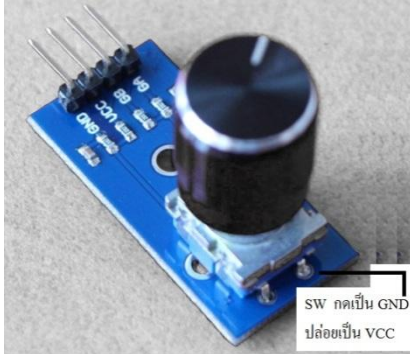


# การทดลอง Rotary Encoder สวิตช์

## บทนำ

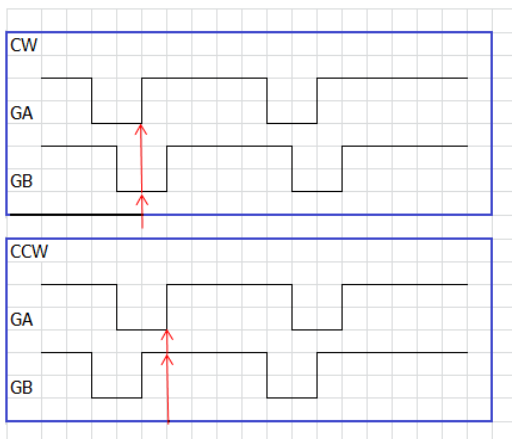


หมายเหตุ หน้าสัมผัสสวิตช์ กับสัญญาณ GA เกี่ยวข้องกัน ยังไม่ได้ใช้

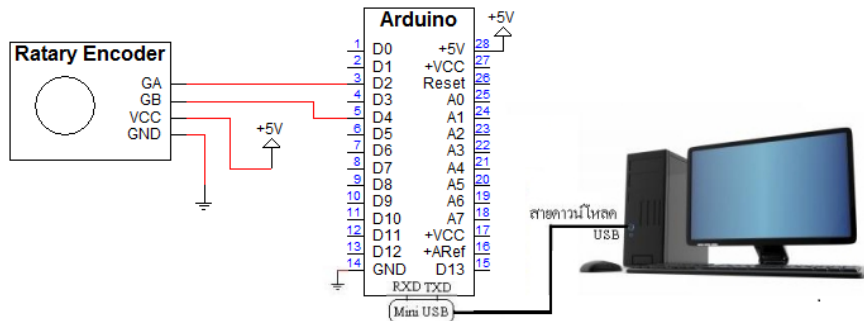
Rotary encoder เป็นอุปกรณ์แปลงการหมุนแบบ 360 องศา เป็นสัญญาณทางไฟฟ้า เพื่อใช้ในการปรับค่าของการทำงานของอุปกรณ์ ซึ่งทำงานต่างจาก Potentiometer คือ สามารถหมุนได้ต่อเนื่องไม่มีสุดปลายในแต่ละด้าน พร้อมสวิตช์ปุ่มกด

- Location Number: 20
- Pulse count: 20
- With Switch: Yes
- Operating life: 30,000 times
- Maximum Rated: 10mA 5V DC
- Encoder pin no welding.

## การทำงาน



การทดลองที่ 1 ใช้การอ่านค่าสัญญาณ GA ถ้าเปลี่ยนระดับจาก LOW เป็น HIGH ก็มาตรวจสอบที่สัญญาณ GB ถ้าเป็น 0 แสดงว่าหมุนตามเข็มนาฬิกาให้เพิ่มค่า encoder0Pos ขึ้น แต่ถ้าเป็น 1 แสดงว่าหมุน ทวนเข็มนาฬิกา ให้ลดค่า encoder0Pos ลง



### โปรแกรม rotary\_1

```
int encoder0PinA = 2;
int encoder0PinB = 4;
int encoder0Pos = 0;
int encoder0PinALast = LOW;
int n = LOW;

void setup() {
  pinMode (encoder0PinA,INPUT);
  pinMode (encoder0PinB,INPUT);
  Serial.begin (9600);
  Serial.println ("Rotary Encoder test 1 ");
}

void loop() {
  n = digitalRead(encoder0PinA);
  if ((encoder0PinALast == LOW) && (n == HIGH)) {
    if (digitalRead(encoder0PinB) == LOW) {
      encoder0Pos++;
    } else {
      encoder0Pos--;
    }
    Serial.println (encoder0Pos);
  }
  encoder0PinALast = n;
}
```

การทดลองที่ 2 ใช้การอินเตอร์รัพท์เมื่อสัญญาณ GA เปลี่ยนจาก LOW ไป HIGH (RISING) สัญญาณ GA ต่ออยู่ที่ขา 2 ซึ่งเป็นสัญญาณ INTO 07' จึงกำหนดสถานะการทำงานอินเตอร์รัพท์ INTO ด้วยคำสั่ง  
attachInterrupt(0, doEncoder, RISING )

### ความหมาย

เมื่อ INTO ได้รับความสัญญาณขอบบวก (เปลี่ยนจาก LOW ไป HIGH) จะไปทำงานที่ฟังก์ชัน doEncoder เพื่อทำการตรวจสอบสถานะของสัญญาณ GB ถ้าเป็น 0 แสดงว่าหมุนตามเข็มนาฬิกาให้เพิ่มค่า

encoder0Pos ขึ้น แต่ถ้าเป็น 1 แสดงว่าหมุน ทวนเข็มนาฬิกา ให้ลดค่า encoder0Pos ลง เช่นเดียวกับการทดลองที่ 1

โปรแกรม rotary\_2\_int

```
#define encoder0PinA 2
#define encoder0PinB 4

volatile unsigned int encoder0Pos = 0;
volatile unsigned int old_val = 0;

void setup() {

  pinMode(encoder0PinA, INPUT);
  digitalWrite(encoder0PinA, HIGH); // turn on pull-up resistor
  pinMode(encoder0PinB, INPUT);
  digitalWrite(encoder0PinB, HIGH); // turn on pull-up resistor

  attachInterrupt(0, doEncoder, RISING ); // encoderA pin 2 on interrupt 0
  Serial.begin (9600);
  Serial.println("Rotary Encoder Test interrupt"); // a personal quirk
  old_val = encoder0Pos;
}

void loop(){
  if(old_val != encoder0Pos){
    Serial.println (encoder0Pos);
    old_val = encoder0Pos;
  }
}

void doEncoder() {
  // Interrupt when PINA goes from low to high//

  if (digitalRead(encoder0PinB) == LOW) {
    encoder0Pos++;
  } else {
    encoder0Pos--;
  }
}
```

การทดลองที่ 3 เพื่อลดข้อผิดพลาดของการนับที่เกิดจากหน้าสัมผัส ใช้การหารหาร

โปรแกรม rotary\_3\_int

```
#define encoder0PinA 2
#define encoder0PinB 4

unsigned char encoder0Pos = 0;

unsigned char old_val = 0;
unsigned char new_val = 0;
```

```

void setup() {

  pinMode(encoder0PinA, INPUT);
  digitalWrite(encoder0PinA, HIGH);    // turn on pull-up resistor
  pinMode(encoder0PinB, INPUT);
  digitalWrite(encoder0PinB, HIGH);    // turn on pull-up resistor

  attachInterrupt(0, doEncoder, RISING ); // encoderA pin 2 on interrupt 0
  Serial.begin (9600);
  Serial.println("Rotary Encoder Test interrupt");          // a personal quirk
  old_val = new_val;
}

void loop(){
  new_val = encoder0Pos>>2;
  if(old_val != new_val){
    Serial.print (encoder0Pos);
    Serial.print (" --- val -- ");
    Serial.println (new_val);
    old_val = new_val;
  }
}

void doEncoder() {
  // Interrupt when PINA goes from low to high//

  if (digitalRead(encoder0PinB) == LOW) {
    encoder0Pos++;
  } else {
    encoder0Pos--;
  }
}
}

```

## คำถาม

1. การกำหนดค่าตัวแปร encoder0Pos ของทั้งสองโปรแกรมต่างกันอย่างไร และให้ผลลัพธ์เป็นเช่นไร
2. ให้อธิบายการทำงานของโปรแกรมในการทดลองที่ 2
3. ให้เขียนโปรแกรมตรวจสอบการหมุนที่ต่างไปจากโปรแกรมในการทดลองที่ 2
4. การทดลองที่ 2 กับ 3 ให้ผลลัพธ์ต่างกันอย่างไร
5. จงอธิบายการทำงานของโปรแกรมในการทดลองที่ 3