ใบงานที่ 15

การสั่งการ และควบคุมอุปกรณ์ Actuators

จุดประสงค์การเรียนรู้

1.ศึกษาการควบคุมมอเตอร์ 2 ทิศทางด้วย L9110

2.ศึกษาการควบคุมการเปิด/ปิดอุปกรณ์ไฟฟ้าด้วยรีเลย์

3.ศึกษาการควบคุมเซอร์โวมอเตอร์ด้วยจอยสติ๊ก

4.ศึกษาการควบคุมสเต็ปมอเตอร์ด้วยโมดูลขับ ULN2003

เครื่องมือและอุปกรณ์การทดลอง

1.เครื่องไมโครคอมพิวเตอร์

2.บอร์ดไมโครคอนโทรลเลอร์ ESP32

3.โปรแกรการทดลอง

4.อุปกรณ์อิเล็กทอรนิกส์สำหรับทดลอง

การทดลองที่ 1 การควบคุมมอเตอร์ 2 ทิศทางด้วย L9110

อุปกรณ์ในการทดลอง

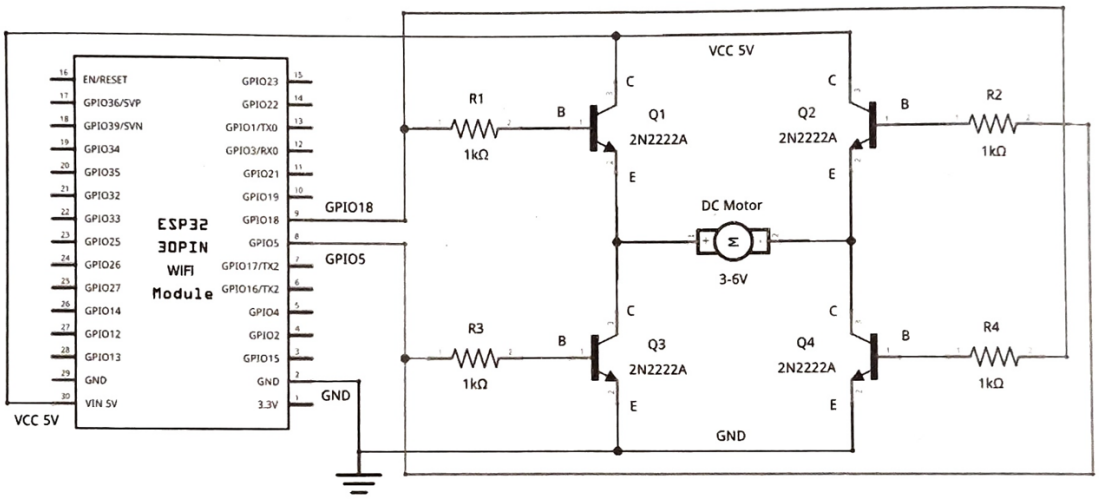
1.บอรด์ NodeMCU ESP32

2.โมดูล Motor DC ด้วย L9110

3.แผงต่อวงจร

4.สายไฟต่อวงจร

ประกอบวงจรตามรูป




```
const int aPin = 18;          //ประกาศตัวแปร aPin ที่ต่ออยู่กับขา D18/GPIO18
const int bPin = 5;           //ประกาศตัวแปร bPin ที่ต่ออยู่กับขา D5/GPIO5
void setup() {
pinMode(aPin, OUTPUT);        //กำหนดให้ตัวแปร aPin หรือขา D18/GPIO18 เป็น Output
pinMode(bPin, OUTPUT);        //กำหนดให้ตัวแปร bPin หรือขา D5/GPIO5 เป็น Output
}
```

```
void loop() {
digitalWrite(aPin, HIGH);        //กำหนดให้ขา D18/GPIO18 มีสถานะเป็น HIGH ทำให้ทรานซิสเตอร์ Q1 และ Q4 ทำงาน มอเตอร์จะหมุนตามเข็มนาฬิกา
digitalWrite(bPin, LOW);         //กำหนดให้ขา D5/GPIO5 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์ Q2 และ Q3 ไม่ทำงาน
delay(5000);                     //หน่วงรอเป็นเวลา 5 วินาที

digitalWrite(aPin, LOW);         //กำหนดให้ขา D18/GPIO18 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์ Q1 และ Q4 ไม่ทำงาน (มอเตอร์หยุดหมุน)
digitalWrite(bPin, LOW);         //กำหนดให้ขา D5/GPIO5 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์ Q2 และ Q3 ไม่ทำงาน (มอเตอร์หยุดหมุน)
delay(2000);                     //หน่วงรอเป็นเวลา 2 วินาที

digitalWrite(aPin, LOW);         //กำหนดให้ขา D18/GPIO18 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์ Q1 และ Q4 ไม่ทำงาน
digitalWrite(bPin, HIGH);        //กำหนดให้ขา D5/GPIO5 มีสถานะเป็น HIGH ทำให้ทรานซิสเตอร์ Q2 และ Q3 ทำงาน มอเตอร์จะหมุนทวนเข็มนาฬิกา
delay(5000);                     //หน่วงรอเป็นเวลา 5 วินาที
digitalWrite(aPin, LOW);         //กำหนดให้ขา D18/GPIO18 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์ Q1 และ Q4 ไม่ทำงาน (มอเตอร์หยุดหมุน)
digitalWrite(bPin, LOW);         //กำหนดให้ขา D5/GPIO5 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์ Q2 และ Q3 ไม่ทำงาน (มอเตอร์หยุดหมุน)
delay(2000);                     //หน่วงรอเป็นเวลา 2 วินาที
}
```

บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 2 การควบคุมการเปิด/ปิดอุปกรณ์ไฟฟ้าด้วยรีเลย์

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

2.โมดูลรีเลย์

3.ชุดหลอดไฟ ขั้วหลอด และปลั๊กไฟ 220 VAC

4.ทรานซิสเตอร์ ชนิด NPN เบอร์ 2N2222

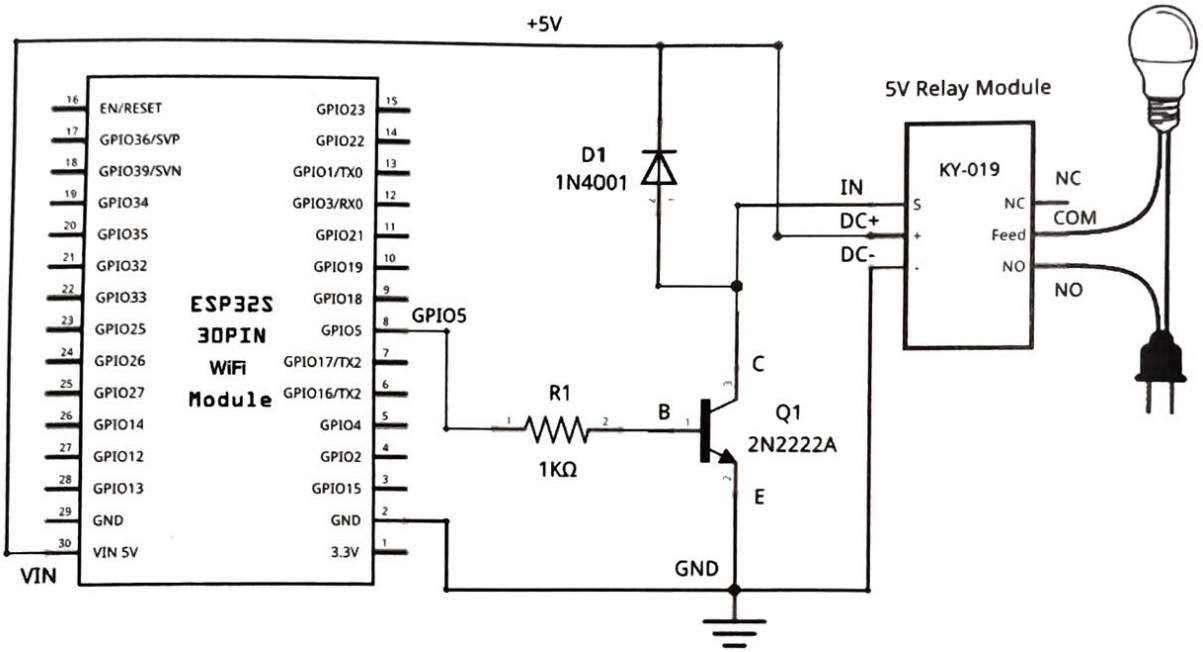
5.ตัวต้านทาน 1kOhm

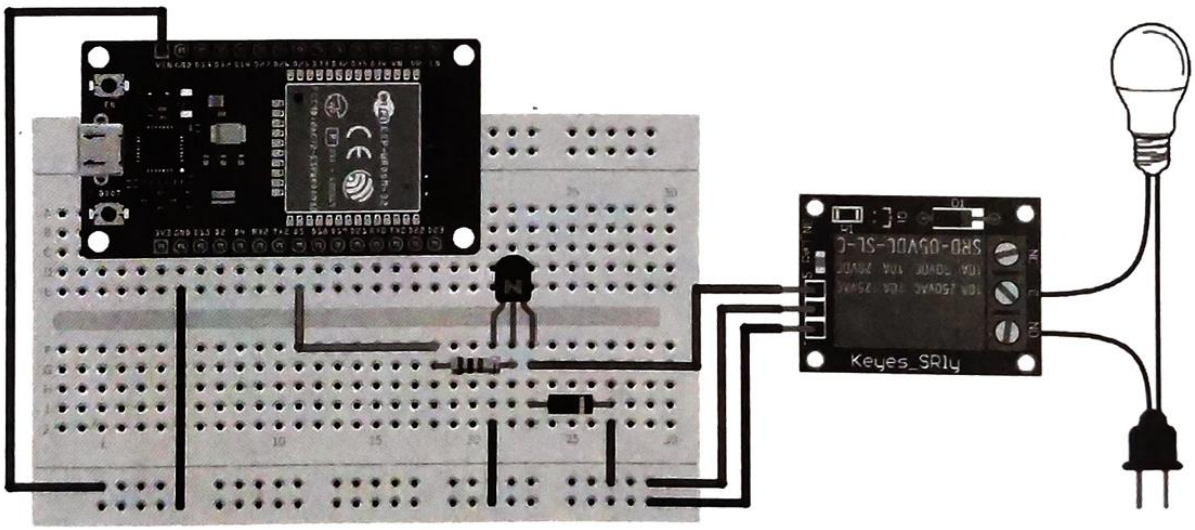
6.ไดโอดเบอร์ 1N4001

7.แผงต่อวงจร

8.สายไฟต่อวงจร

ประกอบวงจรตามรูป





```
void setup() {
  pinMode(5, OUTPUT);              //กำหนดให้ขา D5/GPIO5 เป็น Output
  Serial.begin(115200);
  delay(100);
  Serial.println("Control Relay for ON/OFF Lamp");      //แสดงข้อความ " "
                                                        ออกทาง Serial Monitor
  delay(2000);                     //หน่วงรอเป็นเวลา 2 วินาที
}
void loop() {
  digitalWrite(5, HIGH);           //ป้อนสถานะ HIGH ที่ขา D5/GPIO5 ทำให้หลอดไฟติด
  Serial.println("Relay SW/ON ---> Lamp ON");          //แสดงข้อความ " " ออกทาง
                                                       Serial Monitor เพื่อตรวจสอบ
  delay(2000);                     //หน่วงรอเป็นเวลา 2 วินาที
  digitalWrite(5, LOW);            //ป้อนสถานะ LOW ที่ขา D5/GPIO5 ทำให้หลอดไฟดับ
  Serial.println("Relay SW/OFF ---> Lamp OFF");        //แสดงข้อความ " " ออกทาง
                                                       Serial Monitor เพื่อตรวจสอบ
  delay(2000);                     //หน่วงรอเป็นเวลา 2 วินาที
}
```

บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 3 การควบคุมเซอร์โวมอเตอร์ด้วยจอยสติ๊ก

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

2.เซอร์โวมอเตอร์ SG90

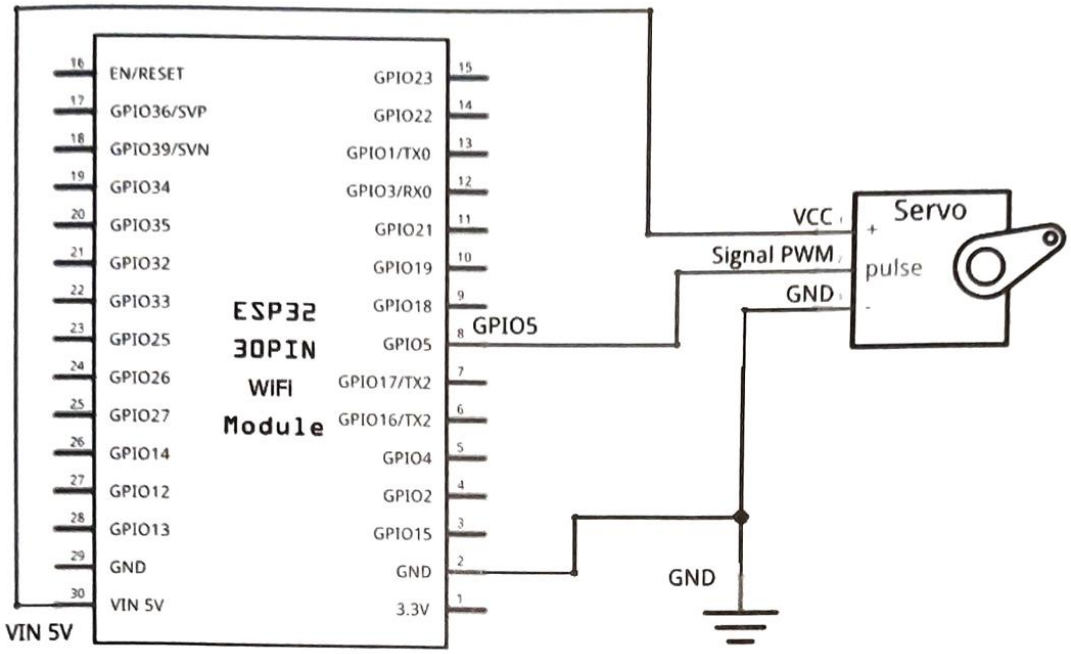
3.โมดูล PS2 XY Joystick

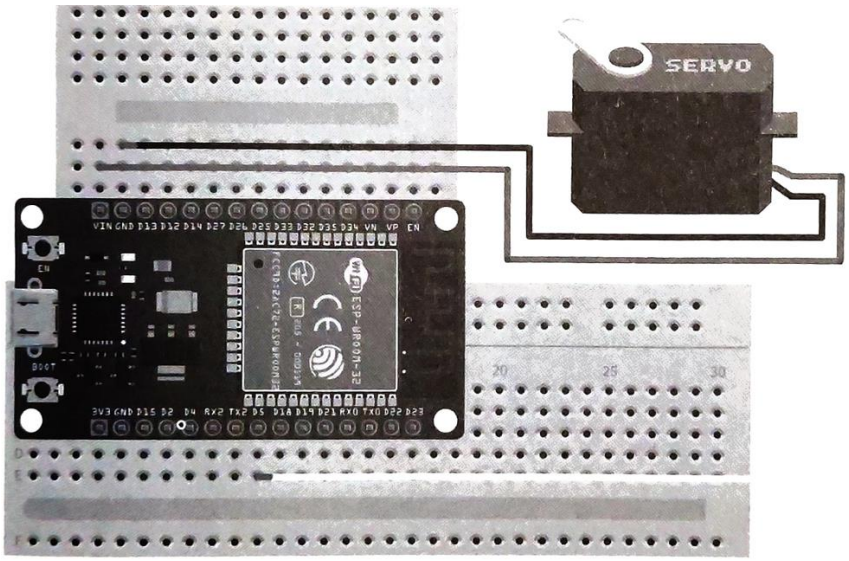
4.ทรานซิสเตอร์ ชนิด NPN เบอร์ 2N2222

5.แผงต่อวงจร

6.สายไฟต่อวงจร

ประกอบวงจรตามรูป



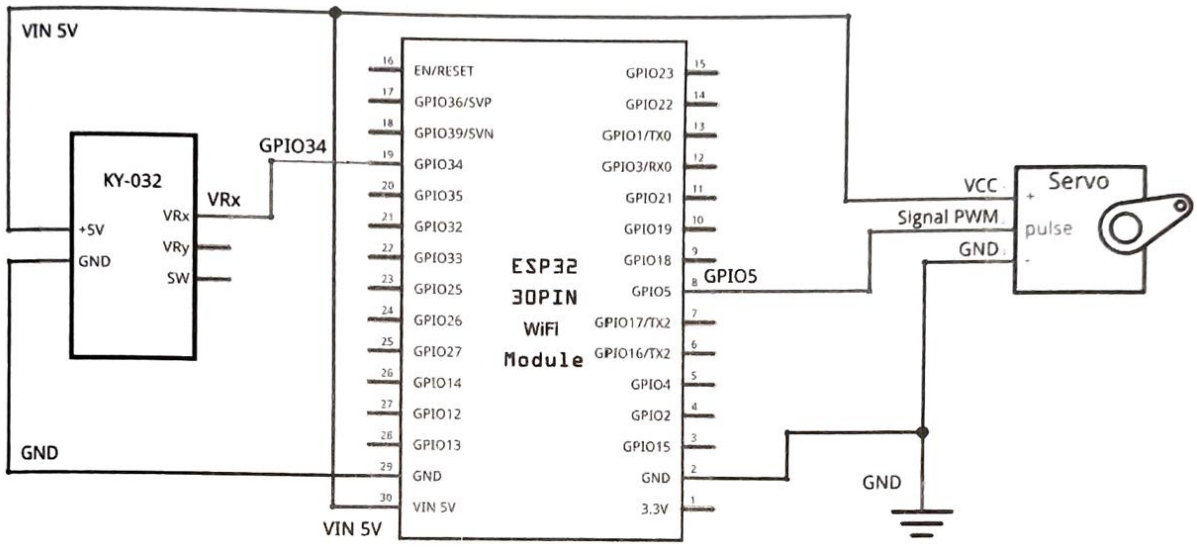


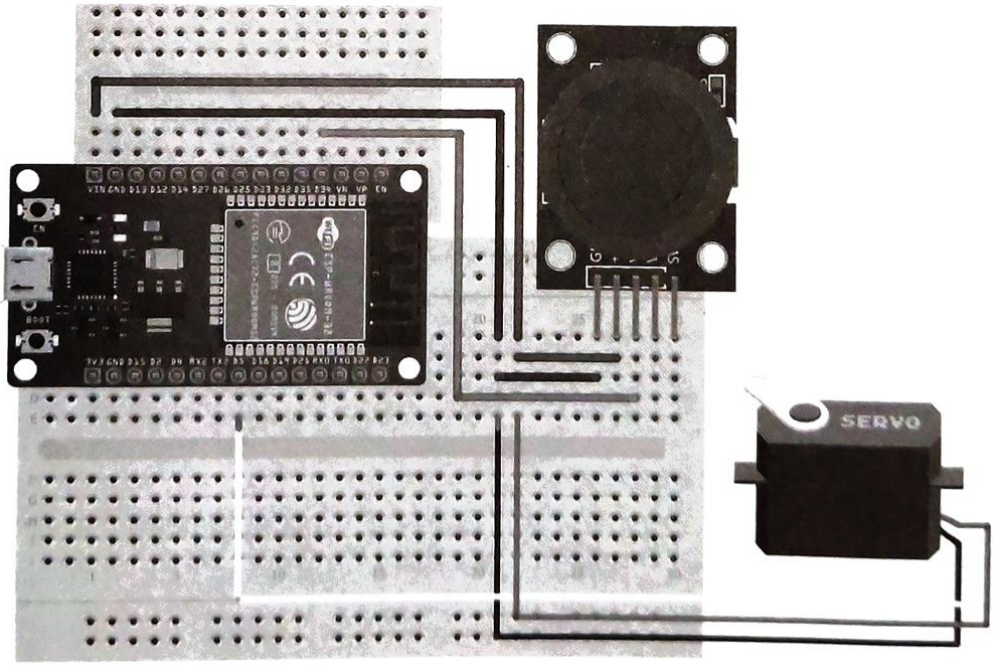
หลังจากเชื่อมต่อสายสัญญาณต่างๆ เรียบร้อยแล้ว ขั้นตอนต่อไปเราจะต้องติดตั้งไลบรารี่ (Library) ที่จำเป็น ซึ่งในที่นี้คือไลบรารี่ Servo.h ให้กับ Arduino IDE เสียก่อน โดยเราสามารถเข้าไปดาวน์โหลดมาติดตั้งได้ที่ https://github.com/RoboticsBrno/ServoESP32 และเมื่อได้เป็นไฟล์ .zip มาแล้ว เราสามารถที่จะ ใช้เมนู Sketch - Include Library - Add .ZIP Library... ช่วยในการติดตั้ง หรือจะแตก zip แล้วก็อปปี้โฟลเดอร์ไปวางไว้ที่ C:\Users\ชื่อผู้ใช้งาน\Documents\Arduino\libraries ก็ได้ หลังจากติดตั้งไลบรารี่ (Library) เสร็จแล้ว ก็ลงมือเขียนโค้ดและอัพโหลดโปรแกรม ซึ่งมีรายละเอียดดังนี้

```
#include <Servo.h>                    //เรียกใช้ไลบรารี่ Servo.h เพื่อควบคุมเซอร์โวมอเตอร์
Servo myServo;                        //สร้างออบเจ็กต์จากคลาส Servo แล้วเก็บค่าลงใน myServo

void setup() {
  Serial.begin(115200);
  myServo.attach(5);                  //กำหนดให้ขา GPIO5 เป็นขาที่ใช้ควบคุมเซอร์โวมอเตอร์
}
void loop() {
  myServo.write(0);                   //หมุนแกนเซอร์โวไปที่ตำแหน่ง 0 องศา
  Serial.println("Servo : 0 Degree");  //แสดงข้อความใน " " ออกทาง Serial Monitor
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
  myServo.write(45);                  //หมุนแกนเซอร์โวไปที่ตำแหน่ง 45 องศา
  Serial.println("Servo: 45 Degree");  //แสดงข้อความใน " " ออกทาง Serial Monitor
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
  myServo.write(90);                  //หมุนแกนเซอร์โวไปที่ตำแหน่ง 90 องศา
  Serial.println("Servo: 90 Degree");  //แสดงข้อความใน " " ออกทาง Serial Monitor
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
  myServo.write(135);                 //หมุนแกนเซอร์โวไปที่ตำแหน่ง 135 องศา
  Serial.println("Servo: 135 Degree"); //แสดงข้อความใน " " ออกทาง Serial Monitor
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
  myServo.write(180);                 //หมุนแกนเซอร์โวไปที่ตำแหน่ง 180 องศา
  Serial.println("Servo: 180 Degree"); //แสดงข้อความใน " " ออกทาง Serial Monitor
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
}
```

ทีนี้ก็ถึงเวลาที่เราจะมาทดสอบโดยการนำเอาโมดูล PS2 XY Joystick เข้ามาช่วยเป็นตัวบังคับแกนหมุน ของเซอร์โวมอเตอร์ให้หมุนไปในทิศทางและตำแหน่งองศาต่างๆ กัน เริ่มต้นจากวงจรเดิมให้เราเพิ่มการเชื่อมต่อ โมดูลจอยสติ๊กเข้าไป ตามรูป โดยให้เชื่อมต่อขา VRX ของโมดูล PS2 XY Joystick เข้ากับขา GPIO34/ADC1_6 ของบอร์ด ESP32 ส่วนขา +5V ก็ให้ต่อเข้ากับขา VIN ของบอร์ด และขา GND ก็ให้ต่อเข้ากับขา GND ของบอร์ด






```
#include <Servo.h>                  //เรียกใช้ไลบรารี่ Servo.h เพื่อควบคุมเซอร์โวมอเตอร์
#define JOYSTICK_X 34               //ประกาศตัวแปร JOYSTICK_X ที่ต่ออยู่กับขา GPIO34
#define SERVO_PIN 5                 //ประกาศตัวแปร SERVO_PIN ที่ต่ออยู่กับขา D5/GPIO5
Servo myServo;                      //สร้างออบเจ็กต์จากคลาส Servo แล้วเก็บค่าลงใน myServo
void setup() {
  pinMode(SERVO_PIN, OUTPUT);       //กำหนดให้ขา D5 หรือ GPIO5 เป็น Output
  myServo.attach(SERVO_PIN);        //กำหนดให้ขา GPIO5 เป็นขาที่ใช้ควบคุมเซอร์โวมอเตอร์
  myServo.write(90);                //หมุนแกนเซอร์โวไปที่ตำแหน่งเริ่มต้นคือ 90 องศา
                                    (เป็นองศาเดียวกันกับคันโยกบนจอยสติ๊กที่จะต้องอยู่
                                    ตรงกึ่งกลาง)
  Serial.begin(115200);
  delay(200);                       //หน่วงรอเป็นเวลา 0.2 วินาที
}
```

```
void loop() {
  int dataX = analogRead(JOYSTICK_X);          //อ่านค่าอนาล็อกจากขา GPIO34/ADC1_6
                                               มาเก็บไว้ที่ตัวแปร dataX
  int angle = map(dataX, 0, 4095, 0, 180);     //ใช้ฟังก์ชั่น map เปลี่ยนช่วงค่าอนาล็อกที่ได้
                                               รับจาก 0-4095 ไปเป็นค่าองศา 0-180 แล้วนำค่าที่ได้
                                               ไปเก็บไว้ในตัวแปร angle
  myServo.write(angle);          //หมุนแกนเซอร์โวไปตามค่าองศาที่อยู่ในตัวแปร angle
  Serial.println(angle);         //แสดงค่าองศาที่อยู่ในตัวแปร angle ออกทาง Serial Monitor
                                 เพื่อใช้ตรวจสอบตำแหน่งองศาของคันโยก
  delay(200);                    //หน่วงรอเป็นเวลา 0.2 วินาที
}
```

บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 4 การควบคุมสเต็ปมอเตอร์ด้วยโมดูลขับ ULN2003

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

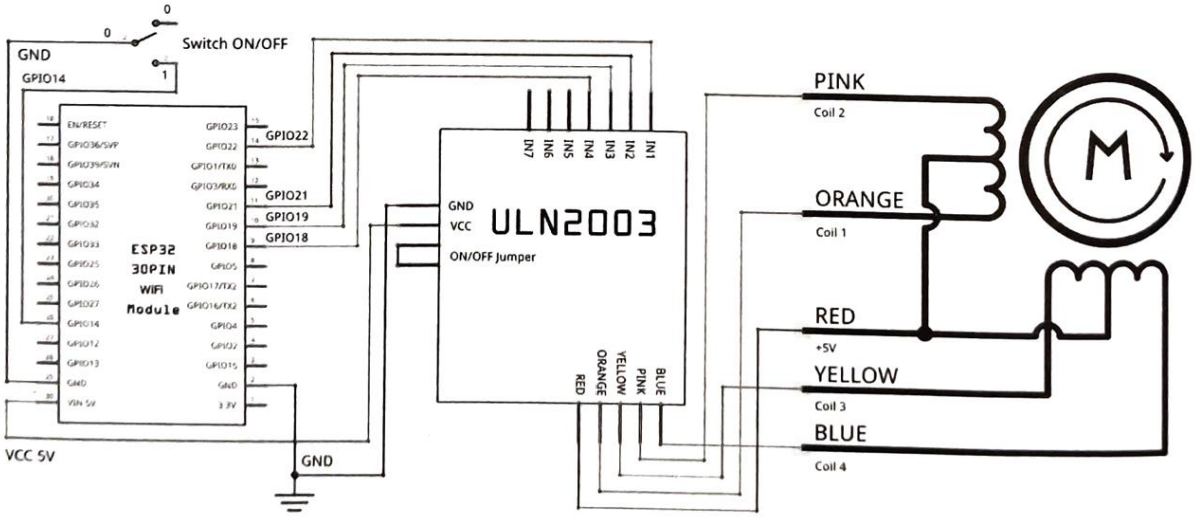
2.สเต็ปเปอร์มอเตอร์ พร้อมชุดขับ ULN2003

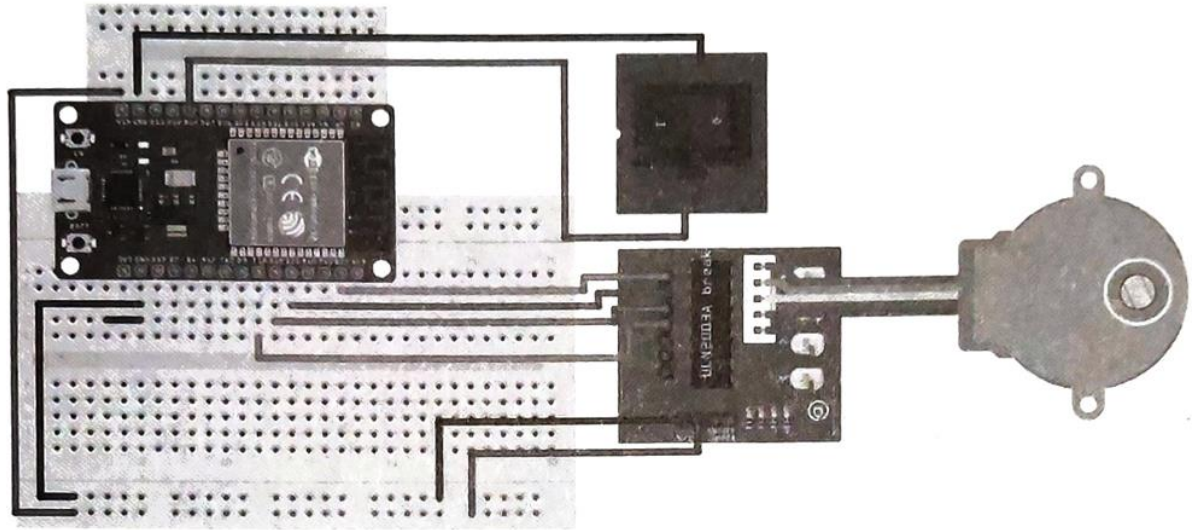
3.สวิตช์เปิด/ปิด

4.แผงต่อวงจร

5.สายไฟต่อวงจร

ประกอบวงจรตามรูป





```
#include <Stepper.h>                    //เรียกใช้ไลบรารี่ Stepper.h เพื่อควบคุมสเต็ปมอเตอร์
#define RETURN_PIN 14                   //ประกาศตัวแปร RETURN_PIN ที่ต่ออยู่กับขา D14/GPIO14
const int stepsPerRevolution = 2048;    //กำหนดให้จำนวนสเต็ปต่อรอบการหมุนมีค่าเป็น 2048
```

```
#define IN1 22                                  //กำหนดตัวแปรที่เชื่อมต่ออยู่กับขา GPIO22
#define IN2 21                                  //กำหนดตัวแปรที่เชื่อมต่ออยู่กับขา GPIO21
#define IN3 19                                  //กำหนดตัวแปรที่เชื่อมต่ออยู่กับขา GPIO19
#define IN4 18                                  //กำหนดตัวแปรที่เชื่อมต่ออยู่กับขา GPIO18

Stepper myStepper(stepsPerRevolution, IN1, IN3, IN2, IN4);  //สร้างออบเจ็กต์จาก
                                คลาส Stepper แล้วเก็บค่าไว้ในตัวแปร myStepper โดยผ่าน
                                ค่าพารามิเตอร์ จำนวนสเต็ปต่อรอบการหมุน, ตัวแปรที่เก็บตำแหน่ง
                                ขา GPIO ของบอร์ดที่ใช้เชื่อมต่อกับโมดูลขับเพื่อควบคุมมอเตอร์
 void setup() {

   pinMode(RETURN_PIN, INPUT_PULLUP);     //กำหนดให้ขา GPIO14 เป็น Input และมีสถานะเป็น HIGH

   myStepper.setSpeed(10);                //กำหนดความเร็วในการหมุนเป็นจำนวนรอบต่อนาที (rpm)
   Serial.begin(115200);
 }
void loop() {

  int switchStatus = digitalRead(RETURN_PIN);      //อ่านค่าดิจิตอลอินพุตจากขา GPIO14
                                                   มาเก็บไว้ที่ตัวแปร switchStatus
  if(switchStatus == LOW){                 //ถ้ามีการเปิดสวิทช์ ค่าในตัวแปร switchStatus จะมีค่า
                                           เป็น 0 หรือ LOW (แต่ถ้าปิดสวิทช์จะเป็น 1 หรือ HIGH)
    Serial.println("clockwise 360 ");  //แสดงข้อความใน " " ออกทาง Serial Monitor
    myStepper.step(stepsPerRevolution);  //หมุนไปตามจำนวนสเต็ปต่อรอบการหมุนที่กำหนดไว้ใน
                                         ตัวแปรคือ 2048 หรือ 1 รอบ (360°) ในทิศทางตามเข็มนาฬิกา (+)
    delay(1000);
    Serial.println("counterclockwise 90 "); //แสดงข้อความใน " " ออกทาง Serial Monitor
    myStepper.step(-512);                //หมุนไปตามจำนวนสเต็ปต่อรอบการหมุนที่กำหนดคือ 512
                                         หรือ 90° ในทิศทางทวนเข็มนาฬิกา (-)
     delay(500);
     myStepper.step(-512);
     delay(500);
     myStepper.step(-512);
     delay(500);
     myStepper.step(-512);
     delay(500);

  }
```

```
  else {                              //ถ้าปิดสวิทช์ ค่าในตัวแปร switchStatus จะมีค่าเป็น 1 หรือ HIGH
    Serial.println("counterclockwise 360 ");  //แสดงข้อความใน " " ออก Serial Monitor
    myStepper.step(-stepsPerRevolution);      //หมุนไปตามจำนวนสเต็ปต่อรอบการหมุนที่กำหนด
                                              ไว้ในตัวแปรคือ 2048 หรือ 1 รอบ (360°) ในทิศทางทวนเข็มนาฬิกา (-)
    delay(1000);
    Serial.println("clockwise 90 ");   //แสดงข้อความใน " " ออกทาง Serial Monitor
    myStepper.step(512);                      //หมุนไปตามจำนวนสเต็ปต่อรอบการหมุนที่กำหนด
                                              คือ 512 หรือ 90° ในทิศทางตามเข็มนาฬิกา (+)
    delay(500);
    myStepper.step(512);
    delay(500);
    myStepper.step(512);
    delay(500);
    myStepper.step(512);
    delay(500);
  }
}
```

จากการทดลองจะเห็นว่า เริ่มต้นที่ขาอินพุต GPIO14 จะถูกกำหนดให้มีสถานะเป็น HIGH ด้วยการเรียก ใช้งานวงจร Pull-up ที่อยู่ภายใน หากในขณะนั้นสวิทช์ปิด (Off) อยู่ จะทำให้สถานะอินพุตที่ขา GPIO14 ยังคง เป็น HIGH เมื่อเข้าเงื่อนไขตรวจสอบ จะแสดงข้อความสถานะการหมุนออกทาง Serial Monitor จากนั้นแกน หมุนของสเต็ปมอเตอร์ก็จะหมุนด้วยความเร็วคงที่ที่ 10 รอบต่อนาที (rpm) ไปในทิศทางทวนเข็มนาฬิกา เป็น จำนวนสเต็ปต่อรอบการหมุนที่ 2048 สเต็ป หรือ 1 รอบ (360°) หยุดรอเป็นเวลา 1 วินาที ต่อด้วยการหมุน กลับมาในทิศทางตามเข็มนาฬิกาครั้งละ 90° จนครบ 1 รอบ (360) เมื่อครบรอบคำสั่งจึงเข้าเงื่อนไขตรวจสอบ ถ้าสวิทช์ยังปิด (Off) อยู่ หรือที่ขาอินพุต GPIO14 ยังมีสถานะเป็น HIGH อยู่ ก็จะดำเนินการตามรอบคำสั่งเดิม ซ้ำอีกครั้ง แต่ถ้าตรวจสอบเงื่อนไขแล้วพบว่า สวิทช์เปิด (On) จะทำให้ขาอินพุต GPIO14 มีสถานะเป็น LOW ข้อความสถานะการหมุนจะถูกแสดงออกทาง Serial Monitor จากนั้นแกนหมุนของสเต็ปมอเตอร์ก็จะหมุนไป ในทิศทางตามเข็มนาฬิกาเป็นจำนวนสเต็ปต่อรอบการหมุนที่ 2048 สเต็ป หรือ 1 รอบ (360°) หยุดรอเป็นเวลา 1 วินาที แล้วต่อด้วยการหมุนกลับมาในทิศทางทวนเข็มนาฬิกาครั้งละ 90° จนครบ 1 รอบ (360°) เมื่อครบ รอบคำสั่งจึงเข้าเงื่อนไขตรวจสอบอีกครั้ง การทำงานของโปรแกรมจะหมุนวนในลักษณะนี้ไปเรื่อยๆ โดยมีสวิทช์ปิด-เปิดเป็นตัวกำหนดสถานะการทำงานของสเต็ปมอเตอร์

บันทึกผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....