

발간등록번호

11-1543000-001205-01

스마트폰 원격 제어 기술을 이용한 하우스 및
농업시설물의 실시간 전원제어 실용화 구현

(Commercialization of real-time control of power
supply system using remote control technology
at green house and agricultural facilities)

주식회사 헤임달

농림축산식품부

제 출 문

농림축산식품부 장관 귀하

이 보고서를 “스마트폰 원격 제어 기술을 이용한 하우스 및 농업 시설물의 실시간 전원 제어 실용화 구현”에 관한 연구 과제의 보고서로 제출합니다.

2016년 03월 16일

주관연구기관명 : 주식회사 해임달

주관연구책임자 : 남 승 균

연 구 원 : 남 동 현

연 구 원 : 심 지 현

요 약 문

I. 제 목

스마트폰 원격 제어 기술을 이용한 하우스 및 농업 시설물의 실시간 전원제어 실용화 구현

II. 연구성과 목표 대비 실적

당초목표	당초연구목표 대비 연구결과
① 원격 전원 제어 시스템을 이용한 하우스 및 농업시설물의 자동개폐장치 전원 관리	TCP/IP 통신을 이용하여 원격지에서 전원을 ON/OFF할 수 있는 관제 소프트웨어를 개발
② 전원 관련 상태 정보 및 에러 발생 시 상황 모니터링	포트별 전원의 상태, 포트별 소비전류 양의 상한, 하한 값을 설정하고 설정된 값을 벗어나면 알람을 발생시키도록 하는 알람기능 구현
③ 자체 시스템 보호 기능(안전장치)	포트별 과전류 방지용 퓨즈를 삽입. 펌웨어단에서도 과전류 발생 시 포트를 일시적으로 차단하는 기능 구현하였음.
④ 단순화 효율적인 UI를 가진 App 개발	농민이 출타중이거나 이동 중일 경우 간단하게 원격전원장치의 상태를 확인하고 전원을 제어할 수 있는 스마트폰용 앱 개발.

III. 연구개발의 목적 및 필요성

1. 연구개발의 목적

- 가. 원격 전원 제어 시스템을 이용한 하우스 및 농업시설물의 자동개폐장치 전원 관리
- 나. 전원 관련 상태 정보 및 에러 발생 시 상황 모니터링
- 다. 자체 시스템 보호 기능 (안전장치)
- 라. 기존사례 구체적 문제점 분석 및 제품 원가, 설치비 일체의 절감을 통한 실용화 방안 모색
- 마. 단순하고 효율적인 UI를 가진 App 개발

2. 연구개발의 필요성

- 가. 국내 외 현황 및 문제점
 - (1) 현재, 원격 전원 제어 장치 및 자동개폐장치는 다양한 농작물의 하우스 재배에서 사용되고 있음. 그 중 대표적인 예로 참외 농가의 현황 및 문제점 파악을 기준으로 삼아 개발 진행하고자 함.
 - (2) 경북 성주시의 참외 재배 면적은 전국 재배 면적의 70% 이상으로 성주 군내 참외 재배 농가는 2013년 기준으로 4,433가구에 이르며 재배 면적은 3,879ha가 넘음.
 - (3) 2010년, 재배 농가는 4,795가구이며 재배 면적은 4,011ha 인 것에 비해 2013년 362가구, 재배 면적은 132ha나 줄어들었음

- (4) 참외 농가 감소 현상의 이유로는 농촌 고령화 및 가격 하락, 농산물 수입 등이 주를 이룸
- (5) 이에, 노동력 절감을 위한 시설 현대화 및 참외 농가 부흥을 위한 신규 후계 농업 경영인 육성, 우수 후계농 추가 지원사업 등의 지원제도가 마련되고 있는 실정임.
- (6) 그 일환으로 2013년 100억원의 사업비를 들여 참외하우스 5천동에 파이프를 교체했으며, 77억원을 들여 7천동에 보온덮개 자동 개폐기를 설치함.
- (7) 경상북도, 성주군, KT는 보온덮개를 여닫기 위해 직접 현장까지 방문하지 않아도 되도록 원격제어 시스템을 시험 보급 설치하여 실시했으나 실용화에는 어려움이 있었음.
- (8) 실용화가 어려웠던 KT 사례의 경우, 구체적 문제점으로는 제품의 크기가 커서 공간을 많이 차지한다는 점과 3G 망을 이용한 원격제어가 되지 않는 점 그리고 단위 기계 가격이 높게 책정되어 보급이 어려운 점 등이 있음.

나. 연구개발이 필요성 전망

- (1) 참외농가 감소현상의 주된 원인은 농촌의 고령화로써 노동력 절감을 위한 시스템 도입 필요
- (2) 참외 보온 덮개의 자동개폐장치가 마련된 후, 원격 제어장치 개발 및 시범사업을 진행하였으나, 고비용 문제로 실용화 미흡
- (3) 이에 원격 제어 시스템의 저비용 보급을 위한 기술적 보완을 통해 실용화 구현
- (4) 저비용 설치 가능 시, 노동력 재고와 재배의 효율성으로 참외 농가의 확대가 이루어 질 수 있고, 고품질 상품 수확이 가능해짐으로써 농가 수익 증대 예상
- (5) 대중화 되어 있는 스마트폰을 사용한 원격제어시스템을 통한 하우스 및 농업시설물의 실시간 전원 제어기술은 그 필요성이 커 사업성이 기대됨.

IV. 연구개발 내용 및 범위

가. 원격 전원 제어 장치 시스템 운영

- (1) 기본적으로는 수동 운영 모드를 이용해 전원을 제어하며 타이머 설정으로 자동 운영 모드도 지원할 수 있도록 함
- (2) 자동 운영 모드의 이용 시, 동작 구간은 24시간 내로 자유설정 가능하도록 설계
- (3) 농민들이 손쉽게 조작할 수 있도록 간단하고 효율적인 기능으로 제작

나. 전원 포트의 소비 전원 상태 모니터링

- (1) AC 전력 측정용 모듈을 탑재하여 전원 포트의 소비 전원 상태를 즉각적으로 확인 할 수 있도록 함
- (2) 예러상황 발생 시 시스템 상태 정보에 대하여 피드백 받을 수 있는 시스템 마련

다. 자체 시스템 보호 기능 (안전장치)

- (1) 본체 시스템의 문제 발생 시 자동 reset 후 시스템을 이전 상태로 복구
- (2) 파워 릴레이 보호회로 내장으로 시스템에 대한 높은 안정성 제공
- (3) 각 전원 출력 포트에 개별 퓨즈를 내장하여 전원 단락과 같은 긴급 상황 시, 본체에 연결된 타 장비가 안전하도록 설계
- (4) 본체 보호를 위한 전원 차단용 브레이크 내장 설계

라. 시제품 제작

- (1) 제어 로직 : 펌웨어 단계에서 물리적 제어, 릴레이 형식으로 제어해서 on/off 구현할 계획
- (2) 통신망 : KT, SK에서 사용 중인 3G 모듈을 탑재해서 구현할 계획, TCP/IP 소켓 통신 모듈도 예비로 장착
- (3) 연계방법 : 최종 시제품의 일반적 성능 검증 후, 통신사와 협의를 통해 시험 사업으로 추진 예정

마. 설치비 절감을 통한 실용화 방안 모색

- (1) 기존 KT 시범 사업 설치 농가의 설치 상태 조사 및 구체적 문제점 분석
- (2) 목표 판매가를 맞출 수 있도록 장비 및 부품의 최적화 설계

V. 연구개발결과

연구목표	실 적	목표달성 여부
가. 원격 전원 제어 시스템을 이용한 하우스 및 농업시설물의 자동개폐장치 전원 관리	TCP/IP 통신을 이용하여 원격지에서 전원을 ON/OFF 할 수 있는 원격 전원관리장치와 원격기에서 이를 제어할 수 있는 관계 소프트웨어를 개발하였으며 다수의 원격전원관리장치를 이용하기 위하여 관계 소프트웨어를 1:N 방식을 가능하도록 구현하였음.	달 성
나. 전원 관련 상태 정보 및 에러 발생 시 상황 모니터링	관계 소프트웨어에는 알람기능을 구현하였음. 알람기능은 포트별 전원의 상태(ON/OFF), 포트별 소비전류의 양의 상한 하한 값을 설정하고 이 설정된 값을 벗어나면 알람을 발생시키도록 하였으며 발생한 알람은 관리자가 "Ack" 하여 알람을 확인한 후 해지하도록 하였으며 알람 이력이 시스템에 쌓이도록 하였음.	달 성
다. 자체 시스템 보호 기능 (안전장치)	과전류 차단하기 위하여 포트별로 퓨즈를 추가하였으며 펌웨어 단에서 과전류 발생 시 시스템을 자동으로 차단하도록 하는 기능을 추가하였음. 시스템의 현재 상황을 플래시 메모리에 기록하도록 하여 일시적으로 전원이 차단된 후 인가되어도 새로 설정이 필요 없이 원래 상태로 복구되는 기능을 추가하였음.	달 성
라. 기존사례 구체적 문제점 분석 및 제품 원가, 설치비 일체의 절감을 통한 실용화 방안 모색	기존 시스템에 대한 분석을 수행하여 문제점과 개선방안을 도출하였으며 제품 원가, 설치비 일체의 절감을 통한 목표 가격을 책정하였음.	달 성
마. 단순하고 효율적인 UI를 가진 App 개발	농민이 출타중이거나 이동 중일 경우 간단하게 원격전원제어장치의 상태를 확인하고 전원을 제어할 수 있는 스마트폰용 앱을 개발하였음.	달 성

Ⅵ. 연구성과 및 성과활용 계획

1. 센서와 연동한 SMART 제어기 추가연구 추진

가. 본 과제는 전류 센서만 추가된 경우로 기울기센서, 접점센서, 광센서 등과 연동하여 액추에이터가 실제로 동작하는 상황을 모니터링하며 실제동작을 피드백하여 더욱 정밀하고 능동적인 제어가 되기 위한 연구를 수행할 계획임.

2. 버섯 재배사 적용 가능성 타진

가. 재배환경에 매우 민감한 버섯 재배사에 본 과제에서 개발한 결과물을 적용하기 위하여 선행조사를 실시하고 버섯 재배사에 최적화된 시스템을 도출하도록 시스템 추가사항을 도출할 계획임.

나. 버섯 재배사에 적용 가능한 시스템을 개발하기 위하여 버섯 재배농가, 버섯 재배 전문가와 협업하여 최적의 시스템을 도출하고 이를 국가 R&D를 통하여 개발할 예정임.

3. 축산 농가 적용 가능성 타진

가. 양돈, 양계 농가에서는 가축들의 사육환경을 제어하기 위하여 환풍기, 온풍기, 사료 급이기 등의 전기기계장치를 운용하고 있으며 현재 축산 농가들은 이를 수작업으로 운용하고 있음.

나. 양돈, 양계 농가에서 운용되는 전기기계장치를 자동화하기 위하여 필요한 구성요소를 파악하고 이를 추가하여 축산 농가에 적용할 수 있는 시스템을 개발할 예정임.

SUMMARY

One of the main goals for this project is to develop a power control device which can turn on or off farm facilities remotely. The power control device uses TCP/IP communication technology and control software can manage N power control devices at the same time. Control software monitors currents and power status for each device continuously and it generates alarms if they go out of the upper and lower limits. These functions help manage powered facilities in a systematic and effective way.

Now, Korean farming areas are experiencing rapid aging population and a shortage of labor as a result. The system can help save labor and cultivate crops in a scientific way. It is considered to be a must-have item to control powered facilities in a vinyl house. Farmers can manage a vinyl house environment by opening or closing ventilation windows or turning on/off heaters with the help of the power control device. They can even monitor and control facilities by their smartphones anytime anywhere.

This project applied vinyl houses for oriental melon in Seouju, Gyeongbuk. The power control devices were installed and they control the power of ventilating facilities mainly. It turns out that a huge amount of labor can be saved and cultivating crops in more systematic and scientific ways. It is expected that farmers can gain more yield and it leads to more profit.

The results of this project are very encouraging and useful. Many researches and trials to apply this technology to other fields should be carried out without fail. Applicable fields might be animal farming areas, mushroom farms and so forth. This technology can be applied to all over the farming areas and can cause a reduction of costs and labor.

CONTENTS

- Chapter 1 Overview of R & D Project and Goals for Achievement
 - Section 1 Overview of R&D Project
 - Section 2 Achievement of R&D Project(Goals Versus Achievement)

- Chapter 2 International and Domestic Current Status of Technical Development
 - Section 1 International and Domestic Current Status of Technical Development

- Chapter 3 Detailed Contents and Results of Project
 - Section 1 Detailed Contents of Project
 - Section 2 Results of Project

- Chapter 4 Goal Achievement and Contribution to Other Related Fields
 - Section 1 Goal Achievement
 - Section 2 Contribution to Other Related Fields

- Chapter 5 Achievement and Plans for Utilize Development Results
 - Section 1 Plans for application in Industrial Fields
 - Section 2 Plans for Technology diffusion
 - Section 3 Plans for Additional R&D Project

- Chapter 6 International Science and Technology Information gathered from the research and development process

- Chapter 7 Research facilities · Equipments

- Chapter 8 Performance Result For Safety Supervision

- Chapter 9 References

목 차

제 1 장	연구개발과제의 개요 및 성과목표	9
제 1 절	연구개발 과제의 개요	9
제 2 절	연구성과(목표 대비 실적)	11
제 2 장	국내외 기술개발 현황	12
제 1 절	국내외 기술개발 현황	12
제 3 장	연구개발수행 내용 및 결과	14
제 1 절	연구개발수행 내용	14
제 2 절	개발결과	47
제 4 장	목표달성도 및 관련분야에의 기여도	70
제 1 절	연구개발목표의 달성도	70
제 2 절	관련분야의 기술발전예의 기여도	71
제 5 장	연구개발 성과 및 성과활용 계획	72
제 1 절	실용화 산업화 계획	72
제 2 절	교육·지도·홍보 등 기술 확산 계획 등	72
제 3 절	추가연구 / 타 연구에 활용 계획	73
제 6 장	연구개발과정에서 수집한 해외과학기술정보	74
제 7 장	연구시설·장비 현황	75
제 8 장	연구실 안전관리 이행실적	76
제 9 장	참고문헌	77

제 1 장 연구개발과제의 개요 및 성과목표

제 1 절 연구개발 과제의 개요

1. 연구개발의 목적

- 가. 원격 전원 제어 시스템을 이용한 하우스 및 농업시설물의 자동개폐장치 전원 관리
- 나. 전원 관련 상태 정보 및 에러 발생 시 상황 모니터링
- 다. 자체 시스템 보호 기능 (안전장치)
- 라. 기존사례 구체적 문제점 분석 및 제품 원가, 설치비 일체의 절감을 통한 실용화 방안 모색
- 마. 단순하고 효율적인 UI를 가진 App 개발

2. 연구개발의 필요성

가. 국내 외 현황 및 문제점

- (1) 현재, 원격 전원 제어 장치 및 자동개폐장치는 다양한 농작물의 하우스 재배에서 사용되고 있음. 그 중 대표적인 예로 참외 농가의 현황 및 문제점 파악을 기준으로 삼아 개발 진행하고자 함.
- (2) 경북 성주시의 참외 재배 면적은 전국 재배 면적의 70% 이상으로 성주 군내 참외 재배 농가는 2013년 기준으로 4,433가구에 이르며 재배 면적은 3,879ha가 넘음.
- (3) 2010년, 재배 농가는 4,795가구이며 재배 면적은 4,011ha 인 것에 비해 2013년 362가구, 재배 면적은 132ha나 줄어들었음
- (4) 참외 농가 감소 현상의 이유로는 농촌 고령화 및 가격 하락, 농산물 수입 등이 주를 이룸
- (5) 이에, 노동력 절감을 위한 시설 현대화 및 참외 농가 부흥을 위한 신규 후계 농업 경영인 육성, 우수 후계농 추가 지원사업 등의 지원제도가 마련되고 있는 실정임.
- (6) 그 일환으로 2013년 100억원의 사업비를 들여 참외하우스 5천동에 파이프를 교체했으며, 77억원을 들여 7천동에 보온덮개 자동 개폐기를 설치함.
- (7) 경상북도, 성주군, KT는 보온덮개를 여닫기 위해 직접 현장까지 방문하지 않아도 되도록 원격제어 시스템을 시험 보급 설치하여 실시했으나 실용화에는 어려움이 있었음.
- (8) 실용화가 어려웠던 KT 사례의 경우, 구체적 문제점으로는 제품의 크기가 커서 공간을 많이 차지한다는 점과 3G 망을 이용한 원격제어가 되지 않는 점 그리고 단위 기계 가격이 높게 책정되어 보급이 어려운 점 등이 있음.

나. 연구개발이 필요성 전망

- (1) 참외농가 감소현상의 주된 원인은 농촌의 고령화로써 노동력 절감을 위한 시스템 도입 필요

- (2) 참외 보온 덮개의 자동개폐장치가 마련된 후, 원격 제어장치 개발 및 시범사업을 진행하였으나, 고비용 문제로 실용화 미흡
- (3) 이에 원격 제어 시스템의 저비용 보급을 위한 기술적 보완을 통해 실용화 구현
- (4) 저비용 설치 가능 시, 노동력 재고와 재배의 효율성으로 참외 농가의 확대가 이루어 질 수 있고, 고품질 상품 수확이 가능해짐으로써 농가 수익 증대 예상
- (5) 대중화 되어 있는 스마트폰을 사용한 원격제어시스템을 통한 하우스 및 농업시설물의 실시간 전원 제어기술은 그 필요성이 커 사업성이 기대됨.

3. 연구개발의 내용

가. 원격 전원 제어 장치 시스템 운영

- (1) 기본적으로는 수동 운영 모드를 이용해 전원을 제어하며 타이머 설정으로 자동 운영 모드도 지원할 수 있도록 함
- (2) 자동 운영 모드의 이용 시, 동작 구간은 24시간 내로 자유설정 가능하도록 설계
- (3) 농민들이 손쉽게 조작할 수 있도록 간단하고 효율적인 기능으로 제작

나. 전원 포트의 소비 전원 상태 모니터링

- (1) AC 전력 측정용 모듈을 탑재하여 전원 포트의 소비 전원을 즉각적으로 확인 할 수 있도록 함
- (2) 에러상황 발생 시 시스템 상태 정보에 대하여 피드백 받을 수 있는 시스템 마련

다. 자체 시스템 보호 기능 (안전장치)

- (1) 본체 시스템의 문제 발생 시 자동 reset 후 시스템을 이전 상태로 복구
- (2) 파워 릴레이 보호회로 내장으로 시스템에 대한 높은 안정성 제공
- (3) 각 전원 출력 포트에 개별 퓨즈를 내장하여 전원 단락과 같은 긴급 상황 시, 본체에 연결된 타 장비가 안전하도록 설계
- (4) 본체 보호를 위한 전원 차단용 브레이크 내장 설계

라. 시제품 제작

- (1) 제어 로직 : 펌웨어 단계에서 물리적 제어, 릴레이 형식으로 제어해서 on/off 구현할 계획
- (2) 통신망 : KT, SK에서 사용 중인 3G 모듈을 탑재해서 구현할 계획, TCP/IP 소켓 통신 모듈도 예비로 장착
- (3) 연계방법 : 최종 시제품의 일반적 성능 검증 후, 통신사와 협의를 통해 시험 사업으로 추진 예정

마. 설치비 절감을 통한 실용화 방안 모색

- (1) 기존 KT 시범 사업 설치 농가의 설치 상태 조사 및 구체적 문제점 분석
- (2) 목표 판매가를 맞출 수 있도록 장비 및 부품의 최적화 설계

제 2 절 연구성과(목표 대비 실적)

연구목표	실 적	목표달성 여부
<p>가. 원격 전원 제어 시스템을 이용한 하우스 및 농업시설물의 자동개폐장치 전원 관리</p>	<p>TCP/IP 통신을 이용하여 원격지에서 전원을 ON/OFF 할 수 있는 원격전원관리장치와 원격기에서 이를 제어할 수 있는 관제 소프트웨어를 개발하였으며 다수의 원격전원관리장치를 이용하되 위하여 관제 소프트웨어를 1:N 방식을 가능하도록 구현하였음.</p> <div style="display: flex; justify-content: space-around;">   </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 원격전원관리장치 관제 소프트웨어 </div>	<p>달성</p>
<p>나. 전원 관련 상태 정보 및 에러 발생 시 상황 모니터링</p>	<p>관제 소프트웨어에는 알람기능을 구현하였음. 알람기능은 포트별 전원의 상태(ON/OFF), 포트별 소비전류의 양의 상한 하한 값을 설정하고 이 설정된 값을 벗어나면 알람을 발생시키도록 하였으며 발생한 알람은 관리자가 “Ack”하여 알람을 확인한 후 해제하도록 하였으며 알람 이력이 시스템에 쌓이도록 하였음.</p> <div style="display: flex; justify-content: space-around;">   </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> 관제 소프트웨어 알람 관리창 </div>	<p>달성</p>
<p>다. 자체 시스템 보호 기능 (안전장치)</p>	<p>과전류 차단하기 위하여 포트별로 퓨즈를 추가하였으며 펌웨어 단에서 과전류 발생 시 시스템을 자동으로 차단하도록 하는 기능을 추가하였음. 시스템의 현재 상황을 플래시 메모리에 기록하도록 하여 일시적으로 전원이 차단된 후 인가되어도 새로 설정이 필요 없이 원래 상태로 복구되는 기능을 추가하였음.</p>	<p>달성</p>
<p>라. 기존사례 구체적 문제점 분석 및 제품 원가, 설치비 일체의 절감을 통한 실용화 방안 모색</p>	<p>기존 시스템에 대한 분석을 수행하여 문제점과 개선방안을 도출하였으며 제품 원가, 설치비 일체의 절감을 통한 목표 가격을 책정하였음.</p>	<p>달성</p>
<p>마. 단순하고 효율적인 UI를 가진 App 개발</p>	<p>농민이 출타중이거나 이동 중일 경우 간단하게 원격전원제어장치의 상태를 확인하고 전원을 제어할 수 있는 스마트폰용 앱을 개발하였음.</p> <div style="text-align: center;">  </div> <div style="text-align: center; margin-top: 5px;"> 개발된 스마트폰 앱 </div>	<p>달성</p>

제 2 장 국내외 기술개발 현황

제 1 절 국내외 기술개발 현황

1. 원격전원제어장치 개발/판매 현황

- 가. 아이소켓사는 IDC 데이터 센터의 대형 전원관리를 위하여 8포트의 원격전원관리기를 개발하여 시판하고 있음.
- 나. (주)노하우정보통신은 원격지에 있는 장비의 전원을 관리하기 위하여 Seetrol IO Socket을 출시하고 있음.
- 다. CosPan은 RPC 시리즈 제품을 출시하였으며 10A 이하의 전기용량을 갖는 1개 이상, 6개 이하의 시스템을 원격에서 각각 전원제어 및 동작 감시를 할 수 있는 표준 통신택 타입의 제품을 출시하고 있음.
- 라. (주)펠코리아는 원격에 위치한 Router, L2/L3 Gigabit Switch, CMTS, Server 등의 장비에 최대 8포트의 출력전원을 안정적으로 공급하고, 내장된 RS-232C 접속을 통하여 원격지에서 각 출력포트 별로 전원 제어, 상태 점검 및 장애 발생 등의 사항을 효율적으로 지원해주는 시스템을 출시하고 있음.
- 마. (주)세나테크놀로지는 RS232 시리얼 라인을 통해 최대 16대까지 확장이 가능한 원격 전원제어기 VTS PM 8H를 출시하고 있음.
- 사. Dataprobe사는 최대 2개까지 장치를 확장할 수 있는 원격전원 제어장치 iBoot을 출시하고 있음.

2. 전류 모니터링 / 과전류 보호회로 내장 제품

- 가. 아이소켓사의 Intelligent PDU는 전류, 전압, 전력과 에너지 소비 전력량 측정, 연결된 모든 기기에 대한 CO2 배출량 계산 및 리포트, 전력 사용 비용 분석 및 스케줄링과 관련된 실시간 저력 측정 및 모니터링 기능을 구현하고 있음.
- 나. (주)CosPan의 RPC-SP6은 과부하시 경고 메시지 표시, 정격 전류 초과 통전 시 자동 Shut Down 기능을 탑재하였음.
- 다. (주)세나테크놀로지사의 VTS PM 8H는 True RMS 전류, 전압, 전력을 실시간으로 측정하는 기능을 탑재하고 있음.

3. 제어방식에 따른 제품현황

- 가. 웹서버 내장방식 : 원격전원제어기 내부에 웹서버를 내장하여 별도의 제어 프로그램 없이 웹 브라우저를 이용하여 원격전원제어기를 제어하는 방식. 주로 1:1 제어에 유리하며 웹 브라우저를 이용하기 때문에 별도의 프로그램을 설치할 필요가 없으며 OS의 종류에 상관없이 제어가 가능함. 스마트폰에서도 별도의 앱을 설치하지 않아도 됨. 원격전원제어기가 서버의 역할을 수행하기 때문에 포트포워딩을 설정하여 주어야 하며 공인고정 IP를 필요로 함. 만약 공유기를 통한 공인유동 IP를 사용한다면 DDNS를 이용하여야 함.

- 나. 로컬 TCP/IP 서버방식 : 원격전원제어기를 서버를 이용하고 별도의 프로토콜을 구현하여 제어하는 방식. 반드시 클라이언트 프로그램인 제어프로그램이 필요함. 1:N에 유리함. 원격전원제어기가 서버의 역할을 수행하기 때문에 포트포워딩을 설정하여 주어야 하며 공인고정 IP를 필요로 함. 만약 공유기를 통한 공인유동 IP를 사용한다면 DDNS를 이용하여야 함.
- 다. 외부 중앙 집중 서버 방식 : 원격전원제어기를 클라이언트로 지정하고 클라이언트가 통신하는 서버를 외부에 중앙 집중 방식으로 설치하는 방식. 원격전원제어기를 제어하기 위해서는 중앙 서버에 접속하여야 함. 1:N 제어에 유리함. 원격전원제어기가 클라이언트이기 때문에 공인고정IP, 포트포워딩, DDNS가 필요 없음. 멀티 공유기가 사용된 환경 및 방화벽이 있는 환경에서도 별도의 설정 없이 연결이 가능함. 중앙 서버에 장애가 발생하며 이 중앙 서버를 사용하는 모든 원격전원제어기를 제어하는 것이 불가능해 짐. INTRANET에서는 원격전원제어기를 사용할 수 없음.
- 라. RS232 제어 방식 : TCP/IP를 통신수단으로 이용하기는 것이 아니라 RS232C 방식을 통신방식으로 사용함. 로컬에서만 제어가 가능하며 원격에서 제어를 구현하기 위하여 RS232C to TCP 컨버터를 이용하는 것이 일반적인 방법임.

4. 관제 소프트웨어 현황

- 가. 중앙 서버를 이용한 관제 소프트웨어 방식 : 중앙 서버에 원격전원제어장치를 등록하고 중앙 서버를 중계자로 이용하여 원격전원제어기를 제어하는 방식. 제어할 수 있는 최대 원격전원제어기의 수는 제한이 없음. 중앙 서버에 웹서버를 탑재하여 웹 브라우저를 이용하는 방식과 전용 클라이언트 프로그램을 사용하는 두 가지 방식이 있음. 중앙 서버에 계정을 개설하는 절차가 필요함
- 나. 사용자 PC를 이용한 관제 소프트웨어 방식 : 여러 대의 원격전원제어기를 등록하여 제어하는 방식. 장치의 IP와 ID, PASSWORD등을 일일이 파악하고 클라이언트 프로그램에 등록하여야 함. 관제하는 원격전원제어기의 수가 많아질수록 클라이언트 프로그램의 부하가 커짐.

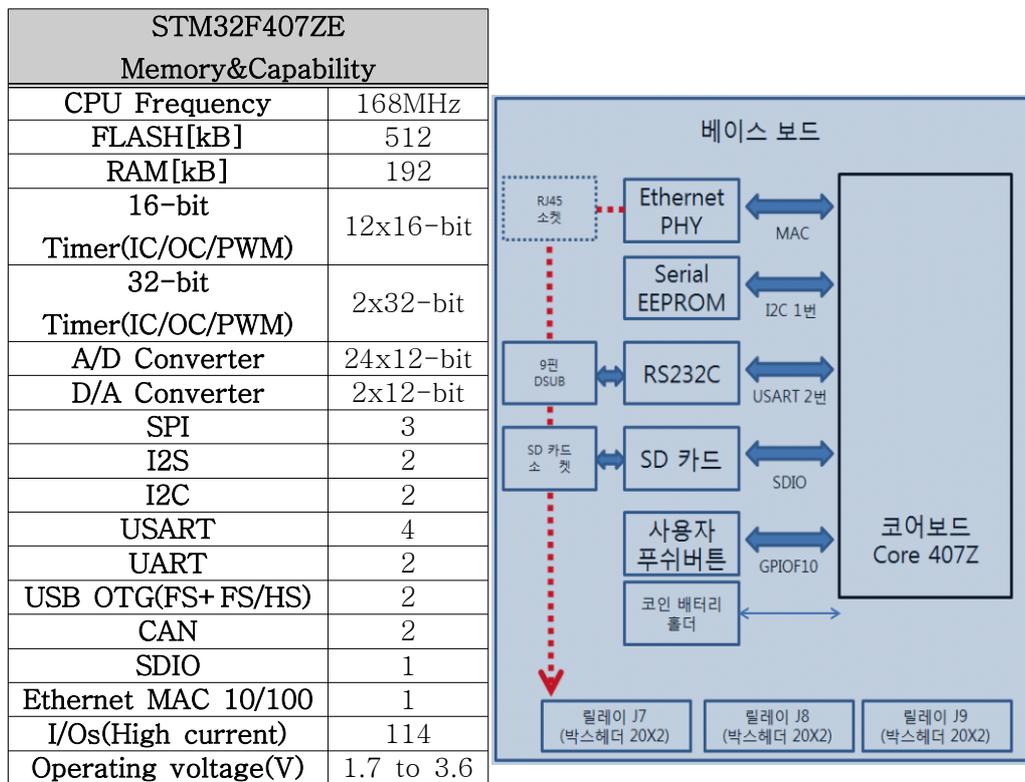
제 3 장 연구개발수행 내용 및 결과

제 1 절 연구개발수행 내용

1. 하드웨어 설계

가. 베이스 보드 설계 및 구성

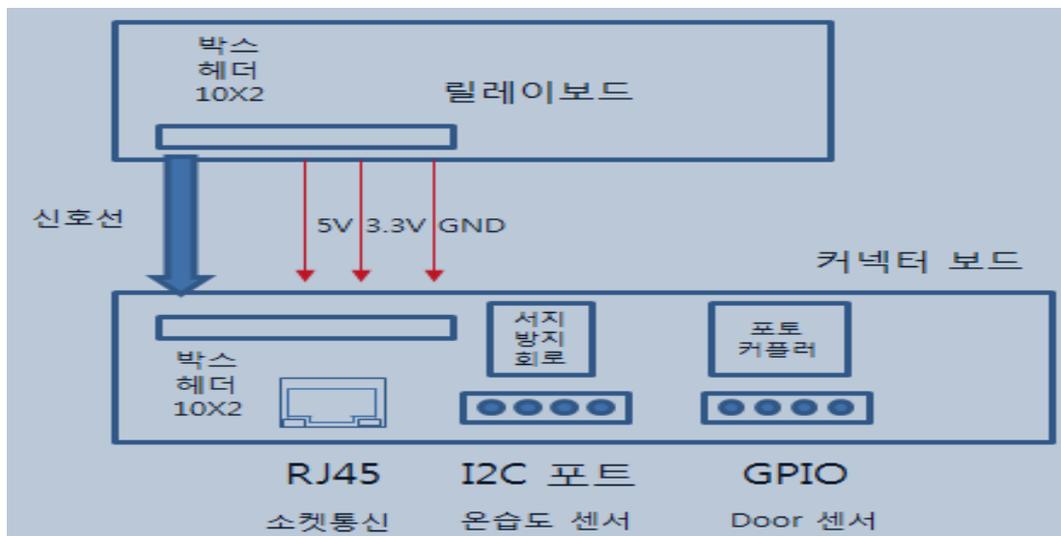
- (1) 베이스 보드의 핵심인 코어보드 MCU는 ST마이크로일렉트로닉스 사의 STM32F407ZET6 (Corex-M4)를 사용함.
- (2) 코어보드의 MCU는 원격전원제어장치의 핵심인 이더넷 기능과, 전류 센싱을 위한 A/D Converter, 동작 스케줄링에 필요한 16/32-bit 타이머, EEPROM에 내부 데이터 기록을 위한 I2C , SD Card 기록을 지원하는 SDIO/SPI, 시리얼 통신을 위한 USART등의 인터페이스를 지원함.
- (3) 또한, 코어보드의 MCU는 최대 168MHz 클럭의 속도로 동작하기 때문에 FreeRTOS의 스케줄링을 통하여 원격전원제어에 필요한 여러 Task의 안정성을 보장하고, 기본적인 네트워크 보안 구현이 가능함.
- (4) 베이스보드는 코어보드의 확장형 보드로, 원격전원제어장치에서 실제로 사용되는 인터페이스부분(I2C, USART, SDIO, Ethernet 등)과 전원 제어 및 센싱을 위한 릴레이보드를 연결하도록 설계됨.



<그림> 코어보드 사양 및 베이스 보드 구성도

나. 커넥터 보드 설계 및 구성

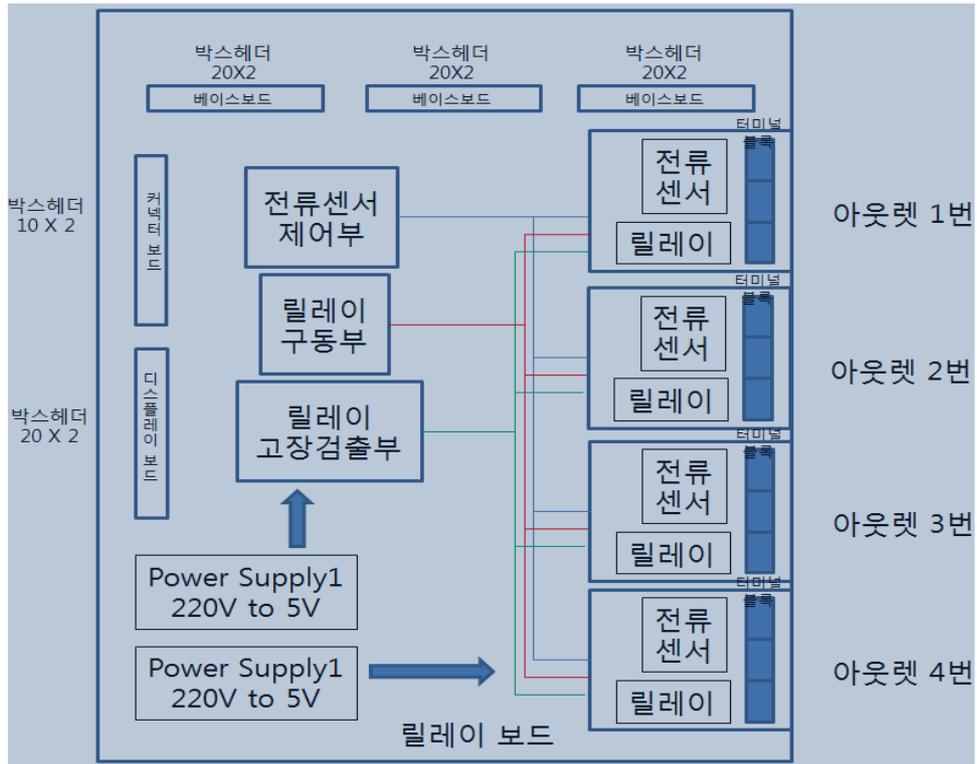
- (1) 커넥터 보드에는 TCP/IP 통신을 위한 RJ45커넥터, 온습도 센서와의 통신을 위한 I2C포트, Door센서를 위한 GPIO포트 3부분으로 구성됨.
- (2) RJ45 커넥터를 위한 전원 3.3V는 베이스보드 ->릴레이보드->커넥터보드의 경로로 전달받음.
- (3) I2C 포트의 전원은 5V를 사용하며, I2C 포트는 베이스 보드에서 I2C2를 매핑하도록 설계됨.
- (4) I2C 포트는 서지 방지회로를 부착함.
- (5) GPIO 포트는 포토커플러를 부착하여 도어 센서의 전원과 GPIO의 전원을 분리하도록 하며, 자석 센서의 채터링 방지를 위한 회로를 추가하도록 설계됨.
- (6) 또한, GPIO 포트는 외부에서 잘못된 역방향 전원인가를 방지하기 위한 다이오드를 설치함.



<그림> 커넥터보드 구성도

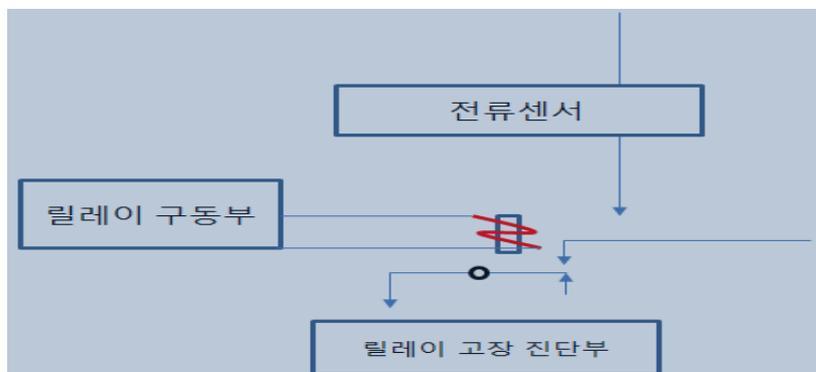
다. 릴레이 보드 설계 및 구성

- (1) 릴레이보드는 원격전원제어장치에서 실제로 전원을 제어하는 역할을 담당하는 보드임.
- (2) 릴레이보드는 220V전원을 연결할 수 있는 아울렛 포트가 4개로 구성되어 있으며, 각 아울렛 포트는 실시간으로 전류의 사용량을 측정하기 위한 전류 센서 및 릴레이로 구성되어 있음.



<그림> 릴레이보드 구성도

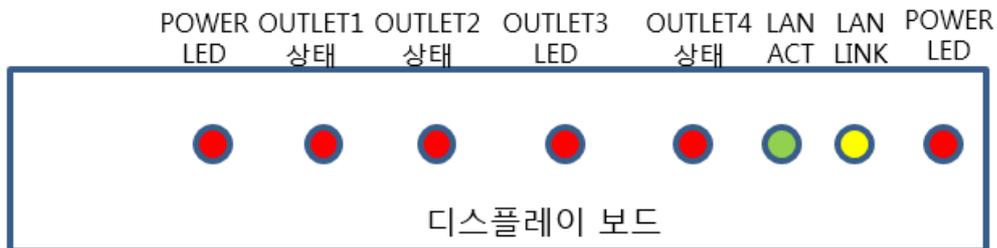
- (3) 릴레이보드에서는 0~10A의 교류전류를 감지 할 수 있는 센서를 사용하며, 아울렛은 220V의 전압을 직접 제어할 수 있도록 설계됨.
- (4) 릴레이 On/Off의 수행상태를 확인 하고, 고장을 진단할 수 있는 릴레이 고장 진단부를 구현함.
- (5) 디지털 GPIO를 이용하여 릴레이를 구동할 수 있는 전류 증폭회로를 구현함.
- (6) 추후, 아울렛을 8포트까지 확장 가능하도록 설계됨.
- (7) 또한, 커넥터/디스플레이 보드와 베이스보드를 연결하도록 설계됨.



<그림> 릴레이보드 내부 동작구성

라. 디스플레이보드 보드 설계 및 구성

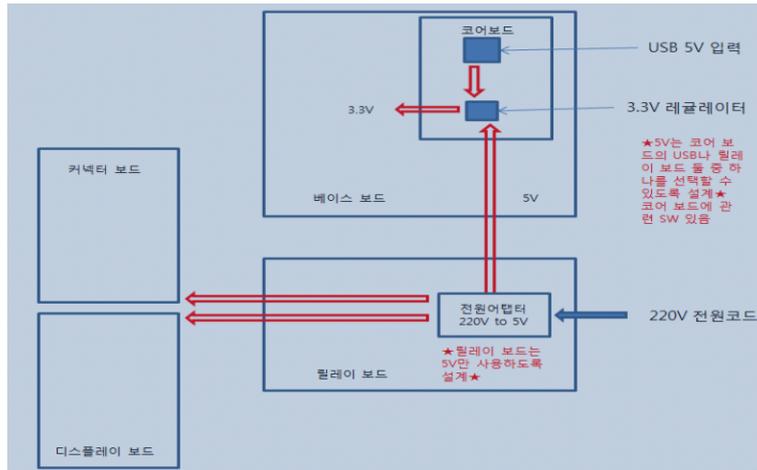
- (1) 디스플레이보드의 전원은 5V를 사용하도록 설계되어 있으며, 20X2 텍스트 LCD가 부착되도록 함.
- (2) LED는 전원의 상태를 나타내는 Power LED, 네트워크의 상태를 나타내는 LAN_ACT_LED/LAN_LINK_LED와 각 아울렛의 상태를 나타내는 OUTLET1, OUTLET2, OUTLET3, OUTLET4을 포함한 총 7개의 LED로 구성됨.
- (3) Power LED는 GPIO를 연결하여 MCU가 동작하고 있음을 나타내며 적색으로 구성함.
- (4) LAN_ACT_LED는 LAN의 동작상태를 나타내며 LAN의 PHY 신호를 연결하며 녹색으로 구성함.
- (5) LAN_LINK_LED는 LAN의 연결상태를 나타내며 LAN의 PHY 신호를 연결하며 황색으로 구성함.
- (6) OUTLET1부터 OUTLET4까지는 LED는 적색으로 구성한다.
- (7) LAN_ACT_LED와 LAN_LINK_LED에는 트랜지스터 전류 증폭회로를 삽입하여 설계함.
- (8) OUTLET LED는 3.3V 레벨을 5V로 변환하는 버퍼를 삽입하였으며, TEXT LCD의 D0에서 D7에도 3.3V레벨을 5V로 변환하는 버퍼를 삽입하여 설계함.



<그림> 디스플레이 보드 전면

마. 원격전원제어장치 하드웨어 전원 구성

- (1) 베이스보드, 커넥터보드, 디스플레이보드의 전원은 릴레이보드에서 공급되도록 구성됨.
- (2) 릴레이보드의 220V to 5V 전원어댑터에서 각 보드의 전원을 공급.
- (3) 릴레이보드는 5V만 사용하도록 설계됨.



<그림> 원격전원제어기 전원 흐름도

2. 펌웨어 설계

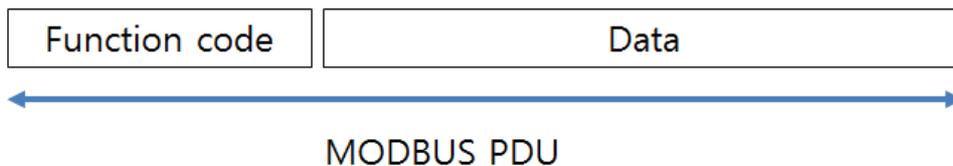
가. 원격전원제어장치의 프로토콜 MODBUS

(1) 개요

- (가) 원격전원제어장치의 프로토콜은 MODBUS 프로토콜을 기반으로 설계됨.
- (나) MODBUS는 PLC(Programmable Logic Controller)등 각종 자동화 장비의 감시 및 제어에 전 세계적으로 널리 사용되고 있는 통신 프로토콜임.
- (다) MODBUS중 TCP/IP 네트워크상에서 사용되는 버전을 MODBUS/TCP라고 하며 이는 Ethernet포트를 이용하여 통신을 하도록 설계됨.

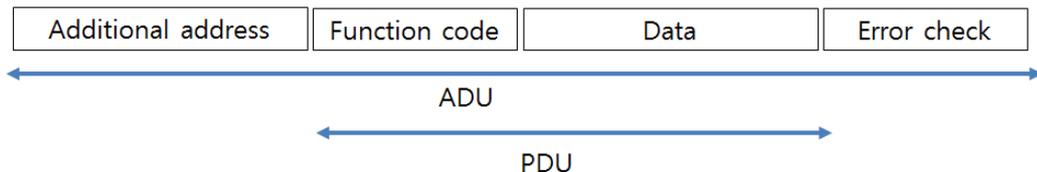
(2) MODBUS Protocol 특징

- (가) Protocol Data Unit : 하위계층과 관계없이 독립적인 데이터 구조



<그림 13> MODBUS PDU 구조

- (나) Application Data Unit : 하위 계층에 따라 최종 결정되는 데이터 구조



<그림 14> MODBUS ADU 구조

- (다) 클라이언트/서버모델 : 클라이언트는 서버에 요청(request)을 보내고 서버는 클라이언트의 요청에 응답(response)하는 방식



<그림 15> 클라이언트/서버 요청 및 응답

(3) 원격전원제어장치용 MODBUS Protocol 기본 설계

(가) 수정된 Modbus 프로토콜은 장치ID, Function Code, Data로 이루어져 있음.

<표 1> SPC 프로토콜 패킷구조 및 설명

필드명	크기	내용
Device ID	1byte	- 장치(서버)에 부여되는 1바이트 식별자 1부터 254까지 부여됨. - Device ID는 보안의 목적으로 사용하며 세션의 개념으로 세션이 부여되면 세션을 기록하는 용도로 사용할 수 있음.
Function code	1byte	- 기능 코드는 서버에 어떤 작업을 요구할지를 기술하는 1바이트 명령코드임.
Data	가변	- 기능코드에 따라 수록되는 데이터가 달라짐

(나) 패킷은 크게 Request 패킷과 Response 패킷으로 구분함. Request 패킷은 클라이언트가 서버에게 요청을 전송하는 패킷을 의미하며 Response 패킷은 서버가 클라이언트에게 응답을 보내는 패킷을 말함.

(다) 명시적으로 Request와 Response 패킷의 구분은 없으며 프로토콜의 이해와 프로토콜의 종류를 쉽게 식별하기 위하여 개념을 도입함.

(라) Request 패킷과 Response 패킷은 기능코드의 종류에 따라 세부적으로 나눠서 설계함.

<표 2> 패킷의 종류 및 설명

패킷 종류	설 명
READ_REQUEST	읽기 요청 패킷
WRITE_SINGLE_REQUEST	단일 레지스터 쓰기 패킷
WRITE_MULTIPLE_REQUEST	복수 레지스터 쓰기 패킷
READ_RESPONSE	읽기 요청 패킷에 대한 응답 패킷
WRITE_SINGLE_RESPONSE	단일 레지스터 쓰기 패킷 응답 패킷
WRITE_MULTIPLE_RESPONSE	복수 레지스터 쓰기 패킷 응답 패킷
EXCEPTION	에러코드 전송 패킷

<표 3> 기능코드 설명

코드	기능명칭	기능설명
00	COMMAND_NONE	명시된 기능코드가 없음을 나타냄
01	READ_BIT_REGISTERS	불린 레지스터를 읽음
02	READ_8BIT_REGISTERS	8비트 레지스터를 읽음
03	READ_16BIT_REGISTERS	16비트 레지스터를 읽음
04	READ_32BIT_REGISTERS	32비트 레지스터를 읽음
05	READ_FLOAT_REGISTERS	32비트 실수 레지스터를 읽음
06	READ_LAST_CODE	READ명령의 마지막 명령을 나타냄 프로그램 코드 안에서 기능코드를 구분할 때 사용됨
07	WRITE_BIT_REGISTER	단일 불린 레지스터에 쓰기
08	WRITE_8BIT_REGISTER	단일 8비트 레지스터에 쓰기
09	WRITE_16BIT_REGISTER	단일 16비트 레지스터에 쓰기
10	WRITE_32BIT_REGISTER	단일 32비트 레지스터에 쓰기
11	WRITE_FLOAT_REGISTER	단일 32비트 실수 레지스터에 쓰기
12	WRITE_SINGLE	단일 WRITE 기능코드의 마지막을 나타냄 프로그램 코드 안에서 기능코드를 구분할 때 사용됨
13	WRITE_MULTIPLE_BIT_REGISTERS	복수 불린 레지스터에 쓰기
14	WRITE_MULTIPLE_8BIT_REGISTERS	복수 8비트 레지스터에 쓰기
15	WRITE_MULTIPLE_16BIT_REGISTERS	복수 16비트 레지스터에 쓰기
16	WRITE_MULTIPLE_32BIT_REGISTERS	복수 32비트 레지스터에 쓰기
17	WRITE_MULTIPLE_FLOAT_REGISTERS	복수 32비트 실수 레지스터에 쓰기
18	WRITE_MULTIPLE	복수 WRITE 기능코드의 마지막을 나타냄 프로그램 코드 안에서 기능코드를 구분할 때 사용됨
19	WRITE_LAST_CODE	WRITE 기능코드의 마지막을 나타냄 프로그램 코드 안에서 기능코드를 구분할 때 사용됨
20	ERROR_CODE	기능코드 실행 중 에러의 발생을 알림
21	READ_SCHEDULE_ITEMS	스케줄의 아이টে임을 읽음
22	ADD_SCHEDULE_ITME	단일의 스케줄 아이টে임을 추가
23	DELETE_SCHEDULE_ITME	단일의 스케줄 아이টে임을 삭제

(4) 원격전원제어장치의 레지스터 그룹

- (가) SPC를 위한 Modbus 프로토콜의 기본 역할을 특정 레지스터를 읽거나 특정 레지스터의 값을 쓰는 것임.
- (나) 레지스터는 1비트, 8비트, 16비트, 32비트, 실수 레지스터로 구분되며 레지스터의 주소로 레지스터의 종류를 구분함.
- (다) 또한, 레지스터는 read/write 겸용과 read-only 레지스터가 있으며, 이들도 주소로 구분됨.

<표 4> 주소별 레지스터의 종류

레지스터	타입	설명
1~500	레코드	이벤트 및 알람 레코드
501~999	레코드	스케줄러 아이템 레코드
1000~1999	Boolean read/write	Digital or Discrete, 1 bit
2000~2999	Boolean read only	Digital or Discrete, 1 bit
3000~3999	8비트 read/write	8bit Integer, 1 character
4000~4999	8비트 read only	8bit Integer, 1 character
5000~5999	16비트 read/write	16bit integer
6000~6999	16비트 read only	16bit integer
7000~7999	32비트 read/write	32bit integer
8000~8999	32비트 read-only	32bit integer
9000~9999	실수(float) read/write	32bit IEEE floating point
10000~19999	실수(float) read-only	32bit IEEE floating point

(5) 레지스터 상세 설명

- (가) 레지스터의 주소는 추상적인 주소이며 실제로 추상적인 주소는 실제 로직과 매핑하는 코드를 구현함.
- (나) 레지스터의 타입에서 고정크기의 비트, 8비트, 16비트 32비트 레지스터가 있으며 레코드는 가변 크기로 주소마다 다른 크기의 데이터를 가질 수 있음.
- (다) 레코드는 타입은 정보의 추가, 조회, 삭제의 기능을 구현할 수 있으며 다른 레지스터들은 읽기 또는 쓰기의 기능만 구현이 가능함.
- (라) 현재는 정의된 레지스터 중 일부만 사용하며, 나머지 레지스터는 추후 확장을 위해 남겨둠.

<표 5> 레지스터 상세 설명

레지스터 구분	레지스터 주소	기능	내용
레코드	501	Read, Add, Delete	시간 스케줄러 아이템 저장소
단일 비트 레지스터	1001	Read/write	아울렛 1번의 ON/OFF 상태
	1002	Read/write	아울렛 2번의 ON/OFF 상태
	1003	Read/write	아울렛 3번의 ON/OFF 상태
	1004	Read/write	아울렛 4번의 ON/OFF 상태
	1005	Read/write	아울렛 5번의 ON/OFF 상태
	1006	Read/write	아울렛 6번의 ON/OFF 상태
	1007	Read/write	아울렛 7번의 ON/OFF 상태
	1008	Read/write	아울렛 8번의 ON/OFF 상태
	1009	Read/write	DHCP ON/OFF 상태
	2000 ~ 2999	Read	Reserved

8비트 레지스터	3001	Read/Write	IP주소의 첫 번째 주소(IP1)
	3002	Read/Write	IP주소의 두 번째 주소(IP2)
	3003	Read/Write	IP주소의 세 번째 주소(IP3)
	3004	Read/Write	IP주소의 네 번째 주소(IP4)
	3005	Read/Write	Subnet주소의 첫 번째 주소(SUBNET1)
	3006	Read/Write	Subnet주소의 두 번째 주소(SUBNET2)
	3007	Read/Write	Subnet주소의 세 번째 주소(SUBNET3)
	3008	Read/Write	Subnet주소의 네 번째 주소(SUBNET4)
	3009	Read/Write	Gateway주소의 첫 번째 주소(GATEWAY1)
	3010	Read/Write	Gateway주소의 두 번째 주소(GATEWAY2)
	3011	Read/Write	Gateway주소의 세 번째 주소(GATEWAY3)
	3012	Read/Write	Gateway주소의 네 번째 주소(GATEWAY4)
	3013	Read/Write	MAC주소의 첫 번째 주소(MAC1)
	3014	Read/Write	MAC주소의 두 번째 주소(MAC2)
	3015	Read/Write	MAC주소의 세 번째 주소(MAC3)
	3016	Read/Write	MAC주소의 네 번째 주소(MAC4)
	3017	Read/Write	MAC주소의 다섯 번째 주소(MAC5)
	3018	Read/Write	MAC주소의 다섯 번째 주소(MAC5)
	3019	Read/Write	장치 ID
	3020	Read/Write	장치명(NAME1)
	3021	Read/Write	장치명(NAME2)
	3022	Read/Write	장치명(NAME3)
	3023	Read/Write	장치명(NAME4)
	3024	Read/Write	장치명(NAME5)
	3025	Read/Write	장치명(NAME6)
	3026	Read/Write	장치명(NAME7)
	3027	Read/Write	장치명(NAME8)
	3028	Read/Write	장치명(NAME9)
	3029	Read/Write	장치명(NAME10)
	3030	Read/Write	Outlet포트의 상태 Read/Write Bit : 7 6 5 4 3 2 1 0 Outlet : O8 O7 O6 O5 O4 O3 O2 O1
	3031	Read/Write	장치 시간(Hours) 입력범위: 00-23
	3032	Read/Write	장치 분(Minutes) 입력범위: 00-59
	3033	Read/Write	장치 초(Seconds) 입력범위: 00-59
3034	Read/Write	장치 년도(Year) 입력범위: 00-99	
3035	Read/Write	장치 달(Month) 입력범위: 01-12	
3036	Read/Write	장치 일자(Date) 입력범위: 01-31	
3037	Read/Write	장치 요일(WeedDay) 입력범위: 01(일요일)-07(토요일)	
4000~ 4999	Read	Reserved	
16비트 레지스터	5001	Read/Write	TCP/IP 리슨 포트(PORT)
	6001	Read Only	장치 시리얼 번호(Serial)
	6002	Read Only	펌웨어 버전
	6003	Read Only	Outlet1 전류값(mA)
	6004	Read Only	Outlet2 전류값(mA)
	6005	Read Only	Outlet3 전류값(mA)
	6006	Read Only	Outlet4 전류값(mA)
	6007	Read Only	Outlet5 전류값(mA)
	6008	Read Only	Outlet6 전류값(mA)
	6009	Read Only	Outlet7 전류값(mA)
	6010	Read Only	Outlet8 전류값(mA)

32비트 레지스터	7001	Read/Write	IP Address
	7002	Read/Write	Subnet Address
	7003	Read/Write	Gateway Address
	7004	Read/Write	장치 요일, 일자, 달, 년도 입출력
	7005	Read/Write	장치 시간 분 초 입출력
	8000 ~ 8099	Read	Reserved
실수형 레지스터	9000 ~ 9999	Read/Write	Reserved
	10001	Read Only	Outlet1 전류값
	10002	Read Only	Outlet2 전류값
	10003	Read Only	Outlet3 전류값
	10004	Read Only	Outlet4 전류값
	10005	Read Only	Outlet5 전류값
	10006	Read Only	Outlet6 전류값
	10007	Read Only	Outlet7 전류값
	10008	Read Only	Outlet8 전류값

(6) 단일 비트 레지스터를 읽기 위한 요청과 응답

(가) 단일 비트 레지스터 읽기와 복수 비트 레지스터 읽기는 동일한 기능코드를 사용함.

(나) 읽을 레지스터의 개수(Quantity of Registers)가 1인지 1이상인지에 따라 단일과 복수가 결정됨.

(라) 단일 비트 레지스터 읽기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers
크기(byte)	1	1	2	2

필드명	설 명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	비트 레지스터의 시작주소
Quantity of Registers	읽을 레지스터의 개수

(7) 단일 비트 레지스터 읽기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	1

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Byte Count	Data 필드의 바이트 수
Data	Request에 읽기를 요청한 Data 1바이트 비트 값 0 -----> 0(0x00, b00000000) 비트 값 1 -----> 1(0x01, b00000001)

(8) 단일 비트 레지스터 읽기 위한 요청과 응답 예제

- (가) 원격전원제어장치의 4개의 각 아울렛에 대하여 읽기를 수행하면 아울렛의 ON/OFF 상태를 제공하며 쓰기를 수행하면 해당 아울렛을 작동하거나(ON) 작동을 멈출(OFF) 수 있음.
- (나) 각 아울렛의 상태는 비트로 기록되지만 TCP/IP에서 기본 전송의 단위가 BYTE이며 1비트를 전송시 에러의 가능성이 있으므로 상태 ON을 255(0xFF)로 표현하며 상태 OFF를 0으로 표현함.

요청패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	1	1001	1
패킷 해설	장치 ID가 100인 장치로부터 비트 레지스터 1001번 한 개를 읽어오라. 1001은 아울렛 1번의 상태를 나타내기 때문에 아울렛 1번의 ON/OFF 상태를 응답으로 요청하는 패킷의 예이다.			

응답패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	1
패킷 내용	100	1	1	1(0x01)
패킷 해설	장치 ID가 100인 장치로부터 비트 레지스터 1001번 한 개를 읽은 결과는 1의 값을 응답으로 돌려주는 패킷 비트 레지스터 1001은 아울렛 1번의 상태를 나타내기 때문에 아울렛 1번이 지금 ON 상태임을 되돌려 주는 패킷의 예이다.			

(9) 복수 비트 레지스터를 읽기 위한 요청과 응답

- (가) 단일 비트 레지스터 읽기와 복수 비트 레지스터 읽기는 동일한 기능코드를 사용함.
- (나) 읽을 레지스터의 개수(Quantity of Registers)가 1인지 1이상인지에 따라 단일과 복수가 결정됨.
- (다) 단일 비트 레지스터의 경우 비트의 0값을 1바이트 0으로 비트의 1의 값을 1바이트의 255로 표현하고 있지만 복수 비트 레지스터의 경우 여러 비트를 동시에 취급하므로 비트를 8개씩 묶어 바이트로 표현하여 값을 돌려둠.

(라) 비트의 순서는 Outlet1번의 상태가 비트 0번의 하위에 오고 Outlet2, Outlet3, Outlet4가 각각 비트 1, 비트 2, 비트 3의 순으로 배치됨.

(마) 복수 비트 레지스터 읽기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers
크기(byte)	1	1	2	2

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	비트 레지스터의 시작주소
Quantity of Registers	읽은 레지스터의 개수

(바) 비트 레지스터 읽기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	가변

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Byte Count	Data 필드의 바이트 수
Data	Request에 읽기를 요청한 Data

(사) 복수비트 레지스터 읽기위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	1	1001	4
패킷 해설	장치 ID가 100인 장치로부터 비트 레지스터 1001번부터 1004번까지 네 개를 읽어오라. 1001은 아울렛 1번, 1002는 아울렛 2번, 1003은 아울렛 3번, 1004는 아울렛 4번의 상태를 나타내고 아울렛 4개의 상태를 동시에 읽어 1바이트의 결과로 통보해 달라고 요청하는 패킷의 예이다.			

응답패킷의 구조	Device ID	Function code	Byte Count	Data																																
크기(byte)	1	1	1	1																																
패킷 내용	100	1	1	15(b00001111)																																
패킷 해설	<p>장치 ID가 100인 장치로부터 비트 레지스터 1001번부터 1004번까지 네 개를 읽은 결과 1바이트의 결과 값 15를 얻었는데 이를 비트별로 해체해 분석하여 보면 아울렛 1번, 아울렛 2번, 아울렛 3번, 아울렛 4번이 모두 ON인 상태임을 나타내는 패킷의 예이다.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Outlet8</th> <th>Outlet7</th> <th>Outlet6</th> <th>Outlet5</th> <th>Outlet4</th> <th>Outlet3</th> <th>Outlet2</th> <th>Outlet1</th> </tr> <tr> <th>비트7</th> <th>비트 6</th> <th>비트 5</th> <th>비트 4</th> <th>비트 3</th> <th>비트 2</th> <th>비트 1</th> <th>비트 0</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>ON</td> <td>ON</td> <td>ON</td> <td>ON</td> </tr> </tbody> </table>				Outlet8	Outlet7	Outlet6	Outlet5	Outlet4	Outlet3	Outlet2	Outlet1	비트7	비트 6	비트 5	비트 4	비트 3	비트 2	비트 1	비트 0	0	0	0	0	1	1	1	1	OFF	OFF	OFF	OFF	ON	ON	ON	ON
Outlet8	Outlet7	Outlet6	Outlet5	Outlet4	Outlet3	Outlet2	Outlet1																													
비트7	비트 6	비트 5	비트 4	비트 3	비트 2	비트 1	비트 0																													
0	0	0	0	1	1	1	1																													
OFF	OFF	OFF	OFF	ON	ON	ON	ON																													

(10) 단일/복수 8비트 레지스터를 읽기 위한 요청과 응답

- (가) 단일 8비트 레지스터 읽기와 복수 8비트 레지스터 읽기는 동일한 기능코드를 사용함.
- (나) 읽을 레지스터의 개수(Quantity of Registers)가 1인지 1이상인지에 따라 단일과 복수가 결정됨.
- (다) 8비트 레지스터의 상태는 8비트(1바이트)이므로 응답패킷의 Data의 크기는 레지스터의 개수와 동일하게 됨.
- (라) 단일/복수 8비트 레지스터 읽기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers
크기(byte)	1	1	2	2

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	8비트 레지스터의 시작주소(3001번지~4999번지)
Quantity of Registers	읽을 레지스터의 개수

(마) 단일/복수 8비트 레지스터 읽기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	가변 길이

필드명	설 명
Device ID	장치의 ID 식별자
Function code	기능코드
Byte Count	Data 필드의 바이트 수
Data	실제로 전달된 데이터

(바) 단일/복수 8비트 레지스터를 읽기 위한 요청과 응답 예제

- SPC에는 4바이트로 IP 주소 정보를 저장하고 있음.
- 아래의 예제는 장치의 IP 주소값을 동시에 가져오는 요청을 보내고 이에 대한 응답을 받고, 장치의 IP 주소값은 192.168.0.15로 가정함.

요청패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	2	3001	4
패킷 해설	장치 ID가 100인 장치로부터 8비트 레지스터 3001번부터 3004번까지 네 개의 레지스터를 읽어오라. 3001은 장치의 IP 주소의 첫 번째 바이트, 3002, 3003, 3004는 두 번째, 세 번째, 네 번째 주소를 나타므로 결국 장치의 IP 전체를 읽어오는 것을 요청하는 패킷의 예이다.			

응답패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	4(가변)
패킷 내용	100	2	4	192 168 0 15
패킷 해설	장치 ID가 100인 장치로부터 IP주소를 읽은 결과가 192.168.0.15임을 알 수 있다.			

(11) 단일/복수 16비트 레지스터를 읽기 위한 요청과 응답

- (가) 단일 16비트 레지스터 읽기와 복수 16비트 레지스터 읽기는 동일한 기능코드를 사용함.
- (나) 읽을 레지스터의 개수(Quantity of Registers)가 1인지 1이상인지에 따라 단일과 복수가 결정됨.
- (다) 16비트 레지스터의 상태는 16비트(2바이트)이므로 응답패킷의 Data의 크기는 레지스터의 개수x2와 동일하게 됨.
- (라) 단일/복수 16비트 레지스터 읽기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers
크기(byte)	1	1	2	2

필드명	설 명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	16비트 레지스터의 시작주소(5001번지~6999번지)
Quantity of Registers	읽을 레지스터의 개수

(마) 단일/복수 16비트 레지스터 읽기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	가변 길이

필드명	설 명
Device ID	장치의 ID 식별자
Function code	기능코드
Byte Count	Data 필드의 바이트 수
Data	실제로 전달된 데이터

(마) 단일/복수 16비트 레지스터를 읽기 위한 요청과 응답 예제

- SPC에는 2바이트로 TCP/IP의 포트번호를 저장하고 있음.
- 장치의 포트번호를 가져오는 요청을 보내고 이에 대한 응답을 받고, 장치의 포트번호는 12500이라고 가정함.

요청패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	3	5001	1
패킷 해설	장치 ID가 100인 장치로부터 16비트 레지스터 5001번부터 한 개의 레지스터를 읽어오라. 5001은 장치의 포트번호를 저장하는 레지스터 번호이다.			

응답패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	2(가변)
패킷 내용	100	3	2	0x30 0xD4
패킷 해설	장치 ID가 100인 장치로부터 포트 번호를 가져온 결과 상위 바이트 0x30와 하위 바이트 0xD4로 이를 조합하여 16비트 정수를 만들면 십진수로 12500이 된다.			

(12) 단일/복수 32비트 레지스터를 읽기 위한 요청과 응답

- (가) 단일 32비트 레지스터 읽기와 복수 32비트 레지스터 읽기는 동일한 기능코드를 사용함.
- (나) 읽을 레지스터의 개수(Quantity of Registers)가 1인지 1이상인지에 따라 단일과 복수가 결정됨.
- (다) 32비트 레지스터의 상태는 32비트(4바이트)이므로 응답패킷의 Data의 크기는 레지스터의 개수x4와 동일하게 됨.
- (라) 단일/복수 32비트 레지스터 읽기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers
크기(byte)	1	1	2	2

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	32비트 레지스터의 시작주소(7001번지~8999번지)
Quantity of Registers	읽을 레지스터의 개수

(마) 단일/복수 32비트 레지스터 읽기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	가변 길이

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Byte Count	Data 필드의 바이트 수
Data	실제로 전달된 데이터

(사) 단일/복수 32비트 레지스터를 읽기 위한 요청과 응답 예제

- SPC에는 4바이트로 IP 주소를 저장하고 있음.
- 앞의 예에서 IP를 8비트 레지스터 4개를 통하여 가져오는 예를 설명하였는데 동일하기 32비트 레지스터 1개를 이용하여 IP를 다룰 수 있도록 SPC를 매핑함.
- 32비트 레지스터의 주소는 7001번이며, IP주소는 192.168.0.15라고 가정함.

요청패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	4	7001	1
패킷 해설	장치 ID가 100인 장치로부터 32비트 레지스터 7001번부터 한 개의 레지스터를 읽어오라. 7001은 장치의 IP 주소를 저장하는 레지스터 번호이다.			

응답패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	4(가변)
패킷 내용	100	4	4	192 168 0 15
패킷 해설	장치 ID가 100인 장치로부터 IP 주소를 가져온 결과 192.168.0.15이다.			

(13) 단일/복수 float비트 레지스터를 읽기 위한 요청과 응답

(가) 단일 float 레지스터 읽기와 복수 float 레지스터 읽기는 동일한 기능코드를 사용함.

(나) 읽을 레지스터의 개수(Quantity of Registers)가 1인지 1이상인지에 따라 단일과 복수가 결정됨.

(다) float 레지스터의 상태는 32비트(4바이트)이므로 응답패킷의 Data의 크기는 레지스터의 개수x4와 동일하게 됨.

(라) 단일/복수 float 레지스터 읽기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers
크기(byte)	1	1	2	2

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	float 레지스터의 시작주소(9001번지~10999번지)
Quantity of Registers	읽을 레지스터의 개수

(마) 단일/복수 float 레지스터 읽기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	가변 길이

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Byte Count	Data 필드의 바이트 수
Data	실제로 전달된 데이터

(바) 단일/복수 float 레지스터를 읽기 위한 요청과 응답 예제

- SPC에는 각 Outlet마다 float로 전류값을 저장하고 있음.
- 아래의 예는 4개의 아울렛에서 전류값을 가져오는 요청과 응답을 예로 표현한 것임.

요청패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	5	10001	4
패킷 해설	장치 ID가 100인 장치로부터 float 레지스터 10001번부터 네 개의 레지스터를 읽어오라. 10001, 10002, 10003, 10004는 각각 Outlet1, Outlet2, Outlet3, Outlet4의 전류값을 저장하는 레지스터 번호이다. 따라서 결국 아울렛 4개의 전류값을 응답 패킷에 실어서 보내도록 한다.			

응답패킷의 구조	Device ID	Function code	Byte Count	Data
크기(byte)	1	1	1	16(가변)
패킷 내용	100	5	16	12.3 58.6 47.2 12.5
패킷 해설	장치 ID가 100인 장치로부터 Outlet 네 개의 전류값을 가져온 결과 Outlet1 = 12.3, Outlet2 = 58.6, Outlet3 = 47.2, Outlet4 = 12.5이다.			

(14) 단일비트 레지스터를 쓰기 위한 요청과 응답

- (가) 단일 비트 레지스터 쓰기와 복수 비트 레지스터 쓰기는 서로 다른 기능코드를 사용함.
- (나) 단일 비트 레지스터 쓰기와 복수 비트 레지스터 쓰기는 쓰고자하는 데이터의 크기가 고정과 가변으로 차이가 나기 때문에 서로 다른 기능코드를 사용하며 프레임의 구조 또한 다름.
- (다) 비트 레지스터의 쓰기를 위한 데이터 값은 읽기와 같이 비트 값이 0일 때 1바이트의 값 0으로 표현하고 비트 값이 1일 때 1바이트의 값 255로 표현함.
- (라) 단일 비트 레지스터 쓰기의 요청 패킷과 응답 패킷은 완전히 동일한 구조를 가짐.
- (마) 단일 비트 레지스터를 쓰기를 요청한 후 쓰기 과정에서 오류가 없다면 동일한 패킷을 반송함.

(바) 만약 오류가 발생하면 오류코드를 응답으로 전송함.

(사) 단일 비트 레지스터 쓰기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	1

필드명	설 명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	비트 레지스터의 시작주소
Data	기록할 바이트 데이터 값(비트 값 0 : 0, 비트 값 1: 255)

(아) 단일 비트 레지스터 쓰기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	1

필드명	설 명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	비트 레지스터의 시작주소
Data	기록할 바이트 데이터 값(비트 값 0 : 0, 비트 값 1: 255)

(자) 단일 비트 레지스터를 쓰기 위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Start Address	Data
크기(byte)	1	1	2	1
패킷 내용	100	7	1001	255
패킷 해설	장치 ID가 100인 장치로부터 비트 레지스터 1001번에 데이터 255의 값을 기록하라. 1001은 아울렛 1번의 상태를 나타내기 때문에 여기에 255값을 쓰는 것은 아울렛 1번을 ON 상태로 만드는 예이다.			

응답패킷의 구조	Device ID	Function code	Start Address	Data
크기(byte)	1	1	2	1
패킷 내용	100	7	1001	255
패킷 해설	요청이 성공적으로 수행되어 응답으로 동일한 패킷을 받는다.			

(15) 단일 8비트 레지스터를 쓰기 위한 요청과 응답

(가) 단일 8비트 레지스터 쓰기와 복수 8비트 레지스터 쓰기는 서로 다른 기능코드를 사용함.

(나) 위에서 설명한 단일비트에 대한 내용과 동일하나 비트 수와 레지스터 주소가 다름.

(다) 단일 8비트 레지스터 쓰기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	1

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	8비트 레지스터의 시작주소
Data	기록할 바이트 데이터 값

(라) 단일 8비트 레지스터 쓰기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	1

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	8비트 레지스터의 시작주소
Data	기록할 바이트 데이터 값

(마) 단일 8비트 레지스터를 쓰기 위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Start Address	Data
크기(byte)	1	1	2	1
패킷 내용	100	8	3017	77
패킷 해설	장치 ID가 100인 장치로부터 8비트 레지스터 3017번에 데이터 77의 값을 기록하라. 3017은 장치의 ID를 나타내기 때문에 여기에 77의 값을 쓰는 것은 장치 ID를 77로 변경하는 결과를 만든다.			

응답패킷의 구조	Device ID	Function code	Start Address	Data
크기(byte)	1	1	2	1
패킷 내용	100	8	3017	77
패킷 해설	요청이 성공적으로 수행되어 응답으로 동일한 패킷을 받는다.			

(16) 단일 16비트 레지스터를 쓰기 위한 요청과 응답

(가) 16비트 레지스터 쓰기와 복수 16비트 레지스터 쓰기는 서로 다른 기능코드를 사용함.

(나) 위에서 설명한 단일비트에 대한 내용과 동일하나 비트 수와 레지스터 주소가 다름

(다) 단일 16비트 레지스터 쓰기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	2

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	16비트 레지스터의 시작주소
Data	기록할 바이트 데이터 값(2바이트 16비트)

(라) 단일 16비트 레지스터 쓰기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	2

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	16비트 레지스터의 시작주소
Data	기록할 바이트 데이터 값(2바이트 16비트)

(마) 단일 16비트 레지스터를 쓰기 위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Start Address	Data
크기(byte)	1	1	2	2
패킷 내용	100	9	5001	0x4 0xE7
패킷 해설	장치 ID가 100인 장치로부터 16비트 레지스터 5001번에 데이터 12500의 값을 기록하라. 5001은 장치의 TCP/IP 포트번호를 나타내기 때문에 여기에 12500의 값을 쓰는 것은 장치의 TCP/IP 포트번호를 12500으로 변경하는 결과를 만든다.			

응답패킷의 구조	Device ID	Function code	Start Address	Data
크기(byte)	1	1	2	1
패킷 내용	100	9	5001	0x4 0xE7
패킷 해설	요청이 성공적으로 수행되어 응답으로 동일한 패킷을 받는다.			

(17) 단일 32비트 레지스터를 쓰기 위한 요청과 응답

(가) 단일 32비트 레지스터 쓰기와 복수 32비트 레지스터 쓰기는 서로 다른 기능 코드를 사용함.

(나) 위에서 설명한 단일비트에 대한 내용과 동일하나 비트 수와 레지스터 주소가 다름

(다) 32비트 레지스터 쓰기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	4

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	32비트 레지스터의 시작주소
Data	기록할 바이트 데이터 값(4바이트 32비트)

(라) 32비트 레지스터 쓰기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	4

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	32비트 레지스터의 시작주소
Data	기록할 바이트 데이터 값(4바이트 32비트)

(마) 단일 32비트 레지스터를 쓰기 위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Start Address	Data			
크기(byte)	1	1	2	4			
패킷 내용	100	10	7001	192	168	0	1
패킷 해설	장치 ID가 100인 장치로부터 32비트 레지스터 7001번에 데이터 192 168 0 1의 값을 기록하라. 7001은 장치의 IP 주소를 나타내기 때문에 여기에 192 168 0 1의 값을 쓰는 것은 장치의 IP 주소를 192.168.0.1로 변경하고자 하는 요청이다.						

응답패킷의 구조	Device ID	Function code	Start Address	Data			
크기(byte)	1	1	2	4			
패킷 내용	100	10	7001	19 2	16 8	0	1
패킷 해설	요청이 성공적으로 수행되어 응답으로 동일한 패킷을 받는다.						

(18) 단일 float 레지스터를 쓰기 위한 요청과 응답

(가) 단일 float 레지스터 쓰기와 복수 float 레지스터 쓰기는 서로 다른 기능코드를 사용함.

(나) 위에서 설명한 단일비트에 대한 내용과 동일하나 비트 수와 레지스터 주소가 다름

(다) float 레지스터 쓰기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	4

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	float 레지스터의 시작주소
Data	기록할 바이트 데이터 값(4바이트 32비트)

(라) float 레지스터 쓰기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Data
크기(byte)	1	1	2	4

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	float 레지스터의 시작주소
Data	기록할 바이트 데이터 값(4바이트 32비트)

(19) 복수 비트 레지스터를 쓰기 위한 요청과 응답

(가) 단일 비트 레지스터 쓰기와 복수 비트 레지스터 쓰기는 서로 다른 기능코드를 사용함.

(나) 단일 비트 레지스터 쓰기와 복수 비트 레지스터 쓰기는 쓰고자하는 데이터의 크기가 고정과 가변으로 차이가 나기 때문에 서로 다른 기능코드를 사용하며 프레임의 구조 또한 다름.

(다) 복수 비트 레지스터 쓰기는 실제로 기록할 데이터를 패킷 끝에 첨부하여 요청 패킷을 만들고 이 때, 데이터의 총 바이트수를 알려주기 위하여 Byte

Count라는 필드를 데이터의 앞 부분에 두어 데이터의 크기를 바이트 수로 기록함.

- (라) 비트 레지스터의 기록할 값은 1비트이므로 복수 비트 레지스터 읽기의 경우와 동일하게 이를 비트 스트림으로 붙여 8비트 단위로 끊어 데이터를 제작함.
- (마) 복수 비트 쓰기에서 오류가 발생하지 않으면 Byte Count와 Data 부분을 제외한 헤더 부분을 그대로 복사하여 응답 패킷으로 전송함.
- (바) 만약 오류가 발생하면 오류코드를 응답으로 전송함.
- (사) 복수 비트 레지스터 쓰기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers	Byte Count	Data
크기(byte)	1	1	2	2	1	Byte Count

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	비트 레지스터의 시작주소
Quantity of Registers	기록할 레지스터의 개수
Byte Count	기록할 데이터의 총 바이트 수
Data	기록할 바이트 데이터 값

- (아) 복수 비트 레지스터 쓰기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers
크기(byte)	1	1	2	2

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	비트 레지스터의 시작주소
Quantity of Registers	기록할 레지스터의 개수

- (자) 복수 비트 레지스터를 쓰기 위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers	Byte Count	Data
크기(byte)	1	1	2	2	1	1
패킷 내용	100	13	1001	4	1	4(b00001111)
패킷 해설	장치 ID가 100인 장치로부터 비트 레지스터 1001번부터 1004번까지 모두 상태를 ON으로 기록하는 패킷					

응답패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	13	1001	4
패킷 해설	요청이 성공적으로 수행되어 응답으로 동일한 패킷을 받는다.			

(20) 복수 8비트 레지스터를 쓰기 위한 요청과 응답

(가) 단일 8비트 레지스터 쓰기와 복수 8비트 레지스터 쓰기는 서로 다른 기능코드를 사용함.

(나) 위에서 설명한 복수 비트에 대한 내용과 동일하나 비트 수와 레지스터 주소가 다름

(다) 복수 8비트 레지스터 쓰기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers	Byte Count	Data
크기(byte)	1	1	2	2	1	Byte Count

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	비트 레지스터의 시작주소
Quantity of Registers	기록할 레지스터의 개수
Byte Count	기록할 데이터의 총 바이트 수
Data	기록할 바이트 데이터 값

(라) 복수 8비트 레지스터 쓰기 응답 패킷 구조

패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers
크기(byte)	1	1	2	2

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드
Starting Address	비트 레지스터의 시작주소
Quantity of Registers	기록할 레지스터의 개수

(마) 복수 8비트 레지스터를 쓰기 위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Starting Address	Quantity of Registers	Byte Count	Data
크기(byte)	1	1	2	2	1	4
패킷 내용	100	14	3001	4	4	192 168 0 15
패킷 해설	장치 ID가 100인 장치로부터 IP 주소 192.168.0.15를 기록하는 예제					

응답패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	14	3001	4
패킷 해설	요청이 성공적으로 수행되어 응답으로 동일한 패킷을 받는다.			

(21) 시간 스케줄 아이템의 구조

- (가) 시간 스케줄 아이템은 특정 시간에 아울렛을 ON/OFF/TOGGLE하게 만드는 스케줄기능을 가진 항목임.
- (나) 스케줄 아이템은 원격전원제어장치의 전원이 OFF되어 있는 상태에서도 기록이 되어 있어야 하므로, 물리적인 저장은 I2C인터페이스를 통한 EEPROM에 기록이 됨.
- (다) EEPROM에는 스케줄 아이템 뿐만 아니라, 네트워크 정보, 기능 정보 등을 저장이 되도록 구현되어 있음.
- (라) EEPROM의 물리적인 용량 관계상 스케줄 아이템의 각 데이터 항목을 비트 단위로 최소화 하여 설계함.
- (마) 시간 스케줄 아이템을 저장하는 레코드 레지스터는 501으로 정의되어 있음.
- (사) 스케줄 아이템 세부 구조

필드명	설명	값의 범위
state (2 bits)	타임 스케줄 아이템의 상태를 나타냄 상태 disabled는 타임 스케줄 아이템이 존재하나 동작하지 않도록 설정된 상태이고 상태 "enalbed"는 타임 스케줄 아이템이 동작하도록 설정한 상태임. 상태 "deleted"는 타임 스케줄 아이템을 삭제한 상태를 나타냄	00: reserved 01: disabled 10: enabled 11: deleted

type (3 bits)	타임 스케줄 아이템의 종류를 나타냄 -특정날짜 타이머 : 월/일/시/분에 동작하는 스케줄. 요일값은 무시됨 -주간 : 특정 요일/시/분마다 반복하는 스케줄. 월/일 설정값은 무시됨 -일간 : 특정 시/분마다 반복하는 스케줄. 월/일/요일값은 무시됨 -시간 : 특정 분마다 반복하는 스케줄. 월/일/요일/시각값은 무시됨 -매분간 : 특정분의 배수분마다 반복하는 스케줄. 월/일/요일/시각값은 무시됨	000: reserved 001: 특정날짜 010: 주간 011: 일간 100: 시간 101: 매분간 110-111 : reserved
month (4 bits)	월을 저장하는 필드	1 - 12
day (5 bits)	일을 저장하는 필드	1 - 31
weekday (3 bits)	요일을 저장하는 필드	1 - 7
hour (5 bits)	시간을 저장하는 필드	0 - 23
minute (6 bits)	분을 저장하는 필드	0 - 59
operation (2 bits)	특정시간에 아울렛을 ON할지 OFF할지 TOGGLE할지를 저장하는 필드	00: reserved 01: ON 10: OFF 11: TOGGLE
reserved (2 bits)	향후 사용을 위하여 예약	
outlet_num (1Byte)	스케줄이 적용될 아울렛을 저장하는 필드	0 - 31

(아) 다음은 사용자가 지정한 시간에 따라 아울렛의 상태를 제어하기 위한 RTC 관련 시간제어 레지스터에 대한 설명임.

레지스터 번호	타입(수행)	설 명	비고
3031	BYTE(Read/Write)	장치의 시간을 읽거나 씀. 시간은 24시간 기준으로 00시부터 23시까지 입력가능.	장치 시간(Hours) 입력범위: 00-23
3032	BYTE(Read/Write)	장치의 분을 읽거나 씀. 분은 00분부터 59분까지 입력가능.	장치 분(Minutes) 입력범위: 00-59
3033	BYTE(Read/Write)	장치의 초를 읽거나 씀. 초는 00초부터 59초까지 입력가능.	장치 초(Seconds) 입력범위: 00-59
3034	BYTE(Read/Write)	장치의 년도를 읽거나 씀. 년도는 2014년 이후의 14만 입력. 00부터 99까지 입력가능.	장치 년도(Year) 입력범위: 00-99

3035	BYTE(Read/Write)	장치의 달을 읽거나 씌. 달은 01월부터 12월까지 입력가능.	장치 달(Month) 입력범위: 01-12								
3036	BYTE(Read/Write)	장치의 날짜를 읽거나 씌. 날짜는 01일부터 31일까지 입력가능.	장치 일자(Date) 입력범위: 01-31								
3037	BYTE(Read/Write)	장치의 요일을 읽거나 씌. 요일은 일요일(01)부터 토요일(07)까지 입력가능.	장치 요일(WeedDay) 입력범위: 01(일요일)-07(토요일)								
7004	DWORD(Read/Write)	장치의 연월일 요일을 동시에 읽거나 씌	데이터 순서: 요일(1byte)-일자(1byte)-달(1byte)-년도(byte) <table border="1" style="margin-left: 20px;"> <tr> <td>연</td><td>월</td><td>일</td><td>요일</td></tr> <tr> <td colspan="3">bit 31</td><td>bit 0</td></tr> </table> (출력되는 순서는 역순임)	연	월	일	요일	bit 31			bit 0
연	월	일	요일								
bit 31			bit 0								
7005	DWORD(Read/Write)	장치의 시간과 분, 초를 동시에 읽거나 씌	데이터 순서 : 0x00(1byte)-시간(1byte)-분(1byte)-초(1byte) <table border="1" style="margin-left: 20px;"> <tr> <td>시간</td><td>분</td><td>초</td><td>0x00</td></tr> <tr> <td colspan="3">bit 31</td><td>bit 0</td></tr> </table>	시간	분	초	0x00	bit 31			bit 0
시간	분	초	0x00								
bit 31			bit 0								

(22) 2.5.15 스케줄 레지스터 읽기를 위한 요청과 응답

(가) 스케줄 아이템 레지스터 읽기 요청 패킷 구조

패킷의 구조	Device ID	Function code	Address of Register	Outlet Number
크기(byte)	1	1	2	1

필드명	설명
Device ID	장치의 ID 식별자
Function Code	기능코드(21, READ_SCHEDULE_ITEMS)
Address of Register	레지스터의 주소(501 ~ 1000)
Outlet Number	해당 스케줄 아이템을 가진 아울렛의 번호(0 : Outlet 1, 1:Outlet 2, 255: All of the Oulets)

(나) 스케줄 아이템 레지스터 읽기 패킷 구조

패킷의 구조	Device ID	Function code	Quantity of Items	Byte Count	Data
크기(byte)	1	1	2	1	Byte Count

필드명	설명
Device ID	장치의 ID 식별자
Function Code	기능코드(21, READ_SCHEDULE_ITEMS)
Quantity of Items	스케줄 아이템의 개수
Byte Count	데이터의 총 바이트 수
Data	데이터의 총 바이트 수만큼의 스케줄 아이템 데이터

(다) 스케줄 아이템 레지스터 읽기 위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Address of Register	Outlet Number
크기(byte)	1	1	2	1
패킷 내용	100	21	501	255
패킷 해설	장치 ID가 100인 장치로부터 모든 아울렛에 설정된 시간 스케줄 아이템을 가져오도록 요청한다.			

응답패킷의 구조	Device ID	Function code	Quantity of Items	Byte Count	Data
크기(byte)	1	1	2	1	180
패킷 내용	100	21	30	180	Time schedule Item
패킷 해설	요청이 성공적으로 수행되어 모든 아울렛에 설정된 시간 스케줄 아이템을 가져온다.				

(23) 단일 스케줄 레지스터 추가를 위한 요청과 응답

(가) 스케줄 아이템 레지스터 추가요청 패킷 구조

패킷의 구조	Device ID	Function code	Address of Register	Schedule Item
크기(byte)	1	1	2	5

필드명	설명
Device ID	장치의 ID 식별자
Function Code	기능코드(21, ADD_SCHEDULE_ITEMS)
Address of Register	레지스터의 주소(501 ~ 1000)
Time Schedule Item	추가하고자 하는 시간 스케줄 아이템 데이터

(나) 스케줄 아이템 레지스터 추가 패킷 구조

패킷의 구조	Device ID	Function Code	Address of Register	Schedule Item_Index	Schedule Item
크기(byte)	1	1	2	1	5

필드명	설명
Device ID	장치의 ID 식별자
Function Code	기능코드(21, ADD_SCHEDULE_ITEMS)
Address of Register	레지스터의 주소(501 ~ 1000)
Schedule Item	추가하고자 하는 스케줄 아이템 데이터

(다) 단일 스케줄 아이템 레지스터 추가를 위한 요청과 응답 예

요청패킷의 구조	Device ID	Function code	Address of Register	Time Schedule Item
크기(byte)	1	1	2	5
패킷 내용	100	21	501	21 3 23 56 20
패킷 해설	장치 ID가 100인 장치로부터 모든 아울렛에 설정된 시간 스케줄 아이템을 가져오도록 요청한다.			

응답패킷의 구조	Device ID	Function code	Address of Register	Time Schedule Item
크기(byte)	1	1	2	5
패킷 내용	100	21	501	21 3 23 56 20
패킷 해설	장치 ID가 100인 장치로부터 추가한 패킷을 그대로 응답 패킷으로 돌려받는다.			

(24) 단일 스케줄 레지스터 삭제를 위한 요청과 응답

(가) 스케줄 아이템 레지스터 삭제 요청 패킷 구조

패킷의 구조	Device ID	Function code	Address of Register	Index of Schedule Item
크기(byte)	1	1	2	1

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드(21, READ_SCHEDULE_ITEMS)
Address of Register	레지스터의 주소(501 ~ 1000)
Index of Schedule Item	삭제하고자 하는 스케줄 아이템의 인덱스

(나) 스케줄 아이템 레지스터 삭제 패킷 구조

패킷의 구조	Device ID	Function code	Address of Register	Index of Schedule Item
크기(byte)	1	1	2	1

필드명	설명
Device ID	장치의 ID 식별자
Function code	기능코드(21, READ_SCHEDULE_ITEMS)
Address of Register	레지스터의 주소(501 ~ 1000)
Time Schedule Item	추가하고자 하는 시간 스케줄 아이템 데이터

(다) 단일 스케줄 아이템 레지스터 삭제를 위한 요청과 응답 예제

요청패킷의 구조	Device ID	Function code	Address of Register	Schedule Item Index
크기(byte)	1	1	2	1
패킷 내용	100	21	501	12
패킷 해설	장치 ID가 100인 장치로부터 모든 아울렛에 설정된 시간 스케줄 아이템을 가져오도록 요청한다.			

응답패킷의 구조	Device ID	Function code	Address of Register	Index of Schedule Item
크기(byte)	1	1	2	1
패킷 내용	100	21	501	1
패킷 해설	장치 ID가 100인 장치로부터 추가한 패킷을 그대로 응답 패킷으로 돌려받는다.			

(25) ERROR 답신 패킷

(가) 전달된 요청을 정상적으로 처리하지 못할 때 ERROR 답신 패킷을 응답패킷으로 전송하여 에러 상황을 알림.

(나) ERROR 상황은 기능이 확장될수록 해당 기능에 대한 오류 처리 항목을 추가해야 하며, 현재는 기본적인 상황만 고려하여 설계함.

(다) ERROR 답신 패킷의 구조

패킷의 구조	Device ID	Function code	Error Code
크기(byte)	1	1	1

필드명	설 명
Device ID	장치의 ID 식별자 예외)Error code 1의 경우 Device ID를 모르는 상황이기 때문에 반드시 Device ID를 0으로 세팅한다.
Function code	기능코드 에러를 반환하는 패킷을 나타내기 위하여 반드시 Function Code 20 을 사용하여야 함
Error Code	에러를 나타내는 0부터 255까지의 에러 코드를 명기

(라) ERROR 코드의 정의

Error Code	Error Code 설명	기타
0	에러가 발생하지 않았습니다.	
1	Device ID가 일치하지 않습니다.	
2	존재하지 않는 레지스터 번호입니다.	
3	ReadOnly 레지스터에 쓰기 시도를 하였습니다.	
4	허용할 수 있는 값의 범위를 초과합니다	
5 ~ 500	ERROR 사항 추가 예정	

(마) 에러상황을 발생하는 패킷전송과 에러 답신 패킷의 예제

요청패킷의 구조	Device ID	Function code	Start Address	Quantity of Registers
크기(byte)	1	1	2	2
패킷 내용	100	1	3999	1
패킷 해설	장치 ID가 100인 장치로부터 비트 레지스터 3999번 한 개를 읽어오라는 명령이지만 레지스터 3999는 비트 레지스터가 아니라 바이트 레지스터이기 때문에 존재하지 않는 레지스터를 읽는 상황이 됨			

응답패킷의 구조	Device ID	Function code	Error code
크기(byte)	1	1	1
패킷 내용	100	20	2
패킷 해설	답신으로 Function code에 20을 세팅하여 에러 패킷임을 나타내고 Error code는 2을 설정하여 존재하지 않는 레지스터를 나타냄		

3. 관제 소프트웨어 설계

- 가. 관제 소프트웨어는 N대의 원격전원관리를 통합으로 제어할 수 있도록 설계한다.
- 나. 각 장치들을 그룹으로 구분할 수 있도록 설계한다.
- 다. 각 장치들이 설치된 장소를 지도상에 표시할 수 있도록 한다.
- 라. 각 장치들을 등록할 때 장치의 별명을 부여하여 장치들을 식별할 수 있도록 한다.
- 다. 각 장치들의 각 콘센트의 전류값과 ON/OFF의 상태를 관찰할 수 있도록 하고 콘센트의 ON/OFF 상태를 변경할 수 있도록 한다.
- 라. 각 장치에 스케줄링을 입력하여 원하는 시간에 장치들이 ON/OFF될 수 있도록 한다.

- 마. 각 장치의 전류값의 한계치를 입력하고 모니터링하여 이상상황에서 알람을 발생하도록 한다.
- 바. 알람발생 상황은 알람창에서 종합적으로 관리할 수 있도록 한다.
- 사. 각 장치에 대한 고장이력을 입력할 수 있는 부분을 구현하여 각 장치에 대한 이력을 기록할 수 있도록 한다.
- 아. 관제 프로그램을 JAVA로 구현하여 OS에 관계없이 관제 프로그램을 실행할 수 있도록 한다.

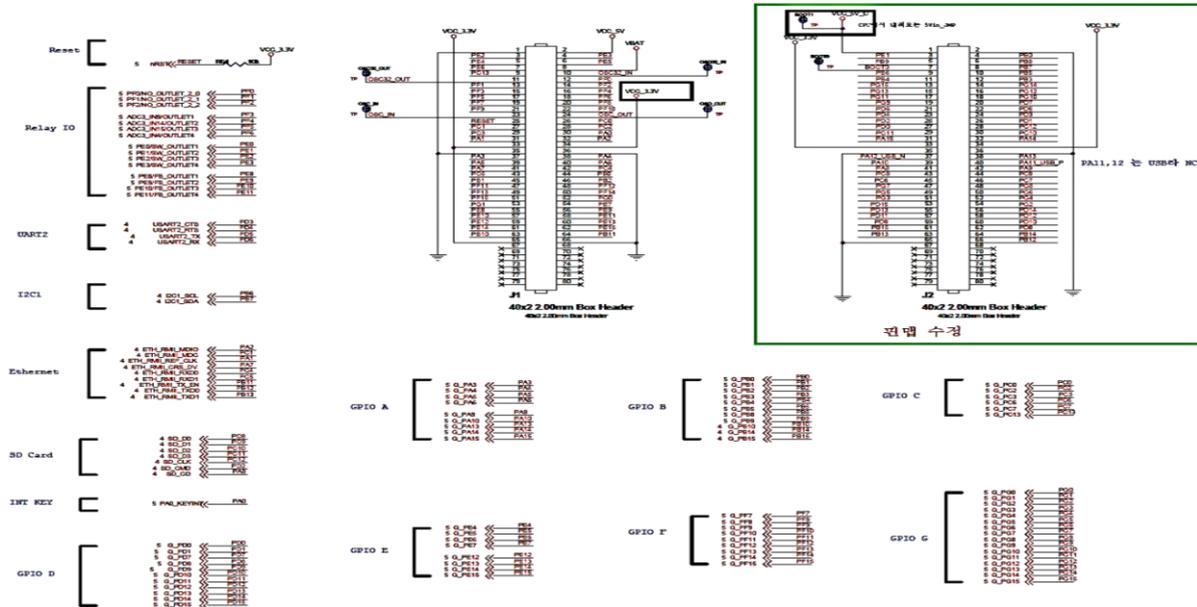
4. 스마트폰 앱 설계

- 가. 스마트폰 앱의 설계는 10대 이하의 원격전원제어장치를 제어할 수 있도록 설계한다.
- 나. 스마트폰 앱에서는 단순히 전원의 상태를 확인하고 전원을 ON/OFF할 수 있는 기능만을 구현한다.
- 다. 장치는 추가 / 삭제를 할 수 있도록 구현한다.
- 라. 장치는 탭뷰의 형식으로 장치와 장치를 자유롭게 전환할 수 있도록 한다.
- 마. 스마트폰의 통신부하를 줄이기 위하여 계속하여 장치의 상태를 가져오지 않고 “상태버튼”을 클릭할 때만 원격전원제어기의 최신 상태를 가져온다.
- 바. 등록된 장치는 스마트폰의 로컬 스토리지에 저장하여 앱을 다시 실행했을 때 등록된 장치가 그대로 남아 있도록 한다.
- 사. 장치의 설정은 한 개의 탭뷰 내에서 모두 조작할 수 있도록 스크롤 뷰를 적용한다.

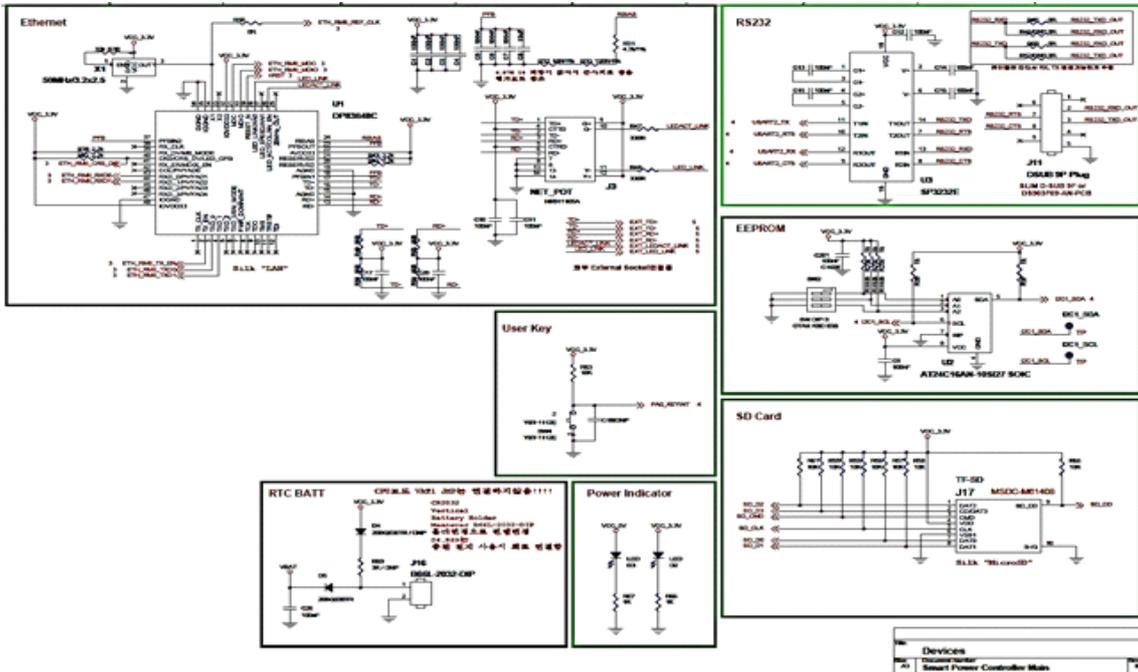
제 2 절 개발 결과

1. 하드웨어 개발

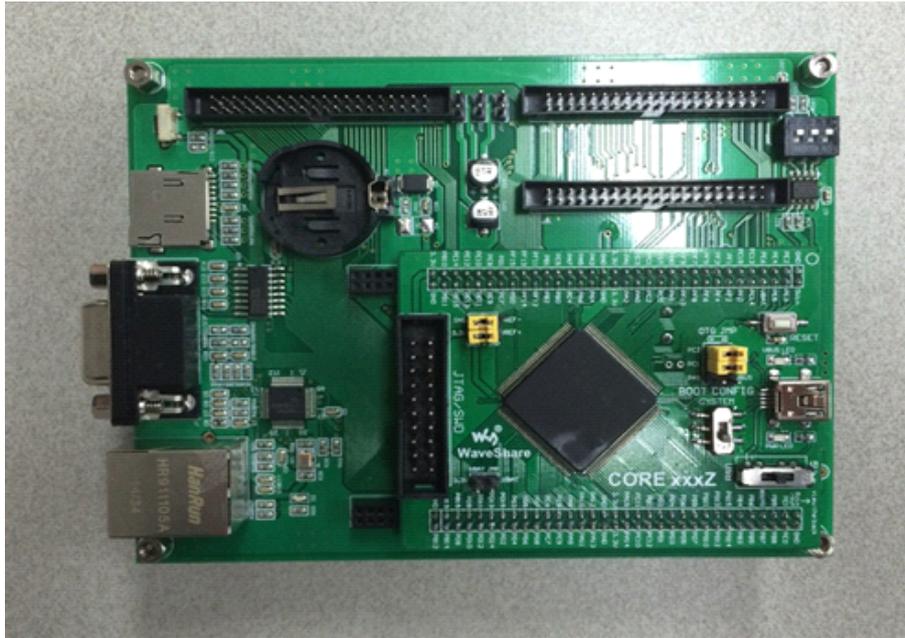
가. 베이스 보드 회로 설계 및 PCB 제작 완료



<그림 > 베이스보드 핀맵

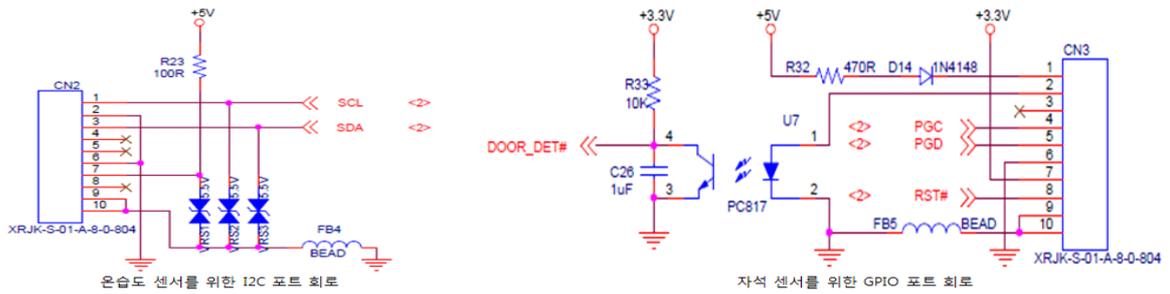


<그림> 베이스보드 주요 인터페이스 회로도

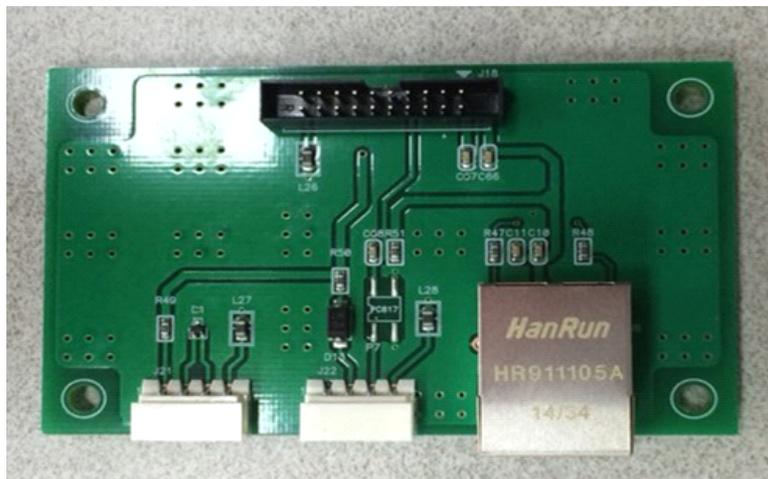


<그림 45> 베이스보드 결과물

나. 커넥터 보드 회로 설계 및 PCB 제작 완료

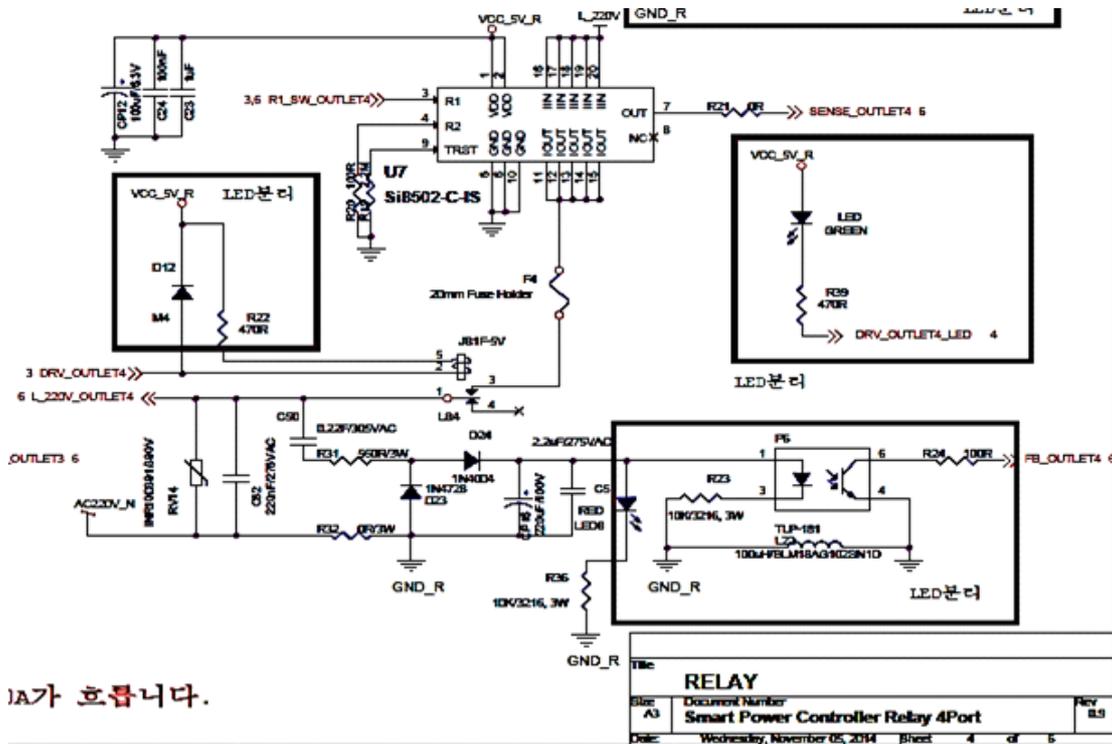


<그림> 커넥터보드 회로도



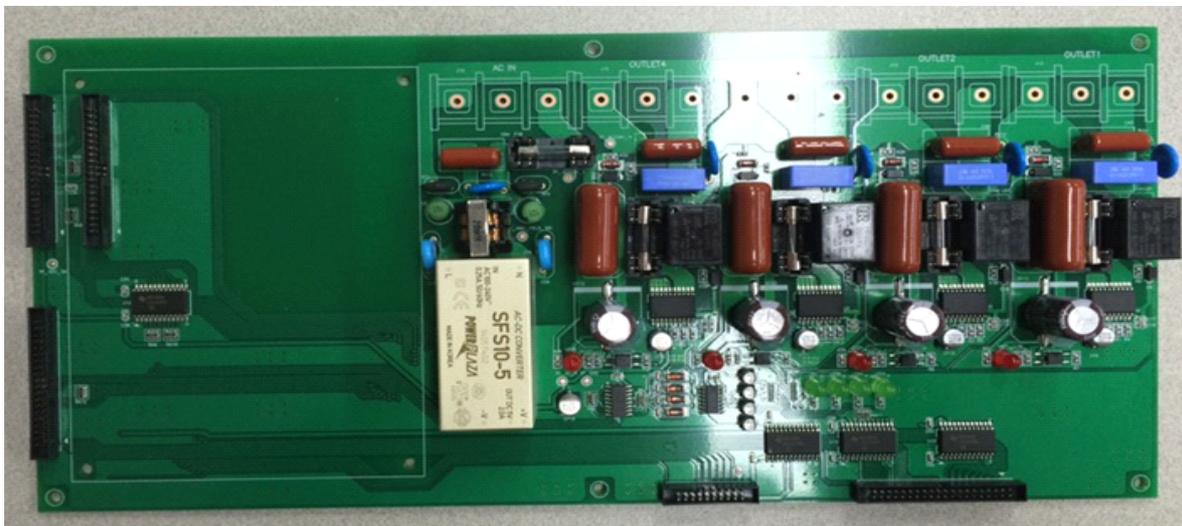
<그림> 커넥터보드 결과물

다. 릴레이 보드 회로 설계 및 PCB 제작 완료



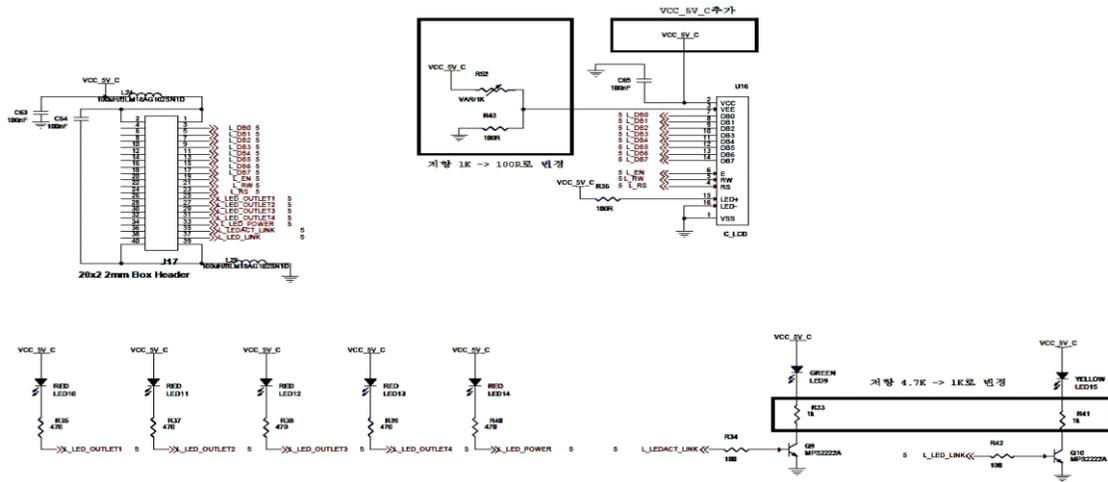
1A가 흐릅니다.

<그림> 릴레이보드 회로도

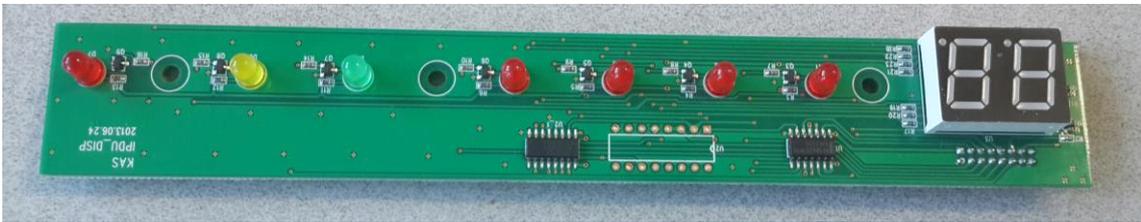


<그림> 릴레이보드 결과물

라. 디스플레이 보드 회로 설계 및 PCB 제작 완료

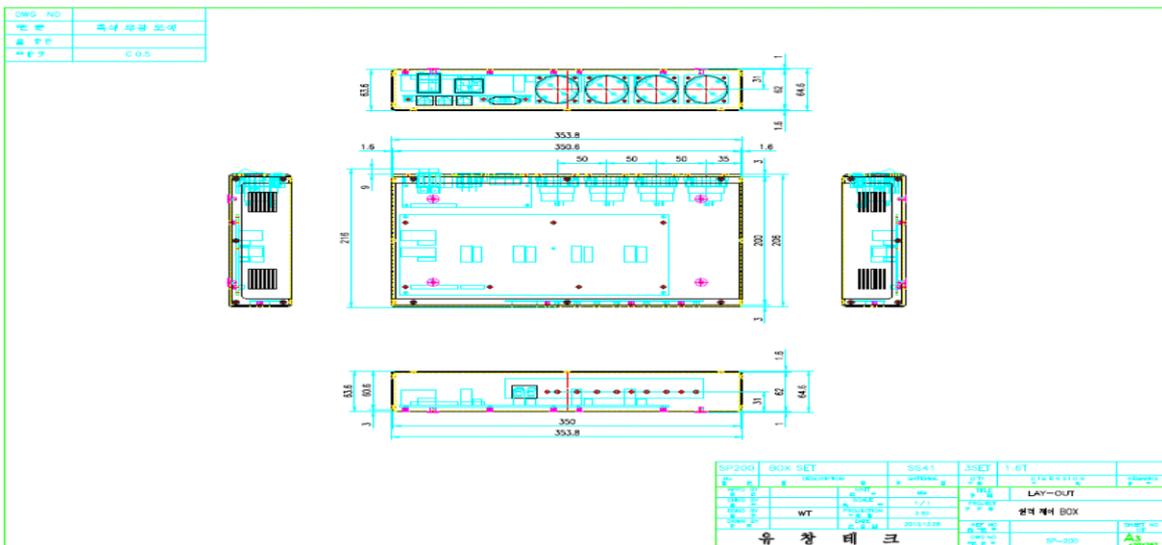


<그림> 디스플레이보드(LCD) 회로도



<그림> 디스플레이보드 결과물

마. 원격전원제어장치 케이스 및 시제품 제작 완료



<그림> 케이스 제작 도면



<그림> 원격전원제어기 시제품(전면, 후면)

2. 펌웨어 개발

가. 원격전원제어장치 펌웨어 구현 완료

(1) 원격전원장치 네트워크 정보 설정 및 전송 기능 구현

(가) 네트워크 상태, MAC 주소, 게이트웨이, IP, 포트정보, DHCP 사용여부를 외부로부터 입력받아 설정

(나) 네트워크 관련 정보를 클라이언트로 전송

(2) 타이머 설정 및 설정 상태 전송 기능 구현

(가) 타이머 설정정보를 외부로부터 받아와 설정(년, 월, 일, 시, 분)

(나) 타이머 설정 정보를 클라이언트로 전송

(3) 전류 값 전송 기능 구현

(가) 전체 4개의 아울렛에 대한 각 전류 값을 클라이언트로 전송

(4) 과전류 차단 오류 복구 기능 구현

(가) 한 아울렛이 10A이상일 경우 해당 포트 차단 및 복구

(나) 각 포트의 전압이 15A이상일 경우 전체 포트 차단 및 복구

(5) 패킷 암호화/복호화 기능 구현 완료

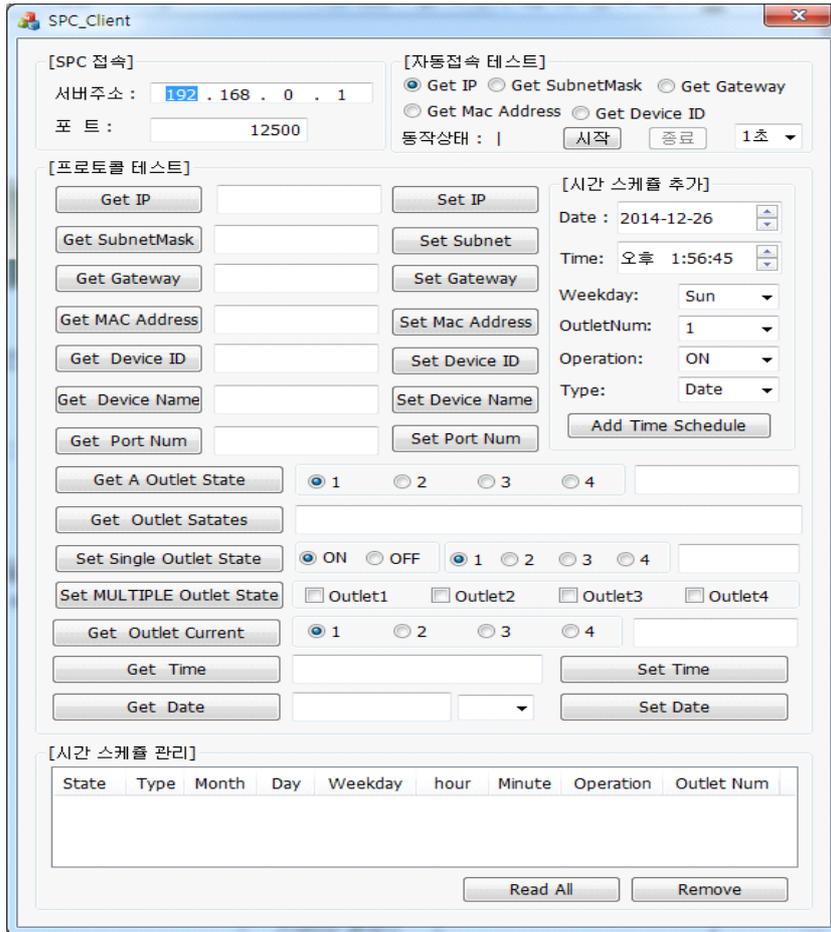
(가) 원격전원제어장치용 Modbus 프로토콜 내부에서 패킷 전송 시 내부 패킷 테이블 변환을 통한 패킷 암호화

(나) 클라이언트 수신 시 Modbus 프로토콜 내부에서 패킷 전송 시 내부 패킷 테이블 변환을 통한 패킷 복호화

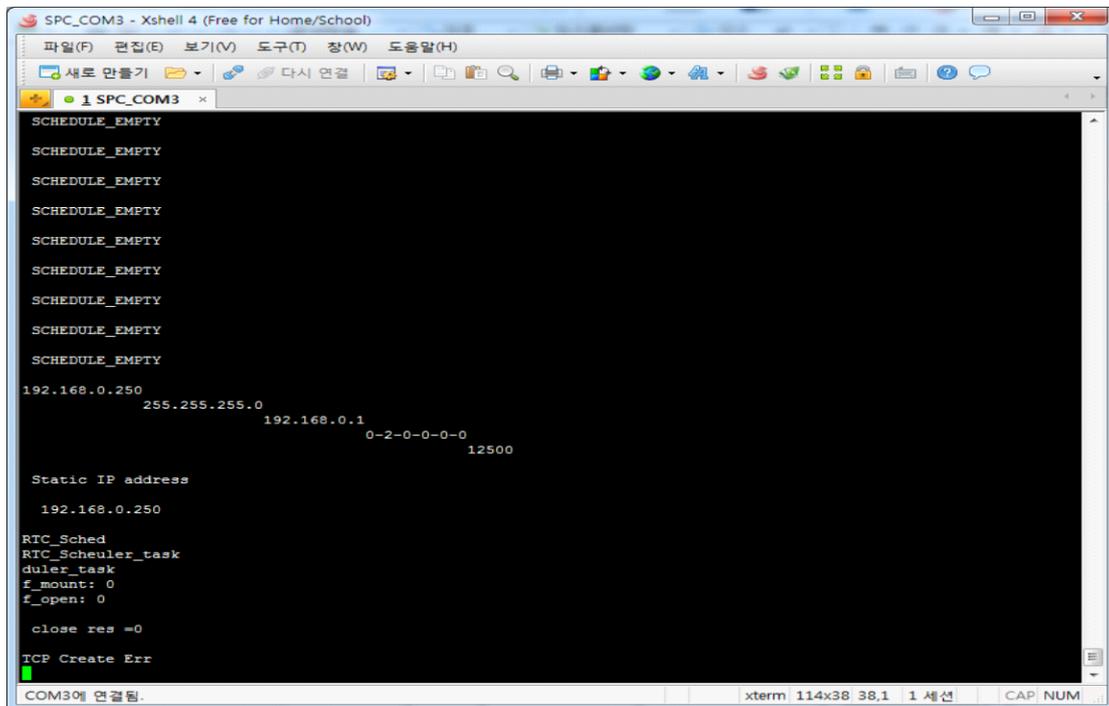
- (6) 네트워크 통신 장애 감지 및 복구 기능 구현
 - (가) TCP/IP 소켓 기반 내부 errno를 이용한 통신 장애 감지
 - (나) 해당 접속 종료 및 복구
- (7) 설정/동작 이력 저장 기능 구현
 - (가) 원격전원제어장치의 내부 설정을 EEPROM에 저장 하여 전원 OFF시에도 데이터 유지
 - (나) 클라이언트 접속 및 동작 상황을 실시간으로 감지하여 SD카드에 이력 저장
- (8) 원격전원제어장치 초기화 기능 구현
 - (가) 초기화 키 입력을 통하여 장치의 내부 설정을 디폴트 값으로 변경
 - (나) 초기 데이터는 EEPROM 내부에 기록

나. 원격전원제어장치 펌웨어 기능 검증

- (1) 원격전원제어장치의 기능 구현은 시제품 제작 용역을 통해 개발을 진행하였으며, 각 기능을 자체적으로 점검하기 위한 기능검증용 클라이언트 프로그램을 개발함.
- (2) 기능검증용 클라이언트 프로그램에서는 원격전원제어장치의 네트워크 정보, 포트 정보, 장치 시간, 아울렛 제어 등의 기본적인 동작 테스트를 검증할 수 있음.
- (3) 프로그램과 장치간의 모은 데이터 전송 및 수신은 앞서 설명한 원격전원제어장치용 MODBUS Protocol을 기반으로 구현함.
- (4) 기능 검증확인용 RS-232를 이용한 시리얼 통신으로 함.



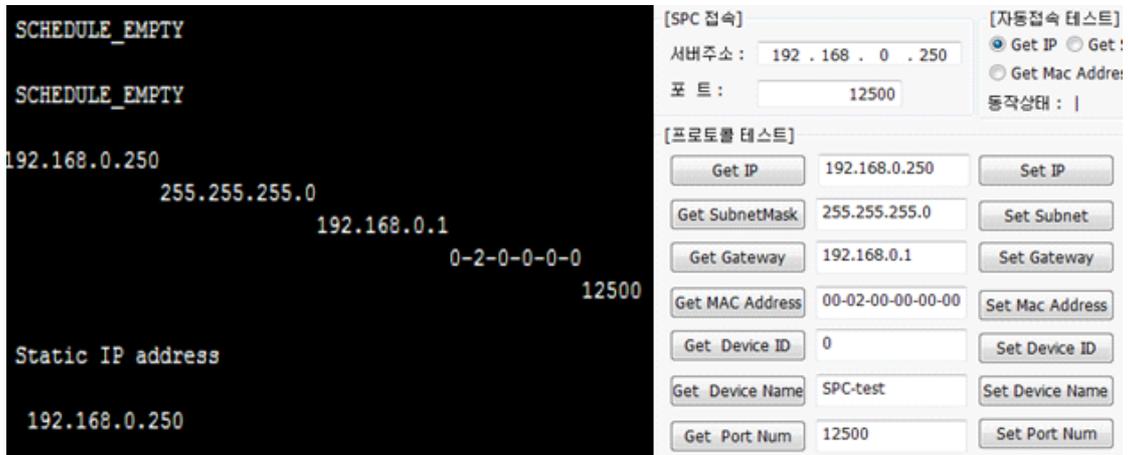
<그림> 원격전원제어기 펌웨어 기능검증 프로그램



<그림> 펌웨어 기능동작 검증을 위한 시리얼 통신 프로그램

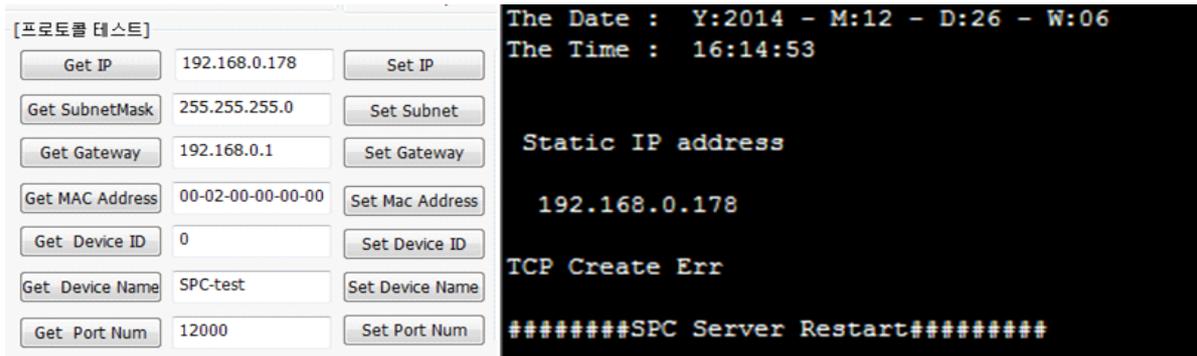
다. 원격전원제어장치 네트워크 정보 설정 및 전송 기능 검증

- (1) 장치를 리셋하여 시리얼 프로그램에서 현재 장치의 네트워크 정보 상태를 확인하고 검증 프로그램 상에서 정상적으로 값을 받아오는 것을 확인함.



<그림> 원격전원장치의 네트워크 정보 수신 테스트

- (2) 검증프로그램 상에서 원격전원제어장치에 연결하여 네트워크 정보를 변경하고 반영이 되는지를 확인함.

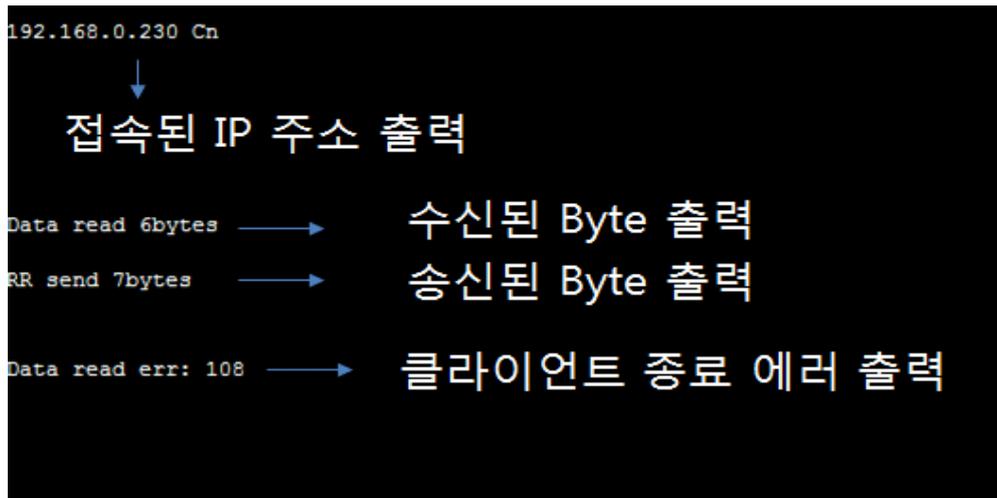


<그림 30> 원격전원제어장치의 네트워크 정보 전송 테스트

- (3) TCP/IP 프로토콜통신기반 통신을 위해 MAC 주소와 서브넷은 임의로 변경하는 것을 금지해야하기 때문에 내부 테스트를 위한 용도로 구현됨.

라. 원격전원제어장치 전송 및 수신 패킷 검증

- (1) 기능검증 프로그램에서 장치의 현재 IP주소를 요청할 경우 위에서 설계한 프로토콜에 따라 다음과 같은 결과를 나타냄.

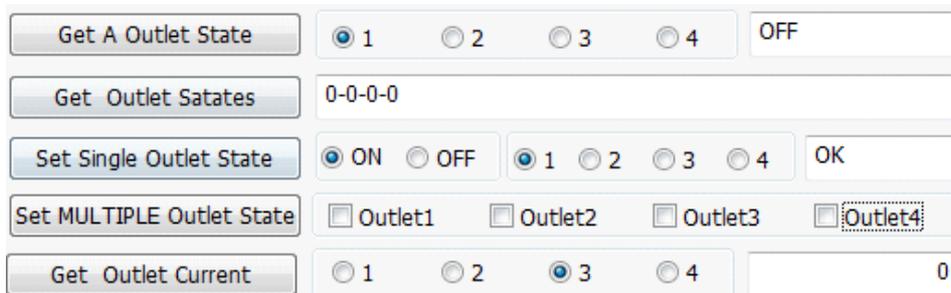


<그림> 원격전원제어장치의 패킷 검증

- (2) 기능검증 프로그램을 실행하는 PC의 아이피 주소는 192.168.0.230이며 단일/복수 32비트 레지스터를 읽기 위한 요청 프로토콜에 따라 원격전원제어 장치는 6byte를 수신함.
- (3) 해당 장치는 단일/복수 32비트 레지스터를 읽기 위한 요청과 응답 프로토콜에 따라 수신된 6byte에서 레지스터 정보와 요청 코드를 분석하여 현재 장치의 IP 정보를 포함한 7byte를 기능 검증 프로그램에 전송함.
- (4) 해당 장치의 오류 동작체크 기능 확인을 위해 프로그램 상에서 고의적으로 비정상적인 종료를 하였으며, 이에 해당하는 클라이언트 종료 에러 출력을 출력함.

마. 원격전원제어장치 아울렛 포트 상태 확인 검증

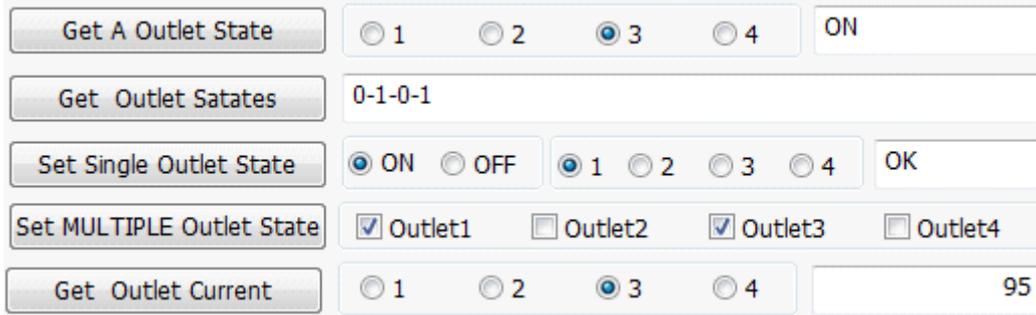
- (1) 기능 검증 프로그램에서는 각 아울렛 포트의 상태 및 제어가 가능함.
- (2) 단일 비트 레지스터 및 복수 비트 레지스터 프로토콜을 이용하여 프로토콜 검증 및 아울렛 동작 상태에 대한 기능을 검증함.



<그림> 원격전원제어장치의 아울렛 포트 상태 확인 및 제어

- (3) Get A Outlet State는 각 포트의 상태를 나타내며, 포트가 켜져 있을 시 ON, 꺼져 있을 시 OFF로 나타난다. 단일 비트 레지스터 프로토콜 검증을 위하여 각각의 포트 상태값을 따로 표시한다.

- (4) Get A OutletStates는 ON일 경우 1, OFF일 경우 0으로 나타나며, 복수 비트 레지스터 프로토콜을 검증을 위하여 4개의 아울렛 포트 값을 한번에 표시한다.(표시는 4, 3, 2, 1 순임)
- (5) Set Single Outlet State는 단일 비트 레지스터 프로토콜을 이용하여 각 포트 상태 값을 따로 제어 함.
- (6) Set Single Outlet States의 경우 복수 비트 레지스터 프로토콜을 이용하여 각 포트 상태 값을 한 번에 제어 함.
- (7) Get Outlet Current는 현재 포트에 연결되어 있는 장비의 전력 값을 표시함.

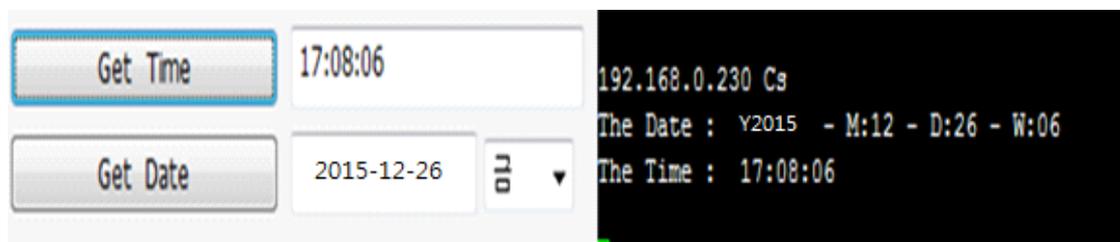


<그림> 원격전원제어장치의 아울렛 포트 상태 확인 및 제어 결과

- (8) 현재 장비의 아울렛은 1번과 3번이 켜진 상태이며, 3번 포트의 전력 값은95mA 임을 나타냄.

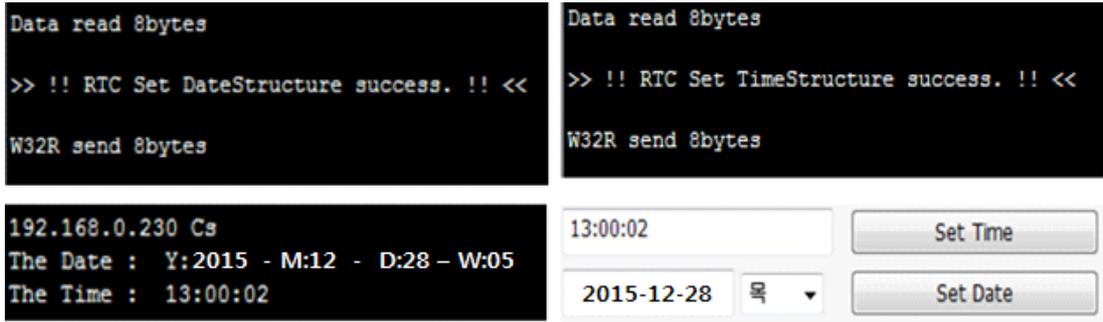
바. 원격전원제어장치 시간 정보 수신 및 설정

- (1) 원격전원장치는 자체적인 아울렛 동작제어를 위한 타이머 기능(스케줄링)이 구현되어 있음.
- (2) 타이머 기능은 기본적으로 현재 시간을 기준으로 설정이 되기 때문에 장치의 현재 시간 수신 및 설정 동작을 우선으로 테스트 함.
- (3) 기능 검증프로그램에서 데이터를 요청할 경우, 장비 내부에서는 요청받은 데이터를 처리 한 후에 시간 정보를 이력으로 저장함.
- (4) 이를 응용하여, 현재 시간 정보를 요청하고 응답할 때의 메시지를 시리얼 프로그램으로 확인함.



<그림> 원격전원제어장치의 시간 정보 수신

- (5) 시리얼 프로그램에서 나타난 메시지는 검증 프로그램의 IP 접속 종료(Close)와 날짜, 시간 정보를 출력함.

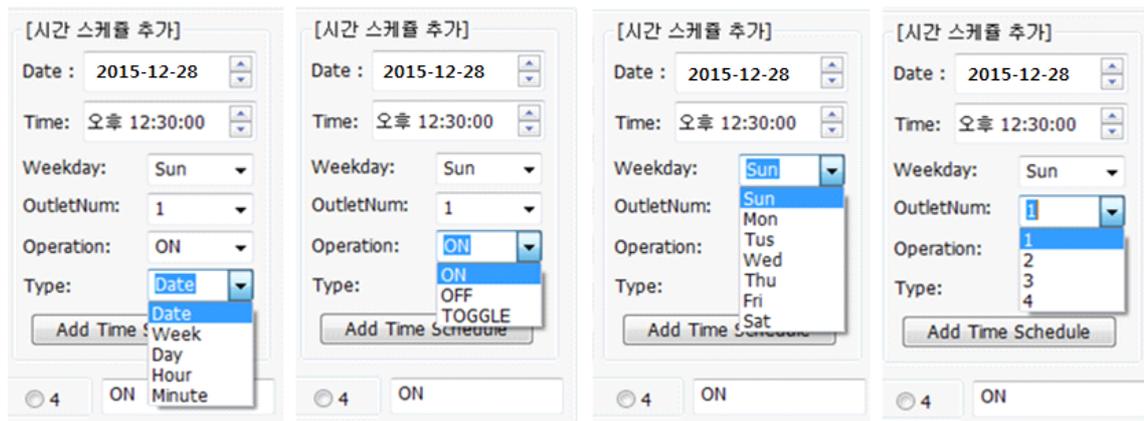


<그림> 원격전원제어장치의 시간 정보 입력

- (6) 시간 설정은 날짜(Date)정보와 시간(Time)정보로 각각 정의 하여 구분되어 있음.
- (7) 원격전원제어장치 프로토콜에 따라 각 패킷은 8byte 크기로 요청 및 응답하는 것을 확인하였으며, 변경된 날짜 및 시간 정보 출력을 확인함.

사. 원격전원제어장치 시간 스케줄 기능 검증

- (1) 원격전원제어장치의 각 포트는 스케줄링 기능에 의하여 예약된 날짜, 시간에 따라 On, Off, Toggle 기능을 수행 할 수 있도록 구현됨.
- (2) 스케줄링 기능은 현재 30개까지의 예약을 할 수 있도록 구현 되어 있으며, 추후 확장 할 수 있도록 설계됨.
- (3) 장치의 전원이 Off된 상태에서도 스케줄링에 관련된 기록(스케줄링 아이템)은 남아 있어야 하기 때문에 물리적으로 저장 할 수 있는 EEPROM에 기록 되도록 구현되어 있음.
- (4) EEPROM에 기록되어 있는 스케줄링 아이템 상태를 확인하기 위하여 원격전원 제어장치의 부팅 시 시리얼 프로그램에 표시되도록 하여 상태를 확인함.



<그림> 원격전원제어장치 스케줄링 아이템 설정

(5) 시간 스케줄 아이템의 설정

- (가) 특정날짜 타이머 : 월/일/시/분에 동작하는 스케줄. 요일 값은 무시됨.
- (나) 주간 : 특정 요일/시/분마다 반복하는 스케줄. 월/일 설정 값은 무시됨.
- (다) 일간 : 특정 시/분마다 반복하는 스케줄. 월/일/요일 값은 무시됨.

- (라) 시간 : 특정 분마다 반복하는 스케줄. 월/일/요일/시각 값은 무시됨.
- (마) 매 분간 : 특정분의 배수분마다 반복하는 스케줄. 월/일/요일/시각 값은 무시됨.

(6) 시간 스케줄의 동작 설정

- (가) On : 해당 포트의 전원을 켜.
- (나) Off : 해당 포트의 전원을 끄.
- (다) Toggle : 해당 포트의 전원이 켜져 있으면 끄고, 꺼져 있으면 켜.

(7) 프로그램 상에서 스케줄 등록상태는 다음과 같이 나타남.

[시간 스케줄 관리]

State	Type	Month	Day	Weekday	hour	Minute	Operation	Outlet Num
2	1	1	1	1	12	30	1	2
2	1	1	1	1	12	30	1	1
2	1	1	1	1	12	30	1	0

<그림> 기능 검증 프로그램 상에서 등록된 스케줄 아이템 상태 출력

(8) 장치를 재부팅 하였을 때 상태는 다음과 같이 나타남.

```

No need to configure RTC...
The Date : Y:2015 - M:12 - D:28 - W:05
The Time : 11:17:15

The current alarm is : 00:00:00
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY

No need to configure RTC...
The Date : 2015 - M:12 - D:28 - W:05
The Time : 11:37:46

The current alarm is : 00:00:00
SCHEDULE_RECORDED
SCHEDULE_RECORDED
SCHEDULE_RECORDED
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY
SCHEDULE_EMPTY

```

<그림> 스케줄 아이템 등록 상태 출력(등록 전 - 왼쪽, 등록 후 - 오른쪽)

(9) 시리얼 상에서 볼 때 스케줄 아이템이 등록하지 않고, 장치를 재부팅 하면, 초기에 저장된 값이 비어 있는 것을 확인 함.

(10) 스케줄 아이템 등록 후, 장치를 하여 시리얼 프로그램을 확인한 결과, 기능 검증 프로그램에서 설정한 3개의 스케줄 아이템이 등록된 것을 확인 함.

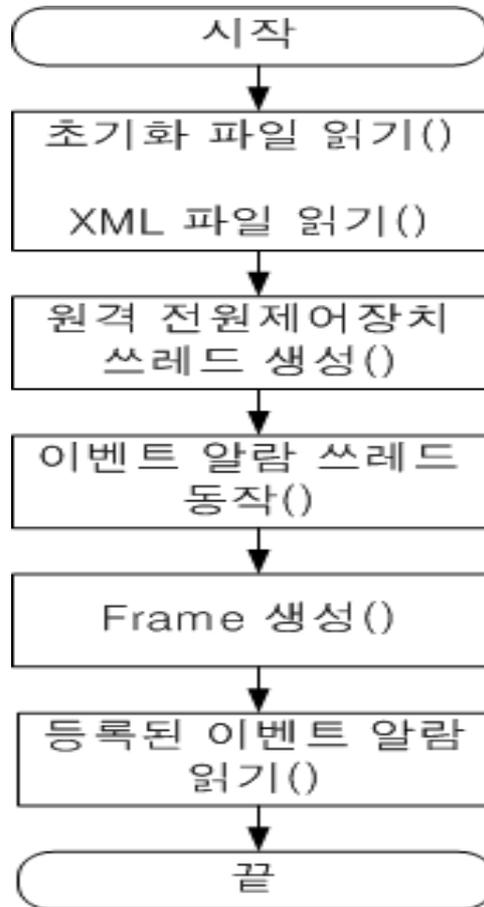
아. 원격전원제어장치 이력 저장 기능 검증

(1) 원격전원제어장치는 내부적으로 장치의 상태, 클라이언트 접속, 송·수신된 데이터 정보 이력 등을 MicroSD Card에 저장하는 기능이 구현되어 있음.

(2) 장치의 내부 이력은 매 시간마다 월-일-시간 순서의 파일 명으로 생성이 되며, 파일 내부에는, 제어 이벤트가 발생한 시간, 분, 초 마다 클라이언트의 IP 주소, 내부 동작상태, 송수신된 데이터 등의 이력이 자동으로 저장되는 것을 확인함.

3. 관제 소프트웨어 개발

가. 통합관제 프로그램 흐름도



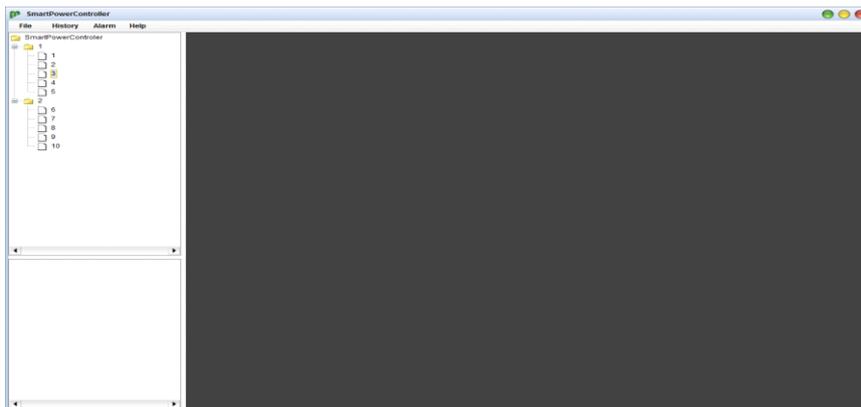
<그림> 프로그램 전체 흐름도

나. 통합관계 프로그램 개발 환경

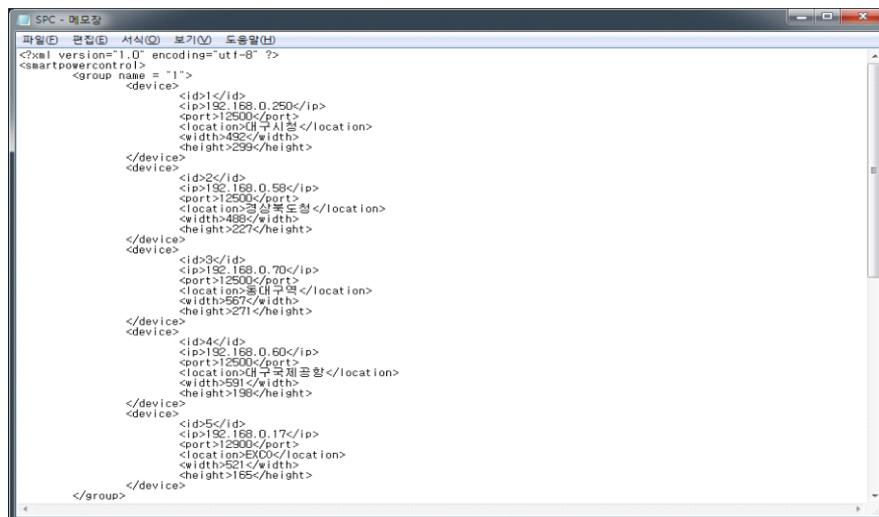
통합관계 시스템 개발 환경	
항 목	세 부 사 양
하드웨어 시스템	Intel I7기반 전용PC
개발언어	JAVA, XML
자바버전	java 1.8.0_05
개발툴	eclipse Luna release 4.4.1

다. 원격전원제어장치 관리 기능 구현

- (1) 장치의 최초등록은 XML파일을 이용하여 등록하고, 프로그램 구동 시 지정된 XML파일에서 장치의 그룹과 ID, IP, PORT, LOCATION의 정보를 가져옴.
- (2) XML파일의 태그는 루트, Group, id, ip, port, location, width, height로 구성됨.

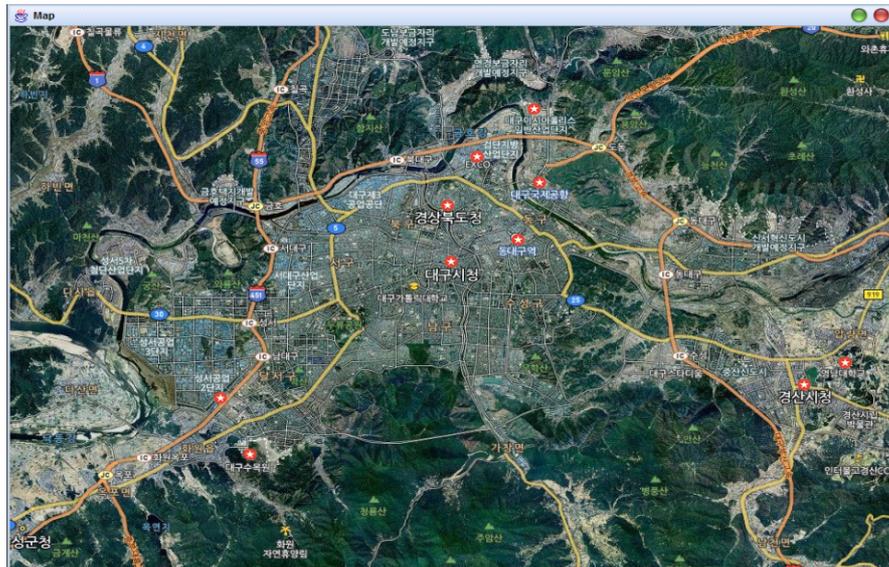


<그림 40> 장치관리 기능화면



<그림 41> XML 파일 예시

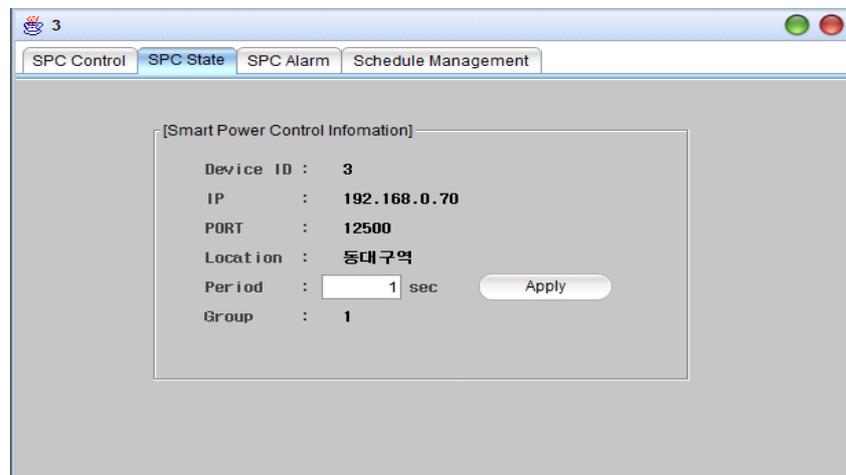
- (3) 해당 장치의 설치 장소를 지도상에 표시할 수 있으며, 표시를 클릭하여 해당 장치를 관리할 수 있음.



<그림 42>. 지도상에 설치된 장치의 모습(가정된 시나리오)

라. 원격전원제어장치 정보 표시기능 구현

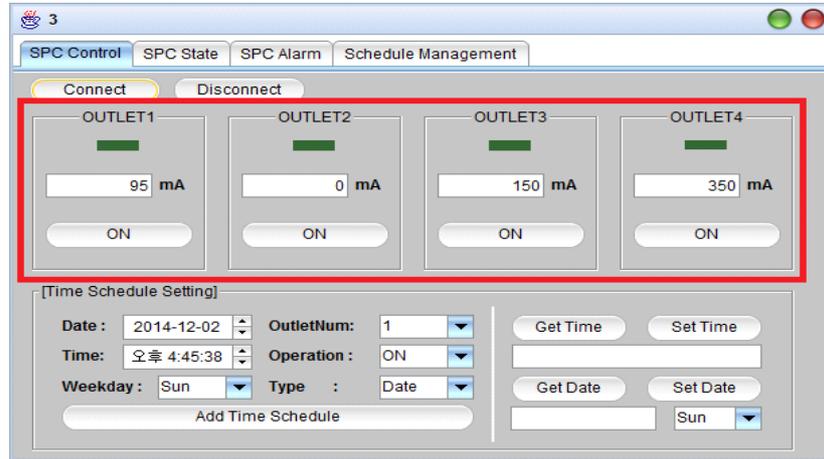
- (1) Device ID : 원격 전원제어장치의 ID
- (2) IP : 원격 전원제어장치의 IP
- (3) PORT : 원격 전원제어장치의 연결 PORT
- (4) LOCATION : 원격 전원제어장치의 설치 장소
- (5) Period : 원격 전원제어장치와 통합관제 프로그램과의 통신 주기
- (6) Group : 원격 전원제어장치의 소속 그룹



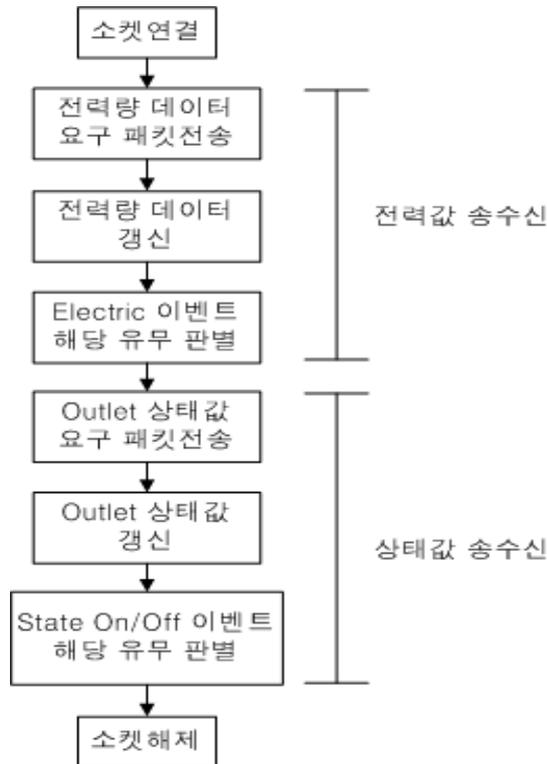
<그림 43> 장치정보 표시 기능화면

마. 원격전원제어장치 아울렛제어 기능 구현

- (1) 원격 전원제어장치 Outlet의 현재 상태값(ON/OFF) 확인 기능
- (2) 원격 전원제어장치 Outlet의 전력값 실시간 모니터링 기능
- (3) 원격 전원제어장치 Outlet의 상태제어 기능



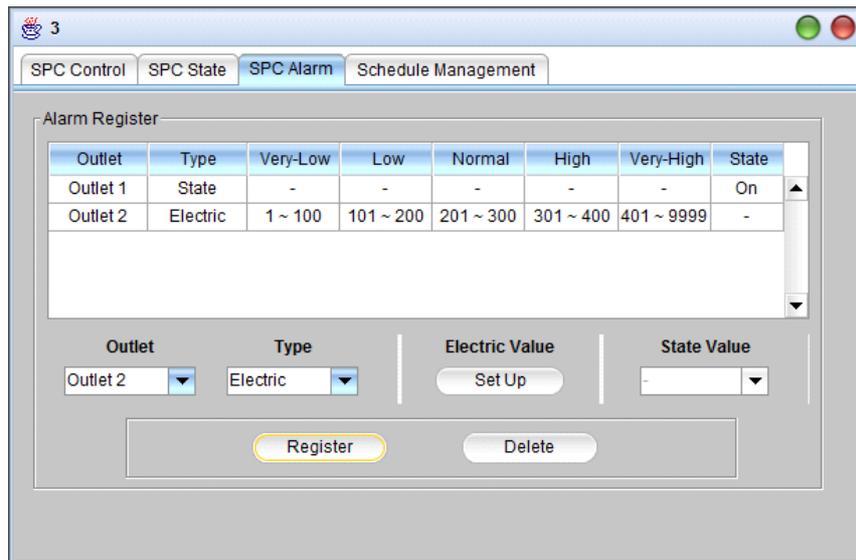
<그림> 장치전원 제어 기능화면



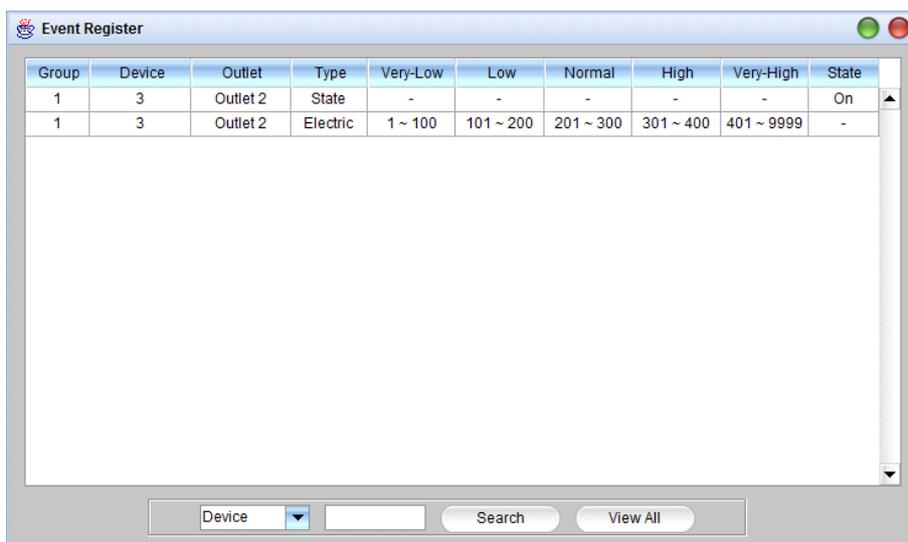
<그림> 상태 업데이트 통신의 흐름도

바. 이벤트 보고/알람 조치 기능 구현

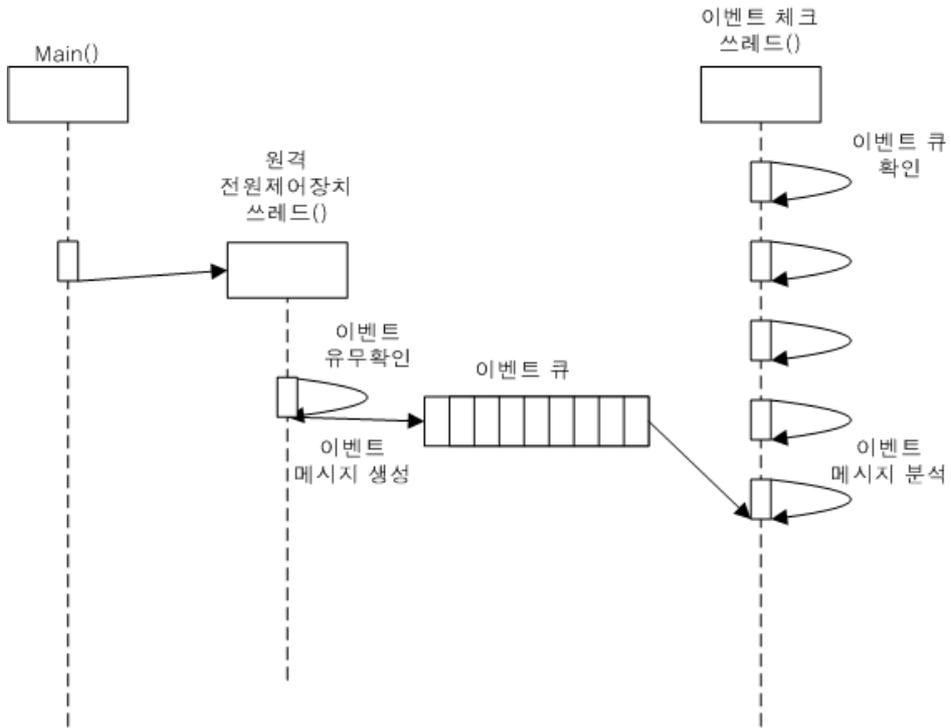
- (1) 이벤트 등록은 각 Outlet별로 Electric, State On, State Off 타입으로 등록가능하며, Electric의 경우 LowLow, Low, Normal, High, HighHigh의 5단계로 나뉨.
- (2) 이벤트 등록 현황 조회는 기 등록된 원격 전원제어장치에서 등록된 이벤트를 통합하여 볼 수 있음.
- (3) 해당 이벤트가 발생하면 알림음과 어떤 알림인지 알려주는 창이 함께 뜨고 해당 이벤트가 다시 정상상태로 돌아가거나 Ack버튼을 눌러 해제할 수 있음.



<그림> 이벤트 등록 화면



<그림> 이벤트 등록 현황 통합 조회화면



<그림> 이벤트 생성 및 소비 시퀀스

사. 고장/조치 이력 저장 기능 구현

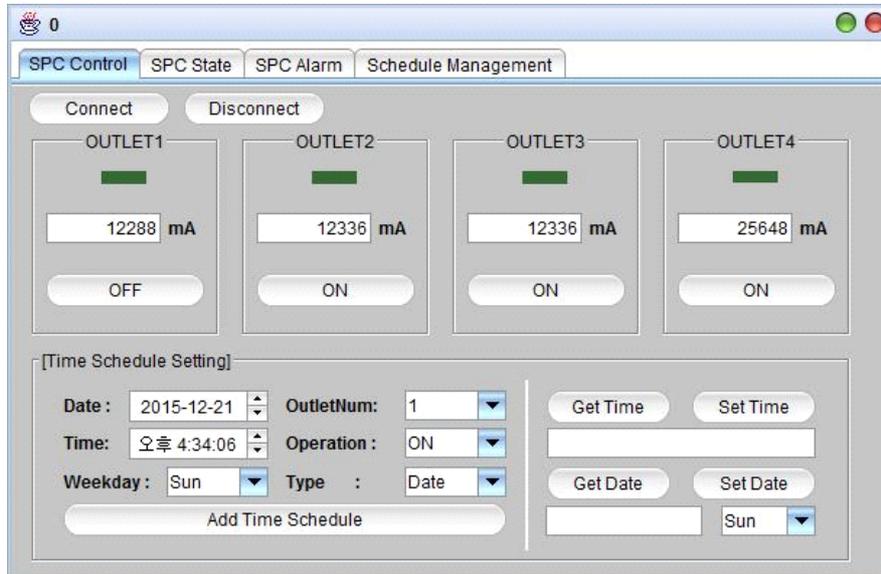
- (1) 고장/조치 이력 저장은 원격 전원제어장치의 ID, 고장/조치날짜, 작성자, 고장증상, 조치내용 항목으로 구성되어 있음.
- (2) 고장/조치 조회는 저장된 모든 고장/조치 이력을 볼 수 있으며, 원격 전원제어장치의 ID 또는 작성자로 검색할 수 있음.



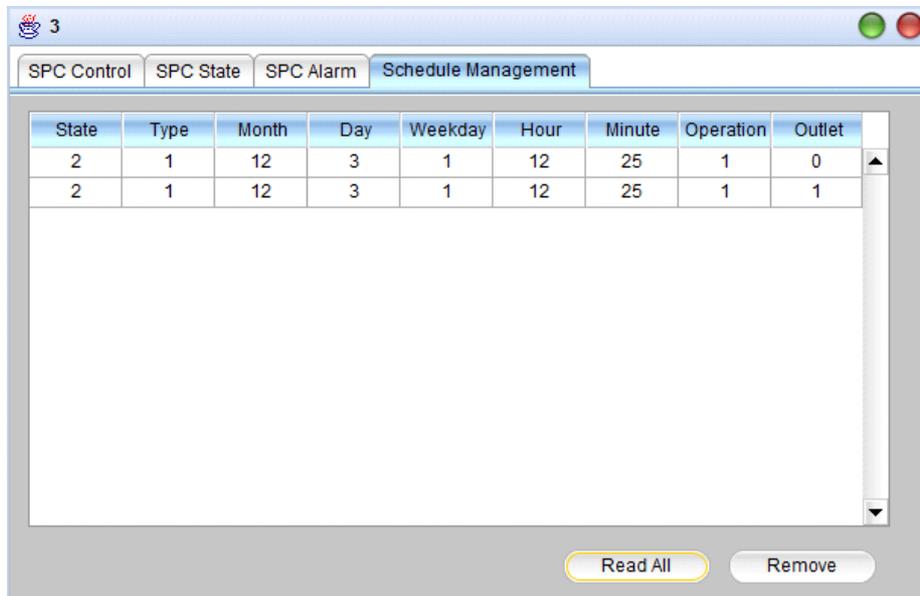
<그림> 고장/조치 저장 화면

아. 스케줄링 기능 구현

- (1) 스케줄링 등록은 날짜, 시간, 요일, Outlet, 작동방법, 타입 등의 항목이 있음.
- (2) 시간동기화는 통합 관제 프로그램에서 세팅한 스케줄링 시간에 원격 전원제어 장치가 정확하게 동작하기 위한 것으로, SetTime 버튼으로 시간을 설정하고 GetTime으로 현재 원격 전원제어장치의 시간을 확인할 수 있음.



<그림> 스케줄링 등록 및 원격 전원제어장치 시간, 날짜 동기화 화면



<그림> 스케줄링 등록 현황 조회 화면

자. 통합관제 프로그램의 최대 관리 수 평가를 위한 에뮬레이터 개발

- (1) 통합 관제 프로그램은 최대 100대의 원격 전원제어장치를 동시에 관리 가능하도록 개발하는 것이 목적임.

- (2) 본 사업의 예산으로는 원격 전원제어장치를 30대 정도 제작이 가능하나 100대 까지 관리 가능한지 테스트를 위해서 추가적으로 70대가 더 필요함.
- (3) 이를 위해 원격 전원제어장치와 동일한 기능을 수행하는 에뮬레이터를 개발하였음.



<그림> 원격 전원제어장치 에뮬레이터 및 클라이언트

원격 전원제어장치 에뮬레이터	
항 목	세 부 사 양
하드웨어 시스템	Intel I7기반 전용PC
개발언어	MFC C++
개발툴	Visual Studio 2010

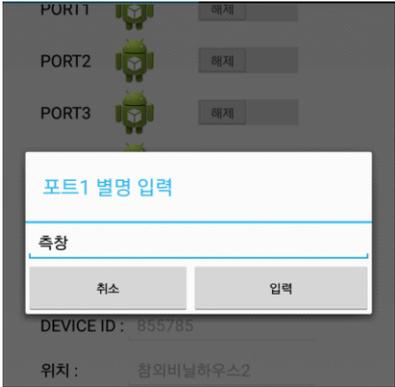
4. 스마트폰 앱 개발

가. 앱 개발 환경

구 성 요 소	내 용
개발 스테이션	Windows 7 64bit
JDK	8u25
안드로이드 SDK	
IDE	eclipse

나. 주요기능

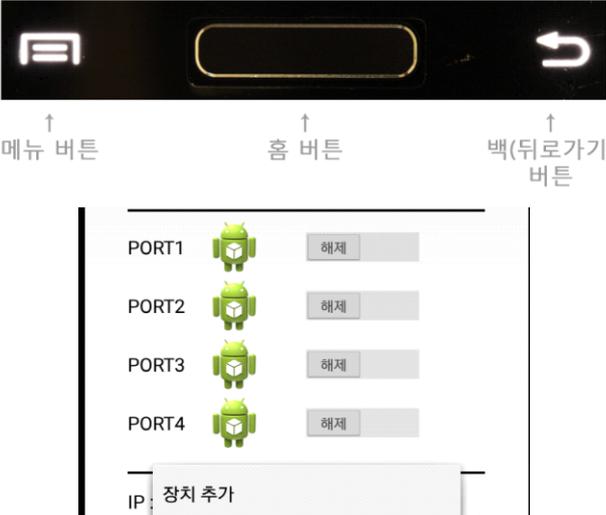
주요기능	설명
장치등록	<p>장치를 앱에 등록하기 위하여 IP, PORT, DEVICE ID, 위치를 입력한다. 앱의 하단의 “정보변경” 버튼을 클릭하면 버튼 상단의 IP, PORT, DEVICE ID, 위치의 에디트 박스가 활성화되며 각 에디트 박스를 클릭하여 각각의 정보를 입력한다. IP, PORT번호 DEVICE ID는 원격전원제어장치에 설정된 정확한 정보를 설정하여야 원격전원제어장치와 연동이 된다. 위치는 장치가 설치된 위치를 구분하기 위하여 입력하는 부가 정보이다.</p> 
상태갱신	<p>앱의 상단에 위치한 “상태갱신” 버튼은 원격전원제어장치의 4포트의 ON/OFF 상태를 가져온다. 원격전원제어장치와 성공적으로 PORT의 슬라이더 버튼의 상태를 자동으로 원격전원제어장치의 상태와 일치시킨다. 특정 포트의 상태를 ON/OFF 시키려면 슬라이더 버튼을 ON이나 OFF쪽으로 밀면 원격전원제어기의 상태를 변경할 수 있다.</p> 
포트명 변경	<p>원격전원제어기는 4개의 콘센트가 있는데 4개의 콘센트에 대한 별명을 부여하여 어떤 장치가 어떤 콘센트에 연결되어 있는지를 쉽게 알 수 있도록 한다.</p>



앱의 중앙부분에 “PORT1”, “PORT2”, “PORT3”, “PORT4”의 텍스트를 클릭하면 각 콘센트에 대한 별명을 입력할 수 있는 다이얼로그가 출력되고 이 다이얼로그의 텍스트 에디트 창에 포트의 별명을 입력하고 “입력” 버튼을 누르면 해당 콘센트의 별명이 변경된다.

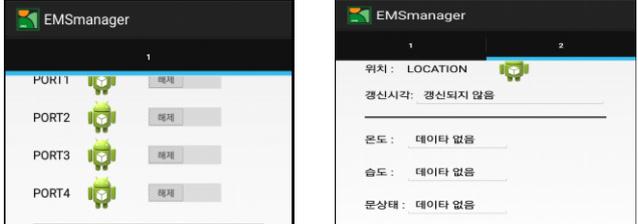
전원 ON/OFF

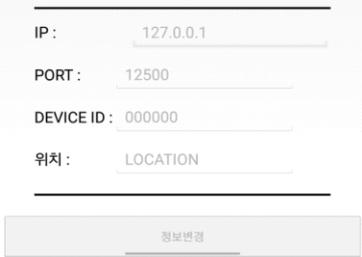
스마트폰의 “메뉴” 버튼을 클릭하면 “장치추가” 메뉴가 출력됨



장치 추가

“장치추가” 메뉴를 클릭하면 앱의 상단에 탭이 추가됨



<p style="text-align: center;">장치 삭제</p>	<p>지우고자 하는 장치의 탭을 선택한 후 탭의 제일 하단에 “장치삭제” 체크박스를 체크한다. 앱을 종료한 후 다시 시작하면 해당 장치가 삭제된다.</p> <div style="text-align: center;">  </div> <p style="text-align: center;"><input checked="" type="checkbox"/> 장치삭제</p>
--	--

제 4 장 목표달성도 및 관련분야에의 기여도

제 1 절 연구개발목표의 달성도

1. 연구개발 목표의 달성도

연구개발 목표	달성도	달성근거
1. 원격전원제어 시스템을 이용한 하우스 및 농업시설물의 자동개폐장치 전원관리	98%	원격에서 전원을 ON/OFF 함으로써 자동개폐장치를 제어할 수 있는 장치를 개발하였음. 개폐장치의 종류에 따라 제어 패널의 구성이 달라지기는 하나 공통적이고 핵심적인 기능인 원격지에서 전원의 제어로 여러 가지 자동개폐장치의 전원관리가 가능함
2. 전원 관련 상태 정보 및 에러 발생 시 상황 모니터링	97%	전원포트에서 허용하는 전원의 상한 값과 하한 값을 설정할 수 있도록 하였으며 전류 값이 지정된 상한 값보다 크거나 하한 값보다 작을 경우 알람을 발생하는 기능을 구현하여 전원에 이상이 있는 상황을 빠르고 정확하게 파악할 수 있도록 하였음. 다만 매우 0.5A미만의 약한 전류에 대한 전류센서의 분해능이 다소 부족하나 일반적인 하우스 및 농업시설물에서는 대부분 1A이상의 큰 전류를 소모하는 기계장치들이 많으므로 큰 문제가 되지 않을 것으로 판단됨.
3. 자체 시스템 보호 기능 (안전장치)	100%	전원포트 당 10A까지의 전류를 견딜 수 있도록 설계하였으며 10A 이상의 전류가 통전될시 펌웨어단에서 회로를 차단하는 기능을 구현하였으며 전원 포트 당 10A 퓨즈를 부착하여 과전류로부터 발생하는 문제점을 원천적으로 차단하였음. 10A 이상의 전류를 소모하는 기기에 대하여는 마그넷 스위치를 2차적으로 부착하여 제어가 가능하도록 권고사항을 마련하였음.
4. 단순하고 효율적인 UI를 가진 App 개발	100%	관계 소프트웨어가 가지고 있는 복잡한 기능을 모두 구현하기 보다는 핵심기능인 전원제어를 구현하였으며 1:N 제어를 구현하기 위하여 탭뷰 형식의 앱을 제작하였으며 사용이 직관적이도록 한 장치의 설정을 한 탭 화면에 포함하도록 스크롤 뷰 형식을 도입하였음.

제 2 절 관련분야의 기술발전예의 기여도

1. 원격 전원 제어 시스템을 이용한 하우스 및 농업시설물의 자동개폐장치 전원 관리
 - 가. 다양한 하우스 농업시설물의 자동개폐장치에 공통적으로 사용할 수 있는 전원관리 장치를 개발하였음.
 - 나. 본 과제에서 개발한 전원제어장치는 4대의 장치의 전원을 제어할 수 있도록 설계하였으며 전원을 관리하여야 하는 장치가 늘어나면 전원제어장치를 여러 대 설치하여야 하지만 하나의 소프트웨어에서 다수의 전원제어장치를 제어할 수 있도록 1대N 제어가 가능한 관계 프로그램을 구현하였음.
 - 다. 지역적으로 분리된 비닐하우스 여러 동이더라도 하나의 관계 소프트웨어에서 통합으로 관리할 수 있도록 하였으며 이동 중 혹은 외출 중이더라도 시설물의 전원관리가 가능하도록 스마트폰 앱을 개발하였음.
 - 라. 포트별로 타이머 기능을 구현하여 정해진 시간에 자동으로 시설을 제어할 수 있는 편리성을 구현하였음.
2. 전원 관련 상태 정보 및 에러 발생 시 상황 모니터링
 - 가. 최초로 전류 값에 대한 알람 시스템을 도입하였으며 전류 값으로 기계의 정상작동 여부를 확인할 수 있도록 하였음.
 - 나. 전류 값의 상한 값과 하한 값을 설정하고 전류 값이 이 범위를 초과하면 알람이 발생하며 발생한 알람을 “ACK” 하기 전까지 계속적으로 시청각적인 경보를 울리도록 하여 이상상황에 대하여 빠른 상황 파악과 빠른 대처가 가능하도록 하였음.
3. 자체 시스템 보호 기능 (안전장치)
 - 가. 전류 제한 퓨즈를 전원포트마다 개별적으로 부착하여 과전류로 인한 제품의 손상을 방지할 수 있도록 하여 향후 고장의 원인을 원천적으로 차단하였음.
 - 나. 펌웨어와 하드웨어의 두 가지 측면에서 과전류 보호 방안을 구현하였으며 과전류가 통전되었을 시 시스템을 리셋해 주지 않으면 동작하지 않도록 하여 과전류로 인한 인적 물적 피해가 발생하지 않도록 많은 노력을 아끼지 않았음.
 - 다. 릴레이 회로에서 서지 전류를 예방하기 위한 회로와 펌웨어 제어방법을 구현하여 릴레이의 수명연장 및 릴레이의 오동작의 확률을 낮추어 제품의 신뢰도가 향상되었고 하였음.
4. 단순하고 효율적인 UI를 가진 App 개발
 - 가. 일반적인 스마트폰 앱에서는 1대1의 원격장치제어가 일반적이지만 본 과제에서는 1대 N의 통합제어를 가능하도록 하여 다수의 제어 장치가 있을 경우 이를 매우 효과적으로 제어할 수 있도록 하였음.
 - 나. 전원제어의 핵심기능만을 구현하여 화면이 작은 스마트폰에서 앱의 직관성을 높였으며 꼭 필요한 복잡한 기능과 대규모의 전원관리는 PC에서 수행하고 소규모의 전원관리는 스마트폰에서도 가능하도록 기획하였음.

제 5 장 연구개발 성과 및 성과활용 계획

제 1 절 실용화 산업화 계획

1. 양산을 위한 제품 최적화

- 가. 양산을 위한 대체부품을 선정하여 제품의 단가를 낮추고 부품의 수급을 원활히 할 계획임.
- 나. 조립 방법을 점검하고 조립 시 시간이 많이 소요되는 부분의 원인을 파악하여 이를 개선하여 조립에 소요되는 인건비를 절감하고 단위 시간당 생산량을 극대화 시킬 수 있는 방안을 마련할 예정임.
- 다. 양산을 위한 업체를 선정하고 양산비용 및 월 생산량을 점검하여 수요에 대한 수급을 원활히 할 계획임.

2. 사용자를 위한 UI 최적화

- 가. 실제 사용자들의 시력, 컴퓨터 활용능력 등을 고려하여 기 개발된 소프트웨어의 UI를 최적화할 계획이며 소프트웨어에 사용된 용어들도 실제 사용자의 수준을 고려하여 필요하다면 변경할 계획임.
- 나. 시범사업을 통한 농민들의 소프트웨어 습득 수준을 파악하여 문제점이 있거나 직관적이지 않은 UI를 개선할 계획임.
- 다. 디자인, 색상, 글자 크기 등에 대한 피드백을 조사하고 이를 적극적으로 반영할 계획임.
- 라. 향후 사업화·실용화 단계에서 실사용자의 의견을 참조하여 어플의 UI 기능을 단순하게 개선할 계획임.
- 마. 향후 사업화 단계에서 비닐하우스나 저온저장고 등 각 농업시설물의 실정에 맞게 온·습도 센서 등을 설치하고 단말기에서 온·습도 등의 정보가 표시되도록 개선할 계획임.

3. 제품 내구성 증진방안 마련

- 가. 시범농가 무료 설치를 추진하고 실제 사용상의 문제점을 파악할 계획임.
- 나. 테스트 베드를 마련하여 반복 사용을 통한 제품 내구성을 테스트할 계획이며 이를 보완하여 불량률을 감소시키고 향후 AS 비용을 절감할 예정임.

제 2 절 교육·지도·홍보 등 기술 확산 계획 등

1. 시범농가 무료 설치를 통한 viral 마케팅 실시

- 가. 성주군에 시범 농가를 지정하고 무상으로 원격전원제어기를 설치하고 실제로 사용하도록 함으로써 주변 농가들에게 자연스럽게 홍보되도록 할 계획임.
- 나. 성주군의 참외 하우스 농가들은 주거지가 대구지역인 농민들이 많고 주로 강설시 등 도로상황의 악화로 제시간에 하우스로 출퇴근하지 못하는 경우가 발생하므로 이

러한 상황에서 원격으로 하우스를 관리하는 실제적인 스토리를 이웃 농가에 홍보할 계획임.

다. 성주군 농업기술 센터와 협약을 맺고 농민기술교육장으로 활용하여 제품에 대하여 많은 농민들이 직접 체험하고 느껴 볼 수 있도록 할 계획임

2. 농민들 눈높이에 맞춘 제품 매뉴얼 / 홍보자료 제작

가. 스마트 기기의 사용이 미숙한 농민들과 시력이 좋지 않은 고령 농민들의 특성을 반영하여 읽기 쉽고 이해하기 쉬운 농민용 홍보자료를 제작할 계획임

나. 홍보자료 제작 시 실제 농민들은 참여시켜 홍보자료에 대한 피드백을 받고 이를 적극 반영할 계획임.

다. 농민들을 위하여 홍보를 위한 글자보다는 사진, 그림 등의 직관적인 자료를 적극 활용할 계획임.

3. 본 제품의 효용성에 대한 정량적 지표마련 및 홍보

가. 본 제품의 정량적 효용성을 과학적으로 측정하기 위하여 관련 대학의 링크사업 애로기술개발사업을 통하여 전문 교수를 협력하여 각종 지표를 정량적으로 측정하여 분석하는 작업을 수행할 계획임.

제 3 절 추가연구 / 타 연구에 활용 계획

1. 센서와 연동한 SMART 제어기 추가연구 추진

가. 본 과제는 전류센서만 추가된 경우로 기울기센서, 접점센서, 광센서 등과 연동하여 액추에이터가 실제로 동작하는 상황을 모니터링하며 실제동작을 피드백하여 더욱 정밀하고 능동적인 제어가 되기 위한 연구를 수행할 계획임.

2. 버섯 재배사 적용 가능성 타진

가. 재배환경에 매우 민감한 버섯 재배사에 본 과제에서 개발한 결과물을 적용하기 위하여 선행조사를 실시하고 버섯 재배사에 최적화된 시스템을 도출하도록 시스템 추가사항을 도출할 계획임.

나. 버섯 재배사에 적용 가능한 시스템을 개발하기 위하여 버섯 재배농가, 버섯 재배 전문가와 협업하여 최적의 시스템을 도출하고 이를 국가 R&D를 통하여 개발할 예정임.

3. 축산 농가 적용 가능성 타진

가. 양돈, 양계 농가에서는 가축들의 사육환경을 제어하기 위하여 환풍기, 온풍기, 사료 급이기 등의 전기기계장치를 운용하고 있으며 현재 축산 농가들은 이를 수작업으로 운용하고 있음.

나. 양돈, 양계 농가에서 운용되는 전기기계장치를 자동화하기 위하여 필요한 구성요소를 파악하고 이를 추가하여 축산 농가에 적용할 수 있는 시스템을 개발할 예정임

제 6 장 연구개발과정에서 수집한 해외과학기술정보

해당 사항 없음

제 7 장 연구시설·장비 현황

시설 및 장비명	규격	구입 가격* (백만원)	구입 년도	용도 (구입사유)	보유기관 (참여형태)
Oscilloscope	TDS360	10	2013	전기시험분석	(주)헤임달 (주관기관)
Dial Thickness Gauge	7301	0.2	2011	계측기(시제품제작)	(주)헤임달 (주관기관)
Digital Calipers	500-182	0.1	2011	전자측정기(시제품 제작)	(주)헤임달 (주관기관)
Digital Hi Tester	0804163	0.1	2011	계측기(시제품 제작)	(주)헤임달 (주관기관)
Vernier Calipers	7165947	0.1	2011	측정기(시제품 제작)	(주)헤임달 (주관기관)
데스크탑PC	P4	1.6	2012	컨트롤(운영시스템)	(주)헤임달 (주관기관)
서버	제온	3	2012	자료관리통합(운영)	(주)헤임달 (주관기관)
노트북	P4	1.7	2012	테스트 (작업관리테스트)	(주)헤임달 (주관기관)

제 8 장 연구실 안전관리 이행실적

1. 산업안전보건법 제 31조 및 동법시행규칙 33조 제1항에 의거 정기안전보건교육 실시

가. 정기 안전보건 교육

- (1) 산업안전 및 사고 예방에 관한 사항
- (2) 산업보건 및 직업병 예방에 관한 사항
- (3) 연구실내 유해, 위험요인에 관한 사항
- (4) 물질안전자료 및 그 밖에 연구실 안전관리에 관한 사항

나. 특별 안전보건 교육

- (1) 취급물질의 성질 및 상태에 관한 사항
- (2) 유해물질이 인체에 미치는 영향
- (3) 안전작업방법 및 보호구 사용에 관한 사항

2. 시설물 안전점검

가. 법적근거 : 산업안전보건법 제15조 및 시행령 12조, 재난 및 안전관리 기본법 제26조

나. 점검방법

- (1) 일일점검 : 시설물 기능유지, 설비 정상가동 등 일상적으로 확인이 필요한 사항 점검
- (2) 월간점검 : 주요시설물에 대한 기능, 성능, 안전상태 등 제반시설에 대해 월 1회 안전점검
- (3) 정기점검 : 해빙기, 하절기, 동절기 등 계절별로 안전사고가 발생하기 쉬운 시설물에 대한 안전상태

제 9 장 참고문헌

해당 사항 없음

주 의

1. 이 보고서는 농림축산식품부에서 시행한 농림축산식품기술료사업의 연구보고서입니다.
2. 이 보고서 내용을 발표할 때에는 반드시 농림축산식품부에서 시행한 농림축산식품기술료사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 아니 됩니다.