หน่วยที่ 5

ออกแบบการควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ Arduino ขับเคลื่อนด้วยดีซีมอเตอร์

หัวข้อเรื่อง

1. สวิตซ์ Pull UP- Pull Down
2. สัญญาณ bounce
3. โปรแกรมย่อย
4. บอร์ดขับมอเตอร์ไฟฟ้ากระแสตรง
5. การเขียนโปรแกรมควบคุมทิศทางของมอเตอร์

สาระสำคัญ

มอเตอร์ไฟฟ้ากระแสตรง (DC motor) เป็นเครื่องกลไฟฟ้าชนิดหนึ่งที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล เมื่อได้รับการป้อนพลังงานไฟฟ้าที่เป็นไฟฟ้ากระแสตรงจะทำให้แกนของมอเตอร์หมุน แต่เนื่องจากการทำงานของมอเตอร์ไฟฟ้ากระแสตรงต้องการกระแสไฟฟ้าในปริมาณสูงกว่าความสามารถที่ไมโครคอนโทรลเลอร์จะจ่ายโดยตรงได้จึงต้องมีวงจรขับมอเตอร์โดยเฉพาะเพื่อทำหน้าที่ขับมอเตอร์ให้ทำงานได้ตามต้องการ

การควบคุมการทำงานของมอเตอร์ไฟฟ้ากระแสตรงนั้นสามารถทำได้เพียงป้อนไฟเข้าที่ขั้วมอเตอร์เมื่อต้องการให้มอเตอร์หมุนและเมื่อต้องการให้มอเตอร์หยุดหมุนก็เพียงหยุดป้อนไฟฟ้า หรือถ้าหากต้องการให้มอเตอร์หมุนกลับทิศทางก็สามารถทำได้โดยการสลับขั้วไฟฟ้าที่จ่ายให้กับมอเตอร์เพียงเท่านี้มอเตอร์ไฟฟ้ากระแสตรงก็จะสามารถหมุนกลับทิศทางได้ในทันที สำหรับวงจรขับมอเตอร์ที่สามารถควบคุมทิศทางการหมุนได้ประกอบด้วยอุปกรณ์ที่ถูกจัดวางที่มีลักษณะคล้ายตัว H ในภาษาอังกฤษจึงเรียกวงจรขับมอเตอร์ไฟฟ้ากระแสตรงในลักษณะนี้ว่าวงจรขับแบบ H- Bridge

สมรรถนะประจำหน่วยการเรียนรู้

ออกแบบการควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์

จุดประสงค์การเรียนรู้

จุดประสงค์ทั่วไป

1. เพื่อให้มีความรู้ความเข้าใจในการเขียนโปรแกรมรับค่าอินพุต
2. เพื่อให้มีความรู้ความเข้าใจในการเขียนโปรแกรมย่อยในการควบคุมมอเตอร์ไฟฟ้ากระแสตรง
3. เพื่อให้สามารถนำความรู้ไปประยุกต์การออกแบบหุ่นยนต์ที่ควบคุมด้วยมอเตอร์ไฟฟ้ากระแสตรง
4. เพื่อให้มีคุณธรรม จริยธรรม ค่านิยมอันพึงประสงค์

จุดประสงค์เชิงพฤติกรรม

1. อธิบายเขียนโปรแกรมรับค่าอินพุตจากลิมิตสวิตซ์
2. อธิบายการเขียนโปรแกรมย่อยในการควบคุมมอเตอร์
3. อธิบายการประยุกต์ออกแบบหุ่นยนต์ที่ควบคุมด้วยไมโครคอนโทรลเลอร์
4. เพื่อให้มีคุณธรรม จริยธรรม ค่านิยมอันพึงประสงค์

แบบทดสอบก่อนเรียน

หน่วยที่ 5 ออกแบบการควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ Arduino

ขับเคลื่อนด้วยดีซีมอเตอร์

<u>คำชี้แจง</u> 1. อ่านคำถามต่อไปนี้แล้วทำเครื่องหมายกากบาท (X) ข้อที่ถูกที่สุดลงในกระดาษคำตอบเพียงข้อเดียว

2. เวลาสำหรับทำแบทดสอบ 10 นาที

****

ข้อที่ 1. ข้อใดคือคำสั่งนี้ใช้ทำหน้าที่สำหรับกำหนดหน้าที่การทำงานของขา I/O ที่เป็น Digital I/O ของไมโครคอนโทรลเลอร์ Arduino

ก. int digitalRead(Pin)

ข. void pinMode(pin,mode)

ค. int analogRead(pin)

ง. digitalWrite(pin,value)

จ. attachInterrupt()

ข้อที่ 2 ข้อใดคือคำสั่งนี้ใช้ทำหน้าที่สำหรับกำหนดสถานะทาง OUTPUT ให้กับ Digital I/O Pin ว่าต้องการให้มีสภาวะทางโลจิกเป็น HIGH หรือ LOW

ก. int digitalRead(Pin)

ข. void pinMode(pin,mode)

ค. int analogRead(pin)

ง. digitalWrite(pin,value)

จ. attachInterrupt()

ข้อที่ 3 เทคนิคการส่งสัญญาณแบบสวิตซ์หรือส่งค่าดิจิตอล 0-1 โดยให้สัญญาณความถี่คงที่การควบคุมระยะเวลาสัญญาณสูงและสัญญาณต่ำ ที่ต่างกัน ก็จะทำให้ค่าแรงดันเฉลี่ยของสัญญาณสวิตซ์ต่างกันด้วยเป็นความหมายของค่าใด

ก. PWM (Pulse Width Modulation)

ข. PLL (Phase log loop)

ค. ACF (Auto Control Frequncy)

ง. AM (Amplitude Modulation)

จ. FM (Frequncy Modulation)

ข้อที่ 4. ค่าสถานะทาง Output ของ Digital Output Pin ที่จะต้องกำหนด ซึ่งสามารถกำหนดค่าสถานะให้กับ Pin ได้ 2 ค่าคือ HIGH และ LOWคือความหมายของข้อใด

ก. Mode

ข. Pin

ค. value

ง. code

จ. delay

ข้อที่ 5. การนำตัวต้านทานต่อเข้ากับ Vcc ทำให้อยู่ในสถานะ “HIGH” ตลอดเวลา และเมื่อกดสวิตซ์กระแสไฟฟ้าจะไหลลง Ground ซึ่งทำให้สถานะเป็นลอจิก “LOW” สถานะตัวต้านทานที่เกิดขึ้นตรงกับข้อใด

ก. Setup Resistor

ข. Pulse Resistor

ค. Pull- Down Resistor

ง. Pull- up Resistor

จ. Reset Resistor

ข้อที่ 6. การกำหนดให้ค่าของตัวแปรเป็นชนิดตัวเลขจำนวนเต็มตรงกับคำตอบในข้อใด

ก. boolean reading1;

ข. pinMode(sw1, INPUT);

ค. digitalWrite(pwm1, 28);

ง. int pwm1 = 5;

จ. delay(2000);

ข้อที่ 7. การกำหนดให้โปรแกรมหน่วงเวลาเป็นเวลา 2 มิลลิวินาที ตรงกับคำตอบในข้อใด

ก. boolean reading1;

ข. pinMode(sw1, INPUT);

ค. digitalWrite(pwm1, 28);

ง. int pwm1 = 5;

จ. delay(2000);

ข้อที่ 8. จากโปรแกรมคำตอบของข้อใดเมื่อเติมลงในช่องว่างแล้วทำให้โปรแกรมนั่นสั่งงานผ่านบอร์ดขับมอเตอร์ให้หยุดหมุน

```
if(reading1==LOW){
    digitalWrite(IA1, LOW);
     .....
    digitalWrite(pwm1, 28);
     delay(2000);
    }
```

ก. digitalWrite(IA1,HIGH);

ข. pinMode(sw1, INPUT);

ค. digitalWrite(pwm1, 28);

ง. digitalWrite(IB1, LOW);

จ. digitalWrite(IB1, HIGH);

ข้อที่ 9. ข้อใดคือความหมายที่ถูกที่สุดของคำสั่งข้างล่าง

```
if(reading1==HIGH && reading2 == LOW){
```

ก. เงื่อนไขทำงานภายใต้วงเล็บเมื่อเป็นจริงของตัวแปร reading1เท่ากับ HIGH แอนด์กับตัวแปรreading2 เท่ากับ LOW

ข. เงื่อนไขทำงานภายใต้วงเล็บเมื่อเป็นไม่จริงของตัวแปร reading1เท่ากับ HIGH แอนด์กับตัวแปรreading2 เท่ากับ LOW

ค. เงื่อนไขทำงานภายใต้วงเล็บเมื่อสถานะลอจิกของตัวแปร reading1เท่ากับ HIGH แอนด์กับตัวแปรreading2 เท่ากับ LOW

ง. เงื่อนไขทำงานภายใต้วงเล็บเมื่อสถานะลอจิกของตัวแปร reading1เท่ากับ HIGH ออร์กับตัวแปรreading2 เท่ากับ LOW

จ. เงื่อนไขทำงานภายใต้วงเล็บเมื่อเป็นไม่จริงของตัวแปร reading1เท่ากับ HIGH ออร์กับตัวแปรreading2 เท่ากับ LOW

ข้อที่ 10. จากโปรแกรมคำตอบของข้อใดเมื่อเติมลงในช่องว่างแล้วทำให้โปรแกรมนั่นสั่งงานผ่านบอร์ดขับมอเตอร์ให้หมุน

```
if(reading1==LOW){
    digitalWrite(IA1, LOW);
     .....
    digitalWrite(pwm1, 28);
     delay(2000);
```

}

ก. digitalWrite(IA1,HIGH);

ข. digitalWrite(IB2,HIGH);

ค. digitalWrite(pwm1, 28);

ง. digitalWrite(IB1,HIGH);

จ. digitalWrite(IB1, HIGH);

หน่วยที่ 5

ออกแบบการควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ Arduino ขับเคลื่อนด้วยดีซีมอเตอร์

5.1 ปุ่มหรือสวิตซ์

สวิตซ์คืออุปกรณ์ที่ตัด-ต่อ ไฟฟ้าเข้าถึงกัน เหมือนเป็นสะพาน สวิตซ์ที่ใช้กันมาก

Button หรือปุ่มกดแบบโรงงาน มักมีขนาดใหญ่ และทนทานมาก

Tact switch เป็นแบบปุ่มกด พบได้บนแผงวงจรอิเล็กทรอนิกส์ บนบอร์ด Arduino จะมีเป็นปุ่มกดสำหรับ Reset

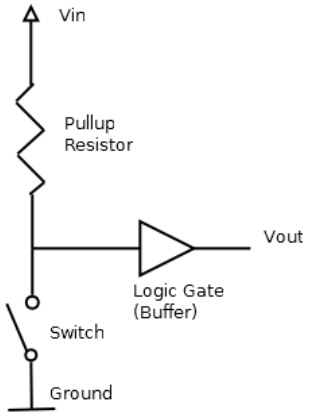
Micro Switch หรือบางทีเรียกว่า Limit Switch

Mercury Switch หรือบางทีเรียกสวิตซ์ปรอทจะใช้ปรอทเป็นตัวนำไฟฟ้า สำหรับสวิตซ์แบบนี้ถูกประยุกต์ไปใช้ตรวจสอบความเอียงได้

วงจร Pull-up , Pull- down

สถานะที่กำหนดให้ขาของอุปกรณ์อิเล็กทรอนิกส์ รอรับอินพุต (INPUT) ขาพอร์ตจะเป็น High Impedance คือมีความต้านทานสูงมาต่ออยู่ ทำให้ขาพอร์ตนั้นเสมือนถูกปล่อยลอย ค่าอินพุตที่อ่านกลับมาได้ มันไม่แน่นอน

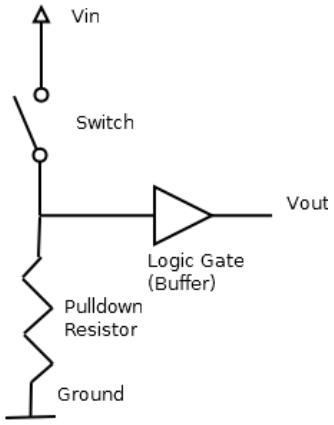
ดังนั้นในงานขาพอร์ตอินพุต วงจรของสวิตซ์ จำเป็นมากที่ต้องมี Pull-up Resistor หรือ Pull-down Resistor เพื่อที่จะกำหนดสภาวะดิจิตอลที่แน่นอน ให้กับอุปกรณ์อิเล็กทรอนิกส์ตามปกติตัวต้านทานที่ใช้ในวงจร Pull-up หรือ Pull-down จะใช้ค่าประมาณ 5kΩ– 20kΩ



รูปที่ 5.1 การต่อ Pull-up Resistor

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

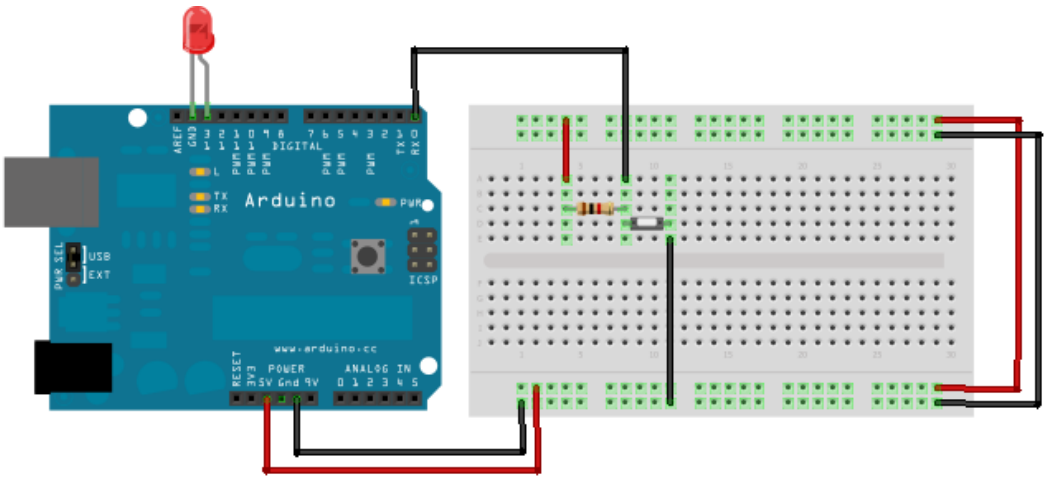
Pull-down Resistor โดยใน Pull-down จะมีลักษณะคล้ายกับ Pull-up Resistor แตกต่างตรงที่สภาวะปกติของ Pull-down จะเป็นลอจิก “LOW” หรือ “0” เมื่อมีการกดปุ่ม กระแสไฟจะไหลเข้าขาอินพุตทำให้ลอจิกเป็น “HIGH” หรือ “1” ได้ การทำงานในลักษณะนี้เรียกว่า Active High



รูปที่ 5.2 การต่อ Pull-down Resistor

การทดลองสวิตช์เบื้องต้น

สวิตช์ไฟแบบกดติดปล่อยดับ โดยสวิตช์จะต่อแบบ Pull Up Resistor ดังรูป 5.3



รูปที่ 5.3 สวิตช์ต่อแบบ Pull Up Resistor ควบคุม เปิด/ปิด LED

โดยวงจรที่ใช้ จะมีการ Pull-Up ตัวต้านทาน เอาไว้ระดับแรงดัน ในสภาวะปกติ จะอยู่ที่ 5V หรือ ลอจิก “1” แต่ถ้าทีการกดปุ่ม แรงดันจะอยู่ที่ 0 V หรือลอจิก “0” หรือเรียกว่า Active Low

โปรแกรมที่ 5.1

```
int buttonPin=2;
 int ledPin=13;
 boolean  buttonState  =0;
 void setup(){
     pinMode(ledPin,OUTPUT);
     pinMode(buttonPin,INPUT);
}
Void loop(){
     buttonState=digitalRead(buttonPin);
          if (buttonstate==HIGH){
               digitalWrite(ledPin,LOW);
               }else{
                digitalWrite(ledPin,HIGH);
               }
          }
```

ตามโปรแกรมที่ 5.1 ปกติเมื่อไม่มีการกดปุ่มไฟ LED จะไม่สว่าง แต่ถ้ามีการกดปุ่ม LED จะติดขึ้น แต่เมื่อปล่อย LED จะดับ ซึ่งต้องกำหนดขาพอร์ต ButtonPin ให้เป็นอินพุต (INPUT) ด้วย โดยใช้คำสั่ง

pinMode(); // ปกติพอร์ตจะถูกกำหนดให้เป็น INPUT อ่านค่าสถานะจาก pin ส่งค่ากลับมาเป็นHIGH หรือ LOW

การอ่านสถานะดิจิตอลใช้คำสั่งdigitalRead(pin)//อ่านค่าสถานะจาก pin ส่งค่ากลับมาเป็น HIGH หรือ LOW

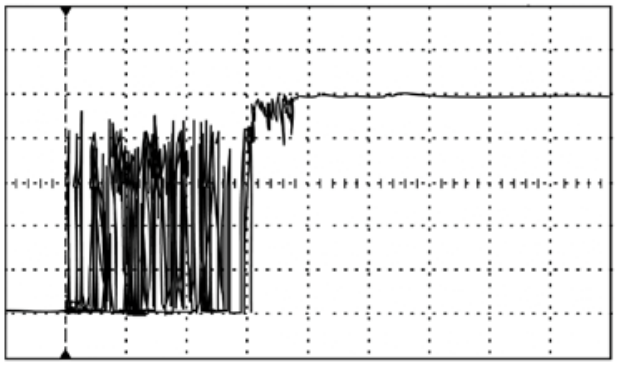
คำสั่ง digitalRead จะส่งค่ากลับมาหรือเรียก Return ค่ากลับ ดังนั้นควรสร้างตัวแปรไว้รับค่าด้วย ตามตัวอย่างจะสร้างตัวแปรชื่อ buttonState ไว้รับค่า(return)

buttonState=digitalRead(buttonPin);

สถานการณ์กดหรือปล่อยสวิตช์อยู่ในตัวแปร buttonState

5.2 สัญญาณ bounce

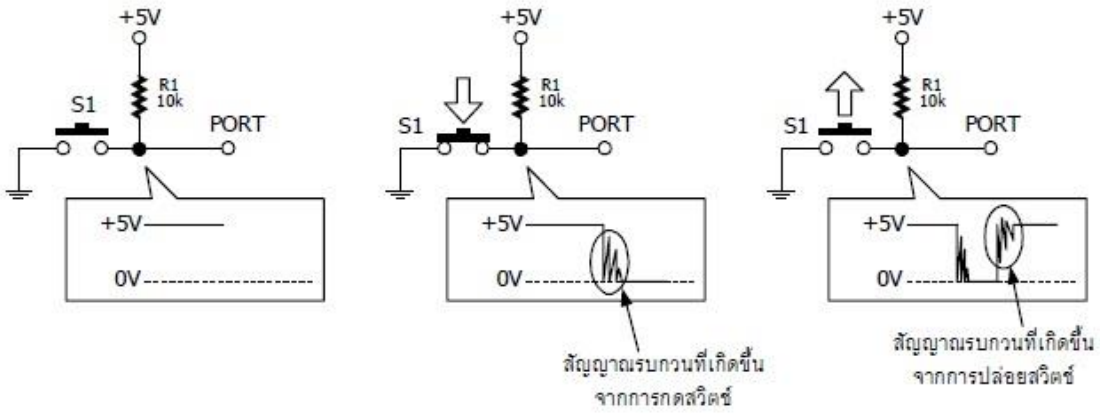
ปัญหาการ bounce ของสัญญาณคือ อาการไกว่ของสัญญาณซึ่งจะเกิดในสั้นๆ 5-50 นาโนวินาที สาเหตุหนึ่งที่ทำให้เกิดในช่วงที่กดปุ่มหน้า contact จะสัมผัสไม่แนบสนิทจะเกิดสัญญาณ bounce คืออาการที่สัญญาณจะสลับเป็น HIGH หรือ LOW อย่างรวดเร็ว สังเกตจากกราฟก่อนที่จะเข้าสู่สภาวะเสถียร



Switch bounce (going from off to on)

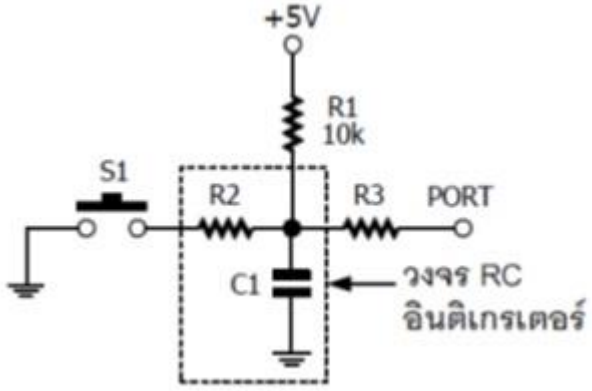
รูปที่ 5.4 สัญญาณการเกิด bounce

โดยทั่วไปแล้วสวิตซ์ใช้จะเป็นสวิตซ์ทางกลที่ประกอบด้วยหน้าสัมผัสโลหะ ในการกดสวิตซ์ให้ต่อวงจรพบว่าหน้าสัมผัสจะมาสัมผัสกันสนิททันที โดยมีช่วงเวลาที่เริ่มสัมผัส และหลุดเป็นช่วงเวลาสั้นๆก่อนที่หน้าสัมผัสของสวิตซ์จะต่ออย่างสมบูรณ์ และเมื่อวัดสัญญาณที่ได้จากสวิตซ์พบว่าสัญญาณมีการสั่น



รูปที่ 5.5 แสดงการเกิดสัญญาณรบกวนเมื่อมีการกดและปล่อยสวิตซ์ในวงจรดิจิตอล

การแก้ปัญหาที่ระดับสัญญาณเกิดการสั่นนี้เรียกว่าการดีเบาซ์(Debounce) หลักการแก้ไขสัญญาณรบกวนแบบบนคือ หน่วงเวลาการเกิดขึ้นของสัญญาณพัลส์เล็กน้อย เพื่อให้วงจรไม่สนใจสัญญาณที่เกิดขึ้นในช่วงเริ่มต้นกดสวิตซ์ ซึ่งทำได้หลายวิธี เช่นวิธีการแรกทำได้โดยใช้อุปกรณ์อิเล็กทรอนิกส์พื้นฐาน อย่างตัวต้านทานและตัวเก็บประจุ โดยต่อกันในลักษณะวงจร RC อินติเกรเตอร์ดังรูปที่ 5.6 ด้วยวิธีการนี้จะช่วยลดผลสัญญาณรบกวนที่เกิดขึ้นจากการกดสวิตซ์ได้ระดับหนึ่ง โดยประสิทธิภาพของวงจรจะขึ้นกับการเลือกค่าของตัวต้านทานและตัวเก็บประจุ หากเลือกค่าของตัวเก็บประจุน้อยเกินไป อาจไม่สามารถลดสัญญาณรบกวนได้ แต่ถ้าเลือกค่ามากเกินไปจะทำให้ความไวในการตรวจจับการกดสวิตซ์ลดลง นั่นคืออาจต้องกดสวิตซ์มากกว่า 1 ครั้งเพื่อให้ได้สัญญาณที่ต้องการ



รูปที่ 5.6 การต่อวงจร RC อินติเกรเตอร์ เพื่อแก้ไขปัญหารบกวนจากกดสวิตซ์

2.) แก้ด้วยซอฟท์แวร์ โดยการหน่วงเวลาเมื่อได้รับสัญญาณไปช่วงหนึ่งแล้วตรวจสอบอีกครั้ง โดยเขียนโปรแกรมให้ตรวจจับการเปลี่ยนแปลงระดับลอจิกของสวิตซ์ ถ้าพบว่ามีการเริ่มกดสวิตซ์ ก็คือระดับลอจิกเปลี่ยนจาก “1” เป็น “0” แสดงว่ากดสวิตซ์สมบูรณ์แล้ว

ระยะเวลาที่ต้องการหน่วงเวลาหาได้จากการใช้ออสซิลโลสโคปวัดระดับสัญญาณ หรือใช้การทดลองป้อนค่าหน่วงเวลาแล้วปรับค่าจนกระทั่งได้ค่าที่แก้ปัญหาการเบาซ์ของสวิตซ์ได้อย่างสมบูรณ์ในโปรแกรมที่ ได้ทดลองโดยการใช้ค่าหน่วงเวลาเท่ากับ 10 มิลลิวินาที ซึ่งสามารถเปลี่ยนได้ที่บรรทัด #define Debounce 10

โปรแกรมที่ 5.2 ไฟล์ DebounceSW.ino โปรแกรมภาษา C ของ Arduino อ่านค่าสวิตซ์เพื่อควบคุม LED แบบมีการแก้สัญญาณรบกวนากกดสวิตซ์หรือดีเบาซ์

```
/*
*Read input from push button for control states of LED
```

```
* change operation of push button to toggle switch
 * File : DebouncedSW.ino*/
#define IN_PIN 7//the number of the input pin
#define OUT_PIN 11//the number of the output pin
#define Debounce 10//debounce time=10ms
 int state=HIGH;//the current state of the output pin
 int reading;//the current reading from the input pin
 int previous = HIGH;//the previous reading from the input pin
 void setup()
     {
      pinMode(IN_PIN,INPUT);
      pinMode(OUT_PIN,INPUT);
     digitalWrite(OUT_PIN,State);
      }
  Void loop()
      {
       reading=digitalRead(IN_PIN);
       if (reading==LOW&&previous==HIGH)
          {
            delay(Debounce);// wait for debounce
        if(digitalRead(IN_PIN)==LOW)
             state=!state;//invert the state of output LED
             digitalWrite(OUT_PIN,state);
              } previous=reading;
```

5.3 โปรแกรมย่อย (Function)

ในการเขียนฟังก์ชั่นนั้น เมื่อโปรแกรมมีขนาดใหญ่และมีการทำงานที่สลับซับซ้อนมากขึ้น การใช้คำสั่งจำพวกตรวจเงื่อนไขต่างๆนั้นไม่เพียงพอในการแก้ปัญหาได้ จึงมีแนวคิดในการแบ่งการทำงานของโปรแกรมออกเป็นส่วนๆโดยแต่ละส่วนก็จะถูกกำหนดให้ทำหน้าที่อย่างใดอย่างหนึ่งจน

เสร็จสิ้นหน้าที่ ซึ่งส่วนของโปรแกรมที่ถูกแบ่งแยกออกมาเพื่อทำหน้าที่อย่างใดอย่างหนึ่ง นั้นจะถูกเรียกว่าโปรแกรมย่อย ซึ่งภาษาซีจะเรียกว่าฟังก์ชั่น (Function) โดยมีรูปแบบดังนี้

รูปแบบของฟังก์ชั่น

รูปแบบการส่งค่ากลับ ชื่อฟังก์ชั่น(ค่าที่ส่งให้ฟังก์ชั่น 1,...2,....n)

{

ชนิดตัวแปร ชื่อตัวแปร

.

.

คำสั่งต่างๆ

.

.

return(ค่าที่ส่งกลับ)

}

รูปแบบการส่งค่ากลับ หมายถึง ชนิดของตัวแปรที่จะใช้ในการส่งผ่านค่าจากโปรแกรมย่อยกลับมายังโปรแกรมหลัก หรือ โปรแกรมที่ทำหน้าที่เป็นผู้เรียกใช้โปรแกรมย่อย ถ้าไม่ต้องการส่งค่ากลับ ให้กำหนดรูปแบบการส่งค่ากลับเป็น **void** แทน โดยรูปแบบการส่งค่ากลับต้องประกาศไว้หน้าที่ของฟังก์ชั่นเสมอ

ชื่อฟังก์ชั่น หมายถึง ชื่อของฟังก์ชั่นที่ต้องการตั้งขึ้น ซึ่งใช้ข้อกำหนดและหลักเกณฑ์ในการตั้งชื่อเหมือนการตั้งชื่อตัวแปร

ค่าที่ส่งให้ฟังก์ชั่น หมายถึง ข้อมูลที่ต้องการส่งผ่านไปให้ฟังก์ชั่นทำงาน ซึ่งต้องกำหนดไว้ภายในวงเล็บ () ที่อยู่หลังชื่อฟังก์ชั่น ซึ่งถ้าต้องการผ่านค่าให้กับฟังชั่นมากกว่า 1 ค่า ให้ใช้เครื่องหมาย คอมม่า(,)เป็นตัวแบ่งแยกข้อมูลแต่ละชุดด้วย แต่ถ้าไม่ต้องการให้มีการส่งผ่านค่าไปให้กับฟังก์ชั่น ควรกำหนดค่าเป็น **void** แทน

ค่าที่ส่งคืนกลับ หมายถึง ค่าข้อมูลที่ต้องการส่งกลับไปยังฟังก์ชั่นผู้เรียก โดยในการส่งค่าคืนกลับไปยังฟังก์ชั่นผู้เรียกนั้นจะใช้คำสั่ง return เป็นคำสั่งในการส่งค่ากลับเสมอ

ชนิดของฟังก์ชั่น

รูปแบบของฟังก์ชั่นที่ใช้ภาษาซี จะมีอยู่ด้วยกันหลายแบบ ขึ้นอยู่กับการออกแบบโปรแกรมและความจำเป็นในการใช้งานของโปรแกรม ซึ่งบางโปรแกรมก็มีการส่งผลการทำงานกลับมาให้กับโปรแกรมที่เป็นฝ่ายเรียกใช้ด้วย บางโปรแกรมก็ไม่มีการส่งค่ากลับคืนมา บางโปรแกรม

ก็มีการส่งผ่านค่าข้อมูลไปให้กับโปรแกรมย่อยเพื่อใช้เป็นค่าการทำงานด้วย ซึ่งสามารถแบ่งออกได้เป็น 2 แบบคือ

1.) ฟังก์ชั่นที่ไม่มีการส่งค่ากลับมายังโปรแกรมผู้เรียก

ฟังก์ชั่นแบบนี้จะมีรูปแบบการส่งค่ากลับเป็น void ซึ่งเมื่อเรียกใช้ก็จะเรียกเฉพาะชื่อของฟังก์ชั่นเท่านั้น ซึ่งฟังก์ชั่นแบบนี้ ยังแบ่งออกเป็น 2 แบบ คือ แบบที่มีการส่งผ่านค่าข้อมูลมาให้กับฟังก์ชั่นและแบบที่ไม่มีการส่งผ่านค่าข้อมูลมาให้ฟังก์ชั่น

2.) ฟังก์ชั่นที่มีการส่งค่ากลับมายังโปรแกรมผู้เรียก

ฟังก์ชั่นแบบนี้ต้องกำหนด รูปแบบการส่งค่ากลับ ไปให้กับฟังก์ชั่นด้วย ซึ่งในการเรียกใช้งานโปรแกรมย่อยแบบนี้ โปรแกรมที่เป็นฝ่ายเรียกใช้ต้องสร้างตัวแปร ซึ่งมีชนิดข้อมูลตรงกับรูปแบบการส่งค่ากลับ ของฟังก์ชั่นด้วย ซึ่งฟังก์ชั่นแบบนี้ ยังมีแบ่งแยกเป็น 2 แบบ คือ แบบที่มีการส่งผ่านค่าข้อมูลมาให้กับฟังก์ชั่น และแบบที่ไม่มีการส่งผ่านค่าข้อมูลมาให้กับฟังก์ชั่น

การส่งผ่านค่าระหว่างฟังก์ชั่น

สำหรับการส่งผ่านค่าข้อมูลจากฟังก์ชั่นที่เป็นฝ่ายผู้เรียก ไปให้กับฟังก์ชั่นที่เป็นฝ่ายถูกเรียกใช้สามารถใช้ได้กับ ฟังก์ชั่นแบบที่มีการส่งค่ากลับมายังฟังก์ชั่นผู้เรียก หรือฟังก์ชั่นที่ไม่มีการส่งค่ากลับมายังฟังก์ชั่นผู้เรียกก็ได้ โดยวิธีการส่งผ่านค่าให้กับฟังก์ชั่นมี 3 แบบหลักคือ

2.1) การส่งผ่านค่าด้วยตัวแปร

2.2) การส่งผ่านด้วยค่า Pointer

2.3) การส่งผ่านค่าด้วย Array

โปรแกรมที่ 5.3 ฟังก์ชั่นที่ไม่มีการส่งค่ากลับมายังโปรแกรมผู้เรียก

```
void  setup()
  {
   Serial.begin(19200);
  }
void loop()
  {
   PlotDot(5);
   HelloWorld();
   PlotDot(10);
   while(1);
```

```
    }
  void  HelloWorld(void);
    {
     Serial.println("Hello World from  Arduino");
     }
    void PlotDot(int num1)
     {
    for(int i =0;i<=num1;i++)
     {
     Serial.print("*");
     }
     Serial.println();
     }
```

ผลลัพธ์

```
**********************************************
 Hello  World  from Arduino
**********************************************
```

จากตัวอย่างเป็นการแสดงให้เห็นการใช้ฟังก์ชั่น แบบไม่มีการส่งค่ากลับมายังโปรแกรมผู้เรียก โดยในตัวอย่างจะมีโปรแกรมย่อยแบบไม่มีการส่งค่ากลับมายังโปรแกรมผู้เรียก 2 แบบ คือ HelloWorld เป็นแบบไม่มีการส่งผ่านข้อมูลให้กับฟังก์ชั่น ส่วน PlotDot จะมีการส่งผ่านค่าข้อมูลให้กับฟังก์ชั่นด้วย โดยต้องส่งค่าทางตัวแปร num1 ซึ่งเป็นแบบ int ให้กับฟังก์ชั่นด้วย โดยค่าของ num1 จะเป็นค่าที่ถูกนำไปใช้เป็นจำนวนการสั่งพิมพ์เครื่องหมาย (*) ของฟังก์ชั่น PlotDot

โปรแกรมที่ 5.4 ฟังก์ชั่นที่ต้องมีการส่งค่ากลับมายังโปรแกรมผู้เรียก

```
 void  setup()
    {
     Serial.begin(19200);
     }
  void  loop(){
     int num0;
```

```
num0=Sum(10,5);
Serial.println(num0);
Serial.println(Sum(8,2));
while(1);
      }
      int Sum(int num1,int num2)
     {
  Serial.print("Result=");
  return(num1+num2);
}
```

ผลลัพธ์

Result=15

Result=10

จากตัวอย่างแสดงให้เห็นการทำงานของโปรแกรมย่อยหรือฟังก์ชั่นแบบที่มีการส่งค่าทั้งไปและกลับโดยจะมีทั้งการส่งผ่านค่าจากโปรแกรมผู้เรียกไปให้โปรแกรมย่อยและเมื่อจบการทำงานของโปรแกรมย่อยก็จะมีการส่งค่ากลับคืนมายังโปรแกรมผู้เรียกด้วย

โปรแกรมที่ 5.5 การส่งผ่านค่าให้ฟังก์ชั่นด้วย Array

```
void  setup()
      {
      Serial.begin(19200);
       }
 void loop()
      {
     char msg1[]={"Hello  World"};
     char msg2[]={'I', 'a', 'm', 'A', 'r', 'd', 'u', 'I', 'n', 'o', '\0'};
     int num1[10];
     show_string(msg1); // เรียกฟังก์ชั่น show_string  โดยผ่านค่า msg1 ไปให้ด้วย
     show_string(msg2); //เรียกฟังก์ชั่น show_string  โดยผ่านค่า msg2 ไปให้ด้วย
     for (int i=0;i<10;i++)//กำหนด  Counter=0.9
```

```
  {
   num1[i]=i; // เติมตัวเลขแบบ int  ใน  Array num1 =0.9
   }
 show_array(num1);//เรียกใช้ฟังก์ชั่น  show_Array โดยผ่านค่า num1 ไปให้ด้วย
  while(1); // วนรอบอยู่กับที่(หยุดการทำงานของโปรแกรม)
  }
 void show_string(char str[])// โปรแกรมย่อย  show_string
  {
   Serial.println(str);//พิมพ์ค่าของ str  ซึ่งเป็น array  แบบ char
  }
 void show_array(int x[10])  //ปรแกรมย่อย show_arry
  {
  for(int i=0;i<10;i++)//กำหนด  Counter = 0.9
  }
  Serial.print(x[i]);//พิมพ์ค่าใน  Array ของ x ซึ่งเป็น Array  แบบ int
   }
  }
```

ผลลัพธ์

```
Hello  World
I am  Arduino
0123456789
```

โปรแกรมที่ 5.5 แสดงให้เห็นการผ่านค่าให้กับฟังก์ชั่นด้วย Array และ String ซึ่ง String ก็เป็น Array แบบหนึ่ง เพียงแต่ต้องมีข้อมูลใน Array ตำแหน่งสุดท้ายเป็นศูนย์เท่านั้น โดยการประกาศสร้างตัวแปรให้ Array ของ msg1 จะใช้รูปแบบการกำหนดค่าให้ Array เป็นแบบข้อความ String โดยตรง ส่วน Array ของ msg2 จะกำหนดเป็นแบบ char ซึ่งการกำหนดเป็นตัวอักษรทีละตัว

5.3 บอร์ดขับมอเตอร์ไฟฟ้ากระแสตรง

5.3.1 BTS7960 H-Bridge DC Motor Drive Model

ใช้สำหรับขับเคลื่อนดีซีมอเตอร์ที่ต้องการกระแสสูงๆใช้สัญญาณ PWM ในการควบคุมความเร็ว รองรับความเร็วของ PWM ได้ถึง 25 KHz สามารถควบคุมมอเตอร์ได้ 1 ตัว และควบคุมหมุนซ้าย ขวาได้ แรงดันไฟเลี้ยงมอเตอร์เท่ากับ 6- 27 VDC กระแสเอาต์พุตสูงสุดเท่ากับ 47 A Max ในทางปฏิบัติควรใช้กระแสไม่เกิน 20 A เพื่อความปลอดภัย

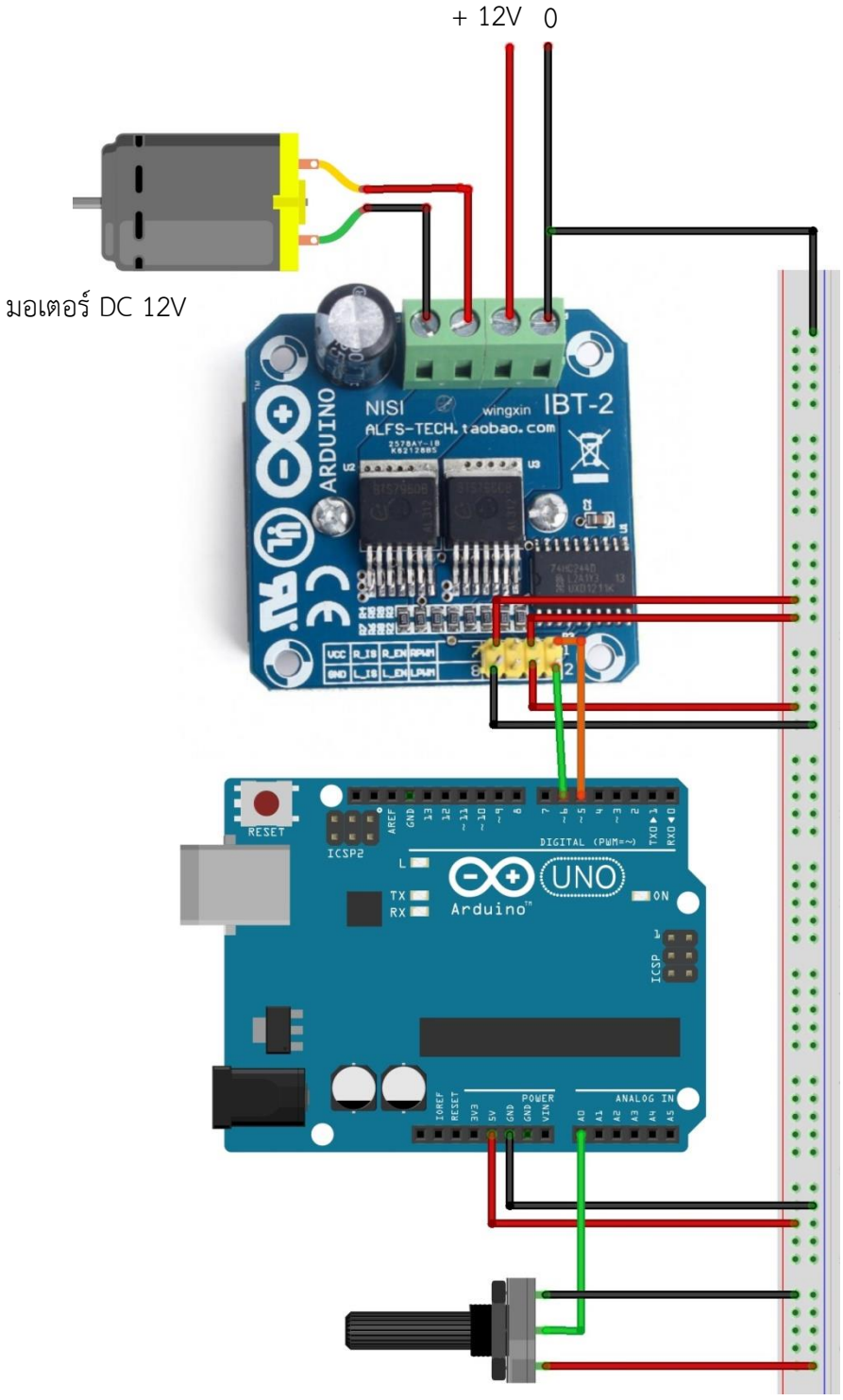
รูปที่ 5.7 บอร์ดไดร์ฟมอเตอร์ DC Model BTS7960

ที่มา: https://www.arduitronics.com/product/983/motor-drive-module-bts7960-43a-with-h-bridge

โปรแกรมที่ 5.6 การใช้งานบอร์ดร่วมกับ Potentiometor เพื่อที่จะขับ IBT-2 โดยการ Full reverse speed จนถึง Full forward speed

```
int SENSOR_PIN = 0; // center pin of the potentiomete 16
int RPWM_OUTPUT=5;
int LPWM_OUTPUT=6;
void setup() {
    pinMode(RPWM_OUTPUT,OUTPUT);
     pinMode(LPWM_OUTPUT,OUTPUT);
 }
void loop() {
    int sensorValue=analogRead(SENSOR_PIN);
    if(sensorValue<512){
   //reward rotion
    int reversePWM=-(sensorValue-512)/2;
    analogWrite(LPWM_OUTPUT,0);
```

```
 analogWrite(RPWM_OUTPUT,reversePWM);
 }
 else{
 //forword rotion
   int forwardPWM=(sensorValue-512)/2;
 analogWrite(LPWM_OUTPUT,forwardPWM);
 analogWrite(RPWM_OUTPUT,0);
  }
}
```



รูปที่ 5.8 การใช้งานบอร์ด BTS7960 H-Bridge DC Motor Drive Model

5.3.2 บอร์ดไดร์ฟมอเตอร์ DC. Smile รุ่น EVO24V9.3

รูปที่ 5.9 บอร์ด EVO24V9.3 H-Bridge DC Motor Drive Model

EVO24V9.3 เป็นบอร์ดขับมอเตอร์แบบชนิดแปรงถ่าน ที่ใช้ไอซี Full bridge motor drive เบอร์ VNH5019 ของ ST Microelectronic โดยออกแบบให้ตัวบอร์ดมีขนาดเล็กน้ำหนักเบา มีความทนทานสูง และการต่อใช้งานง่ายเหมาะสำหรับมอเตอร์ขนาดเล็กไม่เกิน 150 W สามารถขับกระแสต่อเนื่องได้สูงสุด 10 A และขับกระแสชั่วขณะได้ 30 A ที่แรงดันไฟฟ้าสูงสุดที่ 24 V โดยวงจรจะมีระบบแยกสัญญาณด้วย Opto-Isolator เพื่อป้องกันสัญญาณรบกวนที่อาจเกิดขึ้นจากมอเตอร์ และยังสามารถป้องกันกระแสไฟฟ้าย้อนกลับไปยังภาคควบคุม อีกทั้งยังมีระบบป้องกันต่างๆเช่น ระบบป้องกันการต่อไฟเลี้ยงกลับขั้ว ระบบตัดการทำงานเมื่ออุณหภูมิสูงเกินไป และระบบป้องกันความเสียหายที่อาจจะเกิดขึ้นหากระบบควบคุมหลักมีปัญหาเป็นต้น การประยุกต์ใช้งานใช้สำหรับควบคุมความเร็ว ทิศทาง DC Motor สำหรับหุ่นยนต์หรือประยุกต์ใช้ควบคุมอุปกรณ์ไฟฟ้าอื่นๆเช่นหลอดไฟกำลังสูง โซลินอยส์ เป็นต้น

คุณสมบัติ

- IC Full bridge motor drive เบอร์ VNH5019
- ขับกระแสไฟฟ้าชั่วขณะ 30 A
- ขับกระแสไฟฟ้าต่อเนื่อง 10 A ที่แรงดัน 24 VDC
- แรงดันไฟฟ้าอินพุต VCC7-28 VDC
- แรงดันไฟฟ้าสูงสุดเอาต์พุตสูงสุด 0.98 x VCC
- ควบคุมความเร็วมอเตอร์ด้วยการปรับความกว้างของสัญญาณ PWM ที่ความถี่ไม่เกิน 10 KHz

- แยกสัญญาณควบคุมไฟฟ้าด้วย Opto Isolator
- มี Output สำหรับวัดกระแสไฟฟ้าขณะขับโหลดโดยมีสัดส่วน (ประมาณ 140 mv/A)
- แสดงทิศทางการหมุนด้วย LED
- มีวงจรป้องกันการจ่ายไฟเลี้ยงกลับขั้ว
- ตัดการทำงานเมื่ออุณหภูมิสูงเกินกว่า 100 C
- มีฮิทซิงค์ช่วยระบายความร้อน
- ขนาด PCB 50 x 50 mm.
- น้ำหนัก 32 g.

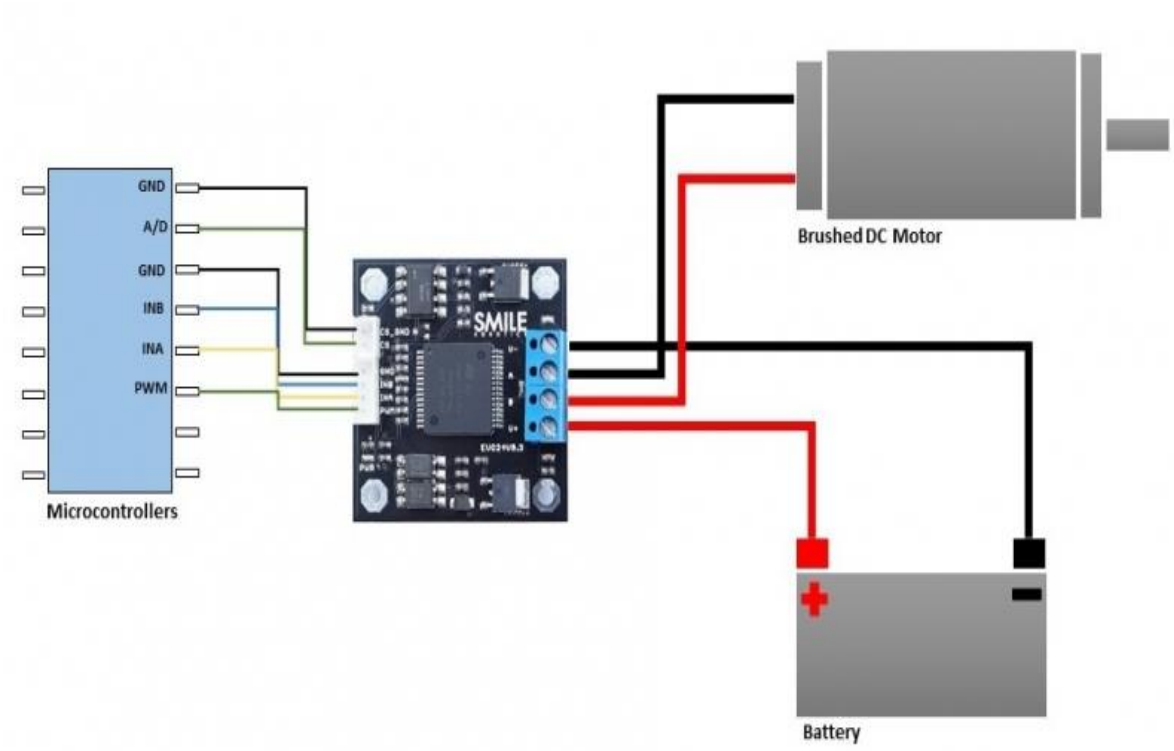
PWM เป็นการสั่งงานด้วยสัญญาณ PWM (Pulse width Modulation)

ในการควบคุมความเร็ว และใช้สัญญาณลอจิกป้อนให้กับ INA และ INB ในการควบคุมทิศทางการหมุนของมอเตอร์ โดยทั่วไปใช้สั่งงานผ่านไมโครคอนโทรลเลอร์หรือไมโครโปรเซสเซอร์ เช่น ARM,AVR ,PIC,Arduino ,NodeMCU และ Raspbery Pi เป็นต้น ในโหมดนี้จะสามารถรองรับความถี่สัญญาณ PWM ได้สูงสุดถึง 10 KHz

ตารางที่ 5.1 อธิบายความหมายของขาใช้งานของบอร์ดไดร์ฟมอเตอร์ รุ่น EVO24V9.3

PIN	Description
V-	กราวด์(Ground) ของแหล่งจ่ายไฟฟ้า เช่น Battery หรือ Power Supply
A	ขั้วต่อไปยังมอเตอร์ เส้นที่ 1
B	ขั้วต่อไปยังมอเตอร์ เส้นที่ 2
V+	ขั้วบวกของแหล่งจ่ายไฟฟ้า เช่น Battery หรือ Power Supply ซึ่งมีแรงดันอยู่ในช่วง 7 – 28 VDC
GND	กราวด์ของสัญญาณควบคุม
INB	สัญญาณลอจิกควบคุมทิศทาง หมุนทิศทางที่ 1
INA	สัญญาณลอจิกควบคุมทิศทางหมุนทิศทางที่ 2
PWM	สัญญาณ Pulse Width Modulation ควบคุมความเร็ว
CS	สัญญาณ Output สำหรับวัดค่ากระแสไฟฟ้าขณะขับโหลด
CS_GND	กราวด์สำหรับวัดค่ากระแสไฟฟ้าขณะขับโหลด

(ที่มา : http://smile-robotics.com/PD837753-สินค้า-evo24v9_3.html)



รูปที่ 5.10 การใช้งานบอร์ด EVO24V9.3 H-Bridge DC Motor Drive Model

(ที่มา : http://smile-robotics.com/PD837753-สินค้า-evo24v9_3.html)

การส่งค่าสัญญาณ Pulse Width Modulation (PWM)

PWM คือเทคนิคการส่งสัญญาณแบบสวิตซ์หรือส่งค่าดิจิตอล 0-1 โดยให้สัญญาณความถี่คงที่ การควบคุมระยะเวลาสัญญาณสูงและสัญญาณต่ำ ที่ต่างกัน ก็จะทำให้ค่าแรงดันเฉลี่ยของสัญญาณสวิตซ์ต่างกันด้วย

สำหรับโมดูล PWM ของ Arduino มีความละเอียด 8 bit หรือ ปรับได้ 255 ระดับ ดังนั้นค่าสัญญาณ 0 โวลต์ถึง 5 โวลต์ จะถูกแสดงเป็นสัญญาณดิจิตอลจะได้ 0 ถึง 255 ซึ่งสามารถเทียบสัดส่วนคำนวณจากเลขจริงเป็นเลขทางดิจิตอลได้

ตัวอย่างที่ 5.1

ถ้าแรงดัน 5 V เท่ากับ 255

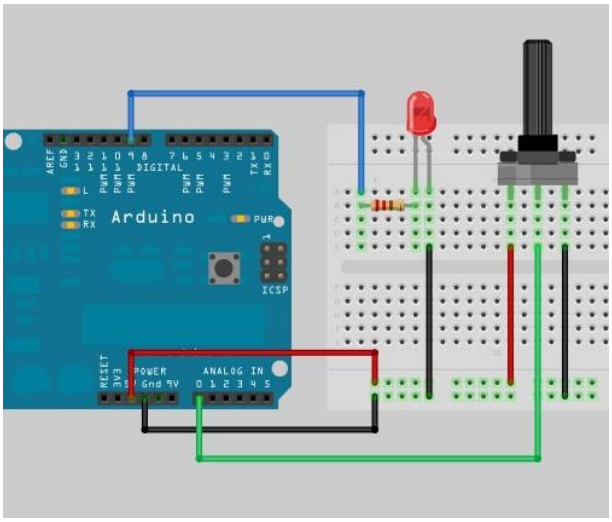
1 V จะมีค่าเท่ากับเท่าไร

= 255/5.0V*1.V

= 51

ถ้าอยากให้ VOUT = 1.0 V สามารถสั่งให้ค่า PWM = 51 นั่นเอง

ตัวอย่างที่ 5.2 วงจรการส่งค่า PWM ควบคุมความสว่างของ LED



รูปที่ 5.11 แสดงวงจรการส่งค่า PWM ควบคุมความสว่าง LED

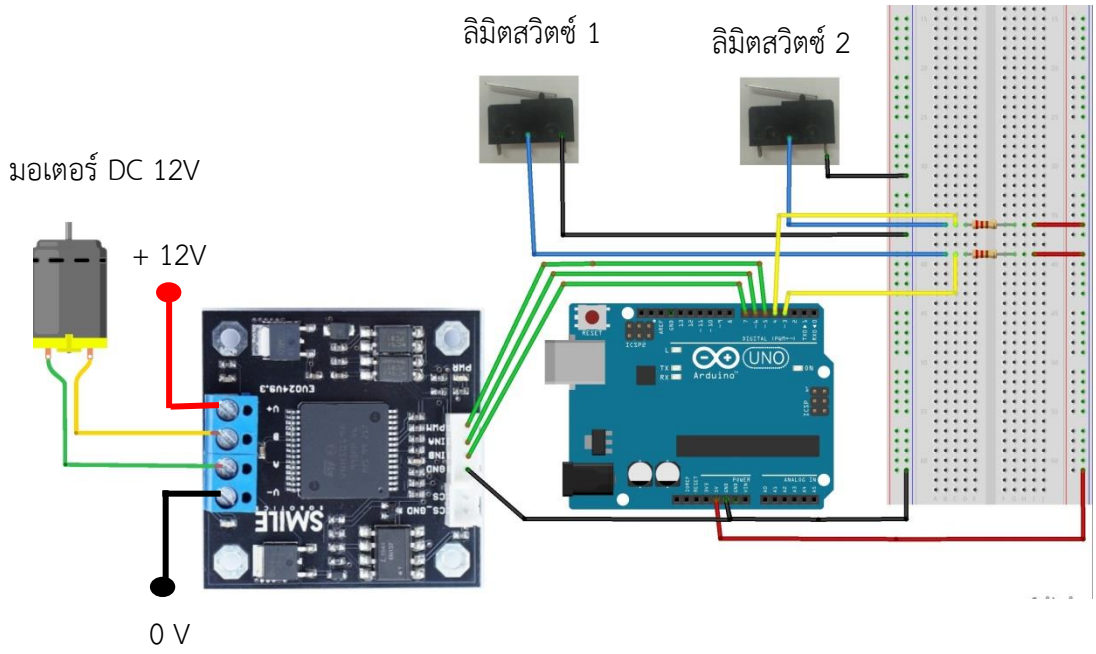
(ที่มา http://www.myarduino.net/article/25/)

โปรแกรมที่ 5.7

```
int senserPin =A0;
int ledPin = 9;
int ledValue;
void setup(){
pinMode(ledPin,OUTPUT);
Serial.begin(9600);
 }
void loop(){
 senserValue=analogRead(SenserPin);
 ledValue=map(senserValue,0,1023,0,255);
  Serial.println(ledValue);
  delay(100);
 analogWrite(ledPin,ledValue);
  }
```

5.4 การเขียนโปรแกรมควบคุมทิศทางของมอเตอร์

การควบคุมทิศทางการหมุนของมอเตอร์โดยใช้บอร์ด EVO24V9.3 H-Bridge DC Motor Drive Model ซึ่งบอร์ดดังกล่าวเป็นบอร์ดที่สามารถขับมอเตอร์ได้จำนวน 1 ตัว สามารถขับมอเตอร์หมุนซ้ายและหมุนขวาได้ โดยมีโปรแกรมที่ 5.8 คอนโทรลและตัวอย่างวงจรใช้งานดังรูปที่ 5.12 โดยมีลิมิตสวิตซ์เป็นตัวเซนเซอร์รับค่าสัญญาณอินพุตให้แก่บอร์ด Arduino เพื่อสั่งงานให้บอร์ด EVO24V9.3 H-Bridge DC Motor ขับมอเตอร์ให้หมุน



รูปที่ 5.12 แสดงการต่อวงจรลิมิตสวิตซ์กับบอร์ด Arduino เพื่อสั่งงานผ่านบอร์ดขับมอเตอร์ให้หยุดและหมุนมอเตอร์

โปรแกรมที่ 5.8

```
int pwm1 = 5;
int sw1 = 3;
int sw2 = 4;
int IA1 = 6;
int IB1 = 7;
boolean reading1;
boolean reading2;
void setup() {
  pinMode(sw1, INPUT);
```

```
    pinMode(sw2, INPUT);
    pinMode(IA1, OUTPUT);
    pinMode(IB1, OUTPUT);
    pinMode(pwm1, OUTPUT);
    digitalWrite(IA1, LOW);
    digitalWrite(IB1, HIGH);
    digitalWrite(pwm1, 28);
   }
void loop() {
   reading1 = digitalRead(sw1); //อ่านค่าจาก limit Sw1
   reading2 = digitalRead(sw2);//อ่านค่าจาก limit  Sw2
   if(reading1==LOW){
     digitalWrite(IA1, LOW);
     digitalWrite(IB1, LOW);
     digitalWrite(pwm1, 28);
     delay(2000);
     digitalWrite(IA1, HIGH);
     digitalWrite(IB1, LOW);
     digitalWrite(pwm1, 28);
     delay(2000);
    }
  if(reading1==HIGH && reading2 == LOW){
     digitalWrite(IA1, LOW);
     digitalWrite(IB1, LOW);
     digitalWrite(pwm1, 28);
     delay(2000);
     digitalWrite(IA1, LOW);
     digitalWrite(IB1, HIGH);
     digitalWrite(pwm1, 28);
     delay(2000);
   }
 }
```

จากโปรแกรมที่ 5.8 การทำงานเมื่อลิมิตสวิตช์1 ถูกกดบอร์ดไมโครคอนโทรลเลอร์ Arduino รับรู้ของสัญญาณอินพุตทำให้โปรแกรมสั่งงานให้ขา IA1 มีสถานะลอจิกเป็น HIGH และ IB1 มีสถานะลอจิกเป็น LOW ทำให้บอร์ด EVO24V9.3 H-Bridge DC Motor ขับมอเตอร์หมุนไปทางซ้าย และหากลิมิตสวิตช์ 2 ถูกกดบอร์ดไมโครคอนโทรลเลอร์ Arduino รับรู้ของสัญญาณอินพุตทำให้โปรแกรมสั่งงานให้ขา IA1 มีสถานะลอจิกเป็น LOW และ IB1 มีสถานะลอจิกเป็น HIGH ทำให้บอร์ด EVO24V9.3 H-Bridge DC Motor ขับมอเตอร์หมุนไปทางขวา

สรุป

PWM เป็นการสั่งงานด้วยสัญญาณ PWM (Pulse width Modulation) ในการควบคุมความเร็ว และใช้สัญญาณลอจิกป้อนให้กับ INA และ INB ในการควบคุมทิศทางการหมุนของมอเตอร์

EVO24V9.3 เป็นบอร์ดขับมอเตอร์แบบชนิดแปรงถ่าน ที่ใช้ไอซี Full bridge motor drive เบอร์ VNH5019 ของ ST Microelectronic โดยออกแบบให้ตัวบอร์ดมีขนาดเล็กน้ำหนักเบา มีความทนทานสูง และการต่อใช้งานง่ายเหมาะสำหรับมอเตอร์ขนาดเล็กไม่เกิน 150 W สามารถขับกระแสต่อเนื่องได้สูงสุด 10 A และขับกระแสชั่วขณะได้ 30 A ที่แรงดันไฟฟ้าสูงสุดที่ 24 V

การควบคุมทิศทางการหมุนของมอเตอร์โดยใช้บอร์ด EVO24V9.3 H-Bridge DC Motor Drive Model ซึ่งบอร์ดดังกล่าวเป็นบอร์ดที่สามารถขับมอเตอร์ได้จำนวน 1 ตัว สามารถขับมอเตอร์หมุนซ้ายและหมุนขวาได้

เอกสารอ้างอิง

เอกชัย มะการ.(2552).เรียนรู้ เข้าใจ ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino. บริษัทอีทีทีจำกัด,กรุงเทพฯ.

การใช้งานบอร์ดร่วมกับ Potentiometor. Motor Drive Module(BTS7960) แก้ไข 2561. สืบค้นจาก : https://www.arduitronics.com/product/983/motor-drive-module-bts7960-43a.

แบบทดสอบหลังเรียน

หน่วยที่ 5 ออกแบบการควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ Arduino ขับเคลื่อนด้วยดีซีมอเตอร์

<u>คำชี้แจง</u> 1. อ่านคำถามต่อไปนี้แล้วทำเครื่องหมายกากบาท (X) ข้อที่ถูกที่สุดลงในกระดาษคำตอบเพียงข้อเดียว

2. เวลาสำหรับทำแบทดสอบ 10 นาที

****

ข้อที่ 1. ปกติค่าความต้านทานที่ใช้วงจร Pull –up หรือ Pull-down จะมีค่าประมาณเท่าใด

ก. 5 - 10 kΩ

ข. 5 – 15 kΩ

ค. 5 - 20 kΩ

ง. 5 – 25 kΩ

จ. 5 - 30 kΩ

ข้อที่ 2. ข้อใดเป็นคำสั่งกำหนดให้พอร์ตเป็นอินพุตอ่านค่าสถานะจาก pin ส่งค่ากลับมาเป็น HIGH หรือ LOW

ก. pinMode(ledpin,OUTPUT);

ข. pinMode(buttonPin,INPUT);

ค.digitalWrite(ledPin,LOW);

ง. digitalRead(ledPin,High);

จ. digitalWrite(ledPin,HIGH);

ข้อที่ 3. ข้อใดคือการอ่านสถานะดิจิตอลและเก็บค่าไว้ที่ตัวแปร

ก. bottonState=digitalRead(buttonPin);

ข. bottonState=anlogRead(buttonPin);

ค. int buttonPin=2;

ง. Boolean buttonState = 0;

จ. int ledPin=13;

ข้อที่ 4. สาเหตุจากการกดสวิตซ์ที่เกิดในช่วงที่กดปุ่มหน้า contact จะไม่สัมผัสแนบสนิทซึ่งจะทำให้เกิดสัญญาณในข้อใด

ก. สัญญาณ Noise

ข. สัญญาณ Pulse

ค. สัญญาณ bounce

ง. สัญญาณ Square

จ. สัญญาณ Sync

ข้อที่ 5. ข้อมูลที่ต้องการส่งผ่านไปให้ฟังก์ชั่นทำงาน ซึ่งต้องกำหนดไว้ในวงเล็บ() ที่อยู่หลังชื่อฟังก์ชั่น คือความหมายของข้อใด

ก. ค่าที่ส่งให้ฟังก์ชั่น

ข. รูปแบบการส่งค่ากลับ

ค. ชื่อฟังก์ชั่น

ง. ตัวแปร

จ. ค่าที่ส่งคืนกลับ

ข้อที่ 6. ค่าข้อมูลที่ต้องการส่งกลับไปยังฟังก์ชั่นผู้เรียก โดยในการส่งค่าคืนกลับไปยังฟังก์ชั่นผู้เรียกนั้นจะใช้คำสั่งในข้อใด

ก. Goto

ข. setMode

ค. pinMode()

ง. void

จ. return

ข้อที่ 7. int sensorValue=analogRead(SENSOR_PIN); มีความหมายหมายตรงกับข้อใด

ก. อ่านค่าแอนาล็อกจาก SENSOR_PIN เก็บค่าไว้ในตัวแปร sesorValue

ข. อ่านค่าดิจิตอลจาก SENSOR_PIN เก็บค่าไว้ในตัวแปร sesorValue

ค. ค่าตัวแปร sensorValueเท่ากับค่า analogRead

ง. ค่าตัวเลขใน sensorValueเท่าตัวเลขanlogRead

จ. ค่าตัวอักษรใน sensorValueเท่าตัวเลขanlogRead

ข้อที่ 8. กำหนดเงื่อนไขให้ reading1 = digitalRead(sw1); อธิบายความหมายของโปรแกรมตรงกับข้อได้ถูกที่สุด

ก. reading1 เท่ากับการอ่านค่าสถานะทางดิจิตอลของ sw1

ข. reading1 เท่ากับการอ่านค่าสถานะทางแอนะล็อกของ sw1

ค. reading1 เท่ากับ sw1

ง. reading1 มีค่าไม่เท่ากับ sw1

จ. reading1 มีค่าทางแรงดันไฟฟ้าของsw1

ข้อที่ 9. เลือกคำตอบข้อใดคือคำตอบที่เติมลงในช่องเพื่อให้มอเตอร์หมุน

```
if(reading1==HIGH && reading2 == LOW){
   digitalWrite(IA1, LOW);
   digitalWrite(IB1, HIGH);
   .....
   delay(2000);
  }
```

ก. digitalWrite(IB1, LOW);

ข. digitalWrite(pwm1, 28);

ค. digitalWrite(IA1, LOW);

ง. reading1 = digitalRead(sw1);

จ. delay(2000);

ข้อที่ 10. การกำหนดค่าตัวแปรข้อใดที่กำหนดให้มีค่าตัวแปรแบบบลูลีนที่เป็นเท็จหรือจริง

ก. int pwm1 = 5;

ข. pinMode(sw1, INPUT);

ค. pinMode(pwm1, OUTPUT);

ง. digitalWrite(IB1, HIGH);

จ. boolean reading1;

เฉลยแบบประเมินผล

หน่วยที่ 5 ออกแบบการควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ Arduino ขับเคลื่อนด้วยดีซีมอเตอร์

ก่อนเรียน		หลังเรียน	
ข้อที่	คำตอบ	ข้อที่	คำตอบ
1.	ข	1.	ค
2.	ง	2.	ข
3.	ก	3.	ก
4.	ค	4.	ค
5.	ง	5.	ก
6.	ง	6.	จ
7.	ง	7.	ก
8.	ง	8.	ก
9.	ก	9.	ข
10.	ง	10.	จ