

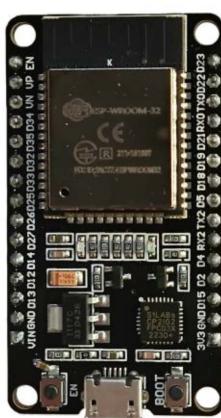
การทดลองที่ 6 การพัฒนาอินเทอร์เน็ตของสิ่งของโดยใช้อุปกรณ์ Blynk

วัตถุประสงค์

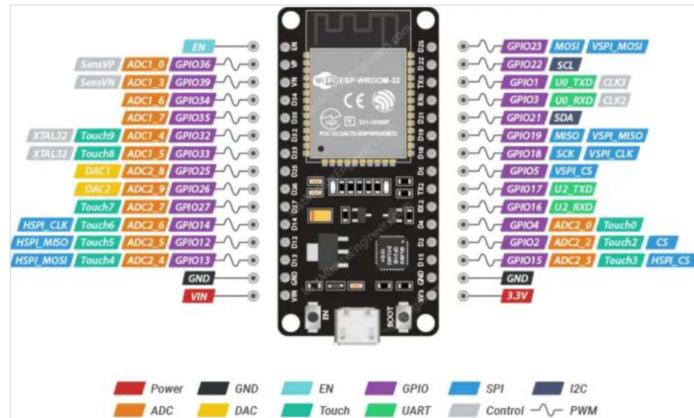
การทดลองนี้มีวัตถุประสงค์เพื่อให้นักศึกษา

- ศึกษาและพัฒนาโปรแกรมสำหรับการเชื่อมต่อระหว่างอุปกรณ์ IoT และแอปพลิเคชัน Blynk
- ควบคุมการเปิด-ปิด LED ผ่านแอปพลิเคชัน Blynk
- แสดงการวัดปริมาณจากเซนเซอร์ เช่น วัดอุณหภูมิ หรือความชื้น ผ่านแอปพลิเคชัน Blynk
- เสริมทักษะการเชื่อมต่อและตรวจสอบจุดบกพร่องของวงจร

การทดลองนี้ใช้บอร์ด ESP32 (ESP-WROOM-32 ภาพที่ 1 (ก) แสดงบอร์ด ESP-WROOM-32 และภาพที่ 1 (ข) แสดงแผนผังขา รายละเอียดของบอร์ด WROOM ESP32 ได้กล่าวไว้แล้วในปฏิบัติการทดลองที่ 5



ก. บอร์ด

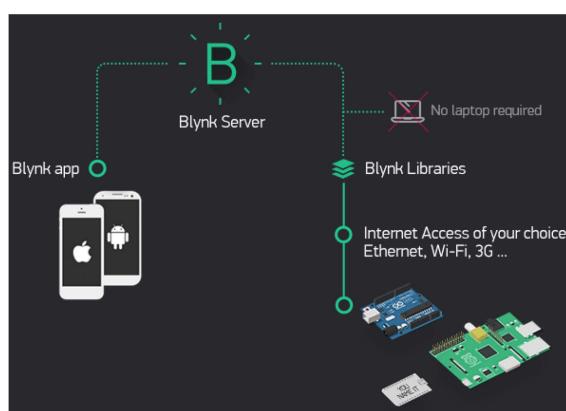


ข. แผนผังขา

ภาพที่ 1 บอร์ด ESP-WROOM-32

ที่มา (ข) <https://lastminuteengineers.com/esp32-pinout-reference/>

แอปพลิเคชัน Blynk เป็นแพลตฟอร์มหนึ่งสำหรับพัฒนาอินเทอร์เน็ตของสิ่งของโดย Blynk มีเซิร์ฟเวอร์สำหรับการสื่อสารระหว่างสมาร์ตโฟนและฮาร์ดแวร์ และ Blynk ไลบรารีสำหรับอุปกรณ์แสดงดังภาพที่ 2



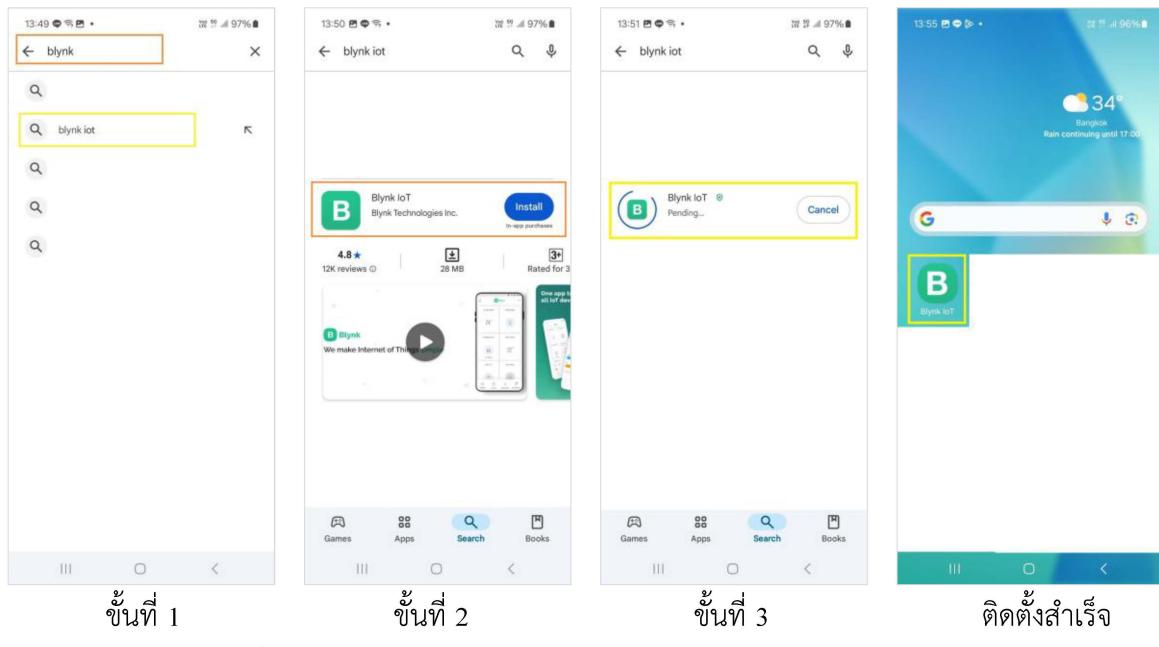
ภาพที่ 2 โครงสร้างของแอปพลิเคชัน Blynk

ที่มา <https://lastminuteengineers.com/esp32-pinout-reference/>

รายละเอียดเพิ่มเติมของแอปพลิเคชัน Blynk กับการพัฒนาระบบอินเทอร์เน็ตของสรรพสิ่ง <https://blynk.io/blynk-iot-low-code-software-platform> ต่อไปนี้จะใช้คำว่า Blynk แทนแอปพลิเคชัน Blynk

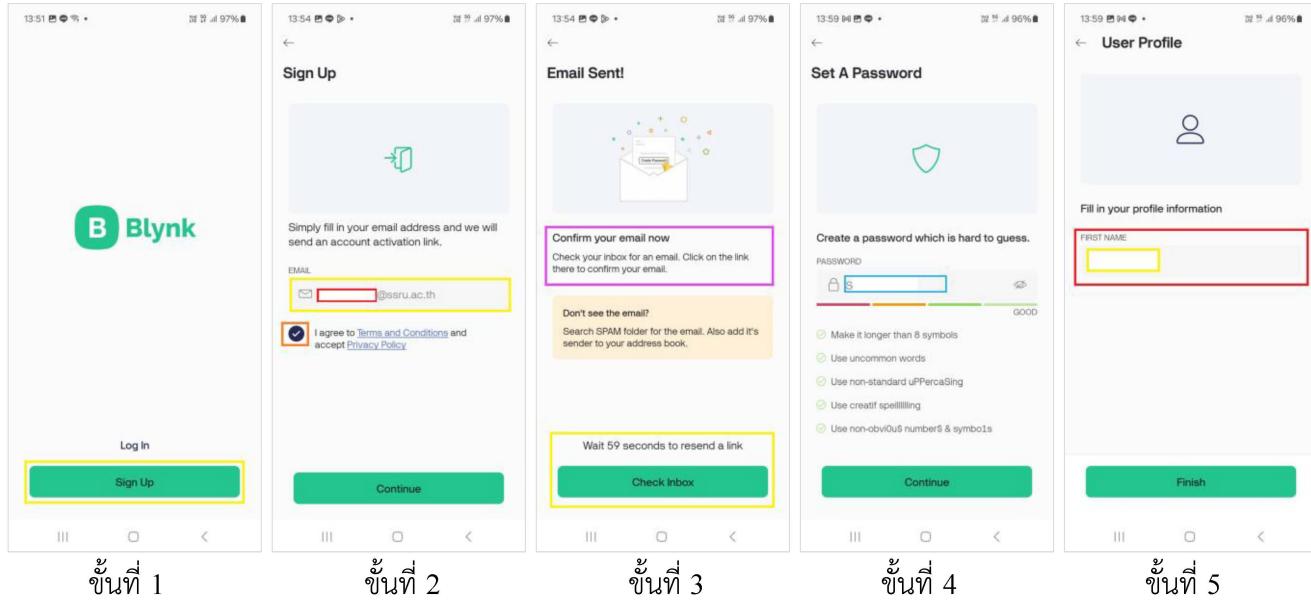
การติดตั้ง Blynk สำหรับระบบปฏิบัติการ Android มีขั้นตอนดังต่อไปนี้

1. ดาวน์โหลดและติดตั้ง Blynk ดังภาพที่ 3



ภาพที่ 3 ดาวน์โหลดและติดตั้ง Blynk

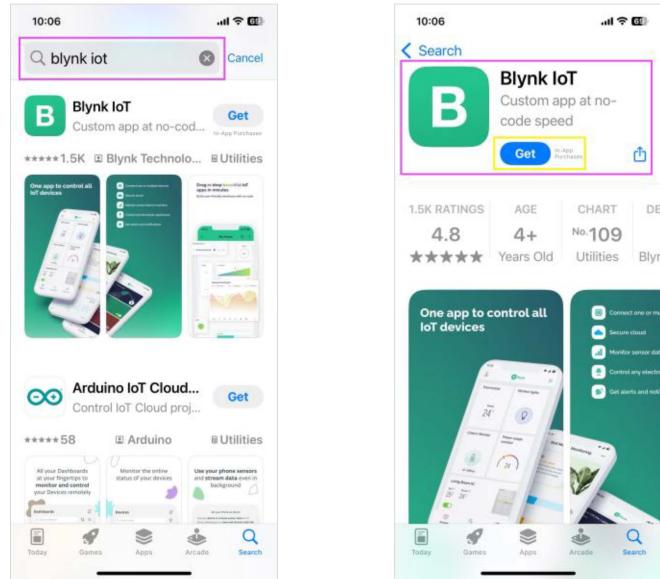
2. สร้างบัญชีผู้ใช้ดังภาพที่ 4 หากมีบัญชีผู้ใช้แล้วเลือก Log in เพื่อเข้าสู่ Blynk



ภาพที่ 4 สร้างบัญชีผู้ใช้

การติดตั้ง Blynk สำหรับอุปกรณ์ปฏิบัติการ iOS มีขั้นตอนดังต่อไปนี้

1. ดาวน์โหลดและติดตั้ง Blynk ดังภาพที่ 5

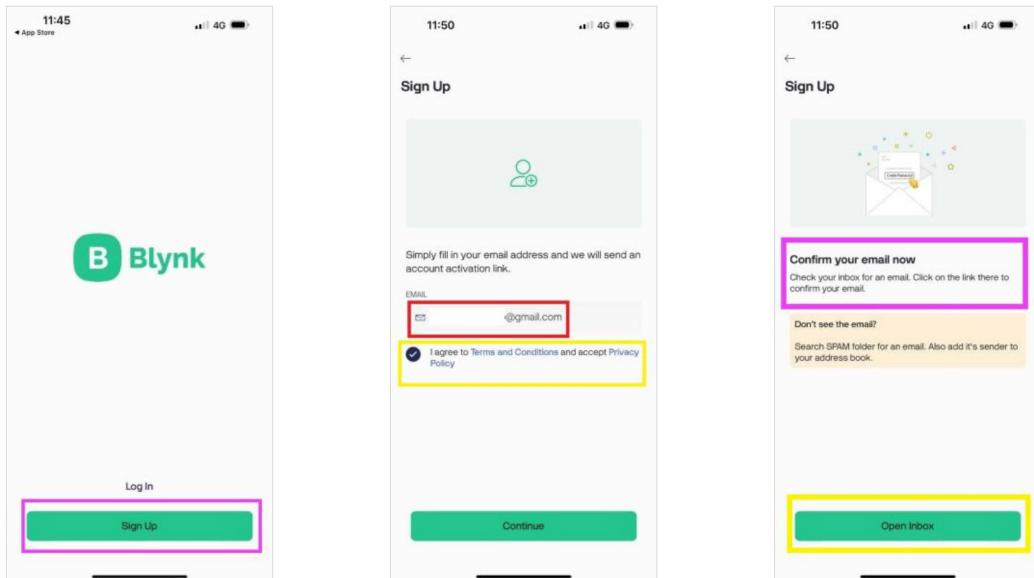


ค้นหา

ติดตั้ง

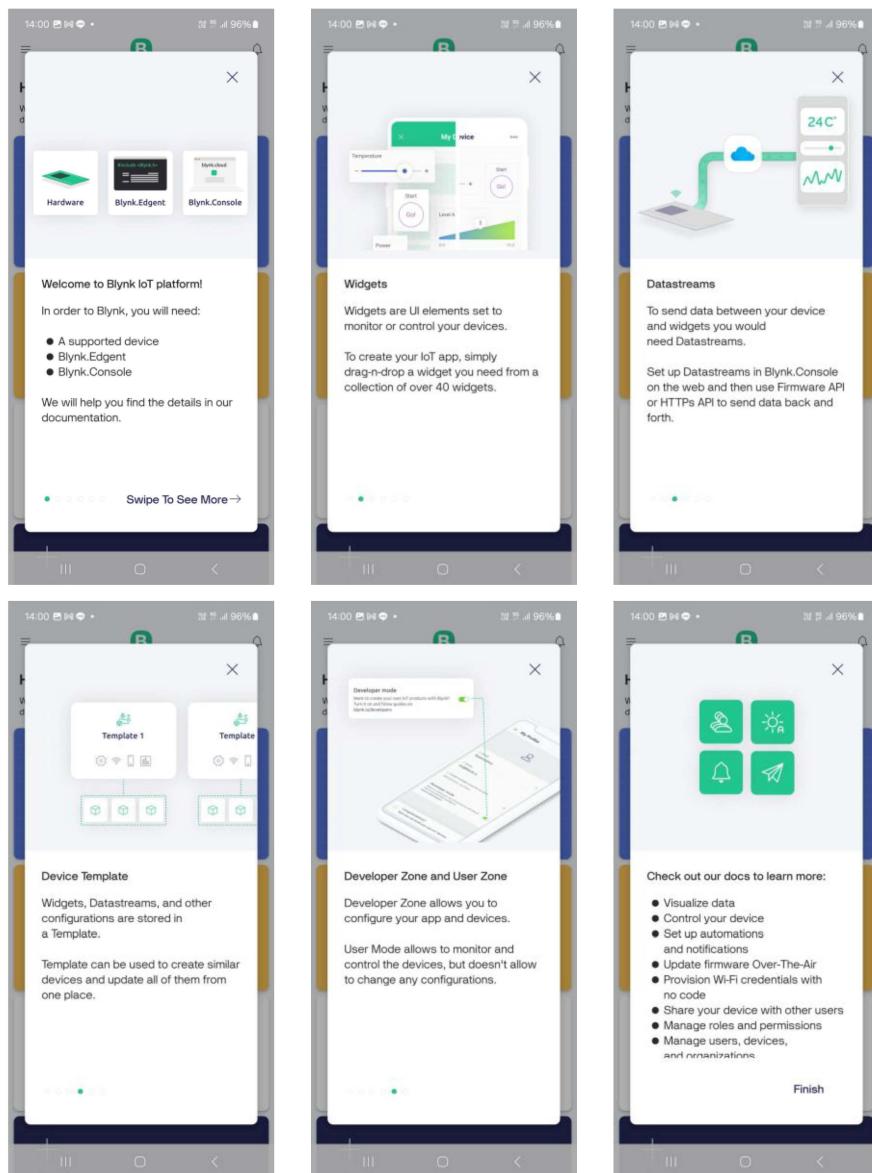
ภาพที่ 5 ดาวน์โหลดและติดตั้ง Blynk

2. สร้างบัญชีผู้ใช้ดังภาพที่ 6 หากมีบัญชีผู้ใช้แล้วเลือก Log in เพื่อเข้าสู่ Blynk



ภาพที่ 6 สร้างบัญชีผู้ใช้

เมื่อเข้าสู่ Blynk ทั้งสองระบบปฏิบัติการจะมีลักษณะคล้ายกัน เมื่อเข้าสู่ Blynk ครั้งแรกจะปรากฏหน้าจอตั้งภาพที่ 7 (แสดงรายละเอียดเกี่ยวกับ Blynk) และหน้าจอหลักของ Blynk แสดงตั้งภาพที่ 8

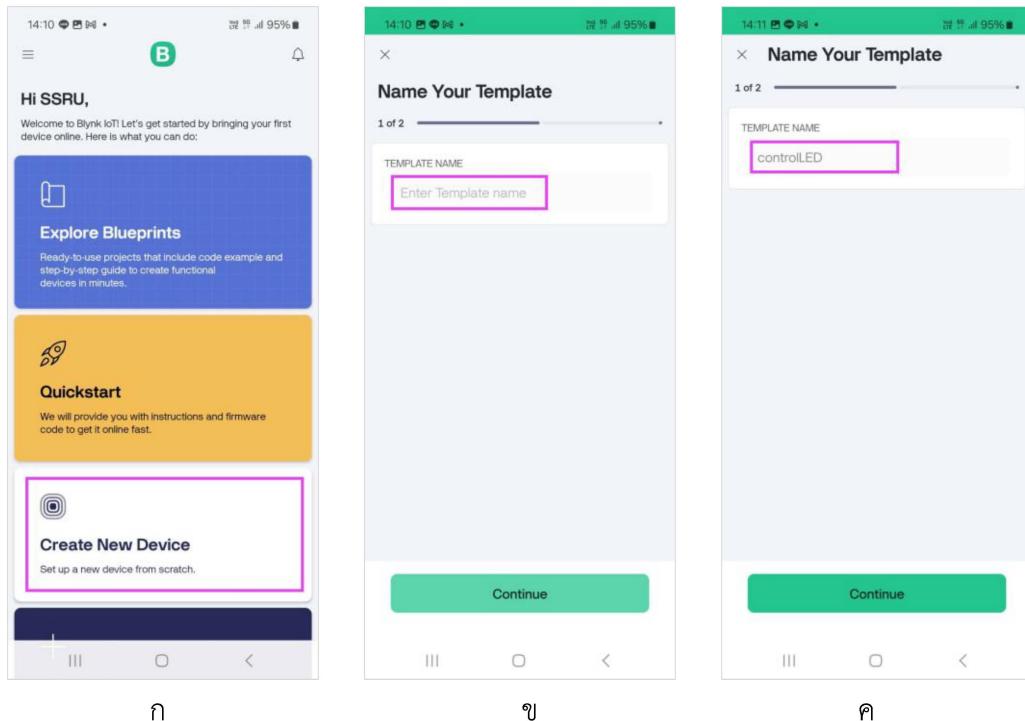


ภาพที่ 7 หน้าจอ Blynk



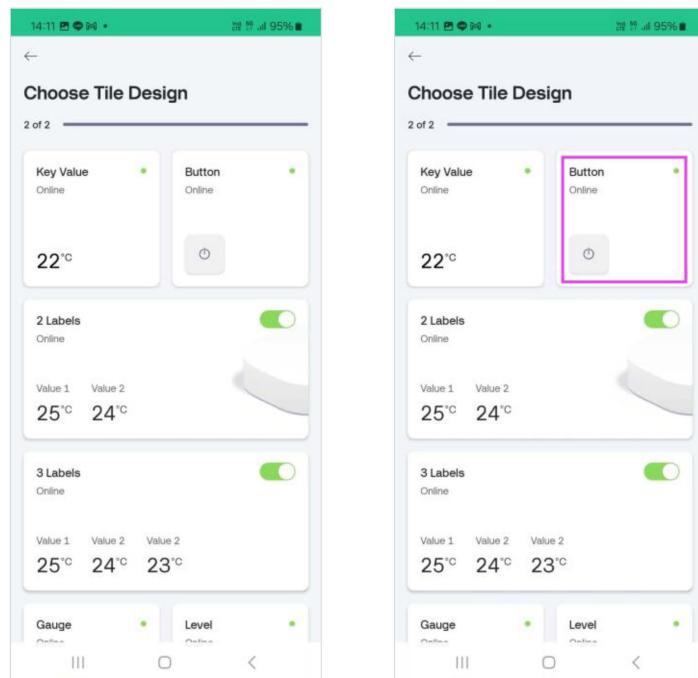
ภาพที่ 8 หน้าจอหลัก Blynk

3. เมื่อคลิกที่ Create New Device ในภาพที่ 9 (ก) จะปรากฏหน้าจอดังภาพที่ 9 (ข) และตั้งชื่อ “controlLed” ดังภาพที่ 9 (ค)



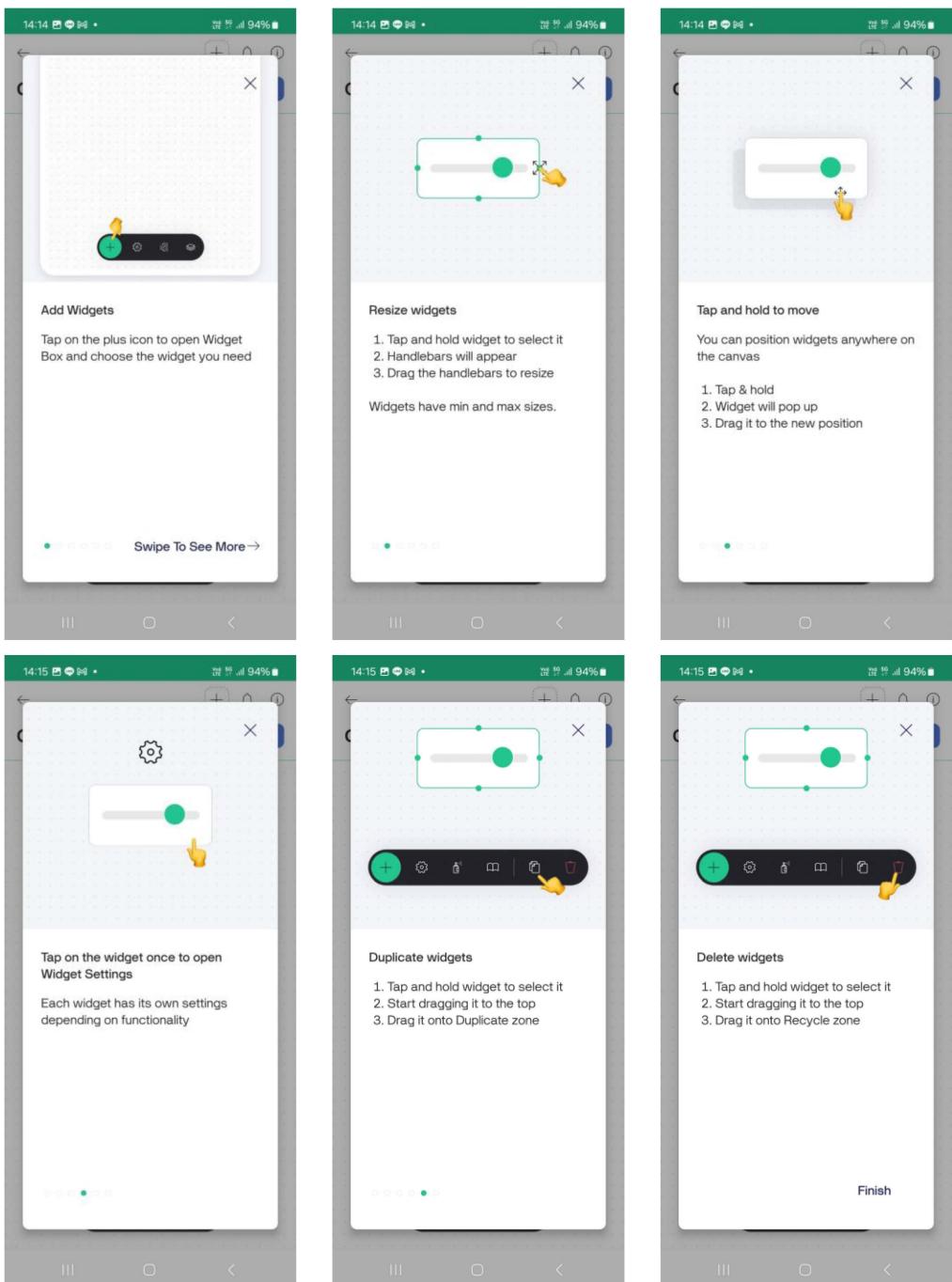
ภาพที่ 9 Create New Device

จากนั้นคลิก Continue จะปรากฏหน้าจอดังภาพที่ 10 เลือก Button



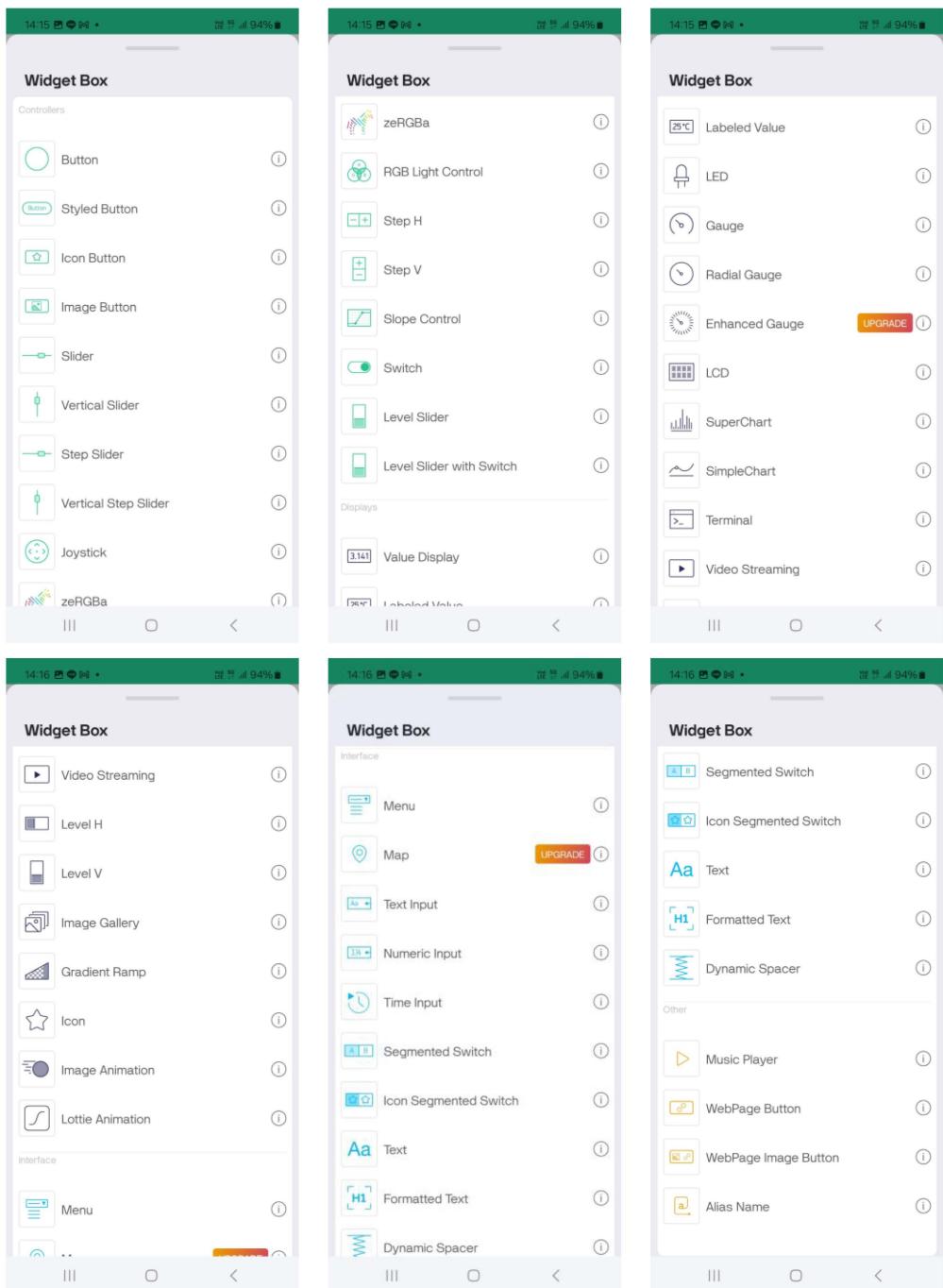
ภาพที่ 10 เลือก Button

เมื่อคลิกที่ Button แล้วจะปรากฏหน้าจอดังภาพที่ 11 เพื่อเลือก Widgets สำหรับแสดงผลและควบคุมอุปกรณ์



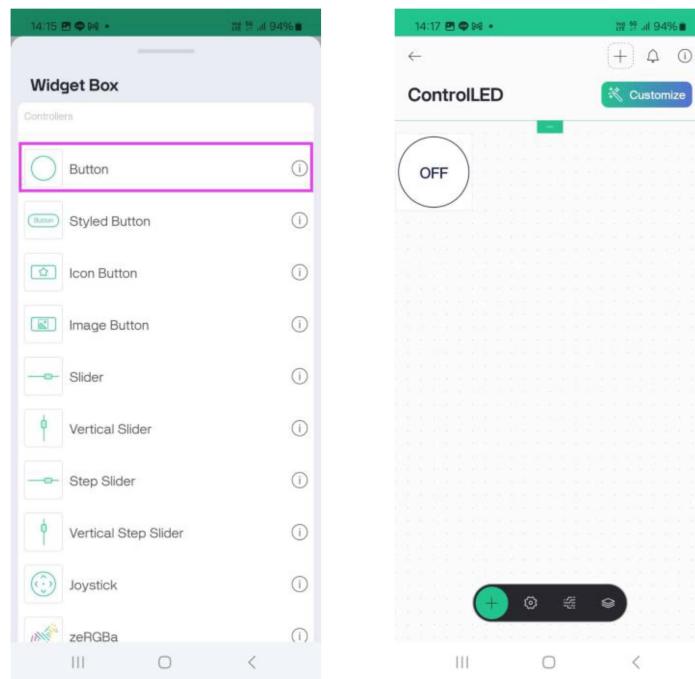
ภาพที่ 11 Widgets

ภายใน Widget Box จะมีให้เลือกหลากหลาย ดังภาพที่ 12



ภาพที่ 12 Widget Box

เลือก Button ดังภาพที่ 13 (ก) จะปรากฏหน้าจอดังภาพที่ 13 (ข)

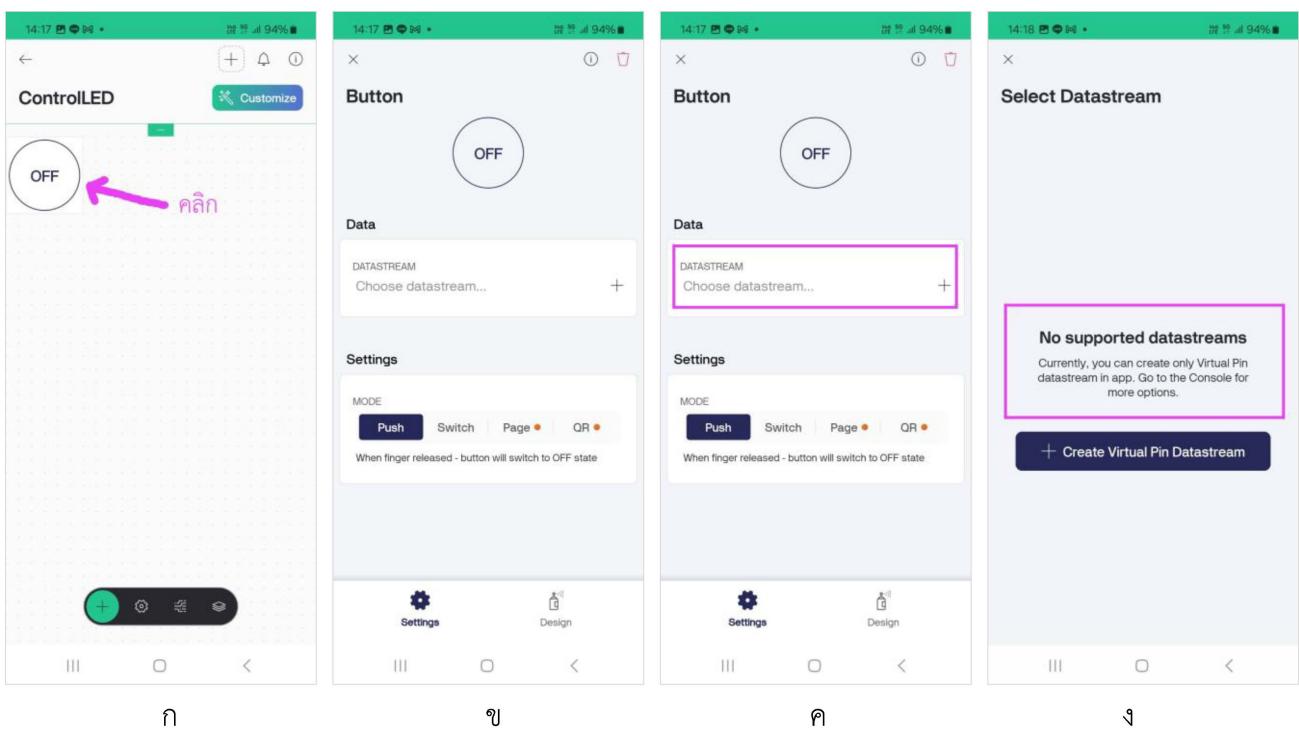


ก

ข

ภาพที่ 13 เลือก Button ใน Widget Box

เมื่อคลิกที่ปุ่ม OFF ในภาพที่ 14 (ก) เพื่อตั้งค่าให้กับ ControlLED จะปรากฏดังภาพที่ 14 (ข) จากนั้นคลิกที่ DATASTREAM ในภาพที่ 14 (ค) จะปรากฏดังภาพที่ 14 (ง) ซึ่งยังไม่ถูกสร้าง จะไปสร้างผ่านเว็บไซต์ รายละเอียดตามขั้นตอนลัดด้านไป



ก

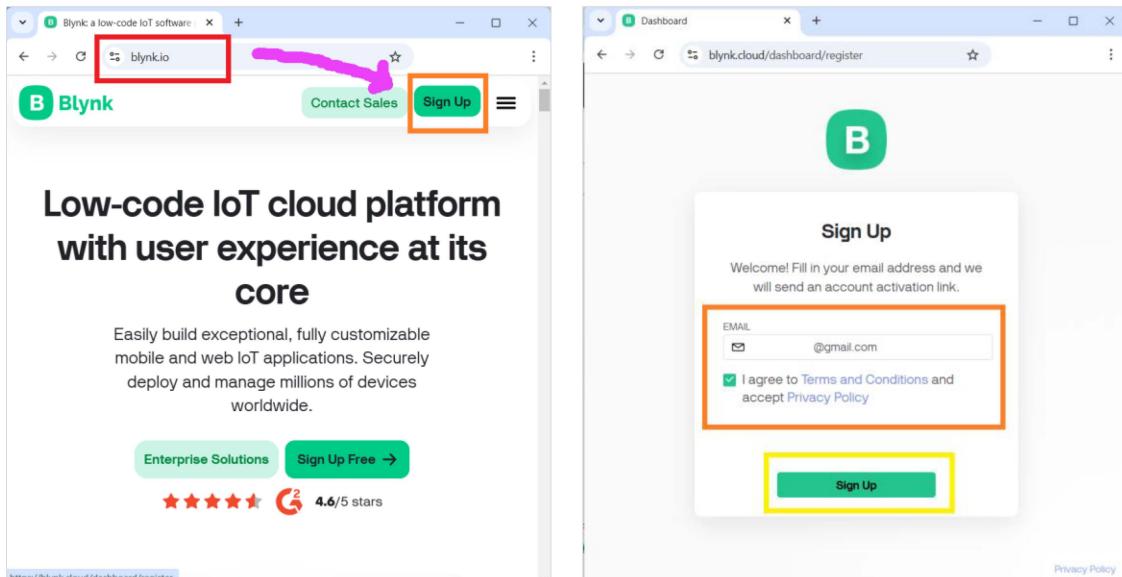
ข

ค

ง

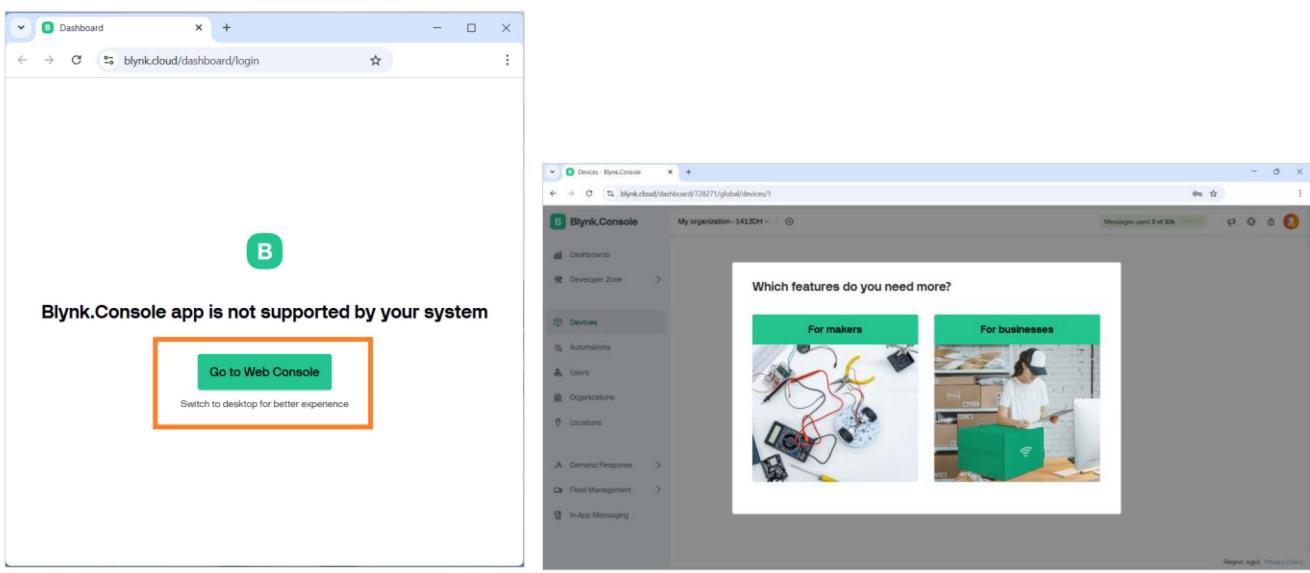
ภาพที่ 14 เลือก Datastreams สำหรับ Button

4. การสร้าง Datastreams ผ่านเว็บไซต์ เข้าสู่เว็บไซต์ blynk.io ดังภาพที่ 15 (ก) จากนั้นเข้าสู่บัญชีผู้ใช้ (ตั้งไว้แล้วในสมาร์ตโฟน) ในภาพที่ 15 (ข)



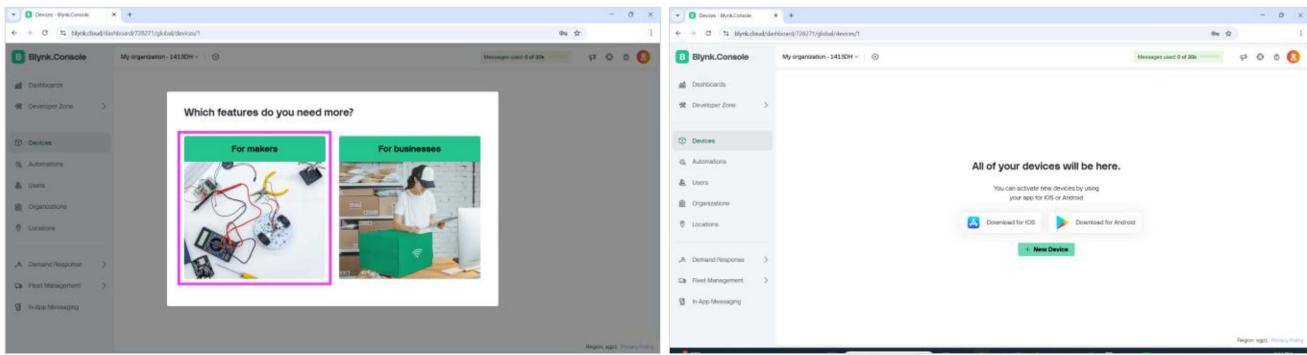
ภาพที่ 15 เว็บไซต์ Blynk

จะปรากฏหน้าจอตั้งภาพที่ 16 (ก) ให้เลือกที่ Go to Web Console จากนั้นจะปรากฏหน้าจอตั้งภาพที่ 16 (ข)



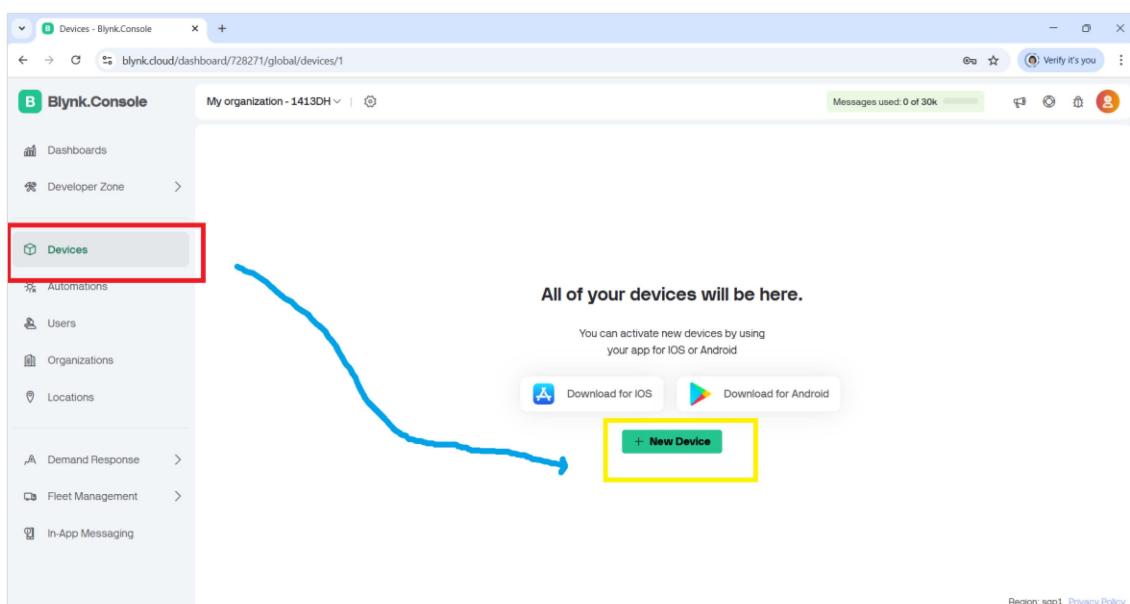
ภาพที่ 16 เลือก Go to Web Console

เลือก For makers ในภาพที่ 17 (ก) จะปรากฏหน้าจอตั้งภาพที่ 17 (ข)



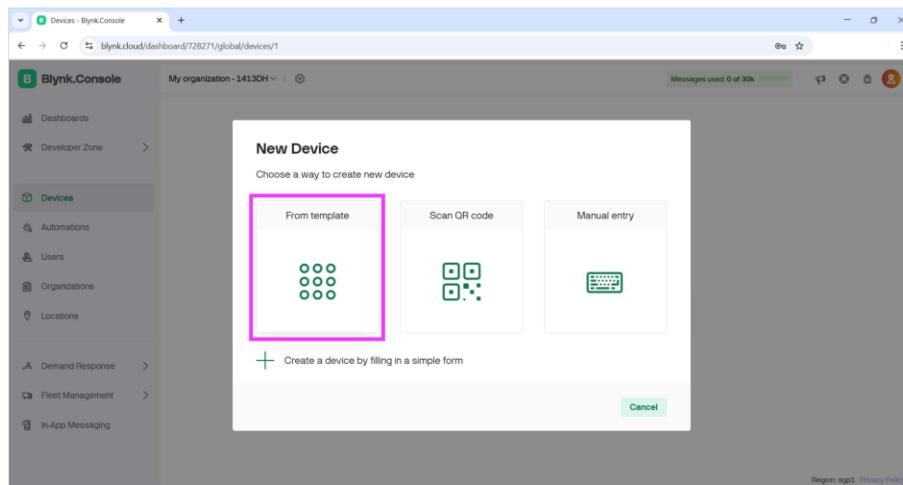
ภาพที่ 17 หน้าจอหลักบนเว็บไซต์

จากนั้น เลือกเมนู Devices ในภาพที่ 18 (สีแดง ด้านซ้ายมือ) และคลิกที่ +New Devices (สีเหลือง)



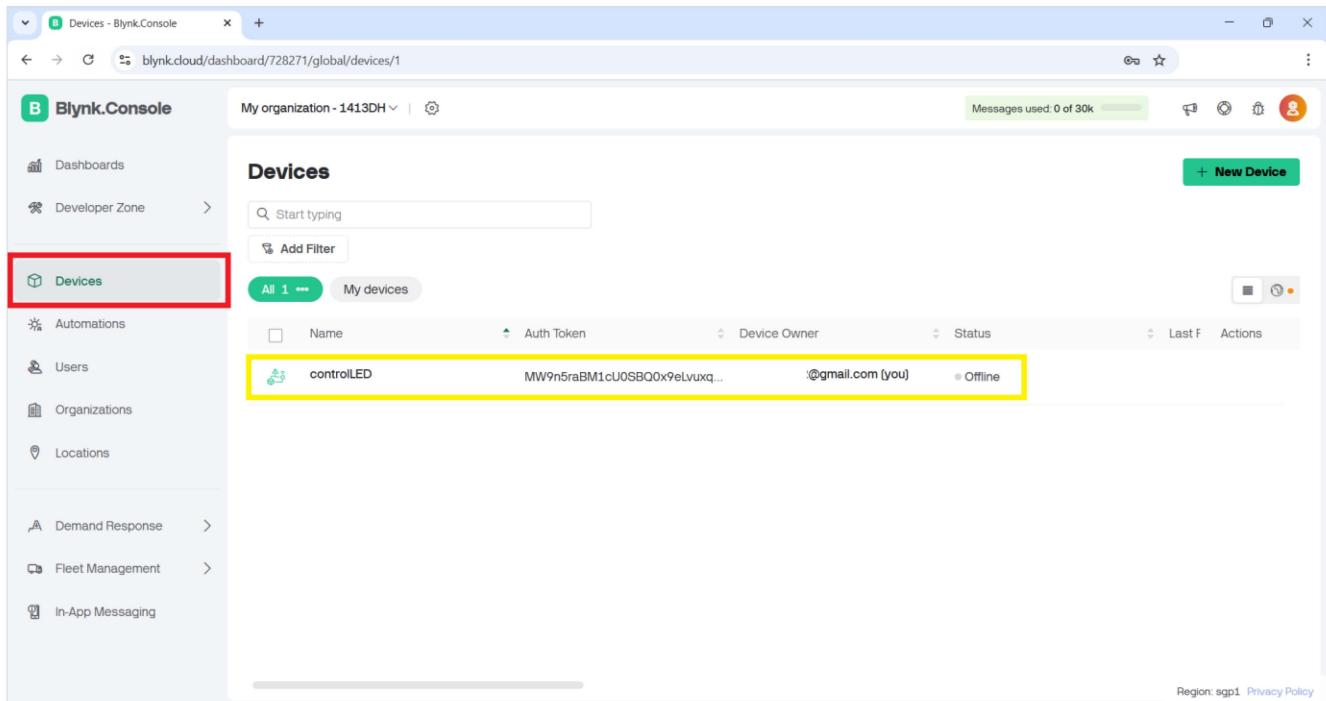
ภาพที่ 18 เลือก Devices

จะปรากฏหน้าจอดังภาพที่ 19 จากนั้นคลิกที่ From template



ภาพที่ 19 New Devices

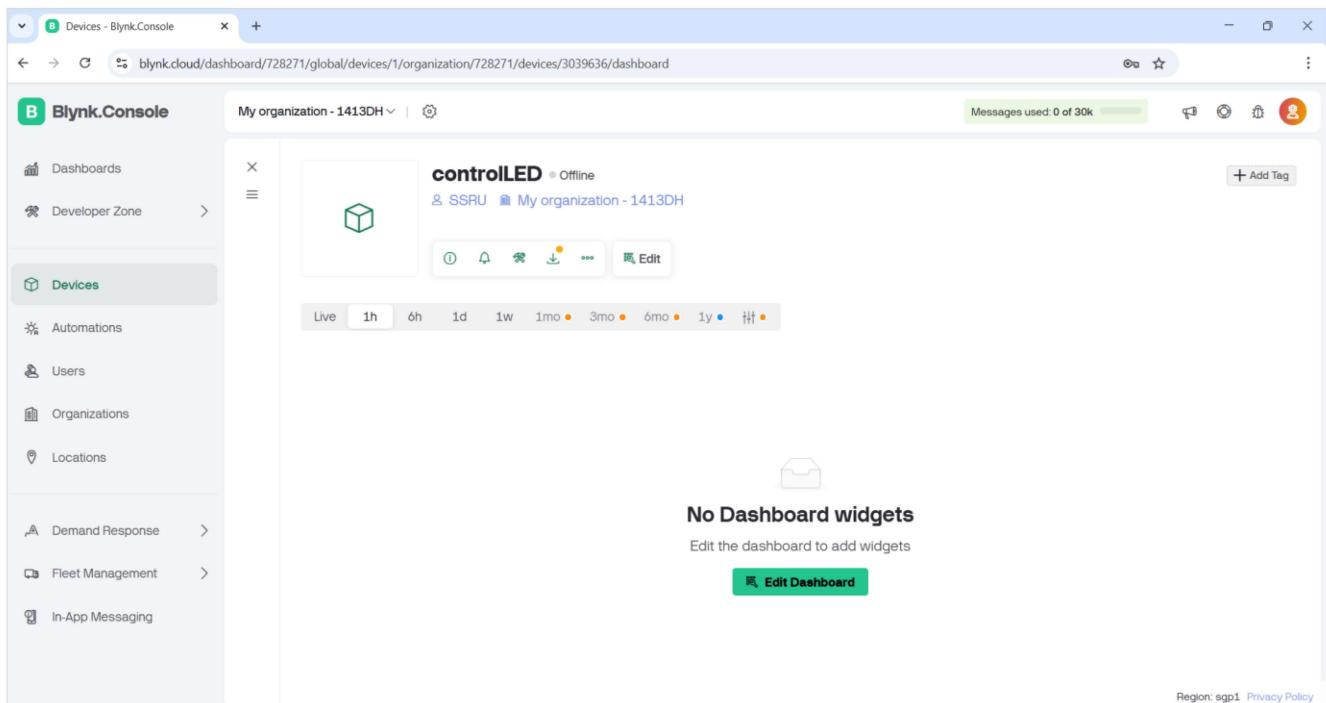
จะปรากฏหน้าจอดังภาพที่ 20 จะสังเกตว่า มี controlLED ที่สร้างในสมาร์ตโฟนปรากฏบนหน้าจอนี้ ต่อไปจะเลือกเมนู Devices (กรอบสีแดงด้านซ้ายมือ) และเลือก controlLED (กรอบสีเหลือง)



The screenshot shows the Blynk Console interface. The left sidebar has a 'Devices' menu item highlighted with a red box. The main area is titled 'Devices' and shows a list of devices. A yellow box highlights the first device in the list, which is named 'controlLED'. The list includes columns for Name, Auth Token, Device Owner, Status, and Last F. The status for 'controlLED' is 'Offline'.

ภาพที่ 20 อุปกรณ์ controlLED

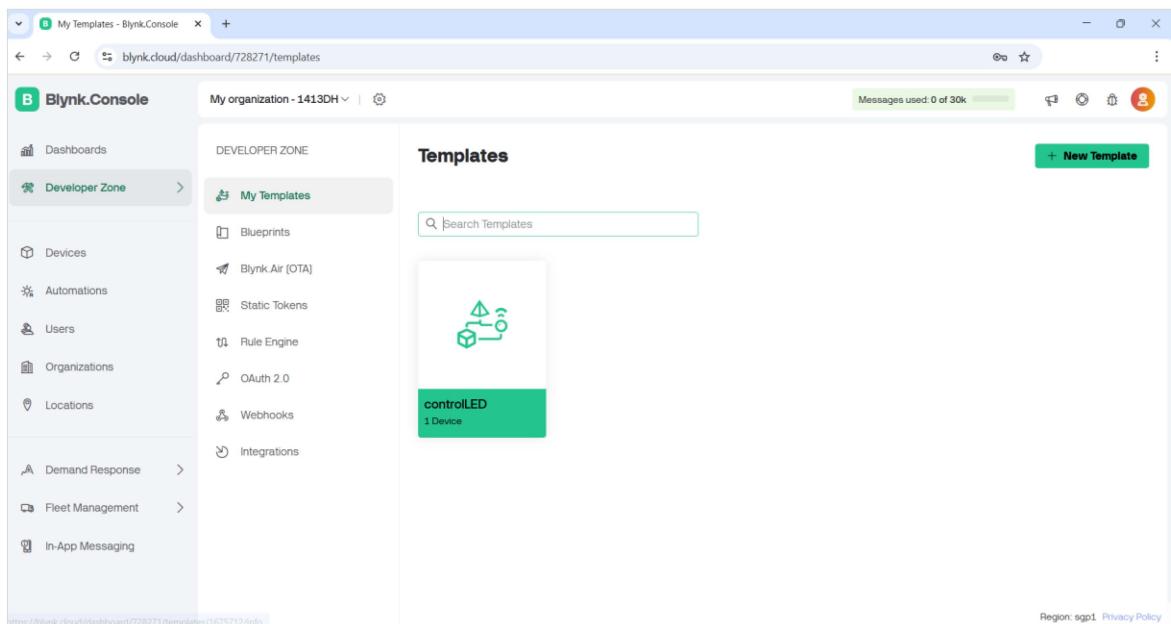
จะปรากฏหน้าจอดังภาพที่ 21 ซึ่งมีรายละเอียดของอุปกรณ์ controlLED



The screenshot shows the Blynk Console interface, specifically the details for the 'controlLED' device. The left sidebar has a 'Devices' menu item highlighted with a red box. The main area shows the device details with a yellow box highlighting the 'controlLED' entry. The device is listed as 'controlLED' with the status 'Offline'. Below the device details, there is a section titled 'No Dashboard widgets' with a green 'Edit Dashboard' button.

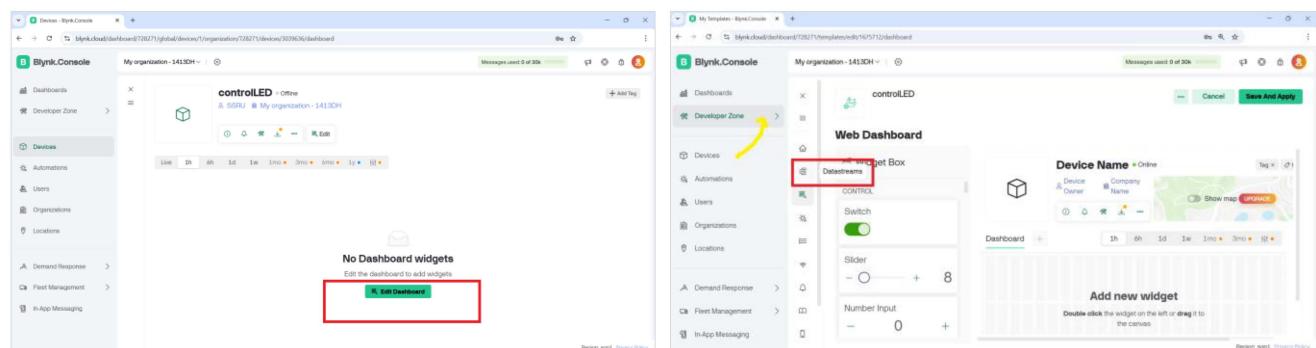
ภาพที่ 21 รายละเอียดของอุปกรณ์ controlLED

ณ จุดนี้หากคลิกไปที่เมนูด้านซ้ายมือ Developer Zone จะปรากฏดังภาพที่ 22



ภาพที่ 22 Developer Zone

ผู้ใช้สามารถกำหนด Datastreams ให้กับอุปกรณ์จากภาพที่ 21 หรือภาพที่ 22 โดยหากใช้ภาพที่ 21 คลิกที่ Edit Dashboard ดังภาพที่ 23 (ก) และสามารถกำหนด Datastreams ได้จากภาพที่ 23 (ข)

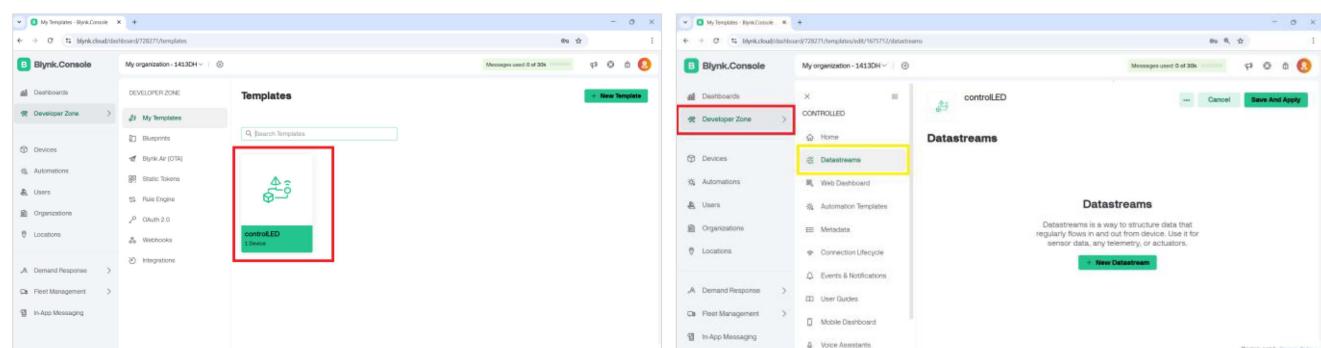


ก

ข

ภาพที่ 23 วิธีเข้าสู่ Datastreams จากภาพที่ 21

ถ้าใช้ภาพที่ 22 คลิกที่ controlLED ดังภาพที่ 24 (ก) และสามารถกำหนด Datastreams ได้จากภาพที่ 24 (ข)

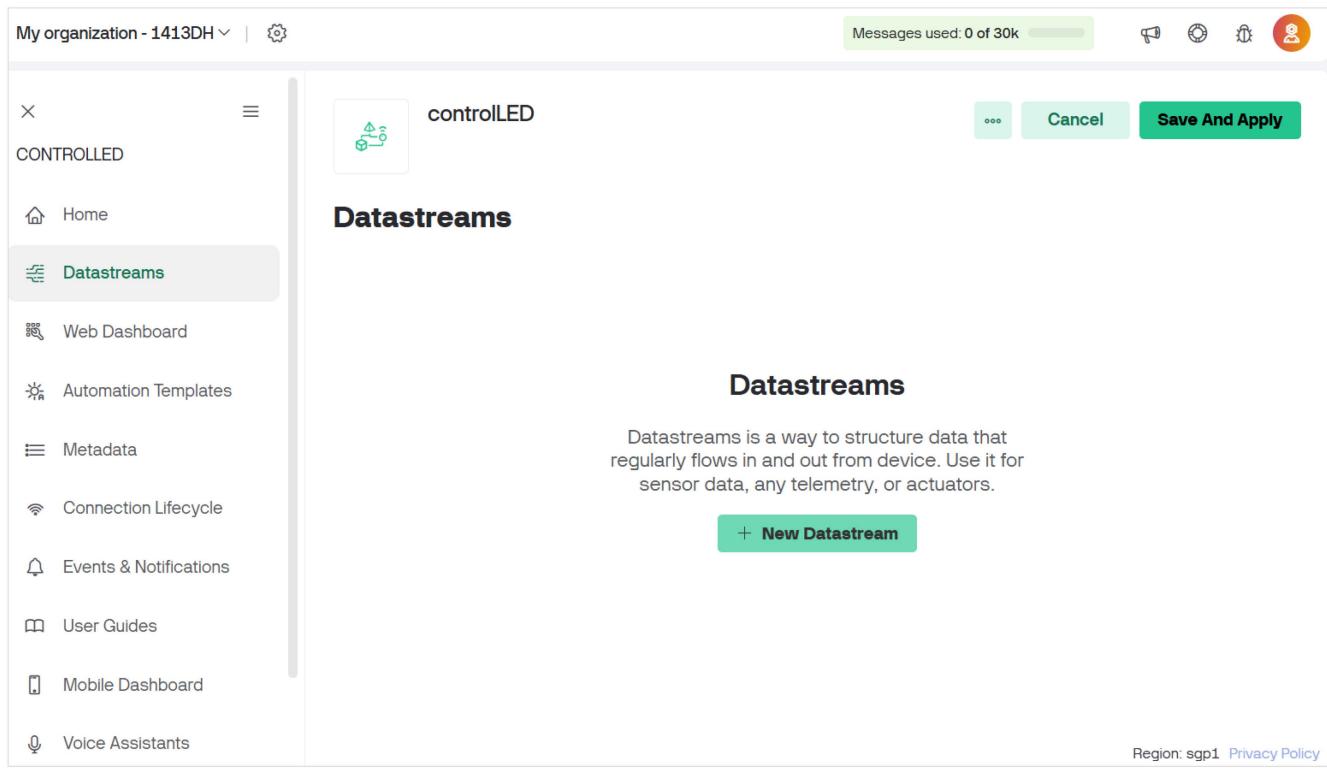


ก

ข

ภาพที่ 24 วิธีเข้าสู่ Datastreams จากภาพที่ 22

จากนั้นจะเข้าสู่หน้าจอ Datastreams ดังภาพที่ 25 และคลิกที่ +New Datastream เพื่อกำหนด Virtual Pin (ภาพที่ 26)



My organization - 1413DH | ⚙️

Messages used: 0 of 30k

...

Cancel

Save And Apply

CONTROLLED

controlled

...

Cancel

Save And Apply

Home

Datastreams

Web Dashboard

Automation Templates

Metadata

Connection Lifecycle

Events & Notifications

User Guides

Mobile Dashboard

Voice Assistants

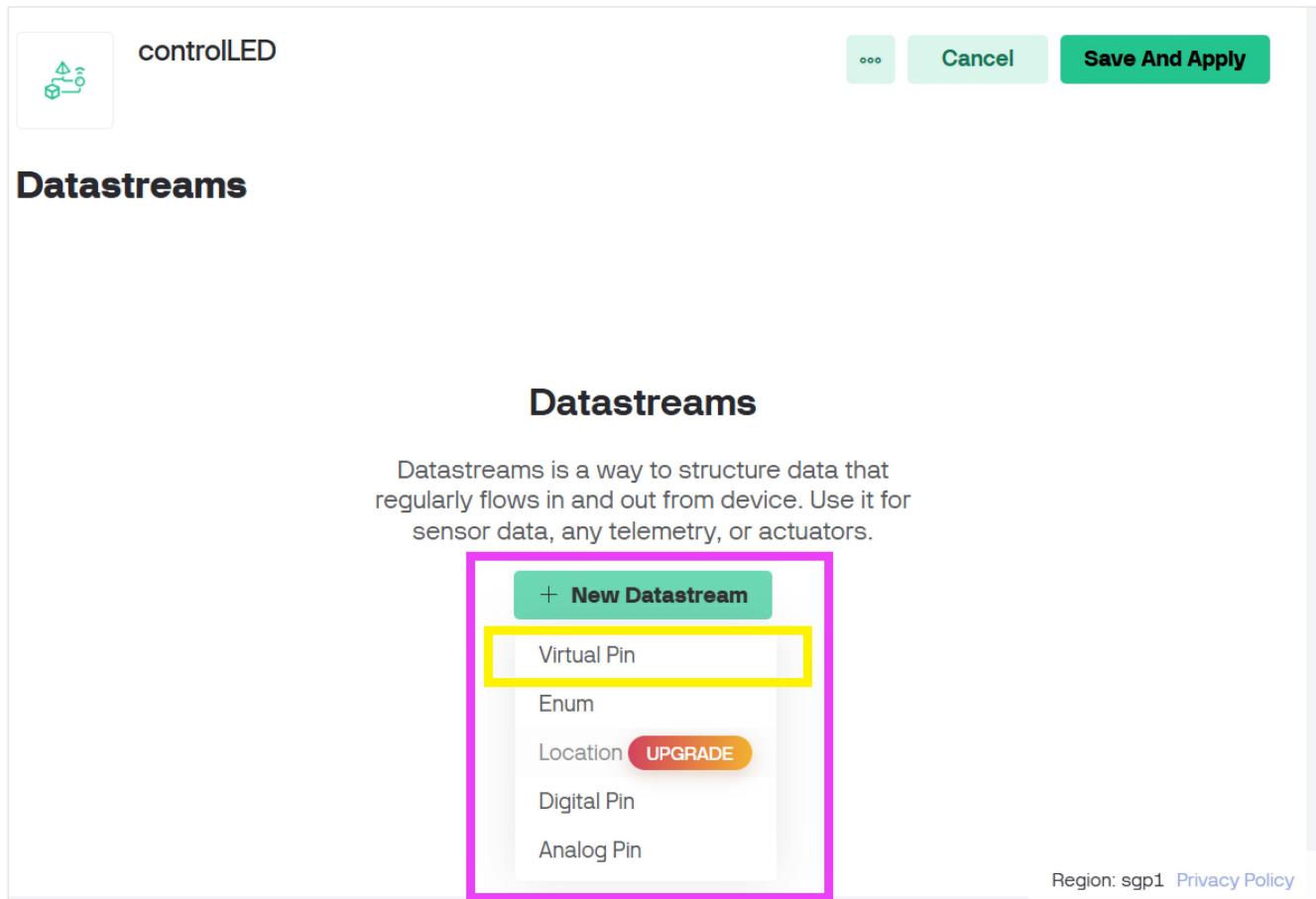
Region: sgp1 Privacy Policy

Datastreams

Datastreams is a way to structure data that regularly flows in and out from device. Use it for sensor data, any telemetry, or actuators.

+ New Datastream

ภาพที่ 25 หน้าจอ Datastreams



controlled

...

Cancel

Save And Apply

...

Cancel

Save And Apply

Datastreams

Datastreams is a way to structure data that regularly flows in and out from device. Use it for sensor data, any telemetry, or actuators.

+ New Datastream

Virtual Pin

Enum

Location UPGRADE

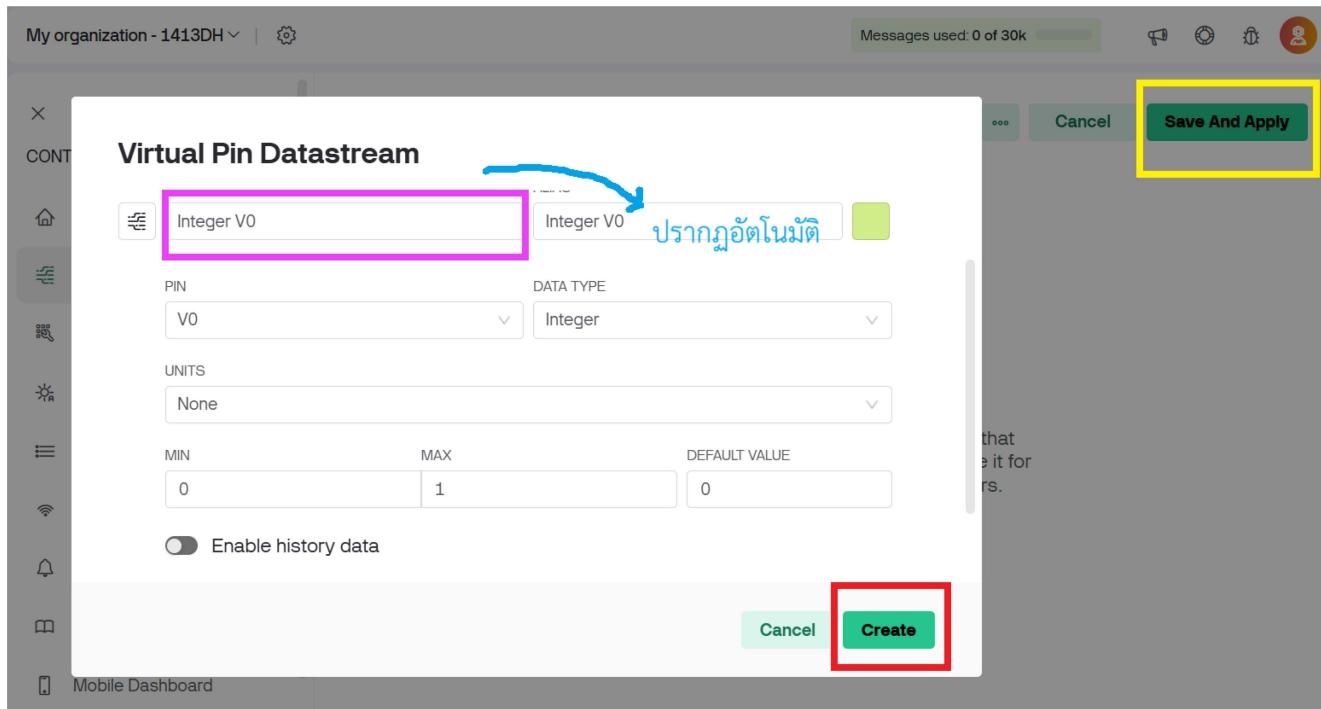
Digital Pin

Analog Pin

Region: sgp1 Privacy Policy

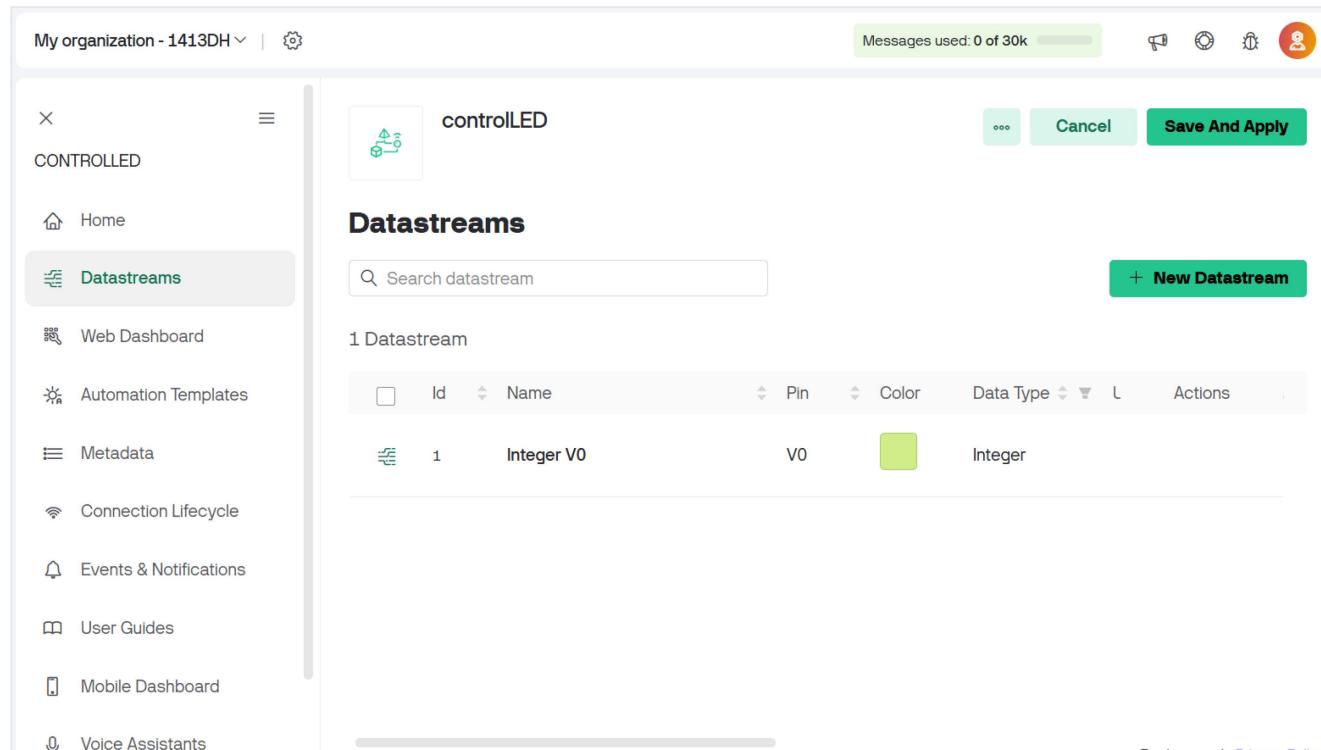
ภาพที่ 26 กำหนด Virtual Pin

ตั้งชื่อตามต้องการ (ภาพที่ 27)



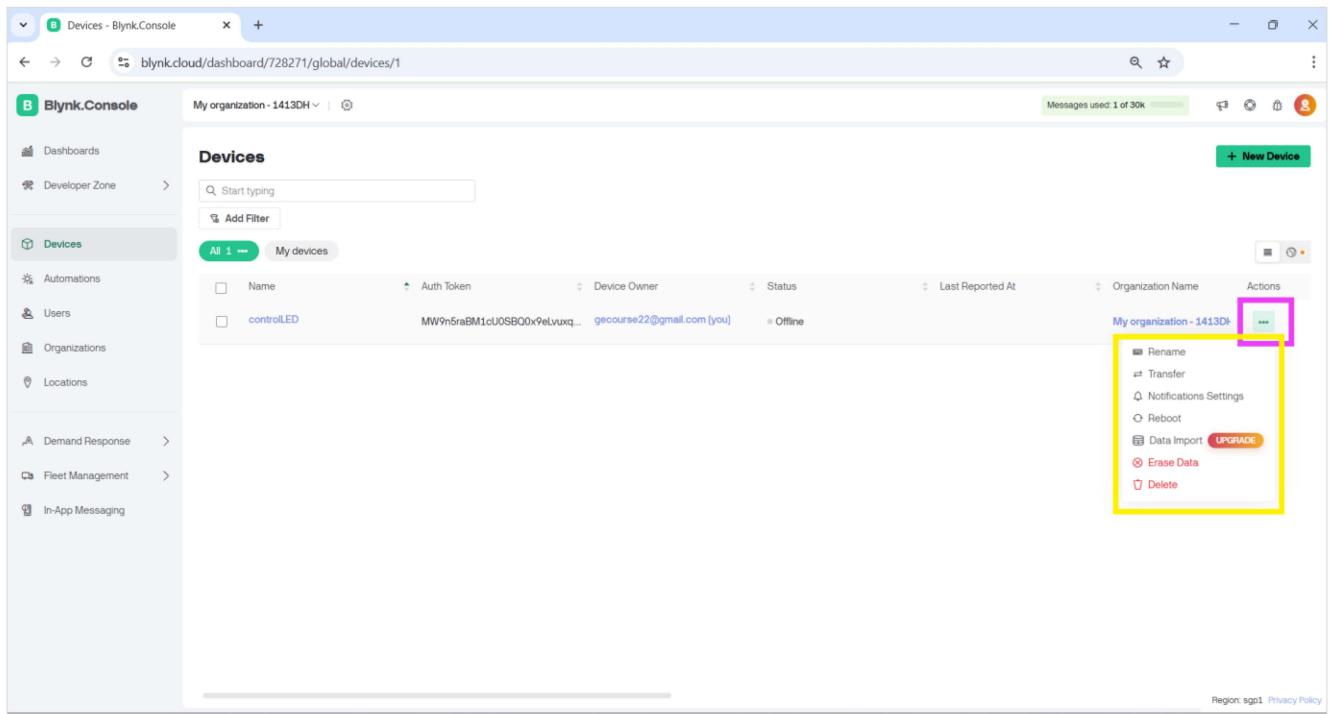
ภาพที่ 27 ตั้งชื่อ Virtual Pin

เมื่อเสร็จสิ้นแล้วจะปรากฏดังภาพที่ 28



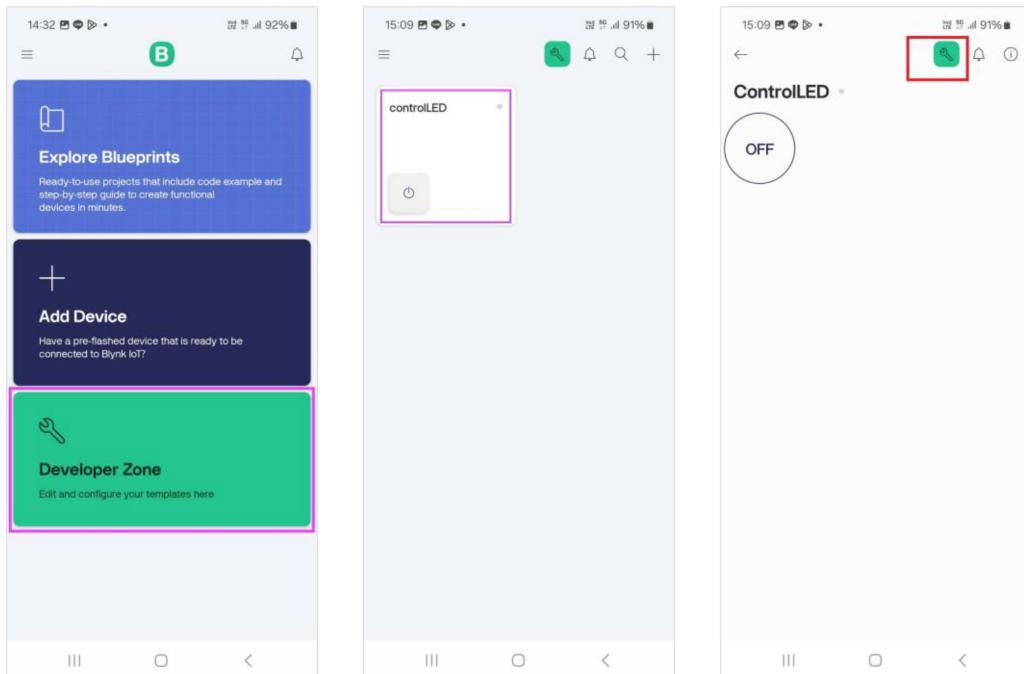
ภาพที่ 28 ตั้งชื่อ Virtual Pin

หากต้องการเปลี่ยนชื่อให้กับอุปกรณ์ (controlLED) สามารถทำได้ดังภาพที่ 29



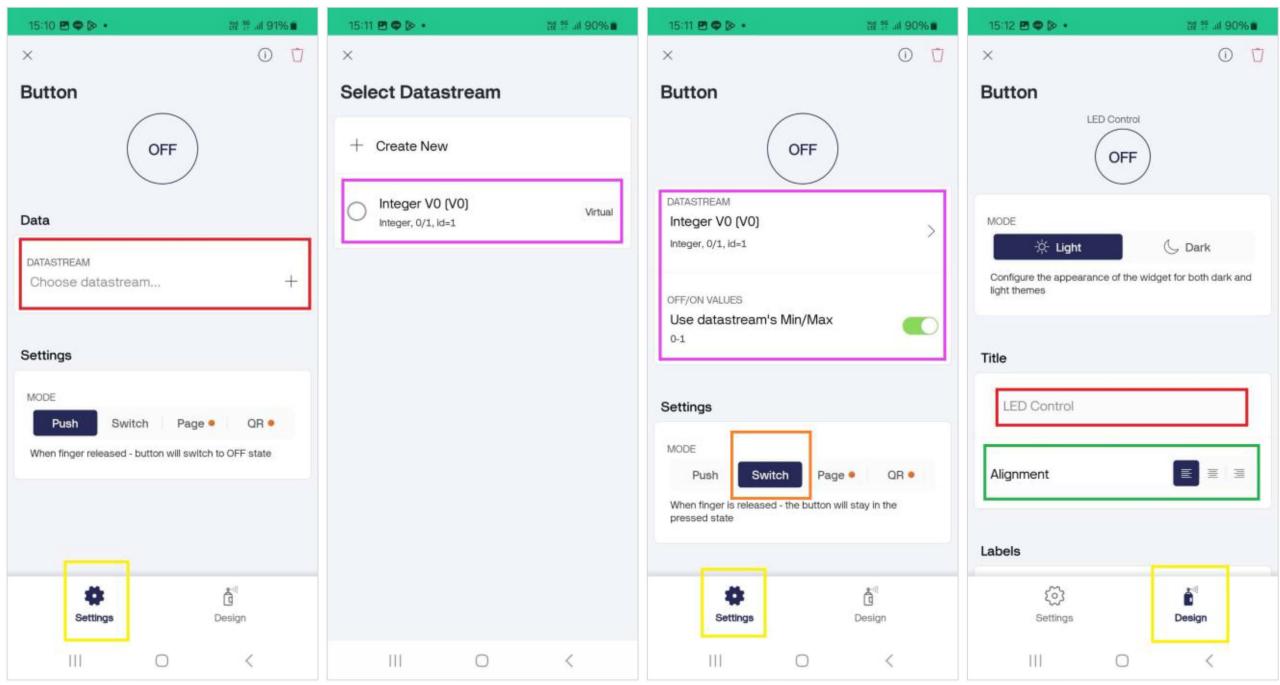
ภาพที่ 29 การเปลี่ยนชื่ออุปกรณ์

เมื่อตั้งค่าให้กับ Datastreams เสร็จแล้วไปที่スマาร์ตโฟน จะปรากฏหน้าจอดังภาพที่ 30 (ก) และคลิกที่ Developer Zone จะปรากฏหน้าจอดังภาพที่ 30 (ข) จากนั้นคลิกบริเวณ controlLED (ภาพที่ 30 ข) จะปรากฏหน้าจอดังภาพที่ 30 (ค) และคลิกที่รูปไข่คงด้านบน (กรอบสีแดง ภาพที่ 30 ค) เพื่อกำหนด Datastreams ให้กับอุปกรณ์



ภาพที่ 30 หน้าจอสมาร์ตโฟนหลังการกำหนด Datastreams ผ่านเว็บไซต์

ตั้งค่า Datastreams ให้กับ LED Control ดังภาพที่ 31 (สามารถปรับแต่งได้)

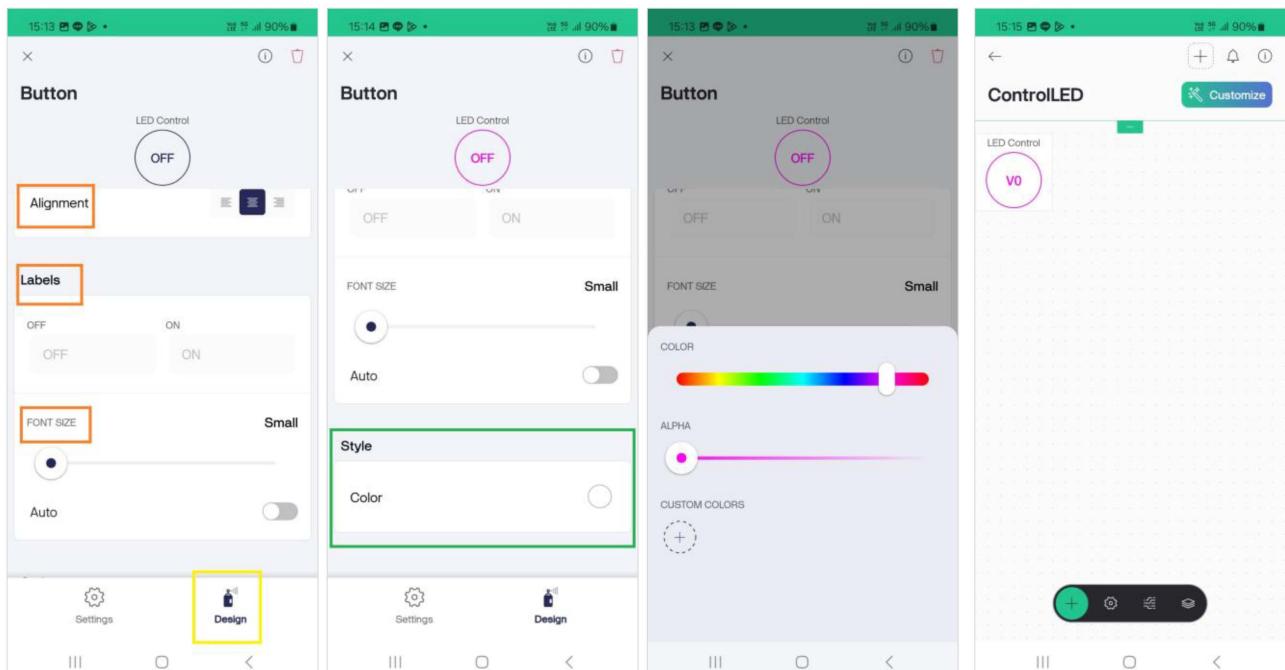


ขั้นตอนที่ 1

ขั้นตอนที่ 2

ขั้นตอนที่ 3

ขั้นตอนที่ 4



ขั้นตอนที่ 5

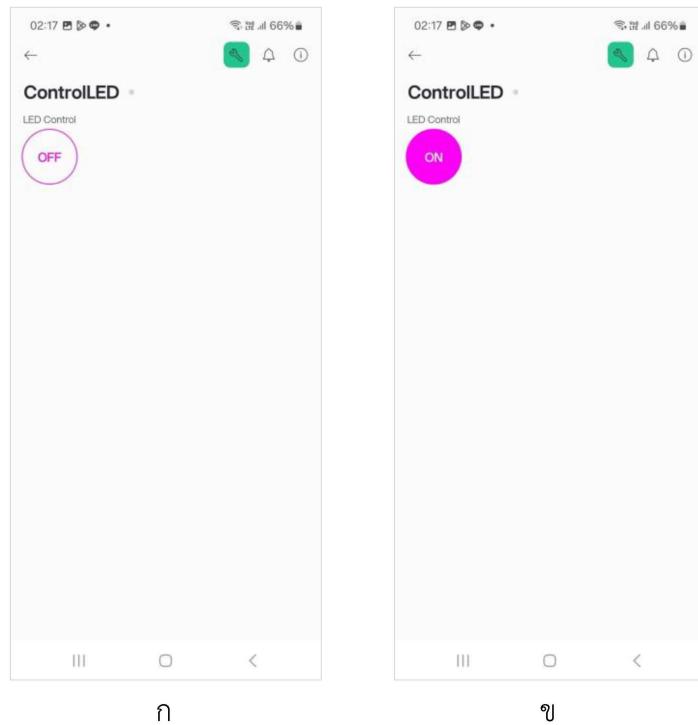
ขั้นตอนที่ 6

ขั้นตอนที่ 7

เสร็จสิ้น

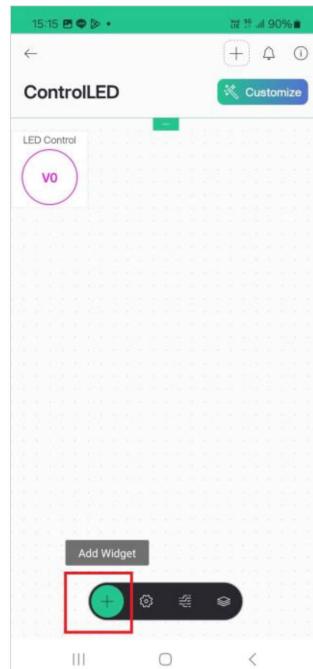
ภาพที่ 31 การกำหนด Datastreams และปรับแต่งหน้าจอ

ภาพที่ 32 แสดงหน้าจอบนสมาร์ตโฟน การกดปุ่มที่ LED Control จะ OFF (ภาพที่ 32 ก) และ ON (ภาพที่ 32 ข) สลับกันไป

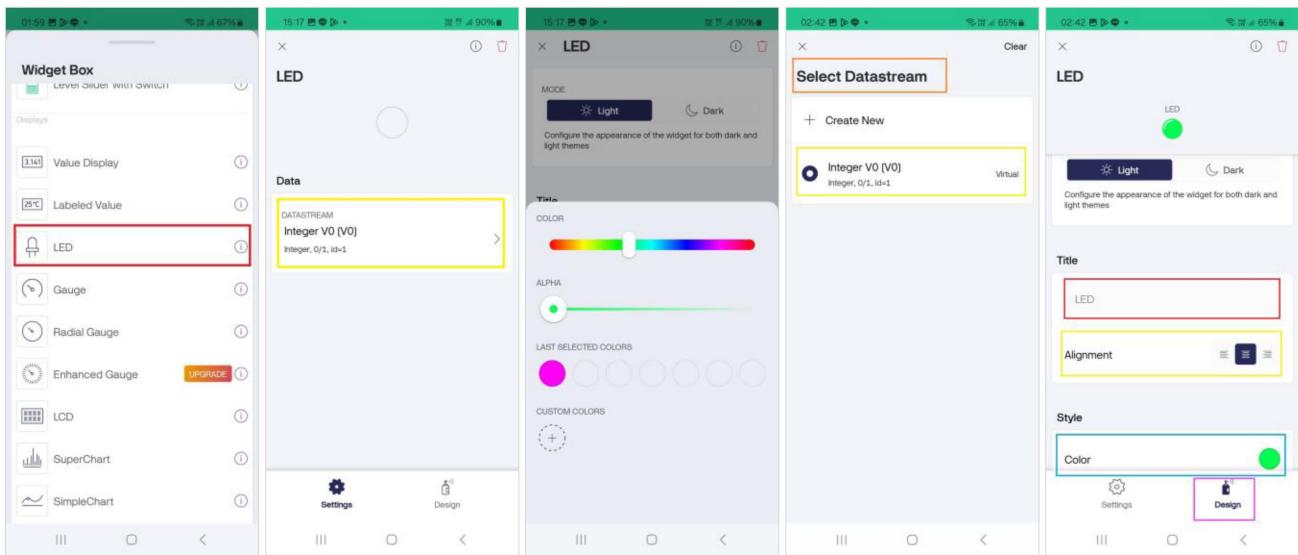


ภาพที่ 32 ภาพหน้าจอสมาร์ตโฟน

เพิ่ม Widget ชื่อ LED เพื่อดูสถานะการกดปุ่ม LED โดยกดเครื่องหมาย + ในภาพที่ 33 จำนวนหนึ่งตามขั้นตอนในภาพที่ 34

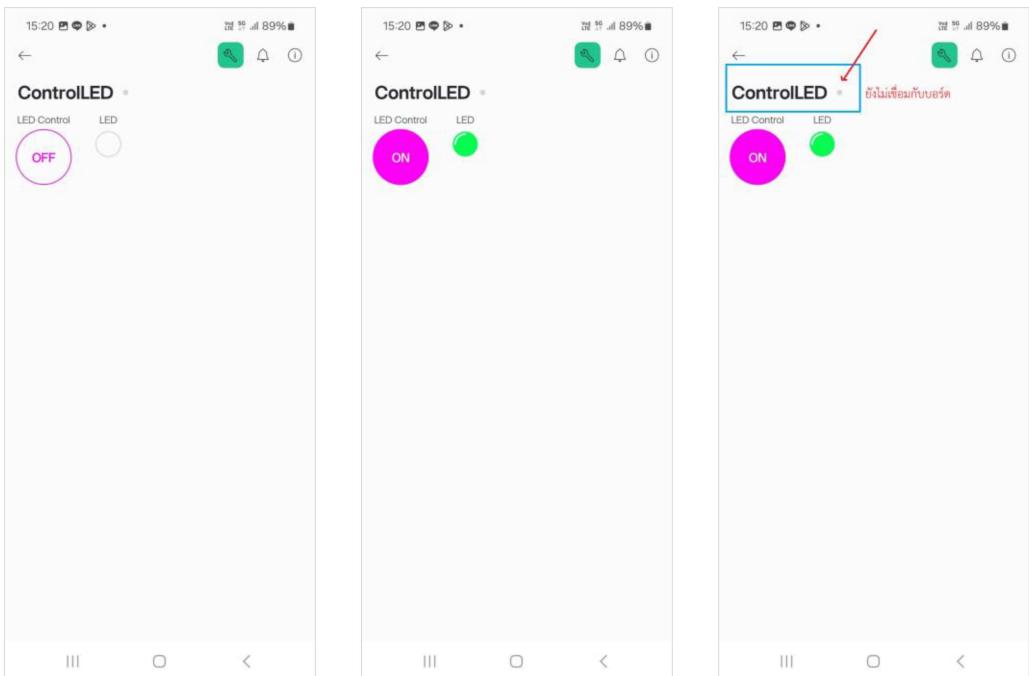


ภาพที่ 33 การเพิ่ม Widget



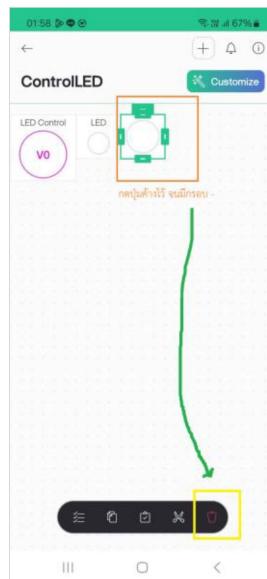
ภาพที่ 34 การปรับแต่ง Widget (LED)

เมื่อปรับแต่ง Widget (LED) เสร็จสิ้นแล้วจะปรากฏดังภาพที่ 35



ภาพที่ 35 หน้าจอสมาร์ตโฟน Widget (controlLED และ LED)

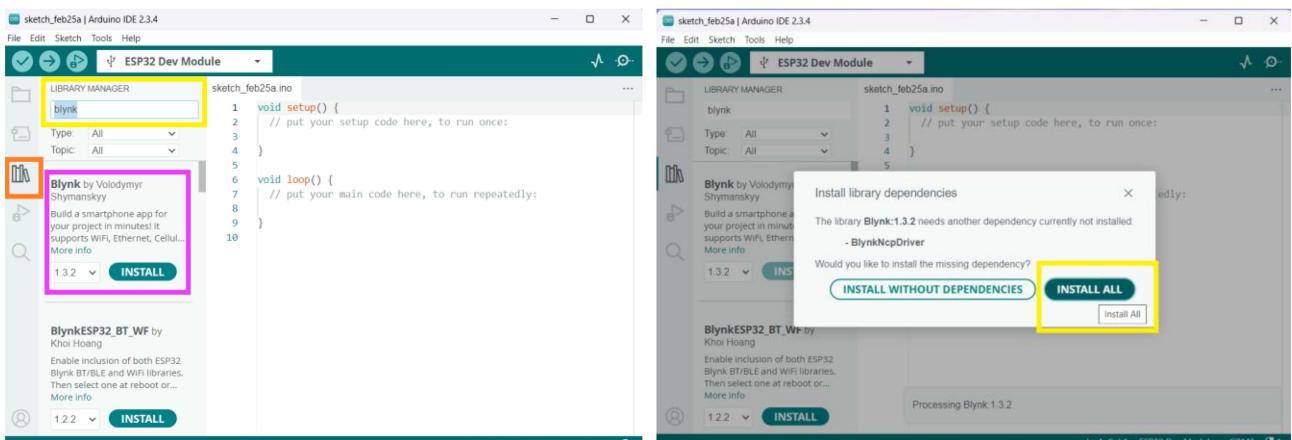
หากต้องการลบ Widget ใดๆ สามารถทำได้โดย คลิกที่ Widget ที่ต้องการลบค้างไว้ จนมีสัญลักษณ์ดังภาพที่ 36 จากนั้น กดปุ่ม delete



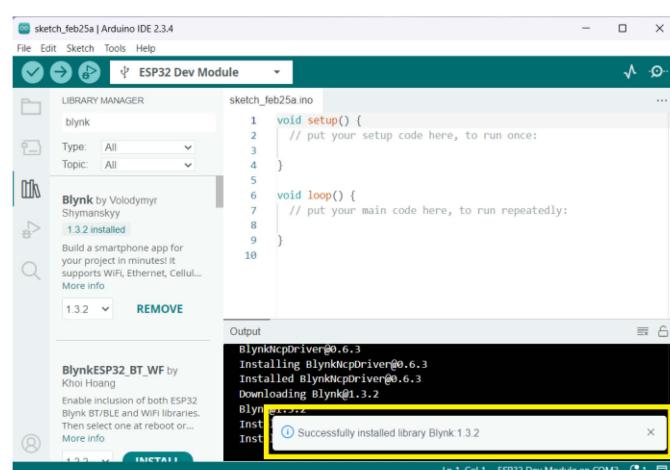
ภาพที่ 36 การลับ Widget

ติดตั้งไลบรารี Blynk สำหรับบอร์ด ESP32

ติดตั้งไลบรารี Blynk สำหรับบอร์ด ESP32 ตามขั้นตอนในภาพที่ 37 และภาพที่ 38 แสดงการติดตั้งเสร็จสมบูรณ์

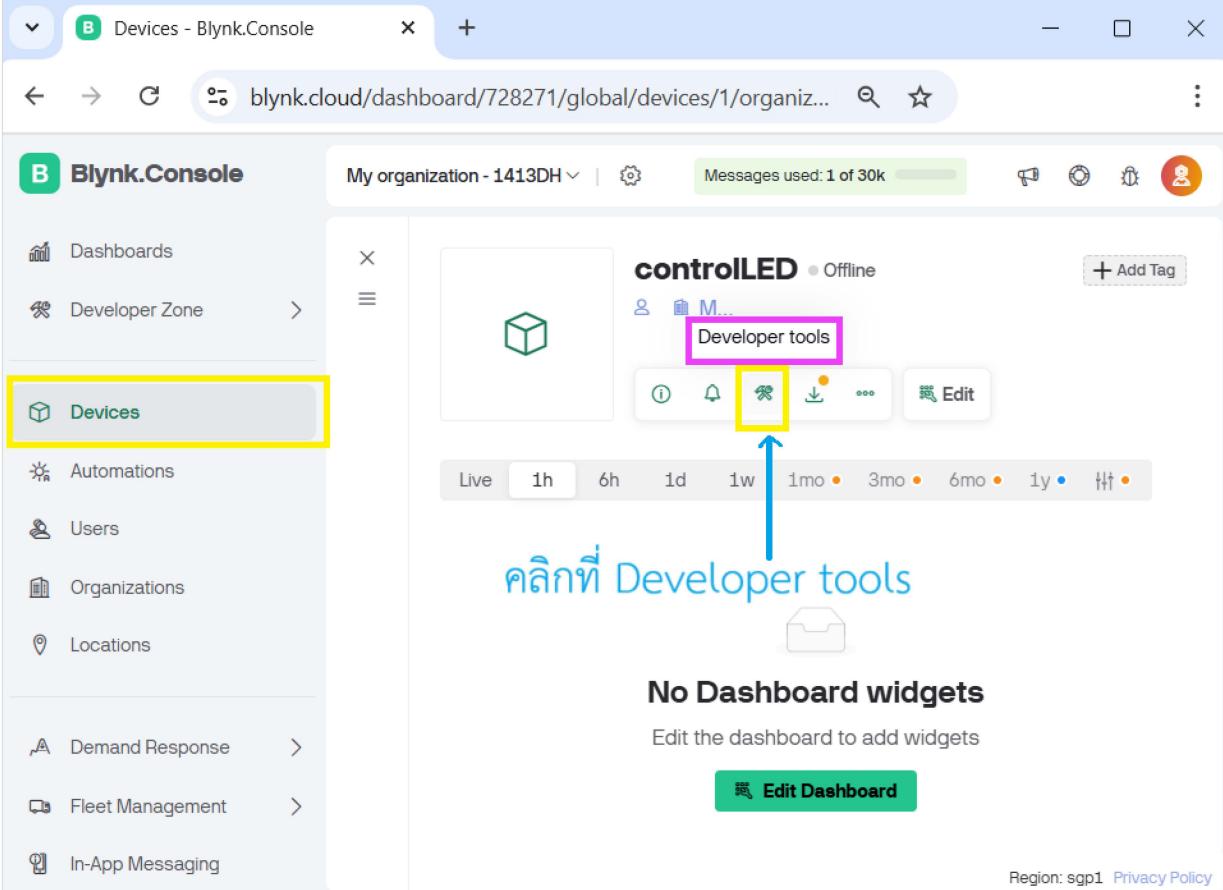


ภาพที่ 37 ติดตั้งไลบรารี Blynk



ภาพที่ 38 ติดตั้งไลบรารี Blynk เสร็จสมบูรณ์

กำหนดตำแหน่ง BLYNK_TEMPLATE_ID , BLYNK_TEMPLATE_NAME, BLYNK_AUTH_TOKEN ของ Blynk โดยตรวจสอบได้จากเว็บไซต์ ดังภาพที่ 39



Devices - Blynk.Console

Blynk.Console

My organization - 1413DH

Messages used: 1 of 30k

controlledLED • Offline

Developer tools

Live 1h 6h 1d 1w 1mo 3mo 6mo 1y

Edit

Devices

Automations

Users

Organizations

Locations

Demand Response

Fleet Management

In-App Messaging

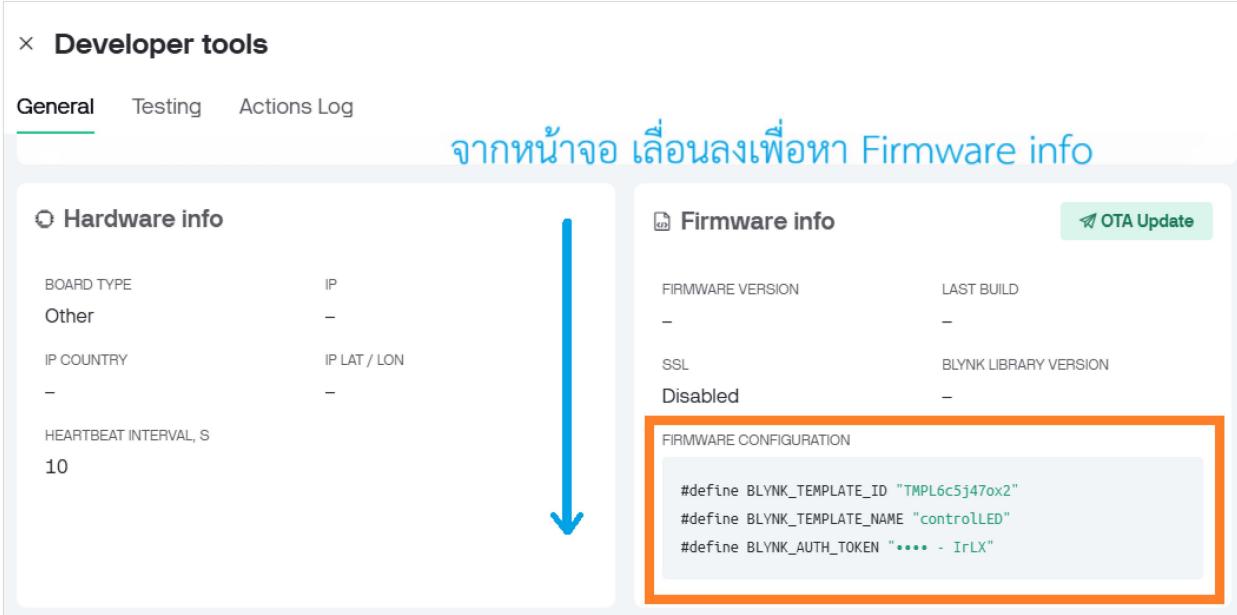
Developer tools

Edit Dashboard

Region: sgp1 Privacy Policy

ภาพที่ 39 ตำแหน่งต่างๆ ของ Blynk

จากนั้นจะปรากฏหน้าจอดังภาพที่ 40 ให้คลิกที่ Developer tools และเลื่อนหน้าจอลงจนพบร Firmware info



Developer tools

General Testing Actions Log

Hardware info

BOARD TYPE: Other IP: -

IP COUNTRY: - IP LAT / LON: -

HEARTBEAT INTERVAL, S: 10

Firmware info

FIRMWARE VERSION: - LAST BUILD: -

SSL: Disabled BLYNK LIBRARY VERSION: -

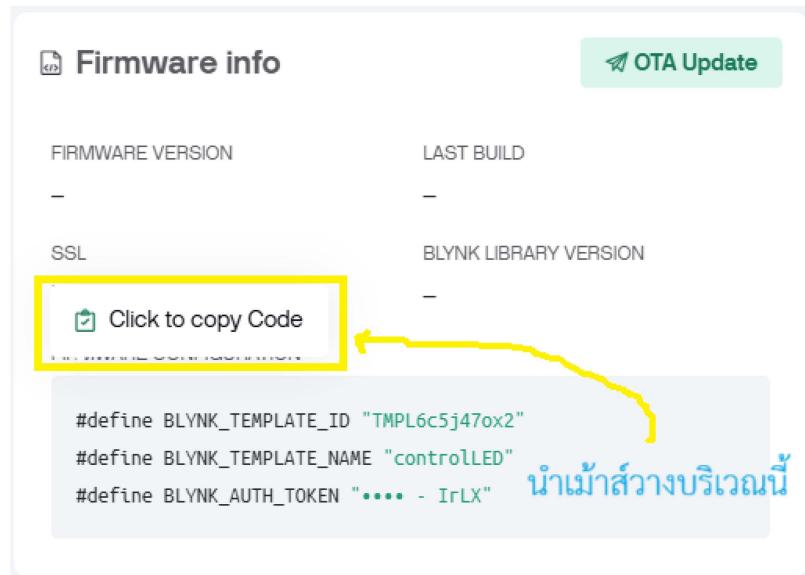
FIRMWARE CONFIGURATION

```
#define BLYNK_TEMPLATE_ID "TMPL6c5j47ox2"
#define BLYNK_TEMPLATE_NAME "controlledLED"
#define BLYNK_AUTH_TOKEN ".... - IrLX"
```

OTA Update

ภาพที่ 40 Firmware info

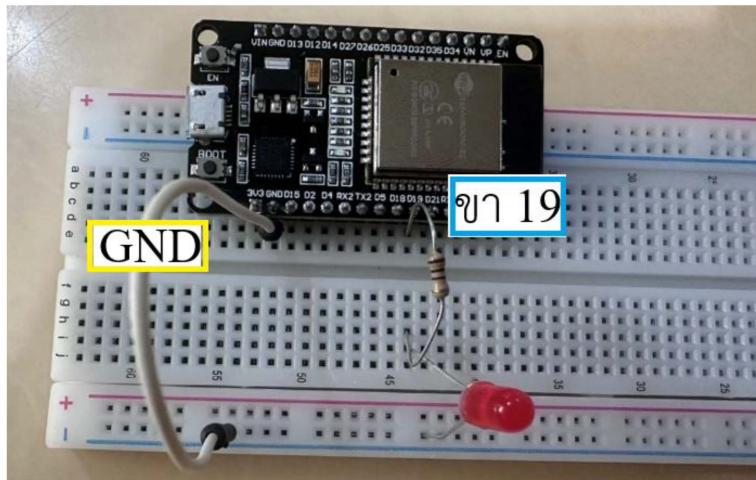
จากหน้าจอ Firmware info สามารถคัดลอกมาวางในโปรแกรมได้ แสดงดังภาพที่ 41



ภาพที่ 41 คัดลอก Firmware info

การทดลองที่ 6.1 การควบคุม LED ผ่าน Blynk

- ต่อวงจรดังภาพที่ 42 โดยให้ขา 19 เป็นเอาต์พุตเชื่อมต่อไปยัง LED (ผ่านความต้านทานค่าต่ำ เพื่อป้องกันความเสียหายให้กับ LED)



ภาพที่ 42 วงจรสำหรับการทดลองข้อ 1

- เขียนโปรแกรมควบคุมการทำงานของ LED ดังภาพที่ 43 (คัดลอกตัวແນ່ງຂອງ Blynk ในภาพที่ 41 มาวางส่วนบนสุดของโปรแกรม ****ค่านี้จะแตกต่างกัน ให้ตรวจสอบที่บัญชีผู้ใช้****)

```

1  #define BLYNK_TEMPLATE_ID "TMPL6Ir1MJobL" //ชื่นอยู่กับของแต่ละคน
2  #define BLYNK_TEMPLATE_NAME "controlledLED"//ชื่นอยู่กับของแต่ละคน
3  #define BLYNK_AUTH_TOKEN ""//ชื่นอยู่กับของแต่ละคน
4
5  // Library ของ Blynk
6  #include <WiFi.h>
7  #include <WiFiClient.h>
8  #include <BlynkSimpleEsp32.h>
9
10 // Wi-Fi Credentials
11 char ssid[] = "nUt";           // ชื่อ Wi-Fi
12 char pass[] = "nattapon_nut";   // รหัสผ่าน Wi-Fi
13
14 // กำหนดพิน LED
15 #define LED_PIN 19
16
17 void setup()
18 {
19     // เริ่มต้น Serial Monitor
20     Serial.begin(115200);
21
22     // กำหนดให้พิน LED เป็น Output
23     pinMode(LED_PIN, OUTPUT);
24
25     // เริ่มต้นการเชื่อมต่อ Blynk
26     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
27
28     Serial.println("Connecting to Blynk...");
29 }
30
31 void loop()
32 {
33     // Blynk Run
34     Blynk.run();
35 }
36
37 // พิ้งก์ชั้นควบคุม LED ผ่าน Virtual Pin V0
38 BLYNK_WRITE(V0)
39 {
40     int pinValue = param.asInt(); // รับค่าจากแอป (0 = OFF, 1 = ON)
41     digitalWrite(LED_PIN, pinValue); // ควบคุมพิน LED
42 }

```

ภาพที่ 43 โปรแกรมสำหรับการทดลองข้อ 2

3. บันทึกผลการทดลอง ถ่ายภาพการทดลอง แคปภาพจากหน้าจอคอมพิวเตอร์ และหน้าจอสมาร์ตโฟน (รายบุคคล กลุ่มเดียวกันก็ให้ใช้ภาพจากสมาร์ตโฟนส่วนตัว)

การทดลองที่ 6.2 แสดงค่าอุณหภูมิและความชื้นผ่าน Blynk

การทดลองนี้จะปรับเปลี่ยนแอปพลิเคชัน Blynk บนสมาร์ตโฟน จากที่ใช้ควบคุมการทำงานของ LED ในการทดลองที่ 5.1 และใช้เซนเซอร์ DHT11 สำหรับวัดอุณหภูมิและความชื้น ทำการตามขั้นตอนดังนี้

- เพิ่ม Widgets เพื่อแสดงอุณหภูมิและความชื้นดังภาพที่ 44

เลือก Gauge สำหรับแสดงอุณหภูมิ และ

เลือก Value Display สำหรับแสดงค่าความชื้น



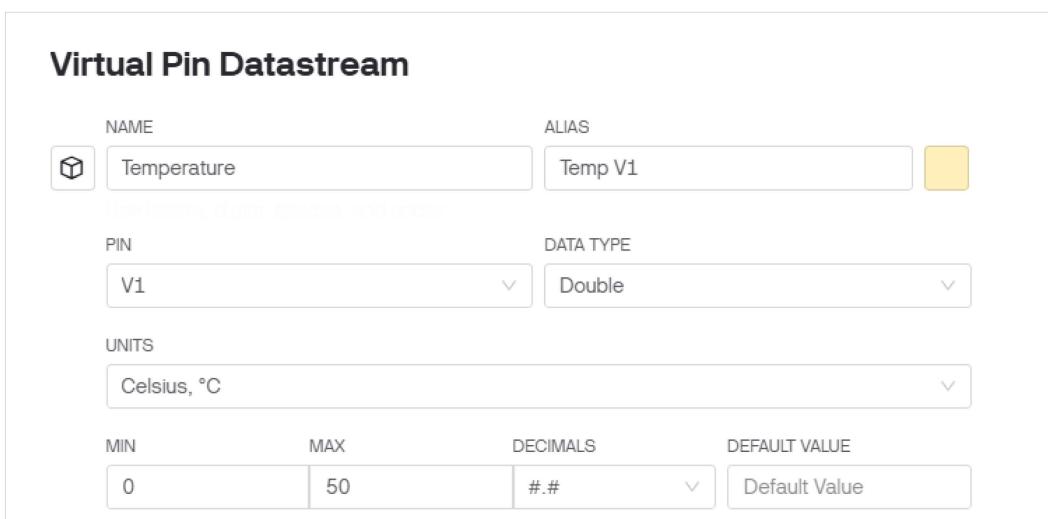
ก เลือก Widget



ข สัญลักษณ์บนหน้าจอ

ภาพที่ 44 เพิ่ม Widgets แสดงอุณหภูมิและความชื้น

- ตั้งค่า Datastreams สำหรับอุณหภูมิ (ตั้งค่าในเว็บไซต์) ดังภาพที่ 45 และสำหรับความชื้นดังภาพที่ 46

A screenshot of the Blynk Virtual Pin Datastream configuration screen. The 'NAME' field is 'Temperature' and the 'ALIAS' field is 'Temp V1'. The 'PIN' field is 'V1' and the 'DATA TYPE' is 'Double'. The 'UNITS' field is 'Celsius, °C'. The 'MIN' field is '0', the 'MAX' field is '50', the 'DECIMALS' field is '#.#', and the 'DEFAULT VALUE' field is 'Default Value'.

ภาพที่ 45 ตั้งค่า Datastreams สำหรับอุณหภูมิ

Virtual Pin Datastream

NAME	ALIAS		
 Humidity	Humidity		
Use letters, digits, spaces, and underscores.			
PIN	DATA TYPE		
V2	Double		
UNITS			
None			
MIN	MAX	DECIMALS	DEFAULT VALUE
0	100	#.##	 Default Value

ภาพที่ 46 ตั้งค่า Datastreams สำหรับความชื้น

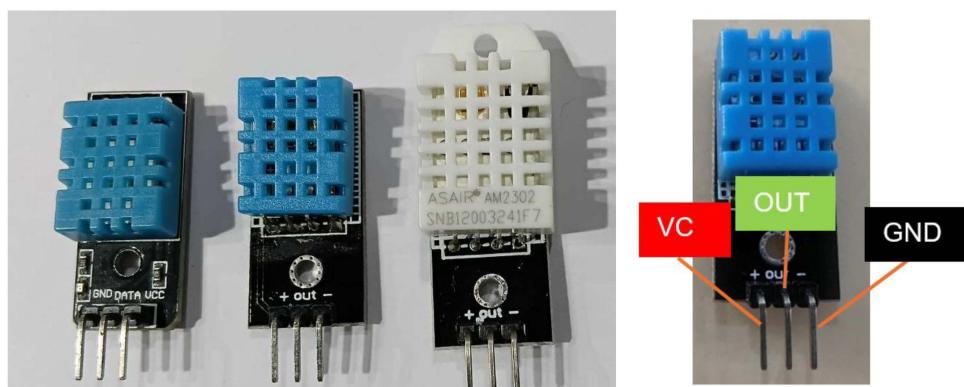
Datastreams ทั้งหมดแสดงดังภาพที่ 47

 controlledLED		Datastreams										   Save And Apply	
		Datastreams										 	
		Datastreams											
		Id	Name	Pin	Color	Data Type	Units	Is Raw	Min	Max	Decimals	Default Value	Actions
	1	Integer V0		V0		Integer		false	0	1	-	0	
	2	Temperature		V1		Double	°C	false	0	50	#.##		
	3	Humidity		V2		Double		false	0	100	#.##		

ภาพที่ 47 Datastreams ทั้งหมด

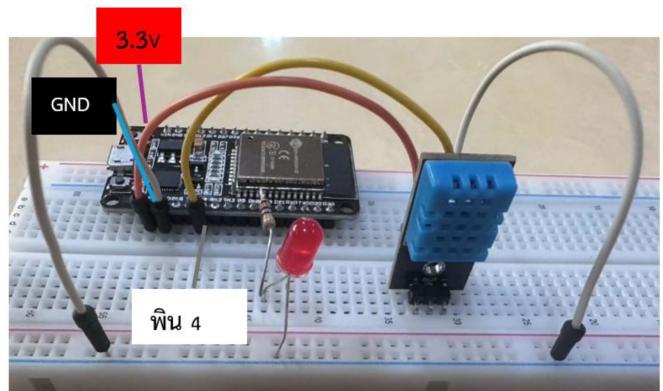
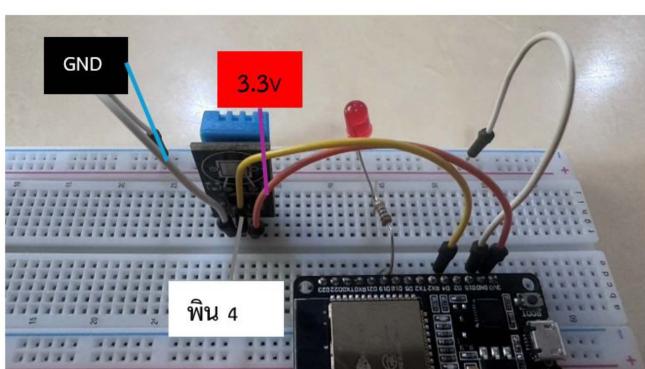
3. ตั้งค่าการแสดงอุณหภูมิและความชื้นบนสมาร์ตโฟนตามที่กำหนดในเว็บไซต์

4. เช่นเซอร์วิส DHT11 เป็นเช่นเซอร์วิสที่วัดได้ทั้งอุณหภูมิและความชื้นสัมพันธ์ ภาพที่ 48 แสดงเช่นเซอร์วิส DHT11



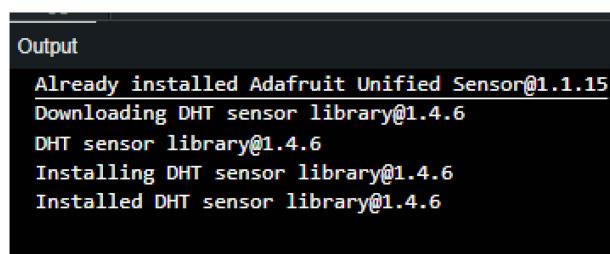
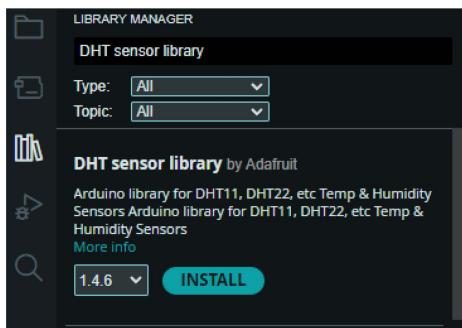
ภาพที่ 48 ตัวอย่างเช่นเซอร์วิสวัดอุณหภูมิและความชื้น

5. ต่อวงจรดังภาพที่ 49 กำหนดขาเอาต์พุตของเซนเซอร์เข้ามิกับคีอ ขา 4

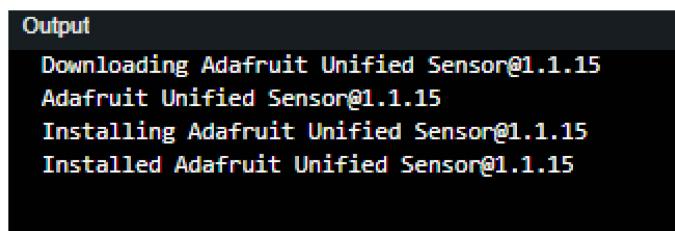
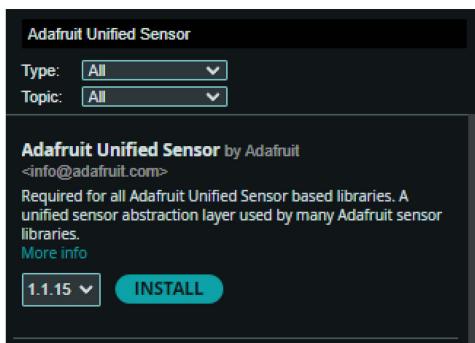


ภาพที่ 49 วงจรสำหรับการทดลองข้อ 5

6. ติดตั้งライบรารี DHT sensor และ Adafruit Unified sensor ดังภาพที่ 50 – 51 เรียงตามลำดับ



ภาพที่ 50 ติดตั้งライบรารี DHT sensor



ภาพที่ 51 ติดตั้งライบรารี Adafruit Unified sensor

7. เขียนโปรแกรมเพื่ออ่านค่าอุณหภูมิ ดังภาพที่ 52 (ดัดлокตามหนังของ Blynk ในภาพที่ 41 มาทางส่วนบนสุดของโปรแกรม ****ค่านี้จะแตกต่างกัน ให้ตรวจสอบที่บัญชีผู้ใช้****)

```

1  #define BLYNK_TEMPLATE_ID "TMPL6Ir1MJobL"
2  #define BLYNK_TEMPLATE_NAME "controlled"
3  #define BLYNK_AUTH_TOKEN "SK9pupq3uR1bsPCV-OELzXjBfb_MJh1U"
4  #include <WiFi.h>
5  #include <WiFiClient.h>
6  #include <BlynkSimpleEsp32.h>
7  #include <DHT.h>
8  char ssid[] = "nUt";           // ชื่อ WiFi
9  char pass[] = "nattapon_nut"; // รหัสผ่าน WiFi
10
11 // กำหนดขาและชานิดของ DHT
12 #define DHTPIN 4                // ขา GPIO ของ ESP32 ที่ต่อกับ DATA ของ DHT11
13 #define DHTTYPE DHT11           // ระบุชนิดเซนเซอร์เป็น DHT11
14 DHT dht(DHTPIN, DHTTYPE);    // สร้างออบเจกต์ dht สำหรับอ่านค่า
15 BlynkTimer timer;           // ตัวตั้งเวลาสำหรับเรียกฟังก์ชันเป็นช่วง ๆ
16
17 void sendDHTData() {
18     // อ่านค่าอุณหภูมิ (องศาเซลเซียส)
19     float t = dht.readTemperature();
20     // อ่านค่าความชื้น (เปอร์เซ็นต์)
21     float h = dht.readHumidity();
22
23     // ตรวจสอบว่าการอ่านค่าติดผลตั้งแต่ไม่ (NaN = Not a Number)
24     if (isnan(t) || isnan(h)) {
25         Serial.println("Failed to read from DHT sensor!");
26         return;
27     }
28     // แสดงข้อมูลใน Serial Monitor
29     Serial.print("Temperature: ");
30     Serial.print(t);
31     Serial.print(" °C, Humidity: ");
32     Serial.print(h);
33     Serial.println(" %RH");
34
35     // ส่งค่าอุณหภูมิไปยัง Virtual Pin V1
36     Blynk.virtualWrite(V1, t);
37     // ส่งค่าความชื้นไปยัง Virtual Pin V2
38     Blynk.virtualWrite(V2, h);
39 }
40
41 void setup() {
42     Serial.begin(115200);
43     dht.begin();
44     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
45     Serial.println("Connecting to Blynk...");
46
47     // ตั้งให้ฟังก์ชัน sendDHTData() ทำงานทุกๆ 2 วินาที
48     timer.setInterval(2000L, sendDHTData);
49 }
50
51 void loop() {
52     Blynk.run();      // ให้ Blynk ทำงาน
53     timer.run();     // ให้ Timer ทำงาน
54 }

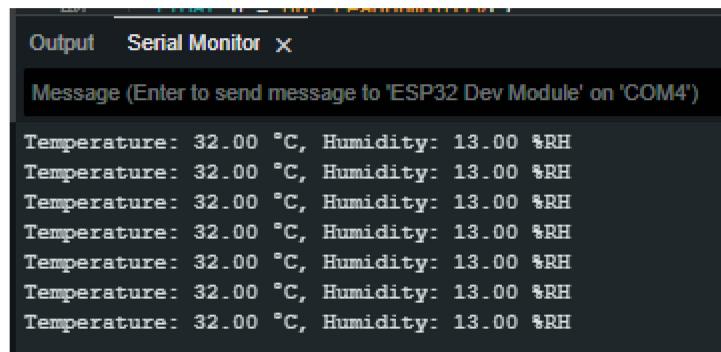
```

ภาพที่ 52 โปรแกรมสำหรับการทดลองข้อ 7

8. เมื่ออัพโหลดโปรแกรมเสร็จสิ้นแล้ว จะแสดงสถานะเชื่อมต่อดังภาพที่ 53 และตัวอย่างผลลัพธ์แสดงบนหน้าจอ serial monitor แสดงดังภาพที่ 54



ภาพที่ 53 หน้าจอสมาร์ตโฟนสำหรับการทดลองที่ 6.2



ภาพที่ 54 ตัวอย่างหน้าจอ serial monitor

9. บันทึกผลการทดลอง ถ่ายภาพการทดลอง แคปภาพจากหน้าจอคอมพิวเตอร์ หน้าจอ serial monitor และหน้าจอสมาร์ตโฟน (รายบุคคล กลุ่มเดียวกันก็ให้ใช้ภาพจากสมาร์ตโฟนส่วนตัว)

รายงานผลปฏิบัติการทดลอง

ให้นักศึกษาส่งรายงานผลปฏิบัติการทดลองเดี่ยว โดยรายงานประกอบด้วย

1. ผลการทดลองที่ให้บันทึกทั้งหมด
2. วิเคราะห์และสรุปผลการทดลอง
3. ส่งรายงานด้วยไฟล์ pdf
4. จัดรูปแบบรายงานกำหนดไว้ในแลป 1

กดส่ง submit ภายในเวลา 11:40 + แนบไฟล์(อัพโหลด) ไม่แนบลิ้ง