

# BLYNK를 이용한 IoT 제어

2022. 7. 14.

한밭대학교 모바일융합공학과  
이동호

# 본 실습의 목표

- IT에 관심있는 고등학생이 IoT 디바이스(Arduino와 센서)를 스마트폰의 Blynk 앱을 통해 제어할 수 있다.
- Blynk 란
  - 스마트폰의 앱과 인터넷을 통해 하드웨어를 쉽게 제어할 수 있는 플랫폼
- Arduino 란
  - 오픈소스 기반으로 마이크로 컨트롤러를 쉽게 제어할 수 있는 프로그램

# 회로 연결

↑ 컴퓨터와 USB로 연결

WEMOS D1  
(ESP8266 MCU)

DHT11  
(온습도센서)

MQ-135  
(유해가스/공기질 센서)

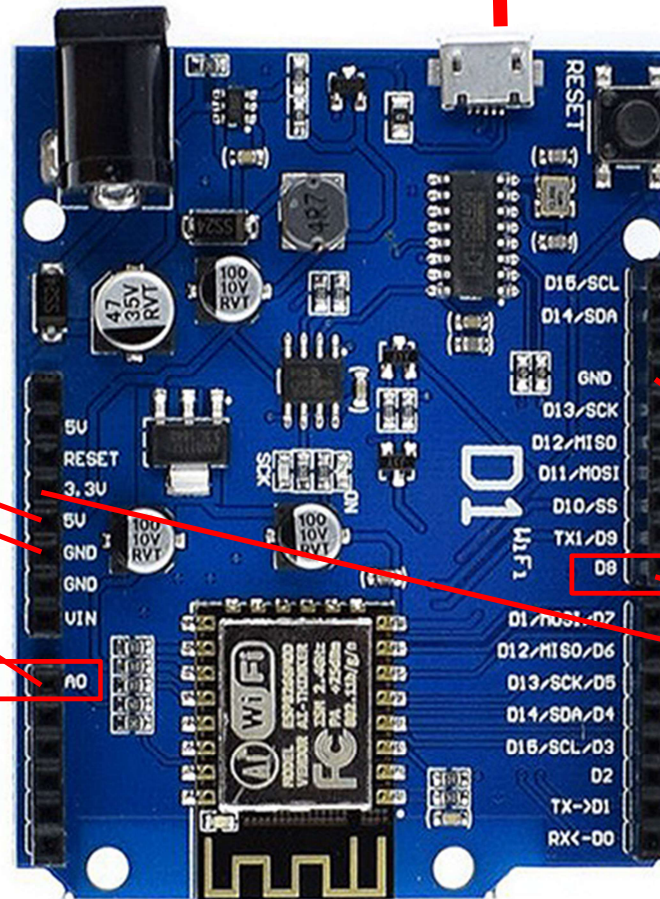


5V

GND

DATA

A0



GND

DATA

3.3V



# 아두이노 설치

- PC → Microsoft Store → Arduino IDE 설치
- Arduino IDE → 파일 → 환경설정 → 추가적인 보드 매니저 URLs에 아래 주소 입력
  - [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
- Arduino IDE → 툴 → 보드 → 보드 매니저 → esp32 검색하여 설치
- Arduino IDE → 툴 → 라이브러리 관리 → Blynk 검색하여 설치
- 스마트폰에 Blynk IoT앱 설치



# Library 설치 및 보드 선택

- 온습도센서 (DHT11)
  - Arduino IDE → 툴 → 라이브러리 관리 → DHT11 검색 → DHT sensor library 설치
- 유해가스센서 (MQ135)
  - Arduino IDE → 툴 → 라이브러리 관리 → MQ135 검색 → MQ135 설치

- 보드 선택 및 포트 선택
  - 툴 → 보드 → ESP8266 Boards → LOLIN(WEMOS) D1 R2 & mini

보드: "LOLIN(WEMOS) D1 R2 & mini"

Upload Speed: "921600"

CPU Frequency: "80 MHz"

Flash Size: "4MB (FS:2MB OTA:~1019KB)"

Debug port: "Disabled"

Debug Level: "None"

lwIP Variant: "v2 Lower Memory"

VTables: "Flash"

C++ Exceptions: "Disabled (new aborts on oom)"

Stack Protection: "Disabled"

Erase Flash: "Only Sketch"

SSL Support: "All SSL ciphers (most compatible)"

MMU: "32KB cache + 32KB IRAM (balanced)"

Non-32-Bit Access: "Use pgm\_read macros for IRAM/PROGMEM"

포트: "COM6"

보드 정보 열기

포트 번호는 각자 다름

# Blynk 계정 만들기

- 웹브라우저에서 blynk.io 방문
- Log In → Create new account → 이메일 넣고 Sign Up
- 이메일 확인 → Create Password → 이름 넣고 Next
- Next 눌러서 짹짹 넘어가기
- Quickstart
  - 1 Hardware → ESP8266, WiFi 선택 후 Next
  - 2 IDE → Arduino 선택 후 Next
  - 3 Blynk Library → Next
  - 4 Code → SSID에 Hanbat\_WLAN\_Guest, Password는 비움
    - 현재 화면 우측 코드①를 WiFi 설정에 사용할 것임
  - 여기서 넘어가지 말고 정지



# 제공된 코드 이용시

- <https://github.com/dhleeenu/arduino>
- 220713BLYNK\_MQ135\_DHT11.ino
- 앞장 코드로 덮어 쓰기

```
/* *****  
Copy code Download As File  
This is a simple demo of sending and receiving some data.  
Be sure to check out other examples!  
***** */  
  
// Template ID, Device Name and Auth Token are provided by the Blynk Cloud  
// See the Device Info tab, or Template settings  
  
#define BLYNK_TEMPLATE_ID "TMPLi6YQ3iXn"  
#define BLYNK_DEVICE_NAME "Quickstart Device"  
#define BLYNK_AUTH_TOKEN "dwSG4zEofpiNuN6T8if0EuKMeNHrn3N_"
```

```
220713BLYNK_MQ135_DHT11  
  
#define BLYNK_TEMPLATE_ID "TMPLyPSgTlA7"  
#define BLYNK_DEVICE_NAME "Quickstart Device"  
#define BLYNK_AUTH_TOKEN "uolV8IQNPqqkEI2ogs3zF1Wz5jdOqAXu"  
  
#define BLYNK_PRINT Serial  
  
#include <ESP8266WiFi.h>  
#include <BlynkSimpleEsp8266.h>  
#include <MQ135.h>  
#include <DHT.h>  
  
char auth[] = BLYNK_AUTH_TOKEN;  
  
char ssid[] = "Hanbat_WLAN_Guest"; //집에서 개인 WiFi로 변경  
char pass[] = ""; //집에서 비밀번호 변경
```

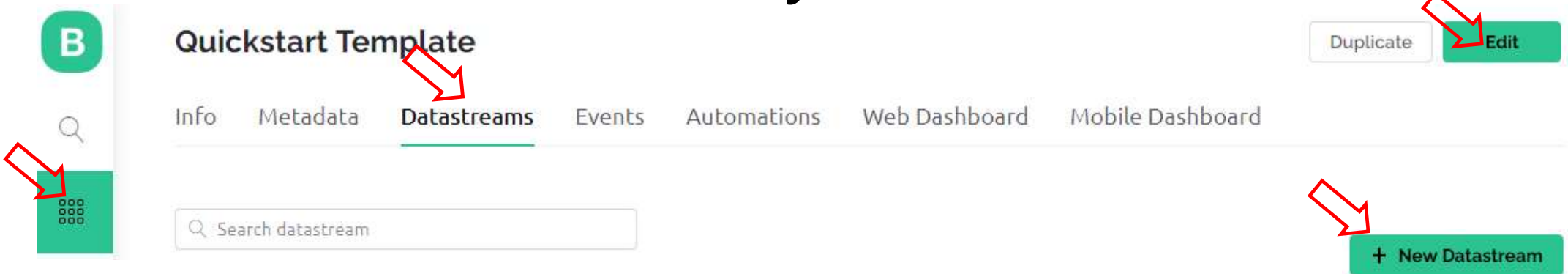
# 아두이노 업로드

- 저장 먼저하고
- ✓ 버튼 눌러서 컴파일 
- 아두이노 하단창에 에러가 없으면
- → 버튼 눌러서 코드를 아두이노에 업로드 
- Arduino IDE → 툴 → 시리얼 모니터링
- 값이 출력된다면 p6의 Blynk 웹사이트로 돌아가서 online 연결 되었는지 확인





# PC에서 Blynk 꾸미기



- + New Datastream  
→ Virtual Pin 추가
  - Humidity, V5, String
  - Temperature, V6, String
  - Air Quality, V2, String

## Virtual Pin Datastream







NAME:  ALIAS:

PIN:  DATA TYPE:

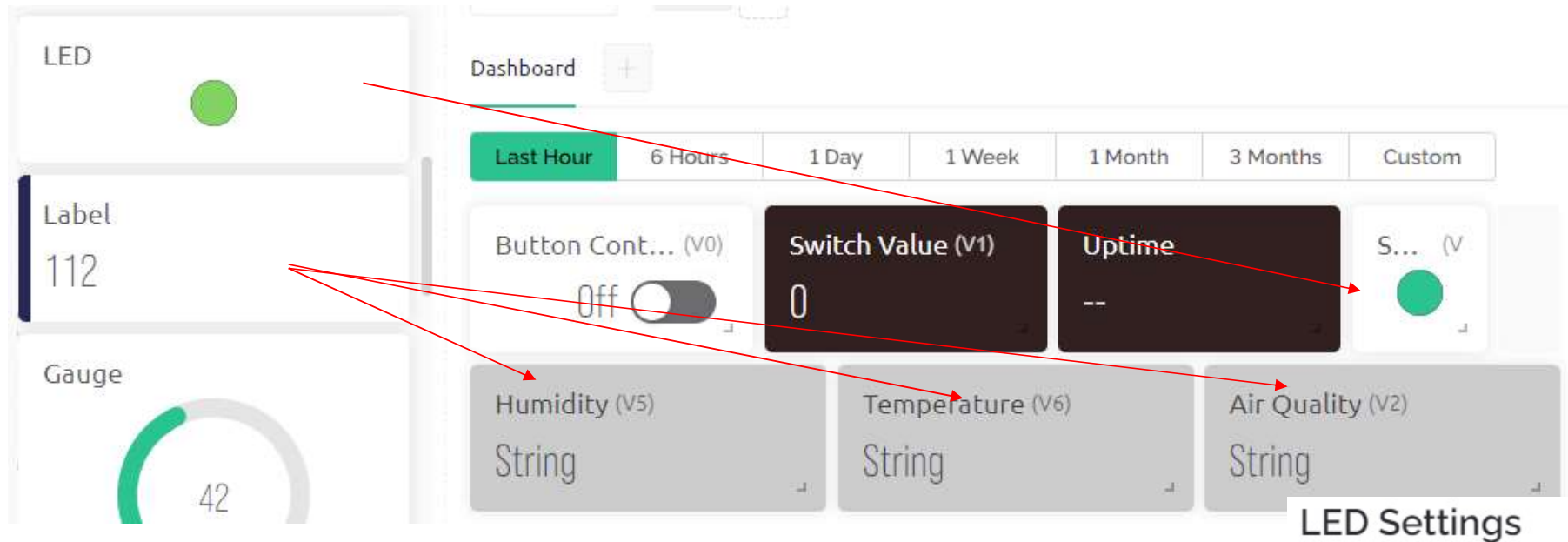
DEFAULT VALUE:

ADVANCED SETTINGS

🔍 Search datastream

Id	Name	Alias	Color	Pin	Data Type	Units
1	Switch Control	Switch Control		V0	Integer	
2	Switch Value	Switch Value		V1	Integer	
4	Button Image	Button Image		V3	String	
5	Humidity	Humidity		V5	String	
6	Temperature	Temperature		V6	String	
7	Air Quality	Air Quality		V2	String	

- Web Dashboard



- Label을 끌어와서 V5, V7, V2 추가
- LED를 끌어와서 오른쪽과 같이 V1 추가

# PC에서 Blynk 동작 확인

The screenshot displays the Blynk PC interface. On the left, a sidebar shows the organization 'My organization - 5988AF' and a list of devices. The 'My devices' section is highlighted. A red arrow points to the 'Quickstart Device' button. The main area shows the 'My devices' section with a list of devices. A red arrow points to the 'Quickstart Device' button. To the right, two panels show the device's status: 'Off' (熄) and 'On' (화).

**My organization - 5988AF**

**My devices**

1 Device

Device name

Quickstart Device

Button C... 0.. ☐ **熄**

Switch V... 0

Uptime --

C-F C

Temperature 30

Humidity 70

Air Quality 17

Button C... 0.. ☒ **화**

Switch V... 1

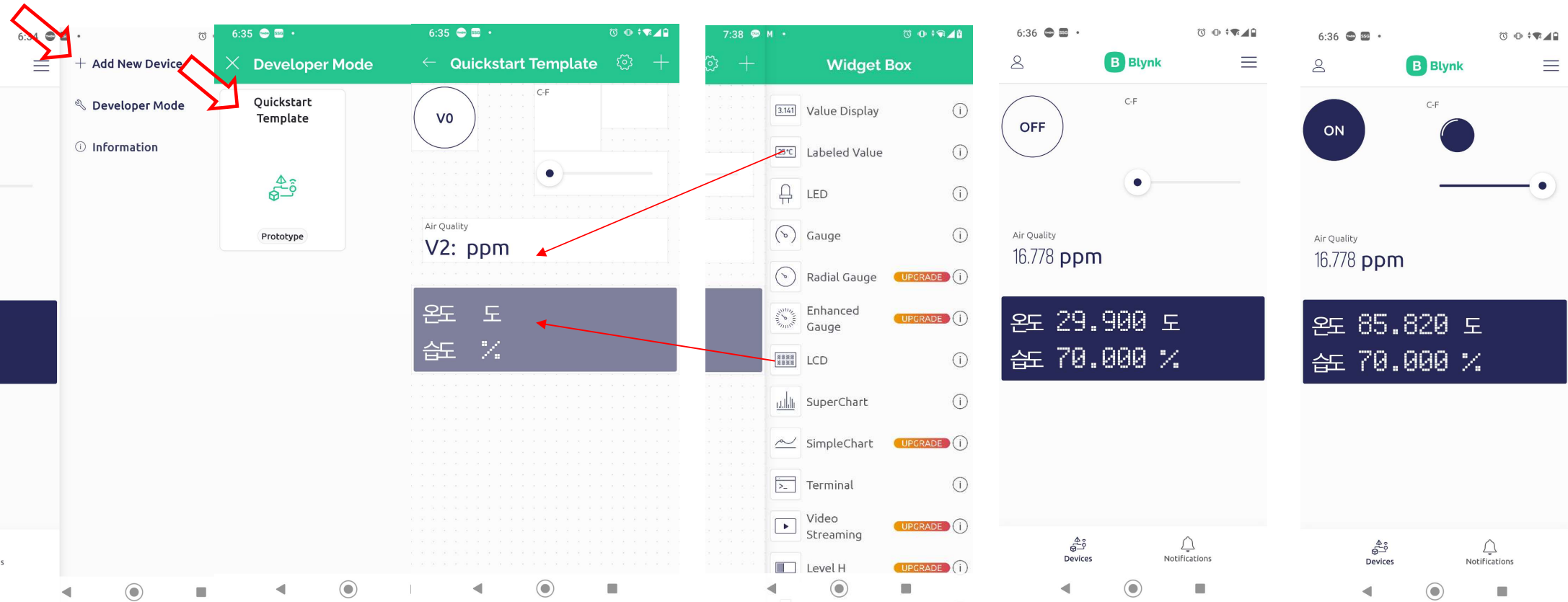
C-F C

Temperature 86

Humidity 70

Air Quality 17

# 스마트폰에서 Blynk 동작 확인



• 각자 자유롭게 화면을 꾸며 보세요.

# 직접 코드 작성시 p14-p17 참조 아두이노 코딩

- MQ135 예제 MQT135\_DHTxx ②와 WiFi 동작①(p6), 두개의 코드를 합쳐볼 것임
  - Arduino IDE → 파일 → 예제 → MQ135 → MQT135\_DHTxx ②
- 아두이노용 모든 센서는 코드가 공개되어 있음
- 가장 간결하고 쓰기 편리한 코드는 아두이노 내의 예제 코드임
- 필요한 기능의 코드를 찾아 조합함으로써 원하는 동작을 만들 수 있음

# 합쳐보자

- ② MQT135\_DHTxx 을 파일 이름 바꾸어 저장
- ①의 Blynk의 WiFi 연결정보를 ②에 넣기

```
#define BLYNK_TEMPLATE_ID      "TMPLuunVp00M"  
#define BLYNK_DEVICE_NAME     "Quickstart Device"  
#define BLYNK_AUTH_TOKEN      "YE1mT-n6jHAF1ssZoeJ1dhFbF0-KggMn"
```

추가  
각자  
코드  
다름

```
#define BLYNK_PRINT Serial
```

```
#include <ESP8266WiFi.h>  
#include <BlynkSimpleEsp8266.h>
```

```
char auth[] = BLYNK_AUTH_TOKEN;
```

```
// Your WiFi credentials.  
// Set password to "" for open networks.  
char ssid[] = "Hanbat_WLAN_Guest";  
char pass[] = "";
```

추가

집에서는 개인 WiFi 로 변경  
- SSID와 password 교체

```
#define PIN_MQ135 A2 // MQ135 Analog Input Pin  
#define DHTPIN 2 // DHT Digital Input Pin  
#define DHTTYPE DHT11 // DHT11 or DHT22, depends on your sensor
```



```
#define PIN_MQ135 A0 // MQ135 Analog Input Pin  
#define DHTPIN 0 // DHT Digital Input Pin  
#define DHTTYPE DHT11 // DHT11 or DHT22, depends on your sensor
```

실제 회로 연결에 맞게 수정

```
MQ135 mql35_sensor(PIN_MQ135);  
DHT dht(DHTPIN, DHTTYPE);
```

```
int value=0; //섭씨-화씨 변환용 변수
```

직접 타이핑하여 추가

```
float temperature, humidity, tc; // Temp and Humid floats, will be measured by the DHT
```

- ①의 함수를 ②에 넣기
  - Blynk의 버튼을 통해 섭씨-화씨 변환에 사용

```
BLYNK_WRITE(V0)
{
  // Set incoming value from pin
  int value = param.asInt();

  // Update state
  Blynk.virtualWrite(V1, value);
}
```



```
BLYNK_WRITE(V0)
{
  // Set incoming value from pin
  value = param.asInt();

  // Update state
  Blynk.virtualWrite(V1, value);
}
```

- 통신 속도 및 Blynk WiFi 변경

```
void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
}
```

직접 타이핑하여  
수정

- loop() 함수 수정

```
void loop() {

  Blynk.run();

  humidity = dht.readHumidity();
  temperature = dht.readTemperature(value);
  tc = dht.readTemperature(0);
  // Check if any reads failed and exit early (to try again).
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  float rzero = mq135_sensor.getRZero();
  float correctedRZero = mq135_sensor.getCorrectedRZero(tc, humidity);
  float resistance = mq135_sensor.getResistance();
  float ppm = mq135_sensor.getPPM();
  float correctedPPM = mq135_sensor.getCorrectedPPM(tc, humidity);
}
```

추가  
섭씨온도를 tc에 저장

섭씨온도값(tc)을 이용  
해 MQ135값 보정



```
Blynk.virtualWrite(V5, humidity); //Blynk 서버의 V5 가상핀에 습도값 전달
Blynk.virtualWrite(V6, temperature); //Blynk 서버의 V6 가상핀에 온도값 전달
Blynk.virtualWrite(V2, correctedPPM); //Blynk 서버의 V2 가상핀에 공기질값 전달
```

추가  
Blynk의 가상핀에 측정값 전달

```
Serial.print(value); //0이면 섭씨, 1이면 화씨
Serial.print(F(" Humidity: ")); //습도:
Serial.print(humidity); //습도값
Serial.print(F("% Temperature: ")); //온도:
Serial.println(temperature); //온도값, 줄바꿈
```

추가  
아두이노 시리얼 모니터로 온습도값 출력

```
Serial.print("MQ135 RZero: ");
Serial.print(rzero);
Serial.print("\t Corrected RZero: ");
Serial.print(correctedRZero);
Serial.print("\t Resistance: ");
Serial.print(resistance);
Serial.print("\t PPM: ");
Serial.print(ppm);
Serial.print("ppm");
Serial.print("\t Corrected PPM: ");
Serial.print(correctedPPM);
Serial.println("ppm");
```

```
delay(1000);
```

1000ms=1초 마다 갱신

```
}
```

# 마무리

- 센서를 추가하여 여러 가지 기능을 만들어 볼 수 있습니다.
- 센서별 코드는 아두이노 예제에서 또는 구글에서 검색하여 찾을 수 있습니다.
- 내가 만든 IoT 디바이스는 5V 전원만 연결하면 24시간 동작하면서 내 폰으로 값을 보내줍니다.
- 한밭대학교 모바일융합공학과에 많은 관심을 갖고 행사에 참여해 주시어 감사합니다.