ใบงานที่ 11

เรื่อง หุ่นยนต์เคลื่อนตามเส้น 90 องศาและนับเส้นทางแยก

วัตถุประสงค์ (เพื่อให้นักเรียนษาสามารถ)

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

<u>ทฤษฎีเบื้องต้น</u>

11.1 คำสั่ง pinMode

คือคำสั่งที่มีไว้สำหรับกำหนดการทำงานของ pin ที่ต้องการใช้งาน ให้ทำงานเป็นแบบ OUTPUT หรือ INPUT การทำงานเป็น output เช่น การสั่งจ่ายแรงดัน 5 volt. การทำงานเป็น input เช่น การสั่งให้อ่านค่าสถานะสวิตช์ไฟ 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 เป็น จริง 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();
```

<u>ขั้นตอนการทดลอง</u>

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 ไฟล์เป็นชื่อ **09_Exp_01**

1.3 เขียนโปรแกรมดังตัวอย่างโค้ดต่อไปนี้

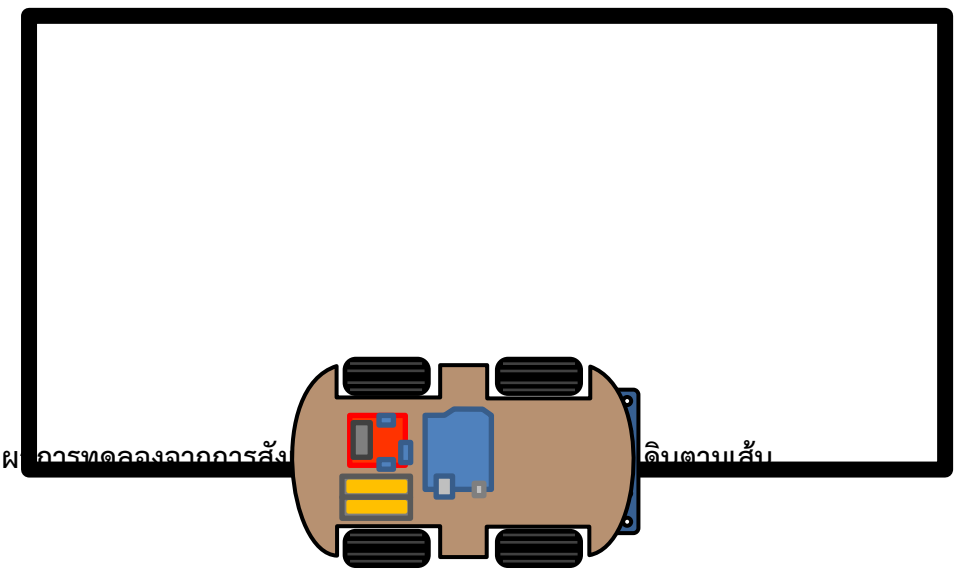
```
//กำหนดชื่อขาที่ต่อใช้งานร่วมกับเซนเซอร์
#define out1 9
#define out2 10
#define out3 11
#define out4 12
#define out5 13

////กำหนดชื่อขาที่ต่อใช้งานร่วมกับบอร์ดไดร์มอเตอร์
#define MotorLeftA 1
#define MotorLeftB 2
#define SpeedMotorLeft 3
#define MotorRightA 4
#define MotorRightB 5
#define SpeedMotorRight 6

//สร้างตัวแปรเพื่อเก็บค่าที่อ่านได้จากเซนเซอร์
byte val_out1;
byte val_out2;
byte val_out3;
byte val_out4;
byte val_out5;

void setup() {
  //กำหนดสถานะของขาที่ต่อใช้งานกับบอร์ดไดร์มอเตอร์
  pinMode(MotorLeftA, OUTPUT);
  pinMode(MotorLeftB, OUTPUT);
  pinMode(SpeedMotorLeft, OUTPUT);
  pinMode(MotorRightA, OUTPUT);
  pinMode(MotorRightB, OUTPUT);
  pinMode(SpeedMotorRight, OUTPUT);

  //กำหนดสถานะของขาที่รับค่าจากเซนเซอร์
  pinMode(out1, INPUT);
  pinMode(out2, INPUT);
  pinMode(out3, INPUT);
  pinMode(out4, INPUT);
```

```
  pinMode(out5, INPUT);
}

void loop() {
  //เขียนคำสั่งเพื่ออ่านค่าจากเซนเซอร์มาเก็บไว้ในตัวแปร
  val_out1 = digitalRead(out1);
  val_out2 = digitalRead(out2);
  val_out3 = digitalRead(out3);
  val_out4 = digitalRead(out4);
  val_out5 = digitalRead(out5);

  //สร้างเงื่อนไขเพื่อตรวจสอบสถานะของเซนเซอร์เพื่อควบคุมทิศทางของหุ่นยนต์
  // เดินหน้าเมื่อเจอริ้เส้นตรง
  if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == LOW
      && val_out4 == HIGH && val_out5 == HIGH) {
    digitalWrite(MotorLeftA, HIGH);
    digitalWrite(MotorLeftB, LOW);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, HIGH);
    digitalWrite(MotorRightB, LOW);
    analogWrite(SpeedMotorRight, 155);
  }
  // เลี้ยวซ้ายเมื่อเจอริ้งซ้าย
  else if (val_out1 == HIGH && val_out2 == LOW && val_out3 == HIGH
           && val_out4 == HIGH && val_out5 == HIGH) {
    digitalWrite(MotorLeftA, LOW);
    digitalWrite(MotorLeftB, HIGH);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, HIGH);
    digitalWrite(MotorRightB, LOW);
    analogWrite(SpeedMotorRight, 155);
  }
  // หมุนซ้ายเมื่อเจอริ้งหักเลี้ยวซ้าย
  else if (val_out1 == LOW && val_out2 == LOW && val_out3 == LOW
           && val_out4 == HIGH && val_out5 == HIGH) {
    digitalWrite(MotorLeftA, LOW);
    digitalWrite(MotorLeftB, HIGH);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, HIGH);
    digitalWrite(MotorRightB, LOW);
    analogWrite(SpeedMotorRight, 155);
    delay(500);
  }
  // เลี้ยวขวาเมื่อเจอริ้งขวา
```

```
  else if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == HIGH
           && val_out4 == LOW && val_out5 == HIGH) {
    digitalWrite(MotorLeftA, HIGH);
    digitalWrite(MotorLeftB, LOW);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, LOW);
    digitalWrite(MotorRightB, HIGH);
    analogWrite(SpeedMotorRight, 155);
  }
  // หมุนขวาเมื่อเจอโค้งหักเลี้ยวขวา
  else if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == LOW
           && val_out4 == LOW && val_out5 == LOW) {
    digitalWrite(MotorLeftA, HIGH);
    digitalWrite(MotorLeftB, LOW);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, LOW);
    digitalWrite(MotorRightB, HIGH);
    analogWrite(SpeedMotorRight, 155);
    delay(500);
  }
  //หยุดเดินเมื่อไม่เจอเส้น
  else {
    digitalWrite(MotorLeftA, HIGH);
    digitalWrite(MotorLeftB, HIGH);
    analogWrite(SpeedMotorLeft, 0);
    digitalWrite(MotorRightA, HIGH);
    digitalWrite(MotorRightB, HIGH);
    analogWrite(SpeedMotorRight, 0);
  }
}
```

1.4 เมื่อเขียนโปรแกรมเสร็จสิ้น ทำการตรวจสอบความถูกต้องโดยกดปุ่ม Verify และทำการ Upload

1.5 ทดลองนำหุ่นยนต์ไปทดสอบบนสนามที่มีลักษณะการตีเส้นสีดำตามรูปแบบดังนี้



บันทึกผลการทดลองจากการสังเกตการเคลื่อนที่ของหุ่นยนต์ตามเส้น

.....

.....

.....

.....

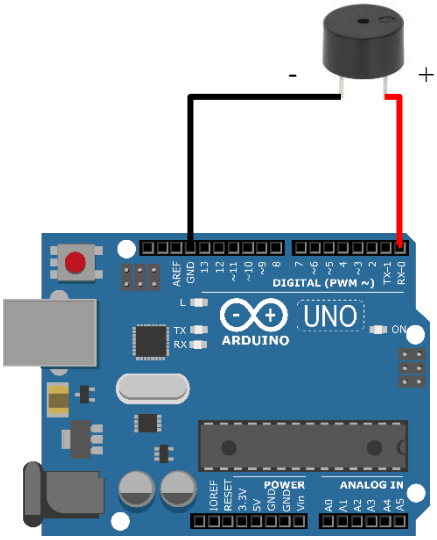
.....

.....

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

2.3 เปิดโปรแกรม Arduino IDE แล้วกดปุ่ม Ctrl + N เพื่อสร้างไฟล์งานใหม่โดย Save ไฟล์เป็นชื่อ **09_Exp_02**

2.3 เขียนโปรแกรมดังตัวอย่างโค้ดต่อไปนี้

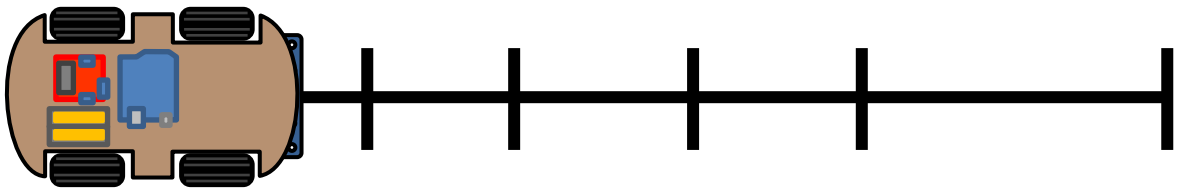
```
//กำหนดชื่อขาที่ต่อใช้งานร่วมกับเซนเซอร์
#define out1 9
#define out2 10
#define out3 11
#define out4 12
#define out5 13

////กำหนดชื่อขาที่ต่อใช้งานร่วมกับบอร์ดไดร์มอเตอร์
#define MotorLeftA 1
#define MotorLeftB 2
#define SpeedMotorLeft 3
#define MotorRightA 4
#define MotorRightB 5
#define SpeedMotorRight 6

#define buzzer 0
byte count = 0;

//สร้างตัวแปรเพื่อเก็บค่าที่อ่านได้จากเซนเซอร์
byte val_out1;
byte val_out2;
byte val_out3;
byte val_out4;
byte val_out5;

void setup() {
  //กำหนดสถานะของขาที่ต่อใช้งานกับบอร์ดไดร์มอเตอร์
  pinMode(MotorLeftA, OUTPUT);
  pinMode(MotorLeftB, OUTPUT);
  pinMode(SpeedMotorLeft, OUTPUT);
  pinMode(MotorRightA, OUTPUT);
  pinMode(MotorRightB, OUTPUT);
  pinMode(SpeedMotorRight, OUTPUT);

  //กำหนดสถานะของขาที่รับค่าจากเซนเซอร์
  pinMode(out1, INPUT);
  pinMode(out2, INPUT);
  pinMode(out3, INPUT);
  pinMode(out4, INPUT);
  pinMode(out5, INPUT);

  pinMode(buzzer, OUTPUT);
}

void loop() {
  //เขียนคำสั่งเพื่ออ่านค่าจากเซนเซอร์มาเก็บไว้ในตัวแปร
  val_out1 = digitalRead(out1);
```

```
  val_out2 = digitalRead(out2);
  val_out3 = digitalRead(out3);
  val_out4 = digitalRead(out4);
  val_out5 = digitalRead(out5);

  //สร้างเงื่อนไขเพื่อตรวจสอบสถานะของเซนเซอร์เพื่อควบคุมทิศทางของหุ่นยนต์
  // เดินหน้าเมื่อเจอเส้นตรง
  if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == LOW
      && val_out4 == HIGH && val_out5 == HIGH) {
    digitalWrite(MotorLeftA, HIGH);
    digitalWrite(MotorLeftB, LOW);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, HIGH);
    digitalWrite(MotorRightB, LOW);
    analogWrite(SpeedMotorRight, 155);
  }
  // เลี้ยวซ้ายเมื่อเจอโค้งซ้าย
  else if (val_out1 == HIGH && val_out2 == LOW && val_out3 == HIGH
           && val_out4 == HIGH && val_out5 == HIGH) {
    digitalWrite(MotorLeftA, LOW);
    digitalWrite(MotorLeftB, HIGH);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, HIGH);
    digitalWrite(MotorRightB, LOW);
    analogWrite(SpeedMotorRight, 155);
  }
  // เลี้ยวขวาเมื่อเจอโค้งขวา
  else if (val_out1 == HIGH && val_out2 == HIGH && val_out3 == HIGH
           && val_out4 == LOW && val_out5 == HIGH) {
    digitalWrite(MotorLeftA, HIGH);
    digitalWrite(MotorLeftB, LOW);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, LOW);
    digitalWrite(MotorRightB, HIGH);
    analogWrite(SpeedMotorRight, 155);
  }
  // เพิ่มค่าในตัวแปรแล้วส่งสัญญาณไปที่ Buzzer เมื่อเจอทางแยก
  else if (val_out1 == LOW && val_out2 == LOW && val_out3 == LOW
           && val_out4 == LOW && val_out5 == LOW) {
    digitalWrite(buzzer,HIGH);
    digitalWrite(MotorLeftA, HIGH);
    digitalWrite(MotorLeftB, LOW);
    analogWrite(SpeedMotorLeft, 155);
    digitalWrite(MotorRightA, HIGH);
    digitalWrite(MotorRightB, LOW);
    analogWrite(SpeedMotorRight, 155);
    delay(100);
```

```
    digitalWrite(buzzer, LOW);
    count ++;
  }
  //หยุดเดินเมื่อไม่เจอเส้น
  else {
    digitalWrite(MotorLeftA, HIGH);
    digitalWrite(MotorLeftB, HIGH);
    analogWrite(SpeedMotorLeft, 0);
    digitalWrite(MotorRightA, HIGH);
    digitalWrite(MotorRightB, HIGH);
    analogWrite(SpeedMotorRight, 0);
  }
}
```

2.4 เมื่อเขียนโปรแกรมเสร็จสิ้น ทำการตรวจสอบความถูกต้องโดยกดปุ่ม Verify และทำการ Upload

2.5 ทดลองนำหุ่นยนต์ไปทดสอบบนสนามที่มีลักษณะการตีเส้นสีดำตามรูปแบบดังนี้



บันทึกผลการทดลองจากการสังเกตการณ์เคลื่อนที่ของหุ่นยนต์เดินตามเส้น

.....

.....

.....

.....

.....

.....

แนบไฟล์หรือลิงค์วิดีโอผลการทดลองที่ 2

คำถามท้ายการทดลอง

1. จากการทดลองเขียนโปรแกรมให้หุ่นยนต์เดินตามเส้นตรงและเส้นโค้งในใบงานการทดลองที่ 10 เมื่อเปรียบเทียบกับการทดลองที่ 11 หุ่นยนต์เดินตามเส้น 90 องศา ในใบงานการทดลองนี้ โค้ดของโปรแกรมมีความแตกต่างหรือมีการเพิ่มเติมอย่าไงร

.....

.....

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....