

# Chap05. 다양한 센서 응용

5.1 온습도 센서

5.2 인체감지 센서

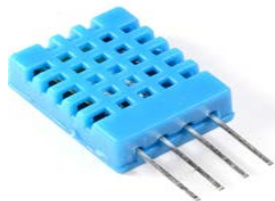
5.3 미세먼지 센서

5.4 초음파 센서

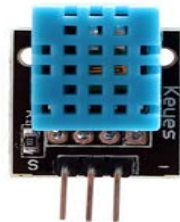
# 5.1 온습도 센서

## ◆ 온습도 센서(DHT11)

- 20~90%의 습도와 0~50℃의 온도를 측정할 수 있는 센서
- 습도는  $\pm 5\%$ , 온도는  $\pm 2^{\circ}\text{C}$ 의 오차 범위
- 측정된 값은 디지털 데이터로 출력



DHT11

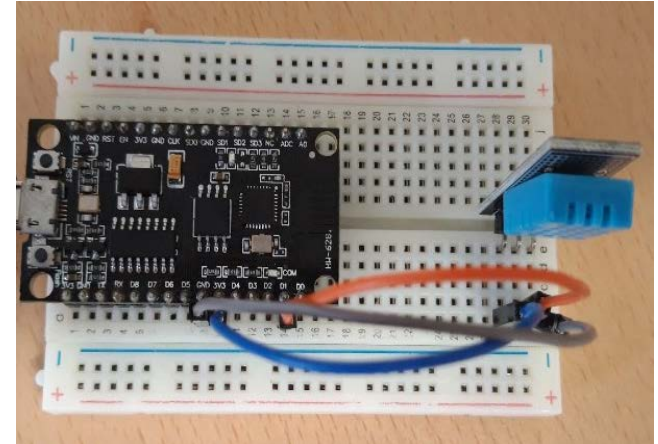
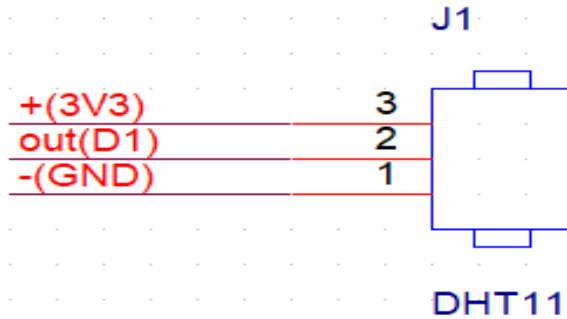
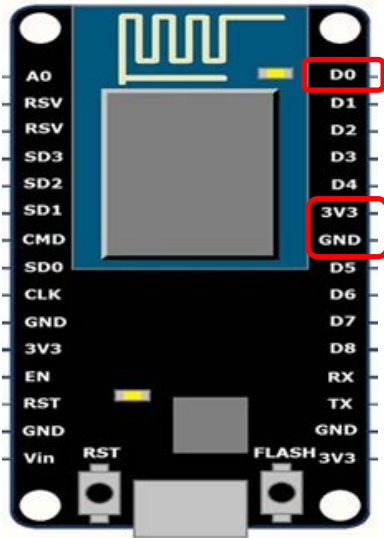


DHT11 모듈

# 5.1 온습도 센서

## ◆ 5.1.1 온습도 센서 값 시리얼 프린터로 출력 - (MCU)

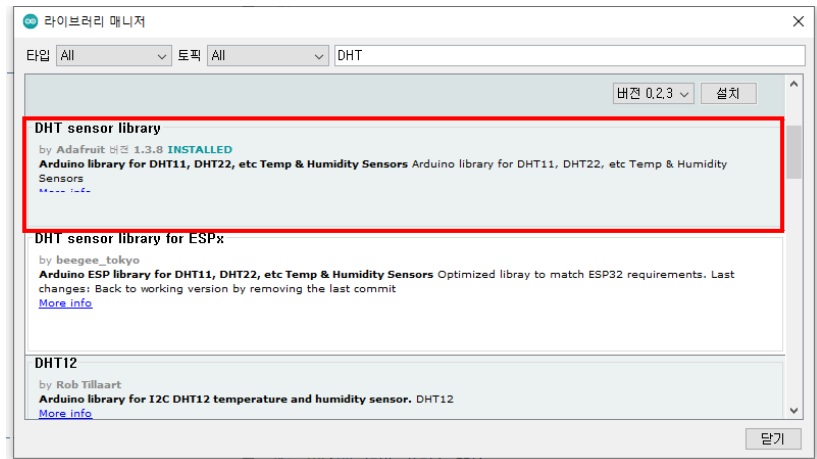
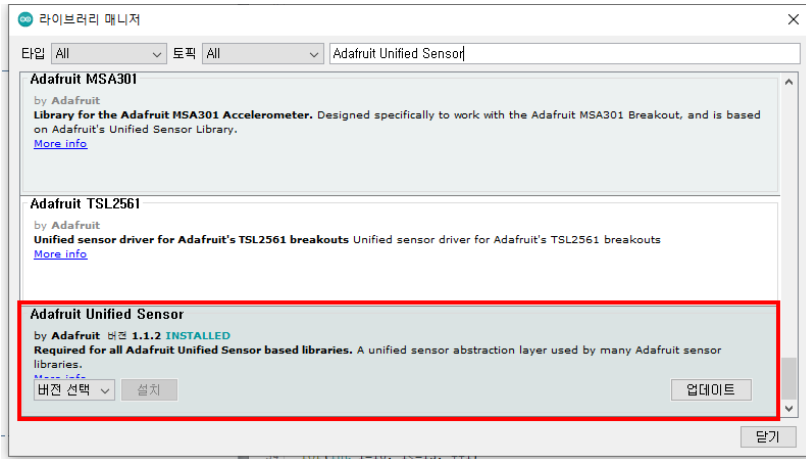
- 회로도 및 연결도



# 5.1 온습도 센서

## ◆ 5.1.1 온습도 센서 시리얼 프린터로 출력

### ● 라이브러리 설치



# 5.1 온습도 센서

## ◆ 5.1.1 온습도 센서 시리얼 프린터로 출력

### ● 소스 코드

```
2 // DHT11 Control
3
4 #include <DHT.h>
5 #define DHTTYPE DHT11
6 #define DHTPIN D1
7
8 DHT dht(DHTPIN, DHTTYPE);
9 void setup()
10 {
11     Serial.begin(9600);
12     dht.begin();
13 }
14
15 void loop()
16 {
17     // put your main code here, to run repeatedly:
18     float h = dht.readHumidity();
19     float t = dht.readTemperature();
20     Serial.print("습도 = ");
21     Serial.print(h);
22     Serial.print(" ");
23     Serial.print("온도 : ");
24     Serial.println(t);
25     delay(500);
26 }
```

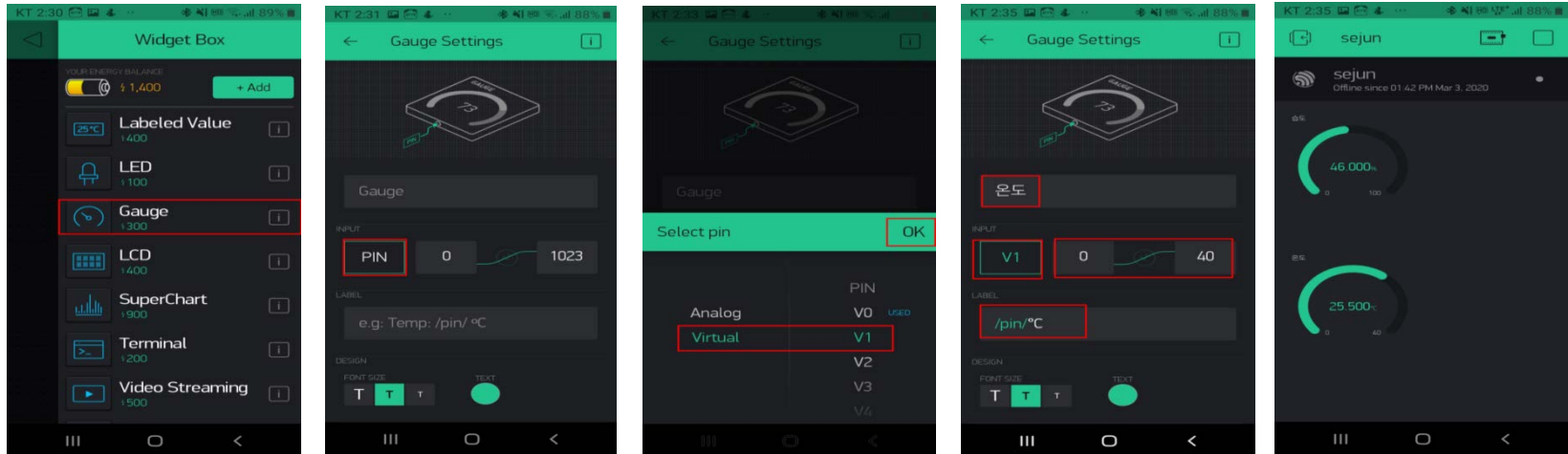
COM9

습도 = 31.00	온도 : 26.20
습도 = 31.00	온도 : 26.20
습도 = 31.00	온도 : 26.20
습도 = 31.00	온도 : 26.20
습도 = 53.00	온도 : 26.20
습도 = 53.00	온도 : 26.20
습도 = 53.00	온도 : 26.20
습도 = 53.00	온도 : 26.20
습도 = 95.00	온도 : 26.40
습도 = 95.00	온도 : 26.40
습도 = 95.00	온도 : 26.40
습도 = 95.00	온도 : 26.40
습도 = 62.00	온도 : 26.40
습도 = 62.00	온도 : 26.40
습도 = 62.00	온도 : 26.40

# 5.1 온습도 센서

## ◆ 5.1.2 온습도 센서 값 Blynk Gauge로 출력(MCU → 폰)

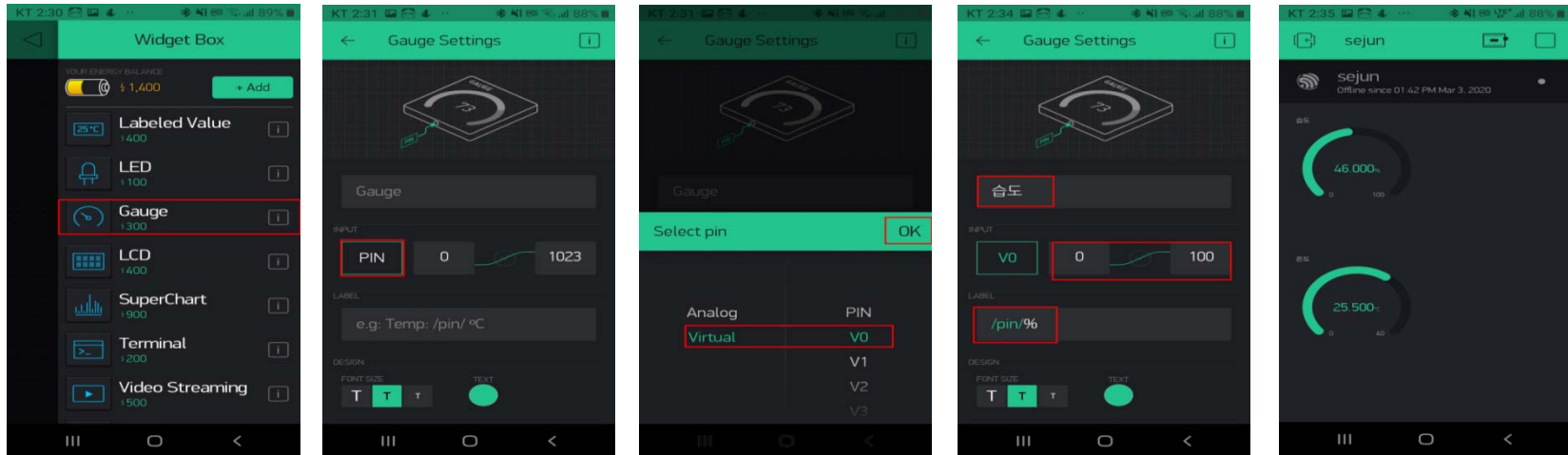
- Widget 설정



# 5.1 온습도 센서

## ◆ 5.1.2 온습도 센서 값 Blynk Gauge로 출력(MCU → 폰)

### ● Widget 설정



# 5.1 온습도 센서

## ◆ 5.1.2 온습도 센서 값 Blynk Gauge로 출력(MCU → 폰)

### ● 소스 코드

```
2 // DHT11 Control Vpin
3
4 #define BLYNK_PRINT Serial
5 #include <ESP8266WiFi.h>
6 #include <BlynkSimpleEsp8266.h>
7 #include <DHT.h>
8 #define DHTTYPE DHT11
9
10 #define DHTPIN D1
11 DHT dht(DHTPIN, DHTTYPE);
12
13 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4";
14
15 // Your WiFi credentials.
16 // Set password to "" for open networks.
17 char ssid[] = "sjpark";
18 char pass[] = "12345678";
```

```
20 void setup()
21 {
22   Serial.begin(9600);
23   Blynk.begin(auth, ssid, pass);
24   dht.begin();
25 }
26
27 void loop()
28 {
29   Blynk.run();
30
31   float h = dht.readHumidity();
32   float t = dht.readTemperature();
33
34   Blynk.virtualWrite(V0,h);
35   Blynk.virtualWrite(V1,t);
36   delay(500);
37 }
```



# 5.1 온습도 센서

## ◆ 실습문제1(MCU → 폰)

- 온습도 센서를 연결하고 하드웨어를 구성하시오.
- Blynk LCD(1)와 Value Display(2)를 이용하여 온도와 습도를 출력하시오.
  - LCD – 1<sup>st</sup> line(온도), 2<sup>nd</sup> line(습도)
  - Value Display(온도), Value Display(습도)

# 5.1 온습도 센서

## ◆ 실습문제2(MCU → 폰)

- 온습도 센서를 연결하고 하드웨어를 구성하시오.
- Blynk Value Display(2), LED(2) 그리고 Button(2)를 이용하여 온도와 습도 값에 따른 버튼을 제어하시오.
  - Value Display(온도), Value Display(습도)
  - LED(온도), LED(습도)-제품 대응(에어컨, 제습기)
  - Button(에어컨 - SWITCH - ON/OFF(에어컨))
  - Button(제습기 - SWITCH - ON/OFF(제습기))

## 5.2 인체감지 센서

### ◆ 인체감지 센서(HC-SR501)

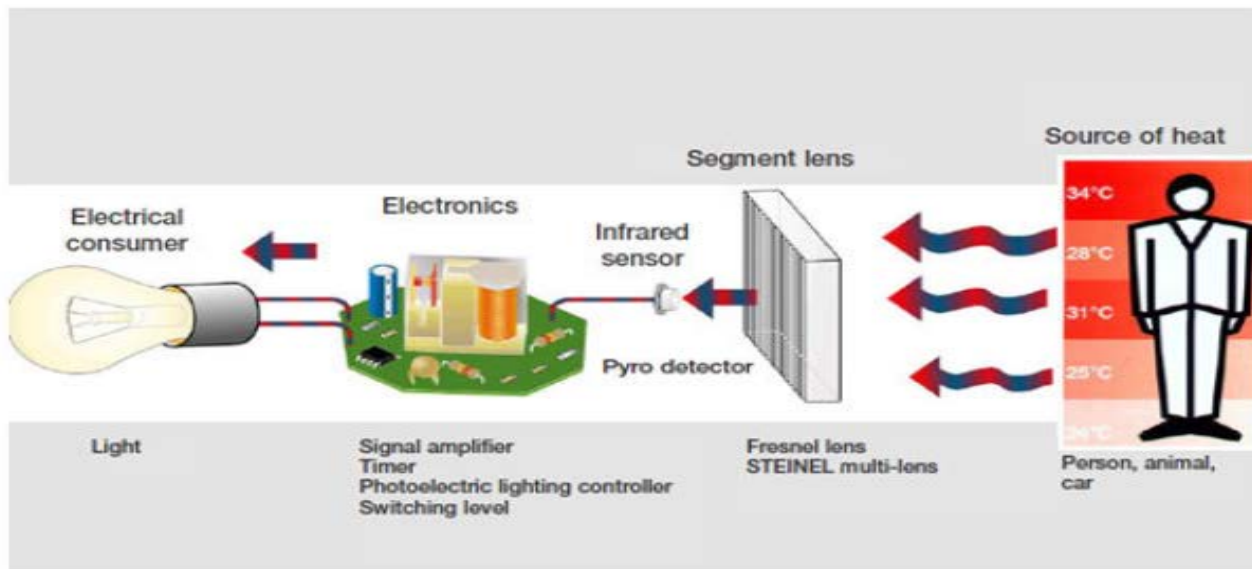
- PIR(Passive Infrared Sensor)는 적외선 감지 센서
- 사람의 인체에서 36.5도의 열에서 방출되는 열파장 감지
- 자동으로 형광등을 켜주는 장치에 부착되어있는 센서모듈
  - 아파트 현관/계단/복도에서 사람 인식



## 5.2 인체감지 센서

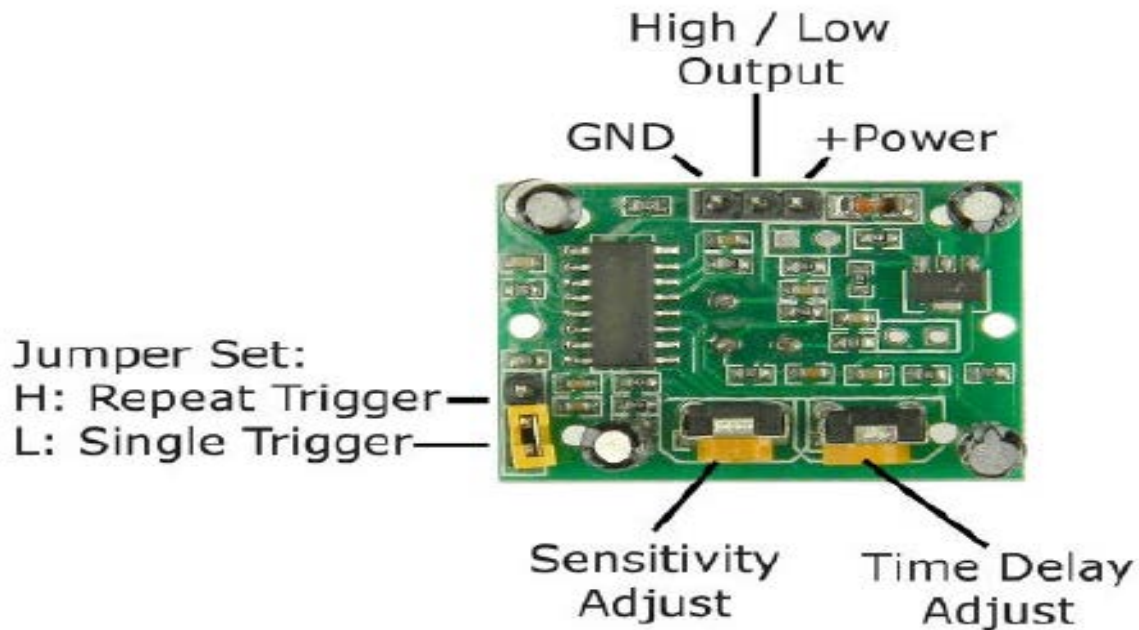
### ◆ 인체감지 센서(HC-SR501) 원리

- 출처: <https://pirtechnology.wordpress.com/2011/09/09/hello-world/>



## 5.2 인체감지 센서

### ◆ 인체감지 센서(HC-SR501) 모듈 인터페이스



## 5.2 인체감지 센서

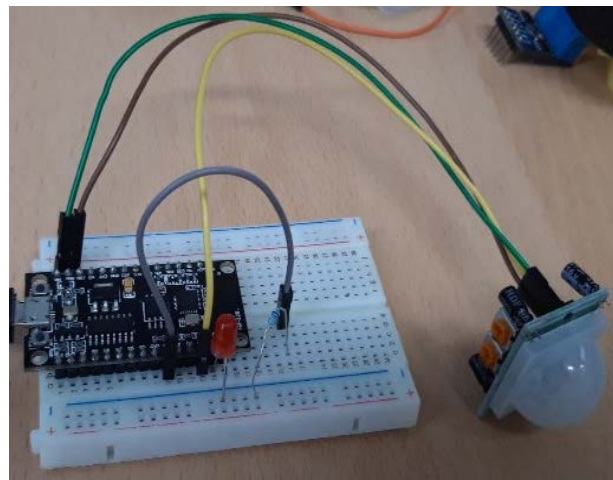
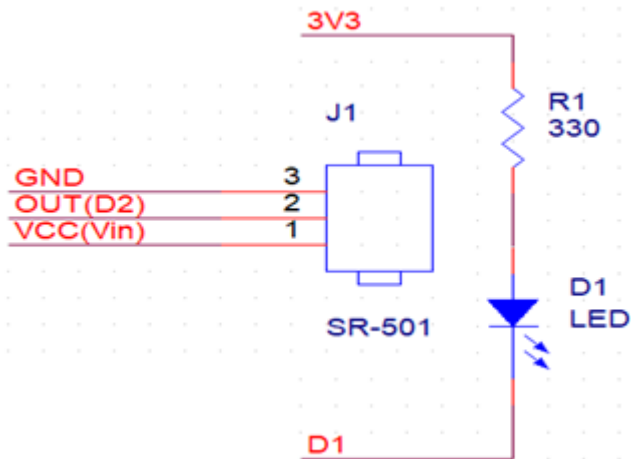
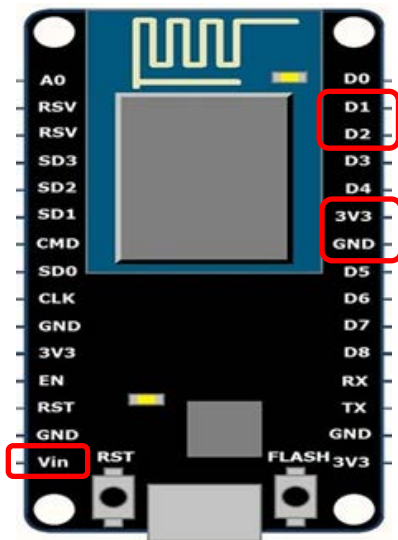
### ◆ 인체감지 센서(HC-SR501) 모듈 인터페이스

- VCC, GND : 입력 전압(5V) 및 GND
- Output : 인체 감지(HIGH), 인체 미감지(LOW)
- Jumper Set : 반복적으로 인체를 감지할 것인지 결정하는 점퍼
- Sensitivity Adjust : 인체를 인식하는 감도 조정
- Time Delay Adjust : 출력신호의 시간 조정(5~200 sec)

## 5.2 인체감지 센서

### ◆ 5.2.1 인체감지 센서로 LED 제어 - (MCU)

- 회로도 및 연결도



## 5.2 인체감지 센서

### ◆ 5.2.1 인체감지 센서로 LED 제어 - (MCU)

- 소스 코드

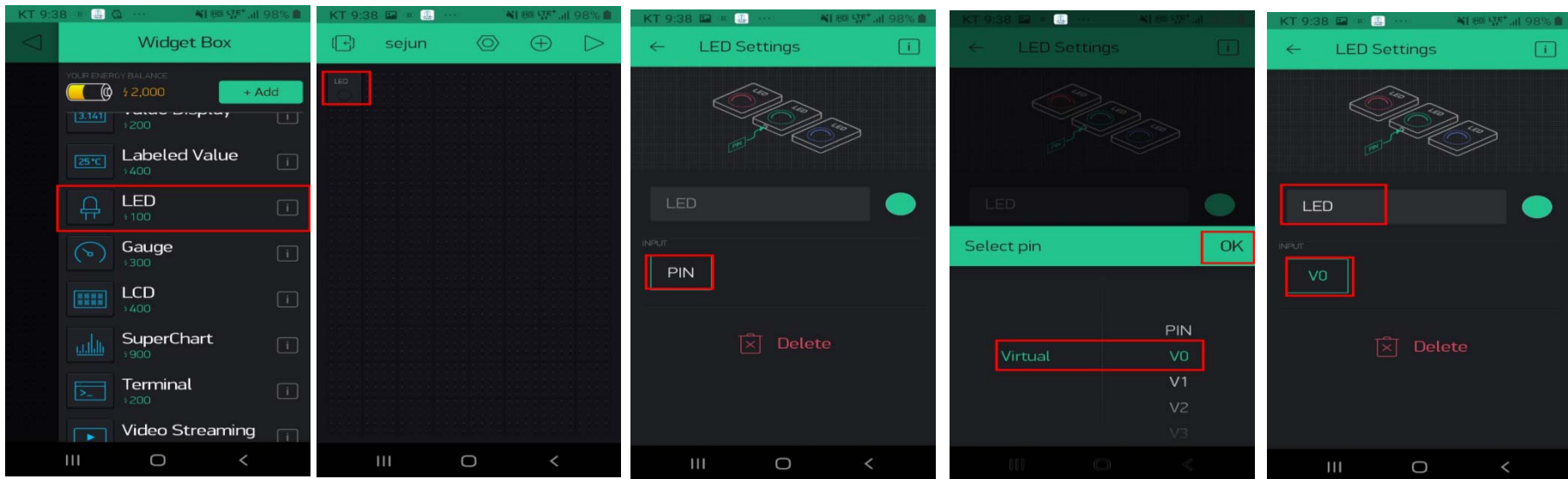
```
2 // PIR Control
3
4 void setup()
5 {
6   pinMode(D1, OUTPUT);
7   pinMode(D2, INPUT_PULLUP);
8   Serial.begin(9600);
9 }
10 void loop()
11 {
12   int pir = digitalRead(D2);
13
14   Serial.print("PIR : ");
15   Serial.println(pir);
16
17   if(pir==1) digitalWrite(D1, 0); // set the LED on
18   else digitalWrite(D1, 1);      // set the LED off
19   delay(500);
20 }
```



## 5.2 인체감지 센서

### ◆ 5.2.2 인체감지 센서로 Blynk LED 제어 - (MCU → 폰)

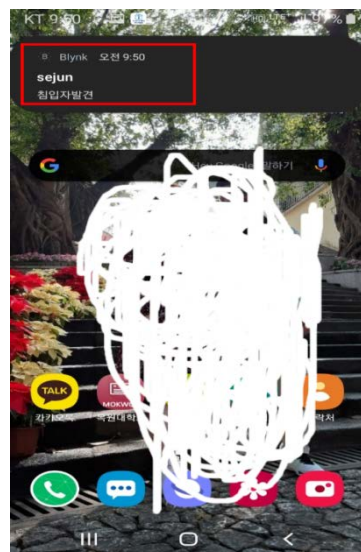
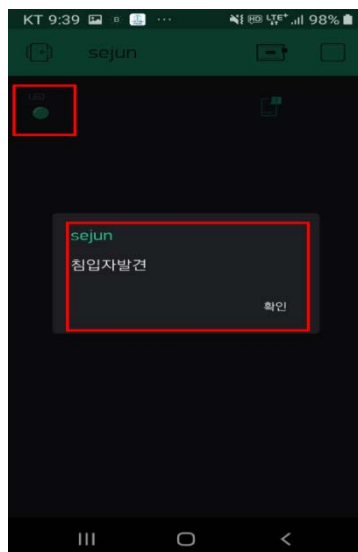
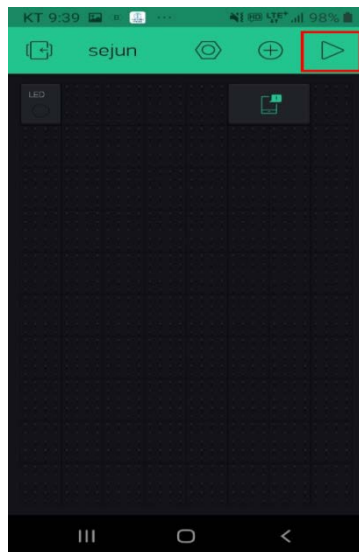
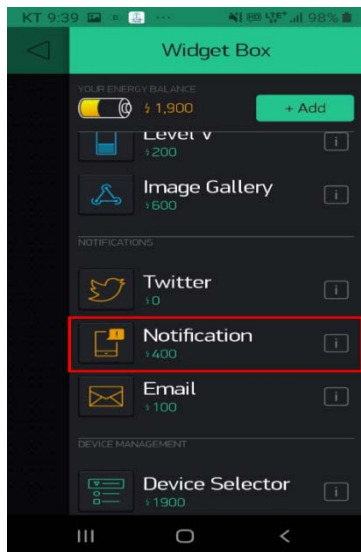
#### ● Widget 설정



## 5.2 인체감지 센서

### ◆ 5.2.2 인체감지 센서로 Blynk LED 제어 - (MCU → 폰)

#### ● Widget 설정



## 5.2 인체감지 센서

### ◆ 5.2.2 인체감지 센서로 Blynk LED 제어 - (MCU → 폰)

#### ● 소스 코드

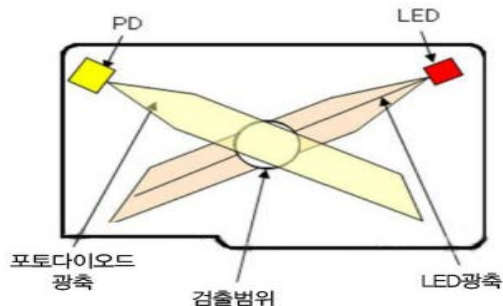
```
2 // PIR Control vpin
3
4 #define BLYNK_PRINT Serial
5 #include <ESP8266WiFi.h>
6 #include <BlynkSimpleEsp8266.h>
7
8 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4";
9
10 // Your WiFi credentials.
11 // Set password to "" for open networks.
12 char ssid[] = "sjpark";
13 char pass[] = "12345678";
14
15 WidgetLED led(V0);
```

```
17 void setup()
18 {
19   Serial.begin(9600);
20   Blynk.begin(auth, ssid, pass);
21   pinMode(D2, INPUT_PULLUP);
22 }
23
24 void loop()
25 {
26   Blynk.run();
27   int pir = digitalRead(D2);
28
29   if(pir==1)
30   {
31     led.on();           // LED 점등
32     Blynk.notify("침입자발견");
33     delay(5000);
34   }
35   else led.off();      // LED 소등
36 }
```

## 5.3 미세먼지 센서

### ◆ 미세먼지 센서(GP2Y1023AU0F)

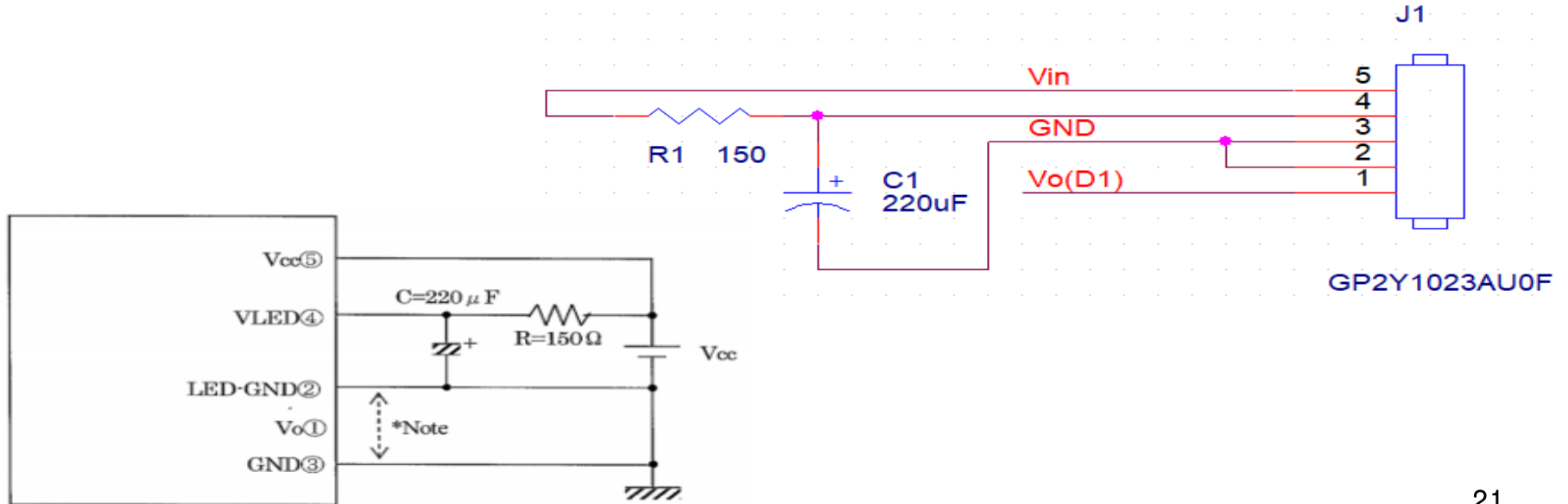
- PWM 펄스 값을 출력하여 공기에서 검출되는 먼지 검출 키트
- 담배 연기와 같은 매우 미세한 입자까지 검출 가능
- 측정 범위(0~250ug/m<sup>3</sup>)



# 5.3 미세먼지 센서

## ◆ 5.3.1 미세먼지 센서 값 시리얼 프린터로 출력 - (MCU)

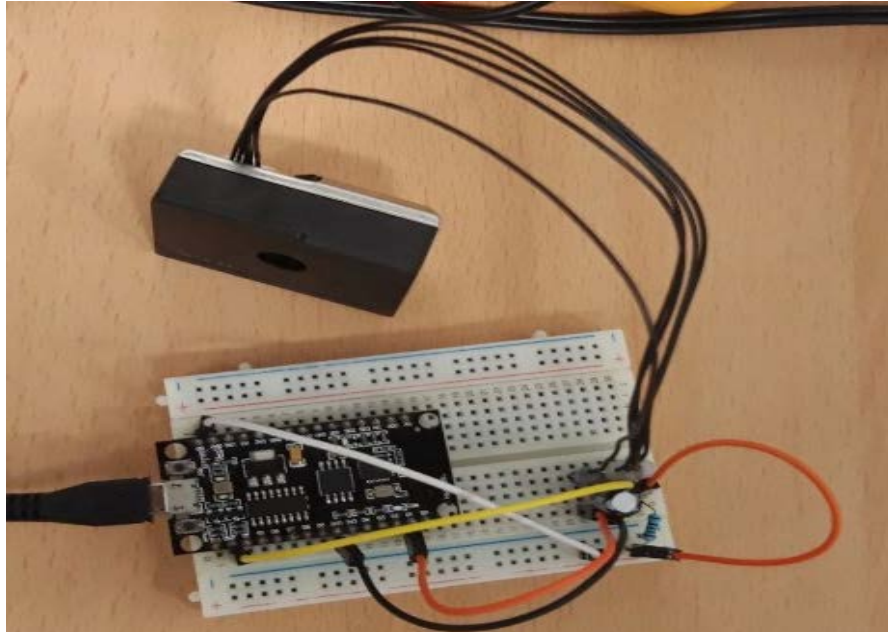
- 구동 회로 및 회로도



## 5.3 미세먼지 센서

### ◆ 5.3.1 미세먼지 센서 값 시리얼 프린터로 출력 - (MCU)

- 연결도



## 5.3 미세먼지 센서

### ◆ 5.3.1 미세먼지 센서 값 시리얼 프린터로 출력 - (MCU)

- 소스 코드

```
2 // 미세먼지 Control -> GP2Y_control
3
4 unsigned long pulse = 0;
5 float ugm3 = 0;
6
7 void setup()
8 {
9   pinMode(D1, INPUT);
10  Serial.begin(115200);
11 }
12
13 void loop()
14 {
15   pulse = pulseIn(D1, LOW, 20000);
16   ugm3 = pulse2ugm3(pulse);
17   Serial.print(ugm3, 4);
18   Serial.println("\t\tug/m3");
19   delay(100);
20 }
21
22 float pulse2ugm3(unsigned long pulse)
23 {
24   float value = (pulse-1400)/14.0;
25   if(value > 300) value = 0;
26   return value;
27 }
```

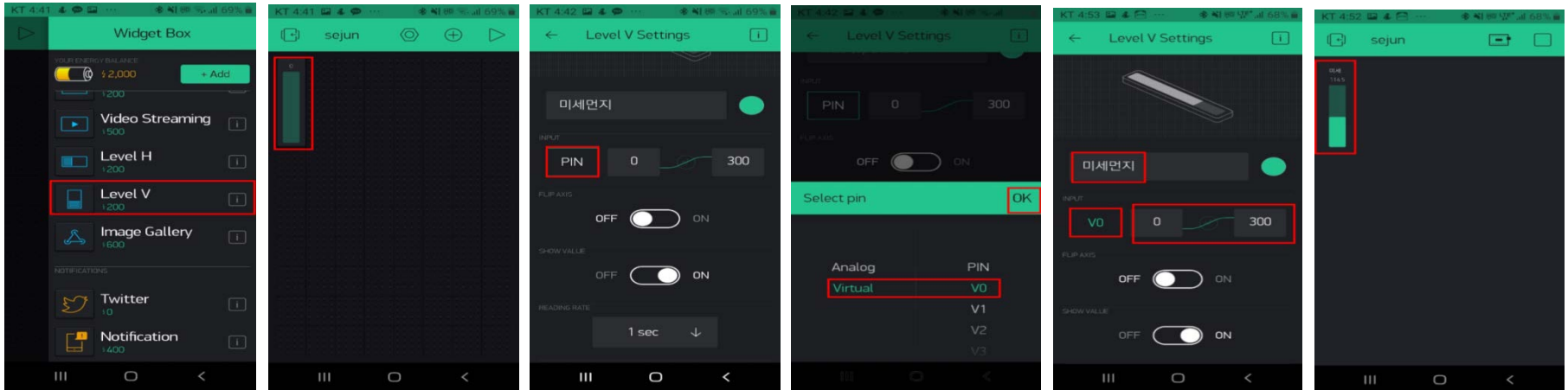
COM9

```
87.6429 ug/m3
120.7857 ug/m3
154.0714 ug/m3
242.2857 ug/m3
220.5000 ug/m3
251.7857 ug/m3
279.1429 ug/m3
0.0000 ug/m3
297.2143 ug/m3
297.1429 ug/m3
297.1429 ug/m3
297.1429 ug/m3
297.2857 ug/m3
297.1429 ug/m3
297.0714 ug/m3
```

# 5.3 미세먼지 센서

## ◆ 5.3.2 미세먼지 센서 값 Blynk LevelV로 출력- (MCU→폰)

### ● Widget 설정





## 5.3 미세먼지 센서

### ◆ 5.3.2 미세먼지 센서 값 Blynk LevelV로 출력- (MCU→폰)

#### ● 소스 코드

```
3 // 미세먼지 Control -> GP2Y_control_vpin(NodeMCU->스마트폰)
4
5 #include <ESP8266WiFi.h>
6 #include <BlynkSimpleEsp8266.h>
7 #define BLYNK_PRINT Serial
8
9 // You should get Auth Token in the Blynk App.
10 // Go to the Project Settings (nut icon).
11 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4";
12
13 char ssid[] = "sjpark";
14 char pass[] = "12345678";
15
16 unsigned long pulse = 0;
17 float ugm3 = 0;
18
19 void setup()
20 {
21   Blynk.begin(auth, ssid, pass);
22   pinMode(D1, INPUT);
23   Serial.begin(9600);
24 }
25
26 void loop()
27 {
28   Blynk.run();
29   pulse = pulseIn(D1, LOW, 20000);
30   ugm3 = pulse2ugm3(pulse);
31   Blynk.virtualWrite(V0, ugm3);
32   delay(100);
33 }
34
35 float pulse2ugm3(unsigned long pulse)
36 {
37   float value = (pulse-1400)/14.0;
38   if(value > 300) value = 0;
39   return value;
40 }
```

## 5.3 미세먼지 센서

### ◆ 실습문제(MCU → 폰)

- 미세먼지 센서를 연결하고 하드웨어를 구성하시오.
- Blynk Value Display(1)를 이용하여 미세먼지 센서 값 출력
- LED(3)를 이용하여 미세먼지 센서 값에 의한 LED 변화 출력
  - 30ug/m<sup>2</sup> 이하 – Green LED, 30ug/m<sup>2</sup> ~ 200ug/m<sup>2</sup> – Yellow LED
  - 200ug/m<sup>2</sup> 이상 – Red LED

# 다양한 센서 응용하기

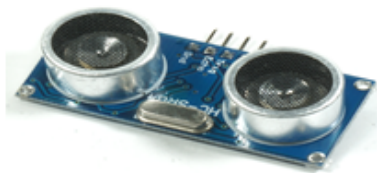
## ◆ 실습문제(MCU→폰)

- 온습도 센서, 인체감지 센서, 미세먼지 센서 각 센서 값을 Blynk로 모두 표시하시오.
- Blynk Controller는 각자 선택하여 출력하시오.

## 5.4 초음파 센서

### ◆ 초음파 거리센서(HC-SR04)

- 약 40Hz 주파수의 초음파 발사 물체에 반사되어 돌아오는 시간 측정
- 물체의 색깔에 상관없이 사용 가능
- 투명한 물체도 감지 가능, 물이나 먼지 등이 있더라도 감지 가능
- 외부에 초음파 발신부가 노출되어야 함

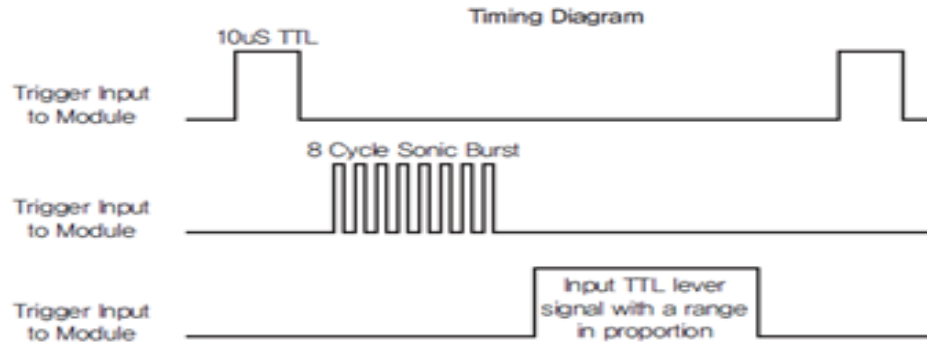


HC-SR04 초음파센서

## 5.4 초음파 센서

### ◆ HC-SR04 타이밍 다이어그램

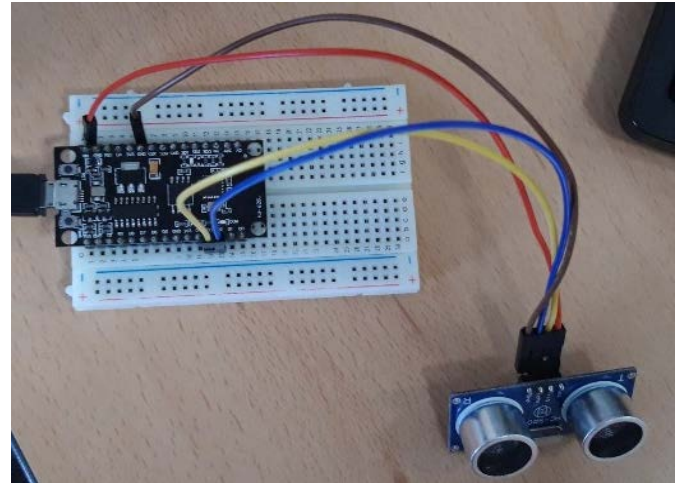
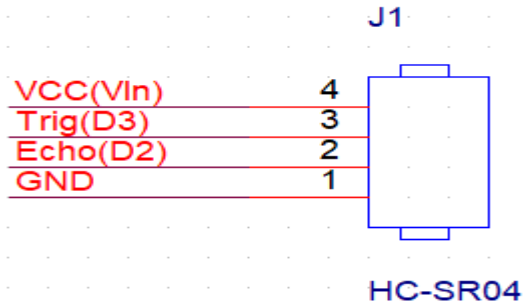
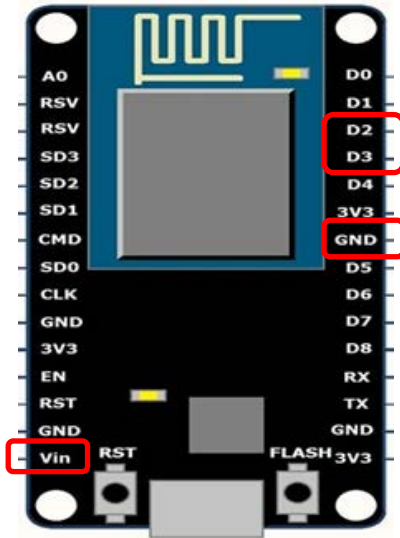
$$L = \frac{\text{에코 펄스 폭} * 340[m/s]}{2} \quad , L : \text{물체와의 거리}$$



## 5.4 초음파 센서

### ◆ 5.4.1 초음파 센서 값 시리얼 프린터로 출력 - (MCU)

- 회로도 및 연결도

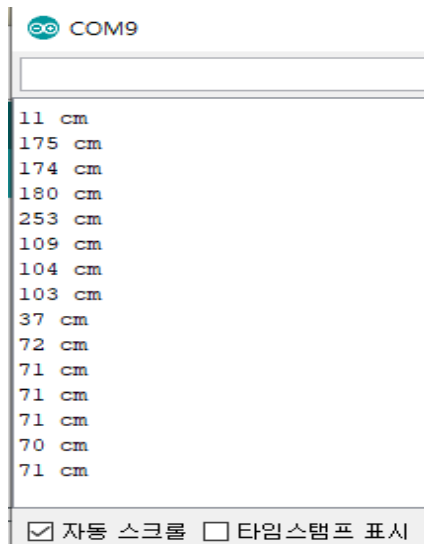


## 5.4 초음파 센서

### ◆ 5.4.1 초음파 센서 값 시리얼 프린터로 출력 - (MCU)

#### ● 소스 코드

```
2 // Ultrasonic Control
3
4 // 펄스 폭과 거리 변수 설정
5 int pulseWidth;
6 int distance;
7
8 void setup()
9 {
10   Serial.begin(9600);
11   pinMode(D3, OUTPUT);
12   pinMode(D2, INPUT);
13   digitalWrite(D3, 0); // 트리거 핀 초기값을 LOW로 한다
14 }
15
16 void loop()
17 {
18   // 10us의 트리거 신호를 HC-SR04로 내보낸다.
19   digitalWrite(D3, 1);
20   delayMicroseconds(10);
21   digitalWrite(D3, 0);
22
23   // Echo 펄스 폭을 측정하며 pulseWidth 변수에 저장한다.
24   pulseWidth = pulseIn(D2, 1);
25   distance = pulseWidth / 58; // 거리 계산
26
27   Serial.print(distance);
28   Serial.println(" cm");
29   delay(100);
30 }
```

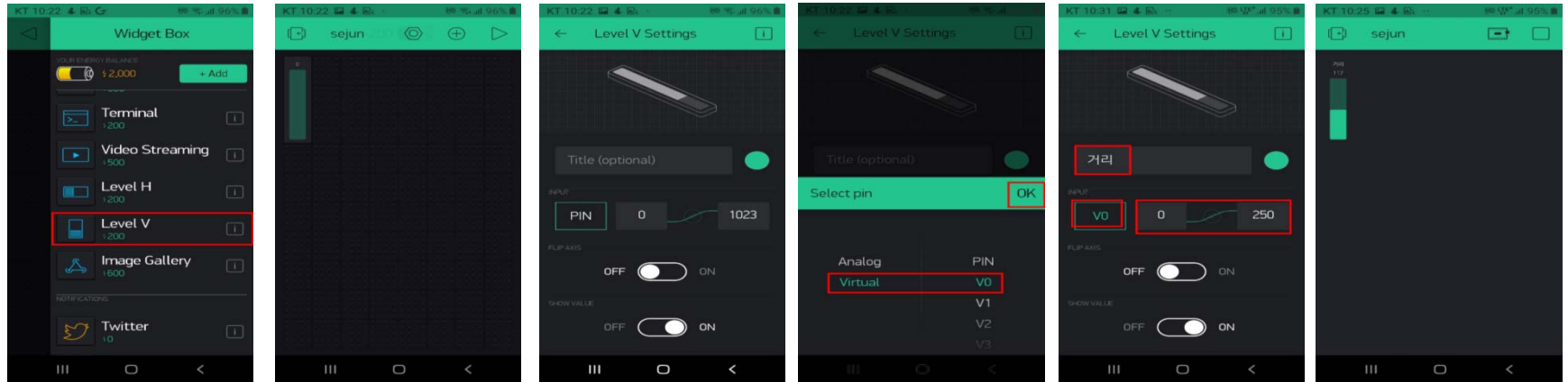


The screenshot shows a serial terminal window titled 'COM9'. The terminal displays a list of distance measurements in centimeters, corresponding to the code's output. The measurements are: 11 cm, 175 cm, 174 cm, 180 cm, 253 cm, 109 cm, 104 cm, 103 cm, 37 cm, 72 cm, 71 cm, 71 cm, 71 cm, 70 cm, and 71 cm. At the bottom of the window, there are two checkboxes: '자동 스크롤' (checked) and '타임스탬프 표시' (unchecked).

## 5.4 초음파 센서

### ◆ 5.4.2 초음파 센서 값 Blynk LevelV로 출력 - (MCU→폰)

- Widget 설정





## 5.4 초음파 센서

### ◆ 5.4.2 초음파 센서 값 Blynk LevelV로 출력 - (MCU→폰)

#### ● 소스 코드

```
2 // Ultrasonic Control Vpin
3
4 #define BLYNK_PRINT Serial
5 #include <ESP8266WiFi.h>
6 #include <BlynkSimpleEsp8266.h>
7
8 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4";
9
10 // Your WiFi credentials.
11 // Set password to "" for open networks.
12 char ssid[] = "sjpark";
13 char pass[] = "12345678";
14
15 // 펄스 폭과 거리 변수 설정
16 int pulseWidth;
17 int distance;
18
19 void setup()
20 {
21   Serial.begin (9600);
22   pinMode (D3, OUTPUT);
23   pinMode (D2, INPUT);
24   Blynk.begin(auth, ssid, pass);
25   digitalWrite (D3, 0); // 트리거 핀 초기값을 LOW로 한다
26 }
```

```
28 void loop()
29 {
30   Blynk.run();
31   // 10us의 트리거 신호를 HC-SR04로 내보낸다.
32   digitalWrite (D3, 1);
33   delayMicroseconds (10);
34   digitalWrite (D3, 0);
35
36   // Echo 펄스 폭을 측정하여 pulseWidth 변수에 저장한다.
37   pulseWidth = pulseIn (D2, 1);
38   distance = pulseWidth / 58; //((float) (340 * du
39
40   Blynk.virtualWrite (V0, distance);
41   delay (100);
42 }
```

## 5.4 초음파 센서

### ◆ 실습문제(MCU→폰, MCU)

- 초음파 센서를 이용하여 DC모터를 제어하시오.
  - 50Cm 미만 - 정지(Blynk LED - RED)
  - 50Cm ~ 100Cm - CCW(Blynk LED - YELLOW)
  - 101Cm 이상 - CW(Blynk LED - GREEN)