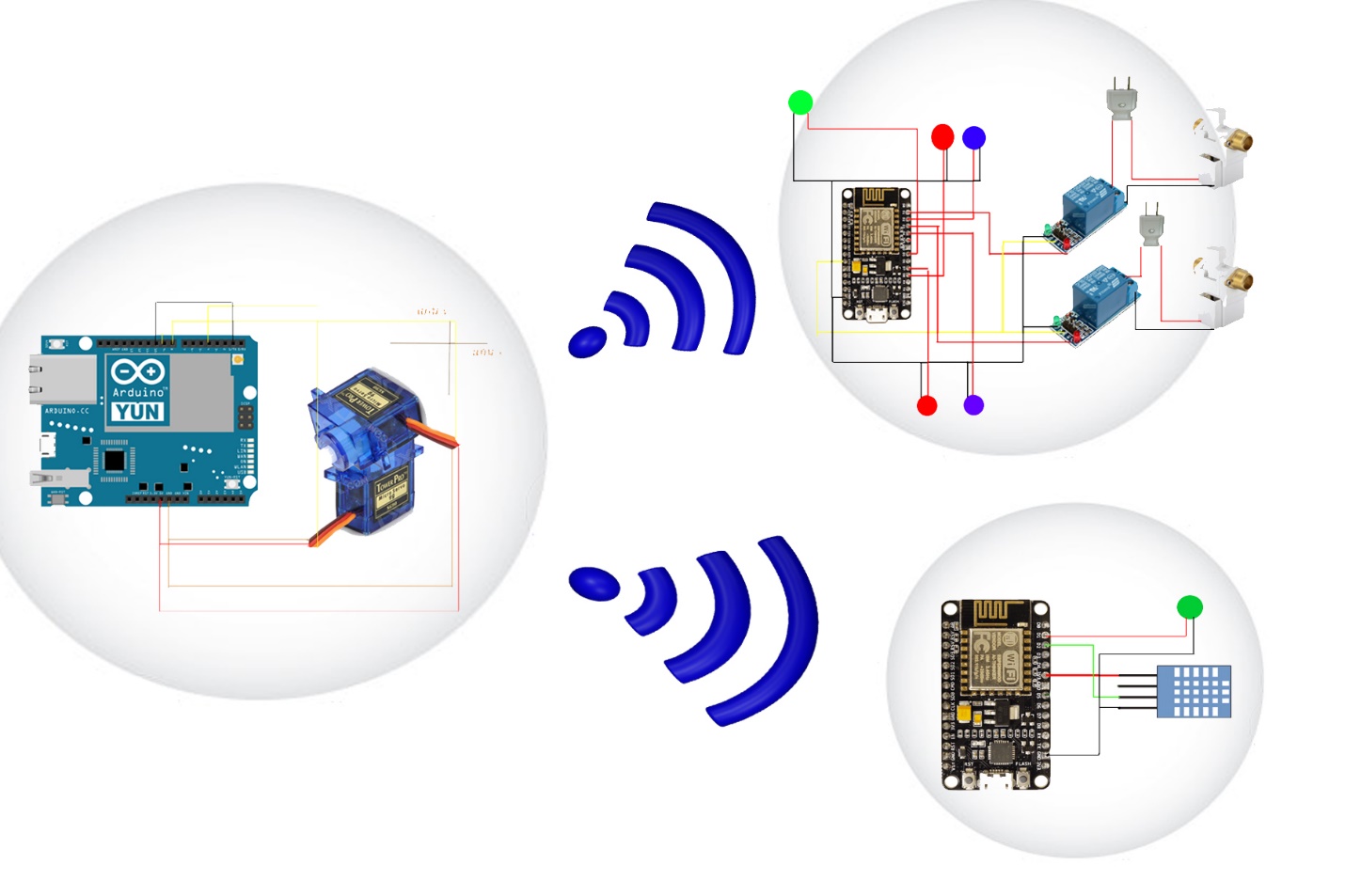
ภาพรวมการทำงานของแผงวงจรในส่วนการควบคุม การ เปิดปิด น้ำ และส่งค่าอุณภูมิ



ขั้นตอนการทำในส่วนของฮาร์ดแวร์

อุปกรณ์มีดังนี้

1.NodeMCU

2.Servo Motor

3.Sensor DHT11

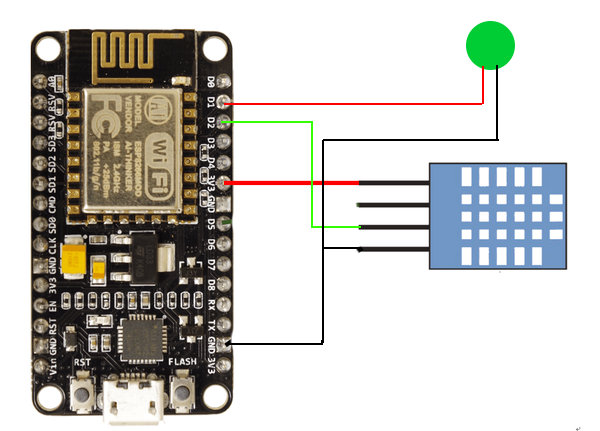
4.Arduino yun

5.รีเลย์ (Relay)

**ลักษณะการทำงาน**

แบ่งออกเป็น 3 ส่วน โดยให้ Arduino yun เป็น Savever หลักทำการรับส่งข้อมูลไปยังฝั่งรับ คือ NodeMCU กับตัว Sensor DHT11 ที่ต่อกันอยู่ และ NodeMCU ต่อกับ รีเลย์ (Relay) และ โซลินอยด์วาล์ว โซลินอยด์

ขั้นตอนที่ 1ทำการต่ออุปกรณ์ NodeMCU เข้ากับตัว Sensor DHT11



1.ต่อสายไฟเส้นสีดำจาก NodeMCU ขา Gnd เข้ากับ Sensor DH11 ขา Gnd

2.ต่อสายไฟเส้นสีแดงจาก NodeMCU ขา Vcc เข้ากับ Sensor DH11 ขา Vcc

3.ต่อสายไฟเส้นสีเขียวจาก NodeMCU ขา D2 เข้ากับ Sensor DH11 ขา out

4.ต่อสายไฟเส้นสีแดงจาก NodeMCU ขา D1 เข้ากับ LED

5.ต่อสายไฟเส้นสีดำจาก NodeMCU ขา Gnd เข้ากับ LED

หลักการทำงานของตัว NodeMCU กับ Sensor DH11 คือ ตัว NodeMCU จะจ่ายไฟเข้าตัว

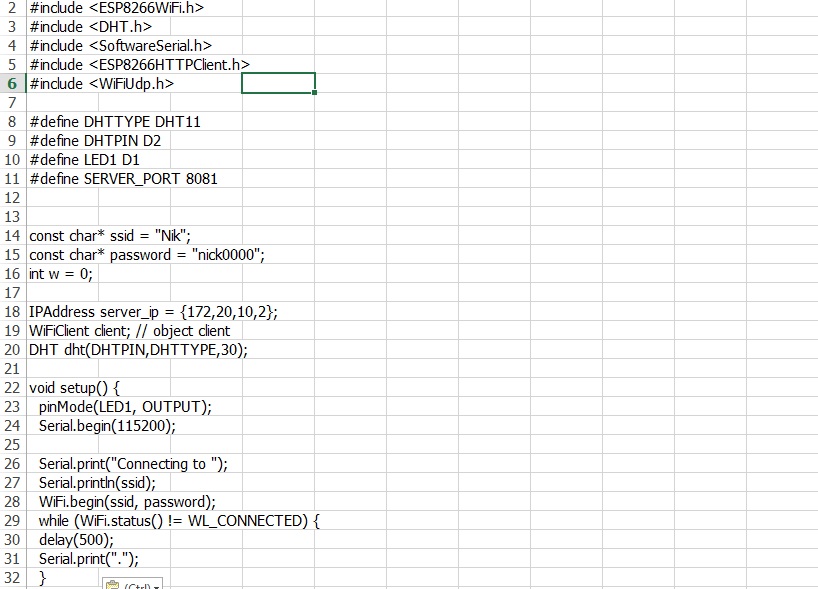
Sensor DH11 5V และต่อขา Gnd เพือให้ตัว Sensor DH11 ครบวงจรพร้อมที่จะทำงาน

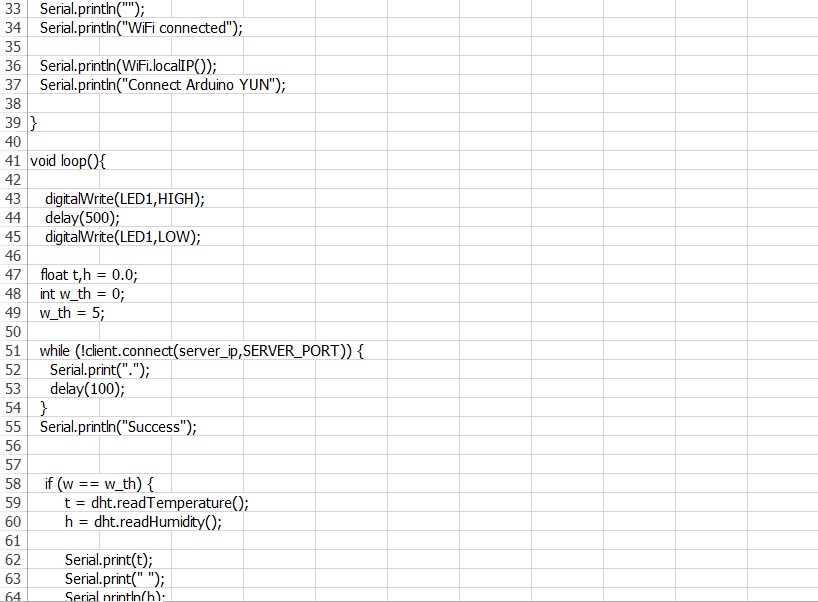
ส่วนขา out ของตัว Sensor DH11 จะส่งค่า อุณภูมิเข้ามาให้ ตัว NodeMCU เพื่อที่ตัว NodeMCU จะส่งค่า

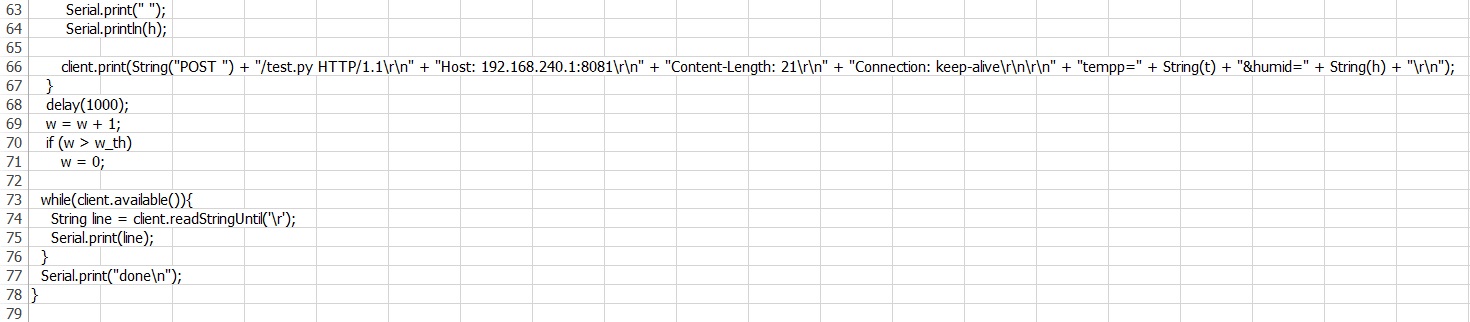
ขึ้นไปบนเว็ปหรือ app

 อุปกรณ์ Sensor DH11 ใช้การสื่อสารกับ MCU ด้วยวิธีการสื่อสารแบบอนุกรุมสองทางโดยใช้สายเส้นเดียว   การสื่อสารแบบนี้จะใช้สายสื่อสารเพียงเส้นเดียวและส่งข้อมูลได้ทั้งจาก MCU ไปที่ตัว DHT11

โค๊ดการทำงานของ Sensor DHT11







อธิบายการทำงานของโค๊ด

บรรทัดที่ 2-11 เป็นการประกาศเรียกใช้งาน liberly และกำหนดค่าขาการใช้งาน

บรรทัดที่ 14-19 เป็นการกำหนดให้ NodeMCU ใช้งานกับ WiFi

บรรทัดที่ 22-37 เป็นการสั่งให้ NodeMCU เชื่อมต่อกับ WiFi และ เมื่อตัว NodeMCU เชื่องต่อกับ Savever

แล้วให้ แสดงหมายเลข IP ของ Server กับ ข้อความว่า Connect Arduino YUN

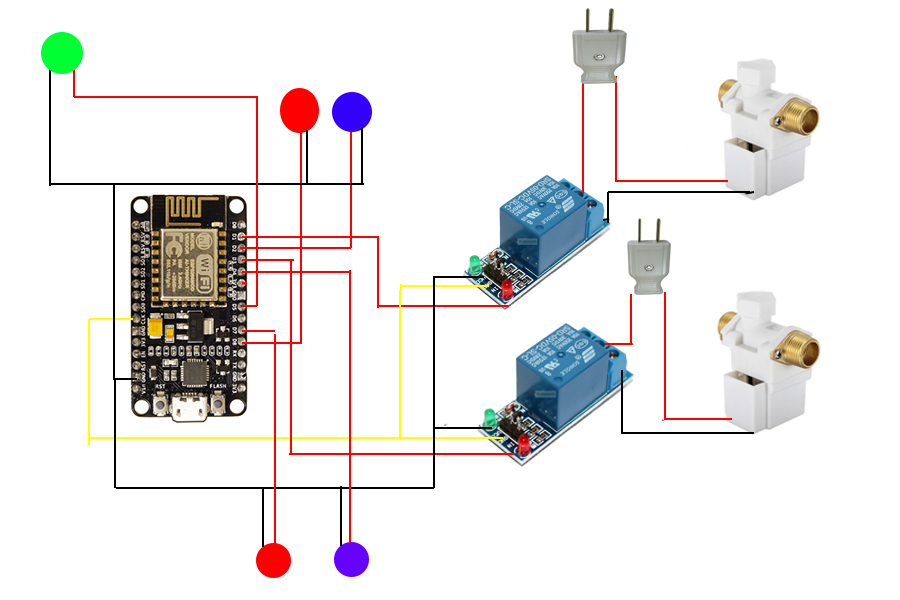
บรรทัดที่ 33-35 สั่งให้ไฟ Led กระพริบ

บรรทัดที่ 47-64 สังให้ Sensor DHT11 ส่งค่า อุณหภูมิ กับ ความชื้น มาแล้ว Plint ค่าออกมา

บรรทัดที่ 66 สั่งให้ส่งข้อมูลไปยัง Savever ที่กำหนดไว้

ขั้นตอนที่ 2

NodeMCU ต่อกับ รีเลย์ (Relay) และ โซลินอยด์วาล์ว โซลินอยด์



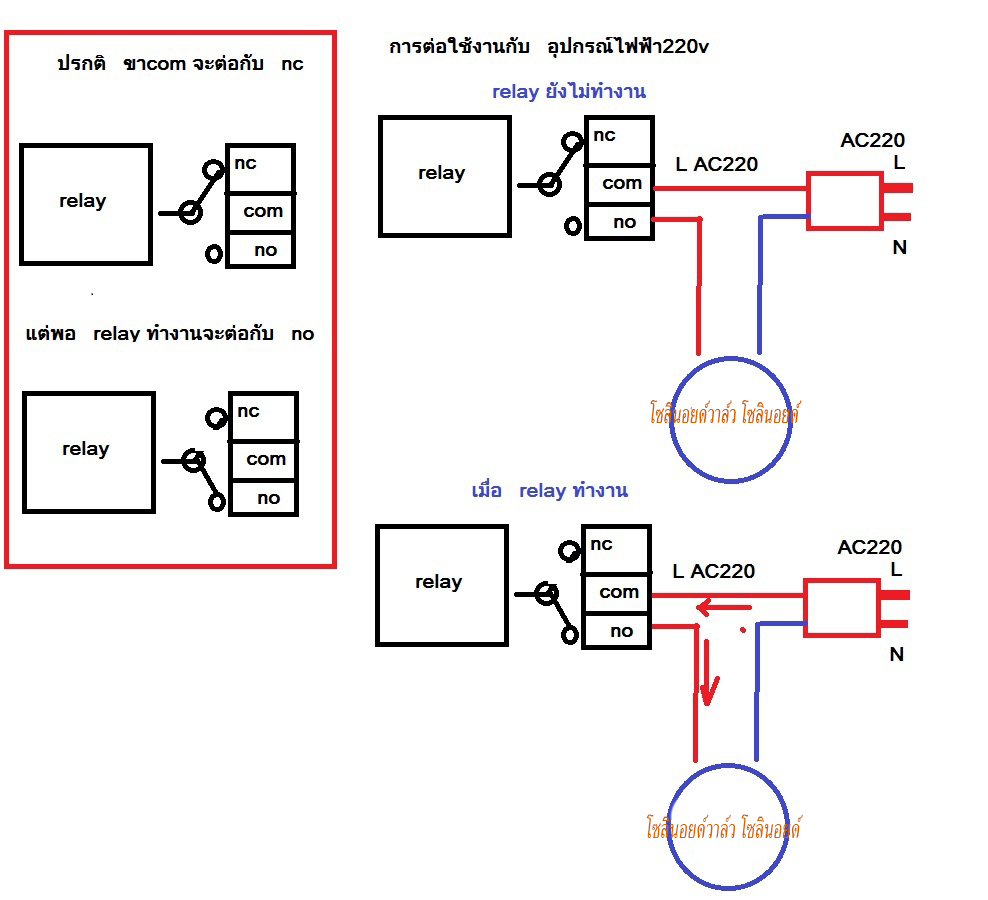
หลักงานทำงานของ NodeMCU กับ รีเลย์ (Relay) และ โซลินอยด์วาล์ว โซลินอยด์

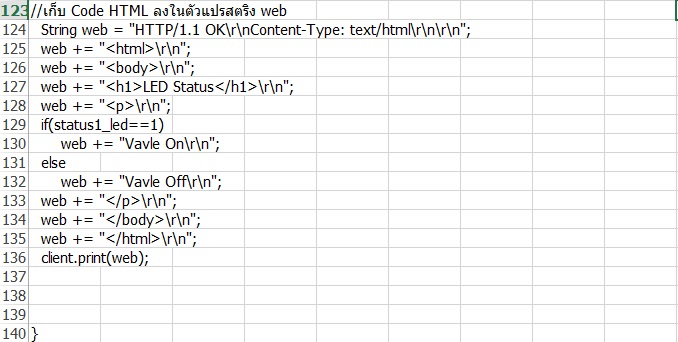
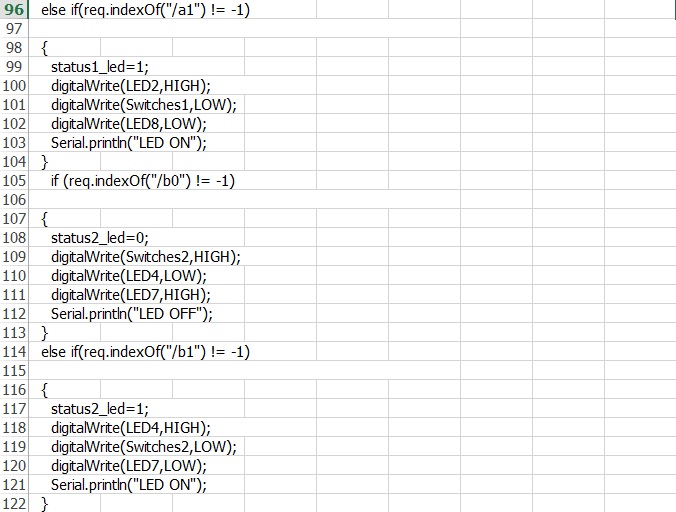
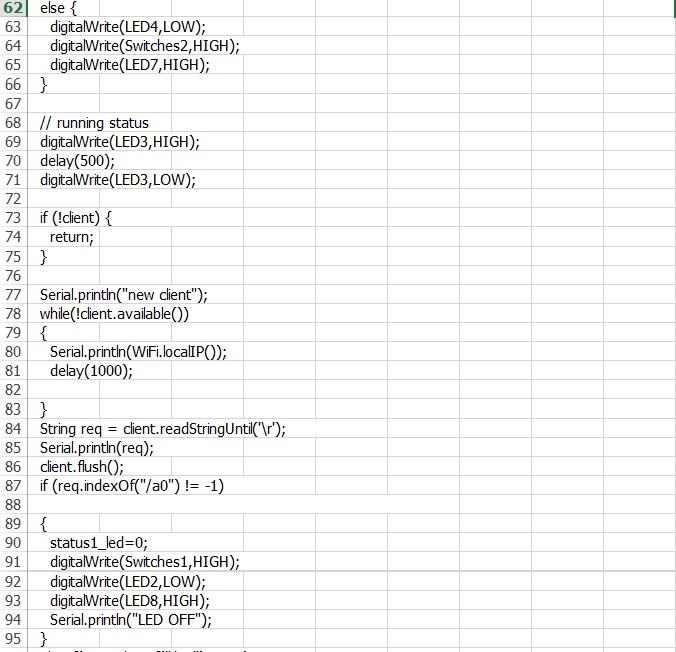
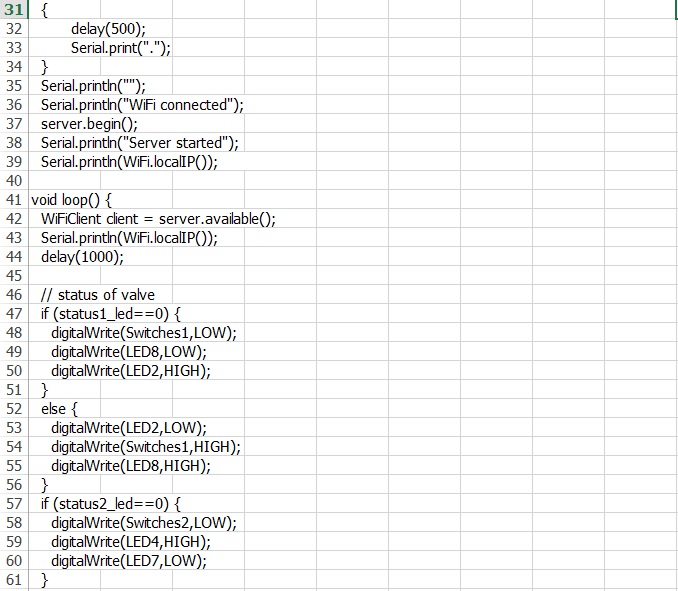
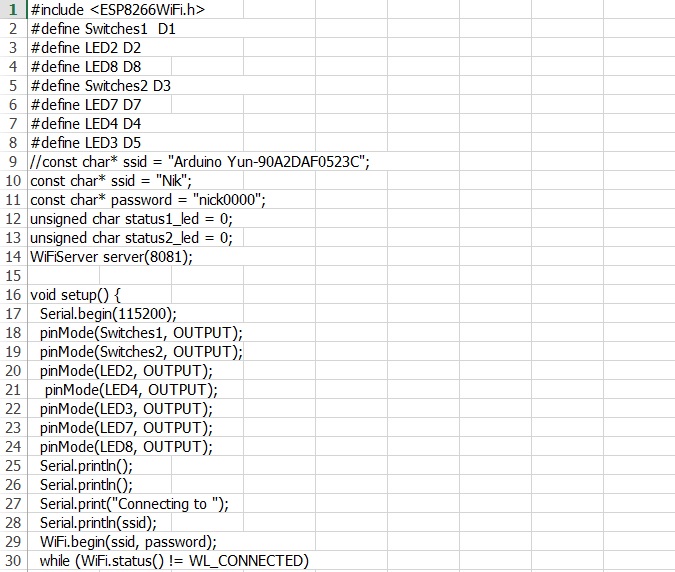
NodeMCU จะรับคำสั่งจาก app หรือ เว็ปไซต์ ที่จะส่งค่ามาเป็น 1 กับ 0 เพื่อให้ NodeMCU รับค่า และส่งต่อไปให้ รีเลย์ (Relay) ใช้ในการเปิดปิด โซลินอยด์วาล์ว โซลินอยด์

โดย รีเลย์ (Relay) จะแยกออกเป็น 2 ตัว โดยตัวบนดูจากรูปข้างบนจะเป็นตัวให้น้ำ และตัวด้านข้าง

จะเป็นตัวให้ปุ่ยทางน้ำ โดยการทำงานจะมีไฟเป็นตัวบอกการปิดเปิดตามการใช้งาน

ภาพการทำงาน รีเลย์ (Relay) กับ โซลินอยด์วาล์ว โซลินอยด์



การเขียนโปรแกรมการทำงานของ NodeMCU ต่อกับ รีเลย์ (Relay) และ โซลินอยด์วาล์ว โซลินอยด์

อธิการการเขียนโปรแกรม

บรรทัดที่ 2 กำหนดขาที่ต่อ สวิกช์ ให้น้ำ เป็นขา D1

บรรทัดที่ 3 กำหนดขาที่ต่อเป็น D2 เปิดไฟสีน้ำเงินเมื่อขา D1 ไม่ทำงาน

บรรทัดที่ 4 กำหนดขาที่ต่อเป็น D8 เปิดไฟสีแดงเมื่อ D1ทำงาน

บรรทัดที่ 5 กำหนดขาที่ต่อ สวิกช์ ให้ยา เป็นขา D3

บรรทัดที่ 6 กำหนดขาที่ต่อเป็น D7 เปิดไฟสีแดงเมื่อ D3 ทำงาน

บรรทัดที่ 7 กำหนดขาที่ต่อเป็น D4 เปิดไฟสีน้ำเงินเมื่อขา D3 ไม่ทำงาน

บรรทัดที่ 8 กำหนดขาที่ต่อเป็น D5 เปิดไฟสีเขียวตลอดเวลาเมื่อมีการส่งข้อมูลไปให้ Savever

บรรทัดที่ 10 กำหนด SSID (อย่าลืมแก้เป็นของตัวเอง)

บรรทัดที่ 11 กำหนด Password(อย่าลืมแก้เป็นของตัวเอง)

บรรทัดทื 12-13 กำหนดตัวแปร ที่เก็บค่าสถานะของ LED

บรรทัดที่ 14 กำหนดใช้งาน TCP Server ที่ Port 80

บรรทัดที่ 17 เปิดใช้ Serial

บรรทัดที่ 18-24 กำหนด Pin ที่ต่อทั้งหมดเป็น Output

บรรทัดที่ 29 เชื่อมต่อกับ AP

บรรทัดที่ 30 รอการเชื่อมต่อ

บรรทัดที่ 36 แสดงข้อความเชื่อมต่อสำเร็จ

บรรทัดที่ 37 เปิด TCP Server

บรรทัดที่ 39 แสดงหมายเลข IP ของ Server

บรรทัดที่ 43 รอรับ การเชื่อมต่อจาก Client

บรรทัดที่ 47-56 ในส่วนของ If เมื่อ status1\_led==0 กำหนดให้ Switches1 เป็น LOW

LED8 เป็น LOW และ LED2 เป็น HIGH ในส่วนของ else กำหนดให้ LED2 เป็น LOW

และ Switches1 เป็น HIGH และ LED8 เป็น HIGH

บรรทัดที่ 57-66 ในส่วนของ If เมื่อ status2\_led==0 กำหนดให้ Switches2 เป็น LOW

LED4 เป็น HIGH และ LED7 เป็น LOW ในส่วนของ else กำหนดให้ LED4 เป็น LOW

และ Switches1 เป็น HIGH และ LED7 เป็น HIGH

บรรทัดที่ 73-74 ถ้าไม่มี Client เข้ามาให้เริ่มกับไปวน loop รอรับใหม่

บรรทัดที่ 84 อ่านค่าที่ได้รับจากclient จากข้อมูลแรกถึง ‘\r’

บรรทัดที่ 85 แสดงค่าที่ได้รับทาง Serial

บรรทัดที่ 87 ตรวจสอบว่า data ที่เข้ามามีข้อความ ”/a0” ถ้ามีให้ทำงานบรรทัดต่อไปถ้าไม่ใช้ให้ลงไปทื่ else

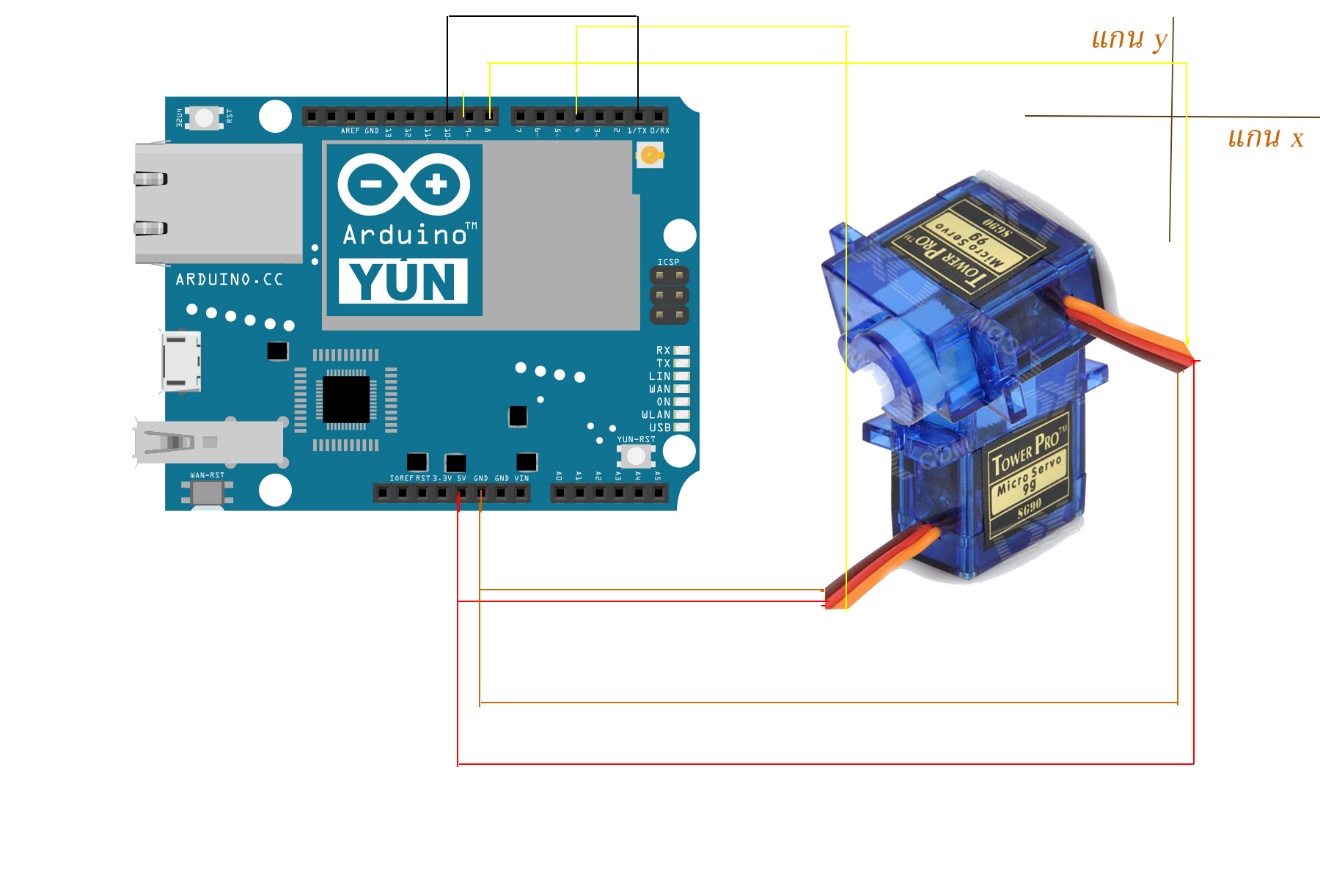
บรรทัดที่ 90 ถ้ามีให้กำหนดค่า ในตัวแปรใน status\_led=0

บรรทัดที่ 96 ตรวจสอบว่า data ที่เข้ามามีข้อความ ”/a1” ถ้ามีให้ทำงานบรรทัดต่อไป

บรรทัดที่ 105 ตรวจสอบว่า data ที่เข้ามามีข้อความ ”/b0” ถ้ามีให้ทำงานบรรทัดต่อไป

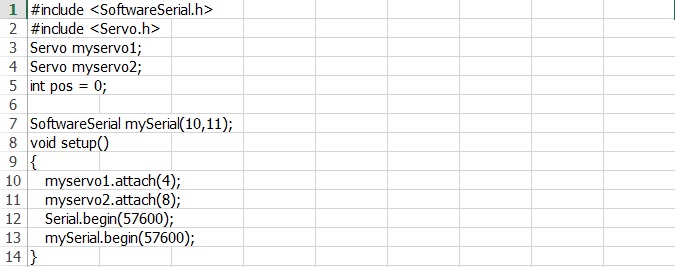
บรรทัดที่ 114 ตรวจสอบว่า data ที่เข้ามามีข้อความ ”/b1” ถ้ามีให้ทำงานบรรทัดต่อไป

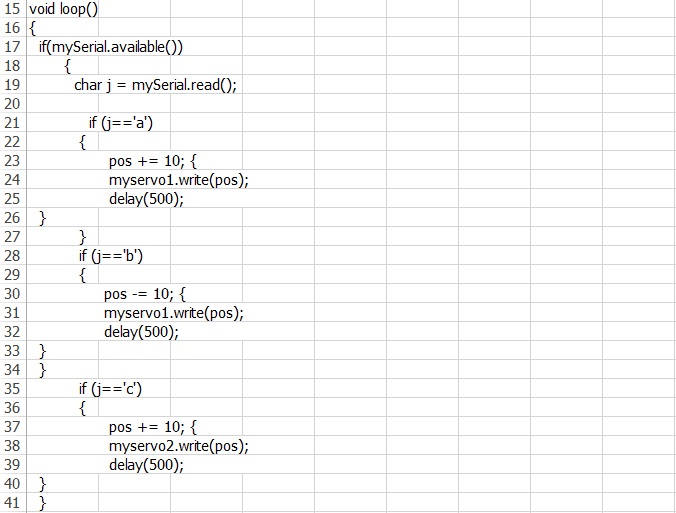
ขั้นตอนที่ 3 Arduino yun กับ Servo Motor

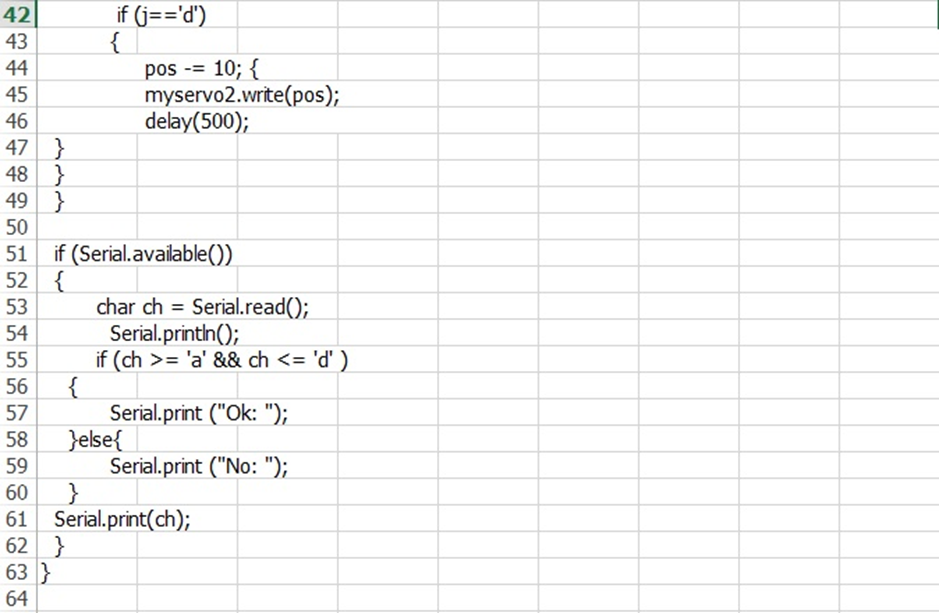


หลักการทำงานของ Arduino yun กับ Servo Motor ใช้ขา 4 กับขา 8 ของ Arduino yun จ่ายสัญญาณพัลซ์เข้ามายัง Servo Motor ขา PWM เพื่อควบคุม Servo Motor และ Servo Motor จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลซ์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุน เป็น Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ

การเขียนโปรแกรม







อธิบายการเขียนโปณแกรม

บรรทัดที่ 1-5 เป็นการประกาศการใช้งาน liberly และประกาศตัวแปร

บรรทัดทื่ 7 ประการการใช้งาน SoftwareSerial และ mySerial

บรรทัดที่ 10 ประการการใช้งานให้ myservo1 ต่อกับขา 4

บรรทัดที่ 11 ประการการใช้งานให้ myservo2 ต่อกับขา 8

บรรทัดที่ 17 ตรวจสอบค่าที่ได้รับจาก mySerial

บรรทัดที่ 19 อ่านค่าที่ได้รับจาก mySerial และเก็บไว้ที่ตัวแปร j

บรรทัดที่ 21-27 ถ้าค่า j เท่ากับ a ให้ myservo1 หมุน 10 องศา

บรรทัดที่ 28-34 ถ้าค่า j เท่ากับ b ให้ myservo1 หมุน -10 องศา

บรรทัดที่ 35-41 ถ้าค่า j เท่ากับ c ให้ myservo2 หมุน 10 องศา

บรรทัดที่ 42-49 ถ้าค่า j เท่ากับ d ให้ myservo2 หมุน -10 องศา

บรรทัดที่ 51-63 ตรวจสอบค่าที่ส่งมา print บน myseria