Infrastructure에서 이벤트 관리를 위하여 GS1 EPCIS 를 사용 중
- 1.1.1.1.38.11. - (미국, FoodLogiQ) FoodLogiQ Connect 솔루션을 제공하며, FoodLogiQ Connect API를 사용하여 기존 시스템 및 데이터와 통합하고 상품 이력추적, 식품안전 준수 및 공급망 투명성 소프트웨어 서비스를 실현. 또한, 시리얼 단위의 상품의 생산, 유통 이벤트 정보를 관리하기 위하여, GS1의 EPCIS를 사용 중이며, 이력추적 기술 개발을 위한 표준으로 GS1의 Traceability를 사용 중

1.1.1.1.39. (스마트 소비 서비스 표준) 상품에 부착하는 QR 표준화와 서비스 연결 기술을 기반으로 네슬레, 메트로 등 식품 생산기업과 유통기업들은 소비자를 위한 서비스 개선을 추진 중

- (스위스 네슬레/독일 메트로) 스위스 네슬레는 자사 식품의 다양한 소비자 서비스 제공을, 그리고 독일 메트로는 매장 상품의 향상된 소비자 서비스 개선을 위해서 GS1 Digital Link 표준을 적용하기 시작
- (스위스, EVERYTHNG) GS1 Digital Link 국제표준 기반의 사물과 서비스를 연결시키는 서비스 플랫폼을 개발하여 글로벌 유통시장에 진입

표 5 - 국외 주요 사업자 서비스 동향

사업자	주요 현황
KUBOTA	- 2017년, 일본 스마트 농업 시스템인 KSAS과 연계가 되는 자율 트랙터(60마력 클래스)의 시범 판매
Aerobotics	- 2019년, 무인 항공기 및 위성 이미지를 통해 인공지능으로 초기 해충 및 질병 탐지가 가능한 엔드투엔드 솔루션(남아프리카공화국)
Agrobot	- 2019년, 24개의 로봇 팔과 단거리 통합 컬러 인식 센서 및 적외선 센서 등을 이용하여 동작하는 딸기 수확 로봇으로 개발 및 판매 중(스페인)
Bosch	- 2018년, IoT 플랫폼을 이용한 딸기 재배 솔루션 제공
1.1.1.1.39.1. CropX	- 2019년, 토양의 상태를 클라우드에 저장하여, 유무선 단말로 접속할 수 있는 하드웨어와 소프트웨어 솔루션 제공
1.1.1.1.39.2. FarmM	- 2018년, IoT 장치를 설치한 농기계로부터 데이터를 클라우드에 저장하여, 실시간 및 과거 경작 이력 분석 및 모바일 스크린을 통한 다수 농기계간의 상호 동작 정보 제공

obile	
Four Growers	- 2019년, 컴퓨터 비전과 기계 학습을 이용하는 토마토 수확 로봇(미국)
Fujitsu	- 2012년 10월, ICT에서 농업 경영을 비약적으로 효율화시켜 언제라도 안전한 식사를 할 수 있는 환경을 실현하기 위해 농부 개개인의 경험 대신 관련 데이터의 축적과 분석, 데이터의 공유와 활용을 통한 과학적 농산물 관리 시스템인 "Akisai" 솔루션을 개발
IBM, FoodLogiQ, ripe.io, SAP	- 2020년 6월, GS1 US와 함께 글로벌 식품 이력추적 플랫폼을 GS1 국제표준에 기반하여 IBM Food Trust, FoodLogiQ, ripe.io, SAP 4개사가 PoC 구축
NEC	- 2018년, 자체 개발한 M2M 플랫폼(Connexive)과 연계하여 농업 ICT 클라우드 서비스를 제공
Precisionhawk	- 2016년, 소형무인기를 활용한 농업용 영상 촬영 및 분석 시스템을 개발 및 판매
PRIVA	- 2017년, 온실 환경 제어 기술을 활용하여 각종 센서와 모니터링 장치가 유기적으로 작동하는 시스템을 개발 - 원예시설 내부의 다수의 블록을 동일 조건으로 제어 가능한 기술을 확보
root AI	- 2019년, 실시간 감지, 젠틀 터치, 지능형 모션 등 인공지능을 이용한 토마토 수확로봇(미국)
SoundTalks (벨기에)	- 2019년 4월, 조기에 돼지 기침 분석을 하여 적절한 조치를 위하여 양돈 농장 곳곳에 마이크를 설치하고 소형 컴퓨터로 농장에서 발생하는 모든 소리를 수집하여 분석하여 호흡기 질병의 발병 가능성을 예측 서비스 제공
알리바바 (중국)	- 2019년 2월, 안면인식 기술을 활용, 돼지를 한 마리씩 구분하여 생애주기 관리 서비스 제공 - 2018년 6월, 돼지 사육장의 이미지를 실시간 분석해 압사 위험성을 자동 감지하여 관리자에게 통보 서비스 제공

○ 표준화 현황 및 전망

표 6 - 주요 표준화 기구의 스마트팜 표준화 현황 및 전망

구분	표준화 기구	표준화 현황
국제 (공식)	ISO	TC234 (Fisheries and aquaculture) 수산양식의 용어, 장비 및 운영에 대한 기술 사양, 양식장 특성화 및 적절한 물리적, 화학적 및 생물학적 조건의 유지, 환경 모니터링, 데이터보고, 추적성 표준을 추진할 예정이며, 수산양식 표준 개발을 ISO TC34 SC17과 Joint WG을 결성(Aquaculture)
		TC23 (SC2-Common Test) 농업용 트랙터의 PTO, 3점링크, 배기가스 측정, 무게중심, 차축 동력, 견인력 테스트, 유압 동력, 조향 능력, 저온 시동 의 테스트 표준화 진행 중 (SC3-Safety and comfort) 농업 및 임업에 사용되는 트랙터, 기계, 시스템, 도구 및 장비의 안전성 표준화 진행 중 (SC6-Equipment for crop protection) 농작물 보호장비에 대한 표준을 담당하여 농업용 드론, 헬기 등을 이용한 항공방제사용에 관한 환경요구조건 및 방제시험방법에 대한 표준화 진행 중 (SC18-Irrigation and drainage equipment and systems) 원격 모니터링 및 제어 기술, 스프링클러의 분사 및 테스트 에 관한 표준화 진행 중 (SC19-Agricultural electronics) 원예, 조경, 관개 관련 장비의 전기전자 분야 및 동물 전자식별 장비 분야의 표준화 진행 중
		TC34 1.1.1.1.39.3. (WG16-Animal welfare) 가축 및 채소의 생산에서 소비까지의 먹이 사슬에 대한 용어, 샘플링, 시험 방법에 관한 표준화 진행 중
	JTC1	SC31 (SC31-Automatic Identification and data capture techniques) 사물의 식별 및 분류체계, RFID, 이력정보 공유를 위한 프레임워크 표준화 완료(GS1 국제표준 반영)
		SC41 (SC41-Internet of things and related technologies) IoT기술의 주요 적용 분야로 스마트팜을 고려하고, 2018년부터 Agricultural IoT라는 이름의 기술보고서 작성 중
	ITU-T	SG13 (Q1-Innovative services scenarios, deployment models and migration issues based on Future Networks) 농업 정보 서비스 모델(Y.saic)에 대한 표준 개발 진행 중
		SG20 (Q2-Requirements, capabilities, and use cases across verticals) 2018년도 IoT 기반의 스마트축산 프레임워크(Y.SLF) 표준화 추진 중
국제 (사실)	GS1	(EPCIS and CBV 2.0 WG) 글로벌 유통 이벤트 데이터 수집 및 공유를 위한 EPCIS 의 표준데이터 모델 인터페이스 개발 진행 중 (Blockchain IG) 유통 데이터에 대한 블록체인 데이터 생태계 구축 진행 중이며, 산업표준으로 식품, 의약품, 가축, 수산물 등의 이력추적 가이드라인 표준화 진행 중 (ONS and Digital Link) 상품, 자산, 장소, 사람 등의 식별자를 키로 하여 서비스를 등록, 검색, 접근할 수 있는 서비스 공유 프레임워크 표준화 중
		GDST (GDST 1.0) 글로벌 수산 및 양식물 이력추적을 위한 식별체계, 프로세스 모델, 이벤트 및 주요 데이터 엘리먼트를 표준화 함

	oneM2M	(WG1-RDM(Requirement and Data Model) WG) 가전 모델 정의를 위한 HAIM(Home Appliance Information Model) 표준 개발을 통하여, 정보 모델의 정의를 전체 산업분야로 확장하여 개발 중 (WG2-SDS(System Design and Security) WG) 3GPP 네트워크와 oneM2M 플랫폼간 연동 표준 개발 진행 중 (WG3-TDE(Testing & Developers Ecosystem) WG) 3GPP 코어 네트워크와 oneM2M 플랫폼간 연동 검증 표준 개발 중 * 아직 스마트팜과 직접적으로 관련된 표준 개발은 진행되고 있지 않음	
	OCF	(Bridging TG) BLE, ZigBee, Z-wave 등에 대한 PoC 구현 및 테스트를 진행 중이고, 브릿지에 대한 공동 테스트케이스 개발과 ZigBee에 대한 검증 완료 * OCF에서는 다양한 사물인터넷 디바이스 간 연결 및 상호 연동하기 위한 플랫폼 제공을 목표로 하고 있으나, 아직 스마트팜과 직접적으로 관련된 표준개발은 진행되고 있지 않음	
	W3C	(Agriculture Community Group) 웹 API 및 서비스를 사용하는 기존 사용자들의 시나리오를 수집하고 분류하여, 웹 개발자와 농업 이해 관계자에게 지능화된 장치를 제공하기 위한 목적으로 2014년에 신설된 그룹이나, 아직까지 활발한 활동을 보이지 않음	
국내	TTA	PG426	(스마트농업) 스마트농업 서비스프레임워크, ICT 융합기술 등 스마트농업의 필수 기술 표준화를 진행하고 있는 그룹으로, 스마트온실 및 축사 관련 상호호환성 표준, 유통 관련 표준 제정 및 지속적인 표준화 추진
	농식품ICT 융합표준포럼		2014년 6월 신설되어, 스마트농업 전반의 유즈케이스, 시설원에 센서 인터페이스, 메타데이터 및 플랫폼, 생산·유통 관련 표준개발을 위한 산·학·연 관련 전문가로 구성되어 협의체 운영
	스마트팜ICT 융합표준화포럼		2017년 9월, 스마트팜 관련 산·학·연 관련 전문가로 구성되어 신설하여, 스마트원예, 스마트축산, 스마트농기계, 국제표준 분과로 나누어 운영 중. 스마트팜 관련 센서/구동기 국가 표준 제정 완료하고, 스마트축사 사양관리기 표준 개발 중

○ 국내 표준화 현황 및 전망

(시설원에 환경 관리 표준) 스마트팜 환경 관리와 관련하여, 국내의 경우 주로 스마트팜 시설 내부의 장치들간 호환성 및 연동성을 제공하기 위한 인터페이스에 대한 표준화가 주로 진행하였으며, RF기반의 스마트관제 프로토콜, 시설원예를 위한 표준 용어, 스마트팜 제어기에 사용되는 스크립트에 대한 표준화가 진행되었으며, FaaS(Farm as a Service) 개발로 비용절감과 데이터의 기밀성과 효율성 배가를 위한 표준 개발이 본격화 될 것으로 전망

- 1.1.1.1.39.4. - (국가기술표준원 스마트팜표준전문위원회) 스마트팜 센서 및 구동기 인터페이스에 관한 H/W 연결 방식과 메타데이터와 관련된 국가표준 제정(2018년)
- 1.1.1.1.39.5. - (TTA 스마트농업 PG(PG426))
- 1.1.1.1.39.6. · 시설원에 장비간 연동을 위한, 유즈케이스를 개발한 바 있으며, 스마트팜 서비스 프레임워크 및 시그널링에 관한 표준을 개발
- 1.1.1.1.39.7. · 클라우드 기반 스마트팜 서비스를 위한 표준은 2016년부터 개발되어 왔으며, 클라우드 컴퓨팅 기술을 기반으로 스마트팜을 운영하는데 필요한 용어, 구성요소 및 서비스 요구사항을 정의 완료하고, 스마트팜운영서비스(FOS based FaaS) 등의 시나리오를 제시
- 1.1.1.1.39.8. - (농식품ICT융합표준포럼) 2014년에 신규 설립 이후, 2015년부터 시설원예, 유통 및 축산 분과 운영을 통해 시설원예 인터페이스와 관련된 표준을 진행 중
- 1.1.1.1.39.9. - (스마트팜ICT융합표준화포럼) 2017년도에 농식품 ICT 관련 산업체들의 의견을 수렴하기 위해 산·학·연 관련 전문가들을 주축으로 신설·운영되고 있으며, 2018년에는 TTA 단체 표준으로 채택된 센서/구동기 인터페이스를 국가표준으로 채택 완료. 2019년에는 축산분야 국가표준을 제정 중

표 7 - 시설원예 분야 국내 표준화 현황

개발기구	표준(안)명	개발연도
국가기술 표준원	KS-X-3265, 스마트온실 구동기 인터페이스	2018
	KS-X-3266, 스마트온실 센서 인터페이스	2018

개발기구	표준(안)명	개발연도
	KS-X-3267, 스마트온실 노드 및 제어기간 RS485 모드버스 인터페이스	2018
	KS-X-3268, 스마트온실 구동기 메타데이터	2018
	KS-X-3269, 스마트온실 센서 메타데이터	2018
TTA PG426	TTAK.KO-10.1164-part1, 스마트팜 용어 정의 - 제1부: 온실	2019
	TTAK.KO-10.1165, 스마트온실 서비스 제어 프로토콜: 기본 서비스 절차	2019
	TTAK.KO-10.1166, 스마트 온실 제어 스크립트 언어 정의	2019
	TTAK.KO-10.1167, 경량형 제어프로토콜(LCP)을 위한 IP 적응 계층(IPAL) 동작 요구사항	2019
	TTAK.KO-10.1171, 스마트온실에서 온실 통합 제어기와 양액기 노드 간 RS485 기반 모드버스 인터페이스	2019
	TTAK.KO-10.1172, 모드버스/RS485 기반 스마트온실 노드/디바이스 등록 절차 및 기술 규격	2019
	TTAK.KO-10.1177, UHF대역 400MHz무선링크기반 스마트 온실 제어 프로토콜	2019
	TTAK.KO-10.1178-part1, 스마트온실 설치 지침서 - 제1부: 통신기술	2019
	TTAK.KO-10.0945/R1, [개정] 스마트온실을 위한 원격 감시용 스마트 영상장치	2019
	TTAK.KO-10.1086, 스마트온실용 센서/구동기 I/O 인터페이스 추상화 모듈	2018
	TTAK.KO-10.1087, 스마트온실 관제를 위한 경량형 제어 프로토콜	2018
	TTAK.KO-10.1088, 스마트팜 센서노드와 게이트웨이간 비연결형 통신프로토콜	2018
	TTAK.KO-10.1089, 클라우드기반 스마트팜 영농작업관리 서비스 인터페이스	2018
	TTAK.KO-10.1089, 클라우드기반 스마트팜 영농작업관리 서비스 인터페이스	2018
	TTAK.KO-10.1008, 스마트온실용 온실운영시스템과 비순환식 양액시스템 간 통신 프로토콜	2017

(스마트팜 실외(과수, 밭작물 등) 로봇 운용 표준) 5G 기반으로 스마트폰, 태블릿 PC 장비를 이용한 ‘원격제어’ 기술 및 조작 없이 자율주행으로 농작업을 하는 ‘무인농작업’ 및 원격으로 농업기계의 위치를 파악하고 농기계 작업 이력데이터, 고장진단 등의 기술이 개발되고 있으며, 이와 관련된 시험평가, 데이터 표준, 개념정립 등의 표준개발이 주를 이룰 것으로 전망

1.1.1.1.39.10. - (국가기술표준원 ISO TC23 전문위원회)

- 1.1.1.1.39.11. · 농업용 무인항공방제기에 사용되는 분무탱크의 안전기준과 연결구성품(노즐, 호스, 연결부, 붐대 등)으로 인한 접촉 사고 방지 표준 제정 완료
- 1.1.1.1.39.12. · 자율주행 트랙터와 관련된 기초적인 기반이 될 국제 표준 부합화 제정 완료
- 1.1.1.1.39.13. · 농업용 무인항공방제의 성능을 확인할 시험방법을 국가표준으로 개발하여 국내에 보급되는 무인항공방제기의 품질기준을 위해 표준제정 추진 중
- 1.1.1.1.39.14. · 농업용 관수장비, 트랙터 통신 및 안전, 자율주행 농업기계의 레벨 정의에 관한 국가 표준 제정 진행 중

표 8 - 농기계/로봇 분야 국내 표준화 현황

개발기구	표준(안)명	개발연도
국가기술 표준원	KS B ****, 농림업용 기계 - 무인항공방제시스템 - 분무성능 시험방법	진행중 (2021)
	KS B ****, 자율주행 농기계 레벨정의	진행중 (2021)
	KS B ISO 8026:2009, 농업용 관수 장비 - 분무기 - 일반 요구 사항 및 테스트 방법	진행중 (2021)
	KS B ISO 9261:2004, 농업용 관수 장비 - 점적단추 및 점적파이프 - 규격과 시험방법	진행중 (2021)
	KS B ISO 11783-2:2018, 농림업용 트랙터 및 기계 - 직렬 제어 및 통신 데이터 네트 워크 - 제2부: 물리 계층	진행중 (2021)
	KS B ISO 25119:2018, 농림업용 트랙터 및 기계류 - 제어시스템 안전관련부품	진행중 (2021)
	KS B ISO 10975:2009, 농림업용 트랙터 및 기계류 - 운전자용 트랙터 및 자주식 농 기계의 자동 유도 시스템 - 안전 요구사항	2019
	KS B ISO 12188:2010, 농림업용 트랙터 및 기계류 - 농업에서 위치와 안내 시스템 시험 절차	2019
	KS B 7948, 농림업용 기계 - 무인항공방제시스템 - 일반 요구조건	2018

1.1.1.1.40. (스마트축사 환경 관리 표준) 국내의 경우, TTA, 농업기술실용화재단 등을 중심으로 2017  
년부터 축사 내·외부의 정보 수집 및 관리를 위한 표준화를 진행

- 1.1.1.1.40.1. - (TTA 스마트농업 PG(PG426)) 축사 내·외부 센서 정보 수집을 위한 인터페이스, 안전  
관련 정보 수집을 위한 인터페이스 및 센서 정보 저장과 공유를 위한 정보 서비스에 대  
한 데이터 스키마에 대한 가이드라인을 제정. 또한, 대상 가축에 대한 특성을 반영하여  
ICT융복합 장비규격 및 서비스 요구사항에 대한 표준 제정
- 1.1.1.1.40.2. - (농업기술실용화재단 FACT 단체표준) 축산시설 자동화 시스템 구축을 위한 기술 표준  
으로 사양관리 장치에 대한 세부 규격을 정하고 있으며, 이 표준에서는 축종별로 사용  
되는 ICT기반 사양관리 장치 19종에 대한 기계적, 전기적 인터페이스 규격과 작동 방  
식에 대한 세부 사항이 포함

표 9 - 축사 환경관리 분야 국내 표준화 현황

개발기구	표준(안)명	개발연도
TTA PG426	TTAK.KO-10.1168, 스마트축사 내기 센서 메타데이터 구조	2019
	TTAK.KO-10.1169, 스마트축사 외기 센서 메타데이터 구조	2019
	TTAK.KO-10.1175-part1, 축산 분야 ICT융복합 장비규격 및 서비스 요구사항 - 제1부: 양돈	2019
	TTAK.KO-10.1175-part1, 축산 분야 ICT융복합 장비규격 및 서비스 요구사항 - 제2부: 양계	2019
	TTAK.KO-10.1175-part1, 축산 분야 ICT융복합 장비규격 및 서비스 요구사항 - 제3부: 한우	2019
	TTAK.KO-10.1175-part1, 축산 분야 ICT융복합 장비규격 및 서비스 요구사항 - 제4부: 낙농	2019
	TTAK.KO-10.1176, 축산 빅데이터 서비스 제공자와 스마트축사 관리시스템 간의 인터 페이스	2019
	TTAK.KO-10.0979/R1 스마트축사를 위한 외기 센서 인터페이스	2019
	TTAK.KO-10.0980/R1, 스마트축사를 위한 내기 센서 인터페이스	2019
	TTAK.KO-10.0981/R1, 스마트축사를 위한 안전 센서 인터페이스	2019
농업기술 실용화재단 (FACT Standard)	SPS-FACT 0001-2018, 스마트축산 사양관리 기기 및 수집정보: I. 대가축	2018
	SPS-FACT 0002-2018, 스마트축산 사양관리 기기 및 수집정보: II. 중가축	2018
	SPS-FACT 0003-2018, 스마트축산 사양관리 기기 및 수집정보: III. 소가축	2018

(축사 사양 관리 표준) 국내의 경우, TTA, 농정원, 농업기술실용화재단, ETRI 등을 중심으로 2018년부터 축종별 ICT융복합 장비규격, 사양관리 기기 등을 중심으로 하는 표준화를 진행하고 있으나, 장비간 인터페이스 및 빅데이터, AI 등의 기술 적용을 위한 표준은 진행되고 있지 않음

- 1.1.1.1.40.3. - (TTA 스마트농업 PG(PG426)) 축종별 축사 내·외부 센서 정보 수집을 위한 하드웨어 인터페이스 데이터 스키마에 대한 기본 가이드라인과 요구사항을 제정
- 1.1.1.1.40.4. - (농업기술실용화재단 FACT 단체표준) 축산시설 자동화 시스템 구축을 위한 사양관리 장치에 대한 세부 규격을 정하고 있으며, 축종별로 사용되는 ICT기반 사양관리 장치 19종에 대한 기계적, 전기적 인터페이스 규격과 작동 방식에 대한 세부 사항을 제정

표 10 - 축사 사양관리 분야 국내 표준화 현황

개발기구	표준(안)명	개발연도
TTA PG426	TTAK.KO-10.1175-part1, 축산 분야 ICT융복합 장비규격 및 서비스 요구사항 - 제1부: 양돈	2019
	TTAK.KO-10.1175-part1, 축산 분야 ICT융복합 장비규격 및 서비스 요구사항 - 제2부: 양계	2019
	TTAK.KO-10.1175-part1, 축산 분야 ICT융복합 장비규격 및 서비스 요구사항 - 제3부: 한우	2019
	TTAK.KO-10.1175-part1, 축산 분야 ICT융복합 장비규격 및 서비스 요구사항 - 제4부: 낙농	2019
	TTAK.KO-10.1176, 축산 빅데이터 서비스 제공자와 스마트축사 관리시스템 간의 인터페이스	2019
농업기술실용화재단 (FACT Standard)	SPS-FACT 0001-2018, 스마트축산 사양관리 기기 및 수집정보: I. 대가축	2018
	SPS-FACT 0002-2018, 스마트축산 사양관리 기기 및 수집정보: II. 중가축	2018
	SPS-FACT 0003-2018, 스마트축산 사양관리 기기 및 수집정보: III. 소가축	2018

(동물복지 인증 관리 표준) 국내의 경우 TTA를 통해 2019년에 최초로 동물복지 농장을 위한 인증시스템과 관련된 서비스 표준화 연구가 시작됨. 동물복지 인증시스템의 개발 수준이 상용화수준에 도달함에 따라 인증과 관련된 시스템을 기반으로 표준 개발이 본격화 될것으로 전망

- 1.1.1.1.40.5. - (TTA 스마트농업 PG(PG426))
- 1.1.1.1.40.6. · 동물복지를 위한 사용자 관리 서비스, 운영현황서 관리 서비스, 자가진단 서비스, 농장관리 서비스, 심사관리 서비스, 인증서 교부 서비스, 가축방역관리 시스템 간 인터페이스 서비스, 상시 관리 서비스 등 농장 동물의 동물복지 인증과 관련된 요구사항을 정의
- 1.1.1.1.40.7. · 동물복지 인증시스템의 개발 수준이 상용화수준에 도달함에 따라 인증 서비스에서의 동물복지 시스템과 농장 시스템 간 인터페이스와 관련된 표준이 진행될 예정

표 11 - 동물복지 분야 국내 표준화 현황

개발기구	표준(안)명	개발연도
TTA PG426	TTAK.KO-10.1179, 동물복지 농장 심사를 위한 ICT기반 인증시스템 서비스 요구사항	2019

1.1.1.1.41. (수산양식 환경 관리 표준) 수산양식 분야의 직접적인 표준화 활동은 없는 상태이나 스마트양식의 필요성에 대한 인식이 확산되고 있어 표준 개발 추진에 많은 노력을 기울일 것으로 예상. 대부분의 양식기자재 개발이 단발성 프로젝트로 진행되어 ICT 융합부품(센서, 제어기, 통신장치 등)의 상호 호환성이 미흡한 실정으로 관련 확립 표준이 시급

(이력 데이터, 추적, 위조방지(블록체인) 표준) 농축수산물의 이력 추적 시스템 구축을 위한 표준화가 이루어지고 있으며, 유통물류 분야의 사실표준화기구인 GS1 국제 표준을 기반으로 한 국내 응용표준안들에 대해 표준화 진행 중

- 1.1.1.1.41.1. - (TTA 스마트농업 PG(PG426)) GS1 EPCIS, Pedigree, ONS 등 각 표준 컴포넌트를 기반으로 한 국내 응용 표준화가 진행 중
- 1.1.1.1.41.2. - (Auto-ID Labs, KAIST) GS1 국제표준 기반 농축산물의 이력데이터에 대한 완주로컬

푸드, 무주천마 등 국내 테스트 베드를 구축하고, 식품연구원 등과 블록체인 플랫폼 호환 기술을 개발 중이며, 국내 농축산 환경에 적용하기 위한 정책자문 활동 중

표 12 - 농축수산물 이력/추적/위조방지 분야 국내 표준화 현황

개발기구	표준(안)명	개발연도
TTA PG426	TTAK.KO-10.1003, EPCIS 기반 농축산물 이력 관리 시스템 구축 지침서	2017
	TTAK.KO-10.1002, 농축산물 GS1 표준 바코드의 레이블 출력 지침서	2017
	TTAK.KO-10.0941, 농산물 식품 생산, 유통 및 소비 정보 서비스	2016
	TTAK.KO-10.0938, 협동조합을 위한 가축 이력 사설 인증 방법	2016

(유통물류 데이터 표준) 농축산물의 유통 데이터를 관리하기 위한 데이터 모델링, 비즈니스 프로세스 정립 등의 이슈에 대해 표준화가 이루어지고 있으며, 유통 분야의 사실 표준기구인 GS1 국제 표준을 기반으로 한 국내 응용표준안들에 대해 표준화 진행 중

- 1.1.1.1.41.3. - (TTA 스마트농업 PG(PG426)) 유통 물류 데이터를 처리하기 위한 EPCIS 스키마 모델링, GS1 SOURCE를 위한 메타데이터 모델링 등 유통 물류 데이터에 대한 표준화 진행 중
- 1.1.1.1.41.4. - (Auto-ID Labs, KAIST) GS1 EPCIS, SOURCE, ONS에 대한 레퍼런스 프로젝트를 개발하여, github 커뮤니티에 공개하고 있으며, 표준화 동향에 맞춰 지속적으로 공개 중

표 13 - 농축수산물 유통물류 분야 국내 표준화 현황

개발기구	표준(안)명	개발연도
TTA PG426	TTAE.OT-10.0421, 농식품 유통 오픈 데이터를 위한 핵심 비즈니스 단계 제시 지침	2017
	TTAK.KO-10.0940, 농축산물 식품 메타데이터 모델링 가이드라인	2016
	TTAK.KO-10.1004, 스마트축사 센서데이터에 대한 EPCIS 이벤트 스키마설계 지침서	2016
	TTAK.KO-10.0939, 농산물 생산 및 유통 환경 모니터링 센서 정보 운용 방법	2016

1.1.1.1.42. (스마트 소비 서비스 표준) 농축수산식품을 포함하는 상품, 장소, 자산, 사람 등의 식별자를 키로 하여, 원산지, 리콜, 구매, 성분정보 등 다양한 서비스를 연결시키는 표준과 기술이 개발 중

- 1.1.1.1.42.1. - (TTA 스마트농업 PG(PG426)) 한국과학기술원 Auto-ID Labs에서 상품, 장소, 자산, 사람 등의 글로벌한 식별자를 키로 하여, 다양한 서비스를 등록, 검색, 접근할 수 있는 ONS(Object Name Service)기술을 개발하고 농축산물 서비스 탐색 구조 표준화 진행 중
- (유통물류진흥원) 생산자, 물류 및 유통기업, 소비자 등의 산업 구성원들의 식별 및 물리적 위치 식별을 위한 국내 지자체, 부서 등의 글로벌 식별자를 부여하는 프로젝트를 진행 중(행안부)

표 14 - 소비 서비스 분야 국내 표준화 현황

개발기구	표준(안)명	개발연도
TTA PG426	TTAK.KO-10.0942, 농축산물 서비스 탐색 구조	2016

○ 국제 표준화 현황 및 전망

(시설원에 환경 관리 표준) IoT 기반 스마트팜 환경관리 기술 관련 국제 표준화는 시작단계이나, 스마트팜에 관한 인식이 높아짐에 따라 국제 표준화가 급격히 확대 될 것으로 예상

- 1.1.1.1.43. - (ITU-T SG13) Future network에서의 서비스시나리오와 관련한 표준항목들을 연구하는 Q1에서는 네트워크 기반의 스마트파밍 개요를 정의하는 국제표준을 세계 최초로 추진하여, 2015년에 Y.4450/Y.2238로 발간. 2019년에는 Y.2243과 Y.2244가 국제표준으로 발간
- (ITU-T SG20) ETRI에서는 국내 스마트팜 업체들의 기술을 반영하기 위해, 스마트팜 서비스를 제공하

기 위한 표준구조, 인터페이스를 정의하는 프레임워크(Y.ISG-FR)표준을 2015년에 제안하여 2020년도에 완료

표 15 - 시설원에 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
ITU-T SG13	Y.2246, Smart Farming Education Service based on u-learning environment	2021
	Y.2245, Service model of the Agricultural Information based Convergence Service	2020
	1.1.1.1.44. Y.2244, Service model for a cultivation plan service at the pre-production stage	2019
	Y.2243, The framework and application model for risk mitigation service based on network	2019
	Y.4450/Y.2238, Overview of Smart Farming based on networks	2015
ITU-T SG20	Y.4466, Framework of Smart Greenhouse Service	2020

(스마트팜 실외(과수, 밭작물 등) 로봇 운용 표준) ISO TC23 중심으로 스마트팜 실외 환경에 필요한 자율주행 농업기계 및 로봇 농작업 제어, 통신, 안전성 및 성능 시험을 위한 국제 표준화를 제정할 것으로 예상

- 1.1.1.1.45. - (ISO TC23 SC2) 농업용 트랙터의 배기가스, 무게중심, 차축동력, 견인력 테스트, 유압 동력, 조향 능력, 저온 시동 등 운행에 관한 시험에 관한 표준 ISO 789 제정 완료하였으며 트랙터 전방, 후방 보호구조물에 대한 정적, 동적시험절차에 대한 ISO 12003, 트랙터 및 농업기계 전반에 전자기 호환성을 평가하기 위한 시험 방법 및 허용 기준에 대한 표준인 ISO 14982가 제정 작업 진행 중
- 1.1.1.1.46. - (ISO TC23 SC3) 자주식 농기계에 대한 운전자 제어장치, 설계원칙, 제동요건 등 주로 안전과 관련된 표준 ISO 4254를 개발하였으며 무인자율주행 트랙터의 기초가 되는 운전자가 제어하는 트랙터의 자동안내시스템의 안전요구사항에 관련된 표준 ISO 18497 제정 중
- 1.1.1.1.47. - (ISO TC23 SC6) 농작물 보호장비에 대한 표준을 담당하여 농업용 드론, 헬기 등을 이용한 무인항공방제사용에 관한 환경요구조건 및 방제시험방법에 대한 표준 ISO 23117 개발 진행 중
- 1.1.1.1.48. - (ISO TC23 SC18) 관개에 관한 원격 모니터링 및 제어기술, 스프링클러의 분사 및 테스트에 관한 표준 ISO 21622 개발 진행 중
- 1.1.1.1.49. - (ISO TC23 SC19) 농업용 트랙터 제어시스템의 기능안전에 관련된 표준 ISO 25119가 2018년도에 제정되었으며, 농임업용 트랙터 및 기계류의 통신 프로토콜 관련하여 데이터 링크 계층, 네트워크 관리, 트랙터 ECU, 진단서비스, 환경 조건에 대한 내성시험 등에 관한 표준인 ISO 23117이 제정 중

표 16 - 농기기/로봇 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
ISO TC23 SC2	ISO/CD 14982, Agricultural and forestry machinery - Electromagnetic compatibility - Test methods and acceptance	진행중 (2022)
	ISO/DIS 12003, Agricultural and forestry tractors - Roll-over protective structures on narrow-track wheeled tractors	진행중 (2021)
	ISO 789, Agricultural tractors - Test procedures	2018
ISO TC23 SC3	ISO 18497, Agricultural machinery and tractors - Safety of highly automated agricultural machines - Principles for design	2018
	ISO 4254, Agricultural machinery - Safety	2018
ISO TC23 SC6	ISO/WD 23117-1, Agricultural and forestry machinery - Unmanned aerial spraying systems	진행중 (2023)
ISO TC23	ISO 21622-1 ~ 4, Remote monitoring and control for irrigation	진행중

개발기구	표준(안)명	개발연도
SC18		(2022)
ISO TC23 SC19	ISO 11783-1~14, Tractors and machinery for agriculture and forestry - Serial control and communications data network	진행중 (2022)
	ISO 25119, Tractors and machinery for agriculture and forestry - Safety-related parts of control systems	2018

1.1.1.1.50. (스마트축사 환경 관리 표준) ITU-T SG20을 중심으로 사물인터넷 기반의 스마트축사 프레임워크에 대한 표준화가 시작되었지만, 스마트축사를 대상으로 요소 기술에 대한 표준화를 진행하는 기구는 아직 없음

1.1.1.1.51. - (ITU-T SG20) 브라질 ANATEL 주도로 IoT에 기반을 둔 스마트축산업을 위한 프레임워크 및 능력을 규정하기 위한 표준(Y.IoT-SLF)을 제안하고, 2022년 최종 승인됨

표 17 - 축사 환경관리 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
ITU-T SG20	Y.4482, Framework and capabilities for Smart Livestock Farming Based on Internet of Things	2022

(축사 사양 관리 표준) ITU-T SG20을 중심으로 사물인터넷(IoT) 기반의 스마트축산업의 개념 및 ICT기술의 적용을 위한 프레임워크에 대한 표준을 진행 중에 있으며, IoT기술의 공급자 측면, 서비스 공급자, 축산 농가 입장에서 계획, 생산, 물류, 추적 등을 다루고 있으며 사양관리는 특히 계획, 생산을 위한 표준으로 아직은 초기 상태

1.1.1.1.51.1. - (ITU-T SG20) 브라질 ANATEL 주도로 IoT에 기반을 둔 스마트축산업을 위한 프레임워크 및 능력을 규정하기 위한 표준(Y.IoT-SLF)을 제안하고, 2022년 최종 승인됨

표 18 - 축사 사양관리 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
ITU-T SG20	Y.4482, Framework and capabilities for Smart Livestock Farming Based on Internet of Things	2022

(동물복지 인증 관리 표준) ITU-T SG20을 중심으로 사물인터넷 기반의 스마트축사 프레임워크에 대한 표준화가 시작되었고, 본 표준에 동물의 고통을 최소화하기 위한 노력으로 실내 온도 조절, 건강 모니터링을 통해 비정상적인 행동을 감지할 수 있는 시스템이 채택되도록 추진 중

1.1.1.1.51.2. - (ITU-T SG20) 브라질 주도로 IoT 기반의 스마트축산업을 위한 프레임워크 및 능력을 규정하기 위한 표준(Y.IoT-SLF)을 제안하였고, 제안한 표준의 응용 서비스 범위에 제안 프레임워크를 활용하여 실내 온도 조절, 건강 모니터링을 통해 동물의 비정상적인 행동을 감지할 수 있음으로써 동물복지에 적용 가능성을 표기

1.1.1.1.51.3. - (ISO TC34 WG16) ISO/TS 34700은 동물복지의 일반원칙, 운송(지상, 바다 그리고 하늘), 도살, 도축, 육우, 식용우, 육용계, 유제품 생산시스템의 내용을 포함. 세계적으로 동물복지에 대한 관심이 증대되고 있고 동물복지 관리를 위한 동물복지 농장 인증 절차, 동물복지 농장 정보 수집 및 운영 방법, 동물복지 데이터 제공 방법 등을 중점 표준화 항목을 선정

표 19 - 동물복지 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
ITU-T SG20	Y.4482, Framework and capabilities for Smart Livestock Farming Based on Internet of Things	2022
ISO TC34 WG16	ISO/TS 34700, Animal welfare management - General requirements and guidance for organizations in the food supply chain	2016

1.1.1.1.52. (수산양식 환경 관리 표준) 현재 수산양식의 환경 관리를 대상으로 표준화를 진행하는 단체는 없지만, ISO TC234에서는 2007년 이후로 수산 양식업 전반에 걸친 표준화를 진행하고 있으며, ITU-T SG20에서는 정보통신기술을 활용한 양식장 관리에 대한 표준화를 준비 중

- 1.1.1.1.52.1. - (ISO TC234) Fisheries and aquaculture 기술위원회가 2007년에 구성되어 수산 양식업에 걸쳐 장비 및 운영, 어류의 화학적·생물학적 상태 관리, 환경 모니터링, 데이터 관리 및 이력 추적 등에 대한 표준화 작업을 진행하고 있으며, 현재까지 9개의 표준 문서를 발간함
- 1.1.1.1.52.2. - (ITU-T SG20) 현재까지 수산양식 환경 관리 관련하여 진행되는 표준 권고안은 없지만, 현재 진행되고 있는 스마트팜 환경 관리 표준과 연계하여 스마트 양식장 시나리오, 요구사항 및 프레임워크에 대한 표준화에 대한 준비가 진행 중

표 20 - 수산양식 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
ISO TC234	ISO/DIS 22948, Carbon footprint for seafood - Product category rules (CFP-PCR)	진행중 (2021)
	ISO 16488:2015, Marine finfish farms - Open net cage - Design and operation	2015
	ISO 16541:2015, Methods for sea lice surveillance on marine finfish farms	2015
	ISO 16741:2015, Traceability of crustacean products - Specifications on the information to be recorded in farmed crustacean distribution chains	2015
	ISO 18537:2015, Traceability of crustacean products - Specifications on the information to be recorded in captured crustacean distribution chains	2015
	ISO 18538:2015, Traceability of molluscan products - Specifications on the information to be recorded in farmed molluscan distribution chains	2015
	ISO 18539:2015, Traceability of molluscan products - Specifications on the information to be recorded in captured molluscan distribution chains	2015
	ISO 12878:2012, Environmental monitoring of the impacts from marine finfish farms on soft bottom	2012
	ISO 12875:2011, Traceability of finfish products - Specification on the information to be recorded in captured finfish distribution chains	2011
	ISO 12877:2011, Traceability of finfish products - Specification on the information to be recorded in farmed finfish distribution chains	2011

(이력 데이터, 추적, 위조방지(블록체인) 표준) GS1 국제 표준을 중심으로 유통 및 물류 비즈니스 데이터에 대한 블록체인 구성 방식에 대한 논의가 진행 중이며, 글로벌 농식품 이력추적 블록체인 플랫폼(예. IBM Food Trust, TE-Food 등)들은 상호 호환성을 위해 GS1 국제표준을 준수. IFT/WWF의 GDST는 국제수산물이력추적 표준인 GDST1.0 완료

- 1.1.1.1.52.3. - (GS1 Blockchain IG) 유통 및 물류 데이터에 대한 온체인, 오프체인 기반 블록체인 데이터 관리 기법에 대해 이슈를 정리하고 있으며, 향후 MSWG로 승격하여 표준과제 진행 예정
- 1.1.1.1.52.4. - (IFT/WWF GDST) 국제수산물 이력추적을 위한 GDST 1.0 표준셋을 GS1 국제표준을 준용해서 개발

표 21 - 농축수산물 이력/추적/위조방지 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
GS1 Blockchain IG	Guideline for Blockchain Application	2020
GDST	Core Normative Standards(v1.0)	2020
	Technical Implementation Guidance(v1.0)	2020
	Basic Universal List of KDEs (spreadsheet)(v1.0)	2020

(유통물류 데이터 표준) 새로운 요구사항과 데이터 기술 발전 추세를 반영하여 신규 EPCIS 표준안에 대

해 표준화가 진행 중

- 1.1.1.1.52.5. - (GS1 EPCIS and CBV 2.0 MSWG) GS1 EPCIS 표준에 대한 Rest-API 바인딩, JSON-LD 데이터 모델, 센서데이터, 인증정보처리 이슈를 포함하여 2.0 표준안에 대해 인터페이스 및 데이터 표준안들을 개정하고 있으며 2020년 완료 예정
- 1.1.1.1.52.6. - (GS1 Event Sharing SMG) GS1 EPCIS 표준을 기반으로 한 수산물 이력추적, 무역 데이터에서의 관리 등의 응용 표준안들이 공표 완료

표 22 - 유통물류 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
GS1 MSWG/SMG	Electronic Product Code Information Service 2.0	2020
	Common Business Vocabulary 2.0	2020
	Foundation for Fish, Seafood and Aquaculture Traceability Implementation	2019
	Foundation for Fish, Seafood and Aquaculture Traceability Guideline	2019
	Fighting Illicit Trade with EPCIS Application Standard	2019
	Global Meat and Poultry Traceability Guideline, Part 1-5 (GS1 System, Beef, Lamb and Sheep, Pork, Poultry)	2015
	Global Meat and Poultry Glossary	2015
	Traceability for Fresh Fruits and Vegetables - Implementation Guide	2015

1.1.1.1.53. (스마트 소비 서비스 표준) GS1 국제표준기구에서는 농축수산식품을 포함하는 상품, 장소, 자산, 사람 등의 식별자를 키로 하여, 원산지, 리콜, 구매, 성분정보 등 다양한 데이터와 서비스를 연결시키는 표준을 개발 중

- (GS1 ONS MSWG) GS1 식별자에 연관된 다양한 서비스를, 글로벌하게 분산 관리되는 ONS에 등록하여, 누구나가 식별자를 이용하여, 서비스를 검색하고, 접근할 수 있는 표준을 제정
- (GS1 Digital Link MSWG) 농축수산식품에 GS1 Digital Link를 부여하고 이를 기반으로 원산지, 리콜, 구매, 성분정보 등의 다양한 ONS, GS1 Digital Link 서비스를 글로벌하게 공유하는 표준을 제정

표 23 - 소비 서비스 분야 국제 표준화 현황

개발기구	표준(안)명	개발연도
GS1 MSWG	Digital Link	2020
	Object Name Service	2013

## 다-2) [표준화 프레임워크] 스마트팜 표준화 체계 수립 및 표준항목 도출 ('19)

### (가) 스마트팜 표준화 프레임워크 개요

스마트팜 분야는 최근 정부의 미래 먹거리 창출을 위한 혁신성장동력 후보로 선정된 핵심 분야로서, 전 세계적으로 농업분야에 대한 혁신을 통한 식량난 해결이 중요한 이슈이다. UN에 따르면 2050년도 국제식량난이 예상되며, 현재 국내 식량자급도 또한 쌀 생산을 제외하고는 30% 하회 중이다.

스마트팜은 생산성 증가는 물론 노동력 절감효과를 통해 농촌 고령화 문제를 해결할 방안으로, 타 분야에 비해 ICT를 통한 생산성 향상이 가장 필요한 분야이나 표준화된 방법이 제공되고 있지 않아 지속적인 중복 투자와 장비·부품 간 상호호환성 문제가 발생하는 분야이기도 하다. 특히 표준 부재로 인해 업체 간 제품호환은 물론 타 회사 제품의 AS 또한 매우 어려운 상황에 놓여 있다. 현재 정부는 '스마트팜'을 혁신성장 선도사업으로 선정, 종합대책 마련하여 기존의 농가 단위의 스마트팜 보급 전략을 보완, 정책대상을 청년 농업인, 전후방 산업으로 확대하고 집적화된 확산거점 조성함으로써, 짧은 기간 내 농업현장을 선도할 ICT 전문 농업인을 육성하고, 생산 혁신을 바탕으로 우리 농업 혁신을 선도하는 거점거점 조성 추진 중이다.

스마트팜 표준화 프레임워크는 환경 변화에 적극 대응하여 융합 산업을 대상으로 정보통신 기술을 활용함으로써 새로운 가치를 창출하는 다양한 IT 융·복합 서비스를 발굴하고, 이를 지원하는 표준화 항목을 도출하기 위해 개발되었다. 이와 관련하여, 스마트팜 기술의 보급 확산을 위한 표준화 프레임워크를 개발하여 우리나라 농촌의 성장 모멘텀을 꺾을 수 있도록 다양한 분야에서 스마트팜 표준화 항목을 도출하고, 이들 중 5개 항목을 선별하여 이에 대한 전략을 수립하였다.

이를 위해 스마트팜 표준화 프레임워크는 다음과 같이 개발되었다.

- 표준화 프레임워크 개발 분야에 대한 최상위의 비즈니스 영역(business domain)을 정의함으로써 현실의 비즈니스 세계에 대한 직관적이고 공통된 이해를 제공하고
- 표준화 프레임워크 개발에 참여하는 다양한 이해당사자 간에 원활한 의사소통을 도모하고 개념 전달과 습득의 용이성을 촉진
- 표준화 프레임워크의 비즈니스 지원 시스템 모델 개발, 서비스 정의, 표준 갭 분석 작업을 위한 기반을 제공

스마트팜 표준화 프레임워크는 ICT 융합 표준화 프레임워크의 개발 분야를 대상으로 다양한 비즈니스 및 서비스가 이루어지는 현실 세계의 비즈니스 생태계(business ecosystem)를 가치 흐름에 따라 추상화한 모형을 기술함으로써, 궁극적으로는 표준화 활동이 시장 생태계를 조성하고 새로운 시장 창출과 산업 활성화, 나아가 국가경쟁력 강화에 기여할 것으로 기대된다.

### (나) 스마트팜 표준화 프레임워크 추진현황

#### ■ 비전 및 전략

스마트팜 표준화 프레임워크에서는 정부의 국정과제와 정책 관련 유관기관의 정책을 기반으로 스마트팜 분야에서 추구하는 사회·경제적 미래상으로서의 비전으로서, 기존 시설원예 환경에 ICT 기술을 접목하여 ① 작물 생육과 환경 데이터를 기반으로 최적 생육환경 조성하고 품질 및 생산성을 제고하고, ② 노동력, 에너지, 농기자재 등의 투입을 최소화함으로써 비용 절감 및 환경 보전에 기여하고, ③ 시스템에 의한 정밀농업을 통해 단순 반복 작업 및 위험한 노동에서의 해방을 제시하는 것을 목적으로 한다.

개발 목적에서 확인된 사회 문제 해결을 도모하기 위해 각 문제에 대응하여 설정된 중점 추진 영역은 현재 상황에서 비전을 달성하기 위해서 현실의 문제 해결을 위한 새로운 서비스가 도출되는 영역으로서, 정부의 정책 방향과 시장 파급효과 및 실현 가능성 등을 고려하여 설정하였다.

- 최적 생육환경 제어를 통한 생산성 및 품질 보장 농업환경
- 작물의 영양과 질병 상태를 조기진단 및 처방하는 농업환경
- 언제 어디서나 제어 가능한 시공간 제약 없는 농업환경
- 복합 에너지 최적 제어기술 및 지능형 농기계를 통한 자동화 농업환경

#### ■ 개념모델 수립

스마트팜 표준화 프레임워크의 개발 분야를 대상으로 다양한 비즈니스 및 서비스가 이루어지는 현실 세계의 비즈니스 생태계(business ecosystem)를 가치 흐름에 따라 추상화한 모형인 개념 모델은 스마트팜 분야의 특징, 요구사항, 기술, 서비스 및 표준 등을 논의하기 위한 기반으로 최상위 수준에서 전체 상황을 이해하는데 있어서 중요하다.

#### ■ 비즈니스 지원 시스템 모델 수립

개념 모델에서 정의된 도메인 내에서 이루어지는 다양한 비즈니스와 도메인 간의 상호 작용을 기술적으로 지원하는 전체 시스템을 모형화한다.

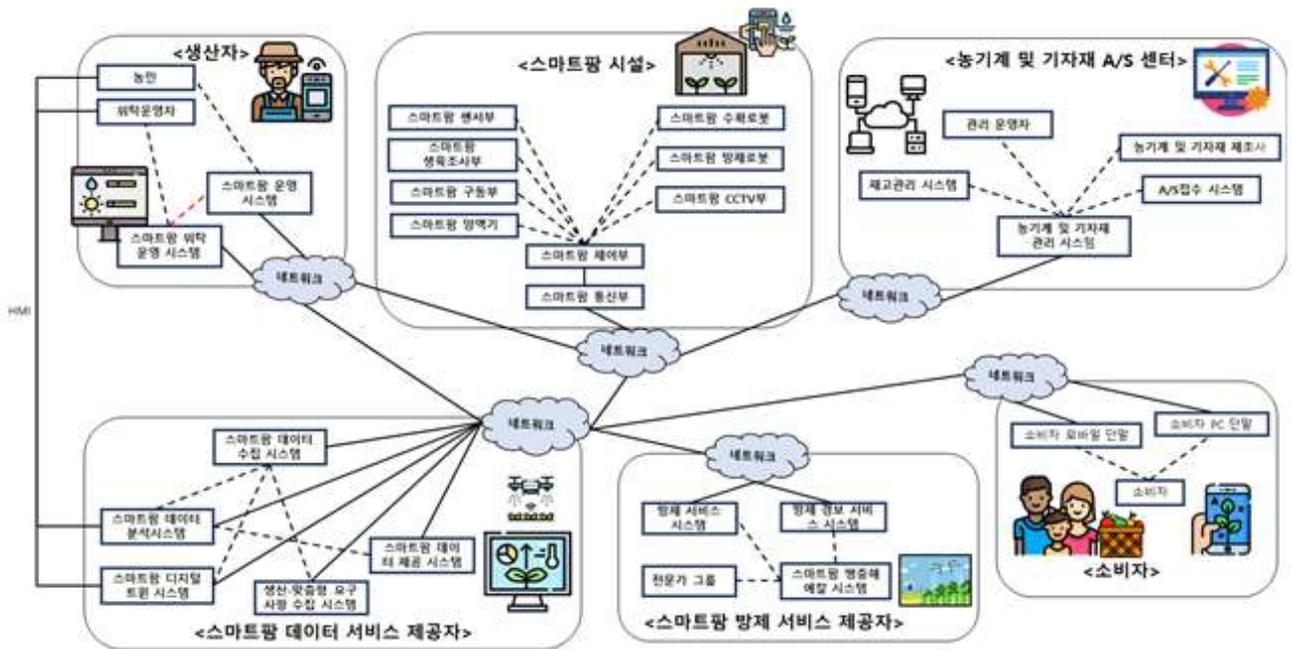


그림 1 - 비즈니스 지원 시스템 모델 다이어그램

■ 서비스 정의

시장의 니즈를 기반으로 향후 다양한 이해당사자에게 높은 시장 가치를 제공할 것으로 예상되는 잠재 서비스를 식별하고, 서비스 제공을 위해 필요한 기술적·기능적 요구사항을 기반으로 각 서비스 별로 유즈 케이스를 개발한다.

표 1 - 중점 추진 영역별 서비스 현황

서비스 패키지	단위 서비스 유즈케이스
알고리즘 기반 스마트팜 자동 제어 서비스	.정밀농업을 위한 온실 제어 알고리즘 제공 서비스 .알고리즘 기반 원격 온실 제어 서비스 .소비자 요구 맞춤형 및 특용 작물 생산 서비스
스마트팜 운영 및 생산 지원 서비스	.조합형 농장 위탁 운영 서비스 .디지털 트윈형 스마트팜 제어 서비스 .생산 예측 서비스
스마트팜 생산 자동화 서비스	.생산 장비 유지보수 알림 서비스 .농산물 자동 수확 서비스 .병해충 예찰 및 경보 서비스 .무인 방제 서비스

※ 서비스 패키지: 단위 서비스의 묶음이나 집합을 의미

■ 표준 갭분석

표준 갭 분석은 서로 다른 기기나 시스템 간에 서비스, 정보 및 데이터를 원활하게 공유, 교환 및 운용 되도록 상호운용성을 보장하는 서비스 제공이 가능하도록 서비스 정의의 각 단위 서비스를 대상으로 표준화 요구사항을 식별하고 이를 충족하는 표준을 도출하는 작업으로서, 도출된 표준은 현재 이용 가능한 식별 표준과 향후 개발이 요구되는 잠재표준으로 구분한다. 서비스 정의를 통해 도출된 서비스 패키지 및 단위 서비스 유즈케이스들을 토대로 각 서비스 구현을 위한 표준 목록을 도출한다.

- 서비스 유즈케이스별 표준 프로파일을 수립하고 이를 통해 도출된 표준화 항목을 기반으로 표준화 요구사항에 따라 △스마트팜 구조 및 설치 △스마트팜 서비스 유즈케이스 및 요구사항 표준 △용어, 데이터 모델, 메타데이터 표준 △스마트팜 내부 제어 프로토콜, 인터페이스 △스마트팜 외부 연동 프로토콜, 인터페이스 △스마트팜 유저 인터페이스 △팜 클라우드 서비스 △농기계, 기자재 관리 표준 범주로 분류
- 스마트팜 표준화 프레임워크에서 제안된 유즈케이스를 실현하기 위해 신규·보완 개발이 필요한 잠재표준 목록을 도출

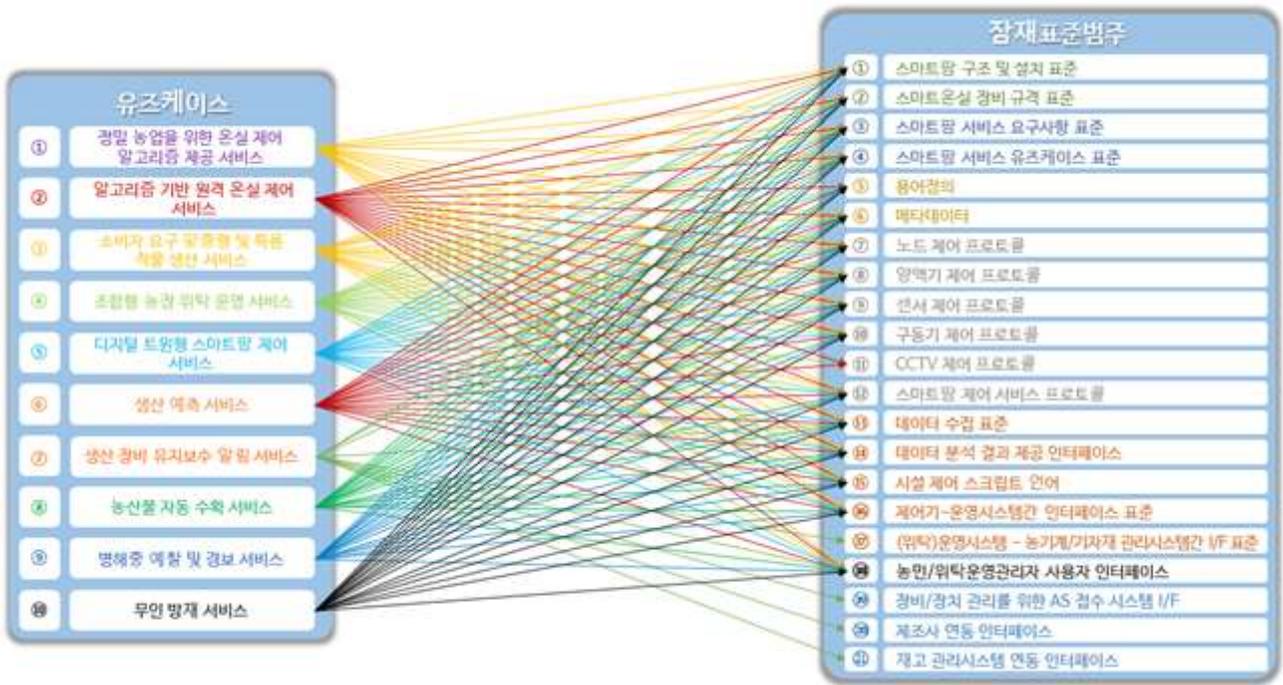


그림 2 - 스마트팜 표준화 프레임워크 서비스 유즈케이스와 범주 매핑

표 2 - 스마트팜 표준화 프레임워크 잠재표준 목록

범주	우선순위 (H/M/L)	잠재표준 항목	추진 중인 표준개발기구
스마트팜 구조 및 설치 표준	H	스마트온실 설치 지침서 - 제1부: 통신기술	TTA PG426
	H	Y.ISG-FR, Framework of Smart Greenhouse Service	ITU-T SG20 Q4
스마트온실 장비 규격 표준	H	스마트 온실 ICT 융복합 장비규격 및 서비스 기준	TTA PG426
	H	스마트 온실을 위한 원격 감시용 스마트 영상 장치	TTA PG426
스마트팜 서비스 요구사항 표준	H	Y.smpp, Service model for the pre-production stage on Smart Farming	ITU-T SG13 Q1
	H	Y.farms, The framework and application model for risk mitigation service based on network	ITU-T SG13 Q1
	H	Y.saic, Service model of the Agricultural Information based Convergence Service	ITU-T SG13 Q1
	M	온실제어 알고리즘 제공 서비스를 위한 알고리즘 제공자, 운영시스템 기능 요구사항	-
	M	스마트팜 데이터 제공 시스템 : 시설제어 알고리즘 제공 기능 요구사항	-
	M	스마트팜 데이터 분석 시스템 : 기능 요구사항	-
	M	농기계 및 기자재 관리 시스템 : 기능 요구사항	-
스마트팜 서비스 유즈케이스 표준	M	온실 제어 알고리즘 제공 서비스 절차 및 요구사항	-
	M	알고리즘 기반 원격 온실 제어 서비스 절차 및 요구사항	-
	M	조합형 농장 위탁운영 서비스 절차 및 요구사항	-
	M	디지털 트윈형 스마트팜 제어 서비스 절차 및 요구사항	-
	M	생산예측 서비스 절차 및 요구사항	-
	M	생산 장비 유지보수 알림 서비스 절차 및 요구사항	-
용어정의	H	스마트팜 용어 정의 - 제1부: 온실	TTA PG426

범주	우선순위 (H/M/L)	잠재표준 항목	추진 중인 표준개발기구
메타데이터	H	농기계 메타데이터	-
	H	스마트팜 장비/장치/농기계 운영정보/수리 이력 메타데이터	-
	H	스마트온실 장비 및 장치 메타데이터 정의 및 포맷 기술	-
	H	스마트온실 시설정보 메타데이터 정의 및 포맷 기술	-
	H	작물 정보 메타데이터 정의 및 포맷 기술	-
	H	병해충 예찰관련 메타데이터 (병해충, 농약, 병충, 방제방법, 영농처방 등)	-
노드 제어 프로토콜	H	경량형 제어프로토콜(LCP)을 위한 IP 적응 계층(IPAL) 동작 요구사항	TTA PG426
	H	모드버스/485 기반 스마트온실 노드/디바이스 등록 절차 및 기술 규격	TTA PG426
	H	스마트팜 장비 연동을 위한 디바이스 드라이버 어플리케이션 프로그래밍 인터페이스	TTA PG426
양액기 제어 프로토콜	H	스마트온실에서 온실 통합 제어기와 양액기 노드 간 RS485 기반 모드버스 인터페이스	TTA PG426
센서 제어 프로토콜	H	UHF 대역 400MHz 무선링크 기반 스마트 온실 제어 프로토콜	TTA PG426
	H	센서 노드와 온실통합제어기간 생육상태 모니터링 인터페이스	-
구동기 제어 프로토콜	H	UHF 대역 400MHz 무선링크 기반 스마트 온실 제어 프로토콜	TTA PG426
CCTV 제어 프로토콜	H	AI기반 작물 상태 및 생육 정보 획득 방법에 대한 지침	-
	M	AI기반 작물 상태 및 생육정보 성능 시험 방법	-
	M	AI기반 작물 상태 및 생육정보 획득 품질 측정 방법	-
스마트팜 제어 서비스 프로토콜	H	스마트온실 서비스 제어 프로토콜: 기본 서비스 절차	TTA PG426
	M	스마트온실 서비스 제어 프로토콜 : 시간 제어 서비스 절차	-
	M	스마트온실 서비스 제어 프로토콜 : 비율 제어 서비스 절차	-
데이터 수집 표준	M	시설, 환경, 생육 상태정보 요청 및 전달 인터페이스 및 프로토콜	-
	M	장비/장치 상태정보 및 실시간 운행데이터 전달 인터페이스 및 프로토콜	-
데이터 분석 결과 제공 인터페이스	M	스마트팜 시설 제어 알고리즘 제공 표준	-
	M	제품을 교체하더라도 일관적인 UX를 제공하기 위한 시설(장비,장치)/환경 관련 정보 시각화 가이드라인	-
	L	제품을 교체하더라도 일관적인 UX를 제공하기 위한 생육상태정보 시각화	-
시설 제어 스크립트 언어	H	스마트 온실 제어 스크립트 언어	TTA PG426
제어기-운영시스템 간 인터페이스 표준	M	스마트팜 제어 알고리즘 탑재 프로토콜	-
	L	시설정보, 장비정보, 작물 정보 등록	-
	M	방제장치 상태 및 제어 프로코콜	-
운영시스템/위탁운영 시스템 - 농기계/기자재 관리 시스템간 인터페이스 표준	M	스마트팜 장비/장치 이상징후 판단 기준 및 알림 절차	-
	M	스마트팜 장비/장치의 항상성 유지를 위한 자동 AS 점검, 장치 교체 요청 등을 자동화 하기 위한 표준화된 인터페이스	-

범주	우선순위 (H/M/L)	잠재표준 항목	추진 중인 표준개발기구
농민/위탁운영관리자 사용자 인터페이스	M	시설정보, 장비정보, 작물정보 등 등록	-
	M	시설원에 제어 알고리즘/레시피 제공 및 승인 관련 절차	-
	L	알고리즘 기반 제어 정보 / 재배결과 분석정보 / 모니터링 시각화 가이드라인	-
	L	신규 특작모델 요구/수신/평가 절차	-
	L	출하시기별 예측생산량 품질정보시각화 가이드라인	-
	L	시설(장비,장치)/환경 유지보수 관련 알림/조치 가이드라인	-
	L	작물자동수확 절차 승인/결과수신 절차	-
	H	병해충 방제 경보 수신 절차	-
	M	병해충 방제지시/결과수신 절차	-
	M	재배작물/스케줄 정보 입력 인터페이스	-
	M	양액/온실환경 관리 레시피 입력 인터페이스	-
	M	작물 생산량 정보 입력 인터페이스	-
	M	직관적 제어명령 인터페이스	-
	H	Y.sfes, Smart Farming Education Service based on u-learning environment	ITU-T SG13 Q1
H	농기계 오작동 알림 설정 인터페이스	-	
데이터 제공 표준	-	표준화 항목 설명	-
장비/장치 관리를 위한 AS 접수 시스템 인터페이스	M	이종 제조사 단말들의 자동 A/S 접수를 위한 표준화된 A/S 센터 연동 표준	-
제조사 연동 인터페이스	M	제조사에게 특정 장비/장치의 A/S 가능 여부 및 부품 보유 현황 조회를 위한 인터페이스	-
	M	시스템에서 재고 부품 부족시 자동으로 제조사에게 장비/장치/부품의 주문 발주를 위한 인터페이스	-
재고 관리시스템 연동 인터페이스	M	지역별 농기계 A/S 센터에서 유지하는 부품들의 재고현황을 확인하기 위한 인터페이스	-

(다) 표준화 항목 도출 및 국내외 표준화 추진 전략

국내·국제 표준화 전략 및 표준화 추진의 효과적 달성을 위해 이 잠재표준 목록에서 표준개발의 선도 가능성, 시장 파급효과 및 시급성을 고려하여, 다음과 같이 우선 추진 잠재표준을 선정하였다.

표 3 - 스마트팜 표준화 프레임워크 우선 추진 잠재표준 목록

범주	우선 추진 잠재표준 항목	추진 중인 표준개발기구
메타데이터	스마트팜 장비/장치/농기계 운영정보/수리 이력 메타데이터	-
	스마트온실 시설정보 메타데이터 정의 및 포맷 기술	-
	작물 정보 메타데이터 정의 및 포맷 기술	-
센서/구동기 제어 프로토콜	UHF 대역 400MHz 무선링크 기반 스마트 온실 제어 프로토콜	TTA PG426
CCTV 제어 프로토콜	AI기반 작물 상태 및 생육 정보 획득 방법에 대한 지침	-

## 라) 스마트팜 표준 확산을 위한 해설서 개발

### 라-1) [표준활용 가이드라인] 국내·국제 스마트팜 표준 해설 및 활용 가이드라인 ('21)

#### (가) 연구 수행 개요

국내에서 스마트팜 기술에 대한 표준화는 한국정보통신기술협회(TTA)의 PG426(스마트농축수산 프로젝트그룹)을 중심으로 스마트팜(온실, 축사, 노지 등) 환경관리에 사용되는 ICT 장치간 호환성 및 연동성 제공을 위한 표준화가 진행되고 있으며, 현재 국가표준으로 전환된 표준을 포함하여 약 69종의 표준이 개발되었다. 또한 2021년에는 수산양식 분야에 대한 표준화로 그 범위를 확대하고 명실공히 스마트팜 표준화의 선두주자로서 자리매김하고 있다. 국외의 경우 ISO와 ITU-T 등의 국제 공적 표준화 기구를 주축으로 스마트팜 관련 표준화가 진행되고 있으며 일부 사실 표준화(de facto) 기구에서도 스마트팜에 대한 표준화에 관심을 가지고 있으나 아직은 시작 단계에 머무르고 있다.

2020년 TTA와 PG426은 국내외에서 개발된 또는 개발되고 있는 표준 규격의 현장 적용성을 제고하기 위하여 스마트팜에 대한 기술적인 개요와 함께 온실, 축사, 데이터 분야와 관련된 표준화 현황과 주요 표준에 대한 해설서를 개발하기로 하였으며, 적용 분야별 표준활용을 위한 맵에 대한 최종 보완 작업을 마무리되어 2021년에 발간하였다.

#### (나) 연구 주요 내용

스마트팜 표본 해설 및 활용 가이드 라인의 주요 내용은 다음과 같다.

1. 스마트팜 분야 표준 활용 맵
  - 가. 스마트팜 개요
  - 나. 스마트팜 국내외 표준화 동향
2. 적용 분야별 스마트팜 표준 현황
  - 가. 시설원예 분야
  - 나. 노지 분야
  - 다. 축사 분야
  - 라. 클라우드/빅데이터 분야
3. 스마트팜 관련 주요 표준 해설서
  - 3.1. 시설원예 분야
    - 가. UHF 대역 400 MHz 무선링크 기반 스마트 온실 제어 프로토콜
    - 나. 스마트 온실에서 온실통합제어기와 양액기 노드 간 RS485 기반 모드버스 인터페이스
    - 다. 모드버스/RS485 기반 스마트온실 노드/디바이스 등록 절차 및 기술 규격
    - 라. 스마트 온실 센서/구동기 노드 및 온실 통합 제어기간 RS485 기반 모드버스 인터페이스
    - 마. 스마트 온실을 위한 센서 인터페이스
    - 바. 스마트 온실을 위한 구동기 인터페이스
    - 사. 온실 관제 시스템 요구사항 프로파일
    - 아. 온실 관제 시스템 (센서노드/제어노드-통합제어기-운용시스템-관리시스템 인터페이스)
  - 3.2. 노지 분야
    - 가. 자율주행 농기계 시험
    - 나. 농업 정보 통신
    - 다. 농업용 드론
  - 3.3. 축산 분야
    - 가. 축산분야 ICT 융복합 장비규격 및 서비스 요구사항
    - 나. 스마트 축사를 위한 외기/내기 센서 인터페이스
    - 다. 스마트 축사를 위한 외기/내기 센서 메타데이터 구조
  - 3.4. 데이터 서비스 분야
    - 가. 클라우드 기반 스마트팜 영농작업관리 서비스 인터페이스
    - 나. 클라우드 기반 온실의 장비 오작동 대응 서비스 인터페이스
    - 다. 클라우드 기반 스마트팜 서비스 요구사항

#### 1) 시설원예 분야 표준 활용맵

스마트팜의 적용 분야 중 시설원예 분야를 위한 표준 활용 맵의 예시 다음 그림과 같으며, 표준기구별 표준 현황은 다음과 같다.

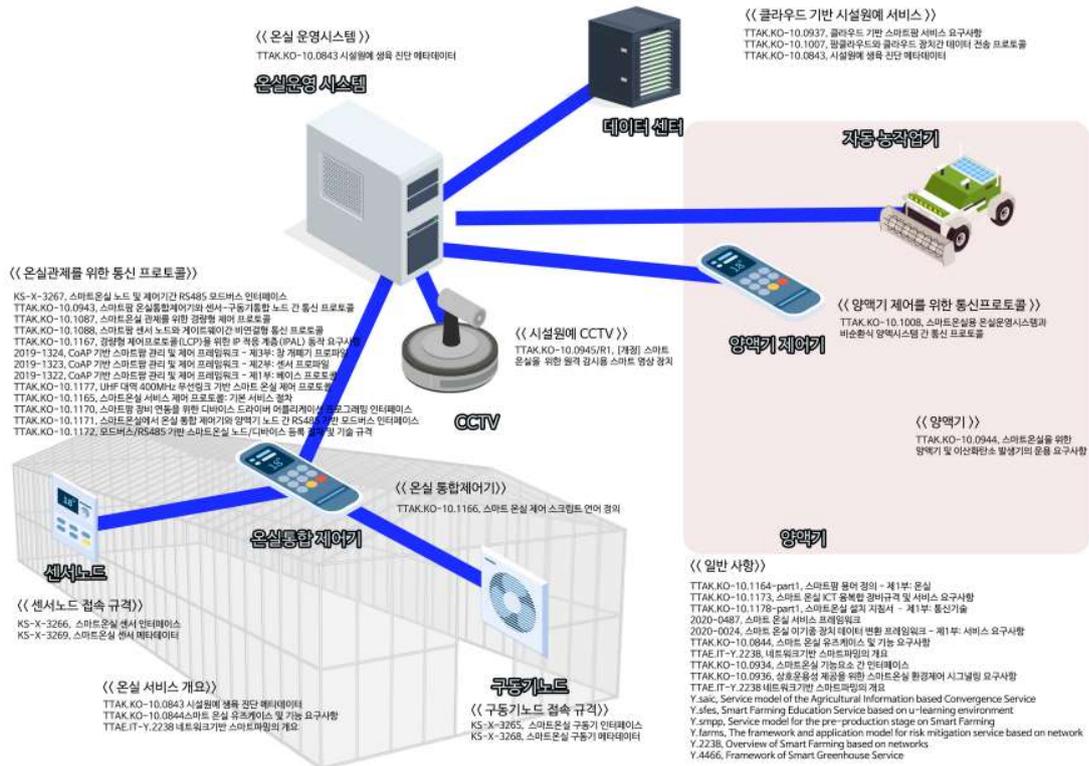


그림 1 - 시설원에 분야 표준 활용 맵

표 1 - 시설원에 분야 TTA 표준 현황

번호	표준번호	표준명	개발년도
1	TTAK.KO-06.0286	온실 관제 시스템 요구사항 프로파일	2012
2	TTAK.KO-06.0288-Part1/R1	온실 관제 시스템-제1부 센서노드와 온실통합제어기간 인터페이스	2015
3	TTAK.KO-06.0288-Part2/R1	온실 관제 시스템-제2부 제어노드와 온실통합제어기간 인터페이스	2015
4	TTAK.KO-06.0288-Part3	온실 관제 시스템-제3부 온실통합제어기와 온실운영시스템간 인터페이스	2012
5	TTAK.KO-06.0288-Part4	온실 관제 시스템-제4부 온실운영시스템과 온실 통합관리시스템간 인터페이스	2013
6	TTAK.KO-10.0843	시설원에 생육 진단 메타데이터	2015
7	TTAK.KO-10.0844	스마트 온실 유즈케이스 및 기능 요구사항	2015
8	TTAK.KO-10.0845 (KS X 3265)	스마트 온실을 위한 구동기 인터페이스	2015
9	TTAK.KO-10.0903 (KS X 3266)	스마트온실을 위한 센서 인터페이스	2016
10	TTAK.KO-10.0934	스마트온실 기능요소 간 인터페이스	2016
11	TTAK.KO-10.0936	상호운용성 제공을 위한 스마트온실 환경제어 시그널링 요구사항	2016
12	TTAK.KO-10.0937	클라우드 기반 스마트팜 서비스 요구사항	2016
13	TTAK.KO-10.0943	스마트팜 온실통합제어기와 센서-구동기통합 노드 간 통신 프로토콜	2016
14	TTAK.KO-10.0944	스마트온실을 위한 양액기 및 이산화탄소 발생기의 운용 요구사항	2016
15	TTAK.KO-10.1008	스마트온실용 온실운영시스템과 비순환식 양액시스템 간 통신 프로토콜	2017
16	TTAK.KO-10.1007	팜클라우드와 클라우드 장치간 데이터 전송 프로토콜	2017
17	TTAK.KO-10.1044 (KS X 3267)	스마트 온실 센서/구동기 및 제어기 간 RS485 기반 모드 버스(MODBUS) 인터페이스	2018
18	TTAK.KO-10.1046 (KS X 3269)	스마트 온실 센서 메타데이터	2018
19	TTAK.KO-10.1045 (KS X 3268)	스마트 온실 구동기 메타데이터	2018

20	TTAK.KO-10.1087	스마트온실 관제를 위한 경량형 제어 프로토콜	2018
21	TTAK.KO-10.1088	스마트팜 센서 노드와 게이트웨이간 비연결형 통신 프로토콜	2018
22	TTAK.KO-10.1164-part1	스마트팜 용어 정의 - 제1부: 온실	2019
23	TTAK.KO-10.1165	스마트온실 서비스 제어 프로토콜: 기본 서비스 절차	2019
24	TTAK.KO-10.1166	스마트 온실 제어 스크립트 언어 정의	2019
25	TTAK.KO-10.1167	경량형 제어프로토콜(LCP)을 위한 IP 적응 계층(IPAL) 동작 요구사항	2019
26	TTAK.KO-10.1170	스마트팜 장비 연동을 위한 디바이스 드라이버 어플리케이션 프로그래밍 인터페이스	2019
27	TTAK.KO-10.1171	스마트온실에서 온실 통합 제어기와 양액기 노드 간 RS485 기반 모드버스 인터페이스	2019
28	TTAK.KO-10.1172	모드버스/RS485 기반 스마트온실 노드/디바이스 등록 절차 및 기술 규격	2019
29	TTAK.KO-10.1173	스마트 온실 ICT 융복합 장비규격 및 서비스 요구사항	2019
30	TTAK.KO-10.1177	UHF 대역 400MHz 무선링크 기반 스마트 온실 제어 프로토콜	2019
31	TTAK.KO-10.1178-part1	스마트온실 설치 지침서 - 제1부: 통신기술	2019
32	TTAK.KO-10.0945/R1	스마트 온실을 위한 원격 감시용 스마트 영상 장치	2019
33	TTAK.KO-10.1244-Part1	스마트 온실 이기종 장치 데이터 변환 프레임워크 - 제1부: 서비스 요구사항	2020
34	TTAK.KO-10.1243-Part1	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제1부: 베이스 프로토콜	2020
35	TTAK.KO-10.1243-Part2	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제2부: 센서 프로파일	2020
36	TTAK.KO-10.1243-Part3	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제3부: 창 개폐기 프로파일	2020
37	TTAK.KO-10.1243-Part4	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제4부: 돈사 사양관리기	2021
38	TTAK.KO-10.1243-Part5	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제5부: 돈사 환경관리기	2021
39	TTAK.IT-Y.4466	스마트 온실 서비스 프레임워크	2021
40	TTAK.KO-10.1245-Part1	스마트팜 데이터 공유를 위한 API - 제1부: 요구 사항	2020
41	TTAE.IT-Y.2238/R1	네트워크기반 스마트팜의 개요	2020

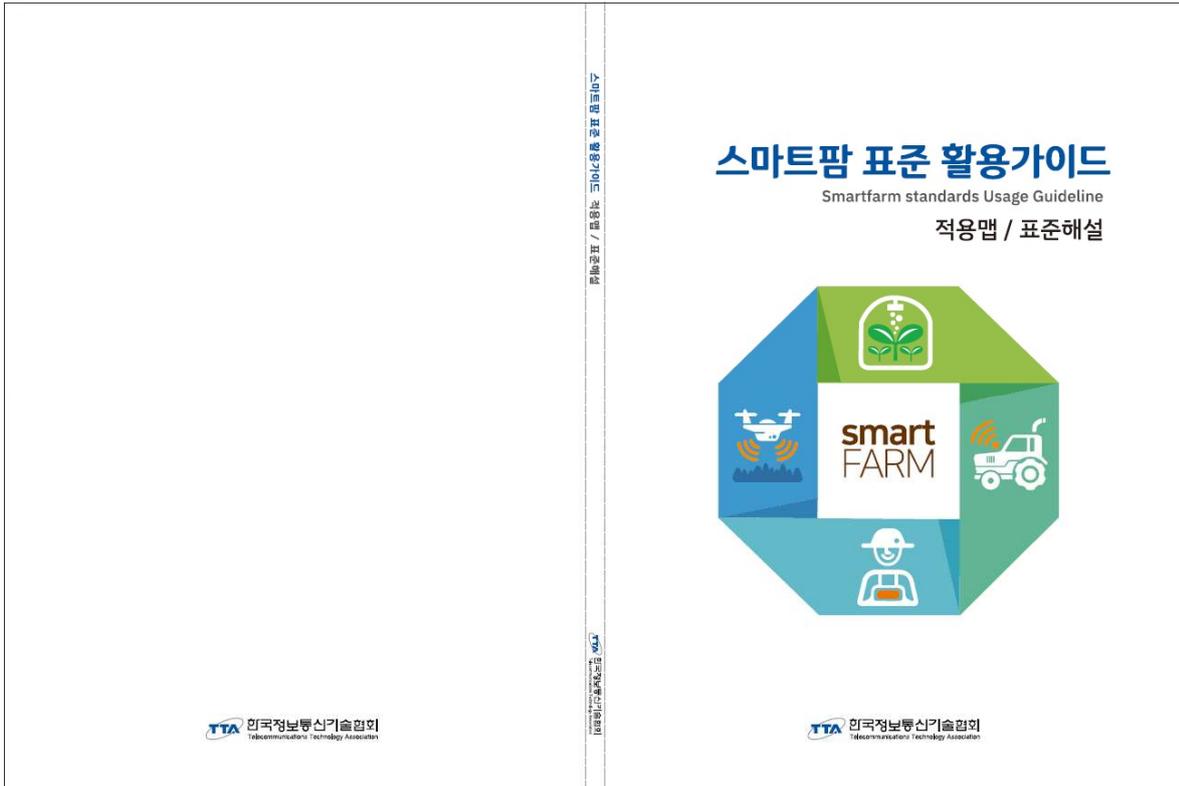
표 2 - 시설원에 분야 KS 표준 현황

번호	표준번호	표준(안)명	개발년도
1	KS-X-3265	스마트온실 구동기 인터페이스	2018
2	KS-X-3266	스마트온실 센서 인터페이스	2018
3	KS-X-3267	스마트온실 노드 및 제어기간 RS485 모드버스 인터페이스	2018
4	KS-X-3268	스마트온실 구동기 메타데이터	2018
5	KS-X-3269	스마트온실 센서 메타데이터	2018
6	KS-X-3286	RS485/모드버스 기반 스마트 온실 노드/디바이스 등록 절차 및 기술 규격	2021
7	KS-X-3288	스마트 온실의 온실 통합 제어기와 양액기 노드 간 RS485 기반 모드 버스 인터페이스	2021
8	KS-X-3287	스마트 온실 노드 메타데이터	2022

표 3 - 시설원에 분야 국제 표준 현황

번호	표준번호	표준(안)명	개발년도	비고
1	Y.ous	Overview of unmanned smart farm based on networks	진행중 (2023)	ITU-T Q1/13
2	Y.ems	Service model for entry-level smart farm	진행중 (2022)	ITU-T Q1/13
3	1.1.1.1.54, Y.2245	1.1.1.1.55, Service model of the Agricultural Information based Convergence Service	2020	ITU-T Q1/13
4	Y.2246	Smart Farming Education Service based on u-learning environment	2021	ITU-T Q1/13

5	1.1.1.1.56. Y.2244	1.1.1.1.57. Service model for a cultivation plan service at the pre-production stage	2019	ITU-T Q1/13
6	Y.2243	The framework and application model for risk mitigation service based on network	2019	ITU-T Q1/13
7	Y.4450/Y.2238	Overview of Smart Farming based on networks	2015	ITU-T Q1/13
8	Y.4466	Framework of Smart Greenhouse Service	2020	ITU-T Q4/20
9	Y.Sup.SmartAgri-usecase	Use cases of IoT based smart agriculture	진행중 (2022)	ITU-T Q2/20



마) [홍보·전시] 스마트팜 기술 및 표준화 동향 홍보

본 사업기간을 통해 다음 표와 같이 국내 전문지를 통해 14건의 스마트팜 표준화 홍보를 추진한 바 있다.

표 1 - 스마트팜 기술 및 표준화 홍보 추진 현황

No	매체명	제목	일시
1	한국농기계신문	스마트팜 ICT 표준화 과정 시급하다	2017
2	정책브리핑	과기정통부, 글로벌 ICT 표준콘퍼런스(GISC2018)개최	2018
3	농기계신문	농업분야 산업혁명 주도하기 위해선 표준화가 필수요소이다.	2019
4	농기계신문	전화 요금 17만 원의 비밀...	2019
5	농기계신문	스마트팜 인공지능 준비를 위한 매우 중요한 시점	2019
6	한국농기계신문	2020년, 스마트팜 표준화 재정비가 필요한 해	2020
7	한국농기계신문	가축방역통합시스템 세계 기술 선도	2020
8	한국농기계신문	유통구조 개선, 농가소득 안정화 촉진	2020
9	한국농기계신문	포스트-코로나 시대, 이젠 기후변화다	2020
10	한국농기계신문	홍보1-스마트팜 데이터 활성화를 위한 제언	2021
11	한국농기계신문	홍보2-스마트팜 글로벌화를 위한 제언	2021
12	한국농기계신문	홍보3- 기후변화 바퀴 속도를 늦출 수 있는 건 오직 탄소 절감뿐	2021
13	한국농기계신문	우리나라 스마트 농기계의 경쟁력 강화를 위한 제언	2022
14	농축산기계신문	“농기계의 통신기술, 체계적 접근해야”	2022

다음은 이들 홍보 기사중 선도성이 있는 일부 기사들을 선별하여 소개하도록 한다.

“2020년, 스마트팜 표준화 재정비 필요”라는 기고를 통해, 농가가 자부담으로 스마트팜 기자재를 구입할 수 있을 정도의 품질을 마련하기 위해서는, 표준화가 매우 필요하다라는 내용을 강조하였다.



한국전자통신연구원  
표준연구부부장

## 2020년, 스마트팜 표준화 재정비 필요

벌써 2020 경자년이 시작될 지도 1달이라는 시간이 흘렀다. 2014년 말부터 본격적으로 시작되었던 스마트팜 표준화 분야도 벌써 5년이라는 시간이 흐르게 되었다. 그간 상호호환이 가능한 한국형 스마트팜 기자재를 만들자는 구호 아래 많은 표준이 제정되었다. 하지만 일부에서는 이러한 성과에 대해 적지 않은 아쉬움을 표하기도 하였다.

이러한 아쉬움이 생긴 이유는 일단 표준만 만들어진다면 스마트팜이 내재한 현재의 모든 문제점을 한꺼번에 해결할 수 있다는 기대감, 그리고 표준을 준용하지 못하는 기존의 제품은 보급 사업에 참여하지 못하는 것이 아니냐는 불안감에 기인한 것이 아닐까 한다.

스마트팜 기자재 표준화에 대한 따가운 시선을 느낀 필자는 과거 수년간 참여하였던 스마트팜 표준화 과정을 곁집념 중, 어릴 적 EIS 마크가 없는 연필은 불만 연필이라고 인식했던 것과는 달리 얼마전 구매한 유명 연필에는 EIS 마크가 없다는 놀라운 사실을 발견했다. 이러한 발견이 놀라

움으로 다가온 이유는 표준을 강제인증제도와 동일시했기 때문일 것이다. 표준이란 자동차 배출가스나 인체 유해성 물질을 강제하기 위한 강제인증제도와는 달리 사용자들의 편익을 위해 이해 당사자 간 상호 동의로 만들어지고 사용되는 것이다. 덕분에 상호호환이 가능한 기자재 생산을 통하여 국내 관련 산업체의 선대와 집중을 가능하도록 하고 이를 통한 규모의 경제와 해외 진출을 위한 기술 초석이 될 것이라는 스마트팜 기자재 표준화에 대한 필자의 믿음이 흔들림이 없다는 것을 확인하게 되었다. 하지만 표준화를 위한 방법론에서 수정이 필요하다는 것에는 수긍할 수밖에 없었다.

필자는 스마트팜 기자재 표준화를 위한 방법으로 다음 다섯 가지 제언을 하고자 한다. 첫 번째로 스마트팜 표준화 대상에 대한 인식을 정리해야겠다. 모든 스마트팜 기자재에 대해 획일적인 표준화보다는 제품의 고유 특성을 수용할 수 있는 다양성이 인정되는 표준화가 이루어져야 한다. 두 번째로 관련 기관 간에 분리된 고유 역할 정립을 통해 스마트팜 표준화에 대한 효율성을 높이고 혼선을 방지해야 한다. 표준 제정, 제품 구현, 시험인증, 기술 확산 기능이 분리되어 상호 협력해야만 다수의 의견이 반영된 표준

화가 이루어질 수 있을 것이다. 세 번째로 산업체들의 적극적인 의견 개진을 통해 표준의 부족한 면만 강조하지 말고 부족한 부분을 제외야 한다. 이를 위해 산업체들은 표준에 대한 이해도를 높이고 부족한 부분을 관련 공청회 등에 적극적으로 참여하여 의견을 개진할 필요가 있다. 네 번째로 정부의 유연한 표준 지원 정책 마련이 필요하다. 표준화에 대한 강력한 정부의 의지에 전적으로 동의한다. 하지만 표준화를 추진하는 진행 방법에 있어서 국내 스마트팜 산업체들의 어려움을 파악하여 단계적인 표준화 정책을 제시하는 것이 필요하다. 마지막으로 보급사업을 주요 매출원으로 하는 현재의 스마트팜 생태계가 건강해질 필요가 있다. 즉 농가가 자비용을 부담하면서까지 스마트팜 기자재를 구매할 수 있는 분위기가 조성되어야 한다. 이를 위해서는 국내 스마트팜 기자재에 대한 객관적인 신뢰도를 높이고 농가의 수익이 증대한다는 객관적인 사례가 필요하다. 자발적으로 구매할 수 있는 제품이어서 말로 해외에서도 경쟁력을 확보할 수 있을 것이기 때문이다.

끝으로 필자는 농가가 자부담으로 스마트팜 기자재를 구매할 수 있는 제품 생산에 도움이 되는 표준을 만들 수 있는 새해가 되길 소망한다.

“가축방역통합시스템 세계 기술 선도”라는 기고를 통해, 본 연구를 통해 개발한 타임라인 전염병 추적기술과 관련하여, COVID-19 대응을 통해 우리의 지혜가 고스란히 담겨있는 가축방역 통합시스템을 구축하고, 더 나아가 세계 시장 선도에 대한 중요성을 강조하였다.



한국전자통신연구원  
표준연구부부장

## 가축방역통합시스템 세계 기술 선도

2020년 1사 분기의 최대의 화두는 역시 국민들의 생명까지도 위협하는 COVID-19라고 해도 과언이 아니라고 하겠다. COVID-19와의 전쟁이 종식되지 않아 아직까지도 주변에서 많은 분들이 어려움을 겪고 있는 지금, COVID-19를 글 소재로 사용하는 것이 매우 조심스럽기만 하고 독자 분들께 사정 양해를 구하고자 한다.

얼마 전까지만 대한민국의 방역 대책에 대해 많은 비난의 여론들이 있었다. 물론 아직까지도 안심할 수는 없지만, COVID-19 확진자 수의 해외 증가율과 비교하여 국내 증가율이 비교적 완만하다는 것은 매우 희명적인 일이다. 더구나 우리나라에서 고안한 '드라이브 스루 방식의 검진 방식'에 대해서 CNN이나 BBC에서는 극찬한 바 있으며, 유럽이나 호주에서도 우리의 독자적인 방식을 도입하고 있다. 미국의 한 의원이 '대한민국에서 검사를 받고 싶다'라고 말한 것은 이미 공공연한 사실

이 되고 있다. 필자는 본지를 통해 COVID-19의 위협으로부터 국민들의 생명을 보호하고자 하는 정부의 의지와 의료진의 희생에 큰 감사를 드리고 싶다. 그리고 무엇보다도 이와같은 어려운 시기에도 한마음으로 질병 위협을 극복하고자 하는 대한민국 국민의 하나인 것이 너무나 자랑스럽다.

아직까지도 치료제가 개발되지 않은 COVID-19로부터의 감염 확산을 최소화시키고 있는 비밀은 무엇일까? 필자는 '정보의 공유'와 '사회적 거리두기'에 비밀이 있다고 본다. 지도에 확진자의 동선을 표시하여 확진자와의 접촉을 피하도록 유도하거나, 매일 확진자 동향을 실시간으로 공유하여 국민들이 긴장할 수 있게 한 것은 정보의 공유의 성공적인 사례라고 말하고 싶다. 또한 이러한 정보 공유를 기반으로 국민들 스스로 사회적 거리두기에 적극 동참한 것이 감염 확산을 최소화시키는 것이라고 생각한다.

아직까지도 진행형인 COVID-19와의 전쟁이지만, 필자는 이번 COVID-19와의 전투경험을 우리의 성장 동력을 위한 밑거름으로 사용하기를 제안하고 싶다. 이미 2010년부터 국제역과 조류독감으로 인해 약 3조 8천억 원에 이르는 대규모 가축 살

처분 피해를 경험한 바 있는 우리는, 지금 우리가 힘겹게 싸우고 있는 COVID-19와의 전투 경험을 대규모 가축 전염병을 선제적으로 대응하기 위한 방안으로 응용할 수 있을 것이라고 생각하기 때문이다.

가령, COVID-19 감염에 대한 역학조사 방법을 국제역이나 조류독감 역학조사에 응용하고, 감염 매개체에 대한 실시간 정보 공유를 통하여 축산 종사자들의 '거리두기'가 가능할 것으로 사료된다. 물론 우리나라에서는 이미 국가가축방역통합시스템을 통해서 이러한 대응을 하고 있지만, 이번 COVID-19를 극복하기 위해 우리들이 강구하고 있는 모든 방법들이 기존 시스템에 녹아 들어간다면, 우리나라는 현장 검증된 마진 방역시스템을 보유할 수 있을 것이라고 사료된다. 그것도 대한민국 국민들 개개인들이 일상생활을 마진 방역시스템을 말이다.

필자는 COVID-19와의 전쟁에서 우리가 이길 것이라고 확신한다. 그리고 이러한 고난의 경험을 토대로, 우리의 지혜가 고스란히 담겨있는 가축방역 통합시스템을 구축하고, 더 나아가 세계 시장 선도를 기대한다.

“유통구조 개선, 농가소득 안정화 촉진”이라는 기고를 통해, 코로나 사태의 장기화로 인한 역경을 기회로 삼아 그동안 익숙하지 않았던 온라인 농산물 거래 방법을 재정비하고 농민들의 소득 창출 구조가 안정화될 수 있도록 다양한 노력이 진행되고 있음을 통해, 존의 오프라인 농산물 시장과 공존-상생하여 국민 편익을 증진할 수 있는 최적의 수단이 마련될 수 있어야 함을 강조하였다.



한국전자통신연구원  
기반표준연구실장

## 유통구조 개선, 농가소득 안정화 촉진

코로나 사태의 장기화로 우리의 삶 또한 빠르게 변화되고 있다. 사회적 거리 두기(social distancing)로 인해 필자는 처음으로 집에서 업무용, 컴퓨터 앞에서 원격 교육을 하게 되었다. 조차도 불편할 뿐만 아니라 집중력이 떨어질 수밖에 없는 비대면 도구들을 다루하면서, '역시 비대면(contact)보다는 대면(contact)'이라는 생각이 앞섰으나, 점차 비대면 중심의 삶의 방식에 빠르게 익숙해져 가는 필자의 모습에 새삼스레 놀랄 것이 사실이다.

얼마 전 '사무실에서 양파 10개 구매하기'라는 미션을 수행하는 내용의 방송을 접한 적이 있다. 이 미션은 구매자가 '온라인 농산물 거래소'를 통해 산지로부터 10에 달하는 양파를 비대면으로 구매하는 것이었다. 이미 온라인 쇼핑물에서 한 달에도 몇 번씩 물건을 구매하는 것에 익숙하기에 그리 신선한 주제는 아니었으리라. 하지만 '농산물 온라인 판매'가 7개월 동안 고작 8일밖에 개장하지 못했다 는 안타까운 실정에 국정감사에서 호된 질책을 받

았다는 점을 상기해 보면, 이 길고 긴 코로나 사태를 극복하기 위한 노력이 곳곳에서 시도되고 있음을 확인할 수 있었다. 온라인 농산물 거래는 비용 절감과 새로운 시장 개척 가능성 측면에서 호평을 받고는 있으나, 모니터 화면만을 보고 좋은 상품을 골라내야만 하거나, 비용 지급을 위해 복잡한 절차를 안내해야 하며, 주문한 상품이 아무런 문제가 없이 제때 도착하기를 기다려야만 하는 번거로움에 대한 우려가 있는 것 또한 사실이다. 하지만 이러한 우리는 '익숙함'의 정도에 따르는 것이기 때문에, 필자의 경험에서처럼 시간이 지남수록 점차 누그러지지 않을까 한다. 필자는 온라인 거래가 중간유통 비용을 획기적으로 절감할 수 있을 뿐만 아니라 농산물의 수요와 공급에 대한 정보를 통해 가격의 안정화 효과를 가져 올 수 있다는 점에서 상당히 중요하다고 강조하고 싶다.

물론 온라인을 통해 농산물을 거래하는 시스템이 더 안정적으로 정착되려면 정부의 정책이나 기술적 측면에서 해결해야 할 것들이 산재해 있다. 우선 생산자와 소비자간 상호 신뢰를 바탕으로 실물을 보지 않고도 농산물의 품질을 선별할 수 있는 신뢰 기반 자동 품질 선별 표준이나 기준이 마련되어야 한다. 또한, 농가소득 증대를 위하여 산지의 농민도 농

산물을 쉽게 판매할 수 있도록, 온라인 거래 형태를 B2B뿐만 아니라 C2C까지도 확대하여 지원할 수 있도록 농산물 거래 시스템이 개선되어야 할 것이다. 다행인 것은 정부 차원에서 농산물을 직접 보지 않고도 구매할 수 있도록 품질 등급에 대한 표준 규격 수립, 품질 기준에 영향을 미치는 다양한 정보의 제공 방안 마련을 위한 지속적인 노력을 기울이고 있을 뿐만 아니라 아직은 일부 농산물로 국한되어 있는 대상을 2022년까지 주요 채소류를 포함한 과수·과채류까지 확대할 계획이라는 것이다.

코로나 사태의 장기화로 인해 국민 모두 많은 어려움을 겪고 있다. 하지만 이러한 역경을 기회로 삼아 그동안 익숙하지 않았던 온라인 농산물 거래 방법을 재정비하고 농민들의 소득 창출 구조가 안정화될 수 있도록 다양한 노력이 진행되고 있는 것은 매우 고무적이다. 물론 이러한 비대면 유통 구조가 청과물 시장과 같은 기존의 유통 방식을 모두 대체할 수는 없을 것이다. 하지만 오늘날 온·오프라인 시장이 생산자와 소비자의 편익을 위한 방향으로 계속 진화하는 것처럼 포스트-코로나 시대에는 이러한 온라인 농산물 거래소가 기존의 오프라인 농산물 시장과 공존-상생하여 국민 편익을 증진할 수 있는 최적의 수단으로 자리매김하길 기대한다.

“포스트-코로나 시대, 이젠 기후변화다”라는 기고를 통해, 시군단위가 아니라 농장 단위로 마이크로기상정보 제공의 중요성과 함께, 보다 적극적으로 기상재해 대응을 통해, K-기상재해 모델을 연구에 대한 중요성을 강조하였다.



한국전자통신연구원  
기반표준연구실장

# 포스트-코로나 시대, 이젠 기후변화다

민족 최고의 병철인 추석이 곧 다가온다. 하지만 이번 추석을 앞두고 마음이 무거워지는 것은 무엇일까? 벌써 반년 넘게 계속되는 코로나19로 지칠 대로 지친 몸과 마음도 모르는 척 찾아온 기록적인 폭우와 초대형 태풍으로 수심이 가득 찬 고향 어르신들의 얼굴을 마주할 자신이 없기 때문은 아닐까 생각한다.

얼마 전 기상청과 환경부가 공동으로 '한국 기후 변화 평가보고서 2020'을 발간한 바 있다. 이 보고서에 따르면 최근 한반도의 기온 및 강수 변동성이 전 지구적인 난반화 현상 및 장기적 기후 변동성의 직접적인 영향을 받고 있으며, 1912년부터 100여 년간 여름철 집중호우는 물론 태풍의 빈도와 강도까지 모두 늘어가고 있다고 한다. 올해 여름의 기록적인 폭우와 태풍은 마치 이 보고서의 내용을 증명이라도 하는 듯하다.

하지만 필자의 경우 최근 감염병 위협에만 너무 몰입한 나머지 기후변화와 같은 주요 위협을 등한시한 것은 아닐까 반성해 본다. 그간 코로나 19 신규 확진자의 증가 추세에만 몰입한 나머지 폭우와 태풍의 위협으로 고향 어르신들의 수심 찬 얼굴을 잠시 잊었다고 생각하나 살짝 쓴웃음이 나온다.

하지만 필자를 변명이라도 해 주는 듯 어느 유명 조사기관의 흥미로운 조사 결과가 발표된 적 있다. 미국의 퓨리서치센터에서는 매년 감염병 확산, 기후변화, 테러, 해외 사이버 공격, 핵무기 확산, 세계 경제 상태, 빈곤, 국가나 민족 간 오랜 갈등, 대규모 이주 등 9개 항목에 대해 각국의 국민이 얼마나 큰 위협이라고 생각하는지 추적 조사하고 있다. 올해의 조사 결과에 따르면 유럽 위주의 8개국은 기후변화를, 우리나라를 비롯한 4개국을 감염병 확산을 가장 큰 위협으로 꼽았다. 그만큼 우리나라의 경우 코로나 19와 같은 감염병에 관한 관심이 높다는 것을 수치적인 결과로 보여 주고 있다. 이러한 범국민적 관심으로 인해 다른 나라에 비해 감염자 증가율이 낮아질 수 있으며, 감염병에 대한

적극적인 대처를 통해 우리나라의 방역 모델(또는 역 모델)까지 만들어 세계로 수출하고 표준화까지 선도하고자 하는 움직임이 가능할 수 있다고 생각된다.

우리나라 농촌이 기상재해로부터 자유롭기 위해서는 스마트팜과 같이 외부 기상의 영향을 덜 받을 수 있는 환경을 조성해 폭우와 태풍으로 인한 피해를 줄이는 것도 한 방법일 수 있겠다. 하지만 노지와 같이 기상재해의 위협에 고스란히 노출된 경우 스마트팜과 같이 기상을 조절하지는 못하더라도 기상변화에 따른 피해를 최소화하기 위한 수단 마련이 꼭 필요하다고 생각한다. 다행히 농촌진흥청에서 이미 농업기상재해 조기경보 시스템을 도입해 농장 단위로 기상재해 정보를 서비스하고 있다. 시군단위와 아나라 농장 단위로 마이크로 기상정보를 제공함으로써 보다 적극적으로 기상재해에 대응할 수 있다는 평가이다. 코로나19 이후 표준역 모델과 같은 성공적인 선화를 만들어낸 우리나라가 기상변화 분야에도 표준기상재해 모델을 만들어 세계시장을 선도할 수 있는 그날을 기대해 본다.

“스마트팜 데이터 활성화를 위한 제언”이라는 기고를 통해, 최근 정부 주도로 스마트팜 분야의 다양한 데이터를 수집하기 위한 노력이 끊임없이 이루어지고 있지만 일부 데이터를 수집하는 현장 관계자들은 데이터 수집 대상과 수집 방법 등이 명확하지 않아 많은 어려움을 하소연하고 있다는 문제점을 제기하고, 데이터 수집이 어려운 이유는 다수 있겠지만, 이는 데이터 수집에 대한 목적과 이유, 그리고 수혜에 대한 공감대를 형성하지 못하여 동기부여가 제대로 이루어지지 못한 점이 가장 큰 문제점으로 지적하였다. 마지막으로 스마트팜 데이터의 활용을 촉진하기 위한 방법으로, 첫째, 올해부터 시작하는 스마트팜 다부처 사업은 물론 공공 성격으로 발생하는 데이터를 정제하고 축적할 수 있는 환경을 조성해야 하고, 둘째, 수집된 데이터들은 전문가는 물론 전문 지식이 없는 국민 누구나 쉽게 접근하여 활용할 수 있도록 개방해야 하며, 셋째, 생산뿐만 아니라 유통·가공·소비 등 다양한 분야의 관계인들이 데이터를 활용한 서비스 모델 제안을 독려해야 함을 강조하였다.

## 박주영 칼럼



한국전자통신연구원  
기반표준연구실장

# 스마트팜 데이터 활성화를 위한 제언

“다음 정류소는 000입니다”라는 안내 방송과 동시에 시내버스 하차를 위한 교통카드를 두 손에 미리 쥘 때는 필자의 모습은 과거를 떠올리게 한다. 시내버스 하차카드 제도란 환승 시 교통카드 환인 혜택을 제공한다는 장점이 있다. 하지만 이보다 는 승객의 동행 태도와 같은 객관적인 데이터를 수집·분석하여 효율적인 노선 조정 등 대중교통 정책 결정으로 시민들에게 교통 편의성을 높여줄 수 있다는 더 큰 장점을 갖고 있다. 하지만 이러한 장점을 몰랐던 시절, 필자는 개인 정보가 노출된다는 우려 하나만으로도 하차카드를 열심히 반대하곤 했다.

최근 정부 주도로 스마트팜 분야의 다양한 데이터를 수집하기 위한 노력이 끊임없이 이루어지고

있다. 하지만 일부 데이터를 수집하는 현장 관계자들은 데이터 수집 대상과 수집 방법 등이 명확하지 않아 많은 어려움을 하소연하고 있다. 데이터 수집이 어려운 이유는 다수 있었지만 필자는 데이터 수집에 대한 목적과 이유, 그리고 수혜에 대한 공감대를 형성하지 못하여 동기부여가 제대로 이루어지지 못한 점이 가장 큰 문제점이라고 본다.

데이터 수집에 대한 동기부여를 제공하기 위해서는 데이터 활용의 목적과 성공 사례를 보여주는 것이 매우 중요하다고 생각한다. 왜냐하면 필자가 하차카드를 반대할 당시 누군가가 하차카드를 통해 수집하는 데이터를 통해 효율적인 버스 노선을 제 공할 수 있었더라는 것을 알려주었다면 하차카드의 의무제와는 상관없이 성실한 하차카드 전도사가 되었을 것이다.

물론 데이터 수집의 목적과 활용 성공 사례를 미리 제시한다는 것은 말처럼 쉽지 않다. 하지만 필자는 우리나라 국민 모두가 이루어진 집단 지성을 이용해 스마트팜 데이터를 활용한 성공모델을 도출한다면 그리 어렵지는 않은리라 생각한다. 예를 들어 버스 혼잡도 예측을 통한 버스 노선 신설, 방송 프로그램 분석을 통한 공적 시장 활성화, 신용카드 사용 연령대와 지역 내역 분석을 통해 제공하는 생권 분석 서비스는 필자의 주장을 증명하는 좋은 사례이다.

필자는 스마트팜 데이터의 활용을 촉진하기 위해 다음의 방법들을 제안하고자 한다. 첫째, 올해부터 시작하는 스마트팜 다부처 사업은 물론 공공 성격으로 발생하는 데이터를 정제하고 축적할 수 있는 환경을 조성해야 한다. 둘째, 수집된 데이터들은 전문가는 물론 전문 지식이 없는 국민 누구나 쉽게 접근하여 활용할 수 있도록 개방해야 한다. 셋째, 생산뿐만 아니라 유통·가공·소비 등 다양한 분야의 관계인들이 데이터를 활용한 서비스 모델 제안을 독려해야 한다.

자이부터 스마트팜 데이터 바다를 마음껏 자맥질하여 보물을 찾아내는 것은 독자 여러분의 몫이다.

## 사실

### 농기계산업도 디지털 전환을

2020년 농기계수출이 10억달러를 훌쩍 넘긴 것으로 잠정 집계되고 있어 지난 2018년 이래 3년 연속 10억달러를 초과하는 성과를 거뒀다. 코로나19의 팬데믹 상황에서 세계경제가 마이너스 성장으로 신음하고 있는 가운데 기대 이상의 실적을 보였다는 것은 향후 어떠한 악조건도 능히 극복할 수 있다는 자력을 입증했다. 하지만 2020년 무역기초 회복으로 무역이 살아날 것이란 기대에서도, 아울러 재난지원금 지급, 비대면 소비활성화에 따른 선진국 중심의 빠른 수입수요 회복세는 세계교역 회복의 모멘텀으로 작용할 것이라는 예상을 하고 있다. 다만 바이오텍 정부 출범에서 비롯되는 미국의 대내외 정책변화와 보호무역의 확산, 백신보급 지연에 따른 인적교류 제약 등 불확실성이 존재한다는 점을 우려했다. WTO(세계무역기구) 분석을 토대로 한 미국에 대한 수출은 6억8,894만달러로 전년 6억5,611만달러에 비해 성장세를 보이며 부진에서 벗어난 것으로 봤다. 국가별 적극적인 경기부양책에 따른 글로벌 경제·수요 회복, 디지털무역확산과 더불어 2020년 무역기초 회복으로 무역이 살아날 것이란 기대에서도, 아울러 재난지원금 지급, 비대면 소비활성화에 따른 선진국 중심의 빠른 수입수요 회복세는 세계교역 회복의 모멘텀으로 작용할 것이라는 예상을 하고 있다. 다만 바이오텍 정부 출범에서 비롯되는 미국의 대내외 정책변화와 보호무역의 확산, 백신보급 지연에 따른 인적교류 제약 등 불확실성이 존재한다는 점을 우려했다. WTO(세계무역기구) 분석을 토대로 한 미국에 대한 수출은 6억8,894만달러로 전년 6억5,611만달러에

“스마트팜 글로벌화를 위한 제언”이라는 기고를 통하여, 이미 우리나라에서는 정부 주도로 스마트팜 기

술의 저변 확대를 위한 투자가 이루어짐에 따라, 스마트팜혁신밸리, 국가표준확산사업 등의 실적을 보이고 있지만 해외 우수 IT 기업들의 본격적인 스마트팜 진출 속도를 잡으려면 이것만으로는 아직 부족하다는 문제점을 제시하였다. 최근 인공지능과 빅데이터 부분에서 타의 추종을 불허하는 해외 IT 기업들과 경쟁을 하려면 어떻게 해야 할까라는 질문을 제시한후, 스마트팜 기술 분야에 유지 보수와 유연한 컴퓨팅 자원 활용이 가능한 클라우드를 적극적으로 도입해야 한다고 강조하였다. 빅데이터 기반 인공지능, 자원의 가상화가 가능한 클라우드 기술을 토대로 상호 호환이 가능한 농기자재를 개발하고, 농민에게 저렴하고도 똑똑한 스마트팜 서비스를 제공한다면 해외 경쟁력 확보가 가능할 것이라는 점을 시사하였다.

박주영 칼럼



한국전자동선연구원  
기반표준연구실장

“달나라에도 스마트팜을 지어드릴 수 있습니다.” 얼마 전 스마트팜 도입을 고민하는 농민들을 대상으로 하는 교육에서, 한 강사의 말에 정내는 순식간에 웅성거렸다. 필자 또한 어지러움의 소리에 그제 고개를 가로저을 수밖에 없었다. 하지만 “단, 전문학적인 비용을 내셔야 합니다.”라는 강사의 장난기 섞인 해명을 통해 스마트팜은 결코 비싸서 안 된다, 지금만 비용 이상의 이득을 얻을 수 있을 것이라는 것을 강조하려는 강사의 의도를 읽을 수 있었다.

최근 시장 조사기관에서는 세계 스마트팜 시장 규모가 연평균 약 16.4%의 성장률을 보이고 있으며 이중 스마트팜 생산시스템 시장이 가장 빠르게 성장하고 있다고 발표했다. 또한, 아직 농업 장비 시장

스마트팜 글로벌화를 위한 제언

에서 차지하는 스마트팜의 비중은 현재 미약하지만, 성장 가능성은 매우 큰 상황이라는 점을 시사했다. 이미 수년 전부터 해외 우수 IT 기업들은 본격적으로 스마트팜 시장에 뛰어들고 있다. 구글의 경우 토양, 수분, 작물 건강에 대한 빅데이터를 수집해 종자, 비료, 농약 살포에 도움을 주는 인공지능 의사결정 지원시스템 기술 개발이 완료되었으며, 아마존에서도 마찬가지로 인공지능, 빅데이터 기술을 활용해 국내 스마트팜과의 접목을 하고 있다.

이미 우리나라에서는 정부 주도 스마트팜 기술의 저변 확대를 위한 투자가 이루어지고 있다. 스마트팜혁신밸리, 국가표준확산사업 등은 정부의 부단한 노력을 보여주는 좋은 예이다. 하지만 해외의 우수 IT 기업들의 본격적인 스마트팜 진출 속도를 잡으려면 이것만으로는 아직 부족하다. 이미 인공지능과 빅데이터 부분에서 타의 추종을 불허하는 해외 IT 기업들과 경쟁을 하려면 어떻게 해야 할까?

필자는 앞에서 언급한 강사의 의도처럼 “저렴하면서도 쓸만한 스마트팜” 기술에 답이 있다고 생

각한다. 이를 위해서는 스마트팜 기술 분야에 유지보수와 유연한 컴퓨팅 자원 활용이 가능한 클라우드를 적극적으로 도입해야 한다고 강조하고 싶다. 빅데이터 기반 인공지능, 자원의 가상화가 가능한 클라우드 기술을 토대로 상호 호환이 가능한 농기자재를 개발하고, 농민에게 저렴하고도 똑똑한 스마트팜 서비스를 제공한다면 해외 경쟁력 확보가 가능하리라 사료된다.

이미 우리는 IMF 위기 때 농업 분야의 주권을 상실하는 쓰디쓴 경험을 한 바 있다. 당시 우리나라가 보유한 5대 중자회사 중 4개 업체가 외국 기업에 팔려나감으로써, 종자에 관한 주권을 상당 부분 상실하였다. 다행히 아직 스마트팜 분야에서는 정부의 끊임없는 관심으로 주권을 보유하고 있다. 하지만 여기에 안주하지 않고 우리나라 스마트팜 기술의 원천화를 무기로 하여 해외 우수 IT 기업들의 공격적 진출에 적극적으로 대처함으로써, 두 번 다시는 농업 분야의 주권을 상실하게 되는 일이 없기를 간절히 기원한다.

사설  
발작물기계화 촉진 ‘빛’이 보인다

지난 12일 한국농기계글로벌센터에서 개최된 본지 주최 발작물기계화 촉진 경쟁제도 개선 토론회가 성공적이었다는 평가를 받고 있다. 특히 이날 자유토론 참가자들이 현장중심의 실상을 근거로 세세한 부문까지 문제제기, 지적과 개선방향을 제시하고 유관 기관·단체는 이를 경청하고 적극 개선의지를 보임으로써 발작물기계화 촉진의 새로운 장이 열릴 것으로 기대돼 매우 고무적이라 하지 않을 수 없다.

더욱이 이랜드시자의 총본회의 권수령과 타당성에 대한 재검토를 강력히 요구하고 발작물기계화를 저해하는 독소를 제거하도록 촉구한 본지 3월31일자 ‘불필요한 규제 요소 걷어내야’라는 주제 사설의 연정산장에서 개최된 토론회가 가능

실추시키고 결과적으로 농가가 요구하는 농기계를 이용할 수 없게 하는 만큼 관련규제를 완화해야 한다는 것이다.

또한 수확기-파종기 공회 이별 개발할 경우 최초 1년은 개발과 현장시행, 2년차에는 내구성시험, 3년차엔 종합검정과 제품등록, 4년차부터 본격 판매가 가능하다고 했다. 이같이 장기간의 공정을 거친 끝에 완료한 100만원 내외의 수확기를 판매하기 위해 미세한 차이가 있을 때마다 종합검정을 거쳐야 하는 제도자체가 현실에 맞지 않는다고 지적했다.

보행형 동력확장기에 대한 제도 개선 요구도 나왔다. 이 기준을 20년간 정지된 상태로 모델로 등용하여 보급돼 왔는데 정비정책을 공감하지 않는바 아니지만 모델별로

“기후변화 바퀴 속도를 늦출 수 있는 건 오직 탄소 절감뿐”이라는 기고를 통해, 우리나라는 파리협정의 목표달성과 시급한 기후변화 대응을 위해 ‘2019 유엔 기후 행동 정상회의’에서 기후 목표 상향동맹에 가입한 바 있으며, 이어 발표한 ‘2050 탄소 중립 추진전략’에서는 농축수산의 스마트화 촉진과 청정에너지 사용 확대를 통해 친환경 농축수산업 체계를 구축하겠다는 의지를 표명했음을 상기시킨 후, 축산업에서 발생하는 온실가스는 주로 가축의 장내 발효와 가축분뇨처리 과정에서 발생하는데, 육류 소비가 증가함에 따라 그 총량이 30년 전에 비해 많이 증가하였다는 문제점을 해결하고자 최근 하루 14만 톤씩 발생하는 가축분뇨를 고체연료나 바이오가스로 만드는 신재생에너지 기술 개발 노력이 이루어지고 있지만, 아직도 유럽 선진국에 비하면 턱없이 부족한 실정을 지적하였다. 이러한 문제점의 원인은 악취로 인한 부지의 선정 문제, 수익성 문제 등을 뛰어넘지 못하기 때문이라는 분석을 제시한 후, 온실가스로 인해 생존의 위협을 받는 지금은 개인의 이익보다는 공동 생존을 위해야 할 때라는 점을 강조하였다.

박주영 칼럼



한국전자동선연구원  
기반표준연구실장

“소가 마스크를 찬다고요?” 유난히도 뜨거운 여름, 아직도 수그라들지 않는 코로나바이러스로 인해 결국 소까지 마스크를 써야 하는구나 라는 안타까운 마음이 들었으나, 사실 이는 방역 목적이 아니라 메탄가스 제적을 목적으로 미국의 한 축산분야 신생기업이 만든 장치이다. 소위 ‘소 분취기’ 온난화를 유발한다는 말이 반추동물(대체재결하는 동물)인 소의 위에서 메탄이 발생하는 것이기 때문에 나온 말이지만, 소 분취기에서 나오는 양에 비해 95%의 메탄이 코로 배출되기 때문에 소 마스크를 통해 메탄가스를 채집하려는 시도는 매우 기발한 발상이 아닐 수 없다.

“최근 지구 기후의 변화가 심상치 않다!” 소 한 마리에서 나오는 메탄가스가 얼마나 된다고 그렇게 부산을 떠나고 하겠지만, 얼마 전

기후변화 속도 늦추는 건 오직 탄소 절감뿐

서유럽의 폭우로 인한 많은 인명 피해와 미국과 캐나다에서 기록한 최고 54.5도와 49.6도의 경이로운 온도가 바로 온실가스로 인한 기후변화가 주범이라는 사실을 떠올리면 소 마스크 개발자의 의도에 공감할 수 있을 것이다.

이미 UNDER(유엔 재난위험경감사무국)의 ‘2000~2019년 세계 재해 보고서’에서는 지난 20년 동안 발생한 전 세계 자연재해는 앞선 동일 기간에 비해 1.7배 증가하고 있음이 시사했다. 이 보고서는 “산업화 이전 대비 2019년의 세계 평균기온이 1.1도 더 차솟으면서 폭염과 가뭄, 홍수, 흑한, 태풍, 산불 등 극한 기상 현상들이 더욱 자주 발생하고 있다”라고 밝히며 “재난관리기구들의 노력으로 많은 인명 피해를 막아내긴 했지만 온실가스 배출 감축에 실패함으로써 재해가 증가하고 있다”라고 덧붙였다.

우리나라는 파리협정의 목표달성과 시급한 기후변화 대응을 위해 ‘2019 유엔 기후 행동 정상회의’에서 기후 목표 상향동맹에 가입한 바 있으며, 이어 발표한 ‘2050 탄소 중립 추진전략’에서는 농축수산의 스마트화 촉진과 청정에너지 사용 확대

를 통해 친환경 농축수산업 체계를 구축하겠다는 의지를 표명했다. 축산업에서 발생하는 온실가스는 주로 가축의 장내 발효와 가축분뇨처리 과정에서 발생하는데, 육류 소비가 증가함에 따라 그 총량이 30년 전에 비해 많이 증가하였다. 이러한 문제점을 해결하고자 최근 하루 14만 톤씩 발생하는 가축분뇨를 고체연료나 바이오가스로 만드는 신재생에너지 기술 개발 노력이 이루어지고 있지만, 아직도 유럽 선진국에 비하면 턱없이 부족한 실정이다. 이는 악취로 인한 부지의 선정 문제, 수익성 문제 등을 뛰어넘지 못하기 때문이라는 분석이다. 하지만 온실가스로 인해 생존의 위협을 받는 지금은 개인의 이익보다는 공동 생존을 위해야 할 때라고 생각된다.

필자는 ‘환경은 우리가 후손에게 물려줘야 할 것이 아니라, 후손으로부터 잠시 빌려 쓰는 것이 아닐까’라고 생각한다. 기후변화라는 바퀴를 즉시 멈추지는 못하더라도 속도를 늦출 수 있도록 우리의 힘을 다하여 온실가스 절감에 매진해야 할 때다.

사설  
지자체 농기계 보조지원 확대를

악재만 있고 호재만 터질만큼도 없다. 농기계산업이 그렇다. 물론 모든 것을 뒤엎어버린 코로나 18 여파로 타격을 받지 않은 분야는 존재하지 않는다. 하지만 농기계산업이 극박여력이 취약한 곳은 그리 흔치 않을 것이다. 종합영업체 몇 개 기업을 제외하고 600여업체 모두가 총영업규모 50억 이하의 영세기업인 탓이다. 결론부터 말하자면 정부가 나서서 우리의 생명산업인 농업을 지켜낼 수 있도록 농기계산업에 힘을 실어주지 않으면 안된다.

최근의 ‘발상불가’ 역시 코로나 19 파장은 물론 농기계생산업체의 경영불안정성과 무관치 않다. 농축수산물 물가는 올 2분기에만 11.9%가 뛰어올라 1991년(12.5%) 이후 30년만에 최대상승률을 기록

라 예년의 절반으로 감축한 것도 있고 심한 경우 70%가량을 줄인 지자체도 있는 것이다. 시골의 절대다수가 이 사업에 의존하고 있는 영세농기계기업의 생명줄이 사실상 끊기는 최악의 환경이 조성된 것이다.

이런 가운데 농기계산업 전반에 걸쳐서 정부의 정책이 어떻게든 지원이 필요하다. 지난해부터 주 52시간제 시행되고 내년부터는 최저임금이 올해보다 5.1% 오른 시달 수 있도록 농기계산업에 힘을 실어주지 않으면 안된다.

이와 관련하여 중소기업중앙회 가교본부에서 이를 제기하기도 했다. 중소기업의 지불능력을

바) [기술분석서] 스마트팜 상호연동을 위한 IoT 기술 분석

바-1) [상호연동을 위한 프레임워크 표준기술 연구] OCF 표준기술 분석 및 스마트팜 기자재간 상호연동성을 제공하기 위한 프레임워크 표준기술 연구('19)

(가) Architecture (아키텍처)

아키텍처는 무선 또는 유선의 연결을 설정하고 흐름을 관리하기 위한 솔루션이며 다음을 제공한다.

- 여러 세그먼트를 위한 커뮤니케이션 및 상호 운용성 프레임 워크, OS, 플랫폼, 통신모드, 운송 및 사용 사례
- 환경을 설명하고 정보를 가능하게 하는 공통적이고 일관된 모델
- 발견 및 연결을 위한 공통 통신 프로토콜
- 일반적인 보안 및 식별 메커니즘
- 혁신 및 제품 차별화를 위한 기회
- 스마트 장치에도 적용 가능한 다양한 장치 기능을 다루는 확장 가능한 솔루션

아키텍처에서 물리적 세계(예: 온도 센서, 전등 또는 가전)의 엔티티를 자원으로 나타낸다. 기업과의 상호작용은 REST(Representational State Transfer) 아키텍처 스타일(예: REST)을 준수하는 운영을 이용하여 자원표현을 통해 이루어진다. 아키텍처는 프레임워크의 전체적인 구조를 정보 시스템으로서 그리고 OCF를 구성하는 기업의 상호관계로 정의한다. 기업은 고유한 식별자(URI) 및 지원 인터페이스를 통해 자원에 대한 RESTful 운영을 가능하게 하여 자원으로 노출된다.

모든 RESTful 작업에는 작업 개시자(클라이언트)와 작업 응답자(서버)가 있다. 프레임워크에서는 역할을 통해 클라이언트와 서버의 개념을 실현한다(5.3.5 참조). 모든 장치는 클라이언트 역할을 하고 서버 역할을 하는 장치에 대해 RESTful 작업을 시작할 수 있다. 마찬가지로, 리소스가 서버 역할을 하는 엔티티를 노출하는 모든 장치. REST 아키텍처 스타일에 따라, 각 RESTful 연산에는 상호작용의 맥락을 이해하는 데 필요한 모든 정보가 포함되며, CREATE, RETRIEVE, UPDATE, DELETE 및 NOTIFY(CRUDN)의 작은 집합을 사용하여 구동된다.

다음 그림은 구조를 묘사한다.

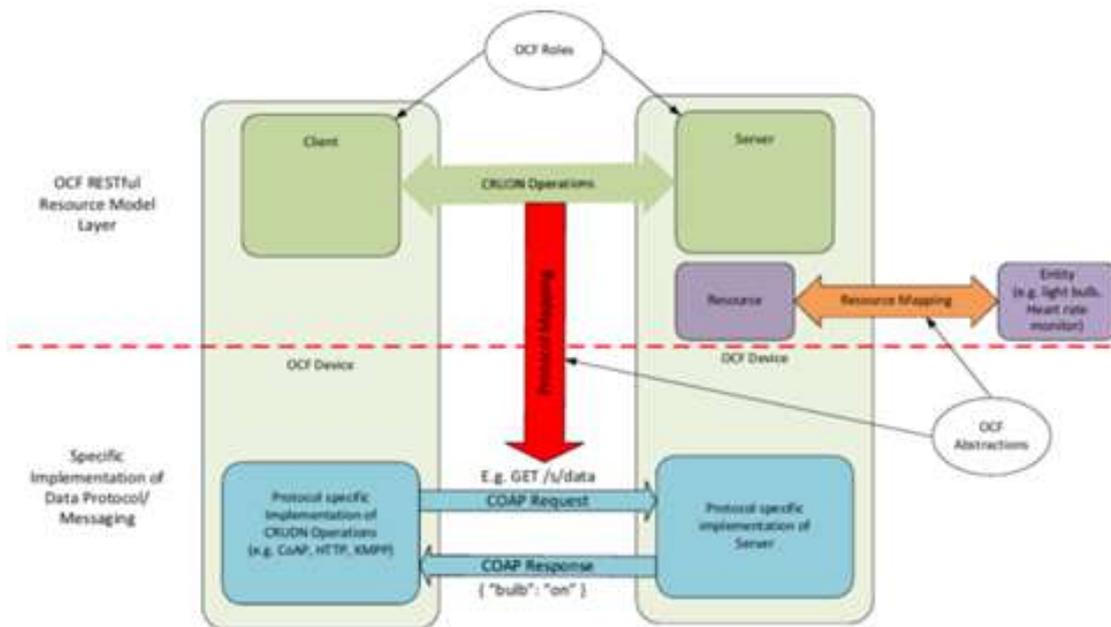


그림 1 - Architecture 개념

아키텍처는 세 가지 주요 측면으로 개념적으로 구성된다.

- Resource model : 필요한 추상화 및 개념을 제공한다. 자원 모델은 논리적으로 모델링하고 애플리케이션과 그 환경에서 논리적으로 운용하는 데 필요한 추상화와 개념을 제공한다. 핵심 자원 모델은 일반적이며 스마트 홈, 산업 또는 자동차와 같은 특정 애플리케이션 영역에 구애받지 않는다. 예를 들어, 자원 모형은 기업을 추상화하는 자원을 정의하고 자원의 표현은 기업의 상태를 매핑한다. 다른 자원 모델 개념을 다른 측면(예: 행동)을 모델링하는 데 사용할 수 있다.
- RESTful 조작 : 일반 CRUDN 조작은 RESTful 패러다임을 사용하여 정의됩니다. 프로토콜과 기술에

관계없이 자원과의 상호 작용을 모델링한다. 그만큼 특정 통신 또는 메시징 프로토콜은 프로토콜 추상화의 일부이며 특정 프로토콜에 대한 자원의 매핑은 5.9.8 에서 제공된다.

- Abstraction : 자원 모델 및 RESTful 조作的 추상화가 맵핑됩니다. 추상화 프리미티브를 사용하여 구체적인 요소에. 엔터티 핸들러는 엔터티를 자원 및 연결 추상화 기본 요소는 논리적 RESTful 조작을 맵핑하는 데 사용됩니다. 데이터 연결 프로토콜 또는 기술. 엔터티 핸들러를 사용하여 매핑 할 수도 있습니다. OCF에서 기본적으로 지원하지 않는 프로토콜을 통해 도달한 엔터티에 대한 자원.

기능 블록 다이어그램은 작동에 필요한 모든 기능을 포함한다. 이러한 기능은 L2 연결, 네트워킹, 전송, 프레임워크 및 애플리케이션 프로파일로 분류된다.

기능 블록은 다음 그림에 나타나 있다.

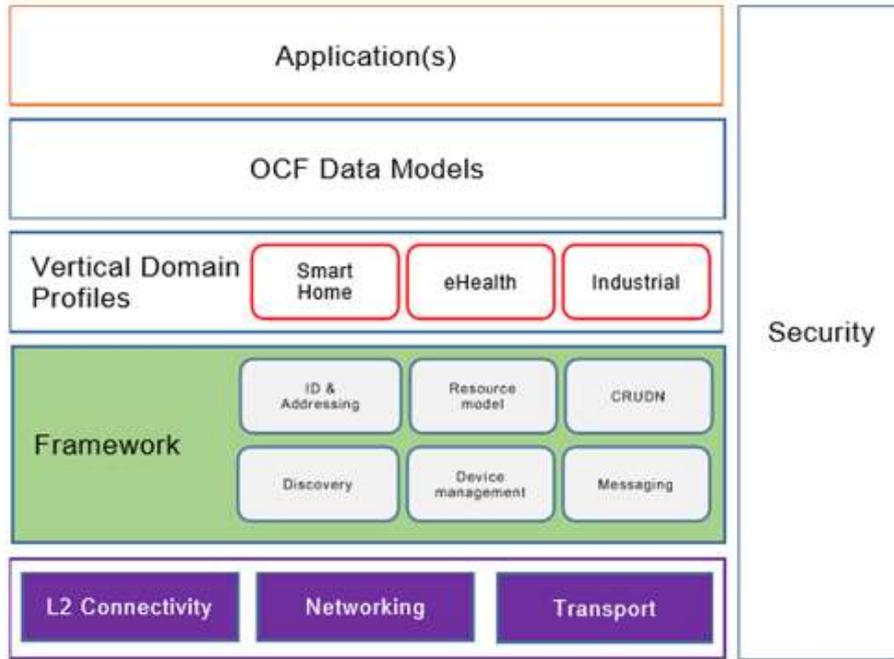


그림 2 - Functional block diagram

- L2 Connectivity: 물리적 및 데이터 링크 계층 연결(예: Wi-Fi TM 또는 Bluetooth ® 연결)을 네트워크에 설정하는 데 필요한 기능 제공
- Networking: 네트워크를 통해 장치 간에 데이터를 교환하는 데 필요한 기능 제공(예: 인터넷)
- Transport: 특정 QoS 제약 조건과 함께 엔드 투 엔드 흐름 전송 제공 전송 프로토콜의 예로는 TCP 및 UDP 또는 IETF에서 개발 중인 새로운 전송 프로토콜(예: DTN)이 있다.
- Framework: 본 문서에 정의된 핵심 기능을 제공한다. 기능 블록은 두 장치 사이의 통신 내용인 요청과 응답의 원천이다.
- Vertical Domain Profiles: 스마트 홈 시장 세그먼트와 같은 특정 기능 제공.

두 장치가 서로 통신할 때 장치의 각 기능 블록은 다음 그림과 같이 피어 장치와 상호 작용한다.

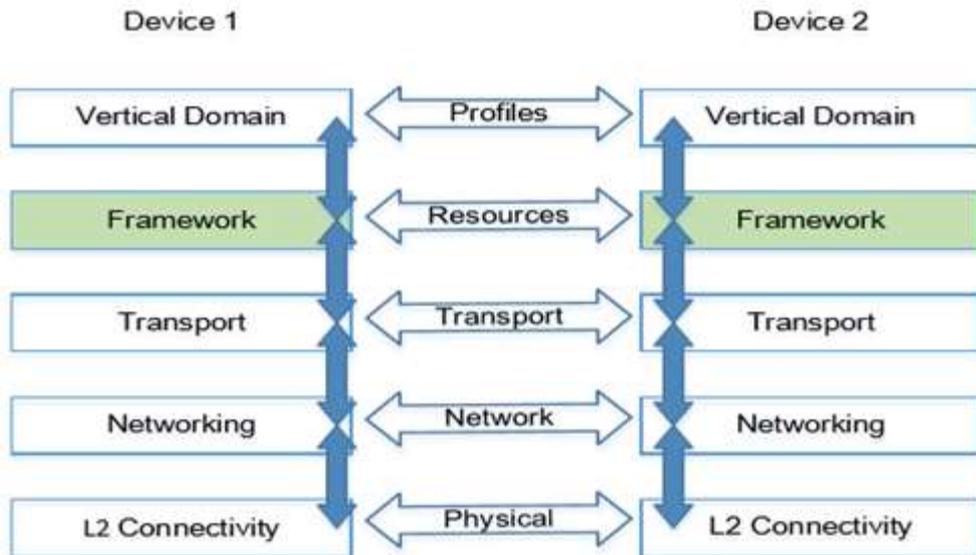


그림 3 - Communication layering model

프레임워크는 작동을 위한 핵심 기능을 제공하는 기능으로 구성된다.

- 식별 및 주소 지정 식별자 및 주소 지정 기능을 정의한다.
- Discovery. 사용 가능한 검색 프로세스를 정의한다.
- 리소스 모델. 자원의 관점에서 실체의 표현 능력을 지정하고 자원을 조작하기 위한 메커니즘을 정의한다.
- CRUDN. 클라이언트와 서버 간의 상호 작용에 대한 일반적인 체계를 제공한다.
- 메시징. RESTful 운영에 대한 특정 메시지 프로토콜(예: CRUDN)을 제공한다. 예를 들어, CoAP는 기본 메시징 프로토콜이다.
- 장치 관리. 장치의 기능 관리 분야를 지정하고 장치 프로비저닝 및 초기 설정뿐만 아니라 장치 모니터링 및 진단도 포함한다.
- 보안. 엔티티에 대한 보안 접근에 필요한 인증, 승인 및 액세스 제어 메커니즘 포함.

상호 작용은 역할로 알려진 논리적 실체 간에 정의된다. 다음 세 가지 역할을 정의한다.

- 클라이언트
- 서버
- 중재자

다음 그림은 스마트폰이 서모스탯으로 요청 메시지를 전송하는 시나리오에서 역할의 예를 보여준다. 원래 요청은 HTTP를 통해 전송되지만, 그 사이에 있는 게이트웨이에 의해 CoAP 요청 메시지로 변환된 후 서모스탯으로 전달된다. 이 예에서, 스마트폰은 클라이언트의 역할을, 게이트웨이는 중재자의 역할을, 서모스탯은 서버의 역할을 맡는다

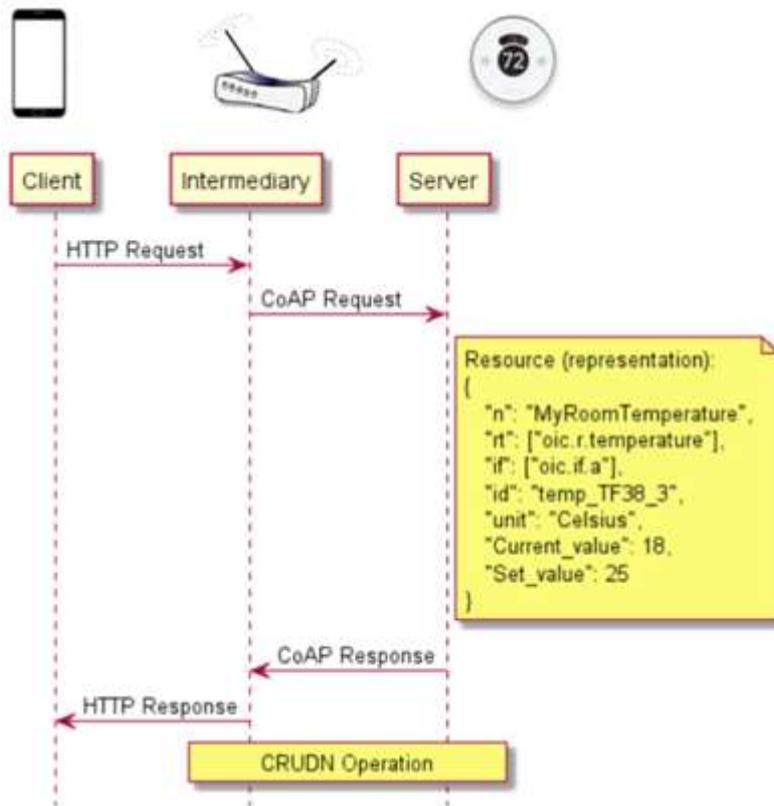


그림 4 - Example illustrating the roles

Example Scenario: Bridging to Non-OCF ecosystem 시나리오의 사용 사례는 OCF가 지원하지 않는 프로토콜을 구현하는 심박수 센서를 모니터링하는 데 사용되는 디스플레이(손목 시계와 같은)이다. 다음 그림은 개념에 대한 상세한 논리적 뷰를 제공한다

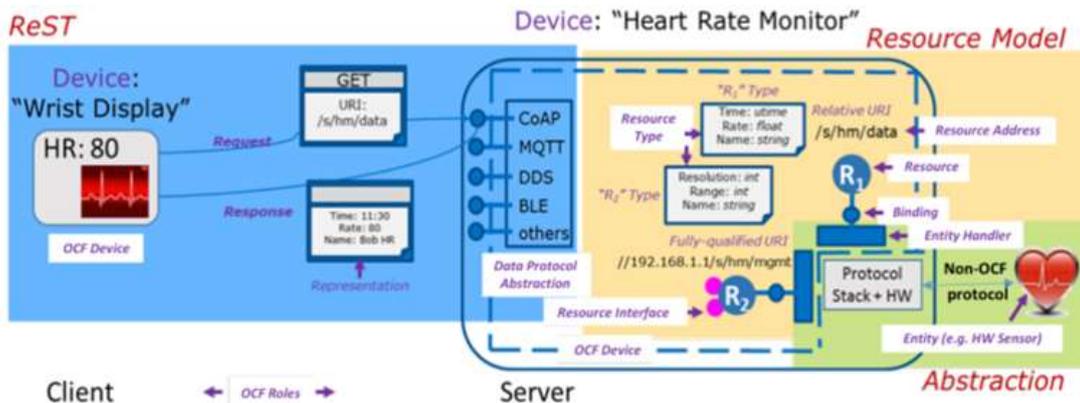


그림 5 - Framework - Architecture Detail

기업 핸들러가 있는 서버를 사용하여 비OCF 장치에 직접 접속함으로써 여러 가지 방법으로 구현될 수 있다.

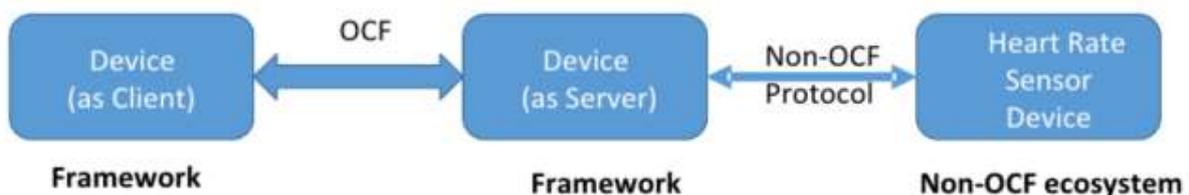


그림 6 - Server bridging to Non-OCF device

시동 시 서버는 OCF가 아닌 시스템(예: Heart Rate Sensor Device)을 검색하고 발견된 각 장치 또는 기능에 대한 리소스를 생성하는 엔티티 핸들러를 실행한다. 기업 처리자는 발견된 각 장치 또는 기능에 대

한 자원을 생성하고 그 자원에 자신을 바인딩한다. 이러한 자원은 서버에서 검색할 수 있다.

리소스가 생성되고 검색이 가능해지면 디스플레이 장치는 이러한 리소스를 검색하고 이 문서에 설명된 메커니즘을 사용하여 해당 리소스를 작동할 수 있다. 그런 다음, 엔티티 처리기에 의해 서버의 리소스에 대한 요청이 해석되고 비OCF 장치에서 지원하는 프로토콜을 사용하여 비 OCF 장치로 전달된다. 그런 다음 비 OCF 장치에서 반환된 정보가 해당 리소스에 대한 적절한 응답에 매핑된다

(나) OCF Cloud architecture (OCF 클라우드 아키텍처)

다음 두 개의 그림은 OCF 클라우드 아키텍처를 나타낸다.

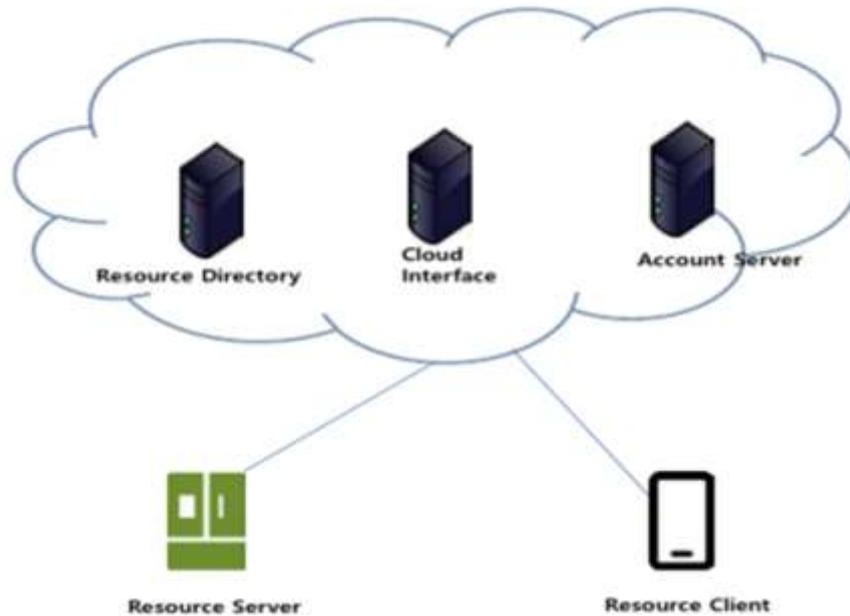


그림 7 - OCF Cloud deployment architecture

클라우드 아키텍처는 다음 세 개의 네트워크 엔티티로 구성된다.

- Cloud Interface Server - OCF 장치가 주로 사용되는 논리적 엔티티 그것은 계정 서버 및 리소스 디렉터리 기능을 캡슐화한다. 클라우드 인터페이스는 패킷 헤더에서 요청 URI에 기반하여 OCF 장치 간에 패킷을 라우팅한다. 클라이언트는 서버에 대한 영구 연결을 유지해야 한다.
- Account Server - 장치 등록, Auth Token 유효성 검사를 처리하고 장치의 로그인 및 토큰 새로 고침 요청을 처리하는 논리적 엔티티이다.
- Resource Directory - 서버에서 게시한 리소스 정보를 보유하고 있는 논리적 엔티티. 리소스를 찾을 때 클라이언트는 서버를 대신하여 리소스 디렉터리로부터 응답을 수신한다. 그런 다음, Resource Directory에 포함된 정보를 사용하여 클라이언트는 서버에 직접 연결된다.

클라이언트가 서버에 액세스하려고 할 때 클라이언트는 클라우드 인터페이스 서버에 연결한 다음, 클라우드 인터페이스는 수신된 메시지를 권한을 확인한 후 표시된 서버로 라우팅한다.

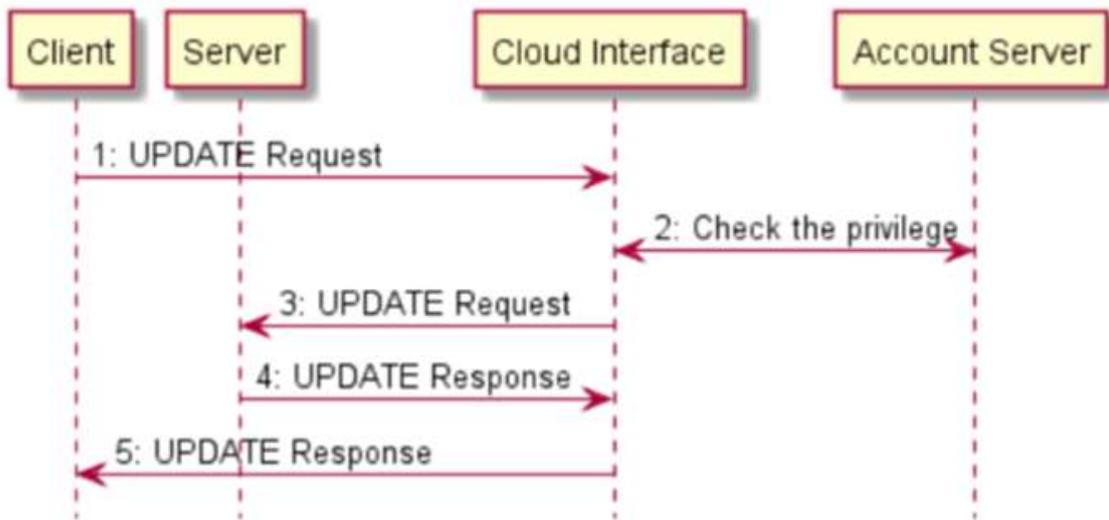


그림 8 - OCF Endpoint routing

(다) Identification and addressing (식별 및 주소지정)

프레임워크에서 요소들간의 적절하고 효율적인 상호작용을 촉진하려면 이러한 요소들을 식별하고, 이름을 지정하고, 다루는 수단이 필요하다. 주소는 리소스 주소를 지정하는 URL과 연결 계층에서 주소를 지정하는 IP 주소일 수 있다.

식별자는 사용의 맥락이나 영역 내에서 명확하지 않다. 필요한 속성을 가진 식별자를 생성하는 데 사용될 수 있는 많은 체계가 있다. 식별자는 식별자가 해당 컨텍스트 또는 도메인 내에서만 모호하지 않을 것으로 예상되며 보장된다는 점에서 상황에 따라 다를 수 있다. 식별자는 또한 이러한 식별자가 공간 및 시간적으로 모든 맥락과 도메인에 걸쳐 모호하지 않게 보장되는 경우 문맥에 독립적일 수 있다.

상황별 식별자는 단일 열거와 같은 단순한 체계에 의해 정의될 수 있거나 주소나 이름을 오버로드하여 정의할 수 있다. 예를 들어 IP 주소는 스마트 홈의 게이트웨이 뒤에 있는 개인 도메인 내에서 식별자가 될 수 있다. 한편, 문맥 독립적인 식별자는 보편적으로 고유한 정체성을 도출하는 더 강력한 체계를 필요로 한다. 예를 들어 UUID(Universally Unique Identifier) 버전 중 하나라도 그렇다. 또한 계층의 루트가 UUID로 식별되는 도메인의 계층을 사용하여 컨텍스트 독립 식별자를 생성할 수 있으며, 하위 도메인은 해당 도메인에 대한 컨텍스트 고유 식별자를 부모의 컨텍스트 독립적인 식별자와 연결하여 컨텍스트 독립 식별자를 생성할 수 있다

URI가 URL인 경우, 자원을 URI를 사용하여 식별하고 동일한 URI로 처리할 수 있다. 경우에 따라 자원은 URI와 다른 식별자를 필요로 할 수 있다. 이 경우 자원은 식별자 값인 속성을 가질 수 있다. URI가 URL 형식일 때, URI를 사용하여 리소스를 처리할 수 있다.

OCF URI는 다음 형식으로 지정된다.

---

ocf://<authority>/<path>?<query>

---

다음은 각 구성 요소가 취하는 값에 대한 설명이다.

URI의 계획은 "ocf"이다. "ocf" 체계는 이 문서에 정의된 의미론, 정의 및 용도를 나타낸다. URI가 "/"(이중 슬래시) 앞의 부분을 생략한 경우, "ocf" 체계를 가정해야 한다.

각 전송 바인딩은 요청자가 네트워크를 통해 전송하기 전에 OCF URI를 전송 프로토콜 URI로 변환하는 방법을 명시하는 책임을 진다. 마찬가지로, 수신기 측에서도 각 전송 바인딩은 수신기의 자원 모델 계층에 인계하기 전에 전송 프로토콜 URI에서 OCF URI를 변환하는 방법을 명시하는 책임이 있다.

OCF URI의 권한은 [OCF 보안]에 정의된 대로 서버의 장치 ID("di") 값이어야 한다.

경로는 서버의 컨텍스트 내에서 리소스를 명확하게 식별하거나 참조하는 문자열이다. 이 버전의 문서에서 경로에는 pct 인코딩되지 않은 ASC가 포함되지 않아야 한다. II 문자 또는 NUL 문자. 경로 앞에 "/"(슬래시)가 있어야 한다. 경로에는 인적 가독성 때문에 "/"(슬래시) 분리된 세그먼트가 있을 수 있다. OCF 컨텍스트에서 "/"(슬래시) 분리 세그먼트는 리소스를 직접 참조하는 단일 문자열(즉, 플랫폼 구조)로 처리되며 계층으로 구문 분석되지 않는다. 서버에서 결과 참조가 호스트의 컨텍스트 내에서 고유할 경우 해싱 또는 다른 방식을 사용하여 경로 또는 경로의 일부 하위 문자열을 단축할 수 있다.

경로가 생성되면 URI의 리소스 또는 수신인에 액세스하는 클라이언트는 이 경로를 불투명한 문자열로

사용해야 하며 구조, 조직 또는 의미론을 추론하기 위해 구문 분석해서는 안 된다.

쿼리 문자열은 각각 "&"로 구분된 "이름값" 세그먼트(이름 값 쌍) 목록을 포함해야 한다. 쿼리 문자열은 메시징에 사용되는 프로토콜의 적절한 구문에 매핑된다. (예: CoAP).

URI는 완전한 자격을 갖췄거나 URI의 상대적 생성일 수 있다.

URI는 해당 리소스의 생성자인 클라이언트에 의해 정의될 수 있다. 그러한 URI는 상대적이거나 절대적일 수 있다. 상대 URI는 호스트되는 장치에 상대적이어야 한다. 또는 OCF 인터페이스에 기반한 사전 정의된 규칙이나 리소스 구성에 기반하여 해당 리소스의 서버에 의해 자동으로 URI가 생성되거나 다른 루트 또는 베이스에 대해 생성될 수 있다.

URI의 절대 경로 참조는 불투명한 문자열로 처리되며, 클라이언트는 URI의 명시적 또는 묵시적 구조를 유추해서는 안 된다. URI는 단순히 주소일 뿐이다. 또한 리소스를 호스팅하는 장치는 각 리소스의 URI를 해당 리소스만 다루는 불투명 문자열로 취급할 것을 권장한다. (예: URI의 "/a"와 "/a/b"는 별개의 주소로 간주되며, 리소스 b는 리소스 a의 자식으로서 해석될 수 없다.)

네임 스페이스에 대한 규칙으로는 상대 URI 접두사 "/oic/"는 OCF 규격에 정의된 URI의 네임스페이스로 예약되어 있으며 OCF 규격에 정의되지 않은 URI에는 사용할 수 없다

네트워크 주소로는 IP 주소를 사용한다.

- 기기가 IP 구성 인터페이스를 사용할 때 IP 주소를 사용한다.
- 장치가 피어의 ID 정보만 가지고 있는 경우, 식별자를 해당 주소에 매핑하는 분해능 메커니즘이 필요하다.

#### (라) Resource model (리소스 모델)

자원 모델은 OCF 생태계의 장치들 사이에 일관성과 핵심 상호운용성을 제공하는 개념과 메커니즘을 정의한다. 그런 다음 자원 모델 개념과 메커니즘을 전송 프로토콜에 매핑하여 장치 간의 통신을 가능하게 한다. 각 전송은 통신 프로토콜 상호운용성을 제공한다. 그러므로 자원 모델은 교통수단과 독립적으로 상호운용성을 정의할 수 있다.

또한, 자원 모델의 개념은 1차 아티팩트의 모델링과 그 관계와 상호운용성에 필요한 의미 정보를 맥락에서 포착할 수 있다. 이렇게 해서 OCF는 단순한 프로토콜 상호운용성을 넘어 웨어러블과 사물인터넷 생태계의 진정한 상호운용성에 필요한 풍부한 의미론을 포착한다.

리소스 모델의 주요 개념은 엔티티, 리소스, 통일 리소스 식별자(URI), 리소스 유형, 속성, 표현, OCF 인터페이스, 수집 및 링크 등이다. 또한, 일반적인 메커니즘은 CREATE, RETRIEVE, UPDATE, DELETE 및 NOTIFY(CRUDN)이다.

이러한 개념과 메커니즘은 프레임워크가 적용되는 다양한 사용 사례에 필요한 풍부한 의미론과 상호운용성을 정의하는 다양한 방법으로 구성될 수 있다. OCF 자원 모형 프레임워크에서 기업은 눈에 보이거나, 상호작용하거나, 조작할 필요가 있으며, 그것은 자원이라 불리는 추상화로 표현된다. 자원은 기업의 상태를 요약하고 나타낸다. 리소스는 URI를 사용하여 식별, 주소 지정 및 이름 지정된다.

속성은 "key=value" 쌍이며 리소스 상태를 나타낸다. 이러한 속성의 스냅샷은 리소스의 표현이다. 표현에 대한 특정 뷰와 해당 뷰에서 적용되는 메커니즘은 OCF 인터페이스로 지정된다. 리소스와의 상호 작용은 표현을 포함하는 요청 및 응답으로 수행된다.

리소스 인스턴스는 리소스 유형에서 파생된다. 한 자원과 다른 자원의 단방향 관계는 연결로 정의된다. 속성 및 링크가 있는 리소스는 모음이다. 속성 집합을 사용하여 리소스 상태를 정의할 수 있다. 이 상태는 해당 리소스로부터의 응답과 해당 리소스에 대한 요청에서 각각 적절한 표현을 사용하여 검색하거나 업데이트할 수 있다.

리소스(및 리소스 유형)는 기능을 나타내며 기능을 노출하는 데 사용될 수 있다. 그 자원과의 상호 작용은 그 능력을 행사하거나 사용하기 위해 사용될 수 있다. 그러한 능력은 발견, 관리, 광고 등과 같은 프로세스를 정의하는 데 사용될 수 있다. 예를 들어, 장치에서 리소스의 발견은 속성 또는 속성이 장치의 리소스를 설명하거나 참조하는 값을 가지는 특정 리소스의 표현을 검색하는 것으로 정의할 수 있다.

표현과 함께 요청 또는 응답에 대한 정보는 전송 프로토콜을 사용하여 직렬화하거나 전송 프로토콜의 페이로드에 캡슐화하여 와이어에 전달될 수 있다.

- 특정 방법은 요청 또는 응답의 표준적 매핑에 의해 결정된다.

#### (마) 자원

리소스는 하나 이상의 리소스 유형으로 정의되어야 한다. 리소스를 생성하기 위한 요청은 해당 리소스를

정의하는 하나 이상의 리소스 유형을 지정해야 한다.

리소스는 장치에서 호스팅된다. 자원은 URI를 가져야 한다. URI는 자원 생성 시 당국이 할당하거나 자원 유형의 규격으로 미리 정의할 수 있다. 다음 그림에 자원 표현이 예시되어 있다.

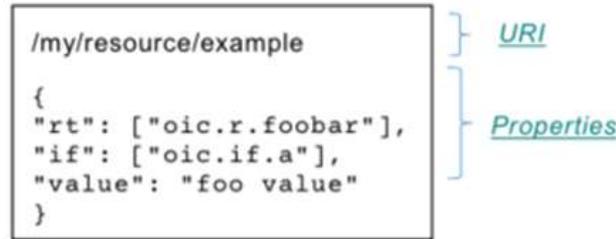


그림 9 - Example resource

핵심 자원은 조항 10(예: 검색, 장치 관리 등)에 정의된 기능적 상호 작용을 활성화하기 위해 이 문서에 정의된 리소스다. 핵심 자원 중, "/oic/res", "/oic/p", "/oic/d"는 모든 기기에서 지원되어야 한다. 기기는 지원하는 기능 상호작용에 따라 다른 핵심 자원을 지원할 수 있다.

속성은 해당 자원과 관련된 메타 정보를 포함하여 자원을 통해 노출되는 측면을 설명한다.

속성은 예를 들어 이름을 가져야 한다. 속성 이름 및 값. 속성 값. 속성은 키-값 쌍으로 표현되며, 여기서 키는 속성 이름이고 속성 값은 <속성 이름> = <속성 값>과 같은 값이다. 예를 들어, "온도" 속성의 속성 이름이 "temp"이고 속성 값이 "30F"인 경우, 속성은 "temp=30F"로 표시된다. 속성의 특정 형식은 인코딩 체계에 따라 달라진다. 예를 들어 JSON에서 속성은 "키": 값(예: "temp": 30)으로 표시된다.

또한 속성 정의는 다음을 포함해야 한다.

- Value Type - Value Type은 속성 값이 취할 수 있는 값을 정의한다. 값 유형은 단순한 데이터 유형(예: 문자열, 부울)이거나 스키마로 정의된 복잡한 데이터 유형일 수 있다. 값 유형이 정의될 수 있다.
- Value Rules는 Property Value가 취할 수 있는 값 집합에 대한 규칙을 정의한다. 그러한 규칙은 값의 범위, 최소-최대, 공식, 열거된 값의 집합, 패턴, 조건부 값 및 다른 속성의 값에 대한 의존성을 정의할 수 있다. 규칙은 속성 값 및 플래그 오류의 특정 값을 검증하는 데 사용될 수 있다.
- Mandatory - 지정된 리소스 유형에 대해 속성이 필수인지 여부를 지정한다.
- Access modes - 속성을 읽을 수 있는지, 쓸 수 있는지 또는 둘 다 사용할 수 있는지 지정한다. "r"은 읽기에 사용되며 "w"는 쓰기에 사용되며, 둘 다 지정될 수 있다. 쓰기는 자동적으로 읽히는 것을 의미하지는 않는다.
- Property Title - 속성을 지정하기 위한 인간 친화적인 이름이며, 일반적으로 전선을 통해 전송되지 않는다.
- Description - 이 속성의 목적 및 예상 사용을 정의하는 설명 텍스트.

일반적으로 속성은 관련된 리소스 내에서만 의미가 있다. 그러나 공통 속성이라고 알려진 모든 리소스에 의해 지원될 수 있는 기본 속성 집합은 리소스 전체에서 의미론을 그대로 유지한다. 즉, 이들의 "키=값" 쌍은 모든 리소스에서 동일한 것을 의미한다. 모든 공통 속성에 대한 상세 표는 공통 속성에 정의되어 있다.

#### (바) 공통 속성

이 절에 정의된 공통 속성은 모든 리소스에 대해 지정할 수 있다. 리소스 유형, 리소스 인터페이스, 이름 및 리소스 ID 등의 속성은 공통 속성으로 정의된다.

공통 속성의 이름은 고유해야 하며 다른 속성에 의해 사용되어서는 안 된다. 새 리소스 유형을 정의할 때 비공통 속성은 기존 공통 속성(예: "rt", "if", "n", "id")의 이름을 사용해서는 안 된다. 새로운 "공통 속성"을 정의할 때, 그 이름이 다른 속성에서 사용되지 않았는지 확인해야 한다. 기존 OCF 정의 리소스 유형의 모든 속성을 확인하여 새 공통 속성 이름의 고유성을 확인할 수 있다. 그러나 리소스 유형의 수가 증가함에 따라 이러한 작업이 번거로워질 수 있다. 향후 이러한 이름 충돌을 방지하기 위해 OCF는 공통 속성에 대한 특정 이름 공간을 예약할 수 있다.

잠재적 접근방식은,

- 특정 접두사(예: "oic")를 지정할 수 있으며 접두사 앞에 있는 이름(예: "oic.psize")은 공통 속성 전용 이름,

- 한두 개의 문자로 구성된 이름은 공통 속성에 대해 예약되어 있으며, 다른 모든 속성은 2자 이상의 길이를 가진 이름을 가져야 한다.
- 공통의 특성은 그 자신을 구별하기 위해 특정 대상에 내포할 수 있다.  
공용 속성 업데이트(액세스 모드로 쓰기를 지원하는) 기능은 "oic.if.rw"(읽기-쓰기) OCF 인터페이스로 제한되므로, 속성이 속성 정의 및 읽기-쓰기 관련 스키마로 정의된 쓰기 액세스를 지원하는 경우에만 읽기-쓰기 OCF 인터페이스를 사용하여 공용 속성을 업데이트할 수 있어야 한다.

모든 자원에 대한 다음과 같은 공통 속성은 다음과 같이 요약된다.

- 리소스 유형("rt") - 이 속성은 해당 리소스의 리소스 유형을 선언하는 데 사용된다. 리소스를 둘 이상의 리소스 유형으로 정의할 수 있으므로 리소스 유형 속성의 속성 값을 사용하여 둘 이상의 리소스 유형을 선언할 수 있다. 예: "rt": ["oic.wk.d", "oic.d.airconditioner"]는 이 속성을 포함하는 리소스가 "oic.wk.d" 리소스 유형 또는 "oic.d.airconditioner" 리소스 유형에 의해 정의된다고 선언한다.
- OCF 인터페이스("if") - 이 속성은 자원이 지원하는 OCF 인터페이스를 선언한다. OCF 인터페이스 속성의 속성 값은 다중 값이 될 수 있으며 지원되는 모든 OCF 인터페이스를 나열한다.
- 이름("n") - 속성은 리소스에 할당된 사람이 읽을 수 있는 이름을 선언한다.
- 리소스 ID("id"): 해당 속성 값은 리소스의 특정 인스턴스에 대한 고유한(호스트 서버의 범위에 걸쳐) 인스턴스 식별자가 되어야 한다. 이 식별자의 인코딩은 장치 및 구현에 따라 달라진다.

#### (사) 자원 유형

리소스 유형은 리소스의 클래스 또는 범주로, 리소스는 하나 이상의 리소스 유형의 인스턴스이다. 리소스의 리소스 유형은 리소스 유형 공통 속성을 사용하거나 리소스 유형 매개 변수를 사용하는 링크에서 선언된다.

리소스 유형은 OCF에 의해 사전 정의되거나 제조업체, 최종 사용자 또는 장치 개발자(벤더 정의 리소스 유형)에 의해 사용자 정의 정의될 수 있다. 리소스 유형 및 이의 정의 세부사항은 대역 외(예: 문서화)으로 전달되거나 API 또는 애플리케이션에 의해 다운로드 및 사용될 수 있는 메타 언어를 사용하여 명시적으로 정의될 수 있다. OCF는 OCF의 RESTful 인터페이스 및 리소스 정의 규격 방법으로 OpenAPI 2.0을 채택했다.

모든 리소스 유형은 IETF RFC 6690(ABNF의 경우 제2조항, 요건의 경우 제3.1조항)의 리소스 유형 특성을 관리하는 ABNF와 ""로 구분되는 요구사항을 사용하여 식별해야 한다.

전체 문자열은 리소스 유형 ID를 나타낸다. ID를 정의할 때 각 세그먼트는 리소스 유형에 적합한 의미를 나타낼 수 있다. 예를 들어, 각 세그먼트는 네임스페이스를 나타낼 수 있다. ID를 정의한 후에는 ID를 불투명하게 사용해야 하며 구현은 개별 세그먼트에서 어떤 정보도 유추해서는 안 된다. 리소스 유형 ID 정의에서 첫 번째 세그먼트로 사용되는 문자열 "oic"은 OCF 정의 리소스 유형에 대해 예약된다. 모든 OCF 정의 리소스 유형은 IETF RFC 6690에 설명된 대로 IANA 핵심 매개변수 레지스트리에 등록되어야 한다.

인스턴스화되거나 생성된 리소스는 해당 리소스의 템플릿인 하나 이상의 리소스 유형을 가져야 한다. 리소스가 준수하는 리소스 유형은 다음 표에 정의된 리소스에 대한 "rt" 공통 속성을 사용하여 선언해야 한다. "rt" 공용 속성의 속성 값은 템플릿으로 사용되는 리소스 유형에 대한 리소스 유형 ID 목록(즉, "rt" 리소스 유형 ID 목록)이어야 한다.

표 1 - Resource Type Common Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Type	"rt"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	Yes	The Property name rt is as described in IETF RFC 6690

리소스 유형은 리소스의 사용자(예: 클라이언트)와 호스트(예: 서버) 간에 명시적으로 검색되거나 암묵적으로 공유될 수 있다.

리소스 유형은 다음과 같이 지정된다.

- 미리 정의된 URI(옵션) - OCF 규격의 특정 리소스 유형에 대해 미리 정의된 URI를 지정할 수 있다. 리소스 유형에 미리 정의된 URI가 있는 경우 해당 리소스 유형의 모든 인스턴스는 미리 정의된 URI

만 사용해야 한다. 다른 리소스 유형의 인스턴스는 미리 정의된 URI를 사용해서는 안 된다.

- 리소스 유형 제목(선택 사항) - 리소스 유형을 지정할 수 있는 친숙한 이름.
- 리소스 유형 ID - 리소스 유형을 식별하는 "rt" 속성의 값(예: "oic.wk.p").
- 리소스 인터페이스 - 리소스 유형에서 지원할 수 있는 OCF 인터페이스 목록
- 속성 - 리소스 유형에 적용되는 모든 속성의 정의. 리소스 유형 정의는 속성이 필수인지, 조건부 필수인지, 선택 사항인지를 정의해야 한다.
- 관련 리소스 유형(선택 사항) - 리소스 유형의 일부로 참조될 수 있는 다른 리소스 유형의 사양(집합에 적용 가능)
- MIME 유형(옵션) - 직렬화를 포함한 리소스에서 지원하는 MIME 유형(예: application/cbor, application/json, application/xml)

다음 두 개의 표는 예시적인 foobar 자원 유형과 관련 속성에 대한 예시를 제공한다.

표 2 - Example foobar Resource Type

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction	M/CR/O
none	"foobar"	"oic.r.foobar"	"oic.if.a"	Example "foobar" Resource	Actuation	O

표 3 - Example foobar Property

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Type	"rt"	"array"	N/A	N/A	R	Yes	Resource Type
OCF Interface	"if"	"array"	N/A	N/A	R	Yes	OCF Interface
Foo value	value	"string"	N/A	N/A	R	Yes	Foo value

For example, an instance of the foobar Resource Type.

```
{
  "rt": ["oic.r.foobar"],
  "if": ["oic.if.a"],
  "value": "foo value"
}
```

For example, a schema representation for the foobar Resource Type.

```
{
  "$schema": "http://json-schema.org/draft-04/schema",
  "type": "object",
  "properties": {
    "rt": {
      "type": "array",
      "items": {
        "type": "string",
        "maxLength": 64
      },
      "minItems": 1,
      "readOnly": true,
      "description": "Resource Type of the Resource"
    },
    "if": {
      "type": "array",
      "items": {
        "type": "string",
        "enum": ["oic.if.baseline", "oic.if.ll", "oic.if.b", "oic.if.lb", "oic.if.rw", "oic.if.r", "oic.if.a", "oic.if.s"]
      },
      "value": {"type": "string"}
    },
    "required": ["rt", "if", "value"]
  }
}
```

다중 값 "rt" 리소스란 포함된 리소스 유형 중 잘 알려진 리소스 유형(예: "oic.wk")을 나타내지 않는 여러 리소스 유형을 가진 리소스를 말한다. 이러한 리소스는 여러 리소스 유형과 연관되므로 여러 리소스 유형 ID의 "rt" 속성 값이 있다(예: "rt": ["oic.r.switch.binary", "oic.r.light.brightness"]). rt 속성 값의 리소

스 유형 ID 순서는 무의미하다. 예를 들어, "rt":["oic.r.switch.binary", "oic.r.light.brightness" 및 "rt":["oic.r.light.brightness", "oic.r.switch.binary"는 같은 의미를 갖는다.

다중 값 "rt" 리소스에 대한 리소스 유형은 다음 조건을 충족해야 한다.

속성 이름은 공통 속성을 제외하고 각 리소스 유형에 대한 속성 이름은 고유해야 한다(다중 값 "rt" 리소스의 범위 내에서). 그렇지 않으면 상충되는 속성 의미론이 있을 것이다. 두 리소스 유형이 동일한 속성 "이름"을 가진 속성을 가진 경우 다중 값 "rt" 리소스는 이러한 리소스 유형으로 구성되지 않아야 한다.

다중 값 "rt" 리소스가 각 리소스 유형에 대한 모든 요구 사항을 충족하고 오픈을 준수각 구성 요소 리소스 유형에 대한 API 2.0 정의. 따라서 다중 값 "rt" 리소스의 필수 속성은 각 리소스 유형의 모든 필수 속성의 조합이어야 한다. 예를 들어, "rt"가 ["oic.r.switch.binary", "oic.r.light.brightness"인 리소스의 필수 속성은 "value"와 "brightness"이며, 여기서 전자는 "oic.r.switch.bridness"의 경우 필수 속성이다.

다중 값 "rt" 리소스 인터페이스 세트는 구성요소 리소스 유형에서 OCF 인터페이스 세트의 조합이어야 한다. OCF 인터페이스에서 CRUDN 동작에 대응한 자원 표현은 해당 OCF 인터페이스에 대해 정의된 스키마의 조합이어야 한다. 다중 값 "rt" 리소스에 대한 기본 OCF 인터페이스는 리소스 유형 간에 보장된 유일한 공통 OCF 인터페이스인 기본 OCF 인터페이스("oic.if.baseline")가 되어야 한다.

각 리소스 유형이 동일한 OCF 인터페이스 집합을 지원하는 경우 명확한 구별을 위해, 결과 다중 값 "rt" 리소스는 기본 OCF 인터페이스("oic.if.baseline")를 가진 동일한 OCF 인터페이스 집합을 가지고 있다.

장치 유형은 장치의 클래스이다. 정의된 각 장치 유형에는 장치가 해당 장치 유형에 대해 구현해야 하는 최소 리소스 유형 목록이 포함된다. 장치는 최소 목록을 넘어 추가 표준 및 공급업체 정의 리소스 유형을 노출할 수 있다. 장치 유형은 리소스 검색에 사용된다.

리소스 유형과 마찬가지로 장치 유형은 리소스 유형 공통 속성 또는 리소스 유형 매개 변수를 사용하는 링크에서 사용할 수 있다. 장치 유형은 사전 정의(ISO/IEC 30118-5:2018) 또는 제조업체, 최종 사용자 또는 장치 개발자(벤더 정의 장치 유형)에 의한 사용자 정의 정의로 정의될 수 있다. 기기 유형 및 이의 정의 세부사항은 문서와 같이 대역 외으로 전달될 수 있다. 모든 장치 유형은 리소스 유형과 동일한 구문 구조 조건을 사용하여 리소스 유형 ID로 식별해야 한다.

#### (아) OCF 인터페이스

OCF 인터페이스는 먼저 리소스에 대한 보기를 제공한 다음 해당 리소스 보기에서 허용되는 요청 및 응답을 정의한다. 따라서 OCF 인터페이스에 의해 제공되는 이 보기는 리소스에 대한 요청 및 응답의 컨텍스트를 정의한다. 따라서 서로 다른 OCF 인터페이스를 대상으로 할 때 리소스에 대한 동일한 요청으로 인해 다른 응답이 발생할 수 있다.

OCF 인터페이스는 본 문서(핵심 OCF 인터페이스), ISO/IEC 30118-5:2018(수직 OCF 인터페이스) 또는 장치의 제조업체, 최종 사용자 또는 개발자(복수 OCF 인터페이스)에 의해 정의될 수 있다.

OCF 인터페이스 속성은 리소스 지원의 모든 OCF 인터페이스를 나열한다. 모든 자원은 적어도 하나의 OCF 인터페이스를 가져야 한다. 기본 OCF 인터페이스는 OCF 규격으로 정의되어야 하며 리소스 유형 정의에서 상속되어야 한다. 모든 OCF 정의 리소스 유형과 관련된 기본 OCF 인터페이스는 리소스 유형 정의의 해당 열거 내에서 먼저 나열된 지원되는 OCF 인터페이스여야 한다(이 문서에 정의된 OCF 정의 리소스 유형에 대해서는 부록 D 참조). 적용 가능한 열거값은 오픈에서 첫 번째 "path"의 "get" 방법에서 참조된 "parameters" 열거에 있다. 리소스 유형에 대한 API 2.0 파일("get"이 없는 경우 "post" 메서드). OCF 규격에 지정된 모든 기본 OCF 인터페이스는 필수 사항이다. OCF 규격에 정의된 OCF 인터페이스 외에 모든 자원은 기본 OCF 인터페이스("oic.if.baseline")를 지원해야 한다.

OCF 인터페이스는 둘 이상의 매체 유형을 허용할 수 있다. OCF 인터페이스는 둘 이상의 매체 유형으로 응답할 수 있다. 허용되는 미디어 유형은 응답 미디어 유형과 다를 수 있다.

미디어 유형은 전송 프로토콜에서 적절한 헤더 매개변수로 지정된다.

자원이 지원하는 OCF 인터페이스는 OCF 인터페이스 다음의 표와 같은 공통 속성을 사용하여 선언해야 한다. 예: "if": ["oic.if.ll", "oic.if.baseline"). OCF 인터페이스 속성의 속성 값은 세그먼트가 "."(점)로 분리된 소문자 문자열이어야 한다. OCF 인터페이스 속성 값에서 첫 번째 세그먼트로 사용되는 문자열 "oic"은 OCF 정의 OCF 인터페이스에 대해 예약된다. OCF 인터페이스 속성 값은 OCF 인터페이스의 정의를 찾는 데 사용될 수 있는 IANA와 유사한 권한에 대한 참조일 수도 있다. 자원 유형은 하나 이상의 OCF 인터페이스를 지원해야 한다.

표 4 - Resource Interface Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
OCF Interface	"if"	"array"	Array of strings, conveying OCF Interfaces	N/A	R	Yes	Property to declare the OCF Interfaces supported by a Resource.

OCF 인터페이스 방법은 별도로 정의된 리소스에 대해 정의된 OpenAPI 2.0 정의를 위반하지 않아야 한다. 정의된 OCF 인터페이스는 다음의 표에 나열되어 있다.

표 5 - OCF standard OCF Interface

OCF Interface	Name	Applicable Operations	Description
baseline	"oic.if.baseline"	RETRIEVE, NOTIFY, UPDATE	The baseline OCF Interface defines a view into all Properties of a Resource including the Meta Properties. This OCF Interface is used to operate on the full Representation of a Resource.
links list	"oic.if.ll"	RETRIEVE, NOTIFY	The links list OCF Interface provides a view into Links in a Collection (Resource). Since Links represent relationships to other Resources, the links list OCF Interfaces may be used to discover Resources with respect to a context. The discovery is done by retrieving Links to these Resources. For example: the Core Resource "/oic/res" uses this OCF Interface to allow discovery of Resource hosted on a Device.
batch	"oic.if.b"	RETRIEVE, NOTIFY, UPDATE	The batch OCF Interface is used to interact with a Collection of Resources at the same time. This also removes the need for the Client to first discover the Resources it is manipulating – the Server forwards the requests and aggregates the responses
read-only	"oic.if.r"	RETRIEVE NOTIFY	The read-only OCF Interface exposes the Properties of a Resource that may be read. This OCF Interface does not provide methods to update Properties, so can only be used to read Property Values.
read-write	"oic.if.rw"	RETRIEVE, NOTIFY, UPDATE	The read-write OCF Interface exposes only those Properties that may be read from a Resource during a RETRIEVE operation and only those Properties that may be written to a Resource during and UPDATE operation.
actuator	"oic.if.a"	RETRIEVE, NOTIFY, UPDATE	The actuator OCF Interface is used to read or write the Properties of an actuator Resource.
sensor	"oic.if.s"	RETRIEVE, NOTIFY	The sensor OCF Interface is used to read the Properties of a sensor Resource.

기준선 OCF 인터페이스를 사용하여 볼 수 있는 표현에는 공통 속성을 포함한 리소스의 모든 속성이 포함된다. 기본 OCF 인터페이스는 모든 리소스 유형에 대해 정의되어야 한다. 모든 자원은 기본 OCF 인터페이스를 지원해야 한다.

기준 OCF 인터페이스는 클라이언트가 리소스의 모든 속성을 검색하고자 할 때 사용된다. 즉, 서버는 리소스의 구현된 모든 속성을 포함하는 리소스 표현으로 응답해야 한다. 서버가 전체 리소스 표현을 다시 보낼 수 없는 경우 오류 메시지로 응답해야 한다. 서버는 부분적인 자원 표현을 반환해서는 안 된다.

UPDATE 사용 관련하여, 기준선 OCF 인터페이스를 사용하면 리소스 유형에 기준선을 사용하여 업데이트할 스키마가 연결된 경우 공통 속성을 제외한 리소스의 모든 속성을 속성 목록 및 원하는 값과 함께 UPDATE 요청을 사용하여 수정할 수 있다. 기본 OCF 인터페이스 외에 리소스에 의해 노출되는 OCF 인터페이스가 UPDATE 의미론을 지원하지 않는 경우, 기준선 OCF 인터페이스를 사용한 UPDATE도 지원되지 않는다

## An example response to a RETRIEVE request using the baseline OCF Interface:

```
{
  "rt": ["oic.r.temperature"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "temperature": 20,
  "units": "C",
  "range": [0,100]
}
```

링크 목록 OCF 인터페이스는 리소스의 링크 목록에 대한 보기를 제공한다. 이 OCF 인터페이스를 통해 볼 수 있는 표현에는 리소스별 링크의 배열(또는 배열)인 속성으로 노출된 링크만 있으므로, 이 OCF 인터페이스는 링크 목록을 조작하거나 상호작용하는 데 사용된다. 링크 목록은 이 OCF 인터페이스를 사용하여 RETRIEVE가 될 수 있다.

OCF 인터페이스는 다음과 같이 정의된다.

- 링크 목록 OCF 인터페이스 이름은 "oic.if.ll"이다.
- 리소스에 링크가 없는 경우, 링크 목록 OCF 인터페이스를 사용하여 RETRIEVE 요청에 응답하여 빈 목록을 반환해야 한다.
- 이 OCF 인터페이스에 의해 결정되는 표현은 요청 클라이언트에 따라 달라진다. 요청 시 OCF-Accept-Content-Format-Version 옵션이 포함된 클라이언트의 경우 응답은 링크 배열인 속성의 속성 값만 포함하므로 oic.if.ll을 사용한 수집 또는 "oic/res" 응답은 링크의 배열이다.
- 링크 목록 OCF 인터페이스를 사용하여 클라이언트가 링크 배열을 관찰할 수 있다.
- 링크 배열의 링크에 대한 모든 CREATE, UPDATE 또는 DELETE 작업은 관찰을 시작한 클라이언트에 전송되는 통지에 적용되는 필터링에 따라 대상 리소스에 대해 정의된 링크 목록 OCF 인터페이스에 대한 전체 리소스 표현을 생성해야 한다.
- Link 삭제 시 Links가 존재하지 않는 경우, 공지에 빈 목록이 전송되어야 한다.

---

```
GET ocf://<devID>/a/room/1?if=oic.if.ll
The response would be the array of OCF Links
```

```
[
  {
    "href": "/the/light/1",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:5555"}
    ]
  },
  {
    "href": "/the/light/2",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:5555"}]
  },
  {
    "href": "/my/fan/1",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:5555"}
      {
        "href": "/his/fan/2",
        "rt": ["oic.r.switch.binary"],
        "if": ["oic.if.a", "oic.if.baseline"],
        "eps": [
          {"ep": "coaps://[2001:db8:a::b1d4]:5555"}
        ]
      }
    ]
  }
]
```

---

배치 OCF 인터페이스는 단일/동일한 요청을 사용하여 리소스 집합과 상호 작용하기 위해 사용된다. 일괄 OCF 인터페이스는 단일 요청으로 연결된 리소스의 속성을 재설정하거나 업데이트하는 데 사용할 수 있다. 배치 OCF 인터페이스는 다음과 같이 정의된다.

- 배치 OCF 인터페이스 이름은 "oic.if.b"이다.
- 수집 리소스에는 URI로 표시되는 연결된 리소스가 있다. 배치 페이로드의 "href" 속성에서 URI는 원격 자원에 대한 완전한 자격과 지역 자원에 대한 상대적 참조를 가져야 한다.
- 원래 요청의 URI를 링크된 리소스의 URI로 대체하여 컬렉션의 각 링크된 리소스를 대상으로 하는 새

로운 요청을 생성하도록 원래 요청을 수정하십시오. 원래 요청의 페이로드는 새 요청의 페이로드에서 복제된다.

- 연결된 리소스의 기본 OCF 인터페이스를 사용할 경우 요청을 전달해야 한다.
- 요청은 관계 유형 "항목" 또는 "호스트"로 식별되는 링크된 리소스에만 전달되어야 한다("rel" 링크 매개 변수가 없을 경우 호스트는 기본 관계 유형 값임). 요청은 "항목" 또는 "호스트" 관계 유형 값을 포함하지 않는 연결된 자원에 전달되지 않는다.
- 수집 리소스 자체의 속성은 "oic.if"를 사용하여 페이로드에 포함될 수 있다.b" OCF 인터페이스: 컬렉션 내의 "항목"과 함께 링크 관계 "자체"가 있는 단일 링크를 노출하고, 링크 해상도가 재귀적 참조로 인해 무한 루프가 될 수 없도록 확인. 예를 들어, 컬렉션의 기본 OCF 인터페이스가 "oic.if.b"인 경우, 서버는 배치 표현에 배치 표현을 무한 루프에 반복적으로 포함할 수 있다.
- 링크 관계 "self"를 사용하여 수집 리소스의 기본 OCF 인터페이스가 노출되고 기본 OCF 인터페이스에 링크가 노출되는 특성이 있는 경우, 이러한 속성은 "self" 링크가 포함된 배치 표현에 포함되지 않아야 한다.
- 컬렉션("자체" 링크 참조 포함)인 링크된 리소스로 전달되는 모든 요청은 링크된 컬렉션 리소스의 기본 OCF 인터페이스를 적용해야 한다.
- 링크된 자원의 모든 응답은 클라이언트에 대한 단일 응답으로 통합되어야 한다. 서버는 시간 창에 대한 응답 시간을 초과하고, 서버는 조건에 따라 적절한 창을 선택할 수 있다.
- 링크된 리소스가 요청을 처리할 수 없는 경우, 빈 응답, 즉 "rep" 속성의 표현으로서 내용이 없는 JSON 객체("{}") 또는 링크된 리소스 유형이 오류 스키마 또는 진단 페이로드(payload)를 제공하는 경우 오류 응답은 링크된 리소스에 의해 반환되어야 한다. 오류를 나타내는 모든 연결된 자원에 대한 이러한 빈 또는 오류 응답은 원래 클라이언트 요청에 대한 집계 응답에 포함되어야 한다.
- 연결된 리소스 중 오류 응답을 반환하는 리소스가 있을 경우 고객에게 전송된 집계 응답도 오류를 표시해야 한다(예: CoAP의 4.xx). 연결된 모든 자원이 성공적인 응답을 반환하는 경우 집계된 응답에는 성공 응답 코드가 포함되어야 한다.
- 집계된 응답은 연결된 각 리소스의 응답을 나타내는 개체 배열이어야 한다. 응답의 각 개체에는 (1) 링크된 리소스의 URI(원격 리소스에 대해 정규화된 URI 또는 로컬 리소스에 대한 상대적 참조)와 (2) 링크된 리소스가 키로 "rep"을 사용한 수집인 경우 개별 응답 개체 또는 개체 배열의 최소 두 가지 항목이 포함되어야 한다. (예: "rep": { <representation of individual response> }. )
- 수집 리소스가 관찰 가능한 것으로 표시된 경우, 수집에서 참조된 링크된 리소스를 배치 OCF 인터페이스를 사용하여 관찰할 수 있다. 수집 자원이 관찰 가능한 것으로 표시되지 않는 경우 수집을 관찰할 수 없으며 수집에 대한 요청 관찰은 요청 유효성 검사가 실패한 경우에 대해 정의된 대로 처리해야 한다. 요청에 대한 모든 응답은 배치 OCF 인터페이스를 사용한 RETRIEVE 작업과 동일한 표현 및 상태 코드를 사용하여 클라이언트에 대한 단일 응답으로 집계되어야 한다.
- 관찰 가능한 링크된 리소스 중 하나가 관찰 요청을 이행하지 못할 경우 배치 관찰 요청에 대한 응답은 전체 요청이 수행되지 않았음을 표시해야 한다. (네트워크 요청을 수행하는 링크와 로컬 리소스를 비교하는 동작은 다를 수 있다.)
- 배치 OCF 인터페이스를 사용하여 관찰 요청을 시작한 클라이언트에 대한 모든 통지는 수집에 대한 배치 표현을 사용해야 한다. 이는 연결된 리소스에 전달된 개별 관찰 요청에서 수집을 호스팅하는 디바이스가 수신한 개별 관찰 통지의 집계입니다.
- 컬렉션의 링크에서 관찰할 수 있는 것으로 표시되지 않는 링크된 자원은 통지를 트리거하지 않아야 하지만, 컬렉션의 배치 OCF 인터페이스에 대한 관찰 요청에 대한 응답과 그에 따른 후속 통지에 포함될 수 있다.
- 각 통지에는 원래 관찰 요청이 다시 처리될 경우 포함되는 모든 링크된 리소스에 대한 최신 값이 포함되어야 한다. 수집을 호스팅하는 서버는 매번 모든 링크된 리소스를 다시 검색하도록 선택하거나, 각 알림에서 링크된 리소스를 검색하지 않도록 캐싱을 사용하도록 선택할 수 있다.
- Linked Resource가 관찰 가능하고 성공적인 Observe 응답으로 응답한 경우, 해당 리소스의 가장 최근에 보고된 값은 가장 최근의 값으로 간주되며 이후의 모든 알림에 보고될 수 있다.
- 컬렉션의 링크는 "oic.if.ll" OCF 인터페이스를 사용하여 관찰해야 한다. "oic.if.ll" OCF 인터페이스 표현 내용이 변경될 때마다 통지가 전송되어야 한다. 즉, 링크가 추가되거나 링크가 제거되거나 링크가 업데이트된 경우. "oic.if.ll" OCF 인터페이스에 대한 알림은 "oic.if.ll" OCF 인터페이스 표현에 있는 모든 링크를 포함해야 한다.
- "oic.if.baseline" OCF 인터페이스를 포함하여 리소스 유형 정의에 정의된 OCF 인터페이스를 사용하여 수집 리소스의 다른 속성을 관찰할 수 있다.
- 고객은 요청에 추가 쿼리 매개 변수를 포함시켜 요청이 전달되는 링크된 리소스를 제한할 수 있다. 서버는 요청에 의해 처리될 연결된 리소스에 대한 선택 도구 "oic.if.b"를 포함하는 요청에서 추가 쿼

리 매개 변수를 처리해야 한다.

- 클라이언트는 배치 OCF 인터페이스 요청에서 RETRIEVE 응답 페이로드와 유사한 페이로드(payload)를 생성하여 배치 OCF 인터페이스를 사용하여 업데이트 작업을 수행해야 한다. 서버는 각각의 "href" 속성과 "rep" 속성의 해당 값에 따라 연결된 각 리소스에 별도의 UPDATE 요청을 보내야 한다.
- "href" 값이 0 길이 문자열 또는 JSON의 ""로 표시된 비어 있는 경우, "rep" 속성은 컬렉션의 링크된 자원에 적용되어야 한다.
- "href"와 링크별 "href"가 비어 있는 항목은 동일한 UPDATE 요청에 혼합해서는 안 된다.
- UPDATE 요청의 모든 속성은 링크된 리소스에서 지원되지 않을 수 있다. 이 경우 링크된 자원이 지원하는 UPDATE 요청의 쓰기 가능한 속성을 수정해야 하며 지원되지 않는 속성은 자동으로 무시해야 한다.
- UPDATE 응답에는 적절한 상태 코드와 함께 링크된 리소스가 제공하는 경우 RETRIEVE 작업과 동일한 페이로드 스키마를 사용하여 업데이트된 값이 포함되어야 한다. 집계 응답 페이로드에는 배치 업데이트가 완료된 후 업데이트된 속성의 알려진 상태가 반영되어야 한다. 업데이트된 리소스에서 페이로드(payload)가 제공되지 않는 경우 해당 리소스에 대해 빈 응답(즉, "rep": {})이 제공되어야 한다.
- 수집은 "oic.if.baseline", "ic.if.rw" 또는 "oic.if"를 사용하여 기존 컬렉션에서 링크를 추가, 수정 또는 제거하는 UPDATE 작업을 지원하지 않아야 한다." OCF 인터페이스

다음의 표에 제시된 예는 모든 경우에 모든 필수 스키마 요소를 포함하지는 않는다. 리소스 유형 "x.org.example.rt.room"의 기본 OCF 인터페이스가 리소스 유형 정의 파일에서 "x.org.example.colour" 및 "x.org.example.size". 속성을 노출하는 "oic.if.rw"으로 지정되었다고 가정한다.

표 6 - Batch OCF Interface Example

<p><b>Resources</b></p>	<pre> /a/room/1 {   "rt": "x.org.example.rt.room",   "if": ["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],   "x.org.example.colour": "blue",   "x.org.example.dimension": "15bx15wx10h",   "links": [     { "href": "/a/room/1", "rel": ["self", "item"], "rt": ["x.org.example.rt.room"], "if": ["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],"p": {"bm": 2} },     { "href": "/the/light/1", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a","oic.if.baseline"], "ins": "11111", "p": {"bm": 2} },     { "href": "/the/light/2", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a","oic.if.baseline"], "ins": "22222", "p": {"bm": 2} },     { "href": "/my/fan/1", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "ins": "33333", "p": {"bm": 2} },     { "href": "/his/fan/2", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "ins": "44444", "p": {"bm": 2} },     { "href": "/the/switches/1", "rel": ["item"], "rt": ["oic.wk.col"], "if":["oic.if.ll", "oic.if.b", "oic.if.baseline"], "ins": "55555", "p": {"bm": 2} }   ] }  /the/light/1 {   "rt": ["oic.r.switch.binary"],   "if": ["oic.if.a", "oic.if.baseline"],   "value": false }  /the/light/2 {   "rt": ["oic.r.switch.binary"],   "if": ["oic.if.a", "oic.if.baseline"],   "value": true }  /my/fan/1 {   "rt": ["oic.r.switch.binary"],   "if": ["oic.if.a", "oic.if.baseline"],   "value": true }  /his/fan/2 {   "rt": ["oic.r.switch.binary"],   "if": ["oic.if.a", "oic.if.baseline"],   "value": false }  /the/switches/1 {   "rt": ["oic.wk.col"],   "if":["oic.if.ll", "oic.if.b", "oic.if.baseline"],   "links": [     {       "href": "/switch-1a",       "rt": ["oic.r.switch.binary"],       "if": ["oic.if.a","oic.if.baseline"],       "p": {"bm": 2}     }   ] } </pre>
	<pre> {   "href": "/switch-1b",   "rt": ["oic.r.switch.binary"],   "if": ["oic.if.a","oic.if.baseline"],   "p": {"bm": 2 } } ] } </pre>

<b>Use of batch, successful response</b>	<pre> Request: GET /a/room/1?if=oic.if.b Becomes the following individual request messages issued by the Device in the Client role GET /a/room/1 (NOTE: uses the Default OCF Interface as specified for the Collection Resource, in this example oic.if.rw) GET /the/light/1 (NOTE: Uses the Default OCF Interface as specified for this Resource) GET /the/light/2 (NOTE: Uses the Default OCF Interface as specified for this Resource) GET /my/fan/1 (NOTE: Uses the Default OCF Interface as specified for this Resource) GET /his/fan/2 (NOTE: Uses the Default OCF Interface as specified for this Resource) GET /the/switches/1 (NOTE: Uses the Default OCF Interface for the Collection that is within the Collection) Response: [   {     "href": "/a/room/1",     "rep": {"x.org.example.colour": "blue", "x.org.example.dimension": "15bx15wx10h"}   },   {     "href": "/the/light/1",     "rep": {"value": false}   },   {     "href": "/the/light/2",     "rep": {"value": true}   },   {     "href": "/my/fan/1",     "rep": {"value": true}   },   {     "href": "/his/fan/2",     "rep": {"value": false}   },   {     "href": "/the/switches/1",     "rep": [       {         "href": "/switch-1a",         "rt": ["oic.r.switch.binary"],         "if": ["oic.if.a", "oic.if.baseline"],         "p": {"bm": 2},         "eps": [           {"ep": "coaps://[2001:db8:a::b1d4]:55555"}         ]       },       {         "href": "/switch-1b",         "rt": ["oic.r.switch.binary"],         "if": ["oic.if.a", "oic.if.baseline"],         "p": {"bm": 2},         "eps": [           {"ep": "coaps://[2001:db8:a::b1d4]:55555"}         ]       }     ]   } ] </pre>
--	---

<b>Use of batch, error response</b>	<p>Should any of the RETRIEVE requests in the previous example fail then the response includes an empty payload for that Resource instance and an error code is sent. The following example assumes errors from "/my/fan/1" and "/the/switches/1"</p> <p>Error Response:</p> <pre>[   {     "href": "/a/room/1",     "rep": {"x.org.example.colour": "blue", "x.org.example.dimension": "15bx15wx10h"}   },   {     "href": "/the/light/1",     "rep": {"value": false}   },   {     "href": "/the/light/2",     "rep": {"value": true}   },   {     "href": "/my/fan/1",     "rep": {}   },   {     "href": "/his/fan/2",     "rep": {"value": false}   },   {     "href": "/the/switches/1",     "rep": {}   } ]</pre>
-------------------------------------	--

<p><b>Use of batch</b></p> <p><b>(UPDATE has POST semantics)</b></p>	<pre>UPDATE /a/room/1?if=oic.if.b [   {     "href": "",     "rep": {       "value": false     }   } ]</pre> <p>Since the "href" value in the UPDATE request is empty, the request is forwarded to all Resources in the Collection and becomes:</p> <pre>UPDATE /a/room/1 { "value": false } UPDATE /the/light/1 { "value": false } UPDATE /the/light/2 { "value": false } UPDATE /my/fan/1 { "value": false } UPDATE /his/fan/2 { "value": false } UPDATE /the/switches/1 { "value": false }</pre> <p>Response:</p> <pre>[   {     "href": "/the/light/1",     "rep": {"value": false}   },   {     "href": "/the/light/2",     "rep": {"value": false}   },   {     "href": "/my/fan/1",     "rep": {"value": false}   },   {     "href": "/his/fan/2",     "rep": {"value": false}   },   {     "href": "/the/switches/1",     "rep":       {       }   } ]</pre> <p>Since <code>/a/room/1</code> does not have a "value" Property exposed by its Default OCF Interface, the UPDATE request will be silently ignored and it will not be included in the UPDATE response.</p> <p>Since the UPDATE request with the links list OCF Interface is not allowed, an empty payload for the <code>/the/switches/1</code> is included in the UPDATE response and an error code is sent.</p>
--	--

<p><b>Use of batch (UPDATE has POST semantics)</b></p>	<pre>UPDATE /a/room/1?if=oic.if.b [   {     "href": "/the/light/1",     "rep": {       "value": false     }   },   {     "href": "/the/light/2",     "rep": {       "value": true     }   },   {     "href": "/a/room/1",     "rep": {       "x.org.example.colour": "red"     }   } ]</pre> <p>This turns /the/light/1 off, turns /the/light/2 on, and sets the colour of /a/room/1 to "red".</p> <p>The response will be same as response for GET /a/room/1?if=oic.if.b with the updated Property values as shown.</p> <pre>[   {     "href": "/a/room/1",     "rep": {"x.org.example.colour": "red",            "x.org.example.dimension": "15bx15wx10h"}   },   {     "href": "/the/light/1",     "rep": {"value": false}   },   {     "href": "/the/light/2",     "rep": {"value": true}   } ]</pre> <p>Example use of additional query parameters to select items by matching Link Parameters.</p> <p>Turn on light 1 based on the "ins" Link Parameters value of "11111"</p> <pre>UPDATE /a/room/1?if=oic.if.b&amp;ins=11111 [   {     "href": "",     "rep": {       "value": false     }   } ]</pre> <p>Similar to the earlier example, "href": "" applies the UPDATE request to all of the Resources in the Collection. Since the additional query parameter ins=11111 selects only links that have a matching "ins" value, only one link is selected. The payload is applied to the target Resource of that link, /the/light/1.</p>
--	--

	<p>Retrieving the item using the same query parameter:</p> <pre>RETRIEVE /a/room/1?if=oic.if.b&amp;ins=11111</pre> <p>Response payload:</p> <pre>[   {     "href": "/the/light/1",     "rep": {       "value": false     }   } ]</pre>
--	--

액추에이터 OCF 인터페이스는 작동할 수 있는 자원을 보기 위한 OCF 인터페이스로, 즉, 자원에 의해 추상화된 엔티티 내부 또는 엔티티의 상태를 변경한다.

- 액추에이터 OCF 인터페이스 이름은 "oic.if.a"이어야 한다.
- 액추에이터 OCF 인터페이스는 해당 개방에 의해 정의된 모든 필수 속성을 리소스 표현에 노출해야 한다. API 2.0 스키마; 액추에이터 OCF 인터페이스는 해당 Open에 의해 정의된 대로 리소스 표현 옵션 속성에도 노출될 수 있음. 대상 디바이스에 의해 구현되는 API 2.0 스키마.

---

For example, a "Heater" Resource (for illustration only):

```
/a/act/heater
{
  "rt": ["acme.gas"],
  "if": ["oic.if.baseline", "oic.if.r", "oic.if.a", "oic.if.s"],
  "settemp": 10,
  "currenttemp" : 7
}
```

The actuator OCF Interface with respect to "Heater" Resource (for illustration only):

a) Retrieving values of an actuator.

Request: GET /a/act/heater?if="oic.if.a"

```
Response:
{
  "settemp": 10,
  "currenttemp" : 7
}
```

b) Correct use of actuator OCF Interface.

Request: POST /a/act/heater?if="oic.if.a"

```
{
  "settemp": 20
}
```

```
Response:
{
  ok
}
```

c) Incorrect use of actuator OCF Interface.

Request: POST /a/act/heater?if="oic.if.a"

```
{
  "if": ["oic.if.s"] ← this is visible through baseline OCF Interface
}
```

```
Response:
{
  Error
}
```

- 
- 이 OCF 인터페이스를 사용하는 RETRIEVE 요청은 존재하는 쿼리 및 필터 매개변수에 따라 이 자원에 대한 표현을 반환해야 한다.
  - 이 OCF 인터페이스를 사용한 UPDATE 요청은 대상 리소스에 업데이트될 속성을 포함하는 페이로드 또는 본문을 제공해야 한다.

센서 OCF 인터페이스는 다음을 감지하는 리소스에서 측정, 감지 또는 기능 특정 정보를 검색하기 위한 OCF 인터페이스이다.

- 센서 OCF 인터페이스 이름은 "oic.if.s"이어야 한다."
- 센서 OCF 인터페이스는 해당 오픈에 의해 정의된 모든 필수 속성을 자원 표현에 노출해야 한다. API 2.0 스키마; 센서 OCF 인터페이스는 해당 Open에 의해 정의된 리소스 표현 선택적 속성에도 노출될 수 있음. 대상 디바이스에서 구현하는 API 2.0 스키마.
- 이 OCF 인터페이스를 사용하는 RETRIEVE 요청은 존재하는 모든 쿼리 및 필터 매개 변수에 따라 해당 리소스에 대한 이 표현을 반환해야 한다. (참고: 예시는 센서의 값 검색과 관련된 것이다.)

읽기 전용 OCF 인터페이스는 읽을 수 있는 속성만 노출한다. 여기에는 읽기 전용, 읽기 전용이지만 쓰기 전용 또는 설정 전용의 속성이 아닌 속성이 포함된다. 리소스에 적용할 수 있는 작업은 RETRIEVE 및 NOTIFY일 뿐이다. RETRIEVE 또는 NOTIFY 이외의 방법을 리소스에 적용하려는 클라이언트의 시도는 오류 응답 코드로 거부되어야 한다.

읽기-쓰기 OCF 인터페이스는 리소스에서 속성을 읽고 설정할 수 있는 일반적인 OCF 인터페이스이다. 리소스에 적용할 수 있는 방법은 RETRIEVE, NOTIFY 및 UPDATE일 뿐이다. RETRIEVE 및 NOTIFY 작동의 경우 "oic.if.r" OCF 인터페이스와 동일한 동작이다. 업데이트 작업의 경우 읽기 전용 속성(즉, 열기에

서 "readOnly=True"로 태그가 지정된 속성)API 2.0 정의)는 UPDATE 페이로드에 있으면 안 된다. RETRIEVE, NOTIFY 또는 UPDATE 이외의 방법을 리소스에 적용하려는 클라이언트의 시도는 오류 응답 코드로 거부되어야 한다.

---

```
Request: GET /a/act/heater?if="oic.if.s"
Response:
{
  "currenttemp": 7
}

Incorrect use of the sensor.

Request: PUT /a/act/heater?if="oic.if.s" ← PUT is not allowed
{
  "settemp": 20 ← this is possible through actuator OCF Interface
}
Response:
{
  Error
}

Another incorrect use of the sensor.

Request: POST /a/act/heater?if="oic.if.s" ← POST is not allowed
{
  "currenttemp": 15 ← this is possible through actuator OCF Interface
}
Response:
{
  Error
}
```

---

리소스 표현은 특정 시간에 리소스의 상태를 캡처한다. 리소스 표현은 요청 및 리소스와의 응답 상호 작용에서 교환된다. 리소스 표시는 리소스의 상태를 검색하거나 업데이트하는 데 사용될 수 있다. 리소스 표현은 데이터 연결 프로토콜 및 기술(예: CoAP, UDP/IP 또는 BLE)에 의해 조작되지 않아야 한다.

많은 시나리오와 맥락에서, 자원은 그들 사이에 암묵적 또는 명시적 구조를 가질 수 있다. 프레임워크는 이러한 구조와 자원 사이의 관계를 모델링하고 매핑하는 수단을 제공한다. 프레임워크의 리소스 구조에 대한 기본 구성 요소는 수집이다. 컬렉션은 복잡한 구조를 모델링할 수 있는 컨테이너를 나타낸다.

자원 관계는 링크(Links) 링크는 리소스 간의 관계를 표현하는 수단으로서 입력된 웹 링크 개념을 수용하고 확장한다. 링크는 다음을 정의하는 매개변수 집합으로 구성된다.

- 컨텍스트 URI,
- 목표 URI,
- 컨텍스트 URI에서 대상 URI까지의 관계 및
- 대상 URI, 링크 관계 또는 컨텍스트에 대한 메타데이터를 제공하는 요소.

대상 URI는 필수 사항이고 링크에 있는 다른 항목은 선택 사항이다. 링크의 추가 항목은 서로 다른 컨텍스트(예: 수집, 검색, 브리지 등)에서 링크의 사용에 기초하여 의무화될 수 있다. 링크 페이로드에 대한 OpenAPI 2.0 스키마는 별도로 제공된다.

링크의 예는 다음과 같다.

---

#### An example of a Link is:

```
{"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "p": {"bm": 3}, "rel": "item"}
```

---

두 개의 링크는 최소한 하나의 파라미터가 다를 때 서로 구별된다. 예를 들어, 여기에 표시된 두 링크는 서로 다르며 동일한 링크 목록에 나타날 수 있다.

---

```
{"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "p": {"bm": 2}, "rel": "item"}
{"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a", "oic.if.baseline"], "p": {"bm": 2}}
```

---

문서에서는 특정 기능에 필요한 매개변수 및 매개변수 값을 위임할 수 있다.

"/oic/res"의 RETRIEVE에 대한 응답으로 반환되는 모든 링크의 경우, 링크에 "rel" 매개 변수가 명시적으로 포함되어 있지 않으면 "rel"="hosts"의 값이 가정되어야 한다. "호스트"의 관계 값은 IETF RFC 6690, IETF RFC 6573에 의한 "항목"의 값, IETF RFC 4287에 의한 "자체"의 값이며, 모두 IANA 링크 관계에 정의된 링크 관계 IANA 등록부에 등록되어 있다.

링크에서 컨텍스트 URI와 대상 URI 사이의 관계는 "rel" JSON 요소를 사용하여 지정되며 이 요소의 값은 특정 관계를 지정한다. 링크의 컨텍스트 URI는 링크가 "앵커" 매개변수를 지정하지 않는 한 링크가 포함된 리소스의 URI(또는 특히 컬렉션)가 암시적으로 되어야 한다. "anchor" 매개변수는 링크의 컨텍스트 URI를 변경하는 데 사용된다. 대상 URI와의 관계는 "anchor"가 지정되었을 때 앵커 URI에 기초한다. "Anchor" 매개변수는 OIC 1.1 링크에 대한 전송 프로토콜 URI(예: "anchor": "caps:[fe80:b1d6]:44444") 및 OCF 1.0 링크(예: "anchor": "ocf://d70373c-4f963-086aaaa)를 사용한다. OIC 1.1과의 역호환성(선택 사항)을 위해, "anchor" 파라미터는 OIC 1.1 링크용 전송 프로토콜 URI를 사용한다

(예)."anchor": "coaps://[fe80::b1d6]:44444".

수집의 맥락에서 "앵커"를 사용하는 예(바닥에는 방이 있고 방은 조명이 있음)는 바닥에는 조명을 링크로 정의할 수 있지만 링크는 조명을 포함하는 방의 URI에 "anchor"를 설정한다. 이렇게 하면 바닥에 있는 모든 조명을 함께 켜거나 끌 수 있으며, 여전히 조명을 포함하고 있는 방에 대해 정의되어 있다(조명은 또한 룸 URI를 사용하여 켤 수도 있다). 예를 들어, 링크에서 "anchor"의 용도는 다음과 같다

```

/a/floor {
  "links": [
    {
      "href": "/x/light1",
      "anchor": "/a/room1",  ** Note: /a/room1 has the item relationship with /x/light1; not /a/floor **
      "rel": "item"
    }
  ]
}

/a/room1 {
  "links": [
    {
      ** Note: /a/room1 "contains" the /x/light since /a/room1 is the implicit context URI **
      "href": "/x/light1",
      "rel": "item"
    }
  ]
}

```

다음은 Parameters (파라미터) 관련한 설명이다.

"in" 매개 변수는 링크 목록에서 특정 링크 인스턴스를 식별한다. "in" 매개변수는 링크 목록의 특정 링크를 수정하거나 삭제하는 데 사용될 수 있다. "in" 매개변수의 값은 링크 목록을 호스팅하는 OCF 장치(서버)에 의한 링크 인스턴스화 시 설정되며, 일단 설정되면 "ins" 매개변수는 링크가 해당 목록의 구성원이 되는 한 수정되지 않아야 한다.

정책 매개 변수는 대상 URI에서 참조하는 리소스에 올바르게 액세스하기 위한 다양한 규칙을 정의한다. 정책 규칙은 키 값 쌍 집합에 의해 구성된다.

정책 매개변수 "p"는 다음에 의해 정의된다.

- "bm" 키: "bm" 키는 8비트 비트마스크로 해석되는 정수 값에 해당한다.

비트마스크의 각 비트는 특정 정책 규칙에 해당한다. 규칙은 다음 표의 "bm"에 대해 지정된다.

표 7 - "bm" Property definition

Bit Position	Policy rule	Comment
Bit 0 (the LSB)	discoverable	The discoverable rule defines whether the Link is to be included in the Resource discovery message via "/oic/res". If the Link is to be included in the Resource discovery message, then "p" shall include the "bm" key and set the discoverable bit to value 1. If the Link is NOT to be included in the Resource discovery message, then "p" shall either include the "bm" key and set the discoverable bit to value 0 or omit the "bm" key entirely.
Bit 1 (2 <sup>nd</sup> LSB)	observable	The Observable rule defines whether the Resource referenced by the target URI supports the NOTIFY operation. With the self-link, i.e. the Link with "rel" value of "self", "/oic/res" can have a Link with the target URI of "/oic/res" and indicate itself Observable. The "self" is defined by IETF RFC 4287 and registered in the IANA Registry for "rel" value defined at IANA Link Relations. If the Resource supports the NOTIFY operation, then "p" shall include the "bm" key and set the Observable bit to value 1. If the Resource does NOT support the NOTIFY operation, then "p" shall either include the "bm" key and set the Observable bit to value 0 or omit the "bm" key entirely.
Bits 2-7	--	Reserved for future use. All reserved bits in "bm" shall be set to value 0.

(참고) "bm"의 모든 비트가 0으로 정의된 경우, 효율성 측정으로 "bm" 키를 "p"에서 완전히 생략할 수 있다. 그러나 비트가 값 1로 설정된 경우 "bm"은 "p"에 포함되어야 하며 모든 비트는 적절하게 정의되어야 한다.

- OCF-Accept-Content-Format-Version 옵션이 요청에 포함되지 않은 경우 나머지 반환의 "sec" 및 "포트"는 응답 페이로드에서만 사용해야 한다. "sec" 및 "port"를 포함하는 요청에 대한 응답으로 전송된 페이로드에서, "eps" 매개변수는 암호화된 연결에 대한 정보를 제공해야 한다.
- "sec" 키: "sec" 키는 대상 URI에서 참조하는 리소스가 암호화된 연결을 통해 액세스되는지 여부를 나타내는 부울 값에 해당한다. "sec"가 true이면 지정된 "포트"를 사용하여 암호화된 연결을 통해 리소스에 액세스한다. "sec"가 false인 경우, 리소스는 암호화되지 않은 연결 또는 암호화된 연결을 통해 액세스된다(이러한 연결이 "sec"가 true인 다른 리소스에 대한 "포트" 설정을 사용하여 이루어지는 경우).
- "포트" 키: "포트" 키는 대상 URI에서 참조하는 리소스를 암호화된 연결을 통해 액세스할 수 있는 포트 번호를 나타내는 데 사용되는 정수 값에 해당한다.
- 암호화된 연결(예: IP를 통한 DTLS)을 통해서만 리소스를 사용할 수 있는 경우
- "p"에는 "sec" 키가 포함되어야 하며, 그 값은 참이어야 한다.
- "p"에는 "포트" 키가 포함되어야 하며, 그 값은 암호화된 연결이 설정될 수 있는 포트 번호가 되어야 한다.
- 암호화되지 않은 연결을 통해서만 리소스를 사용할 수 있는 경우
- "p"에는 "sec" 키가 포함되어야 하며, 그 값은 거짓이거나 "p"는 "sec" 키를 생략해야 하며, 기본값인 "sec"는 거짓이다.
- "p"는 "포트" 키를 생략한다.
- 암호화된 연결과 암호화되지 않은 연결을 통해 리소스를 사용할 수 있는 경우
- "p"에는 "sec" 키가 포함되어야 하며, 그 값은 거짓이거나 "p"는 "sec" 키를 생략해야 하며, 기본값인 "sec"는 거짓이다.
- "p"는 "포트" 키를 생략할 수 있다. "포트" 키가 생략된 경우, 자원은 "sec"가 참인 장치의 다른 리소스와 동일한 "포트" 정보를 사용하여 사용할 수 있어야 한다.
- "포트" 키에 의해 지정된 포트의 리소스에 대한 접근은 암호화된 연결(예: "갑스://")에 의해 이루어져야 한다(참고로 멀티캐스트 검색을 통해 발견된 별도의 포트에서 리소스에 대한 암호화되지 않은 연결이 가능할 수 있다).
- ACL이 리소스에 대한 액세스를 제어한다는 점에 유의하십시오. 암호화된 연결이 성공한다고 해서 요청된 작업이 성공할 수 있는 것은 아니다. (자세한 내용은 ISO/IEC 30118-2:2018 조항 12 참조)

예 1: 검색 가능하지만 관측 가능하지 않으며 인증된 액세스가 CoAPS 포트 33275를 통해 수행되어야 하는 리소스에 대한 정책 매개 변수를 보여준다.

---

```
"p": {"bm": 1}
```

---

예 2: 이것은 자체 링크, 즉 발견 가능하고 관찰 가능한 "/oic/res" 링크를 보여준다.

---

```
{
  "href": "/oic/res",
  "rel": "self",
  "rt": ["oic.wk.res"],
  "if": ["oic.if.ll", "oic.if.baseline"],
  "p": {"bm": 3}
}
```

---

"유형" 매개변수를 사용하여 특정 대상 리소스에 의해 지원되는 다양한 미디어 유형을 지정할 수 있다. "유형" 요소가 생략된 경우 "application/vnd.ocf+cbor"의 기본 유형을 사용해야 한다. 클라이언트가 각 리소스에 대해 이 정보를 검색하면 요청 또는 응답의 해당 헤더 필드에 사용 가능한 표현 중 하나를 사용할 수 있다.

"di" 매개 변수는 "href" 매개 변수의 에 정의된 대상 리소스를 호스팅하는 장치의 장치 ID를 지정한다. 장치 ID는 "href"에 사용된 상대적 참조의 자격을 얻거나 상대적 참조에 대한 OCF Endpoint 정보를 조회하는 데 사용될 수 있다.

"eps" 매개 변수는 대상 리소스의 OCF Endpoint 정보를 나타낸다. "eps"는 항목 배열의 값으로 되어야 하며, 각 항목은 "ep"과 "pri"를 가진 OCF Endpoint 정보를 나타낸다. "ep"은 필수 사항이지만 "pri"는 선택 사항이다.

다음은 여러 개의 OCF endpoint를 가진 "eps"의 예다.

---

```
"eps": [
  {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
  {"ep": "coaps://[fe80::b1d6]:1122"},
  {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
]
```

---

링크에 "eps"가 있는 경우, "eps"의 OCF Endpoint 정보를 사용하여 "href" 매개 변수가 참조하는 대상 리소스에 액세스할 수 있다. 리소스가 노출하는 OCF Endpoint 유형(보안 또는 보안 해제)은 리소스에 요청을 보내는 데 사용할 수 있도록 보장된 연결 유형만 결정한다.

예를 들어, 리소스가 단일 CoAP "ep"만 노출하는 경우, 보안 OCF Endpoint(예: 다른 리소스의 "eps" 정보)에서 CoAPS "ep"을 통해서도 리소스에 액세스할 수 없다는 것을 보증하지 않는다. 또한 주어진 유형의 OCF Endpoint를 노출하면 "ep" 정보를 사용하여 리소스에 대한 액세스가 허가되지 않는다. 액세스 제어 계층에 의해 리소스에 대한 요청이 허가되는지 또는 거부되는지 여부는 "eps" 정보와 분리되며 /acl2 리소스의 구성에 의해 결정된다(자세한 내용은 ISO/IEC 30118-2:2018 조항 13.5.3 참조).

존재하는 경우, 최대 연령 정보(예: IETF RFC 7252에서 정의된 CoAP에 대한 Max-Age 옵션)는 오래된 것으로 간주되기 전에 최대 시간 "eps" 값을 캐시할 수 있음을 결정한다.

각 링크가 개별적으로 액세스할 수 있는 링크 배열로 정의된 하나 이상의 속성을 노출하는 리소스는 컬렉션과 같은 링크 배열에는 속성 이름 "링크"가 권장된다.

링크 배열을 정의하는 리소스 유형(예: 수집, 원자성 측정)이 링크 배열 내에 있을 수 있는 "rt" 값에 제한이 있는 경우, 리소스 유형은 "rts" 속성을 정의한다. 표 11에 정의된 "rts" 속성은 배열의 모든 링크에 대해 허용되는 모든 "rt" 값을 포함할 것이다. 리소스 유형이 "rts" 속성을 정의하지 않거나 "rts" 속성이 빈 배열인 경우 링크 배열에서 "rt" 값이 허용된다.

"rts" 속성을 정의하는 리소스 유형의 모든 인스턴스에 대해, 링크 배열의 모든 링크에 있는 "rt" 링크 매개변수는 "rt" 속성에 포함된 "rt" 값 중 하나가 되어야 한다.

표 8 - Resource Types Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Types	"rts"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	No	An array of Resource Types that are supported within an array of Links exposed by a Resource.

"rts" 속성이 있는 경우, 리소스 유형은 링크 배열에 필요한 "rt" value 를 모두 포함하는 다음 표에 정의된 대로 "rts-m" 속성을 정의한다. "rts-m"이 정의되고 "rts"가 정의되어 빈 배열이 아닌 경우, "rts-m"에 있는 "rt" 값은 "rts"에 있는 값의 일부가 된다. 또한, "rts-m" 속성이 정의된 경우, 자원 정의 및 내검사 장치 데이터에서 "rts-m" 속성이 의무화되어야 한다.

```

/Room1
{
  "rt": ["my.room"],
  "if": ["oic.if.ll", "oic.if.baseline" ],
  "color": "blue",
  "links":
  [
    {
      "href": "/oic/d",
      "rt": ["oic.d.light", "oic.wk.d"],
      "if": [ "oic.if.r", "oic.if.baseline" ],
      "p": {"bm": 1}
    },
    {
      "href": "/oic/p",
      "rt": ["oic.wk.p"],
      "if": [ "oic.if.r", "oic.if.baseline" ],
      "p": {"bm": 1}
    },
    {
      "href": "/switch",
      "rt": ["oic.r.switch.binary"],
      "if": [ "oic.if.a", "oic.if.baseline" ],
      "p": {"bm": 3},
      "mt": [ "application/vnd.ocf+cbor", "application/exi+xml" ]
    },
    {
      "href": "/brightness",
      "rt": ["oic.r.light.brightness"],
      "if": [ "oic.if.a", "oic.if.baseline" ],
      "p": {"bm": 3}
    }
  ]
}

```

"rts-m" 속성을 정의하는 리소스 유형의 모든 인스턴스에 대해, "rts-m" 속성의 "rt" 값의 각 하나에 해당하는 링크의 배열에 적어도 하나의 링크가 있어야 하며, 그러한 모든 링크에 대해 "rt" 링크 매개변수는 "rts-m" 속성에 하나 이상의 "rt" 값을 포함해야 한다.

표 9 - Mandatory Resource Types Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Mandatory Resource Types	"rts-m"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	No	An array of Resource Types that are mandatory to be exposed within an array of Links exposed by a Resource.

다른 리소스에 대한 하나 이상의 참조(연결로 지정)를 포함하는 리소스가 컬렉션입니다. 간단한 자원은 컬렉션과 구별된다. 리소스 참조를 링크로 바인딩하여 모든 리소스를 컬렉션으로 변환할 수 있다. 수집은 계층, 인덱스, 그룹 등을 작성, 정의 또는 지정하는 데 사용될 수 있다.

컬렉션은 라이프사이클 동안 항상 하나 이상의 리소스 유형과 하나 이상의 OCF 인터페이스가 바인딩되

어야 한다. 컬렉션 생성 시간 동안 리소스 유형 및 OCF 인터페이스가 지정된다. 초기 정의된 리소스 유형 및 OCF 인터페이스는 라이프사이클 동안 업데이트될 수 있다. 이러한 초기 값은 리소스의 경우 재지정에 사용되는 메커니즘을 사용하여 재지정할 수 있다. 추가 리소스 유형 및 OCF 인터페이스는 생성 시 또는 이후 컬렉션 라이프사이클 동안 컬렉션에 바인딩될 수 있다.

컬렉션은 0개 이상의 링크가 있는 어레이인 속성을 정의해야 한다. 링크의 대상 URI는 다른 수집 또는 다른 리소스를 참조할 수 있다. 참조된 수집 또는 리소스는 해당 링크(로컬 참조)를 포함하는 수집과 동일한 장치에 있거나 다른 장치(원격 참조)에 있을 수 있다. 어레이에서 링크의 컨텍스트 URI는 해당 속성을 포함하는 컬렉션이어야 한다. (암묵적) 컨텍스트 URI는 링크에서 "앵커"의 값이 링크의 새로운 기초가 되는 "앵커" 매개변수의 명시적 사양을 사용하여 재지정할 수 있다.

자원은 둘 이상의 수집에서 참조될 수 있으므로 고유한 부모-자녀 관계가 보장되지 않는다. 컬렉션과 컬렉션에서 참조되는 리소스 사이에는 사전 정의된 관계가 없다. 즉, 애플리케이션은 컬렉션을 사용하여 관계를 나타낼 수 있지만, 자동으로 암시되거나 정의되는 것은 없다. 컬렉션과 참조된 리소스의 라이프사이클도 서로 독립적이다.

컬렉션에 대해 "drel" 속성이 정의된 경우, 관계를 명시적으로 지정하지 않은 모든 링크는 해당 컬렉션의 맥락에서 이 기본 관계를 상속해야 한다. 기본 관계는 Link에서 Collection과 대상 URI 사이의 암묵적 관계를 정의한다.

다음 예제에서 속성 "링크"는 컬렉션의 링크 목록을 나타낸다. "링크" 속성은 그것의 가치로서, 일련의 아이템과 각 아이템이 링크된다.

```

/my/house ← This is IRI/URI of the Resource
{
  "rt": ["my.r.house"], ← This and the next 3 lines are the Properties of the Resource
  "color": "blue",
  "n": "myhouse",
  "links": [
    { ← This and the next 4 lines are the Parameters of a Link
      "href": "/door",
      "rt": ["oic.r.door"],
      "if": ["oic.if.b", "oic.if.ll", "oic.if.baseline"]
    },
    {
      "href": "/door/lock",
      "rt": ["oic.r.lock"],
      "if": ["oic.if.b", "oic.if.baseline"],
      "type": ["application/cbor", "application/exi+xml"]
    },
    {
      "href": "/light",
      "rt": ["oic.r.light"],
      "if": ["oic.if.s", "oic.if.baseline"]
    },
    {
      "href": "/binarySwitch",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "type": ["application/cbor"]
    }
  ]
}

```

컬렉션은 다음과 같을 수 있다.

- 사전 정의된 컬렉션으로, 컬렉션이 사전 정의되었으며, 컬렉션은 수명 주기 동안 정적이어야 한다. 이러한 수집은 예를 들어 고정 기능을 나타내는 다른 장치 또는 고정 리소스 세트에 구성된 어플라이언스를 모델링하는 데 사용될 수 있다.
- 수집을 호스팅하는 장치에서만 수집이 사용되는 장치 로컬 컬렉션. 이러한 수집은 많은 서버를 하나로 지칭할 때 클라이언트의 단서로 사용될 수 있다.
- 수집이 장치에서 호스팅되지만 다른 장치가 수집에 액세스하거나 업데이트할 수 있는 중앙 집중식 수집.
- 수집이 중앙 집중화되지만 공인 에이전트 또는 당사자가 관리하는 호스팅된 컬렉션

수집은 링크 배열인 속성을 정의해야 한다(속성 이름 "링크" 권장). 또한 리소스 유형별 수집에 대해 다른 속성을 정의할 수 있다. 컬렉션에 대한 필수 및 권장 공통 특성은 다음 표에 제시되어 있다.

표 10 - Common Properties for Collections

Property	Description	Property Name	Value Type	Mandatory
<b>Links</b>	The array of Links in the Collection	Per Resource Type definition	json Array of Links	Yes
<b>Resource Types</b>	The list of allowed Resource Types for Links in the Collection. If this Property is not defined or is null string then any Resource Type is permitted	As defined in Table 11	As defined in Table 11	No
<b>Mandatory Resource Types</b>	The list of Resource Types for Links that are mandatory in the Collection.	As defined in Table 12	As defined in Table 12	No

기본 리소스 유형인 "oic.wk.col"을 수집에 사용할 수 있다. 이 리소스 유형은 컬렉션에 다른 유형이 정의되지 않았거나 컬렉션 생성 시 리소스 유형이 지정되지 않은 경우에만 사용해야 한다. 기본 리소스 유형은 속성 이름 "링크"를 가진 링크 배열을 포함한 공통 속성을 지원한다.

컬렉션의 모든 인스턴스는 기준("oic.if.baseline") OCF 인터페이스 외에 링크 목록("oic.if.baseline") OCF 인터페이스를 지원해야 한다. 컬렉션의 인스턴스는 선택적으로 이 문서 내에 정의된 추가 OCF 인터페이스를 지원할 수 있다. 수집에 대한 기본 OCF 인터페이스는 리소스 유형 정의에 의해 달리 지정되지 않는 한 링크 목록("oic.if.ll")이 되어야 한다.

특정 사용 사례에서는 여러 리소스의 속성이 그룹으로서만 액세스할 수 있어야 하며 클라이언트에 의한 각 리소스의 속성에 대한 개별 액세스가 금지되어야 한다. 원자성 측정 자원 유형은 이 요건을 충족하기 위해 정의된다. 이것은 Batch OCF 인터페이스를 사용하여 수행된다.

원자성 측정은 링크 배열인 속성을 정의해야 한다(속성 이름 "링크"가 권장된다). 또한 자원 유형에 의한 원자성 측정을 위해 다른 속성을 정의할 수 있다. 원자성 측정에 대한 필수 및 권장 공통 특성은 다음 표에 제시되어 있다.

표 11 - Common Properties for Atomic Measurement

Property	Description	Property Name	Value Type	Mandatory
<b>Links</b>	The array of Links in the Atomic Measurement	Per Resource Type definition	json Array of Links	Yes
<b>Resource Types</b>	The list of allowed Resource Types for Links in the Atomic Measurement. If this Property is not defined or is null string then any Resource Type is permitted	As defined in Table 11	As defined in Table 11	No
<b>Mandatory Resource Types</b>	The list of Resource Types for Links that are mandatory in the Atomic Measurement.	As defined in Table 12	As defined in Table 12	No

원자 측정의 규범적 행동은 다음과 같다.

- Batch OCF 인터페이스의 동작("oic.if.b") 원자력 측정의 정의는 다음과 같다.
- Batch OCF Interface의 경우 UPDATE 작동이 허용되지 않는다. UPDATE 작동이 수신되면 오류 코드가 허용되지 않는 방법이 된다.
- 오류 응답에는 연결된 리소스(즉, 연결된 모든 리소스에 대한 빈 응답)가 포함되지 않아야 한다.
- 원자성 측정(즉, 원자성 측정에서 링크의 목표 자원) 내의 모든 링크된 자원에는 다음 조건이 적용된다.
- 원자성 측정 및 원자성 측정 자체의 연계된 자원은 단일 서버에 존재해야 한다.
- 링크된 자원에 대해 CRUDN 작동이 허용되지 않아야 하며, 금지된 오류 코드가 발생되어야 한다.

- 링크된 자원은 "oic.if.ll" OCF 인터페이스를 노출해서는 안 된다. CRUDN 작업은 연결된 리소스에서 허용되지 않으므로, "oic.if.ll" OCF 인터페이스는 결코 접근할 수 없을 것이다.
- 원자성 측정에서 연결된 자원에 대한 링크는 원자성 측정의 "oic.if.ll" 또는 "oic.if.baseline" OCF 인터페이스를 통해서만 접근할 수 있어야 한다.
- 링크된 자원은 "/oic/res"에 나열되지 않아야 한다.
- 원자성 측정의 링크된 자원은 기본 OCF 인터페이스로 "oic.if.a, "oic.if.s, "oic.if.r", 또는 "oic.if.rw" 중 하나를 정의해야 한다.
- 원자력 측정에서 모든 연계된 자원이 관찰 가능해야 하는 것은 아니다. "oic.if"를 사용하여 원자 측정이 관찰되는 경우.b" OCF 인터페이스, 관측 가능한 것으로 표시되지 않은 링크된 자원이 업데이트되거나 상태가 변경될 경우 통지 응답이 생성되지 않아야 한다.
- "oic.if.b"를 사용할 경우 모든 RETRIEVE 및 관찰 반응에 원자성 측정의 모든 링크된 자원이 포함되어야 한다. OCF 인터페이스.
- 원자성 측정은 "oic.if.b"를 지원해야 한다. 및 "oic.if.ll" OCF 인터페이스.
- 원자성 측정에서 링크된 자원의 필터링은 허용되지 않는다. 원자성 측정에 대한 요청에서 하나 이상의 개별 링크된 자원을 선택하는 쿼리 매개변수는 "금지된" 오류 코드를 발생시킨다.
- "rel" 링크 매개변수가 원자성 측정에 포함된 링크에 포함된 경우, "호스트" 또는 "항목" 값을 가져야 한다.
- 원자성 측정의 기본 OCF 인터페이스는 "oic.if.b."

원자성 측정 자원 유형에 대한 접근 권한은 ISO/IEC 30118-2:2018의 조항 12.2.7.2(원자성 측정 자원 유형에 대한 배치 요청에 대한 ACL 고려사항)에 명시된 것과 같다. 리소스 유형은 다음 표에 정의된 "oic.wk.atomic measurement"로 정의된다.

표 12 - Atomic Measurement Resource Type

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction	M/CR/O
none	Atomic Measurement	"oic.wk.atomic measurement"	"oic.if.ll" "oic.if.baseline" "oic.if.b"	A specialisation of the Collection pattern to ensure atomic RETRIEVAL of its referred Resources	RETRIEVE, NOTIFY	O

원자 측정을 위한 특성은 다음 표에 정의되어 있다.

표 13 - Properties for Atomic Measurement

Property	Description	Property name	Value Type	Mandatory
Links	The set of links that point to the linked Resources	Per Resource Type definition	json Array of Links	Yes

3rd Party는 OCF 장치 내에서 추가 (비OCF) 자원을 지정할 수 있다. 또한 3rd Party는 기존 OCF 정의 리소스 유형 내에서 추가 속성을 지정할 수 있다. 추가 3rd Party는 OCF 정의 열거값을 확장할 수 있다. 3rd Party 정의 Device Type은 3rd Party와 OCF 정의 리소스 유형을 모두 노출할 수 있다. 3rd Party 정의 장치 유형은 이 문서에 정의된 모든 OCF 장치에 대한 필수 리소스를 노출해야 한다.

3rd Party 정의 리소스 유형에는 이 문서에 정의된 모든 필수 속성과 수직 지정 필수 속성이 포함되어야 한다. 본 문서에 정의된 대로 일반 속성이 아닌 OCF 네임스페이스의 일부인 파티 정의 리소스 유형은 다음 표의 3rd Party 정의 속성 규칙을 따른다. 표에서 "Domain\_Name"이라는 용어는 새로운 요소를 정의하는 3rd Party가 소유한 도메인 이름을 의미한다.

표 14 - 3rd party defined Resource elements

	Resource Element	Vendor Definition Rules
New 3 <sup>rd</sup> party defined Device Type	"rt" Property Value of "/oic/d"	"x.<Domain_Name>.<Resource identification>"
New 3 <sup>rd</sup> party defined Resource Type	"rt" Property Value	"x.<Domain_Name>.<Resource identification>"
New 3 <sup>rd</sup> party defined Property within the OCF namespace	Property Name	"x.<Domain_Name>.<Property>"
Additional 3 <sup>rd</sup> party defined values in an OCF specified enumeration	Enumeration Property Value	"x.<Domain_Name>.<enum value>"
Additional 3 <sup>rd</sup> party defined Parameter in an OCF specified Property	Parameter key word	x.<Domain_Name>.<parameter keyword>

이 구성표에서 Domain\_Name의 사용과 관련하여 라벨은 DNS 또는 기타 해상도 메커니즘에 나타나는 방식과 반대로 표시된다. 3rd Party 정의 Resource Type은 IANA 제약 RESTful Environments(CoRE) 매개변수 레지스트리에 등록해야 한다.

**For example:**

```
x.com.samsung.galaxyphone.accelerator
x.com.cisco.ciscorouterport
x.com.hp.printerhead
x.org.allseen.newinterface.newproperty
```

Query Parameters는 특성 및 매개변수(링크 부분 포함)는 특정 자원의 선택을 위한 하나의 기준으로 URI의 쿼리 부분에서 사용될 수 있다. 이것은 속성(즉, <재산명> = <필수재산가치>)을 질의 부분 중 하나로 선언함으로써 이루어진다. 쿼리 필터에는 ASCII 문자열만 허용되고, 쿼리 필터에는 NULL 문자가 허용되지 않는다. 즉, ASCII 문자가 있는 속성 값만 조회 필터에 일치시킬 수 있다. 조회의 모든 선언된 속성 또는 링크 매개 변수가 대상의 해당 속성 또는 링크 매개 변수와 일치할 때 리소스가 선택된다.

쿼리가 여러 개의 개별 쿼리 매개변수를 포함하는 경우, 6.2.2에서 설명한 "&"에 의해 구분된다. 클라이언트는 "?ins=11111&rt=oic.r.switch.binary"와 같이 여러 개의 개별 쿼리 매개변수를 적용할 수 있다. 이러한 쿼리가 서버에서 지원되는 경우, 다른 쿼리 매개변수 유형("rt", "ins", "if" 등)을 쿼리 대상에 "모든"으로 일치시켜 이 작업을 수행해야 한다. 이 예에서 이는 "ins"가 "11111"로 채워진 oic.r.switch.binary의 인스턴스로 해결된다. 조회 파라미터의 순서에 유의미한 것은 없다. 클라이언트는 반복 쿼리 매개변수를 사용하여 둘 이상의 리소스 유형을 선택할 수 있다(예: "?rt=oic.r.switch.binary&rt=oic.r.r.r.ramptime". 서버에서 이러한 쿼리를 지원하는 경우, 쿼리의 대상에 대해 "임의" 반복 쿼리 매개 변수를 일치시켜 이 작업을 수행해야 한다. 이 예에서 존재하는 "oic.r.switch.binary" 및/또는 "oic.r.r.r.r.ramptime"의 인스턴스를 선택한다.

클라이언트는 "?if=oic.if"와 같이 단일 쿼리에서 다중 반복 매개변수와 다중 개별 매개변수를 결합할 수 있다.b&ins=1111111&rt=oic.r.switch.binary&rt=oic.r.r.r.ramptime." 서버에서 이러한 쿼리를 지원하는 경우, 반복된 쿼리 매개변수를 "임의"로 일치시킨 다음 다른 쿼리 매개변수 유형을 "모두"로 일치시켜 이 작업을 수행해야 한다. 예에서 "ins"가 11111인 "oic.r.switch.binary" 및/또는 "oic.r.r.r.r.ramptime"의 인스턴스는 일괄 응답에서 선택된다. 참고로 쿼리 문자열 내의 매개변수는 실제 메시징 프로토콜 내에 표시된다.

기본 OCF 인터페이스인 "value", "oic.if.s" 또는 "oic.if.baseline"은 일반 "rt" 쿼리의 확장이다. 서버가 "rt" 쿼리와 함께 다중 값 "rt" 리소스에 대한 RETRIEVE 요청을 수신하는 경우(예: GET /Resample?rt=oic.r.foo), 서버는 쿼리 값이 대상 리소스의 "rt" 속성 값의 항목인 경우에만 응답해야 하며 쿼리 값과 연관된 속성만 다시 전송해야 한다. 예를 들어, "rt"가 있는 리소스를 대상으로 GET /ResExample?rt=oic.r.switch.binary, "oic.r.switch.binary", "oic.r.light.brightness"가 있는 리소스를 대상으로 GET /R.switch.binary의 속성만 수신하면 서버는 응답한다.

요청에 대해 OCF 인터페이스를 선택할 경우 요청 메시지의 리소스 URI에서 조회 매개 변수로 지정해야 한다. 쿼리 매개변수가 지정되지 않은 경우 기본 OCF 인터페이스를 사용해야 한다. 선택한 OCF 인터페이스가 리소스에서 허용되는 OCF 인터페이스 중 하나가 아닌 경우, OCF 인터페이스를 선택하면 오류가 발생하며 서버는 오류 응답 코드로 응답해야 한다.

예를 들어, 기준선 OCF 인터페이스는 대상 리소스의 URI에 있는 쿼리 매개변수 목록에

"if=oic.if.baseline"을 추가하여 선택할 수 있다. 예: "GET /oic/res?if=oic.if.baseline".

배치 OCF 인터페이스 자체에 대한 자세한 내용은 5.5.6.3.4를 참조한다. 조회 또는 업데이트를 위해 수집에서 특정 리소스를 선택하기 위해 배치 OCF 인터페이스와 함께 쿼리 매개 변수를 사용할 수 있다.

이러한 매개변수는 링크 매개변수 값을 매칭하여 컬렉션의 항목을 선택하는 데 사용된다.

배치 OCF 인터페이스를 사용하여 적용된 RETRIEVE 작업에 링크 선택 쿼리 매개 변수를 사용할 경우, 링크 매개변수와 일치하는 수집의 리소스만 반환해야 한다.

배치 OCF 인터페이스를 사용하여 UPDATE 작업이 적용된 상태에서 링크 선택 쿼리 매개 변수를 사용할 경우 링크 매개 변수와 일치하는 리소스만 업데이트해야 한다.

CRUDN(CREATE, RETRIEVE, UPDATE, DELETE, NOTIFY)은 자원을 조작하기 위해 정의된 작업이다. 이러한 작업은 n 서버에 포함된 리소스에 대한 클라이언트에 의해 수행된다.

유효한 CRUDN 작업을 수신할 때 요청의 대상이 되는 리소스를 호스팅하는 서버는 요청에 포함된 OCF 인터페이스에 따라 또는 OCF 인터페이스가 포함되지 않은 경우 리소스 유형에 대한 기본 OCF 인터페이스에 따라 응답을 생성해야 한다.

CRUDN 운영은 메시지에 전달되고 다음 표에 정의된 일련의 매개변수를 이용한다. 장치는 CRUDN 운영 및 운영 응답에 포함된 리소스 표현을 위한 기본 페이로드(컨텐츠) 인코딩 체계로 CBOR를 사용해야 한다. 장치는 다른 페이로드 인코딩 체계를 협상할 수 있다.

표 15 - Parameters of CRUDN messages

Applicability	Name	Denotation	Definition
All messages	<i>fr</i>	From	The URI of the message originator.
	<i>to</i>	To	The URI of the recipient of the message.
	<i>ri</i>	Request Identifier	The identifier that uniquely identifies the message in the originator and the recipient.
	<i>cn</i>	Content	Information specific to the operation.
Requests	<i>op</i>	Operation	Specific operation requested to be performed by the Server.
	<i>obs</i>	Observe	Indicator for an Observe request.
Responses	<i>rs</i>	Response Code	Indicator of the result of the request; whether it was accepted and what the conclusion of the operation was. The values of the response code for CRUDN operations shall conform to those as defined in clause 5.9 and 12.1.2 in IETF RFC 7252.
	<i>obs</i>	Observe	Indicator for an Observe response.

CREATE 작업은 서버에서 새 리소스 생성을 요청하는 데 사용된다. CREATE 작동은 클라이언트에 의해 시작되며 다음 그림과 같이 세 단계로 구성된다.

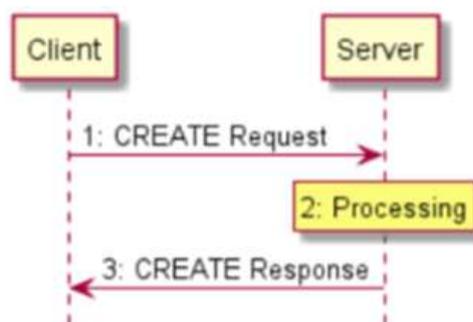


그림 10 - CREATE operation

CREATE 요청 메시지는 클라이언트가 서버에 전송하여 서버에 의해 새 리소스를 생성한다. CREATE 요청 메시지는 다음 매개 변수를 전달한다.

- fr: 클라이언트의 고유 식별자 - ~: 새 리소스 생성을 담당하는 대상 리소스의 URI.
- ri: CREATE 요청 식별자
- cn: 서버에서 생성할 리소스 정보.
- cn에는 생성할 리소스의 URI 및 리소스 유형 속성이 포함된다.
- cn은 생성할 리소스의 추가 속성을 포함할 수 있다.
- op: CREATE

CREATE 요청을 받은 후 서버는 클라이언트가 요청된 리소스를 생성할 적절한 권한을 가지고 있는지 확인할 수 있다. 유효성 검사가 성공하면 서버는 요청된 리소스를 생성한다. 서버는 CREATE 응답 메시지에 포함시키기 위해 CREATE 요청에 ri 매개 변수의 값을 캐시한다.

서버는 클라이언트의 CREATE 요청 메시지에 응답하여 CREATE 응답 메시지를 전송해야 한다. CREATE 응답 메시지에는 다음 매개 변수가 포함된다.

- fr: 서버의 고유 식별자
- to: 클라이언트의 고유 식별자
- ri: CREATE 요청에 포함된 식별자
- cn: 서버에서 생성한 리소스의 정보
- cn에는 생성된 리소스의 URI가 포함된다.
- cn에는 생성된 리소스의 리소스 표시가 포함된다.
- rs: CREATE 작업의 결과

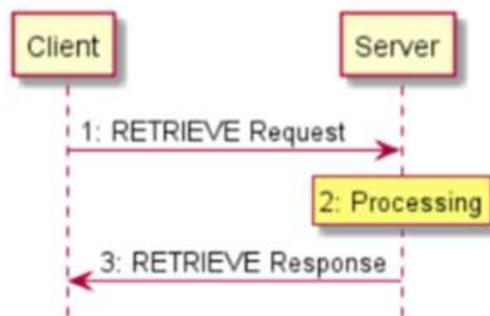


그림 11 - RETRIEVE operation

RETRIEVE 작업은 리소스의 현재 상태 또는 표현을 요청하는 데 사용된다. RETRIEVE 작동은 클라이언트에 의해 시작되며 그림과 같이 세 단계로 구성된다.

RETRIEVE 요청 메시지는 서버에서 리소스 표시를 요청하기 위해 클라이언트에 의해 서버로 전송된다. RETRIEVE 요청 메시지는 다음 매개변수를 전달한다.

- fr: 클라이언트의 고유 식별자
- to: 클라이언트가 목표로 하는 리소스의 URI
- ri: RETRIEVE 요청 식별자
- op: RETRIEVE

RETRIEVE 요청을 받은 후, 서버는 클라이언트가 요청된 데이터를 검색할 수 있는 적절한 권한을 가지고 있고 속성을 읽을 수 있는지 확인할 수 있다. 서버가 응답에 사용할 RETRIEVE 요청에서 ri 매개 변수의 값을 캐시한다.

서버는 클라이언트의 RETRIEVE 요청 메시지에 응답하여 RETRIEVE 응답 메시지를 전송해야 한다. RETRIEVE 응답 메시지에는 다음 매개 변수가 포함된다.

- fr: 서버의 고유 식별자.
- to: 클라이언트의 고유 식별자
- ri: RETRIEVE 요청에 포함된 식별자
- cn: 클라이언트가 요청한 리소스 정보

- cn은 RETRIEVE 요청에 대상인 리소스의 URI를 포함해야 한다.
- rs: RETRIEVE 작동 결과

UPDATE 작업은 부분적으로 UPDATE이거나 OCF 인터페이스와 함께 리소스 내 정보를 완전히 대체하는 것으로, 작업에도 적용된다. UPDATE 작동은 클라이언트에 의해 시작되며 다음 그림과 같이 세 단계로 구성된다.

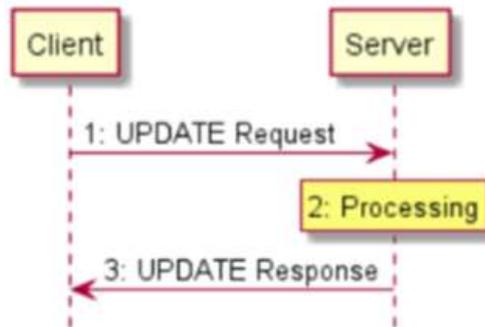


그림 12 - UPDATE operation

UPDATE 요청 메시지는 클라이언트가 서버에 전송하여 서버의 리소스 정보 업데이트를 요청한다. UPDATE 요청 메시지는 다음 매개 변수를 전달한다.

- fr: 클라이언트의 고유 식별자
- to: 정보 업데이트를 대상으로 하는 리소스의 URI
- ri: UPDATE 요청 식별자
- op: 업데이트
- cn: 대상 리소스에서 업데이트할 리소스의 속성을 포함한 정보

UPDATE 요청을 받은 후 서버는 클라이언트가 요청된 데이터를 업데이트할 적절한 권한을 가지고 있는지 확인할 수 있다. 유효성 검사가 성공하면 서버는 UPDATE 요청 메시지의 cn 매개 변수에 포함된 정보에 따라 대상 리소스 정보를 업데이트한다. 서버는 응답에 사용하기 위해 UPDATE 요청에 ri 매개 변수의 값을 캐시한다.

읽기 전용 속성을 포함하는 UPDATE 요청은 잘못된 요청을 나타내는 rs가 있는 서버에서 거부해야 한다. UPDATE 요청은 작업을 지원하는 적용된 OCF 인터페이스를 통해 볼 수 있는 대상 리소스의 속성에만 적용되어야 한다. 존재하지 않는 속성의 업데이트는 무시된다.

업데이트 요청은 대상 리소스의 속성 값이 대상 리소스에 현재 노출되는 값과 동일하더라도 적용해야 한다.

서버는 클라이언트의 관찰 요청에서 확인된 리소스 상태를 모니터링해야 한다.

관찰된 리소스 상태 또는 리소스에 적용된 업데이트 작업의 상태가 변경될 때마다 서버는 관찰 표시와 함께 다른 RETRIEVE 응답을 전송한다. 이 메커니즘은 클라이언트가 통지를 트리거하는 어떤 한계나 한계를 지정하는 것을 허용하지 않으며, 결정은 전적으로 서버에 맡겨진다.

서버는 클라이언트가 요청한 리소스 상태의 관찰된 변경 후 업데이트된 RETRIEVE 응답 메시지를 전송해야 한다. RETRIEVE 응답 메시지는 매개변수를 포함해야 한다.

UPDATE 응답 메시지에는 다음 매개 변수가 포함된다.

- fr: 서버의 고유 식별자
- to: 클라이언트의 고유 식별자
- ri: UPDATE 요청에 포함된 식별자
- rs: UPDATE 요청 결과

UPDATE 응답 메시지에는 다음 매개 변수가 포함될 수 있다.

- cn: UPDATE 요청 처리 후의 리소스 표시

DELETE 작업은 리소스 제거를 요청하는 데 사용된다. DELETE 작동은 클라이언트에 의해 시작되며 다

음 그림과 같이 세 단계로 구성된다.

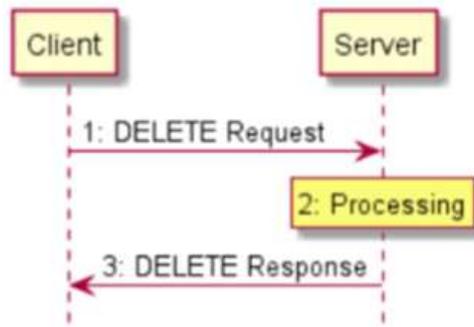


그림 13 - DELETE operation

클라이언트가 서버의 리소스를 삭제하기 위해 서버에 DELETE 요청 메시지를 전송한다. DELETE 요청 메시지는 다음 매개 변수를 전달한다.

- fr: 클라이언트의 고유 식별자
- to: 삭제 대상인 대상 리소스의 URI
- ri: DELETE 요청 식별자
- op: DELETE

DELETE 요청을 수신한 후 서버는 클라이언트가 식별된 리소스를 삭제할 수 있는 적절한 권한이 있는지, 그리고 식별된 리소스가 있는지 여부를 확인할 수 있다. 유효성 검사가 성공하면 서버는 요청된 리소스를 제거하고 모든 관련 정보를 삭제한다. 서버는 응답에 사용하기 위해 DELETE 요청에서 ri 매개 변수의 값을 캐시한다.

서버는 클라이언트에서 보낸 DELETE 요청 메시지에 응답하여 DELETE 응답 메시지를 전송해야 한다. DELETE 응답 메시지는 다음 매개 변수가 포함된다.

- fr: 서버의 고유 식별자
- to: 클라이언트의 고유 식별자
- ri: DELETE 요청에 포함된 식별자
- rs: DELETE 작업의 결과

NOTIFY 작업은 상태 변경에 대한 비동기적 통지를 요청하기 위해 사용된다. NOTIFY 작업은 여기에 정의된 NOTIFICATION 응답 메시지를 사용한다.

알림 응답 메시지는 클라이언트가 식별한 URL에 상태 변경을 알리기 위해 서버에 의해 전송된다. 통지 응답 메시지는 다음과 같은 매개변수를 전달한다.

- fr: 서버의 고유 식별자
- to: 알림 메시지의 리소스 대상 URI
- ri: CREATE 요청에 포함된 식별자
- op: 알림
- cn: 리소스의 업데이트된 상태

사물 인터넷은 광범위한 장치 및 네트워크 기능을 통해 물리적 세계를 감지하고 작동시키는 광범위한 애플리케이션으로 구성되어 있다. 연결 옵션은 IP 계층에 적응하는 것이 바람직하다. 이를 위해 IETF는 Bluetooth®, Wi-Fi, 802.15.4, LPWAN 등을 IPv6에 적응시키는 상당한 작업을 완료했다. 이러한 적응과 더 큰 주소 공간 및 향상된 주소 관리 기능은 IPv6을 OCF 네트워크 계층 기술에 대한 분명한 선택으로 만든다.

노후한 IPv4 중심 네트워크는 복잡한 토폴로지를 지원하도록 진화했지만, 그것의 구축은 주로 단일 네트워크로서 ISP(인터넷 서비스 제공자)에 의해 프로비저닝되었다. 주거용 주택에서 흔히 볼 수 있는 보다 복잡한 네트워크 토폴로지는 대부분 NAT(Private Network Address Translation)과 같은 기술에 의존하는 추가 홈 네트워크 장치를 취득함으로써 소개된다. 이러한 기술은 올바르게 설치하기 위해 전문가의 도움이 필요하며, 홈 네트워크에서는 대개 라우팅, 명명 및 검색 서비스와 같은 구조가 파괴되기 때문에 피해야 한다.

다중 세그먼트 생태계 OCF 주소는 새로운 장치와 관련 라우터의 확산을 야기할 뿐만 아니라 추가 에지 라우터를 도입하는 새로운 서비스도 야기할 것이다. 이러한 모든 새로운 요구사항은 다음 그림과 같은 복잡한 네트워크 토폴로지를 다루기 위한 고급 아키텍처 구조를 필요로 한다.

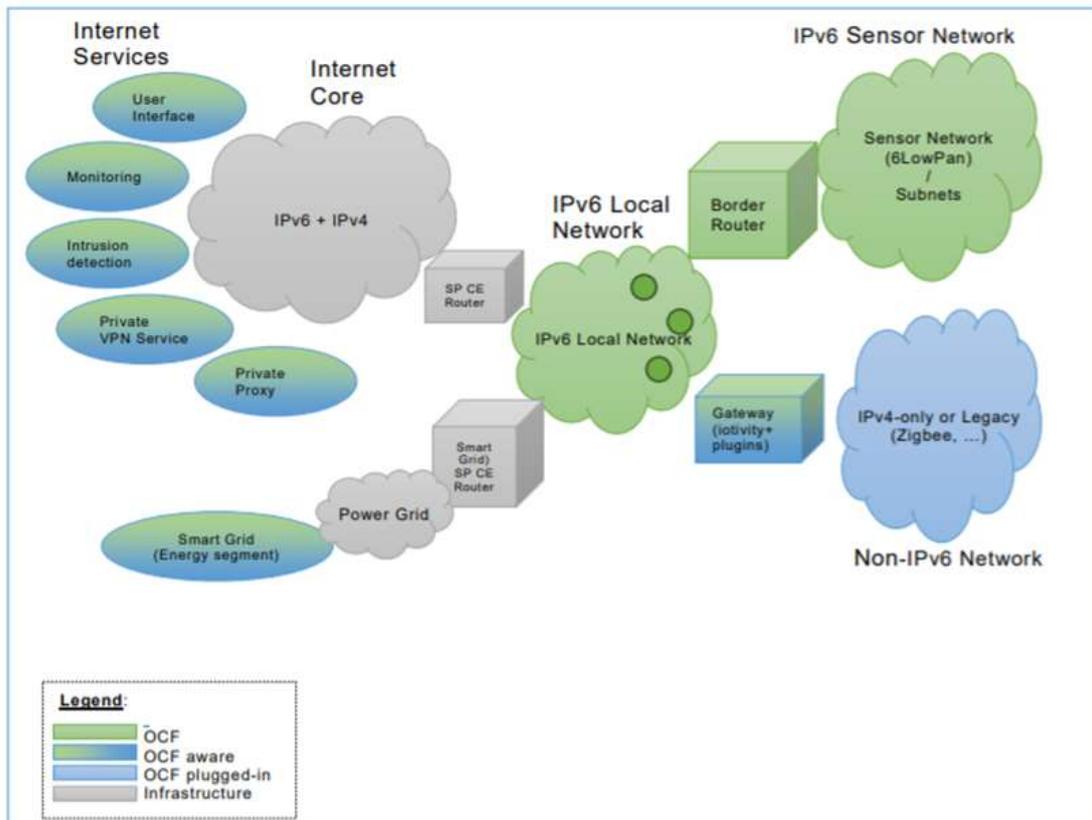


그림 14 - High Level Network & Connectivity Architecture

IETF RFC 6434의 관점에서 IPv6 노드는 라우터 또는 호스트 역할을 가정한다. 노드는 더 늘어날 수 있다.

- 라우터는 IETF RFC 7084에 정의된 고객 에지 라우터 기능을 구현할 수 있다.
- 처리 전력, 메모리, 비휘발성 저장 또는 전송 용량 제한 : 특수 IP 적응 계층(6LoWPAN) 및 전용 라우팅 프로토콜(RPL)이 필요하다. 예를 들어 IEEE 802.14.5, ITU와 같은 저전력 물리적 계층을 통해 전송되는 장치가 포함된다. G9959, Bluetooth Low Energy, DECT Ultra Low Energy 및 NFC(Near Field Communication, NFC).
- 노드는 IPv6와 비 IPv6 네트워크간에 메시지를 변환하고 라우팅 할 수 있다.

OCF는 네트워크 계층에서 IPv6을 권장한다. IPv6 권장 사항의 이유로는 다음과 같다.

- 더 큰 주소 공간 부작용: NAT의 요구사항을 크게 줄인다.
- 보다 유연한 아키텍처 지원 인터페이스당 여러 주소 및 유형: 링크 로컬, ULA, GUA, 다양한 범위의 멀티캐스트 주소, 다중 홈 네트워크를 지원하는 기능향상, 번호변경기능 향상
- 더 강력한 자동 구성 기능: DHCPv6, SLAAC, 라우터 검색 등
- 제한된 노드에서 IP 연결을 가능하게 하는 기술은 IPv6을 기반으로 한다.
- 모든 주요 소비자 운영 체제(IoS, Android, Windows, Linux)는 이미 IPv6을 사용하도록 설정됨
- 전 세계의 주요 서비스 공급업체가 IPv6을 구축하고 있음

IPv6 노드 요구 사항으로는 네트워크 계층 서비스가 노드 간 상호운용성을 보장하려면 모든 노드에서 공통 네트워크 계층을 의무화하는 것이 필수적이다. 프로토콜은 네트워크가 안전하고, 관리 가능하고, 확장 가능하며, 제약되고, 자체 조직적인 중첩 노드를 포함할 수 있도록 해야 한다. OCF는 IPv6을 공통 네트워크 계층 프로토콜로서 의무화하여 모든 디바이스의 상호운용성을 보장한다.

더 많은 기능이 있는 장치에는 다중 스택 장치를 생성하는 추가 프로토콜도 포함될 수 있다. 이 조항의 나머지 부분은 IPv6 호스트, IPv6 제약 호스트 및 IPv6 라우터에 대한 상호운용성 요건에 초점을 맞출 것이다. 다중 스택 게이트웨이 장치에 포함된 다양한 프로토콜 변환 순서는 이 문서의 후속 부록에 수록된 주소일 수 있다.

IPv6 노드는 IPv6을 지원해야 하며 IETF RFC 6434에 명시된 요구 사항을 준수해야 한다.

#### (자) OCF Endpoint

OCF Endpoint의 특정 정의는 사용 중인 Transport Protocol Suite에 따라 달라진다. IPv6 기반 UDP 기반 CoAP의 예에서 OCF Endpoint는 IPv6 주소와 UDP 포트 번호로 식별된다. 각 장치는 요청 및 응답 메시지를 교환할 수 있는 하나 이상의 OCF Endpoint와 연결되어야 한다. 메시지가 OCF Endpoint로 전송될 때 OCF Endpoint와 연결된 장치로 전달되어야 한다. 요청 메시지가 OCF Endpoint로 전달되면 경로 구성 요소는 대상 리소스를 찾기에 충분하다.

장치는 여러 개의 OCF Endpoint와 연결될 수 있다. 예를 들어, 장치는 여러 개의 IP 주소 또는 포트 번호를 가지거나 CoAP와 HTTP 전송 프로토콜을 모두 지원할 수 있다. 장치의 다른 리소스는 동일한 OCF Endpoint로 액세스하거나 다른 리소스를 필요로 할 수 있다. 일부 리소스는 하나의 OCF Endpoint를 사용하고 다른 리소스는 다른 OCF Endpoint를 사용할 수 있다. 그것은 구현에 달려있다.

반면에 OCF Endpoint는 요청 URI로 대상 리소스를 명확하게 지정할 수 있는 방법이 있는 경우에만 여러 장치 간에 공유할 수 있다. 예를 들어 여러 CoAP 서버가 모든 호스팅된 리소스에 대해 고유하게 다른 URI 경로를 사용하고 CoAP 구현 디플렉스들이 경로별로 사용되는 경우, 동일한 CoAP OCF Endpoint를 공유할 수 있다. 그러나 일부 필수 자원(예: "oic.wk.d")에는 미리 결정된 URI(예: "/oic/d")가 필수적이기 때문에 이 문서 버전에서는 이 작업이 불가능하다.

OCF Endpoint는 키 값 쌍 "ep"과 "pri"의 두 항목으로 구성된 OCF Endpoint 정보로 표현된다. "ep"은 다음과 같이 지정된 Transport Protocol Suite 및 OCF Endpoint Locator를 나타낸다.

- Transport Protocol Suite - 요청 및 응답 메시지를 RESTful 트랜잭션(예: CRUDN)으로 교환할 수 있는 프로토콜(예: CoAP + UDP + IPv6)의 조합. Transport Protocol Suite는 URI 스키마 이름으로 표시되어야 한다. 이 문서에 의해 지원되는 모든 구성표 이름은 IANA 등록되어 있으며, 표에 수록되어 있다. 공급업체는 또한 비 IANA 등록 체계 이름을 자신의 용도로 사용할 수 있다(예: ".com.example.foo"), 이는 IETF RFC 7595에 의해 정의된 체계 이름의 구문을 따라야 한다. 벤더가 정의한 구성표 이름의 동작은 이 문서에 의해 정의되지 않는다. OCF Endpoint 정보를 "ep"에 노출할 때 모든 OCF 정의 리소스 유형은 표에 정의된 전송 프로토콜 제품군과 함께 적어도 하나의 "ep"을 포함해야 한다.
- OCF Endpoint Locator - IP 주소로 확인 가능한 주소(예: IPv6 주소 + 포트 번호) 또는 간접 식별자(예: DNS 이름)로 확인 가능. 이 IP 주소는 메시지를 OCF Endpoint 및 관련 장치로 보낼 수 있다. "coap" 및 "coaps"에 대한 OCF Endpoint 로케이터는 "IP 주소: 포트 번호"로 지정해야 한다. "coap+tcp" 또는 "coaps+tcp"에 대한 OCF Endpoint 로케이터는 DNS 이름이 이름 확인 서비스(즉, DNS)를 사용하여 대상 리소스에 대한 유효한 IP 주소로 확인되도록 "IP 주소: 포트 번호" 또는 "DNS 이름"으로 지정해야 한다. 세 번째의 경우, 포트 번호가 누락된 경우, IETF RFC 8323에 정의된 "coap+tcp"(및 "coaps+tcp") 체계에 대해 각각 기본 포트 "5683"(및 "5684")을 가정해야 한다. OCF Endpoint 로케이터는 수신 세션을 수신하기 위한 것이며, 임시 주소는 발신 세션을 시작하기 위한 것이기 때문에 임시 주소를 사용하면 안 된다(IETF RFC 4941) 또한 "/oic/res"에 포함시키면 사생활 문제가 발생할 수 있다(IETF RFC 7721).

"ep"은 Transport Protocol Suite를 나타내는 체계 구성요소 및 OCF Endpoint Locator를 나타내는 권한 구성요소와 함께 URI(IETF RFC 3986) 값을 가져야 한다.

"coap"과 "coaps"의 "ep" 예는 다음과 같다.

---

```
"ep": "coap://[fe80::b1d6]:1111"
```

---

"coap+tcp" 및 "coaps+tcp"의 "ep" 예는 다음과 같다.

---

```
"ep": "coap+tcp://[2001:db8:a::123]:2222"  
"ep": "coap+tcp://foo.bar.com:2222"  
"ep": "coap+tcp://foo.bar.com"
```

---

해당하는 Transport Protocol Suite가 포함된 "ep"의 현재 목록은 다음 표에 나와 있다.



필요로 하며, "eps" 매개변수를 사용하여 필요한 정보(예:)를 표시해야 한다.

OCF 1.0 페이로드의 포트 번호). OIC 1.1과의 선택적 역호환성을 위해 "sec"와 "포트"는 OIC 1.1 페이로드에서만 사용해야 한다.

OCF Endpoint 검색은 클라이언트가 장치 또는 리소스에 대한 OCF Endpoint 정보를 획득하는 프로세스로 정의된다.

장치가 CoAP 메시지의 소스(예: "/oic/res" 응답)인 경우 소스 IP 주소와 포트 번호를 결합하여 장치의 OCF Endpoint Locator를 형성할 수 있다. "coap" 구성표 및 기본 "pri" 값과 함께 장치에 대한 OCF Endpoint 정보를 구성할 수 있다.

즉, CoAP가 포함된 "/oic/res" 응답 메시지는 응답 장치의 OCF Endpoint 정보를 암묵적으로 전달하고, 모든 호스팅된 리소스를 전달하며, CoAP의 동일한 전송 프로토콜로 액세스할 수 있다. "eps" 매개변수가 없는 경우, 클라이언트는 암묵적 발견을 활용하여 대상 자원에 접근할 수 있어야 한다.

OCF Endpoint 정보는 링크의 "eps" 매개변수를 "/oic/res"로 명시적으로 표시할 수 있다.

"/oic/res" 응답은 응답 장치가 호스팅하는 일부 리소스에 대한 OCF Endpoint 정보를 암시적으로 나타낼 수 있다. 그러나 암묵적 발견, 즉 CoAP 응답 메시지의 OCF Endpoint 정보에 대한 추론은 동일한 장치의 일부 리소스에 대해 작동하지 않을 수 있다. 예를 들어, 일부 자원은 CoAPS를 통한 보안 액세스를 허용할 수 있으며, 이 경우 포트 번호를 표시하기 위해 "eps" 매개 변수가 필요하다. 또한 "/oic/res"는 다른 장치에 속하는 대상 자원을 노출시킬 수 있다.

링크의 대상 리소스에 대한 OCF Endpoint를 암묵적으로 유추할 수 없는 경우, 클라이언트가 대상 리소스에 액세스할 수 있는 명시적인 OCF Endpoint 정보를 제공하기 위해 "eps" 매개변수를 포함해야 한다.

"eps" 매개변수가 있는 경우 클라이언트는 이를 활용하여 대상 자원에 접근할 수 있어야 한다.

"coap" 및 "coaps"의 경우 클라이언트는 "eps" 매개 변수의 "ep" 값의 IP 주소를 사용하여 대상 리소스에 액세스할 수 있다.

"coap+tcp" 및 "coaps+tcp"의 경우 클라이언트는 "eps" 매개 변수의 IP 주소를 사용하거나 "eps" 매개 변수의 DNS 이름을 확인하여 대상 리소스에 대한 유효한 IP 주소를 얻을 수 있다.

"eps" 매개 변수가 포트 번호를 생략하는 경우, IETF RFC 8323에 정의된 "coap+tcp"(및 "coaps+tcp") 체계에 대해 기본 포트 "5683"(및 "5684")을 가정해야 한다.

링크의 대상 리소스에 액세스하기 위해 클라이언트는 링크의 "eps" 매개 변수를 사용할 수 있으며, 이 매개 변수가 없는 경우 묵시적 검색에 다시 노출될 수 있다.

이는 일반적으로 다른 장치가 호스팅하는 리소스에 대한 링크를 전송하는 리소스 디렉토리 또는 브리지 장치의 "/oic/res"에 적용된다.

이는 2개의 브리징 장치가 있는 브리지 장치에서 "/oic/res" 응답의 예로서 링크에 "eps" 매개 변수가 있다.

```

    {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
  ],
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/sec/doxm",
    "rt": ["oic.r.doxm"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 1},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:55555"},
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/sec/pstat",
    "rt": ["oic.r.pstat"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 1},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/sec/cred",
    "rt": ["oic.r.cred"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 1},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/sec/acl2",
    "rt": ["oic.r.acl2"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 1},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/myIntrospection",
    "rt": ["oic.wk.introspection"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
    "href": "/oic/res",
    "rt": ["oic.wk.res"],
    "if": ["oic.if.ll", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:66666"},
      {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
    ]
  }

```

```

]
},
{
  "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
  "href": "/oic/d",
  "rt": ["oic.wk.d", "oic.d.light", "oic.d.virtual"],
  "if": ["oic.if.r", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [
    {"ep": "coap://[2001:db8:a::b1d4]:66666"},
    {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
  ]
},
{
  "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
  "href": "/oic/p",
  "rt": ["oic.wk.p"],
  "if": ["oic.if.r", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [
    {"ep": "coap://[2001:db8:a::b1d4]:66666"},
    {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
  ]
},
{
  "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
  "href": "/myLight",
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [
    {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
  ]
},
{
  "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
  "href": "/oic/sec/doxm",
  "rt": ["oic.r.doxm"],
  "if": ["oic.if.baseline"],
  "p": {"bm": 1},
  "eps": [
    {"ep": "coap://[2001:db8:a::b1d4]:66666"},
    {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
  ]
},
{
  "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
  "href": "/oic/sec/pstat",
  "rt": ["oic.r.pstat"],
  "if": ["oic.if.baseline"],
  "p": {"bm": 1},
  "eps": [
    {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
  ]
},
{
  "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
  "href": "/oic/sec/cred",
  "rt": ["oic.r.cred"],
  "if": ["oic.if.baseline"],
  "p": {"bm": 1},
  "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
},
}

```

```

{
  "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
  "href": "/oic/sec/acl2",
  "rt": ["oic.r.acl2"],
  "if": ["oic.if.baseline"],
  "p": {"bm": 1},
  "eps": [
    {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
  ]
},
{
  "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
  "href": "/myLightIntrospection",
  "rt": ["oic.wk.introspection"],
  "if": ["oic.if.r", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
},
{
  "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
  "href": "/oic/res",
  "rt": ["oic.wk.res"],
  "if": ["oic.if.ll", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [
    {"ep": "coap://[2001:db8:a::b1d4]:77777"},
    {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
  ]
},
{
  "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
  "href": "/oic/d",
  "rt": ["oic.wk.d", "oic.d.fan", "oic.d.virtual"],
  "if": ["oic.if.r", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [
    {"ep": "coap://[2001:db8:a::b1d4]:77777"},
    {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
  ]
},
{
  "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
  "href": "/oic/p",
  "rt": ["oic.wk.p"],
  "if": ["oic.if.r", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [
    {"ep": "coap://[2001:db8:a::b1d4]:77777"},
    {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
  ]
},
{
  "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
  "href": "/myFan",
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "p": {"bm": 3},
  "eps": [
    {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
  ]
},
{
  "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",

```

```

[
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/res",
    "rel": "self",
    "rt": ["oic.wk.res"],
    "if": ["oic.if.ll", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:55555"},
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/d",
    "rt": ["oic.wk.d", "oic.d.bridge"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:55555"},
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/p",
    "rt": ["oic.wk.p"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:55555"},
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/mySecureMode",
    "rt": ["oic.r.securemode"],
    "if": ["oic.if.rw", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [

```

```

    "href": "/oic/sec/doxm",
    "rt": ["oic.r.doxm"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 1},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:77777"},
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/sec/pstat",
    "rt": ["oic.r.pstat"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 1},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/sec/cred",
    "rt": ["oic.r.cred"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 1},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/sec/acl2",
    "rt": ["oic.r.acl2"],
    "if": ["oic.if.baseline"],
    "p": {"bm": 1},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/myFanIntrospection",
    "rt": ["oic.wk.introspection"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  }
}
]

```

다음은 CoAP 기반 OCF Endpoint 검색을 설명한다.

- 장치는 스코프 2, 3 및 5(즉, ff02:158, ff03:158 및 ff05:158)로 모든 OCF 노드 멀티캐스트 그룹 (IANA IPv6 멀티캐스트 어드레스 스페이스 레지스트리)에 가입해야 하며 포트 5683에서 수신 대기해야 한다. IETF RFC 7252의 준수를 위해, 장치는 추가로 모든 CoAP 노드 멀티캐스트 그룹에 가입할 수 있다.
- 자원을 발견하려는 고객은 a)에서 정의한 멀티캐스트 그룹에 가입해야 한다.
- 기기는 보안되지 않은 OCF Endpoint를 통해 "/oic/res"를 노출해야 한다.
- 클라이언트는 "5683" 포트에서 스코프 2("ff02::158")가 있는 모든 OCF 노드 멀티캐스트 그룹 주소로 검색 요청(GET 요청)을 보내야 한다. 요청된 URI는 "/oic/res"여야 한다. IETF RFC 7252 준수를 위해 클라이언트는 모든 CoAP 노드 멀티캐스트 그룹에 추가로 전송할 수 있다.
- 검색 요청이 특정 리소스 유형에 대해 의도된 경우, 쿼리 매개변수 "rt"를 원하는 리소스 유형으로 설정된 요청에 포함해야 한다. 리소스 유형을 호스팅하는 장치만 검색 요청에 응답해야 한다.
- "rt" 쿼리 매개변수가 생략되면 모든 기기가 검색 요청에 응답해야 한다.
- 멀티캐스트 요청 처리는 IETF RFC 7252의 조항 8과 IETF RFC 6690의 조항 4.1에 설명되어야 한다.
- 요청을 수신하는 기기는 CBOR 페이로드 인코딩을 사용하여 응답해야 한다. 장치는 멀티캐스트 검색을 위한 CBOR 페이로드 인코딩 지원을 표시해야 한다.

바-2) [스마트팜 기자재 보안 표준기술 연구] 스마트팜에서 사용되는 IoT 기자재에 대한 보안 기술 적용방안 연구 ('19)

(가) 개요

여기에서는 OCF 핵심 프레임워크 및 애플리케이션의 보안을 구현하는 데 사용되는 리소스, 프로토콜 및 규약을 정의한다. 문서의 목적상 ISO/IEC 30118-1:2018에 제시된 용어와 정의를 적용한다. 다음 그림은 OCF 장치 사이의 상호작용을 나타낸다.

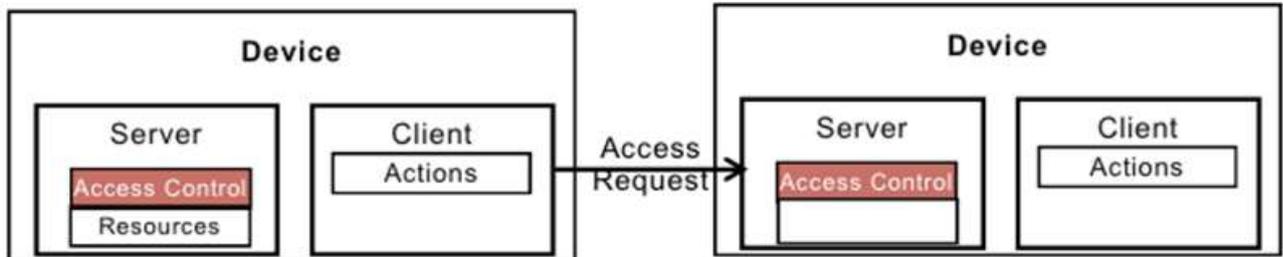


그림 1 - OCF Interaction

장치는 서버에 대한 작업을 수행하는 클라이언트 역할을 구현할 수 있다. 액션은 서버에서 관리하는 리소스에 액세스한다. OCF 스택은 리소스에 대한 액세스 정책을 시행한다. end-to-end 장치 상호작용은 세션 보호 프로토콜(예: DTLS) 또는 데이터 암호화 방법을 사용하여 보호할 수 있다.

OCF 보안 아키텍처의 목표는 리소스 및 리소스 보호를 지원하는 데 사용되는 HW 및 SW의 모든 측면을 보호하는 것이다. 장치 간 상호작용에서 서버 역할을 하는 장치는 리소스를 보유하고 제어하며 보안 메커니즘 집합에 따라 클라이언트 역할을 하는 장치에 해당 리소스에 대한 액세스 권한을 제공한다.

보안 작동 이론은 다음 그림에 나와 있으며 다음 단계에 설명되어 있다.

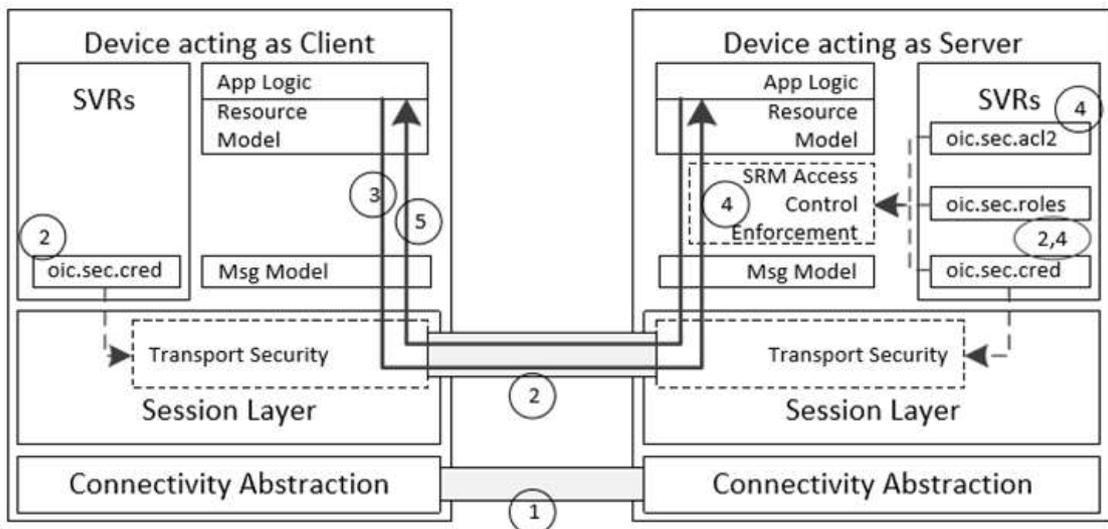


그림 2 - OCF Layers

- 1) 클라이언트는 서버 (자원을 보유한 장치)에 네트워크 연결을 설정한다. 연결 추상화 계층은 연결 옵션의 차이에도 불구하고 장치가 연결될 수 있도록 한다.
- 2) 장치 (예: 서버 및 클라이언트)는 두 장치 간에 상호 인증된 보안 채널이 있거나 없는 메시지를 교환한다.
  - a) 각 장치의 "oic.sec.cred" 리소스는 상호 인증 및 인증서 유효성 검사에 사용되는 자격 증명을 보유한다.
  - b) 보안 채널을 통해 수신된 메시지는 "deviceUUID"와 연결된다. 인증서 자격 증명인 경우 "deviceUUID"는 다른 장치에서받은 인증서에 있다. 대칭 키 자격 증명인 경우 "deviceUUID"는 "oic.sec.cred" 리소스의 자격 증명으로 구성된다.
  - c) 서버는 클라이언트를 여러 역할 ID와 연결할 수 있다. 인증서를 사용한 상호 인증의 경우 역할 인증서에 roleid (있는 경우)가 제공된다. 이들은 클라이언트에 의해 서버로 구성된다. 대칭 키의 경우 허용된 roleid (있는 경우)는 "oic.sec.cred"의 자격 증명으로 구성된다.

- d) 보안되지 않은 채널을 통해 서버가 수신 한 요청은 익명으로 처리되며 "deviceUUID"또는 "roleid"와 관련이 없다.
  - 3) 클라이언트는 서버에 요청을 제출한다.
  - 4) 서버가 요청을 받는다.
    - a) 보안되지 않은 채널을 통해 요청이 수신되면 서버는 요청을 익명으로 처리하며 요청과 관련된 "deviceUUID"또는 "roleid"가 없다.
    - b) 보안 채널을 통해 요청이 수신되면 서버는 "deviceUUID"를 요청과 연결하고 서버는 클라이언트의 모든 유효한 roleid를 요청과 연결한다.
    - c) 그런 다음 서버는 ACL (Access Control List)을 참조하여 다음 기준과 일치하는 ACL 항목을 찾는다.
      - i) 요청 된 자원이 ACE의 자원 참조와 일치
      - ii) 요청 된 작업은 ACE의 "권한"에 의해 허용되며,
      - iii) "subjectUUID"는 특수한 와일드 카드 값 세트 중 하나를 포함하거나 장치가 익명이 아닌 경우 요청과 연관된 클라이언트 장치 ID 또는 요청과 관련된 유효한 "역할"과 일치한다. 와일드 카드 값은 인증 및 암호화 된 세션을 통해 통신하는 모든 장치 또는 인증되지 않은 및 암호화되지 않은 세션을 통해 통신하는 모든 장치와 일치한다. 일치하는 ACE가 있으면 리소스에 대한 액세스가 허용된다. 그렇지 않으면 액세스가 거부된다. 서버의 SRM (Secure Resource Manager)이 액세스를 시행한다.
  - 5) 서버는 클라이언트에게 응답을 보낸다. 자원 보호에는 휴식 중 및 전송 중 데이터 보호가 포함된다. 액세스 제어 메커니즘 외에도 OCF 보안 사양에는 서버에 저장되어있는 동안 리소스의 보안 저장소 사양이 포함되어 있지 않다. 그러나 보안을 위한 유희 보호 보안 스토리지와 액세스 제어의 조합을 통해 리소스가 제공될 것으로 예상된다. 안전한 저장은 하드웨어 보안 또는 유희 데이터 암호화를 사용하여 수행할 수 있다. 보안 스토리지의 정확한 구현에는 일련의 강화 요구 사항이 적용되며 인증 지침이 적용될 수 있다. 반면에 운송 보호 데이터는 이 문서의 표준 부분으로 완전히 지정된다. 전송 보호는 자원 계층 또는 전송 계층에서 제공 될 수있다. 이 문서는 DTLS와 같은 메커니즘을 사용하여 전송 계층에서 전송 보호 기능만 지원한다.
- 참고 : DTLS는 페이로드 전체를 보호하는 대신 패킷 보호를 통해 패킷을 제공한다. 예를 들어 전체 페이로드 전체의 무결성이 필요한 경우 패킷을 전송 계층으로 전달하기 전에 별도의 서명 메커니즘이 이미 설치되어 있어야 한다.

다음 그림은 OCF 보안 적용 지점을 보여줍니다.

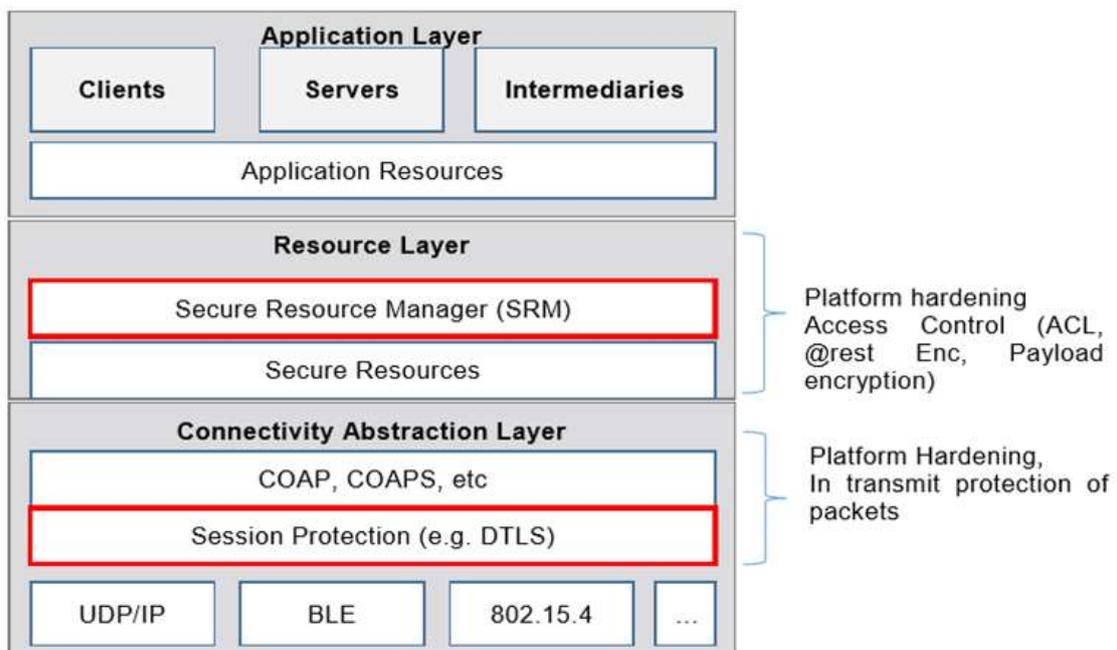


그림 3 - OCF Security Enforcement Points

신뢰할 수 있는 중재자가 장치를 프로비저닝 한 경우 장치는 OCF 클라우드와 통신 할 수 있다.

- "/oic/sec/cred"를 사용하여 장치 및 중재자가 DTLS를 통해 연결
- 중재자는 다음 정보를 사용하여 장치를 프로비저닝한다.
- OCF 클라우드의 URI

- OCF 클라우드에서 검증 할 수있는 토큰
- OCF 클라우드의 UUID

이 문서에서 사용된 OpenAPI 2.0 정의는 표준이다. 여기에는 정의된 모든 페이로드가 표시된 OpenAPI 2.0 정의를 준수해야 한다.

OCF 프레임 워크는 리소스가 서버에서 호스팅되고 액세스 제어 및 권한 부여 메커니즘에 따라 클라이언트가 사용할 수 있다고 가정한다. 엔드 포인트의 리소스는 액세스 제어, 인증 및 기밀성 보호를 통해 보호된다.

액세스 제어의 구현은 리소스에 대한 액세스 정책 세트의 사전 정의에 의존한다. 정책은 로컬 ACL 또는 ACE (Access Control Entry) 형태로 AMS (Access Management Service)에 의해 저장 될 수 있다. 두 가지 유형의 액세스 제어 메커니즘을 적용 할 수 있다.

- 주제별 액세스 제어 (SBAC). 여기서 각 ACE는 요청 엔티티의 주제 (예 : 요청자의 신원)를 자원에 대해 정의 된 정책에 포함 된 주제와 일치시킨다. 요청자의 신원을 확인하려면 인증 프로세스가 필요하다.
- RBAC (역할 기반 액세스 제어). 각 ACE는 자원 정책에 포함 된 역할 식별자를 요청자와 관련된 역할 식별자와 일치시킨다.

컬렉션과 같은 일부 리소스는 적절한 인터페이스가 사용될 때 연결된 리소스에 대한 요청을 생성한다. 이러한 경우 추가 액세스 제어 고려사항이 필요하다.

OCF 액세스 제어 모델에서 리소스 인스턴스에 액세스하려면 관련 ACE가 필요하다. 이러한 관련 ACE가 없으면 리소스에 액세스 할 수 없다.

ACE는 ACE가 주제 (예 : OCF 클라이언트) 및 요청된 리소스와 모두 일치하는 경우에만 적용된다. (1) 장치 ID, (2) 역할 식별자 또는 (3) 와일드카드와 같이 주제를 일치시킬 수있는 여러 가지 방법이 있다. 클라이언트가 서버에 연결하는 방식은 액세스 제어 결정을 내리는 데 관련된 컨텍스트 일 수 있다. 인증된/인증되지 않은 및 암호화된/암호화되지 않은 연결에서의 와일드 카드 일치 액세스 정책이 주제 클래스에 광범위하게 적용될 수 있도록 한다.

---

와일드 카드 일치 정책 예 :

```
"aclist2": [
  { "subject": {"conntype": "anon-clear" },
    "resources":[
      { "wc":"*" }
    ],
    "permission": 31
  },
  {
    "subject": {"conntype": "auth-crypt" },
    "resources":[
      { "wc":"*" }
    ],
    "permission": 31
  },
]
```

---

ACL 형식에 대한 자세한 내용은 이 문서에 정의되어 있다. ACL은 하나 이상의 ACE로 구성된다. ACL은 장치에 대한 액세스 제어 정책을 정의한다. ACL 자원은 SRM 및 PSI의 스토리지 및 처리에 있어 다른 민감한 자원과 동일한 보안 보호가 필요하다. 따라서 ACL 보호를 위해 기본 플랫폼 (HW 및 SW)의 강화를 고려해야 하며 ACL의 범위 지정 수준이 다를 수 있으므로 각 범위 지정 수준마다 강화를 특별히 고려해야 한다. 예를 들어, 물리적 장치는 여러 장치 구현을 호스팅 할 수 있으므로 동일한 장치의 다른 서버에 대한 ACL의 저장소, 사용 및 격리를 보호해야 한다.

서버는 요청을 처리하기 전에 클라이언트가 요청한 자원을 검사한다. 요청자와 요청된 리소스와 일치하는 하나 이상의 ACE 항목을 찾기 위해 액세스 제어 리소스가 검색된다. 일치하는 것이 있으면 권한 및 기간 제약 조건이 적용된다. 일치하는 항목이 두 개 이상 있는 경우 논리적 사용 권한 권한이 겹치는 기간에 적용된다. 서버는 연결 컨텍스트를 사용하여 주체의 인증 여부와 데이터 기밀성 적용 여부를 결정한다. 주제 일치 와일드 카드 정책은 각 측면에서 일치할 수 있다. 사용자가 인증된 경우 역할 또는 장치 ID

에 따라 세부 단위가 증가할 때 제목 일치가 발생할 수 있다.

각 ACE에는 지정된 리소스 요청자에 적용되는 권한 집합이 포함되어 있다. 권한은 CREATE, RETREIVE, UPDATE, DELETE 및 NOTIFY (CRUDN) 작업의 조합으로 구성된다. 요청자는 장치로 인증하고 선택적으로 하나 이상의 역할로 작동한다. 장치는 역할을 주장 할 때 높은 액세스 권한을 얻을 수 있다. 예를 들어 ADMINISTRATOR 역할은 일반적으로 액세스 할 수없는 추가 리소스 및 OCF 인터페이스를 노출 할 수 있다. 서버는 ACL 자원을 로컬로 호스트 할 수 있다. 로컬 ACL은 AMS에 의한 원격 ACL 처리보다 액세스 제어 처리에서 더 큰 자율성을 허용한다.

다음 사용 사례는 액세스 제어 작동을 설명한다.

사용 사례 1 : 그림에 나와있는 것처럼 서버 장치는 4 개의 리소스 (R1, R2, R3 및 R4)를 호스팅한다. 클라이언트 장치 D1은 서버 장치 5에서 호스팅되는 리소스 R1에 대한 액세스를 요청한다. ACL [0]은 리소스 R1에 해당하며 D1을 인증 된 주제로 포함한다. 따라서 로컬 ACL "/oic/sec/acl2/0"이 요청과 일치하기 때문에 장치 D1은 자원 R1에 대한 액세스를 수신한다.

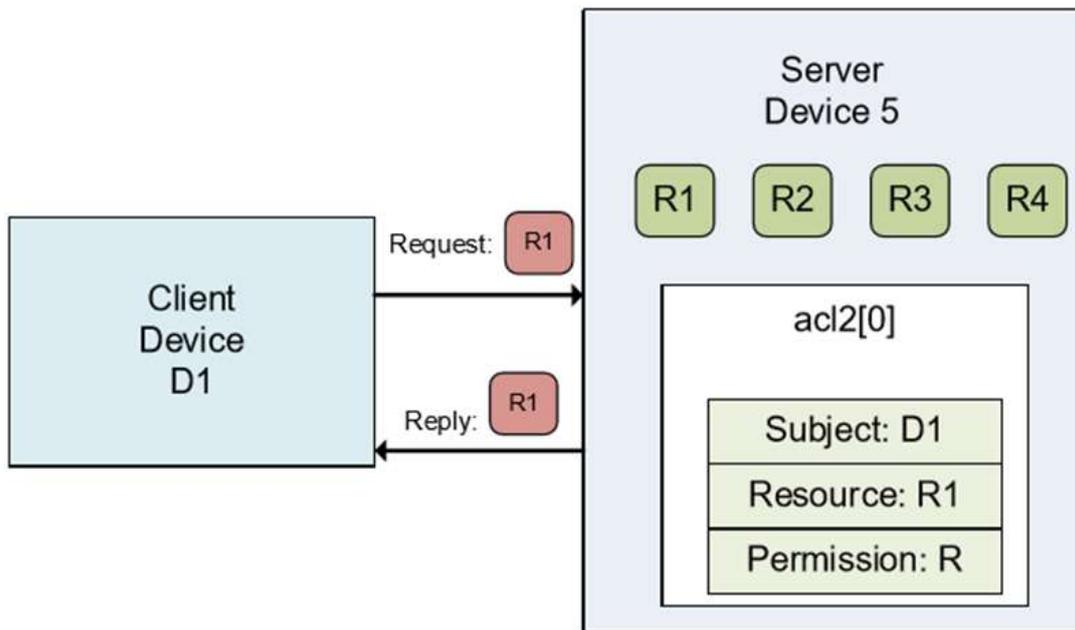


그림 4 - Use case-1 showing simple ACL enforcement

사용 사례 2 : 그림과 같이 클라이언트 R2 액세스는 거부된다. 리소스 R2와 관련된 주제 D2에 대한 로컬 ACL 일치가 없고 AMS 정책이 없기 때문이다.

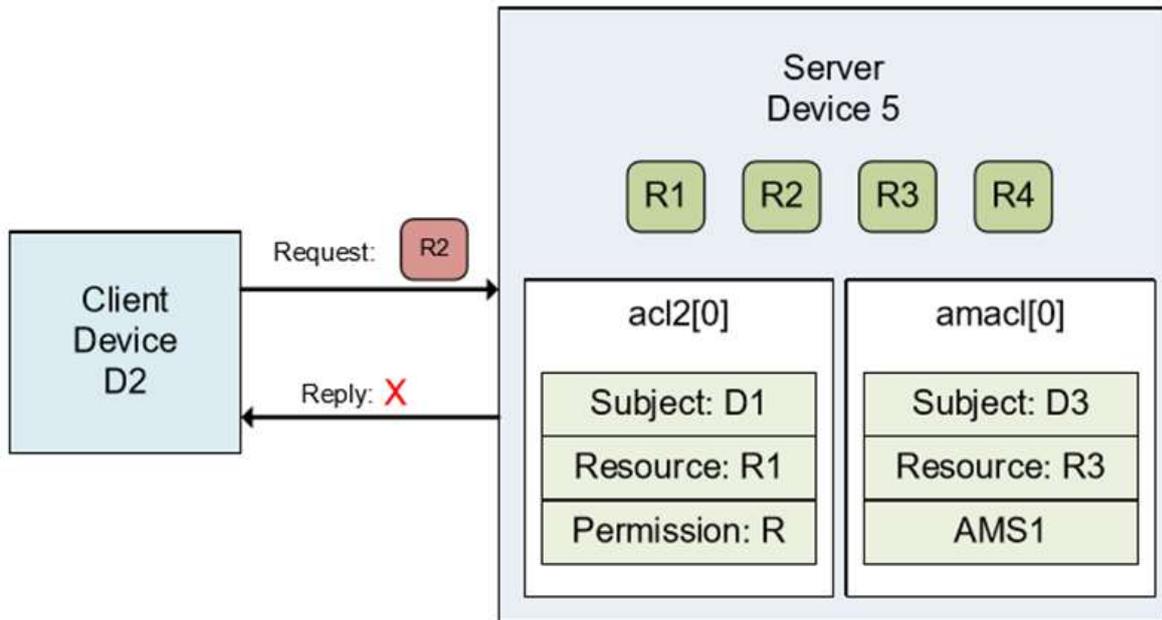


그림 5 - Use case-2 A policy for the request Resource is missing

AMS는 ACL 정책 관리를 향상시킨다. 그러나 이들은 실패의 중심점이 될 수 있다. 네트워크 대기 시간 오버 헤드로 인해 AMS를 통해 ACL 처리 속도가 느려질 수 있다. AMS는 액세스 제어 결정을 중앙 집중화하지만 서버 장치는 시행 의무를 유지한다. 서버는 어떤 자원 세트에 사용할 ACL 메커니즘을 결정해야 한다. "/oic/sec/amacl" 자원은 액세스 결정을 해결하기 위해 AMS를 사용할 자원을 지정하는 ACL 구조이다. "/oic/sec/amacl"은 로컬 ACL ("/oic/sec/acl2")과 함께 사용될 수 있다.

"/oic/sec/acl2.rowneruuid"에 포함 된 장치 식별자에 발급 된 자격 증명을 참조하여 AMS를 인증한다. 서버 장치는 "/oic/sec/acl2.rowneruuid"에 있는 장치 ID를 사용하여 AMS에 대한 연결을 사전에 열 수 있다. 또는 서버가 요청자가 SACL 자원 "/oic/sec/sacl"을 사용하여 적절한 ACE 정책을 얻도록 지시하는 오류 ACCESS\_DENIED\_REQUIRES\_SACL과 함께 자원 액세스 요청을 거부 할 수 있다. "/oic/sec/sacl"서명은 "/oic/sec/acl2.rowneruuid"와 관련된 자격 증명 리소스를 사용하여 확인할 수 있다.

다음 사용 사례는 AMS를 사용한 액세스 제어를 설명한다.

사용 사례 3 : 그림 6에 표시된 것처럼 "/oic/sec/amacl/0"은 Access Manager Server AMS1 서비스를 참조하는 정책과 일치하므로 Device D3은 Perm1 권한으로 Resource R3에 대한 액세스를 요청하고 받는다.

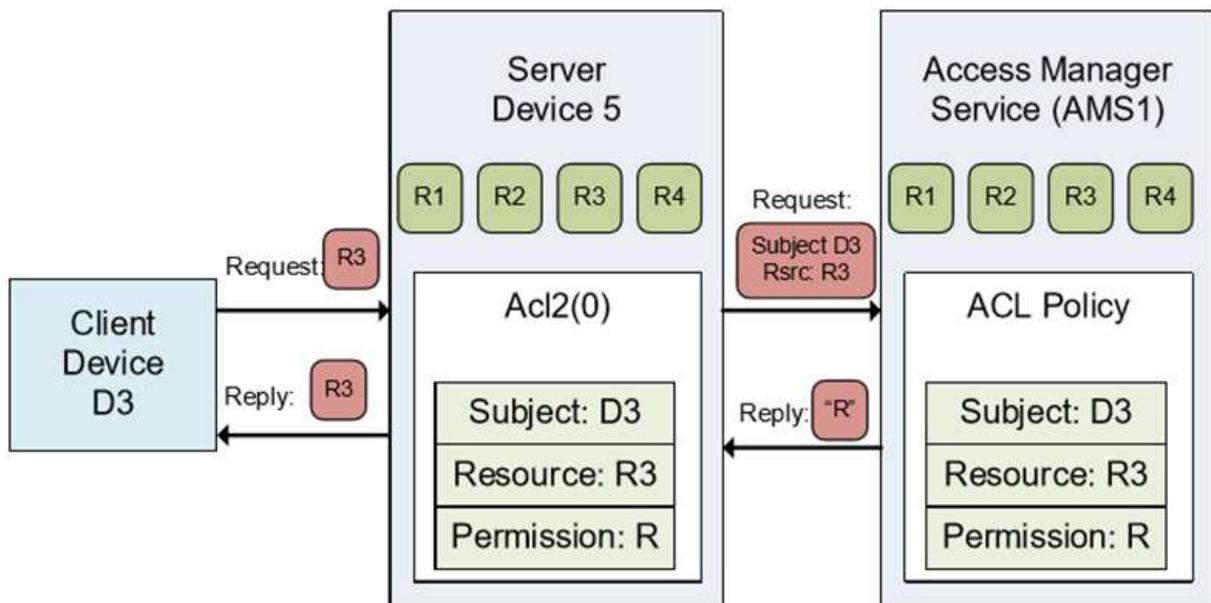


그림 6 - Use case-3 showing AMS supported ACL

사용 사례 4 : 그림 7에 표시된 것처럼 클라이언트 장치 D4는 서버 장치 5에서 리소스 R4에 대한 액세스를 요청한다. 서버 장치 5는 일치하는 ACE를 찾지 못하고 AMS1을 "/oic/로 식별하는 오류를 반환하여 클라이언트 장치 D4를 AMS1로 리디렉션 한다. sec/sacl "리소스 발급자. 장치 D4는 AMS1에 의해 서명된 Sacl1을 획득하고 SACL을 서버 D5에 전달한다. D5는 "/oic/sec/sacl" 리소스의 서명을 확인하고 Perm2 액세스 권한을 부여하는 ACE 정책을 평가한다.

ACE 리디렉션은 D4가 일치하지 않는 이유 코드 (예 : ACCESS\_DENIED\_NO\_ACE)와 함께 오류 결과를 수신 할 때 발생할 수 있다. D4는 "/oic/sec/acl2"리소스를 읽고 AMS를 식별하는 "rowneruid"를 찾은 다음 프로비저닝 요청을 제출한다.이 예에서 AMS는 SACL 리소스를 제공하기로 선택하지만, 로컬 ACL 자원 "/oic/sec/acl2"를 프로비저닝하십시오. 그 후에 요청이 재발행된다. D4는 장치 Onboarding의 일부로 또는 후속 자격 증명 프로비저닝 작업을 통해 AMS에 도입 된 것으로 추정된다.

그렇지 않은 경우 자격 증명 관리 서비스 (CMS)에 문의하여 필요한 자격 증명을 제공 할 수 있다.

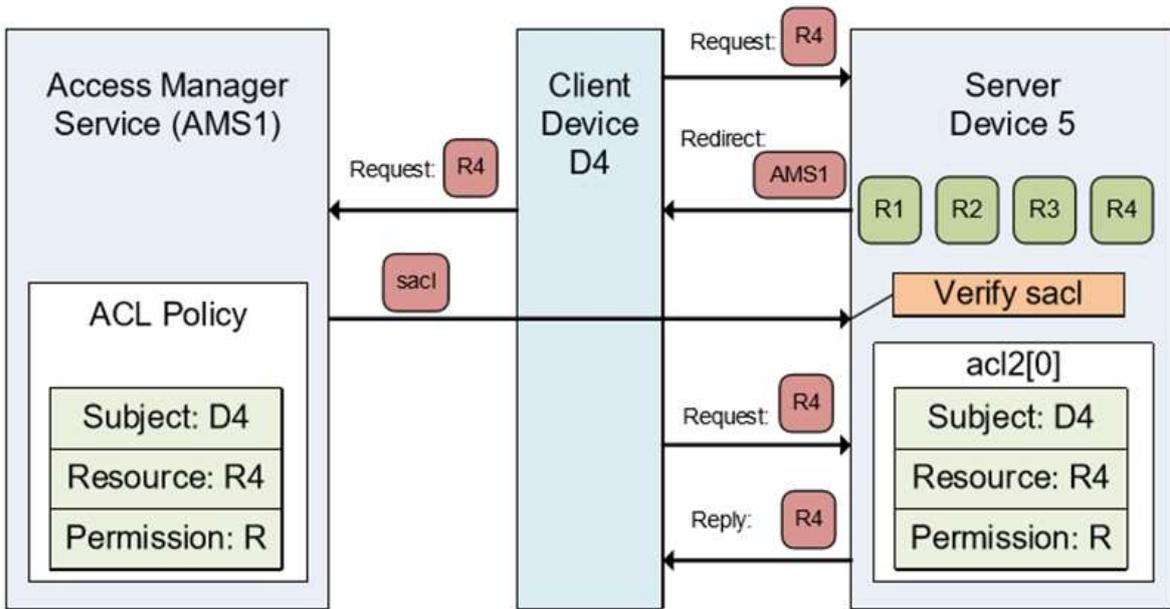


그림 7 - Use case-4 showing dynamicallu obtained ACL from an AMS

그룹 수준 액세스-그룹 범위는 특정 컨텍스트에 대해 그룹화된 장치 그룹에 AC를 적용하는 것을 의미한다. 그룹 수준 액세스는 모든 그룹 구성원이 그룹 데이터에 액세스 할 수 있지만 그룹이 아닌 구성원에게는 명시적 액세스 권한이 부여되어야 함을 의미한다. 역할 자격 증명 및/또는 연결 유형을 사용하여 그룹 수준 액세스가 구현된다.

- OCF 장치 수준 액세스 : OCF 장치 범위는 여러 리소스를 포함 할 수 있는 개별 장치에 AC를 적용하는 것을 의미한다. 장치 수준 액세스는 접근성이 장치 ID로 식별된 장치에 사용 가능한 모든 리소스로 확장됨을 의미한다. 장치에서 AC 메커니즘에 사용되는 자격 증명은 OCF 장치에 따라 다르다.
- OCF 리소스 수준 액세스 : OCF 리소스 수준 범위는 개별 리소스에 AC를 적용하는 것을 의미한다. 리소스 액세스에는 리소스 (서버)를 보유한 엔터티가 요청 엔터티 (클라이언트)가 리소스에 액세스 할 수 있도록 결정하는 방법을 지정하는 ACL이 필요하다.
- 속성 수준 액세스 : 속성 수준 범위는 개별 속성에만 AC를 적용하는 것을 의미한다. 속성 수준 액세스 제어는 단일 속성을 포함하는 리소스를 생성해야만 가능하다.

리소스를 재 설계하는 것이 실용적이지 않은 정적 리소스에 대한 액세스 제어 별도의 액세스 권한이있는 지식 리소스를 참조하는 컬렉션 리소스를 도입하는 것이 적절할 수 있다. "oic.thing"리소스에는 서로 다른 권한이 필요한 Property-1과 Property-2의 두 가지 속성이 있는 예가 그림에 나와 있다.

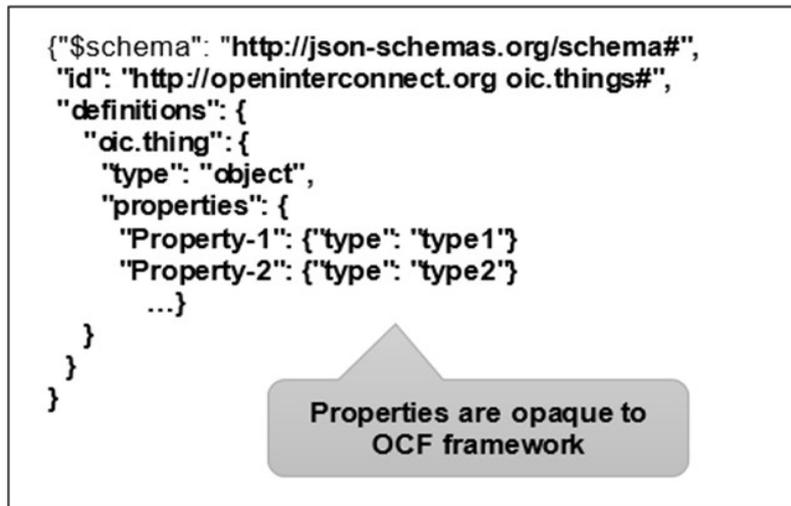


그림 8 - Example Resource definition with opaque Properties

현재 OCF 프레임 워크는 적절한 수준의 정보를 불투명 한 것으로 취급한다. 따라서 ACL 정책의 일부로 다른 권한을 할당 할 수 없다 (예 : Property-1에 대한 읽기 전용 권한 및 Property-2에 대한 쓰기 전용 권한). 따라서 그림 9와 같이 "oic.thing"은 두 개의 새로운 리소스 "oic.RsrcProp-1"과 "oic.RsrcProp-2"로 나뉩니다. 이런 식으로 리소스 수준 ACL을 사용하여 속성 수준 ACL을 얻을 수 있다.

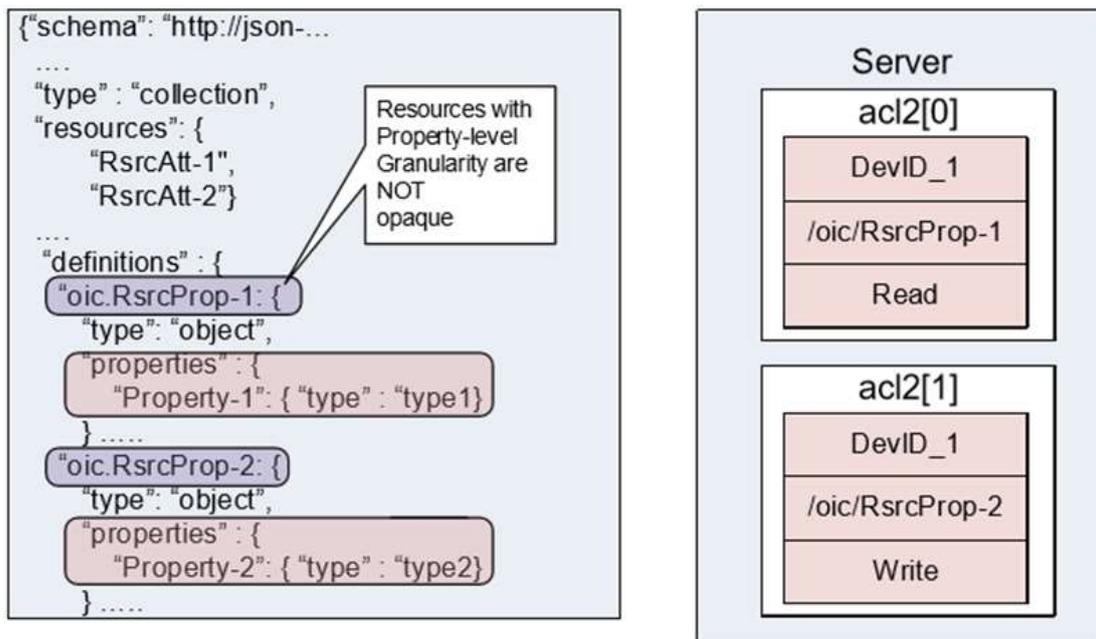


그림 9 - Property Level Access Control

장치가 OCF 환경에서 작동하고 다른 장치와 상호 작용하려면 먼저 Onboard 상태에 있어야 한다. 장치 Onboarding의 첫 번째 단계는 장치를 소유/구매하는 적절한 사용자가 Onboard 도구 (OBT)를 사용하고 OBT를 사용하는 경우 소유자 전송 방법 (OTM) 중 하나를 사용하여 소유권을 설정하는 소유권을 구성하는 것이다. 소유권이 설정되면 OBT는 장치를 프로비저닝 할 수 있는 메커니즘이 되며, 마지막에 장치가 작동하고 OCF 환경에서 다른 장치와 상호 작용할 수 있다. OBT는 OCF 장치에서 호스팅 된다. 그림은 Onboarding 개요를 나타낸다.

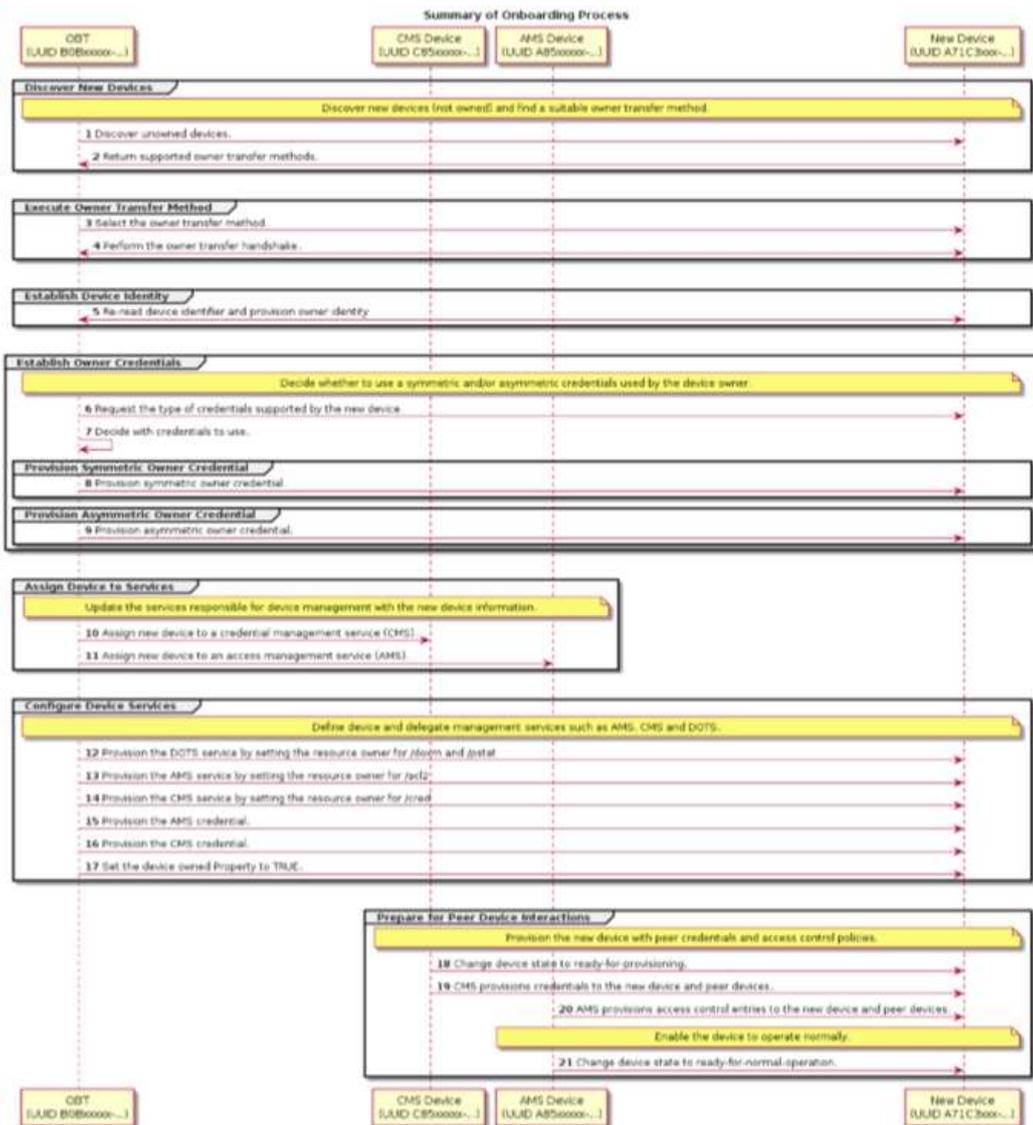


그림 10 - Onboarding Overview

Onboarding 및 프로비저닝 구현은 이 문서 외부에 정의된 서비스를 활용할 수 있다. 다른 서비스를 사용하는 경우 Onboard 중인 장치와 다양한 도구 간의 트러스트가 전이되지 않을 것으로 예상된다. 이는 Onboarding 중인 장치가 Onboarding 프로세스 중에 사용된 모든 도구의 자격 증명을 개별적으로 인증함을 의미한다. 도구가 자격 증명을 공유하지 않거나 아직 설정되지 않은 신뢰 관계를 암시한다.

다음 그림의 플로우차트는 Onboard 중에 관련된 일반적인 단계를 보여준다. Onboarding에는 다양한 비보안 관련 단계가 포함될 수 있지만, 주로 OCF 환경에서 새로운 기기가 작동할 수 있도록 하기 위한 보안 관련 구성에 초점을 맞추고 있다. Onboarding은 일반적으로 기기가 소유 기기가 된 후 운영 환경에 맞게 장치를 구성하는 것으로 시작한다. 여기에는 기기에 액세스할 수 있는 사용자, 수행할 수 있는 작업 및 기기가 다른 장치와 상호 작용하기 위한 사용 권한과 같은 설정 정보가 포함된다.

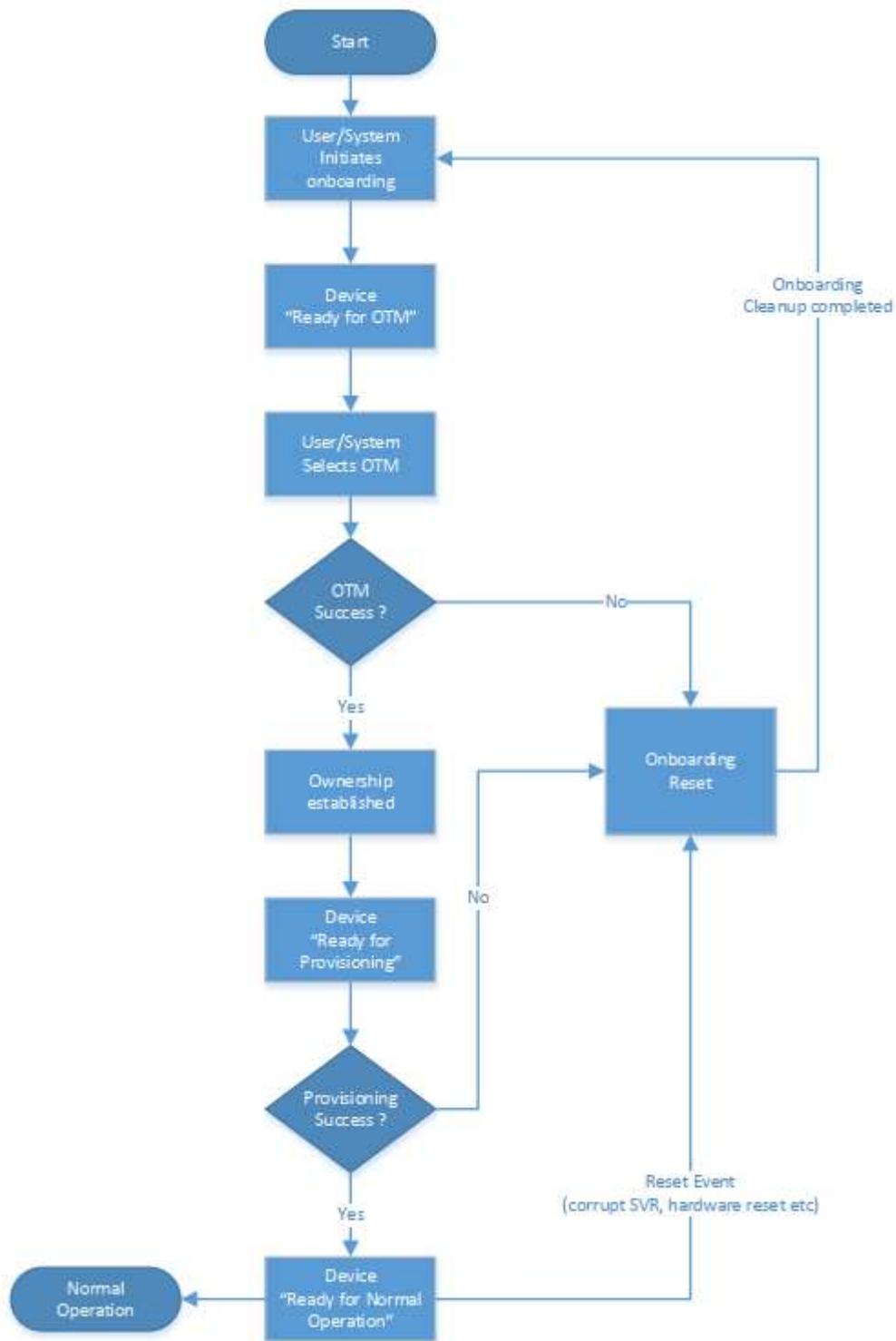


그림 11 - OCF Onboarding Process

장치 소유권을 설정하는 목적은 장치를 소유/구매한 합법적인 사용자가 자신을 장치의 소유자 및 관리자로 주장할 수 있도록하는 것이다. 이는 새 장치와 OBT 도구 간의 소유권 컨텍스트 생성을 포함하고 장치의 운영 제어 및 관리를 주장하는 OBT를 사용하여 수행된다. OBT는 네트워크 관리 콘솔, 장치 관리 도구, 네트워크 작성 도구, 네트워크 프로비저닝 도구, 홈 게이트웨이 장치 또는 홈 자동화 컨트롤러와 같은 도구/서버가 호스팅하는 논리적 개체로 간주될 수 있다. OBT를 호스팅하는 물리적 장치에는 일부 보안 강화 요구 사항이 적용되므로 저장된 자격 증명의 무결성과 기밀성을 유지한다. 장치 소유권을 설정하는 도구/서버를 OBT라고한다.

OBT는 OTM(Own Transfer Methods) 중 하나를 사용하여 장치 소유권을 안전하게 설정한다. 소유자 이전이라는 용어는 새 장치의 경우에도 장치의 제조업체/제공자에서 새 장치의 구매자/법적 사용자에게 소유권이 이전된다고 가정하기 때문에 사용된다.

OTM은 장치 관리 권한이있는 새로운 소유자 (OBT 운영자)를 설립한다. 소유자 양도는 다음을 설정한

다.

- DOTS는 장치의 "/oic/sec/cred" 리소스에서 creds 속성에 소유자 자격 증명 (OC)을 제공한다. 이 OC를 사용하면 후속 상호 작용 중에 장치와 DOTS가 상호 인증 할 수 있다. OC는 DOTS DeviceID를 "/oic/sec/doxm"자원의 rowneruuid 특성과 연관시켜 자원 소유자로 설정한다. DOTS는 소유권 이전의 일부로 장치의 신원을 기록한다.
- 장치 소유자는 OTM을 통해 장치를 신뢰한다.
- 필요할 수 있는 자격 증명을 제공하여 장치 준비를 위한 준비.

장치에 프로비저닝을 시작하는 데 필요한 정보가 있으면 다음 단계는 장치가 작동 할 수 있도록 추가 보안 구성을 프로비저닝하는 것이다. 여기에는 다양한 매개 변수 설정이 포함될 수 있으며 여러 단계가 포함될 수도 있다. 또한 디바이스에서 서버가 호스트하는 다양한 자원에 대한 ACL 프로비저닝도 현재 수행된다. 프로비저닝 단계는 이 단계에만 국한되지 않다. 장치 프로비저닝은 장치 운영 수명주기의 여러 단계에서 발생할 수 있다. 그러나 리소스 및 속성 상태에 대한 특정 보안 관련 프로비저닝은 이 단계에서 발생하여 각 장치가 Onboard 장치 "정상 작동 준비" 상태에 도달한다.

"정상 작동 준비"상태는 사용된 특정 OTM 또는 프로비저닝 대상의 변동성에 관계없이 일관되고 잘 정의되어 있어야 한다. 그러나 개별 OTM 매커니즘 및 프로비저닝 단계는 리소스 및 속성 상태의 추가 구성을 지정할 수 있다. 장치가 "정상 작동 준비"상태에 있어야 하는 최소 필수 구성은 이 문서에 지정되어 있다.

장치와 OCF 클라우드 간의 통신은 단일 로컬 네트워크 내에 있는 장치와 비교하여 다른 기준에 따른다. 장치는 OCF 클라우드 사양의 "CoAPCloudConf"리소스 절에 지정된 대로 중재자가 OCF 클라우드에 연결하도록 구성된다. 프로비저닝에는 원격 연결 및 OCF 클라우드 호스팅 환경을 찾을 수 있는 URL 및 OCF 클라우드 확인 가능한 액세스 토큰과 같은 로컬 세부 정보가 포함된다.

OCF 준수 관리 시스템 (OCMS)은 OCF 장치에 대한 인증 상태 및 정보를 제공하는 OCF에서 유지 관리하는 서비스이다.

OCMS는 <https://www.openconnectivity.org/certification/ocms-cpl.json>에서 호스팅되는 JSON 형식의 인증 제품 목록 (CPL)을 제공해야 한다.

OBT는 URI <https://www.openconnectivity.org/certification/ocms-cpl.json>에 https 연결을 활성화하는 데 필요한 루트 인증서를 보유해야 한다.

OBT는 OCF 보안 도메인 소유자 정책 요구 사항에 따라 URI <https://www.openconnectivity.org/certification/ocms-cpl.json>을 통해 CPL 사본을 주기적으로 새로 고쳐야 한다.

일반적으로 프로비저닝에는 디바이스를 의도한 환경 (Onboarding 프로세스의 일부)으로 가져온 후의 프로세스뿐만 아니라 디바이스의 제조 및 배포 중에 프로세스가 포함될 수 있다. 이 문서에서 보안 프로비저닝에는 소유권 이전 후 프로세스 (소유권 이전 및 Onboarding 중 일부 활동으로 인해 장치에서 일부 데이터가 프로비저닝 될 수 있음에도 불구하고) 프로비저닝 서비스와 상호 작용하기 위한 자격 증명 구성, 보안 관련 리소스 및 자격 증명 구성이 포함된다. 장치가 나중에 연락해야 하는 모든 서비스를 처리한다.

소유권 이전이 완료되면 장치는 CMS 및 AMS와 연결하여 정기적인 작동을 위한 적절한 보안 자격 증명 및 매개 변수를 제공해야 한다. 이러한 매개 변수는 다음을 포함할 수 있다.

- 현재 동일한 OBT에 배포 된 것으로 간주되는 CMS를 통한 보안 자격 증명.
- AMS를 통한 액세스 제어 정책 및 ACL (현재 동일한 OBT에 배포 된 것으로 가정되지만 향후 AMS의 일부일 수 있음).

언급했듯이 확장 가능한 모듈 식 설계를 수용하기 위해 이러한 기능은 향후 별도의 서버로 배포 할 수 있는 서비스로 간주된다. 현재 배포에서는 이러한 서비스가 모두 OBT의 일부로 배포 된 것으로 가정한다. 물리적 배포 시나리오에 관계없이 동일한 보안 강화 요구 사항)은 여기에서 설명하는 도구 및 보안 프로비저닝 서비스를 호스팅하는 모든 물리적 서버에 적용된다.

장치는 보안 프로비저닝 상태를 알고 있다. 자체 인식을 통해 장치 운영 목표를 달성하는 데 필요한 보안 리소스를 프로비저닝하거나 재 프로비저닝하는 데 능동적으로 대처할 수 있다.

잠재적으로 다른 장치 관리 서비스 호스트 사용을 지원할 수 있도록 각 SVR (Device Secure Virtual Resource)에는 리소스의 rowneruuid 속성에 식별된 관련 리소스 소유자가 있다.

DOTS는 "/oic/sec/doxm"및 "/oic/sec/pstat"리소스의 rowneruuid 속성을 DOTS 리소스 소유자 식별자로 업데이트해야 한다.

DOTS는 "/oic/sec/cred"리소스의 rowneruuid 속성을 CMS 리소스 소유자 식별자로 업데이트해야 한다. DOTS는 "/oic/sec/acl2"자원의 rowneruuid 속성을 AMS 자원 소유자 식별자로 업데이트해야 한다.

이러한 OCF 서비스가 구성되면 장치는 사전에 프로비저닝을 요청하고 프로비저닝 요청이 승인되었는지 확인할 수 있다. DOTS는 OCF 서비스와 새로운 장치 간의 보안 연결을 가능하게 하는 자격 증명을 제공해야 한다. DOTS는 OCF 서비스를 시그널링함으로써 클라이언트 지정 프로비저닝을 시작할 수 있다. DOTS는 "/oic/sec/pstat"자원의 tm 속성을 설정하여 서버 지정 프로비저닝을 시작할 수 있다.

"/oic/sec/cred"리소스는 다음을 포함한 여러 유형의 자격 증명을 지원한다.

- 쌍 대칭 키
- 그룹 대칭 키
- 인증서
- 원시 비대칭 키

CMS는 DOTS가 프로비저닝 한 CMS 자격 증명을 사용하여 장치 간 상호 작용을 위한 자격 증명을 안전하게 제공해야 한다.

다음 예는 장치가 피어 장치와 관련된 대칭 키 자격 증명을 업데이트하는 방법을 설명한다. 장치가 업데이트할 자격 증명을 검색한다. 예를 들어, 보안 연결 시도가 실패한다. 장치가 CMS에 업데이트된 자격 증명을 제공하도록 요청한다. CMS가 업데이트 된 대칭 키 자격 증명을 반환한다. CMS는 피어 장치에서 해당 대칭 키 자격 증명을 업데이트한다. 호스트하는 리소스에 대한 요청을 수신한 서버는 리소스를 요청하는 클라이언트가 주장한 역할 식별자를 확인하고 해당 역할 식별자를 서버의 ACL에 설명된 제약 조건과 비교해야 한다. 따라서 클라이언트 장치는 하나 이상의 역할 자격 증명이 제공된다. 각 장치는 자격 정보 리소스 내에서 역할 정보를 속성으로 보유한다.

프로비저닝 된 클라이언트는 인증서 역할 자격 증명에 있는 경우 이 문서에 설명된대로 사용 중인 역할을 주장할 수 있다. 모든 프로비저닝 된 역할은 ACL 시행에 사용된다. 서버에 클라이언트에 대해 여러 역할이 프로비저닝된 경우 역할에 따라 권한이 부여되면 자원에 대한 액세스 권한이 부여된다. ACL 제공은 AMS와 해당 장치 간의 보안 연결을 통해 수행되어야 한다. AMS는 관리하는 각 장치에 대한 ACL 정책을 유지 관리한다. AMS는 기기의 ACL 리소스를 업데이트하여 ACL 정책을 제공해야 한다.

장치가 "/oic/sec/sacl"리소스를 지원하는 경우 AMS는 "/oic/sec/sacl"리소스 발급의 일부로 ACL에 디지털 서명해야 한다. 장치가 서명을 확인하기 위해 사용하는 공개 키는 필요에 따라 CMS에 의해 제공된다. 비대칭 키 유형 또는 서명 된 비대칭 키 유형을 가진 "/oic/sec/cred"리소스가 사용된다. "PublicData"속성에는 AMS의 공개 키가 포함되어 있다.

SRM은 전반적인 보안 운영에서 중요한 역할을 한다. 요컨대, SRM은 리소스 액세스 및 조작 요청에 대해 SVR 관리 및 액세스 제어를 모두 수행한다. SRM은 3가지 주요 기능 요소로 구성된다.

- 자원 관리자 (RM) : 1) 필요에 따라 영구 스토리지 (PSI 사용)에서 SVR로드. 2) 요청이 있을 경우 정책 엔진 (PE)에 리소스를 제공한다. 3) SVR 요청에 응답. SVR이 SRM 메모리에 있는 동안 SVR은 장치별 데이터 저장소 형식과 일치하는 형식으로 되어 있다. 그러나 RM은 JSON 형식을 사용하여 SVR 데이터 구조를 마샬링하여 저장을 위해 PSI로 전달하거나 장치 외부로 이동한다.
- SVR에 대한 액세스 요청을 받고 액세스 제어 정책을 기반으로 하는 정책 엔진(PE)은 "ACCESS\_GRANTED"또는 "ACCESS\_DENIED"를 사용하여 요청에 응답한다. 액세스 결정을 하기 위해 PE는 적절한 ACL을 참조하고 DTLS에 의해 인증 된 주체 (장치 또는 역할)에 따라 요청을 처리할 수 있는 최상의 ACE (Access Control Entry)를 찾는다.
- PSI (Persistent Storage Interface) : PSI는 RM이 자체 메모리 및 스토리지에서 파일을 조작할 수 있도록 API 세트를 제공한다. SRM 설계는 플랫폼의 안전한 실행 환경에서 구현 될 수 있도록 모듈 식이다(가능한 경우).

다음 그림 12는 OCF의 SRM 아키텍처이다.

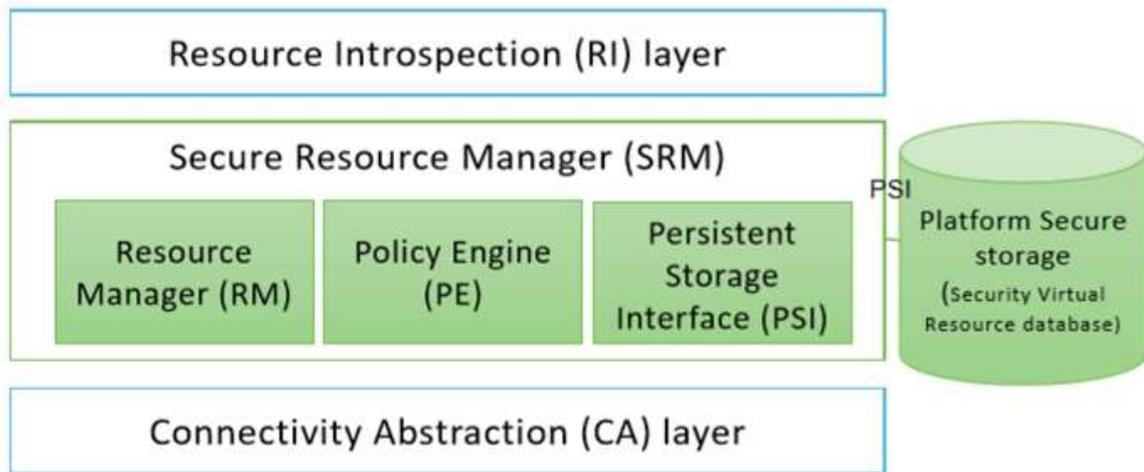


그림 12 - OCF's SRM Architecture

장치는 자격 증명을 사용하여 양방향 통신에서 당사자의 신원과 역할을 증명할 수 있다. 자격 증명은 대칭 또는 비대칭일 수 있다. 각 장치는 해당되는 경우 자체 자격 증명의 비밀 및 공용 부분과 DOTS 또는 CMS에서 제공한 다른 장치의 자격 증명을 저장한다.

그런 다음 이 자격 증명을 보안 통신 세션 (예 : DTLS 사용) 설정에 사용하여 참여 당사자의 신원을 확인한다. 인증된 세션이 설정되면 역할 자격 증명에 사용되어 장치에 대해 하나 이상의 역할을 부여한다.

OCF 클라우드와의 인증된 세션이 설정되면 장치와 관련된 사용자 ID를 확인하기 위해 액세스 토큰이 OCF 클라우드에 제공된다.

#### (나) 감지 프로세스를 위한 보안

감지 메커니즘의 주요 기능은 서버가 호스트 하는 자원에 대한 URI (Universal Resource Identifier) (링크라고 함)를 해당 자원에 대한 속성과 가능한 추가 링크 관계로 보완하는 것이다. (ISO/IEC 301181 : 2018의 조항 10에 따름) 감지 프로세스를 정의할 때 민감한 정보의 보안 또는 애플리케이션의 개인 정보 요구사항을 위반하지 않고 최소 자원 세트만 감지 엔티티에 노출되도록 주의해야 한다. 여기에는 리소스에 포함된 데이터와 해당 메타 데이터가 모두 포함된다.

확장성을 달성하기 위해 이 문서는 각 개별 리소스의 검색 가능성에 대한 권한을 제공하지 않는다. 대신, 자원을 보유한 서버는 각 자원에 대한 ACL을 사용하여 요청자 (클라이언트)가 자원을 보거나 처리할 권한이 있는지 판별한다. "/oic/sec/acl2"자원에는 서버 호스트 자원에 대한 액세스를 제어하는 ACL 항목이 포함되어 있다. ACL 관점에서 리소스의 개인 정보 보호 및 검색 가능성 외에도 검색 프로세스 자체를 보호해야 한다. 이 문서는 발견 프로세스에 대한 다음 요구 사항을 설정한다.

- 1) 발견된 리소스에 대한 무결성 보호 제공.
- 2) 민감한 것으로 간주되는 발견된 리소스에 대한 기밀 보호 기능을 제공한다.

알려진 "/oic/res"리소스에 대해 RETRIEVE 작업 (유니 캐스트 또는 멀티 캐스트)을 수행하여 리소스 검색을 수행한다. 감지 요청은 비보안 채널 (DTLS가없는 멀티캐스트 또는 유니캐스트)을 통해 전송되며 서버는 요청자의 ID를 판별할 수 없다. 이러한 경우 응답하기 전에 클라이언트를 인증하려는 서버는 보안되지 않은 "/oic/res"리소스 응답에 보안 검색 URI (예 : coaps ://IP : PORT/oic/res)를 나열할 수 있다. 이는 보안 검색 URI가 기본적으로 모든 클라이언트가 검색할 수 있음을 의미한다. 그런 다음 클라이언트는 DTLS를 사용하여 보안 검색 URI로 별도의 유니캐스트 요청을 보내야 한다.

보안 검색을 위해 클라이언트에 CRUDN 작업 중 하나 이상 (예 : CRUDN의 비트 단위 OR)을 수행할 권한이 있는 경우에만 관련 ACL2가 있는 모든 자원이 "/oic/res"자원에 대한 응답으로 나열된다. 플래그는 true 여야 한다). 예를 들어, 장치 ID가 "d1"인 클라이언트는 장치 ID가 "d3"인 서버에서 호스트 되는 "/door"자원에 대해 RETRIEVE 요청을 작성한다. 여기서 d3에는 ACL2가 있다.

```

{
  "aclist2": [
    {
      "subject": {"uuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"},
      "resources": [{"href": "/door"}],
      "permission": 2, // RETRIEVE
      "aceid": 1
    }
  ],
  "rowneruuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"
}
{
  "aclist2": [
    {
      "subject": {"authority": "owner", "role": "owner"}
      "resources": [{"href": "/door"}],
      "permission": 2, // RETRIEVE
      "aceid": 2
    }
  ],
  "rowneruuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"
}
{
  "aclist2": [
    {
      "subject": {"uuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"},
      "resources": [{"href": "/door/lock"}],
      "permission": 4, // UPDATE
      "aceid": 3
    }
  ],
  "rowneruuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"
}
{
  "aclist2": [
    {
      "subject": {"conntype": "anon-clear"},
      "resources": [{"href": "/light"}],
      "permission": 2, // RETRIEVE
      "aceid": 4
    }
  ],
  "rowneruuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"
}

```

ACL은 클라이언트 "d1"에 자원에 대한 RETRIEVE 권한이 있음을 나타낸다. 따라서 장치 "d1"이 서버 "d3"의 "/oic/res"리소스에서 검색을 수행하면 응답에 "/door"리소스 메타 데이터의 URI가 포함된다. 클라이언트 "d2"는 두 리소스 모두에 액세스 할 수 있다. ACE2는 "d4"가 업데이트되지 않도록 한다.

d3의 "/oic/res"에 관한 검색 결과가 d1에 전달  
보안 인터페이스의 리소스:

```
[
  {
    "href": "/door",
    "rt": ["oic.r.door"],
    "if": ["oic.if.b", "oic.if.ll"],
    "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1",
  }
]
```

d3의 "/oic/res"에 관한 검색 결과가 d2에 전달  
보안 인터페이스의 리소스:

```
[
  {
    "href": "/door",
    "rt": ["oic.r.door"],
    "if": ["oic.if.b", "oic.if.ll"],
    "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"
  },
  {
    "href": "/door/lock",
    "rt": ["oic.r.lock"],
    "if": ["oic.if.b"],
    "type": ["application/json", "application/exi+xml"]
  }
]
```

d3의 "/oic/res"에 대한 검색 결과가 d4에 전달  
보안 인터페이스의 리소스:

```
[
  {
    "href": "/door/lock",
    "rt": ["oic.r.lock"],
    "if": ["oic.if.b"],
    "type": ["application/json", "application/exi+xml"],
    "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"
  }
]
```

d3의 "/oic/res"와 관련하여 모든 장치에 제공되는 검색 결과

```
[
  {
    "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1",
    "href": "/light",
    "rt": ["oic.r.light"],
    "if": ["oic.if.s"]
  }
]
```

#### (다) 보안 프로비저닝

논리적 각 장치는 장치 ID로 식별된다. 장치는 장치 Onboarding의 일부로 설정된 장치 ID 값으로 식별해야 한다. "/oic/sec/doxm" 리소스는 장치 ID 형식을 지정한다 (예 : "urn : uuid"). 장치 ID는 해당 OCF 보안 도메인의 작동 범위 내에서 고유해야 하며 보편적으로 고유해야 한다.

DOTS는 새로운 장치의 장치 ID가 소유자의 OCF 보안 도메인 범위 내에서 고유한지 확인해야 한다. DOTS는 선택된 새로운 장치 식별자가 이전에 OCF 보안 도메인에 도입된 장치 ID와 충돌하지 않는지 확

인해야 한다. 장치는 "/oic/sec/cred" 리소스를 사용하여 장치 ID와 암호화 자격 증명의 연결을 유지한다. 장치는 피어 장치의 인증 자격 증명을 확인할 때 "/oic/sec/cred" 리소스를 신뢰할 수 있는 것으로 간주한다. 장치는 "/oic/sec/doxm" 리소스에서 장치 ID를 유지 관리한다. "/oic/sec/cred" 리소스에서 자체 및 다른 장치 자격 증명의 자격 증명 목록을 유지 관리한다. 기기 ID를 사용하여 기기 자체의 자격 증명과 다른 기기의 자격 증명을 구분할 수 있다. 또한 "/oic/sec/cred" 리소스에는 장치에 대한 여러 자격 증명도 포함될 수 있다. 장치 ID는 다음과 같아야 한다.

- 독특한
- 불변
- 검증 가능

제조업체 인증서를 사용할 때 인증서는 이 절의 뒷부분에 설명된 대로 장치에 저장된 비밀에 ID를 바인딩해야 한다. OCF 문서에서 플랫폼이라고 하는 물리적 장치는 여러 장치를 호스팅 할 수 있다. 플랫폼은 플랫폼 ID로 식별된다. 플랫폼 ID는 전체적으로 고유해야 하며 무결성 보호 방식 (예 : 보안 저장소 내부 또는 서명 및 확인)으로 장치에 삽입해야 한다. OCF 플랫폼에는 안전한 실행 환경이 있을 수 있으며 고유한 식별자와 비밀을 보호하는 데 사용된다. 플랫폼이 여러 장치를 호스팅 하는 경우 각 장치에 적절하고 별도의 보안을 제공하려면 일부 메커니즘이 필요하다.

장치 소유권은 유익한 조항이다. 장치는 암호화 자격 증명을 사용하여 인증 할 수 있는 ID를 가진 보안 Endpoint인 논리적 엔티티이다. 장치가 처음 초기화될 때 소유되지 않은 장치이다. 장치 소유권 설정은 장치가 자신의 ID를 DOTS에 주장하고 DOTS가 소유자 ID를 제공하는 프로세스이다. 이 교환으로 인해 장치의 소유권 상태가 변경되어 다른 DOTS가 장치에 대한 관리 제어를 주장하지 못하게 된다. 소유권 이전 프로세스는 OBT가 새 장치의 "/oic/sec/doxm" 리소스의 "소유" 속성을 검사하여 소유하지 않은 새 장치를 검색하는 것으로 시작한다. 소유권 이전이 완료되면 다음이 달성된다.

- 1) DOTS는 새로운 장치와 보안 세션을 설정해야 한다.
- 2) 선택적으로 다음 중 하나를 주장한다.
  - a) 플랫폼에 대한 OBT의 근접성 (PIN 사용).
  - b) 플랫폼 벤더, 모델 및 기타 플랫폼 특정 속성을 주장하는 제조업체의 인증서.
- 3) 장치 식별자를 결정한다.
- 4) 장치 소유자를 결정한다.
- 5) 장치 소유자를 지정한다 (예 : OBT의 장치 ID).
- 6) 소유자의 자격 증명으로 기기를 프로비저닝한다.
- 7) 새 장치의 "소유"상태를 TRUE로 설정한다.

참고: OCF 클라우드에 연결된 장치는 여전히 DOTS와의 Onboarding에서 설정된 소유권을 유지한다.

각 개별 소유권 이전 방법의 구현은 선택 사항으로 간주 된다. 그러나 각 장치는 공급 업체별 방법을 포함하지 않는 소유권 이전 방법 중 하나 이상을 구현해야 한다.

이 문서에 포함된 모든 OTM은 선택 사항으로 간주된다. 각 공급 업체는 이 문서에 지정된 OTM 중 하나 이상을 선택하고 구현해야 한다. 그러나 OCF는 공급 업체별 접근 방식이 존재할 것으로 예상된다. 공급 업체가 공급 업체별 OTM과 다른 공급 업체의 OBT간에 상호 운용성을 원할 경우 공급 업체는 OBT 공급 업체와 직접 협력하여 상호 운용성을 보장해야 한다. 그럼에도 불구하고, OTM의 표준화가 선호되는 접근법이다. 이러한 경우 공급 업체별 OTM을 설계하는 공급 업체를 돕기 위해 일련의 지침이 제공된다.

"/oic/sec/doxm" 리소스는 공급 업체 정의 소유자 전송 방법 (OTM)을 수용하도록 확장 할 수 있다. DOTS는 새 장치에 탑재하는 데 가장 적합한 OC를 결정한다. 모든 OTM은 "/oic/sec/doxm" 리소스의 oxms 속성을 사용하여 장치의 Onboarding 기능을 나타낸다. DOTS는 "/oic/sec/cred" 리소스의 "credtype" 속성을 사용하여 장치의 지원되는 자격 증명 유형을 쿼리 해야 한다. DOTS와 CMS는 지원되는 자격 유형에 따라 자격 증명을 제공해야 한다.

그림은 새로운 장치 검색 순서를 보여준다.

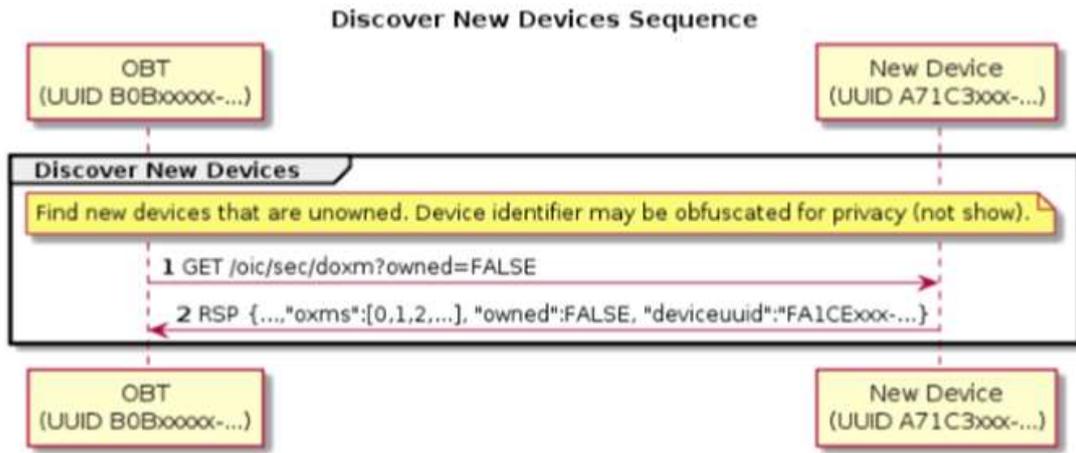


그림 13 - Discover New Device Sequence

표 1 - Discover New Device Details

Step	Description
1	The OBT queries to see if the new device is not yet owned.
2	The new device returns the "/oic/sec/doxm" Resource containing ownership status and supported OTMs. It also contains a temporal device ID that may change subsequent to successful owner transfer. The device should supply a temporal ID to facilitate discovery as a guest device. Clause 7.3.9 provides security considerations regarding selecting an OTM.

공급 업체별 장치 OTM은 공급 업체별 장치 OTM의 결과 인 OC에 대한 "/oic/sec/doxm"리소스 사양을 준수해야한다. 공급 업체별 OTM에는 OBT가 새 장치에 대한 신뢰를 설정하고 선택적으로 새 장치가 OBT에 대한 신뢰를 설정하기 위한 조항이 포함되어야 한다. 새 장치는 OTM 시작시 일부 초기화 단계를 수행해야 할 수도 있다. 예를 들어, 랜덤 PIN 기반 OTM이 시작되면, 새로운 장치는 랜덤 PIN 값을 생성할 수 있다.

OBT는 다른 OTM 단계를 수행하기 전에 사용중인 OTM에 해당하는 값을 "/oic/sec/doxm"의 oxmsel 속성에 POST해야 한다. 이 POST는 소유권 이전이 시작되고 있음을 새 장치에 알립니다. 공급 업체별 OTM의 종료 상태는 새로운 장치가 OBT를 인증하고 OBT가 새로운 장치를 인증하도록 허용해야 한다.

DOTS는 확립된 OTM 세션을 활용하여 소유자 이전에 성공한 후 추가 프로비저닝 단계를 수행할 수 있다. OTM에 성공한 후 새로 Onboard 된 장치를 RFNOP에 배치하기 전에 OBT는 주제가 "명확하지 않은" 또는 "auth-crypt"인 모든 ACE를 제거하고 Resources 배열에는 SVR이 포함된다.

공유키 자격 증명은 Onboarding 에 사용된 DTLS 핸드 셰이크에서 발생하는 key\_block 값을 수락하는 PRF를 사용하여 파생된다. 새로운 장치 및 DOTS는 다음 계산을 사용하여 공급 업체 제품 간의 상호 운용성을 보장해야 한다.

인증서 자격 증명은 안전한 양방향 통신을 위해 장치에서 사용된다. 인증서는 CMS 또는 외부 인증 기관 (CA)에서 발급한다. 이 CA는 장치의 신뢰성을 상호 확립하는 데 사용된다. 인증서 생성에 대한 Onboarding 세부 정보는 이 문서의 이후 버전에서 지정된다.

Just-works OTM은 소유자의 OCF 보안 도메인 내에서 사용하기 위해 장치를 프로비저닝 해야 하는 안전한 연결을 설정하는 데 사용되는 사전 공유키인 대칭 키 자격 증명을 만든다. 추가 자격 증명 및 리소스 프로비저닝은 소유권 설정에 따른 일반적인 단계이다. 사전 공유키를 SharedKey라고 한다.

DOTS는 Just-works OTM을 선택하고 Just-works OTM에 대해 정의 된 암호를 사용하여 DTLS 세션을 설정해야 한다. Just-works OTM에는 다음과 같은 OCF 정의 공급 업체별 암호 모음이 사용된다.

- TLS\_ECDH\_ANON\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_ECDH\_ANON\_WITH\_AES\_256\_CBC\_SHA256

이 값은 IANA에 등록되지 않으며 암호 값은 개인용 예약 영역 (0xFF00 ~ 0xFFFF)에서 할당된다. 할당된 값은 각각 0xFF00 및 0xFF01이다.

Just Works OTM 시퀀스는 다음 그림과 표에 설명된다.

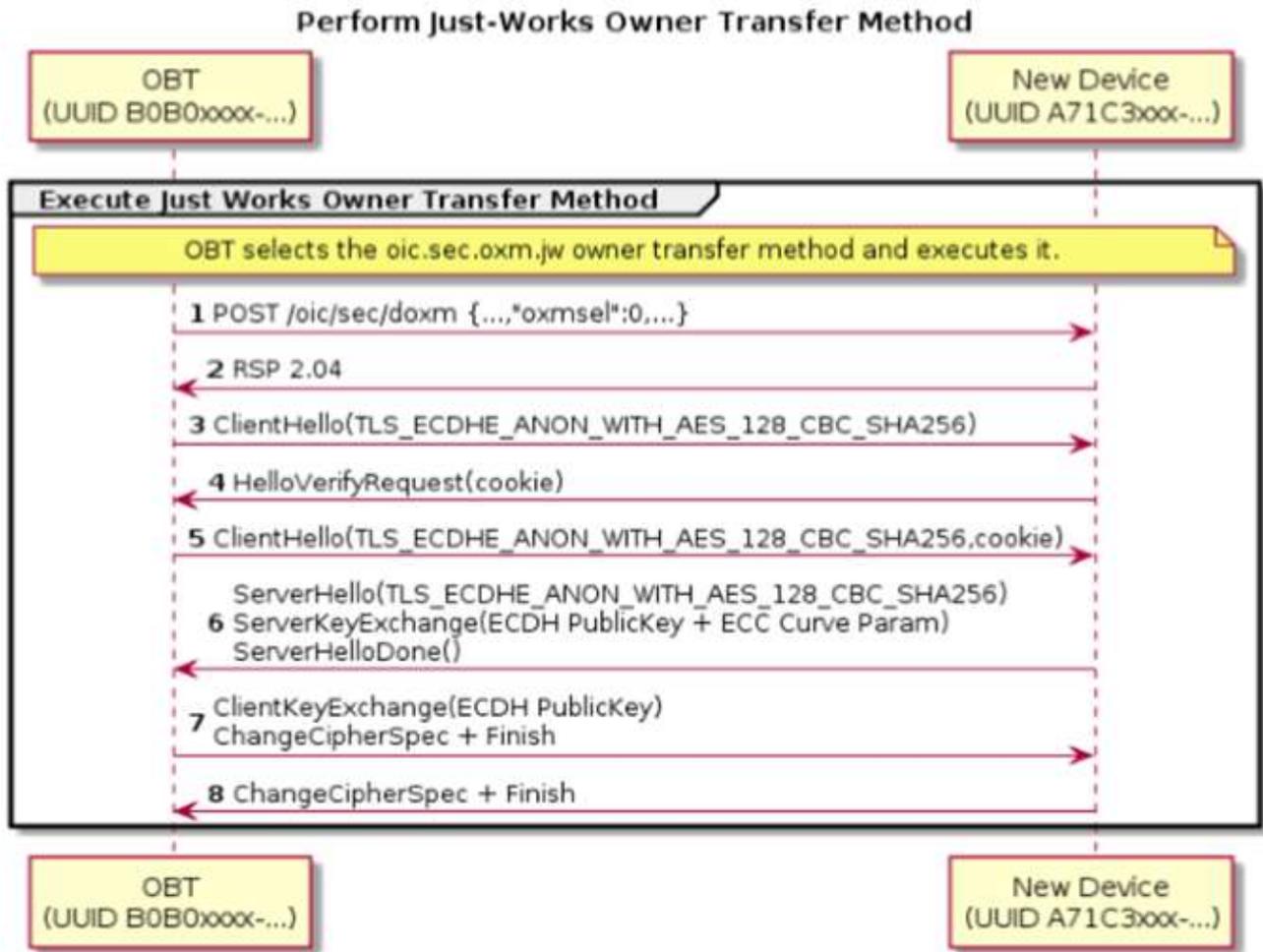


그림 14 - A Just Works OTM

표 2 - A Just Works OTM Details

Step	Description
1, 2	The OBT notifies the Device that it selected the "Just Works" method.
3 - 8	A DTLS session is established using anonymous Diffie-Hellman. <sup>9</sup>
* This method assumes the operator is aware of the potential for man-in-the-middle attack and has taken precautions to perform the method in a clean-room network.	

익명의 Diffie-Hellman 키 계약은 중간자 공격자가 적용된다. 이 방법을 사용하면 OBT와 새 장치 모두 "직접 작동"방법을 수행한다고 가정하고 공격 장치가 없는 비교적 안전한 환경에서 Onboarding이 발생한다고 가정한다. 이 방법은 주장 된 장치 ID가 장치에 안정적으로 바인딩되어 있음을 증명할 수 있는 확실한 방법이 없다.

새 장치는 개인 장치에 민감한 추적을 방지하기 위해 게스트 장치로 간주되는 반면 소유 장치로 전환하기 전에 임시 장치 ID를 사용해야 한다. 장치는 소유자 전송 교환이 발생하는 보안 세션 동안 임시 값과 다를 수 있는 비 일시적 장치 ID를 주장한다. OBT는 주장 된 장치 ID가 이미 사용중인 장치 ID와 충돌하지 않는지 확인한다. 이미 사용 중인 경우 기존 자격 증명을 사용하여 보안 세션을 설정한다. 장치 자격 증명을 설정한 소유되지 않은 장치는 장치가 손상되었거나 손상되었을 수 있다.

임의 PIN 방법은 새 장치와 OBT 사이의 물리적 근접성을 설정하여 중간자 (man-in-the-middle) 공격을 방지할 수 있다. 장치는 대역 외 채널을 통해 OBT와 통신되는 난수를 생성한다. 대역 외 통신 채널의 정의는 장치 OTM의 정의 범위를 벗어납니다. OBT와 새 장치는 키 교환에서 PIN을 사용하여 누군가가 대역 외 채널을 통해 새 장치에 물리적으로 액세스하여 소유권 이전을 승인했음을 나타낸다.



이다. MITM (Man-in-the-Middle) 공격은 공격자가 네트워크에서 활동 중이며 OBT와 장치 간 메시지를 가로채서 수정할 수 있는 경우이다. MITM 공격에서 공격자는 "실시간"으로 즉, 피어가 시간 초과되기 전에 키 교환 메시지에서 PIN을 복구하고 연결 시도를 중단해야한다. PIN을 복구 한 후 키 교환의 인증 단계를 완료 할 수 있다. 여기에 제시된 지침에는 최소 40 비트의 엔트로피가 필요하지만 이 것이 제공하는 보증은 공격자가 사용할 수 있는 리소스에 따라 다르다. 무차별 대입 추측 공격의 병렬화 특성을 고려할 때 더 많은 코어/스레드가 추가됨에 따라 공격은 선형 속도 향상을 향유한다. 보다 보수적인 양의 엔트로피는 64비트이다.

Random PIN OTM은 ECDHE 키 교환을 포함하는 DTLS 암호 스위트를 사용해야 하므로 Random PIN OTM의 보안은 항상 JustWorks OTM의 보안과 동일하다. Random PIN OTM에는 PBKDF2를 사용하여 PIN에서 주요 자료를 파생시키는 옵션도 있다. 이론적 근거는 공격의 각 추측 비용을 조정 가능한 양 (PBKDF2 반복 횟수)만큼 증가시켜 무차별 대입 공격 비용을 늘리는 것이다. 이론적으로 이것은 PIN의 엔트로피 요구 사항을 줄이는 효과적인 방법이다. 불행하게도, 정직한 동료들이 보낸 X- fold 시간의 증가는 공격자가 X- fold 의 시간 증가를 직접적으로 변환하지 않기 때문에 감소량을 정량화하는 것은 어렵다. 이 비대칭은 공격자가 정직한 피어가 사용할 수 없는 특수한 구현 및 하드웨어를 사용할 수 있기 때문이다. 따라서 PIN에 사용할 엔트로피의 양을 결정할 때 구현자는 PBKDF2가 보안을 제공하지 않고 PIN에 충분한 엔트로피가 있는지 확인하는 것이 좋다.

임의 PIN 장치 OTM 보안은 새 장치에서 OBT로 임의로 생성 된 PIN을 통신하기위한 안전한 대역 외 방법이 존재한다는 가정에 따라 다르다. OOB 채널이 일부 또는 전체 PIN을 공격자에게 누설하면 PIN의 엔트로피가 줄어들고 위에서 설명한 공격이 적용된다. 대역 외 메커니즘은 OBT와 새 장치 간의 근접성이 필요하도록 선택해야한다. 공격자는 대역 외 채널을 손상시키지 않은 것으로 가정한다. 예시적인 OOB 채널로서, 장치는 OBT 소프트웨어에 입력 될 PIN을 표시 할 수 있다. 다른 예는 장치가 PIN을 2D 바코드 로 인코딩하고 OBT 장치의 카메라에서 캡처 및 디코딩 할 수 있도록 표시하는 것이다.

제조업체 인증서 기반 OTM은 제조자가 장치에 내장하는 인증서를 사용해야 하며 장치와 OBT 사이의 신뢰 앵커를 결정하는 서명된 OBT를 사용할 수 있다.

제조업체가 포함된 인증서가 반드시 OCF 루트 CA 신뢰 앵커에 연결될 필요는 없다.

일부 환경, 정책 또는 관리자의 경우 장치 특성에 대한 추가 정보를 찾을 수 있다.

인증서 기반 소유권 전송을 사용할 때 장치는 인증서 데이터가있는 비대칭 키를 사용하여 OCF 보안 도메인에서 새 장치를 작동시키는 과정에서 OBT로 ID를 인증해야한다. Onboarding 프로세스에는 몇 가지 개별 단계가 포함된다.

#### 1) 선상 조건

- a) 제조업체 인증서를 포함하는 장치 자격 증명 리소스 ( "/oic/sec/cred")의 자격 증명 요소는 자격 증명을 나타내기 위해 "oic.sec.cred.mfgcert"문자열을 포함하는 "credusage"속성으로 식별해야 한다. 제조업체 인증서를 포함한다.
  - b) 제조업체 인증서 체인은 식별 된 자격 증명 요소의 "공개 데이터"속성에 포함되어야한다.
  - c) 장치는 고유하고 불변의 ECC 비대칭 키 쌍을 포함해야한다.
  - d) 장치가 소유권 이전의 일부로 OBT의 인증을 요구하는 경우, OBT가 등록된 것으로 추정되고 미리 결정된 Trust Anchor가 서명 한 고유하고 불변의 ECC 비대칭 키 쌍에 대한 인증서를 획득 한 것으로 간주된다.
  - e) 사용자가 네트워크 액세스 정보 및 계정 정보 (있는 경우)로 OBT 앱을 구성했다.
- 2) OBT는 ECDSA를 사용하여 장치를 인증하여 서명을 확인해야한다. 또한 장치는 OBT 서명을 확인하기 위해 OBT를 인증 할 수 있다.
  - 3) 인증이 실패하면 장치는 실패 이유를 표시하고 OTM 준비 상태로 돌아갑니다. 인증에 성공하면 장치와 OBT는 협상 된 암호 제품군에 따라 암호화된 링크를 설정해야 한다.

장치와 OBT간에 완전한 상호 인증이 이루어지려면 장치와 OBT 모두 상호 신뢰 앵커 나 인증 기관으로 추적할 수 있어야 한다. 이는 OCF가 인증 기관 (예 : Symantec, Verisign 등)으로부터 모든 후속 OCF Trust Anchor가 파생된 궁극적인 Trust Anchor를 제공하기 위해 서비스를 받아야 할 수도 있음을 의미한다. OBT는 Onboarding 중에 장치를 인증해야 한다. 그러나 장치에 대한 잠재적인 자원 제한으로 인해 장치가 OBT를 인증할 필요는 없다. 장치가 OBT 소프트웨어를 인증하지 않는 경우 악의적인 OBT 소프트웨어가 사용자에게 의해 무의식적으로 배포되거나 악의적으로 배포되어 OCF 보안 도메인 액세스 자격 증명 및/또는 개인 정보가 손상될 수 있다. 임의의 PIN 기반 OTM 시퀀스는 그림 16과 표 4에 설명된 단계로 표시된다.

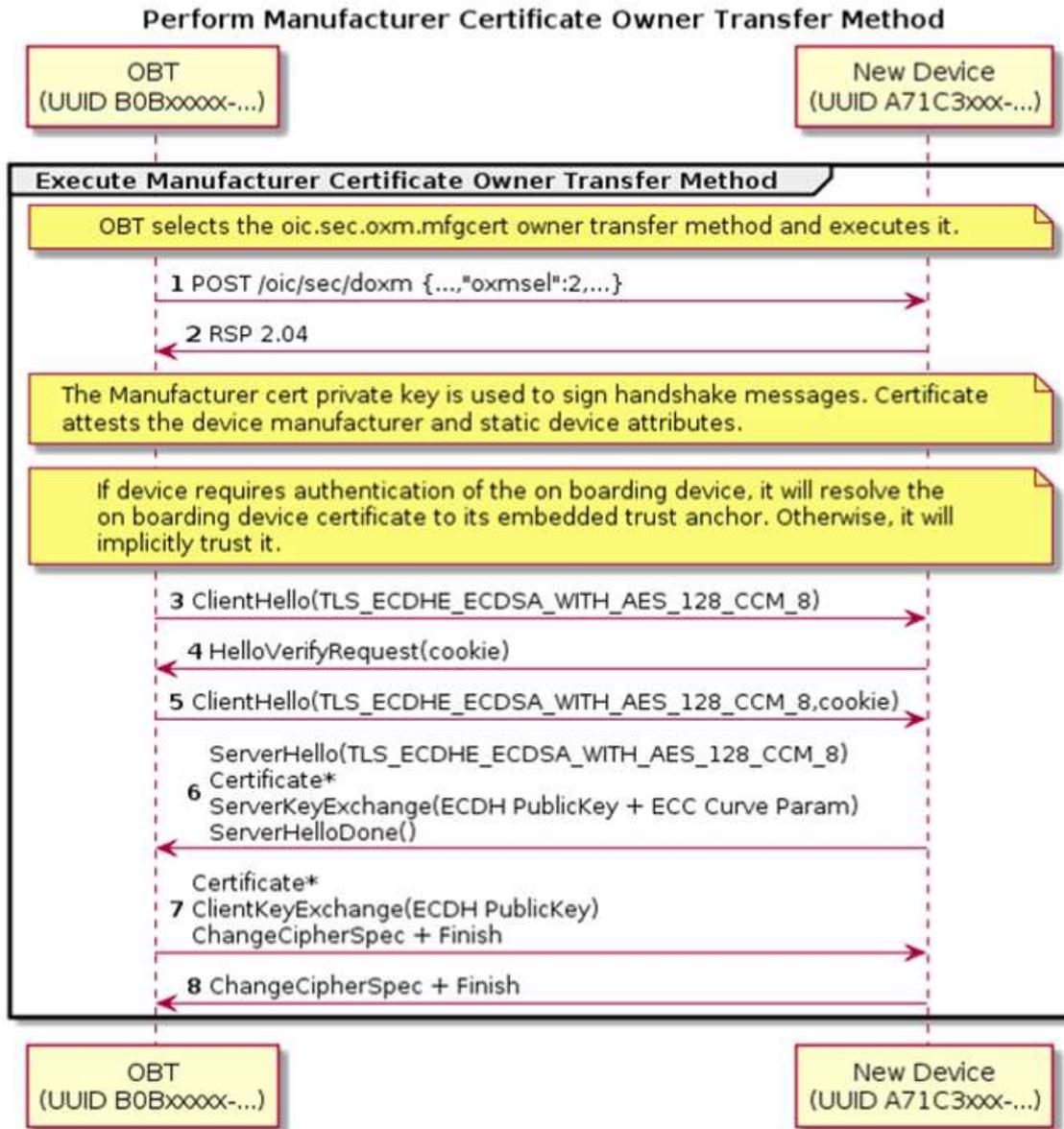


그림 16 - Manufacturer Certificate Based OTM Sequence

표 4 - Manufacturer Certificate Based OTM Details

Step	Description
1, 2	The OBT notifies the Device that it selected the "Manufacturer Certificate" method.
3 - 8	A DTLS session is established using the device's manufacturer certificate and optional OBT certificate. The device's manufacturer certificate may contain data attesting to the Device hardening and security properties.

- 보안 고려 사항

제조업체 인증서 개인 키는 개인 키가 손상 될 수 없음을 충분히 보증하면서 플랫폼에 포함된다.

플랫폼 제조업체는 제조업체 인증서를 발급하고 개인 키 보호 메커니즘을 증명한다.

OCF는 공급 업체가 제조 또는 장치 제약 조건을 수용하는 OTM을 구현해야하는 상황을 예상한다. 이를 위해 Device OTM 리소스를 확장 할 수 있다. 공급 업체별 OTM은 모든 OTM이 따르는 일련의 규칙을 준수해야한다.

- OBT는 장치에서 지원하는 자격 증명 유형을 결정해야한다. 지원되는 자격 증명 유형을 식별하기 위

해 장치의 "/oic/sec/doxm"리소스를 쿼리하면 된다.

- OBT는 장치에 OC를 제공한다.
- OBT는 이후 OBT에 액세스하기 위해 장치 ID 및 자격 증명을 제공한다.
- OBT는 소유권 설정 후 소유자의 OCF 보안 도메인에 액세스하기에 충분한 두 번째 반송파 설정을 제공한다.
- OBT는 추가 프로비저닝 단계를 수행 할 수 있지만 보안 서비스가 수행 할 프로비저닝 작업을 무효화해서는 안된다.
- 공급 업체별 소유자 이전 시퀀스 예

공급 업체별 OTM 시퀀스 예는 다음 그림과 표에 설명된다.

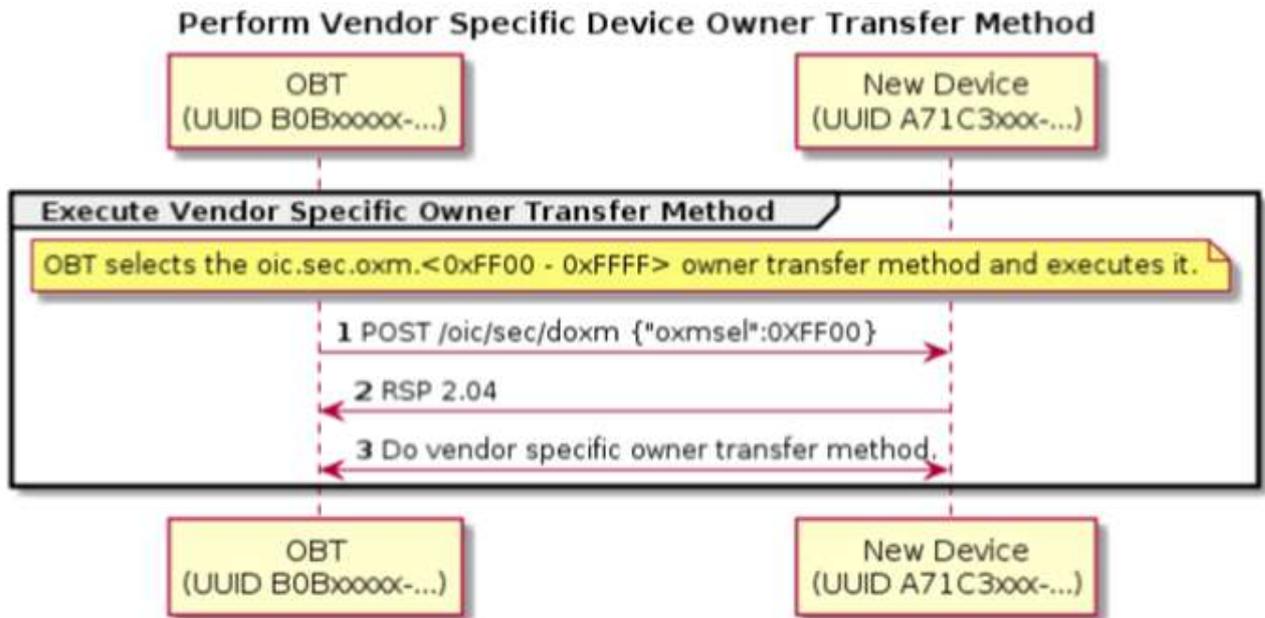


그림 17 - Vendor-specific Owner Transfer Sequence

표 5 - Vendor-specific Owner Transfer Details

Step	Description
1, 2	The OBT selects a vendor-specific OTM.
3	The vendor-specific OTM is applied

OBT와 새 장치가 정의 된 OTM 방법 중 하나를 사용하여 인증되고 암호화 된 연결을 설정하면 소유자 자격 증명은 OBT 또는 기타 기관에서 서명 한 인증서, OCF 보안 도메인 액세스 정보, 프로비저닝 기능, 공유 키 또는 Kerberos 티켓으로 구성 될 수 있다.

그런 다음 OBT는 장치 관리 및 장치 간 통신을위한 추가 자격 증명을 사용하여 새 장치를 프로비저닝 할 수 있다. 이러한 자격 증명은 서명이있는 인증서, 장치 공개 키를 기반으로하는 UAID, PSK 등으로 구성 될 수 있다.

장치의 소유자 자격 증명 (OC)을 설정하는 단계는 다음과 같다.

- 1) OBT는 장치 ID와 장치 소유자를 설정해야 한다.
- 2) 그런 다음 OBT는 장치의 OC를 설정한다.
  - a) 대칭 자격 증명.
  - b) 비대칭 자격 증명.
- 3) 장치 서비스 구성.
- 4) 피어 투 피어 상호 작용을 위한 장치 구성.

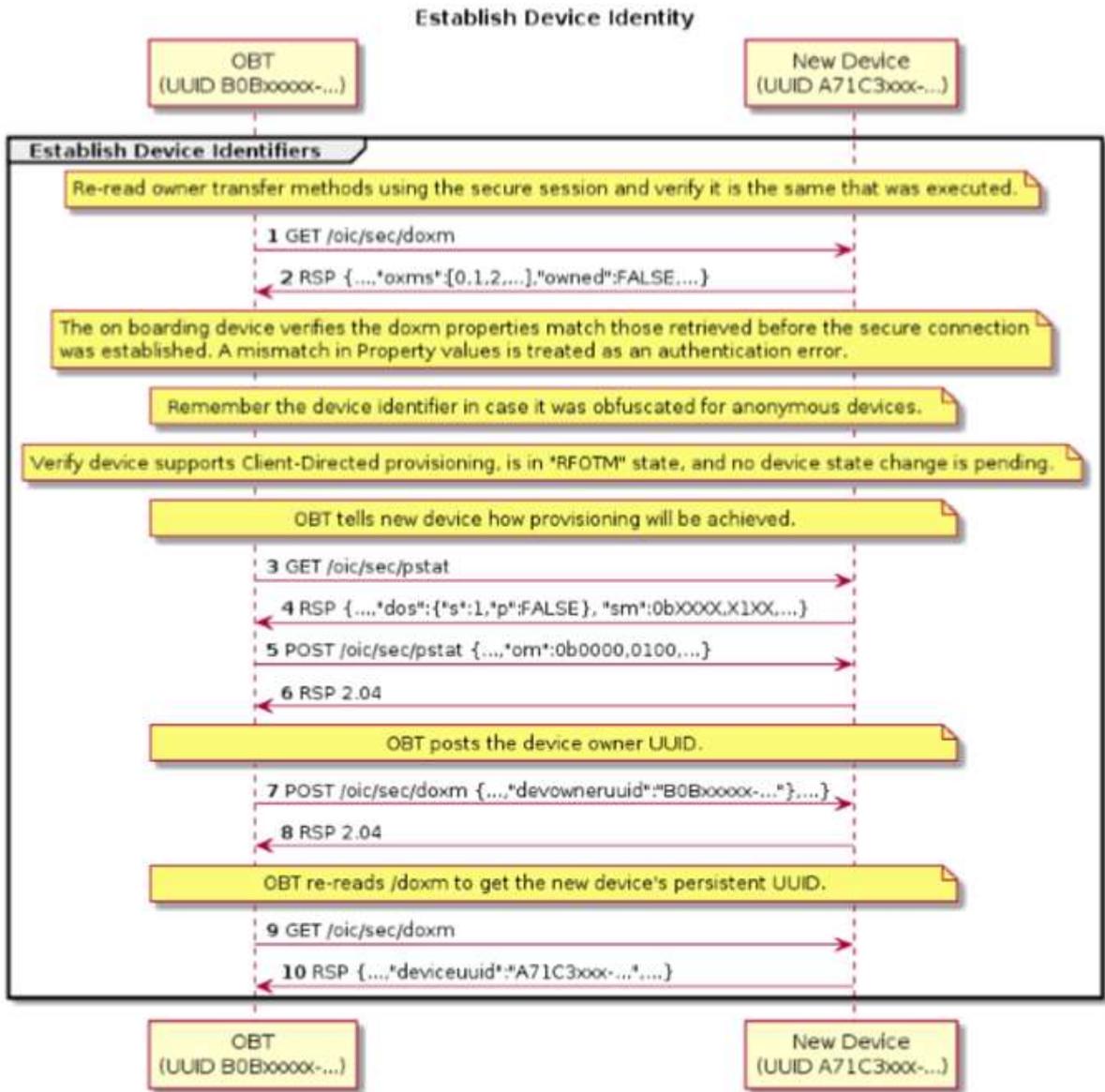


그림 18 - Establish Device Identity Flow

표 6 - Establish Device Identity Details

Step	Description
1, 2	The OBT obtains the doxm properties again, using the secure session. It verifies that these properties match those retrieved before the authenticated connection. A mismatch in parameters is treated as an authentication error.
3, 4	The OBT queries to determine if the Device is operationally ready to transfer Device ownership.
5, 6	The OBT asserts that it will follow the Client provisioning convention.
7, 8	The OBT asserts itself as the owner of the new Device by setting the Device ID to its ID.
9, 10	The OBT obtains doxm properties again, this time Device returns new Device persistent UUID.

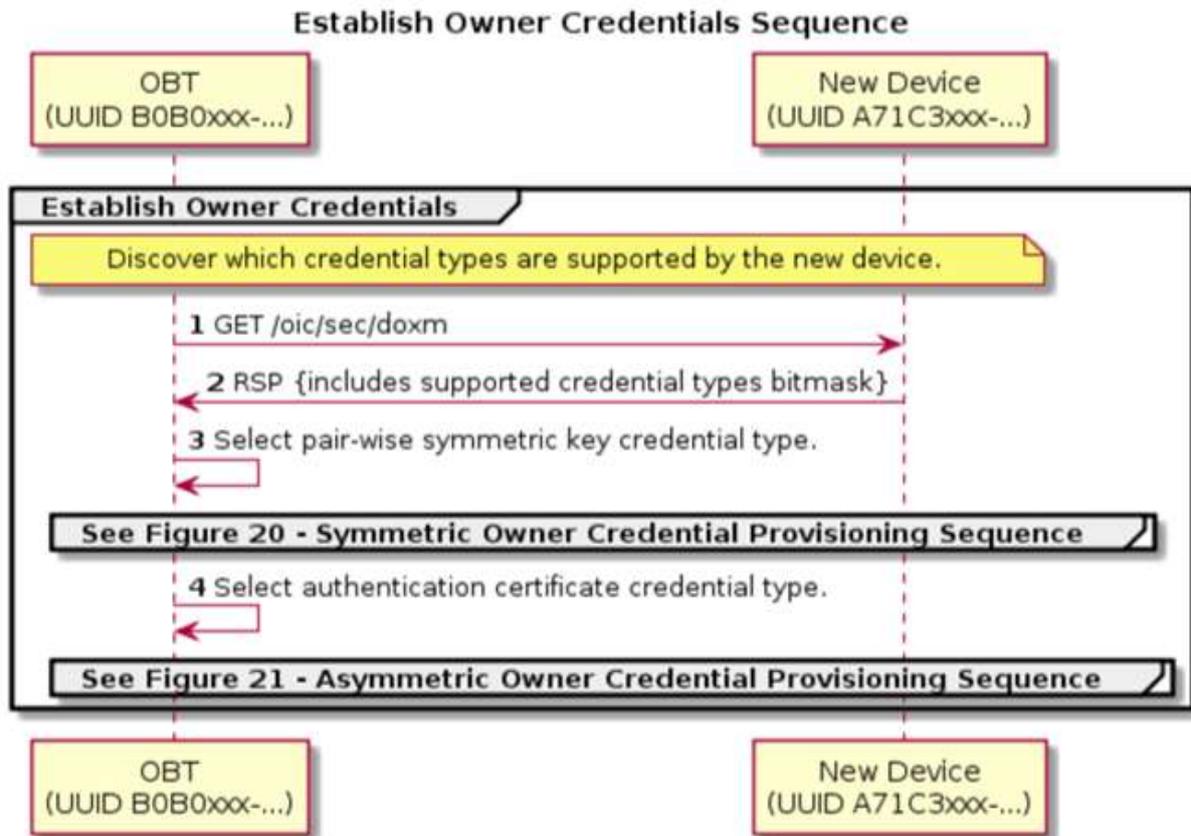


그림 19 - Owner Credential Selection Provisioning Sequence

표 7 - Owner Credential Selection Details

Step	Description
1, 2	The OBT obtains the doxm properties to check ownership transfer mechanism supported on the new Device.
3, 4	The OBT uses selected credential type for ownership provisioning.

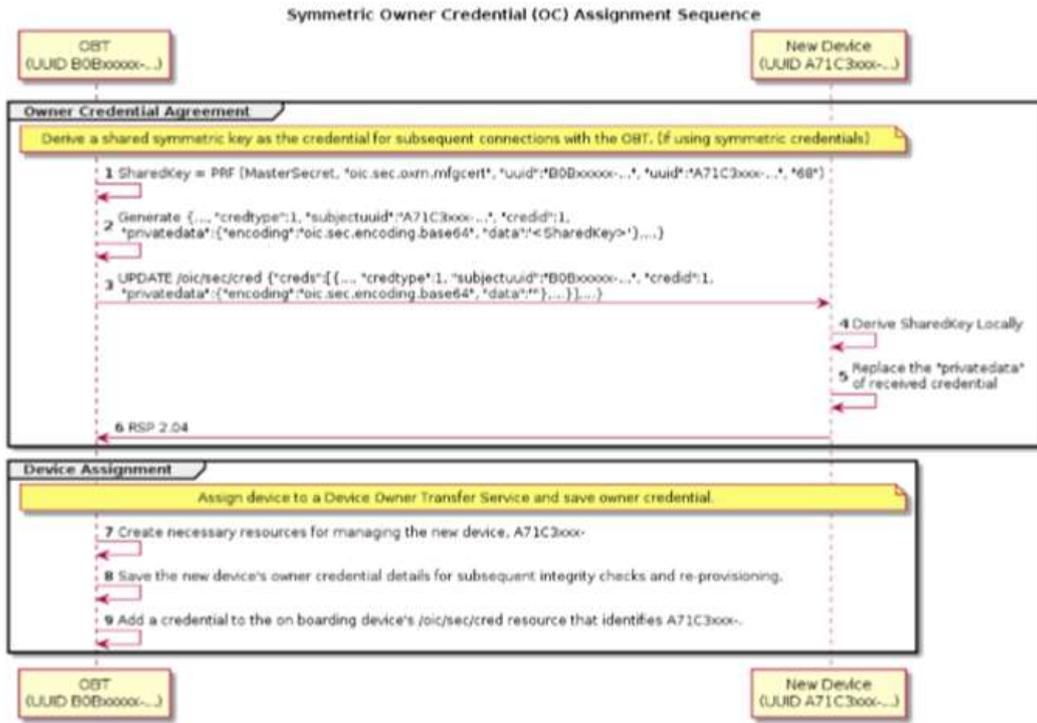


그림 20 - Symmetric Owner Credential Provisioning Sequence

표 8 - Symmetric Owner Credential Assignment Details

Step	Description
1, 2	The OBT uses a pseudo-random-function (PRF), the master secret resulting from the DTLS handshake, and other information to generate a symmetric key credential resource Property - SharedKey.
3	The OBT creates a credential resource Property set based on SharedKey and then sends the resource Property set to the new Device with empty "privatedata" Property value.
4, 5	The new Device locally generates the SharedKey and updates it to the "privatedata" Property of the credential resource Property set.
6	The new Device sends a success message.
7	The onboarding service creates a subjects resource for the new device (e.g./A71C3xxx-...)
8	The onboarding service provisions its "/oic/svc/dots/subjects/A71C3xxx-/cred" resource with the owner credential. Credential type is SYMMETRIC KEY.
9	(optional) The onboarding service provisions its own "/oic/sec/cred" resource with the owner credential for
	new device. Credential type is SYMMETRIC KEY.

특히 OBT가 대칭 소유자 자격 증명을 선택하는 경우 :

- OBT는 7.3.2에 설명된 공유키 자격 증명 계산 방법을 사용하여 공유 키를 생성해야 한다.
- OBT는 대칭 쌍별 키로 식별된 빈 키를 새 장치의 "/oic/sec/cred" 리소스에 보내야 한다.
- OBT의 대칭 소유자 자격 증명을 받으면 새 장치는 7.3.2에 설명된 SharedKey 자격 증명 계산 방법을 사용하여 공유 키를 독립적으로 생성하고 이를 소유자 자격 증명과 함께 저장해야 한다.
- 새 장치는 "/oic/sec/cred" 리소스를 통해 저장된 공유 키 소유자 자격 증명을 사용하여 후속 연결 중에 소유자를 인증해야 한다.

### Asymmetric Owner Credential (OC) Assignment Sequence

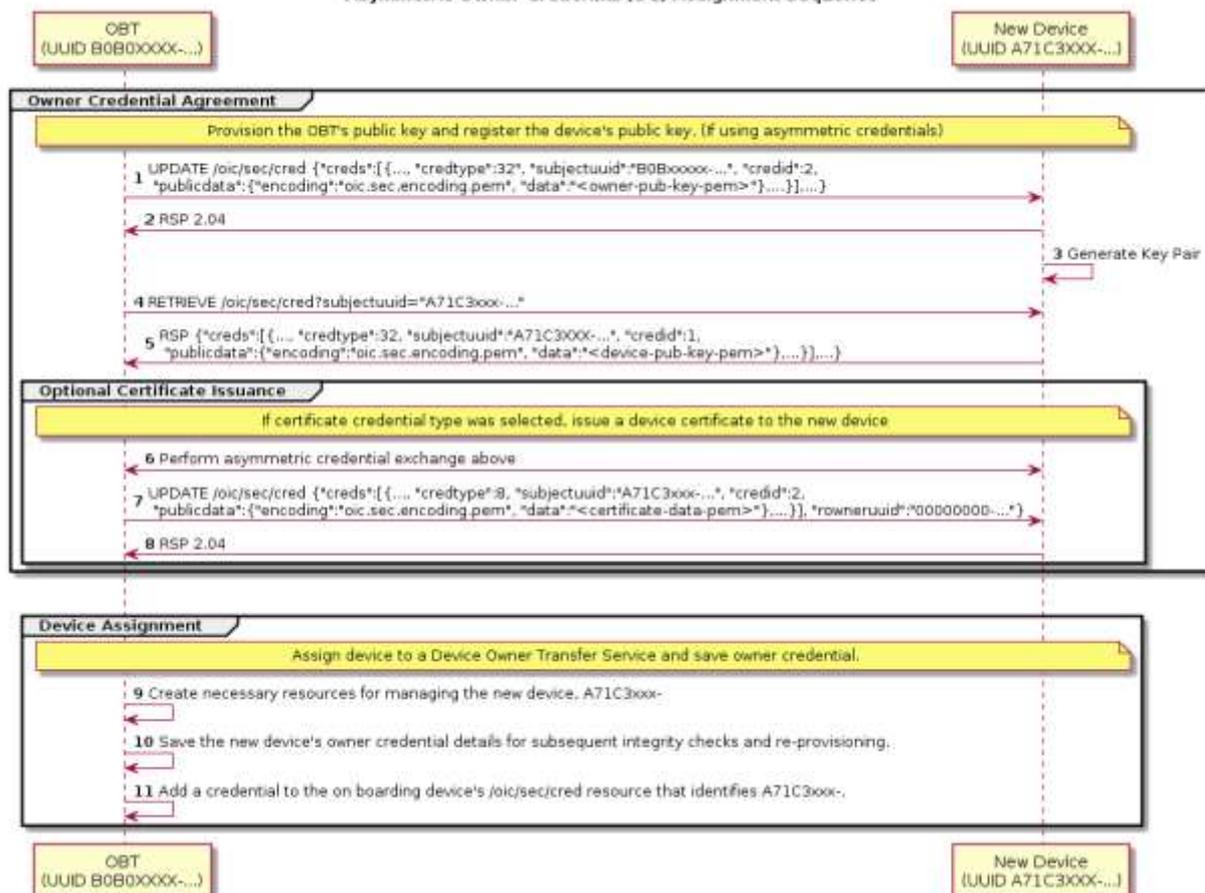


그림 21 - Asymmetric Owner Credential Provisioning Sequence

표 9 - Asymmetric Owner Credential Assignment Details

Step	Description
If an asymmetric or certificate owner credential type was selected by the OBT	
1, 2	The OBT creates an asymmetric type credential
3	The new Device creates an asymmetric key pair.
4, 5	The OBT reads the new Device's asymmetric type credential Resource Property set generated at step 25. It may be used subsequently to authenticate the new Device.
If certificate owner credential type is selected by the OBT	
6-8	The steps for creating an asymmetric credential type are performed. In addition, the OBT instantiates a newly-created certificate (or certificate chain) on the new Device.
9	The onboarding service creates a subjects resource for the new device (e.g./A71C3xxx-...)
10	The onboarding service provisions its "/oic/svc/dots/subjects/A71C3xxx-/cred" resource with the owner credential. Credential type is PUBLIC KEY.
11	(optional) The onboarding service provisions its own "/oic/sec/cred resource" with the owner credential for new device. Credential type is PUBLIC KEY.
12	(optional) The onboarding service provisions its own "/oic/sec/cred" resource with the owner credential for new device. Credential type is CERTIFICATE.

OBT가 비대칭 소유자 자격 증명을 선택한 경우 :

- OBT는 공개 키를 비대칭 암호화 키로 식별 된 새 장치의 "/oic/sec/cred" 리소스에 추가해야 한다.
- OBT는 SubjectID 쿼리 매개 변수를 통해 새 장치의 UUID를 제공하여 새 장치에서 "/oic/sec/cred" 리소스를 쿼리해야 한다. 이에 대한 응답으로, 새로운 장치는 공개 비대칭 암호화 키를 반환해야 하며, OBT는 새로운 장치의 향후 소유자 인증을 위해 유지해야 한다.

OBT가 인증서 소유자 자격 증명을 선택하는 경우 :

- OBT는 새 장치가 반환한 공개 키를 포함하는 리프 인증서를 사용하여 인증서 또는 인증서 체인을 생성해야 하며 상호 신뢰할 수 있는 CA가 서명하고 기정의된 인증서 자격 증명 생성 요구사항을 준수해야 한다.
- OBT는 새로 작성된 인증서 체인을 인증서가있는 비대칭 서명 키로 식별 된 "/oic/sec/cred" 리소스에 추가해야 한다.

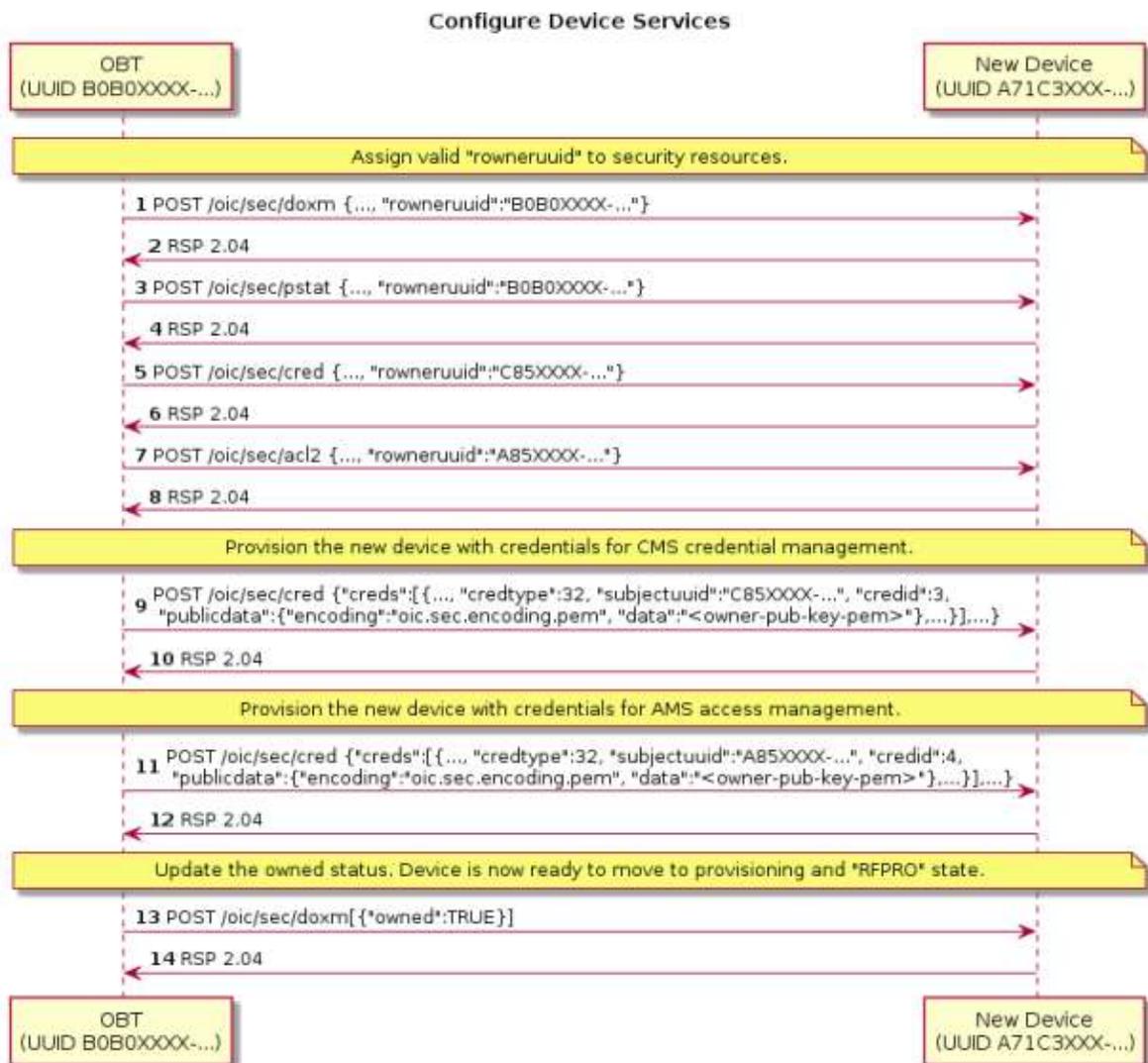


그림 22 - Configure Device Services

표 10 - Configure Device Services Details

Step	Description
1 - 8	The OBT assigns rowneruuid for different SVRs.
9 - 10	Provision the new Device with credentials for CMS
11 - 12	Provision the new Device with credentials for AMS
13 - 14	Update the "oic.sec.doxm.owned" to TRUE. Device is ready to move to provision and RFPRO state.

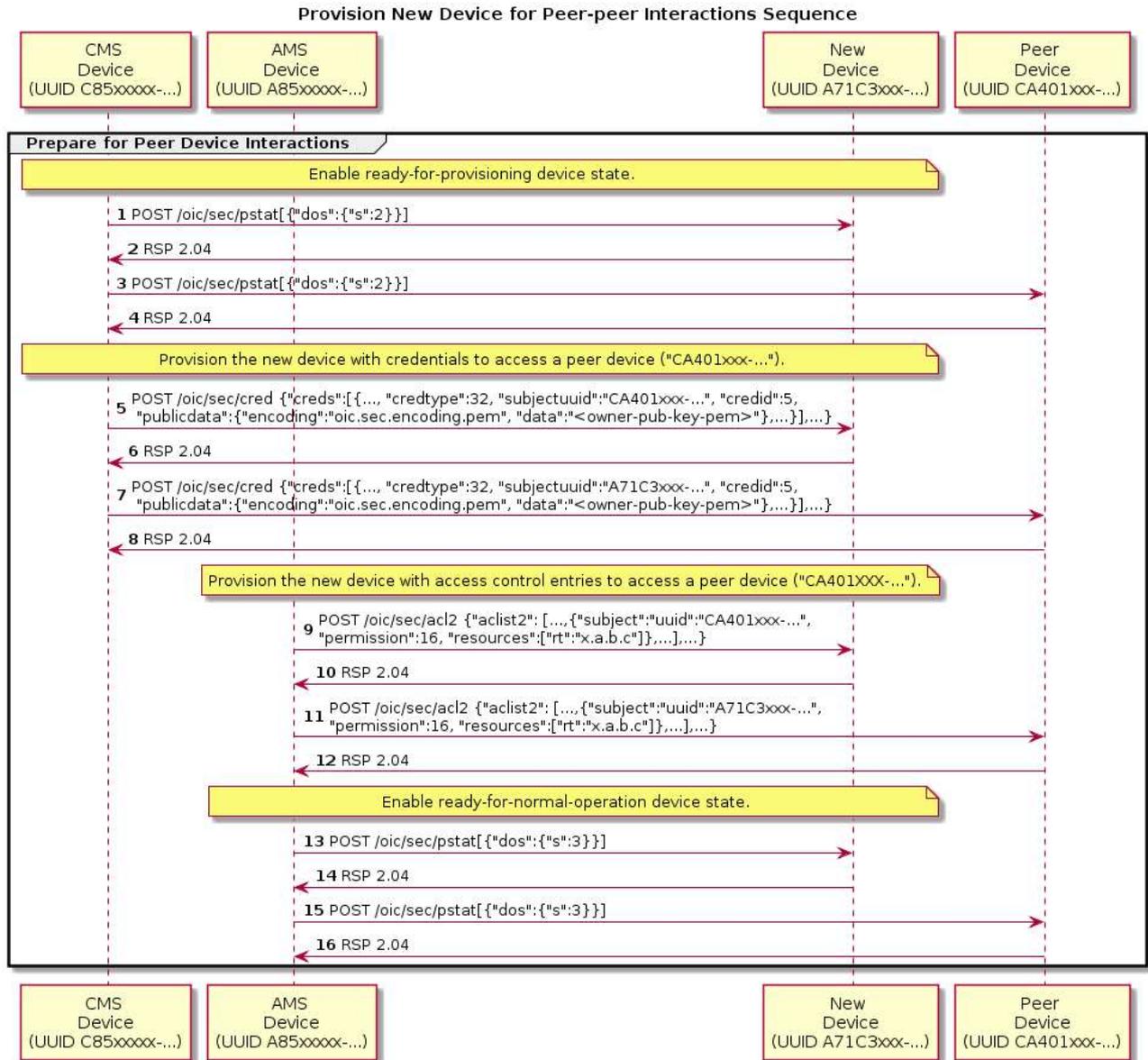


그림 23 - Provision New Device for Peer to Peer Interaction Sequence

표 11 - Provision New Device for Peer to Peer Details

Step	Description
1 - 4	The OBT set the Devices in the ready for provisioning status by setting "oic.sec.pstat.dos" to 2.
5 - 8	The OBT provision the Device with peer credentials
9 - 12	The OBT provision the Device with access control entities for peer Devices.
13 - 16	Enable Device to RFNOP state by setting "oic.sec.pstat.dos" to 3.

OBT 및/또는 OBT 운영자는 새 장치의 소유권을 양도 할 때 허용되는 OTM 목록에 대한 엄격한 요구 사항을 가질 수 있다. 이러한 요구 사항을 결정할 때 고려해야 할 몇 가지 요소는 다음과 같다.

- 각 OTM에 대해 설명 된 보안 고려 사항
- 소유권 이전을 수행하는 데 사용 된 환경에 MITM (Man-in-the-Middle) 공격자가 있을 가능성

예를 들어, OBT 운영자는 Onboard중인 모든 장치가 임의 PIN 또는 제조업체 인증서 OTM을 지원하도록 요구할 수 있다. 이러한 로컬 OTM 정책이 존재하는 경우, OBT는 위의 순서도에서 1 단계 동안 얻은 doxm 내용 (GET "/oic/sec/doxm")에 관계없이 해당 정책에 따라 허용되는 OTM 만 사용하려고 한다. 1 단계가 OBT와 장치 사이의 인증되지 않은 및/또는 암호화되지 않은 연결을 통해 수행되는 경우 GET 요청에 대한 응답 내용이 중간자 공격자에 의해 변조되었을 수 있다. 예를 들어, 새로운 장치가 지원하는 OTM 목록이 공격자에 의해 변경되었을 수 있다. 또한 MITM (Man-in-the-Middle) 공격자는 OBT와 새 장치 간의 DTLS 세션이 실패하도록 할 수 있다. 이러한 경우, 새로운 장치가 OBT에 의해 선택된 OTM을 지원하지 않거나 중간자 (man-in-the-middle)가 OBT 간의 통신에 이러한 실패를 주입했기 때문에 OBT는 세션 실패 여부를 판별 할 방법이 없다. 그리고 새로운 장치. 이 문서의 현재 버전은 OTM 정책과 관련된 디자인 및 사용자 경험을 OBT 구현 세부 사항으로 남겨 둔다.

OCF 장치는 OCF 보안 프로파일에 따라 평가되었을 수 있다. 제조업체의 인증서, OCF 웹 서버 또는 기타 공용 저장소에서 평가 결과에 액세스 할 수 있다. DOTS는 평가 결과를 검토하여 OCF 장치가 보유 할 권한이 있는 OCF 보안 프로파일을 판별하고 OCF 보안 도메인 소유자의 의도 된 세분화 전략에 가장 적합한 평가 된 보안 프로파일의 서브 세트로 디바이스를 구성한다. OCF 장치 공급 업체는 "/oic/sec/sp"리소스의 "supportedprofiles"속성에 대한 제조업체 기본값을 OCF의 테스트 및 인증 프로세스에서 승인 한 것과 일치하도록 설정해야 한다. "/oic/sec/sp"리소스의 "currentprofile"속성은 "supportedprofiles"에 포함 된 값 중 하나로 설정되어야 한다. 장치가 장치 상태 재설정으로 전환되면 제조업체 기본값이 다시 설정된다. OCF 장치는 장치가 RFOTM, RFPRO, SRESET 장치 상태 중 하나 인 경우 "/oic/sec/sp"리소스 만 업데이트 할 수 있어야 하며 보안 프로필의 지시에 따라 업데이트를 허용하지 않을 수 있다. DOTS는 "/oic/sec/sp"리소스의 "supportedprofiles"속성을 장치가 OCF Conformance 테스트의 일부로 달성 한 OCF 보안 프로필 값의 하위 집합으로 업데이트 할 수 있다. DOTS는 값이 "oic.sec.cred.mfgcert"인 "/oic/sec/cred"리소스의 "credusage"속성을 선택하여 OCF 장치와 함께 제공된 제조업체 인증서를 검사하여 적합성 결과를 찾을 수 있다. DOTS는 ocfCPLAttributes 확장 필드에 해당하는 잘 알려진 OCF 웹 사이트 URI를 방문하여 적합성 결과를 더 찾을 수 있다. DOTS는 로컬 정책에 기초하여 보안 프로파일의 서브 세트 (OCF 적합성 테스트에 의해 평가 된 것으로부터)를 선택할 수 있다. OTS 세션이 활성화되어있는 동안 Onboarding의 일부로 DOTS는 Onboarding 후 DOTS 액세스를 허용하도록 ACE 항목을 구성해야 한다.

DOTS는 "/oic/sec/sp"리소스의 "currentprofile"속성을 OCF 보안 도메인 소유자가 의도 한 장치 배포 전략을 가장 정확하게 나타내는 값으로 업데이트해야 한다. CMS는 보안 프로필 값 (예 : "sp-blue-v0 OID")을 사용하여 역할 자격 증명을 발급하여 OCF 보안 도메인 소유자가 보안 프로필에 따라 OCF 보안 도메인을 분할하려는 의도를 나타낸다. CMS는 "/oic/sec/sp"리소스의 supportedprofiles 속성을 검색하여 역할 자격 증명을 발급 할 때 장치에서 지원되는 보안 프로필로 검증 된 역할 이름을 선택한다. CMS가 보안 프로필을 기반으로 역할 자격 증명을 발급하면 AMS는 역할 지정이 포함 된 액세스 제어 항목을 제공한다.

새 장치를 Onboarding하는 과정에서 새 장치와 OBT 사이에 보안 채널이 형성된다. 장치 소유권 상태가 "소유"로 변경되면 프로비저닝을 시작할 수 있다. OBT는 향후 새로운 장치 관리 방법을 결정하고 이후에 장치 프로비저닝 및 진행중인 장치 관리를 완료하는 데 사용해야 하는 지원 서비스를 제공한다. 장치는 서버 지향 또는 클라이언트 지향 프로비저닝 전략을 사용한다. "/oic/sec/pstat"리소스는 프로비저닝 전략 및 현재 프로비저닝 상태를 식별한다. 프로비저닝 서비스는 OCF 보안 도메인에 가장 적합한 프로비저닝 전략을 결정해야 한다.

클라이언트 지향 프로비저닝은 프로비저닝이 필요한 서버를 식별 한 다음 필요한 모든 프로비저닝 의무를 수행하는 프로비저닝 서비스에 의존한다. 클라이언트 지향 프로비저닝의 예는 다음 그림과 표에 설명된다.

OCF Client-directed Provisioning  
with a Single Service Provider

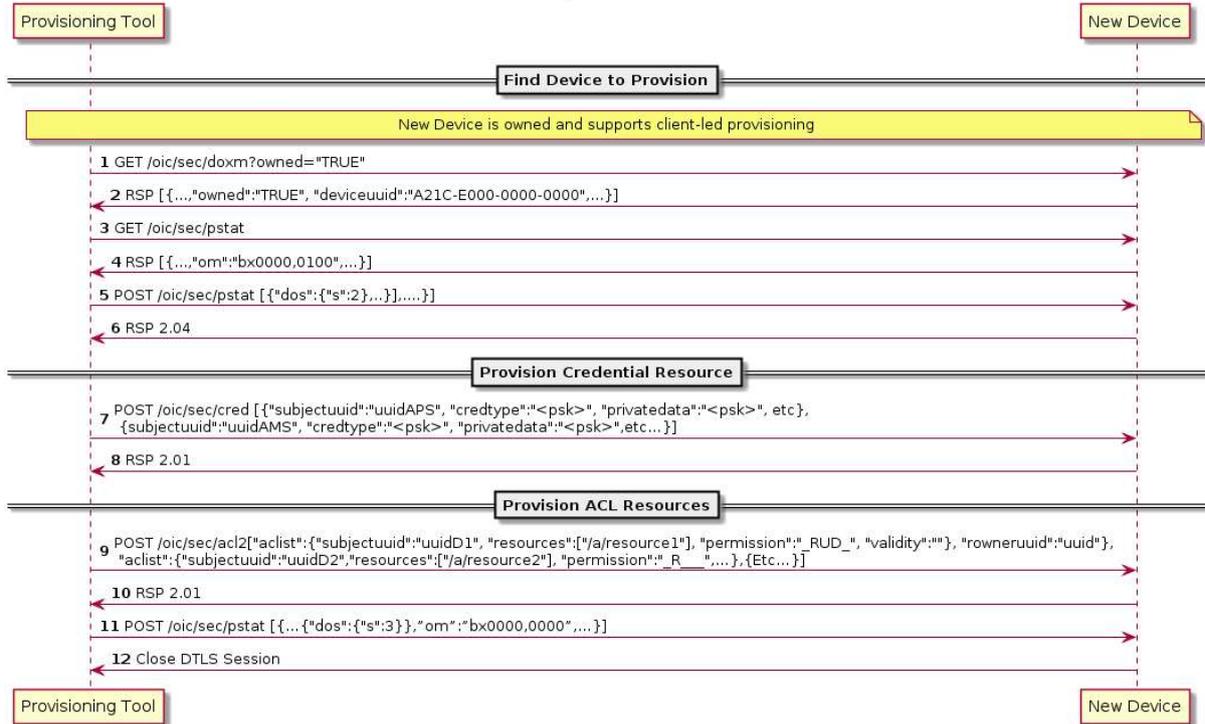


그림 24 - Example of Client-directed provisioning

표 12 - Steps describing Client -directed provisioning

Step	Description
1	Discover Devices that are owned and support Client-directed provisioning.
2	The "/oic/sec/doxm" Resource identifies the Device and it's owned status.
3	Provisioning Tool (PT) obtains the new Device's provisioning status found in "/oic/sec/pstat" Resource
4	The "pstat" Resource describes the types of provisioning modes supported and which is currently configured. A Device manufacturer should set a default current operational mode ("om"). If the "om" isn't configured for Client-directed provisioning, its "om" value can be changed.
5 - 6	Change Device state to Ready-for-Provisioning.
7 - 8	PT instantiates the "/oic/sec/cred" Resource. It contains credentials for the provisioned services and other Devices
9 - 10	PT instantiates "/oic/sec/acl2" Resource.
11	The new Device provisioning status mode is updated to reflect that ACLs have been configured. (Ready-for-Normal-Operation state)
12	The secure session is closed.

서버 지향 프로비저닝은 많은 프로비저닝 작업을 지시하기 위해 서버 (즉, 새 장치)에 의존한다. Onboarding 프로세스의 일부로 서버에서 추가 프로비저닝을 찾는 데 사용하는 지원 서비스가 프로비저닝 된다. 새로운 장치는 자체 프로비저닝 된 상태 중심 접근 방식을 사용하여 현재 프로비저닝 상태를 분석하고 대상 상태로 나아가려고 한다. 이 예에서는 단일 지원 서비스를 사용하여 새 장치를 프로비저닝 한다고 가정한다. 클라이언트 지향 프로비저닝의 예는 다음 그림과 표에 설명된다.

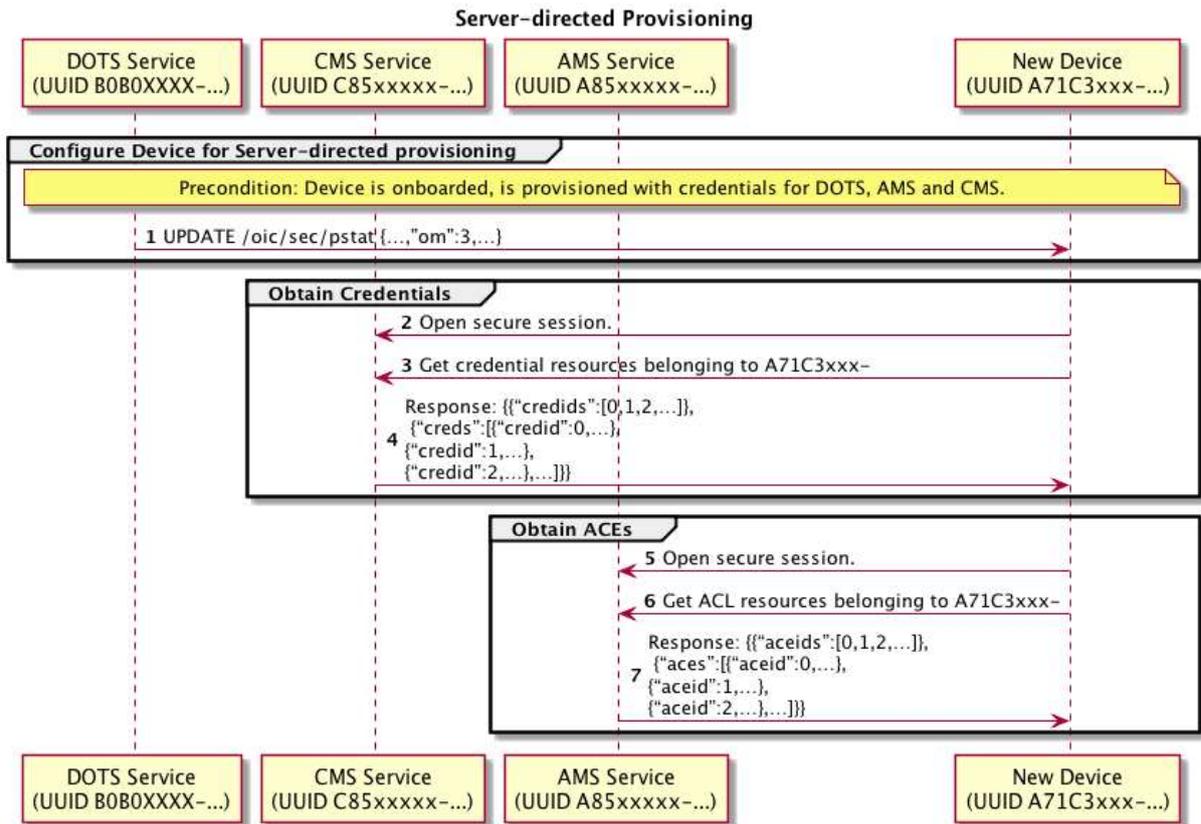


그림 25 - Example of Server-directed provisioning using a single provisioning service

표 13 - Steps for Server-directed provisioning using a single provisioning service

Step	Description
1	The new Device verifies it is owned.
2	The new Device verifies it is in self-provisioning mode.
3	The new Device verifies its target provisioning state is fully provisioned.
4	The new Device verifies its current provisioning state requires provisioning.
5	The new Device initiates a secure session with the provisioning tool using the "/oic/sec/doxm". DevOwner value to open a TLS connection using SharedKey.
8 - 9	The new Devices gets the "/oic/sec/cred" Resources. It contains credentials for the provisioned services and other Devices.
11 - 12	The new Device gets the "/oic/sec/aci2" Resource.
14	The secure session is closed.

8 - 9	The new Devices gets the "/oic/sec/cred" Resources. It contains credentials for the provisioned services and other Devices.
11 - 12	The new Device gets the "/oic/sec/aci2" Resource.
14	The secure session is closed.

여러 지원 서비스가 포함 된 서버 지향 프로비저닝 흐름은 여러 지원 서비스에 프로비저닝 작업을 분산 시킨다. 여러 지원 서비스를 사용하면 프로비저닝 워크로드를 분산하거나 전문화된 지원을 배포할 수 있다. 그림의 예는 프로비저닝 도구를 사용하여 CMS와 AMS의 두 가지 지원 서비스를 구성하는 방법을 보여준다. 예제 단계는 표에 설명되어 있다.

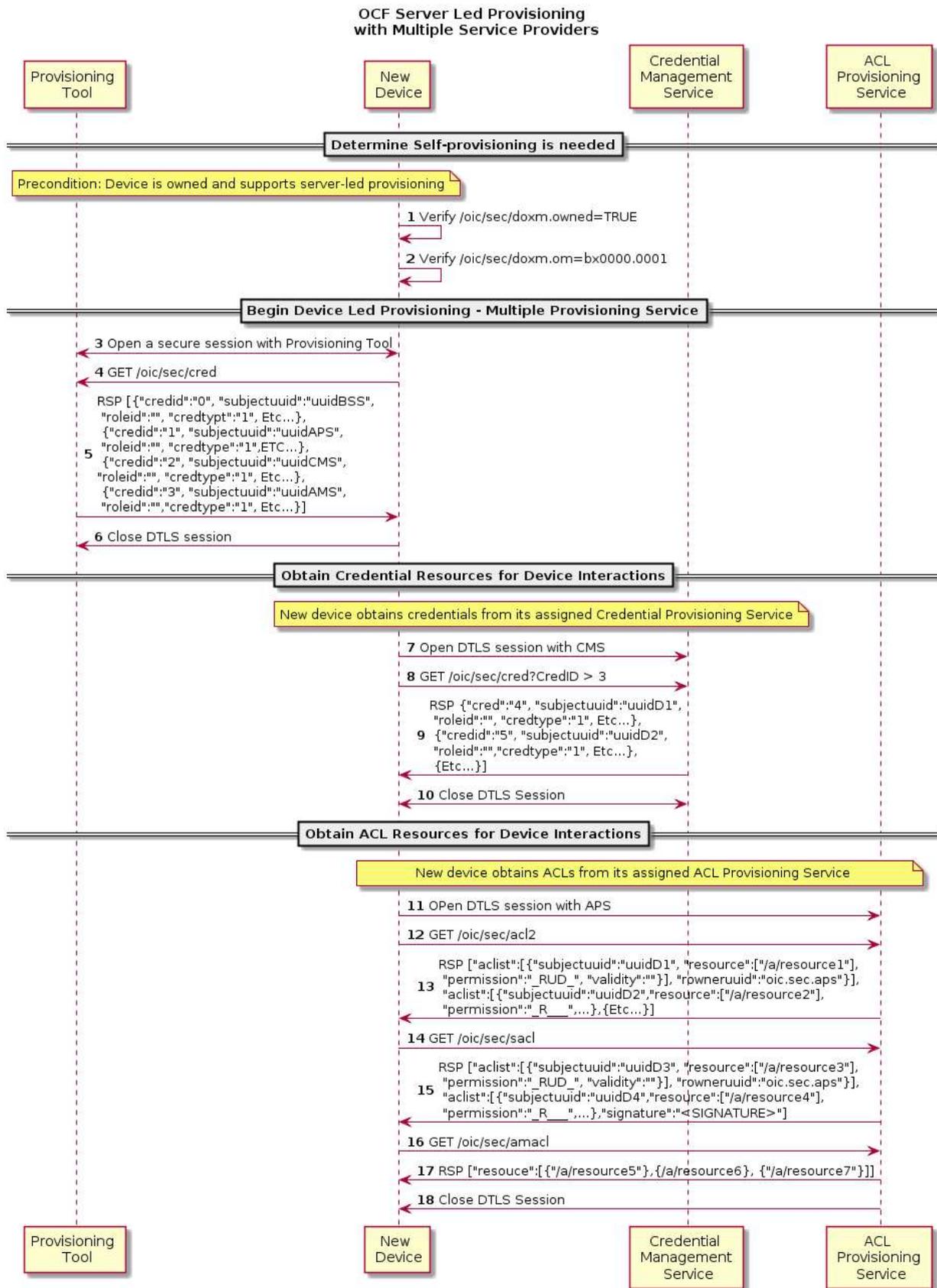


그림 26 - Example of Server-directed provisioning involving multiple support services

표 14 - Steps for Server-directed provisioning involving multiple support services

Step	Description
1	The new Device verifies it is owned.
2	The new Device verifies it is in self-provisioning mode.
3	The new Device initiates a secure session with the provisioning tool using the "/oic/sec/doxm". DevOwner value to open a TLS connection using SharedKey.
4-5	The new Device gets credentials Resource for the provisioned services and other Devices
6	The new Device closes the DTLS session with the provisioning tool.
7	The new Device finds the CMS from the "/oic/sec/cred" Resource, rowneruuid Property and opens a DTLS connection. The new device finds the credential to use from the "/oic/sec/cred" Resource.
8-9	The new Device requests additional credentials that are needed for interaction with other devices.
10	The DTLS connection is closed.
11	The new Device finds the ACL provisioning and management service from the "/oic/sec/acl2" Resource, rowneruuid Property and opens a DTLS connection. The new device finds the ACL to use from the "/oic/sec/acl2" Resource.
12-13	The new Device gets ACL Resources that it will use to enforce access to local Resources.
14-15	The new Device should get SACL Resources immediately or in response to a subsequent Device Resource request.
16-17	The new Device should also get a list of Resources that should consult an Access Manager for making the access control decision.
18	The DTLS connection is closed.

OCF 클라우드에 연결하는 장치는 OCF 클라우드에서 "/oic/sec/account", "/oic/sec/session", "/oic/sec/tokenrefresh"의 SVR에 이어 장치에 있는 "oic.r.coapcloudconf" 리소스를 지원해야 한다.

OCF 클라우드는 중재자를 OCF 클라우드 사용자와 연결하기 위한 보안 메커니즘을 사용해야 한다. 메커니즘 선택은 OCF 클라우드에 달려 있다. 예를 들어, 메커니즘에는 HTTP 인증 (사용자 이름 및 비밀번호 사용) 또는 OAuth 2.0 (OCF Cloud 제공 업체 또는 타사에서 운영 할 수 있는 권한 부여 서버 사용)이 포함된다. OCF 클라우드는 OCF 클라우드 사용자를 인증하는 데 적합한 인증 메커니즘을 사용해야 한다.

중재자와 장치는 보안 세션을 사용하여 장치가 OCF 클라우드에 연결되도록 프로비저닝해야 한다. 중재자는 OCF 클라우드 사양에 설명된대로 OCF 클라우드에서 액세스 토큰을 얻는다. 그런 다음이 액세스 토큰은 5.4.6.5에 설명된대로 장치에서 OCF 클라우드에 등록하는 데 사용된다.

OCF 클라우드는 액세스 토큰 및 중재자가 제공 한 장치 ID가 저장된 맵을 유지 관리한다. 장치 등록시 OCF Cloud는 액세스 토큰의 유효성을 검사하고 TLS 세션을 해당 장치 ID와 연결한다. 중재자는 OCF 클라우드 사양에 설명된대로 장치를 프로비저닝 한다. 중재자는 "oic.r.coapcloudconf" 리소스의 "cis" 속성에 OCF Cloud URI를, "oic.r.coapcloudconf" 리소스의 "sid" 속성에 OCF Cloud UUID를 제공하고 "at "장치의"oic.r.coapcloudconf" 리소스 속성. 프로비저닝 된 "at"는 장치에서 IETF RFC 6750에 정의된 "베어러"토큰 유형의 액세스 토큰으로 처리된다.

액세스 제어를 위해 장치는 공통 이름의 OCF Cloud UUID를 사용하여 OCF 클라우드를 식별해야 한다. OCF 클라우드를 인증하는데 사용되는 End-Entity 인증서의 필드이다. AMS는 중재자가 "oic.r.coapcloudconf" 리소스에 대한 UPDATE 권한이있는 유일한 장치가 되도록 장치에서 ACE2 항목을 구성해야 한다.

AMS는 OCF 클라우드의 요청을 허용하도록 장치에서 ACE2 항목을 구성해야한다. 중재자의 요청에 따라 AMS는 이전 OCF Cloud UUID를 사용하여 이전 ACL2 항목을 제거한다. 이 요청은 중재자가 새 OCF 클라우드에 대해 "oic.r.coapcloudconf"를 구성하기 전에 발생한다. 중재자는 또한 AMS에 OCF Cloud UUID를 새 ACL2 항목의 "대상" 속성으로 설정하도록 요청한다. AMS는 "oic.r.coapcloudconf" 리소스의 "sid" 속성을 현재 OCF Cloud UUID로 사용할 수 있다. AMS는 OCF 클라우드에 대한 와일드카드 항목을 프로비저닝 하거나 장치에 게시된 각 리소스를 나열하는 항목을 프로비저닝 할 수 있다.

OCF Cloud가 장치 등록 중에 응답으로 "redirecturi" 값을 제공하는 경우 리디렉션 된 OCF Cloud는 동일한 OCF Cloud UUID를 가지며 동일한 트러스트 앵커를 사용하는 것으로 간주된다. 그렇지 않으면 제시된 OCF Cloud UUID가 프로비저닝 된 ACL2 항목과 일치하지 않다. 중재자는 다음 표의 속성으로 "oic.r.coapcloudconf" 리소스를 프로비저닝 해야 한다. 프로비저닝 된 세부 정보는 디바이스가 OCF 클라우드에 디바이스 등록을 수행하기 위해 디바이스에서 사용한다.

초기 등록 후 장치는 대신 OCF 클라우드에서 수신 한 업데이트 된 값을 사용해야 한다. OCF 클라우드 사용자가 장치가 OCF 클라우드에 다시 등록하기를 원하는 경우 중재자를 사용하여 "oic.r.coapcloudconf" 리소스를 새로운 값으로 다시 프로비저닝 할 수 있다.

표 15 - Mapping of Properties of the "oic.r.account" and "oic.r.coapcloudconf" Resources

Property Name	oic.r.coapcloudconf	oic.r.account	Description
Authorization Provider Name	apn	authprovider	The Authorization Provider through which Access Token was obtained.
OCF Cloud URL	cis	-	This is the URL connection is established between Device and OCF Cloud.
Access Token	at	accesstoken	The unique token valid only for the Device.
OCF Cloud UUID	sid	-	This is the identity of the OCF Cloud that the Device is configured to use.

(라) 장치 Onboarding 상태 정의

■ 장치 Onboarding 일반

Onboarding 프로세스는 장치의 소유권이 양도되고 관련 구성/서비스가 장치에 프로비저닝 된 후 완료 된다. 그림은 장치 수명주기 동안 장치가 있을 수 있는 다양한 상태를 보여줍니다. "/pstat.dos.s"속성은 "/oic/sec/pstat"리소스 소유자 (예 : "doxs"서비스)에 의해 RW이므로 리소스 소유자는 원격으로 장치 상태를 업데이트 할 수 있다. 장치가 RFNOP 또는 RFPRO 인 경우 ACL을 사용하여 다른 장치에 의한 장치 상태를 원격으로 제어 할 수 있다. 장치 상태가 SRESET 인 경우 장치 OC는 장치 액세스 권한의 유일한 표시 일 수 있다. 장치 소유자는 장치의 RFPRO 전환에 적합한 저수준 일관성 검사 및 재 프로비저닝을 수행 할 수 있다.

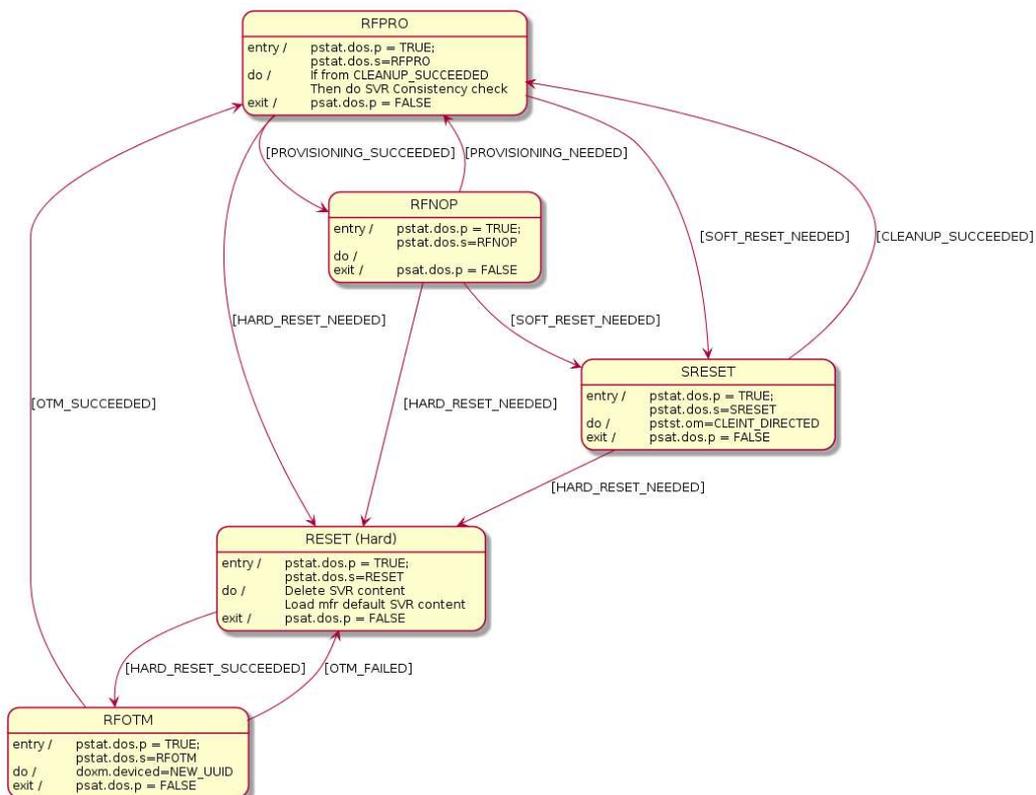


그림 27 - Device state model

다이어그램에서 볼 수 있듯이 프로비저닝 단계가 끝나면 장치는 "정상 작동 준비"상태가 되어 다른 장치와의 상호 운용을 시작하는 데 필요한 모든 것을 갖다. 8.5 절은 장치가 "정상 작동 준비"로 간주되기 위해 보유해야 하는 최소 필수 구성을 명시한다. 정전 또는 장치 고장이 발생한 경우, 장치는 정전/실패 이전과 동일한 상태를 유지해야 한다.

장치 또는 리소스 소유자가 "/pstat.dos.s"를 OBS하면 SRESET으로 전환하면 SVR 일관성 검사가 필요할 수 있는 장치에 대한 조기 경고 알림이 제공된다.

Onboarding이 작동하려면 장치에 다음 리소스가 설치되어 있어야 한다.

- 1) "/oic/sec/doxm" 리소스
- 2) "/oic/sec/pstat" 리소스
- 3) "/oic/sec/cred" 리소스

이 자료에 포함 된 값은 8.2, 8.3, 8.4, 8.5 및 8.6의 상태 정의에 지정되어 있다.

/pstat.dos.s = RESET 상태는 제조업체 기본값으로 "하드"재설정으로 정의된다. 하드 리셋은 또한 장치 자산을 다른 당사자에게 전송할 준비가 된 상태를 정의한다. 플랫폼 제조업체는 플랫폼을 강제로 재설정하는 물리적 메커니즘 (예 : 버튼)을 제공해야 한다. 동일한 플랫폼에서 호스팅되는 모든 장치는 플랫폼 재설정이 주장 될 때 장치 상태를 RESET으로 전환한다. 다음의 자료와 그 특정 속성은 명시된 값을 가져야 한다.

- 1) "/oic/sec/doxm" 리소스의 "소유" 속성은 FALSE로 전환된다.
- 2) "/oic/sec/doxm" 리소스의 "devowneruuid" 속성은 UUID가 아닙니다.
- 3) "/oic/sec/doxm" 리소스의 "소유자" 속성은 이 속성이 구현된 경우 UUID가 아닙니다.
- 4) "/oic/sec/doxm" 리소스의 "deviceuuid" 속성은 제조업체 기본값으로 설정해야 한다.
- 5) "/oic/sec/doxm" 리소스의 "deviceid" 속성은 이 속성이 구현된 경우 제조업체의 기본값으로 재설정된다.
- 6) "/oic/sec/doxm" 리소스의 "sct" 속성은 제조업체의 기본값으로 재설정된다.
- 7) "/oic/sec/doxm" 리소스의 "oxmsel" 속성은 제조업체의 기본값으로 재설정된다.
- 8) "/oic/sec/pstat" 리소스의 "isop" 속성은 FALSE이다.
- 9) "/oic/sec/pstat" 자원의 "dos" 속성이 업데이트되어야 한다 : dos.s는 "RESET" 상태와 같고 dos.p는 "FALSE"와 같아야 한다.
- 10) "/oic/sec/pstat" 리소스의 "om"(작동 모드) 속성은 제조업체 기본값으로 설정해야 한다.
- 11) "/oic/sec/pstat" 리소스의 "sm"(지원되는 작동 모드) 속성은 제조업체 기본값으로 설정해야 한다.
- 12) "/oic/sec/pstat", "/oic/sec/doxm", "/oic/sec/acl2" 및 "/oic/sec/cred" 리소스의 "rowneruuid" 속성은 UUID가 아닙니다.
- 13) "/oic/sec/sp" 리소스의 "supportedprofiles" 속성은 제조업체 기본값으로 설정해야 한다.
- 14) "/oic/sec/sp" 리소스의 "currentprofile" 속성은 제조업체 기본값으로 설정해야 한다.

### ■ 장치 준비 OTM 상태 정의

다음 리소스 및 해당 특정 속성은 장치가 소유권 이전 준비 상태가 되면 지정된 값을 갖다.

- 1) "/oic/sec/doxm" 리소스의 "소유" 속성은 FALSE 여야 하며 TRUE로 전환된다.
- 2) "/oic/sec/doxm" 리소스의 "소유자" 속성은 이 속성이 구현된 경우 UUID가 아닙니다.
- 3) "/oic/sec/doxm" 리소스의 "devowneruuid" 속성은 UUID가 아닙니다.
- 4) 이 속성이 구현된 경우 "/oic/sec/doxm" 리소스의 "deviceid" 속성이 UUID가 될 수 없다. "/oic/d"의 di 속성값은 정의되어 있지 않다.
- 5) "/oic/sec/doxm" 리소스의 "deviceuuid" 속성은 제조업체 기본값으로 설정해야 한다.
- 6) "/oic/sec/pstat" 리소스의 "isop" 속성은 FALSE이다.
- 7) "/oic/sec/pstat" 리소스의 "dos"는 업데이트 되어야 한다 : "dos.s"는 "RFOTM" 상태와 같고 dos.p는 "FALSE"와 같다.
- 8) "/oic/sec/cred" 리소스는 선택된 OTM이 요구하는 경우 자격 증명을 포함해야 한다.

다음 리소스 및 해당 특정 속성은 장치가 프로비저닝 준비가 되면 입력 한 값을 가져야 한다.

- 1) "/oic/sec/doxm" 리소스의 "소유" 속성은 TRUE이다.
- 2) "/oic/sec/doxm" 리소스의 "devowneruuid" 속성은 UUID가 아닙니다.
- 3) "/oic/sec/doxm" 리소스의 "deviceuuid" 속성은 UUID가 아니어야 하며 RFOTM 처리 중에 결정된 값으로 설정되어야 한다. 또한 "/oic/d" 자원의 "di" 특성 값은 "/oic/sec/doxm" 자원의 "deviceid" 특성과 동일해야 한다.

- 4) "/oic/sec/doxm" 리소스의 "oxmsel" 속성은 소유권 이전 중에 사용된 실제 OTM 값을 가져야 한다.
- 5) "/oic/sec/pstat" 리소스의 "isop" 속성은 FALSE이다.
- 6) "/oic/sec/pstat" 리소스의 "dos"는 업데이트되어야 한다. "dos.s"는 "RFPRO" 상태와 같고 "dos.p"는 "FALSE"와 같다.
- 7) 설치된 모든 자원의 "rowneruuid" 속성은 유효한 자원 소유자 (즉, 주어진 자원을 인스턴스화 하거나 업데이트할 권한이 있는 엔티티)로 설정해야 한다. "rowneruuid"를 설정하지 않으면 고아 자원이 발생할 수 있다.
- 8) "/oic/sec/cred" 리소스는 "rowneruuid" 및 "devowneruuid" 속성

다음의 자료와 그 특정 속성은 장치가 정상 작동 준비 상태가 될 때 지정된 값을 가져야 한다.

- 1) "/oic/sec/doxm" 리소스의 "소유" 속성은 TRUE이다.
- 2) "/oic/sec/doxm" 리소스의 "devowneruuid" 속성은 UUID가 아닙니다.
- 3) "/oic/sec/doxm" 리소스의 "deviceuuid" 속성은 UUID가 아니어야 하며 OTM 중에 구성된 ID로 설정해야 한다. 또한 "/oic/d"의 "di" 속성 값은 deviceuuid와 동일해야 한다.
- 4) "/oic/sec/doxm" 리소스의 "oxmsel" 속성은 소유권 이전 중에 사용된 실제 OTM 값을 가져야 한다.
- 5) "/oic/sec/pstat" 리소스의 "isop" 속성은 RFNOP로의 전환이 완료되면 서버에서 TRUE로 설정해야 한다.
- 6) "/oic/sec/pstat" 리소스의 "dos"는 업데이트되어야 한다. "dos.s"는 "RFNOP" 상태와 같고 dos.p는 "FALSE"와 같다.
- 7) 설치된 모든 자원의 "rowneruuid" 속성은 유효한 자원 소유자 (즉, 주어진 자원을 인스턴스화 하거나 업데이트할 권한이 있는 엔티티)로 설정해야 한다. "rowneruuid"를 설정하지 않으면 고아 자원이 발생한다.
- 8) "/oic/sec/cred" 리소스는 "rowneruuid" 및 "devowneruuid" 속성

소프트 리셋 상태는 정의되어 있다 (예 : "/pstat.dos.s"= SRESET).이 상태로 진입하면 장치가 작동하지 않지만 현재 소유자가 소유하고 있음을 의미한다. 장치는 원래 Onboarding 중에 제공된 OC를 사용하여 DOTS (예 : "rt"= "oic.r.doxs")를 인증하여 SRESET을 종료 할 수 있다 (단, OTM/doxm.oxms를 사용할 필요는 없음).

DOTS는 SVR의 일관성 검사를 수행해야하며 필요한 경우 장치가 RFPRO로 전환 될 수 있도록 충분히 다시 프로비저닝 해야 한다.

그림은 SRESET의 OBT Sanity Check Sequence를 나타낸다.

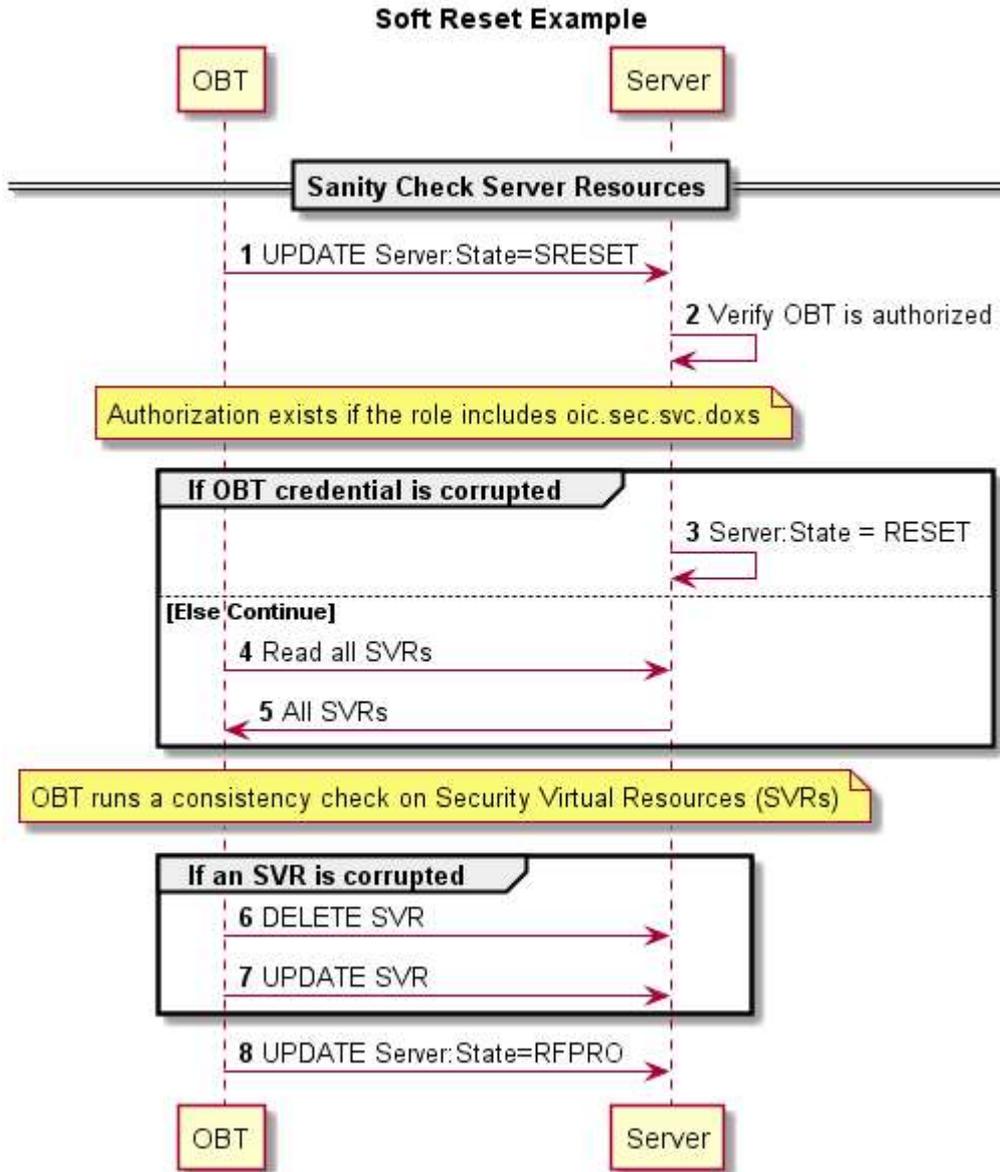


그림 28- OBT Sanity Check Sequence in SRESET

DOTS는 RFPRO 장치 상태로 최종 전환하기 전에 SVR의 상태 점검을 수행해야한다. DOTS 자격 증명을 찾을 수 없거나 손상된 것으로 판단되면 장치 상태가 재설정으로 전환된다. DOTS 자격 증명에 DOTS의 유효성을 검사하지 못하면 장치는 SRESET 상태로 유지되어야한다. 이는 비 DOTS 장치가 시도 할 수 있는 서비스 거부 공격을 완화한다. SRESET에있을 때, 다음 자료와 그 특정 속성은 지정된 값을 가져야한다.

- 1) "/oic/sec/doxm" 리소스의 "소유" 속성은 TRUE이다.
- 2) "/oic/sec/doxm" 리소스의 "devowneruid" 속성은 null이 아닙니다.
- 3) "/oic/sec/doxm" 리소스의 "소유자" 속성은 이 속성이 구현된 경우 null이 아닙니다.
- 4) "/oic/sec/doxm" 리소스의 "deviceuid" 속성은 null이 아닌 상태로 유지된다.
- 5) "/oic/sec/doxm" 리소스의 "deviceid" 속성은 null이 아닙니다.
- 6) "/oic/sec/doxm" 리소스의 "sct" 속성은 그 값을 유지해야 한다.
- 7) "/oic/sec/doxm" 리소스의 "oxmsel" 속성은 그 값을 유지해야 한다.
- 8) "/oic/sec/pstat" 리소스의 "isop" 속성은 FALSE이다.
- 9) "/oic/sec/pstat.dos.s" 속성은 SRESET이다.
- 10) "/oic/sec/pstat" 리소스의 "om"(작동 모드) 속성은 "clientdirected mode"이다.
- 11) "sm"(지원되는 작동 모드) "/oic/sec/pstat" 리소스의 속성은 장치 소유자 (일명 DOTS)에 의해 업데이트 될 수 있다.
- 12) "/oic/sec/pstat", "/oic/sec/doxm", "/oic/sec/acl2", "/oic/sec/amacl", "/oic/sec/의 "rowneruid" 속성 및 "/oic/sec/cred" 리소스는 장치 소유자 (일명 DOTS)가 재설정하고 다시

프로비저닝 할 수 있다.

(마) 보안 자격 증명 관리

■ **자격 증명 수명주기 일반**

OCF 자격 증명 수명주기에는 (1) 생성, (2) 삭제, (3) 새로 고침, (4) 발급 및 (5) 해지 단계가 있다.

CMS는 장치에 자격 증명 리소스를 제공해야한다. 장치는 "/oic/sec/cred"리소스의 rowneruuid 속성을 CMS가 보안 연결을 설정하는 데 사용하는 자격 증명의 DeviceID와 일치시켜 CMS가 인증되었는지 확인해야한다. CMS를 사용하여 만든 자격 증명 리소스에는 특수한 자격 증명 발급 프로토콜 및 메시지가 포함될 수 있다. 여기에는 인증 기관 (CA)과 같은 공개 키 인프라 (PKI), 키 배포 센터 (KDC)와 같은 대칭 키 관리 또는 DOTS, CMS 또는 AMS에 의한 프로비저닝 작업의 일부가 포함될 수 있다.

CMS는 알려진 손상된 자격 증명 리소스를 삭제해야한다. 기기 (예 : 자격 증명 리소스가 호스팅 되는 기기)는 만료된 자격 증명 리소스를 삭제해야한다. 메모리 및 저장 공간을 관리하기 위해 만료된 자격 증명 리소스를 삭제할 수 있다. OCF 키 관리에서 삭제는 자격 증명 일시 중단과 같다.

자격 증명 새로 고침은 만료되기 전에 수행 될 수 있다. CMS는 자격 증명 새로 고침을 수행해야한다. "/oic/sec/cred"리소스는 Period 속성을 사용한 만료를 지원한다. 자격 증명 새로 고침은 자격 증명 만료되거나 암호화 된 바이트의 최대 임계 값을 초과 할 때 적용될 수 있다. 자격 증명 새로 고침 방법은 키 새로 고침을 수행 할 때 사용 가능한 옵션을 지정한다. 기간 속성은 자격 증명 만료되는시기를 알려줍니다. 장치는 기존의 자격 증명을 새로 고치기 위해 현재 만료되지 않은 자격 증명을 사용하는 자격 증명 새로 고침 방법을 사용하여 새로운 자격 증명을 미리 얻을 수 있다. 장치에 내부 시간 원본이 없으면 CMS에서 정기적으로 현재 시간을 가져와야한다. CMS 자격 증명 만료되도록 허용 된 경우 DOTS 서비스를 사용하여 CMS 자격 증명을 장치에 다시 제공 할 수 있다. Onboarding 설정된 자격 증명의 만료가 허용되는 경우 DOTS는 장치를 다시 Onboard하여 장치 소유자 전송 단계를 다시 적용해야한다. 모든 장치는 하나 이상의 자격 증명 새로 고침 방법을 지원해야 한다.

CMS에서 발급 한 자격 증명에는 해지 기능이있을 수 있다. 해지 방법에 정상적인 만료 기간 전에 해지 할 자격 증명을 식별하는 해지 개체를 제공해야하는 경우 원래 발급 된 자격 증명을 대체하는 해지 정보가 포함 된 자격 증명 리소스가 생성된다. 해지 개체 만료는 해지 된 자격 증명의 해지와 일치해야 만료시 해지 개체가 정리된다.

자격 증명이나 장치에 해지를 적용하는 것이 개념적으로 합리적이다. 장치 해지에서는 해지 된 장치와 관련된 모든 자격 증명에 해지 대상으로 간주되어야한다. 장치를 분실, 도난 당하거나 손상된 경우 장치 해지가 필요하다. 취소 된 장치에서 자격 증명을 삭제하지 못할 수도 있다.

"/oic/sec/cred"리소스는 인증 및 데이터 보호에 사용되는 여러 암호화 키 및 기타 정보를 지원하는 자격 증명 유형 속성을 유지 관리한다. 지원되는 자격 증명 유형에는 쌍별 대칭 키, 그룹 대칭 키, 비대칭 인증 키, 인증서 (예 : 서명 된 비대칭 키) 및 공유 비밀 (예 : PIN/암호)이 포함된다.

CMS는 정확히 하나의 다른 페어 단위 대칭 자격 증명을 피어 장치에 제공해야한다. CMS는 관리되는 장치에 제공하는 쌍별 대칭 키를 저장해서는 안된다. 쌍 키는 임시 키 계약 프로토콜을 통해 설정할 수 있다.

"/oic/sec/cred"리소스의 PrivateData 속성에는 대칭 키가 포함되어 있다. PublicData 속성은 pairwise 키를 포함하는 피어 장치로 암호화 된 토큰을 포함 할 수 있다. OptionalData 속성에는 해지 상태가 포함될 수 있다. 장치 구현자는 PrivateData를 비공개로 유지하기 위해 강화 된 키 저장 기술을 적용해야한다.

장치 구현자는 "/oic/sec/cred", "/oic/sec/crl", "/oic/sec/roles", "/oic/sec/csr"리소스의 적절한 무결성, 기밀성 및 액세스 보호를 적용해야한다.

그룹 키는 장치 그룹 (3 이상)간에 공유되는 대칭 키이다. 그룹 키는 그룹 참가자간에 데이터를 효율적으로 공유하는 데 사용된다. 그룹 키는 장치의 인증을 제공하지 않지만 그룹의 멤버 자격 만 설정한다. CMS는 그룹 대칭 키 자격 증명을 그룹 구성원에게 제공해야한다. CMS는 그룹 구성원을 유지 관리한다.

"/oic/sec/cred"리소스의 PrivateData 속성에는 대칭 키가 포함되어 있다. PublicData 속성에는 그룹 이름이 포함될 수 있다. OptionalData 속성에는 해지 상태가 포함될 수 있다. 장치 구현자는 PrivateData를 비공개로 유지하기 위해 강화 된 키 저장 기술을 적용해야한다.

장치 구현자는 "/oic/sec/cred", "/oic/sec/crl", "/oic/sec/roles", "/oic/sec/csr"리소스의 적절한 무결성, 기밀성 및 액세스 보호를 적용해야한다.

비대칭 인증 키 자격 증명에는 공개 키와 개인 키 쌍이 포함되거나 공개 키만 있다. 개인 키는 장치 인증 문제에 서명하는 데 사용된다. 공개 키는 장치 인증 시도 응답을 확인하는 데 사용된다. "/oic/sec/cred"리소스의 PrivateData 속성에는 개인 키가 포함되어 있다. PublicData 속성에는 공개 키

가 포함되어 있다. OptionalData 속성에는 해지 상태가 포함될 수 있다.

장치 구현자는 PrivateData를 비공개로 유지하기 위해 강화 된 키 저장 기술을 적용해야한다. 장치는 개인 키만 장치에 의해 알 수 있도록 내부에서 비대칭 인증 키 쌍을 생성해야한다. 장치간에 개인 키 자료를 전송해야하는 경우는 9.3.4.2를 참조하십시오. 장치 구현자는 "/oic/sec/cred", "/oic/sec/crl", "/oic/sec/roles", "/oic/sec/csr"리소스의 적절한 무결성, 기밀성 및 액세스 보호를 적용해야한다.

장치는 오프 장치 키 페어 생성 및 프로비저닝을 허용 할 때 업계 표준의 높은 보증 기술을 사용해야한다. 이러한 키 페어의 사용은 특히 키 페어가 변경 불가능하고 프로비저닝 후 변경 또는 교체 할 수없는 경우 최소화해야한다. Onboarding의 일부로 사용될 때,이 키 페어는 디바이스가 DOTS 또는 사용자가 디바이스 Onboarding을 수락하도록 설득하기 위해 인증서에 제조업체가 지정한 속성을 소유하고 있음을 증명하는 데 사용될 수 있다.

KEK (비대칭 키 암호화 키) 자격 증명은 키를 배포하거나 저장할 때 대칭 키를 래핑하는 데 사용된다. "/oic/sec/cred"리소스의 PrivateData 속성에는 개인 키가 포함되어 있다. PublicData 속성에는 공개 키가 포함되어 있다. OptionalData 속성에는 해지 상태가 포함될 수 있다. 장치 구현자는 PrivateData를 비공개로 유지하기 위해 강화 된 키 저장 기술을 적용해야한다. 장치 구현자는 "/oic/sec/cred", "/oic/sec/crl", "/oic/sec/roles", "/oic/sec/csr"리소스의 적절한 무결성, 기밀성 및 액세스 보호를 적용해야한다.

인증서 자격 증명은 CMS 또는 외부 인증 기관 (CA)에서 발급 한 인증서와 함께 제공되는 비대칭 키이다. 인증서 등록 프로토콜은 인증서를 획득하고 소유 증명을 설정하는 데 사용된다. 발급 된 인증서는 비대칭 키 자격 증명 리소스와 함께 저장된다. 인증서 폐기 상태와 같은 인증서 수명주기 관리에 유용한 다른 개체는 자격 증명 리소스와 연결된다. 비대칭 키 자격 증명 리소스 또는 자체 서명 인증서 자격 증명을 사용하여 경로 유효성 검사를 종료한다. "/oic/sec/cred"리소스의 PrivateData 속성에는 개인 키가 포함되어 있다. PublicData 속성에는 발급 된 인증서가 포함된다. OptionalData 속성에는 해지 상태가 포함될 수 있다. 장치 구현자는 PrivateData를 비공개로 유지하기 위해 강화 된 키 저장 기술을 적용해야한다. 장치 구현자는 "/oic/sec/cred", "/oic/sec/crl", "/oic/sec/roles", "/oic/sec/csr"리소스의 적절한 무결성, 기밀성 및 액세스 보호를 적용해야한다.

공유 비밀 자격 증명은 다른 OCF 자격 증명 유형을 지원하지 않는 외국 시스템이나 장치에 대한 장치 액세스 권한을 부여하는 PIN 또는 암호를 유지하는 데 사용된다. "/oic/sec/cred"리소스의 PrivateData 속성에는 비밀번호 변경 및 확인에 유용한 PIN, 비밀번호 및 기타 값이 포함되어 있다. 가능한 경우 PublicData 속성에 사용자 또는 계정 이름이 포함될 수 있다. OptionalData 속성에는 해지 상태가 포함될 수 있다. 장치 구현자는 PrivateData를 비공개로 유지하기 위해 강화 된 키 저장 기술을 적용해야한다. 장치 구현자는 "/oic/sec/cred", "/oic/sec/crl", "/oic/sec/roles", "/oic/sec/csr"리소스의 적절한 무결성, 기밀성 및 액세스 보호를 적용해야한다.

OCF 보안 도메인에서 통신하는 동안 인증 및 전송 보안을 달성하기 위해 통신 당사자의 공개 키와 개인 키를 포함하는 인증서를 사용할 수 있다. 인증서 및 개인 키는 로컬 또는 원격 인증 기관 (CA)에서 발급 할 수 있다. 로컬 CA의 경우 X.509 기반의 CRL (인증서 해지 목록)을 사용하여 ID 증명을 확인한다. 원격 CA의 경우 OCSP (Online Certificate Status Protocol)를 사용하여 ID 및 유효성 증명을 확인할 수 있다.

OCF 인증서 및 OCF CRL (인증서 해지 목록) 형식은 X.509 형식의 하위 집합이며 타원 곡선 알고리즘과 DER 인코딩 형식 만 허용되며 X.509의 대부분의 선택적 필드는 지원되지 않으므로 형식이 충족된다. 제한된 장치의 요구 사항 서버의 인증서 및 CRL 관리에 대해서는 인증서 및 CRL의 리소스를 저장, 검색 및 파싱하는 프로세스가 보안 리소스 관리자 계층에서 수행된다. 관련 인터페이스는 상위 계층에 노출 될 수 있다. SRM은 서버의 보안 시행 지점이므로 인증서 및 CRL의 데이터는 SVR 데이터베이스에 저장 및 관리된다. CMS는 발급 된 인증서의 인증서 수명주기를 관리한다. DOTS는 새로 Onboard 될 때 장치에 CMS를 할당해야한다. 발급 CMS는 발급 한 인증서에 대한 인증서 해지를 처리해야한다. 인증서 개인 키가 손상된 경우 CMS는 인증서를 해지해야한다. 장치에서 CRL을 사용하는 경우 CMS는 정기적으로 (예 : 3 개월마다) 관리하는 장치의 "/oic/sec/crl"리소스를 업데이트해야한다.

OCF 인증서 형식은 IETF RFC 5280에 정의 된 X.509 형식 (버전 3 이상)의 하위 집합이다.이 절은 OCF 응용 프로그램 내에서 X.509 인증서를 사용하려는 커뮤니티를 위해 쉽게 사용할 수 있도록 프로파일을 개발한다. X.509 기술. X.509 v3 인증서 형식은 OCF 특정 확장의 형식 및 의미에 대한 추가 정보와 함께 자세히 설명된다. 지원되는 표준 인증서 확장도 나열된다. 인증서 형식 : OCF 인증서 프로파일은 IETF RFC 5280에서 파생되었다. 그러나 이 문서는 더 이상 사용되지 않으며 OCF 컨텍스트에서 사용되지 않는 "issuerUniqueID"및 "subjectUniqueID"필드를 지원하지 않는다. 이러한 필드가 인증서에있는 경우 준수 엔티티는 해당 내용을 무시해야한다. 인증서 인코딩 : 준수하는 엔티티는 ISO/IEC 8825-1에 정의 된 DER (Distinguished Encoding Rules)을 사용하여 인증서를 인코딩해야한다. 인증서 계층 및 암호화 매개 변수. OCF는 공개 키 인프라 (즉, 루트 CA, 중간 CA 및 EE 인증서)를위한 3 계층 구조를 지원한다. OCF

공인 CA는 ECC (Elliptic Curve Cryptography) 키 (secp256r1 - OID : 1.2.840.10045.3.1.7)를 사용하고 인증서 서명에 ecdsaWithSHA256 (OID : 1.2.840.10045.4.3.2) 알고리즘을 사용해야한다. 다음 절은 OCF 인증서 프로파일에 지원되는 표준 및 사용자 정의 확장을 지정한다.

다음 표는 루트 CA 인증서에 필요한 X.509 v1 필드를 설명한다.

표 16 - X.509 v1 fields for Root CA Certificates

V1 Field	Value / Remarks
signatureAlgorithm	ecdsa-with-SHA256 (OID: 1.2.840.10045.4.3.2)
Version	v3 (value is 2)
SerialNumber	SHALL be a positive integer, unique among all certificates issued by a given CA
Issuer	SHALL match the Subject field
Subject	SHALL match the Issuer field
notBefore	The time at which the Root CA Certificate was generated. See 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.
notAfter	No stipulation for expiry date. See 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.
Subject Public Key Info	id-ecPublicKey (OID: 1.2.840.10045.2.1) secp256r1 (OID:1.2.840.10045.3.1.7)

다음 표에서는 루트 CA 인증서에 필요한 X.509 v3 확장을 설명한다.

표 17 - X.509 v3 extensions for Root CA Certificates

Extension	Required/Optional	Criticality	Value / Remarks
authorityKeyIdentifier	OPTIONAL	Non-critical	N/A
subjectKeyIdentifier	OPTIONAL	Non-critical	N/A
keyUsage	REQUIRED	Critical	keyCertSign (5) & cRLSign (6) bits shall be enabled. digitalSignature(0) bit may be enabled. All other bits shall not be enabled.
basicConstraints	REQUIRED	Critical	cA = TRUE pathLenConstraint = not present (unlimited)

프로필 다음 표는 중간 CA 인증서에 필요한 X.509 v1 필드를 설명한다.

표 18 - X.509 v1 fields for Intermediate CA Certificates

V1 Field	Value / Remarks
signatureAlgorithm	ecdsa-with-SHA256 (OID: 1.2.840.10045.4.3.2)
Version	v3 (value is 2)
SerialNumber	SHALL be a positive integer, unique among all certificates issued by Root CA
Issuer	SHALL match the Subject field of the issuing Root CA
Subject	(no stipulation)
notBefore	The time at which the Intermediate CA Certificate was generated. See clause 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.
notAfter	No stipulation for expiry date. See clause 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.
Subject Public Key Info	id-ecPublicKey (OID: 1.2.840.10045.2.1) secp256r1 (OID: 1.2.840.10045.3.1.7)

다음 표에서는 중간 CA 인증서에 필요한 X.509 v3 확장을 설명한다.

표 19 - X.509 v3 extensions for Intermediate CA Certificates

Extension	Required/Optional	Criticality	Value / Remarks
authorityKeyIdentifier	OPTIONAL	Non-critical	N/A
subjectKeyIdentifier	OPTIONAL	Non-critical	N/A
keyUsage	REQUIRED	Critical	keyCertSign (5) & cRLSign (6) bits shall be enabled. digitalSignature (0) bit may be enabled All other bits shall not be enabled.
basicConstraints	REQUIRED	Critical	cA = TRUE pathLenConstraint = 0 (can only sign End-Entity certs)
certificatePolicies	OPTIONAL	Non-critical	(no stipulation)
cRLDistributionPoints	OPTIONAL	Non-critical	1 or more URIs where the Certificate Revocation List (CRL) from the Root can be obtained.
authorityInformationAccess	OPTIONAL	Non-critical	OCSP URI – the URI of the Root CA's OCSP Responder

다음 표는 블랙 보안 프로파일에 사용되는 엔드 엔티티 인증서에 필요한 X.509 v1 필드를 설명한다.

표 19 - X.509 v1 fields for End-Entity Certificates

V1 Field	Value / Remarks
signatureAlgorithm	ecdsa-with-SHA256 (OID: 1.2.840.10045.4.3.2)
Version	v3 (value is 2)
SerialNumber	SHALL be a positive integer, unique among all certificates issued by the Intermediate CA
Issuer	SHALL match the Subject field of the issuing Intermediate CA
Subject	Subject DN shall include: o=OCF-verified device manufacturer organization name.  The Subject DN may include other attributes (e.g. cn, c, ou, etc.) with no stipulation by OCF.
notBefore	The time at which the End-Entity Certificate was generated. See clause 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.
notAfter	No stipulation. See clause 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.
Subject Public Key Info	id-ecPublicKey (OID: 1.2.840.10045.2.1) secp256r1 (OID:1.2.840.10045.3.1.7)

다음 표에서는 최종 엔터티 인증서에 필요한 X.509 v3 확장을 설명한다.

표 20 - X.509 v3 extensions for End-Entity Certificates

Extension	Required/ Optional	Criticality	Value / Remarks
authorityKeyIdentifier	OPTIONAL	Non-critical	N/A
subjectKeyIdentifier	OPTIONAL	Non-critical	N/A
keyUsage	REQUIRED	Critical	digitalSignature (0) and keyAgreement(4) bits SHALL be the only bits enabled
basicConstraints	OPTIONAL	Non-Critical	cA = FALSE pathLenConstraint = not present
certificatePolicies	OPTIONAL	Non-critical	End-Entity certificates chaining to an OCF Root CA SHOULD contain at least one PolicyIdentifierId set to the OCF Certificate Policy OID – (1.3.6.1.4.1.51414.0.1.2) corresponding to the version of the OCF Certificate Policy under which it was issued. Additional manufacturer-specific CP OIDs may also be populated.
extendedKeyUsage	REQUIRED	Non-critical	The following extendedKeyUsage (EKU) OIDs SHALL both be present: <ul style="list-style-type: none"> <li>• serverAuthentication - 1.3.6.1.5.5.7.3.1</li> <li>• clientAuthentication - 1.3.6.1.5.5.7.3.2</li> </ul> Exactly ONE of the following OIDs SHALL be present: <ul style="list-style-type: none"> <li>• Identity certificate - 1.3.6.1.4.1.44924.1.6</li> <li>• Role certificate - 1.3.6.1.4.1.44924.1.7</li> </ul> End-Entity certificates SHALL NOT contain the anyExtendedKeyUsage OID (2.5.29.37.0)
subjectAlternativeName	REQUIRED UNDER CERTAIN CONDITIONS	Non-critical	The subjectAltName extension is used to encode one or more Role ID values in role certificates, binding the roles to the subject public key. When the extendedKeyUsage (EKU) extension contains the Identity Certificate OID (1.3.6.1.4.1.44924.1.6), the subjectAltName extension SHOULD NOT be present. If the EKU extension contains the Role Certificate

			OID (1.3.6.1.4.1.44924.1.7), the subjectAltName extension SHALL be present and populated as follows: Each GeneralName in the GeneralNames SEQUENCE which encodes a role shall be a directoryName, which is of type Name. Name is an X.501 Distinguished Name. Each Name shall contain exactly one CN (Common Name) component, and zero or one OU (Organizational Unit) components. The OU component, if present, shall specify the authority that defined the semantics of the role. If the OU component is absent, the certificate issuer has defined the role. The CN component shall encode the role ID. Other GeneralName types in the SEQUENCE may be present, but shall not be interpreted as roles. The role, and authority shall be encoded as ASN.1 PrintableString type, the restricted character set [0-9a-z-A-z '()+,./:=?].
cRLDistributionPoints	OPTIONAL	Non-critical	1 or more URIs where the Certificate Revocation List (CRL) from the Intermediate CA can be obtained.
authorityInformationAccess	OPTIONAL	Non-critical	OCSP URI – the URI of the Intermediate CA's OCSP Responder
OCF Compliance	OPTIONAL	Non-critical	See 9.4.2.2.4
Manufacturer Usage Description (MUD)	OPTIONAL	Non-critical	Contains a single Uniform Resource Locator (URL) that points to an on-line Manufacturer Usage Description concerning the certificate subject. See 9.4.2.2.5
OCF Security Claims	OPTIONAL	Non-critical	Contains a list of security claims above those required by this OCF Compliance version or Security Profile. See 9.4.2.2.6
OCF CPL Attributes	OPTIONAL	Non-critical	Contains the list of OCF Attributes used to perform OCF Certified Product List lookups

OCF Compliance Extension은 장치 유형, 제조업체, OCF 버전 및 장치의 보안 프로파일 준수를 올바르게 식별하기 위해 필요한 매개 변수를 정의한다. 확장은 장치가 구현하는 OCF 문서의 특정 기본 버전을 제공하는 "ocfVersion" 필드를 전달한다. "ocfVersion" 필드는 세 개의 정수 시퀀스 ("major", "minor" 및 "build")를 포함해야 한다. 예를 들어 엔터티가 OCF 사양 1.3.2를 준수하도록 인증된 경우 "ocfVersion"의 "major", "minor" 및 "build" 필드는 "1", "3"으로 설정된다. 및 "2". 보안 프로파일은 "ocfVersion"을 사용하여 지정된 기본 버전의 OCF 문서를 준수 함을 나타낸다. "securityProfile" 필드는 인증서와 관련된 하나 이상의 지원되는 보안 프로파일의 ocfSecurityProfile OID (14.8.3 절)를 문자열 형식 (UTF-8)으로 전달해야 한다. 인증서와 관련된 모든 보안 프로파일은 이 필드로 식별해야 한다. 확장은 또한 "DeviceName"과 "deviceManufacturer"라는 두 개의 문자열 필드 (UTF-8)를 포함해야 한다. 이 필드에는 장치 이름과 제조업체에 대한 사람이 읽을 수 있는 설명이 각각 있다. OCFCompliance 확장 (OID - 1.3.6.1.4.1.51414.1.0)의 ASN.1 정의는 다음과 같이 정의된다.

```

id-OCF OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1)
                                private(4) enterprise(1) OCF(51414) }

id-ocfX509Extensions OBJECT IDENTIFIER ::= { id-OCF 1 }

id-ocfCompliance OBJECT IDENTIFIER ::= { id-ocfX509Extensions 0 }

ocfVersion ::= SEQUENCE {
    major    INTEGER,
             --Major version number
    minor    INTEGER,
             --Minor version number
    build    INTEGER,
             --Build/Micro version number
}

ocfCompliance ::= SEQUENCE {
    version          ocfVersion,
                    --Device/OCF version
    securityProfile  SEQUENCE SIZE (1..MAX) OF ocfSecurityProfileOID,
                    --Sequence of OCF Security Profile OID strings
                    --Clause 14.8.2 defines valid ocfSecurityProfileOIDs
    deviceName       UTF8String,
                    --Name of the device
    deviceManufacturer UTF8String,
                    --Human-Readable Manufacturer
                    --of the device
}

```

제조업체 사용 설명 (MUD) 확장의 목표는 장치가 올바르게 작동하는 데 필요한 액세스 및 네트워크 기능을 네트워크에 신호로 보내는 수단을 제공하는 것이다. MUD 확장을 사용하면 액세스 제어를보다 쉽게 달성하고 대규모로 배포 할 수 있다. 이 글을 쓰는 현재의 MUD v3 확장 초안은 다음과 같다.

<https://tools.ietf.org/html/rfc8520#section-11>

MUD v3 확장의 ASN.1 정의는 다음과 같이 정의된다.

```

MUDURLExtnModule-2016 {      iso(1) identified-organization(3) dod(6)
                              internet(1) security(5) mechanisms(5) pkix(7)
                              id-mod(0) id-mod-mudURLExtn2016(88) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN
-- EXPORTS ALL --
IMPORTS

EXTENSION
FROM PKIX-CommonTypes-2009
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-pkixCommon-02(57) }
id-pe
FROM PKIX1Explicit-2009
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-pkix1-explicit-02(51) } ;
MUDCertExtensions EXTENSION ::= { ext-MUDURL, ... }
ext-MUDURL EXTENSION ::= { SYNTAX MUDURLSyntax
                            IDENTIFIED BY id-pe-mud-url }

id-pe-mud-url OBJECT IDENTIFIER ::= { id-pe 25 }

MUDURLSyntax ::= IA5String

```

9.4.2.2

END

OCF 보안 클레임 확장은 제조업체/통합 업체가 장치의 보안 상태에 대해 OCF 준수 버전 또는 장치가 나타내는 OCF 보안 프로파일의 요구보다 높은 보안 클레임을 나타내는 OID 목록을 정의한다. 이 확장의 목

적은 보안에 대한 주장을 프로그래밍 방식으로 평가하여 일부 플랫폼/정책/관리자가 Onboard 또는 도전 중인 사항을 더 잘 이해할 수 있도록 하는 것이다. OCF 보안 클레임 확장 (OID - 1.3.6.1.4.1.51414.1.1)의 ASN.1 정의는 다음과 같이 정의된다.

```
id-OCF OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1)
                                private(4) enterprise(1) OCF(51414) }

id-ocfX509Extensions OBJECT IDENTIFIER ::= { id-OCF 1 }

id-ocfSecurityClaims OBJECT IDENTIFIER ::= { id-ocfX509Extensions 1 }

claim-secure-boot                ::= ocfSecurityClaimsOID { id-ocfSecurityClaims 0 }
--Device claims that the boot process follows a procedure trusted
--by the firmware and the BIOS

claim-hw-backed-cred-storage ::= ocfSecurityClaimsOID { id-ocfSecurityClaims 1 }
--Device claims that credentials are stored in a specialized hardware
--protection environment such as a Trusted Platform Module (TPM) or
--similar mechanism.

ocfSecurityClaimsOID ::= OBJECT IDENTIFIER

ocfSecurityClaims ::= SEQUENCE SIZE (1..MAX) of ocfSecurityClaimsOID
```

OCF 인증 제품 목록 확장은 OCF 준수 관리 시스템 인증 제품 목록 (OCMS-CPL)을 활용하는 데 필요한 매개 변수를 정의한다. 이 조항은 OCMS-CPL을 활용하려는 경우에만 적용된다. OBT는 이러한 속성을 사용하여 장치의 준수 수준을 확인할 수 있다. 이 확장은 OCF CPL 속성 (PANA), 모델 및 버전을 제공한다.

'cpl-at-IANAPen'IANA 개인 기업 번호 (PEN)는 IANA PEN 목록에 설정된 제조업체의 고유 한 PEN을 제공한다. 최종 제품에있는 'cpl-at-IANAPen'필드는 OCF 인증 동안보고 된 정보와 동일해야한다.

(<https://www.iana.org/assignments/enterprisenumbers> 참조)

'cpl-at-model'은 OCF 인증 제품의 모델 이름을 나타낸다. 최종 제품에서 발견되는 '모델 별 cpl'필드는 OCF 인증시보고 된 정보와 동일해야한다. 'cpl-at-version'은 OCF 인증 제품 버전을 나타낸다. 최종 제품에서 발견되는 '버전 별 cpl'필드는 OCF 인증 중에보고 된 정보와 동일해야한다. OCF CPL 속성 확장 (OID - 1.3.6.1.4.1.51414.1.2)의 ASN.1 정의는 다음과 같이 정의된다.

```
id-OCF OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1)
                                private(4) enterprise(1) OCF(51414) }

id-ocfX509Extensions OBJECT IDENTIFIER ::= { id-OCF 1 }

id-ocfCPLAttributes OBJECT IDENTIFIER ::= { id-ocfX509Extensions 2 }

cpl-at-IANAPen ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 0 }
cpl-at-model   ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 1 }
cpl-at-version ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 2 }

ocfCPLAttributes ::= SEQUENCE {
    cpl-at-IANAPen    UTF8String,
                    --Manufacturer's registered IANA Private Enterprise Number
    cpl-at-model     UTF8String,
                    --Device OCF Security Profile
    cpl-at-version   UTF8String
                    --Name of the device
}
```

이러한 인증서 확장은 IETF RFC 5280의 표준 부분이므로 이 문서에는 RFC의 조항 번호가 포함되어 있으며 참조로 포함시킨다. 각 확장이 여기에 요약되어 있으며 RFC 정의에 대한 모든 수정 사항이 나열된다. 장치는 여기에 나열된 확장을 구현하고 이해해야한다. RFC의 다른 확장은 이 문서에 포함되어 있지 않으므로 필요하지 않다. 5.4.6.4는 확장 처리 및 특정 확장이 없을 때 취해야 할 조치를 포함하여 인증서 체인을 검증 할 때 장치가 구현해야 하는 사항을 설명한다.

AKI(Authority Key Identifier) 확장은 인증서에 서명하는 데 사용되는 개인 키에 해당하는 공용 키를 식별할 수 있는 수단을 제공한다. 이 문서는 이 확장의 참조된 정의를 다음과 같이 수정한다.

AuthorityKeyIdentifier 시퀀스의 권한CertIssuer 또는 권한CertSerialNumber 필드는 허용되지 않으며

keyIdentifier만 허용된다. 이렇게 되면 다음과 같은 문법적 정의가 나온다.

```
id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { id-ce 35 }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier          [0] KeyIdentifier
}

KeyIdentifier ::= OCTET STRING
```

SKI (Subject Key Identifier) 확장은 특정 공개 키가 포함 된 인증서를 식별하는 수단을 제공한다. 이 문서는 이 확장의 참조 정의를 다음과 같이 수정한다.

주체 키 식별자는 인증서의 SubjectPublicKeyInfo 필드에 포함 된 공개 키 또는 고유 한 값을 생성하는 방법에서 파생되어야 한다. 이 문서는 BIT STRING subjectPublicKey 값의 256 비트 SHA-2 해시를 권장한다 (태그, 길이 및 사용되지 않는 비트 수 제외). 인증서 체인을 확인하는 장치는 키 식별자를 계산하는 특정 방법을 가정해서는 안되며, 인증서에 표시된 키 식별자를 기준으로 인증 경로 구성에서 일치하는 AKI 및 SKI 만 기반으로 해야 한다.

EKU 확장이 있고 이 값이 역할 인증서임을 나타내는 XXXXXX 값을 갖는 경우 주체 대체 이름 (subjectAltName) 확장이 존재하고 아래 설명 된 대로 해석된다. EKU가 없거나 다른 값이 있는 경우 subjectAltName 확장자가 없어야 한다 (SHOULD). subjectAltName 확장은 역할 인증서에서 하나 이상의 역할 ID 값을 인코딩하여 역할을 주체 공개 키에 바인딩하는 데 사용된다. subjectAltName 확장은 IETF RFC 5280에 정의되어 있다.

```
id-ce-subjectAltName OBJECT IDENTIFIER ::= { id-ce 17 }

SubjectAltName ::= GeneralNames

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName          [0] OtherName,
    rfc5322Name        [1] IA5String,
    dNSName            [2] IA5String,
    x400Address        [3] ORAddress,
    directoryName      [4] Name,
    ediPartyName       [5] EDIPartyName,
    uniformResourceIdentifier [6] IA5String,
    iPAddress          [7] OCTET STRING,
    registeredID       [8] OBJECT IDENTIFIER }

EDIPartyName ::= SEQUENCE {
    nameAssigner      [0] DirectoryString OPTIONAL,
    partyName         [1] DirectoryString }
```

역할을 인코딩하는 GeneralNames SEQUENCE의 각 GeneralName은 Name 유형의 directoryName 이어야 한다. 이름은 X.501 식별 이름이다. 각 이름은 정확히 하나의 CN (Common Name) 구성 요소와 0 개 또는 하나의 OU (Organizational Unit) 구성 요소를 포함해야 한다. OU 컴포넌트는 존재하는 경우 역할의 의미를 정의한 권한을 명시해야 한다. OU 구성 요소가 없으면 인증서 발급자가 역할을 정의한 것이다. CN 컴포넌트는 역할 ID를 인코딩해야 한다. SEQUENCE에 다른 GeneralName 유형이 존재할 수 있지만 역할로 해석되지 않아야 한다. 따라서 인증서 발급자가 subjectAltName 확장명에 비 역할 이름을 포함하는 경우 확장명을 중요로 표시해서는 안 된다. 역할 및 권한은 ASN.1 PrintableString 유형, 제한된 문자 세트 [0-9a-z-A-Z '() +,-./: =?]로 인코딩해야 한다.

키 사용 확장은 인증서에 포함 된 키의 목적 (예 : 암호화, 서명, 인증서 서명)을 정의한다. 둘 이상의 작업에 사용할 수 있는 키를 제한해야 할 경우 사용 제한이 적용될 수 있다. 이 문서는 이 확장의 참조 정의를 수정하지 않는다.

기본 제약 조건 확장은 인증서의 주체가 CA인지 여부와 이 인증서를 포함하는 유효한 인증 경로의 최대 깊이를 식별한다. 이 확장이 없으면 인증서는 다른 인증서의 발급자가 될 수 없다. 이 문서는 이 확장의 참조 정의를 수정하지 않는다.

확장 키 사용은 인증 된 공개 키를 사용할 수 있는 허용 된 목적을 설명한다. 장치는 인증서를 받으면 인증서가 제공되는 상호 작용의 컨텍스트를 기반으로 목적을 결정하고 해당 용도로 인증서를 사용할 수 있는지 확인한다. 이 문서는 이 확장의 참조 정의를 다음과 같이 수정한다.

CA는 이 확장을 중요로 표시해야 한다. CA는 anyExtendedKeyUsage OID (2.5.29.37.0)와 함께 인증서를 발급해서는 안된다.

OCF 특정 목적 맞이를 나타내는 지정된 OID 목록은 다음과 같다.

- 신원 인증서 1.3.6.1.4.1.44924.1.6
- 역할 인증서 1.3.6.1.4.1.44924.1.7

[ISO/IEC 8825-1]에 정의된 ASN.1 고유 인코딩 규칙 (DER 인코딩 방법)을 사용하여 CRL을 인코딩해야 한다.

X.509 v2 CRL [X.509] [X9.55] 용 ANSI X9, ISO/IEC 및 ITU-T에 의해 정의된 확장은 추가 속성을 CRL과 연관시키는 방법을 제공한다. 이 인증서 프로파일에서 다음 X.509 확장 목록이 지원되어야 한다.

- 권한 키 식별자 (선택 사항, 중요하지 않음)-권한 키 식별자 확장은 CRL에 서명하는 데 사용된 개인 키에 해당하는 공개 키를 식별하는 수단을 제공한다. 준수하는 CRL 발급자는 키 식별자 방법을 사용해야 하며 발급된 모든 CRL에 이 확장을 포함해야 한다.
- CRL 번호 (선택 사항, 중요하지 않음)-CRL 번호는 지정된 CRL 범위 및 CRL 발급자에 대해 단조 증가하는 시퀀스 번호를 전달하는 중요하지 않은 CRL 확장이다.

CRL 항목 확장 : X.509 v2 CRL 용 ISO/IEC, ITU-T 및 ANSI X9로 정의된 CRL 항목 확장은 추가 속성을 CRL 항목 [X.509] [X9.55]와 연결하는 방법을 제공한다. 이 문서는 CRL 항목에 대한 특정 확장의 사용에 대한 권장 사항을 제공하지 않지만, 준수하는 CA는 중요하지 않은 한 CRL에서 해당 확장을 사용할 수 있다.

OCF 준수 엔티티는 TLS 버전 1.2를 지원해야 한다. 호환 엔티티는 IETF RFC 7251에 정의된 대로 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8 암호 스위트를 지원해야 하며 TLS v1.2 사양에 정의된 추가 암호를 지원할 수 있다.

장치 인증서 및 개인 키는 cred Resource에 보관된다. CRL은 해지 목록을 유지 관리하기 위해 정의된 별도의 cri 리소스로 유지 관리되고 업데이트된다. cred 리소스에는 장치와 관련된 인증서 정보가 포함되어 있다. PublicData 속성에는 장치 인증서와 CA 인증서 체인이 있다. PrivateData 속성은 인증서와 페어링된 장치 개인 키를 보유한다. 인증서 해지 목록 리소스는 CMS를 통해 얻은 해지된 인증서 목록을 유지 관리하는 데 사용된다. 장치는 인증서 경로 확인의 일부로 해지된 인증서를 고려해야 한다. CRL 리소스가 오래되었거나 전체 목록을 유지 관리하기에 플랫폼 리소스가 부족한 경우 장치는 CMS에 현재 해지 상태를 쿼리해야 한다.

CMS (예 : 허브 또는 스마트 폰)는 새 장치에 대한 인증서를 발급한다. CMS는 자체 인증서와 키 쌍을 가져야 한다. 인증서는 a) 루트 CA 역할을 하는 경우 자체 서명되거나 b) 하위 CA 역할을 하는 경우 신뢰 계층 구조에서 상위 CA가 서명 한 것이다. 두 경우 모두 인증서는 9.4.2에 설명된 형식을 가져야 한다. CMS의 CA는 장치의 공개 키와 개인 키의 소유 증명을 검색하고 이 CA 인증서로 서명된 장치의 인증서를 생성한 다음 CMS는 이를 CA 인증서 체인을 포함하여 장치로 전송해야 한다. 선택적으로 CMS는 하나 이상의 역할 인증서를 전송할 수도 있으며, 이는 9.4.2 절에 설명된 형식을 갖는다. 각 역할 인증서의 subjectPublicKey는 장치 인증서의 subjectPublicKey와 일치해야 한다. 그림의 순서에서 CSR (Certificate Signing Request)은 IETF RFC 2986에서 PKCS # 10으로 정의되며 여기에 참조로 포함된다. 클라이언트 지향 모델에 대한 인증서 전송의 순서 흐름은 그림에 설명되어 있다.

- 1) CMS는 장치에서 인증서를 요청하는 CSR을 검색한다. 이 CSR에서 장치는 요청한 UUID를 Subject에 넣고 공개 키를 SubjectPublicKeyInfo에 배치해야 한다. 장치는 제시할 공개 키를 결정한다. 인증에 사용하기 위해 이미 제공한 키일 수도 있고, 존재하지 않는 경우 내부적으로 새로운 키 페어를 생성하여 공개 부분을 제공할 수도 있다. 키 쌍은 인증서가 OCF 인증에 사용하도록 제한되므로 허용된 암호 스위트와 호환 가능해야 한다.
- 2) 장치에 사전 프로비저닝된 키 페어가 없고 자체적으로 키 페어를 생성할 수 없는 경우 인증서를 사용할 수 없다. 장치는 "/oic/sec/doxm"의 sct 속성에서 0x8 비트 위치를 0으로 설정하여 이 사실을 알리고 "/oic/sec/csr" 리소스가 존재하지 않는다는 오류를 반환한다.
- 3) CMS는 개인 키와의 연결을 유지하기 위해 발급된 인증서와 CA 체인을 동일한 신임을 사용하여 지정된 장치로 전송해야 한다. 그림 29의 인증서를 전송하는 데 사용되는 자격 증명 유형 ("oic.sec.cred")은 POST의 여러 자격 증명을 CMS에서 장치로 포함하여 역할 인증서를 전송하는 데에도 사용된다. 신원 인증서는 credusage 속성이 "oic.sec.cred.cert"로 설정되고 역할 인증서는 credusage 속성이 "oic.sec.cred.rolecert"로 설정되어 저장된다.



그림 29 - Client-directed Certificate Transfer

CRL 발급의 유일한 전제 조건은 CMS (예 : 허브 또는 스마트 폰)에 해지 인증서를 등록하고 CRL에 서명하고이를 장치로 전송하는 기능이 있어야 한다.

CMS가 CRL을 장치로 보낸다.

아래 나열된 인증서 해지 사유로 인해 각 장치에서 CRL이 업데이트된다.

- 발급자 이름 변경
- 장치와 CA 간의 연결 변경
- 인증서 손상
- 해당 개인 키의 손상이 의심

CRL은 OCF 보안 도메인의 모든 액세스 가능한 장치에 업데이트되어 제공 될 수 있다. 특별한 경우에 장치는 지정된 CMS에 CRL을 요청할 수 있다. CRL을 업데이트하고 제공하는 두 가지 옵션이 있다.

- CMS가 각 장치에 CRL을 푸시
- 각 장치는 주기적으로 CRL 업데이트 요청

클라이언트 지향 모델에 대한 CRL 전송의 순서 흐름은 그림에 설명되어 있다.

- 1) CMS는 CRL 리소스 속성을 검색 할 수 있다.
- 2) 장치가 CMS에 CRL을 보내도록 요청하면 최신 CRL을 장치로 전송해야한다.



그림 30 - Client-directed CRL Transfer

서버 지향 모델에 대한 CRL 전송의 순서 흐름은 그림에 설명되어 있다.

- 1) 장치가 CRL 리소스 속성 "tupdate"를 CMS로 검색한다.
- 2) CMS가 지정된 "tupdate"시간 이후에 업데이트 된 CRL 정보를 인식하면 CRL을 장치로 전송할 수 있다.

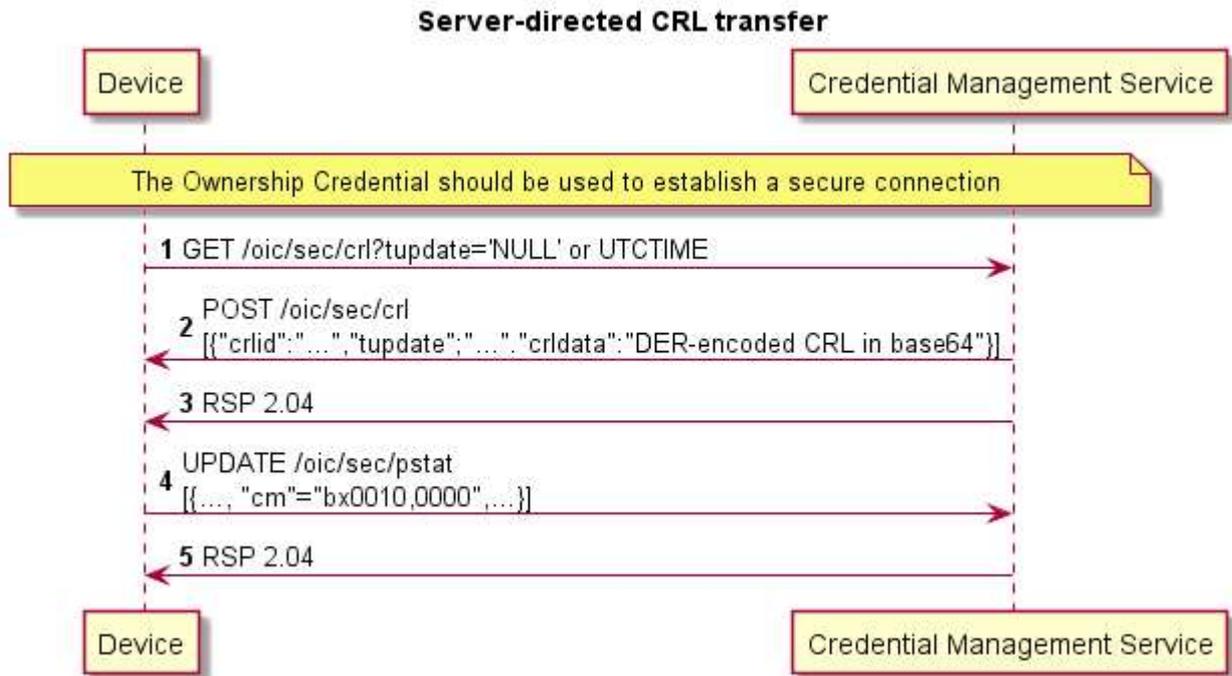


그림 31 - Server-directed CRL Transfer

(바) 장치 인증

클라이언트가 서버의 제한된 리소스에 액세스하는 경우 서버는 클라이언트를 인증해야한다. 클라이언트는 액세스를 요청하면서 서버를 인증해야한다. 클라이언트는 서버가 액세스 제어 결정에 사용할 수 있는 하나 이상의 역할을 주장 할 수도 있다. 인증서로 장치 인증을 수행 할 때 역할이 주장 될 수 있다.

인증에 대칭 키를 사용할 때 서버 장치는 ServerKey를 포함해야한다. 메시지를 교환하고 psk\_identity\_hint를 서버의 장치 ID로 설정하십시오. 클라이언트는 서버의 장치 ID로 설정된 주체 ID와

PSK의 자격 증명 유형을 가지고 있는지 확인해야 한다. 그렇지 않을 경우 클라이언트는 알 수 없는 `_psk_identity` 오류 또는 기타 적절한 오류로 응답해야 한다.

클라이언트가 적절한 PSK 자격 증명을 찾은 경우, 클라이언트로 응답해야 함클라이언트의 장치 ID로 설정된 `psk_identity_hint`가 포함된 교환 메시지. 서버는 일치하는 주제 ID와 유형을 가진 자격 증명을 가지고 있는지 확인해야 한다. 그렇지 않을 경우 서버는 알 수 없는 `_psk_identity` 또는 기타 적절한 오류 코드로 응답해야 한다. 만일 그렇다면, 그것은 DTLS 프로토콜로 계속되어야 하며, 클라이언트와 서버 모두 결과적인 초기 비밀을 계산해야 한다.

인증서를 사용하여 인증 할 때 클라이언트와 서버는 각각 해당 인증 자격 증명에 저장된 인증서 체인을 선택한 인증 암호 제품군의 일부로 포함해야 한다. 각 장치는 피어 장치가 제공 한 인증서 체인의 유효성을 검사해야 한다. 각 인증서 서명은 공개 키가 "oic.sec.cred.trustca"서류와 함께 "/oic/sec/cred"리소스 내에 있을 때까지 확인되어야 한다. "/oic/sec/cred"에있는 자격 증명 리소스는 인증서 경로 유효성 검사를 종료하는 데 사용된다. 또한 유효 기간 및 해지 상태는 위의 모든 인증서에 대해 확인해야 하지만 현재 인증서의 해지 상태 (CRL 또는 OCSP 응답)를 얻지 못하면 다른 모든 확인 검사가 성공하면 인증서 사용을 계속 허용 할 수 있다. 사용 가능한 경우 해지 정보를 사용하여 인증서의 해지 상태를 확인해야 한다. 해지 정보를 참조하는 URL은 인증 기관 (AccessInformationAccess 또는 `crlDistributionPoints` 확장을 통해)에서 검색해야 한다. 다른 메커니즘을 사용하여 CRL 또는 OCSP 응답과 같은 관련 해지 정보를 수집 할 수 있다. 각 장치는 자격 증명의 해당 주제 ID를 사용하여 피어 장치를 식별해야 한다. 디바이스는 IETF RFC 5280의 5.4.2 절에있는 인증서 경로 유효성 검증 알고리즘을 따라야 한다.

- End-Entity가 아닌 모든 인증서에 대해 장치는 기본 제약 조건 확장이 존재하고 확장의 `cA` 부울이 참인지 확인해야 한다. 둘 중 하나가 거짓이면 인증서 체인을 거부해야 한다. `pathLenConstraint` 필드가 있으면 장치는이 인증서와 End-Entity 인증서 사이의 인증서 수가 `pathLenConstraint`보다 작거나 같은지 확인한다. 특히 `pathLenConstraint`가 0 인 경우가 인증서는 End-Entity 인증서 만 발급 할 수 있다. `pathLenConstraint` 필드가 없으면 체인 길이에 제한이 없다.
- End-Entity가 아닌 모든 인증서에 대해 장치는 키 사용 확장이 있고 `keyCertSign` 비트가 선언되어 있는지 확인해야 한다.
- 장치는 Authority Key Identifier 확장을 사용하여 발급 인증서를 빠르게 찾을 수 있다. 장치는이 확장이없는 인증서를 거부해서는 안되며 주제 이름이이 인증서의 발급자 이름과 동일한 발급자 인증서의 공개 키로 유효성 검사를 시도해야 한다.
- 체인의 최종 엔터티 인증서에는 제시된 목적에 적합한 확장 키 사용 (EKU)이 포함되어 있는지 확인해야 한다. EKU 확장이 포함되지 않은 End-Entity 인증서는 어떤 목적에도 유효하지 않으므로 거부해야 한다. 다른 유효한 EKU가있는 경우에도 `anyExtendedKeyUsage` OID (2.5.29.37.0)가 포함 된 모든 인증서를 거부해야 한다.
- 장치는 인증서 체인에 대한 "전 이적 EKU"를 확인해야 한다. 체인의 발급자 인증서 (최종 엔터티가 아닌 모든 인증서)는 모두 인증서 체인이 제시되는 목적에 유효해야 한다. 발급자 인증서는 EKU 확장이 포함되어 있고 해당 목적에 대한 EKU OID가 확장에 나열되어 있거나 EKU 확장이없는 경우 목적에 유효하다. 발급자 인증서에는 인증서 발급 권한이있는 EKU 확장명과 전체 EKU 목록이 포함되어야 한다. EKU 확장이없는 발급자 인증서는 모든 목적에 유효하다. 이것은 EKU 확장이없는 EndEntity 인증서와 다르다.

장치가 내선 번호를 인식하지 못하면 중요 필드를 검사해야 한다. 필드가 참이면 장치는 반드시 인증서를 거부해야 한다. 필드가 FALSE이면 장치는 반드시 확장이없는 것처럼 인증서를 처리하고 그에 따라 진행해야 한다. 이는 체인의 모든 인증서에 적용된다.

인증서 역할 자격 증명을 사용하여 클라이언트에서 서버로의 역할 어설 션에 대해 설명한다. 서버가 인증서 자격 증명 유형을 지원하지 않으면 클라이언트는 인증서로 역할을 주장하려고 시도해서는 안된다. 인증서를 사용한 인증 후 클라이언트는 서버의 역할 리소스를 사용하려는 역할 인증서로 업데이트하여 하나 이상의 역할을 주장 할 수 있다. 역할 자격 증명은 인증서 자격 증명이어야 하며 인증서 체인을 포함해야 한다. 서버는 10.3 절에 명시된대로 각 인증서 체인을 검증해야 한다. 또한 장치 인증에 사용되는 End-Entity 인증서의 공개 키는 모든 역할 (End-Entity) 인증서의 공개 키와 동일해야 한다. 또한 엔터티 인증 및 역할 인증서의 주제 고유 이름이 일치해야 한다. 주장 된 역할은 인증서의 `subjectAltName` 확장으로 인코딩된다. `subjectAltName` 필드는 여러 값을 가질 수 있으므로 단일 인증서로 클라이언트에 적용되는 여러 역할을 인코딩 할 수 있다. 또한 서버는 역할 인증서의 EKU 확장에 1.3.6.1.4.1.44924.1.7 값이 포함되어 인증서가 역할을 주장하는 데 사용될 수 있음을 나타내는지 확인해야 한다. 그림은 클라이언트 장치가 서버에 역할을 부여하는 방법을 설명한다.

## Asserting Certificate Role Credentials

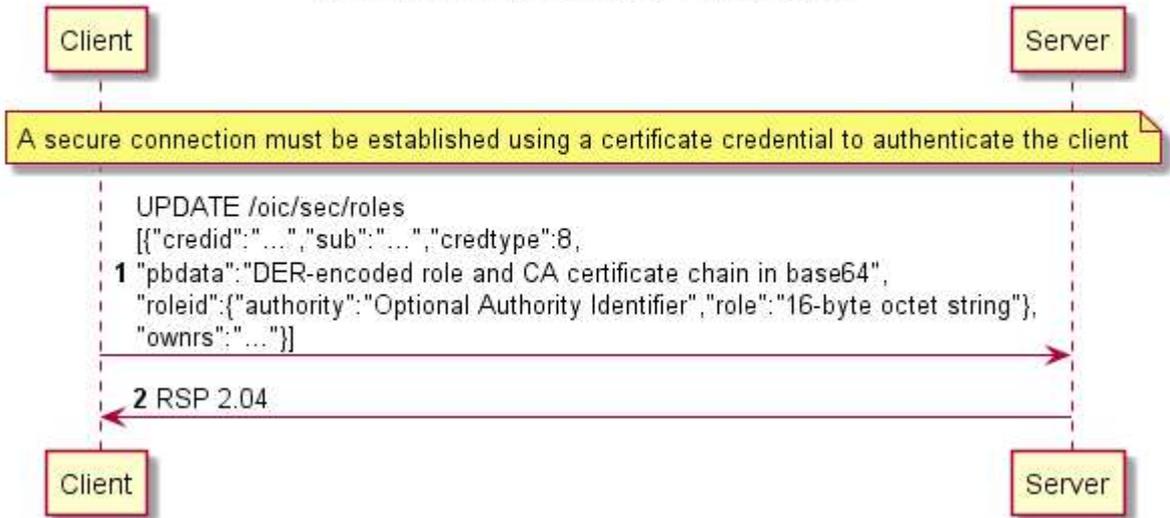


그림 32 - Client-directed Certificate Transfer

- 1) 응답은 성공을 나타내는 "204 No Content" 또는 오류를 나타내는 4xx를 포함해야 한다. 서버가 인증서 자격 증명을 지원하지 않으면 "501 Implemented"가 반환되어야 한다.
- 2) 클라이언트가 주장한 역할은 서버가 선택한 기간 동안 유지 될 수 있다. 기간은 역할 인증서의 유효 기간을 초과하지 않아야 한다. 새로운 CRL 정보를 얻으면 "/oic/sec/roles"의 인증서를 확인하고 인증서가 해지되거나 만료되면 역할이 제거된다.
- 3) 서버는 클라이언트가 역할을 자주 재 검증하는 비용을 피하기 위해 0이 아닌 기간을 선택해야 한다. 서버는 인증서의 유효 기간을 지속 기간으로 사용하여 CMS가 유효 기간을 효과적으로 결정할 수 있도록 하는 것이 좋다.
- 4) 작성 호출로 전송된 데이터 형식은 자격 증명 목록이어야 한다 ( "oic.sec.cred", 표 28 참조). 인증서 유형을 나타내는 credtype 8이 있어야 하며 PrivateData 필드는 없어야 한다. "oic.sec.cred" 개체와 인증서에 중복된 필드의 경우 인증서 값이 유효성 검사에 사용된다. 예를 들어, 자격 증명에 기간 필드가 설정된 경우 서버는 인증서의 유효 기간을 정식으로 취급해야 한다. 역할 ID 데이터 (권한, 역할)와 유사하다.
- 5) 인증서는 base64의 DER 인코딩된 인증서 체인과 같이 인코딩되어야 한다.
- 6) 고객은 "/oic/sec/roles" 리소스를 얻어 이전에 어설션된 역할을 결정할 수 있다. 자격 증명 객체 배열이 반환된다. 현재 연결되고 인증된 고객의 신원에 해당하는 유효한 인증서가 없으면 빈 배열 (예 : [])이 반환된다.

OCF 루트 CA 트러스트 앵커에 연결된 인증서를 사용하는 각 장치는 "oic/sec/cred" 리소스에 OCF 루트 CA 인증서를 안전하게 저장하고 인증서를 변경할 수 없는 강화된 메모리 위치에 이 리소스를 물리적으로 저장해야 한다.

장치는 IETF RFC 5280의 5.4.2 절에 있는 인증서 경로 유효성 검증 알고리즘을 준수해야 한다. 또한 다음은 모범 사례이며 디지털 인증서를 처리하는 모든 OCF 호환 응용 프로그램에서 준수해야 한다.

OCF 호환 애플리케이션은 X.509 인증서에서 notBefore 및 notAfter 필드를 처리할 때 IETF RFC 5280 조항 4.1.2.5, 4.1.2.5.1 및 4.1.2.5.2를 준수해야 한다. 또한 모든 인증서에 대해 notAfter 값은 발급 CA의 notAfter 값을 초과하지 않아야 한다.

응용 프로그램에 의존하는 경우 모든 인증서의 해지 상태를 확인해야 하지만 현재 다른 모든 확인 검사에 성공하면 응용 프로그램에서 인증서의 해지 상태 (CRL 또는 OCSP 응답)를 얻지 못한 경우에도 인증서 사용을 계속 허용할 수 있다.

모든 루트 및 중개 인증 기관 (CA) 인증서에 대해, 장치는 기본 제약 확장이 존재하고, 플래그가 중요하며, 확장의 cA 부울 값이 참인지 확인해야 한다. 이 중 하나라도 거짓이면 인증서 체인이 거부된다. pathLenConstraint 필드가 있으면 장치는 이 인증서와 End-Entity 인증서 사이의 인증서 수가 pathLenConstraint보다 작거나 같은지 확인한다. 특히 pathLenConstraint가 0인 경우 이 인증서는 End-Entity 인증서만 발급할 수 있다. pathLenConstraint 필드가 없으면 체인 길이에 제한이 없다. End-Entity 인증서의 경우 basicConstraints 확장이 존재하는 경우, 중요 플래그로 지정되고, cA 부울 값이 FALSE이며, pathLenConstraint ASN.1 시퀀스를 포함하지 않아야 한다. pathLenConstraint ASN.1 시퀀스가 Integer 값으로 존재하거나 null 값으로 존재하면 End-Entity 인증서는 거부된다. OCF

호환 PKI 구현에서 미래의 유연성을 촉진하기 위해 모든 OCF 호환 루트 CA 인증서는 pathLenConstraint를 포함하지 않아야한다. 이를 통해 루트 CA 트러스트 앵커를 변경하지 않고도 향후 중간 CA의 추가 계층을 구현할 수 있다.

모든 인증서에 대해 장치는 키 사용 확장이 있고 중요도 플래그가 지정되어 있는지 확인해야한다. 루트 및 중간 CA 인증서의 경우, keyCertSign (5) 및 crlSign (6) 비트 만 주장된다. End-Entity 인증서의 경우, digitalSignature (0) 및 keyAgreement (4) 비트만 주장된다.

anyExtendedKeyUsage OID (2.5.29.37.0)를 포함하는 모든 최종 엔티티 인증서는 거부된다. 다양한 TLS 구현과의 호환성을 위해서는 serverAuthentication (1.3.6.1.5.5.7.3.1) 및 clientAuthentication (1.3.6.1.5.5.7.3.2)에 대한 OID가 필요하다. 현재 End-Entity 인증서는 Identity (1.3.6.1.4.1.44924.1.6) 및 Role (1.3.6.1.4.1.44924.1.7) 목적으로 모두 사용할 수 없다. 따라서 두 OID 중 정확히 하나가 존재해야하며 두 OID를 모두 포함하는 EKU 확장명을 가진 End-Entity 인증서는 거부된다.

OCF 루트 CA에 연결되는 최종 엔티티 인증서는 OCF 인증서 정책 OID (1.3.6.1.4.1.51414.0.1.2)로 설정된 하나 이상의 PolicyIdentifierId를 포함해야한다 (1.3.6.1.4.1.51414.0.1.2). 문제가 제기되었다. 추가 제조업체 별 CP OID도 채울 수 있다.

앞에서의 장치 인증 메커니즘은 장치가 "/oic/sec/cred"에 규정 된 기준을 충족하는 다른 장치와 통신할 수 있는 권한이 있음을 나타낸다. "/oic/sec/acl2"리소스는 추가로 특정 리소스에 대한 액세스를 제한하는 데 사용된다. 여기에서는 약간 다른 기준을 사용하는 OCF 클라우드에 대한 장치 인증을 설명한다. OCF 클라우드에 액세스하는 장치는 TLS 세션을 설정해야한다. 상호 인증 된 TLS 세션은 서버 인증서 및 클라이언트 인증서를 사용하여 설정된다. 각 장치는 장치 등록 중에 할당 된 액세스 토큰을 기반으로 식별된다. OCF Cloud에는 액세스 토큰, 사용자 ID 및 장치 ID를 매핑하는 OCF Cloud 연관 테이블이 있다. 장치 등록은 장치가 RFNOP 상태에있을 때 발생한다. 장치 등록 후 업데이트 된 액세스 토큰, 장치 ID 및 사용자 ID는 장치에서 OCF 클라우드와의 후속 연결에 사용된다.

장치는 인증서 기반 자격 증명을 사용하여 TLS 연결을 설정해야한다. 중재자가 장치를 프로비저닝 한 후 연결을 설정해야한다. TLS 세션은 IETF RFC 8323에 지정된대로 장치와 OCF 클라우드간에 설정된다. OCF 클라우드는 장치의 신뢰할 수 있는 항목에있는 신뢰 앵커로 서명 된 인증서를 제공해야한다. 이 cred 항목은 중재자가 구성해야한다. 장치는 장치의 "/oic/sec/cred"리소스 항목에 포함 된 자격 증명을 기반으로 OCF 클라우드의 ID를 확인해야한다. OCF 클라우드는 장치에서 제공 한 제조업체 인증서의 유효성을 검사해야한다. OCF 클라우드 사용자는 장치가 연결하는 OCF 클라우드를 신뢰한다고 가정한다. OCF 클라우드 사용자의 동의없이 OCF 클라우드 연결이 발생하지 않아야한다. OCF 클라우드 사용자는 OCF 클라우드 제공자와 서비스 계약을 체결했거나 제조업체에서 제공 한 OCF 클라우드를 사용한다고 가정한다. 인증이 실패하면 장치가 "oic.r.coapcloudconf"리소스의 "clec"속성을 장치가 지원하는 경우 실패한 상태에 대해 업데이트해야한다. 인증에 성공하면 장치 및 OCF 클라우드는 협상 된 암호 제품군에 따라 암호화 된 링크를 설정해야한다. 그림은 OCF Cloud와의 장치 연결 순서 및 표에 설명된다.

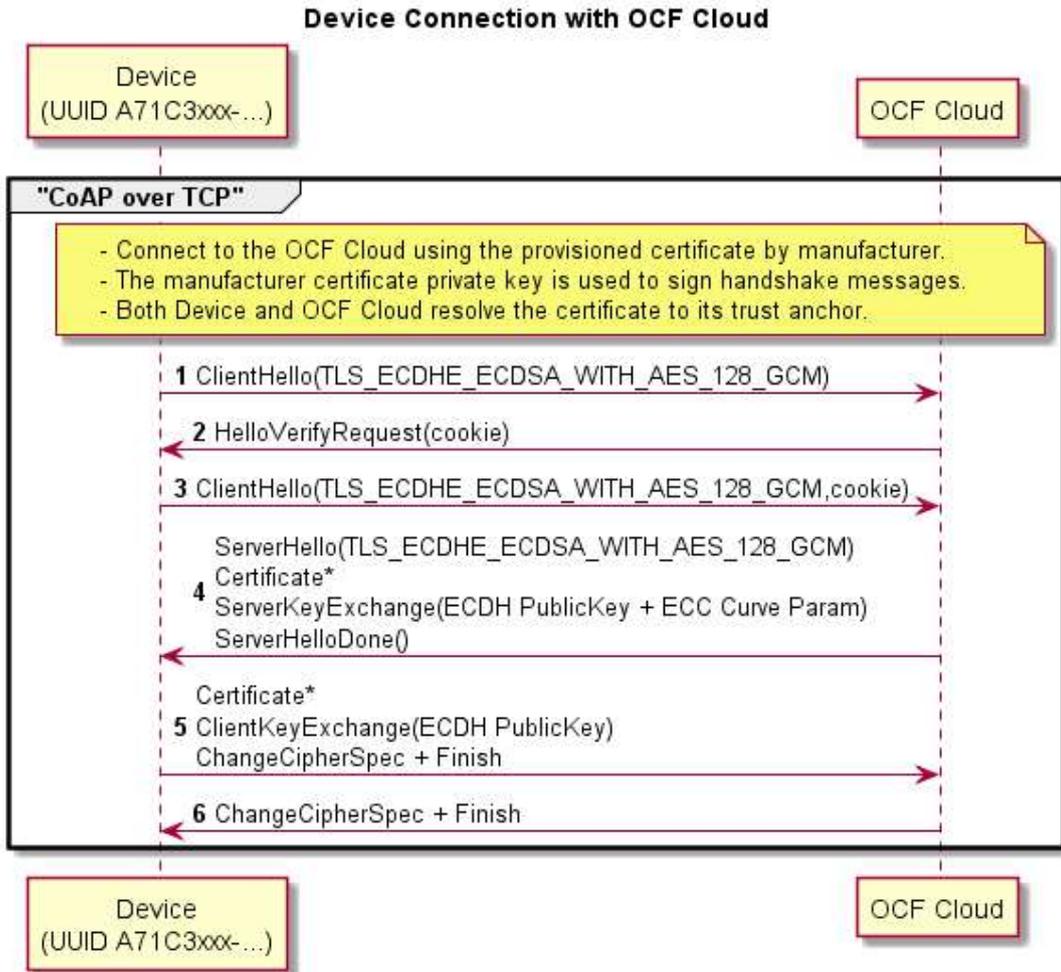


그림 33 - Device connection with OCF Cloud

표 21 - Device connection with the OCF Cloud flow

Steps	Description
1 - 6	TLS connection between the OCF Cloud and Device. The Device's manufacturer certificate may contain data attesting to the Device hardening and security properties

OCF 서버가 OCF 클라우드를 통해 전송된 요청을 수신하면 OCF 서버는 OCF 클라이언트의 ID가 아닌 OCF 클라우드의 ID를 사용하여 해당 요청을 허용한다. OCF 클라우드가 사용자가 OCF 클라우드를 통해 OCF 클라이언트와 OCF 서버간에 의도한 상호 작용만 허용하는 메커니즘이 없고 다른 모든 상호 작용을 거부하는 경우 OCF 클라이언트는 OCF 클라우드를 통해 요청을 제출하여 높은 권한을 얻을 수 있다. 보안 관점에서는 바람직하지 않다. 결과적으로 OCF 클라우드 구현은 OCF 클라우드를 통해 요청을 제출할 때 OCF 클라우드가 OCF 클라이언트가 높은 권한을 얻지 못하게 하는 메커니즘을 제공할 것으로 예상된다.

본 문서에서, 메커니즘의 세부 사항은 구현에 맡겨져 있다. 제조업체 인증서에 대한 보안 고려 사항은 OCF 클라우드를 사용한 장치 인증에도 적용된다. 장치는 IETF RFC 6125에서 정의한대로 서버 ID 인증 요구 사항에 따라 OCF 클라우드의 TLS 인증서를 확인해야 한다. "/oic/d" 리소스의 "uid" 및 "di" 속성 값은 일부 규제 지역에서 개인 식별 정보로 간주될 수 있으며 OCF Cloud는 해당 규제 기관에 적절한 보호 기능을 제공할 것으로 예상된다.

(사) 메시지 무결성과 기밀성

메시지 기밀성과 무결성을 제공하는 보안 메커니즘을 사용하여 클라이언트와 서버 간의 보안 통신이 도청, 변조 또는 메시지 재생으로부터 보호된다.

디바이스는 IETF RFC 6347에 정의된 보안 통신을 위한 DTLS를 지원해야 한다. TCP를 사용하는 디바

이는 IETF RFC 5246에 정의된 보안 통신을 위한 TLS v1.2를 지원해야 한다. OCF 장치는 (D) TLS 버전 1.2 이상을 지원해야 하며 버전 1.1 이하를 지원해서는 안 된다. 이 버전의 보안 문서에는 아직 멀티 캐스트 세션 의미가 정의되어 있지 않다.

클라이언트와 서버 간의 유니 캐스트 메시지의 경우 두 장치 모두 서로를 인증해야 한다. 송신 장치는 선택된 암호 제품군에서 정의한대로 메시지를 암호화 및 인증해야 하며 수신 장치는 메시지를 처리하기 전에 메시지를 확인 및 암호 해독해야 한다.

OCF 클라우드와 장치 간의 메시지는 장치와 OCF 클라우드가 서로를 인증하는 경우에만 교환된다. 비대칭 암호 스위트는 보안 세션을 설정하고 OCF 클라우드와 장치 간의 암호화/암호 해독을 위해 사용되어야 한다. 메시지를 전송하는 OCF 엔드 포인트는 암호화 제품군을 사용하여 메시지를 암호화하고 인증해야 하며 OCF 엔드 포인트는 메시지를 처리하기 전에 메시지를 확인 및 암호 해독해야 한다.

사용 가능한 암호 스위트는 상황에 따라 다를 수 있다. 이 절에는 소유권 이전 및 정상 작업 중에 허용되는 암호 제품군이 나열되어 있다. 다음 RFC는 OCF에 사용된 암호 제품군에 대한 추가 정보를 제공한다.

IETF RFC 4279 : (D) TLS에서 사전 공유 키 (PSK) 사용을 지정한다.

IETF RFC 4492 : (D) TLS에서 타원 곡선 암호화 사용 지정

IETF RFC 5489 : ECDHE (Elliptic Curve Diffie-Hellman) 및 PSK를 사용하는 암호 제품군 사용을 지정한다.

IETF RFC 6655 및 IETF RFC 7251 : ECDHE와 함께 AES-CCM 모드 암호화 제품군을 지정한다.

#### - Just Works 방식 암호 스위트

Just Works OTM은 다음과 같은 (D) TLS 암호화 제품군을 사용할 수 있다.

- TLS\_ECDH\_ANON\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_ECDH\_ANON\_WITH\_AES\_256\_CBC\_SHA256

Just Works OTM을 지원하는 모든 장치는 다음을 구현해야 한다.

- TLS\_ECDH\_ANON\_WITH\_AES\_128\_CBC\_SHA256 (값 0xFF00)

Just Works OTM을 지원하는 모든 장치는 다음을 구현해야 한다.

- TLS\_ECDH\_ANON\_WITH\_AES\_256\_CBC\_SHA256 (값 0xFF01)

#### - 랜덤 PIN 방법 암호 스위트

무작위 PIN 기반 OTM은 다음(D)을 사용할 수 있다. TLS 암호 세트.

- TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_ECDHE\_PSK\_WITH\_AES\_256\_CBC\_SHA256,

All Devices supporting Random Pin 기본 OTM은 구현되어야 한다 :

- TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256

#### - 인증서 방법 암호 스위트

제조업체 인증서 기반 OTM은 다음(D)을 사용할 수 있는 TLS 암호 세트.

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM,
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM

다음 원곡선 사용:

secp256r1 (IETF RFC 4492 참조) 제조업체 인증서 기반 OTM을 지원하는 모든 장치는 다음을 구현해야 한다.

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8

제조업체를 지원하는 장치

인증서 기반 OTM은 다음을 구현해야 한다.

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM

- 대칭 키에 대한 암호 스위트

다음 암호 집합은 (D)에 대해 정의된다.PSK를 이용한 TLS 통신

- TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_ECDHE\_PSK\_WITH\_AES\_256\_CBC\_SHA256,
- TLS\_PSK\_WITH\_AES\_128\_CCM\_8, (\* 8개의 OCTET 인증 태그\*)
- TLS\_PSK\_WITH\_AES\_256\_CCM\_8,
- TLS\_PSK\_WITH\_AES\_128\_CCM, (\* 16개의 OCTET 인증 태그 \*)
- TLS\_PSK\_WITH\_AES\_256\_CCM,

모든 CCM 기반 암호 스위트도 인증에 HMAC-SHA-256을 사용한다.

모든 기기는 다음을 구현해야 한다.

- TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256,

기기는 다음을 구현해야 한다.

- TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256,
- TLS\_ECDHE\_PSK\_WITH\_AES\_256\_CBC\_SHA256,
- TLS\_PSK\_WITH\_AES\_128\_CCM\_8,
- TLS\_PSK\_WITH\_AES\_256\_CCM\_8,
- TLS\_PSK\_WITH\_AES\_128\_CCM,
- TLS\_PSK\_WITH\_AES\_256\_CCM

- 비대칭 자격 증명을위한 암호 제품군

다음 암호 집합은 (D)에 대해 정의된다.비대칭 키 또는 인증서를 사용한 TLS 통신:

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM

다음 원곡선 사용:

secp256r1 (IETF RFC 4492 참조) 비대칭 자격 증명을 지원하는 모든 장치는 다음을 구현해야 한다.

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8

비대칭 기능을 지원하는 모든 장치

자격 증명은 다음을 구현해야 한다.

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM,
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM

- OCF 클라우드 자격 증명을위한 암호화 제품군

인증서와의 TLS 통신에 대해 다음과 같은 암호 집합이 정의된다.

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384,
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384,
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384

OCF 클라우드 인증서 자격 증명을 지원하는 모든 장치는 다음을 구현해야 한다.

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256

OCF 클라우드 인증서 자격 증명을 지원하는 모든 장치는 다음을 구현해야 한다.

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256,
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384,
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384,
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384

(아) 접근 제어

서버는 애플리케이션 자원을 요청자에게 노출시키기 전에 애플리케이션 자원에 대한 액세스 제어를 시행한다. 서버의 보안 계층은 보안 포트를 통해 액세스가 수신되면 요청자를 인증한다. "주체"라고 하는 인증된 요청자는 요청자의 신원, 역할을 지정하는 ACL 항목을 일치 시키거나 제목 와일드 카드를 사용하여 인증된 요청자를 일치시킬 수 있다. 요청이 보안되지 않은 포트를 통해 도착하면 익명 요청자의 제목 와일드 카드 일치를 사용하는 ACL 정책 만 허용된다. 요청한 리소스가 ACL 항목과 일치하지 않으면 액세스가 거부된다.

ACL 리소스를 프로비저닝하기 전에 보안 가상 리소스에 대한 액세스 권한이 부여 될 수 있는 장치 Onboarding과 관련된 문서화 된 예외가 있다. 2 세대 ACL (즉, "/oic/sec/acl2")에는 ACE2 액세스 정책이 적용되는 리소스를 일치시키기 위해 리소스 참조 배열을 사용하는 리소스 일치 알고리즘을 사용하는 ACE2 (액세스 제어 항목) 배열이 포함되어 있다. 매칭은 ACE2 "resources"속성 (13 절 참조)의 값을 요청된 리소스와 비교하는 것으로 구성된다. 리소스는 두 가지 방식으로 일치한다.

- 1) 호스트 참조 ( "href")
- 2) 리소스 와일드 카드 ( "wc")

ACE2 매칭 요소에 존재할 경우, Host Reference (href) 속성은 리소스 매칭에 사용된다.

- href 속성은 존재하는 경우 리소스 이름과 정확히 일치하는 것을 찾는 데 사용된다.

리소스 와일드 카드 일치가 존재하는 경우, "oic.sec.ace2.resource-ref"구조에 포함 된 와일드 카드 속성을 사용하여 와일드 카드 (wc) 표현식을 사용하여 여러 리소스를 일치시켜야 한다. "oic.sec.ace2.resource-ref"구조에 포함 된 와일드 카드 속성을 사용하여 와일드 카드 표현식을 사용하여 여러 자원을 일치시킬 수 있다. 와일드 카드 일치 문자열은 다음 표에 정의되어 있다.

표 22 - ACE2 Wildcard Matching Strings Description

String	Description
"+"	Shall match all Discoverable Non-Configuration Resources which expose at least one Secure OCF Endpoint.
"_"	Shall match all Discoverable Non-Configuration Resources which expose at least one Unsecure OCF Endpoint.
"**"	Shall match all Non-Configuration Resources.

참고 검색 가능한 리소스는 "/oic/res"리소스에 나타나는 반면, 검색 할 수 없는 리소스는 다른 컬렉션 리소스에는 나타날 수 있지만/res 컬렉션에는 나타나지 않는다.

ACE2 "resources"속성에 여러 항목이 포함 된 경우 각 배열 요소에 대해 논리 OR이 적용된다. 예를 들어, "resources"속성의 첫 번째 배열 요소에 "href"= "/a/light"가 포함되고 "resources"속성의 두 번째 배열 요소에 "href"= "/a/led"가 포함 된 경우 Resources 두 "href"기준 중 하나와 일치하는 것은 일치하는 리소스 세트에 포함된다.

## 예제 1 리소스 일치를 위한 JSON

```
{
//Matches Resources named "/x/door1" or "/x/door2"
  "resources":[
    {
      "href":"/x/door1"
    },
    {
      "href":"/x/door2"
    }
  ]
}
```

## Example 2 JSON for Resource matching

```
{
// Matches all Resources
  "resources":[
    {
      "wc":"*"
    }
  ]
}
```

ACE 제목이 와일드 카드 문자열 "\*"로 지정되면 모든 요청자가 일치한다. OCF 서버는 OCF 클라이언트를 인증 할 수 있지만 필수는 아닙니다.

예 : 와일드 카드 일치위한 JSON

//인증 된 모든 주제와 일치하며 기밀 보호가 설정되어 있다.

```
"subject" : {
  "conntype" : "auth-crypt"
}
```

//인증되지 않고 기밀 보호 기능이없는 모든 주제와 일치한다.

```
"subject" : {
  "conntype" : "anon-clear"
}
```

ACE 주체가 역할로 지정된 경우, 요청자는

- 1) 요청자가 대칭 키 자격 증명으로 인증했으며 자격 증명 리소스에 자격 증명 항목의 roleid 속성에 해당 역할이 있거나
- 2) 요청자가 인증서로 인증을 받았으며 평가시 요청자의 인증서 공개 키를 사용하여 역할 리소스에 유효한 역할 인증서가 있다.

OCF 서버는 다음 순서로 일치하는 ACE2 일치 알고리즘을 적용해야 한다.

- 1) "/oic/sec/sacl"리소스가 존재하고 서명 확인에 성공하면 ACE2 항목은 3단계의 로컬 ACE2 항목 집합에 영향을 미친다. 서버는 업데이트 후 최소한 한 번 서명을 확인해야 한다.
- 2) 로컬 "/oic/sec/acl2"리소스가 일치하도록 ACE2 항목을 제공한다.
- 3) 이러한 모든 기준이 충족되면 액세스 권한이 부여된다.
  - a) 요청자는 ACE2 "대상"속성과 일치한다.
  - b) 요청 된 자원은 ACE2 자원 속성과 일치하며 요청 된 자원은 로컬 서버에 존재해야한다.
  - c) "기간"속성 제약이 충족되어야 한다.
  - d) "허가"속성 제약이 적용되어야 한다.

여러 ACE2 항목이 리소스 요청과 일치하면 일치하는 모든 ACE에 대한 권한 통합이 부여된 유효 권한을 정의한다. 예 : Perm1 = CR --- 인 경우; Perm2 = -UDN; 그런 다음 UNION (Perm1, Perm2) = CRUDN이다.

서버는 부여된 유효 권한에 따라 액세스를 시행해야 한다. 링크가 포함된 리소스에 대한 일괄 요청에는 연결된 리소스에 액세스 할 때 추가로 고려해야 한다.

이 clause는 원래 클라이언트 요청을 허용하는 Colection과 일치하는 ACE가 있다고 가정 할 때 서버가 클라이언트에서 해당 서버에 호스트된 Colection으로 배치 OCF 인터페이스 요청을 수신 할 때 발생하는 추가 권한 부여 프로세스를 처리한다. 이 clause를 위해 Colection을 호스트하는 서버를 "Colection Host"라고 한다. 추가 권한 부여 프로세스는 링크된 자원이 colection host에서 호스트되는지 또는 링크된 자원이 다른 서버에서 호스트되는지에 따라 다르다.

- Colection 호스트에서 호스트된 링크된 자원에 대해 생성된 각 요청에 대해 Colection 호스트는 ACE2 일치 알고리즘을 적용하여 링크된 자원이 생성된 요청을 처리할 수 있는지 여부를 다음과 같이 설명해야 한다.
- 요청자는 원래 사용자 요청을 보낸 사용자 이어야한다.
- 요청된 자원은 링크된 자원이어야하며, 다음 중 하나 이상을 사용하여 일치시켜야한다.
- 링크된 자원과 일치하는 자원 와일드 카드
- ACE2의 "resources"배열에있는 "href"속성과 연결된 Resource의 로컬 경로가 정확히 일치한다.
- ACE2의 "resources"배열에있는 "href"속성과 연결된 Resource의 전체 URI가 정확히 일치한다.

참고로 링크된 자원의 전체 URI는 링크의 "앵커"특성 (있는 경우)과 링크의 "href"특성을 연결하여 얻는다. 로컬 경로는 전체 URI에서 결정될 수 있다.

링크된 자원이 생성된 요청을 처리할 수 없는 경우 Colection host는 해당 케이스를 배치 OCF 인터페이스에 지정된대로 원래 클라이언트 요청에 대해 집계된 응답을 작성할 때 요청을 처리할 수 없는 링크된 자원으로 처리해야 한다.

사) [정책활용] 스마트팜 표준용어 활성화 방안 제안 (1건)

전통적인 농업 기술과 ICT가 융합된 스마트팜 기술은 농업뿐만 아니라 과학기술, 정보통신, 유통과 가공 등 어느 하나의 특정 분야에 국한되지 않고 광범위한 분야에 도입되어 발전하고 있다. 각의 분야의 기술적인 전문성에 기반하여 상호 경쟁적인 체제를 만들어가는 것은 국제적으로 그 중요성이 날로 부각되고 있는 스마트팜 기술의 발전 속도를 한층 빨라지게 하는 효과를 기대할 수 있을 것이다. 하지만, 스마트팜 기술과 관련된 용어에 대하여 각 분야에서 서로 달리 해석하거나 이종의 기술분야로 분류하는 등 서로간의 이해의 차이로 인하여 불필요한 혼란 또한 가중되고 있으며 이는 스마트팜의 확산과 보급에 장애물로 작용하고 있다.

이를 위하여 본 연구과제에서는 스마트팜 분야에서 사용되는 다양한 용어들을 분석하고 표준용어를 개발하고 있으며 이의 일환으로 2019년 한국정보통신기술협회(TTA) 단체표준으로서 스마트 온실 분야에 대한 용어 정의 표준(129개 용어, TTAK.KO-10.1164-part1) 개발을 완료하였으며, 패밀리 표준으로서 스마트 축사 분야에 대한 용어 정의 표준(85개 용어, TTAK.KO-10.1164-part 2)을 개발하였으며 2020년 12월 최종 승인될 예정이다.

하지만, 앞서 언급한 바와 같이 스마트팜은 다양한 기술이 융합된 산업으로서, 농가와 농가의 종사자는 물론 스마트팜 기자재 제조업체 및 시설업자, 서비스 제공 사업자 그리고 각 부처의 정책담당 등의 이해관계자에게 공통적으로 적용될 수 있는 표준 용어를 개발하는 것이 무엇보다 중요하다 할 수 있다. 이에 본 연구에서는 스마트농업 관련 정책과 제도, 연구개발을 수행하는 정부 및 기관과 함께 스마트팜 용어의 표준화 추진에 대한 타당성과 현장 적용성 등에 대한 논의를 지속적으로 진행하고 있으며, 본 연구를 통해 개발된 스마트 온실 및 축사 분야 용어 정의 표준의 활성화 방안 마련을 위하여 노력하고 있다.

이를 위해 2019년에는 농촌진흥청 정책부서를 통해 스마트 온실 분야 용어 정의 표준안에 대한 검토를 의뢰(공문)하고 접수된 의견을 적극 반영하여 표준 규격에 최종 반영한 바 있으며, 2020년에는 10월에 개발을 완료하여 12월에 최종 표준 승인 예정인 스마트 축사 용어 정의 표준안에 대한 검토와 함께 각 부처 및 기관별로 다양한 형태로 개발되고 있는 용어 정의를 통합하여 범부처 간 표준화된 용어의 사용을 적극 권장하고 이를 활성화하는 방안에 대한 협의를 11월 중에 요청하는 것을 검토하고 있다.

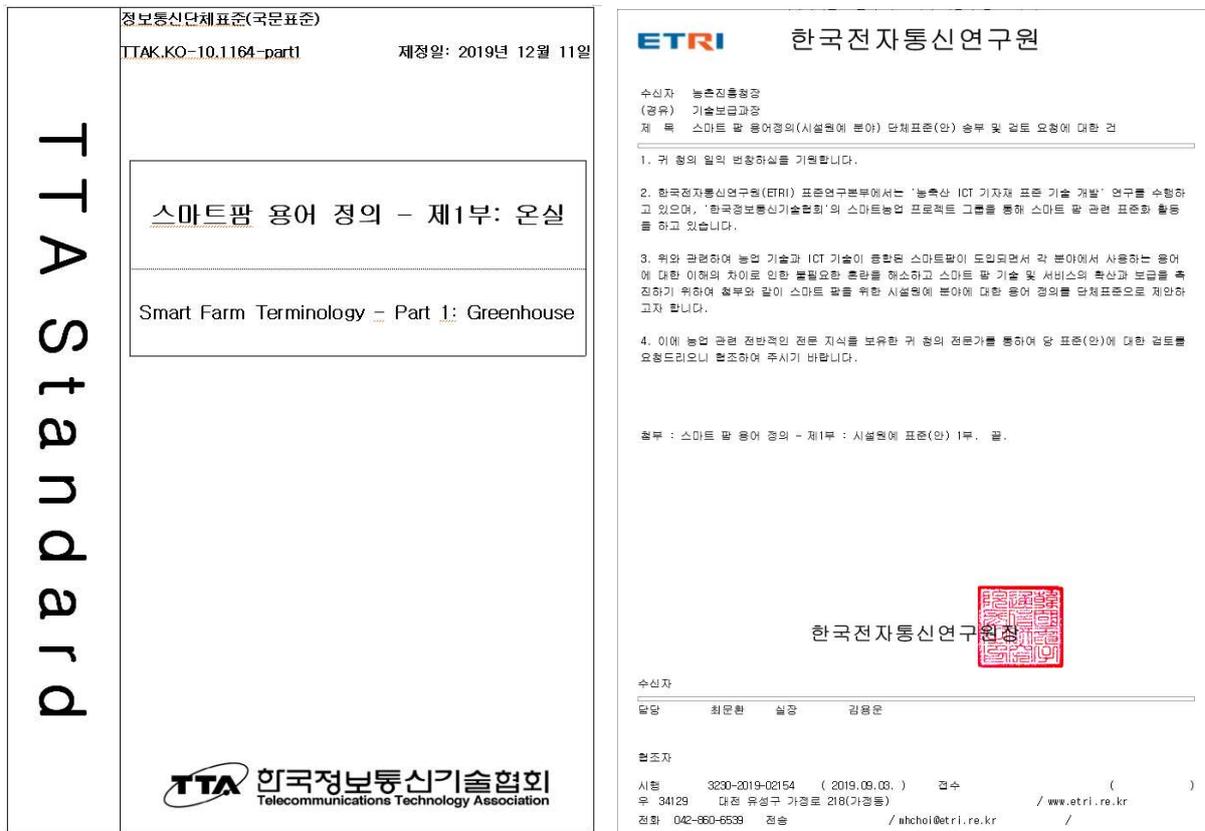


그림 1 - 스마트 온실 용어 표준안 및 검토 요청 공문

### 3. 연구개발과제의 수행 결과 및 목표 달성 정도

#### 1) 연구수행 결과

##### (1) 정성적 연구개발성과

---

###### 가. 국내단체표준 개발

- TTA PG426을 통한 국내 단체표준 신규 과제제안 (11종)
  - 노지스마트팜에서 사용되는 용어정의를 국내 단체표준 신규 과제제안
    - 스마트팜 용어정의 - 제3부: 노지/과수 (2021-P2458)
  - 우사에서 사용되는 다양한 사양관리기들을 클라우드로 관제하기 위한 OpenAPI 2.0 기반 프로파일을 국내 단체표준 신규 과제제안
    - CoAP 기반 스마트팜 관리 및 제어 프레임워크-제6부: 우사사양관리기 (2021-P2439)
- TTA PG426을 통한 국내 단체표준 제정 추진 (22종)
  - ITU-T SG20에서 국제표준으로 승인된 Y.4466에 대한 부합화 추진
    - 스마트 온실 서비스 프레임워크
  - 돈사에서 사용되는 사양관리기(11종)를 클라우드로 관제하기 위한 OpenAPI 2.0 기반 프로파일을 국내단체표준으로 채택
    - CoAP 기반 스마트팜 관리 및 제어 프레임워크-제4부: 돈사사양관리기
  - 돈사에서 사용되는 환경관리기(10종)를 클라우드로 관제하기 위한 OpenAPI 2.0 기반 프로파일을 국내단체표준으로 채택
    - CoAP기반 스마트팜 관리 및 제어 프레임워크-제5부: 돈사환경관리기
- TTA PG426을 통한 국내 단체표준 개정 추진 (1종)
  - 무선통신기술(RF)을 활용하여 스마트팜 관제를 위한 통신프로토콜 국내단체표준의 개정
    - UHF 대역 400 MHz 무선링크 기반 스마트 온실 제어 프로토콜

###### 나. 국가표준 개발

- ISO TC23/SC19 국내 전문위원회를 통해, ISO 11783-3과 11783-4의 KS 부합화 추진
  - 농림업용 트랙터 및 기계류 - 직렬 제어 및 통신 데이터 네트워크 - 제3부: 데이터링크 계층(KS B ISO 11783-3:2018)
  - 농림업용 트랙터 및 기계류 - 직렬 제어 및 통신 데이터 네트워크 - 제4부: 네트워크 계층(KS B ISO 11783-4:2011)
- ※ 해당 건들은 ISO TC23 SC19 국내 전문위원회에서 현재 검토중이며 2022년 제정예정

###### 다. 국제표준 개발

- 국제표준(IEC) 승인 1건
  - IEC TC100 TA-17 원격회의와 국내전문위원회 활동을 통해 추진된, 차량용 인포테인먼트 서비스의 국제 표준승인 ('21.8)
    - IEC 63246-1, Configurable car infotainment services (CCIS) - Part 1: General
- IoT 기반의 스마트축산 서비스를 제공하기 위한 요구사항과 기능구조에 관한 표준화 진행 (ITU-T Y.IoT-SLF) (4건)
  - Y.IoT-SLF- Proposed modifications on Clause 7, "SLF model" (2월)
  - Y.IoT-SLF -Proposed modifications on clause 8 (6월)
  - Y.IoT-SLF- Proposed modification on clause 8 and 9 (10월)

- 
- Y.IoT-SLF- Proposed modifications to draft Recommendation Y.IoT-SLF(12월)

- 스마트팜 유즈케이스에 관한 표준화 진행 (ITU-T Y.Sup.Smart Agri-usecase) (2건)
  - Y.Sup.SmartAgri-usecases: Proposed new usecase “IoT cloud based smart agriculture” (6월)
  - Proposed items to be discussed for Y.Sup.SmartAgri-usecases (10월)
- ※ 상기 개발중인 표준안은 2022년에 국제표준 승인 예정
- 국제표준의장단 수임과 연계한 주도적인 ITU-T 국제표준 개발 (2명)
  - ITU-T Y.IoT-SLF 표준 개발 에디터 (박주영, 최문환)
  - ITU-T Y.Sup.Smart Agri-usecase표준 개발 에디터 (박주영, 최문환)

## 라. PoC S/W 구현

- 본 사업에서 개발하고 있는 국내 표준 기술들의 구현 가능성을 판단하고 표준 개선을 위해, 클라우드 기반의 스마트팜 관제 서비스의 프로토타입 구현 (2건)
    - ICT 기자재 표준화를 선도하기 위해, 기존의 RS485기반의 농축산 ICT 기자재가 아닌 클라우드-지향적인 기자재 관련 표준화를 TTA에서 진행 중
    - 국내 단체표준으로 추진중인 클라우드-지향적인 기자재를 위한 통신 프로토콜의 PoC 추진
      - MQTT기반 스마트팜 운영노드/클라우드 연동 프로그램 (8월)
      - MQTT기반 스마트팜 센서/구동기/제어기 노드 연동 프로그램(8월)
  - 상기 개발한 2건의 S/W에 대해 국가공인기관(KTC) 시험성적서 발급 추진 (1건)
    - MQTT기반 스마트팜 장치 OCF표준기술검증 PoC 시스템 (1건, 8월)
      - [S/W 시험성적서] MQTT기반 스마트팜 장치 OCF표준기술검증 PoC 시스템
- 

## (2) 정량적 연구개발성과(해당 시 작성하며, 연구개발과제의 특성에 따라 수정이 가능합니다)

### 가. IPR 확보

- AI 및 빅데이터 기반의 스마트팜 서비스에 관한 요구가 급증함에 따른, 1) 스마트팜 데이터 수집, 2) 데이터의 전송 방법에 관한 신규 기술 설계를 통한 특허 출원 및 등록 (출원: 16, 등록:5)
  - 국내특허 등록 (5건)
    - IoT 노드 자동 구성 시스템 및 방법 (출원:'17, 등록:'19)
    - 통합 제어 라우팅 시스템 및 그 방법 (출원:'18, 등록:'19)
    - 센서 교체가 가능한 범용 센서 노드 및 그 구성 방법 (출원:'18, 등록:'19)
    - LCP를 이용하는 스마트온실 시스템 (출원:'18, 등록:'20)
    - 스마트팜 데이터셋 유효성 검증을 위한 수집 데이터 선별 장치 및 그 방법 (출원:'20, 등록:'22)
  - 국내특허 출원 (5건)
    - IoT 디바이스 인터페이스 시스템 및 그 방법 (출원:'18)
    - 스마트팜 게이트웨이 (출원:'20)
    - 스마트팜 관리 방법 및 장치 (출원:'20)
    - 스마트팜 환경 및 생육 데이터 저장 시스템 (출원:'20)
    - 타임라인 기반의 전염병 추적조사 시스템 및 그 방법 (출원:'20)
    - 스마트팜 기자재 공간 위치 인식 장치 및 그 방법 (출원:'20)
    - 스마트 팜에서의 비료 사용량 처방 시스템 (출원:'21)
    - 스마트팜 데이터 전송 장치 및 그 방법 (출원:'21)
    - 영상장치를 이용한 식물 생육 측정 시스템 및 그 방법 (출원:'21)
    - 컨테이너를 활용한 조사료 재배 장치 및 그 시스템 (출원:'22)
    - 야생동물 식별을 통해 학습효과 방지가 가능한 야생동물 퇴치 시스템 및 그 방법 (출원:'22)

### 나. 논문(비SCI)

○ 스마트팜 표준화 동향, 시설원예/노지/축산 관련 연구 결과물에 대한 논문지 및 국내/국제 컨퍼런스 논문 제출 (18건)

- A STUDY ON MACHINE LEARNING WEB SERVICE USING SPREADSHEETS ('17)
- 스마트팜 기술 개발 및 표준화 동향('17)
- Implementation of docker-based smart greenhouse data analysis platfor ('18)
- LCP for Managing Smart Greenhouse ('18)
- Implementation of greenhouse service control protocol using Python on Raspberry PI ('19)
- 스마트온실에서 RS485 MODBUS기반 IoT 디바이스 속성 정보 자동구성 및 데이터 송수신 최적화방안 ('19)
- ETRI Insight 표준화동향: 스마트팜 (12월) ('19)
- ITU-T SG20 스마트팜 표준화 동향 ('20)
- IoT 기반 스마트온실 서비스 프레임워크 표준 소개 ('20)
- 스마트팜 장치 표준 기술 분석 및 검증 ('20)
- Implementation of Smart Farm Devices using Open Source Software ('21)
- 스마트축산 표준화 동향 ('21)
- 기상 IoT 센서 데이터를 활용한 서리 예측 연구 ('21)
- Understanding IoT Climate Data based Predictive Model for Outdoor Smart Farm ('21)
- 스마트 농업을 위한 유즈케이스 표준개발 연구 ('22)
- 데이터기반스마트팜전망 ('22)
- IoT 기반 스마트축산을 위한 서비스 프레임워크 표준화 연구 ('22)
- IoT 기반 강아지 급이기 설계 및 구현 ('22)

#### 다. 홍보·전시

○ 스마트팜 ICT 표준화 기술 홍보

- 스마트팜 ICT 표준화와 미래 기술 발전 분야에 대해 전문 언론 매체(한국농기계신문)를 통한 홍보 (14건)
  - 한국농기계신문, 스마트팜 ICT 표준화 과정 시급하다 ('17)
  - 정책브리핑, 과기정통부, 글로벌 ICT 표준컨퍼런스(GISC2018)개최 ('18)
  - 농기계신문, 농업분야 산업혁명 주도하기 위해선 표준화가 필수요소이다. ('19)
  - 농기계신문, 전화 요금 17만 원의 비밀... ('19)
  - 농기계신문, 스마트팜 인공지능 준비를 위한 매우 중요한 시점 ('19)
  - 한국농기계신문, 2020년, 스마트팜 표준화 재정비가 필요한 해 ('20)
  - 한국농기계신문, 가축방역통합시스템 세계 기술 선도 ('20)
  - 한국농기계신문, 유통구조 개선, 농가소득 안정화 촉진 ('20)
  - 한국농기계신문, 포스트-코로나 시대, 이젠 기후변화다 ('20)
  - 한국농기계신문, 홍보1-스마트팜 데이터 활성화를 위한 제언 ('21)
  - 한국농기계신문, 홍보2-스마트팜 글로벌화를 위한 제언 ('21)
  - 한국농기계신문, 홍보3- 기후변화 바퀴 속도를 늦출 수 있는 건 오직 탄소 절감뿐 ('21)
  - 한국농기계신문, 우리나라 스마트 농기계의 경쟁력 강화를 위한 제언 ('22)
  - 농축산기계신문, “농기계의 통신기술, 체계적 접근해야” ('22)

#### 라. 도서저술

○ 시설원예 기술 및 표준화 개론을 수록한 시설원예학 공저 (3건)

- [서적] 스마트팜개론, ISBN 979-11-6257-127-9 ('20)
- [서적] 삼고 시설원예학 스마트팜 개론, ISBN 979-89-7187-265-9 ('21)
- [서적] 스마트팜개론 (시설원예), ISBN 979-11-6257-182-8 ('21)

○ 스마트팜 표준화 전략맵 작성 (2건)

- 스마트팜분야 국내/국제 표준화에 대한 전략수립을 위한 전략맵 작성
  - [보고서] TTA 표준화전략맵 v.2022 (스마트팜)
  - [보고서] TTA 표준화전략맵 v.2019 (스마트팜)

※ 표준화전략맵 스마트팜 분과장 수임 (박주영)

○ 스마트팜 표준화 활용을 위한 표준 해설 및 활용가이드라인 개정 (1건)

- [보고서] 스마트팜 표준화 활용을 위한 표준 해설 및 활용가이드라인

※ 해설 및 활용 가이드라인 분과장 수입 (박주영)

마. 표준화지원

○ 스마트팜 기자재 정보관리 관련 국내단체표준 개발 지원(3건)

- 스마트팜산업체들의 요구를 수용하여, 효율적으로 스마트팜 기자재들의 정보관리 방식을 제안하고 이를 농업실용화재단 포럼을 통해 단체표준으로 채택
  - 스마트온실 ICT 기자재 MIB 정보체계 - 제1부 일반사항('20)
  - 스마트온실 ICT 기자재 MIB 정보체계 - 제2부 온실통합제어기 ('20)
  - 스마트온실 ICT 기자재 MIB 정보체계 - 제3부 양액기('20)

○ 스마트팜 국가표준 확산을 위한 기술지원 ('20~'22)

- 스마트팜 관련 국가표준 (KS X3267, KS X3268, KS X 3269, KS X3265, KS X 3266 등) 총 5건에 대한 국내 기업체들의 이해도 증진과 수정 요구사항 수렴을 위한 주도적인 지원

< 정량적 연구개발성과표 >

(단위 : 건, 천원)

성과지표명	연도		1단계	2단계	3단계	계	가중치 (%)
			('17~'18)	('19~'20)	('21~'22)		
특허출원 (건)		목표(단계별)	4	4	2	10	10
		실적(누적)	5	6	5	16	
특허등록 (건)		목표(단계별)	0	2	4	6	10
		실적(누적)	0	4	1	5	
단체표준제안 (총)		목표(단계별)	30	20	10	60	10
		실적(누적)	42	28	22	92	
단체표준채택 (총)		목표(단계별)	15	25	20	60	10
		실적(누적)	24	34	34	92	
단체표준개정 (건)		목표(단계별)	3	2	0	5	10
		실적(누적)	0	2	1	3	
국가표준제정 (총)		목표(단계별)	3	3	4	10	20
		실적(누적)	23	0	2	25	
국제기고서 (건)		목표(단계별)	6	8	8	22	10
		실적(누적)	11	14	13	38	
국제표준채택 (건)		목표(단계별)	0	1	2	3	20
		실적(누적)	0	1	2	3	
SW구현 (건)		목표(단계별)	1	2	1	4	
		실적(누적)	3	5	3	11	
비SCI논문 (건)		목표(단계별)	2	3	2	7	
		실적(누적)	2	6	8	16	
학술발표 (건)		목표(단계별)	0	0	0	0	
		실적(누적)	0	0	0	0	
교육지도 (건)		목표(단계별)	0	0	0	0	
		실적(누적)	14	14	5	33	
정책활용 (건)		목표(단계별)	1	1	0	2	
		실적(누적)	1	0	0	1	
홍보전시 (건)		목표(단계별)	1	2	1	4	
		실적(누적)	2	7	5	14	
기타 (건)		목표(단계별)	0	0	0	0	
		실적(누적)	0	32	6	38	
계			66	73	54	193	
			127	153	107	387	

\* 1] 전담기관 등록·기탁 지표: 논문[에스시아이 Expanded(SCIE), 비SCIE, 평균Impact Factor(IF)], 특허, 보고서원문, 연구시설·장비, 기술요약정보, 저작권(소프트웨어, 서적 등), 생명자원(생명정보, 생물자원), 표준화(국내, 국제), 화합물, 신물질 등을 말하며, 논문, 학술발표, 특허의 경우 목표 대비 실적은 기재하지 않아도 됩니다.

\* 2] 연구개발과제 특성 반영 지표: 기술실시(이전), 기술료, 사업화(투자실적, 제품화, 매출액, 수출액, 고용창출, 고용효과, 투자유치), 비용 절감, 기술(제품)인증, 시제품 제작 및 인증, 신기술지정, 무역수지개선, 경제적 파급효과, 산업지원(기술지도, 교육지도, 인력양성(전문 연구인력, 산업연구인력, 졸업자수, 취업, 연수프로그램 등), 법령 반영, 정책활용, 설계 기준 반영,

타 연구개발사업에의 활용, 기술무역, 홍보(전시), 국제화 협력, 포상 및 수상, 기타 연구개발 활용 중 선택하여 기재합니다  
(연구개발과제 특성별로 고유한 성과지표를 추가할 수 있습니다).

(3) 세부 정량적 연구개발성과(해당되는 항목만 선택하여 작성하되, 증빙자료를 별도 첨부해야 합니다)

[과학적 성과]

□ 논문(국내외 전문 학술지) 게재

번호	논문명	학술지명	주저자명	호	국명	발행기관	SCIE 여부 (SCIE/비SCIE)	게재일	등록번호 (ISSN)	기여율
1	A STUDY ON MACHINE LEARNING WEB SERVICE USING SPREADSHEETS	ICTC 2017	윤치열		대한민국	한국통신학회	비SCIE	2017-10-18		100
2	스마트팜 기술 개발 및 표준화 동향	한국통신학회	윤치열		대한민국	한국통신학회	비SCIE	2017-11-11		100
3	Implementation of docker-based smart greenhouse data analysis platfor	ICTC 2018	현욱		대한민국	한국통신학회	비SCIE	2018-10-16		100
4	LCP for Managing Smart Greenhouse	ICTC 2018	박주영		대한민국	한국통신학회	비SCIE	2018-10-16		100
5	Implementation of greenhouse service control protocol using Python on Raspberry PI	ICACT 2019	현욱		대한민국	한국통신학회	비SCIE	2019-02-18		100
6	스마트온실에서 RS485 MODBUS기반 IoT 디바이스 속성 정보 자동구성 및 데이터 송수신 최적화방안	한국통신학회	허미영		대한민국	한국통신학회	비SCIE	2019-06-20		100
7	ETRI Insight 표준화동향: 스마트팜 (12월)	E T R I Insight	윤성현		대한민국	한국전자통신연구원	비SCIE	2019-10-18		100
8	ITU-T SG20 스마트팜 표준화 동향	SEP inside	박주영		대한민국	한국특허전략개발원	비SCIE	2020-06-01		100
9	IoT 기반 스마트온실 서비스 프레임워크 표준 소개	한국통신학회	최문환		대한민국	한국통신학회	비SCIE	2020-08-13		100
10	스마트팜 장치 표준 기술 분석 및 검증	한국통신학회	윤성현		대한민국	한국통신학회	비SCIE	2020-08-13		100
11	Implementation of Smart Farm Devices using Open Source Software	ICACT 2021	윤성현		대한민국	한국통신학회	비SCIE	2021-02-09		100
12	스마트축산 표준화 동향	JCCI2021	박주영		대한민국	한국통신학회	비SCIE	2021-04-28		100
13	기상 IoT 센서 데이터를 활용한 서리 예측 연구	한국인공지능학술대회	박주영		대한민국	한국인공지능학술대회	비SCIE	2021-09-29		100
14	Understanding IoT Climate Data based Predictive Model for Outdoor Smart Farm	ICTC2021	박주영		대한민국	한국통신학회	비SCIE	2021-10-21		100
15	스마트 농업을 위한 유즈케이스 표준개발 연구	한국통신학회	최문환		대한민국	한국통신학회	비SCIE	2022-02-11		100
16	데이터기반스마트팜전망	한국통신학회	윤성현		대한민국	한국통신학회	비SCIE	2022-02-11		100
17	IoT 기반 스마트축산을 위한 서비스 프레임워크	한국통신학회	박주영		대한민국	한국통신학회	비SCIE	2022-02-11		100

	표준화 연구									
18	IoT 기반 강아지 급이기 설계 및 구현	한국멀티미디어학회	박주영		대한민국	한국멀티미디어학회	비SCIE	2022-05-13		100

□ 국내 및 국제 학술회의 발표

번호	회의 명칭	발표자	발표 일시	장소	국명

□ 기술 요약 정보

연도	기술명	요약 내용	기술 완성도	등록 번호	활용 여부	미활용사유	연구개발기관 외 활용여부	허용방식

□ 보고서 원문

연도	보고서 구분	발간일	등록 번호
2018	TTA 표준화전략맵 v.2018	2018.11.07	
2021	TTA 표준화전략맵 v.2022	2021.12.22	

□ 생명자원(생물자원, 생명정보)/화합물

번호	생명자원(생물자원, 생명정보)/화합물 명	등록/기탁 번호	등록/기탁 기관	발생 연도

[기술적 성과]

□ 지식재산권(특허, 실용신안, 의장, 디자인, 상표, 규격, 신품종, 프로그램)

번호	지식재산권 등 명칭 (건별 각각 기재)	국명	출원				등록			기여율	활용 여부
			출원인	출원일	출원 번호	등록 번호	등록인	등록일	등록 번호		
1	타임라인 기반의 전염병 추적조사 시스템 및 그 방법	대한민국	한국전자통신연구원	20.07.29	2020-0094830					100	
2	스마트팜 기자재 공간 위치 인식 장치 및 그 방법	대한민국	한국전자통신연구원	20.08.31	2020-0110293					100	
3	스마트 팜에서의 비료 사용량 처방 시스템	대한민국	한국전자통신연구원	21.01.29	2021-0013481					100	
4	스마트팜 데이터 전송 장치 및 그 방법	대한민국	한국전자통신연구원	21.10.21	2021-0141242					100	
5	영상장치를 이용한 식물 생육 측정 시스템 및 그 방법	대한민국	한국전자통신연구원	21.12.07	2021-0174102					100	
6	IoT 디바이스 인터페이스 시스템 및 그 방법	대한민국	한국전자통신연구원	18.11.28	2018-0149215					100	
7	스마트팜 환경 및 생육 데이터 저장 시스템	대한민국	한국전자통신연구원	20.05.11	2020-0056168					100	
8	스마트팜 게이트웨이	대한민국	한국전자통신연구원	20.02.06	2020-0014068					100	
9	스마트팜 관리 방법 및 장치	대한민국	한국전자통신연구원	20.02.25	2020-0023039					100	
10	컨테이너를 활용한 조사료 재배 장치 및 그 시스템	대한민국	한국전자통신연구원	22.10.14	2022-0132648					100	
11	야생동물 식별을 통해 학습효과 방지가 가능한 야생동물 퇴치 시스템 및 그 방법	대한민국	한국전자통신연구원	22.10.14	2022-0132653					100	
12	IoT 노드 자동 구성 시스템 및 방법	대한민국	한국전자통신연구원	17.11.28	2017-0160966	한국전자통신연구원	19.07.26	2006219		100	

			원				원			
13	통합 제어 라우팅 시스템 및 그 방법	대한민국	한국전자통신연구원	18.07.09	2018-0079456		한국전자통신연구원	19.08.22	2015532	100
14	센서 교체가 가능한 범용 센서 노드 및 그 구성 방법	대한민국	한국전자통신연구원	18.11.28	2018-0149214		한국전자통신연구원	19.09.24	2026968	100
15	LCP를 이용하는 스마트온실 시스템	대한민국	한국전자통신연구원	18.11.28	2018-0149213		한국전자통신연구원	20.01.14	2068131	100
16	스마트팜 데이터셋 유효성 검증을 위한 수집 데이터 선별 장치 및 그 방법	대한민국	한국전자통신연구원	20.12.11	2020-0173163		한국전자통신연구원	22.10.18	2457824	100

### ○ 지식재산권 활용 유형

※ 활용의 경우 현재 활용 유형에 √ 표시, 미활용의 경우 향후 활용 예정 유형에 √ 표시합니다(최대 3개 중복선택 가능).

번호	제품화	방어	전용실시	통상실시	무상실시	매매/양도	상호실시	담보대출	투자	기타

### □ 저작권(소프트웨어, 서적 등)

번호	저작권명	창작일	저작자명	등록일	등록 번호	저작권자명	기여율
1	영상 기반 생육 센서 하드웨어 및 해석 시스템 시제품 구현 및 표준 지원 프로그램	2022-09-01	윤성현, 박주영, 최문환	2022-10-19	C-2022-040036	한국전자통신연구원	100
2	애플리케이션 기반 스마트팜 센서/구동기/제어기 노드 연동 프로그램	2021-05-21	윤성현, 박주영, 최문환	2021-08-17	C-2021-032558	한국전자통신연구원	100
3	애플리케이션 기반 스마트팜 운영노드/클라우드 연동 프로그램	2021-05-21	윤성현, 박주영, 최문환	2021-08-17	C-2021-032559	한국전자통신연구원	100
4	아이오티(IoT) 센서 기반 스마트팜 노드 모니터링 프로그램	2020-06-02	윤성현, 박주영, 이창규, 최문환	2020-07-14	C-2020-023220	한국전자통신연구원	100
5	아이오티(IoT) 센서기반 스마트팜 노드 프로그램	2020-06-02	윤성현, 박주영, 이창규, 최문환	2020-07-14	C-2020-023222	한국전자통신연구원	100
6	아이오티(IoT) 센서 기반 스마트팜 모니터링 웹저인터페이스(UI)	2020-06-02	윤성현, 박주영, 이창규, 최문환	2020-07-14	C-2020-023223	한국전자통신연구원	100
7	경량형 제어 프로토콜 기반 스마트온실 구동기 및 센서 노드 제어 프로그램	2019-03-01	현욱, 박주영, 허미영	2019-05-02	C-2019-012508	한국전자통신연구원	100
8	경량형 제어 프로토콜 기반 스마트온실 통합 서비스 제어 프로그램	2019-03-01	현욱, 박주영, 허미영	2019-05-02	C-2019-012509	한국전자통신연구원	100
9	아이피(IP)기반 스마트 온실 노드 제어 프로토타입 및 큐티(Qt)기반 응용	2018-08-01	현욱, 박주영, 허미영	2018-10-12	C-2018-027052	한국전자통신연구원	100
10	농업빅데이터 시각화 분석을 위한 데이터 수집기	2018-08-01	현욱, 박주영, 허미영	2018-10-12	C-2018-027053	한국전자통신연구원	100
11	스마트온실 노드제어 및 서비스제어 프로토콜 프로토타입	2018-08-01	현욱, 박주영, 허미영	2018-10-12	C-2018-027054	한국전자통신연구원	100

### □ 신기술 지정

번호	명칭	출원일	고시일	보호 기간	지정 번호

### □ 기술 및 제품 인증

번호	인증 분야	인증 기관	인증 내용		인증 획득일	국가명
			인증명	인증 번호		

□ 표준화

○ 국내표준

번호	인증구분 <sup>1)</sup>	인증여부 <sup>2)</sup>	표준명	표준인증기구명	제안주체	표준종류 <sup>3)</sup>	제안/인증일자
1	단체규격	제안	스마트 온실 구동기 메타데이터	TTA	한국전자통신연구원	신규	2018-06-27
2	단체규격	제안	스마트 온실 센서 메타데이터	TTA	한국전자통신연구원	신규	2018-06-27
3	단체규격	제안	스마트 온실 센서/구동기 및 제어기 간 RS485 기반 모드버스(MODBUS) 인터페이스	TTA	한국전자통신연구원	신규	2018-06-27
4	단체규격	제안	스마트온실 관제를 위한 경량형 제어 프로토콜	TTA	한국전자통신연구원	신규	2018-12-19
5	단체규격	제안	[개정] 스마트 온실을 위한 원격 감시용 스마트 영상 장치	TTA	한국전자통신연구원	개정	2019-12-11
6	단체규격	제안	스마트축사 외기 센서 메타데이터 구조	TTA	한국전자통신연구원	신규	2019-12-11
7	단체규격	제안	스마트축사 내기 센서 메타데이터 구조	TTA	한국전자통신연구원	신규	2019-12-11
8	단체규격	제안	경량형 제어프로토콜(LCP)을 위한 IP 적응 계층(IPAL) 동작 요구사항	TTA	한국전자통신연구원	신규	2019-12-11
9	단체규격	제안	스마트온실 서비스 제어 프로토콜: 기본 서비스 절차	TTA	한국전자통신연구원	신규	2019-12-11
10	단체규격	제안	스마트팜 용어 정의 - 제1부: 온실	TTA	한국전자통신연구원	신규	2019-12-11
11	단체규격	제안	[개정] 네트워크기반 스마트팜의 개요	TTA	한국전자통신연구원	신규	2020-12-10
12	단체규격	제안	스마트팜 용어 정의 - 제2부: 축사	TTA	한국전자통신연구원	신규	2020-12-10
13	단체규격	제안	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제3부: 창 개폐기 프로파일	TTA	한국전자통신연구원	신규	2020-12-10
14	단체규격	제안	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제2부: 센서 프로파일	TTA	한국전자통신연구원	신규	2020-12-10
15	단체규격	제안	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제1부: 베이스 프로토콜	TTA	한국전자통신연구원	신규	2020-12-10
16	단체규격	제안	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제5부: 돈사 환경관리기	TTA	한국전자통신연구원	신규	2021-12-08
17	단체규격	제안	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제4부: 돈사 사양관리기	TTA	한국전자통신연구원	신규	2021-12-08
18	단체규격	제안	[개정] UHF 대역 400 MHz 무선링크 기반 스마트 온실 제어 프로토콜	TTA	한국전자통신연구원	개정	2021-12-08
19	단체규격	제안	스마트 온실 서비스 프레임워크	TTA	한국전자통신연구원	신규	2021-12-08
20	단체규격	제안	스마트팜 용어 정의 - 제3부: 노지 및 과수원	TTA	한국전자통신연구원	신규	2022-12-07
21	단체규격	제안	CoAP 기반 스마트팜 관리 및 제어 프레임워크 - 제6부: 우사 사양관리기	TTA	한국전자통신연구원	신규	2022-12-07
22	한국산업규격(KS)	제안	스마트 온실 센서/구동기 노드 및 온실 통합 제어기 간 RS485 기반 모드버스 인터페이스	국가기술표준원	한국전자통신연구원	신규	2018-12-23
23	한국산업규격(KS)	제안	스마트 온실 구동기 메타데이터	국가기술표준원	한국전자통신연구원	신규	2018-12-23
24	한국산업규격(KS)	제안	스마트 온실 구동기 메타데이터	국가기술표준원	한국전자통신연구원	신규	2018-12-23
25	한국산업규격(KS)	제안	농림업용 트랙터 및 기계류 — 직렬 제어 및 통신 데이터 네트워크 — 제3부: 데이터링크 계층	국가기술표준원	한국전자통신연구원	신규	2023 공시예정
26	한국산업규격(KS)	제안	농림업용 트랙터 및 기계류 — 직렬 제어 및 통신 데이터 네트워크 — 제4부: 네트워크 계층	국가기술표준원	한국전자통신연구원	신규	2023 공시예정

\* 1) 한국산업규격(KS) 표준, 단체규격 등에서 해당하는 사항을 기재합니다.

\* 2) 제안 또는 인증 중 해당하는 사항을 기재합니다.

\* 3) 신규 또는 개정 중 해당하는 사항을 기재합니다.

○ 국제표준

번호	표준화 단계구분 <sup>1)</sup>	표준명	표준기구명 <sup>2)</sup>	표준분과명	의장단 활동여부	표준특허 추진여부	표준개발 방식 <sup>3)</sup>	제안자	표준화 번호	제안일자
1	IS	ITU-T Y.4466 Framework of smart greenhouse service * 2019...	ITU-T	Q4/20	Y	N	IS	박주영	Y.4466	2020-01-13
2	IS	IEC 63246-1 Configurable car infotainment services (CCIS) - Part 1: General	IEC	TA17	Y	N	IS	박주영	IEC 63246-1	2021-08-18
3	IS	ITU-T Y.4482:2022, "Requirements and framework for smart liv...	ITU-T	Q2/20	Y	N	IS	박주영	Y.4482	2022-08-30
4	WD	ITU-T Y.ISG-fr : Proposed texts for scope, terms, definitions and conventions	ITU-T	Q4/20	Y	N	IS	박주영	C9	2017-07-17
5	WD	Y.ISG-fr: Proposed texts for terms definition	ITU-T	Q4/20	Y	N	IS	박주영	C193	2017-09-04
6	WD	Y.ISG-fr: Proposed text for information and control flow of smart greenhouse	ITU-T	Q4/20	Y	N	IS	박주영	C203	2017-09-04
7	WD	Proposed work plan for two stale work items in Q.4/20	ITU-T	Q4/20	N	N	TR	박주영	C263	2017-09-04
8	WD	Y.ISG-fr - Proposed modification on functional requirements	ITU-T	Q4/20	Y	N	IS	박주영	C21	2018-01-15
9	WD	Y.ISG-fr-detail description of functional element in referen...	ITU-T	Q4/20	Y	N	IS	박주영	C20	2018-01-18
10	WD	Y.ISG-fr-Proposed text for detail description of SG service framework	ITU-T	Q4/20	Y	N	IS	박주영	C365R2	2018-01-18
11	CD	Y.ISG-fr-Proposed text for reference architecture	ITU-T	Q4/20	Y	N	IS	박주영	C388R1	2018-05-06
12	CD	Y.ISG-fr - Proposed modification of functional requirements	ITU-T	Q4/20	Y	N	IS	박주영	C21	2018-05-07
13	CD	Y.ISG-fr: Proposed updated text of clause 8	ITU-T	Q4/20	Y	N	IS	박주영	C508	2018-12-07
14	CD	Y.ISG-fr: Proposed updated text of clause 9	ITU-T	Q4/20	Y	N	IS	박주영	C509	2018-12-07
15	CD	ISO/IEC JTC1/JETI (Emerging Technology and Innovation) Technology Trend Report (TTR)	ITU-T	JETI	N	N	TR	박주영	AG2_NO 320	2019-07-08
16	DIS	Y.ISG-fr: proposed modifications on clause 10	ITU-T	Q4/20	Y	N	IS	박주영	C37	2019-07-22
17	DIS	Y.ISG-fr: proposed modifications on Appendix II	ITU-T	Q4/20	Y	N	IS	박주영	C38	2019-07-22
18	DIS	Y.ISG-fr: proposed modifications on clause 9	ITU-T	Q4/20	Y	N	IS	박주영	C36	2019-07-22
19	DIS	Y.ISG-fr: proposed modifications on clause 8	ITU-T	Q4/20	Y	N	IS	박주영	C35	2019-07-22
20	DIS	Y.ISG-fr: proposed modifications on clause 7	ITU-T	Q4/20	Y	N	IS	박주영	C34	2019-07-22
21	DIS	Y.ISG-fr: proposed new terms and editorial changes from clause 3 to clause 5	ITU-T	Q4/20	Y	N	IS	박주영	C32	2019-07-22
22	DIS	Y.ISG-fr: proposed	ITU-T	Q4/20	Y	N	IS	박주영	C33	2019-07-22

번호	표준화 단계구분 <sup>1)</sup>	표준명	표준기구명 <sup>2)</sup>	표준분과명	의장단 활동여부	표준특허 추진여부	표준개발 방식 <sup>3)</sup>	제안자	표준화 번호	제안일자
		modifications on clause 6								
23	DIS	Y.ISG-FR: Proposed overall modifications	ITU-T	Q4/20	Y	N	IS	박주영	C46	2019-09-23
24	FDIS	Y.ISG-fr : Proposed text for consent	ITU-T	Q4/20	Y	N	IS	박주영	C655	2019-10-31
25	CD	Y.IoT-SLF; Proposed modification of scope and structure	ITU-T	Q2/20	Y	N	IS	박주영	C7	2020-04-02
26	CD	Y.IoT-SLF: Proposed modification on SLF model and associated community view in clause 7	ITU-T	Q2/20	Y	N	IS	박주영	C777	2020-07-16
27	CD	Y.IoT-SLF: Proposed requirements and security consideration of SLF service	ITU-T	Q2/20	Y	N	IS	박주영	C778	2020-07-16
28	WD	Y.SmartAgri-usecases: Proposed modification on use case of smart greenhouse in clause 8.1	ITU-T	Q2/20	Y	N	IS	최문환	DOC10	2020-10-20
29	CD	Y.IoT-SLF: Proposed modifications on Clause 7, "SLF model"	ITU-T	Q2/20	Y	N	IS	박주영	DPC10	2021-02-22
30	CD	Y.IoT-SLF - Proposed modifications on clause 8	ITU-T	Q2/20	Y	N	IS	박주영	DOC08	2021-06-28
31	WD	Y.Sup.SmartAgri-usecases : Proposed new usecase "IoT cloud based smart agriculture"	ITU-T	Q2/20	Y	N	IS	최문환	DOC11	2021-06-28
32	WD	Proposed items to be discussed for Y.Sup.SmartAgri-usecases	ITU-T	Q2/20	Y	N	IS	최문환	C977R1	2021-10-11
33	CD	Y.IoT-SLF - Proposed modifications on clause 8 and 9	ITU-T	Q2/20	Y	N	IS	박주영	DOC08	2021-10-11
34	CD	Contribution on ITU-T Y.SLF "Framework and capabilities for smart livestock farming based on Internet of things" based on the revision of all text (yellow marks)	ITU-T	Q2/20	Y	N	IS	박주영	C972	2021-12-14
35	DIS	Proposed modification on draft Recommendation ITU-T Y.IoT-SLF, "Requirements and framework for smart livestock farming based on the Internet of things"	ITU-T	Q2/20	Y	N	IS	박주영	C5	2022-04-07
36	FDIS	Proposed modification on draft Recommendation ITU-T Y.IoT-SLF, "Requirements and framework for smart livestock farming based on the Internet of things" for consent	ITU-T	Q2/20	Y	N	IS	박주영	C083R1	2022-07-18
37	CD	Y.Sup.SmartAgri-usecases : Proposal for a new use case "Connected agricultural machinery-Tractor"	ITU-T	Q2/20	Y	N	IS	최문환	C104R1	2022-07-28
38	CD	Proposal for a new use case "UAV-assisted pest control"	ITU-T	Q2/20	Y	N	IS	최문환	C102R1	2022-07-28
39	CD	Y.Sup.SmartAgri-usecases	ITU-T	Q2/20	Y	N	IS	최문환	C104R1	2022-07-28

번호	표준화 단계구분 <sup>1)</sup>	표준명	표준기구명 <sup>2)</sup>	표준분과명	의장단 활동여부	표준특허 추진여부	표준개발 방식 <sup>3)</sup>	제안자	표준화 번호	제안일자
		: Proposal for a new use case "Connected agricultural machinery-Tractor"								
40	CD	Y.Sup.SmartAgri-usecases : Proposal for a new use case "Automated dairy farm-IoT based milking system"	ITU-T	Q2/20	Y	N	IS	최문환	C105R1	2022-07-28
41	CD	Y.Sup.SmartAgri-usecases : Proposal for updated texts from caluse 8.4 to 8.7	ITU-T	Q2/20	Y	N	IS	최문환	C14	2022-10-13

- \* 1) 국제표준 단계 중 신규 작업항목 제안(NP), 국제표준초안(WD), 위원회안(CD), 국제표준안(DIS), 최종국제표준안(FDIS), 국제표준(IS) 중 해당하는 사항을 기재합니다.
- \* 2) 국제표준화기구(ISO), 국제전기기술위원회(IEC), 공동기술위원회1(JTC1) 중 해당하는 사항을 기재합니다.
- \* 3) 국제표준(IS), 기술시방서(TS), 기술보고서(TR), 공개활용규격(PAS), 기타 중 해당하는 사항을 기재합니다.



고용 창출

순번	사업화명	사업화 업체	고용창출 인원(명)		합계
			yyyy년	yyyy년	
합계					

고용 효과

구분			고용 효과(명)	
고용 효과	개발 전	연구인력		
		생산인력		
	개발 후	연구인력		
		생산인력		

비용 절감(누적)

순번	사업화명	발생연도	산정 방법	비용 절감액(천원)
합계				

경제적 파급 효과

(단위: 천원/년)

구분	사업화명	수입 대체	수출 증대	매출 증대	생산성 향상	고용 창출 (인력 양성 수)	기타
해당 연도							
기대 목표							

산업 지원(기술지도)

순번	내용	기간	참석 대상	장소	인원

기술 무역

(단위: 천원)

번호	계약 연월	계약 기술명	계약 업체명	계약업체 국가	기 징수액	총 계약액	해당 연도 징수액	향후 예정액	수출/수입

[사회적 성과]

법령 반영

번호	구분 (법률/시행령)	활용 구분 (제정/개정)	명 칭	해당 조항	시행일	관리 부처	제정/개정 내용

□ 정책활용 내용

번호	구분 (제안/채택)	정책명	관련 기관 (담당 부서)	활용 연도	채택 내용
1	제안	스마트팜 용어정의(시설원에분 야) 단체표준안 송부 및 검토 요청에 대한 건	농촌진흥청 (기술보급과)	2019	채택

□ 설계 기준/설명서(시방서)/지침/안내서에 반영

번호	구분 (설계 기준/설명서/지침/안내서)	활용 구분 (신규/개선)	설계 기준/설명서/ 지침/안내서 명칭	반영일	반영 내용

□ 전문 연구 인력 양성

번호	분류	기준 연도	현황													
			학위별				성별		지역별							
			박사	석사	학사	기타	남	여	수도권	충청권	영남권	호남권	기타			

□ 산업 기술 인력 양성

번호	프로그램명	프로그램 내용	교육 기관	교육 개최 횟수	총 교육 시간	총 교육 인원

□ 다른 국가연구개발사업에의 활용

번호	중앙행정기관명	사업명	연구개발과제명	연구책임자	연구개발비

□ 국제화 협력성과

번호	구분 (유치/파견)	기간	국가	학위	전공	내용

□ 홍보 실적

번호	홍보 유형	매체명	제목	홍보일

□ 포상 및 수상 실적

번호	종류	포상명	포상 내용	포상 대상	포상일	포상 기관
1	포상	장관상	스마트팜 확산 기 여에 따른 포상	개인	2018.12.31	농림축산식품부
2	수상	IEC 1960 Award	IEC 1960 Award	개인	2022.06.30.	IEC

[인프라 성과]

□ 연구시설·장비

구축기관	연구시설/ 연구장비명	규격 (모델명)	개발여부 (○/×)	연구시설·장비 종합정보시스템* 등록여부	연구시설·장비 종합정보시스템* 등록번호	구축일자 (YY.MM.DD)	구축비용 (천원)	비고 (설치 장소)

\* 「과학기술기초법 시행령」 제42조제4항제2호에 따른 연구시설·장비 종합정보시스템을 의미합니다.

[그 밖의 성과](해당 시 작성합니다)

---

해당 없음

---

(4) 계획하지 않은 성과 및 관련 분야 기여사항(해당 시 작성합니다)

---

- 스마트팜 ICT기자재 국가표준 확산 지원 사업\*) 지원
  - 본 과제를 통해 개발한 국가표준 (KSX3267, 3268, 3269)의 확산을 위한 지원사업에 전문가로 참여
  - 관련 농산업체 개발자들과의 면담을 통한, 현장 의견 수렴과 선도 표준 요구사항 도출 (클라우드 기반 스마트팜 관제, 식물 생육 센서, 축사 환경 센서 등)
  - 관련 업계 전문가들로 구성된 컨설턴트 정기교육(월1회)에 지속 참석을 통한, 표준 해석 지원

\* 스마트팜 ICT기자재 국가표준 확산 지원 사업

- 스마트팜 표준의 고도화 및 표준 제·개정 연계 컨설팅, 제품개선, 검정 바우처 등 표준 적용·확산을 위한 단계별 지원
    - 참여업체 중심으로 품질향상 및 호환성 증대
    - 표준규격에 기반한 스마트팜 산업 생태계 구축 교두보 마련
-

## 2) 목표 달성 수준

추진 목표	달성 내용	달성도(%)
○ 농축산 ICT 부품 및 기자재 국내단체표준 개발(누적 60종 이상) 및 스마트팜 데이터 전송/제어 국내단체표준 개발 (3종 이상)	○ 스마트팜 관제 프로토콜 등 국내 단체표준 제안(92종), 국내 단체표준 제정 (92종)	○ 100%
○ 농축산 ICT 기자재 국내단체표준 개정 (5건 이상)	○ 영상장치, 스마트팜 구조 등 국내 단체표준 개정 (3건)	○ 60%
○ 국가표준(KSC)개발 (8종 이상)	○ RS485기반의 스마트팜 관제프로토콜, 스마트팜 센서 및 구동기 메타데이터, 농업용 기계류 통신 프로토콜 관련 국가 표준개발 (25종)	○ 100%
○ 스마트 온실/축산/유통 관련 국제표준 개발 (3종 이상)	○ ITU-T 및 IEC를 통한 국제표준 (IS) 3건 개발 및 관련 국제 표준 기고서 38건 채택	○ 100%
○ PoC (Proof of Concept) 기능 구현을 통한 표준 검증	○ 스마트팜 표준 기술을 검증하기 위한 IoT 기반의 스마트팜 관제 프로토콜, 인공지능 생육감지 프로그램 등 11건 등록	○ 100%

## 4. 목표 미달 시 원인분석

### 1) 목표 미달 원인(사유) 자체분석 내용

---

- 농축산 ICT 기자재 국내단체표준 개정
    - 국내단체표준의 개정 건의 경우, 표준 제정 이후 일정 시간 이후 이해 당사자간의 의견 수렴에 따라 시급성을 요하는 표준을 대상으로 개정 작업을 수행함
    - 본 사업 수행 기간 중에 개정보다는 신규 제정에 대한 이해당사자들의 의견에 따라, 신규 제정 업무에 치중하였음
- 

### 2) 자체 보완활동

---

- 농축산 ICT 기자재 국내단체표준 개정
    - 지속적인 표준 이해 당사자들의 의견 청취와 표준의 노후에 따른 개정 요구 발생시 적극 대응하여 표준 개정 작업에 임할 예정임
- 

### 3) 연구개발 과정의 성실성

---

- 국내단체표준을 개정하기 위해서는 일반적으로 표준 제정 이후 일정 시간 이후(2~3년), 이해 당사자간의 의견 수렴에 따라 시급성을 요하는 표준을 대상으로 개정 작업을 수행하고 있음. 하지만 스마트팜 관련 표준화가 아직은 초기 단계이기 때문에 표준의 개정보다는 필요로 하는 신규 제정 표준 요구가 상대적으로 높았음. 특히 최근 클라우드 기반의 스마트팜 관제 플랫폼에 대한 현장의 요구가 있었기에, 시급성을 띄는 기존의 표준 개정건에 대해서는 개정 작업을 수행하였으나, 그 밖의 것들에 대해서는 클라우드 기반의 시설원예/축사 관제 서비스에 관한 신규 제정에 치중하였음  
**(국내단체표준제정 목표: 60종, 실적: 92종)**
- 

## 5. 연구개발성과의 관련 분야에 대한 기여 정도

본 연구 사업이 진행되기 전에는 스마트팜 관련 국내 표준화 활동이 매우 저조하였으나, 본 연구 사업을 통하여 스마트팜 관련 국내 단체표준화 활동의 기반 마련에 기여함.

본 연구 사업의 시작 시점에서 초기 단계였던 국제표준화 기구의 의장단(에디터)직을 수임하면서, 국내 스마트팜 관련 기술의 국제 표준화 교두보 마련에 적극 기여함.

국내 전문지 혹은 세미나 등을 통하여 관련 표준화 절차 및 표준 기술 소개를 정기적으로 수행함으로써, 관련 산학연 표준 전문가 양성에 기여함.

---

## 6. 연구개발성과의 관리 및 활용 계획

본 연구 사업을 통해 축적된 노하우와 국내/국제 표준화 기반을 바탕으로, 후속 과제 등을 통하여 본 연구 사업을 통해 발생한 연구개발 성과물의 개선에 노력할 계획임.

< 연구개발성과 활용계획표(예시) >

구분(정량 및 정성적 성과 항목)		연구개발 종료 후 5년 이내	
국외논문	SCIE		
	비SCIE		
	계		
국내논문	SCIE		
	비SCIE		
	계		
특허출원	국내		
	국외		
	계		
특허등록	국내	0.3	
	국외		
	계	1.5	
인력양성	학사		
	석사		
	박사		
	계		
사업화	상품출시		
	기술이전		
	공정개발		
제품개발	시제품개발		
비임상시험 실시			
임상시험 실시 (IND 승인)	의약품	1상	
		2상	
		3상	
	의료기기		
진료지침개발			
신의료기술개발			
성과홍보			
포상 및 수상실적			
정성적 성과 주요 내용			

### < 별첨 자료 >

중앙행정기관 요구사항	별첨 자료
1.	1) 자체평가의견서
	2) 연구성과 활용계획서

# 자체평가의견서

## 1. 과제현황

		과제번호		317015-6	
사업구분	첨단생산기술개발사업				
연구분야				과제구분	단위
사업명	첨단생산기술개발사업				주관
총괄과제	기재하지 않음			총괄책임자	기재하지 않음
과제명	농축산 ICT 기자재 표준 기술 개발			과제유형	(기초,응용,개발)
연구개발기간	2017.04.21 - 2022.12.31 (5년 9개월)			연구책임자	
연구기간 연구개발비 (천원)	연차	기간	정부	민간	계
	1차년도	17.04-17.12	180,000		180,000
	2차년도	18.01-18.12	300,000		300,000
	3차년도	19.01-19.12	300,000		300,000
	4차년도	20.01-20.12	300,000		300,000
	5차년도	21.01-21.12	300,000		300,000
	6차년도	22.01-22.12	300,000		300,000
	계		1,680,000		1,680,000
참여기업					
상대국	상대국연구개발기관				

※ 총 연구기간이 5차년도 이상인 경우 셀을 추가하여 작성 요망

2. 평가일 : 2022.12.22.

3. 평가자(연구책임자) :

소속	직위	성명
한국전자통신연구원	책임	박주영

4. 평가자(연구책임자) 확인 :

본인은 평가대상 과제에 대한 연구결과에 대하여 객관적으로 기술하였으며, 공정하게 평가하였음을 확약하며, 본 자료가 전문가 및 전문기관 평가 시에 기초자료로 활용되기를 바랍니다.

확 약 

## I. 연구개발실적

※ 다음 각 평가항목에 따라 자체평가한 등급 및 실적을 간략하게 기술(200자 이내)

### 1. 연구개발결과의 우수성/창의성

■ 등급 : 우수

본 연구개발결과는 ICT 기자재가 반드시 필요로하는 스마트팜 환경 (시설원예, 축사, 노지)에서 기자재간 상호연동이 가능하게 하는 표준기술을 설계하고 관련 이해 당사자들의 의견을 수렴하여 국내단체 표준, 국가표준(KS)으로 진행한바 있으며, 또한 국내 기술을 기반으로 국제표준화 기구에서 주도적인 국제표준을 개발하였기에 우수하다고 판단됨

### 2. 연구개발결과의 파급효과

■ 등급 : 우수

국내 스마트팜 관련 기업들의 경쟁력을 강화시키고 농가 소득증대에 이바지하고자 농림축산식품부에서 수행하고 있는 국가표준확산사업에 본 연구개발결과물(KSX 3267, 3268, 3269)이 활용되고 있으며, 관련 산업체에 표준기술의 소개는 물론 이해 당사자와 협력을 통한 표준화를 진행함으로써 관련 표준 파급에 매우 큰 효과가 있었다고 판단됨  
연구 개발 성과물의 유관 산업에 대한 파급 효과에 있어, 그 파급효과는 업체마다 중복된 장치를 개발하기 위해 소요되는 S/W 및 H/W 개발비를 절감할 수 있을 뿐만 아니라 스마트팜 ICT 장치 모델별 유지비 또한 절감할 수 있다고 판단됨.

### 3. 연구개발결과에 대한 활용가능성

■ 등급 : 우수

최근 전 세계적으로 스마트팜 표준화에 대한 관심이 높아지고 있는 추세이며, 본 연구 개발을 통하여 개발한 국제표준이 널리 참조되고 있는 추세임. 향후 본 연구개발결과물을 기반으로 하는 국내/국제협력(표준화, 기술개발 등)이 가능하리라 사료됨

### 4. 연구개발 수행노력의 성실도

■ 등급 : 우수

본 연구개발의 주요 목표인 표준개발은 물론, 표준기술의 검증, 홍보, 전략 수립 등을 위하여 매우 성실하게 연구를 수행하였다고 사료됨. 특히 IoT, 클라우드, 인공지능 기반의 스마트팜 관제 플랫폼을 위한 선행 표준 기술을 개발함에 있어 프로토타입 S/W 개발과 테스트베드를 운영한바 있으며, 국내 전문지, 학술 세미나 등을 통해 표준화 및 스마트팜 표준기술 홍보에 많은 노력이 있었음

### 5. 공개발표된 연구개발성과(논문, 지적소유권, 발표회 개최 등)

■ 등급 : 보통

본 연구개발과제의 특성상 SCI 등의 학술 성과를 올릴 수는 없었음. 하지만 현장의 의견을 수렴하는 도중에 발생하는 요구사항을 만족시키기 위한 IPR 확보에 노력한 바 있음.

## II. 연구목표 달성도

세부연구목표 (연구계획서상의 목표)	비중 (%)	달성도 (%)	자체평가
농축산 ICT 부품 및 기자재 국내단체표준 개발(누적 60종 이상) 및 스마트팜 데이터 전송/제어 국내단체표준 개발 (3종 이상)	35%	100%	우수
농축산 ICT 기자재 국내단체표준 개정 (5건 이상)	5%	60 %	다소 미흡
국가표준(KSC)개발 (8종 이상)	20%	100%	우수
스마트 온실/축산/유통 관련 국제표준 개발 (3종 이상)	30%	100%	우수
PoC (Proof of Concept) 기능 구현을 통한 표준 검증	10%	100%	우수
합계	100점		

## III. 종합의견

### 1. 연구개발결과에 대한 종합의견

본 연구개발은 아직까지 초기 단계에 머물고 있던 국내 스마트팜 관련 표준화의 마중물 역할을 충실히 한 매우 의미있는 연구 개발이었다고 판단됨.

### 2. 평가시 고려할 사항 또는 요구사항

본 연구개발과제의 경우 표준화를 주요 목적으로 하기 때문에, 일반적으로 타 기술개발연구나 학술연구 성과물을 평가시 고려되는 학술적 성과, 제품 개발, 매출이나 사업화 척도가 아닌, 국내/국제 표준화 성과물로 평가를 요함

### 3. 연구결과의 활용방안 및 향후조치에 대한 의견

본 연구결과물은 국내 스마트팜 관련 산업체들에게 지속적으로 활용될 예정이며, 추후 개정등의 요구사항이 발생할 경우 지속적인 관리(개정, 폐지 등)가 이루어질 필요가 있다고 판단됨.

## IV. 보안성 검토

○ 연구책임자의 보안성 검토의견, 연구개발기관 자체의 보안성 검토결과를 기재함

※ 보안성이 필요하다고 판단되는 경우 작성함.

1. 연구책임자의 의견

해당 사항 없음

2. 연구개발기관 자체의 검토결과

해당 사항 없음

# 연구성과 활용계획서

## 1. 연구과제 개요

사업추진형태	<input type="checkbox"/> 자유응모과제 <input checked="" type="checkbox"/> 지정공모과제	분 야	스마트팜	
연구과제명	농축산 ICT 기자재 표준 기술 개발			
주관연구개발기관	한국전자통신연구원		주관연구책임자	박주영
연구개발비	정부지원 연구개발비	기관부담연구개발비	기타	총연구개발비
	1,680,000,000			1,680,000,000
연구개발기간	2017.04.21. - 2022.12.31 (5년 9개월)			
주요활용유형	<input type="checkbox"/> 산업체이전 <input type="checkbox"/> 교육 및 지도 <input type="checkbox"/> 정책자료 <input checked="" type="checkbox"/> 기타( 표준화 ) <input type="checkbox"/> 미활용 (사유: )			

## 2. 연구목표 대비 결과

당초목표	당초연구목표 대비 연구결과
① 농축산 ICT 부품 및 기자재 국내단체표준 개발 (누적 60종 이상) 및 스마트팜 데이터 전송/제어 국내단체표준 개발 (3종 이상)	스마트팜 관제 프로토콜 등 국내 단체표준 제안(92종), 국내 단체표준 제정 (92종)
② 농축산 ICT 기자재 국내단체표준 개정 (5건ㄹ)	영상장치, 스마트팜 구조 등 국내 단체표준 개정 (3건)
③ 국가표준(KSC)개발 (8종 이상)	RS485기반의 스마트팜 관제 프로토콜, 스마트팜 센서 및 구동기 메타데이터, 농업용 기계류 통신 프로토콜 관련 국가 표준개발 (25종)
④ 스마트 온실/축산/유통 관련 국제표준 개발 (3종 이상)	ITU-T 및 IEC를 통한 국제표준 (IS) 3건 개발 및 관련 국제 표준 기고서 38건 채택
⑤ PoC (Proof of Concept) 기능 구현을 통한 표준 검증	스마트팜 표준 기술을 검증하기 위한 IoT 기반의 스마트팜 관제 프로토콜, 인공지능 생육 감지 프로그램 등 11건 등록

\* 결과에 대한 의견 첨부 가능

### 3. 연구목표 대비 성과

(단위 : 건수, 백만원, 명)

성과 목표	사업화지표									연구기반지표										
	지식 재산권			기술 실시 (이전) 사업화						S W 구현	학술성과				교육 지도	인력 양성	정책 활용·홍 보		기타 (타 연구 활용 등)	
				특 허 출원	특 허 등록	품 종 등록	단 체 표 준 제 안	단 체 표 준 채 택	단 체 표 준 개 정		국 가 표 준 제 정	국 제 기 고 서	국 제 표 준 채 택	논문			논 문 평 균 IF	학 술 발 표		정 책 활 용
	SC I	비 SC I																		
단위	건	건	건	건	건	건	종	종	종	건	건	건		건		명	건	건		
가중치	10	10		10	10	10	20	10	20											
최종목표	10	6		60	60	5	8	20	3	3		7						2	4	
6 차 연 도	목 표	2	2			10		2	4	2	1		1						1	
	실 적	2	1			22	1	2	8	1	1		4			5			2	5
달 성 률 (%)		100	50			100	100	100	100	50	100		100			100			100	100

### 4. 핵심기술

구분	핵심기술명
①	스마트팜 관제 참조모델 국제 표준
②	IoT 기반 스마트 축산 국제 표준
③	스마트팜 유즈케이스 국제 표준
④	RS485 기반 스마트팜 관제 프로토콜

5. 연구결과별 기술적 수준

구분	핵심기술 수준					기술의 활용유형(복수표기 가능)					
	세계 최초	국내 최초	외국기술 복	외국기술 제	외국기술 소화·흡수	외국기술 개선·개량	특허 출원	산업체이전 (상품화)	현장애로 결	정책 자료	기타
①의 기술	v	v							v	v	
②의 기술	v	v							v	v	
③의 기술	v	v							v	v	
④의 기술		v							v	v	

\* 각 해당란에 v 표시

6. 각 연구결과별 구체적 활용계획

핵심기술명	핵심기술별 연구결과활용계획 및 기대효과
①의 기술	본 기술을 기반으로 스마트팜 관제 데이터모델, 프로토콜 등의 후속 표준 개발
②의 기술	본 기술을 기반으로 스마트축산 관련 서비스 구조, 데이터모델, 프로토콜 등의 후속 표준 개발
③의 기술	ITU-T FG-AI4A 활동과 연계한 스마트팜 관련 국제표준화 신규 항목 도출 및 후속 표준 개발
④의 기술	국가표준확산사업과 연계한 국내 스마트팜 ICT 기자재 상호연동 프로토콜로 활용

7. 연구종료 후 성과창출 계획

(단위 : 건수, 백만원, 명)

성과 목표	사업화지표									연구기반지표									
	지식 재산권			기술 실시 (이전) 사업화						S W 구 현	학술성과				교육 지 도	인 력 양 성	정책 활용홍 보		기 타 (타 연 구 활 용 등)
											논문	학 술 발 표	정 책 활 용	홍 보 전 시					
	특 허 출 원	특 허 등 록	품 종 등 록	단 체 표 준 제 안	단 체 표 준 채 택	단 체 표 준 개 정	국 가 표 준 제 정	국 제 기 고 서	국 제 표 준 채 택	SC I	비 SC I	논 문 평 균 IF	학 술 발 표	교 육 지 도	인 력 양 성	정 책 활 용	홍 보 전 시	기 타 (타 연 구 활 용 등)	
단위	건	건	건	건	건	건	종	종	종	건	건	건	건	명	건	건			
가중치	10	10		10	10	10	20	10	20										
최종목표	10	6		60	60	5	8	20	3	3		7				2	4		
연구기간 내 달성실적	16	5		92	92	3	25	38	3	11		16		33		1	14	38	
연구종료 후 성과창출 계획		1				2													

8. 연구결과의 기술이전조건(산업체이전 및 상품화연구결과에 한함)

해당 없음



#### 주 의

1. 이 보고서는 농림축산식품부에서 시행한 첨단생산기술개발사업의 연구보고서입니다.
2. 이 보고서 내용을 발표하는 때에는 반드시 농림축산식품부에서 시행한 첨단생산기술개발사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 됩니다.