

## แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

### IoT Approaches to Manufacturing System

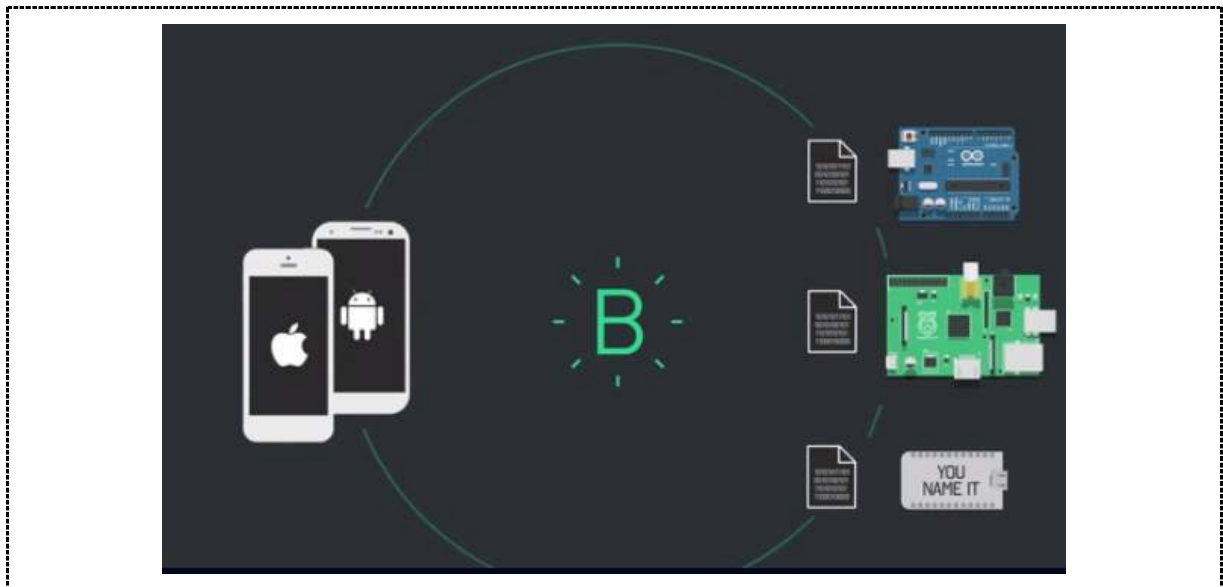
3/4 - แนะนำ MQTT Protocol และ IoTs Broker

- การโปรแกรมใช้งาน Blynk Broker
- การโปรแกรมใช้งาน IFTTT Broker
- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

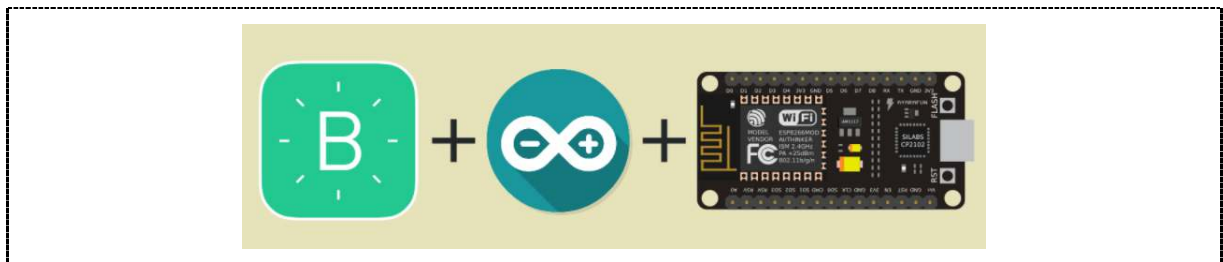
### 1/3 การโปรแกรมใช้งาน Blynk Broker

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

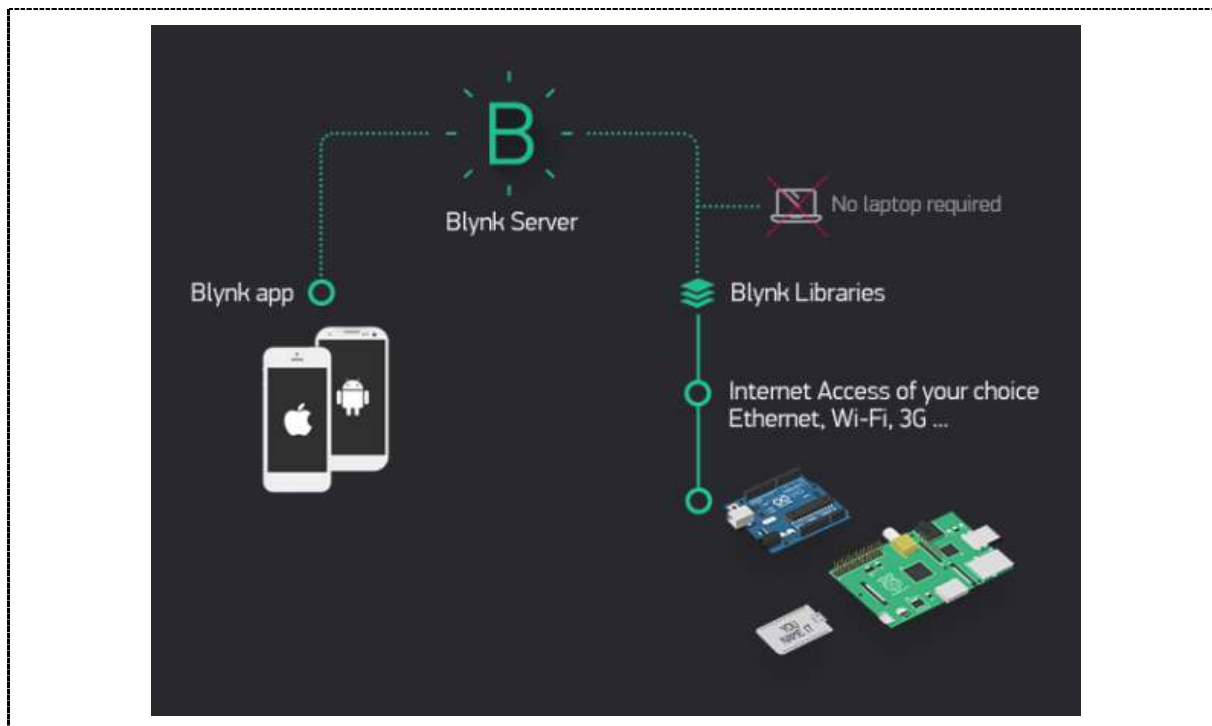
#### 1.1. เนื้อหา



Blynk เป็น cloud platform ที่ให้บริการฟรี สำหรับ IOT, Blynk เป็น Application สำเร็จรูปสำหรับงาน IOT มีความน่าสนใจคือการเขียนโปรแกรมที่ง่าย ไม่ต้องเขียน App เองสามารถใช้งานได้จริง Real time สามารถเชื่อมต่อ Device ต่างๆเข้ากับ Internet ได้อย่างง่ายดาย ไม่ว่าจะเป็น Arduino, Esp8266, Esp32, Nodemcu, Raspberry 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

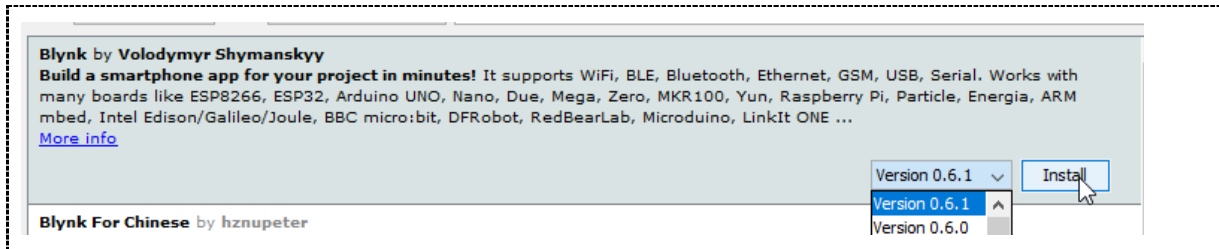


## 1.2 อ่านเพิ่มเติม

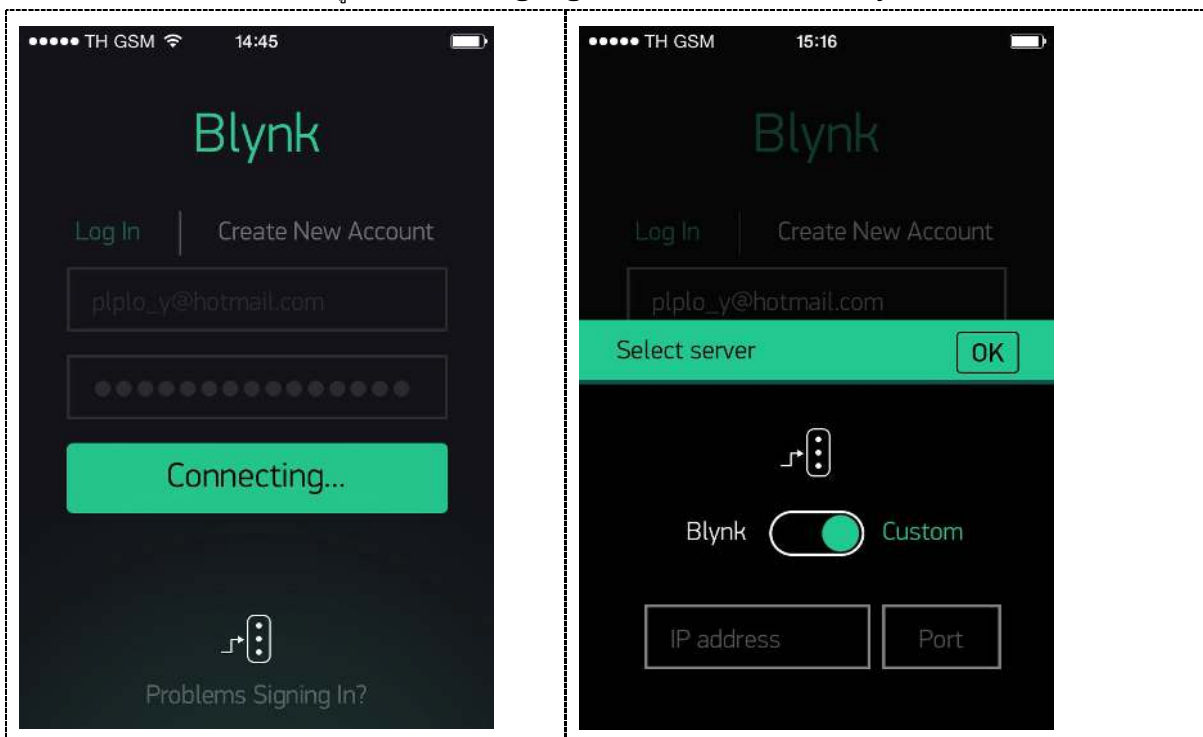
- <https://www.blynk.cc/>, <http://docs.blynk.cc/>
- <http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>
- <https://www.9arduino.com/article/59/app-สำเร็จรูป-blynk-nodemcu-esp8266-ตอนที่1-blynk-คืออะไร>
- <http://www.ayarafun.com/2015/08/easy-iot-play-with-blynk/>
- <https://github.com/blynkkk/blynk-server>
- <http://thaiopensource.org/มาเล่น-blynk-กับ-esp8266-กัน/>

### 1.3. การทดลอง

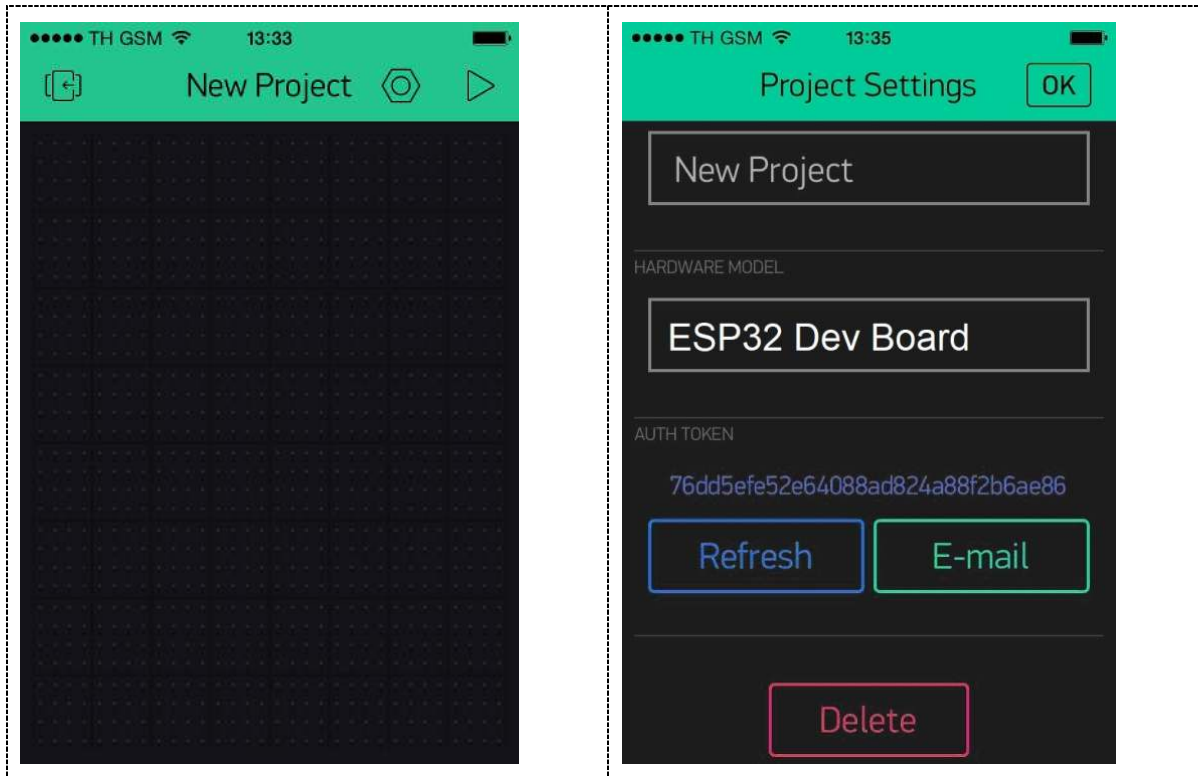
1. ติดตั้ง Blynk Application บนมือถือ
2. ติดตั้ง library Blynk เพื่อใช้งานกับ Esp32 บน Arduino IDE เลือกใช้ **Version 0.6.1** หรือติดตั้งจาก <https://github.com/blynkkk/blynk-library/archive/master.zip>



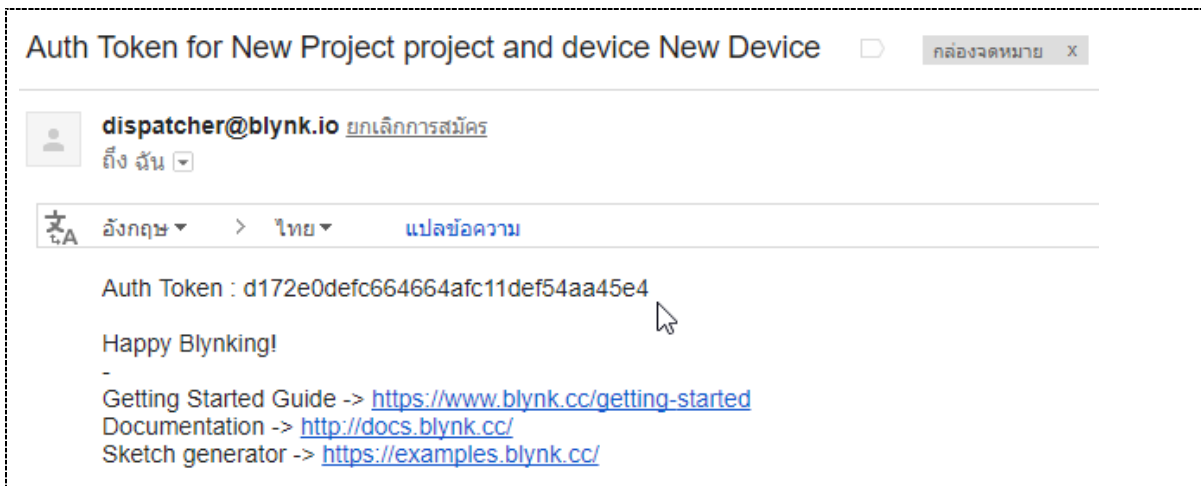
3. เริ่มต้นใช้งาน Blynk เราต้องสมัคร ลงทะเบียน เพื่อใช้งานก่อน ให้เลือกที่คำว่า Create New account เพื่อสร้างการเชื่อมต่อกับ application กับ Email ของผู้ใช้งาน
4. Login เข้า Blynk
5. เลือก Connect เข้ากับ server ของ Blynk ( Blynk ให้ผู้ใช้เลือก Custom Server ได้โดยใส่ IP Address ของ Server เรายกคที่รูป Problems Signing In แล้วเลื่อน scroll จาก Blynk ไป Custom )



6. ต่อมาเป็นการสร้าง Project ของเราด้วย Blynk ให้กดที่สัญลักษณ์หกเหลี่ยมมุมขวามบนเพื่อตั้งค่า
7. โดยในหน้านี้เราสามารถตั้งชื่อ Project ของเราและเลือกรูปแบบ Hardware ที่เราจะใช้ได้ Hardware ให้เลือก ESP32 Dev Board



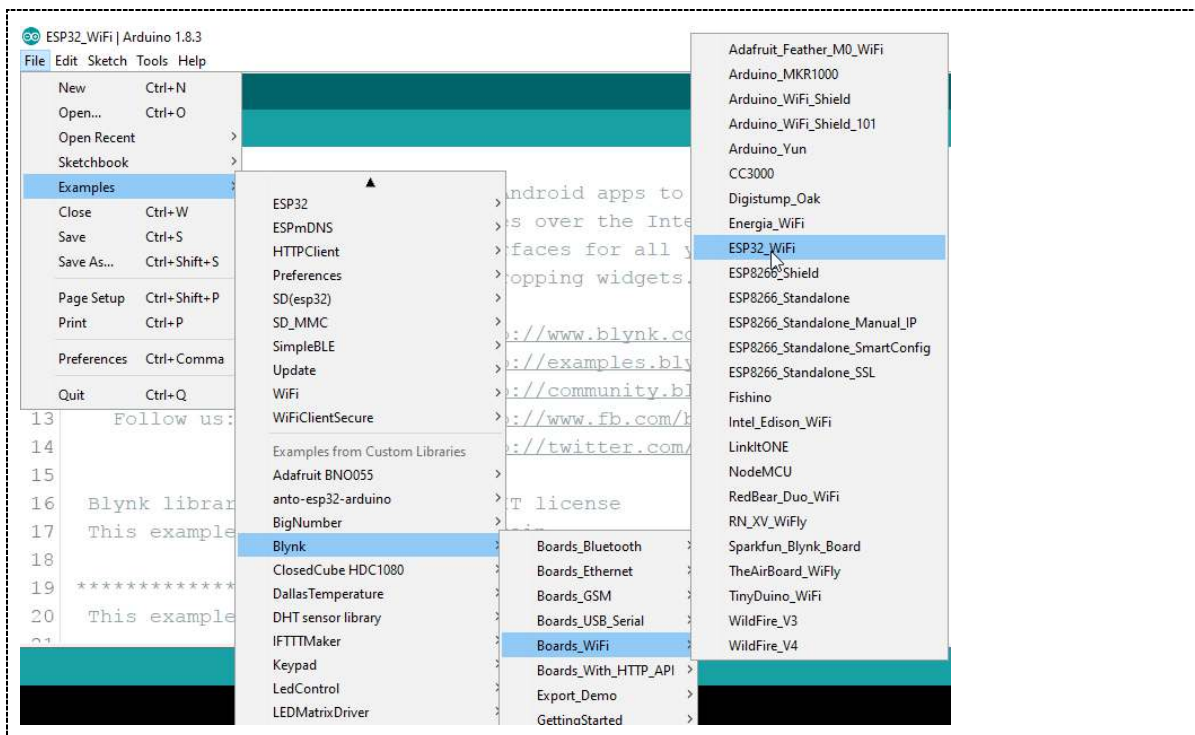
8. โดยทุกๆครั้งที่เริ่มสร้างโปรเจคใหม่ AUTH TOKEN จะถูกเปลี่ยนใหม่เสมอ ซึ่ง KEY นี้เองที่เป็นเสมือนกุญแจสำหรับเชื่อมต่อ โดยที่เราไม่ต้องใช้ user, password เราสามารถกดที่คำว่า” E-mail” เพื่อส่ง KEY นี้เข้าเมลเราได้



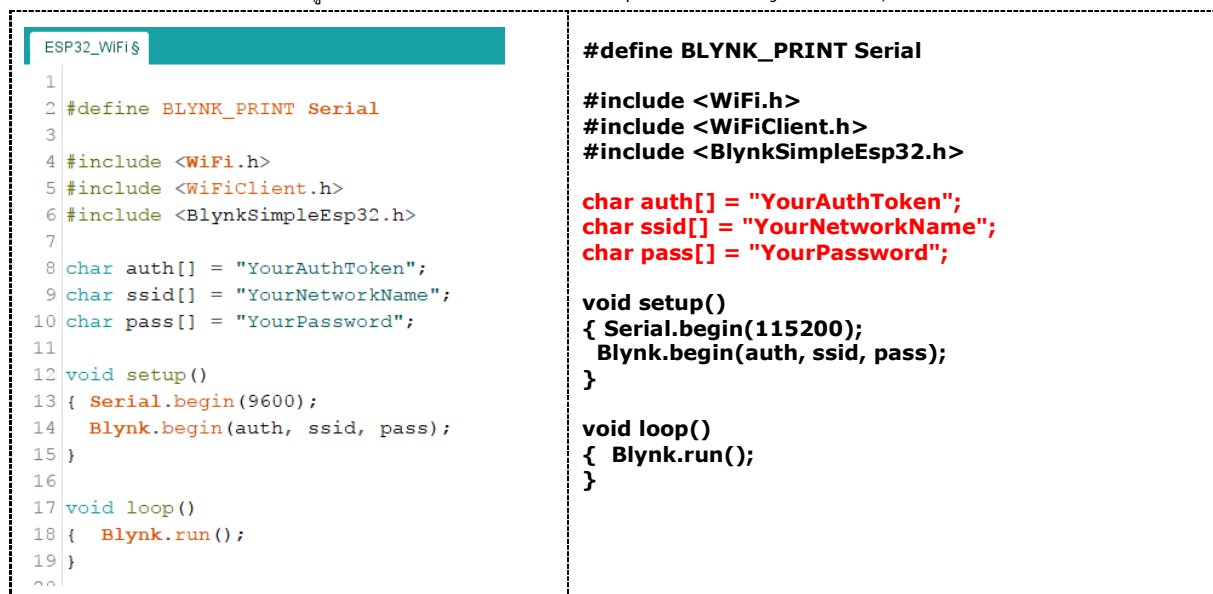
## การทดลองที่ 1 – Blynk Control

### 9. การใช้งานบน Arduino IDE

หลังจากเราลง Library Blynk เรียบร้อยแล้วจะมี Example ที่ติดมาด้วยจำนวนมาก ซึ่งในบทความนี้จะขอแนะนำตัวอย่างง่ายๆ เริ่มต้นไปกับ File → Example → Blynk → Board\_Wifi → ESP32\_Wifi



### 10. ในโค้ดนี้เราจะเห็นว่ารูปร่างแบบการสั่งงานใน void loop() มีเพียง Blynk.run(); เพื่อสั่งงานจากภายนอก



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

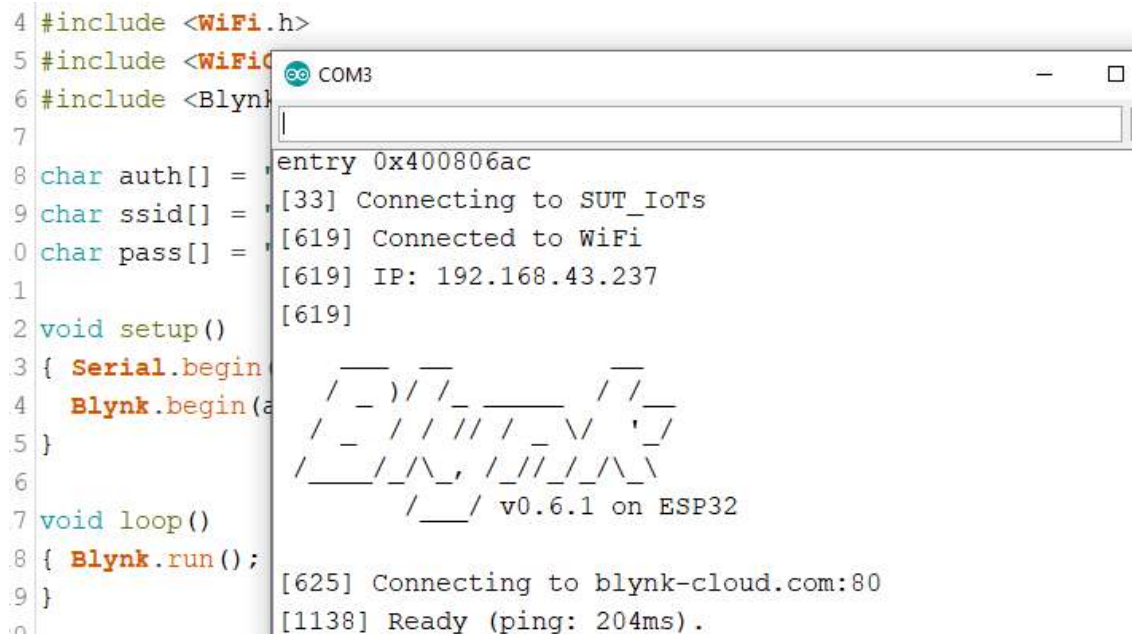
```
8 char auth[] = "d172e0defc664664afc11def54aa45e4";
9 char ssid[] = "testVirus";
10 char pass[] = "1510031510";
11
```

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

และในกรณีเราเปลี่ยนรูปแบบการเชื่อมต่อ Wifi เป็นแบบ IP Address เราสามารถเขียนโค้ดได้ว่า

```
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));
```

11. จากนั้นให้ทำการ Upload ลง board แล้วเปิด Serial Monitor จนกระทั่งมีข้อความขึ้นแบบนี้

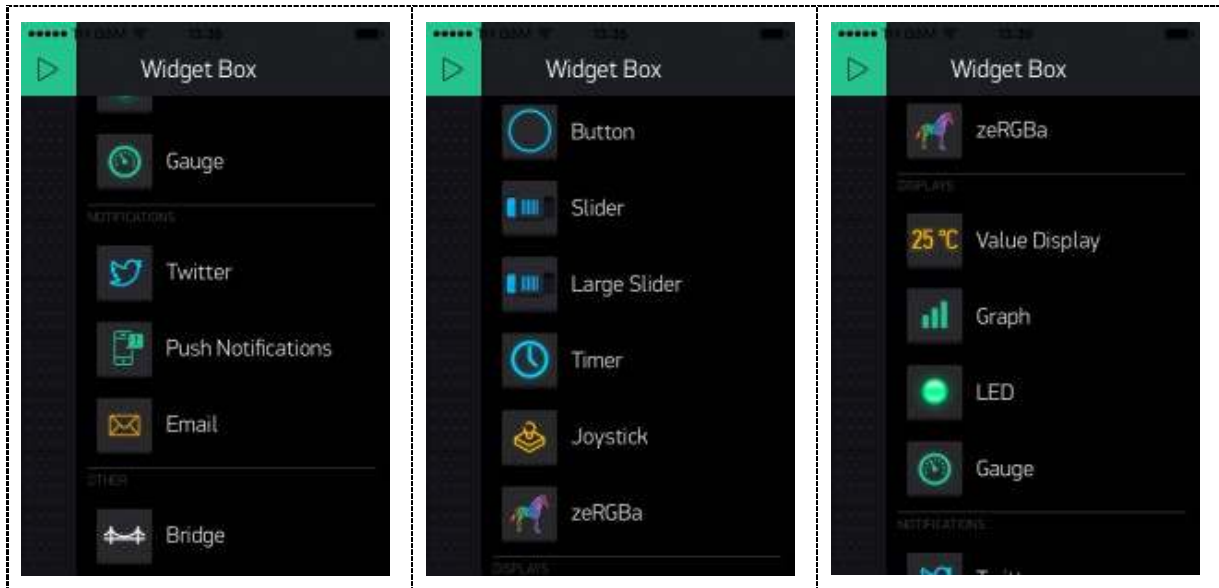


```
4 #include <WiFi.h>
5 #include <WiFiClient.h>
6 #include <Blynk.h>
7
8 char auth[] = "entry 0x400806ac";
9 char ssid[] = "[33] Connecting to SUT_IoTs";
10 char pass[] = "[619] Connected to WiFi";
11
12 void setup()
13 {
14   Serial.begin(115200);
15   Blynk.begin(auth, ssid, pass);
16 }
17
18 void loop()
19 {
20   Blynk.run();
21 }
22
```

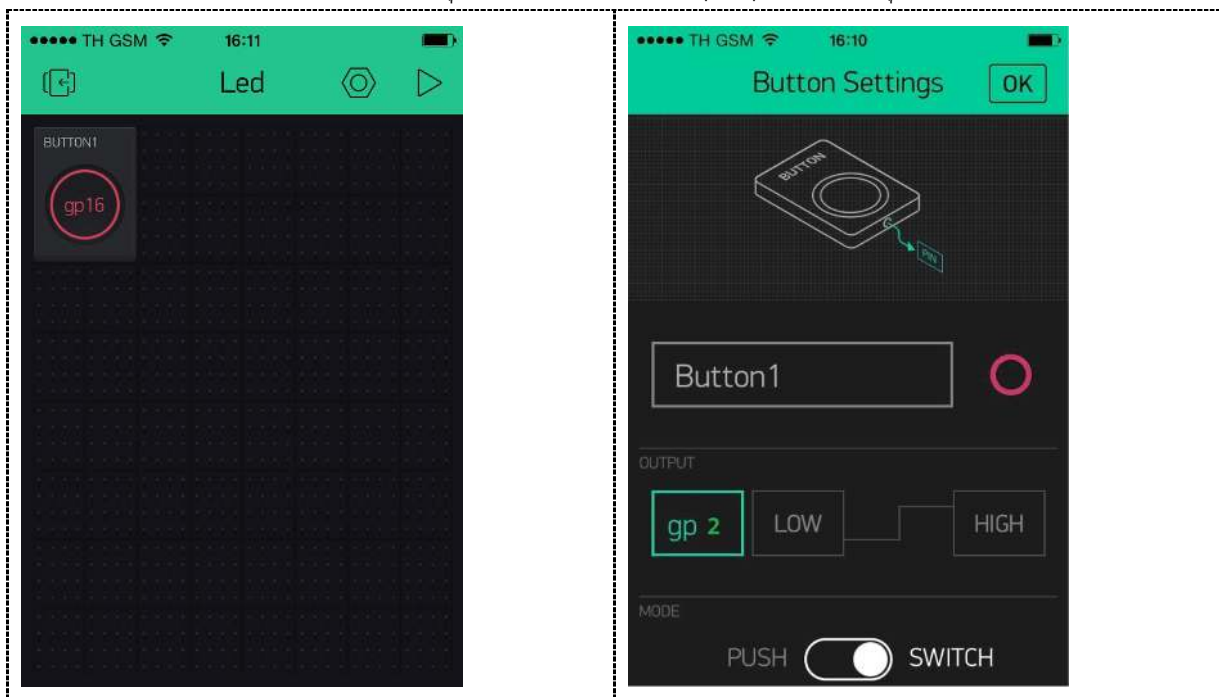
Serial Monitor (COM3) output:

```
[33] Connecting to SUT_IoTs
[619] Connected to WiFi
[619] IP: 192.168.43.237
[619]
[625] Connecting to blynk-cloud.com:80
[1138] Ready (ping: 204ms).
```

12. สร้างโปรเจกของเรบบน Blynk โดยการกดที่พื้นที่ว่างเปล่าตรงไหนก็ได้ในหน้า New project จะปรากฏหน้าต่างของ Widget ให้เราเลือกขึ้นมา

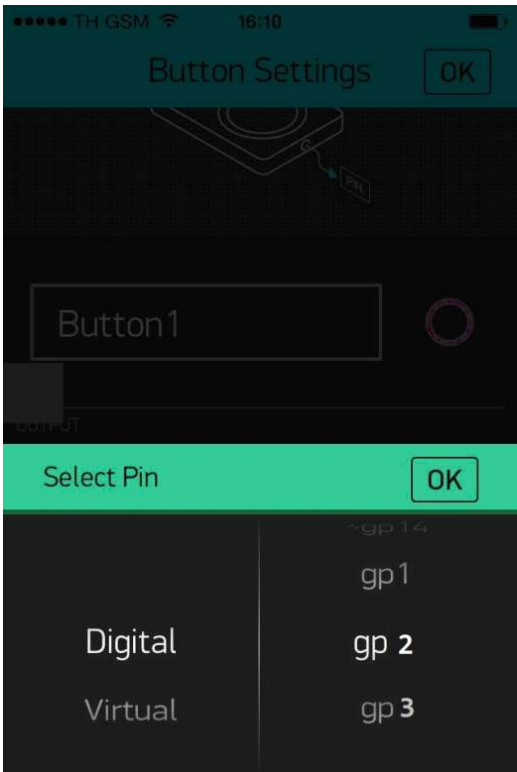


13. ในบทความรอบนี้เราจะลองให้ดูอะไรที่ง่ายๆ ก่อน ให้ลองเลือก Button widget มาลงบนพื้นที่ว่างเปล่ามา 1 อัน จากนั้นเราจะมาตั้งค่าการใช้งานปุ่ม Button กันโดยกดไปที่รูป Button ที่เราเลือกจะปรากฏหน้าต่างแบบนี้ ซึ่งในหน้านี้เราสามารถเปลี่ยนชื่อปุ่มได้ และเลือกโหมด output pin ที่ต่อกับอุปกรณ์จาก board ของเราได้





14. 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

15. เมื่อแก้ไขโปรแกรมแล้วจะได้ ดังนี้

```
#define BLYNK_PRINT Serial

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

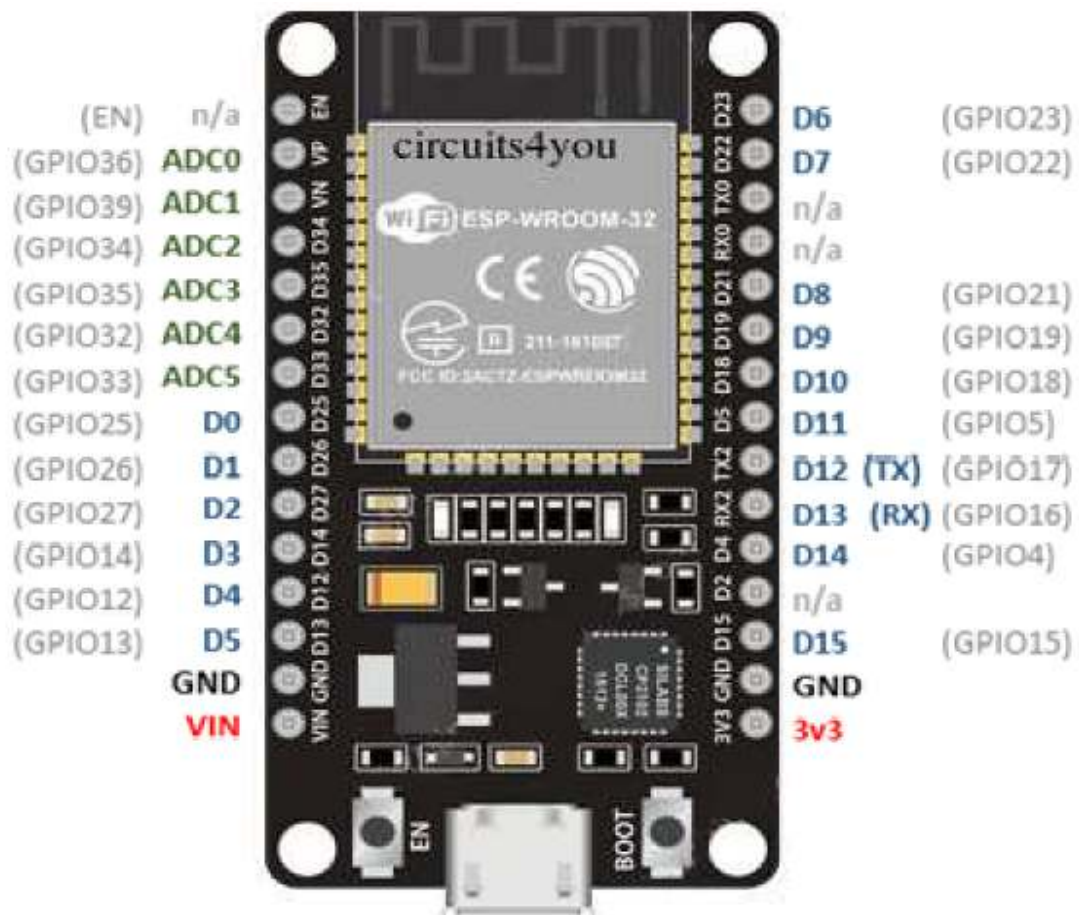
char auth[] = "f311e6ae4aa9421cb0f29b531b305b";
char ssid[] = "SUT_IoTs";
char pass[] = "MaiMeeJingJing";

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

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

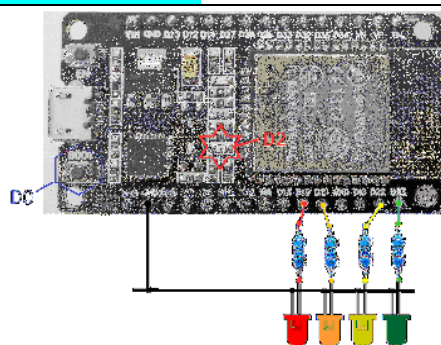


## ESP32 PINOUT



16. ทดสอบการทำงานโดยการคุม LED GPIO2 หรือ On Board DOIT ESP32 Kit Ver1

17. ให้ทำการทดสอบ Quiz-301 : 4 External LED



## การทดลองที่ 2 – Blynk Control and Monitor

19. 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

```
// ESP 32

#define BLYNK_PRINT Serial

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

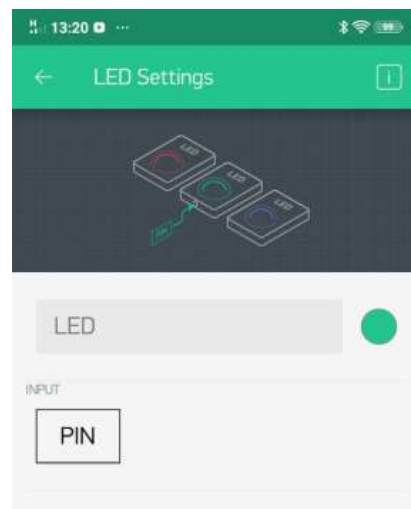
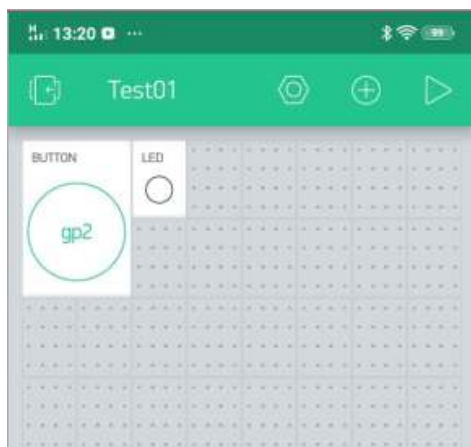
const int btnPin = 0; // D0
boolean btnState = false;
WidgetLED blynk_LED(V5);
BlynkTimer timer; // Announcing the timer

char auth[] = "f311e6ae4aa94210f29b53061b305b";
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);
}

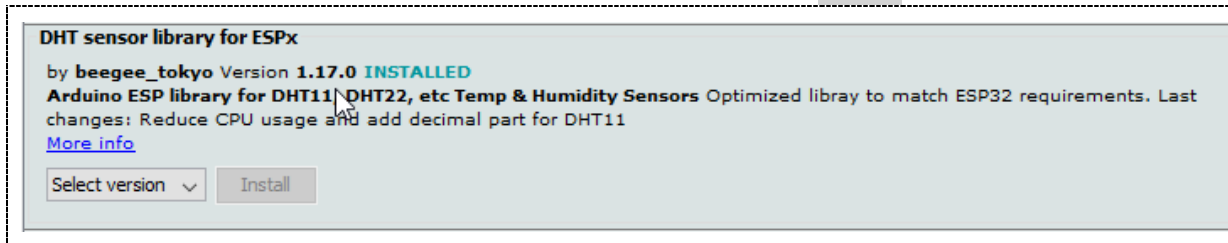
void loop()
{
  Blynk.run();
  timer.run(); // running timer every 250ms
}
```



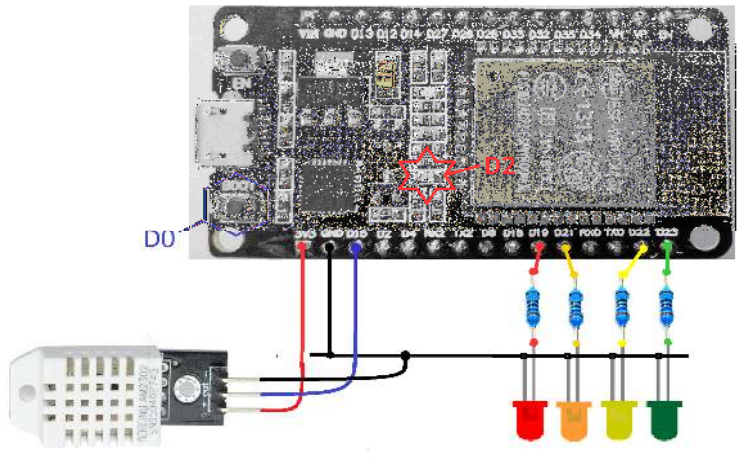

|  |  |   |
|--|--|---|
|  |  | <pre> / _ _ / _ _ / _ _ / _ _ / / _ _ / _ _ / _ _ / _ _ / / _ _ / _ _ / _ _ / _ _ / / _ _ / _ _ / _ _ / _ _ / / _ _ / _ _ / _ _ / _ _ / v0.6.1 on ESP32  [4678] Connecting to blynk-cl [4892] Ready (ping: 177ms). LED Status = 1 LED Status = 0 LED Status = 1 LED Status = 0 [37614] Heartbeat timeout [39617] Connecting to blynk-cl [42631] Login timeout [44631] Connecting to blynk-cl [46263] Ready (ping: 1621ms). </pre> |
|  |  |   |

20. 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 V1.17.0** แล้วทำการติดตั้ง



21. ต่อวงจร DHT-22 เข้าที่ขา D15 และทดสอบการทำงานของโปรแกรม

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

void setup()
{ Serial.begin(115200);
  dht.setup(pinDHT_22, DHTesp::DHT22);
}

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);
}
```

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

```
// ESP-32

#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[] = "f311e6ae4aa9421cb0f29b53061b3b";
char ssid[] = "SUT_IoTs";
char pass[] = "MaiMeeJingJing";

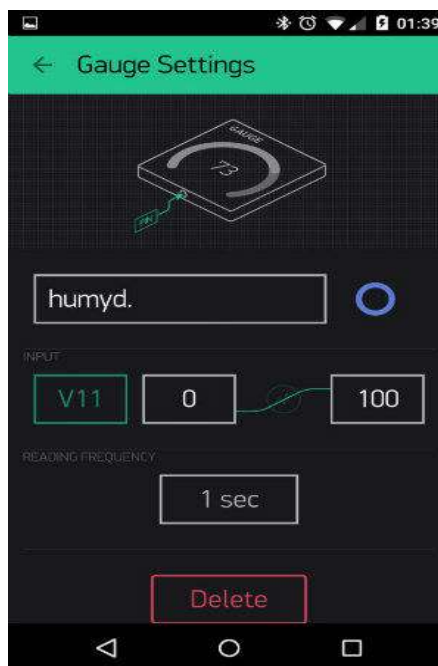
const int pinDHT_22 = 15; // D15
float temperature = 12.34, humidity = 56.78;

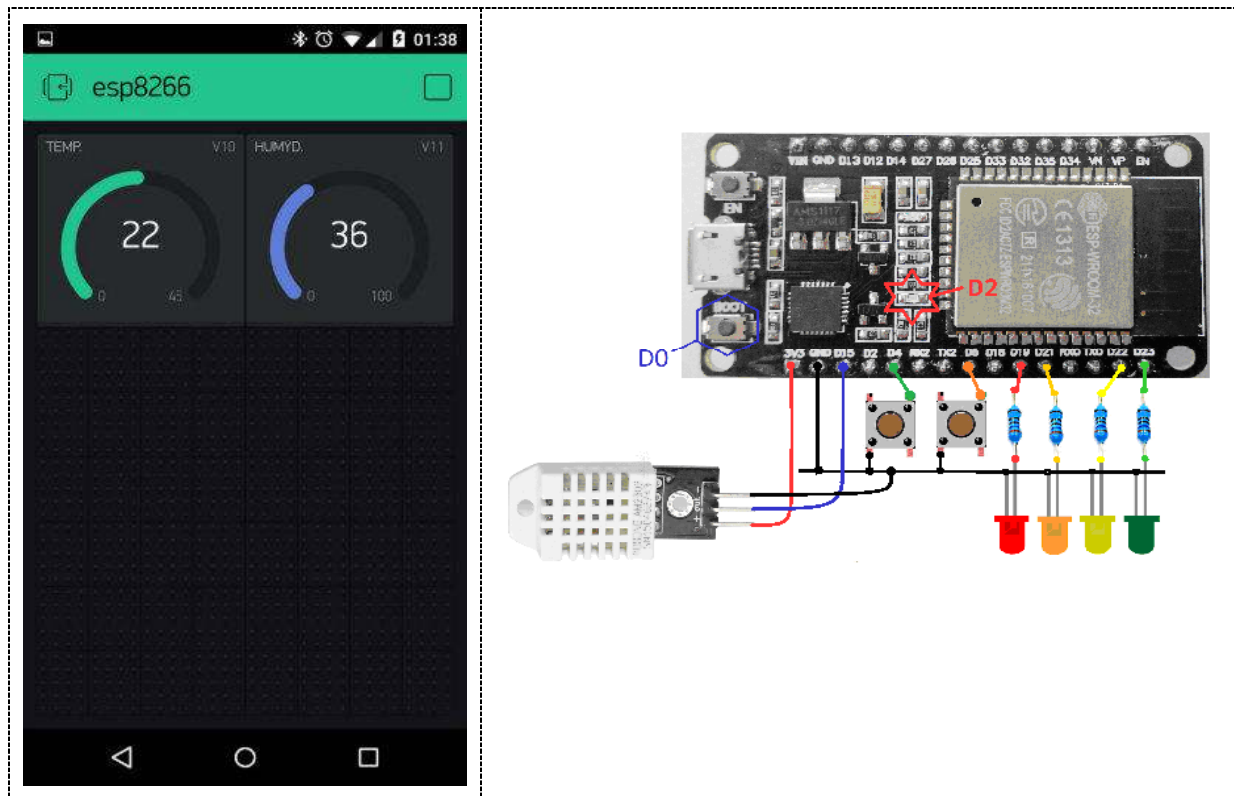
void myTimerEvent()
{
  temperature = dht.getTemperature();
  humidity = dht.getHumidity();
  Blynk.virtualWrite(V10, temperature);
  Blynk.virtualWrite(V11, humidity);
  Serial.print(" Temp('C) >> ");      Serial.print(temperature, 1);
  Serial.print(", Humidity(%) >> ");  Serial.println(humidity, 1);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

void setup()
{
  Serial.begin(115200);
  dht.setup(pinDHT_22, DHTesp::DHT22);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(1000L, myTimerEvent);
}

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

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



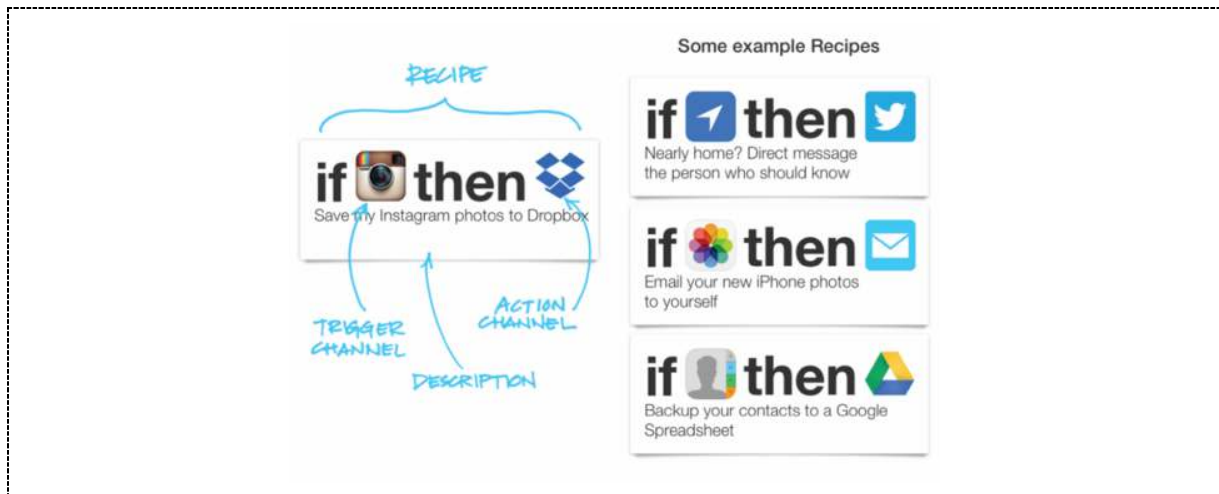


24. ให้ทำการทดสอบ QA-2: DHT22, 2-Switch and 4-LED









อาจจะฟังดูง่าย ๆ เบสิค แต่ถ้ามองแค่นี้เราคงไม่จำเป็นต้องพึ่งบริการของเจ้า IFTTT ให้ปวดสมอง แต่สิ่งที่ IFTTT ทำได้เจ๋งกว่านั้นก็คือ เราสามารถทำงานแบบอัตโนมัติร่วมกับ Social Media ต่างๆ รวมไปถึง productivity tool อีกหลายช่องทาง โดย IFTTT เรียกสิ่งเหล่านี้ว่า Channel ไม่ว่าจะเป็น facebook, twitter, drop box, evernote, instagram และอีกมากมาย



โดยเราสามารถสร้าง Recipe ที่เชื่อมโยง Channel ต่างๆ เข้าด้วยกันได้ อย่างเช่น ถ้า **"อัปรูปใหม่ลงใน Instagram"** ให้ **"เซฟรูปลงใน Dropbox"** ที่นี้ทุกครั้งที่เราอัปโหลดรูปใหม่ลงไปใน Instagram รูปเหล่านั้นก็จะถูกเก็บไว้ใน Dropbox ของเราด้วยโดยอัตโนมัติ

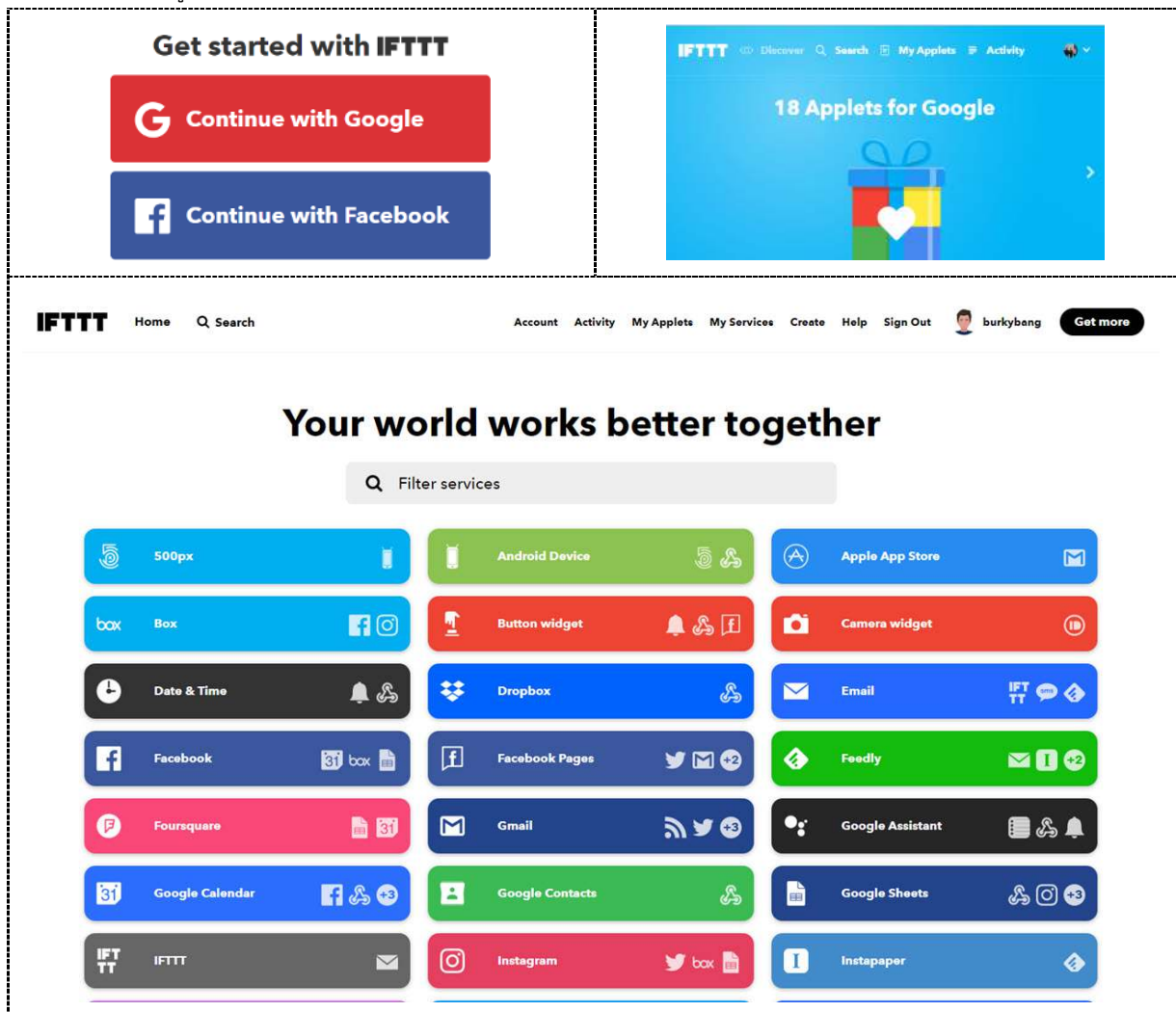
การสร้าง Personal Recipe ของ IFTTT เรียกว่าแทบจะไม่มีขอบเขต ขอเพียงแต่เราสร้างกฎที่ต้องการให้ IFTTT ทำภายใต้ Channel ต่างๆ มากมายที่เรามีไว้ให้ ถ้าอยากได้ไอเดีย ก็สามารถคลิกเข้าไปดูในแต่ละ Channel ได้ว่ามีใครเค้าสร้าง "สูตร" แบบไหนเอาไว้แล้วบ้าง หรือเข้าไปดูที่ <https://ifttt.com/recipes> เพื่อจะได้ลองเอามาใช้ให้เหมาะกับความต้องการ ที่สำคัญยังสามารถเปิดหรือปิด Recipe ต่างๆ ได้ตลอด ถ้าช่วงนี้ยังไม่อยากให้ IFTTT ทำตามเงื่อนไขอันนี้ก็ปิดไปก่อนได้ อยากให้ทำอีกเมื่อไหร่ก็ค่อยเปิดขึ้นมาใช้งานต่อได้ทุกเมื่อ ก็แค่สั่งให้เจ้า IFTTT ทำงานแทนเราเท่านั้นเอง

## 2.2 อ่านเพิ่มเติม

- <https://randomnerdtutorials.com/esp32-http-post-ifttt-thingspeak-arduino/>
- <https://marooter.blogspot.com/2017/06/week04-arduino-iots-sms.html>
- <http://www.olrepublic.com/careerlab/เทคนิคการทำงาน-อ่านแล้วใช้เลย-Working-Tips/1697-IFTTT-If-This-Then-That.html>
- <https://circuitdigest.com/microcontroller-projects/sending-sms-using-esp8266>
- <http://tesrteam.blogspot.com/2015/12/how-to-use-esp8266-nodemcu-post-update.html>
- <http://www.ioxhop.com/article/47/esp8266-esp8285-กับการส่งการแจ้งเตือนเข้า-line>

## 2.3 การทดลอง

1. ลงชื่อเข้าใช้งาน IFTTT ที่ <https://ifttt.com/discover> เพื่อความสะดวกแนะนำให้ใช้ Google Account ในการเข้าสู่ระบบ

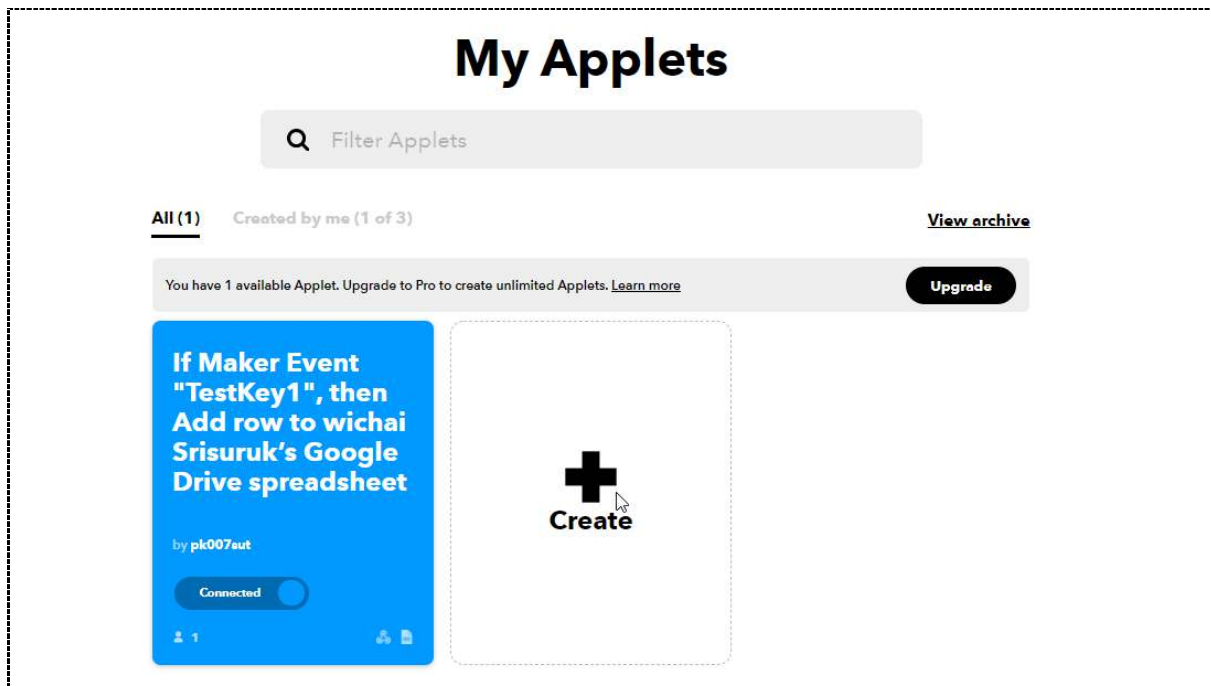


### การทดลองที่ 3 If Time to LINE

2. IFTTT Test\_1 → If Time Then LINE



- If **Date&Time** Then **LINE**

2.1 สร้าง Applet ใหม่ → Create









2.2 กำหนด If this → Date&Time








|   |  |
|---|--|
|  <h2>Connect Date &amp; Time</h2> <p>Step 1 of 6</p> <p>Turn on Applets that run on an hourly, daily, weekly, monthly or yearly basis using this service. It's automatically tailored to your time zone.</p> <p><b>Connect</b></p> |  <h2>Connect Date &amp; Time</h2> <p>Select your current time zone to connect the Date &amp; Time service.</p> <p>Time zone<br/>(GMT+07:00) Bangkok</p> <p><b>Connect</b></p> |
| <p><b>Every day at</b></p> <p>This Trigger fires every single day at a specific time set by you.</p>  | <p><b>Every day at</b></p> <p>This Trigger fires every single day at a specific time set by you.</p> <p>Time<br/>12 AM</p> <p>00 Minutes</p> <p><b>Create trigger</b></p>  |

### 2.3 กำหนด Then that → SMS

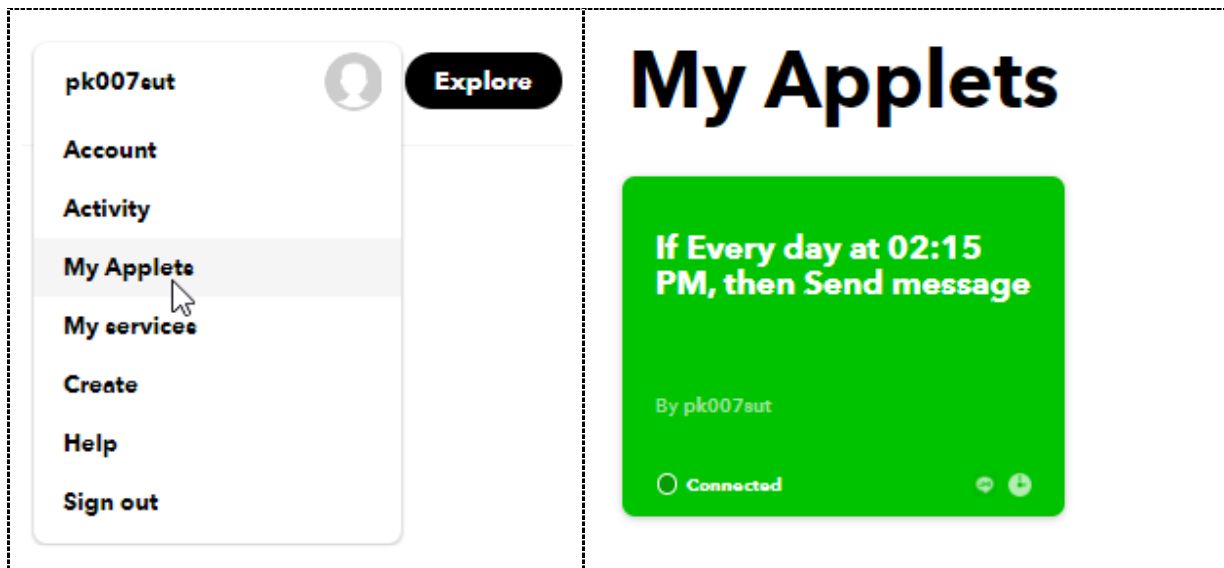
|   |  |
|---|--|
| <h1>if  then  that</h1> |  |
|    |  <h2>Connect LINE</h2> <p>Step 3 of 6</p> <p>LINE is a global messaging app used in over 230 countries and regions. LINE offers fun and free voice, video, and chat communication across multiple platforms. Receive event notifications from LINE Notify official account.</p> <p><b>Connect</b></p> |

|   |  |
|---|--|
|  <p>LINE - Google Chrome<br/>access.line.me/dialog/oauth/weblogin?response_type=code&amp;client_id...</p> <p><b>LINE</b></p> <p>อีเมล <input type="text"/></p> <p>รหัสผ่าน <input type="password"/></p> <p><b>เข้าสู่ระบบ</b></p> <p>เกี่ยวกับ LINE   © LINE Corporation</p> |  <p><b>IFTTT</b><br/>IFTTT, Inc.</p> <p>ระบบจะส่งข้อโปรไฟล์และรายชื่อห้องแชทไปยังผู้ให้บริการ IFTTT คุณสามารถยกเลิกการเชื่อมต่อได้ที่ "หน้าของฉัน" ใน LINE Notify</p> <p>เมื่อเห็นด้วย บัญชีทางการ LINE ของ "LINE Notify" จะถูกเพิ่มให้เป็นเพื่อนกับคุณ</p> <p><b>ยกเลิก</b> <b>เห็นด้วยและเชื่อมต่อ</b></p> |
| <p><b>Send message</b></p> <p>This Action will post a message to LINE.</p>  | <p><b>Recipient</b></p> <p>รับการแจ้งเตือนแบบตัวต่อตัว <input type="text"/></p> <p>Message destination</p> <p><b>Message</b></p> <p>Test at{{CheckTime}}</p> <p><b>Add ingredient</b></p> <p><b>Photo URL</b></p> <p><input type="text"/></p> <p><b>Add ingredient</b></p> <p><b>Create action</b></p> <p>กำหนดข้อความที่ต้องการส่ง ในช่อง Message</p>   |

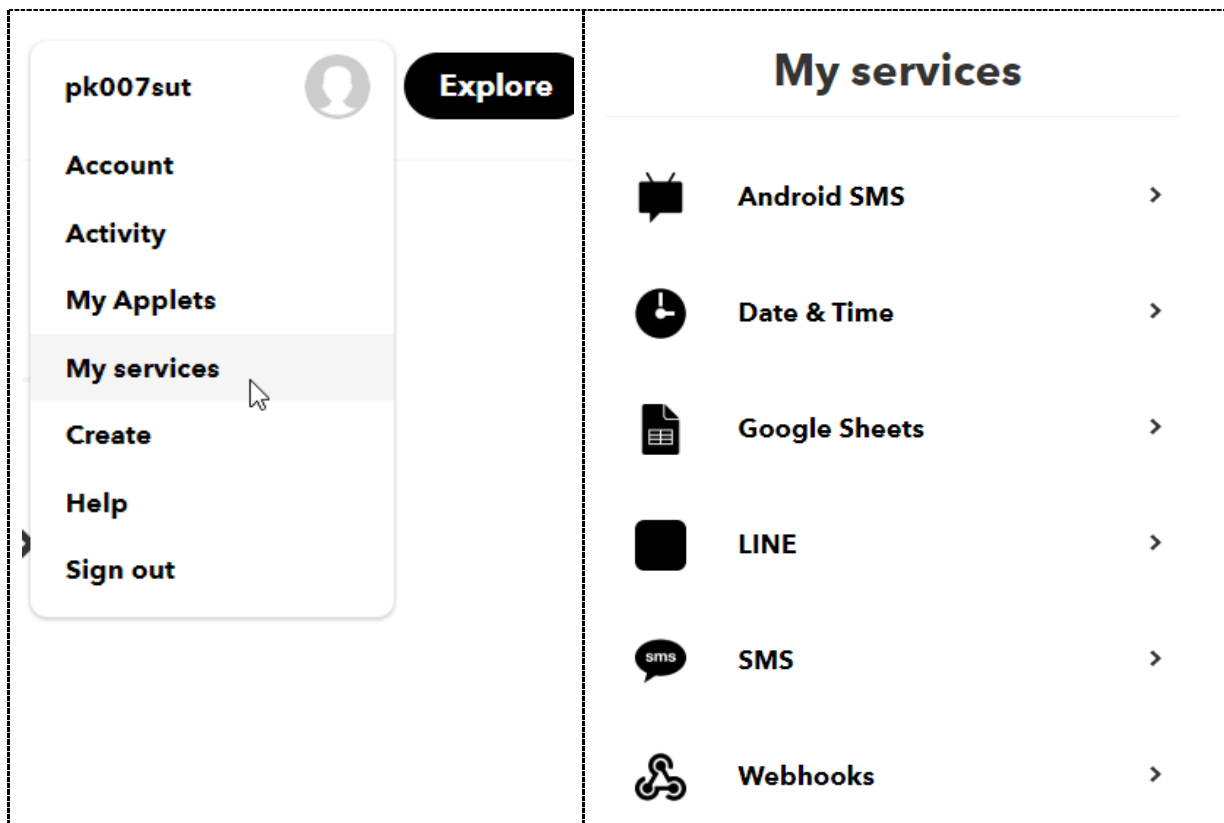
|   |  |
|---|--|
| <h2>Review and finish</h2> <p>Step 6 of 6</p> <div></div> <div><p>If Every day at 02:15 PM, then<br/>Send message</p><p>by pk007sut <span>43/140</span></p></div> <div><p>Receive notifications<br/>when this Applet runs <input checked="" type="checkbox"/></p></div> <div><p><b>Finish</b></p></div> | <div></div> <div><p><b>If Every day at 02:15<br/>PM, then Send<br/>message</b></p><p>By pk007sut</p></div> <div><h3>Applets</h3><p>Applets connect two or more services together<br/>and help you do something that you couldn't do<br/>with just one service alone. <a href="#">Learn more</a></p><p><b>Got it</b></p></div> <div><p><b>Connected</b> </p></div> |
|---|--|

2.4 เมื่อถึงเวลาที่กำหนด จะมีข้อความที่ตั้งไว้ส่งเข้า LINE

2.5 ตรวจสอบ APPET ที่มี และตรวจสอบ Service ที่เปิดใช้



2.6 ตรวจสอบ Service ที่เปิดใช้





## การทดลองที่ 4 WebHooks to IFTTT – Social Alert

3. IFTTT Test3 → Trigger from Input

3.1 ตั้งค่า IFTTT → If **WebHooks** Then **LINE**

## if this then that



**Receive a web request**

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Event Name

The name of the event, like "button\_pressed" or "front\_door\_opened"

**Create trigger**

Event\_Name = Test\_Key

## if then that



**Recipient**

Message destination

**Message**

Value 1:  <br>  
 Value 2:  <br>  
 Value 3:

**Add ingredient**

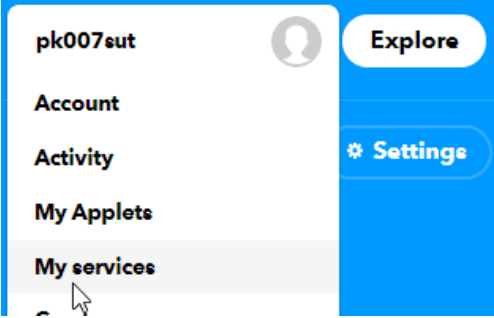


**Photo URL**

**Add ingredient**

**Create action**

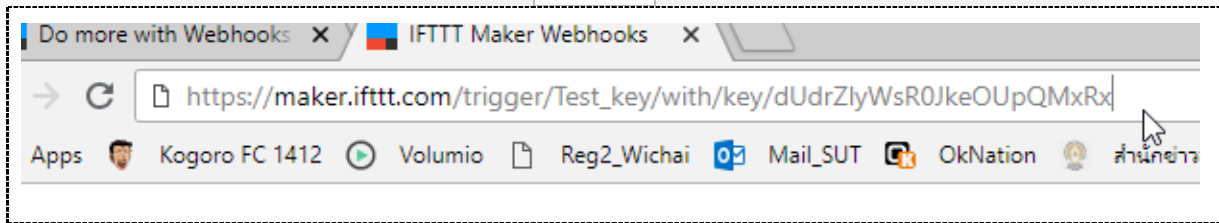
Add ingredient 3 ค่า คือ Value1, Value2 และ Value3

## 3.2 ทดสอบการทำงานของ WebHook

|   |  |
|---|--|
|    | คลิก My Service  |
|    | คลิก WebHooks  |
|   | คลิก Documentation   |
| <p>Your key is: dUdrZlyWsR0JkeOU[REDACTED]Rx <b>Key</b></p> <p>◀ Back to service</p> <p><b>To trigger an Event</b></p> <p>Make a POST or GET web request to: <b>Event_Name</b></p> <p><code>https://maker.ifttt.com/trigger/[Test_Key]/with/key/dUdrZlyWsR0JkeOUqHkRx</code></p> <p>With an optional JSON body of: <b>Payload</b></p> <p><code>{ "value1" : "[REDACTED]", "value2" : "[REDACTED]", "value3" : "[REDACTED]" }</code></p> <p>The data is completely optional, and you can also pass value1, value2, and value3 as query parameters on to the Action in your Recipe.</p> <p>You can also try it with <code>curl</code> from a command line <b>Curl command line</b></p> <p><code>curl -X POST https://maker.ifttt.com/trigger/Test_Key/with/key/dUdrZlyWsR0JkeOUqHkRx</code></p> <p><b>Test It</b></p> | <p>ป้อน Test_Key ที่ Event_Name</p> <p>ป้อน 123 ที่ Value1</p> <p>คลิก Test It</p> |
| รอสักครู่จะมี SMS ส่งเข้าที่ LINE   |  |

3.3 หรือจะทดสอบด้วย Web browser!!! ด้วยการพิมพ์ URL ก็สามารทำได้

[https://maker.ifttt.com/trigger/Test\\_Key/with/key/dUdrZlyWsR0JkeOUxQMxRx](https://maker.ifttt.com/trigger/Test_Key/with/key/dUdrZlyWsR0JkeOUxQMxRx)



3.4 โปรแกรม ESP32 ตามตัวอย่างแต่อย่าลืมแก้ไข 4 อย่าง คือ Wifi\_SSID, Wifi\_Password, IFTTT API Key, IFTTT Event Name

- #define WIFI\_SSID "SUT\_IoTs"
- #define WIFI\_PASS "MaiMeeJingJing"
- #define IFTTT\_API\_KEY "zcUfCzbSuczozvzOzvgFvsdgDFffs"
- #define IFTTT\_EVENT\_NAME "Test\_Key"

```
#include <WiFi.h>
#include <HTTPClient.h>

#define WIFI_SSID "SUT_IoTs"
#define WIFI_PASS "MaiMeeJingJing"
#define WebHooksKey "zcUfCzbSuczozvzOzvgFvsdgDFffs"
#define WebHooksEventName "Test_Key"

#define testSwitch 0 // D0

void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  Serial.println("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
  pinMode(testSwitch, INPUT_PULLUP);
  randomSeed(analogRead(33));
}

void loop() {
  if (digitalRead(testSwitch) == LOW) {
    String serverName = "http://maker.ifttt.com/trigger/" + String(WebHooksEventName) + "/with/key/" + String(WebHooksKey);
    String httpRequestData = "value1=" + String(random(100)) + "&value2=" + String(random(200)) + "&value3=" + String(random(300));
    Serial.println("Server Name : " + serverName);
    Serial.println("json httpRequestData : " + httpRequestData);
    if (WiFi.status() == WL_CONNECTED) {
      HTTPClient http;
      http.begin(serverName);
      http.addHeader("Content-Type", "application/x-www-form-urlencoded");
      int httpResponseCode = http.POST(httpRequestData);
      Serial.print("HTTP Response code: ");
      Serial.println(httpResponseCode);
      http.end();
    }
  }
}
```

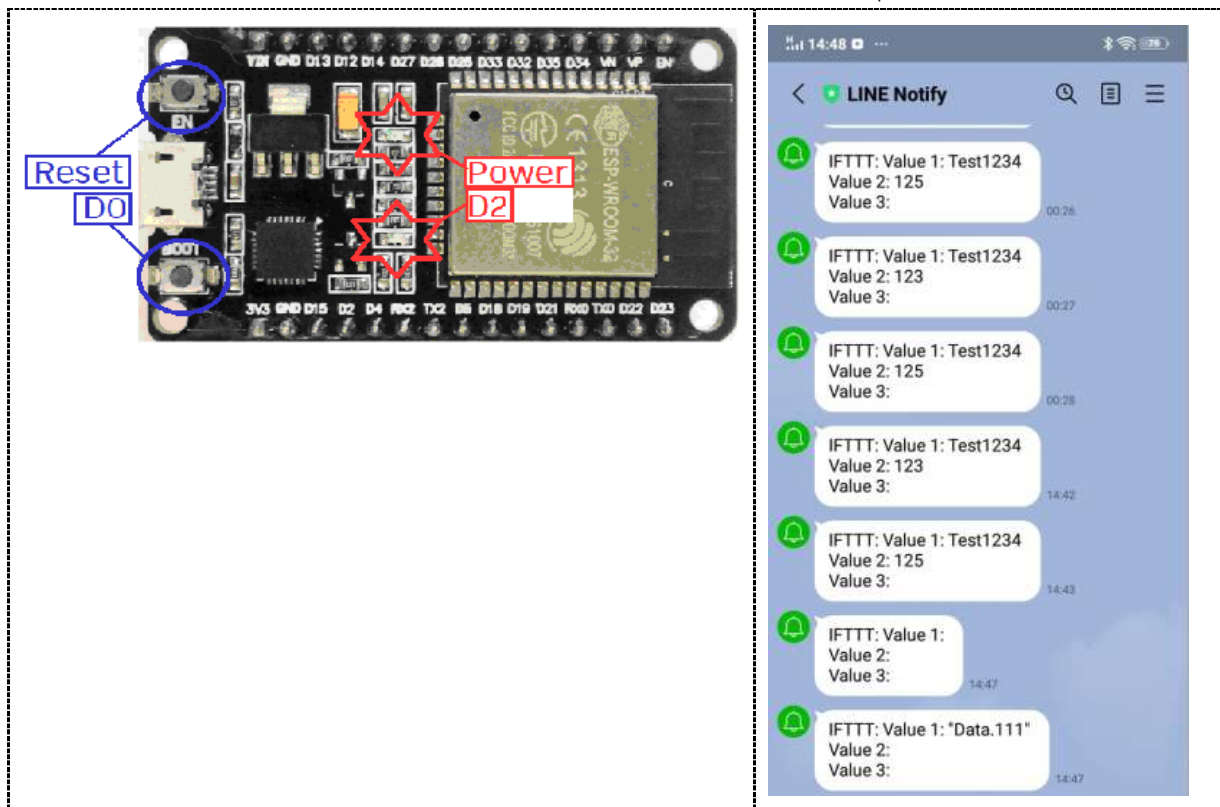
```

if (httpResponseCode == 200)
  Serial.println("Successfully sent");
else
  Serial.println("Failed!");
}
else {
  Serial.println("WiFi Disconnected");
}

Serial.print(" >> Wait for 10 Sec --> ");
for (int i = 9; i >= 0; i--) {
  Serial.print(i);
  delay(1000);
}
Serial.println(" >> Ready");
}
}

```

### 3.5 ทดสอบการทำงานของระบบ เมื่อระบบต่อไฟได้แล้วให้กดปุ่ม BOOT หรือ DO



```

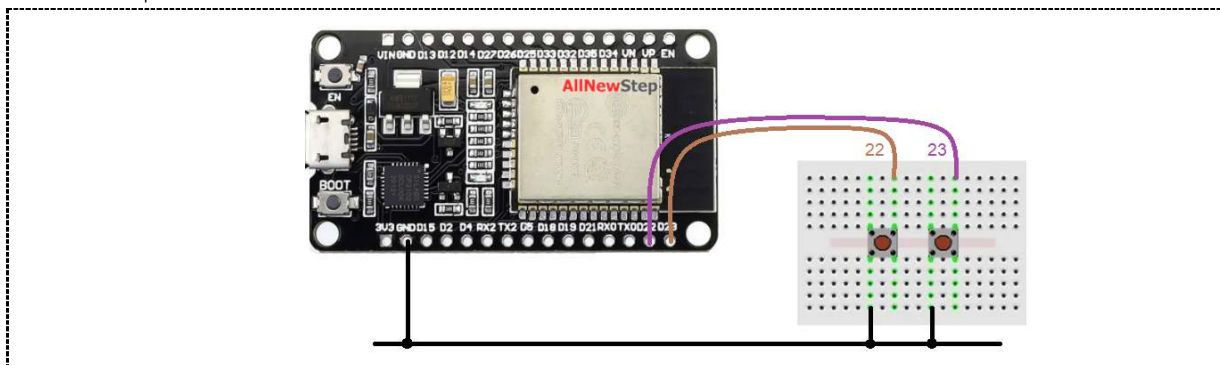
Connecting
.....
Connected to WiFi network with IP Address: 192.168.1.30
Server Name :http://maker.ifttt.com/trigger/Test_Key/with/key/cUfCbSucovOzD3WCxCxHod
json httpRequestData :value1=76&value2=72&value3=121
HTTP Response code: 200
Successfully sent
>> Wait for 10 Sec --> 9876543210 >> Ready
Server Name :http://maker.ifttt.com/trigger/Test_Key/with/key/cUfCbSucovOzD3WCxCxHod
json httpRequestData :value1=61&value2=78&value3=238
HTTP Response code: 200
Successfully sent
>> Wait for 10 Sec --> 9876543210 >> Ready

```

### Quiz-303 – Social Alert

ทดสอบการส่งข้อมูลไป ☐ LINE สำหรับสวิตช์กด 2 ตัว

- กดปุ่ม B ที่ต่อกับ ESP32- ให้ส่งข้อความ “Door Open Alarm”
- กดปุ่ม C ที่ต่อกับ ESP32- ให้ส่งข้อความ “Intruders Alarm”



## การทดลองที่ 5 WebHooks to IFTTT – Google Spread Sheet

## 4. IFTTT Test4 → Triger from Sensor

## 4.1 ทดสอบ ESP32 + DHT22

**DHT sensor library for ESPx**by beegee\_tokyo Version 1.17.0 **INSTALLED****Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors** Optimized libra  
changes: Fix negative temperature problem (credits @helijunky)[More info](#)

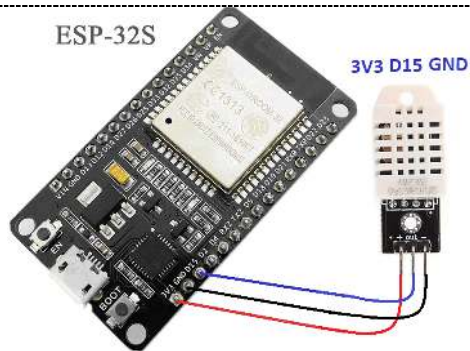
- ให้แน่ใจว่าลง DHT Sensor Library for ESPx by beegee\_tokyo Version 1.17.0

- ต่อวงจร DHT22 to D15

DHT22 (+, out, -)

ESP32 (3V3, D15, GND)

โหลดโปรแกรมทดสอบ



```
#include "DHTesp.h"
#define Pin_DHT22 15
```

```
DHTesp dht;
void setup()
{ Serial.begin(115200);
  dht.setup(Pin_DHT22, DHTesp::DHT22);
}
```

```
void loop()
{ float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();
  Serial.print("\nTemperature('C) = ");
  Serial.print(temperature, 1);
  Serial.print("\tHumidity(%) = ");
  Serial.print(humidity, 1);
  delay(5000);
}
```

4.2 ตั้งค่า IFTTT → If WebHooks Then Google Spreadsheet

if  this then that



Complete trigger fields



Receive a web request

the Maker service receives a web request to notify it of an event. For information on trigger fields, see the [Webhooks](#) page. For information on trigger fields, see the [Webhooks](#) page.

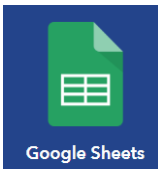
Event Name

test\_GSheet

The name of the event, like "button\_pressed" or "front\_door\_opened"

Event\_Name = test\_GSheet

if  then  that



Add row to spreadsheet

This action will add a single row to the bottom of the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created after 2000 rows.



| <h2>Add row to spreadsheet</h2> <p>This action will add a single row to the bottom of the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created after 2000 rows.</p> <h3>Spreadsheet name</h3> <div>IFTTT_Maker_Webhooks_Events</div> <p>Will create a new spreadsheet if one with this title doesn't exist</p> <h3>Formatted row</h3> <div>OccurredAt     EventName     Value1     Value2     Value3</div> <p>Use "   " to separate cells</p> <h3>Drive folder path</h3> <div>IFTTT/MakerWebhooks/EventName</div> <p>Format: some/folder/path (defaults to "IFTTT")</p> <div>Create action</div> | Sheet Name |
|--|------------|
| Add ingredient   |            |
| Column Data  |            |
| Add ingredient   |            |
| Path   |            |
| Add ingredient   |            |
| Create Action  |            |

#### 4.3 ทดสอบการทำงานของระบบผ่าน Web URL ของ maker.ifttt.com ตาม EventName ที่กำหนด

Make a POST or GET web request to:

`https://maker.ifttt.com/trigger/test_GSheet/with/key/dUdrZlyWsR0JkeOUpQMxRx`

With an optional JSON body of:

`{ "value1" : " 123 ", "value2" : " 456 ", "value3" : " 7890 " }`

The data is completely optional, and you can also pass `value1`, `value2`, and `value3` passed on to the Action in your Recipe.

You can also try it with `curl` from a command line.

```
curl -X POST -H "Content-Type: application/json" -d '{"value1":"123","value2":"456","value3":"7890"}' https://maker.ifttt.com/trigger/test_GSheet/with/key/dUdrZlyWsR0JkeOUpQMxRx
```

Test It

ใครฟิของฉัน > IFTTT > MakerWebhooks > test\_GSheet ▾

ชื่อ ↑

เจ้าของ



IFTTT\_Maker\_Webhooks\_Events

ฉัน

|                  |             |     |     |      |
|------------------|-------------|-----|-----|------|
| July 31, 2018 at | test_GSheet | 123 | 456 | 789  |
| July 31, 2018 at | test_GSheet | 123 | 456 | 789  |
| July 31, 2018 at | test_GSheet | 123 | 456 | 789  |
| July 31, 2018 at | test_GSheet | 123 | 456 | 7890 |
| July 31, 2018 at | test_GSheet | 123 | 456 | 7890 |

## 4.4 โปรแกรม ESP32

```

#include <WiFi.h>
#include <HTTPClient.h>
#include "DHTesp.h"

#define WIFI_SSID "SUT_IoTs"
#define WIFI_PASS "MaiMeeJingJing"
#define WebHooksKey "zcUfCzbSuczozvzOzvgFvsdgDFffs"
#define WebHooksEventName "test_GSheet"
#define My_NAME "F3204-ServerRoom"

#define Pin_DHT22 15

DHTesp dht;

void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  Serial.println("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
  dht.setup(Pin_DHT22, DHTesp::DHT22);
}

void loop() {
  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();
  Serial.println();
  Serial.print("\nTemperature('C) = ");
  Serial.print(temperature, 1);
  Serial.print("\nHumidity(%) = ");
  Serial.print(humidity, 1);
  String serverName = "http://maker.ifttt.com/trigger/" + String(WebHooksEventName) + "/with/key/" +
String(WebHooksKey);
  String httpRequestData = "value1=" + String(My_NAME) + "&value2=" + String(temperature) + "&value3=" +
String(humidity);
  Serial.println();
  Serial.println("Server Name >> " + serverName);
  Serial.println("json httpRequestData >> " + httpRequestData);
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(serverName);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    int httpResponseCode = http.POST(httpRequestData);
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    http.end();
    if (httpResponseCode == 200)
      Serial.println(" --> Successfully sent");
    else
      Serial.println(" --> Failed!");
  }
  else {
    Serial.println("WiFi Disconnected");
  }
  int WaitTime = 60;
  Serial.print(" >> Wait for next time --> ");
  for (int i = WaitTime; i >= 0; i -= 5) {
    Serial.print(",");
    Serial.print(i);
    delay(5000);
  }
}
}

```

```

....
Connected to WiFi network with IP Address: 192.168.1.30

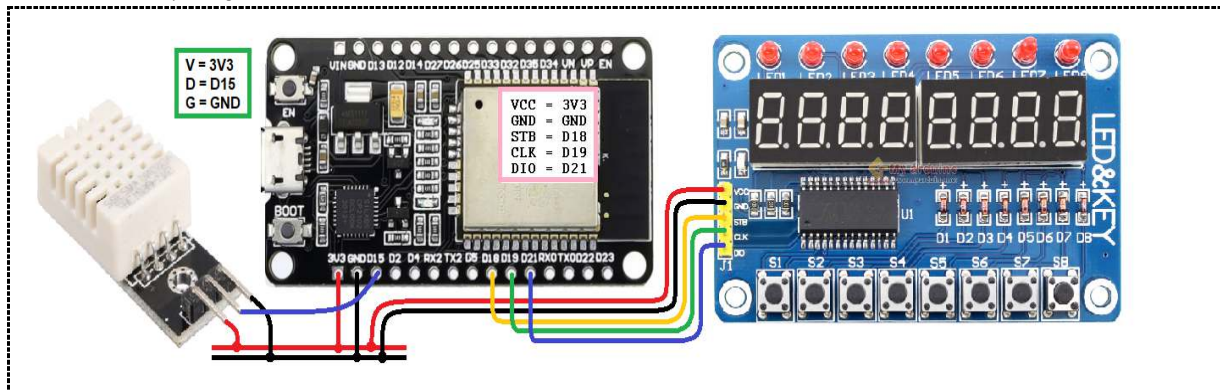
Temperature('C) = 27.5 Humidity(%) = 46.9
Server Name :http://maker.ifttt.com/trigger/test_GSheet/with/key/cUfCbSucovOzD3WCxCxHod
json httpRequestData :value1 = B3601234 Mr.Wichai Srisuruk & value2 = 27.50 & value3 = 46.90
HTTP Response code: 200
--> Successfully sent
>> Wait for next time --> ,60,55,50,45,40,35,30,25,20,15,10,5,0

Temperature('C) = 27.5 Humidity(%) = 47.0
Server Name :http://maker.ifttt.com/trigger/test_GSheet/with/key/cUfCbSucovOzD3WCxCxHod
json httpRequestData :value1 = B3601234 Mr.Wichai Srisuruk & value2 = 27.50 & value3 = 47.00
HTTP Response code: 200
--> Successfully sent
>> Wait for next time --> ,60,55

```

### Quiz\_304 – Data Logger and Social Alarm

- ส่งข้อมูลอุณหภูมิไปยัง Google Spreadsheet (ทำแล้วในข้อ QB4)
- หากอุณหภูมิที่อ่านได้เกิน 28°C ให้แจ้งเตือนผ่าน \_\_\_\_ และบอกด้วยว่าอุณหภูมิเท่าใด  
☐ SMS, ☐ FB Page, ☐ FB Massager, ☐ Twitter, ☒ LINE
- แสดงอุณหภูมิที่ 7\_Segment Display TM1638 Board

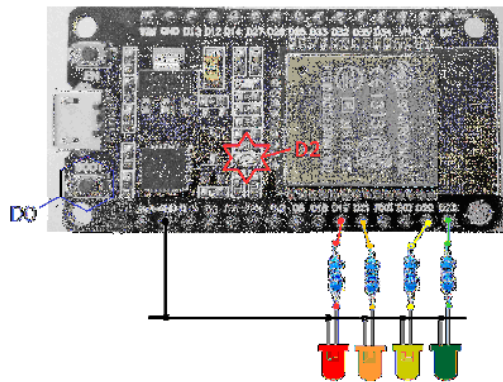


แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต  
IoT Approaches to Manufacturing System

ชื่อ-สกุล :

3/3. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz\_301 – 4 External LED Control



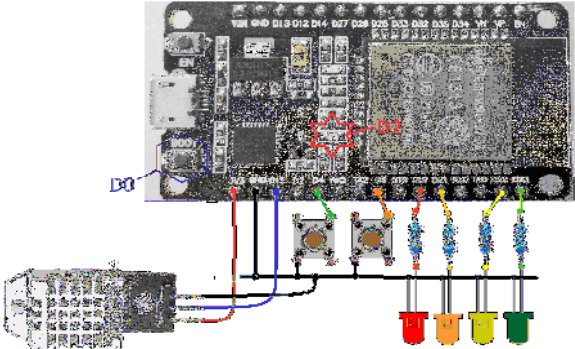
< Test Code >

รูปหน้าจอ Blynk

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

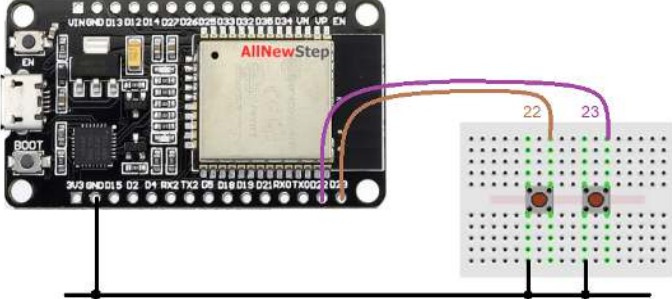
Quiz\_302 – DHT22 + 4 LED + 2 Switch

|   |
|---|
|  |
| < Test Code >   |
| รูปหน้าจอ Blynk   |
| รูปการต่อวงจร – 1   |
| รูปการต่อวงจร – 2   |

**Quiz\_303 – Social Alert**

ทดสอบการส่งข้อมูลไป ☐ LINE สำหรับสวิตช์กด 3 ตัว

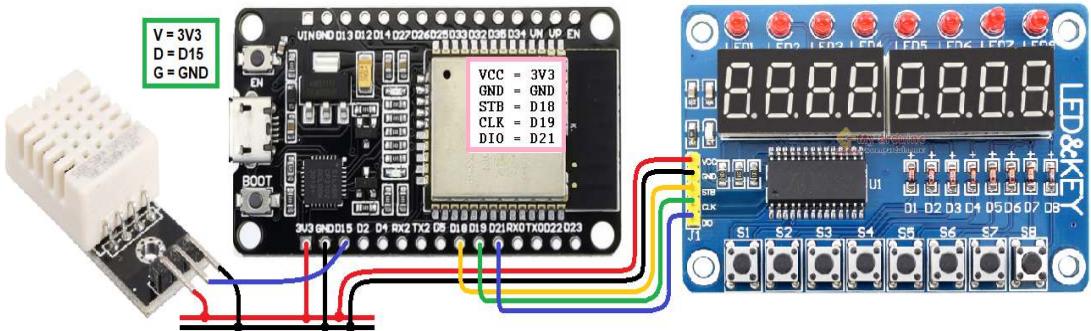
- กดปุ่ม B ที่ต่อกับ ESP32- ให้ส่งข้อความ “Door Open Alarm”
- กดปุ่ม C ที่ต่อกับ ESP32- ให้ส่งข้อความ “Intruders Alarm”

|  |
|--|
|  |
| < Test Code >  |
| รูปการต่อวงจร – 1  |
| รูปการต่อวงจร – 2  |
| รูปหน้าจอ LINE ผลการทดสอบ  |



## Quiz\_304 – Data Logger and Social Alarm

- ส่งข้อมูลอุณหภูมิไปยัง Google Spreadsheet (ทำแล้วในข้อ QB4)
- หากอุณหภูมิที่อ่านได้เกิน 28°C ให้แจ้งเตือนผ่าน \_\_\_\_ และบอกด้วยว่าอุณหภูมิเท่าใด  
☐ SMS, ☐ FB Page, ☐ FB Messenger, ☐ Twitter, ☒ LINE
- แสดงอุณหภูมิที่ 7\_Segment Display TM1638 Board



< Test Code >

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

รูปหน้าจอ LINE ผลการทดสอบ