

창의 SW 설계 5

17011849 박영주

17011871 하윤영

17011885 박세정

최종 보고서

초기 계획

		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 초음파 센서 연결 하윤영: 온습도 센서 연결
11주차		부품 연결	박영주: 스케치 코드 작성 박세정: 아두이노 회로 연결 하윤영: 아두이노 회로 연결
12주차		블루투스 이용하여 스마트폰 연결	박영주: 앱인벤터 제작 박세정: 블루투스연결 하윤영: 블루투스 스케치 보드 작성
13주차		스마트폰 어플 제작	박영주: 앱인벤터 제작 박세정: 앱인벤터 화면 디자인 하윤영: 앱인벤터와 아두이노 연결
14주차		실험	박영주: 스마트폰 블루투스 연결 박세정: 센서간 연결 하윤영: 스케치 보드 점검
15주차		시연 및 보완	
16주차		결과	

최종 진행 일정

		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 온습도 센서 + LCD 연결 하윤영: 초음파 센서 연결
11주차		부품 연결, 블루투스 스마트폰 연결	박영주: 블루투스 연결 및 부품 연결 스케치 보드 작성 박세정: 아두이노 회로 연결, 앱인벤터 블루투스 연결 하윤영: 아두이노 회로 연결, 릴레이 자료조사
12주차		전원 제어 실험 및 스마트폰 어플 스케치	박영주: 릴레이 모듈 연결 및 실행 박세정: 멀티탭 분리, 앱인벤터 화면 디자인 하윤영: 릴레이 모듈 실험
13주차		전원제작 완성 및 스마트폰 어플 제작	박영주: 앱인벤터 전원 제어 ON/OFF 블록 작성 박세정: 앱인벤터 화면 디자인, 앱인벤터 기본 블록 작성 하윤영: 어플에 삽입할 기능 알고리즘 작성, 합치기
14주차		제품 만들기	박영주: 제품 제작, 사용시간 계산하는 코드 작성 박세정: 제품 제작, 사용시간 앱인벤터 블록 작성 하윤영: 제품 제작, 사용량 계산법 조사
15주차		스마트폰 어플 완성	박영주: 앱인벤터로 사용량, 사용시간 출력 블록 작성 박세정: 앱인벤터로 알림 블록 작성 하윤영: 앱인벤터의 사용시간 계산법, 알림 기능 조사
16주차		결과	

제안한 시스템 및 기능



히터의 미세먼지 정도 표시



일정 거리 이상 다가올 시
위험 감지 알람



온/습도 측정



스마트폰을 통해 일별 사용량,
현재 온습도 표시



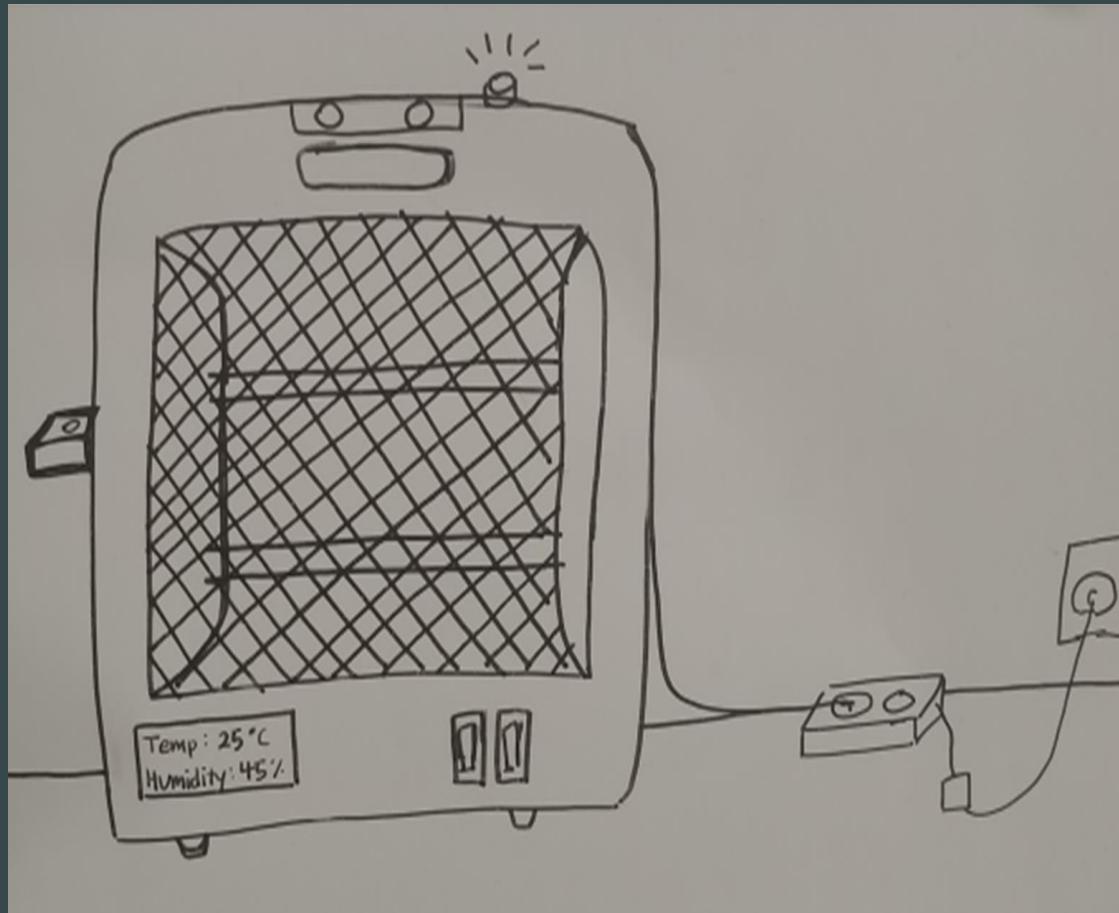
현재 온습도, 세기를
디스플레이로
표시



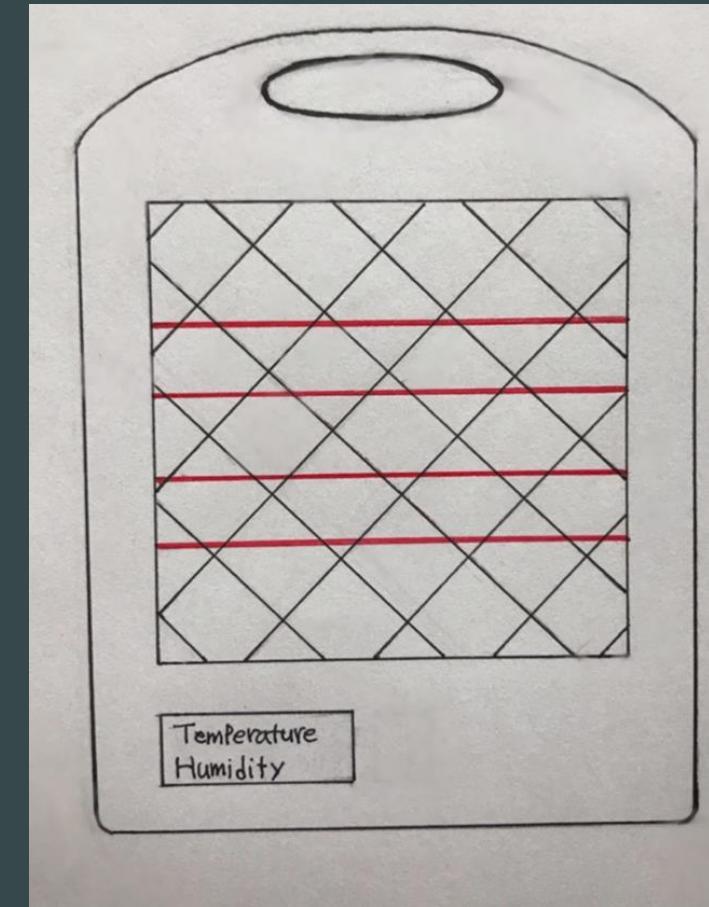
스마트폰을 이용한
원격 조정

제안한 시스템 및 기능

초기 스케치

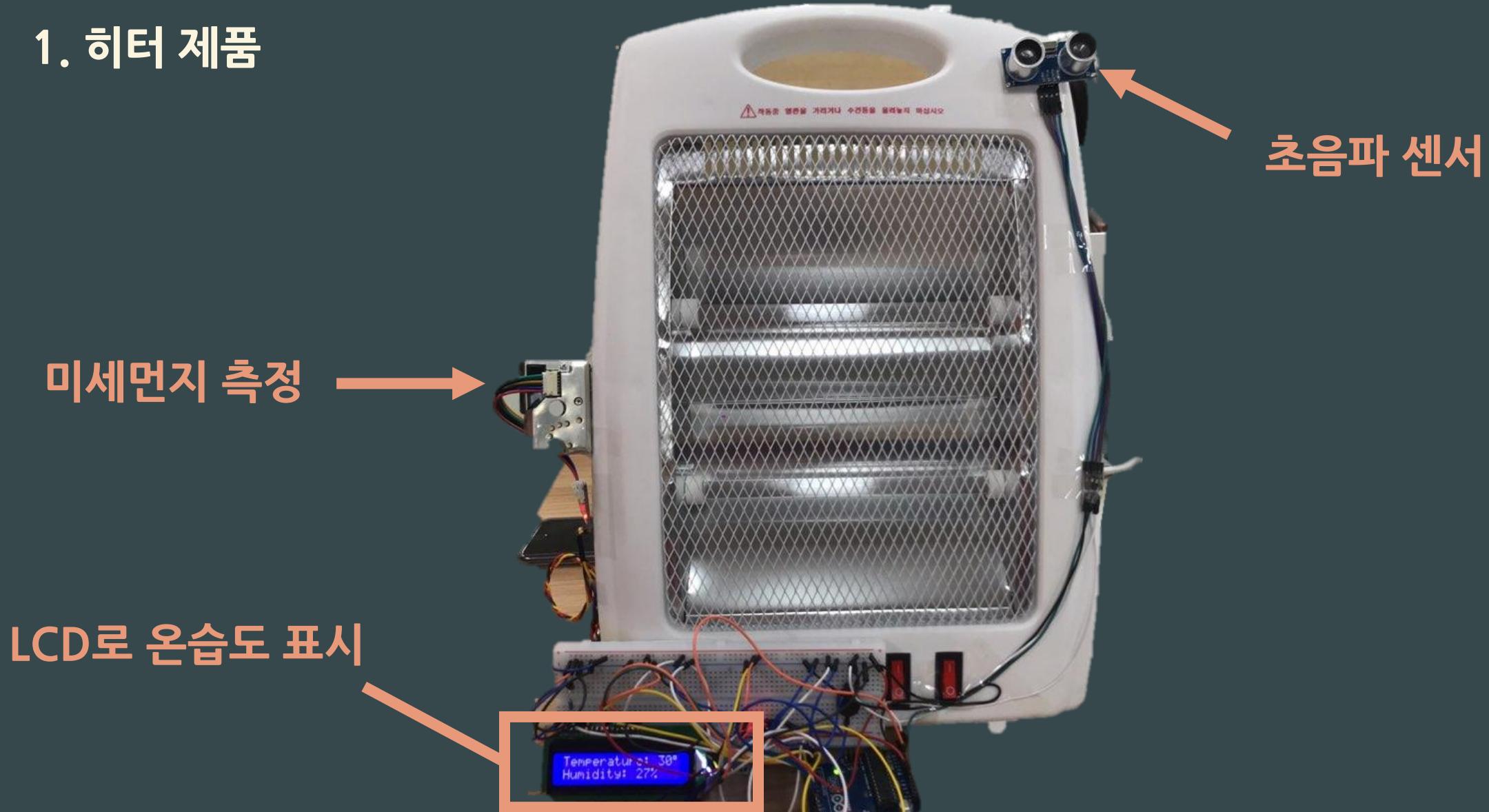


예상 제품 스케치



제안한 시스템 및 기능

1. 히터 제품



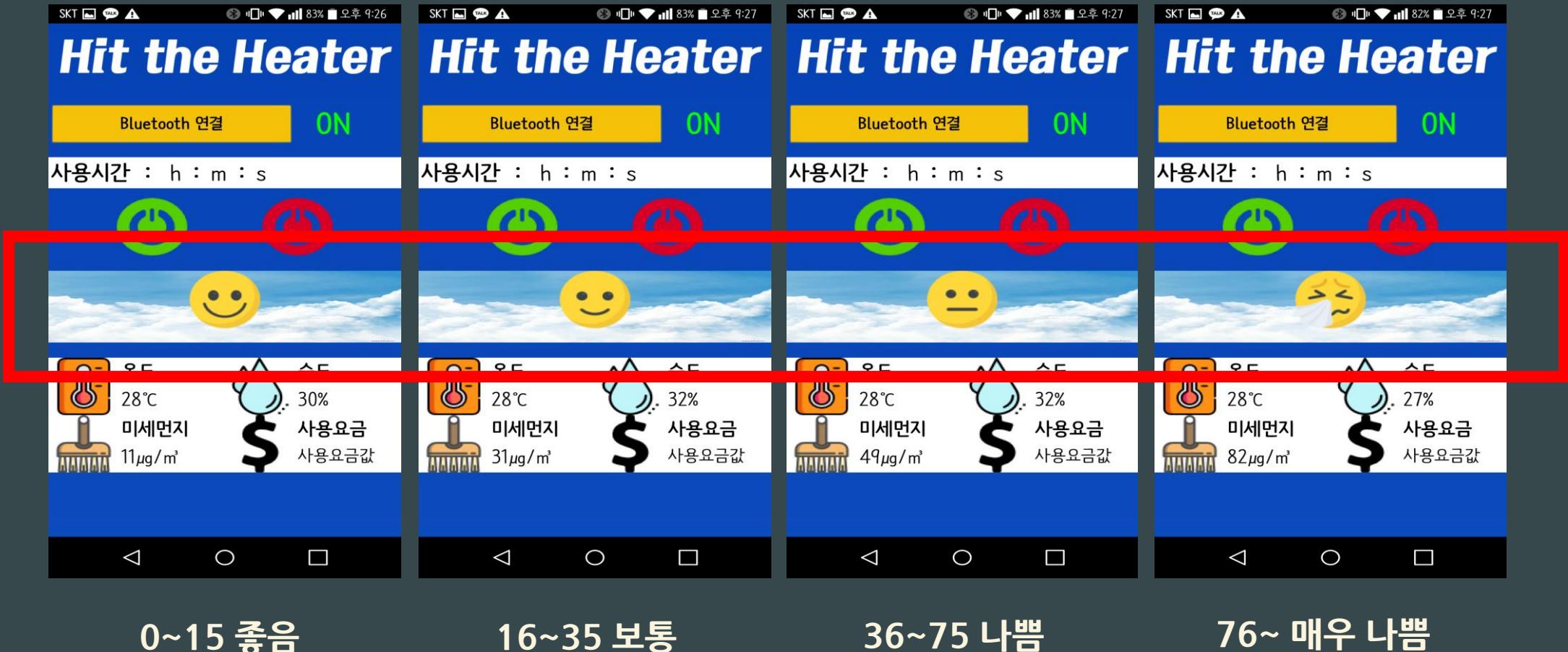
제안한 시스템 및 기능

2. 어플리케이션 구현



제안한 시스템 및 기능

3. 미세먼지 기준



청소 및 환기 요망

제안한 시스템 및 기능

4. 사용자 알림 효과



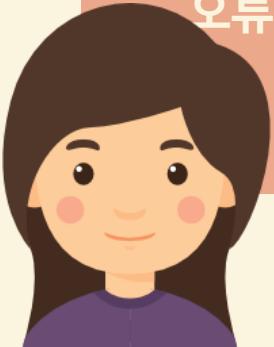
사용량 200을 초과하여
전력량요금이 변경되었을 경우
진동으로 알람

히터와 10cm 이내로
5초 이상 유지 되어 있을 경우
위험 알람

팀원별 구체적인 역할 및 기여점

미세먼지 센서 연결, 블루투스 연결, 릴레이모듈 코드 작성, 제품 제작, 전기사용량 계산법(c언어로 구현), 사용시간 출력코드, 최종 아두이노 스케치 코드 작성, 오류 코드 수정, 앱인벤터-블루투스 제어, ON/OFF 제어, 리스트 설정, 사용시간 누적 및 타이머 계산 등

기여점 - 아두이노 스케치 코드, 아두이노와 앱인벤터 연결, 전체적인 오류 해결 위주의 역할 수행



17011849
박영주

초음파 센서 연결, 아두이노 회로 연결, 릴레이모듈 시연 및 자료조사, 어플 알고리즘 작성 및 코드 합치기, 제품 제작, 저온화상 및 안전거리, 미세먼지 기준치 조사, 전기사용량, 사용요금 조사, 아이디어, 앱인벤터 - 진동알림, 아두이노 오류수정 등

기여점 - 아두이노와 어플에 관련된 자료조사 및 아두이노 회로 오류 해결 위주의 역할 수행



17011871
하윤영

온습도 센서, LCD 연결, 아두이노 회로 최종 연결, 멀티탭 분리, 어플 구성, 앱인벤터 - 화면디자인, 블루투스제어, 화면 블록 연결, 사용요금계산, 미세먼지 표정 변화, 사용량 초과 시 진동 알림 설정, 출력값 분리하기, 변수 설정, 오류 수정 등

기여점 - 아두이노 센서 구성, 회로 연결 및 앱인벤터 화면디자인과 블록 설정 위주의 역할



17011885
박세정

유사 제품명 및 사이트 주소

<lot 원격제어 스마트 플러그 전기히터>/ 스마트플러그+히터 제품

http://shopping.interpark.com/product/productInfo.do?prdNo=6052932293&utm_source=google&utm_medium=remarketing&utm_campaign=incorp_DAP_20180131&gclid=EAIalQobChMlpuSiZGp3wIVTraWCh1CWgkwEAQYASABEgKat_D_BwE

참고문헌 및 사이트 주소

<초음파센서+부저센서 참고 사이트>

https://blog.naver.com/sumokto_2002/221308994547

<미세먼지 센서 참고 사이트>

<https://m.blog.naver.com/dnqlslaka/221125802242>

<http://www.makewith.co/page/project/989/story/2321/>

<LCD+온습도 참고 사이트>

<https://blog.naver.com/dsz08082/221321020369>

<https://blog.naver.com/dokkosam/221172166556>

<릴레이를 이용한 전원 제어 참고 사이트>

deneb21.tistory.com/m/222

<https://blog.naver.com/codingspecialist/221039892975>

www.makewith.co/page/project/309/story/630

<https://blog.naver.com/codingspecialist/221039900151>

<https://blog.naver.com/compass1111/221288414858>

<https://cafe.naver.com/arduinoguide/597>

<https://blog.naver.com/compass1111/221288414858>

참고문헌 및 사이트 주소

〈전기사용료〉

<http://cyber.kepco.co.kr/ckeprco/front/jsp/CY/E/E/CYEEHP00101.jsp>

<http://cyber.kepco.co.kr/ckeprco/front/jsp/CY/J/A/CYJAPP000.jsp#>

〈앱인벤터-블루투스 연결〉

<https://blog.naver.com/icbanq/221006800539>

<http://blog.naver.com/PostView.nhn?blogId=rhkd824&logNo=220542365789&parentCategoryNo=&categoryNo=&viewDate=&isShowPopularPosts=false&from=postView>

〈앱인벤터-센서값 받기〉

<https://blog.naver.com/milly0304/221284602786>

<https://blog.naver.com/nasu0210/220684394910>

<https://m.blog.naver.com/PostView.nhn?blogId=roboholic84&logNo=220489586458&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F>

<http://www.iamamaker.kr/ko/projects/앱-인벤터로-미세먼지-온습도-측정기-앱-만들기/>

〈앱인벤터-진동알림〉

<https://elpissoft.blog.me/220500157325>

<https://puravidaapps.com/extensions.php>

참고문헌 및 사이트 주소

<미세먼지 수치 참고자료-기준>

<https://terms.naver.com/entry.nhn?docId=5570493&cid=60371&categoryId=60371>

<저온화상 참고자료-기준>

<https://blog.naver.com/hl7060/221413768333>

<https://m.post.naver.com/viewer/postView.nhn?volumeNo=17255220&memberNo=21480402&vType=VERTICAL>

<앱인벤터-사용시간 누적 계산>

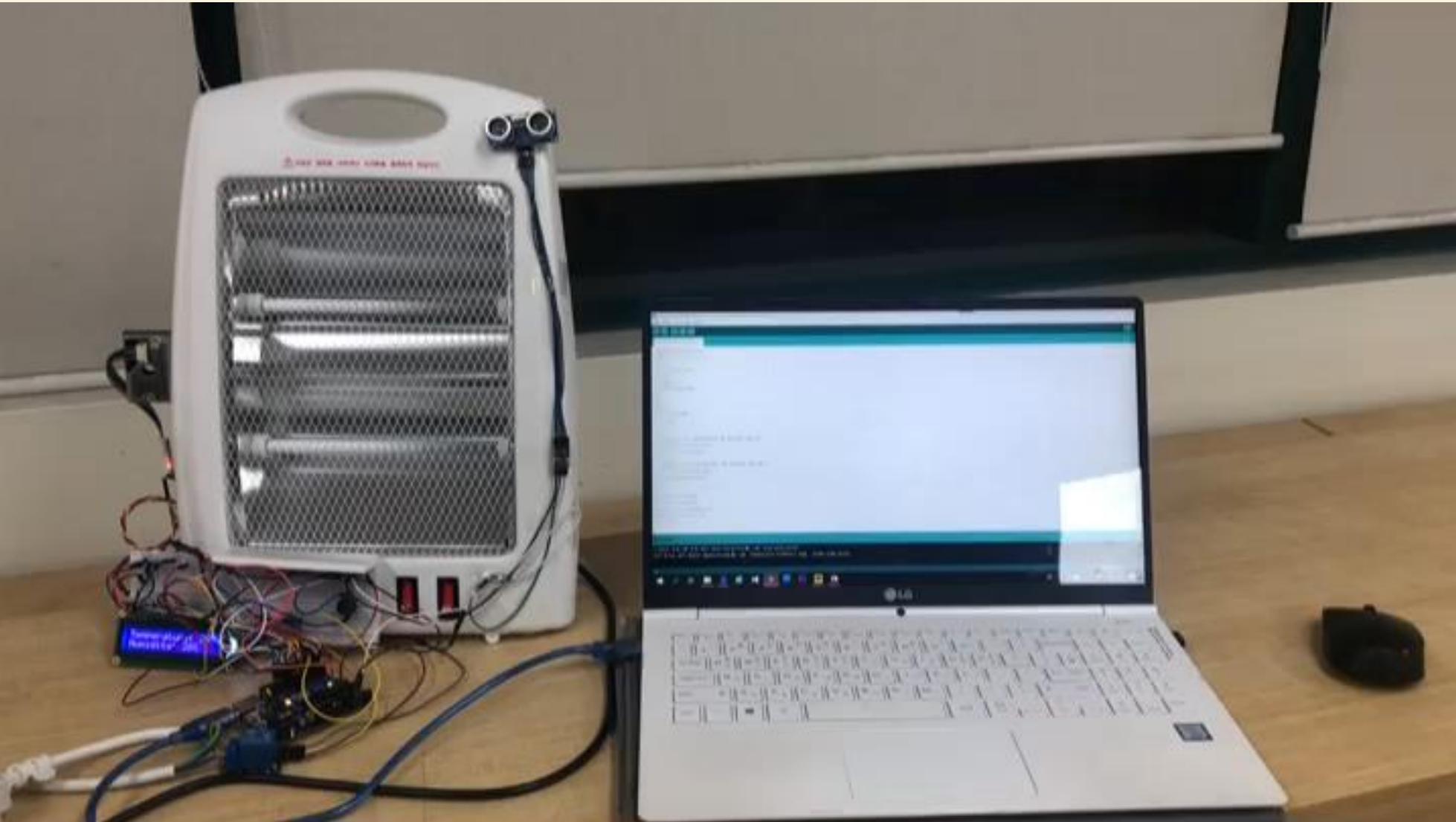
<https://aia.bizadmin.co.kr/스톱워치/>

<https://m.blog.naver.com/joon060707/221409007345>

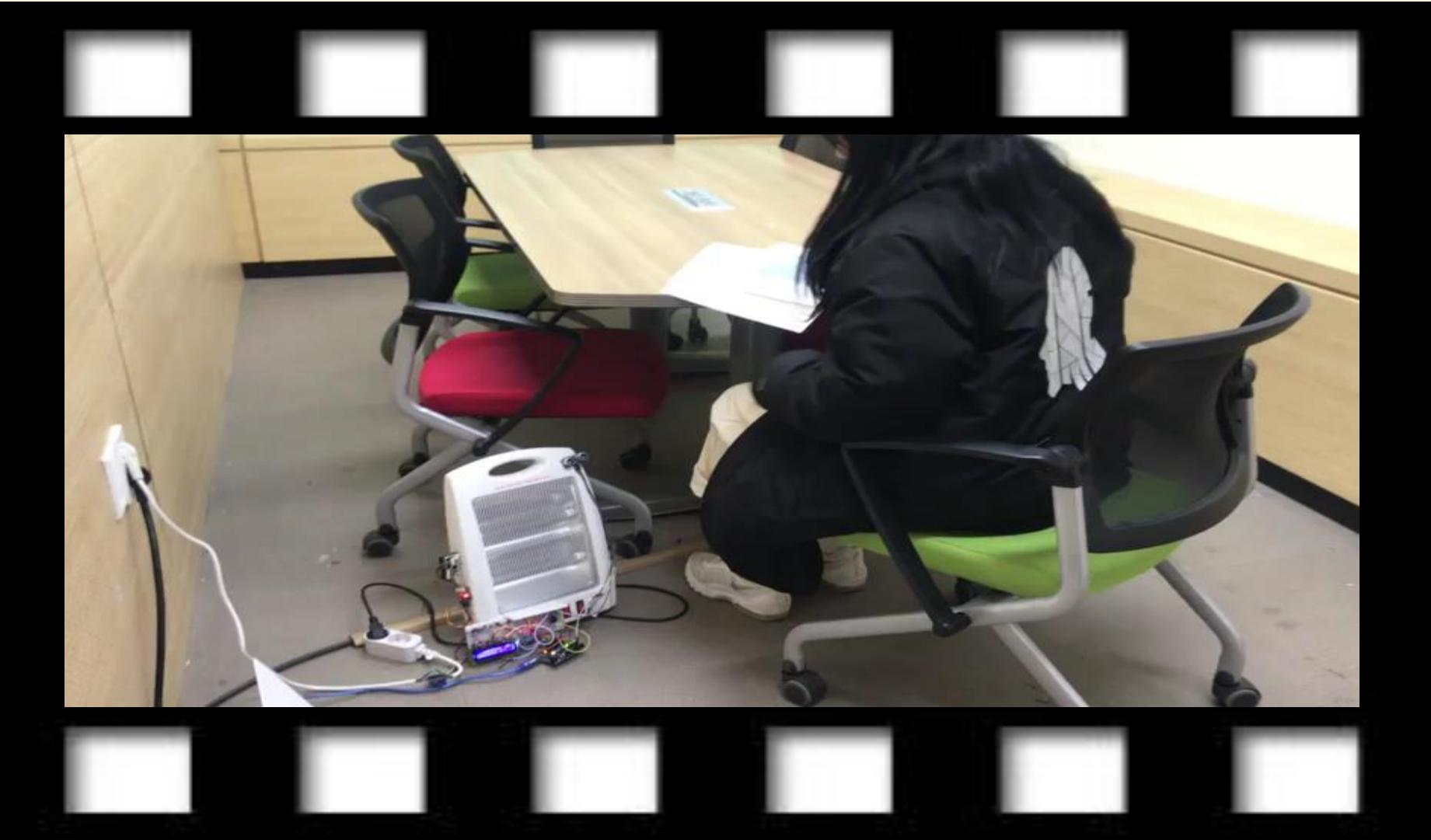
<https://blog.naver.com/milly0304/221284602786>

<https://www.arduino.cc/en/Reference/Millis>

시연동영상



시연동영상



추가 재료 구입

번호	추가 재료 목록	재료별 가격	가격 총합
1	HM-608 히터	13070	13070
2	브레드보드 830핀(MB-102)	1290	14360
3	아두이노 PM2.5 미세먼지센서+어댑터(GP2Y1014AU)	11000	25360
4	멀티 콘센트 일반형 2구	3000	28360
5	우드락 2T 10*5(cm)	100	28460

→ 최종 금액 : 28460원

진행 보고서

5 (15주차) 미발표자료

진행상황

<15주차 진행상황>

앱인벤터 완성

금액 초과 시 진동알림 설정

히터 완성

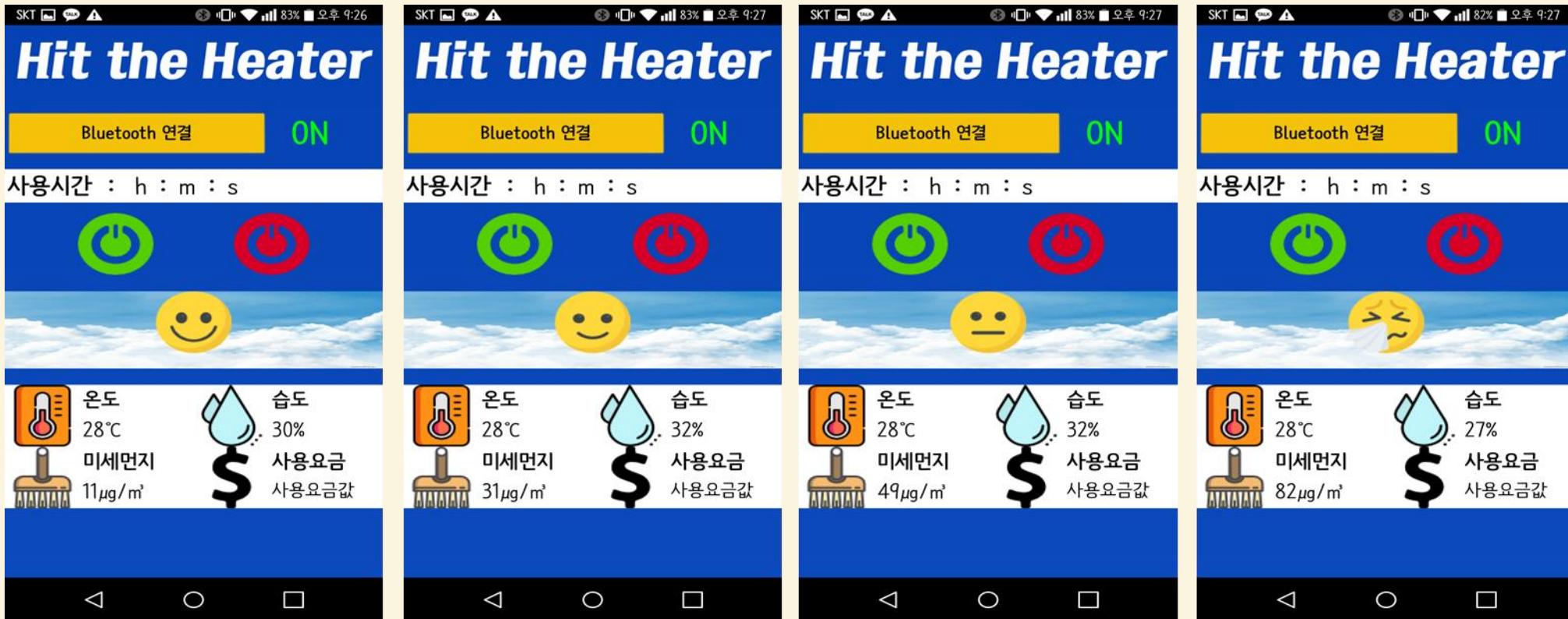
미세먼지 정도를 얼굴로 표시-> 나쁨일 경우 청소 요망

주별 설계//역할

		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 온습도 센서 + LCD 연결 하윤영: 초음파 센서 연결
11주차		부품 연결, 블루투스 스마트폰 연결	박영주: 블루투스 연결 및 부품 연결 스케치 보드 작성 박세정: 아두이노 회로 연결, 앱인벤터 블루투스 연결 하윤영: 아두이노 회로 연결, 릴레이 자료조사
12주차		전원 제어 실험 및 스마트폰 어플 스케치	박영주: 릴레이 모듈 연결 및 실행 박세정: 멀티탭 분리, 앱인벤터 화면 디자인 하윤영: 릴레이 모듈 실험
13주차		전원제작 완성 및 스마트폰 어플 제작	박영주: 앱인벤터 전원 제어 ON/OFF 블록 작성 박세정: 앱인벤터 화면 디자인, 앱인벤터 기본 블록 작성 하윤영: 어플에 삽입할 기능 알고리즘 작성, 합치기
14주차		제품 만들기	박영주: 제품 제작, 사용시간 계산하는 코드 작성 박세정: 제품 제작, 사용시간 앱인벤터 블록 작성 하윤영: 제품 제작, 사용량 계산법 조사
15주차		스마트폰 어플 완성	박영주: 앱인벤터로 사용량, 사용시간 출력 블록 작성 박세정: 앱인벤터로 알림 블록 작성 하윤영: 앱인벤터의 사용시간 계산법, 알림 기능 조사
16주차		결과	

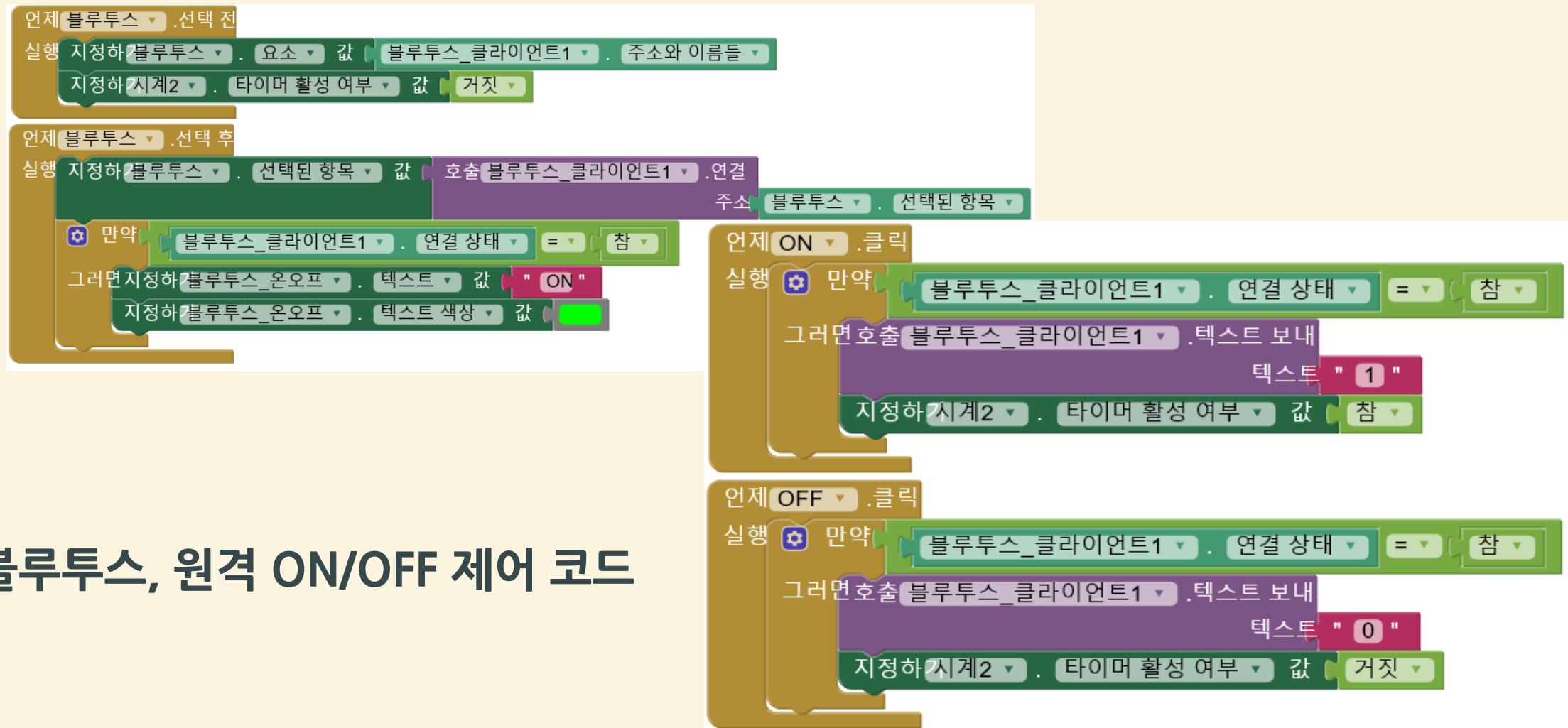
주요 설계/제작 진행상황

1. 앱인벤터 완성 화면



주요 설계/제작 진행상황

2. 앱인벤터 블록 부분

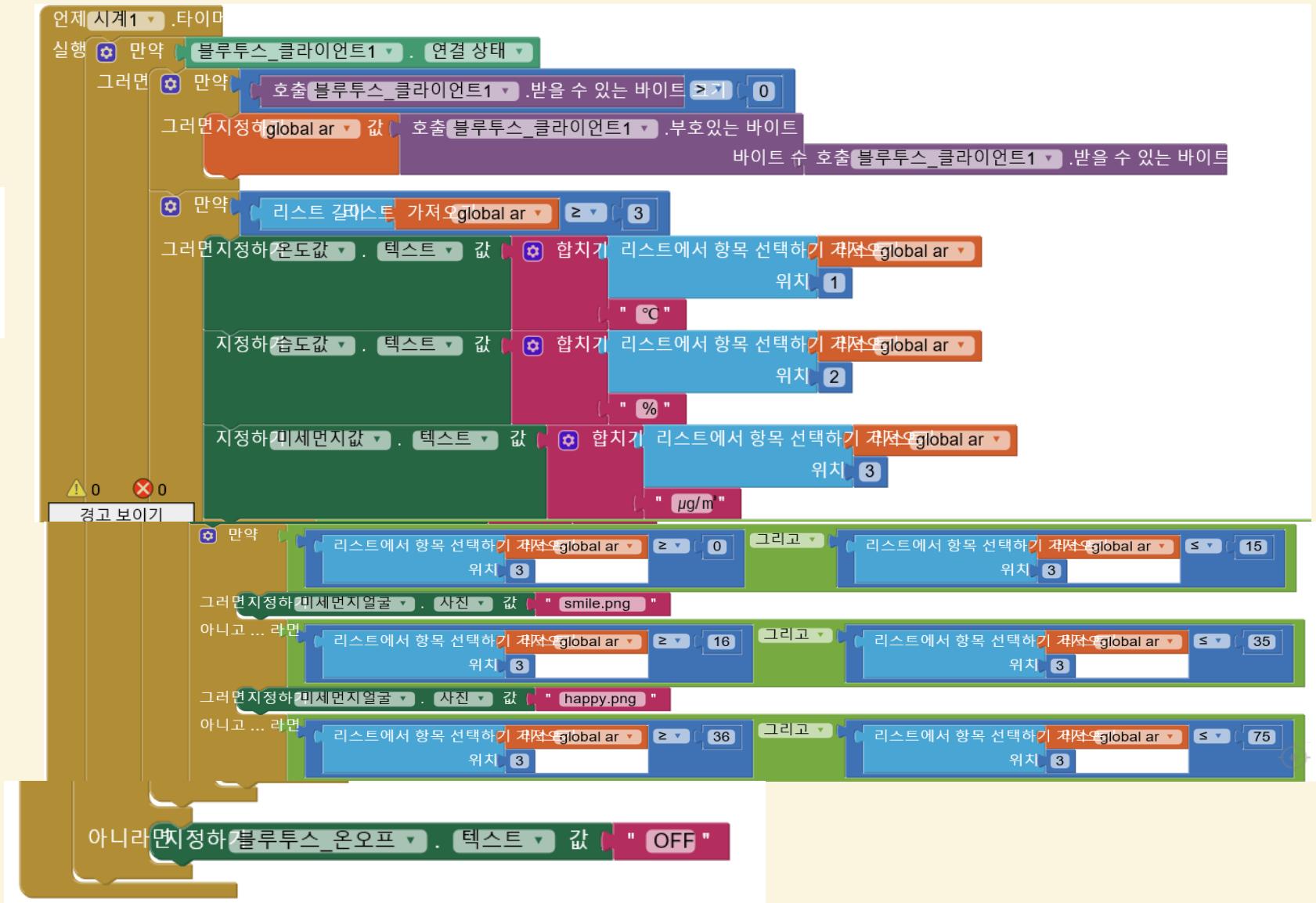


블루투스, 원격 ON/OFF 제어 코드

주요 설계/제작 진행상황

2. 앱인벤터 블록 부분

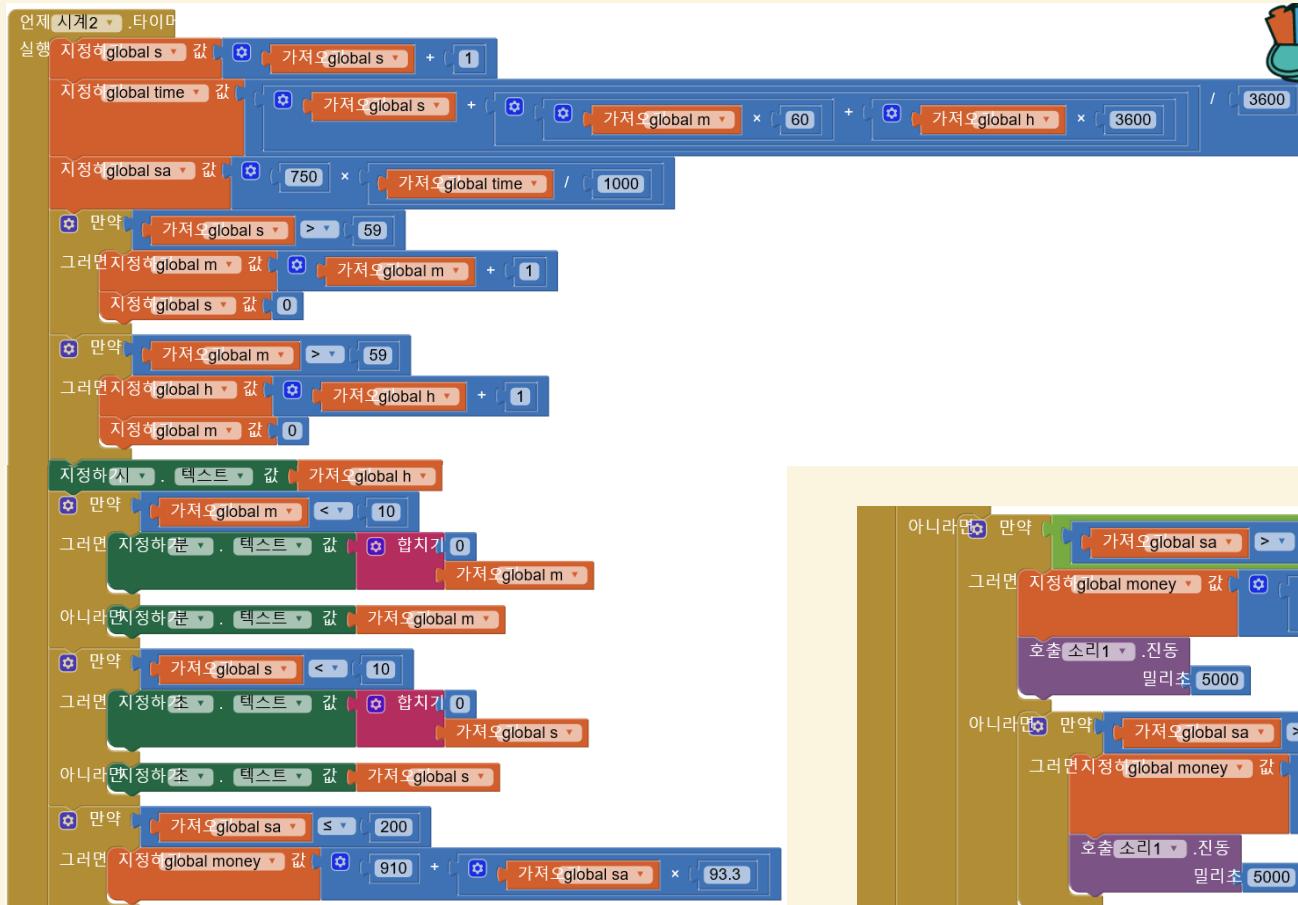
전역변수 `ar` 값 0
전역변수 `time` 값 0
전역변수 `h` 값 0
전역변수 `money` 값 0
전역변수 `m` 값 0
전역변수 `sa` 값 0
전역변수 `s` 값 0



센서 값 받아오기/
미세먼지 표정 변화 설정

주요 설계/제작 진행상황

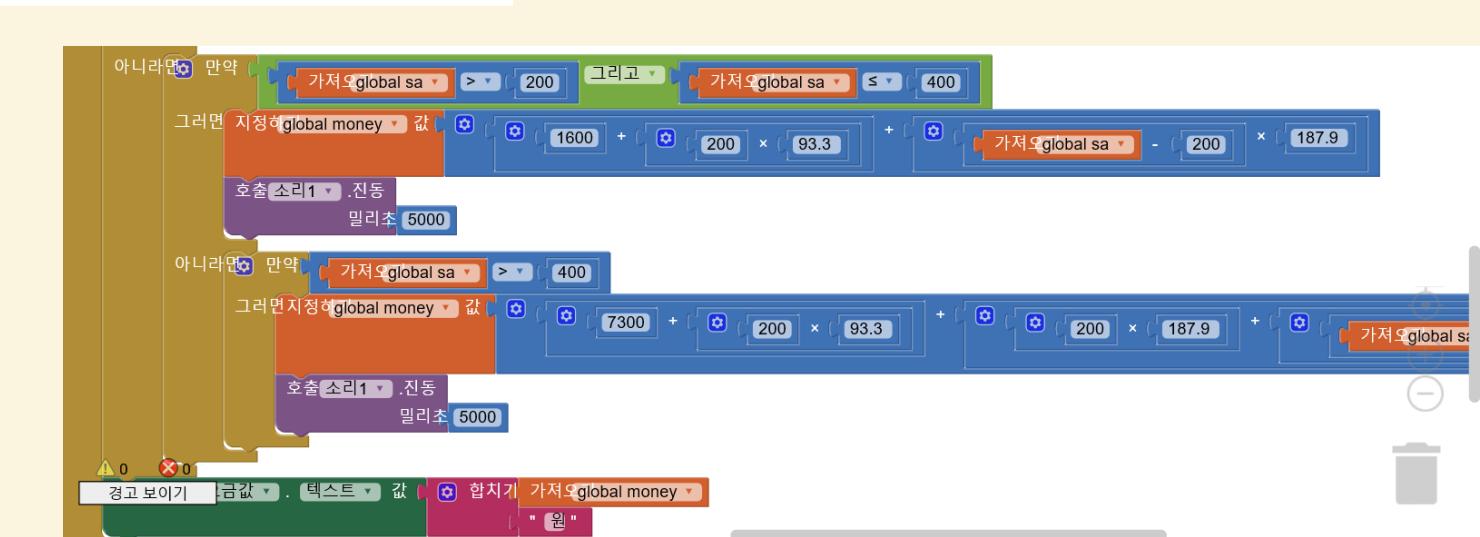
2. 앱인벤터 블록 부분



사용시간 출력
사용시간 누적

사용요금 계산

사용량 초과시 진동 알람



주요 설계/제작 진행상황

3. 아두이노 최종 스케치 코드

```
#include <DHT.h>
#include <DHT_U.h>
#include <LiquidCrystal.h>
#include <SoftwareSerial.h> //시리얼통신 라이브러리 호출
SoftwareSerial btSerial(5,4); //PX,TX시리얼 통신을 위한 객체선언
LiquidCrystal lcd(7,8,9,10,11,12);
String myString="";

#define DHT11_PIN 2
int Sensor_pin = 2;
DHT my_sensor(Sensor_pin,DHT11);

float value=0;
float voltage=0;
float dustDensity=0;
```

```
int Echo_Pin=15;
int Trig_Pin=16;
int ALARM=13;
int cnt=0;

int relay=6;
byte buffer[3];
```

주요 설계/제작 진행상황

3. 아두이노 최종 스케치 코드

```
void setup(){  
    lcd.begin(16,2); //lcd  
    Serial.begin(9600);  
    btSerial.begin(9600); //블루투스 시리얼  
    my_sensor.begin();  
  
    pinMode(3,OUTPUT); //미세먼지 센서  
    pinMode(A0,INPUT);  
  
    pinMode(Echo_Pin,INPUT); //초음파센서  
    pinMode(Trig_Pin,OUTPUT);  
  
    pinMode(relay,OUTPUT); //릴레이모듈  
}  
}
```

주요 설계/제작 진행상황

3. 아두이노 최종 스케치 코드

```
void loop(){
    char data=btSerial.read();
    if (Serial.available()) {
        btSerial.write(Serial.read());
    }
    if (btSerial.available()) {
        char myChar=(char)btSerial.read();
        myString+=myChar;
        delay(5);
    }
    int temp= my_sensor.readTemperature();
    int humi=my_sensor.readHumidity();
    Serial.print("Temperature: ");
    Serial.print(temp);
    Serial.print(" °C");
    Serial.print(" Humidity: ");
    Serial.print(humi);
    Serial.println(" %");
    lcd.setCursor(0,0);
    lcd.print("Temperature: ");
    lcd.print(temp);
    lcd.print((char)223);
    lcd.print("C");
    lcd.setCursor(0,1);
    lcd.print("Humidity: ");
    lcd.print(humi);
    lcd.print("%");
    digitalWrite(3,LOW);
    delayMicroseconds(10);
    value=analogRead(A0);
    delayMicroseconds(10);
    digitalWrite(3,HIGH);
    delayMicroseconds(10);
```

주요 설계/제작 진행상황

3. 아두이노 최종 스케치 코드

```
voltage=value*5.0/1024.0;  
dustDensity=(voltage)/0.005;  
Serial.print("Dust Density : ");  
Serial.print(dustDensity);  
Serial.println(" µg/m³");
```

```
long duration,distance;  
digitalWrite(Trig_Pin,HIGH);  
delayMicroseconds(10);  
digitalWrite(Trig_Pin,LOW);  
duration=pulseIn(Echo_Pin,HIGH);  
  
distance=((float)(340*duration)/10000)/2;  
  
Serial.print("Distance: ");  
Serial.print(distance);  
Serial.println(" cm");
```

주요 설계/제작 진행상황

3. 아두이노 최종 스케치 코드

```
if(distance<=10){  
    cnt++;  
    if(cnt>=5){  
        tone(ALARM,1000);  
    }  
}  
else{  
    noTone(ALARM);  
    cnt=0;  
}  
  
if(data=='1'){ //앱인벤터에서 1을 받아오면 전원 ON  
    digitalWrite(relay,HIGH);  
    delayMicroseconds(10);  
}  
if(data=='0'){ //앱인벤터에서 0을 받아오면 전원 OFF  
    digitalWrite(relay,LOW);  
    delayMicroseconds(10);  
}  
  
buffer[0]=(byte)temp;  
buffer[1]=(byte)humi;  
buffer[2]=(byte)dustDensity;  
btSerial.write(buffer,3);  
delay(2000);  
}
```

팀원 별 구체적인 역할

17011849 박영주 - 앱인벤터 사용시간 누적 계산, 스케치코드 최종 수정, 앱인벤터 센서값 분리하기

17011871 하윤영 - 제품 최종 제작, 우드락, 진동알림 자료조사, 스케치코드 수정, 히터 오류일 때 아두이노 센서 재연결

17011885 박세정 - 앱인벤터 사용요금 계산, 앱인벤터 사용량 계산법, 앱인벤터 진동 설정, 앱엔벤터 센서값 분리하기, 앱인벤터 화면디자인

참고문헌 및 사이트

〈앱인벤터-사용시간 누적 계산〉

<https://aia.bizadmin.co.kr/스톱워치/>

<https://m.blog.naver.com/joon060707/221409007345>

<https://blog.naver.com/milly0304/221284602786>

<https://www.arduino.cc/en/Reference/Millis>

진행 보고서

4 (13/14주차)

진행상황

<13/14주차 진행상황>

전기사용량 계산-> 스케치 코드에 추가

스케치코드 완성

앱인벤터 화면/블록 제작 중

(릴레이와 아두이노 연결)

제품 제작(스케치)

주별 설계//역할

		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 온습도 센서 + LCD 연결 하윤영: 초음파 센서 연결
11주차		부품 연결, 블루투스 스마트폰 연결	박영주: 블루투스 연결 및 부품 연결 스케치 보드 작성 박세정: 아두이노 회로 연결, 앱인벤터 블루투스 연결 하윤영: 아두이노 회로 연결, 릴레이 자료조사
12주차		전원 제어 실험 및 스마트폰 어플 스케치	박영주: 릴레이 모듈 연결 및 실행 박세정: 멀티탭 분리, 앱인벤터 화면 디자인 하윤영: 릴레이 모듈 실험
13주차		전원제작 완성 및 스마트폰 어플 제작	박영주: 앱인벤터 전원 제어 ON/OFF 블록 작성 박세정: 앱인벤터 화면 디자인, 앱인벤터 기본 블록 작성 하윤영: 어플에 삽입할 기능 알고리즘 작성, 합치기
14주차		제품 만들기	박영주: 제품 제작 박세정: 제품 제작 하윤영: 제품 제작
15주차		시연 및 보완	
16주차		결과	변경 전

주별 설계//역할

Task		팀원 구체적 역할
9주차	초안 작성 확정 및 부품구매	
10주차	센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 온습도 센서 + LCD 연결 하윤영: 초음파 센서 연결
11주차	부품 연결, 블루투스 스마트폰 연결	박영주: 블루투스 연결 및 부품 연결 스케치 보드 작성 박세정: 아두이노 회로 연결, 앱인벤터 블루투스 연결 하윤영: 아두이노 회로 연결, 릴레이 자료조사
12주차	전원 제어 실험 및 스마트폰 어플 스케치	박영주: 릴레이 모듈 연결 및 실행 박세정: 멀티탭 분리, 앱인벤터 화면 디자인 하윤영: 릴레이 모듈 실험
13주차	전원제작 완성 및 스마트폰 어플 제작	박영주: 앱인벤터 전원 제어 ON/OFF 블록 작성 박세정: 앱인벤터 화면 디자인, 앱인벤터 기본 블록 작성 하윤영: 어플에 삽입할 기능 알고리즘 작성, 합치기 변경후
14주차	제품 만들기	박영주: 제품 제작, 사용시간 계산하는 코드 작성 박세정: 제품 제작, 사용시간 앱인벤터 블록 작성 하윤영: 제품 제작, 사용량 계산법 조사
15주차	스마트폰 어플 완성	박영주: 앱인벤터로 사용량, 사용시간 출력 블록 작성 박세정: 앱인벤터로 알림 블록 작성 하윤영: 앱인벤터의 사용시간 계산법, 알림 기능 조사
16주차	결과	

주요 설계/제작 진행상황

1. 전기사용량 요금 계산 코드 만들기(C언어)

```
1 #include<stdio.h>
2 #pragma warning(disable:4996)
3 int main() {
4     float second,time,sa;
5     int bill,cnt1,cnt2,cnt3;
6     cnt1 = 200000 / 750;
7     cnt2 = 400000 / 750;
8     scanf("%f", &second);
9     time = second/3600;//초를 시간으로 변환
10    sa = 750 * time / 1000;
11    if (sa <= 200) {
12        bill =(int)( 910 + sa*93.3);
13    }
14    else if (sa> 200 && sa <= 400) {
15        bill = (int)(1600 + (93.3 * 750 * cnt1) / 1000 + (187.9 * 750 * (time - cnt1)) / 1000);
16    }
17    else if (sa > 400) {
18        bill= (int)(7300 + (93.3 * 750 * cnt1) / 1000 + (187.9 * 750 * cnt2)/ 1000 + (280.6*750*(time-cnt2))/1000);
19    }
20    printf("%d\n", bill);
21    printf("%f\n", sa);
22    return 0;
23 }
```

주요 설계/제작 진행상황

1. 전기사용량 요금 계산 코드 만들기

2. 조건선택

대가족요금 /
생명유지장치
요금

- 5인 이상 가구
- 출산가구
- 3자녀 이상 가구
- 생명유지장치
- 해당없음

복지할인요금

2018.11.06 ~ 2018.12.05

사용기간

9 kWh

사용량

요금계산

4. 상세계산내역

■ 월간 9kWh 사용시 전기요금 계산(주거용)

기본요금(원미만 절사) : 910원

전력량요금(원미만 절사) : 839원

필수사용량 보장공제(원미만 절사) : 749원

필수사용량 보장공제 : 749원

- 필수사용량 보장공제 : 749원 (910원 + 839원 - 4,000원 < 1,000원 (감액 후 최저
요금))

전기요금계(기본요금 + 전력량요금 - 필수사용량 보장공제)
: 910원 + 839원 - 749원 = 1,000원

부가가치세(원미만 4사 5입) : 1,000원 × 0.1 = 100원

전력산업기반기금(10원미만 절사) : 1,000원 × 0.037 = 37원

910+839=1749

C:\WINDOWS\system32\cmd.exe

43200 ← 초(second)
1749 ← 사용요금(won)
9.000000 ← 사용량(kWh)
계속하려면 아무 키나 누르세요 . . .

주요 설계/제작 진행상황

1. 전기사용량 요금 계산 코드 만들기

2. 조건선택

대가족요금 / 5인이상 가구
생명유지장치
요금 출산가구
 3자녀이상 가구
 생명유지장치
 해당없음

복지 할인요금 해당없음

사용기간 ~

사용량 kWh

요금계산

4. 상세계산내역

■ 월간 27kWh 사용시 전기요금 계산(주거용)

기본요금(원미만 절사) : 910원

전력량요금(원미만 절사) : 2,519원

· 1단계 : $27\text{kWh} \times 93.3\text{원} = 2,519.1\text{원}$

필수사용량 보장공제(원미만 절사) : 2,429원
필수사용량 보장공제 : 2,429원
- 필수사용량 보장공제 : 2,429원 ($910\text{원} + 2,519\text{원} - 4,000\text{원} < 1,000\text{원}$ (감액 후 최저요금))

전기요금계(기본요금 + 전력량요금 - 필수사용량 보장공제)
 $: 910\text{원} + 2,519\text{원} - 2,429\text{원} = 1,000\text{원}$

부가가치세(원미만 4사 5입) : $1,000\text{원} \times 0.1 = 100\text{원}$

910+2519=3429

C:\WINDOWS\system32\cmd.exe

```
129600
3429
27.000000
계속하려면 아무 키나 누르십시오 . . .
```

주요 설계/제작 진행상황

1. 전기사용량 요금 계산 코드 만들기

2. 조건선택

- 대가족요금/
생명유지장치
요금
- 5인이상 가구
 - 출산가구
 - 3자녀이상 가구
 - 생명유지장치
 - 해당없음

복지할인요금 해당없음

4. 상세계산내역

■ 월간 300kWh 사용시 전기요금 계산(주거용)

기본요금(원미만 절사) : 1,600원

전력량요금(원미만 절사) : 37,450원

·1단계 : 200kWh × 93.3원 = 18,660원

·2단계 : 100kWh × 187.9원 = 18,790원

$$1600 + 37450 = 39050$$

2. 조건선택

- 대가족요금/
생명유지장치
요금
- 5인이상 가구
 - 출산가구
 - 3자녀이상 가구
 - 생명유지장치
 - 해당없음

복지할인요금 해당없음

4. 상세계산내역

■ 월간 750kWh 사용시 전기요금 계산(주거용)

기본요금(원미만 절사) : 7,300원

전력량요금(원미만 절사) : 154,450원

·1단계 : 200kWh × 93.3원 = 18,660원

·2단계 : 200kWh × 187.9원 = 37,580원

·3단계 : 350kWh × 280.6원 = 98,210원

$$7300 + 154450 = 161750$$

C:\> C:\WINDOWS\system32\cmd.exe

1440000

39050

300.000000

계속하려면 아무 키나 누르십시오 . . .

C:\> C:\WINDOWS\system32\cmd.exe

3600000

161750

750.000000

계속하려면 아무 키나 누르십시오 . . .

주요 설계/제작 진행상황

2. 스케치 코드

```
#include <DHT.h>
#include <DHT_U.h>
#include <LiquidCrystal.h>
#include <SoftwareSerial.h> //시리얼통신 라이브러리 호출
SoftwareSerial btSerial(5,4); //PX,TX시리얼 통신을 위한 객체선언
LiquidCrystal lcd(7,8,9,10,11,12);
String myString="";

#define DHT11_PIN 2
int Sensor_pin = 2;
DHT my_sensor(Sensor_pin,DHT11);
```

```
float value=0;
float voltage=0;
float dustDensity=0;

int Echo_Pin=15;
int Trig_Pin=16;
int ALARM=13;
int cnt=0;

int relay=6;
```

주요 설계/제작 진행상황

2. 스케치 코드

```
void setup(){
    lcd.begin(16,2); //lcd
    Serial.begin(9600);
    btSerial.begin(9600); //블루투스 시리얼
    my_sensor.begin();

    pinMode(3,OUTPUT); //미세먼지 센서
    pinMode(A0,INPUT);

    pinMode(Echo_Pin,INPUT); //초음파센서
    pinMode(Trig_Pin,OUTPUT);

    pinMode(relay,OUTPUT); //릴레이모듈
}
```

주요 설계/제작 진행상황

2. 스케치 코드

```
void loop(){
    char data=btSerial.read();
    if (Serial.available()) {
        btSerial.write(Serial.read()); //블루투스측 내용을 시리얼모니터에 출력
    }
    if (btSerial.available()) {      //시리얼 모니터 내용을 블루추스 측에 WRITE
        char myChar=(char)btSerial.read(); //btSerial int 값을 char 형식으로 변환
        myString+=myChar;   //수신되는 문자를 myString에 모두 붙임(1바이트씩 전송되는 것을 연결)
        delay(5); //수신 문자열 끊김 방지
    }
}
```

주요 설계/제작 진행상황

2. 스케치 코드

```
int temp= my_sensor.readTemperature();
int humi=my_sensor.readHumidity();
Serial.print("Temperature: ");
Serial.print(temp);
Serial.print(" °C");
Serial.print(" Humidity: ");
Serial.print(humi);
Serial.println(" %");
btSerial.print("Temperature: ");
btSerial.print(temp);
btSerial.print(" °C");
btSerial.print(" Humidity: ");
btSerial.print(humi);
btSerial.println(" %");
```

```
Lcd.setCursor(0,0);
Lcd.print("Temperature: ");
Lcd.print(temp);
Lcd.print((char)223);
Lcd.print("C");
Lcd.setCursor(0,1);
Lcd.print("Humidity: ");
Lcd.print(humi);
Lcd.print("%");

digitalWrite(3,LOW);
delayMicroseconds(10);
value=analogRead(A0);
delayMicroseconds(10);
digitalWrite(3,HIGH);
delayMicroseconds(10);
```

주요 설계/제작 진행상황

2. 스케치 코드

```
voltage=value*5.0/1024.0;  
dustDensity=(voltage)/0.005;  
Serial.print("Dust Density : ");  
Serial.print(dustDensity);  
Serial.println(" µg/m³");  
btSerial.print("Dust Density : ");  
btSerial.print(dustDensity);  
btSerial.println(" µg/m³");
```

```
long duration,distance;  
digitalWrite(Trig_Pin,HIGH);  
delayMicroseconds(10);  
digitalWrite(Trig_Pin,LOW);  
duration=pulseIn(Echo_Pin,HIGH);  
  
distance=((float)(340*duration)/10000)/2;
```

```
Serial.print("Distance: ");  
Serial.print(distance);  
Serial.println(" cm");  
btSerial.print("Distance: ");  
btSerial.print(distance);  
btSerial.println(" cm");  
  
if(distance<=10){  
    cnt++;  
    if(cnt>=5){  
        tone(ALARM,1000);  
    }  
}  
else{  
    noTone(ALARM);  
    cnt=0;  
}
```

주요 설계/제작 진행상황

2. 스케치 코드

```
double pre=0;
double current=millis();
double time1=((current-pre)/1000)/3600;
float sa;//전기사용량
int bill;//전기사용요금
sa=(float)750*time1/(1000);//전기사용량 계산
if (sa <= 200) {
    bill =(int)( 910 + sa*93.3);
}
else if (sa> 200 && sa <= 400) {
    bill = (int)(1600 + 200*93.3 + (sa-200)*187.9);
}
else if (sa > 400) {
    bill= (int)(7300 + 200*93.3 + 200*187.9 + (sa-400)*280.6);
}

Serial.println(time1,14);
Serial.println(sa,14);
Serial.println(bill);
btSerial.print("Time : ");
btSerial.print(time1,6);
btSerial.println(" hour");
btSerial.print("Electric Use: ");
btSerial.print(sa,6);
btSerial.println(" KWH");
btSerial.print("Bill: ");
btSerial.print(bill);
btSerial.println(" WON");
```

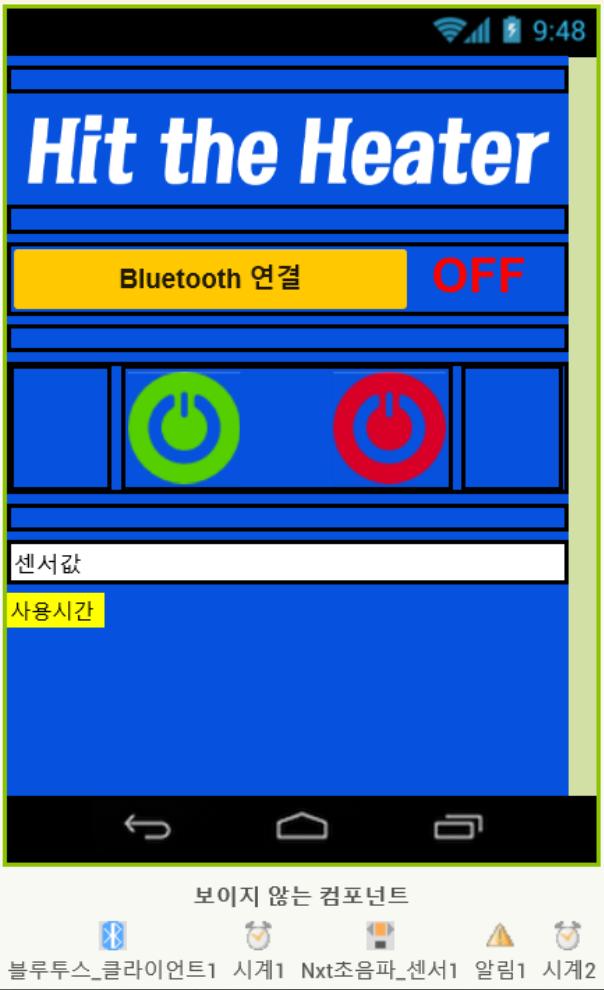
주요 설계/제작 진행상황

2. 스케치 코드

```
if(data=='1'){ //앱인벤터에서 1을 받아오면 전원 ON  
    digitalWrite(relay,HIGH);  
    delayMicroseconds(10);  
  
}  
if(data=='0'){ //앱인벤터에서 0을 받아오면 전원 OFF  
    digitalWrite(relay,LOW);  
    delayMicroseconds(10);  
}  
delay(2000); //2초마다 이를 반복  
}
```

주요 설계/제작 진행상황

3. 앱인벤터 블록 코드 작성-진행중



```
언제 블루투스 .선택 전
실행 지정하 블루투스 .요소 값 블루투스_클라이언트1 .주소와 이름들

언제 블루투스 .선택 후
실행 지정하 블루투스 .선택된 항목 값 호출 블루투스_클라이언트1 .연결 주소 블루투스 .선택된 항목

만약 블루투스_클라이언트1 .연결 상태 = 참
그러면 지정하 블루투스_온오프 .텍스트 값 "ON"
지정하 블루투스_온오프 .텍스트 색상 값

언제 시계1 .타이마
실행 만약 블루투스_클라이언트1 .연결 상태 = 참
그러면 지정하 센서값 .텍스트 값 호출 블루투스_클라이언트1 .텍스트 받기 바이트 수 호출 블루투스_클라이언트1 .받을 수 있는 바이트

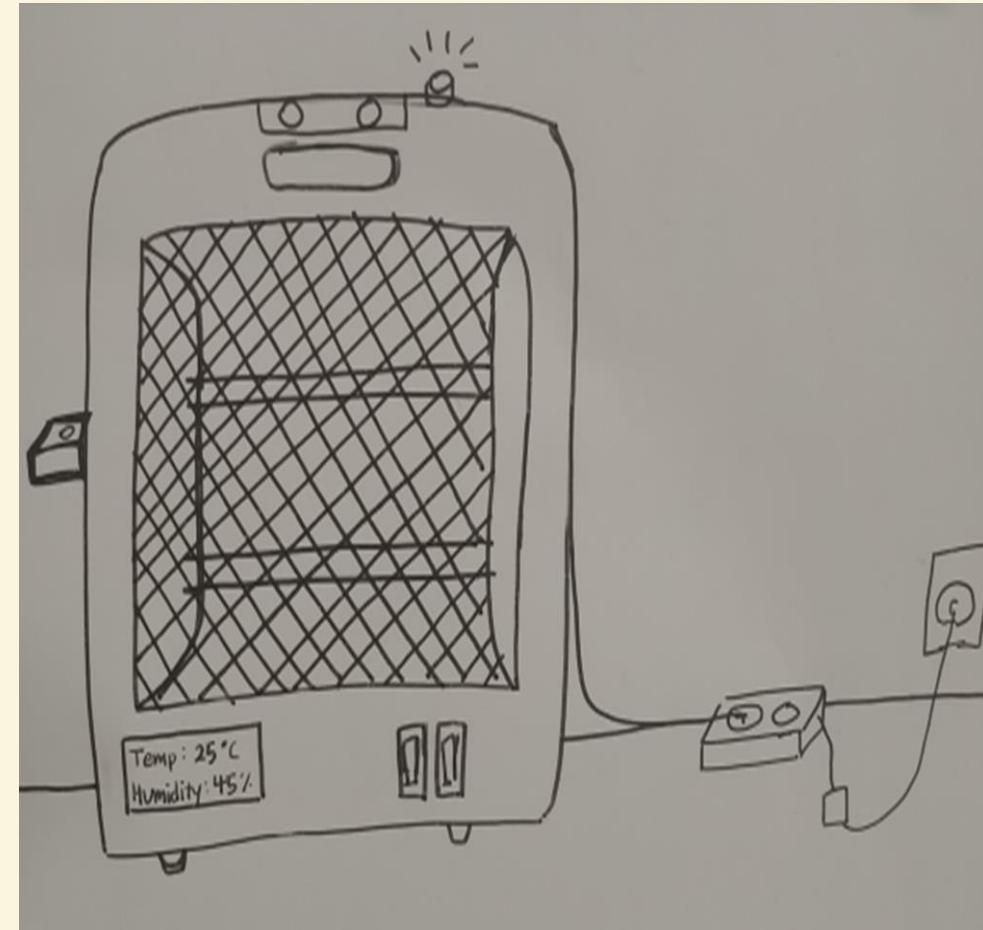
언제 ON .클릭
실행 만약 블루투스_클라이언트1 .연결 상태 = 참
그러면 호출 블루투스_클라이언트1 .텍스트 보내 텍스트 "1"

언제 OFF .클릭
실행 만약 블루투스_클라이언트1 .연결 상태 = 참
그러면 호출 블루투스_클라이언트1 .텍스트 보내 텍스트 "0"
지정하 레이블1 .텍스트 값 호출 시계1 .DurationToMinutes duration
호출 시계1 .기간 시작
끝 호출 시계1 .지금
```

<미완성>
On을 했을 때부터 Off까지 사용시간/
사용요금을 출력하지 못함

주요 설계/제작 진행상황

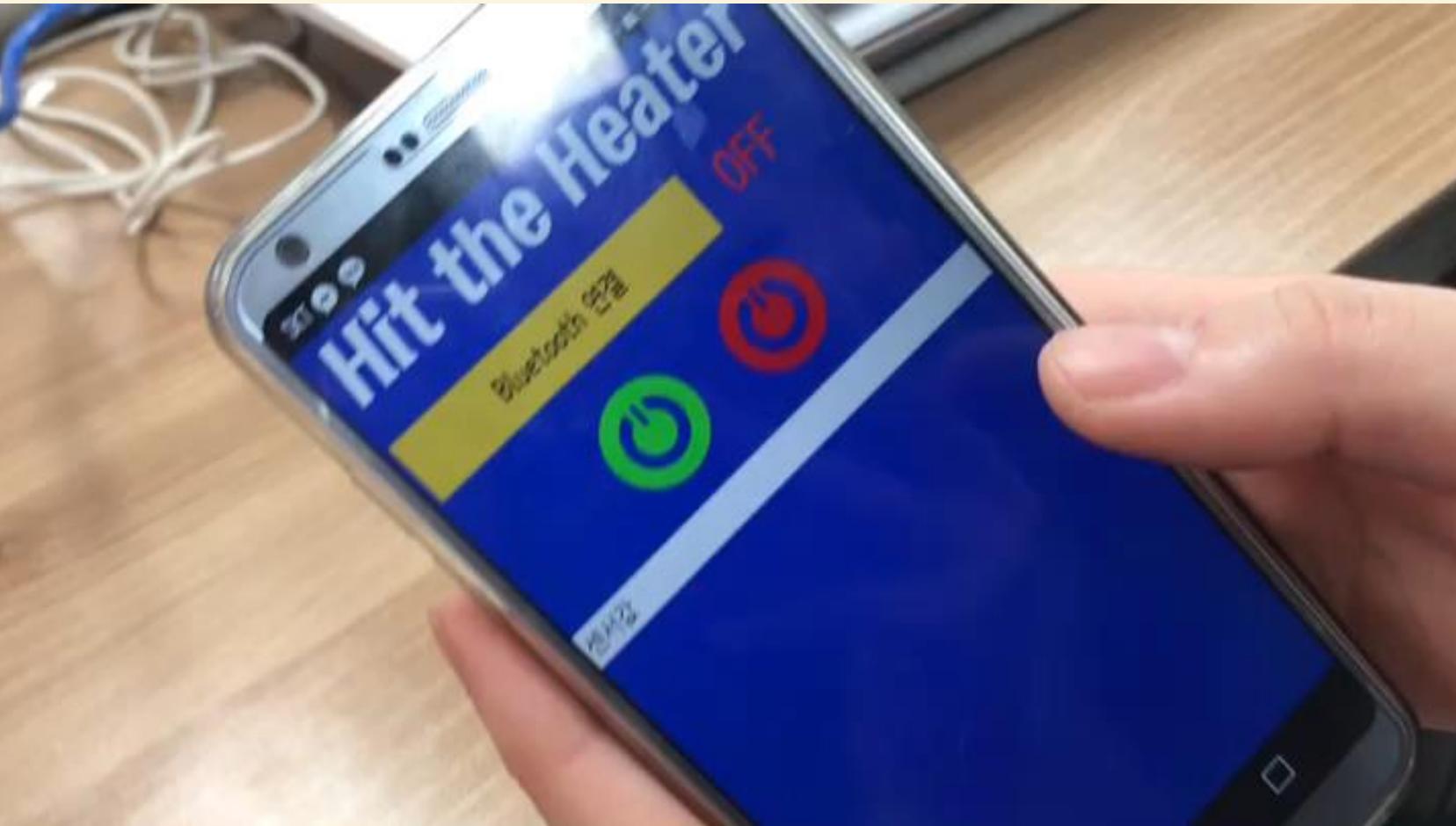
4. 제품 제작



1차시 때 제품 스케치와 비교한 모습

주요 설계/제작 진행상황

5. 현재 진행상황까지의 결과



팀원 별 구체적인 역할

17011849 박영주 - 전기사용요금 계산 코드(C언어) 작성, 아두이노 스케치코드
사용량계산 작성, 앱인벤터 전기사용요금, 사용시간 출력 과정 중,
제품 제작

17011871 하윤영 - 전기사용요금 계산 코드(C언어), 전기사용요금 자료조사, 제품 제작
, 발표

17011885 박세정 - 앱인벤터 전기사용요금, 사용시간 출력 과정 중, 앱인벤터를 이용한
타이머 계산, 팝업알림기능 자료조사, 제품 제작, ppt 제작

참고문헌 및 사이트

〈앱인벤터〉

<https://blog.naver.com/milly0304/221284602786>

<https://blog.naver.com/nasu0210/220684394910>

〈전기사용료〉

<http://cyber.kepco.co.kr/ckeprco/front/jsp/CY/E/E/CYEEHP00101.jsp>

진행 보고서

3 (12주차)

진행상황

〈12주차 진행상황〉

전원 제어(릴레이 모듈)

릴레이 모듈까지 넣은 스케치 코드 작성

앱인벤터 화면

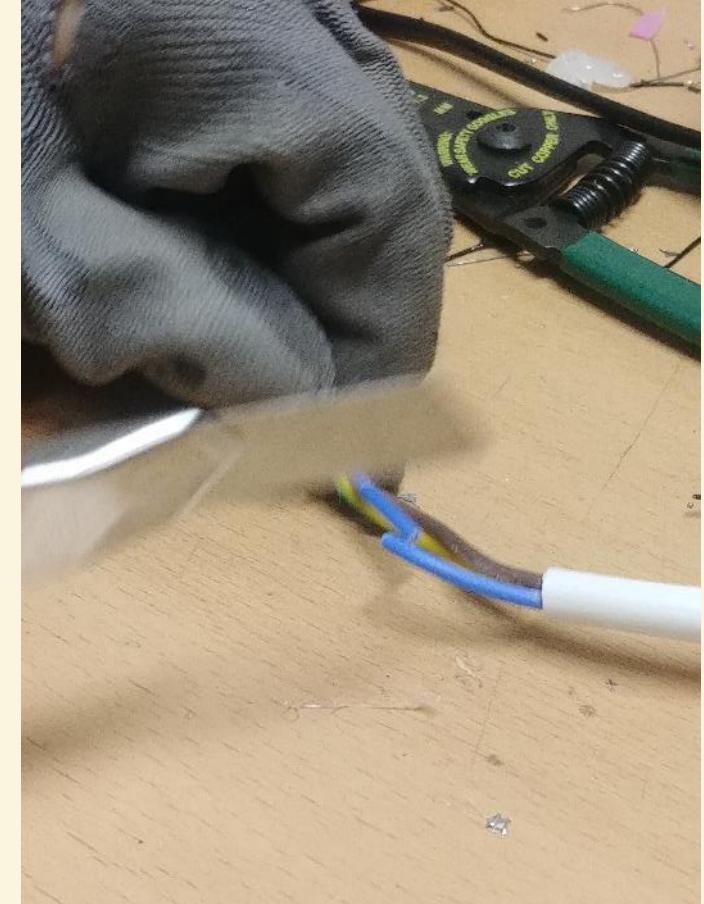
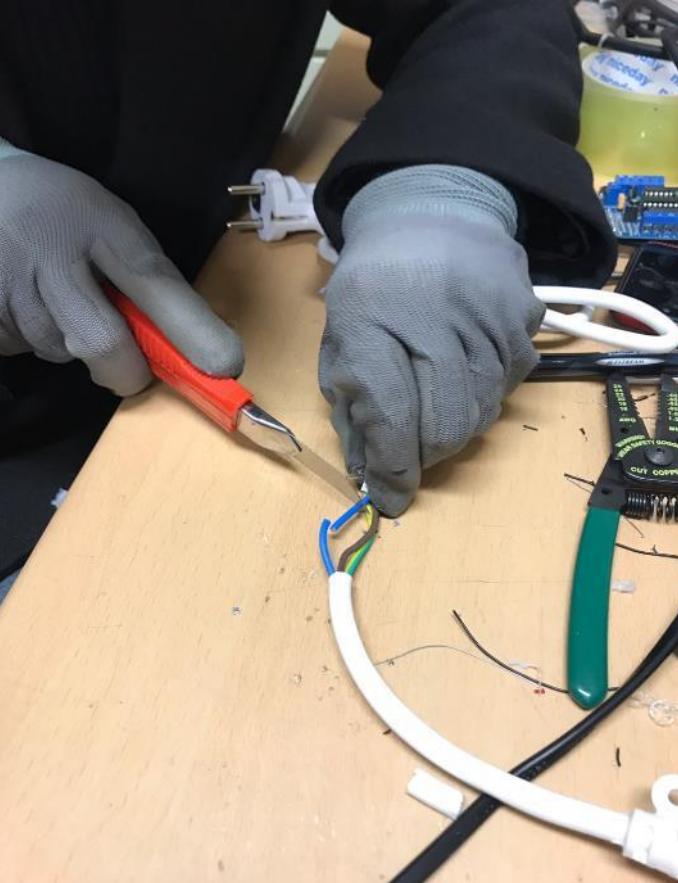
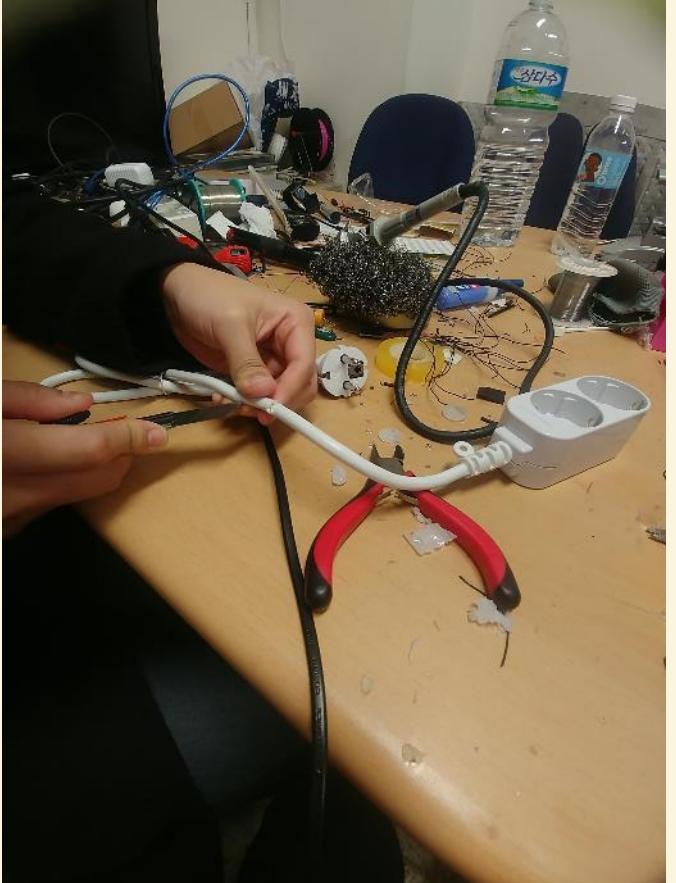
전기사용량 수치 조사

주별 설계//역할

		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 온습도 센서 + LCD 연결 하윤영: 초음파 센서 연결
11주차		부품 연결, 블루투스 스마트폰 연결	박영주: 블루투스 연결 및 부품 연결 스케치 보드 작성 박세정: 아두이노 회로 연결, 앱인벤터 블루투스 연결 하윤영: 아두이노 회로 연결, 릴레이 자료조사
12주차		전원 제어 실험 및 스마트폰 어플 스케치	박영주: 릴레이 모듈 연결 및 실행 박세정: 멀티탭 분리, 앱인벤터 화면 디자인 하윤영: 릴레이 모듈 실험
13주차		전원제작 완성 및 스마트폰 어플 제작	박영주: 앱인벤터 전원 제어 ON/OFF 블록 작성 박세정: 앱인벤터 화면 디자인, 앱인벤터 기본 블록 작성 하윤영: 어플에 삽입할 기능 알고리즘 작성, 합치기
14주차		제품 만들기	박영주: 제품 제작 박세정: 제품 제작 하윤영: 제품 제작
15주차		시연 및 보완	
16주차		결과	

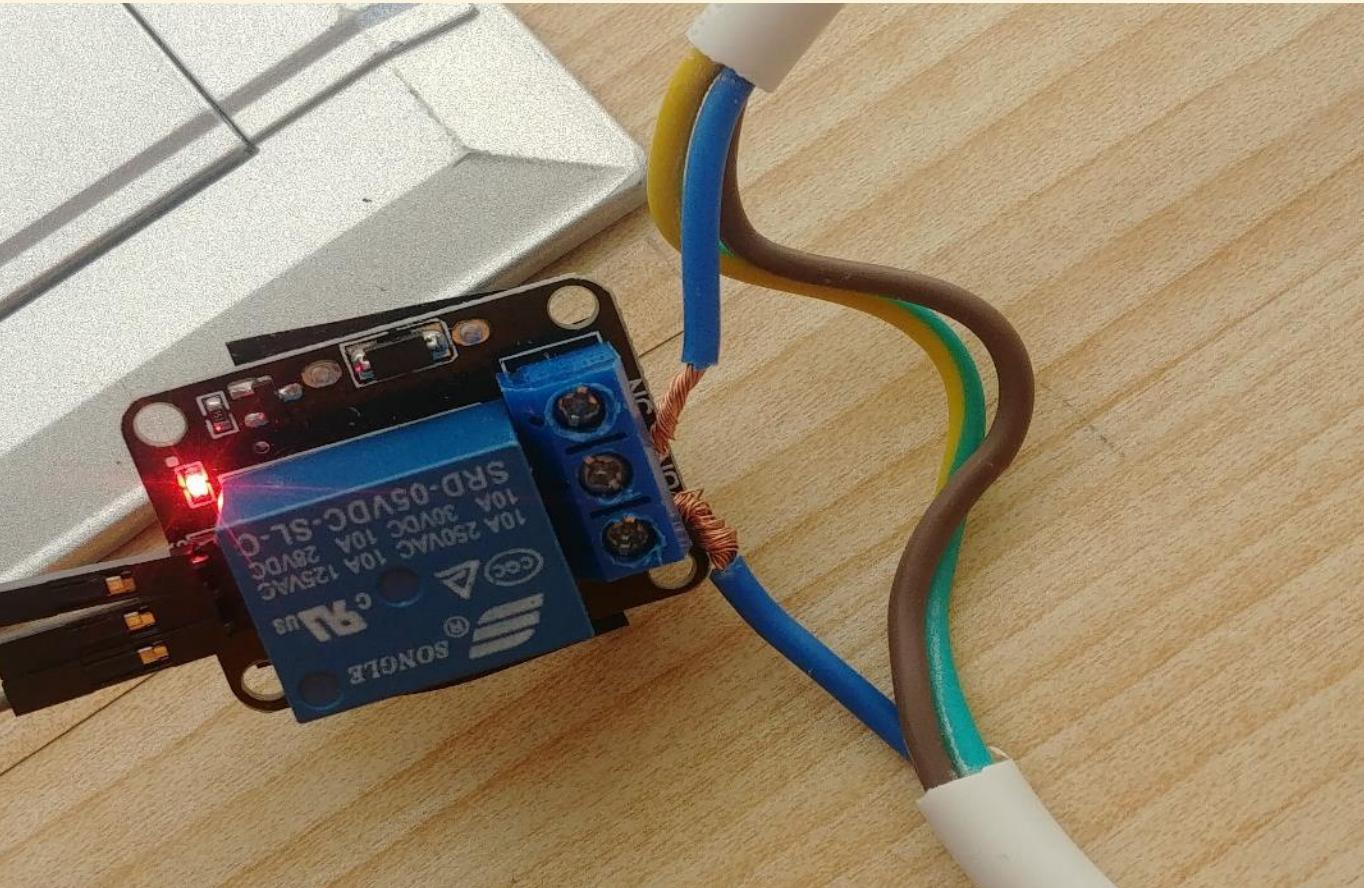
주요 설계/제작 진행상황

1. 전원 제어 실험 과정



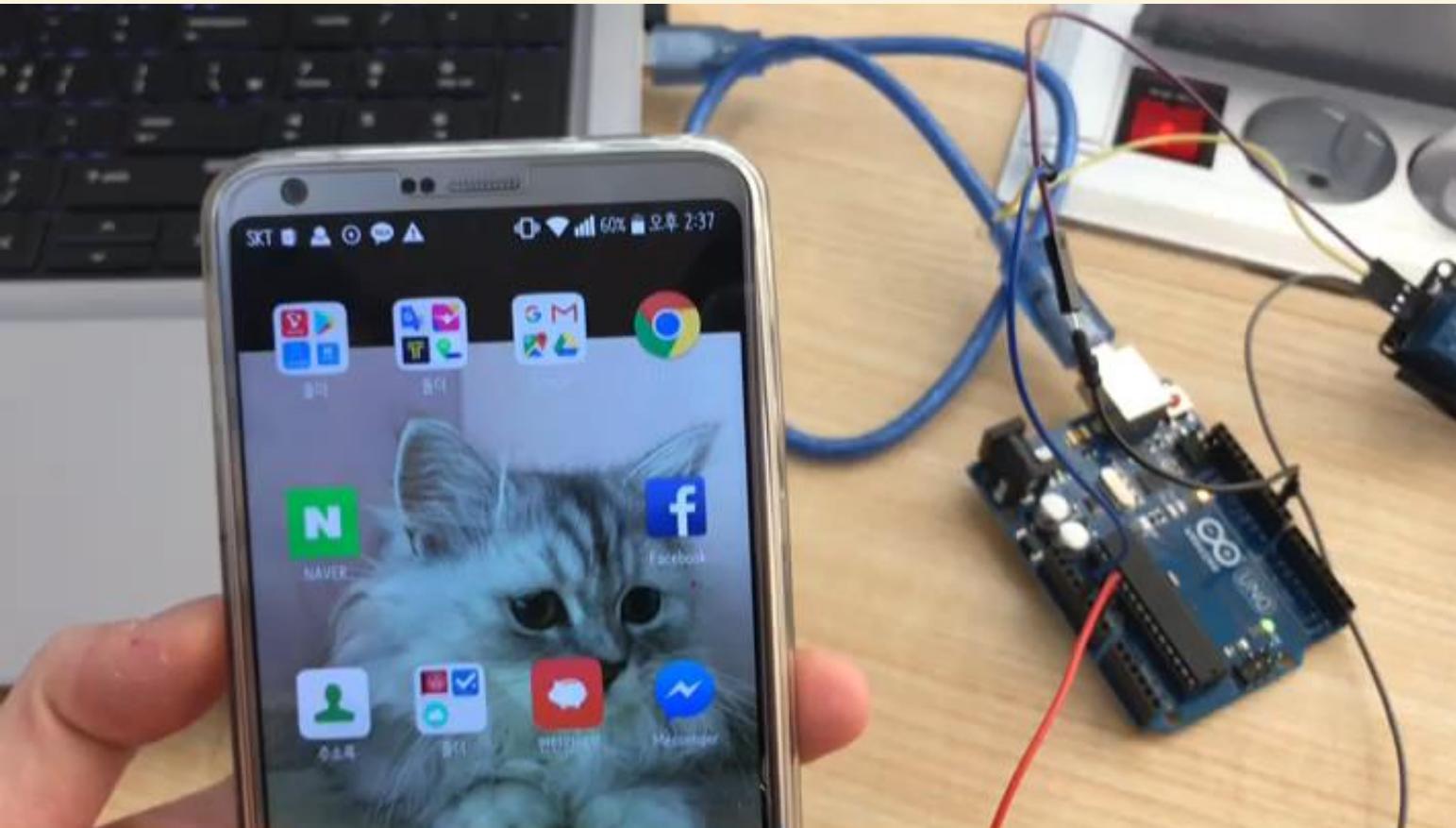
주요 설계/제작 진행상황

1. 전원 제어 실험 과정



주요 설계/제작 진행상황

1. 전원 제어 실험 과정



주요 설계/제작 진행상황

2. 스케치 코드 작성

```
#include <DHT.h>
#include <DHT_U.h>
#include <LiquidCrystal.h>
#include <SoftwareSerial.h> //시리얼통신 라이브러리
//호출
SoftwareSerial btSerial(5,4); //PX,TX시리얼 통신을 위한 객체선언
LiquidCrystal lcd(7,8,9,10,11,12);
String myString="";

#define DHT11_PIN 2
int Sensor_pin = 2;
DHT my_sensor(Sensor_pin,DHT11);

float value=0;
float voltage=0;
float dustDensity=0;
```

```
int Echo_Pin=15;
int Trig_Pin=16;
int ALARM=13;
int cnt=0;

int relay=6;
```

주요 설계/제작 진행상황

2. 스케치 코드 작성

```
void setup(){  
    lcd.begin(16,2); //lcd  
    Serial.begin(9600);  
    btSerial.begin(9600); //블루투스 시리얼  
    my_sensor.begin();  
  
    pinMode(3,OUTPUT); //미세먼지 센서  
    pinMode(A0,INPUT);  
  
    pinMode(Echo_Pin,INPUT); //초음파센서  
    pinMode(Trig_Pin,OUTPUT);  
  
    pinMode(relay,OUTPUT); //릴레이모듈  
}
```

주요 설계/제작 진행상황

2. 스케치 코드 작성

```
void loop(){
    int data;
    if (Serial.available()) {
        btSerial.write(Serial.read()); //블루투스측 내용을 시리얼모니터에 출력
    }
    if (btSerial.available()) {      //시리얼 모니터 내용을 블루추스 측에 WRITE
        char myChar=(char)btSerial.read(); //btSerial int 값을 char 형식으로 변환
        myString+=myChar;   //수신되는 문자를 myString에 모두 붙임(1바이트씩 전송되는 것을 연결)
        delay(5); //수신 문자열 끊김 방지
    }
    if(data==1){ //앱인벤터에서 1을 받아오면 전원 ON
        digitalWrite(relay,HIGH);
        delayMicroseconds(10);
    }
    if(data==0){ //앱인벤터에서 0을 받아오면 전원 OFF
        digitalWrite(relay,LOW);
        delayMicroseconds(10);
    }
}
```

주요 설계/제작 진행상황

2. 스케치 코드 작성

```
long duration,distance;  
digitalWrite(Trig_Pin,HIGH);  
delayMicroseconds(10);  
digitalWrite(Trig_Pin,LOW);  
duration=pulseIn(Echo_Pin,HIGH);  
  
distance=((float)(340*duration)/10000)/2;  
  
Serial.print("Distance: ");  
Serial.print(distance);  
Serial.println(" cm");  
btSerial.print("Distance: ");  
btSerial.print(distance);  
btSerial.println(" cm");
```

```
if(distance<=10){  
    cnt++;  
    if(cnt>=5){  
        tone(ALARM,1000);  
    }  
}  
else{  
    noTone(ALARM);  
    cnt=0;  
}
```

주요 설계/제작 진행상황

2. 스케치 코드 작성

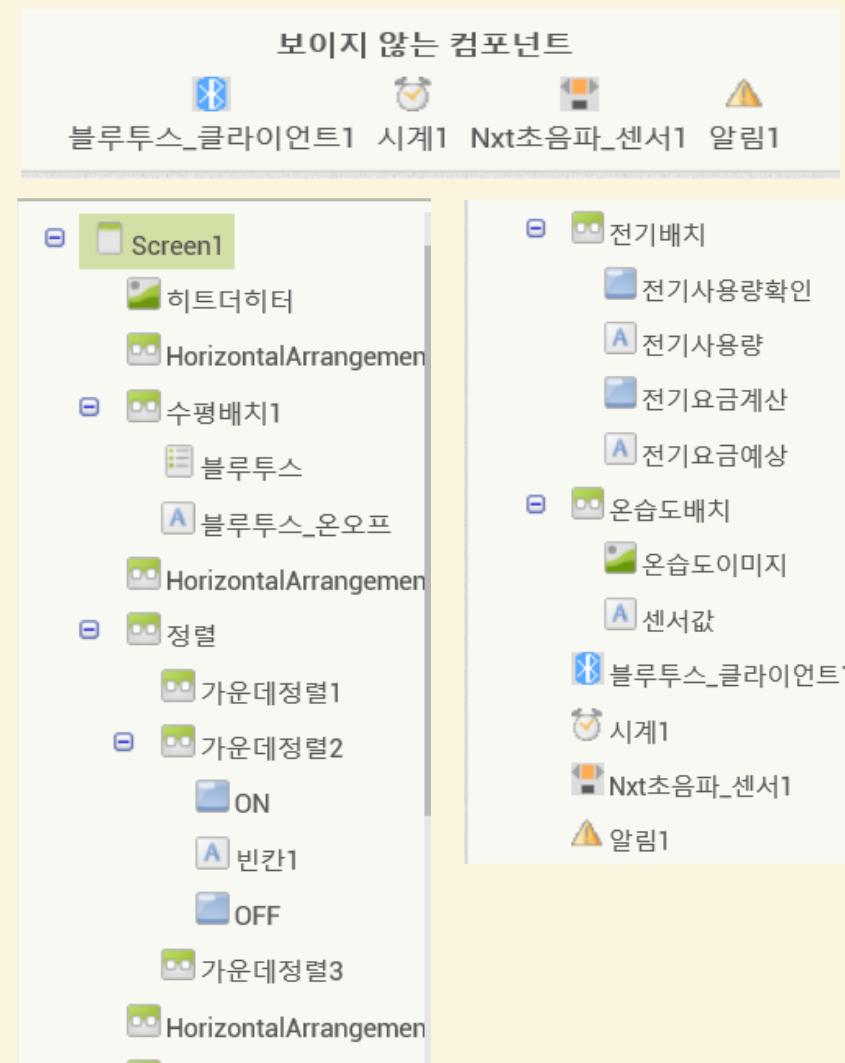
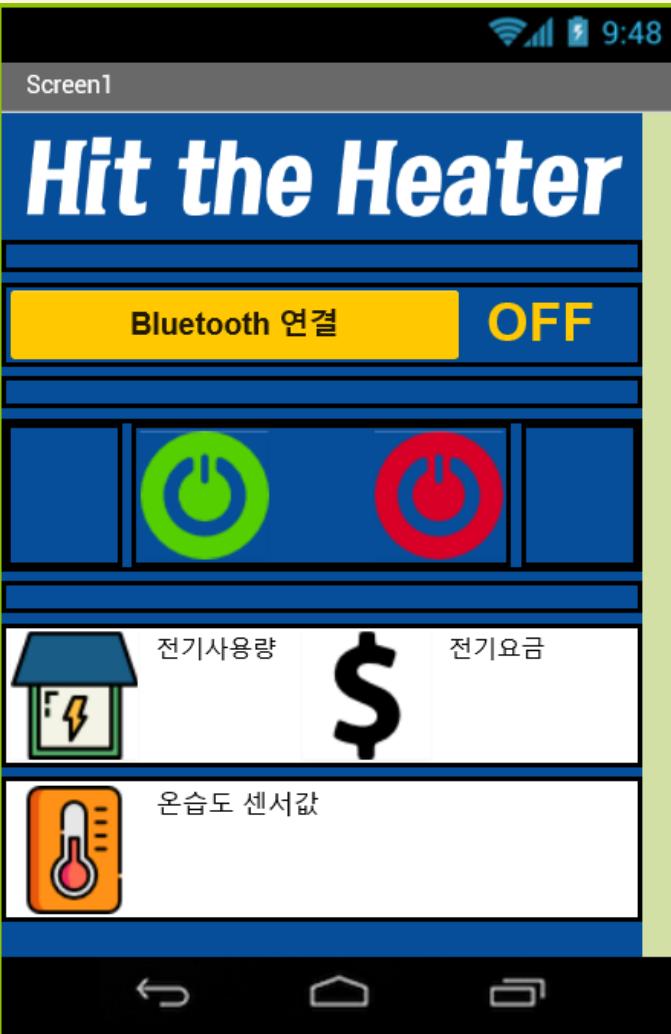
```
digitalWrite(3,LOW);
delayMicroseconds(280);
value=analogRead(A0);
delayMicroseconds(40);
digitalWrite(3,HIGH);
delayMicroseconds(9680);

voltage=value*5.0/1024.0;
dustDensity=(voltage)/0.005;
Serial.print("Dust Density : ");
Serial.print(dustDensity);
Serial.println(" µg/m³");
btSerial.print("Dust Density : ");
btSerial.print(dustDensity);
btSerial.println(" µg/m³");

delay(2000); //2초마다 이를 반복
}
```

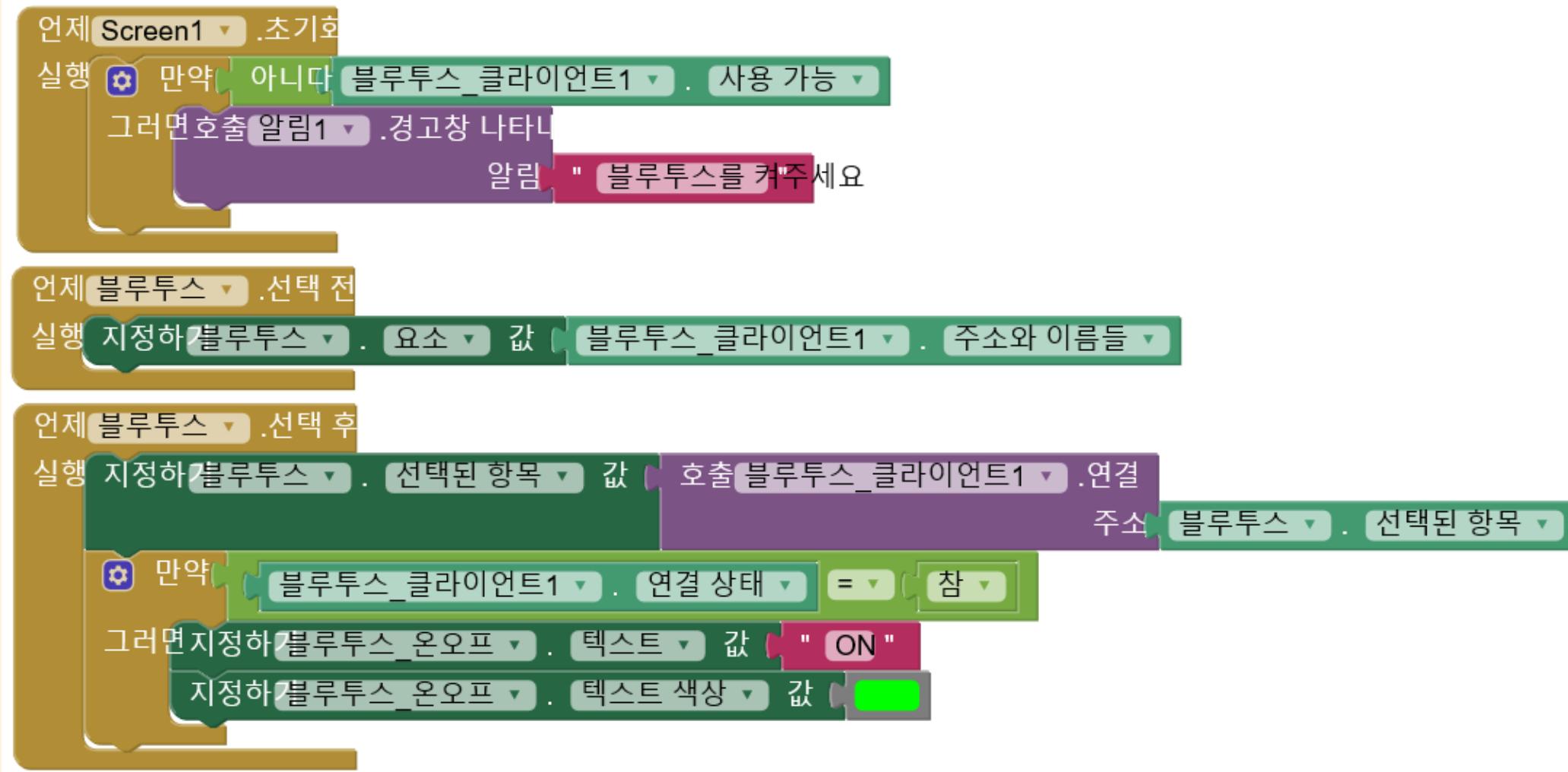
주요 설계/제작 진행상황

2. 앱인벤터 제작 과정



주요 설계/제작 진행상황

2. 앱인벤터 제작 과정(블록 부분-미완성/블루투스 연결까지 진행)



주요 설계/제작 진행상황

2. 앱인벤터 시 제작 해야할 일(어플 제작 시 생각해야 할 요소)
 1. ON과 OFF 사용시간 구하기 -> 전기사용료 계산과 연관
 2. 전기사용량, 전기 요금 계산하기(사용시간 구하기 후 작성) /온습도, 먼지 센서 값 받아오기
 3. 일정거리와 일정 요금 이상 초과될 시에 알람
 4. 어플 안에서 사용자가 일정요금을 지정하여 요금 초과 시 알람 뜨는 기능 알아보기

주요 설계/제작 진행상황

3. 전기 사용료

기본요금(원/호)			전력량요금(원/kWh)		
사용량 구간	저압	고압	사용량 구간	저압	고압
100kWh이하사용	410	410	100kWh까지	60.7	57.6
101~200kWh사용	910	730	다음 100kWh까지	125.9	98.9
201~300kWh사용	1,600	1,260	다음 100kWh까지	187.9	147.3
301~400kWh사용	3,850	3,170	다음 100kWh까지	280.6	215.6
401~500kWh사용	7,300	6,060	다음 100kWh까지	417.7	325.7
500kWh초과사용	12,940	10,760	500kWh 초과	709.5	574.6

제품규격

모델명	HM-608
브랜드	JNIX
제품크기	310 x 380 x 145
정격전압	220V / 60Hz
소비전력	750W
발열량	645Kcal/h
중량	2kg

소비전력 750W * 시간 %33 = ?(kWh)

주요 설계/제작 진행상황

3. 전기 사용료

주택용 전력(저압)

- 주거용 고객(아파트 고객 포함), 계약전력 **3kW 이하**의 고객
- 독신자 합숙소(기숙사 포함) 또는 집단주거용 사회복지시설로서 고객이 주택용 전력의 적용을 희망하는 경우 적용
- 주거용 오피스텔 고객(주거용 오피스텔이란? 주택은 아니지만 실제 주거용도로 이용되는 오피스텔)

적용일자 : 2016년 12월 1일

기본요금(원/호)		전력량 요금(원/kWh)	
200kWh 이하 사용	910	처음 200kWh 까지	93.3
201 ~ 400kWh 사용	1,600	다음 200kWh 까지	187.9
400kWh 초과 사용	7,300	400kWh 초과	280.6

주택용 전력(고압)

- 고압으로 공급받는 가정용 고객에게 적용

적용일자 : 2016년 12월 1일

기본요금(원/호)		전력량 요금(원/kWh)	
200kWh 이하 사용	730	처음 200kWh 까지	78.3
201 ~ 400kWh 사용	1,260	다음 200kWh 까지	147.3
400kWh 초과 사용	6,060	400kWh 초과	215.6

주요 설계/제작 진행상황

3. 전기 사용료

총 사용량 요금계산		전기제품 사용량 계산			
제품 종류	소비전력	평균 사용시간		월간 사용량	
기타	750 W	일간	1 시간	22.5 kWh	
선택	0 W	일간	0 시간	0 kWh	
선택	0 W	일간	0 시간	0 kWh	
선택	0 W	일간	0 시간	0 kWh	
선택	0 W	일간	0 시간	0 kWh	
합계				22	kWh

계약종별 전기요금 계산

전기요금계산기 사용방법

1. 계약종별선택

- 주택용(저압)**
- 주택용(고압)
- 일반용(갑)I
- 일반용(갑)II
- 일반용(을)
- 1주택 수 가구
- 교육용(갑)
- 교육용(을)
- 산업용(갑)I
- 산업용(갑)II
- 산업용(을)
- 임시(갑)
- 임시(을) 300kW▼
- 임시(을) 300kW▲
- 임시(을) 300kW▼ 시
- 간대별 요금사용
- 가로등(을)
- 심야전력(갑)
- 농사용(갑)
- 농사용(을)

2. 조건선택

대가족요금 /	<input type="radio"/> 5인이상 가구
생명유지장치	<input type="radio"/> 출산가구
요금	<input type="radio"/> 3자녀이상 가구
	<input type="radio"/> 생명유지장치
	<input checked="" type="radio"/> 해당없음

복지 할인요금 해당없음

사용기간 ~

사용량 kWh

요금계산

3. 계산된결과

계산된 금액
은
1,130 원 입니다.

4. 상세계산내역

■ 월간 22kWh 사용시 전기요금 계산(주거용)

기본요금(원미만 절사) : 910원

전력량요금(원미만 절사) : 2,052원

필수사용량 보장공제(원미만 절사) : 1,962원

필수사용량 보장공제 : 1,962원

- 필수사용량 보장공제 : 1,962원 (910원 + 2,052원 - 4,000원 < 1,000원 (감액 후 최저요금))

전기요금계(기본요금 + 전력량요금 - 필수사용량 보장공제)

$$910원 + 2,052원 - 1,962원 = 1,000원$$

부가가치세(원미만 4사 5입) : $1,000원 \times 0.1 = 100원$

전력산업기반기금(10원미만 절사) : $1,000원 \times 0.037 = 30원$

청구금액(전기요금계 + 부가가치세 + 전력산업기반기금)

$$1,000원 + 100원 + 30원 = 1,130원(10원미만 절사)$$

전기요금표 보기

프린트

팀원 별 구체적인 역할

17011849 박영주 - 릴레이 모듈 제어 실험 및 연결, 전체 스케치 코드 작성, 발표

17011871 하윤영 - 릴레이 모듈 제어 실험, 전기사용료 조사

17011885 박세정 - 멀티탭 분리 및 연결, 앱인벤터 화면 디자인, ppt 제작

참고문헌 및 사이트

〈릴레이 제어〉

<https://cafe.naver.com/arduinoguide/597>

<https://blog.naver.com/compass1111/221288414858>

〈앱인벤터〉

<https://blog.naver.com/icbanq/221006800539>

<http://blog.naver.com/PostView.nhn?blogId=rhkd824&logNo=220542365789&parentCategoryNo=&categoryNo=&viewDate=&isShowPopularPosts=false&from=postView>

〈전기사용료〉

<http://cyber.kepco.co.kr/ckeepco/front/jsp/CY/E/E/CYEEHP00101.jsp>

진행 보고서

2 (11주차)

진행상황

<11주차 진행상황>

센서 연결 및 코드 합치기

블루투스 연결

릴레이 실험

앱인벤터 진행 중

<추가 구입>

멀티 콘센트 일반형 2구 3000원

주별 설계//역할

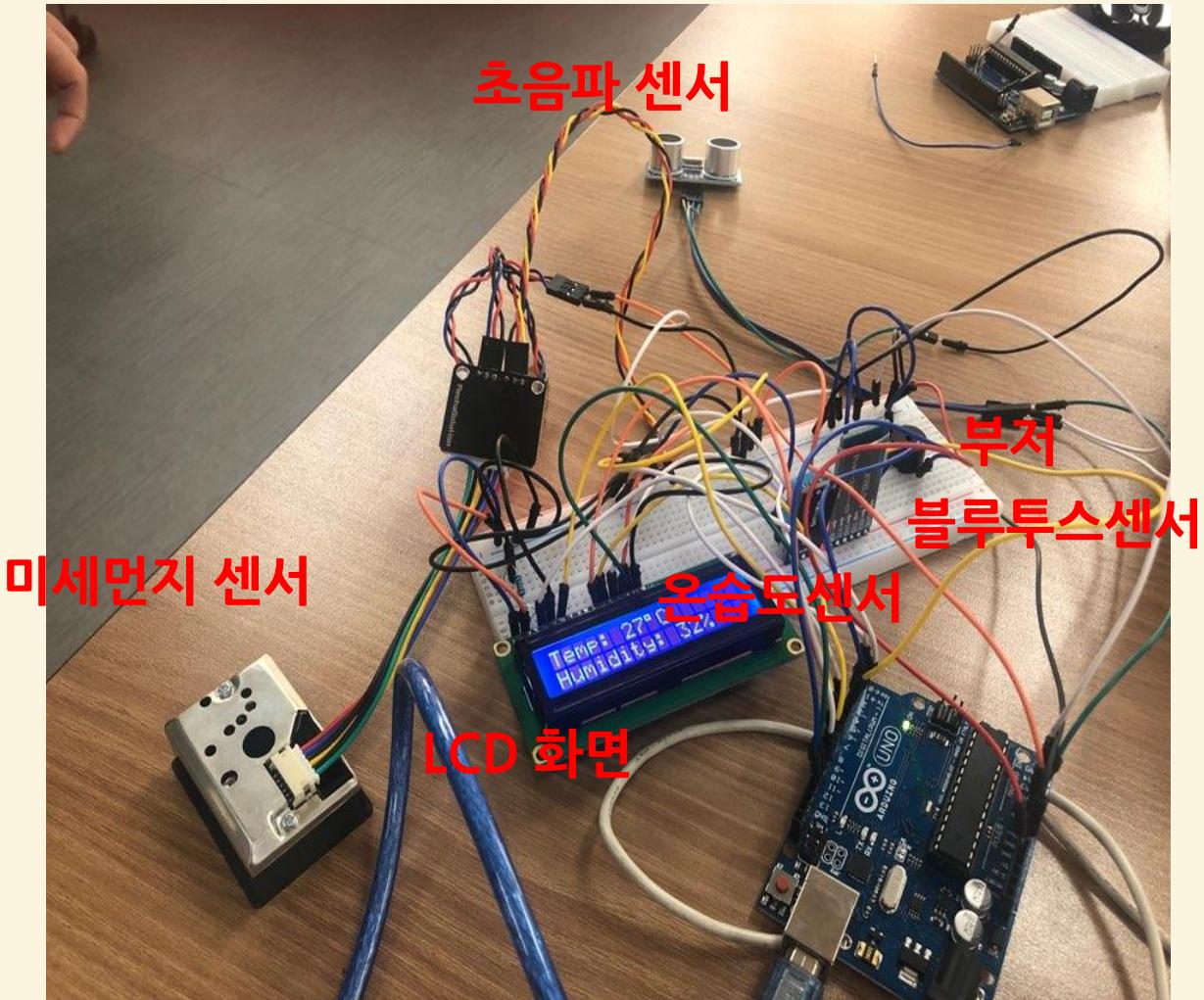
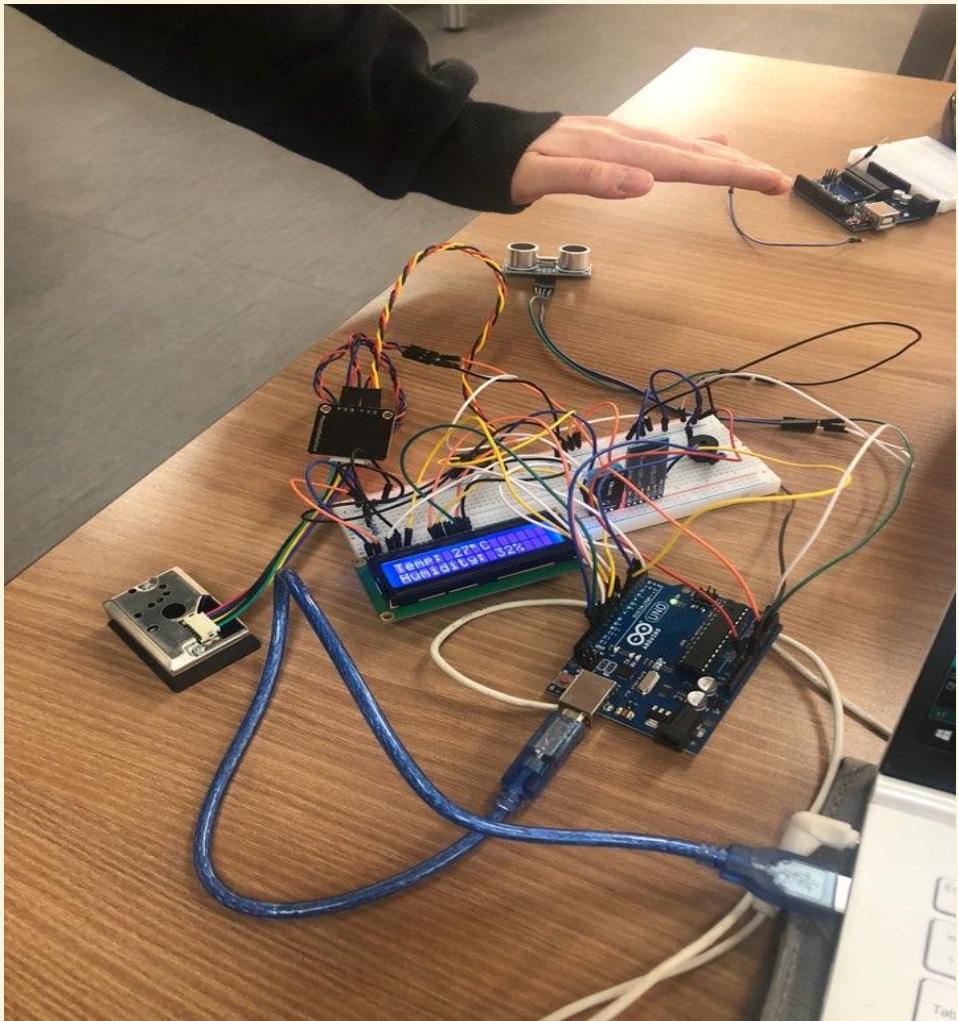
		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 온습도 센서 + LCD 연결 하윤영: 초음파 센서 연결
11주차		부품 연결	박영주: 스케치 코드 작성 박세정: 아두이노 회로 연결 하윤영: 아두이노 회로 연결
12주차		블루투스 이용하여 스마트폰 연결	박영주: 앱인벤터 제작 박세정: 블루투스연결 하윤영: 블루투스 스케치 보드 작성
13주차		스마트폰 어플 제작	박영주: 앱인벤터 제작 박세정: 앱인벤터 화면 디자인 하윤영: 앱인벤터와 아두이노 연결
14주차		전원 제어 및 실험	박영주: 원격제어 및 실험 박세정: 원격제어 및 실험 하윤영: 원격제어 및 실험
15주차		시연 및 보완	
16주차		결과	

주별 설계//역할

		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 온습도 센서 + LCD 연결 하윤영: 초음파 센서 연결
11주차		부품 연결, 블루투스 스마트폰 연결	박영주: 블루투스 연결 및 부품 연결 스케치 보드 작성 박세정: 아두이노 회로 연결, 앱인벤터 블루투스 연결 하윤영: 아두이노 회로 연결, 릴레이 자료조사
12주차		전원 제어 실험 및 스마트폰 어플 스케치	박영주: 릴레이 모듈 연결 및 실행 박세정: 멀티탭 분리, 앱인벤터 화면 디자인 하윤영: 릴레이 모듈 실험
13주차		전원제작 완성 및 스마트폰 어플 제작	박영주: 앱인벤터 전원 제어 ON/OFF 블록 작성 박세정: 앱인벤터 화면 디자인, 앱인벤터 기본 블록 작성 하윤영: 어플에 삽입할 기능 알고리즘 작성, 합치기
14주차		제품 만들기	박영주: 제품 제작 박세정: 제품 제작 하윤영: 제품 제작
15주차		시연 및 보완	
16주차		결과	

주요 설계/제작 진행상황

1. 부품 연결



주요 설계/제작 진행상황

2. 스케치 코드

```
#include <DHT.h>
#include <DHT_U.h>
#include <LiquidCrystal.h>
#include <SoftwareSerial.h> //시리얼통신 라이브러리 호출
SoftwareSerial btSerial(4,5); //PX,TX시리얼 통신을 위한 객체선언
LiquidCrystal lcd(7,8,9,10,11,12);

#define DHT11_PIN 2
int Sensor_pin = 2;
DHT my_sensor(Sensor_pin,DHT11);

float value=0;
float voltage=0;
float dustDensity=0;

int Echo_Pin=15;
int Trig_Pin=16;
int ALARM=13;
int cnt=0;
```

```
void setup(){
    lcd.begin(16,2);
    Serial.begin(9600);
    btSerial.begin(9600); //블루투스 시리얼
    my_sensor.begin();

    pinMode(3,OUTPUT);
    pinMode(A0,INPUT);

    pinMode(Echo_Pin,INPUT);
    pinMode(Trig_Pin,OUTPUT);
}
```

주요 설계/제작 진행상황

```
void loop(){
    if (Serial.available()) {
        btSerial.write(Serial.read());
    }
    if (btSerial.available()) {
        char myChar=(char)btSerial.read();
        myString+=myChar;
        delay(5); }
    int temp= my_sensor.readTemperature();
    int humi=my_sensor.readHumidity();
    lcd.setCursor(0,0);
    lcd.print("Temp: ");
    lcd.print(temp);
    lcd.print((char)223);
    lcd.print("C");
    lcd.setCursor(0,1);
    lcd.print("Humidity: ");
    lcd.print(humi);
    lcd.print("%");
```

```
Serial.print("Temperature: ");
Serial.print(temp);
Serial.print(" °C");
Serial.print(" Humidity: ");
Serial.print(humi);
Serial.println(" %");
btSerial.print("Temperature: ");
btSerial.print(temp);
btSerial.print(" °C");
btSerial.print(" Humidity: ");
btSerial.print(humi);
btSerial.println(" %");
```

주요 설계/제작 진행상황

```
long duration,distance;
digitalWrite(Trig_Pin,HIGH);
delayMicroseconds(10);
digitalWrite(Trig_Pin,LOW);
duration=pulseIn(Echo_Pin,HIGH);

distance=((float)(340*duration)/10000)/2;

Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
btSerial.print("Distance: ");
btSerial.print(distance);
btSerial.println(" cm");

if(distance<=10){
    cnt++;
    if(cnt>=5){
        tone(ALARM,1000);
    }
}

else{
    noTone(ALARM);
    cnt=0;
}

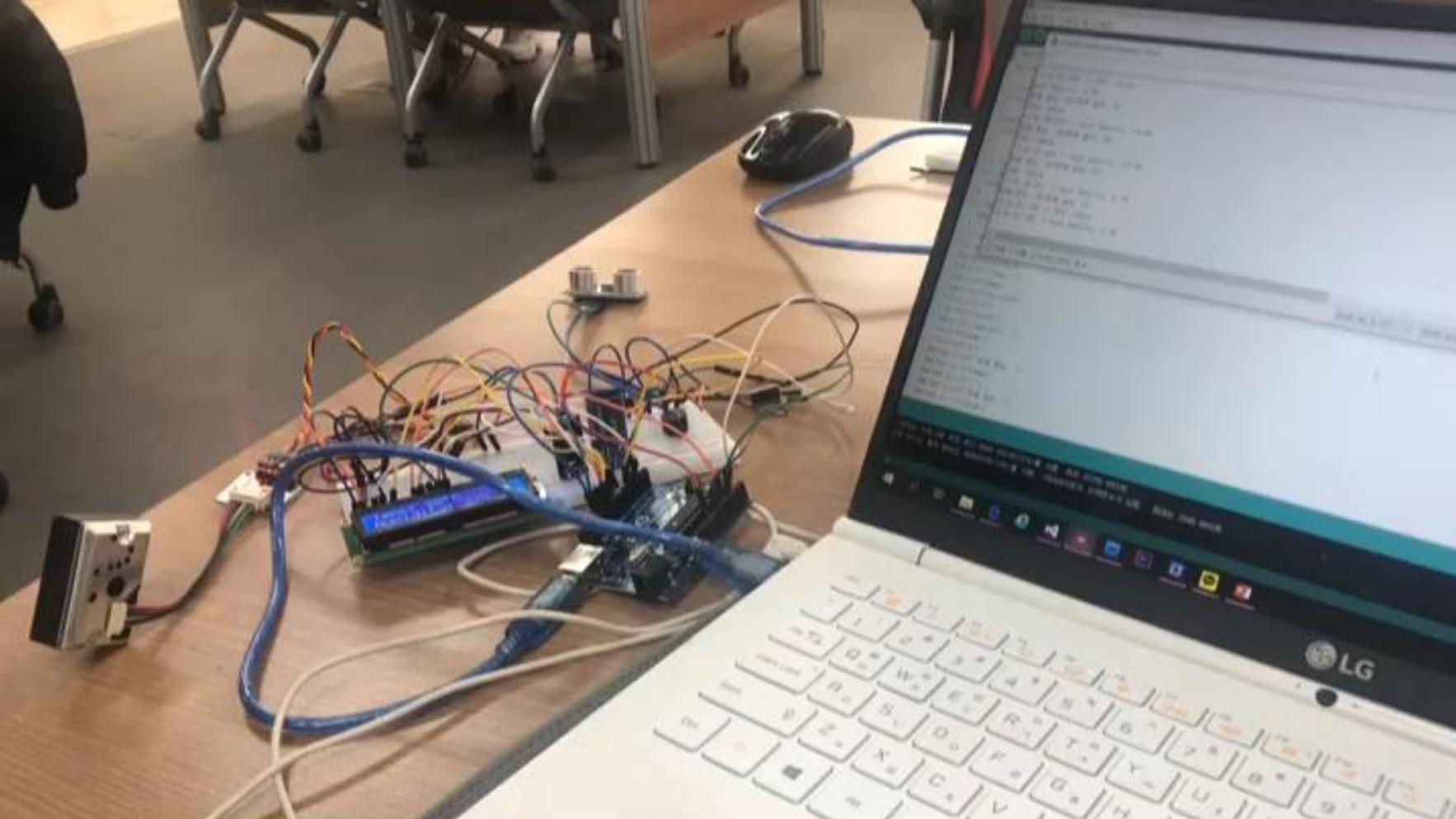
digitalWrite(3,LOW);
delayMicroseconds(280);
value=analogRead(A0);
delayMicroseconds(40);
digitalWrite(3,HIGH);
delayMicroseconds(9680);

voltage=value*5.0/1024.0;
dustDensity=(voltage)/0.005;
Serial.print("Dust Density :");
Serial.println(dustDensity);
btSerial.print("Dust Density :");
btSerial.println(dustDensity);

delay(2000);}
```

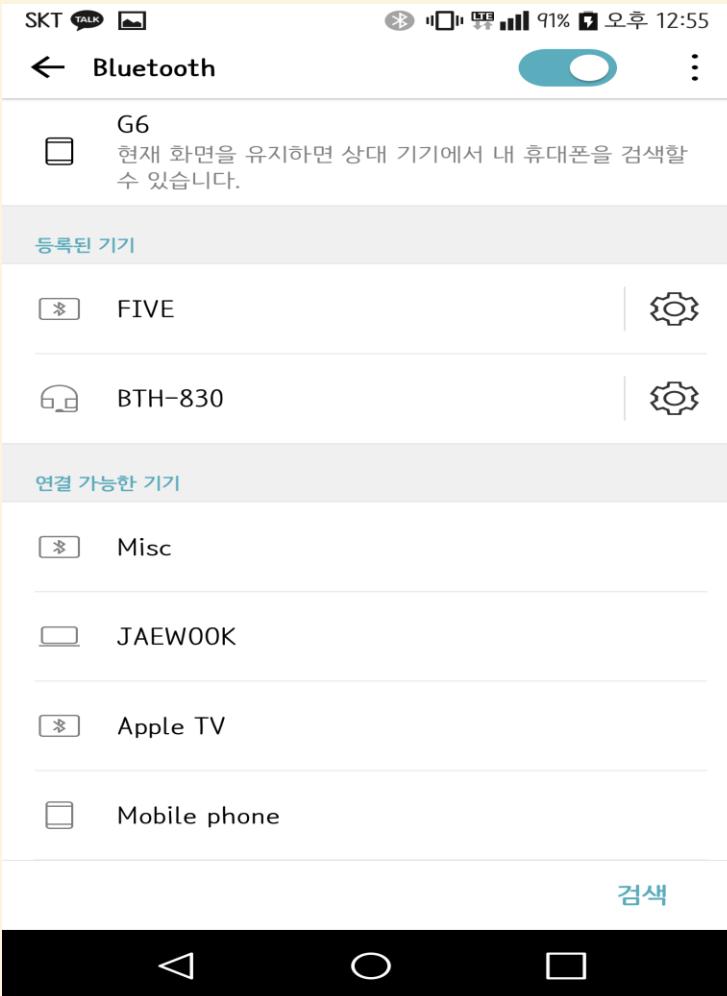
주요 설계/제작 진행상황

3. 실험 영상



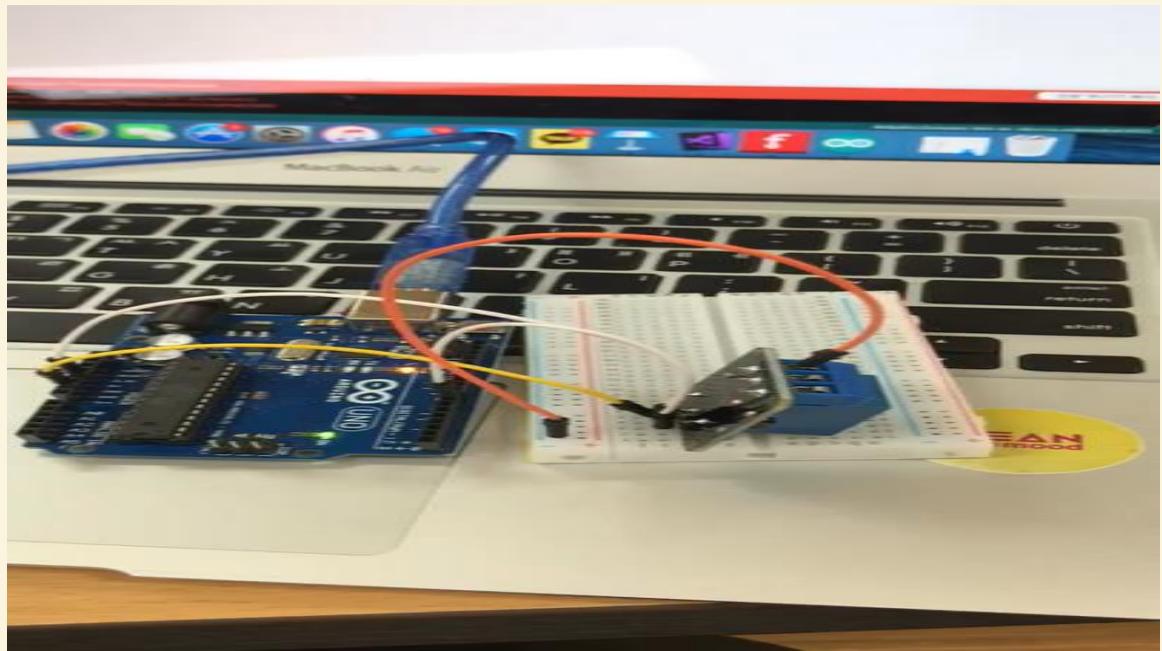
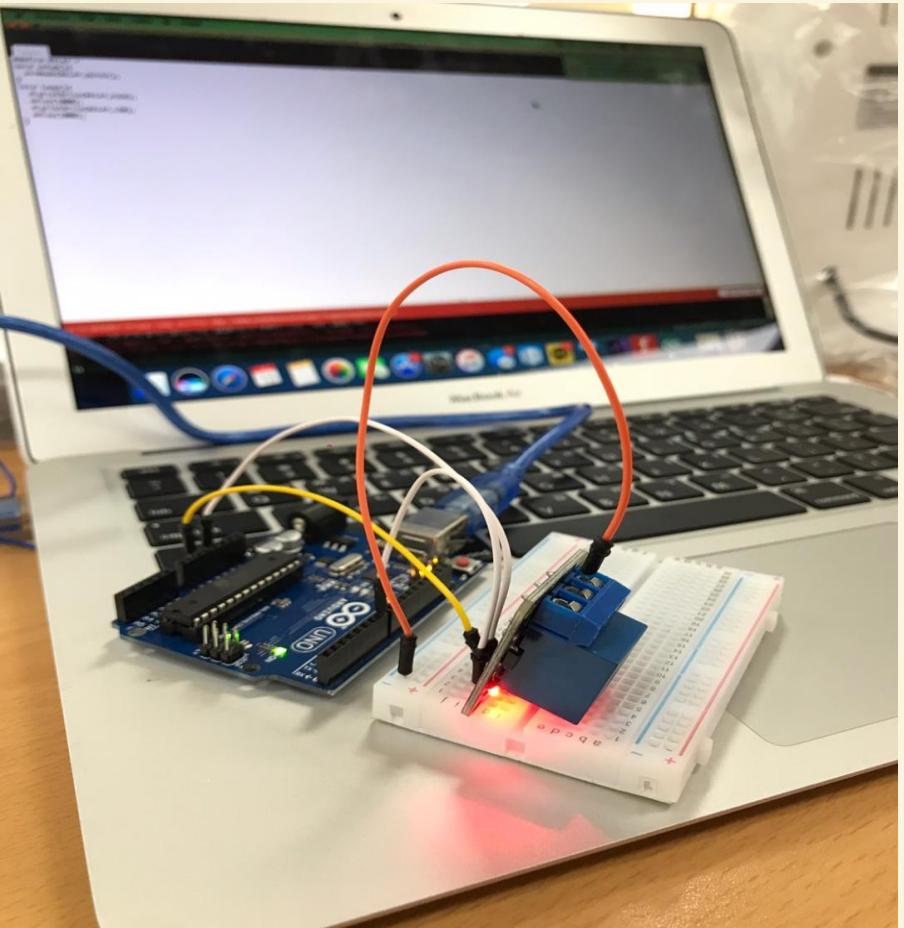
주요 설계/제작 진행상황

4. 앱인벤터로 블루투스 연결/ 센서 값 받아오기



주요 설계/제작 진행상황

5. 릴레이 실험



주요 설계/제작 진행상황

5. 릴레이 실험

```
#define RELAY 7
void setup(){
    pinMode(RELAY,OUTPUT);
}
void loop(){
    digitalWrite(RELAY,HIGH);
    delay(1000);
    digitalWrite(RELAY,LOW);
    delay(1000);
}
```

팀원 별 구체적인 역할

17011849 박영주 - 블루투스 연결, 최종 합본 스케치 보드 작성, 발표

17011871 하윤영 - 아두이노 회로 중 오류 해결(초음파/부저), 릴레이 실험

17011885 박세정 - 아두이노 회로 연결, 앱인벤터 화면스케치, PPT 작성

참고문헌 및 사이트

〈릴레이 제어〉

<https://blog.naver.com/compass1111/221288414858>

아두이노 센서 연결 및 블루투스는 전에 했던 과제 참고하여 연결함

진행 보고서

1 (10주차)

진행상황

<10주차 진행상황>

- 미세먼지 센서 연결 및 작동
- 초음파 센서 연결 및 작동
- 온습도 센서와 LCD 연결 및 작동

<추가구입>

- 아두이노 PM2.5 미세먼지센서 - 11,000원
- 브레드보드 830핀 - 1,290원

주별 설계//역할

		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 초음파 센서 연결 하윤영: 온습도 센서 연결
11주차		부품 연결	박영주: 스케치 코드 작성 박세정: 아두이노 회로 연결 하윤영: 아두이노 회로 연결
12주차		블루투스 이용하여 스마트폰 연결	박영주: 앱인벤터 제작 박세정: 블루투스연결 하윤영: 블루투스 스케치 보드 작성
13주차		스마트폰 어플 제작	박영주: 앱인벤터 제작 박세정: 앱인벤터 화면 디자인 하윤영: 앱인벤터와 아두이노 연결
14주차		실험	박영주: 스마트폰 블루투스 연결 박세정: 센서간 연결 하윤영: 스케치 보드 점검
15주차		시연 및 보완	
16주차		결과	

주별 설계//역할

		Task	팀원 구체적 역할
9주차		초안 작성 확정 및 부품구매	
10주차		센서 개별 실행 시도	박영주: 미세먼지 센서 연결 박세정: 온습도 센서 + LCD 연결 하윤영: 초음파 센서 연결
11주차		부품 연결	박영주: 스케치 코드 작성 박세정: 아두이노 회로 연결 하윤영: 아두이노 회로 연결
12주차		블루투스 이용하여 스마트폰 연결	박영주: 앱인벤터 제작 박세정: 블루투스연결 하윤영: 블루투스 스케치 보드 작성
13주차		스마트폰 어플 제작	박영주: 앱인벤터 제작 박세정: 앱인벤터 화면 디자인 하윤영: 앱인벤터와 아두이노 연결
14주차		전원 제어 및 실험	박영주: 원격제어 및 실험 박세정: 원격제어 및 실험 하윤영: 원격제어 및 실험
15주차		시연 및 보완	
16주차		결과	

주요 설계/제작 진행상황

1. 미세먼지 센서 제어

```
int dustPin=A0;
float dustVal=0;
float dustDensityug=0;
float calcVoltage=0;
int ledPower=2;

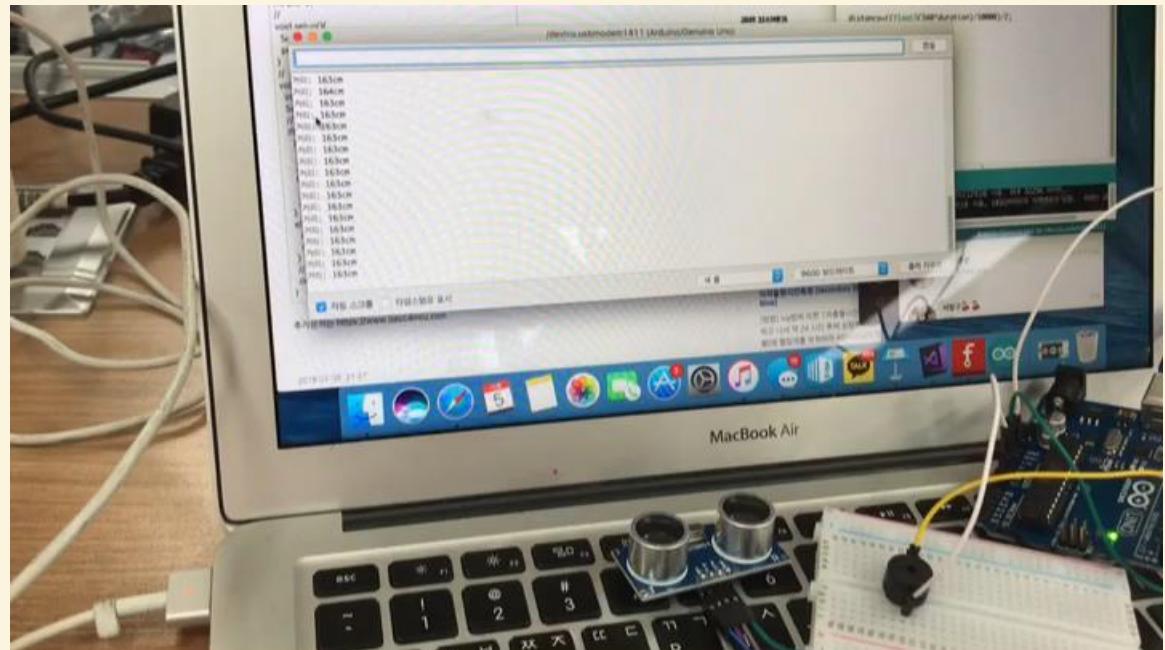
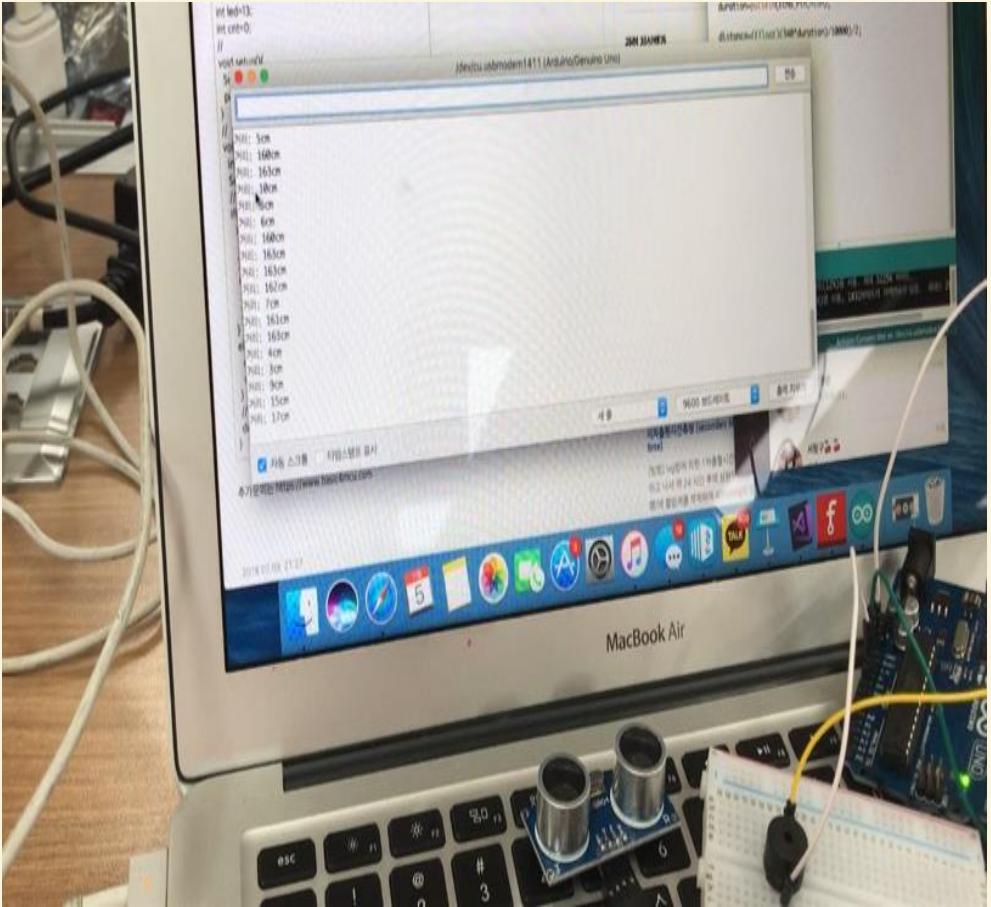
void setup() {
    Serial.begin(9600);
    pinMode(ledPower,OUTPUT);
}
```

```
void loop() {
    digitalWrite(ledPower,LOW);
    delayMicroseconds(280);
    dustVal=analogRead(dustPin);

    calcVoltage=dustVal*(5.0/1024); //전압 구하기
    delayMicroseconds(9680);
    digitalWrite(ledPower,HIGH);
    delayMicroseconds(9680);
    delay(1000);
    dustDensityug=(0.17*calcVoltage-0.1)*1000;//전압을
    미세먼지 단위로 변환하기
    Serial.print("Dust density(ug/m3) =");
    Serial.println(dustDensityug);
    delay(500);
}
```

주요 설계/제작 진행상황

2. 초음파 센서 제어



주요 설계/제작 진행상황

2. 초음파 센서 제어

```
int Echo_Pin=13;
int Trig_Pin=12;
int ALARM=2;
int cnt=0;

void setup(){
    Serial.begin(9600);
    pinMode(Echo_Pin,INPUT);
    pinMode(Trig_Pin,OUTPUT);
}
```

```
void loop(){
    long duration,distance;
    digitalWrite(Trig_Pin,HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig_Pin,LOW);

    duration=pulseIn(Echo_Pin,HIGH);

    distance=((float)(340*duration)/10000)/2;
```

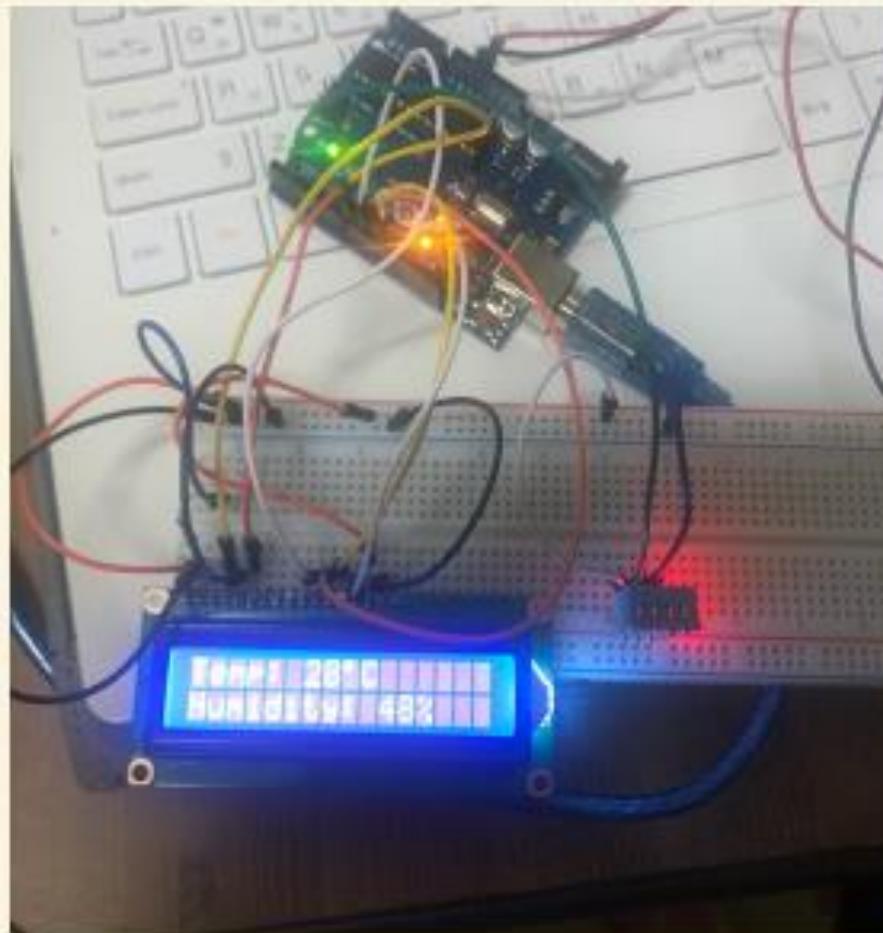
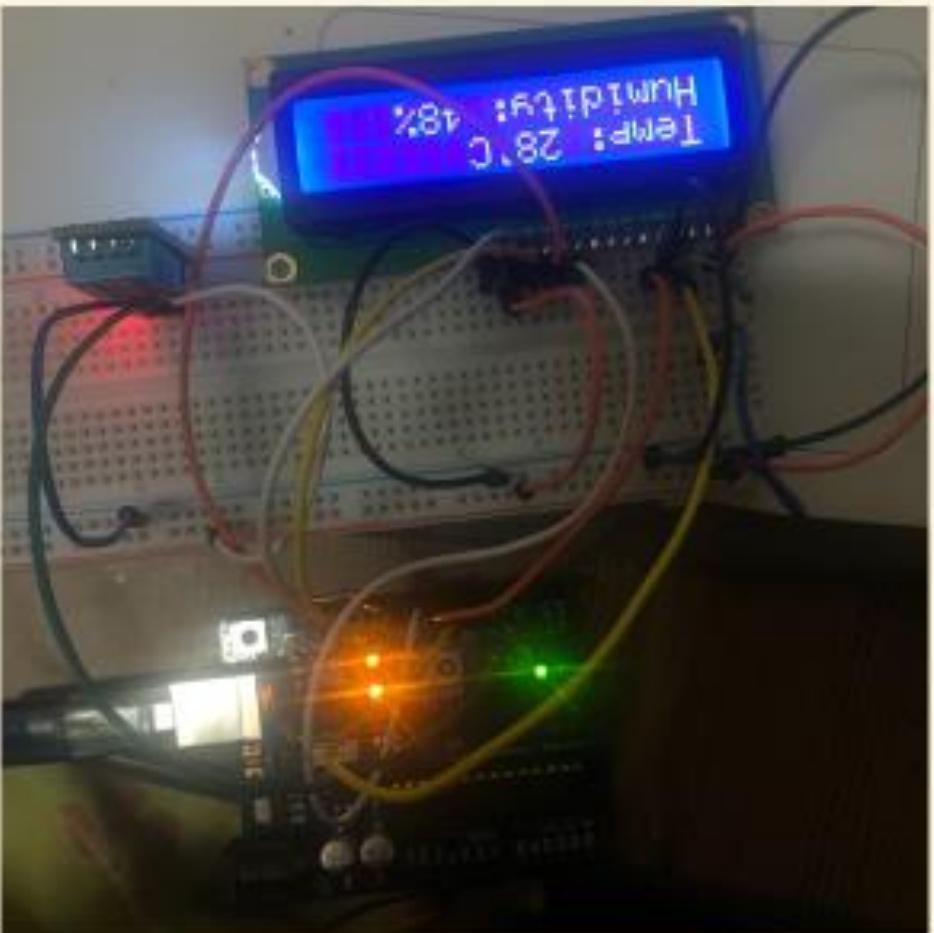
```
Serial.print("거리: ");
Serial.print(distance);
Serial.println("cm");

if(distance<=10){
    cnt++;
    if(cnt>=5){
        tone(ALARM,1000);
    }
}
else{
    noTone(ALARM);
    cnt=0;
}

delay(2000);
```

주요 설계/제작 진행상황

3. 온습도센서 + Lcd



주요 설계/제작 진행상황

3. 온습도센서 + Lcd

```
#include <LiquidCrystal.h>

#include <DHT.h>
#include <DHT_U.h>

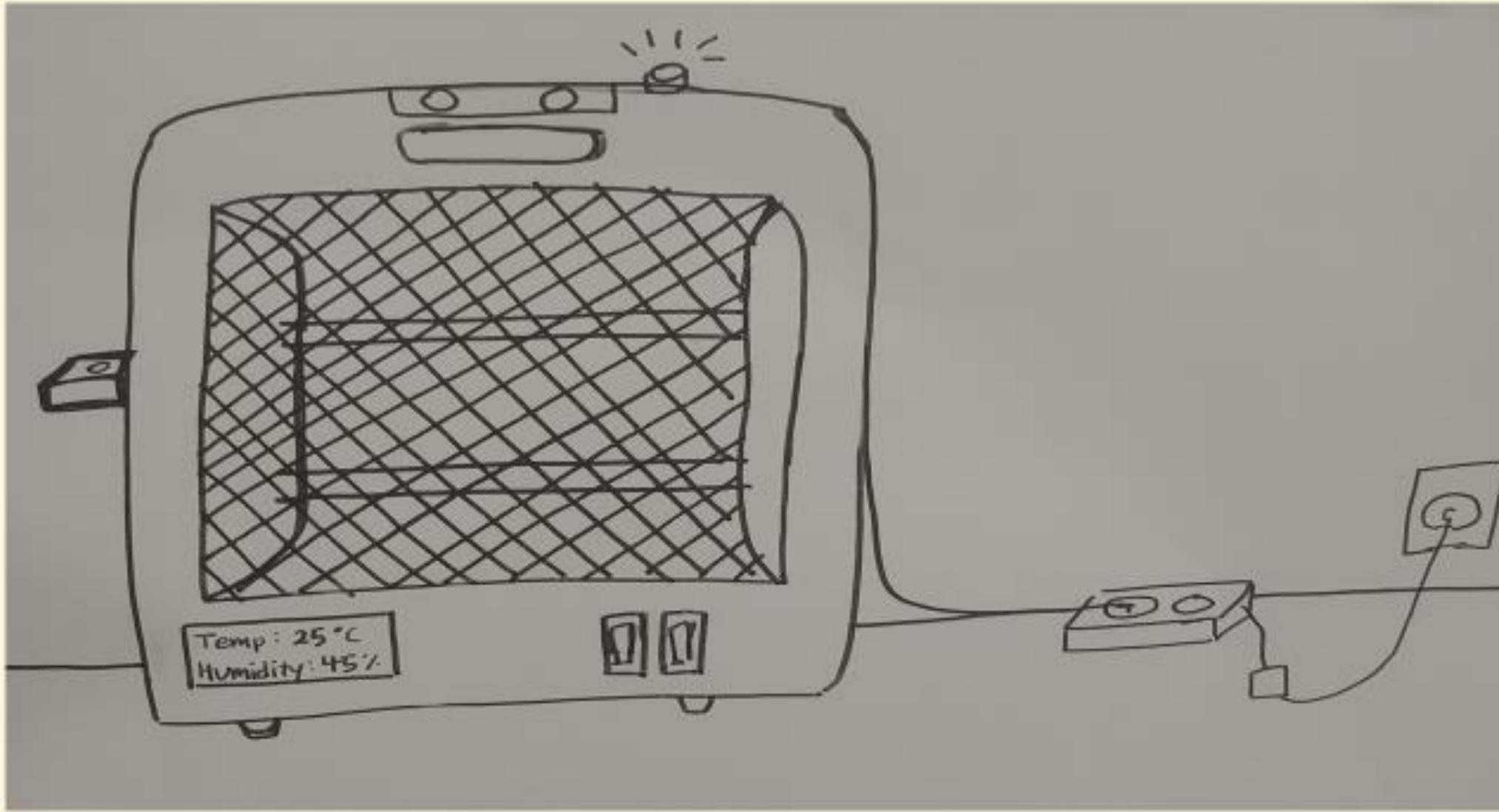
LiquidCrystal lcd(7,8,9,10,11,12);

#define DHT11_PIN 2
int Sensor_pin = 2;
DHT my_sensor(Sensor_pin,DHT11);

void setup(){
    lcd.begin(16,2);
    Serial.begin(9600);
    my_sensor.begin();
}
```

```
void loop(){
    int temp= my_sensor.readTemperature();
    int humi=my_sensor.readHumidity();
    lcd.setCursor(0,0);
    lcd.print("Temp: ");
    lcd.print(temp);
    lcd.print((char)223);
    lcd.print("C");
    lcd.setCursor(0,1);
    lcd.print("Humidity: ");
    lcd.print(humi);
    lcd.print("%");
    Serial.print("현재 온도: ");
    Serial.print(temp);
    Serial.print("/");
    Serial.print("현재 습도: ");
    Serial.println(humi);
    delay(2000);
}
```

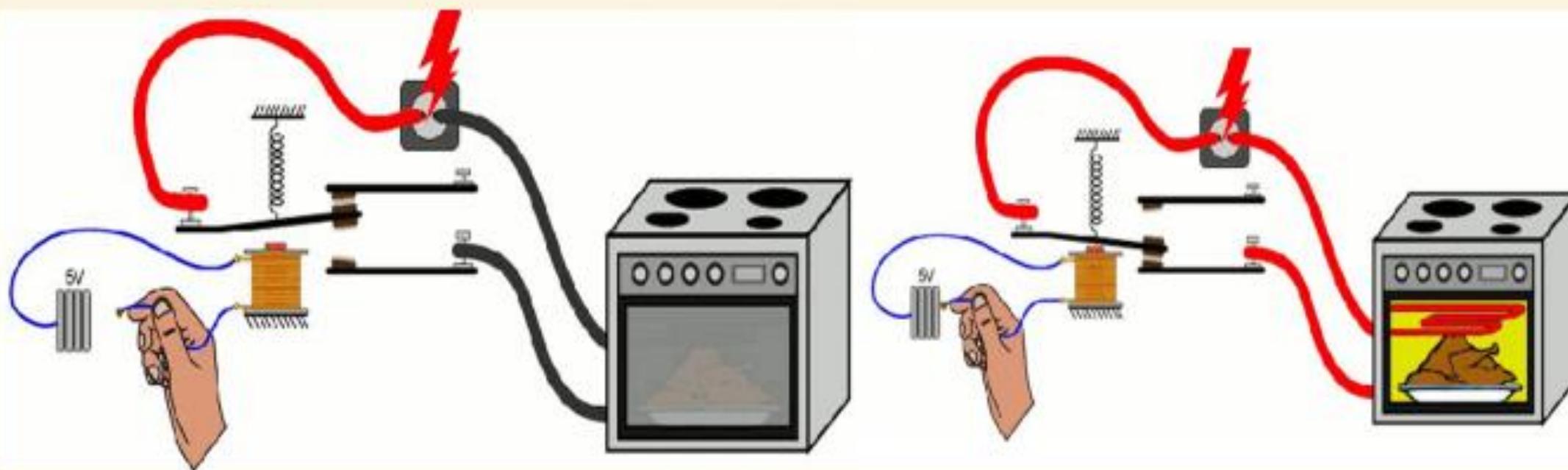
“하이트 더 하이터” 제품 스케치



On/Off 전원제어

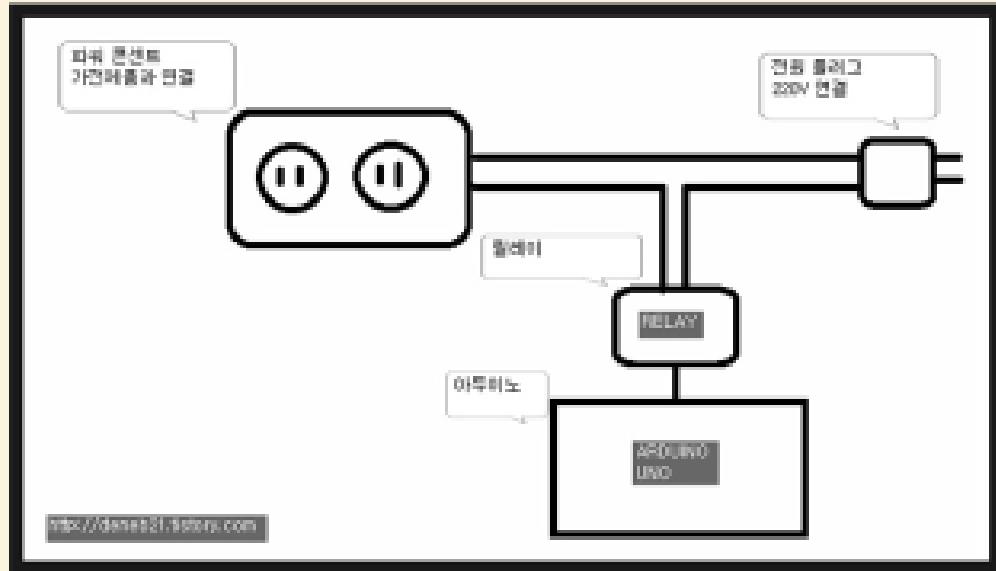
제안 1

릴레이를 이용하여 220V 전원 제어



On/Off 전원제어

제안 1



릴레이에 5V 전기가 가해지면(릴레이 규격에 따라 동작 전압이 다름) 릴레이 안의 전자석이 자성을 띄어 스위치를 ON 한다.
5V전기를 끊으면 전자석이 자성을 잃고 스프링에 의해 원복되면서 전류도 끊어지게 된다.

On/Off 전원제어

제안 2

서브 모터를 이용하여 스위치 On/Off

EX)



[https://m.blog.naver.com/PostView.nhn?
blogId=sbkdh267&logNo=22092035706
5&categoryNo=60&proxyReferer=&prox
yReferer=https%3A%2F%2Fwww.google.
co.kr%2F](https://m.blog.naver.com/PostView.nhn?blogId=sbkdh267&logNo=220920357065&categoryNo=60&proxyReferer=&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F)

팀원 별 구체적인 역할

17011849 박영주 - 미세먼지 센서 연결 및 작동 , 온습도 센서 + LCD 연결 및 작동

17011871 하윤영 - 초음파 센서 + 부저 센서 연결 및 작동, ppt 제작

17011885 박세정 - 온습도 센서 + LCD 연결 및 작동, ppt 제작, 발표

참고문헌 및 사이트

〈초음파센서+부저센서 참고 사이트〉

https://blog.naver.com/sumokto_2002/221308994547

〈미세먼지 센서 참고 사이트〉

<https://m.blog.naver.com/dnqlslaka/221125802242>

<http://www.makewith.co/page/project/989/story/2321/>

〈LCD+온습도 참고 사이트〉

<https://blog.naver.com/dsz08082/221321020369>

<https://blog.naver.com/dokkosam/221172166556>

참고문헌 및 사이트

〈릴레이를 이용한 전원 제어 참고 사이트〉

deneb21.tistory.com/m/222

<https://blog.naver.com/codingspecialist/221039892975>

www.makewith.co/page/project/309/story/630

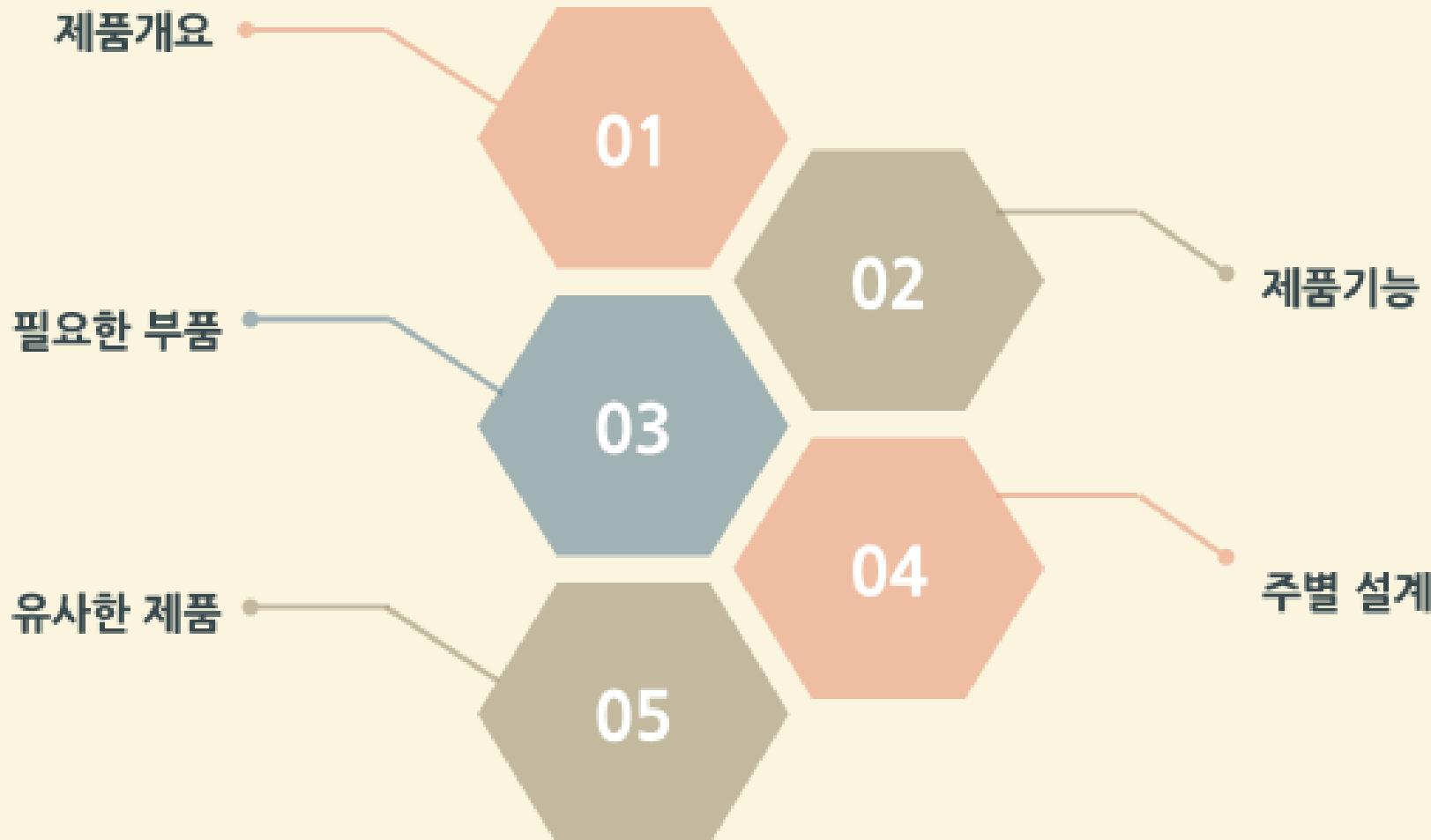
<https://blog.naver.com/codingspecialist/221039900151>

〈서브 모터 참고 사이트〉

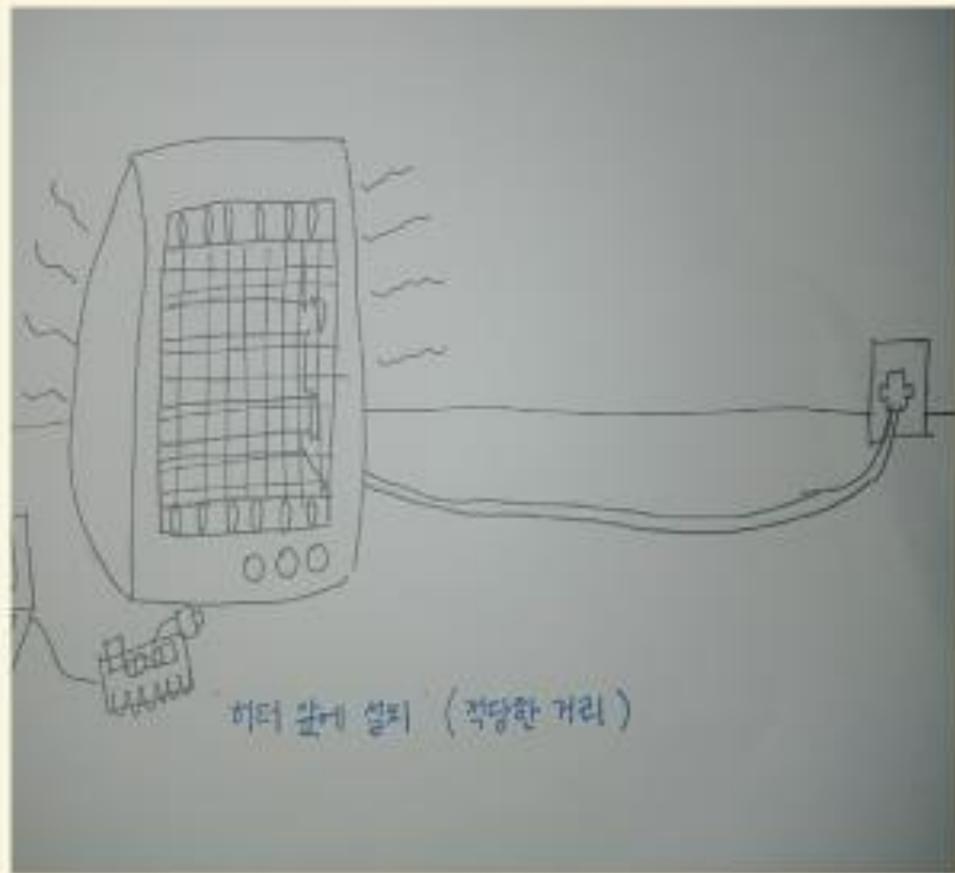
<https://m.blog.naver.com/PostView.nhn?blogId=sbkdh267&logNo=220920357065&categoryNo=60&proxyReferer=&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F>

설계 제작자

INDEX



제품개요



“히트 더 히터”

“난로를 가정이나 회사에서 안전하고 건강하게 사용한다”

날씨가 추워짐에 따라 난로 사용이 증가할 것이다.
가정이나 회사 내에서 사용하는 히터(전기 난로)는 저온화상이나
열로 인한 피부 질병에 노출 될 위험이 크다고 한다.
따라서 우리는 이를 예방하기 위해, 일정 안전 거리를 설정하여
이를 침범할 경우 위험경고를 알리며,
히터 필터에 끼는 먼지를 측정하는 기능을 추가하였다.
또한 온도와 습도를 측정하여 스마트폰 어플을 통해 확인 할 수 있으며
어플에 적정온습도를 표시한다.

제품 기능



히터의 미세먼지 정도 표시



일정 거리 이상 다가올 시
위험 감지 알람



온/습도 측정

제품 기능



스마트폰을 통해 일별 사용량,
현재 온습도 표시



현재 온습도, 세기를
디스플레이로
표시



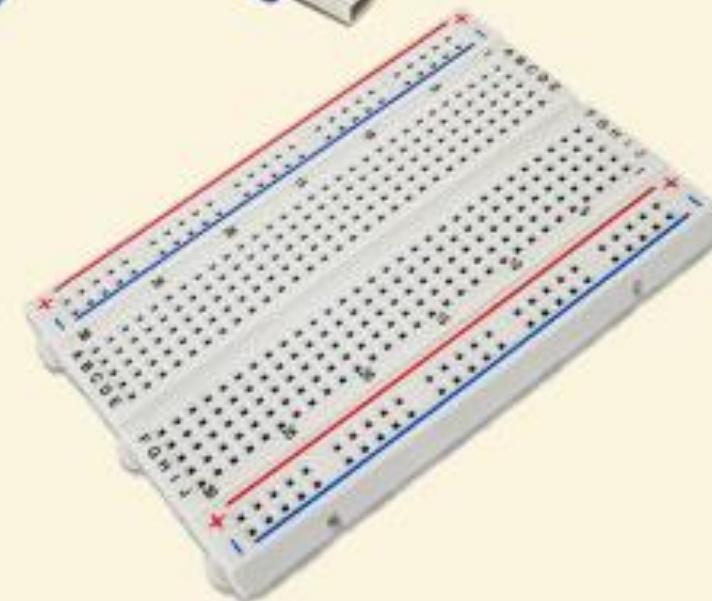
스마트폰을 이용한
원격 조정

제품 기능

- 1** 히터의 단위에 따른 전기 사용량 표시
- 2** 온습도 표시
- 3** 일정 금액 초과시 알람
- 4** 미세먼지 양 표시 및
일정 미세먼지 수치 초과 시 경고 메시지



필요한 부품



필요한 센서

아두이노 LM35D 아날로그 온도센서 (LM35D Analog Temperature Sensor Linear Module DC 4V-30V)

판매가격 : 2,200원 (부가세 포함)

상품코드 : [3205]ZAS-MIZ-C4

배송일 : 당일발송

남은수량 : 40개 이상

구매수량 : 1 개

추가&대량구매 요청 [053-588-4080](#)

바로구매

장바구니

NAVER

네이버ID로 간편구매
네이버페이

N Pay 구매

핀

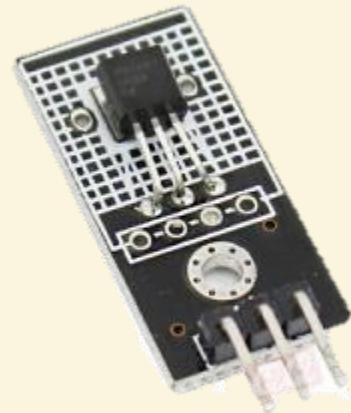
[계좌첫등록] 하면 1천원 드립니다!



페이코 ID

PAYCO 바로구매

울린다.



The screenshot shows a smartphone displaying a product page from m.smartstore.naver.com. The product is an 'Analog Temperature Sensor' (LM35D). The page includes a product image, a title, price information (5,130원), and purchase buttons for Naver Pay and a shopping cart. The phone's status bar at the top shows signal strength, battery level (86%), and the time (10:41).

온습도센서

온습도 센서를 이용해 설정 온도, 습도를 앱을 통해 표시한다.

아두이노 PM2.5 미세먼지센서 GP2Y1014AU, 먼지센서 어댑터 5,400원 16,600원 67%

5,130원 (우리카드) 청구할인카드 더보기 TOP

구매하기 장바구니



미세먼지센서

미세먼지 센서를 이용해 히터 내부의 필터를 측정한다.

센서 온도범위



-0~60°C

단, 초음파센서의 경우 온도변화에 따라
오차범위가 생길 수 있으므로
실습 과정에서 오차가 생길 시
적외선 센서로 변경할 가능성 있음



견딜 수 있을 것이라 예상



iot 히터



전체 이미지 뉴스 동영상 지도 더보기

설정 도구

검색결과 약 320,000개 (0.29초)

iot 히터 관리



스마트 히터



전체 이미지 뉴스 동영상 지도 더보기

설정 도구

사물인터넷

<https://protot.in>
사물인터넷
인터넷 우편으로

G마켓 - IoT

<http://www.gmarket.co.kr>
주문수 예상 배
에너지소비로드

이마트, 롯데

<http://www.lotteMart.co.kr>

히터 IoT



전체 이미지 뉴스 동영상 지도 더보기

설정 도구

검색결과 약 338,000개 (0.43초)

히터 IoT 보기

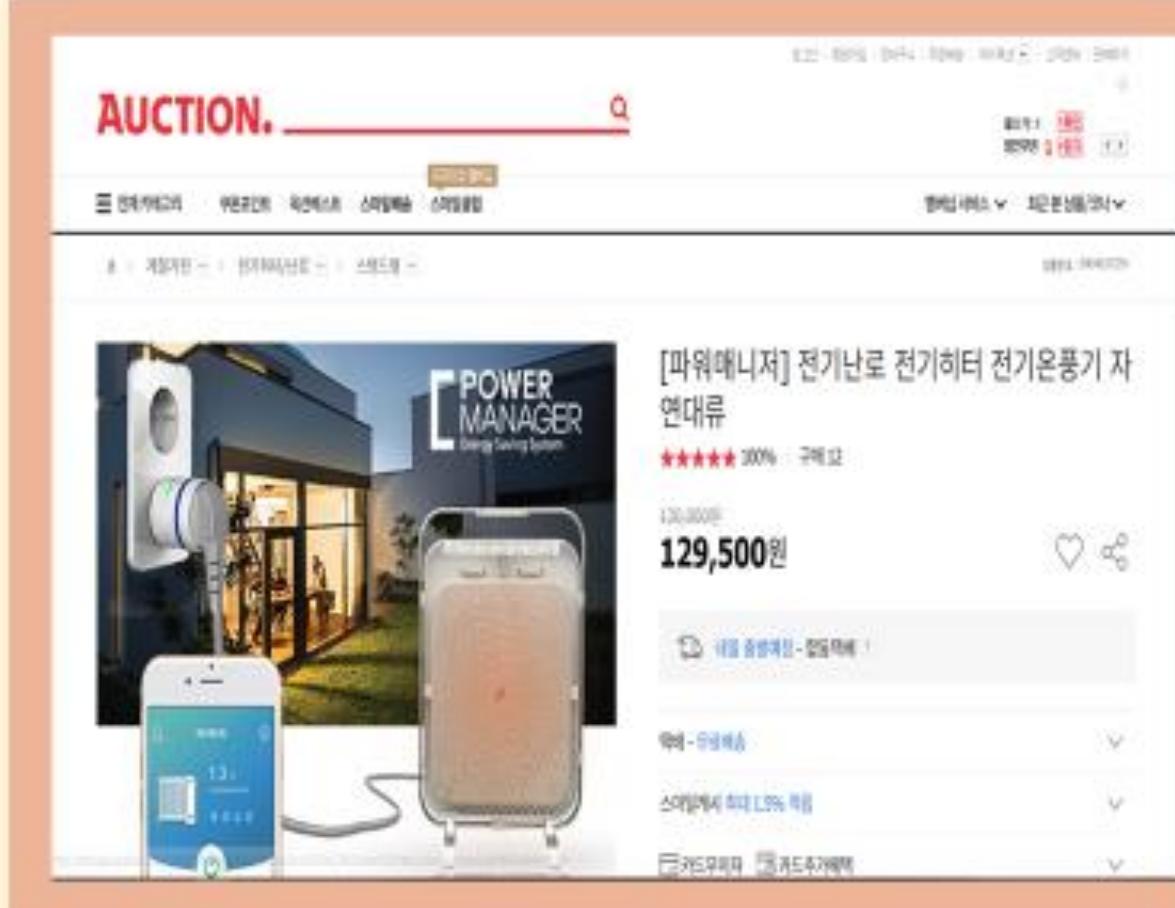
tv홈쇼핑
온풍기 히
₩449,800
Interpark
무료 배송

Google은 상품 판매의 당사자가 아닙니다.

히터 IoT 관련 이미지



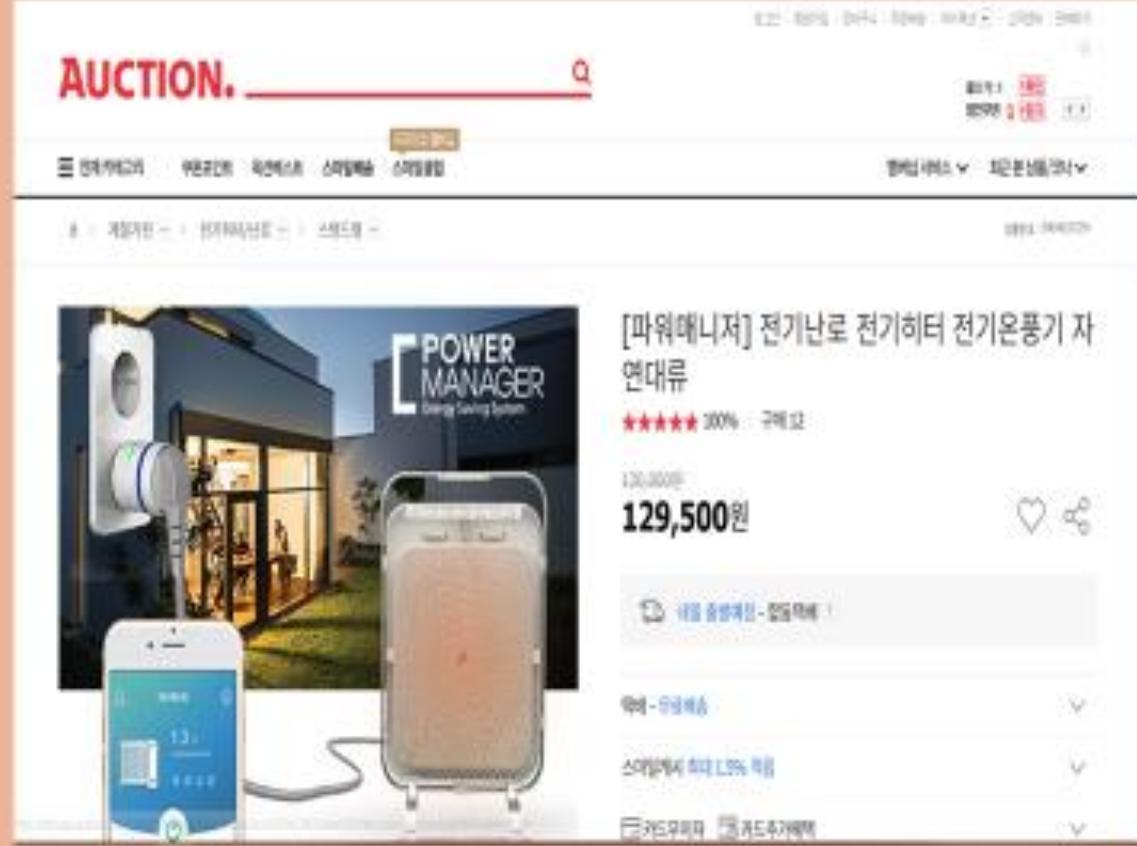
유사한 제품



히터 자체에 이용된 IoT 없음
이 히터의 기능: 친환경 원적외선 난방기

하지만 스마트 플러그에 꽂으면 스마트 난방기로 기능
→ 1. 히터 원격 on/off
2. 전자제품의 실시간 전기요금 및 전기사용량 확인
3. 대기전력 차단
4. 과열/과전압 차단

유사한 제품



The screenshot shows a product listing on a Korean auction platform. The product is a "POWER MANAGER Energy Saving System" (파워매니저 전기난로 전기히터 전기온풍기 자연대류). It has a 5-star rating (300 reviews) and is priced at 129,500원. The listing includes a smartphone displaying the app interface and a physical device with a remote control.

[파워매니저] 전기난로 전기히터 전기온풍기 자연대류
★★★★★ 300% 구매고
129,500원

따라서 유사한 기능은 히터에서 작동하는 것이 아니라 스마트 플러그를 이용해 서비스를 제공

검색 결과 스마트 히터, iot 히터 제품으로 시중에 나온 결과가 없음

Q&A