

สเต็ปมอเตอร์ (STEPPING MOTOR)

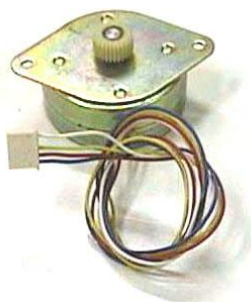
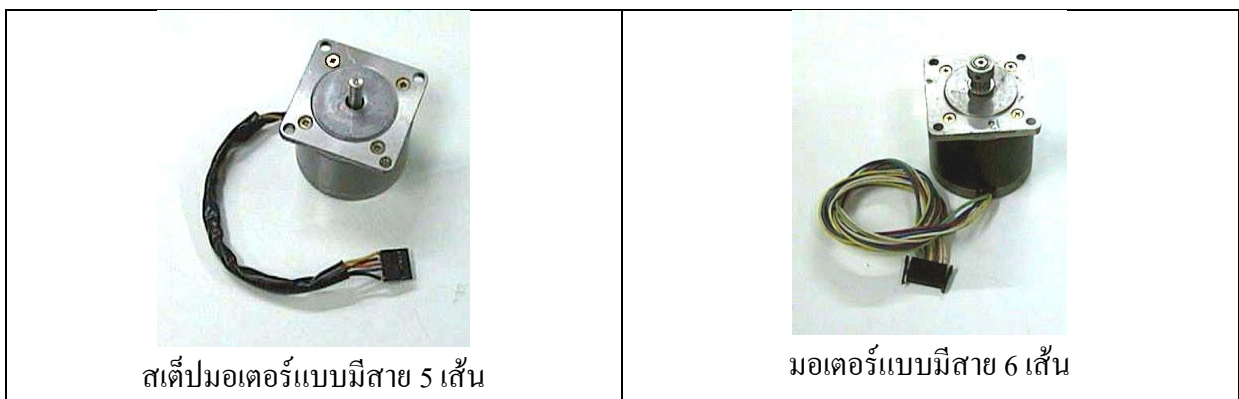
มอเตอร์ (MOTOR)

- Step Stepper Motor
- DC Motors
- DC Servo Motor
- AC Servo Motor

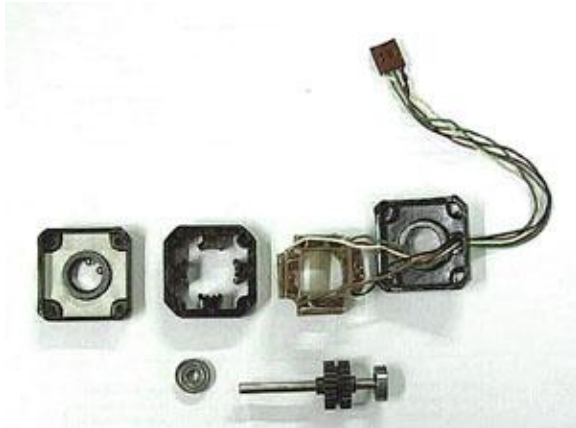
สเต็ปมอเตอร์ (STEPPING MOTOR)

สเต็ปมอเตอร์ เป็นมอเตอร์ที่ขับเคลื่อนด้วยพัลส์ ลักษณะการขับเคลื่อน จะหมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่อง แต่มีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะขับเคลื่อนได้ 1,1.5,1.8 หรือ 2 องศา แล้วแต่โครงสร้างของมอเตอร์ลักษณะที่ นำมอเตอร์ไปใช้ จะเป็นงานที่ต้องการตำแหน่งแม่นยำ เช่น ระบบขับเคลื่อนหัวแม่พิมพ์ในเครื่องพิมพ์ (PRINTER)ระบบขับเคลื่อนหัวอ่านในเครื่องอ่านบันทึกเหล็ก ระบบขับเคลื่อนตำแหน่งของปากกาใน X-Y PLOTTER เป็นต้น

ชนิดและโครงสร้างของสเต็ปมอเตอร์



สเต็ปมอเตอร์หลายแบบไบโพลาร์



โครงสร้างสเต็ปมอเตอร์

สเต็ปมอเตอร์แบบต่างๆ

สเต็ปมอเตอร์ มีส่วนประกอบที่สำคัญ 2 ส่วนคือ ROTOR (ส่วนที่หมุนได้) และ STATOR (ส่วนที่อยู่กับที่) STATOR จะประกอบด้วยขดลวดหลายขด ส่วน ROTOR จะเป็นแม่เหล็กถาวร หรือ อื่นๆ ซึ่งมีอยู่หลายแบบ เช่น

1. แบบแม่เหล็กถาวร (PERMANENT MAGNET_PM)

สเต็ปมอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆ โพล และมีโรเตอร์ (ROTOR) ทำด้วยแม่เหล็กถาวร รูปทรงเป็นกระบอกฟันเลื่อย เมื่อป้อนไฟกระแสตรง ให้กับขดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าผลักต่อโรเตอร์ ทำให้มอเตอร์หมุน มอเตอร์แบบ PM จะเกิดแรงจลน์ให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด ขนาดของ Step angle จะเป็น 1.8, 7.5, 15, 30, 45 และ 90 องศา

2. แบบแปรค่ารีลักแตนซ์ (VARIABLE RELUCTANCE- VR)

สเต็ปมอเตอร์แบบ VR โรเตอร์ทำจากสารเฟอร์โรแมกเนติกกำลังอ่อน หรือที่เรียกว่าเหล็กอ่อน มอเตอร์แบบนี้ ในขณะที่ไม่จ่ายพลังงานไฟฟ้าเข้ามอเตอร์ โรเตอร์สามารถหมุนได้อย่างอิสระ โรเตอร์มีลักษณะเป็นฟันเลื่อย รูปทรงกระบอกโดยจะมีความสัมพันธ์ โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะไปหมุนโรเตอร์ ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนซ์ต่ำที่สุด ตำแหน่งที่จะเกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุดดังนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างกันไป ก็ทำให้มอเตอร์ หมุนไปตำแหน่งต่างๆ กัน โรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

3. แบบผสม(HYBRID-H)

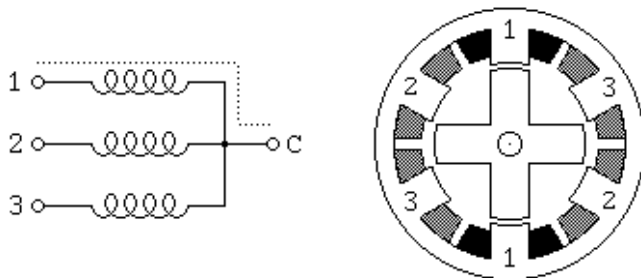
สเต็ปมอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างดี ทำให้ได้มุม การหมุนและครั้งน้อยและแม่นยำ ข้อดีก็คือ ให้แรงบิดสูงและมีขนาดกระทัดรัด และให้แรงจลนศาสตร์โรเตอร์นิ่งกับที่ตอนไม่จ่ายไฟ

ศัพท์เกี่ยวกับ สเต็ปมอเตอร์

- PHASE เป็นส่วนของขดลวด ระหว่างปลายสายกับ CENTER TAP หรือทั้งขดถ้าไม่มี CENTER TAP
- Phase Angle หมายถึงจำนวนองศาที่เกิดจากการหมุนในแต่ละ สเต็ป
- FULL STEP MODE หมายถึงการหมุนแบบ FULL STEP คือ การหมุนแต่ละสเต็ป จะได้มุมเท่ากับ Phase Angle
- HALF STEP MODE เป็นการหมุนแบบ HALF STEP การหมุนแต่ละสเต็ปจะได้มุม เป็น ครึ่งหนึ่งของ Phase Angle

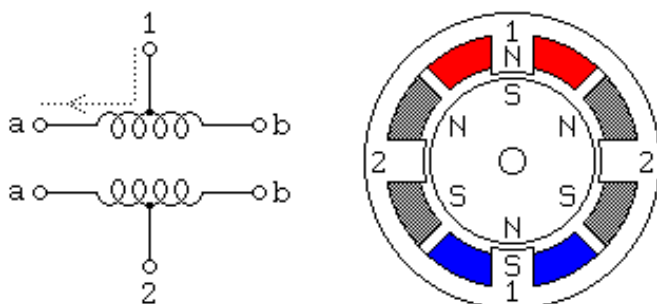
การทำงานของ Stepping Motor

Variable Reluctance Motors



Unipolar Motors

both Permanent magnet and hybrid stepping motors with 5 or 6 wires



การกระตุ้นเฟส (Phase Excitation)

การกระตุ้นเฟส หมายถึง การจ่ายกระแสไฟฟ้าเข้าที่ขดลวดของมอเตอร์ เพื่อให้มอเตอร์ทำงาน จากลักษณะการทำงานของสเต็ปมอเตอร์ การป้อนกระแสไฟฟ้าจึงต้องป้อนเป็น Pulse ให้กับมอเตอร์ ความเร็วในการหมุน ทิศทางของการหมุนของมอเตอร์นี้ จะขึ้นอยู่กับรูปแบบการป้อน Pulse วิธีการป้อน Pulse มี 3 วิธี ดังนี้

1. การกระตุ้นเฟสเดียว (1 Phase Excitation)

เป็นการให้กระแสไฟฟ้าไหลผ่านขดลวดครั้งละ 1 ขด แบบนี้กำลังงานสูญเสียจะต่ำ

2. การกระตุ้นสองเฟส (2 Phase Excitation)

เป็นการให้กระแสไฟฟ้าไหลผ่านขดลวดครั้งละ 2 ขดพร้อมกัน แบบนี้กำลังงานสูญเสียจะเป็นสองเท่าของแบบแรก

3. การกระตุ้นทั้งหนึ่งและสองเฟส (1-2 Phase Excitation)

เป็นการกระตุ้นด้วยแบบที่ 1 และ 2 สลับกัน แบบนี้จะได้การหมุนเป็นแบบ Half Step mode

การควบคุมให้ทำงานแบบ FULL STEP MODE สามารถกระทำได้สองวิธีคือ

เฟส	1	2	3	4
สแต็ป 1	1	1	0	0
สแต็ป 2	0	1	1	0
สแต็ป 3	0	0	1	1
สแต็ป 4	1	0	0	1

1 Phase Excitation

เฟส	1	2	3	4
สแต็ป 1	1	0	0	0
สแต็ป 2	0	1	0	0
สแต็ป 3	0	0	1	0
สแต็ป 4	0	0	0	1

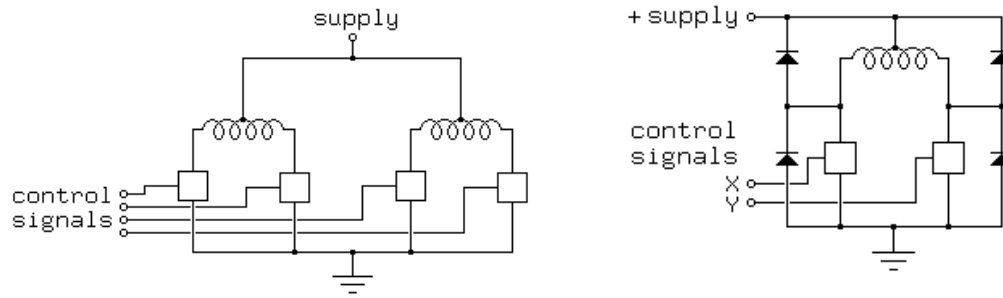
2 Phase Excitation

เมื่อ "1" หมายถึง มีกระแสไหล และ "0" หมายถึง ไม่มีกระแสไฟฟ้าไหล

การควบคุมให้ทำงานแบบ Half STEP MODE

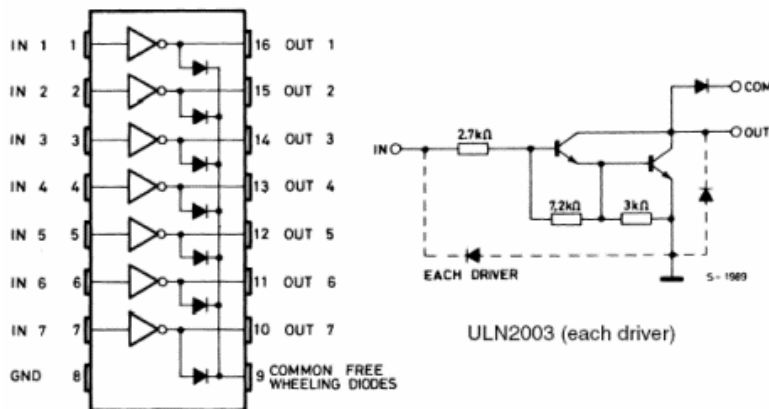
เฟส	1	2	3	4
สแต็ป 1	1	1	0	0
สแต็ป 2	1	0	0	0
สแต็ป 3	0	1	1	0
สแต็ป 4	0	1	0	0
สแต็ป 5	0	0	1	1
สแต็ป 6	0	0	1	0
สแต็ป 7	1	0	0	1
สแต็ป 8	0	0	0	1

Unipolar : Driving Circuits



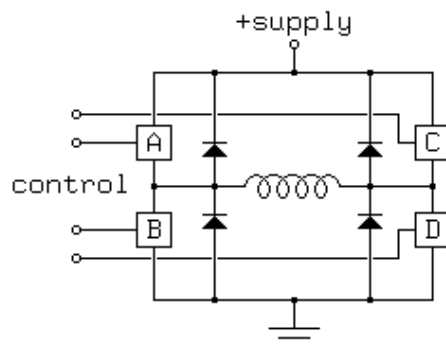
The boxes represents switches(or transistors) turning on and off motor coils.

ULN2003A Darlington Arrays



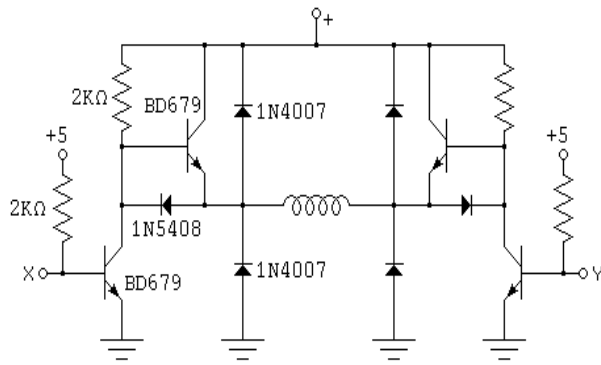
- 7 darlington transistors with TTL and CMOS compatible inputs
- ULN2803 : same as the ULN2003, except that it is in an 18-pin package, and contains 8 darlington
- $V_{CE(max)} = 50$ volts
- $I_C(max) = 500$ mA

Bipolar : Driving Circuits



H-bridge driver

- Forward mode, switches A and D closed.
- Reverse mode, switches B and C closed.



Practical H-bridge driver

Stepping Motor สำหรับการทดลอง

5V 4-phase Stepper Motor+ Driver Board ULN2003

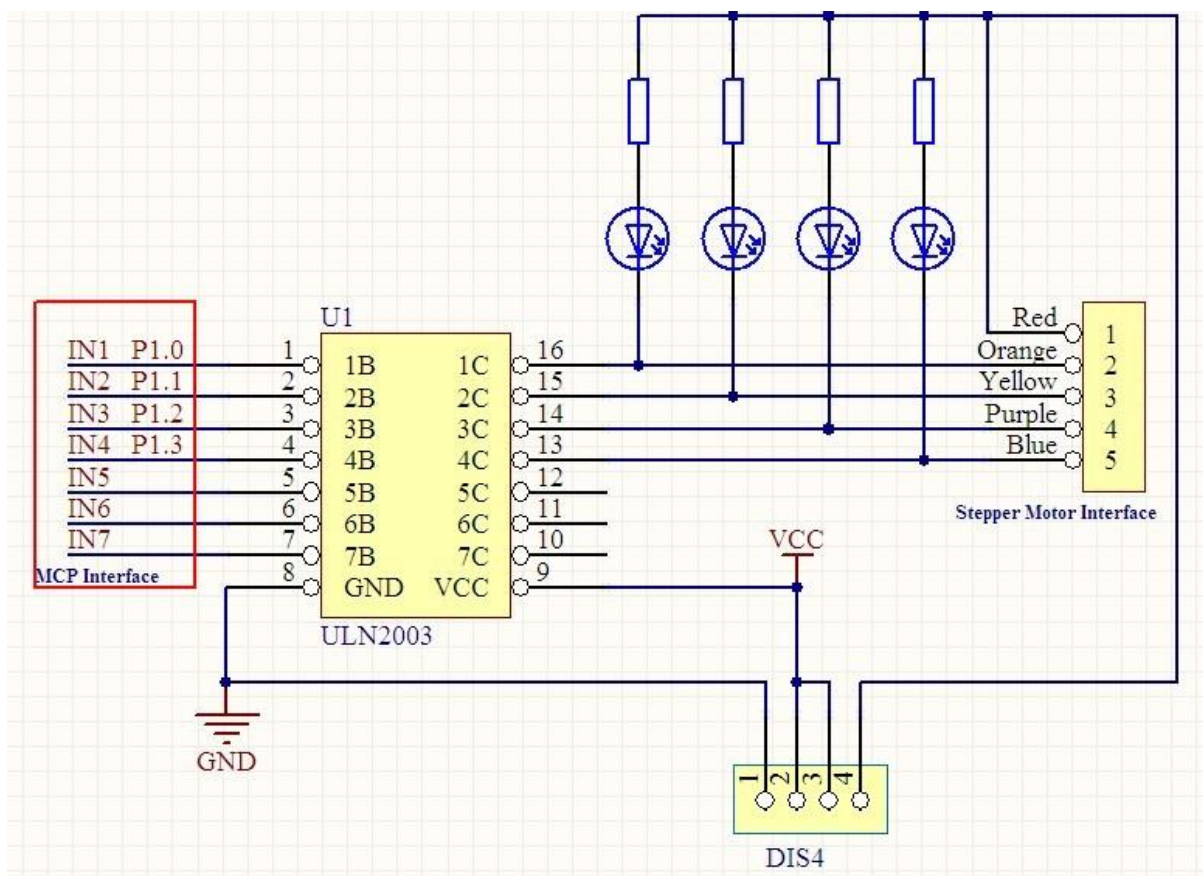


รายละเอียด

- 4 Phase 5 Wire Connection
- 100% Brand New
- Phase : 4
- Current : 160 mA per winding (320 mA in 4-step mode) Measured: 250mA stopped, 200 mA running fast
- Resistance : 31 Ω per coil winding (from Red wire to any coil)
- Voltage : 5V DC
- Step Angle (8-Step sequence: Internal Motor alone): 5.625° (64 steps per revolution)
- Step Angle (4-Step sequence: Internal Motor alone): 11.25° (32 steps per revolution)

- Gear Reduction ratio: 1 / 64 (Not really exact: probably 63.68395.:1)
- SO: it takes (64*64 = 4096 steps per output shaft revolution.. In 8-step sequence.
- SO: it takes (32*64 = 2048 steps per output shaft revolution.. In 4-step sequence.
- NOTE: Arduino "Stepper Library" runs in 4-step mode
- No-Load Pull-Out Frequency : 800pps
- No-Load Pull-In Frequency : 500pps
- Pull-In Torque : $\geq 78.4\text{mN.m}$
- Wiring Instruction : A (Blue), B (Pink), C (Yellow), D (Orange), E (Red, Mid-Point)
- Weight : 30g

uln2003 driver board circuit



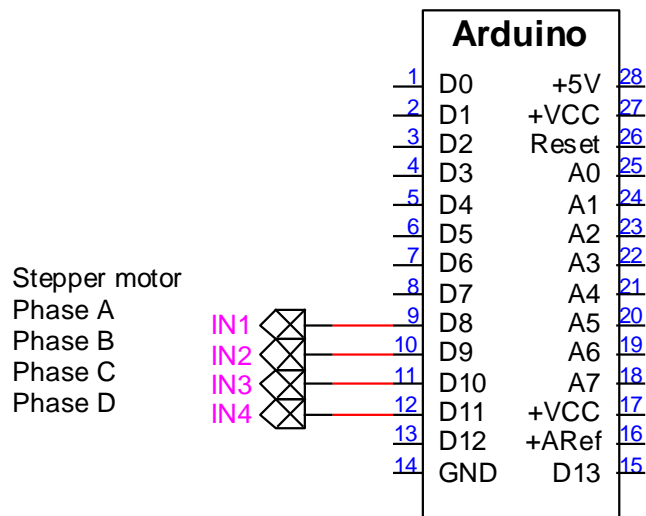
ข้อควรระวัง

USB USB 1.x และ 2.x จ่ายกระแสได้ 100 mA

USB USB 3.x จ่ายกระแสได้ 150 mA

การทดลอง

วงจร



โปรแกรม

```
#define phaseA 8
#define phaseB 9
#define phaseC 10
#define phaseD 11
byte step;
int i;

void setup() {
  pinMode(phaseA, OUTPUT); // sets the pin as output
  pinMode(phaseB, OUTPUT); // sets the pin as output
  pinMode(phaseC, OUTPUT); // sets the pin as output
  pinMode(phaseD, OUTPUT); // sets the pin as output
  step = 1;
}

void rotate(){
  switch (step) {
    case 1:
      digitalWrite(phaseA, HIGH);
      digitalWrite(phaseB, HIGH);
      digitalWrite(phaseC, LOW);
      digitalWrite(phaseD, LOW);
      break;
    case 2:
      digitalWrite(phaseA, LOW);
      digitalWrite(phaseB, HIGH);
      digitalWrite(phaseC, HIGH);
      digitalWrite(phaseD, LOW);
      break;
    case 3:
      digitalWrite(phaseA, LOW);
      digitalWrite(phaseB, LOW);
      digitalWrite(phaseC, HIGH);
      digitalWrite(phaseD, HIGH);
```



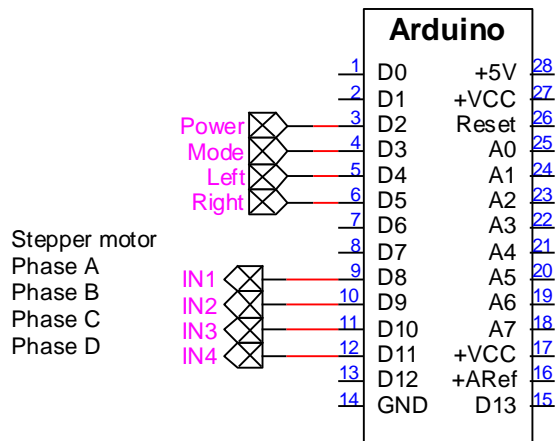
```

    break;
case 4:
    digitalWrite(phaseA, HIGH);
    digitalWrite(phaseB, LOW);
    digitalWrite(phaseC, LOW);
    digitalWrite(phaseD, HIGH);
    break;
default:
    break;
}
}
void ccw(){
    if(step<4) step++;
    else step = 1;
    rotate();
    delay(5);
}
void cw(){
    if(step>1) step--;
    else step = 4;
    rotate();
    delay(5);
}
void loop() {
    for (i=0; i <= 2048; i++){
        cw();
    }
    for (i=0; i <= 2048; i++){
        ccw();
    }
}

```

งานมอบหมาย





การทำงาน

เมื่อเปิดไฟ ระบบอยู่ใน Standby mode สไลด์อยู่ตำแหน่ง Home (ซ้ายสุด หรือ ขวาสุด)

Standby mode สวิตซ์ที่ใช้งานได้คือ

Power ใช้สำหรับเปิดการทำงาน เมื่อแตะสวิตซ์ Power ระบบจะเข้าสู่ Manual mode

Manual mode ควบคุมตำแหน่งสไลด์ได้ด้วย สวิตซ์ Left และ Right สวิตซ์ที่ใช้งานได้คือ

Power ใช้สำหรับปิดการทำงานให้กลับไปสู่ Standby mode

Left สั่งให้สไลด์เลื่อนไปทางซ้าย แต่ถ้าอยู่ตำแหน่งซ้ายสุดแล้วไม่ต้องเลื่อน

Right สั่งให้สไลด์เลื่อนไปทางขวา แต่ถ้าอยู่ตำแหน่งขวาสุดแล้วไม่ต้องเลื่อน

Mode สั่งให้การทำงานไปสู่ระบบอัตโนมัติ (Auto mode)

Auto mode สไลด์จะเลื่อนเองจากซ้ายสุดไปขวาสุด แล้วจะเลื่อนกลับ เป็นไปโดยอัตโนมัติ สวิตซ์ที่ใช้งานได้คือ

Power ใช้สำหรับปิดการทำงานให้กลับไปสู่ Standby mode

Mode สั่งให้การทำงานไปสู่ระบบ Manual (Manual mode)

คำแนะนำการเขียนโปรแกรม

กำหนดค่าตัวแปรและค่าคงที่ที่ต้องใช้

```
byte Operation_mode;
```

```
#define standby 0
```

```
#define manual 1
```

```
#define auto 2
```

```
#define phaseA 8
```

```
#define phaseB 9
```

```
#define phaseC 10
```

```
#define phaseD 11
```

```
Int position;
```

```
#define left false
#define right true
boolean direction = left;
```

ฟังก์ชัน Setup

```
void setup() {
    กำหนดหน้าที่ของขาสัญญาณต่างๆ
    กำหนด Operation_mode = Standby
    กำหนด position_mode = 0
    เคลื่อนสไลด์มาตำแหน่ง home
}
```

ฟังก์ชัน loop โปรแกรมหลัก

```
void loop() {
    switch (Operation_mode)
    {
        case standby:
            Standby_operation();
            break;
        case manual:
            Manual_operation();
            break;
        case auto:
            Auto_operation();
            break;
    }
}
```

ฟังก์ชัน Standby_operation

```
void Standby_operation(){
    if(position_sensor != home){
        เคลื่อนสไลด์ไปทาง home
    }
}
```

```
        if(sw_power_on) Operation_mode = manual;
    }
```

ฟังก์ชัน Manual_operation

```
void Manual_operation(){
    if(sw_left_on) shift_left();
    if(sw_right_on) shift_right();
    if(sw_power_on) Operation_mode = standby;
    if(sw_mode_on){
        direction = left;
        Operation_mode = auto;
    }
}
```

ฟังก์ชัน Auto_operation

```
void Manual_operation(){
    if(sw_mode_on) Operation_mode = manual;
    if(sw_power_on) Operation_mode = standby;
    if(direction == left){
        shift_left();
        if(sensor == สุดทาง) direction = right;
    }else{
        shift_right();
        if(sensor == สุดทาง) direction = left;
    }
}
```

หมายเหตุ เบื้องต้นใช้การนับ (Counter) แทน sensor ถ้า counter = 0 แสดงว่าซ้ายสุด ถ้า counter = xxx แสดงว่าขวาสุด

เพิ่มเติม

<https://arduino-info.wikispaces.com/SmallSteppers>

<https://www.youtube.com/watch?v=TWMai3oirnM>