

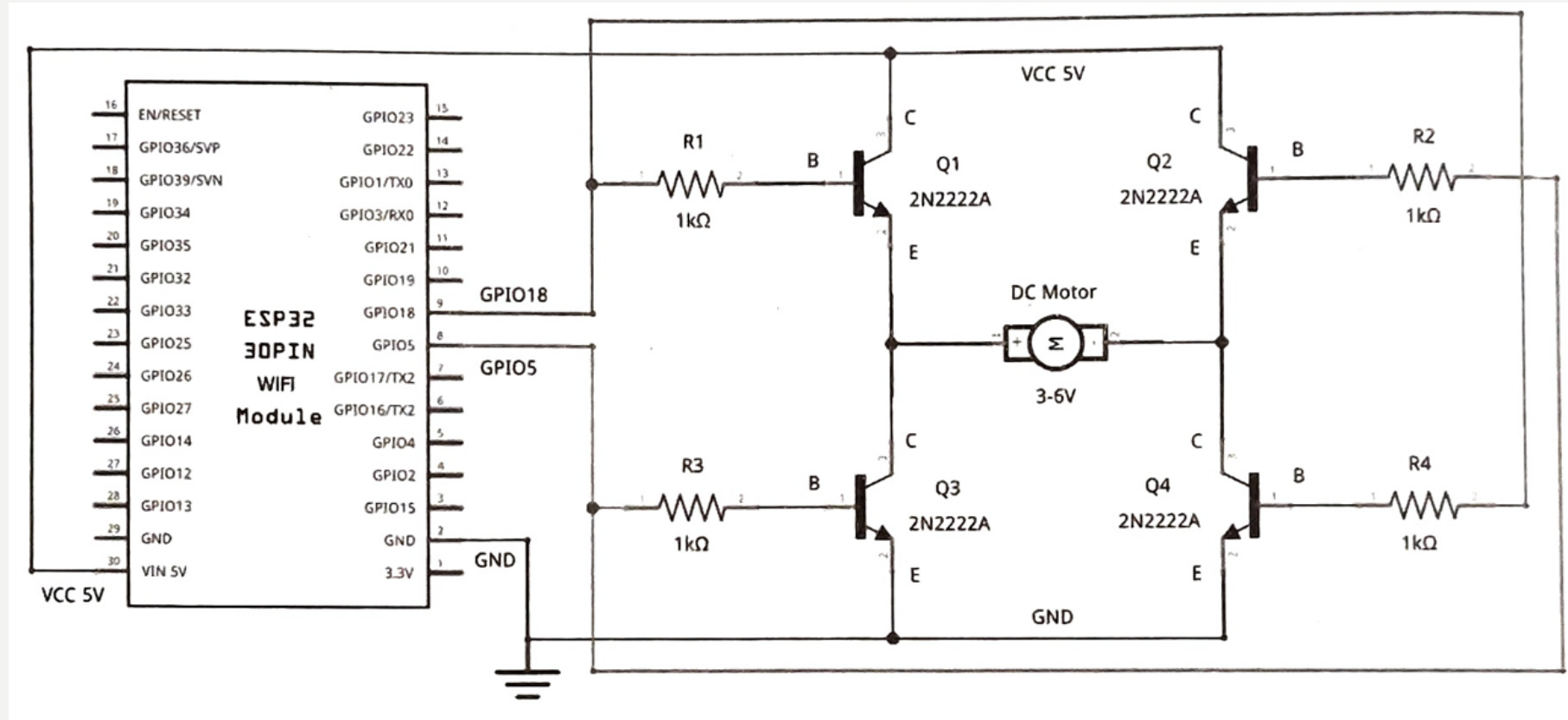
บทที่ 15

การสั่งการ และควบคุมอุปกรณ์ Actuators

จุดประสงค์การเรียนรู้

1. ศึกษาการควบคุมมอเตอร์ 2 ทิศทางด้วย L9110
2. ศึกษาการควบคุมการเปิด/ปิดอุปกรณ์ไฟฟ้าด้วยรีเลย์
3. ศึกษาการควบคุมเซอร์โวมอเตอร์ด้วยจอยสติ๊ก
4. ศึกษาการควบคุมสเต็ปมอเตอร์ด้วยไมโครลัมป์ ULN2003

การทดลองที่ 1 การควบคุมมอเตอร์ 2 ทิศทางด้วย L9110



```
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
}
```

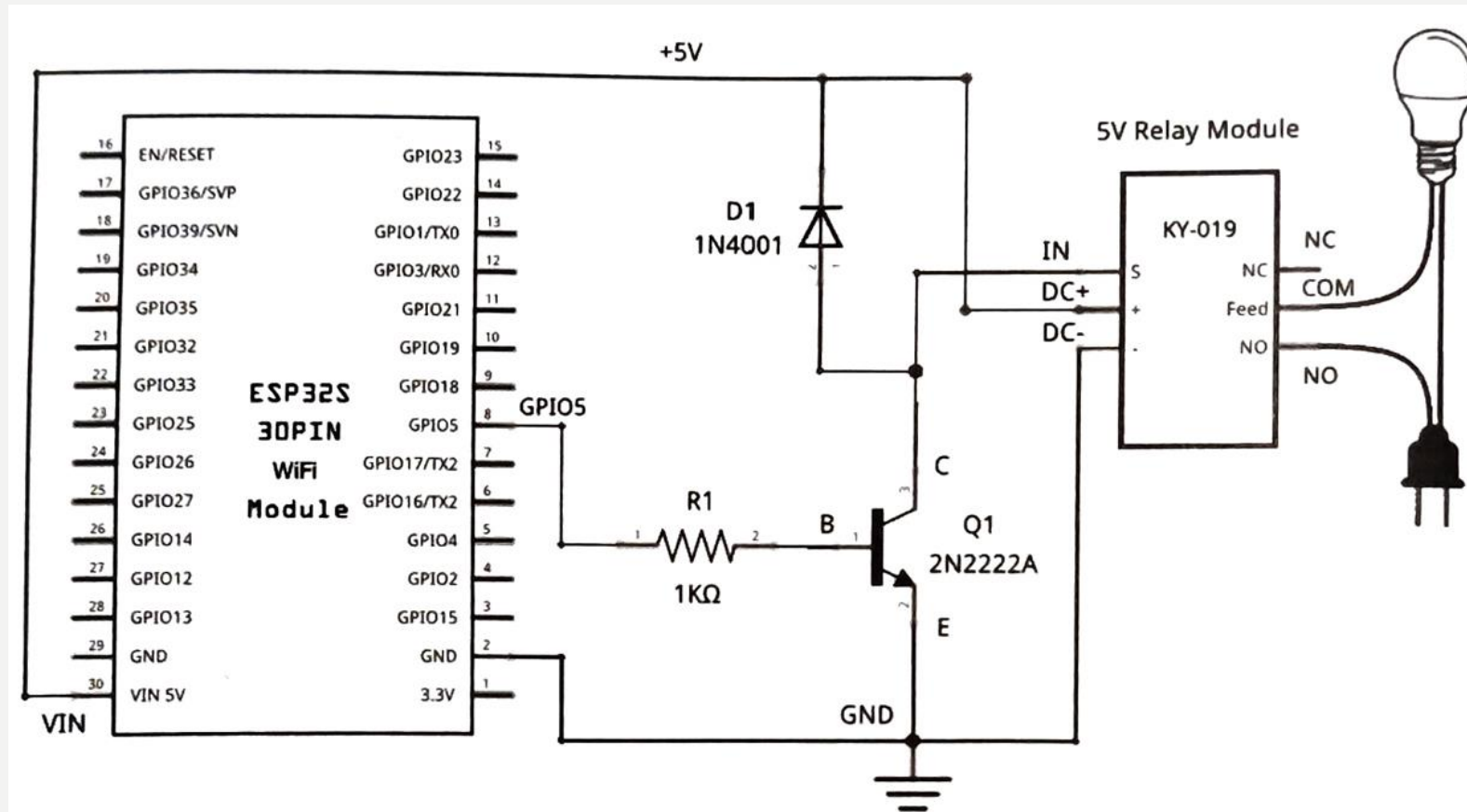
```
void loop() {
digitalWrite(aPin, HIGH); //กำหนดให้ขา D18/GPIO18 มีสถานะเป็น HIGH ทำให้ทรานซิสเตอร์
                           Q1 และ Q4 ทำงาน มอเตอร์จะหมุนตามเข็มนาฬิกา
digitalWrite(bPin, LOW);  //กำหนดให้ขา D5/GPIO5 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์
                           Q2 และ Q3 ไม่ทำงาน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

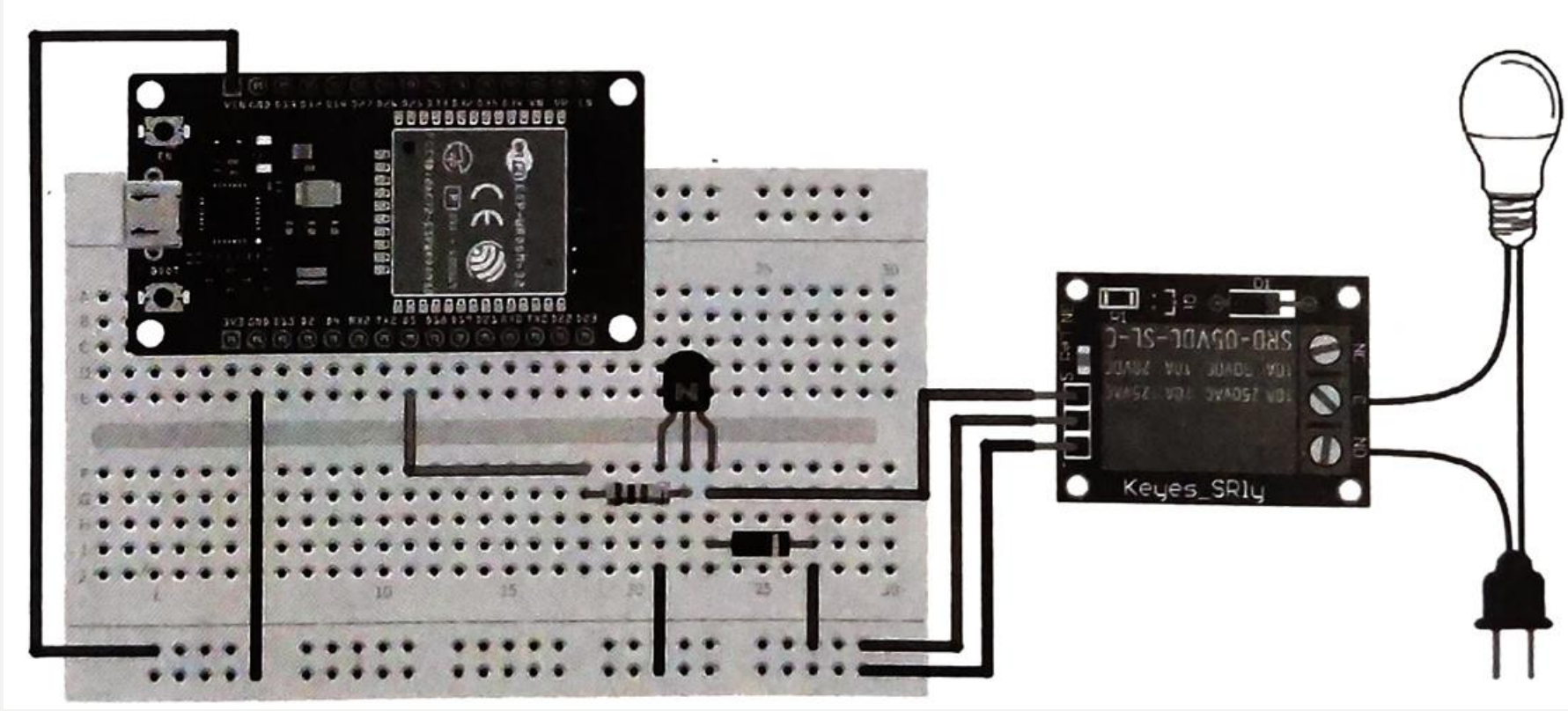
digitalWrite(aPin, LOW);  //กำหนดให้ขา D18/GPIO18 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์
                           Q1 และ Q4 ไม่ทำงาน (มอเตอร์หยุดหมุน)
digitalWrite(bPin, LOW);  //กำหนดให้ขา D5/GPIO5 มีสถานะเป็น LOW ทำให้ทรานซิสเตอร์
                           Q2 และ Q3 ไม่ทำงาน (มอเตอร์หยุดหมุน)
delay(2000);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

การทดลองที่ 2 การควบคุมการเปิด/ปิดอุปกรณ์ไฟฟ้าด้วยรีเลย์

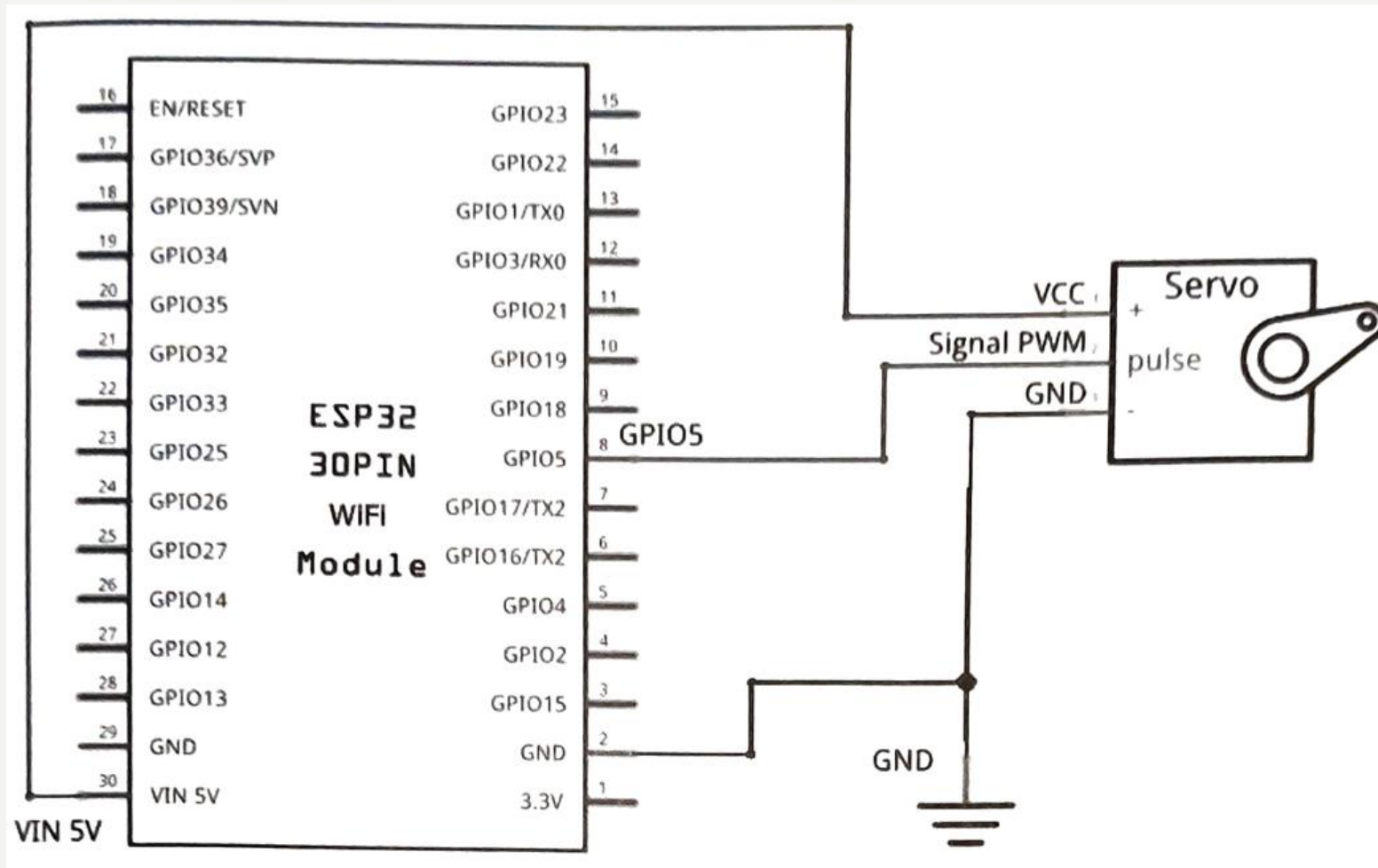


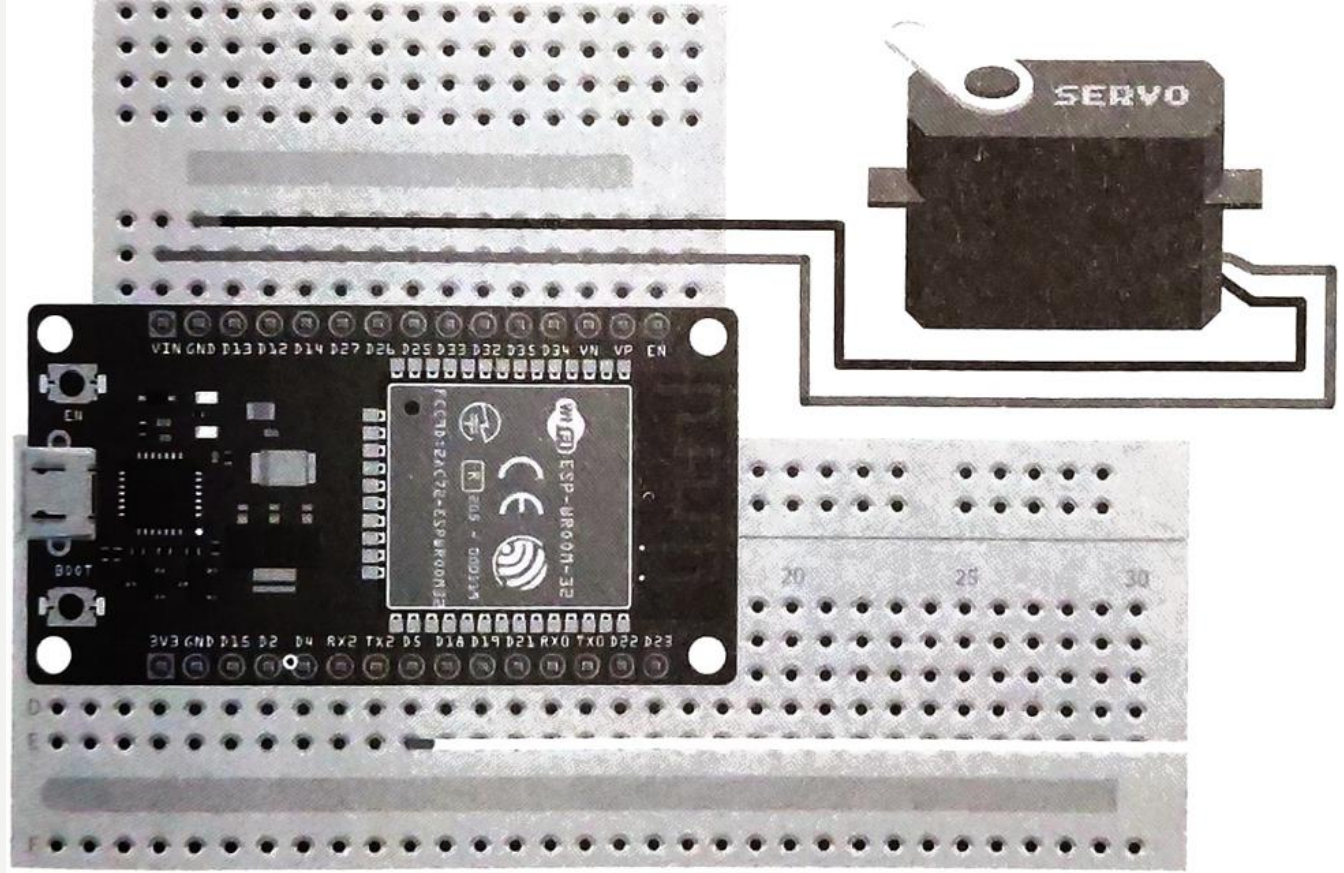



```
void setup() {  
  pinMode(5, OUTPUT);           //กำหนดให้ขา D5/GPIO5 เป็น Output  
  Serial.begin(115200);  
  delay(100);  
  Serial.println("Control Relay for ON/OFF Lamp"); //แสดงข้อความ " "  
                                     //ออกทาง Serial Monitor  
  delay(2000);                  //หน่วงรอเป็นเวลา 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  
}
```

การทดลองที่ 3 การควบคุมเซอร์โวมอเตอร์ด้วยจอยสติ๊ก

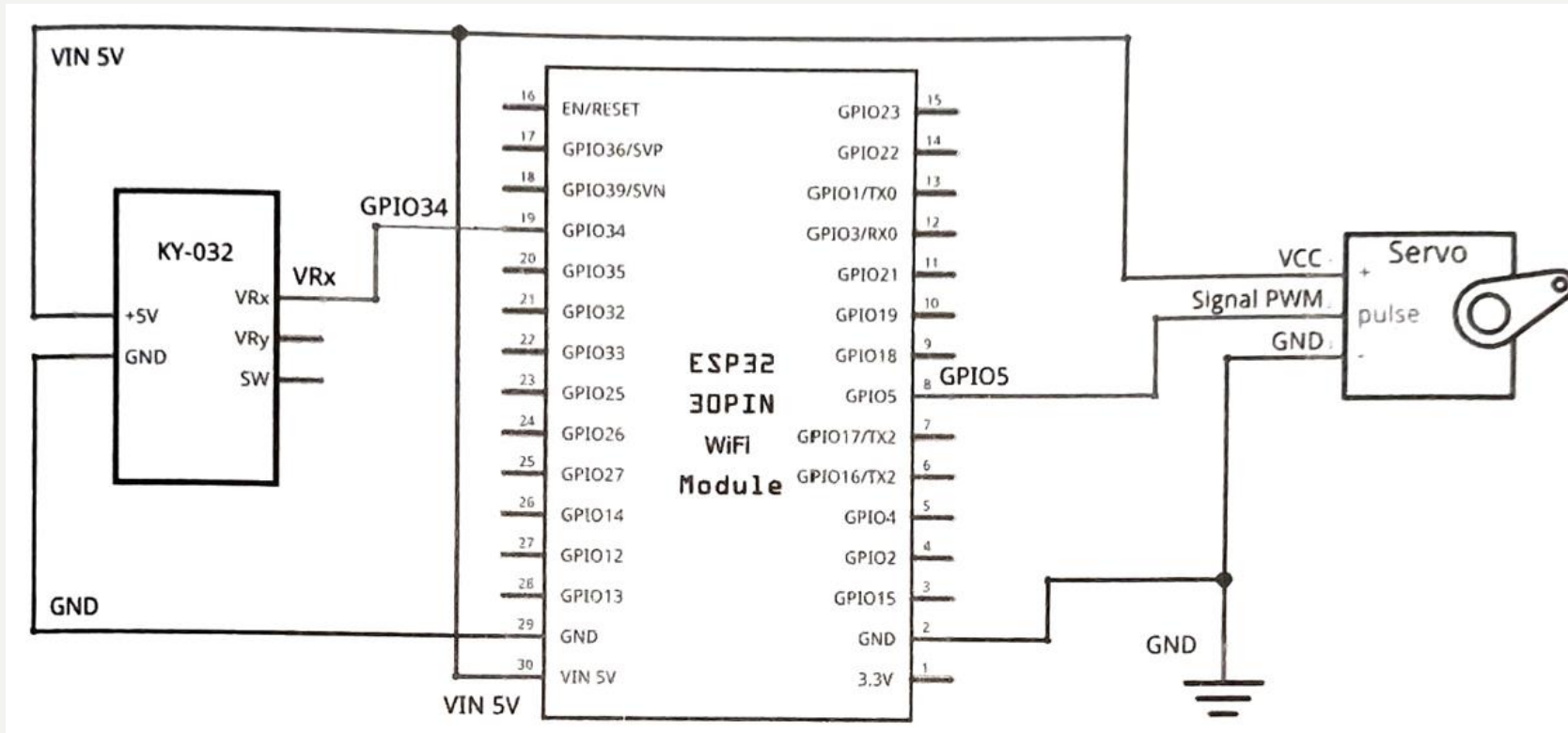




```
#include <Servo.h> //เรียกใช้ไลบรารี Servo.h เพื่อควบคุมเซอร์โวมอเตอร์
Servo myServo; //สร้างออบเจกต์จากคลาส Servo แล้วเก็บค่าลงใน myServo

void setup() {
  Serial.begin(115200);
  myServo.attach(5); //กำหนดให้ขา GPIO5 เป็นขาที่ใช้ควบคุมเซอร์โวมอเตอร์
}
```

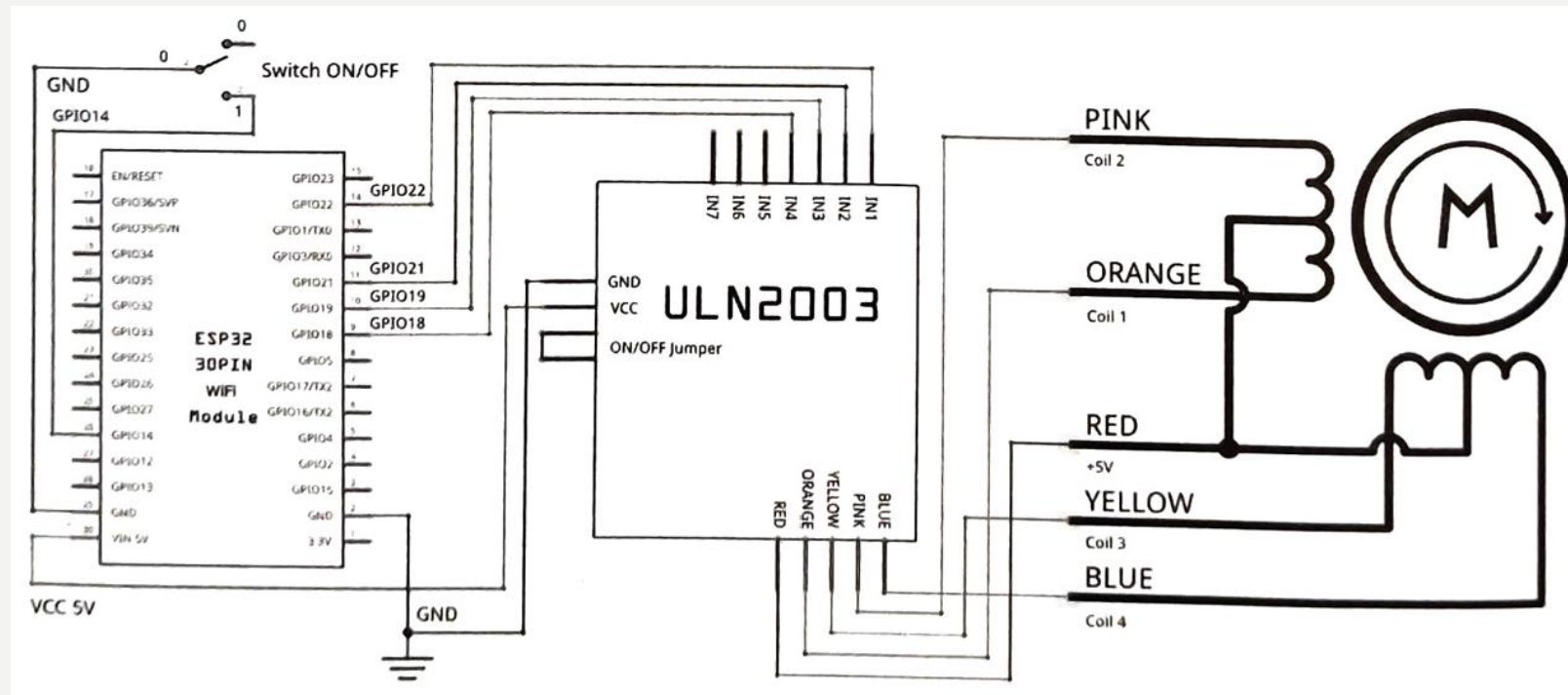
```
void loop() {  
  myServo.write(0);           //หมุนแกนเซอร์โวไปที่ตำแหน่ง 0 องศา  
  Serial.println("Servo : 0 Degree"); //แสดงข้อความใน " " ออกทาง Serial Monitor  
  delay(3000);                //หน่วงรอเป็นเวลา 3 วินาที  
  myServo.write(45);          //หมุนแกนเซอร์โวไปที่ตำแหน่ง 45 องศา  
  Serial.println("Servo: 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  
}
```

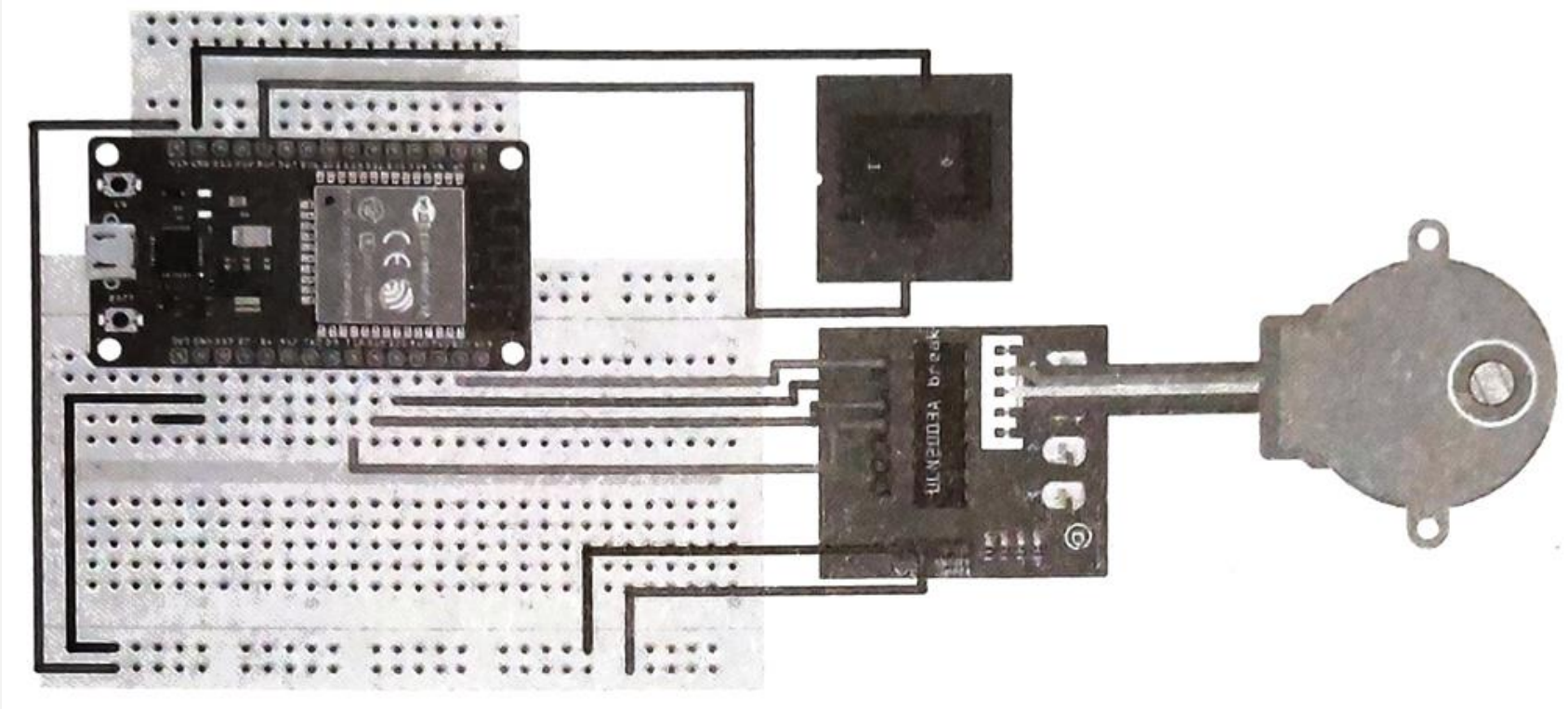



```
#include <Servo.h> //เรียกใช้ไลบรารี Servo.h เพื่อควบคุมเซอร์โวมอเตอร์
#define JOYSTICK_X 34 //ประกาศตัวแปร JOYSTICK_X ที่ต่ออยู่กับขา GPIO34
#define SERVO_PIN 5 //ประกาศตัวแปร SERVO_PIN ที่ต่ออยู่กับขา D5/GPIO5
Servo myServo; //สร้างออบเจกต์จากคลาส Servo แล้วเก็บค่าลงใน myServo
void setup() {
  pinMode(SERVO_PIN, OUTPUT); //กำหนดให้ขา D5 หรือ GPIO5 เป็น Output
  myServo.attach(SERVO_PIN); //กำหนดให้ขา GPIO5 เป็นขาที่ใช้ควบคุมเซอร์โวมอเตอร์
  myServo.write(90); //หมุนแกนเซอร์โวไปที่ตำแหน่งเริ่มต้นคือ 90 องศา
  //เป็นองศาเดียวกันกับคันโยกบนจอยสติ๊กที่จะต้องอยู่ตรงกึ่งกลาง)

  Serial.begin(115200);
  delay(200); //หน่วงรอเป็นเวลา 0.2 วินาที
}
```


การทดลองที่ 4 การควบคุมสแต็ปมอเตอร์ด้วยโมดูลขับ ULN2003





```
#include <Stepper.h> //เรียกใช้ไลบรารี 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); //สร้างออบเจกต์จาก
// 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){                    //ถ้ามีการเปิดสวิตช์ ค่าในตัวแปร switchStatus จะมีค่า  
                                                    เป็น 0 หรือ LOW (แต่ถ้าปิดสวิตช์จะเป็น 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 { //ถ้าปิดสวิตช์ ค่าในตัวแปร switchStatus จะมีค่าเป็น 1 หรือ HIGH
  Serial.println("counterclockwise 360 "); //แสดงข้อความใน " " ออก Serial Monitor
  myStepper.step(-stepsPerRevolution); //หมุนไปตามจำนวนสเต็ปต่อรอบการหมุนที่กำหนด
  //ในตัวแปรคือ 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);
}
}
```