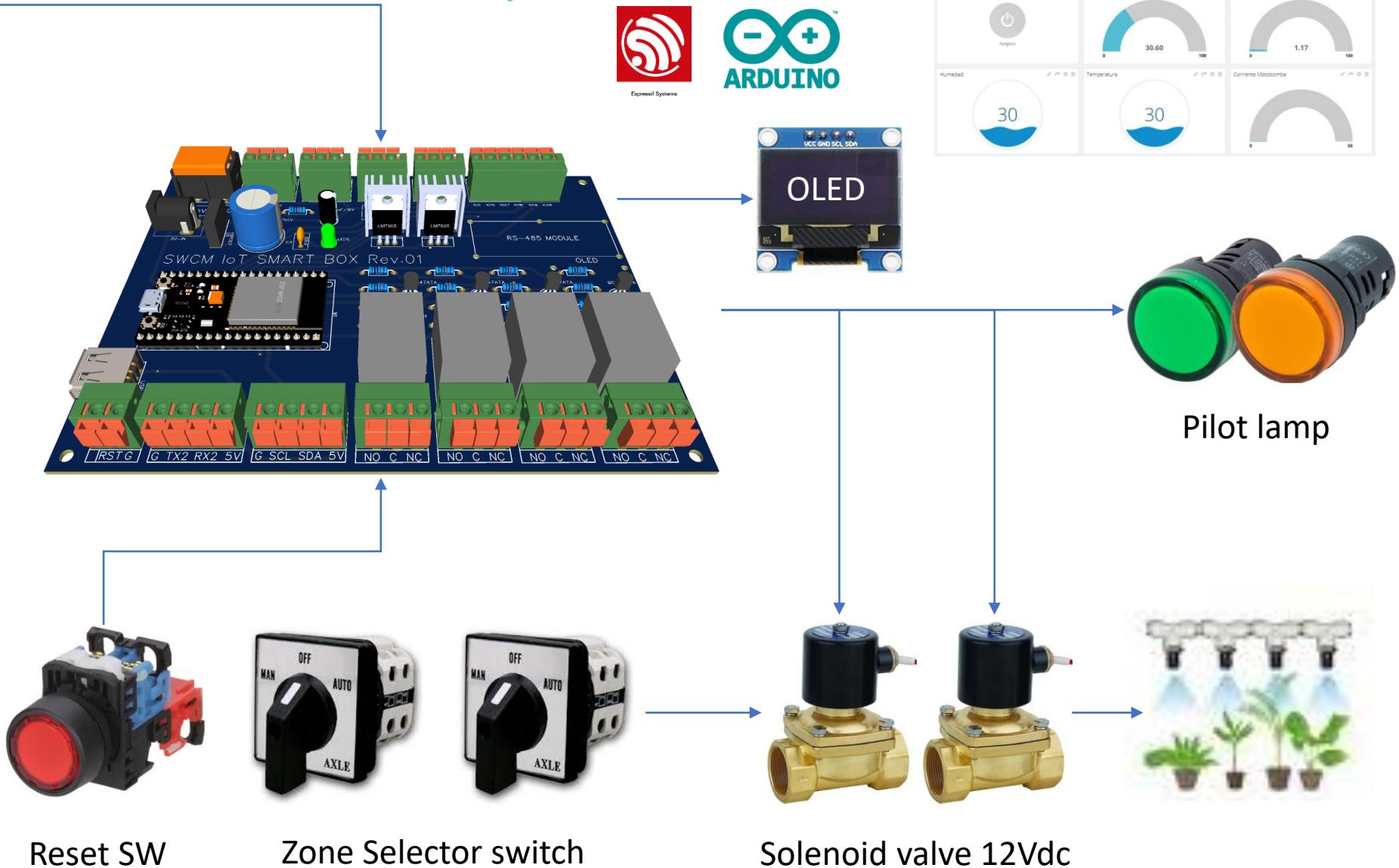


SWCM Smart Farm IoT Box

# SWCM Smart Farm IoT

- **Smart farming** คือ การนำเทคโนโลยีสมัยใหม่เข้ามา ผสมผสานเข้ากับงานด้านการเกษตรเพื่อช่วยแก้ไขปัญหา ต่างๆ ให้กับเกษตรกร หรือผู้ที่สนใจได้นำเอาไป ประยุกต์ใช้ในชีวิตประจำวันหรือสวนของตนเอง เทคโนโลยีฟาร์มอัจฉริยะนั้นต้องยุ่บනแนวคิดของการทำเกษตรสมัยใหม่ที่เรียกว่า เกษตรแม่นยำสูง เป็นกลยุทธ์ ในการทำการเกษตรที่เป็นมิตรกับสิ่งแวดล้อมโดยทำให้เกษตรกรสามารถปรับการใช้ทรัพยากรให้สอดคล้องกับ สภาพของพื้นที่มากที่สุด รวมไปถึงเรื่องการดแลอย่างมี ประสิทธิภาพ ซึ่งแนวคิดนี้สามารถนำไปปรับใช้ได้ทั้ง ฟาร์มแบบเปิดและฟาร์มแบบปิด ฟาร์มอัจฉริยะนี้จะมี ความแตกต่างกับฟาร์มธรรมดาก็อยู่ตรงที่ การใช้ทรัพยากร นั้นทำได้อย่างแม่นยำและตรงต่อความต้องการของพืชที่ เราจะปลูกซึ่งช่วยลดการสูญเสียทรัพยากรและได้ผลผลิต ที่ออกมาตรงตามความต้องการของผู้ดูแลมากที่สุด

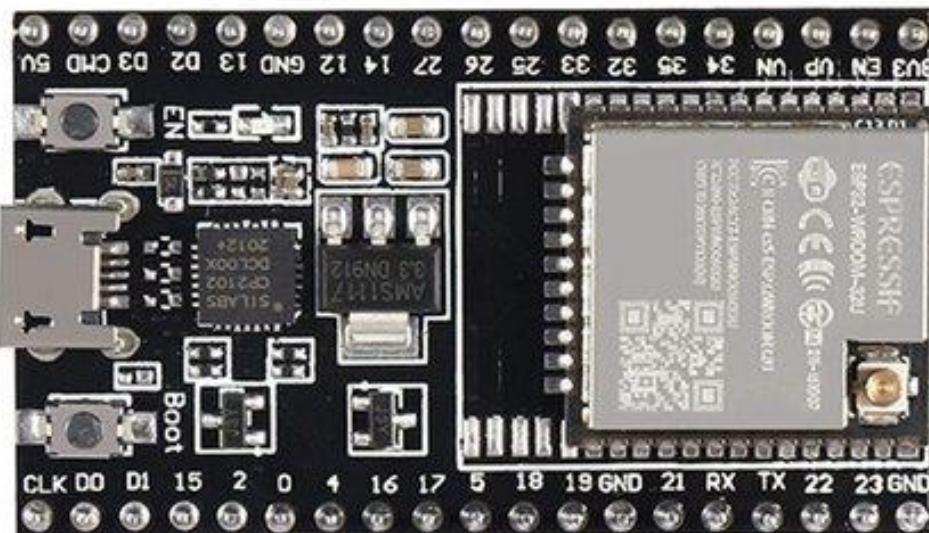
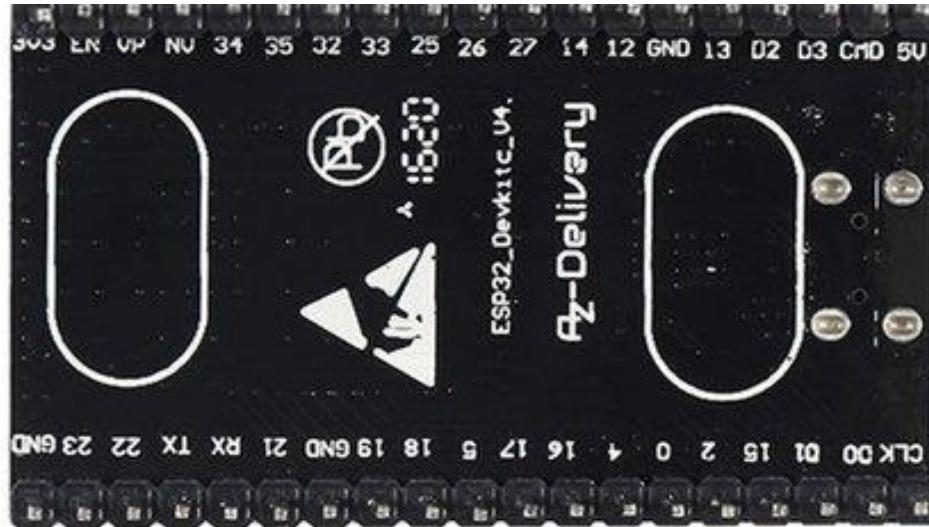
# หลักการทำงาน SWCM Smart Farm IoT

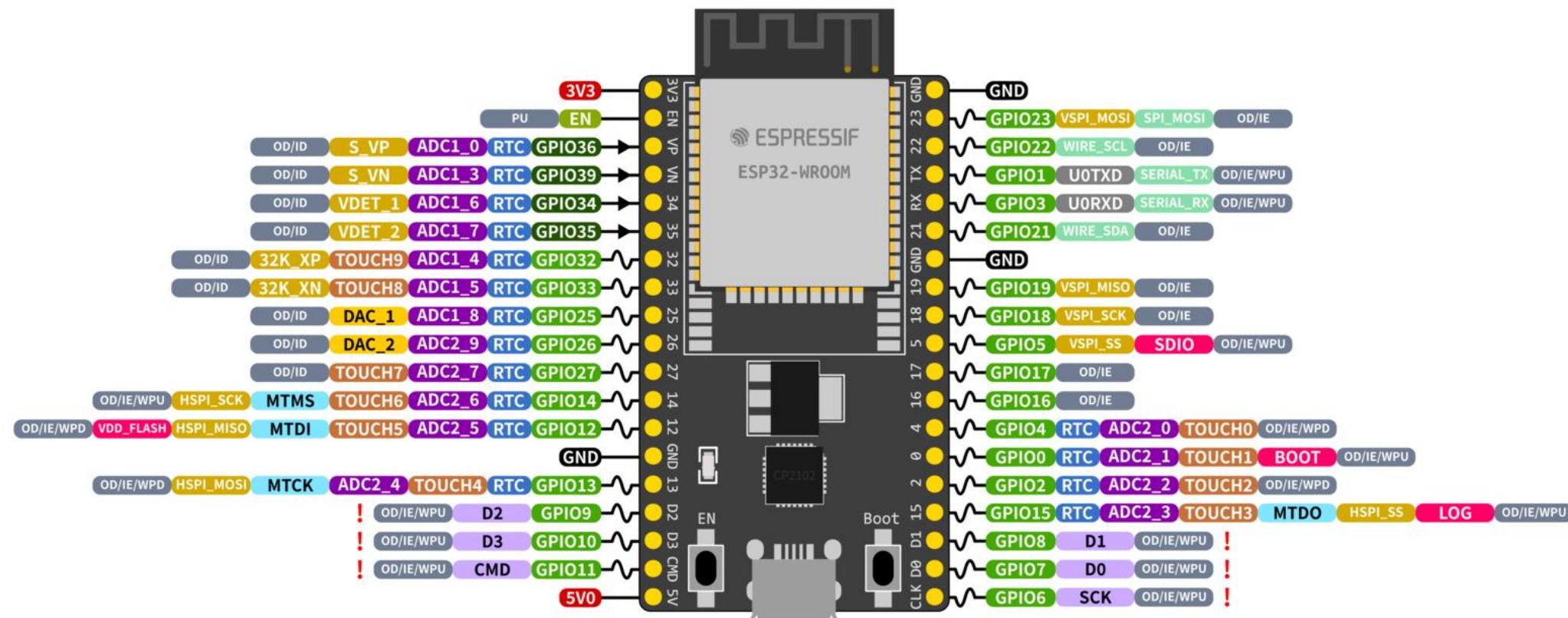


# ESP32-WROOM-32U

## DevkitC V4

- **ESP32-WROOM-32U** พัฒนาโดยบริษัท DOIT มีขาต่อใช้งานทั้งหมด 38 ขา มีโมดูล Wi-Fi + Bluetooth + Bluetooth LE MCU + Touch/Temp Sensor ทำงานแบบ Dual Core มีความเร็ว 160 Mhz มี SRAM 512K หน่วยความจำ Flash สำหรับอัพโหลดโปรแกรมขนาด 16 M มีขา GPIO 38 ขา ความละเอียดในการอ่านค่า ADC 12 Bit สามารถเขียนโปรแกรมผ่าน Arduino IDE เพื่อนำเขียน Arduino ได้



**ESP32 Specs**

32-bit Xtensa® dual-core @240MHz  
 Wi-Fi IEEE 802.11 b/g/n 2.4GHz  
 Bluetooth 4.2 BR/EDR and BLE  
 520 KB SRAM (16 KB for cache)  
 448 KB ROM  
 34 GPIOs, 4x SPI, 3x UART, 2x I2C,  
 2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,  
 1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet

→	PWM Capable Pin
→	GPIO Input Only
→	GPIO Input and Output
→	Digital-to-Analog Converter
→	JTAG for Debugging
→	External Flash Memory (SPI)
→	Analog-to-Digital Converter
→	Touch Sensor Input Channel
→	Other Related Functions
→	Serial for Debug/Programming
→	Arduino Related Functions
→	Strapping Pin Functions

**RTC** RTC Power Domain (VDD3P3\_RTC)  
**GND** Ground  
**PWD** Power Rails (3V3 and 5V)  
 ! Pin Shared with the Flash Memory  
*Can't be used as regular GPIO*

**GPIO STATE**

**WPU:** Weak Pull-up (Internal)  
**WPD:** Weak Pull-down (Internal)  
**PU:** Pull-up (External)  
**IE:** Input Enable (After Reset)  
**ID:** Input Disabled (After Reset)  
**OE:** Output Enable (After Reset)  
**OD:** Output Disabled (After Reset)

# DHT22 / AM2302 Module - โมดูลวัดอุณหภูมิและความชื้น



DHT22 โมดูลเซนเซอร์วัดความชื้นและอุณหภูมิในตัวเดียว มีความแม่นยำสูง มีตัวต้านทาน Pull up มาแล้วสามารถต่อขาทดลองได้เลยไม่ต้องต่อเพิ่ม



ถ้าต้องการความถูกต้องแม่นยำในการวัดอุณหภูมิและความชื้น  
แนะนำตัวที่เลย DHT22 High Accuracy Digital Temperature  
and Humidity Sensor DHT22 ใช้สำหรับวัดอุณหภูมิและ  
ความชื้น ออกแบบมาให้วดได้แม่นยำกว่ารุ่น DHT11 ใช้ง่าย  
สามารถนำ DHT22 ไปเปลี่ยนแทน DHT11 ได้เลย เพราะโค้ด  
**Arduino DHT22** เชี่ยวนะเมื่อนักบิน

Accuracy humidity +2%RH(Max +5%RH); temperature +-0.2Celsius

Resolution or sensitivity humidity 0.1%RH; temperature 0.1Celsius

Repeatability humidity +-1%RH; temperature +-0.2Celsius

Humidity hysteresis +-0.3%RH

Long-term Stability +-0.5%RH/year

Sensing period Average: 2s

Interchangeability fully interchangeable

# Capacitive Soil Moisture Sensor - เซ็นเซอร์วัดความชื้นในดิน

---

- เซ็นเซอร์วัดความชื้นในดินแบบ Capacitive ปรับปรุงจากแบบเดิมที่เป็นแบบ resistor ใช้แผ่นทองแดงไปสัมผัสกับดินโดยตรง ซึ่งมีข้อเสียคือ หัววัดจะไม่ทน และเป็นสนิม ทำให้ค่าคลาดเคลื่อนได้ง่าย
- เซ็นเซอร์วัดความชื้นในดินแบบ Capacitive ใช้ทน ไม่ลอก ไม่เป็นสนิม
- ใช้หลักการตรวจสอบประจุของวัสดุ ถ้ามีค่าประจุมากแสดงว่าแสดงว่าชื้นมาก
- เนื่องจากใช้หลักการของการวัดประจุ ดังนั้นแผ่นเซ็นเซอร์จึงไม่ต้องสัมผัสกับดินหรือวัสดุโดยตรง จึงทนทานและแม่นยำกว่า
- ไม่ดูดวัดความชื้นในดินนี้ ให้ค่าออกเป็นค่า Analog ถ้าใช้กับ Arduino Uno ก็จะได้ค่าระหว่าง 0-614 สามารถนำค่านี้ไปเขียนตรวจสอบเงื่อนไขในโปรแกรมได้โดยตรง

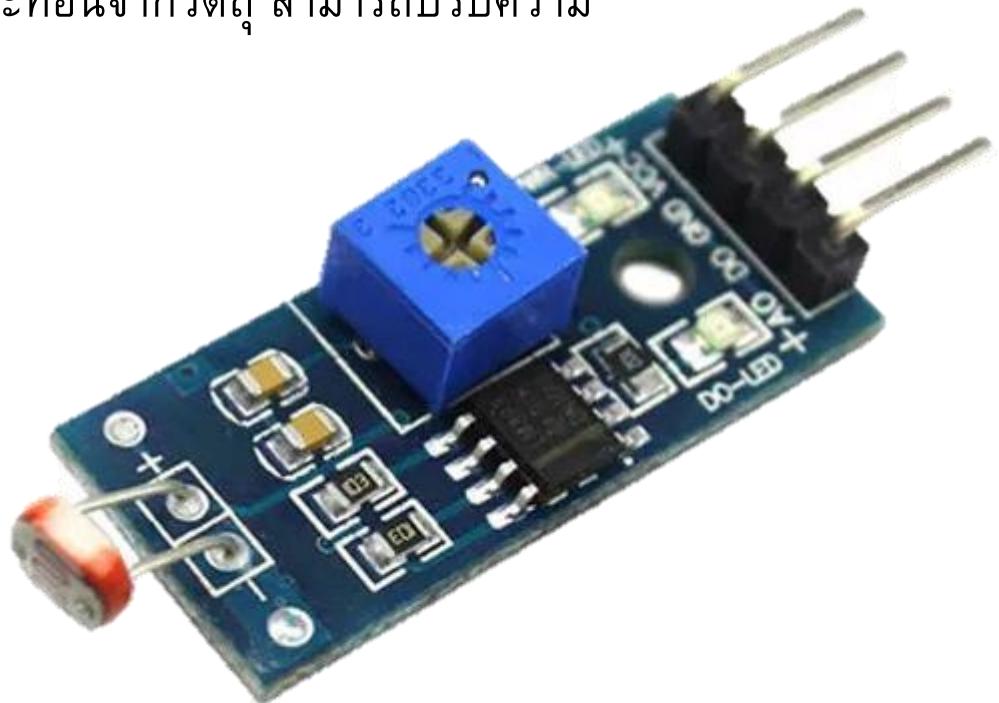


# LDR Photoresistor Sensor Module - โมดูลวัดแสง

---

- โมดูลวัดความเข้มของแสง Photoresistor LDR Light Sensor Module (LDR) แบบ 4-Pin
- ใช้วัดความเข้มของแสงได้ทั้งจากแหล่งกำเนิดแสงโดยตรง หรือจากการสะท้อนจากวัตถุ สามารถปรับความสว่างที่ทำให้ OUTPUT เป็น HIGH ได้

- **Features:**
- Can detect ambient brightness and light intensity
- Adjustable sensitivity (via blue digital potentiometer adjustment)
- Operating voltage 3.3V-5V
- Output Type
- Analog voltage output -A0
- Digital switching outputs (0 and 1) -D0
- With fixed bolt hole for easy installation
- Small board PCB size: 3cm \* 1.6cm
- Power indicator (red) and the digital switch output indicator (green)
- Using LM393 comparator chip, stable



# Selector switch mode – ปุ่มเลือกโหมด

---



- ชีล็อกเตอร์สวิตท์นี้จะทำงานโดยอาศัยการบิดด้วยมือเพื่อให้หน้า Contact ที่อยู่ภายในตัวชีล็อกเตอร์สวิตท์เปลี่ยนสถานะการทำงานในแต่ละตำแหน่ง ซึ่งลักษณะการบิดชีล็อกเตอร์มีอยู่ 2 แบบ
  - บิดค้าง เหมาะสำหรับงานที่ใช้การทำงานเป็นระยะเวลานานๆ เช่น บิดเพื่อทำการเปิดไฟแสดงสถานะทำงานของเครื่องจักร เป็นต้น
  - บิดเด้งกลับ (spring return) เหมาะสำหรับใช้ในการจูนตำแหน่งการทำงาน (จะแม่นกว่า) เช่น การทำงานของเครน เป็นการทำงานไม่ซับซ้อน จะใช้สิ่งการทำงานในระยะเวลาสั้นๆ
  - มีกุญแจ และแบบ เลือกตามลักษณะการทำงาน และ ที่ถูกออกแบบมา สามแบบนี้ นั้น มาเพื่อความปลอดภัย

# 5V Relay module 4 Channel

---

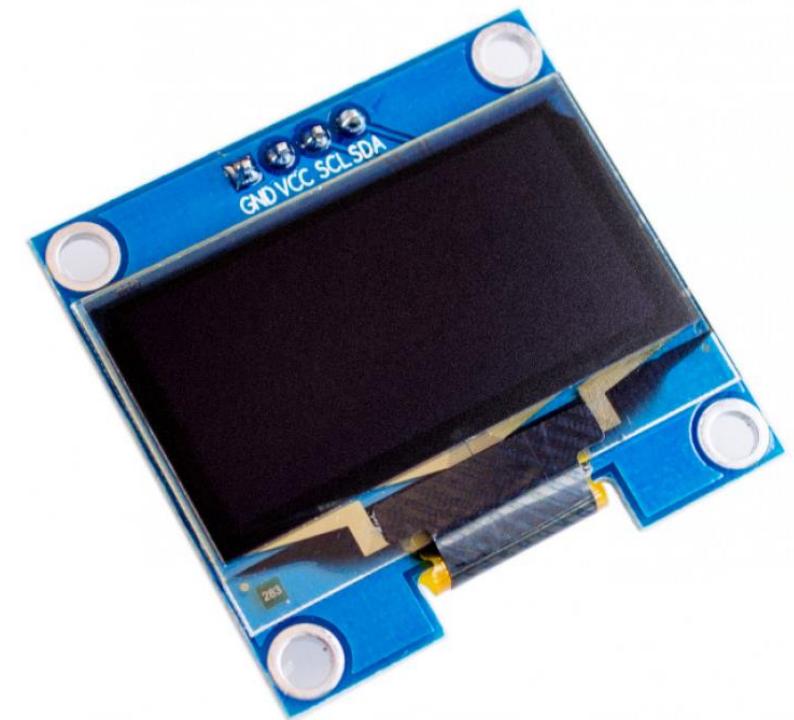


- รีเลย์ (Relays) คือ อุปกรณ์อิเล็กทรอนิกส์ที่ใช้กันอย่างกว้างขวางใน วงจรควบคุมอัตโนมัติ ใช้ในการเปิดและปิดอุปกรณ์ไฟฟ้าต่างๆ โดย ใช้อำนาจแม่เหล็กไฟฟ้าทำให้วงจรไฟฟ้าทำงาน รีเลย์มีส่วนประกอบ สำคัญคือ ขดลวด และส่วนของหน้าسمผสทำหน้าที่คล้ายสวิตซ์

# 0.96" OLED Display Module

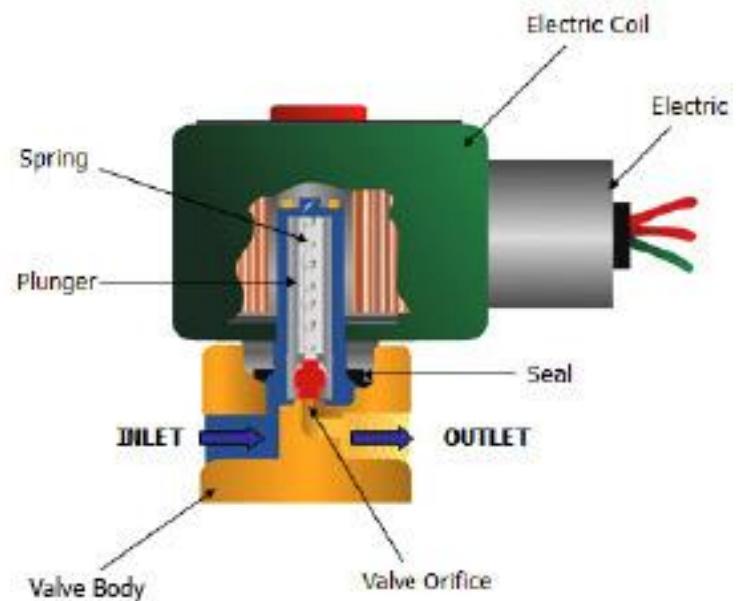
---

- จอแสดงผลแบบ OLED LCD สำหรับ Arduino หน้าจอ 128x64 ขนาด 0.96" เชื่อมต่อแบบ IIC สีขาว ใช้ไฟได้ทั้ง 3.3V หรือ 5V ให้จอสว่างแสดงผลมองเห็นได้อย่างชัดเจน และประยุกต์ไฟ สามารถดูภาพกราฟฟิกส์เป็นรูปต่าง ๆ หรือทำเป็นเมนูตามแบบที่ต้องการได้
- OLED (Organic Light Emitting Diodes) คือจอภาพที่มีลักษณะคล้ายแผ่นฟิล์ม ซึ่งมีส่วนประกอบเป็นสารอินทรีย์ที่สามารถเปล่งแสงเองได้เมื่อได้รับพลังงานไฟฟ้า เรียกว่ากระบวนการอิเล็กโทรลูมิเนเซนส์ (Electroluminescence) โดยที่ไม่ต้องพึ่งพาแสง Backlight และจะไม่มีการเปล่งแสงในบริเวณที่เป็นภาพสีดำ ส่งผลให้สีดำเน้นคำสนิท อีกทั้งยังช่วยลดลงงานอีกด้วย



# 12V Solenoid valve - โซลินอยด์วาล์ว

- โซลินอยด์วาล์ว (Solenoid Valve) คือ วาล์วที่ทำงานด้วยไฟฟ้าที่ใช้ควบคุมปริมาตรของ ของไหหลังไหหลผ่านห่อ โดยการเปิดหรือปิดที่รู orifices ของตัววาล์ว การทำงานของโซลินอยด์วาล์ว 2/2 โดยทั่วไปมีการควบคุมให้เปิดปิดได้ด้วย 3 ระบบ คือระบบเปิดปิดโดยตรง (Direct acting) ระบบเปิดปิดทางอ้อม (Pilot control) และระบบลูกผสม (Combined Operation)
- 1. ระบบเปิดปิดโดยตรง (Direct acting)
- โซลินอยด์วาล์ว 2 ทาง ที่มีระบบการทำงานแบบเปิดปิดโดยตรงนั้น มีทางเข้าหนึ่งทางและ ทางออกหนึ่งทาง ทุน (plunger) ซึ่งมีชิลอยู่ปลายด้านล่างทำหน้าที่เปิดและปิดรูทางผ่าน (orifice) ของของไหเมื่อจ่ายไฟฟ้าเข้าหรือหยุดจ่ายไฟที่ค้อยล์ของวาล์ว
- ข้อควรระวัง !!
- ในการใช้วาล์วที่ทำงานด้วยระบบนี้คือเมื่อมีการเพิ่มความดัน (pressure) ของของไหใน ระบบ จะทำให้ต้องใช้แรงมากขึ้นในการเปิดวาล์ว หากความดันของของไหสูงกว่าที่กำลัง ของคอยล์จะเปิดวาล์วได้ วาล์วนั้นก็จะไม่ทำงานถึงแม้ว่า จะมีการจ่ายไฟฟ้าแล้วก็ตาม



# Pilot Lamp LED

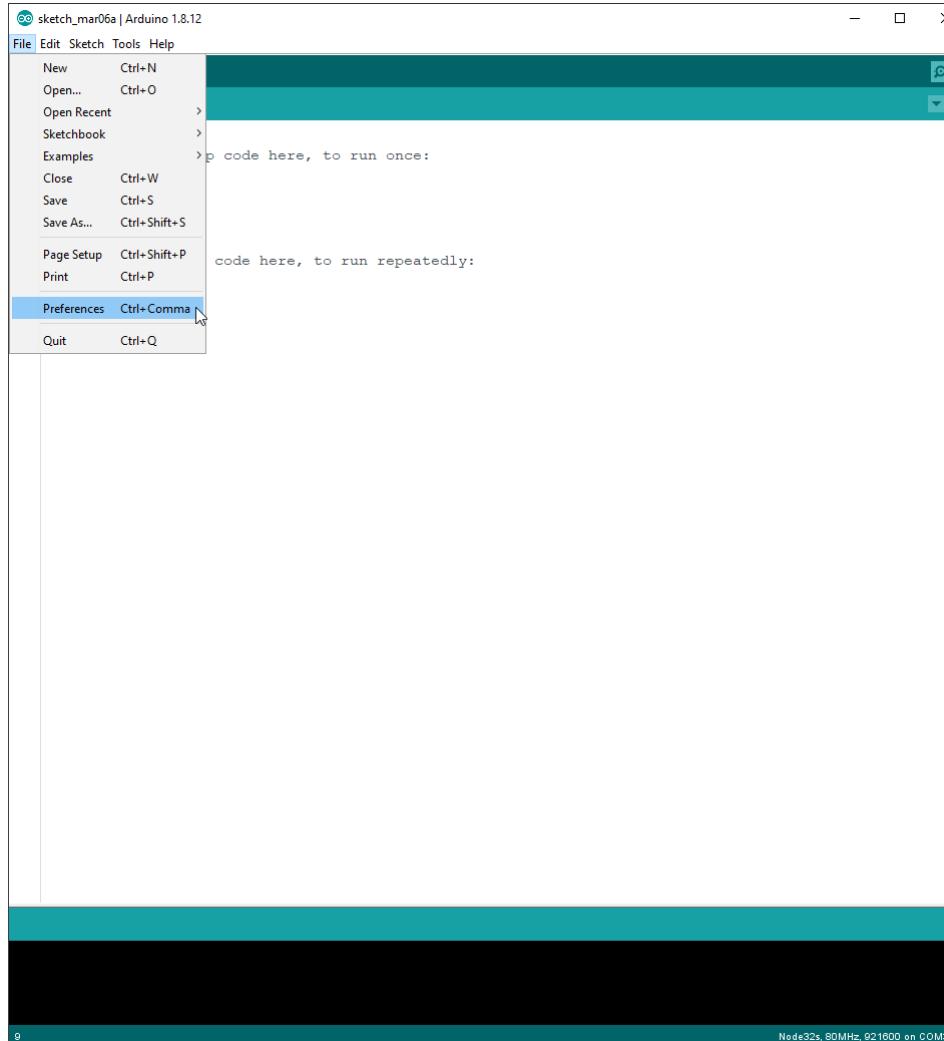
---



- หลอดไฟ LED Pilot Lamp ใช้แรงดันไฟฟ้าเลี้ยง 12V เหมาะสำหรับการนำไปใช้งานในการแสดงสถานะการทำงานต่างๆ มักจะเจอในตู้คอนโทรล

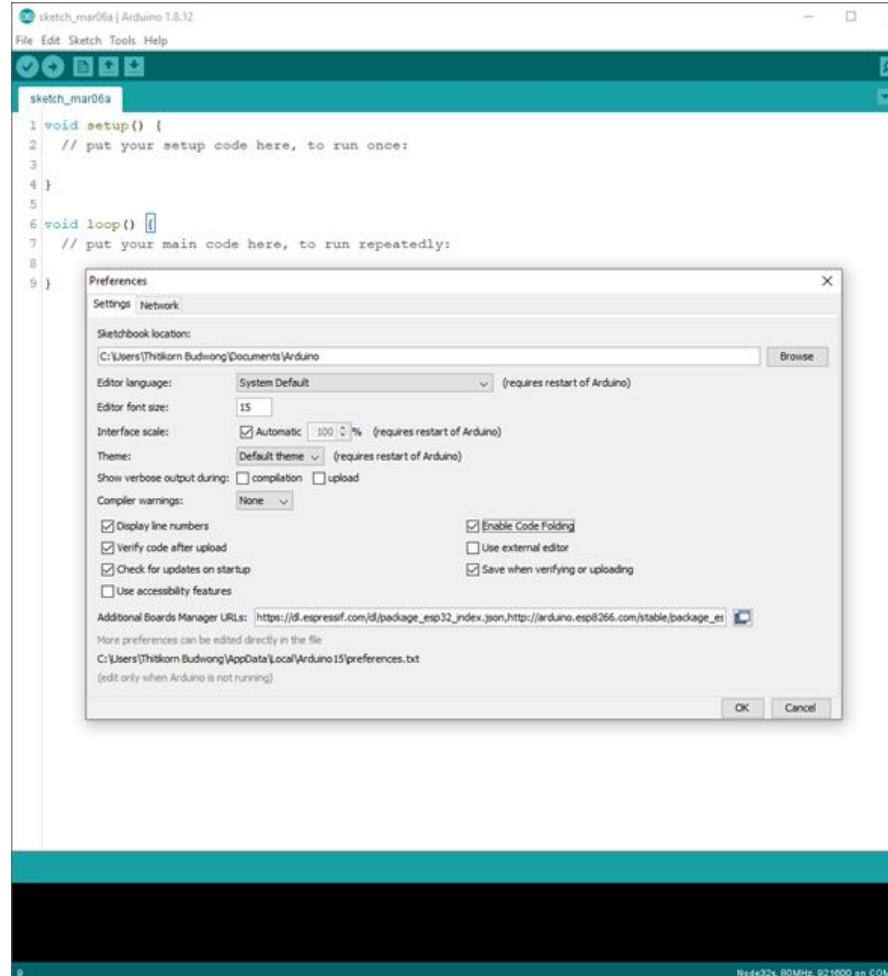
# การใช้งาน และการติดตั้ง Arduino core for ESP32

## ขั้นตอนที่ 1 เข้าเมนู File > Preferences



# การใช้งาน และการติดตั้ง Arduino core for ESP32

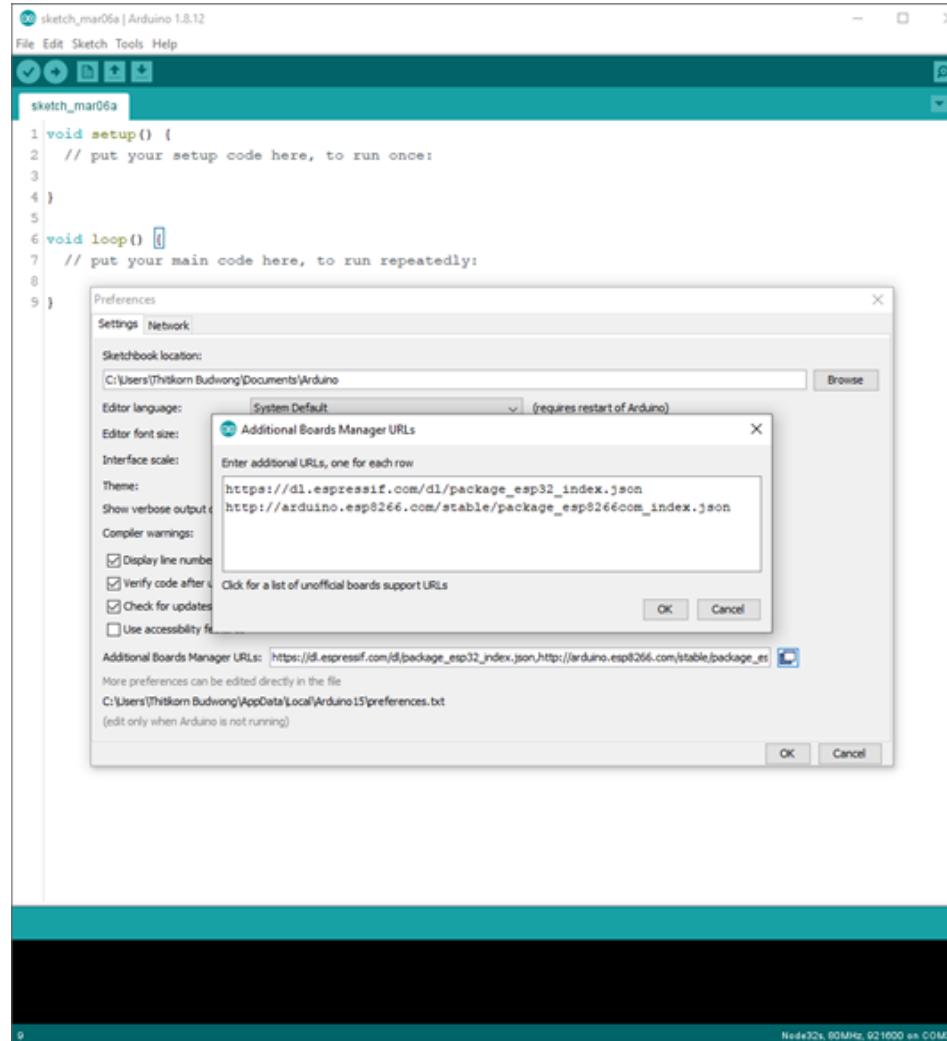
## ขั้นตอนที่ 2 กรอกข้อมูลลิงค์ของบอร์ด ดังนี้



[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) , [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

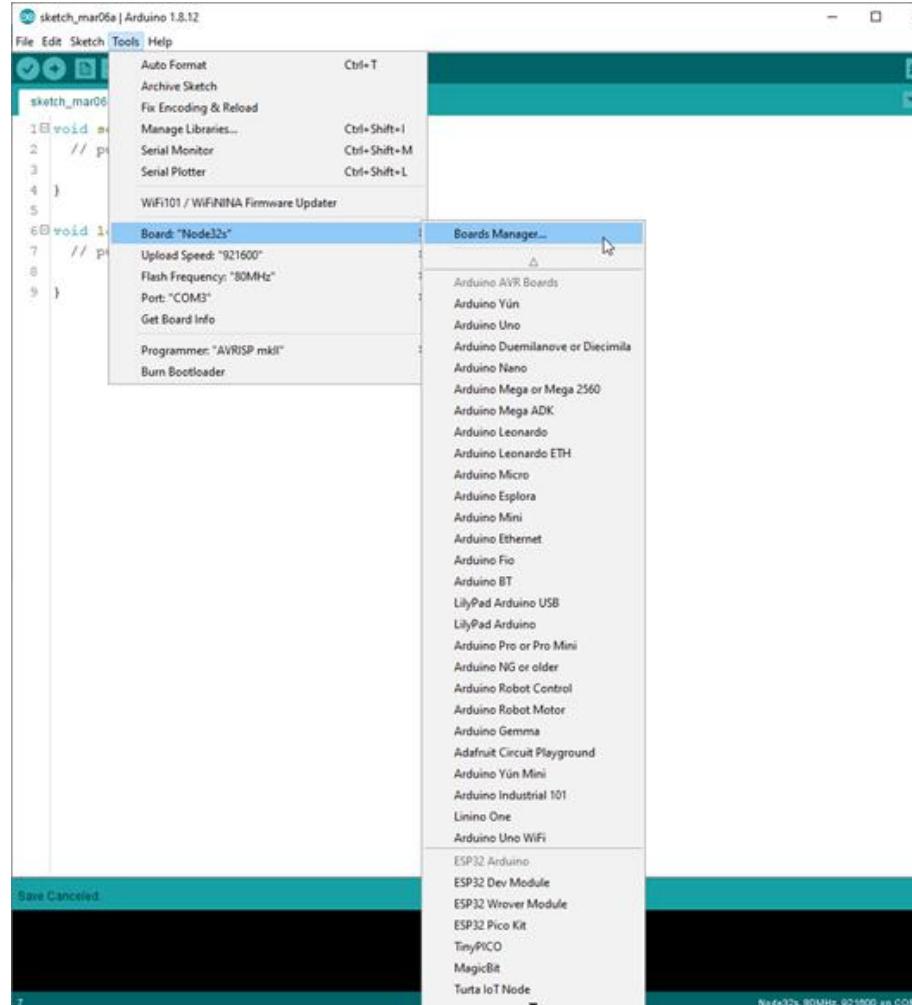
# การใช้งาน และการติดตั้ง Arduino core for ESP32

## ขั้นตอนที่ 3 เข้าไปที่ช่อง Additional Board Manager URLs



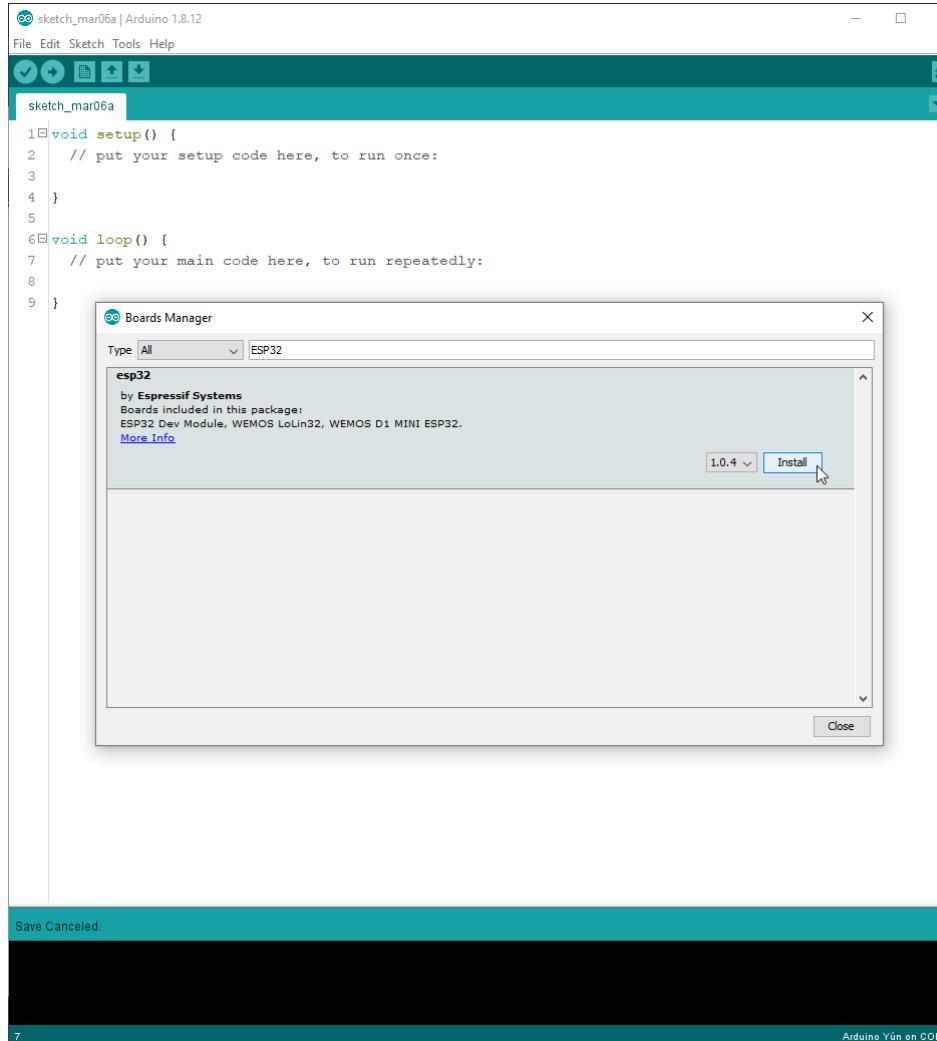
# การใช้งาน และการติดตั้ง Arduino core for ESP32

## ขั้นตอนที่ 4 เปิดเมนู Tools > Board > Board Manager...



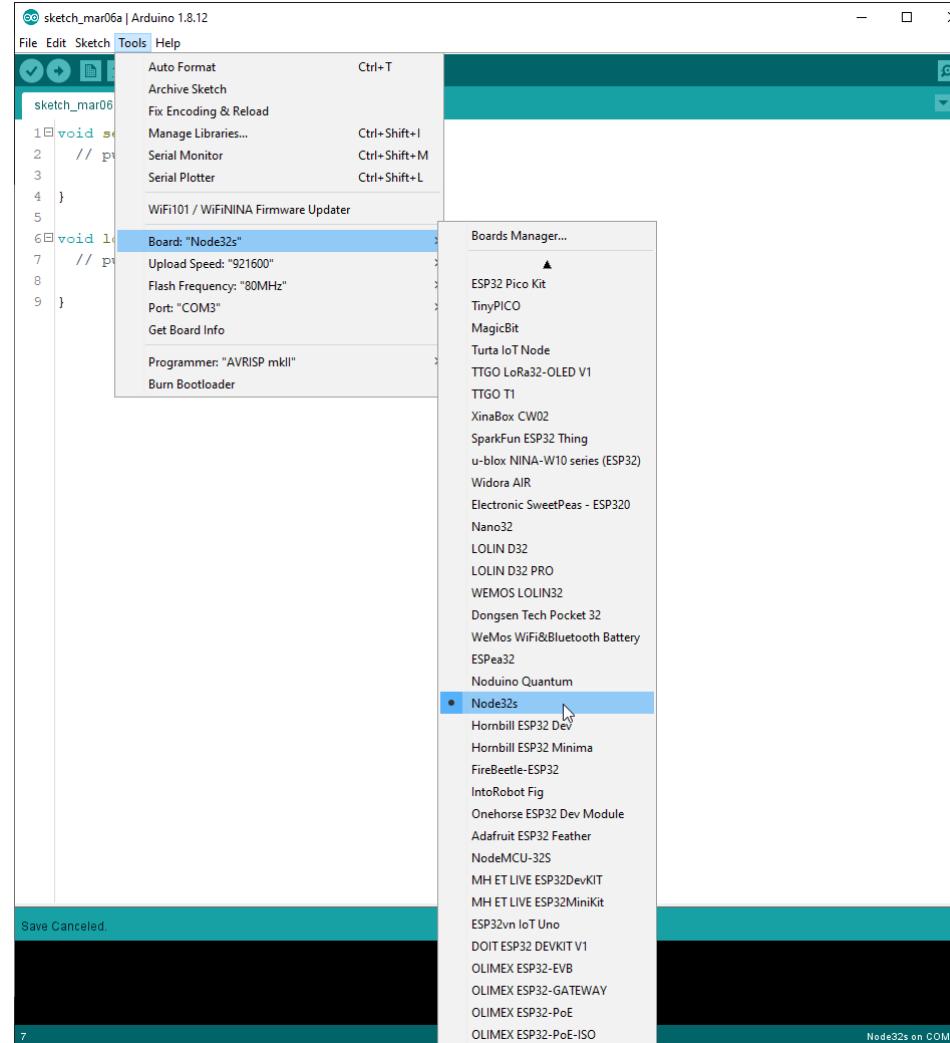
# การใช้งาน และการติดตั้ง Arduino core for ESP32

## ขั้นตอนที่ 5 ค้นหา ESP32 จากนั้น Install



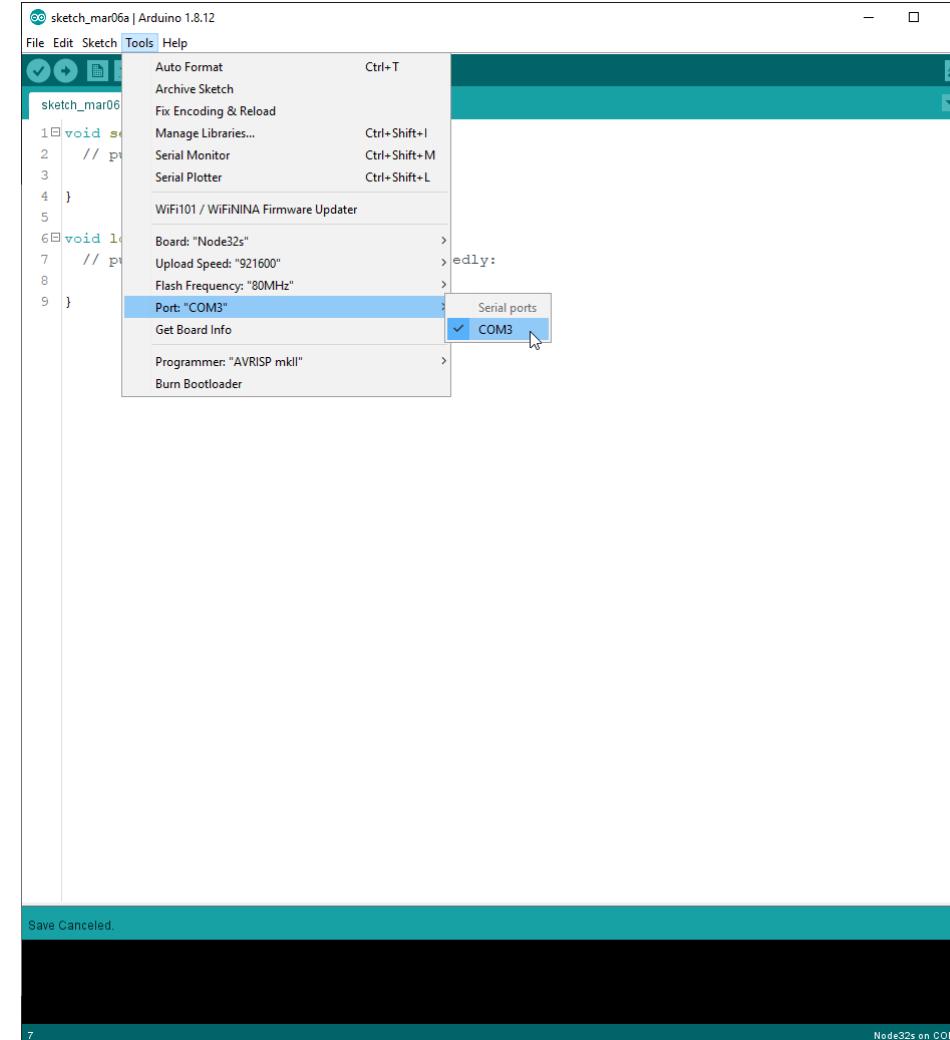
# ทดสอบอัปโหลดโคดลงบอร์ด Arduino ESP32

## ขั้นตอนที่ 1 เลือกบอร์ดไปที่เมนู Tools > Board: Node32s



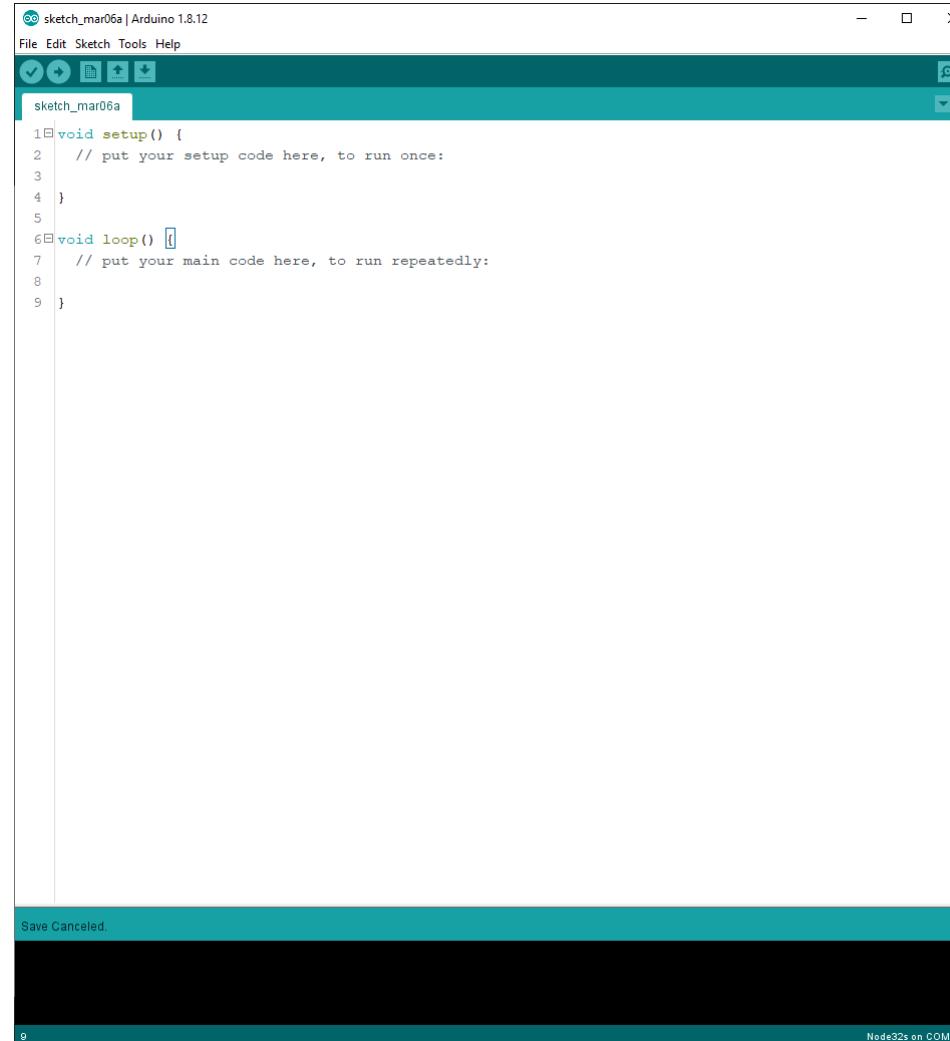
# ทดสอบอัปโหลดโคดลงบอร์ด Arduino ESP32

## ขั้นตอนที่ 2 เลือก Comport ไปที่เมนู Tool > Port:



# ทดสอบอัพโหลดโคดลงบอร์ด Arduino ESP32

ขั้นตอนที่ 3 เปิดโคดตัวอย่างที่เมนู File > Examples > ESP32 > ChipID > GetChipID



The screenshot shows the Arduino IDE interface with the title bar "sketch\_mar06a | Arduino 1.8.12". The main code editor window displays the following C++ code:

```
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8 }
```

At the bottom of the code editor, a message "Save Canceled." is displayed. In the bottom right corner of the IDE, the text "Node32s on COM3" is visible.

# ทดสอบอัปโหลดโคดลงบอร์ด Arduino ESP32

## ขั้นตอนที่ 4 กดปุ่ม Upload

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** GetChipID | Arduino 1.8.12
- Menu Bar:** File Edit Sketch Tools Help
- Sketch Editor:** The code for 'GetChipID' is displayed:

```
1 uint64_t chipid;
2
3 void setup() {
4     Serial.begin(115200);
5 }
6
7 void loop() {
8     chipid=ESP.getEfuseMac(); //The chip ID is essentially its MAC address(length: 6 bytes).
9     Serial.printf("ESP32 Chip ID = %04X", (uint16_t)(chipid>>32)); //print High 2 bytes
10    Serial.printf("%08X\n", (uint32_t)chipid); //print Low 4bytes.
11
12    delay(3000);
13
14 }
```
- Serial Monitor:** Shows the output of the uploaded sketch:

```
Done uploading.
Leaving...
Hard resetting via RTS pin...
```
- Status Bar:** Node32s on COM3

# ทดสอบอัพโหลดโคดลงบอร์ด Arduino ESP32

## ขั้นตอนที่ 5 กดปุ่ม Serial Monitor เพื่อดูผลลัพธ์

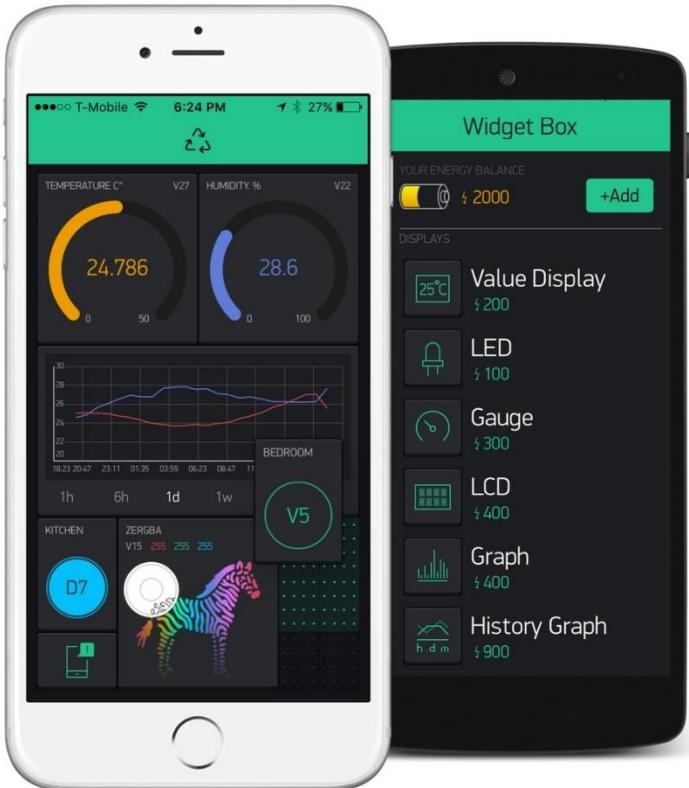
The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** GetChipID | Arduino 1.8.12
- File Menu:** File Edit Sketch Tools Help
- Sketch Area:** Contains the code for "GetChipID":

```
1 uint64_t chipid;
2
3 void setup() {
4     Serial.begin(115200);
5 }
6
7 void loop() {
8     chipid=ESP.getEfuseMac(); //The chip ID is essentially its MAC address(length: 6 bytes).
9     Serial.printf("ESP32 Chip ID = %04X", (uint16_t)(chipid>>32)); //print High 2 bytes
10 }
```
- Serial Monitor:** Set to COM3, Baud rate 115200, Autocroll checked. Displays the following output:

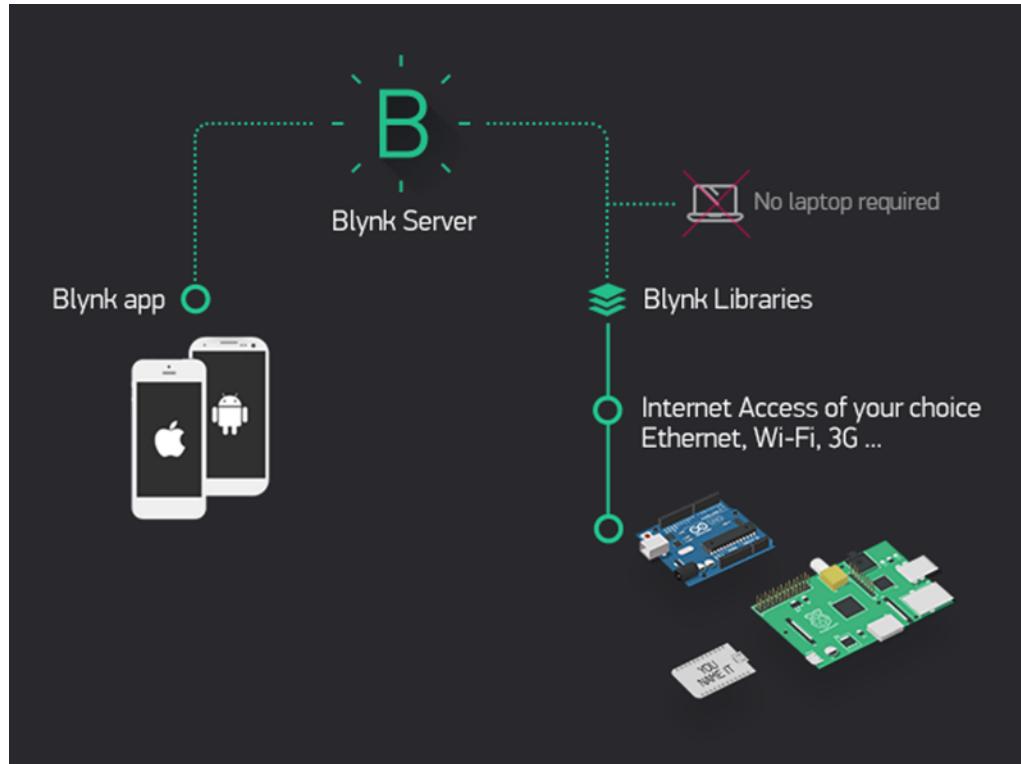
```
16:56:27.896 -> ESP32 Chip ID = 08E37AA4AE30
16:56:30.897 -> ESP32 Chip ID = 08E37AA4AE30
16:56:33.887 -> ESP32 Chip ID = 08E37AA4AE30
16:56:36.902 -> ESP32 Chip ID = 08E37AA4AE30
```
- Bottom Status Bar:** Leaving... Hard resetting via RTS pin...
- Page Number:** 10
- Page Footer:** Node32s on COM3

# การใช้งาน Application Blynk



- Blynk คือ Application สำหรับทำงาน IOT มีความน่าสนใจคือการเขียนโปรแกรมที่ง่าย ไม่ต้องเขียน App เองสามารถใช้งานได้อย่าง Real time สามารถเชื่อมต่อ Device ต่าง ๆ เช้ากับ Internet ได้อย่างง่ายดาย ไม่ว่าจะเป็น Arduino, Esp8266, Esp32, Nodemcu, Raspberry Pi นำมาแสดงบน Application ได้อย่างง่ายดาย และที่สำคัญ Application Blynk ยังฟรี และรองรับในระบบ IOS และ Android

# โครงสร้างการเชื่อมต่อ Blynk Server



- Blynk เป็น platform ที่ช่วยให้ผู้ใช้งานสามารถเชื่อมต่อบอร์ดชนิดต่าง ๆ ที่เชื่อมต่ออินเตอร์เน็ตได้แล้วสามารถควบคุมการทำงานได้ง่ายขึ้น บริการของ Blynk มี 3 ส่วน
  - Mobile App สำหรับควบคุมหรือดูข้อมูล
  - Cloud Service สำหรับเก็บข้อมูลจากอุปกรณ์
  - Library สำหรับเขียนโค้ดเชื่อมต่อกับ Cloud Service

# การติดตั้ง Blynk (legacy)

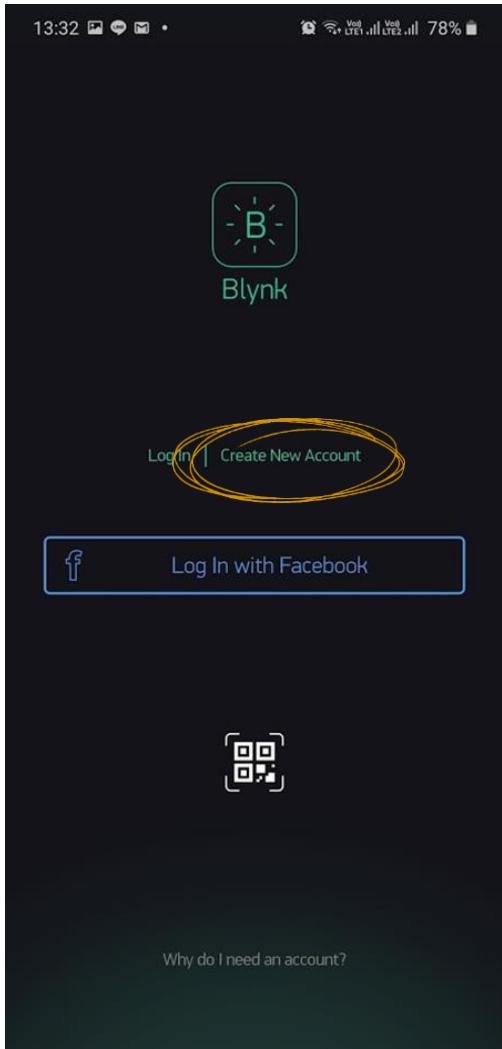


- เริ่มด้วยการค้นหาชื่อ Application ให้พิมพ์คำว่า “blynk” ใน App Store หรือ google play และทำการติดตั้ง



# สร้าง user account

คลิกที่ Create New Account

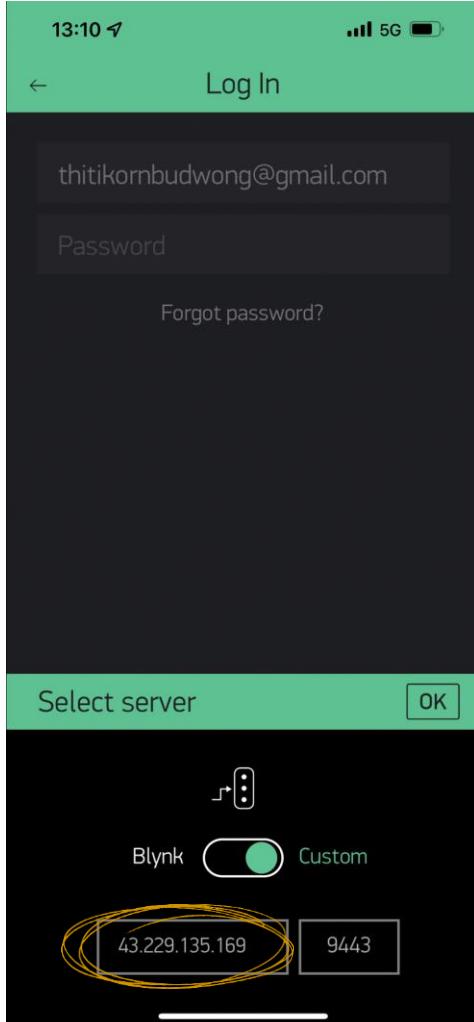


ตั้งค่า Server Settings โดยคลิกที่ Icon ตามรูป

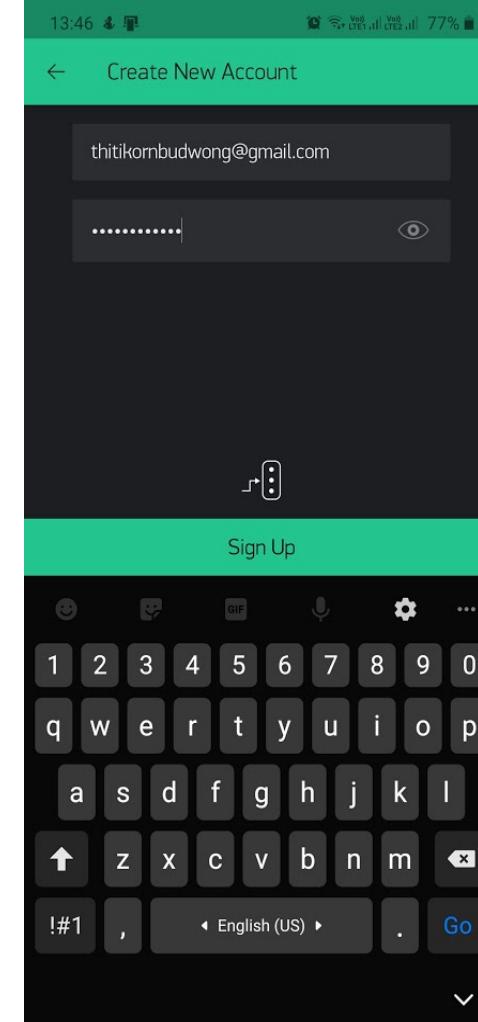


# สร้าง user account

เลือก CUSTOM Blynk Server  
เป็น 43.229.135.169 กด OK

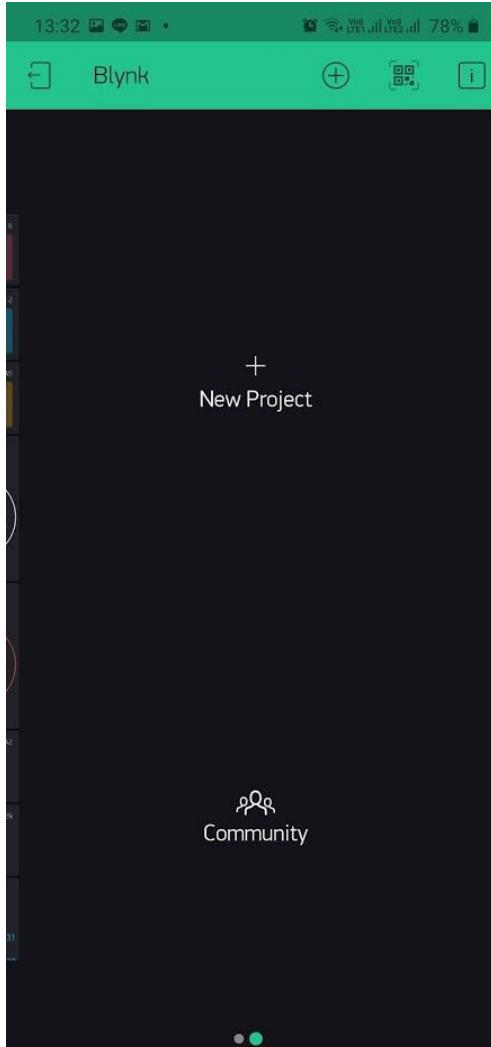


ป้อน Email และตั้ง รหัสใช้งาน คลิก Sign Up หรือ  
ย้อนกลับไปหน้าแรก แล้ว Log In with Facebook

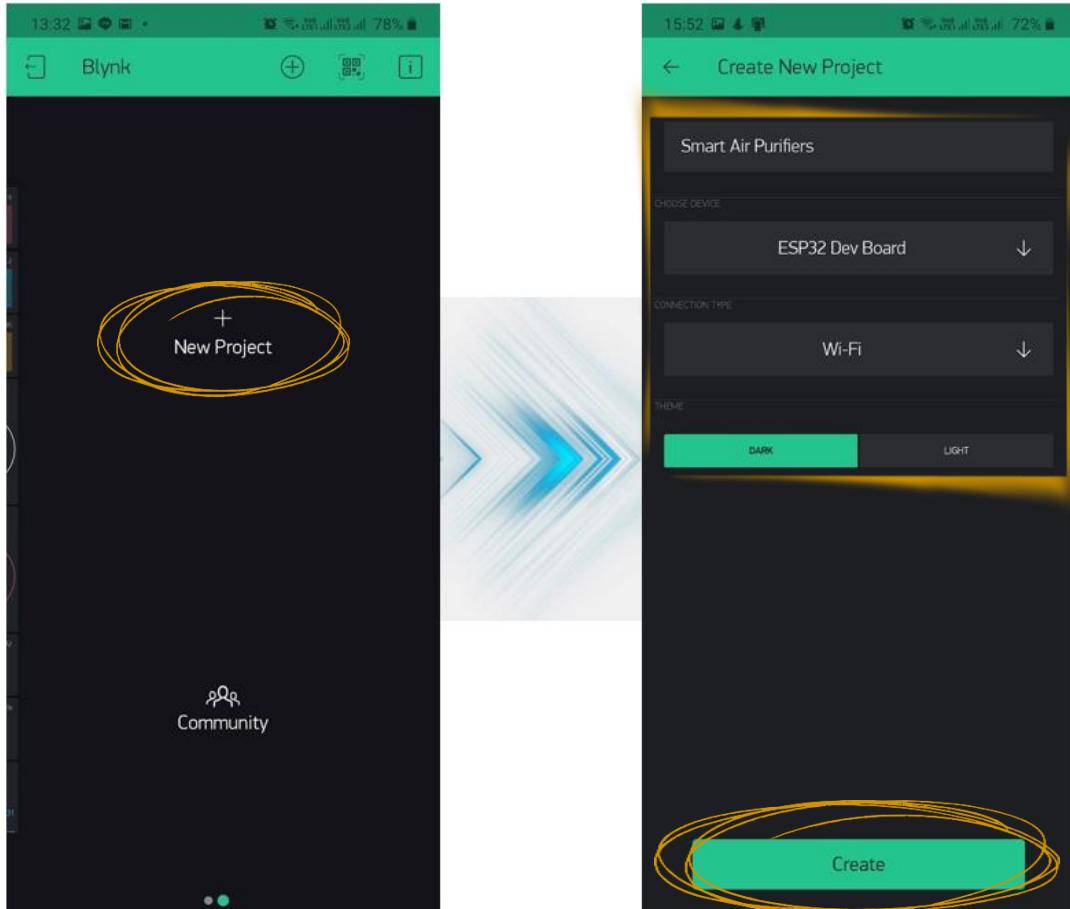


# สร้าง user account

หากเปิด Server สำเร็จ ก็จะเข้าหน้าตามนี้เลยครับ เป็นอันเสร็จสิ้น



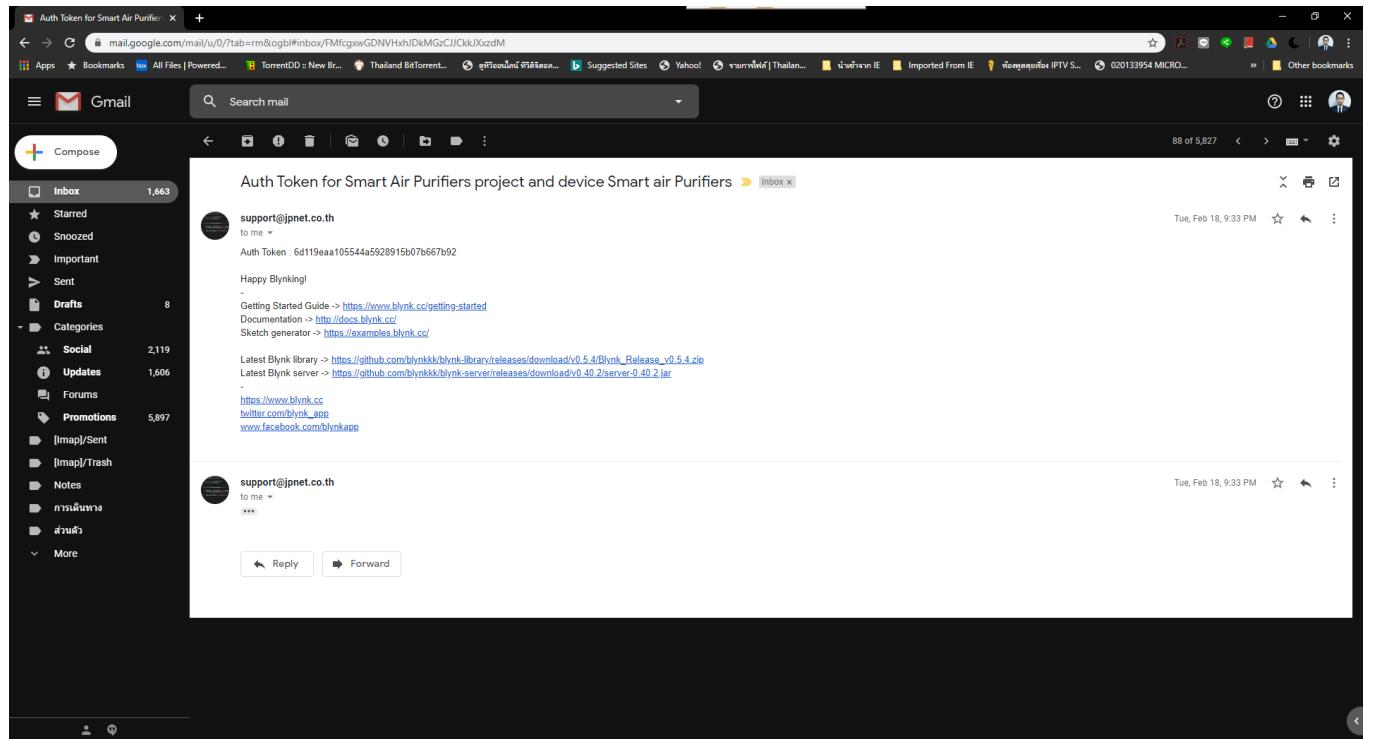
# การสร้างโปรเจคใหม่



- ขั้นตอนต่อจะเป็นการสร้างโปรเจคใหม่ ให้คลิกที่ “New Project” และใส่ชื่อโครงการให้เรียบร้อย จากตัวอย่างจะตั้งชื่อโครงการเป็น “Smart Air Purifiers” และเลือกประเภทของบอร์ดพัฒนาเป็น “ESP32 Dev Board” เลือก Connection type เป็น Wi-fi THEME ก็คือสีพื้นของ APP สามารถเลือกเป็น “LIGHT” หรือ “DARK” จากนั้นคลิกที่ปุ่ม “Create” จะเป็นการสร้างโปรเจค

# การสร้างโปรเจคใหม่

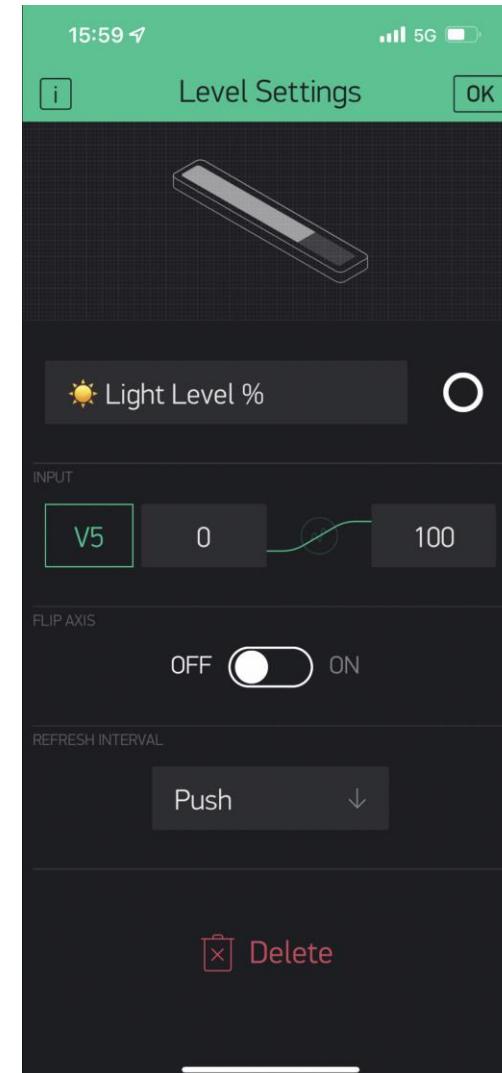
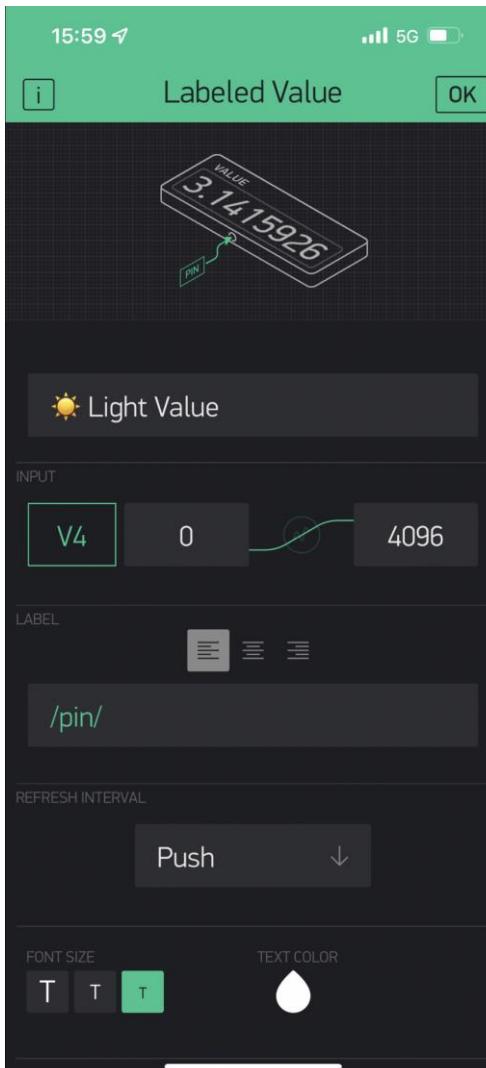
- หลังจากที่สร้างโปรเจคเสร็จเรียบร้อย เราจะได้รับรหัส TOKEN ที่ Blynk Server ส่งมาให้ ตามรูปตัวอย่าง ข้างล่าง รหัสนี้ก็คือตัวแทนของบอร์ดพัฒนานั้นเอง Blynk Server จะรักับบอร์ดเราได้และรู้ว่าตัวไหนอยู่ที่ไหนก็อาศัยรหัส TOKEN เป็นตัวอ้างอิง





# เพิ่ม Widget Blynk Application ในโครงการ

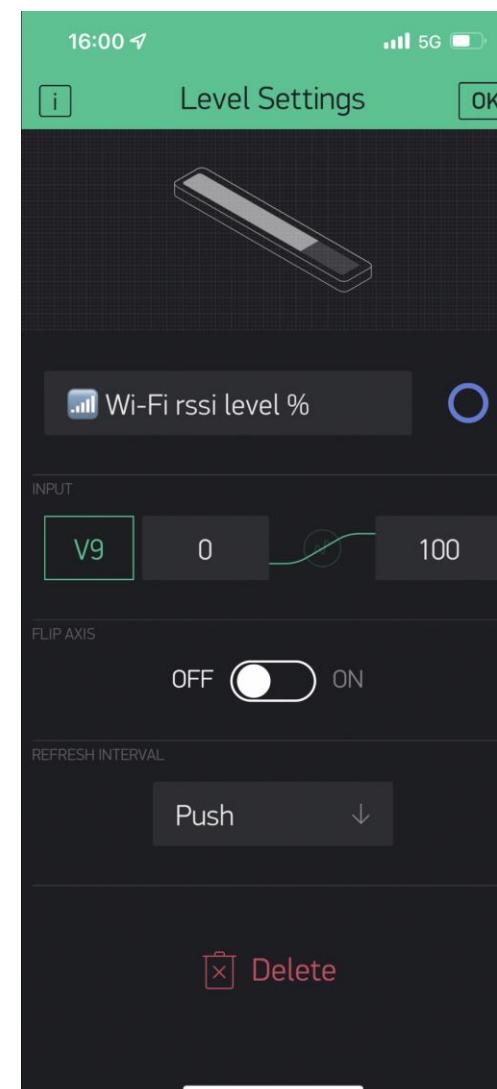
- ส่วนที่ 1 จะประกอบด้วย LIGHT VALUE และแถบบาร์เปอร์เซ็นต์ของ LIGHT LEVEL จะเชื่อมต่อกับขา V4 และ V5 ตามลำดับ





# เพิ่ม Widget Blynk Application ในโครงการ

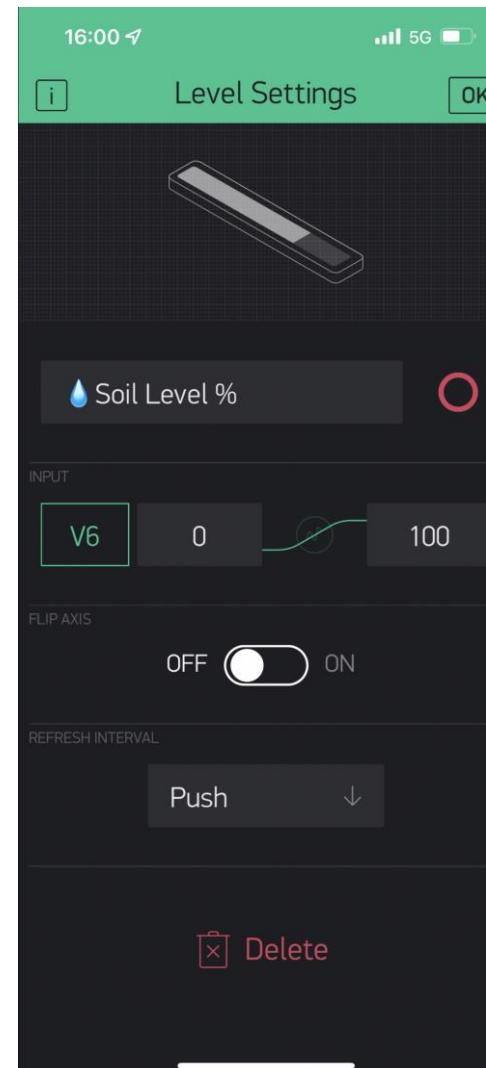
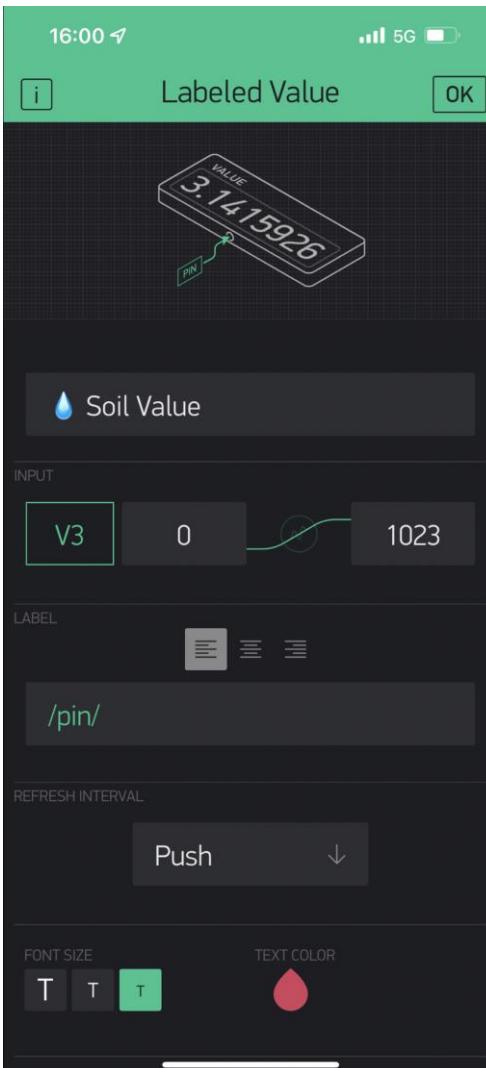
- ส่วนที่ 2 จะประกอบด้วย WI-FI RSSI และແນບນາງເປົ້ອງເຮັດວຽກ  
WI-FI RSSI LEVEL จะເຊື່ອມຕ່ອກັນຂາ V10 ແລະ V9 ຕາມລຳດັບ





# เพิ่ม Widget Blynk Application ในโครงการ

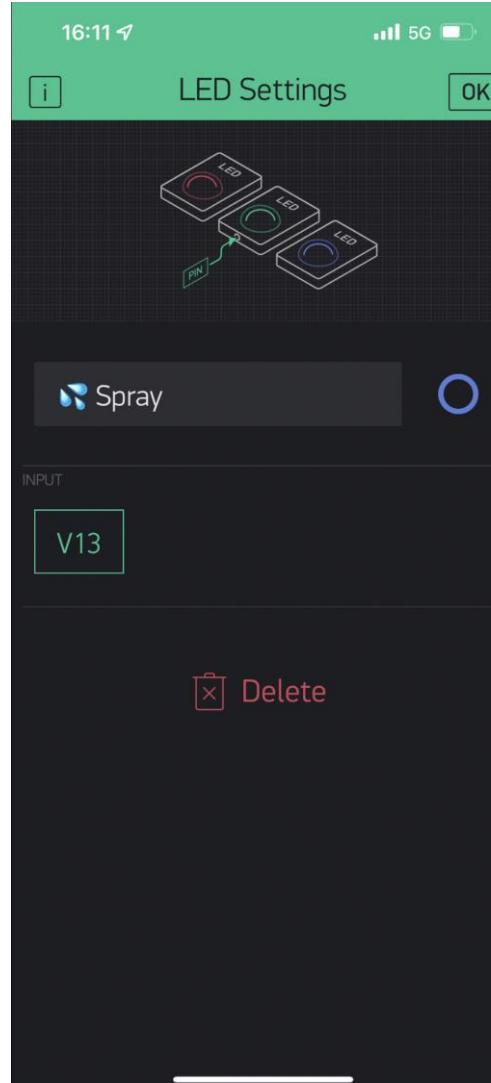
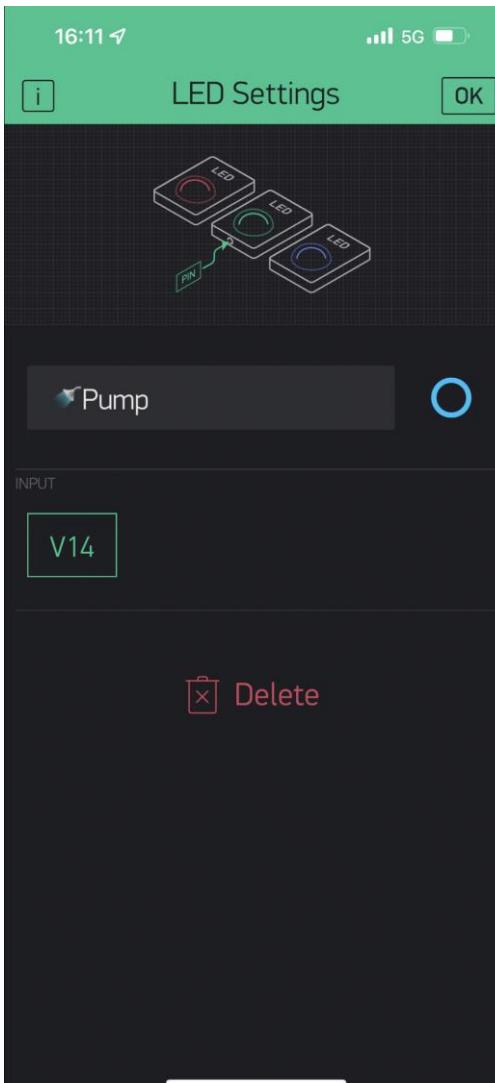
- ส่วนที่ 3 จะประกอบด้วย SOIL VALUE และแบบบาร์เปอร์เซ็นต์ของ SOIL VALUE LEVEL จะเชื่อมต่อกับขา V3 และ V6 ตามลำดับ





# เพิ่ม Widget Blynk Application ในโครงการ

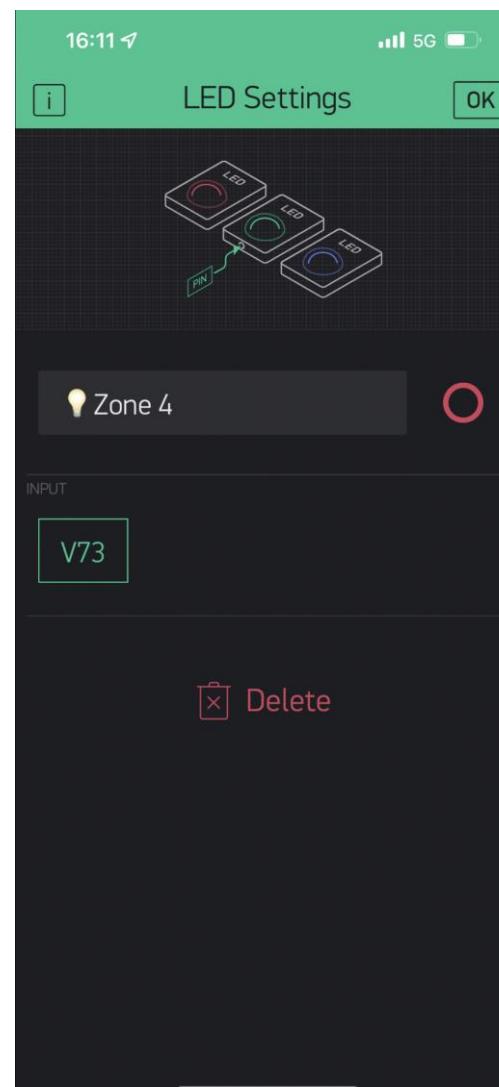
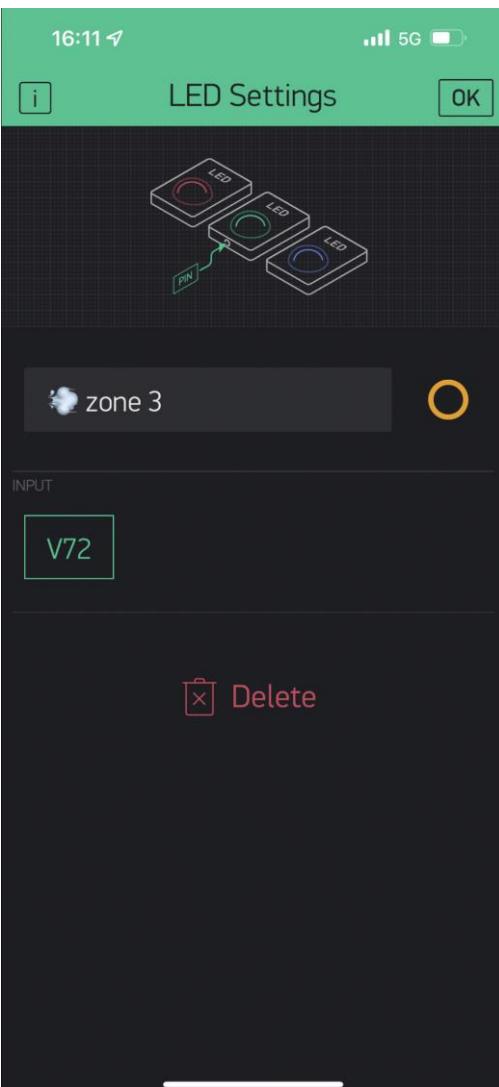
- ส่วนที่ 4 จะประกอบด้วย PUMP LED, SPRAY LED, ZONE 3 LED และ ZONE 4 LED จะเชื่อมต่อกับขา V14, V13, V72 และ V73 ตามลำดับ





# เพิ่ม Widget Blynk Application ในโครงการ

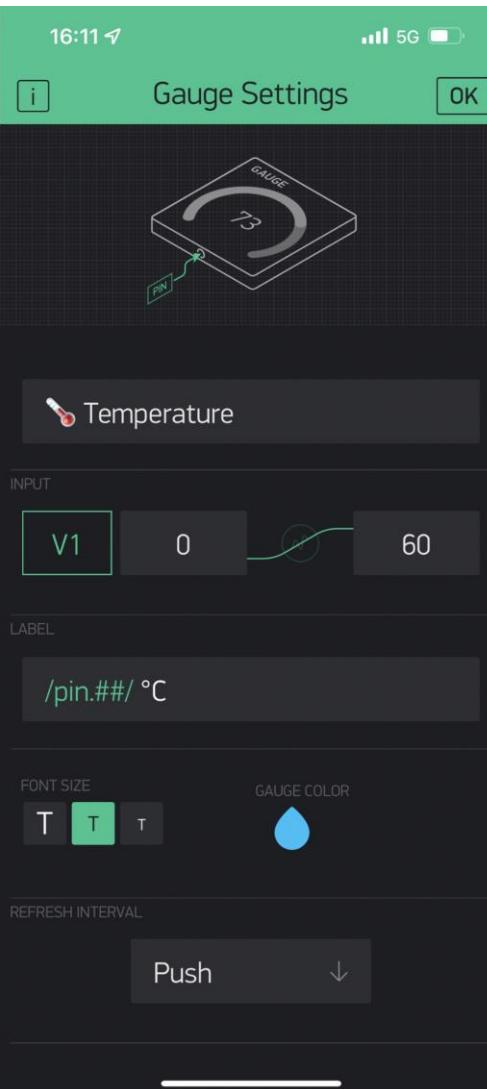
- ส่วนที่ 4 จะประกอบด้วย PUMP LED, SPRAY LED, ZONE 3 LED และ ZONE 4 LED จะเชื่อมตอกับขา V14, V13, V72 และ V73 ตามลำดับ

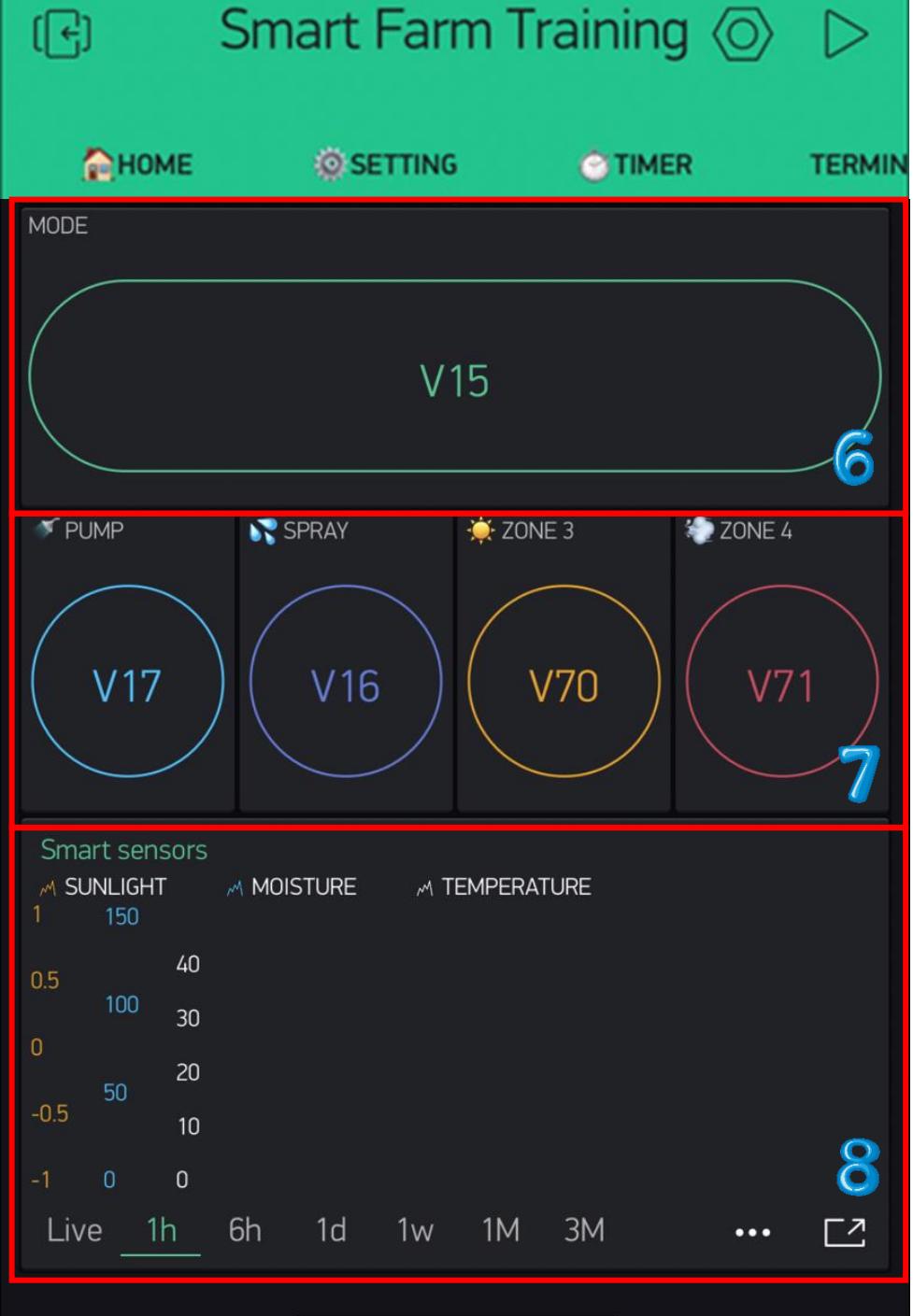




# เพิ่ม Widget Blynk Application ในโครงการ

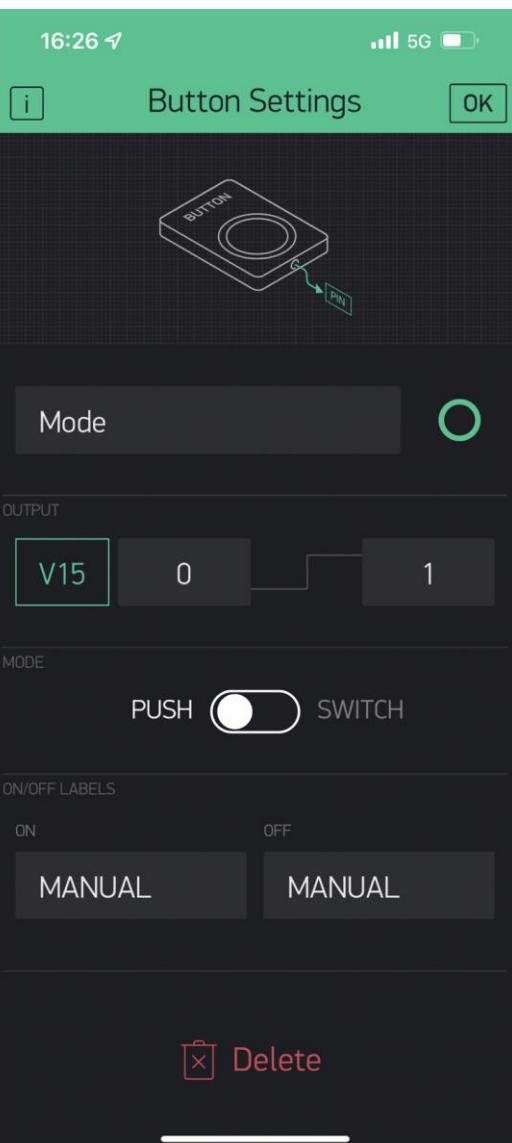
- ส่วนที่ 5 จะประกอบด้วย TEMPERATURE GAUGE และ HUMIDITY GAUGE จะเชื่อมต่อกับขา V1 และ V2 ตามลำดับ





# เพิ่ม Widget Blynk Application ในโครงการ

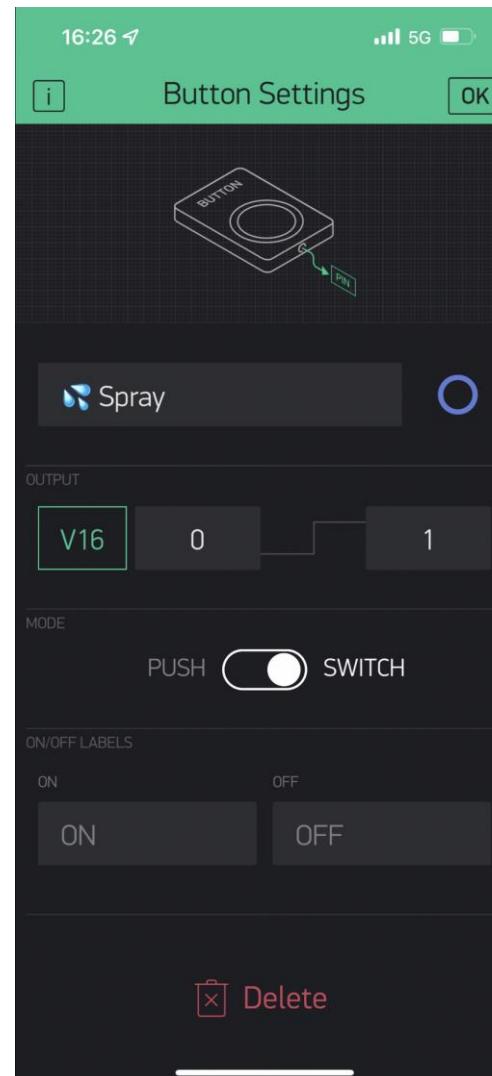
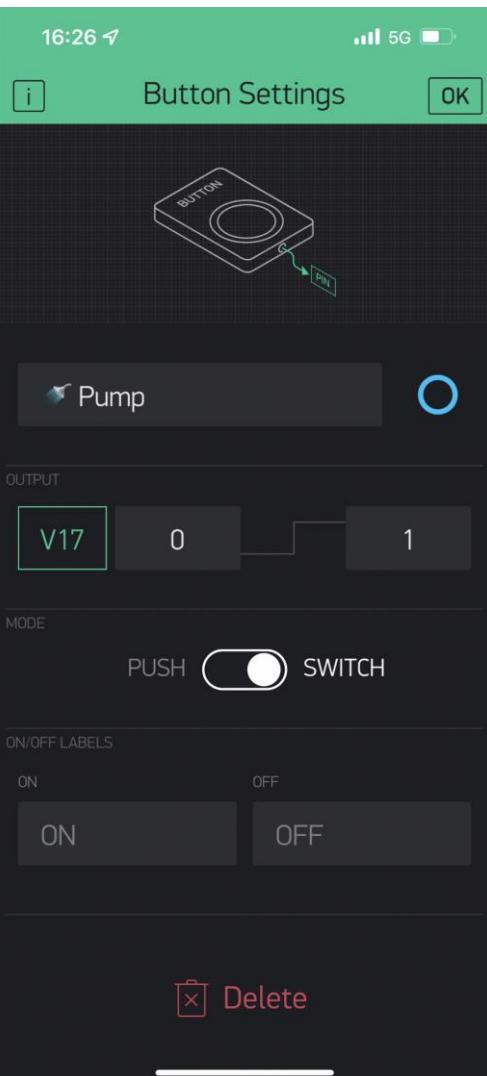
- ส่วนที่ 6 จะประกอบด้วย MODE BOTTON จะเชื่อมต่อกับขา V15





# เพิ่ม Widget Blynk Application ในโครงการ

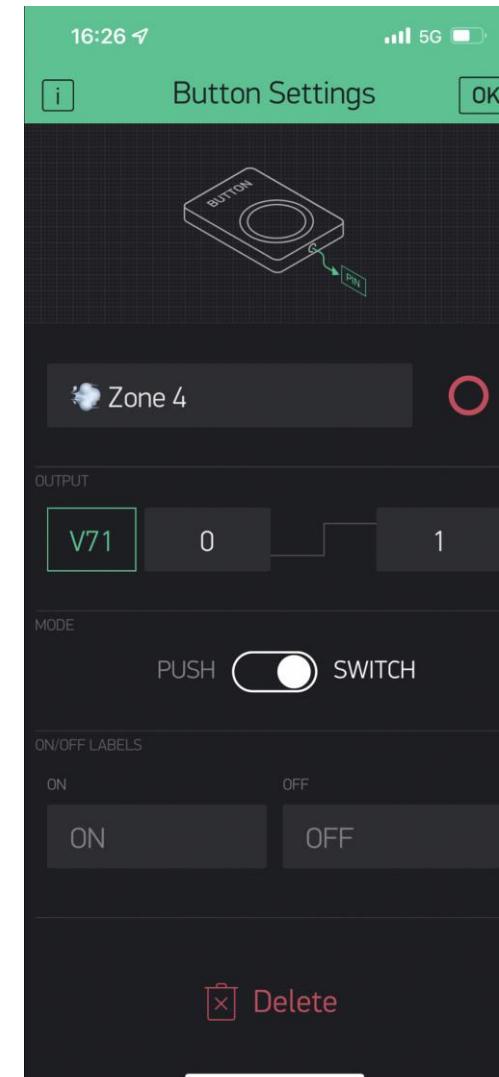
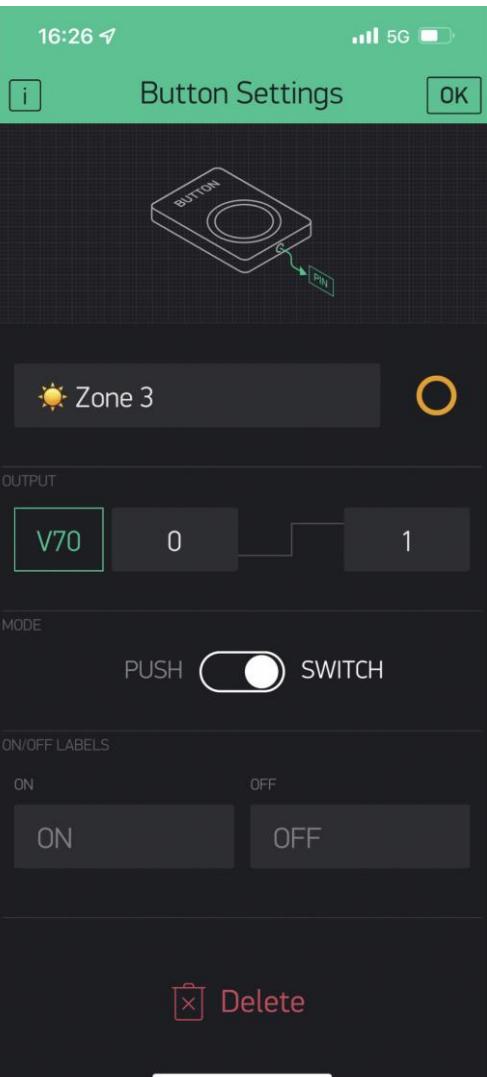
- ส่วนที่ 7 จะประกอบด้วย PUMP BOTTON และ SPRAY BOTTON จะเชื่อมต่อกันขา V17 และ V16 ตามลำดับ





# เพิ่ม Widget Blynk Application ในโครงการ

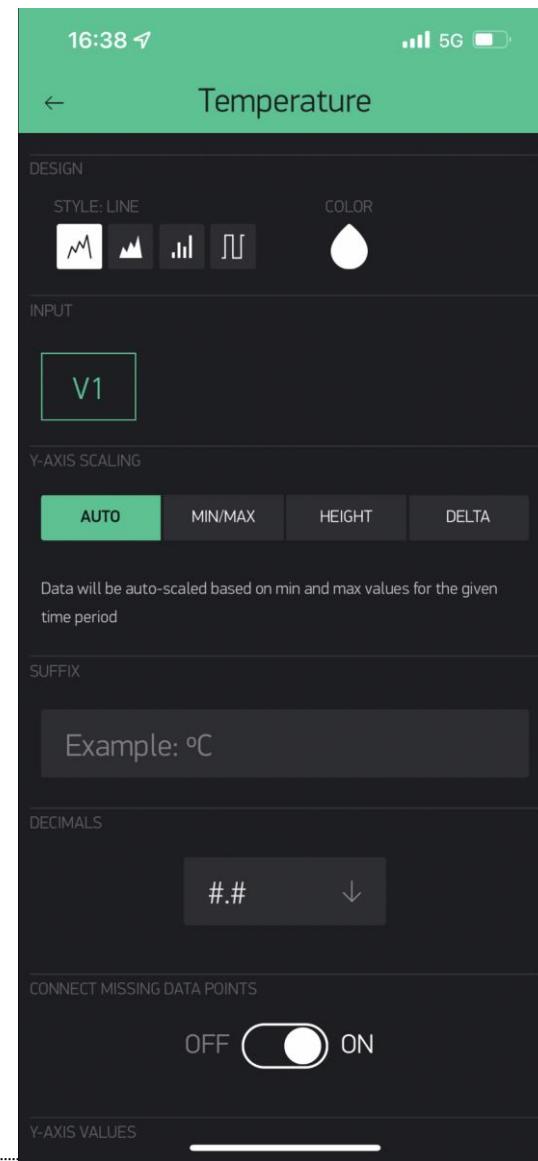
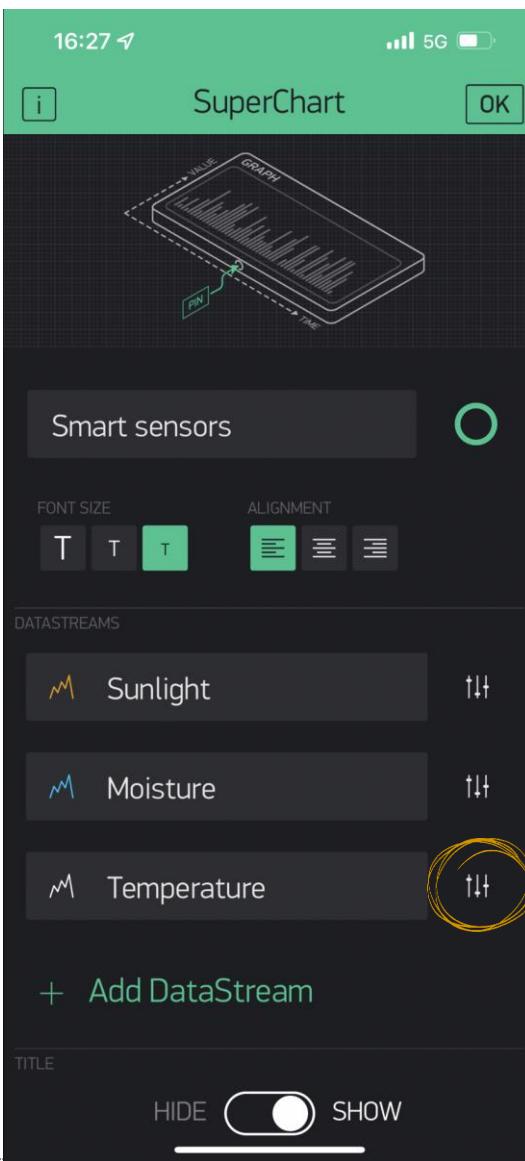
- ส่วนที่ 7 จะประกอบด้วย ZONE 3 BOTTON และ ZONE 4 BOTTON จะเชื่อมต่อกันขา V70 และ V71 ตามลำดับ





# เพิ่ม Widget Blynk Application ในโครงการ

- ส่วนที่ 8 จะประกอบด้วย SUPER CHART

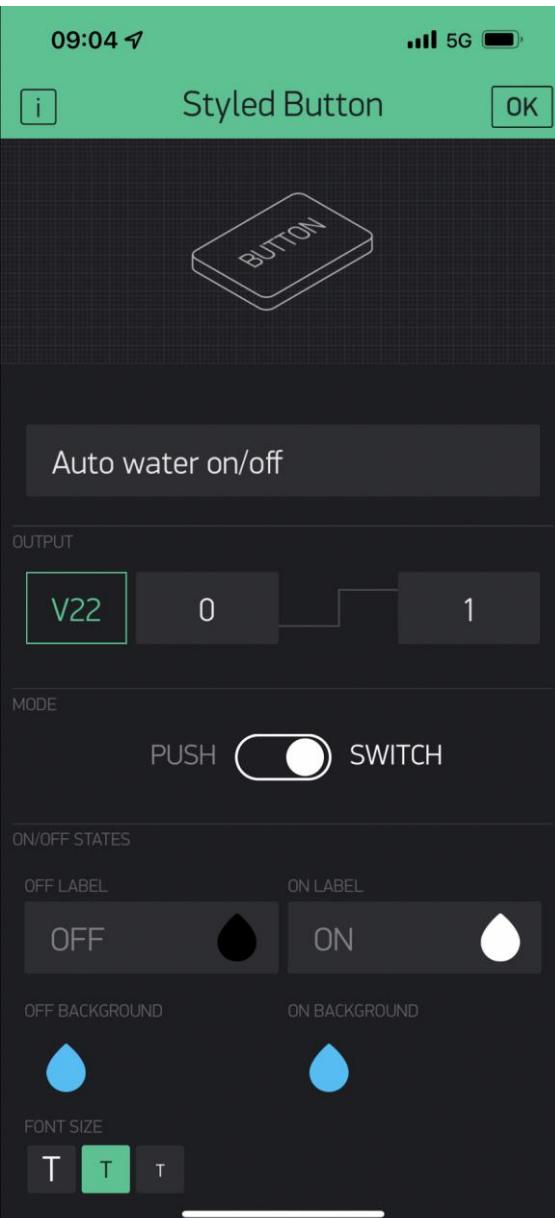
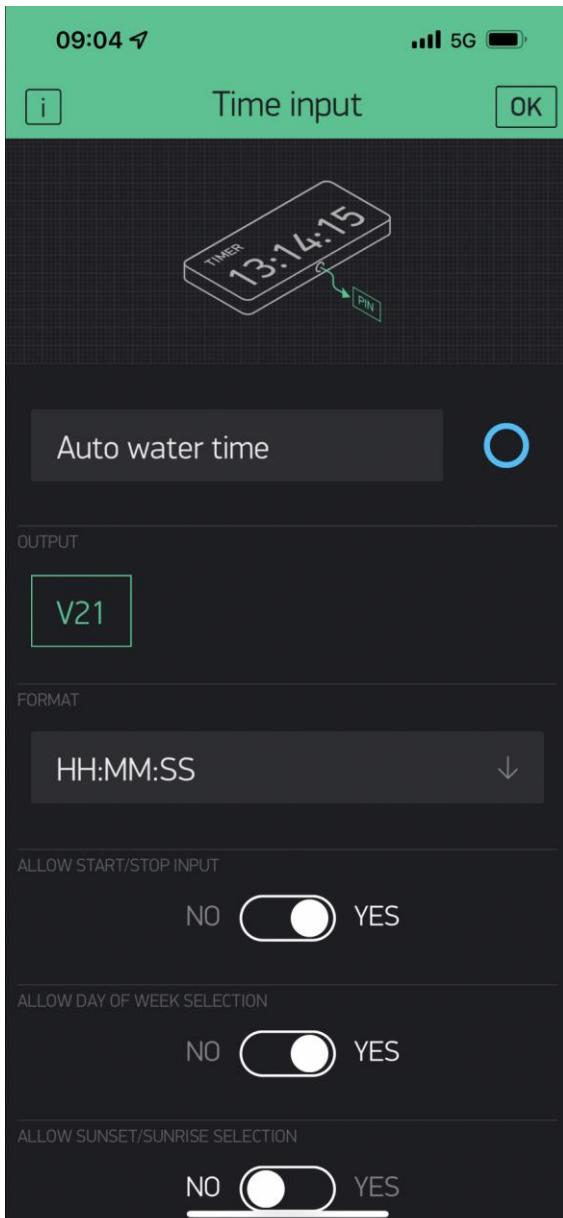


13:23

5G



• ส่วนที่ 9 จะประกอบด้วย Time input และ Styled Button

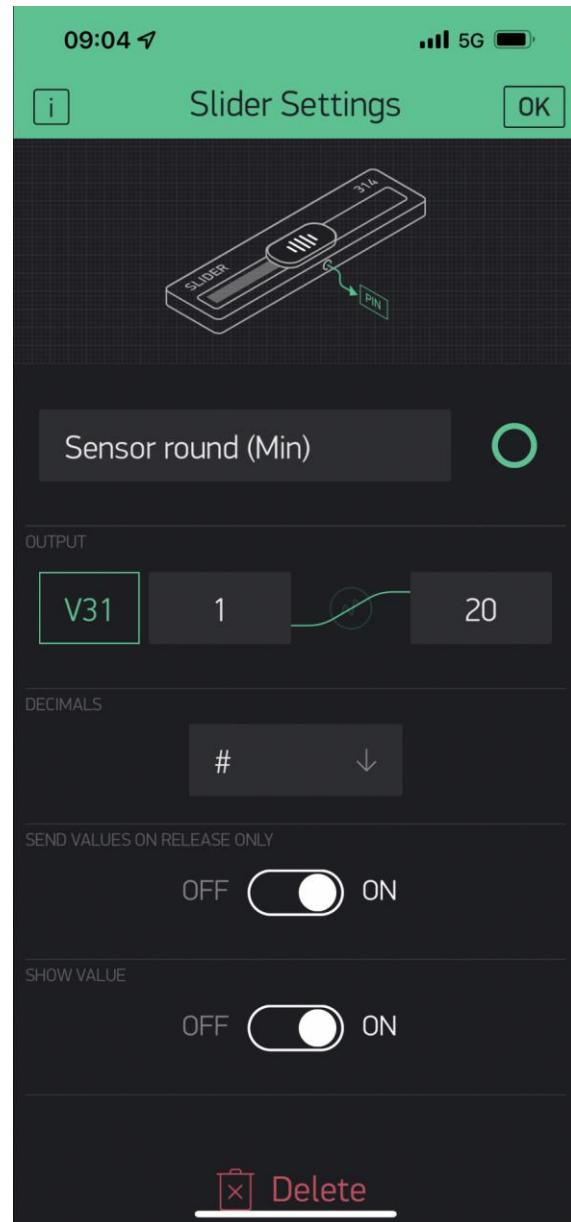


13:23

5G



- ส่วนที่ 10 จะประกอบด้วย Slider Settings

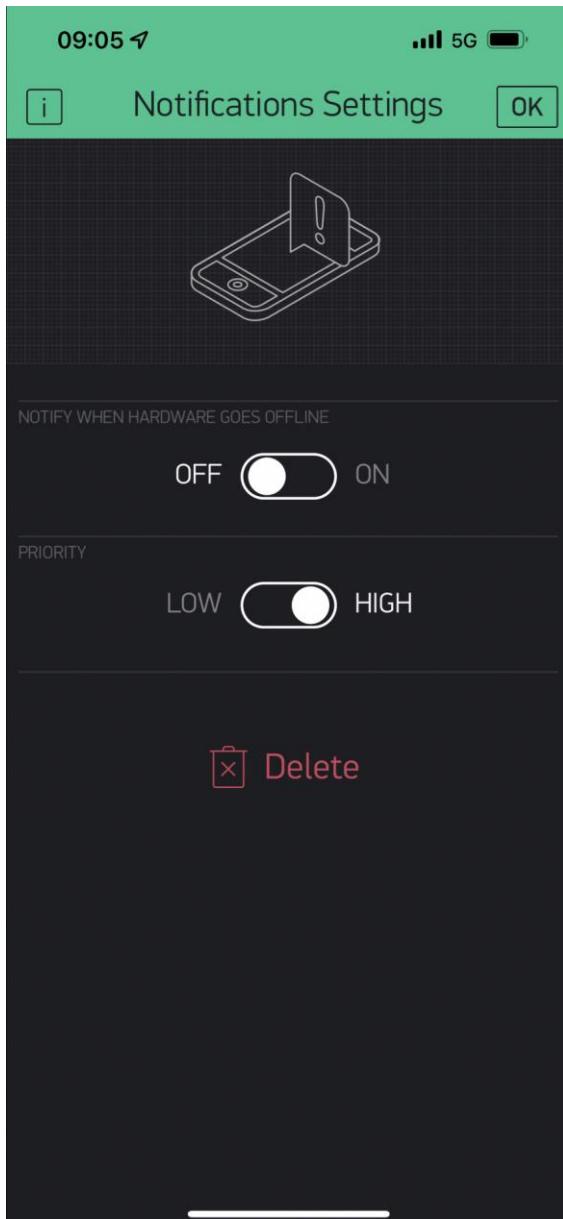
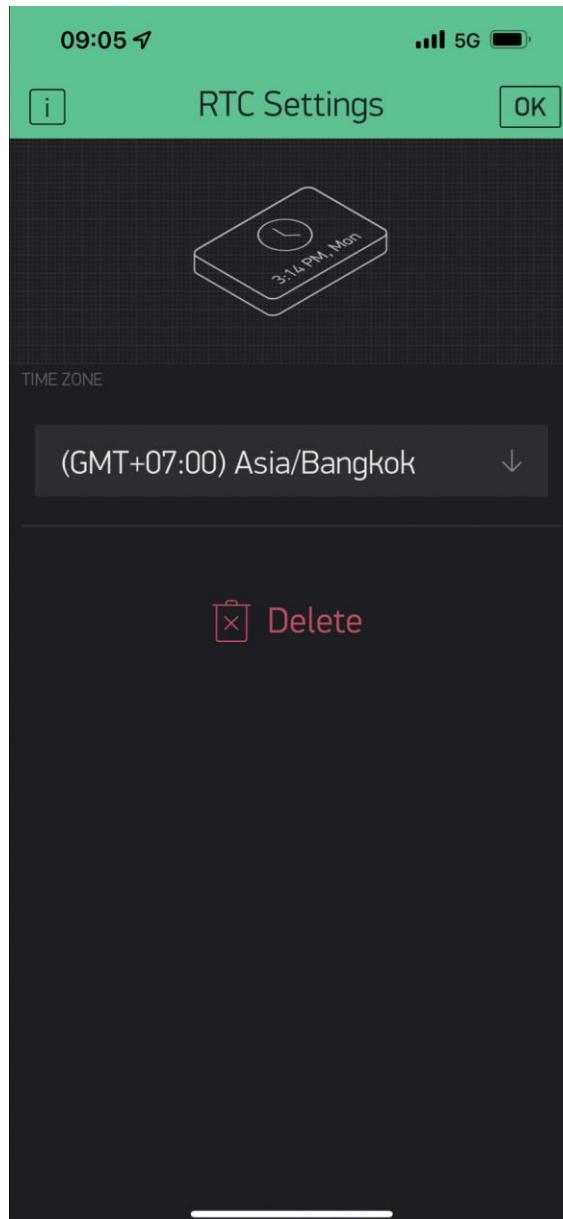


13:23

5G



- ส่วนที่ 11 จะประกอบด้วย RTC Setting และ Notification Settings

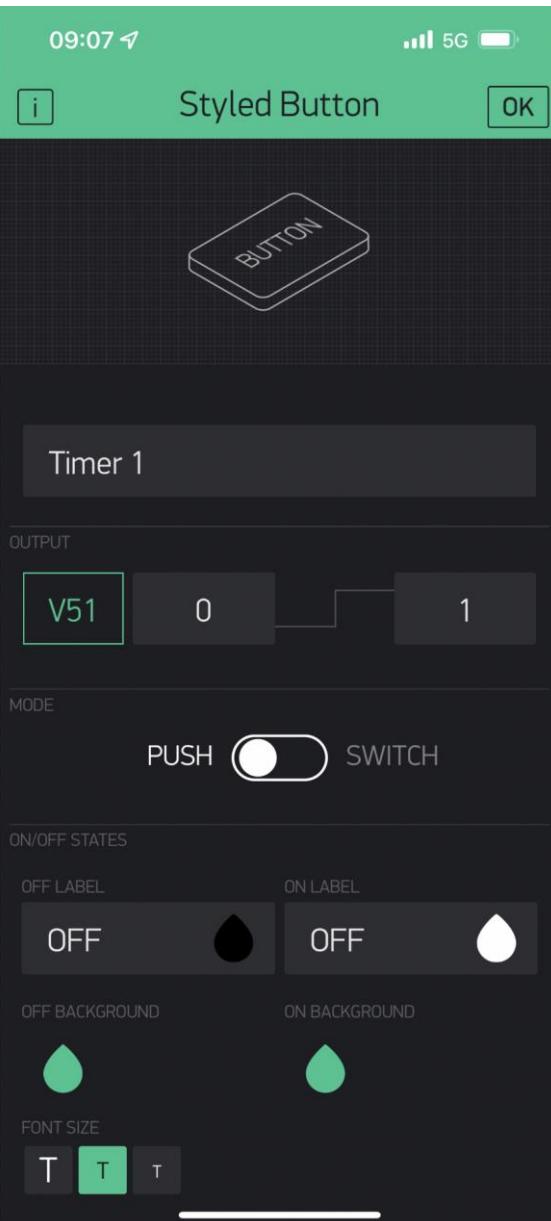
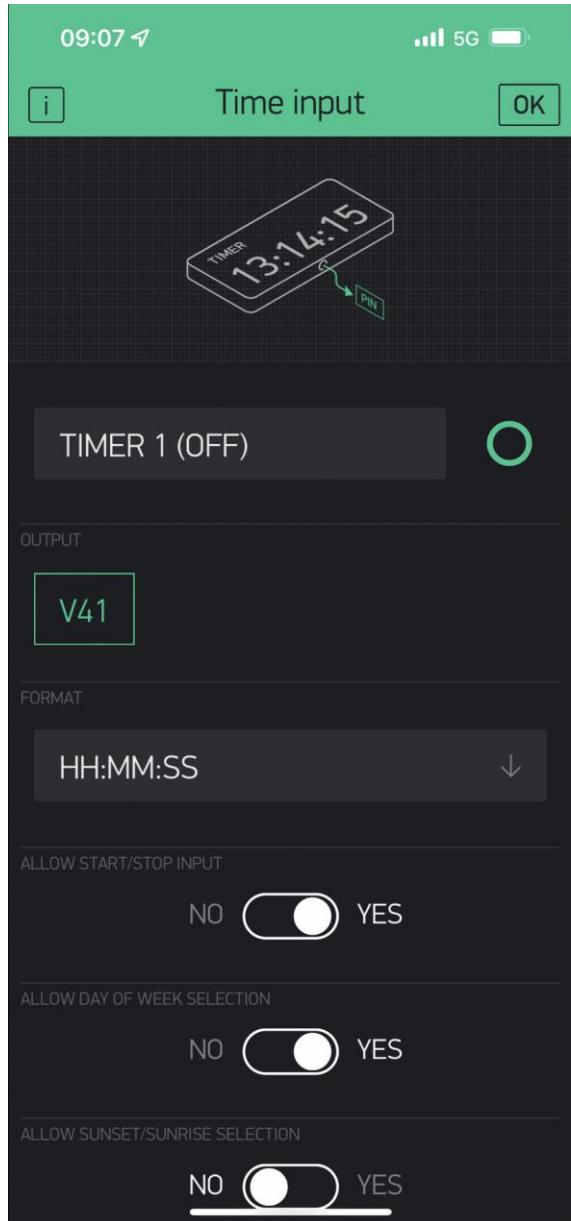


13:24

5G



- ส่วนของ Timer setting จะประกอบด้วย Time input และ Styled Button

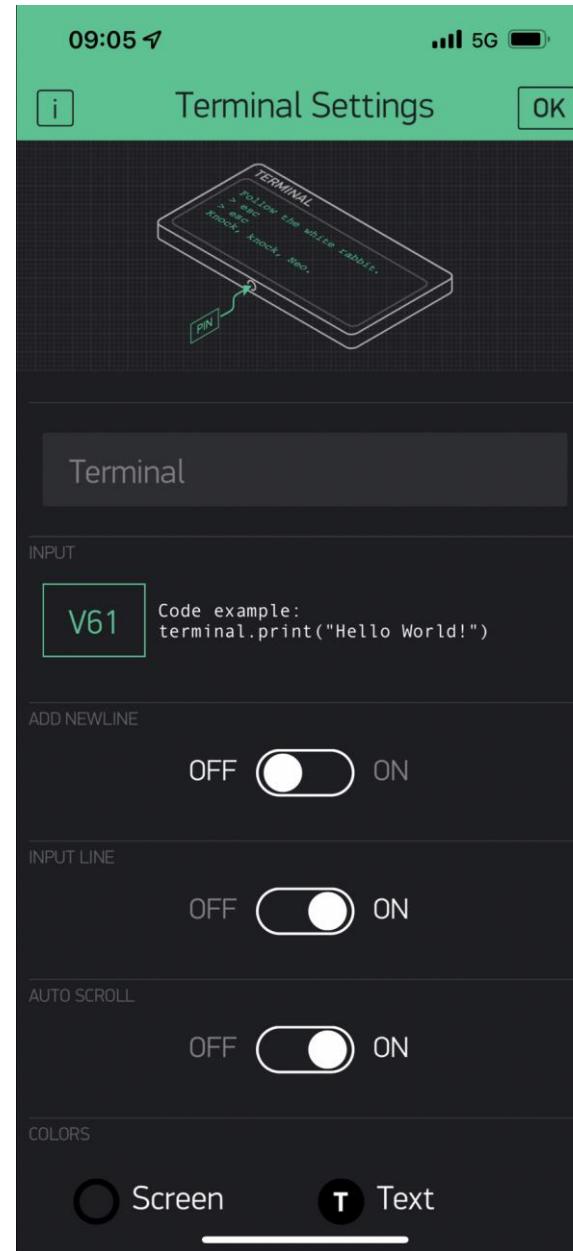


09:30 ↗

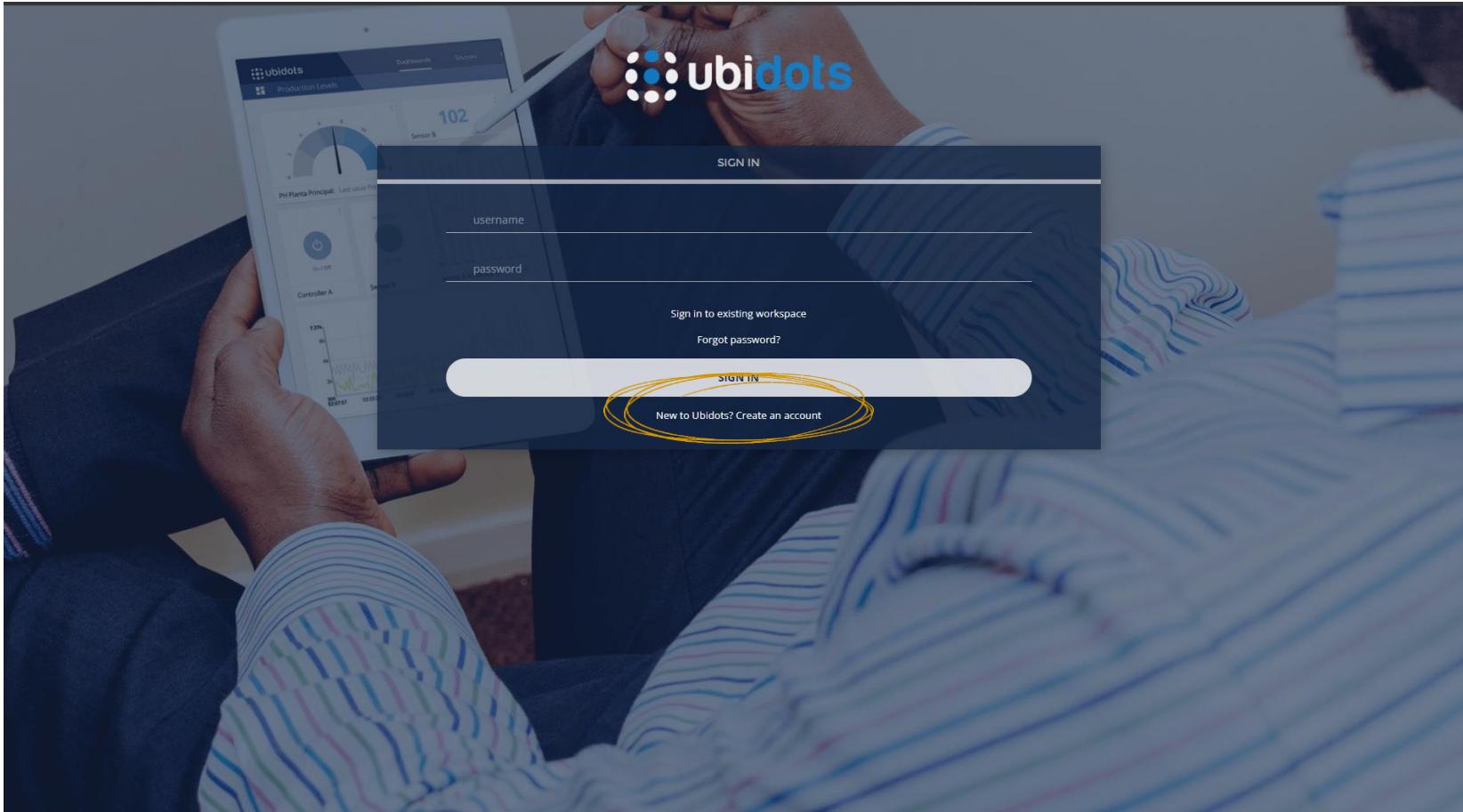
5G



- ส่วนของ Terminal ประกอบด้วย Terminal Settings



# Ubidots IoT Platform - Creating a new account



# Ubidots IoT Platform - Creating a new account

How would you like to use Ubidots?

Whether you're prototyping or going to market, Ubidots is designed to accelerate your IoT initiatives.

**For Educational or Personal Use**

Join 60,000+ students, makers and professors using our **FREE** Ubidots STEM platform to prototype, learn, or teach IoT.

[TAKE ME TO UBIODOTS STEM](#)

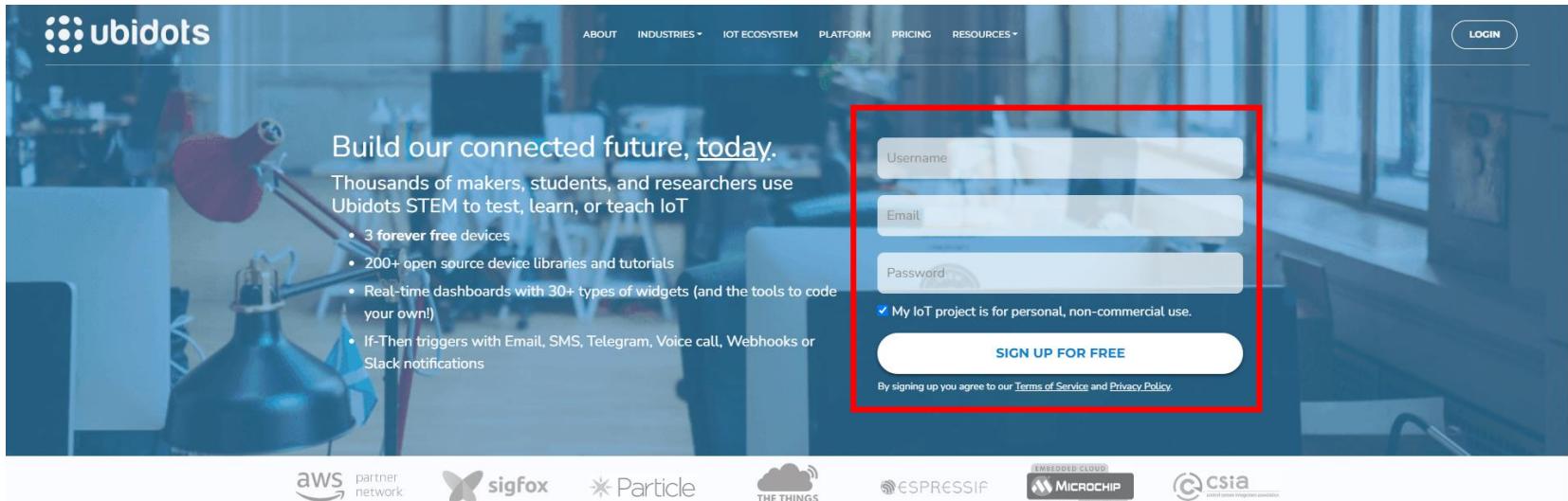
**For Business**

Join 1,000+ System Integrators, OEMs and IoT Entrepreneurs building connected products and services with Ubidots.

[CONTINUE →](#)

[Close](#)

# Ubidots IoT Platform - Creating a new account



The screenshot shows the Ubidots IoT Platform's sign-up page. At the top, there's a navigation bar with links for ABOUT, INDUSTRIES, IOT ECOSYSTEM, PLATFORM, PRICING, and RESOURCES. On the right side of the header is a "LOGIN" button. The main content area features a blue-tinted background image of a laboratory or workshop. To the left, there's promotional text: "Build our connected future, today." followed by a list of features: "3 forever free devices", "200+ open source device libraries and tutorials", "Real-time dashboards with 30+ types of widgets (and the tools to code your own!)", and "If-Then triggers with Email, SMS, Telegram, Voice call, Webhooks or Slack notifications". To the right of the text is a sign-up form enclosed in a red box. The form includes input fields for "Username", "Email", and "Password", and a checkbox labeled "My IoT project is for personal, non-commercial use." Below the form is a large blue "SIGN UP FOR FREE" button. At the bottom of the sign-up form, a small note states: "By signing up you agree to our [Terms of Service](#) and [Privacy Policy](#)".

**Build our connected future, today.**

Thousands of makers, students, and researchers use Ubidots STEM to test, learn, or teach IoT

- 3 forever free devices
- 200+ open source device libraries and tutorials
- Real-time dashboards with 30+ types of widgets (and the tools to code your own!)
- If-Then triggers with Email, SMS, Telegram, Voice call, Webhooks or Slack notifications

Username

Email

Password

My IoT project is for personal, non-commercial use.

SIGN UP FOR FREE

By signing up you agree to our [Terms of Service](#) and [Privacy Policy](#).

**aws partner network** **sigfox** **Particle** **THE THINGS NETWORK** **ESPRESSIF** **EMBEDDED CLOUD MICROCHIP DESIGN PARTNER** **csia**

**Education is Innovation**

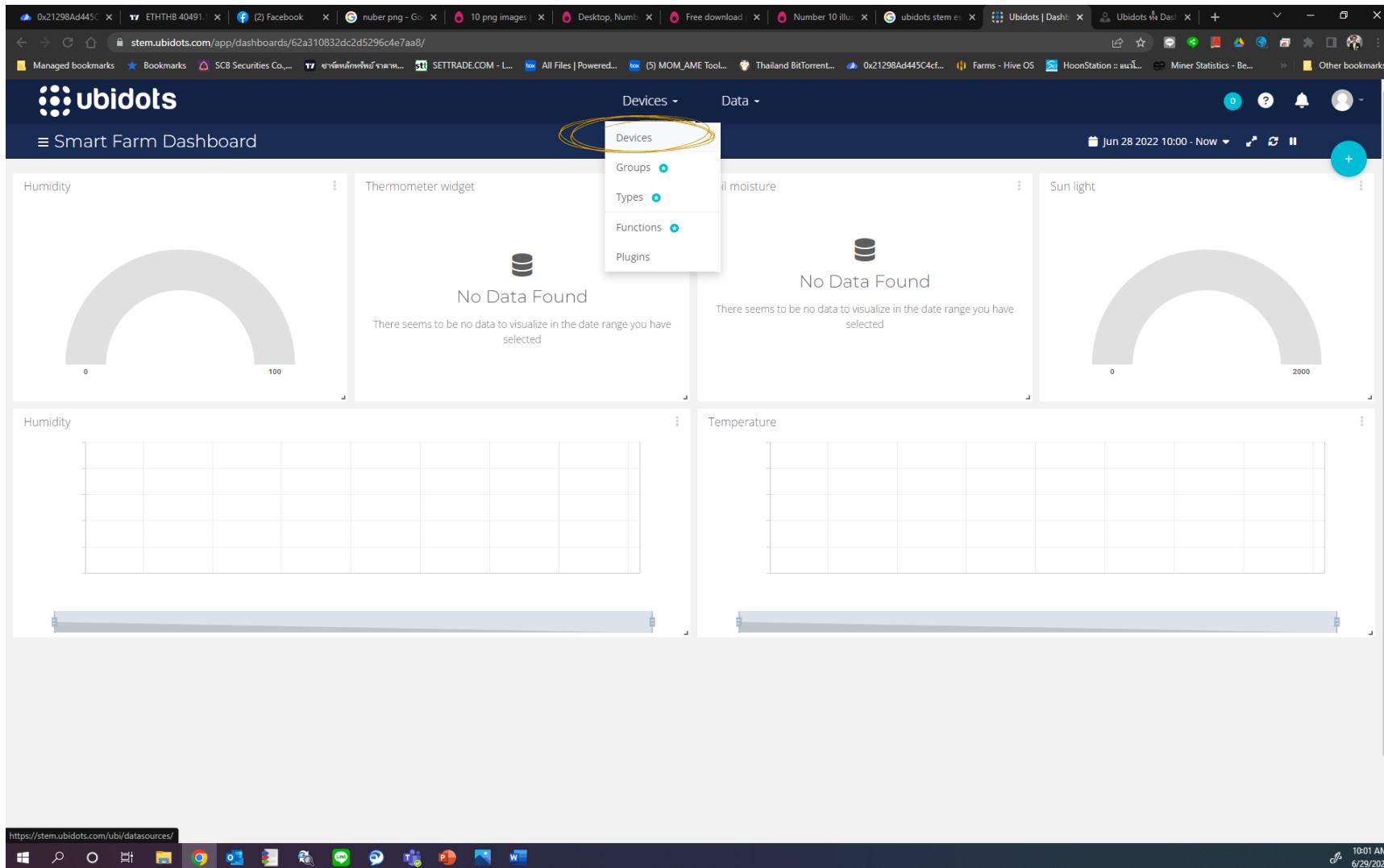
Ubidots and our partners are committed to the expansion and availability of problem solving technologies. We believe personal tech exploration will yield global externalities towards the economization of precious resources many take for granted – clean water, quality food, and personal safety to name a few.

 Environmental Preservation

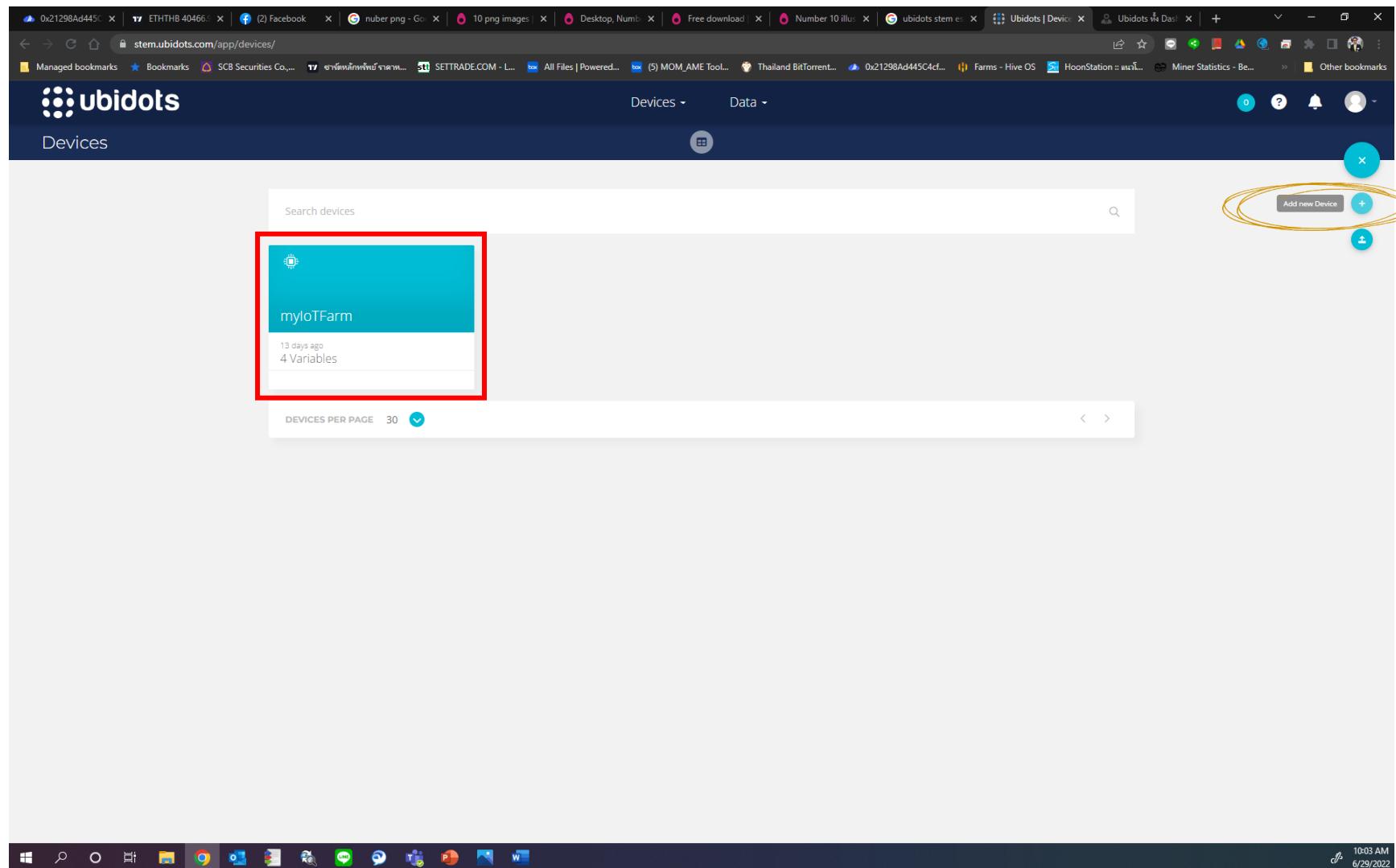
 Developing Regions

 STEAM Education

# ไปที่หัวข้อ Devices แล้วทำการสร้าง Device ใหม่



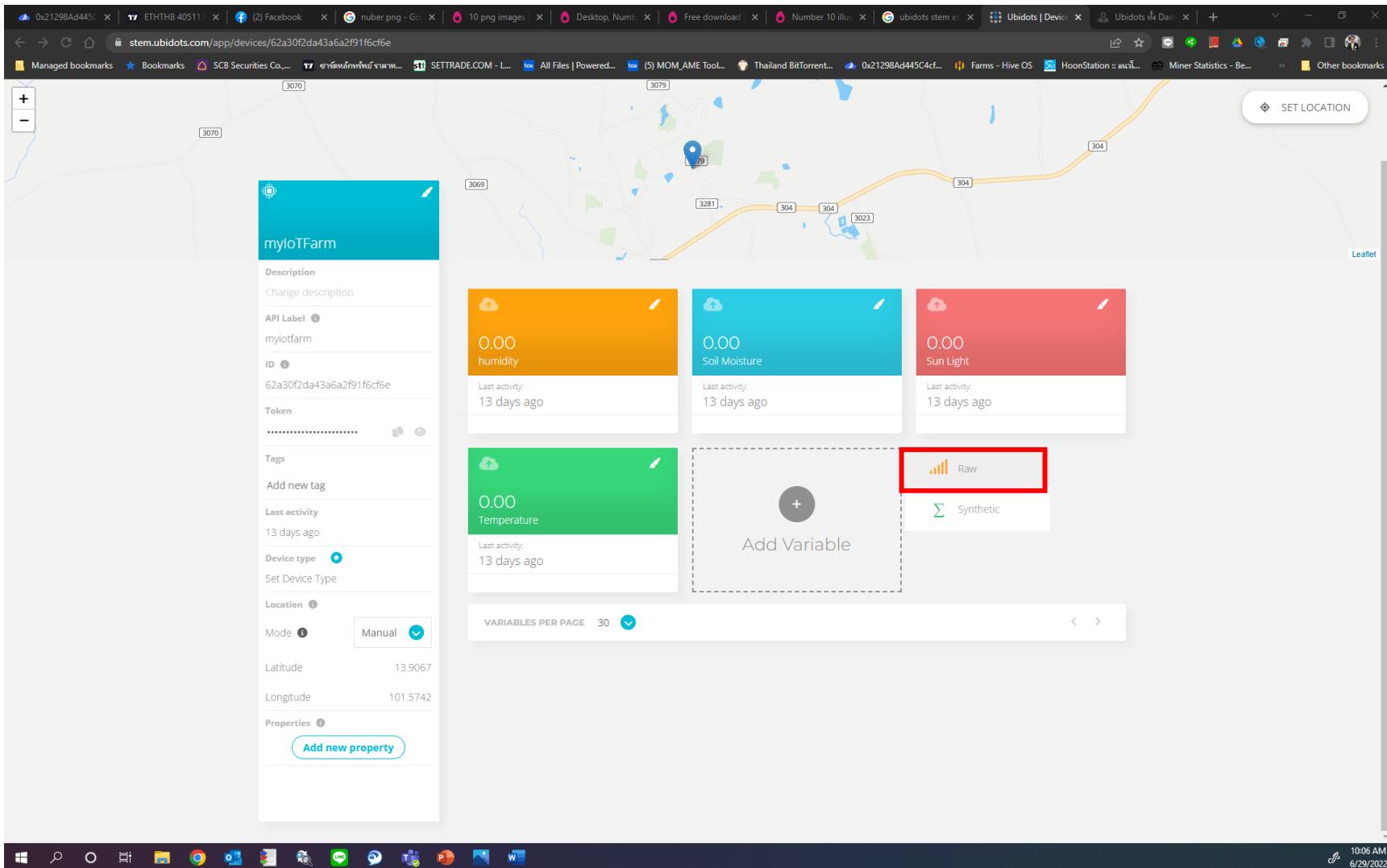
# กดที่ Add new Device และพิมพ์ชื่อ Device ตามต้องการ



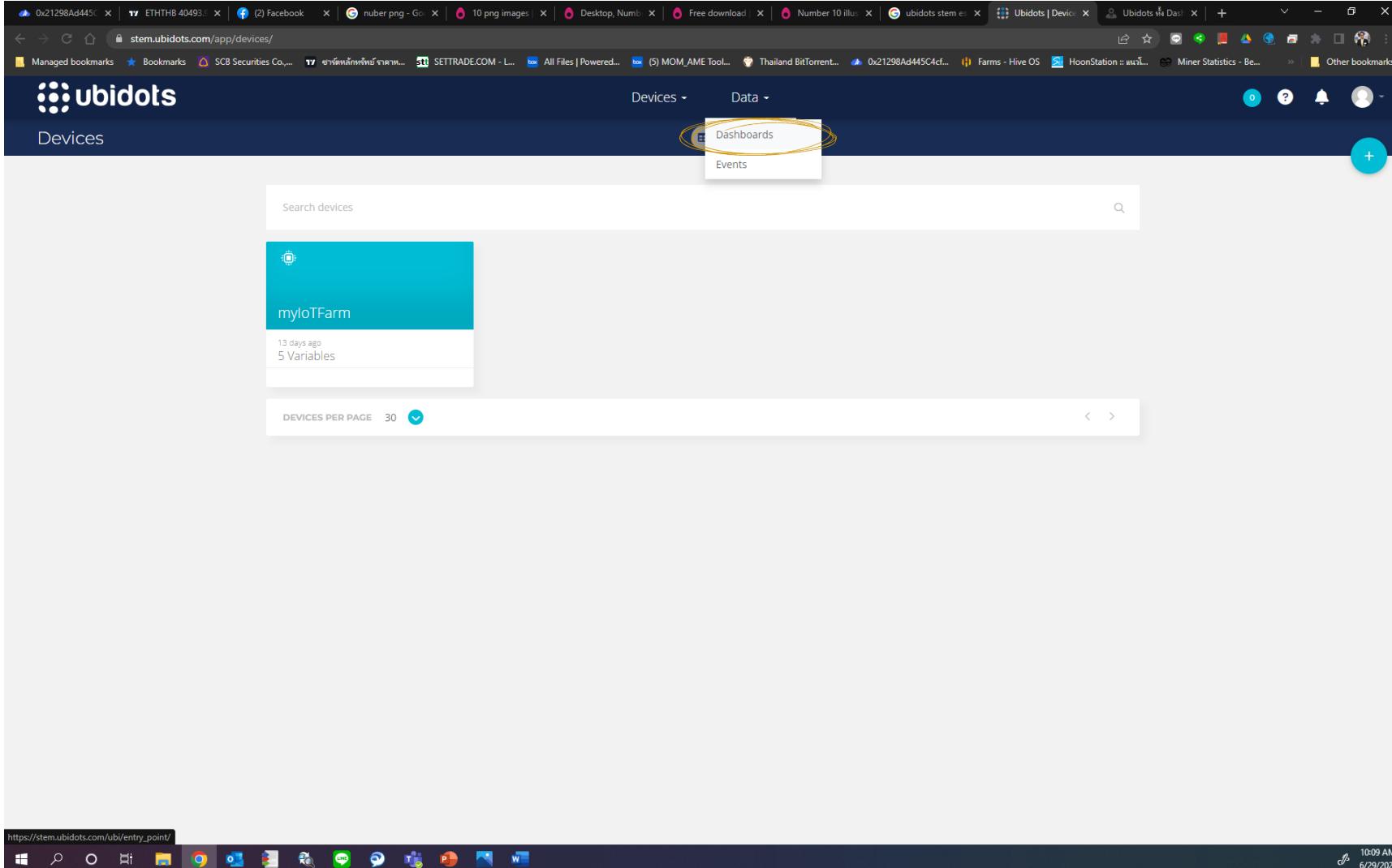
# สร้างตัวแปรใหม่โดยกด Add Variable

The screenshot shows the Ubidots platform interface. On the left, there is a sidebar for a device named "myIoTFarm" with fields for Description, API Label, ID, Token, Tags, Last activity, Device type, Location, Mode, Latitude, Longitude, and Properties. At the bottom of this sidebar is a blue button labeled "Add new property". The main area displays a map with several nodes and their IDs (e.g., 3070, 3079, 3069, 3281, 304, 3023) and a list of variables. The variables are represented by colored cards: orange for humidity (0.00), cyan for Soil Moisture (0.00), red for Sun Light (0.00), green for Temperature (0.00), and a card with a plus sign labeled "Add Variable" which is highlighted with a red dashed box. Below the variable list is a pagination control showing "VARIABLES PER PAGE" set to 30.

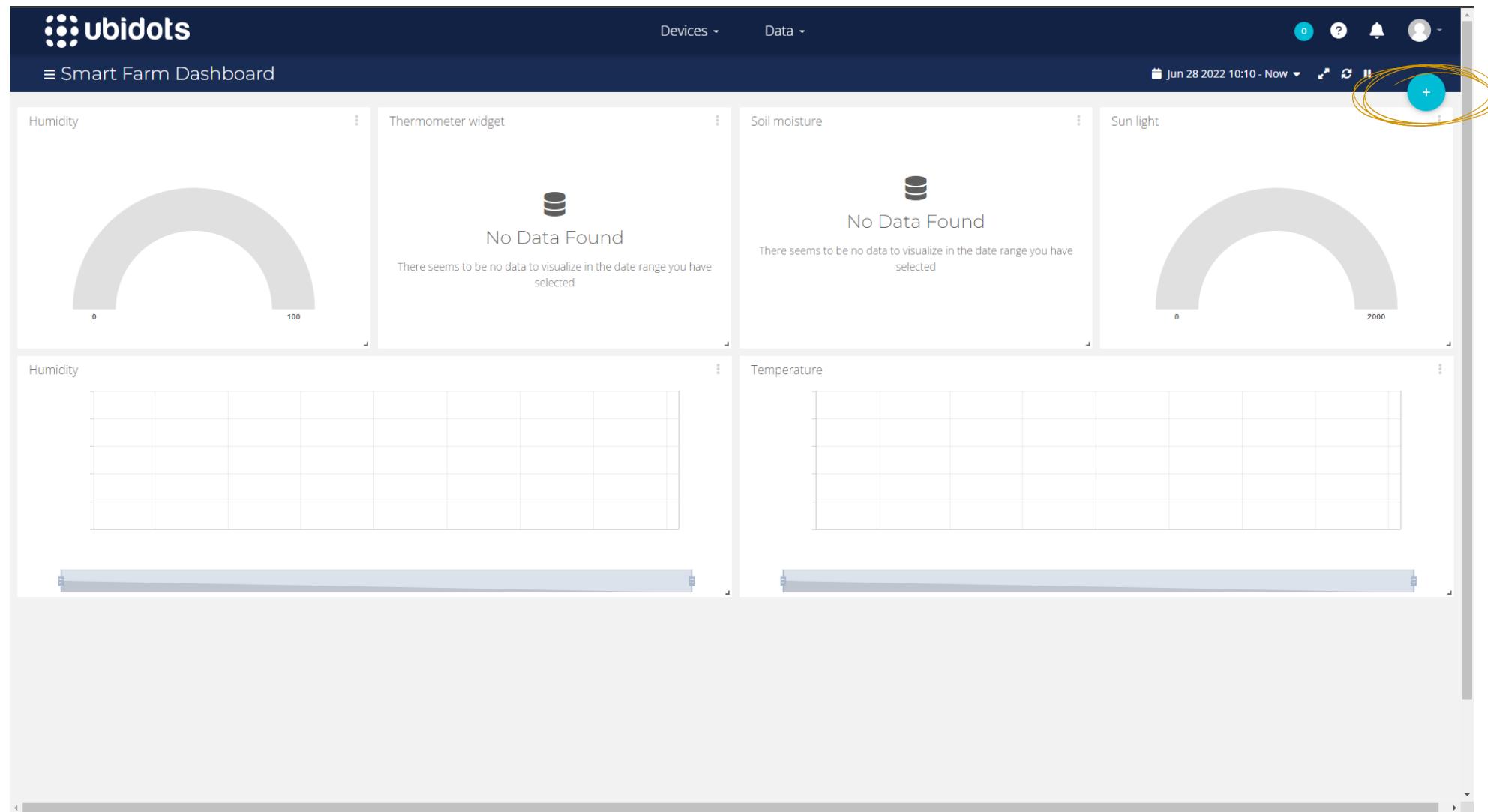
# สร้างตัวแปรปกติให้เลือกเป็น Raw



# สร้าง Dashboard



# ทำการเพิ่ม Widget เข้าไปใน dashboard โดยกดที่เครื่องหมาย +



# Add new widget - เลือกการแสดงผลเป็นแบบ Gauge

The screenshot shows the Ubidots Smart Farm Dashboard interface. At the top, there are navigation menus for 'Devices' and 'Data'. Below the header, there are four main sections: 'Humidity' (a gauge chart), 'Thermometer widget' (a chart showing 'No Data Found'), 'Soil moisture' (a chart showing 'No Data Found'), and 'Temperature' (a chart showing 'No Data Found'). A red box highlights a modal window titled 'Add new widget' which lists various visualization options.

**Add new widget**

Bar chart	Battery	Clock	Devices Table
Double Axis	Gauge	HTML Canvas	Histogram
Image	Indicator	Line chart	Manual input
Map widget	Metric	Pie	Rose chart
Scatter	Slider	Switch	Tank
Text	Thermometer	Values table	Variables table

# จากนั้นให้เลือก + Add variable

The image shows the Ubidots Smart Farm Dashboard interface. On the left, there are four cards: 'Humidity' (a gauge), 'Thermometer widget' (No Data Found), 'Soil moisture' (No Data Found), and 'Temperature' (a chart). A red box highlights the 'Gauge' configuration dialog on the right.

**Gauge**

[BACK](#)

**Data**

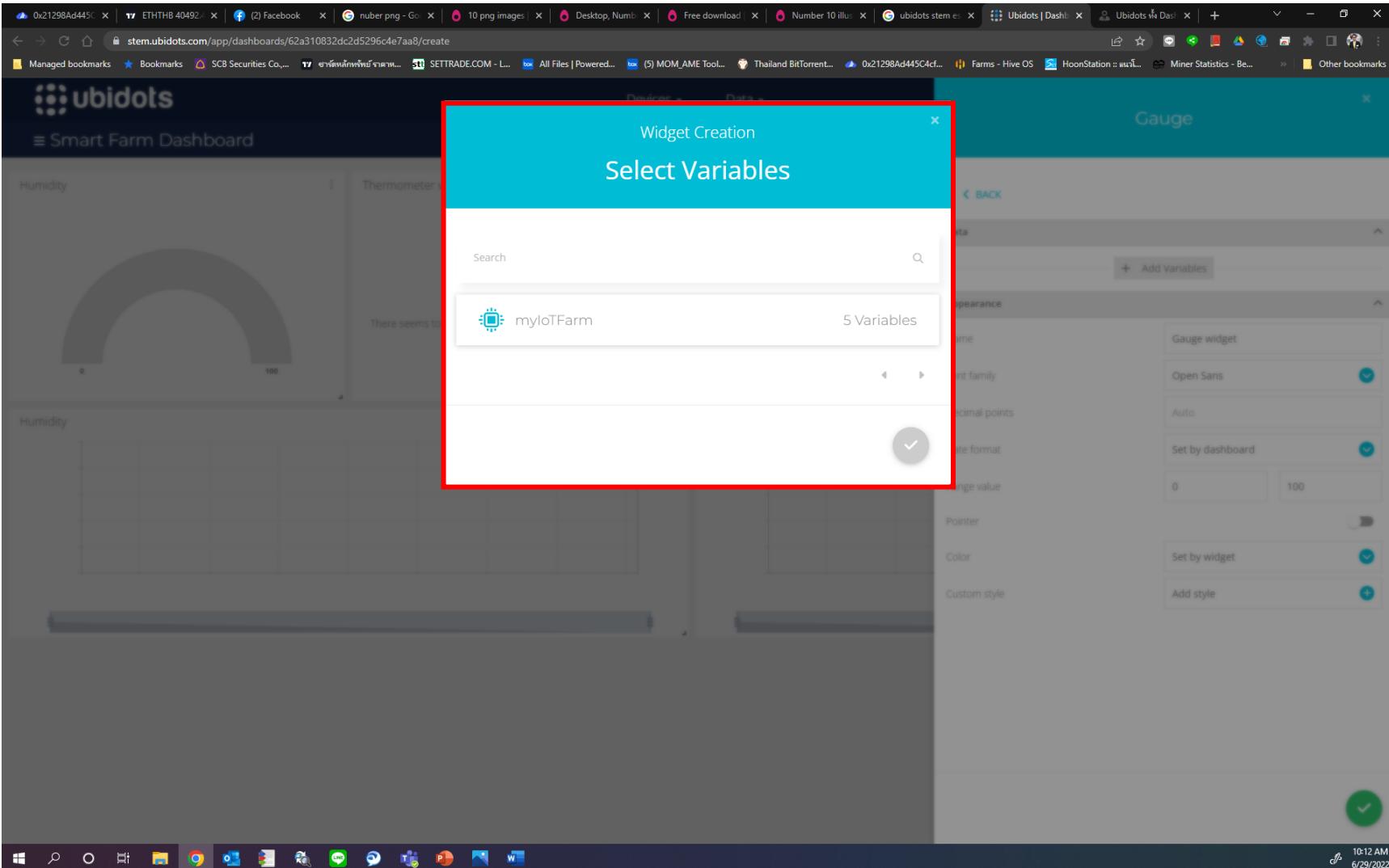
+ Add Variables

**Appearance**

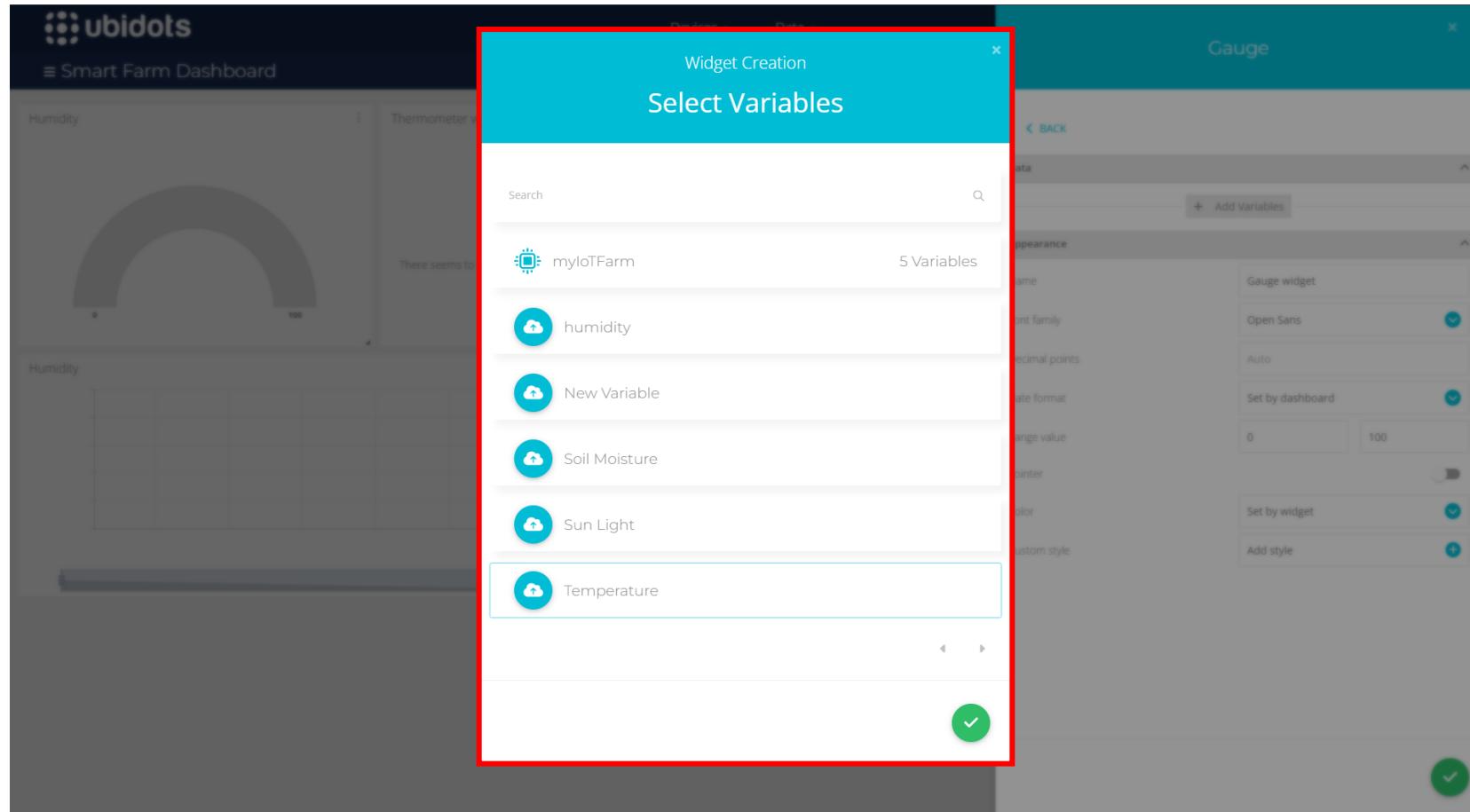
Name: Gauge widget  
Font family: Open Sans  
Decimal points: Auto  
Date format: Set by dashboard  
Range value: 0 to 100  
Pointer:  
Color: Set by widget  
Custom style: Add style

✓

# ເລືອກຕົວແປຣ



# เลือกตัวแปร Temperature



# Ubidots Coding

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** SWCM\_SmartFarm\_IoT\_Rev1 - config.h | Arduino 1.8.19
- Menu Bar:** File, Edit, Sketch, Tools, Help
- Toolbar:** Includes icons for Save, Undo, Redo, Open, Upload, and Download.
- Sketch Navigator:** Shows tabs for SWCM\_SmartFarm\_IoT\_Rev1, Auto\_mode, Blynk\_app, Connection, Function, Line\_notify, config.h (highlighted), helper.h, and timer.
- Code Editor:** Displays the config.h code for the SWCM\_SmartFarm\_IoT\_Rev1 sketch. A red box highlights the following section of code:

```
23 const char *UBIDOTS_TOKEN = "BBFF-Rd01QXKprppS3GIFmGRCqSbacSgtUU"; // Put here your Ubidots TOKEN
24 const char *DEVICE_LABEL = "myiotfarm"; // Put here your Device label to which data will be sent
25 const char *VARIABLE_LABEL = "Humidity"; // Put here your Variable label to which data will be sent
26 const char *VARIABLE_LABEL2 = "soil-moisture";
27 const char *VARIABLE_LABEL3 = "light";
28 const char *VARIABLE_LABEL4 = "temperature";
```

**Status Bar:** Done Saving.

**Page Number:** 8

**Bottom Status:** Node32s, Default, 80MHz, 921600, None on COM5

# Complete Dashboard

