

온실 속 화초

최원빈, 오현창

목차

table of contents

- 1 개요
- 2 블록도 및 흐름도
- 3 코드 및 시연영상
- 4 보완점 및 고찰

1

개요

식량 자급률이 23%대 수준인 한국에서 농업생산성이 월등하고, 미래 식량으로 각광받는 GMO(유전자 변형)식물 생산이 불가피하다고 생각이 들었다.

하지만, GMO 식물은 까다로운 생산 조건을 가진다고 한다.

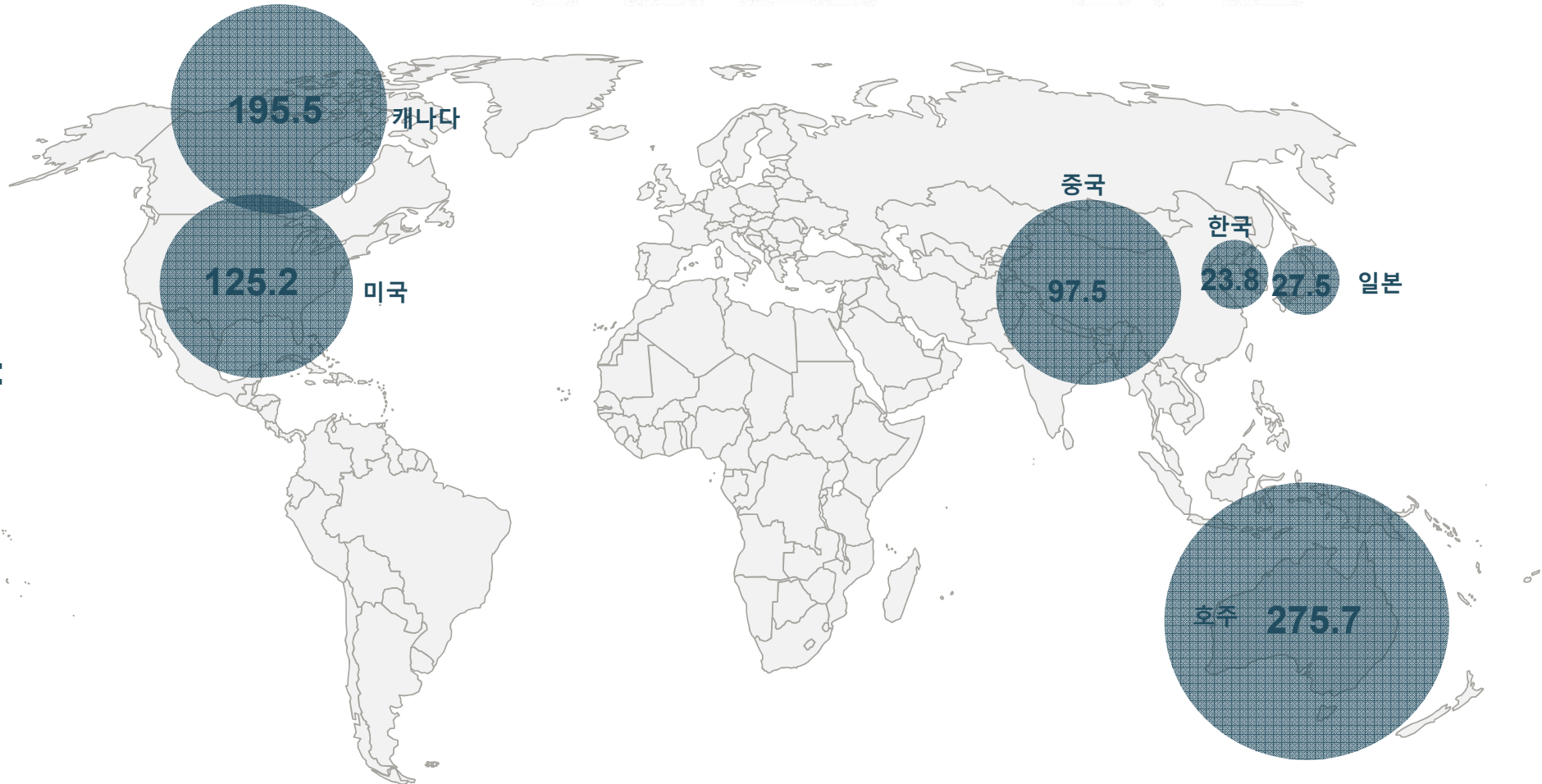
우리는 이러한 조건에 맞추어
자동화 온실 시스템을 구성하였다.

서버를 통해서 데이터를 관리하고,
주기적으로 모니터링 할 수가 있다.

또한, 자동시스템으로 즉각적인 대처가 가능하다.

곡물자급률

2021년 곡물자급률 18.5% '곤두박질'



세계 평균 :
102.5%

대한민국
자급률 :
23.8%

2

블록도 및 흐름도

보드 구성도



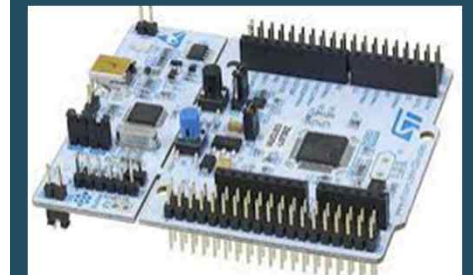
아두이노(환경값 측정)

>



서버(통신)

>

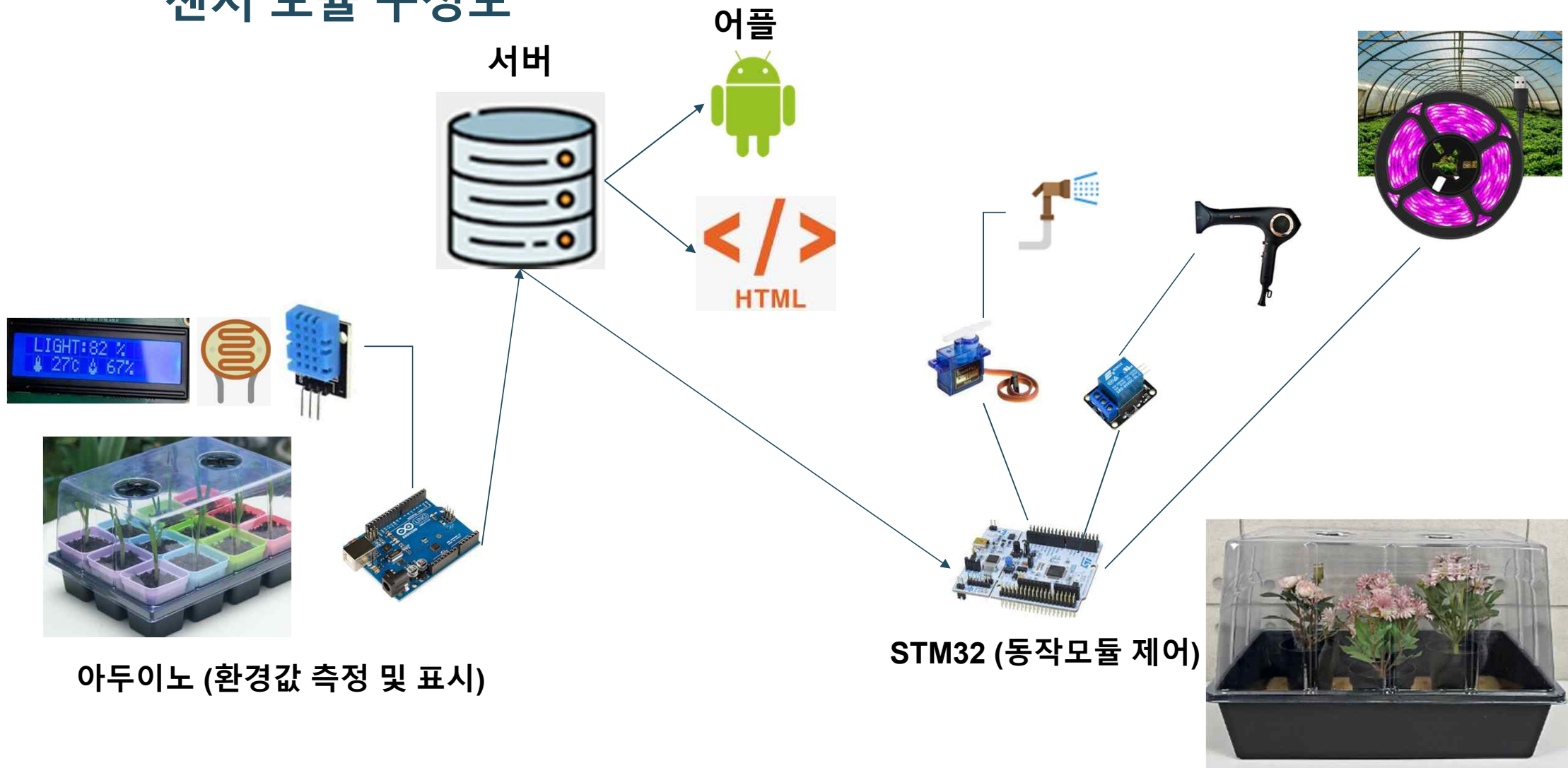


STM32(동작모듈 제어)

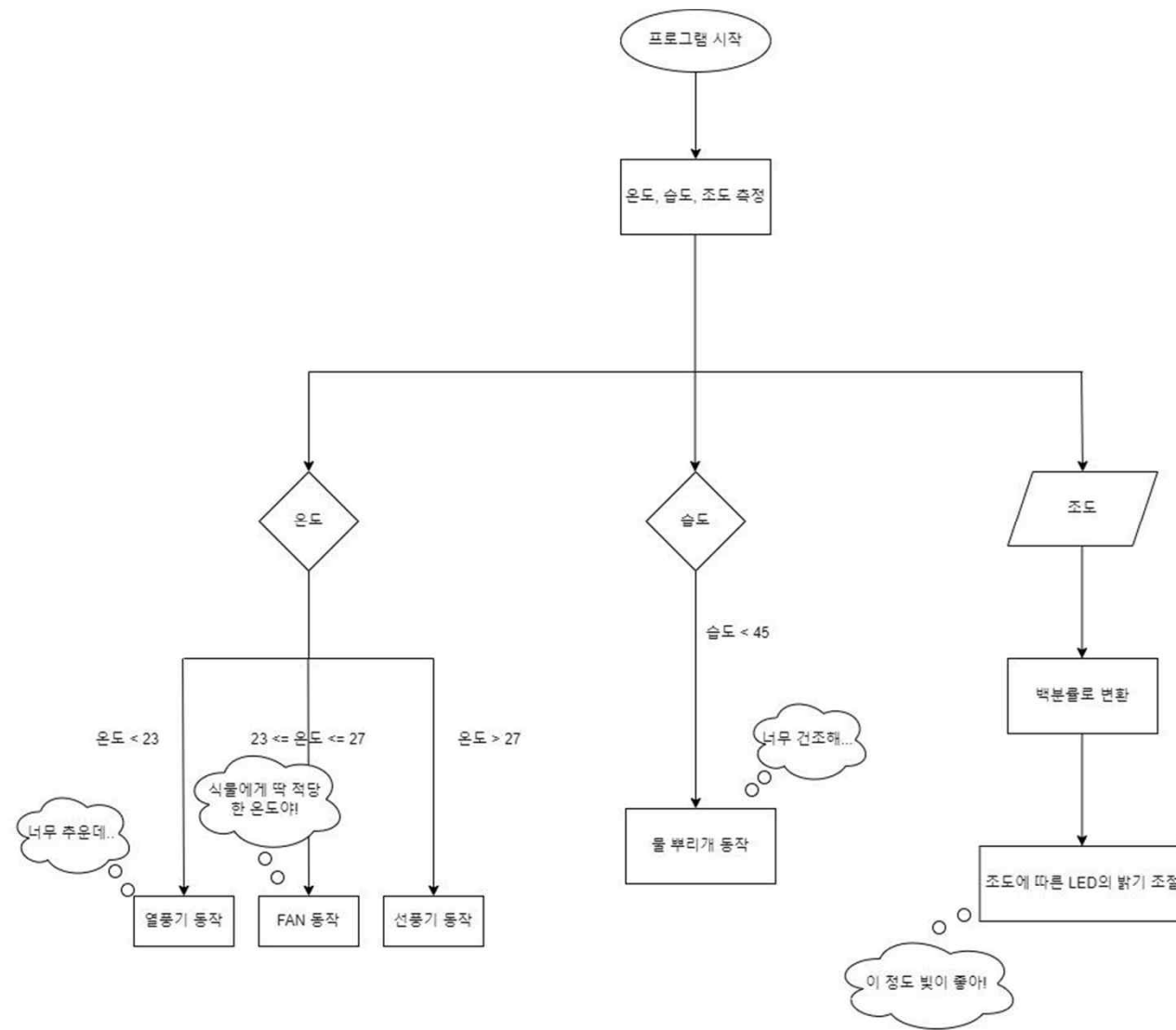
센서 모듈 구성도



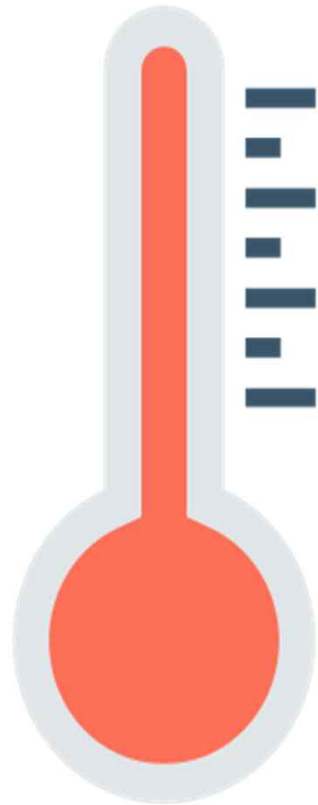
센서 모듈 구성도



센서 모듈 흐름도



모듈 흐름도(온도)



27°C 이상 선풍기 동작

적절한 온도 : 미니팬 동작

23°C 미만 드라이기 동작

3

코드 및
시연영상

코드(통신)

■ 블루투스 + 서버

```
// 블루투스 버퍼에 저장한 다음에 서버로 보내기
if ( now - before >= setime) {
    sprintf(sendBuf, "[%s]GMO@d@d@d\n", "KSH_BT", num1, num2, num3b);
    BTSerial.write(sendBuf, strlen(sendBuf)); //명령처리
    delay(1000);

    sprintf(sendBuf, "[%s]GMO@d@d@d\n", "OHC_SQL", num1, num2, num3b);
    BTSerial.write(sendBuf, strlen(sendBuf)); //DB전달
    before = now ;
}
```

```
MariaDB [iotdb]> select * from sensor;
```

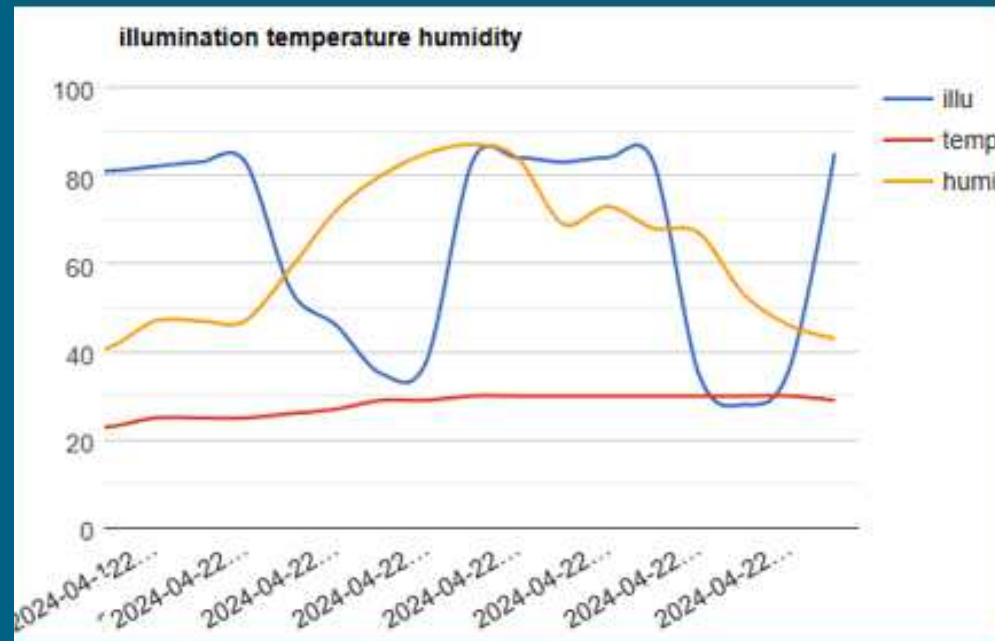
id	name	date	time	illu	temp	humi
437	OHC_BT	2024-04-19	16:17:25	83	23	41
438	OHC_BT	2024-04-19	16:17:35	83	23	41
439	OHC_BT	2024-04-19	16:17:46	83	23	41
440	OHC_BT	2024-04-19	16:17:56	83	23	41
441	OHC_BT	2024-04-19	16:18:06	83	23	41
442	OHC_BT	2024-04-19	16:18:16	82	23	41
443	OHC_BT	2024-04-19	16:18:26	82	23	41

```
7 rows in set (0.001 sec)
```

- 버퍼에 값을 저장한
다음 10초 마다 서버에
전달

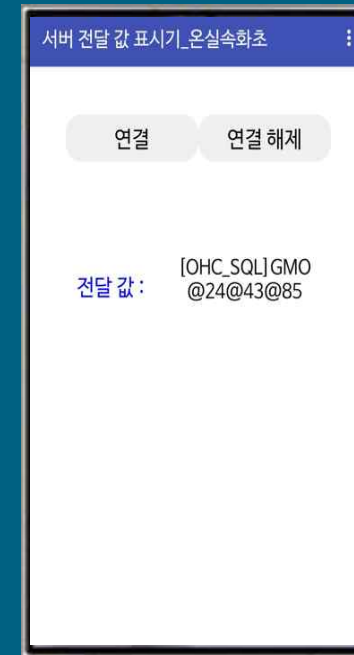
코드(통신)

- 블루투스 + 서버



코드(어플)

■ 어플 UI 와 블루투스 연결 및 동작 코드



코드(LCD)

- LCD 동작을 위한 코드
(온도, 습도, 조도)

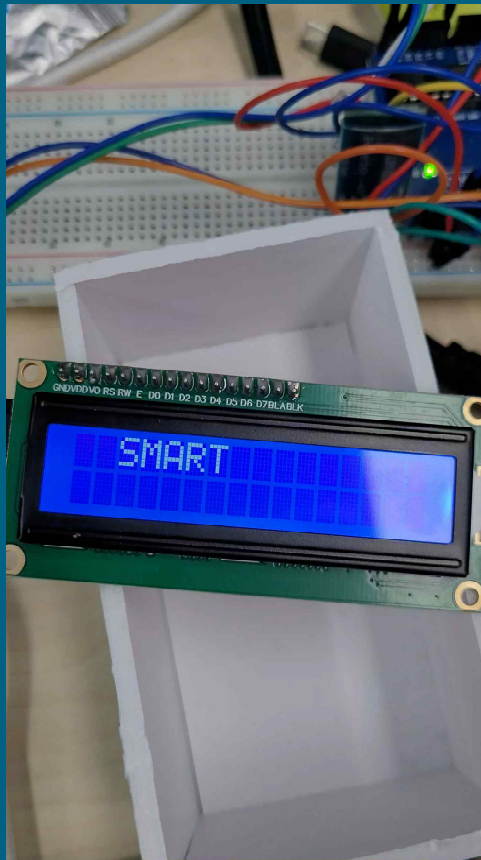
```
// 값을 환산
int num3b = map(num3, 0, 1023, 0, 100);

// 변환된 조도 값 출력(num3)
lcd.setCursor(1,0);
lcd.print("LIGHT:");
lcd.print(num3b);
delay(500);
lcd.setCursor(10,0);
lcd.print("%");

//LCD에 온도값(num1) 출력
lcd.setCursor(1,1);
lcd.write(0);
sprintf(str_1, "%02d", t_Value);
lcd.setCursor(3,1);
lcd.print(str_1);
lcd.write(1);

//LCD에 습도값(num2) 출력
lcd.setCursor(7,1);
lcd.write(2);
sprintf(str_2, "%02d", h_Value);
lcd.setCursor(9,1);
lcd.print(str_2);
lcd.print("%");
```


- LCD 시연영상



코드(온도)

■ 열풍기



- SIGNAL 핀에 1이 들어올 경우 COM과 NO핀의 회로를 닫아 끊어져 있는 전선에 220V가 흐를 수 있도록 제어

- 열풍기

```
if(num1 < 23){  
    HAL_GPIO_WritePin(RELAY_ON_OFF_GPIO_Port, RELAY_ON_OFF_Pin, 1);  
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 0);  
}
```

```
else if(!strcmp(pArray[1], "DRY"))  
{  
    if(!strcmp(pArray[2], "ON")){  
        HAL_GPIO_WritePin(RELAY_ON_OFF_GPIO_Port, RELAY_ON_OFF_Pin, 1);  
    }  
    else if(!strcmp(pArray[2], "OFF")){  
        HAL_GPIO_WritePin(RELAY_ON_OFF_GPIO_Port, RELAY_ON_OFF_Pin, 0);  
    }  
}
```

- 온도가 23도 미만이거나 명령을 줬을 경우 동작하도록 설정

- 열풍기 시연영상



코드(온도)

- 선풍기, 팬

```
// 선풍기 동작 세팅
if ( num1 > 27){
    pinMode(6,OUTPUT);
    digitalWrite(6,HIGH);
}
```

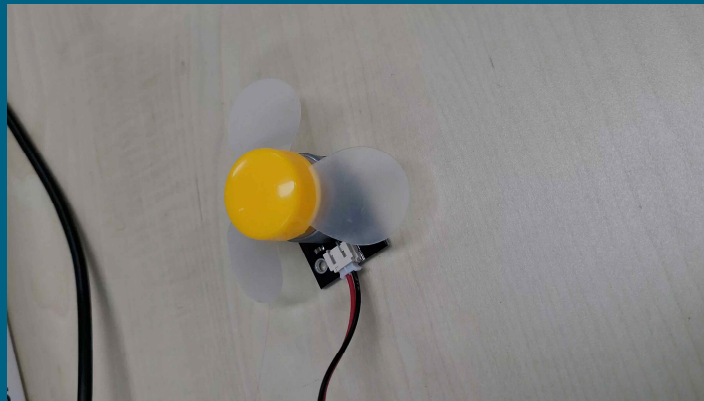
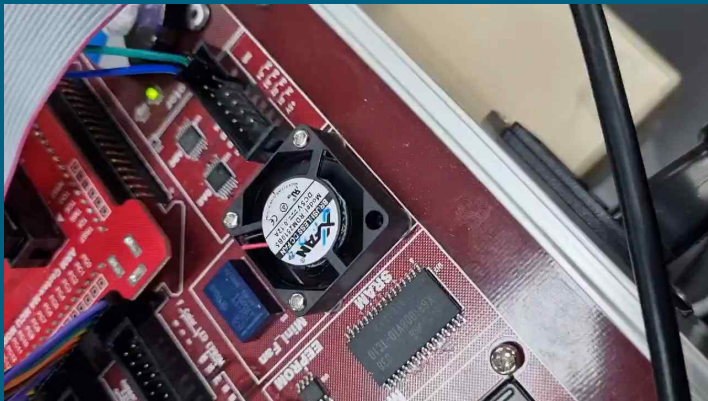
- 온도가 27도를 넘어갈 경우
선풍기 동작

```
}
else if(num1 >= 23 && num1 <= 27){
    HAL_GPIO_WritePin(RELAY_ON_OFF_GPIO_Port, RELAY_ON_OFF_Pin, 0);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);;
}
```

```
4 else if(!strcmp(pArray[1], "COOL"))
5 {
6     if(!strcmp(pArray[2], "ON")){
7         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
8     }
9     else if(!strcmp(pArray[2], "OFF")){
10        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 0);
11    }
12 }
```

- 적절한 온도이거나 명령어를
줬을 경우 릴레이를
이용하여 팬 동작

- 선풍기, 팬 시연영상



코드(습도)

- 스프레이(서보모터)

```
if(num2 < 45){  
    for(int k = 0; k < 10 ; k++){ // default = 10  
        __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_1,125); // degree 180  
        __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_2,125);  
        while(m_cnt - 5 < bt_sec){}  
        bt_sec = m_cnt;  
  
        __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_1,25); // degree 0  
        __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_2,25);  
        while(m_cnt - 5 < bt_sec){}  
        bt_sec = m_cnt;  
    }  
}
```

- 건조한 환경인 경우 서보 모터를 이용하여 스프레이 10회 분사

- 스프레이(서보모터)

- 직접 분사 횟수를 설정하여 동작 명령을 줄 수도 있음
- 분사 횟수를 설정 안 할 경우 디폴트 값으로 10회 분사

```
else if(!strcmp(pArray[1], "WATER"))
{
    if(pArray[2] != NULL){
        num2 = atoi(pArray[2]);
        for(int k = 0; k < num2 ; k++){ // user input water count
            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_1,125); // degree 180
            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_2,125);
            while(m_cnt - 5 < bt_sec){}
            bt_sec = m_cnt;

            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_1,25); // degree 0
            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_2,25);
            while(m_cnt - 5 < bt_sec){}
            bt_sec = m_cnt;
        }
    }
    else
        for(int k = 0; k < 10 ; k++){ // default = 10
            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_1,125); // degree 180
            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_2,125);
            while(m_cnt - 5 < bt_sec){}
            bt_sec = m_cnt;

            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_1,25); // degree 0
            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_2,25);
            while(m_cnt - 5 < bt_sec){}
            bt_sec = m_cnt;
        }
}
```


- 스프레이(서보모터) 시연영상



- LED STRIP을 이용한 조도 제어

```
int red = 255;  
int green = 0;  
int blue = 255;
```

```
int brightness = map(100 - num3, 0, 100, 0, 255);
```

```
red = (red * brightness) / 255;  
green = (green * brightness) / 255;  
blue = (blue * brightness) / 255;  
__HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_3,255 - red); // arduino LED input  
__HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_1,255 - green);  
__HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_2,255 - blue);
```

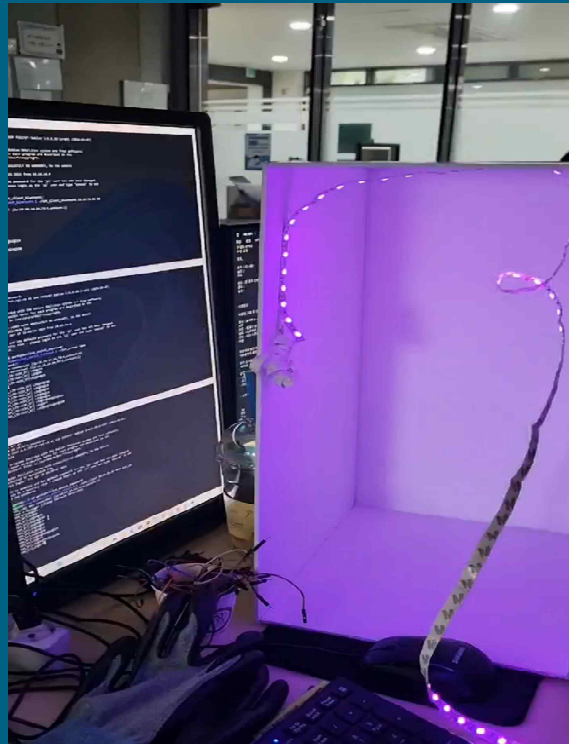
- 입력 받는 조도 값에 따라 LED 밝기 조절
- 식물 광합성에 도움이 되는 RED, BLUE 컬러만 활성화

- LED STRIP을 이용한 조도 제어

```
else if(!strcmp(pArray[1], "LED"))
{
    if(!strcmp(pArray[2], "ON")){
        if(pArray[3] != NULL)
            red = 255 - atoi(pArray[3]);
        else red = 0;
        if(pArray[4] != NULL)
            green = 255 - atoi(pArray[4]);
        else green = 0;
        if(pArray[5] != NULL)
            blue = 255 - atoi(pArray[5]);
        else blue = 0;
        __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_3, red);
        __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, green);
        __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2, blue);
    }
    else if(!strcmp(pArray[2], "OFF")){
        __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_3, 254);
        __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, 254);
        __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2, 254);
    }
}
```

- LED ON, OFF 명령과 직접 RGB 값을 입력하여 색상 조절 기능 구현

- LED STRIP 시연 영상





4

보완점 및 고찰

보완점 및 고찰

항목	최원빈	오현창
보완점	전선 상태 불량(피복)	하드웨어 구성에 대한 아쉬움
고찰	<p>내가 원하는 대로 동작이 되지 않을 때 소프트웨어인지 하드웨어인지 확실하지 않아서, 디버깅의 중요성을 느꼈다.</p> <p>하드웨어를 만질 땐 꼭 안전에 유의해야 할 것 같다.</p>	<p>직접 구현해보니, 생각처럼 되는게 잘 없었다. 접촉 불량을 염두에 두지 않아서, 시간을 많이 뺏긴 것 같다.</p> <p>자료조사를 하면서, 생각보다 식량난을 걱정해야 할 것 같았다.</p>



감사합니다.