



소스코드

METAPLACE

• MepL\_Farm TOWN •



# 스마트 팜



교육명

스마트농업 IoT 코딩 교육

교육기간

2023. 01. 03 – 2023. 1. 05

스마트팜은 정보통신기술(ICT)을 활용하여 농경지를 모니터링하고 조절함으로써 생산성을 향상시키며, 노동력과 자원 사용을 효율적으로 관리합니다.

스마트팜은 생산량 증가뿐만 아니라 농산물의 품질과 안전성 향상, 지속 가능한 농업을 실현하는 데에 기여합니다.



정보통신기술(ICT)을 활용하여 농업을 지능화하는 혁신적인 농업 방식



센서, 자동제어 시스템, 빅데이터, 인공지능 등의 첨단 기술을 도입





## PRODUCT PROCESS

스마트팜에서는 재배, 포장, 유통, 판매 과정이 모두 자동화와 최신 기술의 활용을 통해 효율성을 극대화하고, 농산물의 품질과 안전성을 보장하는데 중점이 두어집니다.

**센서 및 장치 배치:**

- 농경지에 다양한 종류의 센서와 IoT 장치를 배치합니다.
- 센서는 토양, 대기, 물 등의 다양한 환경 요소를 모니터링할 수 있습니다.

**데이터 수집:**

- 배치된 센서 및 장치는 실시간으로 데이터를 수집하고, 이를 중앙 데이터 플랫폼으로 전송합니다.
- 수집된 데이터는 온도, 습도, 토양 상태, 강수량 등 농작물 생육과 관련된 정보를 포함합니다.

**데이터 통합 및 분석:**

- 중앙 데이터 플랫폼에서는 수집된 데이터를 통합하고 분석합니다.
- 빅데이터 기술과 인공지능 알고리즘을 활용하여 유용한 정보를 도출하고 생육환경에 대한 통찰력을 얻습니다.

**자동제어 및 의사결정:**

- 분석된 데이터를 기반으로 자동제어 시스템이 환경 조절을 수행합니다.
- 농작물의 최적 생육 조건을 제공하기 위해 온도, 습도, 물 공급 등을 자동으로 조절합니다.

**원격 모니터링 및 관리:**

- 스마트폰 앱이나 웹 인터페이스를 통해 농가 주인은 농경지를 언제 어디서나 모니터링하고 관리할 수 있습니다.
- 원격으로 시스템을 제어하고 데이터를 확인하여 실시간으로 대응할 수 있습니다.





# MEPL\_FARM



## 제품명

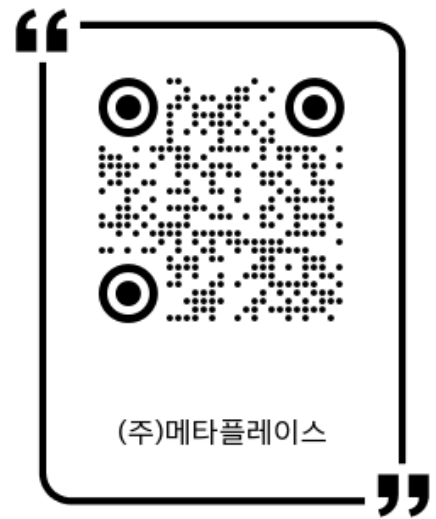
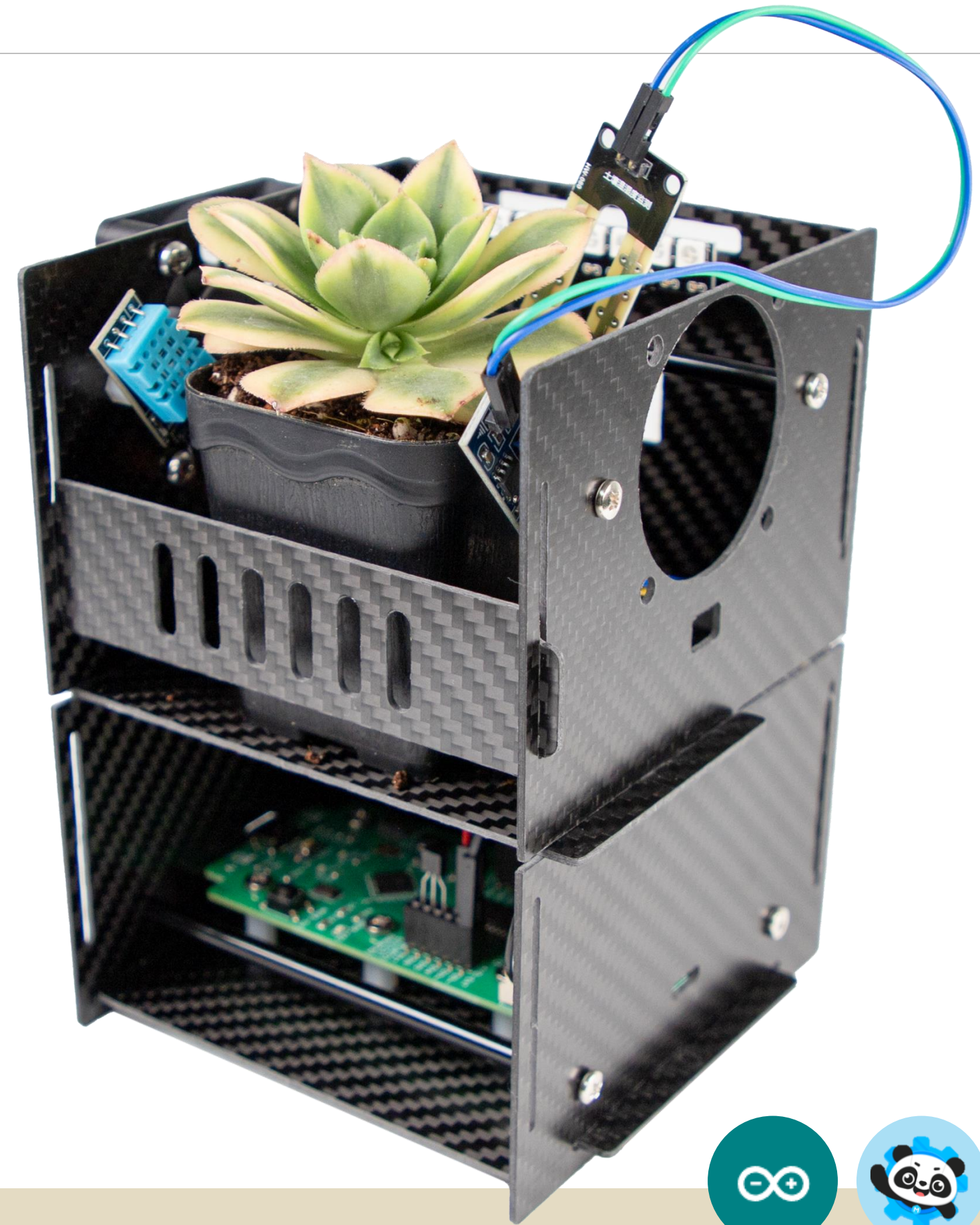
메플팜(MePL\_Farm)

## 기본사양

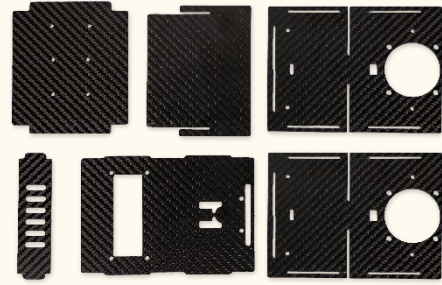
- 마이크로 보드
- Arduinio IDE로 프로그래밍
- (주)메타플레이스 메플보드

## 강의자료

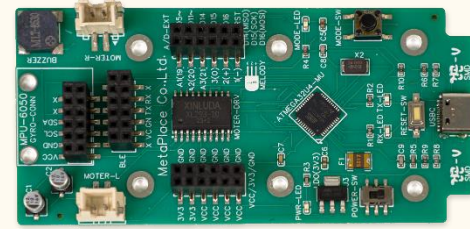
- 소스코드 : 깃허브
- 예제 및 교육자료 : [\(주\)메타플레이스 자료실](#)



# 파티머리



카본케이스



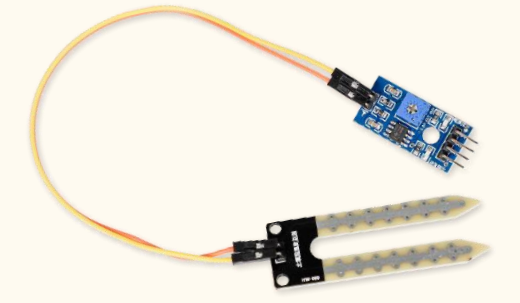
메플이노 보드



HM-10 블루투스



I2C LCD Display



토양습도센서



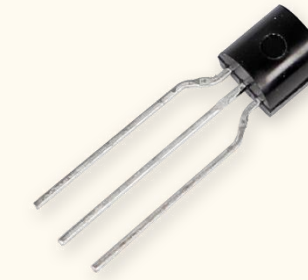
5V 쿨링팬



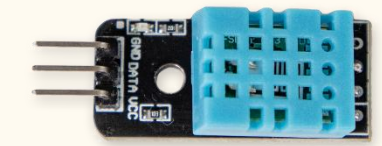
네오픽셀 LED



100mm 서포트



멜로디 IC



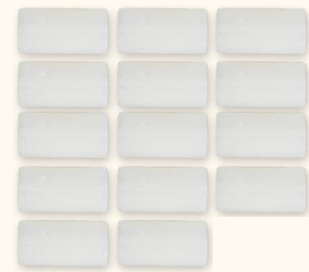
DHT11 온습도센서



쿨링팬 탭나사(4\*10mm)



3mm 나사



11mm 서포트(14개)



점퍼케이블



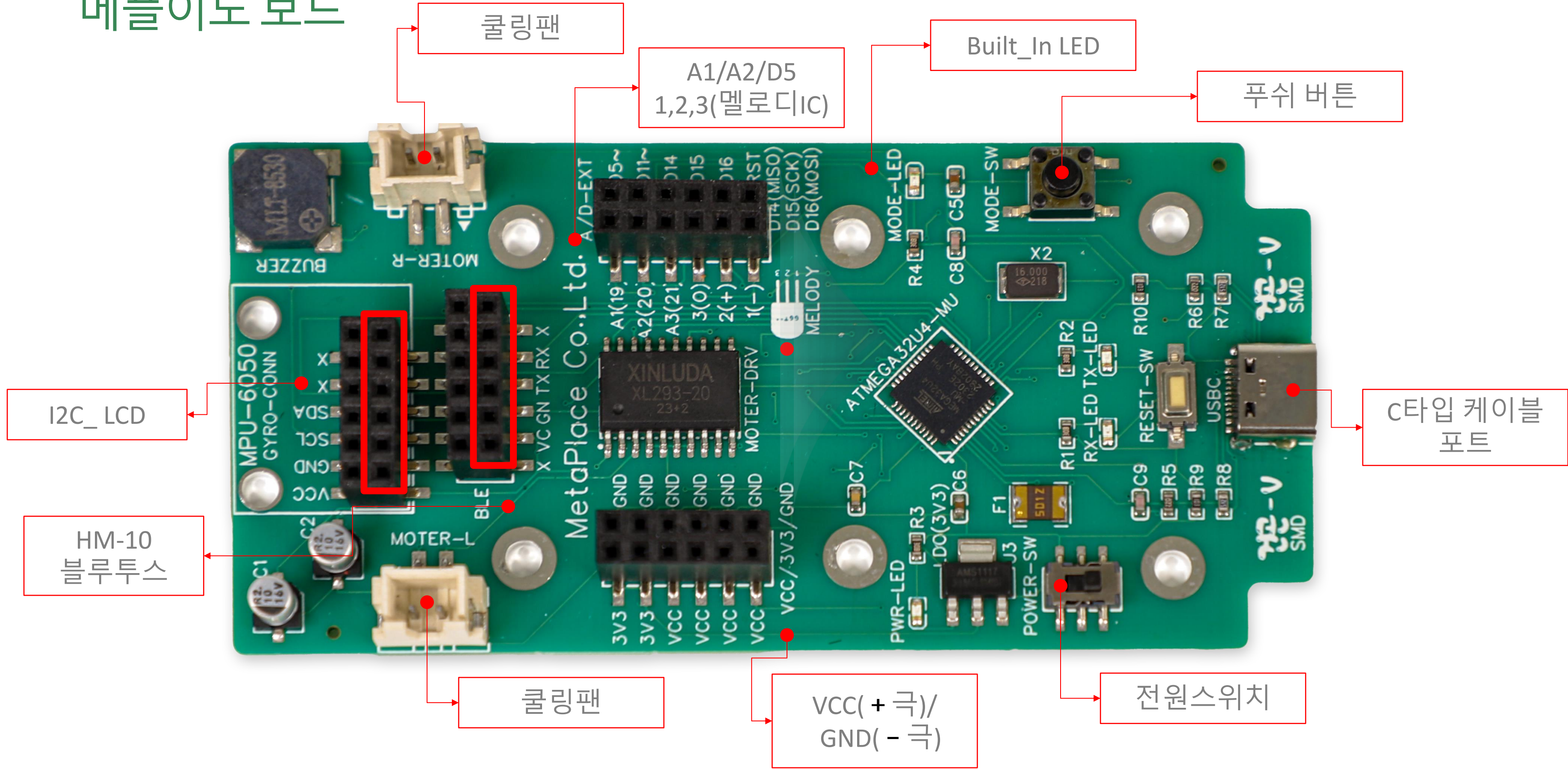
십자드라이버



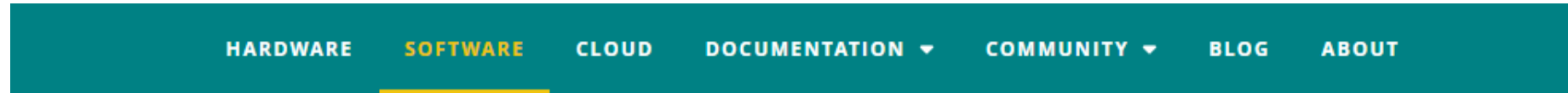
c타입 케이블



# 메플이노 보드



https://www.arduino.cc

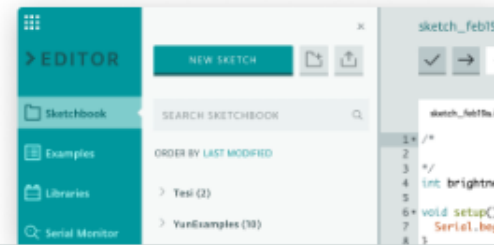


### Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

CODE ONLINE

GETTING STARTED



## Downloads



### Arduino IDE 2.2.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

#### SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

#### DOWNLOAD OPTIONS

**Windows** Win 10 and newer, 64 bits

**Windows** MSI installer

**Windows** ZIP file

**Linux** AppImage 64 bits (X86-64)

**Linux** ZIP file 64 bits (X86-64)

**macOS** Intel, 10.14: "Mojave" or newer, 64 bits

**macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

## Download Arduino IDE & support its progress

Since the 1.x release in March 2015, the Arduino IDE has been downloaded **77,958,276** times — impressive! Help its development with a donation.

\$3

\$5

\$10

\$25

\$50

Other

CONTRIBUTE AND DOWNLOAD

JUST DOWNLOAD

## Stay in the Loop: Join Our Newsletter!

As a beginner or advanced user, you can find inspiring projects and learn about cutting-edge Arduino products through our **weekly newsletter!**

email \*

I confirm to have read the [Privacy Policy](#) and to accept the [Terms of Service](#) \*

I would like to receive emails about special deals and commercial offers from Arduino.

SUBSCRIBE & DOWNLOAD

JUST DOWNLOAD



sketch\_dec31a | Arduino IDE 2.2.1

File Edit Shell Tools Help

Arduino Micro

```
sketch_dec31a.ino
1 void setup() {
2   // put your setup code here, to run once:
3 }
4
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8 }
9
10
```

Serial Monitor x

Message (Enter to send message to 'Arduino Micro' on 'COM9') New Line 9600 baud

Ln 10, Col 1 Arduino Micro on COM9

Tools Help

- Auto Format Ctrl+T
- Archive Sketch
- Manage Libraries... Ctrl+Shift+I
- Serial Monitor Ctrl+Shift+M
- Serial Plotter

---

Firmware Updater

Upload SSL Root Certificates

Board: "Arduino Micro" ▶

Port: "COM9" ▶

Get Board Info

---

Programmer ▶

Burn Bootloader

Arduino Leonardo ETH

- ✓ Arduino Micro
- Arduino Esplora

Serial ports

- ✓ COM9 (Arduino Micro)



sketch\_dec31a | Arduino IDE 2.2.1

File Edit Sketch Tools Help

Arduino Micro

LIBRARY MANAGER

dht

Type: All

Topic: All

Humidity Sensors Arduino...  
[More info](#)

3.4.4 **INSTALL**

**DHT sensor library** by Adafruit  
1.4.6 installed  
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for...  
[More info](#)  
1.4.6 **REMOVE**

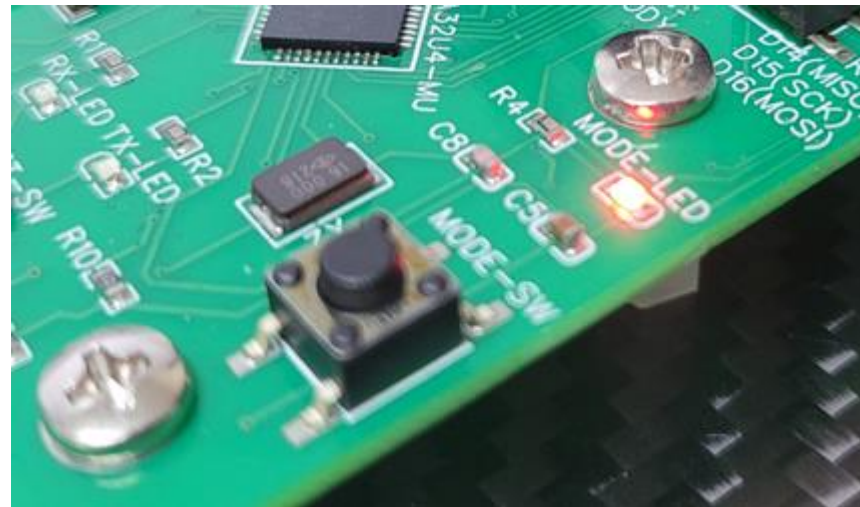
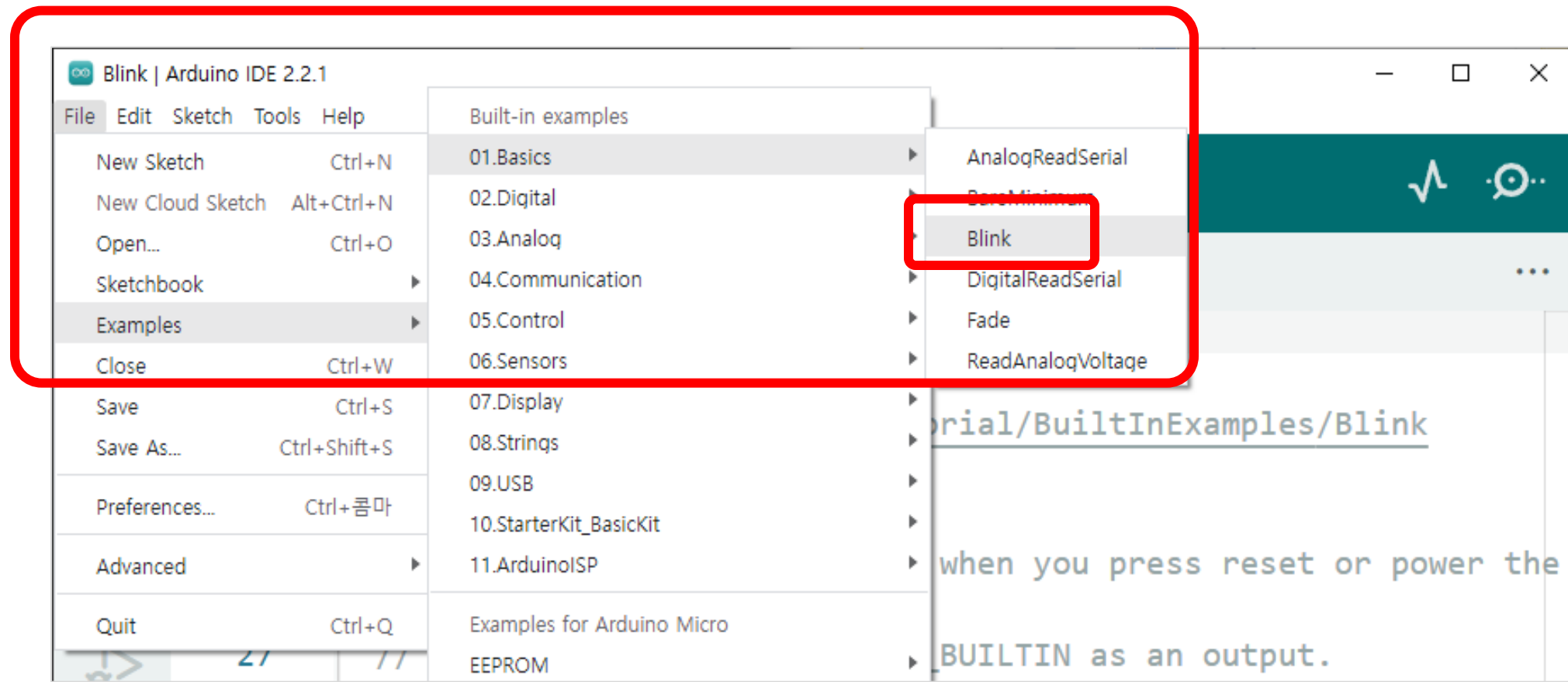
**DHT sensor library for ESPx** by beegee\_tokyo  
1.19 installed  
Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized library to...  
[More info](#)

```
1 void setup() {
2   // put your setup code here, to run once:
3 }
4
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8 }
9
10
```

Serial Monitor x

Message (Enter to send message to 'Arduino Micro' on 'COM9') New Line 9600 baud

Ln 10, Col 1 Arduino Micro on COM9





## LEVEL 1

- 01\_Led
- 02\_Button\_Led
- 03\_Buzzer
- 04\_LCD\_display
- 05\_DHT
- 06\_00\_Neopixel\_LED
- 06\_01\_Neopixel\_LED
- 06\_02\_Neopixel\_LED
- 07\_00\_Fan
- 07\_01\_Fan\_button
- 08\_Soil\_Moisture
- 09\_BT\_Naming
- 10\_BT\_LED

## LEVEL 2

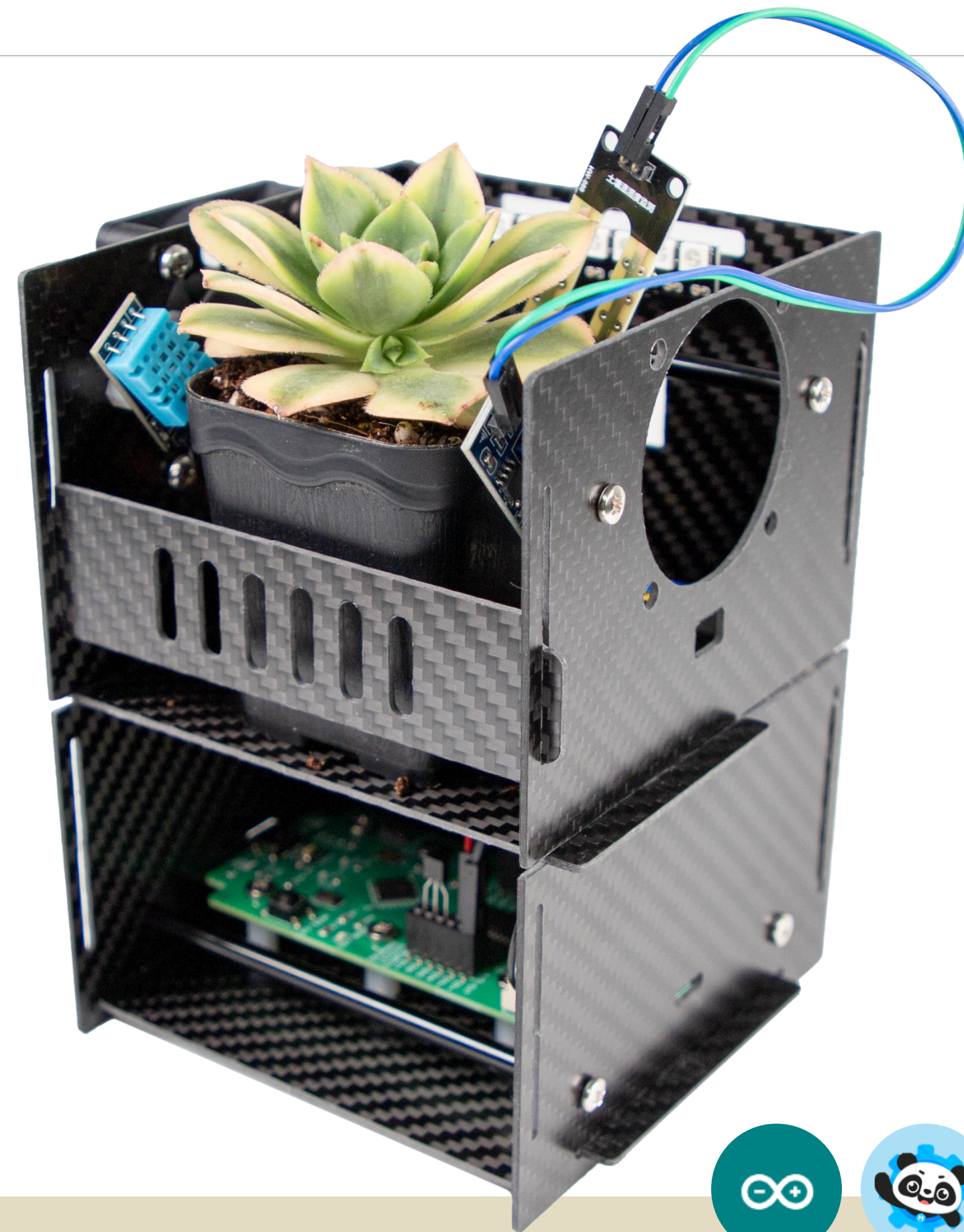
- 01\_project\_LCD\_DHT\_Soil\_Display
- 02\_project\_LCD\_DHT\_Fan
- 03\_project\_LCD\_DHT\_NeoPixel
- 04\_00\_project\_LCD\_Soil\_Melody
- 04\_01\_project\_LCD\_Soil\_Melody\_stopbutton
- 05\_00\_Project\_BT\_LED\_Fan
- 05\_01\_Project\_BT\_Soil\_DHT\_NeoLED
- 01\_정보표시(LCD, 온습도, 토양수분)
- 02\_습도제어(LCD, 습도, 팬)
- 03\_온도제어(LCD, 온도, 네오픽셀)
- 04\_00\_토양습도제어(LCD, 토양습도센서, 멜로디IC)
- 04\_01\_토양습도제어(LCD, 토양습도센서, 멜로디IC)\_버튼으로멈추기
- 05\_00\_블루투스로 제어하기\_LED\_FAN 동작시키기
- 05\_01\_블루투스로 제어하기\_토양습도, 온습도, 네오픽셀 제어하기



MEPL\_FARM



# LEVEL 1





- 메플팜의 모든 센서들의 핀번호 배열 참조
- 모든 파일 상단에 표시되어 있음

06\_Teacher\_BT.ino

```
1
2 Product name : MepL_Farm PCB_ver
3 Board : Arduino_Micro
4 MCU : BuiltIn(ATMEGA32U4-MU)
5
6
7 //---- 센서의 핀 설계
8 Neopixel_Led_Pin number is 5
9 Buzzer_pin       number is 6 (BuiltIn)
10 Led_pin          number is 13(BuiltIn)
11 Button_pin      number is A0(BuiltIn)
12 Soil_Sensor     number is A1
13 DHTPIN          number is A2
14 Fan A_Front     number is 7 // 쿨링팬 왼쪽 A의 전진방향
15 Fan A_Back      number is 4 // 쿨링팬 왼쪽 A의 후진방향
16 Fan A_Front     number is 8 // 쿨링팬 오른쪽 B의 전진방향
17 Fan A_Back      number is 12 // 쿨링팬 오른쪽 B의 후진방향
18
```

- BuiltIn 13번 핀 LED를 1초마다 ON/OFF
- Serial.println 시리얼 모니터로 메시지 확인해보기

```
21 // 디지털핀 설계
22 // 13번 디지털핀(led_pin)
23 #define Led_pin 13
24
25 //-----
26 // 부팅시 만 실행
27 void setup()
28 {
29     // 시리얼통신 초기화(9600)bps(bits per second)
30     Serial.begin(9600);
31
32     // 디지털핀 초기화(OUTPUT)
33     pinMode(Led_pin, OUTPUT);
34 }
35 // 부팅시 만 실행
```

```
37 //-----
38 // 메인(반복루프)
39 void loop()
40 {
41     // LED ON
42     Serial.println("LED ON");
43     digitalWrite(Led_pin, HIGH);
44     delay(1000);
45     Serial.println("LED OFF");
46     digitalWrite(Led_pin, LOW);
47     delay(1000);
48 }
49 // 메인 끝
50
```



```
21 // 디지털핀 설계
22 // 13번 디지털핀(led_pin)
23 #define Led_pin 13
24 // 아날로그핀 설계
25 #define Button_pin A0
26 // 버튼상태 변수 선언
27 int Button_state = 0;
28
29 //-----
30 // 부팅시 만 실행
31 void setup() {
32     // 시리얼통신 초기화(9600)bps(bits per second)
33     Serial.begin(9600);
34     // 디지털핀 초기화(OUTPUT)
35     pinMode(Led_pin, OUTPUT);
36 }
37 // 부팅시 만 실행
38
```

```
39 //-----
40 // 메인(반복루프)
41 void loop() {
42     // 모드스위치(푸쉬스위치) 눌림 512이상 입력값 생성
43     // 아날로그 입력값[0~1023]
44     if (analogRead(Button_pin) > 512) {
45         // 모드스위치(푸쉬스위치) 누르고 기다리는 시간
46         delay(500);
47
48         if (Button_state == 0) {
49             Button_state = 1;
50             // LED ON
51             digitalWrite(Led_pin, HIGH);
52             Serial.println("LED ON");
53         }
54         else if (Button_state == 1) {
55             Button_state = 0;
56             // LED OFF
57             digitalWrite(Led_pin, LOW);
58             Serial.println("LED OFF");
59         }
60     }
61 }
62 // 메인 끝
```

```

21 // 디지털핀 설계
22 #define Led_pin 13
23 #define Buzzer_pin 6
24 // 아날로그핀 설계
25 #define button_pin A0
26 // 버튼상태 변수 선언
27 int Button_state = 0;
28
29
30 //-----
31 // 부팅시 만 실행
32 void setup() {
33     // 시리얼통신 초기화(9600)bps(bits per second)
34     Serial.begin(9600);
35     // 디지털핀 초기화(OUTPUT)
36     pinMode(Led_pin, OUTPUT);
37     pinMode(Buzzer_pin, OUTPUT);
38 }
39 // 부팅시 만 실행
40

```

```

41
42 //-----
43 // 메인(반복루프)
44 void loop()
45 {
46     // 모드스위치(푸쉬스위치) 눌림 512이상 입력값 생성
47     if (analogRead(button_pin) > 512)
48     {
49         // 모드스위치(푸쉬스위치) 누르고 기다리는 시간
50         delay(500);
51
52         if (Button_state == 0) {
53             Button_state = 1;
54             // BUZZER ON
55             digitalWrite(Buzzer_pin, HIGH);
56             Serial.println("BUZZER ON");
57         }
58         else if (Button_state == 1) {
59             Button_state = 0;
60             // BUZZER OFF
61             digitalWrite(Buzzer_pin, LOW);
62             Serial.println("BUZZER OFF");
63         }
64     }
65 }
66 // 메인 끝

```

- I2C LCD 디스플레이스 (16문자 \* 2줄 출력가능)
- 라이브러리 설치후 프로그래밍

The screenshot shows the Arduino IDE 2.2.1 interface. The top bar indicates the board is set to "Arduino Micro". The "LIBRARY MANAGER" panel on the left shows the search results for "liquidcrystal". The "LiquidCrystal I2C" library by Frank de Brabander is highlighted, with version 1.1.2 installed. The sketch editor on the right shows the code for "04\_LCD\_display.ino", which includes the "LiquidCrystal\_I2C" library and initializes an I2C LCD display.

```
2 Mep1_Farm
3 BuiltIn(ATMEGA32U4-MU)
4 PCB_ver
5
6 -- 핀 설계
7 pixelsPin  number is 5 (NeoPixels Pin)
8 buzzer_pin  number is 6
9 Led_pin    number is 13(built in)
10 button_pin number is A0(built in)
11 Soil_Sensor number is A1
12 DHTPIN     number is A2
13
14 */
15
17 #include <LiquidCrystal_I2C.h>
18 LiquidCrystal_I2C lcd(0x27, 16, 2);
```



```
21 #include <LiquidCrystal_I2C.h>
22 // set the LCD address to 0x27(0x3F)
23 // for a 16 chars and 2 line display
24 LiquidCrystal_I2C lcd(0x27, 16, 2);
25
26 //-----
27 // 부팅시 만 실행
28 void setup() {
29     // LCD 초기화
30     lcd.init();
31     // 백라이트 활성화
32     lcd.backlight();
33     // LCD 화면 지우기
34     lcd.clear();
35 }
```

```
36
37 void loop() {
38     //0행 0열 커서위치
39     lcd.setCursor(0,0);
40     lcd.print("Hello,MetaPlace!");
41     //1행 0열 커서위치
42     lcd.setCursor(0,1);
43     lcd.print("Happy Day!");
44 }
45 }
```

```

21 #include "DHT.h"
22 // DHT 센서 핀 정의
23 #define DHTPIN A2
24 // DHT 센서의 종류
25 #define DHTTYPE DHT11
26 // 정의된 핀과 센서 유형으로 dht 객체 생성
27 DHT dht(DHTPIN, DHTTYPE);
28
29 //-----
30 // 부팅시 만 실행
31 void setup() {
32     // 시리얼통신 초기화(9600)bps(bits per second)
33     Serial.begin(9600);
34     // 온습도센서를 초기화 합니다.
35     dht.begin();
36 }
37

```

```

38 //-----
39 // 메인 (반복루프)
40 void loop()
41 {
42     delay(2000);
43
44     // 습도와 온도 값을 측정하여 읽어옵니다.(h는 습도, t는 온도)
45     int h = dht.readHumidity();
46     int t = dht.readTemperature();
47
48     // Serial monitor (0, 0)에 습도값을 출력합니다.( Humi : XX.XX% )
49     Serial.println("Outputs temperature and humidity value!");
50     Serial.print("Humi : ");
51     Serial.print(h);
52     Serial.println("%");
53     delay(100);
54
55     Serial.print("Temp : ");
56     Serial.print(t);
57     Serial.println("°C");
58     delay(200);
59     Serial.println();
60 }

```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Micro' on 'COM9') New Line 9600 baud

Temp : 22°C

Outputs temperature and humidity value!

Humi : 36%

Temp : 22°C

The screenshot displays the Arduino IDE 2.2.1 interface. On the left, the Library Manager shows the 'Adafruit NeoPixel' library by Adafruit, version 1.12.0, which is installed. The main editor window shows the code for '06\_00\_Neopixel\_LED.ino'. The code defines pins for a button, soil sensor, DHT sensor, and fans. It also includes the Adafruit NeoPixel library and defines the number of pixels (8) and the pin (5) used for the LED strip. The initialization of the NeoPixel object is highlighted with a red box.

```
11 button_pin      number is  A0(DU11C11)
12 Soil_Sensor     number is  A1
13 DHTPIN          number is  A2
14 Fan A_Front     number is  7      // 쿨링팬 왼쪽  A의 전진방향
15 Fan A_Back      number is  4      // 쿨링팬 왼쪽  A의 후진방향
16 Fan A_Front     number is  8      // 쿨링팬 오른쪽 B의 전진방향
17 Fan A_Back      number is  12     // 쿨링팬 오른쪽 B의 후진방향
18
19 *//
20
21 #include <Adafruit_NeoPixel.h>
22
23 #define numPixels 8 //스트립 개수
24 #define pixelsPin 5 //PWM 핀번호 사용
25
26 Adafruit_NeoPixel pixels(numPixels, pixelsPin, NEO_GRB + NEO_KHZ800);
27 // (NEO_GRB + NEO_KHZ800): 색상 형식과 통신 속도를 지정
28
```



```

21 //네오픽셀 라이브러리
22 #include <Adafruit_NeoPixel.h>
23 //스트립 개수
24 #define numPixels 8
25 //PWM 핀번호 사용(D5~)
26 #define pixelsPin 5
27 // (NEO_GRB + NEO_KHZ800): 색상 형식과 통신 속도를 지정
28 Adafruit_NeoPixel pixels(numPixels, pixelsPin, NEO_GRB + NEO_KHZ800);
29
30 //-----
31 // 부팅시 만 실행
32 void setup() {
33     Serial.begin(9600);
34     pixels.begin();
35     //밝기는 0에서 255까지의 값
36     pixels.setBrightness(10);
37 }

```

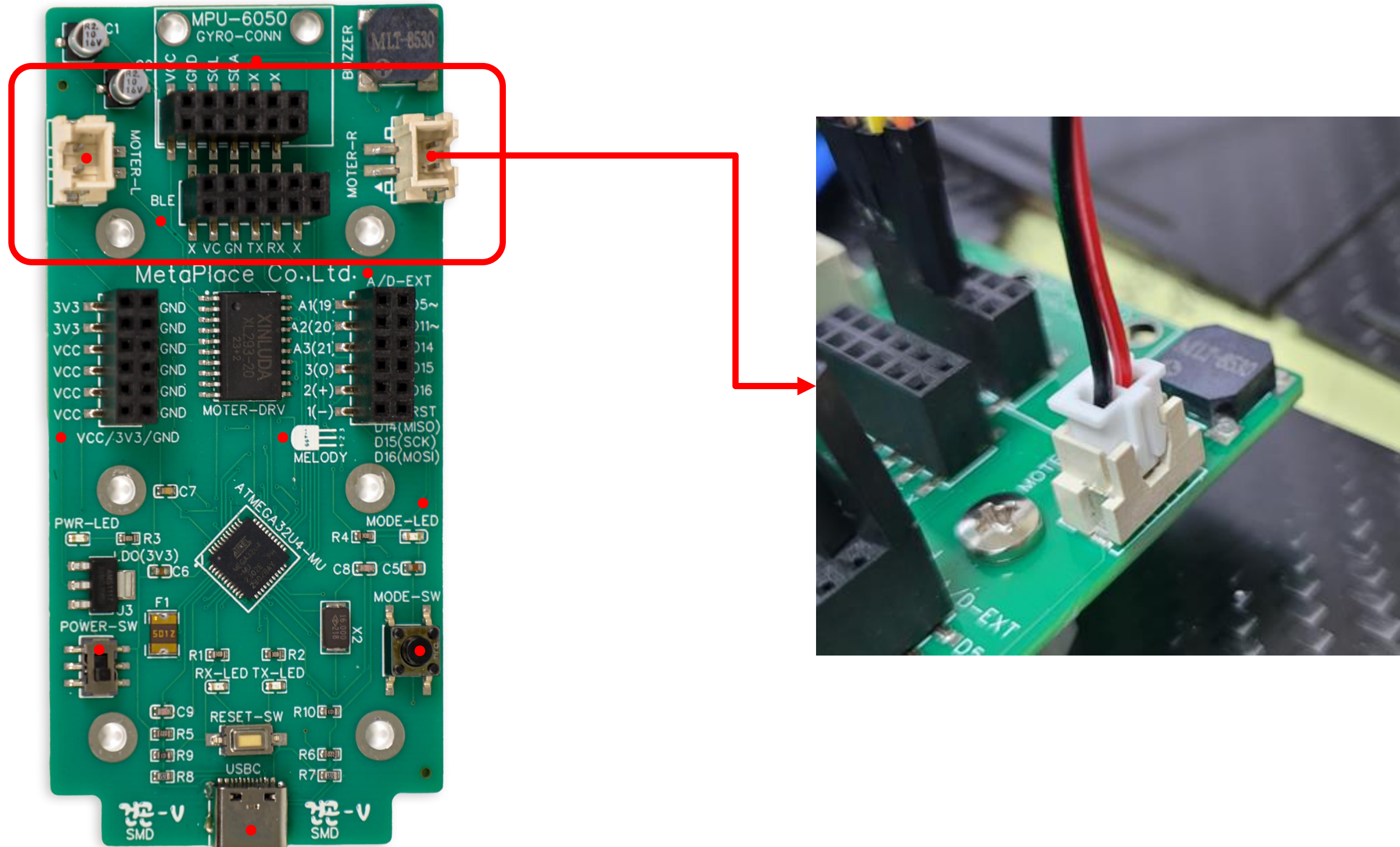
```

38
39 void loop() {
40     pixels.clear();
41     pixels.show();
42     delay(100);
43
44     // 첫 번째 픽셀을 빨간색으로 설정
45     pixels.setPixelColor(0,255,0,0);
46     pixels.show();
47     delay(200);
48     // 두 번째 픽셀을 녹색으로 설정
49     pixels.setPixelColor(1,0,255,0);
50     pixels.show();
51     delay(200);
52     // 세 번째 픽셀을 파란색으로 설정
53     pixels.setPixelColor(2,0,0,255);
54     pixels.show();
55     delay(200);
56 }

```

```
21 //네오픽셀 라이브러리
22 #include <Adafruit_NeoPixel.h>
23 //스트립 개수
24 #define numPixels 8
25 //PWM 핀번호 사용(D5~)
26 #define pixelsPin 5
27 // (NEO_GRB + NEO_KHZ800): 색상 형식과 통신 속도를 지정
28 Adafruit NeoPixel pixels(numPixels, pixelsPin, NEO_GRB + NEO_K
29 uint32_t yellowPixel = pixels.Color(255, 255, 0);
30 uint32_t magentaPixel = pixels.Color(255, 0, 255);
31 uint32_t cyanPixel = pixels.Color(0, 255, 255);
32
59 //청록
60 pixels.setPixelColor(3,0,255,255);
61 pixels.show();
62 delay(200);
63
64 pixels.setPixelColor(4,yellowPixel);
65 pixels.show();
66 delay(200);
67
68 pixels.setPixelColor(5,magentaPixel);
69 pixels.show();
70 delay(200);
71
72 pixels.setPixelColor(6,cyanPixel);
73 pixels.show();
74 delay(200);
75
76 pixels.setPixelColor(7,255,255,255);
77 pixels.show();
78 delay(200);
79 }
```

- 쿨링팬은 메플이노 보드의 왼쪽이 A / 오른쪽이 B 로 기준한다.





- 쿨링팬은 모터드라이브를 통해 작동한다.
- 쿨링팬이 5초 동안 작동하고 멈춘다.

```

21 #define A_Front 7 // 쿨링팬 왼쪽 A의 전진방향
22 #define A_Back 4 // 쿨링팬 왼쪽 A의 후진방향
23 #define B_Front 8 // 쿨링팬 오른쪽 B의 전진방향
24 #define B_Back 12 // 쿨링팬 오른쪽 B의 후진방향
25
26 // 모터가 활성화되었는지 여부를 나타내는 변수
27 bool motorsActivated = false;
28
29 //-----
30 // 부팅시 만 실행
31 void setup() {
32     pinMode(A_Front, OUTPUT);
33     pinMode(A_Back, OUTPUT);
34     pinMode(B_Front, OUTPUT);
35     pinMode(B_Back, OUTPUT);
36
37     motorsActivated = true;
38 }

```

```

39
40 void loop() {
41     if (motorsActivated) {
42         // 모터 A를 전진 방향으로 회전
43         digitalWrite(A_Front, LOW);
44         digitalWrite(A_Back, HIGH);
45
46         // 모터 B를 전진 방향으로 회전
47         digitalWrite(B_Front, HIGH);
48         digitalWrite(B_Back, LOW);
49
50         delay(5000); // 시간 지정 (예: 5초)
51
52         motorsActivated = false;
53
54     } else {
55         // 모터 A와 모터 B를 정지
56         digitalWrite(A_Front, LOW);
57         digitalWrite(A_Back, LOW);
58         digitalWrite(B_Front, LOW);
59         digitalWrite(B_Back, LOW);
60     }
61 }

```

- 버튼을 누르면 쿨링팬이 작동하고, 다시 누르면 꺼진다.
- 버튼을 누르고 팬이 돌아가는 delay 시간을 충분히 주면서 눌러본다.

```

21 // 아날로그핀 설계
22 #define Button_pin A0
23 // 버튼상태 변수 선언
24 int Button_state = 0;
25
26 #define A_Front 7 // 쿨링팬 왼쪽 A의 전진방향
27 #define A_Back 4 // 쿨링팬 왼쪽 A의 후진방향
28 #define B_Front 8 // 쿨링팬 오른쪽 B의 전진방향
29 #define B_Back 12 // 쿨링팬 오른쪽 B의 후진방향
30
31 // 모터가 활성화되었는지 여부를 나타내는 변수
32 bool motorsActivated = false;
33
34 //-----
35 // 부팅시 만 실행
36 void setup() {
37     pinMode(A_Front, OUTPUT);
38     pinMode(A_Back, OUTPUT);
39     pinMode(B_Front, OUTPUT);
40     pinMode(B_Back, OUTPUT);
41
42     // 버튼 핀을 입력으로 설정
43     pinMode(Button_pin, INPUT);
44     // 초기에는 모터 비활성화 상태로 설정
45     motorsActivated = false;
46 }
47

```

```

48 //-----
49 // 메인(반복루프)
50 void loop()
51 {
52     // 버튼이 눌렸을 때
53     if (analogRead(Button_pin) > 512) {
54         delay(500);
55         motorsActivated = true; // 모터 활성화
56
57         if (Button_state == 0) {
58             Button_state = 1;
59             // 모터 A를 전진 방향으로 회전
60             digitalWrite(A_Front, LOW);
61             digitalWrite(A_Back, HIGH);
62
63             // 모터 B를 전진 방향으로 회전
64             digitalWrite(B_Front, HIGH);
65             digitalWrite(B_Back, LOW);
66         }
67         else if (Button_state == 1) {
68             Button_state = 0;
69             // 모터 A와 모터 B를 정지
70             digitalWrite(A_Front, LOW);
71             digitalWrite(A_Back, LOW);
72             digitalWrite(B_Front, LOW);
73             digitalWrite(B_Back, LOW);
74         }
75     }
76 }

```

```
21 // 토양 온습도 센서 핀 번호
22 #define Soil_Sensor A1
23
24 void setup() {
25     Serial.begin(9600);
26 }
27
28 void loop() {
29     // 토양 온습도 센서의 값을 읽어옴
30     int soilSensorValue = analogRead(Soil_Sensor);
31
32     // 시리얼 모니터에 출력
33     Serial.print("토양 온습도 센서 값: ");
34     Serial.println(soilSensorValue);
35
36     delay(1000); // 1초마다 측정
37 }
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Micro' on 'COM9')

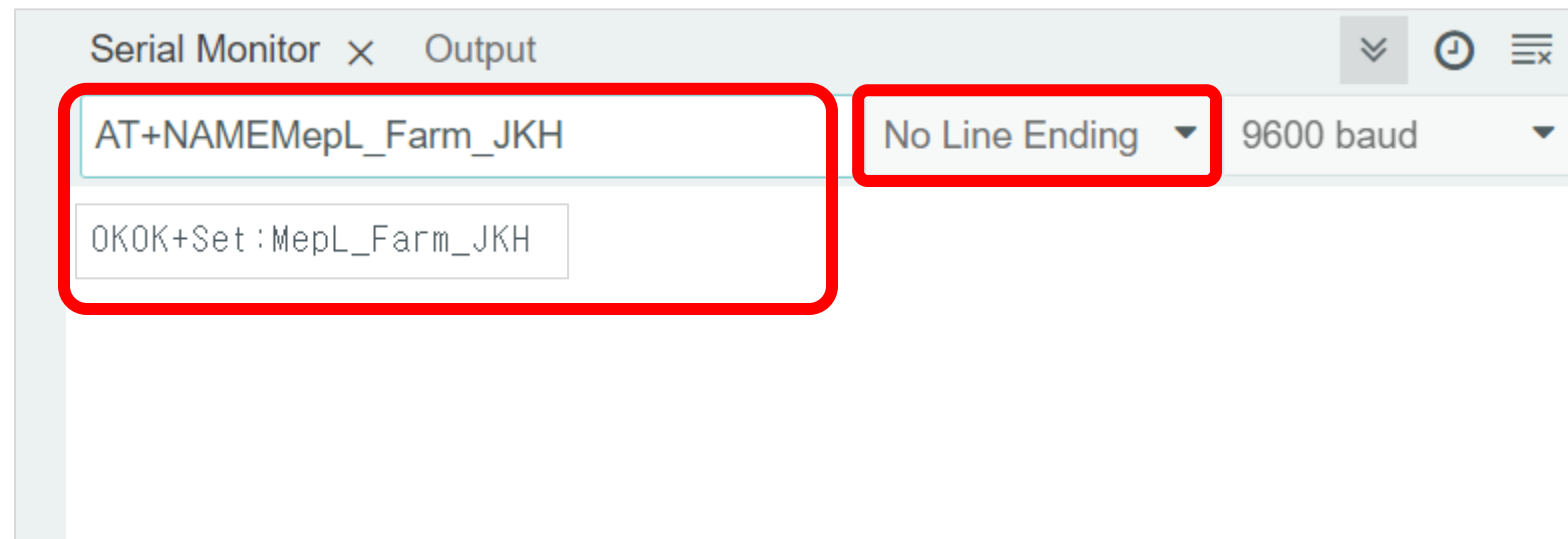
```
토양 온습도 센서 값: 1023
토양 온습도 센서 값: 1022
토양 온습도 센서 값: 1023
토양 온습도 센서 값: 481
토양 온습도 센서 값: 591
토양 온습도 센서 값: 581
```



```

21  /* 참고사항 : 아두이노우노 호환보드 사용시에는
22  // 시리얼통신 라이브러리 호출
23  #include <SoftwareSerial.h>
24  SoftwareSerial bluetooth_Serial(2,12);
25  // Meplino board에서는 사용하지 않음
26  */
27
28  void setup() {
29      Serial.begin(9600);
30
31      //블루투스 시리얼 개방
32      Serial1.begin(9600);
33  }
34

```



```

35  void loop() {
36  /* Serial Monitor에서 setting
37  "line ending 없음"
38  AT (OK)
39  AT+NAME이름 (OK+Set:이름)
40  */
41
42  // bluetooth_Serial 값이 있으면
43  if (Serial1.available())
44  {
45      //블루투스측 내용을 시리얼모니터에 출력
46      Serial.write(Serial1.read());
47  }
48
49  if (Serial.available())
50  {
51      // 시리얼모니터 내용을 블루투스측에 출력
52      Serial1.write(Serial.read());
53  }
54  }

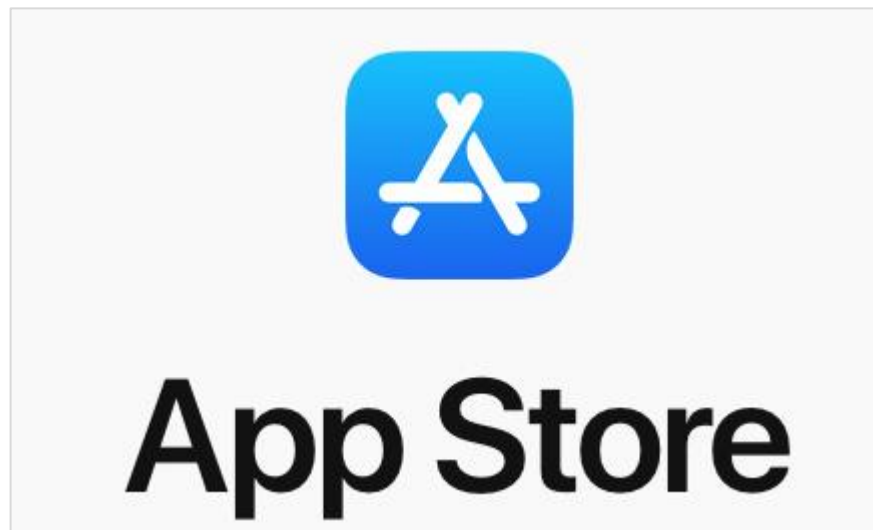
```

Serial1

① - 스마트스토어 검색




안드로이드  
스마트스토어



🔍 **Ble mcu Controller** ✕

② 위의 이름으로 검색하기

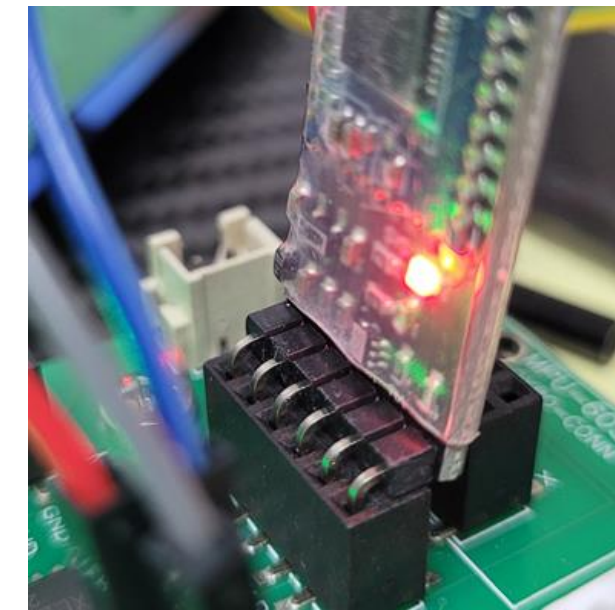
Apps / Productivity / BLE MCU Controller



**BLE MCU Controller**  
BLE Meplino Controller  
Free  
1.0 for iPhone, iPad and more

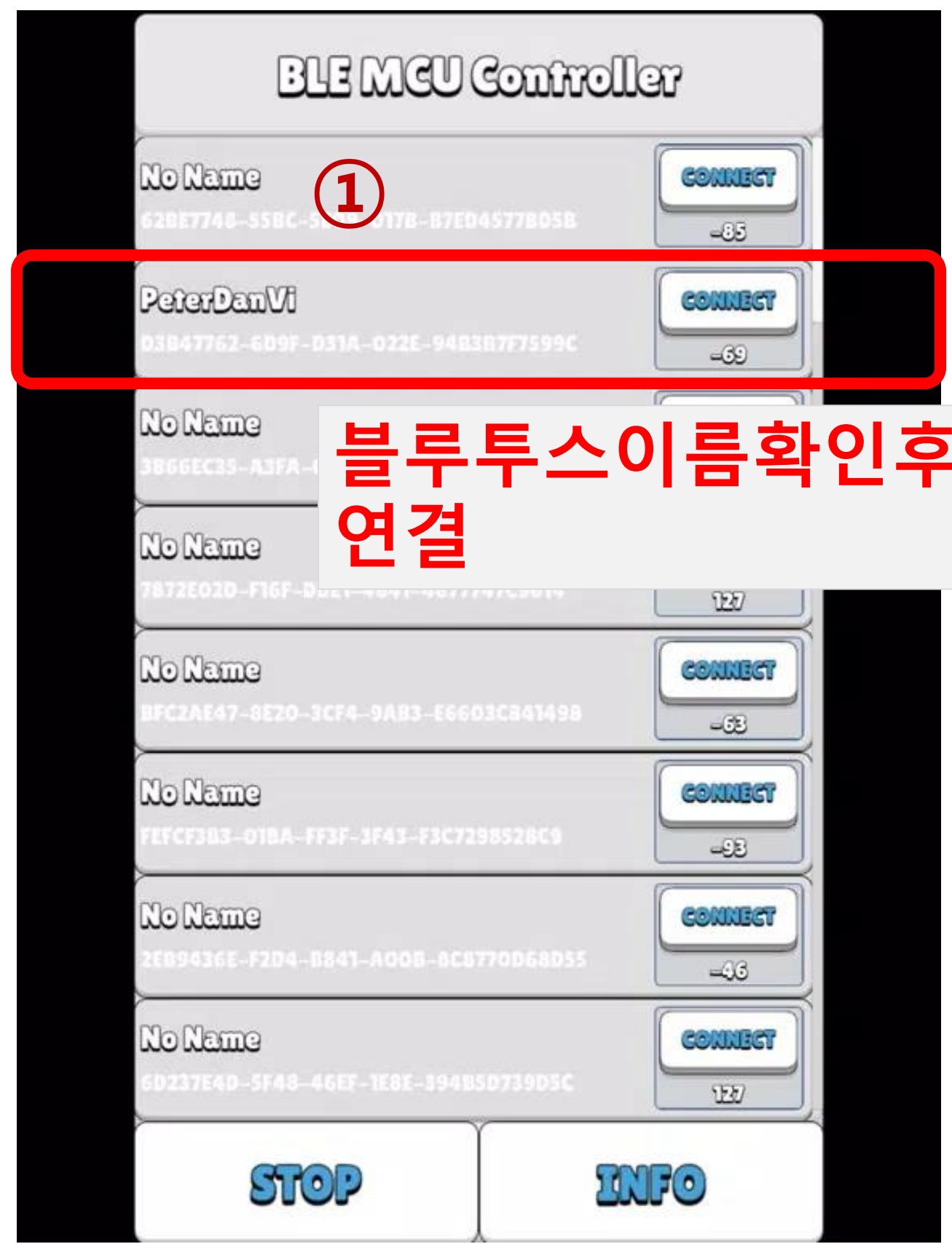
OHSANG KWON Developer      46.5 MB Size      Aug 12, 2024 Update Date      Productivity Category      4+ Age Rating

🍏 for iPhone      📄  
[How to install .IPA file](#)

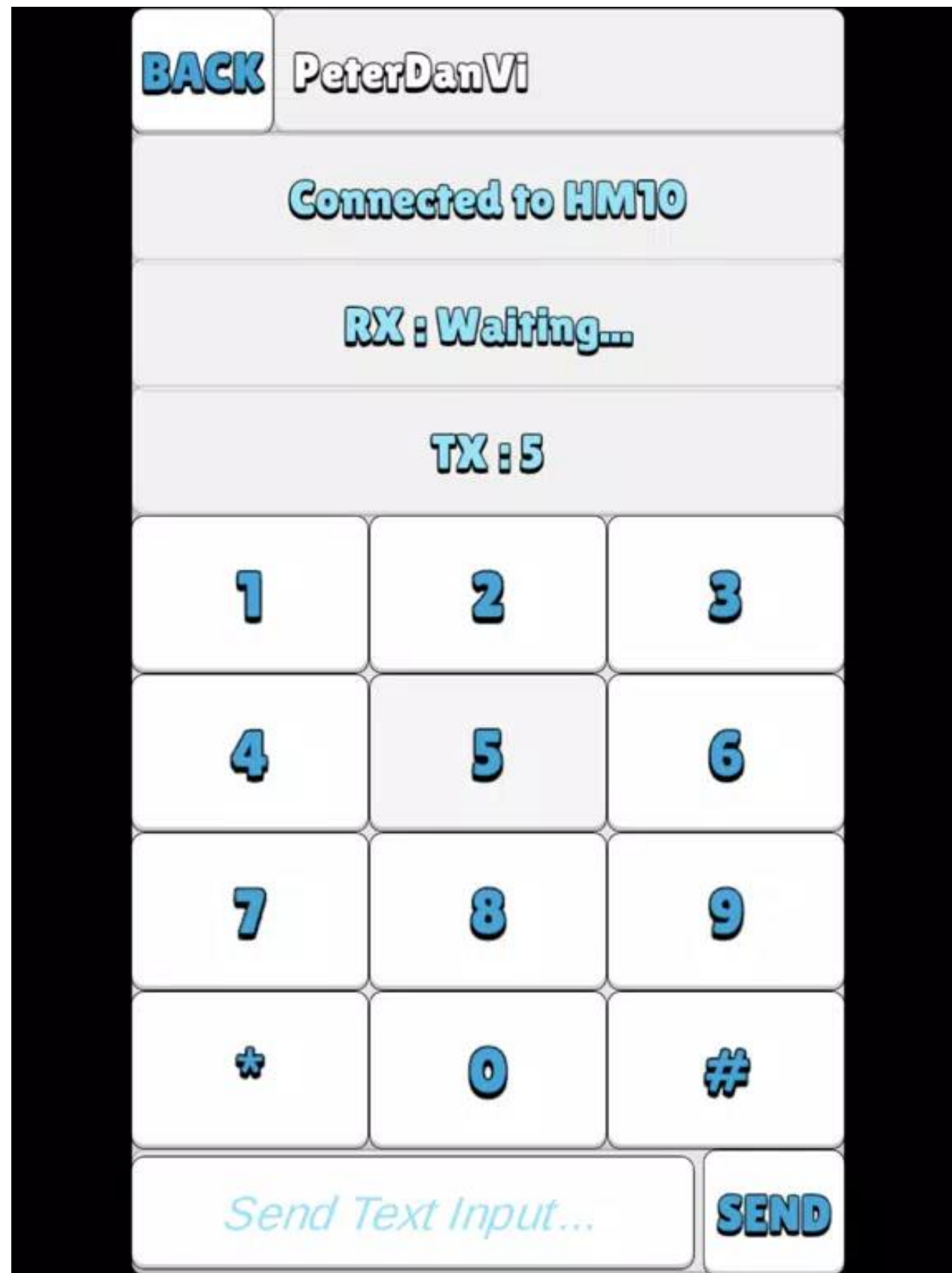


- 블루투스 연결되면,  
LED가 계속 켜져있다.

- 블루투스를 켜고, 이름으로 검색하여 컨트롤러 모드로 접속한다.







1. PCB보드의 LED를 ON/OFF
2. Fan ON/OFF
3. 온습도센서값 / 토양습도센서값 보기
4. 네오픽셀 라이트 ON/OFF

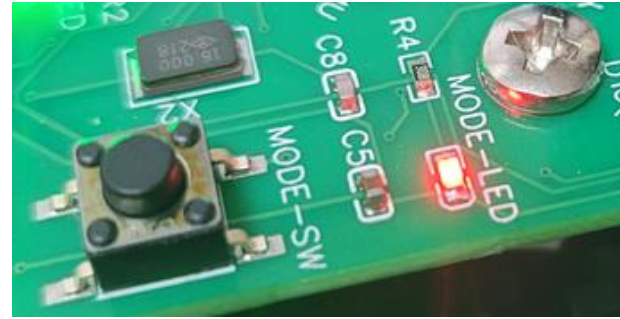
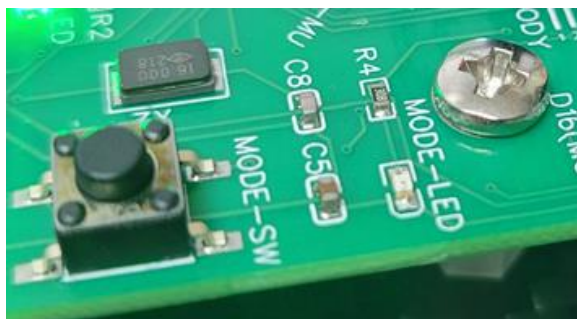
5. Level2\_06\_Teacher\_BT 폴더자료  
업로드하면 됩니다.

- 자신이 만든 IoT 스마트팜 코드를 생성해보아요!

```

21  /* 참고사항 : 아두이노우노 호환보드 사용시에는
22  // 시리얼통신 라이브러리 호출
23  #include <SoftwareSerial.h>
24  SoftwareSerial bluetooth_Serial(2,12);
25  // Meplino board에서는 사용하지 않음
26  */
27
28  // 블루투스에서 받을 빈 문자열 선언
29  String bluetooth_String = "";
30  #define Led_pin 13
31
32  void setup() {
33    Serial.begin(9600);
34    //블루투스 시리얼 개방
35    Serial1.begin(9600);
36    pinMode(Led_pin, OUTPUT);
37  }
38

```



```

39  void loop() {
40    // Bluetooth_Rx_Tx:
41    // bluetooth_Serial ASCII형식
42    if (Serial1.available()) {
43      bluetooth_String = Serial1.read();
44    }
45
46    // bluetooth_String 값이 있다면
47    if (bluetooth_String != "") {
48      // ASCII 1 = "49"
49      if (bluetooth_String == "49") {
50        Serial.println("LED ON!");
51        // LED ON
52        digitalWrite(Led_pin, HIGH);
53      }
54
55      // 2 = "50"
56      if (bluetooth_String == "50") {
57        Serial.println("LED OFF!");
58        // LED OFF
59        digitalWrite(Led_pin, LOW);
60      }
61    }
62  }

```

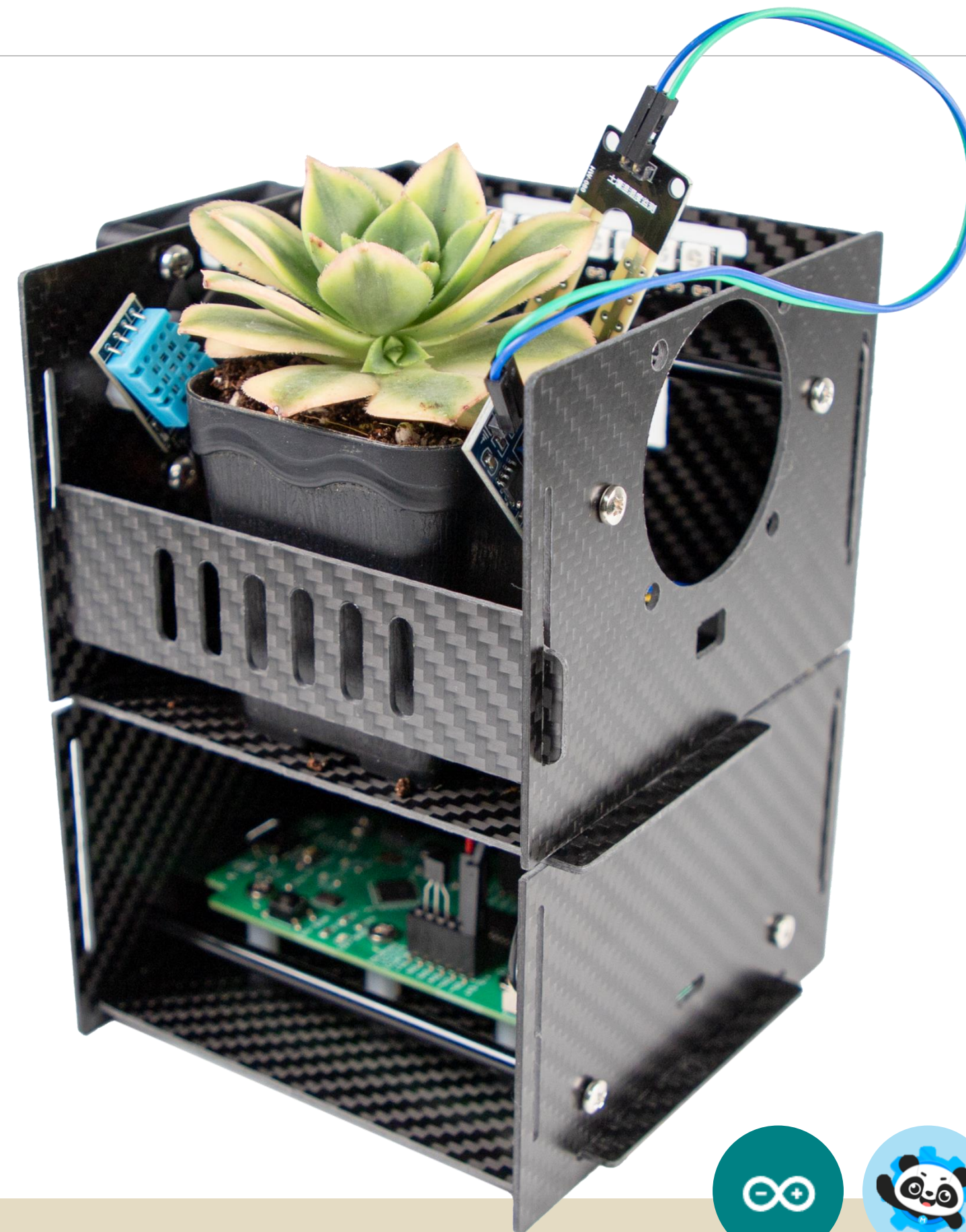




MEPL\_FARM



# LEVEL 2





```

21 #include <LiquidCrystal_I2C.h>
22 #include "DHT.h"
23
24 // DHT 센서 핀 정의
25 #define DHTPIN A2
26 // DHT 센서의 종류
27 #define DHTTYPE DHT11
28 // 토양 온습도 센서 핀 번호
29 #define Soil_Sensor A1
30
31 // set the LCD address to 0x27(0x3F)
32 // for a 16 chars and 2 line display
33 LiquidCrystal_I2C lcd(0x27, 16, 2);
34 // 정의된 핀과 센서 유형으로 dht 객체 생성
35 DHT dht(DHTPIN, DHTTYPE);

```

```

36
37 //-----
38 // 부팅시 만 실행
39 void setup() {
40     // 시리얼통신 초기화(9600)bps
41     Serial.begin(9600);
42     // LCD 초기화
43     lcd.init();
44     // 백라이트 활성화
45     lcd.backlight();
46     // LCD 화면 지우기
47     lcd.clear();
48     // 온습도센서를 초기화 합니다.
49     dht.begin();
50 }
51

```

```

52 void loop() {
53     delay(2000);
54     // 습도와 온도 값을 측정하여 읽어옵니다.(h는 습도, t는 온도)
55     int h = dht.readHumidity();
56     int t = dht.readTemperature();
57     // 토양 온습도 센서의 값을 읽어옴
58     int soilSensorValue = analogRead(Soil_Sensor);
59     //LCD 화면 지우기
60     lcd.clear();
61     //0행 0열 커서위치
62     lcd.setCursor(0,0);
63     lcd.print("H:");
64     lcd.print(h);
65     lcd.print("%, ");
66     //연속하여 프린트
67     lcd.print("T:");
68     lcd.print(t);
69     lcd.print("C");
70     //1행 0열 커서위치
71     lcd.setCursor(0,1);
72     lcd.print("Soil_value:");
73     lcd.print(soilSensorValue);
74 }

```

```

21  #include <LiquidCrystal_I2C.h>
22  #include "DHT.h"
23
24  // DHT 센서 핀 정의
25  #define DHTPIN A2
26  // DHT 센서의 종류
27  #define DHTTYPE DHT11
28  // 토양 온습도 센서 핀 번호
29  #define Soil_Sensor A1
30  #define A_Front  7      // 쿨링팬 왼쪽  A의 전진방향
31  #define A_Back   4      // 쿨링팬 왼쪽  A의 후진방향
32  #define B_Front  8      // 쿨링팬 오른쪽 B의 전진방향
33  #define B_Back  12     // 쿨링팬 오른쪽 B의 후진방향
34
35  LiquidCrystal_I2C lcd(0x27, 16, 2);
36  // 정의된 핀과 센서 유형으로 dht 객체 생성
37  DHT dht(DHTPIN, DHTTYPE);
38
39  //-----
40  // 부팅시 만 실행
41  void setup() {
42      // 시리얼통신 초기화(9600)bps
43      Serial.begin(9600);
44      // LCD 초기화
45      lcd.init();
46      // 백라이트 활성화
47      lcd.backlight();
48      // LCD 화면 지우기
49      lcd.clear();
50      // 온습도센서를 초기화 합니다.
51      dht.begin();
52
53      pinMode(A_Front, OUTPUT);
54      pinMode(A_Back, OUTPUT);
55      pinMode(B_Front, OUTPUT);
56      pinMode(B_Back, OUTPUT);
57  }
58

```

```

59 void loop()
60 {
61   delay(2000);
62   // h는 습도, t는 온도
63   int h = dht.readHumidity();
64   int t = dht.readTemperature();
65   //LCD 화면 지우기
66   lcd.clear();
67
68   // 습도를 원하는 값으로 변경하여 테스트
69   if(h > 37) {
70     //0행 0열 커서위치
71     lcd.setCursor(0,0);
72     lcd.print("H:");
73     lcd.print(h);
74     lcd.print("%, ");
75     lcd.print("T:");
76     lcd.print(t);
77     lcd.print("C");
78
79     //1행 0열 커서위치
80     lcd.setCursor(0,1);
81     lcd.print("It's High...Fan On!");
82

```

```

83   // 모터 A를 전진 방향으로 회전
84   digitalWrite(A_Front, LOW);
85   digitalWrite(A_Back, HIGH);
86   // 모터 B를 전진 방향으로 회전
87   digitalWrite(B_Front, HIGH);
88   digitalWrite(B_Back, LOW);
89   }
90   else {
91     lcd.clear();
92     lcd.setCursor(0,0);
93     lcd.print("H:");
94     lcd.print(h);
95     lcd.print("%, ");
96     lcd.print("Fan OFF!");
97     lcd.setCursor(0,1);
98     lcd.print("Optimal Humidity!");
99
100    // 모터 A와 모터 B를 정지
101    digitalWrite(A_Front, LOW);
102    digitalWrite(A_Back, LOW);
103    digitalWrite(B_Front, LOW);
104    digitalWrite(B_Back, LOW);
105   }
106 }

```



```

21 #include <Adafruit_NeoPixel.h>
22 #include <LiquidCrystal_I2C.h>
23 #include "DHT.h"
24
25 // DHT 센서 핀 정의
26 #define DHTPIN A2
27 // DHT 센서의 종류
28 #define DHTTYPE DHT11
29 // 토양 온습도 센서 핀 번호
30 #define Soil_Sensor A1
31 #define A_Front 7 // 쿨링팬 왼쪽 A의 전진방향
32 #define A_Back 4 // 쿨링팬 왼쪽 A의 후진방향
33 #define B_Front 8 // 쿨링팬 오른쪽 B의 전진방향
34 #define B_Back 12 // 쿨링팬 오른쪽 B의 후진방향
35 #define numPixels 8 // 스트립 개수
36 #define pixelsPin 5 // PWM 핀번호 사용
37
38 LiquidCrystal_I2C lcd(0x27, 16, 2);
39 // 정의된 핀과 센서 유형으로 dht 객체 생성
40 DHT dht(DHTPIN, DHTTYPE);
41 // (NEO_GRB + NEO_KHZ800): 색상 형식과 통신 속도를 지정
42 Adafruit_NeoPixel pixels(numPixels, pixelsPin, NEO_GRB + NEO_KHZ800);
43

```

```

44 //-----
45 // 부팅시 만 실행
46 void setup() {
47     // 시리얼 통신 초기화(9600)bps
48     Serial.begin(9600);
49     // LCD 초기화
50     lcd.init();
51     // 백라이트 활성화
52     lcd.backlight();
53     // LCD 화면 지우기
54     lcd.clear();
55     // 온습도센서를 초기화 합니다.
56     dht.begin();
57
58     pinMode(A_Front, OUTPUT);
59     pinMode(A_Back, OUTPUT);
60     pinMode(B_Front, OUTPUT);
61     pinMode(B_Back, OUTPUT);
62     pixels.begin();
63     // 밝기는 0에서 255까지의 값을 가짐
64     pixels.setBrightness(10);
65     pixels.clear();
66 }
67

```

```
68 void loop()
69 {
70     delay(2000);
71     // h는 습도, t는 온도
72     int h = dht.readHumidity();
73     int t = dht.readTemperature();
74
75     //LCD 화면 지우기
76     lcd.clear();
77
78     // 온도를 확인하여 값을 적절히 변경하기
79     if(t < 23) {
80         //0행 0열 커서위치
81         lcd.setCursor(0,0);
82         lcd.print("T:");
83         lcd.print(t);
84         lcd.print("C, It's Cold");
85         //1행 0열 커서위치
86         lcd.setCursor(0,1);
87         lcd.print("Light ON!");
88     }
```

```
89     for(int i=0; i<8; i++) {
90         pixels.setPixelColor(i,255,0,0);
91         pixels.show();
92     }
93 }
94 else {
95     for(int i=0; i<8; i++) {
96         pixels.setPixelColor(i,0,0,0);
97         pixels.show();
98     }
99
100     lcd.clear();
101     lcd.setCursor(0,0);
102     lcd.print("T:");
103     lcd.print(t);
104     lcd.print("C, ");
105     lcd.print("Light OFF!");
106     lcd.setCursor(0,1);
107     lcd.print("Optimal temp!");
108 }
109 }
```

```
21  #include <Adafruit_NeoPixel.h>
22  #include <LiquidCrystal_I2C.h>
23  #include "DHT.h"
24
25  // DHT 센서 핀 정의
26  #define DHTPIN A2
27  // DHT 센서의 종류
28  #define DHTTYPE DHT11
29  // 토양 온습도 센서 핀 번호
30  #define Soil_Sensor A1
31  #define A_Front 7 // 쿨링팬 왼쪽 A의 전진방향
32  #define A_Back 4 // 쿨링팬 왼쪽 A의 후진방향
33  #define B_Front 8 // 쿨링팬 오른쪽 B의 전진방향
34  #define B_Back 12 // 쿨링팬 오른쪽 B의 후진방향
35  #define numPixels 8 // 스트립 개수
36  #define pixelsPin 5 // PWM 핀번호 사용
37  #define Buzzer_pin 6 // 멜로디 IC 또는 부저로 사용
38
39  LiquidCrystal_I2C lcd(0x27, 16, 2);
40  // 정의된 핀과 센서 유형으로 dht 객체 생성
41  DHT dht(DHTPIN, DHTTYPE);
42  // (NEO_GRB + NEO_KHZ800): 색상 형식과 통신 속도를 지정
43  Adafruit_NeoPixel pixels(numPixels, pixelsPin, NEO_GRB + NEO_KHZ800);
44
```



```
45 //-----  
46 // 부팅시 만 실행  
47 void setup() {  
48     // 시리얼통신 초기화(9600)bps  
49     Serial.begin(9600);  
50     pinMode(Buzzer_pin, OUTPUT);  
51     // LCD 초기화  
52     lcd.init();  
53     // 백라이트 활성화  
54     lcd.backlight();  
55     // LCD 화면 지우기  
56     lcd.clear();  
57     pixels.begin();  
58     // 밝기는 0에서 255까지의 값을 가짐  
59     pixels.setBrightness(10);  
60     pixels.clear();  
61 }  
62
```

```
63 void loop() {  
64     delay(2000);  
65     lcd.clear();  
66     // 토양 온습도 센서의 값을 읽어옴  
67     int soilSensorValue = analogRead(Soil_Sensor);  
68     Serial.println(soilSensorValue);  
69     // 토양습도값을 확인하여 값을 적절히 변경하기  
70     if(soilSensorValue > 600) {  
71         //0행 0열 커서위치  
72         lcd.setCursor(0,0);  
73         lcd.print("Soil:");  
74         lcd.print(soilSensorValue);  
75         //1행 0열 커서위치  
76         lcd.setCursor(0,1);  
77         lcd.print("Refill Water!");  
78     }
```

```
79     for(int i=0; i<8; i++) {
80         pixels.setPixelColor(i,255,0,0);
81         pixels.show();
82     }
83     // BUZZER ON
84     digitalWrite(Buzzer_pin, HIGH);
85     Serial.println("BUZZER ON");
86 }
87 else {
88     for(int i=0; i<8; i++) {
89         pixels.setPixelColor(i,0,255,0);
90         pixels.show();
91     }
```

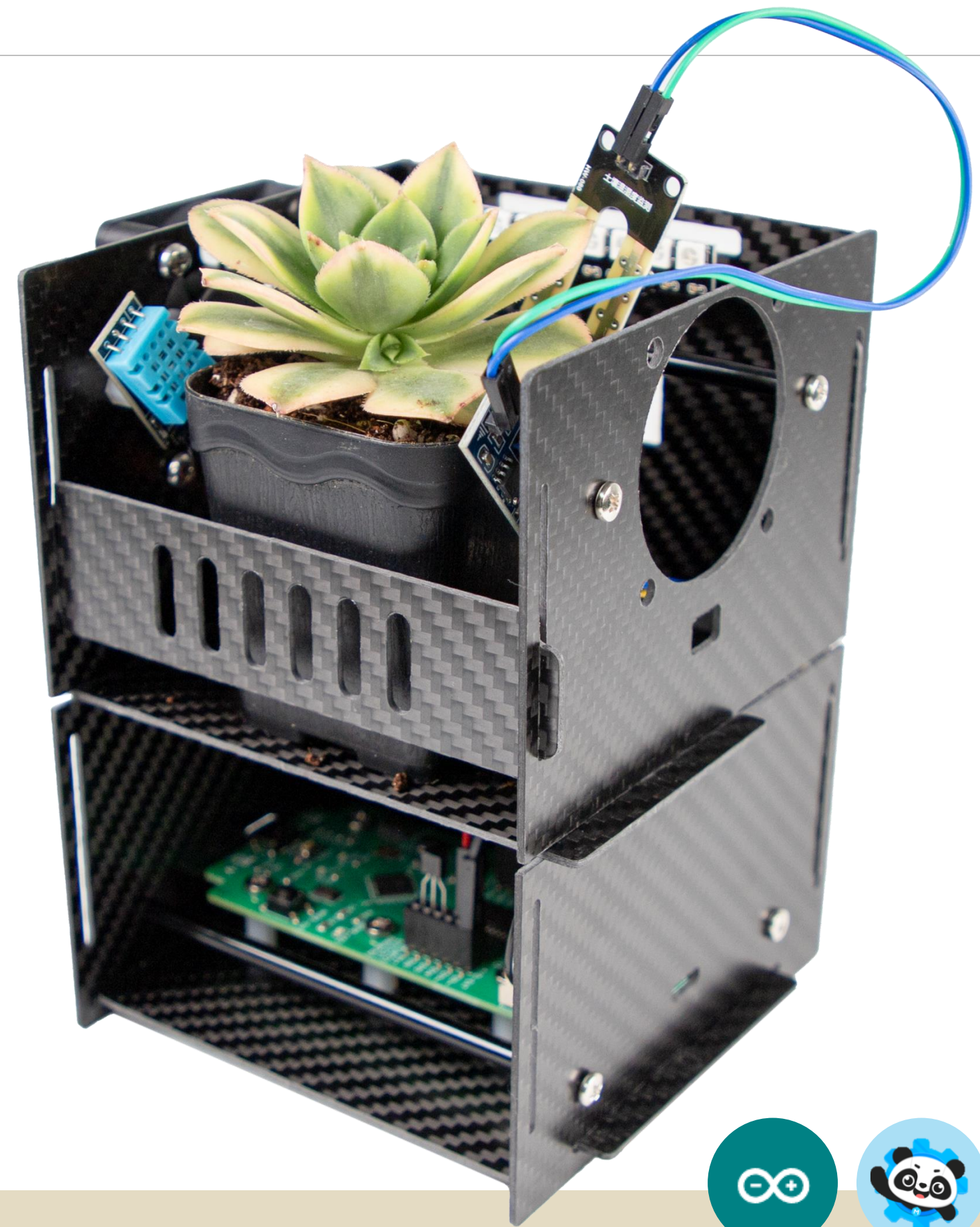
```
92     lcd.clear();
93     lcd.setCursor(0,0);
94     lcd.print("Good S_Moisture!");
95     lcd.setCursor(0,1);
96     lcd.print("Light is Green!");
97     // BUZZER OFF
98     digitalWrite(Buzzer_pin, LOW);
99     Serial.println("BUZZER OFF");
100 }
101 }
```



MEPL\_FARM



# LEVEL 3





```

21 #include <Adafruit_NeoPixel.h>
22 #include <LiquidCrystal_I2C.h>
23 #include "DHT.h"
24
25 // DHT 센서 핀 정의
26 #define DHTPIN A2
27 // DHT 센서의 종류
28 #define DHTTYPE DHT11
29 // 토양 온습도 센서 핀 번호
30 #define Soil_Sensor A1
31 #define A_Front 7 // 쿨링팬 왼쪽 A의 전진방향
32 #define A_Back 4 // 쿨링팬 왼쪽 A의 후진방향
33 #define B_Front 8 // 쿨링팬 오른쪽 B의 전진방향
34 #define B_Back 12 // 쿨링팬 오른쪽 B의 후진방향
35 #define numPixels 8 // 스트립 개수
36 #define pixelsPin 5 // PWM 핀번호 사용
37 #define Buzzer_pin 6 // 멜로디 IC 또는 부저로 사용
38 #define Button_pin A0
39 #define Led_pin 13
40

```

```

41 // 버튼상태 변수 선언
42 int Button_state = 0;
43 int LED_state = 0;
44 int FAN_state = 0;
45 String bluetooth_String = "";
46
47 LiquidCrystal_I2C lcd(0x27, 16, 2);
48 // 정의된 핀과 센서 유형으로 dht 객체 생성
49 DHT dht(DHTPIN, DHTTYPE);
50 // (NEO_GRB + NEO_KHZ800): 색상 형식과 통신 속도를 지정
51 Adafruit_NeoPixel pixels(numPixels, pixelsPin, NEO_GRB + NEO_KHZ800);
52

```

```
53 //-----  
54 // 부팅시 만 실행  
55 void setup() {  
56     // 시리얼통신 초기화(9600)bps  
57     Serial.begin(9600);  
58     //블루투스 시리얼 개방  
59     Serial1.begin(9600);  
60     // LCD 초기화  
61     pinMode(Buzzer_pin, OUTPUT);  
62     pinMode(Led_pin, OUTPUT);  
63  
64     lcd.init();  
65     // 백라이트 활성화  
66     lcd.backlight();  
67     // LCD 화면 지우기  
68     lcd.clear();  
69
```

```
70     //0행 0열 커서위치  
71     lcd.setCursor(0,0);  
72     lcd.print("Hello,MetaPlace!");  
73     //1행 0열 커서위치  
74     lcd.setCursor(0,1);  
75     lcd.print("Happy Day!");  
76  
77     pixels.begin();  
78     // 밝기는 0에서 255까지의 값을 가짐  
79     pixels.setBrightness(10);  
80     pixels.clear();  
81 }  
82
```

```
83 void loop() {
84     // Bluetooth_Rx_Tx:
85     // bluetooth_Serial ASCII형식
86     if (Serial1.available()) {
87         bluetooth_String = Serial1.read();
88     }
89 }
```

```
90 // bluetooth_String 값이 있다면
91 if(bluetooth_String != "") {
92     // ASCII Code 1 = "49"
93     if (bluetooth_String == "49" && LED_state==0) {
94         LED_state= 1;
95         Serial.println("LED ON!");
96         digitalWrite(Led_pin, HIGH);
97     }
98
99     else if(bluetooth_String == "49" && LED_state==1) {
100         LED_state= 0;
101         Serial.println("LED OFF!");
102         digitalWrite(Led_pin, LOW);
103     }
104 }
```



```
105   if(bluetooth_String == "50" && FAN_state==0)  {
106       FAN_state= 1;
107       Serial.println("FAN ON!");
108       // 모터 A를 전진 방향으로 회전
109       digitalWrite(A_Front, LOW);
110       digitalWrite(A_Back, HIGH);
111       // 모터 B를 전진 방향으로 회전
112       digitalWrite(B_Front, HIGH);
113       digitalWrite(B_Back, LOW);
114   }
115   else if(bluetooth_String == "50" && FAN_state==1)  {
116       FAN_state= 0;
117       Serial.println("FAN OFF!");
118       // 모터 A와 모터 B를 정지
119       digitalWrite(A_Front, LOW);
120       digitalWrite(A_Back, LOW);
121       digitalWrite(B_Front, LOW);
122       digitalWrite(B_Back, LOW);
123   }
```

```
124
125       //블루투스 값을 초기화
126       bluetooth_String = "";
127   }
128 }
```

## - 05번 파일에 추가하여 작성한다

```

41 // 버튼상태 변수 선언
42 int Button_state = 0;
43 int LED_state = 0;
44 int FAN_state = 0;
45 int LCD_state = 0;
46 int NEO_LED_state = 0;
47 String bluetooth_String = "";
48

```

```

128 if(bluetooth_String == "51" && LCD_state==0) {
129     LCD_state= 1;
130     delay(1000);
131     // h는 습도, t는 온도
132     int h = dht.readHumidity();
133     int t = dht.readTemperature();
134     // 토양 온습도 센서의 값을 읽어옴
135     int soilSensorValue = analogRead(Soil_Sensor);
136     //LCD 화면 지우기
137     lcd.clear();
138     //0행 0열 커서위치
139     lcd.setCursor(0,0);
140     lcd.print("H:");
141     lcd.print(h);
142     lcd.print("%, ");
143     //연속하여 프린트
144     lcd.print("T:");
145     lcd.print(t);
146     lcd.print("C");
147     //1행 0열 커서위치
148     lcd.setCursor(0,1);
149     lcd.print("Soil_value:");
150     lcd.print(soilSensorValue);
151 }

```

- **05번 파일에 추가**하여 작성한다

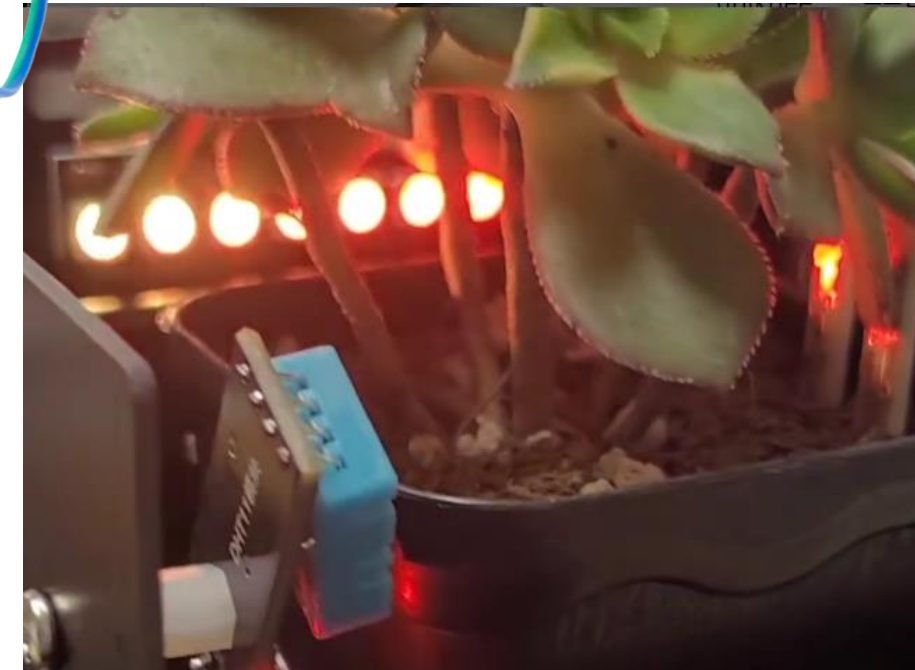
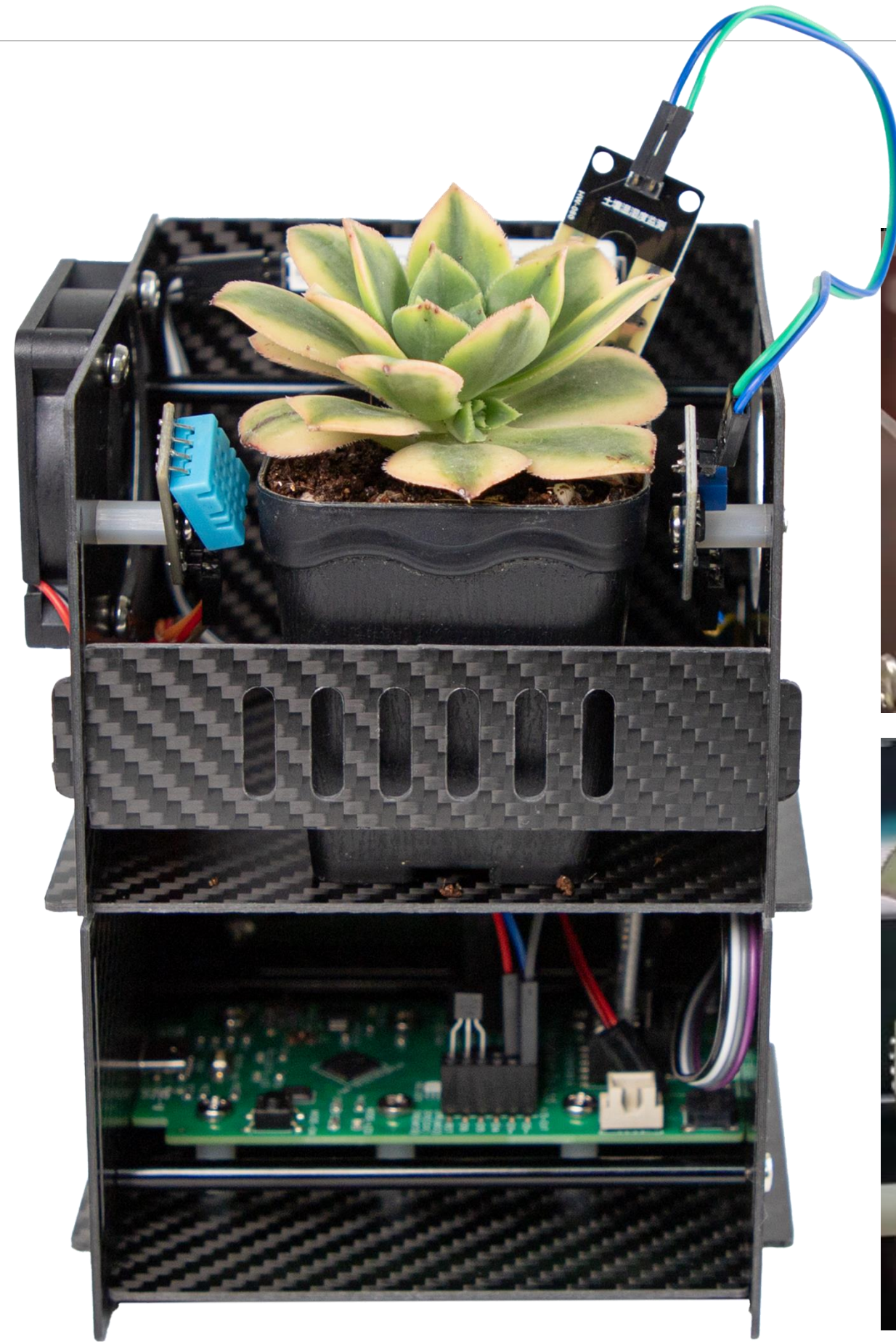
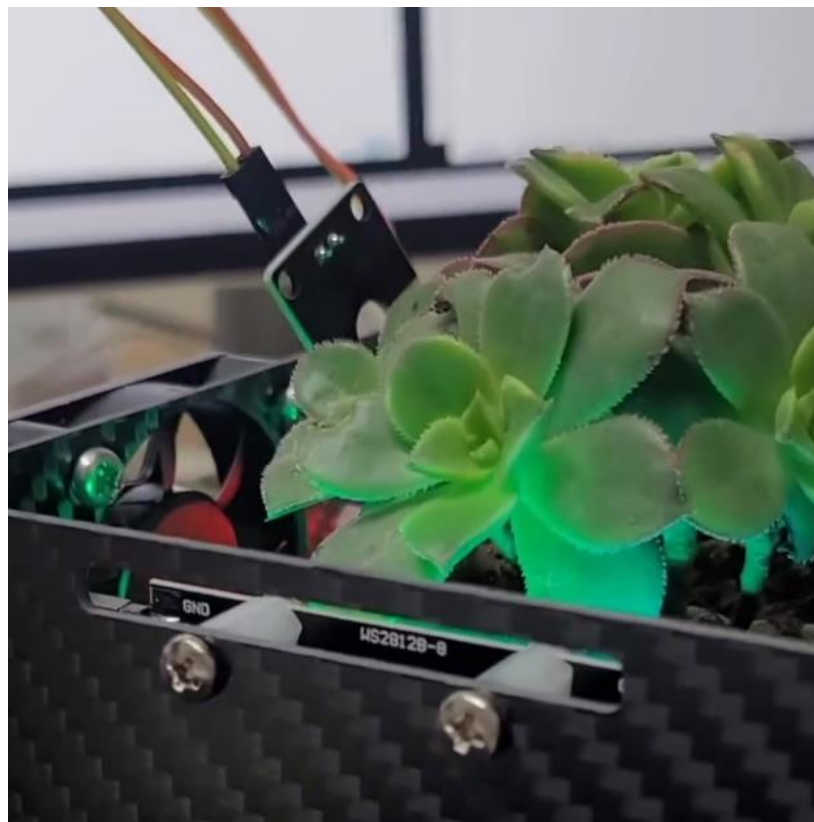
```
153  else if(bluetooth_String == "51" && LCD_state==1)  {  
154      LCD_state= 0;  
155      // LCD 화면 지우기  
156      lcd.clear();  
157      //0행 0열 커서위치  
158      lcd.setCursor(0,0);  
159      lcd.print("Hello,MetaPlace!");  
160      //1행 0열 커서위치  
161      lcd.setCursor(0,1);  
162      lcd.print("Happy Day!");  
163  }  
164
```



**- 05번 파일에 추가**하여 작성한다

```
165  if(bluetooth_String == "52" && NEO_LED_state==0)  {
166      |       NEO_LED_state = 1;
167      |       Serial.println("Light ON!");
168      |       for(int i=0; i<8; i++)  {
169      |           |       pixels.setPixelColor(i,255,255,0);
170      |           |       pixels.show();
171      |       }
172      |   }
173
174      |   else if(bluetooth_String == "52" && NEO_LED_state==1)  {
175      |       NEO_LED_state = 0;
176      |       Serial.println("Light OFF!");
177      |       for(int i=0; i<8; i++)  {
178      |           |       pixels.setPixelColor(i,0,0,0);
179      |           |       pixels.show();
180      |       }
181      |   }
182      |   //블루투스 값을 초기화
183      |   bluetooth_String = "";
184      |   }
185  }
```









GREEN GARDEN

THANK YOU



CONTACT

전화. 033-252-4787 | 메일. metaplace@naver.com

(주)메타플레이스  
강원특별자치도 춘천시 후석로 462번길 7 춘천ICT벤처센터 313호  
2024. 대표 : 정경희 / 기획 및 개발 : 권오상 / 디자인 : 마수민

