

Part04a - Blynk Broker

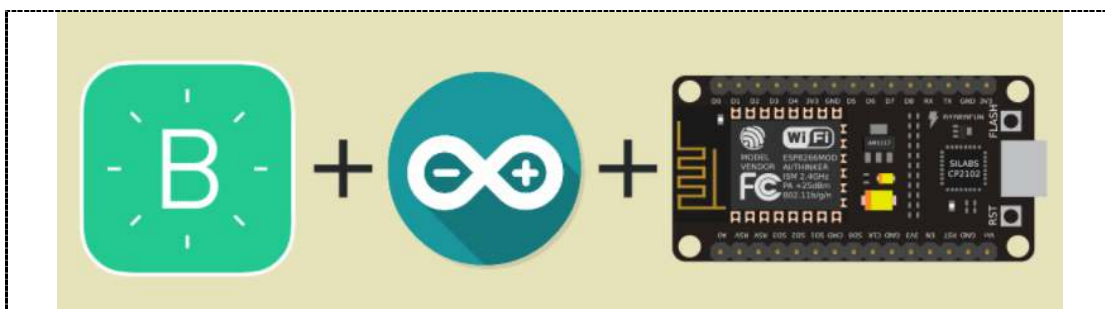
Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet.

<http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>

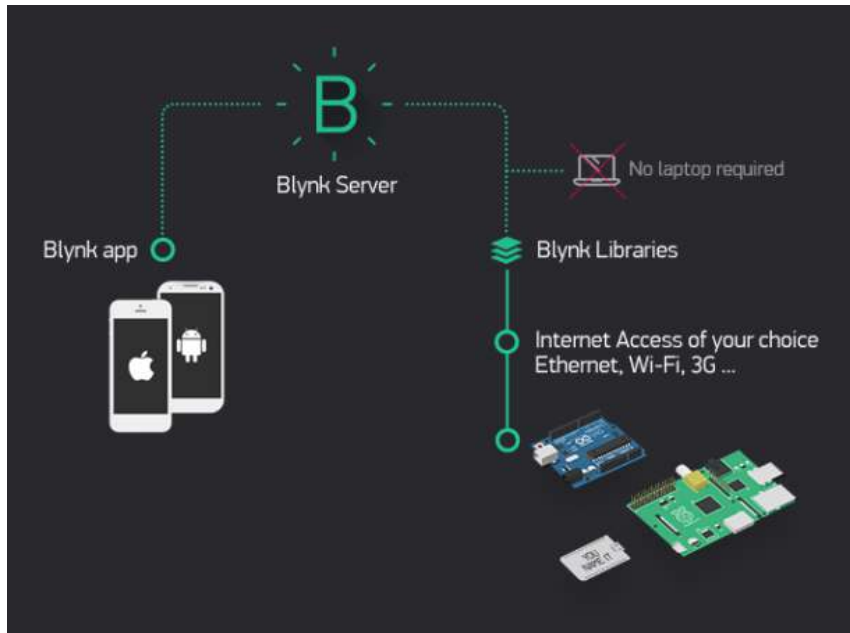
1b. เนื้อหา



Blynk เป็น cloud platform ที่ให้บริการฟรี สำหรับ IOT, Blynk เป็น Application สำเร็จรูปสำหรับงาน IOT มีความน่าสนใจคือการเขียนโปรแกรมที่ง่าย ไม่ต้องเขียน App เองสามารถใช้งานได้แบบ Real time สามารถเชื่อมต่อ Device ต่างๆเข้ากับ Internet ได้อย่างง่ายดาย ไม่ว่าจะเป็น Arduino, Esp8266, Esp32, Nodemcu, Rasberry pi นำมาแสดงบน Application ได้อย่างง่ายดาย แล้วที่สำคัญ Application Blynk ยังฟรี และ รองรับในระบบ IOS และ Android อีกด้วย



Blynk สามารถเชื่อมต่ออุปกรณ์ Device ของเราเข้ากับ internet ได้อย่างง่ายดาย ไม่ว่าจะเป็น Arduino, ESP8266, Rasberry pi หรือแม้แต่อื่นๆ ที่รวมเอา widget ต่างๆมาควบคุมแทนการเขียน code ยากๆ ไม่เพียงเท่านั้น ทางเลือกในการเชื่อมต่อเข้ากับ Blynk server เรายังสามารถใช้ได้ทั้ง WiFi และเครือข่ายมือถือ โดยสามารถ Download application นี้ได้ฟรีทั้งระบบ IOS และ Android

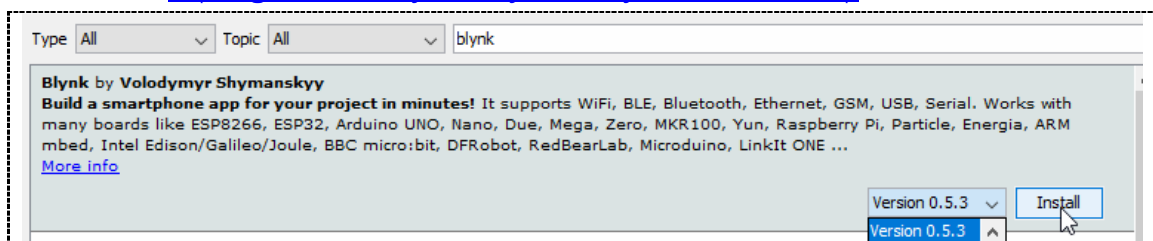


อ่านเพิ่มเติม

- <https://www.blynk.cc/>, <http://docs.blynk.cc/>
- <http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>
- <https://www.9arduino.com/article/59/app-สำเร็จรูป-blynk-nodemcu-esp8266-ตอนที่-1-blynk-คืออะไร>
- <http://www.ayarafun.com/2015/08/easy-iot-play-with-blynk/>
- <https://github.com/blynkkk/blynk-server>
- <http://thaiopensource.org/มาเล่น-blynk-กับ-esp8266-กัน/>

3b. การทดลอง

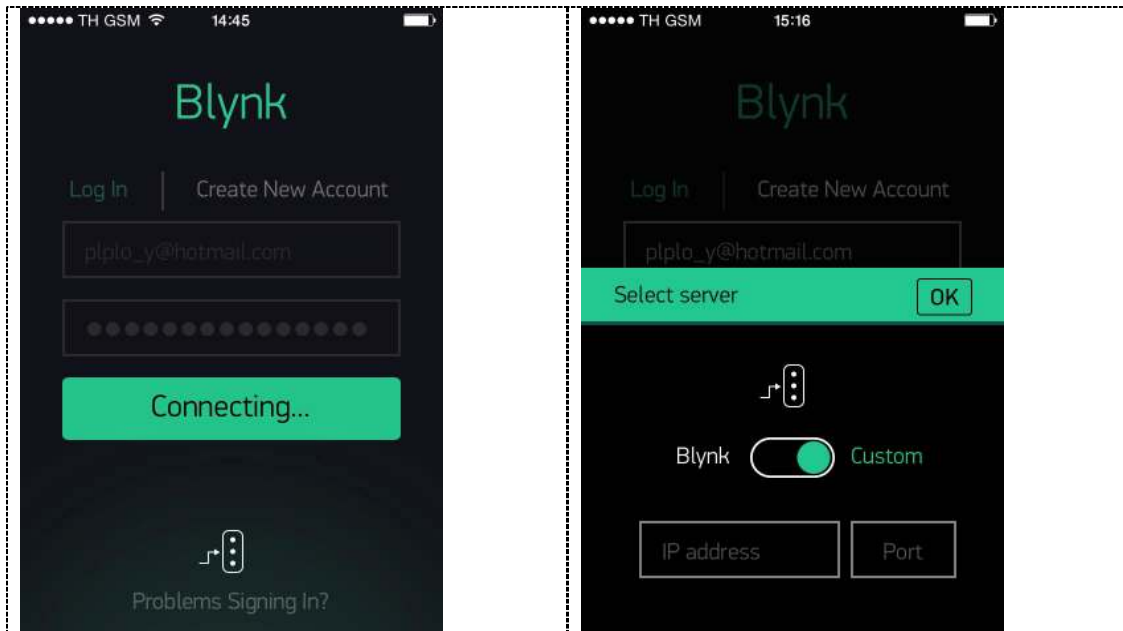
1. ติดตั้ง Blynk Application บนมือถือ
2. ติดตั้ง library Blynk เพื่อใช้งานกับ ESP32 บน Arduino IDE เลือกใช้ **Version 0.5.3** หรือติดตั้งจาก <https://github.com/blynkkk/blynk-library/archive/master.zip>



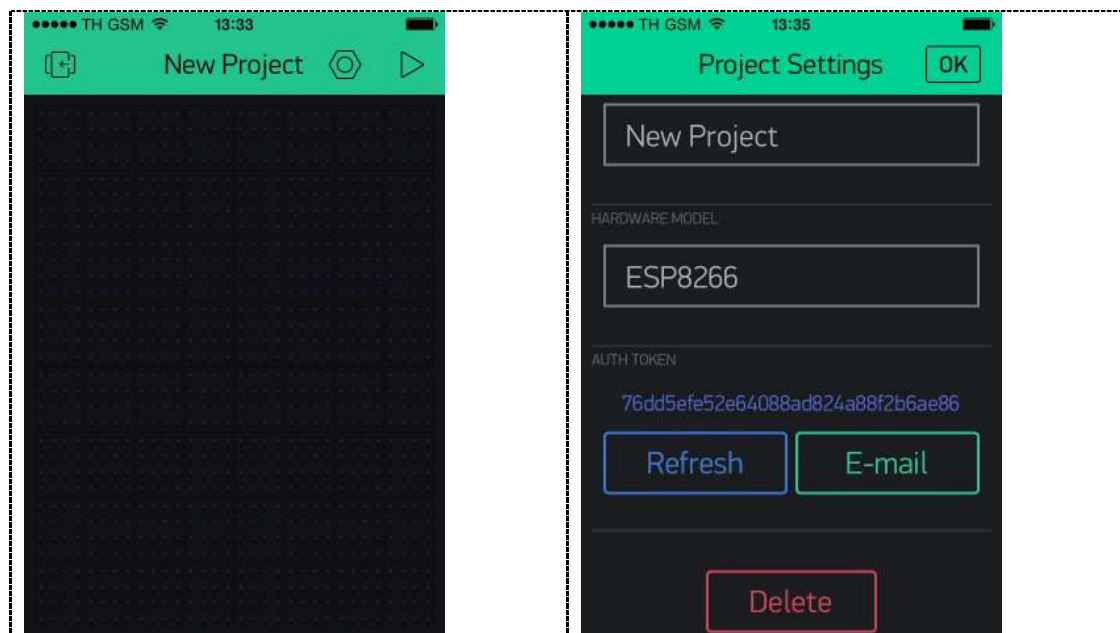
3. เริ่มต้นใช้งาน Blynk

เราต้องสมัคร ลงทะเบียน เพื่อใช้งานก่อน ให้เลือกที่คำว่า Create New account เพื่อสร้างการเชื่อมต่อกับ application กับ Email ของเรา

นอกจากนี้ เรายังสามารถ Connect เข้ากับ server ของ Blynk ของเราเองได้ โดยเลือก Custom และใส่ IP Address ของ Server เราเอง โดยกดที่รูป Problems Signing In แล้วเลื่อน scroll จาก Blynk ไป Custom



ต่อมาเป็นการสร้าง Project ของเราด้วย Blynk ให้กดที่สัญลักษณ์หกเหลี่ยมมุมขวบนเพื่อตั้งค่า โดยในหน้านี้เราสามารถตั้งชื่อ Project ของเราและเลือกรูปแบบ Hardware ที่เราจะใช้ได้ Hardware ให้เลือก ESP8266

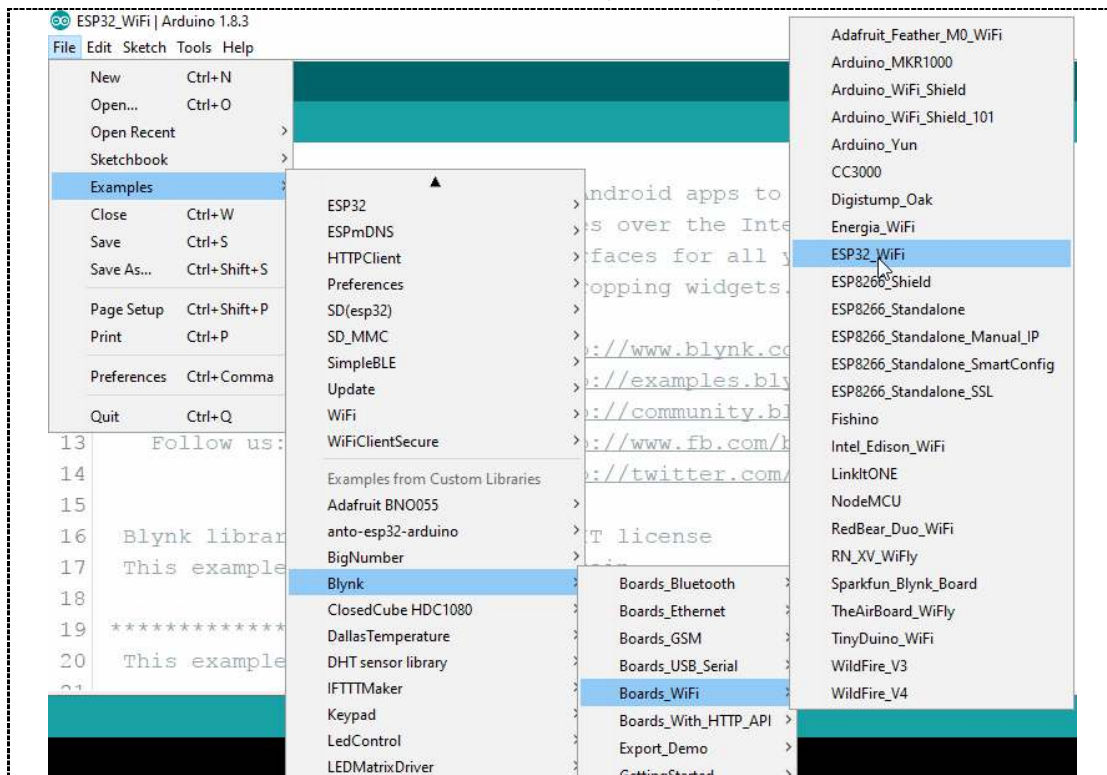


โดยทุกครั้งที่เริ่มสร้างโปรเจกใหม่ AUTH TOKEN จะถูกเปลี่ยนใหม่เสมอ ซึ่ง KEY นี้เองที่เป็นเสมือนกุญแจสำหรับเชื่อมต่อ โดยที่เราไม่ต้องใช้ user, password เราสามารถกดที่คำว่า "E-mail" เพื่อส่ง KEY นี้เข้าเมลเราได้



4. การใช้งานบน Arduino IDE

หลังจากเราลง Library Blynk เรียบร้อยแล้วจะมี Example ที่ติดมาด้วยจำนวนมาก ซึ่งในบทความนี้จะขอแนะนำตัวอย่างง่ายๆ เริ่มต้นไปกับ File → Example → Blynk → Board_Wifi → ESP32_Wifi



ในโค้ดนี้เราจะสังเกตเห็นว่ารูปแบบการสั่งงานสั้นมากใน void loop() มีเพียง Blynk.run(); เพื่อสั่งงานจากภายนอก

<pre>ESP32_WiFi\$ 1 2 #define BLYNK_PRINT Serial 3 4 #include <WiFi.h> 5 #include <WiFiClient.h> 6 #include <BlynkSimpleEsp32.h> 7 8 char auth[] = "YourAuthToken"; 9 char ssid[] = "YourNetworkName"; 10 char pass[] = "YourPassword"; 11 12 void setup() 13 { Serial.begin(9600); 14 Blynk.begin(auth, ssid, pass); 15 } 16 17 void loop() 18 { Blynk.run(); 19 }</pre>	<pre>#define BLYNK_PRINT Serial #include <WiFi.h> #include <WiFiClient.h> #include <BlynkSimpleEsp32.h> char auth[] = "YourAuthToken"; char ssid[] = "YourNetworkName"; char pass[] = "YourPassword"; void setup() { Serial.begin(115200); Blynk.begin(auth, ssid, pass); } void loop() { Blynk.run(); }</pre>
---	--

และในบรรทัด char auth[] = "YourAuthToken"; ให้ไป copy key ของ Auth Token จากใน Email ที่เราได้รับมาใส่

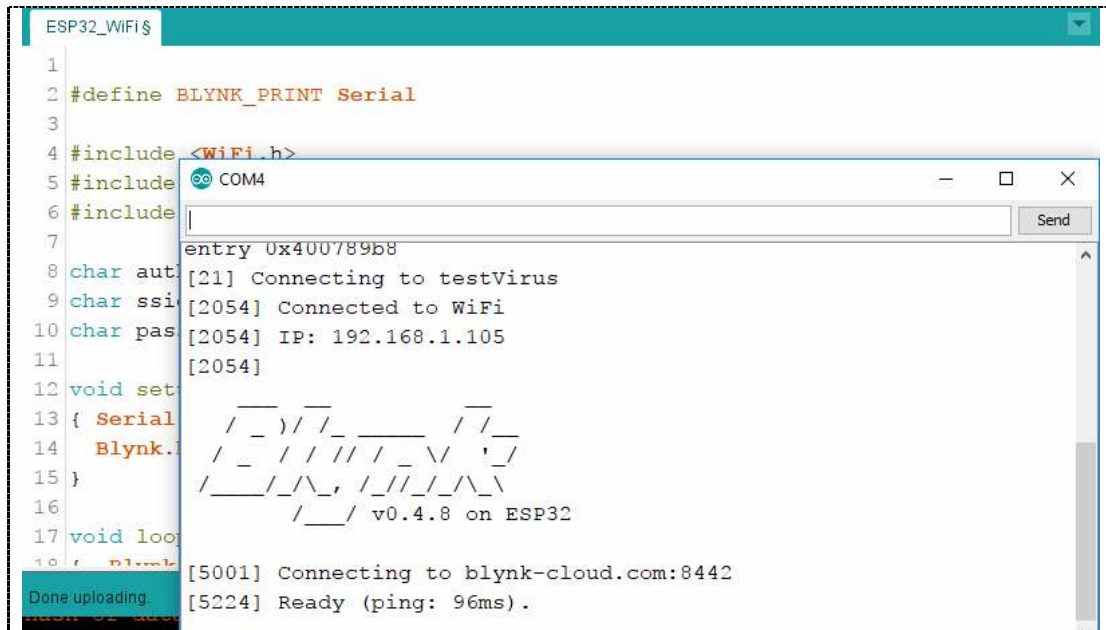
<pre>8 char auth[] = "d172e0defc664664afc11def54aa45e4"; 9 char ssid[] = "testVirus"; 10 char pass[] = "1510031510"; 11</pre>

บรรทัด Blynk.begin(auth,"ssid","pass"); ให้ใส่ชื่อ Wifi ที่เราใช้เชื่อมต่อกับ app ใน ssid และ password wifi ใน pass

และในกรณีเราเปลี่ยนรูปแบบการเชื่อมต่อ Wifi เป็นแบบ IP Address เราสามารถเขียนโค้ดได้ว่า

<pre>Blynk.begin(auth, SSID, pass)); to Blynk.begin(auth, SSID, pass, "your_host"); or to Blynk.begin(auth, SSID, pass, IPAddress(XXX,XXX,XXX,XXX));</pre>
--

จากนั้นให้ทำการ Upload ลง board แล้วเปิด Serial Monitor จนกระทั่งมีข้อความขึ้นแบบนี้



The image shows a code editor window with the title 'ESP32_WIFI\$' and a Serial Monitor window titled 'COM4'. The code in the editor is as follows:

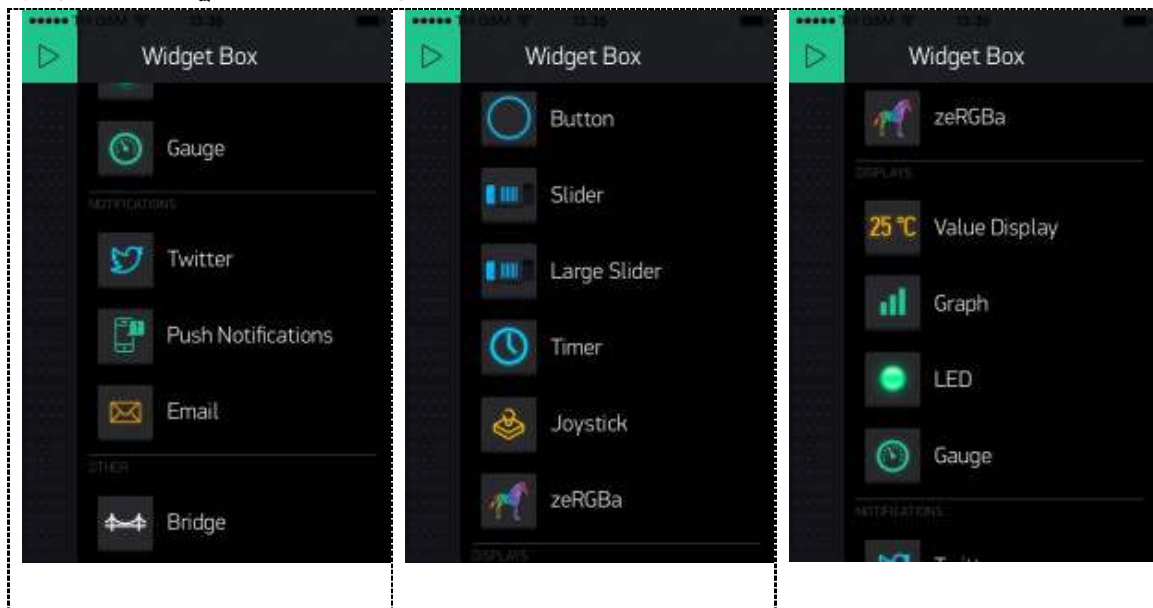
```
1
2 #define BLYNK_PRINT Serial
3
4 #include <WiFi.h>
5 #include
6 #include
7
8 char aut[21] Connecting to testVirus
9 char ssid[2054] Connected to WiFi
10 char password[2054] IP: 192.168.1.105
11 [2054]
12
13 void setup()
14 {
15   Serial.begin(115200);
16   Blynk.begin(auth, ssid, password, BLYNK_SERVER_URL, BLYNK_PORT);
17   Blynk.config(BLYNK_PRINT);
18 }
19
20 void loop()
21 {
22   Blynk.run();
23 }
```

The Serial Monitor window shows the following output:

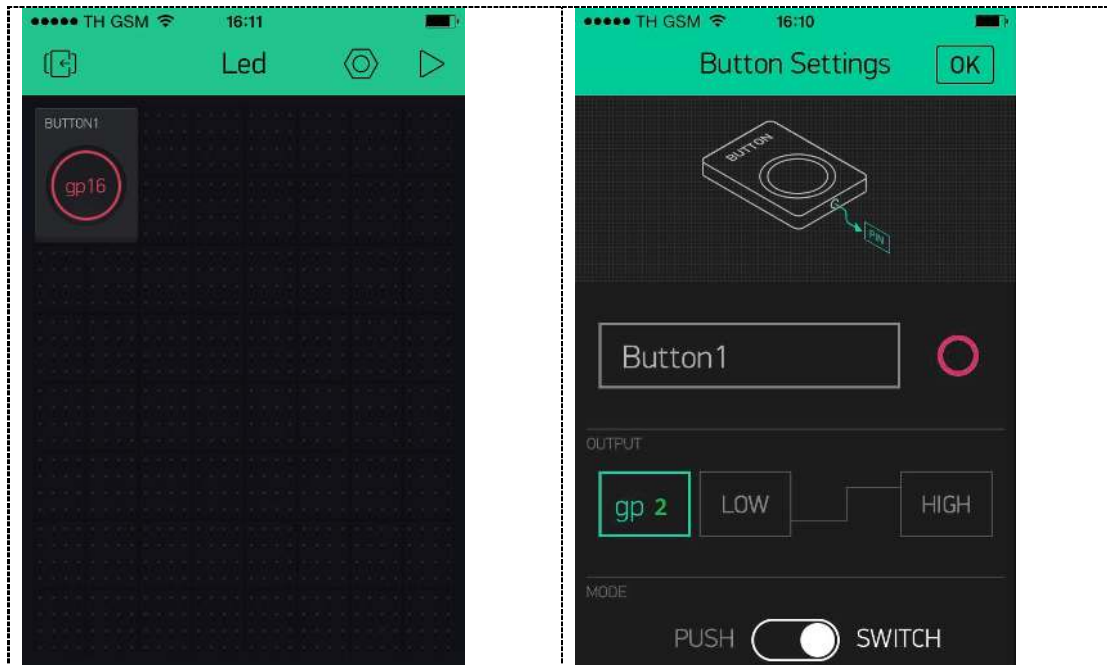
```
entry 0x400789b8
[21] Connecting to testVirus
[2054] Connected to WiFi
[2054] IP: 192.168.1.105
[2054]
[5001] Connecting to blynk-cloud.com:8442
[5224] Ready (ping: 96ms).
```

5. สร้างโปรเจกของเรปบน Blynk

ต่อมาเราจะเริ่มสร้างโปรเจกของเรปบน Blynk โดยการกดที่พื้นที่ว่างเปล่าตรงไหนก็ได้ในหน้า New project จะปรากฏหน้าต่างต่างของ Widget ให้เราเลือกขึ้นมา



ในบทความรอบนี้เราจะลองให้ดูอะไรที่ง่ายๆ ก่อน ให้ลองเลือก Button widget มาลงบนพื้นที่ว่างเปล่ามา 1 อัน จากนั้นเราจะมาตั้งค่าการใช้งานปุ่ม Button กันโดยกดไปที่รูป Button ที่เราเลือกจะปรากฏหน้าต่างแบบนี้ ซึ่งในหน้าต่างนี้เราสามารถเปลี่ยนชื่อปุ่มได้ และเลือกโหมด output pin ที่ต่อกับอุปกรณ์จาก board ของเราได้



เลือกรูปแบบ pin จะให้เป็นขา Digital หรือ Virtual ก็ได้ ซึ่งรูปแบบ Virtual จะไม่ใช้การรับค่าจากขาตรงๆ เป็นเหมือนการสร้างตัวแปรมาเก็บค่าอีกที และเลือกขา GPIO ให้ตรงกับ อุปกรณ์ที่เราจะต่อ เมื่อเสร็จแล้วกด ► เพื่อให้ App เริ่มทำงาน

- Button1
- SWITCH
- Select Pin
 - # Virtual → Variable
 - # Digital → Real Port
 - # gp2 < ESP32 On Board LED >
- กด ► เพื่อให้ App เริ่มทำงาน

ซึ่งรูปแบบการตั้งค่า Button นี้จะแทนการเขียนรูปแบบโค้ดเดิมๆใน void setup () ได้เลย

```
12 void setup()
13 { pinMode(2, OUTPUT);
14   Serial.begin(115200);
15   Blynk.begin(auth, ssid, pass);
16 }
```

เมื่อแก้ไขโปรแกรมแล้วจะได้ ดังนี้

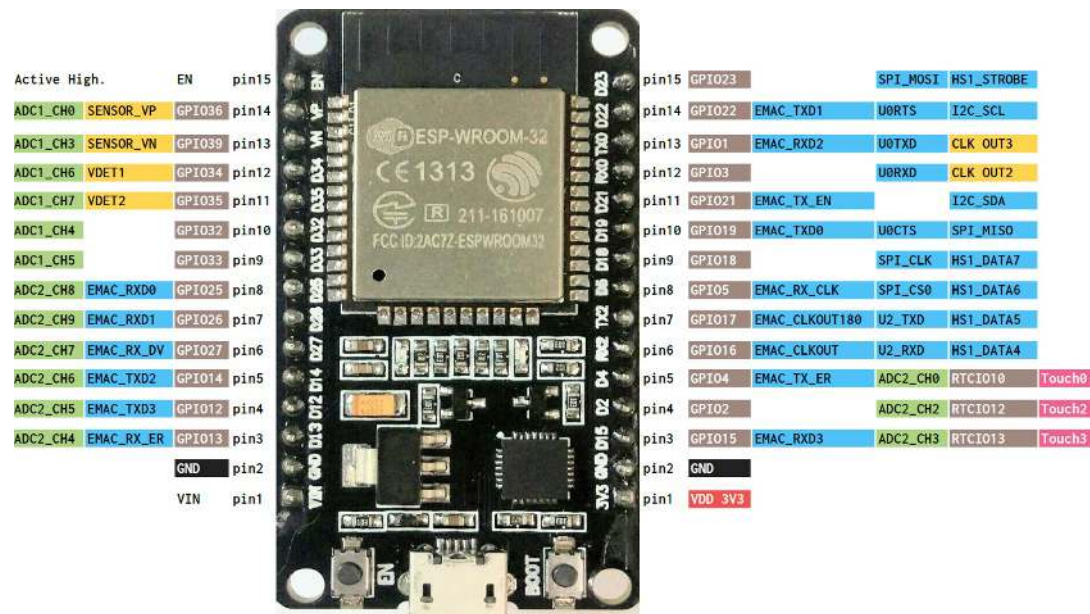
```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "f311e6ae4aa9421cb0f29b53061b305b";
char ssid[] = "SUT_IoTs";
char pass[] = "MaiMeeJingJing";

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

void loop()
{ Blynk.run();
}
```

ESP32 PINOUT



- ทดสอบการทำงานโดยการคุม LED GPIO2 หรือ On Board DOIT ESP32 Kit Ver1
- ให้ทำการทดสอบ Q-B1 : 4 External LED

8. การทดสอบอ่านสวิตช์ DO แล้วแสดงผลที่ LED Port V5 บน Blynk

- จาก Web <http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>
- จาก Web <https://community.blynk.cc/t/how-to-turn-on-widget-leds/643>
- ทดสอบโปรแกรม ให้แก้ไข (1/3)Auth, (2/3)SSID และ (3/3>Password

```
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

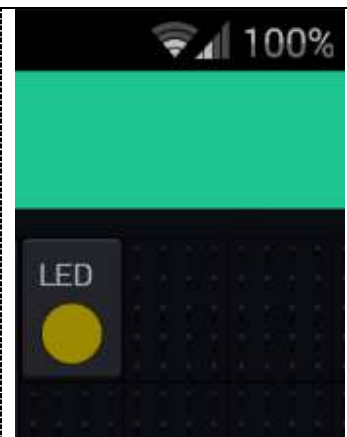
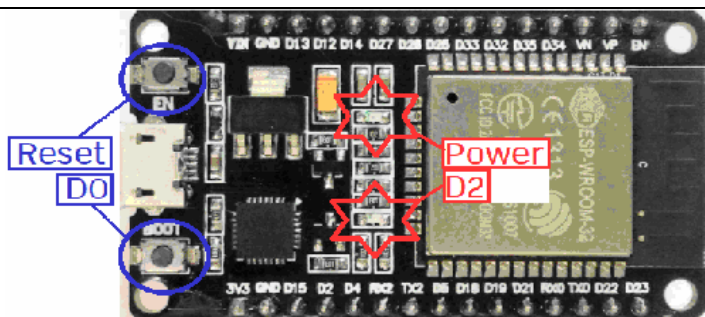
const int btnPin = 0; // DO
boolean btnState = false;
WidgetLED blynk_LED(V5);
BlynkTimer timer; // Announcing the timer

char auth[] = "f311e6ae4aa9421cb0f29b53061b305b";
char ssid[] = "SUT_IoTs";
char pass[] = "MaiMeeJingJing";

void myTimerEvent()
{
  boolean isPressed = (digitalRead(btnPin) == LOW);
  if (isPressed != btnState)
  {
    if (isPressed)
      blynk_LED.on();
    else
      blynk_LED.off();
    btnState = isPressed;
    Serial.print(" LED Status = ");
    Serial.println(btnState);
  }
}

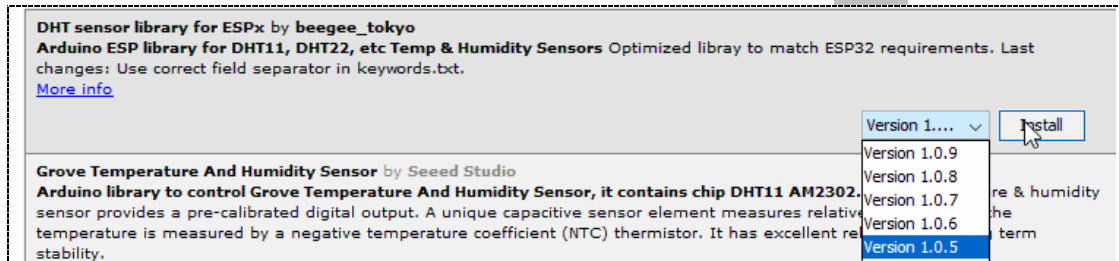
void setup()
{
  Serial.begin(115200);
  pinMode(btnPin, INPUT_PULLUP);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(250L, myTimerEvent);
}

void loop()
{
  Blynk.run();
  timer.run(); // running timer every 250ms
}
```



9. การทดสอบอ่านอุณหภูมิด้วย DHT-22 แล้วแสดงผลที่ Blynk

- จาก Web <http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>
- Install **DHT22 Library** เลือก **DHT Sensor library for ESPx V1.0.5** แล้วทำการติดตั้ง



10. ต่อวงจร DHT-22 เข้าที่ขา D15 และทดสอบการทำงานของโปรแกรม

The diagram shows an ESP32 development board with a DHT-22 sensor connected to it. The sensor's VCC pin is connected to the board's VCC pin (labeled D0), and its GND pin is connected to the board's GND pin. The sensor's data pin is connected to the board's D15 pin. The board's D15 pin is also connected to a 10k pull-up resistor, which is connected to the board's VCC pin. The sensor's output is connected to a 10k pull-down resistor, which is connected to the board's GND pin. The board's D15 pin is also connected to a 10k pull-up resistor, which is connected to the board's VCC pin. The sensor's output is connected to a 10k pull-down resistor, which is connected to the board's GND pin.

A close-up of the DHT-22 sensor module, showing the white plastic housing and the black PCB with the sensor chip. The chip is labeled 'AOSUNG AM2302 SN150402743'. The module has three pins: VCC, GND, and DATA. The text '+ DG' is visible next to the pins.

```
#include "DHTesp.h"
DHTesp dht;
const int pinDHT_22 = 15;

void setup()
{ Serial.begin(115200);
  dht.setup(pinDHT_22);
}

void loop()
{ float temperature = dht.getTemperature();
  float humidity = dht.getHumidity();
  Serial.print(" Temp('C) >> "); Serial.print(temperature, 1);
  Serial.print(", Humidity(%) >> "); Serial.println(humidity, 1);
  delay(2000);
}
```

11. ทดสอบการทำงานของโปรแกรม ให้แก้ไข (1/3)Auth, (2/3)SSID และ (3/3>Password

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "DHTesp.h"

BlynkTimer timer;    // Announcing the timer
DHTesp dht;

char auth[] = "f311e6ae4aa9421cb0f29b53061b305b";
char ssid[] = "SUT_IoTs";
char pass[] = "MaiMeeJingJing";

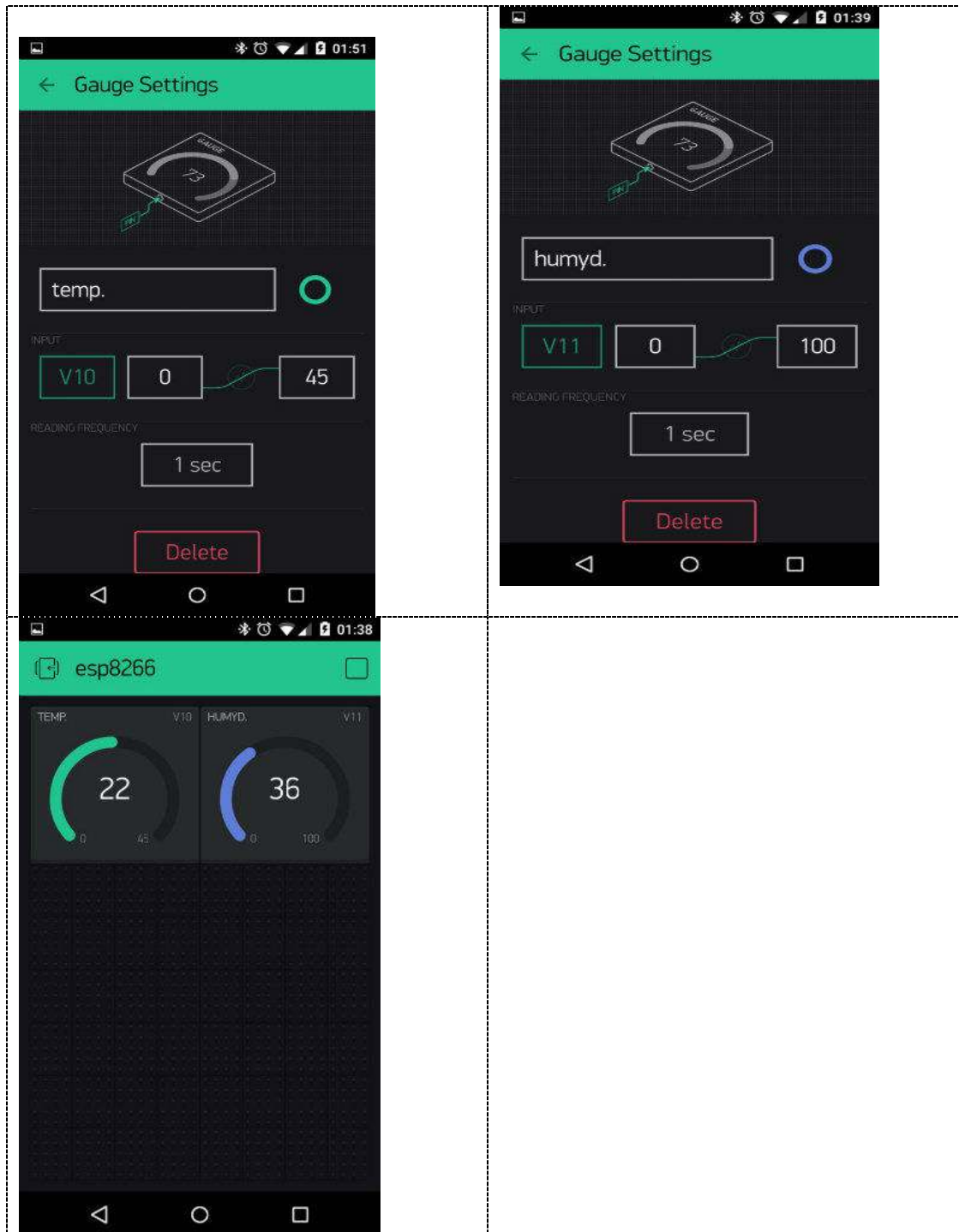
const int pinDHT_22 = 15; // D15

void myTimerEvent()
{ float temperature = dht.getTemperature();
  float humidity = dht.getHumidity();
  Blynk.virtualWrite(V10, temperature);
  Blynk.virtualWrite(V11, humidity);
  Serial.print(" Temp('C) >> ");          Serial.print(temperature, 1);
  Serial.print(", Humidity(%) >> ");      Serial.println(humidity, 1);
}

void setup()
{ Serial.begin(115200);
  dht.setup(pinDHT_22);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(1000L, myTimerEvent);
}

void loop()
{ Blynk.run();
  timer.run(); // running timer every 250ms
}
```

12. ที่ Blynk ให้ใช้ Gauge และ Port V10 และ Port V11

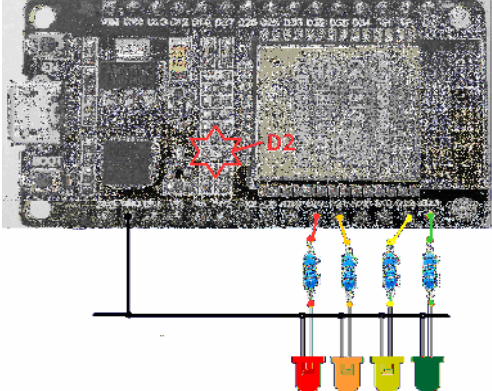


13. ให้ทำการทดสอบ Q-B2: DHT22, Switch and 4LED

PCเลขที่ _____ รหัส _____ ชื่อ-สกุล _____

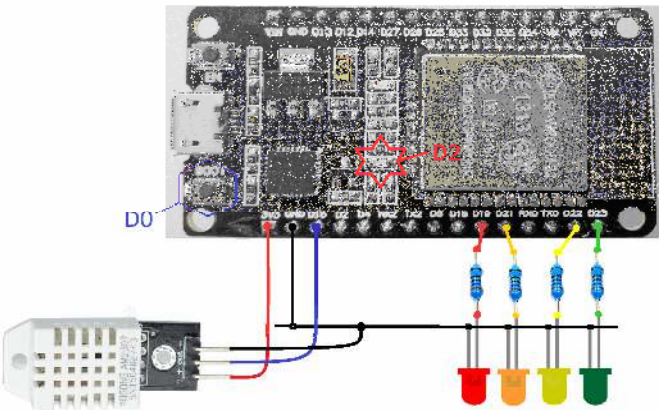
3. คำถามท้ายการทดลอง- ให้เขียนโปรแกรมเพื่อทำงานต่อไปนี้

Q-B1 - ปรับแก้โปรแกรมสำหรับควบคุม 4 LED ดังรูป



- D19 → Red
- D21 → Orange
- D22 → Yellow
- D23 → Green

Q-B2 - จาก Q1เพิ่มเติมให้อ่านแสดงอุณหภูมิ ความชื้น จาก DHT-11, Switch และควบคุม 4 LED



- D0 → Switch
- D19 → Red
- D21 → Orange
- D22 → Yellow
- D23 → Green
- D15 → DHT11

คะแนน ความสวยงามของ Widget = _____ / 10