

หน่วยที่ 9: การโปรแกรมควบคุมการทำงานของมอเตอร์เซอร์โว

จุดประสงค์

- สามารถเขียนโปรแกรมควบคุมการทำงานของมอเตอร์เซอร์โวได้
- สามารถควบคุมการทำงานและแสดงผลผ่าน Blynk ได้

กิจกรรม

9.1 กิจกรรม 1: ควบคุมและอ่านค่าองศาของมอเตอร์เซอร์โวผ่าน Blynk

โจทย์กิจกรรม

ให้ผู้เรียนเขียนโปรแกรมควบคุมและอ่านค่าองศาของมอเตอร์เซอร์โวผ่าน Blynk

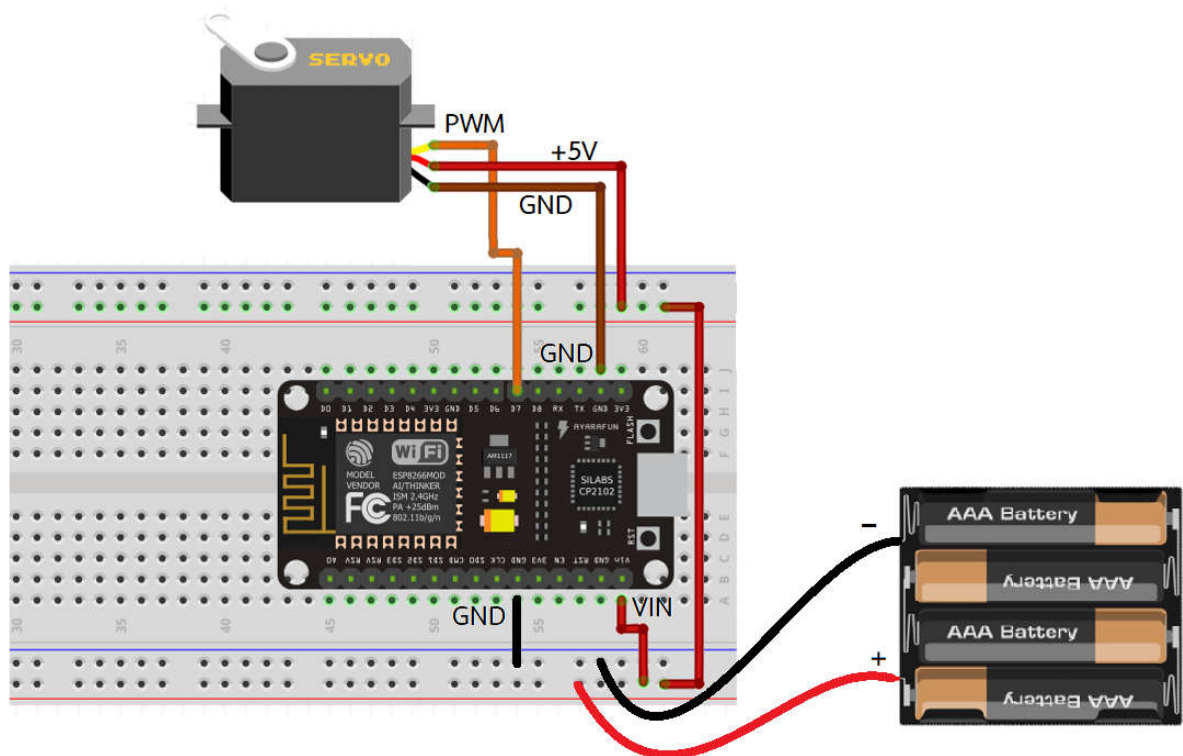
- ให้ผู้เรียนต่อวงจรและติดตั้งมอเตอร์เซอร์โวตามขั้นตอน
- เขียนโปรแกรมเพื่อควบคุมและแสดงผลผ่าน Blynk
- สร้างอุปกรณ์ใน Blynk เพื่อควบคุมและแสดงผล

เครื่องมือและอุปกรณ์

- | | |
|--|---------------|
| ○ บอร์ด NodeMCU/ESP8266 V.2 | 1 ตัว |
| ○ คอมพิวเตอร์ (แบบตั้งโต๊ะหรือพกพา) ที่มีอุปกรณ์รับสัญญาณ Wifi | 1 เครื่อง |
| ○ สมาร์ทโฟน (Android/iOS) พร้อมสัญญาณ 3G | 1 เครื่อง |
| ○ มอเตอร์เซอร์โว | 1 ตัว |
| ○ แบตเตอรี่ขนาด AA 4 ก้อนพร้อมที่ใส่ | 1 ชุด |
| ○ สายต่อชนิดตัวผู้ทั้งสองด้าน | เท่าที่จำเป็น |

ลำดับขั้นตอนการทำงาน

1. ให้ต่อวงจรตามรูปที่ 1



รูปที่ 1 วงจรสำหรับกิจกรรมที่ 1

2. ให้สร้าง Value Display ต่อกับขาจำลอง Digital V0 ใน Blynk เพื่อดูค่าที่เราปรับค่า

3. สร้าง Slider ต่อกับตั้งค่าจำลอง Digital V1 ใน Blynk และแก้ค่าสูงสุดเป็น 180
4. เขียนโค้ดตามในรูปที่ 2 ลงใน Arduino IDE

```

1 #include <Servo.h> //เรียกใช้ไลบรารีเซอร์โว
2 Servo myservo; //สร้างตัวแปรสำหรับใช้งานเซอร์โว
3 int degree; //สร้างตัวแปรสำหรับส่งค่าองศา
4 int readdegree; //สร้างตัวแปรสำหรับรับค่าองศา
5 #define BLYNK_PRINT Serial //ฟังก์ชันการแสดงผลการเชื่อมต่อของ blynk
6 #include <ESP8266WiFi.h> //เรียกใช้ไลบรารีESP8266WiFi
7 #include <BlynkSimpleEsp8266.h> //เรียกใช้ไลบรารีBlynkSimpleEsp8266
8 char auth[] = " "; //หมายเลขที่เคที่ได้รับจากการสร้างโปรเจกต์ในblynk
9 char ssid[] = " "; //Username ของ wifi ที่ต้องการเชื่อมต่อ
10 char pass[] = " "; //Password ของ wifi ที่ต้องการเชื่อมต่อ
11 void setup() {
12     myservo.attach(D7); //กำหนดเซอร์โวดังชื่อว่าmyservo ที่ขา D7
13     Serial.begin(9600); //กำหนดความเร็วในการส่งข้อมูล
14     Blynk.begin(auth, ssid, pass); //กำหนดค่าการเชื่อมต่อ
15 }
16 void loop()
17 {
18     Blynk.run(); //เรียกใช้งานBlynk
19     readdegree = myservo.read(); //ตัวแปรรับค่าจากคำสั่งอ่านสัญญาณที่ส่งมาเซอร์โว
20     Blynk.virtualWrite(V0, readdegree); //ส่งค่าองศาที่อ่านได้ไปยังขา จำลอง V0 เพื่อเอาแมเน Blynk
21 }
22 BLYNK_WRITE(V1) //ฟังก์ชันส่งงานจากBlynk
23 {
24     degree = param.asInt(); //ตัวแปรเก็บค่าที่รับมาจากอุปกรณ์แมเน Blynk
25     myservo.write(degree); //ส่งงานเซอร์โวดังชื่อ myservo ให้หมุนตามค่าที่รับมาจากBlynk
26 }

```

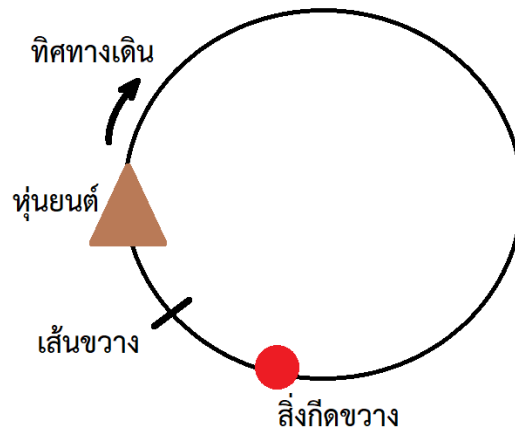
รูปที่ 2 โค้ดของกิจกรรมที่ 1

5. เชื่อมต่อกับแอปพลิเคชัน Blynk
6. ทำการทดลองและสังเกตผลการทดลอง

9.2 กิจกรรมที่ 2: หุ่นยนต์กวาดสิ่งกีดขวาง

โจทย์กิจกรรม

ให้ผู้เรียนสร้างและเขียนโปรแกรมหุ่นยนต์กวาดสิ่งกีดขวางตามรูปที่ 3



รูปที่ 3 เส้นทางที่ให้หุ่นยนต์เดินตามเส้น (ตัวอย่าง)

- สั่งให้หุ่นยนต์เดินตามเส้น
- เมื่อพบสิ่งกีดขวาง ให้ทำการผลักรถให้ออกนอกเส้นทางเดิน และเดินต่อไป
- เมื่อถึงเส้นชัย (เส้นตัดขวาง) ให้หยุดการทำงาน

เครื่องมือและอุปกรณ์

- | | |
|--|---------------|
| ○ บอร์ด NodeMCU/ESP8266 V.2 | 1 ตัว |
| ○ คอมพิวเตอร์ (แบบตั้งโต๊ะหรือพกพา) ที่มีอุปกรณ์รับสัญญาณ Wifi | 1 เครื่อง |
| ○ โมดูลเซนเซอร์อัลตราโซนิก | 1 ตัว |
| ○ สมาร์ทโฟน (Android/iOS) พร้อมสัญญาณ 3G | 1 เครื่อง |
| ○ หุ่นยนต์ที่ประกอบเสร็จเรียบร้อยแล้วจากกิจกรรมก่อนหน้านี้ | 1 ชุด |
| ○ แบตเตอรี่ขนาด AA 4 ก้อนพร้อมที่ใส่ | 1 ชุด |
| ○ สายต่อชนิดตัวผู้ทั้งสองด้าน | เท่าที่จำเป็น |

ลำดับขั้นตอนการทำงาน

1. ต่อดังตามรูปที่ 4 (หรือใช้วงจรจากกิจกรรมที่ 1 กับหุ่นยนต์จากกิจกรรมก่อนหน้านี้)


```

1 #include <Servo.h> //เรียกใช้ไลบรารีเซอร์โว
2 Servo myarm; //สร้างตัวแปรสำหรับใช้งานเซอร์โว
3 #define PWM_ML D2 //สร้างตัวแปร PWM สำหรับมอเตอร์ซ้าย คอกับขา D5
4 #define in1_ML D4 //สร้างตัวแปร in1 สำหรับมอเตอร์ซ้าย คอกับขา D4
5 #define in2_ML 10 //สร้างตัวแปร in2 สำหรับมอเตอร์ซ้าย คอกับขา GPIO10
6 #define in1_MR D8 //สร้างตัวแปร in1 สำหรับมอเตอร์ขวา คอกับขา D8
7 #define in2_MR D9 //สร้างตัวแปร in2 สำหรับมอเตอร์ขวา คอกับขา D9
8 #define PWM_MR D3 //สร้างตัวแปร PWM สำหรับมอเตอร์ขวา คอกับขา D6
9 #define IRSensorL D5 //ขาเอาต์พุตของเซ็นเซอร์ คอที่ขา D5
10 #define IRSensorR D6 //ขาเอาต์พุตของเซ็นเซอร์ คอที่ขา D6
11 int LsensorValue, RsensorValue; //สร้างตัวแปรสำหรับเก็บค่าจากเซ็นเซอร์
12 int Speed = 450; //สร้างตัวแปร Speed สำหรับกำหนดความเร็ว 0-1023
13 #include <Ultrasonic.h> //เรียกใช้ไลบรารีUltrasonic.h
14 Ultrasonic ultrasonic(D0, D1); //ฟังก์ชันกำหนดขา Tric ที่ขา D0 Echo ที่ขา D1
15 int distanceCM; //ตัวแปรเก็บค่าระยะทางเป็นเซนติเมตร
16 void setup()
17 {
18     myarm.attach(D7); //กำหนดเซอร์โวดังชื่อว่าmyarm ที่ขา D7
19     pinMode(PWM_ML, OUTPUT); //กำหนดโหมดการทำงานของเอาต์พุต
20     pinMode(in1_ML, OUTPUT);
21     pinMode(in2_ML, OUTPUT);
22     pinMode(PWM_MR, OUTPUT);
23     pinMode(in1_MR, OUTPUT);
24     pinMode(in2_MR, OUTPUT);
25     pinMode(IRSensorL, INPUT); //กำหนดพอร์ต์ IRSensorL(D7) เป็น อินพุต
26     pinMode(IRSensorR, INPUT); //กำหนดพอร์ต์ IRSensorR(D8) เป็น อินพุต
27     myarm.write(10); //กำหนดค่าเริ่มต้นให้เซอร์โวมุมอยู่ที่ 90 องศา
28 }
29 void loop()
30 {
31     distanceCM = ultrasonic.read(CM); //รับค่าจากอัลตราโซนิกมาแปลงเป็นเซนติเมตรเก็บไว้ในตัวแปร
32     if (distanceCM <= 5)
33     {
34         Stop();
35         myarm.write(160); //กำหนดมุมให้เซอร์โวมุมอยู่ที่ 160 องศา
36         delay(700); //หน่วงเวลารอการทำงานของเซอร์โว
37         myarm.write(10); //กำหนดมุมให้เซอร์โวมุมอยู่ที่ 90 องศา
38     }
39     else
40     {
41         Follow();
42     }
43 }
44 void Follow() //ฟังก์ชันสำหรับเดินตามเส้น
45 {
46     LsensorValue = digitalRead(IRSensorL); //เก็บค่าจากเซ็นเซอร์ซ้ายในตัวแปร
47     RsensorValue = digitalRead(IRSensorR); //เก็บค่าจากเซ็นเซอร์ขวาในตัวแปร
48     if (LsensorValue == 1 && RsensorValue == 0) //ถ้าเซ็นเซอร์ซ้ายเจอสีดำ และ เซ็นเซอร์ขวาเจอสีขาว
49     {
50         TurnLeft(); //เลี้ยวซ้าย
51     }
52     else if (LsensorValue == 0 && RsensorValue == 1) //ถ้าเซ็นเซอร์ซ้ายเจอสีขาว และ เซ็นเซอร์ขวาเจอสีดำ
53     {
54         TurnRight(); //เลี้ยวขวา
55     }
56     else if (LsensorValue == 0 && RsensorValue == 0) //ถ้าเซ็นเซอร์เจอสีขาวทั้งคู่
57     {
58         ForWard(); //เดินตรง
59     }
60     else //เจออะไรก็แล้วแต่
61     {
62         Stop(); //หยุด
63     }
64 }

```

รูปที่ 5 โค้ดสำหรับกิจกรรมที่ 2 (มีต่อหน้าต่อไป)


```

65 void ForWard() //ฟังก์ชันเดินหน้า
66 {
67     //สั่งให้มอเตอร์ซ้าย เดินหน้า
68     digitalWrite(in1_ML, HIGH); //สั่งงานให้ in1 สำหรับมอเตอร์ซ้าย เป็น HIGH
69     digitalWrite(in2_ML, LOW); //สั่งงานให้ in2 สำหรับมอเตอร์ซ้าย เป็น LOW
70     analogWrite(PWM_ML, Speed); //สร้างสัญญาณ PWM สำหรับมอเตอร์ซ้าย
71     //สั่งให้มอเตอร์ขวา เดินหน้า
72     digitalWrite(in1_MR, HIGH); //สั่งงานให้ in1 สำหรับมอเตอร์ขวา เป็น HIGH
73     digitalWrite(in2_MR, LOW); //สั่งงานให้ in2 สำหรับมอเตอร์ขวา เป็น LOW
74     analogWrite(PWM_MR, Speed); //สร้างสัญญาณ PWM สำหรับมอเตอร์ขวา
75 }
76 void TurnLeft() //ฟังก์ชันหมุนซ้าย
77 {
78     //สั่งให้มอเตอร์ซ้าย ถอยหลัง
79     digitalWrite(in1_ML, LOW); //สั่งงานให้ in1 สำหรับมอเตอร์ซ้าย เป็น LOW
80     digitalWrite(in2_ML, HIGH); //สั่งงานให้ in2 สำหรับมอเตอร์ซ้าย เป็น HIGH
81     analogWrite(PWM_ML, Speed); //สร้างสัญญาณ PWM สำหรับมอเตอร์ซ้าย
82     //สั่งให้มอเตอร์ขวา เดินหน้า
83     digitalWrite(in1_MR, HIGH); //สั่งงานให้ in1 สำหรับมอเตอร์ขวา เป็น HIGH
84     digitalWrite(in2_MR, LOW); //สั่งงานให้ in2 สำหรับมอเตอร์ขวา เป็น LOW
85     analogWrite(PWM_MR, Speed); //สร้างสัญญาณ PWM สำหรับมอเตอร์ขวา
86 }
87 void TurnRight() //ฟังก์ชันหมุนขวา
88 {
89     //สั่งให้มอเตอร์ซ้าย เดินหน้า
90     digitalWrite(in1_ML, HIGH); //สั่งงานให้ in1 สำหรับมอเตอร์ซ้าย เป็น HIGH
91     digitalWrite(in2_ML, LOW); //สั่งงานให้ in2 สำหรับมอเตอร์ซ้าย เป็น LOW
92     analogWrite(PWM_ML, Speed); //สร้างสัญญาณ PWM สำหรับมอเตอร์ซ้าย
93     //สั่งให้มอเตอร์ขวา ถอยหลัง
94     digitalWrite(in1_MR, LOW); //สั่งงานให้ in1 สำหรับมอเตอร์ขวา เป็น LOW
95     digitalWrite(in2_MR, HIGH); //สั่งงานให้ in2 สำหรับมอเตอร์ขวา เป็น HIGH
96     analogWrite(PWM_MR, Speed); //สร้างสัญญาณ PWM สำหรับมอเตอร์ขวา
97 }
98 void Stop() //ฟังก์ชันหยุดเคลื่อนที่
99 {
100     //สั่งให้มอเตอร์ซ้าย หยุด
101     digitalWrite(in1_ML, LOW); //สั่งงานให้ in1 สำหรับมอเตอร์ซ้าย เป็น LOW
102     digitalWrite(in2_ML, LOW); //สั่งงานให้ in2 สำหรับมอเตอร์ซ้าย เป็น LOW
103     analogWrite(PWM_ML, 0); //สร้างสัญญาณ PWM เป็น 0 เพื่อ หยุด
104     //สั่งให้มอเตอร์ขวา หยุด
105     digitalWrite(in1_MR, LOW); //สั่งงานให้ in1 สำหรับมอเตอร์ขวา เป็น LOW
106     digitalWrite(in2_MR, LOW); //สั่งงานให้ in2 สำหรับมอเตอร์ขวา เป็น LOW
107     analogWrite(PWM_MR, 0); //สร้างสัญญาณ PWM เป็น 0 เพื่อ หยุด
108 }
109 void BackWard() //ฟังก์ชันถอยหลัง
110 {
111     //สั่งให้มอเตอร์ซ้าย ถอยหลัง
112     digitalWrite(in1_ML, LOW); //สั่งงานให้ in1 สำหรับมอเตอร์ซ้าย เป็น LOW
113     digitalWrite(in2_ML, HIGH); //สั่งงานให้ in2 สำหรับมอเตอร์ซ้าย เป็น HIGH
114     analogWrite(PWM_ML, Speed); //สร้างสัญญาณ PWM สำหรับมอเตอร์ซ้าย
115     //สั่งให้มอเตอร์ขวา ถอยหลัง
116     digitalWrite(in1_MR, LOW); //สั่งงานให้ in1 สำหรับมอเตอร์ขวา เป็น LOW
117     digitalWrite(in2_MR, HIGH); //สั่งงานให้ in2 สำหรับมอเตอร์ขวา เป็น HIGH
118     analogWrite(PWM_MR, Speed); //สร้างสัญญาณ PWM สำหรับมอเตอร์ขวา
119 }

```

รูปที่ 5 โค้ดสำหรับกิจกรรมที่ 2 (ต่อ)