

교과목 포트폴리오

PA 20191779 김시연

목차

1. 졸업 작품

- Oled와 블루투스 동시 연결
- 센서로 측정한 값 안드로이드로 전송
- 안드로이드로 수중모터펌프 제어하기

2. 졸업 작품을 마치고

- 간단한 소감

1

졸업 작품

- Oled와 블루투스 동시 연결
- 센서로 측정한 값 안드로이드로 넘기기
- 안드로이드로수중모터펌프 제어하기

1 졸업 작품

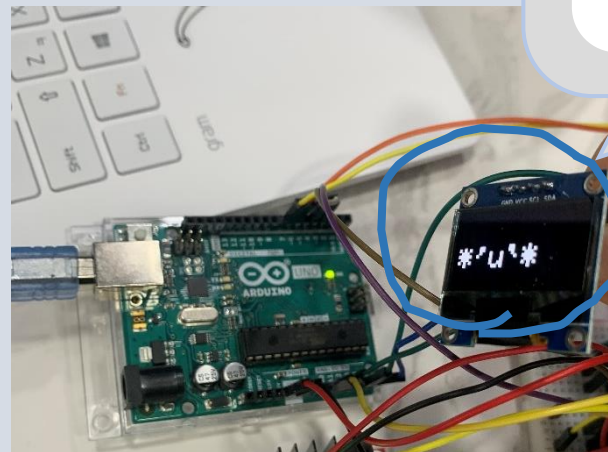
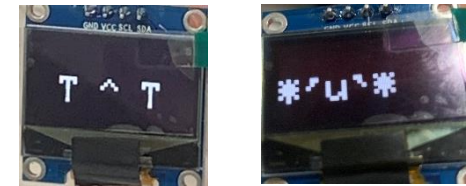
- Oled와 블루투스 동시 연결 : Oled

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
...
#define OLED_RESET -1
// 리셋핀 #(또는 -1 아두이노와 리셋핀을 연결하는 경우)
...
void setup() {
  Serial.begin(9600); // 디버깅 메시지 출력을 위해 시리얼 모니터 활성화
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D
    for 128x64
    Serial.println(F("SSD1306 Not Connected"));
    for(;;);
    // SSD1306에 주소할당이 되지 않으면 무한루프
  }
```

```
// 버퍼 지움
display.clearDisplay();
// 텍스트 출력
display.setTextSize(3);
display.setTextColor(WHITE);
```

```
display.setCursor(0,27);
display.println("*'u`*");
//display.println("T ^ T")
display.display();
delay(2000);
display.clearDisplay();
}
```

OLED 화면



1 졸업 작품

- Oled와 블루투스 동시 연결 : Bluetooth

```
// hc-06_01.ino
#include <SoftwareSerial.h>

Int Tx=6;
Int Rx=7;
SoftwareSerial btSerial(Tx,Rx);

void setup() {
  Serial.begin(9600); //시리얼 초기화
  btSerial.begin(9600);
}

void loop() {
  if(btSerial.available()) {
    /*블루투스 통신이 성공할 경우
    센서로 읽은 값을
    문자열로 안드로이드에 값을 전송하여
    안드로이드에 값을 나타내도록 코드 작성*/
  }
}
```



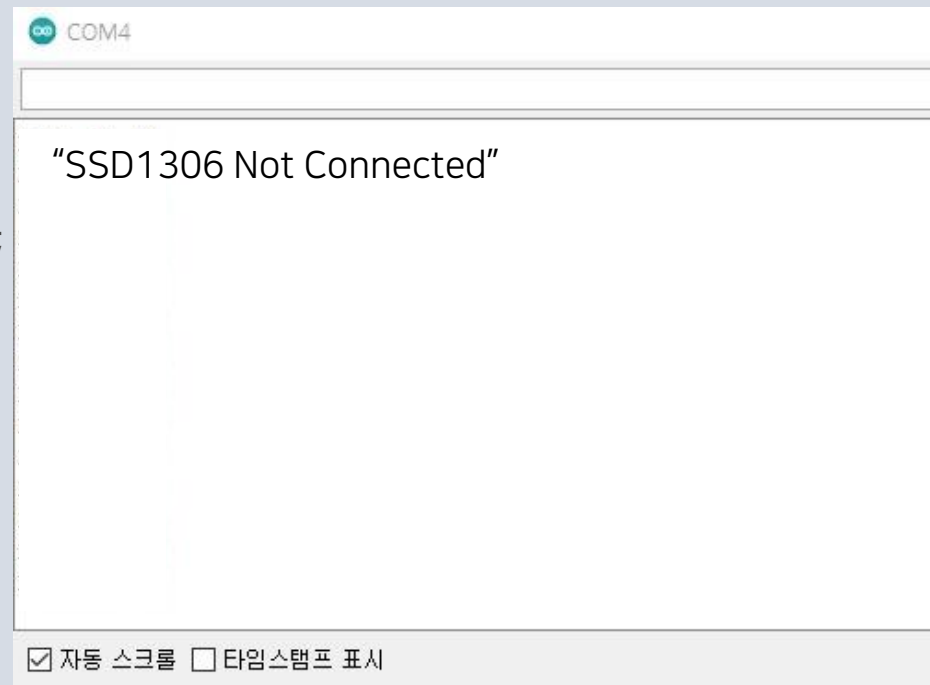
1 졸업 작품

- Oled와 블루투스 동시 연결 : 오류 발생

동시에 연결하고 작동하면 오류가 뜨는 문제가 발생

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
...
#define OLED_RESET -1
// 리셋핀 #(또는 -1 아두이노와 리셋핀을 연결하
// 는 경우)
...
void setup() {
  Serial.begin(9600); // 디버깅 메시지 출력을 위
  해 시리얼 모니터 활성화
  // SSD1306_SWITCHCAPVCC = 내부 3.3V 차
  지 펌프 회로를 켜둔다.
  if(!display.begin(SSD1306_SWITCHCAPVCC,
  0x3D)) { // Address 0x3D for 128x64
  Serial.println(F("SSD1306 Not
  Connected"));
  for(;;); // SSD1306에 주소할당이 되지 않으면
  무한루프
  }}

void loop(){
  // 버퍼 지움
  display.clearDisplay();
  // 텍스트 출력
  display.setTextSize(3);
  display.setTextColor(WHITE);
  display.setCursor(0,27);
  //display.println("'u `*");
  display.println("T ^ T");
  display.display();
  delay(2000);
}
```



1 졸업 작품

- Oled와 블루투스 동시 연결 : 오류 해결

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

```
SoftwareSerial btSerial(Tx,Rx);
#define OLED_RESET 4
// 리셋핀 # (또는 4 아두이노와 리셋핀을 연결하는 경우)
```

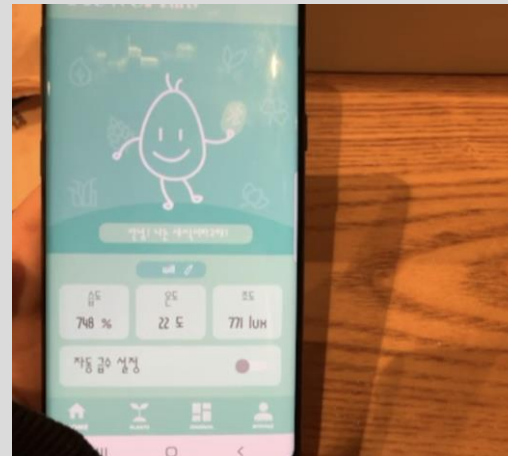
```
Void setup(){
  Serial.begin(9600);
  btSerial.begin(9600);
  Serial.setTimeout(50);          //블루투스 통신 속도
```

```
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.display();
  // 버퍼 지움
  display.clearDisplay();
}
```

```
Voie loop(){
  ... 블루투스를 활용한 제어 코드
}
```

```
SoftwareSerial btSerial(Tx,Rx);
#define OLED_RESET -1
// 리셋핀 # (또는 -1 아두이노와 리셋핀을 연결하는 경우)
```

-1이었던 리셋핀을 -1이 아닌 다른 숫자로 바꾼 후 작동 해보았더니 블루투스와 oled가 함께 작동하였다.



조건에 맞지 않아
슬픈 표정을 하고 있는 Oled

1 졸업 작품

- 센서로 측정한 값 안드로이드로 전송

```
Serial.println(btSerial.read()); // 켜는게 48 끄는게 49 //블루투스 값  
delay(1000);
```

```
/* to android through bt */ //시리얼모니터에 센서값 전송
```

```
btSerial.print((String)jodo);  
btSerial.print(",");  
btSerial.print((String)sensorVal);  
btSerial.print(",");  
int temp = (int)CalcTemp(_rawObject);  
btSerial.print((String)temp);  
btSerial.print(",");  
delay(1);  
if(GetObject()) {  
    delay(500);  
}
```

- Jodo = 조도 값을 나타내는 변수
- sensorVal = 수분 값을 나타내는 변수
- Temp = 온도를 나타내는 변수

각 변수를 지정하여 블루투스로
값을 읽을 수 있도록 코드를 작성한 후
","를 기준으로 순서대로 jodo, sensorVal, temp를
읽어올 수 있도록 안드로이드 상에서 코딩하였다.

1 졸업 작품

- 안드로이드로 수중모터펌프 제어 구현



안드로이드 화면 상에 수중 모터 펌프 작동을 제어할 수 있는 버튼을 구현하여

사용자가 원하면 버튼을 ON하여 일정 조건에 부합하면
자동으로 물을 줄 수 있도록 하고,

사용자가 원하지 않을 시 OFF하여
ON으로 바뀌기 전까지 자동 급수 기능이 작동되지 않도록 구현하였다.

(자동 급수 기능만 작동되지 않고
센서로 측정한 값은 항상 안드로이드로 보낼 수 있도록 구현하였다.)

1 졸업 작품

- 안드로이드로 수중모터펌프 제어 코드 작성 : 오류 발생

```
if(btSerial.available()) {  
    char bt;  
    bt = btSerial.read();  
    if(bt == '1'){  
        if(sensorVal >= 650){  
            Motor(LOW,0); //150의 출력으로 정방향 회전  
            delay(1000); //3000ms 즉 3초간 대기  
            Motor(LOW,0); //255(최대)의 출력으로 정방향 회전  
            delay(1000);  
            Motor(LOW,0); //정지  
            delay(1000); }  
        while(bt == '0'){  
            if(sensorVal >= 650) {  
                Motor(HIGH,150); //150의 출력으로 정방향 회전  
                delay(1000); //3000ms 즉 3초간 대기  
                Motor(HIGH,255); //255(최대)의 출력으로 정방향 회전  
                delay(1000);  
                Motor(HIGH,0); //정지  
                delay(1000); }  
            else Motor(LOW,0); }  
            Motor(LOW,0);  
        }  
    }
```

안드로이드 상에서 버튼 ON이면 0을 보내고
OFF일 경우 1을 보내도록 코드 구현

오류 발생

- ① If 문에서 빠져 나오지 못함
- ② 펌프가 계속 작동하지 않음
(건조 상태로 유지해서 계속 돌아가야 하는게 정상)
- ③ 센서로 측정된 값이 처음 한 번만 들어오고
이후로는 들어오지 않음.

각종 오류 발생으로 코드문을 다시 작성하였다.

1 졸업 작품

- 안드로이드로 수중모터펌프 제어 코드 작성 : 오류 해결

```
Void setup(){
  bt = btSerial.read();      //블루투스 read문
  if(bt == 48) {
    blue = 0;                //급수조절 on
  } else if (bt == 49) {
    blue = 1;                //급수조절 off
  }
  Serial.println(blue);
  if(blue == 0) {
    Serial.print("블루투스 켜기 값 : " + blue);
    if(sensorVal >= 650) {
      Motor(HIGH,255); //150의 출력으로 정방향 회전
      delay(2000); //3000ms 즉 3초간 대기
      Motor(HIGH,0); //정지
      delay(1000);
    }
  } else if (blue == 1) {
    Serial.print("블루투스 끄기 값 : " + blue);
    Motor(LOW,0); //정지
    //delay(1000);
    Serial.println(blue);
  }
  delay(1000);
}
```

안드로이드 상에서 ON/OFF한 값을 불러와 값을 확인해보니
ON일 때는 48, OFF일 때는 49라는 값이 나타났다.
bt값이 48일 경우, blue 변수 값에 0을
bt값이 49일 경우, blue 변수 값에 1로 저장하도록 코드를 작성하였다.

변수 blue 값이 0이고,
습도 값이 650 이상(건조 상태)일 때
수중 모터 펌프가 작동하도록 코드를 작성하였다.

변수 blue 값이 1일 경우,
모터가 작동하지 않도록 코드를 작성하였다.



처음에는 원하는 방식으로 구현되도록 생각나는 대로 코드를 구현하였고, 계속하여 오류가 발생하여 통신시에 발생하는 값들을 시리얼 모니터로 출력하여 알아보고 그 값을 이용하여 제어하는 코드 구문을 다시 작성하였다. 구현하고자 하는 것들을 차근차근 보면서 하나하나 수정하여 코드 작성을 마무리하였다.

2 졸업 작품을 마치고

- 간단한 소감

2 졸업 작품을 마치고

- 간단한 소감

처음에는 어떻게 해야 할지 걱정이 많았는데 하면서 팀원들과 소통도 잘 되고
계획했던 것을 하나씩 성공하고 좋은 말을 많이 들어 하는 내내 그리고 완성하고 나서 매우 뿌듯한 활동이었다.
물론 모든 순간이 다 순탄하고 완벽했던 것은 아니었지만
그 과정을 통해 사람들과 소통하는 법도 배우고 새로운 정보에 대해 알게 되어 스스로에게 매우 의미 있는 시간이었다고 생각한다.
일학년 때는 관심이 많이 없었던 아두이노에 대해 더 자세하고 많은 것을 알게 되어 재밌는 활동으로 기억될 것 같다.

감사합니다