

หน่วยที่ 8 การควบคุมดีซีมอเตอร์และสเตปมอเตอร์ด้วย ARDUINO

สาระสำคัญ

มอเตอร์ไฟฟ้าเป็นอุปกรณ์ที่นิยมใช้กันอย่างแพร่หลายในโรงงานอุตสาหกรรม เป็นอุปกรณ์ที่ใช้ควบคุมเครื่องจักรกลในงานอุตสาหกรรม มอเตอร์มีหลายแบบหลายชนิดที่ใช้ให้เหมาะสมกับงาน ดังนั้นเราจึงต้องทราบถึงความหมายและชนิดของมอเตอร์ไฟฟ้า ตลอดจนคุณสมบัติการใช้งานของมอเตอร์แต่ละชนิด เพื่อให้เกิดประสิทธิภาพสูงสุดในการใช้งานของมอเตอร์นั้นๆ มอเตอร์มอเตอร์ไฟฟ้า (Motor) หมายถึงเครื่องกลไฟฟ้าชนิดหนึ่ง ที่เปลี่ยนแปลงพลังงานไฟฟ้ามาเป็นพลังงานกล

เนื้อหาสาระการเรียนรู้

- 8.1 ความรู้เบื้องต้นเกี่ยวกับมอเตอร์ไฟฟ้า
- 8.2 การควบคุมดีซีมอเตอร์ด้วย Arduino
- 8.3 การควบคุมสเตปมอเตอร์ด้วย Arduino

จุดประสงค์การเรียนรู้

จุดประสงค์ทั่วไป

1. เพื่อให้มีความรู้ความเข้าใจเกี่ยวกับการควบคุมดีซีมอเตอร์และสเตปมอเตอร์ด้วย Arduino
2. เพื่อให้สามารถนำความรู้ไปประยุกต์ใช้ในการเขียนโปรแกรมกำหนดการทำงานด้วย Arduino
3. เพื่อให้ตระหนักถึงความสำคัญของการควบคุมดีซีมอเตอร์และสเตปมอเตอร์ด้วย Arduino

จุดประสงค์เชิงพฤติกรรม

1. อธิบายความรู้เบื้องต้นเกี่ยวกับมอเตอร์ไฟฟ้าได้
2. เชื่อมต่อดีซีมอเตอร์ด้วย Arduino ได้
3. ควบคุมดีซีมอเตอร์ด้วย Arduino ได้
4. เชื่อมต่อสเตปมอเตอร์ด้วย Arduino ได้
5. ควบคุมสเตปมอเตอร์ด้วย Arduino ได้

แบบทดสอบก่อนเรียน หน่วยที่ 8

เรื่อง การควบคุมดีซีมอเตอร์และสเตปมอเตอร์ด้วย Arduino

เรื่อง การควบคุมดีซีและสเตปมอเตอร์ด้วย Arduino

ใช้เวลา 20 นาที

วิชา ไมโครคอนโทรลเลอร์เบื้องต้น

รหัสวิชา (2127-2107)

ระดับชั้น ประกาศนียบัตรวิชาชีพ (ปวช.)

สาขาวิชา เมคคาทรอนิกส์

คำชี้แจง 1. แบบทดสอบมีทั้งหมด 10 ข้อ (10 คะแนน)

2. ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ลงในกระดาษคำตอบ

1. มอเตอร์มอเตอร์ไฟฟ้า (Motor) หมายถึง

- ก. เครื่องกลไฟฟ้าที่เปลี่ยนแปลงพลังงานกลเป็นพลังงานไฟฟ้า
- ข. เครื่องกลไฟฟ้าที่เปลี่ยนแปลงพลังงานไฟฟ้าเป็นพลังงานแสง
- ค. เครื่องกลไฟฟ้าที่เปลี่ยนแปลงพลังงานกลเป็นพลังงานแสง
- ง. เครื่องกลไฟฟ้าที่เปลี่ยนแปลงพลังงานไฟฟ้าเป็นพลังงานกล

2. มอเตอร์ไฟฟ้ากระแสตรงเป็นต้นกำลังขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรมเพราะ

- ก. ทนทาน
- ข. ราคาถูก
- ค. ปรับความเร็วได้
- ง. ประหยัดไฟฟ้า

3. กระแสไฟฟ้าในขดลวดสนามแม่เหล็ก (Field Coil) จะสร้างขั้วแม่เหล็กแบบใด

- ก. ขั้วสูง-ต่ำ
- ข. ขั้วเหนือ-ใต้
- ค. ขั้วซ้าย-ขวา
- ง. ขั้วบน-ล่าง

4. DC Motor ประกอบด้วย 2 ส่วนหลักๆ ได้แก่

- ก. ขั้วแม่เหล็กเหนือ / ใต้
- ข. พุนอาร์มาเจอร์ / สเตเตอร์
- ค. มอเตอร์ไฟฟ้ากระแสสลับ / มอเตอร์ไฟฟ้ากระแสตรง
- ง. ขดลวดอาร์มาเจอร์ / ขดลวดสนามแม่เหล็ก

5. หลักการทำงานของ DC Motor คือ

- ก. แรงดันสลับตัดผ่านสนามแม่เหล็ก เกิดทอร์กขึ้นในทิศที่เหมาะสม
- ข. กระแสสลับตัดผ่านสนามแม่เหล็ก เกิดทอร์กขึ้นในทิศที่เหมาะสม
- ค. แรงดันสลับตัดผ่านสนามแม่เหล็ก เกิดทอร์กขึ้นในทิศที่เหมาะสม
- ง. กระแสตรงตัดผ่านสนามแม่เหล็ก เกิดทอร์กขึ้นในทิศที่เหมาะสม

6. วงจร H-Bridge สามารถควบคุมมอเตอร์ได้อย่างไร

- ก. ควบคุมความเร็วรอบของมอเตอร์
- ข. ควบคุมทิศทางของมอเตอร์
- ค. ควบคุมทิศทาง และความเร็วรอบของมอเตอร์
- ง. ควบคุมการเดินหน้าและถอยหลังของมอเตอร์

7. หากต้องการให้หมุนตามเข็มนาฬิกา (Clockwise :CW) ให้สวิตช์ทำงานอย่างไร

- ก. S2 และ S3 ปิดวงจร และ S1 และ S4 เปิดวงจร
- ข. S1 และ S4 ปิดวงจร
- ค. S2 และ S3 เปิดวงจร
- ง. S1 และ S4 ปิดวงจร และ S2 และ S3 เปิดวงจร

8. อุปกรณ์สารกึ่งตัวนำที่นิยมใช้ในวงจร H-Bridge คือ

- ก. DIODE หรือ ZENERDIODE
- ข. TRIAC หรือ DIAC
- ค. TRANSISTER
- ง. MOSFET หรือ IGBT

9. หลักการของการทำงานของ Stepper Motor คือ

- ก. การมอดูเลตสัญญาณเฟสเซอร์
- ข. การมอดูเลตสัญญาณตามแอมพลิฟายด์
- ค. การมอดูเลตสัญญาณตามความถี่
- ง. การจ่ายไฟเข้าไปที่ละขดตามลำดับ

10. วงจร Darlington เป็น IC สำเร็จรูปมีชื่อเรียกว่า

- ก. L298P
- ข. 7812
- ค. 1N4001
- ง. ULN2003AN

หน่วยที่ 8

การควบคุมดีซีมอเตอร์และสเตปมอเตอร์ด้วย Arduino

8.1 ความรู้เบื้องต้นเกี่ยวกับมอเตอร์ไฟฟ้า

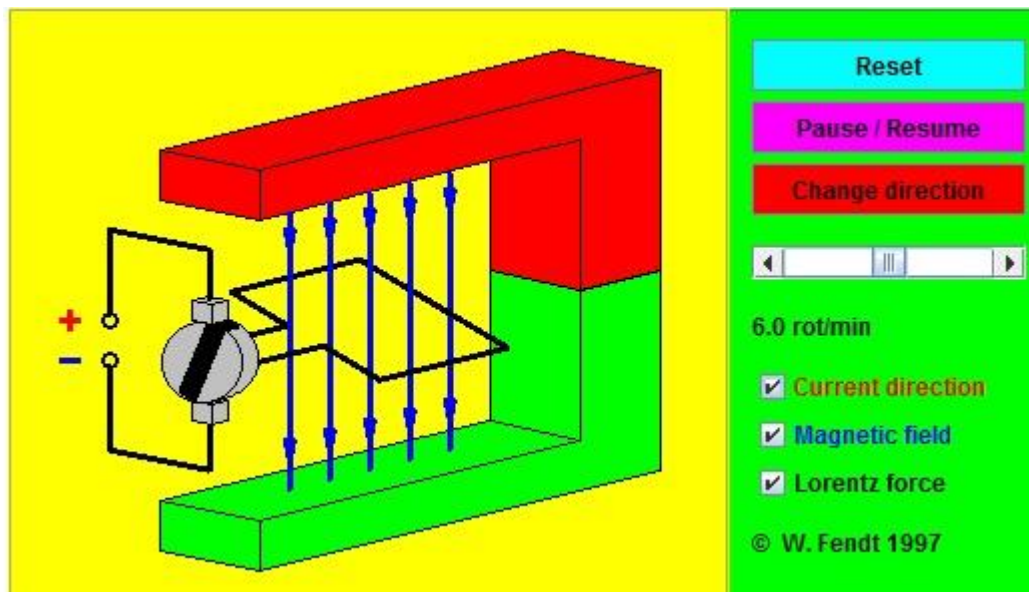
มอเตอร์ไฟฟ้าเป็นอุปกรณ์ที่นิยมใช้กันอย่างแพร่หลายในโรงงาน เป็นอุปกรณ์ที่ใช้ควบคุมเครื่องจักรกลในงานอุตสาหกรรม มอเตอร์มีหลายแบบหลายชนิดที่ใช้ให้เหมาะสมกับงาน ดังนั้นเราจึงต้องทราบถึงความหมายและชนิดของมอเตอร์ไฟฟ้า ตลอดจนคุณสมบัติการใช้งานของมอเตอร์แต่ละชนิด เพื่อให้เกิดประสิทธิภาพสูงสุดในการใช้งานของมอเตอร์นั้นๆ

มอเตอร์มอเตอร์ไฟฟ้า (Motor) หมายถึงเครื่องกลไฟฟ้าชนิดหนึ่งที่เปลี่ยนแปลงพลังงานไฟฟ้ามาเป็นพลังงานกล มอเตอร์ไฟฟ้าที่ใช้พลังงานไฟฟ้าเปลี่ยนเป็นพลังงานกล มีทั้งพลังงานไฟฟ้ากระแสสลับ และพลังงานไฟฟ้ากระแสตรง

8.1.1 มอเตอร์ไฟฟ้าแบ่งออกตามชนิดของกระแสไฟฟ้า ได้ 2 ชนิดดังนี้

- มอเตอร์ไฟฟ้ากระแสสลับ(Alternating Current Motor)
 1. มอเตอร์ไฟฟ้ากระแสสลับชนิด 1 เฟส
 - สปลิตเฟสมอเตอร์ (Split-Phase motor)
 - คาปาซิเตอร์มอเตอร์ (Capacitor motor)
 - รีพลักซ์ชันมอเตอร์ (Repulsion-type motor)
 - ยูนิเวอร์แซลมอเตอร์ (Universal motor)
 - เชดเดดโพลมอเตอร์ (Shaded-pole motor)
 2. มอเตอร์ไฟฟ้ากระแสสลับชนิด 2 เฟส
 3. มอเตอร์ไฟฟ้ากระแสสลับชนิด 3 เฟส
- มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor) แบ่งออกเป็น 3 ชนิดได้แก่
 1. มอเตอร์แบบอนุกรมหรือเรียกว่าซีรีย์มอเตอร์ (Series Motor)
 2. มอเตอร์แบบขนานหรือเรียกว่าชันทมอเตอร์ (Shunt Motor)
 3. มอเตอร์ไฟฟ้าแบบผสมหรือเรียกว่าคอมปาวด์มอเตอร์ (Compound Motor)

มอเตอร์ไฟฟ้ากระแสตรงเป็นต้นกำลังขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรมเพราะมีคุณสมบัติที่ดีเด่นในด้านการปรับความเร็วได้ ตั้งแต่ความเร็วต่ำสุดจนถึงสูงสุด นิยมใช้กันมากในโรงงานอุตสาหกรรม

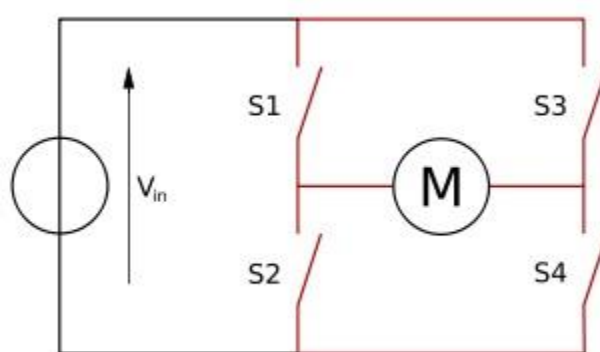


รูปที่ 8.2 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

(ที่มา www.Walter-fendt.de/ph14e/electricmotor.htm myarduino.net)

8.1.3 การควบคุมทิศทาง และความเร็วยรอบของมอเตอร์

การควบคุมทิศทาง และความเร็วยรอบของมอเตอร์ ไม่ว่าจะเป็นการขับเคลื่อนสายพาน หรือล้อของหุ่นยนต์ไม่ให้หมุนแบบอิสระและสามารถควบคุมได้ ดังนั้นจึงมีวงจรที่ใช้ในการควบคุมมอเตอร์ขึ้นมา แบบที่นิยมใช้กันเรียกว่าวงจร “H-Bridge”



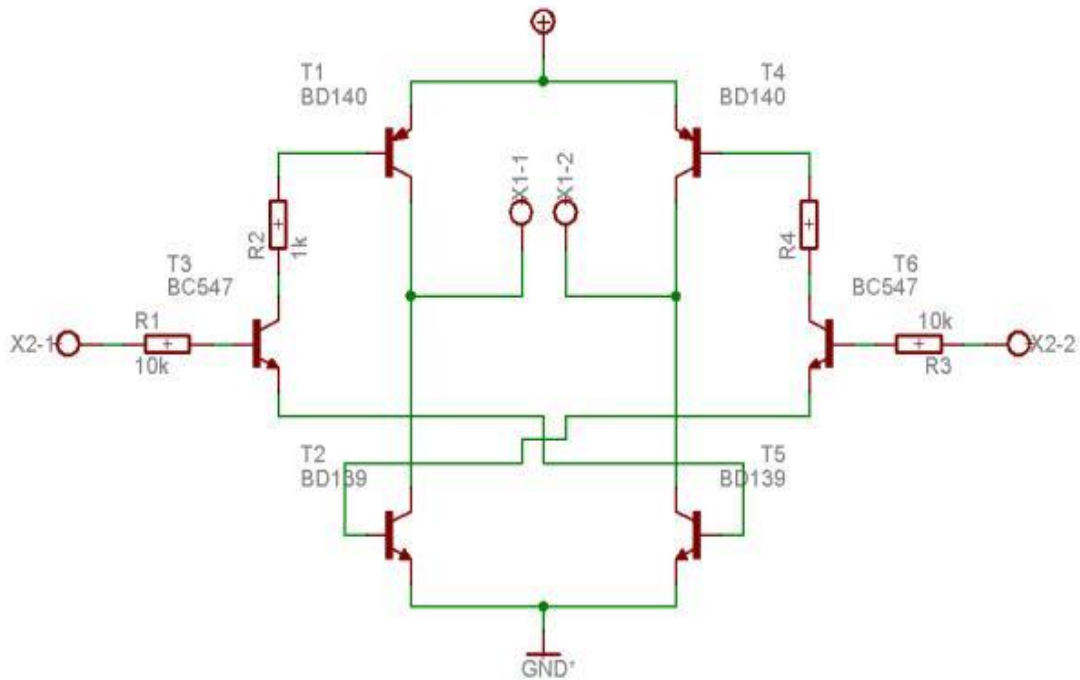
รูปที่ 8.3 การควบคุมมอเตอร์ H-Bridge

(ที่มา www.Arduitrronics.com)

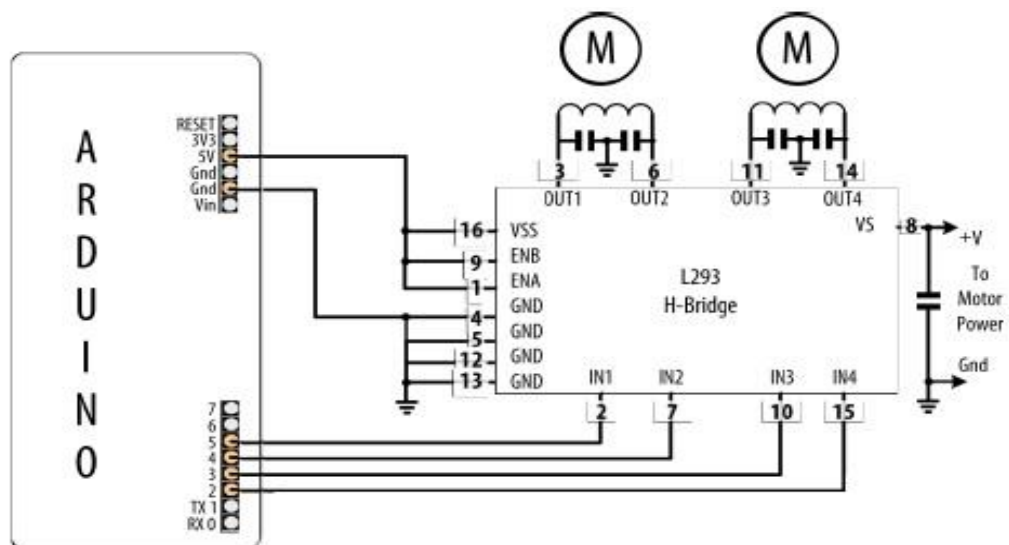
วงจรนี้ทำหน้าที่ได้ทั้งคุมทิศทางและความเร็วของมอเตอร์ โดยเริ่มจากการคุมทิศทางการหมุน โดยปกติหากต้องการกลับทิศการหมุนของมอเตอร์กระแสตรง วิธีหนึ่งที่ได้คือกลับทิศแหล่งจ่ายวงจร H-Bridge ด้านบน

หากต้องการให้หมุนตามเข็มนาฬิกา (Clockwise :CW) ให้ S1 และ S4 ปิดวงจร และให้ S2 และ S3 เปิดวงจร หากต้องการให้หมุนทวนเข็มนาฬิกา (Conter Clockwise :CCW) ให้ S2 และ S3 ปิดวงจร และให้ S1 และ S4 เปิดวงจร

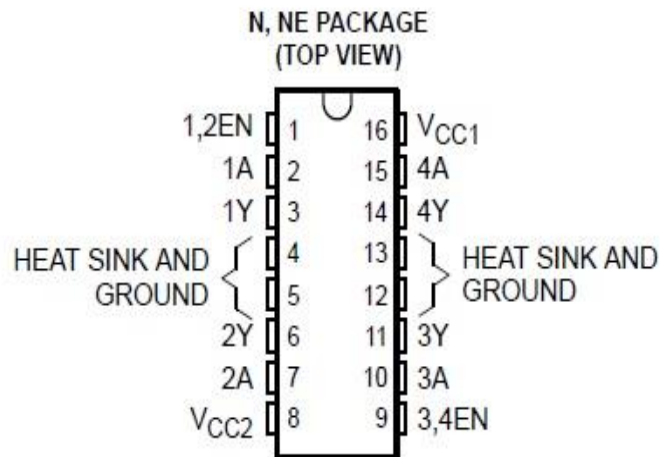
สังเกตว่าสวิตช์จะทำงานเป็นคู่ คู่แรกทำงาน คู่สองต้องเปิดวงจร และในทางตรงข้ามก็คือคู่สองทำงาน คู่แรกต้องเปิดวงจร ต่อมาทำให้การเปิดปิดเป็นแบบที่ง่ายกว่าเดิม โดยใช้อุปกรณ์สารกึ่งตัวนำเช่น MOSFET หรือ IGBT หรืออื่นๆ แล้วแต่ความเหมาะสม เช่นขนาดกระแสแรงดันที่ต้องการควบคุม



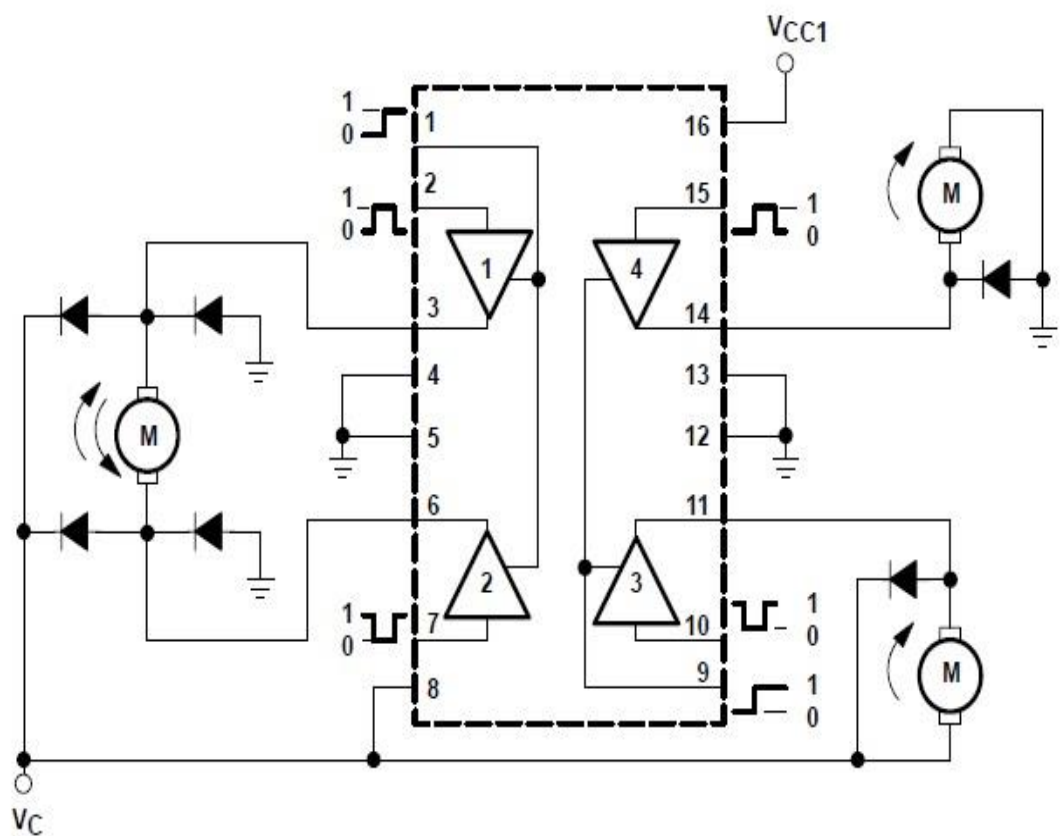
รูปที่ 8.4 Schematic ของวงจร H-Bridge
(ที่มา www.Arduitrronics.com)



รูปที่ 8.5 การต่อบอร์ด Arduino ควบคุม H-Bridge
(ที่มา www.Arduitrronics.com)



block diagram



NOTE: Output diodes are internal in L293D.

รูปที่ 8.6 การต่อไอซี L293D ควบคุมดีซีมอเตอร์
(ที่มา www.Arduitrronics.com)

โปรแกรมที่ 8.1 การต่อบอร์ด Arduino ควบคุม H-Bridge

```
const int in1Pin = 5; // H-Bridge input pins
const int in2Pin = 4;
const int in3Pin = 3; // H-Bridge pins for second motor
const int in4Pin = 2;

void setup(){
  Serial.begin(9600);
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
  pinMode(in3Pin, OUTPUT);
  pinMode(in4Pin, OUTPUT);
  Serial.println("+ - sets direction of motors, any other key stops motors");
}

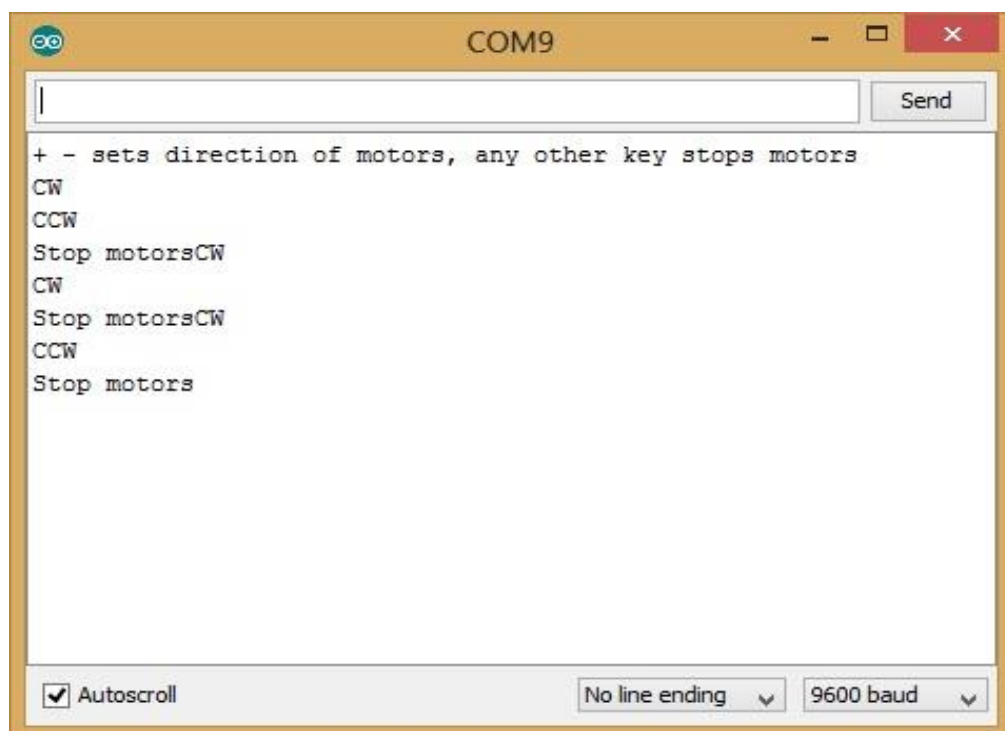
void loop(){
  if ( Serial.available() ) {
    char ch = Serial.read();
    if (ch == '+')
    {
      Serial.println("CW");
      // first motor
      digitalWrite(in1Pin,LOW);
      digitalWrite(in2Pin,HIGH);
      //second motor
      digitalWrite(in3Pin,LOW);
      digitalWrite(in4Pin,HIGH);
    }
    else if (ch == '-')
    {
      Serial.println("CCW");
      digitalWrite(in1Pin,HIGH);
      digitalWrite(in2Pin,LOW);
      digitalWrite(in3Pin,HIGH);
    }
  }
}
```

```

        digitalWrite(in4Pin,LOW);
    }
else
{
    Serial.print("Stop motors");
    digitalWrite(in1Pin,LOW);
    digitalWrite(in2Pin,LOW);
    digitalWrite(in3Pin,LOW);
    digitalWrite(in4Pin,LOW);
}
}
}
}

```

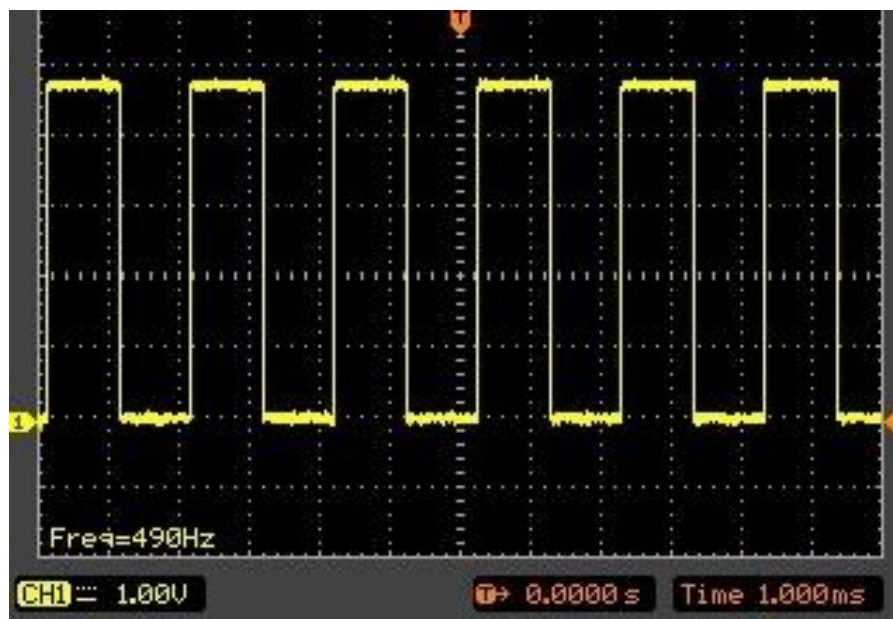
เมื่ออัปโหลดโปรแกรมเสร็จแล้วทุกอย่างถูกต้อง หน้าจอแสดงข้อความ Serial Monitor โดยจะเห็นมอเตอร์หมุนทิศทางหนึ่งตอนกด "+" และทิศตรงข้ามตอนกด "-" การกดอย่างอื่นจะหยุดหมุน



รูปที่ 8.7 Serial Monitor
(ที่มา www.Arduitrronics.com)

8.2 การควบคุมติ้ชี่มอเตอร้ด้วย Arduino

การควบคุมความเร็วรอบของ DC Motor ด้วยเทคนิค Pulse Width Modulation (PWM)



รูปที่ 8.8 ออสซิลอโคปแสดงสัญญาณ PWM
(ที่มา www.Arduitrronics.com)

PWM (Pulse Width Modulation) คือ การมอดูเลตสัญญาณให้มีความกว้างตามสัดส่วนที่กำหนด บนความถี่ Carrier ที่ต้องการใช้งาน ซึ่งโดยปกติแล้วจะใช้ประโยชน์ในการควบคุมการเปิดปิดของวงจรอิเล็กทรอนิกส์กำลัง เช่น วงจรบัค วงจรบูส วงจรบัคบูส เป็นต้น

นอกจากนี้ยังยังสามารถใช้ประกอบกับวงจร H - Bridge เพื่อควบคุมความเร็วรอบของมอเตอร์ หรือว่าวงจรพื้นฐานเช่นต้องการจะหรี่หลอด LED ก็ได้อีกด้วย พารามิเตอร์ที่ใช้ระบุรูปร่างหน้าตาของ PWM ที่สำคัญมี 2 ค่าด้วยกันคือ ความถี่ของคลื่นพาหะ (Carrier Frequency) และ อัตราส่วนหน้าที่ (Duty Ratio)

Carrier Frequency ใน Arduino มีความถี่ประมาณ 490 Hz ในกรณีที่ใช้ Library ปกติใน Arduino IDE และยังสามารถปรับให้มีความถี่สูงขึ้นเป็นค่าอื่นๆ เช่น 31.25 kHz ได้อีกด้วย

Duty Ratio คือ สัดส่วนของเวลาที่จ่ายแรงดันต่อคาบของ Carrier Frequency ซึ่งกรณีให้เห็นในรูปที่ 8.8 คือเวลาที่จ่ายแรงดัน 1.02 มิลลิวินาที ต่อคาบ 2.04 มิลลิวินาที ซึ่งจะได้ Duty Ratio = 50%

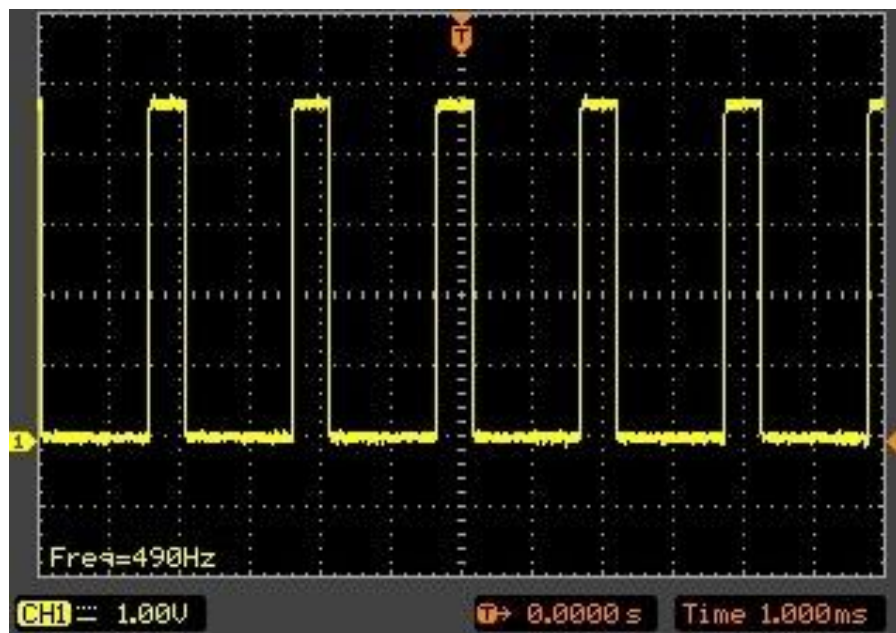
โปรแกรมที่ 8.3 การให้แรงดัน PWM ออกที่ขา 3 โดยมี Duty Ratio = 50%

```
int ledPin = 3;    // LED connected to digital pin 9
int val = 0;      // variable to store the read value

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the pin as output
}

void loop()
{
  analogWrite(3, 128);
  // analogRead values go from 0 to 1023, analogWrite values from 0 to 255
}
```

รูปที่ 8.9 แสดงให้เห็นว่ามี Carrier Frequency เท่าเดิม เพราะยังใช้ PWM จาก Library ปกติ ต่อไปปรับให้เวลาที่เปิด เป็น 0.51 มิลลิวินาที ซึ่งจะได้ Duty Ratio = 25%

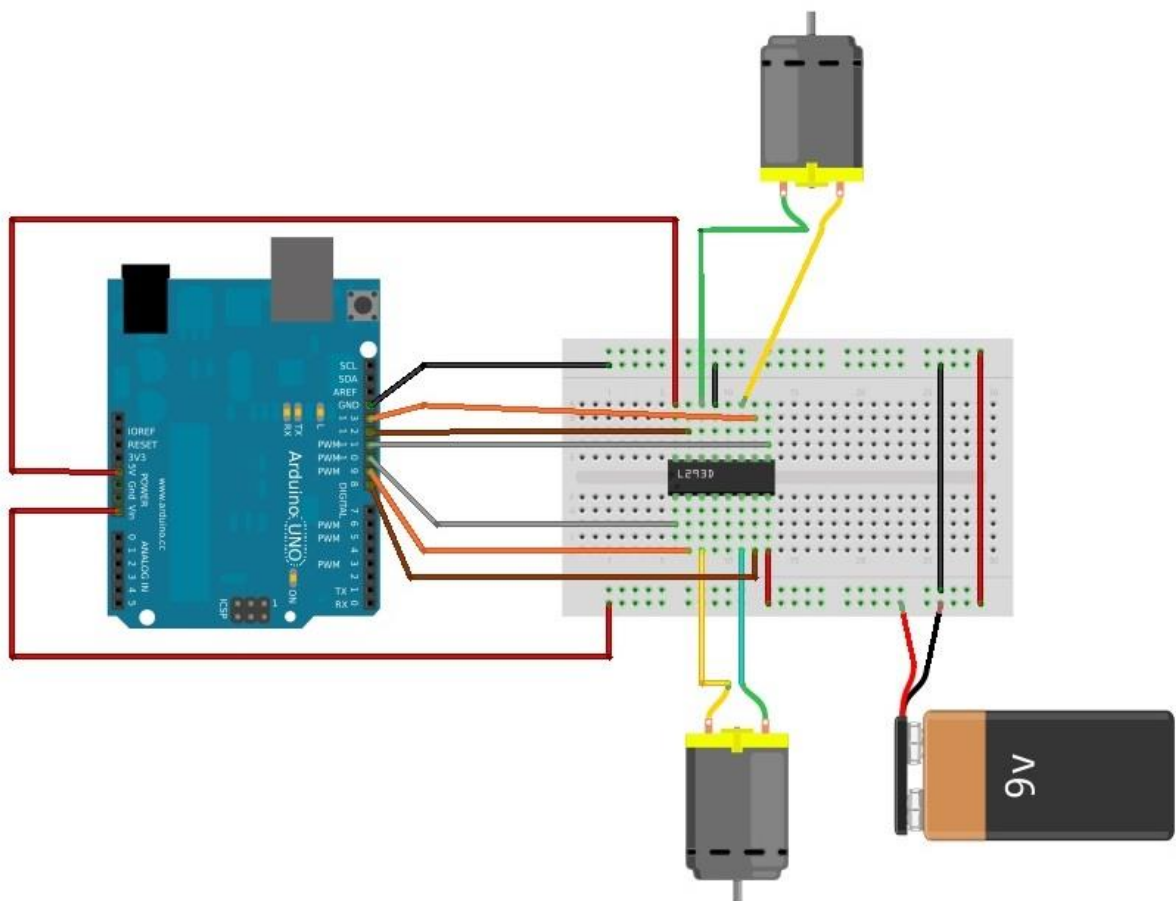


รูปที่ 8.9 สัญญา PWM ความถี่ Carrier ที่ 490 Hz
(ที่มา www.Arduitrronics.com)

พิจารณารูปที่ 8.9 กำหนดให้ PWM Port 3 ในการกำเนิดสัญญาณนี้ โดยใช้คำสั่ง analogWrite

(3, 128) ซึ่งหมายถึงการเขียนให้ออก Port 3 โดยมีสัดส่วน Duty Ratio คือ 128/256 หรือ 50% นั่นเอง การใช้คำสั่ง analogWrite แต่สัญญาณไปออกขา Digital(PWM) ก็เป็นเพราะ Arduino Board อย่าง UNO ไม่มี Analog Output channel มีแต่ PWM Output ที่ Pin ด้าน Digital

สัญญาณ PWM ถึงแม้จะมีความถี่ Carrier ที่ 490 Hz สามารถใช้ในการควบคุมมอเตอร์กระแสตรงโดยใช้วงจร H-Bridge โดย PWM ใช้ได้กับ IC เบอร์ L298P และ L293 หรือ Motor Shield นอกจากนี้ถ้าจะต้องวงจรสร้าง H-Bridge กรณีที่มอเตอร์มีขนาดใหญ่กว่า 2 A โดยต่อเข้ากับขา INPUT ของ IC ที่ใช้และต่อกราวด์ของ Arduino กับแหล่งจ่ายแรงดันหลักของมอเตอร์



รูปที่ 8.10 วงจร H-Bridge โดย PWM ใช้กับ IC เบอร์ L293D

(ที่มา www.Parduitronics.com)

อีกหนึ่งรูปแบบที่มีการประยุกต์ใช้งานคือวงจร Switching Power Supply คือการปรับให้ Arduino กำเนิดสัญญาณ PWM แบบความถี่สูงๆ เช่น 31.25 kHz การใช้งาน Arduino ในส่วนนี้ต้องเข้าไปแก้ไข Register จำนวน 4 ตัวใน Atmel 328 ที่ใช้เป็นตัวไมโครคอนโทรลเลอร์ของ Arduino UNO ใช้สำหรับการทำงานกับวงจร Power Electronics เพราะความถี่ 490 Hz ไม่เพียงพอสำหรับการทำ Switching Power Supply แบบต่างๆ ก็เพราะว่าขนาดของ Capacitor กับ Inductor ในวงจรเพื่อใช้ Filter ความถี่จะต้องมีขนาดใหญ่มาก

โปรแกรมที่ 8.9 เป็นการสร้าง PWM ที่ Carrier Frequency 31.25 kHz โดยมี Duty Ratio = 50% หรือ

กำหนดให้มีค่า 128 ในโปรแกรมที่ 8.4 นั้นเอง และให้มีสัญญาณออกที่ขา 11

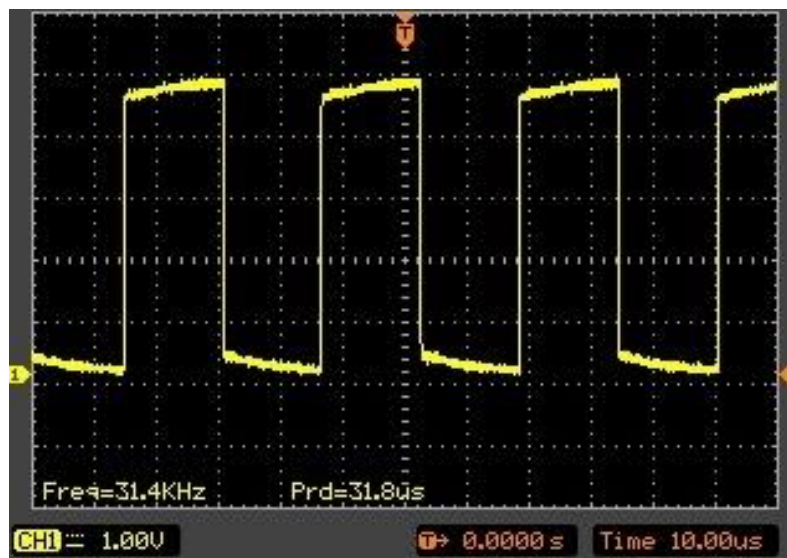
โปรแกรมที่ 8.4

การสร้าง PWM ที่ Carrier Frequency 31.25 kHz โดยมี Duty Ratio = 50%

```
void setup(){
    pwmSetup();
}

void loop(){
    int Duty = 128; //type in the voltage you want. Refer to guide on limiting
    OCR2A = Duty;
}

void pwmSetup(){ //just run once at setup
    pinMode(11, OUTPUT); //OCR2A
    TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM20);
    //phase correct pwm 31250hz
    TCCR2B = _BV(CS20);
    //change this as datasheet says to mainly get different pwm frequencies
    OCR2A = 0;
    OCR2B = 0;}
```

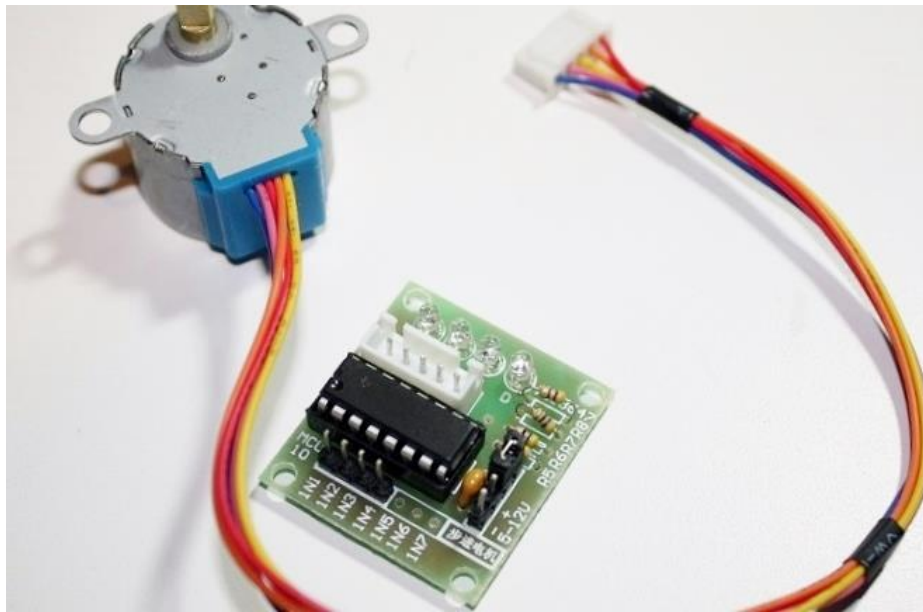


รูปที่ 8.11 สัญญาณจากออสซิลโลสโคปมีความถี่ 31.25 kHz
(ที่มาจาก www.Arduitrronics.com)

สังเกตได้จากรูปที่ 8.11 สัญญาณจากออสซิลโลสโคปมีความถี่ 31.25 kHz (อ่านได้ประมาณ 31.4 kHz) โดยจะเห็นว่า Duty ratio 50% ตามที่ตั้งไว้เป็น 128 ตามต้องการ

8.3 การควบคุมสเตปมอเตอร์ด้วย Arduino

ในหัวข้อที่ 8.1 ได้นำเสนอการควบคุม DC Motor สาเหตุที่ต้องศึกษา DC Motor ก่อนก็เพราะว่าการทำงานของมอเตอร์ชนิดอื่นๆ เช่น Stepper Motor จะมีพื้นฐานมาจาก DC Motor ในบางด้านแต่ Stepper Motor จะมีความซับซ้อนกว่าทั้งหลักการทำงานและการควบคุม

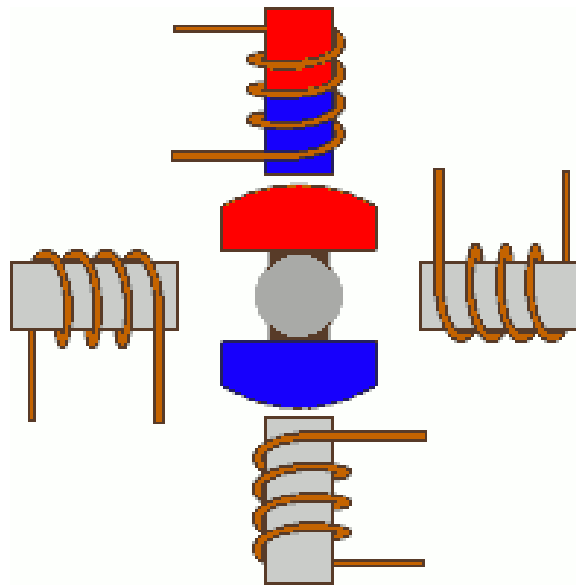


รูปที่ 8.12 บอร์ดขับสเตปมอเตอร์
(ที่มา www.Arduitrronics.com)

Stepper Motor หรือ Stepping Motor มีข้อดีที่สำคัญคือการควบคุมตำแหน่งของการหมุนได้อย่างแม่นยำ โดยไม่ต้องใช้การควบคุมแบบป้อนกลับ (Feedback Control) ด้วยเหตุนี้ จึงเป็นที่นิยมใช้ในอุปกรณ์ที่ต้องการควบคุมตำแหน่งและมุมอย่างแม่นยำ เช่น พริ้นเตอร์ สแกนเนอร์ เครื่องเล่นแผ่นดิสก์ เป็นต้น

8.3.1 หลักการทำงานของ Stepper Motor

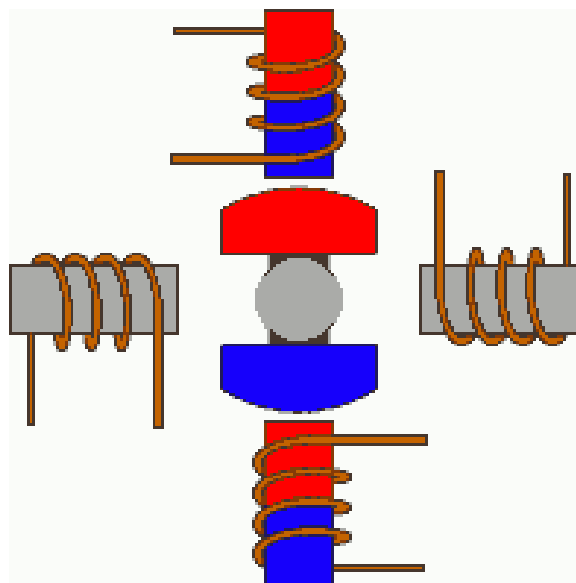
หลักการทำงานของ Stepper Motor คือการบังคับให้แม่เหล็กถาวรบนแกนโรเตอร์ หมุนไปตามทิศการบังคับของขดลวดที่ติดตั้งบนสเตเตอร์ จะซับซ้อนกว่า DC Motor ตรงที่การบังคับให้หมุนนั้นไม่ได้เป็นแค่การใส่แรงดันคงที่ไปที่ขั้วขั้วกลับเท่านั้น ต้องใส่แรงดันให้ถูกต้องจากจังหวะที่ควรจะเป็นดังรูปที่ 8.12 ถึงจะหมุนได้



รูปที่ 8.13 การจ่ายกระแสไฟฟ้าเข้าไปที่ขดลวดทีละขด

(ที่มา www.Pcbheaven.com/wikipages/How_Stepper_Motors_Work/)

จะเห็นว่ามีขดลวดที่ควบคุมการหมุน โดยแต่ละขดห่างกัน 90 องศา การหมุนก็จะทำโดยการจ่ายกระแสเข้าไปที่ขดลวดทีละขด เพื่อทำให้เกิดสนามแม่เหล็ก ซึ่งจะดึงดูดให้แม่เหล็กถาวรที่อยู่บนโรเตอร์เคลื่อนที่ โดยทิศของการหมุนก็จะขึ้นกับลำดับการจ่ายกระแสเข้าไปที่ขดลวด โดยการบังคับในลักษณะนี้เรียกว่า Single coil excitation หรือการกระตุ้นทีละขดลวด โดยจะมีการกระตุ้นหรือการจ่ายกระแสเข้าขดลวดอยู่ 4 จังหวะต่อการหมุน 1 รอบ อีกรูปแบบหนึ่งจะซับซ้อนกว่าแต่จะให้ทอร์คมากกว่า คือการป้อนแบบ Full Step Drive หรือการป้อนแบบทีละ 2 ขดลวด



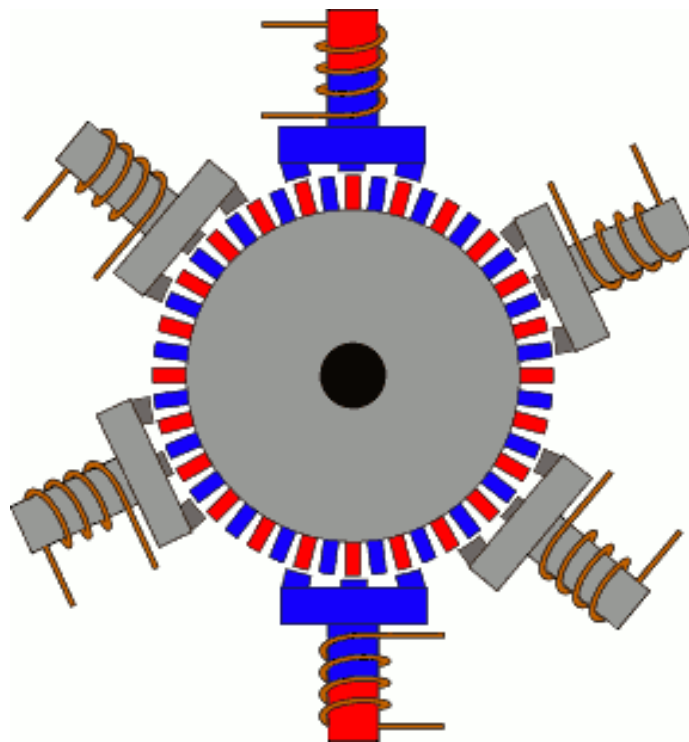
รูปที่ 8.14 การจ่ายกระแสไฟฟ้าเข้าไปที่ขดลวดทีละสองขด

(ที่มา www.Pcbheaven.com/wikipages/How_Stepper_Motors_Work/)

ถ้าต้องการควบคุมให้มีความละเอียดมากขึ้นอีก สามารถทำได้โดยสามารถคุมให้มอเตอร์หมุนได้ละเอียดขึ้น จาก 90 องศา เหลือ 45 องศา โดยที่ไม่ต้องปรับเปลี่ยนตัวฮาร์ดแวร์ใดๆ เป็นเพียงการเปลี่ยนวิธีการจ่ายกระแสเข้า ขดลวดเท่านั้น

แต่ Stepper Motor ที่ใช้จริงมีการพัฒนาต่อ โดยเพิ่มจำนวนขดลวดและปรับให้แม่เหล็กถาวรมีซี่ (Teeth) ซึ่งทำหน้าที่เป็นจำนวนซี่ของแม่เหล็กมากขึ้น การเพิ่มจำนวนซี่โดยแต่ละซี่เป็นซี่เหนือและซี่ใต้สลับกันนั้น สามารถทำได้โดยเอาเหล็กที่มีสภาพความนำแม่เหล็กมาขึ้นรูปเป็นจานที่มีซี่อยู่รอบๆ จากนั้นขั้นตอนการควบคุมก็มี หลักการเดียวกับ Stepper Motor ที่หมุนได้ที่ละ 90 องศา

เนื่องจาก Stepper Motor ที่ใช้กันจริงๆ มีซี่และขดลวดมาก จึงทำให้สามารถควบคุมการหมุนได้ละเอียด มากๆ โดยการควบคุม 1 Cycle จะทำให้มอเตอร์หมุนไป 0.9 - 5 องศา แล้วแต่เทคนิคที่ใช้ในการควบคุมการหมุน



รูปที่ 8.15 การเพิ่มจำนวนขดลวดและปรับให้แม่เหล็กถาวรมีซี่ (Teeth)

(ที่มา www.Pcbheaven.com/wikipages/How_Stepper_Motors_Work/)

สังเกตเห็นว่า มีขดลวดอยู่ 6 ขดลวด การทำงานจะต้องควบคุมเป็นขั้น ให้ครบ 75 ขั้นจึงหมุนได้ครบ 1 รอบ หรือ 5 องศาต่อการเรียงขดลวด 6 ขดก็ไม่ได้วางห่างกัน 60 องศาแบบตรงๆ ถ้าทำแบบนี้มอเตอร์จะไม่ หมุน แต่ต้องเรียงให้คู่แรกคือขดลวดบนสุดและล่างสุดวางห่างจากคู่ที่สอง และสามเป็นมุม $60 + 5 = 65$ องศา เพื่อให้มันหมุนทีละ 5 องศา มุมต่างนี้มีผลต่อการควบคุมและการเขียนโปรแกรม

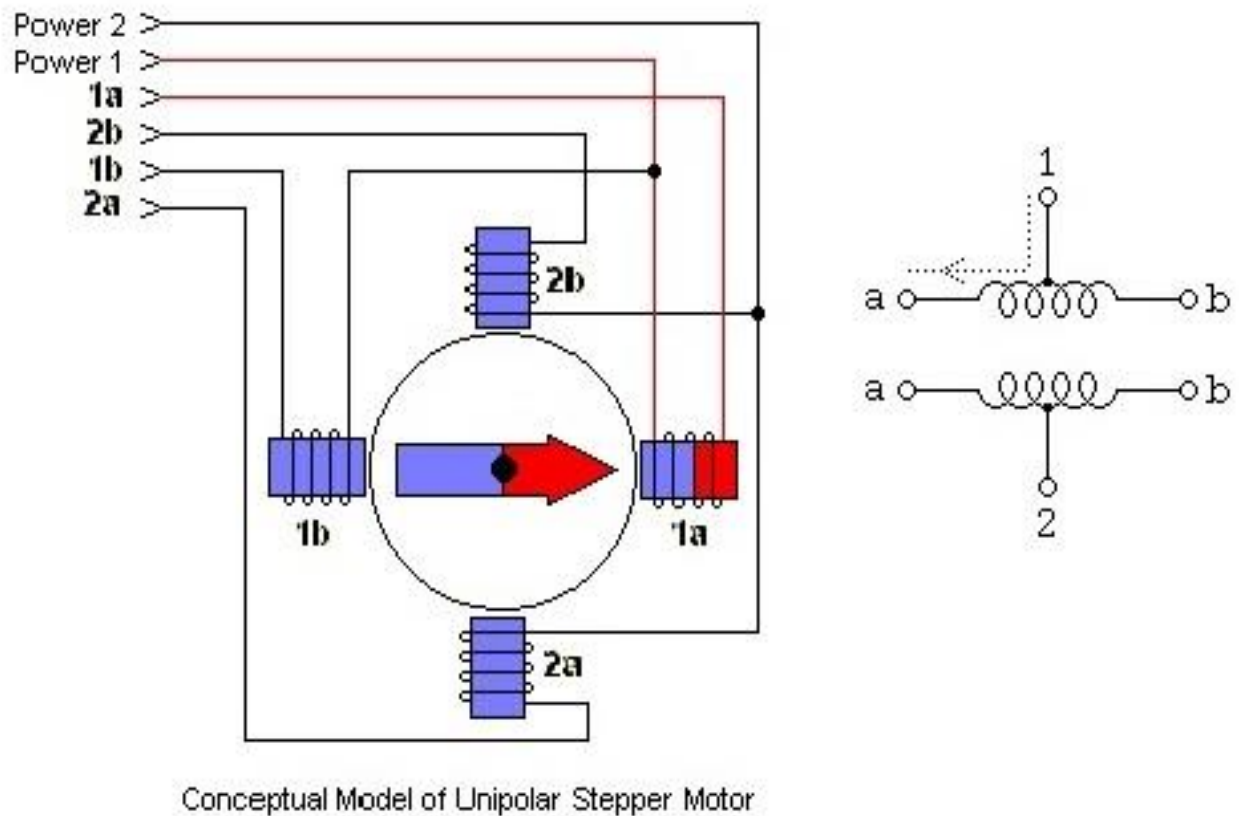


รูปที่ 8.16 ลักษณะของแกนโรเตอร์แบบแม่เหล็กถาวรมีซี่ (Teeth)
(ที่มา www.Pcbheaven.com)

อุปกรณ์จริงอย่างรูปที่ 8.16 จะเห็นว่ามี 2 จาน แต่ละจานมี 50 ซี่ ทำให้รวมมีซี่ที่สร้างเป็นขั้วเหนือและใต้รวม 100 ซี่

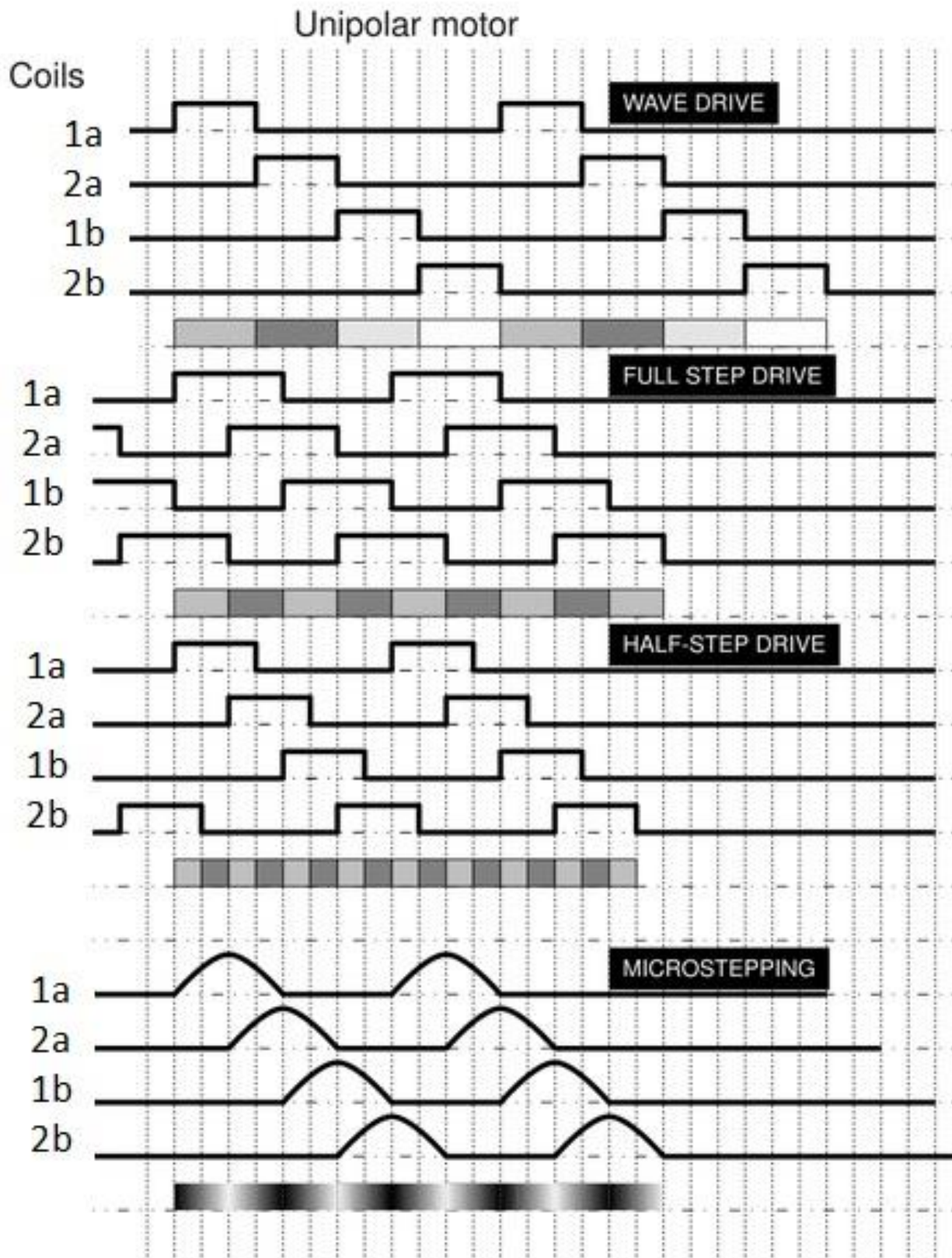
สำหรับ Stepper Motor ที่มีใช้กันอยู่จะมี 2 แบบ ขึ้นกับการต่อขดลวดภายในมอเตอร์ ได้แก่แบบ Unipolar และแบบ Bipolar แบบที่นิยมใช้คือแบบ Unipolar เพราะการควบคุมง่ายกว่า เนื่องจากไม่ต้องกลับทิศของกระแสที่ป้อนเข้าไปที่ขดลวด แต่ถ้าเป็นแบบ Bipolar ก็ทำได้ยาก แต่สามารถประยุกต์ใช้งานได้มากกว่า ใน Stepper Motor จริงๆนั้นขดลวดจะทำงานเป็น 2 ชุด (ถึงแม้จะมีขดลวดมากกว่า 4 ก็ตาม) ถ้าตามรูปที่ 8.17 คือ a กับ b โดยมี 1 กับ 2 เป็นแหล่งจ่ายไปบวกลบ และ 1a 1b 2a 2b เป็นตัวกำหนดทิศทางการไหลของกระแสผ่านขดลวดทั้งสอง

การทำให้ Stepper Motor หมุนแบบง่ายที่สุดก็คือ การจ่ายไฟเข้าไปที่ละขดตามลำดับ 1a 2a 1b 2b เรียกวิธีการควบคุมแบบนี้ว่า "Wave Drive" การทำให้ Stepper Motor หมุนแบบ Full Step Drive ทำได้โดยให้กระแสไหลผ่านขดลวดที่ละ 2 ขด ทำให้ได้ทอร์กกออกมามากกว่าแบบแรกประมาณ 2 เท่า ซึ่งเป็นวิธีที่ library ของ Arduino เลือกใช้

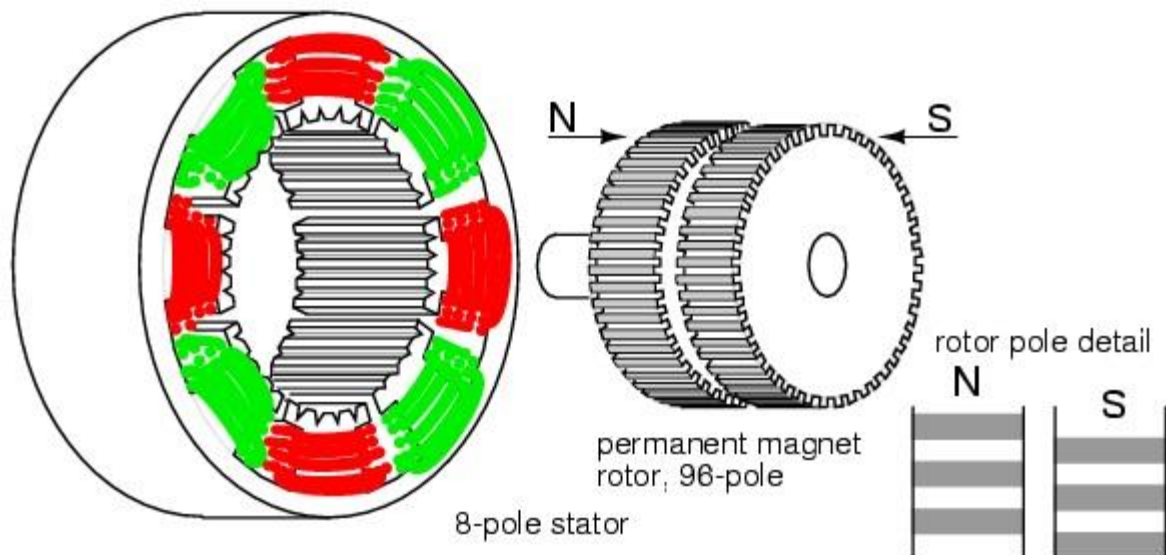


รูปที่ 8.17 Unipola Stepper motor

(ที่มา www.Pcbheaven.com/wikipages/How_Stepper_Motors_Work/)



รูปที่ 8.18 การป้อนกระแสไฟฟ้าให้กับ Stepper motor แบบต่างๆ
(ที่มา www.Pcbheaven.com/wikipages/How_Stepper_Motors_Work/)



รูปที่ 8.19 การวางขั้วแม่เหล็กถาวรใน Stepper motor

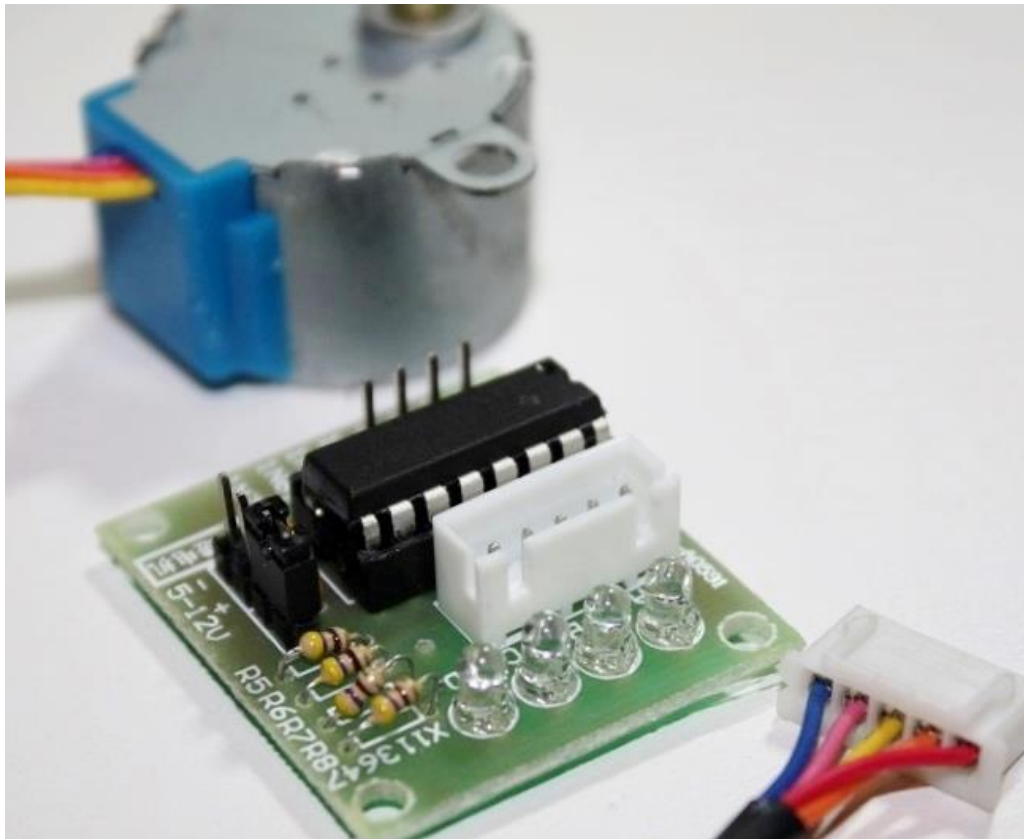
(ที่มา www.Pcbheaven.com/wikipages/How_Stepper_Motors_Work/)

ข้อมูลที่สำคัญอีกอย่างที่ควรทราบคือการทำงานของ Stepper Motor นั้นใช้กระแสที่สูงมากกว่าที่ Arduino สามารถขับได้ ดังนั้นจึงต้องใช้วงจรขยายสัญญาณเรียกว่า "วงจรร Darlington" เป็น IC สำเร็จรูปเรียกว่า ULN2003AN หรือรุ่นใกล้เคียง

ตัวอย่างสเปกของ Stepper Motor ที่จำหน่ายโดยหัวไปรูน 28BYJ-48 – 5V Stepper Motor

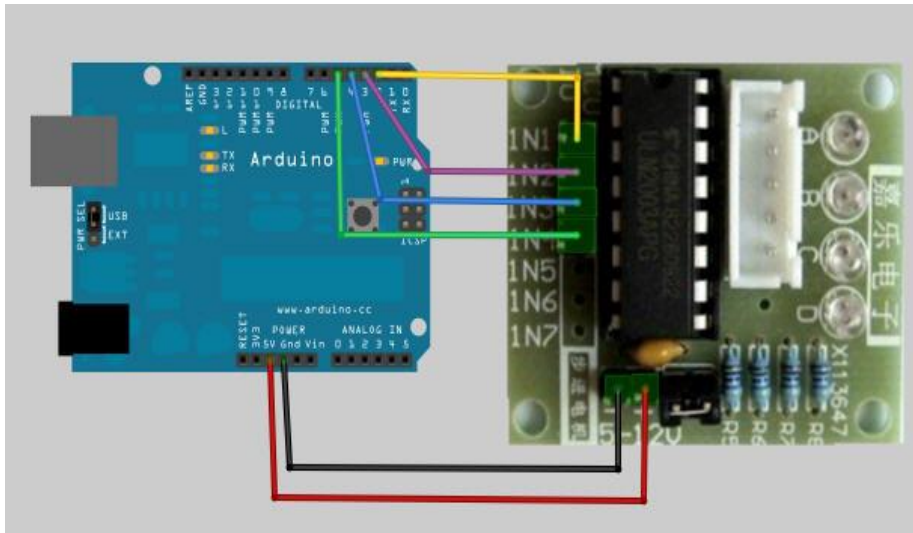
- Rated voltage : 5VDC
- Number of Phase 4
- Speed Variation Ratio 1/64
- Stride Angle $5.625^\circ / 64$
- Frequency 100Hz
- DC resistance $50\Omega \pm 7\% (25^\circ\text{C})$
- Idle In-traction Frequency > 600Hz
- Idle Out-traction Frequency > 1000Hz
- In-traction Torque > 34.3mN.m(120Hz)
- Self-positioning Torque > 34.3mN.m
- Friction torque 600-1200 gf.cm
- Pull in torque 300 gf.cm
- Insulated resistance > 10M Ω (500V)
- Insulated electricity power 600VAC/1mA/1s

- Insulation grade A
- Rise in Temperature $<40K(120Hz)$
- Noise $<35dB(120Hz)$



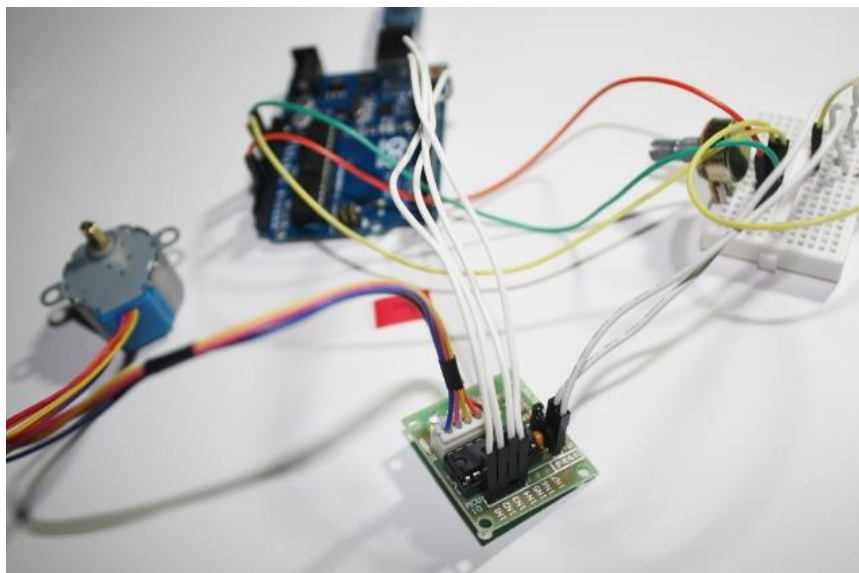
รูปที่ 8.20 วงจร Darlington" เป็น IC สำเร็จรูปเบอร์ ULN2003AN
(arduitronics.com)

การนำไปใช้งาน เป็นตัวอย่างการนำ Stepper Motor ต่อร่วมกับ Potentiometer หรือ R ปรับค่าได้มาควบคุมตำแหน่งการหมุนของ Stepper Motor



รูปที่ 8.21 การต่อ Stepper motor ใช้งานร่วมกับ Arduino
(ที่มา www.Arduitrronics.com)

จากนั้นก็ต่อ Potentiometer เข้าที่ช่อง A2



รูปที่ 8.22 การต่อ Stepper motor ใช้งานร่วมกับ Arduino และ Potentiometer
(ที่มา www.Arduitrronics.com)

จะเห็นว่า R ปรับค่าได้หรือ Potentiometer จะเป็นตัวกำหนดความเร็วและทิศทางการหมุนของ Stepper Motor

โปรแกรมที่ 8.4 Full Step 1 เฟส

```
int motorPin1 = 8;// Blue - 28BYJ48 pin 1
int motorPin2 = 9;// Pink - 28BYJ48 pin 2
int motorPin3 = 10;// Yellow - 28BYJ48 pin 3
int motorPin4 = 11;// Orange - 28BYJ48 pin 4
int motorSpeed = 4; //variable to set stepper speed
char val = '0';

void setup() {
    //declare the motor pins as outputs
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(motorPin3, OUTPUT);
    pinMode(motorPin4, OUTPUT);
    Serial.begin(9600);
}

void loop(){
    if (Serial.available())
    {
        val = Serial.read();
        Serial.println(val);
    }
    if (val == '1'){
        clockwise();
    }
    if (val == '2'){
        counterclockwise();
    }
    if (val == '0'){
        digitalWrite(motorPin1, LOW);
        digitalWrite(motorPin2, LOW);
        digitalWrite(motorPin3, LOW);
        digitalWrite(motorPin4, LOW);
    }
}
```

```
    }  
    }  
void counterclockwise ()  
    // 1  
    digitalWrite(motorPin1, HIGH);  
    digitalWrite(motorPin2, LOW);  
    digitalWrite(motorPin3, LOW);  
    digitalWrite(motorPin4, LOW);  
    delay(motorSpeed);  
    // 2  
    digitalWrite(motorPin1, LOW);  
    digitalWrite(motorPin2, HIGH);  
    digitalWrite(motorPin3, LOW);  
    digitalWrite(motorPin4, LOW);  
    delay (motorSpeed);  
    // 3  
    digitalWrite(motorPin1, LOW);  
    digitalWrite(motorPin2, LOW);  
    digitalWrite(motorPin3, HIGH);  
    digitalWrite(motorPin4, LOW);  
    delay(motorSpeed);  
    // 4  
    digitalWrite(motorPin1, LOW);  
    digitalWrite(motorPin2, LOW);  
    digitalWrite(motorPin3, LOW);  
    digitalWrite(motorPin4, HIGH);  
    delay(motorSpeed);  
}  
void clockwise(){  
    // 1  
    digitalWrite(motorPin1, LOW);  
    digitalWrite(motorPin2, LOW);
```

```
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
delay(motorSpeed);
// 2
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
delay (motorSpeed);
// 3
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay(motorSpeed);
// 4
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay(motorSpeed);
}
```

โปรแกรมที่ 8.5 Code Full Step 2เฟส

```
int motorPin1 = 8;// Blue - 28BYJ48 pin 1
int motorPin2 = 9;// Pink - 28BYJ48 pin 2
int motorPin3 = 10;// Yellow - 28BYJ48 pin 3
int motorPin4 = 11;// Orange - 28BYJ48 pin 4
int motorSpeed = 3; //variable to set stepper speed
char val = '0';

void setup() {
    //declare the motor pins as outputs
```

```
pinMode(motorPin1, OUTPUT);
pinMode(motorPin2, OUTPUT);
pinMode(motorPin3, OUTPUT);
pinMode(motorPin4, OUTPUT);
Serial.begin(9600);
}
void loop(){
  if (Serial.available())
  {
    val = Serial.read();
    Serial.println(val);
  }
  if (val == '1'){
    clockwise();
  }
  if (val == '2'){
    counterclockwise();
  }
  if (val == '0'){
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
  }
}
void counterclockwise (){
  // 1
  digitalWrite(motorPin1, HIGH);
  digitalWrite(motorPin2, LOW);
  digitalWrite(motorPin3, LOW);
  digitalWrite(motorPin4, HIGH);
  delay(motorSpeed);
}
```

```
// 2
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay (motorSpeed);
// 3
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
delay(motorSpeed);
// 4
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, HIGH);
delay(motorSpeed);
}
void clockwise(){
    // 1
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, HIGH);
    delay(motorSpeed);
    // 2
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
    delay (motorSpeed);
```

```

// 3
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay(motorSpeed);
// 4
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
delay(motorSpeed);
}

```

โปรแกรมที่ 8.6 Code Half Step

```

int motorPin1 = 8;// Blue - 28BYJ48 pin 1
int motorPin2 = 9;// Pink - 28BYJ48 pin 2
int motorPin3 = 10;// Yellow - 28BYJ48 pin 3
int motorPin4 = 11;// Orange - 28BYJ48 pin 4
int motorSpeed = 4; //variable to set stepper speed
char val = '0';

void setup() {
  //declare the motor pins as outputs
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  if (Serial.available())
  {

```

```
        val = Serial.read();
        Serial.println(val);
    }
    if (val == '1'){
        clockwise();
    }
    if (val == '2'){
        counterclockwise();
    }
    if (val == '0'){
        digitalWrite(motorPin1, LOW);
        digitalWrite(motorPin2, LOW);
        digitalWrite(motorPin3, LOW);
        digitalWrite(motorPin4, LOW);
    }
}

void counterclockwise (){
    // 1
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay(motorSpeed);
    // 2
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay (motorSpeed);
    // 3
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
```

```
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay(motorSpeed);
// 4
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
delay(motorSpeed);
// 5
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
delay(motorSpeed);
// 6
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, HIGH);
delay (motorSpeed);
// 7
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
delay(motorSpeed);
// 8
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
```

```
        delay(motorSpeed);
    }
void clockwise(){
    // 1
    digitalWrite(motorPin4, HIGH);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin1, LOW);
    delay(motorSpeed);
    // 2
    digitalWrite(motorPin4, HIGH);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin1, LOW);
    delay (motorSpeed);
    // 3
    digitalWrite(motorPin4, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin1, LOW);
    delay(motorSpeed);
    // 4
    digitalWrite(motorPin4, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin1, LOW);
    delay(motorSpeed);
    // 5
    digitalWrite(motorPin4, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin1, LOW);
```

```
delay(motorSpeed);  
// 6  
digitalWrite(motorPin4, LOW);  
digitalWrite(motorPin3, LOW);  
digitalWrite(motorPin2, HIGH);  
digitalWrite(motorPin1, HIGH);  
delay (motorSpeed);  
// 7  
digitalWrite(motorPin4, LOW);  
digitalWrite(motorPin3, LOW);  
digitalWrite(motorPin2, LOW);  
digitalWrite(motorPin1, HIGH);  
delay(motorSpeed);  
// 8  
digitalWrite(motorPin4, HIGH);  
digitalWrite(motorPin3, LOW);  
digitalWrite(motorPin2, LOW);  
digitalWrite(motorPin1, HIGH);  
delay(motorSpeed);  
}
```

สรุปเนื้อหาสาระสำคัญ

หลักการการทำงานของ Stepper Motor คือการบังคับให้แม่เหล็กถาวรบนแกนโรเตอร์ หมุนไปตามทิศการบังคับของขดลวดที่ติดตั้งบนสเตเตอร์ จะซับซ้อนกว่า DC Motor ตรงที่การบังคับให้หมุนนั้นไม่ได้เป็นแค่การใส่แรงดันคงที่ไปที่ขั้วบวกลบเท่านั้น ต้องใส่แรงดันให้ถูกต้องจากจังหวะที่ควรจะเป็น โดยมีขดลวดที่ควบคุมการหมุนแต่ละขดห่างกัน 90 องศา การหมุนก็จะทำโดยการจ่ายกระแสเข้าไปที่ขดลวดทีละขด เพื่อทำให้เกิดสนามแม่เหล็ก ซึ่งจะดึงดูดให้แม่เหล็กถาวรที่อยู่บนโรเตอร์เคลื่อนที่ โดยทิศของการหมุนก็จะขึ้นกับลำดับการจ่ายกระแสเข้าไปที่ขดลวด โดยการบังคับในลักษณะนี้เรียกว่า Single coil excitation หรือการกระตุ้นทีละขดลวด มีการกระตุ้นหรือการจ่ายกระแสเข้าขดลวดอยู่ 4 จังหวะต่อการหมุน 1 รอบ อีกรูปแบบหนึ่งจะซับซ้อนกว่าแต่จะให้ทอร์คมากกว่าคือการป้อนแบบ Full Step Drive หรือการป้อนแบบทีละ 2 ขดลวด ถ้าต้องการควบคุมให้มีความละเอียดมากขึ้นอีก สามารถทำได้โดยควบคุมให้มอเตอร์หมุนได้ละเอียดขึ้นจาก 90 องศา เหลือ 45 องศา โดยที่ไม่ต้องปรับเปลี่ยนตัวฮาร์ดแวร์ใดๆ



แบบฝึกหัดหน่วยที่ 8

เรื่อง การควบคุมดีซีและสเตปมอเตอร์ด้วย Arduino

ใช้เวลา 20 นาที

คำชี้แจง แบบฝึกหัดมีทั้งหมด 2 ตอน ประกอบด้วยตอนที่ 1 และตอนที่ 2 (20 คะแนน)

1. แบบฝึกหัดตอนที่ 1 เป็นคำถามแบบถูก-ผิด มีทั้งหมด 10 ข้อ (10 คะแนน)
2. แบบฝึกหัดตอนที่ 2 เป็นคำถามแบบปรนัย มีทั้งหมด 10 ข้อ (10 คะแนน)



แบบฝึกหัดตอนที่ 1

คำชี้แจง ให้ผู้เรียนกาเครื่องหมายถูก ✓ ในข้อที่คิดว่าถูก และกาเครื่องหมายผิด ✗ ในข้อที่คิดว่าผิด

- 1. Stepping motor มีข้อดีที่สำคัญคือการควบคุมตำแหน่งของการหมุนได้อย่างแม่นยำ
- 2. การทำงานของ Stepper motor การบังคับให้แม่เหล็กถาวรบนแกนโรเตอร์ หมุนไปตามทิศการบังคับของขดลวดที่ติดตั้งบนสเตเตอร์
- 3. การบังคับให้ Stepping motor หมุนนั้นไม่ได้เป็นแค่การใส่แรงดันคงที่ไปที่ขั้วขดลวดเท่านั้น ต้องใส่แรงดันให้ถูกต้องจากจังหวะที่ควรจะเป็น
- 4. ขดลวด Stepping motor ที่ควบคุมการหมุนแบบสี่ขั้ว แต่ละขดห่างกัน 60 องศา
- 5. การหมุน Stepping motor ทำโดยการจ่ายกระแสเข้าไปที่ขดลวดทีละขดเพื่อทำให้เกิดสนามแม่เหล็ก ซึ่งจะดึงดูดให้แม่เหล็กถาวรที่อยู่บนสเตเตอร์เคลื่อนที่
- 6. ทิศของการหมุน Stepping motor ขึ้นกับลำดับการจ่ายกระแสเข้าไปที่ขดลวด
- 7. การบังคับ Stepping motor ในลักษณะนี้เรียกว่า Single coil excitation หรือการกระตุ้นทีละขดลวด
- 8. การกระตุ้น Stepping motor หรือการจ่ายกระแสเข้าขดลวดอยู่ 4 จังหวะต่อการหมุน 1 รอบ
- 9. การป้อน Stepping motor แบบ Full Step Drive หรือการป้อนแบบทีละ 2 ขดลวด
- 10. สามารถควบคุมการหมุน Stepping motor ได้ละเอียดมาก โดยการควบคุม 1 Cycle จะทำให้มอเตอร์หมุนไป 0.9 - 5 องศา

 แบบฝึกหัดตอนที่ 2

คำชี้แจง ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ให้ครบทุกข้อ

1. มอเตอร์ไฟฟ้าแบ่งออกตามการใช้ของกระแสไฟฟ้าได้ 2 ชนิด คือ
 - ก. มอเตอร์ไฟฟ้ากระแสสูง / มอเตอร์ไฟฟ้ากระแสต่ำ
 - ข. มอเตอร์ไฟฟ้ากระแสสลับ / มอเตอร์ไฟฟ้ากระแสตรง
 - ค. มอเตอร์ไฟฟ้ากระแสบวก / มอเตอร์ไฟฟ้ากระแสลบ
 - ง. มอเตอร์ไฟฟ้ากระแสบ้าน / มอเตอร์ไฟฟ้ากระแสโรงงาน
2. กระแสไฟฟ้าที่ไหลเข้าไปในมอเตอร์ไฟฟ้ากระแสตรงจะไหลเข้าไปที่ใด
 - ก. คอมมิวเตเตอร์ / ขดลวดอาร์มาเจอร์
 - ข. ขดลวดอาร์มาเจอร์ / ขดลวดสนามแม่เหล็ก
 - ค. แปรงถ่าน / คอมมิวเตเตอร์
 - ง. แปรงถ่าน / ขดลวดอาร์มาเจอร์
3. คุณสมบัติของเส้นแรงแม่เหล็กเป็นอย่างไร
 - ก. หักล้างกันในทิศทางตรงข้าม
 - ข. เสริมแรงทิศทางเดียวกัน
 - ค. หักล้างกันในทิศทางตรงข้าม และ เสริมแรงทิศทางเดียวกัน เกิดแรงบิดในตัวอาร์มาเจอร์
 - ง. เกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน
4. ทุ่นอาร์มาเจอร์คือ
 - ก. ส่วนที่หมุน
 - ข. ส่วนอยู่กับที่
 - ค. ขดลวดสนามแม่เหล็ก
 - ง. ขดลวด
5. การนำกระแสตรงมาตัดผ่านสนามแม่เหล็ก ทำให้เกิดทอร์กขึ้นในทิศที่เหมาะสมเพื่อ
 - ก. สร้างให้เกิดการหมุนของสเตเตอร์
 - ข. สร้างให้เกิดการหมุนของขดลวด
 - ค. สร้างให้เกิดการหมุนของทุ่นอาร์มาเจอร์
 - ง. สร้างให้เกิดการหมุนของขดลวดสนามแม่เหล็ก

6. วงจรที่ใช้ในการควบคุมมอเตอร์แบบที่นิยมใช้กันเรียกว่า
 - ก. A-Bridge
 - ข. H-Bridge
 - ค. R-Bridge
 - ง. T-Bridge
7. หลักการแก้ไขสัญญาณรบกวนทำได้หลายแบบ เช่นต่อกันในลักษณะวงจร
 - ก. วงจร RL อินทิเกรเตอร์
 - ข. วงจร RLC อินทิเกรเตอร์
 - ค. วงจร LC อินทิเกรเตอร์
 - ง. วงจร RC อินทิเกรเตอร์
8. หากต้องการให้หมุนทวนเข็มนาฬิกา (Conter Clockwise :CCW) ให้
 - ก. S1 และ S4 ปิดวงจร
 - ข. S2 และ S3 ปิดวงจร และ S1 และ S4 เปิดวงจร
 - ค. S1 และ S4 ปิดวงจร และ S2 และ S3 เปิดวงจร
 - ง. S2 และ S3 ปิดวงจร
9. พารามิเตอร์ที่ใช้ระบุรูปร่างหน้าตาของ PWM ที่สำคัญคือ
 - ก. วงจรบัค วงจรบูส
 - ข. วงจรบัคบูส
 - ค. ความถี่ของคลื่นพาหะ (Carrier Frequency) และอัตราส่วนหน้าที่ (Duty Ratio)
 - ง. ฟังก์ชันชะลอการทำงานในหน่วยนาโนวินาที
10. PWM (Pulse Width Modulation) คือ
 - ก. การมอดูเลตสัญญาณตามความถี่
 - ข. การมอดูเลตสัญญาณตามแอมพลิจูด
 - ค. การมอดูเลตสัญญาณเฟสเซอร์
 - ง. การมอดูเลตสัญญาณให้มีความกว้างตามสัดส่วนที่กำหนด

ปฏิบัติการทดลองหน่วยที่ 8

เรื่อง การควบคุมดีซีมอเตอร์และสเตปมอเตอร์ด้วย Arduino

คำชี้แจง ให้ผู้เรียนทุกคนทำการทดลองตามปฏิบัติการทดลองหน่วยที่ 8 เรื่อง การควบคุมดีซีมอเตอร์และสเตปมอเตอร์ด้วย Arduino ใช้เวลา 180 นาที (20 คะแนน)

จุดประสงค์เชิงพฤติกรรม

1. สามารถใช้ฟังก์ชันได้ถูกต้อง
2. สามารถแก้ปัญหาในการทำงานของบอร์ด Arduino Uno R3 ได้
3. สามารถต่อใช้งานและอัปโหลดโปรแกรมให้กับบอร์ด Arduino Uno R3 ได้

อุปกรณ์การทดลอง

1. เครื่องคอมพิวเตอร์และโปรแกรม Arduino IDE 1.6.9	1	ชุด
2. USB Cable Arduino Uno R3	1	เส้น
3. Arduino Uno R3 Board	1	บอร์ด
4. Hook-up Wires	10	เส้น
5. Breadboard	1	แผง
6. A Momentary Switch or Button	1	ตัว
7. 9V DC Motor	1	ตัว
8. 10k Ohm Potentiometer	1	ตัว
9. TIP120 Transistor	1	ตัว
10. 1N4001 Diode	1	ตัว
11. 9V Battery	1	ตัว
12. Stepper Motor		
13. U2004 Darlington Array (if using a unipolar stepper)	1	ตัว
14. SN754410ne H-Bridge (if using a bipolar stepper)	1	ตัว
15. Power Supply Appropriate for your Particular Stepper	1	ตัว

ข้อควรระวัง

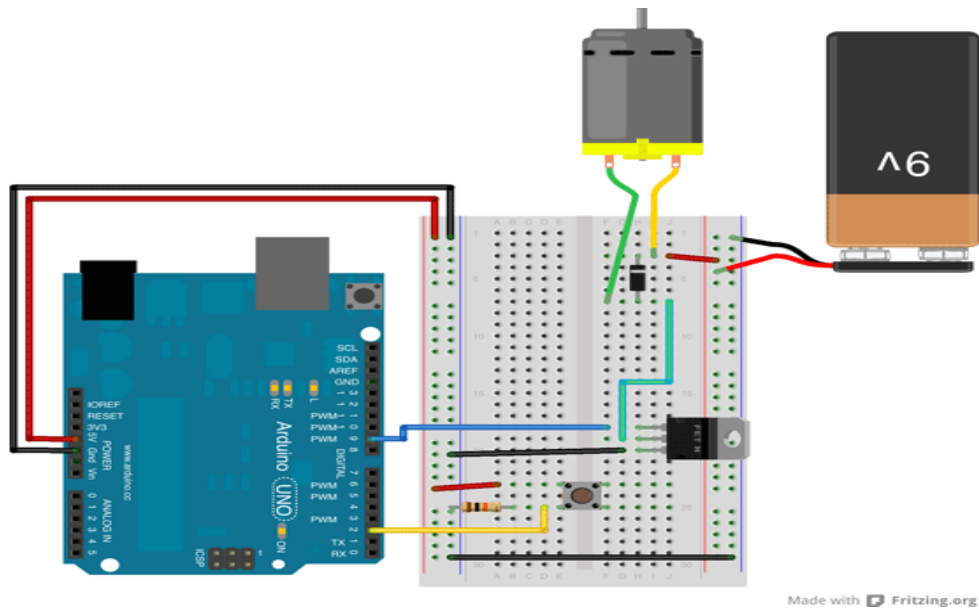
1. ควรระวังไม่วางบอร์ด Arduino Uno R3 หรือซีลต่างๆ บนโต๊ะโลหะหรือที่วางที่เป็นโลหะเพราะอาจเกิดการลัดวงจรของภาคจ่ายไฟได้
2. ไม่ควรต่อสายต่อวงจรในบอร์ด Arduino Uno R3 ทิ้งไว้ ควรถอดสายต่อวงจรออกให้หมด เพราะผลการทดลองอาจเกิดการผิดพลาดไม่เป็นไปตามทฤษฎีได้
3. ไม่ควรถอดสายสายโหนด USB เข้าออกตลอดเวลา เพราะอาจทำให้ภาคจ่ายไฟของบอร์ด Arduino Uno R3 เสียหายได้

การทดลองที่ 8.1 การควบคุมมอเตอร์ไฟฟ้ากระแสตรงด้วย Arduino

Hardware Required

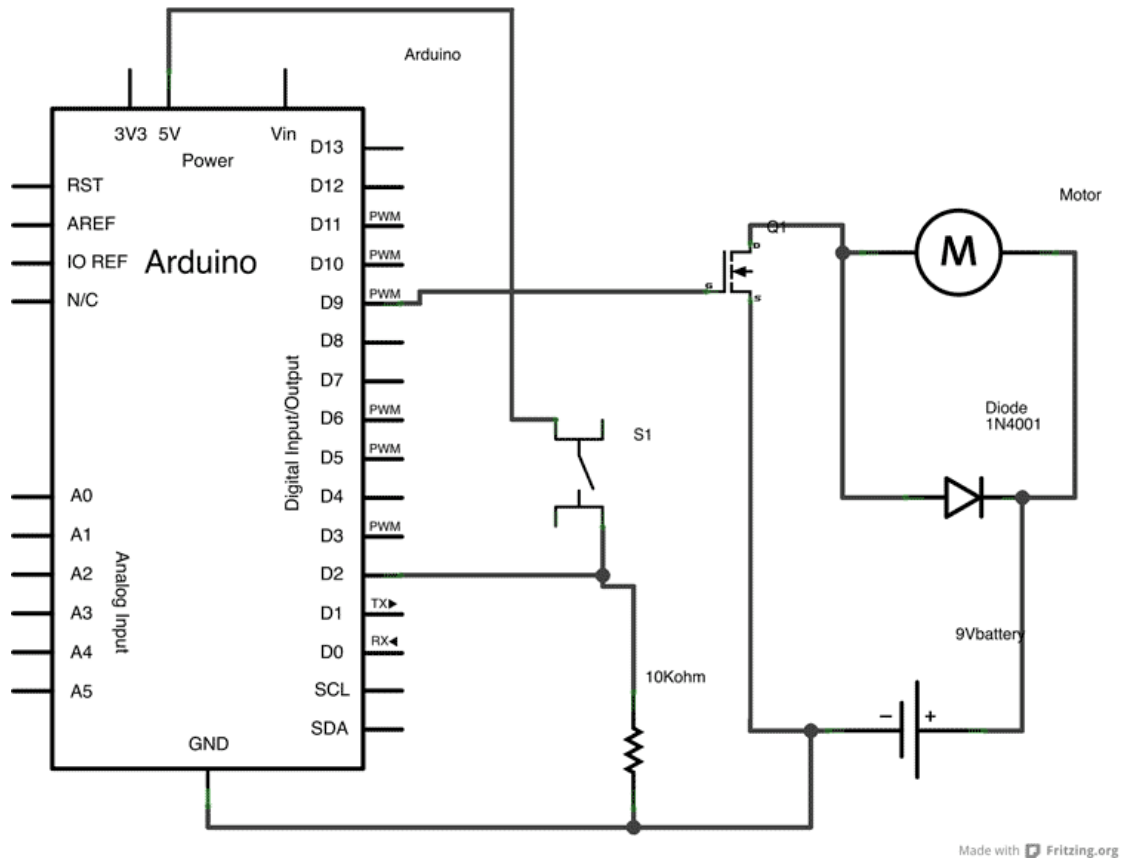
1. Arduino Board
2. A momentary switch or button
3. 10k ohm resistor
4. breadboard
5. hook-up wire
6. 9V DC motor
7. TIP120 transistor
8. 1N4001 diode
9. 9V battery

Circuit



รูปที่ 8. 22 การควบคุมมอเตอร์ไฟฟ้ากระแสตรงด้วย Arduino

Schematic



Code

```
/*  
by Scott Fitzgerald  
http://www.arduino.cc/en/Tutorial/TransistorMotorControl  
This example code is in the public domain.  
*/  
  
// give a name to digital pin 2, which has a pushbutton attached  
int pushButton = 2;  
// the transistor which controls the motor will be attached to digital pin 9  
int motorControl = 9;  
// the setup routine runs once when you press reset:  
  
void setup() {  
  // make the pushbutton's pin an input:  
  pinMode(pushButton, INPUT);  
  // make the transistor's pin an output:  
  pinMode(motorControl, OUTPUT);  
}  
// the loop routine runs over and over again forever:  
void loop() {  
  // read the state of the button and check if it is pressed  
  if(digitalRead(pushButton) == HIGH){  
    // ramp up the motor speed  
    for(int x = 0; x <= 255; x++){  
      analogWrite(motorControl, x);  
      delay(50);  
    }  
    // ramp down the motor speed  
    for(int x = 255; x >= 0; x--){  
      analogWrite(motorControl, x);  
      delay(50);  
    }  
  }  
}
```

```
}  
delay(1);    // delay in between reads for stability  
}
```

ผลการทดลอง

.....

.....

.....

.....

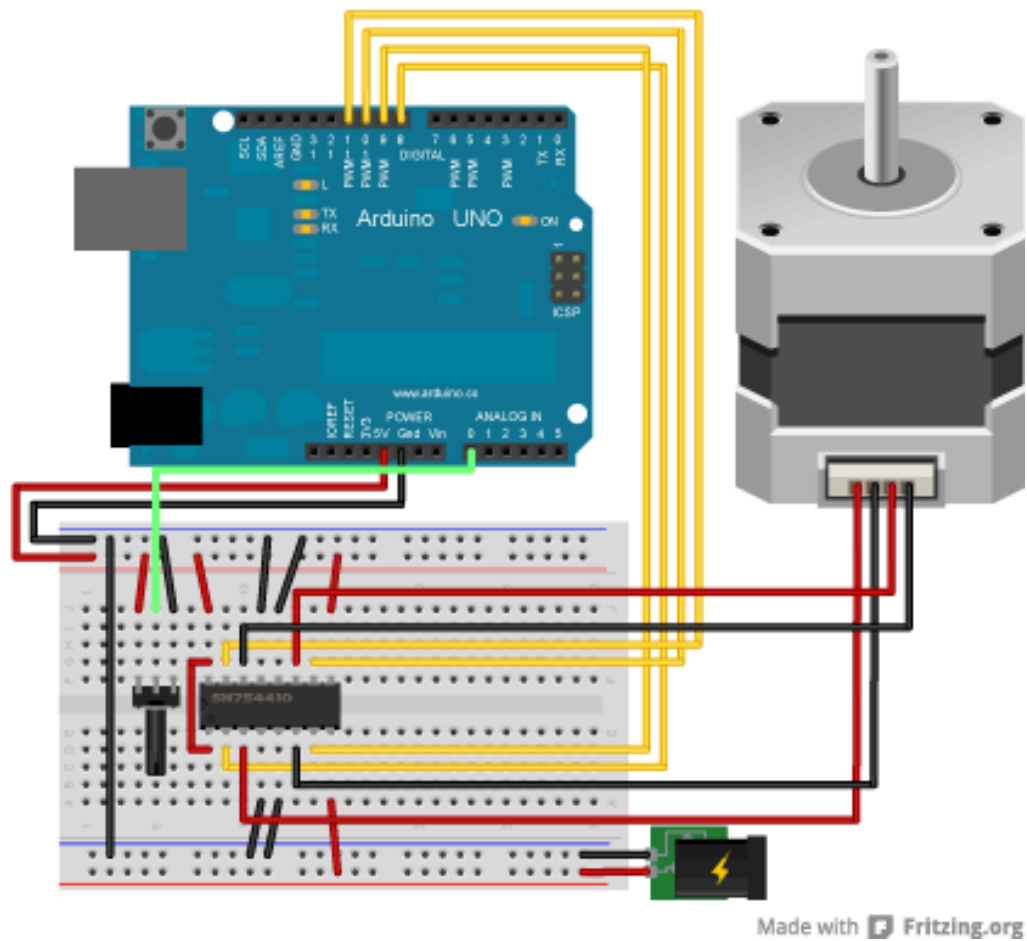
.....

การทดลองที่ 8.2 การควบคุมสเตปมอเตอร์ด้วย Arduino

Hardware Required

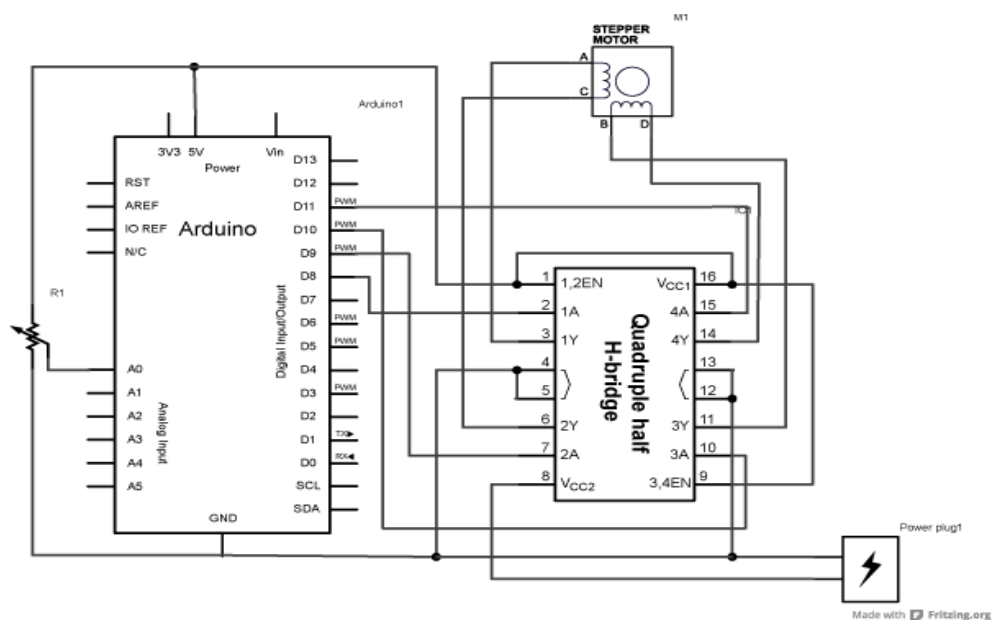
1. Arduino or Genuino Board
2. 10k ohm potentiometer
3. stepper motor
4. U2004 Darlington Array (if using a unipolar stepper)
5. SN754410ne H-Bridge (if using a bipolar stepper)
6. power supply appropriate for your particular stepper
7. hook-up wires
8. breadboard

Circuits Bipolar Motor Circuit



รูปที่ 8. 22 การควบคุมสเตปมอเตอร์ด้วย Arduino

Schematic



Code

```
/*  
  Stepper Motor Control - speed control  
  Created 30 Nov. 2009  
  Modified 28 Oct 2010  
  by Tom Igoe  
*/  
  
#include <Stepper.h>  
const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution  
// for your motor  
// initialize the stepper library on pins 8 through 11:  
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);  
int stepCount = 0; // number of steps the motor has taken  
void setup() {  
  // nothing to do inside the setup  
}  
void loop() {  
  // read the sensor value:  
  int sensorReading = analogRead(A0);  
  // map it to a range from 0 to 100:  
  int motorSpeed = map(sensorReading, 0, 1023, 0, 100);  
  // set the motor speed:  
  if (motorSpeed > 0) {  
    myStepper.setSpeed(motorSpeed);  
    // step 1/100 of a revolution:  
    myStepper.step(stepsPerRevolution / 100);  
  }  
}
```

ผลการทดลอง

.....

.....

.....

.....

แบบทดสอบหลังเรียน หน่วยที่ 8

เรื่อง การควบคุมดีซีมอเตอร์และสเตปมอเตอร์ด้วย Arduino

เรื่อง การควบคุมดีซีและสเตปมอเตอร์ด้วย Arduino

ใช้เวลา 20 นาที

วิชา ไมโครคอนโทรลเลอร์เบื้องต้น

รหัสวิชา (2127-2107)

ระดับชั้น ประกาศนียบัตรวิชาชีพ (ปวช.)

สาขาวิชา เมคคาทรอนิกส์

คำชี้แจง 1. แบบทดสอบมีทั้งหมด 10 ข้อ (10 คะแนน)

2. ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ลงในกระดาษคำตอบ

1. มอเตอร์มอเตอร์ไฟฟ้า (Motor) หมายถึง

- ก. เครื่องกลไฟฟ้าที่เปลี่ยนแปลงพลังงานไฟฟ้าเป็นพลังงานแสง
- ข. เครื่องกลไฟฟ้าที่เปลี่ยนแปลงพลังงานกลเป็นพลังงานไฟฟ้า
- ค. เครื่องกลไฟฟ้าที่เปลี่ยนแปลงพลังงานไฟฟ้าเป็นพลังงานกล
- ง. เครื่องกลไฟฟ้าที่เปลี่ยนแปลงพลังงานกลเป็นพลังงานแสง

2. มอเตอร์ไฟฟ้ากระแสตรงเป็นต้นกำลังขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรมเพราะ

- ก. ราคาถูก
- ข. ทนทาน
- ค. ประหยัดไฟฟ้า
- ง. ปรับความเร็วได้

3. กระแสไฟฟ้าในขดลวดสนามแม่เหล็ก (Field Coil) จะสร้างขั้วแม่เหล็กแบบใด

- ก. ขั้วเหนือ-ใต้
- ข. ขั้วสูง-ต่ำ
- ค. ขั้วบน-ล่าง
- ง. ขั้วซ้าย-ขวา

4. DC Motor ประกอบด้วย 2 ส่วนหลักๆ ได้แก่

- ก. ฟันอาร์มาเจอร์ / สเตเตอร์
- ข. ขั้วแม่เหล็กเหนือ / ใต้
- ค. ขดลวดอาร์มาเจอร์ / ขดลวดสนามแม่เหล็ก
- ง. มอเตอร์ไฟฟ้ากระแสสลับ / มอเตอร์ไฟฟ้ากระแสตรง

5. หลักการทำงานของ DC Motor คือ

- ก. กระแสสลับตัดผ่านสนามแม่เหล็ก เกิดทอร์กขึ้นในทิศที่เหมาะสม
- ข. แรงดันสลับตัดผ่านสนามแม่เหล็ก เกิดทอร์กขึ้นในทิศที่เหมาะสม
- ค. กระแสตรงตัดผ่านสนามแม่เหล็ก เกิดทอร์กขึ้นในทิศที่เหมาะสม
- ง. แรงดันสลับตัดผ่านสนามแม่เหล็ก เกิดทอร์กขึ้นในทิศที่เหมาะสม

6. วงจร H-Bridge สามารถควบคุมมอเตอร์ได้อย่างไร

- ก. ควบคุมทิศทางของมอเตอร์
- ข. ควบคุมความเร็วรอบของมอเตอร์
- ค. ควบคุมการเดินหน้าและถอยหลังของมอเตอร์
- ง. ควบคุมทิศทาง และความเร็วรอบของมอเตอร์

7. หากต้องการให้หมุนตามเข็มนาฬิกา (Clockwise :CW) ให้สวิตช์ทำงานอย่างไร

- ก. S1 และ S4 ปิดวงจร
- ข. S2 และ S3 ปิดวงจร และ S1 และ S4 เปิดวงจร
- ค. S1 และ S4 ปิดวงจร และ S2 และ S3 เปิดวงจร
- ง. S2 และ S3 ปิดวงจร

8. อุปกรณ์สารกึ่งตัวนำที่นิยมใช้ในวงจร H-Bridge คือ

- ก. TRIAC หรือ DIAC
- ข. DIODE หรือ ZENERDIODE
- ค. MOSFET หรือ IGBT
- ง. TRANSISTER

9. หลักการของการทำงานของ Stepper Motor คือ

- ก. การมอดูเลตสัญญาณตามแอมพลิจูด
- ข. การมอดูเลตสัญญาณเฟสเซอร์
- ค. การจ่ายไฟเข้าไปที่ละขดตามลำดับ
- ง. การมอดูเลตสัญญาณตามความถี่

10. วงจร Darlington เป็น IC สำเร็จรูปมีชื่อเรียกว่า

- ก. 7812
- ข. L298P
- ค. ULN2003AN
- ง. 1N4001