โรงเรียนบัวขาว

เอกสารประกอบการเรียน

เพื่อการพัฒนา การประกอบ และเขียนโค้ดโปรแกรม IoT

1

นายร่วมชาติ ชัยนา

ตำแหน่งครู วิทยฐานะครูชำนาญการพิเศษ

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี

โรงเรียนบัวขาว สังกัดองค์การบริหารส่วนจังหวัดกาฬสินธุ์

คำนำ

ในอดีตเทคโนโลยีเป็นเพียงสื่อกลางที่ใช้ในการติดต่อสื่อสารของผู้ใช้รายหนึ่ง ถึงผู้ใช้อีกรายหนึ่ง แต่ต่อมามีการพัฒนาแนวคิด Internet of Thing ขึ้น และได้เริ่มมีการวิจัย และสร้างขึ้นเพื่อใช้งานจริงในปัจจุบัน ซึ่งแนวคิดดังกล่าวเป็นแนวคิดที่ได้พัฒนาขีดความสามารถของอุปกรณ์เทคโนโลยีขึ้นไปอีกขั้น ที่นอกจากจะเป็นสื่อกลางในการติดต่อสื่อสารได้แล้ว ยังสามารถติดต่อสื่อสารกับอุปกรณ์ด้วยกันเอง เพื่อทำงานร่วมกัน และช่วยอำนวยความสะดวกแก่ผู้ใช้งาน

ในปัจจุบันเทคโนโลยีเข้ามามีบทบาทต่อการดำเนินชีวิตของมนุษย์เรามากขึ้นกว่าเมื่อก่อนมาก ซึ่งการใช้เครือข่ายอินเทอร์เน็ต ไม่ว่าจะเป็นการสื่อสาร การศึกษา การค้นหาข้อมูล เป็นต้น จนเครือข่ายอินเทอร์เน็ตเป็นเรื่องปรกติในสังคมปัจจุบัน และด้วยความที่เทคโนโลยีมีการพัฒนาไปอย่างกว้างขวาง จึงได้มีการริเริ่มแนวคิด Internet of Thing หรือ IoT เป็นการเชื่อมโยงเทคโนโลยีทั้งหมดเข้าด้วยกัน สำหรับในเอกสารฉบับนี้ก็จะมีการกล่าวถึง อุปกรณ์เกี่ยวกับ IoT การเชื่อมต่ออุปกรณ์ต่างๆ และคำสั่งที่จำเป็นสำหรับใช้งาน เป็นต้น ในการจัดทำเอกสารในครั้งนี้หวังว่าจะเป็นประโยชน์ต่อผู้ที่ต้องการศึกษา หรือมีความสนใจไม่มากก็น้อยหากมีข้อผิดพลาดประการใด ผู้จัดทำ และรวบรวมข้อมูลก็ขออภัยมา ณ ที่นี้ด้วย และขอขอบคุณเจ้าของบทความและผู้เผยแพร่โค้ดโปรแกรมเป็นอย่างยิ่ง ที่ให้ความอนุเคราะห์ข้อมูลในการจัดทำเอกสารในครั้งนี้

ด้วยความขอบคุณจากผู้จัดทำ

หัวข้อ/ รายการอุปกรณ์	หน้า

1. บอร์ดทดลอง NodeMCU V2 NodeMCU V2 ESP8266 Development Kit ESP-12F/N 1
2. บอร์ดทดลอง Breadboard 400 holes 1
3. Relay 5V 1 ช่อง แบบ Active High บอร์ดรีเลย์ 5V สำหรับ Arduino ESP8266 ESP32 2
4. บอร์ด Relay 2 ช่อง 5V relay 5v แบบ Active High/Low 10A 250V สำหรับ Arduino และ Microcontroller 2
5. Relay Module 4 Channel 5V relay 4 ช่อง แบบ isolation control Relay Module Shield 250V/10A แบบ Active Low for Arduino 4
6. กล่องใส่ของพลาสติก Electronics box กล่องเก็บอุปกรณ์อิเล็กทรอนิกส์ 225x150x60mm 5
7. สายไฟ จัมเปอร์ ผู้-ผู้ ยาว 10cm. 5
8. สายไฟ จัมเปอร์ ผู้-เมีย ยาว10cm. 5
9. ตัวต้านทาน 10K 1/4W Metal film 1% 10 k 6
10. ตัวต้านทาน 100 K 1/4W Metal film 100k 1% 6
11. ตัวต้านทาน 10 โอห์ม 10r 1/4W Metal film 1% 6
12. ตัวต้านทาน 3.3 K 1/4W Metal film 1% 3.3k 6
13. ตัวต้านทานปรับค่าได้ 100k potentiometer R 100 k 7
14. สาย Micro USB อย่างดี ยาว 50cm จ่ายกระแส 3A สำหรับ Arduino ESP8266 ESP32 NodeMCU 7
15. USB Charger Doctor 7

ชุดหลอดไฟแสงสว่างและเซ็นเซอร์ควบคุม

16. LDR Photoresistor 10mm ldr 10mm 8
17. LDR 20mm Photocell Photoresistor ตัวต้านทานปรับค่าได้ตามแสง LDR ขนาด 20MM 9
18. LDR 20mm Photocell Photoresistor ตัวต้านทานปรับค่าได้ตามแสง LDR ขนาด 20MM 10
19. LED 5mm 13
20. LED traffic light module 5V traffic light module 14
21. RGB LED module full color LED module โมดูลไฟ LED RGB 20
22. LED เส้นผ่านศูนย์กลาง 68mm 23
23. NeoPixel Ring 8 WS2812 RGB LED เส้นผ่านศูนย์กลาง 32mm 25

ชุดจอภาพแสดงผล

24. จอแสดงผล OLED สีขาว 128x64 แบบ I2C ขนาด 1.3" สีขาว OLED Display I2C Module 1.3" for Arduino 30
25. 1602 LCD (Blue Screen) 16x2 LCD with backlight of the LCD screen 35
26. 1602 2004 LCD Adapter Plate IIC I2C Interface for Arduino 36
27. Active Buzzer Module 37

ชุดเซ็นเซอร์ควบคุมต่างๆ

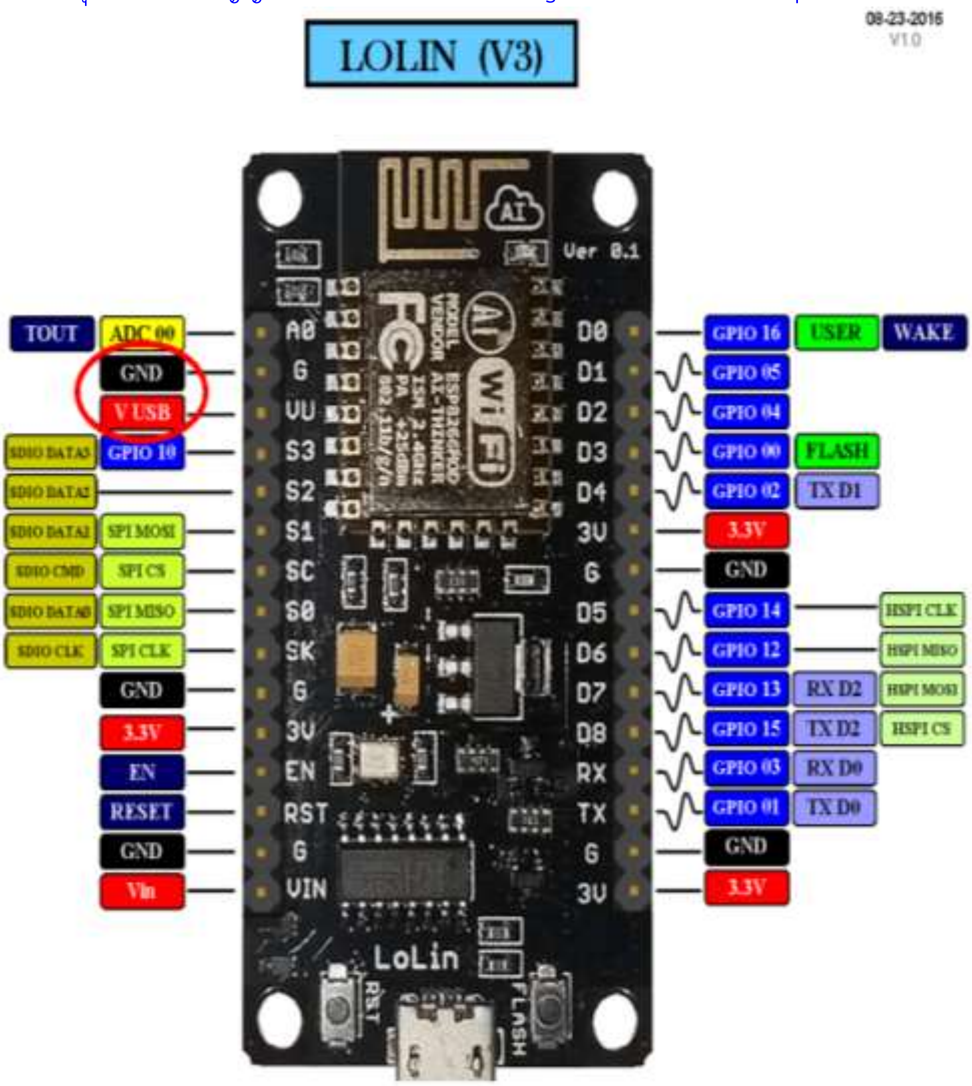
Sensor ตรวจจับการเคลื่อนไหว PIR (HC-SR501) 40
28. Ultrasonic SR04 เซนเซอร์ Ultrasonic Module HC-SR04 Distance Measuring 42

Transducer Senso
29. Soil Moisture Sensor Module เซ็นเซอร์วัดความชื้นในดิน 45
สอนแสดงค่าความชื้นในดินในแอพ Blynk (ดูรูปประกอบท้ายเล่ม) 47
30. Temprature and Humidity Sensor DHT11 module 48
บอร์ด ESP8266 กับ DHT11 แสดงค่าอุณหภูมิและความชื้นด้วยหน้าจอ LCD I2C 54
31. Micro PIR human body sensing module PIR module infrared module 54
เซนเซอร์ PIR ตรวจจับความเคลื่อนไหวของสิ่งมีชีวิต
32. Digital Temperature Temp Sensor Probe DS18B20 For Thermometer 57
Waterproof 100CM
33. โมดูลแปลง Adapter DS18B20 เซนเซอร์วัดอุณหภูมิแบบกันน้ำ แปลงสายไฟให้เป็นขา 60
แบบก้างปลา
34. ISD1820 Voice Board Module On-board Microphone Sound Recording โมดูล 62
อัดเสียงบันทึกเสียงและเล่นเสียง ควบคุมด้วย Arduino ได้

การใช้งานผ่าน Blynk และตัวอย่างการใช้งานต่างๆ

การเชื่อมต่อไวไฟ ด้วย ESP8266 ให้แสดง IP และ Mac Address และให้หลอดไฟติดเมื่อสำเร็จ 69
แอพพลิเคชั่น Blynk ผ่าน Relay ควบคุมหลอดไฟฟ้า 70
Smart IoT ESP8266 ควบคุมปั๊มน้ำ และตรวจเช็คความชื้นในดิน ด้วยแอพ Blynk บนมือถือ 72
App สำเร็จรูป Blynk Nodemcu esp8266 (ตอนที่ 1 Blynk คืออะไร) 74
ขั้นตอนการใช้งานแอพ Blynk วัดค่าความชื้นในดิน 77
ขั้นตอนในโปรแกรม Arduino IDE 81
NodeMCU Wifi Blynk Button ควบคุมเปิดปิดไฟ LED 84
การใช้งาน ESP8266 ตั้งเวลาเปิด/ปิดไฟ ด้วย 96
โปรเจคเปิด-ปิด LED ผ่าน WiFi ด้วย Blynk 99
Nodemcu esp8266 Arduino IDE เปิดปิดไฟผ่าน Website 103
ทำ โครงงาน โครงการ โปรเจคอะไรดี โปรเจคจบ (Arduino Web Program App) 109
เรามาทำความรู้จักเกี่ยวกับ Json กันก่อน 112
Arduino เชื่อมต่อกับ Database (Database to Arduino) 115
การเก็บข้อมูลจาก Nodemcu Esp8266 ลงใน ฐานข้อมูล Database Mysql (Arduino to database) 118
การเก็บข้อมูลจาก Arduino ลงใน ฐานข้อมูล Database Mysql (Arduino to database) 121
วัดอุณหภูมิ ด้วย Nodemcu Esp8266 อัพค่าขึ้น thingspeak.com กันเถอะ 124
ค้นหา Address I2C อุปกรณ์ ด้วย Arduino 128
Nodemcu Esp8266 กับการส่งการแจ้งเตือนเข้า LINE ส่งข้อความ ภาพ และสติกเกอร์ 130
Fritzing ซอฟต์แวร์ออกแบบวงจรและแผ่นปริ้นให้กับ Arduino 135
แก้ปัญหา wdt reset ของ Nodemcu esp8266 เขียน Code พัฒนาด้วย Arduino ide 138
แก้ปัญหา NodeMcu Esp8266 ขึ้น error: espcomm_upload_mem failed 139
แก้ปัญหา ใช้งาน Nodemcu ไม่ได้ package esp8266 is unknown 140
รถบังคับ Rc Car wifi ด้วย Nodemcu esp8266 Arduino IDE ควบคุมผ่านมือถือ android 142

โปรเจคเครื่องรดน้ำต้นไม้อัตโนมัติ ด้วย Arduino ราคาถูก พร้อม Code ตัวอย่าง 148
NodeMCU Esp8266 เปิดปิดไฟควบคุมผ่าน App มือถือ แสดงผลบนจอ LCD 151
Mini Project ตลับเมตรไร้สาย แสดงบนจอ LCD พัฒนาด้วย Arduino IDE 156
Mini Project Arduino Nodemcu แจ้งเตือน น้ำท่วม น้ำล้น ผ่าน Line Notify 158
Mini Porject Arduino PH Sensor (วัดคุณภาพน้ำความเป็นกรดเป็นด่าง) 160
Mini Project Arduino Nodemcu แจ้งเตือนฝนตกผ่าน Line 163
สอนวัดอุณหภูมิ DHT22 แสดงผ่านหน้าจอ LCD ด้วย Nodemcu และ Arduino 166
มาสร้างอุปกรณ์ ตัดสัญญาณ Wifi (Wi-Fi Jamming) ด้วย Nodemcu esp8266 168



ADC	PIN DE SALIDA ANALÓGICA (el rango es entre +0V y +1V dividido en 1023 intervalos).
SPI	BUS SPI (*Serial Peripheral Interface*)
HSPI	BUS HSPI (*Hardware Serial Peripheral Interface*).
SDIO	PINES PARA INICIO DEL ESP8266 DESDE UNA TARJETA SD. Para activar el modo SDIO el pin GPIO 15 debe estar en tensión cuando se enciende la placa.
TX/RX	COMUNICACIÓN SERIE TX/RX. Los pines GPIO01 y GPIO02 están conectados al puerto USB a través del conversor UART.

1. บอร์ดทดลอง NodeMCU V2 NodeMCU V2 ESP8266 Development Kit ESP-12F/N

NodeMCU V2 รุ่นล่าสุด เปลี่ยนจากรุ่นเดิมที่ใช้ชิฟ ESP-12E เป็นชิฟ ESP-12F/N รุ่นล่าสุด ที่ปรับปรุงสายอากาศให้รับส่งสัญญาณได้ดีขึ้น

มีขา GPIO PWM I2C 1-Wire ADC และ SPI เพิ่มเข้ามา มีเสาอากาศในตัว บอร์ดมีขนาดเล็กลง บอร์ดกว้าง 2.5CM ใช้ชิฟ USB เบอร์ CP2102 ในการติดต่อกับคอมพิวเตอร์เพื่อลงโปรแกรม สามารถลง Firmware NodeMCU และเขียนโปรแกรมด้วยภาษา Lau และ Arduino ได้

ข้อมูล NodeMCU V2 LUA based ESP8266-12E NodeMCU V2 Development Kit

Arduino-like hardware IO

Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware. Code like arduino, but interactively in Lua script.

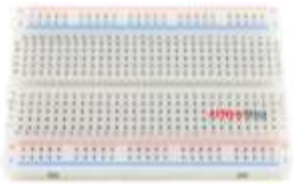
Nodejs style network API

Event-driven API for network applicaitons, which faciliates developers writing code running on a 5mm*5mm sized MCU in Nodejs style. Greatly speed up your IOT application developing process.

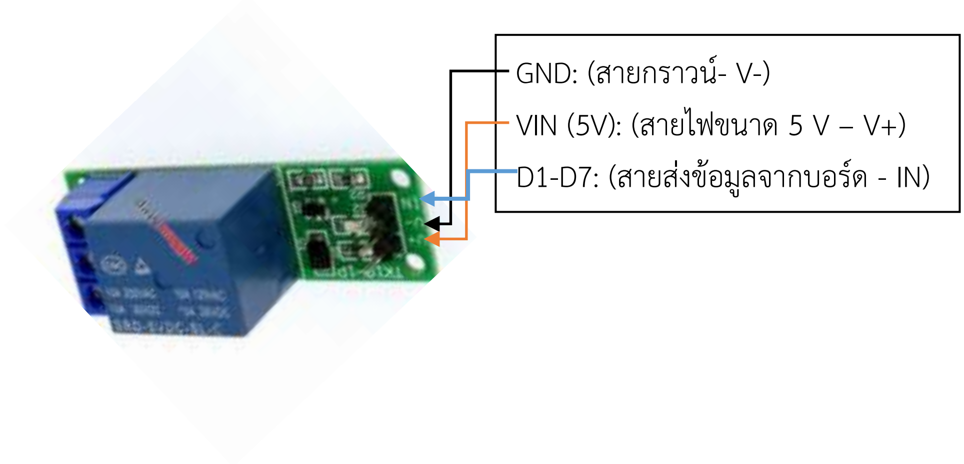
การใช้งาน NodeMCU อัพโหลด

1. เช็คว่าลง Driver แล้ว โดยจะเห็น Comport
2. เช็คว่า ESP8266 ทำงานปกติ โดยจะเห็นชื่อ Wi-Fi
3. ถ้าลงไดรเวอร์แล้วยังไม่เห็น comport หรืออัพโหลดไม่เข้า ให้กดปุ่ม Flash ค้างไว้ กดและปล่อยปุ่ม Reset แล้วปล่อยปุ่ม Flash ก็จะมองเห็น comport และ อัพโหลดได้ปกติ

2. บอร์ดทดลอง Breadboard 400 holes



3. Relay 5V 1 ช่อง แบบ Active High บอร์ดรีเลย์ 5V สำหรับ Arduino ESP8266 ESP32



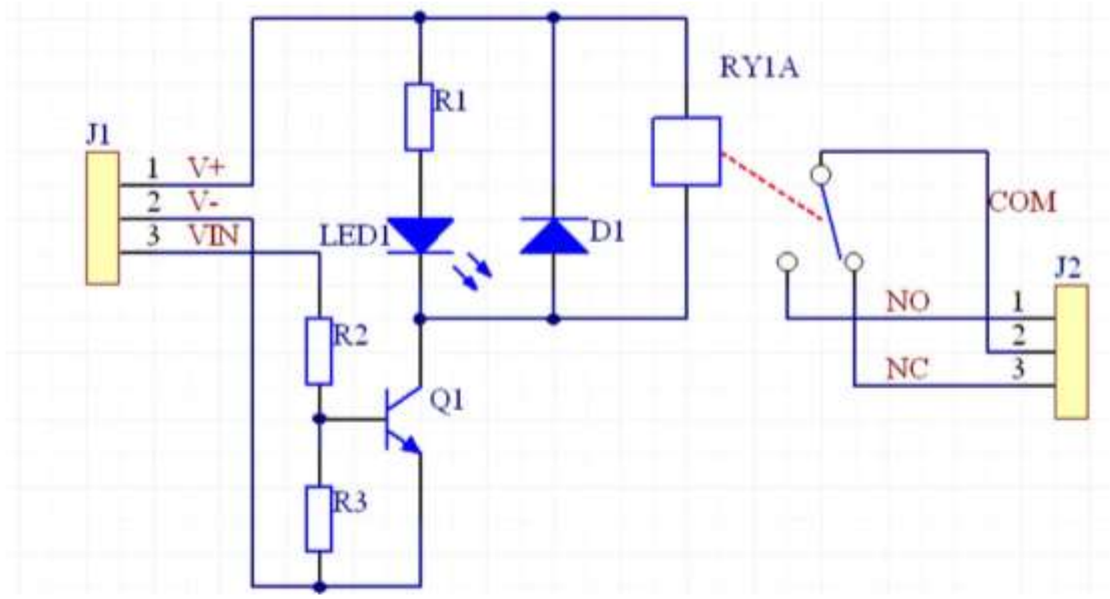
Relay 5V 1 ช่อง แบบ Active High บอร์ดรีเลย์ 5V สำหรับ Arduino ESP8266 ESP32

บอร์ดรีเลย์ 5V 1 ช่องทำงานแบบ Active High โมดูลใช้ไฟเลี้ยง 5V พร้อมไฟ LED แสดงผลการทำงาน สำหรับควบคุมสวิตช์ไฟ 250VAC 10A หรือไฟ 30VDC 10A

การต่อใช้งาน

- V+ : 5V
- V- : GND
- IN : สัญญาณควบคุมรีเลย์ เมื่อจ่ายไฟ 5V บอร์ดทำงาน

วงจรบอร์ด Relay 5V 1 ช่อง

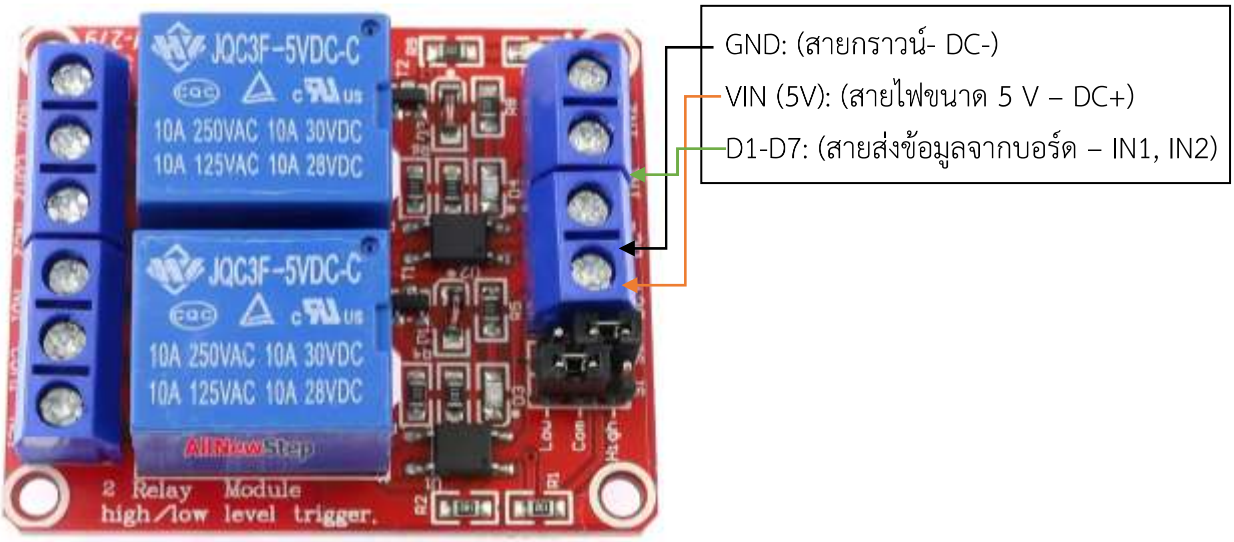


4. บอร์ด Relay 2 ช่อง 5V relay 5v แบบ Active High/Low 10A 250V สำหรับ Arduino และ Microcontroller

บอร์ด Relay 2 ช่อง 5 โวลต์แบบ Active Hight 10A 250V สำหรับ Arduino และ Microcontroller

Active High

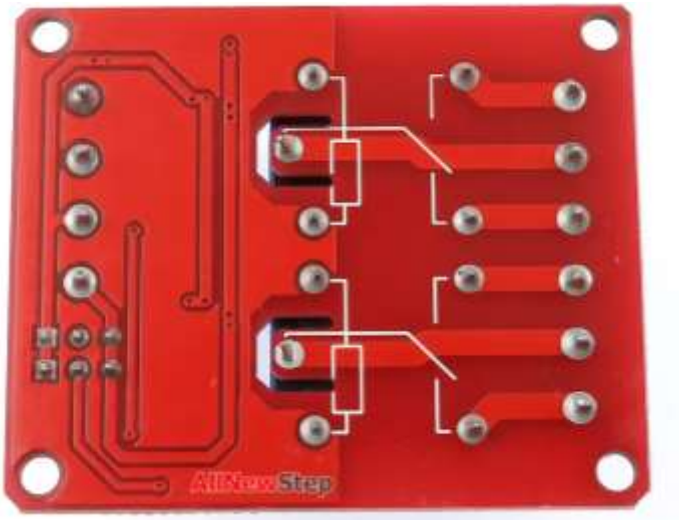
วงจรเป็นแบบ แยกกราวด์ Opto isolated Relay แยกการควบคุมของ mcu กับ ไฟแรงดันสูงออกจากกัน จึงปลอดภัยกับอุปกรณ์ที่ควบคุม เช่น arduino
สามารถควบ คุมเปิด/ปิด รีเลย์ได้ 2 ช่อง ใช้ไฟเข้า 5 โวลต์ ส่งสัญญาณควบคุมแบบ Active High หรือ Active Low เลือกโหมดได้โดยการเซตจัมเปอร์ สามารถเลือกการทำงานได้แบบอิสระทั้ง 2 ช่อง



รายละเอียด บอร์ดรีเลย์ 2 ช่อง สำหรับ Arduino

การ เชื่อมต่อมาตรฐานที่สามารถใช้ควบคุมได้โดยตรงจากไมโครคอนโทรนเลอร์ เช่น Arduino , 8051, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic)

- ใช้ไฟฟ้าที่ 5 โวลต์
- สามารถ 5 โวลต์จากบอร์ด arduino ที่มีขา 5 โวลต์ได้
- ใช้ควบคุมไฟฟ้าแรงสูงได้ที่ DC30V 10A , AC250V 10A
- มีไฟบอกสถานะการทำงานของรีเลย์ทุกตัว
- เชื่อมต่อด้วยขั้วสกรู ทำให้ติดตั้งได้ง่ายและสะดวก
- ใช้กระแสขับ relay แต่ละตัวที่ 15-20 mA
- การส่งสัญญาณควบคุมรีเลย์เป็นแบบ Active High / Active Low

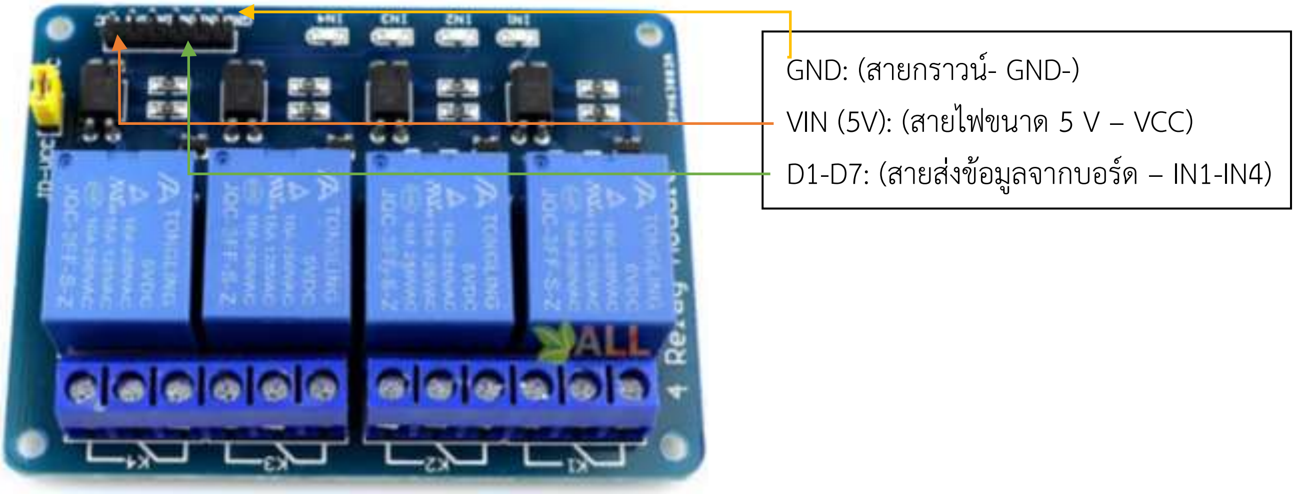


5. Relay Module 4 Channel 5V relay 4 ช่อง แบบ isolation control Relay Module Shield 250V/10A แบบ Active Low for Arduino

รายละเอียดสินค้า

4 Channel 5V relay isolation control Relay Module Shield 250V/10A

บอร์ดรีเลย์ 5 โวลต์ แบบแยกอิสระ Isolation Control 4 ช่อง แบบ Active LOW ทำงานเมื่อมีสัญญาณ 0 มาทริก พร้อมไฟ LED แสดงผลการทำงาน



รายละเอียด 4 Channel 5V relay isolation control Relay Module Shield 250V/10A

- 4-Channel Relay breakout
- Power supply range from 5V~7.5V
- Onboard Photocoupler isolation
- Equiped with high-current relay, AC250V 10A ; DC30V 10A.
- Relay Output Indicator LED
- TTL logic interface can be directly connected to microcontroller (Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430,etc)

6. กล่องใส่ของพลาสติก Electronics box กล่องเก็บอุปกรณ์อิเล็กทรอนิกส์ 225x150x60mm

7. สายไฟ จัมเปอร์ ผู้-ผู้ ยาว 10cm. จำนวน 40 เส้น

8. สายไฟ จัมเปอร์ ผู้-เมีย ยาว10cm. จำนวน 40 เส้น

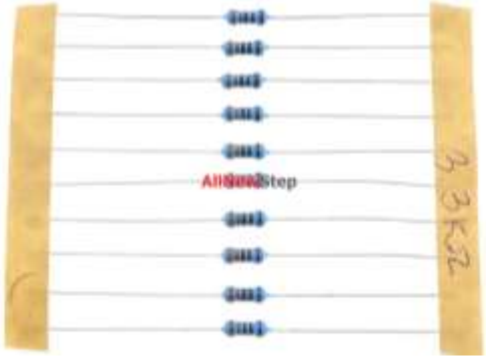
9. ตัวต้านทาน 10K 1/4W Metal film 1% 10 k จำนวน 10 ชิ้น

10. ตัวต้านทาน 100 K 1/4W Metal film 100k 1% จำนวน 10 ชิ้น

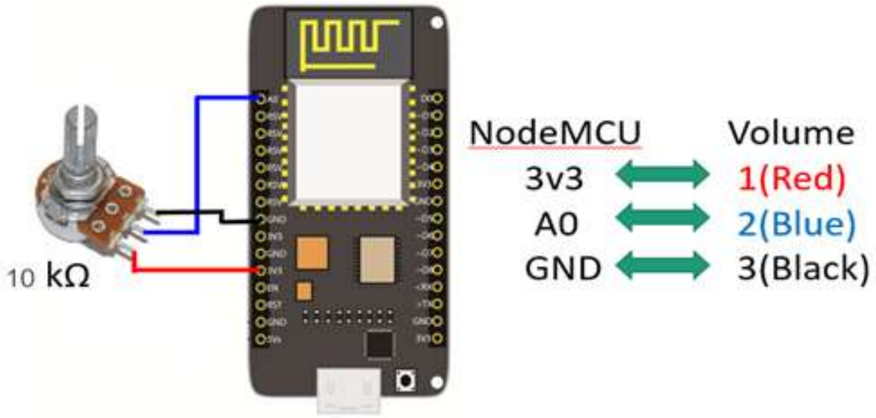
ตัวต้านทานแบบคงที่ 100k คลาดเคลื่อน 1 % จำนวน 10 ชิ้น

11. ตัวต้านทาน 10 โอห์ม 10r 1/4W Metal film 1% จำนวน 10 ชิ้น

12. ตัวต้านทาน 3.3 K 1/4W Metal film 1% จำนวน 10 ชิ้น 3.3k



13. ตัวต้านทานปรับค่าได้ 100k potentiometer R 100 k



14. สาย Micro USB อย่างดี ยาว 50cm จ่ายกระแส 3A สำหรับ Arduino ESP8266 ESP32 NodeMCU

15. USB Charger Doctor

USB Charger Doctor

วัดได้ทั้งกระแสไฟฟ้า(A) และแรงดันไฟฟ้า (V) ของอุปกรณ์ไฟฟ้าที่เสียบ USB เช่น โทรศัพท์มือถือ , อุปกรณ์คอมพิวเตอร์ , บอร์ด Arduino

รายละเอียด USB Charger Doctor

Size:2.05"(L)X0.79"(W)x0.59"(H)

Workable range: U=3~7.5V I=0~2.5A

Resolution: Voltage10mV ;Current 1mA

Error range: Voltage <±1% Current<±2%

Full-scale voltage drop: 200mV

Work temperature:0-60℃

Simple and convenient to use.

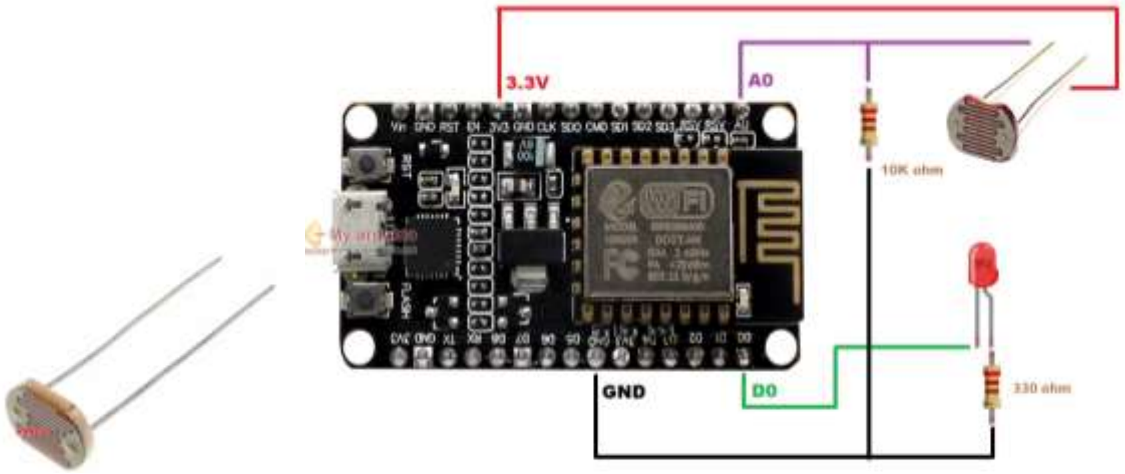
Portable and easy to carry.

The voltage and current will be displayed circularly.

plug and play, it can work without additional power supply.

The USB tester can test voltage and current of the power supply equipment with USB port .

16. LDR Photoresistor 10mm ldr 10mm



คำสั่ง Analog Read

val = analogRead(pin)

pin หมายถึง ขา analog ของ NodeMCU ESP8266 ที่ต้องการจะอ่าน

val หมายถึง ตัวแปร int สำหรับเก็บค่า สัญญาณ analog มีค่า 0-1023 ที่ขา pin

ตัวอย่างคำสั่ง Analog Read

ต้องการอ่านค่า ขา Analog A0 เก็บในตัวแปล Val

val = analogRead(A0)

LDR.ino

```
int ledPin = D0;
int analogPin = A0; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขาA0
int val = 0;
void setup() {
  pinMode(ledPin, OUTPUT);  // sets the pin as output
  Serial.begin(9600);
}
void loop() {
  val = analogRead(analogPin);  //อ่านค่าสัญญาณ analog ขาA0 ที่ต่อกับ LDR
```

```
  Serial.print("val = "); // พิมพ์ข้อมความส่งเข้าคอมพิวเตอร์ "val = "
  Serial.println(val); // พิมพ์ค่าของตัวแปร val
  if (val < 500) { // ค่า 100 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
    digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
  }
  else {
    digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
  }
  delay(100);
}
```

17. LDR 20mm Photocell Photoresistor ตัวต้านทานปรับค่าได้ตามแสง LDR ขนาด 20MM

analog gray sensor line finder sensor tracking module electronic building block electronics เซนเซอร์วัดแสง LDR+LED

เซนเซอร์วัดความเข้มแสงที่ตามองเห็น ภายในโมดูลเป็น LDR และใช้แสงนำทางเป็น LED สามารถอ่านค่าความเข้มแสงของวัตถุที่แสง LED ตกกระทบและสะท้อนกลับมา เป็นค่า 0-1023 แบบ Analog
ใช้สำหรับหาวัตถุ หรือตรวจจับเส้นขาว/ดำ

A1429.ino

```
/*
  Example By ArduinoAll.com
  OUT -> A0            VCC -> 3.3-5V            GND -> GND
*/
void setup() {
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}
void loop() {
  Serial.print("Light sensor value : ");
  int val = analogRead(A0);
  Serial.println(val);
  delay(500);
}
```

Specifications

Working voltage: 3.3V or 5V

Working current: < 20mA

Operating temperature range: -10 ° C ~ +70 ° C

Detection resolution: 10%

Interface type: analog signal output

Size: 24mm x 21mm

Weight: 3g

Interface definition

OUT: signal output

VCC: Power Supply (VCC)

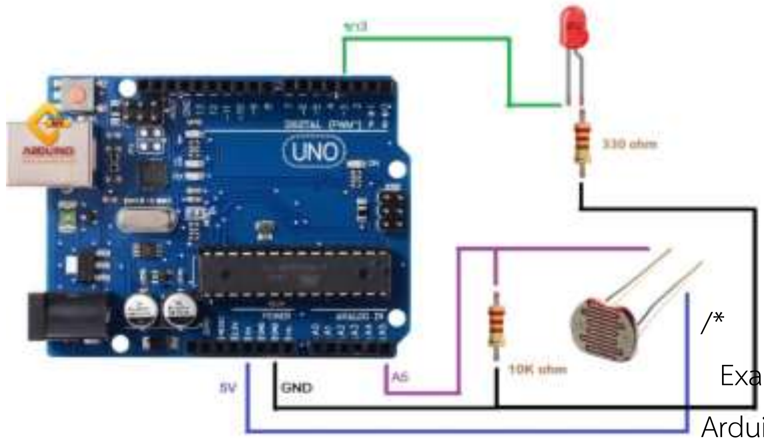
GND: Ground (GND)

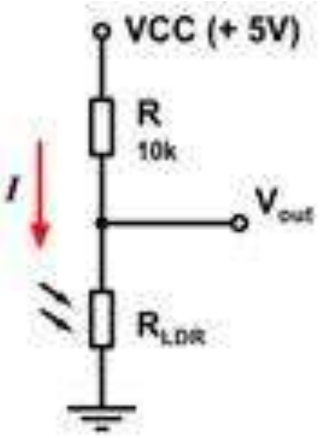
18. LDR 20mm Photocell Photoresistor ตัวต้านทานปรับค่าได้ตามแสง LDR ขนาด 20MM

ตัวต้านทานปรับค่าได้ตามแสง สำหรับวัดความเข้มของแสงแล้วเปลี่ยนเป็นความต้านทาน ถ้าแสงมากความต้านทานจะน้อย แสงน้อยความต้านทานจะมากขึ้น ขนาดใหญ่ 20mm

ข้อมูล LDR 20mm Photocell Photoresistor ตัวต้านทานปรับค่าได้ตามแสง LDR ขนาด 20MM

Coated with epoxy Good reliability High sensitivity Fast response Good spectrum characteristic Size: 20mm (50-paise size)





ตัวอย่างการต่อใช้งาน A1429.ino

```
// Read LDR by NodeMCU ESP8266
// Result is sent to serial port.
// Pin A0 = analog input
/*
 Example By
ArduinoAll.com
 OUT -> A0
 VCC -> 3.3-5V
 GND -> GND
*/
```

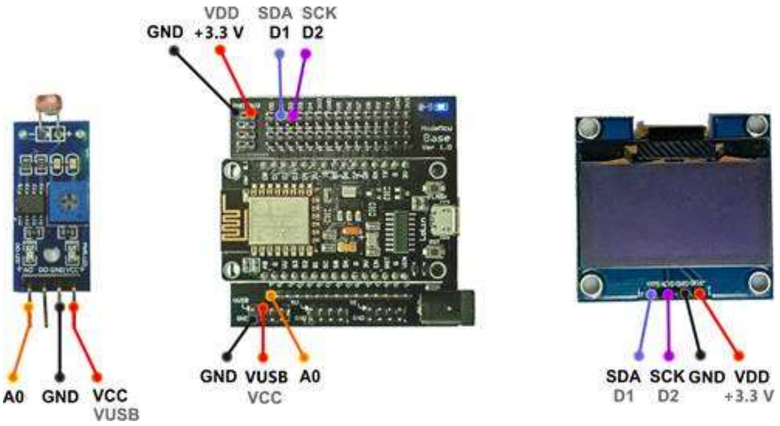
```
int LDR_Sensor = A0;
int data_LDR;
void setup() {
    Serial.begin(9600);    // Set serial speed to 9600 bps
}
void loop() {
    data_LDR = analogRead(LDR_Sensor);    // Read Analog Data
    Serial.println(data_LDR);
    delay(500);
}
```

สัญญาณ Analog InPut ตัวต้านทานปรับค่าได้ตามแสง LDR ควบคุมเปิดปิดหลอดไฟ LED

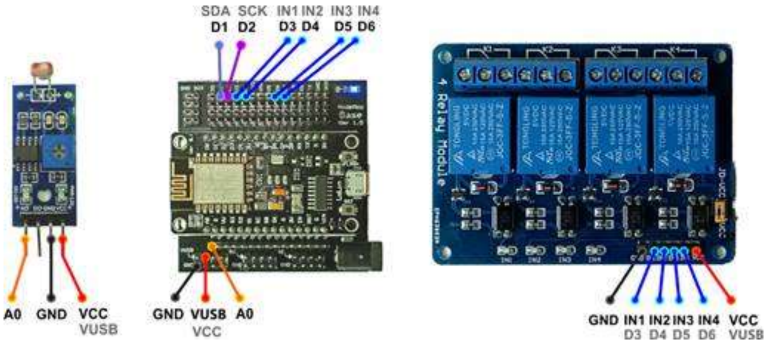
```
int ledPin = 3;
int analogPin = 5; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขาที่5
int val = 0;
void setup() {
  pinMode(ledPin, OUTPUT);  // sets the pin as output
  Serial.begin(9600);
}
void loop() {
  val = analogRead(analogPin);  //อ่านค่าสัญญาณ analog ขา5 ที่ต่อกับ LDR
  Serial.print("val = "); // พิมพ์ข้อมความส่งเข้าคอมพิวเตอร์ "val = "
  Serial.println(val); // พิมพ์ค่าของตัวแปร val
  if (val < 100) { // ค่า 100 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
    digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
  }
  else {
    digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
  }
  delay(100);
}
```

การเชื่อมต่อ NodeMCU v.3 กับ โมดูลวัดแสงแบบแอนะล็อกและจอแสดงผล OLED Display

```
// Read LDR by NodeMCU ESP8266
// result is sent to OLED & Serial
#include <Wire.h>
#include <SH1106.h>
```

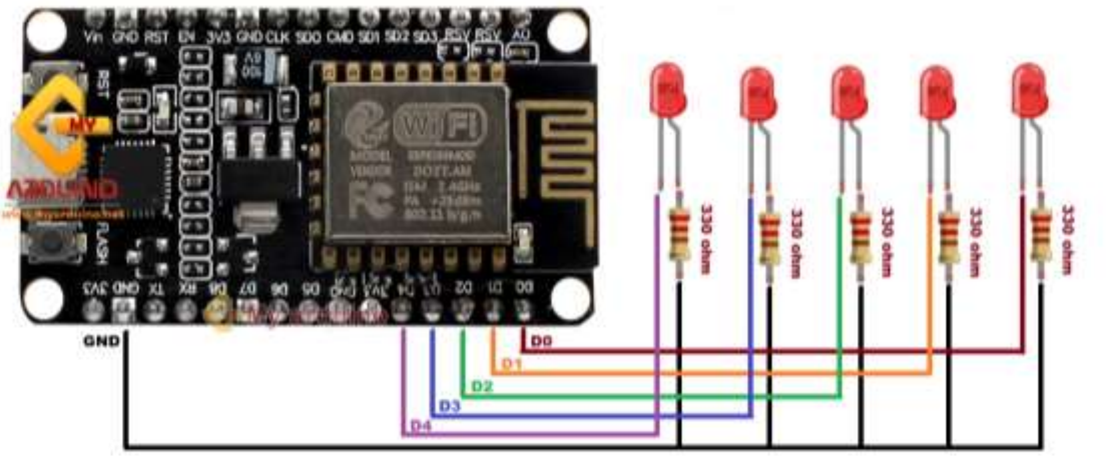


```
// SH1106(Addr, SDA, SCL)
SH1106 display(0x3C, D1, D2);  // OLED
// Pin A0 = analog input
int LDR_Sensor = A0;
int data_LDR;
float volt_LDR;
float illu_LDR;
void setup() {
    display.init();
    // Set serial speed to 9600 bps
    Serial.begin(9600);
}
void Calculate() {
    data_LDR = analogRead(LDR_Sensor);
    volt_LDR = (3.3/1024)*data_LDR;
    illu_LDR = 42.0 * pow(volt_LDR, -3.15);
}
void show_OLED() {
    display.clear();
    display.drawRect(0, 0, 128, 64);
    display.setFont(ArialMT_Plain_10);
    display.drawString(15,0, "LDR Sensor Monitor");
    display.drawString(5, 15, "data_LDR = ");
    display.drawString(65, 15, String(data_LDR));
    display.drawString(5, 30, "volt_LDR = ");
    display.drawString(65, 30, String(volt_LDR) + " V");
    display.drawString(5, 45, "illu_LDR = ");
    display.drawString(65, 45, String(illu_LDR) + " lux");
     display.display();
}
void show_Serial() {
    Serial.print("\t data_LDR = ");
    Serial.print(data_LDR);
    Serial.print("\t volt_LDR = ");
    Serial.print(volt_LDR);
```



```
    Serial.print("\t illu_LDR = ");
    Serial.println(illu_LDR);
}
void loop() {
    Calculate();
    show_OLED();
    show_Serial();
    delay(500);
}
```

19. LED 5mm



ตัวอย่างการต่อ LED หลายหลอด

```
int led1 = D0;              int led2 = D1;
int led3 = D2;              int led4 = D3;          int led5 = D4;
void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  Serial.begin(9600);
}
void loop() {
```

```
  for (int x = 0; x < 1023; x = x++) {
    analogWrite(led1, x);
    analogWrite(led2, x);
    analogWrite(led3, x);
    analogWrite(led4, x);
    analogWrite(led5, x);
    delay(10);
    Serial.println(x);
  }
  delay(500);
  for (int x = 1023; x > 0; x = x--) {
    analogWrite(led1, x);
    analogWrite(led2, x);
    analogWrite(led3, x);
    analogWrite(led4, x);
    analogWrite(led5, x);
    delay(10);
    Serial.println(x);
  }
  delay(500);
}
```

20. LED traffic light module 5V traffic light module

LED traffic light module 5V traffic light module

Color: red yellow green

LED: 8mm x3

Brightness: normal brightness

Voltage: 5V

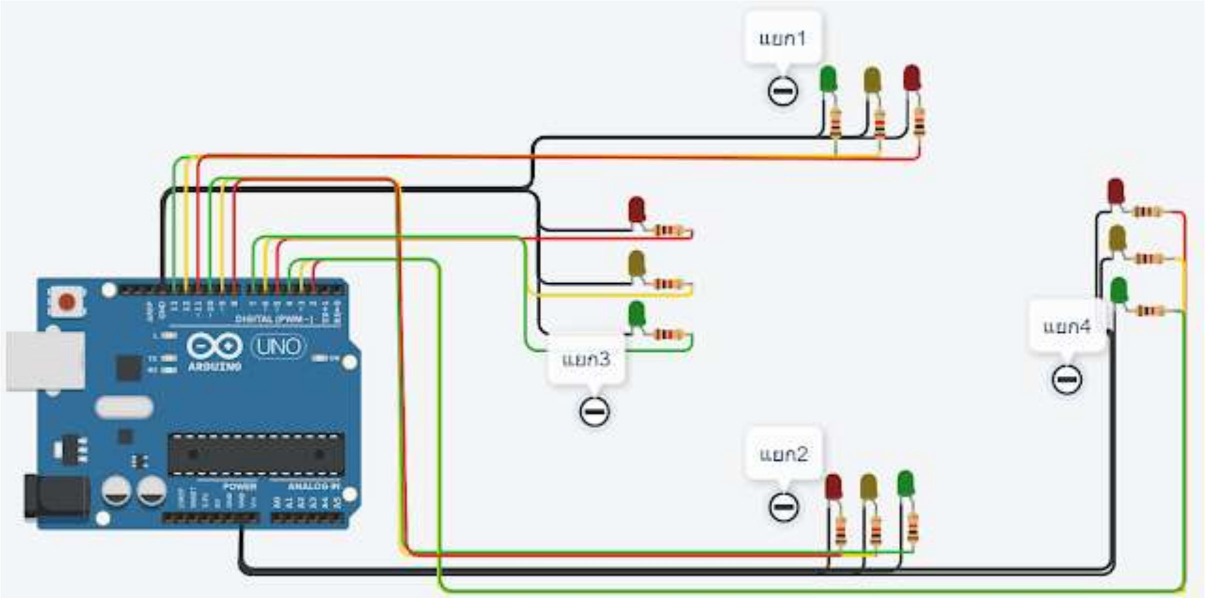
Size: 56x21x11mm

Input: digital signal output

Interface: common cathode red yellow green separate controlThe green light is darker, the yellow light is brighter than the green light, and the red light is the brightest.

```
// ตัวอย่างโคดโดย www.arduinoall.com
// ไม่สงวนลิขสิทธิ์สำหรับแบ่งปันความรู้
// สงวนลิขสิทธิ์ ห้ามนำไปใช้เชิงพานิชย์ ควรมีความคิดตั้งใจ สร้างผลงานด้วยตัวเองบ้าง  ก็อปปี้ผลงานคนอื่น
= ขโมย
void setup() {
  //Example By ArduinoAll.com
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}
void loop() {
  ledOn(11);
  delay(1000);
  ledOn(12);
  delay(1000);
  ledOn(13);
  delay(1000);
}
void ledOn(int pin){
  digitalWrite(13,0);
  digitalWrite(12,0);
  digitalWrite(11,0);
  digitalWrite(pin,1);
}
```

ขั้นตอนการ ต่อวงจรไฟจราจร



//ส่วนแรก กำหนดตัวแปร และตั้งค่า

```
int y1g = GPIO9;
int y1y =GPIO1;
int y1r = GPIO3;
int y2g = D8;
int y2y = D7;
int y2r = D6;
int y3g = D5;
int y3y = D4;
int y3r = D3;
int y4g = D2;
int y4y = D1;
int y4r = D0;

void setup()
{
  pinMode(y1g, OUTPUT);
  pinMode(y1y, OUTPUT);
  pinMode(y1r, OUTPUT);
  pinMode(y2g, OUTPUT);
  pinMode(y2y, OUTPUT);
  pinMode(y2r, OUTPUT);
  pinMode(y3g, OUTPUT);
  pinMode(y3y, OUTPUT);
```

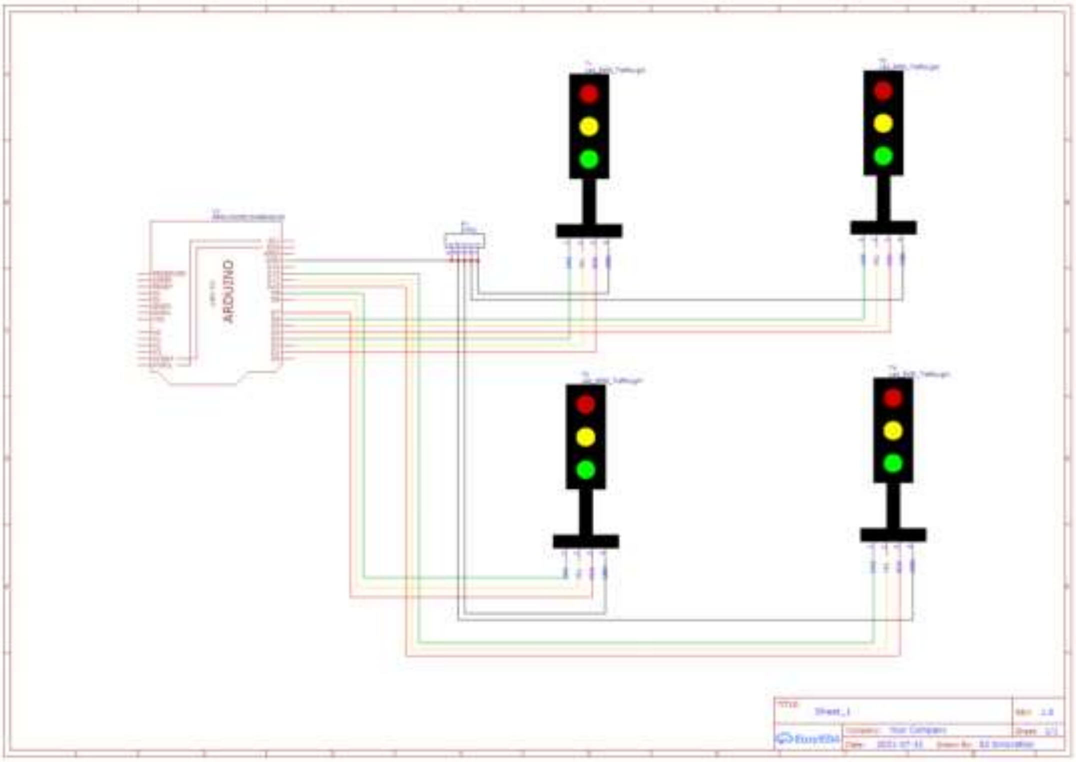
```
  pinMode(y3r, OUTPUT);
  pinMode(y4g, OUTPUT);
  pinMode(y4y, OUTPUT);
  pinMode(y4g, OUTPUT);
}
```

//ส่วนต่อมา กำหนดโปรแกรม

```
void loop()
{
  //รอบแรก ไฟเขียว แยก 1-2
  //แยก 1 เขียว
  digitalWrite(y1g, HIGH);
  digitalWrite(y1y, LOW);
  digitalWrite(y1r, LOW);
  //แยก 2 เขียว
  digitalWrite(y2g, HIGH);
  digitalWrite(y2y, LOW);
  digitalWrite(y2r, LOW);
  //แยก 3 แดง
  digitalWrite(y3g, LOW);
  digitalWrite(y3y, LOW);
  digitalWrite(y3r, HIGH);
  //แยก 4 แดง
  digitalWrite(y4g, LOW);
  digitalWrite(y4y, LOW);
  digitalWrite(y4r, HIGH);
    delay(3000); // Wait for 1000 millisecond(s)
//ไฟกระพริบ แยก 1-2
  //กระพริบ แยก 1-2 เปิด
  digitalWrite(y1g, LOW);
  digitalWrite(y1y, HIGH);
  digitalWrite(y2g, LOW);
  digitalWrite(y2y, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
   //กระพริบ แยก 1-2 ปิด
  digitalWrite(y1y, LOW);
```

```
  digitalWrite(y2y, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
   //กระพริบ แยก 1-2
  digitalWrite(y1y, HIGH);
  digitalWrite(y2y, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    //กระพริบ แยก 1-2
  digitalWrite(y1y, LOW);
  digitalWrite(y2y, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
  //เหลืออะไรอีกน้อ
  //ไฟแดง แยก 1-2  ไฟเขียวแยก 3-4
  //ไฟกระพริบ แยก 3-4
}
```

อีกตัวอย่างการต่อไปจราจร 2 เสา

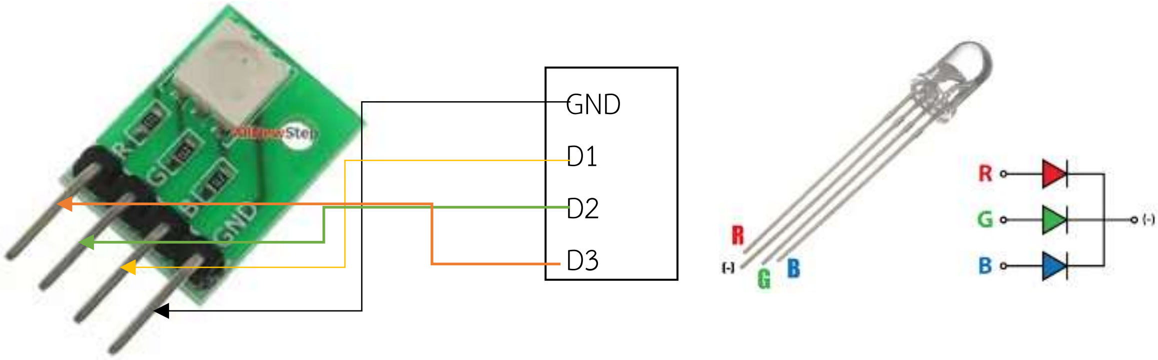



```
// กำหนดไฟเลน 1
int RED1 = D1; // ขา Digital 1 เป็น  ไฟแดงเลน 1
int YEL1 = D2; // ขา Digital 2 เป็น  ไฟเหลืองเลน 1
int GRN1 = D3; // ขา Digital 3 เป็น  ไฟเขียวเลน 1

// กำหนดไฟเลน 2
```

```
int RED2 = D4; // ขา Digital 4 เป็น  ไฟแดงเลน 2
int YEL2 = D5; // ขา Digital 5 เป็น  ไฟเหลืองเลน 2
int GRN2 = D6; // ขา Digital 6 เป็น  ไฟเขียวเลน 2

void setup(){
    pinMode(RED1, OUTPUT);
    pinMode(YEL1, OUTPUT);
    pinMode(GRN1, OUTPUT);
    pinMode(RED2, OUTPUT);
    pinMode(YEL2, OUTPUT);
    pinMode(GRN2, OUTPUT);
}

void loop(){
    // ให้สัญญาณไฟเหลือง เลน 1 ก่อน
    digitalWrite(YEL1, HIGH); // ให้สัญญาณไฟเหลือง เลน 1
    digitalWrite(GRN1, LOW);
    delay(2500); //2.5วินาที

    // ปิดไฟเหลืองสองฝั่ง, จากนั้นให้สัญญาณไฟแดงเลน 1, กับให้สัญญาณไฟเขียวเลน 2 พร้อมกัน
    digitalWrite(RED1, HIGH); //ให้สัญญาณไฟแดง เลน 1
    digitalWrite(GRN2, HIGH); //ให้สัญญาณไฟเขียว เลน 2
    digitalWrite(YEL1, LOW);
    digitalWrite(YEL2, LOW);
    digitalWrite(RED2, LOW);
    delay(15000); //15วินาที

    // ให้สัญญาณไฟเหลือง เลน 2
    digitalWrite(YEL2, HIGH); // ให้สัญญาณไฟเหลือง เลน 2
    digitalWrite(GRN2, LOW);
    delay(2500); //2.5วินาที

    // ปิดไฟเหลืองสองฝั่ง, จากนั้นให้สัญญาณไฟแดงเลน 2,กับให้สัญญาณไฟเขียวเลน 1 พร้อมกัน
    digitalWrite(RED2, HIGH); //ให้สัญญาณไฟแดง เลน 2
    digitalWrite(GRN1, HIGH); //ให้สัญญาณไฟเขียว เลน 1
```

```
    digitalWrite(YEL1, LOW);
    digitalWrite(RED1, LOW);
    digitalWrite(YEL2, LOW);
    delay(15000); //15วินาที
}
```

21. RGB LED module full color LED module โมดูลไฟ LED RGB



RGB LED module full color LED module โมดูลไฟ LED RGB
โมดูลหลอดไฟ LED 3 สี สามารถสั่งแสดงผลแต่ละสี แยกได้อิสระ สั่งโปรแกรมให้ผสมสี เป็นแสงสีต่าง ๆ ได้ โมดูล LED ขนาดเล็กเสียบกับบอร์ด Arduino เพื่อทดลองได้ทันที ไม่ต้องต่อสายไฟเพิ่ม

RGB full color LED module Module parameters

Use 5050 full color super bright LED

With current limiting resistor to prevent burning LED

Can be connected to various single-chip

High level Bright LED

Working voltage: 3.3V/5V

Weight: 4g

PCB size: 15mmx10.6mm

PCB color: green

Can be directly plugged into the Arduino motherboard, no DuPont line connection required

โค้ดตัวอย่างผสมสีหลอดไฟ LED RGB

```
int R = D1;
int G = D2;
int B = D3;
void setup() {
pinMode(R, OUTPUT);
pinMode(G, OUTPUT);
```

```
pinMode(B, OUTPUT);
Serial.begin(9600);
}
void loop() {
// ปิดไฟทั้งหมด
analogWrite(R, 0);
analogWrite(G, 0);
analogWrite(B, 0);
delay(500);
// เปิดไฟสีแดง
analogWrite(R, 255);
analogWrite(G, 0);
analogWrite(B, 0);
delay(500);
// เปิดไฟสีเขียว
analogWrite(R, 0);
analogWrite(G, 0255);
analogWrite(B, 0);
delay(500);
// เปิดไฟสีน้ำเงิน
analogWrite(R, 0);
analogWrite(G, 0);
analogWrite(B, 255);
delay(500);
// เปิดไฟทุกสี จะได้แสงสีขาว
analogWrite(R, 255);
analogWrite(G, 255);
analogWrite(B, 255);
delay(500);
}
```

หรืออีกตัวอย่างเป็นโปรแกรมที่ทำให้หลอดไฟ LED แต่ละสีค่อย ๆ ติดและดับ

```
#define R_pin D0 // Red GPIO16
#define G_pin D1 // Green GPIO5
#define B_pin D2 // Blue GPIO4
int start_R = 255; // Initialize red as 255 (maximum)
```

```
int start_G = 0; // Initialize green as zero
int start_B = 0; // Initialize blue as zero
void setup() {
  pinMode(R_pin, OUTPUT); // Initialize the R_pin as an output
  pinMode(G_pin, OUTPUT); // Initialize the G_pin as an output
  pinMode(B_pin, OUTPUT); // Initialize the G_pin as an output;
}
void loop() {
  for (int i = 0; i < 256; i++)               // Set loop
  {
    start_B = 0;                              // Clear BLUE data
    start_G++;                                // Increase GREEN data
    displayRGB(start_R, start_G, start_B);    // Drive LED with PWM
    delay(30);                                // Short delay
  }
  for (int i = 0; i < 256; i++)
  {
    start_R--;                                // Decrease RED data
    displayRGB(start_R, start_G, start_B);    // Drive LED with PWM
    delay(30);                                // Short delay
  }
  for (int i = 0; i < 256; i++)
  {
    start_R = 0;                              // Clear RED data
    start_B++;                                // Increase BLUE data
    displayRGB(start_R, start_G, start_B);    // Drive LED with PWM
    delay(30);                                // Short delay
  }
  for (int i = 0; i < 256; i++)
  {
    start_G--;                                // Decrease GREEN data
    displayRGB(start_R, start_G, start_B);    // Drive LED with PWM
    delay(30);                                // Short delay
  }
  for (int i = 0; i < 256; i++)
```

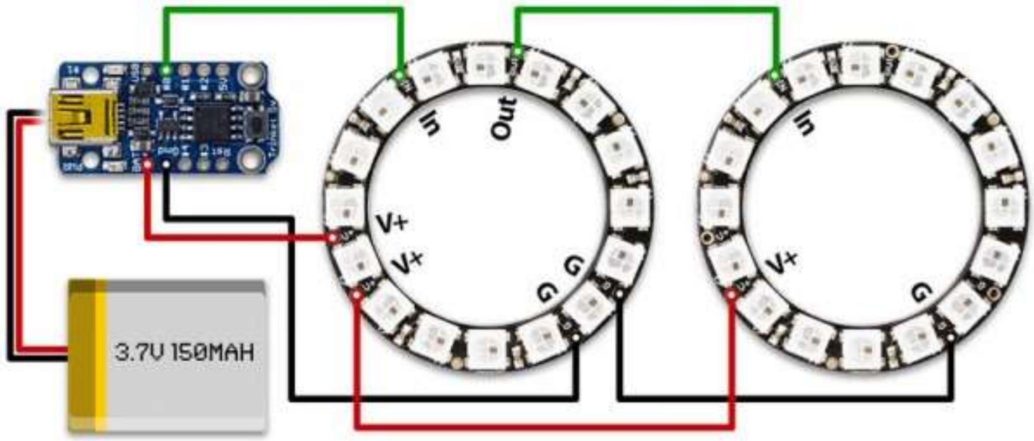
```
  {
    start_G = 0;                        // Clear GREEN data
    start_R++;                          // Increase RED data
    displayRGB(start_R, start_G, start_B);    // Drive LED with PWM
    delay(30);                          // Short delay
  }
  for (int i = 0; i < 256; i++)
  {
    start_B--;                          // Decrease BLUE data
    displayRGB(start_R, start_G, start_B);    // Drive LED with PWM
    delay(30);                          // Short delay
  }
}
void displayRGB(int R, int G, int B)        // Drive LED with PWM by using analogWrite
function
{
  analogWrite(B_pin, B);
  analogWrite(G_pin, G);
  analogWrite(R_pin, R);
}
```

22. LED เส้นผ่านศูนย์กลาง 68mm

NeoPixel Ring 12 WS2812 RGB LED

วงแหวน LED RGB NeoPixel Ring 12 ดวง ใช้ไฟเลี้ยง 5VDC ต่อสายสัญญาณเพียง 1 เส้น ควบคุม LED แต่ละดวงได้อย่างอิสระ LED มี 3 สี RGB ในดวงเดียว สามารถกำหนดว่าจะให้ดวงไหนสีอะไรก็ได้ตามต้องการ

No.	Symbol	PIN	Function Description
1	+5V	Power	5V power supply
2	DIN	Data Input	Input the control signal
3	GND	Ground	Earthing
4	DOUT	Data Output	Output the control signal, and connect to next panel's DIN



Sketch di simulazione

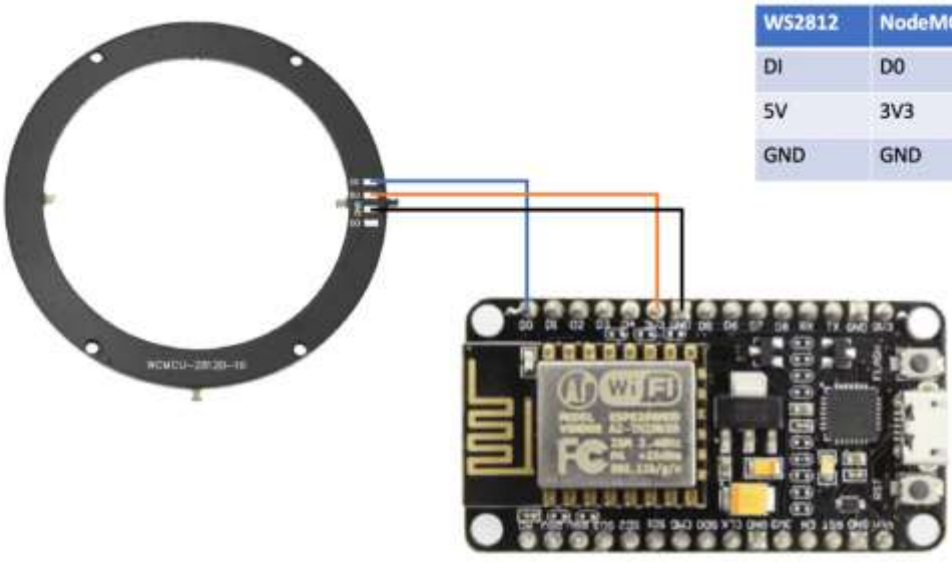
```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif
 #define PIN   6
#define NUMPIXELS    72
 Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
 int delayval = 1000; // delay for half a second
int colorH[] = {150,0,0};
int colorM[] = {0,150,0};
int colorS[] = {0,0,150};
byte H = 8;
byte M = 59;
byte S = 0;

void setup() {
  pixels.begin(); // This initializes the NeoPixel library.
  Serial.begin(9600);
  for ( int i=0; i<NUMPIXELS; i++ ) { pixels.setPixelColor(i, pixels.Color(0,0,0)); pixels.show();
delay( 5 ); }
}

void loop() {
  if ( S > 59 ) {
    pixels.setPixelColor(M, pixels.Color(0,0,0));  // prev min
    pixels.setPixelColor(59, pixels.Color(0,0,0)); // last second
```

```
    M++;
    S = 0;
  }
  if ( M > 59 ) {
    pixels.setPixelColor((60+H), pixels.Color(0,0,0)); // prev hour
    pixels.setPixelColor(59, pixels.Color(0,0,0)); // last second, min
    H++;
    M = 0;
    S = 0;
  }
  if ( H > 12 ) {
    pixels.setPixelColor((60+H), pixels.Color(0,0,0)); // last hour
    pixels.setPixelColor(59, pixels.Color(0,0,0));     // last second, min
    H = 0;
    M = 0;
    S = 0;
  }

  pixels.setPixelColor((S-1), pixels.Color(0,0,0));
  pixels.setPixelColor(S, pixels.Color(colorS[0],colorS[1],colorS[2]));
  pixels.setPixelColor(M, pixels.Color(colorM[0],colorM[1],colorM[2]));
  pixels.setPixelColor((60+H), pixels.Color(colorH[0],colorH[1],colorH[2]));
  pixels.show();
     delay( delayval );
  S++;
}
```

23. NeoPixel Ring 8 WS2812 RGB LED เส้นผ่านศูนย์กลาง 32mm

NeoPixel Ring 8 WS2812 RGB LED

วงแหวน LED RGB NeoPixel Ring 8 ดวง ใช้ไฟเลี้ยง 5VDC ต่อสายสัญญาณเพียง 1 เส้น ควบคุม LED แต่ละดวงได้อย่างอิสระ LED มี 3 สี RGB ในดวงเดียว สามารถกำหนดว่าจะให้ดวงไหนสีอะไรก็ได้ตามต้องการ



WS2812	NodeMCU
DI	D0
5V	3V3
GND	GND

Install FastLED Library

เข้าไปที่เมนู Sketch–> Include Library –> Manage Libraries..

จากนั้นก็ Search คำว่า FastLED ก็จะเจอ Library ตัวนี้แหละครับ แล้วก็ Install ได้เลย

เริ่มต้นใช้งาน Neopixel ws2812

มาดูกันที่ตัวอย่างแรกของ FastLED เลยครับกับ Blink โดยสามารถเปิดจากไฟล์ตัวอย่างกันได้เลย

ตัวอย่างที่ 1

```
#include "FastLED.h"
// How many leds in your strip?
#define NUM_LEDS 16 // neopixel ring ที่ผมใช้มีทั้งหมด 16 ดวงด้วยกัน
#define DATA_PIN D1 //ขาที่ต่อเป็น D1 เป็น GPIO5 แต่เวลาใช้งาน ให้ใช้ 1 แทน ถ้าเป็น D4 ก็ใช้ 4
// Define the array of leds
CRGB leds[NUM_LEDS];
void setup() {
   FastLED.addLeds<WS2812, DATA_PIN, RGB>(leds, NUM_LEDS);
     }
void loop() {
  // Turn the LED on, then pause
  leds[0] = CRGB::Red;
  FastLED.show();
  delay(500);
    // Now turn the LED off, then pause
```

```
  leds[0] = CRGB::Black;
  FastLED.show();
  delay(500);
}
```

ตัวอย่างที่ 2 เราก็จะเปลี่ยนในส่วนของบรรทัดนี้กัน จาก RGB เป็น GRB คราวนี้ แดงเป็นแดง เขียวเป็นเขียว เรียบร้อย

```
FastLED.addLeds<WS2812, DATA_PIN, GRB>(leds, NUM_LEDS);
```

การกำหนดสีของไฟ LED ผ่าน FastLED Library

ในส่วนของสีของ LED นั้นสามารถกำหนดได้ในแต่ละดวงตาม Array ของ LED ได้เลย โดยสามารถทำได้หลายแบบยกตัวอย่างเช่น

กำหนดตามสีของ Web/HTML Color Code

```
leds[i] = CRGB::HotPink;
```

กำหนดตามสีแยกกันแบบ RGB

```
leds[i].setRGB( 255, 68, 221);
```

กำหนดสีตาม Hex Color Code

```
 leds[i] = 0xFF44DD;
```

ใครถนัดใช้แบบไหนในการกำหนดสีก็เลือกตามที่สะดวกและตามการใช้งานเลยครับ คราวนี้เรามาลองใช้งานทำไฟวิ่งกันหลายๆแบบดูบ้าง ตาม Code ข้างล่างนี้ครับ

```
#include "FastLED.h"
// How many leds in your strip?
#define NUM_LEDS 16 // neopixel ring ที่ผมใช้มีทั้งหมด 16 ดวงด้วยกัน
#define DATA_PIN D1 //ขาที่ต่อเป็น D1 เป็น GPIO5 แต่เวลาใช้งาน ให้ใช้ 1 แทน ถ้าเป็น D4 ก็ใช้ 4
// Define the array of leds
CRGBArray<NUM_LEDS> leds;
void setup() {
  FastLED.addLeds<WS2812, DATA_PIN, GRB>(leds, NUM_LEDS);
}
void loop() {
  // Turn the LED on, then pause
  for (int i = 0; i <= 15; i++)
  {
    leds[i] = CRGB::Gray;
    FastLED.show();
    delay(100);
    leds[i] = CRGB::Black;
    FastLED.show();
  }
  for (int i = 15; i >= 0; i--)
```

```
  {
    leds[i] = CRGB::Gray;
    FastLED.show();
    delay(100);
    leds[i] = CRGB::Black;
    FastLED.show();
  }
  for (int i = 0; i <= 15; i++)
  {
    leds[i] = CRGB(i,i,i);
    FastLED.show();
    delay(100);
  }
  for (int i = 0; i <= 32; i++)
  {
    leds = CRGB(i,0,0);
    FastLED.show();
    delay(100);
  }
  for (int i = 0; i <= 32; i++)
  {
    leds = CRGB(32-i,i,0);
    FastLED.show();
    delay(100);
  }
  for (int i = 0; i <= 32; i++)
  {
    leds = CRGB(0,32-i,i);
    FastLED.show();
    delay(100);
  }
}
```

ตัวอย่างที่ 3

Sketch_di_imulazione.ino

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h>
#endif
```

```
 #define PIN 6
#define NUMPIXELS 72
 Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
 int delayval = 1000; // delay for half a second
int colorH[] = {150,0,0};
int colorM[] = {0,150,0};
int colorS[] = {0,0,150};
byte H = 8;
byte M = 59;
byte S = 0;
 void setup() {
  pixels.begin(); // This initializes the NeoPixel library.
  Serial.begin(9600);
  for ( int i=0; i<NUMPIXELS; i++ ) { pixels.setPixelColor(i, pixels.Color(0,0,0)); pixels.show();
delay( 5 ); }
}
void loop() {
  if ( S > 59 ) {
    pixels.setPixelColor(M, pixels.Color(0,0,0));  // prev min
    pixels.setPixelColor(59, pixels.Color(0,0,0)); // last second
    M++;
    S = 0;
  }
  if ( M > 59 ) {
    pixels.setPixelColor((60+H), pixels.Color(0,0,0)); // prev hour
    pixels.setPixelColor(59, pixels.Color(0,0,0)); // last second, min
    H++;
    M = 0;
    S = 0;
  }
  if ( H > 12 ) {
    pixels.setPixelColor((60+H), pixels.Color(0,0,0)); // last hour
    pixels.setPixelColor(59, pixels.Color(0,0,0));     // last second, min
    H = 0;
    M = 0;
    S = 0;
  }
```

```
  pixels.setPixelColor((S-1), pixels.Color(0,0,0));
  pixels.setPixelColor(S, pixels.Color(colorS[0],colorS[1],colorS[2]));
  pixels.setPixelColor(M, pixels.Color(colorM[0],colorM[1],colorM[2]));
  pixels.setPixelColor((60+H), pixels.Color(colorH[0],colorH[1],colorH[2]));
  pixels.show();

  delay( delayval );
  S++;
}
```

24. จอแสดงผล OLED สีขาว 128x64 แบบ I2C ขนาด 1.3" สีขาว OLED Display I2C Module 1.3" for Arduino

จอแสดงผล OLED 128x64 แบบ I2C ขนาด 1.3" สีขาว OLED Display I2C Module 1.3" for Arduino

ตำแหน่งขา GND , VCC , SCL , SDA

จอแสดงผลแบบ OLED LCD สำหรับ Arduino หน้าจอ 128x64 ขนาด 1.3" เชื่อมต่อแบบ IIC ใช้ไฟได้ทั้ง 3.3V หรือ 5V ให้จอสว่างแสดงผลมองเห็นได้อย่างชัดเจน และประหยัดไฟ สามารถวาดภาพกราฟฟิกส์เป็นรูปต่าง ๆ หรือทำเป็นเมนูตามแบบที่ต้องการได้

ติดตั้งไลบารี Arduino OLED

ติดตั้งไลบารี 2 ตัวดังนี้ oled และ Adafruit_GFX

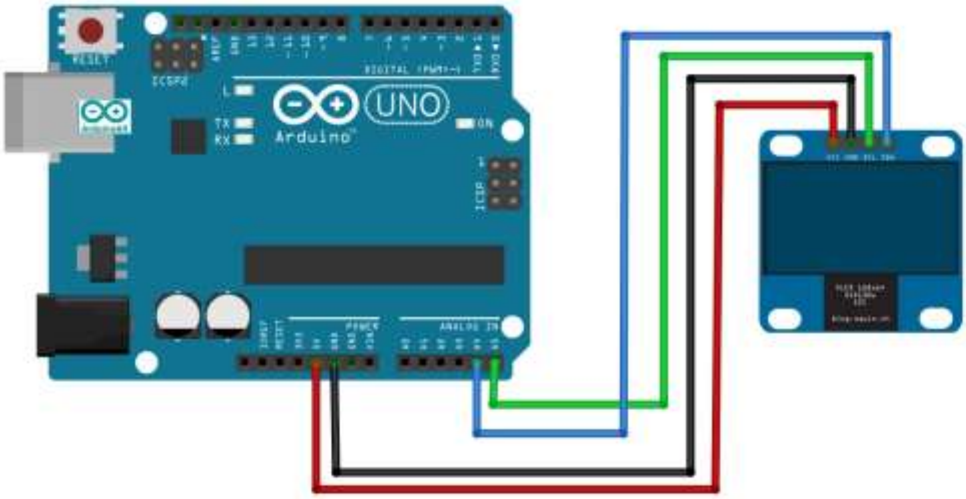
ต่อวงจร Arduino Uno กับจอ OLED ดังนี้

VCC - 5V

GND - GND

SDA - A4, D1

SCL - A5, D2

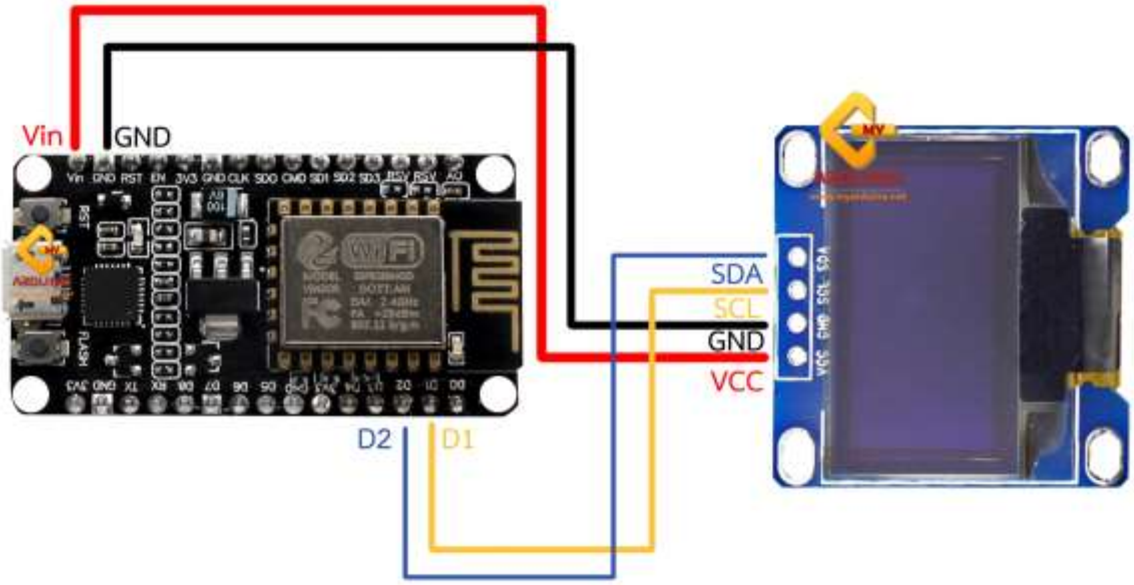


อัพโหลดโคดตัวอย่าง Arduino OLED นี้แล้วดูผลลัพธ์

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // pixel ความกว้าง
#define SCREEN_HEIGHT 64 // pixel ความสูง
 // กำหนดขาต่อ I2C กับจอ OLED
#define OLED_RESET -1 //ขา reset เป็น -1 ถ้าใช้ร่วมกับขา Arduino reset
Adafruit_SSD1306 OLED(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
 int var = 0; // ตัวแปรสำหรับทดสอบแสดงผล
void setup() {
Serial.begin(9600);
if (!OLED.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // สั่งให้จอ OLED เริ่มทำงานที่ Address 0x3C
Serial.println("SSD1306 allocation failed");
} else {
Serial.println("ArdinoAll OLED Start Work !!!");
}
}
 void loop() {
OLED.clearDisplay(); // ลบภาพในหน้าจอทั้งหมด
OLED.setTextColor(WHITE, BLACK); //กำหนดข้อความสีขาว ฉากหลังสีดำ
OLED.setCursor(0, 0); // กำหนดตำแหน่ง x,y ที่จะแสดงผล
OLED.setTextSize(2); // กำหนดขนาดตัวอักษร
OLED.println("OLED"); // แสดงผลข้อความ ALL
```

```
 OLED.setTextSize(1);
OLED.print("welcome to");
OLED.println(" All");
OLED.setTextColor(BLACK, WHITE); //กลับสีข้อความกับพื้นหลัง
OLED.print("www.arduinoall.com");
OLED.setTextColor(WHITE, BLACK);
OLED.setCursor(60, 0);
OLED.setTextSize(2);
OLED.println(var, DEC);
 OLED.setCursor(0, 40);
OLED.setTextSize(2);
OLED.println("ArduinoALL"); // แสดงผลข้อความ ALL
 OLED.display(); // สั่งให้จอแสดงผล
var++;
delay(500);
}
#include <Wire.h>
#include <ACROBOTIC_SSD1306.h>
void setup()
{
  Wire.begin();
  oled.init();                      // Initialze SSD1306 OLED display
  oled.clearDisplay();              // Clear screen
  oled.setTextXY(0,0);              // Set cursor position, start of line 0
  oled.putString("ACROBOTIC");
  oled.setTextXY(1,0);              // Set cursor position, start of line 1
  oled.putString("industries");
  oled.setTextXY(2,0);              // Set cursor position, start of line 2
  oled.putString("Pasadena,");
  oled.setTextXY(2,10);             // Set cursor position, line 2 10th character
  oled.putString("CA");
}
void loop()
{
}
```

NodeMCU ESP8266 จอแสดงผล OLED 128x64 แบบ I2C



1.เชื่อมต่ออุปกรณ์

NodeMCU ESP8266 -> จอแสดงผล OLED 128x64 แบบ I2C

- Vin->VCC
- GND->GND
- D1->SDA
- D2->SC

2. เมื่อเชื่อมต่ออุปกรณ์เรียบร้อยแล้ว ให้ทำการดาวน์โหลด Library ข้างล่าง แล้วแยกโฟลเดอร์ออกไปใส่ไว้ที่ C:\Users\.....\Documents\Arduino\libraries

3. จากนั้นทำการ copy โค๊ดด้านล่างไปใส่ไว้ในโปรแกรม Arduino IDE

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 16
Adafruit_SSD1306 display(OLED_RESET);
void setup() {
display.begin(SSD1306_SWITCHCAPVCC, 0x3c); //สั่งให้จอ OLED เริ่มทำงานที่ Address 0x3C
display.clearDisplay(); // ลบภาพในหน้าจอทั้งหมด
display.setTextSize(1); // กำหนดขนาดตัวอักษร
display.setTextColor(WHITE);
display.setCursor(0,0); // กำหนดตำแหน่ง x,y ที่จะแสดงผล
display.println(" OLED 0.96 TESTER ");
display.setCursor(0,10);
```

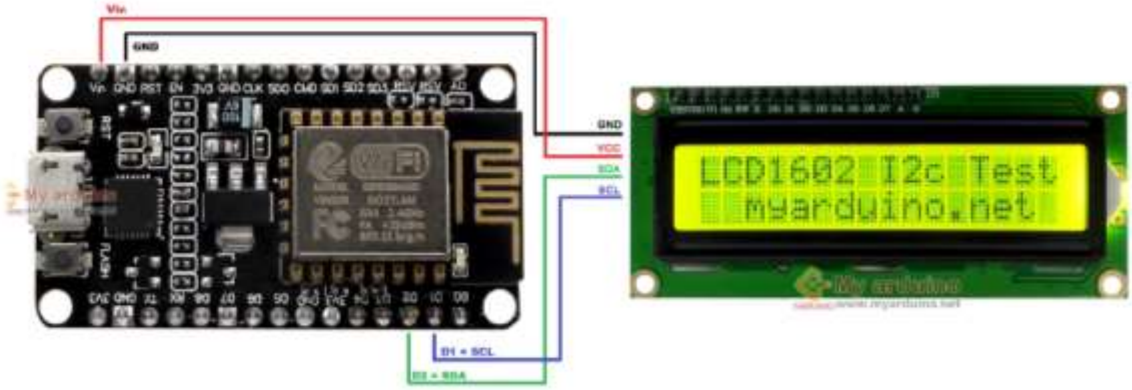
```
display.setTextSize(2);
display.setTextColor(BLACK, WHITE); //กำหนดข้อความสีขาว ฉากหลังสีดำ
display.println(" Myarduino");
display.setCursor(0,32);
display.setTextSize(1);
display.setTextColor(WHITE);
display.println("128 x 64 Pixels 0.96");
display.setCursor(0,48);
display.setTextSize(1);
display.setTextColor(WHITE);
display.println(" www.myarduino.net "); // แสดงผลข้อความ www.Myarduino.net
display.display();
}
void loop() {
}
```

ตัวอย่างการเขียนโปรแกรม แสดงหน้าจอ ตัวอย่างเบื้องต้น

```
#include <Wire.h>
#include <ACROBOTIC_SSD1306.h>
void setup()
{
  Wire.begin();
  oled.init();                      // Initialze SSD1306 OLED display
  oled.clearDisplay();              // Clear screen
  oled.setTextXY(0,0);              // Set cursor position, start of line 0
  oled.putString("ACROBOTIC");
  oled.setTextXY(1,0);              // Set cursor position, start of line 1
  oled.putString("industries");
  oled.setTextXY(2,0);              // Set cursor position, start of line 2
  oled.putString("Pasadena,");
  oled.setTextXY(2,10);             // Set cursor position, line 2 10th character
  oled.putString("CA");
}
void loop()
{
}
```

25. 1602 LCD (Blue Screen) 16x2 LCD with backlight of the LCD screen

เกี่ยวกับการใช้งานจอภาพ LCD Display 1602 2004 ด้วย Module IIC/I2C Interface LCD Adapter IIC I2C Interface for Arduino ข้อดีของ Module LCD Adapter IIC I2C Interface for Arduino ก็คือสามารถช่วยลด ขา Pin จาก 6 ขาเหลือเพียง 2 ขาเท่านั้น และยังช่วยป้องกันการต่อสายไฟสัญญาณรบกวนได้อีกด้วย



addressตั้งต้นของจอ LCD มีอยู่สองแบบ 0x27 และ 0x3F

ให้ต่ออุปกรณ์ตามรูปข้างบน แล้ว อัพโหลด โค้ดตัวอย่างด้านล่างลง NodeMCU ESP8266

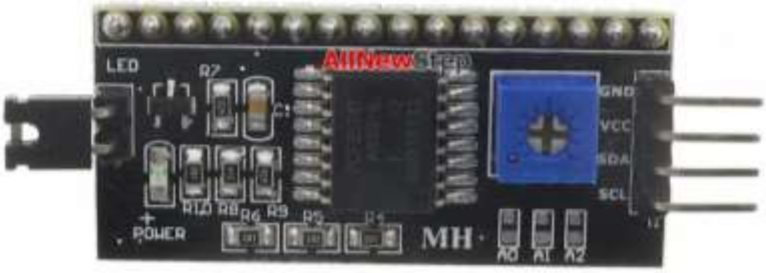
ตัวอย่างการเชื่อมต่อ แบบที่ 1

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2); // addressตั้งต้นของจอ LCD มีอยู่สองแบบ 0x27 และ 0x3F
void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่0 แถวที่ 1 เตรียมพิมพ์ข้อความ
  lcd.print("LCD1602 I2c Test"); //พิมพ์ข้อความ "LCD1602 I2c Test"
  lcd.setCursor(2, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรกำแหน่งที3 แถวที่ 2 เตรียมพิมพ์ข้อความ
  lcd.print("myarduino.net"); //พิมพ์ข้อความ "myarduino.net"
}
void loop() {
}
```

ตัวอย่างแบบที่ 2

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F,16,2);   // 0x27 or 0x3F
char* teams[]={"Liverpool", "Chelsea", "Man City", "Arsenal", "Man U","Barcelona","Real
madrid"}; //ข้อความที่เปลี่ยนไป
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
}
void loop() {
  for (int i = 0; i <= 6; i++){
   lcd.setCursor(2,0);
   lcd.print(teams[i]);
   delay(2000);
   lcd.clear();
  }
}
```

26. 1602 2004 LCD Adapter Plate IIC I2C Interface for Arduino



ถ้าคุณมีปัญหากับการต่อสายไฟหลายเส้น กับจอ LCD ทำให้สับสน หรือทำให้ขา Arduino ไม่พอใช้งาน โมดูล I2C LCD ตัวนี้เป็นตัวเลือกที่แนะนำ เพราะสามารถเชื่อมต่อกับหน้าจอ LCD รุ่น 16x2 หรือรุ่น 20x4 โดยใช้สายไฟเพียง 2 เส้น สะดวก ง่าย ช่วยให้เหลือขา Arduino ไว้ใช้งานที่สำคัญอย่างอื่น ใช้งานง่ายมีไลบารีมาตรฐานพร้อมใช้งาน

รายละเอียด 1602 2004 LCD Adapter Plate IIC I2C Interface for arduino

Supply voltage: 5V

backlight และ contrast ปรับค่าได้ด้วย potentiometer

Compatible for 1602 LCD

Weight:5g

Size:5.5x2.3x1.4cm

ตัวอย่างโคด 1602 2004 LCD Adapter Plate IIC I2C Interface for arduino

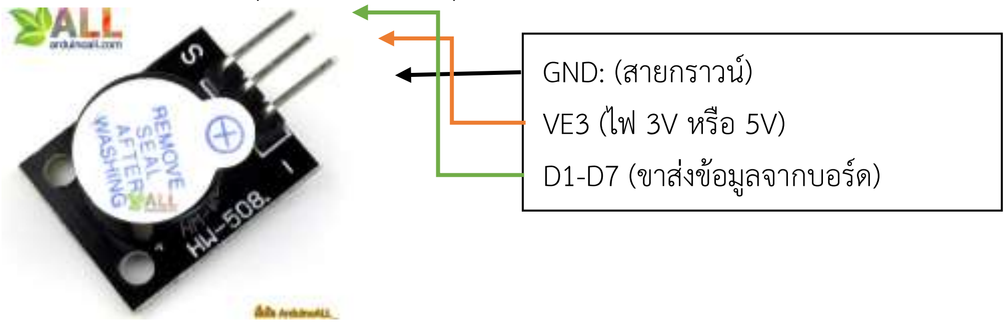
```
// แก้ไขข้อความแสดงผล ที่ตัวสีชมพู
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR 0x27 // หรือ 0X3F
#define BACKLIGHT_PIN 3
LiquidCrystal_I2C lcd(I2C_ADDR,2,1,0,4,5,6,7);

void setup()
{
//lcd.init (16,2); // <
lcd.begin (20,4); // <
// Switch on the backlight
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
lcd.setBacklight(HIGH);
lcd.home (); // ไปที่ตัวอักษรที่ 0 แถวที่ 1
lcd.setCursor(3,1); // ไปที่ตัวอักษรที่ 3 แถวที่ 2
lcd.print("Welcome ALL To");
lcd.setCursor(5,2); // ไปที่ตัวอักษรที่ 5 แถวที่ 3
lcd.print("ArduinoAll");
}
void loop()
{
}
```

27. Active Buzzer Module

รายละเอียดสินค้า

Active Buzzer Module

This is Arduino compatible active speaker buzzer alarm module for Arduino



Arduino buzzer module

Active speaker Compatible with PC- printer- car audio system DIY

Specifications:

Voltage: 5V

Color: Black + silver gray

Package dimension: 77x42x13mm

Weight: 5g

วิธีการต่อใช้งาน

Buzzer ----> Arduino UNO R3

VCC ----> 5V

I/O ----> 2

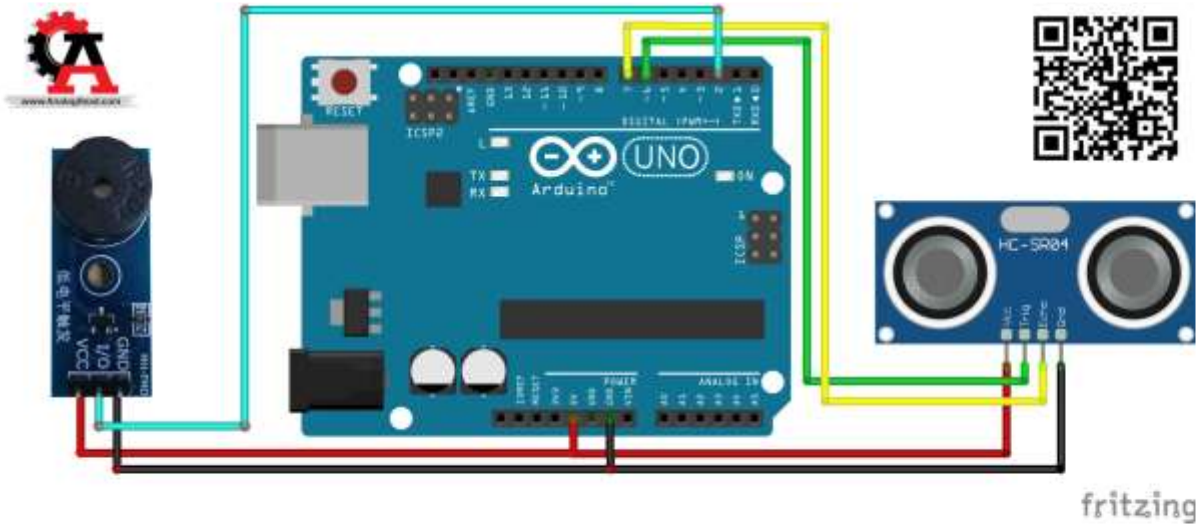
GND ----> GND

Arduino_active_buzzer_begin.ino

```
#define BUZZER_PIN 2
void setup() {
  pinMode(BUZZER_PIN,OUTPUT);
}
void loop() {
  digitalWrite(BUZZER_PIN,HIGH);   //ปิดเสียงเตือน
  delay(1000);
  digitalWrite(BUZZER_PIN,LOW);    //เปิดเสียงเตือน
  delay(1000);
}
```

ตัวอย่างการใช้งาน

1.จะมีเสียงเตือนดังดังถี่ขึ้นเมื่อมีวัตถุเข้าใกล้



```
#define BUZZER_PIN 2
#define echoPin 7
#define trigPin 6
void setup() {
  Serial.begin(9600);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {
  Serial.println(Ultrasonic());
  int cm = Ultrasonic();
  if (cm < 60 && cm != 0) {     //เมื่อวัตถเข้าใกลสียงเตือนดังถี่ขึ้น
    digitalWrite(BUZZER_PIN, HIGH);  //ปิดเสียงเตือน
    delay(cm * 16);
    digitalWrite(BUZZER_PIN, LOW);   //เปิดเสียงเตือน
    delay(cm * 16);
  }
  else {                              //เมื่อห่างมากกว่า 60 cm จะไม่มีเสียง
    digitalWrite(BUZZER_PIN, HIGH);   //ปิดเสียงเตือน
    delay(500);
  }
}
int Ultrasonic() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration / 58.2;
  return distance;

}
```

Sensor ตรวจจับการเคลื่อนไหว PIR (HC-SR501)

หากเราพูดถึงการใช้งาน Sensor ตรวจจับความเคลื่อนไหว (Motion Sensor) โดยทั่วไปแล้วจะมีลักษณะการทำงานของ Sensor อยู่หลากหลายประเภท วันนี้แอดมินจะหยิบยกหลักการทำงานของ Sensor ตรวจจับความเคลื่อนไหวทั้ง 3 ประเภท

1. Passive Infrared Sensor (PIR)

HC-SR501 PIR ตรวจจับการเคลื่อนไหว Motion Sensor

เป็น Sensor ที่ตรวจจับความร้อนจากร่างกาย เมื่อมีการขยับหรือเคลื่อนไหวร่างกาย โยอาจจะตรวจจับจากการสะท้อนของแรง ความร้อน โดยที่ไม่มีการปล่อยหลังงานออกมาจาก Sensor ก็จะเป็นประเภท Sensor HC-SR501 ที่เราจะหยิบยกมาพูดกันในวันนี้

2. เคลือนเสียง Ultrasonic

HC-SR04 Sensor Ultrasonic Module Distance

เป็น Sensor ที่มีการปล่อยคลื่นเสียงอัลตร้าโซนิคออกมา และจะใช้การตรวจจับจากการสะท้อนกลับมาของคลื่นเสียงเมื่อกระทบต่อวัตถุ

3. เคลื่อน Microwave

HB100 Doppler Sensor Microwave Radar Technology

เป็น Sensor ที่มีการปล่อยพลังงานคลื่นย่านไมโครเวฟออกมาและตรวจจับการสะท้อนเมื่อวัตถุมีการเคลื่อนไหวเคลื่อนที่ ด้วยการตรวจจับด้วยความร้อยในพื้นที่ต้องการ ความร้อนที่วัดได้จะมีการเปลี่ยนแปลงระดับรังสีอินฟาเรดที่ (สิ่งมีชีวิจทุกชนิดจะ แผ่รังสีอินฟราเรดเป็นเอกลักษณ์ของตัวมันเอง)ข้อดี สามารถทะลุสิ่งกีดขวางได้

PIR (Passive Infrared)

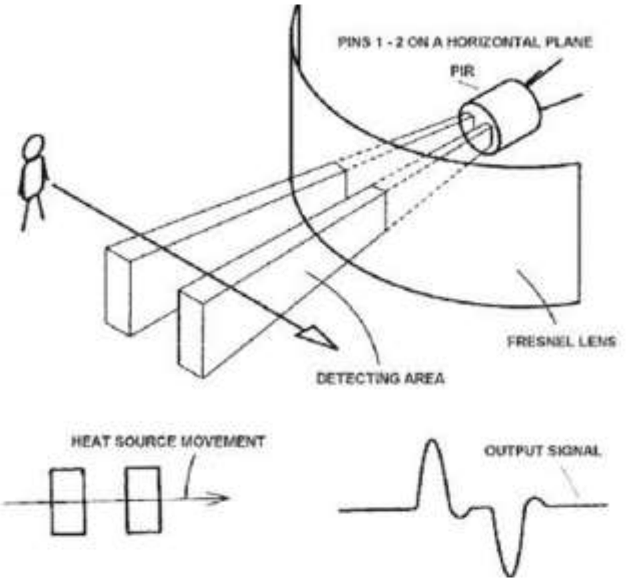
คืออุปกรณ์ที่ใช้ตรวจจับเคลือนรังสี Infrared จากวัตถุ ผ่านอุปกรณ์รวมแสง (เลนส์) มายังตัว Pyro Electyic ทำหน้าที่เปลี่ยนแปลงพลังงานความร้อนจากรังสี Infrared เป็นพลังงานไฟฟ้า เพียงสัญญาณ Infrared เล็กน้อยสามารถตรวจจับได้

ส่วนประกอบของ Sensor ก็จะประกอบด้วย

1. เลนส์ ใช้สำหรับควบคุม โฟกัสพื้นที่ในการตรวจจับความเคลื่อนไหว
2. เซนเซอร์ เป็นตัวแปลงพลังงานความร้อนที่ได้รับจากรักสีอินฟราเรด มาเป็นพลังงานไฟฟ้า

PIR Motion Sensor

คืออุปกรณ์ Sensor ชนิดหนึ่งที่ใช้ตรวจจับการเคลื่อนไหวโดยใช้วิธีการตรวจจับด้วยคลื่นรังสี Infrared ที่กระจายมาจาก มนุษย์ สัตว์ ที่มีการเคลื่อนไหว PIR จึงนำมาประยุคใช้งานทางด้านการตรวจจับการเคลื่อนไหวของสิ่งมีชีวิต



ภาพจาก learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work

งานที่เหมาะสมกับ Sensor PIR Motion Sensor

PIR หรือ Motion Sensor มักจะใช้ในงานที่มีการเคลื่อนไหวและเปิดชั่วคราวเช่น การเปิดไฟในทางเดินสวน ทางเดินนอกอาคาร เปิดไฟในห้องน้ำ ความปลอดภัยตรวจจับขโมย เป็นต้น

คำแนะนำสำหรับการใช้งาน PIR Motion Sensor

1. ไม่ควรติดตั้ง Sensor กลางแดดหรือสัมผัสกับแสงแดดโดยตรง เพราะความร้อนอาจจะทำให้ Sensor ทำงานผิดปกติ
2. ไม่ควรใช้งานที่มีการสั่นสะเทือนแรงๆ

นี้ก็เป็นการอธิบายหลักการทำงานของ Sensor ตรวจจับความเคลื่อนไหวแบบมึนๆคราวนี้มาถึงตัวอย่างการใช้งาน Sensor PIR Motion Sensor HC-SR501 กันเลยครับ

ตัวอย่างการต่อใช้งาน

Arduino	HC-SR501
D2	OUTPUT

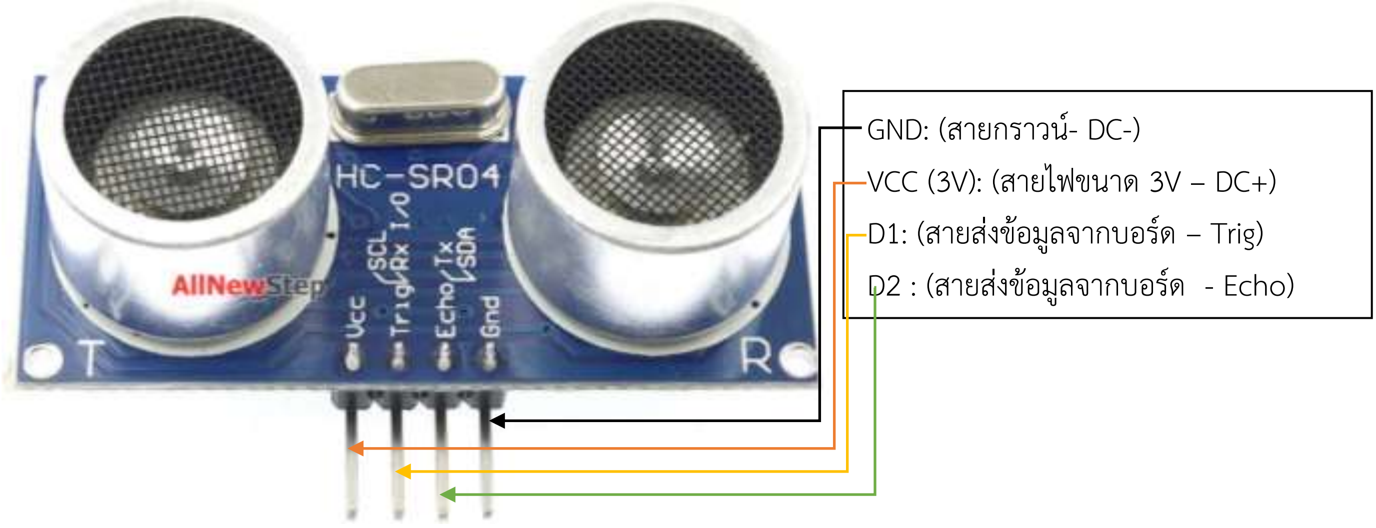
5V 5V

GND GND

Code Arduino + HC-SR501 PIR

```
/*
  จากบทความ สอนใช้งาน Sensor ตรวจจับการเคลื่อนไหว PIR (HC-SR501)
  https://www.9arduino.com/article/73/
*/
const int buttonPin = 2;
const int ledPin =  13;
int buttonState = 0;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
    delay(10000) ;
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

28. Ultrasonic SR04 เซนเซอร์ Ultrasonic Module HC-SR04 Distance Measuring Transducer Sensor



Sensor hc-sr04 เป็น Sensor ใช้วัดระยะทางด้วยคลื่นเสียงย่านอัลตร้าโซนิค โดนการส่งคลื่นอัลตร้าโซนิคและสะท้อนกลับมาหาวัตถุ จากนั้นคำนวนระยะเวลาที่คลื่นอัลตร้าโซนิคได้กระทบวัตถุแล้วกลับมายัง Sensor ก็จะได้ค่า ระยะทาง

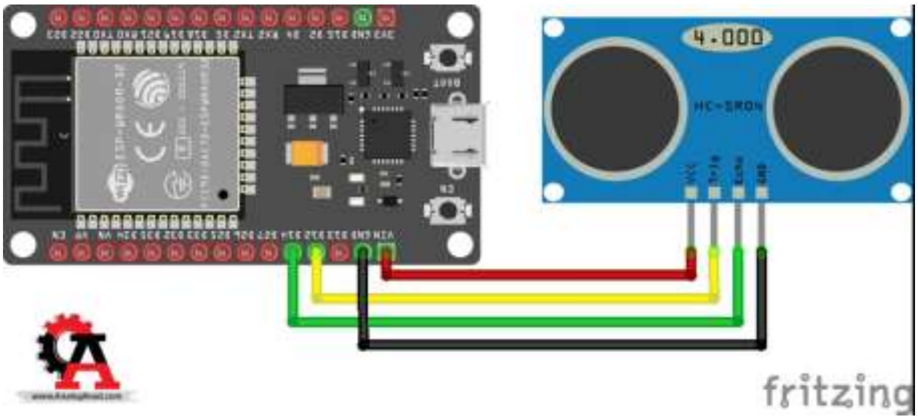
วิธีการต่อขา ใช้งาน โมดูลวัดระยะทาง Ultrasonic Module Distance Measuring Transducer Sensor กับ Arduino

- Vcc - 5v
- Gnd - Gnd
- Trig – D1
- Echo – D2

ตัวอย่างโคด Arduino โมดูลวัดระยะทาง Ultrasonic Module Distance

```
/*  การใช้งาน hc-sr04 Arduino และตัวอย่างโปรแกรม  https://www.9arduino.com/article/73/
*/
#define echoPin 3 // Echo Pin
#define trigPin 2 // Trigger Pin
#define LEDPin 13 // Onboard LED
int maximumRange = 200; // Maximum range needed
int minimumRange = 0; // Minimum range needed
long duration, distance; // Duration used to calculate distance
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  //Calculate the distance (in cm) based on the speed of sound.
  distance = duration / 58.2;
  if (distance >= maximumRange || distance <= minimumRange) {
    Serial.println("-1");  //เมื่ออยู่นอกระยะให้ใช้ Print -1
    digitalWrite(LEDPin, HIGH);
  }
  else {
```

```
    Serial.println(distance);  //แสดงค่าระยะทาง
    digitalWrite(LEDPin, LOW);
  }
  delay(100);
}      // The speed of sound is 340 m/s or 29 microseconds per centimeter.
      // The ping travels out and back, so to find the distance of the
      // object we take half of the distance travelled.
      return microseconds / 29 / 2;
    }
```



NodeMCU ESP32 ESP8266 กับ Ultrasonic แจ้งเตือนแอพพลิเคชั่น Line

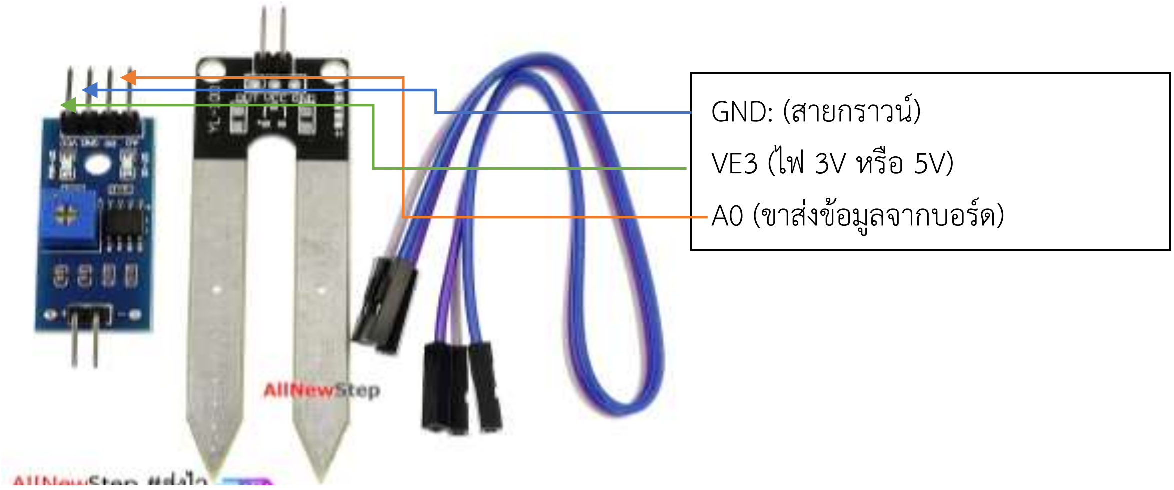
```
#ifdef ESP32
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <TridentTD_LineNotify.h>
#define SSID        "xxxx"   //ใส่ชื่อ Wifi
#define PASSWORD    "xxxx"   //ใส่รหัส Wifi
#define LINE_TOKEN  "xxxx"   //ใส่ TOKEN
void setup() {
  Serial.begin(115200);
  Serial.println(LINE.getVersion());
  WiFi.begin(SSID, PASSWORD);
  Serial.printf("WiFi connecting ",  SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(400);
  }
  Serial.printf("\nWiFi connected\nIP : ");
  Serial.println(WiFi.localIP());
```

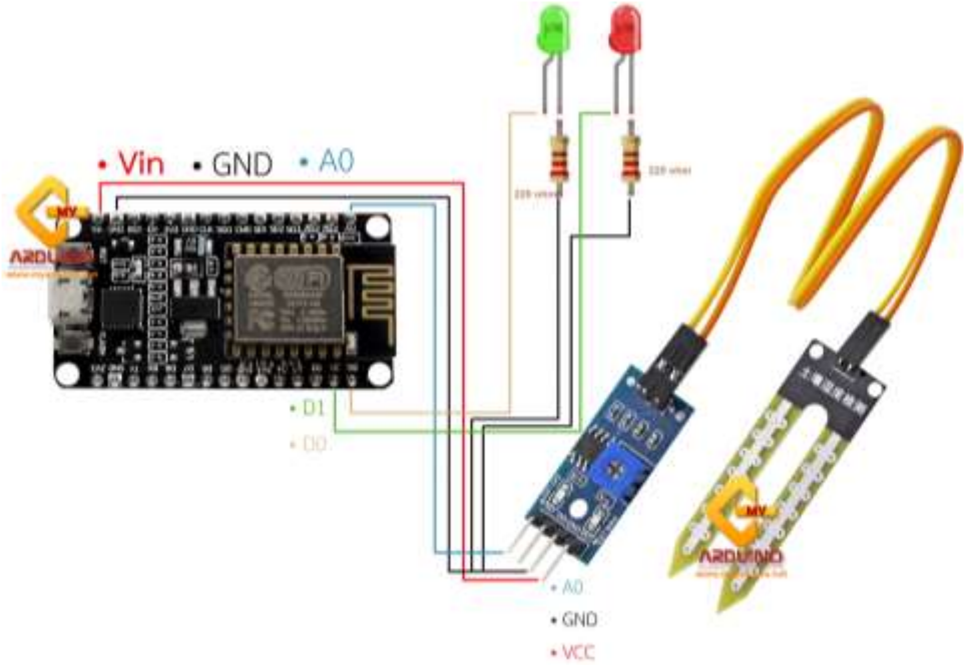
```
  LINE.setToken(LINE_TOKEN);
}
void loop() {
  int cm = Ultrasonic(D1,D2);    //ESP8266
//  int cm = Ultrasonic(12,14);  //ESP32
  if (cm < 20) {                  //หากน้อยกว่า 20cm แจ้งเตือนไลน์
    LINE.notify("ระยะ = " + String(cm) + " cm");
    delay(1000);
  }
}
int Ultrasonic(byte trigPin,byte echoPin) {            //ฟังก์ชั่นอ่านค่าระยะทาง
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration / 58.2;
  return distance;
}
```

29. Soil Moisture Sensor Module เซ็นเซอร์วัดความชื้นในดิน

Soil Moisture Sensor Module

เซนเซอร์วัดความชื้นในดิน ให้เอาต์พุตเป็นอะนาล็อกป้อนให้กับวงจร Arduino เพื่อนำค่าไปเปรียบเทียบตามต้องการ ใช้งานง่าย





โค้ดตัวอย่างการใช้งานโมดูลกับบอร์ด Arduino UNO

```
int sensorPin = A0;
void setup() {
  Serial.begin(9600); // set up serial port for 9600 baud (speed)
  delay(500); // wait for display to boot up
}
void loop() {
   int sensorValue;
  sensorValue = analogRead(sensorPin);
  sensorValue = map(sensorValue, 0, 1023, 0, 100);
  Serial.print("Soil moisture: ");
  Serial.print(sensorValue);
  Serial.println(" %");
    delay(500); //wait for half a second, so it is easier to read
}
```

อีกตัวอย่างการต่อ Soil Sensor

```
int sensorPin = A0;  //sensor input pin
int moisture =0;      // variable to store the value coming from the sensor
void setup() {
  Serial.begin(9600);
}
void loop() {
  // read the value from the sensor
```

```
  int moisture = analogRead(sensorPin);
  Serial.print("Soil Moisture = ");
  Serial.println(moisture);
  delay(500);
}
```

อีกตัวอย่างหนึ่งการใช้ Soil Sensor ควบคุมการเปิด-ไฟ

```
int ledPin = 2;
int ledPin3 = 3;
int analogPin = 5; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขาที่5
int val = 0;
void setup() {
pinMode(ledPin, OUTPUT); // sets the pin as output
pinMode(ledPin3, OUTPUT); // sets the pin as output
Serial.begin(9600);
}

void loop() {
val = analogRead(analogPin); //อ่านค่าสัญญาณ analog ขา5 ที่ต่อกับ Soil Moisture Sensor Module
v1
Serial.print("val = "); // พิมพ์ข้อมความส่งเข้าคอมพิวเตอร์ "val = "
Serial.println(val); // พิมพ์ค่าของตัวแปร val
if (val < 500) {
digitalWrite(ledPin, LOW); // สั่งให้ LED ที่ Pin2 ดับ
digitalWrite(ledPin3, HIGH); // สั่งให้ LED ที่ Pin3 ติดสว่าง
}
else {
digitalWrite(ledPin, HIGH); // สั่งให้ LED ที่ Pin2 ติดสว่าง
digitalWrite(ledPin3, LOW); // สั่งให้ LED ที่ Pin3 ดับ

}
delay(100);
}
```

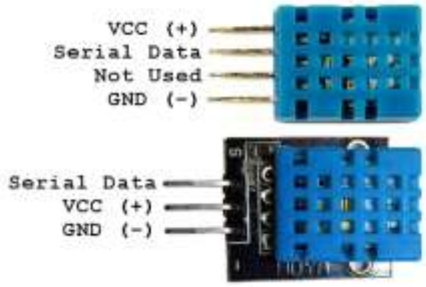
สอนแสดงค่าความชื้นในดินในแอพ Blynk (ดูรูปประกอบท้ายเล่ม)

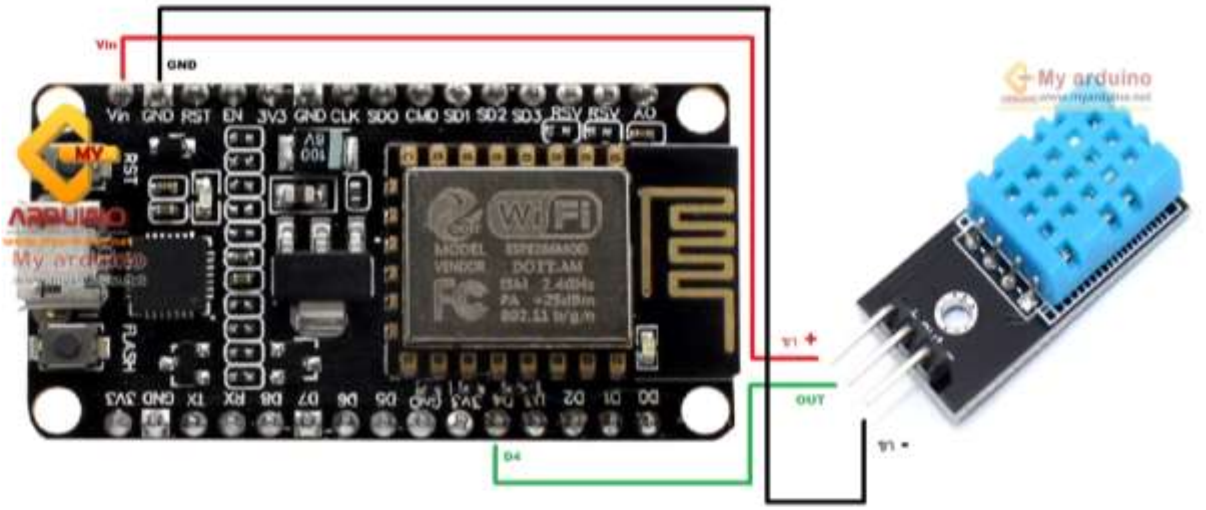
ตัวอย่างโค้ด

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define LED1 D2 // เสียบPin D2
char auth[] = "xxxxxxxxxxxxxxxxxxxxxxxxxx"; //Auth Token ที่ได้จากเมลล์
char ssid[] = "xxxxxxxxxxxxxx"; //ใส่ชื่อไวไฟ
char pass[] = "xxxxxxxxxxxxxx"; // ใส่พาสเวิดไวไฟ
int soil = 0;
int SoilSensor = A0; // เสียบPin A0
float Soil_auto = 0;
void setup() {
Serial.begin (9600);
pinMode(LED1, OUTPUT);
Blynk.begin(auth, ssid, pass);
}

BLYNK_WRITE(V1)
{
Soil_auto = param.asInt(); //กำหนดค่าความชื้นในดินจากBlynk
}
void loop(){
Blynk.run();
soil = analogRead(SoilSensor);
float LevelSoil = (soil * (100.0 / 1024.0))-100;
float LevelSoil1 = LevelSoil*(-1);
Serial.print("soil = ");
Serial.println(LevelSoil1);
Blynk.virtualWrite(V2, LevelSoil1);
if (LevelSoil1 < Soil_auto ) {
digitalWrite(LED1, HIGH); //หากความชื้นในดินน้อยกว่าค่าที่ตั้งไว้ LED1 ทำงาน
}
else {
digitalWrite(LED1, LOW);
```

```
}
}
```

30. Temprature and Humidity Sensor DHT11 module





DHT_Unified_Sensor.ino

```
// DHT Temperature & Humidity Sensor
// Unified Sensor Library Example
// Written by Tony DiCola for Adafruit Industries
// Released under an MIT license.
// REQUIRES the following Arduino libraries:
// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library
// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit_Sensor
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN 2     // Digital pin connected to the DHT sensor
// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --
// Pin 15 can work but DHT must be disconnected during program upload.
// Uncomment the type of sensor in use:
```

```
//#define DHTTYPE    DHT11     // DHT 11
#define DHTTYPE    DHT22     // DHT 22 (AM2302)
//#define DHTTYPE    DHT21     // DHT 21 (AM2301)
// See guide for details on sensor wiring and usage:
//   https://learn.adafruit.com/dht/overview
DHT_Unified dht(DHTPIN, DHTTYPE);
uint32_t delayMS;
void setup() {
  Serial.begin(9600);
  // Initialize device.
  dht.begin();
  Serial.println(F("DHTxx Unified Sensor Example"));
  // Print temperature sensor details.
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  Serial.println(F("------------------------------------"));
  Serial.println(F("Temperature Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("°C"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("°C"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("°C"));
  Serial.println(F("------------------------------------"));
  // Print humidity sensor details.
  dht.humidity().getSensor(&sensor);
  Serial.println(F("Humidity Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("%"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("%"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("%"));
  Serial.println(F("------------------------------------"));
  // Set delay between sensor readings based on sensor details.
```

```
  delayMS = sensor.min_delay / 1000;
}
void loop() {
  // Delay between measurements.
  delay(delayMS);
  // Get temperature event and print its value.
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println(F("Error reading temperature!"));
  }
  else {
    Serial.print(F("Temperature: "));
    Serial.print(event.temperature);
    Serial.println(F("°C"));
  }
  // Get humidity event and print its value.
  dht.humidity().getEvent(&event);
  if (isnan(event.relative_humidity)) {
    Serial.println(F("Error reading humidity!"));
  }
  else {
    Serial.print(F("Humidity: "));
    Serial.print(event.relative_humidity);
    Serial.println(F("%"));
  }
}
```

DHTtester.ino (ตัวอย่างการเชื่อมต่อ)

```
// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

// REQUIRES the following Arduino libraries:
```

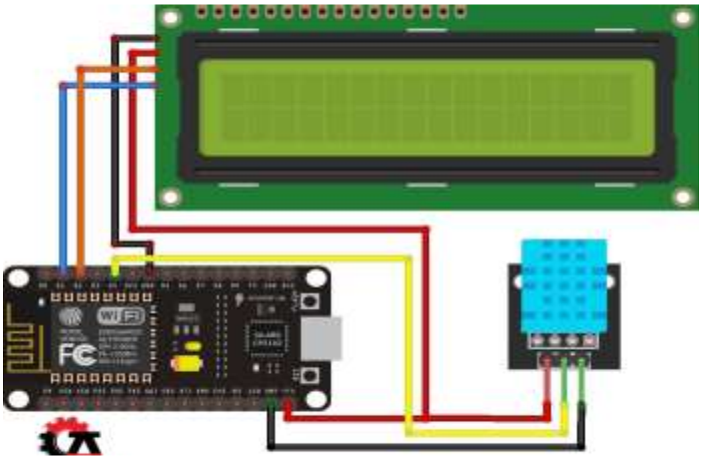
```
// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library
// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit_Sensor
#include "DHT.h"
#define DHTPIN 2     // Digital pin connected to the DHT sensor
// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --
// Pin 15 can work but DHT must be disconnected during program upload.
// Uncomment whatever type you're using!
//#define DHTTYPE DHT11   // DHT 11
#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21   // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
// to 3.3V instead of 5V!
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 3 (on the right) of the sensor to GROUND (if your sensor has 3 pins)
// Connect pin 4 (on the right) of the sensor to GROUND and leave the pin 3 EMPTY (if your
sensor has 4 pins)
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.
// Note that older versions of this library took an optional third parameter to
// tweak the timings for faster processors.  This parameter is no longer needed
// as the current DHT reading algorithm adjusts itself to work on faster procs.
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
  dht.begin();
}
void loop() {
  // Wait a few seconds between measurements.
  delay(2000);
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
```

```
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);
  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));
}
```

อีกตัวอย่างของการต่อ DH11

```
#include "DHT.h"
DHT dht;
void setup()
{
```

```
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");
  dht.setup(2); // data pin 2
}

void loop()
{
  delay(dht.getMinimumSamplingPeriod());
  float humidity = dht.getHumidity(); // ดึงค่าความชื้น
  float temperature = dht.getTemperature(); // ดึงค่าอุณหภูมิ
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1);
  Serial.print("\t\t");
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  Serial.println(dht.toFahrenheit(temperature), 1);
  delay(1000);
}
```



การต่อวงจร

LCD ----> ESP8266

GND ----> GND

VCC ----> Vin

SDA ----> D2

SCL ----> D1

DHT11 ----> ESP8266

\+ ----> Vin

OUT ----> D4

\- ----> GND

บอร์ด ESP8266 กับ DHT11 แสดงค่าอุณหภูมิและความชื้นด้วยหน้าจอ LCD I2C

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
```

```
#define DHTPIN D4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  Serial.begin (115200);
  dht.begin();
  lcd.begin();
  lcd.display();        //เปิดหน้าจอ
  lcd.backlight();      //เปิดไฟ backlight
  lcd.clear();          //ล้างหน้าจอ
}

void loop() {
  float h = dht.readHumidity();              //รับค่าความชื้น
  float t = dht.readTemperature();           //รับค่าอุณหภูมิ
  lcd.setCursor(0, 0);
  lcd.print("Temp = "+String(t,1)+" C");    //แสดงค่าอุณหภูมิ
  lcd.setCursor(0, 1);
  lcd.print("Humi = "+String(h,1)+" %");    //แสดงค่าความชื้น
  delay(1000);
  lcd.clear();
}
```

31. Micro PIR human body sensing module PIR module infrared module

เซนเซอร์ PIR ตรวจจับความเคลื่อนไหวของสิ่งมีชีวิต

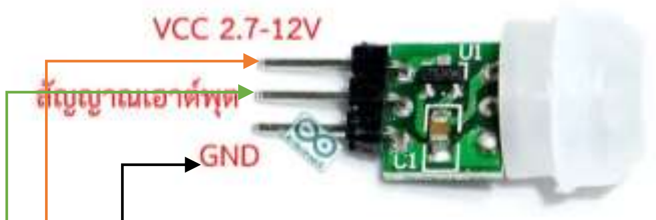
เซนเซอร์ Micro PIR ขนาดเล็กประหยัดไฟ

สำหรับตรวจจับการเคลื่อนไหวของสิ่งมีชีวิต โดยใช้การวัดความร้อนโดยแสงอินฟาเรด ทำงานได้ในช่วงไฟเลี้ยงกว้าง 2.7-12V ใช้พลังงานในการทำงานต่ำเพียง <0.1mA

ใช้งานง่าย จ่ายไฟให้ที่ขา VCC และ GND

การตรวจจับการเคลื่อนไหว จะให้สัญญาณ 0 หรือ 1 ออกมาทางขาสัญญาณเอาต์พุต VOUT

ใช้คำสั่ง digitalRead() อ่านค่าเข้าประมวลผล



GND: (สายกราวน์- DC-)
VCC (3V): (สายไฟขนาด 3V –VCC) หรือต่อกับ 5V
D1-D7 (สายส่งข้อมูลจากบอร์ด)

PIR Motion Sensor Module -> ESP8266
+5V -> 5V or 3V
Gnd -> Gnd
Output -> D1-D7

PIR_Sound.ino

```
int buzzer = D5;
int PIR_Pin = D2;
int led1=D6;
int sensor_value =0;

void setup() {
  pinMode(buzzer, OUTPUT);
  pinMode(led1, OUTPUT);

  Serial.begin(9600);
  pinMode(PIR_Pin, INPUT);
}

void loop() {
  int sensor_Value=digitalRead(PIR_Pin);
   if (sensor_Value == HIGH) {
    Serial.println("PIR Motion Sesor : Power ON");
    digitalWrite(led1,HIGH);
    tone(buzzer,440);
    delay(100);
  }
  else {
    Serial.println("PIR Motion Sensor : Power OFF");
    noTone(buzzer);
    digitalWrite(led1,LOW);
```

```
  }
  delay(1000);
}
```

อีกตัวอย่างหนึ่ง การต่อกับลำโพง

M0015_ESP8266.ino

```
int buzzer = 5;
int digitalPin = 4;
int val = 0;
void setup() {
  pinMode(buzzer, OUTPUT);
  pinMode(digitalPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  val = digitalRead(digitalPin);
  Serial.print("val = ");
  Serial.println(val);
  if (val == 0) {
    digitalWrite(buzzer, HIGH);
  }
  else {
    digitalWrite(buzzer, LOW);
  }
  delay(100);
}
```

ตัวอย่างCode การเชื่อมต่อกับหลอดไฟ

```
int ledPin= 13;
int inputPin= 3;
void setup(){
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);
}
void loop(){
  int value= digitalRead(inputPin);
  if (value == HIGH)
```

```
  {
    digitalWrite(ledPin, HIGH);
    delay(1000) ;
  }
  else
  {
    digitalWrite(ledPin, LOW);
delay(1000) ;
  }
}
```

32. Digital Temperature Temp Sensor Probe DS18B20 For Thermometer Waterproof 100CM

สำหรับน้องๆที่ต้องการวัดอุณหภูมิ ในน้ำ หรือในห้องเย็น โดยใชช้ Arduino ทางร้านแนะนำ Sensor Ds18b20 สามารถวัดอุณหภูมิได้ -55°C ถึง 125°C ความผิดพลาด +/-0.5 °C (ในช่วงระหว่าง -10°C ถึง 85°C) สามารถนำไปใช้ในงานโปรเจคได้หลากหลายไม่ว่าจะเป็นการหาค่าความเย็นในตู้เย็น โรงเพาะเห็ด ตู้แช่ยา เลี้ยงปลา จนไปถึงอะไรก็ได้ที่ต้องการวัดอุณหภูมิมีความชื้นสูงหรือในน้ำ ได้ทั้งหมด

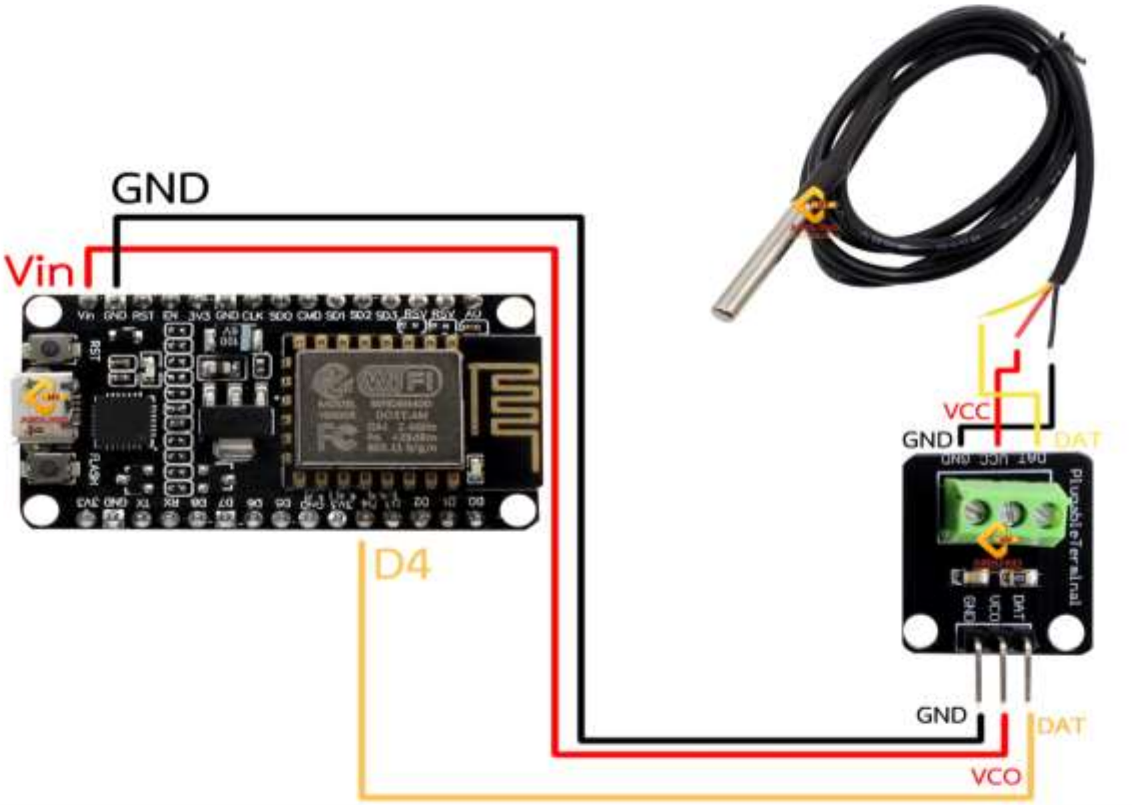
1.เชื่อมต่ออุปกรณ์ตามด้านล่าง

ESP8266 -> Plug able Terminal

- Vin -> VCC
- GND -> GND
- D4 -> DAT

Plug able Terminal -> DS18B20 Full Waterproof

- VCC -> สายสีแดง
- GND -> สายสีดำ
- DAT -> สายสีเหลือง



Code ตัวอย่าง Sensor DS18B20 DS18B20.ino

```
/*
วิธีการใช้งาน Sensor DS18B20 Arduino วัดอุณหภูมิในน้ำ
บทความ จาก https://www.ab.in.th/b/18
*/
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2 //กำหนดขา
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
void setup(void) {
 Serial.begin(9600);
 Serial.println("Dallas Temperature IC Control Library");
 Serial.println("by https://www.ab.in.th");
 sensors.begin();
}
void loop(void) {
 Serial.print(" Requesting temperatures...");
 sensors.requestTemperatures(); //อ่านข้อมูลจาก library
 Serial.print("Temperature is: ");
```

```
 Serial.println(sensors.getTempCByIndex(0)); // แสดงค่า อูณหภูมิ sensor 0
delay(1000);
}
```

วิธีการใช้งาน Arduino DS18B20

1. ลงไลบารี DS18B20 และ OneWire ที่เมนู Library Manager พิมพ์ 18b20 และ OneWire
2. ต่อสายดังนี้

    สายสีแดง VCC - 5V

    สายสีดำ GND - GND

    สายสีเหลือง DATA - 7

    ต่อ R 4.7K-10K ระหว่างสายสีเหลืองกับสีแดง

DS18820.ino

```
/*
// ลิงค์โคดนี้มาจากร้าน ArduinoAll.com เท่านั้น
// ถ้าเปิดมาจากลิงค์ร้านอื่น แสดงว่าก็อปปี้ ละเมิดลิขสิทธิ์ มาจากร้าน ArduinoAll.com
// สงวนลิขสิทธิ์ ห้ามนำไปใช้เชิงพานิชย์ ควรมีความคิดตั้งใจ สร้างผลงานด้วยตัวเองบ้าง  ก็อปปี้ผลงานคนอื่น
= ขโมย
// ไม่สงวนลิขสิทธิ์สำหรับแบ่งปันความรู้
ต่อสายดังนี้
สายสีแดง VCC - 5V                    สายสีดำ GND - GND
สายสีเหลือง DATA – 7                  ต่อ R 4.7K-10K ระหว่างสายสีเหลืองกับสีแดง
*/

#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

void setup(void)
{
  Serial.begin(9600);
  Serial.println("ArduinoAll DS18B20 TEST");
  sensors.begin();
}
```

```
void loop(void)
{
  Serial.print("Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");
  Serial.print("Temperature for the device 1 (index 0) is: ");
  Serial.println(sensors.getTempCByIndex(0));
  delay(1000);
}
```

รายละเอียด Digital Temperature Temp Sensor Probe DS18B20 For Thermometer Waterproof 100CM

Power Supply Range: 3.0V to 5.5V

Adjustable Resolution: 9 - 12

Operating Temperature Range: -55℃ to +125℃

Output lead: Red (VCC), Black (GND), Yellow (DATA)

Cable length: 100 cm

Stainless Steel Tube Size: 6x 45mm

ข้อมูลเพิ่มเติม Digital Temperature Temp Sensor Probe DS18B20 For Thermometer Waterproof 100CM

http://www.hobbytronics.co.uk/ds18b20-arduino

33. โมดูลแปลง Adapter DS18B20 เซนเซอร์วัดอุณหภูมิแบบกันน้ำ แปลงสายไฟให้เป็นขาแบบก้างปลา

โมดูลแปลง Adapter DS18B20 เซนเซอร์วัดอุณหภูมิแบบกันน้ำ แปลงสายไฟให้เป็นขาแบบก้างปลา บอร์ดสำหรับใช้กับ เซนเซอร์วัดอุณหภูมิแบบกันน้ำ DS18B20 แปลงสายสัญญาณ จากสายไฟ 3 เส้นหัวต่อแบบ terminal block ให้เป็นขาแบบก้างปลา มี R pullup สัญญาณแล้ว พร้อมสกรีนตำแหน่งแต่ละขา การต่อสายกับ DS18B20 มี 2 แบบ เช็คสีของสายไฟดังนี้

แบบที่ 1

สีแดง : VCC

สีดำ : GND

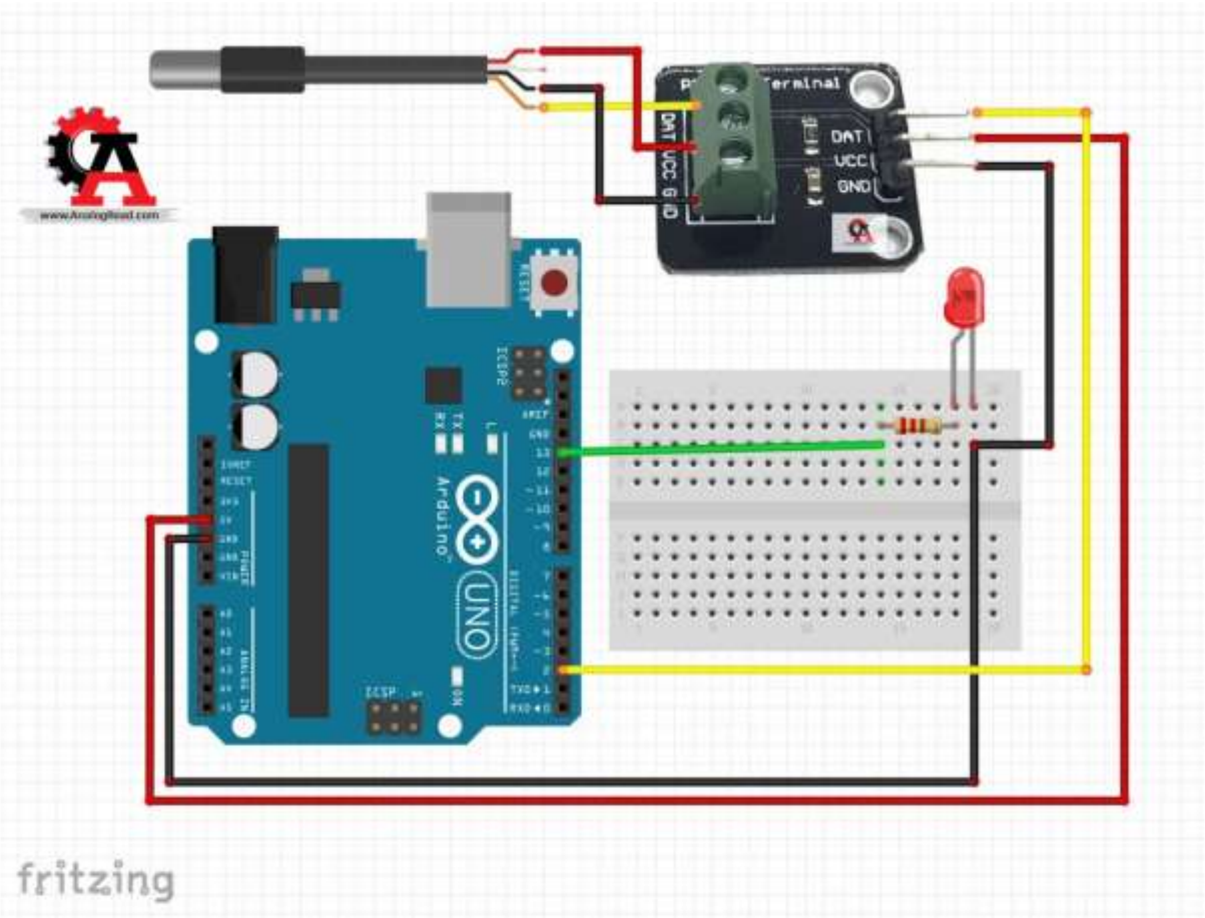
สีเหลือง : DATA

แบบที่ 2

สีแดง : VCC ---> 3V or 5V

สีเหลือง : GND ---> GND

สีเขียว : DATA ---> D2




```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
#define LEDPIN 13
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
void setup(void)
{
  Serial.begin(9600);
  pinMode(LEDPIN,OUTPUT);
```

```
  sensors.begin();
}
void loop(void)
{
  sensors.requestTemperatures();
  float temp = sensors.getTempCByIndex(0);
  Serial.print("Temperatures = ");
  Serial.print(temp);
  Serial.println(" C");
  if (temp >= 50) {            //เมื่ออุณหภูมิมากกว่า50°C LEDจะสว่าง
    digitalWrite(LEDPIN,HIGH);
    Serial.println("LED : ON");
  }
  else{                        //เมื่ออุณหภูมิน้อยกว่า50°C LEDจะดับ
    digitalWrite(LEDPIN,LOW);
    Serial.println("LED : OFF");
  }
}
```

34. ISD1820 Voice Board Module On-board Microphone Sound Recording โมดูลอัดเสียงบันทึกเสียงและเล่นเสียง ควบคุมด้วย Arduino ได้

รายละเอียดสินค้า

ISD1820 Voice Board Module (On-board Microphone) Sound Recording Module

ถ้าใครต้องการโมดูลเล่นเสียงที่เล่นเสียงได้ 1 เสียงcแต่เปลี่ยนเสียงเล่นได้ไม่จำกัด ใช้งานง่ายและมีราคาถูก ขอแนะนำ โมดูลบันทึกและเล่นเสียง ใช้ชิพ ISD1820 ไฟเลี้ยง 3-5V บันทึกได้สูงสุด 10 วินาที สามารถกดบันทึกเสียง เล่นเพลง เล่นเพลงแบบวนซ้ำที่ปุ่มกดบนโมดูล หรือเขียนโคด Arduino ควบคุม อัดเสียง เล่นเสียงได้

Arduino สอนใช้งาน ISD1820 กับ Arduino โมดูลอัดเสียงและเล่นเสียง ควบคุมด้วย Arduino

การใช้งานอัดเสียงและเล่นเสียงบนโมดูล ISD1820

สวิตช์ FT สำหรับต่อเป็นไมค์ออกเครื่องเสียง : เลื่อนไปที่ OFF

สวิตช์ REPEAT สำหรับเล่นซ้ำไปเรื่อย ๆ : เลื่อนไปที่ OFF

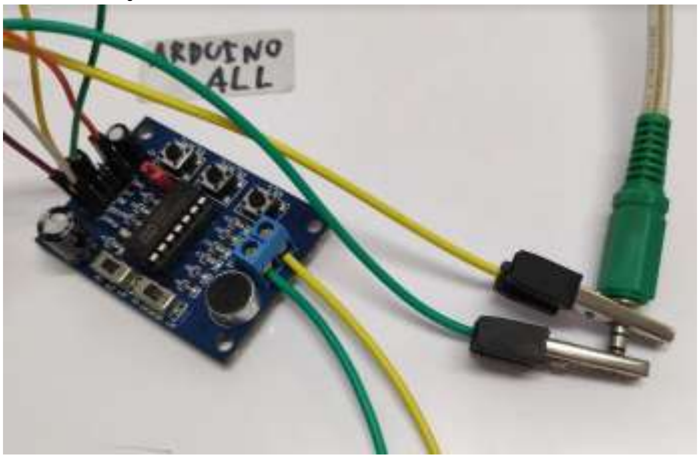
จ่ายไฟ 3-5V ที่ช่อง VCC

กดปุ่ม REC : อัดเสียงที่ไมค์ จะเห็นไฟสีแดงบนโมดูลสว่าง

กดปุ่ม PLAYE : เล่นเสียงหนึ่งครั้ง

กดปุ่ม PLAYL : เล่นเสียงในขณะที่กด

SPEAKER : สัญญาณเอาต์พุต ต่อกับลำโพง 0.25W หรือต่อเป็นสัญญาณอินพุตให้เครื่องขยายเสียง ตัวอย่างดังรูป



การควบคุมโมดูล ISD1820 ด้วย Arduino

ISD1820 โมดูลนี้จะมีขาสำหรับต่อควบคุมแทนการกดปุ่ม ในบอร์ดจะมีสกรีนขาว่าแทนปุ่มไหนการควบคุมทำได้ง่ายมาก เพียงจ่ายไฟสัญญาณ 1 หรือ 3-5V ที่ขาที่ต้องการ ก็จะทำงานทันที

สำหรับจัมเปอร์สีแดงบนบอร์ดเป็นตัวเปิด/ปิดโมดูล ถ้าต้องการให้โมดูลทำงานต้องเสียบจัมเปอร์ P2 ได้ด้วย ถ้าต้องการปิดสามารถถอดจัมเปอร์ออกได้

โคดตัวอย่างนี้เราจะทดลองอัดเสียงเป็นเวลา 5 วินาที แล้วเล่นเสียงเป็นเวลาอีก 5 วินาที ทำได้โดยจ่ายไฟสัญญาณ 1 ที่ขา REC เป็นเวลา 5 วินาที

1. ต่อวงจร ISD1820 ดังนี้

VCC - 5V

GND - GND

REC - 3

PLAYE – 2

2. อัพโหลดโคดตัวอย่าง Arduino ISD1820 นี้แล้วดูผลลัพธ์

ISD1820-Arduino.ino

```
// Example By ArduinoAll
// VCC - 5V
// PLAYE - 2
// REC - 3
 void setup() {
pinMode(2, OUTPUT); // ต่อกับขา PLAYE
pinMode(3, OUTPUT); // ต่อกับขา REC
Serial.begin(9600);
}
```

```
 void loop() {
Serial.println("Recoarding 5 sec...");
digitalWrite(3, 1); // กดปุ่ม REC
delay(5000);
digitalWrite(3, 0); // ปล่อยปุ่ม REC
Serial.println("Play Sound");
digitalWrite(2, 1); // กดปุ่ม PLAYE
digitalWrite(2, 0); // ปล่อยปุ่ม PLAYE
delay(10000);
}
```

3. ผลทดลองอัดเสียง Arduino ISD1820

วันนี้ทางร้านจะมาแนะนำ Module ISD1820 เป็น Module ราคาถูกสำหรับเล่นเสียงเพลง อัดเสียงได้มีปุ่มกดสำหรับเล่นเสียงและอัดเสียง ที่สำคัญราคาถูกมาก เหมาะสำหรับนัก diy ที่ต้องการใช้งานเกี่ยวกับทางด้านเสียงพัฒนาโปรเจคได้ เช่น ถังขยะพูดได้ ข้อความต้อนรับเมื่อมีคนเปิดประตูเป็นต้น มาเริ่มกันเลย

การใช้งาน ISD1820 บันทึกเสียง ต่อดังนี้

VCC - 5V

GND - GND

Speaker - ต่อลำโพง 0.5W หรือต่อเป็น input ของเครื่องขยายเสียง

กดปุ่ม REC เพื่อบันทึกเสียงจากไมค์บนโมดูล

กดปุ่ม PLAYE เพื่อเล่นจนจบเสียง

กดปุ่ม PLAYL เพื่อเล่นในขณะที่กดปุ่มค้าง

สวิตช์ REPEAT เลื่อนไปที่ ON จะเล่นซ้ำเรื่อย ๆ เลื่อนที่ OFF จะไม่เล่นซ้ำ

สวิตช์ FT สำหรับส่งสัญญาณออกไปลำโพงโดยตรง เลื่อนไปที่ ON จะเป็นเหมือนไมค์ออกลำโพง เลื่อนไปที่ OFF เพื่อใช้งานบันทึกเสียงปกติ

ขา REC , PLAYE , PLAYL ทำงานเมื่อมีสัญญาณ 1 มาทริก Active high สามารถใช้ Arduino ควบคุมสั่งควบคุมการทำงานได้

ช่องจัมเปอร์ P2 สำหรับเปิด/ปิดบอร์ด เสียบจัมเปอร์ต่อขาทั้งสองให้เชื่อมกันเพื่อให้บอร์ทำงาน เอาจัมเปอร์ออกเพื่อปิดการทำงาน

แนะนำให้ตั้งสวิตช์ FT ให้ตั้งไปที่ OFF ตลอด ระหว่างที่เล่นเสียงกลับ ห้ามเลื่อนสวิตช์ FT ไปที่ ON สัญญาณจะชนกัน

การทำงานของตัว Module ISD1820

VCC - ใช้แรงดันแหล่งจ่ายไฟเลี้ยง 3.3V

Gnd - ต่อกับ Gnd ของแหล่งจ่าย

REC - เมื่อป้อน Logic High หรือ กดปุ่ม Module ก็จะทำการอัดเสียงต่างๆเพิ่มที่จะนำไปใช้ในการเล่นเพลง

Play - ใช้ในการเล่นเสียงที่เราบันทึกไว้

Speaker Output - SP+ SP- ใช้ต่อกับลำโพงที่มีขยาดความความต้านทาน 8 Ohm

MIC - เป็น Microphone ใช้สำหรับอัดเสียง

การต่อสายร่วมกับ Arduino

Module ISD1820 Arduino

VCC ----> 3.3

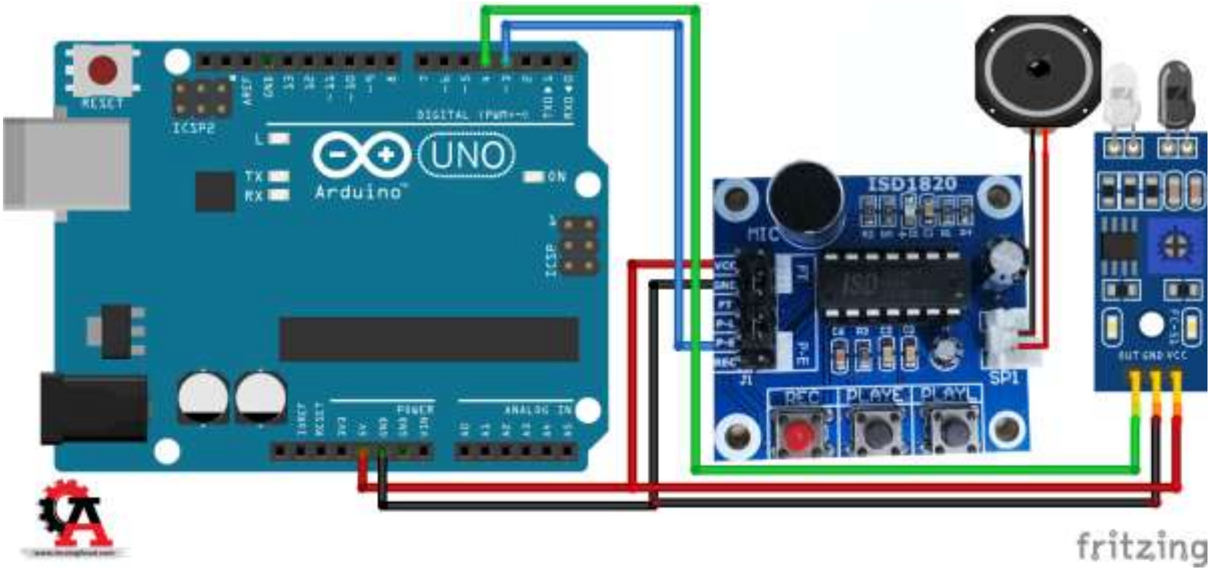
GND ----> GND

FT ----> 5 (D5)

Play L (PL) ----> 4 (D3)

Play E(P-E) ---->3 (D2)

REC ----> 2 (D1)



การต่อใช้งาน

ISD1820 ----> Arduino UNO R3

VCC ----> 5V

GND ----> GND

PLAYE (P-E) ----> 2

REC --> 3

ตัวอย่างโค้ด

```
// Example By ArduinoAll
// VCC - 5V
// PLAYE – 2 (D1)
// REC – 3 (D2)
 void setup() {
pinMode(D1, OUTPUT); // ต่อกับขา PLAYE
```

```
pinMode(D2, OUTPUT); // ต่อกับขา REC
Serial.begin(9600);
}
 void loop() {
Serial.println("Recoarding 5 sec...");
digitalWrite(3, 1); // กดปุ่ม REC
delay(5000);
digitalWrite(3, 0); // ปล่อยปุ่ม REC
Serial.println("Play Sound");
digitalWrite(2, 1); // กดปุ่ม PLAYE
digitalWrite(2, 0); // ปล่อยปุ่ม PLAYE
delay(10000);
}
```

Infrared Module ----> Arduino UNO R3

VCC ----> 5V

GND ----> GND

OUT ----> 3

ตัวอย่างโปรแกรม Code ตัวอย่าง

```
#define REC 2
#define PLAY_E 3
#define FT 5
#define playTime 5000  // เล่นเสียง 5 วินาที
#define recordTime 3000 // บันทึกเสียง 3 วินาทีสามารถบันทึกได้สูงสุด 10 วินาที

void setup()
{
 pinMode(REC,OUTPUT);// set the REC pin as output
 pinMode(PLAY_E,OUTPUT);// set the PLAY_e pin as output
 pinMode(FT,OUTPUT);// set the FT pin as output
 Serial.begin(9600);// set up Serial monitor
}

void loop() {
   while (Serial.available() > 0) {
         char inChar = (char)Serial.read();
           if(inChar =='p' || inChar =='P'){
           digitalWrite(PLAY_E, HIGH);
```

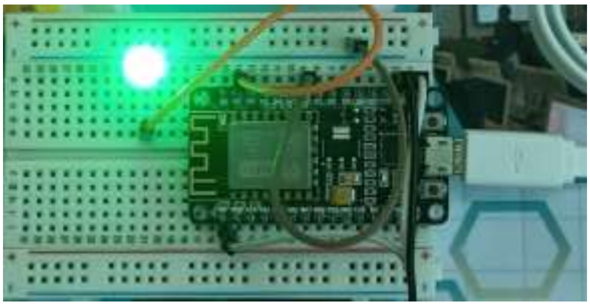
```
            delay(50);
            digitalWrite(PLAY_E, LOW);
              Serial.println("Playbak Started");
            delay(playTime);
              Serial.println("Playbak Ended");
            break;
            }

            else if(inChar =='r' || inChar =='R'){
              digitalWrite(REC, HIGH);
              Serial.println("Recording started");
              delay(recordTime);
              digitalWrite(REC, LOW);
              Serial.println("Recording Stopped ");
            }

      Serial.println("###Serial Monitor Exited");
   }// wihile
Serial.println("### Enter r to record, p to play");
 delay(500);
}
```

หลักการทำงาน เมื่ออัพโหลดโปรแกรมแล้วให้เรา เปิดหน้า Serial monitor ขึ้นมา จากนั้นให้พิมพ์ตัว P ลงไป Module จะทำการเล่นเสียงที่ได้บันทึกไว้เป็นเวลา 5 วินาที และพิมพ์ตัว R เพื่อทำการบันทึกเสียง 3 วินาที

วิธีการนำไปใช้งาน อาจจะดัดแปลงจากโปรแกรมในส่วนที่เล่นเพลง เป็นการกดปุ่ม หรือ ใช้ Sensor มาดัดแปลงเป็นโปรเจคเพิ่มเติมก็ได้เช่นกัน

แอพพลิเคชั่น Blynk

การเชื่อมต่อไวไฟ ด้วย ESP8266 ให้แสดง IP และ Mac Address ออกมา พร้อมกับให้หลอดไฟติดเมื่อเชื่อมต่อสำเร็จ



คำสั่ง

```
#include <ESP8266WiFi.h>
const char* ssid     = "@BuakhaoWiFi";
const char* password = "";
int led1 =D1;
void setup() {
  pinMode (led1, OUTPUT);
  Serial.begin(115200);
  delay(10);  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
   delay(500);
    Serial.print(".");
    digitalWrite (led1, HIGH);
    delay(1000);
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  Serial.print("MAC address: ");
  Serial.println(WiFi.macAddress());
```

```
}
void loop() {
  delay(100);
}
```

การเชื่อมต่อไวไฟ ผ่าน Blynk เพื่อควบคุมหลอดไฟ

รายการ

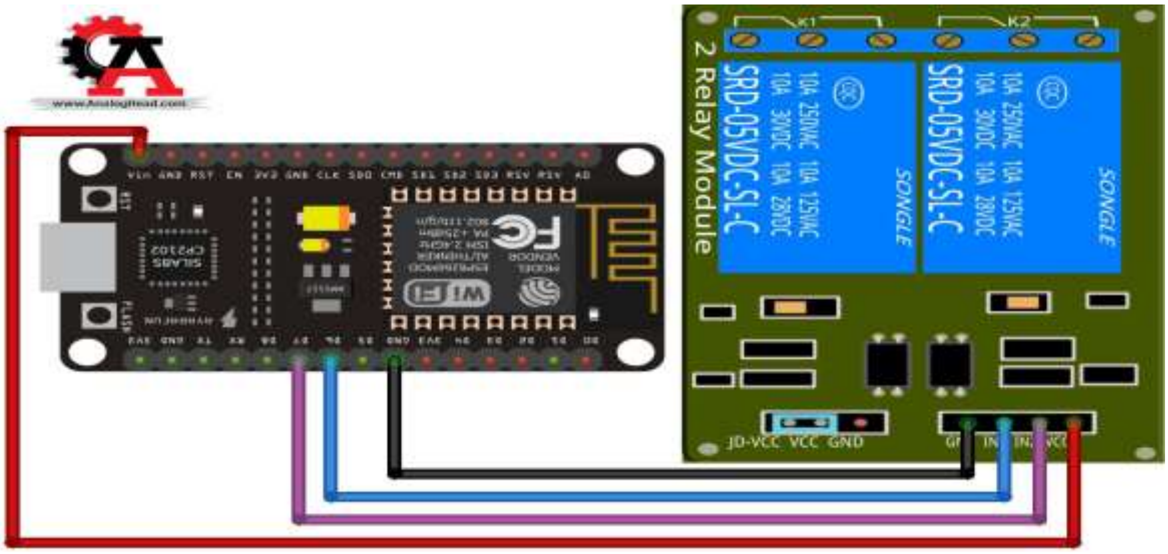
ESP8266 Relay 5V 2 cannel

การต่อวงจร

Relay ----> ESP8266

Gnd ----> Gnd IN1 ----> D6 IN2 ----> D7

Vcc ----> Vin




```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = "xxxx";  //token
char ssid[] = "xxxx";  //ชื่อไวไฟ
char pass[] = "xxxx";  //รหัสไวไฟ
//กำหนดขาrelay
#define lamp1 D6
#define lamp2 D7
#define BLYNK_Green     "#23C48E"
#define BLYNK_Blue      "#04C0F8"
#define BLYNK_Red       "#D3435C"
#define BLYNK_White     "#ffffff"
BlynkTimer timer;
WidgetLED led1(V10);
```

```
WidgetLED led2(V11);
BLYNK_CONNECTED() {
  Blynk.syncAll();
}
BLYNK_WRITE(V1)
{
  int statuslamp = 0;
  statuslamp = param.asInt();
  Serial.println("ST1="+String(statuslamp));
  if(statuslamp==true){
    led1.on();
    pinMode(lamp1,HIGH);
  }
  else{
    led1.off();
    pinMode(lamp1,LOW);
  }
}
BLYNK_WRITE(V2)
{
  int statuslamp = 0;
  statuslamp = param.asInt();
  Serial.println("ST2="+String(statuslamp));
  if(statuslamp==true){
    led2.on();
    pinMode(lamp2,HIGH);
  }
  else{
    led2.off();
    pinMode(lamp2,LOW);
  }
}
void led_blink() {
  digitalWrite(2,!digitalRead(2));
}
```

```
void setup() {
  Serial.begin(115200);
  pinMode(2,OUTPUT);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(1000L, led_blink);
  Serial.println();
  pinMode(lamp1,OUTPUT);
  pinMode(lamp2,OUTPUT);
  pinMode(lamp1,LOW);
  pinMode(lamp2,LOW);
}
void loop() {
  Blynk.run();
  timer.run();
}
```

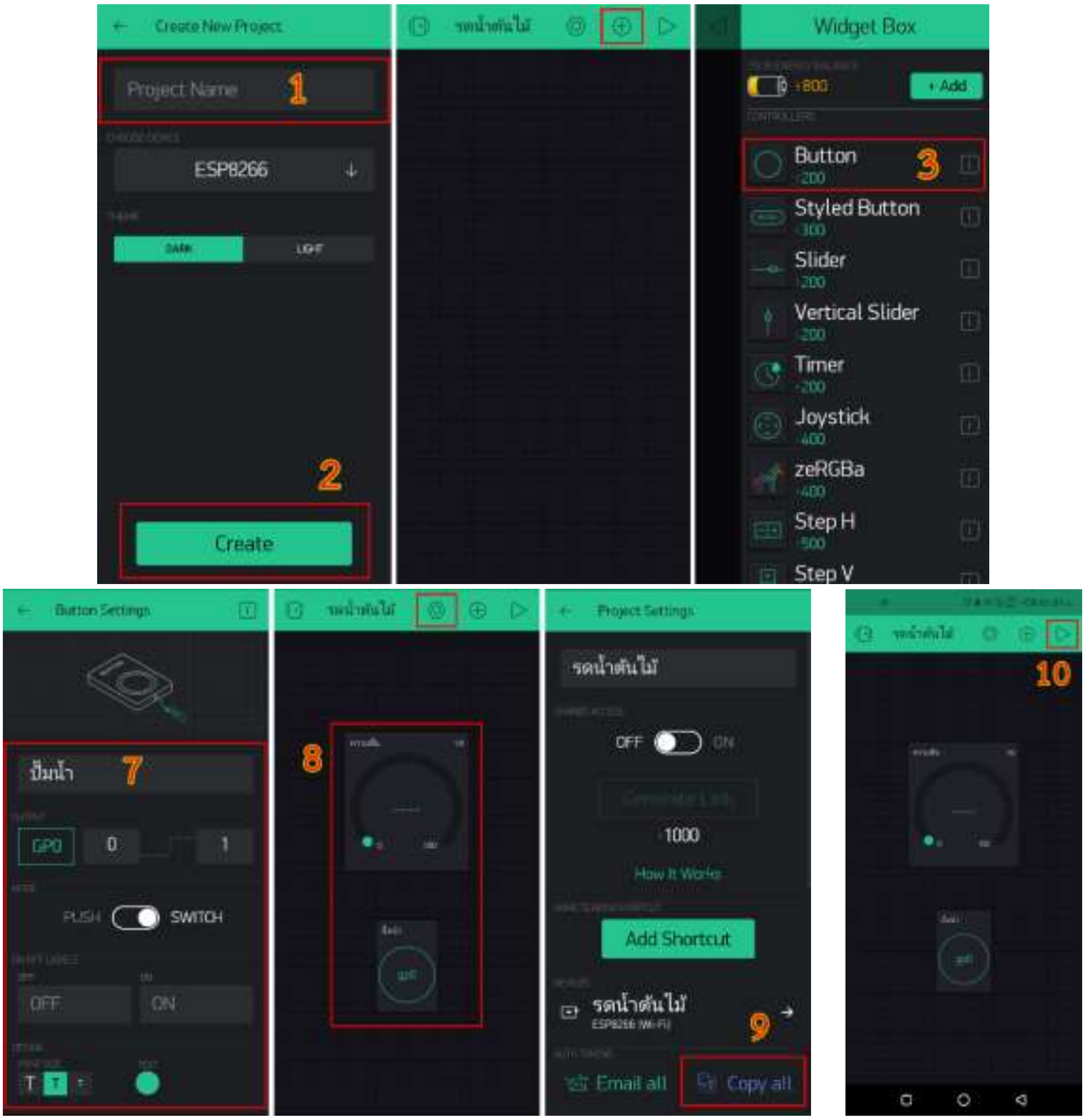
Smart IoT ESP8266 ควบคุมปั๊มน้ำ และตรวจเช็คความชื้นในดิน ด้วยแอพ Blynk บนมือถือ

รายการอุปกรณ์ที่ต้องใช้

- บอร์ด Shield Nodemcu V2 Relay 4ch
- Soil Moisture Sensor Module (วัดความชื่นในดิน)
- Pump Water DC ปั้มน้ำ DC 6-12V
- สาย Jumper เมีย + เมีย

ตัวอย่างโปรแกรม

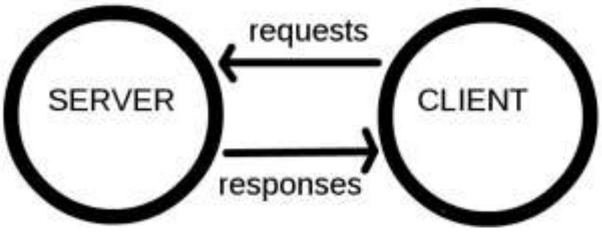
```
// www.ec-bot.com
// IOT ESP8266 สั่งซื้อได้ที่ http://www.ec-bot.com/p/244
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = "XXX"; // Token App Blynk
char ssid[] = "XXX"; // ชื่อ Wi-Fi
char pass[] = "XXX"; // รหัส Wi-Fi
int SR = 0; // ประกาศตัวแปร SR มีค่าเท่ากับ 0
int temp = 0; // ประกาศตัวแปร temp มีค่าเท่ากับ 0
BLYNK_READ(V0)
{
  Blynk.virtualWrite(V0, temp); // แสดงค่า Temp ออกช่องทาง V0
}
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass); // เชื่อมต่อ auth , ssis , pass
}
void loop()
{
  Blynk.run();
  SR = analogRead(A0); // ให้ตัวแปล SR อ่านค่าจาก Pin A0
  temp = map(SR, 1023, 0, 0, 100); // ตัวแปล Temp คือ แปลงค่าของตัวแปร SR จาก 1023 ถึง 0 เป็น
0 ถึง 100
  delay(50);
}
```



App สำเร็จรูป Blynk Nodemcu esp8266 (ตอนที่ 1 Blynk คืออะไร)

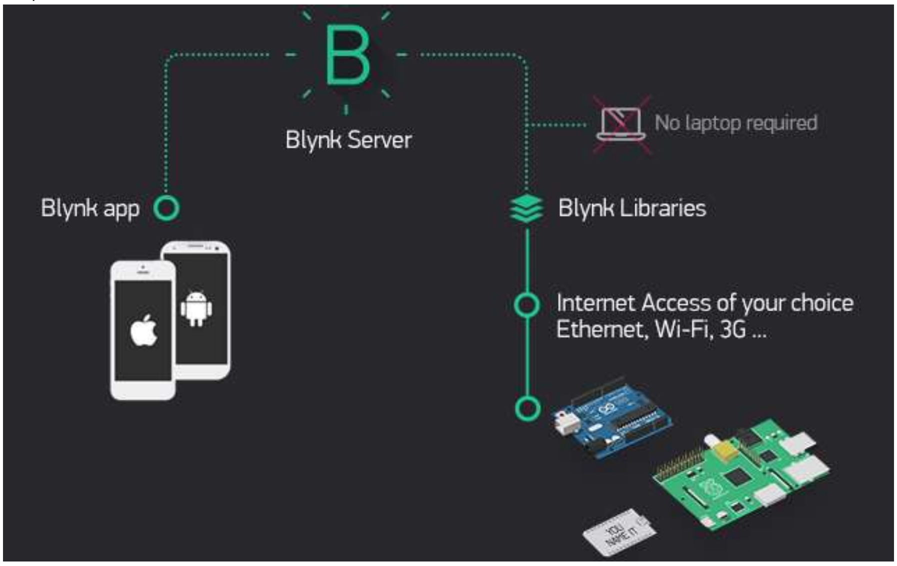
Blynk คือ Application สำเร็จรูปสำหรับงาน IOT มีความน่าสนใจคือการเขียนโปรแกรมที่ง่าย ไม่ต้องเขียน App เองสามารถใช้งานได้อย่าง Real time สามารถเชื่อมต่อ Device ต่างๆเข้ากับ Internet ได้อย่างง่ายดาย ไม่ว่าจะเป็น Arduino, Esp8266, Esp32, Nodemcu,Rasberry pi นำมาแสดงบน Application ได้อย่างง่ายดาย แล้วที่สำคัญ Application Blynk ยังฟรี และ รองรับในระบบ IOS และ Android อีกด้วย เป็นอะไรที่น่าสนใจมากๆใช่ไหมครับ คราวนี้เรามาเริ่มกันเลย

ในยุคสมัยก่อน การเขียนโปรแกรมเชื่อมต่อกันระหว่าง อุปกรณ์ 2 ชิ้นเข้าด้วยกันมักจะใช้งานในลักษณะของ Server >>> Client ทำให้เกิดข้อจำกัดต่างๆมากมาย ยกตัวอย่าง เราต้องการเปิดปิดไฟ ผ่านหน้าเว็บ เราก็จะให้ Arduino เป็น Server และ เครื่องคอมพิวเตอร์ (Client) เป็นเครื่องลูก ข้อจำกัดที่เกิดขึ้นคือทรัพยากร เช่น CPU RAM ROM ของเราอาจจะไม่พอ มักจะเจอปัญหาเอ๋อบ่อย ค้างไปดื้อๆ ก็มี ทำให้การเขียนโปรแกรมเป็นไปได้ยากต้องประหยัดทรัพยากรให้ได้มากที่สุดเพื่อจะให้สามารถทำงานได้ และการเซ็ต Netword เป็นไปได้ยาก ส่วนใหญ่มักจะใช้ในวง Lan หรือถ้าต้องการ ควบคุบผ่าน Wan จะต้อง Forword Set ระบบ Network จนปวดหัว



ภาพที่ 2 การเชื่อมต่อแบบ Server to Client

ต่อมาเป็นยุคของ Cloud เกิดขึ้น บวกกับมี Chip Wifi ราคาถูก Esp8266 ถูกผลิตขึ้นมา แต่ด้วยข้อจำกัดทางด้านทรัพยากร จึงมือวิธีการคิดว่า ถ้านำข้อมูลไปใส่ลงใน Server เลยละแล้วให้ Device ของเราเรียกเข้าไปแก้ไข หรืออ่านข้อมูลโดยตรง ทำให้ความฉลาดของตัวอุปกรณ์ของเราไม่มีวันสิ่นสุดหมดข้อจำกัดหลายอย่าง Device กลายเป็นแค่ตัวรับ Data และส่ง Data มาแสดงเท่านั้น ทำให้ Chip Esp8266 จึงได้รับความนิยมในปัจจุบัน



ภาพที่ 3 ภาพรวมของระบบ Network Blynk

วิธีการทำงานของ Blynk เริ่มจาก อุปกรณ์ เช่น Arduino esp8266 Esp32 Rasberry Pi เชื่อมต่อไปยัง Server ของ Blynk โดยตรง สามารถรับส่งข้อมูลหากันได้

คอมพิวเตอร์ Smartphone ก็จะเชื่อมต่อกับ Server ของ Blynk โดยตรง กลายเป็นว่า มี Server เป็นสะพานให้เชื่อต่อหากันจึงหมดปัญหาและข้อจำกัดทุกอย่างทำให้อุปกรณ์ของเรามีความฉลาดมากขึ้น

การออกแบบในลักษณะ ภาพที่ 3 เป็นที่นิยมมากในปัจจุบัน เพราะไม่จำเป็นต้อง Set อุปกรณ์ Network ต่างๆ ให้ปวดหัว

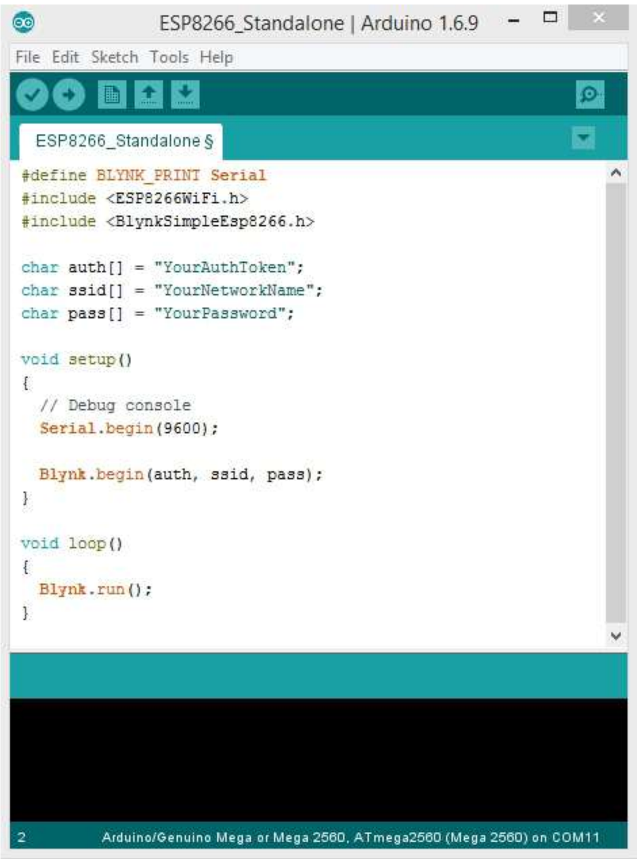
พูดถึงระบบไปแล้วคราวนี้เราจะมาดูความสามารถของ Application Blynk ดูบ้างว่าสามารถทำอะไรได้บ้าง

ภาพที่ 4 ข้อมูลหน้าจอที่จะมาแสดงใน App ของเรา

จากภาพที่ 4 เราสามารถเลือก หน้าจอของภาพ คำอธิบาย เกจ์วัดต่างๆ ก็สามารถออกแบบได้เองได้อย่างอิสระอีกด้วย ต้องการอะไรไม่ต้องการอะไรเราสามารถเลือกได้ตาทความของเราได้เลย

ภาพที่ 5 หน้าจอของ App ที่เราได้ทำการสร้างขึ้นมา

จากภาพที่ 5 เราสามารถจับลากจัดเลียงปรับขนาดให้เหมาะสมตรงตามความต้องการของเรา



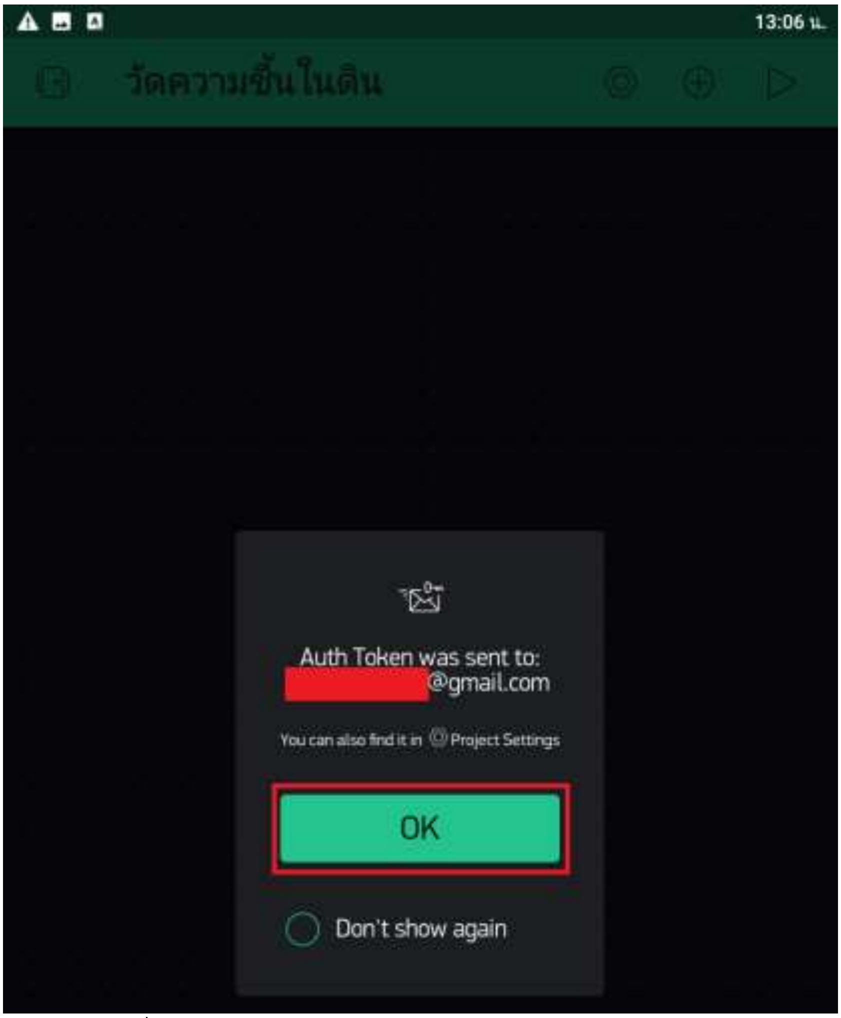
ภาพที่ 6 เขียนโปรแกรมง่ายนิดเดียว

ขั้นตอนการใช้งานแอพ Blynk

1. เมื่อเข้าแอพ Blynk และทำการ Login เข้ามาเรียบร้อยแล้ว ให้กด New Project จากนั้นทำตามขั้นตอนดังรูป

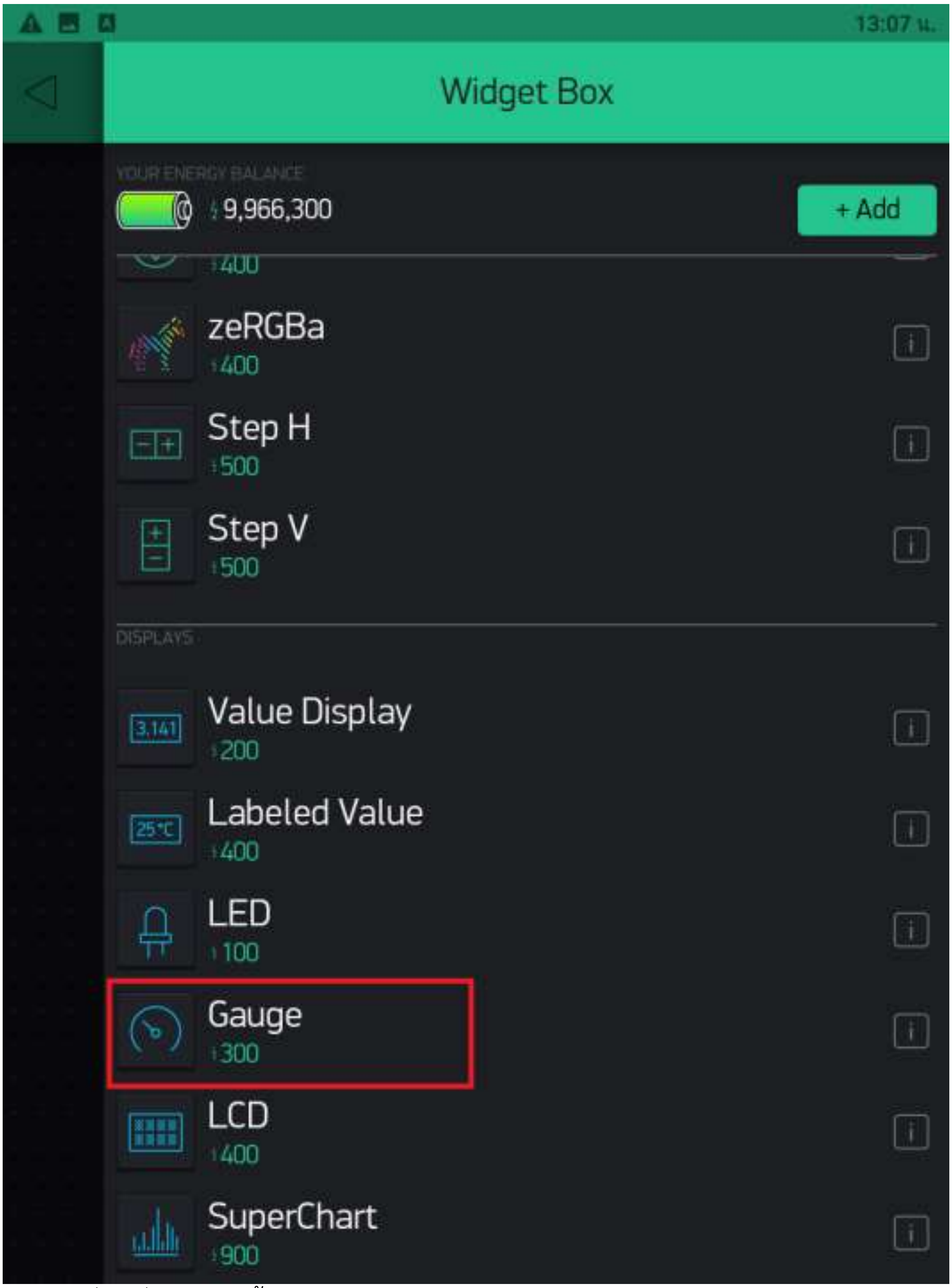
1. ตั้งชื่อโปรเจค (วัดความชื่นในดิน)
2. เลือกบอร์ด (ESP8266)
3. เลือกไวไฟ (Wi-Fi)
4. เลือกธีมตามใจชอบ (DARK, LIGHT)
5. กด Create (Create)

2. แอพจะส่ง Token ไปยังอีเมลล์ของเรา หลังจากนั้นกด OK ***หมายเหตุ* “เดี๋ยวเราจะนำ Token นี้ไปใส่ในโค้ด Arduino IDE ของเรา เดี๋ยวผมจะสอนในขั้นตอนท้าย ๆ ครับ”**

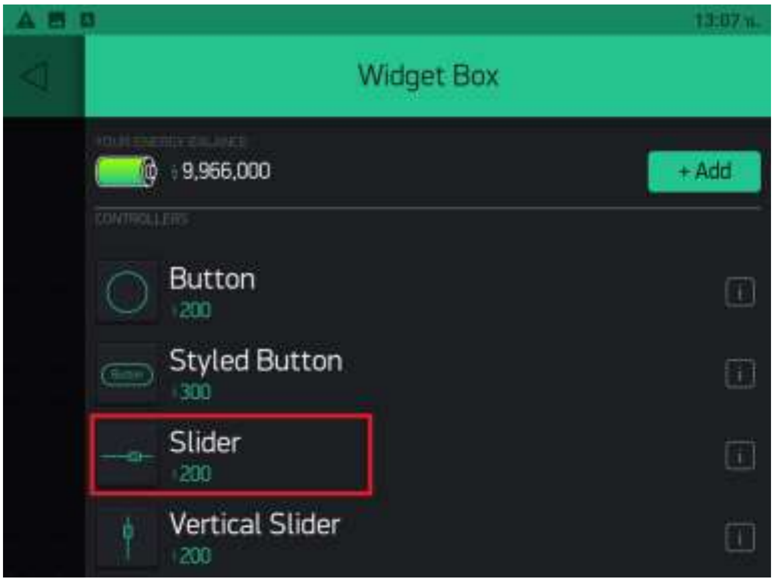


3. กดตรงเครื่องหมาย **บวก** ดังรูป

4. จะเจอหน้าต่างแถบเครื่องมือต่าง ๆ มากมาย ในที่นี้เราจะเลือกเครื่องมือ **“Gauge”** ดังรูป



5. คลิกเพิ่มเครื่องมืออีกครั้ง และเลือกเป็น “**Slider**” ดังรูป

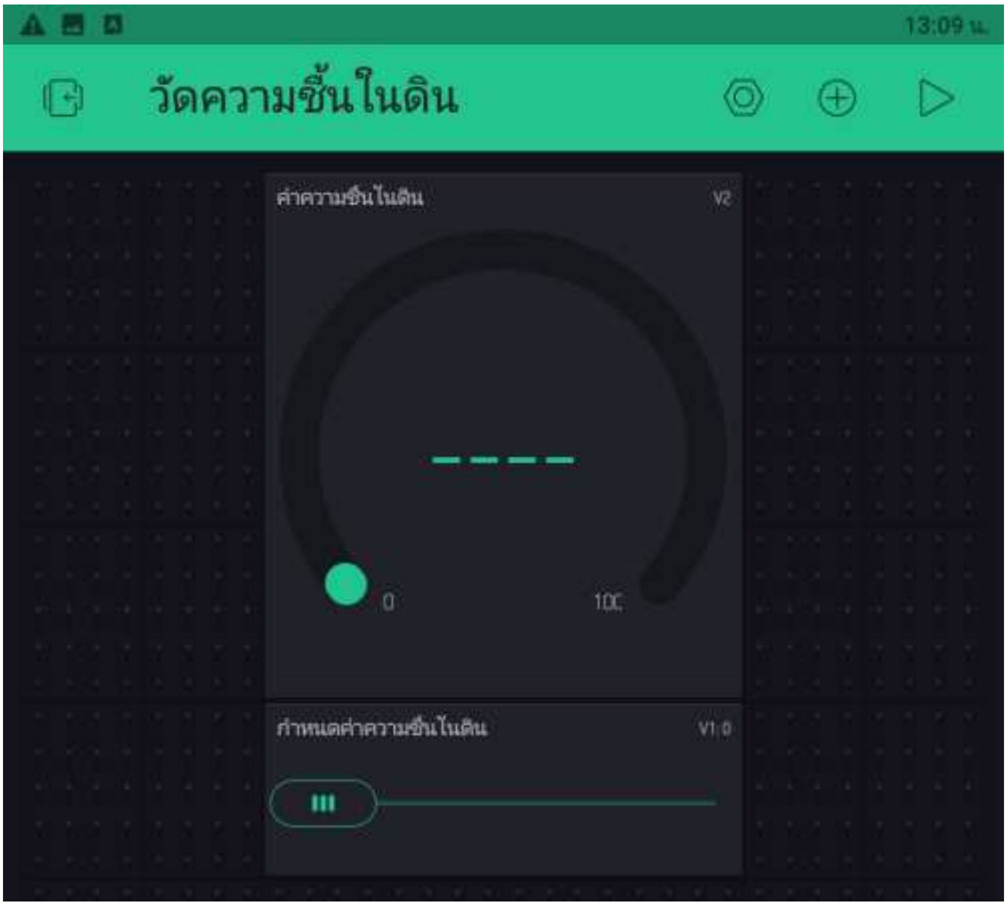


6. คลิกที่ตัว **Gauge** หนึ่งครั้ง จะเจอหน้าต่างให้เซ็ทค่าตาม ดังรูป

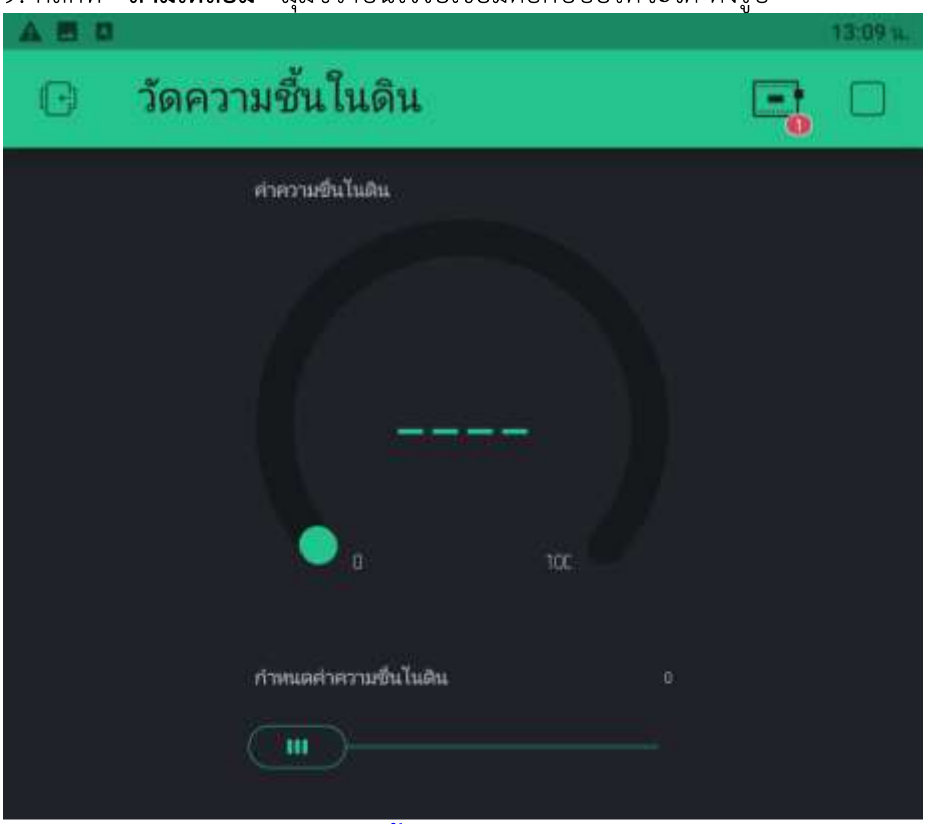
1. ใส่ชื่อ (ค่าความชื้นในดิน)
2. เลือกเป็น V2
3. ใส่ 100
4. ใส่ pin (pin.#/%)

7. คลิกที่ตัว **Slider** หนึ่งครั้ง จะเจอหน้าต่างให้เซ็ทค่าตาม ดังรูป

8. เมื่อเซ็ทค่าครบทุกอันแล้วเราจะได้หน้าตาแอพ ดังรูป



9. คลิกที่ **“สามเหลี่ยม”** มุมขวาบนไว้รอเชื่อมต่อกับบอร์ดจะได้ ดังรูป



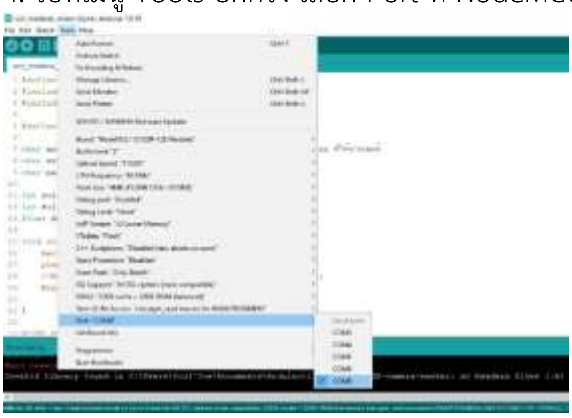
ขั้นตอนในโปรแกรม Arduino IDE

1. ไปที่อีเมลของเรา หาเมลที่ชื่อ Blynk จะเห็น Token ที่ได้มา ดังรูป และทำการคัดลอกไว้

2. ดาวน์โหลดโค้ดที่ผมอัพไว้ให้เมื่อข้างต้น และเปิดไฟล์ขึ้นมา นำ Token ที่ได้ไปวาง ดังรูป และใส่ชื่อ ไวไฟและรหัสให้เรียบร้อย

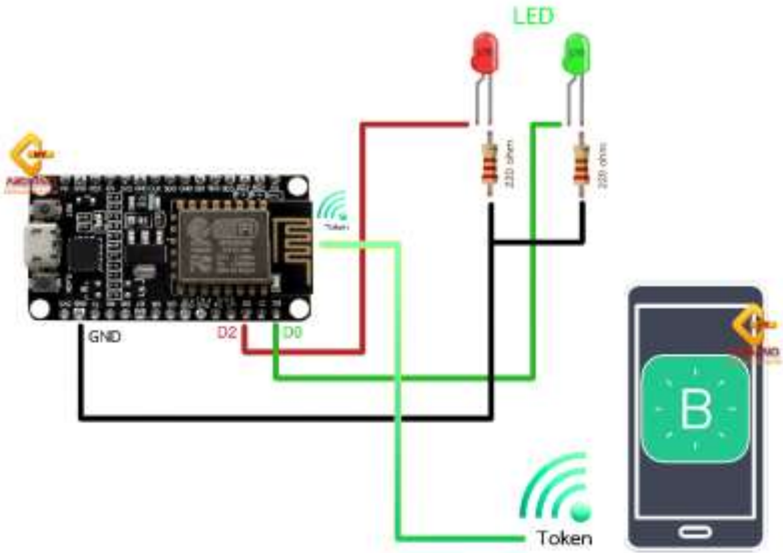
3. หลังจากนั้นให้เข้าไปเมนู Tools ⟶ Board : “xxxxxxx” ⟶ ESP8266 Board (3.0.2) ⟶ เลือกรุ่นของบอร์ด

4. ไปที่เมนู Tools อีกครั้ง เลือก Port ที่ NodeMCU ESP8266 เชื่อมต่ออยู่ ในที่นี้เป็น COM8



5. กดอัปโหลดโปรแกรมลงบอร์ด รอจนเสร็จ

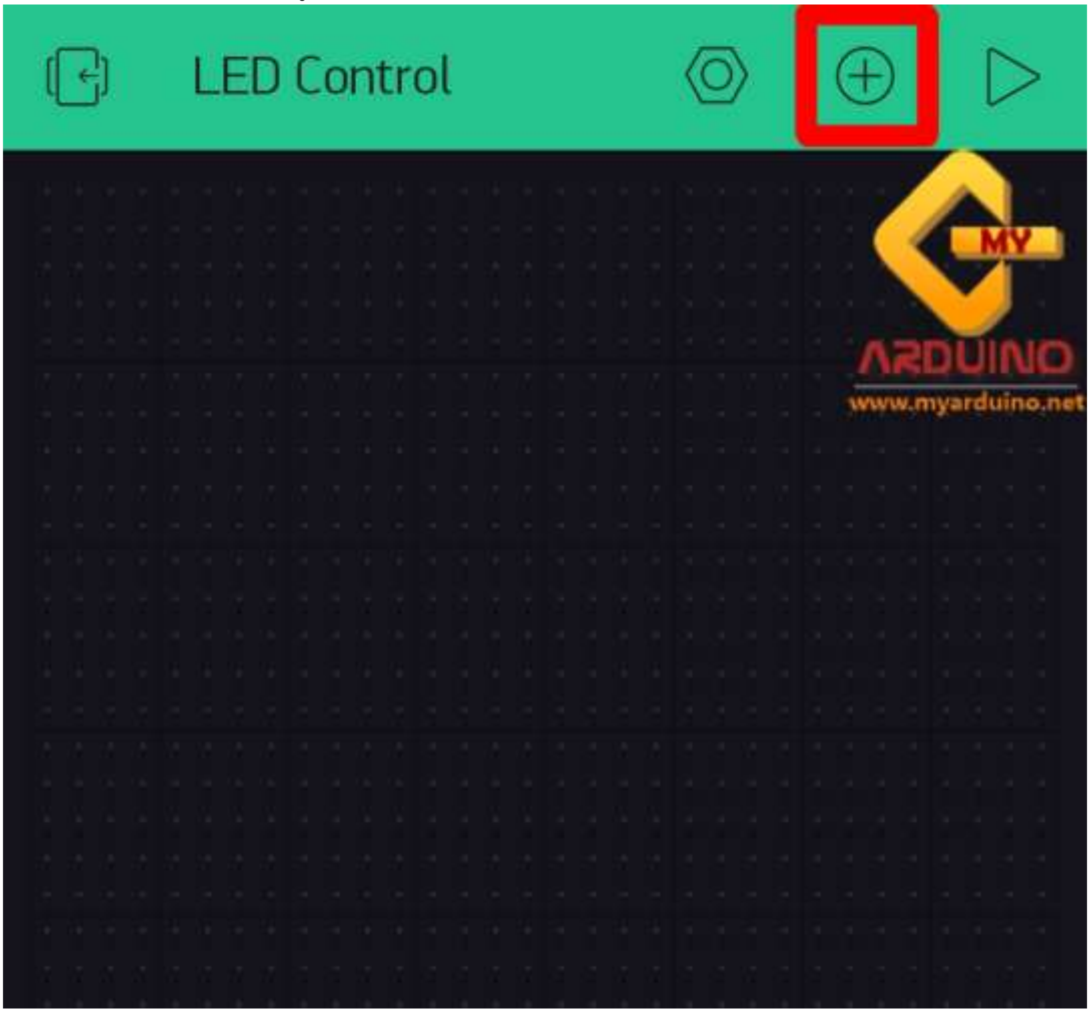
NodeMCU Wifi Blynk Button ควบคุมเปิดปิดไฟ LED



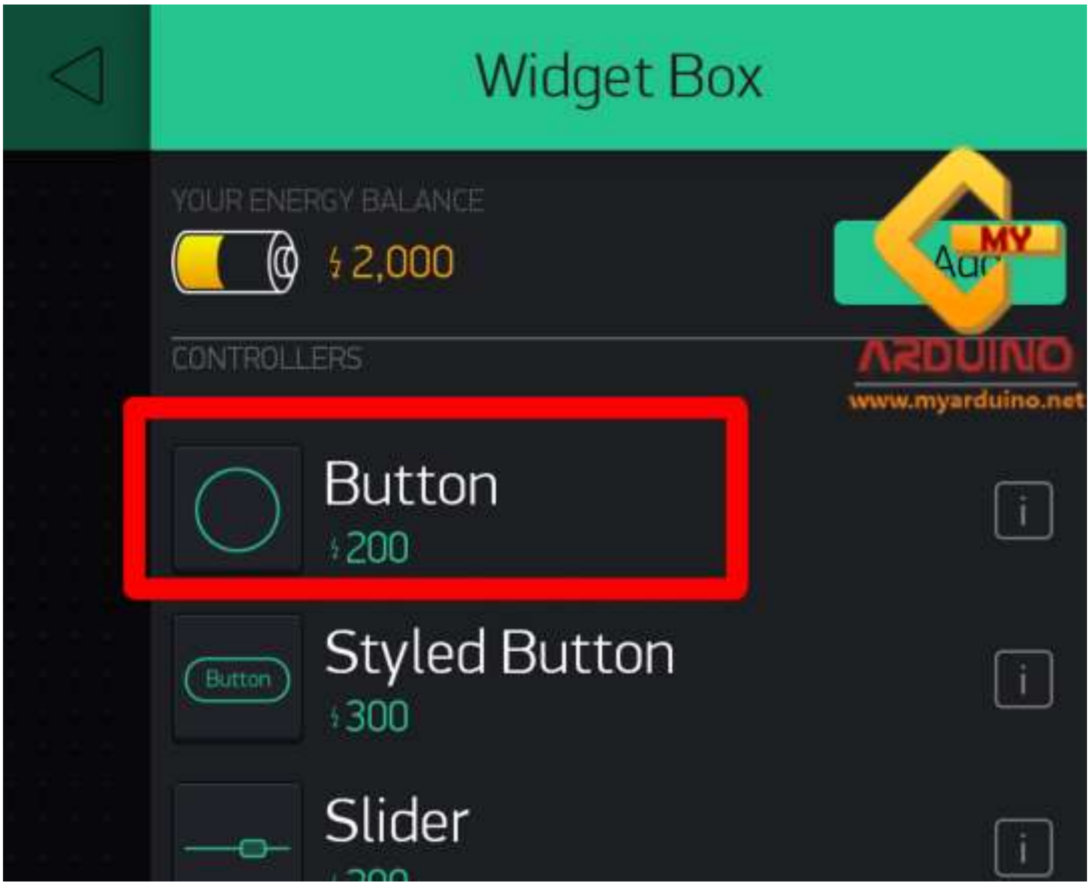
NodeMCU ESP8266 -> LED

- D0 -> LED
- D2 -> LED

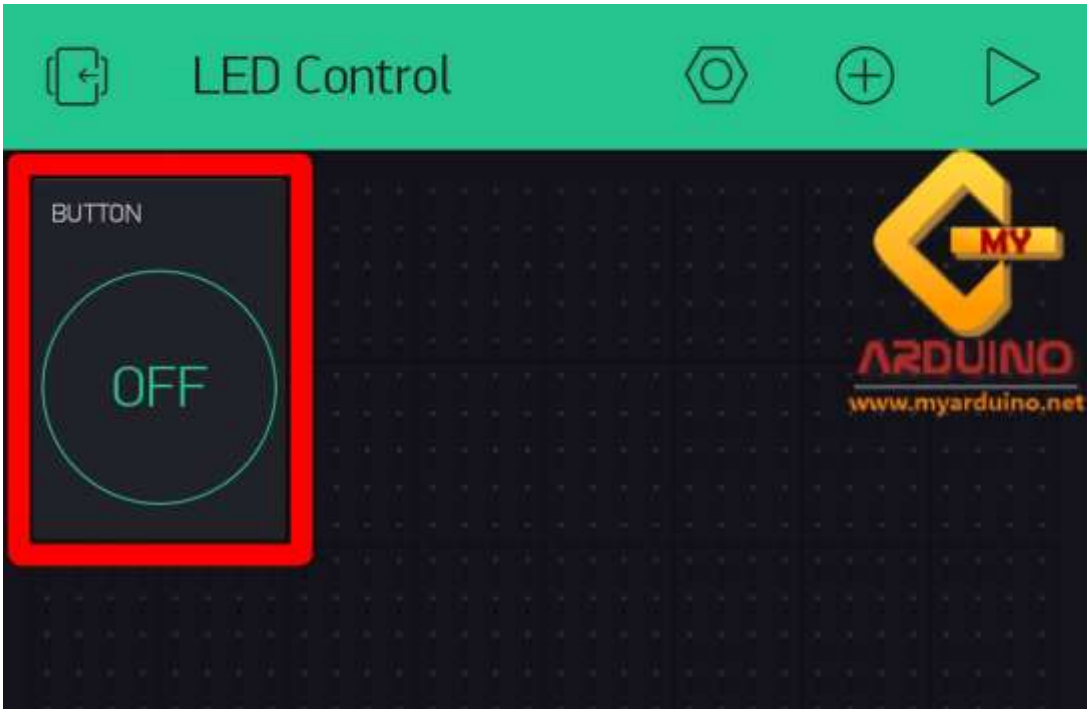
2. เมื่อทำการติดตั้ง **Library, แอพฯ ของ** Blynk และเชื่อมต่อเครือข่ายระหว่าง Blynk และมือถือ เรียบร้อยแล้ว ให้เลือกไปที่เครื่องมือรูปเครื่องหมายบวกในหน้าจอแอพ Blynk



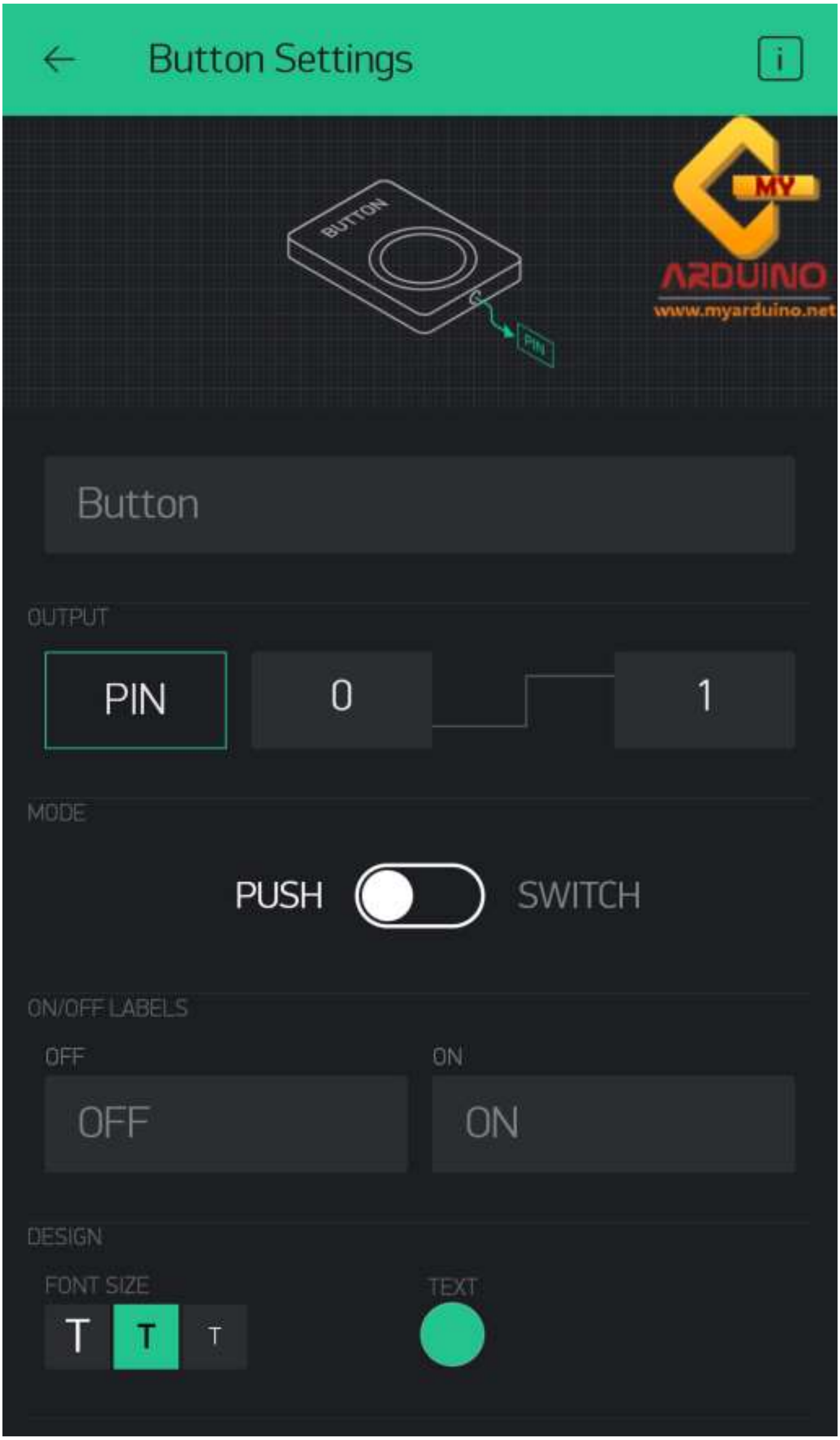
3.จะเห็นได้ว่าจะมี **Widget** ชุดคำสั่งต่างๆปรากฏขึ้นมาให้เลือกมากมาย ในที่นี้ทางเราเลือก **Widget Button** มาเป็นตัวอย่าง



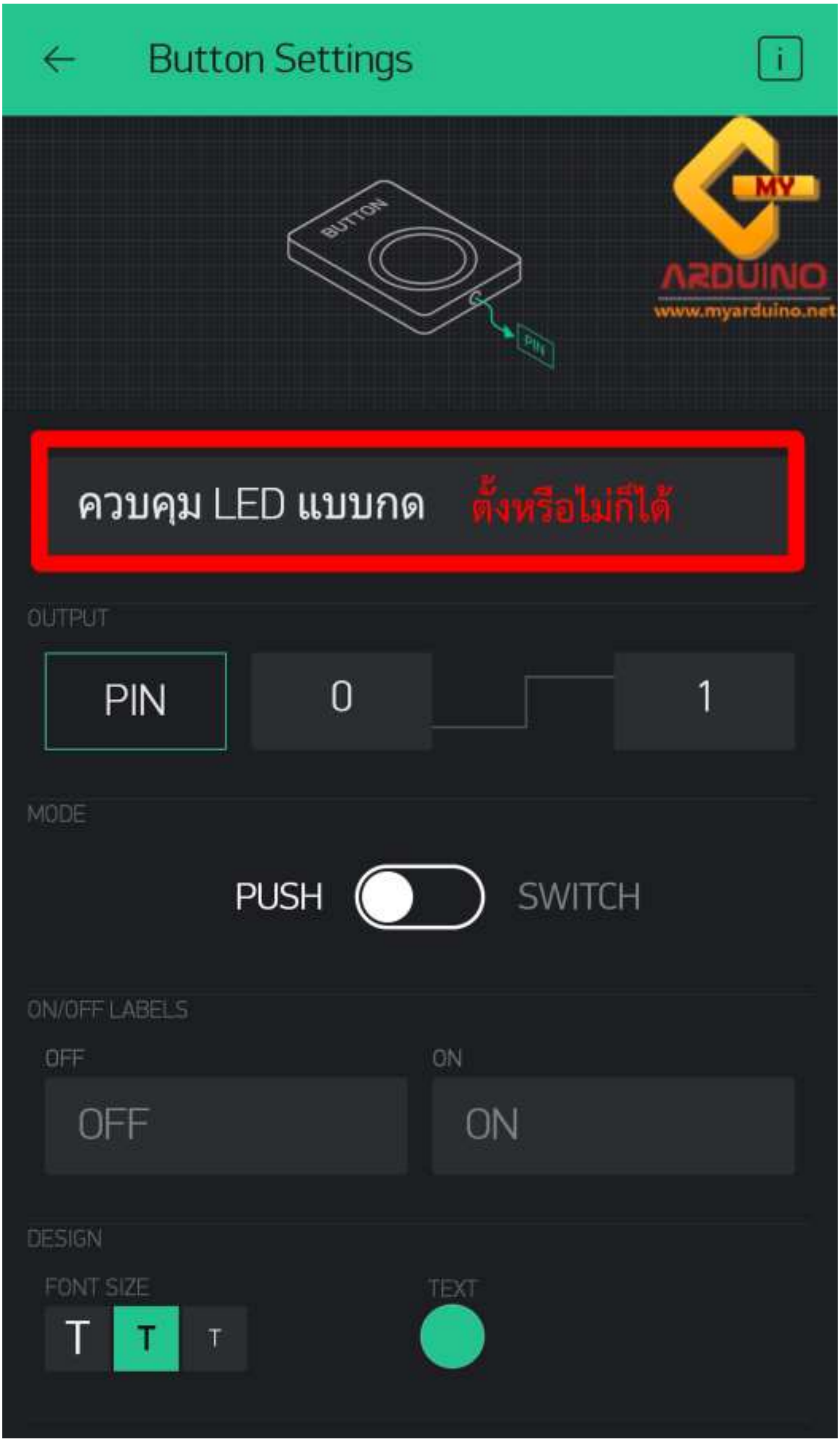
4.เมื่อทำการเลือก **Widget Button** มาแล้วให้กดเข้าไปที่ Widget



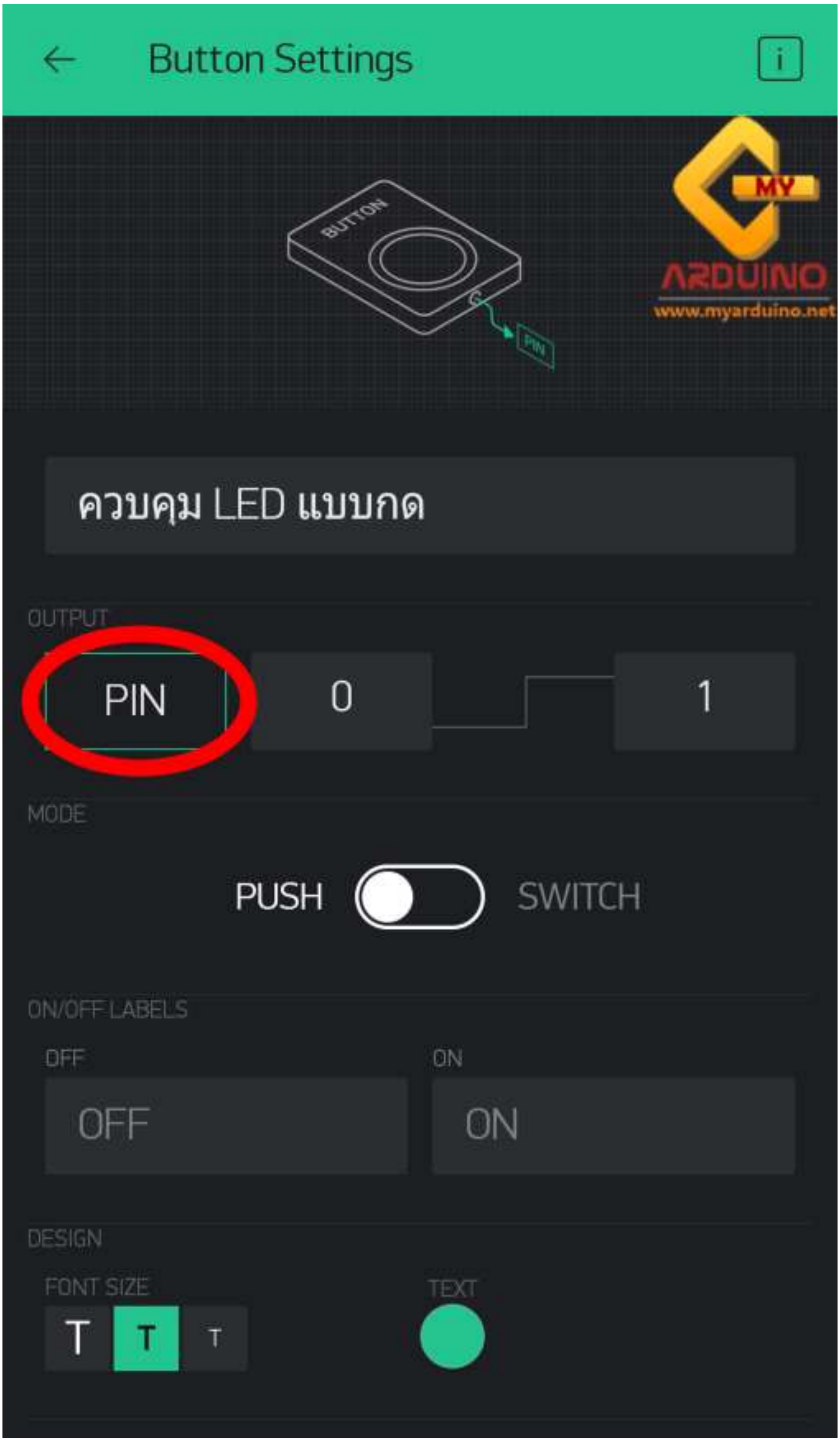
5.เมื่อเข้ามาแล้วจะพบกับหน้า **Setting ของ Widget**



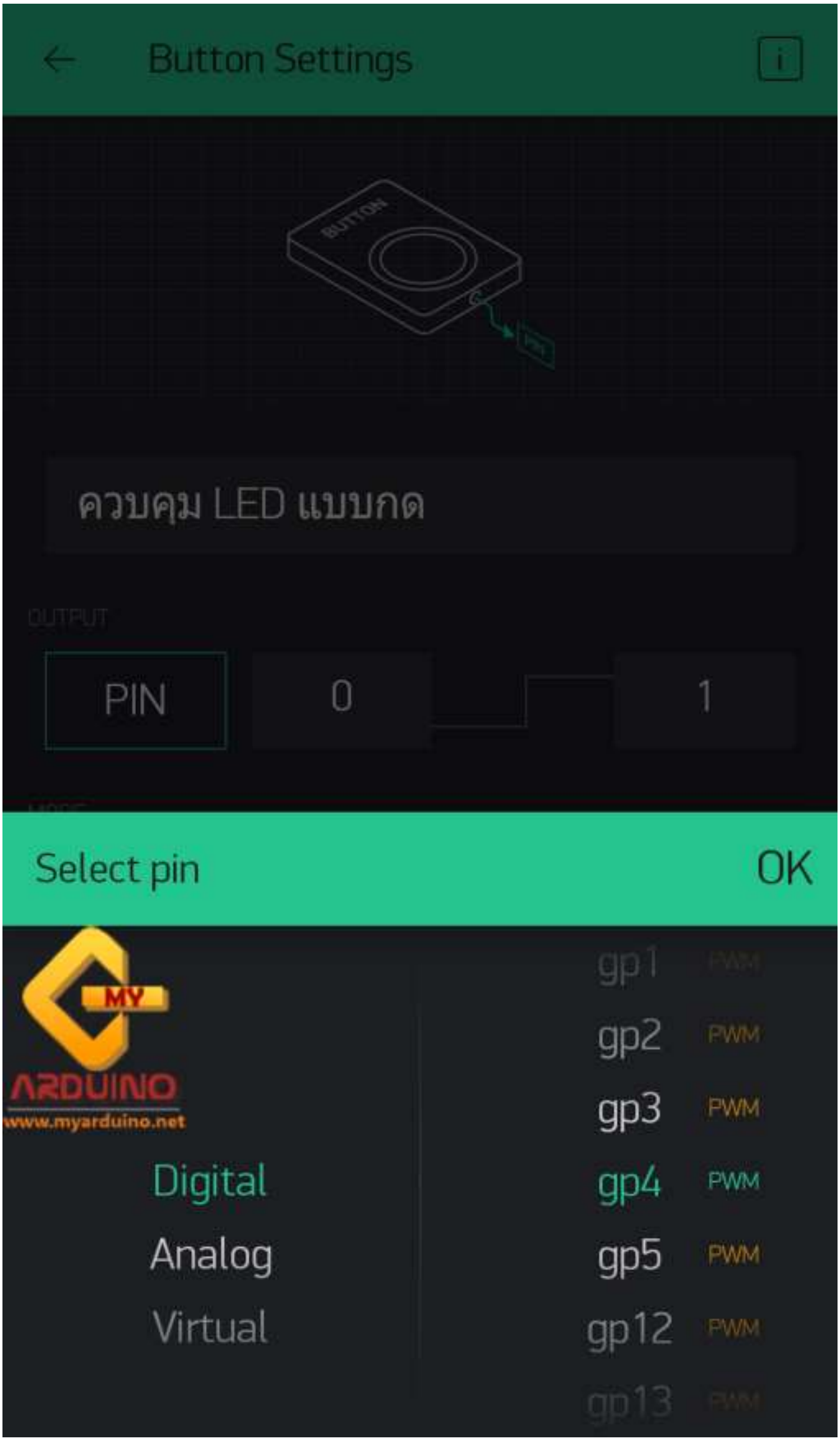
6.จากนั้นให้ทำการตั้งชื่อ **Widget** (จะตั้งหรือไม่ตั้งก็ได้)



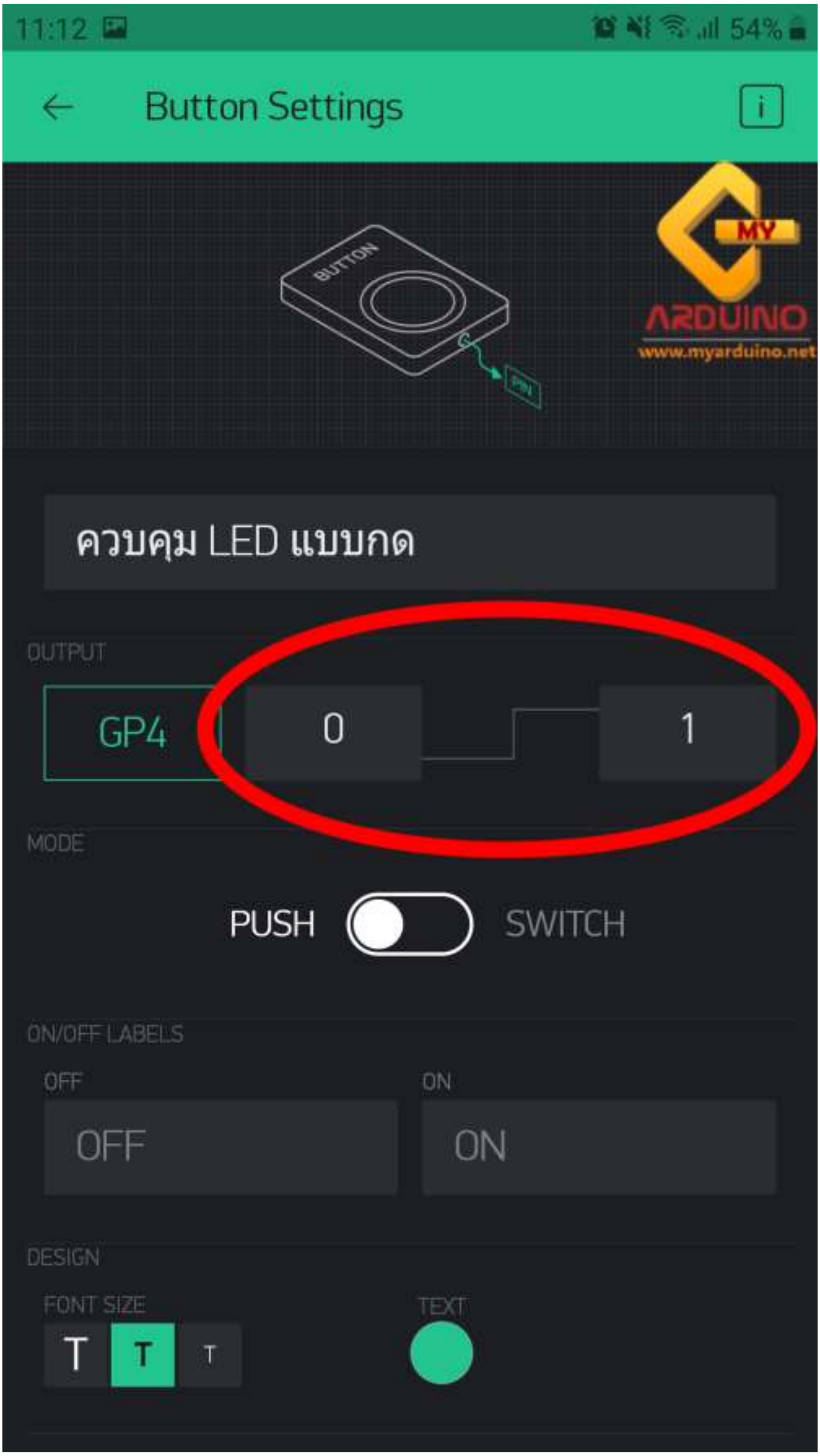
7.จากนั้นกดไปที่เมนู PIN



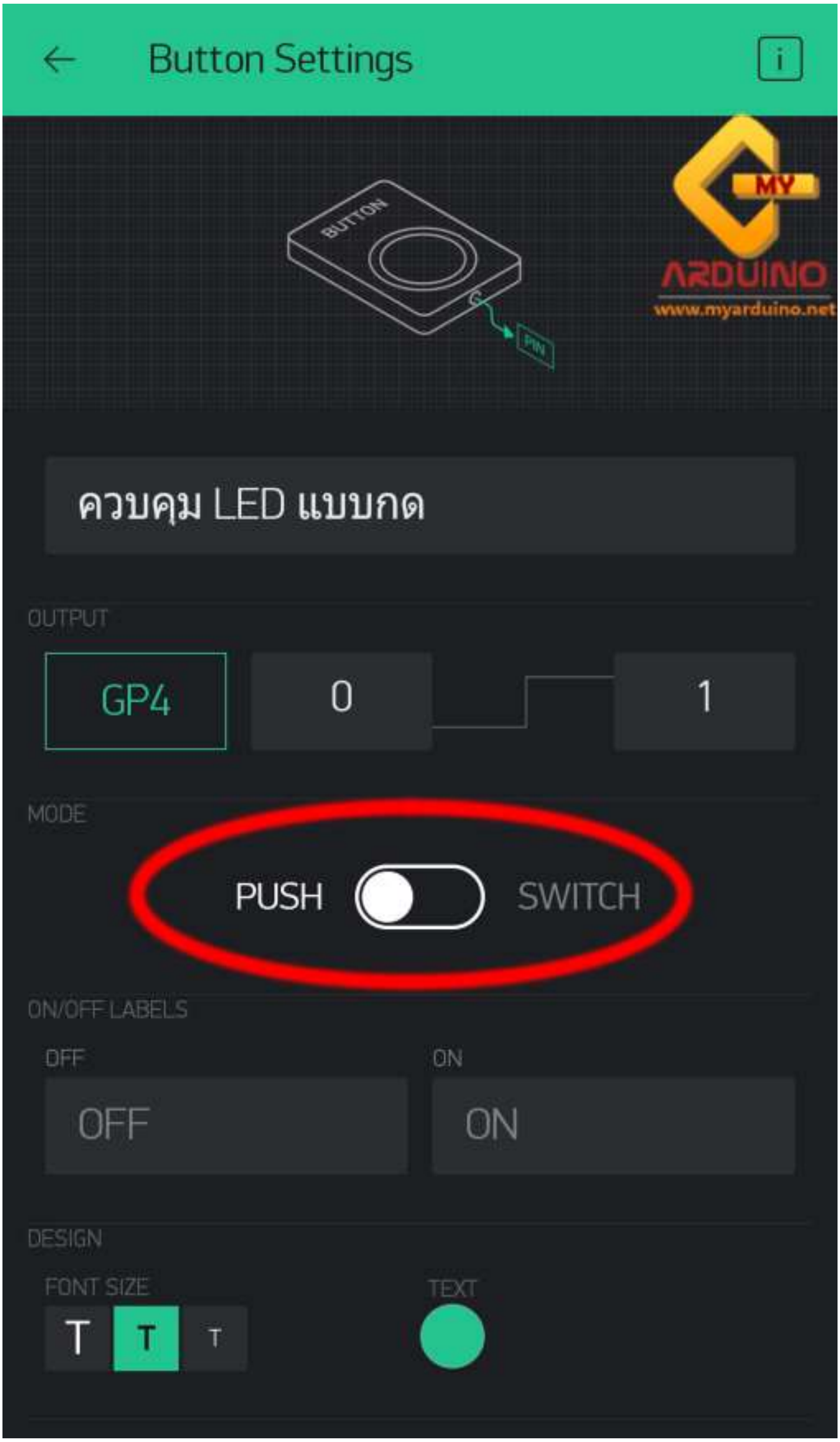
8.เมื่อเข้ามาที่เมนู **PIN** แล้วให้เลือก **GPIO ของขา PIN** บนบอร์ด ที่จะใช้ส่งสัญญาณข้อมูลให้กับ **LED ตัวที่ 1** และกด OK (ในที่นี้ทางเราเชื่อมต่อกับขา D2 หมายเลข **GPIO คือ GP4**)



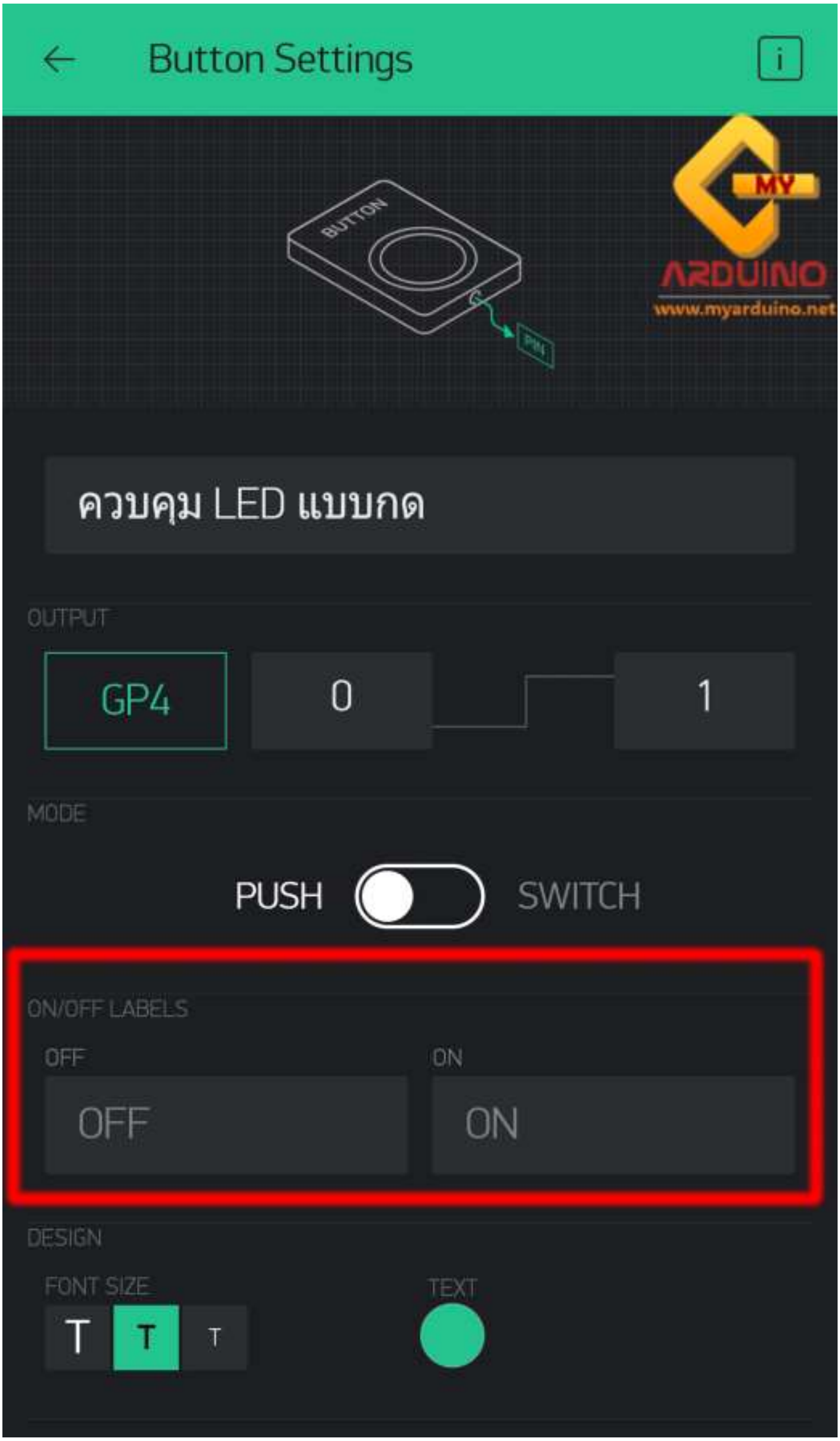
9. จากนั้นทำการตั้งค่าสัญญาณว่าจะให้การสัญญาณเป็นลักษณะใด (จะตั้งหรือไม่ก็ได้)



10.จากนั้นให้เลือกโหมดการทำงาน (MODE) ว่าจะให้เป็นแบบ **“PUSH” หรือ “SWITCH”** (ทางเราเลือก PUSH)

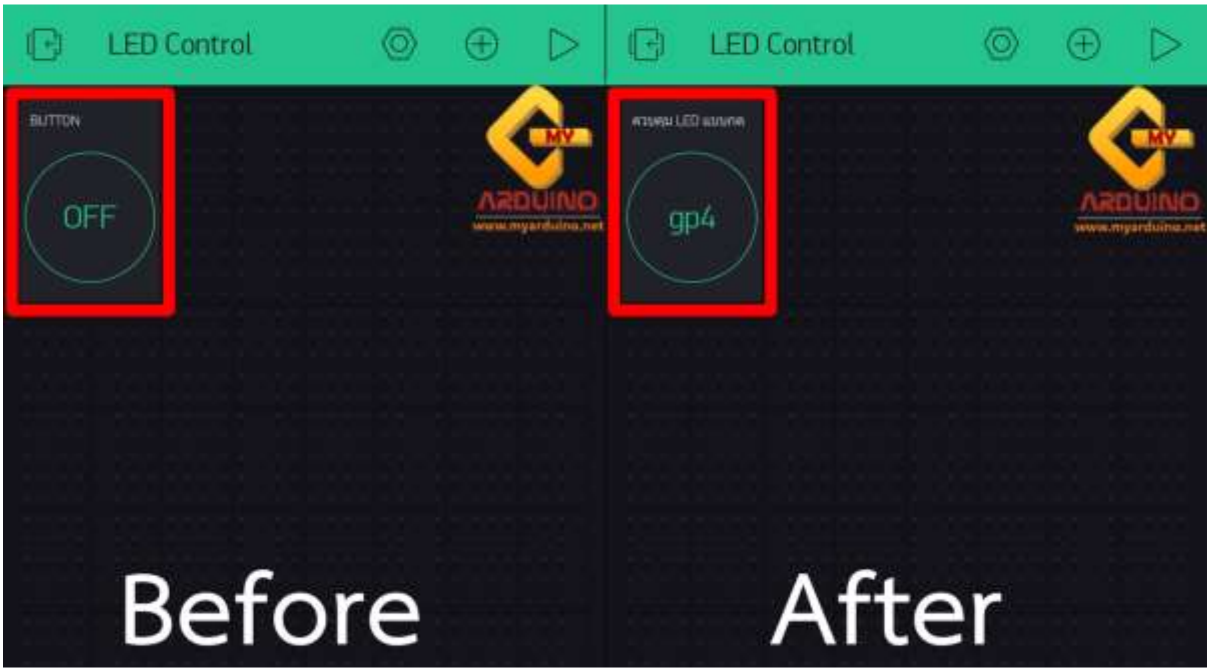


11.จากนั้นทำการตั้งชื่อ **LABELS** ว่าจะให้การทำงานในแต่ละช่วงชื่อว่าอะไร (จะตั้งหรือไม่ก็ได้)



12.เมื่อทำการตั้งค่าเสร็จแล้วให้กดไปที่ลูกศรมุมซ้ายบน

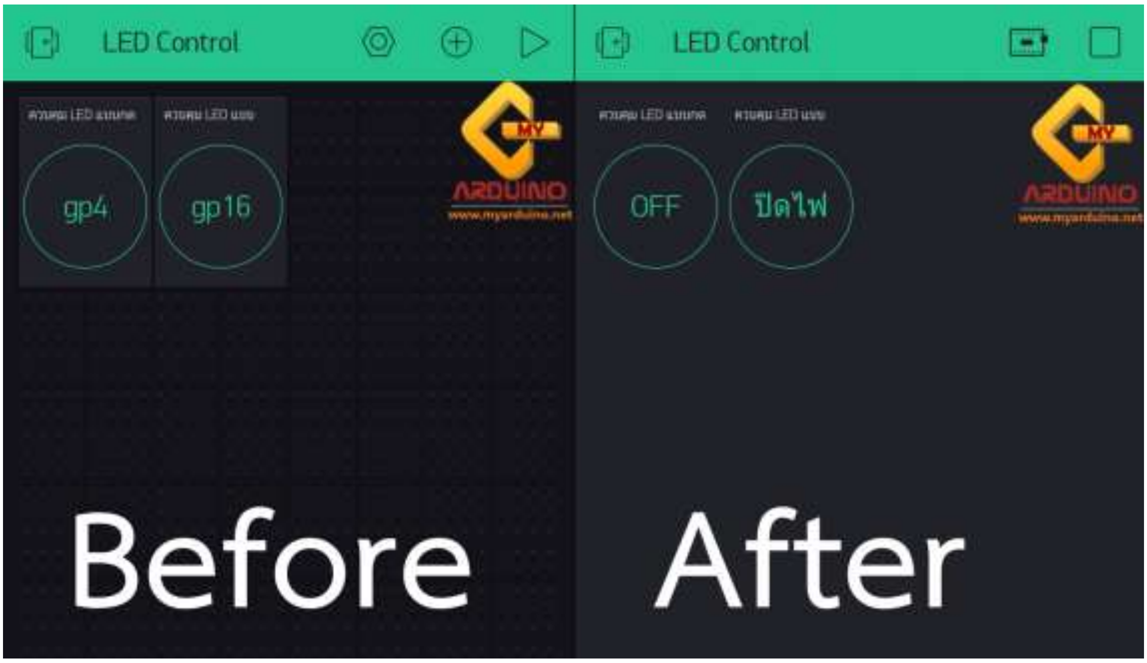
13.เมื่อออกมาแล้วจะเห็นได้ว่าหน้าตาของ Widget ที่เลือกมานั้นแตกต่างจากเดิมเล็กน้อย



14.ให้ทำการสร้าง **WidgetButton** ที่ 2 ขึ้นมาอีกครั้ง แต่ในครั้งนี้ในช่อง **MODE** ให้เปลี่ยนจาก **PUSH** เป็น **SWITCH** (ในที่นี้ทางเราเชื่อมต่อกับขา **D0 หมายเลข GPIO คือ GP16**)

15.เมื่อได้ **Widget** ทั้ง 2 มาแล้วให้กดไปที่ปุ่มสามเหลี่ยมที่อยู่มุมขวาบนเพื่อเริ่มการใช้งานอุปกรณ์ที่เชื่อมต่อไว้

16.จะเห็นได้ว่าเมื่อกดเริ่มต้นการใช้งานแล้วหน้าจอจะเปลี่ยนไปเล็กน้อย



17. เริ่มต้นการใช้งานให้ทดลองกดไปที่ **Widget** แรกที่เซ็ต **MODE PUSH** ไว้ จะเห็นได้ว่าในโหมดนี้จะเป็นการกดที่ **Widget** เพื่อส่งสัญญาณไปยังบอร์ดเพื่อให้ **LED** ติด แต่เมื่อปล่อยมือออกจาก **Widget ไฟ LED** ก็จะดับทันที

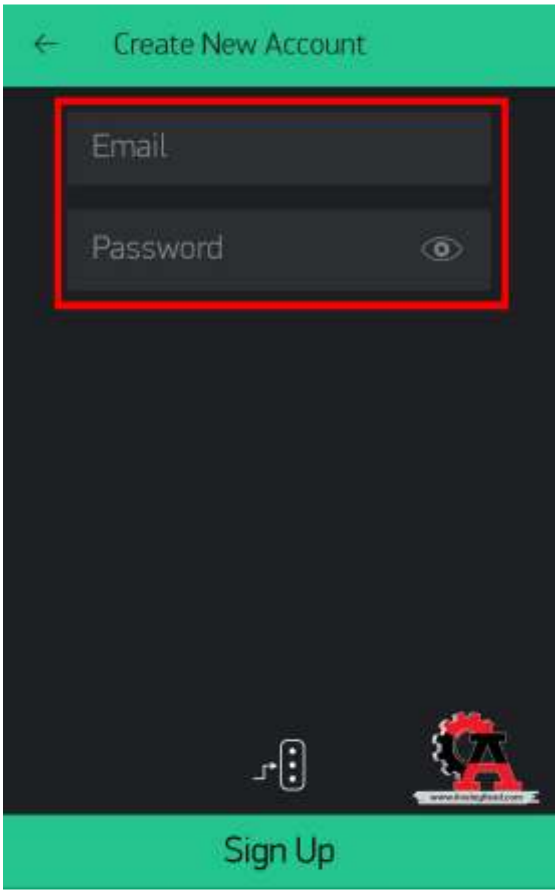
18. ต่อมาให้ทดลองกดไปที่ **Widget ที่ 2** ที่เซ็ต **MODE SWITCH** ไว้ จะเห็นได้ว่าในโหมดนี้จะเป็นการทำงานแบบสวิตซ์ไฟตามชื่อ คือ เมื่อทำการกดที่ Widget ที่ 2 แล้วไฟจะติดและเมื่อปล่อยมือจาก Widget ไฟก็จะยังไม่ดับ แต่เมื่อกดไปที่ **Widget** อีกครั้งไฟ LED ก็จะดับ

19. นอกเหนือจากนั้นทั้ง 2 Widget ก็ยังสามารถใช้งานพร้อมกันได้

การใช้งาน ESP8266 ตั้งเวลาเปิด/ปิดไฟ ด้วย
วิธีการใช้งานแอพพลิเคชั่นBlynk

1. โหลดแอพ **Blynk** จาก http://j.mp/blynk_Android หรือ http://j.mp/blynk_iOS
2. เปิดแอพพลิเคชั่นกด **CeateNewAccount**

3. ใส่ **email** และ **password**

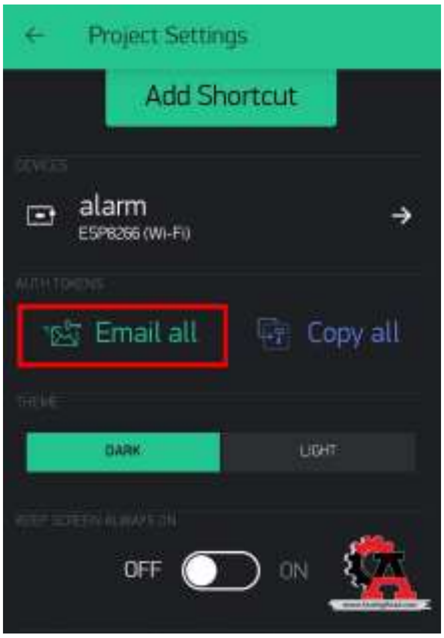


4. กดปุ่มสแกน **QRcode**

5. สแกน **QRcode**

6.กดปุ่มตั้งค่า **รูปหกเหลี่ยม**

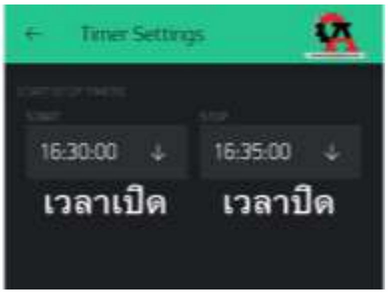
7.กดปุ่ม **Email all** เพื่อส่ง **Token** ไปยัง **Email**



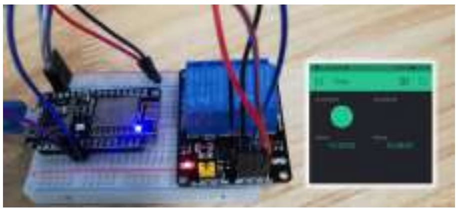
8.แก้ไข Tokens ชื่อไวไฟ และ**รหัสผ่านไวไฟ**

ตัวอย่างการใช้งาน

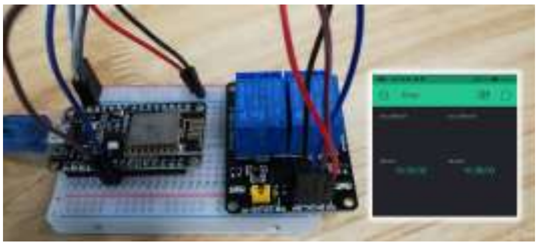
1.กดที่เวลา เพื่อตั้งค่าเวลา เปิด/ปิด



2.เมื่อเวลาตรงกับที่ตั้งค่า ไฟจะสว่าง



3.เมื่อถึงเวลาปิด ไฟจะดับ



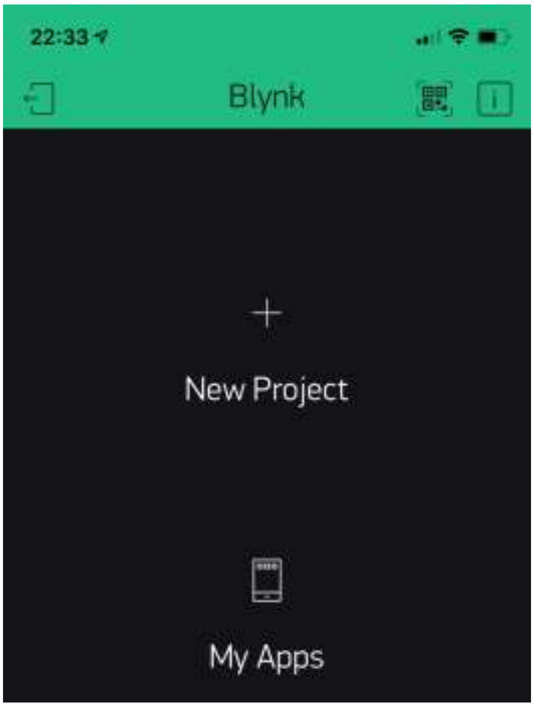
การใช้งานรีเลย์การไฟกระแสสลับ220Vเพื่อควบคุมเปิด/ปิด หลอดไฟ

****ขณะใช้อยู่กับไฟบ้าน220Vไม่ควรใช้มือจับวงจร ให้ถอดปลั๊กออกทุกครั้งหากต้องการแก้ไข

โปรเจคเปิด-ปิด LED ผ่าน WiFi ด้วย Blynk

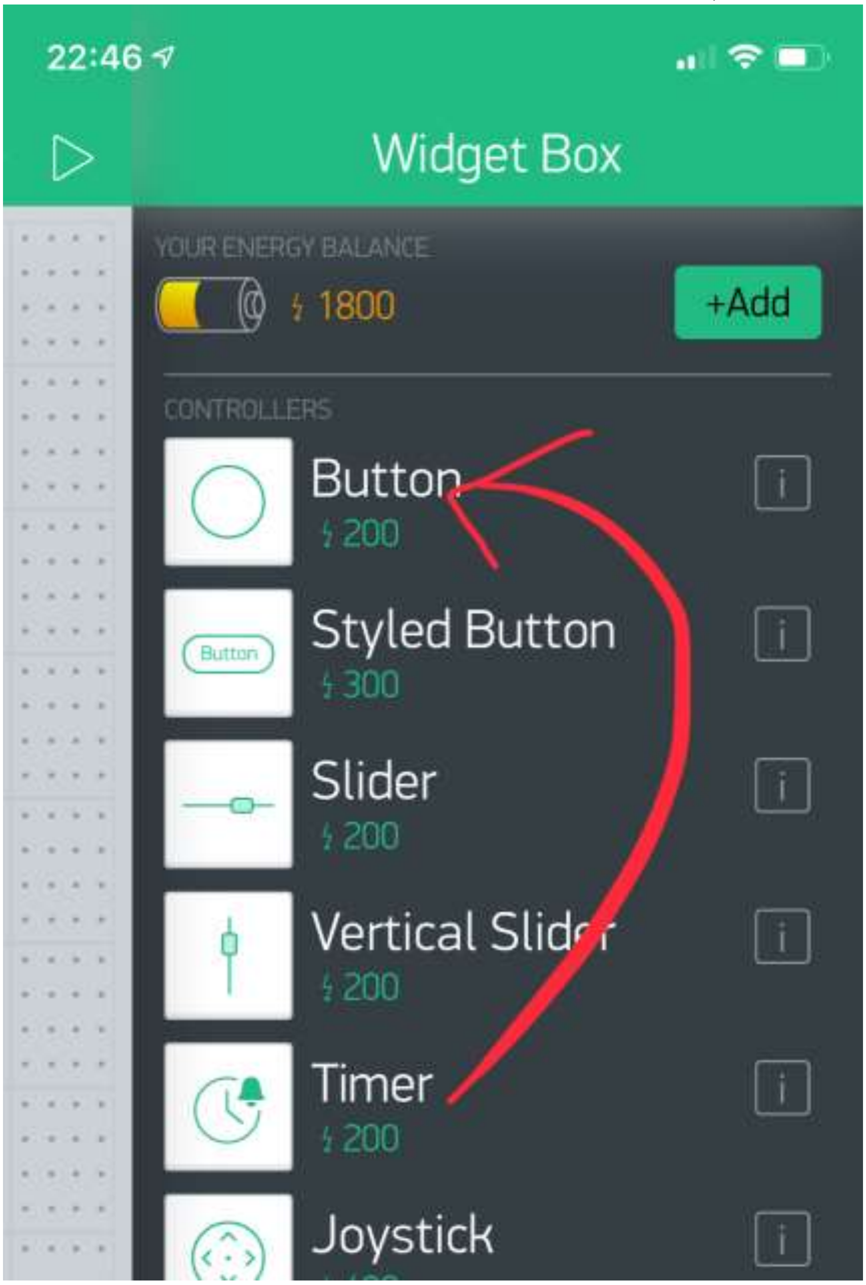
เริ่มสร้างโปรเจคใน Blynk

1. กด New Project เพื่อทำการสร้างโปรเจคใหม่

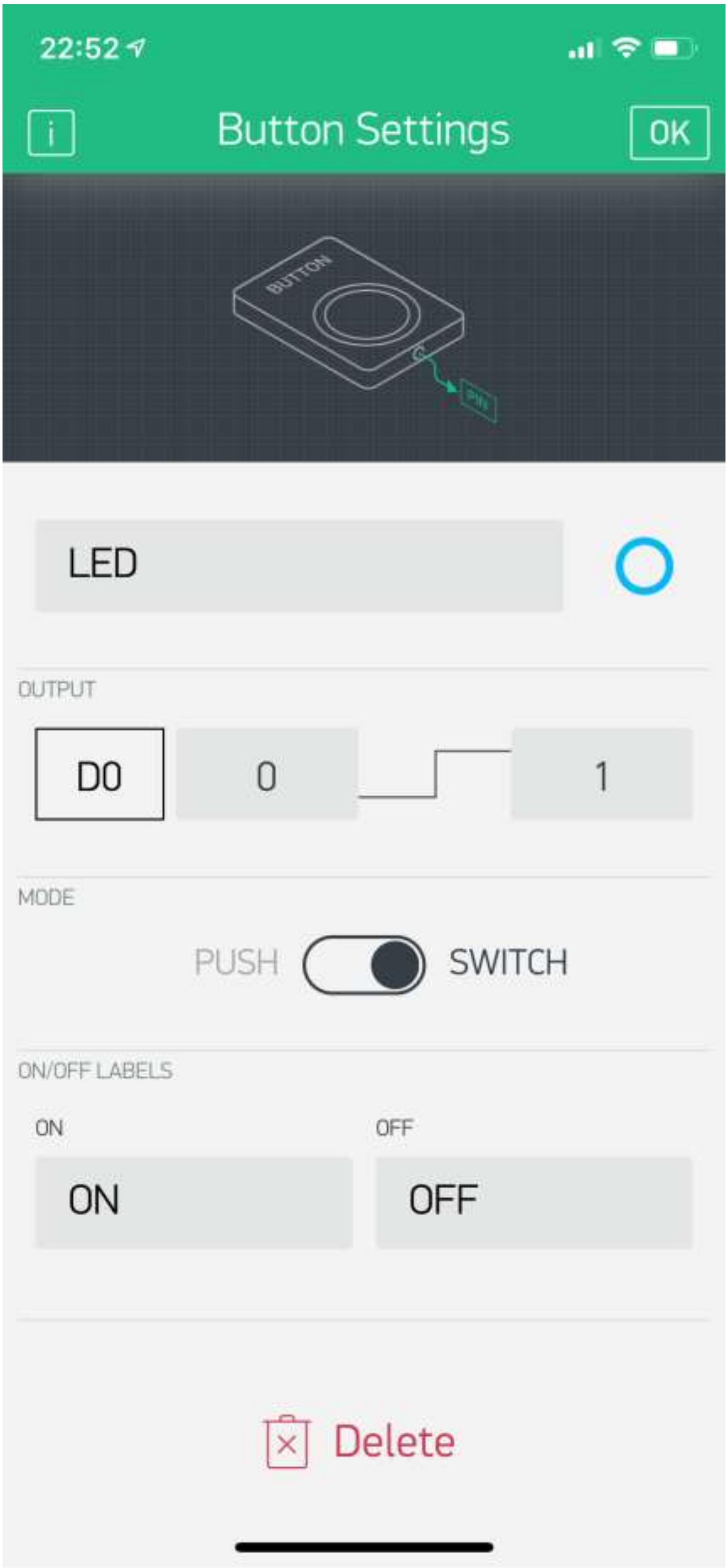


2. จากนั้นตั้งชื่อโปรเจค ทำการเลือกชนิดของ**บอร์ดไมโครคอนโทรเลอร์** และ การเชื่อมต่อแบบ **WiFi**

3. เมื่อกด **Create Project** ระบบจะทำการส่ง Token ไปยัง e-mail ที่เราได้ทำการลงทะเบียนเอาไว้ เก็บ Token นี้ไว้ใช้ Authenticate ตอนเขียนโค๊ด ต่อไปทำการสร้างปุ่มง่ายๆขึ้นมาอันนึง



4. กำหนด ชื่อ **Label** อะไรก็ได้ ในที่นี้ผมตั้ง **LED** และ อย่าลืมกำหนดขา **Digital Output** ในที่นี้เลือก **D0** หรือ **digital** ขาไหนก็ได้ ผมเริ่มจาก **D0** เพราะเดียวโปรเจคหน้าจะใช้ขาอื่นต่อ


22:52
Button Settings
OK
BUTTON
PIN
LED
OUTPUT
D0
0
1
MODE
PUSH
SWITCH
ON/OFF LABELS
ON
OFF
ON
OFF
Delete

เขียนโค๊ด

ให้ทำการติดตั้ง Library Blynk สำหรับ Arduino IDE ก่อน ซึ่งผมจะใช้วิธีการ git clone ซึ่งไวดี แต่ถ้าสะดวกจะดาวน์โหลด ให้ใช้วิธีนี้

https://github.com/blynkkk/blynk-library/archive/master.zip

เปิด Arduino IDE > Sketch > Include Libraries > Add .ZIP library...

```
/**
 * @project Controle LED via WiFi with Blynk
 * @author  Teerapong Singthong <st.teerapong@gmail.com>
 * @desc    medium.com/iamgoangle
 */
 #define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
// Blynk Auth Token
char auth[] = "xxx";
// WiFi
char ssid[] = "xxx";
char pass[] = "xxx";
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
}
void loop()
{
  Blynk.run();
}
```

แก้ไขตัวแปร ดังนี้

char auth[] ด้วย Blynk Token ที่ได้มา

char ssid[] ชื่อ WiFi SSID

char pass[] รหัสผ่าน WiFi

จากนั้น Burn ลง NodeMCU ด้วยโค๊ดชุดข้างบน แล้วดูผลลัพธ์ที่ Serial monitor

Nodemcu esp8266 Arduino IDE เปิดปิดไฟผ่าน Website

อุปกรณ์ที่ต้องเตรียม

Nodemcu V2 esp8266

สาย Jumper Female to Female

ตัวต้านทาน R220 ohm

หลอดไฟ LED (ในตัวอย่างนี้พี่ใช้หลอดไฟ LED น้องๆ สามารถใช้ Relay มาต่อเพื่อควบคุมไฟ DC AC ได้เหมือนกันครับ)

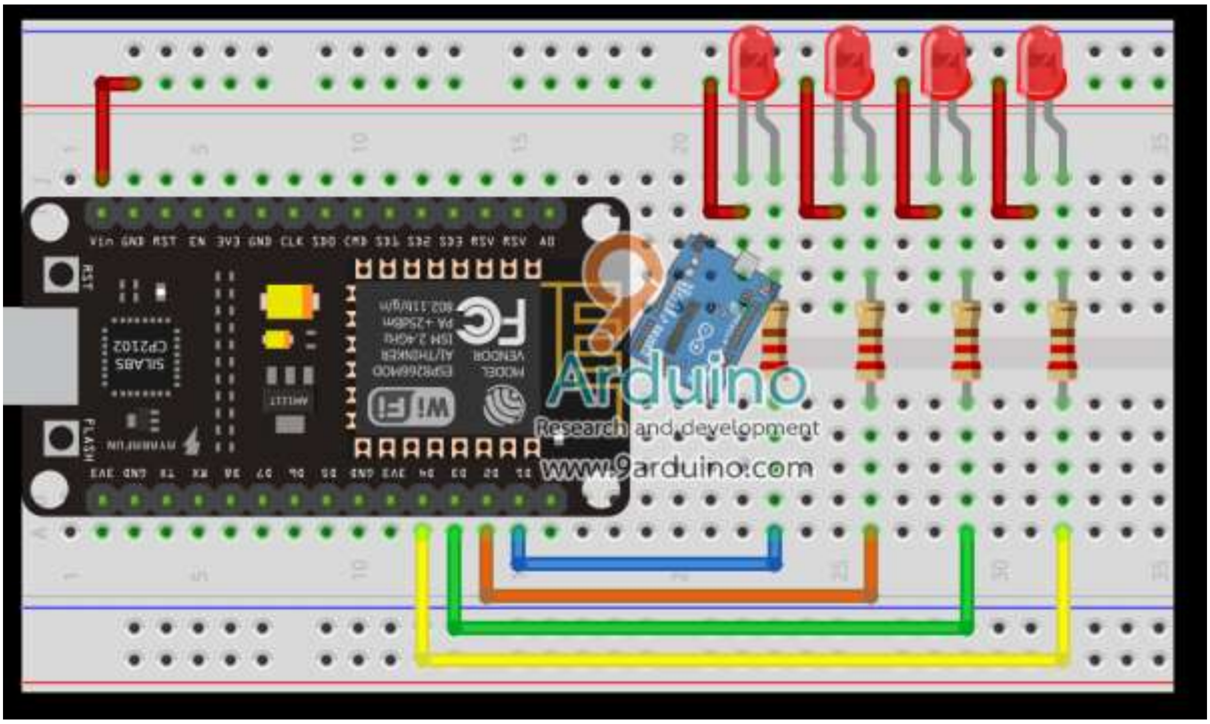
คอมพิวเตอร์ที่ลงโปรแกรม Arduino IDE และ ติดตั้งบอร์ด Nodemcu หากน้องๆยังไม่ติดตั้ง คลิก

ระบบเครือข่าย Lan มี Wifi ด้วยน่ะครับ

มาเริ่มกันเลย การต่อพี่ต่อดังนี้

จากก่อนหน้านี้มีน้องเข้ามาขอ Code ควบคุมการเปิดปิดไฟผ่าน Web site วันนี้เลยหยิบ Nodemcu Esp8266 มาทำชุดเปิดปิดไฟแบบง่ายๆ ด้วยการเขียน Code บน Arduino IDE ครับ

ก่อนอื่นต้องทำความเข้าใจก่อน น่ะครับ ในระบบบทความนี้เราจะทำให้ Nodemcu ของเราเป็น Web Server และ มี Device อื่นๆ เช่น Notebook โทรศัพท์มือถือ เป็น Client เข้าไปเชื่อมต่อ ไปยัง Nodemcu ในกรณีนี้ จะสามารถใช้ได้เพียงแค่ใน วง Lan แต่ถ้าต้องการออกนอกวง Lan หรือ Wan จะต้องเข้าไป Forword Port ที่ Router ชี้ IP และ Port มายัง Nodemcu มาเริ่มกันเลยครับ



Nodemcu Esp8266 On Web.ino

```
/*
  Nodemcu esp8266 Arduino IDE เปิดปิดไฟผ่าน Website
  https://www.ab.in.th/b/27
*/
```

```
#include <ESP8266WiFi.h>
const char* ssid = "9arduino";
const char* password = "tv357911itv";
int ledPin1 = D1; // ขา D1
int ledPin2 = D2; // ขา D2
int ledPin3 = D3; // ขา D3
int ledPin4 = D4; // ขา D4
int ch1 = 0;
int ch2 = 0;
int ch3 = 0;
int ch4 = 0;
WiFiServer server(80);
void setup() {
  Serial.begin(9600);
  delay(100);
  // ประกาศขา เป็น Output
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  // เริ่มต้น ให้ Logic ตำแหน่งขาเป็น HIGH
  digitalWrite(ledPin1, HIGH);
  digitalWrite(ledPin2, HIGH);
  digitalWrite(ledPin3, HIGH);
  digitalWrite(ledPin4, HIGH);
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

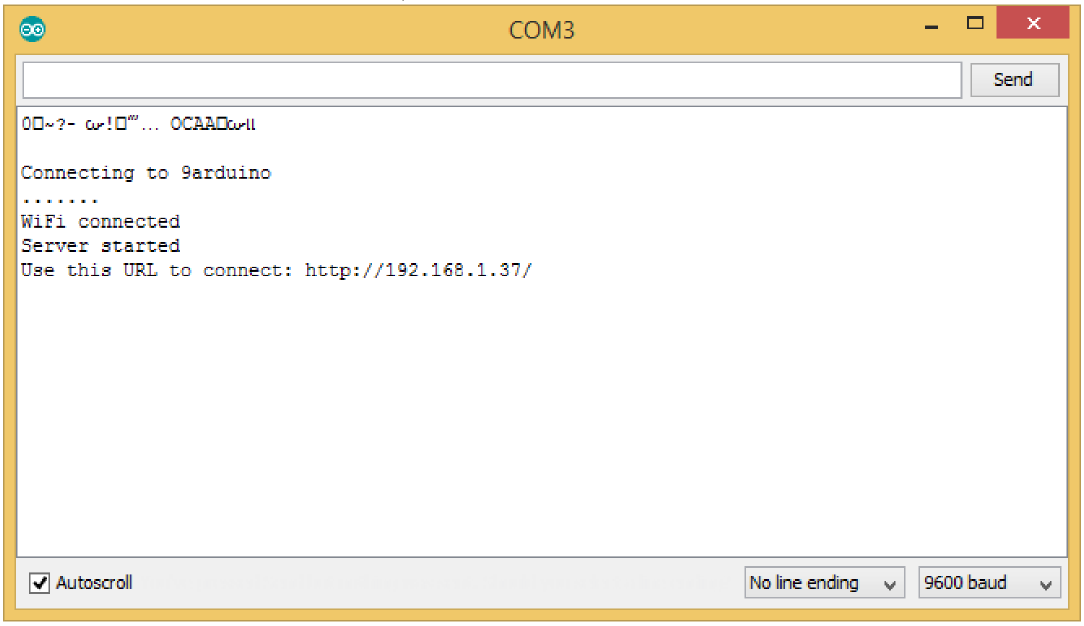
```
  Serial.println("");
  Serial.println("WiFi connected");
  // Start the server
  server.begin();
  Serial.println("Server started");
  // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");
}
void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  // Wait until the client sends some data
  Serial.println("new client");
  while (!client.available()) {
    delay(1);
  }
  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();
  // Match the request
  // รับ Link ช่อง 1
if (request.indexOf("/LED1=OFF") != -1) {
    digitalWrite(ledPin1, LOW);
    ch1 = 1;
  }
  if (request.indexOf("/LED1=ON") != -1) {
    digitalWrite(ledPin1, HIGH);
    ch1 = 0;
```

```
}
// รับ Link ช่อง 2
if (request.indexOf("/LED2=OFF") != -1) {
  digitalWrite(ledPin2, LOW);
  ch2 = 1;
}
if (request.indexOf("/LED2=ON") != -1) {
  digitalWrite(ledPin2, HIGH);
  ch2 = 0;
}
// รับ Link ช่อง 3
if (request.indexOf("/LED3=OFF") != -1) {
  digitalWrite(ledPin3, LOW);
  ch3 = 1;
}
if (request.indexOf("/LED3=ON") != -1) {
  digitalWrite(ledPin3, HIGH);
  ch3 = 0;
}
// รับ Link ช่อง 4
if (request.indexOf("/LED4=OFF") != -1) {
  digitalWrite(ledPin4, LOW);
  ch4 = 1;
}
if (request.indexOf("/LED4=ON") != -1) {
  digitalWrite(ledPin4, HIGH);
  ch4 = 0;
}
// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("");
client.println("");
client.print("Led pin D1 : ");
```

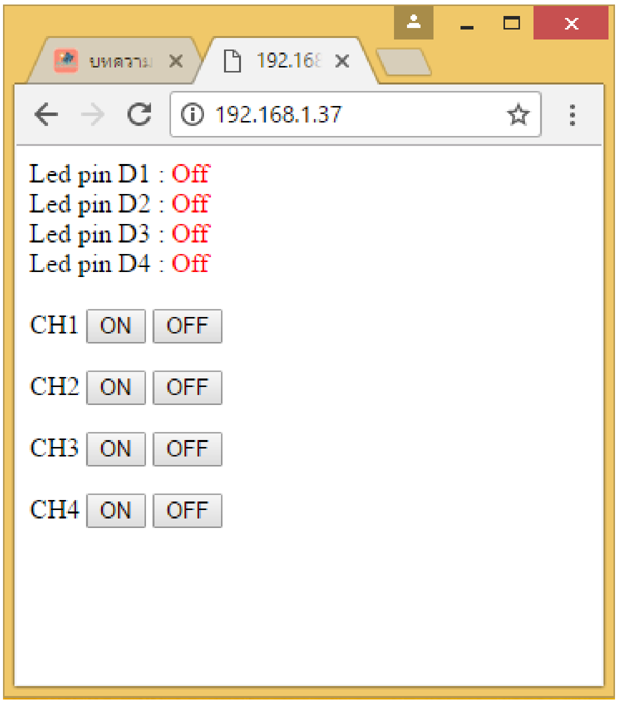
```
  if (ch1 == 1) {
    client.print("<font color='red'>Off </font><br>");
  } else {
    client.print("<font color='green'>On </font><br>");
  }
  client.print("Led pin D2 : ");
  if (ch2 == 1) {
    client.print("<font color='red'>Off </font><br>");
  } else {
    client.print("<font color='green'>On </font><br>");
  }
  client.print("Led pin D3 : ");
  if (ch3 == 1) {
    client.print("<font color='red'>Off </font><br>");
  } else {
    client.print("<font color='green'>On </font><br>");
  }
  client.print("Led pin D4 : ");
  if (ch4 == 1) {
    client.print("<font color='red'>Off </font><br>");
  } else {
    client.print("<font color='green'>On </font><br>");
  }
  client.println("<br>");
  client.println("CH1 <a href=\"/LED1=ON\"\"><button> On </button></a><a
href=\"/LED1=OFF\"\"><button> OFF </button></a><br>");
  client.println("CH2 <a href=\"/LED2=ON\"\"><button> On </button></a><a
href=\"/LED2=OFF\"\"><button> OFF </button></a><br>");
  client.println("CH3 <a href=\"/LED3=ON\"\"><button> On </button></a><a
href=\"/LED3=OFF\"\"><button> OFF </button></a><br>");
  client.println("CH4 <a href=\"/LED4=ON\"\"><button> On </button></a><a
href=\"/LED4=OFF\"\"><button> OFF </button></a><br>");
  client.println("<br>");

  delay(1);
```

```
  Serial.println("Client disonnected");
  Serial.println("");
}
```

เมื่อ อัพโหลด Code เสร็จ ก็ให้กด ปุ่ม Serial moniter บน Arduino IDE จะแสดงข้อความตามภาพ



จากนั้น โปรแกรมจะแสดง IP ให้เราเข้าไปยัง URL ที่โปรแกรมแจ้งแสดงออกมาครับ จากภาพคือเว็บ http://192.168.1.37/



ก็จะสามารถเปิดปิดไฟได้ตามที่ต้องการเลยครับ

ทำ โครงงาน โครงการ โปรเจคอะไรดี โปรเจคจบ (Arduino Web Program App)

นักเรียนนักศึกษาทั้งมหาลัย วิทยาลัย ในชั้นปีสุดท้ายมักจะมาขอคำปรึกษาสอบถาม เกี่ยวกับหัวข้อโครงงาน โปรเจคจบ Arduino กันบ่อยๆ สำหรับน้องๆ ที่กำลังหาไอเดีย ทำหัวข้อวิจัย โปรเจคจบ วันนี้ทางร้านจะสรุปให้ฟังถึงวิธีการหาไอเดียทำโปรเจคใหม่ๆ (เป็นตามความเข้าใจของทางร้าน น่ะครับผิดถูก ขออภัยด้วยครับ) ข้อมูลมาจากเว็บไซต์ https://www.ab.in.th/article/41

เริ่มต้นกันเลย สำหรับวิธีเก่าๆ ที่อาจารย์ในมหาลัยจะแนะนำน้องๆคือ

หาระบบเก่าๆ ที่ยังไม่เป็นระบบสารสนเทศ นำมาดัดแปลง ให้เป็นระบบสารสนเทศ ยกตัวอย่างเช่น ระบบคลังสินค้า ในอดีตร้านค้ามักจะจดลงในสมุด ทำให้เกิดข้อผิดพลาดได้ง่าย เกิดความซ้ำซ้อนของข้อมูล อาจเกิดการสูญหายของข้อมูลสมุดหาย น้องๆก็สามารถออกแบบระบบ โดยใช้ Web หรือจะเขียนเป็นโปรแกรม Ui จำพวก VB.net C# แล้วเก็บลง Database ก็จะสามารถ ช่วยลดความผิดพลาดของ ระบบได้อีกด้วย

นี้เป็นวิธีที่อาจารย์หลายท่านสอนในห้องเรียน

แต่สำหรับ Arduino AB-Maker คิดแบบในห้องเรียนมันก็ไม่ใช่จริงป่าว เราเลยมีวิธีการคิดหัวข้อโปรเจค ดังนี้เลย

- ไปเดินตลาด ที่ชุมชน สังเกตปัญหาที่เกิดขึ้น จดปัญหาที่เราสังเกตเจอแล้วจดลงกระดาษเป็นข้อๆ
- สรุปปัญหาที่เกิดขึ้นแล้วเอาวิชาที่เรามีอยู่ มาแก้ปัญหาได้อย่างไร
- เลือกปัญหาที่เกิดขึ้นที่คิดว่าใช่สำหรับเรา ศึกษาความเป็นไปได้ของโปรเจค
- เลือก Hardware หรือ Software ให้เหมาะสม
- หาอาจารย์ที่ปรึกษา ที่คิดว่าสามารถช่วยเราได้ อาจารย์ดีมีชัยไปกว่าครึ่ง

ถัดมาการเสนอหัวข้อโปรเจค หรือสอบหัวข้อโปรเจค

จุดที่ที่สำคัญที่สุดคือ ขอบเขต ขอบเขตตอนเสนอพยายาม เสนอระบบให้น้อยไว้ก่อนเพราะอาจารย์มักจะเพิ่มเติมให้เราเอง หากเราเสนอไปเยอะๆ งานอาจจะบานปลายได้ และพยายามต่อรองกับอาจารย์ให้ได้มากที่สุด งานเราจะเยอะจะน้อยขึ้นกับการต่อรองของเราเก่งแค่ไหน

การเลือกอาจารย์ ก็สำคัญเช่นกัน เพราะ พฤติกรรมของอาจารย์ที่ปรึกษา ส่วนใหญ่เวลาสอบโปรเจค อาจารย์ที่ปรึกษามักจะช่วยเรา

ตัวอย่างที่ 1 (จากภาพคนกำลังหาเลข ล็อตตารี่ ที่ต้องการ)

ภาพจาก https://www.thaich8.com/news_detail/2952

จากภาพปัญหาที่เกิดขึ้น ผู้ซื้อ เลือกเลขล๊อตตารี่ที่ต้องการไม่ได้ แถมต้องไปหาเลขด้วยตัวเอง แก้ปัญหาด้วยการ พัฒนาระบบ Web หรือ App เพื่อเป็นการจองเลขในล๊อตถัดไป หรือขายออนไลท์ กันไปเลยผู้ซื้อจะได้ค้นหาเลขที่ตัวเองต้องการง่ายๆผ่าน App หรือ Website

ตัวอย่างที่ 2 (ปัญหารถติด)

ภาพจาก ytimg.com/

จากภาพปัญหาคือ รถติด

แก้ปัญหาด้วยการ ออกแบบระบบตรวจสอบปริมาณรถบนท้องถนน ติดไว้ตามเสาร์ไฟฟ้า อาจจะใช้ Raspberry pi Arduino ตรวจจับ จากนั้นส่ง Data ขึ้น Cloud เพื่อแจ้งบอกสถานะผ่าน App หรือ Web ว่า ถนนเส้นนี้รถติดควรหลีกเลี่ยง เป็นต้น โปรเจคนี้จะเป็นงานลักษณะ Internet of thinks (IOT)

ตัวอย่างที่ 3 (ปัญหาน้ำป่า)

ภาพจาก bangkokbiznews.com

จากภาพปัญหาที่เกิดขึ้นคือ น้ำป่าจากภูเขา

แก้ปัญหาด้วยการ ออกแบบระบบวัดความแรงของน้ำ สีของน้ำ เพื่อแจ้งเตือน หากเกิดน้ำป่าไหลหลาก อาจจะใช้ Arduino, Nodemcu ในการวัดความแรงของน้ำ สีของน้ำ ส่งขึ้น Cloud หากมีการไหลเร็วและสีของน้ำเปลี่ยนไปให้แจ้งเตือนผ่านทาง Line เป็นต้น

ปัญหาที่ 4 (แย่งซื้อของ)

ภาพจาก https://i.ytimg.com

จากภาพปัญหาที่เกิดขึ้นคือ แย่งซื้อของ

แก้ปัญหาด้วยการ ทำระบบจองคิวด้วย Arduino ปริ้นด้วยเครื่องปริ้นความร้อน

ปัญหาที่ 5 (น้ำท่วม)

ภาพจาก http://www.matichon.co.th/

จากภาพเกิดปัญหา น้ำท่วมทาง ท่วมเมือง

แก้ปัญหาด้วยการ ออกแบบระบบเฝ้าระวัง น้ำท่วม โดยการใช้ Arduino, Nodemcu ตรวจวัดระดับของน้อง ตามคลอง จุดเสี่ยงต่างๆ หรือจะทำการถ่ายภาพ แล้วส่งไปเก็บใน Cloud ทุกๆ 15 นาที หากน้ำอยู่ในขั้น วิกฤต ให้แจ้งเตือนผ่าน Line เพื่อเตรียมพร้อม ยกของหนีน้ำ และยังเก็บสถิติของระดับน้ำในลำคลองเพื่อ ดูแลเรืองปัญหาน้ำแล้ง ได้อีกด้วย ตัวอย่าง เรดาร์ สทิงพระ สงขลา

http://www.hatyaicityclimate.org/flood/cam/6

นี้ก็เป็นแนวคิดให้น้องๆ พัฒนาโปรเจคในอนาคต เทรนต่อไปในอนาคต เขาจะไม่มอง เป็นสาย Hardware,Software,Web developer แยกกันไปอีกแล้ว ทุกอย่างมันกำลังจะรวมกัน สู้ๆ ครับน้อง

เรามาทำความรู้จักเกี่ยวกับ Json กันก่อน

Json หรือมีอีกชื่อหนึ่งเรียกว่า JavaScript Object Notation คือ รูปแบบของข้อมูลที่ใช้สำหรับแลกเปลี่ยน ข้อมูลที่มีขนาดเล็ก เป็นคำ หรือตัวแปรที่สามารถทำความเข้าใจได้ง่าย และถูกสร้างและอ่านได้ง่ายภายใต้ภาษา JavaScript (Java Script Programming Language)

โครงสร้างของ Json นั้นใช้ลักษณะภาษาของ JavaScript แต่ไม่ถูกมองว่าเป็นภาษาโปรแกรม กลับถูกมองว่าเป็นภาษาในการแลกเปลี่ยนข้อมูลมากกว่า ตัวอย่างของ Json โครงสร้างดังนี้

```
[
  {"firstname":"name11","lastname":"name12"}
  {"firstname1":"name21","lastname1":"name22"}
]
```

ปัจจุบัน มีการนำ json ไปใช้งานทางด้าน App การสื่อสารแบบต่างๆ การดึงค่าต่างๆ เช่น api พยากรณ์อากาศ ราคาทองคำ ราคาน้ำมัน มี api เยอะแยะมากมากให้เราสามารถนำมาใช้งาน ดึงมาใช้ หรือมาแสดงได้ครับ

สามารถศึกษาข้อมูลเพิ่มเติมได้ที่ http://www.json.org/

เมื่อเข้าใจ เกี่ยวกับ Json แล้วคราวนี้เรามาเริ่มต้นเขียน Code เพื่อดึงค่า Json มาใช้งานกัน

ก่อนอื่นเริ่มสร้าง Json ด้วย PHP จาก Code (testjson.php)

```
<?php
$arr = [
  [
    "ch1" => "1",
    "ch2" => "2",
    "ch3" => "3",
    "ch4" => "4"
  ]
];
echo json_encode($arr);
?>
```

สามารถดูตัวอย่างได้ที่เว็บ : http://9arduino.nisit.net/api/json.php

ถัดมาก็มาถึง ในส่วนของ Nodemcu esp8266 (nodemcu read Json.ino)

```
#include
const char* ssid     = "Use Wifi";
const char* password = "Pass Wifi";
const char* host = "9arduino.nisit.net";  //Doamin ที่ต้องการดึงค่ามาใช้
int ch1,ch2,ch3,ch4 = 0;
void setup() {
```

```
  Serial.begin(9600);
  pinMode(D4, OUTPUT);
  delay(10);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
int value = 0;
void loop() {
  delay(2000);
  ++value;
  Serial.print("connecting to ");
  Serial.println(host);
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  String url = "/api/json.php";    //ดึงค่าจากไฟล์ http://9arduino.nisit.net/api/json.php
  Serial.print("Requesting URL: ");
  Serial.println(url);
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
```

```
            "Connection: close\r\n\r\n");
  unsigned long timeout = millis();
  while (client.available() == 0) {
    if (millis() - timeout > 5000) {
      Serial.println(">>> Client Timeout !");
      client.stop();
      return;
    }
  }
// ในส่วนของการดึง Json โดยการดึง ตัวแปรที่ชื่อว่าตัวแปรมาใช้งาน
// ยกตัวอย่าง ตัวแปร ch1 ค่าที่ได้จะเป็น 1 แสดงออกมา เราสามารถนำ ตัว แปร ch1 ไปใช้งานต่างๆได้เช่น
การแสดงข้อความออกจอ LCD เปิดปิดไฟตามกำหนด
  if(client.find("")){
      client.find("ch1");  //
      ch1 = client.parseFloat();
      client.find("ch2");
      ch2 = client.parseFloat();
      client.find("ch3");
      ch3 = client.parseFloat();
      client.find("ch4");
      ch4 = client.parseFloat();
  }
  Serial.print("Output = ");
  Serial.println(ch1);
  Serial.print("Output = ");
  Serial.println(ch2);
  Serial.print("Output = ");
  Serial.println(ch3);
  Serial.print("Output = ");
  Serial.println(ch4);
  Serial.print("Output = ");
  Serial.println();
  Serial.println("closing connection");
}
```

ดังในภาพ ในส่วนของไฟล์ json php เราสามารถ อ่านค่าจาก database นำมาแสดงยัง Nodemcu

Arduino เชื่อมต่อกับ Database (Database to Arduino)

ดีครับ จากวันก่อนมีน้องๆ ในกลุ่ม Arduino Thailand เข้ามาถามเรื่องการให้ Arduino อ่านค่าจาก Database (Mysql) วันนี้เลยทำบทความเล่นๆ อ่านเข้าใจง่ายถึงวิธีการง่าย (แต่ไม่ปลอดภัยหากใช้งานจริง) เผื่อเอาไปใช้กับโปรเจคนักเรียนนักศึกษา มาเริ่มกันเลย

ก่อนอื่นจะอธิบายหลักการทำงานก่อน

ขั้นที่ 1 น้องๆ ต้องออกแบบ เว็บไซต์ไม่ว่าจะเป็นภาษา Html php asp.net แล้วแต่น้องชอบ ดึงขอมูล จาก Database มาแสดงบนเว็บไซต์ ก่อน อาจจะระบุ ว่า Output= เพื่อให้ Arduino ค้นหาข้อมูลจากเว็บไซต์

ขั้นที่ 2 น้องๆ ก็ให้ Arduino อ่านหน้าเว็บไซต์ และ ค้นหาตัวแปรที่เราต้องการเช่น จากตัวอย่าง พี่ให้ Arduino อ่านหน้าเว็บไซต์ ค้นหาคำว่า "Output=" แล้วนำค่าด้านหลังที่ได้มาเก็บในตัวแปรชนิด Int ของ Arduino

ก็จบขั้นตอนง่ายๆ แบบไม่ปลอดภัย ในกรณีที่จำนำไปใช้จริงควรจะ มีการเข้ารหัส

เครื่องมือที่ต้องใช้

1. Arduino Uno R3
2. Ethernet shield w5100
3. Server หรืออาจจะจำลอง Server โดยใช้ Appserve หรือ Xamp
4. Web Server ไม่ว่าจะเป็น IIS หรือ apache
5. PHP จากตัวอย่างเลือกใช้ 5.6
6. Mysql
7. อุปกรณ์ เครือข่าย Network

ไฟล์ ทั้งหมด

Name	Date modified	Type	Size
config	5/7/2559 15:32	PHP Script	1 KB
index	5/7/2559 15:32	PHP Script	1 KB
sql	5/7/2559 15:10	Text Document	1 KB

1. สร้าง Database ก่อน พี่เลือกใช้ Mysql

```
CREATE TABLE `test` (
  `id` int(11) NOT NULL auto_increment,
  `data` int(11) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
  INSERT INTO `test` VALUES (1, 152);
```

2. ตัวอย่าง Code ฝั่ง php ไฟล์ Index.php

```
9Arduino.com
require_once("config.php");
$sql = "SELECT * FROM  test WHERE  id = 1;";
$result = mysql_query($sql);
    $num_record = mysql_num_rows($result);
    while ($dbarr = mysql_fetch_row($result)) {
    echo "output=";
    echo "$dbarr[1]";
    }
?>
```

3. ไฟล์ Config.php เพื่อเชื่อมต่อกับ Server

```
$host = "localhost";
$user = "root";
$pass = "123456";
$db = "testtdbs";
mysql_connect($host, $user, $pass) or die("Could not connect to database");
mysql_select_db($db) or die("Could not connect to database");
mysql_query("SET NAMES utf8")
?>
```

4. ตัวอย่าง Code ฝั่ง Arduino + Ethernet shield w5100

```
    #include <SPI.h>
    #include <Ethernet.h>
    byte mac[] = {
      0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
    };
    IPAddress ip(192, 168, 1, 177);
    IPAddress myDns(192, 168, 1, 1);
    EthernetClient client;
     //char server[] = "arduino.nisit.net";    //กรณีใช้เป็น Domain  จะต้อง ตั้งค่า DNS ที่
    myDns ด้วย
    IPAddress server(103,233,194,42);     //กรณีใช้เป็น IP
```

```
unsigned long lastConnectionTime = 0;             // last time you connected to the server, in milliseconds
const unsigned long postingInterval = 10L * 1000L; // delay between updates, in milliseconds
int value;
void setup() {
  Serial.begin(9600);
  while (!Serial) {
    ;
  }
  delay(1000);
  Ethernet.begin(mac, ip, myDns);
  // print the Ethernet board/shield's IP address:
  Serial.print("My IP address: ");
  Serial.println(Ethernet.localIP());
}
void loop() {
  if (client.available()) {
    //char c = client.find("output=");
    //Serial.write(c);
  }
  if (millis() - lastConnectionTime > postingInterval) {
    httpRequest();
    if(client.find("")){
      client.find("output=");  // ค้นหาคำว่า output= ในเว็บ แล้วดึงค่าตัวแปรออกมาแสดง
      value = client.parseFloat();
      Serial.print("Output = ");
      Serial.println(value);
    }
  }
}
void httpRequest() {
  client.stop();
  if (client.connect(server, 80)) {
    Serial.println("connecting...");
```

```
    // send the HTTP GET request:
    client.println("GET /~file/readmysql/index.php HTTP/1.1");  // Url ที่ต้องการวิ่งไปอ่านไฟล์
    client.println("Host: www.arduino.cc");
    client.println("User-Agent: arduino-ethernet");
    client.println("Connection: close");
    client.println();
    lastConnectionTime = millis();
  } else {q
    Serial.println("connection failed");
  }
}
```

จบขั้นตอนการดึงค่าจาก Database ครับ หวังว่าจะสามารถนำไปพัฒนาได้ในอนาคตน่ะครับ

การเก็บข้อมูลจาก Nodemcu Esp8266 ลงใน ฐานข้อมูล Database Mysql (Arduino to database)

จากบทความครั้งที่แล้ว ที่ได้เขียนบทความเกี่ยวกับการส่งค่าจาก Arduino Uno R3 + W5100 ส่งข้อมูลไปเก็บลง Database น้องๆได้เรียกร้องให้เขียนบทความโดยใช้ Nodemcu Esp8266 บ้าง ก็เลยเขียนบทความนี้ขึ้นมาให้น้องๆได้นำไปพัฒนาต่อยอดในอนาคตครับ

จริงๆแล้วเราจะใช้หลักการเดียวกันกับ บทความเก่า โดยใช้หลักการของ Methol get ในการส่งข้อมูลไปยัง Database โดยจะฝาก ตัวแปรที่เราต้องการจะส่ง ไปกับ URL ยกตัวอย่างเช่น

http://www.domain.com/index.php?temp=25

หรือในรูปแบบ ip

http://192.168.1.1/index.php?temp=25

ค่า 25 คือ ตัวแปรที่เราต้องการจะฝากไปยัง URL เพื่อส่งต่อไปเก็บลงใน Database

ตัวอย่างไฟล์ ขออณุญาติหยิบ ไฟล์ชุดเดิมมาใช้น่ะครับ

รูปไฟล์

1. Sql Blackup Database Mysql

```
CREATE TABLE `temp` (
  `id` int(11) NOT NULL auto_increment,
  `temp` varchar(30) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
INSERT INTO `temp` VALUES (1, '99.2');
```

2. Config.php (ใช้ระบุข้อมูลการเชื่อมต่อเข้า Database)

```
$host = "localhost";
$user = "root";
$pass = "123456";
$db = "send";
mysql_connect($host, $user, $pass) or die("Could not connect to database");
mysql_select_db($db) or die("Could not connect to database");
mysql_query("SET NAMES utf8")
?>
```

3. temp.php (ใช้รับค่าจาก Arduino และบันทึกลงใน Mysql)

```
    require_once("config.php");
    $temp = $_GET['Temp'];
    $sql = "UPDATE temp SET temp = '$temp' WHERE id =1";
    $sql_query = mysql_query($sql);
    if ($sql_query) {
        echo "Complete";
    } else {
        echo "Error";
    }
    ?>
```

4. Code Arduino (เลือกใช้ Arduino Uno R3 + W5100)

```
/*
บทความ พัฒนาระบบ Nodemcu ส่งค่าไปเก็บยัง database
*/
#include <ESP8266WiFi.h>
const char* ssid      = "SSID Wifi";
const char* password = "Pass Wifi";
const char* host = "data.sparkfun.com";  //ใส่ IP หรือ Host ของเครื่อง Database ก็ได้
const char* variable   = "25";  //ตัวแปรที่ต้องการจะส่ง
void setup() {
  Serial.begin(9600);
  delay(10);
  Serial.println();
  Serial.println();
```

```
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
int value = 0;
void loop() {
  delay(5000);
  ++value;
  Serial.print("connecting to ");
  Serial.println(host);
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  String url = "/temp/temp.php?temp=";      //ชุด Directory ที่เก็บไฟล์ และตัวแปรที่ต้องการจะฝาก
  url += variable;                          //ส่งค่าตัวแปร
  Serial.print("Requesting URL: ");
  Serial.println(url);
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  unsigned long timeout = millis();
  while (client.available() == 0) {
    if (millis() - timeout > 5000) {
      Serial.println(">>> Client Timeout !");
```

```
      client.stop();
      return;
    }
  }
  // Read all the lines of the reply from server and print them to Serial
  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
  Serial.println();
  Serial.println("closing connection");
}
```

การเก็บข้อมูลจาก Arduino ลงใน ฐานข้อมูล Database Mysql (Arduino to database)

หลักการทำงานเราจะใช้ Methol Get ของ php ในการฝาก ข้อมูลไปกับ Url ยกตัวอย่างเช่น
http://www.domain.com/index.php?temp=25
หรือในรูปแบบ ip
http://192.168.1.1/index.php?temp=25
จากนั้นให้ PHP ในการรับข้อมูลและบันทึกลงใน Database

ตัวอย่างไฟล์

Name	Date modified	Type	Size
arduino_to_database	15/2/2559 23:27	File folder	
config	15/2/2559 13:17	PHP Script	1 KB
sql.sql	15/2/2559 23:22	SQL File	1 KB
temp	15/2/2559 13:16	PHP Script	1 KB

1. Sql Blackup Database Mysql

```
CREATE TABLE `temp` (
  `id` int(11) NOT NULL auto_increment,
  `temp` varchar(30) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
INSERT INTO `temp` VALUES (1, '99.2');
```

2. Config.php (ใช้ระบุข้อมูลการเชื่อมต่อเข้า Database)

```
$host = "localhost";
$user = "root";
$pass = "123456";
$db = "send";
mysql_connect($host, $user, $pass) or die("Could not connect to database");
mysql_select_db($db) or die("Could not connect to database");
mysql_query("SET NAMES utf8")
?>
```

3. temp.php (ใช้รับค่าจาก Arduino และบันทึกลงใน Mysql)

```
require_once("config.php");
$temp = $_GET['Temp'];
$sql = "UPDATE temp SET temp = '$temp' WHERE id =1";
$sql_query = mysql_query($sql);
if ($sql_query) {
    echo "Complete";
} else {
    echo "Error";
}
?>
```

4. Code Arduino (เลือกใช้ Arduino Uno R3 + W5100)

```
#include
#include
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress server(192,168,0,101);  // ใช้ในกรณีเป็น IP
//char server[] = "www.google.com";    // ใช้ในกรณีเป็น Domain
IPAddress ip(192, 168, 0, 177); // IP ของ Arduino
EthernetClient client;

void setup() {
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
```

```
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    Ethernet.begin(mac, ip);
  }
  delay(1000);
  Serial.println("connecting...");
}
void loop()
{
  if (client.connect(server, 80)) {
    Serial.println("connected");
    client.println("GET /temp/temp.php?temp=25 HTTP/1.1");
    client.println("Host: www.google.com");
    client.println("Connection: close");
    client.println();
  }
  else {
    Serial.println("connection failed");
  }
}
```

วัดอุณหภูมิ ด้วย Nodemcu Esp8266 อัพค่าขึ้น thingspeak.com กันเถอะ

สวัสดีครับ วันนี้เราจะมา อธิบายการส่งค่า อุณหภูมิ ไปยัง thingspeak.com เพื่อเก็บข้อมูล โดยใช้ NodeMCU Esp8266 และ DHT11 เก็บค่าอุณหภูมิ เป็นกราฟ ครับ

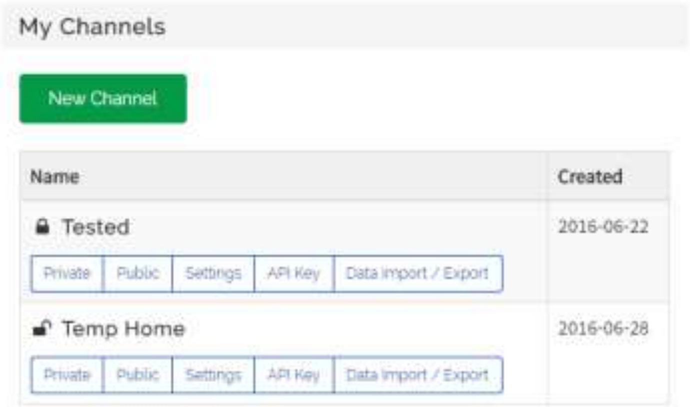
เริ่มต้นจาก สมัคร สมาชิกเว็บไซต์ thingspeak.com อ่านบทความเพิ่มเติมได้ที่นี้

อุปกรณ์ที่ต้องใช้ มีดังต่อไปนี้

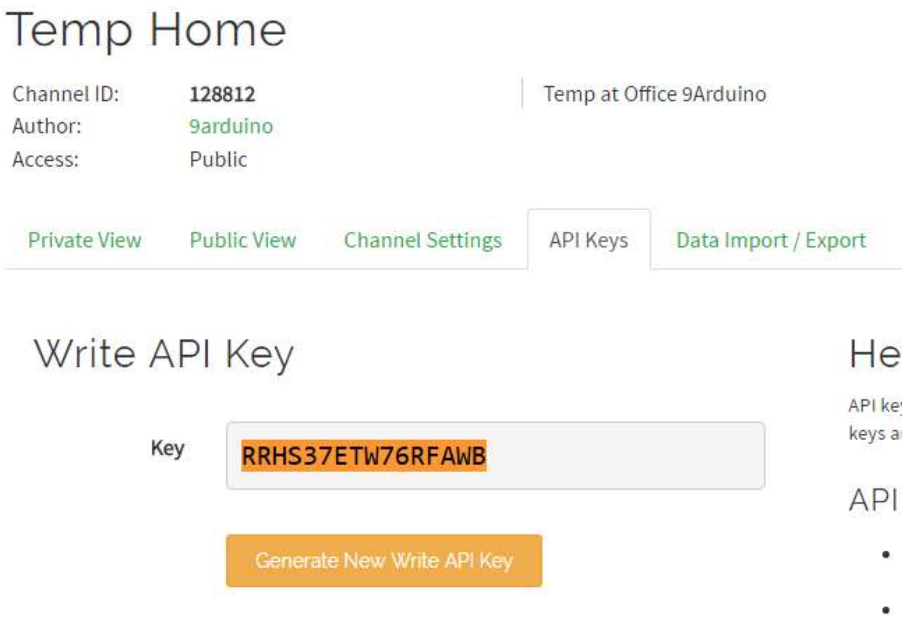
1. Nodemcu Esp8266
2. Sensor อุณหภูมิ DHT11 หรือจะใช้ DHT 21 DHT 22 ก็ได้ แล้วแต่เพื่อนๆชอบ

มาเริ่มต้นสร้างช่อง Channel บน thingspeak.com

1. เริ่มจาก Login เข้าระบบ แล้ว กดที่ปุ่ม New Channel



2. จากนั้น ตั้งค่าช่องของคุณตามต้องการ (ข้อมูลหลักๆ ที่ต้องการ ดังนี้)
   1. Name ชื่อช่อง Channel
   2. Description คำอธิบาย
   3. Field 1 - Field 8 คือ ชื่อของตาราง thingspeak.com ยอมให้สร้างได้ 8 กราฟ
   4. Make Public ต้องการให้แสดงเป็น สาธารณะหรือไม่
3. เมื่อสร้างช่อง Channel เสร็จคราวนี้ ก็คลิกไปยังช่องที่เราสร้างขึ้นแล้วกด ที่แถบ API Key



หลังจากนั้นให้เรา จำ ที่ Write API Key จากรูปคือ RRHS37ETW76RFAWB เก็บค่านี้ไว้นำไปแก้ใน Code ของ Nodemcu esp8266

คราวนี้ต่อวงจรกัน ใช้งานดังนี้

NodeMCU Esp 8266 Sensor DHT11

1. Pin VCC 5V VCC
2. Pin Gnd Gnd
3. D4 Out

Code ตัวอย่าง

```
#include <ESP8266WiFi.h>
#include "DHT.h"
#define DHTPIN D4   // ขา Out ของ Sensor ต่อเข้าขา D4 ของ Esp8266
//เลือกชนิดของ Sensor
#define DHTTYPE DHT11     //DHT 11
//#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21   // DHT 21 (AM2301)
DHT dht(DHTPIN, DHTTYPE);
const char* ssid     = "SSID Wifi";    // SSID Wifi
const char* password = "Password Wifi";   // Password Wifi
const char* host = "api.thingspeak.com";    // Host ของ thingspeak ไม่ต้องแก้ไข
const char* api   = "RRHS37ETW76RFAWB";  //API Key ที่เราจำไว้ ในขั้นต้นเมื่อกี้

void setup() {
  Serial.begin(9600);
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("DHTxx test!");
  dht.begin();
}

int value = 0;

void loop() {
  delay(5000);
  ++value;
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");
  Serial.print("Heat index: ");
  Serial.print(hic);
  Serial.print(" *C ");
  Serial.print(hif);
```

```
  Serial.println(" *F");
  Serial.print("connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }

  // We now create a URI for the request
  String url = "/update?api_key=";
  url += api;
  url += "&field1=";
  url += t;
  url += "&field2=";
  url += h;
// เราจะส่งข้อมูล https://api.thingspeak.com/update?api_key=RRHS37ETW76RFAWB&field1=(อุณหภูมิ)&field2=(ความชื่น)
  Serial.print("Requesting URL: ");
  Serial.println(url);
  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  delay(10);
  // Read all the lines of the reply from server and print them to Serial
  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
  Serial.println();
  Serial.println("closing connection");
}
```

ค้นหา Address I2C อุปกรณ์ ด้วย Arduino

มีน้องๆเข้ามาปรึกษา เรื่อง Address บ่อยมากครับเกี่ยวกับ Sensor Module ต่างๆ ที่เชื่อมต่อผ่านทาง I2C IIC (SDA SCL) อุปกรณ์ซื้อไปแล้วใช้ไม่ได้บ้าง ไม่ยอให้ทำงานตามที่สั่ง อุปกรณ์ที่จะเจอบ่อย สุดคือจอภาพ LCD แบบ I2C เพราะหลักๆทางร้าจะขายอยู่ 2 Address คือ 0x27 หรือ 0x3F วันนี้เราก็จะมาคุยเรื่อง นี้กันเลยดีกว่า ครับแบบเจาลึกสักหน่อย

หลักการของ I2C IIC ถ้าเรียบเสมือนก็คือ เหมือนเราเดินเข้าไปในห้างสรรพสินค้า แล้วมีพนักงานภายในห้าง ประกาศว่า รถยนต์ ป้ายทะเบียน 1กก 35 กทม ไปเลื่อยรถด้วยค่ะ ทุกคนในห้างได้ยินกันหมด แต่รถยนต์ป้ายทะเบียน 1กก 35 กทม เป็นของน้องไหม ถ้าใช่น้องก็จะเดินไปเลือนรถ ตามพนักงานแจ้ง แต่ถ้าไม่ใช่น้องๆก็จะอย่เฉยๆ เดินเล่นในห้างต่อไป หลักการก็ประมาณนี้ครับ ป้ายทะเบียนก็เหมือน Address ของเรานั้นเอง งงป่าว บางทีแอดอ่านก็งงเอง 55+

เรามาเข้าเรื่องของเรากันเลยดีกว่า ครับ วันนี้จะทำการยกตัวอย่างการหา Address ของจอ LCD แบบ I2C น่ะครับ

การต่อวงจร ดังต่อไปนี้เลย

Arduino Uno	Module I2C
A4	SDA
A5	SCL
5V	5V
GND	GND

มาในส่วนของ Code กันเลย

```
// I2C Scanner
// Written by Nick Gammon
// Date: 20th April 2011
/*
  จากบทความ ค้นหา Address I2C ด้วย Arduino กันเถอะ
  https://www.9arduino.com/article/69/
*/
 #include <Wire.h>
void setup() {
  Serial.begin (9600);
  // Leonardo: wait for serial port to connect
  while (!Serial)
    {
    }
  Serial.println ();
  Serial.println ("www.9arduino.com ...");
```

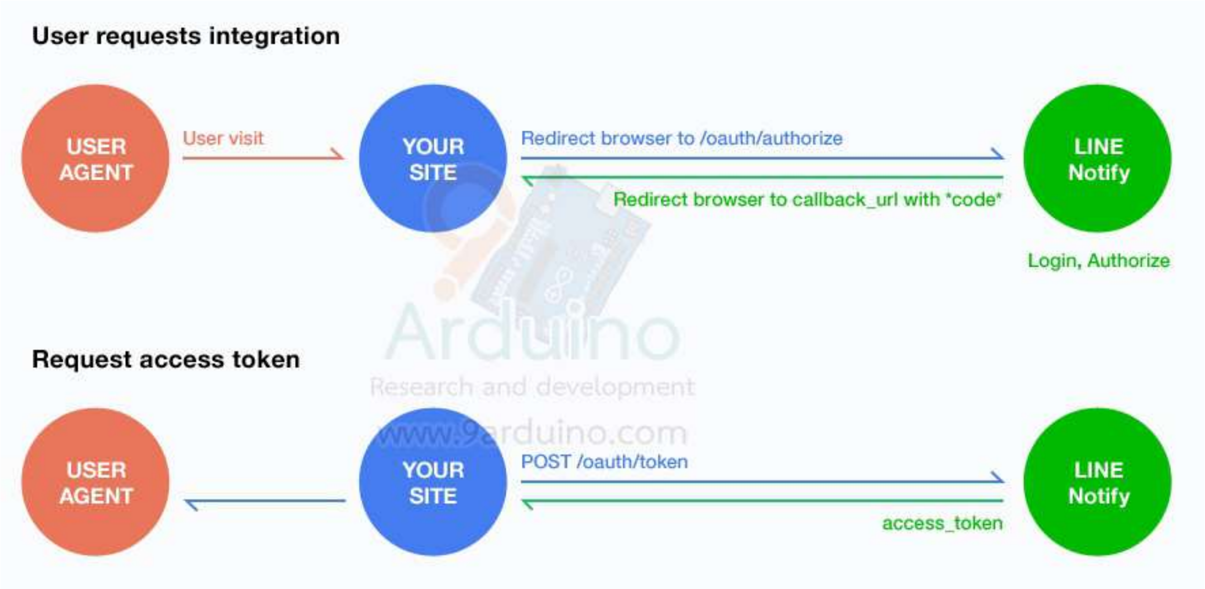
```
  Serial.println ("I2C scanner. Scanning ...");
  byte count = 0;
  Wire.begin();
  for (byte i = 8; i < 120; i++)  // Loop ค้นหา Address
  {
    Wire.beginTransmission (i);
    if (Wire.endTransmission () == 0)
      {
      Serial.print ("Found address: ");
      Serial.print (i, DEC);
      Serial.print (" (0x");
      Serial.print (i, HEX);
      Serial.println (")");
      count++;
      delay (1);
      }
  }
  Serial.println ("Done.");
  Serial.print ("Found ");
  Serial.print (count, DEC);
  Serial.println (" device(s).");
}
void loop() {
}
```

Nodemcu Esp8266 กับการส่งการแจ้งเตือนเข้า LINE ส่งข้อความ ภาพ และสติกเกอร์

ทราบกันดีว่าปัจจุบัน App Line มีการเปิด API ให้อุปกรณ์ Arduino Nodemcu Esp8266 สามารถเชื่อมต่อ Line แล้วส่งข้อความแจ้งเตือนผ่าน Line ก็ได้ มาสักระยะแล้ว วันนี้เราได้นำบทความที่เป็นปัจจุบันล่าสุด นำมาสอนให้น้องๆได้ใช้กันแบบไม่มีปัญหาน่ะครับ

หลักการทำงานของ Line notify

อุปกรณ์ Nodemcu Esp8266 Esp32 Arduino+W5100 จะทำการส่งข้อความ ผ่าน Methol Post ไปยัง Server ของ Line NotiFy อุปกรณ์จะต้องรองรับ SSL ด้วยน่ะครับ



บทความนี้เราจะมาพูดถึงการสมัคร Line Notify กันครับ ก่อนอื่นน้องๆจะต้องมี ID Line ก่อนน่ะครับ Email และ Password ที่เข้า Line ได้ อันนี้ทางร้านจะไม่ขอพูดน่ะครับเพราะเชื่อว่าทุกคนทำได้อยู่แล้ว เรามาเริ่มกันเลย ครับ

1. ก่อนอื่นเปิดเว็บ นี้ก่อนเลย ครับ https://notify-bot.line.me/th/

2. จากนั้นจะขึ้น ตามภาพ ให้กดที่ปุ่ม เข้าสู่ระบบ ที่มุมขวาบน

3. Login ด้วย Email และ Password Line ของเราเข้าไป

4. ให้ทำการกดที่ ออก Token ตามภาพ

5. ในส่วนนี้ จะเป็นการตั้งชื่อ และให้ส่งข้อความ หาเราเองหรือ ส่งข้อความเข้ากลุ่ม ที่เราต้องการ

บริเวณสีแดง เป็นส่วนที่ให้เราใส่ชื่อของ Bot ของเราคงไป อาจจะตั้งชื่อว่า Robot แจ้งเตือนไฟไหม้ น้ำร้องล่วงอันนี้ก็แล้วแต่น้องๆน่ะครับ

บริเวณสีชมพู หากเราเลือกในช่องนี้ คือให้ Nodemcu เลือกส่งข้อความหาเราเองเท่านั้น

บริเวณสีเขียว เป็นกลุ่มที่เราต้องการให้ Nodemcu ส่งข้อความไปหากลุ่มของเรา

จากนั้นให้ทำการกดปุ่ม ออก Token (อันนี้ Admin ก็ไม่เข้าใจทำไมต้องแปลออกมาคำว่าออก Token ทำไมไม่ใช้คำว่าสร้าง จะได้เข้าใจง่ายขึ้น)

6. จากนั้นให้เรา คัดลอก Token เก็บไว้ เราจะเอาส่วนนี้ไปใช้ในโปรแกรมของเรา

จบแล้วครับ ง่ายไหมเอย เดี๋ยวมาลุยกันต่อในส่วนของ Code

จากครั้งที่แล้วเราได้พูดถึงการสมัคร Line Notify เรียบร้อย อย่าลืม Copy ในส่วนของ Token ด้วยน่ะครับเราต้องใช้ด้วยน่ะครับ เรามาเริ่มกันเลยทางร้านแนะนำให้ใช้ Library ของ พี่ TridentTD ชื่อว่า

https://github.com/TridentTD/TridentTD_LineNotify

เรามาเริ่มกันเลย ครับ

1. ก่อนอื่นให้เรา Download Library มาติดตั้งในเครื่องก่อนน่ะครับ

Download : ติดตั้งไม่เป็นสามารถอ่านบทความ เพิ่มเติมได้น่ะครับ https://www.ab.in.th/b/32 (ใช้วิธีที่ 2 ติดตั้งแบบ Winzip)

2. มาในส่วนของ Code อันนี้เป็นส่วนหนึ่งของ ใน Code ตัวอย่างน่ะครับ

```
/*
 * ไลบรารี่ TridentTD_LineNotify version 2.1
 * ใช้สำหรับ ส่ง แจ้งเตือนไปยัง LINE สำหรับ ESP8266 และ ESP32
 * สามารถส่งได้ทั้ง ข้อความ , สติกเกอร์ และรูปภาพ(ด้วย url)
 * -----------------------------------------------------
 * ให้ save เป็น file ต่างหากก่อนถึงจะส่ง Line Notify ภาษาไทยได้
 */
```

```
#include <TridentTD_LineNotify.h>
#define SSID        "-------------------"
#define PASSWORD    "-------------------"
#define LINE_TOKEN  "-------------------"

void setup() {
  Serial.begin(115200); Serial.println();
  Serial.println(LINE.getVersion());
    WiFi.begin(SSID, PASSWORD);
  Serial.printf("WiFi connecting to %s\n",  SSID);
  while(WiFi.status() != WL_CONNECTED) { Serial.print("."); delay(400); }
  Serial.printf("\nWiFi connected\nIP : ");
  Serial.println(WiFi.localIP());
  // กำหนด Line Token
  LINE.setToken(LINE_TOKEN);
  // ตัวอย่างส่งข้อความ
  LINE.notify("อุณหภูมิ เกินกำหนด");

  // ตัวอย่างส่งข้อมูล ตัวเลข
  LINE.notify(2342);          // จำนวนเต็ม
  LINE.notify(212.43434,5);   // จำนวนจริง แสดง 5 หลัก

  // เลือก Line Sticker ได้จาก https://devdocs.line.me/files/sticker_list.pdf
  LINE.notifySticker(3,240);        // ส่ง Line Sticker ด้วย PackageID 3 , StickerID 240
  LINE.notifySticker("Hello",1,2);  // ส่ง Line Sticker ด้วย PackageID 1 , StickerID 2  พร้อมข้อความ

  // ตัวอย่างส่ง รูปภาพ ด้วย url
  LINE.notifyPicture("https://preview.ibb.co/j6G51n/capture25610417181915334.png");
  LINE.notifyPicture("จตุ
ธาตุ","https://www.fotoaparat.cz/storage/pm/09/10/23/670915_a5351.jpg");
}
void loop() {
  delay(1);
}
```

ในส่วนของ Code ลองปรับแต่งกันดูได้เลย ครับ คำอธิบายหลังคอมเม้นเรียบร้อยแล้ว

Fritzing ซอฟต์แวร์ออกแบบวงจรและแผ่นปริ้นให้กับ Arduino

โปรแกรม Fritzing เป็นโปรแกรมที่ช่วยในการออกแบบวงจรสำหรับบอร์ดต่างๆ เช่น RaspberryPi ,Arduino รุ่นต่างๆช่วยให้วางในตำแหน่งที่เหมาะสม ที่สำคัญยังเป็นโปรแกรมฟรี (ซอฟต์แวร์โอเพนซอร์ส) รองรับทั้ง Windows 32Bit , Windows 64Bit, Linux และ MacOS ช่วยในการออกแบบวงจรลงบน BreadBoard วาดวงจร Schemetic และการออกแบบแผ่นปริ้น (PCB)

โปรแกรมนี้เหมาะกับการนำไปใช้งานออกแบบวงจรเพื่อแนบในเล่มเอกสารโปรเจค Arduino วาดวงจรโปรเจค หรือทำรายงานส่งคุณครูทางร้านแนะนำตัวนี้เลย ครับใช้งานง่าย มีอุปกรณ์เยอะแนะมากมาย

คุณสมบัตรพื้นฐานการทำงานของตัวโปรแกรม

1. จำลองการสร้างวงจรจริง ขึ้นบน Breadboard
2. สามารถทำการ Rebild วงจรที่สร้างในโปรแกรม Reitzing
3. แก้ไขลายวงจร ให้ถูกต้อง
4. เปลี่ยนแปลงคุณสมับติของอุปกรณ์เช่น ค่าของตัวต้านทาน ตัวเก็บประจุ เบอร์ของ ทราซิตเตอร์
5. สามารถออกแบบ Design PCB โดยการลากวางอุปกรณ์ลงไปในตามตำแหน่งที่ต้องการบน PCB
6. สามารถนำโปรเจคของเราไปแชร์บน Internet ได้ทันที

Fritzing สามารถ Download ได้ที่ http://fritzing.org/ สามารถใช้ได้หลายระบบปฏิบัติการ เช่น Windows Mac OS และ Linux การติดตั้งไม่ยาก สำหรับ Windows สามารถแตกไฟล์ Winzip แล้วสามารถใช้งานได้ทันที

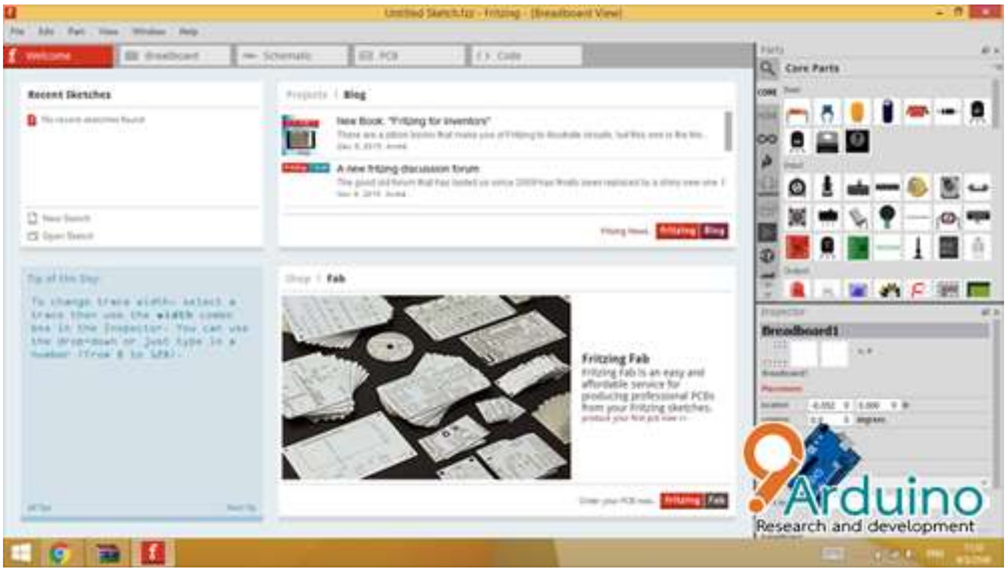
เพื่อความรวดเร็วในการดาวน์โหลดทาง AB.in.thได้นำไฟล์มาเก็บไว้ใน Server ส่วนตัว สามารถ Download ได้ ดังนี้

Windows 32 Bit : http://download.ab.in.th/files/fritzing32.zip

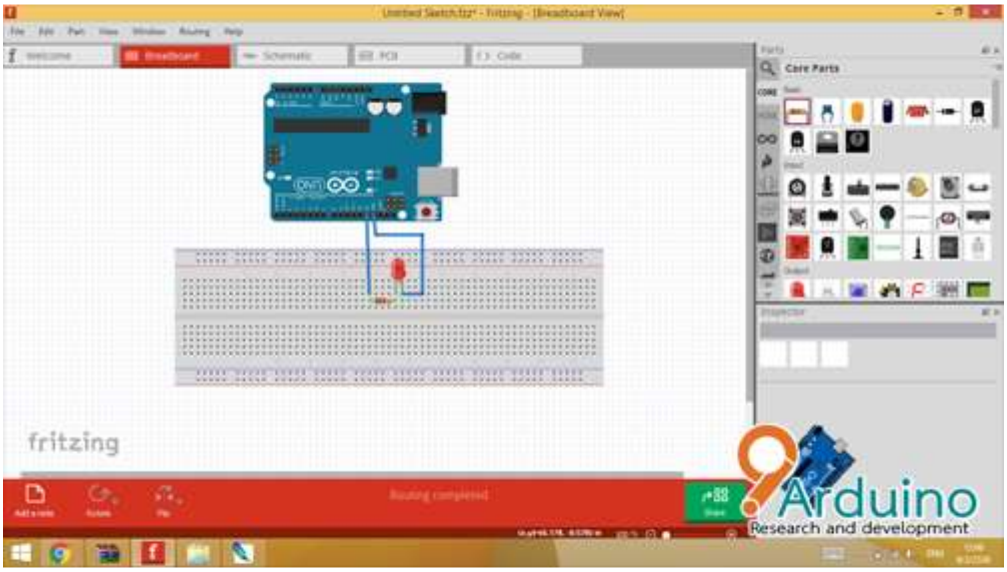
Windows 64 Bit : http://download.ab.in.th/files/fritzing64.zip

Version 0.9.3b was released on June 2, 2016.

1. เริ่มต้นเปิดโปรแกรม หน้าตาดังนี้

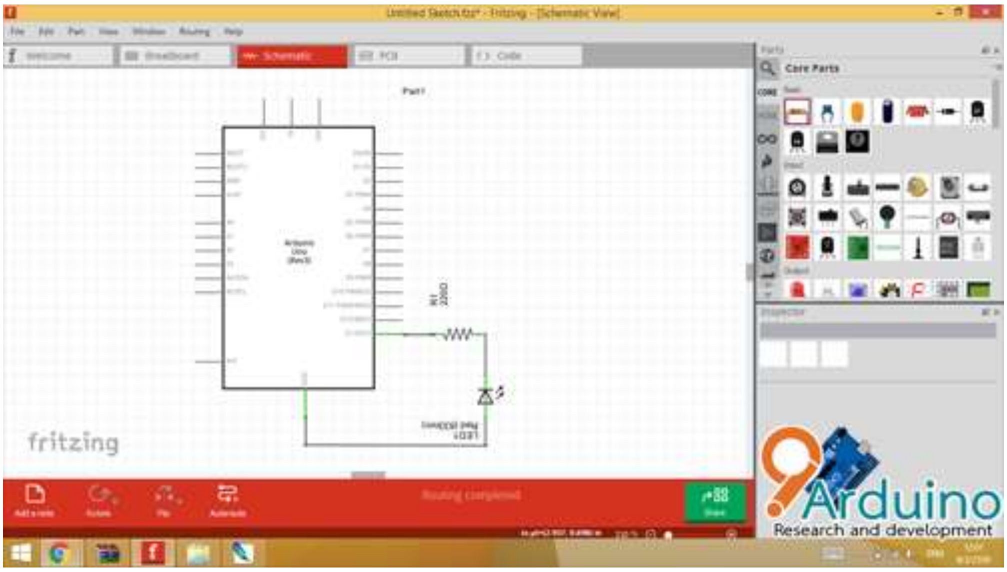


2. ออกแบบวงจร ลงใน BreadBoard

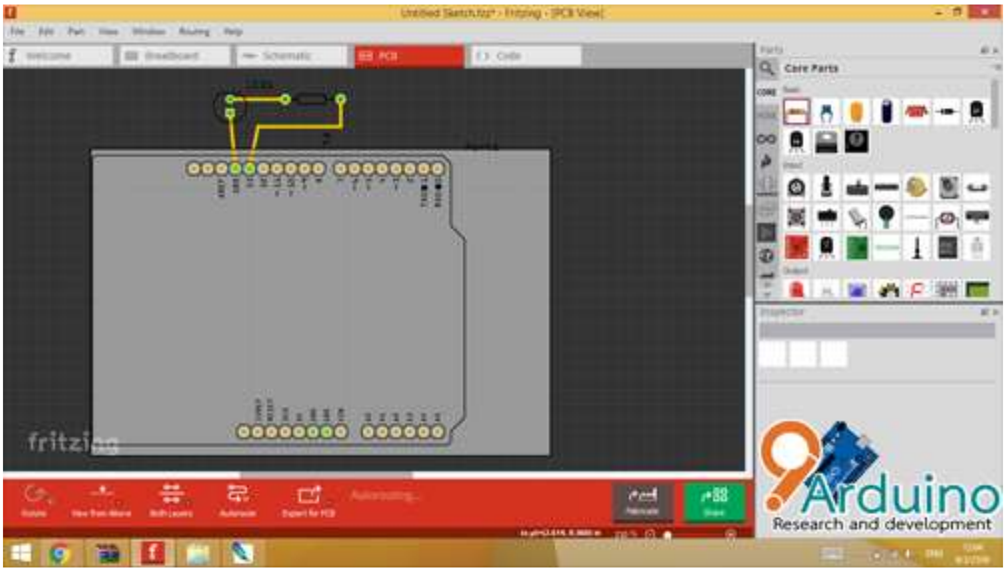


3. เครื่องมือและอุปกรณ์ มีให้เลือกมากมาย และสามารถเพิ่มเติมได้อีกเลื่อยๆ

4. การวาดวงจร Schemetic



5. การออกแบบแผ่นปริ้น



สำหรับ ตัวอุปกรณ์บางชิ้น ที่น้องๆไม่สามารถหาลูกค้าของทางร้านสามารถขอแบ่งจากทางร้านได้นะครับ ทางร้านเก็บไว้ค่อนข้างเยอะเลยไม่ว่าจะเป็นพวก Nodemcu Arduino Sensor ต่างๆ ไม่ว่าจะเป็น Dht11 Ds18b20 Dht22 Soil Sensor Pir เป็นต้น

แก้ปัญหา wdt reset ของ Nodemcu esp8266 เขียน Code พัฒนาด้วย Arduino ide

สำหรับเพื่อนๆ ที่เขียนโปรแกรม กับเจ้า Nodemcu esp8266 ด้วย Arduino IDE มักจะเจอปัญหากับ wdt reset ส่วนตัวผมก็เจอปัญหานี้ประจำ วันนี้จะมีวิธีการสังเกต Code มาฝาก กันครับ ว่า wdt reset เกิดขึ้นเพราะอะไร

จากข้อความนี้

ets Jan 8 013,rst cause:2 boot mode:(1,6)

wdt reset

ให้สักเกตที่คำว่า cause:2 จดตัวเลขนี้ไว้ครับ มาดูคำอธิบายกัน ครับว่าคืออะไร

Cause:0 Undefined

Cause:1 Power reboot

Cause:2 External reset wake up Deepsleep

Cause:4 Hardware Wdt reset

ปัญหาพวกนี้บางคนอาจจะเคยเจอแน่นอนครับ บอกเลยว่าปวดหัวสุดๆ

ส่วนตัวมักจะ ชอบเจอ Cause:2 ครับ มักจะเกิดจาก การตั้ง Delay นานเกินไป หรือเร็วเกินไป บางทีลืมใส่ Delay ในชุดที่สำคัญครับ วิธีการแก้ปัญหาของผม (อันนี้ส่วนตัวผิดถูกขออภัยด้วยน่ะครับ)

ขั้นแรก พยายามแบ่ง ส่วน Process ไว้ เป็นส่วนๆ โดยการ Debug ส่วนตัวจะทำแบบโง่ๆเลยครับ โดยการ serial.print ไว้ตาม Process ของโปรแกรม if while for เพื่อหาจุดที่เกิดปัญหา ว่าเกิดจากจุดไหน ไล่สาเหตุก่อน เมื่อจับจุดได้แล้ว สังเกตุดู แล้วหาทางแก้ดูครับ จากเหตุการที่บ่อยๆ มีดังนี้ครับ

กรณีที่ 1 หากเกิดจาก จุดที่ delay เป็นระยะเวลานานๆ ให้ ทำการแก้โดย ยกตัวอย่างเช่น

```
delay(6000);
```

เราอาจจะแก้ โดยการใช้ คำสั่ง For เข้าไปแทน

```
for (int i=0; i <= 600; i++){
    delay(10);
}
```

กรณีที่ 2 อาจจะ delay เร็วเกินไป จาก Code เช่น ตัวอย่างการกดปุ่มแช่ โดยการใช้ while โปรแกรมจะ while ใน Loop นี้จนกว่าจะกดปล่อย ถึงจะหลุด Loop ยกตัวอย่างเช่น

```
while (digitalRead(buttonPin) == LOW)
```

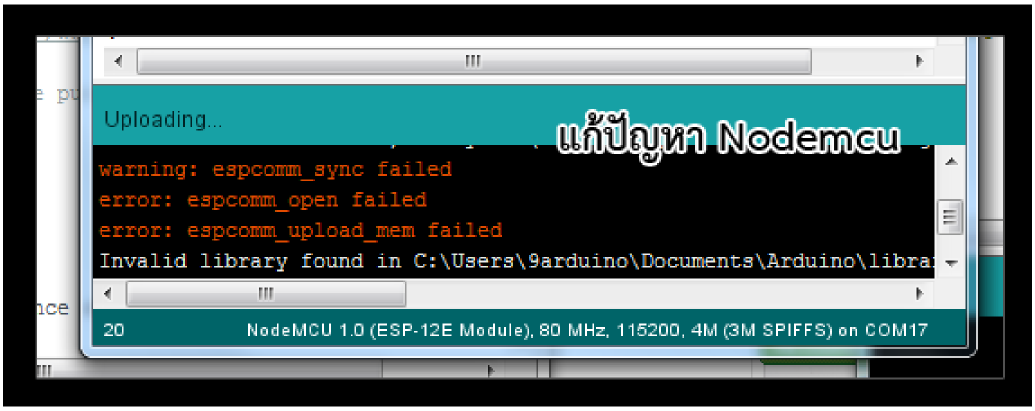
ในส่วนของคำสั่งนี้ ความคิดส่วนตัวน่ะครับ อาจจะทำให้ Nodemcu loop เร็วเกินไป จน Ram หมด ทำให้ wdt reset เราอาจจะแก้ โดยการใส่ Delay ให้มันหน่อย เพื่อลดความเร็วใน Loop นี้ครับ

```
while (digitalRead(buttonPin) == LOW) {delay(10);}
```

จาก Code นี้หากใช้ร่วมกับ Arduino มักจะไม่ค่อยเจอปัญหา แต่พอมาใช้ กับ nodemcu esp8266 จะเจอปัญหานี้ประจำครับ

หลักจากประสบการณ์ที่เคยเจอ ย้ำน่ะครับว่าเป็นความคิดเห็นส่วนตัว จากตัวเอง อาจจะไม่ถูกต้อง 100 เปอร์เซ็นก็ขออภัยด้วย ครับ

แก้ปัญหา NodeMcu Esp8266 ขึ้น error: espcomm_upload_mem failed



แก้ปัญหา NodeMcu Esp8266 ขึ้น warning: espcomm_sync failed

สำหรับน้องๆ มือใหม่ที่กำลังหัดเล่น Nodemcu esp8266 เป็นครั้งแรก อาจจะเจอปัญหา Upload ไม่เข้า ขึ้น Error แสดงข้อความว่า ตามรูปภาพ ไม่ต้องตกใจน่ะครับ

warning: espcomm_sync failed

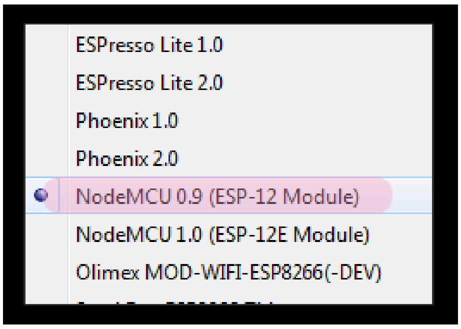
error: espcomm_open failed

error: espcomm_upload_mem failed

วันนี้ พี่จะแนะนำวิธีการแก้ปัญหา ครับ เบื้องต้นให้น้องครับ มาเริ่มกันเลย

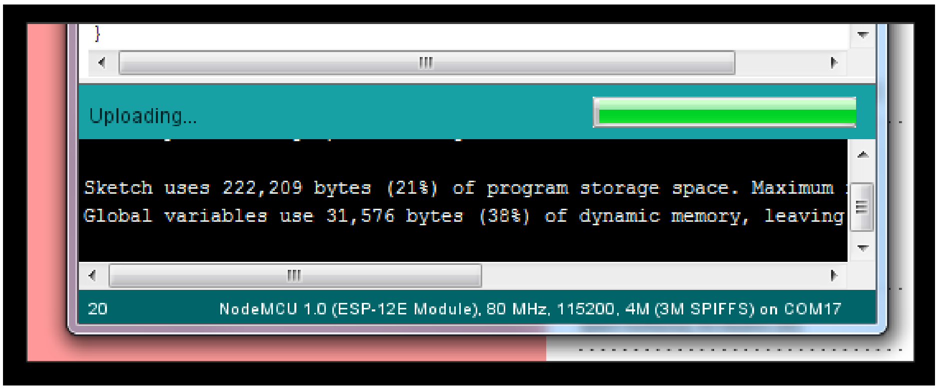
สิ่งแรก ตรวจสอบ Com port ก่อน ขึ้นหรือไม่ หากไม่ขึ้นให้ลงไดร์เวอร์ให้ถูกต้อง

วิธีที่ 1. เปลี่ยนมาเลือกบอร์ด Nodemcu เป็น เวอร์ชั่น 0.9 แทน แต่หากอัพโหลดแล้ว เข้าแต่ไม่ทำงาน ให้เลือก Nodemcu 1.0 แทนครับ



ส่วนใหญ่เมื่อเลือกบอร์ดนี้ก็จะจบปัญหาหมดปัญหาครับ

วิธีที่ 2. หากจำเป็นต้องการใช้บอร์ดเวอร์ชั่น 1.0 หรือ Board รุ่นอื่นๆ เมื่อ กด Upload รอจนหลอด Upload ตามภาพเต็ม



ให้กดปุ่ม FLASH บนบอร์ด ของ Nodemcu 1 ครั้ง ก็จะทำการอัพโหลดโปรแกรมให้

วิธีที่ 3. ถอดอุปกรณ์ ออกให้หมด ลอยขา Nodemcu ไว้ ไม่ต้องต่ออุปกรณ์ใดๆ อัพเสร็จค่อยเสียบอุปกรณ์ เพราะบางครั้งอาจจะไปโดนขาที่ห้ามใช้งาน ทำให้อัพโหลดไม่ได้เช่นกัน ครับ

วิธีที่ 4. หลายๆท่านมักจะชอบเลือกบอร์ดผิด เป็น Generic ESP8266 Module หากเลือกบอร์ดนี้ ในส่วนของการ Auto flash จะไม่ทำงาน ให้กดปุ่ม Flash แบบวิธีที่ 2 ด้วยครับ หรือเลือกบอร์ดใหม่ เป็น Nodemcu 1.0 ก็จะสามารถใช้งานได้ปกติ

เพิ่มเติมให้อีกนิด

บางครั้งให้ลองเปลี่ยนสาย USB ด้วยน่ะครับ เพราะเคยเจอสาย บางรุ่นเช่น blackberry เจอ Comport แต่ไม่สามารถอัพโหลดโปรแกรมได้

หากทำตามวิธีที่กล่าวมาขั้นต้นแล้วไม่สามารถอัพโหลด ให้ส่งสัยว่าสินค้ามีปัญหาหรือเสียครับ

แก้ปัญหา ใช้งาน Nodemcu ไม่ได้ package esp8266 is unknown

สวัสดีครับ หลังจากที่ไม่ได้เขียนบทความหลายเดือนเลย วันนี้มีน้องๆ ที่มาซื้อบอร์ด Nodemcu Esp8266 แล้วไม่สามารถติดตั้งบอร์ด แล้ว Compile ได้ วันนี้ทางร้านเลยเลือกมาเขียนบทความไว้เลยแล้วกันครับ (ไม่พูดมากเจ็บคอ มาเริ่มกันเลย)

ข้อความ Error เวลา Complie จะแสดงว่า

Board generic (platform esp8266, package esp8266) is unknown

Error compiling for board Generic ESP8266 Module.

หรือ

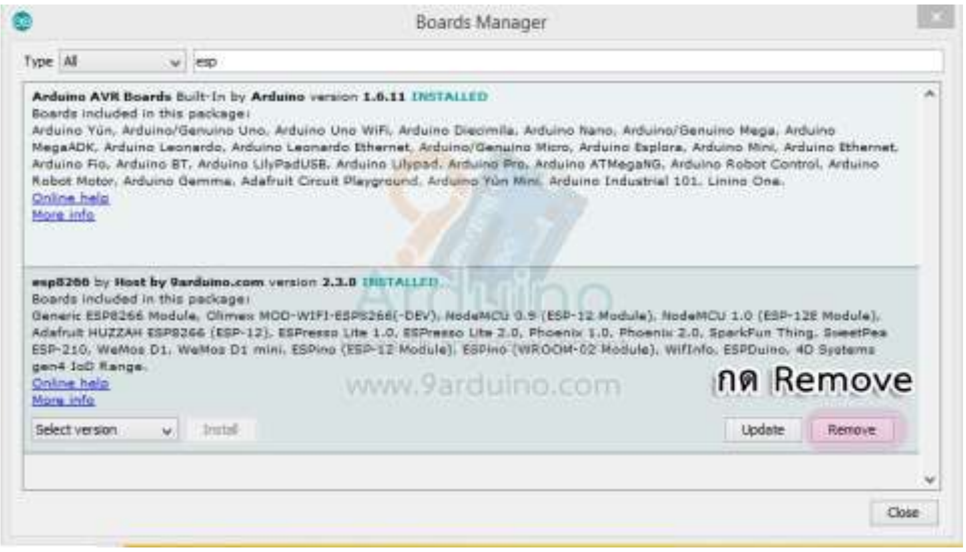
Board nodemcu (platform esp8266, package esp8266) is unknown

Error compiling for board Generic ESP8266 Module.

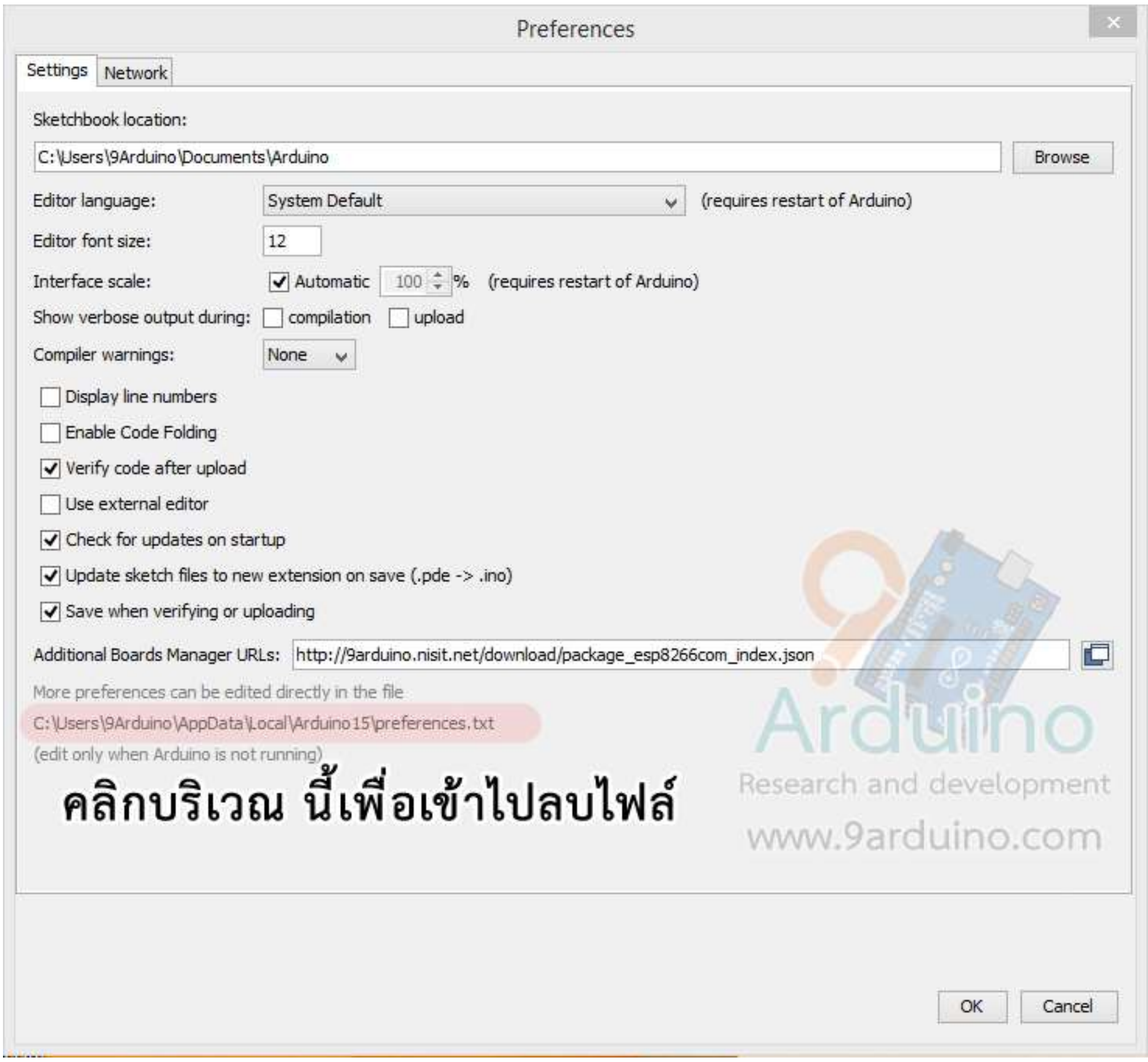
สาเหตุไม่แน่ชัดว่าเกิดจากอะไร แต่ถ้าให้ทางร้านเดาก็น่าจะเกิดจากไฟล์ที่ติดตั้งมีปัญหา โหลดไม่ครบ ไฟล์เสียหาย ก็เป็นไปได้หมดทุกทางครับ มาเริ่มกันเลย

1. ให้ทำการ ลบการติดตั้งบอร์ด esp8266 ก่อนเลยเป็นอันดับแรก

กดไปที่ Tools > Board > Boards Manager จากนั้นทำการ Remove esp ที่เราติดตั้งออกให้หมด

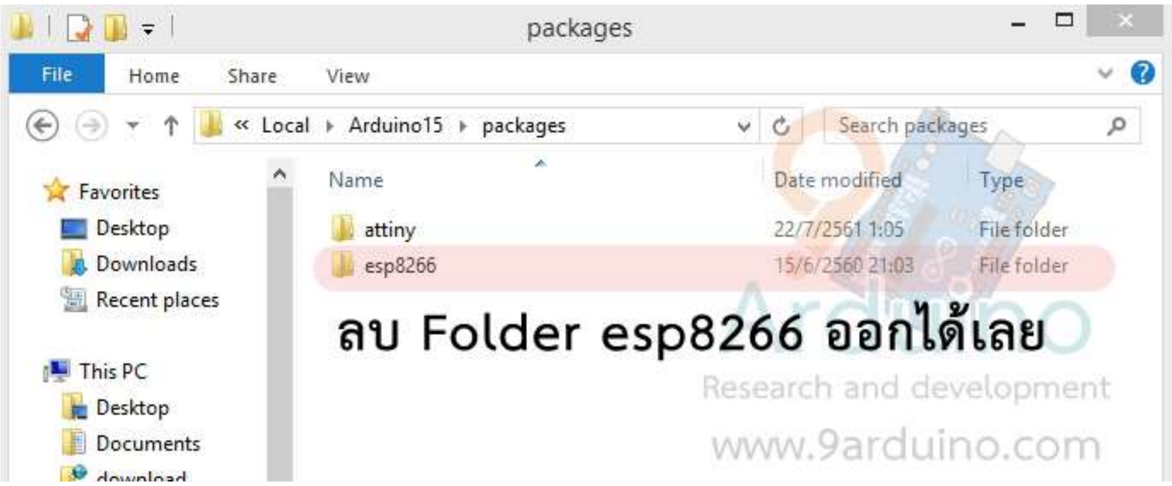


2. ให้เข้าไปที่ File > Preferences



จากนั้นให้ทำการกด บริเวณตามรูป เพื่อเข้าไปยังที่ติดตั้ง Arduino

3. ให้เข้าไปที่ Folder Packages จะเจอ Folder esp ให้ทำการลบทิ้งได้เลย อย่าให้เหลือซาก หากลบไม่ได้ให้ทำการปิดโปรแกรม Arduino IDE ออกด้วยน่ะครับ



4. ทำการติดตั้งใหม่อีกรอบ ตาม ลิงค์นี้ได้เลย https://www.ab.in.th/b/38 รับประกันจะสามารถใช้งานบอร์ด Arduino ได้แน่นอน ครับ

รถบังคับ Rc Car wifi ด้วย Nodemcu esp8266 Arduino IDE ควบคุมผ่านมือถือ android

วันก่อนมีน้องๆ เข้ามาขอคำปรึกษาที่ร้าน โดยเขาจะทำรถบังคับควบคุมผ่าน Wifi ซึ่งบทความส่วนใหญ่ จะเป็นบทความเกี่ยวกับ รถบังคับโดยใช้ Arduino + bluetooth HC06 เป็นหลัก ระบบเดิมที่น้องเขาได้พัฒนา เป็นการควบคุมผ่าน App โดยการเชื่อมต่อ TCP มักจะเจอปัญหา Nodemcu เอ๋อบ้าง ช้าบ้างไม่ทันใจวัยรุ่น เลยให้คำปรึกษาน้องไปว่าให้ใช้ UDP แทน เนื่องจากการส่งข้อมูลของ TCP จะเป็นลักษณะการสนทนามีการตอบกลับ ทำให้การส่งข้อมูลทำได้ช้ากว่า UDP เนื่องจาก UDP จะส่งข้อมูลออกไปทิศทางเดียวไม่มีการติดต่อกลับมาจึงทำให้มีความเร็งสูงกว่า TCP จากที่ให้คำปรึกษาน้องๆไป เลยนำมาเขียนบทความเรื่องรถบังคับผ่าน Wifi ด้วย Nodemcu esp8266 Arduino IDE.

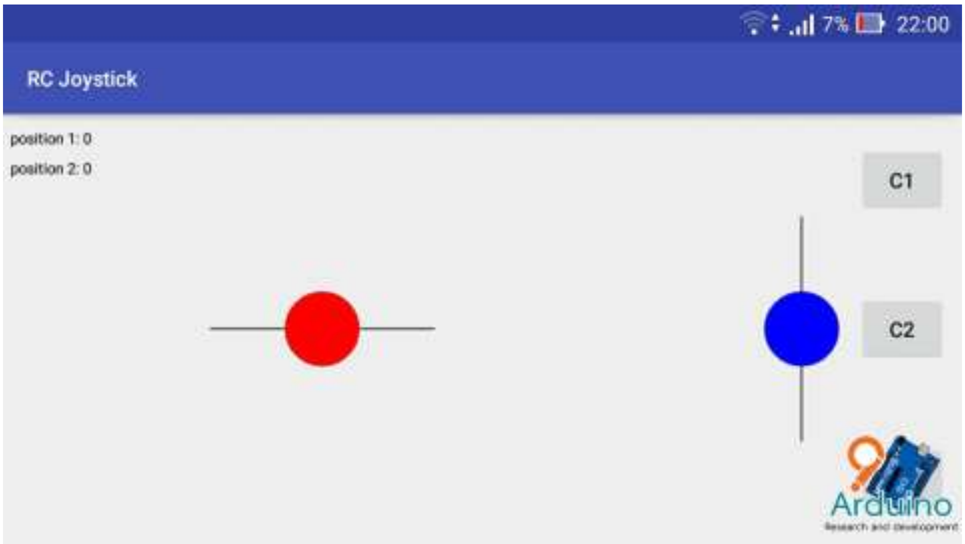
มาเริ่มทำกันเลย !

เตรียมอุปกรณ์

1. Nodemcu V2
2. รถบังคับเก่า (เป็นรถดริฟ ของเล่นเก่า)
3. Drive Motor เลือกใช้สินค้า L298 Dual Motor Driver Module 2A (ไดร์ขับมอเตอร์) รหัสสินค้า F01001
4. อุปกรณ์อื่นๆ เช่นสายไฟ เป็นต้น

เพื่อให้เป็นการไม่เสียเวลา ขี้เกียดเขียนเว็บด้วย จึงหา App android ที่ GUI ใช้งานได้ง่ายๆ แล้วมาดัก Packet การส่งเอาเพื่อเอามาใช้กับงานของเราจึงแนะนำ App ที่มีชื่อว่า RC Joystick

หน้าตาของ App RC Joystick



หลักการทำงานของรถบังคับดริฟ คันนี้ จะแบ่งออกเป็น 2 ชุดคือ

1. ชุดบังคับเลี้ยวซ้าย ขวา
2. ชุดมอเตอร์ขับเคลื่อน เดินหน้า ถอยหลัง

ถอดวงจรที่มีอยู่ออกมาให้หมดแล้วทำการบัดกรีสายไฟ เลี้ยวซ้ายขวา และเดินหน้า ถอยหลัง 2 เส้น เตรียมไว้ครับ

การต่อวงจรของชุดรถบังคับของเราต่อได้ดังนี้

ส่วนของ Code ของ Nodemcu IDE เราจะใช้ Arduino IDE ในการพัฒนา Code มีดังนี้

```
/*
Arduino RC Car Drift
By Pakorn  Rattanaying
https://www.9arduino.com
ตามบทความ  https://www.9arduino.com/article/53/
*/
#include <ESP8266WiFi.h>
#include <WiFiUDP.h>
int status = WL_IDLE_STATUS;
const char* ssid = "9arduino";  //  your network SSID (name)
const char* pass = "Pass wifi";       // your network password
unsigned int localPort = 2390;      // Port สำหรับการเชื่อมต่อ
byte packetBuffer[512]; //buffer to hold incoming and outgoing packets
// A UDP instance to let us send and receive packets over UDP
WiFiUDP Udp;

int a;
int b;
int c;
int d;
int pw1;
int pw2;

const int pin1 = D1;
const int pin2 = D2;
const int pin3 = D3;
const int pin4 = D4;
const int pwm1 = D5;
const int pwm2 = D6;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
```

```
  pinMode(pin1, OUTPUT);
  pinMode(pin2, OUTPUT);
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  pinMode(pwm1, OUTPUT);
  pinMode(pwm2, OUTPUT);

        digitalWrite(pin1, HIGH);
        digitalWrite(pin2, HIGH);
        digitalWrite(pin3, HIGH);
        digitalWrite(pin4, HIGH);

  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  // setting up Station AP
  WiFi.begin(ssid, pass);

  // Wait for connect to AP
  Serial.print("[Connecting]");
  Serial.print(ssid);
  int tries=0;
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    tries++;
    if (tries > 30){
      break;
    }
  }

  Serial.println();
printWifiStatus();
  Serial.println("Connected to wifi");
```

```
  Serial.print("Udp server started at port ");
  Serial.println(localPort);
  Udp.begin(localPort);
}

void loop()
{
  int noBytes = Udp.parsePacket();
  if ( noBytes ) {
    Serial.print(millis() / 1000);
    Serial.print(":Packet of ");
    Serial.print(noBytes);
    Serial.print(" received from ");
    Serial.print(Udp.remoteIP());
    Serial.print(":");
    Serial.println(Udp.remotePort());

    // We've received a packet, read the data from it
    Udp.read(packetBuffer,noBytes); // read the packet into the buffer

/* motor 1 */
      Serial.print("Motor 1 ");
      Serial.print(packetBuffer[2],DEC);
      Serial.print(" ");
      Serial.print(packetBuffer[3],DEC);
      Serial.print(" ");
      a = packetBuffer[2],DEC;
      b = packetBuffer[3],DEC;
      if (a == 255 and b >= 106){      //เลี้ยวซ้าย
        Serial.print(" Left ");
        pw1 = map(b, 255, 106, 0, 255);
        digitalWrite(pin1, LOW);
        digitalWrite(pin2, HIGH);
        }
```

```
    else if(a == 0 and b <= 150 and b >= 1){ //เลี้ยวขวา
      Serial.print(" Right ");
      pw1 = map(b, 1, 150, 0, 255);
      digitalWrite(pin2, LOW);
      digitalWrite(pin1, HIGH);
      }

    else{
      Serial.print(" Stop ");
      pw1 = 0;
      digitalWrite(pin1, HIGH);
      digitalWrite(pin2, HIGH);
      }
   Serial.println(pw1);

/* motor 2 */
    Serial.print("Motor 2 ");
    Serial.print(packetBuffer[6],DEC);
    Serial.print(" ");
    Serial.print(packetBuffer[7],DEC);
    Serial.print(" ");
    c = packetBuffer[6],DEC;
    d = packetBuffer[7],DEC;
    if (c == 255 and d >= 106){      //ถอยหลัง
      Serial.print(" BACK ");
      pw2 = map(d, 255, 106, 0, 255);
      digitalWrite(pin4, LOW);
      digitalWrite(pin3, HIGH);
      }

    else if(c == 0 and d <= 150 and d >= 1){     //เดินหน้า
      Serial.print(" GO ");
      pw2 = map(d, 1, 150, 0, 255);
      digitalWrite(pin3, LOW);
      digitalWrite(pin4, HIGH);
```

```
        }

      else{
        Serial.print(" stop ");
        pw2 = 0;
        digitalWrite(pin3, HIGH);
        digitalWrite(pin4, HIGH);
        }

      Serial.print(pw2);
    Serial.println();
  } // end if

}

void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  // print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
}
```

Code นี้ขอขอบนักศึกษา วิทยาลัยเทคนิคหาดใหญ่ที่ได้เขียนไว้ให้เพื่อนๆ ได้นำไปดัดแปลงในอนาคต จากนั้นทำการทดสอบเดินหน้าถอยหลัง

โปรเจคเครื่องรดน้ำต้นไม้อัตโนมัติ ด้วย Arduino ราคาถูก พร้อม Code ตัวอย่าง

ชุดรดน้ำอัตโนมัติด้วย Arduino เป็นต้นแบบให้น้องได้ศึกษาระบบ Smart Farm พื้นฐาน นำไปพัฒนาต่อในอนาคต โดยใช้ Arduino Uno R3 สามารถนำไปพัฒนาใน งาน IOT หรือจะพัฒนาในแอพพลิเคชั่น Blynk ได้อีกด้วย. (โปรเจคนี้จะอยู่ที่ประมาณ 500 บาท) สามารถขอตัวอย่างเอกสารรูปเล่มโปรเจค จาก Admin เพิ่มเติมได้เลยครับ

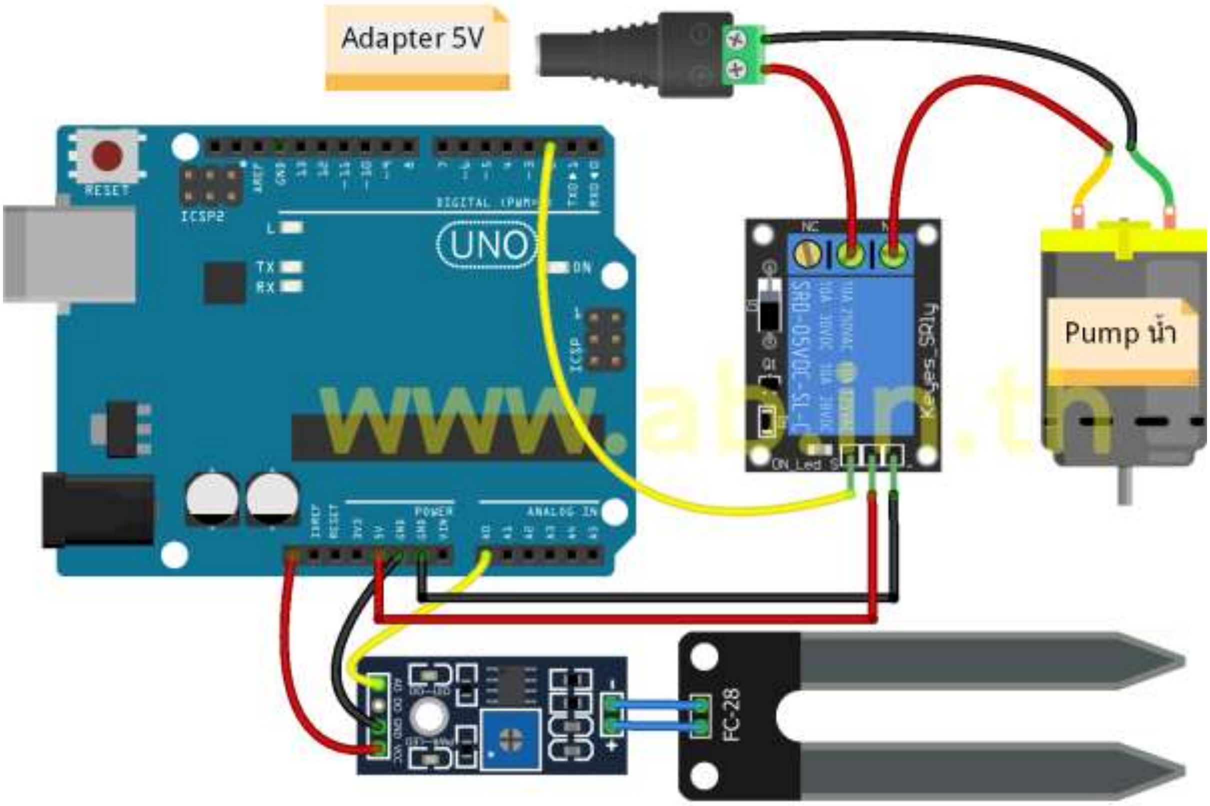
อุปกรณ์ที่ต้องใช้มีดังต่อไปนี้

Arduino Uno R3 พร้อมสาย USB (ในช่วงที่ราคาแพงสามาเปลี่ยนเป็น Wemos D1 ได้น่ะครับ ราคาจะถูกกว่า)

1. Module รีเลย์ relay 1 Chanel 250V/10A Active HIGH II
2. Soil Moisture Sensor Module วัดความชื้นในดิน
3. ปั๊มน้ำ DC ขนาดเล็ก
4. สายยางปั้มน้ำ DC ยาว 1 เมตร
5. Adapter 5V 1A หม้อแปลง 5V 1 แอมป์
6. Power Adapter Jack ตัวเมีย
7. สายแพร Jumper Male to Female ยาว 20CM จำนวน 10 เส้น

สามารถซื้อชุด เซ็ต นำไปประกอบและอัพโหลดโปรแกรมได้เลยได้ที่ คลิก https://www.ab.in.th/contactus

ในส่วนของวงจรการต่อสายมีดังต่อไปนี้



ในส่วนของโปรแกรมตัวอย่าง

https://gist.github.com/AB-Project/c52c6d9ef3f8ccdc62f44b2905c6f692

```
/*
ตัวอย่างโปรเจค รดน้ำต้นไม้อัตโนมัติ การต่ออุปกรณ์
อ่านเพิ่มเติมได้ใน Link https://www.ab.in.th/b/10
การต่อวงจร https://dx.lnwfile.com/_/dx/_raw/nf/s5/ew.png
*/
const int analogInPin = A0;
const int relay = 2;
```

```
int sensorValue = 0;        // ตัวแปรค่า Analog
int outputValue = 0;        // ตัวแปรสำหรับ Map เพื่อคิด %

void setup() {
  Serial.begin(9600);
  pinMode(relay, OUTPUT);
}

void loop() {
  sensorValue = analogRead(analogInPin);
  outputValue = map(sensorValue, 0, 1023, 100, 0);

  Serial.print("Soil Moisture = ");
  Serial.print(outputValue);
  Serial.println(" %");

  if (outputValue <= 40) {  //ตั้งค่า % ที่ต้องการจะรดน้ำต้นไม้
    digitalWrite(relay, HIGH);
  }

  else {
    digitalWrite(relay, LOW);
  }
  delay(1000);
}
```

หลักการทำงานของระบบรดน้ำต้นไม้อัตโนมัติ

โดยปกติแล้ว พืชจะต้องการความชื้นในดินอยู่ที่ประมาณ 40% (แล้วแต่พืชในแต่ละชนิดด้วยน่ะครับ) Sensor วัดความชื้นในดินจะทำการวัดค่าความชื้นในดิน โดยสัญญาณที่ได้ออกมาจะเป็นแบบ Analog ป้อนไปยัง Arduino เพื่อประมวลผล เมื่อ Arduino ได้ทำการวัดตรงตามเงื่อนไข ก็จะทำการส่ง Logic ไปยัง Relay เพื่อทำการเปิดปั้มน้ำไปรดน้ำต้นไม้ แค่นี้ก็จะได้ โปรเจครดน้ำต้นไม้แบบง่ายๆ

อัพเดดเพิ่มเติม

เนื่องด้วยในช่วงที่อุปกรณ์ เซมิคอนดักเตอร์ (Semiconductor) คือ สารกึ่งตัวนำ ทำให้บอร์ด Arduino Uno มีราคาแพงขึ้นทางร้านเลยอยากจะ แนะนำอุปกรณ์ใหม่ที่เหมือน Arduino หน้าตาเหมือน Arduino แต่ราคาถูกกว่า คือ Wemos D1 Uno การต่อใช้งานสามารถต่อได้เหมือนกันกับ Arduino และยังสามารถพัฒนาบอร์ดให้สามารถใช้งานในงาน IOT ได้อีกด้วย สามารถใช้งานร่วมกับ App Blynk เพื่อใช้งาน

ทางด้าน โปรเจค IOT เปิดปิดปั้มน้ำผ่าน App หรือจะดูความชื้นในดินได้อย่าง Real Time และสั่งรดน้ำได้อย่าง Real Time อีกด้วย ส่วนของโปรแกรมสามารถ ขอ Admin ทางร้านได้ฟรีเหมือนเดิม ครับ

NodeMCU Esp8266 เปิดปิดไฟควบคุมผ่าน App มือถือ

สวัสดีครับวันนี้ ทางร้านก็มี โปรเจคเล็กๆ มาแจกน้องๆน่ะครับเกี่ยวกับการเขียน App ควบคุมระบบเปิดปิดไฟ แบบง่ายสุดๆ (ต้มน้ำฉีกซองทานได้เลย) ด้วย Nodemcu Esp8266 และ App บน Android มาเริ่มกันเลย

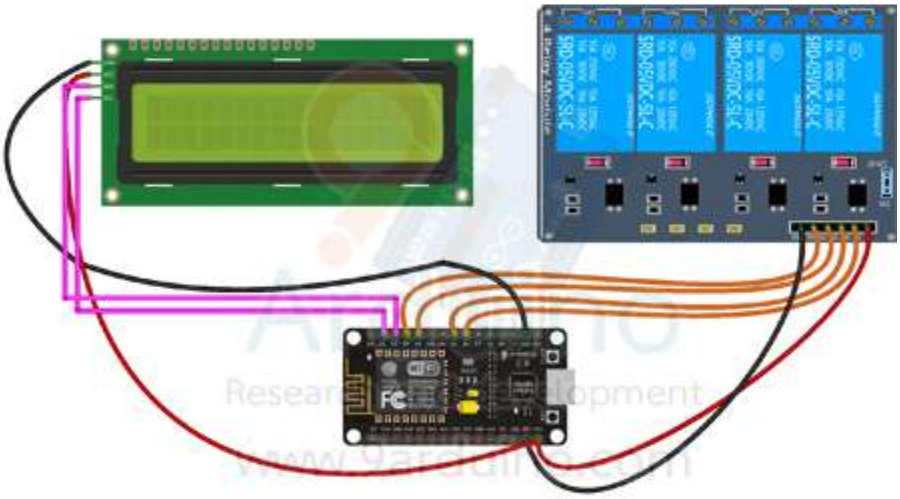
อุปกรณ์ มีดังต่อไปนี้ครับ

1. NodeMCU Esp8266
2. สายไฟ
3. Relay 4ch แบบ Active LOW
4. จอแสดงผลLCD I2C

สำหรับน้องๆที่ยังไม่ติดตั้ง Arduino ให้สามารถใช้งาน NodeMCU คลิกเลยครับ

มาเริ่มต้นกันเลย

1. ต่อวงจรดังนี้



ตำแหน่งขา NodeMCU Esp8266	อุปกรณ์
1. ขา D1	SCL
2. ขา D2	SDA
3. ขา Vin	VCC ของจอ LCD
4. ขา GND	GND ของจอ LCD
5. ขา Vin	VCC ของ Relay
6. ขา GND	GND ของ Relay
7. ขา D0	IN1 ของ Relay
8. ขา D5	IN2 ของ Relay
9. ขา D6	IN3 ของ Relay
10. ขา D7	IN4 ของ Relay

2. เมื่อน้องๆ ประกอบก็เริ่มเขียนโปรแกรม ครับ

Add Library ตัวนี้ด้วยน่ะครับ

Download Library LiquidCrystal_I2 C : http://download.ab.in.th/download.php?file=Arduino-LiquidCrystal-I2C-library-master.zip

Code ตัวอย่าง

```
/* ตัวอย่าง โดย 9Arduino.com พัฒนาเอง
การต่อสาย
ขา D1      SCL
ขา D2      SDA
ขา Vin     VCC ของจอ LCD
ขา GND     GND ของจอ LCD
ขา Vin     VCC ของ Relay
ขา GND     GND ของ Relay
ขา D0      IN1 ของ Relay
ขา D5      IN2 ของ Relay
ขา D6      IN3 ของ Relay
ขา D7      IN4 ของ Relay
จากบทความ Link https://www.ab.in.th/b/11
สำหรับน้องๆที่ต้องการไฟล์ App ไปพัฒนาต่อสามารถ ขอไฟล์เพิ่มเติมได้ที่
https://www.ab.in.th/contactus
*/
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3f, 16, 02);
const char* ssid = "9arduino";      // ชื่อ SSID Wifi
const char* password = "1234567890";       // รหัส Password Wifi
WiFiServer server(80);

  int val1 = 1;
  int val2 = 1;
  int val3 = 1;
  int val4 = 1;
void setup() {
  Serial.begin(9600);
```

```
delay(10);
lcd.begin();
lcd.backlight();
lcd.print("Server Off..");
lcd.setCursor(0, 1);

pinMode(D0, OUTPUT);
pinMode(D5, OUTPUT);
pinMode(D6, OUTPUT);
pinMode(D7, OUTPUT);

digitalWrite(D0, 1);
digitalWrite(D5, 1);
digitalWrite(D6, 1);
digitalWrite(D7, 1);

// Connect to WiFi network
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");

// Start the server
server.begin();
Serial.println("Server started");
lcd.clear();
```

```
  lcd.print("Server On IP :");
  lcd.setCursor(0, 1);

  // Print the IP address
  Serial.println(WiFi.localIP());
  lcd.print(WiFi.localIP());      // แสดง IP ผ่านจอ
  lcd.setCursor(0, 2);
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String req = client.readStringUntil('\r');
  Serial.println(req);
  client.flush();

  // Match the request
  // ch1
  if (req.indexOf("/?ch1=0") != -1)
    val1 = 1;
  else if (req.indexOf("/?ch1=1") != -1)
    val1 = 0;
  // ch2
  else if (req.indexOf("/?ch2=0") != -1)
    val2 = 1;
```

```
  else if (req.indexOf("/?ch2=1") != -1)
    val2 = 0;
  // ch3
  else if (req.indexOf("/?ch3=0") != -1)
    val3 = 1;
  else if (req.indexOf("/?ch3=1") != -1)
    val3 = 0;
  // ch4
  else if (req.indexOf("/?ch4=0") != -1)
    val4 = 1;
  else if (req.indexOf("/?ch4=1") != -1)
    val4 = 0;
  else {
    Serial.println("invalid request");
    client.stop();
    return;
  }

  // Set GPIO2 according to the request
  digitalWrite(D0, val1);
  digitalWrite(D5, val2);
  digitalWrite(D6, val3);
  digitalWrite(D7, val4);

  client.flush();
  // Prepare the response
  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n\r\n\r\nGPIO is now ";
  s += "\n";

  client.print(s);
  delay(1);
  Serial.println("Client disonnected");
}
```

Mini Project ตลับเมตรไร้สาย แสดงบนจอ LCD พัฒนาด้วย Arduino IDE

ในบทความนี้เราจะมาสอนเกี่ยวกับ การใช้งาน Sensor อัลตราโซนิก HC-SR4 ในการวัดระยะทางกัน โดยใช้ Nodemcu V.3 พัฒนาด้วยโปรแกรม Arduino IDE หลักการทำงานของ Sensor อัลตราโซนิก .

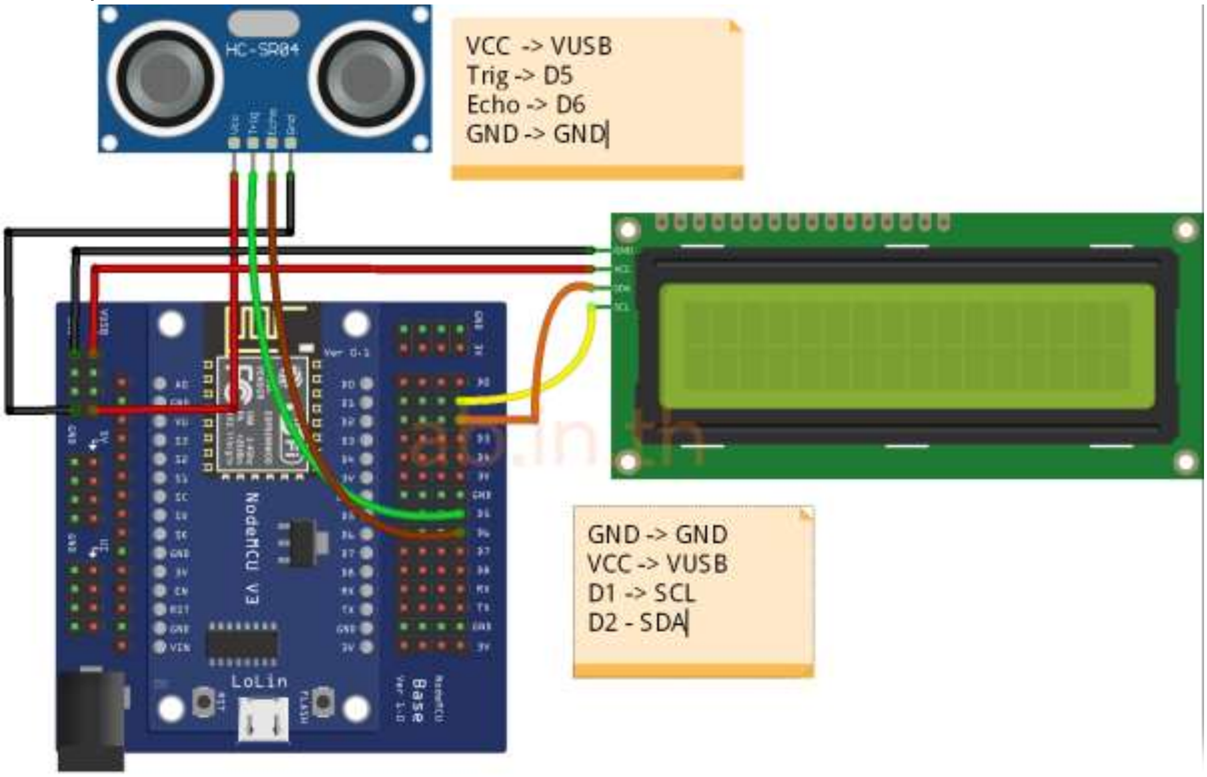
อัลตราโซนิก ก็คือคลื่นเสียงชนิดหนึ่งย่านความถี่ หนึ่ง ปกติ หูของมนุษย์จะได้ยินความถี่ ระหว่าง 20 เฮิรตซ์ -20 กิโลเฮิรตซ์ หากความถี่ต่ำ หรือสูงกว่านี้ เราก็จะไม่ได้ยิน ช่วงประมาณ 18กิโลเฮิรตซ์ขึ้นไป เราจะเรียกความถี่ช่วงนั้นว่า คลื่นอันตร้าโซนิค เราสามารถนำคลื่นนี้ไปใช้งานได้หลากหลายเช่นการไล่แมว หมา หนู หรือใช้แรงสั่นในการล้างอุปกรณ์ ออกนอกเรื่องมากันเยอะแล้ว เข้าเรื่องเลยแล้วกันครับ

หลักการทำงานของ Sensor อัลตราโซนิก นำมาใช้วัดระยะทางโดยตัว Sensor จะส่งเสียงออกแล้ว แล้วสะท้อนวัตถุกลับมาจากนั้นจะมาคำนวนระยะเวลาที่เสียงเดินทางกลับมา กลายเป็นระยะทางที่วัดได้

อุปกรณ์มีดังต่อไปนี้ ราคาไม่เกิน 500 บาท

1. Nodemcu V.3
2. Shield V1.0 For Nodemcu V.3
3. Sensor HC-SR-04
4. จอ LCD I2C 1602
5. สายไฟแบบ Female to Female 20CM จำนวน 10 เส้น
6. Adapter 9V สำหรับนำไปใช้งานจริงไม่ต้องต่อกับ Computer

การต่ออุปกรณ์สามารถต่อได้ดังนี้



มาถึงส่วนของ Code กันเลย

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);   //Module IIC/I2C Interface บางรุ่นอาจจะใช้ 0x3f
#define trigPin D5
#define echoPin D6
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  lcd.begin();
  lcd.backlight();
  lcd.home();
  lcd.print("Distance ");
}
void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);  // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  lcd.setCursor(0, 1);
  if (distance >= 200 || distance <= 0){
    Serial.println("Out of range");
    lcd.print("Out of range");
  } else {
    Serial.print(distance);
    Serial.println(" cm");
    lcd.print(distance);
    lcd.print(" CM        ");
  }
  delay(500);
}
```

แล้วเรายังสามารถพัฒนาโปรเจคนี้จนสามารถดูค่าต่างๆผ่านทาง โทรศัพท์ได้อีกด้วย

Mini Project Arduino Nodemcu แจ้งเตือน น้ำท่วม น้ำล้น ผ่าน Line Notify

โปรเจคนี้พัฒนาโปรแกรมด้วย Arduino IDE เป็นโปรเจคที่แนะนำเกี่ยวกับการใช้ Sensor Water Level และการแจ้งเตือนผ่าน Line เหมาะสำหรับน้องๆที่ศึกษาเกี่ยวกับ Nodemcu การแจ้งเตือนผ่าน Line เป็นเบสิก พื้นฐานสำหรับน้องๆเลยครับ.

ก่อนอื่นน้องๆจะต้องตั้งค่า ID Line ของน้องๆ ก่อนเลยครับ ลองอ่าน Link นี้ได้เลย

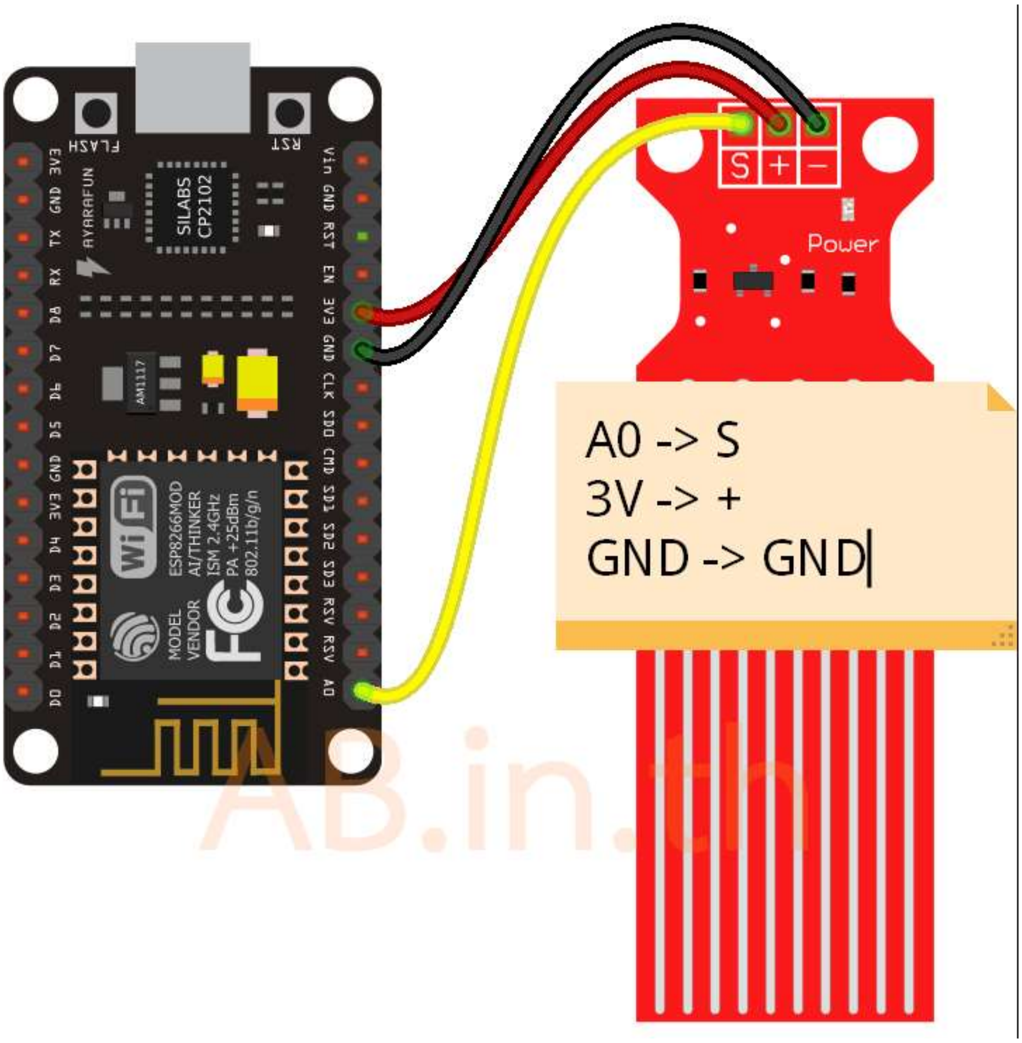
https://www.ab.in.th/b/6

อุปกรณ์มีดังต่อไปนี้ ราคาไม่เกิน 400 บาท

1. Nodemcu V2
2. Sensor Water Level
3. สายไฟแบบ Female to Female 20 cm 10 เส้น
4. Adapter 5V 2A USB

หรือสั่งซื้อได้ที่ คลิก https://www.ab.in.th/contactus

การต่อสายวงจร



ตัวอย่าง Code

```
/*
ติดตั้ง Library Line Notify ด้วยน่ะครับ
https://github.com/TridentTD/TridentTD_LineNotify
บทความจากเว็บ https://www.ab.in.th/b/7
*/
#include <TridentTD_LineNotify.h>
#define SSID        "9arduino"    //SSID Wifi
#define PASSWORD    "tv357911itv"   //Pass Wifi
#define LINE_TOKEN  "--------------------"    //Token ที่ได้จากการสมัคร ID Line
int waterlevel = 300;   //ปรับระดับน้ำ

void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println(LINE.getVersion());
  WiFi.begin(SSID, PASSWORD);
  Serial.printf("WiFi connecting to %s\n",  SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(400);
  }
  Serial.printf("\nWiFi connected\nIP : ");
  Serial.println(WiFi.localIP());

  // กำหนด Line Token
  LINE.setToken(LINE_TOKEN);
}

void loop()
{
  int val = analogRead(A1); // read input value
  Serial.print("Water Level :  ");
  Serial.println(val);
```

```
  delay(2000);

  if (val >= waterlevel) {
    Serial.println("Send Line");
    LINE.notify("น้ำล้นแล้วจ่า !!");
    while (val >= 200) delay(10);
  } else {

  }
}
```

หลักการทำงาน

Sensor Water Level เป็น Sensor แบบ Analog จะป้อนสัญญาณไปยัง Nodemcu เพื่อให้ Nodemcu ตัดสินใจ โดยเราจะปรับค่าระดับน้ำ ที่ ตัวแปร

```
int waterlevel = 300;   //ปรับระดับน้ำ
```

ปรับได้ตามที่เราต้องการเลยครับให้เหมาะสมกับหน้างานน้องๆ เมื่อเข้าเงื่อนไขการทำงาน Nodemcu ก็จะทำการส่งข้อมูลไปยัง Server Line Notify จากนั้น ก็จะมีข้อความแจ้งเตือนมายัง ID Line ของน้องๆ ครับ

Mini Porject Arduino PH Sensor (วัดคุณภาพน้ำความเป็นกรดเป็นด่าง)

จากที่ไม่ได้เขียนบทความมานานวันนี้ทางร้านจะมาแนะนำการใช้งาน Arduino กับ Sensor PH Meter เพื่อวัดคุณภาพน้ำความเป็นกรดเป็นด่างของน้ำ หรือสารละลายต่างๆครับสามารถนำมาใช้งานกับโปรเจคหลากหลายเช่น การวัดสารละลายในน้ำความเป็นกรดเป็นด่างของสารละลาย งานปลูกพื้นไฮโดรโปนิกส์ทางการเกษตร การบำบัดน้ำเสีย เป็นต้น

ค่า PH คืออะไร

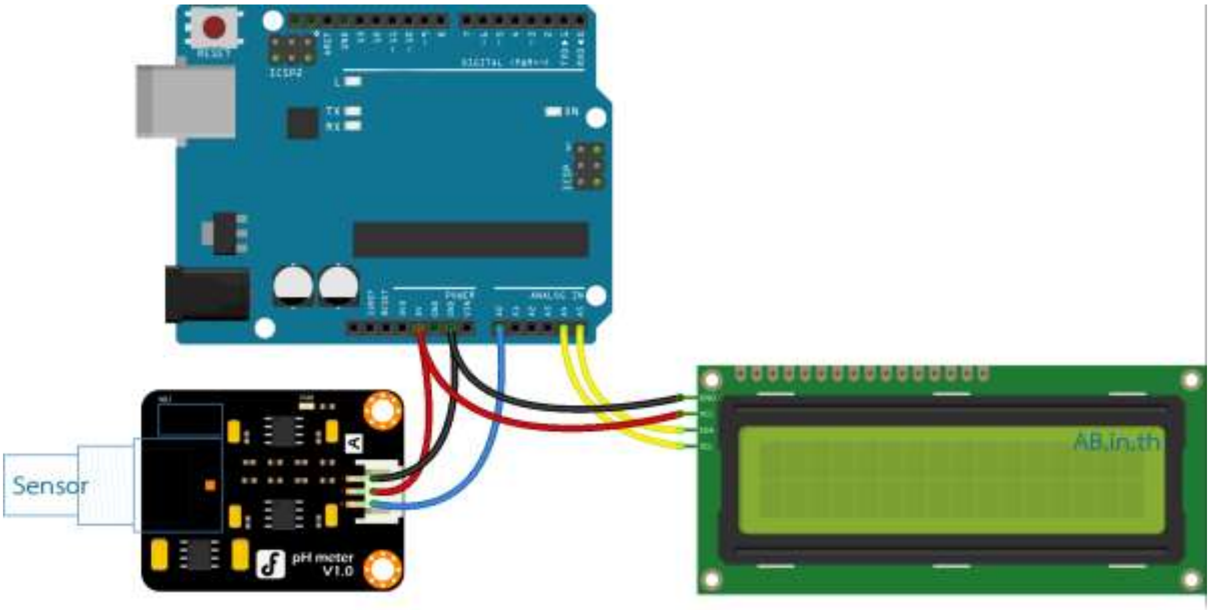
PH คือหน่วยำหรับการวัดความเป็นกรดเป็นด่าง ความเข้มของไฮโดรเจนไอออน โดยค่าที่ได้จะอยู่ในช่วง 0 ถึง 14 โดนน้ำบริสุทธิ์จะมีค่าเท่ากับ 7 หากต่ำกว่า 7 จะเป็นกรด และมากกว่า 7 จะมีค่าเป็นด่าง

น้องๆก็รู้จักเกี่ยวกับ PH และการนำไปใช้งานกันแล้ว ทางร้านก็จะมาแนะนำเกี่ยวกับโปรเจค Ph Meter สำหรับวัดความเป็นกรดเป็นด่าง แนะนำไปใช้งานพัฒนา Project ต่อยอดได้เลยครับ

อุปกรณ์ Mini Project Arduino PH Meter Sensor มีดังต่อไปนี้

1. Arduino Uno R3
2. จอแสดงผล LCD 1602 I2C
3. Ph Sensor
4. สายไฟ Male to Female

การต่อวงจรมีดังต่อไปนี้



Code โปรแกรมตัวอย่าง

```
/*
Project Arduino PH Meter
https://www.ab.in.th/b/85
*/
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
float calibration_value = 21.34;
int phval = 0;
unsigned long int avgval;
int buffer_arr[10], temp;
void setup()
{
  Serial.begin(9600);
  lcd.init();
  lcd.begin(16, 2);
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("PH Meter Project");
  delay(2000);
  lcd.clear();
}
```

```
void loop() {
  for (int i = 0; i < 10; i++)
  {
    buffer_arr[i] = analogRead(A0);
    delay(30);
  }
  for (int i = 0; i < 9; i++)
  {
    for (int j = i + 1; j < 10; j++)
    {
      if (buffer_arr[i] > buffer_arr[j])
      {
        temp = buffer_arr[i];
        buffer_arr[i] = buffer_arr[j];
        buffer_arr[j] = temp;
      }
    }
  }
  avgval = 0;
  for (int i = 2; i < 8; i++)
    avgval += buffer_arr[i];
  float volt = (float)avgval * 5.0 / 1024 / 6;
  float ph_act = -5.70 * volt + calibration_value;
  lcd.setCursor(0, 0);
  lcd.print("pH Val:");
  lcd.setCursor(8, 0);
  lcd.print(ph_act);
  delay(1000);
}
```

Mini Project Arduino Nodemcu แจ้งเตือนฝนตกผ่าน Line

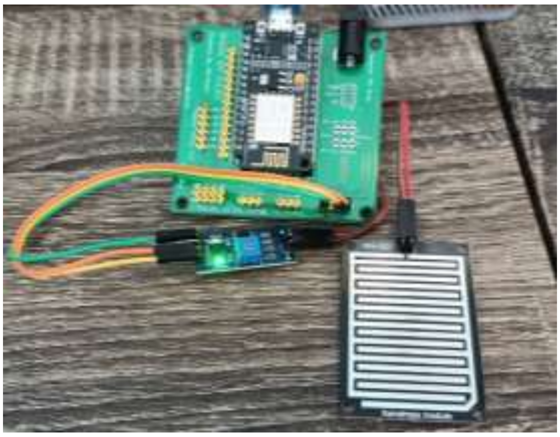
การทำและใช้งาน Arduino Nodemcu 8266 V2 เซนเซอร์วัดปริมาณน้ำฝน

สำหรับบทความนี้จะมีเนื้อหาเกี่ยวกับการทำและใช้งานของบอร์ด Arduino Nodemcu 8266 V2 เซนเซอร์วัดปริมาณน้ำฝน เนื่องจากมีบางคนยังไม่มีความรู้เกี่ยวกับตัวบอร์ด Arduino Nodemcu 8266 V2 จึงขอยกตัวอย่างการใช้งานของบอร์ด ขึ้นมา

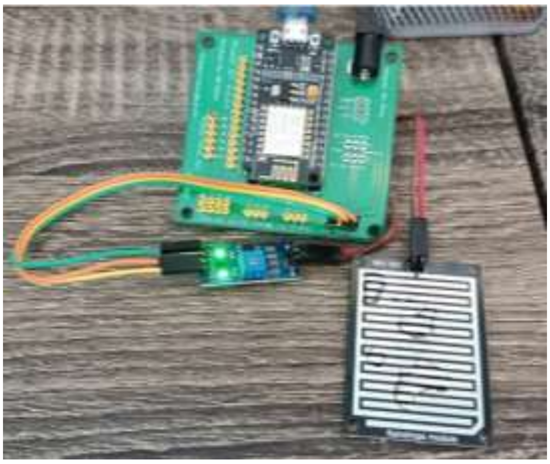
อุปกรณ์ที่ใช้งานมีดังนี้

1. Nodemcu V.2 Program Arduino IDE
2. Rain Sensor Modules เซนเซอร์น้ำฝน Arduino
3. สายแพร Jumper Female to Female ยาว 20CM จำนวน 10 เส้น
4. Shield Nodemcu AB-Maker V1.3 For Nodemcu Esp8266 V2

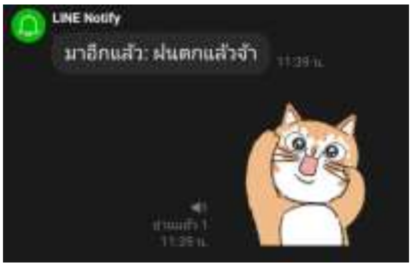
การต่อวงจรบนตัวบอร์ด ดูได้ดังภาพนี้



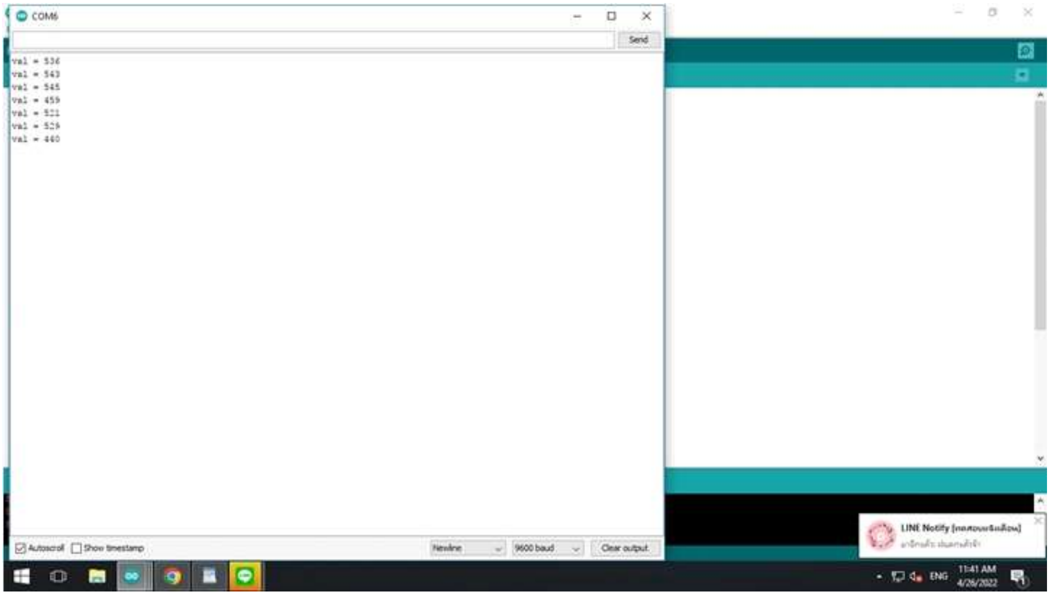
และอันนี้เป็นการต่อเสร็จแล้วและนำมาทดลองใช้ โดยใส่น้ำลงไป



ผลลัพธ์ที่ได้คือ



และมันจะไปแจ้งเตือนที่ไลน์ เพราะเราได้เขียนโค้ดลงไปให้มันไปแจ้งเตือนในตัวไลน์ ดังนี้



โปรแกรมสำหรับพัฒนาโปรเจคนี้

```
/*
    Mini Project Arduino Nodemcu แจ้งเตือนฝนตกผ่าน Line
    https://www.ab.in.th/b/100
    อุปกรณ์ประกอบด้วย
      - บอร์ด Arduino Nodemcu 8266 V2
      - เซนเซอร์น้ำฝน
      - สายจัมพ์ เมีย-เมีย
      - shield (สินค้าทางร้าน)
*/
int analogPin = A0;
int val = 0;
int x = 0;
#include <TridentTD_LineNotify.h>
#define SSID        "arduino"      // ให้ใส่ ชื่อ Wifi ที่จะเชื่อมต่อ
#define PASSWORD    "PassWifi"     // ใส่ รหัส Wifi
#define LINE_TOKEN  "8DYAF27UyN3P53KaGz3h6lDILU4aBMAgYZ8wlAkMPLa"   // บรรทัดที่ 13 
ใส่ รหัส TOKEN ที่ได้มาจากข้างบน
String message = "ฝนตกแล้วจ้า";     // เปลี่ยนเป็นข้อความที่เราต้องการ
void setup() {
  Serial.begin(9600);
  Serial.println(LINE.getVersion());
  WiFi.begin(SSID, PASSWORD);
```

```
  Serial.printf("WiFi connecting to %s\n",  SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(400);
  }
  Serial.printf("\nWiFi connected\nIP : ");
  Serial.println(WiFi.localIP());

  // กำหนด Line Token
  LINE.setToken(LINE_TOKEN);

}

void loop() {
  val = analogRead(analogPin);    //อ่านค่าสัญญาณ analog
  Serial.print("val = ");          // พิมพ์ข้อมความส่งเข้าคอมพิวเตอร์ "val = "
  Serial.println(val);             // พิมพ์ค่าของตัวแปร val
  if (val < 500 and x == 0) {      // สามารถกำหนดปรับค่าได้ตามสถานที่ต่างๆ ตัวอย่างนี้ถ้าน้อยกว่า 500
แล้วส่งข้อความทางไลน์
    LINE.notify(message);          //ข้อความจะส่งไปทางไลน์
    delay(1000);
  }
  if (val < 500) {                 // สามารถกำหนดปรับค่าได้ตามสถานที่ต่างๆ
    x = 1;
  }
  else {
    x = 0;
  }
  delay(1000);
}
```

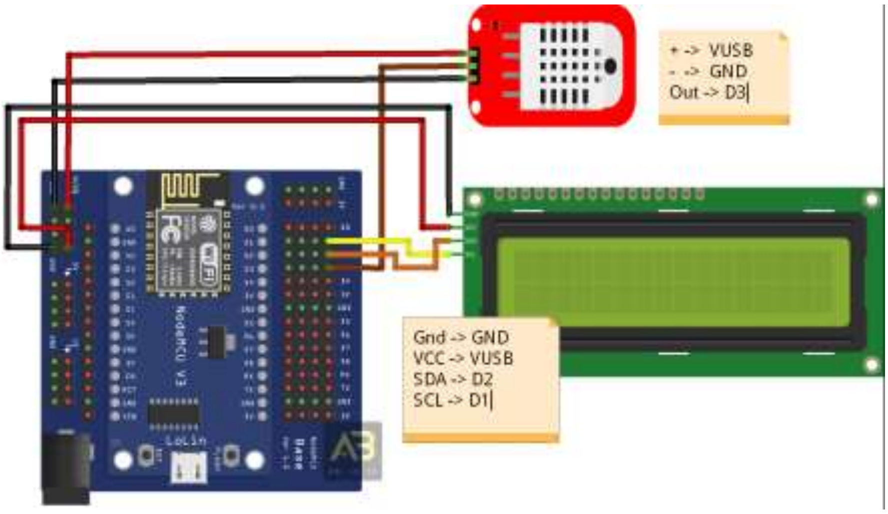
สอนวัดอุณหภูมิ DHT22 แสดงผ่านหน้าจอ LCD ด้วย Nodemcu และ Arduino

สวัสดีครับ มีพี่ จากจังหวัด อุบลราชธานี เข้ามาปรึกษา เกี่ยวกับการวัดอุณหภูมิ ด้วย Sensor DHT22 ต้องการแสดงบนหน้าจอแบบ Real Time และต้องการเก็บข้อมูลลง Database ทางร้านก็เลยเขียนบทความเกี่ยวกับการวัดอุณหภูมิ ความชื้น มาแสดงบนหน้าจอ LCD I2C แล้วกันน่ะครับ

ในโปรเจคนี้จะพูดถึงสอนการใช้งาน Nodemcu DHT22 แสดงค่าบน จอ LCD น่ะครับ โปรเจคนี้สามารถใช้ร่วมกับ Arduino ได้ทุกรุ่น ด้วยน่ะครับ

อุปกรณ์ชุดที่ 1 ใช้เป็น Nodemcu esp8266 V3

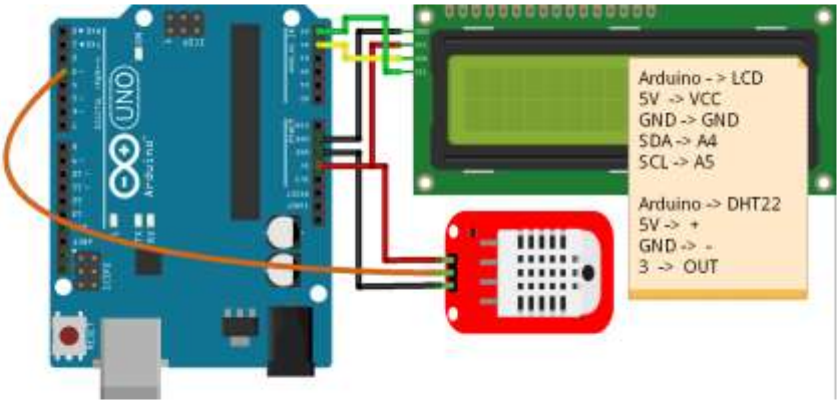
1. Nodemcu (ทางร้านเลือก Nodemcu V3 + Shield 1.0 เพื่อให้เสียบสายไฟได้ง่าย)
2. Sensor DHT22
3. จอ แสดงข้อมูล LCD I2C 1602
4. สายไฟ แบบ Female to Female



ารต่อใช้งาน โดยใช้ Nodemcu V.3 + Shield 1.0

อุปกรณ์ชุดที่ 2 ใช้เป็น Arduino Uno R3

1. Arduino Uno R3 พร้อมสาย USB
2. Sensor DHT22
3. จอ แสดงข้อมูล LCD I2C 1602
4. สายไฟ แบบ Male to Female



มาดู Code ตัวอย่างกันเลย

```
/* Download Library LCD I2 C : http://download.ab.in.th/download.php?file=Arduino-LiquidCrystal-I2C-library-master.zip
Download Library DHT22 : http://download.ab.in.th/download.php?file=DHT-sensor-library-master.zip */
#include "DHT.h"
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);   //Module IIC/I2C Interface บางรุ่นอาจจะใช้ 0x3f
#define DHTPIN D3     //Pin ขาสำหรับ เสียบกับ Nodemcu
//#define DHTTYPE DHT11   // DHT 11
#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21   // DHT 21 (AM2301)
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  dht.begin();
  lcd.begin();
  lcd.backlight();
}

void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);
```

```
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");
  Serial.print("Heat index: ");
  Serial.print(hic);
  Serial.print(" *C ");
  Serial.print(hif);
  Serial.println(" *F");
  lcd.home();
  lcd.print("Temp : ");
  lcd.print(t);
  lcd.setCursor(0, 1);
  lcd.print("Humidity : ");
  lcd.print(t);
  lcd.print("%");

}
```

มาสร้างอุปกรณ์ ตัดสัญญาณ Wifi (Wi-Fi Jamming) ด้วย Nodemcu esp8266

คำเตือน !! ห้ามนำไปเกรียนน่ะครับ เราสอนเพื่อเป็นการเรียนรู้ เดี่ยวลุงตู่ ห้ามนำเข้าสินค้าครับ การใช้อุปกรณ์พวกนี้ ไปเกรียน ผิด พรบ. คอมพิวเตอร์น่ะครับ โดนจับปรับหลายบาท

มาพูดถึงหลักการของ Jammer Technology ที่แท้จริงกันก่อน

Jammer คือ ทำหน้าที่ตัดสัญญาณคลื่นวิทยุ โทรศัพท์มือถือ วิทยุสื่อสาร ไม่ให้สามารถส่งสัญญาณอกไปอีกจุดได้หรือรับสัญญาณจากอีกจุดได้ แปลว่าไปรบกวนสัญญาณ นั้นเอง

อุปกรณ์ Jammer จะทำงานโดยการรบกวนสัญญาณ ตัดสัญญาณ ไม่ว่าจะเป็น Bluetooth Wifi เพื่อไม่ให้สามารถใช้งานได้ โดยการส่งสัญญาณรบกวนให้มความหนาแน่นมาก ทำให้ช่องสัญญาณเต็ม ไม่สามารถใช้งานต่างๆได้ กติการของการใช้ wifi คือ ต้องรอให้สัญญาณว่างก่อนถึงจะสามารถใช้งานหรือรับส่งข้อมูลได้ หลักการของบทความนี้ก็คือ รบกวนให้ช่องสัญญาณเตฺม ทำให้สัญญาณถูกตัด และไม่สามารถใช้งาน Wifi ได้

สำหรับในบทความนี้เราจะพูดถึงการตัดสัญญาณ Wifi โดยใช้ อุปกรณ์ Nodemcu esp8266 ที่เราคุ้นเคยรู้จักกันนำมาตัดสัญญาณ Wifi ข้อดีของมันคือราคาถูกพกพาไปไหนได้อย่างสะดวก คำเตือนอย่าเอาไปเกรียนน่ะครับ

อุปกรณ์ ที่ต้องมี

NodeMCU V2 LUA based ESP8266-12E NodeMCU V2

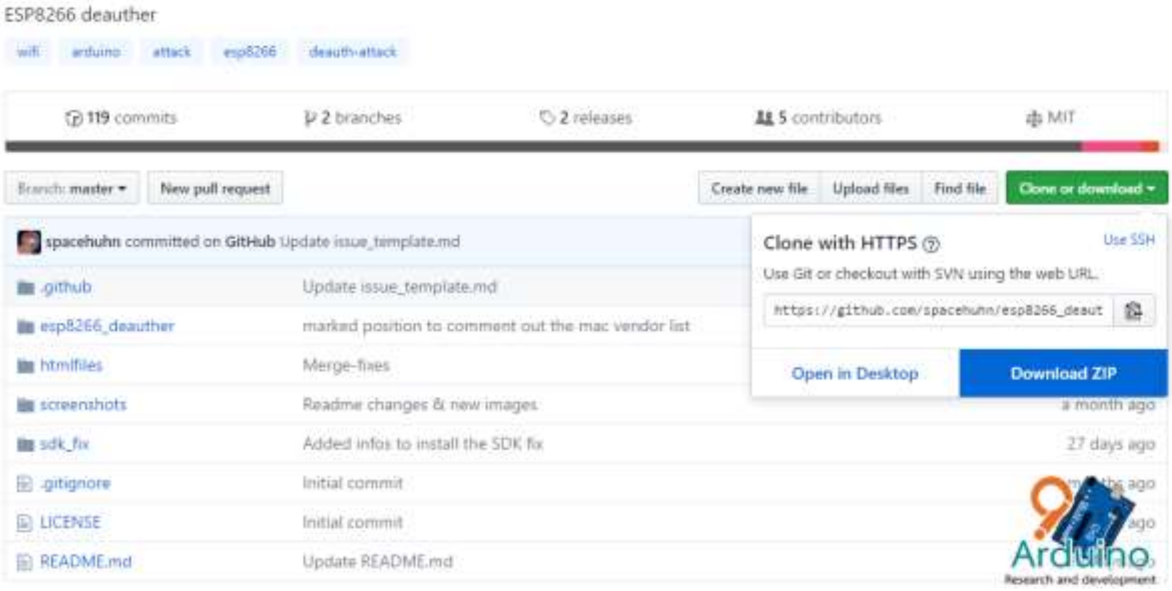
มาเริ่มต้นกันเลยครับ

1. เริ่มจาก ownload ไฟล์ครับ

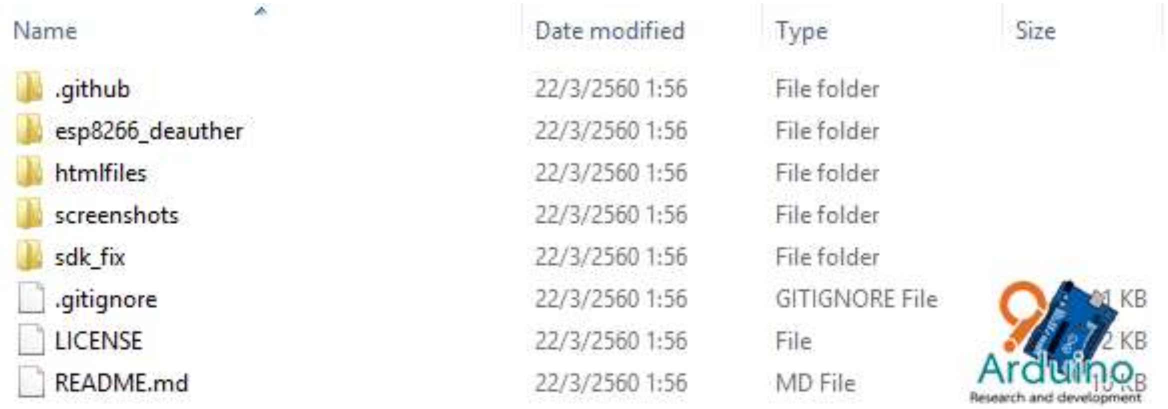
Link 1 : http://download.ab.in.th/download.php?file=esp8266_deauther-master.zip

Link 2 : https://github.com/spacehuhn/esp8266_deauther

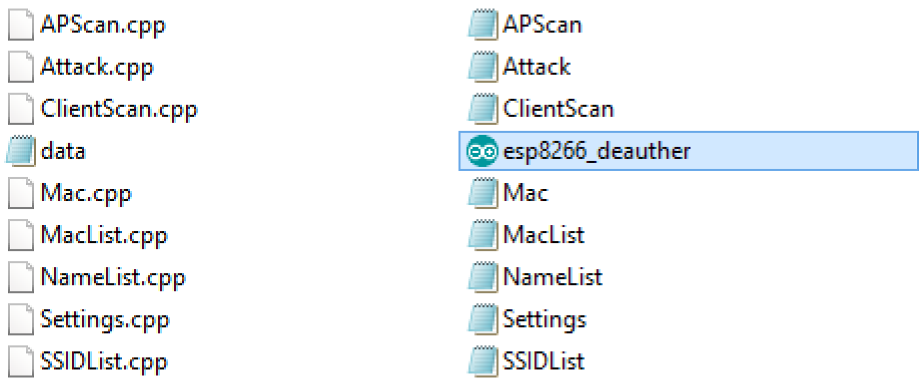
มาไว้ในเครื่องครับ ในส่วนตรงนี้โหลด เป็น Zip มาเลยก็ได้ครับ



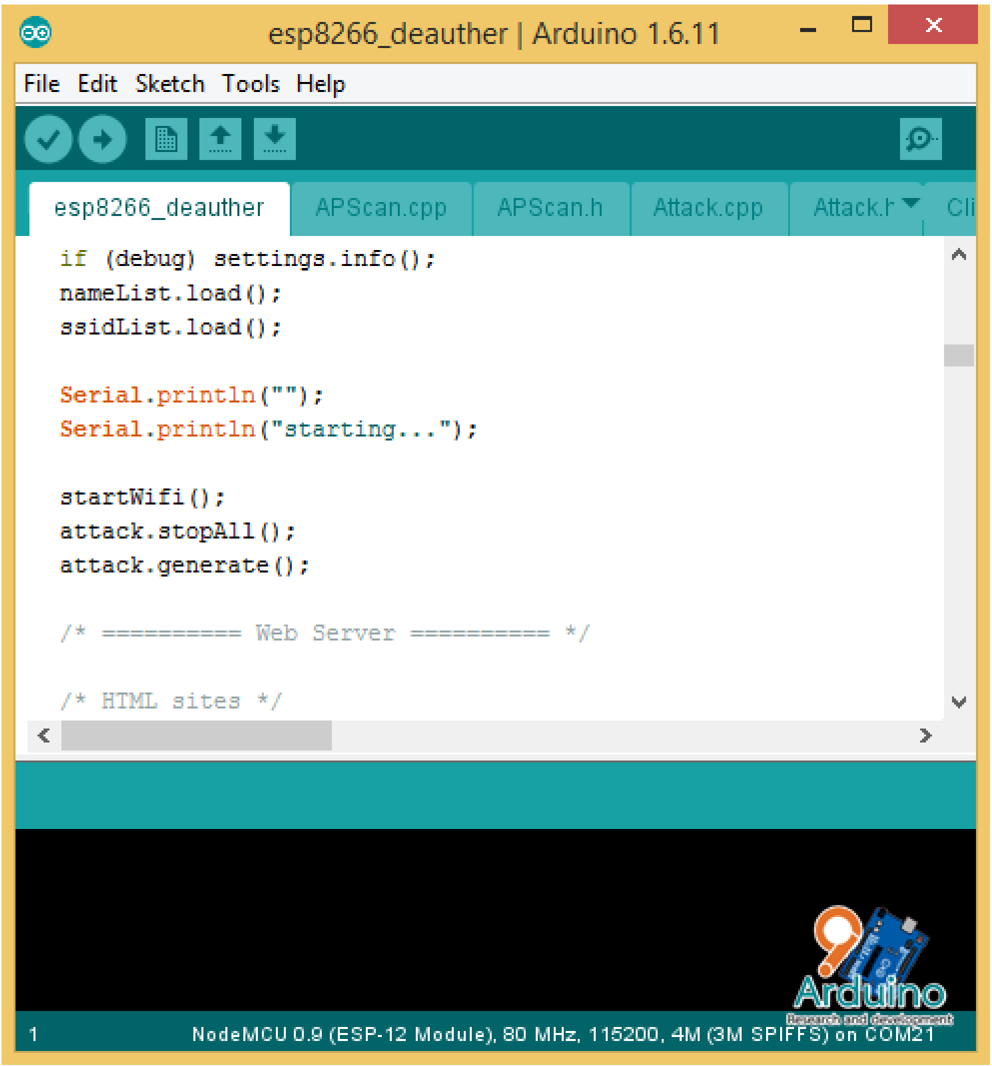
2. เมื่อ Download เสร็จให้แตกไฟล์ ออกมาครับ



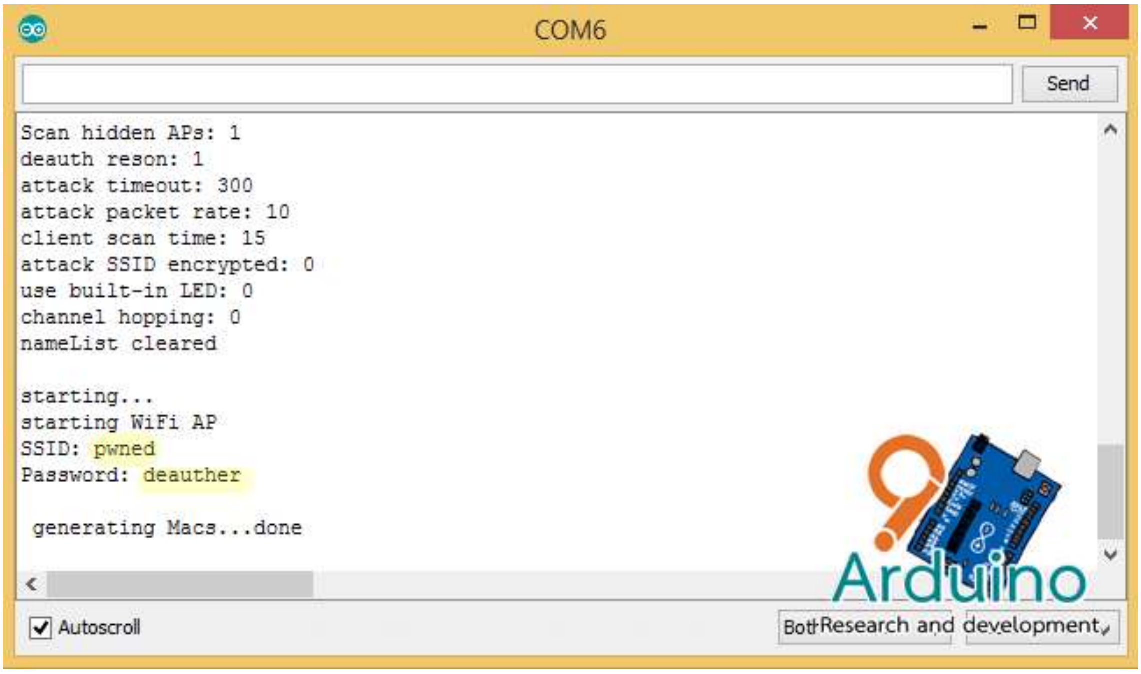
3. จากนั้นให้เข้าไปที่ esp8266_deautcher แล้วเปิดไฟล์ esp8266_deauther.ino ด้วยโปรแกรม Arduino IDE หาก Compile ไม่ผ่านให้ลองเปลี่ยนมาใช้ Arduino IDE เวอร์ชั่น 1.6.9 แทนครับ



4. จากนั้นให้ทำการเลือก Port เลือกบอร์ด Nodemcu ตามที่ท่านใช้งาน แล้วอัพโหลดโปรแกรมใส่ Nodemcu ได้เลยครับ



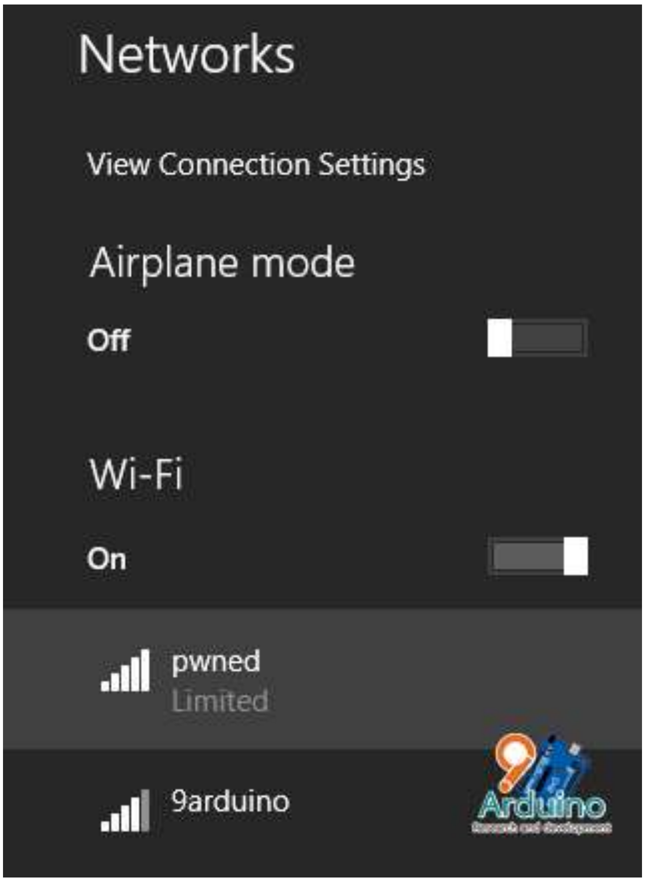
5. เมื่อ Upload เสร็จให้ทำกด กด Serial monitor เพื่อดูชื่อ SSID และ Password ตามภาพ



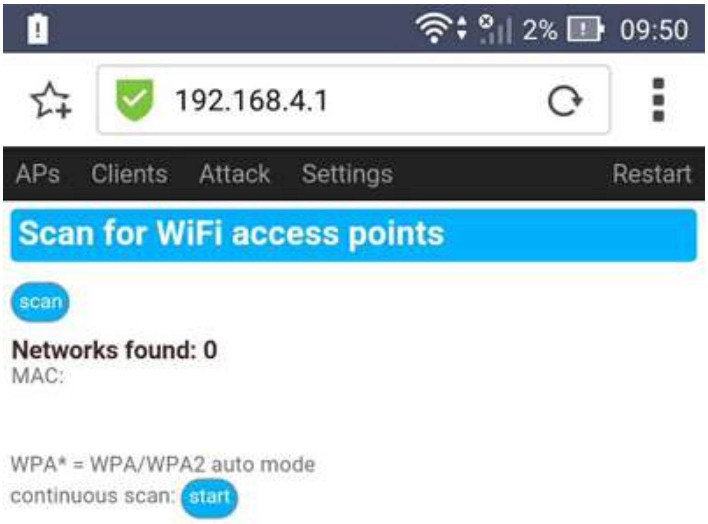
จากภาพ ชื่อ Wifi SSID ของของเราคือ pwned

Password deauther

6. ให้เราทำการเชื่อมต่อ Wifi เพื่อเข้าไปจัดการ โจมตีต่างๆ

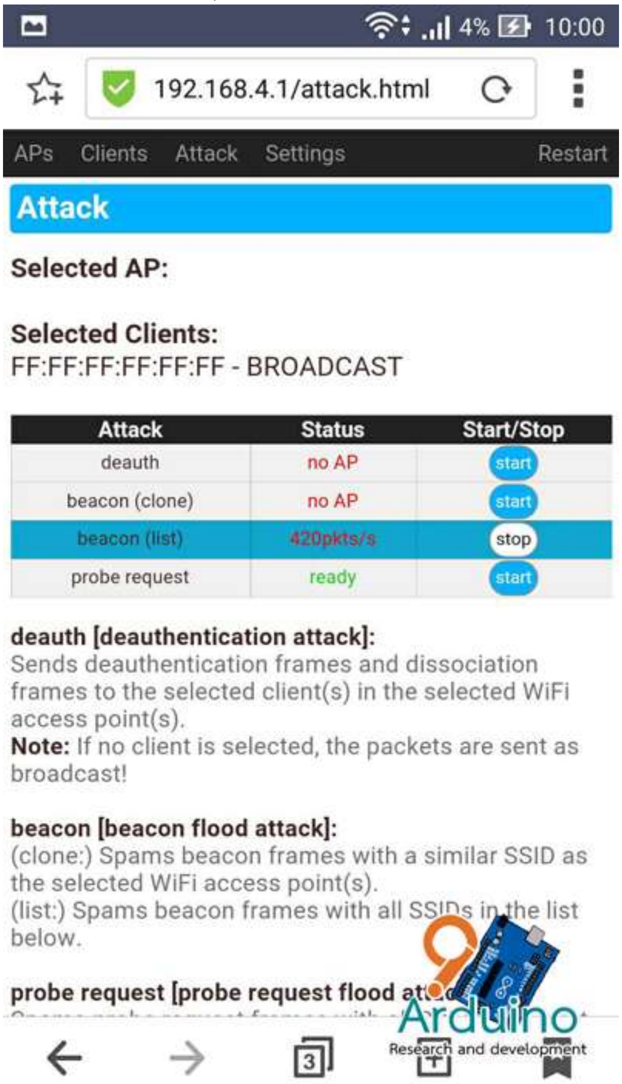


เมื่อเชื่อมต่อ Wifi ของ Nodemcu แล้วให้เราเข้า เข้า เว็บ http://192.168.4.1/ เพื่อจัดการค่าต่างๆ
เมื่อเข้าเว็บดังกล่าวจะขึ้นหน้าตาเว็บ ดังรูปด้านล่าง

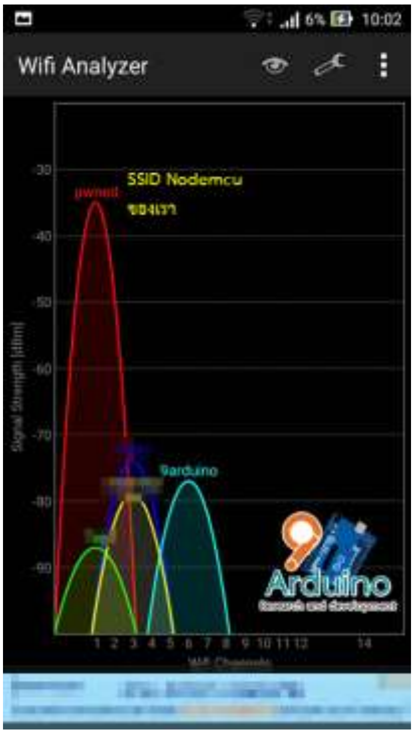


มื่อเรากด ปุ่ม เมนูด้านบนที่ Setting จะมีข้อมูลให้เราแก้ไข เช่น SSID Password ต่างๆ สามารถกดไขแล้วกด Save ได้เลย

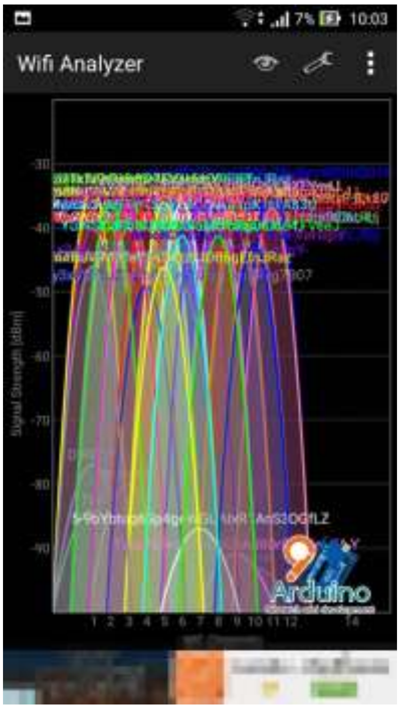
ถัดมาให้กด ที่ปุ่ม Attack เพื่อสร้าง SSID ที่เราต้องการอาจจะ Random หรือ สร้างชื่อที่เราต้องการจะบอกรักแฟน ก็ได้ครับ เมื่อสร้างเสร็จให้กดปุ่ม Start เพื่อเริ่ม โจมตี



เมื่อเรา กด เริ่มโจมตี เจ้า Nodemcu ของเราจะส่งชื่อ SSID ที่เราได้สร้างไว้ สร้างจนเต็มจนไม่สามารถเชื่อมต่อ wifi ได้



เมื่อทำการโจมตี จะกลายเป็น



คำเตือน. เรียนรู้เพื่อนำไปทดสอบทดลอง อย่าไปเกรียนกันน่ะครับ เดี๋ยวโดนจับจะหาว่าไม่เตือนน่ะครับ ขอขอบคุณพี่เล็ก windows 98se สำหรับแหล่งเรียนรู้ต่างๆด้วยน่ะครับ

สำหรับวิธีการแก้ปัญหา

1. หาต้นตอของปัญหาแล้วจัดการมัน แจ้งตำรวจ การกระทำแบบนี้ผิด พรบ.คอมพิวเตอร์ ครับ
2. ใช้ Router คุณภาพสูงๆ ที่รองรับ 5G ก็จะสามารถหนีผ่านบทความนี้ด้วย