**종합설계 프로젝트 수행 보고**

|  |  |
| --- | --- |
| **프로젝트명** | **IoT기반 에너지 매니저시스템** |
| **팀번호** | **S1-7** |
| **문서제목** | **수행계획서( O )**  **2차발표 중간보고서( )**  **3차발표 중간보고서( )**  **최종결과보고서( )** |

**기업연계 : Net-KTI**

**2021.03.08**

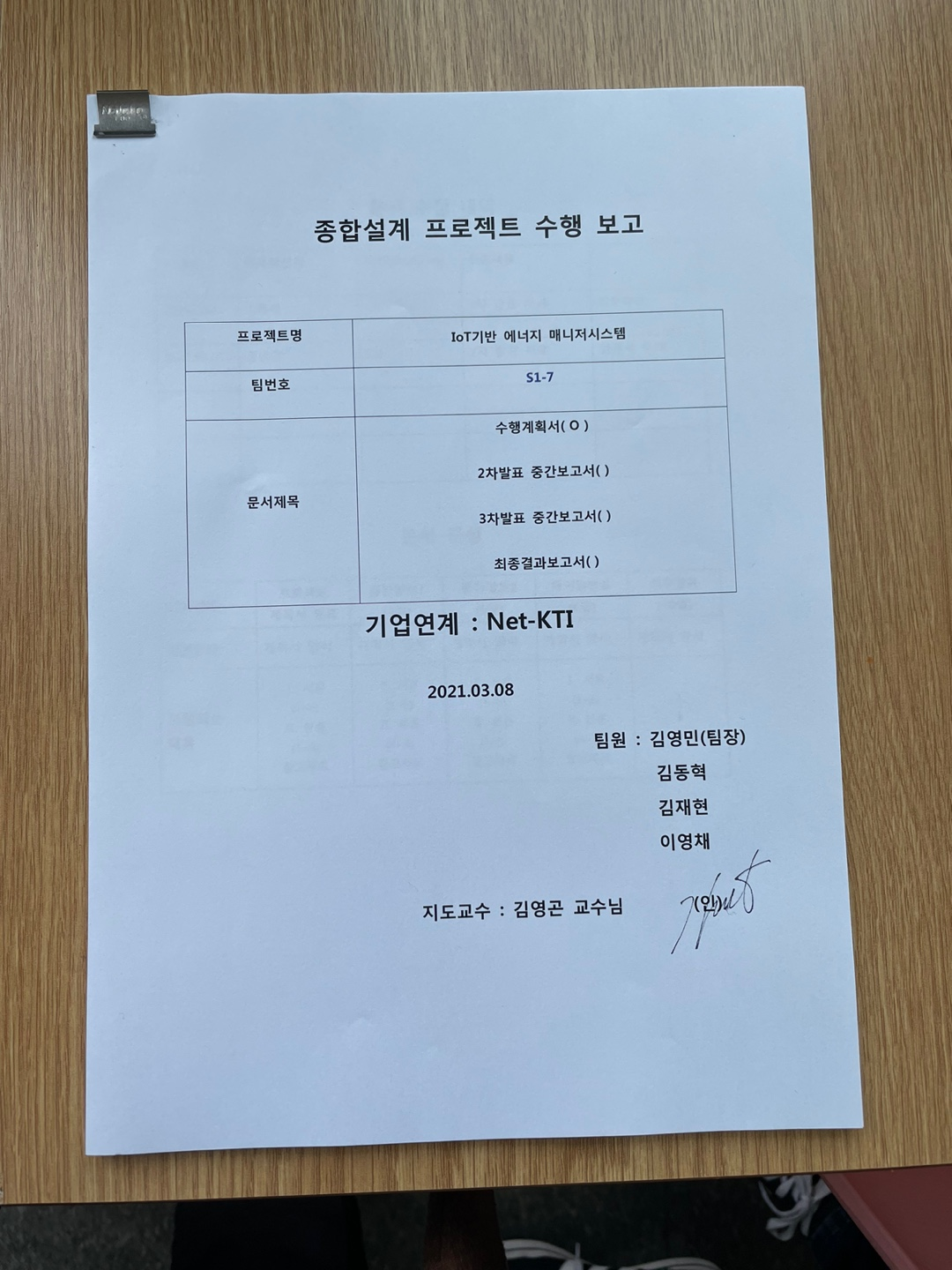
**팀원 : 김영민(팀장)**

**김동혁**

**김재현**

**이영채**

**지도교수 : 김영곤 교수님 (인)**



**문서 수정 내역**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **작성일** | **대표작성자** | **버전(Revision)** | **수정내용** |  |
| 2021.2.14 | 김동혁 | 1.0 | 1차 발표 자료 | 최초작성 |
| 2021.03.01 | 김영민 | 2.0 | 2차 발표 자료 | 설계서 추가 |
|  |  |  |  |  |
|  |  |  |  |  |

**문서 구성**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **진행단계** | **프로젝트**  **계획서 발표** | **중간발표1**  **(3월)** | **중간발표2**  **(5월)** | **학기말발표**  **(6월)** | **최종발표**  **(10월)** |
| **기본양식** | 계획서 양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 |
| **포함되는**  **내용** | Ⅰ. 서론  (1~6)  Ⅱ. 본론  (1~3)  참고자료 | Ⅰ. 서론  (1~6)  Ⅱ. 본론  (1~4)  참고자료 | Ⅰ. 서론  (1~6)  Ⅱ. 본론  (1~5)  참고자료 | Ⅰ. 서론  (1~6)  Ⅱ. 본론  (1~7)  참고자료 | I  II  III |

**이 문서는 한국산업기술대학교 컴퓨터공학부의**

**“종합설계”교과목에서 프로젝트**

“IoT기반 에너지 매니저시스템”**을 수행하는**

**(S1-7, 김영민,김재현,김동혁,이영채)들이 작성한 것으로 사용하기 위해서는 팀원들의 허락이 필요합니다.**

**목차**

[**1.** **서론** 1](#_Toc65776168)

[**1-1. 작품선정 배경 및 기존 기술 분석** 1](#_Toc65776169)

[**1-2 프로젝트 개발 목표** 2](#_Toc65776170)

[**1-3 팀 역할 분담** 3](#_Toc65776171)

[**1-4 개발 일정** 4](#_Toc65776172)

[**1-5 개발 환경** 5](#_Toc65776173)

[**2.** **본론** 7](#_Toc65776174)

[**2-1. 개발 내용** 8](#_Toc65776175)

[**2-2 문제 및 해결 방안** 11](#_Toc65776176)

[**2-3 수행 시나리오** 12](#_Toc65776177)

[**2-4 상세 설계** 15](#_Toc65776178)

[**필요 기술 및 참고 문헌** 25](#_Toc65776179)

[**1) 관련 연구 사례** 25](#_Toc65776180)

[**2) 사용 라이브러리** 25](#_Toc65776181)

[**3) 화면 설계** 26](#_Toc65776182)

[**4) 머신 러닝** 26](#_Toc65776183)

[**5) Google Cloud Platform** 26](#_Toc65776184)

**기업 질문사항 및 답변**

**기업 담당자: 김태형**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **질문 번호** | **PPT page** | **질문내용** | **팀에서 고려사항** | **답변** |
| **1** | **3p** | **주제의 적용을 매장으로 선택하고 진행하고 있는데 실제로 사용되고 있는 공간이 있는지?**  **있다면 어떤 공간인지 알려주실 수 있나요** | **저희 팀에서는 학교 내 에 있는 맘스터치와 같은 프랜차이즈 매장에 도입하는 것을 생각하고 있습니다.** | **인건비 부담으로 인해 무인화가 트렌드가 되어 카페 / 무인독서실 / 대형 프랜차이즈 등 다수의 업종에 적용되어 있습니다.** |
| **2** |  | **이런 시스템을 적용시킬 수 있는 장소**  **(1번에서 고려해주신 장소)의 샘플데이터를 받을 수 있나요?**  **(전력, 온도 등등..)** | **일단 저희가 전력데이터를 수집하기위해 접촉/ 비접촉 전류센서를 사용하고 있고, 온 습도, 모션인식을 통해 손님수를 측정하는데 더 고려할 요소가 있는지 궁금합니다.** | **샘플 데이터는 보안 및 개인정보문제로 외부 변환이 안되어 제공이 어렵습니다.**  **재실감지 센서라고 레이더센서/마이크로파 등을 이용하여 입출입 인원 파악이 가능한 센서가 있으니 찾아보시기 바랍니다.** |
| **3** |  | **전력수집 및 제어 대상으로 혹시 저희가 설정한 것 이외에 더 추가할 만한 것을 추천해 주실 수 있을까요?** | **저희가 고려한 바는**  **냉난방기구, 조명, 스크린 정도를 고려하고 있는데** | **온도센서 / 스마트플러그를 활용하면 온냉장 시설의 항온 유지 기능이 구현 가능합니다. (수족관 등 온도변화에 민감한 동식물 관리) - 산소포화도 센서 및 발생기를 추가하여 일정한 환경 구성 가능** |

표 1

|  |  |
| --- | --- |
| **서론** | |
| **주요 내용** | **1-1. 작품선정 배경 및 기존 기술 분석**  **1-2. 프로젝트 개발 목표**  **1-3. 팀 역할 분담**  **1-4. 개발 일정**  **1-5. 개발 환경** |
| **1-1. 작품선정 배경 및 기존 기술 분석** | |
| **1) 작품선정 배경**  -에너지를 효율적으로 관리하기 위한 시스템의 필요  -한창 떠오르는 4차 산업혁명의 기술 적용에 적합  이에 따라 기존 유사 기술들을 분석한 결과,   |  |  | | --- | --- | | 관련 연구 및 사례 | 개선점 | | 스마트빌딩  통합관리·E절감솔루션 | * 관련 전문 지식을 보유하지 않은 일반 사용자도 이용하기 편하도록 인터페이스를 구현 | | 오 파워 | * IoT를 사용하여 에너지 장치 각각의 데이터를 수집하고 제어한다. | | 글로벌 스마트시티(스마트 매장 에너지 관리) | * 실내 온도, 밀집도와 같은 환경 요소들도 파악하여 서비스 제공 * 빅데이터를 사용하여 적절한 에너지 사용 방법 알림 |   와 같은 개선점을 찾을 수 있었음.  이에 따른 구체적 해결방안으로  **에너지 사용량을 파악하기 위해 에너지 사용량을 시각화 해야함**.  - 사용량을 그래프를 사용하여 사용자가 알아보기 쉽게 제공.  - 지정한 사용량 초과 및 비정상적인 사용량 인식 시에 사용자에게 알림 제공  **각 에너지 사용량을 측정하는 센서 데이터들을 모아 딥러닝을 통해 자동적인 관리 시스템을 구축한다.**  - 각 요인에 따라 자동적으로 기기들을 제어하고, 수동적인 제어 역시 스마트폰이나, PC로 제어 가능하다.  - 제어된 로그들은 사용자에게 앱이나 웹의 형태로 제공된다. | |
| **1-2 프로젝트 개발 목표** | |
| 1. **연구 및 개발 목표**   IoT를 사용하여 다양한 에너지 사용량을 측정하고, 측정된 데이터를 빅데이터를 통해 분석하여 에너지를 효율적으로 사용하도록 사용자에게 알맞은 사용방법을 추천할 수 있도록 한다. 또한 인공지능에 의해 각종 에너지를 통제하고 사용자가 원격으로도 제어가 가능하도록 한다.   1. **기능정의**  * 사용자에게 에너지 사용량을 시각화하여 보여준다. * 에너지 소비 솔루션을 인공지능을 통해 제공한다. * 에너지 소비 피크 시간에 대한 정보를 인공지능을 통해 분석하여 제공한다. * 절감 효과에 대한 결과 데이터를 제공한다. * 에너지는 전기, 온도로 정의한다. * 사용량에 대한 기준을 정하고, 초과했을 때 알림 기능을 제공한다. * 웹으로 모니터링 시스템을 제공한다. * 모든 에너지는 스마트폰 어플리케이션을 통해 제어가 가능하다. * 에너지 소비에 대한 제어는 인공지능과 사용자에 의해 제어된다. * 날씨, 유동인구, 시간 데이터에 대한 정보를 사용하여 데이터를 분석한다. * 에너지 데이터는 외부 서버 데이터베이스에 저장된다. * 음성 인식을 통해 제어가 가능하도록 한다. | |
| **1-3 팀 역할 분담** | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **김재현** | **이영채** | **김영민** | **김동혁** | | **자료수집** | * **HTML, JSP** * **Android Studio** | * **AWS RDS** * **영상 처리** | * **Google colab** * **머신 러닝 모델** | * **라즈베리파이** * **Atmega, 센서** | | **설 계** | * **정보 웹 페이지** * **앱** | * **데이터 베이스 설계** * **모션 인식 개발** | * **머신 러닝 모델 적용** | * **라즈베리파이와 Atmega 연동** | | **구 현** | * **웹페이지에 정보 표시** * **앱으로도 호환** | * **데이터 베이스 구축하여 데이터 관리** * **카메라를 이용한 모션 인식** | * **머신 러닝을 통해 결과 예측** | * **라즈베리파이에서 수집한 데이터 전송** * **Atmega를 이용한 데이터 수집** | | **테스트** | * **머신 러닝 - 예측 결과** * **웹 페이지, 앱 연동** * **라즈베리파이, Atmega 연동 및 데이터 수집** | | | | | |
| **1-4 개발 일정** | |
|  | |
| **1-5 개발 환경** | |
| |  |  |  | | --- | --- | --- | | 스마트  플러그 | NODEMCU | Esp8266 탑재.  인터넷 접속과 데이터 수집을 수행 | | **ACS-712** | **교류전류데이터 수집** | | **DHT-22** | **온습도 측정** | | **5V relay** | **5v 구동 릴레이, 전자기기 전원제어** | | **AC-DC컨버터** | **교류전원을 사용하여 보드, 센서를 구동** |  |  |  |  | | --- | --- | --- | | 허브 | Arduino uno | 센서데이터 수집용(analog) | | **SCT-013** | **비접촉 전류센서 (배전반 사용)** | | **Raspberry 3b+** | **영상처리, 아두이노 전원제공, 인터넷 접속** | | **Pi camera** | **손님인식용 카메라** | | **적외선**  **송,수신 센서** | **리모콘의 신호를 복제하여 기기제어** |  |  |  |  | | --- | --- | --- | | S/W | 서버 | Apache Tomcat 9.0 / Ubuntu 18.0.4 LTS | | **DB** | **RDS for MySQL 8.0** | | **머신 러닝** | **Google Colaboratory** | | **시각화** | **Grafana** | | **모션 인식** | **OpenCV** | | |

|  |  |
| --- | --- |
| **본론** | |
| **주요 내용** | **2-1. 개발 내용**   1. **환경 데이터 수집** 2. **스마트 플러그** 3. **제어 허브(배전반 측정용)** 4. **머신 러닝** 5. **웹** 6. **데이터 베이스**   **2-2. 문제 및 해결 방안**  **2-3. 수행 시나리오**   1. **구성도** 2. **데이터 수집 시나리오** 3. **데이터 저장 & 머신 러닝 시나리오** 4. **사용자 인터페이스 시나리오** 5. **제어 시나리오**   **2-4. 상세 설계**   1. **유스케이스 다이어그램** 2. **클래스 다이어그램**   **2-1) Display**  **2-2) MachineLearning**  **2-3) Sensor**   1. **PeopleRecogSensor** 2. **ampSensor** 3. **DHTSensor**   **2-4) Control**   1. **IRControl**    1. **AirControl**    2. **MonitorControl** 2. **PlugControl** 3. **GUI 화면 설계**    1. **로그인 화면**    2. **메인 화면(전체 모니터링)**    3. **실시간 기상데이터**    4. **사용비율 비교**    5. **에너지 사용 예측**    6. **기간별 사용량**    7. **에너지 제어**    8. **문의** |
| **2-1. 개발 내용** | |
| **1) 환경 데이터 수집**    **2) 스마트 플러그**    **3) 제어 허브 (배전반 측정용)**    **4) 머신 러닝**    **5) 웹**    **6) 데이터 베이스** | |
| **2-2 문제 및 해결 방안** | |
| |  |  | | --- | --- | | **문제** | **시스템을 사용하기 위한 인터페이스는 어떻게 구현할 것인가** | | **해결 방안** | **스프링 프레임워크를 이용하여 웹을 구현하고 Apache Tomcat을 이용해 서버를 운영한다. 웹은 사용자에게 실제 에너지 사용량, 예상 에너지 사용량, 에너지 절감량, 기기 제어 등의 기능을 제공한다.** |  |  |  | | --- | --- | | **문제** | **전력데이터를 어떻게 수집할 것인가?** | | **해결 방안** | **전력을 바로 구할수 있는 센서는 존재하지 않았고 전류센서로 전압을 곱해 전력을 구할 수밖에 없었다.**  **다만 여기서 일반적인 전류센서의 경우 접촉 방식으로 전선을 단선하여 측정하는 방식이었는데 저 전류의 경우 크게 위험하지않지만 에어컨이나 높은 전류가 흐르는 제품의 경우 단선하여 연결하는데에 큰 부담이 되었다. 이에따라 조사중 비접촉 전류센서인 SCT 013 제품을 찾아 배전반에 매설하는 방식으로 전력을 측정하도록 하였다.** |  |  |  | | --- | --- | | **문제** | **제품을 어떻게 제어할 것인가?** | | **해결 방안** | **단순한 제품들이라면 그저 전원을 끊고 연결하기만하면 충분하지만 전선이 직접적으로 연결되어있지않은 즉, 비접촉 전류센서를 이용하는 제품의 경우 제어가 어려웠다. 에어컨이 가장 큰 경우였는데 고민중 IR(적외선) 신호를 이용해 기존 리모콘의 적외선 신호를 복제하여 원거리에서 풍량, 온도등을 제어하는 방법을 고안했다. 이 방법은 단순하게 에어컨 뿐만 아니라 전선이 매설되어있는 전자기기 (프로젝터등) 과 같은 기기의 제어에도 효과적일 것이라 생각한다.** |  |  |  | | --- | --- | | **문제** | **사람 인식을 함에 있어, 정면 외의 모습은 인식이 불가한 문제를 어떻게 해결할 것인가?** | | **해결 방안** | **사람의 뒷 모습이나 옆 모습, 앉아있는 모습까지 학습시키는 것에 무리가 있다고 판단하여 움직임을 감지하는 방법으로 변경하기로 했다. 데이터베이스에 전송할 때는 5분 내에 움직임이 감지되면 1을 전송하여주고, 이외에는 0을 전송하도록 했다.** |  |  |  | | --- | --- | | **문제** | **머신러닝을 어떤 분야에 적용할 것인가?** | | **해결 방안** | **초기에 대략적인 구상으로만 머신러닝을 해당 졸업 작품에 접목하여 적용 범위가 불명확했었다. 팀원들과 충분히 토의한 결과 수집한 데이터를 통해 미래의 전력 사용량을 예측하는 분야에서 머신러닝을 접목시킬 수 있다고 결론을 도출하였다. 현재 딥러닝 모델중 LSTM 모델과 Facebook에서 개발한 Prophet 모델을 사용 모델로 고려중으고, 현재 수집 중인 데이터에 대해 더 효율적인 모델을 찾기 위해 노력하겠다.** | | |
| **2-3 수행 시나리오** | |
| 1. **구성도** 2. **데이터 수집 시나리오**     * + 1. **매장 내 전력사용량을 전류센서를 통해 수집한다.**        2. **매장 내 실내온습도를 DHT22 센서를 통해 수집한다.**        3. **매장 내 카메라를 이용하여 매장내 손님수를 분석하여 라즈베리파이에  유선으로 저장한다.** 3. **시스템 수행 시나리오 – 데이터 저장 & 머신러닝**      1. **라즈베리파이에서 센서데이터를 데이터베이스로wifi를 통해 전송한다.** 2. **데이터베이스에서는 구글 클라우드로 데이터를 제공해준다.** 3. **기상청으로부터 외부 날씨 데이터를 API를 통해 받아온다.** 4. **수집한 데이터들을 통한 머신러닝을 통해 예측전력값을 구한다.** 5. **예측데이터를 데이터 베이스에 저장한다.** 6. **시스템 수행 시나리오 – 사용자 인터페이스**    * + 1. **데이터베이스에 저장된 데이터를 시각화하여 웹서버로 전송한다.**        2. **웹서버는 사용자에게 인터페이스를 제공한다.** 7. **시스템 수행 시나리오 – 제어**      * + - 1. **사용자 인터페이스에서 제어명령을 내린다.**       2. **해당 명령은 웹서버를 거쳐 원격제어 모듈로 전송된다.**       3. **IR신호, 및 릴레이를 사용하여 전열기구, 전등, 플러그, 모니터 등을 제어한다.** | |
| **2-4 상세 설계** | |
| 1. **유스케이스 다이어그램**      1. **클래스 다이어그램**     **2-1) Display**   |  |  |  | | --- | --- | --- | | **설명** | **웹에서 데이터 베이스의 정보를 이용하여 시간별 예측 전력 사용량, 현재 전력 사용량, 기기를 제어해 절감된 전력량을 시각화하여 사용자한테 제공한다.** | | | **메서드** | **public showChart(self, object):void** | **데이터를 시각화하여 사용자한테 제공한다.** | | **public getData(String,datetime) : object** | **데이터 베이스에서 현재 사용중인 데이터를 가져온다.** | | **public getPredictData(String): float[][]** | **데이터 베이스에서 머신러닝을 통해 구한 예측 전력값을 가져온다.** |   **2-2) MachineLearning**   |  |  |  | | --- | --- | --- | | **설명** | **Google Colaboratory를 통해 데이터 베이스에 저장된 현재 사용전력, 시간, 온습도, 인구 밀도를 활용하여 이후의 예측 소비 전력을 계산한다.** | | | **메서드** | **public getData(String, datetime, datetime) : object** | **데이터 베이스에서 머신러닝에 활용할 데이터를 가져온다.** | | **public AnalyzeData(object): float** | **데이터를 머신러닝을 통해 분석하여 예측값을 도출한다.** | | **public sendData(String, datetime, float) : int** | **머신러닝을 통해 도출한 값을 데이터베이스로 보낸다.** |   **2-3) Sensor**   |  |  |  | | --- | --- | --- | | **설명** | **카메라, 온습도 센서, 조명 센서를 추상화한 클래스. 각각의 센서 클래스들이 상속하여 사용한다.** | | | **메서드** | **public collectData(void) : void** | **각 센서로부터 데이터를 gpio및 아날로그 핀을 통해 수집한다.** | | **public sendData(void) : void** | **수집된 데이터를 사전에 정의해둔 sql서버로 전송한다.** | | **public setUpDatetime(void) : int** | **아두이노 보드의 wifi연결을 통해 NTP서버로부터 시간을 받아와 동기화 한다.** |     **1) PeopleRecogSensor**   |  |  |  | | --- | --- | --- | | **설명** | **카메라를 이용해 매장내의 손님수를 API를 사용하여 인식한다.** | | | **메서드** | **public collectData(void) : void** | **각 센서로부터 데이터를 gpio및 아날로그 핀을 통해 수집한다.** | | |   **2) ampSensor**   |  |  |  | | --- | --- | --- | | **설명** | **ACS712센서 및 SCT013센서를 이용해 전류를 측정한다.** | | | **메서드** | **public sendData(int, datetime, float) : int** | **전류센서의 데이터를 데이터베이스 로 현재 시간과 함꼐 전송한다.** | | | | **public collectData(int) : int** | **전류데이터를 센서로부터 수집한다.** |   **3) DHTSensor**   |  |  |  | | --- | --- | --- | | **설명** | **DHT22 센서를 이용하여 매장내의 온/습도를 측정한다.** | | | **메서드** | **public sendData(int, datetime, int, int) : int** | **온습도 센서의 데이터를 데이터베이스로 전송한다.** | | | | **public collectData(int) : int[]** | **온습도 데이터를 측정하고 이에 대한 결과를 배열로 반환한다.** |   **2-4) Control**   |  |  |  | | --- | --- | --- | | **설명** | **자체 제작한 스마트 플러그, 배전반을 조작하는 인터페이스**  **인터페이스를 이용하여 각각의 컨트롤러에 연결된 기기를 제어한다.** | | | **메서드** | **public deviceOn(int): int** | **디바이스의 전원을 연결된 릴레이를 통해 켜는 함수이다.** | | **public deviceOff(int): int** | **디바이스의 전원을 연결된 릴레이를 통해 끄는 함수이다.** |   **1) IRControl**   |  |  |  | | --- | --- | --- | | **설명** | **적외선 리모컨을 사용하는 제품의 제어를 담당하는 클래스이다.**  **AirControl과 MonitorControl에 상속을 해준다.** | | | **메서드** | **public copySignal(void): ulong** | **리모컨의 IR신호를 시스템에 저장받는 함수이다.** |   **1-1) AirControl**   |  |  |  | | --- | --- | --- | | **설명** | **IRControl에서 상속을 받아 바람 세기와 온도를 조절하는 클래스이다.** | | | **메서드** | **public controlTemp(ulong) : int** | **냉난방기의 온도를 조절하는 함수이다.** | | **public controlPower(ulong) : int** | **냉난방기의 바람 세기를 조절을 하는 함수이다.** |     **1-2) MonitorControl**   |  |  |  | | --- | --- | --- | | **설명** | **IRControl에서 상속을 받아 TV를 제어하는 클래스이다.** | | | **메서드** | **public deviceTVOn(ulong): void** | **TV의 전원을 켜는 함수이다.** | | **public deviceTVOff(ulong): void** | **TV의 전원을 끄는 함수이다.** |   **2) PlugControl**   |  |  |  | | --- | --- | --- | | **설명** | **스마트 플러그의 제어를 하는 클래스이다.**  **Control interface를 사용한다.** | | | **메서드** | **public relayControl(void): int** | **릴레이를 통해 스마트 플러그의 장비를 제어한다.** |   **2-5) DB\_interface**   |  |  |  | | --- | --- | --- | | **설명** | **AWS RDS를 사용할 수 있는 데이터베이스 인터페이스. 서버 주소, 사용자 정보를 이용하여 데이터 베이스를 사용한다.** | | | **메서드** | **public loginServer(IPAddress, char, char): int** | **데이터 베이스에 매장별 접속을 할 수 있게 해주는 함수이다.** | | **public addNewStore() : int** | **데이터베이스에 새로운관리대상 매장을 추가하는 메소드** |  1. **GUI 화면 설계**   **웹 화면설계**   * 1. **로그인 화면**      * 1. **메인화면(전체 모니터링)**      * 1. **실시간 기상데이터**      * 1. **사용비율 비교**      * 1. **에너지 사용 예측**      * 1. **기간별 사용량**      * 1. **에너지 제어**      * 1. **문의**     **상세 설명**   * **사용자는 매장에 부여된 아이디와 비밀번호를 통해 웹으로 접근을 한다.** * **로그인 후 전체적인 에너지의 모니터링을 하는 메인 페이지가 출력된다.** * **왼쪽의 네비게이션 바를 통해 원하는 메뉴로 접근한다.** * **각각의 메뉴를 통해 사용자에게 유용한 정보를 제공한다.** * **제어 메뉴에서는 사용자가 전자기기를 제어할 수 있도록 한다.** | |

|  |
| --- |
| **필요 기술 및 참고 문헌** |
| **1) 관련 연구 사례** |
| **- 2017년 차세대에너지연구소 고유과제 결과 보고서**  [**https://scienceon.kisti.re.kr/commons/util/originalView.do?cn=TRKO201800035622&dbt=TRKO&rn=**](https://scienceon.kisti.re.kr/commons/util/originalView.do?cn=TRKO201800035622&dbt=TRKO&rn=)  **- KT meg**  [**http://www.enet.or.kr/?module=file&act=procFileDownload&file\_srl=54172&sid=a4e9f96b0af328c594ce40959e4adba9&module\_srl=148**](http://www.enet.or.kr/?module=file&act=procFileDownload&file_srl=54172&sid=a4e9f96b0af328c594ce40959e4adba9&module_srl=148) **pp. 2-18**  **- 빌딩 에너지 관리 시스템의 기술 및 표준화 동향**  [**https://www.google.co.kr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi9s\_rrjaftAhUXyIsBHZkKB7sQFjADegQIARAC&url=https%3A%2F%2Ftta.or.kr%2Fdata%2FreportDown.jsp%3Fnews\_num%3D3548&usg=AOvVaw0K5dIwbvH1Xv2kY4e79-zY**](https://www.google.co.kr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi9s_rrjaftAhUXyIsBHZkKB7sQFjADegQIARAC&url=https%3A%2F%2Ftta.or.kr%2Fdata%2FreportDown.jsp%3Fnews_num%3D3548&usg=AOvVaw0K5dIwbvH1Xv2kY4e79-zY)  **- 오파워**  [**https://www.hellot.net/new\_hellot/magazine/magazine\_read.html?code=205&sub=001&idx=44952&page=1&list=all**](https://www.hellot.net/new_hellot/magazine/magazine_read.html?code=205&sub=001&idx=44952&page=1&list=all)  **- 글로벌 스마트 시티**  [**http://www.k-smartcity.kr/service1/service07.php**](http://www.k-smartcity.kr/service1/service07.php) |
| **2) 사용 라이브러리** |
| * **아두이노 DHT 라이브러리** [**https://github.com/adafruit/DHT-sensor-library**](https://github.com/adafruit/DHT-sensor-library) * **ESP8266 인터넷시간 알아보는 라이브러리** [**https://github.com/mcxiaoke/ESPDateTime**](https://github.com/mcxiaoke/ESPDateTime) * **Mysql connector 척벨 (**[**https://github.com/ChuckBell/MySQL\_Connector\_Arduino**](https://github.com/ChuckBell/MySQL_Connector_Arduino)**)** * **SCT-013 사용관련 회로참고** [**https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/how-to-build-an-arduino-energy-monitor-measuring-current-only?redirected=true**](https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/how-to-build-an-arduino-energy-monitor-measuring-current-only?redirected=true) |
| **3) 화면 설계** |
| **- 라임 ICT** [**http://limeict.co.kr/product\_cb.html**](http://limeict.co.kr/product_cb.html) |
| **4) 머신 러닝** |
| * **시계열 예측** [**https://www.tensorflow.org/tutorials/structured\_data/time\_series**](https://www.tensorflow.org/tutorials/structured_data/time_series) * **Prophet** [**https://facebook.github.io/prophet/docs/quick\_start.html#python-api**](https://facebook.github.io/prophet/docs/quick_start.html) * **LSTM** [**https://brunch.co.kr/@chris-song/9**](https://brunch.co.kr/@chris-song/9) |
| **5) Google Cloud Platform** |
| * **역 지오 코딩** [**https://cloud.google.com/solutions/reverse-geocoding-geolocation-telemetry-cloud-maps-api?hl=ko**](https://cloud.google.com/solutions/reverse-geocoding-geolocation-telemetry-cloud-maps-api?hl=ko) * **IP Geolocation API** [**https://geo.ipify.org/**](https://geo.ipify.org/) |