

ใบงานที่ 11

เรื่อง หุ่นยนต์เคลื่อนที่ตามเส้น 90 องศาและนับเส้นทางแยก

วัตถุประสงค์ (เพื่อให้นักเรียนสามารถ)

1. เขียนโปรแกรมควบคุมหุ่นยนต์เคลื่อนที่ตามเส้น 90 องศาได้
2. เขียนโปรแกรมควบคุมหุ่นยนต์เคลื่อนที่นับเส้นทางแยกได้
3. บันทึกผลการทดลองได้ถูกต้อง
4. สรุปผลการทดลองได้ตามวัตถุประสงค์

เครื่องมือและอุปกรณ์

- | | |
|------------------------------------|-----------------|
| 1. คอมพิวเตอร์ | จำนวน 1 เครื่อง |
| 2. หุ่นยนต์เดินตามเส้นแบบอัตโนมัติ | จำนวน 1 ตัว |
| 3. สาย USB Serial | จำนวน 1 สาย |

ทฤษฎีเบื้องต้น

11.1 คำสั่ง pinMode

คือคำสั่งที่มีไว้สำหรับกำหนดการทำงานของ pin ที่ต้องการใช้งาน ให้ทำงานเป็นแบบ OUTPUT หรือ INPUT การทำงานเป็น output เช่น การส่งจ่ายแรงดัน 5 volt. การทำงานเป็น input เช่น การสั่งให้อ่านค่าสถานะสวิทช์ไฟ ว่าเปิดหรือปิดอยู่

```
pinMode(pin ที่ต้องการใช้งาน, OUTPUT หรือ INPUT);
```

11.2 คำสั่ง digitalWrite

คือคำสั่งที่มีไว้กำหนดการทำงานของ pin ที่ต้องการใช้งาน ให้มีสถานะลอจิกเป็น 1 (HIGH = จ่ายแรงดัน 5 volt.) หรือ 0 (LOW = หยุดจ่ายแรงดัน 5 volt.)

```
digitalWrite(pin ที่ต้องการใช้งาน, HIGH หรือ LOW);
```

11.3 คำสั่ง if

คำสั่ง if เป็นคำสั่งสำหรับใช้ตรวจสอบเงื่อนไข เพื่อสั่งให้โปรแกรมเลือกทำงาน ตามผลลัพธ์ที่ได้จากการตรวจสอบเงื่อนไขของคำสั่ง โดยมีรูปแบบคำสั่งดังนี้

```
if (เงื่อนไข) {  
    คำสั่งที่ต้องกระทำเมื่อเงื่อนไขเป็นจริง  
}
```

การทำงานของโปรแกรม เมื่อใช้การตรวจสอบเงื่อนไขแบบนี้คือ ถ้าเงื่อนไขเป็นจริงก็จะทำงานตามคำสั่งที่อยู่หลังเงื่อนไข แต่ถ้าเงื่อนไขเป็นเท็จก็จะข้ามคำสั่งที่อยู่หลังเงื่อนไขไป

11.4 คำสั่ง if – else

คำสั่ง if-else เป็นการสั่งตรวจสอบเงื่อนไขเช่นเดียวกับ if แต่ใช้สำหรับตรวจสอบเงื่อนไขที่มีเพิ่มขึ้นอีก 1 ทางเลือก โดยมีรูปแบบคำสั่งดังนี้

```
if (เงื่อนไข) {  
    คำสั่งที่ต้องกระทำเมื่อเงื่อนไขเป็นจริง  
}  
else {  
    คำสั่งที่ต้องกระทำเมื่อเงื่อนไขเป็นจริง  
}
```

ซึ่งจากรูปแบบการใช้คำสั่ง if-else แบบนี้มีความหมายเหมือนกับประโยคที่ว่า ถ้าเงื่อนไขเป็นจริง ให้ทำอย่างนี้ ไม่เช่นนั้นให้ทำอย่างนั้น ซึ่งจะเห็นได้ว่าโปรแกรมจะมีทางเลือกในทางทำงานเพิ่มขึ้นมากกว่าการใช้คำสั่ง if อีก 1 ทางเลือก รวมเป็น 2 ทาง โดยทางเลือกแรก เป็นทางเลือกที่โปรแกรมจะทำงานเมื่อ เงื่อนไขเป็นจริง ส่วนทางเลือกที่ 2 เป็นทางเลือกที่จะให้โปรแกรมทำงานเมื่อเงื่อนไขเป็นเท็จ

11.5 คำสั่ง if – else แบบหลายเงื่อนไข

คำสั่ง if-else แบบหลายเงื่อนไขเป็นการสั่งตรวจสอบเงื่อนไขเช่นเดียวกับ if-else แต่ใช้สำหรับการตรวจสอบเงื่อนไขที่มีเงื่อนไขมากกว่า 1 เงื่อนไข โดยมีรูปแบบคำสั่งดังนี้

```
if (เงื่อนไข 1) {  
    คำสั่งที่ต้องกระทำเมื่อเงื่อนไขเป็นจริง  
}
```

```
else if (เงื่อนไข 2) {  
    คำสั่งที่ต้องกระทำเมื่อเงื่อนไขเป็นจริง  
}  
.  
.  
else if (เงื่อนไข n) {  
    คำสั่งที่ต้องกระทำเมื่อเงื่อนไขเป็นจริง  
}  
else {  
    คำสั่งที่ต้องกระทำเมื่อเงื่อนไขเป็นจริง  
}
```

ซึ่งจากรูปแบบการใช้คำสั่ง if-else แบบนี้ มีความหมายเหมือนกับประโยคที่ว่า ถ้าเงื่อนไขที่ 1 เป็นจริง ให้ทำงานที่ 1 ไม่เช่นนั้นให้ตรวจสอบเงื่อนไขที่ 2 และถ้าเงื่อนไขที่ 2 เป็นจริง ให้ทำงานที่ 2 ไม่เช่นนั้น ตรวจสอบเงื่อนไขที่ 3 และถ้าเงื่อนไขที่ 3 เป็นจริง ให้ทำงานที่ 3 ไม่เช่นนั้นให้ตรวจสอบเงื่อนไขที่ n และ เลือกทำงานตามคำสั่งเงื่อนไขที่ n

ซึ่งจะเห็นได้ว่า เราสามารถที่จะทำการเพิ่มเติมเงื่อนไขให้กับโปรแกรมเพื่อเป็นทางเลือกในการเลือกทำงานตามคำสั่งต่างๆตามความเหมาะสมได้หลายทางเลือก

11.6 คำสั่ง delay();

คำสั่งนี้ใช้สำหรับการหน่วงเวลาหยุดรอก่อนที่จะไปทำงานคำสั่งบรรทัดถัดไป โดยมีค่าเป็น mS (milli-Second) ซึ่งสามารถกำหนดค่าการหน่วงเวลาที่เป็นค่าตัวเลขจำนวนเต็มได้ระหว่าง 1 ถึง 4,294,967,295 mS โดยที่ ค่าเวลาของ 1 mS จะมีค่าเท่ากับ 1/1000 วินาที โดยมีรูปแบบคำสั่งดังนี้

```
delay();
```

การทดลองที่ 1 หุ่นยนต์เดินตามเส้น 90 องศา

1.1 นำสาย USB Serial เชื่อมต่อกับบอร์ด Arduino UNO R3 บนหุ่นยนต์

1.2 เปิดโปรแกรม Arduino IDE แล้วกดปุ่ม Ctrl + N เพื่อสร้างไฟล์งานใหม่โดย Save ไฟล์

เป็นชื่อ 09_Exp_01

1.3 เขียนโปรแกรมดังตัวอย่างโค้ดต่อไปนี้

```
1 //กำหนดชื่อขาที่ต่อใช้งานร่วมกับเซนเซอร์
2 #define out1 9
3 #define out2 10
4 #define out3 11
5 #define out4 12
6 #define out5 13
7
8 ////กำหนดชื่อขาที่ต่อใช้งานร่วมกับบอร์ดไดรฟ์มอเตอร์
9 #define MotorLeftA 1
10 #define MotorLeftB 2
11 #define SpeedMotorLeft 3
12 #define MotorRightA 4
13 #define MotorRightB 5
14 #define SpeedMotorRight 6
15
16 //สร้างตัวแปรเพื่อเก็บค่าที่อ่านได้จากเซนเซอร์
17 byte val_out1;
18 byte val_out2;
19 byte val_out3;
20 byte val_out4;
21 byte val_out5;
22
23 void setup() {
24 //กำหนดสถานะของขาที่ต่อใช้งานกับบอร์ดไดรฟ์มอเตอร์
25 pinMode(MotorLeftA, OUTPUT);
26 pinMode(MotorLeftB, OUTPUT);
27 pinMode(SpeedMotorLeft, OUTPUT);
28 pinMode(MotorRightA, OUTPUT);
29 pinMode(MotorRightB, OUTPUT);
30 pinMode(SpeedMotorRight, OUTPUT);
31
32 //กำหนดสถานะของขาที่รับค่าจากเซนเซอร์
33 pinMode(out1, INPUT);
34 pinMode(out2, INPUT);
35 pinMode(out3, INPUT);
36 pinMode(out4, INPUT);
```

```
37 pinMode(out5, INPUT);
38 }
39
40 void loop() {
41 //เขียนคำสั่งเพื่ออ่านค่าจากเซนเซอร์มาเก็บไว้ในตัวแปร
42 val_out1 = digitalRead(out1);
43 val_out2 = digitalRead(out2);
44 val_out3 = digitalRead(out3);
45 val_out4 = digitalRead(out4);
46 val_out5 = digitalRead(out5);
47
48 //สร้างเงื่อนไขเพื่อตรวจสอบสถานะของเซนเซอร์เพื่อควบคุมทิศทางของหุ่นยนต์
49 // เดินหน้าเมื่อเจอเส้นตรง
50 if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == LOW
51     && val_out4 == HIGH && val_out5 == HIGH) {
52     digitalWrite(MotorLeftA, HIGH);
53     digitalWrite(MotorLeftB, LOW);
54     analogWrite(SpeedMotorLeft, 155);
55     digitalWrite(MotorRightA, HIGH);
56     digitalWrite(MotorRightB, LOW);
57     analogWrite(SpeedMotorRight, 155);
58 }
59 // เลี้ยวซ้ายเมื่อเจอโค้งซ้าย
60 else if (val_out1 == HIGH && val_out2 == LOW && val_out3 == HIGH
61     && val_out4 == HIGH && val_out5 == HIGH) {
62     digitalWrite(MotorLeftA, LOW);
63     digitalWrite(MotorLeftB, HIGH);
64     analogWrite(SpeedMotorLeft, 155);
65     digitalWrite(MotorRightA, HIGH);
66     digitalWrite(MotorRightB, LOW);
67     analogWrite(SpeedMotorRight, 155);
68 }
69 // หมุนซ้ายเมื่อเจอโค้งหักเลี้ยวซ้าย
70 else if (val_out1 == LOW && val_out2 == LOW && val_out3 == LOW
71     && val_out4 == HIGH && val_out5 == HIGH) {
72     digitalWrite(MotorLeftA, LOW);
73     digitalWrite(MotorLeftB, HIGH);
74     analogWrite(SpeedMotorLeft, 155);
75     digitalWrite(MotorRightA, HIGH);
76     digitalWrite(MotorRightB, LOW);
77     analogWrite(SpeedMotorRight, 155);
78     delay(500);
79 }
80 // เลี้ยวขวาเมื่อเจอโค้งขวา
```

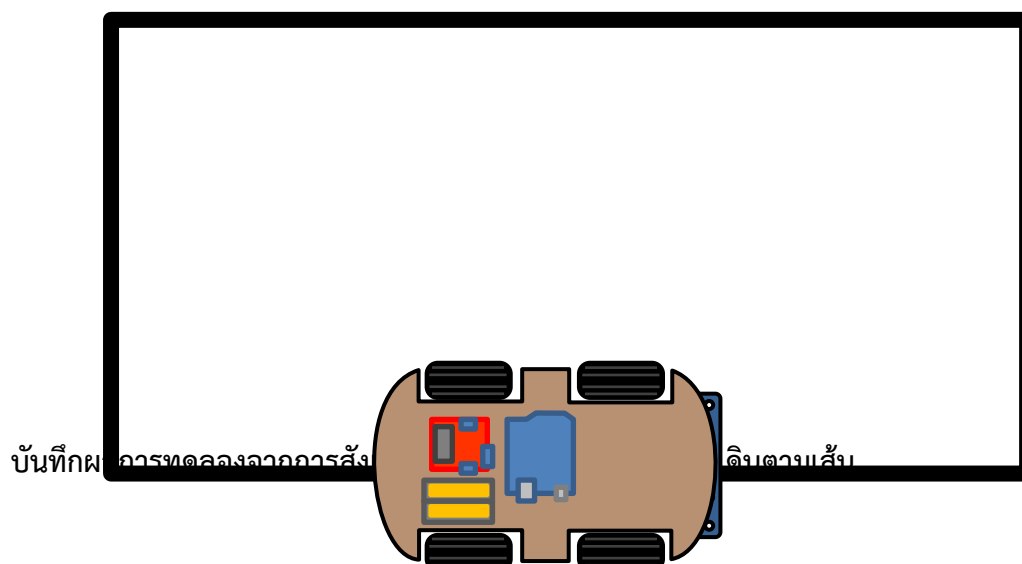
```

81  else if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == HIGH
82         && val_out4 == LOW && val_out5 == HIGH) {
83      digitalWrite(MotorLeftA, HIGH);
84      digitalWrite(MotorLeftB, LOW);
85      analogWrite(SpeedMotorLeft, 155);
86      digitalWrite(MotorRightA, LOW);
87      digitalWrite(MotorRightB, HIGH);
88      analogWrite(SpeedMotorRight, 155);
89  }
90  // หมุนขวาเมื่อเจอโค้งหักเลี้ยวขวา
91  else if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == LOW
92         && val_out4 == LOW && val_out5 == LOW) {
93      digitalWrite(MotorLeftA, HIGH);
94      digitalWrite(MotorLeftB, LOW);
95      analogWrite(SpeedMotorLeft, 155);
96      digitalWrite(MotorRightA, LOW);
97      digitalWrite(MotorRightB, HIGH);
98      analogWrite(SpeedMotorRight, 155);
99      delay(500);
100 }
101 //หยุดเดินเมื่อไม่เจอเส้น
102 else {
103     digitalWrite(MotorLeftA, HIGH);
104     digitalWrite(MotorLeftB, HIGH);
105     analogWrite(SpeedMotorLeft, 0);
106     digitalWrite(MotorRightA, HIGH);
107     digitalWrite(MotorRightB, HIGH);
108     analogWrite(SpeedMotorRight, 0);
109 }
110 }

```

1.4 เมื่อเขียนโปรแกรมเสร็จสิ้น ทำการตรวจสอบความถูกต้องโดยกดปุ่ม Verify และทำการ Upload

1.5 ทดลองนำหุ่นยนต์ไปทดสอบบนสนามที่มีลักษณะการตีเส้นสีดำตามรูปแบบดังนี้



.....

.....

.....

.....

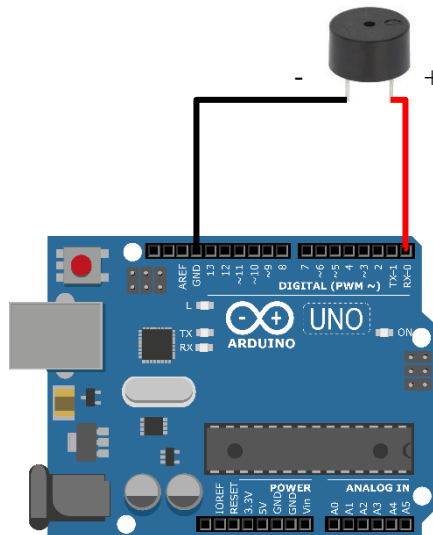
.....

.....

แนบไฟล์หรือลิงค์วิดีโอผลการทดลองที่ 1

การทดลองที่ 2 หุ่นยนต์นับเส้นทางแยก

2.1 จงต่ออุปกรณ์ลำโพง Buzzer ร่วมกับบอร์ด Arduino บนหุ่นยนต์ดังรูปที่ 9.1



รูปที่ 11.1 การต่อลำโพง Buzzer ร่วมกับบอร์ด Arduino

2.2 นำสาย USB Serial เชื่อมต่อกับบอร์ด Arduino UNO R3 บนหุ่นยนต์

วิชาหุ่นยนต์เบื้องต้น รหัส 20105-2121

ภาคเรียนที่ 2 ประจำปีการศึกษา 2565

แผนกวิชาช่างอิเล็กทรอนิกส์ วิทยาลัยเทคนิคชลบุรี

ครูผู้สอน นายธีรศักดิ์ เชียงหลง

2.3 เปิดโปรแกรม Arduino IDE แล้วกดปุ่ม Ctrl + N เพื่อสร้างไฟล์งานใหม่โดย Save ไฟล์
เป็นชื่อ 09_Exp_02

2.3 เขียนโปรแกรมดังตัวอย่างต่อไปนี้


```
1 //กำหนดชื่อขาที่ต่อใช้งานร่วมกับเซนเซอร์
2 #define out1 9
3 #define out2 10
4 #define out3 11
5 #define out4 12
6 #define out5 13
7
8 ////กำหนดชื่อขาที่ต่อใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์
9 #define MotorLeftA 1
10 #define MotorLeftB 2
11 #define SpeedMotorLeft 3
12 #define MotorRightA 4
13 #define MotorRightB 5
14 #define SpeedMotorRight 6
15
16 #define buzzer 0
17 byte count = 0;
18
19 //สร้างตัวแปรเพื่อเก็บค่าที่อ่านได้จากเซนเซอร์
20 byte val_out1;
21 byte val_out2;
22 byte val_out3;
23 byte val_out4;
24 byte val_out5;
25
26 void setup() {
27 //กำหนดสถานะของขาที่ต่อใช้งานกับบอร์ดไมโครคอนโทรลเลอร์
28 pinMode(MotorLeftA, OUTPUT);
29 pinMode(MotorLeftB, OUTPUT);
30 pinMode(SpeedMotorLeft, OUTPUT);
31 pinMode(MotorRightA, OUTPUT);
32 pinMode(MotorRightB, OUTPUT);
33 pinMode(SpeedMotorRight, OUTPUT);
34
35 //กำหนดสถานะของขาที่รับค่าจากเซนเซอร์
36 pinMode(out1, INPUT);
37 pinMode(out2, INPUT);
38 pinMode(out3, INPUT);
39 pinMode(out4, INPUT);
40 pinMode(out5, INPUT);
41
42 pinMode(buzzer, OUTPUT);
43 }
44
45 void loop() {
46 //เขียนคำสั่งเพื่ออ่านค่าจากเซนเซอร์มาเก็บไว้ในตัวแปร
47 val_out1 = digitalRead(out1);
```

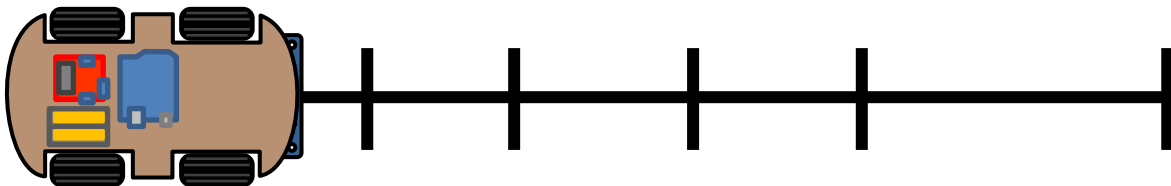
```
48 val_out2 = digitalRead(out2);
49 val_out3 = digitalRead(out3);
50 val_out4 = digitalRead(out4);
51 val_out5 = digitalRead(out5);
52
53 //สร้างเงื่อนไขเพื่อตรวจสอบสถานะของเซ็นเซอร์เพื่อควบคุมทิศทางของหุ่นยนต์
54 // เดินหน้าเมื่อเจอเส้นตรง
55 if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == LOW
56     && val_out4 == HIGH && val_out5 == HIGH) {
57     digitalWrite(MotorLeftA, HIGH);
58     digitalWrite(MotorLeftB, LOW);
59     analogWrite(SpeedMotorLeft, 155);
60     digitalWrite(MotorRightA, HIGH);
61     digitalWrite(MotorRightB, LOW);
62     analogWrite(SpeedMotorRight, 155);
63 }
64 // เลี้ยวซ้ายเมื่อเจอโค้งซ้าย
65 else if (val_out1 == HIGH && val_out2 == LOW && val_out3 == HIGH
66         && val_out4 == HIGH && val_out5 == HIGH) {
67     digitalWrite(MotorLeftA, LOW);
68     digitalWrite(MotorLeftB, HIGH);
69     analogWrite(SpeedMotorLeft, 155);
70     digitalWrite(MotorRightA, HIGH);
71     digitalWrite(MotorRightB, LOW);
72     analogWrite(SpeedMotorRight, 155);
73 }
74 // เลี้ยวขวาเมื่อเจอโค้งขวา
75 else if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == HIGH
76         && val_out4 == LOW && val_out5 == HIGH) {
77     digitalWrite(MotorLeftA, HIGH);
78     digitalWrite(MotorLeftB, LOW);
79     analogWrite(SpeedMotorLeft, 155);
80     digitalWrite(MotorRightA, LOW);
81     digitalWrite(MotorRightB, HIGH);
82     analogWrite(SpeedMotorRight, 155);
83 }
84 // เพิ่มค่าในตัวแปรแล้วส่งสัญญาณไปที่ Buzzer เมื่อเจอทิวแยก
85 else if (val_out1 == LOW && val_out2 == LOW && val_out3 == LOW
86         && val_out4 == LOW && val_out5 == LOW) {
87     digitalWrite(buzzer, HIGH);
88     digitalWrite(MotorLeftA, HIGH);
89     digitalWrite(MotorLeftB, LOW);
90     analogWrite(SpeedMotorLeft, 155);
91     digitalWrite(MotorRightA, HIGH);
92     digitalWrite(MotorRightB, LOW);
93     analogWrite(SpeedMotorRight, 155);
94     delay(100);
```

```

95     digitalWrite(buzzer, LOW);
96     count++;
97   }
98   //หยุดเดินเมื่อไม่เจอเส้น
99   else {
100    digitalWrite(MotorLeftA, HIGH);
101    digitalWrite(MotorLeftB, HIGH);
102    analogWrite(SpeedMotorLeft, 0);
103    digitalWrite(MotorRightA, HIGH);
104    digitalWrite(MotorRightB, HIGH);
105    analogWrite(SpeedMotorRight, 0);
106  }
107 }
    
```

2.4 เมื่อเขียนโปรแกรมเสร็จสิ้น ทำการตรวจสอบความถูกต้องโดยกดปุ่ม Verify และทำการ Upload

2.5 ทดลองนำหุ่นยนต์ไปทดสอบบนสนามที่มีลักษณะการตีเส้นสีดำตามรูปแบบดังนี้



บันทึกผลการทดลองจากการสังเกตการณ์เคลื่อนที่ของหุ่นยนต์เดินตามเส้น

.....

.....

.....

.....

.....

.....

แนบไฟล์หรือลิงค์วิดีโอผลการทดลองที่ 2

คำถามท้ายการทดลอง

