

ใบงานที่ 17

การเชื่อมต่อ และควบคุมอุปกรณ์ IoT ผ่านอินเทอร์เน็ตด้วยแพลตฟอร์ม Blynk IoT

จุดประสงค์การเรียนรู้

1. ศึกษาการเปิด/ปิดไฟบ้านผ่านอินเทอร์เน็ตบนแพลตฟอร์ม Blynk

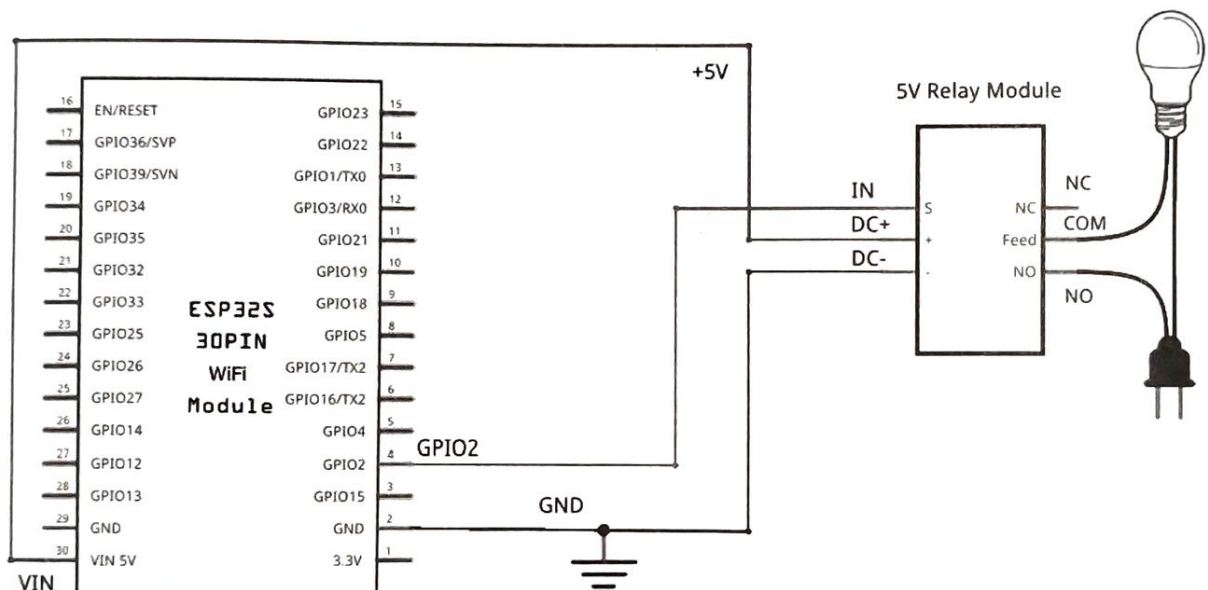
เครื่องมือและอุปกรณ์การทดลอง

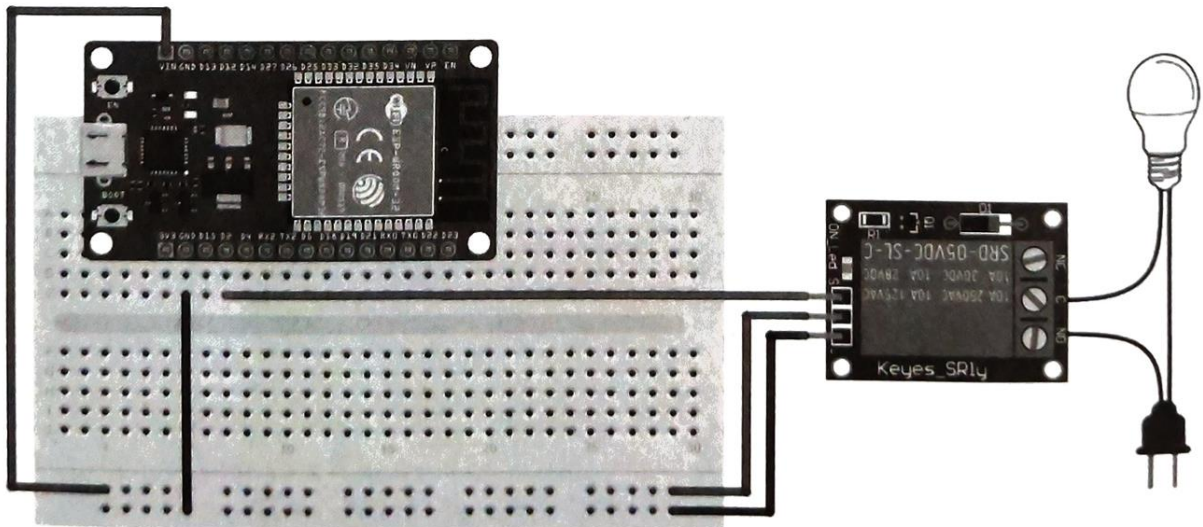
1. เครื่องไมโครคอมพิวเตอร์
2. บอร์ดไมโครคอนโทรลเลอร์ ESP32
3. โปรแกรมการทดลอง
4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลองที่ 1 การเปิด/ปิดไฟบ้านผ่านอินเทอร์เน็ตบนแพลตฟอร์ม Blynk

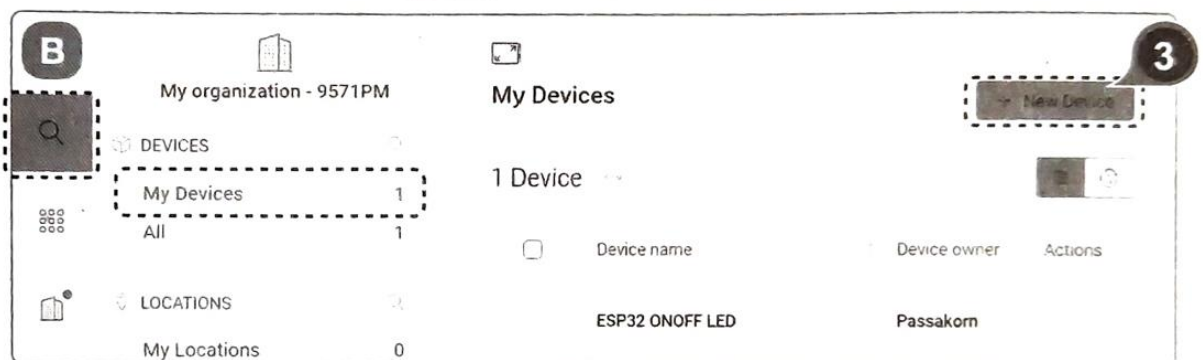
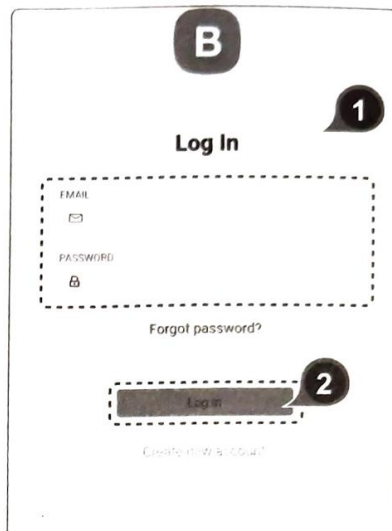
1. บอร์ด NodeMCU ESP32
2. โมดูลรีเลย์
3. ชุดหลอดไฟ ขั้วหลอด และปลั๊กไฟ 220 VAC
4. แผงต่อวงจร
5. สายไฟต่อวงจร

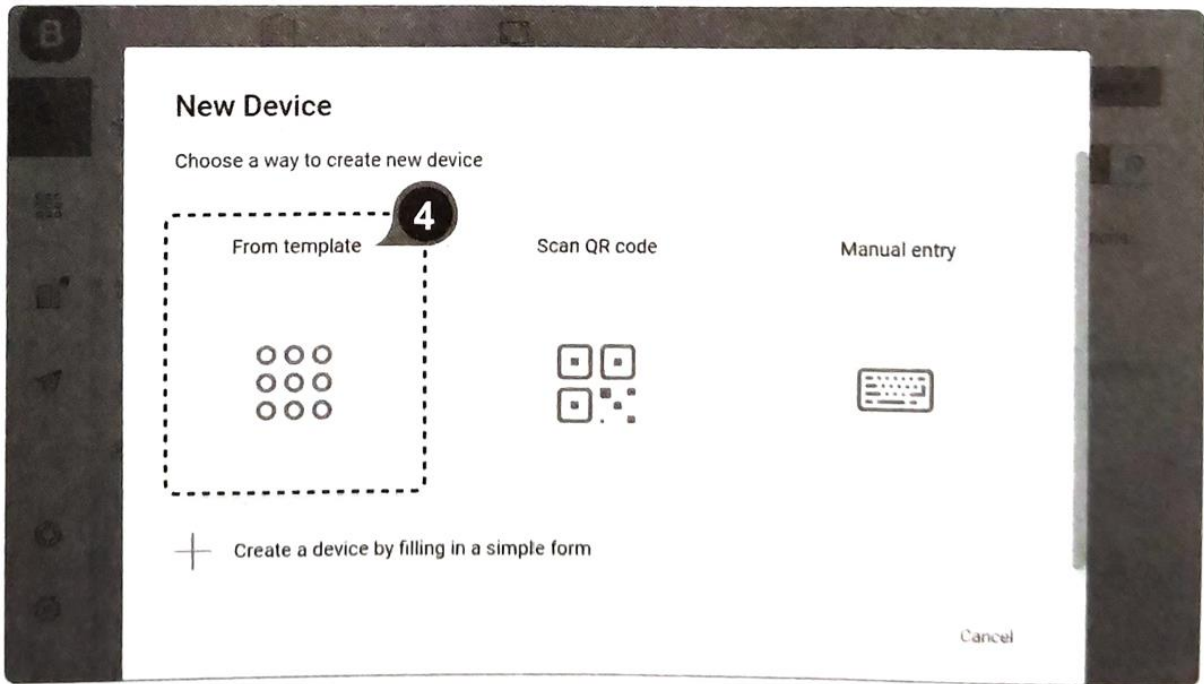
ประกอบวงจรตามรูป



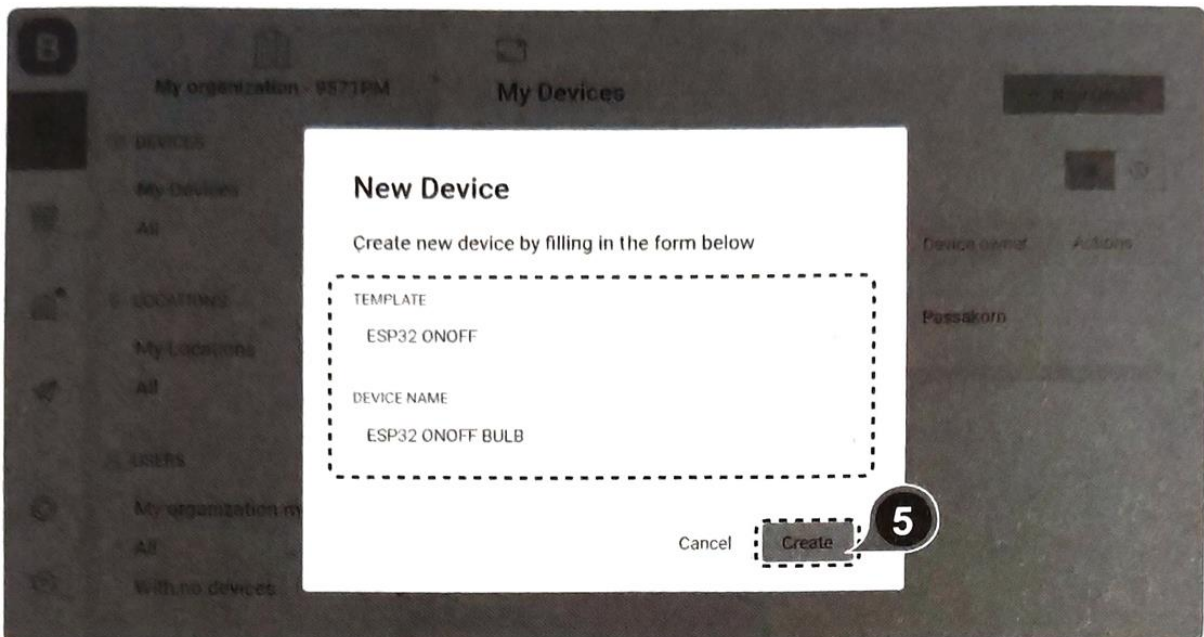


1. ไปที่ <https://blynk.cloud/dashboard/login> เพื่อเปิด หน้า Log In ของ Blynk.Console ขึ้นมา ดังรูป
2. กรอกอีเมลและพาสเวิร์ด ที่ลงทะเบียนไว้แล้วให้เรียบร้อย แล้วคลิกปุ่ม Log In
3. ที่ Main Menu คลิกเมนู Search » My Devices แล้ว คลิกที่ปุ่ม + New Device ดังรูป เพื่อเพิ่มอุปกรณ์ (ในที่นี้ เราจะไม่สร้าง Template เพิ่ม แต่เราจะใช้วิธีเพิ่ม Device ที่เป็นการเปิด/ปิดไฟบ้านด้วยรีเลย์เข้าไปใน Template เดิมที่เรามีอยู่ เพื่อจัดให้เป็นกลุ่มของอุปกรณ์ประเภทเดียวกัน นั่นคือ Switch ข้อดีของการเพิ่ม Device ไว้ใน Template เดียวกันคือ หากต้องการเปลี่ยนแปลงการตั้งค่าใดๆ จะ สามารถอัปเดตการตั้งค่านั้น ให้กับอุปกรณ์ทุกตัวที่อยู่ภายใต้Template เดียวกันได้พร้อมกันในคราวเดียว)

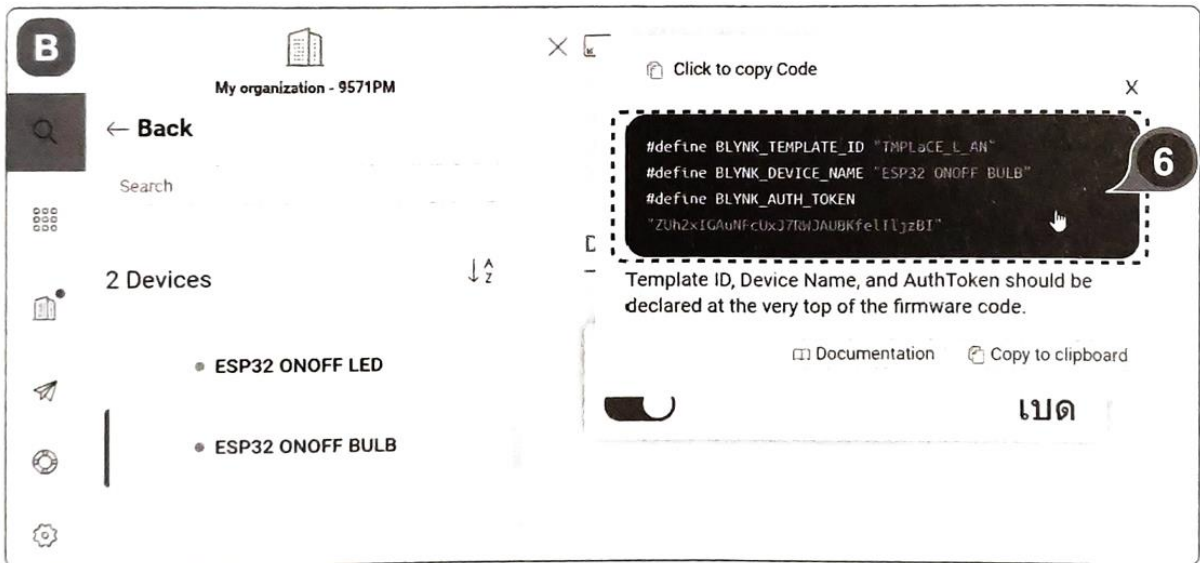




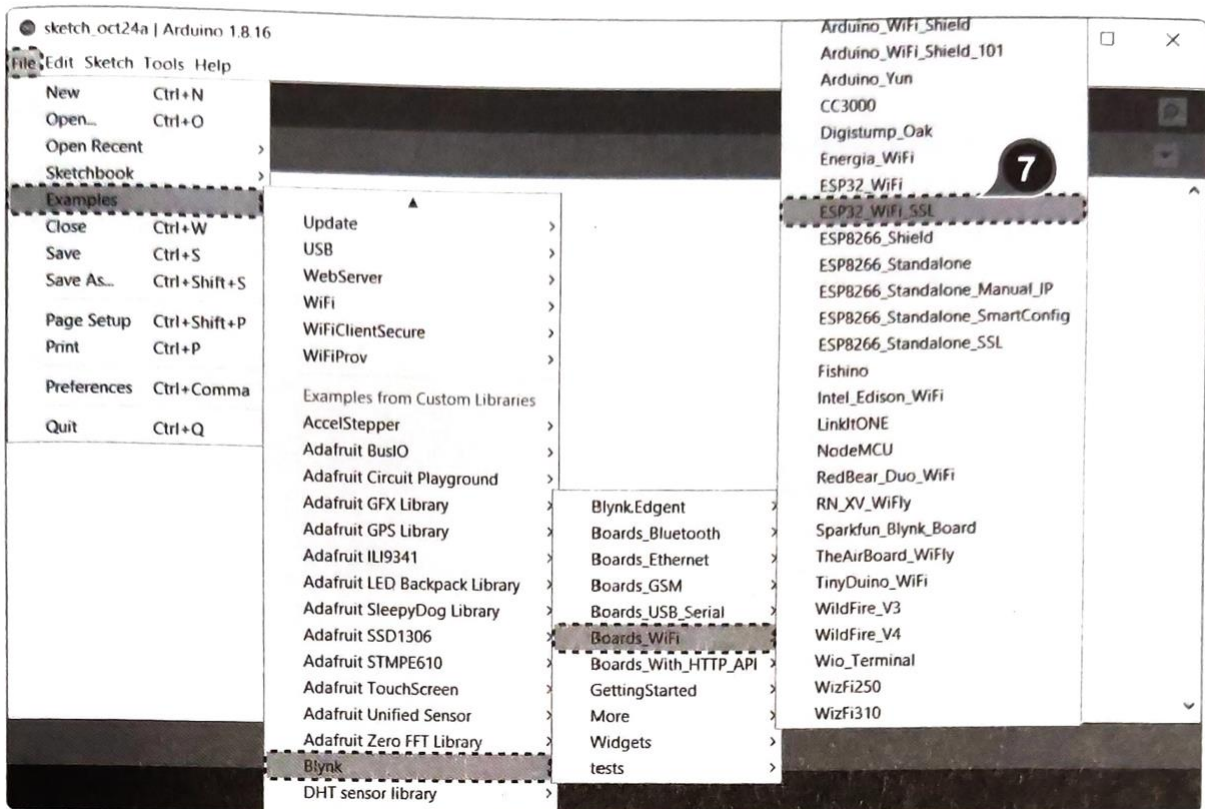
4. ในหน้า New Device คลิกปุ่ม From template เพื่อเลือกที่จะสร้างอุปกรณ์ใหม่โดยเลือกจาก Template



5. คลิกเลือกชื่อ Template ที่ต้องการ (ในที่นี้คือเทมเพลตเดิม ESP32 ONOFF) และตั้งชื่อให้กับอุปกรณ์ (ในที่นี้คือ ESP32 ONOFF BULB) จากนั้นคลิกปุ่ม Create



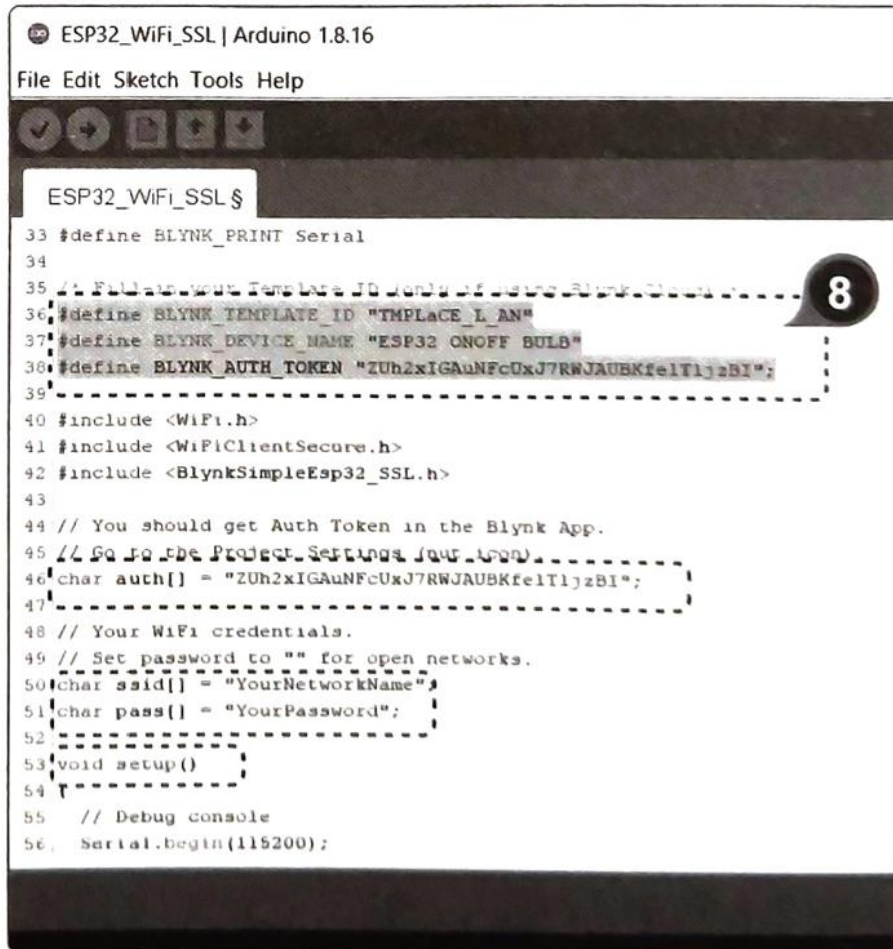
6. ที่หน้า My Devices จะเห็นอุปกรณ์ที่ถูกสร้างเพิ่มขึ้นมา (ในที่นี้คือ ESP32 ONOFF BULB) ทางด้านขวา จะเห็นกรอบโค้ดที่ระบบได้อำนวยความสะดวก โดยได้สร้างโค้ดพร้อมใส่ข้อมูลและรหัสที่จำเป็นต้องใช้ทั้ง Template ID, Device Name และ Auth Token ไว้ให้เรียบร้อยแล้ว ให้เรานำมาใส่ไว้ในกรอบโค้ด แล้วคลิกเพื่อ Copy Code เอาไว้ จากนั้นหันไปลงมือต่อวงจรควบคุมการเปิด/ปิดไฟบ้าน ตามรูปผังการ ต่อวงจรที่ 1 และ 2 อาจจะต้องเพิ่มจากการเปิด/ปิดหลอดไฟ LED ในตัวอย่างที่แล้วก็ได้ เพราะเลือกใช้ ดิจิตอลเอาต์พุตคนละขากัน



7. เปิด Arduino IDE ขึ้นมา ในที่นี้เราจะใช้โค้ดโปรแกรมจากไฟล์ตัวอย่างที่ให้มาพร้อมกับ Blynk Libraries ในตอนติดตั้ง โดยคลิกที่เมนู File > Examples > Blynk > Boards_WiFi > ESP32_WIFI_SSL

หมายเหตุ

หากไม่สามารถเชื่อมต่อกับ Blynk. Cloud ได้ โดยในหน้าต่าง Serial Monitor แสดงข้อความว่า Secure connection failed ให้กลับมา เปลี่ยนไฟล์ตัวอย่างในขั้นตอนนี้เป็น ESP32_WiFi แทน หรือจะ แก้บรรทัดคำสั่งของไฟล์เดิม โดย ให้เปลี่ยนจาก WiFiClientSecure.h และ BlynkSimpleEsp32_SSL.h ไปเป็น WiFiClient.h และ Blynk SimpleEsp32.h ก็ได้



```

ESP32_WiFi_SSL | Arduino 1.8.16
File Edit Sketch Tools Help

ESP32_WiFi_SSL §
33 #define BLYNK_PRINT Serial
34
35 // Fill in these Template ID, Device Name and Auth Token
36 #define BLYNK_TEMPLATE_ID "TMPLaCE_L_AN"
37 #define BLYNK_DEVICE_NAME "ESP32 ONOFF BULB"
38 #define BLYNK_AUTH_TOKEN "ZUh2xIGAUNFcUxJ7RWJAUBKfe1T1jzBI";
39
40 #include <WiFi.h>
41 #include <WiFiClientSecure.h>
42 #include <BlynkSimpleEsp32_SSL.h>
43
44 // You should get Auth Token in the Blynk App.
45 // Go to the Project Settings (App -> About)
46 char auth[] = "ZUh2xIGAUNFcUxJ7RWJAUBKfe1T1jzBI";
47
48 // Your WiFi credentials.
49 // Set password to "" for open networks.
50 char ssid[] = "YourNetworkName";
51 char pass[] = "YourPassword";
52
53 void setup()
54 {
55 // Debug console
56 Serial.begin(115200);

```

8. วาง (Paste) ข้อมูลโค้ด Template ID, Device Name และ Auth Token ที่ Copy มา ลงไปแทนที่โค้ดเดิมตรงตำแหน่งดังรูป

9. ใส่ข้อมูลและแก้ไขโค้ดเพิ่มเติมตามตัวอย่าง เช่น Your AuthToken, YourNetworkName และ YourPassword อาจเปลี่ยนความเร็วในการอัปโหลดเป็น 115200 (หรือจะใช้เป็น 9600 ตามเดิมก็ได้) เวลาจะดูผลทางหน้าต่าง Serial Monitor ต้องกำหนดความเร็วนี้ให้ตรงกันด้วย เสร็จแล้วให้อัปโหลดโค้ด โปรแกรมลงบนบอร์ด (ในที่นี้คือ DOIT ESP32 DevKit V1)

```

#define BLYNK_PRINT Serial //รายงานสถานะการเชื่อมต่อของ Blynk ออก Serial Monitor

#define BLYNK_TEMPLATE_ID "TMPLaCE_L_AN"
#define BLYNK_DEVICE_NAME "ESP32 ONOFF BULB"
#define BLYNK_AUTH_TOKEN "ZUh2xIGAUNFcUxJ7RWJAUBKfe1T1jzBI";

```

```

#include <WiFi.h> //เรียกใช้ไลบรารี WiFi.h
#include <WiFiClientSecure.h> //เรียกใช้ไลบรารี WiFiClientSecure.h
#include <BlynkSimpleEsp32_SSL.h> //เรียกใช้ไลบรารี BlynkSimpleEsp32_SSL

char auth[] = "YourAuthToken"; //ก๊อปปี้รหัส Token ที่ได้รับมาใส่ลงใน " " เช่น char auth[]
= "ZUh2xIGAuNFcUxJ7RWJAUBKfelTljzBI";

char ssid[] = "YourNetworkName"; //ใส่ชื่อ SSID ของ WiFi Network ที่จะให้อุปกรณ์ IoT
เชื่อมต่อ ลงใน " "
char pass[] = "YourPassword"; //ใส่รหัสผ่านที่จะเชื่อมต่อกับ WiFi Network ลงใน " "

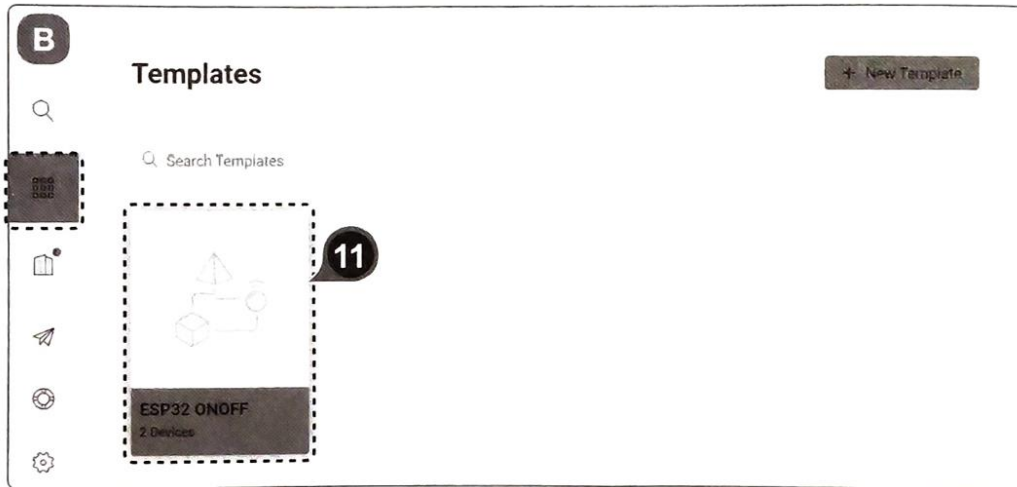
void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass); //เริ่มต้นกระบวนการเชื่อมต่อกับ WiFi Network
  และ Blynk Cloud
}

void loop()
{
  Blynk.run(); //ระบบ Blynk เริ่มการทำงาน
}

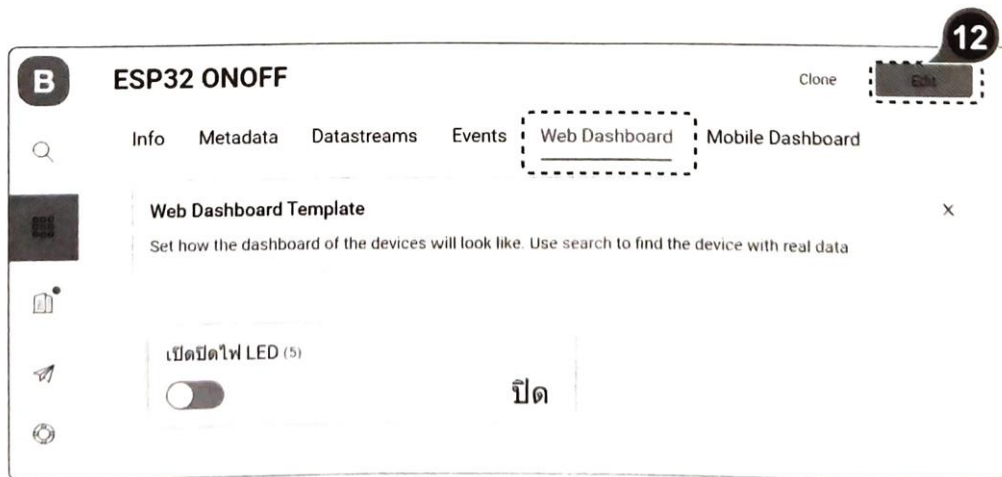
```

10. เปิดหน้าต่าง Serial Monitor ขึ้นมาตรวจสอบสถานะ ในการเชื่อมต่อ ถ้าสามารถเชื่อมต่อกับอินเทอร์เน็ตผ่านทาง WiFi และสามารถเชื่อมต่อกับ Blynk. Cloud ได้แล้ว จะได้รับหมายเลข IP Address และข้อความ Ready ดังรูป เป็นอันว่าเราสามารถ ที่จะเชื่อมต่อไปยังอุปกรณ์ IoT ผ่านทาง Blynk. Cloud ได้แล้ว

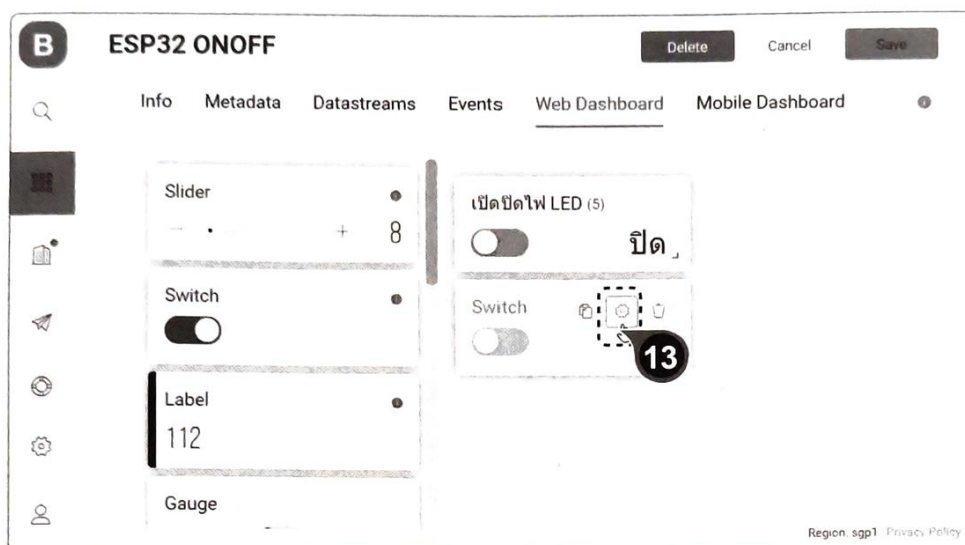
11. กลับไปที่ Blynk.Console เพื่อสร้าง Dashboard ที่ Main Menu คลิกเมนู Templates จากนั้นคลิก ที่ Templates ที่เราสร้างขึ้น (ในที่นี้คือ ESP32ONOFF)



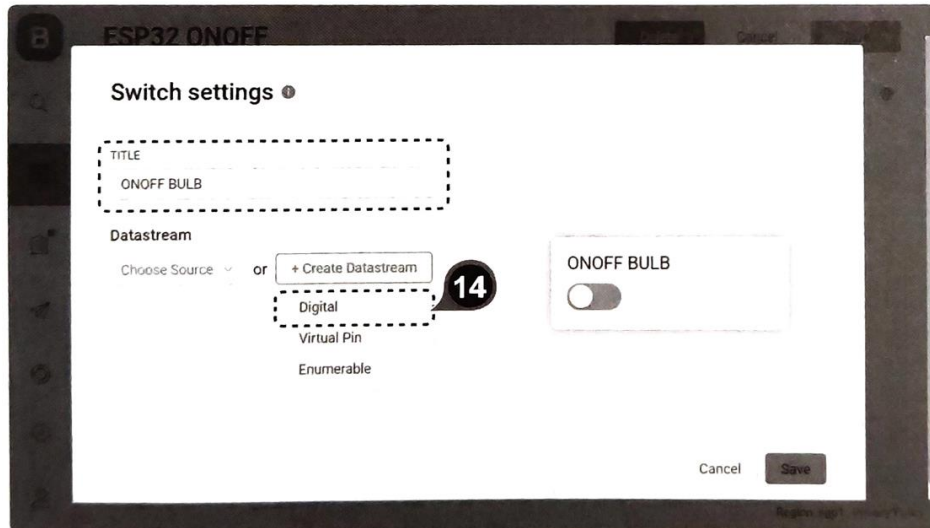
12. จากนั้นคลิกไปที่ Web Dashboard แล้วคลิกปุ่ม Edit



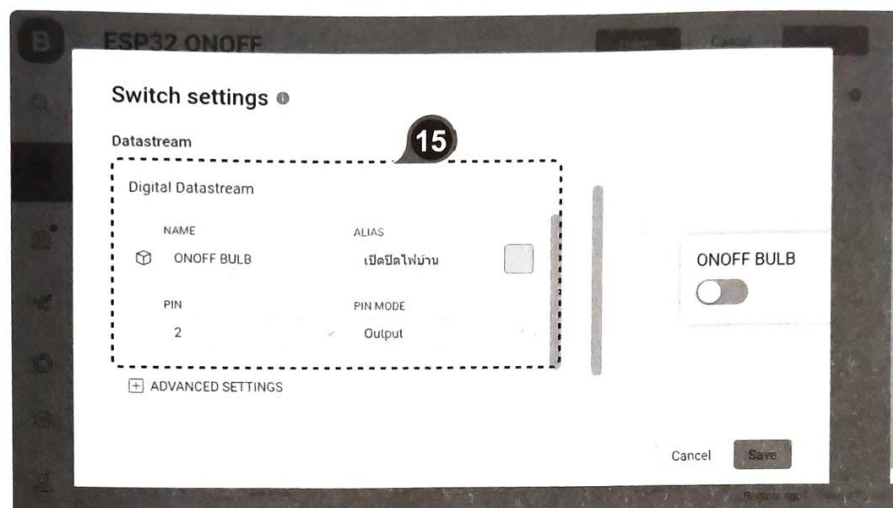
13. ดับเบิลคลิกหรือใช้เมาส์ลาก Widget Switch มาวางไว้บน Dashboard ทางขวามือ คลิกที่ตัวเลือก ⚙ ตั้งค่า เพื่อเข้าไปตั้งค่าการทำงานให้กับปุ่ม



14. ที่ TITLE ตั้งชื่อให้กับการทำงานของปุ่ม (ในที่นี้คือ ONOFF BULB) จากนั้นที่ Datastream สามารถ คลิกเลือกการตั้งค่า Datastream ที่เคยสร้างเอาไว้สำหรับใช้กับอุปกรณ์อื่นๆ ก่อนหน้านี้นี้ได้ แต่ในที่นี้ ต้องการสร้าง Datastream เพื่อการตั้งค่าขึ้นมาใหม่ ที่ + Create Datastream คลิกเลือกสถานะ การทำงานให้กับขาของ อุปกรณ์ (ในที่นี้เลือกเป็น Digital)

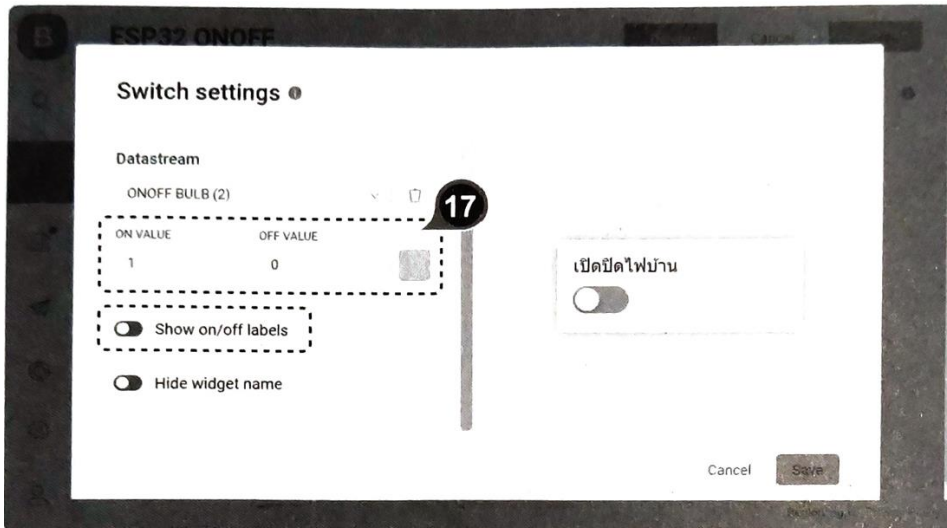


15. เมื่อเข้ามาในส่วนของการตั้งค่า Digital Datastream ในช่อง ALIAS ตั้งชื่อแ่งที่จะให้ปรากฏอยู่บนปุ่ม Switch ได้ (ในที่นี้ให้ชื่อว่า เปิดปิดไฟบ้าน) ในช่อง PIN ให้เลือกขาของอุปกรณ์ที่จะใช้ควบคุมการเปิด/ ปิดไฟ บ้าน (ในที่นี้เลือก 2 หรือก็คือขา D2/GPIO2) และในช่อง PIN MODE ให้เลือกโหมดหรือสถานะ การทำงาน ให้กับขาดังกล่าว (ในที่นี้เลือก Output)

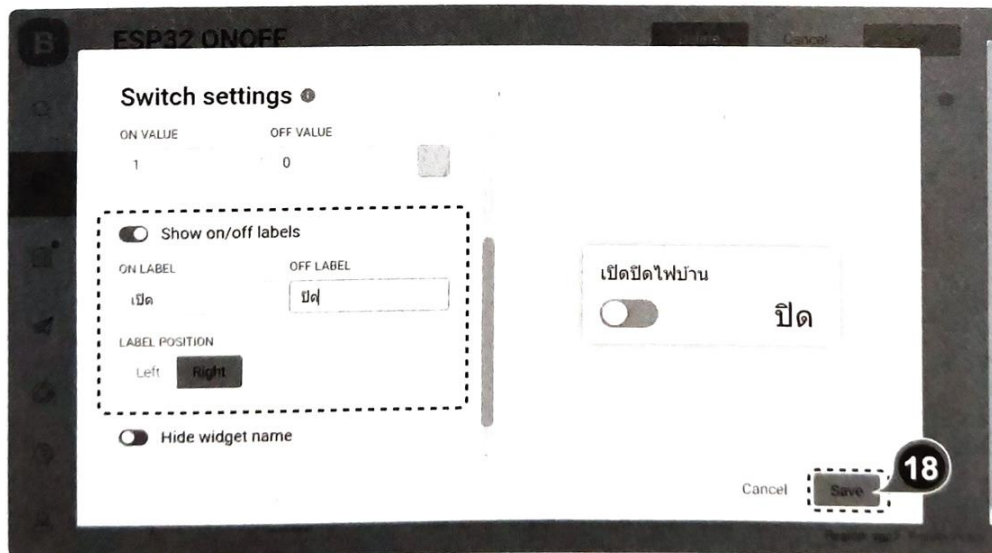


16. จากนั้นคลิกปุ่ม Create เพื่อสร้าง Digital Datastream

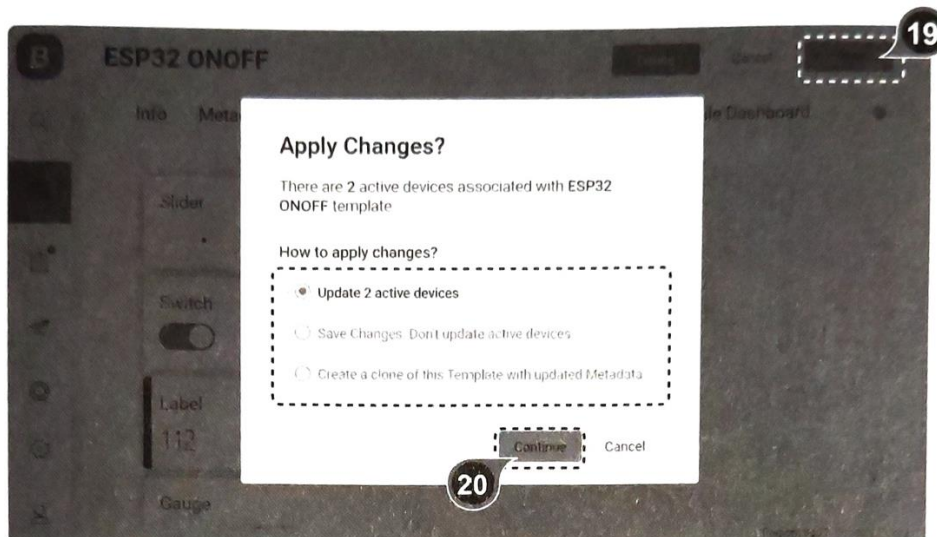
17. ที่ Datastream กำหนดสถานะ ON และ OFF ให้กับปุ่ม Switch เมื่อ Switch อยู่ในสถานะ ON ให้ค่า เป็น 1 ที่ขาอุปกรณ์ หลอดไฟบ้านจะติด แต่ถ้า Switch อยู่ในสถานะ OFF ให้ค่าเป็น 0 ที่ขาอุปกรณ์ หลอดไฟ บ้านจะดับ จากนั้นคลิกเลือกสีที่จะให้ปรากฏเวลาปุ่ม Switch ทำงาน สุดท้ายแตะเพื่อเปิดสวิทซ์ที่ Show on/off labels เพื่อแสดงป้ายข้อความกำกับให้เห็น



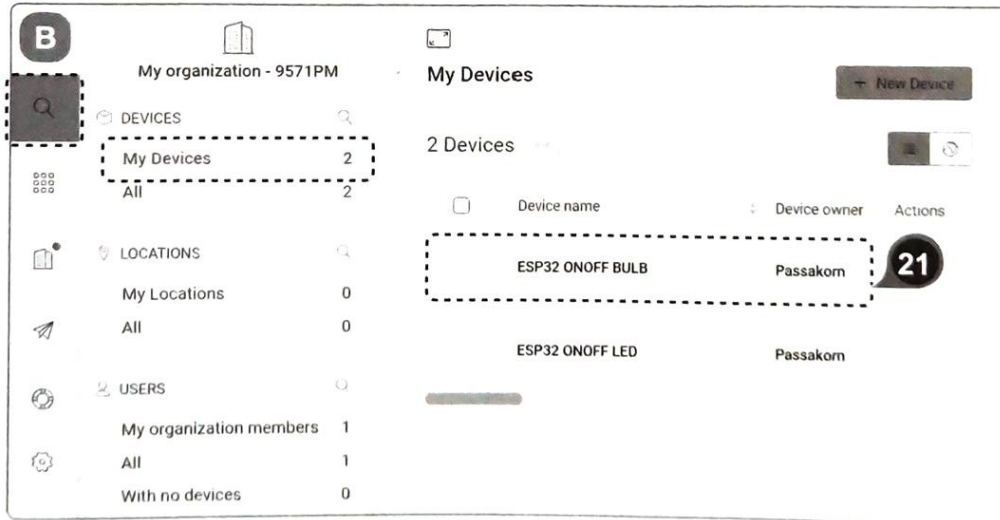
18. กำหนดรายละเอียดให้กับป้ายข้อความที่กำกับบนปุ่ม เมื่อ Switch ON หลอดไฟติด ให้แสดงข้อความกำกับว่า เปิด แต่ถ้า Switch OFF หลอดไฟดับ ให้แสดงข้อความกำกับว่า ปิด จากนั้นคลิกเลือกการจัดวางตำแหน่งของป้ายข้อความกำกับ (ในที่นี้เลือก Right หรือวางไว้ทางขวา) เสร็จแล้วคลิกปุ่ม Save



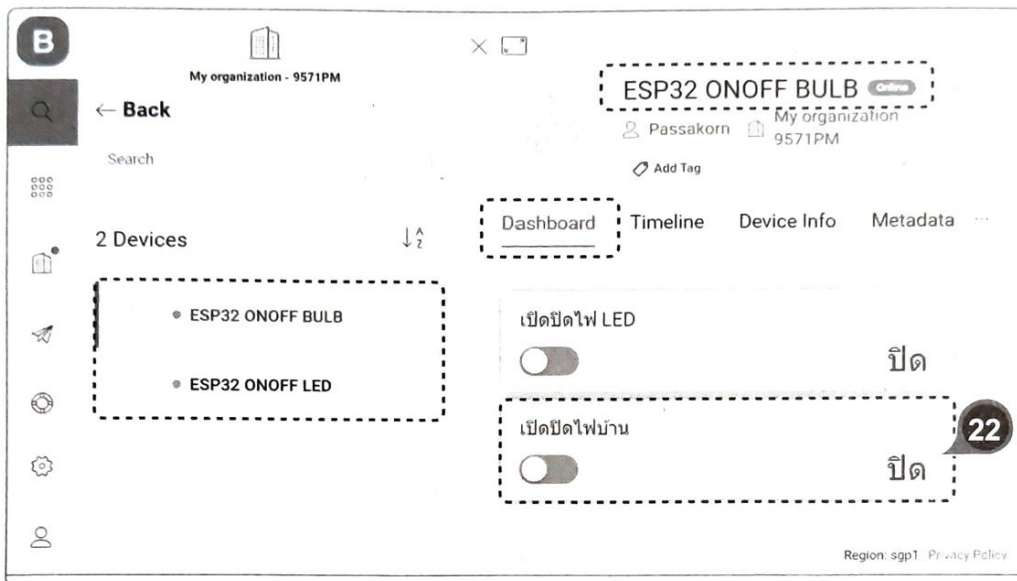
19. กลับมาที่หน้า Web Dashboard คลิกปุ่ม Save



20. จะปรากฏหน้าต่างให้เลือกว่าเมื่อ Save แล้วจะส่งผลให้เกิดการเปลี่ยนแปลงอย่างไร อัปเดตอุปกรณ์ที่กำลังทำงานอยู่ 2 ตัวในเวลานี้, บันทึกการเปลี่ยนแปลง แต่ยังไม่ต้องอัปเดตอุปกรณ์ที่กำลังทำงานอยู่ และสร้างสำเนาของ Template นี้ ด้วยการอัปเดต Metadata (ในที่นี้เลือก Update 2 active devices) จากนั้นคลิกปุ่ม Continue



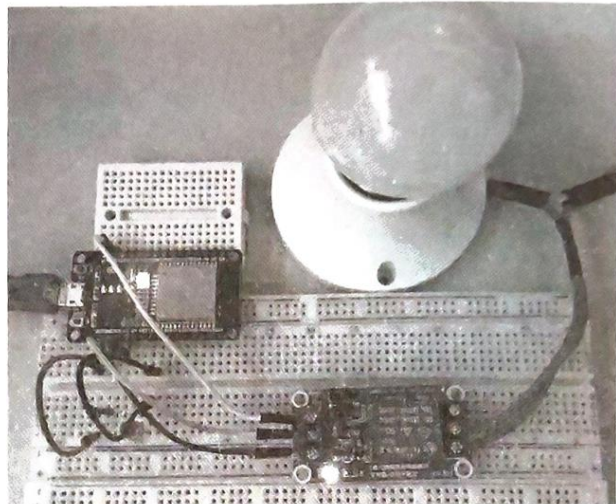
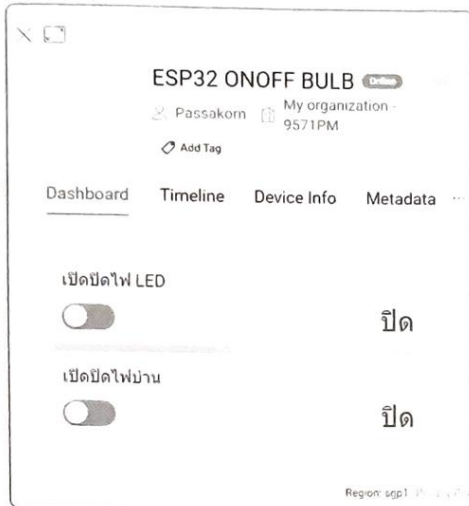
21. ที่ Main Menu คลิกเมนู Search » My Devices แล้วคลิกที่อุปกรณ์หรือ Device ที่เราเพิ่งสร้างล่าสุด (ในที่นี้คือ ESP32 ONOFF BULB)



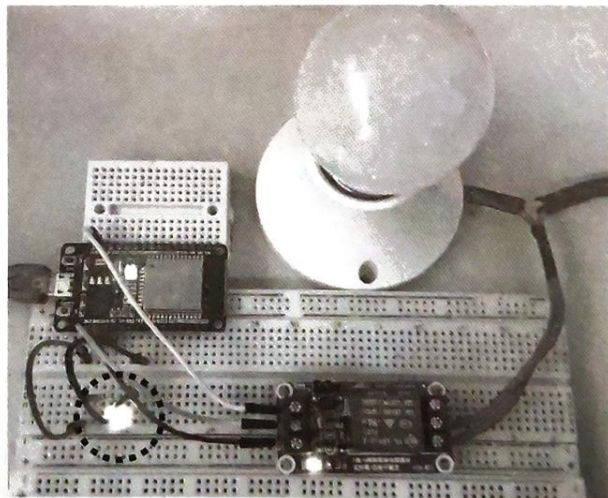
22. ในที่นี้เรามีอุปกรณ์ (Devices) ที่เป็นสวิตช์ปิด/เปิดอยู่ 2 ตัว ตัวหนึ่งเป็นของหลอดไฟ LED ที่มีขา D5/GPIO5 เป็น Digital Output ส่วนอีกตัวเป็นของรีเลย์ที่ต่ออยู่กับหลอดไฟบ้าน โดยมีขา D2/GPIO2 เป็น Digital Output แต่ละตัวจะมีปุ่มสวิตช์สำหรับควบคุมการปิด/เปิดหลอดไฟของตัวเองอยู่บน Dashboard และด้วยเหตุที่อุปกรณ์ทั้งสองอยู่ภายใต้ Template เดียวกัน และใช้ Template ID ร่วมกัน จึงทำให้ไม่ว่าจะมีอุปกรณ์ตัวหนึ่งตัวใดหรือทั้งสองตัวอยู่ในสถานะ Online หรือมีไฟสีเขียวโชว์ที่หน้าชื่ออุปกรณ์บ่งบอกถึงสถานะว่ามีการเชื่อมต่อกับ Blynk Cloud อยู่ ก็จะสามารถใช้งานปุ่มสวิตช์ที่อยู่บน Dashboard ควบคุม การปิด/เปิดหลอดไฟ LED และหลอดไฟบ้านได้ ยกตัวอย่างเช่น ในกรณีนี้ถึงแม้ว่าอุปกรณ์ ESP32 ONOFF LED จะมี

สถานะ Offline อยู่ แต่เราก็สามารถใช้ปุ่มสวิตช์บน Dashboard ของอุปกรณ์ ESP32 ONOFF BULB ซึ่งมีสถานะ Online อยู่ ควบคุมการปิด/เปิดหลอดไฟ LED และหลอดไฟบ้านได้

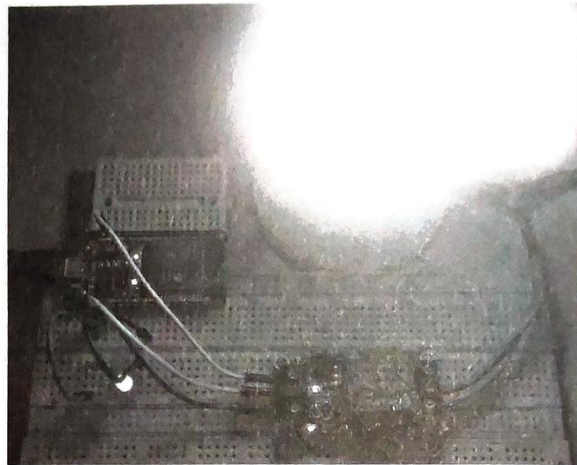
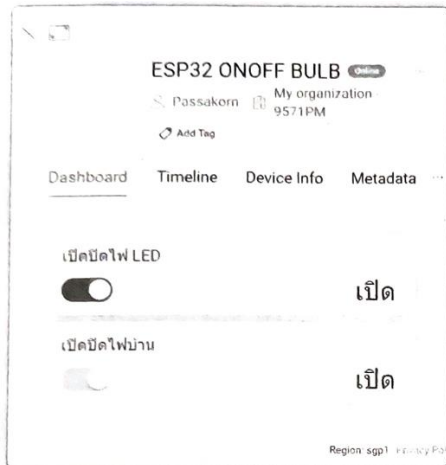
23. ทดสอบด้วยการคลิกเลือกอุปกรณ์ (Devices) ตัวที่มีสถานะ Online อยู่ (ในที่นี้คือ ESP32 ONOFF BULB) แล้วคลิกปุ่มสวิตช์ที่อยู่บน Dashboard เพื่อเปิด/ปิดหลอดไฟ LED และหลอดไฟบ้าน ถ้าเป็นหลอดไฟ LED เมื่อปุ่มสวิตช์เป็น ON หรือ เปิด ดิจิตอลเอาต์พุตที่ขา D5/GPIO5 จะเป็น 0 หลอดไฟ LED จะ “ติดสว่าง” แต่ถ้าปุ่มสวิตช์เป็น OFF หรือ ปิด ดิจิตอลเอาต์พุตที่ขา D5/GPIO5 จะเป็น 1 หลอดไฟ LED จะ “ดับ” แต่ถ้าเป็นหลอดไฟบ้าน เมื่อปุ่มสวิตช์เป็น ON หรือ เปิด ดิจิตอลเอาต์พุตที่ขา D2/GPIO2 จะเป็น 1 ทำให้ รีเลย์ที่มีสถานะเป็น Active HIGH ทำงาน ต่อกระแสไฟให้ครบวงจร หลอดไฟบ้านจะ “ติดสว่าง” แต่ถ้า ปุ่มสวิตช์เป็น OFF หรือ ปิด ดิจิตอลเอาต์พุตที่ขา D2/GPIO2 จะเป็น 0 ทำให้รีเลย์ไม่ทำงาน กระแสไฟ ไหลไม่ครบวงจร หลอดไฟบ้านจะ “ดับ” เป็นอันว่าเราสามารถที่จะควบคุมการเปิด/ปิดหลอดไฟ LED และ ไฟบ้านจาก Dashboard ของ Blynk.Console จากบนคอมพิวเตอร์ผ่านทางอินเทอร์เน็ตได้แล้ว ต่อไป เราจะไปสร้างปุ่มสวิตช์ไว้บน Dashboard ของ Blynk.Apps เพื่อไว้ใช้ควบคุมการปิด/เปิดไฟบ้านจากบน สมาร์ทโฟนกันดูบ้าง



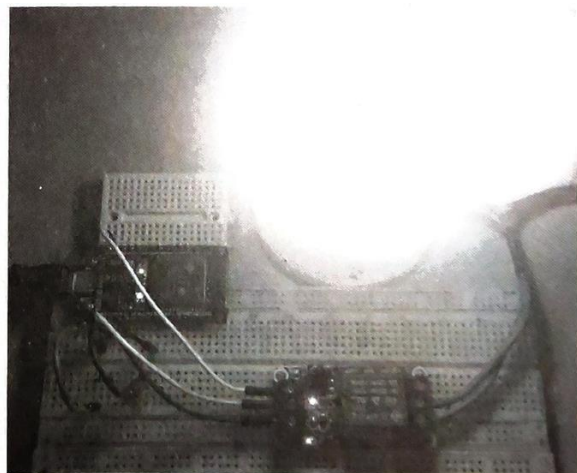
เมื่อปิดสวิตซ์ไฟ LED และไฟบ้านที่ Dashboard หลอดไฟ LED และไฟบ้านจะ “ดับ”



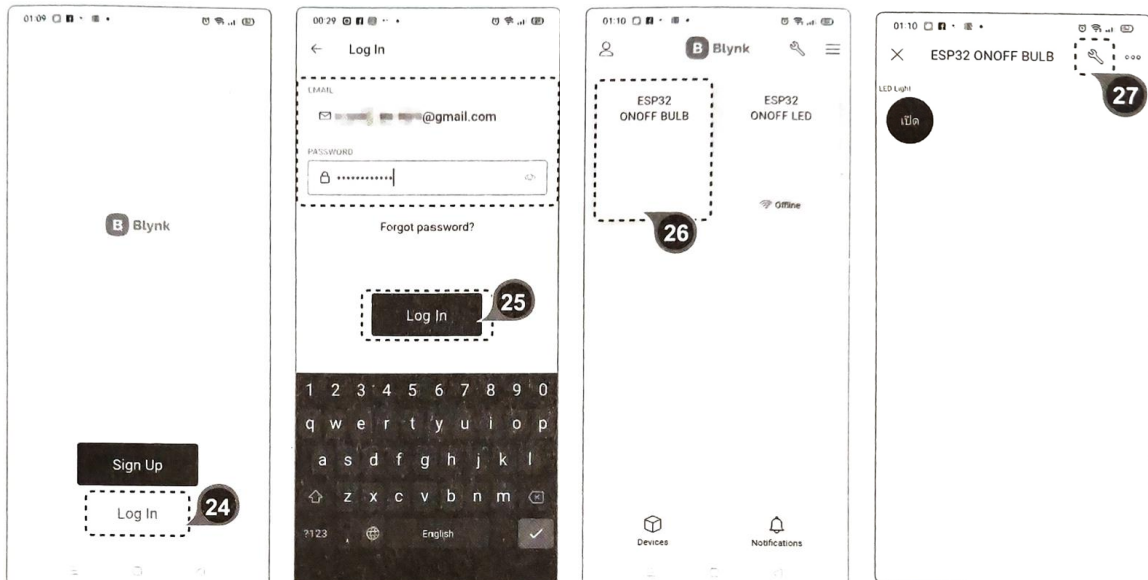
เมื่อเปิดสวิตซ์ไฟ LED แต่ปิดไฟบ้านที่ Dashboard หลอดไฟ LED จะ “ติดสว่าง” แต่ไฟบ้านจะ “ดับ”



เมื่อเปิดทั้งสวิตช์ไฟ LED และไฟบ้านที่ Dashboard หลอดไฟ LED และไฟบ้านจะ “ติดสว่าง”



เมื่อปิดสวิตช์ไฟ LED แต่เปิดไฟบ้านที่ Dashboard หลอดไฟ LED จะ “ดับ” แต่ไฟบ้านจะ “ติดสว่าง”

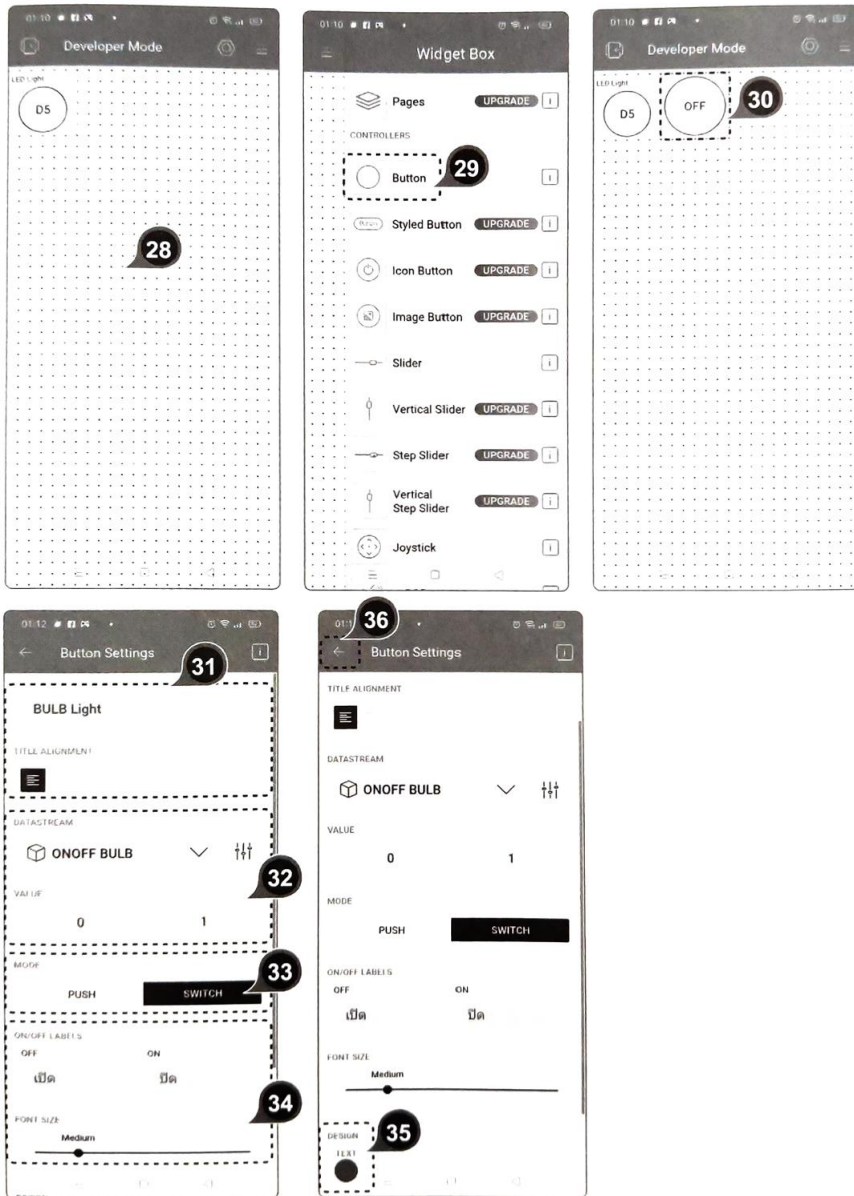


24. เริ่มจากเปิดแอป Blynk IoT จากบนสมาร์โฟนหรือแท็บเล็ตขึ้นมา แตะที่ Log In

25. จากนั้นกรอกอีเมล และพาสเวิร์ดที่ได้ลงทะเบียนไว้ก่อนหน้านี้ เสร็จแล้วแตะที่ปุ่ม Log In

26. ในหน้านี้เราจะเห็น Device ที่เราได้สร้างเพิ่มไว้บน Blynk Console ก่อนหน้านี้ (นั่นคือ ESP32 ONOFF BULB) ให้แตะที่ Device ดังกล่าวเพื่อเข้าไปสร้าง Dashboard (ที่ Device ถ้าอุปกรณ์ไม่ได้เชื่อมต่อกับ Blynk Cloud จะปรากฏสัญลักษณ์ Offline)

27. ในหน้า Dashboard จะเห็นปุ่มสวิตช์ปิด/เปิดหลอดไฟ LED เราได้เคยสร้างเอาไว้ในตัวอย่างที่แล้ว ให้เราแตะที่ 2 เพื่อเข้าสู่หน้าแก้ไขหรือสร้าง Dashboard

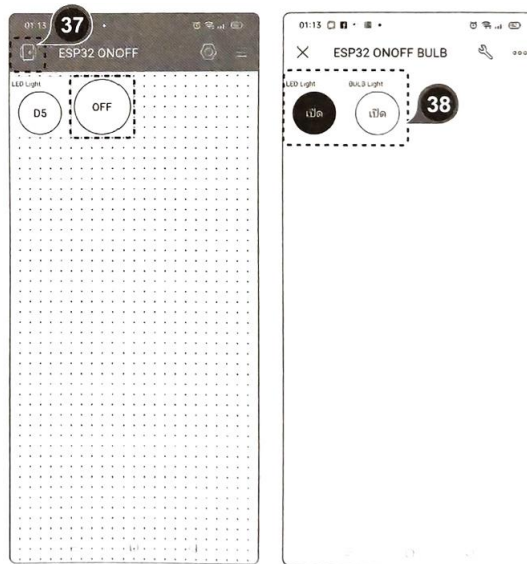


28. แตะลงบนพื้นที่ว่าง เพื่อเปิด หน้า Widget Box

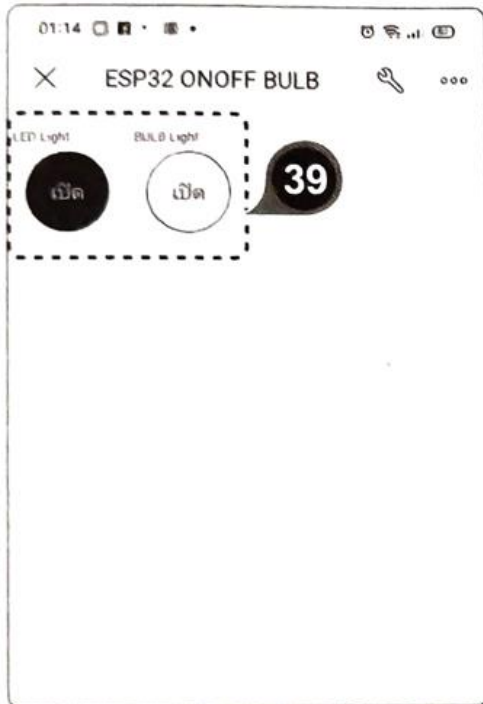
29. ในหน้า Widget Box ดังรูป ให้เราเลือกรูปแบบของปุ่ม ควบคุมหรือส่วนแสดงผล ต่างๆ ที่ต้องการ ในที่นี้ เรา ต้องการปุ่มสวิตช์สำหรับไว้ ใช้เปิด/ปิดไฟ ดังนั้นให้แตะ เลือก Button

30. ปุ่ม Button จะถูกนำมาวาง ไว้บนแผงหน้าปัดหรือแผงควบคุม ให้เราแตะที่ปุ่ม Button นั้น เพื่อเข้าไป ตั้ง ค่าการทำงานให้กับปุ่ม

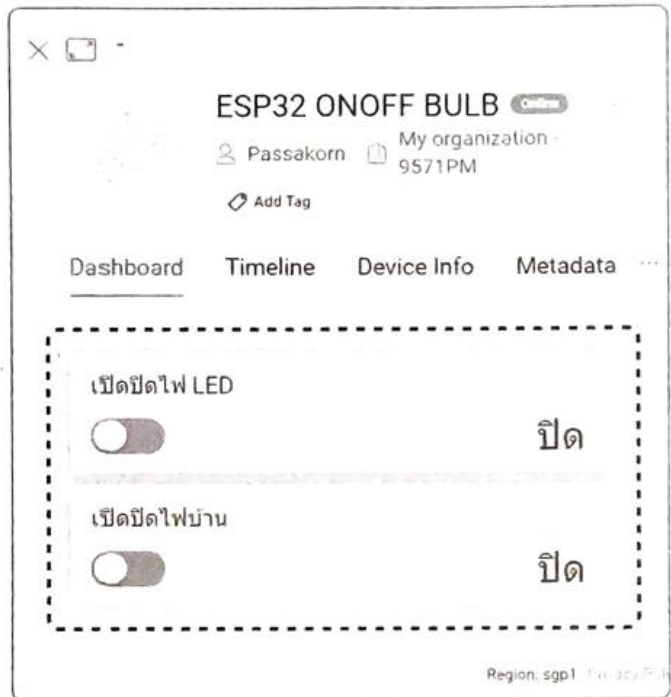
31. ที่ TITLE ตั้งชื่อให้กับปุ่ม (ในที่นี้คือ BULB Light) และเลือกการจัดวางตำแหน่งของข้อความ
32. ที่ DATASTREAM ให้เราแตะเพื่อเลือก Data Stream หรือก็คือข้อมูลการตั้งค่าการทำงานให้กับปุ่ม สวิตช์ ซึ่งในที่นี้เราได้เคยสร้าง Data Stream หรือตั้งค่าการทำงานเอาไว้แล้วก่อนหน้านี้บน Blynk Console โดยตั้งชื่อไว้ว่า ONOFF BULB (สามารถแตะที่ปุ่ม H เพื่อเปิดดูรายละเอียดของการตั้งค่า นั้นได้) ดังนั้นให้เราแตะที่ Select Data Stream แล้วเลือก ONOFF BULB ทั้งนี้ค่าสถานะการทำงาน (VALUE) ของขาอุปกรณ์ คือ OFF มีค่าเป็น 0 และ ON มีค่าเป็น 1 ก็จะถูกนำมาใช้งานด้วย ดังรูป
33. เลือกโหมดการทำงานให้กับปุ่มสวิตช์ ในที่นี้เลือก SWITCH คือเป็นแบบ กด “ติด” กดอีกครั้ง “ “ดับ” (ส่วน PUSH จะเป็นแบบ กด “ติด” ปล่อย “ดับ”)
34. กำหนดข้อความที่จะกำกับไว้บนปุ่ม ในที่นี้เมื่อปุ่มอยู่ในสถานะ Switch OFF (หลอดไฟดับ) ให้แสดงข้อความกำกับบนปุ่มว่า ปิด แต่ถ้าปุ่มอยู่ในสถานะ Switch ON (หลอดไฟติด) ให้แสดงข้อความกำกับ บนปุ่มว่า ปิด และเลือกปรับขนาดของตัวอักษรบนปุ่ม
35. เลือกสีของปุ่มที่จะให้แสดงเมื่ออยู่ในสถานะ Switch ON (ในที่นี้เลือก สีแดง)
36. เสร็จแล้วแตะที่ปุ่มเพื่อย้อนกลับไปยังพื้นที่สร้าง Dashboard
37. จะเห็นปุ่มสวิตช์ที่เราตั้งค่าการทำงานเอาไว้แล้ว ให้เราแตะ ที่ปุ่ม เพื่อเชื่อมต่อไปยัง อุปกรณ์ IoT ที่อยู่ปลายทาง หรือในที่นี้ก็คือบอร์ด DOIT ESP32 DevKit V1 ที่ใช้ควบคุมการเปิด/ปิดหลอดไฟ LED และ ไฟบ้าน
38. เมื่อเชื่อมต่อได้แล้วจะสังเกตเห็นว่าปุ่มสวิตช์ที่สร้างไว้บน Blynk. Apps พร้อมทำงาน ดังรูป



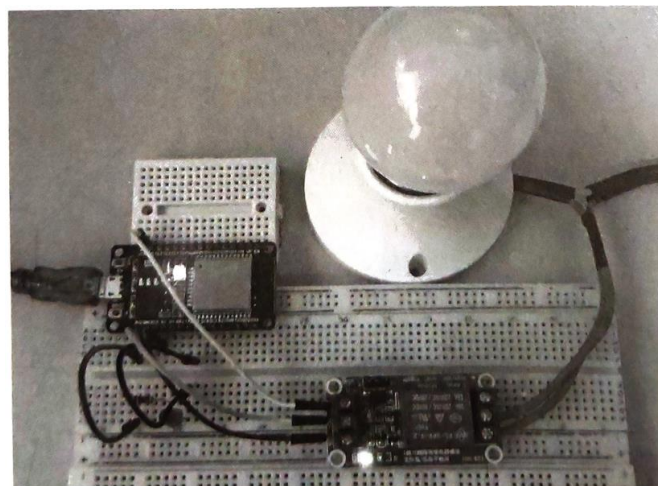
39. ทดสอบสวิตช์ปิด/เปิดของหลอดไฟ LED และไฟบ้านที่อยู่บน Dashboard ของ Blynk.Apps ด้วยการแตะที่ปุ่ม ปิด ของสวิตช์ทั้ง 2 ตัว (ข้อความบนปุ่มจะเปลี่ยนเป็น เปิด ดังรูป) สถานะเอาต์พุตที่ขา GPIO5 จะเป็น 1 หลอดไฟ LED จะ “ดับ” และขา GPIO2 จะเป็น 0 (รีเลย์ไม่ทำงาน) หลอดไฟบ้านจะ “ดับ” ระหว่างนี้ จะเห็นว่าปุ่มสวิตช์ที่อยู่บน Dashboard ของ Blynk.Console ก็จะตอบสนองไปตามการควบคุม สวิตช์ที่อยู่บน Dashboard ของ Blynk.Apps ด้วย



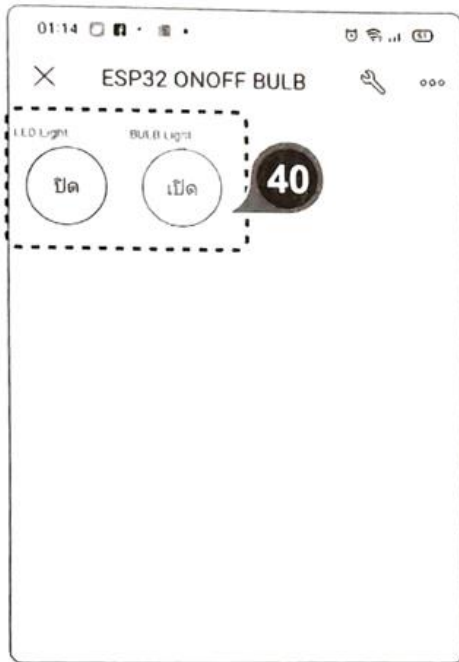
Dashboard ของ Blynk.Apps บนสมาร์ทโฟน



Dashboard ของ Blynk.Console บนคอมพิวเตอร์



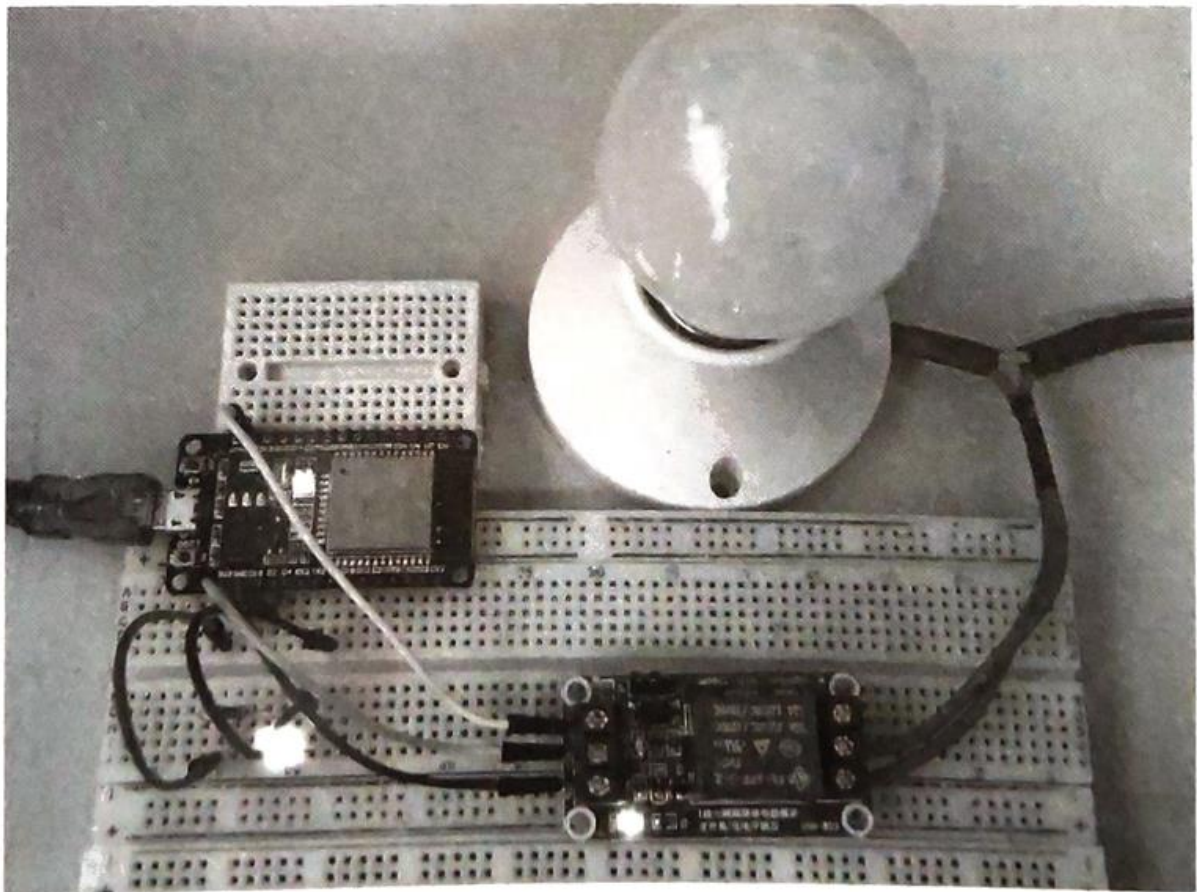
เมื่อปิดสวิตช์ไฟ LED และไฟบ้านที่ Dashboard ทั้งหลอดไฟ LED และไฟบ้านจะ “ดับ 40. บน Dashboard ของ Blynk.Apps ทดสอบแตะที่ปุ่ม เปิด ของสวิตช์หลอดไฟ LED (ข้อความบนปุ่มจะเปลี่ยนเป็น ปิด ดังรูป) สถานะเอาต์พุตที่ขา GPIOs จะเป็น 0 หลอดไฟ LED จะ “ติดสว่าง” ระหว่างนี้ จะเห็นว่าปุ่มสวิตช์ปิด/เปิดของหลอดไฟ LED ที่อยู่บน Dashboard ของ Blynk.Console จะตอบสนอง โดยเปลี่ยนเป็น เปิด โดยอัตโนมัติ



Dashboard ของ Blynk.Apps บนสมาร์ทโฟน

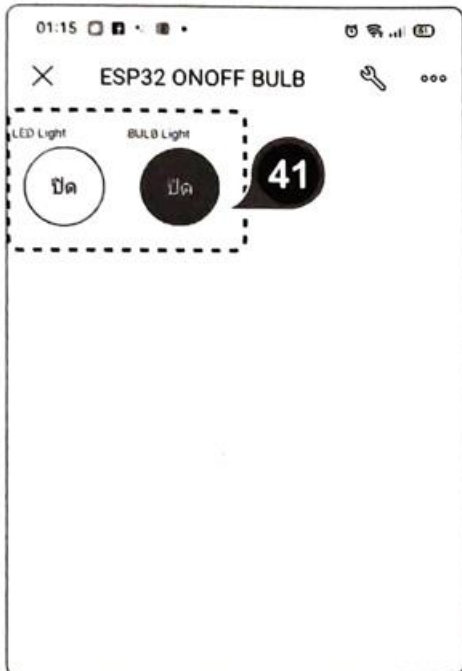


Dashboard ของ Blynk.Console บนคอมพิวเตอร์



เมื่อเปิดสวิตช์ไฟ LED ที่ Dashboard หลอดไฟ LED จะ “ติดสว่าง”

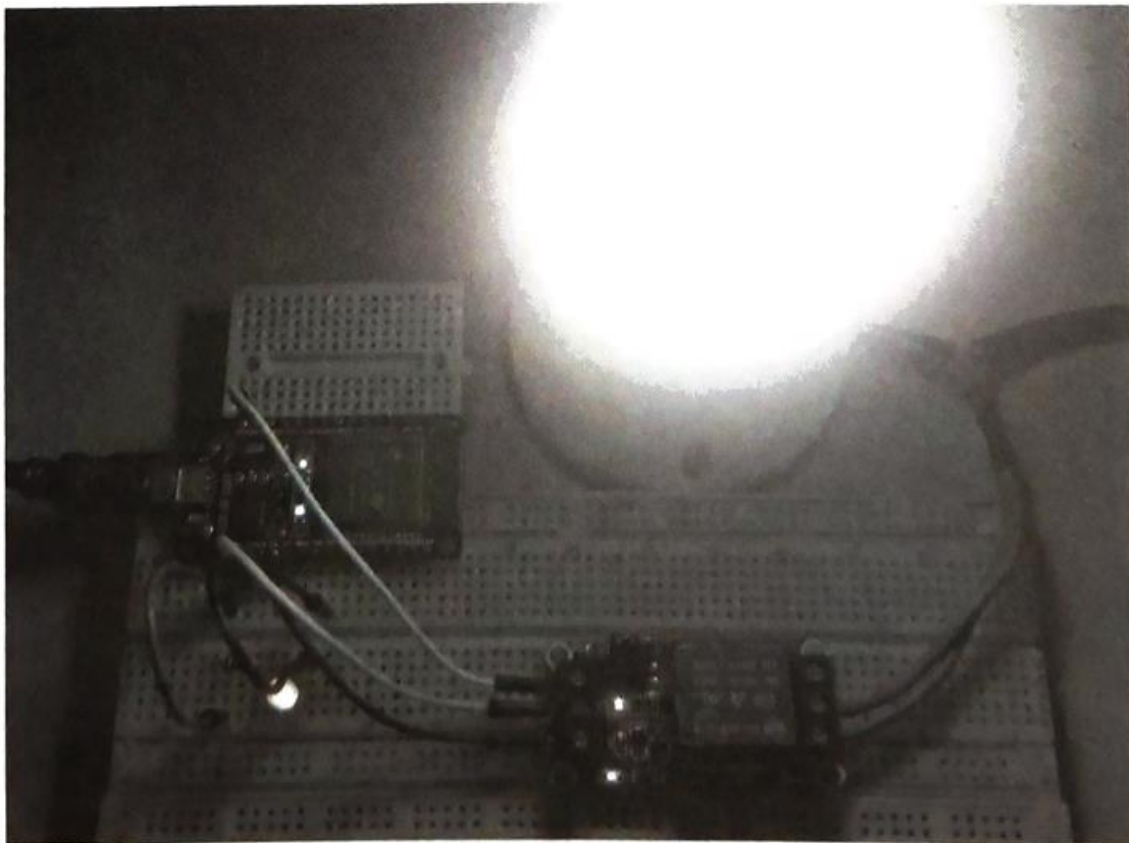
41. บน Dashboard ของ Blynk.Apps ทดสอบแตะที่ปุ่ม เปิด ของสวิตช์ไฟบ้าน (ข้อความบนปุ่มจะเปลี่ยนเป็น ปิด ดังรูป) สถานะเอาต์พุตที่ขา GPIO2 จะเป็น 1 รีเลย์จะทำงาน ไฟบ้านจะ “ติดสว่าง” ระหว่างนี้ จะเห็นว่าปุ่มสวิตช์ปิด/เปิดของไฟบ้านที่อยู่บน Dashboard ของ Blynk.Console จะตอบสนองโดยเปลี่ยนเป็น เปิด โดยอัตโนมัติ



Dashboard ของ Blynk.Apps บนสมาร์ตโฟน

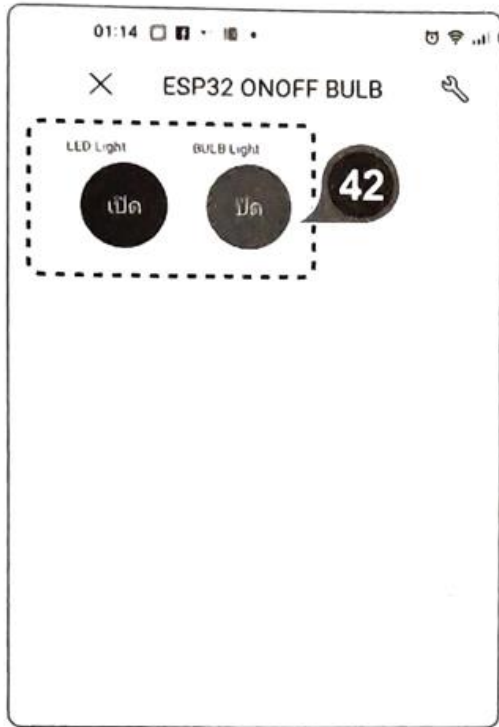


Dashboard ของ Blynk.Console บนคอมพิวเตอร์

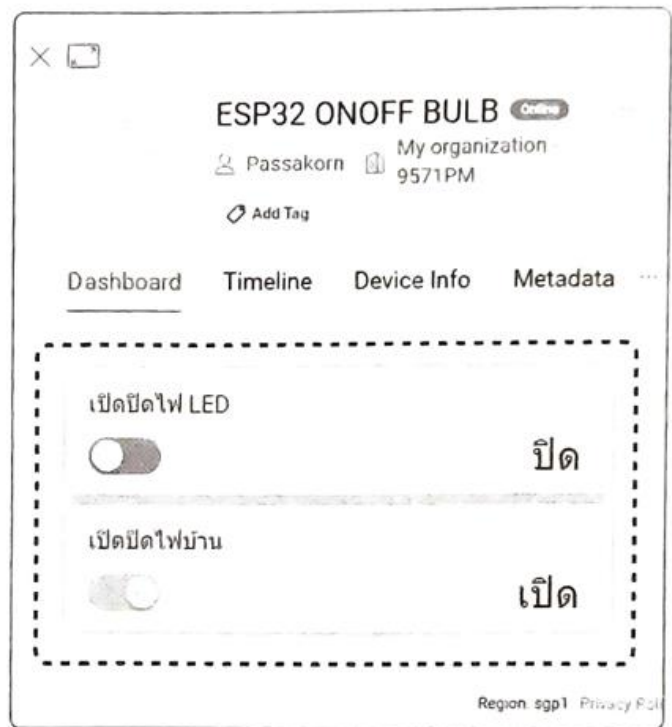


เมื่อเปิดสวิตช์ไฟ LED ที่ Dashboard หลอดไฟ LED จะ “ติดสว่าง”

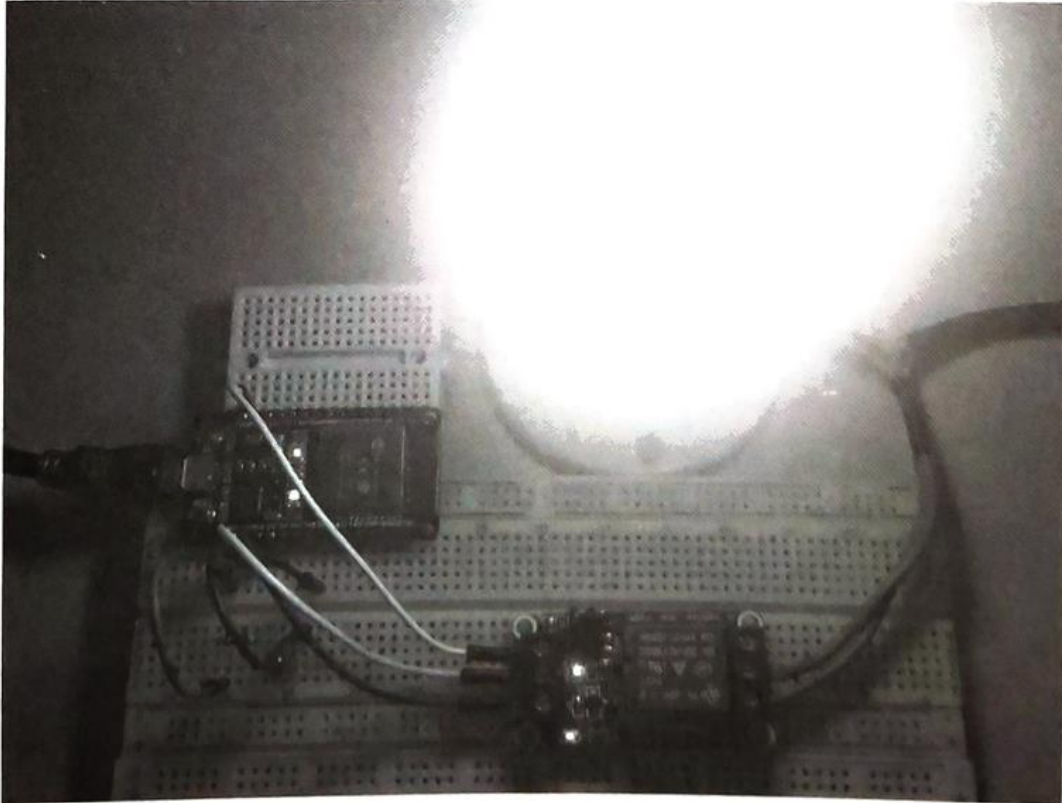
42. บน Dashboard ของ Blynk.Apps ทดสอบแตะที่ปุ่ม ปิด ของสวิตช์หลอดไฟ LED (ข้อความบนปุ่มจะเปลี่ยนเป็น เปิด ดังรูป) สถานะเอาต์พุตที่ขา GPIO5 จะเป็น 1 หลอดไฟ LED จะ “ดับ” ระหว่างนี้จะเห็นว่า ปุ่มสวิตช์ปิด/เปิดของหลอดไฟ LED ที่อยู่บน Dashboard ของ Blynk.Console จะตอบสนองโดยเปลี่ยน เป็น ปิด โดยอัตโนมัติ เป็นอันว่าเราสามารถที่จะควบคุมการเปิด/ปิดหลอดไฟ LED และไฟบ้าน ได้ทั้ง จากบน Dashboard ของ Blynk Apps และ Blynk.Console ผ่านทางอินเทอร์เน็ตได้แล้ว



Dashboard ของ Blynk.Apps บนสมาร์ทโฟน



Dashboard ของ Blynk.Console บนคอมพิวเตอร์



เมื่อเปิดสวิตช์ไฟบ้านที่ Dashboard รีเลย์จะทำงาน ไฟบ้านจะ “ติดสว่าง”

บันทึกผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....