

ใบงาน

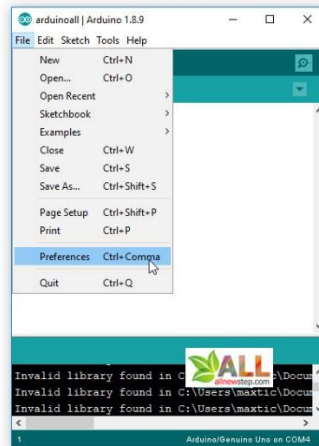
ชื่องาน ทดสอบ GPIO

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เน็ตเฟสในระบบสมองกลฝังตัวและไอโอที

ขั้นตอนการทำงาน

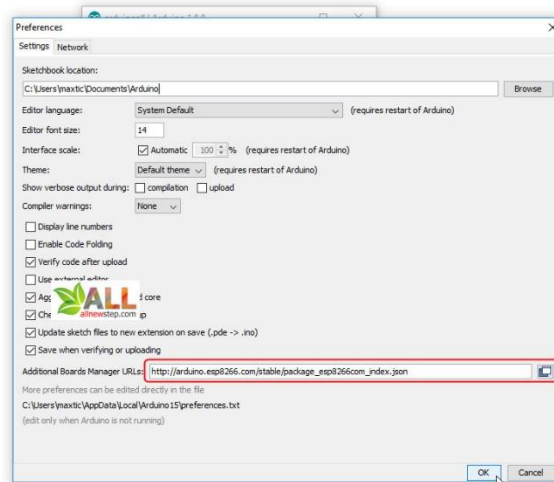
1. เพิ่มบอร์ดลงในโปรแกรม Arduino IDE

1.1. ไปที่เมนู File > Preferences

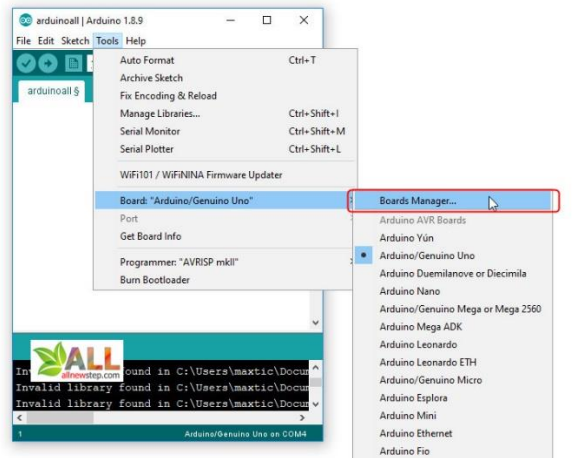


1.2. ที่ช่อง Additional Boards Manager URLs เพิ่มคำว่า

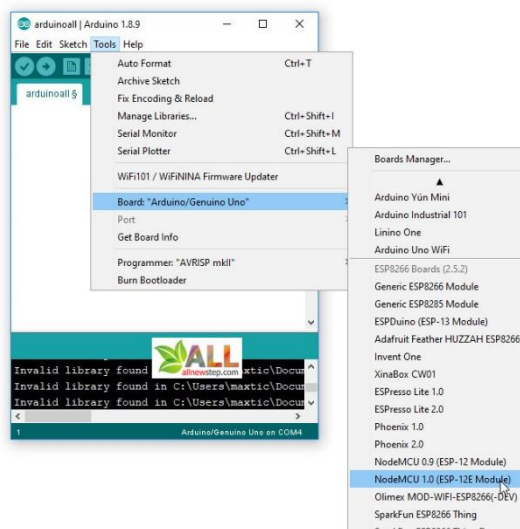
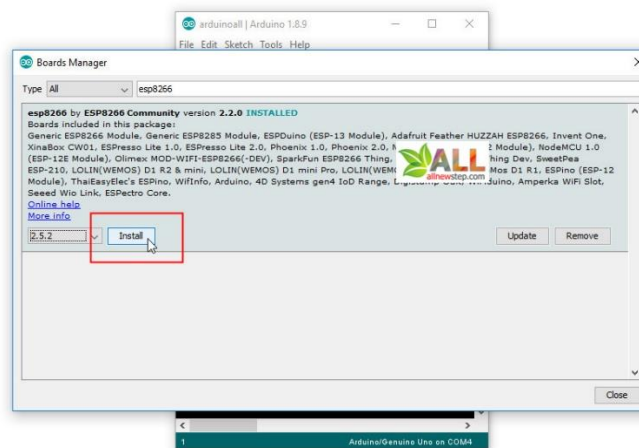
http://arduino.esp8266.com/stable/package_esp8266com_index.json



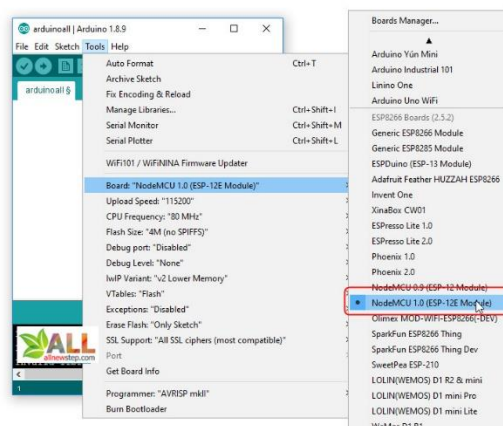
1.3. เพิ่มบอร์ดที่เมนู Tools > Board > Board Manager...



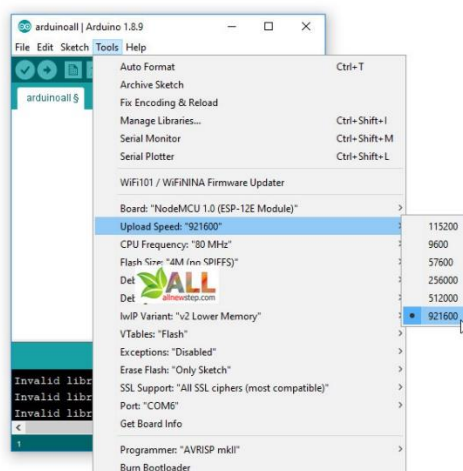
1.4. พิมพ์ค้นหาคำว่า esp8266 แล้วกดปุ่ม Install



- ติดตั้งไดรเวอร์ CP2102 และ CH340 แนะนำให้ลง 2 ตัวพร้อมกันเลยจะได้สะดวก ใช้ได้กับ NodeMCU ทุกรุ่น ดาวน์โหลดไดรเวอร์ที่นี่
<https://www.arduinoall.net/arduino-tutor/code/DriverNodeMCU.rar>
- เสียบบอร์ด NodeMCU เข้ากับเครื่องคอม เลือกบอร์ด NodeMCU 1.0(ESP-12E Module) และ Com port ให้ถูกต้อง



- ที่ Upload Speed คือความเร็วในการส่งข้อมูล ยิ่งค่ามากยิ่งอัปโหลดเสร็จเร็ว เราสามารถเลือกเป็นค่า 921600 ได้เลย แต่บางกรณีอาจมีสัญญาณรบกวน ทำให้อัปโหลดไม่สำเร็จ ก็ให้ลองปรับมาเป็นค่าต่ำลง เช่น 115200 แทน



- อัปโหลดโค้ด Arduino ตัวอย่างนี้ แล้วดูผลลัพธ์

```
void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}
```

```
void loop() {  
  Serial.println("ArduinoAll TESTED");  
  digitalWrite(2, 0);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, HIGH);  
  digitalWrite(2, 1);  
  delay(1000);  
}
```

ใบงาน

ชื่องาน งานควบคุมอุปกรณ์ output เปิดปิดไฟ LED

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เฟสในระบบสมองกลฝังตัวและไอโอที

ใบความรู้

คำสั่ง Digital Write

เป็นคำสั่งที่ใช้กำหนดสัญญาณ HIGH LOW ของขาดิจิตอลของ NodeMCU ESP8266

HIGH คือลอจิก 1 ปล่อยไฟออกแรงดัน 5V

LOW คือลอจิก 0 กำหนดขานั้นให้เป็นกราวด์ 0V

`digitalWrite(PiN,Status)`

PiN หมายถึง ขา Digital ของ NodeMCU ESP8266 ที่จะสั่งงาน ให้เป็น HIGH หรือ LOW

Status หมายถึง สถานะ HIGH หรือ LOW

ตัวอย่างคำสั่ง Digital Write

ต้องการให้ขา Digital ขา D0 เป็นสถานะ HIGH

`digitalWrite(D0,HIGH)`

คำสั่ง `pinMode(led1, status);`

เป็นคำสั่งกำหนดการทำงานของขา

led1 คือ ขาที่ต้องการกำหนดสถานะการทำงาน

status คือ สถานะการทำงาน มี Input อ่านค่าสถานะลอจิกขานั้น และ Output ปล่อยสัญญาณลอจิก 1 0

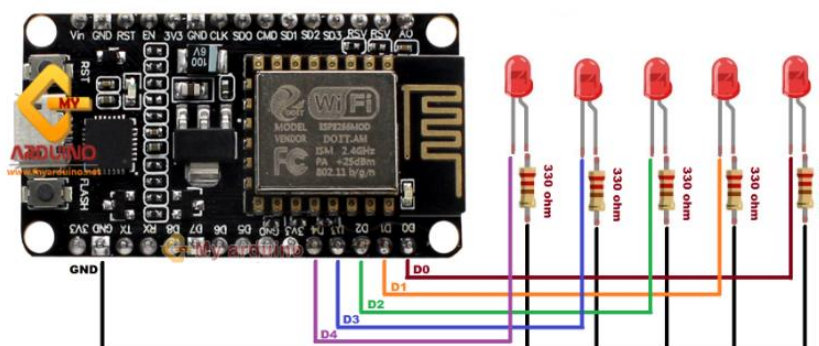
ตัวอย่างคำสั่ง `pinMode(led1, status);`

ต้องการให้ขา Digital ขา D0 เป็น Output ปล่อยสัญญาณดิจิตอล 1 0 HIGH LOW

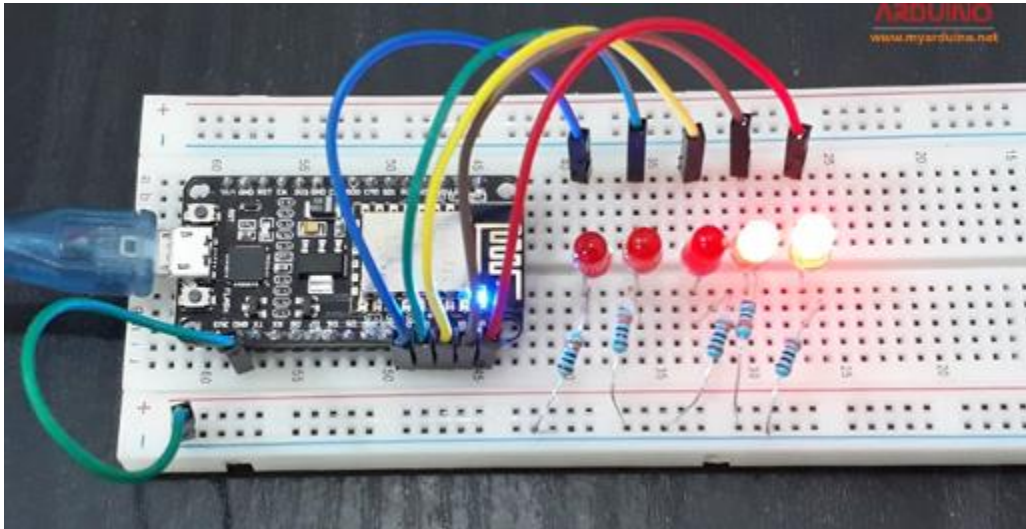
`pinMode(D0, Output);`

ขั้นตอนการทำงาน

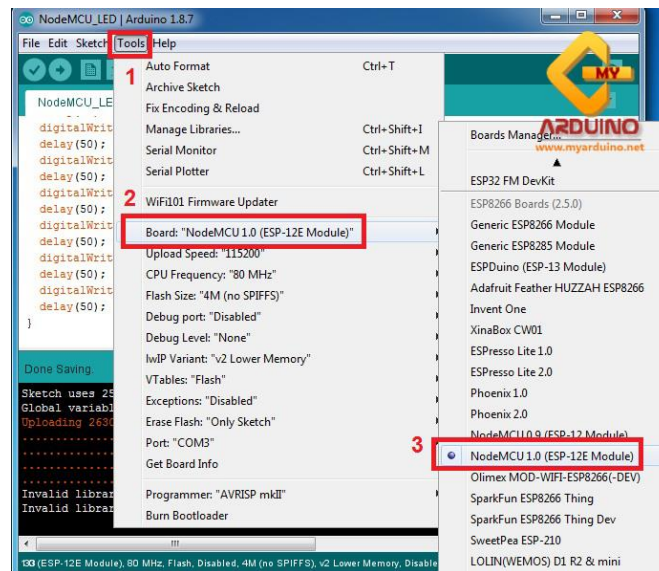
1. ออกแบบการเชื่อมต่อวงจรตามโจทย์กำหนดให้



2. ต่อวงจรระหว่าง NodeMCU ตัวต้านทาน LED ลงบน Breadboard ตามที่ออกแบบไว้



3. เสียบบอร์ด NodeMCU เข้ากับเครื่องคอมพิวเตอร์ และ Comport



4. เขียนโปรแกรม และทำการอัปโหลดโค้ด Arduino แสดงผลลัพธ์ที่ได้

```
int LEDD0 = D0; // ขา D0
int LEDD1 = D1; // ขา D1
int LEDD2 = D2; // ขา D2
int LEDD3 = D3; // ขา D3
int LEDD4 = D4; // ขา D4
void setup() {
  pinMode(LEDD0, OUTPUT); // กำหนดการทำงานของขา D0 ให้เป็น Output
  pinMode(LEDD1, OUTPUT);
  pinMode(LEDD2, OUTPUT);
```

```

pinMode(LEDD3, OUTPUT);
pinMode(LEDD4, OUTPUT);

digitalWrite(LEDD0, LOW);
digitalWrite(LEDD1, LOW);
digitalWrite(LEDD2, LOW);
digitalWrite(LEDD3, LOW);
digitalWrite(LEDD4, LOW);

}
void loop()
{
    digitalWrite(LEDD0, HIGH); // สั่งให้ ขา D0 ปล่อยลอจิก 1 ไฟ LED ติด
    delay(50); // หน่วงเวลา 50mS
    digitalWrite(LEDD1, HIGH); // สั่งให้ ขา D1 ปล่อยลอจิก 1 ไฟ LED ติด
    delay(50);
    digitalWrite(LEDD2, HIGH); // สั่งให้ ขา D2 ปล่อยลอจิก 1 ไฟ LED ติด
    delay(50);
    digitalWrite(LEDD3, HIGH); // สั่งให้ ขา D3 ปล่อยลอจิก 1 ไฟ LED ติด
    delay(50);
    digitalWrite(LEDD4, HIGH); // สั่งให้ ขา D4 ปล่อยลอจิก 1 ไฟ LED ติด
    delay(50);
    digitalWrite(LEDD0, LOW); // สั่งให้ ขา D0 ปล่อยลอจิก 0 ไฟ LED ดับ
    delay(50);
    digitalWrite(LEDD1, LOW); // สั่งให้ ขา D1 ปล่อยลอจิก 0 ไฟ LED ดับ
    delay(50);
    digitalWrite(LEDD2, LOW); // สั่งให้ ขา D2 ปล่อยลอจิก 0 ไฟ LED ดับ
    delay(50);
    digitalWrite(LEDD3, LOW); // สั่งให้ ขา D3 ปล่อยลอจิก 0 ไฟ LED ดับ
    delay(50);
    digitalWrite(LEDD4, LOW); // สั่งให้ ขา D4 ปล่อยลอจิก 0 ไฟ LED ดับ
    delay(50);
}

```

ใบงาน

ชื่องาน งานควบคุมอุปกรณ์ output เปิดปิดไฟบ้าน ด้วย Relay Module

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เฟสในระบบสมองกลฝังตัวและไอโอที

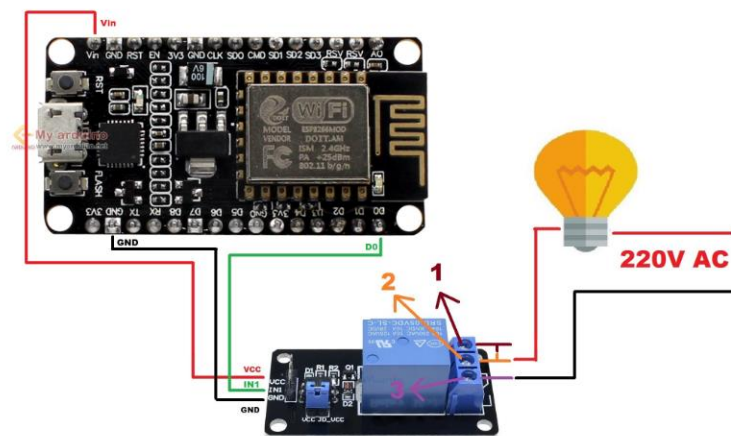
ใบความรู้

Relay Module -> NodeMCU ESP8266

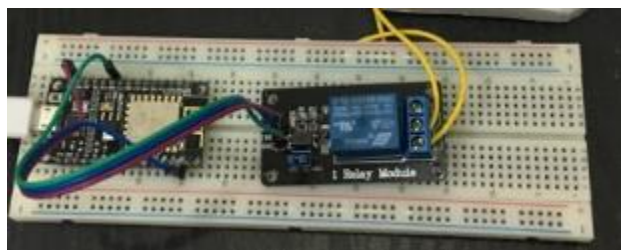
- gnd -> GND
- Vcc -> 5V
- In-> ขาD0

ขั้นตอนการทำงาน

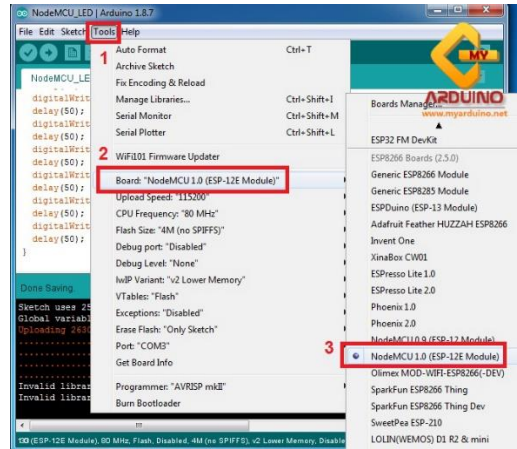
5. ออกแบบการเชื่อมต่อวงจรตามโจทย์กำหนดให้



6. ต่อวงจรระหว่าง NodeMCU ตัวต้านทาน LED ลงบน Breadboard ตามที่ออกแบบไว้



7. เสียบบอร์ด NodeMCU เข้ากับเครื่องคอมพิวเตอร์ เลือกบอร์ด และ Comport



8. เขียนโปรแกรม และทำการอัปโหลดโค้ด Arduino แสดงผลลัพธ์ที่ได้

```
int Relay1 = D0; // กำหนดขาใช้งาน
void setup()
{
  pinMode(Relay1, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา D0 เป็น OUTPUT
  digitalWrite(Relay1, HIGH);
}
void loop()
{
  digitalWrite(Relay1, LOW); // ส่งให้ไฟติด
  delay(1000); // ดีเลย์ 1000ms
  digitalWrite(Relay1, HIGH); // ส่งให้ไฟดับ
  delay(1000); // ดีเลย์ 1000ms
}
```

ใบงาน

ชื่องาน งานควบคุมอุปกรณ์ output แสดงผลออกจอ LCD 1602 แบบ I2C

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เฟสในระบบสมองกลฝังตัวและไอโอที

ใบความรู้

จอแสดงผล LCD 1602 ตัวที่ 1 -> NodeMCU ESP8266

- gnd -> GND
- Vcc -> 5V
- SDA -> ขาD2 (NodeMCU ESP8266)
- SCL -> ขาD1 (NodeMCU ESP8266)

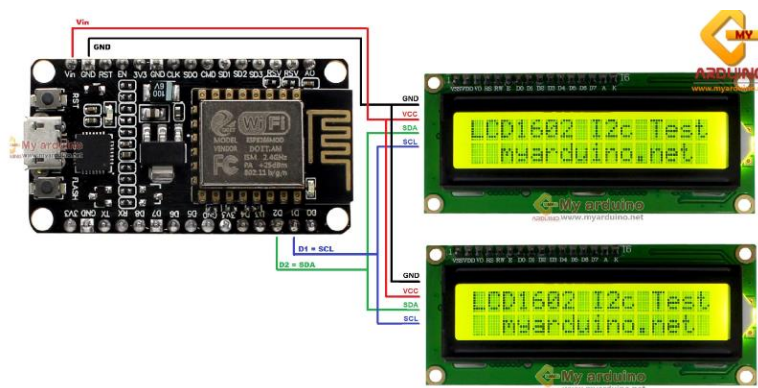
จอแสดงผล LCD 1602 ตัวที่ 2 -> NodeMCU ESP8266

- gnd -> GND
- Vcc -> 5V
- SDA -> ขาD2 (NodeMCU ESP8266)
- SCL -> ขาD1 (NodeMCU ESP8266)

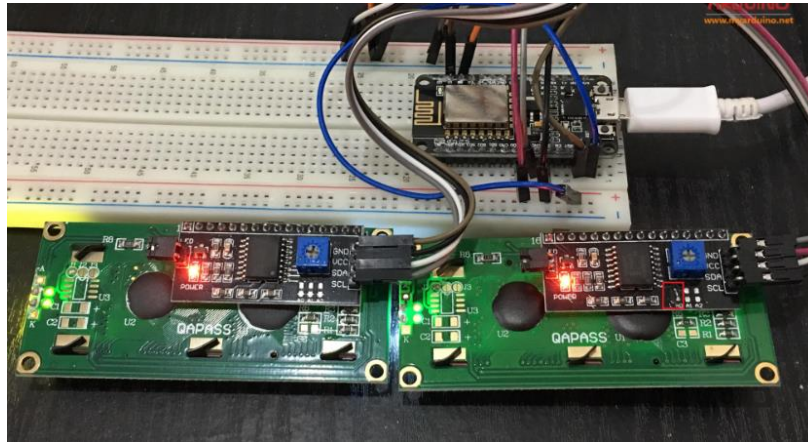
•

ขั้นตอนการทำงาน

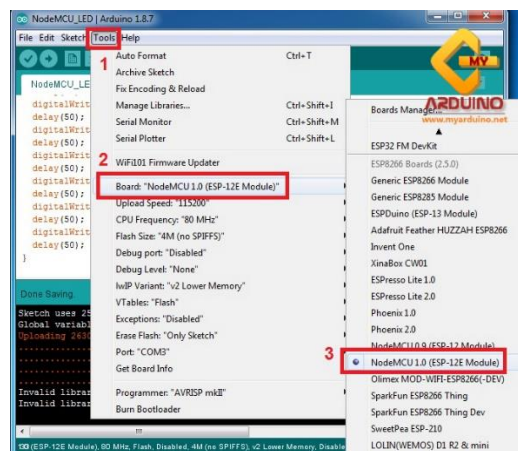
9. ออกแบบการเชื่อมต่อวงจรตามโจทย์กำหนดให้



10.ต่อวงจรระหว่าง NodeMCU ลงบน Breadboard ตามที่ออกแบบไว้



11.เสียบบอร์ด NodeMCU เข้ากับเครื่องคอมพิวเตอร์ เลือกบอร์ด และ Comport



12.เขียนโปรแกรม และทำการอัปโหลดโค้ด Arduino แสดงผลลัพธ์ที่ได้

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
  lcd.begin();
  lcd.backlight();
}
void loop() {
  LCD1(); // สั่งให้ไปทำงานในฟังก์ชัน LCD1
  LCD2(); // สั่งให้ไปทำงานในฟังก์ชัน LCD2
}
```

```

void LCD1() {
    LiquidCrystal_I2C lcd(0x27, 16, 2); //เลือก Address จอ LCD ที่จะเขียนข้อความ
    lcd.begin();
    lcd.setCursor(0, 0); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่0 แถวที่ 1 เตรียม
พิมพ์ข้อความ
    lcd.print("LCD1602 I2c No1"); //พิมพ์ข้อความ "LCD1602 I2c No1"
    lcd.setCursor(2, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่3 แถวที่ 2 เตรียม
พิมพ์ข้อความ
    lcd.print("myarduino.net"); //พิมพ์ข้อความ "myarduino.net"
}

void LCD2() {
    LiquidCrystal_I2C lcd(0x26, 16, 2); //เลือก Address จอ LCD ที่จะเขียนข้อความ
    lcd.begin();
    lcd.setCursor(0, 0); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่0 แถวที่ 1 เตรียม
พิมพ์ข้อความ
    lcd.print("LCD1602 I2c No2"); //พิมพ์ข้อความ "LCD1602 I2c No2"
    lcd.setCursor(2, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่3 แถวที่ 2 เตรียม
พิมพ์ข้อความ
    lcd.print("myarduino.net"); //พิมพ์ข้อความ "myarduino.net"
}

```

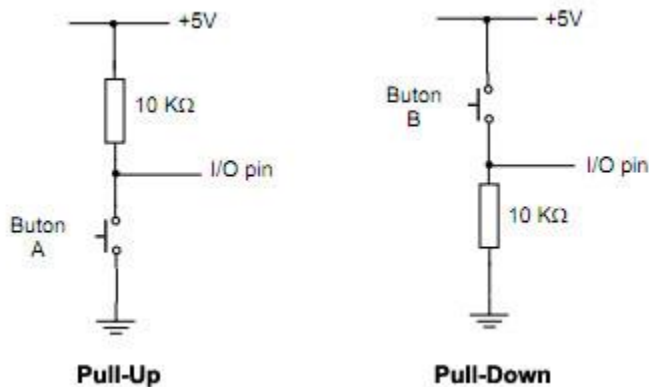
ใบงาน

ชื่องาน งานควบคุมอุปกรณ์ input กดปุ่มสวิตช์ ควบคุมเปิดปิดหลอดไฟ LED

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เฟสในระบบสมองกลฝังตัวและไอโอที

ใบความรู้

การต่อสวิตช์มี 2 แบบ คือ pull up โดยจ่ายสัญญาณ 1 ให้กับสวิตช์ อีกแบบคือ pull down โดยจ่ายสัญญาณ 0 ให้กับสวิตช์ โดยผ่านตัวต้านทานเพื่อใช้คงค่าสถานะ ป้องกันไฟ 5V และ 0V ชนกันลัดวงจร



การต่อสวิตช์มี 2 แบบคือ

1. Pull UP คือการคงค่าสัญญาณ 1 ให้กับขา Digital Arduino ที่มาต่อกับสวิตช์ เมื่อสวิตช์ถูกกดจะให้สัญญาณ 0
2. Pull Down คือ การคงค่าสัญญาณ 0 ให้กับขา Digital Arduino ที่มาต่อกับสวิตช์ เมื่อสวิตช์ถูกกดจะให้สัญญาณ 1

คำสั่ง Digital Read

เป็นคำสั่งที่ใช้อ่านค่าสถานะขาดิจิตอลของ Arduino ว่าเป็น 5V หรือ 0V

HIGH คือลอจิก 1 ปล่อยไฟออกแรงดัน 5V

LOW คือลอจิก 0 กำหนดขานั้นให้เป็นกราวด์ 0V

`digitalRead(PiN)`

PiN หมายถึง ขา Digital ของ Arduino ที่ต้องการอ่าน

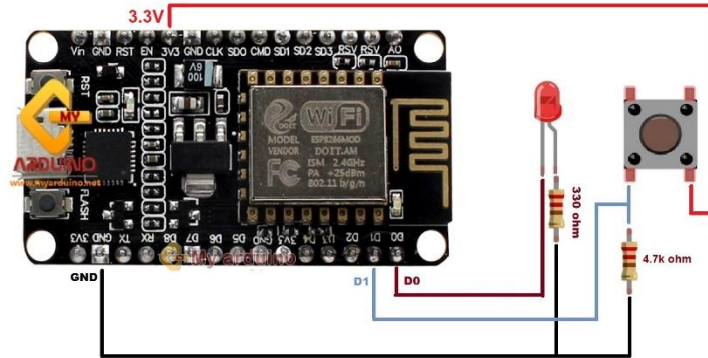
ตัวอย่างคำสั่ง Digital Read

ต้องการอ่านค่า สถานะขา 13 ว่าเป็น 1 หรือ 0 เก็บในตัวแปร Val

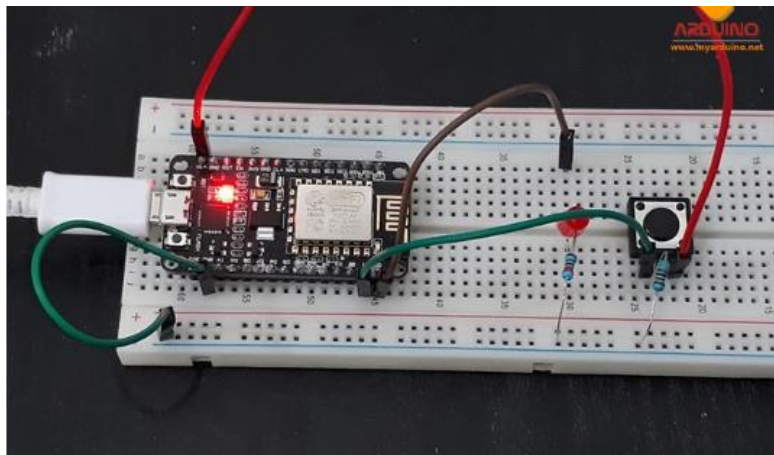
Val = `digitalRead(13)`

ขั้นตอนการทำงาน

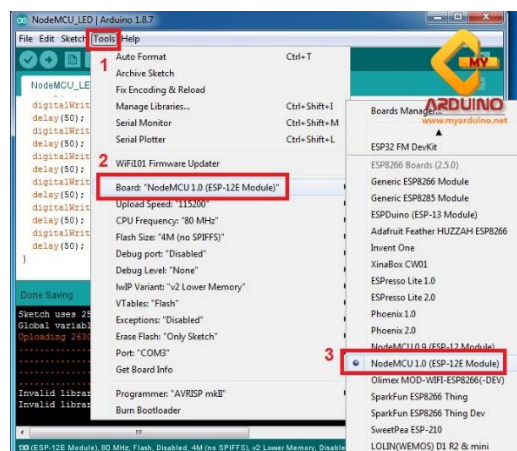
1. ออกแบบการเชื่อมต่อวงจรตามโจทย์กำหนดให้



2. ต่อวงจรระหว่าง NodeMCU ลงบน Breadboard ตามที่ออกแบบไว้



2. เสียบบอร์ด NodeMCU เข้ากับเครื่องคอมพิวเตอร์ เลือกบอร์ด และ Comport



3. เขียนโปรแกรม และทำการอัปโหลดโค้ด Arduino แสดงผลลัพธ์ที่ได้

```
int led1 = D0; // กำหนดขาใช้งาน
int buttonPin = D1;
int buttonState = 0;
void setup()
{
  pinMode(led1, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา D0 เป็น OUTPUT
  pinMode(buttonPin, INPUT); // กำหนดขาทำหน้าที่ให้ขา D1 เป็น INPUT รับค่า
  จากสวิตช์
  digitalWrite(led1, LOW);
  Serial.begin(9600);
}
void loop()
{
  buttonState = digitalRead(buttonPin); // อ่านค่าสถานะขาD0
  if (buttonState == HIGH) { //กำหนดเงื่อนไขถ้าตัวแปร buttonState เก็บ ค่า
  1(HIGH) ให้ทำในปีกกา
    digitalWrite(led1, HIGH); // ไฟ LED 1ติด
    Serial.print("LED : ");
    Serial.println(buttonState);
  }
  else { //ถ้าตัวแปร buttonState ไม่ได้เก็บ ค่า 1(HIGH) คือ ตัวแปร buttonState
  เก็บค่า 0(LOW) อยู่ ให้ทำปีกกาข้างล่าง
    digitalWrite(led1, LOW); // ไฟ LED 1ดับ
    Serial.print("LED : ");
    Serial.println(buttonState);
  }
}
```

ใบงาน

ชื่องาน งานควบคุมอุปกรณ์ input วัดอุณหภูมิและความชื้นด้วย Sensor DHT11

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เฟสในระบบสมองกลฝังตัวและไอโอที

ใบความรู้

NodeMCU ESP8266 -> Sensor DHT11 วัดอุณหภูมิและความชื้น

- 5V -> ขา1
- GND -> ขา4
- ขาD4 = GPIO 2 -> ขา2

โหลด Library Sensor วัดอุณหภูมิและความชื้น DHT11 ติดตั้งในโปรแกรม Arduino IDE

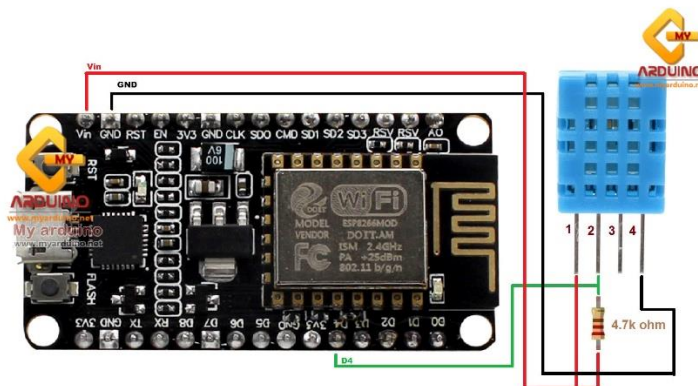
- <http://www.mediafire.com/download/6qh8q1g0kmokl4g/DHT11.rar>

วิธีลง Library

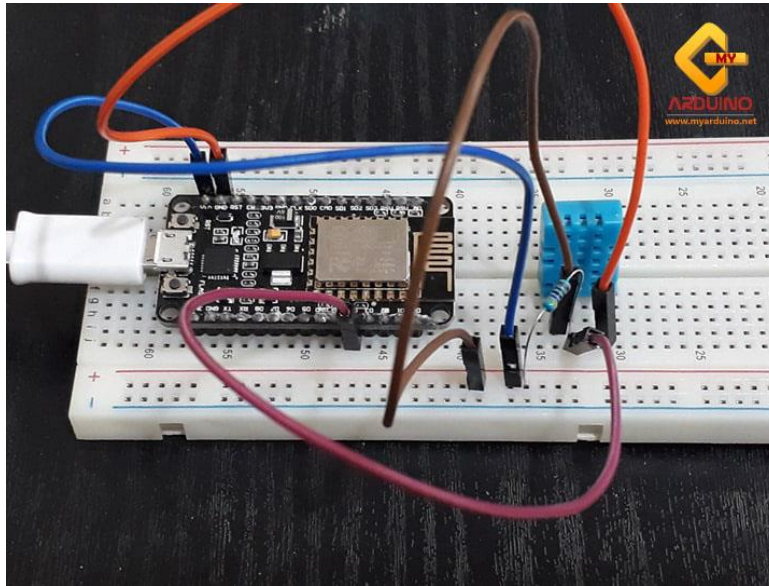
- สอนใช้งาน NodeMCU ESP8266 ติดตั้ง Library ในโปรแกรม Arduino IDE เชื่อมต่อกับ Sensor ต่างๆ

ขั้นตอนการทำงาน

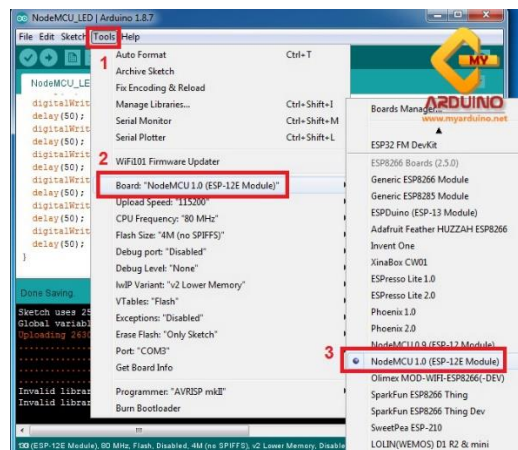
13.ออกแบบการเชื่อมต่อวงจรตามโจทย์กำหนดให้



2. ต่อดวงจรระหว่าง NodeMCU ลงบน Breadboard ตามที่ออกแบบไว้



14.เสียบบอร์ด NodeMCU เข้ากับเครื่องคอมพิวเตอร์ เลือกบอร์ด และ Comport



15.เขียนโปรแกรม และทำการอัปโหลดโค้ด Arduino แสดงผลลัพธ์ที่ได้

```
#include "DHT.h"
DHT dht;
void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");

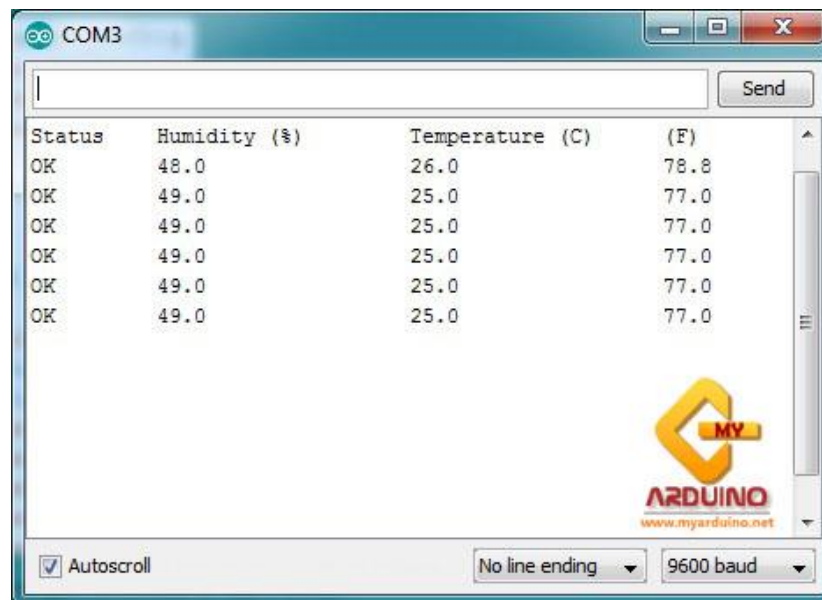
  dht.setup(2); // data pin 2
```

```

}

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);
}

```



ใบงาน

ชื่องาน งานควบคุมอุปกรณ์ input วัดความเข้มแสงด้วยเซ็นเซอร์ LDR

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เฟสในระบบสมองกลฝังตัวและไอโอที

ใบความรู้

เชื่อมต่ออุปกรณ์ตามด้านล่าง

NodeMCU ESP8266 -> หลอดไฟ LED

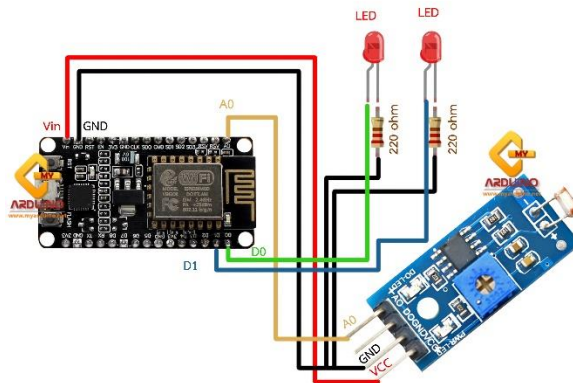
- ขา2 -> LED
- ขา3 -> LED

NodeMCU ESP8266 -> เซนเซอร์วัดความสว่างความเข้มแสง LDR Photoresistor Sensor Module

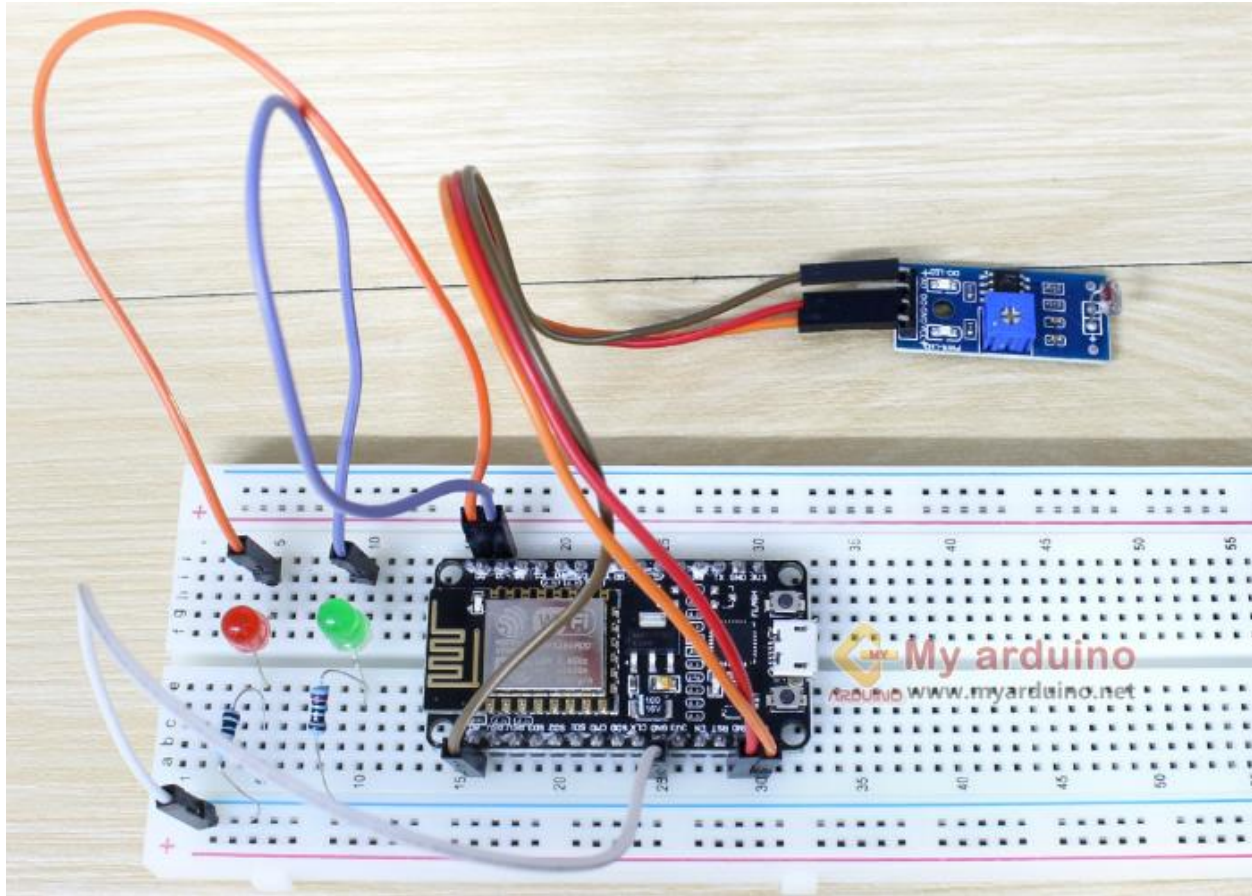
- A5 -> A0
- 5V -> VCC
- GND -> GND

ขั้นตอนการทำงาน

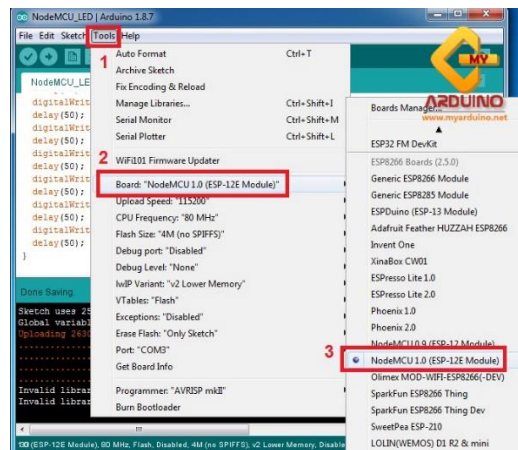
1. ออกแบบการเชื่อมต่อวงจรตามโจทย์กำหนดให้



2. ต่อวงจรระหว่าง NodeMCU ลงบน Breadboard ตามที่ออกแบบไว้



2. เสียบบอร์ด NodeMCU เข้ากับเครื่องคอมพิวเตอร์ และ Comport



3. เขียนโปรแกรม และทำการอัปโหลดโค้ด Arduino แสดงผลลัพธ์ที่ได้

```
int ledPin = D0;
int ledPin3 = D1;
int analogPin = A0; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขาที่5
int val = 0;
```

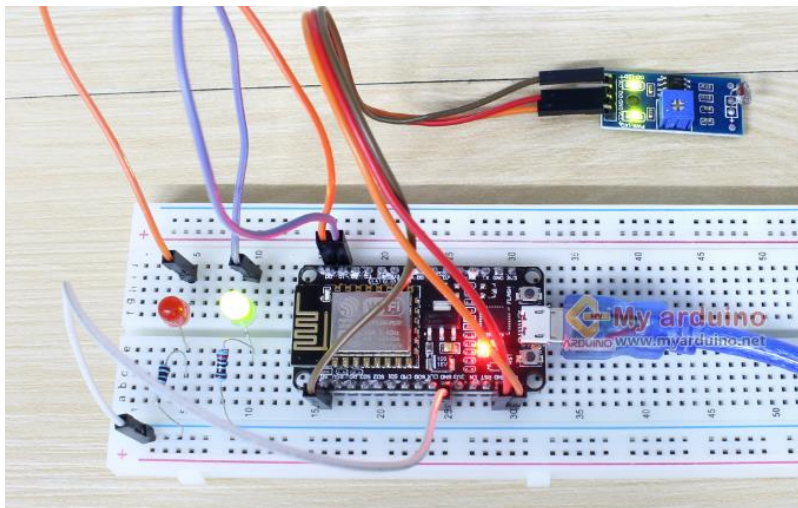
```

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);
  Serial.print("val = ");
  Serial.println(val); // พิมพ์ค่าของตัวแปร val
  if (val < 900) { // ค่า 900 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
    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);
}

```



ใบงาน

ชื่องาน งานควบคุมอุปกรณ์ input เซ็นเซอร์ตรวจจับวัตถุ IR Infrared

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เฟสในระบบสมองกลฝังตัวและไอโอที

ใบความรู้

Sensor IR Infrared ตรวจจับสิ่งกีดขวาง ตรวจจับวัตถุข้างหน้า ตัว Sensor จะใช้หลักการสะท้อนของแสงอินฟราเรด ตัวส่งยิงแสงออกไปตกกระทบวัตถุ และรับค่าการสะท้อนกลับเข้าตัวรับ

NodeMCU ESP8266 -> หลอดไฟ LED

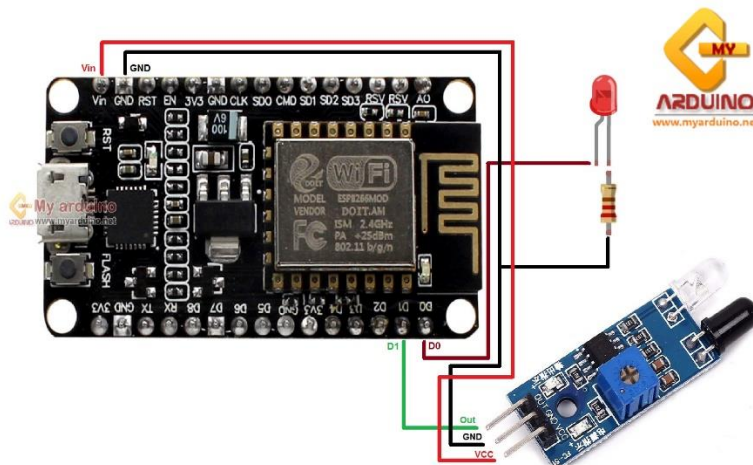
- ขาD0 -> LED1

NodeMCU ESP8266 -> เซ็นเซอร์ตรวจจับวัตถุ IR Infrared

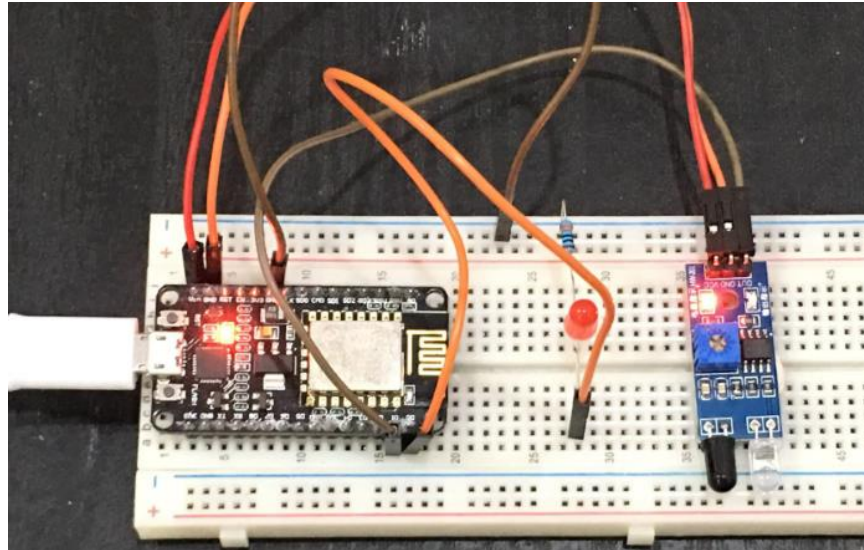
- D1 -> A0
- Vin -> Vcc
- GND -> GND

ขั้นตอนการทำงาน

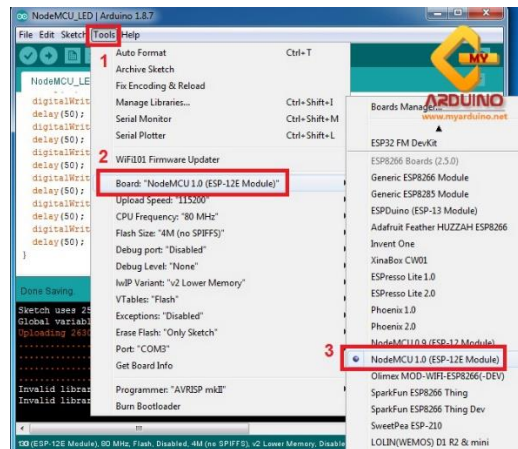
16.ออกแบบการเชื่อมต่อวงจรตามโจทย์กำหนดให้



2. ต่อวงจรระหว่าง NodeMCU ลงบน Breadboard ตามที่ออกแบบไว้



17.เสียบบอร์ด NodeMCU เข้ากับเครื่องคอมพิวเตอร์ เลือกบอร์ด และ Comport



18.เขียนโปรแกรม และทำการอัปโหลดโค้ด Arduino แสดงผลลัพธ์ที่ได้

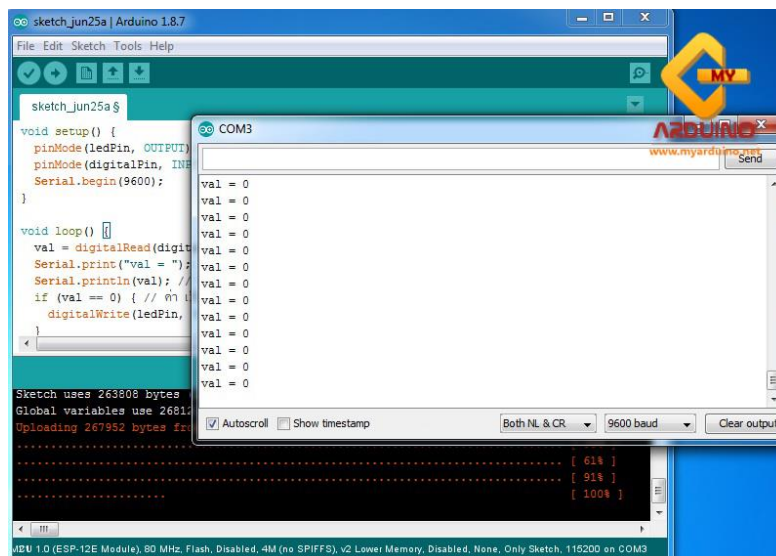
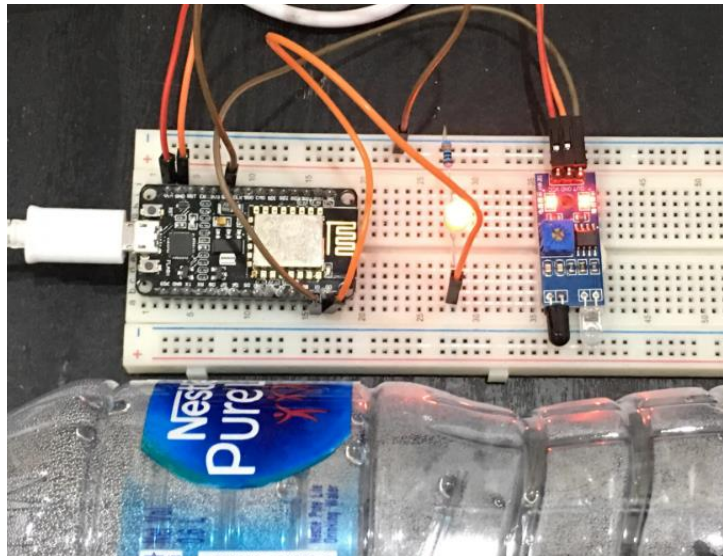
```
int ledPin = D0;
int digitalPin = D1;
int val = 0;
void setup() {
  pinMode(ledPin, OUTPUT); // sets the pin as output
  pinMode(digitalPin, INPUT); // sets the pin as input
  Serial.begin(9600);
}

void loop() {
```

```

    val = digitalRead(digitalPin); //อ่านค่าสัญญาณ digital ขาD1 ที่ต่อกับ เซ็นเซอร์
    ตรวจสอบวัตถุ IR Infrared
    Serial.print("val = "); // พิมพ์ข้อความส่งเข้าคอมพิวเตอร์ "val = "
    Serial.println(val); // พิมพ์ค่าของตัวแปร val
    if (val == 0) { // ค่า เป็น 0 ตรวจสอบเจอวัตถุ สั่งให้ไฟ LED ติด
        digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
    }
    else {
        digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
    }
    delay(100);
}

```



ใบงาน

ชื่องาน งานวัดอุณหภูมิและความชื้นด้วย Sensor DHT11 แสดงผลออกจอ LCD 1602

วิชา 20901-9203 ชื่อวิชา การอินเทอร์เฟสในระบบสมองกลฝังตัวและไอโอที

ใบความรู้

การต่อวงจร

LCD ----> ESP8266

GND ----> GND

VCC ----> Vin

SDA ----> D2

SCL ----> D1

DHT11 ----> ESP8266

+ ----> Vin

OUT ----> D4

- ----> GND

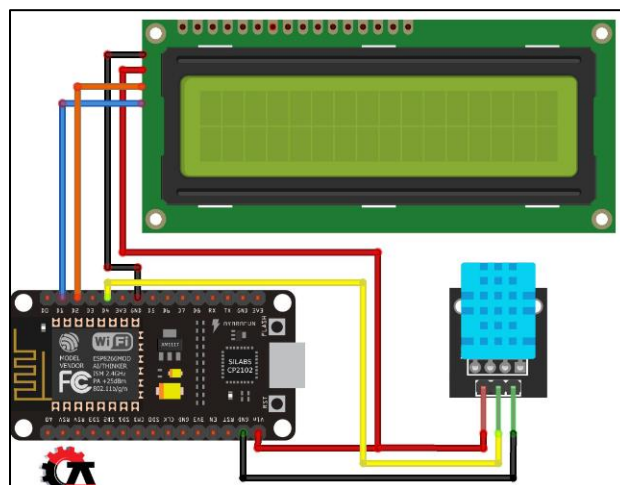
โหลด library โมดูลจอ LCD NodeMCU ESP8266

LiquidCrystal_I2C

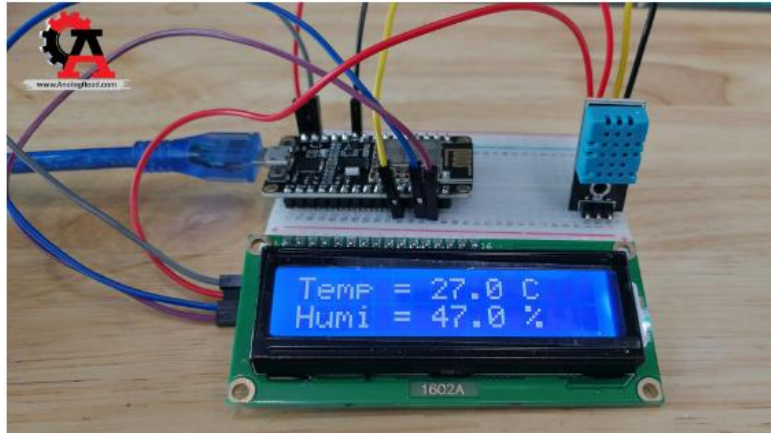
https://www.arduinoall.net/arduino-tutor/code/LiquidCrystal_i2c.rar

ขั้นตอนการทำงาน

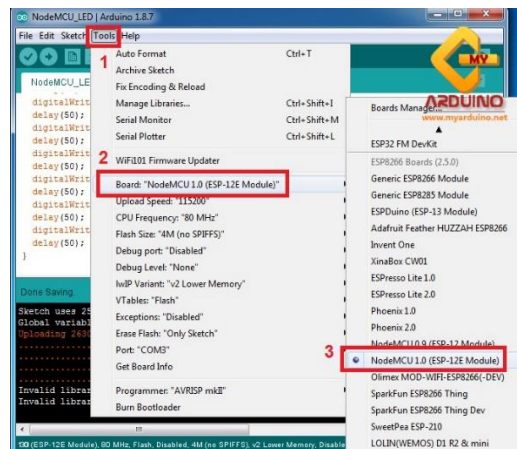
1. ออกแบบการเชื่อมต่อวงจรตามโจทย์กำหนดให้



2. ต่อวงจรระหว่าง NodeMCU ลงบน Breadboard ตามที่ออกแบบไว้



2. เสียบบอร์ด NodeMCU เข้ากับเครื่องคอมพิวเตอร์ เลือกบอร์ด และ Comport



3. เขียนโปรแกรม และทำการอัปโหลดโค้ด Arduino แสดงผลลัพธ์ที่ได้

```
#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();
}

```

