ใบงานที่ 9
เรื่อง การใช้โปรแกรมควบคุมหุ่นยนต์บังคับด้วยมือ

วัตถุประสงค์ (เพื่อให้นักเรียนษาสามารถ)

1. ต่อใช้งานโมดูลจอยสวิตช์ร่วมกับ Arduino ได้ถูกต้อง
2. เขียนโปรแกรมอ่านค่าจากโมดูลจอยสวิตช์ได้ถูกต้อง
3. เขียนโปรแกรมควบคุมหุ่นยนต์บังคับด้วยมือได้ถูกต้อง
4. บันทึกผลการทดลองได้ถูกต้อง
5. สรุปผลการทดลองได้ตามวัตถุประสงค์

เครื่องมือและอุปกรณ์

1. คอมพิวเตอร์	จำนวน 1 เครื่อง
2. บอร์ด Arduino UNO R3	จำนวน 1 บอร์ด
3. สาย USB Serial	จำนวน 1 สาย

<u>ทฤษฎีเบื้องต้น</u>

9.1 ลักษณะการทำงานของสวิตช์

บอร์ด Arduino มี Digital I/O PINs สำหรับใช้งาน ซึ่งคำว่า I/O หมายถึง Input และ Output นั่นหมายความว่า นอกจาก Pins เหล่านี้จะสามารถเป็น Output เพื่อไปควบคุมอุปกรณ์ต่างๆได้แล้ว มันยังสามารถใช้เป็นตัว Input หรือตัวรับสัญญาณทางไฟฟ้าต่างๆ เพื่อใช้ควบคุมการทำงานของวงจรได้อีกด้วย

การที่เราใช้ Digital Pins ของ Arduino เป็นตัวรับค่าสัญญาณ คือการที่เราให้ Pins นั้น ๆ เป็นตัว Input โดยค่าที่ Arduino อ่านได้จะมีอยู่ 2 ค่า คือ HIGH และ LOW ขึ้นอยู่กับ Pin นั้น มีสัญญาณหรือกระแสไฟฟ้าเข้ามาหรือไม่

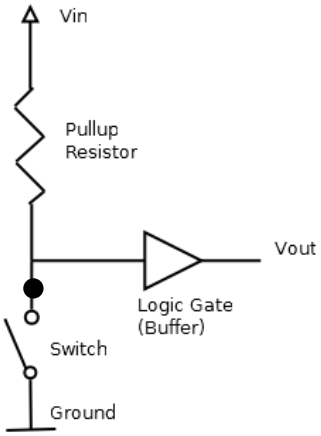
สถานะที่กำหนดให้ขาของอุปกรณ์อิเล็กทรอนิกส์ รอรับอินพุต (INPUT) ขาพอร์ตจะเป็น High Impedance คือมีความต้านทานสูงมากต่ออยู่ ทำให้ขาพอร์ตนั้นเสมือนถูกปล่อยลอย ค่าอินพุทที่อ่านกลับมาได้ไม่แน่นอน

ดังนั้นในงานขาพอร์ตอินพุท วงจรของสวิตช์จำเป็นมากที่ต้องมี Pull-up Resistor หรือ Pull-down Resistor เพื่อที่จะกำหนดสภาวะดิจิตอลที่แน่นอนให้กับอุปกรณ์อิเล็กทรอนิกส์ ตามปกติตัวต้านทานที่ใช้ในวงจร Pull-up หรือ Pull-down จะใช้ประมาณ 5k Ohm - 20k Ohm

9.2 การต่อใช้งานสวิตช์

9.2.1 วงจร Pull-up Resister

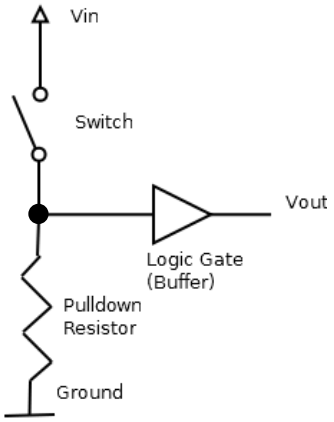
Pull-up Resistor คือการนำตัวต้านทานต่อเข้ากับ Vcc (+5V) เพื่อให้แรงดันอยู่คงที่ ทำให้อยู่ในสถานะ “HIGH” หรือ “1” ตลอดเวลา และเมื่อกดสวิตช์กระแสไฟฟ้าจะไหลลง Ground ทันที ซึ่งทำให้สถานะเป็นลอจิก “LOW” หรือ “0” และ การทำงานลักษณะนี้จะเรียกว่า Active Low เพราะว่าจะเขียนโปรแกรมที่ทำงาน เมื่อลอจิกเป็น “LOW” ส่วนใหญ่ เราจะเห็นต่อสวิตช์นิยมใช้แบบ Pull-up มากกว่า



รูปที่ 9.1 วงจร Pull-up Resister

9.2.2 วงจร Pull-down Resister

Pull-down Resistor โดยใน Pull-down จะมีลักษณะคล้ายกับ Pull-up Resistor แตกต่างตรงที่ สภาวะปกติของ Pull-down จะเป็นลอจิก “LOW” หรือ “0” เมื่อมีการกดปุ่มกระแสไฟจะไหลเข้าขาอินพุต ทำให้ลอจิกเป็น “HIGH” หรือ “1” ได้ การทำงานในลักษณะนี้จะเรียกว่า Active High



รูปที่ 9.2 วงจร Pull-down Resister

9.3 คำสั่งที่ใช้สำหรับอ่านค่าจากสวิตช์

เนื่องจากสวิตช์เป็นอุปกรณ์ที่ส่งข้อมูลมายังไมโครคอนโทรลเลอร์ ดังนั้นในการอ่านค่าจากสวิตช์ จะต้องกำหนดสถานะของขาดิจิตอลบนบอร์ด Arduino ที่ต่ออยู่กับสวิตช์และใช้คำสั่งเพื่ออ่านค่าที่ได้ โดยสามารถใช้คำสั่งได้ดังนี้

```
pinMode(PIN,MODE);
```

คำสั่ง pinMode เป็นคำสั่งที่ใช้ในการกำหนดสถานะขาของ Arduino โดยจะมีอยู่ 2 โหมด คือ Input และ Output ซึ่งโหมด Input ก็คือกำหนดให้ค่านั้นใช้สำหรับการรับค่า ส่วน Output คือกำหนดให้ขานั้นใช้ในการส่งค่า จากตัวอย่างเช่น pinMode(2,INPUT);

```
digitalRead(PIN);
```

คำสั่งนี้จะใช้ในการอ่านค่า Digital จากขาที่เราต้องการ เช่น หากต้องการที่จะอ่านค่าขา D2 จะเขียนได้เป็น digitalRead(2);

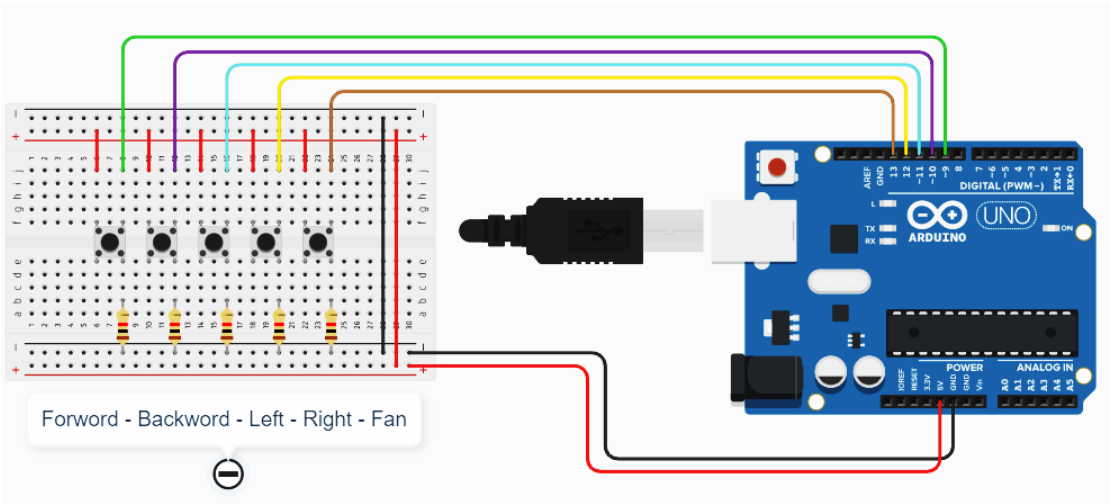
โดยค่าที่ได้จากคำสั่งนี้ ก็จะมีแค่ 0 หรือ 1 เท่านั้น การนำไปใช้งานต้องกำหนดตัวแปรสำหรับเก็บค่า ชื่อ buttonState ก็จะเขียนได้ในลักษณะนี้

```
int buttonState;
buttonState = digitalRead(2);
```

ขั้นตอนการทดลอง

การทดลองที่ 1

1.1 นำโมดูลจอยหรือสวิตช์มาต่อใช้งานร่วมกับบอร์ด Arduino ของหุ่นยนต์ ตามตัวอย่างดังรูปที่ 9.3



รูปที่ 9.3 การต่อใช้งานสวิตช์ร่วมกับบอร์ด Arduino

1.2 นำสาย USB Serial เชื่อมต่อกับบอร์ด Arduino UNO R3 บนหุ่นยนต์

1.3 เปิดโปรแกรม Arduino IDE แล้วกดปุ่ม Ctrl + N เพื่อสร้างไฟล์งานใหม่โดย Save ไฟล์เป็นชื่อ **07_Exp_01**

1.4 เขียนโปรแกรมดังตัวอย่างโค้ดต่อไปนี้

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

//ขาที่ต่อใช้งานร่วมกับสวิตช์
#define SWforword  9
#define SWbackword 10
#define SWleft     11
#define SWright    12
#define SWfan      13

//สร้างตัวแปรเพื่อเก็บค่าที่อ่านได้จากสวิตช์
byte SWF, SWB, SWL, SWR, SWf;

void setup() {

```

```
  Serial.begin(9600);

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

  //กำหนดขาที่ 8 9 10 และ 11 ให้เป็นโหมด INTPUT
  pinMode(SWforword, INPUT);
  pinMode(SWbackword, INPUT);
  pinMode(SWleft, INPUT);
  pinMode(SWright, INPUT);
  pinMode(SWfan, INPUT);
}

void loop() {

  //อ่านค่าจากสวิตช์มาเก็บไว้ในตัวแปรเพื่อนำไปใช้
  SWF = digitalRead(SWforword);
  SWB = digitalRead(SWbackword);
  SWL = digitalRead(SWleft);
  SWR = digitalRead(SWright);
  SWf = digitalRead(SWfan);

  //สร้างเงื่อนไขเพื่อตรวจสอบการกดสวิตช์
  if (SWF == HIGH) {                    //เดินหน้า
    Serial.println("Forword");
  }
  else if (SWB == HIGH) {               //เดินถอยหลัง
    Serial.println("Backword");
  }
  else if (SWL == HIGH) {                //หมุนซ้าย
    Serial.println("Turn Left");
  }
  else if (SWR == HIGH) {                //หมุนขวา
    Serial.println("Turn Right");
  }
  else if (SWf == HIGH) {                //เปิดมอเตอร์พัดลม
    Serial.println("Fan Boost");
  }
  else {                                 //หยุด
    Serial.println("Stop");
  }
}
```

1.5 เมื่อเขียนโปรแกรมเสร็จสิ้น ทำการตรวจสอบความถูกต้องโดยกดปุ่ม Verify และทำการ Upload

1.8 ทดลองกดสวิตช์แล้วสังเกตหน้าจอ Serial Monitor

บันทึกผลการทดลองจากการสังเกตหน้าจอ Serial Monitor

.....

.....

.....

.....

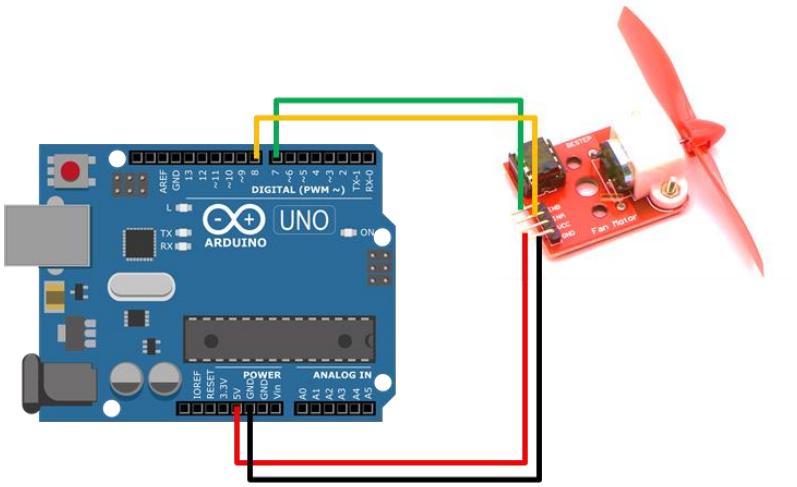
.....

.....

แนบไฟล์หรือลิงค์วิดีโอผลการทดลองที่ 1

<u>การทดลองที่ 2</u>

2.1 นำโมดูลมอเตอร์พัดลมมาต่อใช้งานร่วมกับบอร์ด Arduino ของหุ่นยนต์ ตามตัวอย่างดังรูปที่ 8.4



รูปที่ 9.4 การต่อใช้งานโมดูลมอเตอร์พัดลมร่วมกับบอร์ด Arduino

2.2 นำสาย USB Serial เชื่อมต่อกับบอร์ด Arduino UNO R3 บนหุ่นยนต์

2.3 เปิดโปรแกรม Arduino IDE แล้วกดปุ่ม Ctrl + N เพื่อสร้างไฟล์งานใหม่โดย Save ไฟล์เป็นชื่อ **07_Exp_02**

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

//ขาที่ต่อใช้งานร่วมกับสวิตช์
#define SWforword  9
#define SWbackword 10
#define SWleft     11
#define SWright    12
#define SWfan      13

//สร้างตัวแปรเพื่อเก็บค่าที่อ่านได้จากสวิตช์
byte SWF, SWB, SWL, SWf;

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

  //กำหนดขาที่ 8 9 10 และ 11 ให้เป็นโหมด INTPUT
  pinMode(SWforword, INPUT);
  pinMode(SWbackword, INPUT);
  pinMode(SWleft, INPUT);
  pinMode(SWright, INPUT);
  pinMode(SWfan, INPUT);
}

void loop() {

```

```
38  //อ่านค่าจากสวิตช์มาเก็บไว้ในตัวแปรเพื่อนำไปใช้
39  SWF = digitalRead(SWforword);
40  SWB = digitalRead(SWbackword);
41  SWL = digitalRead(SWleft);
42  SWR = digitalRead(SWright);
43  SWf = digitalRead(SWfan);
44
45  //สร้างเงื่อนไขเพื่อตรวจสอบการกดสวิตช์
46  if (SWF == HIGH) {                    //เดินหน้า
47    digitalWrite(MotorLeftA, HIGH);
48    digitalWrite(MotorLeftB, LOW);
49    analogWrite(SpeedMotorLeft, 255);
50    digitalWrite(MotorRightA, HIGH);
51    digitalWrite(MotorRightB, LOW);
52    analogWrite(SpeedMotorRight, 255);
53  }
54  else if (SWB == HIGH) {               //เดินถอยหลัง
55    digitalWrite(MotorLeftA, LOW);
56    digitalWrite(MotorLeftB, HIGH);
57    analogWrite(SpeedMotorLeft, 255);
58    digitalWrite(MotorRightA, LOW);
59    digitalWrite(MotorRightB, HIGH);
60    analogWrite(SpeedMotorRight, 255);
61  }
62  else if (SWL == HIGH) {               //หมุนซ้าย
63    digitalWrite(MotorLeftA, LOW);
64    digitalWrite(MotorLeftB, HIGH);
65    analogWrite(SpeedMotorLeft, 255);
66    digitalWrite(MotorRightA, HIGH);
67    digitalWrite(MotorRightB, LOW);
68    analogWrite(SpeedMotorRight, 255);
69  }
70  else if (SWR == HIGH) {               //หมุนขาา
71    digitalWrite(MotorLeftA, HIGH);
72    digitalWrite(MotorLeftB, LOW);
73    analogWrite(SpeedMotorLeft, 255);
74    digitalWrite(MotorRightA, LOW);
75    digitalWrite(MotorRightB, HIGH);
76    analogWrite(SpeedMotorRight, 255);
77  }
```

```
78   else {                                    //หยุด
79     digitalWrite(MotorLeftA, HIGH);
80     digitalWrite(MotorLeftB, HIGH);
81     analogWrite(SpeedMotorLeft, 255);
82     digitalWrite(MotorRightA, HIGH);
83     digitalWrite(MotorRightB, HIGH);
84     analogWrite(SpeedMotorRight, 255);
85   }
86 }
```

1.5 เมื่อเขียนโปรแกรมเสร็จสิ้น ทำการตรวจสอบความถูกต้องโดยกดปุ่ม Verify และทำการ Upload

1.8 ทดลองกดสวิตช์แล้วสังเกตการเคลื่อนไหวของหุ่นยนต์

บันทึกผลการทดลองจากการสังเกตการเคลื่อนไหวของหุ่นยนต์

.....

.....

.....

.....

.....

แนบไฟล์หรือลิงค์วิดีโอผลการทดลองที่ 1

1.9 จงแก้ไขเพิ่มเติมโค้ดจากโปรแกรมเดิม **“โดยสั่งให้มอเตอร์พัดลมทำงานเมื่อกด SW5”**

บันทึกโค้ดโปรแกรมที่แก้ไขเพิ่มเติม

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....