

종합설계 프로젝트 수행 보고서

프로젝트명	모션감지센서를 이용한 스마트 스위치
팀번호	S4-3
문서제목	수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 최종결과보고서(O)

2020.12.4.

팀원 : 전민서 (팀장)
김세현

지도교수 : 나보균 교수 (인)
지도교수 : 한경숙 교수 (인)

문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2020.1.9	전민서(팀장)	1.0	서론	최초작성
2020.2.27	전민서(팀장)	2.0	상세설계	내용추가
2020.5.2		2.1	2-1 ~ 2-5	내용추가 및 수정
2020.6.25	전민서(팀장)	3.0	본론	내용추가 및 수정
2020.6.27		3.1	본론	내용추가 및 수정
2020.12.4	전민서(팀장)	4.0	3	내용추가

문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (5월)	학기말발표 (6월)	최종발표 (12월)
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 “모션감지센서를 이용한
 스마트 스위치”를 수행하는
 (S4-3, 전민서, 김세현)팀원들이 작성한 것으로 사용하기
 위해서는 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성	4
2. 기존 연구/기술동향 분석	4
3. 개발 목표	5
4. 팀 역할 분담	5
5. 개발 일정	6
6. 개발 환경	6

II. 본론

1. 개발 내용	7
2. 문제 및 해결방안	8
3. 시험시나리오	8
4. 상세 설계	10
5. Prototype 구현	21
6. 시험 / 테스트 결과	24
7. Coding & DEMO	25

III. 결론

1. 연구 결과	34
2. 작품제작 소요재료 목록	35

참고자료	36
------------	----

I. 서론

1. 작품선정 배경 및 필요성

최근 자기 전에 침대에서 스마트폰이나 독서 등의 행위를 하다가 잠드는 사람이 많다. 이 때 전등 스위치가 침대 가까이 있으면 전등을 끄고 자겠지만 스위치가 멀리 있어 전등을 끄려면 침대 밖을 나가야 하는 번거로움이 생기고 이로 인해 전등을 끄지 않고 자거나 깜빡하고 전등을 못 끄고 잠들어버리는 경우가 생기고 있다.

불을 켜 채로 잠이 드는 경우 눈꺼풀 위로 형광등 빛을 감지하여 뇌가 깨어있는 상태가 되고 깊게 잠들 수 없게 된다. 우리 뇌는 깊게 잠들 때 성장호르몬을 분비하고 자정작용을 통해 노폐물을 제거하며 낮에 쌓인 정보를 정리하는데, 폭 잠들지 못하면 불면증을 일으켜 우울증에 걸릴 확률이 높아지며, 성장기인 경우 성장 발달을 저하한다. 노인에게는 치매를 일으킬 확률이 높아진다.

또한 간편하게 자신의 수면의 질이 어떤지 간단하게 체크를 할 수 있어 병원만큼 자세하진 않으나 자신이 뒤척여서 깊게 잠드는지, 얼마나 오래 폭 잠드는지 간이진단이 가능하다.

2. 기존 연구/기술동향 분석

기존의 유사시스템으로는 스마트홈과 밀접하게 연관되어 있는 스마트 스위치가 있다. 스마트 스위치는 처음엔 간단하게 타이머 기능을 내장한 것으로 시작했다. 시간이 되면 점등 및 소등이 되어 모닝콜과 취침 모드를 노린 것이었다. 여기서 더 발전한 것이 모바일 연동 기능이 있는 스마트 스위치로 핸드폰으로 언제 어디서나 집 안의 전등을 제어할 수 있게 되었다. 집 안에서 제어하는 것에서 벗어나 전등을 깜빡하고 끄지 않고 외출하거나 귀가 중에 집 밖에서도 핸드폰을 통해 제어할 수 있게 된 것이다. 최근엔 인공지능과도 결합 되고 있는데 대표적인 상품이 SKT 스마트 스위치이다. 이 상품은 핸드폰으로 제어가 가능할 뿐 아니라 SKT의 인공지능 음성인식 디바이스인 NUGU와도 연동이 가능해져 사용자의 음성으로 스위치를 제어할 수 있다는게 특징이다.

그러나 위의 시스템 모두 사용자가 직접(목소리로라도) 제어해 스위치를 끄는 방법이며 센서로의 확장성을 생각 못 한 것으로, 본 프로젝트는 PIR 센서를 연동해 센서를 이용해 사용자가 직접 제어할 필요가 없으며 방법 기능도 추가한 스마트 스위치로 확장성과 편의성을 꾀했다.

3. 개발 목표

- 1) 어플리케이션, 전등 스위치, 모션 감지 센서, 서버의 네트워크 구축
- 2) 어플리케이션을 이용해 전등 스위치 원격 제어
- 3) 모션 감지 센서를 이용, 사용자의 움직임이 감지되지 않을 때(잠들었을 경우) 방의 전등 소등
- 4) 방의 전등이 꺼져있는 동안의 사용자 수면 시간과 수면 시간 중 뒤척임 횟수를 바탕으로 수면 패턴 시각화
- 5) 숙면에 대한 정보 제공
- 6) 현관이나 창문 등에서 움직임이 감지될 경우(침입자가 나타날 경우) 방의 전등 점등 및 모바일 푸쉬 발송.

4. 팀 역할 분담

	전민서	김세현
자료탐색	설계 및 개발환경, 하드웨어 및 기능 개요, 아두이노 구현, 웹서버 구현 및 그 외 전반적인 내용	어플리케이션 기능 세부 구현 및 API, AWS 서비스, DB 구현
설계	시스템 설계, 하드웨어 설계, 아두이노 및 AWS 웹 서버 설계	Application 설계, 스마트폰-서버 데이터 간 설계
구현	아두이노 웹 서버 구현, AWS 웹 서버 구현, Application 개발 보조	Application 개발, AWS 서비스, DB, 센서→서버 데이터 관리
테스트	하드웨어 테스트, 아두이노 및 AWS 웹 서버 동작 테스트, 아두이노 기능 동작 테스트(자동 소등 및 뒤척임 로그), 휴대폰-아두이노간 연결테스트(Wi-Fi), 어플리케이션 기능 테스트(수면분석 기능 테스트, 분석 데이터 업로드), 어플리케이션-AWS서버 간 데이터 통신 테스트, AWS 서버 작동 및 DB 테스트	

5. 개발 일정

순서	항목	'19년 12월	'20년 1월	2월	3월	4월	5월	6월	7월	8월	9월	10월
1	종합설계 제안서 제작 및 발표 준비											
2	PIR센서 및 전등 스위치 구매, 하드웨어 준비											
3	프로토타입 설계(센서-스위치 연동, 자동 소등)											
4	세부 설계 (앱-센서 네트워크 구축, 앱 제작)											
5	프로토타입 구현											
6	프로토타입 테스트 및 오류 수정											
7	세부 구현											
8	시제품 테스트 및 오류 수정											
9	최종보고서 작성 및 제품 마무리											
10	논문 및 보고서 작성											

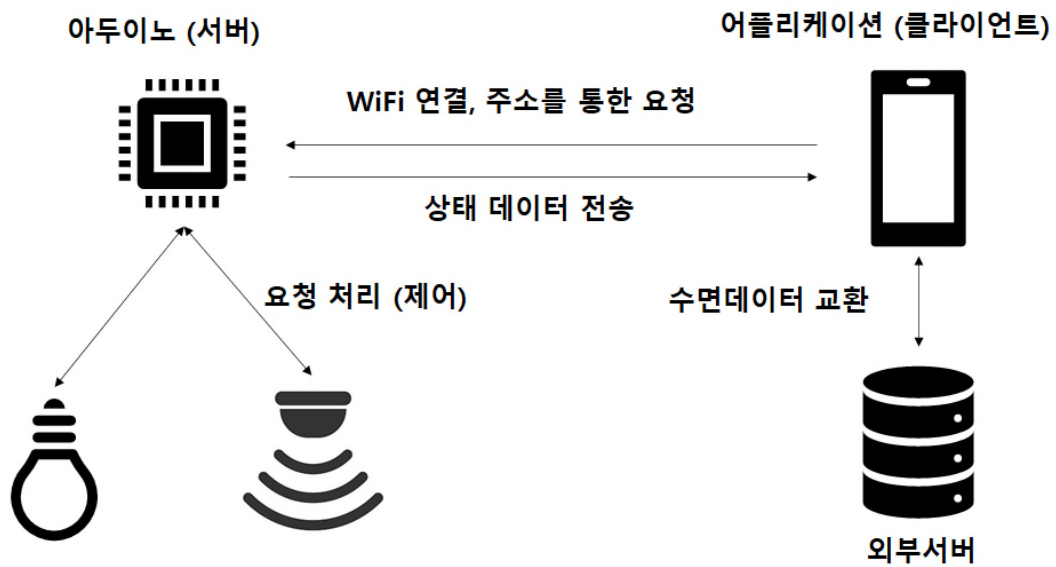
6. 개발 환경

- 1) 개발언어 : JAVA, C/C++, jsp
- 2) 통신 : Wi-Fi
- 3) 사용 프로그램 : Android Studio, Arduino, Apache Tomcat, MySQL, AWS
- 안드로이드 버전 최소 요구사항 : 6.0 (Marshmallow, API Level 23)
- 4) 서버 : AWS
- 5) DB : MySQL
- 6) 모듈 : NodeMCU, PIR 센서

II. 본론

1. 개발 내용

1) 시스템 구성도



2) 기능

(1) 원격제어

- 스위치의 Wi-Fi 모듈과 어플리케이션 동기화
- 어플리케이션을 통해 스위치 On-Off 제어

(2) 자동소등

- 센서 기능 시간 설정
- 수면 기준 설정(기준 시간당 뒤척임 횟수)
- 기능 시간 중 수면 기준 도달 시 소등

(3) 방범기능

- 기능 시간 중 센서 감지 시 점등
- 감지 시 휴대전화로 푸쉬 송신

(4) 수면관리

- 자동 소등의 기능 시간 중 뒤척임 로그 저장
- 로그를 그래프화 및 어플리케이션에 출력
- 서버 동기화
- 서버에서 사용자의 수면 패턴 그래프 평균화

2. 문제 및 해결방안

- 1) PIR 센서 민감도를 앱을 통해 제어하려 하였으나, 이 기능을 수행할 수 있는 센서는 시중에 판매되지 않아 센서 자체의 조절 기능을 활용하여야 함.
- 2) 아두이노 웹 서버 내에 rtc모듈을 달지 않아 서버 작동시 최초에 어플리케이션에서 시간 동기화가 필요.(기능 구현 완료)

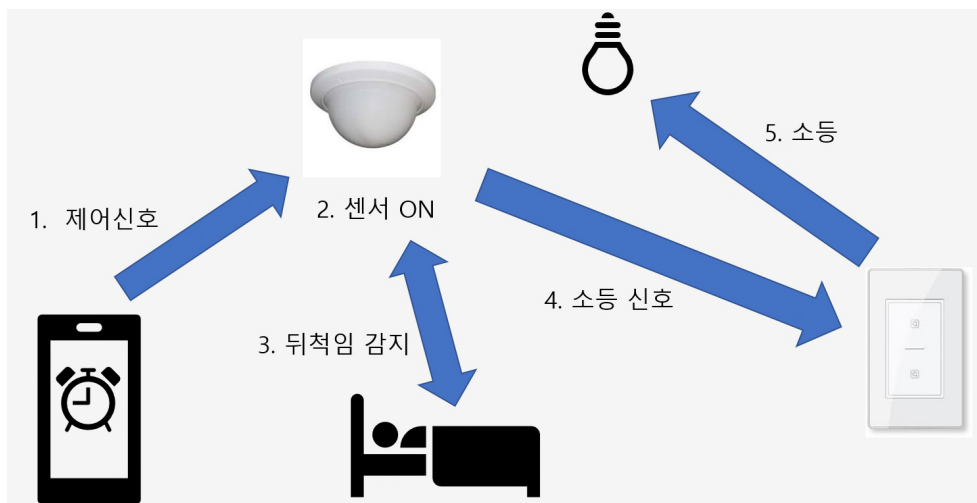
3. 시험시나리오

1) 원격제어

- 모바일 어플리케이션을 통해 스위치 원격제어.

2) 자동소등

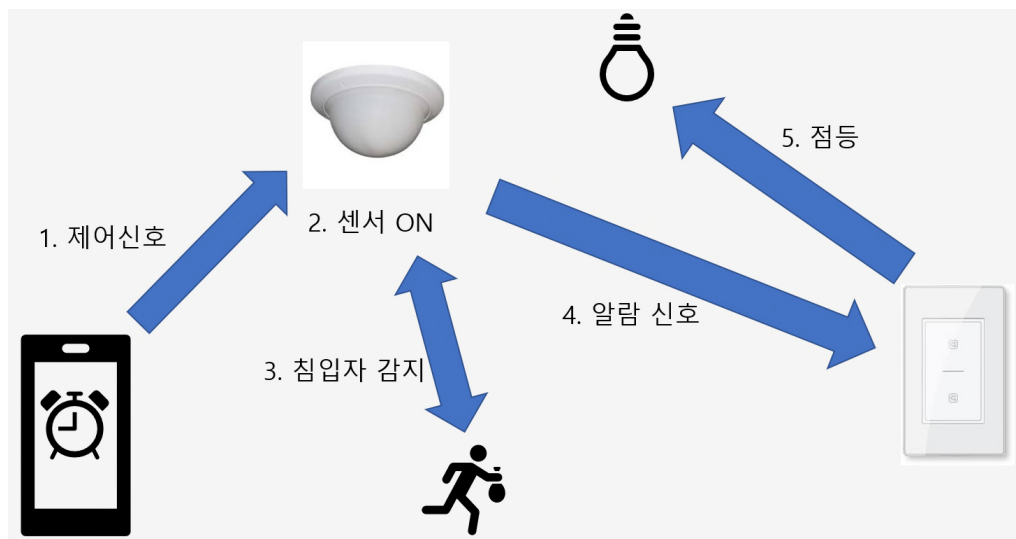
- 사용자의 수면 감지, 방의 전등을 자동으로 소등.



- ① 사용자가 앱을 통해 특정 시각으로 타이머를 맞추면 해당 시각에 PIR 센서가 켜진다.
- ② 기준에 따라 센서를 작동한다. (예: 10분마다 30초 동안 센서 on)
- ③ 기준 시간 동안 센서 감지가 없으면 스위치에 신호를 보낸다.
- ④ 신호 수신한 스위치가 전등을 끈다.

3) 방법기능

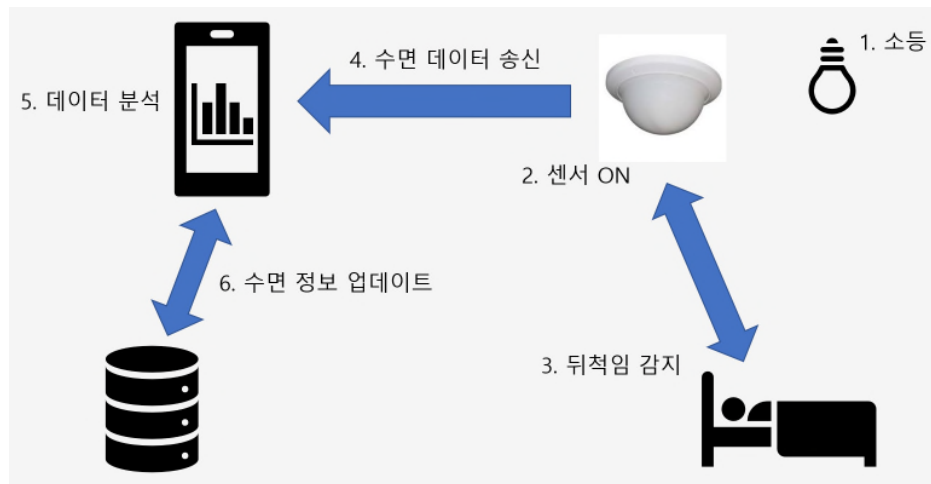
- 창문 및 현관의 센서를 통해 침입자 감지, 알람 신호



- ① 사용자가 앱을 통해 특정 시각에 타이머를 맞추면 해당 시각에 PIR센서를 작동한다.
- ② 설정된 시간 동안 PIR 센서를 상시 작동한다.
- ③ PIR 센서에 신호 감지 시 스위치에 신호를 송신한다.
- ④ 전등 점등 및 메시지를 송신한다.

4) 수면관리

- 수면 중 뒤척임 감지, 사용자에게 정보를 그래프화 하여 수면 패턴 분석.
- 서버와 동기화, 사용자들의 평균값을 구해 사용자 제공.
- 숙면에 도움이 되는 정보를 사용자 제공.



- ① PIR 센서 소등 시간 동안 기능이 작동된다.
- ② 작동 중 사용자의 뒤편을 감지한다.
- ③ 서버에 시간을 기록한다.
- ④ 전등 점등 시 PIR 센서 및 기록을 멈춘다.
- ⑤ 사용자에게 앱을 통해 수면 시간 및 뒤편 정도를 그래프로 제공한다.

4. 상세설계

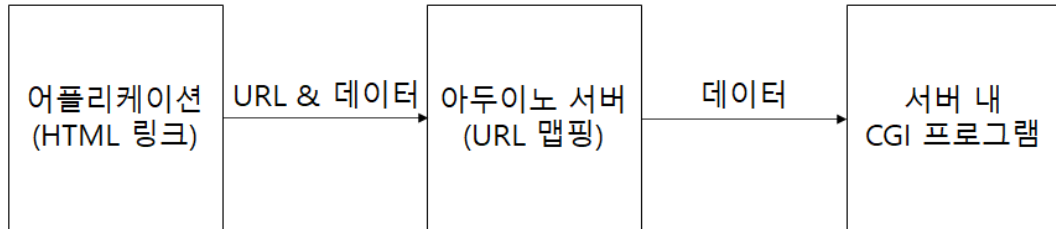
- 기능별 세부 설계

공통

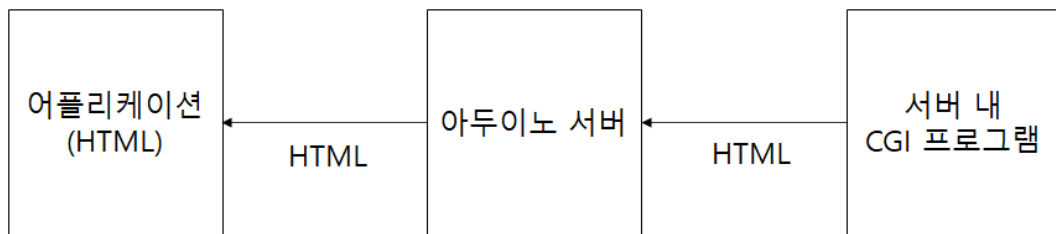
- 1) 아두이노에서 웹 서버 객체 생성
- 2) 웹 서버 시작
- 3) 클라이언트 요청
- 4) 어플리케이션을 통해 선언된 주소로 접속 (ex) 192.168.0.18/test)
- 5) 주소에 따른 처리함수 호출

•작품의 아두이노 서버는 CGI를 사용하고 있음.

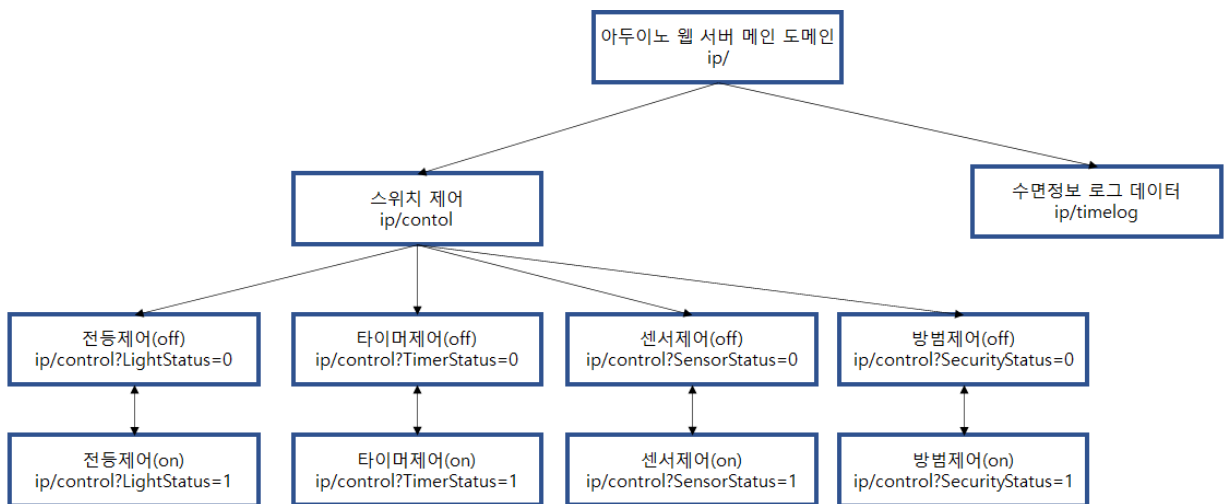
1) 사용자가 어플리케이션 조작 시



2) 서버에서 사용자에게 데이터 반환



작품 내 CGI 작동원리



URL 구성도

1 - 스위치 제어

- 1) 주소는 ip/control
- 2) control 페이지는 전등, 센서, 방범기능의 상태를 보내고 어플리케이션은 이를 스위치 형식으로 나타냄.

1-1 - 전등 제어

- 1) 스위치 원격제어 화면에 진입
- 2) 전등 스위치 on/off를 누르면 리스너에서 이벤트 처리.
- 3) on/off 상태에 따라 http request를 아두이노 서버에 전송.
ex) ip/control.cgi?LightStatus=1 -> 전등 ON, ip/control.cgi?LightStatus=0 -> 전등 OFF
- 4) 아두이노 서버에서 request에 따른 명령어 처리.

1-2 - 센서 제어

- 1) 스위치 원격제어 화면 진입.
- 2) 센서 스위치 on/off를 누르면 리스너에서 이벤트 처리.
- 3) on/off 상태에 따라 http request를 아두이노 서버에 전송.
ex) ip/control.cgi?SensorStatus=1 -> 센서 ON, ip/control.cgi?SensorStatus=0 -> 센서 OFF
- 4) 아두이노 서버에서 request에 따른 명령어 처리.
- 5) 센서 작동 & 전등 점등 시에 설정된 조건이 만족 시 전등 소등, 센서는 계속 작동 중이며 센서 작동 & 전등 소등 시에 뒤척임 감지 후 로그 기록

1.3 - 방법 제어

- 1) 스위치 원격제어 화면 진입
- 2) 스위치 on/off를 누르면 리스너에서 이벤트 처리.
- 3) on/off 상태에 따라 http request를 아두이노 서버에 전송.
ex) ip/control.cgi?SecurityStatus=1 -> 방법 센서 ON,
ip/control.cgi?SecurityStatus=0 -> 방법 센서 OFF
- 4) 아두이노 서버에서 request에 따른 명령어 처리.
- 5) 방법 제어 작동 중 센서 감지가 일어나면 전등을 점등하고 어플리케이션으로 푸쉬 송신.

2 - 수면 관리

- 1) 센서 기능 작동 후 센서 작동 중이고 전등은 소등 된 상태일 때 센서 감지시마다 시간 로그 축적, 서버에 기록.
- 2) ip/timelog 접속 시 로그 파일 출력.
- 3) 어플리케이션에서 출력 된 로그 파일을 내부 저장소에 복사.
- 4) 복사한 로그 데이터를 MPAndroidChart 라이브러리를 이용해 꺾은 선 그래프로 시각화.
- 5) 별도의 버튼 터치 시 로그 데이터를 AWS 웹 서버를 통해 DB에 저장.
- 6) AWS 서버의 DB는 이용자들의 로그 데이터를 축적, 평균화.
- 7) 사용자는 AWS의 평균 데이터와 본인의 수면 데이터를 비교 가능.
- 8) 수면 Tip 메뉴는 수면이 어려운 사용자를 위해 간단한 정보 제공.

- 모듈별 설계

아두이노

- 객체

Server -> ESP8266WebServer(int port) 함수를 이용해 생성하는 아두이노 웹 서버 객체

WiFi -> ESP8266을 위한 와이파이 클래스인 ESP8266WiFiClass의 객체. 와이파이 관련 동작을 수행하는 객체

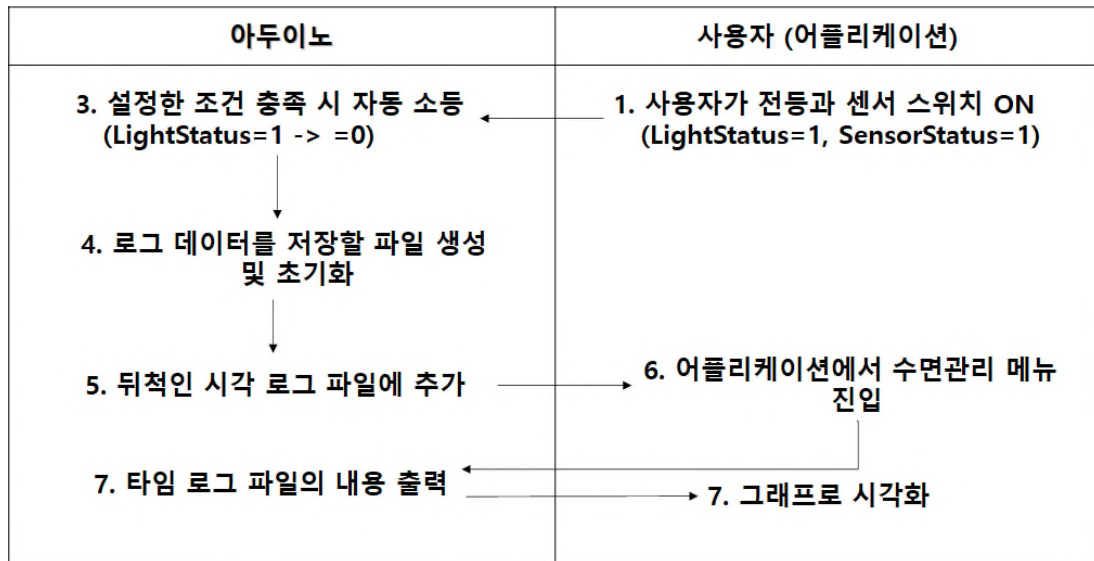
File -> 수면 기록 로그를 저장할 로그 파일 객체

- 함수

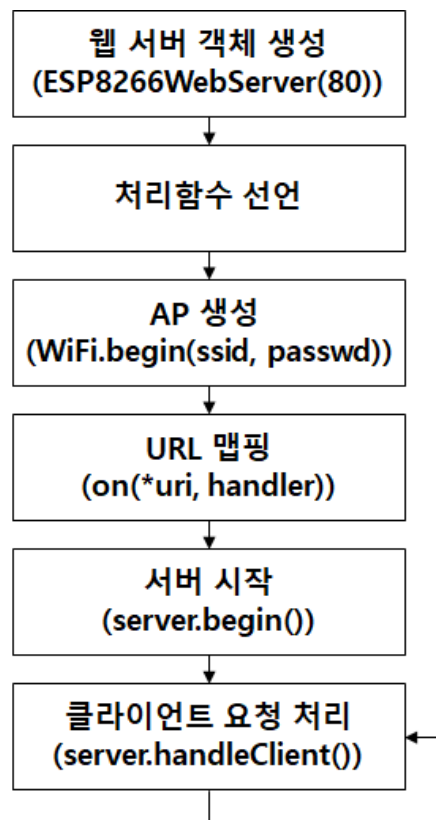
제공되는 함수				
함수	매개변수	반환값	역할	기타
ESP8266WebServer(int port)	포트번호	-	웹서버 객체생성.	HTTP 프로토콜 사용하므로 매개변수 80
void on(const char*uri, THandlerFunction handler)	주소, 처리 할 handler 함수	-	주소와 처리 함수를 연결	
void begin()	-	-	서버 시작	
void handleClient()	-	-	클라이언트 요청 처리	
unsigned long millis(void)	-	밀리초 단위 시간	프로그램 시작 이후의 경과 시간을 밀리초 단위로 반환.	반환 값은 4바이트 크기로 약 50일이 지나면 오버플로가 발생, 0이 됨.
void delay(unsigned long ms)	밀리초 단위 지연시간	-	지정한 시간만큼 프로그램 실행을 일시 중지.	
File SPIFFS::open(char *path, char *mode)	파일 이름, 파일 열기 옵션	파일 열기에 실패할 경우 NULL	수면 데이터 로그 파일 생성 및 쓰기	NODEMCU의 플래시 메모리 4mb, 1mb는 펌웨어 공간으로 여유공간은 약 3mb

void send(int code, char* content_type, String content)	HTTP 응답 코드, 전송 내용 종류, 전송 내용	-	서버에서 어플리케이션으로 데이터를 post	
void digitalWrite(uint8_t pin, uint8_t value)	핀 번호, value(HIGH or LOW)	-	전등 제어	
void digitalRead(uint8_t pin)	핀 번호	H I G H or LOW	센서로부터 디지털 값을 읽어 반환	
setTime(int hour, int minute, int second, int day, int month, int year)	시, 분, 초, 일, 월, 년	-	서버의 시간값 설정	설정 된 서버의 초기 시간 값은 2020-1-1, 00:00:00으로 서버 기동 시 어플리케이션에서 시간 동기화 필요

구현할 함수				
함수	매개변수	반환값	역할	기타
void mainpage()	-	-	서버 IP, 시간, 센서 상태 출력	브라우저에서 “ip/” 로 접근 가능
void controlpage()	-	-	전등 및 센서를 조작 하는 페이지 출력	“ip/control” 에서 접근. get으로 조작처리를 받아 controlcgi 작동
Void controlcgi()	-	-	controlpage에서 get 한 명령어 처리	사용자가 submit한 조작 데이터를 처리, 센서 및 전등 조작
void timecgi()	-	-	서버 시간 값 조작	get으로 시간 값을 받아 서버 시간 조작. 초기 서버 시간 값은 2020년 1월 1일 0시 0분 0초
void timelog()	-	-	사용자의 수면 중 뒤척임 타임 로그 데이터 출력	“ip/timelog” 에서 접근. 뒤척임 시각의 년/월/일/시까지 출력
String timenow()	-	현재시간	현재시간을 출력	



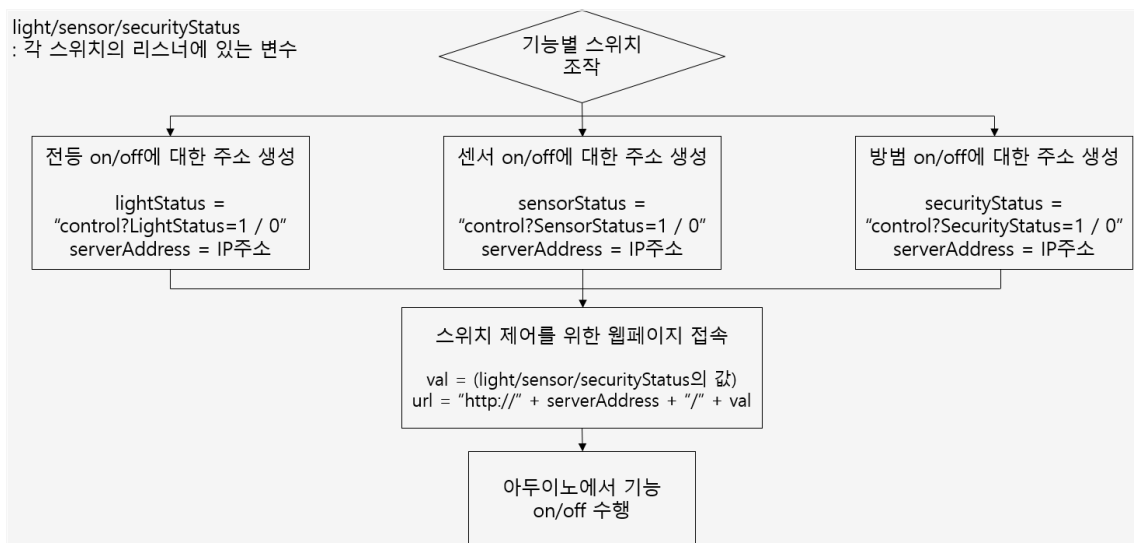
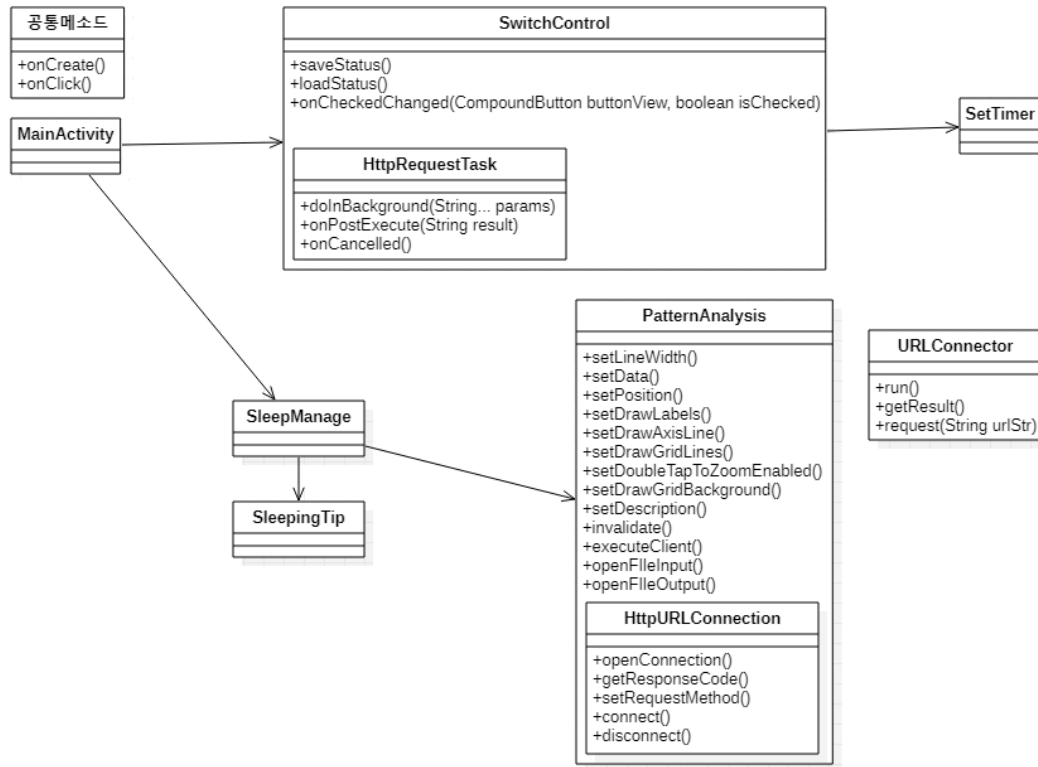
전등제어-수면관리 흐름도



-스레드 흐름도

어플리케이션

- 클래스 다이어그램



어플리케이션 원격제어 흐름도

- 메소드(함수)

각 액티비티 별 공통 메소드			
직접 구현할 메소드			
이름	매개변수	반환값	용도
onCreate(Bundle savedInstanceState)	액티비티의 이전 저장된 상태	void	액티비티 생성 시 1회 호출됨.
onClick(View view)	뷰	void	버튼 클릭 시 호출되는 콜백 메소드

MainActivity - 홈 화면에 대한 클래스			
각 기능을 선택하거나 앱 종료하는 버튼만 존재하며, 이 화면에서 특별한 기능을 수행하지 않음.			

SwitchControl - 스위치 원격제어 화면에 대한 클래스			
직접 구현할 메소드			
이름	매개변수	반환값	용도
saveStatus	-	void	스위치 상태 저장
loadStatus	-	void	스위치 상태 불러오기
onCheckedChanged	in CompoundButton buttonView, in boolean isChecked	void	스위치 클릭 시 호출되는 메소드

HttpRequestTask - http request 작업에 대한 클래스로, SwitchControl 내부 클래스. 여기서 사용할 API/라이브러리: okhttp			
doInBackground(String... params)	작업에 대한 매개변수 문자열	문자열	스레드 생성 후 백그라운드에서 실행되는 작업 메소드
onPostExecute(String result)	결과값 문자열	void	UI 업데이트를 위해 UI 스레드에서 실행되는 메소드
onCancelled()	-	void	작업 취소 시 실행되는 메소드

주요 멤버(데이터) - SwitchControl 클래스			
이름	타입	메소드	용도
light	Switch	onCreate()	전등 on/off에 대한 스위치
sensor	Switch	onCreate()	센서 on/off에 대한 스위치
security	Switch	onCreate()	방범기능 on/off에 대한 스위치
주요 멤버(데이터) - HttpRequestTask 클래스			
serverAddress	String	클래스의 멤버 변수	서버 주소 저장 목적
val	String	doInBackground()	서버에서 접속할 페이지 이름을 넘겨받는 변수
url	String	doInBackground()	serverAddress와 val을 합쳐서 접속할 페이지 주소를 만드는 목적

SleepManage - 수면 관리 기능 화면에 대한 클래스
수면 패턴 분석/숙면 Tip 화면 또는 이전 화면으로 돌아가는 버튼만 존재하여 이들 버튼에 대한 처리 코드만 있으며, 특별한 기능을 수행하지 않음.

PatternAnalysis - 수면 패턴 분석 화면에 대한 클래스			
여기서 사용할 라이브러리: MPAndroidChart			
MPAndroidChart에서 제공되는 주요 메소드 (꺾은선 그래프)			
이름	매개변수	반환값	용도
setLineWidth(float width)	폭(소수점 포함)	void	꺾은선의 폭을 설정
setData(ChartData data)	차트 데이터	void	차트에 값을 추가
setPosition(XAxisPosition pos)	X축 위치	void	X축이 어느 위치에 나타날지 결정
setDrawLabels(boolean enabled)	사용유무(boolean)	void	축 레이블을 나타낼지 결정
setDrawAxisLine(boolean enabled)	사용유무(boolean)	void	축 선을 그릴지 결정
setDrawGridLines(boolean enabled)	사용유무(boolean)	void	축의 기준선(0, 1, 2 식으로) 그릴지 결정
setDoubleTapToZoomEnabled(boolean enabled)	사용유무(boolean)	void	두 번 탭할 때 확대/축소를 허용할지 결정
setDrawGridBackground(boolean enabled)	사용유무(boolean)	void	차트 그리지는 영역 배경에 사각형 배경을 넣을지 결정
setDescription(String desc)	설명 문자열	void	오른쪽 아래 코너에 나올 차트 설명 텍스트 삽입
invalidate()	-	void	차트를 새로고침.

PatternAnalysis - 수면 패턴 분석 화면에 대한 클래스			
파일 입출력 관련 표준 자바 라이브러리에서 제공되는 메소드			
이름	매개변수	반환값	용도
openFileInput(String name)	파일 이름	파일 입력 스트림	입력 전용 파일 오픈
openFileOutput(String name, int mode)	파일 이름, 출력 모드	파일 출력 스트림	출력 전용 파일 오픈

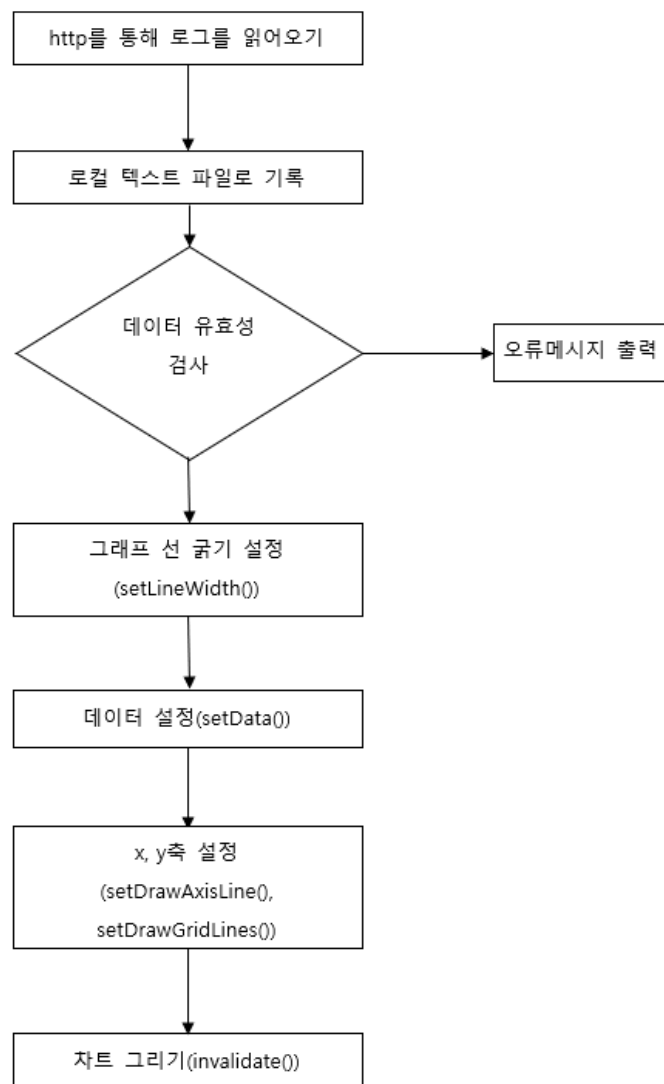
PatternAnalysis - 수면 패턴 분석 화면에 대한 클래스			
자체 구현 메소드			
이름	매개변수	반환값	용도
executeClient()	-	문자열, 성공시 null 반환	클라이언트에서 실제 서버로 전송하는 메소드

SleepingTip - 숙면 Tip 화면에 대한 클래스			
(자료 가져올 곳: 대한수면학회 http://www.sleepmed.or.kr/)			
숙면 Tip은 텍스트 뷰에 해당 내용을 넣는 방식으로 제공할 예정.			

URLConnection - http 접속에 활용되는 클래스

자체 구현 메소드

이름	매개변수	반환값	용도
run	-	void	URL 접속 실행
getResult	-	String	결과값 수신
request	String urlStr	String	http 요청 전송



MPAndroidChart 라이브러리를 이용한 차트 생성 흐름도

5. Prototype 구현

- 구현 진행도

항목	세부 항목	진행률
기본	공간 구현	미구현
	회로도 구현	구현
	서버 - 어플리케이션 통신	부분구현
원격제어	전등 제어 (on/off)	구현
	전등 타이머	부분구현
자동소등	센서 기능 설정	구현
	수면 기준 설정	부분구현
방범기능	기능 시간 중 센서 감지 시 점등	미구현
	감지 시 휴대전화로 메시지 송신	미구현
수면관리	자동 소등의 기능 시간 중 뒤척임 로그 저장	미구현
	로그를 그래프화 및 어플리케이션에 출력	미구현
	서버 동기화	미구현
	서버에서 사용자의 수면 패턴 그래프 평균화	미구현

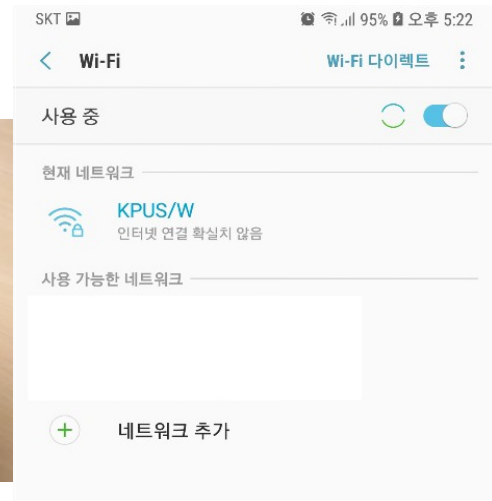
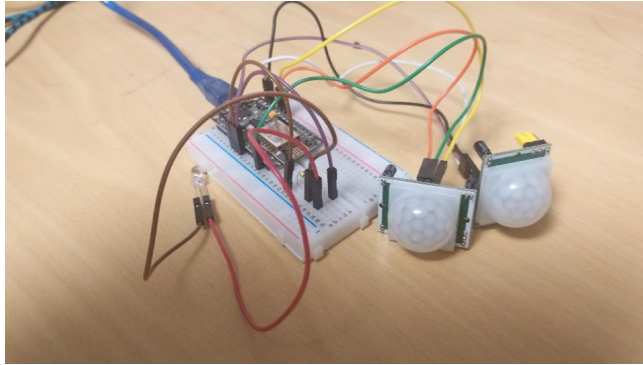
- 프로토타입 시행 시나리오

- 원격제어 기능

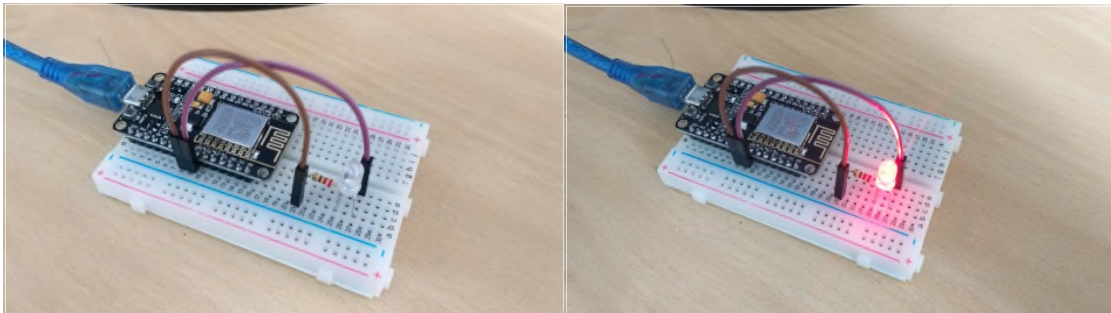
1. WiFi 확인 (ssid : KPUS/W pw : 1234567890)
2. 어플리케이션의 홈 -> 스위치 제어 메뉴 진입
3. 전등의 스위치 조작 (ON / OFF)
4. 전등 점등(ON) 후 타이머 조작(ON)
5. 일정시간(5초) 후 소등되는 것을 확인. (계속)

- 자동소등 기능

6. 전등 점등(ON)
7. 센서의 스위치 조작 (ON)
8. 움직임이 없으면 일정시간 후(5초) 소등 되는 것을 확인.
9. 전등 점등(ON)
10. 센서 앞에서 손을 흔들었다가 멈춤.
11. 센서에서 입력 값이 일정시간(약 10초) 동안 없어 소등되는 것을 확인.
(센서에서 움직임이 감지 될 경우 HIGH 신호를 약 5초간 유지한다는 것을 알림.)



하드웨어 및 WiFi



전등 원격 조작 테스트를 위해 간략화한 모습

가상의 방과 사람(환경)을 구현하지 못한 상태로 기본적인 기능 테스트만 할 수 있어 환경에 따른 변수는 없었으며 핵심 기능인 수면 패턴 관련 기능은 구현이 안 되어 아두이노 서버 기능 동작 및 어플리케이션과의 통신 여부 테스트, 전등 및 센서의 스위치 조작 등의 기초적인 기능만 구현 및 테스트가 가능했다.

1) 홈



2) 스위치 제어



테스트 한 어플리케이션 UI

6. 시험 / 테스트 결과

번호	테스트 내용	결과	기타
1	아두이노 서버 - 모바일 간 연결	양호	연결은 양호하나 어플리케이션의 조작이 가끔씩 썩히는 경우가 있음. 통신상의 에러인 것으로 보임.
2	시간 동기화	양호	서버의 초기 시간은 2020-1-1 00:00:00, 서버 작동 시 초기에 동기화 필요
3	전등 조작	양호	
4	타이머 조작	양호	기본 값은 5초, 시간 설정 가능
5	방법 센서 조작	양호	
6	방법 감지	양호	전등 깜빡임 3번 후 푸쉬 알람
7	푸쉬 알람	양호	WirePusher 앱 설치 및 키 값 필요, 키 값을 아두이노 서버에 설정하는 기능 미구현, 기본 키 값은 시연 예정 중인 핸드폰 키 값.
8	소등 센서 조작	양호	
9	자동 소등	양호	센서 HIGH 유지시간은 5초로 센서에 감지가 된 적이 있다면 꺼지는 시간이 조건 보다 5초 더 길어짐을 유의
10	타임 로그 추가	부분 양호	타임 로그가 추가 되는 것은 맞으나 센서의 HIGH 값이 단일이 아닌 5초간 유지되서인지 단시간 내에 여러번이 추가되는 현상이 벌어짐.

11	수면 패턴 분석 기능	부분 양호	1시간 동안 뒤척임이 없는 경우 data가 안 들어가 다음 뒤척임이 있는 시간의 값으로 꺾은선이 이어져 1시간 동안 뒤척임이 없는 경우 0을 기입하거나 자료 값이 없을 경우 0을 찍는 그래프 속성의 존재 여부를 조사 할 필요가 있음.
----	-------------	-------	--

7. Coding & DEMO

- 주요 CODE

아두이노

```
void controlpage(){ //control form - led, timer, sensor, security

    String message = "";
    message += "<html>";
    message += "<body>";

    message += "<FORM method=\"get\" action=\"/control.cgi\">";
    message += "<P><INPUT type=\"radio\" name=\"LightStatus\" value=\"1\">Light ON";
    message += "<P><INPUT type=\"radio\" name=\"LightStatus\" value=\"0\">Light OFF";

    message += "<P><INPUT type=\"radio\" name=\"TimerStatus\" value=\"1\">Timer ON";
    message += "<P><INPUT type=\"radio\" name=\"TimerStatus\" value=\"0\">Timer OFF";

    message += "<P><INPUT type=\"radio\" name=\"SensorStatus\" value=\"1\">Sensor ON";
    message += "<P><INPUT type=\"radio\" name=\"SensorStatus\" value=\"0\">Sensor OFF";

    message += "<P><INPUT type=\"radio\" name=\"SecurityStatus\" value=\"1\">Security ON";
    message += "<P><INPUT type=\"radio\" name=\"SecurityStatus\" value=\"0\">Security OFF";
    message += "<P><INPUT type=\"submit\" value=\"Submit\"> </FORM>";

    message += "</body>";
    message += "</html>";

    server.send(200, "text/html", message);
}
```

스위치 조작 페이지 구현(HTML)

각 스위치의 value를 FORM을 통해 submit하고, 그 데이터를 control.cgi로 보내는 코드.

```

void controlcgi(){ //control cgi - led, timer, sensor, security

if(server.argName(0) == "LightStatus"){
    int state = server.arg(0).toInt();

    Light_state = state;
    digitalWrite(Light, Light_state);

    String message = "";
    message += "<html>";
    message += "<body>";
    message += "Currently Light is ";
    message += (Light_state ? "1" : "0");
    message += "</body>";
    message += "</html>";

    server.send(200, "text/html", message);
}

```

스위치 조작 프로그램 구현(CGI) - 전등

get으로 받은 스위치 조작 데이터를 분류해 value값에 따라 센서 및 전등 조작.
위의 코드는 전등의 value값을 받아 전등을 조작하는 코드.

```

if (!client.connect("wirepusher.com", 80)){ //Needs to download WirePusher(APP) - id
    Serial.println("connection failed");
    return;
}
client.print(String("GET ") + "/send?id=E49JmpkX3&title=Alert!&message=Invader Detected&type=Default" + " HTTP/1.1\r\n" +
    "Host: " + "wirepusher.com" + "\r\n" +
    "Connection: close\r\n\r\n");
client.stop();
digitalWrite(Light, Light_state);

```

모바일 푸시 알람 구현

WirePusher이라는 앱을 받아 기기의 ID값을 이용해 모바일 푸시 알람 구현한 코드.
HTTP GET을 통해 앱을 다운 받은 기기의 ID값, 제목과 내용을 포함한 url에 접속하면 푸시 알람 전송.

어플리케이션

```
light.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
        String lightStatus;  
        if(light.isChecked()) {  
            lightStatus = 'control.cgi?LightStatus=1';  
            Toast.makeText(context: SwitchControl.this, text: '전등 스위치 ON', Toast.LENGTH_SHORT).show();  
        }else {  
            lightStatus = 'control.cgi?LightStatus=0';  
            Toast.makeText(context: SwitchControl.this, text: '전등 스위치 OFF', Toast.LENGTH_SHORT).show();  
        }  
        saveStatus();  
  
        String serverAddress = '192.168.4.1'; //아두이노 서버 주소 (포트:80)  
        HttpRequestTask requestTask = new HttpRequestTask(serverAddress);  
        requestTask.execute(lightStatus);  
    }  
});
```

스위치 조작 코드 (전등)

어플리케이션의 스위치 조작 시 대응 되는 기능 및 value를 포함한 url에 HTTP Request를 해서 아두이노 조작.

```

private String request(String urlStr) {
    StringBuilder output = new StringBuilder();
    try {
        java.net.URL url = new URL(urlStr);
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        if (conn != null) {
            conn.setConnectTimeout(10000);
            conn.setRequestMethod("GET");
            conn.setDoInput(true);
            conn.setDoOutput(true);

            int resCode = conn.getResponseCode();
            if (resCode == HttpURLConnection.HTTP_OK) {
                BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
                String line = null;
                while(true) {
                    line = reader.readLine();
                    if (line == null) {
                        break;
                    }
                    output.append(line + "\n");
                }

                reader.close();
                conn.disconnect();
            }
        }
    } catch (Exception ex) {
        //Log.e("SampleHTTP", "Exception in processing response.", ex);
        ex.printStackTrace();
    }

    return output.toString();
}

```

HTML을 읽어 String으로 반환하는 코드(선언부분)

URL에 접속해 버퍼에 URL의 내용을 저장, 버퍼의 내용을 String에 저장해 반환하는 코드.

```

URLConnection url = new URLConnector("http://192.168.4.1/");
url.start();

```

```

try{
    url.join();
}
catch (InterruptedException e){

}

```

```

String result = url.getResult();

```

HTML을 읽어 String으로 반환하는 코드(실행부분)

URL Connector()의 매개변수는 내용을 저장하려는 페이지의 URL로 위의 String request를 포함한 코드를 거쳐 String result에 저장됨.

//차트 작성

```
lineChart = (LineChart)findViewById(R.id.line_chart);
```

//차트 데이터 집합 설정 및 그래프 설정

```
LineDataSet lineDataSet = new LineDataSet(entries, label: '시간 당 뒤척인 횟수');  
lineDataSet.setLineWidth(2);  
lineDataSet.setCircleRadius(6);  
lineDataSet.setCircleColor(Color.parseColor(colorString: '#FFA1B4DC'));  
lineDataSet.setColor(Color.parseColor(colorString: '#FFA1B4DC'));  
lineDataSet.setDrawCircleHole(true);  
lineDataSet.setDrawCircles(true);  
lineDataSet.setDrawHorizontalHighlightIndicator(false);  
lineDataSet.setDrawHighlightIndicators(false);  
lineDataSet.setDrawValues(false);
```

```
LineData lineData = new LineData(labels, lineDataSet);  
lineChart.setData(lineData);
```

//축 설정

```
XAxis xAxis = lineChart.getXAxis();  
xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);  
xAxis.setTextColor(Color.BLACK);  
xAxis.enableGridDashedLine(lineLength: 8, spaceLength: 24, phase: 0);
```

```
YAxis yLAxis = lineChart.getAxisLeft();  
yLAxis.setTextColor(Color.BLACK);
```

```
YAxis yRAxis = lineChart.getAxisRight();  
yRAxis.setDrawLabels(false);  
yRAxis.setDrawAxisLine(false);  
yRAxis.setDrawGridLines(false);
```

//차트 그리기

```
lineChart.setDoubleTapToZoomEnabled(false);  
lineChart.setDrawGridBackground(false);  
lineChart.invalidate();
```

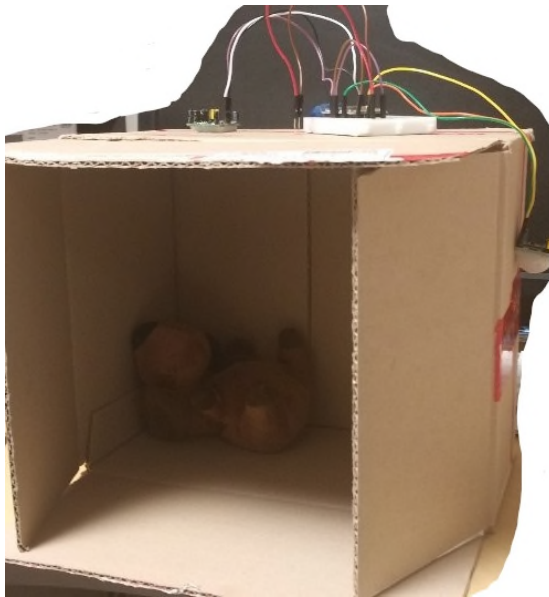
데이터를 입력받아 그래프를 그리는 코드

arraylist entries에 x값과 y값을 담아둔 후 LineDataSet의 매개변수로 그래프 작성. linedataset으로 그래프 정의 후 x축과 y축의 속성 정의. 그 후 그래프 출력.

- 구현 진행도

항목	세부 항목	진행률
기본	공간 구현	부분구현
	회로도 구현	구현
	서버 - 어플리케이션 통신	부분구현
원격제어	전등 제어 (on/off)	구현
	전등 타이머	구현
자동소등	센서 기능 설정	구현
	수면 기준 설정	부분구현
방범기능	기능 시간 중 센서 감지 시 점등	구현
	감지 시 휴대전화로 푸쉬알림 송신	구현
수면관리	자동 소등의 기능 시간 중 뒤척임 로그 저장	구현
	로그를 그래프화 및 어플리케이션에 출력	구현
	서버 동기화	미구현
	서버에서 사용자의 수면 패턴 그래프 평균화	미구현

- DEMO



전체적인 구성



내부 구성(점등 중)

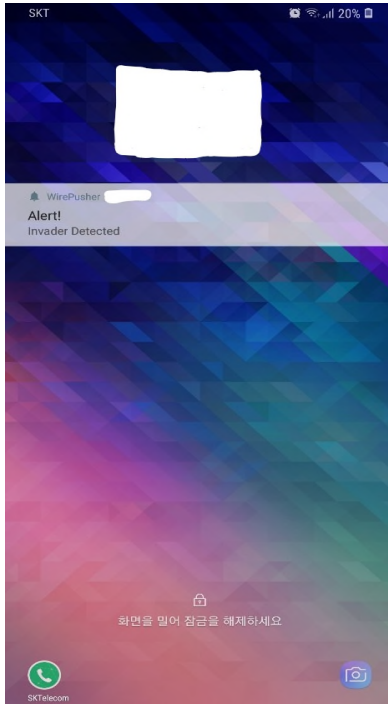
상자를 방으로 가정한 축소 환경 구성. 위에는 회로도와 전등 및 자동 소등 기능을 위한 센서가, 오른쪽 바깥에는 외부로부터의 침입감지를 위한 방법기능용 센서가 있다. 내부를 보면 사람 대신 인형이 대체 하고 있으며 천장에 센서와 전등이 보이는 것을 알 수 있다.



개선 된 스위치 제어 어플리케이션 화면 UI.

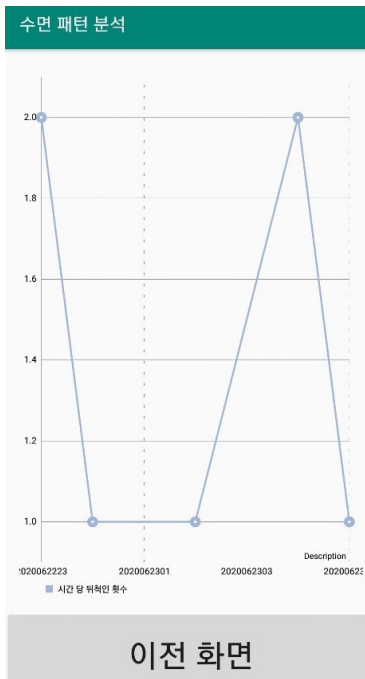
크기 조절 및 타이머 스위치를 추가해 전등 스위치 밑에 추가 하였다. 프로토타입때 방법기능 부분구현이었으나 방법 센서 감지시 전등이 3번 깜빡이고 핸드폰에 푸쉬 알람이 가도록 구현하였다.

- 어플리케이션 스위치 제어 화면 UI



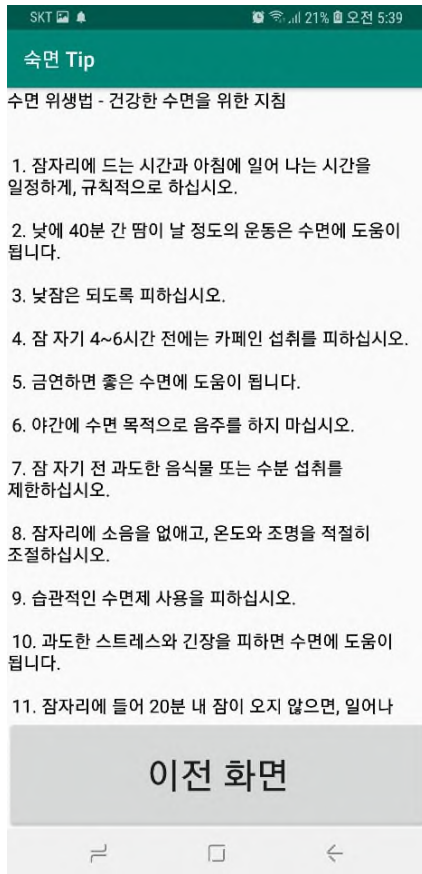
방법 기능을 켜 상태에서 움직임 감지 시 핸드폰으로 알람이 간 화면. 방법 기능 푸쉬 알람을 받으려면 WirePusher라는 어플리케이션 설치 후 생성된 ID값을 아두이노 웹 서버에 입력해야 한다.

- 방법 기능 푸쉬 알람이 도착한 모바일 화면



2020년 6월 22일 23시부터 다음날 6시까지의 뒤척임을 감지한 로그파일을 꺾은선 그래프로 시각화한 수면 패턴 분석 기능. 중간에 한번도 안 뒤척인 시간은 자료 입력이 안되어 0에 찍히지 않고 다음 입력 데이터로 선이 이어져있다. 이 부분은 뒤척이지 않은 시각은 0을 입력하는 것이 필요해보인다.

- 수면 패턴 분석 기능(어플리케이션)



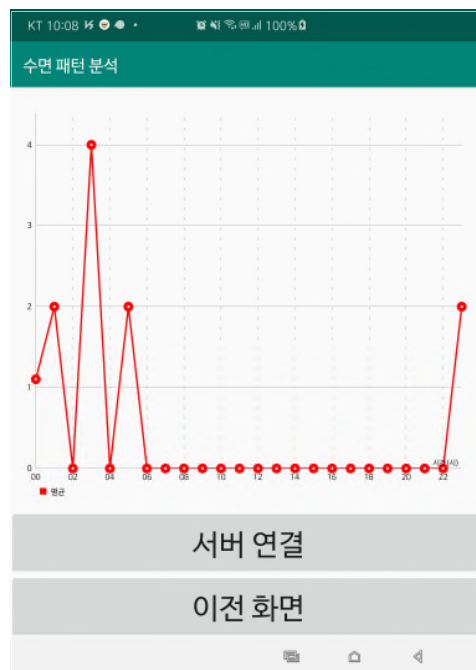
수면에 잘 들지 못하는 사용자를 위한 숙면에 관한 짝막한 Tip들을 텍스트로 출력. 출처는 대한수면학회.

- 숙면 Tip 어플리케이션 화면

Ⅲ. 결론

1. 연구결과

위의 DEMO와 구현 진행도에서의 기능을 모두 구현 완료 하였으며 아래는 전에 구현 못 한 서버에서 사용자의 수면 패턴 그래프 평균화 장면이다. 여러 사용자는 서버 연결을 통해 앞서 아두이노에 저장 된 자신의 수면데이터를 외부의 서버에 저장하고 서버는 사용자들의 시간별 평균 데이터를 반환해 사용자는 자신이 다른 사람에 비해 얼마나 뒤쳐있는건지 자가검사를 할 수 있다.



2. 작품제작 소요재료 목록

HW 부품		갯 수
	Node MCU v1.0 (아두이노 보드)	1
	PIR sensor (적외선 센서)	2
	LED 전등	1
	브레드보드 400홀	1
	저항 220옴	1
	점퍼케이블 암-수 20cm	8
	점퍼케이블 수-수 20cm	2

참고자료

참고 서적 - 사물인터넷을 위한 ESP8266 프로그래밍
그림으로 쉽게 설명하는 안드로이드 프로그래밍 개정4판 (천인국 저)

참고한 사이트

http://webs.co.kr/index.php?mid=http&document_srl=365338
<https://twinparadox.tistory.com/587>
<http://blog.naver.com/PostView.nhn?blogId=boxer61kg&logNo=150141157618>
http://www.eduino.kr/product/detail.html?product_no=239&cate_no=24&display_group=1
<https://wonsohana.wordpress.com/2018/05/18/android-텍스트-파일-외부-또는-내부에-저장하기/>
https://m.blog.naver.com/alice_k106/220504885730
<https://yuja-kong.tistory.com/26>
<https://nowonbun.tistory.com/267>
<https://nota.tistory.com/50>
<https://hyeonstorage.tistory.com/205>
<https://coding-factory.tistory.com/66>
<https://github.com/PhilJay/MPAndroidChart/wiki/Interaction-with-the-Chart>
<https://blog.naver.com/beodeulpiri/220737193210>
<https://namget.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-MPAndroidChart-LineChart-%EC%86%8D%EC%84%B1-%ECAA%95%EB%A6%AC-Example>
(<https://m.blog.naver.com/PostView.nhn?blogId=greatsk553&logNo=220800324470>)
<https://under-desk.tistory.com/183>
<https://lktprogrammer.tistory.com/143>
<https://nowonbun.tistory.com/502>
<https://playground.arduino.cc/Code/Time/>
https://blog.naver.com/ann_arbor/221354618824
<https://blog.naver.com/nabileral/221956526651>
http://www.hardcopyworld.com/gnuboard5/bbs/board.php?bo_table=lecture_esp&wr_id=44
<https://steemit.com/kr/@codingart/61-nodemcu-web-server-led-on-off>
<https://blog.naver.com/smartinfo/221744905280>
<https://chocoball.tistory.com/entry/Hardware-ESP01-or-ESP8266-using-5>
<https://circuits4you.com/2019/01/05/connecting-esp32-to-wifi-network/>
<https://postpop.tistory.com/97>