

Chap02. 출력장치 제어

Contents

2.1 LED 제어하기

2.2 RGBLED 제어하기

2.3 LCD 제어하기

2.4 Buzzer 제어하기

2.1 LED 제어하기

◆ LED(Light Emitting Diode)

- 전기 신호를 빛으로 출력하는 반도체 소자
- 고효율, 반영구적 수명
- 가정용 실내등, 산업용 특수등, 자동차용 전조등 및 실내등



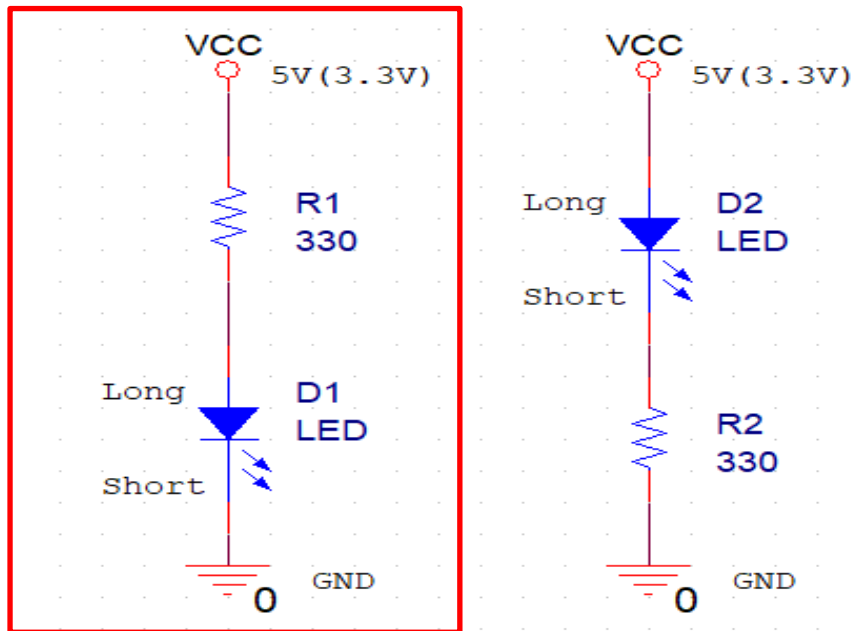
실험에 사용할 LED와 기호



다양한 형태의 LED

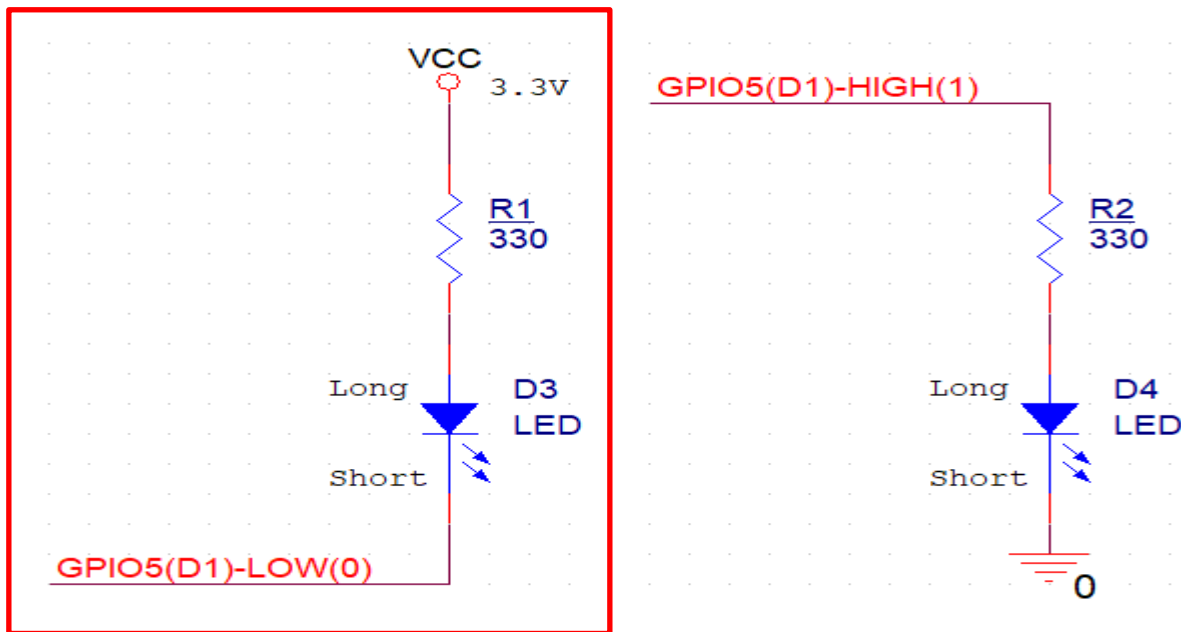
2.1 LED 제어하기

◆ 일반적인 전자회로 LED 켜기(On) 회로도



2.1 LED 제어하기

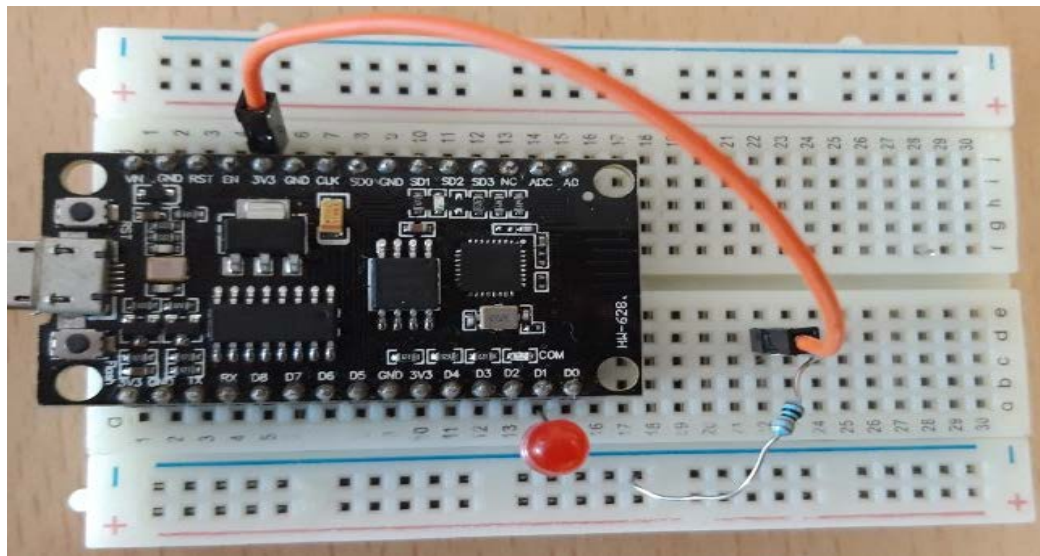
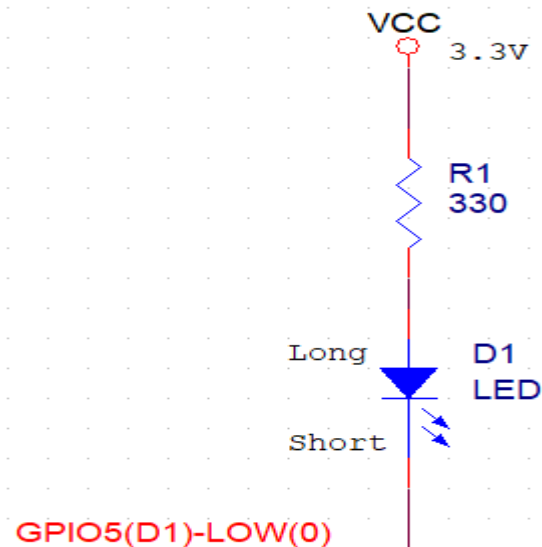
◆ NodeMCU를 이용한 LED 켜기(On) 회로도



2.1 LED 제어하기

◆ 2.1.1 LED 깜박이기(점멸)

● 회로도 및 연결도



2.1 LED 제어하기

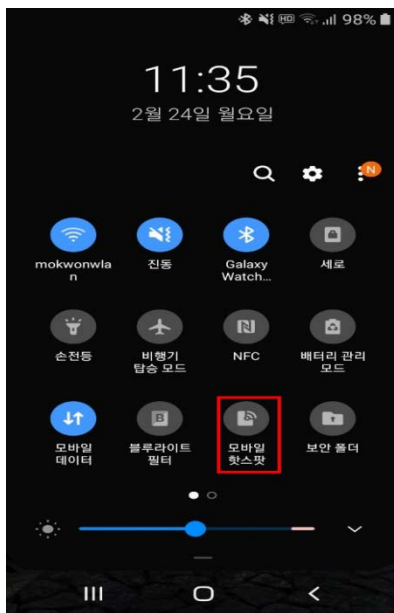
◆ 2.1.1 LED 깜박이기(점멸)

- 소스 코드

```
2 //LED 깜박이기(점멸) -> led_on_off
3
4 // only first time
5 void setup() {
6     pinMode(D1, OUTPUT);    // (D1)Output Mode(5:GPIO5)
7 }
8
9 // the loop function runs over and over again forever
10 void loop() {
11     digitalWrite(D1, 0);    // (D1)turn the LED on(5:GPIO5)
12     delay(1000);            // wait for a second
13     digitalWrite(D1, 1);    // (D1)turn the LED off(5:GPIO5)
14     delay(1000);
15 }
```

2.1 LED 제어하기

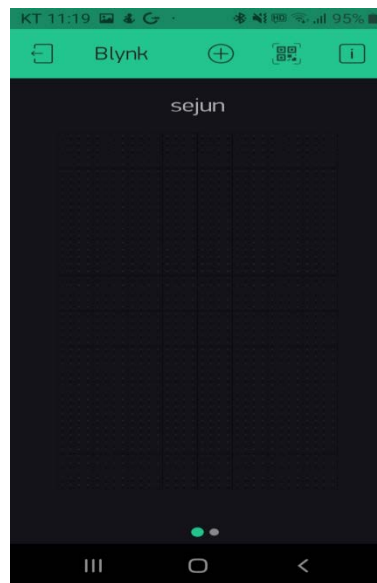
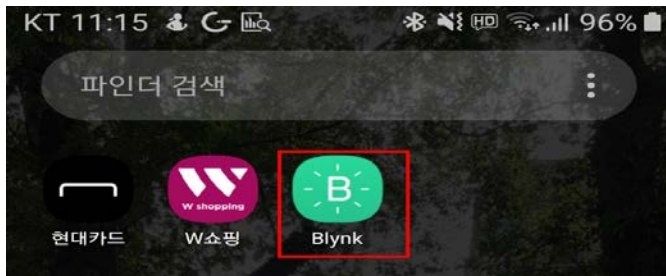
◆ 모바일 핫스팟 설정



2.1 LED 제어하기

◆ 2.1.2 Blynk LED On/Off 제어1(가상핀) - (폰→MCU)

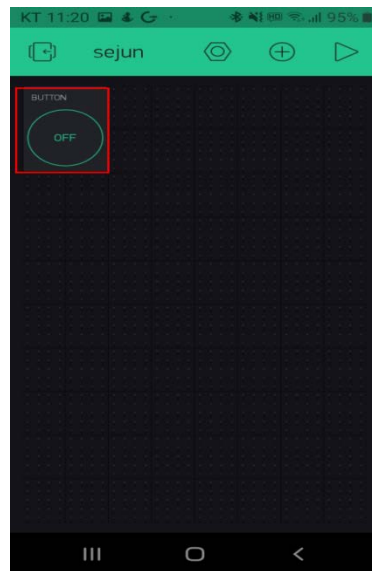
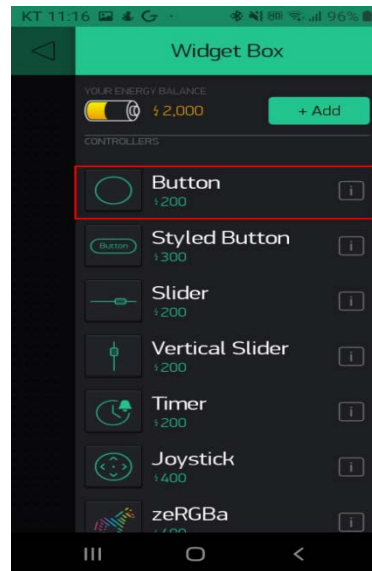
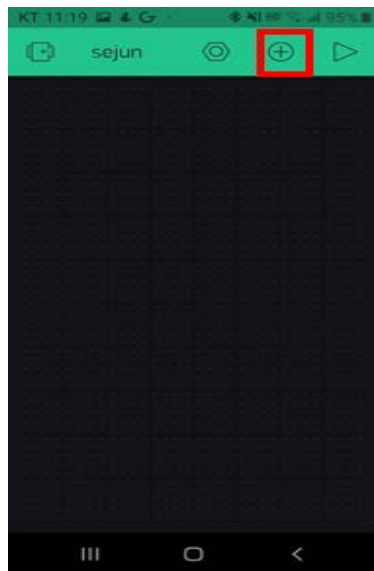
- Blynk App 실행



2.1 LED 제어하기

◆ 2.1.2 Blynk LED On/Off 제어1(가상핀) - (폰→MCU)

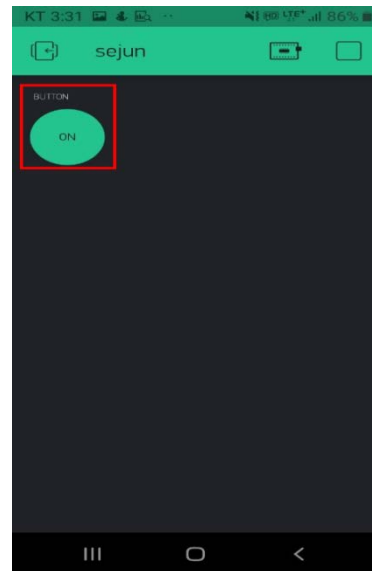
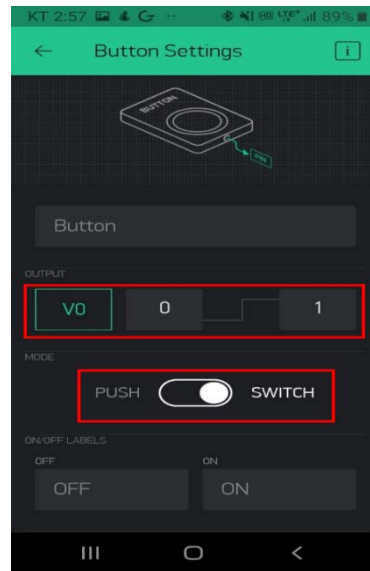
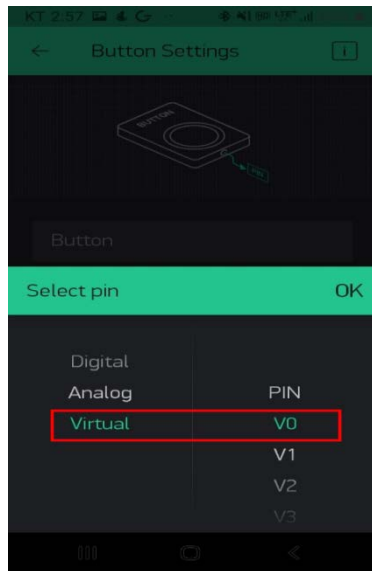
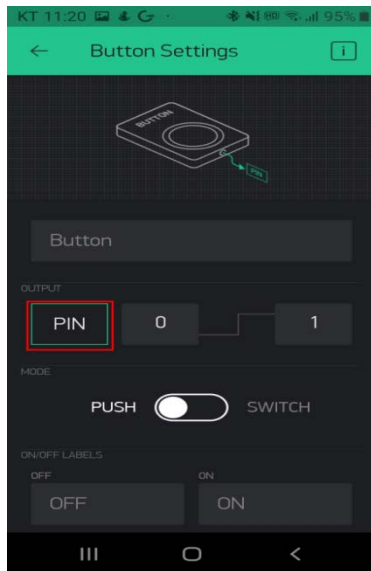
- 화면 클릭으로 widget 추가



2.1 LED 제어하기

◆ 2.1.2 Blynk LED On/Off 제어1(가상핀) - (폰→MCU)

- Widget 설정



2.1 LED 제어하기

◆ 2.1.2 Blynk LED On/Off 제어1(가상핀) - (폰→MCU)

- 소스 코드

```
-
2 //LED ON/OFF CONTROL -> led_on_off_vpin(스마트폰 -> NodeMCU)
3
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #define BLYNK_PRINT Serial
7
8 // You should get Auth Token in the Blynk App.
9 // Go to the Project Settings (nut icon).
10 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4"; // 이메일 토큰으로 변경
11
12 // Your WiFi credentials.
13 // Set password to "" for open networks.
14 char ssid[] = "sjpark";
15 char pass[] = "12345678";
16
17 int pinData=0;
```

2.1 LED 제어하기

◆ 2.1.2 Blynk LED On/Off 제어1(가상핀) - (폰→MCU)

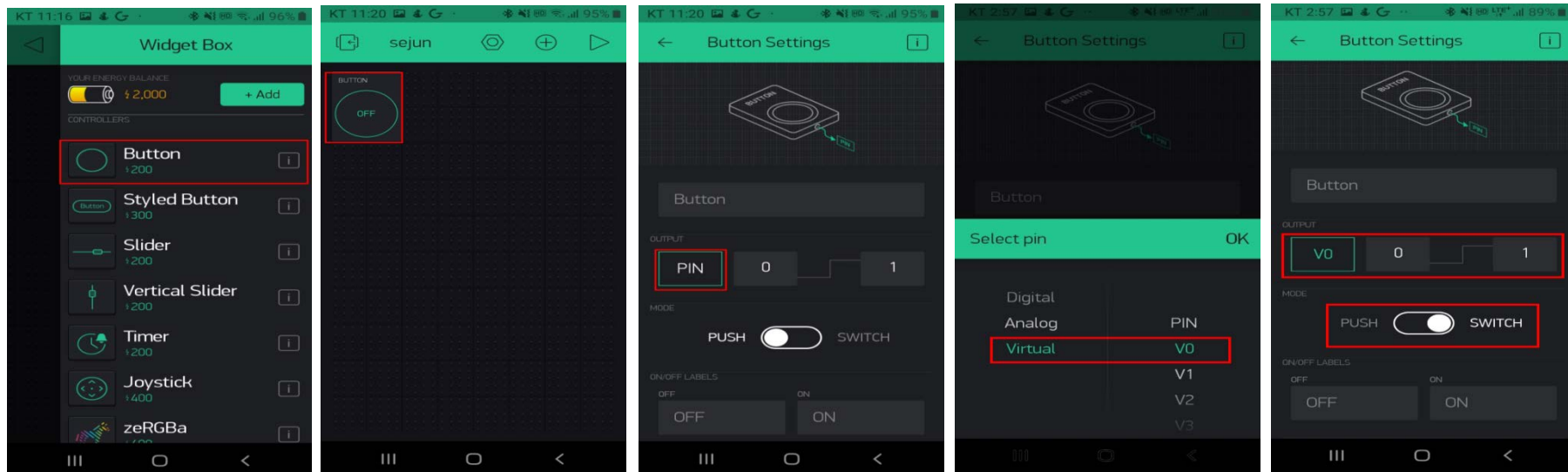
- 소스 코드

```
17 void setup()
18 {
19   Blynk.begin(auth, ssid, pass);
20   pinMode(D1, OUTPUT);
21 }
22
23 BLYNK_WRITE(V0)
24 {
25   pinData = param.asInt();    //매개변수로 들어옴
26 }
27
28 void loop()
29 {
30   Blynk.run();
31   if(pinData==1)digitalWrite(D1,0); // LED ON
32   else digitalWrite(D1,1);         // LED OFF
33 }
```

2.1 LED 제어하기

◆ 2.1.3 Blynk LED On/Off 제어2(가상핀) - (폰→폰, MCU)

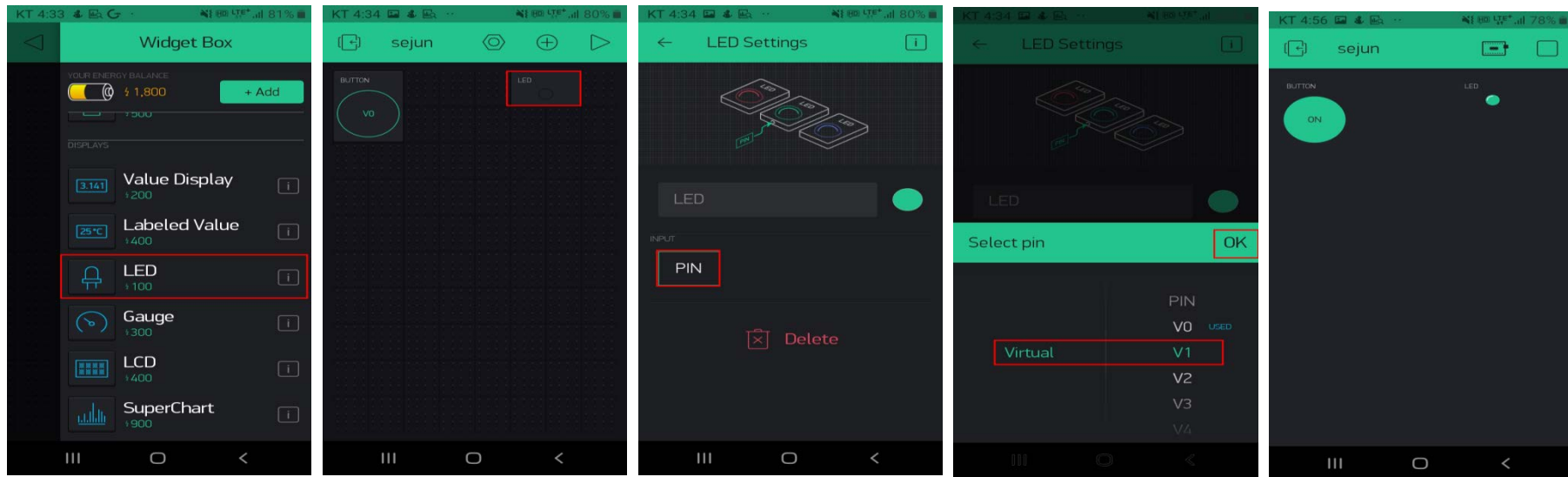
● Widget 설정



2.1 LED 제어하기

◆ 2.1.3 Blynk LED On/Off 제어2(가상핀) - (폰→폰, MCU)

● Widget 설정



2.1 LED 제어하기

◆ 2.1.3 Blynk LED On/Off 제어2(가상핀) - (폰→폰, MCU)

● 소스 코드

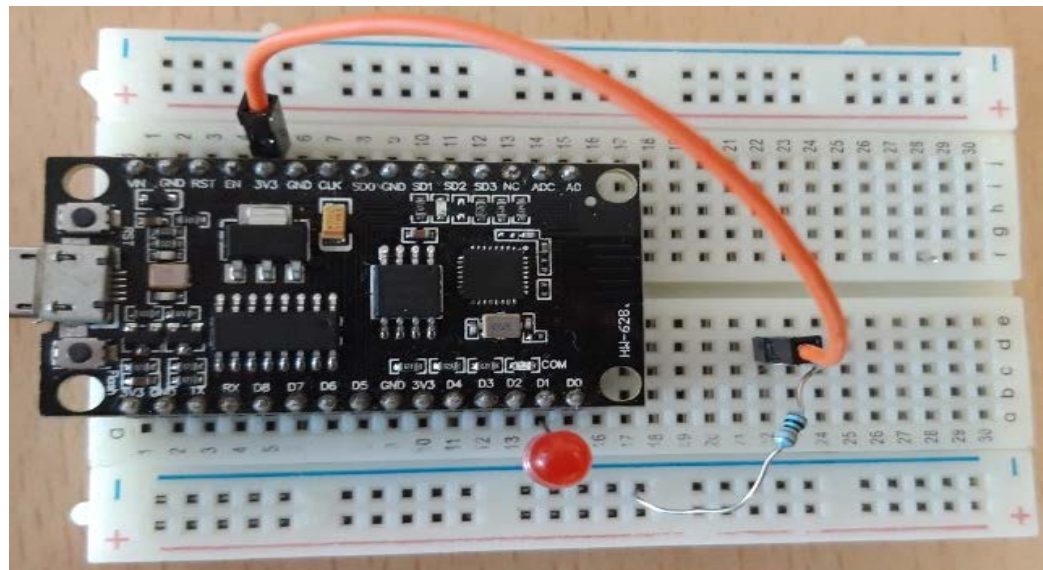
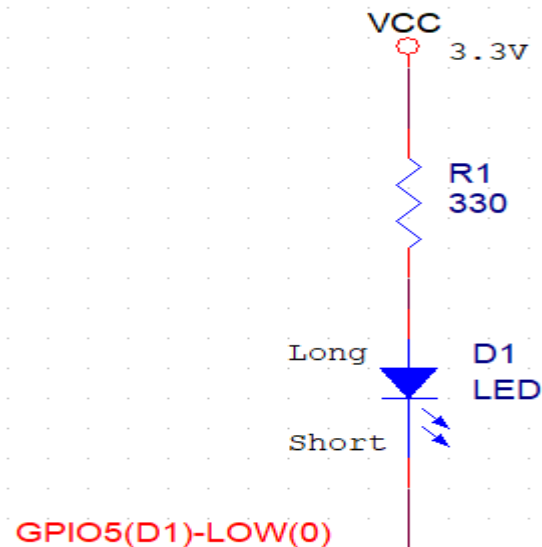
```
2 //LED ON/OFF CONTROL -> led_on_off_vpin1 (스마트폰->폰, NodeMCU)
3
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #define BLYNK_PRINT Serial
7
8 // You should get Auth Token in the Blynk App.
9 // Go to the Project Settings (nut icon).
10 char auth[] = "KWIEYYjACwL_4t3z-5wNjHSxw3Uftqy4"; // 이메일 토큰으
11
12 // Your WiFi credentials.
13 // Set password to "" for open networks.
14 char ssid[] = "sjpark";
15 char pass[] = "12345678";
16
17 int pinData=0;
18
19 WidgetLED led(V1);
```

```
21 void setup()
22 {
23   Blynk.begin(auth, ssid, pass);
24   pinMode(D1, OUTPUT);
25 }
26
27 BLYNK_WRITE(V0)
28 {
29   pinData = param.asInt(); // 매개변수로 들어옴
30 }
31
32 void loop()
33 {
34   Blynk.run();
35   if (pinData==1)
36   {
37     led.on();
38     digitalWrite(D1, 0); // LED ON
39   }
40   else
41   {
42     digitalWrite(D1, 1); // LED OFF
43     led.off();
44   }
45 }
```


2.1 LED 제어하기

◆ 2.1.4 LED 밝기 조절(dimming)

- 회로도 및 연결도



2.1 LED 제어하기

◆ 2.1.4 LED 밝기 조절(dimming)

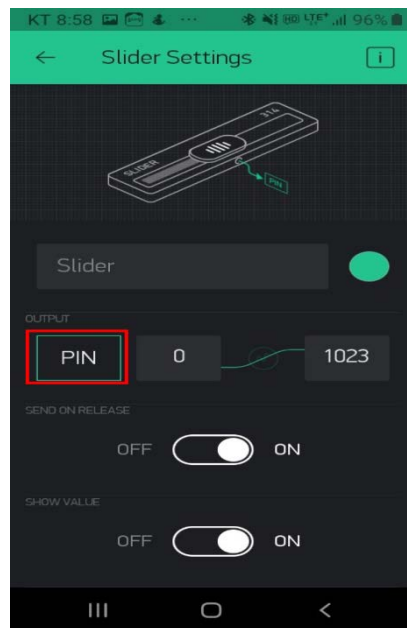
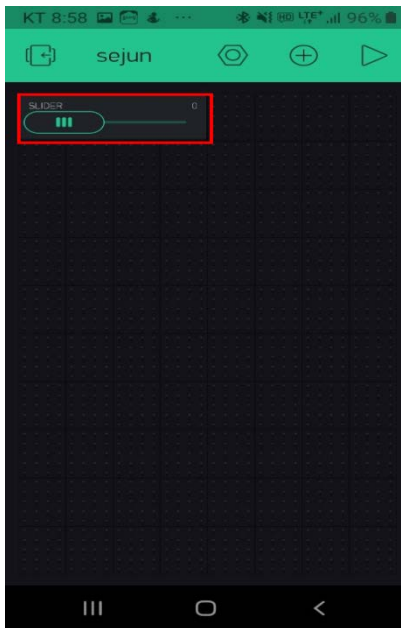
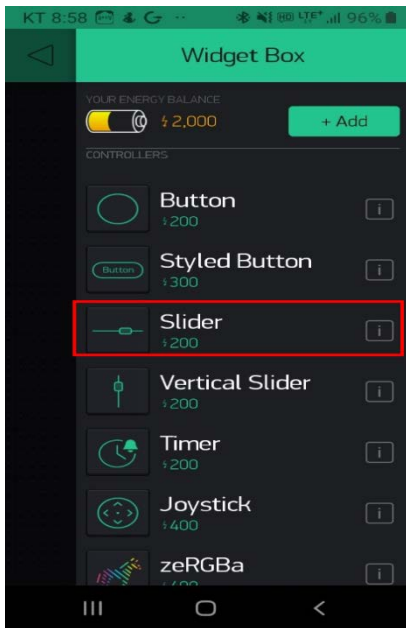
- 소스 코드

```
2 //LED dimming(밝기조절) -> led_dimming
3
4 int i;
5
6 void setup()
7 {
8   pinMode(D1, OUTPUT);      //(D1)Output Mode (5:GPIO15)
9 }
10
11 void loop()
12 {
13   for(i=0; i<=1024; i++)
14   {
15     analogWrite(D1,i);      // LED 밝기 조절
16     delay(5);               // 0.005 초간 지연
17   }
18   for(; i>=0; i--)
19   {
20     analogWrite(D1,i);
21     delay(5);
22   }
23 }
```

2.1 LED 제어하기

◆ 2.1.5 Blynk LED 밝기 조절 - 가상핀(폰 → MCU)

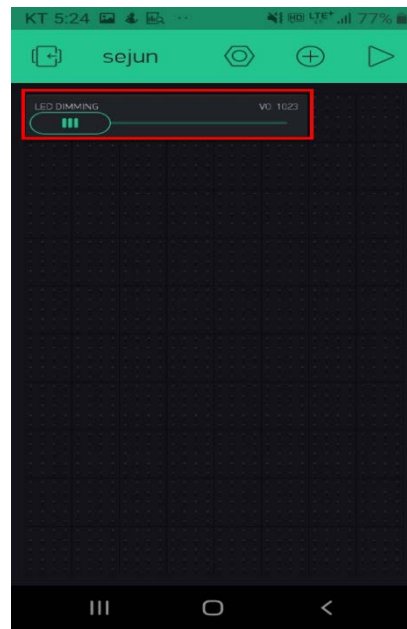
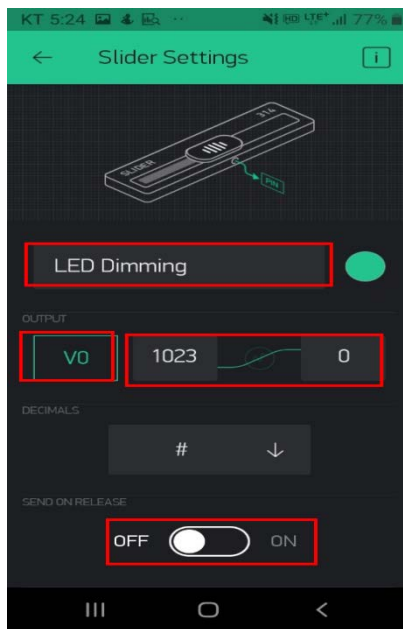
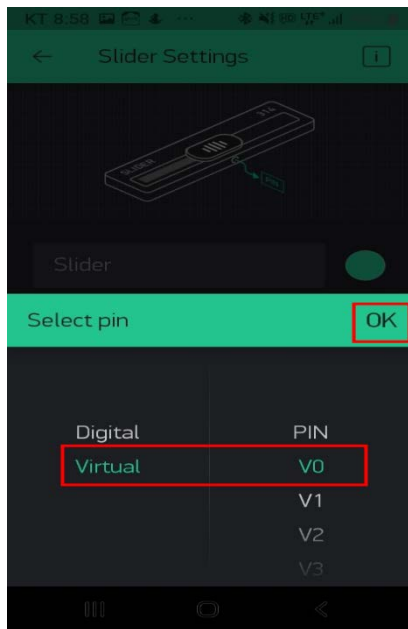
- Widget 설정



2.1 LED 제어하기

◆ 2.1.5 Blynk LED 밝기 조절 - 가상핀(폰 → MCU)

- Widget 설정



2.1 LED 제어하기

◆ 2.1.5 Blynk LED 밝기 조절 - 가상핀(폰 → MCU)

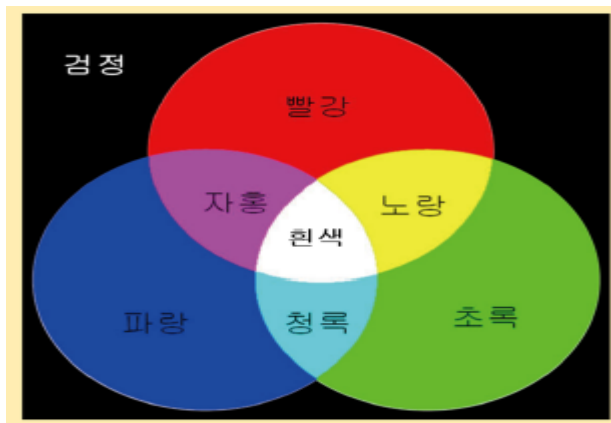
● 소스 코드

```
2 //LED dimming(밝기조절) -> led_dimming_vpin(스마트폰 -> NodeMCU)
3
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #define BLYNK_PRINT Serial
7
8 // You should get Auth Token in the Blynk App.
9 // Go to the Project Settings (nut icon).
10 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4"; // 이메일 토
11
12 // Your WiFi credentials.
13 // Set password to "" for open networks.
14 char ssid[] = "sjpark";
15 char pass[] = "12345678";
16
17 int pinData=0;
18 void setup()
19 {
20     Serial.begin(9600);
21     Blynk.begin(auth, ssid, pass);
22     pinMode(D1,OUTPUT);
23 }
24
25 BLYNK_WRITE(V0)
26 {
27     pinData = param.asInt(); //매개변수로 들어옴
28 }
29
30 void loop()
31 {
32     Blynk.run();
33     analogWrite(D1,pinData);
34 }
```

2.2 RGB LED 제어하기

◆ 빛의 삼원색

- 빛의 삼원색은 각각 Red, Green, Blue 의미
- 3가지 색(0~255)의 조합을 통하여 다양한 색 표현 가능



2.2 RGB LED 제어하기

◆ RGB LED

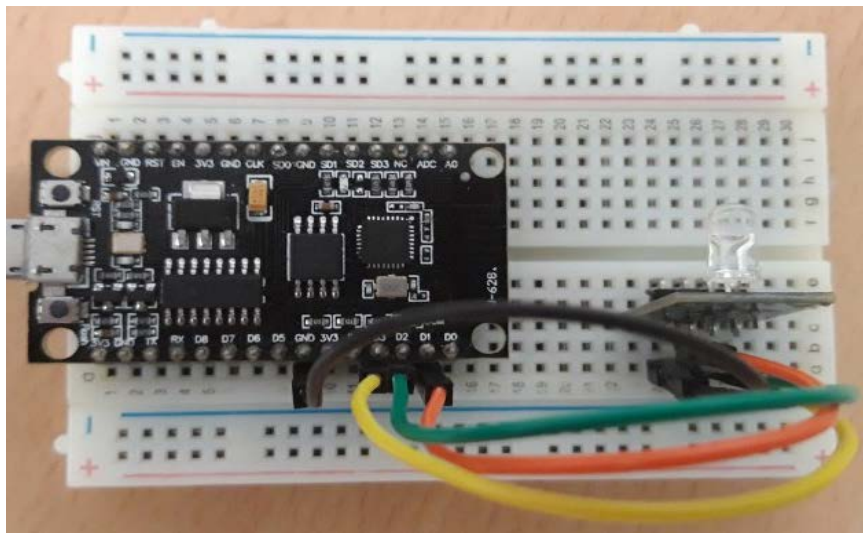
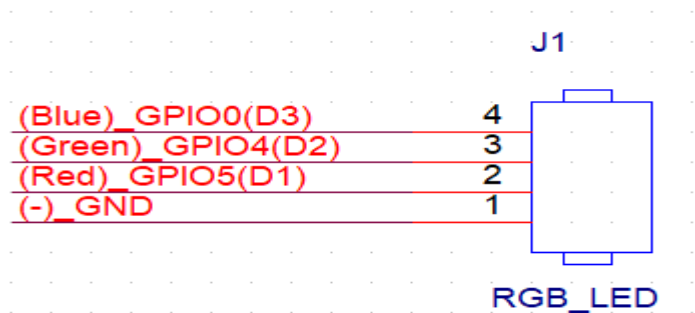
- 빛의 삼원색을 조절하여 다양한 색을 표현하는 LED
 - 빨강(Red), 초록(Green), 파랑(Blue)
- 각각의 색이 0~255단계로 조절
- 간판, 조명기구 등에 사용
- 모든 색이 출력될 때 백색 빛 출력



2.2 RGBLED 제어하기

◆ 2.2.1 RGBLED 깜박이기(점멸)

- 회로도 및 연결도



2.2 RGBLED 제어하기

◆ 2.2.1 RGBLED 깜박이기(점멸)

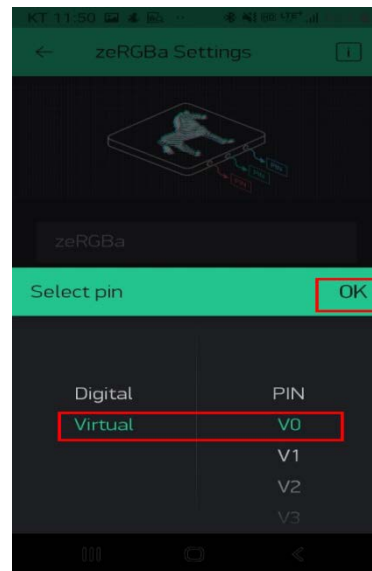
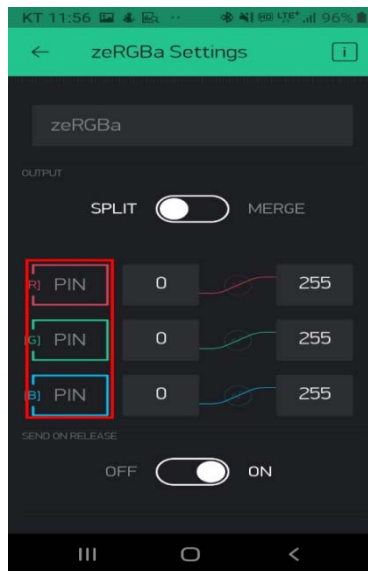
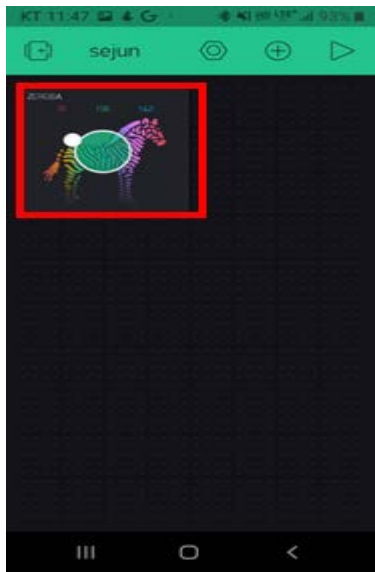
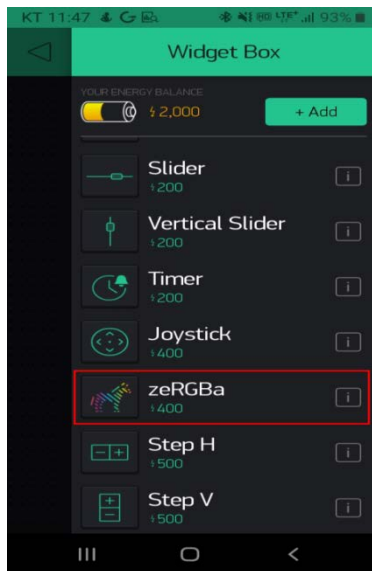
● 소스 코드

```
2 // RGBLed on/off 깜박이기 -> RGBled_on_off
3
4 const int RedLed=D1;    //R=D1 (GPIO5)
5 const int GreenLed=D2;  //G=D2 (GPIO4)
6 const int BlueLed=D3;   //B=D3 (GPIO0)
7
8 void setup()
9 {
10     pinMode(RedLed, OUTPUT);
11     pinMode(GreenLed, OUTPUT);
12     pinMode(BlueLed, OUTPUT);
13
14     analogWrite(RedLed,0);
15     analogWrite(GreenLed,0);
16     analogWrite(BlueLed,0);
17 }
18
19 void RGBLed_Output(int r, int g, int b)
20 {
21     analogWrite(RedLed,r);
22     analogWrite(GreenLed,g);
23     analogWrite(BlueLed,b);
24 }
25
26 void loop()
27 {
28     RGBLed_Output(1023, 1023, 1023);
29     delay(1000);
30     RGBLed_Output(0, 0, 0);
31     delay(1000);
32 }
```

2.2 RGBLED 제어하기

◆ 2.2.2 Blynk RGBLED 제어 – 가상핀(split)

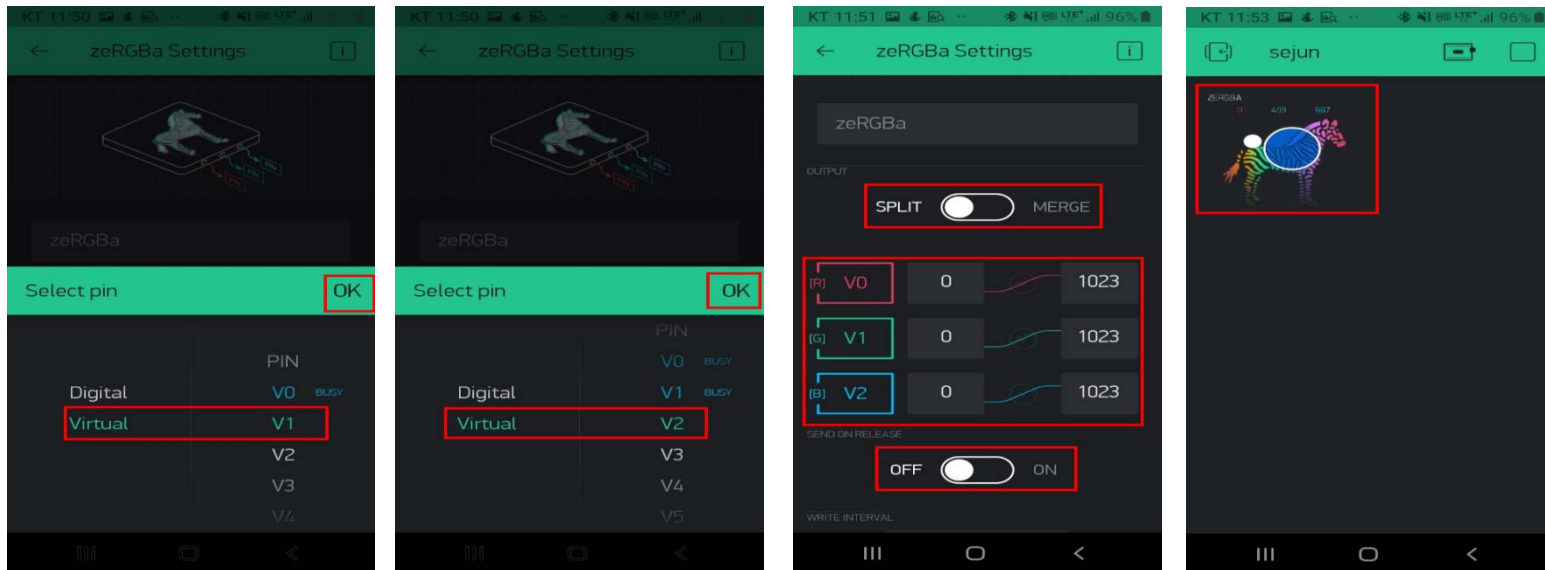
● Widget 설정



2.2 RGBLED 제어하기

◆ 2.2.2 Blynk RGBLED 제어 – 가상핀(split)

● Widget 설정



2.2 RGBLED 제어하기

◆ 2.2.2 Blynk RGBLED 제어 - 가상핀(split)

● 소스 코드

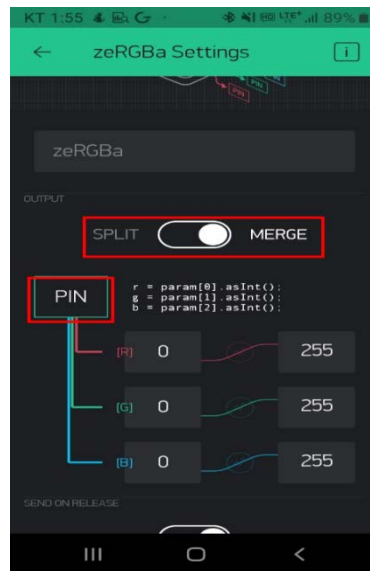
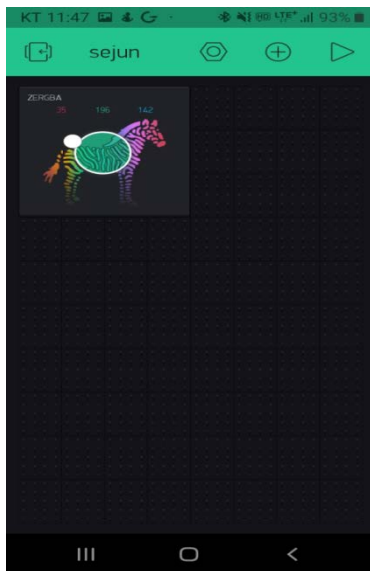
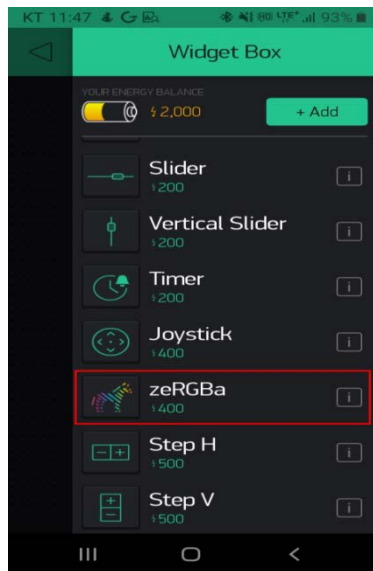
```
2 // RGBLed Control(split) - RGBled_control_vpin_split (스마트폰->NodeMCU)
3
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #define BLYNK_PRINT Serial
7
8 // You should get Auth Token in the Blynk App.
9 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4"; // 이메일 토큰으로 변경
10 char ssid[] = "sjpark";
11 char pass[] = "12345678";
12
13 int r=0;
14 int g=0;
15 int b=0;
```

```
17 void setup()
18 {
19   // Debug console
20   Serial.begin(9600);
21   Blynk.begin(auth, ssid, pass);
22 }
23
24 BLYNK_WRITE(V0) {
25   r=param.asInt();
26 }
27
28 BLYNK_WRITE(V1) {
29   g=param.asInt();
30 }
31
32 BLYNK_WRITE(V2) {
33   b=param.asInt();
34 }
35 void loop()
36 {
37   Blynk.run();
38   analogWrite(D1,r);
39   analogWrite(D2,g);
40   analogWrite(D3,b);
41 }
```

2.2 RGBLED 제어하기

◆ 2.2.3 Blynk RGBLED 제어 – 가상핀(merge)

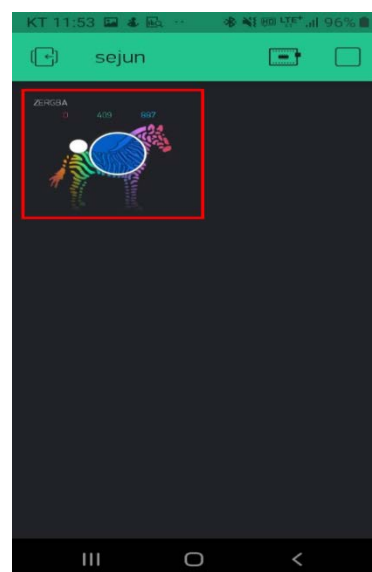
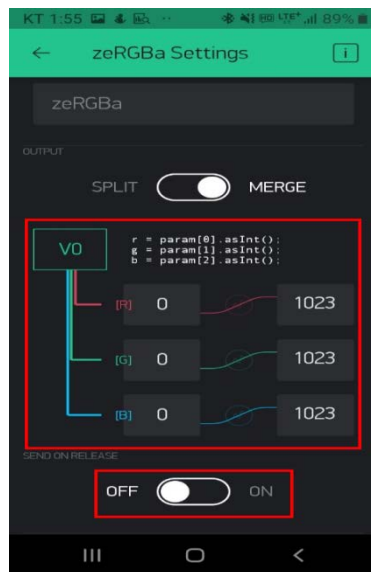
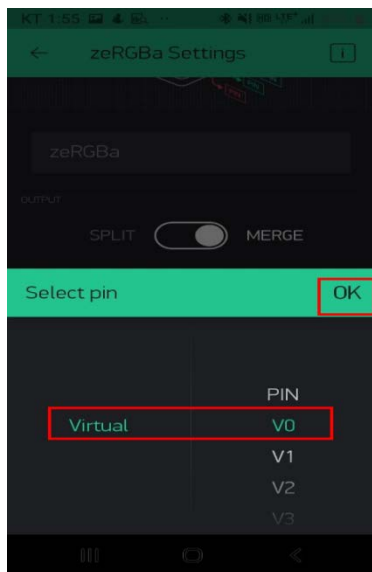
- Widget 설정



2.2 RGBLED 제어하기

◆ 2.2.3 Blynk RGBLED 제어 – 가상핀(merge)

- Widget 설정



2.2 RGBLED 제어하기

◆ 2.2.3 Blynk RGBLED 제어 - 가상핀(merge)

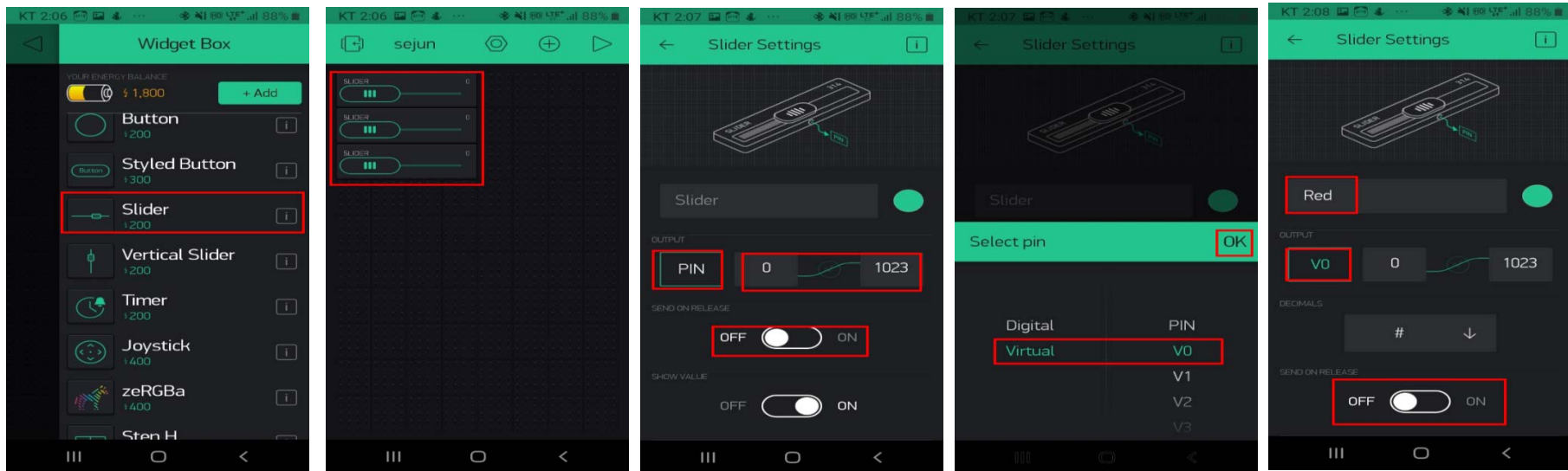
● 소스 코드

```
2 // RGBLed Control(merge) - RGBled_control_vpin_merge (스마트폰->NodeMCU)
3
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #define BLYNK_PRINT Serial
7
8 // You should get Auth Token in the Blynk App.
9 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4"; //
10 char ssid[] = "sjpark";
11 char pass[] = "12345678";
12
13 int r=0;
14 int g=0;
15 int b=0;
16
17 void setup()
18 {
19   Serial.begin(9600);
20   Blynk.begin(auth, ssid, pass);
21 }
22
23 BLYNK_WRITE(V0) {
24   r=param[0].asInt();
25   g=param[1].asInt();
26   b=param[2].asInt();
27 }
28
29 void loop()
30 {
31   Blynk.run();
32   analogWrite(D1,r);
33   analogWrite(D2,g);
34   analogWrite(D3,b);
35 }
```

2.2 RGBLED 제어하기

◆ 2.2.4 Blynk RGBLED 제어 – 가상핀(RGB BAR)

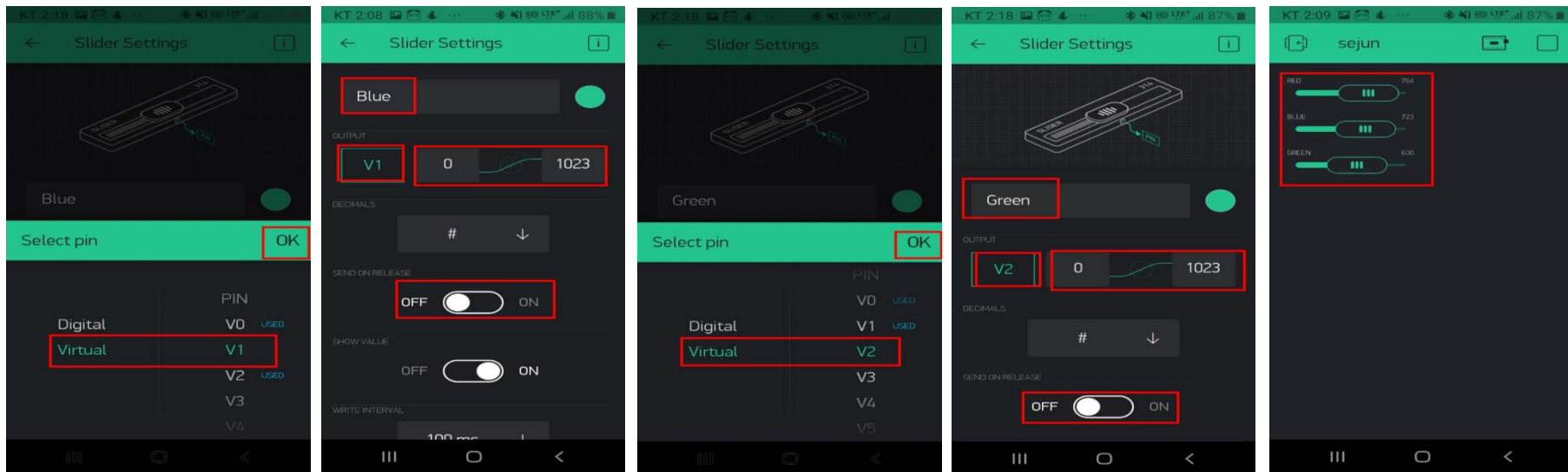
● Widget 설정



2.2 RGBLED 제어하기

◆ 2.2.4 Blynk RGBLED 제어 – 가상핀(RGB BAR)

- Widget 설정



2.2 RGBLED 제어하기

◆ 2.2.4 Blynk RGBLED 제어 - 가상핀(RGB BAR)

● 소스 코드

```
2 // RGBLed Control(RGB) - RGBled_control_vpin_RGB(스마트폰->NodeMCU)
3
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #define BLYNK_PRINT Serial
7
8 // You should get Auth Token in the Blynk App.
9 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4";
10 char ssid[] = "sjpark";
11 char pass[] = "12345678";
12
13 int r=0;
14 int g=0;
15 int b=0;
16
17 void setup()
18 {
19   Serial.begin(9600);
20   Blynk.begin(auth, ssid, pass);
21 }
22
23 BLYNK_WRITE(V0) {
24   r=param.asInt();
25 }
26
27 BLYNK_WRITE(V1) {
28   g=param.asInt();
29 }
30
31 BLYNK_WRITE(V2) {
32   b=param.asInt();
33 }
34
35 void loop()
36 {
37   Blynk.run();
38   analogWrite(D1,r);
39   analogWrite(D2,g);
40   analogWrite(D3,b);
41 }
```

2.3 LCD 제어하기

◆ LCD(Liquid Crystal Display)

- 얇은 액정판 아래 조명을 비추는 장치
- 액정판의 전류 흐름을 제어하여 문자나 그림 표시



2.3 LCD 제어하기

◆ I2C(Inter Integrated Circuit)

- Phillips사에서 개발된 규격(TWI – Two Wire Interface)
- SDA(Serial Data line), SCL(Serial Clock Line) 두 선으로 통신
- Master와 Slave로 구분(Master가 통신 주관)
- PCF8574 IC 사용(LCD 모듈 I²C 통신 제어)

2.3 LCD 제어하기

◆ LCD 관련 함수

- LiquidCrystal_I2C
 - lcd(I²C 주소, en, rw, rs, d4, d5, d6, d7, 백라이트, 백라이트 제어)
 - lcd란 이름으로 I²C에 연결된 LCD 모듈 설정
- lcd.begin(행, 열)
 - lcd란 이름의 LCD 모듈 크기 정의

2.3 LCD 제어하기

◆ LCD 관련 함수

- lcd.clear()
 - lcd란 이름의 LCD 모듈 화면의 모든 표시 지움(커서 왼쪽 위로 이동)
- lcd.home()
 - lcd란 이름의 LCD 모듈 커서 왼쪽 위로 이동
- lcd.setCursor(행, 열)
 - lcd란 이름의 LCD 모듈의 커서 원하는 위치로 이동

2.3 LCD 제어하기

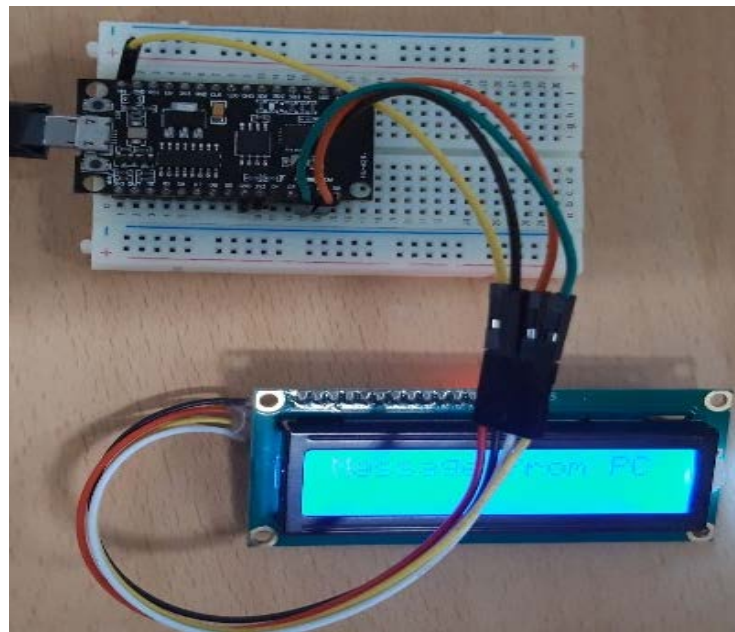
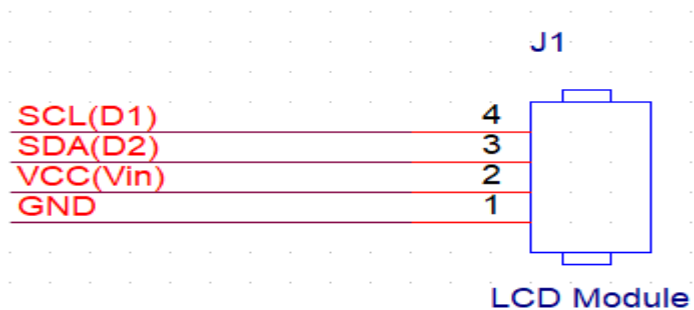
◆ LCD 관련 함수

- lcd.print(데이터)
 - lcd란 이름의 LCD 모듈에 데이터 출력
- lcd.noBacklight();
 - lcd란 이름의 LCD 모듈 백라이트 소등
- lcd.backlight();
 - lcd란 이름의 LCD 모듈 백라이트 점등

2.3 LCD 제어하기

◆ 2.3.1 LCD 제어하기(텍스트 표시)

- 회로도 및 연결도



2.3 LCD 제어하기

◆ 2.3.1 LCD 제어하기(텍스트 표시)

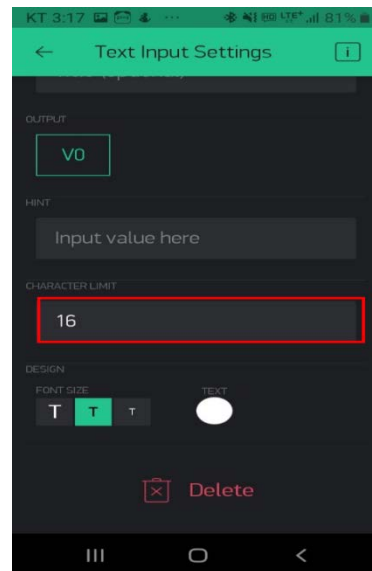
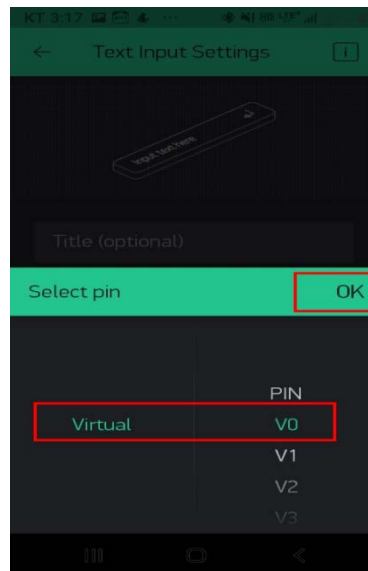
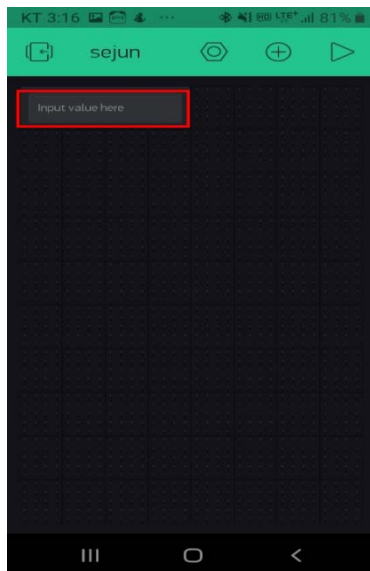
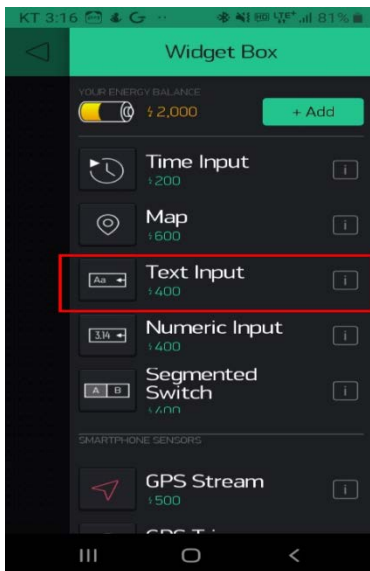
- 소스 코드

```
2 // LCD Control) - lcd_control
3
4 // I2C 통신 라이브러리 설정
5 #include <Wire.h>
6 // I2C LCD 라이브러리 설정
7 #include <LiquidCrystal_I2C.h>
8
9 // LCD I2C address 설정 PCF8574:0x27, PCF8574A:0x3F
10 LiquidCrystal_I2C lcd(0x20, 16, 2);
11
12 void setup()
13 {
14     Serial.begin(9600);
15     lcd.init();           // LCD 초기화
16     lcd.backlight();      // LCD 백라이트 on
17 }
18
19 void loop()
20 {
21     lcd.setCursor(0,0);
22     lcd.print("Embeded IoT Exp.");
23 }
```

2.3 LCD 제어하기

◆ 2.3.2 Blynk LCD 제어(가상핀)

- Widget 설정(글자 입력 후 보내기 버튼 누르기)



2.3 LCD 제어하기

◆ 2.3.2 Blynk LCD 제어(가상핀)

● 소스 코드

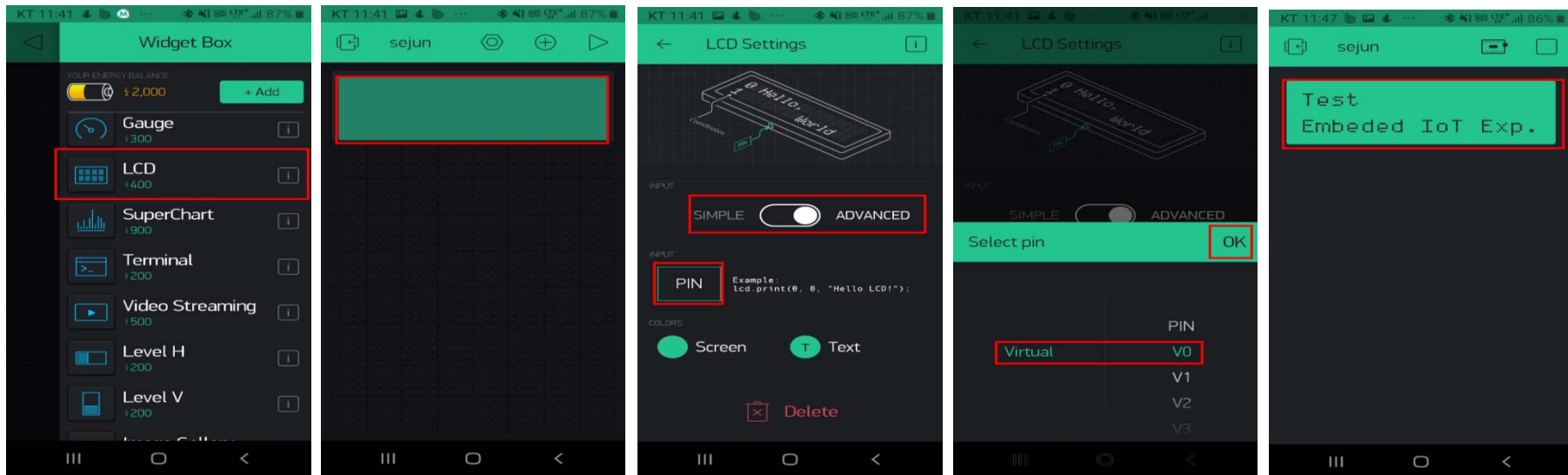
```
1 // LCD Control(Text)_vpin
2 // LCD Control(Text)_vpin
3 #define BLYNK_PRINT Serial
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 // I2C LCD 라이브러리 설정
7 #include <LiquidCrystal_I2C.h>
8
9 // I2C 통신 라이브러리 설정
10 #include <Wire.h>
11 // LCD I2C address 설정 PCF8574:0x27, PCF8574A:0x3F
12 LiquidCrystal_I2C lcd(0x20, 16, 2);
13
14 // You should get Auth Token in the Blynk App.
15 // Go to the Project Settings (nut icon).
16 char auth[] = "KWIEYYjACwL_4t3z-5wNjHSxw3Uftqy4"; //
17
18 // Your WiFi credentials.
19 // Set password to "" for open networks.
20 char ssid[] = "sjpark";
21 char pass[] = "12345678";
22
23 String strInput="";
```

```
25 void setup()
26 {
27     Serial.begin(9600);
28     Blynk.begin(auth, ssid, pass);
29
30     lcd.init();
31     lcd.backlight();
32 }
33
34 BLYNK_WRITE(V0) {
35     strInput=param.asStr();
36     lcd.clear();
37     lcd.setCursor(0,0);
38     lcd.print(strInput);
39 }
40
41 void loop()
42 {
43     Blynk.run();
44 }
```

2.3 LCD 제어하기

◆ 2.3.3 Blynk 가상 LCD 제어(가상핀)

- Widget 설정(한글 가능)



2.3 LCD 제어하기

◆ 2.3.3 Blynk 가상 LCD 제어(가상핀)

● 소스 코드

```
2 // LCD Control(Text)_vpin -> lcd_control_vpinlcd(NodeMCU->스마트폰)
3
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #define BLYNK_PRINT Serial
7
8 // I2C 통신 라이브러리 설정
9 #include <Wire.h>
10 // I2C LCD 라이브러리 설정
11 #include <LiquidCrystal_I2C.h>
12 // LCD I2C address 설정 PCF8574:0x27, PCF8574A:0x3F
13 LiquidCrystal_I2C lcd(0x20, 16, 2);
14
15 // You should get Auth Token in the Blynk App.
16 // Go to the Project Settings (nut icon).
17 char auth[] = "KWIEYYjACwL_4t3z-5wNjHSxw3Uftqy4";
18
19 char ssid[] = "sjpark";
20 char pass[] = "12345678";
21
22 String strInput="";
23
24 WidgetLCD vLCD(V0);
25
26 void setup()
27 {
28     Blynk.begin(auth, ssid, pass);
29 }
30
31 void loop()
32 {
33     Blynk.run();
34     vLCD.print(0,0,"Test");
35     vLCD.print(0,1,"Embeded IoT Exp.");
36 }
```

2.4 Buzzer 제어하기

◆ 버저(Buzzer)

- 전기적 신호로 진동판을 진동시켜 소리를 출력하는 부품
- 마그네틱 버저, 피에조 버저, 기계식 버저 등이 있음
- 피에조 버저는 일정 주파수를 입력시켜 다양한 음을 낼 수 있음



마그네틱 버저



피에조 버저

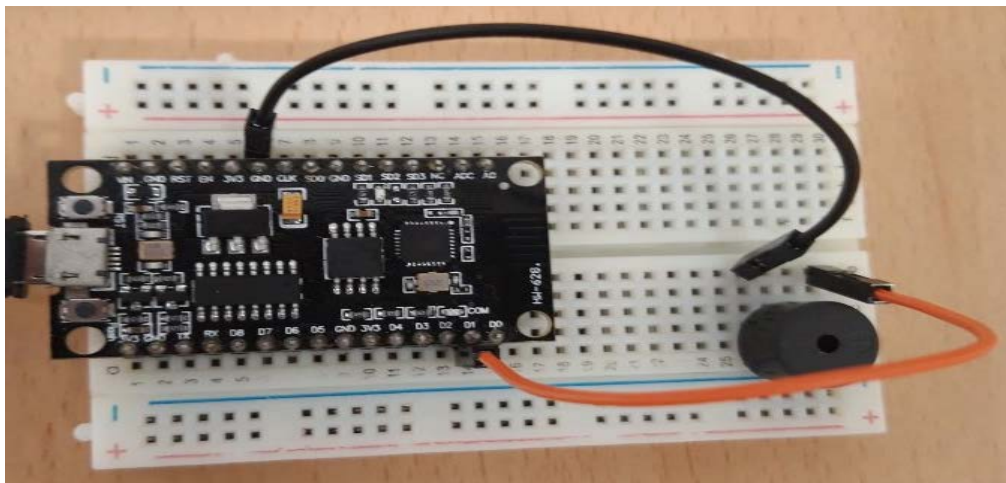
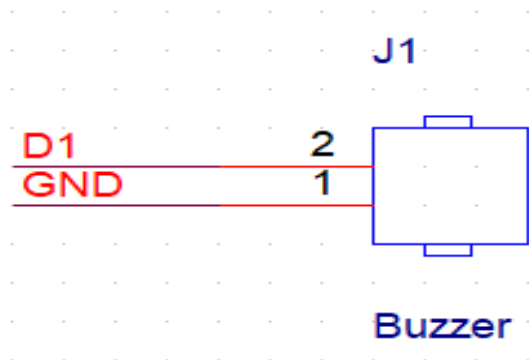


기계식 버저

2.4 Buzzer 제어하기

◆ 2.4.1 버저(피에조) 제어하기

- 회로도 및 연결도



2.4 Buzzer 제어하기

◆ 2.4.1 버저(피에조) 제어하기

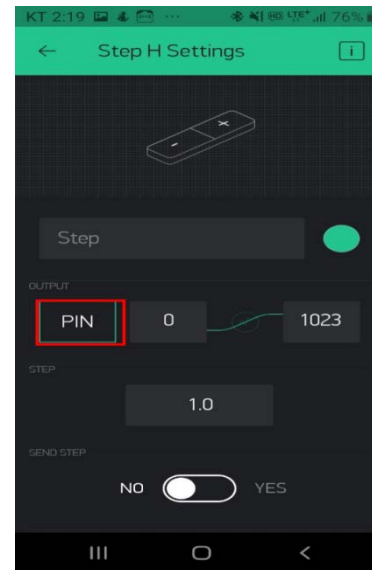
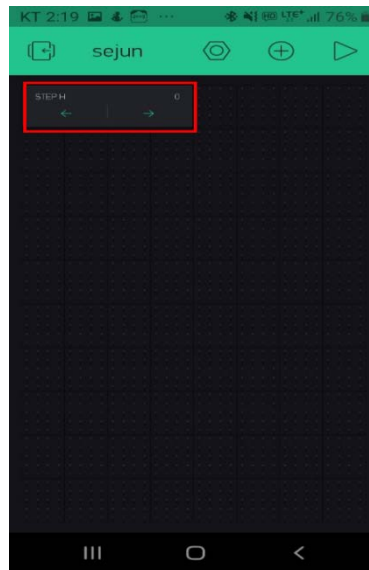
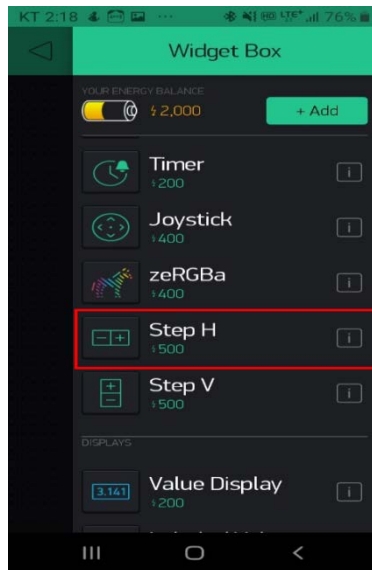
- 소스 코드

```
2 // Buzzer Control -> buzzer_control
3
4 int fre[] = {523, 587, 650, 698, 783, 880, 987, 1046};
5
6 void setup()
7 {
8     pinMode(D1, OUTPUT);
9 }
10
11 void loop()
12 {
13     for(int i=0; i<=7; i++)
14     {
15         tone(D1, fre[i], 200);
16         delay(500);
17     }
18 }
```


2.4 Buzzer 제어하기

◆ 2.4.2 Blynk 버저 제어(가상핀)

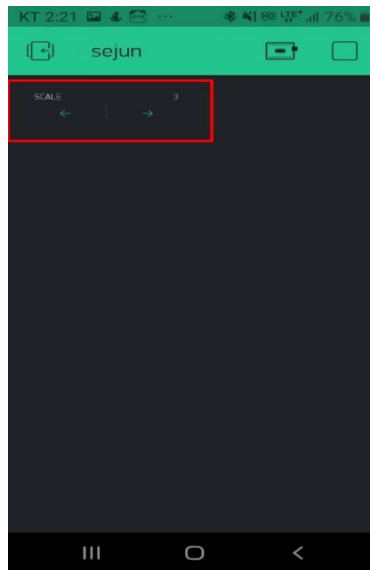
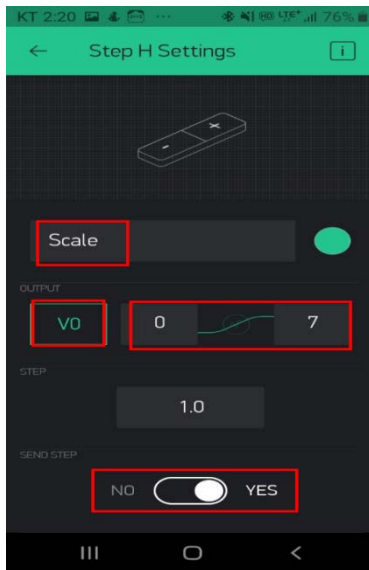
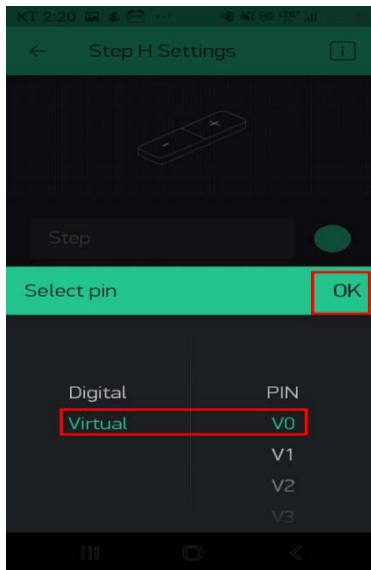
- Widget 설정



2.4 Buzzer 제어하기

◆ 2.4.2 Blynk 버저 제어(가상핀)

- Widget 설정



2.4 Buzzer 제어하기

◆ 2.4.2 Blynk 버저 제어(가상핀)

● 소스 코드

```
2 // Buzzer Control -> buzzer_control_vpin(스마트폰->NodeMCU)
3
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6 #define BLYNK_PRINT Serial
7
8 // You should get Auth Token in the Blynk App.
9 // Go to the Project Settings (nut icon).
10 char auth[] = "KWiEYYjACwL_4t3z-5wNjHSxw3Uftqy4";
11
12 char ssid[] = "sjpark";
13 char pass[] = "12345678";
14
15 int fre[] = {523, 587, 650, 698, 783, 880, 987, 1046};
16 int scale;
17
18 BLYNK_WRITE(V0)
19 {
20     scale = param.asInt();
21 }
```

```
23 void setup()
24 {
25     Blynk.begin(auth, ssid, pass);
26     pinMode(D1, OUTPUT);
27 }
28
29 void loop()
30 {
31     Blynk.run();
32     tone(D1, fre[scale], 200);
33     delay(500);
34 }
```