

(옆면)

(앞면)

320018-  
2

보안 과제( ), 일반 과제( O ) / 공개( O ), 비공개( )발간등록번호( O )  
첨단생산기술개발사업 2021년도 최종보고서

발간등록번호

11-1543000-003998-01

ICT 기반  
가축분뇨  
발생량  
모니터링  
및 신속  
수거·처리  
시스템  
개발

최  
종  
보  
고  
서

2021

농림축산식품부  
농림식품기술기획평가원

# ICT 기반 가축분뇨 발생 량 모니터링 및 신속 수거· 처리시스템 개발

2022.03.21

주관연구기관 / 전북대학교산학협력단  
협동연구기관 / 동문이엔티(주)  
협동연구기관 / 강원대학교산학협력단  
협동연구기관 / 한국농업기술진흥원

농림축산식품부  
(전문기관)농림식품기술기획평가원

제출문

## 제 출 문

농림축산식품부 장관 귀하

본 보고서를 "ICT 기반 가축분뇨 발생량 모니터링 및 신속 수거·처리시스템 개발"(개발기간 : 2020 . 04. ~ 2021 . 12.)과제의 최종보고서로 제출합니다.

2022.03.21

주관연구기관명 : 전북대학교산학협력단 (대표자) 조희환

협동연구기관명 : 동문이엔티(주) (대표자) 송요일

협동연구기관명 : 강원대학교산학협력단 (대표자) 장철성

협동연구기관명 : 한국농업기술진흥원 (대표자) 안호근

주관연구책임자 : 조용진

협동연구책임자 : 김권보

협동연구책임자 : 나창식

협동연구책임자 : 한철우

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

최종보고서					보안등급		
					일반[√], 보안[ ]		
중앙행정기관명	농림축산식품부			사업명	첨단생산기술개발사업		
전문기관명 (해당 시 작성)	농림식품기술기획평가원			내역사업명 (해당 시 작성)	ICT 융복합		
광고번호	농축2020-80호			총괄연구개발 식별번호 (해당 시 작성)			
				연구개발과제번호	320018-2		
기술분류	국가과학기술 표준분류	LB0805 농업생산 시설/환경	40 %	ED0701 계측센서/부품	30 %	EE0902 USN기술	30 %
	농림식품과학기술분류	RC 농림식품기계	20 %	RC02 축산업기계	30 %	RC0202 축산업시설-환경	50 %
총괄연구개발명 (해당 시 작성)	국문						
연구개발과제명	국문 ICT 기반 가축분뇨 발생량 모니터링 및 신속 수거-처리시스템 개발						
	영문 Development of the ICT-Based Monitoring and Rapid Collecting Disposal System for Livestock Manure Management						
주관연구개발기관	기관명	전북대학교 산학협력단		사업자등록번호	402-82-15262		
	주소	(우 54896)전북 전주시 덕진구 백제대로 567		법인등록번호	210171-0005625		
연구책임자	성명	조용진		직위	조교수		
	연락처	직장전화	063-270-2615	휴대전화	010-3596-5080		
전자우편		choyj@jbnu.ac.kr		국가연구자번호	1089-5975		
연구개발기간	전체	2020. 04. 29. - 2021. 12. 31. ( 1년 9개월)					
	단계	1단계	2020. 04. 29. - 2021. 12. 31. ( 1년 9개월)				
연구개발비 (단위: 천원)	정부지원 연구개발비	기관부담 연구개발비	그 외 기관 통의 지원금 지방자치단체 기타( )				연구개발비 외 지원금
	현금	현금	현물	현금	현물	현금	현물
총계	747,000	-	249,200	-	-	747,000	249,200
1단계	1년차	320,000	-	106,800	-	320,000	106,800
	2년차	427,000	-	142,400	-	427,000	142,400
공동연구개발기관 등 (해당 시 작성)	기관명	책임자	직위	휴대전화	전자우편	비고	
공동연구개발기관	동문이엔티(주)	김권보	이사	010-3381-4056	gweonho@hanmail.net	공동	중소기업
	강원대학교 산학협력단	나창식	교수	010-8445-1651	changsa@kangwon.ac.kr	공동	대학
	한국농업기술진흥원	한철우	연구원	010-9667-4200	hanchew7@afrc.or.kr	공동	기타
연구개발기관 외 기관	익산시청	김석우	행정관리팀장	010-2659-1704	ks001@korea.kr	지자체	지자체
연구개발담당자 실무담당자	성명	이봉기		직위	박사후연구원		
연락처	직장전화	031-290-7887		휴대전화	010-9785-4601		
	전자우편	dkways@skku.edu		국가연구자번호	1094-1751		

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

2022년 03월 21일

연구책임자: 조용진

주관연구개발기관의 장: 전북대학교 산학협력단장 (직인)  
 공동연구개발기관의 장: 동문이엔티(주) (직인)  
 공동연구개발기관의 장: 강원대학교 산학협력단장 (직인)  
 공동연구개발기관의 장: 한국농업기술진흥원 (직인)

농림축산식품부장관·농림식품기술기획평가원장 귀하

## < 요약 문 >

※ 요약문은 5쪽 이내로 작성합니다.

사업명		첨단생산기술개발사업		총괄연구개발 식별번호 (해당 시 작성)			
내역사업명 (해당 시 작성)		ICT 융복합		연구개발과제번호		320018-2	
기술분류	국가과학기술 표준분류	LB0805 농업생산 시설/환경	40 %	ED0701 계측센서/부품	30 %	EE0902 USN기술	30%
	농림식품 과학기술분류	RC 농림식품기계	20 %	RC02 축산업기계	30 %	RC0202 축산업시설·환경	50%
총괄연구개발명 (해당 시 작성)							
연구개발과제명		ICT 기반 가축분뇨 발생량 모니터링 및 신속 수거·처리시스템 개발					
전체 연구개발기간		2020. 04. 29. - 2021. 12. 31. ( 1년 9개월)					
총 연구개발비		총 996,200 천원 (정부지원연구개발비: 747,000 천원, 기관부담연구개발비 : 249,200 천원, 지방자치단체: - 천원, 그 외 지원금: -천원)					
연구개발단계		기초[ ] 응용[ ] 개발[√] 기타(위 3가지에 해당되지 않는 경우)[ ]		기술성숙도 (해당 시 기재)		착수시점 기준( ) 종료시점 목표( )	
연구개발과제 유형 (해당 시 작성)							
연구개발과제 특성 (해당 시 작성)							
연구개발 목표 및 내용	최종 목표		ICT 기반 가축분뇨 발생량 모니터링 및 신속 수거·처리시스템 개발				
	전체 내용		<ul style="list-style-type: none"> <li>● 고농도 오염물질을 함유한 가축분뇨는 농장주의 경험을 기반으로 한 분뇨 수거주기 선정으로 인해 슬러리피트에 일정기간 이상 체류하면서 NH<sub>3</sub> 및 H<sub>2</sub>S 같은 다양한 유해가스를 발생시키며, 가축의 생산성 하락과 민원을 유발하여 축산업의 존속을 위협하는 요인으로 부각되고 있다.</li> <li>● 따라서 가축분뇨로 인한 비점오염원을 근본적으로 차단하고, 적정시기에 가축분뇨가 수거되지 않아 발생하는 민원문제, 토지계로의 유실 문제 등을 해결할 수 있도록 가축분뇨의 발생·수거·처리·관리·예측 전주기 플랫폼을 구축하여 체계적으로 관리하는 기술개발이 시급한 상황이다.</li> <li>● 본 연구개발은 ICT/IoT를 이용하여 가축분뇨의 발생량을 실시간 모니터링 및 데이터화하고, 지자체와 연계하여 가축분뇨 처리 현황을 실시간 분석 및 제공함으로써 가축분뇨 처리/수거업체가 최적의 수거 및 처리(전자 인수인계 시스템)를 할 수 있도록 하고자 한다.</li> <li>● 또한, 가축분뇨 발생·수거·처리 과정에서의 빅데이터를 분석하여, 가축분뇨로 인한 환경노출과 환경 영향을 보다 정확히 예측하고 처리시설 및 관리 대책을 효율적으로 수립하기 위해 정책 자료로 제공하고자 한다.</li> </ul>				
1단계		목표		<ul style="list-style-type: none"> <li>○ 축사별 가축분뇨 처리와 저장에 대한 실태 및 현황분석</li> <li>○ ICT/IoT 이용 저장조 내 분뇨의 적정 수거시점 제시</li> <li>○ ICT/IoT 기반 가축분뇨 발생량 모니터링 시스템 개발</li> <li>○ 가축분뇨 발생량 모니터링시스템을 평가하기 위한 검정방법 및 검정시스템 구축</li> <li>○ 가축분뇨 발생량 모니터링 네트워크 시스템</li> <li>○ 빅데이터 기술 적용 데이터베이스 체계 기반 가축 분뇨 신속</li> </ul>			

		<p>수거 처리 시스템 개발</p> <p>○ ICT/IoT 기반 통합 관리 시스템 개발</p> <p>&lt;전북대학교산학협력단&gt;</p> <p>○ 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축 및 영상 감시시스템 개발</p> <ul style="list-style-type: none"> <li>- 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축</li> <li>- 영상정보를 활용한 가축분뇨 발생량 모니터링 기술 개발</li> </ul> <p>○ 빅데이터 기술적용 가축분뇨 신속 수거 처리 시스템 개발</p> <ul style="list-style-type: none"> <li>- 빅데이터 기술 적용 가축분뇨 발생 예측모델 개발</li> <li>- 상정보를 활용한 가축분뇨 발생량 모니터링 위/변조 방지 기술 개발</li> </ul> <p>&lt;동문이엔티&gt;</p> <p>○ ICT/IoT 기반 가축분뇨 발생량 모니터링 시스템 설계 및 제작</p> <ul style="list-style-type: none"> <li>- 가축분뇨 발생량 모니터링 센서 선정 및 controller 개발</li> <li>- 가축분뇨 발생량 측정 정보 전송 펌웨어 개발</li> <li>- 가축분뇨 저장조 측정인자 실시간 모니터링 시스템 개발</li> </ul> <p>○ ICT/IoT 기반 가축분뇨 발생량 모니터링 시스템 개발 및 현장 적용 시험</p> <ul style="list-style-type: none"> <li>- 현장용 단말장치 시스템 개발</li> <li>- 가축분뇨 발생량 모니터링 센서 통합 시스템 현장 적용 시험</li> <li>- 가축분뇨 발생량 측정 정보 전송 시스템 현장 적용 시험</li> </ul> <p>&lt;강원대학교산학협력단&gt;</p> <p>○ 양돈분뇨 관리 현황 조사</p> <ul style="list-style-type: none"> <li>- 국내 돈사형태별 돈분의 저장·처리 현황조사</li> <li>- 익산시 축산업 데이터 기반 테스트베드 환경 분석</li> </ul> <p>○ 돈분 저장조 내 분뇨 발생량 &amp; 환경관리 측정인자 개발</p> <ul style="list-style-type: none"> <li>- 돈분 저장조 내 분뇨의 적정 수거주기를 판단할 수 있는 모니터링 인자 설정</li> </ul> <p>○ 저장조 모니터링 인자별 적정 분뇨수거시점 인지 및 센싱 방법 제시</p> <ul style="list-style-type: none"> <li>- ICT 기반 저장조 내 돈분 물성 및 저장량 변화, 악취물질 발생량 모니터링</li> <li>- 분뇨 저장조의 적정 수거 제어인자 선정 및 수거 시점 분석</li> </ul> <p>○ 분뇨 발생량 현장 실증 시험 및 분석</p> <ul style="list-style-type: none"> <li>- 테스트베드 모니터링 시스템의 보완 및 실증연구 데이터 분석</li> </ul> <p>&lt;한국농업기술진흥원&gt;</p> <p>○ 가축분뇨 발생량 모니터링시스템을 평가하기 위한 검정 방법 분류 체계 구축</p> <ul style="list-style-type: none"> <li>- 국내외 개발 중이거나 상용화가 완료된 가축분뇨 모니터링 관련 기자재의 분류체계 구축</li> </ul> <p>○ 가축분뇨 모니터링 관련 기자재 검정시스템 구축</p> <ul style="list-style-type: none"> <li>- 가축분뇨 발생량 모니터링 관련 기자재 검정시스템 구축 및 평가</li> </ul>
--	--	--

<지식재산권>	<기술실사>	<사업화>	<학술성과>
- 출원 : 3건	- 기술이전 : 1건 - 기술료 : 5백만원	- 제품화 : 1건 - 고용창출 : 2명	- SCI : 2건 - 비SCI : 2건 - 학술발표 : 8건
<인력양성>	<정책활용·홍보>	<기타>	
- 인력양성 : 4명	- 정책활용 : 1건 - 홍보 : 1건	- 1건	

연구개발성과

- <전북대학교산학협력단>
- 서버사이드 S/W를 통해 Database 구축
  - GPS를 이용한 이동정보의 관리를 위한 Web API 개발
  - 발생량 예측, 환경영향 분석 등을 위한 빅데이터 처리 Web API 개발
- <동문이엔티>
- 가축분뇨 저장조의 온도, 습도, 수위, pH, ORP 측정장치 개발
  - ICT/IoT 기반 측정장치를 통해 수집된 자료의 서버 전송을 위한 통합통신장치 개발

<강원대학교산학협력단>  
 ○ 양돈분뇨 관리 현황 조사  
 ○ 센서 이용 양돈분뇨 저장조 내 분뇨의 발생량과 물성 변화, 악취 발생경향 파악 및 적정 분뇨 수거 시기 제시  
 <한국농업기술진흥원>  
 ○ 가축분뇨 모니터링 센서 구조 및 성능시험방법 개발  
 - 가축분뇨 모니터링 센서의 주요 측정치로는 수위(비접촉식), EC, pH, ORP 등 외에도 여러 가지가 존재하는데 가장 중요하다고 사료되는 3가지에 대한 성능시험방법 및 시스템 등을 구축하였음

연구개발성과 활용계획 및 기대 효과

가. 기대성과  
 ○ 시제품 장치는 고도화 하여 국외에 수출  
 ○ 가축분뇨 발생·수거·처리·관리·예측 전주기 플랫폼을 통한 체계적인 모니터링  
 ○ ICT/IoT 장치를 이용하여 가축분뇨의 발생량을 실시간 측정  
 ○ 전국의 분뇨처리 사항을 실시간 분석  
 ○ 스마트팜 확산 사업의 지원기준 활용에 따른 국가표준기반 제품들의 확산, 시장 규모 확대, 업체의 전문성 및 경쟁력 강화, 제품 단가 하락을 통하여 농가의 경제성 향상  
 ○ ICT기자재검정을 통한 제품의 신뢰성 제고, 품질 향상에 따른 스마트축사 활용도 제고 및 도입 농가 확산  
 나. 파급효과

구분	내용	비고
기술적측면	<ul style="list-style-type: none"> <li>가축분뇨 발생·수거·처리·관리·예측 전주기 플랫폼 개발</li> <li>가축분뇨 발생 예측 모델링 알고리즘화</li> <li>영상취득 및 분석을 통한 분뇨발생량 위변조 방지</li> </ul>	
산업적측면	<ul style="list-style-type: none"> <li>전주기 플랫폼 활성화로 신산업 창출 : 177조원 규모</li> <li>가축분뇨 관련 IT회사 신규시장 창출 기대</li> <li>가축분뇨 관련 제조,설비,설비 등의 신규시장 창출 기대</li> </ul>	177조원 규모
사회적측면	<ul style="list-style-type: none"> <li>가축분뇨 수거 지체로 인한 손실비용 절감</li> <li>가축분뇨의 신속한 수거관리로 악취관련 민원 최소화</li> <li>체계적인 가축분뇨 관리로 비점오염원의 20%를 개선</li> </ul>	
국가행정측면	<ul style="list-style-type: none"> <li>환경오염 최소화 및 손실비용 절감으로 국가경쟁력 강화</li> <li>전국가축분뇨 발생·수거·처리·관리·예측 전주기 모니터링</li> <li>가축분뇨 수집·운반과정의 분뇨 이동관련 자료 신뢰성 확보</li> <li>가축분뇨로 인한 환경노출과 환경영향 예측</li> <li>가축분뇨 관리에 따른 정책입안 활용</li> </ul>	

연구개발성과의 비공개여부 및 사유

해당사항 없음

연구개발성과의 등록·기탁 건수	논문	특허	보고서 원문	연구 시설·장비	기술 요약 정보	소프트 웨어	표준	생명자원		화합물	신품종	
								생명 정보	생물 자원		정보	실물
4	3	1	-	-	-	-	-	-	-	-	-	-

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)	축산 스마트 팜	가축분뇨	사물인터넷	가축분뇨관리 플랫폼	가축분뇨 수거/처리 시스템
영문핵심어 (5개 이내)	Livestock smart farm	Livestock manure	Internet of things	Livestock manure management platform	Livestock manure collection/processing system

## < 목 차 >

1. 연구개발과제의 개요
2. 연구개발과제의 수행 과정 및 수행내용
3. 연구개발과제의 수행 결과 및 목표 달성 정도
4. 목표 미달 시 원인분석(해당 시 작성)
5. 연구개발성과 및 관련 분야에 대한 기여 정도
6. 연구개발성과의 관리 및 활용 계획

# 1. 연구개발과제의 개요

## 1-1. 연구개발 배경

○ 축산분뇨의 발생량을 사육규모별로 추산하여 살펴보면 2010년 기준 허가대상 규모 22,083 천톤/년, 신고대상 규모 14,239 천톤/년, 신고대상 규모 5,239천톤/년으로 나타난다. 젓소와 돼지의 경우 허가 대상 규모에서 발생하는 축산분뇨의 비중이 크며, 3,535 천톤/년 (50.6%)과 14,482 천톤/년 (69.5%)이며, 신고대상 규모에서 발생하는 축산분뇨가 2,680 천톤/년 (38.3%)과 5,199 천톤/년 (25.0%)으로 나타난다. 한우의 경우허가와 신고대상 규모에서 각각 4,065 천톤/년 (29.6%), 6,361 천톤/년 (46.3%)로 나타나고 있어 다른 축종과 비교하여 신고대상 규모에서의 발생량이 큰 것으로 나타난다.

표 7. 규모별 축산분뇨 발생량 추산 축산분뇨처리통계(환경부, 2010, 2011)

구분	마리수(두), (비율, %)				축산분뇨 발생량(천톤/년), (비율, %)			
	양계	젓소	한우	양돈	양계	젓소	한우	양돈
허가대상	8,849,978 (61.3)	256,916 (50.6)	813,004 (29.6)	7,780,058 (69.5)	22,083 (53.1)	3,535 (50.6)	4,065 (29.6)	14,482 (69.5)
신고대상	4,259,500 (29.5)	194,746 (38.3)	1,272,021 (46.3)	2,792,733 (25.0)	14,239 (34.3)	2,680 (38.3)	6,361 (46.3)	5,199 (25.0)
신고미만	1,335,063 (9.2)	56,435 (11.1)	663,397 (24.1)	615,231 (5.5)	5,239 (12.6)	777 (11.1)	3,317 (24.1)	1,145 (5.5)
합계	14,444,541 (100.0)	508,097 (100.0)	2,748,422 (100.0)	11,188,022 (100.0)	41,562 (100.0)	6,992 (100.0)	13,743 (100.0)	20,827 (100.0)

○ 2019년 발표된 통계청의 자료에 따르면 모든 축종에 대하여 사육농가는 축산농가의 대규모 기업화 등에 따라 전체 농가 수는 감소하고 있으나 가축분뇨 발생량은 11년도 구제역의 영향으로 감소된 사례를 제외하고 꾸준히 증가하고 있음.

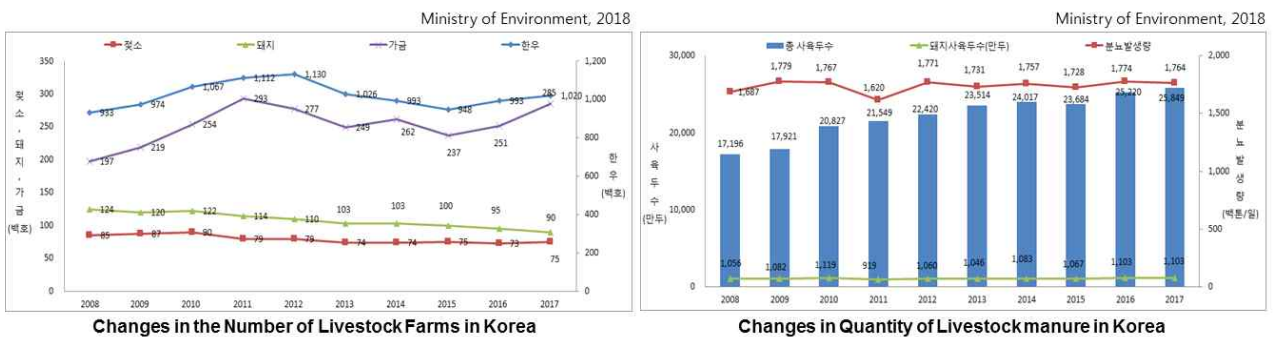


그림 3. 축산농가수 변화(좌측)와 가축분뇨발생량 변화(우측)

○ 가축분뇨의 관리 및 이용에 관한 법률에 의하여 2012년 이후 축산 농가의 가축분뇨의 방류가 금지된 이후, 제도권 아래의 체계적인 관리로 가축분뇨의 무분별한 처리는 감소하고 있지만, 가축분뇨는 여전히 국내의 하천 및 근해의 녹조 발생 원인 및 악취 등 환경 문제의 원인이 되



고 있다.

- 2012년 환경부 축산분뇨 공공처리시설 운영관리카드의 자료를 기초로 집계한 결과 가축분뇨 공공처리시설의 시설용량은 14,015톤/일로 증가하였으며, 공공처리 슬러지 발생량은 184,894 톤/년으로 증가한 것으로 나타났다.
- 공공처리 슬러지는 공공처리 시설의 유입유기물 부하, 주처리 공정의 특성, 슬러지 분리 장치의 효율, 약품사용량 등 관리적인 상황 변화에 의해 그 발생량이 큰 차이를 나타내고 있다.
- 공공처리시설에서 발생하는 전처리 슬러지의 발생량은 유기물의 농도부하가 높은 경우 증가하는 경향이 있으며 주처리 공정에서의 잉여 슬러지 발생량은 호기적 처리와 혐기적 처리간에 차이를 보이며 이론적으로 호기적 처리의 경우 주처리 생물학적 반응조의 유입 유기물의 약 50%가 미생물 균체로 전환되고, 혐기성 처리의 경우 유입 유기물의 약 7~10%가 미생물 균체로 전환되는 특성이 있어 호기성 처리 시설에서의 슬러지 발생량이 많다. 또한 공정 측면에서는 생물학적 유기물의 분해·처리 기술에 의존하기 보다는 응집·침전에 의한 물리화학적 처리 공정에 의존도가 높은 공정일수록 슬러지의 발생량이 증가한다.
- 발생 슬러지는 대부분 퇴비화, 매립, 소각의 방법으로 처리되고 있으며, 사업특성상 지자체의 단체장이 운영책임을 지고 있는 시설로서 슬러지의 처리는 지역별 특성에 부합하도록 처리하고 있다. 특히 퇴비화의 경우 농경지로의 투입은 불가능한 상황으로 조경용으로 사용하는 부숙토, 도로 절개지 등의 복원용 유기물로 사용하는 녹생토 등으로 활용되고 있다.
- 축산분뇨 공공처리 슬러지 발생량은 하수슬러지 약 386 만톤/년 수준과 비교하면 매우 미미한 양으로 발생하고 있으나 공공처리 슬러지는 하수슬러지와 마찬가지로 폐기물로 분류·관리되고 있어 농경지 환원이 제한되고, 처리에 많은 비용이 요구되고 있는 상황으로 감량화에 대한 다양한 방법이 시도 되고 있다.

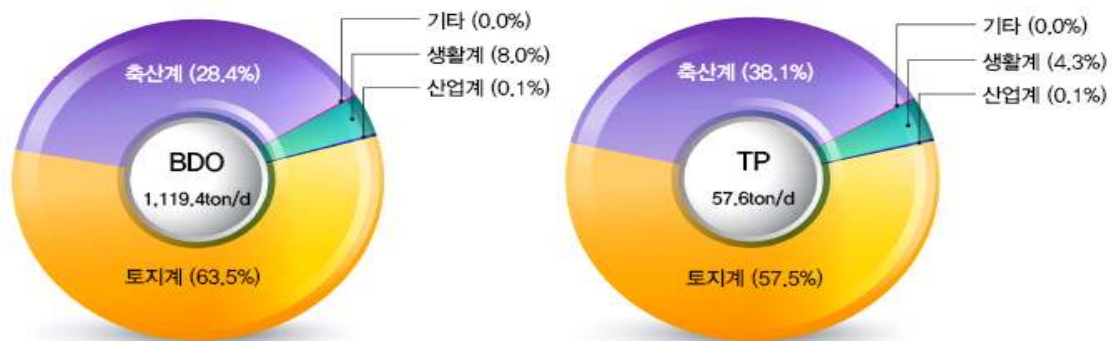
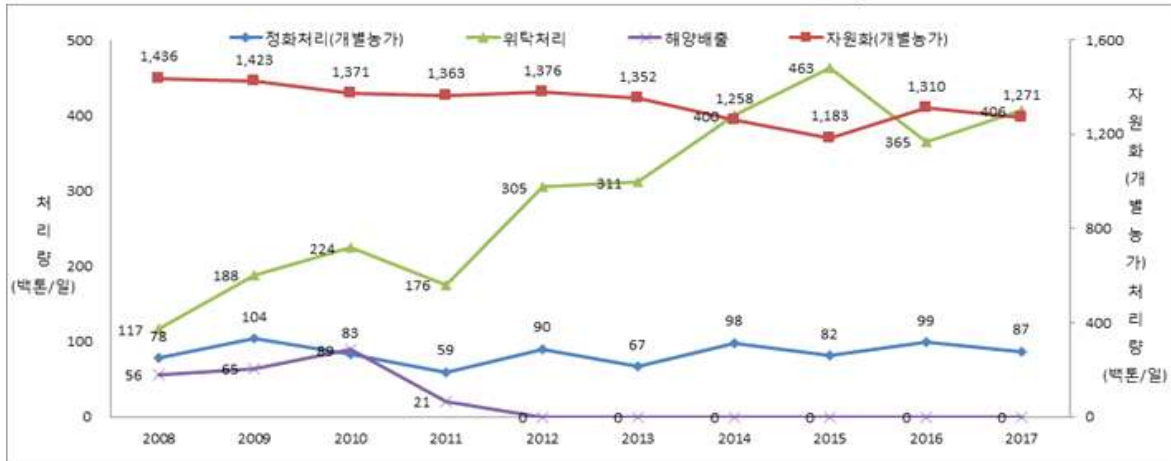


그림 4. 수질오염의 원인인 비점오염원의 BOD 및 TP 기여율

- 축산계는 상수원 및 하수원의 오염원인 BOD와 TP를 각각 28.4%, 38.1%를 차지하고 있어, 가축분뇨 수집·운반·처리 과정에 투명성만을 확보하더라도 국내 비점오염원을 크게 줄일 수 있다. 또한, 제때 수거되지 않는 가축 분뇨로 인하여 토지계로 유실(비점오염원 상승) 및 민원발생이 증가하고 있는 추세이다. 녹조의 발생 원인이 되는 비점오염원은 상수원 및 하수원에 밀접한 농경지에 살포된 가축분뇨 퇴비와 액비의 수계 유출량과 축사 및 퇴비사, 방치 가축분뇨 등에 의한 가축분뇨의 부적정 관리에 의한 강우 유출로 발생한다.



### Throughput of Livestock Manure by Type of Treatment in Korea

그림 5. 가축분뇨의 처리 현황 변화

- 가축분뇨의 처리는 자원화 및 정화처리는 감소하고 있으며 가축분뇨의 위탁처리는 증가 추세를 보이고 있다. 자원화 및 위탁처리의 경우 가축분뇨의 퇴비 및 액비화가 대부분을 차지한다. 통계청의 자료는 위탁처리 실적자료를 제출하지 않은 농가는 일괄적으로 자원화에 포함하고 있어, 국내 축산업의 고질적 문제인 무허가 축산농가의 가축분뇨 발생량 추적 불가 문제와 함께 실제 통계상 잡히지 않는 가축분뇨에 대한 사후 처리 관리 미흡을 유발하고 있다. 가축분뇨의 처리에 대한 관리 체계의 허점은 2018년에 발생한 대청호의 녹조문제와 같은 환경 문제를 야기한다. 축산계에서 발생하는 비점오염원을 예방하고 악취 및 수질오염 등 환경 문제를 개선하기 위해서는 가축분뇨의 체계적인 모니터링과 관리가 필요하다.



그림 6. 가축분뇨의 사후 처리 및 관리 미흡으로 인한 녹조 발생의 예(2018년, 대청호)

- 현재 가축분뇨의 관리를 위한 정부 부처에는 환경부, 농림축산식품부, 해양수산부, 농촌진흥청 등이 있으며, 농림축산식품부는 가축분뇨 자원화 및 이용에 관한 업무, 환경부는 축산농가 인허가 및 지도, 단속에 관한 업무를 담당하고 있으며, 농촌진흥청은 비료관리 및 가축분뇨 처리기술 개발 및 보급에 관한 업무를 수행하고 있다.
- 이러한 관리 주체의 다양화로 인해 가축분뇨 및 퇴·액비는 관리주체별로 지도단속, 시설평가, 시설지원 등 개별 목적을 가지고 운용되고 있어 종합적인 관리가 이루어지지 않고 있다. 이러한 관리체계의 미흡은 축산업을 환경오염의 주범으로 평가받도록 하고 있고 축산업을 위축시키는 가장 큰 요인의 하나가 되고 있다.

- 가축분뇨는 발생 단계에 따라 관리하여야 되는 요소와 관리주체에 차이가 발생한다. 발생단계의 주요 관리요소는 관리주체에 따라 시·군 지자체, 농가로 구분할 수 있다. 지자체는 가축분뇨 발생량을 근거로 이에 대한 적절한 시설을 갖추었는지, 적정하게 관리하고 환경오염의 문제가 없는지 등 법적 규제 및 민원중심의 관리가 대부분이다. 그 외 발생단계의 관리요소는 농가가 관리주체가 되는 요소들로서 이후 처리 및 순환단계를 어떻게 가져갈 것인가를 정하는 매우 중요한 요인이 되고 있다. 따라서 가축분뇨의 발생 단계와 단계에 따른 관리주체의 체계적이고 유기적인 가축분뇨 관리가 필요하다.

표 8. 가축분뇨 발생단계 일반적 관리요소

항목	내용	관리주체
발생 축종	어떠한 축종에서 발생한 가축분뇨인가?	농가
설치 기준	축사 및 처리시설의 기준에 적합하며 발생하는 분뇨를 처리할 수 있는 충분한 용량인가?	지자체
환경오염	가축분뇨의 누수가 없고 토양이나 하천에 흘러 가지 않는가?	지자체, 농가, 환경부
발생성분	어떠한 축사(번식, 비육 등)에서 발생한 분뇨인가?	농가
발생방식	축사의 형태(스크레퍼, 슬러리 등)는 무엇인가?	농가
고액분리	분과 뇨가 분리되는가?	농가
소독약	소독약 사용이 많은가	농가
항생제 사용	항생제 사용이 많은가	농가
저장 기간	단순 저장하여 혐기상태로 보관된 기간은?	농가
집수조 용량	집수조의 용량이 크고 가축분뇨의 성상은 균질한가?	농가
부하량/농도	세척수 사용량에 따른 가축분뇨의 농도가 얼마인가?	농가
청소/청결	얼마나 자주 물청소를 실시하는가?	농가
환경개선제	사료, 음수 드잉 미생물 재제 등 환경개선제를 사용하는가?	농가
악취	농장 내 악취의 수준과 발생하는 가축분뇨의 악취 수준은?	농가, 지자체
수거	공동자원화시설 등으로 적기에 수거가 되는가?	농가, 지자체

- 가축분뇨에 의하여 발생하는 악취 및 수질 오염 등의 환경 문제는 사회 공동의 문제인데, 가축 분뇨의 발생단계에서의 관리는 축산농가에 의존하여 진행되고 있다. 이러한 가축분뇨의 관리는 축산농가에 부담을 줄 뿐 아니라 도덕적 해이에 의한 가축분뇨의 부적정한 관리를 야기하여 가축분뇨에 의한 사회적, 환경적 문제를 악화시킨다. 가축 분뇨를 수거하여 관리하는 가축 분뇨 처리시설은 혐오시설로 인식되어 추가 건설 및 증설이 지역주민의 반대로 난항이 있어 가축 분뇨 처리 용량의 한계 문제로 분뇨의 수거 및 관리의 어려움이 있다.

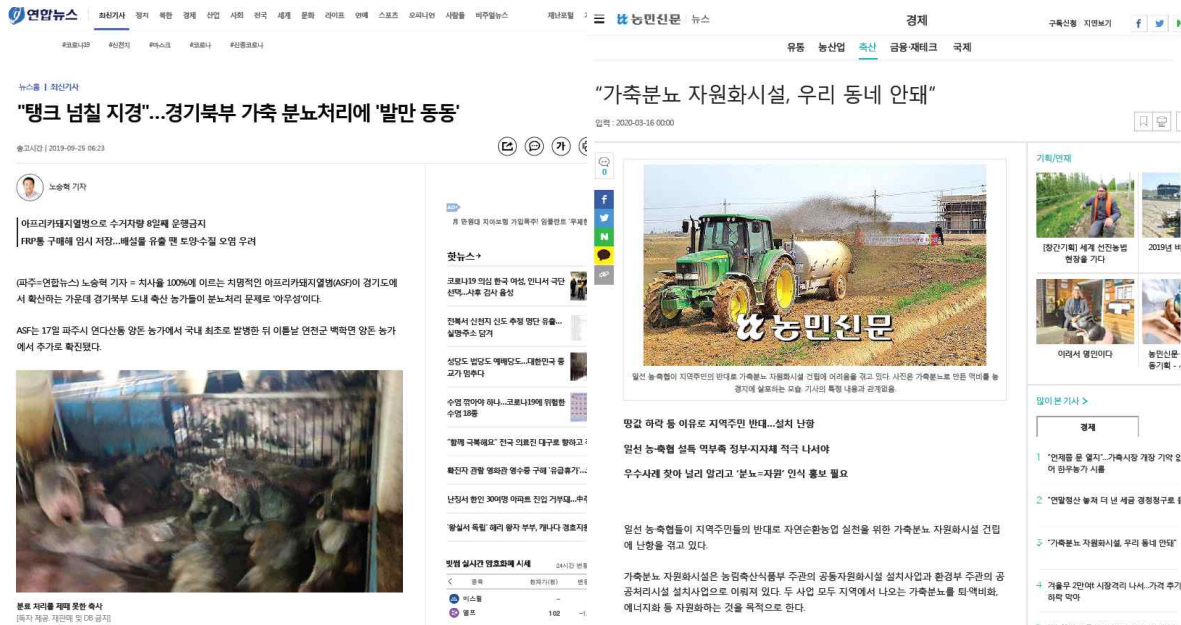


그림 7. 가축분뇨의 적기 처리 실패와 가축분뇨 처리시설 증설로 인해 발생한 사회 문제 관련 기사

- 따라서 신규 가축분뇨 처리시설 건설이 제한적인 상황에서 가축분뇨의 제도권 하의 체계적인 관리를 위해서는 가축분뇨 발생 단계 별로 시스템화하여 관리하여야 한다. 축사의 일상 관리라고 할 수 있는 청소, 소독제, 항생제, 세척수 등은 축산 농가의 자체 관리에 해당하는 문제이지만, 가축분뇨의 저장조 관리 및 분뇨 수거 등은 축산 농가 단독으로 해결할 수 없는 문제라고 볼 수 있다.
- 2019년 하반기 돼지 열병 문제로 인하여 돼지 축산 농가의 이동이 통제된 상황에서 축사에서 발생하여 저장조에 저장되는 가축분뇨의 배출이 원활하게 이루어지지 않아 사회적 문제로 대두된 사례가 있다. 가축분뇨의 수거 후 공동자원화 시설 등의 처리 시설에서의 관리는 한국환경공단에서 ‘가축분뇨전자인계관리시스템’을 수립하여 수거차량의 이동 및 처리시설에서의 최종처리를 관리하지만, 개별 축산농가의 가축분뇨의 저장조 관리에 대한 부분은 사용자인 농장주에 의존하여 저장조 관리 및 모니터링이 누락되어 있다. ‘가축분뇨의 관리 및 이용에 관한 법률’에 의하면 발생한 가축분뇨의 수거 및 최종 처리는 배출시설과 관련하여 환경부 소관의 업무이고 매일 축사에서 발생하는 가축분뇨는 퇴비 및 액비와 관련하여 농림축산식품부 소관의 업무이기 때문에 개별 목적에 의한 이원화된 관리가 이루어지기 때문이다.

- 따라서 환경부 산하의 ‘가축분뇨전자인계관리시스템’에서 누락된 개별 축산농가의 가축분뇨 발생 및 저장조 관리를 위해서는 이러한 개별 축산농가의 가축분뇨 발생 및 저장조 관리를 위한 체계적이고 자동화된 시스템 및 플랫폼이 필요하다.
- 가축분뇨의 수거 이후 최종 처리까지 진행하기 이전의 가축분뇨의 발생 및 저장조 관리에 대하여 축산농가에 의존하여 진행되었던 문제를 해결하고 체계적이고 효율적인 가축분뇨의 관리를 위하여 ICT 기반의 가축분뇨 발생 및 저장조 모니터링 등 가축분뇨 관리기술이 필요하다. 가축분뇨의 부적정 처리를 방지하고 가축분뇨의 제도권 아래에서의 관리를 유도하기 위해서는 가축분뇨의 발생 및 저장 단계에서의 체계적인 모니터링과 감시 미 관리가 요구된다.
- 이를 위해서는 축사에서 가축분뇨의 발생 및 개별 축산농가의 저장조에 대한 환경 및 발생량 관리를 위한 인자를 도출하고, 각 인자를 모니터링 할 수 있는 기술과 모니터링된 관리 인자의 정보를 분석, 통합하는 기술을 개발하여, 가축분뇨 발생 및 처리 주체인 축산농가, 수거업체, 지자체에 대하여 그 정보와 대처방법을 제공함과 동시에 지역 및 국가 단위의 관리망에 정보를 전파하는 시스템의 구축이 필요하다.

## 1-2. 연구개발 개요

가축분뇨로 인한 비점오염원을 근본적으로 차단하고, 적정시기에 가축분뇨가 수거되지 않아 발생하는 민원문제, 토지계로의 유실 문제 등을 해결할 수 있도록 가축분뇨의 발생·수거·처리·예측의 전주기 플랫폼을 구축하여 체계적으로 관리하고자 함이다.

본 연구개발은 ICT/IoT를 이용하여 가축분뇨의 발생량을 실시간 모니터링 및 데이터화하고, 지자체와 연계하여 가축분뇨 처리 현황을 실시간 분석 및 제공함으로써 가축분뇨 처리/수거업체가 최적의 수거 및 처리(전자 인수인계 시스템)를 할 수 있도록 하고자 한다.

또한, 가축분뇨 발생·수거·처리 과정에서의 빅데이터를 분석하여, 가축분뇨로 인한 환경노출과 환경 영향을 보다 정확히 예측하고 처리시설 및 관리 대책을 효율적으로 수립하기 위해 정책 자료로 제공하고자 한다.

따라서, 가축분뇨 이동의 투명성 확보와 농장의 가축분뇨의 최적 수거 및 처리를 통하여 환경오염 최소화 및 그로 인한 손실비용을 절감하여 국내 축산업의 환경문제를 개선하여 세계화된 시장에서의 축산업의 국가경쟁력을 강화하고자 한다.

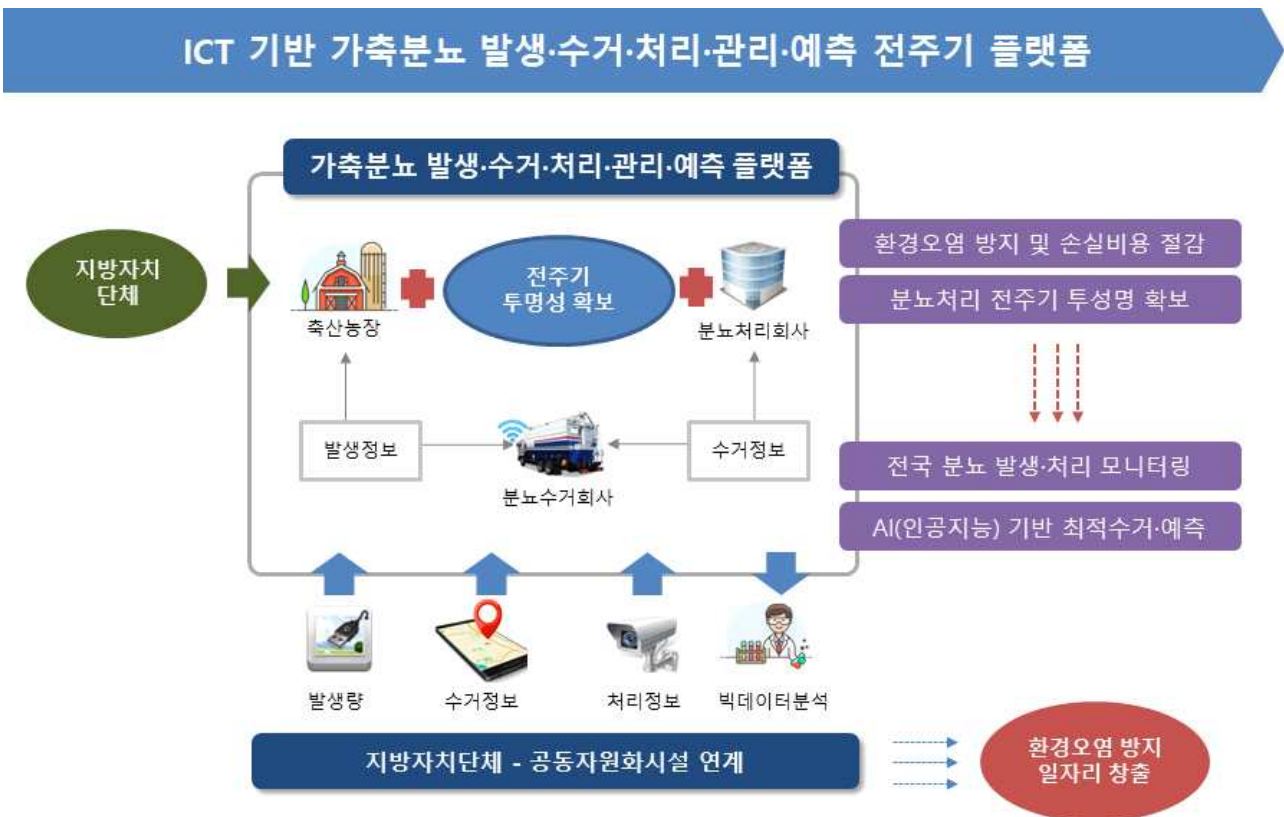


그림 8. ICT 기반 가축분뇨 발생 수거 처리 관리 예측 전주기 플랫폼

### 1-3. 연구개발 내용

- ICT/IoT기반 측정 장치를 통해 가축분뇨 저장조의 온도, 수위, pH, ORP를 실시간 측정함.
- 통합 네트워크 시스템을 통하여 서버로 측정된 Data를 전송하여 실시간 빅데이터를 구축함.
- 토지계로의 유출을 방지하기 위해 수거 적정시기에 대한 정보를 제공하는 한편, 수거차량의 이동정보를 관리함.
- 분뇨처리 회사의 처리 현황 등을 실시간 모니터링 하여 수거 및 처리에 대한 투명성 확보함.
- Open API를 통하여 전국 가축분뇨처리 현황 및 통계 등을 지자체에 제공함.
- 빅데이터 분석을 통하여 환경영향 예측 및 관리대책의 효율성에 대한 정책 자료로 제공함.

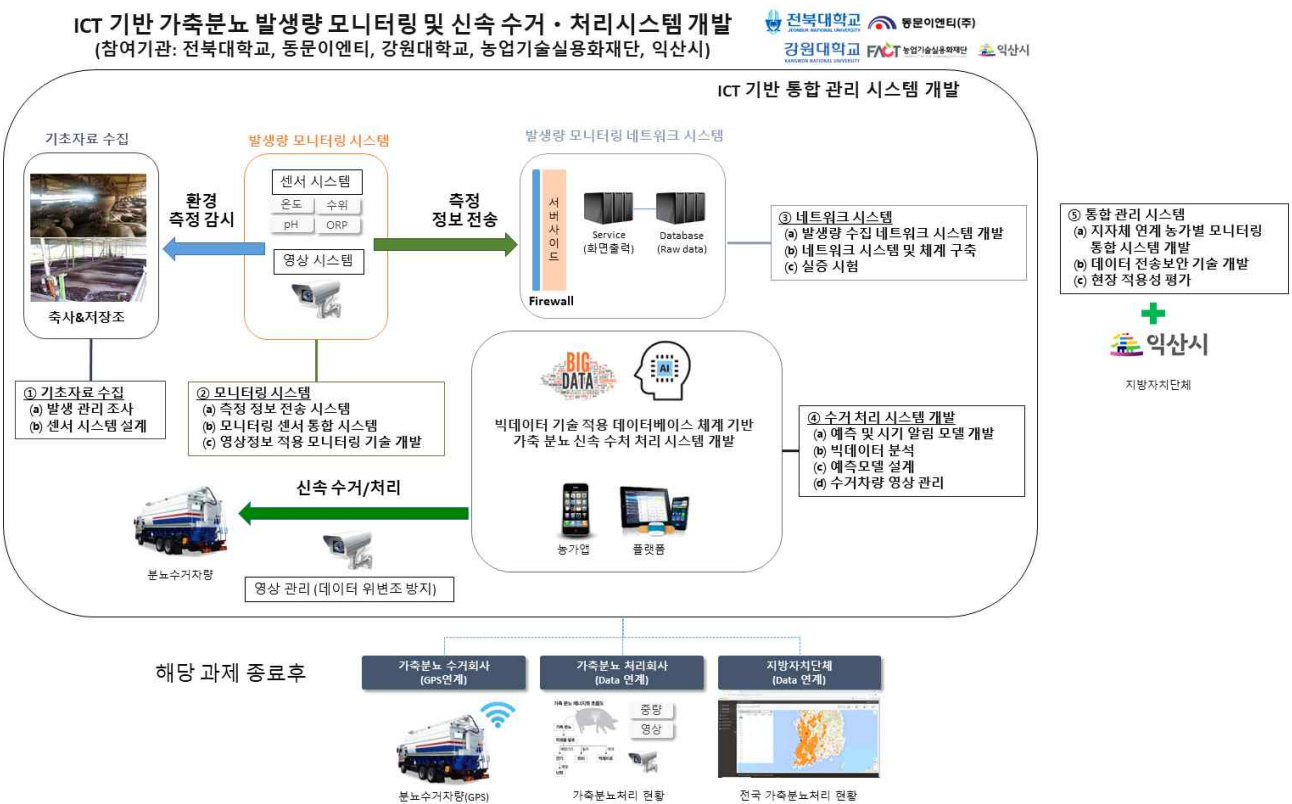


그림 9. ICT 기반 가축분뇨 발생량 모니터링 및 신속 수거 처리시스템 개발 내용

구분	연구개발 대상	유형
ICT/IoT기반 측정장치	● 가축분뇨 저장조의 온도, 습도, 수위, pH, ORP 측정장치	시제품 제작
통합통신장치	● ICT/IoT 기반 측정장치를 통해 수집된 자료를 서버로 전송	시제품 제작
Server	● 서버사이드 S/W를 통해 Database 구축	SW개발
수거차량	● 최적 수거 서비스 시스템 ● 수거차량 GPS를 이용하여 이동정보의 관리	SW개발
처리회사	● API를 협의하여 서버사이드S/W 연동	SW개발
지방자치단체	● 지역별 처리현황 및 통계정보를 Open API로 제공	SW개발
빅데이터	● 발생량 예측, 환경영향 분석 등	SW개발

가. 최종목표

- ICT 기반 가축분뇨 발생량 모니터링 및 신속 수거·처리시스템 개발

나. 세부목표

- 축사별 가축분뇨 처리와 저장에 대한 실태 및 현황분석
- ICT/IoT 이용 저장조 내 분뇨의 적정 수거시점 제시
- ICT/IoT 기반 가축분뇨 발생량 모니터링 시스템 개발
- 가축분뇨 발생량 모니터링시스템을 평가하기 위한 검정방법 및 검정시스템 구축
- 가축분뇨 발생량 모니터링 네트워크 시스템
- 빅데이터 기술 적용 데이터베이스 체계 기반 가축 분뇨 신속 수거 처리 시스템 개발
- ICT/IoT 기반 통합 관리 시스템 개발

다. 연차별 개발목표 및 내용

1-1. 1차년도 (주관연구기관, 전북대학교)

- 연구개발 목표  
: 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축 및 영상 감시시스템 개발
- 개발 내용 및 범위
  - 1) 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축
    - 축산농가 가축분뇨 발생량 수집을 위한 네트워크 시스템 구축
    - 가축분뇨 발생량 정보의 가축분뇨 수거업체 연계 가능한 네트워크 시스템 및 체계 구축
  - 2) 영상정보를 활용한 가축분뇨 발생량 모니터링 기술 개발
    - 가축분뇨 저장조 내의 영상정보 취득 알고리즘 설계
    - 가축분뇨 저장조 내 발생량 관리 및 감시를 위한 영상 시스템 개발

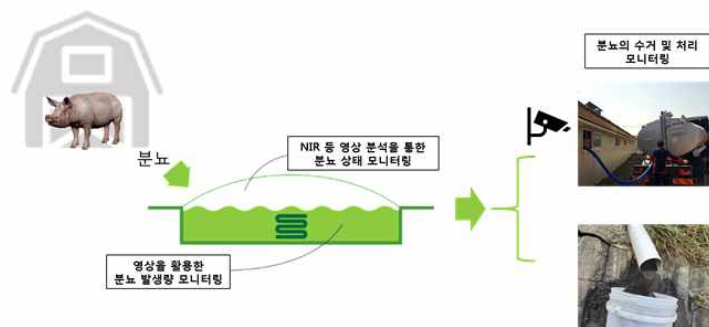


그림 10. 영상정보를 활용한 가축분뇨 모니터링 개요도



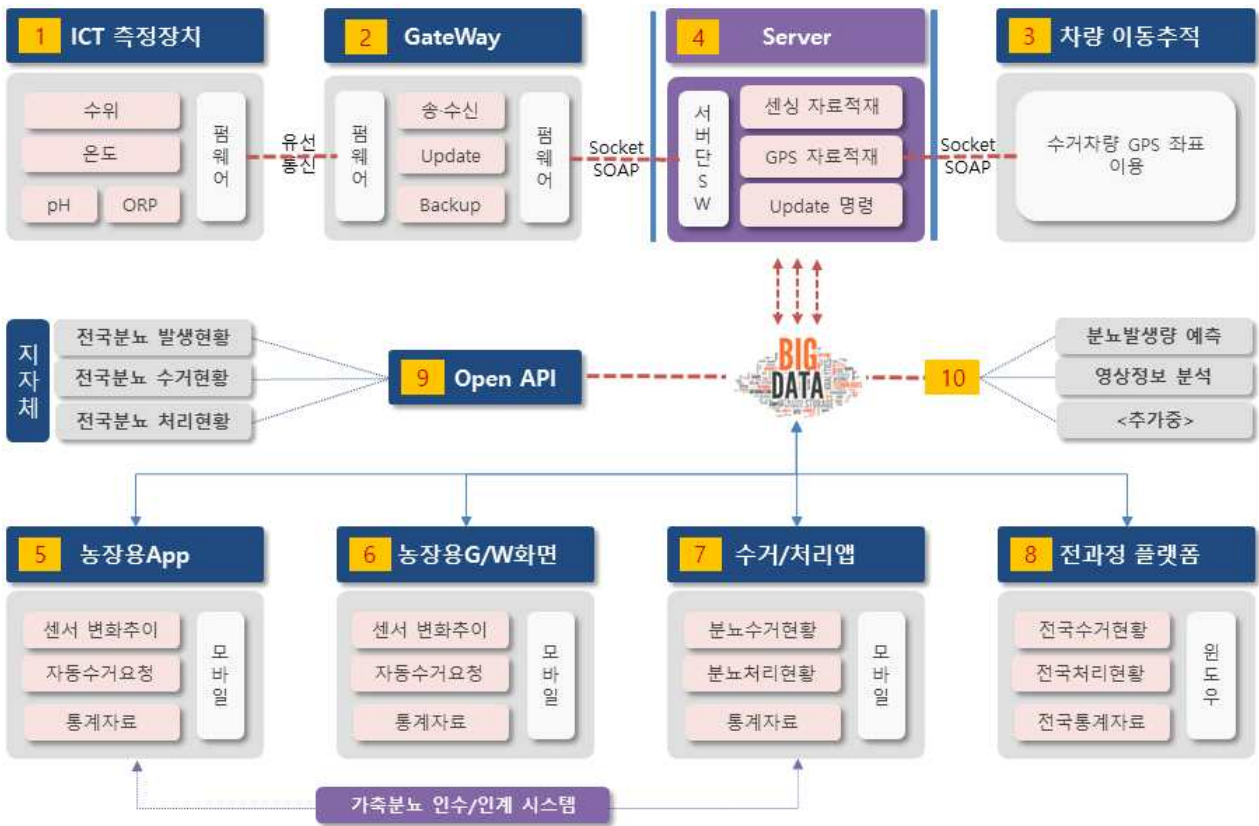


그림 11. 가축분뇨 신속 수거 처리 시스템 개요도

표 9. 가축분뇨 신속 수거 처리 시스템 개요도 내용 설명

번호	명칭	내용
1	ICT/IoT 기반 측정 장치	농장 가축분뇨 저장조내에 온도, 수위, pH, ORP 검지부를 설치하고, 계측된 자료를 통합 통신장치에 유선(RS-485)으로 전송한다.
2	통합 통신장치	ICT/IoT 기반 측정 장치로부터 계측된 Data(온도, pH, ORP, 수위)값을 Server Side SW로 전송하는 장치
3	차량이동 추적장치	수거차량 GPS 좌표를 이용
4	Server Side SW	수신한 정보의 Database 구축
5	농장용 App	농장전용 App으로 가축분뇨 저장조 내 분뇨량, 온도, pH, ORP등의 변화추이를 그래프로 출력
6	GateWay 구동 SW	농장전용으로 가축분뇨 저장조 내 분뇨량, 온도, pH, ORP등의 변화추이를 그래프로 출력
7	분뇨수거/처리 App	농장에서부터 분뇨수거 요청 등의 내용을 열람
8	발생,수거,처리 현황 플랫폼	발생,수거,처리 전과정의 현황을 모니터링
9	Open API	농장 저장조 분뇨상태 현황 및 발생, 수거, 처리 등의 Data를 열람할 수 있도록 API를 제공
10	빅데이터 영상정보 분석	시계열기반 분뇨 발생량 예측모델 개발 분뇨발생량 위변조방지를 위한 영상정보 확보 및 분석

1-2. 1차년도 (협동연구기관 1, 동문이엔티)

○ 연구개발 목표

: ICT/IoT 기반 가축분뇨 발생량 모니터링 제품 설계 및 제작

○ 개발 내용 및 범위

1) 가축분뇨 발생량 모니터링 센서 선정 및 센서 Controller 개발

- 축산농가 가축분뇨 저장조 내 환경 및 발생량 측정 센서 선정 및 센서 Controller 개발
- 가축분뇨 발생량 모니터링을 위한 센싱 정보 기반 저장조 내 측정 인자의 실시간 정밀 모니터링 시스템 개발
- 발생량 원격 모니터링을 위한 센서 선정 및 제작 설치

2) 가축분뇨 발생량 측정 정보 전송 시스템 개발

- 측정 정보 수집을 위한 유선 전송 시스템 개발
- 수집 데이터의 통합 관리를 위한 통신장치 개발



그림 12. ICT/IoT 가축분뇨 저장조 환경 및 발생량 모니터링 센서 통합시스템 개요

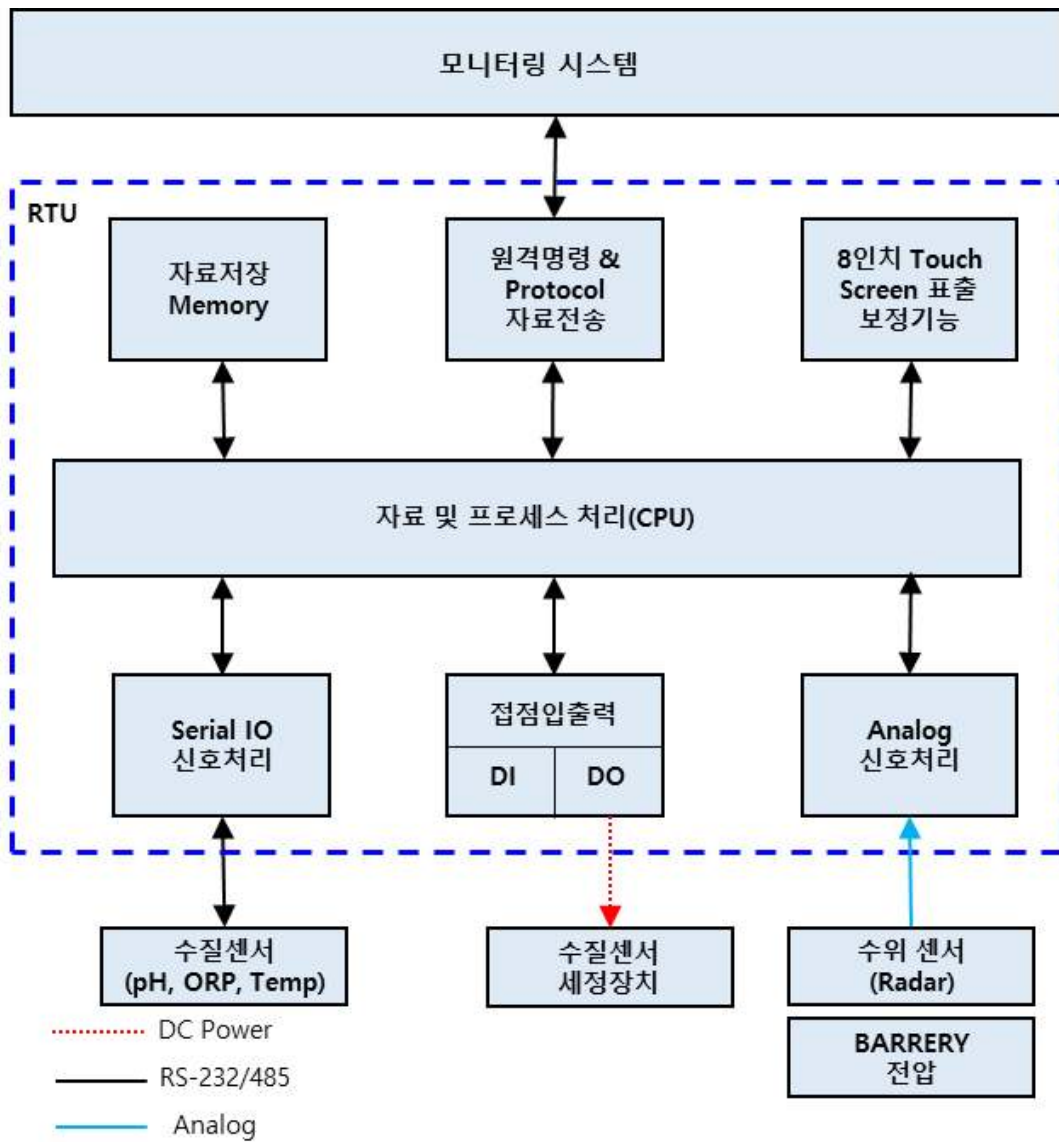


그림 13. ICT/IoT 기반 가축분뇨 저장조 환경 및 발생량 모니터링 센서 통합시스템 다이어그램

1-3. 1차년도 (협동연구기관 2, 강원대학교)

○ 연구개발 목표

: 축사형태별 돈분 처리와 저장에 대한 실태 조사 및 저장조 모니터링 인자 선정

○ 개발 내용 및 범위

1) 양돈분뇨 관리현황 조사

- 국내 축사형태별 돈분 저장조 종류, 관리 및 처리 현황조사

◎ 돈사 형태에 따른 가축분뇨 발생량 분석

◎ 돈분 발생량 대비 처리현황 및 통계정보 분석

- 익산시 축산업 데이터 기반 테스트베드 환경 분석 (현장 조사)

2) 돈사 및 저장조 형태에 따른 분뇨 발생량 관리 및 적정 처리 시점 결정을 위한 환경 및 발생량 측정 인자 개발

- 양돈분뇨 저장조 내 분뇨의 적정 수거주기를 판단할 수 있는 모니터링 인자 설정

◎ 선행연구 및 문헌조사 수행

- ICT 기반 테스트베드 농가 내 저장조 모니터링

◎ 제1협동기관(동문이엔티)와 협력하여 현장 내 모니터링 센서 설비

◎ 분뇨 물성 및 악취 모니터링 (온도, pH, ORP, NH<sub>3</sub>, H<sub>2</sub>S)

◎ 물성변화 분석항목: COD<sub>cr</sub>, STOC, T-N, NH<sub>4</sub>-N, NO<sub>x</sub>-N, T-P, Ortho-P, SS, VSS, TS, TVS

◎ 분뇨 발생량 모니터링 (수위감지: 영상, 레벨센서, 레이저)

- 적정 처리 시점 결정을 위한 환경 및 발생량 측정 인자 선정

1-4. 1차년도 (협동연구기관 3, 한국농업기술진흥원)

○ 연구개발 목표

: 가축분뇨 발생량 모니터링시스템을 평가하기 위한 검정 방법 분류체계 구축

○ 개발 내용 및 범위

- 국내외 개발 종이거나 상용화가 완료된 가축분뇨 모니터링 관련 기자재의 분류체계 구축

- 가축분뇨 모니터링 기자재의 신뢰성 분석요소 발굴(내환경성, 내구성 등)

- 전기적·기계적 연결규격 및 통신방법 등의 확보로 호환성 확보

- 작성완료 된 분류체계를 활용하여 축산기자재 표준(안) 개발활용

- 개발·유통 중인 가축분뇨 모니터링 기자재의 성능지표 조사하여 성능시험개발방법에 활용

## 2-1. 2차년도 (주관연구기관, 전북대학교)

### ○ 연구개발 목표

: 빅데이터 기술적용 가축분뇨 신속 수거 처리 시스템 개발

### ○ 개발 내용 및 범위

#### 1) 빅데이터 기술 적용 가축분뇨 발생 예측모델 개발

- 인공지능기술을 이용한 시계열기반 가축분뇨 상태 예측 및 수거 작업 시기 알림 모델 개발
- 축사 형태 및 저장조 형태에 따른 가축분뇨 발생량 관리 및 적정 처리 시점 결정을 위한 빅데이터 분석
- 축사 형태 및 저장조 형태에 따른 효율적 가축분뇨 관리를 위한 발생량 예측모델 설계

#### 2) 영상정보를 활용한 가축분뇨 발생량 모니터링 위/변조 방지 기술 개발

- 가축분뇨 저장조 내의 영상 정보의 신뢰성 확보를 위한 센싱 정보 비교 분석
- 가축분뇨 저장조 내 발생량 위/변조 방지 및 신뢰도 확보를 위한 영상 분석 알고리즘 개발



그림 14. 가축분뇨 관리 위한 발생량 예측모델 설계 및 현황분석의 예

## 2-1. 2차년도(협동연구기관 1, 동문이엔티)

### ○ 연구개발 목표

: ICT/IoT 기반 가축분뇨 발생량 모니터링 제품 개발 및 현장 적용 시험

### ○ 개발 내용 및 범위

#### 1) 현장용 단말장치 시스템 개발

- 현장(농장) 전용으로 저장조내 분뇨량, 온도, pH, ORP등의 변화추이 그래프 출력
- 현장용지시계 ~ 단말장치간 통신 프로토콜 설계/구현
- UI(User Interface) 설계/ 구현

#### 2) 가축분뇨 발생량 모니터링 센서 통합 시스템 현장적용 시험

- 현장측정기 교정 / 확인법 : 측정기 자체점검→ 기기 교정/확인→ 결과 검토 및 기기 교정
- 표준물질 정도 관리법 : 표준물질 측정→ 비교test→ 결과 검토 및 기기 교정
- 현장측정기 확인결과 목표값 미달 시 관련 자료를 참조하여 측정결과 유효성 검토 시행

#### 3) 가축분뇨 발생량 측정 정보 전송시스템 현장적용 시험

- 축산농가 가축분뇨 저장조 내 데모 장비 설치 후 테스트베드 구축
- 기능시험 : 측정소 내 하드웨어, 소프트웨어 및 기타장비에 대한 설치 및 기본 기능시험
- 일반검사 : 축분 측정 장비의 설치 시 실시하는 장비 성능 시험을 위한 수분석 결과 와 비교 시험
- 통합시험 : 축분 측정장비 및 전송시스템을 통한 데이터 수집 및 전송시험
- 사용자 평가 : 실제 운영될 설비상에서 사용자 중 일부 인원을 선정하여 정상적으로 사용 및 운영되는가를 평가하여 운영상의 문제점을 도출 후 운영에 대비

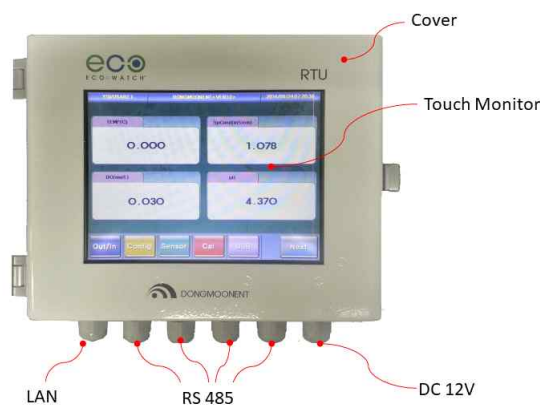


그림 15. 가축분뇨 발생량 측정 정보 전송 시스템 개요

2-3. 2차년도 (협동연구기관 2, 강원대학교)

○ 연구개발 목표

: 축사형태별 적정 돈분 처리를 위한 분뇨 수거시점 설정 및 현장실증 시험 평가

○ 개발 내용 및 범위

1) 저장조 모니터링 인자별 및 적정 분뇨수거시점 인지 및 센싱 방법 제시

- 저장조 내 분뇨 발생량 측정 센서의 적정 처리시점 분석

◎ 익산시 테스트베드 농가 기반 현장 모니터링 데이터 기반 분뇨 물성 및 가스 발생량과 분뇨 발생량의 상관관계 규명

- 축산농가 가축분뇨 저장조 내 환경 및 발생량 측정 통합센서 설계를 위한 측정 인자별 제어시점 설정

- 현장 모니터링 결과를 바탕으로 시제품 개발을 위한 근거데이터 제공

2) 분뇨 발생량 현장실증 시험 및 분석

- 제작된 시제품의 테스트베드 적용 및 데이터 분석

◎ 돈분 적정 처리시점 모니터링 시스템 적용에 따른 저장조 내 환경 개선효과 분석

◎ 슬러리 피트 분석항목: STOC, T-N, NH<sub>4</sub>-N, NO<sub>x</sub>-N, TSS, TVSS, TS, TVS

◎ 냄새물질 분석항목: NH<sub>3</sub>, H<sub>2</sub>S

- 시제품의 정확도 평가 및 보완사항 제시

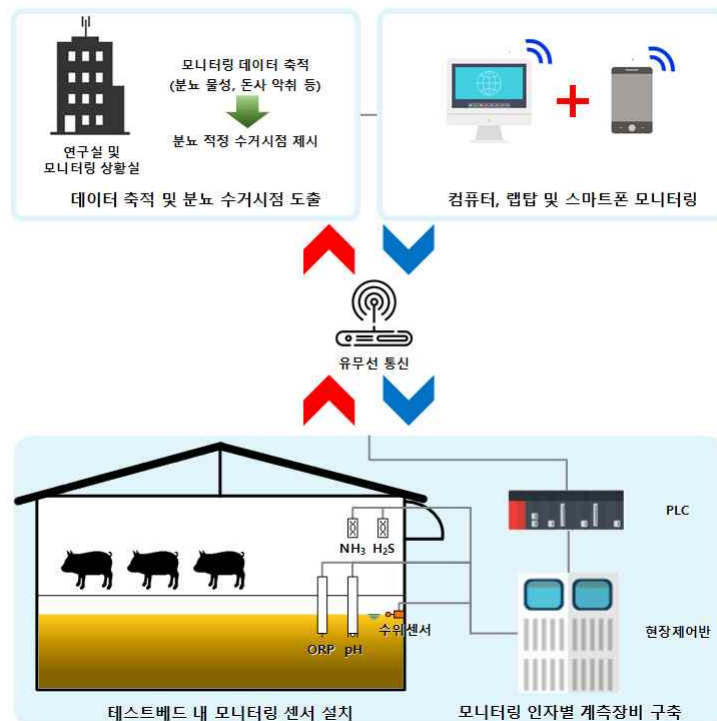


그림 16. 테스트베드 적용 가축분뇨 적정 처리시점 모니터링 시스템의 구성

2-4. 2차년도 (협동연구기관 3, 한국농업기술진흥원)

○ 연구개발 목표

: 가축분뇨 모니터링 관련 기자재 검정시스템 구축

○ 개발 내용 및 범위

- 가축분뇨 모니터링 관련 기자재의 성능 및 신뢰성 시험용 시스템 구축
- 가축분뇨 모니터링 기자재의 내환경성, 내구성 검정방법(안) 개발



그림 17. 가축분뇨 모니터링 기자재 성능시험 시스템



<주관기관: 전북대학교>

구분	연도	연구개발의 목표	연구개발의 내용
1차년도	2020	<ul style="list-style-type: none"> <li>◦ 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축 및 영상 감시시스템 개발</li> </ul>	<ul style="list-style-type: none"> <li>● 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축                             <ul style="list-style-type: none"> <li>- 축산농가 가축분뇨 발생량 수집을 위한 네트워크 시스템 구축</li> <li>- 가축분뇨 발생량 정보의 가축분뇨 수거 업체 연계 가능한 네트워크 시스템 및 체계 구축</li> </ul> </li> <li>● 영상정보를 활용한 가축분뇨 발생량 모니터링 기술 개발                             <ul style="list-style-type: none"> <li>- 가축분뇨 저장조 내의 영상정보 취득 알고리즘 설계</li> <li>- 가축분뇨 저장조 내 발생량 관리 및 감시를 위한 영상 시스템 개발</li> </ul> </li> </ul>
2차년도	2021	<ul style="list-style-type: none"> <li>◦ 빅데이터 기술적용 가축분뇨 신속 수거 처리 시스템 개발</li> </ul>	<ul style="list-style-type: none"> <li>● 빅데이터 기술 적용 가축분뇨 발생 예측 모델 개발                             <ul style="list-style-type: none"> <li>- 인공지능기술을 이용한 시계열기반 가축분뇨 상태 예측 및 수거 작업 시기 알림 모델 개발</li> <li>- 축사 형태 및 저장조 형태에 따른 가축분뇨 발생량 관리 및 적정 처리 시점 결정을 위한 빅데이터 분석</li> </ul> </li> <li>- 축사 형태 및 저장조 형태에 따른 효율적 가축분뇨 관리를 위한 발생량 예측 모델 설계</li> <li>● 영상정보를 활용한 가축분뇨 발생량 모니터링 위/변조 방지 기술 개발                             <ul style="list-style-type: none"> <li>- 가축분뇨 저장조 내의 영상 정보의 신뢰성 확보를 위한 센싱 정보 비교 분석</li> <li>- 가축분뇨 저장조 내 발생량 위/변조 방지 및 신뢰도 확보를 위한 영상 분석 알고리즘 개발</li> </ul> </li> </ul>

<협동기관 1: 동문이엔티>

구분	연도	연구개발의 목표	연구개발의 내용
1차년도	2020	<ul style="list-style-type: none"> <li>◦ ICT/IoT 기반 가축분뇨 발생량 모니터링 시스템 설계 및 제작</li> </ul>	<ul style="list-style-type: none"> <li>● 가축분뇨 발생량 모니터링 센서 선정 및 controller 개발                             <ul style="list-style-type: none"> <li>- 축산농가 가축분뇨 저장조 내 환경 및 발생량 측정 센서 선정</li> <li>- 선정 센서에 대한 controller 개발(PCB 설계/개발, PCB 제작, 케이싱 제작)</li> </ul> </li> <li>● 가축분뇨 발생량 측정 정보 전송 펌웨어 개발                             <ul style="list-style-type: none"> <li>- 센서(수위, 온도, pH, ORP) Driver 개발</li> <li>- 통신방식 선정(RS 485, TCP/IP)</li> <li>- 통신 프로토콜 설계/구현</li> <li>- UI(User Interface) 설계/ 구현</li> </ul> </li> <li>● 가축분뇨 저장조 측정인자 실시간 모니터링 시스템 개발                             <ul style="list-style-type: none"> <li>- 실시간 모니터링 data 표출, 그래프 구현 등 관제시스템 기능선정/개발</li> <li>- 자체 DB 설계/구현</li> <li>- UI(User Interface) 설계/ 구현</li> </ul> </li> </ul>
2차년도	2021	<ul style="list-style-type: none"> <li>◦ ICT/IoT 기반 가축분뇨 발생량 모니터링 시스템 개발 및 현장 적용 시험</li> </ul>	<ul style="list-style-type: none"> <li>● 현장용 단말장치 시스템 개발                             <ul style="list-style-type: none"> <li>- 현장(농장) 전용으로 저장조내 분뇨량, 온도, pH, ORP등의 변화추이 그래프 출력</li> <li>- 현장용지시계 ~ 단말장치간 통신 프로토콜 설계/구현</li> <li>- UI(User Interface) 설계/ 구현</li> </ul> </li> <li>● 가축분뇨 발생량 모니터링 센서 통합 시스템 현장 적용 시험                             <ul style="list-style-type: none"> <li>- 축산농가 가축분뇨 저장조 내 환경 및 발생량 측정 센서 시스템 현장 적용 시험</li> <li>- 가축분뇨 발생량 모니터링을 위한 센싱 정보 기반 저장조 내 측정 인자의 실시간 모니터링 시스템 현장 적용 시험</li> </ul> </li> <li>● 가축분뇨 발생량 측정 정보 전송 시스템 현장 적용 시험                             <ul style="list-style-type: none"> <li>- 측정 정보 수집을 위한 유선 전송 시스템 현장 적용 시험</li> <li>- 수집 데이터의 통합 관리를 위한 통신장치 현장 적용 시험</li> </ul> </li> </ul>

<협동기관 2: 강원대학교>

구분	연도	연구개발의 목표	연구개발의 내용
1차년도	2020	<ul style="list-style-type: none"> <li>◦ 축사형태별 돈분 처리와 저장에 대한 실태 조사 및 저장조 모니터링 인자 선정</li> </ul>	<ul style="list-style-type: none"> <li>● 양돈분뇨 관리현황 조사                             <ul style="list-style-type: none"> <li>- 국내 축사형태별 돈분 저장조 종류, 관리 및 처리 현황조사</li> <li>- 익산시 축산업 데이터 기반 테스트베드 환경 분석 (현장 조사)</li> </ul> </li> <li>● 돈사 및 저장조 형태에 따른 분뇨 발생량 관리 및 적정 처리 시점 결정을 위한 환경 및 발생량 측정 인자 개발                             <ul style="list-style-type: none"> <li>- 양돈분뇨 저장조 내 분뇨의 적정 수거주기를 판단할 수 있는 모니터링 인자 설정</li> <li>- ICT 기반 테스트베드 농가 내 저장조 모니터링</li> <li>- 적정 처리 시점 결정을 위한 환경 및 발생량 측정 인자 선정</li> </ul> </li> </ul>
2차년도	2021	<ul style="list-style-type: none"> <li>◦ 축사형태별 적정 돈분 처리를 위한 분뇨 수거시점 설정 및 현장실증 시험 평가</li> </ul>	<ul style="list-style-type: none"> <li>● 저장조 모니터링 인자별 및 적정 분뇨수거시점 인지 센싱 방법 제시                             <ul style="list-style-type: none"> <li>- 저장조 내 분뇨 발생량 측정 센서의 적정 처리시점 분석</li> <li>- 축산농가 가축분뇨 저장조 내 환경 및 발생량 측정 통합센서 설계를 위한 측정 인자별 제어시점 설정</li> <li>- 현장 모니터링 결과를 바탕으로 시제품 개발을 위한 근거데이터 제공</li> </ul> </li> <li>● 분뇨 발생량 현장실증 시험 및 분석                             <ul style="list-style-type: none"> <li>- 제작된 시제품의 테스트베드 적용 및 데이터 분석</li> <li>- 시제품의 정확도 평가 및 보완사항 제시</li> </ul> </li> </ul>

<협동기관 3: 한국농업기술진흥원>

구분	연도	연구개발의 목표	연구개발의 내용
1차년도	2020	<ul style="list-style-type: none"> <li>◦ 가축분뇨 발생량 모니터링시스템을 평가하기 위한 검정 방법 분류체계 구축</li> </ul>	<ul style="list-style-type: none"> <li>● 국내외 개발 중이거나 상용화가 완료된 가축분뇨 모니터링 관련 기자재의 분류체계 구축</li> <li>- 가축분뇨 모니터링 관련 기자재의 구조조사 방법활용</li> <li>- 전기적·기계적 연결규격 및 통신방법 등의 확보로 호환성 확보</li> <li>- 작성완료 된 분류체계를 활용하여 축산기자재 표준(안) 개발활용</li> <li>- 개발·유통 중인 가축분뇨 모니터링 기자재의 성능지표 등도 함께 조사하여 성능시험 개발방법 등에 활용</li> </ul>
2차년도	2021	<ul style="list-style-type: none"> <li>◦ 가축분뇨 모니터링 관련 기자재 검정시스템 구축</li> </ul>	<ul style="list-style-type: none"> <li>● 가축분뇨 발생량 모니터링 관련 기자재 검정시스템 구축 및 평가</li> <li>- 가축분뇨 모니터링 관련 기자재의 성능시험용 시스템 구축</li> <li>- 온도, 습도, 암모니아, 더스트챔버 등을 활용할 예정이며 1차년도 연구결과에 따라 구축</li> </ul>

## 1-4. 연구개발 대상의 국내·외 현황

### 가. 국내외 기술 수준 현황

#### ○ 가축분뇨의 환경오염 예방 및 자원화를 통한 통합적 관리기술 체계 구축

- 가축분뇨의 퇴비화 기술이 환경오염 예방과 자원순환적인 면에서 필요하나 톱밥사용량의 증가와 가격의 상승으로 톱밥절약형 퇴비화 기술개발의 필요성이 지적되었음 (축산경제연구원, 2006)
- 가축분뇨의 환경오염 예방 및 자원화를 통한 통합적 관리기술 체계 구축 처리비용과 편리성 면에서 효율적인 액비화방법이 현장에서는 운영 방법이 통일되지 못하고 있어 현장 실용화를 위한 운영 매뉴얼의 확립이 필요함 (공동자원화센터 평가, 2012)
- 가축분뇨의 자동화 모니터링 및 관리 기술이 일부 선도 농가를 통해 추진되고 있으나 지역 전반에 운영되고 있지는 못하고 있음. 우리나라 현장 실정에 맞는 모니터링 및 자동화 기술적 체계 개발이 필요함 (안성 농장, 2013)
- 가축분뇨가 경제적으로 지역에서 활성화 될 수 있도록 ICT기술과 연계하는 통합적 운송 관리 시스템이 국내 연구진에 의해 시험적으로 연구되고 있으나 현장에 접목하지는 못한 상태임 (건국대학교, 2012)

#### ○ 퇴·액비 고부가가치화 및 ICT 활용 가축분뇨 퇴·액비 통합관리체계 분야

- 축산농가는 가축사육과정에서 분과 뇨를 발생, 뇨는 정화처리 또는 액비로 처리되며 분은 퇴비화처리 됨
- 농촌사회에 축산존립을 위하여 축산에서 경종농업에 도움이 되도록 가축분뇨를 고품질 퇴비와 액비로 생산할 필요가 절실하여, 생산 및 관리 시스템 개발과 체계 확립 기초연구 추진 중임
- 가축분뇨의 적정자원화 문제에 대한 현안 해결을 위해 IT, BT를 이용한 새로운 공정개발과 관리시스템 구축 및 현장적용 지원을 위한 기초연구 추진 중임
- 2000년대 후반에 접어들면서 가축분뇨를 이용한 대표적인 연구로 농촌진흥청 주관의 “자연순환농업연구사업단”이 본격적으로 구성, 연구사업이 추진되었음
- 전체적으로 가축분뇨의 처리 및 이용에 관하여 종합적으로 추진되었으나 상호 연계적이지 못하였고 주요 연구과제별로 독립적으로 수행되어 현장에서 요구하는 통합적인 장점은 발휘하지 못했음

#### ○ 가축분뇨 예방 및 위험관리를 위한 예측/감시 체계 구축 운영

- 정밀 가축분뇨 관리 시스템은 기술, 자료, 지식, 효율적인 전문가의 지원에 의하여 문제를 해결하기 위한 컴퓨터 기반 시스템을 적용하고 있음(Power, 1997)
- 미국의 새로운 가축분뇨 관리법에는 농경지에 가축분뇨를 정밀(Precision and accuracy)하게 사용하는 법을 제정함, 이로 인하여 global positioning systems (GPS)와, 유량계(flow meters)가 적용되어 사용하고 있음
- 가축분뇨 퇴·액비의 지역순환을 확대하기 위해서는 지역단위 양분 총량제 대비 ICT 기반 통합 양분관리시스템 구축시스템에 대한 연구가 이루어지고 있음

○ 가축분뇨의 처리·이용 동향

- 가축분뇨 슬러리 중 오제스키 바이러스 생존과 온도와의 관계 연구 (Anette Botner, 1991)
- 가축분뇨 중 바이러스의 불화성화 연구 (B. Hass, R.AH 등, 1995)
- 양돈슬러리 중 병원성 동물 바이러스의 파일럿 열적처리의 효과 (Claire Tuner 등, 1999)
- 가축의 전염병 오염과 인체에 미치는 영향; Infection animal disease and human health(M. J. Otte and D.Grace, 2012)
- 호기, 혐기소화액으로부터 막분리를 이용한 영양물질의 회수 연구(M. Zhang 등, 2013)
- 가축분뇨로부터 영양물질 회수를 위한 광배양법의 방법론에 관한 연구 (D.A Burke, 2013)
- 퇴비발효열 이용 재이용퇴비의 건조기술 개발 연구(축산환경연구소, 2013)
- 일본은 돈사에서 발생하는 악취를 줄이고 부존자원을 활용하기 위하여 액상 발효사료 제조기술을 실용화하는 연구를 수행하고 있음 (Choi 등, 2012)
- 여름철 슬러리 내 활발해진 미생물에 의해 고형물 분해도가 높아지면서 슬러리 내 BOD와 VFA의 농도가 증가됨(Williams와 Evans, 1981)

○ 국내 표준화현황

- 현재, 축산기자재와 관련하여 사양관리기기 25종에 대해서는 한국농업기술진흥원에서 단체표준을 통하여 ‘스마트축산 사양관리 기기 및 정보수집’으로 정의하고 있음
- 가축의 종류에 따라 대가축(SPS-FACT 0001-7293), 중가축(SPS-FACT0002-7294), 소가축(SPS-FACT0003-7295)으로 정의하고 있으나, 가축분뇨에 대한 모니터링, 처리규정은 정의되어 있지 않음
- 가축의 종류에 따라 대가축(SPS-FACT 0001-7293), 중가축(SPS-FACT0002-7294), 소가축(SPS-FACT0003-7295)으로 정의하고 있으나, 가축분뇨에 대한 모니터링, 처리규정은 정의되어 있지 않음

○ 국내 기타현황(검정방법)

- 한국농업기술진흥원에서는 농업기계화촉진법 시행규칙 중 농업기계의 범위에 작년 6월 포함된 농축산물 생산 환경조절장치에 축산ICT기자재 내·외기 및 안전센서 19종, 사양관리기기 25종 등 총 44종을 검정하고자 검정방법(안) 등을 작성하고 전문가 협의회를 실시하였으며 2020년 중 시행하고자 추진하고 있음
- 그러나, 가축 분뇨 모니터링에 관련된 센서, 기자재 등에 대해서는 검정방법(안) 등을 작성하지 않은 상태이므로 이번 연구과제를 통하여 검정기준 및 검정방법, 검정시스템 등을 구축하고 검정업무를 수행하고자 함

나. 산업 현황

○ 퇴·액비 고부가가치화 및 ICT 활용 가축분뇨 퇴·액비 통합관리체계

- 가축분뇨 자원화 산업을 위한 국내 사업체는 비교적 소규모의 기업형태로 운영되고 있으며, 새로운 기술개발을 위한 자체 연구능력은 미흡한 상태임
- 주요 가축분뇨 사업 분야도 퇴비화, 액비화, 악취제어 등 오염관리 분야에 머물고 있음. 따라서

축종별 가축분뇨의 고품질화, 새로운 소재 개발, ICT 모바일 유통관리 시스템 등 선진적 기술 개발을 지원하는 기업이 부족함

- 가축분뇨를 이용한 고품질의 사업화를 위해서는 각 분야의 전문적 기술을 연계하는 기술적 컨소시엄 체계가 필요하며 이를 지역 단위별로 통합, 현장에 접목하는 일관성 있는 시스템의 구축이 필요함

○ 축산 ICT 융합 사양관리 단체표준 제정

- 스마트축산 ICT 융합분야 센서(19종)의 단체표준을 한국정보통신기술협회를 통해 제정함. 스마트축산 ICT 융합분야 센서는 각 축사의 내외기 및 안전 분야로 나누어 구분함 (그림 8).

구분	축산 센서 단체표준 대상 기종
안전 센서	정전 센서, 누전 센서, 아크 센서, 낙뢰보호기 등
내기 센서	기온 센서, 습도 센서, CO2 센서, 암모니아 센서, 조도 센서, O2센서, 차압 센서, 풍속 센서 등
외기 센서	기온 센서, 풍향 센서, 풍속 센서, 감우 센서, 습도 센서, 일사 센서, 일조 센서 등

그림 8. 스마트팜 축산분야 센서 단체표준 제정 대상기종

- 스마트 축산분야 센서에 대한 공동규격은 농촌진흥청에서 개발하였으며, 한국농업기술진흥원에서 한국정보통신기술협회의 스마트농업 프로젝트 그룹인 PG426을 통해 단체표준 제정 진행함.

다. 시장 현황

○ 국내 고품질 퇴·액비 플랜트 시장

- 국내 퇴·액비 생산 시장의 규모는 매년 설치되는 설비의 규모와 연관되어 있음. 2013년 기준 퇴·액비의 설치 수는 약 2천여 개소이며, 생산 되는 퇴·액비의 정부 지원금 규모는 약 900억 원, 농가 부담금의 규모는 150억 원임
- 국내 축산농가의 퇴비화 시설은 한우, 젓소, 양돈, 양계로 나누어 볼 수 있음. 전 축종의 년 간 분뇨 발생량은 약 4,600만 톤 정도이며 그 가운데 본 연구의 주요 연구 대상인 양돈 분뇨는 약 1,800만 톤 정도 임.
- 국내 축산농가의 퇴비화 시설은 한우, 젓소, 양돈, 양계로 나누어 볼 수 있음. 전 축종의 년 간 분뇨 발생량은 약 4,600만 톤 정도이며 그 가운데 본 연구의 주요 연구 대상인 양돈 분뇨는 약 1,800만 톤 정도 임.
- 양돈 분뇨는 주로 액비유통센터, 가축분뇨공동자원화센터에서 처리되고 있으며 이 시설은 2007년부터 정부 주도로 지원되어 매년 늘어나고 있는 상황임.
- 고품질 퇴·액비 플랜트 장치는 기존 단순한 퇴·액비 설비에 추가적으로 설치하여 고품질의 퇴·액비를 생산하는 장치로, 기존의 가축분뇨 공동자원화시설에 설치되었던 퇴·액비화 장치와는 별도로 소요되는 장치임. 예를 들어 1일 100톤의 가축분뇨를 공동자원화시설에서 고품질화 하기 위하여 고품질 퇴·액비 플랜트를 설치할 경우 약 25억 원의 장치비가 소요될 것으로 예상 됨

- 고품질 퇴·액비 플랜트 장치 시장의 규모: 2015년부터 가축분뇨 공동자원화센터 및 2016년부터 액비유통센터는 비료관리법에 따라 “비료생산 업 등록 의무화”로 전환되어 가축분뇨의 고품질화를 목표로 운영되고 있음
- 공동자원화센터 및 액비유통센터의 장치 설비 시장 : 센터 1개당 25억(추산 : 농림부)
- 생산되어지는 고품질액비의 시장규모
  - ◎ 액비의 리터당 가격: 본 연구 사업으로 개발되어 생산되는 고품질 액비는 양돈분뇨 1톤 당 약 100리터의 농축액비를 생산할 수 있음(10배 농축 액비). 농축 되어진 액비의 질소 농도는 약 1%(총질소 10,000ppm) 정도로 예상되어지며, 시중 유사제품의 가격과 비교하여 볼 때 최소한 리터 당 약 1,000원의 경제성은 확보 가능 함. 따라서 양돈분뇨 1톤 당 100리터 × 1,000원/리터 는 100,000원/톤 정도의 가치로 예상됨
  - ◎ 유사제품과의 가격 비교: 시중 유사 액비 제품으로는 당밀의 경우 N, P, K 1-0-5의 경우 리터 당 2,000원 이며, EM제재의 경우 리터 당 4,500 - 14,500원 정도 이므로 본 양돈분뇨 농축액비의 리터 당 1,000원은 상당히 보수적인 가격으로 산정하였음
  - ◎ 공동자원화센터에서의 액비 생산량과 경제가치: 1개소의 공동자원화센터에서의 년 간 액비처리량은 약 3만 톤 정도 임. 봄과 가을의 발효액비 살포 계절을 제외 한 여름과 겨울철의 액비는 농지에 살포 할 수 없으므로 이 기간에 발생 된 분뇨는 고품질 액비 생산 방식으로 전환함. 따라서 고품질 액비의 처리량은 년 간 처리량의 50%인 약 1.5만 톤 정도임
  - ◎ 고품질액비는 발생 분뇨 톤 당 약 100리터 정도가 생산되므로 1.5만 톤 × 100리터/톤은 1,500,000리터, 즉 1,500톤/년 임. 1,500,000리터 × 1,000원/리터 는 1,500,000,000원 (15억 원)/년. 따라서 1개소의 공동자원화센터에서 생산 할 수 있는 액비의 경제성은 년 간 약 15억 원 정도임. 이 금액은 매년 고품질 액비를 생산하는 공동자원화센터의 개소 수가 증가할 경우 액비의 시장은 기하급수적으로 증가 할 것으로 예상됨
- 상기 액비의 시장 규모 예측 자료는 축산분야 중 양돈 분야 가운데 전체 발생량의 불과 1/4 정도에서 생산되는 예를 설명한 것임. 따라서 가축분뇨 발생 및 저장조 관리 모니터링을 효율적이고 체계적으로 시행한다면 가축분뇨의 자원화를 통한 시장 형성은 추산된 시장 규모보다 더 커질 수 있을 것으로 판단함



라. 지식재산권 현황

- 가축분뇨의 모니터링과 관련하여 “가축분뇨 and 모니터링”의 검색식을 이용하여 분석한 결과 국내의 관련 특허는 총 154개의 특허로 확인됨
- 가축분뇨와 관련하여 대두되는 문제 중 악취와 수질오염과 관련한 기술 위주로 지식재산권을 확인 할 수 있었으며, 본 연구과제와 유사한 가축분뇨의 관리를 위한 기술로 “가축분뇨 통합 관리 시스템 및 그 방법(등록번호: 10-1201865)”와 “가축분뇨 전자인계 관리 시스템 및 방법 (등록번호: 10-1742908)”을 확인 할 수 있었음
- 가축분뇨 통합관리 시스템 및 그 방법(등록번호: 10-1201865)
  - 해당 특허는 가축분뇨의 종합적인 관리체계 구축을 통해 개별, 지역 및 국가단위의 축산업계 전반에 대한 분뇨처리를 관리하는 기술에 관한 것 임
  - 해당 특허는 가축분뇨를 분뇨액과 고형분으로 분리 및 침전시켜 퇴비 및 액비로 분류하여 관리 하는 분뇨처리장치와 분뇨처리장치에서 발생하는 바이오가스 및 각각의 퇴비 및 액비의 발생 량을 관리서버에서 통합관리할 수 있도록 구성하는 시스템임
  - 해당 시스템은 ‘1. 가축분뇨를 집수하기 위한 설비’, ‘2. 분과 뇨를 분리 저장하기 위한 설비’, ‘분 리된 액비와 퇴비의 가스 발생 및 발생량을 모니터링하기 위한 설비’ 등 해당 시스템을 구성하기 위한 초기 투자비용이 많은 것으로 판단되며, 기존의 축산농가의 저장조에는 적용할 수 없고 해당 시스템을 사용하기 위한 농장을 따로 구축하여야 된다는 문제점을 가지고 있음

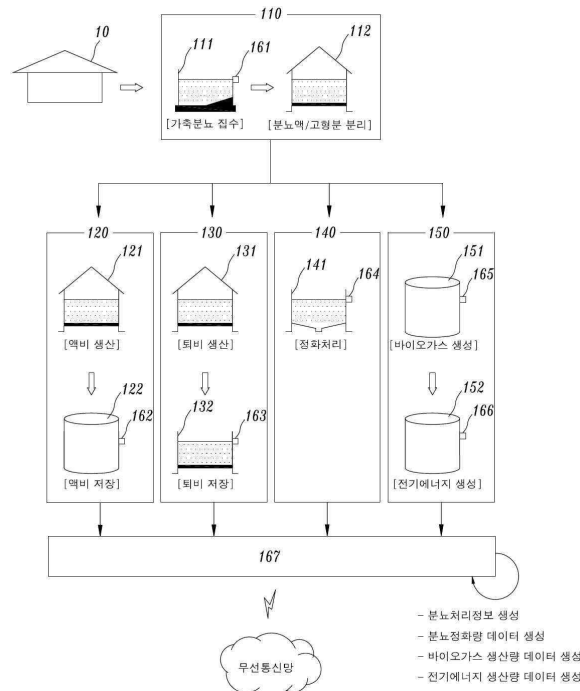


그림 9. “가축분뇨 통합관리 시스템 및 그 방법”의 개념도

- 가축분뇨 전자인계 관리 시스템 및 방법(등록번호: 10-1742908)
  - 배출자의 단말로부터 가축분뇨 배출을 위한 전자인계서를 입력받고, 전자인계서에 지정된 운반 자의 단말에 전자 인계서를 전달하며, 운반자에 의한 가축분뇨 수집운반 과정을 모니터링함. 가

축분뇨의 배출, 수집운반, 처리 및 액체비료의 운반, 살포의 전 과정을 입력하고 관리하는 시스템을 통해, 사용자들(배출자, 수집운반자, 처리자)에게는 적정처리를 유도하고, 관리자는 시스템을 통해 적정처리를 확인하고 불법처리 시 추적할 수 있도록 하는 시스템임.

- 해당 특허는 가축분뇨의 수거 및 최종 처리의 관리 및 감시에 초점이 맞추어져 저장조에 저장된 가축분뇨를 농가에서 인력으로 파악하여 수거운반자를 호출하도록 구성이 되어 있어 자동화되고 체계적인 가축분뇨의 발생 및 저장조 감시 및 관리에 적합하지 않음

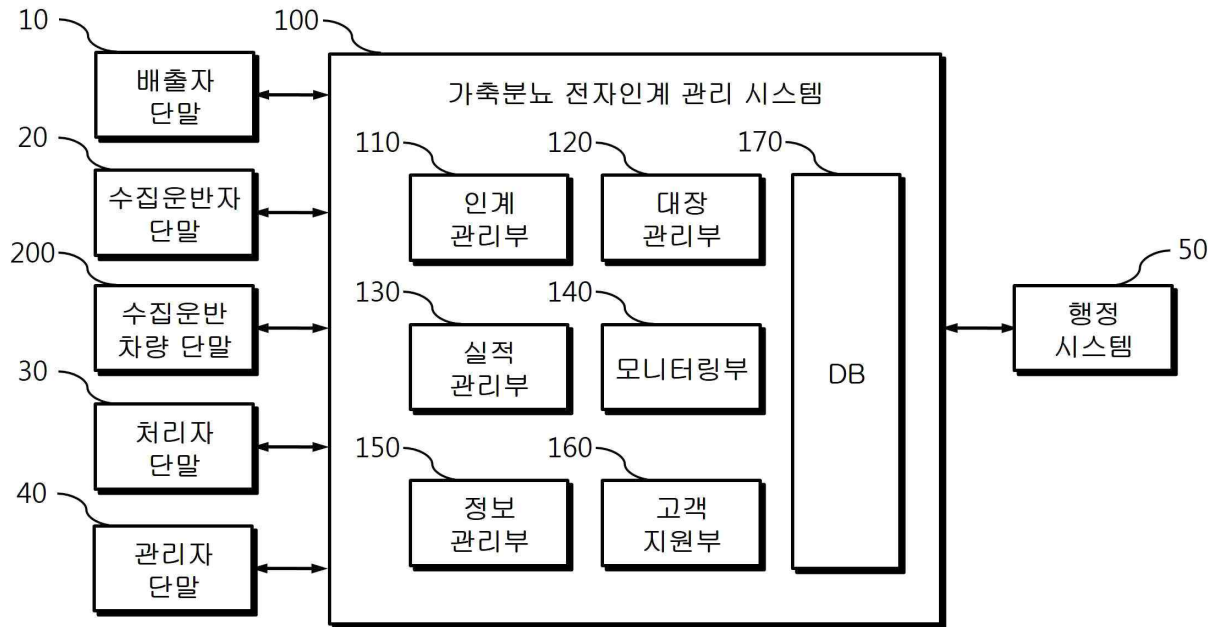


그림 10 “가축분뇨 전자인계 관리 시스템 및 방법”의 개념도

- 선행 특허는 현대화된 선진 축산 농가에 가축분뇨의 관리를 위한 전체 통합 설비를 구축하는 것(‘가축분뇨 통합관리 시스템 및 그 방법’ 등)과 가축분뇨의 수거 및 최종처리 관리(‘가축분뇨 전자인계 관리 시스템 및 방법’ 등)를 위한 것으로 구분할 수 있어 자동화되고 체계적인 가축분뇨의 발생 및 저장조 감시·관리를 위하여 기존의 축산 농가의 가축분뇨 저장조 등 관련 시설에 적용하기에는 적합하지 않아 기존의 개별 축산농가에 적용하기 위해서는 본 연구를 통하여 개발하고자 하는 시스템을 통한 ICT 기반의 가축분뇨 발생 및 저장조 모니터링 등 가축분뇨 관리기술이 필요하다고 판단함.

○ 정량적 평가 지표

(단위 : 건수, 백만원, 명)

성과 목표											연구기반지표								
	지식 재산권			기술 실시 (이전)		사업화					기술 인증	학술성과			교육 지도	인력 양성	정책 활용 홍 보		기 타 (타 연구 활용 등)
	특 허 출 원	특 허 등 록	품 종 등 록	건 수	기 술 료	제 품 화	매 출 액	수 출 액	고 용 창 출	투 자 유 치		논문		학 술 발 표			정 책 활 용	홍 보 전 시	
												SC I	비 SC I						
단위	건	건	건	건	백 만 원	건	백 만 원	백 만 원	명	백 만 원	건	건	건	건	명	건	건		
가중치	10	10		10		10	10		10				5		5	10	5	5	
최종 목표																			
1차년도	1							1				1	3						
2차년도	2	1		1		1		1			2		4		3	1	1	1	
소 계	3	1		1		1		2			2	1	7		3	1	1	1	
종료 1차년도		2					100			1	1				1				
종료 2차년도							100												
종료 3차년도							250												
종료 4차년도							500												
종료 5차년도							700												
소 계		2					1,650			1	1				1				
합 계	3	3		1		1	1,650		2	1	3	1	7		4	1	1	1	

\* 단계별 연구성과 목표는 향후 중간/최종/추적평가 등의 정량적 평가지표로 활용됨

\*\* 연구성과는 연구개발계획에 맞춰 도출하고 예시와 같이 작성

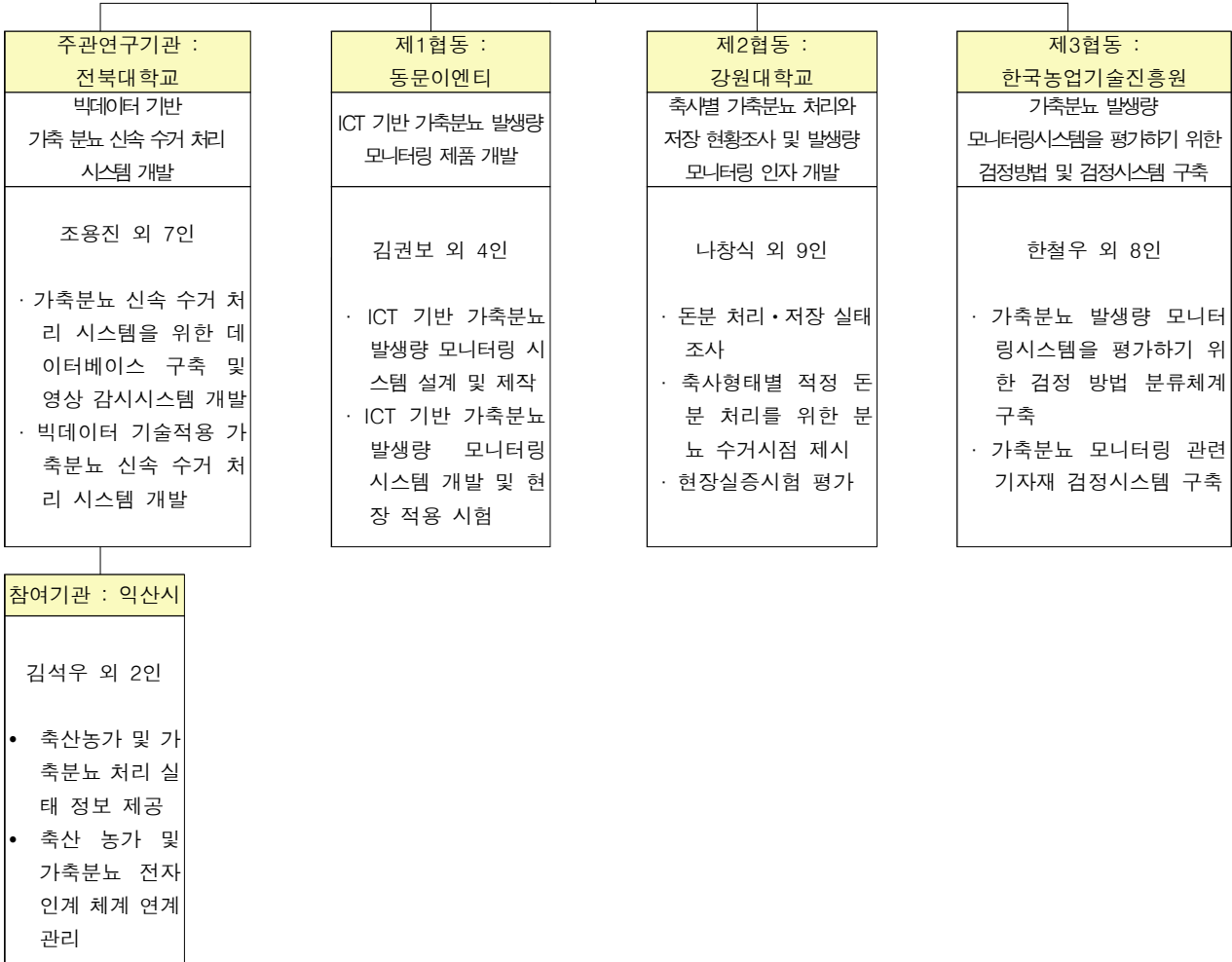
\*\*\* 가중치 총합 100을 기준으로 성과목표지표별 중요도, 난이도에 따라 배분하되 가중치 총합이 100이 되도록 배분(사업화지표에 60 이상 배분)

# 1-5. 연구개발의 추진전략·방법 및 추진체계

## 가. 연구개발 추진체계

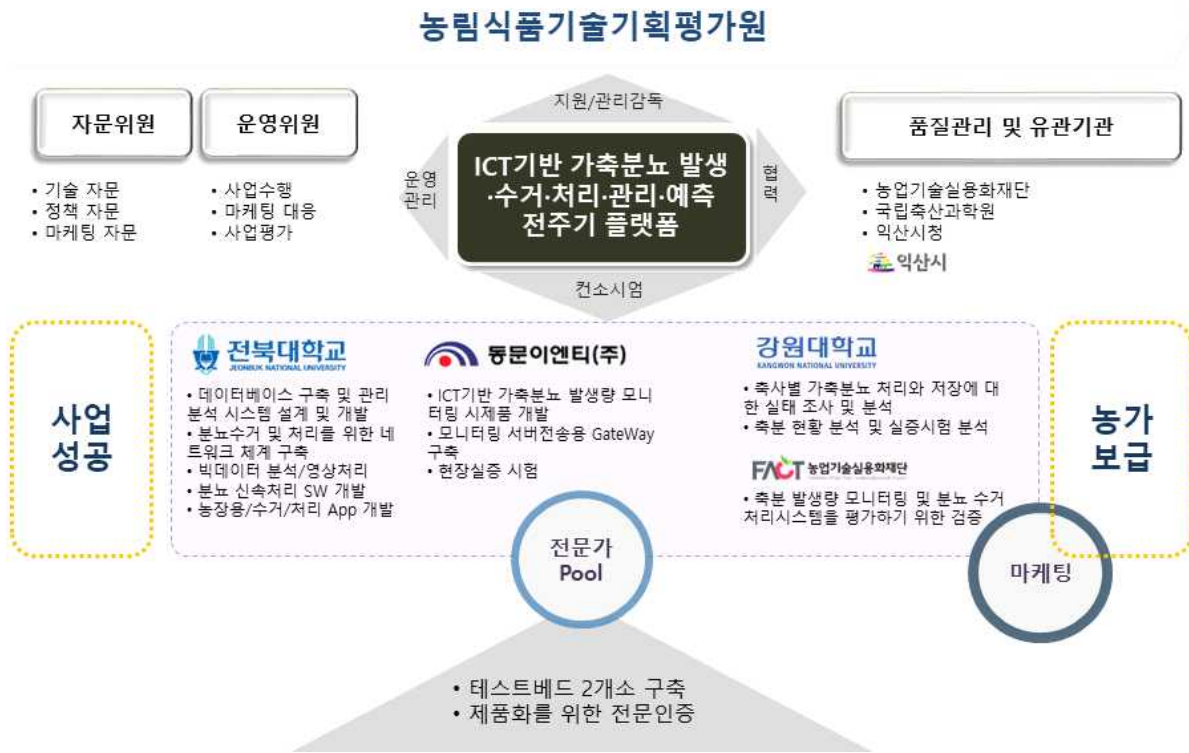
연구개발과제		총 참여 연구원
과제명	ICT 기반 가축분뇨 발생량 모니터링 및 신속 수거·처리시스템 개발	주관연구책임자 (조용진) 외 총 34명

기관별 참여 현황		
구분	연구기관수	참여연구원수
대기업	-	-
중견기업	-	-
중소기업	1	5
대학	2	19
국공립(연)	-	-
출연(연)	-	-
기타	2	12





나. 기관별 역할구분



구분	주요업무	비고
주관기관 전북대학교	<ul style="list-style-type: none"> <li>분석/기획/설계</li> <li>데이터베이스 구축 및 관리 분석 시스템 설계 및 개발</li> <li>영상분석 및 네트워크 체계 구축</li> <li>빅데이터 분석을 이용한 분뇨발생량 예측모델 개발</li> <li>농장용 수거 처리 App 개발</li> <li>분뇨 신속처리 SW 개발</li> </ul>	
협동기관1 동문이엔티	<ul style="list-style-type: none"> <li>ICT/IoT 기반 가축분뇨 발생량 모니터링 시제품 개발</li> <li>모니터링 서버 전송용 시스템 구축</li> <li>현장 실증 시험</li> </ul>	
협동기관2 강원대학교	<ul style="list-style-type: none"> <li>양돈분뇨 발생·관리현황 조사</li> <li>분뇨 발생량 및 분뇨물성, 악취발생 경향 분석</li> <li>저장조 모니터링 인자 선별 및 적정 분뇨 수거시점 제시</li> <li>현장실증 시험 및 분석</li> </ul>	
협동기관3 한국농업기술진흥원	<ul style="list-style-type: none"> <li>가축분뇨 발생량 모니터링시스템을 평가하기 위한 검정방법 및 검정시스템 구축</li> </ul>	
참여기관 익산시청	<ul style="list-style-type: none"> <li>농장의 일반사항 (축종, 규모, 위치특성, 지역 농업 특성 등)</li> <li>가축분뇨 관리현황 (저장주기, 분뇨처리방법 및 시설 현황, 정상 가동 여부, 관리관행, 투입비용, 냄새 저감제 지원 등)</li> <li>축사시설 형태 (돈사구조, 바닥형태, 분뇨 저장조 현황 등)</li> <li>테스트베드 농가 확보</li> </ul>	

## 2. 연구개발과제의 수행 과정 및 수행 내용

### <빅데이터 기반 가축 분뇨 수거처리 시스템 개발, 전북대학교>

#### 1. 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축 및 영상 감시시스템 개발

##### 1-1. 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축

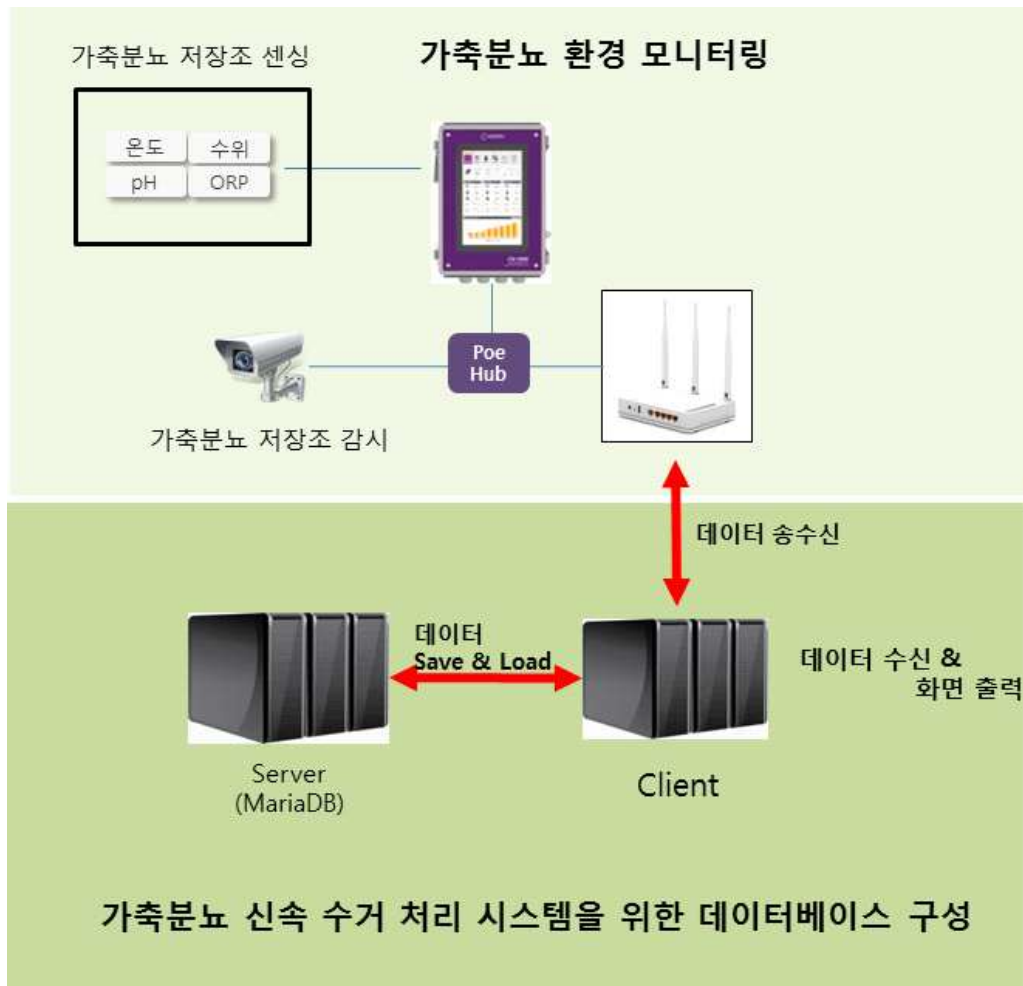


그림 1-13. 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구성

- 빅데이터 기반 가축 분뇨 수거처리 시스템을 개발하기 위해서는 협동기관인 동문이엔티에서 개발하는 가축분뇨 환경 모니터링 시스템에서 전송하는 데이터를 수신하여 화면 출력 등의 방식으로 사용자에게 정보를 제공하고 전송받은 데이터를 서버에 저장하여 테스트베드의 가축분뇨 발생량 정보를 데이터베이스화하여 익산시의 테스트베드의 가축분뇨의 효율적인 관리를 위한 빅데이터를 구축하여야 한다.
- 가축분뇨 환경 모니터링 시스템으로부터 전송받은 데이터를 관리하기 위한 데이터베이스와 전송받은 데이터를 저장하여 빅데이터를 구축하기 위한 서버를 구성하였다.
- 가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축을 위하여 구성한 개발 환경은 표 1-1과 같다.

표 1-13 데이터베이스 개발 환경

		사양
OS	개발 언어	ubuntu18.06 64bit Python + PyQt5 + MariaDB
	개발 IDE	Pycharm + Anaconda3

- 가측분뇨를 센싱한 데이터를 수신하고 데이터베이스에 저장하기 위한 기본 프로그램 구성은 아래와 같다.

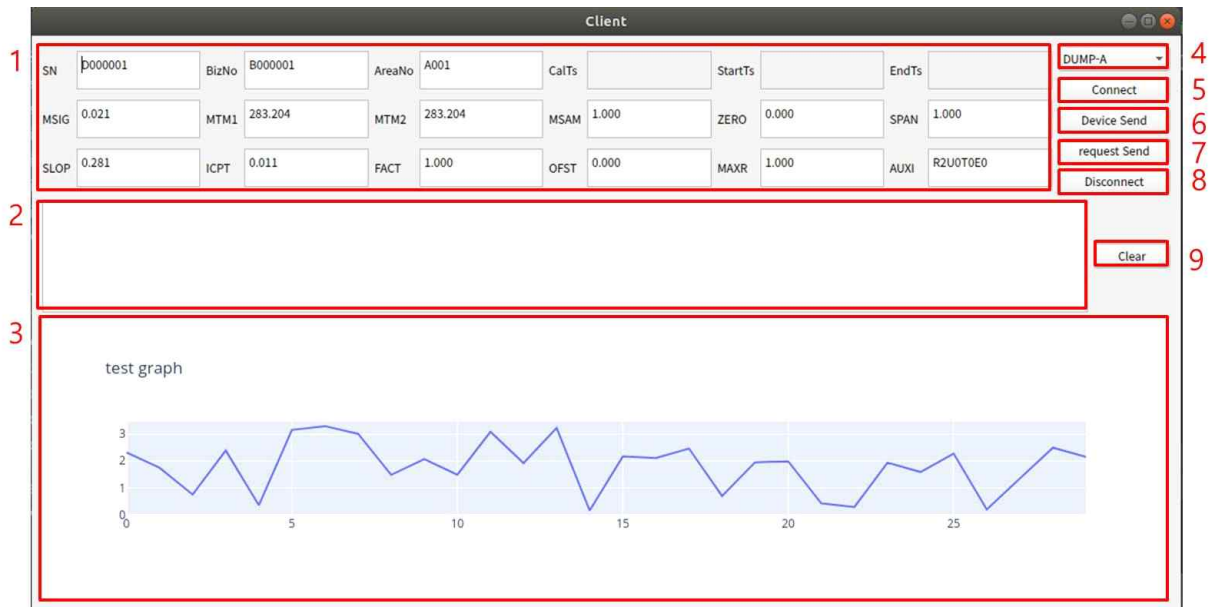


그림 1-14. Dummy Sensor-Client

- 각 프로그램의 구성에 대한 내용은 순차적으로 다음과 같다.

(가) 데이터 입력부 : TEST를 위해 임의의 값을 설정할수 있는 모듈, 시간은 현시간으로 추출되어 비활성화 되어 있으며 나머지 값들은 수정해서 server 단에 전달가능하다.

(나) log 출력부: server-client 통신 시 진행사항을 알 수 있는 로그를 출력한다(그림 2).

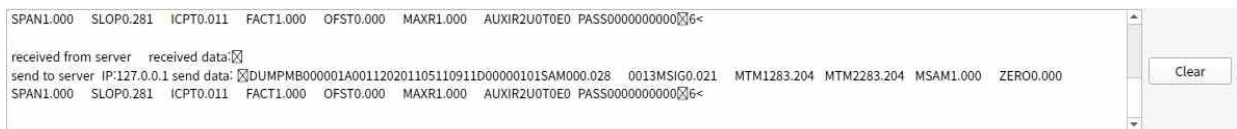


그림 1-15. 통신 시 발생 로그

(다) data 시각화 출력부 : 관련 값들에 대한 시각화를 담당하는 위치 현재는 임의의 값을 출력하고 있으며 차후 다양한 그래프와 시각화 표현을 추가하는 것이 가능하다.

(라) 명령어 설정부 : 해당 프로그램은 환경부 수질 관련 프로토콜을 기반으로 하고 있으며 해당 명령어에 따라 통신과정이 상이하기 때문에 포함되어야 할 명령어를 설정할 수 있는 영역이다.



(마) 커넥트 버튼 : 해당 버튼은 server와의 통신을 위해 우선 소켓을 열어 연결하는 역할을 수행하는 버튼이다. server켜져 있는데 상태에서는 하기 코드와 같이 구동된다.

```
def connectBtn_clicked(self):
    self.clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    self.logTextEdit.appendPlainText("Host: " + str(HOST) + " Port: " + str(PORT))
    ret = self.clientSocket.connect_ex((HOST, PORT))
    if ret == 0:
        self.logTextEdit.appendPlainText("connected")
    else:
        self.logTextEdit.appendPlainText("not connected   errNo: " + str(ret))
```

그림 1-16. 서버 커넥트 코드 예

우선 서버 통신에 사용될 소켓을 생성하고 로그창에 연결이 완료되었음을 알리며, 이후 소켓으로 부쳐 데이터가 전달될 때 까지 대기한다. 유효한 data가 전달될 경우 그에 맞게 구동된다.

(바) 디바이스 send 버튼 : dummy로 만든 데이터를 전달하기 위해 dummy 센서로써 data를 전송하도록 하는 버튼이다.

```
def devSendBtn_clicked(self):
    msg = bytearray(b'device')
    myprotocol = protocol.protocol('DL', 'DUMP-A')
    self.clientSocket.sendall(msg)
    self.logTextEdit.appendPlainText("send" + str(msg.decode()))
    sdCnt = 0
    varstr = self.makeDataSet()
```

그림 1-17. 데이터 전송 코드 예

해당 버튼이 클릭되어 지면 device라는 데이터가 전달되고 서버는 디바이스라는 것을 인식한 후, 실제 데이터를 요청하는 패키지를 전달하게 된다.

```
while True:
    data = self.clientSocket.recv(1024)
    self.logTextEdit.appendPlainText("received from server   received data:" + str(data.decode()))
    sddata = myprotocol.makeCProtocol(data, varstr)
    ckdata = sddata.decode()
    if sdCnt == int(ITMENU):
        sddata = bytearray(EOT.encode())
        self.clientSocket.sendall(sddata)
        self.logTextEdit.appendPlainText("send to server  IP:" + HOST + " send data: " + str(ckdata))
        break
    if ckdata[0] == STX:
        sdCnt = sdCnt + 1
    self.clientSocket.sendall(sddata)
    self.logTextEdit.appendPlainText("send to server  IP:" + HOST + " send data: " + str(ckdata))
```

그림 1-18. 데이터 수신 코드 예

그 후, DATA를 받기위해 소켓은 무한 루프상에서 대기하게 된다.

(사) 리퀘스트 버튼 : 해당 client에서 서버에게 리퀘스트 패킷을 전달해 주는 버튼, 현재는 사용할 필요성이 없어서 비활성화 상태

(아) 디스커넥트 버튼 : server-client 간의 소켓을 닫고 연결을 끊는 역할을 하는 버튼이다.

```
def disconnectBtn_clicked(self):
    msg = bytearray(b'disconnect')
    self.clientSocket.sendall(msg)
    self.logTextEdit.appendPlainText("send msg: " + str(msg.decode()))
    data = self.clientSocket.recv(1024)
    self.logTextEdit.appendPlainText("received msg: " + str(data.decode()))
    self.logTextEdit.appendPlainText("shutdown. disconnect server " ..)
    self.clientSocket.close()
```

그림 1-19. 연결 해제 코드 예

통신이 완료되고 난 이후 프로그램이나 센서가 꺼질 경우, 정상적으로 소켓을 닫고 제거해주어야 차후 다시 통신을 이어나갈수 있기 때문에 해당 작업을 진행해 준다.

(자) 클리어 버튼 : log 출력부에 쌓여있는 로그를 지워주는 역할을 하는 버튼, 현재는 디버깅을 위해서 비활성화 상태이다.

- 센싱한 데이터의 데이터 베이스 서버 세팅은 아래와 같이 수행하였다.

```
conn = mysql.connector.connect(**config)
print(conn)

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

server_socket.bind((HOST, PORT))

server_socket.listen()

client_socket, addr = server_socket.accept()

print('Connected by', addr)

myprotocol = protocol.protocol('VPN', 'DUMP-A')
```

그림 1-20. 서버 세팅 코드 예

Client와의 통신을 위한 소켓을 생성하고 bind-listen-accept의 과정을 거쳐서 연결된다. 해당 코드에서는 MariaDB를 사용을 위한 최초 세팅도 같이 진행한다.

- 데이터베이스 서버와 협동기관에서 제공할 클라이언트간의 통신은 아래와 같이 수행한다.

```

while True:
    data = client_socket.recv(1024)
    if data.decode() == 'device':

        print('Received from',_addr_,' data: ',_data.decode())

        enq = bytearray(ENQ.encode())
        client_socket.sendall(enq)
        print("send from",_addr_,' enq:',_enq.decode())

    elif data.decode() == 'request':
        print('Received from',_addr_,' data: ',_data.decode())
        data = myprotocol.getDBdata(conn)

        client_socket.sendall(data)

    elif data.decode() == 'disconnect':
        print('Received from',_addr_,' data: ',_data.decode())
        client_socket.sendall(data)
        break
    else:
        print('Received from',_addr_,' data: ',_data.decode())
        sddata = myprotocol.makeSProtocol(data,_conn)
        if sddata != None:
            client_socket.sendall(sddata)
            print("send from",_addr_,' sddata:',_sddata.decode())
    
```

그림 1-21. server-client간 통신 코드의 예

client가 접속되고 server의 accept가 완료되면 client socket으로부터 전달 되는 data를 받기 위해 무한 루프의 상태로 대기상태가 된다. 만약 통신에 허용되는 data를 전달 받았을 경우 각 경우에 맞게 구동을 하게 된다. server-client 간의 통신 과정은 다음과 같다.



그림 1-22. 서버 & 클라이언트 간의 통신과정

해당 과정간에 저장 자료 요청 및 요청응답 패키지의 구성은 다음과 같이 구성되어 있다.

- 저장자료 요청

문장의 시작	사업장 (수계코드)	방류구번호 (측정소코드)	명령 일시	명령어	구분 자	측정 기기 ID	시작 일시	끝 일시	문장의 끝	오류검정 코드	패킷의 끝
<STX>				DUMP	M H A				<ET X>	<CHK>	<CR>
(1)	(7)	(4)	(14)	(4)	(1)	(7)	(14)	(14)	(1)	(2)	(1)

그림 1-23. 저장 자료 요청 패키지

- 저장자료 요청응답 (구분자 : M, H)

헤더								
문장의 시작	명령어	구분자	사업장코드 (수계코드)	방류구코드 (측정소코드)	DCD	측정 일시	측정기 기ID	항목 수
<STX> (1)	DUMP (4)	M H (1)	(7)	(4)	(1)	(14)	(7)	(2)

항목1						테일러					
항목 코드	측정값	상태 정보	측정상수	측정상수1		측정 상수n	.....	항목n	문장의 끝	오류검정 코드	패킷의 끝
				상수 코드	상수 값				<ETX>	<CHK>	<CR>
(5)	(10)	(2)	(2)	(4)	(10)				(1)	(2)	(1)

그림 1-24. 요청응답 패키지

통신상에서 해당 data들은 server상의 db에 저장된다.

```

MariaDB [(none)]> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| jbdb |
| mysql |
| performance_schema |
| testdb |
+-----+
5 rows in set (0.015 sec)

Database changed
MariaDB [testdb]> show tables
-> ;
+-----+
| Tables_in_testdb |
+-----+
| REQ_INFO |
| RES_INFO |
+-----+

```

그림 1-25. MariaDB 구성 예 1

그림 1-26. MariaDB 구성 예 2

자료 요청 및 응답패키지는 DB의 REQ\_INFO, RES\_INFO table에 저장되며, 최종적으로 client에서 전달받은 데이터들은 아래와 같이 DB의 테이블에 저장된다.

```

MariaDB [testdb]> select * from REQ_INFO
-> ;
+-----+-----+-----+-----+-----+-----+
| BIZ_ID | AREA_ID | DEV_ID | CMD_DT | STS_DT | ETS_DT |
+-----+-----+-----+-----+-----+-----+
| B000001 | A001 | D000001 | 2020-09-15 15:58:40 | 2020-09-08 15:58:40 | 2020-09-15 15:58:40 |
| B000001 | A001 | D000001 | 2020-09-15 15:59:03 | 2020-09-08 15:59:03 | 2020-09-15 15:59:03 |
| B000001 | A001 | D000001 | 2020-10-08 14:13:49 | 2020-10-01 14:13:49 | 2020-10-08 14:13:49 |
| B000001 | A001 | D000001 | 2020-10-08 14:17:19 | 2020-10-01 14:17:19 | 2020-10-08 14:17:19 |
| B000001 | A001 | D000001 | 2020-10-08 14:19:55 | 2020-10-01 14:19:55 | 2020-10-08 14:19:55 |
| B000001 | A001 | D000001 | 2020-10-12 11:57:56 | 2020-10-05 11:57:56 | 2020-10-12 11:57:56 |
| B000001 | A001 | D000001 | 2020-10-12 11:59:55 | 2020-10-05 11:59:55 | 2020-10-12 11:59:55 |
| B000001 | A001 | D000001 | 2020-10-12 12:29:41 | 2020-10-05 12:29:41 | 2020-10-12 12:29:41 |
| B000001 | A001 | D000001 | 2020-10-12 12:49:09 | 2020-10-05 12:49:09 | 2020-10-12 12:49:09 |
| B000001 | A001 | D000001 | 2020-10-12 12:56:05 | 2020-10-05 12:56:05 | 2020-10-12 12:56:05 |
| B000001 | A001 | D000001 | 2020-10-12 17:26:58 | 2020-10-05 17:26:58 | 2020-10-12 17:26:58 |
| B000001 | A001 | D000001 | 2020-10-12 17:55:02 | 2020-10-05 17:55:02 | 2020-10-12 17:55:02 |
| B000001 | A001 | D000001 | 2020-10-14 15:17:18 | 2020-10-07 15:17:18 | 2020-10-14 15:17:18 |
| B000001 | A001 | D000001 | 2020-10-14 15:18:25 | 2020-10-07 15:18:25 | 2020-10-14 15:18:25 |
| B000001 | A001 | D000001 | 2020-10-14 15:19:50 | 2020-10-07 15:19:50 | 2020-10-14 15:19:50 |
| B000001 | A001 | D000001 | 2020-10-14 15:26:51 | 2020-10-07 15:26:51 | 2020-10-14 15:26:51 |
| B000001 | A001 | D000001 | 2020-10-14 15:33:04 | 2020-10-07 15:33:04 | 2020-10-14 15:33:04 |
| B000001 | A001 | D000001 | 2020-10-14 15:56:21 | 2020-10-07 15:56:21 | 2020-10-14 15:56:21 |
| B000001 | A001 | D000001 | 2020-10-19 14:28:00 | 2020-10-12 14:28:00 | 2020-10-19 14:28:00 |
| B000001 | A001 | D000001 | 2020-11-05 10:46:59 | 2020-10-29 10:46:59 | 2020-11-05 10:46:59 |
+-----+-----+-----+-----+-----+-----+

```

그림 1-27. 데이터 베이스 상 저장 모습 REQ\_INFO

```

MarlaDB [testdb]> select * from RES_INFO;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BIZ_ID | AREA_ID | DEV_ID | MSR_DT | OFST | ITEM_CD | MSR_VAR | STAT_VAR | MSIG | MTM1 | MTM2 | MSAM | ZERO | SPAN |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| SLOP | ICPT | FACT | OFST | MAXR | AUXI | | | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| B000001 | A001 | D000001 | 2020-09-15 15:58:40 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 | |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-09-15 15:59:03 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 273.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-10-08 14:13:49 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-10-08 14:17:19 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-10-08 14:19:55 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-10-14 15:18:26 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-10-14 15:20:11 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-10-14 15:56:23 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-10-19 14:28:00 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
| B000001 | A001 | D000001 | 2020-11-05 10:46:59 | SAM00 | 0.02800000 | 00 | 0.02100000 | 283.20400000 | 283.20400000 | 1.00000000 | 0.00000000 | 1.00000000 |
| 0.28100000 | 0.01100000 | 1.00000000 | 0.00000000 | 1.00000000 | R2U0T0E0 | | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.000 sec)

```

그림 1-28. 데이터 베이스 상 저장 모습 RES\_INFO

## 1-2. 영상정보를 활용한 가축분뇨 발생량 모니터링 기술 개발

- 영상정보를 통한 가축분뇨 발생량 모니터링을 위하여 액비화된 가축분뇨의 성상을 영상장비 및 영상처리 방법을 통하여 분석하는 방법을 제안하였다.



그림 1-29. 분석 액비 샘플

- 영상정보에서 가축분뇨의 성상을 추정할 수 있는 모니터링 기술을 개발하기 위하여 익산시 왕궁면에 소재한 돈사의 가축분뇨 저장조에서 상기 그림과 같은 액비 샘플을 채취하여 액비 샘플의 영상정보를 측정하였다.
- 영상정보 획득을 위해 사용한 영상 장비는 Point Grey 사의 flea2 fl2-03s2c CCD 카메라로 1/3" CCD 센서 크기의 640 X 480 해상도의 컬러 카메라이다. 조명장치는 할로겐 3파장 빛을 이용하였다.
- 영상처리 알고리즘은 채취한 샘플의 용액의 형상을 검출하기 위한 전처리 과정과 각 성상의 영상 단면적을 구하기 위한 영역검출 과정으로 이루어져 있다. 전처리 과정에서는 입력된 영상의 인식이 용이하도록 획득한 영상을 보정하고, 가장자리를 검출하기 위해서 소벨 및 라플라스 가장자리 연산자를 이용하였다. 또한 전체 영상에 대한 농도 히스토그램을 구하여 이를 평활화하여 배경과 구별하였다. 마지막으로 검출된 샘플의 시료영역을 분리하여 용액이 포함된 부분만을 남도록 하였다. 이렇게 획득한 부분의 각 화소의 RGB값과 HSI값을 구하였다. 각 화소의 영상정보는 평균값으로 측정하였다.

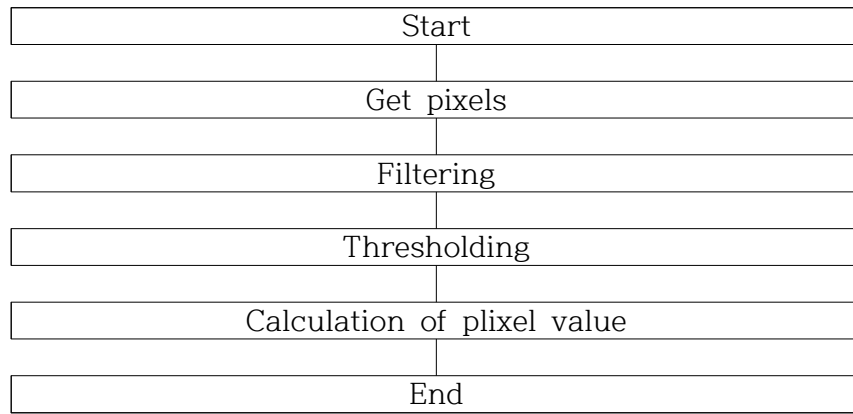


그림 1-30. 영상처리 과정 순서도

- 액비 샘플은 총 9회 채취하여 서로 다른 농도의 둔슬러리를 이용하여 오염물질의 농도에 따른 영상정보 차이를 분석하고자 하였다. 시료의 BOD, COD, SS, VS, T-N, T-P 등의 오염물질 농도분석은 공정시험법에 의해 측정되었으며, CCD 카메라에 의해 획득한 영상정보는 Red, Green, Blue, Hue, Saturation, Intensity를 구하였다.
- 광원은 10w의 일반 조명을 사용하였으며 일정한 조명환경을 조성하고 빛의 유출을 막기 위해 유색 판자를 이용하여 샘플 후미부분을 가렸다.
- CCD 카메라를 통하여 획득한 영상정보와 시료의 오염물질 농도를 포함한 이화학적 농도값을 분석한 결과는 아래의 표와 같다.

표 1-14 오염물질 농도에 대한 해당 영상값

	영상값						오염물질 농도 (mg/l)						
	R	G	B	H	S	I	BOD	COD	SS	T-N	T-P	NH <sub>4</sub> -N	NO <sub>3</sub> -N
1	53	39	56	262.0	0.194	0.341	17651	15147	26000	9193	183	1187	14
2	54	43	60	259.2	0.209	0.325	17215	16269	29000	9097	182	1212	25
3	60	38	56	285.7	0.200	0.375	18523	16467	26000	8904	190	1236	21
4	62	42	56	269.5	0.211	0.390	15604	17655	40000	10685	232	1780	39
5	48	42	58	248.3	0.196	0.378	15906	18850	40000	11616	317	2378	165
6	30	39	56	221.3	0.167	0.632	18272	23760	85000	13156	371	2425	129
7	205	205	158	142.8	0.712	0.892	12282	23364	63000	14162	349	2331	695
8	218	207	141	137.9	0.720	0.894	13289	24948	59000	11744	371	2378	718
9	33	42	57	220.6	0.177	0.571	8215	8435	13600	6276	91	1408	35

- 오염물질의 경우 각 시료간의 뚜렷한 특성을 찾을 수 없었기 때문에, 영상정보와 각 시료의 오염물질 농도와의 상관관계를 살펴보기 위해 각 변수간의 결정계수(R<sup>2</sup>)를 구하고자 하였다.
- 영상정보 R값은 COD값과 결정계수가 0.9213으로 가장 높았으며, 영상정보 G값은 BOD값과 결정계수가 0.9019로 가장 높게 나타났다. 영상정보 B값은 오염물질 농도값 들과 낮은 결정계수 값을 가져 오염물질 농도와의 상관관계가 적은 것으로 판단하였다. 영상정보 H값의 경우 BOD와의 결정계수가 0.9496으로 매우 높게 나타났으며 S의 경우 0.9 이상의 결정계수를 가지는 오염물질 농도는 측정되지 않았다. I의 경우 역시 0.9 이상의 결정계수를 가지는 오염물질 농도는 측정되지 않았다. 따라서 영상정보 중 R 값을 이용하면 COD의 농도를 추정할 수 있을 것으로 판단하였으며, 영상정보 G값과 H값을 이용하면 BOD의 농도를 추정할 수 있을 것으로 판단하였다. 그러나 SS, T-N, T-P, NH<sub>4</sub>-N, NO<sub>3</sub>-N의 농도를 추정하기에는 가시광 영역에서

획득할 수 있는 영상정보인 RGB, HSI 정보는 적합하지 않다는 것을 확인하였다. 좀 더 명확한 상관관계 분석을 위해서는 1차년도 선정된 테스트베드 농가로부터 다양한 시료와 다양한 농도 상황의 가축분뇨를 통하여 새로운 검증이 필요할 것으로 판단된다.

- 가축분뇨 발생량 모니터링을 위한 영상획득을 위해서는 테스트베드에서의 가축분뇨 영상 획득 환경을 구성되어야 할 필요가 있다.
- 시스템 개발 및 실험을 위하여 전라북도 익산시청의 협조를 통해 두곳의 테스트베드를 선정하였다. 선정된 농장의 정보는 아래와 같다.

1) 현대화 농장 : 전라북도 익산시 낭산면 소재



그림 1-31. 현대화 농장 지도

- 현대화 농장 운영 현황

- 전체 사육 두수 : 2,200 두 내외
- 축분 처리 프로세스 :

저류조(축분 비축, 그림 2) → 고액분리기(일 1회, 부정기적) → 고형 및 액비 별도 보관

\* 고형/액비 별도 보관시 처리방법

1. 고형분 교반시설 : 고형분 부숙 진행 (그림 19)
2. 액비저장조 : 호기액비화조에서의 발효 전 저장 (200톤 × 2)
  - 가. 가축분뇨 자원화시설 표준 설계에 따라 시공
  - 나. 지하저장조 크기 (200 m<sup>3</sup>) 기준 높이 3 m 이내 예상

\*\* 고형/액비 수거 시기

1. 고형분 수거 : 부정기적

2. 액비 수거 : 주 1회 (액비저장조 또는 호기액비화조 내 액비 수거)



그림 1-32. 축분 저장조 (저류조)



그림 1-33. 고히분 교반시설



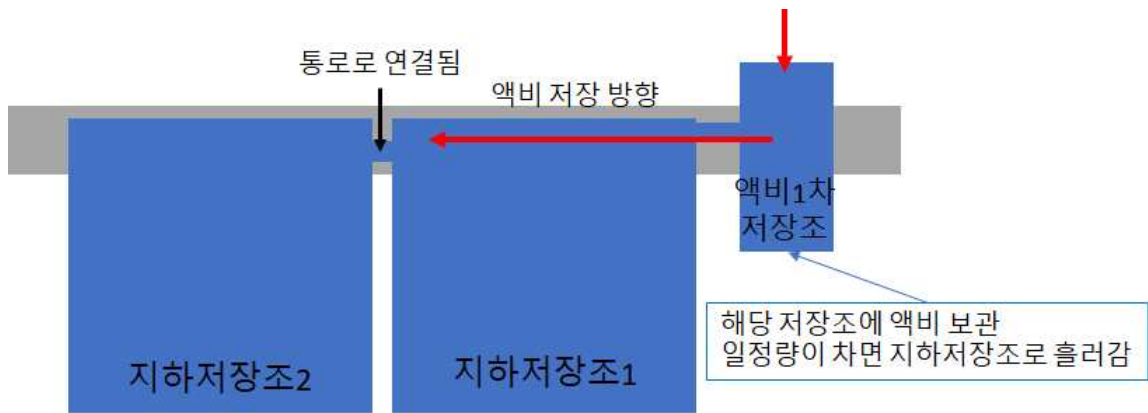


그림 1-34. 액비 저장조 side view



그림 1-35. 액비 1차 저장조



그림 1-36. 액비 저장조



그림 1-37. 액비 저장조 위치 및 예상 규모

2) 재래식 농장 : 전라북도 익산시 왕궁면 소재



그림 1-38. 재래식 농장 지도



그림 1-39. 축분 저장조 (각 축사 방향에서 축분 유입)

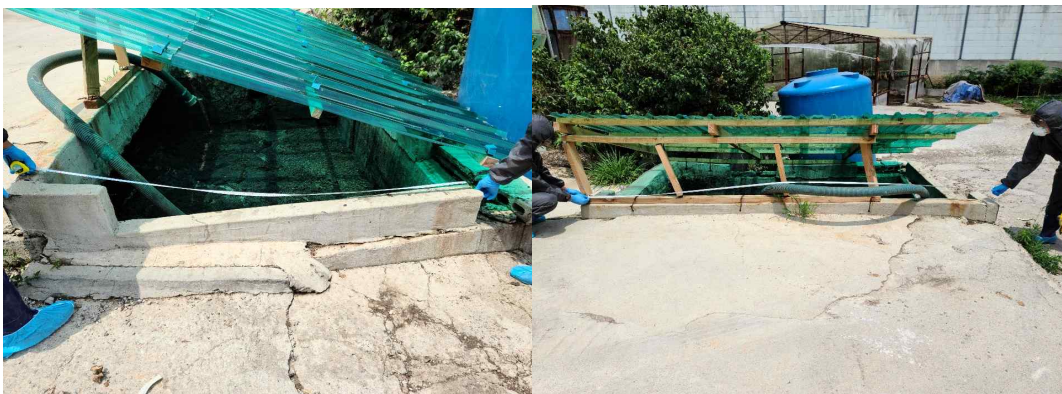


그림 1-40. 저장조 노출부 (220cm X 370cm 내외)



그림 1-41. 저장조 지하부 (전체 370cm X 650cm 내외)

- 재래식 농장 운영 현황

- 전체 사육 두수 : 1500 두 내외
- 축분 처리 프로세스 : 저류조 → 저장 (그림 26, 27, 28)
  - \* 2-3일 간격으로 7톤 또는 15톤 규모 운반 차량에 의해 지역별 공동축분 자원화시설로 축분 수거

- 가축분뇨 저장조는 야외에 노출되어 있기 때문에 영상정보를 통하여 발생정보를 모니터링하기 위해서는 저장조 안을 야외의 노출된 환경에서 영상 장비를 이용하여 감시하여야 한다. CCTV 용 카메라(Qinroiot, ATTENTION, China)를 이용하여 가축분뇨 저장조 내 발생한 분뇨량을 모니터링하는 영상 시스템을 구축하였다. 카메라는 wifi 2.4G 연결을 통하여 영상이 전송되며 축산환경 노출에 견딜 수 있도록 방진/방수가 가능한 IP66급의 카메라를 선정하였다.



그림 1-42. 영상정보 모니터링 위한 CCTV

## 2. 빅데이터 기술적용 가축분뇨 신속 수거 처리 시스템 개발

### 2-1. 빅데이터 기술 적용 가축분뇨 발생 예측모델 개발

- 빅데이터 기술 적용 가축분뇨 발생 예측모델 개발을 위하여 시계열 형태로 축적되는 가축분뇨 발생량 데이터베이스를 분석하기 위한 시계열 빅데이터 분석 알고리즘에 대하여 조사하였다.
- 시계열은 가변적이며 무작위적인 인식에 근거하는 성질을 의미한다. 따라서 시계열 데이터는 데이터의 생성 특성에 따라 연속적으로 생성되는 연속시계열과(continuous time series) 이산적으로 생성되는 이산시계열로(discrete time series) 구분된다. 연속적으로 생성되는 시계열 데이터들은 일정한 관측 시차를 나타내며 관측되므로 대부분의 데이터들은 이산시계열의 형태를 지니게 된다. 이러한 시계열 데이터들은 시간에 밀접한 영향을 받기에 관측 시점들 간의 시차(time lag) 데이터 분석에서 중요한 역할을 한다. 시계열 자료 분석에 이용되는 수학적 모델은 데이터의 분포 모양에 따라 크게 선형모형과 비선형모형으로 나누어지며, 선형모형 중 대표적으로 많이 쓰이는 모델은 Autoregressive Moving Average 이고 비선형 모형에는 혼돈 시계열을 생성할 수 있는 모델로 Fuzzy Logic, 신경망 관련의 Neural Network, 자생발생적인 알고리즘을 이용하는 Genetic Algorithm 등이 속한다.
- 시계열 데이터 분석 모델의 요소
  - 시계열 데이터 분석은 기술적으로 시계열 데이터들 사이의 관계를 규명하여(Transfer function detection) 시스템의 프로세스를 관리(Management)하거나 나아가 예측(Forecasting)하는데 응용되고 있다. 이러한 분석 및 예측은 크게 패턴을 지니는 부분과 랜덤한 부분을 구분하는 단계로써 분석이 시작된다. ARCH 모델이나 ARIMA 모델에서 제시되는 랜덤 요소들과는 다르게 시계열 데이터 분석에서 다루는 패턴요소(pattern)는 크게 네 가지로 분류된다.
  - 시계열 데이터를 이루는 주요한 네 가지 패턴은 경향 패턴(trend pattern), 주기 패턴(cycle pattern), 계절성 패턴(seasonality pattern), 그리고 불규칙적인 패턴(irregular pattern)이다. 경향 패턴은 전체적인 데이터의 흐름이 증가 또는 감소하는지, 혹은 안정적인지를 보여주는 장기적인 변화추세를 의미한다. 이 때 시계열 데이터들은 직선 경향이나 이차선 경향 등을 보일 수 있으며 연구 목적과 데이터 특성에 따라 다양한 통계적인 모형이 데이터 분석에 이용되고 있다. 다음으로 주기 패턴은 일정한 기간을 주기로 순환적인 모양을 나타낸다. 결과적으로 시간의 흐름에 따라 상하로 반복되는 변동이 일정한 추세선을 따라 변화하는 것이 주기 패턴이며 이는 삼각함수와 비슷하게 보여 지지만 실질적으로 주기 패턴을 설명하기 위한 함수는 계산 및 구축이 매우 까다롭다. 계절성 패턴은 주, 월, 분기, 년과 같은 시간 단위 별로 유사한 패턴이 반복되는 모습을 보인다. 주기 패턴과 계절성 패턴의 차이점은 계절성 패턴의 주기가 주기 패턴의 주기인 2~3 년보다 짧다는 점이다. 계절성 패턴은 대체로 계절에 따른 영향이 보여 지므로 주기가 1 년 이내로 설정된다. 주기 패턴과 계절성 패턴의 분석에는 ACF 혹은 Fourier method 가 이용된다. 마지막으로 불규칙적인 패턴은 일정한 규칙성이 없고 예측 불가능하게 우연적인 변동이 발생하는 부분을 의미하는데, 대개 수학적인 모델에서 다루는 오차항(error)이 불규칙한 패턴에 해당된다.
- 시계열 데이터 분석 모델 구축 동향
  - 시계열 데이터에 존재하는 경향 패턴, 주기 패턴, 계절성 패턴 그리고 불규칙적인 패턴에 적용되는 실 분석법들과 관련 모델의 구축은 통계학 기반의 데이터 분석법에 기초한다. 시계열 데

이터 분석에는 회귀 분석법(Regression method), Box-Jenkins 법(ARMA and ARIMA), 지수 평활법(Exponential Smoothing) 등이 데이터 특성 및 연구적 목적 차이에 의해 선별적으로 참조되고 있다. 회귀 분석법과 BOX-JENKINS 법은 수학적 이론 모형에 의존하기 때문에 시간에 따른 변동량이 크게 관찰되는 시계열 데이터 분석에 적합하다. 반면에 지수 평활법은 비교적 직관적인 분석법이기 때문에 시간에 따른 변동량이 적은 시계열 데이터에 적용이 용이하다.

- 회귀 분석법(Regression method)

- 회귀(Regress)의 정의는 옛 상태로 돌아가는 것을 의미하며, 회귀 분석은 영국의 유전학자 Francis Galton 에 의해 제시되었다. 회귀 분석법은 종속변수를 설명할 수 있는 독립변수의 존재로써 모델링 되는 가장 기본적인 수학적 예측 모델로, 수학적으로 오랜 기간 동안 연구적으로 검증되며 사용되어오고 있는 방법이다. 통계학에서 다루는 회귀 분석은 관찰되는 연속형 변수들에 대한 독립변수와 종속변수 사이의 상관관계 식을 도출하여 얻은 모형의 적합성을 검증하는 것이다. 연구자들은 회귀 분석 모델을 통하여 결과에 유의한 영향을 끼치는 설명 변수를 수학적으로 설명할 수 있지만, 미래 데이터 예측에는 방법론의 본질적인 한계를 지닌다. 이는 회귀 분석 모델이 설명 변수 예측치를 지니지 않기에 발생하는 한계이다. 한계를 보완하기 위해 연구자들은 회귀 분석 모델의 설명변수를 다룰 때 시간적인 시점  $t$  를 직관적으로 이용하는 대신 자유도에 따라  $t-1$ ,  $t-2$  등을 보완하며 사용한다.

$$Y_t = \mu + \alpha_1 X_{1t} + \alpha_2 X_{2t} + \dots + \alpha_p X_{pt} + e_t$$

수식 1-1. 회귀 분석법(Regression method)

- Box-Jenkins 법(ARMA model)

- ARMA 는 AR(Auto Regressive)과 MA(Moving Average) 두 모형의 결합 모델로써, 데이터에 대한 복합 수학 모델을 구축하여 미래 값을 예측하는 방법이다. 우선 단 하나의 변수에만 연구자가 관심을 가지는 경우 이를 일변량(univariate) 시계열 분석이라고 하는데 이러한 일변량 데이터의 예측치는 과거 값에만 의존하는 특징을 지닌다. 여기서 도출되는 AR(Auto regressive) 모델은 설명변수가 종속변수의 과거 값으로 사용되는 모델이다. 반면 MA(Moving average) 모델은 설명변수가 오차항의 과거 값으로 사용되는 모델이다. MA 모델에서는 종속변수의 미래 값을 예측하기 위하여 설명 변수의 이전 값이 필요하다. 따라서 MA 모델에서는 시점  $t$  의 이전인  $t-1$  의 설명변수 값을 사용하게 된다. 또한 MA 모델에서 시차(time lag)를 고려할 때는  $t-1$ ,  $t-2$ ,  $t-3$  등을 고려해야 한다.

$$AR(p) \text{ model } \parallel x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \dots + \varphi_p x_{t-p} + \epsilon_t$$

$$MA(q) \text{ model } \parallel x_t = \epsilon_t + \theta \epsilon_{t-1}$$

$$x_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q}$$

$$ARMA \text{ model } \parallel x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \dots + \varphi_p x_{t-p} + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q}$$

수식 1-2. Box-Jenkins 법(ARMA model)

- 일반적인 시계열 데이터 분석에 대한 접근법으로는 자체 추정법(self-projecting)이 주로 사용되는데, 이 방법은 최소 개수의 데이터로 신속한 분석 및 처리가 가능하다는 장점을 지닌다. 신속함의 장점에 기인하여 주로 단기적인 예측에 이용되기 때문에 자체 추정법은 여타 다른 데이터 분석법에 대한 초기 분석법으로 이용된다. 그러나 자체 추정법은 적합한

모형을 찾는 체계적인 방법이 존재하지 않으며, 모형의 적합성에 대한 이론적인 검증이 어렵다는 한계를 지닌다. 따라서 자체 추정법은 실험적인 설계와 오류를 반복하며 찾아야 한다는 단점을 지니고 있다. 또한 장기적인 예측에는 자체 추정법이 적합하지 않기 때문에 연구자들은 Box-Jenkins method 를 이용한 ARIMA 모델을 자체 추정법보다 선호하게 되었다.

- ARIMA(Autoregressive Integrated Moving-average)는 넓은 범위의 시계열 데이터의 표현이 가능한 모델이다. ARIMA 는 Box-Jenkins 모델을 다중변수(Transfer function detection)에 적용하는 일반적인 모델로써 구축과정에서 다른 분석법에 비해 특징적인 구축 과정을 지닌다. 구축 과정은 우선 통계적으로 상관함수를 이용하여 주어진 시계열 데이터에 적합한 모델을 식별하는데서 시작하는데, 이 단계의 목적은 모델을 구축하기 위한 중요한 후보 변수 군들을 도출하는 것이다. 다음으로 최소제곱법을 이용하여 오차 제곱합이 최소가 되는 매개변수를 추정하는 과정이다. 이 때 최대 우도 추정법(Maximum likelihood estimation)이 일반적으로 쓰인다. 최소제곱법을 통해서 모델 설계자는 내부의 실제적인 수치를 결정할 수 있다. 다음으로는 구축된 모델을 확인하는 과정인데, 이는 일반적인 통계법을 활용하여 모델의 타당성을 검증하는 것을 의미한다. 이러한 구축과정을 통해 결과적으로 가장 적합한 ARIMA 모델이 선정되면 마지막으로 모델을 예측에 사용한다. 구축된 모델을 활용하여 연구자들은 데이터 예측치와 실제 관측치 사이의 신뢰구간을 계산하여 모델의 정확성을 평가할 수 있다. 시계열 데이터를 모형화하는 ARMA 모델이 시계열 데이터의 주기(cycle)를 이용하기 때문에, 경향(trend)이나 계절성(seasonality) 특징을 지니는 시계열 데이터는 분별과정(Differencing) 통해 경향 혹은 계절성 특징을 없애고 ARMA 모델을 적용시켜야 한다.

- 이동평균법과 지수 평활법(Moving average and Exponential smoothing)
  - ARMA 와 구분되는 평활법은 일정한 모델을 이용하지 않고 시간 축을 기준으로 과거 값들의 평균을 이용하여 미래 예측치를 구하는 방법이다. 평활법은 크게 이동평균법(moving average)과 지수 평활법(Exponential smoothing)으로 나눌 수 있다. 시계열 데이터는 일정한 주기 혹은 불규칙성을 지니고 있으므로 과거 관측의 평균을 통해 데이터의 추세를 파악하여 예측치를 구하는데 이를 이동평균법이라고 정의한다. 데이터의 평균 혹은 중앙치(median) 각각에 동일한 가중치를 부여하여 예측치로 사용하는 이동평균법과는 다르게, 지수 평활법은 기준 시점에서 가까울수록 많은 가중치를 부여하고 시간적으로 멀어질수록 가중치의 값을 감소시켜 데이터 간의 변화를 수적으로 부여한다. 단순지수(Simple exponential smoothing), 이중지수(Double exponential smoothing), 계절지수 평활법(Winsters Seasonal exponential smoothing) 등이 지수 평활법을 이용한 실질적인 모델이다. 결과적으로 이동평균법의 목적이 계절주기 조정을 통한 추세(trend)분석이라면 지수 평활법의 목적은 데이터 예측(forecasting)이다.

$$\hat{Y}_{t+1} = MA_{t,m} = \frac{Y_t + Y_{t-1} + \dots + Y_{t-m+1}}{m}$$

수식 1-3. 이동평균법과 지수 평활법(Moving average and Exponential smoothing)

- Box-jenkins 법과 지수 평활법은 시계열 데이터의 Y 값인  $\{Y_t; t=1,2,\dots,T\}$  만을 이용하여 분석을 진행하므로 일변량 시계열 분석법에 속한다. 반면 단순히 Y 값뿐만 아니라 입력되는 설명변수들이 존재하는 경우를 학자들은 다중 시계열 분석법(Multiple Time Series Analysis)이라고 분류한다. 다중 시계열 분석에는 회귀분석법과 다변량 ARIMA 모델이 속한다. 또한 시계열 데이터를 다룸에 있어 변수에 따른 데이터의 분석 방법적인 특성을 파악한 뒤 실제 분석에 앞서 시간도표(time plot)를 설계하는 과정이 선행되어야 한다. 시간도표는 시계열 데이터의 시간적인 순서를 특징으로 지니는 점에 근거하여 Y 축을  $Y_t$  값 그리고 X 축을 시간으로 고정한 뒤 산점도를 그려 시계열 자료의 개략적인 구조를 파악하는 것이다. 따라서 시간 도표를 구축하는 과정은 모든 시계열 분석의 시작이다.

- 더미 데이터 및 강원대에서 조사하였던 데이터를 활용하여 샘플 데이터베이스를 구축하였으나, 익산시의 테스트베드에서 발생하는 가축분뇨의 발생량 모니터링 추적 감시를 위해서는 테스트베드에서 발생하는 가축분뇨의 발생량을 측정하여 데이터베이스를 구축하고 해당 데이터베이스의 정보를 바탕으로 상기 빅데이터 기반 시계열 분석 방법을 토대로 가축분뇨 발생량 모니터링에 최적화된 빅데이터 기반 예측모델을 구축하여야 한다.
- 시계열기반 가축분뇨 상태 예측 및 수거 작업 시기 알림 모델 개발을 위하여 동문이엔티에서 개발한 가축분뇨 발생량 측정 센서 시스템으로부터 전송되는 데이터와 수거차량 위치 모니터링을 위한 GPS 정보 데이터, 저장조 내 수위 감시를 위한 영상 정보 데이터를 수집하고 이를 분석하여 시범서비스를 구현하였다.

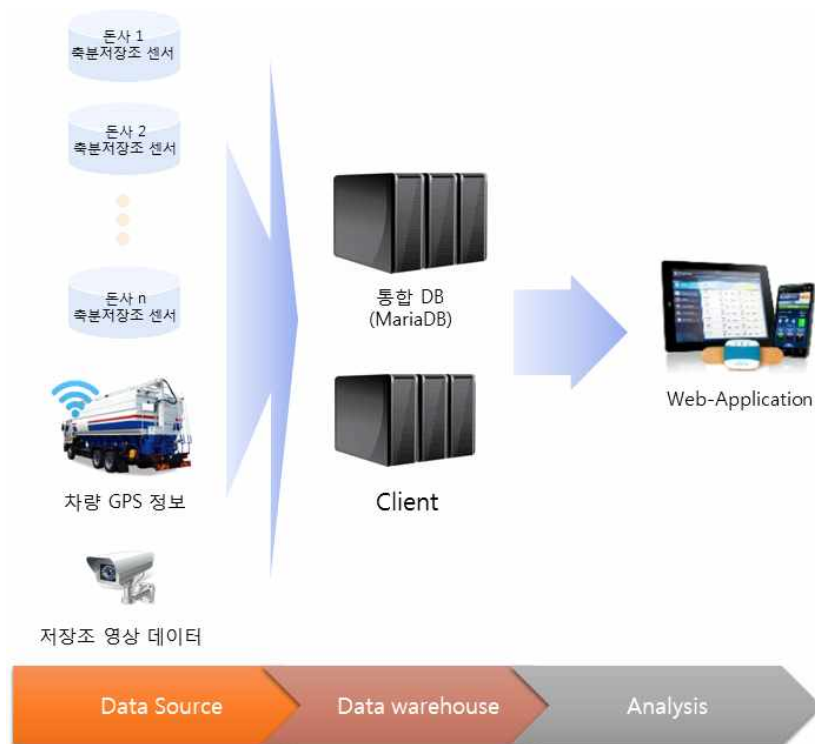


그림 143. 데이터 수집 및 시계열기반 가축분뇨 상태 예측 및 수거 작업 시기 알림 모델 구현

- Node.js를 이용하여 인공지능기술을 이용한 시계열기반 가축분뇨 상태 예측 및 수거 작업 시기 알림 모델 구현하였으며, 시범 서비스를 위해 접근 가능한 농가의 수를 총 50개로 하여 축분저장조의 센서와 GPS정보, 저장조 영상 데이터의 데이터를 수집 및 관리하도록 하였다. 구현한 파일 구성은 아래와 같다.

- consts.js: DB 관련 데이터 정의
- DBCtrl.js: DB 제어 함수 정의
- index\_new\_.js: server 제어 프로그램
- client\_.js: client 제어 프로그램
- sensorsocket.js: 센서 소켓 제어 함수 정의
- websocket\_.js: web 소켓 제어 함수 정의
- index\_.js: http 서비스 관련 프로그램

- Node.js를 이용하여 인공지능기술을 이용한 시계열기반 가축분뇨 상태 예측 및 수거 작업 시기 알림 모델을 구현한 파일의 구체적인 각각 파일의 내용은 아래와 같다.

- consts.js

```
// DB 관련 변수 모음
const localhost = "localhost" // localhost ip
const dbport = 3306 // port number
const main_usr = "mainprocess" // server db 접근 아이디
const sub_usr = "subprocess" // client db 접근 아이디
const ser_usr = "ser_user" // http db 접근 아이디
const main_pw = "choo0211" // server pw
const sub_pw = "choo0211" // client pw
const ser_pw = "choo0211" // http pw
const cn_limit = 50 // 최대 동시 db 접근 횟수

module.exports = [ // 변수 모듈화
  localhost,
  dbport,
  main_usr,
  sub_usr,
  main_pw,
  sub_pw,
  cn_limit,
  ser_usr,
  ser_pw,
];
```

- DBCtrl.js



```

const mariadb = require('mariadb'); //마리아 DB 모듈 로딩
const vals = require('./consts.js');//DB 관련 데이터 모듈 로딩

const mpool = mariadb.createPool({// server 에서 사용할 DB 선언
  host: vals.localhost, port:vals.dbport,
  user: vals.main_usr, password: vals.main_pw,
  connectionLimit: vals.cn_limit
});

const spool = mariadb.createPool({//client 에서 사용할 DB 선언
  host: vals.localhost, port:vals.dbport,
  user: vals.sub_usr, password: vals.sub_pw,
  connectionLimit: vals.cn_limit
});

```

```

async function GetTable(dbname,tbname)// db 내 특정 table 데이터 리턴 함수
{
  let conn, rows;//db 커넥터 및 열 데이터 변수 선언
  try{
    conn = await mpool.getConnection();// db 연결
    conn.query('USE ' + dbname);// db 사용 쿼리 함수
    rows = await conn.query('SELECT * FROM '+tbname);//table 데이터 리턴 쿼리 함수
    console.log(rows) // 데이터 출력
  }
  catch(err){
    console.log(err) //log 에러 출력
    throw err;
  }
  finally{
    if (conn)
      conn.release();//db 연결 해제
      conn.end();
    return rows;
  }
};

```

```

async function PutData(dbname,tbname,strdata)// 데이터 db 입력 함수
{
  let conn;//db 커넥터 변수 선언
  try{
    console.log(strdata.toString());// 쿼리 데이터 로그 출력
    var datalist = [];// 리스트 선언
    conn = await mpool.getConnection();// db 연결
    conn.query('USE ' + dbname);// db 사용 쿼리 함수
    datalist = strdata.split(',');// 데이터를 쉼표 단위로 split
    var check = await conn.query("select EXISTS ( SELECT MSR_TS FROM "+ tbname+" WHERE DEV_NO='"+'+datalist[0]+''+ " AND MSR_TS='"+datalist[1]+'') AS chk");
    //현재 데이터가 db에 존재하는지 중복을 확인 중복일 경우 1 중복이 아닐경우 0
    console.log(check[0].chk);// 중복 여부 결과 로그출력
    if (check[0].chk === 0)//중복이 아닐경우
      await conn.query("INSERT INTO "+ tbname + " VALUES (" + strdata + ")");//db 데이터 입력 쿼리
  }
  catch(err){
    console.log(err)
    throw err;
  }
  finally{
    if (conn)
      conn.release();
      conn.end();
  }
};

```

```

async function GetLatestTime(dno){// 가장최근의 날짜 데이터 출력
  let conn;
  // let late = "";
  try{
    conn = await spool.getConnection();// db 연결
    conn.query('USE testdb');// db 사용 쿼리 함수
    var rows = await conn.query("SELECT MSR_TS FROM DATA_INFO WHERE DEV_NO=" + "" + dno + "" + " ORDER BY MSR_TS DESC ");
    //날짜 데이터를 내림차순으로 정렬하여 데이터를 받아옴
    var temp = rows[0];//가장 최근의 데이터 추출
    var d = new Date();//날짜 변수 선언
    var late = new Date(temp.MSR_TS-(d.getTimezoneOffset()*60000)).toISOString().slice(0,19).replace('T','').replace(/-/g,'').replace(/:/g,'').replace(/'/g,'');
    //KST시간대로 변경하여 날짜 데이터 추출
    // late = "20210501000000"
  }
  catch(err){
    console.log(err)
    throw err;
  }
  finally{
    if (conn)
      conn.release();
      conn.end();
    return late;
  }
}

```

```

async function GetData(dno){// 센서 데이터 취득 함수
  let conn,rows;
  // let late = "";
  try{
    conn = await spool.getConnection();// db 연결
    conn.query('USE testdb');// db 사용 쿼리 함수
    rows = await conn.query("SELECT * FROM DATA_INFO WHERE DEV_NO=" + "" + dno + "" + " ORDER BY MSR_TS DESC ");//특정 센서 데이터 출력 쿼리
    // late = "20210501000000"
  }
  catch(err){
    console.log(err)
    throw err;
  }
  finally{
    if (conn)
      conn.release();
      conn.end();
    return rows;
  }
}

```

```

async function GetTableCol(dbname,tbname)//db의 컬럼 데이터 취득 함수
{
  let conn, rows;//db 커넥터 및 열 데이터 변수 선언
  try{
    conn = await mpool.getConnection();// db 연결
    conn.query('USE ' + dbname);// db 사용 쿼리 함수
    rows = await conn.query('SHOW COLUMNS FROM '+tbname);// 컬럼 데이터 취득
    console.log(rows)// 로그 출력
  }
  catch(err){
    console.log(err)
    throw err;
  }
  finally{
    if (conn)
      conn.release();
      conn.end();
    return rows;
  }
};

```

```

module.exports = { // 함수 모듈화
  getTable:GetTable,
  putData:PutData,
  getLatestTime:GetLatestTime,
  getData:GetData,
  getTableCol:GetTableCol,
};

```

- index\_new\_.js

```
var net = require('net');// net 모듈 로딩
const mariadb = require('mariadb');// 마리아 db 모듈 로딩
//const vals = require('./consts.js');
const { getTable, putData ,getLatestTime} = require('./DBCtrl.js'); // DB 제어 함수 모듈 로딩

let headerSpilt = [4,1,11,1,14,7,2]//헤더 프로토콜 규격
let itemSpilt = [5,10,2,2]//아이템 프로토콜 규격
let reqSpilt = [7,4,14,4,1,7,14,14]// req 프로토콜 규격
var ip1 = '223.171.66.198'; // sensor 1 번 ip
var ip2 = '223.171.66.199'; // sensor 2 번 ip
var port = 5000; // sensor 포트 넘버
var sport = 22222; // server 포트 넘버

//정의된 프로토콜 토큰 정의
const ENQ = "\u0005" //통신 시작 토큰
const ACK = "\u0006"//정상 통신 답변 토큰
const STX = "\u0002"//문장 시작 토큰
const CR = "\u000d"
const ETX = "\u0003"//문장 종료 토큰
const EOT = "\u0004"//통신 종료토큰
const NAK = "\u0015"//비정상 통신 답변 토큰

var server = net.createServer(function(sock){// 소켓을 포함한 서버 선언

  console.log(sock.address() + "connected");//연결된 소켓 로그 출력
  let senNo = [];// 센서 리스트
  let cnt= 0;//센서 수 변수
```

- index\_new\_.js

```
else if (pack.act == 'TIME'){//time msg 입력시
console.log("TIME in");// msg 로그 출력
getLatestTime(pack.id).then((late)=>{//가장 최근의 날짜 데이터 취득
console.log(late); // 날짜 출력
tData = {// 시간 관련 json 데이터 생성
type:"TIMEINFO",
id:pack.id,
data:late
}
sock.write(JSON.stringify(tData)); //문자열화 이후 소켓으로 전송
})
.catch((errMsg) => {
console.log(errMsg);//에러로그
});
}
else if (pack.act == 'PUT'){//put msg 입력시
console.log("PUT in");// msg 로그 출력
console.log(pack.data);
putData("testdb","DATA_INFO",pack.data).then(()=>{//db 데이터 입력 함수
console.log("Input data succeeded!");//로그 출력

inputinfo={//input 정보 json data 선언
type:'INPUTINFO',
id:pack.id,
data:'success'
}
sock.write(JSON.stringify(inputinfo));// 문자열화 된 데이터를 소켓으로 전송
})
.catch((errMsg) => {
console.log(errMsg);//log 출력
});
}
sock.on('data',function(data){// data 수신시 소켓 함수
console.log('클라이언트가 보낸 : ',data.toString());// 클라이언트 전송 로그

pack = JSON.parse(data); //json 파싱 함수
if (pack.act == 'DEV'){// dev msg 입력시
console.log("DEV in");// msg 로그 출력
getTable("testdb","DEV_INFO").then((rows)=>{// db의 DEV_INFO 테이블 데이터 취득
var strdata = [];// string 리스트
for(var i = 0;i<rows.length;i++){//전체 데이터 취득을 위한 for문

devData={//디바이스 데이터 json 데이터 변수 선언
biz:String(rows[i].BIZ_ID),//영역장 아이디
area:String(rows[i].ARE_ID),//지역 아이디
ip:String(rows[i].DEV_IP),//활당 아이피
dno:String(rows[i].DEV_NO),//센서 번호
did:String(rows[i].DEV_ID)//센서 아이디
}

strdata.push(JSON.stringify(devData));//json 데이터의 string 변환 이후 리스트 추가
senNo.push(rows[i].DEV_NO);//센서 번호 리스트에 번호 추가
}

devinfo={//디바이스 정보 관련 json 변수 선언
type:'DEVINFO',
data:strdata.join(',')//토큰을 추가한 데이터
}

sock.write(JSON.stringify(devinfo));//소켓으로 전송
})
.catch((errMsg) => {
console.log(errMsg);//에러 발생시 에러로그
});
}
}
```

```

else if (pack.act == 'DATA'){//data msg 입력시
  console.log("DATA in"); //로그 출력
}
else if (pack.act == 'DONE'){//done msg 입력시
  if(pack.data == 'client'){
    cnt = 0;
  }
  else if(pack.data == 'sensor'){
    cnt++;
  }
  if(cnt<senNo.length){
    startData = {
      type:"SRTDEV",//센서 종료 메시지 설정
      id: String(senNo[cnt])
    }
    sock.write(JSON.stringify(startData));// 문자열화 된 데이터를 소켓으로 전송
  }
  else{
    endData={
      type:'END',//종료 메시지 설정
      data:""
    }
    sock.write(JSON.stringify(endData));// 문자열화 된 데이터를 소켓으로 전송
    senNo = [];//센서 번호
  }
}
}

```

```

server.on('error',function(err){// error 수신시 소켓 함수
  console.log('err' + err);
});

server.listen(sport,function(){// 소켓 listen 실행시
  console.log("listening on ",sport);
})

```

## - client\_.js

```

var net = require('net'); // net 모듈 로딩
const mariadb = require('mariadb');//마리아 db 모듈 로딩
var sensor = require('./sensorsocket.js');// 센서 소켓 모듈 로딩
//const vals = require('./consts.js');
const { getTable, putData ,getLatestTime} = require('./DBCtrl.js');//DB 제어 함수중 필요 함수만 부분 로딩

var sport = 22222;
let ohour = 600000;
//let ohour = 3600000;

//정의된 프로토콜 토큰 정의
const ENQ = "\u0005" //통신 시작 토큰
const ACK = "\u0006"//정상 통신 답변 토큰
const STX = "\u0002"//문장 시작 토큰
const CR = "\u000d"
const ETX = "\u0003"//문장 종료 토큰
const EOT = "\u0004"//통신 종료토큰
const NAK = "\u0015"//비정상 통신 답변 토큰

var serversocket = net.connect({port : sport});// 포트 선언을 포함한 서버 소켓 선언

```

```

serversocket.on('connect',function(){//서버 소켓이 connect 메세지 수신시
  console.log('connected to server');//연결 로그 출력
  console.log('서버와 연결 성공');//연결 확인 로그 출력
  let devInfo = [];//디바이스 정보 리스트 선언
  let senSocks = [];//센서 소켓 리스트 선언
  let timeid = null;//반복 플래그 선언
  // var sen = new sensor(ip1,5000,devInfo,this);

  setupData = { //json 형태의 Dev메세지 생성
    act:"DEV",
    data:''
  }
  serversocket.write(JSON.stringify(setupData));//json 데이터 문자열화 후 서버 소켓으로 전달

  serversocket.on('data',function(chunk){// 서버소켓으로 data 메세지 수신 시
    console.log('서버가 보낸 : ',chunk.toString());// 수신 로그 출력
    packet = JSON.parse(chunk); //json 데이터 파싱
    if(packet.type == 'DEVINFO'){// 패킷 타입이 DEVINFO 일경우
      if (senSocks != []){
        senSocks = [];//센서 소켓 리스트 초기화
      }
      if (devInfo !=[]){
        devInfo = [];//디바이스 리스트 초기화
      }

      var temp = packet.data.toString().split('/');// '/' 로된 단위로 스플릿하여 문자열 리스트로 저장
      for(var i = 0;i<temp.length;i++){// 문자열 리스트를 순회하면서 해당 데이터의 디바이스 정보를 추출하여 devinfo 리스트에 저장
        console.log(temp[i]); // 데이터의 로그 출력
        devInfo.push(JSON.parse(temp[i])); // devInfo 리스트에 관련 데이터 추가
        var sen = new sensor(devInfo[i].ip,5000,devInfo[i],serversocket);
        senSocks.push(sen);
      }
      console.log(devInfo);// devInfo 데이터 로그 출력

      resData={ // response 데이터 생성
        act:'DONE',// 완료 메세지
        data:'client'// 데이터 메세지
      }
      serversocket.write(JSON.stringify(resData));// response 데이터 의 문자열화 및 서버소켓 전송
    }
  })//sensor socket table
}

```

```

else if(packet.type == 'SRIDEV'){ // 패킷 타입이 SRIDEV 일경우
  var sen = new sensor(devInfo[Number(packet.id)-1].ip,5000,devInfo[Number(packet.id)-1],serversocket); // 센서와 통신할 센서 소켓 생성
  senSocks.push(sen); //센서 리스트에 센서 주소 저장
  senSocks[Number(packet.id)-1].write(ENV, 'ascii');//센서 시작을 알리는 프로토콜 로인을 센서 소켓에 전달
}

else if(packet.type == 'TIMEINFO'){// 패킷 타입이 TIMEINFO 일경우

  var d = new Date();// 날짜 데이터 생성
  var today = new Date(d.getTime() - (d.getTimezoneOffset()*60000)).toISOString().slice(0,19).replace('T', '').replace(/-/g,'').replace(/:/g,'').replace(/\./g,'');
  //kst 기준의 날짜 데이터를 생성, 현재 시각을 저장
  var packData = ""; //패킷 데이터가 온 string 변수 선언 및 초기화
  packData = packData + devInfo[Number(packet.id)-1].biz+devInfo[Number(packet.id)-1].area+today + 'DUMP' + 'H'
  + devInfo[Number(packet.id)-1].did + packet.data+today;
  //프로토콜 기준에 따른 패킷 데이터 생성
  var ch = makeChecksum(packData);// 체크섬 생성
  var temp = $TX + packData + $TX + String.fromCharCode(ch[0],ch[1]) + $CR // 체크섬을 포함한 전체 패킷 생성

  console.log(temp); //패킷 로그
  senSocks[Number(packet.id)-1].write(temp, 'ascii');// 아스키 코드로 인코딩된 패킷데이터를 센서 소켓으로 보냄
}

else if(packet.type == 'INPUTINFO'){// 패킷 타입이 INPUTINFO 일경우
  senSocks[Number(packet.id)-1].write(ACK, 'ascii');// 아스키 코드로 인코딩된 패킷데이터를 센서 소켓으로 보냄
}
else if(packet.type == 'END'){// 패킷 타입이 END 일경우
  console.log("All end");//통신 종료 로그
  devInfo = [];//디바이스 리스트 초기화
  senSocks = [];//센서 소켓 리스트 초기화
}
}
}

```

```

timeid = setTimeout(function setTrigger(){//일정 시간 반복을 위한 메크로 함수 선언
  //
  // resData={
  //   act:'DONE',
  //   data:'client'
  // }
  //
  // setupData = { // DEV 메시지를 포함한 json 데이터 생성
  //   act:"DEV",
  //   data:''
  // }
  // serversocket.write(JSON.stringify(setupData));// 서버소켓 으로 데이터 전송
  // timeid = setTimeout(setTrigger,ohour);//특정 시간 이후 재실행되도록 함수 선언
  // },ohour);
});

serversocket.on('end',function(){// end 메시지를 서버 소켓에서 수신받았을 경우
  console.log('클라이언트 접속 종료');// 클라이언트 접속 종료 로그 출력
});

serversocket.on('error',function(err){// error 메시지를 서버 소켓에서 수신받았을 경우
  console.log(err);// 에러 로그 출력
});

serversocket.on('timeout',function(chunk){// timeout 메시지를 서버 소켓에서 수신받았을 경우
  console.log("connect timeout");// 타임아웃 로그 출력
});

```

```
function makeChecksum(pack){ // 체크섬 비트 생성 함수
  var index;
  var checksum = 0x0;
  var checkArr = [];
  var tempt = String(pack);
  for(index =0;index<tempt.length;index++){
    checksum += tempt.charCodeAt(index);
  }

  chk = checksum & 0x00ff;
  checkArr.push(((chk & 0xf0) >> 4) | 0x30);
  checkArr.push(((chk & 0x0f) | 0x30));

  return checkArr;
}
```

## - sensorsocket.js

```
const { Socket } = require('dgram'); // 소켓 모듈 로딩
const net = require('net'); // net 모듈 로딩

let headerSpilt = [4,1,11,1,14,7,2] //헤더 프로토콜 규격
let itemSpilt = [5,10,2,2] //아이템 프로토콜 규격
let reqSpilt = [7,4,14,4,1,7,14,14] // req 프로토콜 규격

//정의된 프로토콜 토큰 정의
const ENQ = "\u0005" //통신 시작 토큰
const ACK = "\u0006" //정상 통신 답변 토큰
const STX = "\u0002" //문장 시작 토큰
const CR = "\u000d"
const ETX = "\u0003" //문장 종료 토큰
const EOT = "\u0004" //통신 종료토큰
const NAK = "\u0015" //비정상 통신 답변 토큰

module.exports = function (_host,_port,_info,_sock){ // 센서 소켓 연결 모듈 선언 (ip,port,devInfo[],client socket)
```

```
const sensocket = new net.Socket(); //소켓 생성

sensocket.connect({host:_host,port:_port},function(){ // 센서 소켓 연결 함수 (ip,port)
  let _minfo = _info; // 시진 devinfo 저장
  console.log('센서와 연결 성공');

  sensocket.on('data',function(chunk){ // 소켓 메시지 처리
    console.log('센서가 보낸 : ',chunk);
    if(chunk == ACK){ //ACK일때, 해당패킷은 ENQ 전송이후 정상 연결시 수신
      //서버에 보낼 json 정보
      startinfo = {
        act:'TIME',
        id:_minfo.dno
      }
      _sock.write(JSON.stringify(startinfo)); //server-client 소켓으로 정보 전달
      console.log("Client: ACK ok");
    }
    else if(String.fromCharCode(chunk[0]) == STX){ // 정을 받는 모든 패킷의 시작 신호,
      //센서에서 전달하는 data 정보이므로 DB에 저장할 데이터를 파싱하여 서버에 전달
      inputData = {
        act:"PUT",
        id:_minfo.dno,
        data:parseData(_minfo,chunk) //프로토콜에 맞게 파싱
      }

      _sock.write(JSON.stringify(inputData));
      console.log("Client: STX ok");
    }
    else if(chunk == EOT){ // 보낸 정보를 모두 송신한 이후 연결 종료를 선언하는 패킷

      console.log("Client: EOT ok");

      sensocket.end();
    }
  });
});
```

```

sensocket.on('end',function(){// 소켓 종료시 발생하는 이벤트 정의

    resData={
        // 센서의 활동이 종료되었음을 서버에 전달하기위해 만드는 json 데이터
        act: 'DONE',
        data: 'sensor'
    }
    _sock.write(JSON.stringify(resData));
    console.log('센서 접속 종료');
});
sensocket.on('error',function(err){//소켓 에러시 발생하는 이벤트 정의, 연결상에서는 모든에러가 파악되지 않기도 할

    console.log(err);
    resData={
        // 센서의 활동이 종료되었음을 서버에 전달하기위해 만드는 json 데이터
        act: 'DONE',
        data: 'sensor'
    }
    _sock.write(JSON.stringify(resData));
    console.log('센서 접속 에러 발생');

    sensocket.end();
});

sensocket.on('timeout',function(chunk){// 소켓 연결시간이 초과되었을 경우 발생하는 이벤트 정의

    console.log("센서 접속 timeout");
    resData={
        // 센서의 활동이 종료되었음을 서버에 전달하기위해 만드는 json 데이터
        act: 'DONE',
        data: 'sensor'
    }
    _sock.write(JSON.stringify(resData));
    sensocket.end();
});
});

return sensocket;

```

```

function parseData(info, strData){// 데이터 파싱 함수
let data = strData.toString('utf-8').slice(1,strData.length-4);// strData 파싱
let retStr = []; //결과 스트림 리스트 선언
let ret = "";// 결과값

for(var i= 0;i< headerSplit.length; i++){//헤더 프로토콜 규격에 따라 순회하며 데이터 추출

    var token = data.slice(0,headerSplit[i]);// 프로토콜의 리스트화

    if(headerSplit[i] == 1 || headerSplit[i] == 7||headerSplit[i] == 4){// 규격에 따른 데이터 스플릿(문장의 시작 토큰, 사업장 번호, 지역 번호)
        data = data.slice(headerSplit[i],data.length);

        continue;
    }
    else if (headerSplit[i] == 11){// 규격에 따른 데이터 스플릿(사업장 번호, 지역 번호)
        var biz = token.slice(0,7);
        var area = token.slice(7,token.length);
        if ((info.biz == biz) && (info.area == area)){
            token = info.dno;
        }
        retStr.push(token);//데이터를 결과 스트림 리스트에 저장
        data = data.slice(headerSplit[i],data.length);
    }
    else if(headerSplit[i] == 2){// 규격에 따른 데이터 스플릿(아이템 항목)
        var k = Number(token);
        data = data.slice(headerSplit[i],data.length);

        for(var l = 0;l<k;l++){

            retStr.push(data.slice(5,15));
            retStr.push(''+data.slice(15,17)+'');

            data = data.slice(19,data.length)
        }
    }
    else{
        retStr.push(''+token+'');
        data = data.slice(headerSplit[i],data.length);
    }
}
ret = retStr.join(',');
return ret
};

```



- websocket\_.js

```

const wsModule = require('ws');// ws 모듈 로딩
const mariadb = require('mariadb');// maria db 모듈 로딩
//const vals = require('./consts.js');

const pool = mariadb.createPool({ // db 커넥터 생성
  host: "localhost", port:3306,
  user: "test01", password: "choo0211",
  connectionLimit: 20
});

async function DBconnection();// db 연결 함수 선언
{
  var conn = await pool.getConnection();//커넥터 연결
  return conn;
};

async function getGPSData(con)// gps data 취득 함수
{
  var gpsdata = "";
  conn = con;
  conn.query('USE GPS_SERVER');// gps 서버 설정
  var row = await conn.query('SELECT * FROM GPS_INFO');// gps 정보 취득

  for( var i= 0;i<row.length;i++){ // 열 데이터를 순회하면서 관련 데이터 취득
    |gpsdata = gpsdata + row[i].GPS_LAT+", "+row[i].GPS_LAG + " ";
  }

  return gpsdata;
};

```

```

module.exports = function(_server)// 웹 소켓 서버 모듈 선언
{
  const wss = new wsModule.Server({server:_server}); // 웹소켓 서버 모듈 선언
  let priv = "";//기존의 데이터 저장
  let timerid = null;// 재실행 플래그 설정
  let connc = null; //db 커넥터 변수

  DBconnection().then((conn)=>{ // db 커넥션 함수 선언
    connc=conn;//db 커넥터 배정
  })
  .catch((errMsg) => {
    console.log(errMsg);//에러 로그
  });

  wss.on('connection', (ws) => { //web 소켓에서 connection 메시지 수신
    ws.on('message', (message) => { // 메시지 수신 시
      message = JSON.parse(message);//json 데이터 파싱
      switch (message.event) { //이벤트에 따른 action 실행
        case 'open': //소켓 오픈시
          console.log("Received: %s", message.event);
          break;
        case 'req': //소켓에서 데이터 요청 패킷 전달시
          timerid = setTimeout(function sendGpsData(){ // 재실행 함수 실행
            getGPSData(connc).then((gpsdata)=>{// gps 데이터 취득 함수 실행
              if (priv == "" || priv != gpsdata ){ //이전 데이터가 바뀌거나 같지 않을 경우
                priv = gpsdata;// gps 데이터 설정
                var sendData={ //전송할 json 데이터 설정
                  event:'res',
                  data:null
                };
                sendData.data = gpsdata; //전송 패킷에 gps 데이터 설정
                ws.send(JSON.stringify(sendData)); // 웹 소켓에 전송
                console.log(sendData);// 전송된 메시지 로그 출력
              }
            })
          }.catch((errMsg) => {
            console.log(errMsg);//에러 로그 출력
          });
          timerid = setTimeout(sendGpsData,1000); //1초마다 재전송
        },1000);
          break;

```

```

        case 'eot': // 종료 데이터 요청시
            if(timerid!=null){
                clearTimeout(timerid); // 재전송 함수 클리어
            }
            timerid = null;
        default:
    }
    ws.on("close",function(code,reason){ //웹소켓이 close 될 시
        if(timerid!=null){
            clearTimeout(timerid); // 재실행 함수 클리어
        }
        timerid = null;
    });
});
});
});
}

```

- index.js의 경우 해당 프로그램 코드가 길어 첨부로 대체한다.
- 빅데이터 분석을 위한 프로그램 구조도는 아래의 그림과 같다.

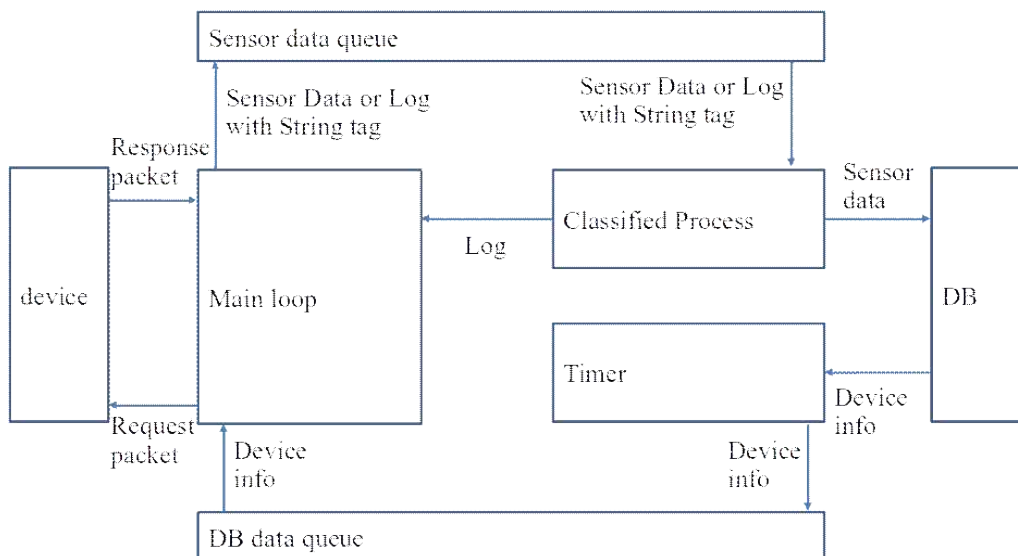


그림 69 빅데이터 분석 프로그램 구조도

- 프로그램의 구조도에서 device는 가축분뇨 관련 데이터를 수집하는 sensor이며, DB는 센서에서 수집된 data를 저장하는 database, 'Sensor data queue'는 sensor에서 수집된 정보들을 classified Process에 전달하기 queue, 'DB data queue'는 DB에서 꺼내온 정보들을 main loop에 전달하기 위한 queue, 'Main loop'는 DB data queue로 전달받은 기본정보를 통해서 센서에 request 패킷을 전달하고, 그에 따른 response 패킷을 파싱한다. 파싱한 정보중 DB에 저장할 데이터들만 추려서 sensor data queue에 전달한다. 'Timer'기능으로 매시간 주기적으로 실행되기 위한 thread, DB에 저장된 센서관련 정보를 가져와 DB data queue에 전달하도록 하였으며, Classified Process: Sensor data queue로 전달받은 데이터를 처리, 데이터에 달린 tag에 따라 DB에 명령할 구문과 log출력할 구문을 분류하고, 액션을 취하도록 하였다.

- 가축분뇨 수거차량의 위치를 GPS로 파악하여 지도상에 표기하고 가축분뇨의 수거 예측 시

기를 확인하기 위한 그래프를 출력하기 위한 DATA 분석 및 GPS 맵핑 프로그램의 구조도는 아래의 그림과 같다.

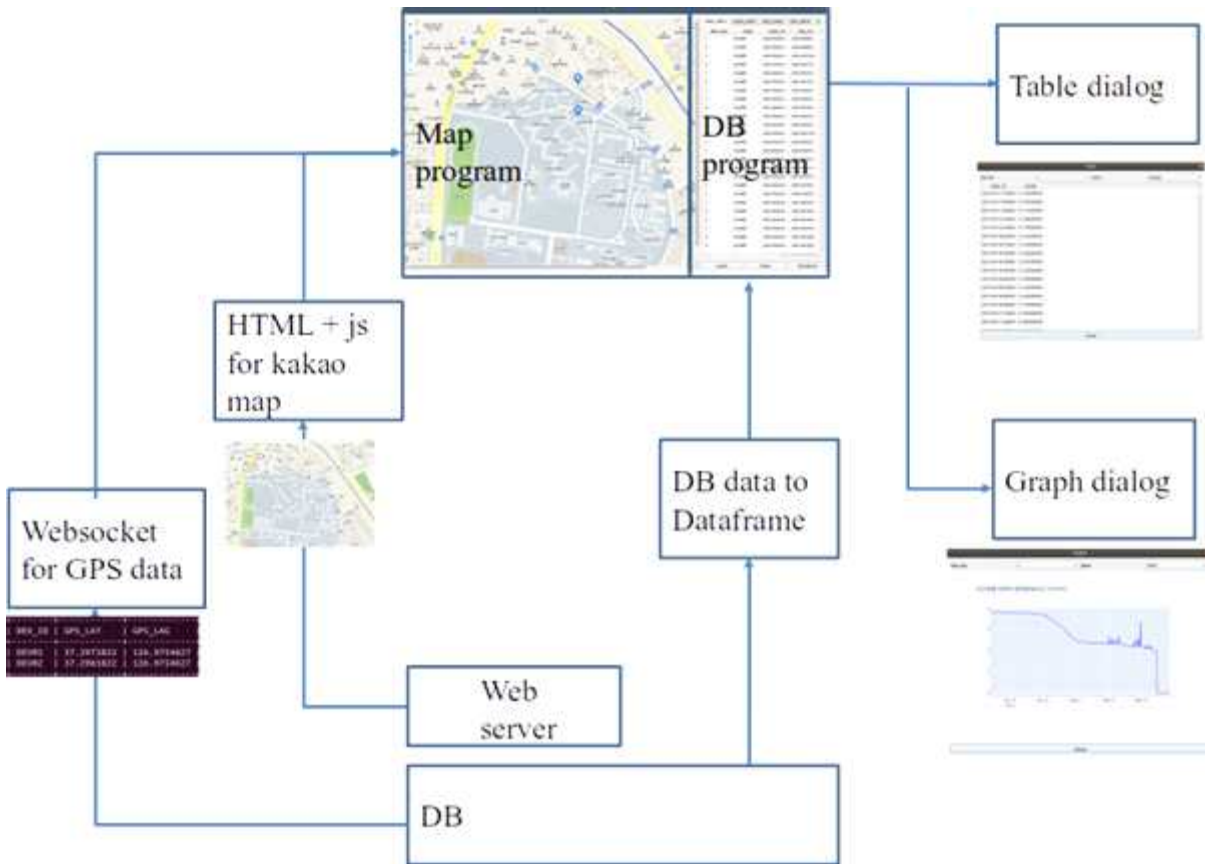


그림 70 DATA 분석 및 GPS 맵핑 프로그램의 구조도

- 프로그램의 개발환경은 ubuntu 18.04와 mariadb를 사용하였으며, 개발언어는 javascript(server), python(client)를 사용하였다. 서버의 개발플랫폼은 node.js이다.
- 프로그램의 파일 구성은 아래와 같다.
  - server
    - index.js: Kakao map API를 통해서 생성한 map 과 좌표를 포함한 web page service 를 제공하는 파일. Node.js 위에서 동작함
    - gpsWebSocketForServer.py: DB Table에 정의되어 있는 GPS정보를 실시간 체크하고, 갱신이 확인되었을 시에는, 갱신된 정보를 websocket을 통해 전달하는 파일
  - client
    - dbcntler.py: DB 접근위한 기능들을 재정의한 클래스 정의
    - clientui.py: Map과 DB접근이 가능한 프로그램
    - tableDialog.py: DB에 저장된 수집 데이터를 table(표)양식으로 구성하여 별도로 확인 할 수 있는 dialog가 정의.(clientui.py의 모듈로 사용)
    - graphDialog.py: DB에 저장된 수집 데이터를 graph 양식으로 구성하여 별도로 확인 할 수 있는 dialog가 정의.(clientui.py의 모듈로 사용)
- DATA 분석 및 GPS 맵핑 프로그램의 구조도에서 각 기능이 수행하는 일은 다음과 같다.

- Map Program: web server 에서 제공하고 있는 page 에 접속, 관련 정보를 web 컨트롤에 출력해주는 프로그램
- Web server: map 관련 정보와 HTML page를 HTTP 로 서비스하는 server
- HTML + js for kakao map: 제공되는 web page는 HTML 양식을 베이스로 하며 웹페이지 상에서 구동되어야 하는 기능들을 위해 일부 javascript로 구성되어 있음,HTML상에서 map과 위,경도 정보를 원활하게 사용을 위해서 kakao map API 를 사용하고 있음
- websocket for GPS Data: DB에서 좌표 정보가 변경될 시, web상에서도 빠르게 갱신하기 위해 java script에서도 사용할 수 있는 websocket을 기반으로 통신
- DB Program: DB 내의 정보를 table형식으로 확인 할 수 있도록 하는 프로그램
- DB: 실제 센서 데이터를 담고 있는 database
- DB data to Dataframe: DB 프로그램에서 관련 데이터를 끌어 올 때 해당 데이터들을 dataframe 형식으로 저장 하기 위한 기능, dataframe일 경우 그래프를 생성과 table 생성 이 유리함
- Table dialog:디바이스 별, 항목별 table 출력을 제공하는 다이얼로그
- Graph dialog:디바이스 별, 항목별 graph트 출력을 제공하는 다이얼로그
- 함수 상세

gpsWebSocketForServer.py

함수	accept		
	<pre> async def accept(websocket, path):# client 연결이 되었을 경우 할 액션을 코루틴 함수로 정의  priv_data = getGpsData() # 최초실행시 기준이 될 데이터 값을 설정 priv = "" # 전달 할 메시지 변수 초기화 for i in priv_data:     priv = priv + str(i[1]) + ',' + str(i[2]) + '/' # 받아온 위도 경도값으로 보낼 문자열 생성 await websocket.send(priv) #웹소켓을 통해 문자열 전달  while True:     gps_data = getGpsData()      if priv_data != gps_data: #최초 진행이 아닐경우 현재 DB속 데이터가 이전 타임 데이터와 동일한지 체크         temp = "" #다를 경우 사용할 변수 초기화         for i in gps_data:             temp = temp + str(i[1])+',' + str(i[2])+ '/'#db로 부터 위도 경도 값 받아 문자열로 변환         await websocket.send(temp)# web socket을 통해 문자열 전달         priv_data = gps_data # 기존의 데이터는 갱신 </pre>		
return	void	param	Websocket,String
기능	비동기식 코루틴 함수, websocket 에서 전달 받은 GPS data와 현재 DB에 저장되어 있는 data가 동일한지 확인, 동일하지 않다면 해당 데이터를 다시 websocket으로 전달하여 좌표를 갱신하는 함수		

함수	getGpsData		
	<pre> def getGpsData(): # db로 부터 gps 데이터를 받아 오는 함수 conn = mysql.connector.connect(**config) # db 커넥터를 통해서 db에 접속 myCursor = conn.cursor()# 명령어 실행을 위한 커서 객체 정의 dtstr = "SELECT DEV_ID, GPS_LAT, GPS_LAG FROM GPS_INFO ORDER BY DEV_ID ASC" # 커서에 전달할 명령어 정의, DEV_ID 기준 오름차순으로 위 myCursor.execute(dtstr)# 해당 명령어 실행 data = myCursor.fetchall()# 명령어 실행후 리턴된 값을 받아와야 할 경우 사용하는 함수  newdata = []# 빈 리스트 선언 for i in data:#받아온 튜플 값을 list 값으로 변경후에 단일 list에 해당 데이터(센서 번호, 위도, 경도)를 분리 저장     temp = []     temp.append(list(i)[0])     temp.append(list(i)[1])     temp.append(list(i)[2])     newdata.append(temp) return newdata # 생성된 list 리턴 </pre>		

return	list	param	void
기능	DB의 gps data와 기기 번호를 가져올수 있는 쿼리문을 생성하여 실행하고, 쿼리문의 결과 값을 list 화 하는 함수		
함수	startserver		
<pre>def startserver():# 코루틴을 통해서 비동기식 형태로 연속적 실행을 하기 위한 함수 정의 start_server = websockets.serve(accept, "192.168.0.6", 9998) # 서버단의 웹소켓 설정, 서버의 ip 와 포트를 인수로 한다. asyncio.get_event_loop().run_until_complete(start_server) # 해당 함수가 종료될때 까지 사용할 수 있는 시스템 루프를 요청후, 시스템 루프를 asyncio.get_event_loop().run_forever()# 해당함수 종료가 될때까지 계속 진행하도록 루프를 설정</pre>			
return	void	param	void
기능	websocket 서버를 생성하는 함수, 코루틴을 사용하여 비동기식으로 구성되어 있으며, 시스템 상에서 시스템 루프를 받아 관련함수를 실행 시키는 기능을 함		

dbcntler.py

함수	dbConnect		
<pre>def dbConnect(self): #db연결 함수 if self.conn is None: # 현재 커넥트가 되어있지 않을 경우 설정된 config 파일로 db 연결 self.conn = self.mySql.connect(**self.config)</pre>			
return	void	param	void
기능	db 연결에 이상 없을 경우 DB로 부터 커넥터를 생성하고 해당 클래스의 변수에 저장하는 함수		

함수	dbDisconnect		
<pre>def dbDisconnect(self):# db 연결 종료 함수 if self.conn is not None:#현재 커넥트가 설정되어 있을 경우 self.conn = None# 커넥트 제거 self.mySql = mysql.connector#새로운 db 접근 클래스 선언</pre>			
return	void	param	void
기능	현재 설정되어 있는 db 커넥트를 제거하고 새로운 db 와 연결할 수 있는 db 연결 클래스를 배정하는 함수		

함수	setConfig		
<pre>def setConfig(self, config):# 전달 받은 컨피그 정보를 저장하는 함수 self.config = config</pre>			
return	void	param	Config
기능	db 연결에 필요한 정보를 담은 config 변수를 db 클래스의 config 변수에 저장 하는 함수		

함수	getCursor		
<pre>def getCursor(self): # db를 컨트롤할 수 있는 커넥터를 전달 받는 함수 if self.conn is not None: #커넥트 배정이 안되어 있을 경우 return self.conn.cursor() # 커넥터 배정 else: return None # 아닐경우는 None을 리턴  </pre>			
return	Cursor	param	void

기능	db 커넥터로부터 쿼리문 형식의 문자열을 받아 SQL을 실행 시킬수 있는 cursor를 배정받는 함수
----	--

clientui.py

함수	initUI		
	<pre>def initUI(self):      self.web = QWebEngineView() # web 컨트롤이 가능한 뷰 선언     self.tableTab = QTabWidget()      bottomWindowLayout = QVBoxLayout()     buttonLayout = QGridLayout()      self.btGraph = QPushButton("Graph") # 그래프 다이얼로그를 호출 할 버튼     self.btTable = QPushButton("Table") # table 다이얼로그를 호출 할 버튼     self.btShutdown = QPushButton("Shutdown") # 종료 버튼      buttonLayout.addWidget(self.btGraph , 0, 0)     buttonLayout.addWidget(self.btTable, 0, 1 )     buttonLayout.addWidget(self.btShutdown, 0, 2)      bottomWindowLayout.addWidget(self.tableTab)     bottomWindowLayout.addLayout(buttonLayout)      mainLayout = QHBoxLayout()     mainLayout.addWidget(self.web)     mainLayout.addLayout(bottomWindowLayout)     mainLayout.setStretchFactor(self.web,7)     mainLayout.setStretchFactor(bottomWindowLayout, 3)     self.setLayout(mainLayout)</pre>		
return	void	param	void
기능	Map&DB 프로그램의 ui를 설정하는 함수		

함수	initCtrl		
	<pre>def initCtrl(self):     self.btGraph.clicked.connect(self.btGraph_clicked)     self.btShutdown.clicked.connect(self.btShutdown_clicked)     self.btTable.clicked.connect(self.btTable_clicked)     self.dbConnect() # 컨트롤 초기화 시점에서 DB 연결</pre>		
return	void	param	void
기능	프로그램 내에서 사용할 컨트롤과 함수를 연결하고 그외 사항들에 대한 초기화를 하는 함수		

함수	dbConnect		

```

def dbConnect(self): # DB 연결 함수
    config = { # DB 접근을 위한 정보 정의
        "user": USERNAME,
        "password": PASSWORD,
        "host": DBHOST,
        "database": DBNAME,
        "port": DBPORT
    }
    self.dbctrl.setConfig(config) # 정의된 정보로 파일을 세팅
    self.dbctrl.dbConnect() # 실제 DB 연결 선언

    tablelist = self.dbGetTableName()# 연결후 출력할 table 이름을 받아옴

    for i in tablelist: #tab 컨트롤을 table이름 만큼 생성하여 토글을 통해 간단히 볼 수 있도록 UI를 구성하고 데이터를 불러오는 함수
        tab = QWidget() # tab 컨트롤 생성
        mTable = QTableWidgetItem() #tab 안에 들어갈 table 위젯 생성
        collist = self.dbGetColName(i) #db에 컬럼으로 설정될 데이터를 전달받음

        mTable.verticalHeader().setVisible(False) #열 인덱스는 필요 없으므로 숨김
        row = self.dbGetRowNum(i) #현재 데이터의 총 열의 개수를 연산하여 전달 받음
        col = len(collist) # 컬럼의 개수 산출
        mTable.setColumnCount(col) # table 컨트롤의 컬럼 개수를 설정
        if row == 0: # 데이터가 없는 경우, 빈 table 을 출력
            mTable.setRowCount(3)
        else:
            mTable.setRowCount(row + 1)# n+1의 열 개수를 확보
            data = self.getTableData(i)# 현재 출력될 정보(수집된 수치)를 모두 전달받아옴
            cnt = 0
            for j in data:# 저장된 데이터 수 만큼 table에 한행씩 저장
                self.putItem(mTable, cnt, j)
                cnt = cnt + 1

        mTable.setHorizontalHeaderLabels(collist)# 컬럼네임 설정
        mTable.setSortingEnabled(True)# 컬럼 네임 보이게 설정
        # mTableView.showGrid()
        tbox = QVBoxLayout()# tab을 레이아웃에 설정
        tbox.addWidget(mTable) # tab 컨트롤에 table 컨트롤을 추가
        tab.setLayout(tbox)# 실제 레이아웃에 추가
        self.tableTab.addTab(tab, i) #tab page를 추가

```

return	void	param	void
기능	DB 연결을 담당하는 함수, 그리고 DB 연결이후, DB 관련 기본 출력을 위해서 해당함수는 tab control과 각 항목 tab을 생성함, 생성한 tab항목에는 각 DB table에 저장되어있는 데이터를 QTableWidgetItem의 표 형식으로 저장하여 출력함		
함수	displayMap		

```

def displayMap(self,url):
    self.web.setUrl(url) # 해당 URL(지도가 구성된 웹페이지)을 web에 출력

```

return	void	param	QUrl
기능	출력할 URL 을 QUrl 형식으로 전달 받고, 해당 url의 webpage를 web 컨트롤에 출력하는 함수		

함수	btShutdown_clicked		
----	--------------------	--	--

```

def btShutdown_clicked(self):
    QApplication.instance().quit #프로그램 종료

```

return	void	param	void
기능	프로그램의 종료버튼에 연결되는 컨트롤 함수, 버튼이 클릭되면 프로그램이 종료됨		

함수	btTable_clicked		
	<pre>def btTable_clicked(self):     dlg = tableDialog.tableDlg(self.dbctrl) #table 다이얼로그를 출력     dlg.exec_()</pre>		
return	void	param	void
기능	table daialog를 불러오는 함수, table 버튼에 연결되어 있음		

함수	btGraph_clicked		
	<pre>def btGraph_clicked(self):     dlg = graphDialog.GraphDlg(self.dbctrl) #그래프 다이얼로그를 출력     dlg.exec_()</pre>		
return	void	param	void
기능	graph daialog를 불러오는 함수, table 버튼에 연결되어 있음		

함수	putItem		
	<pre>def putItem(self,table,index,ldata):# table의 item에 data를 삽입하는 함수      cnt = 0     for l in ldata: # 열데이터에 삽입하되, 데이터형이 datetime일 경우 문자열변환을 별개로 해서 삽입         if (str(type(l))=="&lt;class 'datetime.datetime'&gt;"):             temp = l.strftime('%Y%m%d%H%M%S') # datetime 일때 문자열 변환         else:             temp = str(l) # 일반 문자열 변환         table.setItem(index,cnt,QTableWidgetItem(temp)) #table의 item에 삽입         cnt= cnt + 1</pre>		
return	void	param	QTable,int,list
기능	전달받은 list 데이터를 index에 맞는 item 항목에 분리해서 삽입하는 함수, 다른 변수들은 string으로 캐스팅에 저장하되, datetime 자료형의 경우 string 변환법이 달라 구분해 놓음		

함수	getTableData		
	<pre>def getTableData(self,tableName): #DB를 통해서 table에 출력 데이터를 받아 오는 함수     myCursor = self.dbctrl.getCursor() # db 커넥터를 받아옴      dtstr = "SELECT * FROM " +tableName #각테이블 네임으로 설정된 데이터 호출 쿼리문 작성     myCursor.execute(dtstr)# 쿼리문 실행     data = myCursor.fetchall()# 쿼리문 실행후의 데이터를 전달받음      newdata = []     for d in data:# 전달 받은 데이터를 list 화 함         temp = list(d)         newdata.append(temp)     return newdata</pre>		
return	list	param	string
기능	DB 테이블 명을 인수로 하여 해당 DB테이블의 전체 데이터를 받고, 리스트화 하는 함수		

함수	dbGetRowNum		



```
def dbGetRowNum(self,tableName): # db로 부터 row 개수를 전달 받는 함수
    myCursor = self.dbctrl.getCursor()# db 커넥터를 받아옴

    dtstr = "select count(*) as cnt from " + tableName # 항목의 개수 확인을 위한 쿼리문 작성
    myCursor.execute(dtstr)# 쿼리문 실행
    data = myCursor.fetchall() #쿼리문의 결과를 전달받음
    cnt = list(data[0])
    return cnt[0] # 항목의 개수 리턴
```

return	int	param	String
기능	DB 테이블 명을 인수로 하여 해당 테이블 데이터의 개수를 확인 하는 함수		
함수	dbGetColName		

```
def dbGetColName(self,tablename):# db로 부터 컬럼네임을 전달 받는 함수
    myCursor = self.dbctrl.getCursor()# db 커넥터를 받아옴

    dtstr = "show columns from " + tablename # 컬럼의 네임들을 받을 수 있는 쿼리문 작성
    myCursor.execute(dtstr)# 쿼리문 실행
    data = myCursor.fetchall()#쿼리문의 결과 전달 받음

    newdata = []
    for c in data:# 해당 데이터들을 list 화 함
        temp = list(c)[0]
        newdata.append(temp)
    return newdata
```

return	list	param	string
기능	DB 테이블 명을 인수로 하여 해당 테이블의 컬럼 네임들을 받아오는 함수		

함수	dbGetTableName
----	----------------

```
def dbGetTableName(self):# db로 부터 table name을 전달 받는 함수
    myCursor = self.dbctrl.getCursor()# db 커넥터를 받아옴
    dtstr = "show tables" # 현재 table의 네임을 받아오는 쿼리문 작성
    myCursor.execute(dtstr)# 쿼리문 실행
    data = myCursor.fetchall()# 쿼리문 결과 전달받음
    newdata = []
    for t in data:# 테이블 네임을 리스트로 만들어 리턴
        temp = list(t)[0]
        newdata.append(temp)
    return newdata
```

return	list	param	void
기능	DB 상에 정의되어 있는 모든 테이블 네임을 받아, list 화 하는 함수		

함수	devGetDevName
----	---------------

```
def devGetDevName(self):# db로 부터 디바이스 ID를 전달 받는 함수
    myCursor = self.dbctrl.getCursor()# db 커넥터를 받아옴

    dtstr = "SELECT DEV_ID FROM DEV_INFO ORDER BY DEV_ID ASC"# 오름차순으로 DEV_ID를 정렬하여 받아오는 쿼리문 작성
    myCursor.execute(dtstr)# 쿼리문 실행
    data = myCursor.fetchall()# 쿼리문의 결과 전달받음

    newdata = []
    for n in data:# DEV ID를 리스트화 하여 리턴
        temp = list(n)[0]
        newdata.append(temp)
    return newdata
```

return	list	param	void
--------	------	-------	------

기능	DB내의 디바이스 관련 정보를 저장하고 있는DEV_INFO에서 DEV_ID 정보를 전달 받아 리스트화 하는 함수, 해당 데이터는 오름차순으로 저장되어 있음		
함수	getDevIP		
	<pre>def getDevIP(self,devName):# db로 부터 디바이스 IP를 전달 받는 함수 myCursor = self.dbctrl.getCursor()# db 커넥터를 받아옴 dtstr = "SELECT DEV_IP FROM DEV_INFO WHERE DEV_ID='"+devName + "'#DEV_ID와 매치되는 디바이스 IP를 받아오는 쿼리문 작성 myCursor.execute(dtstr)#쿼리문 실행 data = myCursor.fetchall()# 쿼리문 결과값을 데이터로 저장 return List(data[0])[0]</pre>		
return	string	param	string
기능	string으로 정의된 DEV_ID를 통해 해당 번호와 매칭되는 ip를 찾아 전달하는 함수		

tableDialog.py

함수	initUI		
	<pre>def initUI(self): topWindowLayout = QGridLayout()  devid = QLabel("Dev No") dataItem = QLabel("Item") self.cbDevId = QComboBox()# 테이블 변경인자 , 디바이스 번호 결정 self.cbDataItem = QComboBox()# 테이블 변경인자 , 출력 항목에 대한 결정  topWindowLayout.addWidget(devid,0,0) topWindowLayout.addWidget(self.cbDevId,0,1) topWindowLayout.addWidget(dataItem,0,2) topWindowLayout.addWidget(self.cbDataItem,0,3)  bottomLayout = QVBoxLayout()  self.tableView = QTableWidget()#data 출력을 위한 table 컨트롤 self.btClose = QPushButton("Close")  bottomLayout.addWidget(self.tableView) bottomLayout.addWidget(self.btClose) bottomLayout.setStretchFactor(self.tableView,9) bottomLayout.setStretchFactor(self.btClose,1)  mainLayOut = QVBoxLayout() mainLayOut.addLayout(topWindowLayout) mainLayOut.addLayout(bottomLayout) mainLayOut.setStretchFactor(topWindowLayout,1) mainLayOut.setStretchFactor(bottomLayout,9)  self.setLayout(mainLayOut)</pre>		
return	void	param	void
기능	table dialog 관련 UI를 초기화 하는 함수		

함수	initCtrl		
----	----------	--	--

<pre>def initCtrl(self):     self.btClose.clicked.connect(self.btClose_clicked)     self.cbDevId.currentIndexChanged.connect(self.changeComboBox)     self.cbDataItem.currentIndexChanged.connect(self.changeComboBox)</pre>			
return	void	param	void
기능	다이얼로그 내에서 사용할 컨트롤과 함수를 연결하고 그외 사항들에 대한 초기화를 하는 함수		

함수	btClose_clicked		
<pre>def btClose_clicked(self):     self.close()</pre>			
return	void	param	void
기능	close 버튼과 연결되어 있는 함수, 해당 버튼 실행시 다이얼로그 종료		

함수	changeComboBox		
<pre>def changeComboBox(self):     self.tableUpdate(self.cbDevId.currentText(), self.cbDataItem.currentText())</pre>			
return	void	param	void
기능	combobox 의 값이 변경시 액션을 실행하는 함수, 현재 설정되어 있는 DEV_ID와 수집항목을 확인후, 그에 맞게 table을 업데이트 하는 함수		

함수	initDB		
<pre>def initDB(self):#db 초기화      config = {# db 연결에 필요한 정보 설정         "user": USERNAME,         "password": PASSWORD,         "host": DBHOST,         "database": DBNAME,         "port": DBPORT     }     self.dbctrl.setConfig(config)# 컨피그 파일 세팅     self.dbctrl.dbConnect()# db 연결</pre>			
return	void	param	void
기능	해당 다이얼로그가 사용할 db 커넥터를 생성하고 db에 연결하는 함수		

함수	initCbDevId		

```

def initCbDevId(self):# devID 정보 초기화
self.cbDevId.clear()# 콤보박스 초기화
myCursor = self.dbctrl.getCursor()# db 커넥터 배정
dtstr = "SELECT DEV_NO FROM DEV_INFO ORDER BY DEV_NO ASC" # DEV_NO 오름차순으로 정렬하여 받아오는 쿼리문 작성
myCursor.execute(dtstr)#쿼리문 실행
data = myCursor.fetchall()#쿼리문 결과 데이터 전달 받음

newdata = []
for i in data:# DEV_ID 값들을 리스트하여 콤보박스 항목으로 설정
    temp = list(i)[0]
    self.cbDevId.addItem(str(temp))
    newdata.append(temp)

return newdata

```

return	list	param	void
기능	프로그램 시작시 DEV_ID 관련 combobox를 초기화 하는 함수 , 해당 함수가 실행되면 db에 저장되어 있는 DEV_ID를 받아와서 combobox item 항목으로 추가하고 해당 데이터 재 사용을 위해 list화 해서 리턴함		
함수	initCbDataItem		

```

def initCbDataItem(self):# 출력 수치 정보 초기화
self.cbDataItem.clear()# 콤보박스 초기화
myCursor = self.dbctrl.getCursor()# db 커넥터 배정
dtstr = "SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='DATA_INFO';"#DATA_INFO 에 설정되어있는 모든 컬럼
myCursor.execute(dtstr)# 쿼리문 실행
data = myCursor.fetchall()# 쿼리문 실행 결과를 전달 받음

newdata = []
for i in data:# 전달받은 데이터중 '_'가 포함된 항목은 수치 항목이 아니므로 제외한 나머지 항목을 리스트화 해서 리턴
    temp = list(i)[0]
    if str(temp).find('_') != -1:
        continue
    self.cbDataItem.addItem(temp)
    newdata.append(temp)

return newdata

```

return	list	param	void
기능	프로그램 시작시 저장 데이터 항목 관련 combobox를 초기화 하는 함수 , 해당 함수가 실행되면 db에 저장되어 있는 수집 데이터 항목을 받아와서 combobox item 항목으로 추가하고 해당 데이터 재 사용을 위해 list화 해서 리턴함		

함수	tableUpdate		
----	-------------	--	--

```

def tableUpdate(self,dno,ditem):# 실제 table data 업데이트 하는 함수
self.tableView.clear()# table view 클리어
myCursor = self.dbctrl.getCursor()# db 커넥트 배정
dtstr = "SELECT MSR_TS," + ditem + " FROM DATA_INFO WHERE DEV_NO=" + "'" + dno + "'" + " ORDER BY MSR_TS ASC"# 디바이스 번호로
myCursor.execute(dtstr)# 쿼리문 실행
data = myCursor.fetchall()#쿼리 결과 데이터 전달받음

temp = []
for i in data:# 전달 받은 정보 list화
    temp.append(list(i))

collist = ["MSR_TS",ditem]#table 에 설정할 컬럼네임 list 생성
self.tableView.verticalHeader().setVisible(False) #행의 헤더 숨김
row = self.dbGetRowNum("DATA_INFO") #DATA_INFO 데이터의 개수 전달받음
col = len(collist)# 컬럼의 개수 설정
self.tableView.setColumnCount(col)# table의 컬럼개수 설정
if row == 0:# 데이터가 없을 경우
    self.tableView.setRowCount(3)
else:
    self.tableView.setRowCount(row+1)# 데이터 +1의 행으로 설정
    cnt = 0
    for j in temp:
        self.putItem(self.tableView,cnt,j) #테이블에 각 item 추가
        cnt = cnt + 1
self.tableView.setHorizontalHeaderLabels(collist)# 테이블의 컬럼 네임 설정
self.tableView.horizontalHeader().setSectionResizeMode(0, QHeaderView.ResizeToContents)# 항목 섹션의 크기 조정
self.tableView.horizontalHeader().setSectionResizeMode(1, QHeaderView.ResizeToContents)# 항목 섹션의 크기 조정
self.tableView.setSortingEnabled(True)

```

return	void	param	string, string
기능	DEV_ID와 수집항목 name을 이용하여 해당 항목의 수집시간과 수치를 DB로 부터 전달 받는 함수, 전달 받은 데이터는 다이얼로그에 정의된 table에 정리되어 출력됨		

함수	putItem		
<pre>def putItem(self,table,index,ldata): #테이블에 데이터를 집어 넣는 함수  cnt = 0 for l in ldata:#datetime일 경우는 String 변환 폼에 맞게 변경하고 아닌 데이터는 str로 변경하여 데이터 추가     if (str(type(l))=="&lt;class 'datetime.datetime'&gt;"):         temp = l.strftime('%Y%m%d%H%M%S')# datetime의 string 변환     else:         temp = str(l)     table.setItem(index,cnt,QTableWidgetItem(temp))#table 위젯에 아이템 삽입     cnt= cnt + 1# 다음행</pre>			
return	void	param	QTable,int,list
기능	테이블에 각 data들을 항목에 맞게 저장하는 함수, datetime은 별개의 string변환법을 사용		

함수	dbGetColName		
<pre>def dbGetColName(self,tablename):# db로 부터 컬럼네임을 전달 받는 함수 myCursor = self.dbctrl.getCursor()# db 커넥터를 받아옴  dtstr = "show columns from " + tablename# 컬럼의 네임들을 받을 수 있는 쿼리문 작성 myCursor.execute(dtstr)# 쿼리문 실행 data = myCursor.fetchall()#쿼리문의 결과 전달 받음  newdata = [] for c in data:# 해당 데이터들을 list 화 함     temp = list(c)[0]     newdata.append(temp) return newdata</pre>			
return	list	param	string
기능	DB의 table 이름을 인수로 해서 해당테이블에 설정되어있는 컬럼 네임을 받아오는 함수		

함수	dbGetRowNum		
<pre>def dbGetRowNum(self,tableName):# db로 부터 row 개수를 전달 받는 함수 myCursor = self.dbctrl.getCursor()# db 커넥터를 받아옴  dtstr = "select count(*) as cnt from " + tableName# 항목의 개수 확인을 위한 쿼리문 작성 myCursor.execute(dtstr)# 쿼리문 실행 data = myCursor.fetchall()#쿼리문의 결과를 전달받음 cnt = list(data[0]) return cnt[0]# 항목의 개수 리턴</pre>			
return	int	param	string
기능	DB의 table 이름을 인수로 해서 해당테이블에 저장된 데이터 개수를 받아오는 함수		

graphDialog.py

함수	initUI		

```

def initUI(self):
    topWindowLayout = QGridLayout()

    devid = QLabel("Dev No")
    dataItem = QLabel("Item")
    self.cbDevId = QComboBox()# 그래프 변경인자 , 디바이스 번호 결정
    self.cbDataItem = QComboBox()# 그래프 변경인자 , 출력 항목에 대한 결정

    topWindowLayout.addWidget(devid,0,0)
    topWindowLayout.addWidget(self.cbDevId,0,1)
    topWindowLayout.addWidget(dataItem,0,2)
    topWindowLayout.addWidget(self.cbDataItem,0,3)

    bottomLayout = QVBoxLayout()

    self.graphView = QWebEngineView()#그래프 출력을 위한 web 컨트롤
    self.btClose = QPushButton("Close")

    bottomLayout.addWidget(self.graphView)
    bottomLayout.addWidget(self.btClose)
    bottomLayout.setStretchFactor(self.graphView,9)
    bottomLayout.setStretchFactor(self.btClose,1)

    mainLayOut = QVBoxLayout()
    mainLayOut.addLayout(topWindowLayout)
    mainLayOut.addLayout(bottomLayout)
    mainLayOut.setStretchFactor(topWindowLayout,1)
    mainLayOut.setStretchFactor(bottomLayout,9)

    self.setLayout(mainLayOut)

```

return	void	param	void
기능	graph dialog 관련 UI를 초기화 하는 함수		

함수	initCtrl
----	----------

```

def initCtrl(self):
    self.btClose.clicked.connect(self.btClose_clicked)
    self.cbDevId.currentIndexChanged.connect(self.changeComboBox)
    self.cbDataItem.currentIndexChanged.connect(self.changeComboBox)

```

return	void	param	void
기능	다이얼로그 내에서 사용할 컨트롤과 함수를 연결하고 그외 사항들에 대한 초기화를 하는 함수		

함수	btClose_clicked
----	-----------------

```

def btClose_clicked(self):
    self.close()

```

return	void	param	void
기능	close 버튼과 연결되어 있는 함수, 해당 버튼 실행시 다이얼로그 종료		
함수	changeComboBox		
<pre>def changeComboBox(self): # 콤보박스 변경시 그래프를 업데이트 해주는 함수     self.graphUpdate(self.cbDevId.currentText(), self.cbDataItem.currentText()) # 디바이스 ID와 Item에 맞게 그래프를 업데이트</pre>			
return	void	param	void
기능	combobox 의 값이 변경시 액션을 실행하는 함수, 현재 설정되어 있는 DEV_ID와 수집항목을 확인후, 그에 맞게 graph를 업데이트 하는 함수		
함수	initDB		
<pre>def initDB(self): #db 초기화      config = {# db 연결에 필요한 정보 설정         "user": USERNAME,         "password": PASSWORD,         "host": DBHOST,         "database": DBNAME,         "port": DBPORT     }     self.dbctrl.setConfig(config)# 컨피그 파일 세팅     self.dbctrl.dbConnect()# db 연결</pre>			
return	void	param	void
기능	해당 다이얼로그가 사용할 db 커넥터를 생성하고 db에 연결하는 함수		
함수	initCbDevId		
<pre>def initCbDevId(self):# devID 정보 초기화     self.cbDevId.clear() # 콤보박스 초기화     myCursor = self.dbctrl.getCursor() # db 커넥터 배정     dtstr = "SELECT DEV_NO FROM DEV_INFO ORDER BY DEV_NO ASC" # DEV_NO 오름차순으로 정렬하여 받아오는 쿼리문 작성     myCursor.execute(dtstr)#쿼리문 실행     data = myCursor.fetchall()#쿼리문 결과 데이터 전달 받음      newdata = []     for i in data:# DEV_ID 값들을 리스트하여 콤보박스 항목으로 설정         temp = list(i)[0]         self.cbDevId.addItem(str(temp))         newdata.append(temp)      return newdata</pre>			
return	list	param	void
기능	프로그램 시작시 DEV_ID 관련 combobox를 초기화 하는 함수 , 해당 함수가 실행되면 db에 저장되어 있는 DEV_ID를 받아와서 combobox item 항목으로 추가하고 해당 데이터 재 사용을 위해 list화 해서 리턴함		
함수	initCbDataItem		
<pre>def initCbDataItem(self):# 출력 수치 정보 초기화     self.cbDataItem.clear()# 콤보박스 초기화     myCursor = self.dbctrl.getCursor()# db 커넥터 배정     dtstr = "SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='DATA_INFO';"#DATA_INFO 에 설정되어있는 모든 컬     myCursor.execute(dtstr)# 쿼리문 실행     data = myCursor.fetchall()# 쿼리문 실행 결과를 전달 받음      newdata = []     for i in data:# 전달받은 데이터중 '_'가 포함된 항목은 수치 항목이 아니므로 제외한 나머지 항목을 리스트화 해서 리턴         temp = list(i)[0]         if str(temp).find('_') != -1:             continue         self.cbDataItem.addItem(temp)         newdata.append(temp)     return newdata</pre>			

return	list	param	void
기능	프로그램 시작시 저장 데이터 항목 관련 combobox를 초기화 하는 함수, 해당 함수가 실행되면 db에 저장되어 있는 수집 데이터 항목을 받아와서 combobox item 항목으로 추가하고 해당 데이터 재 사용을 위해 list화 해서 리턴함		

함수	graphUpdate
----	-------------

```
def graphUpdate(self,dno,ditem):# 실제 그래프를 업데이트 하는 함수

df = pd.DataFrame(columns=["MSR_TS",ditem])# 그래프 그리는데 필요한 데이터를 데이터 프레임으로 구성하기위해 선언
myCursor = self.dbctrl.getCursor()#db 커넥터 배경
dtstr = "SELECT MSR_TS," + ditem + " FROM DATA_INFO WHERE DEV_NO=" + "'" + dno + "'" + " ORDER BY MSR_TS ASC"# 디바이스 번
myCursor.execute(dtstr)# 쿼리문 실행
data = myCursor.fetchall()#쿼리 결과 데이터 전달받음

cnt = 0
for i in data:# 리스트로 dataframe 을 만든뒤 그래프화함
    temp = list(i) #list화로 변경
    df.loc[cnt] = temp #데이터 프레임이 행단위로 저장
    cnt = cnt + 1 # 다음행
fig = self.drawGraph(df)# 생성된 데이터 프레임으로 그래프 생성
html = self.showqt(fig)# 그래프를 그린 html 을 생성
self.graphView.setHtml(html) # 웹 컨트롤에 html 설정
```

return	void	param	string,string
--------	------	-------	---------------

기능	DEV_ID와 수집항목 name을 이용하여 해당 항목의 수집시간과 수치를 DB로 부터 전달 받는 함수, 전달 받은 데이터는 plotly로 그래프 화되고, 관련 그래프 정보는 HTML화 되어 출력됨		
----	---	--	--

함수	drawGraph
----	-----------

```
def drawGraph(self,dataf): # 데이터프레임 데이터로 그래프를 그리는 함수

dfList = dataf.columns.tolist() #데이터프레임을 list화 함
line1 = go.Scatter(x=dataf[dfList[0]],y=dataf[dfList[1]],)#시간대별 수치로 데이터를 삽입함, 그래프는 선이있는 scatter형
layout = go.Layout(title="시간대별 데이터 변화량"+"("+dfList[0]+"/"+dfList[1]+")" ) # 해당 그래프의 레이아웃 설정
fig = go.Figure(data=line1,layout=layout)#설정된 옵션으로 그래프 그림
return fig# 그린 그래프 데이터를 리턴
```

return	fig	param	dataframe
--------	-----	-------	-----------

기능	Dataframe의 데이터를 이용해서 그래프를 그리는 함수, 해당 그래프는 plotly package로 그려지고 있으며,그래프 형식은 선이 표시되는 점그래프(line)임, HTML에 올려서 그리기위해 fig 형식으로 저장		
----	---	--	--

함수	showqt
----	--------

```
def showqt(self,fig):#그래프를 html 로 변환하는 함수
raw_html = '<html><head><meta charset="utf-8" />' #캐릭터 셋 설정
raw_html += '<script src="https://cdn.plot.ly/plotly-latest.min.js"></script></head>' # java script 설정 HTML에서 그림을 그리
raw_html += '<body>'
raw_html += '<div style = "display:flex;flex-direction:row;justify-content:center;align-item:center">' # 그래프 레이아웃
raw_html += go.plot(fig, include_plotlyjs=False, image_width=466, image_height=466, output_type='div')# 생성한 그래프 를 HTML
raw_html += '</div>'
raw_html += '</body></html>'
return raw_html #생성한 html 데이터를 리턴
```

return	string	param	fig
--------	--------	-------	-----

기능	fig 파일을 전달받은 뒤, fig를 포함한 html을 만들기 위해 문자열 을 생성하고 반환하는 함수		
----	--	--	--



- 아래의 그림은 DATA 분석 및 GPS 맵핑 프로그램을 통하여 GPS의 위치를 수집하고 수거 시기를 예측하는 프로그램의 실행모습이다.

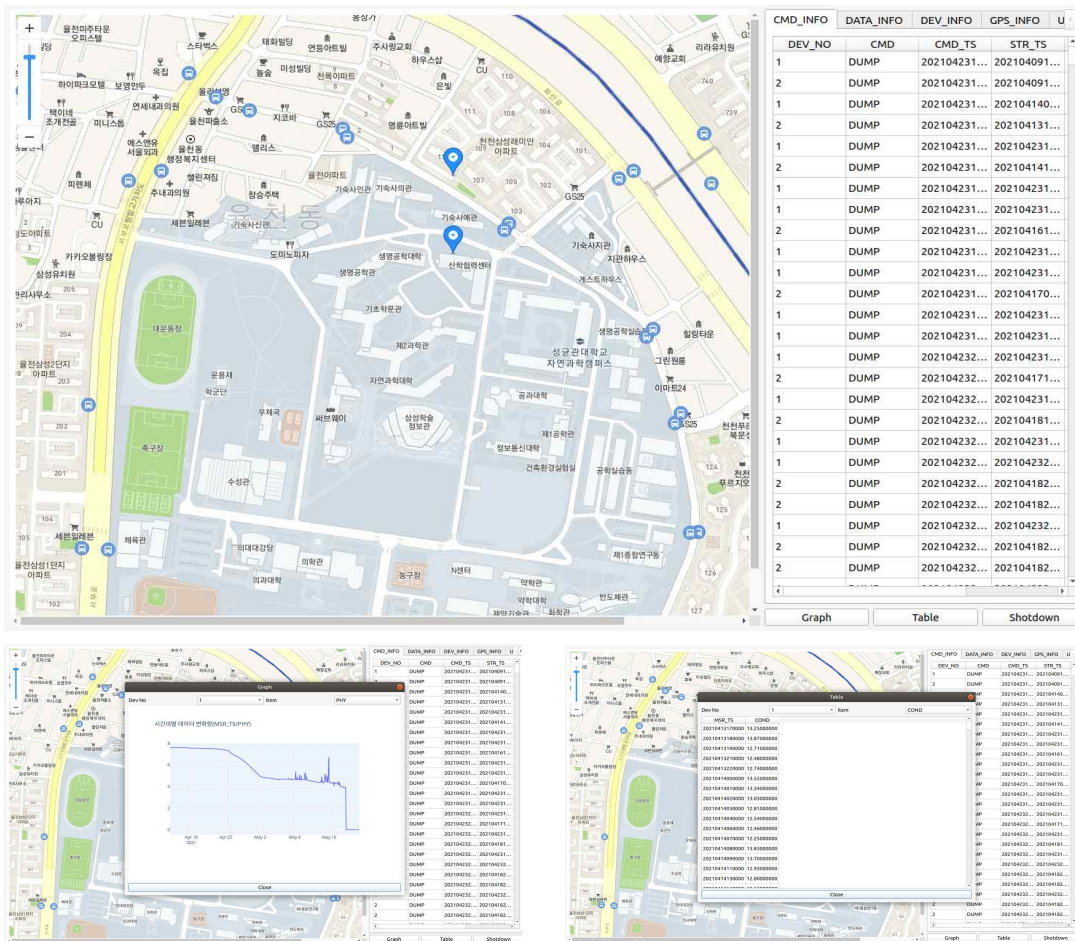


그림 113 DATA 분석 및 GPS 맵핑 프로그램 실행 모습

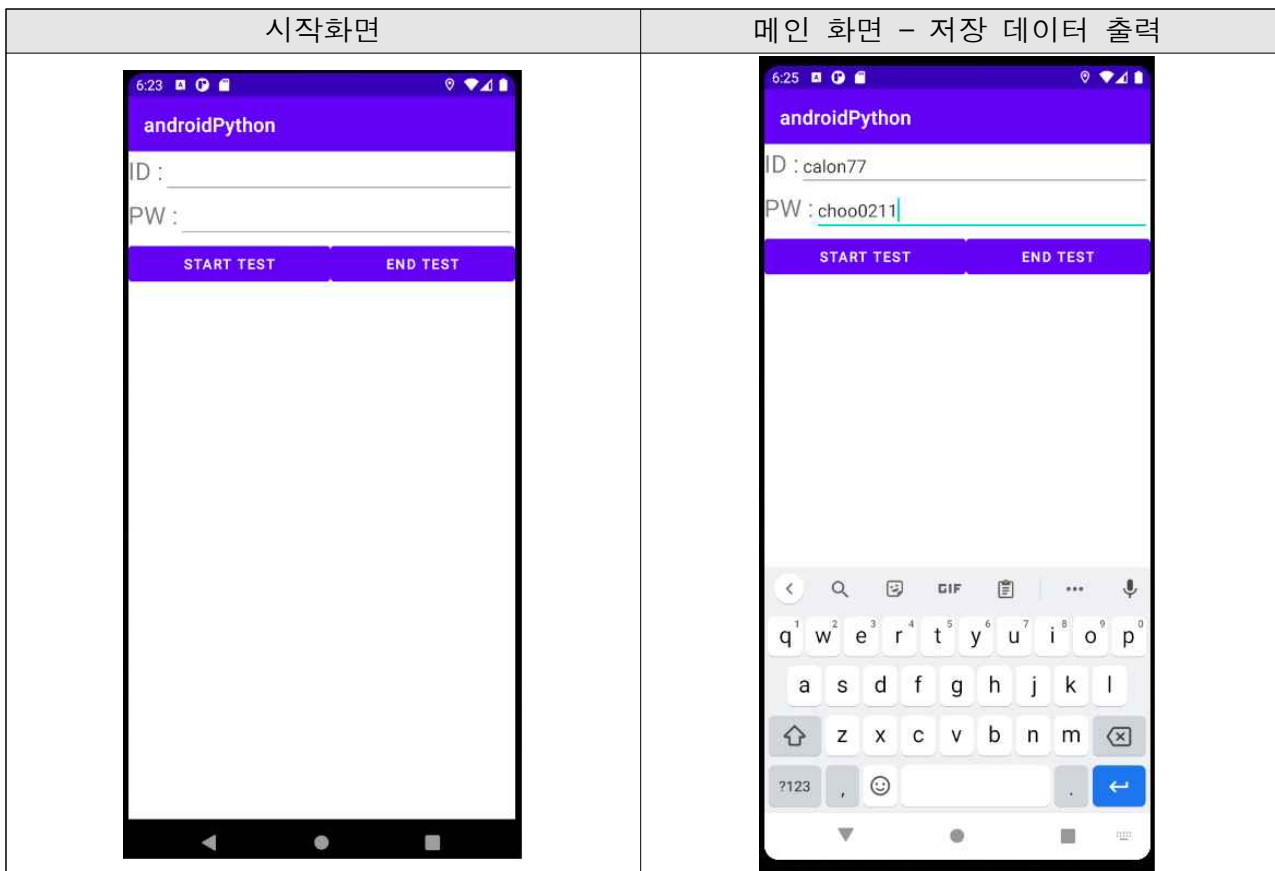
- 빅데이터 기술적용 가축분뇨 신속 수거 처리 시스템의 실제 Web-API를 구현한 그림은 아래와 같다.

표 57 빅데이터 기술적용 가축분뇨 신속 수거 처리 시스템 실제 Web-API

시작화면	메인 화면 - 저장 데이터 출력																																																																																																																																																																																																																																	
	 <table border="1"> <thead> <tr> <th>MSR_TS</th> <th>PHY</th> <th>ORP</th> <th>COND</th> <th>TEMP</th> <th>WATER</th> <th>NH3</th> <th>H2S</th> <th>CO</th> </tr> </thead> <tbody> <tr><td>20210902140000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>0.7</td></tr> <tr><td>20210902130000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>1.44</td></tr> <tr><td>20210902120000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>41.69</td></tr> <tr><td>20210902110000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>27.46</td></tr> <tr><td>20210902100000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>96.67</td></tr> <tr><td>20210902090000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>2.79</td></tr> <tr><td>20210902080000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>2.57</td></tr> <tr><td>20210902070000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>3.42</td></tr> <tr><td>20210902060000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>3.01</td></tr> <tr><td>20210902050000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>3.1</td></tr> <tr><td>20210902040000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>1.7</td></tr> <tr><td>20210902030000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>1.22</td></tr> <tr><td>20210902020000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>1.76</td></tr> <tr><td>20210902010000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>1.41</td></tr> <tr><td>20210902000000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>2.3</td></tr> <tr><td>20210901230000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>2.48</td></tr> <tr><td>20210901220000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>2.2</td></tr> <tr><td>20210901210000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>2.17</td></tr> <tr><td>20210901200000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>2.36</td></tr> <tr><td>20210901190000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.44</td><td>0</td><td>0</td><td>2.04</td></tr> <tr><td>20210901180000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.35</td><td>0</td><td>0</td><td>1.82</td></tr> <tr><td>20210901170000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.42</td><td>0</td><td>0</td><td>1.54</td></tr> <tr><td>20210901160000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>1.67</td></tr> <tr><td>20210901150000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>3.71</td><td>0</td><td>0</td><td>1.33</td></tr> </tbody> </table>	MSR_TS	PHY	ORP	COND	TEMP	WATER	NH3	H2S	CO	20210902140000	0	0	0	0	3.71	0	0	0.7	20210902130000	0	0	0	0	3.71	0	0	1.44	20210902120000	0	0	0	0	3.71	0	0	41.69	20210902110000	0	0	0	0	3.71	0	0	27.46	20210902100000	0	0	0	0	3.71	0	0	96.67	20210902090000	0	0	0	0	3.71	0	0	2.79	20210902080000	0	0	0	0	3.71	0	0	2.57	20210902070000	0	0	0	0	3.71	0	0	3.42	20210902060000	0	0	0	0	3.71	0	0	3.01	20210902050000	0	0	0	0	3.71	0	0	3.1	20210902040000	0	0	0	0	3.71	0	0	1.7	20210902030000	0	0	0	0	3.71	0	0	1.22	20210902020000	0	0	0	0	3.71	0	0	1.76	20210902010000	0	0	0	0	3.71	0	0	1.41	20210902000000	0	0	0	0	3.71	0	0	2.3	20210901230000	0	0	0	0	3.71	0	0	2.48	20210901220000	0	0	0	0	3.71	0	0	2.2	20210901210000	0	0	0	0	3.71	0	0	2.17	20210901200000	0	0	0	0	3.71	0	0	2.36	20210901190000	0	0	0	0	3.44	0	0	2.04	20210901180000	0	0	0	0	3.35	0	0	1.82	20210901170000	0	0	0	0	3.42	0	0	1.54	20210901160000	0	0	0	0	3.71	0	0	1.67	20210901150000	0	0	0	0	3.71	0	0	1.33
MSR_TS	PHY	ORP	COND	TEMP	WATER	NH3	H2S	CO																																																																																																																																																																																																																										
20210902140000	0	0	0	0	3.71	0	0	0.7																																																																																																																																																																																																																										
20210902130000	0	0	0	0	3.71	0	0	1.44																																																																																																																																																																																																																										
20210902120000	0	0	0	0	3.71	0	0	41.69																																																																																																																																																																																																																										
20210902110000	0	0	0	0	3.71	0	0	27.46																																																																																																																																																																																																																										
20210902100000	0	0	0	0	3.71	0	0	96.67																																																																																																																																																																																																																										
20210902090000	0	0	0	0	3.71	0	0	2.79																																																																																																																																																																																																																										
20210902080000	0	0	0	0	3.71	0	0	2.57																																																																																																																																																																																																																										
20210902070000	0	0	0	0	3.71	0	0	3.42																																																																																																																																																																																																																										
20210902060000	0	0	0	0	3.71	0	0	3.01																																																																																																																																																																																																																										
20210902050000	0	0	0	0	3.71	0	0	3.1																																																																																																																																																																																																																										
20210902040000	0	0	0	0	3.71	0	0	1.7																																																																																																																																																																																																																										
20210902030000	0	0	0	0	3.71	0	0	1.22																																																																																																																																																																																																																										
20210902020000	0	0	0	0	3.71	0	0	1.76																																																																																																																																																																																																																										
20210902010000	0	0	0	0	3.71	0	0	1.41																																																																																																																																																																																																																										
20210902000000	0	0	0	0	3.71	0	0	2.3																																																																																																																																																																																																																										
20210901230000	0	0	0	0	3.71	0	0	2.48																																																																																																																																																																																																																										
20210901220000	0	0	0	0	3.71	0	0	2.2																																																																																																																																																																																																																										
20210901210000	0	0	0	0	3.71	0	0	2.17																																																																																																																																																																																																																										
20210901200000	0	0	0	0	3.71	0	0	2.36																																																																																																																																																																																																																										
20210901190000	0	0	0	0	3.44	0	0	2.04																																																																																																																																																																																																																										
20210901180000	0	0	0	0	3.35	0	0	1.82																																																																																																																																																																																																																										
20210901170000	0	0	0	0	3.42	0	0	1.54																																																																																																																																																																																																																										
20210901160000	0	0	0	0	3.71	0	0	1.67																																																																																																																																																																																																																										
20210901150000	0	0	0	0	3.71	0	0	1.33																																																																																																																																																																																																																										
																																																																																																																																																																																																																																		
																																																																																																																																																																																																																																		

- 시작화면에서 권한 있는 사용자가 로그인하여 동문이엔티에서 개발한 저장조 측정시스템으로부터 전송 받은 각 농가의 저장조 상태를 출력하고, 가축분뇨 저장조의 처리를 위해 수거 시기를 예측하기 위한 그래프를 출력하도록 하였다. GPS 위치와 가축분뇨 저장조의 영상 상태를 출력하도록 하였다.
- 실제 수거차량과의 연계를 위해 모바일 GPS 앱을 통해 Web-API와 GPS 위치 정보를 통신하도록 하였다. start test 버튼을 눌러 gps를 활성화하면 Web-API에 GPS의 위치를 출력하도록 하였다. 해당 수거차량용 모바일 GPS 송출 API의 코드는 첨부 2와 같다.

표 58 수거차량용 모바일 GPS 송출 API



## 2-2. 영상정보를 활용한 가축분뇨 발생량 모니터링 위/변조 방지 기술 개발

- 영상정보를 활용한 가축분뇨 발생량 모니터링 위/변조 방지 기술은 저장조 내 발생한 가축분뇨의 수위를 영상정보로 획득하여 센서시스템에서 측정된 정보의 위/변조를 감시하고자 하는 기술이다.



그림 1-121. 가축분뇨 발생량 측정 센서 시스템

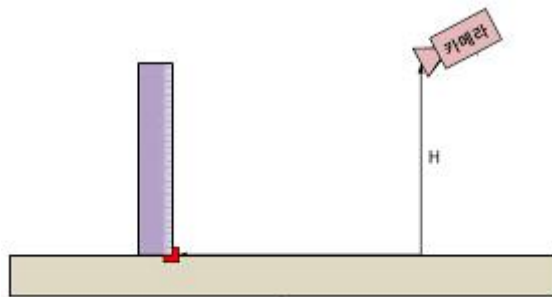


그림 1-122. 가축분뇨 발생량 영상정보 감시 예

- 상기 그림과 같이 가축분뇨 저장조 상의 기준 척자를 일정한 간격에서 카메라로 촬영하여 가축분뇨의 발생 및 수거 시 발생하는 수위의 변화를 탐지하고자 한다. 수위센서에서 발생한 정보와 캘리브레이션하여 영상정보를 활용한 가축분뇨 발생량 모니터링 위/변조 방지 기술의 정확성을 향상시키고 실제 수위 센서와 영상정보를 통해 획득한 수위정보의 차이를 검증할 필요성이 있다.
- 협동기관인 동문이엔티에서 1차년도에 설치가 된 센서 시스템에서 전송하는 데이터와의 비교 검증을 위하여 기술 개발과정에서 선정한 IP카메라를 통하여 영상정보를 통한 수위 정보를 측정하고 수위 센서를 통해 발생한 가축분뇨의 발생량을 검증 필요가 있다.

- 발생한 가축분뇨는 저장조에 축적되므로 축적된 가축분뇨의 발생량과 수거 시 감소되는 양을 영상으로 감시하기 위하여 “가축분뇨 신속 수거 처리 시스템을 위한 데이터베이스 구축” 과정에서 선정된 카메라를 익산시 낭산면에 소재한 농가의 저장조에 시범 설치하였다. 아래의 그림은 카메라가 설치된 저장조와 저장조의 수위를 영상으로 확인하기 위하여 설치한 카메라의 모습이다.



그림 123 익산시 낭산면 돈사의 저장조



그림 124 저장조의 수위 영상 획득을 위해 설치한 카메라

- 저장조의 구조상 실제 높이와의 캘리브레이션을 위한 도구를 설치할 수 없기 때문에 단일 카메라를 이용한 영상 내 거리 추정을 위한 캘리브레이션 과정을 수행하였다.
- 실제로 카메라의 CCD의 위치와 렌즈의 위치를 정확히 찾아내기 어렵기 때문에 거리 측정에 어려움이 따르게 된다. 그러므로 거리를 측정하고자 하는 기준으로부터 초점의 위치까지의 거리를 찾아내야 한다.

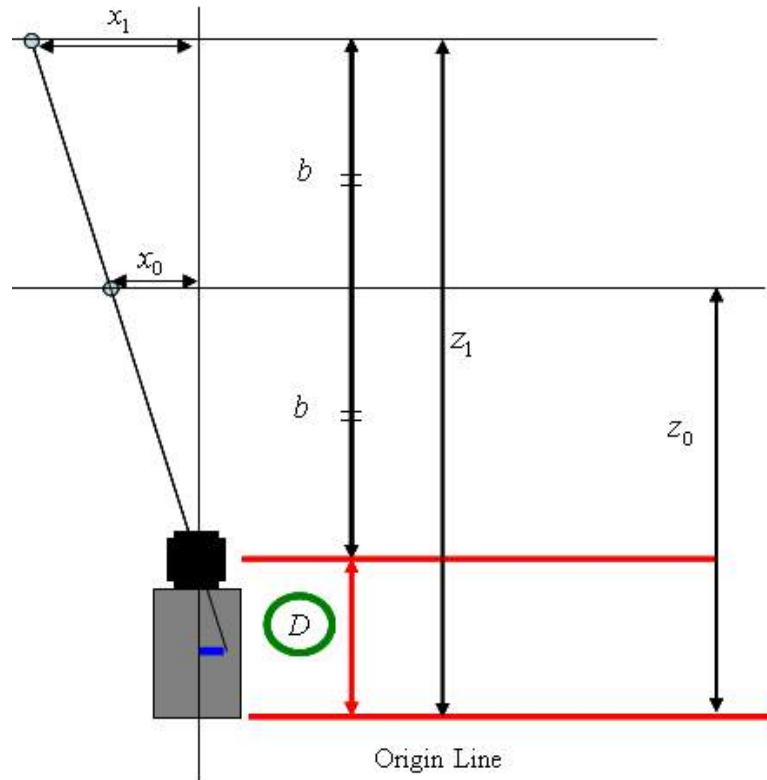


그림 125 초점과 거리의 관계

- 타겟의 중심과 카메라를 일치 시키고 초점으로부터 거리가 2배 되는 점은 CCD상에서 같은 Pixel을 차지한다. 따라서 다음과 같은 식  $b = z_1 - z_0$  과 식  $D = z_0 - b$  을 이용하여 시스템의 기준에서 초점의 위치까지 거리 측정을 할 수 있다. 여기서  $z_0, z_1$ 은 동일한 픽셀 좌표를 같은 서로 다른 거리이다.
- 초점의 위치를 결정한 후에는 카메라로부터 얻어지는 영상에서 픽셀의 좌표값에 따라 초점거리가 어떠한 패턴으로 나타나는지 확인해 볼 필요가 있다. 먼저 타겟에서 아래 그림과 같이 일정한 거리의 포인트를 지정하여 초점거리를 계산한다. 타겟의 중심을 (0,0)으로 잡고 x좌표 -100 ~ 100mm, y좌표 -100 ~ 100mm 사이의 점들을 동일한 간격(20mm)으로 두고 총 100개의 점을 측정하였다. 카메라를 좌우로 회전하여 거리를 측정하게 되므로 x방향에 관한 초점거리만 고려하였다. 아래의 그림에서 한 포인트가 CCD상에 투영될 때 타겟의 중심으로부터 포인트까지의 거리를 알고 또 CCD상의 실제거리를 알 수 있다. 여기서  $P_x$ 는 Pixel의 개수이고  $u$ 는

CCD 단위소자의 x방향 크기이다. 따라서 식  $f = \frac{z \times P_x \times u}{x}$  을 이용하여 초점거리를 찾아낼 수 있다.

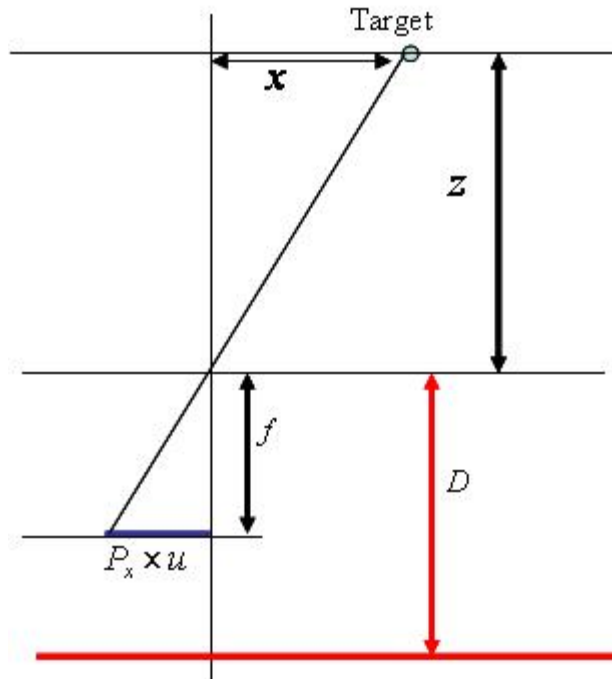


그림 126 초점과 거리의 관계

- 높이의 기준점을 잡기 위해서는 저장조에 저장된 축분의 수면 위치를 찾는 것이 중요하다. 아래의 그림은 가축분뇨 저장조에서의 분뇨의 수면과 저장조 벽면의 경계를 나타낸다.



그림 127 저장된 가축분뇨 경계면과 저장조 벽면의 경계

- 아래의 그림과 같이 해당 경계면과 저장조의 중간 높이를 측정하여 영상을 통한 가축분뇨의 수위를 유추하도록 하였다. 고정된 카메라에서 저장조의 중간 높이에 해당하는 직선은 일정한 영상 좌표에 위치하므로 계산상의 편의를 위하여 해당 길이를 계산함으로써 수위를 추정할 수 있도록 하였다.



그림 128 가축분뇨 수의 추정을 위한 측정 높이

- 해당 높이를 추정하기 위해 카메라의 초점거리와 거리에 대하여 캘리브레이션 후 두 관계에 대한 식을 통해 측정 높이를 계산하였다.
- 가축분뇨의 쌓인 정도의 경계면(수면)을 영상에서 구별하기 위하여 적응형 이진화 방법을 통하여 전처리 하여 배경 등의 잡음을 제거하고 레이블링을 통해 구별된 영역에서 다시 한번 잡음을 제거 후 영상의 우측 영역에서 경계면(수면)을 추출하도록 하였다. 아래의 그림은 경계면(수면)을 구분하기 위한 영상처리 과정을 나타내며, 실제 영상처리 과정을 통해 가축분뇨와 벽면의 경계를 추출하는 모습을 나타낸다.

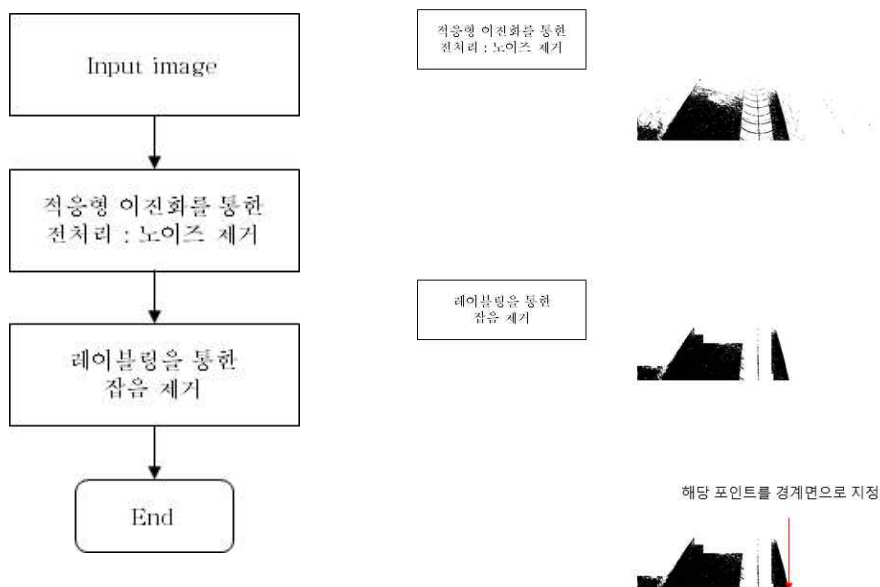


그림 129 경계면 추출과정 (좌 : 순서도, 우 : 실제 처리 과정)



- 영상처리를 이용한 저장조의 수위 추정 알고리즘의 성능을 비교하기 위하여 동문이엔티에서 개발한 저장조 측정 시스템에서 측정한 가축분뇨 저장조의 수위와 비교 검증하였다. 익산시 낭산면에 소재한 농가의 계측데이터 (21년 09월 03일 ~ 21년 09월 05일, 총 3일간 1시간 간격 측정)를 통하여 영상처리를 이용한 저장조의 수위 추정 알고리즘의 성능을 평가하였다.

표 59 계측한 가축분뇨 저장조의 수위와 영상처리를 이용한 저장조의 수위 추정 알고리즘의 결과 비교

측정 일시	수위(m)	수위(m)	오차	측정 일시	수위(m)	수위(m)	오차	측정 일시	수위(m)	수위(m)	오차
20210903000000	3.71	3.623	0.087	20210904000000	3.27	3.197	0.073	20210905000000	3.36	3.271	0.089
20210903010000	3.71	3.68	0.03	20210904010000	3.27	3.235	0.035	20210905010000	3.35	3.291	0.059
20210903020000	3.71	3.643	0.067	20210904020000	3.28	3.18	0.1	20210905020000	3.37	3.315	0.055
20210903030000	3.06	3.005	0.055	20210904030000	3.29	3.247	0.043	20210905030000	3.36	3.276	0.084
20210903040000	3.05	3.001	0.049	20210904040000	3.29	3.273	0.017	20210905040000	3.37	3.345	0.025
20210903050000	3.05	3.011	0.039	20210904050000	3.28	3.214	0.066	20210905050000	3.35	3.323	0.027
20210903060000	3.07	3.014	0.056	20210904060000	3.27	3.266	0.004	20210905060000	3.37	3.284	0.086
20210903070000	3.09	3.069	0.021	20210904070000	3.29	3.258	0.032	20210905070000	2.99	2.955	0.035
20210903080000	3.07	3.021	0.049	20210904080000	3.3	3.275	0.025	20210905080000	3.37	3.358	0.012
20210903090000	3.07	3.009	0.061	20210904090000	3.3	3.206	0.094	20210905090000	3.37	3.331	0.039
20210903100000	3.06	3.037	0.023	20210904100000	3.3	3.203	0.097	20210905100000	3.37	3.334	0.036
20210903110000	3.11	3.089	0.021	20210904110000	3.29	3.259	0.031	20210905110000	3.38	3.355	0.025
20210903120000	3.16	3.089	0.071	20210904120000	3.3	3.231	0.069	20210905120000	3.36	3.354	0.006
20210903130000	2.95	2.864	0.086	20210904130000	3.3	3.266	0.034	20210905130000	3.37	3.281	0.089
20210903140000	3.17	3.145	0.025	20210904140000	3.3	3.208	0.092	20210905140000	3.39	3.361	0.029
20210903150000	3.16	3.069	0.091	20210904150000	3.28	3.22	0.06	20210905150000	3.35	3.301	0.049
20210903160000	3.15	3.054	0.096	20210904160000	3.29	3.197	0.093	20210905160000	3.38	3.368	0.012
20210903170000	3.19	3.151	0.039	20210904170000	3.35	3.349	0.001	20210905170000	3.44	3.431	0.009
20210903180000	3.18	3.154	0.026	20210904180000	3.35	3.316	0.034	20210905180000	3.4	3.303	0.097
20210903190000	3.26	3.231	0.029	20210904190000	3.35	3.349	0.001	20210905190000	3.39	3.335	0.055
20210903200000	3.27	3.204	0.066	20210904200000	3.37	3.291	0.079	20210905200000	3.35	3.327	0.023
20210903210000	3.27	3.196	0.074	20210904210000	3.35	3.339	0.011	20210905210000	3.39	3.341	0.049

20210903220000	3.28	3.202	0.078	20210904220000	3.36	3.331	0.029	20210905220000	3	2.981	0.019
20210903230000	3.27	3.211	0.059	20210904230000	3.36	3.296	0.064	20210905230000	3.45	3.382	0.068
평균	3.21125	3.157167	0.054083		3.307917	3.258583	0.049333		3.345	3.300125	0.044875

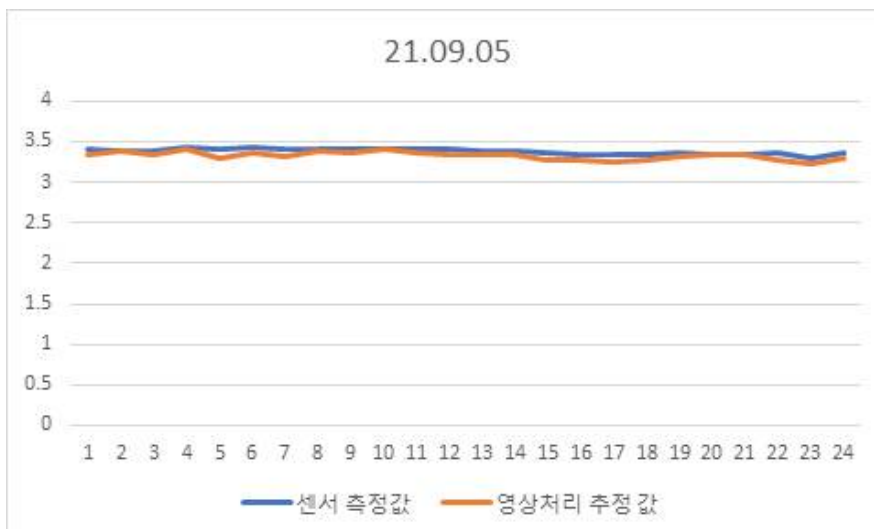
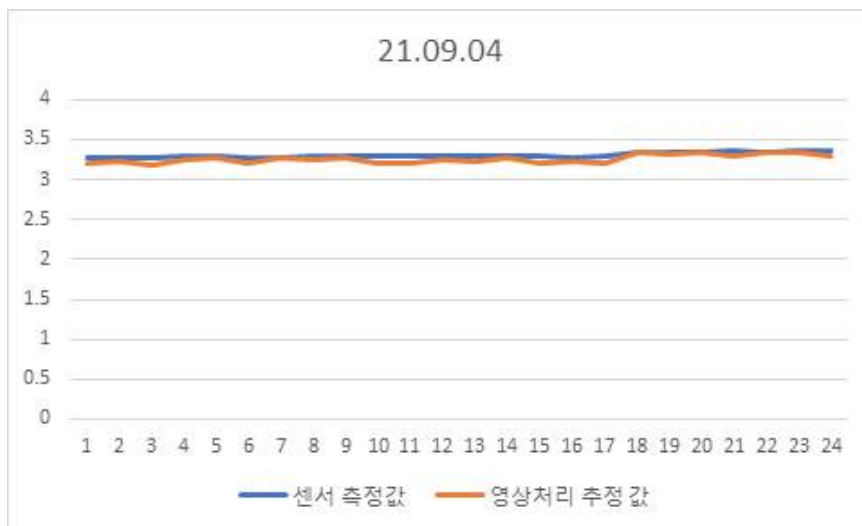
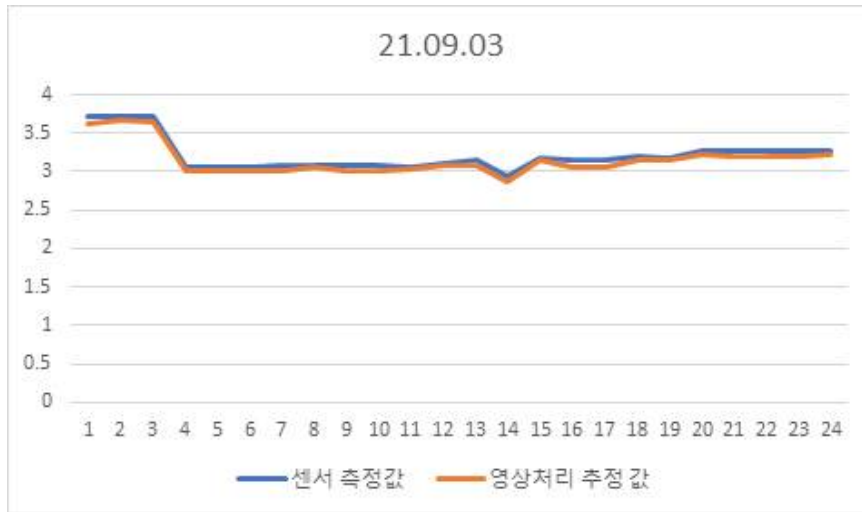


그림 130 저장조 수위에 대한 센서 측정값과 영상처리 추정값 비교(세로축 : 높이(m), 가로축 : 시간(h))

- 영상처리를 이용한 저장조의 수위 추정 알고리즘과 실제 센서를 통해 측정된 데이터의 오차의 평균은 0.049m로 수위센서에서 가축분뇨 발생량의 위/변조 방지를 위하여 실제 수위 센서를 통해 측정된 결과에 대한 레퍼런스 값으로 영상정보를 통해 획득한 수위정보를 통해 검증할 수 있을 것으로 판단된다.

## [첨부 1] index.js의 프로그램 코드

```
varhttp=require('http'); // http 모듈 import
varfs=require('fs'); // file system 모듈 import
constwebsocket=require('./websocket_'); //websocket 모듈 import
const { getData, getTable, getTableCol } =require('./DBCtrl.js'); // DB 관련 모듈 import
consturl_=require('url');
constExcel=require('exceljs');

letindex=0;
letdevpath='./img'
letdevname='cam'
letdataset= []; // DB 에 저장된 데이터 전체를 저장하는 array
letdevnos= []; // sensor index array

letphydata= []; //phy 데이터 array
letorpdata= []; //phy 데이터 array
letconddata= []; //phy 데이터 array
lettempdata= []; //phy 데이터 array
letwaterdata= []; //phy 데이터 array
letnh3data= []; //phy 데이터 array
leth2sdata= []; //phy 데이터 array
letcodata= []; //phy 데이터 array
lettimedata= []; //phy 데이터 array

letcolm= []; // 전체 컬럼 name array
letscolm= []; // 실제 데이터 호출에 사용되는 컬럼 array
letserver=null; //server 저장 변수
letview=0;
letvindex=0;

asyncfunctionExcelTest(_devlist,_collist,_rowlist,res){
//엑셀 워크북 생성 및 시트 생성
constworkbook=newExcel.Workbook();
varfilePath=__dirname+"/sensordata.xlsx"
varfileName="sensordata.xlsx"// 원본파일명
letcolumns= [];
for (iin_devlist){
varworksheet=workbook.addWorksheet("sensor"+_devlist[i]);

//대표행(타이틀행) 설정 및 입력
for (jin_collist){
```

```

columns.push({header:_collist[j].replace(/W/gi, ""), key:_collist[j].
toLowerCase().replace(/W/gi, ""), width:20});
}
worksheet.columns=columns;
for (kin_rowlist[Number(i)]){
//데이터 추가 (행단위 추가)
varts=_rowlist[Number(i)][k][0];
var_phy=_rowlist[Number(i)][k][1];
var_orp=_rowlist[Number(i)][k][2];
var_cond=_rowlist[Number(i)][k][3];
var_temp=_rowlist[Number(i)][k][4];
var_water=_rowlist[Number(i)][k][5];
var_nh3=_rowlist[Number(i)][k][6];
var_h2s=_rowlist[Number(i)][k][7];
var_co=_rowlist[Number(i)][k][8];
worksheet.addRow({msr_ts:ts,phy:_phy,orp:_orp,cond:_cond,temp:_temp,water:_water,nh3:_
nh3,h2s:_h2s,co:_co});
}
//엑셀 데이터 저장
columns= [];
}
res.setHeader("Content-Disposition", "attachment;filename="+encodeURIComponent(fileName));
res.setHeader("Content-Type", "application/vnd.openxmlformats-officedocument.spreadshee
tml.sheet");
awaitworkbook.xlsx.write(res)
.then(function (data) {
res.end();
console.log('File write done.....!');
});
//종료
console.log("complete excel file.");
};

```

```

getTable("testdb", "DEV_INFO").then((rows)=>{ //sensor index와 데이터를 db에서 받아오는 함
수
for(varj=0; j<rows.length;j++ ){
devnos.push(rows[j].DEV_NO); //sensor index 추출
getData(rows[j].DEV_NO).then((rows)=>{//sensor index에 따른 데이터 추출
//console.log(rows);
vard=newDate();
vartemp= [];
for(vari=0;i<rows.length;i++){

```

```

var temper = []
temper.push(new Date(rows[i].MSR_TS - (d.getTimezoneOffset() * 60000)).toISOString().slice(0
, 19).replace('T', '').replace(/-/gi, '').replace(/:/gi, '').replace(/' '/gi, ''));
//UTC 시간을 한국 시간에 맞게 변경
temper.push(rows[i].PHY);
temper.push(rows[i].ORP);
temper.push(rows[i].COND);
temper.push(rows[i].TEMP);
temper.push(rows[i].WATER);
temper.push(rows[i].NH3);
temper.push(rows[i].H2S);
temper.push(rows[i].CO);
temp.push(temper); //한 세트의 데이터를 array에 저장
}
dataset.push(temp); // 특정 센서의 dataset으로 묶어 array에 저장
})
.catch((errMsg) => {
console.log(errMsg);
});
}
})
.catch((errMsg) => {
console.log(errMsg);
});
getTableCol("testdb", "DATA_INFO").then((rows) => { //DATA에 매칭되는 필요한 컬럼 추출

for(var j=0; j<rows.length; j++) {
vartes="" + rows[j].Field + "";
if ((tes=="MSR_TS") || (tes.indexOf('_') === -1)) //graph 생성에 필요한 컬럼
colm.push(tes);
if (tes.indexOf('_') === -1) { //table 생성에 필요한 컬럼
scolm.push(tes);
}
}
})
.catch((errMsg) => {
console.log(errMsg);
});

//websocket(server);

```

```

varapp=http.createServer(function(request,response){//http 프로토콜을 사용하는 서버 생성
// let { url, headers, method } = req;
letdev=0;//센서 인덱스 초기화
letitem=0;//데이터 인덱스 초기화
/*
for(var m = 0;m<devnos.length;m++){
dataset[m]//전체 데이터셋 호출
}
*/
var_url=request.url;// 요청 패킷의 url 저장
var_method=request.method;// 요청 패킷의 메소드 타입 저장
var_headers=request.headers; // 요청 패킷의 헤더 저장

varhtml="";//서비스할 html 텍스트 데이터 초기화

constpath=url_.parse(request.url,true).pathname;// url의 pathname 추출

if (request.method=='GET'){// Method가 GET일 경우
devnos= [];
phydata= [];
orpdata= [];
conddata= [];
tempdata= [];
waterdata= [];
nh3data= [];
h2sdata= [];
codata= [];
timedata= [];
if (path=='/table' || path=='/graph' || path=='/download'){

getTable("testdb","DEV_INFO").then((rows)=>{ //sensor index와 데이터를 db에서 받아오는 함수
}

for(varj=0; j<rows.length;j++ ){
devnos.push(rows[j].DEV_NO); //sensor index 추출
}
for(varq=0; q<devnos.length;q++ ){
getData(devnos[q]).then((rows)=>{//sensor index에 따른 데이터 추출
//console.log(rows);
vard=newDate();
vartemp= [];
for(vari=0;i<rows.length;i++){

```

```

var temper = []
temper.push(new Date(rows[i].MSR_TS - (d.getTimezoneOffset() * 60000)).toISOString().slice(0, 19).replace('T', '').replace(/-/gi, '').replace(/:/gi, '').replace(/' '/gi, ''));
//UTC 시간을 한국 시간에 맞게 변경
temper.push(rows[i].PHY);
temper.push(rows[i].ORP);
temper.push(rows[i].COND);
temper.push(rows[i].TEMP);
temper.push(rows[i].WATER);
temper.push(rows[i].NH3);
temper.push(rows[i].H2S);
temper.push(rows[i].CO);
temp.push(temper); //한 세트의 데이터를 array에 저장
}
dataset.push(temp); // 특정 센서의 dataset으로 묶어 array에 저장
})
.catch((errMsg) => {
console.log(errMsg);
});
}
if (path == '/download') {
ExcelTest(devnos, colm, dataset, response);

}

if (path == '/table') { //table상의 sensor index 가 변경될 경우
var query = url.parse(request.url, true).query; //url의 쿼리 스트링을 변수에 저장
var file = '';
if (query.sensor != undefined) // 쿼리 스트링이 없을 경우, 현재 index로 출력
{
dev = query.sensor; // 쿼리 스트링에서 센서 인덱스 추출
}
for (var i = 0; i < dataset[dev].length; i++) {
//data set에서 출력에 필요한 데이터를 로컬 데이터로 옮겨옴
timedata.push(dataset[dev][i][0]);
phydata.push(dataset[dev][i][1]);
orpdata.push(dataset[dev][i][2]);
conddata.push(dataset[dev][i][3]);
tempdata.push(dataset[dev][i][4]);
waterdata.push(dataset[dev][i][5]);
nh3data.push(dataset[dev][i][6]);
h2sdata.push(dataset[dev][i][7]);
}
}
}

```



```
codata.push(dataset[dev][1][8]);
}
```

//data table의 html 텍스트 생성

```
ghtml=`<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" /> <!--//utf-8 인코딩-->
```

```
<title> Data Table </title> <!--//title name 설정-->
```

```
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script> <!--//플로티 관련 자바스크립트 추가-->
```

```
<!--// CSS-->
```

```
<style>
```

```
table {
```

```
width: 100%;
```

```
border: 1px solid #444444;
```

```
border-collapse: collapse;
```

```
}
```

```
th, td {
```

```
border: 1px solid #444444;
```

```
padding: 5px;
```

```
background:#ACFFEF;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<script>
```

```
////////요소 생성//////////
```

```
let optsrc = [`${devnos}`];
```

```
let itmsrc = [`${colm}`];
```

```
makeTableForm(); // table 폼 생성
```

```
function makeTableForm(){
```

```
var seldiv = add_div(); // select 컨트롤 영역 추가
```

```
var btndiv = add_div();
```

```
var tbldiv = add_div(); // 테이블 컨트롤 영역 추가
```

```
add_select(seldiv,optsrc); // select 컨트롤 추가
```

```
add_btn(btndiv);
```

```
add_table(tbldiv,'table',itmsrc); // table 추가
```

```
}
```

```
function add_btn(obj){
```

```
var parentNode = obj;
```

```
var newForm = document.createElement('form'); // set attribute (form)
```

```

    newForm.name = 'btnForm';
// newForm.method = 'get';
// newForm.action = '/table';
newForm.target = '_self';// 현재 창에서 로드
// create element (input)
var input1 = document.createElement('input');
// set attribute (input)
input1.setAttribute("type", "button");
input1.setAttribute("name", "excel");
input1.setAttribute("value", "Download"); //실제 전달할 센서 인덱스
newForm.appendChild(input1);
// append form (to body)
parentnode.appendChild(newForm);
// submit form
// newForm.submit();// 만들어진 폼을 서버에 전달(쿼리 스트링으로 전달)
newForm.id = 'devbtn' // 폼의 id 설정
newForm.name = 'devbtn' // 폼의 이름 설정

}
function add_div(){// 컨트롤 영역 추가 함수
var body = document.body // html의 body 태그 받아옴
var div = document.createElement('div'); // 태그 영역 추가
//var newtxt = document.createTextNode('머선129');
body.appendChild(div); //child 추가
return div;
}
function add_select(obj,_list){
var parentnode = obj; //부모 노트 저장
var cform = document.createElement('form'); //폼태그 추가
var sel = document.createElement('SELECT'); //선택 컨트롤 태그 추가
var opts = {};// 선택 컨트롤 옵션 초기화
for(var icount = 0;icount<_list.length;icount++)
{
opts[icount]=_list[icount]; //index 추가
}
for(key in opts) { //list내용을 option으로 추가
var opt = document.createElement("OPTION");
opt.value = key;
opt.text = opts[key];
sel.options.add(opt);
}
sel[`${dev}`].selected = true; //selected item
cform.id = 'devitm' // 폼의 id 설정

```

```

sel.id = 'iSelectbox'; // 셀렉트 컨트롤의 id 설정
cform.name = 'devitm' // 폼의 이름 설정
sel.name = 'iSelectbox';// 셀렉트 컨트롤의 이름 설정
cform.appendChild(sel)// 폼양식에 셀렉트 컨트롤 추가
parentnode.appendChild(cform);// 부모노드에 폼 추가
}
function add_table(obj,name,_list){
var parentnode = obj;//부모 노드 저장
obj.id = name; //부모노드의 id 설정
var dtbl = document.createElement('table'); // 테이블 태그 추가
var tbltr = document.createElement('tr');// 테이블 행 추가
for (var j = 0;j<_list.length;j++){
var tblth = document.createElement('th'); //테이블 헤더 태그 추가
var htxt = document.createTextNode(_list[j]); //테이블 헤더 노드 설정
tblth.appendChild(htxt); // text 데이터 추가
tbltr.appendChild(tblth); // 테이블 열에 추가
}
dtbl.appendChild(tbltr);//실제 테이블에 헤더 추가
var tblbody = document.createElement('tbody');// 테이블 body 태그 설정
tblbody.id = "datatable"; // 테이블 body id 설정
add_tablebody(tblbody); //실제 데이터를 tbody 영역에 생성
dtbl.appendChild(tblbody);// table에 tbody 추가
obj.appendChild(dtbl); //부모영역에 테이블 추가
}
function add_tablebody(obj){
var tbody = obj; // 부모 노드 저장
// 로컬에 저장된 array를 html에 추가
var dtime = [${timedata}]
var dphy = [${phydata}]
var dorp = [${orpdata}]
var dcond = [${conddata}]
var dtemp = [${tempdata}]
var dwater = [${waterdata}]
var dnh3 = [${nh3data}]
var dh2s = [${h2sdata}]
var dco = [${codata}]
for (var i = 0; i<dtime.length; i++){
var tritm = document.createElement('tr');// table 행 생성
//table의 열 추가 및 데이터 삽입
var tdttime = document.createElement('td');
var txt1 = document.createTextNode(dtime[i]);
tdttime.appendChild(txt1);
tritm.appendChild(tdttime);
}
}

```

```

var tdphy = document.createElement('td');
var txt2 = document.createTextNode(dphy[i]);
tdphy.appendChild(txt2);
tritm.appendChild(tdphy);
var tdorp = document.createElement('td');
var txt3 = document.createTextNode(dorp[i]);
tdorp.appendChild(txt3);
tritm.appendChild(tdorp);
var tdcond = document.createElement('td');
var txt4 = document.createTextNode(dcond[i]);
tdcond.appendChild(txt4);
tritm.appendChild(tdcond);
var tdtemp = document.createElement('td');
var txt5 = document.createTextNode(dtemp[i]);
tdtemp.appendChild(txt5);
tritm.appendChild(tdtemp);
var tdwater = document.createElement('td');
var txt6 = document.createTextNode(dwater[i]);
tdwater.appendChild(txt6);
tritm.appendChild(tdwater);
var tdhn3 = document.createElement('td');
var txt7 = document.createTextNode(dnh3[i]);
tdhn3.appendChild(txt7);
tritm.appendChild(tdhn3);
var tdh2s = document.createElement('td');
var txt8 = document.createTextNode(dh2s[i]);
tdh2s.appendChild(txt8);
tritm.appendChild(tdh2s);
var tdco = document.createElement('td');
var txt9 = document.createTextNode(dco[i]);
tdco.appendChild(txt9);
tritm.appendChild(tdco);
tbody.appendChild(tritm);
}
}
function deleteTbody(obj){// table의 데이터 삭제
while(obj.rows.length>0){
obj.deleteRow(0);
}
}
if ( window.addEventListener ){
window.addEventListener("load",selectEvent,false); //select 컨트롤이벤트 리스너 추가
} else if ( window.attachEvent ) {

```

```

window.attachEvent("onload",selectEvent); //onload 에 selectEvent 함수 설정
} else {
window.onload = selectEvent; //리스너가 이벤트를 확인 시 selectEvent 함수 실행
}
function selectEvent() {
// 폼이름,선택박스이름 으로 선택박스에 접근
// onchange 이벤트를 적용
document.devitm.iSelectbox.onchange = selectTable;
document.devbtn.onclick = createExcel;
} // this.value 로 이벤트가 발생한 곳,자신(this)의 value값을 출력
function createExcel(){

//document.body.innerHTML = '';
var newForm = document.createElement('form'); // set attribute (form)
newForm.name = 'newForm';
newForm.method = 'get';
newForm.action = '/download';
newForm.target = '';// 현재 창에서 로드
document.body.appendChild(newForm);
// submit form
newForm.submit();
}

function selectTable() {
// value 가 '0000'인 선택하기를 클릭했을때는 얼럿이 뜨지 않도록 예외처리
dno = [`${devnos}`]
// TABLE = document.getElementById('table');
// var tbody = TABLE.getElementsByTagName("tbody")[0];
//sensor 인덱스를 서버에 전달
for (var k = 0;k<dno.length;k++)
if ( this.value == k ){
document.body.innerHTML = '';
var newForm = document.createElement('form'); // set attribute (form)
newForm.name = 'newForm';
newForm.method = 'get';
newForm.action = '/table';
newForm.target = '_self';// 현재 창에서 로드
// create element (input)
var input1 = document.createElement('input');
// set attribute (input)
input1.setAttribute("type", "hidden");
input1.setAttribute("name", "sensor");
input1.setAttribute("value", k); //실제 전달할 센서 인덱스

```

```

// append input (to form)
newForm.appendChild(input1);
// append form (to body)
document.body.appendChild(newForm);
// submit form
newForm.submit();// 만들어진 폼을 서버에 전달(쿼리 스트링으로 전달)
// deleteTbody(tbody)
// break;
}
}
</script>
</body>
</html>`
response.writeHead(200,{'Content-Type':'text/html; charset=utf-8'});
response.end(ghtml);
}

if(path=='/graph'){//graph에서 select 컨트롤이 변경될 경우의 요청
varquery=url_.parse(request.url, true).query;//쿼리 스트링 추출
if (query.sensor!=undefined)//쿼리 스트링이 없을 경우 기존 값으로 출력
{
dev=query.sensor;//센서 인덱스
item=query.item;//데이터 인덱스
}
for (varl=0;l<dataset[dev].length;l++){
//data set에서 출력에 필요한 데이터를 로컬 데이터로 옮겨옴
timedata.push(dataset[dev][l][0]);
phydata.push(dataset[dev][l][1]);
orpdata.push(dataset[dev][l][2]);
conddata.push(dataset[dev][l][3]);
tempdata.push(dataset[dev][l][4]);
waterdata.push(dataset[dev][l][5]);
nh3data.push(dataset[dev][l][6]);
h2sdata.push(dataset[dev][l][7]);
codata.push(dataset[dev][l][8]);
}
//data 그래프의 html 텍스트 생성
ghtml=
`<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title> Data Graph </title>

```

```

<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
</head>
<body>
<script>
////////요소 생성//////////
let optsrc = [`${devnos}`];
let itmsrc = [`${scolm}`];
makeGraphForm();
function makeGraphForm(){
var seldevdiv = add_div();
var selitemdiv = add_div();
var gphdiv = add_div();
gphdiv.id = 'graph';
add_select(seldevdiv,optsrc,'dev',`${dev}`);
add_select(selitemdiv,itmsrc,'item',`${item}`);
add_graph(`${item}`);
}
function add_div(){
var body = document.body
var div = document.createElement('div');
//var newtxt = document.createTextNode('머선129');
body.appendChild(div);
return div;
}
function add_select(obj,_list,name,index){
var parentnode = obj;
var cform = document.createElement('form');
var sel = document.createElement('SELECT');
var opts = {};
for(var icount = 0;icount<_list.length;icount++)
{
opts[icount]=_list[icount];
}
for(key in opts) {
var opt = document.createElement("OPTION");
opt.value = key;
opt.text = opts[key];
sel.options.add(opt);
}
sel[index].selected = true;
cform.id = name
sel.id = name +'Selbox';
cform.name = name

```

```

sel.name = name +'Selbox';
cform.appendChild(sel)
parentnode.appendChild(cform);
}
function add_graph(itemnum){
var dgrp = document.getElementById('graph');
//로컬 데이터를 html에 설정
var dtime = [${timedata}];
var dphy = [${phydata}];
var dorp = [${orpdata}];
var dcond = [${conddata}];
var dtemp = [${tempdata}];
var dwater = [${waterdata}];
var dnh3 = [${nh3data}];
var dh2s = [${h2sdata}];
var dco = [${codata}];
// data 인덱스에 따라서 그래프 생성 및 출력
if ( itemnum == 0 ){
Plotly.newPlot( dgrp, [{
x: [${timedata}],
y: [${phydata}] }],
{ margin: { t: 0 } }
);
} else if ( itemnum == 1){
Plotly.newPlot( dgrp, [{
x: [${timedata}],
y: [${orpdata}] }],
{ margin: { t: 0 } }
);
} else if ( itemnum == 2 ){
Plotly.newPlot( dgrp, [{
x: [${timedata}],
y: [${conddata}] }],
{ margin: { t: 0 } }
);
} else if ( itemnum == 3){
Plotly.newPlot( dgrp, [{
x: [${timedata}],
y: [${tempdata}] }],
{ margin: { t: 0 } }
);
} else if ( itemnum == 4 ){
Plotly.newPlot( dgrp, [{

```



```

x: [${timedata}],
y: [${waterdata}] }],
{ margin: { t: 0 } }
);
} else if ( itemnum == 5 ){
Plotly.newPlot( dgrp, [{
x: [${timedata}],
y: [${nh3data}] }],
{ margin: { t: 0 } }
);
} else if ( itemnum == 6 ){
Plotly.newPlot( dgrp, [{
x: [${timedata}],
y: [${h2sdata}] }],
{ margin: { t: 0 } }
);
} else if ( itemnum == 7 ){
Plotly.newPlot( dgrp, [{
x: [${timedata}],
y: [${codata}] }],
{ margin: { t: 0 } }
);
}
}
if ( window.addEventListener ){
window.addEventListener("load",selectEvent,false);
} else if ( window.attachEvent ) {
window.attachEvent("onload",selectEvent);
} else {
window.onload = selectEvent;
}
function selectEvent() {
// 폼이름,선택박스이름 으로 선택박스에 접근
// onchange 이벤트를 적용
document.dev.devSelbox.onchange = selectTable;
document.item.itemSelbox.onchange = selectGraph;
} // this.value 로 이벤트가 발생한 곳,자신(this)의 value값을 출력
function selectTable() {
// value 가 '0000'인 선택하기를 클릭했을때는 얼럿이 뜨지 않도록 예외처리
dno = [${devnos}]
INDEX = document.getElementById('itemSelbox');
var ind = INDEX.options[INDEX.selectedIndex].value;
for (var k = 0;k<dno.length;k++)

```

```

if ( this.value == k ){
document.body.innerHTML = "";
var newForm = document.createElement('form'); // set attribute (form)
newForm.name = 'newForm';
newForm.method = 'get';
newForm.action = '/graph';
newForm.target = '_self';
// create element (input)
var input1 = document.createElement('input');
// set attribute (input)
input1.setAttribute("type", "hidden");
input1.setAttribute("name", "sensor");
input1.setAttribute("value", k);
// append input (to form)
newForm.appendChild(input1);
var input2 = document.createElement('input');
// set attribute (input)
input2.setAttribute("type", "hidden");
input2.setAttribute("name", "item");
input2.setAttribute("value", ind);
// append input (to form)
newForm.appendChild(input2);
// append form (to body)
document.body.appendChild(newForm);
// submit form
newForm.submit();
// deleteTbody(tbody)
// break;
}
}
function selectGraph() {
// value 가 '0000'인 선택하기를 클릭했을때는 얼럿이 뜨지 않도록 예외처리
ino = [${scolm}];
for (var k = 0;k<ino.length;k++){
if ( this.value == k ){
add_graph(k)
break;
}
}
}
}
</script>
</body>
</html>`

```

```

response.writeHead(200,{'Content-Type':'text/html; charset=utf-8'});
response.end(ghtml);
}

})
.catch((errMsg)=>{
console.log(errMsg);
});
}

```

//로컬에서 사용할 각 데이터 array 의 초기화

```
if(path=='/gps'){//gps 요청시
```

```

if (server!=null)
{
server.close();//웹소켓 서버가 열려 있을 경우 서버 종료
// process.exit();
}

```

```
server=http.createServer(function(request, response) {//웹소켓 서버 재생성
```

```

console.log((newDate()) +' Received request for '+request.url);
response.writeHead(404,{'Content-Type':'text/plain; charset=utf-8'});
response.end();
});
server.listen(38000, function() {//웹소켓 서버를 listen 상태로 변경, 포트는 38000
console.log((newDate()) +' Server is listening on port 38000');
});

```

```
websocket(server);//서버에 웹소켓을 올림
```

```
//gps map의 html 텍스트 생성
```

```

ghtml=`<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> gps tracking </title>
</head>
<body>
<div id='result'></div>
<div id='map' style='width:1200px;height:900px;display:inline-block;'></div>
<script src="//code.jquery.com/jquery-3.3.1.min.js"></script>

```

```

<script type='text/javascript'
src='//dapi.kakao.com/v2/maps/sdk.js?appkey=3d26beb4a7274c9377527d8dbfea1fff'></sc
ript>
<script>
var timeid = null;//소켓 재생성을 위한 타임아웃 컨트롤 변수
var container = document.getElementById('map'); // 카카오 맵 추가
mapOptions = {
center: new kakao.maps.LatLng(37.2951822, 126.9754627),// 맵의 위치 설정
level: 3//맵의 확대 정도
};
var map = new kakao.maps.Map(container, mapOptions);// 실제 카카오맵 생성
var zoomControl = new kakao.maps.ZoomControl(); // 확대 축소 컨트롤 생성
map.addControl(zoomControl, kakao.maps.ControlPosition.LEFT);//위치를 왼쪽으로 하여 컨
트롤 추가
var markers = []; //마커 저장 array
function startWebsocket() {
var ws = new WebSocket('ws://localhost:38000'); //web소켓 생성
ws.onmessage = function(message){
let msg = JSON.parse(message.data);
markers = []
switch(msg.event){
case 'res':// response 데이터 이벤트 확인후 필요한 데이터 get
var msgSplit = msg.data.split(' '); // string 파싱
setMarkers(null);// 현재 마커 삭제
for(var i = 0; i<msgSplit.length-1;i++){
var numSplit = msgSplit[i].split(',') // string을 파싱하여 위도 경도 추출
addMarker(new kakao.maps.LatLng(numSplit[0],numSplit[1]));// 위도 경도로 맵에 마커 출
력
}
break;
default:
}
}
ws.onclose = function(message){
// connection closed, discard old websocket and create a new one in 5s
if(timeid!=null){
// clearInterval(timeid);
clearTimeout(timeid);//타임아웃 함수 제거
}
ws = null
setTimeout(startWebsocket, 5000) //타임아웃 함수로 소켓 재생성
}
// 소켓 접속이 되면 호출되는 함수

```

```

ws.onopen = function(message){
console.log('websocket is opened!!!!');
timeid = setTimeout(sendMessage,1000);
// timeid = setInterval(sendMessage,1000);
};
// 소켓 서버로 부터 메시지가 오면 호출되는 함수.
function sendMessage(){ //서버에 요청 패킷 전송
let sendData = {event: 'req', data: null};
let test = JSON.parse(JSON.stringify(sendData));
ws.send(JSON.stringify(sendData));
//textMessage객체의 값 초기화
}
function addMarker(position){// 위도 경도를 통해서 마커를 생성하는 함수
var marker = new kakao.maps.Marker({
position: position
});
marker.setMap(map);
markers.push(marker);
}
function setMarkers(map) { //맵에 마커 추가
for (var i = 0; i <markers.length; i++) {
markers[i].setMap(map);
}
}
startWebsocket();
</script>
</body>
</html>`
response.writeHead(200,{'Content-Type':'text/html; charset=utf-8'});
response.end(ghtml);
}
if(path=='/img')
{

letstrlist= [];
varquery=url_.parse(request.url, true).query;//url의 쿼리 스트링을 변수에 저장
if (query.device!=undefined)// 쿼리스트링이 없을 경우, 현재 index로 출력
{
view=Number(query.device);// 쿼리 스트링에서 센서 인덱스 추출
}
if (query.file!=undefined)// 쿼리스트링이 없을 경우, 현재 index로 출력
{

```

```

vindex=Number(query.file);
}
varppath=devpath+'/'+devname+String(view+1) +'!';
fs.readdir(ppath, (err, filelist) => { // 하나의 데이터씩 나누어 출력

filelist.forEach(file=> {
console.log(file);
})

for(varz=0;z<filelist.length;z++)
{
tstr=""+filelist[z] +"";
strlist.push(tstr);
}
filename=ppath+filelist[Number(vindex)];

getTable("testdb","DEV_INFO").then((rows)=>{ //sensor index와 데이터를 db에서 받아오는 함수
}

for(varj=0; j<rows.length;j++ ){
devnos.push(rows[j].DEV_NO); //sensor index 추출
}
//이미지 출력의 html 텍스트 생성
ghtml=`<!DOCTYPE html>
<head>
<meta charset="utf-8" />
<title>my test</title>
</head>
<body>
<div>

</div>
<script>
let optsrc = [${strlist}];
let devsrc = [${devnos}];
makeImgForm();
function makeImgForm(){
//var imgdiv = add_div();
var seldevdiv = add_div();
var selimgdiv = add_div();
//imgdiv.id = 'image';
add_select(seldevdiv,devsrc,'dev',${view});
add_multi_select(selimgdiv,optsrc,"image",${vindex});

```

```

}
function add_div(){
var body = document.body
var div = document.createElement('div');
//var newtxt = document.createTextNode('머선129');
body.appendChild(div);
return div;
}
function add_select(obj,_list,name,index){
var parentnode = obj;
var cform = document.createElement('form');
var sel = document.createElement('SELECT');
var opts = {};
for(var icount = 0;icount<_list.length;icount++)
{
opts[icount]=_list[icount];
}
for(key in opts) {
var opt = document.createElement("OPTION");
opt.value = key;
opt.text = opts[key];
sel.options.add(opt);
}
sel[index].selected = true;
cform.id = name
sel.id = name + 'Selbox';
cform.name = name
sel.name = name + 'Selbox';
cform.appendChild(sel)
parentnode.appendChild(cform);
}
function add_multi_select(obj,_list,name,index){
var parentnode = obj;
var cform = document.createElement('form');
var sel = document.createElement('SELECT');
sel.setAttribute('multiple',true);
sel.setAttribute('size',10);
var opts = {};
for(var icount = 0;icount<_list.length;icount++)
{
opts[icount]=_list[icount];
}
for(key in opts) {

```

```

var opt = document.createElement("OPTION");
opt.value = key;
opt.text = opts[key];
sel.options.add(opt);
}
sel[index].selected = true;
cform.id = name
sel.id = name + 'Selbox';
cform.name = name
sel.name = name + 'Selbox';
cform.appendChild(sel)
parentnode.appendChild(cform);
}
if ( window.addEventListener ){
window.addEventListener("load",selectEvent,false);
} else if ( window.attachEvent ) {
window.attachEvent("onload",selectEvent);
} else {
window.onload = selectEvent;
}
function selectEvent() {
// 폼이름,선택박스이름 으로 선택박스에 접근
// onchange 이벤트를 적용
document.image.imageSelbox.onchange = selectImage;
document.dev.devSelbox.onchange = selectDevice;
} // this.value 로 이벤트가 발생한 곳,자신(this)의 value값을 출력
function selectImage() {
// value 가 '0000'인 선택하기를 클릭했을때는 얼럿이 뜨지 않도록 예외처리
ino = [${strlist}]
INDEX = document.getElementById('imageSelbox');
var ind = INDEX.options[INDEX.selectedIndex].value;
for (var k = 0;k<ino.length;k++)
if ( this.value == k ){
document.body.innerHTML = '';
var newForm = document.createElement('form'); // set attribute (form)
newForm.name = 'newForm';
newForm.method = 'get';
newForm.action = '/img';
newForm.target = '_self';
// create element (input)
var input1 = document.createElement('input');
// set attribute (input)
input1.setAttribute("type", "hidden");

```



```

input1.setAttribute("name", "file");
input1.setAttribute("value", INDEX.options[k].value);
// append input (to form)
newForm.appendChild(input1);
// append form (to body)
document.body.appendChild(newForm);
// submit form
newForm.submit();
}
}

function selectDevice() {
// value 가 '0000'인 선택하기를 클릭했을때는 알럿이 뜨지 않도록 예외처리
dno = [`${devnos}`]
INDEX = document.getElementById('devSelbox');
JINDEX = document.getElementById('imageSelbox');
var ind = INDEX.options[INDEX.selectedIndex].value;
for (var k = 0;k<dno.length;k++){
if ( this.value == k ){
document.body.innerHTML = '';
var newForm = document.createElement('form'); // set attribute (form)
newForm.name = 'newForm';
newForm.method = 'get';
newForm.action = '/img';
newForm.target = '_self';

// create element (input)
var input1 = document.createElement('input');
// set attribute (input)
input1.setAttribute("type", "hidden");
input1.setAttribute("name", "device");
input1.setAttribute("value", ind);
// append input (to form)
newForm.appendChild(input1);

var input2 = document.createElement('input');
// set attribute (input)
input2.setAttribute("type", "hidden");
input2.setAttribute("name", "file");
input2.setAttribute("value", JINDEX.options[0].value);
// append input (to form)
newForm.appendChild(input2);
// append form (to body)
document.body.appendChild(newForm);

```

```

// submit form
newForm.submit();
}
}
}
</script>
</body>
</html>`
response.writeHead(200,{'Content-Type':'text/html; charset=utf-8'});
response.end(ghtml)
})
.catch((errMsg)=>{
console.log(errMsg);
});

})
}

if(path=='/image')//패스가 image 일 경우
{

fs.readFile(filename, function(err, data){
console.log('picture loading...');//사진 출력 로그
response.writeHead(200);
response.write(data);//사진 출력
response.end();
});

}

}

if(_url=='/favicon.ico'){// 에러 발생시 출력
response.writeHead(404);
response.end();
return;
}

});

```

```
const port = process.env.PORT || 8080; // 8080 포트 설정
app.listen(port, () => { // 실제 http 서버 상태를 listen으로 변경
  console.log(`listening on ${port}`);
});
```

## [첨부 2] 수거차량용 모바일 GPS 송출 API

```
package com.example.android.python;

import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.util.List;
import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity implements View.OnClickListener,
    LocationListener {
    Button connect_btn; // ip 받아오는 버튼
    Button disconnect_btn;
    EditText id_edit; // ip 에디트
    EditText pw_edit; // ip 에디트

    // 소켓통신에 필요한것

    private LocationManager locationManager; //gps 매니저 선언
    private List<String> listProviders; // string 리스트 선언
    private String html = ""; //html 주소 스트링
    private Handler mHandler; // android 핸들러
```

```

privateSocketsocket;// 네트워크 소켓 선언

privateDataOutputStreamdos;// outputstream 선언
privateDataInputStreamdis;// inputstream 선언

// private String ip = "192.168.0.19"; // IP 번호
privateStringip="59.3.55.93"; // IP 번호

privateintport=5000; // port 번호
publicintsn=29;
publicdoublelng1,lng2,lng3;//위도 데이터
publicdoublelat1,lat2,lat3;//경도 데이터

privatebooleanetimeflag=false;

@Override
protectedvoidonCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
connect_btn = (Button)findViewById(R.id.login_btn); // connect 버튼 컨트롤 확보
disconnect_btn = (Button) findViewById(R.id.signup_btn); // disconnect 버튼 컨트롤 확보
connect_btn.setOnClickListener(this);//버튼 리스너 선언
disconnect_btn.setOnClickListener(this);//버튼 리스너 선언

id_edit = (EditText)findViewById(R.id.id_edit);// 아이디 텍스트 에딧 선언
pw_edit = (EditText)findViewById(R.id.pw_edit);// 비번 텍스트 에딧 선언

lng1 =0.0;//위도 경도 초기화
lng2 =0.0;
lng3 =0.0;
lat1 =0.0;
lat2 =0.0;
lat3 =0.0;

if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION)
!=PackageManager.PERMISSION_GRANTED) {
return;
} //gps 퍼미션 확인
locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);//로케이션
매니저 할당
LocationlastKnownLocation=locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDE
R);// gps 모듈을 통한 데이터 확보
if (lastKnownLocation !=null) {
lng1 =lastKnownLocation.getLongitude();
lat1 =lastKnownLocation.getLatitude();
}
}

```

```

    }
    lastKnownLocation
    =locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);// 네트워크 ap
    모듈을 통한 데이터 확보
    if (lastKnownLocation !=null) {
        lng2 =lastKnownLocation.getLongitude();
        lat2 =lastKnownLocation.getLatitude();

    }

    lastKnownLocation
    =locationManager.getLastKnownLocation(LocationManager.PASSIVE_PROVIDER);// passive 프로바
    이더를 통한 데이터 확보
    if (lastKnownLocation !=null) {
        lng3 =lastKnownLocation.getLongitude();
        lat3 =lastKnownLocation.getLatitude();
    }

    listProviders =locationManager.getAllProviders();// 전체 데이터를 string 에 저장
    boolean [] isEnabled =newboolean[3];
    for(inti=0; i<listProviders.size();i++) {// 이전정보와 같은지 확인후 플래그 설정
    if(listProviders.get(i).equals(LocationManager.GPS_PROVIDER)) {
        isEnabled[0] =locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, this);
    } elseif(listProviders.get(i).equals(LocationManager.NETWORK_PROVIDER)) {
        isEnabled[1] =locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, this);
    } elseif(listProviders.get(i).equals(LocationManager.PASSIVE_PROVIDER)) {
        isEnabled[2] =locationManager.isProviderEnabled(LocationManager.PASSIVE_PROVIDER);

locationManager.requestLocationUpdates(LocationManager.PASSIVE_PROVIDER, 0, 0, this);
    }
    }
    }

@Override
publicvoidonClick(Viewv) {
    switch(v.getId()){
    caseR.id.login_btn:// ip 받아오는 버튼

        timeflag =true: //타임 플래그 설정
        // connect();
        break:

```

```

caseR.id.signup_btn:
    timeflag =false;
break:

}
}

voidconnect(){
    mHandler =newHandler(Looper.myLooper());//스레드 루퍼 핸들러
Log.w("connect","연결 하는중");
// 받아오는 내용
ThreadcheckUpdate=newThread() { // 스레드 선언
publicvoidrun() {
Stringnewip= ip;
try {
    socket =newSocket(newip, port); // 소켓 생성
Log.w("서버 접속됨", "서버 접속됨");
} catch (IOExceptione1) {
Log.w("서버접속못함", "서버접속못함");
e1.printStackTrace();
}

Log.w("edit 넘어가야 할 값 : ","안드로이드에서 서버로 연결요청");

try {
    dos =newDataOutputStream(socket.getOutputStream()); // output에 보낼꺼 넣음
    dis =newDataInputStream(socket.getInputStream()); // input에 받을꺼 넣어짐

Stringtemp="" +Integer.toString(sn)+" "+Double.toString(lat2)+" "+Double.toString(lng2);// 디바이스 번
호와 위도 경도 스트링 저장
dos.writeUTF(temp);// outputstream buffer 에 write

} catch (IOExceptione) {
e.printStackTrace();
Log.w("버퍼", "버퍼생성 잘못됨");
}
Log.w("버퍼","버퍼생성 잘됨");
}

};
// 소켓 접속 시도, 버퍼생성
checkUpdate.setDaemon(true); // 스레드 데몬 설정
checkUpdate.start();// 스레드 시작
}

@Override

```

```

public void onLocationChanged(@NonNull Location location) { // gps 데이터 변경시 이벤트
    double latitude=0.0;
    double longitude=0.0;

    if(location.getProvider().equals(LocationManager.GPS_PROVIDER)) { //gps 모듈을 통한 데이터 변경시
        lat1 =location.getLatitude();
        lng1 =location.getLongitude();

    }

    if(location.getProvider().equals(LocationManager.NETWORK_PROVIDER)) { //gps 모듈을 통한 데이터
    변경시
        lat2 =location.getLatitude();
        lng2 =location.getLongitude();

    }

    if (timeflag ==true) {
    connect(); // 스레드 함수
    }

    }

    if(location.getProvider().equals(LocationManager.PASSIVE_PROVIDER)) { //passive 프로바이더를 통한
    데이터 변경시
        lat3 =location.getLatitude();
        lng3 =location.getLongitude();

    }

    }

    @Override
    public void onProviderEnabled(@NonNull String provider) { //로케이션 프로바이더가 활성화 되었을 경우
    if
    (ActivityCompat.checkSelfPermission(this,
    android.Manifest.permission.ACCESS_FINE_LOCATION)
    !=PackageManager.PERMISSION_GRANTED) {
    return; // 퍼미션 확인
    }
    //gps 데이터 업데이트
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, this);
    locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, this);
    locationManager.requestLocationUpdates(LocationManager.PASSIVE_PROVIDER, 0, 0, this);
    }

    @Override
    public void onProviderDisabled(@NonNull String provider) {
    }
}

```



```

@Override
protected void onStart() { // 최초 액티비티를 실행시 이벤트
    super.onStart();
    if (ActivityCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_COARSE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        // 권한이 없을 경우 최초 권한 요청 또는 사용자에게 의한 재요청 확인
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
            android.Manifest.permission.ACCESS_FINE_LOCATION) &&
            ActivityCompat.shouldShowRequestPermissionRationale(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION)) {
            // 권한 재요청
            ActivityCompat.requestPermissions(this,
                new String[] { android.Manifest.permission.ACCESS_FINE_LOCATION,
                    android.Manifest.permission.ACCESS_COARSE_LOCATION }, 100);
            return;
        } else {
            ActivityCompat.requestPermissions(this,
                new String[] { android.Manifest.permission.ACCESS_FINE_LOCATION,
                    android.Manifest.permission.ACCESS_COARSE_LOCATION }, 100);
            return;
        }
    }
}

```

```

@Override
protected void onPause() { // 최초 액티비티를 멈추시 이벤트
    super.onPause();
    locationManager.removeUpdates(this); // locationManager 제거
}

```

```

@Override
protected void onResume() {
    super.onResume();
    if (ActivityCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        return;
    }

    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, this);
    locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, this);
    locationManager.requestLocationUpdates(LocationManager.PASSIVE_PROVIDER, 0, 0, this);
}

```

}

# <ICT 기반 가축분뇨 발생량 모니터링 제품 개발, 동문이엔티>

## 1. 가축분뇨 발생량 모니터링 센서선정 및 Controller 개발

### 1-1. 가축분뇨 발생량 등 모니터링 센서 현황조사

#### 1-1-1. 가축분뇨 발생량 모니터링

##### (1) 발생량 모니터링 장비

- 가축분뇨 발생량 모니터링을 위한 방법으로 축분 저장조에 수위계를 사용하여 발생량을 모니터링 하였다.

##### (2) 발생량 모니터링 방식 및 장비선정

- 현재 국내외에서 많이 사용하는 수위 측정 방식에는 부자식(float level meter), 압력식 (pressure level meter), 정전용량식(capacitance level meter), 레이더식(radar type level meter), 초음파식(ultrasonic level meter), 기포식(air purge level meter)이 있다.
- 다양한 수위 측정방식 중 온도보상이 필요치 않고 측정 거리에 관계 없이 오차가 일정한 레이더식 수위계를 사용하여 축분 발생량을 모니터링 하였다.

표 1-1 측정 방식별 수위계 비교표

구 분	수위계		
	부자식	압력식	정전 용량식
예 시 도			
특 징	<ul style="list-style-type: none"> <li>• 관측점에서 부자의 수면 접촉을 통해 안정적인 수위 자료의 획득</li> <li>• 1 ~ 30m ±1.0cm</li> </ul>	<ul style="list-style-type: none"> <li>• 수심과 수압이 비례하는 원리로, 수압을 평형상태로 검출하여 그 변위를 증폭하여 전기신호를 변환</li> <li>• 1 ~ 20m</li> </ul>	<ul style="list-style-type: none"> <li>• 1 ~ 수 10m ±1%</li> </ul>
장 점	<ul style="list-style-type: none"> <li>• 온도 및 악천 후 영향없음</li> <li>• 피 측정체의 점도, 밀도의 영향이 없음</li> </ul>	<ul style="list-style-type: none"> <li>• 센서의 크기가 작고, 설치가 간편함</li> <li>• 센서 가격이 저렴함</li> </ul>	<ul style="list-style-type: none"> <li>• 액체, 분체, 입체 모두 측정 가능</li> </ul>
단 점	<ul style="list-style-type: none"> <li>• 설치구조물이 있어야 함</li> <li>• 초기 설치가 어려움</li> </ul>	<ul style="list-style-type: none"> <li>• 개수로를 설치해야 함</li> <li>• 홍수시 유지보수가 어려움</li> <li>• 압력소자의 온도 및 기압 영향 보정 필요</li> </ul>	<ul style="list-style-type: none"> <li>• 전극 부착물 영향</li> <li>• 밀도 영향 받음.</li> </ul>

표 1-2 측정 방식별 수위계 비교표(계속)

구 분	수위계		
	레이더식	초음파식	기포식
예 시 도			
특 징	<ul style="list-style-type: none"> <li>• 마이크로웨이브를 수면에 발사하여 반사파의 도달시간을 수위로 변환</li> <li>• 1~ 30m ±0.25%</li> </ul>	<ul style="list-style-type: none"> <li>• 초음파를 수면에 발사하여 반사된 초음파의 도달시간을 수위로 변환</li> <li>• 0.35 ~ 15m</li> </ul>	<ul style="list-style-type: none"> <li>• 수심과 수압이 비례하는 원리로, 수압을 평형상태로 검출하여 기포를 발생하는 기계적 Torque 출력을 전기적으로 변환</li> <li>• 1~ 13m</li> </ul>
장 점	<ul style="list-style-type: none"> <li>• 비접촉식으로 설치가 용이</li> <li>• 높은 정밀도</li> <li>• 표면유속 측정가능</li> <li>• 간단한 교정</li> </ul>	<ul style="list-style-type: none"> <li>• 비접촉식으로 설치가 용이</li> <li>• 유지관리가 간편</li> </ul>	<ul style="list-style-type: none"> <li>• 경사 또는 곡선으로 설치가능</li> <li>• 동결에 우려가 있는 수위측정 시 적합</li> </ul>
단 점	<ul style="list-style-type: none"> <li>• 주변 지형이나 구조물에 간섭받을 수 있음.</li> </ul>	<ul style="list-style-type: none"> <li>• 초음파의 전파속도가 기온, 풍속 등에 영향을 받음</li> </ul>	<ul style="list-style-type: none"> <li>• 수위측정시 압축공기로 측정관을 불어내기 때문에 질소탱크나 공기펌프가 필요</li> <li>• 기포발생 압축공기를 주기적으로 점검해 주어야 함</li> </ul>

1-1-2. 가축분뇨 환경모니터링

(1) 환경 모니터링 장비

- 가축분뇨 환경 모니터링 장비로 현재 수질측정용 자동모니터링장비 중 가축분뇨의 열악한 환경에 견딜 수 있는 내구성 있는 장비를 선정하여 모니터링 하였다.

(2) 환경 모니터링 방식 및 장비선정

- 장비의 선정은 현장여건, 유지관리 용이성, 장비 가격 등을 검토하여 선정하였다.

○ YSI EX01(YSI, U.S.A.)

- 수질자동모니터링에 전세계적으로 많이 사용되고 있는 장비로 내구성, 정확도 등이 뛰어나지만, 축산분뇨 모니터링에 적용하기에는 가격이 고가라는 단점이 있다.

○ Troll AT600(In-Situ, U.S.A.)

- 수질자동모니터링에 전세계적으로 많이 사용되고 있는 장비로 내구성, 정확도 등이 어느정도 검증되었으나 가격이 비교적 고가이다.
- M100(기술과 환경, Korea)
  - 수질자동모니터링에 국내에서 제작된 장비이지만 내구성, 정확도에 대한 검증이 필요하며 가격이 비교적 고가이다.
- ASPRT2151D(Nengshi, China)
  - 가격이 저렴하여 농가 보급형으로 활용 가능하다는 판단으로 실험실 내에서 내구성, 정확도에 대한 검증 후 1차년도에 설치 하였으나, 현장 설치시 열악한 환경으로 인하여 내구성이 급격히 떨어져서(2주 ~ 1달) 다른 제품으로 대체를 시켰다.
- C4E, pH-ORP 센서(Aqualabo, France)
  - 축분 환경모니터링에 사용되었던 실적이 있으며(액비 모니터링), 내구성도 검증된 장비로서 중저가형으로 농가 보급형으로 적용 가능한 것으로 판단된다.
  - 일체형이 아닌 센서방식이지만 센서 거치대를 제작하여 일체형과 같이 활용 가능토록 하였다.

## 1-2. 가축분뇨 발생량 등 모니터링 측정항목 검토

### 1-2-1. 모니터링 측정항목 검토 개요

- 당초 수위, pH, ORP, 온도 4항목에 대한 모니터링 계획이었으나 추가적으로 간단히 돈분의 성상 및 화학적 변화를 나타내는 항목이 있는지 검토 하였다.
- 검토 결과 EC에 대한 추가 모니터링이 필요할 것으로 검토되었다.

### 1-2-2. 모니터링 측정항목별 특징

#### (1) 수위(Level)

- 돈분의 발생량을 모니터링하기 위한 필수 인자이다.

#### (2) pH & ORP

- 호기 및 무산소 zone에서 ORP 값과 질소의 상관관계가 매우 높아 질소원의 간접지시계로서 사용가능하다.
- 가축분뇨 처리에 적용하여 가축분뇨 공동 자원화를 위해 사용될 항목으로 pH와 ORP를 측정하여 부숙도 판정 후 고품질 액비 생산에 활용가능하다.
- 질소가 암모니아로 전환될 때 질소성분은 산을 중화시키기 때문에 pH 값을 통해 간접지시계로 사용 가능하다.
- pH 7 부근으로 유지되어야 하며 이는 메탄생선세균의 최적 생장 pH이다.
- ORP -300mv 이하로 유지되어야 하며 이는 유기탄소의 메탄환원 조건이다.

#### (3) EC(Electrical Conductivity)

- 퇴비화 과정중 EC의 변화는 유기물이 분해되면서 발생하는 이온들로 인한 것으로, 전기전

도도는 calculated ionic strength 와 매우 높은 상관관계가 있다.

- 전기전도도는 slurry(액비) 속의 비료성분과 밀접한 상관관계를 가지고 있다.

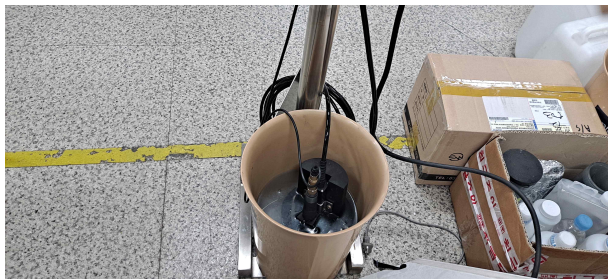
#### (4) 온도(Temperature)

- 양질의 가축분뇨 퇴비화를 위해 최소 60℃이상으로 고르게 분포되어야 한다.
- 퇴비화가 진행되는 동안 퇴비더미의 온도변화 및 분포 양상이 고르지 확인해야 한다.
- 퇴비화 속도와 온도 변화의 상관관계가 있기 때문이다.
- 고온이 나타나는 이유는 유기물질의 산화와 더불어 미생물로부터 기원된 산화효소 작용에 의한 발열반응이 진행되는 결과로서 기질중의 수분증발을 촉진시키고 병원성 미생물을 살균하고 잡초씨앗을 불활화 시킨다.

### 1-3. 센서 내구성 test 및 최적 센서 선정

#### 1-3-1. 1차년도 센서 선정

- 농가 보급형으로 활용 가능한 센서 선정을 위하여 1차년도에 ASPRT2151D(Nengshi, 중국) 장비를 활용하여 실험실에서 정확도, 내구성 등에 대한 test를 실시 하였다.
- 기간 : 2020. 6 ~ 2020. 9 (약 3개월)



장비 test



제작장비(센서 + wiper)

그림 1-1. 센서 test

- 실험실 test에서는 사용 가능할 것으로 판단되어 현장적용 결과 축분의 열악한 환경으로 인하여 내구성이 현저하게 떨어지는 것으로 확인 되었다.
- ASPRT2151D(Nengshi, China) 장비 내구성 기간 : 2주 ~ 1달

#### 1-3-2. 2차년도 센서 선정

- 1차년도 현장적용 장비의 내구성 문제로 인하여 2차년도에는 기존의 모니터링 사업에서 내구성 및 정확도가 검증된 장비(Aqualabo, France)를 사용하여 적용성 시험을 실시한 결과 최초 설치(2021.10.07.) 이후 현재까지 약 3개월간 안정적으로 측정이 되고 있는 것으로 확인 되었다.



기존 장비(Nengshi, China)



교체장비(Aqualabo, France)

그림 1-2. 환경 모니터링 장비 교체

#### 1-4. 센서 controller 개발/제작

##### 1-4-1. 센서 Controller 개발

- 선정된 센서(발생량 및 환경측정)의 데이터를 저장 및 출력하는 역할과 더불어 서버의 관제 프로그램과 통신할 수 있도록 모뎀 제어 역할을 수행한다.
- 이를 통하여 사용자는 수월하게 측정 데이터를 수집 및 분석할 수 있으며 시스템에 문제가 발생할 경우 현장에서 확인하여 센서 및 기타 장비의 점검이 가능하다.
- 사용자 편의를 도모하기 위하여 touch screen 방식 채택하였다.
- 또한 controller 개발시 현장의 효율적·경제적 관리를 위하여 악취성분(NH<sub>3</sub>, H<sub>2</sub>S, CO) 측정 값도 통합하여 관리 가능토록 설계/제작하여 현장적용시험을 실시 하였다.



그림 1-3. 센서 Controller 개발/제작

표 1-3 센서 controller 제원

구 분	사 양
모 델	Eco::watch SD-100
C P U	180MHz, 32bit
R A M	512Mb Flash memory
저장용량	8G Micro SD
Ethernet	TCP/IP
RS232C(RS485)	2ch
Relay(Digital Output)	10ch
USB	1ch(데이터 다운)
화면 출력	4"Full touch screen, 정전터치
외부 재질	알루미늄
취부 방식	벽부형 설치
전 원	DC 12V, 50mA(최대 180mA)
규 격	160(W) × 90(D) × 160(H)
소비전력	1W 미만(최소)/2W(최대)



## 2. 가축분뇨 발생량 측정정보 전송 펌웨어 개발

### 2-1. 가축분뇨 발생량 측정정보 전송 펌웨어 개발

- 현장의 각 센서에서 측정한 항목을 취합하여 시간데이터 작성 및 서버에서 데이터 요청시 응답하여 현장에서 저장된 데이터를 확인 및 저장할 수 있도록 설계
- 측정항목은 pH, ORP, EC, 온도, NH3, H2S, CO2, 수위 총 8개로 구성
- 서버와의 통신은 TCP/IP 통신

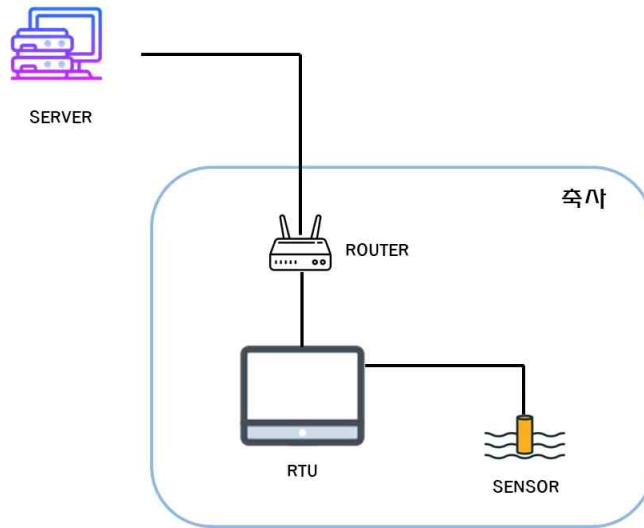


그림 2-1. 장비 기본 구성도

#### (1) 센서 구성

- 현장 측정용 센서는 총 5개로 구성

표 2-1 모니터링 센서 구성

구 분	측정항목	통신방식	비 고
센서1	pH, ORP	RS485	
센서2	온도, EC	RS485	
센서3	NH3	RS485	
센서4	H2S, CO2	RS485	
센서5	수위	4~20mA	

#### (2) RTU-센서간 통신

- 센서1 ~ 센서4는 해당 센서들이 요구하는 RS485 통신으로 설계
- 수위센서는 4~20mA 신호를 RTU에서 수신하여 수위를 선정

#### (3) RTU-서버간 통신

- RTU에 연결된 라우터의 고정 IP를 이용하여 서버간 TCP/IP 통신으로 설계
- RTU-서버간 통신은 TMS 프로토콜을 사용
- RTU-서버간 접속은 서버에서 원하는 시간에 장비에 접속하여 데이터를 송수신

### 3. 가축분뇨 저장조 실시간 모니터링 시스템 개발/설치

#### 3-1. 모니터링 및 전송 시스템 개발

##### 3-1-1. 메인화면

- 메인화면에서는 기본적으로 현재 측정 데이터가 출력된다.
- 한 화면에는 최대 2개의 센서 데이터를 보여주며 3개 이상의 센서가 연결되었을 EO 페이지 이동 기능을 사용하여(화면의 LR버튼) 이동하도록 설계 하였다.

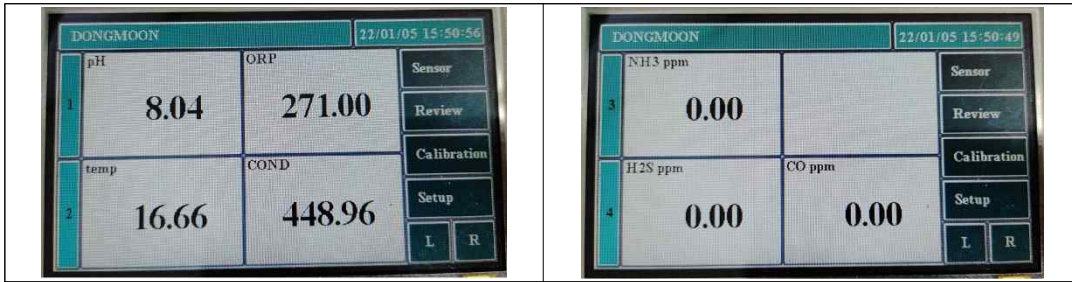


그림 3-1. 메인화면 구성 및 데이터 확인을 위한 페이지 변경

##### 3-1-2. 리뷰화면

- 저장된 1시간 단위의 데이터를 기간을 검색하여 확인 가능하도록 설계 되었다.
- 한 화면에 8개의 데이터가 확인 가능하며 UP, DOWN 버튼으로 이전, 이후 데이터를 확인 가능하도록 구성하였다.

Review				EXIT	SAVE	UP	
start time	21	11	05	00	00	LOAD	DN
end time			05	15	51		
DATE	pH	ORP	COND	TEMP	NH3		
21.11.05 00:00	7.48	-449.94	27688	22.88	14.09		
21.11.05 01:00	7.49	-450.00	27762	22.85	10.44		
21.11.05 02:00	7.51	-451.00	27809	22.86	9.04		
21.11.05 03:00	7.52	-451.95	27788	22.87	8.17		
21.11.05 04:00	7.53	-452.94	27790	22.89	7.58		
21.11.05 05:00	7.54	-453.00	27778	22.89	7.03		
21.11.05 06:00	7.56	-453.99	27811	22.90	5.77		
21.11.05 07:00	7.57	-454.95	27803	22.88	5.99		

그림 3-2. 데이터 검색 기능

- 검색한 데이터를 USB에 저장하는 기능을 구현하였다.(excel csv 파일)

	A	B	C	D	E	F	G	H	I	J
99	2021-11-05	0:00	7.482	-449.94	27688.3	22.88	2.07	14.087	0	0
100	2021-11-05	1:00	7.493	-450	27761.91	22.85	2.07	10.441	0	0
101	2021-11-05	2:00	7.506	-451	27809.42	22.86	2.07	9.037	0	0
102	2021-11-05	3:00	7.52	-451.95	27787.95	22.87	2.09	8.173	0	0
103	2021-11-05	4:00	7.535	-452.937	27790.42	22.89	2.09	7.584	0	0
104	2021-11-05	5:00	7.545	-453	27778.4	22.89	2.09	7.027	0	0
105	2021-11-05	6:00	7.559	-453.987	27811.22	22.9	2.09	5.774	0	0
106	2021-11-05	7:00	7.571	-454.947	27803.36	22.88	2.09	5.993	0	0
107	2021-11-05	8:00	7.582	-455	27830.75	22.87	2.09	5.558	0	0
108	2021-11-05	9:00	7.586	-455.911	27880.66	22.91	2.06	4.333	0	0
109	2021-11-05	10:00	7.601	-456	27863.22	22.89	2.05	3.541	0	0
110	2021-11-05	11:00	7.608	-456.954	27843.53	22.87	2.06	1.564	0	0
111	2021-11-05	12:00	7.604	-456	27926.77	22.82	2.06	48.108	0	0
112	2021-11-05	13:00	7.599	-456	28082.77	22.85	2.08	23.34	0	0
113	2021-11-05	14:00	7.601	-456	27961.53	22.98	2.1	76.649	0	0
114	2021-11-05	15:00	7.599	-456	28274.38	22.9	2.13	57.643	0	0

그림 3-3. USB에 저장된 데이터 파일

### 3-1-3. 셋업화면

- 시간, LAN, 현장코드, 수위, 타이머, 세정장치 총 6개 설정기능으로 구성되었다.
- DATE는 현재 장비의 시간을 나타낸다.
- LAN은 장비의 IP와 PORT 설정이 가능하다.
- CODE
  - RTU-서버간 구성되어 있는 TMS 프로토콜에는 수계코드와 측정소코드가 존재한다.
  - 현장에서 위치를 달리한 동일 장비가 있을 경우에는 서버에서 이 코드를 이용하여 데이터를 분류할 수 있다.
  - 해당 코드를 장비에서 설정할 수 있다.
- WATER
  - 수위센서는 센서의 설정 또는 특성에 따라 4mA는 0m를 표현하지만 20mA는 5m, 10m 등 다를 수 있다.
  - 해당기능은 센서의 특성에 맞게 계산식을 설정하여 수위센서에서 보내주는 데이터와 RTU가 표출하는 데이터가 일치할 수 있도록 설정해 주는 기능이다.
- TIMER : 타이머 설정 기능
- CLEAN
  - 환경측정 장비의 세척을 위한 기능이다.
  - 세척 장비는 현장의 조작버튼 또는 RTU의 릴레이를 사용하여 동작 가능하다.
  - RTU는 릴레이 6개를 사용하여 세척장치의 상/하/좌/우 이동, 세정, 물배출 동작을 관리하는 기능이다.
  - 해당 기능은 위의 각 동작을 구분하여 테스트하거나 연속된 세정동작을 테스트 할 수 있도록 되어 있다.



그림 3-4. 장비 셋업 화면

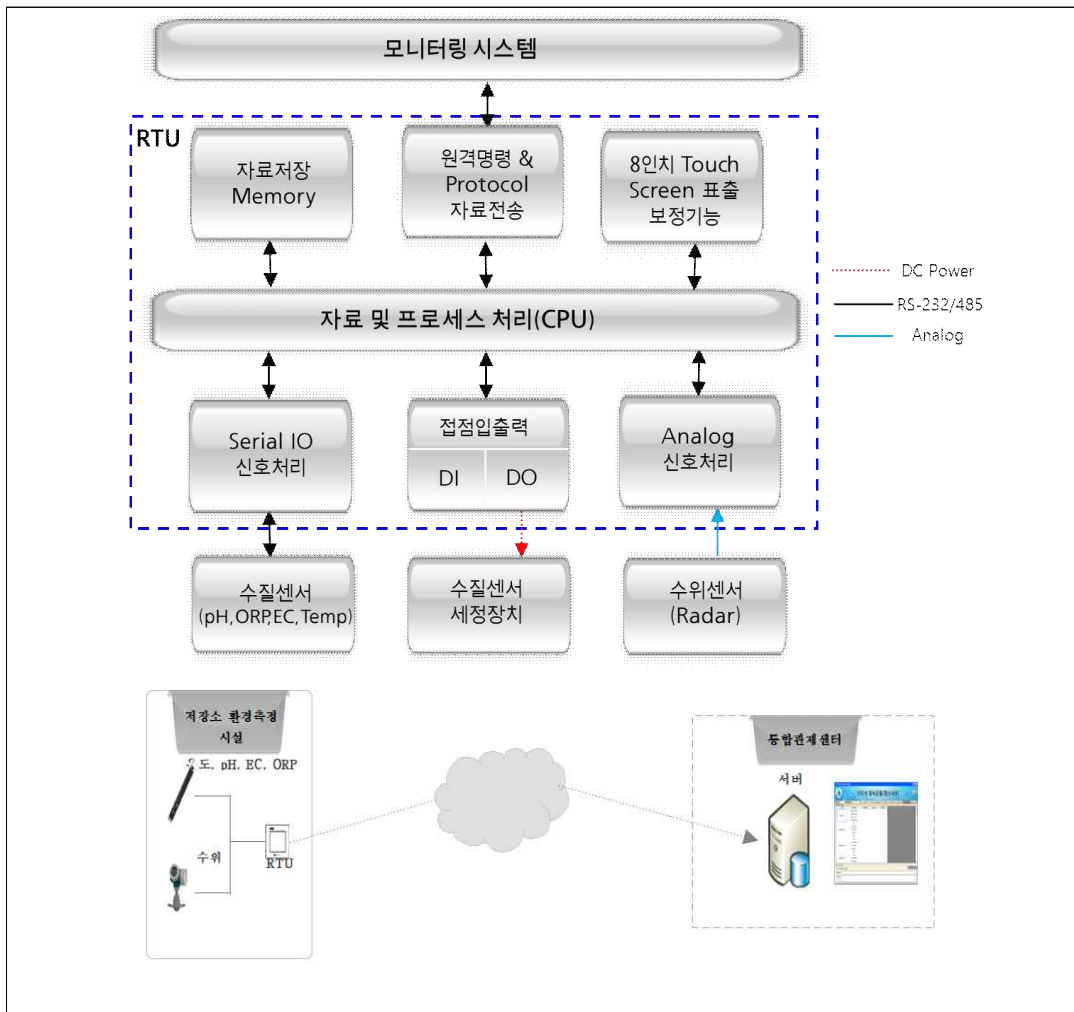


그림 3-5. 센서 Controller 자료처리 모식도

### 3-2. 모니터링 및 전송 시스템 설치

- 센서(발생량 및 환경), 센서 컨트롤러를 통합하여 모니터링 시스템을 개발 하였으며 익산시 청, 전북대학교와 협의 후 테스트베드로 선정된 익산시 소재의 2농장에 개발된 모니터링 시스템을 설치하였다.
- 설치 이전에 과제 수행기관인 전북대학교, 강원대학교와 함께 현장 답사를 실시하였으며 설

치위치, 주변 환경에 대한 조사를 실시하였다.(2020. 7.)

- 설치 후 낭상면 소재의 현대화 농장은 현재까지 지속적으로 모니터링 중이나, 왕궁면 소재의 재래식 농장은 2021. 9월 축산시설을 철거로 인하여 모니터링 시스템을 철거한 상태이다.

### 3-1-1. 모니터링 대상농장

#### (1) A농장

- 익산시 낭상면 소재(현대화 농장)
- 사육돈수 약2,200두(모돈 190두, 육성돈 2,010두)

#### (2) B농장

- 익산시 왕궁면 소재(재래식 농장)
- 육성돈 1,500두

### 3-1-2. 모니터링 시스템 설치

- 시스템 설치를 위하여 해당 농가들과 사전 협의를 완료하고 2020. 11월 모니터링 시스템을 설치하고, 전북대학교 개발 시스템과의 프로토콜 일치작업 등을 수행한 후 2021. 4월 모니터링을 시작 하였다.



설치 전



센서 거치대 설치



수위계 설치



설치 완료

그림 3-6. 모니터링 시스템 설치(A농장)



설치 전



센서 거치대 및 수위계 설치



모니터링시스템 설치 중



설치 완료

그림 3-7. 모니터링 시스템 설치(B농장)

## 4. 현장용 단말장치 시스템 개발

### 4-1. 현장용 단말장치(Penal PC)



그림 4-1. 현장용 단말장치 메인화면

#### 4-1-1. 주요기능

- 10인치 터치 화면에 최적화된 제어 UI 구현
- 리눅스, 윈도우 운영체제 동작 가능한 크로스 플랫폼 구현
- 통신 장비 관리, 개별 항목관리, 측정그룹 관리로 사용자의 요구사항에 맞게 측정항목추가 및 화면구성 변경이 가능하도록 설계
- 통신로그 보기를 통한 시스템 진단 기능 탑재
- 원격 접속/제어를 통한 원격 진단 가능 하도록 시스템 설계

#### 4-1-2. 측정화면

- 화면 그룹화 기능으로 표출하고자 하는 그룹 추가/삭제
- 지정된 항목 클릭으로 이력 그래프 조회
- 지정된 항목 클릭으로 이름/단위 수정
- 각 항목별 동작상태 실시간 확인

#### 4-1-3. 자료 조회

- 전 항목 조회/ 그룹별 선택 조회
- 시작시간 끝시간 구간지정 후 자료 조회

#### 4-1-4. 그래프

- 측정항목 추가/삭제로 항목별 경향 비교
- 2개의 화면 분할로 그룹별 경향 비교
- 측정화면에서 항목 데이터 클릭 시 그래프 자동 전환 기능

#### 4-1-5. 항목관리

- 입력신호 지정을 통한 항목 추가
- 항목이름/단위/측정주기 사용자 수정 가능
- 항목사용/사용안함 기능

- 측정화면에서 항목 이름 클릭하여 측정그룹 지정 기능

#### 4-1-6. 설정

- 통신 네트워크 설정
- 장비이름 설정
- 장비 측정그룹 추가/삭제
- 통신로그 보기
- 미수신 데이터 수동 덤프 명령
- 장비 시간 동기화 명령

### 4-2. 현장용 단말장치 설비사양

#### 4-2-1. 패널PC



<b>Processor</b>	<b>BCM2837 1.2GHz</b>
<b>Ram</b>	2 Gb
<b>Network</b>	100Mbit Ethernet
<b>Power</b>	12V 2A
<b>Display</b>	10 inch
<b>Storage</b>	16 Gb
<b>Interface</b>	정전식 터치

그림 4-2. 현장용 단말장치 사양

#### 4-2-2. LTE Router



<b>Processor</b>	<b>580Mhz</b>
<b>Ram</b>	16Mb
<b>LTE Speed</b>	150Mbps
<b>OS</b>	Linux
<b>Power</b>	5V 2A
<b>Environment</b>	-30°C~ 70°C

그림 4-3. LTE Router

### 4-3. 현장용 단말장치 통신 네트워크 구성

#### 4-2-1. 통신 네트워크 구성

- LTE망을 사용하여 통신 신뢰성 보장
- 4G 통신망을 사용하기 때문에 데이터 사용의 부담을 최소화 하고자 WIFI를 통한 통신 접속도 가능함
- 장소에 구애받지 않고 설치 가능토록 망 설계



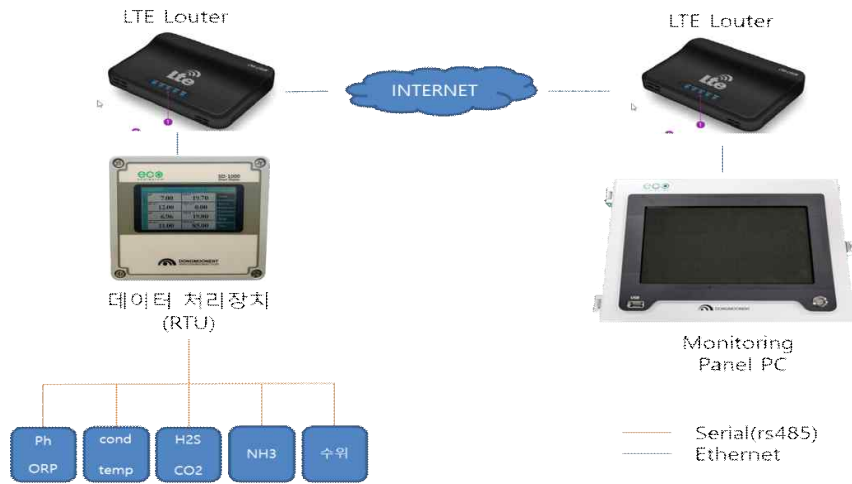


그림 4-4. LTE 네트워크 구성도

#### 4-4. 현장용 단말장치 데이터 처리

##### 4-4-1. RTU 실시간 데이터 저장

- 초당 실시간 데이터를 취합하여 원격지 RTU에서 1차 저장

##### 4-4-2. Penal PC 평균데이터 저장

- RTU에 저장된 구간 데이터를 자동 또는 수동으로 요청해 2차 저장
- 통신이 불안정할 경우 데이터 수동덤프가 가능토록 설계



그림 4-5. 데이터 처리 흐름도

## 5. 가축분뇨 발생량 등 모니터링 센서 통합시스템 현장적용시험

### 5-1. 현장적용시험 목적

- 가축분뇨 저장 현장의 열악한 환경에서도 내구성을 유지하고, 농가 보급형으로 사용 가능한 최적의 센서를 찾아내고 통합시스템(센서, Controller)의 현장 적용성을 시험하였다.

### 5-2. 현장적용시험 내용 및 범위

통합시스템 구성	테스트 베드 지점 선정 후 TEST	현장 적용
<ul style="list-style-type: none"> <li>• 시스템 구성 자재선정 및 수급(센서,RTU,부대시설 등)</li> <li>• RTU 프로그램 개발</li> <li>• 개발실 내 TEST</li> </ul>	<ul style="list-style-type: none"> <li>• 저장조내 지점 선정</li> <li>• 센서가대 및 부대시설 설치</li> </ul>	<ul style="list-style-type: none"> <li>• 현장 적용하여 성능평가</li> </ul>

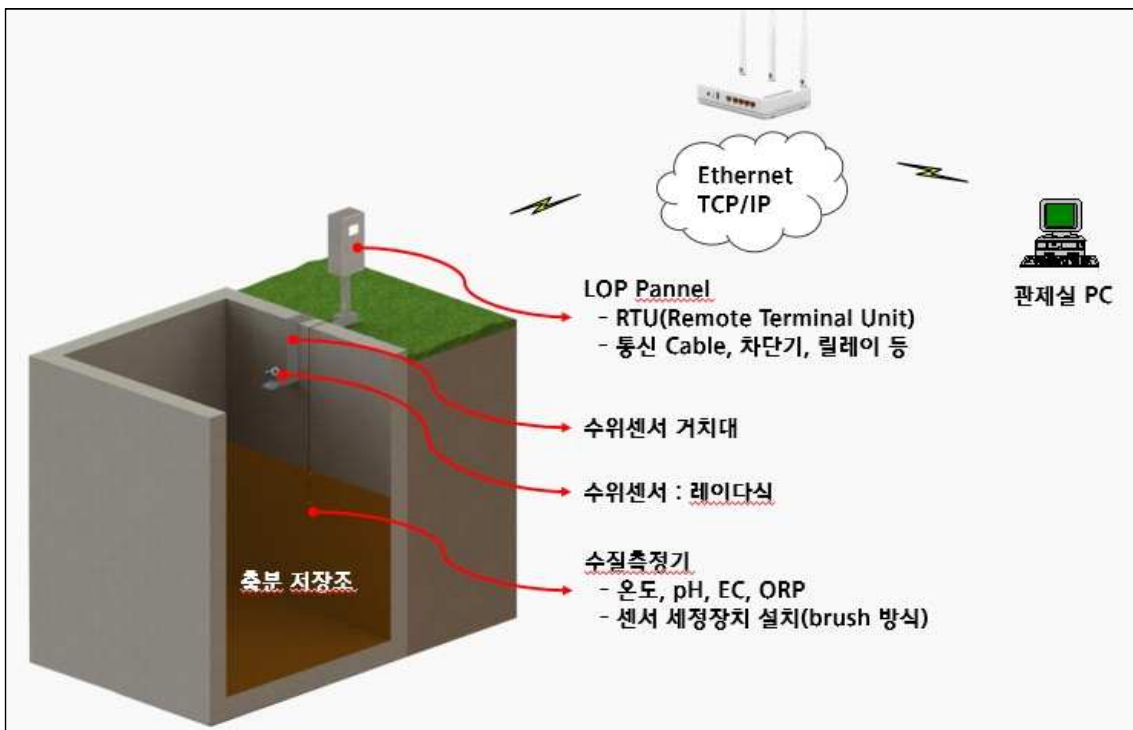


그림 5-1. 통합모니터링 시스템 설치 모식도

- 축분 저장조 상부에 발생량 모니터링을 위한 수위계 설치(레이더식)
- 축분 수표면 약 1m 하부에 센서 거치대를 활용한 환경 모니터링 센서 설치
- 세정장치는 당초 Air jet 방식을 고려하였으나, 축분의 점성으로 인하여 효과가 없어 brush 방식으로 변경
- LOP pannel 내부에 통합센서 controller(RTU) 및 무선모뎀(LTE)을 설치하여 관제실로 자료 전송
- RTU는 측정 데이터를 자동으로 저장하며, USB로 자료취득 가능

### 5-3. 현장적용시험 추진내역

#### ○ 센서 현장설치

- 장소 : 전북 익산시 소재 테스트베드 2개농장
- 일시 : 2021년 3월 9일 ~ 3월 11일
- 내용 : 환경센서(온도, pH, ORP, EC) 및 수위센서 설치

#### ○ 서버와 데이터 송수신 시작

- 일시 : 2021년 4월 6일 ~ 4월 8일
- 내용 : 악취센서(NH<sub>4</sub>, H<sub>2</sub>S, CO) 추가설치 및 송수신 확인(해당센서 프로토콜 구현 및 RTU 업데이트)  
센서 세정장치 설치(세정장치에 주기적으로 접점신호를 보내도록 기능 추가)  
서버와 RTU간 통신 프로토콜을 구현하여 장비 업데이트 후 상호간 데이터 송수신 확인

#### ○ 장비 점검

- 일시 : 2021년 5월 3일 ~ 5월 4일
- 내용 : 악취센서(NH<sub>4</sub>, H<sub>2</sub>, CO)의 정상가동은 확인하였으나 필터(수분제거)수명으로 인하여 정상데이터 측정 불가  
환경센서는 내구성이 떨어져 작동 불가

#### ○ 센서교체, 덤프데이터 송수신 기능 개선 등

- 일시 : 2021년 6월 9일 ~ 6월 11일
- 내용 : 환경센서 교체  
현장 RTU는 임베디드 장비의 스펙으로 한달 이상의 데이터를 한번에 읽어오는데 어려움이 있음

#### ○ 악취센서 측정방식 변경에 따른 RTU 업데이트

- 일시 : 2021년 7월 20일 ~ 7월 22일
- 내용 : 악취센서를 상시적으로 측사에 배치하여 측정시 센서 필터의 소모가 과도함을 확인  
전처리 장비를 추가하여 평소에는 차단하여 센서 필터의 소모를 최소화  
타이머를 설치하여 타이머 동작시 전처리 장비를 개방하는 접점신호를 보내도록 업데이트  
장비는 접점신호를 받으면 일정시간 대기 후 측정을 시작하도록 기능 업데이트

#### ○ 환경센서 교체 및 RTU 업데이트

- 일시 : 2021년 10월 6일 ~ 10월 8일
- 내용 : 환경센서의 내구성 결여로 ASPRT2151D(Nengshi, China)에서 내구성이 검증된 개별센서(Aqualabo, France)로 교체  
센서에 가대를 설치하여 일체형 센서와 같이 사용 가능토록 제작  
교체된 센서의 프로토콜로 RTU 업데이트  
환경센서도 측정시에만 전원을 켜도록 RTU 기능 업데이트

○ 세정방식 변경에 따른 장비 업데이트

- 일시 : 2021년 12월 29일 ~ 12월 30일
- 내용 : 환경센서의 내구성 향상을 위하여 세정방식을 측정시에만 저장조로 이동하고 대기 시간에는 수조에서 세척시키는 방식으로 변경  
세척장치는 현장의 조작버튼 또는 RTU의 릴레이를 사용하여 동작  
세척장치는 각 모터의 동작모드(수동/자동) 상태를 점접신호로 RTU에 전달  
RTU는 릴레이 6개를 사용하여 상/하/좌/우, 세정, 물배출 동작을 관리  
설정된 시간에 따라 장비가 세정동작을 하도록 RTU 업데이트

5-4. 현장적용시험 결과

- 통합 모니터링 시스템 설치 후 2021년 4월 관제서버와의 프로토콜 일치작업을 완료하고 자동 모니터링을 시작하였다.
- 축분발생량(수위센서)은 2021년 4월 설치 후 지속적으로 자동계측이 진행되었다.
- 그러나 환경모니터링의 경우 1차 선정된 저가형의 환경센서 ASPRT2151D(Nengshi, China)는 축분 저장조에 침적형으로 설치함으로 인하여 축분의 열악한 환경에 견디지 못하고 자주 내구성이 결여되는 상황이 발생하였다.
- 기존 축분저장조 모니터링에서 내구성이 검증된 개별센서(Aqualabo, France)를 2021년 10월에 설치한 이후에는 지속적으로 모니터링이 가능하였다.
- 또한 환경센서의 내구성 증대를 위하여 기존의 침적형에서 측정시에만 축분 저장조에서 측정하고(10분) 대기시간(50분)에는 센서를 자동세척 할 수 있는 센서 자동세척시스템을 개발하여 2021년 12월에 설치완료 하였다.
- 현장 모니터링 결과는 수위값은 2021년 4월 설치 이후, 환경모니터링은 2021년 10월 개별센서(Aqualabo, France)를 설치한 이후의 자료를 활용하였다.

○ 현장 모니터링 결과

표 5-1 현장 적용성시험(모니터링) 결과

구 분	Temperature(℃)	pH	ORP	EC (mS/cm)	Level
평균값	20.59	7.38	-411.95	20.59	2.48
최대값	27.80	7.81	-315.96	27.80	3.71
최소값	13.42	6.90	-456.95	13.42	1.22

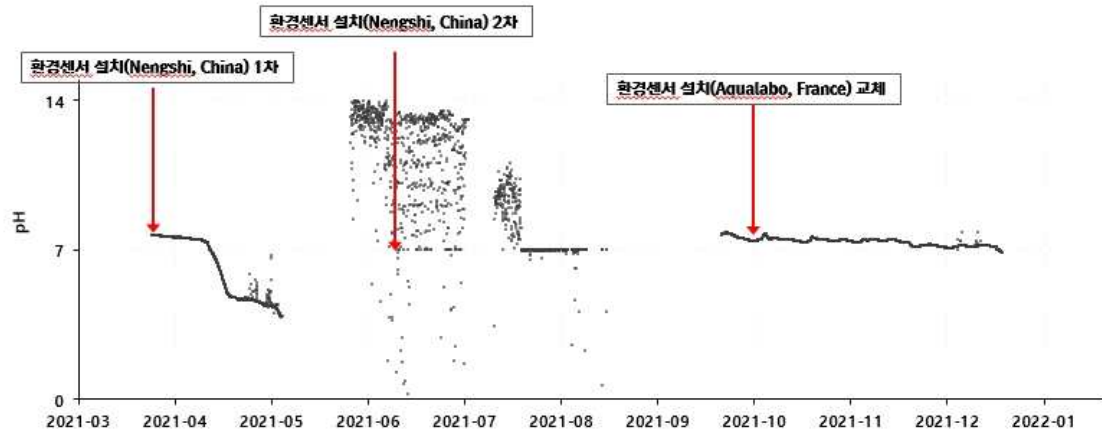


그림 5-2. 현장적용성시험 결과

## 5-5. 환경센서 자동세척시스템 개발

### 5-5-1. 필요성

- 1차 선정된 보급형 센서 ASPRT2151D(Nengshi, China)를 사용하여 현장 모니터링 운영 결과 환경센서는 침적형으로 사용할 경우 축산분뇨의 열악한 환경으로 인하여 내구성에 문제가 되는 것으로 확인이 되었다.
- 따라서 환경센서를 측정시간 동안에만 침적시키고 대기시간에는 저장조에서 분리시켜 세척 및 대기시키는 별도의 장치를 고안하여 환경센서의 내구성을 증대시키고자 하였다.

### 5-5-2. 환경센서 자동세척시스템 특징

- 환경센서를 와이어윈치를 이용하여 원하는 위치까지 끌어올린 다음 세척장치로 이송하여 air compressor를 이용하여 세척 후 원하는 시간에 구동장치가 작동하여 측정 위치로 이동 가능하며, 원하는 시간만큼 주기적으로 자동운전 가능토록 설계 및 제작하였다.
- 또한 실외의 축산분뇨 저장시설에 설치 하여야 하므로 동절기를 대비하여 heating cable을 사용하여 보온처리 하였다.
- 세척수는 1일(24시간)에 한번씩 자동교체 되도록 설계 하였다.



그림 5-3. 환경센서 자동세척 시스템

표 5-2 환경센서 자동세척시스템 제원

구분	사양	구분	사양
1. 구동장치		3. 세정장치	
구동방식	와이어 윈치	세정방식	Air 방식
형식	기어드 감속기	탱크용량	40L
전원	DC12V	구성	컴프레샤, 솔밸브, 불탑, 자동드레인밸브
속도	40rpm	재질	STS304
2. 이송장치		4. 프레임	
구동방식	체인방식(#40)	재질	STS304
형식	기어드 감속기	크기	400 × 1200 × 1700(H)
전원	DC12V		
속도	16rpm		

5-5-3. 환경센서 자동세척장치 설치

- 2021. 12월 환경센서 자동세척시스템을 설치하고 시범 운영을 실시하였다.
- 수동 및 자동운전이 가능하도록 RTU 업데이트도 병행하였다.



그림 5-4. 환경센서 자동세척 시스템 설치

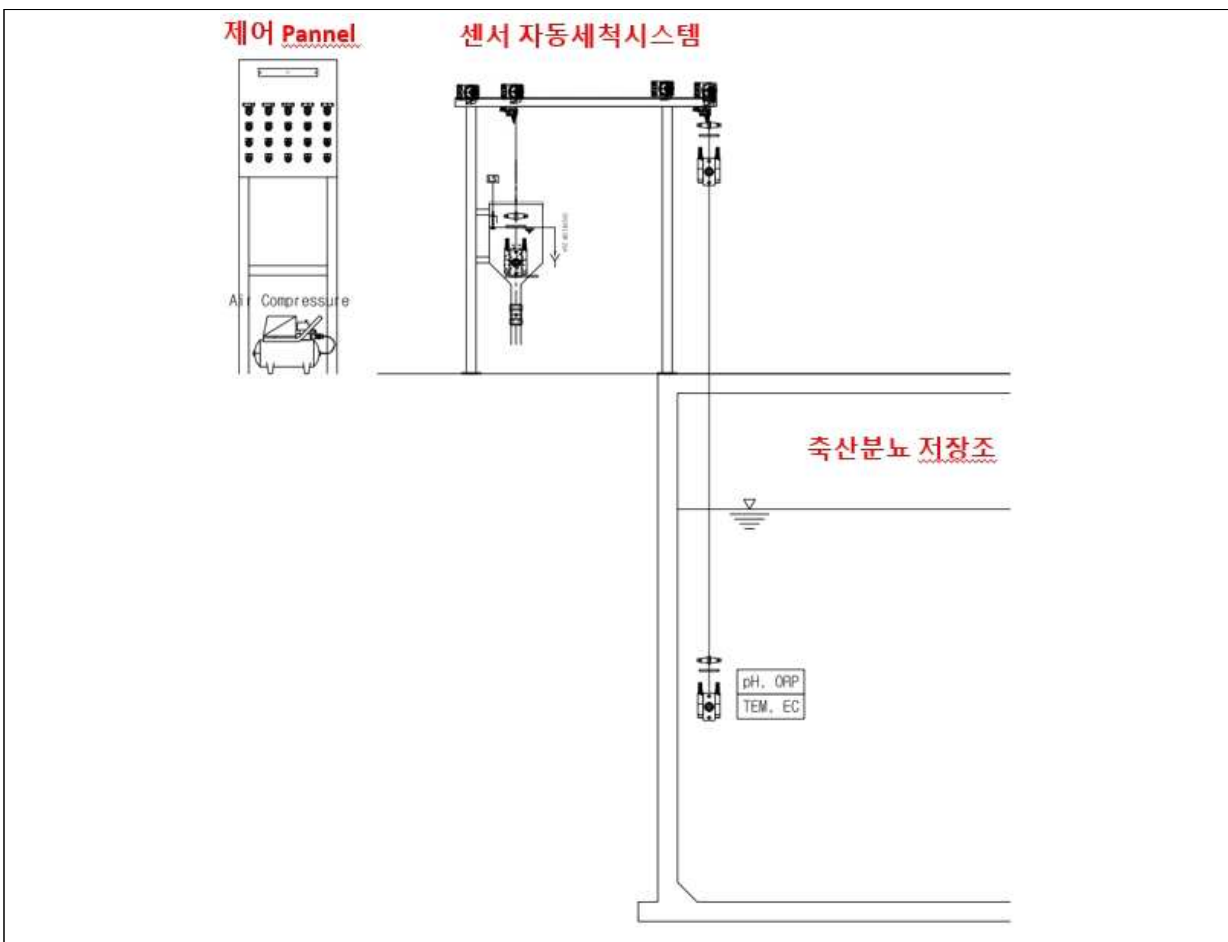


그림 5-5. 환경센서 자동세척 시스템 설치도면

## 6. 가축분뇨 발생량 측정정보 전송시스템 현장 적용시험

### 6-1. 측정정보 수집을 위한 전송 시스템 현장 적용성 시험

- 현장적용시험 초기에는 센서에서 측정 데이터를 상시적으로 수신하여 1시간 평균 데이터를 작성하였다.
- 또한 축분저장조 현장의 특성상 센서의 내구성이 검증된 장비의 사용이 요구되었다.
- 측정 알고리즘 변경
  - RTU는 한시간 단위로 설정한 타이머의 접점 신호를 받는다.
  - NH<sub>3</sub>, H<sub>2</sub>S, CO<sub>2</sub> 센서는 상시 전원이 인가되어 있는 상태이지만 평시에는 전처리 장비에 의해 차단되어 필터의 소모를 최소화 하도록 하였다.
  - 환경센서(온도, pH, ORP, EC)는 측정시간에만 전원을 인가하도록 구성하였다.
  - 타이머는 접점신호 발생 후 전처리 장비의 차단을 해제한다.
  - RTU는 접점신호를 받은 후 바로 측정을 시작하지 않고 대기시간(TIMER 기능에서 설정 가능)을 갖고, 이때 RTU에서 환경센서의 전원을 인가한다.
  - 설정된 시간 만큼의 시간이 지나면 각 센서는 측정을 시작한다.
  - 측정시간(TIMER 기능에서 설정 가능) 동안 데이터를 취합하여 평균데이터를 생성한 후 환경 센서는 RTU에서 전원을 차단하고 NH<sub>3</sub>, H<sub>2</sub>S, CO<sub>2</sub> 센서는 설정된 타이머에 의하여 다시 차단 상태로 변환된다.
- 특이상황(전기적 이슈 등)을 제외하고 정상적인 데이터 측정 및 저장확인이 가능하다.

### 6-2. 수집 데이터의 통합 관리를 위한 통신장치 현장 적용시험

- 통합관리 서버와의 데이터 송수신을 위한 고정IP 라우터를 설치하였다.
- 서버간 통신 프로토콜은 추후 범용성을 위하여 한국환경공단에서 사용하는 자동측정기기 통신표준규격(TMS 프로토콜)을 사용하였다.
- 덤프 데이터 송수신 방식으로 현장 또는 서버 상황으로 데이터 미전송시 추후 원하는 시기의 데이터 수신이 가능하다.



# <축사별 가축분뇨 처리와 저장 현황조사 및 모니터링 인자 선정, 강원대학교>

## 1. 양돈분뇨 관리현황 조사

### 1-1. 국내 돈사형태별 돈분의 저장·처리 현황조사

#### (1) 국내 현황조사 개요

2020년 (3분기까지) 기준 국내 주요 경제가축 (한우, 젓소, 돼지, 닭)의 총 사육두수는 198,567천 두였으며, 이중 돼지는 11,220천 두로 조사되어 전체의 약 5.7%를 차지하고 있다. 지역적으로 나눠 살펴보면 광역시·도 범위에서 충청남도는 전체 돼지의 20.9%인 2,342천 두를 사육하고 있으며, 경상북도와 전라북도가 그 뒤를 따르고 있다. 2010년 전국적으로 발생한 구제역으로 인해 상당량의 돼지가 살처분됨으로 인해 감소했던 돼지의 사육두수는 그 다음해 빠르게 회복하였으며, 이후 꾸준히 증가하였다.

가축의 사육으로 인해 발생하는 연평균 가축분뇨량은 약 50,170천 t으로 산출되며, 이중 돈분이 기여하는 비율은 전체의 41.6% (20,887천 t/년)로 국내 모든 가축 중 가장 높은 수치를 나타내어 돈분의 처리가 국내 분뇨처리의 핵심 현안인 것으로 파악되었다.

정부의 지속적인 가축분뇨 자원화 정책으로 인해 2017년 기준 돈분 발생량의 약 78.0%가 퇴·액비 자원화되고 있으며, 돈분의 높은 수분함량과 양분함량으로 인해 개별농가 수준의 퇴·액비 자원화, 정화처리가 타 가축에 비해 어려운 바 돈분의 약 60.6%는 가축분뇨 공동자원화시설이나 가축분뇨 공공처리시설에서 위탁처리되고 있다 (관계부처 합동, 2018)

표 3-1 돈분 처리현황 (관계부처 합동, 2018)

처리방법	구분	전체	소계	퇴비화	액비화	정화처리	기타
자체 처리	처리량 (1,000 t/년)	27,600	10,880	5,021	2,720	1,465	1,674
	비율 (%) <sup>*</sup>	100.0	39.4	18.2	9.9	5.3	6.1
공공(위탁) 처리	처리량 (1,000 t/년)	27,600	16,720	10,042	3,751	2,661	266
	비율 (%)	100.0	60.6	36.4	13.6	9.6	1.0

\*비율, 전체 처리량인 27,600,000 t/년 기준

돈분의 발생형태는 돈사의 배분구조에 따라 나뉘질 수 있으며, 배분구조는 크게 슬러리, 스크레퍼, 인력수거로 나뉜다. 분뇨 혼합식인 슬러리에 비해, 스크레퍼와 인력수거는 분과 뇨가 분리되어 배출·저장된다. 가축분뇨의 발생 후 위탁시설의 적정 수거시점을 개발하는 목적을 가진 본 연구에서 돈사의 형태별 분뇨의 발생·저장 유형은 돼지농가의 분뇨 관리방향을 정하는 중요한 요인이므로, 본 챕터에서는 돈사의 형태에 따른 분뇨의 저장·처리 현황을 국내 통계자료를 이용하여 분석하였다.

#### (2) 국내 현황조사 결과

##### ○ 돈사의 형태 및 배분구조

돈사의 형태는 크게 환기시설을 제외한 모든 부분이 막혀 인공적으로 돼지의 사육환경을 조절하는 무창돈사와 일반적으로 돈사의 측면부를 윈치커튼을 이용하여 개폐가 가능하도록 설계하

여 온도 및 환기를 조절하는 개방형 돈사가 있으며, 일부 비닐하우스 등 구조물을 이용하여 축사로 사용하는 간이(기타) 돈사로 구분할 수 있다.

2015년 기준 전국의 돈사 형태별 비율은 무창돈사 46.0%, 개방형 돈사 40.8%, 간이(기타) 돈사 13.2%로 조사되었으며, 돼지 사육의 고도화 및 악취 민원 증가로 인한 돈사 보수 등으로 인해 무창돈사가 증가하는 추세이다.

특히 돈사 유래 악취 확산을 방지하기 위해 농림축산식품부와 환경부에서는 축산환경종합개선 대책안에 악취저감시설을 의무화하는 방안을 포함하고 있으며, 바이오필터, 바이오커튼 등의 악취저감시설은 악취물질이 배출되는 지점을 중점적으로 관리하는 방법으로 무창돈사에서만 그 효과를 기대할 수 있으므로, 사실상 개방형 돈사의 무창돈사화가 강제되고 있는 상황이다.

표 3-2 돈사 형태별 농가수 (통계청, 2015)

분류	전체	무창돈사	개방형 돈사	간이(기타) 돈사
농가 수 (호)	4,375	2,013	1,784	578
비율 (%)	100.0	46.0	40.8	13.2

돈사의 배분구조는 앞서 언급한 것과 같이 크게 슬러리, 스크레퍼, 인력수거로 구분되며, 일부 깔짚돈사도 존재한다. 배분구조는 돈사의 형태의 영향은 크게 받지 않으나 저장되는 분뇨의 형태에 영향을 미치므로 이에 따른 저장조의 종류는 구별되어진다.

분뇨가 혼합되어 저장되는 슬러리 돈사는 설비가 간단하고 돈분뇨 관리 시 분뇨 수거 및 처리가 용이하여 노동력이 적게 들어 사육규모가 커질수록 슬러리 돈사를 채택하는 경향을 보인다(이 등, 2014). 분과 뇨를 분리하여 저장하는 스크레퍼 돈사는 구배가 있는 콘크리트로 바닥이 구성되며, 스크레퍼에 의해서 분과 뇨가 자연스럽게 분리 저장된다. 분뇨의 혼합 시 추가적인 고액분리 과정이 요구되나, 스크레퍼의 경우 분뇨가 분리되어 수거되면서 자원화가 유리해지는 장점을 지닌다.

일반적으로 콘크리트 평사를 이용하는 농가에서 분뇨를 인력수거하며, 뇨는 별도의 저장조에 저장하고, 분은 인력수거를 통해 퇴비사에 저장·처리한다. 기타 방법으로는 깔짚돈사 형태가 존재하며, 톱밥 등 깔짚을 돈사에 깔아 사육기간 동안의 분뇨를 깔짚이 흡수하도록 한다. 돼지가 출하된 후 깔짚과 섞인 분뇨는 함께 제거되어지며, 회수된 분뇨+깔짚 혼합물은 퇴비화를 진행한다.

2011년에서 2019년까지 돼지농가의 배분구조를 비교한 결과 슬러리 돈사의 비중은 꾸준히 증가한 반면, 스크레퍼, 인력수거 (바닥식-깔짚 & 바닥식-깔짚없음)은 감소하는 경향을 보인다(통계청, 2020). 통계청에 따르면 2019년 기준 70.5% 이상의 돼지농가에서 슬러리 돈사를 채택하여 분뇨를 저장하고 있었으며 바닥식-깔짚 12.1%, 스크레퍼 9.0%, 바닥식-깔짚없음 8.4% 순으로 나타났다.

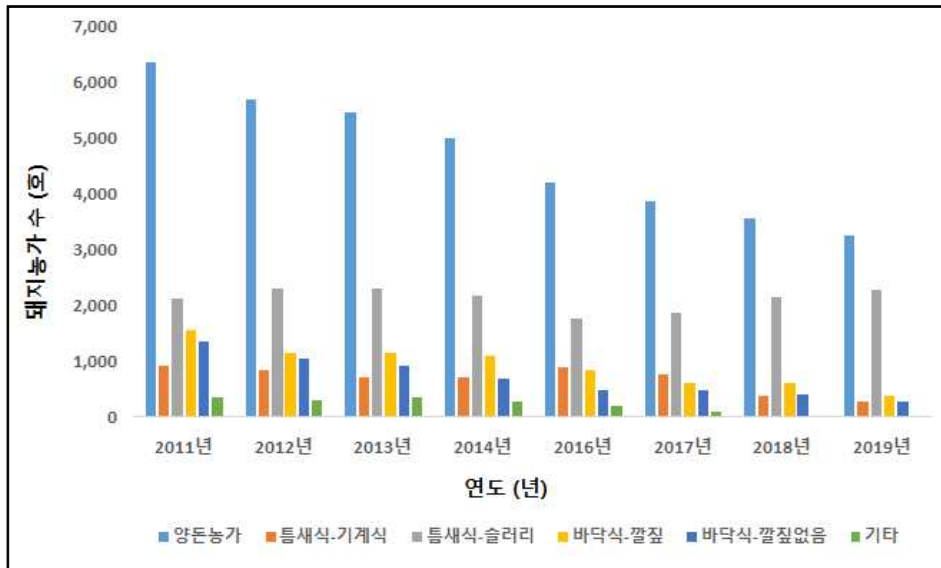


그림 3-1. 돈사 배분구조별 농가 수 변화 (통계청, 2020)

표 3-3 2019년 기준 돈사 배분구조별 농가 비율 (통계청, 2020)

분류	전체	틈새식		바닥식	
		스크래퍼	슬러리	깔짚	깔짚없음
돼지농가 수 (호)	3245	293	2288	392	272
비율 (%)	100.0	9.0	70.5	12.1	8.4

○ 분뇨의 이송 및 관리

슬러리 돈사의 규격은 농가의 환경에 따라 상이하나 표준축사설계도에 제시된 깊이는 120cm 이며, 분뇨가 섞여 있는 슬러리의 특성상 분뇨 이송이 용이하다는 점을 이용하여 슬러리 피트 내 분뇨의 처리를 위한 이동을 위해 다양한 방법이 사용된다 (농림축산식품부, 2016).

- 슬러리 피트 바닥에 분뇨 이송관을 설비하여 돈분 저장조로 이송하는 방식이 있으며, 시설이 간단하고 돈방의 보온, 유지·관리가 용이하다.
- 슬러리 피트 바닥에 구배를 주거나 피트에 관 혹은 이송피트를 설비하며, 피트와 저장조 사이에 높이차를 두어 중력에 의한 낙하방식으로 이송되는 방법으로 피트와 연결된 출구를 열고 닫음으로써 자연스럽게 분뇨가 저장조로 이동하도록 설계되었다. 시설이 간단하고 돈방의 보온, 유지·관리가 용이하다.
- 돈분을 스크래퍼 혹은 컨베이어 벨트로 긁어내어 분과 뇨를 분리하여 농가 내 자체 자원화 시 고액분리 비용을 줄일 수 있으나 운영시설의 관리가 필요하고, 돈방의 하부가 개방되어 돈방 내 단열효과가 저하된다.
- 슬러리 피트 내 고압호스와 이송펌프를 이용하여 강제로 저장조로 분뇨가 이송된다.

슬러리 피트 내 분뇨는 저장기간 중 고형물이 피트 바닥쪽으로 침전됨에 따라 바닥부분에서부터 이송시키는 설비가 필요하며, 주기적으로 피트 바닥에 남아 이송되지 않는 분뇨를 제거해주어야 한다.

위탁처리 업체에 분뇨를 위탁하는 돼지 농가에서 슬러리 피트에서 저장조로 이송된 분뇨는 슬러리 형태 그대로 위탁되거나, 농가 내에서 자체적으로 고액분리된 후 액상분뇨만 위탁되어지며, 수거하는 분뇨의 성상은 위탁시설의 처리공정 혹은 처리조 부하에 따라 시설에서 돼지농가에 요

구한다.

#### ○ 돈분 저장조의 종류

돼지농가의 돈분 저장조는 일반적으로 지하에 매장된 육면체 형태의 콘크리트 저장조가 이용되고 있으며, 지상의 액비저장조를 돈분 저장조로 사용하는 농가도 적지 않다. 돈분의 이송방법에 따라 관 또는 피트와 연결되거나 고압호스를 연결해 놓고 있다.

돈사의 슬러리 피트가 일정량 이상 차면 돈분 저장조로 분뇨를 이송시키는데 현재 저장조의 수위 조절은 저장조 입구에서 직접 확인하는 육안 확인을 하고 있으며, 일부 농가에서는 저장조 입구의 덮개도 설치해두지 않아 외부로부터 비나 눈이 그대로 유입되고 있기 때문에 관리자의 실수나 적절한 시기에 분뇨가 수거되지 못해 돈분 저장조가 넘치는 문제가 종종 발생한다.

돼지농가의 관리자 경험에 의존한 분뇨수거체계는 돈분 저장조의 관리 미비로 인한 환경문제를 제어할 수 있는 그 어떤 안전장치도 갖춰져 있지 않아 이로 인한 환경문제를 야기할 수 있으므로 돈분이 외부로 유출되는 상황을 막을 수 있는 처리시스템 구축이 필요하다.

#### ○ 돈분 저장조 내 분뇨 성상

돈분 저장조 내 분뇨의 성상을 조사하기 위해서 배분구조가 슬러리 형태인 농가 4 곳의 분뇨를 분석하였다.

조사된 농가들의 데이터를 기준으로 각 분석항목별 저장조 내 분뇨 범위는 총 고형물 (total solids, TS): 11.9 ~ 106.1 g/L, 총 휘발성고형물 (total volatile solids, TVS): 7.3 ~ 82.7 g/L, 총 부유성고형물 (total suspended solids, TSS): 3.7 ~ 112.5 g/L, 암모니아성 질소 (NH<sub>4</sub>-N): 1,383.5 ~ 5,577.0 g/L, 총 질소 (T-N): 2,266.2 ~ 8,812.2 mg/L, 용해성 인 (O-P): 25.6 ~ 66.4 mg/L, 총 인 (T-P): 25.6 ~ 867.9 mg/L로 조사되었다.



매립형 콘크리트 저장조 (덮개 無)



매립형 콘크리트 저장조 (덮개 有)



지상형 콘크리트 저장조



돈분 저장용 액비저장조

그림 3-2. 돈분저장조 모습

표 3-4 농가별 돈분 저장조 내 분뇨 성분 분석 결과

농가	TS	TVS	TSS	TVSS	NH <sub>4</sub> -N	T-N	O-P	T-P	COD <sub>cr</sub>	BOD <sub>5</sub>
	g/L					mg/L				
a 농가 육성사	11.9 ± 0.2	7.3 ± 0.2	5.7 ± 0.1	4.8 ± 0.0	1,383.5 ± 11.0	2,266.2 ± 10.2	52.6 ± 2.1	402.4 ± 2.9	27,534.0 ± 770.0	<b>11,380.0</b> ± 361.7
b 농가 비육사	32.5 ± 0.0	21.8 ± 0.2	21.3 ± 0.7	18.0 ± 0.5	2,152.7 ± 22.7	3,609.0 ± 114.0	66.4 ± 1.8	718.7 ± 1.1	58,592.0 ± 1,304.0	<b>22,440.0</b> ± 1,527.4
b 농가 비육사	39.5 ± 0.9	27.4 ± 0.8	29.5 ± 1.2	23.9 ± 0.8	2,135.0 ± 17.6	3,991.2 ± 4.4	50.9 ± 1.4	867.9 ± 5.6	69,839.8 ± 3,395.0	<b>28,120.0</b> ± 1,304.0
c 농가 비육사	15.0 ± 0.2	8.7 ± 0.1	3.7 ± 0.1	3.4 ± 0.0	2,145.1 ± 6.1	3,712.6 ± 14.3	25.6 ± 0.2	120.2 ± 4.0	15,525.9 ± 152.0	<b>6,100.0</b> ± 249.8
c 농가 비육사	13.5 ± 0.2	7.6 ± 0.2	3.9 ± 0.1	3.2 ± 0.0	1,740.0 ± 6.5	3,227.4 ± 9.0	32.0 ± 0.2	111.0 ± 0.8	13,179.1 ± 214.0	<b>4,640.0</b> ± 138.6
d 농가 비육사	106.1 ± 0.3	82.7 ± 0.6	112.5 ± 4.1	92.7 ± 3.2	5,577.0 ± 47.3	8,812.1 ± 136.0	46.6 ± 0.4	167.8 ± 0.9	<b>27,485.8</b> ± 716.0	-

저장조 내 분뇨 물성은 각 농가별로 매우 상이하였으며, 이는 농가별 사료 및 사양방식, 청소주기, 저장조 내 체류기간 등 다양한 요인의 차이에서 비롯된 것으로 판단된다.

## 1-2. 익산시 축산업 데이터 기반 테스트베드 환경 분석 (현장조사)

### (1) 현장조사 개요

국내 축산농가에서 가축분뇨의 위탁처리 시 배출부터 운반/최종처리까지 전 과정은 인터넷과 모바일을 이용하여 실시간으로 관리되는 가축분뇨 전자인계관리시스템을 이용해야 한다. 한국환경공단에서 운영하는 가축분뇨 전자인계관리시스템이 가축분뇨의 처리 전주기 (발생-수거-처

리) 플랫폼으로 거듭나기 위해서는 위탁처리하는 축산농가에서 발생하는 전체 가축분뇨가 올바르게 처리되는 과정을 확인할 수 있도록 가축분뇨 발생량을 고려해야 한다.

위탁되는 분뇨를 시작으로 모니터링을 실시하는 현 시스템에서 농가에서 발생한 분뇨의 전량을 고려하는 방향으로의 시스템 확장은 일부 비양심적인 축산농가와 위탁처리업체의 무단 살포, 부적절한 퇴·액비화 등을 방지할 수 있고, 최적 위탁시점을 자동으로 위탁시설에 보고하여 가축분뇨 위탁시점을 자동으로 관리할 수 있으며, 발생한 분뇨는 시스템하에서 안정적으로 처리되고 있다는 사회적 인식을 확보함으로써 경축순환 활성화에 이바지 할 수 있을 것으로 기대된다. 이를 위해 농가 내에서 발생한 분뇨의 저장조의 수위를 모니터링하여 발생한 전체 분뇨의 수거시점을 판단하는 시스템을 개발이 필요하며, 본 연구를 위해 MOU를 맺은 익산시와 협의하여 가축분뇨를 위탁처리하는 농가 2곳을 테스트베드로 선정하였다.

본 연구에서는 농가의 분뇨 발생 현황과 위탁처리 방식의 특징을 모니터링 인자 패턴 변화 분석 시 적용하기 위해서 선정된 테스트베드 농가의 사육현황, 배분구조, 분뇨관리 방안 등에 대한 현장조사를 수행하였다.

## (2) 테스트베드 환경분석 결과

### (가) 익산시

#### ○ 익산시의 돼지 사육현황 및 축산관련 정책 기초

테스트베드 농가가 위치한 익산시는 “악취 저감으로 시민이 행복한 환경 친화도시 조성”이라는 비전을 내세워 악취저감을 위해 유형별 악취발생 시설에 대한 관리 및 규제를 강화하는 움직임을 보이고 있으며, 축산농가 또한 신고대상 외 악취배출시설로 분류하여 관리 계획을 수립하였다 (익산시 환경관리과, 2020).

2019년 말 기준, 익산시의 축산농가는 총 1,247호, 전체 가축 사육두수는 5,785천 두로 보고되었으며, 이중 돼지 사육농가는 186호, 돼지 사육두수는 191,807 두로 조사되었다. 익산시의 돼지 사육두수는 전체 사육두수의 3.3% 수준이나 가축분뇨는 연간 357,048 t을 발생시켜 전 축종 중 가장 높은 가축분뇨 발생 기여도 (62.9%)를 나타내어, 익산시에서 처리되어야 하는 주된 가축분뇨는 양돈분뇨로 확인되었다.

표 3-5 익산시 축종별 농가 현황 및 가축분뇨 발생량 (2019.12.31. 기준)

분류	계	한육우	젖소	돼지	가금*
농가수 (호)	1,247	869	37	186	155
사육두수 (두)	5,785,905	36,198	2,113	191,807	5,555,787
사육비율 (%)	100.0	0.6	0.04	3.3	96.0
가축분뇨 발생량 (t/년)	567,717	181,008	29,076	357,049	584
가축분뇨 발생 기여도 (%)	100.0	31.9	5.1	62.9	0.1

\*가금, 닭과 오리의 합으로 가축분뇨 발생량 산정 시 육계와 산란계의 가축분뇨 배출원단위의 평균을 이용함.

돼지농가에서 주된 악취발생 원인은 개방된 사육시설 및 분뇨관리 시설 (돈사, 분뇨 저장시설, 퇴·액비화 시설), 부적절한 양돈분뇨 처리, 악취저감시설 미비 등이 있으며, 이를 위해서 익산

시에서는 퇴·액비시설 밀폐 및 악취저감시설 설치 등 시설 개선사업과 분뇨의 신속수거 저장량 최소화, 악취저감 미생물 (광합성균) 보급 등의 지원계획을 수립·추진하고 있다 (익산시 축산과, 2020). 특히 익산시 왕궁면의 경우 80여 농가가 돼지 7만여 마리를 사육하고 있는 축산악취 심각지역으로 환경부는 ‘왕궁 정착농원 특별관리지역 지정 고시’를 통해 국비 123억원을 들여 왕궁면 일대의 돈사의 매입을 진행하고 있다. 익산시의 악취저감 계획 중 분뇨의 신속수거 및 저장량 최소화는 본 연구의 결과를 바탕으로 관리시스템 및 기술의 고도화가 가능할 것으로 판단된다.

○ 익산시의 분뇨 관리 여건 분석

전라북도 최북단에서 충청남도과 맞닿아 있는 익산 주변은 논산, 부여, 서천, 군산, 김제, 완주 등 주변 지역에 가축분뇨 처리를 위한 시설들이 충분히 위치하고 있기 때문에 적절한 관리가 어려운 개별농가의 자원화시설이나 정화처리시설 운영보다 위탁처리가 악취 등 가축분뇨 관련 문제를 완화시킬 수 있는 방안일 것으로 판단된다.

가축분뇨 발생량의 절반 이상을 차지하는 양돈분뇨의 주기적인 위탁처리를 위해 익산시에서는 가축분뇨 공동자원화시설 (우주원 영농조합법인) 1개소, 가축분뇨 공공처리시설 (익산시 가축분뇨 공공처리시설) 1개소를 운영중이며, 군산에 위치해 있으나 익산군산축협 가축분뇨 공동자원화센터도 익산시 내 양돈분뇨를 위탁처리하고 있다. 특히 익산시 가축분뇨 공공처리시설은 익산시 전역의 양돈분뇨를 포괄적으로 관리하나 왕궁면의 분뇨처리를 우선적으로 수행하며, 짧은 수거주기를 통해 왕궁면 돼지농가 내 악취를 저감하고자 노력하고 있다.

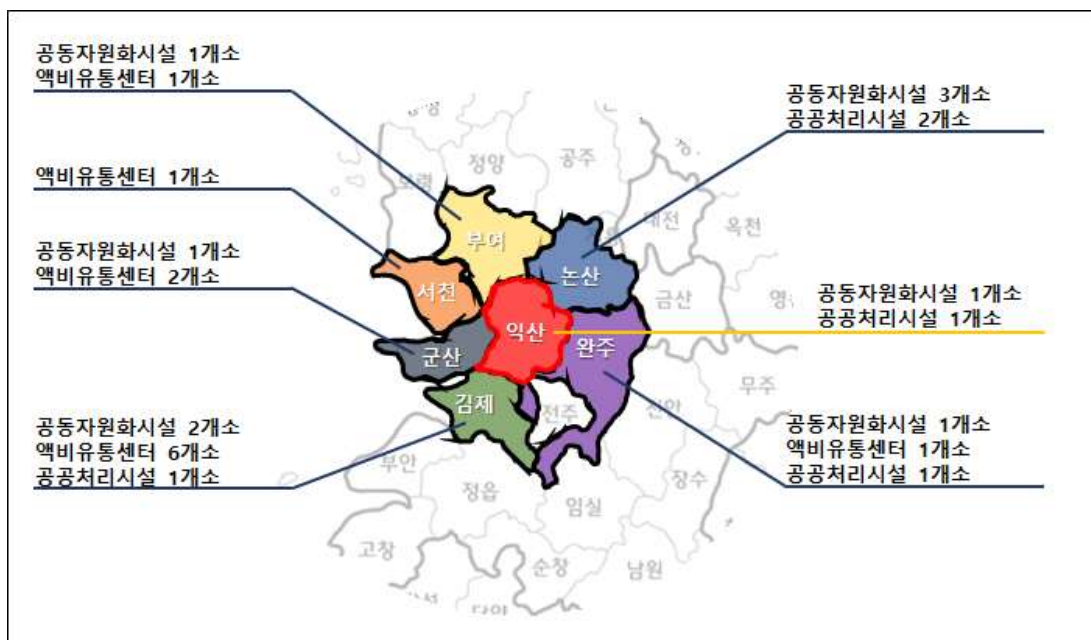


그림 3-3 익산시와 주변 시군의 가축분뇨 처리시설 현황

(나) 테스트베드 농가 현장조사

테스트베드 농가는 총 2곳이 선정되었으며, 각 농가의 운영현황 및 분뇨관리 현황을 중심으로 현장조사 내용을 기술하였다.

일반적으로 가축분뇨 처리시설에 위탁 시 위탁된 양돈분뇨의 성상은 발생한 원분뇨 (슬러리) 형태 그대로의 분뇨와 농가에서 고액분리된 액상분뇨로 구분할 수 있으며, 선정된 2개의 농가 중 A 농가 (익산시 낭산면)는 고액분리된 액상분뇨, B 농가 (익산시 왕궁면)는 원분뇨 (슬러리)를 각각 위탁하고 있다.

본 연구에서는 농가에서 위탁하는 저장조 내 분뇨의 수위 및 pH, ORP, EC를 측정하여 분뇨의 성상변화를 관찰하였으며, 추가적으로 분뇨 저장조에서 발생하는 악취물질 ( $\text{NH}_3$ ,  $\text{H}_2\text{S}$ )의 수준도 측정하였다.

○ A 농가 (고액분리된 액상분뇨 위탁) 개요

A 농가는 모든 190두, 자돈 및 육성/비육돈을 약 2,010두 사육하고 있는 돼지농가로 8동에 이르는 전 축사가 모두 밀폐형 (창문형), 슬러리 돈사로 설비되어 있으며, 측벽에 있는 환기팬을 이용하여 강제배기를 하고 있다.

분뇨처리를 위한 시설로는 분뇨 성상별 저장조, 퇴·액비화시설, 액비저장조, 고액분리기를 갖추고 있으며, 돈사에는 안개분무시설, 바이오커튼 (탈취제 사용) 등 악취저감시설도 구축하고 있다.

- A 농가의 운영현황 및 조사된 정보는 <표 3-6>에 간략히 요약 제시하였다.



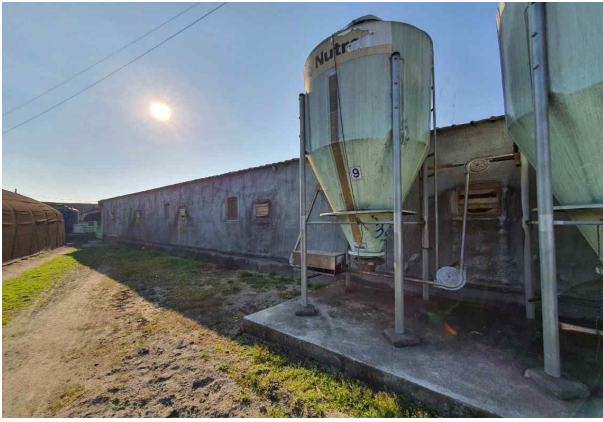
그림 3-4. A 농가의 위성사진 및 분뇨관리시설의 위치 (A, 원분뇨 저장조; B, 고액분리기; C, 퇴비화시설; D, 액상분뇨 저장조; E, 액비화시설; F, 액비저장조)

\*액상분뇨 저장조 입구에서 수위, 분뇨 성상 변화인자, 환경 영향인자 모니터링

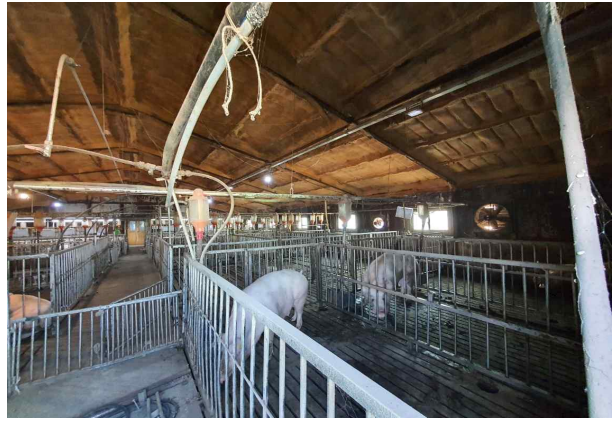


표 3-6 A 농가 현장조사 결과 요약

현황	내용
사육현황	- 모든 190두, 자돈 및 육성/비육돈 2,010두
운영 현황	<p>축사형태 및 시설 구축</p> <ul style="list-style-type: none"> <li>- 무창/슬러리 돈사</li> <li>- 전면 콘크리트 슬랏 (통로 제외)</li> <li>- 구배없는 슬러리피트 바닥</li> <li>- 환기: 측벽배기</li> <li>- 퇴비화시설, 액비화시설 (하우스 밀폐)</li> <li>- 악취저감시설:                             <ul style="list-style-type: none"> <li>· 돈사 내부 안개분무</li> <li>· 측벽배기 외부 바이오필터 (약품 이용). 현재 사용하지 않음.</li> </ul> </li> </ul>
분뇨배출 방법	<ul style="list-style-type: none"> <li>- 슬러리 형태의 분뇨 발생</li> <li>- 분뇨 발생량: 10 t/일</li> <li>- 펌프를 이용한 슬러리 피트 내 분뇨 강제 이송</li> <li>- 슬러리피트 비우는 주기: 분뇨가 가득차면 이송</li> </ul>
분뇨관리 현황	<p>처리방법</p> <ul style="list-style-type: none"> <li>- 고액분리: 약 1회/일 (부정기적)</li> <li>- 고액분리 후 액상분뇨 위탁처리</li> <li>- 일부 액상분뇨 액비화 후 위탁처리</li> </ul> <p>분뇨수거 주기</p> <ul style="list-style-type: none"> <li>- 상황에 따라 변경</li> <li>- 액상분뇨: 월 2~3회 위탁 (익산군산축협 가축분뇨 공동자원화센터, 매주 수요일, 25 t/회).</li> <li>- 액비 및 고상분뇨: 비주기적</li> </ul>
분뇨관리 시설용량	<ul style="list-style-type: none"> <li>- 액상분뇨 저장조: 400 t, 깊이 4 m, 위탁지점 ⇒ 모니터링!</li> <li>- 퇴비화시설: 800 t, 현재 非사용</li> <li>- 액비화시설: 600 t, 장기포기</li> <li>- 액비저장조: 400 t, 생산된 액비 저장</li> </ul>
기타 시설 및 분뇨관리 특이사항	<ul style="list-style-type: none"> <li>- 원분뇨 저장조와 액상분뇨 저장조의 입구 개방 ⇒ 악취발생</li> <li>- 퇴비화시설은 고액분리된 고상분뇨의 저장용도로만 사용 ⇒ 위탁처리</li> <li>- 하우스로 밀폐 (출입구는 개방)된 액비화시설 내부 암모니아가 상당히 높아 눈을 뜰수 없을 지경 (작업자 위험요소)</li> <li>- 악취저감을 위해 바이오커튼 설비 예정</li> <li>- 익산시에서 자동 악취포집장치를 이용 농장 주변의 악취를 주기적으로 확인</li> </ul>



돈사 형태 (밀폐형, 측벽배기)



돈사 내부 (슬러리 돈사, 전면 콘슬랏)



원분뇨 저장조 (콘크리트 매립)



액상분뇨 저장조 (우), 액비저장조 (좌)



퇴비화시설 (SCB, 운영 X)



액비화시설 (장기포기)

그림 3-5. A 농가 분뇨관리시설 모습

○ A 농가의 분뇨관리 현황

10 t/일의 분뇨가 발생하는 A 농가는 각 축사별 슬러리 피트 내 분뇨가 가득차면 펌프를 이용하여 강제로 원분뇨 저장조로 이송시키며, 원분뇨 저장조 내 슬러리는 약 1회/일 일정시간 동안 고액분리된 후 고상분뇨는 퇴비화시설, 액상분뇨는 액상분뇨 저장조에 보관된다.

A 농가의 액상분뇨 저장조는 약 400 t의 크기를 가지고 있고, 월 2~3회 익산군산축협 가축분뇨 공동자원화센터로 위탁되어지며, 저장된 액비의 경우 주변 경종농가에 비주기적으로 연락하여 수거해가도록 한다. 특히 일정한 양의 액상분뇨를 주기적으로 수거해 가는 공동자원화센터와 달리 경종농가의 경우 액비의 시비가 가능한 시기 (10월~4월)에 액비 수거량이 증가한다.

고액분리된 고상분뇨 또한 경종농가에서 수거하며, 현재 퇴비화시설은 가동하지 않고있다. 퇴

비화시설의 경우 가축분뇨 퇴비단여과법 (Slurry Composting & Biofiltration, SCB)을 적용하여 바닥에 산기관과 여과액 이동통로가 설치되어 있으며, 상부의 교반기는 현재 철거하여 운영하지 않고 있다.

액비화시설은 밀폐형 콘크리트로 시공되어 있으며, 악취발생을 최소화하기 위해 비닐하우스로 밀폐 (입구 제외)하였는데, 이로 인해 비닐 하우스 내 암모니아 가스 농도가 굉장히 높은 것으로 나타났다 (암모니아 가스로 인해 눈을 뜰 수 없는 수준). 돈사 내 악취확산을 방지하기 위해 A 농가에서는 추후 측벽배기 바깥으로 바이오커튼을 설치하기로 계획하고 있으나, 액상분뇨 저장조의 입구는 현재 개방형으로 액상분뇨에서 배출되는 악취가스가 그대로 배출되고 있었고, 저장된 액상분뇨는 상층부에 두꺼운 스크층을 지니고 있었다.

### ○ A 농가의 분뇨 성상

A 농가의 분뇨 처리단계별 분뇨의 샘플을 채취하고, 이화학적 성상을 분석하였다. 슬러리 피트의 분뇨가 직접적으로 보관되는 돈분저장조의 경우 고액분리가 되지않은 상태로 타 처리단계에 비해 높은 고형물 및 유기물 농도, 질소 및 인 농도를 나타내었으며, 모니터링이 진행되는 액상분뇨 저장조의 경우 고형물 감소로 인한 유기물 농도 및 T-P 농도의 감소를 확인할 수 있었다.

A 농가의 액비조는 장기포기를 하고 있었으나 미생물의 부족으로 인해 분뇨 내  $\text{NH}_3$  탈기만 일어나고 있는 것으로 판단되었으며, 액비처리 후 액비저장조 내  $\text{NH}_4\text{-N}$  농도도 여름철 1,516.8 mg/L, 가을철 2,957.3 mg/L로 매우 높은 수준을 나타내 정상적인 생물학적 처리공정의 운전이 이뤄지지 않고 있었다.

A 농가의 여름철과 가을철 분뇨성상을 비교한 결과 액상분뇨 저장조를 기준으로 가을철에 모든 분석항목의 농도가 높았으며, 육안으로도 높은 농도의 고형물로 인한 분뇨 성상의 차이가 느껴졌다. 모니터링된 시스템 측정값과 분뇨 성상의 비교를 통해 적정 수거시점을 판단하는 근거로 데이터를 활용하였다.

표 3-7 A 농가의 처리단계별 분뇨 정상

A 농가	TS	TVS	TSS	TVSS	NH <sub>4</sub> -N	T-N	O-P	T-P	COD	BOD	
	g/L					mg/L					
여름철	돈분 저장조	33.9 ± 0.4	24.0 ± 0.4	25.1 ± 0.1	20.7 ± 0.0	2,445.3 ± 13.9	4,033.9 ± 6.4	163.3 ± 1.2	709.9 ± 21.6	70,173.7 ± 2,268.0	21,720.0 ± 965.6
	액상분뇨 저장조	16.7 ± 0.0	7.6 ± 0.0	5.8 ± 0.3	4.7 ± 0.1	2,530.5 ± 38.7	4,153.2 ± 13.6	147.5 ± 0.5	185.1 ± 4.8	10,980.1 ± 271.0	1,983.0 ± 94.5
	액비조	17.0 ± 0.5	7.8 ± 0.3	6.9 ± 0.0	5.3 ± 0.0	1,752.2 ± 12.2	3,339.2 ± 41.2	107.9 ± 1.4	227.3 ± 7.8	10,562.6 ± 103.0	1,863.8 ± 67.5
	액비 저장조	16.8 ± 0.1	7.7 ± 0.1	6.3 ± 0.0	4.8 ± 0.0	1,516.8 ± 11.4	2,929.8 ± 25.7	132.7 ± 0.8	181.3 ± 1.1	10,713.6 ± 230.0	1,869.7 ± 65.0
가을철	돈분 저장조	64.1 ± 2.5	50.4 ± 2.0	62.0 ± 2.9	51.9 ± 2.5	4,081.2 ± 12.2	6,420.8 ± 195.0	168.1 ± 1.8	1,317.1 ± 1.6	108,466.3 ± 4,417.8	-
	액상분뇨 저장조	26.1 ± 0.6	11.8 ± 6.8	19.9 ± 1.0	14.8 ± 0.6	2,978.9 ± 18.4	4,558.0 ± 7.4	71.7 ± 1.3	499.7 ± 2.9	29,975.2 ± 577.7	-
	액비조	13.3 ± 0.3	5.5 ± 0.2	5.0 ± 0.2	3.5 ± 0.1	2,665.9 ± 11.1	3,256.5 ± 4.8	61.2 ± 1.2	154.6 ± 0.7	7,022.1 ± 249.6	-
	액비 저장조	14.3 ± 0.3	7.0 ± 0.2	6.2 ± 0.1	4.3 ± 0.1	2,957.3 ± 4.1	3,384.4 ± 4.8	63.7 ± 0.6	46.6 ± 0.4	10,202.2 ± 489.9	-

○ B 농가 (고액분리된 액상분뇨 위탁) 개요

B 농가는 육성/비육돈 1,500두를 사육하고 있는 비육농가로 5동인 전 축사가 모두 개방형 (원치커튼), 슬러리 돈사로 설비되어 있으며, 지붕에 있는 환기팬을 이용하여 강제로 지붕배기를 하고 있었다. 분뇨처리를 위한 시설로는 원분뇨 저장조와, 퇴비화시설을 갖추고 있었으나, 퇴비화시설은 근처 농가 소유의 밭에 뿌릴 한우분뇨를 저장하기 위한 용도로 사용되고 있다. 왕궁면의 양돈단지에 위치한 B 농가의 분뇨저장조는 개방형으로 설비되어 있었고, 바로 붙어있는 주변 돼지농가들도 모두 개방형 돈사형태를 취하고 있어 악취가 많이 발생하고 있었다. 왕궁면 축산단지 특별관리기간의 연장으로 B 농가의 경우 내년 (2021년) 9월까지만 운영하고 돈사를 매각하기로 결정함에 따라 본 연구의 모니터링 기간에 영향을 미치지 않을 것으로 판단되어 농가의 운영현황 및 조사된 정보를 <표 3-8>에 간략히 요약하여 제시하였다.



그림 3-6. B 농가의 위성사진 및 분뇨관리시설의 위치 (A, 원분뇨 저장조; B, 퇴비화시설)  
 \*액상분뇨 저장조 입구에서 수위, 분뇨 성상 변화인자, 환경 영향인자 모니터링

○ B 농가의 분뇨관리 현황

7.5 t/일의 분뇨가 발생하는 B 농가는 각 축사별 슬러리 피트 내 분뇨가 가득차면 슬러리 피트의 밸브를 열어 원분뇨 저장조로 이송시키는데 이때, 돈사와 저장조 사이의 높이차에 의해서 자연적으로 이송되어진다. 고액분리 없이 저장조에 저장된 그대로의 분뇨를 익산시 가축분뇨 공공처리시설에 위탁처리하며, 일반적으로 주에 1~2회, 액상분뇨 7~15t을 시설에서 수거해가고 있다. 원분뇨 저장조의 저장용량은 280t으로 깊이는 3.5 m 로 조사되었으며, 원분뇨 저장조의 입구의 넓이가 A 농가에 비해 상대적으로 넓었다. 이는 보다 많은 악취가스의 확산을 야기하므로 환경영향 요인인 NH<sub>3</sub>, H<sub>2</sub>S의 측정을 위해서 입구를 막을 필요가 있는 것으로 판단되었다. 농장주가 돈사를 팔고 해당 지역을 나갈 예정이기 때문에 악취저감을 위한 노력은 확인 할 수 없었다.

표 3-8 B농장 현장조사 결과 요약

현황	내용
사육현황	- 육성/비육돈 1,500두
운영 현황	<ul style="list-style-type: none"> <li>- 원치커튼 (개방형)/슬러리 돈사 (5동)</li> <li>- 전면 콘크리트 슬랏 (통로 제외)</li> <li>- 구배없는 슬러리피트 바닥</li> <li>- 환기: 지붕배기</li> <li>- 퇴비화시설</li> <li>- 악취저감시설 無</li> </ul>
분뇨배출 방법	<ul style="list-style-type: none"> <li>- 슬러리 형태의 분뇨 발생</li> <li>- 분뇨 발생량: 7.5 t/일</li> <li>- 돈분 저장조가 돈사에 비해 아래에 위치. 슬러리 피트의 문을 열어 분뇨 자연 이송</li> </ul>
분뇨관리 현황	<ul style="list-style-type: none"> <li>- 고액분리 無</li> <li>- 저장된 돈분 슬러리 전량 위탁처리</li> </ul>
분뇨수거 주기	<ul style="list-style-type: none"> <li>- 상황에 따라 변경</li> <li>- 돈분 슬러리: 주 1~2회 위탁 (익산시 가축분뇨 공공처리시설, 7~15 t/회)</li> </ul>
분뇨관리 시설용량	<ul style="list-style-type: none"> <li>- 돈분 저장조: 280 t, 깊이 3.5 m, 위탁지점 ⇒ 모니터링!</li> <li>- 돈분 저장조 입구 넓이: 12m<sup>2</sup></li> <li>- 퇴비화시설: 非사용</li> </ul>
기타 시설 및 분뇨관리 특이사항	<ul style="list-style-type: none"> <li>- 돈분 저장조의 입구 개방 ⇒ 악취발생</li> <li>- 전반적으로 개방된 돈사와 분뇨저장조로 인한 악취가 심각</li> <li>- 퇴비화시설은 한우분뇨 저장조로 사용</li> <li>- 왕궁면 축산단지에 대한 특별관리기간으로 2021년 9월 축사 판매 (폐업) 예정</li> </ul>



돈사 형태 (개방형, 원치커튼, 슬러리)



돈사 외부 (지붕배기)



원분뇨 저장조 (콘크리트 매립)



액상분뇨 저장조 (우), 액비저장조 (좌)

그림 3-7. B 농가 분뇨관리시설 모습

○ B 농가의 분뇨 성상

B 농가의 분뇨는 슬러리 피트에서 돈분저장조로 이송된 후 특별한 처리없이 바로 위탁시설로 수거되어 정화처리되어지며, 저장된 분뇨의 성상은 A 농가에 비해 대부분 높게 나타났다. 이는 분뇨 내 고형물 및 유기물, 비료물질의 성상이 높게 나타나는 육성/비육돈만 사육하는 사육현황과 급여하는 사료의 영향 등 농가의 사양 특징 때문으로 예상된다. 여름철과 가을철의 분뇨 성상을 비교한 결과 고형물 및 유기물, 비료물질 모든 항목에서 유사한 농도를 나타내었으며, 분뇨 물성 모니터링 인자의 측정 결과와 보다 조밀한 샘플링 주기를 가져 저장·수거기간에 따른 분뇨 물성변화에 대한 비교·분석이 필요할 것으로 판단되었다.

표 3-9 B 농가의 돈분저장조 내 분뇨 성상

B 농가		TS	TVS	TSS	TVSS	NH <sub>4</sub> -N	T-N	O-P	T-P	COD	BOD
		g/L				mg/L					
여름철	돈분	49.2	38.7	41.9	36.8	2,350.0	3,950.7	320.3	737.9	78,832.2	20,880.0
	저장조	± 1.6	± 1.4	± 1.6	± 1.4	± 13.0	± 37.0	± 3.1	± 28.2	± 646.5	± 509.1
가을철	돈분	47.2	33.6	40.4	31.8	3,409.7	5,328.4	49.9	852.0	68,170.1	-
	저장조	± 1.1	± 0.9	± 1.0	± 0.8	± 15.2	± 15.4	± 0.5	± 4.6	± 1,393.0	-

## 2. 돈분 저장조 내 분뇨 발생량 & 환경관리 측정인자 개발

### 2-1. 돈분 저장조 내 분뇨의 적정 수거주기를 판단할 수 있는 모니터링 인자 설정

돼지농가의 분뇨 저장조가 가득 찼을 시 가축분뇨 위탁처리업체에서 수거하는 것은 경제적으로 가장 바람직한 방향이지만 강우나 홍수로 인한 가축분뇨 저장량의 급격한 증가나 업체 운반 차량의 사정으로 인해 해당 시점에 분뇨를 수거하지 못한다면 수거되지 못한 가축분뇨가 환경으로 유출되어 오염원으로 작용할 수 있다.

원분뇨 저장조 내 분뇨의 적정 수거주기를 판단하기 위해서 가축분뇨의 발생량을 직접적으로 확인할 수 있는 모니터링 인자로써 수위가 있으며, 수위는 저장조의 용적 대비 높이를 이용하여 % 값을 제시함으로써 분뇨의 높이 혹은 저장조 내 분뇨의 비율 등을 나타낼 수 있다. 양돈분뇨의 발생량과 저장량을 직접적으로 측정하는 수위 이외에 저장조 내 체류하고 있는 분뇨의 성상 변화로 인한 위탁처리 시설에서의 고액분리 불량, 악취물질 및 온실가스 발생, 생물학적 이용가능 유기물의 감소로 인한 바이오가스 회수 효율 저하 등 분뇨의 저장이 환경적으로 미치는 영향을 최소화하기 위한 보조 모니터링 인자의 개발이 필요하다.

따라서, 선행 문헌조사를 통해 저장조 내 분뇨의 성상 변화와 환경적 영향을 적절히 대변할 수 있는 모니터링 인자를 선정하였다.

#### (1) 돈분의 저장에 따른 성상 변화

슬러리 돈사에서 배출되는 분뇨의 경우 슬러리 피트에서부터 저장조까지 최소 7일에서 최장 6개월까지 저장되며, 저장조의 표면을 제외하고 혐기적인 상태가 지속됨에 따라 분뇨 발생 시와는 다른 특징을 나타낸다 (박 등, 2010).

슬러리 피트 혹은 저장조 내 양돈분뇨는 저장기간이 길어짐에 따라서 상층에 있는 고형물이 바닥으로 침전하면서 표면층, 중간층, 바닥층으로 나뉘어 각기 다른 고형성분을 가진 층이 형성된다고 보고되었다 (Arogo 등, 2000). 또한 혐기상태에서 여름철의 높은 온도로 인해 저장조의 온도 또한 영향을 받으며, 높은 온도에서 미생물의 활성이 높아짐에 따라 돈분 슬러리 내 고형물의 가수분해가 발생하게 되고, 부유성 고형물의 양을 증가시킨다. 박 등 (2010)에 따르면 돈분의 저장형태에 따른 유기물 및 pH의 거동을 관찰했을 때, 개방형 저장조에서 돈분의 표면에서는 호기성 미생물의 산화작용이 발생하고, 하부에서는 혐기성 미생물에 의한 유기물 분해작용이 일어나 유기물이 감소한다고 보고하였다. 또한 pH의 경우 돈분의 저장기간에 따라 유기물의 분해 산물로 발생하는 CO<sub>2</sub>가 외부로 배출되는 개방형 저장조에서 밀폐형 저장조보다 pH의 상승이 급격하게 나타났다. 따라서 저장기간이 길어지면 길어질수록 돈분 내 유기물의 손실이 발생하고, 온실가스의 배출이 증가할 것으로 예측된다. Mansyur & Choi (2012)에 따르면 총 60일간 돈분 슬러리의 저장기간에 따른 분뇨의 물성을 분석한 결과, pH, TS, VS, 인산염이 유의적으로 증가한 반면, EC, T-N, NH<sub>4</sub>-N, tCOD, sCOD는 유의적으로 감소했다고 보고하였다.

종합하면 온도, 저장조의 유형 등 다양한 요인이 영향을 미치나 돈분의 저장기간에 따라서 주로 분뇨 내 고형물의 변화, 유기물 분해, pH, EC의 변화가 나타나며, 이로 인해 돈분의 자원화 혹은 정화처리 시 처리과정 (특히 고액분리)에 영향을 미치는 것으로 판단된다.

#### (2) 돈분의 저장에 따른 악취물질의 배출

가축분뇨에서 발생하는 악취의 원인은 미생물의 활동에서 기인하며, 돈사 내 대표적인 악취물



질로는 NH<sub>3</sub>와 황화합물 (H<sub>2</sub>S methyl mercaptan), 휘발성유기산이 있다. 악취는 주로 미 소화 된 영양소, 돼지 체내 분비물, 장내 미생물의 대사 중간생산물 및 최종 생산물이 혼합된 분뇨가 배설될 때 함께 배출된 미생물의 활동에 의해 발생한다 (Rappert & Muller, 2005).

분뇨의 혼합 시 뇨 중의 요소가 분 내 urease의 작용으로 인해 분해가 가속되어 저장 1일 만에 NH<sub>3</sub>가 최대로 발생하기 때문에 슬러리 돈사에서 암모니아의 발생 정도는 매우 크다 (환경부, 2008). Hoff 등 (2012)에 따르면 슬러리 피트 내 분뇨 제거 전, 중, 후를 기준으로 NH<sub>3</sub>와 H<sub>2</sub>S의 농도는 분뇨의 제거 중 가장 높은 수준을 나타므로, 분뇨의 이송-저장-수거 과정에서의 악취물질 발생량을 확인·비교할 필요가 있다. Sommer 등 (1993)에 따르면 저장조 표면에 스크럼으로 인한 지각이 발달될 시 돈분 표면에 지각이 생성되지 않을 시에 비해 돈분 내 NH<sub>3</sub>의 손실이 20% 낮았는데, 이는 저장되는 분뇨의 특성에 따라서 NH<sub>3</sub> 발생량에서 차이가 생길 수 있음을 나타내는 결과로 분뇨의 성상과 NH<sub>3</sub> 배출량의 관계를 확인할 필요를 암시하는것이라 할 수 있다. 분뇨가 혐기적 조건에서 장기간 체류할 때 주로 생성되는 H<sub>2</sub>S은 황산염의 환원, 황함아미노산의 분해가 주요 원인이며 (윤 & 문, 2019), NH<sub>3</sub>에 비해 발생량이 적으나 낮은 최소감지농도를 가지고, 악취 강도가 높기 때문에 돈사 내에서 낮은 농도로 유지되어야 한다. 윤 등 (2019)이 피트 내 돈분을 단순저장 했을 때 악취물질의 변화를 관찰한 결과 복합악취는 총 저장기간인 7주차까지 꾸준히 발생량이 증가하였으며, 특히 4주와 5주 사이에 급격하게 배출되었다. 또한 황화합물의 경우 Methyl mercaptan의 배출량은 저장 5주차까지 꾸준히 상승하다가 이후 완화되었고, H<sub>2</sub>S는 저장기간이 길어질수록 배출량이 늘어나는 경향을 보였는데, 미생물군집을 비교한 결과 황환원균 (sulfate reducing bacteria)인 Desulfovibrio의 우점도가 꾸준히 증가하는 것으로 분석되었다.

종합적으로 돈분의 성분 및 저장기간 등에 따라 악취물질의 배출량은 상이하며, 발생하는 악취물질은 돼지농가의 운영 환경에 영향을 미치는 요인으로 실제 농가에서 돈분 이송-저장-수거 시 발생하는 악취물질을 돈분 발생량과 돈분의 성상, 수거주기 등과 함께 고려하여 적정 수거시점 도출이 필요하다.

### (3) 돈분 저장조 모니터링 인자의 선정

#### ○ 돈분의 성상 모니터링

돈분이 저장조 내에서 체류하는 동안 돈분 내에서는 다양한 생물학적 반응이 발생한다. 저장기간 중 돈분의 성상변화를 관찰하기 위해 다양한 계측장비를 통해 주기적으로 데이터화 하는 것이 직접적인 방법이나, 고농도의 고형물을 지닌 분뇨의 이화학적 성상을 측정하기 위해 고액분리, 열처리 등 부수적인 전처리 장비가 필요하고, 장소, 비용적인 부분이 제한되기 때문에 모니터링 설비를 설치하기 간단하고, 유지·관리가 쉬우며, 모니터링 패턴의 변화를 이용하여 분뇨 내 성상을 간접적으로 파악할 수 있는 모니터링 인자의 선정이 필요하다.

돈분의 수소이온 농도를 확인할 수 있는 인자인 pH는 저장조 내 돈분의 생물학적 반응에 중요한 영향을 미치며, 생물학적 반응에 따른 돈분 성상의 변화를 감지할 수 있는 인자로 활용된다. 저장기간동안 돈분은 표면을 제외한 부분에서 혐기적인 상태를 유지하며, pH가 중성인 상태에서 분뇨 내 유기물질의 가수분해가 이뤄진다. 가수분해 및 유기물의 분해 산물로 발생하는 CO<sub>2</sub>는 돈분에서 공기중으로 배출될 시 pH는 상승하게 된다. 가수분해 반응의 경우 온도가 중요 영향인자로 작용하기 때문에 계절에 따른 돈분 내 성상 변화를 pH를 이용하여 간접적으로 확인할 수 있을 것으로 판단된다.

돈분 내 어떤 물질간 반응은 서로 전자를 잃거나 전자를 얻는 방법이며, 이런 물질의 산화환원

력을 나타내는 지표가 oxidation-reduction potential (ORP)이다. ORP를 이용하여 산화제나 환원제의 농도를 확인할 수는 없으나, 이를 이용하여 물질의 화학반응을 예측할 수 있다. 일반적으로 생물학적 공정 운전 시 호기적 조건에서는 유기물의 산화 및 질산화 반응에 의해 ORP가 증가하나, 혐기적 조건에서는 ORP가 감소한다.

pH와 함께 수질특성을 파악하기 위해 주로 사용되는 전기전도도 (electrical conductivity, EC)는 물에 용해된 물질의 양을 나타내며, 특히 물질 내 함유된 총용존 고형물 (total dissolved solids, TDS)과 상호연관관계를 가지고 있다. 분뇨 내 EC에 주로 영향을 미치는 이온은  $\text{NH}_4$ , Na, Ca, Mg, K, Cl,  $\text{SO}_4^{2-}$ ,  $\text{CO}_3^{2-}$ 이며, 이중 체류 시간에 따른 정상변화를 보여주는  $\text{NH}_4$ ,  $\text{SO}_4^{2-}$ ,  $\text{CO}_3^{2-}$  등의 이온의 변화를 감지할 수 있을 것으로 판단된다.

돈분의 저장기간 중 분뇨 내는 혐기적인 상태로 유지되며, 일반적으로 강제적인 공기의 공급이 이루어지지 않으면 분뇨의 표면 외에 분뇨 내부 용존산소(dissolved oxygen, DO)의 변화가 없기 때문에 DO를 모니터링 인자로 선정하지 않았다.

#### ○ 돈분 저장조에서의 악취물질 모니터링

돈분 저장조에서 발생하는 악취는 미소화된 영양소, 장내 미생물의 중간대사물 및 최종 배설물을 대상으로 한 미생물의 반응, 체 내 분비물 등으로 인해 발생하게 되며, 약 168 가지 이상의 다양한 악취물질이 발생한다 (Mackie 등, 1998). 다양한 악취물질 중 최소감지농도가 높은 아민류, 방향족 질소화합물, 황화합물은 축산시설에서 발생하는 악취에 대한 기여도가 높으며, 이에 따른 주요 악취물질은  $\text{NH}_3$ ,  $\text{H}_2\text{S}$ , methyl mercaptan, dimethyl sulfide, demethyl disulfide로 환경부 (2008)에 따르면  $\text{NH}_3$ ,  $\text{H}_2\text{S}$ 가 핵심 악취물질로 두 물질이 상대적으로 고농도로 발생하는 것으로 조사된바 있다.

$\text{NH}_3$ 의 주요 영향인자는 온도, pH, 공기의 흐름 등이며, 이로 인해 대기로 방출된다 (Aarnick, 1997).  $\text{H}_2\text{S}$ 는  $\text{NH}_3$ 에 비해 발생량은 적으나 최소감지농도가 높고, 썩은 계란 냄새 등 높은 악취강도를 나타낸다. 황함유 단백질의 분해에 의해서 주로 발생하며, 낮은 pH 조건에서 amino acid decarboxylation이 진행되어 방출된다 (환경부, 2008; 김승수, 2017).

따라서 본 연구에서는 대표적인 분뇨 유래 악취물질인  $\text{NH}_3$ ,  $\text{H}_2\text{S}$ 를 모니터링하여 돈분 저장조에서의 분뇨 정상변화에 따른 배출 경향을 파악하고, 악취물질로 인한 환경영향을 감소시킬 수 있는 분뇨 수거시점을 선정하기 위한 연구를 진행하였다.

### 2-2. Lab-scale 돈분 저장조 이용 시뮬레이션 실험

#### ○ 돈분 저장조 내 분뇨의 물성 변화 실험 디자인

일시에 많은 양의 분뇨가 유입되는 돈분 저장조 내 분뇨의 저장기간에 따른 물성 변화를 확인하기 위해 Lab-scale 시뮬레이션 시험을 수행하였다.

실험을 위해 lab-scale 돈분 저장조 (Depth 1.04 m, Area 0.058  $\text{m}^2$ , 60 L)를 제작하였으며 (그림 3-8), 돈분의 배출주기는 1회/주, 분뇨 발생량은 표준축사설계도 (농림축산식품부, 2016) 내 비육돈사의 표준사육밀도 (0.9  $\text{m}^2/\text{두}$ )와 돼지의 배출원단위 (5.1 L/일/두)를 이용하여 산출한 2.3 L/주를 기준으로 실험을 수행하였다.

- 실험은 총 84일 수행하였으며, 사용된 돈분의 평균 성상은 표 3-8에 제시하였다.

- 물성변화 분석항목 : Total solids (TS), Total volatile solids (TVS), Total suspended

solids (SS), Total volatile suspended solids (VSS), Soluble total organic carbon (STOC), Total nitrogen (T-N), Ammonium nitrogen (NH<sub>4</sub>-N), Nitrite & nitrate (NO<sub>x</sub>-N)

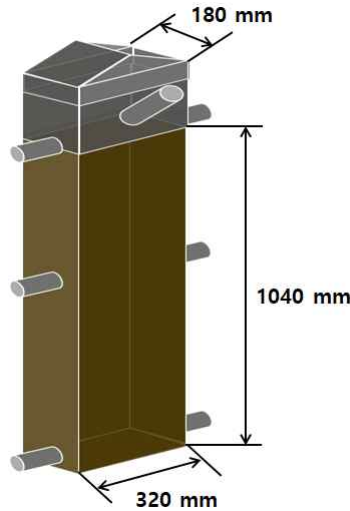


그림 3-8. Lab-scale 돈분 저장조 디자인

표 3-10. B 농가의 돈분저장조 내 분뇨 정상

Parameters	Concentration (mg/L)
TS	46,179.2 ± 22,581.9
TVS	33,227.1 ± 17,742.2
TSS	39,567.5 ± 24,987.0
TVSS	30,565.0 ± 19,152.8
STOC	12,305.8 ± 5,039.9
NH <sub>4</sub> -N	2,719.3 ± 522.8
T-N	5,187.4 ± 2,015.2

○ 돈분 저장조 내 분뇨의 물성 변화 실험 결과

주별 유입 돈분과 돈분 저장조 내 체류하는 분뇨의 성상을 분석하여 돈분 저장조 내 체류시간에 따른 물성변화를 나타냈다.

유입 돈분 내 고형물 함량은 변이가 컸으며, 지속적인 돈분의 유입으로 인해 저장조 내 분뇨의 고형물과 유기물 농도는 초반 21일까지는 감소, 정체 후 꾸준히 우상향하였으며, 각 항목별 증가율은 TS 1,250.3 mg/L/d, TVS 915.0 mg/L/d, TSS 1,371.8 mg/L/d, TVSS 976.4 mg/L/d로 나타났다.

21일까지의 고형물 함량 감소는 초반에 유입된 분뇨가 시간이 지남에 따라 침전으로 고형물이 돈분 저장조의 바닥부근으로 가라앉았기 때문으로 판단된다.

양돈 현장에서 발생하는 악취물질의 대부분은 유기물이 혐기적으로 분해되면서 발생하며, 돈분의 유입으로 인해 우상향하는 유기물 농도로 인해 돈분 저장조 내 분뇨의 장기체류는 필연적으로 악취발생을 수반할 것으로 판단된다.

분뇨 내 용해되어 있는 유기물인 STOC 농도는 실험 시작 이래로 급격히 감소하다가 35일 시점부터는 일정하게(매우 완만한 감소) 유지되는 경향을 나타냈었는데, 이는 고형물에 포함된 유기물에 비해 미생물이 STOC를 이용하기 용이하여, anaerobic digestion 반응으로 혐기성 미생

물에 의한 유기물의 감소가 발생했기 때문인 것으로 판단된다.

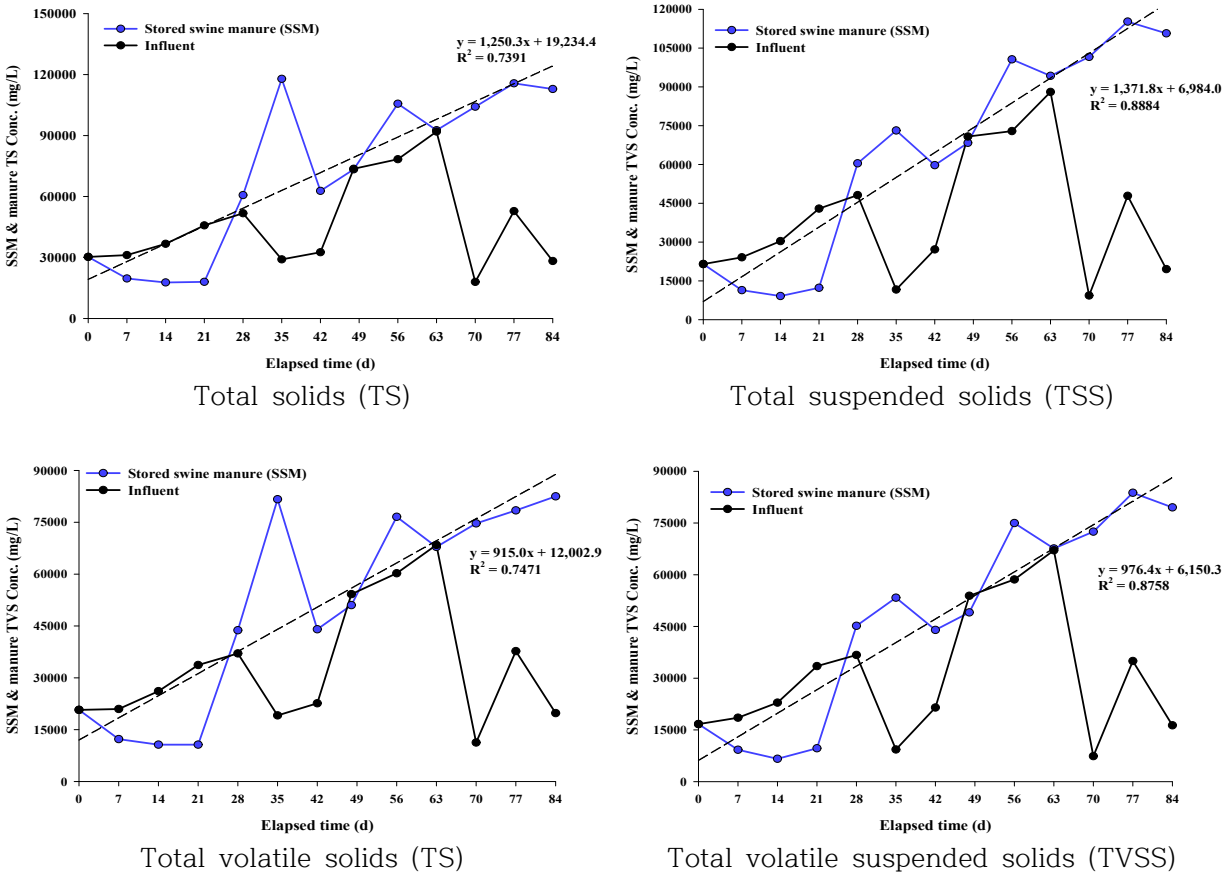


그림 3-9. 돈분의 저장시간에 따른 고형물 및 유기물 농도 변화

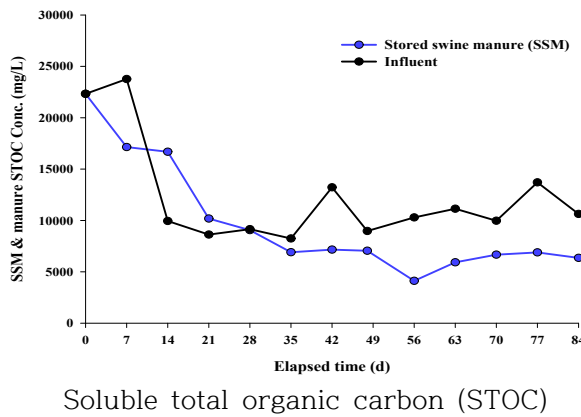
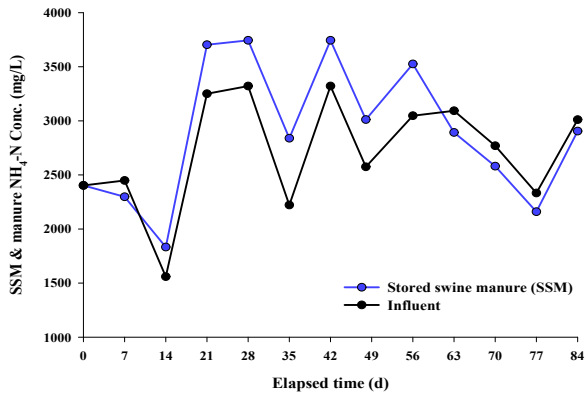


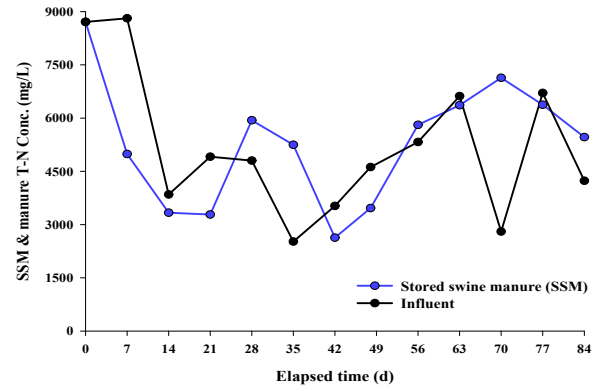
그림 3-10. 돈분의 저장시간에 따른 STOC 농도 변화

실험 시작 21일까지의 고형물 함량 감소는 고형물에 영향을 받는 T-N 농도 감소를 유발하였으며, 이는 시간이 지남에 따라 고형물이 돈분 저장조의 바닥부근으로 가라앉았기 때문으로 판단된다. 또한 유입된 분뇨 내 T-N 농도가 증가하고 돈분 저장조 내 고형물의 농도가 상승함에 따라 저장조 내 분뇨의 T-N 농도 또한 증가하는 경향이 나타났다.

암모니아성 질소의 경우 대체로 유입되는 분뇨 내 암모니아성 질소 농도의 변화 경향을 따라 갔으며, 약 60일경 유입 돈분의  $\text{NH}_4\text{-N}$  농도가 저장조 분뇨 내  $\text{NH}_4\text{-N}$  농도를 역전하는데 이는 분뇨 저장기간 중 유기물의 분해로 인한 pH 상승, pH 상승에 따른 암모니아 탈기 상승에 의한 것으로 판단된다.



Ammonium nitrogen (NH<sub>4</sub>-N)



Total nitrogen (T-N)

그림 3-11. 돈분의 저장시간에 따른 질소화합물 농도 변화

### 3. 저장조 모니터링 인자별 적정 분뇨수거시점 인지 및 센싱 방법 제시

#### 3-1. ICT 기반 저장조 내 돈분 물성 및 저장량 변화, 악취물질 발생량 모니터링

##### (1) 테스트베드 농가 내 돈분 저장조 모니터링시스템 구축

현장조사 시 테스트농가 A와 B의 저장조는 직육면체의 콘크리트 매립형 구조를 가지고 있으며, 상단 일부가 뚫린 곳으로 분뇨를 유출시키는 개방형의 저장조 토출구를 가지고 있었다. A 농가는 매립된 내부의 저장조의 표면적인 100 m<sup>2</sup>에 비해 현저히 좁은 약 2m<sup>2</sup> (저장조 전체 표면적의 2%) 토출구를 가지고 있으나, B 농가는 저장조의 표면적 80m<sup>2</sup> 중 12m<sup>2</sup>가 그대로 개방되어 있어 저장조의 표면관찰과 외부 오염원 (비, 먼지, 곤충 등)이 유입되기에 쉬운 조건이었다. 따라서 저장조 내 분뇨의 물성과 발생하는 악취물질을 측정하기 위해 각 농가의 입구를 밀폐하는 하우징 작업을 실시하였으며, 각각의 센서와 컨트롤 박스도 함께 설치하였다.

각 농가의 저장조 토출구 형태가 상이했기 때문에 하우징 형태에 따른 센서의 설비 위치도 조정하였다. A 농가는 저장조의 입구가 좁은 형태로 입구 자체를 추가적인 공간 없이 막을 수 있는 덮개를 설치하였으며, B 농가는 저장조의 입구가 넓고 표면이 불균일하며 비가림 시설이 있어 단순한 덮개 형태로는 입구를 막기 어려웠기 때문에 비가림 시설의 구조를 활용하여 저장조 입구 상부에 일정한 공간을 두고 덮을 수 있는 시설을 설비하였다.

분뇨 저장조 내부의 분뇨 물성과 수위를 확인하기 위해 협동연구기관인 동문이엔티에서 선정한 pH, ORP, EC, 수위 센서를 설치하였으며, 저장조 내 분뇨에서 발생하는 악취물질인 NH<sub>3</sub>, H<sub>2</sub>S를 측정하기 위해 각 테스트베드에 센서를 구축하였다. A 농가는 저장조 토출구 측면에 토출구 덮개 옆으로 철판을 설치해 그 위에 센서를 설치하였으며, B 농가는 비가림 시설 내부에 브라켓을 연결하여 각 센서를 위치시켰다 (그림 3-12).

각 센서에서 측정된 분뇨의 물성과 악취 농도는 저장조 측면에 설치한 컨트롤 패널에 실시간으로 표시되도록 하여 사용자가 현재 분뇨의 상태를 관찰할 수 있도록 하였으며, 측정된 데이터는 서버의 용량을 고려하여 1시간 간격으로 내부의 저장소에 저장되었다.

### A 농가



저장조 전경



저장조 덮개 및 센서 설비 모습

### B 농가



저장조 전경



센서 설비 모습

그림 3-12. 테스트베드 농가 돈분 저장조 내 설치된 센서와 컨트롤 박스의 모습

2주 간격으로 농가를 방문하여 각 농가별 저장조 분뇨샘플링 하였으며, 저장조 내 분뇨 물성을 분석하여 계절의 변화에 따른 분뇨물성의 변화와 농가별 분뇨 농도변이를 관찰하였다. 모니터링되어 확보된 각 데이터 (pH, ORP, EC, NH<sub>3</sub>, H<sub>2</sub>S, 수위, 온도)는 해당 시간대의 데이터를 동시에 비교하여 분뇨의 유입·유출, 체류에 따른 각 인자들의 변화를 분석함으로써, 분뇨의 수거시점 선정에 적합한 인자를 선정하고자 하였다.

#### (2) 저장조 내 돈분 물성 변화

코로나 및 ASF의 발생으로 인해 농가의 선정 및 센서 설비에 시일이 소요되었으며, 2021년 3월부터 각 농가의 분뇨 저장조의 모니터링을 실시하였다. 모니터링되는 분뇨 물성과 악취물질 농도와 연결하여 실제 저장되는 분뇨의 물성을 주기적으로 분석하였다.

A 농가는 돈사 슬러리피트에서 저장되어있던 분뇨를 농장주의 판단에 따라 일시에 원분뇨 저장조로 유입한 후 농가에 설치되어있는 스크류 타입 고액분리기를 활용하여 고액분리하고, 큰 고형물들이 제거된 상태의 액상분뇨가 분뇨 저장조로 유입되는 분뇨 저장 형태를 가지고 있다. 본 연구에서 모니터링을 실시한 저장조는 실제 공동자원화시설에서 분뇨를 수거해가는 가장 마지막 저장조로 전단에 액상분뇨를 저장하는 1차 저장조가 존재하며, 1차 저장조의 분뇨가 차 일정 수준이상으로 수위가 올라가면 1차 저장조 상단에 최종 저장조와 연결되어있는 관을 통해서 최종 저장조로 유입되도록 설계되어 있다. 따라서 1차 저장조를 거치면서 고액분리 후 남아있는 입자 큰 고형물들이 가라앉고, 침전으로 고형물이 한층 낮아진 액상분뇨가 최종 분뇨 저장조로 유입될 것으로 판단하였다.

A농가의 분뇨 저장조의 모니터링은 3월부터 10월까지 수행되었으며, 이 시기 계절별 분뇨 성

상 분석결과, 봄철의 분뇨가 타 계절의 분뇨에 비해 높은 농도를 보이는 것으로 나타났다. 타 계절에 비해 봄철의 분뇨는 고형물이 높은 특징을 나타내는데 이는 두 가지 측면에서 이유를 찾을 수 있다. 첫 번째로 평균 기온이 높아 돼지들의 음수량이 높은 여름철과 가을철은 그만큼 분뇨의 농도도 희석되며, 상대적으로 봄철 분뇨의 고형물을 비롯한 다른 성상들이 높을 수 있다. 본 연구에서는 저장조 내 분뇨를 대상으로 환경영향을 분석했기 때문에 돼지의 사료섭취량과 음수량을 확인하지는 못하였으나, 오 등 (2017)에 따르면 높은 환경온도는 돼지의 물 섭취빈도가 증가하였으며, 음수량을 증가시켜 돼지의 사료섭취욕구 저하를 유도한다고 보고한 바, 배설되는 분뇨의 성상에 여름철의 높은 온도가 봄철에 비해 영향을 미쳤을 가능성이 높다고 판단된다. 또한 국립축산과학원 (2009)에 따르면 음수량이 증가함에 따라 뇨 배설량이 증가하는 경향을 보여, 증가된 뇨 배설량으로 인한 저장조 분뇨의 성상 변화가 영향을 미친 결과로 사료된다.

두 번째로는 분뇨의 수거와 액비 생산시기로 봄철 본격적인 식물의 생육에 앞서 시비되는 가축분뇨 액비 생산을 위한 분뇨량을 확보하기 위해 많은 양의 분뇨를 수거해가기 때문에 저장조 내 수위가 일시에 감소하면서 샘플링 시 저장조 하부에 침전되어 있던 고형물이 함께 회수되었을 가능성이 있다. 분뇨 저장조 수위 모니터링 결과 봄철 수위가 급격하게 낮은 지점과 일치하는 것으로 볼 때 저장조 수위에 의한 침전 고형물의 영향이 크게 작용했을 것으로 예상된다.

전체적으로 봄철의 고형물 (TS, SS)과 유기물 (VS, VSS, COD), TN, TP의 함량은 높게 나타났으나 용존되어 있는 물질의 성상을 나타내는 NH<sub>4</sub>-N, OP의 성상은 전체 모니터링 기간동안 크게 변화하지 않는 것으로 나타났다. 이는 다량의 고형물이 유입되거나 저장조에 체류하는 중에 비용해성물질이 용해성 물질로 변화하는 미생물 반응이 충분히 일어나기 어려운 조건이었기 때문으로 예상되며, 저장조 내 분뇨를 장기 체류시키는 것이 아닌 주기적으로 공동자원화시설에 위탁하고 다시 저장조로 분뇨를 유입하는 분뇨 순환구조 상 용존물질로 변화할 수 있는 충분한 시간이 부족했을 것으로 판단된다.

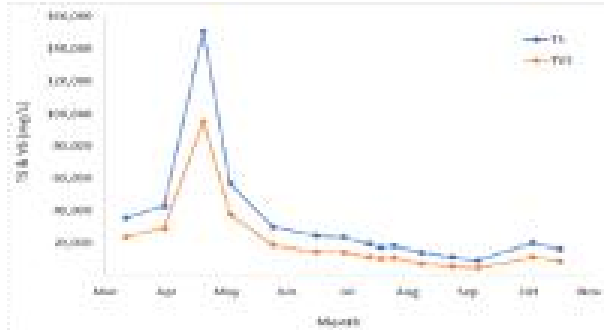
선행연구에 따르면 분뇨 내 NH<sub>4</sub>-N의 농도는 NH<sub>3</sub>의 농도와 양의 상관관계를 가지는데, 전체 계절동안 급격한 변화없이 유지되는 NH<sub>4</sub>-N 농도를 볼 때, 저장조 내 악취 발생 시 NH<sub>3</sub>의 농도는 당시 분뇨의 성상보다는 저장된 분뇨의 양과 주변의 온도, 유입·유출 등으로 인한 저장조 분뇨의 유동에 의한 영향이 클 것으로 예상된다.

A 농가 분뇨의 평균 pH는 7.9로 봄철에 비해 여름과 겨울의 평균 pH가 8.1로 약간 높게 나타났다.

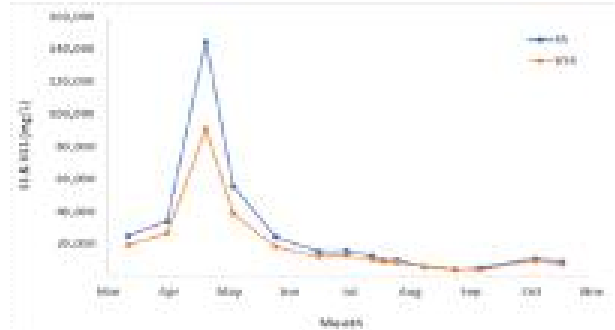
표 3-11. A 농가 저장조 내 분뇨의 평균적인 성상 및 계절별 분뇨 성상 변화

Parameters	Total	Spring (Mar-May)	Summer (Jun-Aug)	Autumn (Sep-Oct)
MC (%)	96.6 ± 3.6	93.7 ± 5.0	98.1 ± 0.5	98.5 ± 0.4
TS (mg/L)	33,920.1 ± 36,042.5	63,182.7 ± 50,065.5	18,473.5 ± 4,582.5	15,304.4 ± 5,298.4
TVS (mg/L)	21,293.6 ± 23,176.8	40,940.0 ± 31,171.8	10,941.4 ± 3,226.0	8,746.7 ± 3,361.0
SS (mg/L)	26,621.8 ± 36,625.4	56,615.6 ± 50,733.4	10,505.6 ± 4,401.2	8,346.3 ± 3,220.5
VSS (mg/L)	19,314.3 ± 22,617.1	38,678.9 ± 30,204.4	9,042.8 ± 3,178.3	7,001.9 ± 2,847.8
NH <sub>4</sub> -N (mg/L)	2,998.1 ± 320.9	3,008.2 ± 366.2	3,073.7 ± 293.0	2,854.0 ± 320.8
TN (mg/L)	4,620.9 ± 2,337.3	6,415.1 ± 3,223.6	3,616.9 ± 669.3	3,563.4 ± 940.3
OP (mg/L)	76.2 ± 38.0	115.7 ± 39.2	52.5 ± 7.9	58.4 ± 4.3
TKP (mg/L)	1,071.7 ± 1,843.4	2,149.2 ± 2,939.4	456.5 ± 249.4	412.6 ± 380.6
COD (mg/L)	41,201.9 ± 30,437.6	62,590.9 ± 35,973.0	20,980.4 ± 3,768.5	58,504.8 ± 32,152.4
STOC (mg/L)	5,224.2 ± 2,512.3	7,469.4 ± 2,862.1	3,831.6 ± 579.0	3,746.5 ±
pH	7.9 ± 0.4	7.6 ± 0.4	8.1 ± 0.2	8.1 ± 0.1

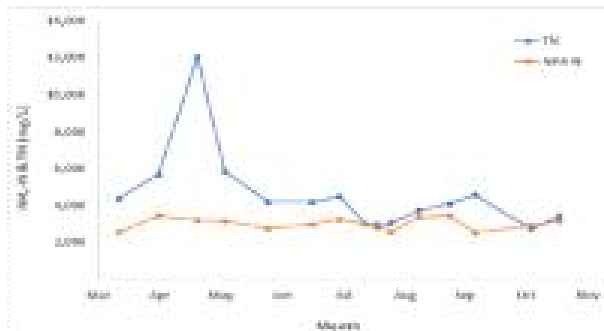




TS & VS 농도 변화



SS & VSS 농도 변화



NH<sub>4</sub>-N & TN 농도 변화



OP & TP 농도 변화

그림 3-13. A 농가 저장조 내 분뇨의 월별 정상 변화

B 농가는 돈사 슬러리피트에서 저장되어있던 분뇨의 수위를 고려하여 농장주가 일시에 분뇨 저장조로 유입하여, 슬러리피트 내 원분뇨 그대로 저장조로 유입되며, 슬러리피트 내 일정 높이에 저장조와 연결된 관이 있어 실시간으로 발생된 분뇨의 양에 따라 저장조로 유입되는 일반적인 자연유하식 분뇨 유입과 다른 형태를 가지고 있다. B 농가 내 저장조는 하나이며, 2~3일에 한번씩 공공처리시설에서 수거해가고 있다. 원분뇨가 그대로 유입됨에 따라서 고형물 농도가 높은 분뇨가 체류하고 있으며, 주기적인 위탁과 분뇨의 유입으로 인해 저장조 내 분뇨의 수위는 계속해서 변하지만 일정 수위가 계속 유지되고 있다.

B 농가의 분뇨 저장조의 모니터링은 3월부터 10월까지 수행되었으며, 계절에 따른 변화보다는 샘플링 시기마다 분뇨 성상의 변동이 크게 관찰되었다. 평균적으로 봄철에서 다른 계절에 비해 분뇨 성상이 높게 나타났으나 그 절대적인 수준이 A 농가에 비해 낮은데 이는 봄철 낮은 분뇨 저장량으로 인해 거의 저장조 바닥의 분뇨를 확보할 수 있었던 A 농가에 비해 일정 수위가 지속적으로 유지되는 B 농가는 유입된 분뇨 내 큰 고형물은 저장조 바닥에 가라앉은 상태로 반응조 전체를 교반 후 샘플링 할 수 있는 상황이 아니었기 때문에 상대적으로 고형물 농도가 낮은 중간층의 분뇨를 샘플링 해 나타난 결과로 판단된다. 또한 B 농가의 분뇨농도는 가을철 급격히 낮아졌는데, 이는 B 농가가 10월까지만 운영하고 농장을 폐업하는 시기였기 때문에 돼지사육두수 감소함에 따라 나타난 현상으로 해석할 수 있다.

A 농가도 마찬가지겠으나, 분뇨 내 다양한 고형물이 고액분리 없이 모두 유입되는 B 농가의 저장조 내부는 고형물의 질량에 따라 층이 나뉘져 있을 것으로 예상되며, 특정 샘플링을 통해서 저장조 내부 분뇨 전체를 대변하기는 사실 어렵다고 할 수 있다. 하지만 분뇨물성 인자들은 용존물질에 따라 반응하며, 악취의 경우 분뇨 표면에서 발생하므로 반응조 중간에서 샘플링한 결과를 분석에 활용하였다.

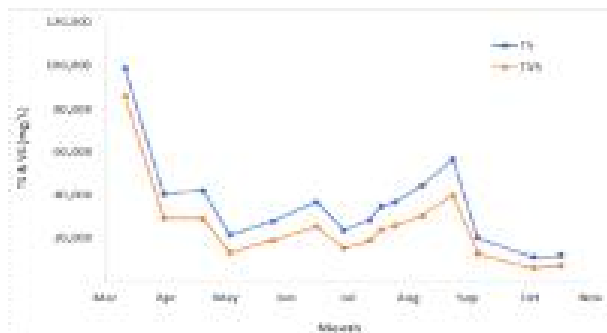
용존되어 있는 물질의 성상을 나타내는 NH<sub>4</sub>-N, OP의 성상은 TN, TP의 성상만큼 변이가 심하

지 않았으며, 점차 낮아지는 경향을 나타내었는데, 이는 돈사 내 음수량 증가와 폐업으로 인한 사육두수 감소에서 기인한 것으로 사료된다.

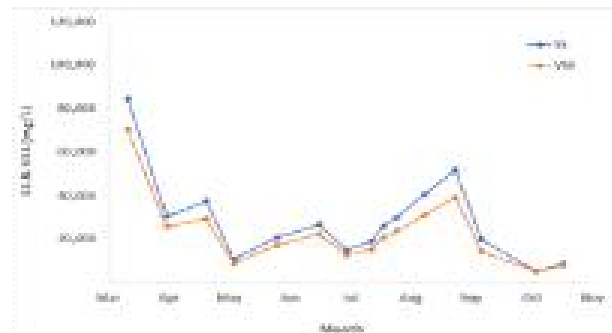
B 농가 분뇨의 평균 pH는 7.3으로 봄철 6.8, 여름철 7.5, 가을철 7.8로 나타났으며, 평균적으로 A 농가에 비해 낮은 수준을 보여, pH에 영향을 받는 NH<sub>3</sub>의 휘발이 A 농가에 비해 더 적게 나타났을 것으로 예상된다.

표 3-12. B 농가 저장조 내 분뇨의 평균적인 성상 및 계절별 분뇨 성상 변화

Parameters	Total	Spring (Mar-May)	Summer (Jun-Aug)	Autumn (Sep-Oct)
MC (%)	96.3 ± 1.8	95.7 ± 2.5	96.2 ± 1.1	98.6 ± 0.5
TS (mg/L)	36,311.0 ± 18,114.9	43,257.3 ± 24,633.7	37,241.5 ± 10,830.8	14,521.1 ± 4,750.1
TVS (mg/L)	25,699.1 ± 15,475.7	32,189.3 ± 22,299.6	25,621.4 ± 7,986.4	9,065.6 ± 3,517.7
SS (mg/L)	30,880.6 ± 23,046.1	39,644.4 ± 34,518.5	29,896.0 ± 12,647.6	11,161.1 ± 7,601.1
VSS (mg/L)	25,070.2 ± 19,800.9	33,370.0 ± 30,505.2	23,525.4 ± 9,120.9	9,060.2 ± 4,813.2
NH <sub>4</sub> -N (mg/L)	2,585.8 ± 487.0	3,007.3 ± 89.1	2,523.6 ± 341.1	1,813.2 ± 179.4
TN (mg/L)	4,015.9 ± 1,546.7	5,173.3 ± 1,359.2	3,775.9 ± 722.9	2,058.6 ± 1,595.8
OP (mg/L)	134.8 ± 167.3	250.7 ± 249.2	81.1 ± 34.9	40.0 ± 12.4
TKP (mg/L)	870.2 ± 492.6	725.1 ± 401.4	914.5 ± 618.0	777.3 ± 527.1
COD (mg/L)	50,972.3 ± 19,325.4	55,829.0 ± 13,971.0	54,693.5 ± 18,491.2	25,806.5 ± 23,266.4
STOC (mg/L)	13,164.9 ± 7,699.2	21,540.3 ± 4,467.4	9,837.5 ± 4,652.2	2,955.5 ±
pH	7.3 ± 0.5	6.8 ± 0.4	7.5 ± 0.4	7.8 ± 0.2



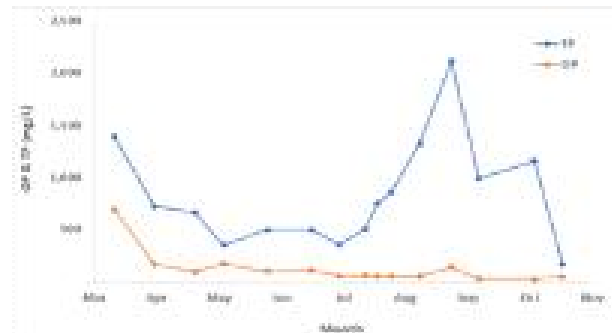
TS & VS 농도 변화



SS & VSS 농도 변화



NH<sub>4</sub>-N & TN 농도 변화



OP & TP 농도 변화

그림 3-14. B 농가 저장조 내 분뇨의 월별 성상 변화

(3) 분뇨 저장조 모니터링 결과 분석 (분뇨 물성 및 악취)

농가 분뇨 저장조에 설치된 센서들을 이용하여 분뇨의 적정 수거시점을 파악하기 위한 모니터링을 실시하였다. 각 농가별 모니터링 기간으로는 A 농가 5개월, B 농가 6개월로 전체 모니터링 기간 동안 중복되는 센서들의 값을 이용하여 분뇨의 적정 수거시점을 선정하고자 하였다.

(3-1) 분뇨 저장조 모니터링 기간 내 데이터 확보 및 현장 애로사항

실제 모니터링 결과, 현장에서 센서의 품질, 농가 내 센서 설치 상황과 농가의 분뇨 유입·수거 관행, 고농도의 분뇨 및 악취 발생으로 인해 센서의 수명이 극심하게 단축되는 결과를 확인할 수 있었다. 특히 A 농가의 경우 전체 모니터링 기간 중 EC와 NH<sub>3</sub> 센서를 제외한 나머지 센서들에서 측정값을 활용할 수 있는 기간이 약 33.3% 수준밖에 되지 않았으며, B 농가 역시 pH와 ORP는 35.6%로 절반 (50%)을 넘지 못해 확보된 데이터를 이용한 분석이 어렵다고 판단되었다 (표 3-13).

표 3-13. 모니터링 기간 중 각 센서별 데이터 활용가능 기간

농가	측정인자	데이터 활용가능 기간			
		측정일수 (데이터 포인트)	데이터 활용 일수 (데이터 포인트) 일	비율 (%) %	
A	분뇨 저장량	수위	150 (3,600)	100	
	분뇨 물성	pH	50 (1,200)	33.3	
		ORP	150 (3,600)	50 (1,200)	33.3
		EC	102 (2,448)	68.0	
	악취	NH <sub>3</sub>	51 (1,224)	34.0	
		H <sub>2</sub> S	77 (1,848)	51.3	
B	분뇨 저장량	수위	180 (4,320)	100	
	분뇨 물성	pH	64 (1,536)	35.6	
		ORP	180 (4,320)	64 (1,536)	35.6
		EC	148 (3,552)	82.2	
	악취	NH <sub>3</sub>	140 (3,360)	77.8	
		H <sub>2</sub> S	172 (4,128)	95.6	

농가 내 모니터링이 제대로 이뤄지지 않은 주된 원인은 센서의 고장, 수명 단축으로 이를 유발하는 세부적인 요인은 다음과 같다.

○ 분뇨 물성 (pH, ORP, EC) 센서

- ① pH, ORP, EC, 온도를 측정하는 센서는 세트로 결합되어 분뇨 저장조의 중간 높이에 설치되었으나, 분뇨의 연속적인 수거로 인해 수위가 계속해서 낮아짐에 따라 센서가 공기중에 노출되어 전극부분이 건조되는 현상이 발생 (A 농가)  
⇒ 농가 분뇨 수거관행으로 인한 센서 유지·관리 미비

② 이후 동일한 센서들을 다시 설치하고 모니터링한 결과 EC 센서를 제외하고 설치 후 정상적인 값을 읽지 못한채 다시 고장이 발생 (A 농가).

⇒ 센서 자체의 낮은 내구성

③ 고농도의 분뇨로 인해 센서가 정상적으로 작동하더라도 센서 주변으로 쌓이는 많은 고형물들과 오염물질들로 인해 약 2개월을 버티지 못하는 극한의 저장조 환경

⇒ 센서 자체의 낮은 내구성 & 고농도 분뇨로 인한 극한 환경

#### ○ 악취 (NH<sub>3</sub>, H<sub>2</sub>S) 센서

① 고농도의 분뇨로 인해 발생하는 다량의 악취물질은 센서의 수명을 악화시켜 악취 센서의 교체 주기를 정상 범위 (6개월 ~1년)보다 짧게 만듦 (약 3~4개월) (A&B 농가)

⇒ 고농도 악취 발생으로 인한 센서 수명 단축

② NH<sub>3</sub>의 경우 흡착성이 강한 물질로 저장조 내부에 설비된 악취센서의 센서부에 흡착되어 정확한 값을 읽지 못하고 실제보다 더 낮은 값을 나타내게 하며, 분뇨의 증발로 인해 생성된 수증기는 센서와 센서 내부의 회로를 부식시켜 센서에 회복불능의 고장을 유발 (A 농가)

⇒ 악취물질의 특성으로 인한 센서 민감도 저하 및 습기로 인한 센서 손상

위 문제들로 인해 실험 전과정의 농가 내 분뇨 저장조 모니터링 데이터를 확보할 수 없었으며, 특히 A 농가에서 모니터링 시 문제점이 많이 도출되었다. A 농가의 경우 저장조가 다수로 모니터링된 저장조가 실제 분뇨수거차량이 분뇨를 수거해가는 최종 저장조였기 때문에 분뇨의 수거가 활발한 겨울에서 봄철까지 수위가 급격하게 낮아져공기중으로 센서가 노출되어 센서의 수명에 영향을 미치게 되었다.

또한 악취센서의 경우 분뇨 내부에 위치해야 하는 분뇨 물성센서와 달리 저장조 표면 위에 위치해야하는데 A 농가는 분뇨 표면과 악취 센서의 설비 위치가 가깝고 저장조 입구의 밀폐로 인해 생성된 수증기가 센서 표면에 직접적으로 닿아 센서 부식의 원인으로 작용하였다. 이로인해 연속적으로 측정된 값이 부족하여 후단의 통계분석 시 A 농가의 데이터는 사용하지 않았다.

B 농가의 모니터링 결과는 확보된 데이터 수가 충분하다고 판단되어 통계분석을 위해 활용하였으며, 악취센서의 경우 중간의 센서 교체를 제외하고 거의 모든 모니터링 기간동안 데이터를 확보할 수 있었다. A 농가와 달리 B 농가의 분뇨 저장조는 입구를 덮개 형태가 아닌 입구의 상부에 추가적인 공간을 더 제공한 후 밀폐하는 형태의 하우징을 시공하였으며, 악취 센서와 분뇨 표면과의 거리도 충분히 이격하였기 때문에 센서의 상태가 측정에 용이한 상태로 오래 유지할 수 있었다.

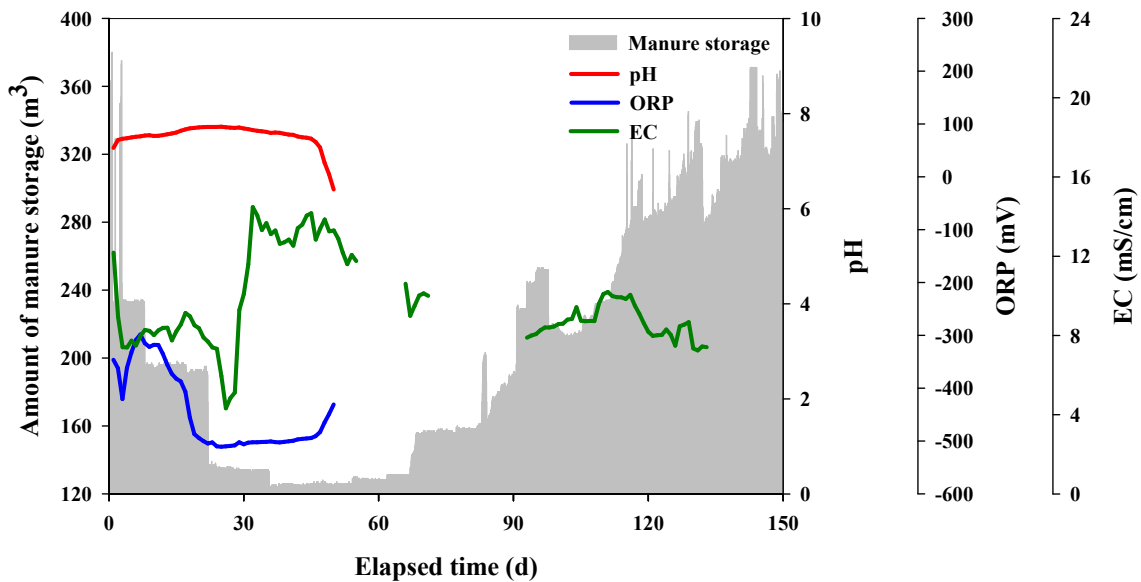
#### (3-2) 각 모니터링 인자별 측정 결과 분석 (전 기간 기준)

A 농가의 분뇨 저장조 모니터링 결과, 모니터링 초반 분뇨 수거로 인해 분뇨 저장량이 적어짐에 따라 EC는 급격하게 하락하는 경향을 나타내었으나, 분뇨 저장량이 낮게 유지되는 동안 갑작스럽게 상승하는 경향이 추가적으로 나타나 분뇨 저장량에 따른 EC 데이터의 일관성은 나타나지 않았으며, pH에서도 분뇨 감소에 따른 특이점은 관찰되지 않았다. 악취의 경우 분뇨 저장량이 일정하게 유지되는 구간에서도 NH<sub>3</sub>와 H<sub>2</sub>S의 농도의 변동이 심하게 나타났으나 분뇨 저장

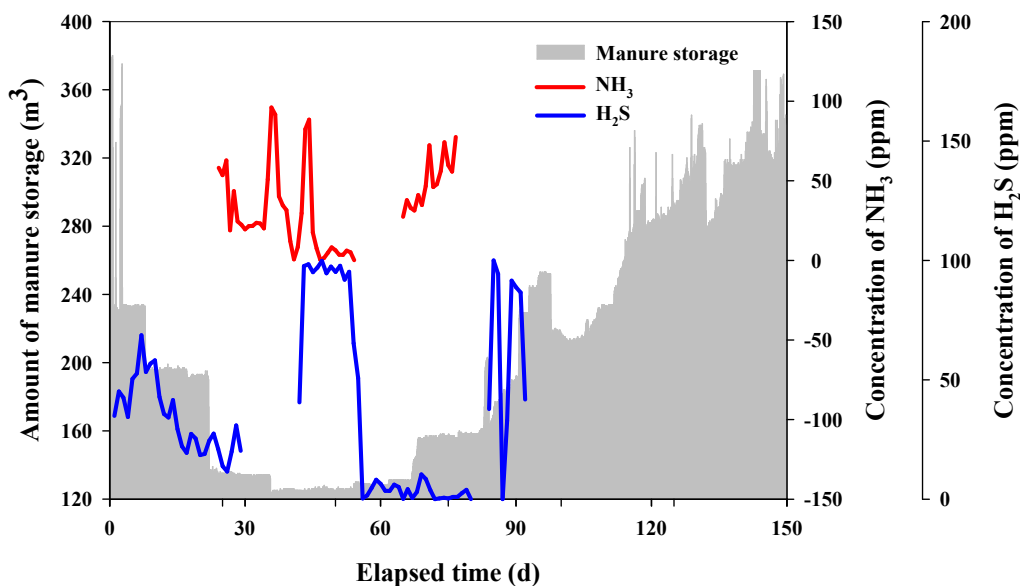
조의 환경조건에 따른 악취물질의 농도 변화를 관찰하기 어려웠다 (그림 3-15).

B 농가의 분뇨 저장조 모니터링 결과, 분뇨 물성 센서 중 가장 많은 모니터링 데이터를 지닌 EC에서 분뇨 저장량이 상승함에 따라 상승하는 경향을 나타냈다. 이후 분뇨의 수위가 변동성을 지니며 유지될 때 꾸준히 낮아지는 경향을 확인할 수 있었는데, 용액 내 전해질 이온이 많을수록 높아지는 EC 센서의 특성상 저장조 분뇨의 물성 분석을 통해 나타난 NH<sub>4</sub>-N과 O-P의 농도 감소 경향과 일치하는 결과라고 할 수 있다 (그림 3-16).

악취농도의 경우 분뇨 저장량이 많아짐에 따라 악취발생 농도도 증가하는 경향을 확인할 수 있었으며, 환경적 요인 중 외부 기온의 증가로 NH<sub>3</sub>와 H<sub>2</sub>S가 증가하는 경향을 확인할 수 있었다 (그림 3-16).

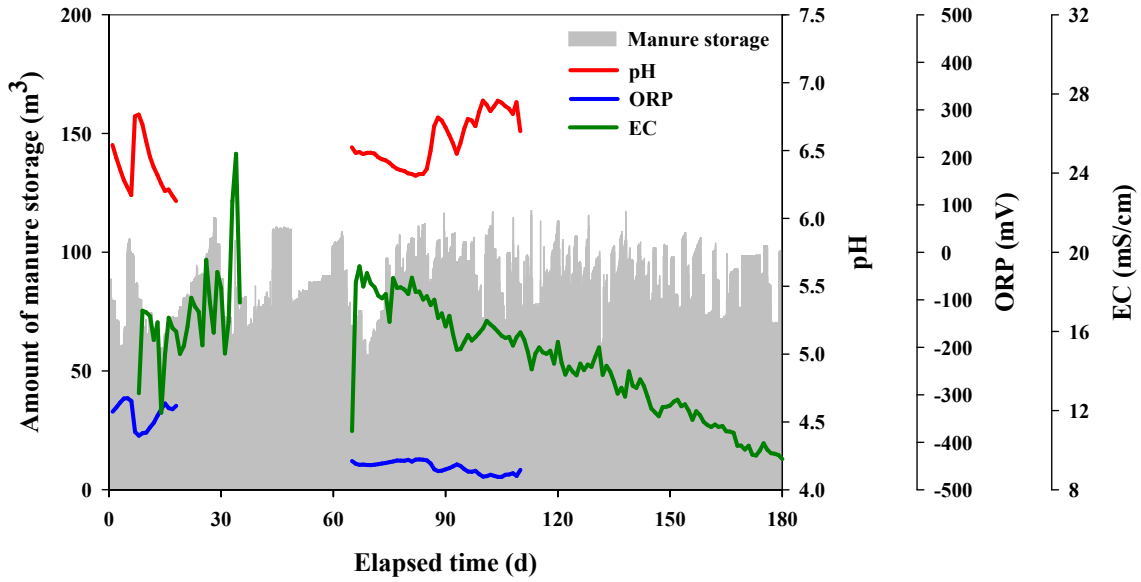


저장량 & 분뇨 물성 (pH, ORP, EC)의 변화

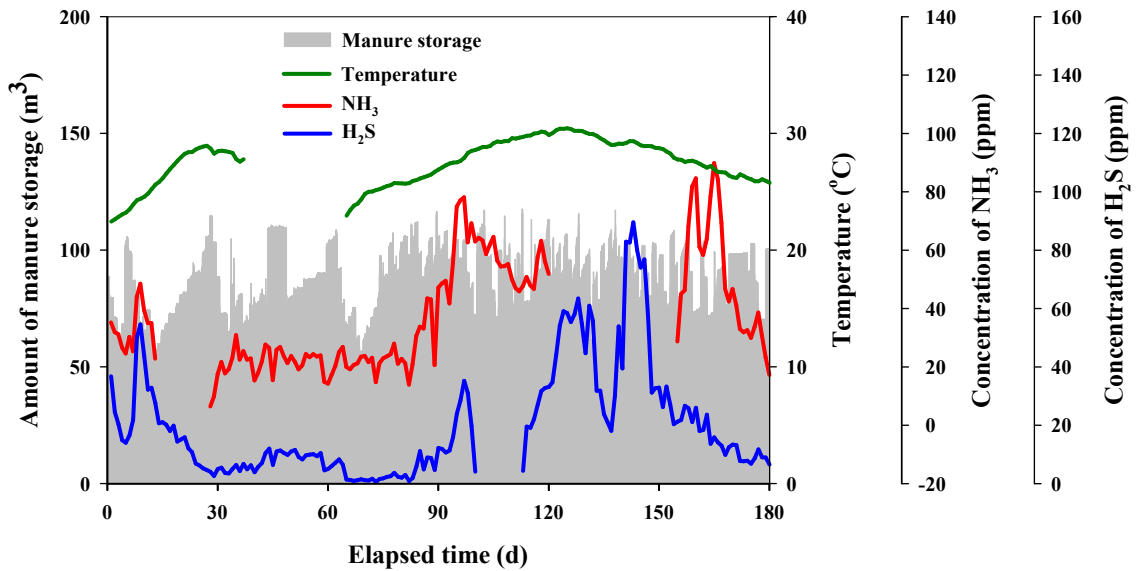


저장량 & 악취 농도 (NH<sub>3</sub>, H<sub>2</sub>S)의 변화

그림 3-15. 모니터링 기간 중 각 센서별 측정 결과 - A 농가



저장량 & 분뇨 물성 (pH, ORP, EC)의 변화



저장량 & 악취 농도 (NH<sub>3</sub>, H<sub>2</sub>S)의 변화

그림 3-16. 모니터링 기간 중 각 센서별 측정 결과 - B 농가

(3-3) 각 모니터링 인자별 측정 결과 분석 (주 단위 기준)

각 모니터링 인자별 분뇨 저장량에 따른 변화 양상을 파악하기 위해 주 단위로 나눠서 경향 분석을 진행하였다.

○ 분뇨의 유입/유출에 따른 모니터링 인자들의 변화 (Case 1)

그림 3-17는 분뇨 유입과 유출로 인해 급격한 저장량 변동이 발생하는 지점으로, 저장조 내 분뇨의 첫번째 유출 시점에 EC가 급격하게 하강하는 모습을 보이거나 이후 분뇨의 유출 시점에서는 특이적으로 낮아지거나 변화하는 모습을 보이지 않고, 타 인자들에 비해 변동성이 큰 모습을 보여주었다 [(a)].

NH<sub>3</sub>와 H<sub>2</sub>S의 경우 분뇨가 유입, 유출됨에 따라 분뇨 저장조 표면의 유동이 발생하여 순간적으로 악취물질의 농도 농도가 큰폭으로 올라가거나 내려가는 모습을 관찰할 수 있으며 [(b)], 데이터의 상승, 하강 양상이 두 악취물질 NH<sub>3</sub>와 H<sub>2</sub>S가 유사하게 움직이는 것을 확인하였다. 특히 슬러리피트에서 분뇨의 유입 시 악취의 농도가 평균적으로 상승하는 경향을 확인할 수 있었다 [(c)].

pH와 ORP의 경우 분뇨의 유입, 유출로 인한 변화를 관찰하지 못하였으며, pH의 경우 2210시간 이후 꾸준히 상승하는데, 이는 앞전에 유입된 슬러리의 pH가 높아 체류하던 분뇨와 혼합됨에 따라 올라가게 되는 것으로 판단된다. 또한 pH가 상승함에 따라 NH<sub>3</sub>의 농도도 함께 상승하는 것이 관찰되었다 [(d)].

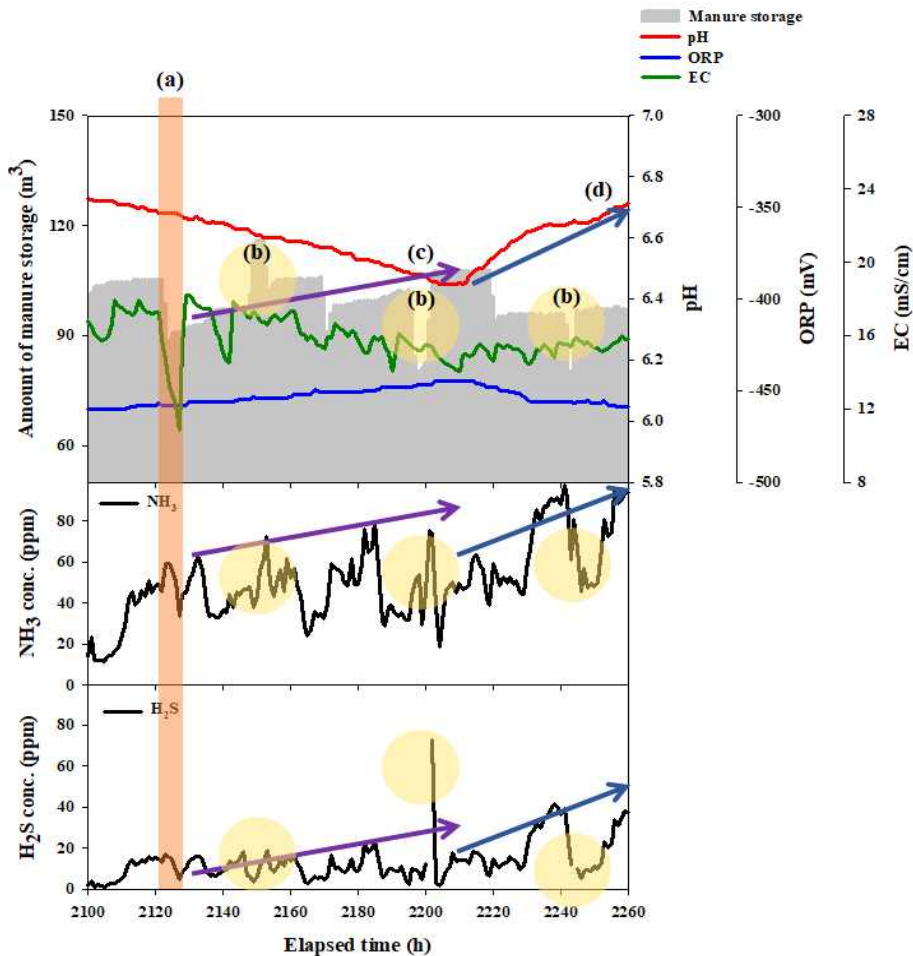


그림 3-17. 분뇨 유입/유출에 따른 모니터링 인자들의 변화 (Case 1)

○ 분뇨의 유입/유출에 따른 모니터링 인자들의 변화 (Case 2)

그림 3-18는 분뇨 유입과 유출로 인해 급격한 저장량 변동이 발생하는 지점으로, 2300~2410 시간까지 급격한 저장량의 감소가 발생했을 때 NH<sub>3</sub> 농도 역시 급격하게 감소하는 것을 관찰할 수 있었다 [(a)]. 특히 꾸준히 감소하는 분뇨 저장량으로 인해 중간에 측정이 끊어지기도 했으나 황화수소 역시 감소하는 추세를 나타내었다 [(b)].

해당기간 동안 pH가 6.5 이상으로 그림 3-17에 비해 높고, 분뇨의 수위도 높게 유지됨에 따라 NH<sub>3</sub>의 농도가 전반적으로 높게 나타나는 것으로 확인할 수 있으며, 이에 따라 NH<sub>3</sub>의 농도는 분뇨 저장량과 pH에 따라 영향을 받는 것으로 사료된다.

pH와 ORP는 저장조로 유입되는 슬러리의 성상에 따라 유입 이후 혼합되어 서서히 변화하는 모습을 보인 것과 달리 EC는 분뇨의 유입과 유출과정에서 특이적인 변화를 나타내지 않았다 [(C)]. 저장되어 있는 분뇨와 상태가 상이한 분뇨가 유입될 시 pH와 ORP의 변화를 확인할 수 있으나, 유입과 유출 시점을 확인할 수 있는 인자일 뿐 절대적인 분뇨 저장량에 따른 변화를 감지할 수 있는 모니터링 인자는 아닌 것으로 판단하였다.

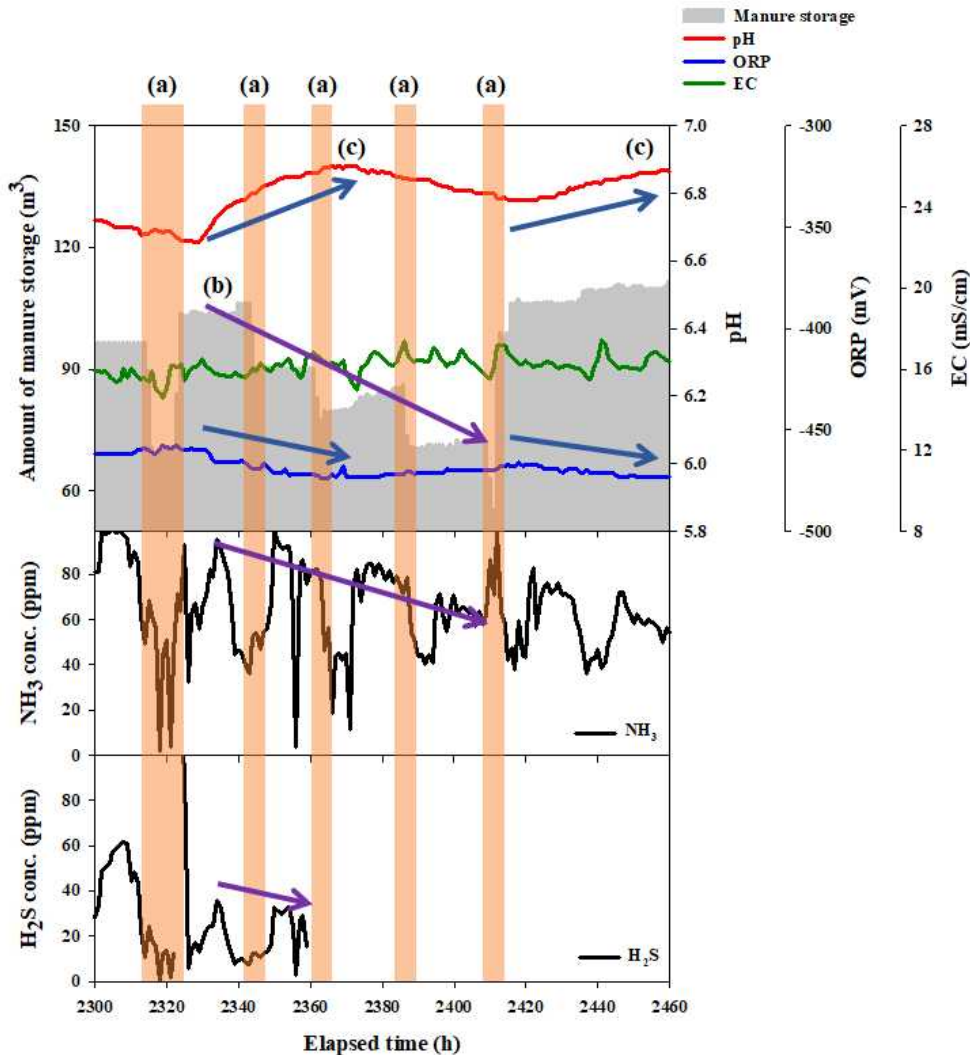


그림 3-18. 분뇨 유입/유출에 따른 모니터링 인자들의 변화 (Case 2)



○ 분뇨 저장량의 지속적인 증가에 따른 모니터링 인자들의 변화 (Case 3)

그림 3-19은 지속적으로 저장조로 분뇨가 유입되어 저장량이 꾸준히 증가하는 지점의 그래프이다. 저장조 내 분뇨 저장량이 늘어남에 따라 분뇨 물성 측정인자인 pH는 꾸준히 감소하고 있는데, 이는 저장조 내부에서 발생하는 혐기소화 과정중 가수분해 후 산생성 단계 혹은 NH<sub>3</sub>의 휘발로 인한 것으로 사료되며, ORP와 EC는 특이적인 경향을 확인할 수 없었다 [(a)].

분뇨의 저장량이 증가함에 따라 악취측정 인자인 NH<sub>3</sub>와 H<sub>2</sub>S는 지속적으로 증가하였으며, 이는 분뇨의 저장기간이 증가하여 꾸준히 분뇨가 저장조 내 적재됨에 따라 발생할 수 있는 부정적인 환경영향으로 사료되므로, 악취물질의 발생농도를 저장조 내 분뇨의 수거시점 결정 시 저장조 수위와 함께 관찰할 수 있는 보조인자로 활용할 수 있을 것으로 판단된다.

분뇨 저장량 관찰 결과 급격한 분뇨의 유입, 유출이 일어난 시점에서 급격히 증가하는 악취농도를 제외하고 분뇨 물성 측정인자들의 변화는 관찰되지 않았다. 그러나 악취물질의 경우에도 급격한 분뇨발생량에 관계없이 발생 농도의 증감이 꾸준하게 나타나 전체적으로 증가하는 경향은 확인할 수 있으나 저장량이 급격히 증가하는 특정시점을 정확히 파악하기에는 어려울 것으로 판단되었다.

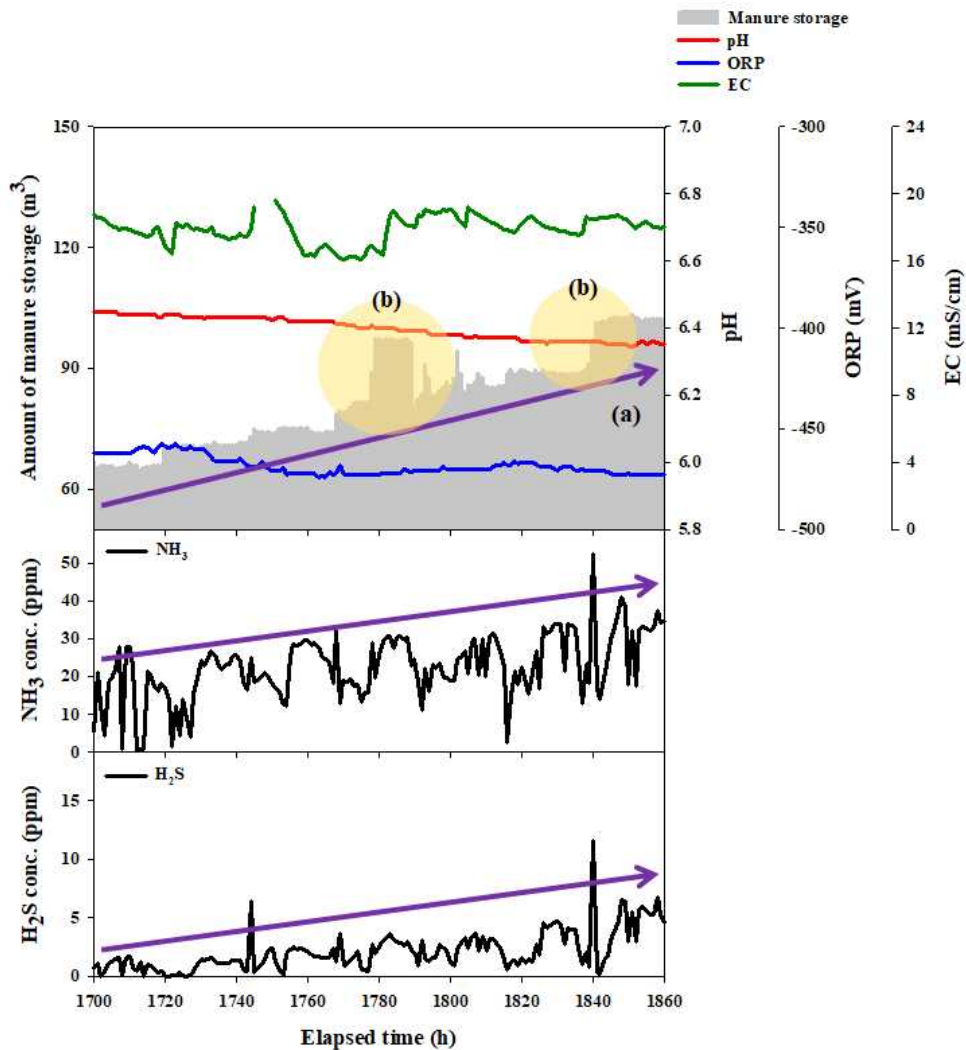


그림 3-19. 분뇨저장량의 지속적인 증가에 따른 변화 관찰 (Case 3)

#### (4) 분뇨 저장조 모니터링 결과 분석 (분뇨 저장량)

농가 분뇨 저장조 내 분뇨의 수위를 측정하기 위해 레이저 수위센서를 설치하였으며, 분뇨 표면의 높이를 감지하여 가장 직접적으로 분뇨 발생량을 산출할 수 있는 측정 인자로 사용하였다. 전체 모니터링 기간 동안 수위센서의 손상, 데이터 송신 이상 등의 문제는 발견되지 않았으며, 설치된 모든 센서 중 가장 안정적으로 정확한 값을 송출하여 데이터로 수집할 수 있었다.

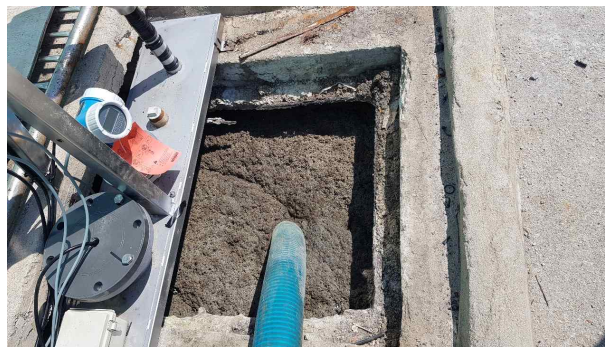
##### (4-1) 분뇨 저장량 환산

모니터링 시스템에서 송출하고 있는 수위 값을 각 농가 저장조의 유효 용적과 높이를 단순 대입하여 분뇨 저장량 및 주기별 분뇨 발생량을 산출하였으나 농가의 사육두수 대비 예상되는 분뇨 발생량과 차이가 나타났으며, 농가의 분뇨 관리방법과 센서의 설비 위치 등 다양한 요인을 포괄적으로 고려할 수 있는 인자의 확보가 필요하였다.

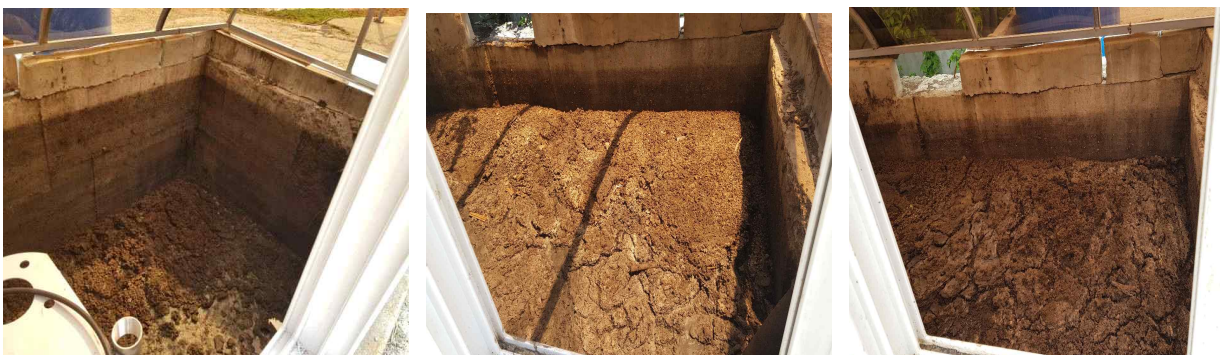
#### ○ 분뇨 저장조 수위를 단순 분뇨 저장량으로 환산 시 부정확성 유발 원인

##### ① 분뇨 저장조 내 스킴층의 발생

B 농가와 같이 슬러리피트 내 분뇨를 추가적인 처리없이 분뇨 저장조로 직접 유입시키는 농가에서는 분뇨 저장조 상부에 스킴의 영향으로 인한 발생량 측정 오차가 유발되었다. B 농가에서는 저장조 관찰 시 마다 저장조 표면 약 30~50cm가 부상한 고형물 및 곤충 사체로 뒤덮혀 있었으며, 이는 주변의 온도가 상승하고, 곤충들의 성장이 증가하는 여름철에 더 크게 발생하였다.



부드러운 거품형태의 스킴층 (A 농가)



기온의 상승에 따라 점차 두터워지는 스킴층의 변화 (B 농가)

그림 3-20. 각 농가별 분뇨 저장조에 형성된 스킴층의 모습

분뇨 유입 시 고액분리 후 저장조로 이송하는 A 농가의 분뇨 저장조에는 B 농가와 같은 단단하고 외부 이물질이 다량 섞여있는 스킴층의 형태가 아니라 분뇨 내 고형물의 부상으로 인해 발생하는 부드러운 스킴층이 형성되며, 그 형성 높이도 상대적으로 낮은편이었다.

② 분뇨 저장조 내 분뇨 고형물 침전층의 발달

공동자원화시설에서 수거해가는 분뇨는 스킴층 아래에 위치한 액상상태의 분뇨로 저장조 바닥에 침전된 분뇨 및 상부의 스킴은 수거해가지 않는다. 따라서 장기간 운영되어 왔던 분뇨 저장조 내 바닥에는 상당량의 분뇨 고형물 층이 발달해 있을 것으로 예상되며, 이로 인해 저장조 내 실제 분뇨 저장량은 설비 용량보다 현저히 적을 것으로 판단된다.

③ 수위센서의 설비 위치 (슬러리분뇨 유입지점에 근접)

저장조 모니터링 기간 내 항상 형성되어 있는 스킴층은 여름철 두껍고 딱딱하게 형성되며, 슬러리피트 내 분뇨의 유입 시 이 스킴층 위로 분뇨가 들어오게 된다. 이 때 단단한 스킴층의 발달로 인해 유입된 분뇨가 스킴층 아래 분뇨와 혼합되는 것이 아닌 스킴층의 위로 일시적으로 쌓이게 되며, 저장조 내 수위가 일정하에 상승하는 것이 아니라 분뇨가 유입된 측면의 수위만 상승하게 되어 저장조 수면의 높이가 고르지 못하게 되는 현상이 발생하고, 이후 전체적인 수위는 천천히 상승하게 된다.

B 농가의 수위센서는 슬러리피트 내 분뇨가 유입되는 방향에 가깝게 설치되어 있으며, 이는 실시간 저장량 산출에 오차로 작용할 가능성이 존재하게 된다. 따라서 수위센서의 경우 분뇨 유입시 일시적으로 급격한 상승을 방지하기 위해 분뇨 유입부분의 반대편에 설치하는 것이 바람직할 것으로 판단된다.



설비된 수위센서의 위치

슬러리분뇨 유입지점 아래의 불균일한 스킴층

그림 3-21. B농가 분뇨 저장조 내 수위센서의 설치 위치와 슬러리 분뇨 유입 지점 모습

농가 내 저장조의 실제 분뇨저장 용량이 저장 경향, 저장조 내 고형물 퇴적 정도 등 다양한 요인에 의해서 실제 유입된 분뇨가 저장되는 저장량이 변동될 수 있어 설계 분뇨저장 용량과 다르므로 수위에 따른 분뇨저장량의 보정이 요구된다. 따라서 항상 일정한 값을 나타내는 현상을 이용하여 현재의 상태를 보정할 수 있는 방안이 필요하며, 본 연구에서는 일정하게 분뇨를 수거해가는 공동자원화시설의 차량의 최대 위탁량 (A 농가 - 23.5톤/회; B 농가 - 7톤/회)을 기준으로 수위 및 수위에 따른 분뇨 저장량을 보정하였다. 즉 보정된 값은 설계 저장가능 용량 대비 실제 저장가능 용량이라고 할 수 있다.

실제 가축분뇨 공동자원화시설 차량의 최대 위탁량에 따른 수위의 변화를 측정하여 모니터링 값에 대입하여 분뇨저장 변동량 보정값을 산출하였다.

A 농가의 경우 육안상으로도 형성되는 스킴층의 양이 적고, 최종 분뇨 저장조 전단에 저장조를 두어 침전조로서 활용하기 때문에 저장조 바닥에 쌓여있는 고형물의 양도 적으로 것으로 예상되었으며, 산출된 분뇨저장 변동량 보정값은 1.904로 나타났다. B 농장의 경우 두터운 스킴층의 존재와 다량의 분뇨 고형물 침전에 예상되었으며, 분뇨저장 변동량 보정값 산출 결과 1.635로 나타났다.

산출된 보정값을 이용하여 수위에 따른 분뇨 저장량을 수정하였으며, 적정 수거시스템의 구축 시 이런 유형의 분뇨 저장량 보정작업은 일정 주기로 수행하여 저장량의 정확도를 향상시킬 필요가 있을 것으로 판단된다.

표 3-14. 분뇨위탁량 모니터링 결과 및 분뇨저장 변동량 보정값 산출 (A 농가)

항목	결과	비고	
A 농가	총 위탁량 (톤)	2,187.0	모니터링 결과 분석치
	총 위탁일수 (일)	32	
	총 위탁횟수 (회)	85	
	위탁주기 (일/회)	4.7	
	위탁일 당 위탁 횟수 (회/일)	2.66	
	모니터링 상 회당 위탁량 (톤/회)	25.71	
	이론적인 회당 위탁량 (톤/회)	23.5	위탁차량의 최대 이송가능 용량 고려
	분뇨저장 변동량 보정값*	1.094	전기간 분뇨저장 변동량에 적용

\*분뇨 수거차량의 분뇨 수거 시 모니터링되는 수위변동에 따라 산출

표 3-15. 분뇨위탁량 모니터링 결과 및 분뇨저장 변동량 보정값 산출 (B 농가)

항목	결과	비고	
B 농가	총 위탁량 (톤)	2,190.4	모니터링 결과 분석치
	총 위탁일수 (일)	91	
	총 위탁횟수 (회)	192	
	위탁주기 (일/회)	2.0	
	위탁일 당 위탁 횟수 (회/일)	2.11	
	모니터링 상 회당 위탁량 (톤/회)	11.45	
	이론적인 회당 위탁량 (톤/회)	7	위탁차량의 최대 이송가능 용량 고려
	분뇨저장 변동량 보정값*	1.635	전기간 분뇨저장 변동량에 적용

\*분뇨 수거차량의 분뇨 수거 시 모니터링되는 수위변동에 따라 산출

산출된 보정값을 적용하여 분뇨저장량을 환산하고, 모니터링 기간 내 변동되는 분뇨 저장량 데이터를 이용하여 양돈농가에서 실제 발생하는 분뇨량과 가축분뇨 발생원단위를 이용한 분뇨 발생 예측량이 얼마나 유사한지 비교해 보았다.

A 농가의 경우 모니터링 초반 낮아진 저장조 수위가 좀처럼 변하지 않는 구간이 존재하는데 이는 전단에 있는 저장조 내에 분뇨가 모두 차지 않아 최종 저장조로 넘어오지 않았기 때문에 봄철에는 전단에 있는 저장조 내 분뇨까지 대부분 수거해 A 농가 내 저장조가 거의 비게되며, 여름·가을동안 최종 저장조의 수위가 상승하게되는 것을 확인할 수 있었다. 자연유하식이 아닌 슬러리피트를 농장주가 열고 이를 고액분리하여 유입되는 분뇨 관리시스템을 가지고 있기 때문에 저장조의 수위가 급격하게 증감하는 것을 확인할 수 있었다.

주별로 평균 분뇨발생량을 확인한 결과 1주간  $74.7 \pm 163.6 \text{ m}^3$ , 2주간  $149.3 \pm 251.8 \text{ m}^3$ , 월간  $273.8 \pm 231.8 \text{ m}^3$ 의 발생량을 산출할 수 있었으며, 이는 이론적인 분뇨발생량과 대비하여 주간, 2주간, 월간 각각 261.4, 209.1, 122.0% 수준인 것으로 나타났다. 위 결과를 통해 A 농가에서는 주간, 2주간에 비해 월간으로 분뇨 발생량을 산출 시 더 실제적인 양돈분뇨 발생량을 산출할 수 있을 것으로 예상된다.

B 농가의 경우 모니터링 기간 내내 일정한 수위를 유지하면서 자주 변동하는 모습을 확인할 수 있었는데 이 시기마다 분뇨 저장조로의 분뇨 유입 및 수거가 발생하는 것으로 예측할 수 있다. 주별로 평균 분뇨발생량을 확인한 결과 주간  $58.0 \pm 33.0 \text{ m}^3$ , 2주간  $116.0 \pm 55.3 \text{ m}^3$ , 월간  $251.3 \pm 99.5 \text{ m}^3$ 의 발생량을 산출할 수 있었으며, 이는 이론적인 분뇨발생량과 대비하여 주간, 2주간, 월간 각각 108.3, 108.3, 109.5% 수준인 것으로 나타났다. A 농가와 다르게 B 농가에서는 모니터링 분뇨 발생량 값과 이론적인 분뇨발생량이 거의 유사했으며, 이는 단일 분뇨 저장조 운영과 고액분리없이 분뇨 그대로 저장조로 유입되는 농가의 관리 특징에서 기인된 결과로 판단된다.

A와 B 농가의 구축된 분뇨 저장조 모니터링 시스템의 센서 중 수위 센서를 이용한 분뇨 저장량 산출 및 농가의 분뇨 발생량 산출이 타 센서들에 비해 직관적으로 값을 확인하고 대응할 수 있어 수거시점 선정에 유리할 것으로 판단되며, 저장조 내 유효 용적 대비 현재 저장량 등 비율적인 표현을 통해 저장조의 현재 분뇨 저장상황을 사용자와 분뇨수거시설에 제공할 수 있을 것으로 예상된다. 비교적 장기간의 데이터 확보를 통한 월간, 분기간, 연간 농가의 분뇨 발생량 산출 시 보다 정확한 가축분뇨 발생량을 산출할 수 있을 것으로 예상되며, 발생된 가축분뇨의 전체 처리과정을 추적하는 수단으로 활용할 수 있을 것으로 판단된다.

표 3-16. 분뇨저장조 모니터링을 통한 실제 분뇨발생량 산출 및 이론치 대비 분뇨발생 비율 비교 (A 농가)

분뇨발생량 산출기준 주 월	이론적인 분뇨발생량			실제 분뇨발생량 (모니터링)			이론치 대비 분뇨발생 비율			
	주간 톤/주	2주간 톤/2주	월간 톤/월	주간 톤/주	2주간 톤/2주	월간 톤/월	주간 %	2주간 %	월간 %	
1				361.1	361.1		459.7	229.9		
2	1 (4월)			0.0		361.1	-		107.3	
3				0.0	0.0		-	-		
4				0.0			-			
5	2 (5월)			0.0		0.0	-			
6				0.0		0.0	-	-	-	
7				0.0	0.0		-	-		
8	3 (6월)			0.0			-			
9				0.0	0.0	0.0	-	-	-	
10				0.0			-			
11	4 (7월)	78.54	107.1	229.5	0.0	0.0	-	-		
12				0.0			-			
13					203.8	203.8	298.9	259.5	129.8	88.8
14	5 (8월)				0.0		-			
15				0.0	0.0		-	-		
16				0.0			-			
17	6 (9월)				95.1	175.5	580.4	121.0	111.7	172.4
18					80.4			102.4		
19					68.6	84.1		87.3	53.5	
20	6 (9월)				15.5		402.2	19.8		119.5
21					128.0			162.9	520.8	
22					690.1	818.1		878.7		
합계 (톤)					1,642.6					
평균				74.7±163.6	149.3±251.8	273.8±231.8	261.4±283.8	209.1±185.5	122.0±35.9	

표 3-17. 분뇨저장조 모니터링을 통한 실제 분뇨발생량 산출 및 이론치 대비 분뇨발생 비율 비교 (B 농가)

분뇨발생량 산출기준 주 월	이론적인 분뇨발생량			실제 분뇨발생량 (모니터링)			이론치 대비 분뇨발생 비율			
	주간 톤/주	2주간 톤/2주	월간 톤/월	주간 톤/주	2주간 톤/2주	월간 톤/월	주간	2주간	월간	
							%			
1				85.6				159.9		
2				7.8	93.4			14.6	<b>93.4</b>	
3	1 (4월)			10.3		143.3		19.2		62.5
4				32.3	42.6			60.3	<b>42.6</b>	
5				66.0				123.3		
6				83.7	149.7			156.2	<b>149.7</b>	
7	2 (5월)			58.2		214.8		108.7		93.6
8				2.9	61.2			5.5	<b>61.2</b>	
9				18.1				33.8		
10				11.3	29.4			21.0	<b>29.4</b>	
11	3 (6월)			65.6		173.2		122.4		75.5
12				46.0	111.5			85.9	<b>111.5</b>	
13		53.6	107.1	229.5	50.9			95.0		
14				77.3	128.2			144.4	<b>128.2</b>	
15	4 (7월)			71.9		398.7		134.3		173.7
16				135.5	207.4			253.1	<b>207.4</b>	
17				101.3				189.1		
18				73.9	175.2			138.0	<b>175.2</b>	
19				106.2				198.3		
20	5 (8월)			86.1	192.3	343.0		160.8	<b>192.3</b>	149.4
21				51.9				96.8		
22				77.8	129.7			145.3	<b>129.7</b>	
23				31.8				59.4		
24	6 (9월)			55.3	87.1	234.8		103.2	<b>87.1</b>	102.3
25				41.6				77.7		
26				58.7	100.3			109.6	<b>100.3</b>	
합계 (톤)				1,507.9						
평균				58.0±33.0	116.0±55.3	251.3±99.5		<b>108.3±61.6</b>	<b>108.3±51.6</b>	<b>109.5±43.4</b>

### 3-2. 분뇨 저장조의 적정 수거 제어인자 선정 및 수거 시점 분석

분뇨 저장조 모니터링을 통해 수위를 이용한 분뇨 저장량과 분뇨 발생량은 주기적인 보정을 통해 신뢰성 있는 값을 산출할 수 있다고 판단하였으며, 이외에 분뇨 물성과 악취물질의 농도 중 분뇨 저장량과 연동하여 적정 수거 시점을 확인할 수 있는 인자를 선정하기 위해 각 항목별 상관관계 분석 및 회귀분석을 진행하였다.

비교 항목들이 모두 측정된 기간의 데이터를 기반으로 SPSS (Statistics Package for Social Science) 프로그램을 이용해 피어슨 상관계수를 조사하였으며, 이를 통해 각 항목별 상관관계를 분석하였다. 변수 x와 y의 존재 시, x가 증가할 때 y도 증가하는 것을 양의 상관관계라고 하며, x가 증가할 때 y가 감소하는 것을 음의 상관관계라고 한다. 두 항목별 산출된 피어슨 상관계수는 아래 표 3-18에 나온 기준을 바탕으로 상관관계의 강도를 분류할 수 있으며, 1이나 -1에 가까울수록 강한 상관관계 (양 혹은 음)를 가지고, 0에 가까울수록 상관관계가 없다고 할 수 있다.

분석한 데이터는 모든 데이터가 상대적으로 많이 확보되어있는 B 농가의 모니터링 데이터를 이용하였으며, 전체 데이터 중 모든 데이터가 저장된 시점을 추려 상관관계 분석에 이용하였다.

표 3-18. 피어슨상관계수에 따른 상관관계 해석

상관계수 (r)의 범위	상관관계
-0.1 ~ -0.7	강한 음적 상관관계
-0.7 ~ -0.3	뚜렷한 음적 상관관계
-0.3 ~ -0.1	약한 음적 상관관계
-0.1 ~ +0.1	거의 무시될 수 있는 상관관계
+0.1 ~ +0.3	약한 양적 상관관계
+0.3 ~ +0.7	뚜렷한 양적 상관관계
+0.7 ~ +1.0	강한 양적 상관관계

#### (1) 분뇨 저장조 내 모니터링 인자간 상관관계 분석

피어슨 상관계수 분석결과, NH<sub>3</sub>가 pH (0.748), 온도 (0.734)와 통계적으로 유의적인 강한 양의 상관관계 (p<0.01)를 나타내는 것으로 분석되었으며, 이외에 H<sub>2</sub>S (0.585), 분뇨저장량 (0.390)과 통계적으로 유의적인 뚜렷한 양의 상관관계 (p<0.01)를 나타내었다. 즉, NH<sub>3</sub>의 경우 pH와 온도, H<sub>2</sub>S, 분뇨저장량이 증가할 때 함께 증가하는 것으로 분석되었다. H<sub>2</sub>S의 경우 NH<sub>3</sub> (0.585), ORP (0.576), pH (0.444)와 통계적으로 유의적인 뚜렷한 양의 상관관계 (p<0.01)를 나타냈으며, EC (-0.471)와는 뚜렷한 음의 상관관계 (p<0.01)를 나타내고 있었다. 분뇨 저장량은 ORP (-0.376)와 뚜렷한 음의 상관관계 (p<0.05)를 나타내고 있어 저장조 내 분뇨 저장량이 증가할수록 ORP는 낮아지는 경향을 관찰할 수 있을 것으로 예상된다. 통계적으로 저장량과 온



도 (0.624)가 뚜렷한 양의 상관관계 ( $p < 0.01$ )를 가진다고 산출되었는데 이는 여름시기 급수량으로 인해 증가된 슬러리피트 분뇨가 다량 다른 계절들보다 다량 유입되고 있기 때문인 것으로 판단된다.

표 3-19. 분뇨 저장조 모니터링 인자간 상관관계 (B 농가)

	pH	ORP	EC	NH <sub>3</sub>	H <sub>2</sub> S	Temp
ORP	-0.307					
p-value	0.048					
EC	-0.471	-0.124				
p-value	0.002	0.434				
NH <sub>3</sub>	<b>0.748</b>	-0.221	-0.477			
p-value	<b>&lt;.0001</b>	0.160	0.001			
H <sub>2</sub> S	0.444	0.576	-0.471	0.585		
p-value	0.003	<b>&lt;.0001</b>	0.002	<b>&lt;.0001</b>		
Temp	0.455	-0.449	-0.199	<b>0.734</b>	0.131	
p-value	0.002	0.003	0.205	<b>&lt;.0001</b>	0.409	
Storage amount	0.150	-0.376	-0.208	0.390	0.015	0.624
p-value	0.343	0.014	0.187	0.011	0.926	<b>&lt;.0001</b>

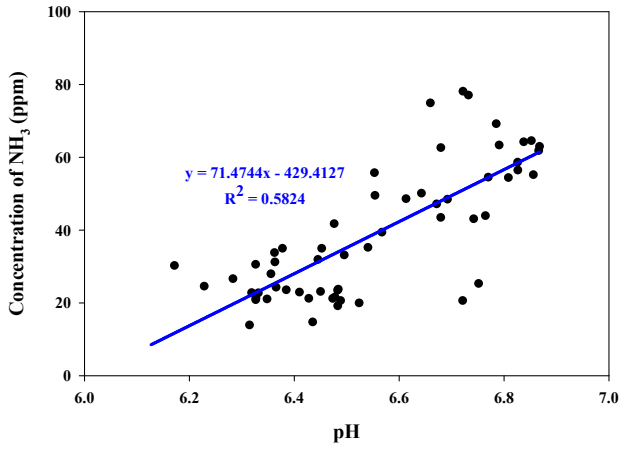
이를 바탕으로 분뇨 저장조 내 저장량, 저장조 주변 환경영향 인자인 NH<sub>3</sub>, H<sub>2</sub>S를 기준으로 각 모니터링 인자들의 분포 및 회귀식을 산출하였다.

최소 뚜렷한 상관관계를 지니고 있는 항목들을 가지고 회귀분석을 진행한 결과 NH<sub>3</sub>와 pH의 선형회귀식이 약 58.2%의 실제 데이터를 반영할 수 있는 것으로 분석되었으며, 이외의 분석항목인 온도, EC, 분뇨 저장량은 상관성은 나타냈지만 선형적으로 유의적인 영향을 미치지 않았다.

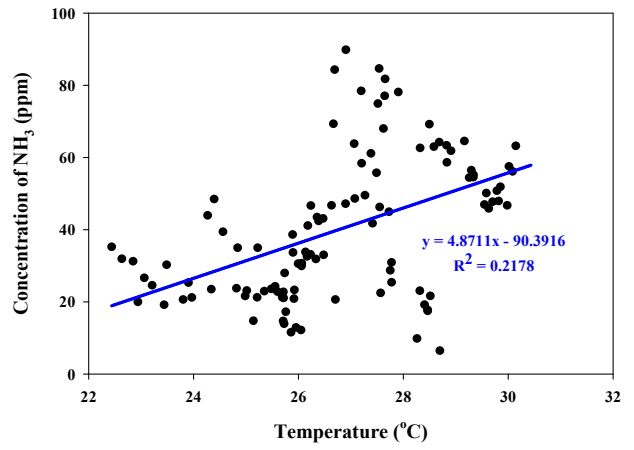
마찬가지로 최소 뚜렷한 상관관계를 지니고 있는 항목들을 가지고 회귀분석을 진행한 결과 H<sub>2</sub>S와 pH, ORP, NH<sub>3</sub>는 상관성은 나타냈지만 선형적으로 유의적인 영향을 미치지 않았으며, 분뇨 저장량의 경우도 EC와 ORP 사이에 선형적으로 유의적인 영향을 미치지 않는 것으로 분석되었다.

분석 결과와 같이 각 인자 간 상관관계는 있으나 선형적인 유의성은 나타나지 않아 해당 관계

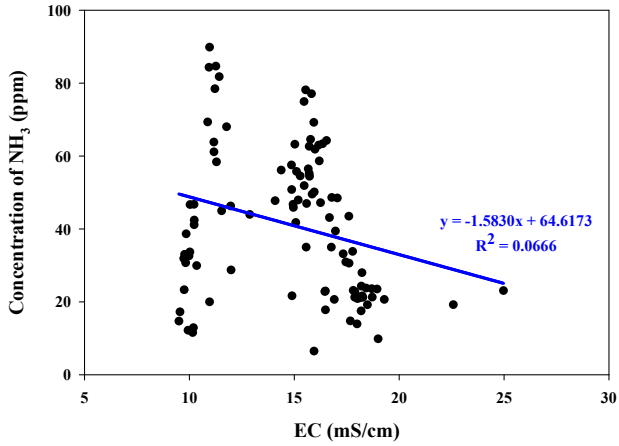
식별을 이용한 분뇨 저장량 예측은 무리인 것으로 판단된다.



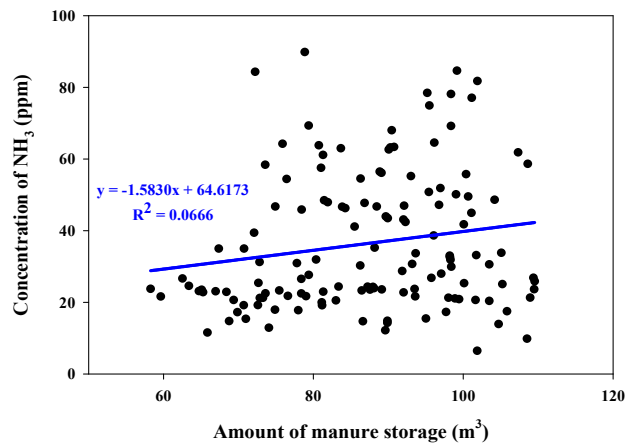
저장조 분뇨 내 pH에 따른 NH<sub>3</sub> 농도 변화



저장조 내 온도에 따른 NH<sub>3</sub> 농도 변화

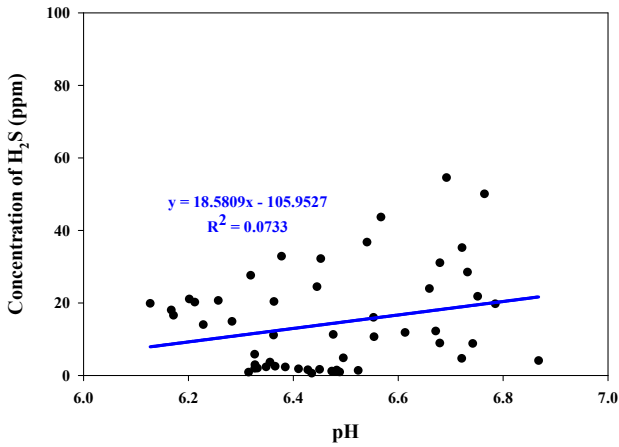


저장조 분뇨 내 EC에 따른 NH<sub>3</sub> 농도 변화

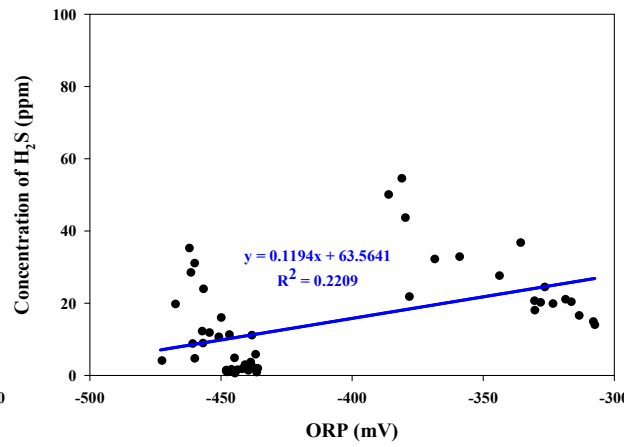


분뇨 저장량에 따른 NH<sub>3</sub> 농도 변화

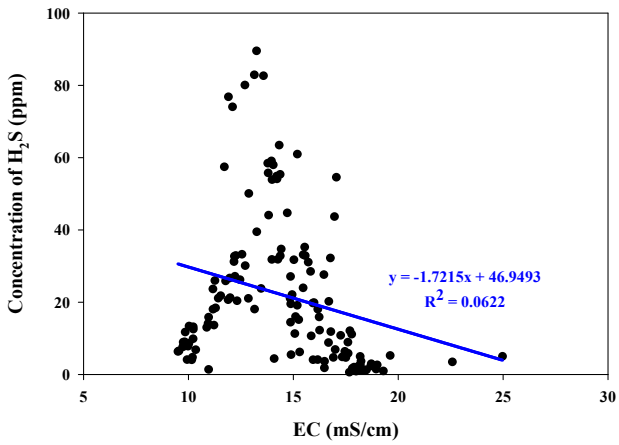
그림 3-22. 저장조 모니터링 인자에 따른 NH<sub>3</sub> 농도 변화



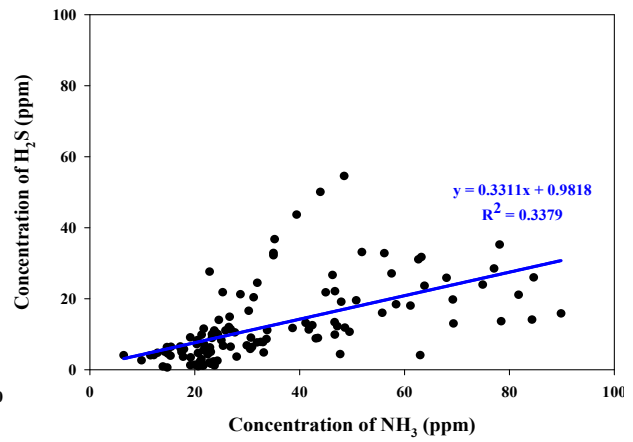
저장조 분뇨 내 pH에 따른 H<sub>2</sub>S 농도 변화



저장조 분뇨 내 ORP에 따른 H<sub>2</sub>S 농도 변화

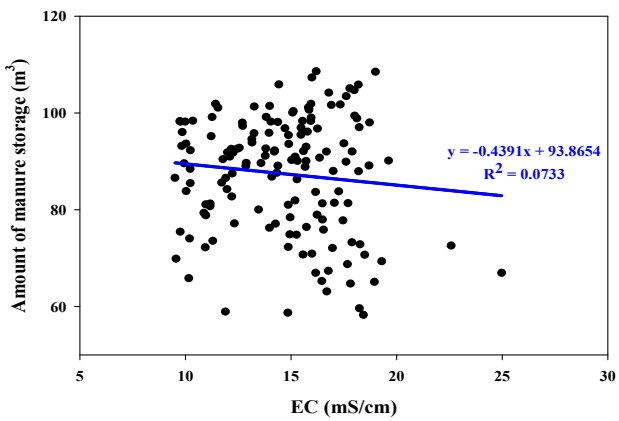


저장조 분뇨 내 EC에 따른 H<sub>2</sub>S 농도 변화

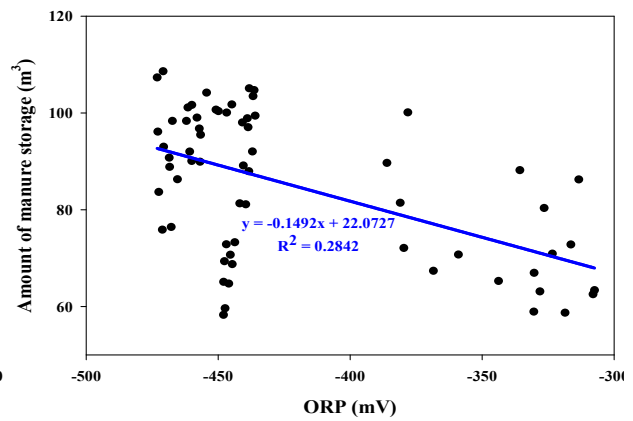


NH<sub>3</sub> 농도에 따른 H<sub>2</sub>S 농도 변화

그림 3-23. 저장조 모니터링 인자에 따른 H<sub>2</sub>S 농도 변화



저장조 분뇨 내 EC에 따른 분뇨 저장량 변화



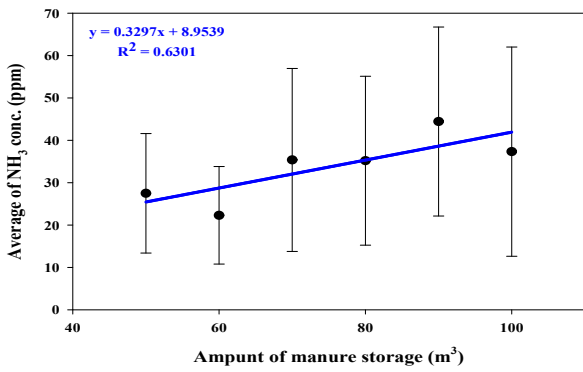
저장조 분뇨 내 ORP에 따른 분뇨 저장량 변화

그림 3-24. 저장조 모니터링 인자에 따른 분뇨 저장량 변화

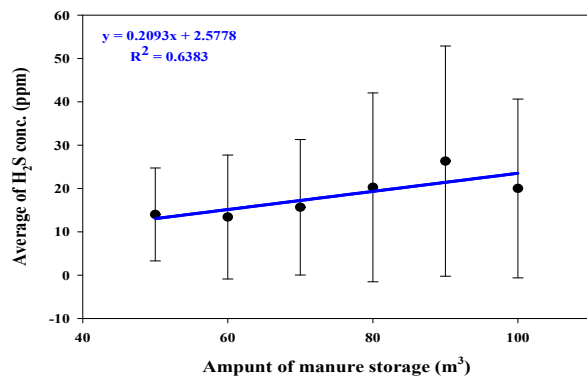
(2) 악취물질과 복합 인자간 상관관계 분석

상관관계 분석결과 악취물질은 저장조 내 분뇨 저장량과 온도 등 환경적인 영향에 의해 증가하는 것으로 나타났으며, 각 인자의 수준별 평균 값을 이용하여 선형적 관계식을 산출해 분석하였다.

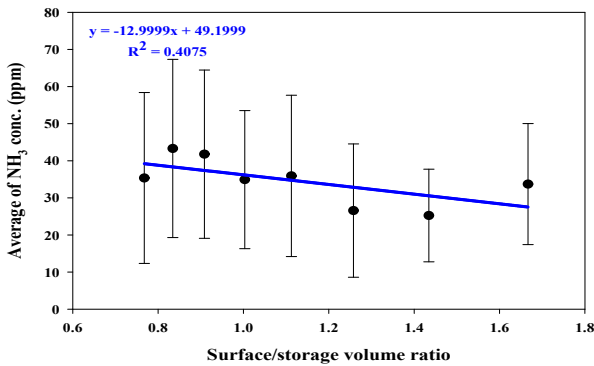
분석결과 NH<sub>3</sub> 농도 변화에 대한 선형적 상관관계가 높은 것은 분뇨 저장량 (R<sup>2</sup>=0.6301)이었으며, H<sub>2</sub>S의 경우 저장조의 표면적/용적비 (R<sup>2</sup>=0.6829)가 가장 높은 선형적 상관관계를 가진 것으로 나타났다. 높은 수준의 선형적 상관관계는 아니지만 각 악취물질의 발생농도에 영향을 미치는 인자를 확인할 수 있었다.



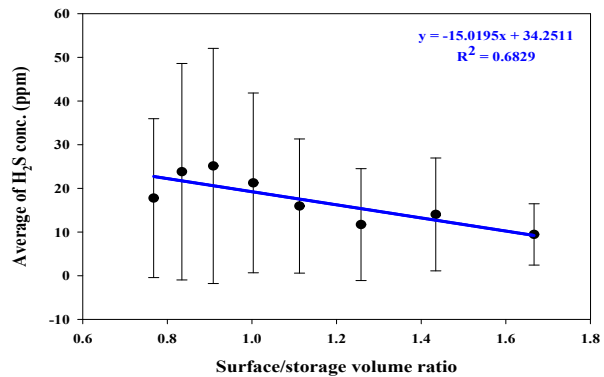
분뇨 저장량에 따른 NH<sub>3</sub> 농도 변화



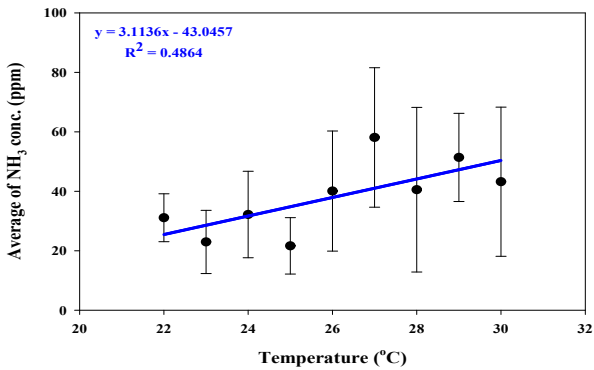
분뇨 저장량에 따른 H<sub>2</sub>S 농도 변화



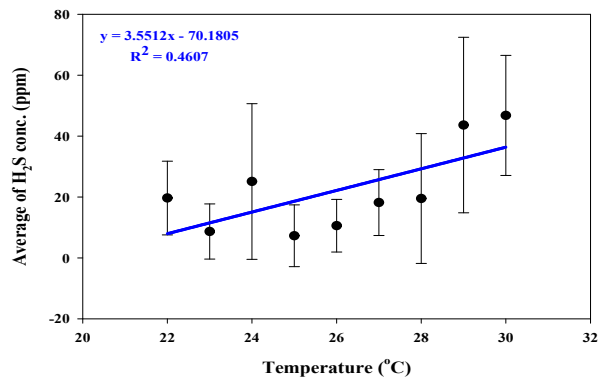
저장조의 표면적/용적비에 따른 NH<sub>3</sub> 농도 변화



저장조의 표면적/용적비에 따른 H<sub>2</sub>S 농도 변화



온도에 따른 NH<sub>3</sub> 농도 변화



온도에 따른 H<sub>2</sub>S 농도 변화

그림 3-25. 저장조 모니터링 인자에 따른 NH<sub>3</sub>, H<sub>2</sub>S 농도 변화

분뇨 저장조 내  $\text{NH}_3$ 와  $\text{H}_2\text{S}$ 의 농도에 대한 환경인자의 복합적인 영향을 고려하기 위해서 분뇨 저장량과 온도에 따른 각 악취물질의 농도변화를 분석하였다.

분뇨 저장량과 온도 조건에서 발생하는  $\text{NH}_3$  농도를 관찰한 결과 분뇨저장량  $70 \text{ m}^3$  이상, 온도  $26^\circ\text{C}$  이상에서  $60\text{ppm}$  이상으로  $\text{NH}_3$  농도가 급격히 증가하는 것을 확인할 수 있었으며,  $\text{H}_2\text{S}$  농도의 관찰 결과, 매우 높은 온도를 띄는 수준은 분뇨  $80 \text{ m}^3$  이상, 온도  $27^\circ\text{C}$  이상이며,  $\text{H}_2\text{S}$ 는 낮은 온도인  $26^\circ\text{C}$  이하에서도  $60\text{ppm}$  이상을 보여  $\text{H}_2\text{S}$ 의 농도를 낮추기 위해서는 분뇨 저장량  $70 \text{ m}^3$ 이하로 유지시켜야 할 것으로 판단된다.

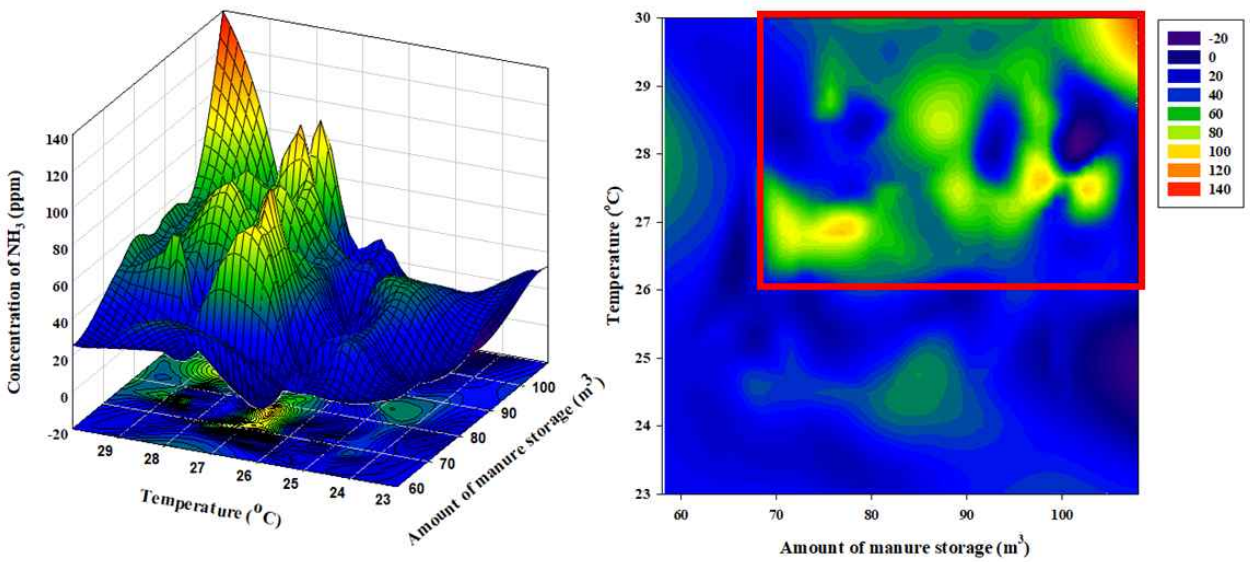


그림 3-26. 분뇨 저장량 및 온도에 따른  $\text{NH}_3$  농도의 변화

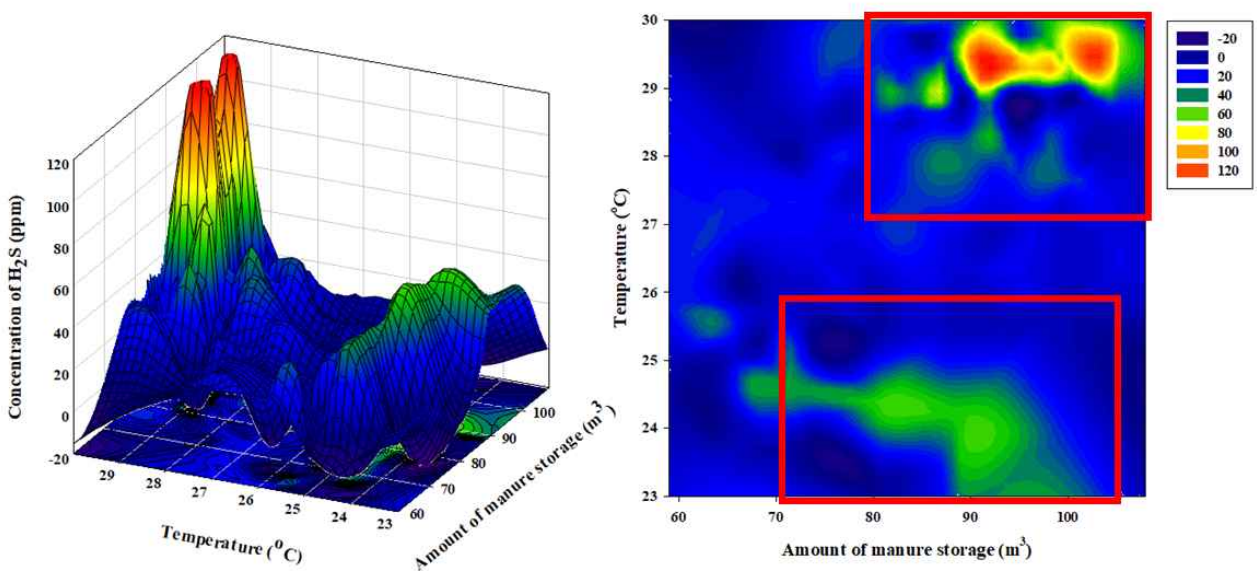


그림 3-27. 분뇨 저장량 및 온도에 따른  $\text{H}_2\text{S}$  농도의 변화

### (3) 저장조 내 분뇨의 적정 수거시점 제어인자 및 제어시점 선정

분뇨 저장조의 모니터링 데이터를 종합해 볼 때, 분뇨 저장량에 따른 각 모니터링 인자별 영향은 다음과 같다.

#### ○ pH

- 분뇨 저장조 내 분뇨의 유입 시 저장조 내 분뇨와 유입되는 분뇨 사이의 pH 차이로 인해 분뇨의 유입 후 pH가 서서히 변화하는 경향
- 분뇨의 유출 시에는 값이 크게 변동하지 않음.
- 분뇨의 체류시간이 길어질 시 분뇨 내 혐기소화 과정으로 인한 산 생성 및 암모니아 휘발로 pH가 낮아지는 경향 발생
- pH가 낮아질 경우  $\text{NH}_3$  농도가 낮아져 저장조 환경에 영향

#### ○ ORP

- 분뇨 저장량이 증가함에 따라 계속해서 값이 낮아지는 경향을 보이며, pH와 마찬가지로 저장조 내 분뇨의 성분과 유입되는 분뇨의 성분이 상이할 시 새로 유입된 분뇨와 기존 저장조 내 분뇨가 혼합되면서 서서히 변화함.
- 통계분석 결과에 따르면 온도가 낮아짐에 따라 ORP가 상승하는 음의 상관관계를 가지며, ORP가 증가함에 따라  $\text{H}_2\text{S}$ 도 증가하는 뚜렷한 양의 상관관계를 지님.

#### ○ EC

- 모니터링 기간 내 값의 잦은 변동을 보여주면서 특정한 시점을 제어하는데 무리가 있을 것으로 판단되며, 표 3-19 에 분석된 결과와 같이 EC가 낮아질 때  $\text{NH}_3$ ,  $\text{H}_2\text{S}$ 의 농도가 높아지는 음의 상관관계를 가지는 것으로 나타남.

#### ○ $\text{NH}_3$

- 저장조 내 분뇨 저장량과 온도의 상승 시 함께 상승하는 음의 상관관계를 가지며, 분뇨의 유입, 유출시 발생하는 저장조 분뇨 표면의 유동으로인해 급격한 상승과 하강이 반복될 수 있음.

#### ○ $\text{H}_2\text{S}$

- pH, ORP,  $\text{NH}_3$ 와 뚜렷한 양의 상관관계를 나타내고, EC와는 음의 상관관계를 가진 것으로 분석되었으나, 분뇨 저장량을 감지·확인하기 어려움.

#### ○ 수위

- 저장조 내 분뇨의 양과 유입되는 분뇨의 양을 직접적으로 확인할 수 있는 핵심 모니터링 인자로 이를 이용하여 농가의 분뇨 생산량 데이터 확보와 분뇨 처리과정의 투명성을 제고할 수 있음. 저장조 분뇨 상부의 스크럼층 발생, 저장조 바닥의 고형물 침전·고착 등 다양한 요인으로 인해 주기적으로 고정된 양을 회수하면서 수위 변화 대비 분뇨 저장량 보정 값을 산출 적용해 주어야 함.

○ 온도

- 대표적인 악취물질인 NH<sub>3</sub>의 농도와 통계적으로 유의적인 강력한 양의 상관관계를 지니고 있으며, 계절을 대표하는 인자로 계절별 저장용량의 변화 등 다양한 수거 시점을 위한 중요한 요인 중 하나로 이용될 수 있음.

따라서 본 연구에서는 저장조 내 분뇨의 수거시점을 결정하는 핵심인자로 수위를 선정하였으며, 농장 환경에 영향을 미치는 악취물질의 제어를 위해 보조인자로 온도, NH<sub>3</sub>, H<sub>2</sub>S를 이용하여 분뇨의 수거시점을 결정할 필요가 있는 것으로 판단 하였다.

B 농가에서 저장조 내 발생 악취를 최소화하면서 분뇨를 수거하기 위해 적용되는 분뇨 저장량은 약 70 m<sup>3</sup> 이하로 저장조의 수위로는 1.43 m에 해당하며, 전체 분뇨 저장조의 유효용적의 40.9%에 해당하는 수준이다. 해당 값은 모니터링 기간의 주요 온도인 23 ~ 30°C 범위 내에서 적용할 수 있으며, 이를 벗어나는 조건에서는 추가적인 모니터링을 통해 빅데이터를 확보하고 제어 시점을 개발할 수 있을 것으로 사료된다.

위 결과를 바탕으로, B 농가의 저장조 내 분뇨 저장량을 70 m<sup>3</sup> 이하로 유지 시 발생하는 악취물질의 저감율을 분석하였다. 먼저 B 농가의 전체 모니터링 기간 중 NH<sub>3</sub>와 H<sub>2</sub>S의 평균 농도는 각각 36.5±22.2, 19.7±21.2 ppm이었으며, 분뇨 저장량 70 m<sup>3</sup> 이상의 조건에서 NH<sub>3</sub>와 H<sub>2</sub>S의 평균 농도는 각각 38.4±22.6, 20.7±22.0 ppm, 분뇨 저장량 70 m<sup>3</sup> 이하의 조건에서는 각각 22.3±11.5, 12.8±13.2 ppm로 산출되었다. B 농가의 저장조 내 분뇨 저장량을 70 m<sup>3</sup> 이하로 유지 시 악취물질의 저감율은 전체 모니터링 기간 대비 NH<sub>3</sub>, H<sub>2</sub>S 각각 38.7, 35.2%로 나타났고, 분뇨 저장량 70 m<sup>3</sup> 이상의 조건 대비 각각 41.8, 38.2%로 산출되었다 (표 3-20).

표 3-20. B 농가 분뇨 저장조의 수거시점 적용 시 악취저감 효과

기준	악취 평균 농도 (ppm)			분뇨 저장량 고정 시 악취저감 효과 (%)		
	모니터링 전 기간 (a)	70 m <sup>3</sup> 이상 (b)	70 m <sup>3</sup> 이하 (c)	전 기간 기준 저감율 [(a-c)/a*100]	70 m <sup>3</sup> 이상 기준 저감율 [(b-c)/b*100]	
악취 종류	NH <sub>3</sub>	36.5±22.2	38.4±22.6	22.3±11.5	38.7	41.8
	H <sub>2</sub> S	19.7±21.2	20.7±22.0	12.8±13.2	35.2	38.2

#### 4. 분뇨 발생량 현장 실증 및 분석

분뇨 저장조 모니터링 데이터를 전기간에 걸쳐 확보하기 어려웠던 A 농가를 대상으로 분석된 문제점을 보완하여 모니터링 시스템을 재구축하였으며, 그 데이터를 분석하였다.

##### 4-1. 분뇨 저장조 모니터링 시스템 보완 시 고려사항

선행 분뇨 저장조 모니터링 과정에서 A 농가에서는 수위를 제외한 대부분의 항목에서 원활한 데이터의 확보가 이뤄지지 않았으며, 이를 보완하기 위해 센서 및 측정을 위한 구조물을 추가적으로 설치하여 안정적인 데이터 확보를 위해 노력하였다.

분뇨 물성 데이터 확보를 위한 센서의 경우 기존의 pH, ORP, EC, 온도 센서가 고농도의 분뇨에 내구성이 급감하여 교체 후에도 데이터 확보를 못한 사례가 있었던 바 협동연구기관인 (주) 동문이엔티에서 보다 내구성이 높은 센서를 선정하여 설치하였다. 또한 A 농가의 경우 급격한 수위 변동에 의해 센서가 공기중에 노출될 가능성이 있으므로 센서의 설치 깊이를 재조정하였으며, 센서 전극부위에 분뇨 고형물이 침적되는 것을 방지하고자 주기적으로 센서를 세척할 수 있는 시스템을 구축하였다.

약취센서의 경우 기존에는 수위, 분뇨 물성센서와 나란히 위치하여 측정을 하였으나, 저장조 내 분뇨의 표면과 지나치게 가깝고, 분뇨에서 발생된 수증기가 직접적으로 약취센서의 측정부에 닿아 수분에 의한 센서 내부의 부식 및 고장이 발생하거나 NH<sub>3</sub> 같은 흡착성 가스로 인해 데이터의 민감도가 낮아지는 문제가 지속적으로 발생하였다. 따라서 모니터링 시스템의 보완 시 분뇨 저장조에서 확산되는 가스 수분의 센서에 대한 영향을 최소화하고자 1시간을 주기 (30분 약취 측정 / 30분 센서 세척)로 약취 센서의 측정부를 주기적으로 세척하는 시스템을 추가로 구축하였다.



약취 측정 및 세척 시스템 구축

센서로 유입되는 공기의 흐름  
(a) 측정-약취; (b) 세척-신선한 공기

그림 3-28. 약취 측정 및 세척 시스템의 모습과 공기의 흐름

약취 센서로 유입되는 공기의 흐름을 2개의 방향으로 나누고 각 방향마다 시간을 설정하여 공기의 흐름을 제어할 수 있는 전동밸브를 설치하였다. 약취물질의 측정 시 분뇨 저장조 방향의



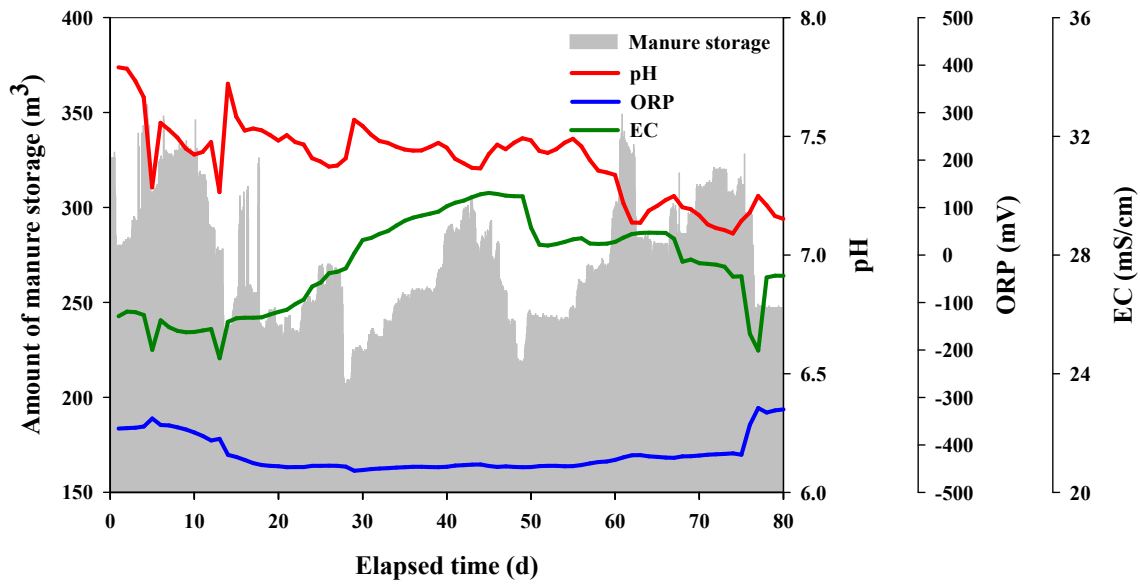
밸브가 열리고 외부 방향의 밸브가 닫혀 가스가 센서로 유입되게하고 (그림 3-28의 (a)), 측정 후 센서의 세척을 위해 분뇨 저장조 방향의 밸브가 닫히고, 외부 방향의 밸브가 열려 신선한 공기가 센서의 측정부로 유입됨으로써 악취 가스에 노출 시 흡착되어있는 가스들을 세척하도록 하여 센서의 민감도 유지와 센서 수명 연장 효과를 얻을 수 있도록 하였다. 시스템 내부에 공기의 흐름을 만들기 위해 시스템 내 공기 유출부에는 송풍기 (150 m<sup>3</sup>/h)를 부착하였고, 자연적으로 확산에 의한 배출이 아니기 때문에 각 악취 데이터 확보 시 농도 단위가 아닌 발생량 단위로 환산하여 산출하였다.

#### 4-2. 개선된 모니터링 시스템의 측정 결과 분석

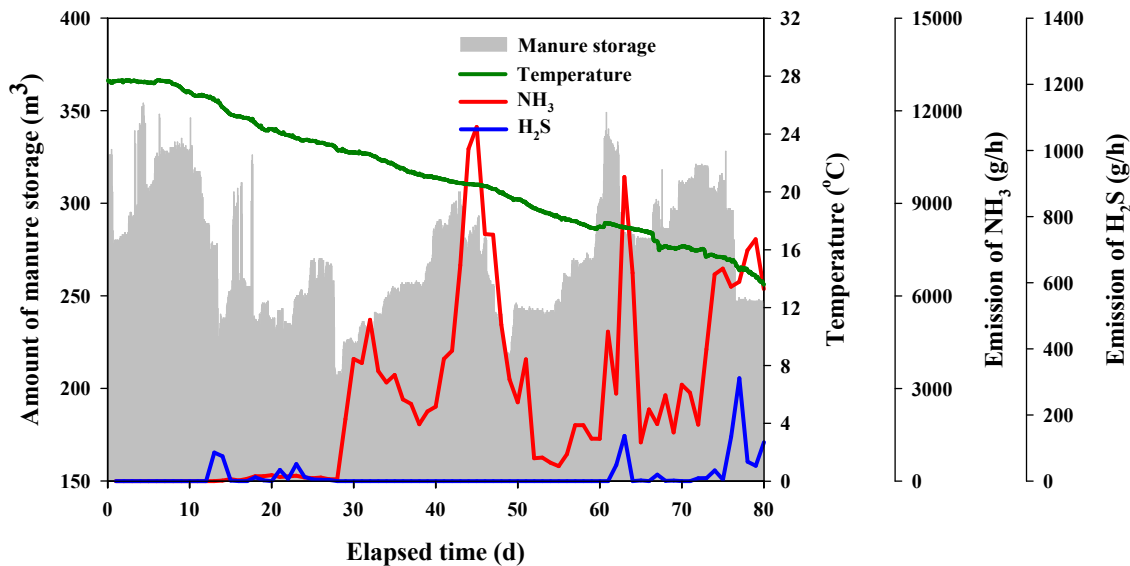
##### (1) 분뇨 저장조 모니터링 결과 분석 (분뇨 물성 및 악취)

A 농가의 분뇨 저장조 모니터링 결과, 전 기간에 걸쳐 센서의 교체 없이 분뇨 물성 및 악취 데이터를 모두 확보할 수 있었다. 앞선 B 농가의 분석 결과처럼 분뇨 물성 (pH, ORP, EC) 데이터는 그 당시 분뇨의 성상을 대변할 수 있으나 적정 수거시점을 파악하기 위한 분뇨 저장량과의 연관성은 확인하기 어려웠다. 그러나 지속적으로 온도가 감소함에 따라 pH도 지속적으로 하락하는 경향을 보여 주변 환경영향에 의해 분뇨 내 물성도 영향을 받는 것으로 예상된다.

악취물질의 경우 시스템 설치 후 정상적인 측정이 가능했던 28일 이후부터의 결과를 살펴보면 NH<sub>3</sub>의 경우 저장조 내 분뇨의 양이 급격하게 변화할 때 높은 발생량을 보이는 것으로 나타났으며, 이후 분뇨 저장량이 꾸준히 증가하는 시점에는 NH<sub>3</sub> 발생량이 다시 낮아지는 경향을 나타냈다. 이는 저장 수준보다는 내부에서 발생하는 유동이 NH<sub>3</sub>의 발생량에 미치는 영향이 더 크다는 것을 의미한다. 또한 평균 온도가 꾸준히 감소함에 따라 유사한 분뇨 저장량을 가짐에도 발생량은 낮아지는 경향을 확인할 수 있었다. H<sub>2</sub>S의 경우 전반적으로 낮은 발생량을 나타내었으며, 저장량에 따른 연관성을 확인할 수 없었다 (그림 3-29).



저장량 & 분뇨 물성 (pH, ORP, EC)의 변화



저장량 & 온도 & 악취 농도 (NH<sub>3</sub>, H<sub>2</sub>S)의 변화

그림 3-29. 개선된 모니터링 시스템의 운영 시 각 센서별 측정 결과 - A 농가

(2) 분뇨 저장조 내 모니터링 인자간 상관관계 분석

피어슨 상관계수 분석결과, 분뇨 저장량은 pH (-0.699), 온도 (-0.446)과 통계적으로 유의적인 뚜렷한 음의 상관관계 ( $p < 0.01$ )를 나타내는 것으로 분석되었으며, NH<sub>3</sub> (0.313)와는 통계적으로 유의적인 뚜렷한 양의 상관관계 ( $p < 0.01$ )를 나타내고 있다. 즉 분뇨 저장량이 감소할수록 pH와 온도는 높게 나타났고, 분뇨 저장량이 많아질수록 NH<sub>3</sub> 발생량은 증가하였다. 분뇨 저장량

과 NH<sub>3</sub> 상관관계의 강도는 B 농가 모니터링 결과로 산출된 분뇨 저장량과 NH<sub>3</sub> 상관관계 강도 (0.390)와 유사하게 분석되었다. 그 외 통계적으로 유의적인 강한 양의 상관관계 ( $p < 0.01$ )를 가진 항목으로는 온도와 pH (0.837), H<sub>2</sub>S와 ORP (0.912)가 있었으며, 음의 상관관계 ( $p < 0.01$ )는 H<sub>2</sub>S와 온도 (-0.761) 사이의 강도가 가장 높게 분석되었다. NH<sub>3</sub>의 경우 온도 (-0.524)와 pH (-0.445)와 통계적으로 유의적인 뚜렷한 음의 상관관계를 지니는 것으로 분석되었으며, 이 역시 B 농가에서 산출한 결과와 유사하였다. H<sub>2</sub>S는 pH (-0.638), EC (-0.595)와 통계적으로 유의적인 뚜렷한 음의 상관관계 ( $p < 0.01$ )을 나타내었다 (표 3-21).

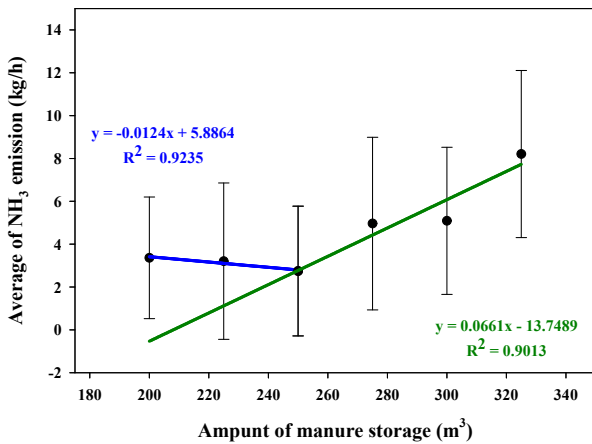
표 3-21. 분뇨 저장조 모니터링 인자간 상관관계 (A 농가)

	pH	ORP	EC	TEMP	저장량	저장 변동량	NH <sub>3</sub> (g)
ORP	-0.574						
p-value	<.0001						
EC	0.158	-0.574					
p-value	0.189	<.0001					
TEMP	<b>0.837</b>	-0.690	0.220				
p-value	<.0001	<.0001	0.066				
저장량	-0.699	0.133	0.020	-0.446			
p-value	<.0001	0.271	0.865	<.0001			
저장 변동량	-0.002	0.096	-0.126	0.052	0.116		
p-value	0.984	0.426	0.294	0.668	0.337		
NH <sub>3</sub> (g)	-0.445	0.481	-0.153	-0.524	0.313	0.153	
p-value	0.0001	<.0001	0.204	<.0001	0.008	0.204	
H <sub>2</sub> S (g)	-0.638	<b>0.912</b>	-0.595	<b>-0.761</b>	0.110	-0.035	0.535
p-value	<.0001	<.0001	<.0001	<.0001	0.363	0.770	<.0001

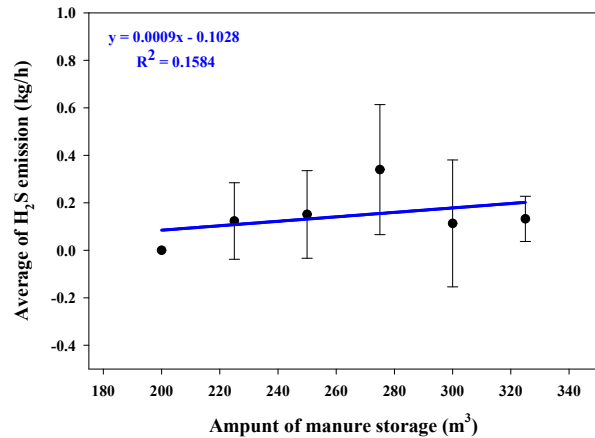
앞선 B 농가 모니터링 데이터 분석을 통해 적정 수거시점 산정 시 주요 인자로 이용되는 수위 (분뇨 저장량)와 더불어 악취물질의 농도를 보조인자로 사용하였으며, 이에 분뇨 저장량 및 온도에 따른 악취물질 발생량과 관계식을 분석하였다.

분석결과 분뇨 저장량 200 ~ 250 m<sup>3</sup>까지 NH<sub>3</sub>의 발생량은 0.0124 kg/h/m<sup>3</sup>씩 선형적으로 감소하는 경향을 보였으며 (R<sup>2</sup>=0.9235), 평균 3.1 kg/h 수준으로 발생하였고, 분뇨 저장량 250 ~ 325 m<sup>3</sup> 에서 0.0661 kg/h/m<sup>3</sup>씩 선형적으로 상승하는 경향을 보여주었다 (R<sup>2</sup>=0.9013). H<sub>2</sub>S 발생량의 경우 분뇨 저장량에 따른 유의적인 선형 회귀식을 도출할 수 없었다.

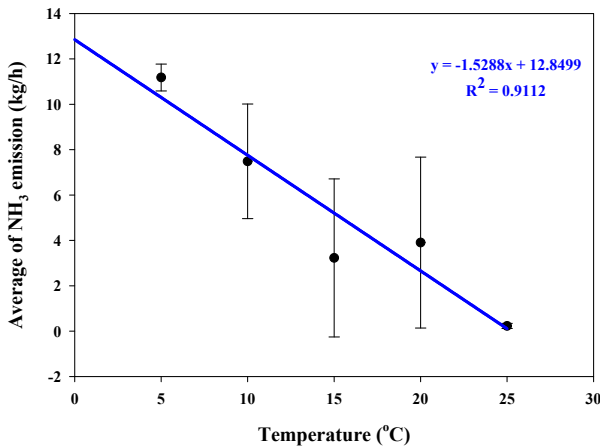
온도 영향을 분석한 결과, NH<sub>3</sub> 발생량은 0 ~ 25°C 사이에서 -1.5288 kg/h/°C 씩 선형적으로 감소하는 경향이 나타났으며 (R<sup>2</sup>=0.9112), 황화수소의 경우 온도가 상승함에 따라 꾸준히 상승하는 경향을 보였으나 낮은 선형적 상관관계를 확인할 수 있었다 (그림 3-30).



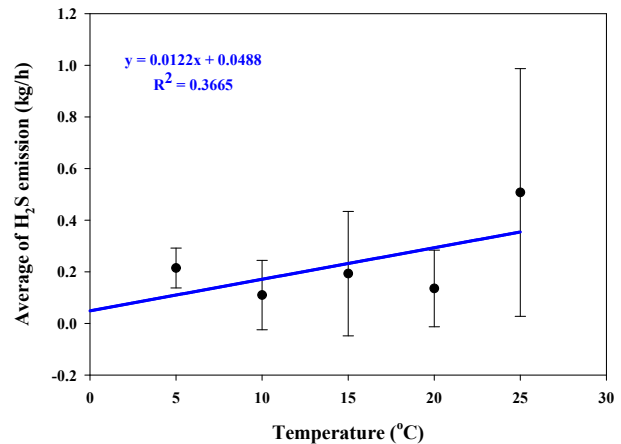
분뇨 저장량에 따른 NH<sub>3</sub> 농도 변화



분뇨 저장량에 따른 H<sub>2</sub>S 농도 변화



온도에 따른 NH<sub>3</sub> 농도 변화



온도에 따른 H<sub>2</sub>S 농도 변화

그림 3-30. 저장조 모니터링 인자에 따른 NH<sub>3</sub>, H<sub>2</sub>S 농도 변화 (A 농가)

분뇨 저장량과 악취물질의 선형적 상관관계 분석 결과, 분뇨 저장량 250 m<sup>3</sup> 이상에서 NH<sub>3</sub>의 농도가 급격하게 증가하는 것으로 나타남에 따라 분뇨 저장량을 250 m<sup>3</sup> (수위: 2.5 m) 이하로 유지하는 것이 A 농가의 수거조건으로 적합할 것으로 사료된다. 전체 모니터링 기간이 약 3달 수준으로 적기 때문에 보다 정밀한 수거시점을 선정하기 위해서는 1년 이상의 장기 데이터 확보 후 결정하는 것이 바람직하나 연구과제의 기간상 해당 조건을 활용하였다.

분뇨 저장량 250m<sup>3</sup>는 A 농가 분뇨 저장조 유효용적의 62.5%에 해당하는 수준으로 모니터링 기간 중 온도 범위인 0 ~ 25°C 범위 내에서 적용할 수 있을 것으로 판단되며, 위 조건을 유지 시 발생하는 악취물질의 저감율을 산출한 결과, A 농가의 전체 모니터링 기간 중 NH<sub>3</sub>와 H<sub>2</sub>S의 평균 발생량은 각각 3.86±3.74, 0.16±0.21 kg/h였으며, 분뇨 저장량 250 m<sup>3</sup> 이상의 조건에서 NH<sub>3</sub>와 H<sub>2</sub>S의 평균 농도는 각각 4.28±3.80, 0.19±0.25 kg/h, 분뇨 저장량 250 m<sup>3</sup> 이하의 조건에서는 각각 3.23±3.55, 0.12±0.16 kg/h로 산출되었다. B 농가의 저장조 내 분뇨 저장량을 250 m<sup>3</sup> 이하로 유지 시 악취물질의 저감율은 전체 모니터링 기간 대비 NH<sub>3</sub>, H<sub>2</sub>S 각각 16.3, 20.7%로 나타났고, 분뇨 저장량 250 m<sup>3</sup> 이상의 조건 대비 각각 24.5, 36.5%로 산출되었다 (표 3-22).

표 3-22. A 농가 분뇨 저장조의 수거시점 적용 시 악취저감 효과

기준	악취 평균 농도 (ppm)			분뇨 저장량 고정 시 악취저감 효과 (%)		
	모니터링 전 기간 (a)	70 m <sup>3</sup> 이상 (b)	70 m <sup>3</sup> 이하 (c)	전 기간 기준 저감율 [(a-c)/a*100]	70 m <sup>3</sup> 이상 기준 저감율 [(b-c)/b*100]	
악취 종류	NH <sub>3</sub>	3.86±3.74	4.28±3.80	3.23±3.55	16.3	24.5
	H <sub>2</sub> S	0.16±0.21	0.19±0.25	0.12±0.16	20.7	36.5

## <가축분뇨 발생량 모니터링시스템을 평가하기 위한 검정방법 및 검정시스템 구축, 한국농업기술진흥원>

### 1. 상용화 및 연구된 가축분뇨 모니터링 기자재 스펙 및 성능지표 조사

- 가축분뇨의 처리 분야에서 생물학적 처리조가 아닌 슬러리 피트와 분뇨 저장조에 센서를 설치하여 고정적으로 모니터링 선행연구가 거의 없으므로 본 과제에서는 선정된 측정요소를 위한 측정하기 위하여 스마트축사에서 활용하기 위하여 스마트팜 코리아에 등록된 센서를 대상으로 조사하였음
- 연구팀 중 강원대학교에서는 분뇨 저장조 내의 가축분뇨를 모니터링하기 위한 핵심인자로 슬러리피트(저장조)의 수위, 보조인자로 온습도, 악취 발생 특성요소(pH, ORP, NH<sub>3</sub>, H<sub>2</sub>S, VOC) 등을 선정하였음
- (레벨센서(수위)) 슬러리분뇨의 수위를 측정하기 위한 레벨센서는 접촉식 레벨센서(플로트 스위치)와 비접촉식 레벨센서(초음파 레벨센서) 등으로 나누어지며 슬러지분뇨의 경우에는 센서를 오염시킬 수 있는 물질을 많이 포함하고 있으므로 비접촉식을 활용하거나 접촉식을 활용할 경우에는 필터 등을 부착할 필요성이 있음
- (온습도) 온도나 습도를 측정하기 위한 센서 또한 접촉센서와 비접촉센서로 나누어지는데, 접촉센서의 경우 피측정물에 직접 센서를 접촉하여 측정하며, 비접촉센서는 적외선으로 온도를 감지하는 등의 센서가 대표적임
- 온도나 습도를 측정하기 위한 센서는 측정하기 위한 감지위치가 오염될 경우에는 데이터의 신뢰성 등에 대한 문제가 발생하고, pH 센서의 온도 보정등에도 영향을 주므로 최대한 비접촉식을 활용하는게 바람직함
- (pH, ORP) pH센서는 전극을 활용하여 측정하게 되는데 전극부분이 오염될 경우에 데이터 신뢰성에 대한 문제점이 발생하므로 가축분뇨를 모니터링 하기 위하여 다른 지표를 검토하는 것도 필요하다고 판단되나 pH나 ORP 등이 가축분뇨를 모니터링 하는데 중요한 역할을 하므로 전극을 오염시키지 않기 위한 필터 등이 부착된 센서를 활용필요
- (NH<sub>3</sub>, H<sub>2</sub>S) 암모니아나 황화수소의 경우에는 전기화학식 가스센서를 활용하는데 일반적으로 먼지가 많은 환경에서 활용되므로 측정부가 오염될 경우에는 센서를 교체할 필요성이 있음
- 축사에서 활용되고 있는 암모니아나 황화수소 센서의 경우에는 6개월 마다 교체를 하는 것을 권장하고 있음
- (VOC) VOC는 감소된 가스 분자 수에 비례하여 전도성이 변하는 센서를 의미하는데 혼성 가스 센서로 여러 가지 가스에 대한 것을 측정하므로 가스의 타입이나 PPM 등을 보여주지는 않는 문제점이 있음
- 그러나, VOC가 악취에 영향을 미친다는 것은 타 연구에서 증빙된바가 있으므로 측정할 필요성이 있음
- (공통) 공통적으로 분진, 고습, 슬러지 및 유해가스 등이 있는 환경에서 운영되는 센서는 비접촉식이되 내환경성이 확보된 물건이 필요하나, pH 센서의 경우에는 비접촉식이 없으

므로 지속적인 관리가 필요

- 비접촉식의 경우에도 유해가스와 먼지 등의 문제로 측정부가 오염되기 때문에 6개월 마다 교체하는 것을 고려할 필요성이 있음
- 스마트팜 코리아에 축산기자재로 등록된 레벨센서, pH 센서, ORP 및 VOC 등은 존재하지 않음
- 암모니아 및 온도, 습도 센서 등의 스펙을 조사한 결과 일반적으로  $\pm 3\%$  이내의 오차율을 가지는 것으로 나타났으며 이는 성능기준으로 제시가 가능할 것으로 판단됨
- 레벨, VOC, pH, ORP 등의 센서는 축산에 활용하기 위한 것은 없으나 비슷한 환경을 가질 것으로 판단되는 오수처리장 등에서 사용되는 것을 추가적으로 조사하였음

순번	센서명	측정대상	제조업체	사진	제원
1	396-MNS2 -8W2-TSS TL03	온도	Monnit		Power : 2.0 ~ 3.8 VDC 계측범위 : $-7 \sim 60\text{ }^{\circ}\text{C}$ 측정오차 : $\pm 1\%$ 이내
2	YDTH-06- B-A	온도·습도	VCC-GN D studio		Power : 2.0 ~ 3.8 VDC 계측범위 : $-20 \sim 70\text{ }^{\circ}\text{C}$ (온도) $0 \sim 80\%RH$ (습도) 측정오차 : $\pm 1\%$ 이내
3	AM2302	온도·습도	AOSONG (ASAIR)		Power : 3.5 ~ 5.5 DC 계측범위 : $-40 \sim 80\text{ }^{\circ}\text{C}$ (온도) $0 \sim 100\%RH$ (습도) 측정오차 : $\pm 0.5\text{ }^{\circ}\text{C}, \pm 3\%RH$ 이내
4	WSN-802	온도·습도	우성하이텍 주)		Power : 220 VAC 계측범위 : $-19 \sim 60\text{ }^{\circ}\text{C}$ (온도) $0 \sim 99.99\%RH$ (습도) 측정오차 : $\pm 0.5\text{ }^{\circ}\text{C}, \pm 2.5\%RH$ 이내



순번	센서명	측정대상	제조업체	사진	제원
5	THD-DD1-T	온도·습도	오토닉스		Power : 24 VDC 계측범위 : -19.9 ~ 60 °C(온도) 0 ~ 99.9%RH(습도) 측정오차 : ±0.8°C, ±2%RH 이내
6	SH-VT260	온도·습도· CO <sub>2</sub>	소하테크		Power : 12 VDC ~ 24V 200mA 이상 계측범위 : -25 ~ 85 °C(온도) 0 ~ 99.9%RH (습도) 0 ~ 3000ppm, 0 ~ 5000ppm, 0 ~ 10000ppm, (CO <sub>2</sub> ) 측정오차 : ±0.3°C, ±2%RH , ±3% 이내
7	HCPV-220 NH	온도·습도	CONOTEC		Power : 5 VDC 계측범위 : -40 ~ 85 °C(온도) 0 ~ 100%RH(습도) 측정오차 : R(25°C) = 10kΩ ±1%, ±3%RH 이내
8	FTS-30	온도	우성하이텍 (주)		Power : 220 VAC 계측범위 : -19 ~ 60 °C(온도) 측정오차 : ±1% 이내
9	THD-R-T	온도·습도	오토닉스		Power : 24 VDC 계측범위 : -19.9 ~ 60 °C(온도) 0 ~ 99.9%RH(습도) 측정오차 : ±1°C, ±3%RH 이내
10	GOTO-420	온도	지니스 코리아		Power : 15~30 VDC 계측범위 : -20 ~ 50 °C(온도) 측정오차 : ±0.3°C이내
11	KSH-7310	온도·습도	코리아 디지털		Power : 5 VDC 계측범위 : -40 ~ 60 °C(온도) 0 ~ 99%RH(습도) 측정오차 : ±0.3°C, ±3%RH 이내

순번	센서명	측정대상	제조업체	사진	제원
12	RHS+ HUMIDITY SENSOR	온도·습도	Munters		Power : 12 VDC 계측범위 : -10~70°C(온도) 0 ~ 100%RH(습도) 측정오차 : ±3%RH이내
13	HM1500LF (HPP805A 031)	습도	메저먼트 스페셜티스		Power : 5 VDC 계측범위 : 0 ~ 100%RH(습도) 측정오차 : ±2%RH이내
14	DOL 114	온도·습도	DOL Sensors		Power : 11 ~ 30 VDC 계측범위 : -40~60°C(온도) 0 ~ 100%RH(습도) 측정오차 : ±0.5°C, ±3%RH 이내
15	WT-1000B	온도·습도· EC	미래센서		Power : 9 ~ 15 VDC 계측범위 : 0 ~ 60°C(온도) 0 ~ 99.9%RH(습도) 0 ~ 6.0dS/m(EC) 측정오차 : ±0.5°C, ±1%RH, ±1% 이내
16	UA54-NH3	암모니아 온도	데키스트		Power : 2.0 ~ 3.8 VDC 계측범위 : 0~100 ppm(NH <sub>3</sub> ) -40 ~ 85°C(온도) 측정오차 : ±3 % 이내
17	NH3/CR-2 00-CAL	암모니아	MEMVR APOR		Power : - (저항변화) 계측범위 : 0~100 ppm(NH <sub>3</sub> ) 측정오차 : ±5 % 이내
18	ZEOS-NH 3	암모니아 이산화탄소 이산화황	WINSEN		Power : 3.7 ~ 9.0 VDC 계측범위 : 0~100 ppm(NH <sub>3</sub> ) 0~1,000 ppm(CO <sub>2</sub> ) 0~20 ppm(SO <sub>2</sub> ) 측정오차 : 자료없음
19	DA-100	암모니아 등 독성가스	GASDNA		Power : 20 ~ 30 VDC 계측범위 : 0~100 ppm(NH <sub>3</sub> ) 측정오차 : ±1 % 이내

순번	센서명	측정대상	제조업체	사진	제원
20	RG10-NH3-100	암모니아	데키스트		Power : 12 VDC 계측범위 : 0~100ppm(NH <sub>3</sub> ) 측정오차 : ±3 % 이내
21	NH3/MR-100	암모니아	데키스트		Power : - (저항변화) 계측범위 : 0~100ppm(NH <sub>3</sub> ) 측정오차 : ±3 % 이내

표 4-5 수질측정기 1

순번	센서명	측정대상	제조업체	사진	제원
1	S200-ORP	ORP	SECHANG G		Power : 220 VAC 계측범위 : -1,000~1,000 mV 정확도 : 10 mV
2	M-시리즈	수온 EC ORP pH 수위 (옵션추가 가능)	기술과 환경 (주)		Power : 12 VDC 계측범위 : -5 ~ 75 °C(수온) 0 ~ 100 mS(EC) -1,200 ~ 1,200 mV(ORP) 0 ~ 14 (pH) 0 ~ 60 m (수위) 정확도 : ±0.1 °C(수온) ±1 %(EC) ±20 mV(ORP) ±0.2(pH) ±0.15 m(수위)
3	Manta 시리즈	수온 EC ORP pH 수위 (옵션추가 가능)	(주) 이스텍		Power : 배터리팩(정보없음) 계측범위 : -5 ~ 70 °C(수온) 0 ~ 112 mS(EC) -999 ~ 999 mV(ORP) 0 ~ 14 (pH) 0 ~ 10 m (수위) 정확도 : ±0.1 °C(수온) ±1 %(EC) ±20 mV(ORP) ±0.2(pH) ±0.003 m(수위)
4	AP시리즈	수온 EC ORP pH 수위 (옵션추가 가능)	YOUNG IN ST		Power : 배터리팩(정보없음) 계측범위 : -5 ~ 70 °C(수온) 0 ~ 112 mS(EC) -1000 ~ 1000 mV(ORP) 0 ~ 14 (pH) 0 ~ 100 m (수위) 정확도 : ±0.1 °C(수온) ±1 %(EC) ±5 mV(ORP) ±0.1(pH) ±1 %(수위)

표 4-6 수질측정기 2

순번	센서명	측정대상	제조업체	사진	제원
5	MK45	수온 EC ORP pH 수위	하이드로 넷		Power : 배터리팩(정보없음) 계측범위 : -5 ~ 50 °C(수온) 0 ~ 100 mS(EC) -999 ~ 999 mV(ORP) 0 ~ 14 (pH) 0 ~ 20 m (수위) 정확도 : ±0.1 °C(수온) ±1 %(EC) ±25 mV (ORP) ±0.2 (pH) ±0.1 %(수위)
6	ASPRT215 1D	수온 ORP pH			Power : 배터리팩(정보없음) 계측범위 : 0 ~ 110 °C(수온) 정보없음 (ORP) 0 ~ 14 (pH) 정확도 : 정보없음

## 2. 전기적·기계적 연결규격 제시

- 전기적·기계적 연결규격의 경우, 한국농업기술진흥원에서 금년도 농림축산식품부와 제정 추진한 국가표준의 표준을 활용한 연결방법을 준수하는 것이 맞다고 판단하였으나, 기종별로 매우 다양한 센서를 장착한 수질측정 센서의 경우에는 기존의 연결규격을 적용하는 것은 불가능하다고 판단하였음
- 따라서, 수질측정 센서 중 2 기종을 구매하여 연결규격 등을 조사하고 축산기자재 국가표준 개정에서 반영하도록 의견을 제출하고자 하였음 (한국농업기술진흥원 스마트팜사업팀 협의)
- 기존 축산기자재 중 센서에 대한 기계적·전기적 연결규격은 다음과 같음
- 수질측정 센서 중 기술과창조(주)에서 제작한 M-1000의 센서부, 연결부, 케이스 등을 조사하였음
- M-1000은 현장에서 수질을 측정하기 위한 센서로 실제 오염 등에 대비하기 위하여 연결커넥터부의 이물질 방지캡, 센서부 슬러지 오염방지를 위한 캡, 센서를 부착하기 위한 지그 등이 부착되어있음
- 실제로 가축분뇨 저장고에서 활용하기 위해서 이물질 방지캡, 슬러지 오염방지 캡 등의 기계적 장치는 기본적인 구조에 포함하여야 할 것으로 판단되며 캡, 센서 등이 황하수소나 암모니아에서 실제 견딜 수 있는지를 판단할 필요성이 있음
- 기존 센서 연결규격으로는 전원, 신호선 1,2, 그라운드 등의 4선을 최대로 판단하였으나 이러한 복합센서를 포함하기 위하여 신호선 수를 증가시킬 필요성이 있음
- 시설온실 스마트팜 분야에서는 통신호환성 부분을 확보하기 위하여 '디폴트 레지스트맵'에 대한 확산사업을 진행하고 있으므로 수질측정센서가 아닌 가축분뇨를 모니터링 하기 위한 센서에서는 이러한 통신프로토콜을 표준으로 제정하여 운용할 필요성이 있음

표 3-7 축산기자재 센서 기계적 전기적 연결규격

기계적 연결장치 규격				전기적 연결장치 규격		작동 방식
접속 단자	계전기 형식*	작동 신호 결선식별	동력선 식별	작동 신호 전압	동력선 전압	
① 커넥터 ② 터미널 단자	<직류> ① 2A2B	<직류> ① + : 적색 ② 접지: 검정색	<직류> ① + : 적색 ② 접지: 검정색	<직류> ① 5V ② 12V ③ 24V ④ 48V	<직류> ① 5V ② 12V ③ 24V ④ 48V	① 스위치방향 (상하) ② 스위치 작동 모드 (ON OFF)
	<교류> ① 220V(단상) ② 220V(삼상) ③ 380V(삼상)	<교류> ① 신호 : 0 ② 접지 : G	<교류단상> ① R상 : 적색 ② T상 : 검정색	<교류> ① 220V 60Hz	<교류> ① 220V(단상) ② 220V(삼상) ③ 380V(삼상)	
			<교류삼상> ① R상 : 적색 ② S상 : 갈색 ③ T상 : 검정색			

### 3. RS485 통신프로토콜 표준(안) 제시

- 축산분야에서 ICT기자재를 활용한 장비는 지속적으로 운용되고 왔으며 표준화요소 발굴을 위하여 현재 활용되고 있는 통신방법 등의 체계를 확인할 필요성을 인지하였음
- 스마트 축사에서 가축분뇨를 모니터링하기 위해 활용되는 센서는 암모니아, 황하수소, 이물질 및 먼지 등이 많이 발생하여 이러한 부분에 대하여 내환경성이 확보된 장비를 권장하고 있음
- 가축분뇨 저장고 외부의 모니터링을 하기 위한 센서는 일반적으로 활용되는 기상대에 비하여 암모니아, 황하수소 등에 대한 영향을 많이 받는 위치에 설치되므로 이에 대한 내환경성을 충분히 확보할 것을 권장하고 있음
- 현재 표준화가 완료된 센서 중 가축분뇨 저장고 모니터링을 위한 센서는 온도, 습도, 암모니아 및 이산화탄소 등이 있으며 이에 대한 내용을 국가표준인 KS X 3279 에 준수여부와 현재 사용 중인 통신방식을 조사한 결과 다음과 같이 나타났음

표 3-8. 축산 스마트팜 업체 조사대상

연번	업체명	연번	업체명	연번	업체명
1	주식회사 근육	20	코리아데이터코퍼레이션(주)	39	아주산업
2	가농인터내셔널	21	태원트레이드	40	(주)혜송
3	그리심산업	22	태진테크	41	주식회사 동성코코팠
4	데키스트	23	팜플러스	42	이정산업시스템
5	문터스코리아	24	하이템	43	(주)씨아이티이엔지
6	삼성산업	25	범진시스템	44	주식회사 금보
7	샘스	26	한별이레	45	(주)프리에너지
8	성광토탈시스템	27	시스트로닉스(주)	46	주식회사 두리
9	씨아이티시스템	28	산일냉열	47	태창바이오(주)
10	디티씨	29	성진냉열	48	현대축산기계주식회사 농업회사법인
11	아이온텍	30	주식회사 유로젠	49	대성바이오텍
12	주식회사 우리농장연구소	31	대산테크	50	농업회사법인(주)진영케이지
13	(주)이모션	32	이화글로벌	51	웰돈산업
14	제이에이테크	33	제이씨텍	52	(주)건지
15	주식회사성광	34	알림시스템	53	(주)비엔씨하이텍
16	주식회사 성광이엔지	35	리얼팜	54	(주)유샘
17	주식회사 정림	36	주식회사 대로	55	대성이엔지
18	지원	37	과학시스템	56	주식회사 삼우
19	케어시스템	38	(주)보람	57	

- 온도센서(외기)는 총 56개 업체에서 47개 모델을 생산하고 있으며 표준에서 제시



하고 있는 측정범위, 측정오차, 통신방식 등을 준수여부를 조사한 결과, 표 3과 같이 나타났음

표 3-9. 온도센서(외기) 측정범위 및 측정오차, 통신방식 등 조사결과

	측정범위	측정오차		통신방식
기준	(-40 ~ 60) °C	(±0.5) °C	RS-485	25
준수	28	26	기타	20
비준수	19	21	비고 (미응답)	2

- 업체를 대상으로 조사한 결과, 총 47개의 모델 중 28개 모델이 측정범위를 준수하고 있었으며 19개는 준수하지 않고 있는 것으로 나타났음
- 주요 미준수 내용으로는 내부 온도센서와 동일하게 영하 30 °C 또는 영하 20 °C를 측정한계로 하는 경우가 많았으며, 실제로 국내에서 역대 최저기온은 영하 32.6 °C이므로 표준에서 제시하고 있는 외기 온도센서의 경우에는 측정범위를 유지한 상태에서 업체에서 이러한 부분을 준수하는 방향으로 유도하는 것이 좋다고 사료됨
- 또한, 센서 제조업체에서 제시하고 있는 스펙을 근거로 확인하였을 때, 측정오차의 경우에는 47개 중 26개의 센서가 측정오차를 준수하고 있는 것으로 나타나 정확도수준에 대해서도 표준을 준용하도록 유도할 필요성이 있다고 사료됨
- 데이터 취득 방식의 경우에는 데이터를 취득하기 위한 복합환경제어기 또는 PC, MCU 등의 위치가 내기센서 등에 비하여 상대적으로 원거리에 설치되어 RS-485 방식을 25개가 사용하고 나머지는 아날로그 데이터를 직접 측정하거나 Wifi, USB-AT, 전용통신선 등을 활용하는 것으로 나타났음
- 습도센서(외기)는 총 24개 모델을 생산하고 있으며 표준에서 제시하고 있는 측정범위, 측정오차, 통신방식 등을 준수여부를 조사한 결과, 표 5와 같이 나타났음
- 업체를 대상으로 조사한 결과, 총 24개의 모델 중 24개 모델이 측정범위를 준수하는 것으로 나타났음

표 3-10. 습도센서(외기) 측정범위 및 측정오차, 통신방식 등 조사결과

	측정범위	측정오차		통신방식
기준	(0 ~ 100) %	(±5) %	RS-485	20
준수	24	24	기타	2
비준수	0	0	비고 (미응답)	2

- 센서 제조업체에서 제시하고 있는 스펙을 근거로 확인하였을 때, 측정오차의 경우에는 24개 중 24개의 센서가 측정오차를 준수하고 있는 것으로 나타났으나 조사 결과에서는 (±2 ~ 3) % 범위의 측정오차가 대다수 이므로 표준기준을 수정할 필요가 있을 것으로 사료됨
- 데이터 취득 방식의 경우에는 24개 중 20개 모델이 RS485를 준용하여 활용하고 있었으며 2개 모델은 아날로그 방식, 2개는 미응답으로 나타났음
- 암모니아 센서는 총 31개 모델을 생산하고 있으며 표준에서 제시하고 있는 측정 범위, 측정오차, 통신방식 등을 준수여부를 조사한 결과, 표 7과 같이 나타났음
- 업체를 대상으로 조사한 결과, 총 31개의 모델 중 31개 모델이 측정범위를 준수 하는 것으로 나타났음

표 3-11. 암모니아 센서 측정범위 및 측정오차, 통신방식 등 조사결과

	측정범위	측정오차		통신방식
기준	(0 ~ 100) ppm	(±5) %	RS-485	23
준수	31	24	기타	8
비준수	0	7	비고 (미응답)	0

- 암모니아센서는 측정범위가 두가지 형태로 나타났는데 최대 측정범위가 100 ppm 이 대다수였으며, AMPHENOL SGX SEOSNRTECH에서 생산한 암모니아 센서만 최대 측정범위가 300 ppm 인 것으로 나타나 측정범위에 대한 표준변경은 필요가 없을 것으로 사료됨
- 또한, 센서 제조업체에서 제시하고 있는 스펙을 근거로 확인하였을 때, 측정오차의 경우에는 31개 중 24개의 센서가 측정오차를 준수하고 있는 것으로 나타났으나 조사결과에서는 (±3 ~ 5) % 범위의 측정오차가 대다수였으며 ±10% 오차를

가지는 장비도 있는 것으로 나타났는데 암모니아의 경우 농가에서 가장 많이 활용 되는 영역이 (0 ~ 10) ppm 이므로 큰 오차를 가지는 센서는 현장보급이 안되도록 할 필요성이 있음

- 데이터 취득 방식의 경우에는 31개 중 23개 모델이 RS485를 준용하여 활용하고 있었으며 8개 모델은 아날로그 방식으로 나타났음
- 축산에서 사용할 수 있는 축산용 RS485 Modbus 통신프로토콜을 제시하고, 단체 표준(안) 제시하여, 2021년 스마트팜 국가표준 확산사업에 활용 중에 있으며, 수위센서 등을 포함하여 단체표준 제정 요청

※ 스마트팜용 센서/모드 제품명 제제사(타행)인

통신방식	제 품 명	제 품 명	제 품 명
1	기온센서	1	기온센서
2	기압센서	2	기압센서
3	기습센서	3	기습센서
4	기습센서	4	기습센서
5	기습센서	5	기습센서
6	기습센서	6	기습센서
7	기습센서	7	기습센서
8	기습센서	8	기습센서
9	기습센서	9	기습센서
10	기습센서	10	기습센서
11	기습센서	11	기습센서
12	기습센서	12	기습센서
13	기습센서	13	기습센서
14	기습센서	14	기습센서
15	기습센서	15	기습센서
16	기습센서	16	기습센서
17	기습센서	17	기습센서
18	기습센서	18	기습센서
19	기습센서	19	기습센서
20	기습센서	20	기습센서
21	기습센서	21	기습센서
22	기습센서	22	기습센서
23	기습센서	23	기습센서
24	기습센서	24	기습센서
25	기습센서	25	기습센서
26	기습센서	26	기습센서
27	기습센서	27	기습센서
28	기습센서	28	기습센서
29	기습센서	29	기습센서
30	기습센서	30	기습센서
31	기습센서	31	기습센서

※ 제제사별

제 품 명	제 품 명
1	기온센서
2	기압센서
3	기습센서
4	기습센서
5	기습센서
6	기습센서
7	기습센서
8	기습센서
9	기습센서
10	기습센서
11	기습센서
12	기습센서
13	기습센서
14	기습센서
15	기습센서
16	기습센서
17	기습센서
18	기습센서
19	기습센서
20	기습센서
21	기습센서
22	기습센서
23	기습센서
24	기습센서
25	기습센서
26	기습센서
27	기습센서
28	기습센서
29	기습센서
30	기습센서
31	기습센서

※ 상세표시

제 품 명	제 품 명
1	기온센서
2	기압센서
3	기습센서
4	기습센서
5	기습센서
6	기습센서
7	기습센서
8	기습센서
9	기습센서
10	기습센서
11	기습센서
12	기습센서
13	기습센서
14	기습센서
15	기습센서
16	기습센서
17	기습센서
18	기습센서
19	기습센서
20	기습센서
21	기습센서
22	기습센서
23	기습센서
24	기습센서
25	기습센서
26	기습센서
27	기습센서
28	기습센서
29	기습센서
30	기습센서
31	기습센서

※ 주요기능명

제 품 명	제 품 명
1	기온센서
2	기압센서
3	기습센서
4	기습센서
5	기습센서
6	기습센서
7	기습센서
8	기습센서
9	기습센서
10	기습센서
11	기습센서
12	기습센서
13	기습센서
14	기습센서
15	기습센서
16	기습센서
17	기습센서
18	기습센서
19	기습센서
20	기습센서
21	기습센서
22	기습센서
23	기습센서
24	기습센서
25	기습센서
26	기습센서
27	기습센서
28	기습센서
29	기습센서
30	기습센서
31	기습센서

그림 3-1. 축산 스마트팜 RS485 통신 프로토콜

- 본 연구에서는 악취 발생량 등 현장 상태 정보 수집의 센서 및 영상 자료에 대하여 전송 프로토콜을 RS485통신에 의해 전송하고 있으나, 향후 해당 기술을 보급하기 위해서는 RS485통신에 따른 데이터 량과 거리에 대한 검토를 통한 현장 적용 실용화 검토의 보완이 필요함

4. 가축분뇨 모니터링 기자재 시험방법 개발




- 앞에서 가축분뇨 저장고 내부를 모니터링을 위한 측정지표로 수온, EC, ORP, pH, 수위 등이 통합된 센서를 주로 사용되고 있는 것으로 나타났음
- EC, pH, 수온, ORP 센서의 경우, 표준용액에 온도보정을 통하여 센서의 성능을 측정하므로 첨부 문서와 같이 시험방법을 작성하였으며, 이에 대한 시험방법 등은 차년도 전문가협의회 등을 거쳐 정식으로 재단내규 등으로 등재할 예정임
- EC, pH, 수온 등의 센서는 아래와 같이 시험하였음

연번	업체명	모델명	측정대상	측정오차	측정불확도
1	디○○○○팜	SCF-01A	EC, pH	1.4 dS/m, 0.2 pH	0.5 dS/m, 0.5 pH
2	비○○○스	SFS111	EC	0.3 dS/m	0.5 dS/m
3	스○○○○○스	SPB_PH001	EC, pH, 수온	0.1 dS/m, 0.3 pH, 0.2 °C	0.5 dS/m, 0.5 pH, 0.5 °C
4	미○○서	WT1000B	EC, 수온	1.5 dS/m, 1.2 °C	0.5 dS/m, 0.9 °C
5	에○○○○○즈	SB-M139	EC, pH	0.4 dS/m, 0.5 pH	0.5 dS/m, 0.5 pH
6	성○○○○크	SDSE-03-07	EC, pH	0.1 dS/m, 0.1 pH	0.5 dS/m, 0.5 pH

- 시험결과, 업체에서 사용 중인 센서의 오차범위보다 측정오차가 낮게 나타나 이에 대한 시험방법으로 충분히 활용가능하다고 판단하였음
- 수위센서의 경우, 가축분뇨 저장고의 높이가 통상 6 m 인 것으로 고려 하였을 때 최소 3 m 이상의 수조에서 시험을 하여야 한다고 판단되어 이에 대한 시험장비는 아래와 같이 제작하여 동문 ENT 등에서 개발된 시험용 센서를 대상으로 시험 후 결과를 바탕으로 전문가 협의회 등을 추진할 예정임



표 3-14 수질측정기 구조분석 (M1000)

순번	구분	사진	특징
1	전체사진		<ol style="list-style-type: none"> <li>1. 휴대용 DAQ, 배터리 (현장측정용도)</li> <li>2. 접촉부의 이물질을 막기 위한 캡등</li> <li>3. 센서를 동일 위치에 위치하기 위한 거치대 등</li> </ol>
2	센서부		<ol style="list-style-type: none"> <li>1. 센서부에 슬러지 등이 들어가지 않도록 캡부착</li> </ol>
3	DAQ		<ol style="list-style-type: none"> <li>1. 휴대용 DAQ로 여러 가지 데이터 등을 저장</li> </ol>

## [첨부 1] 센서 등 시험방법

### 1. 수온 센서

#### 1-1. 시험조건

- (1) 기준기와 검정센서는 사용하기 전 검출부를 증류수로 세척하고 깨끗한 거름종이 또는 탈지면 등으로 닦아내 건조시킨다.
- (2) 측정은 설정온도에  $\pm 0.5\text{ }^{\circ}\text{C}$  이내, 상대습도  $\pm 3\text{ \% R.H}$  이내 항온항습조 내에서 안정화된 후 시험을 실시한다.
- (3) 이 시험조건에서 규정하지 않은 사항은 「KS C 1603(측온 저항체), KS A 0511(온도 측정방법 통칙), 한국계량측정협회(비금속 열전대의 표준교정절차)」 기술표준에 기술된 사항을 준용한다.

#### 1-2. 시험항목

- (1) 기준기 및 검정센서의 온도
- (2) 측정정확도

#### 1-3. 측정방법

- (1) 기준기와 검정센서를 그림1과 같이 항온조에 설치한다. 이때 기준기와 검정센서를 항온항습조 내의 동일한 위치에 5 cm 이내의 간격으로 설치한다.
- (2) 액체 항온조의 온도가 8개 지점의 측정점( $-10^{\circ}\text{C}$ ,  $-5^{\circ}\text{C}$ ,  $0^{\circ}\text{C}$ ,  $5^{\circ}\text{C}$ ,  $10^{\circ}\text{C}$ ,  $15^{\circ}\text{C}$ ,  $20^{\circ}\text{C}$ ,  $25^{\circ}\text{C}$ )에 대해서 검정하고자 하는 온도에 도달한 후 온도 안정화시간(최소 3분상)이후에 기준기와 검정센서의 온도값을 3분동안 수집한다.

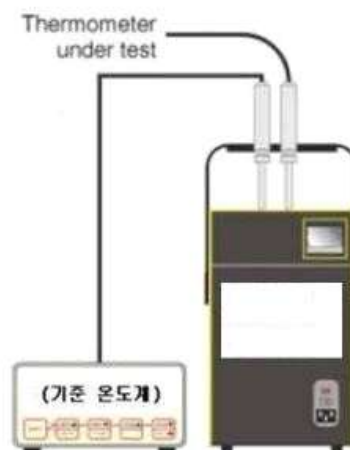


그림 12. 수온센서 검정 액체항온조

- (3) 각 측정점에서 센서 값과 기준기 값을 비교하여 측정정확도를 산출한다.

#### 1-4. 시험장비 및 계측기

기준 저항식온도계, 전압측정기(-100 mV ~ 100 mV), 고안정도 액체 항온조(-40 ~ 130 °C)

### 2. EC센서

#### 2-1. 시험조건

- (1) 센서를 사용하기 전 검출부를 증류수로 3회 이상 세척하고 깨끗한 거름종이 또는 탈지면으로 닦아 사용한다.
- (2) 표준용액은 염화칼륨(KCL)으로 각각 84  $\mu\text{s}/\text{cm}$ , 1413  $\mu\text{s}/\text{cm}$ , 12880  $\mu\text{s}/\text{cm}$  EC값을 갖는 용액으로 한다.
- (3) 표준용액의 온도는 10~40°C로 유지한다.
- (4) 시험은 낮은 농도부터 높은 농도 순으로 진행한다.
- (5) 전기전도도의 온도 의존성을 보정하고, 비교를 위해 25°C의 상온에서의 값으로 보정하는 것이 필요하므로 아래와 같이 25°C에서 측정 전기전도도를 보정해준다. 자체 보정 기능이 있을 경우 측정값을 그대로 사용한다.

$$C_{25} = \frac{C_T}{|(T-25)| \times 0.0249 + 1}$$

여기서,

$C_{25}$  : 25°C 보정 값( $\mu\text{s}/\text{cm}$ )     $C_T$  : 임의온도에서의 EC 측정값( $\mu\text{s}/\text{cm}$ )

$T$  : 측정 시 시료 온도

#### 2-2. 측정항목

- (1) 표준용액 온도
- (2) 측정정확도
- (3) 연속측정

#### 2-3. 시험방법

##### (1) 표준용액 온도

시험조건에 따라 설치된 EC센서와 같은 위치에 온도센서를 설치하여 매 1초 간격으로 수집하며 그 평균값으로 한다.

##### (2) 측정정확도

- (가) 각 표준용액 100 mL를 비커에 채운 후 검정센서를 담가 센서 값이 안정화 된 후 1분간 측정하고 세척 후 다시 같은 표준용액에 담가 측정한다.
- (나) 위 과정을 15회 반복한 후 각 반복에서 표준용액 농도와 센서 값을 비교하여 측정 정확도를 산출한다.

##### (3) 연속측정

농도가 가장 높은 표준용액에 센서를 담가 24시간 방치 후 세척하여 재현성 측정정

확도 방법에 따라 시험을 진행한다.

#### 2-4. 시험장비 및 계측기

온습도계, 데이터수집장치(전압, 전류계), 표준용액 등

### 3. pH센서

#### 3-1. 시험조건

- (1) 센서를 사용하기 전 검출부를 증류수로 3회 이상 세척하고 깨끗한 거름종이 또는 탈지면으로 닦아 사용한다.
- (2) 표준용액은 각각 4 pH, 7 pH, 10 pH를 갖는 용액으로 한다.
- (3) pH 표준용액의 온도는 10~40℃로 유지한다.
- (4) 시험은 낮은 pH부터 시작한다.
- (5) 온도에 따라 pH의 변화가 생기므로 정확한 측정을 위해서는 보정이 필요하다. KS M 0011에 따라 사용되는 표준용액은 [별첨 2]과 같이 보간법을 이용하여 하거나 그 외에 표준용액은 표준용액 제조사가 제시한 온도표에 따라 보정한다. 자체 보정기능이 있을 경우 측정값을 그대로 사용한다.

#### 3-2. 시험항목

- (1) 표준용액 온도
- (2) 측정정확도
- (3) 연속측정

#### 3-3. 시험방법

- (1) 표준용액 온도  
시험조건에 따라 설치된 pH센서와 같은 위치에 온도센서를 설치하여 매 1초 간격으로 수집하며 그 평균값으로 한다.
- (2) 측정정확도  
(가) 각 표준용액 100 mL를 비커에 채운 후 검정센서를 담가 센서 값이 안정화 된 후 1분간 측정하고 세척 후 다시 같은 표준용액에 담가 측정한다.  
(나) 위 과정을 15회 반복한 후 각 반복에서 표준용액 농도와 센서 값을 비교하여 측정정확도를 산출한다.
- (3) 연속측정방법  
농도가 가장 높은 표준용액에 센서를 담가 24시간 방치 후 세척하여 재현성 측정정확도 방법에 따라 시험을 진행한다.

#### 3-4. 시험장비 및 계측기



온습도계, 데이터수집장치(전압, 전류계), 표준용액 등

[별첨 1] pH 온도보상 방법

pH 표준용액의 온도별 pH 값은 아래 표와 같으며 기재되지 않은 온도에서의 pH값은 앞 뒤의 값을 보간하여 사용한다.

$$y = \frac{y_2 - y_1}{x_2 - x_1} \times (x - x_1) + y_1$$

$y$  = 측정온도에서의 pH 값,                       $y_1$  = 측정온도 아래 pH 값

$y_2$  = 측정온도 위 pH 값                               $x$  = 측정온도

$x_1$  = 측정온도 아래 온도 값                       $x_2$  = 측정온도 위 온도 값

온도 °C	pH 값										
	옥살산염		프탈산염		중성 인산염		인산염		붕산염		탄산염
	제1종	제2종	제1종	제2종	제1종	제2종	제1종	제2종	제1종	제2종	제2종
0	1.666	1.67	4.003	4.00	6.984	6.98	7.534	7.53	9.464	9.46	10.32
5	1.668	1.67	3.999	4.00	6.951	6.95	7.500	7.50	9.395	9.40	10.24
10	1.670	1.67	3.998	4.00	6.923	6.92	7.472	7.47	9.332	9.33	10.18
15	1.672	1.67	3.999	4.00	6.900	6.90	7.448	7.45	9.276	9.28	10.12
20	1.675	1.68	4.002	4.00	6.881	6.88	7.429	7.43	9.225	9.22	10.06
25	1.679	1.68	4.008	4.01	6.865	6.86	7.413	7.41	9.180	9.18	10.01
30	1.683	1.68	4.015	4.02	6.853	6.85	7.400	7.40	9.139	9.14	9.97
35	1.688	1.68	4.024	4.02	6.844	6.84	7.389	7.39	9.102	9.10	9.92
38	1.691	1.69	4.030	4.03	6.840	6.84	7.384	7.38	9.081	9.08	-
40	1.694	1.69	4.035	4.04	6.838	6.84	7.380	7.38	9.068	9.07	9.89

### 3. 연구개발과제의 수행 결과 및 목표 달성 정도

#### 1) 연구수행 결과

##### (1) 정성적 연구개발성과

구분	세부연구개발 목표	가중치	평가의 착안점 및 기준	달성도
최종 평가	○ 박테리아기반가축분뇨신속수거처리시스템개발	25 %	○ 시계열기반 분뇨 발생량 예측모델 개발 ○ 축산분뇨 발생량, 신속 수거/처리 현황 플랫폼 구축 ○ 축산농가용 모니터링 App 개발 ○ 분뇨수거 회사 및 처리회사용 App 개발	100%
	○ ICT 기반가축분뇨 발생량 모니터링 제품개발	25 %	○ 가축분뇨 저장조 내 온도, 수위, pH, ORP 통합센서시스템 ○ 통합센서시스템으로부터 저장된 데이터를 시간별 저장 ○ 현장용 데이터 서버 전송 단말장치	100%
	○ 축사별가축분뇨차와저장현황조사및발생량모 니터링인자기발	25 %	○ 센서 이용 양돈분뇨 저장조 내 분뇨의 발생량과 물성 변화, 악 취 발생경향 파악하여 적정 분뇨 수거 시기 제시	100%
	○ 가축분뇨 발생량 모니터링시스템을 평가하기 위한 검정방법 및 검정시스템 구축	25 %	○ 검정방법(안) 개발 ○ 검정시스템 구축	100%

##### (2) 정량적 연구개발성과(해당 시 작성하며, 연구개발과제의 특성에 따라 수정이 가능합니다)

성과 목표	연구기반지표																			
	지식 재산권			기술 실시 (이전)		사업화					기술 인증	학술성과				교육 지도	인력 양성	정책 활용-홍 보		기 타 (타 연 구 활 용 등)
	특 허 출 원	특 허 등 록	품 종 등 록	건 수	기술 료	제 품 화	매 출 액	수 출 액	고 용 창 출	투 자 유 치		논문		학 술 발 표	정 책 활 용			홍 보 전 시		
												SC I	비 SC I						논 문 평 균 IF	
단위	건	건	건	건	백 만 원	건	백 만 원	백 만 원	명	백 만 원	건	건	건	건	명	건	건			
1차년도	1								1			1		3						
2차년도	2	1		1		1			1			2		4		3	1	1	1	
소 계	3	1		1		1			2			2	1	7		3	1	1	1	
달성 건수	3			1		1			2			2	2	433		4	1	1	1	

(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	국내 말, 사슴, 염소 분뇨 유래 양분 발생량 및 농경지 부하량 산정	동물자원 연구	김기원	32(4)	대한민국	한국축산학회	비SCIE	2021.12	2287-3317	100
2	Comparing Performances of CNN, BP, and SVM Algorithms for Differentiating Sweet Pepper Parts for Harvest Automation	applied sciences	이봉기	11(20)	스위스	MDPI	SCIE	2021.10.14	2076-3417	100
3	Development of a Real-Time Controlled Bio-Liquor Circulation System for Swine Farms: A Lab-Scale Study	ANIMALS	김승수	11(2)	스위스	MDPI	SCIE	2021.01.26	2076-2615	100
4	거대역새 현장 전처리 통합작업기 개발	농업생명 과학연구	권구범	54(4)	대한민국	경상대학교 농업생명과학연구원	비SCIE	2020.08.10	1598-5504	100
5	Estimation of Nutrient Loading to Farmland and Production from Manure of Horse, Deer and Goat in Korea	동물자원 연구	심수민	4	대한민국	강원대학교 동물자원공동연구소	비SCIE	2021-12	1225-2964	100

- 각 논문들은 연구과제를 수행하던 중 확보된 기술과 실험 결과를 바탕으로 작성되었으며, 각 논문에는 농림식품기술기획평가원의 지원에 따른 결과임을 명시하는 사사를 분명히 표기하였습니다.

□ 국내 및 국제 학술회의 발표

번호	회의 명칭	발표자	발표 일시	장소	국명
1	국내 폐사가축 처리현황 조사 및 적정 관리방안 개발	공다은	2021.11.09	온라인	대한민국
2	KS X 3267을 기반으로 디지털농업기자재 호환성(RS485) 시험을 위한 검정시스템 및 검정방법 개발	한철우	2021.10.27	제주 소노벨	대한민국
3	가축분뇨 신속 수거 처리 시스템을 위한 웹서비스 구현	이봉기	2021.10.27	제주 소노벨	대한민국
4	ICT 기반 가축분뇨 신속 수거 처리 시스템을 위한 통신 프로토콜	이봉기	2021.04.30	국립농업과학원 농업공학부	대한민국
5	현장 맞춤형 경축순환농업 확산을 위한 기초정보 확보 연구	공다은	2021.07.08	충북대학교 농업생명과학대학 첨단바이오연구센터	대한민국
6	경축순환 활성화를 위한 가축분뇨 공동자원화시설 운영의 저해요인 및 개선사항 조사	심수민	2020.08.27	온라인	대한민국
7	가축분뇨 발생량 모니터링 시스템 개요	이봉기	2020.06.15	온라인	대한민국
8	가축분뇨 및 녹조 슬러지의 탄화 고품 연료화	이봉기	2020.06.15	온라인	대한민국

□ 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

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, 신제품, 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	가축분뇨 저장조 스마트 관리 시스템	대한민국	전북대학 교산학협 력단	2020.12 .28	10-2020 -018491 5				100		
2	노지용 스마트 제초 로봇	대한민국	전북대학 교산학협 력단	2021.05 .21	10-2021 -006531 7				100		
3	노지용 친환경 제초기	대한민국	전북대학 교산학협 력단	2021.05 .21	10-2021 -006530 36				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
1, 2, 3	√		√	√						

저작권(소프트웨어, 서적 등)

번호	저작권명	창작일	저작자명	등록일	등록 번호	저작권자명	기여율

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) 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	표준화단계구분 <sup>1)</sup>	표준명	표준기구명 <sup>2)</sup>	표준분과명	의장단 활동여부	표준특허 추진여부	표준개발 방식 <sup>3)</sup>	제안자	표준화 번호	제안일자

\* 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.

[경제적 성과]

□ 시제품 제작

번호	시제품명	출시/제작일	제작 업체명	설치 장소	이용 분야	사업화 소요 기간	인증기관 (해당 시)	인증일 (해당 시)
1	측분 측정용 센서 자동세척장치	22.02.28	동문이엔티(주)	전북 익산 농가	측분 측정용 센서 자동세척	2년		

□ 기술 실시(이전)

번호	기술 이전 유형	기술 실시 계약명	기술 실시 대상 기관	기술 실시 발생일	기술료 (해당 연도 발생액)	누적 징수 현황
1	통상실시권	가축분뇨 저장조 관리 시스템	전북대학교	2021.08.12	5,000,000	

\* 내부 자금, 신용 대출, 담보 대출, 투자 유치, 기타 등

□ 사업화 투자실적

번호	추가 연구개발 투자	설비 투자	기타 투자	합계	투자 자금 성격*

□ 사업화 현황

번호	사업화 방식 <sup>1)</sup>	사업화 형태 <sup>2)</sup>	지역 <sup>3)</sup>	사업화명	내용	업체명	매출액		매출 발생 연도	기술 수명
							국내 (천원)	국외 (달러)		

\* 1) 기술이전 또는 자기실시

\* 2) 신제품 개발, 기존 제품 개선, 신공정 개발, 기존 공정 개선 등

\* 3) 국내 또는 국외

□ 매출 실적(누적)

사업화명	발생 연도	매출액		합계	산정 방법
		국내(천원)	국외(달러)		
합계					

□ 사업화 계획 및 무역 수치 개선 효과

성과		센서를 활용하여 축산분뇨 자동 모니터링			
사업화 계획	사업화 소요기간(년)	5년			
	소요예산(천원)	100,000			
	예상 매출규모(천원)	현재까지	3년 후	5년 후	
		-	100,000	500,000	
	시장 점유율	단위(%)	현재까지	3년 후	5년 후
			국내	-	100
국외			-	100	100
향후 관련기술, 제품을 응용한 타 모델, 제품 개발계획		현 시제품의 디자인, 성능 등을 단계적으로 개선하여 보다 compact하고 설치 용이하고 내구성이 있는 제품 개발			
무역 수치 개선 효과(천원)	수입대체(내수)	현재	3년 후	5년 후	
		-	-	-	
	수출	-	50,000	200,000	

□ 고용 창출

순번	사업화명	사업화 업체	고용창출 인원(명)		합계
			2020년	2021년	
1	신규고용창출인력	동문이엔티(주)	1	1	2
합계			1	1	2

□ 고용 효과

구분			고용 효과(명)
고용 효과	개발 전	연구인력	5
		생산인력	-
	개발 후	연구인력	7
		생산인력	2

□ 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)	명칭	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	농업공학 분야 (축산용 디지털 농기자재) 관련 정책건의	농림축산식품부	2022.01.12	농림축산식품부에 '축산ICT기자재 보급 확산을 위한 지원대상자 평가방안'을 정책건의

□ 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
1	인력양성	2020		1			1						1
2	인력양성 및 취업	2021	박사	석사	학사	기타	남	여	수도권	충청권	영남권	호남권	기타
					3		3					3	

□ 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
1	지방일간지 홍보	농업인신문	축산농가 분뇨저장조 모니터링 시스템 개발	2021.11.05

□ 포상 및 수상 실적

번호	종류	포상명	포상 내용	포상 대상	포상일	포상 기관

**[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)

\* 「과학기술기본법 시행령」 제42조제4항제2호에 따른 연구시설·장비 종합정보시스템을 의미합니다.

**[그 밖의 성과](해당 시 작성합니다)**

- 우수논문발표상 수상 (한국농업기계학회 2020년 추계 공동학술대회)

**(4) 계획하지 않은 성과 및 관련 분야 기여사항(해당 시 작성합니다)**

## 2) 목표 달성 수준

추진 목표	달성 내용	달성도(%)
○ 빅데이터 기반 가축 분뇨 신속 수거 처리 시스템 개발	○ 시계열 기반 분뇨 발생량 예측모델 개발 ○ 축산 분뇨 발생량, 신속 수거/처리 현황 플랫폼 구축 ○ 축산 농가용 모니터링 App 개발 ○ 분뇨 수거 회사 및 처리회사용 App 개발	100%
○ ICT 기반 가축 분뇨 발생량 모니터링 제품 개발	○ 가축 분뇨 저장조 내 온도, 수위, pH, ORP 통합 센서 시스템 ○ 통합 센서 시스템으로부터 저장된 데이터를 시간별 저장 ○ 현장용 데이터 서버 전송 단말장치	100%
○ 축사별 가축 분뇨 처리와 저장 현황 조사 및 발생량 모니터링 인자 개발	○ 센서 이용 양돈 분뇨 저장조 내 분뇨의 발생량과 물성 변화, 악취 발생 경향 파악하여 적정 분뇨 수거 시기 제시	100%
○ 가축 분뇨 발생량 모니터링 시스템을 평가하기 위한 검증 방법 및 검증 시스템 구축	○ 검증 방법(안) 개발 ○ 검증 시스템 구축	100%

4. 목표 미달 시 원인분석(해당 시 작성합니다)

1) 목표 미달 원인(사유) 자체분석 내용

---

2) 자체 보완활동

---

3) 연구개발 과정의 성실성

---

## 5. 연구개발성과의 관련 분야에 대한 기여 정도

- 시제품 장치는 고도화 하여 국외에 수출
- 가축분뇨 발생·수거·처리·관리·예측 전주기 플랫폼을 통한 체계적인 모니터링
- ICT/IoT 장치를 이용하여 가축분뇨의 발생량을 실시간 추적
- 전국의 분뇨처리 사항을 실시간 분석

구분	내용	비고
기술적측면	<ul style="list-style-type: none"> <li>● 가축분뇨 발생·수거·처리·관리·예측 전주기 플랫폼 개발</li> <li>● 가축분뇨 발생 예측 모델링 알고리즘화</li> <li>● 영상취득 및 분석을 통한 분뇨발생량 위변조 방지</li> </ul>	
산업적측면	<ul style="list-style-type: none"> <li>● 전주기 플랫폼 활성화로 신산업 창출 : 177조원 규모</li> <li>● 가축분뇨 관련 IT회사 신규시장 창출 기대</li> <li>● 가축분뇨 관련 제조,설비,설비 등의 신규시장 창출 기대</li> </ul>	177조원 규모
사회적측면	<ul style="list-style-type: none"> <li>● 가축분뇨 수거 지체로 인한 손실비용 절감</li> <li>● 가축분뇨의 신속한 수거관리로 악취관련 민원 최소화</li> <li>● 체계적인 가축분뇨 관리로 비점오염원의 20%를 개선</li> </ul>	
국가행정측면	<ul style="list-style-type: none"> <li>● 환경오염 최소화 및 손실비용 절감으로 국가경쟁력 강화</li> <li>● 전국가축분뇨 발생·수거·처리·관리·예측 전주기 모니터링</li> <li>● 가축분뇨 수집·운반과정의 분뇨 이동관련 자료 신뢰성 확보</li> <li>● 가축분뇨로 인한 환경노출과 환경영향 예측</li> <li>● 가축분뇨 관리에 따른 정책입안 활용</li> </ul>	

## 6. 연구개발성과의 관리 및 활용 계획

### 가. 과제 종료 이후 현장적용 방안

대상	개소	적용방안	비고
익산 소재 농장	2	<ul style="list-style-type: none"> <li>● ICT기반측정장치 시제품 2 Set 설치</li> <li>● 통합 통신장비 시제품 2 Set 설치</li> <li>● 현장 검증</li> </ul>	
익산시	1	<ul style="list-style-type: none"> <li>● 분뇨 발생,수거,처리,현황 플랫폼</li> </ul>	

구분	항목	내용	비고
대상고객	농장	전국 돈사 : 5,250	
인프라 구축	테스트베드	<ul style="list-style-type: none"> <li>● 농장 2개소 테스트베드 구축</li> </ul>	
	시제품제작	<ul style="list-style-type: none"> <li>● ICT/IoT 기반 측정장치 : 2 Set</li> <li>● 통합 통신장비 : 2 Set</li> </ul>	
	S/W개발	<ul style="list-style-type: none"> <li>● Server Side SW</li> <li>● 농장용 App</li> <li>● 분뇨수거 회사 및 처리회사 App</li> <li>● 발생,수거,처리 현황 플랫폼</li> <li>● 현장용 단말장치 시스템</li> <li>● Open API</li> </ul>	
	주요 서비스	<ul style="list-style-type: none"> <li>● ICT/IoT 기반 가축분뇨 발생량 모니터링</li> <li>● 가축분뇨 신속 수거·처리 서비스</li> <li>● ICT 기반 통합관리 서비스</li> </ul>	
연구개발	빅데이터	<ul style="list-style-type: none"> <li>● 전국 분뇨 발생량 분석 및 예측량 산출</li> </ul>	
	영상처리	<ul style="list-style-type: none"> <li>● 분뇨발생량 위변조 방지시스템적용</li> </ul>	

### 나. 수요처 및 국내시장 규모 : 177조원

(단위:억원)

구분	산출근거	금액
돈사	경영체(5,250) × 평균 보유 저장조(3) × 800만원(설치비포함)	126,000
우사	경영체(102,917) × 평균 보유 저장조(3) × 800만원(설치비포함)	1,646,672
합계		1,772,672

※ 출처: 농업경영체 등록정보 조회 서비스

**< 연구개발성과 활용계획표 >**

구분(정량 및 정성적 성과 항목)		연구개발 종료 후 5년 이내	
국외논문	SCIE	1	
	비SCIE	-	
	계	1	
국내논문	SCIE	-	
	비SCIE	-	
	계	0	
특허출원	국내	-	
	국외	-	
	계	0	
특허등록	국내	2	
	국외	-	
	계	2	
인력양성	학사	-	
	석사	1	
	박사	-	
	계	1	
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
2.	1) 2)

[뒷면지]

주 의

1. 이 보고서는 농림축산식품부에서 시행한 첨단생산기술개발 연구개발사업 ICT 기반 가축분뇨 발생량 모니터링 및 신속 수거·처리시스템 개발연구개발과제 최종보고서이다.
2. 이 연구개발내용을 대외적으로 발표할 때에는 반드시 농림축산식품부(농림식품기술기획평가원 전문기관)에서 시행한 첨단생산기술개발 연구개발사업의 결과임을 밝혀야 한다.
3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 된다.