ใบงานที่ 4

Pulse Width Modulation (PWM)

จุดประสงค์การเรียนรู้

1.ศึกษาการวัดแรงดัน Analog จาก PWM

2.ศึกษาการปรับเพิ่ม/ลดความสว่างของหลอดไฟ LED

3.ศึกษาการใช้ Serial Monitor ควบคุมความสว่างของหลอดไฟ LED

4.ศึกษาการปรับความสว่างของหลอดไฟ LED ด้วยตัวต้านทานปรับค่าได้

เครื่องมือและอุปกรณ์การทดลอง

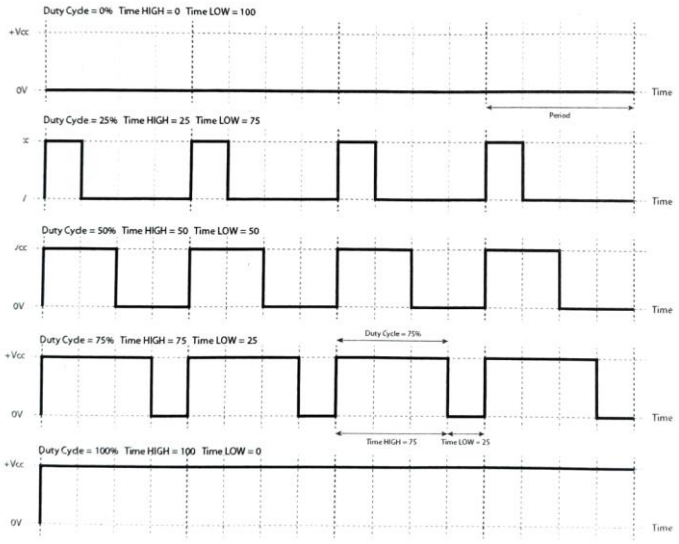
1.เครื่องไมโครคอมพิวเตอร์

2.บอรด์ NodeMCU ESP32

3.โปรแกรมการทดลอง

4.อุปกรณ์อิเล็กทอรนิกส์สำหรับทดลอง

PWM ย่อมาจาก Pulse Width Modulation เป็นการสร้างสัญญาณเอาต์พุตแบบดิจิตอลหรือสัญญาณ พัลส์ (Pulse) ที่เป็นรูปคลื่นแบบสี่เหลี่ยม (Square Wave) โดยสัญญาณพัลส์ที่เกิดขึ้นจะมีความต่อเนื่องและ มีความถี่หรือคาบเวลาที่คงที่ แต่จะมีช่วงความกว้างของรูปคลื่น (Pulse Width) ทั้งซีกบวก (สถานะเป็น 1 หรือ HIGH) และซีกลบ (สถานะเป็น 0 หรือ LOW) ที่แตกต่างกัน ซึ่งค่าที่แตกต่างกันใน 1 คาบเวลา (Period) นี้ จะนำมาคิดสัดส่วนเทียบเป็นเปอร์เซ็นต์ของสถานะ HIGH และ LOW ต่อสัญญาณพัลส์ใน 1 คาบเวลา โดยสัดส่วน เปอร์เซ็นต์ของสถานะ HIGH ในแต่ละคาบ จะถูกเรียกว่า Duty Cycle และเมื่อนำเอาค่าเปอร์เซ็นต์ต่างๆ เหล่านี้ มาใช้แทนค่าทางอนาล็อก ก็จะได้เป็นสัญญาณเอาต์พุตแบบอนาล็อกไปควบคุมอุปกรณ์ต่างๆ เช่น ความสว่าง ของหลอดไฟ, ความเร็วรอบของมอเตอร์ดีซี, การหมุนของเซอร์โว ฯลฯ ซึ่งเป็นการควบคุมในแง่ปริมาณที่ปกติ แล้วจะไม่สามารถสั่งการได้ด้วยสัญญาณดิจิตอลโดยตรง



การทดลองที่ 1 การวัดแรงดัน Analog จาก PWM

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

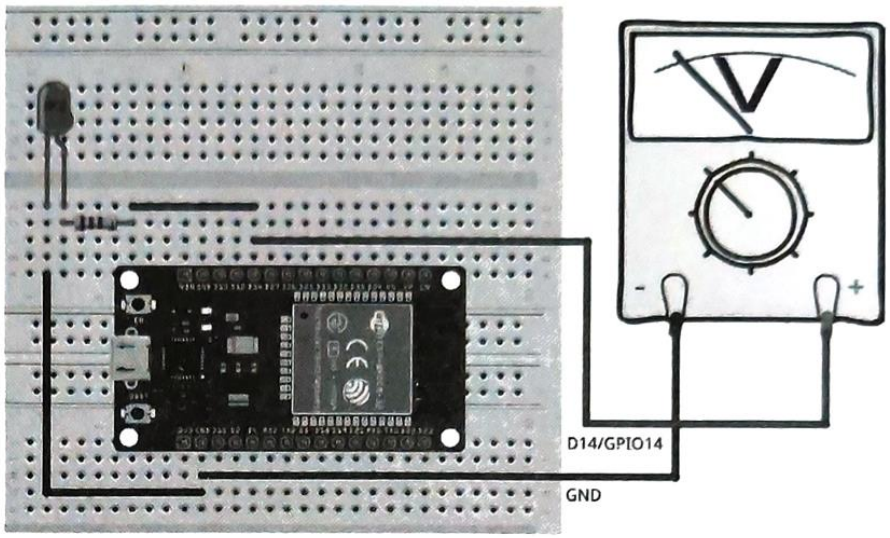
2. หลอดไฟ LED

3.ตัวต้านทาน (Resistor)

4.แผงต่อวงจร

5.สายไฟต่อวงจร

ประกอบวงจรตามรูป



ตัวอย่างนี้จริงๆ แล้วไม่ต้องต่อวงจรหรืออุปกรณ์อิเล็กทรอนิกส์ใดๆ เพิ่มเติมให้กับบอร์ดก็ได้ เพียงแค่หา ดิจิตอลมัลติมิเตอร์มาใช้วัดค่าแรงดัน Analog ที่ได้จาก PWM โดยในขั้นแรกให้เราเอาสายวัดสีแดงที่เป็นขั้วบวก (+) ของมัลติมิเตอร์ แตะที่ขาเอาต์พุตของสัญญาณ PWM ซึ่งในที่นี้ใช้เป็นขา D14 หรือ GPIO14 ส่วนสายวัด สีดำที่เป็นขั้วลบ (-) ให้แตะที่ขากราวด์ (G) จากนั้นปรับฟังก์ชั่นการวัดของมัลติมิเตอร์ไปที่ Voltage (V) เลือกให้ มีการปรับย่านวัดแบบ Auto แต่เพื่อให้เห็นภาพการเปลี่ยนแปลงค่าแรงดันเอาต์พุตที่ชัดเจนขึ้น ในที่นี้เราจะต่อ หลอดไฟ LED และตัวต้านทาน 2209 ไว้ที่ขาเอาต์พุตของสัญญาณ PWM ด้วย สุดท้ายเมื่อทุกอย่างเรียบร้อย ก็ลงมือเขียนโค้ดและอัพโหลดโปรแกรม โดย

```
#define LED 14                          //กำหนดให้ขา D14/GPIO14 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED
const int pinChannel = 0;               //ประกาศตัวแปรที่ใช้เก็บข้อมูลช่องสัญญาณ PWM
const int freq = 5000;                  //ประกาศตัวแปรที่ใช้เก็บข้อมูลความถี่สัญญาณ PWM
const int resolution = 10;              //ประกาศตัวแปรที่ใช้เก็บข้อมูลความละเอียดช่องสัญญาณ PWM

void setup() {
  Serial.begin(115200);                 //เริ่มการรับส่งข้อมูลด้วยความเร็ว Baud Rate 115200
  delay(100);

  ledcSetup(pinChannel, freq, resolution);    //ตั้งค่าเพื่อกำหนดรูปแบบของสัญญาณ PWM
  ledcAttachPin(LED, pinChannel);             //กำหนดขาและช่องสัญญาณที่ใช้ในการควบคุม
}

void loop() {
  //ค่าตัวเลขหรือ Duty Cycle ที่ใช้สร้างสัญญาณ PWM
  Serial.println("0% PWM");           //พิมพ์ 0% PWM ออกทางหน้าจอ Serial Monitor
  ledcWrite(pinChannel, 0);           //กำหนดให้แสดงค่าตัวเลขในแบบอนาล็อกที่ช่องสัญญาณ PWM
                                      เป็น 0 ซึ่งก็คือค่า Duty Cycle ที่ 0% วัดแรงดันได้ 0V
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
  Serial.println("25% PWM");          //พิมพ์ 25% PWM ออกทางหน้าจอ Serial Monitor
  ledcWrite(pinChannel, 255);         //กำหนดให้แสดงค่าตัวเลขในแบบอนาล็อกที่ช่องสัญญาณ PWM
                                      เป็น 255 ซึ่งก็คือค่า Duty Cycle ที่ 25% วัดแรงดันได้ 0.825V
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
  Serial.println("50% PWM");          //พิมพ์ 50% PWM ออกทางหน้าจอ Serial Monitor
  ledcWrite(pinChannel, 512);         //กำหนดให้แสดงค่าตัวเลขในแบบอนาล็อกที่ช่องสัญญาณ PWM
                                      เป็น 512 ซึ่งก็คือค่า Duty Cycle ที่ 50% วัดแรงดันได้ 1.65V
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
  Serial.println("75% PWM");          //พิมพ์ 75% PWM ออกทางหน้าจอ Serial Monitor
  ledcWrite(pinChannel, 767);         //กำหนดให้แสดงค่าตัวเลขในแบบอนาล็อกที่ช่องสัญญาณ PWM
                                      เป็น 767 ซึ่งก็คือค่า Duty Cycle ที่ 75% วัดแรงดันได้ 2.475V
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
  Serial.println("100% PWM");         //พิมพ์ 100% PWM ออกทางหน้าจอ Serial Monitor
  ledcWrite(pinChannel, 1023);        //กำหนดให้แสดงค่าตัวเลขในแบบอนาล็อกที่ช่องสัญญาณ PWM
                                      เป็น 1023 ซึ่งก็คือค่า Duty Cycle ที่ 100% วัดแรงดันได้ 3.3V
  delay(3000);                        //หน่วงรอเป็นเวลา 3 วินาที
}
```

บันทึกผลการทดลอง

.....

.....

.....

การทดลองที่ 2 การปรับเพิ่ม/ลดความสว่างของหลอดไฟ LED

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

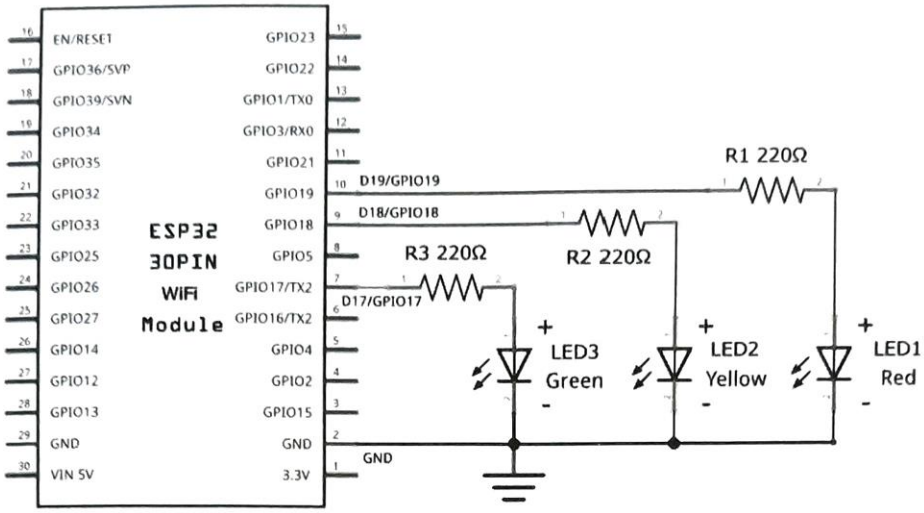
2. หลอดไฟ LED

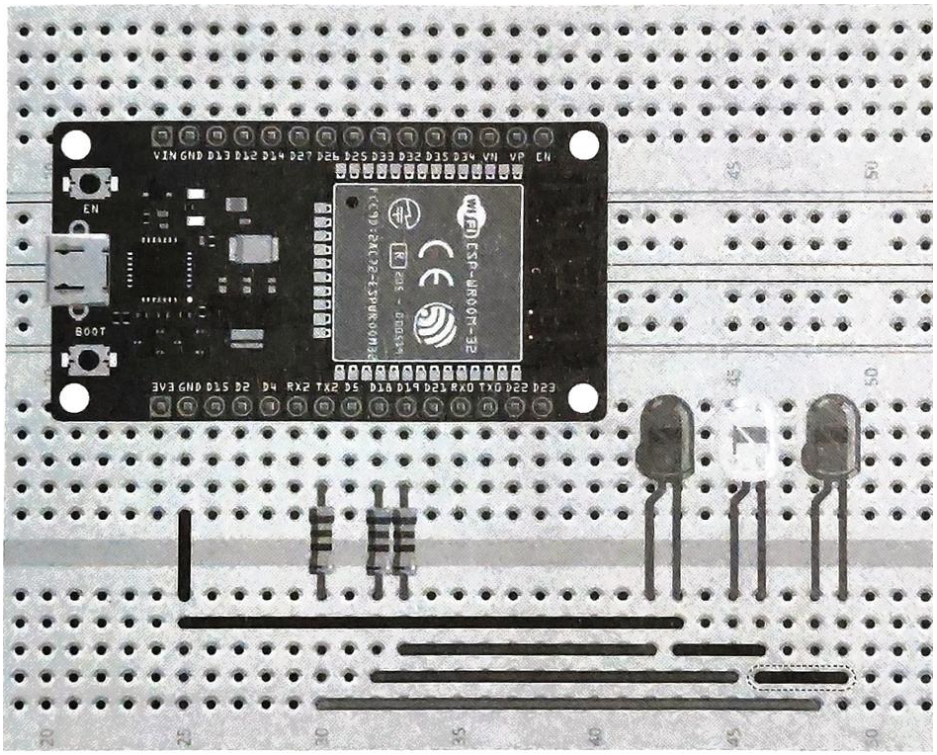
3.ตัวต้านทาน (Resistor)

4.แผงต่อวงจร

5.สายไฟต่อวงจร

ประกอบวงจรตามรูป





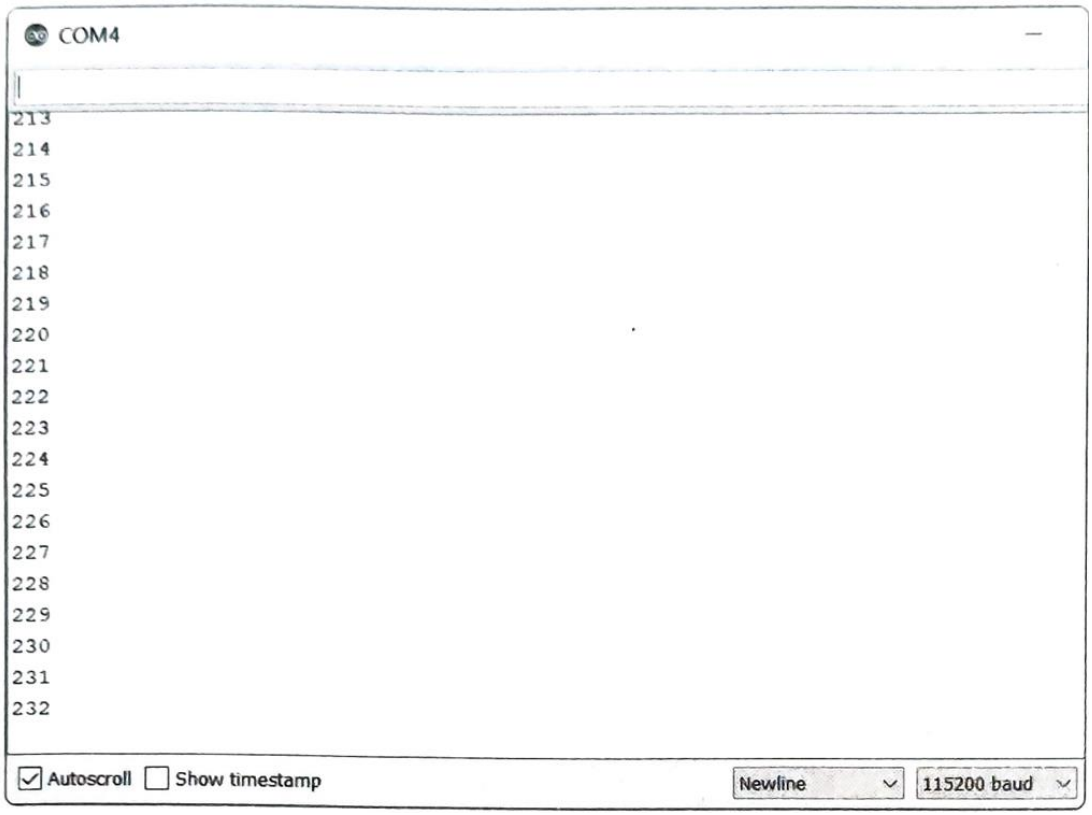
```
#define LED1 17                    //กำหนดให้ขา D17/GPIO17 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED1
#define LED2 18                    //กำหนดให้ขา D18/GPIO18 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED2
#define LED3 19                    //กำหนดให้ขา D19/GPIO19 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED3

const int pinChannel = 0;          //ประกาศตัวแปรที่ใช้เก็บข้อมูลช่องสัญญาณ PWM
const int freq = 1000;             //ประกาศตัวแปรที่ใช้เก็บข้อมูลความถี่สัญญาณ PWM
const int resolution = 8;          //ประกาศตัวแปรที่ใช้เก็บข้อมูลความละเอียดช่องสัญญาณ PWM

int value_PWM;                     //ประกาศตัวแปรเป็นตัวเลขจำนวนเต็ม

void setup()
{
  Serial.begin(115200);            //เริ่มการรับส่งข้อมูลด้วยความเร็ว Baud Rate 115200
  ledcSetup(pinChannel, freq, resolution);   //ตั้งค่าเพื่อกำหนดรูปแบบของสัญญาณ PWM
  ledcAttachPin(LED1, pinChannel);   //กำหนดขาที่ใช้ในการควบคุมสัญญาณเอาต์พุต
  ledcAttachPin(LED2, pinChannel);
  ledcAttachPin(LED3, pinChannel);
}

void loop()
{
  for(value_PWM=0; value_PWM<255; value_PWM++)   //กำหนดค่าเริ่มต้นของการนับ,
                                                 เงื่อนไข และนับเพิ่มครั้งละ
  {
    ledcWrite(pinChannel, value_PWM);            //แสดงค่าตัวเลขหรือค่า Duty Cycle
                                                 ที่ช่องสัญญาณ PWM
    Serial.println(value_PWM);       //พิมพ์ค่าตัวเลขหรือค่า Duty Cycle ใน Serial Monitor
    delay(10);                       //หน่วงเวลารอแต่ละค่าที่แสดง
  }
  for(value_PWM=255; value_PWM>0; value_PWM--)   //กำหนดค่าเริ่มต้นของการนับ,
                                                 เงื่อนไข และนับลดครั้งละ
  {
    ledcWrite(pinChannel, value_PWM);            //แสดงค่าตัวเลขหรือค่า Duty Cycle
                                                 ที่ช่องสัญญาณ PWM
    Serial.println(value_PWM);       //พิมพ์ค่าตัวเลขหรือค่า Duty Cycle ใน Serial Monitor
    delay(10);                       //หน่วงเวลารอแต่ละค่าที่แสดง
  }
}
```



บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 3 การใช้ Serial Monitor ควบคุมความสว่างของหลอดไฟ LED

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

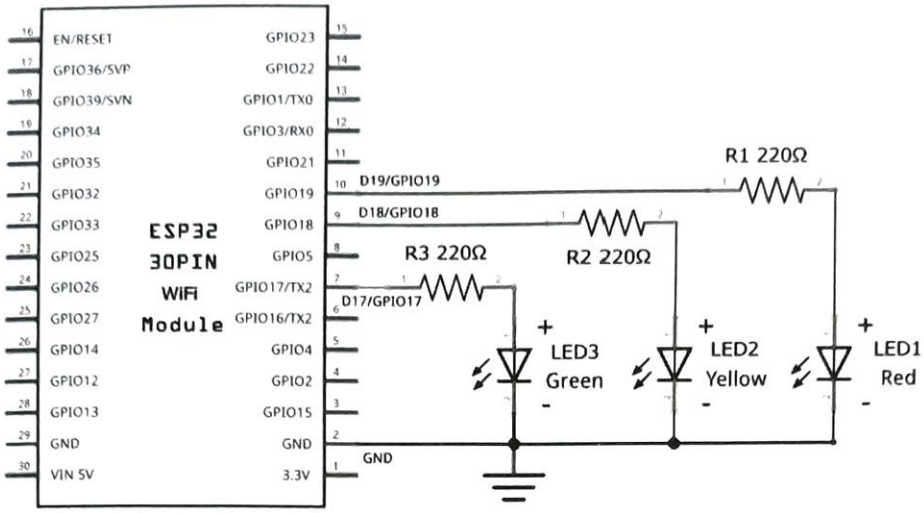
2. หลอดไฟ LED

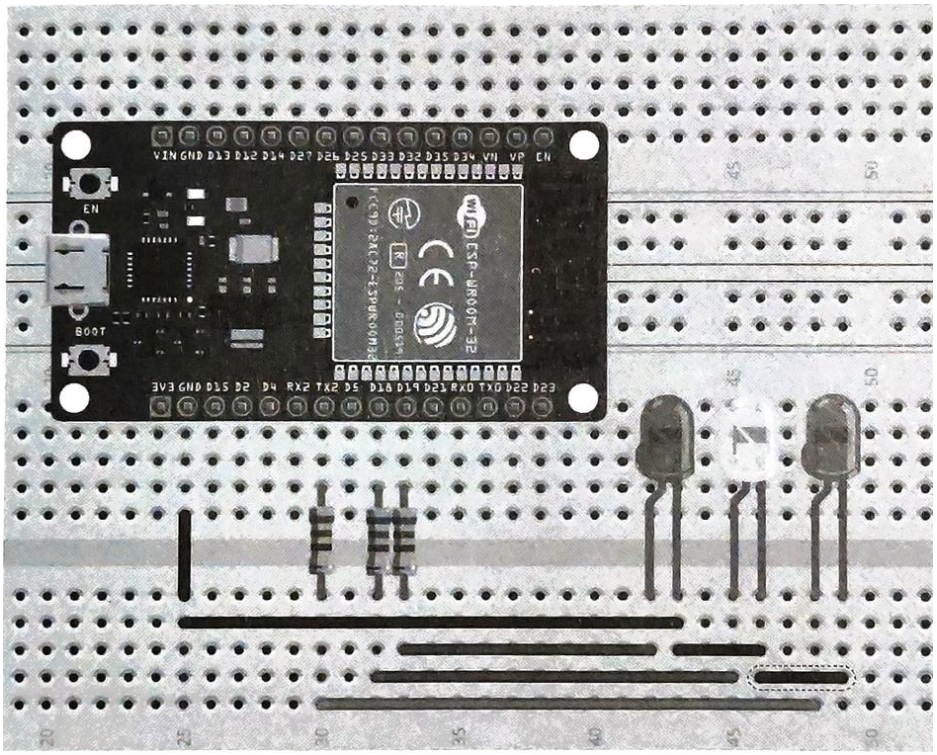
3.ตัวต้านทาน (Resistor)

4.แผงต่อวงจร

5.สายไฟต่อวงจร

ประกอบวงจรตามรูป





```
#define LED1 17                    //กำหนดให้ขา D17/GPIO17 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED1
#define LED2 18                    //กำหนดให้ขา D18/GPIO18 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED2
#define LED3 19                    //กำหนดให้ขา D19/GPIO19 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED3

const int pinChannel = 0;          //ประกาศตัวแปรที่ใช้เก็บข้อมูลช่องสัญญาณ PWM
const int freq = 1000;             //ประกาศตัวแปรที่ใช้เก็บข้อมูลความถี่สัญญาณ PWM
const int resolution = 8;          //ประกาศตัวแปรที่ใช้เก็บข้อมูลความละเอียดช่องสัญญาณ PWM

int value_PWM;                     //ประกาศตัวแปรเป็นตัวเลขจำนวนเต็ม
void setup()
{
  Serial.begin(115200);            //เริ่มการรับส่งข้อมูลด้วยความเร็ว Baud Rate 115200
  delay(100);
  Serial.println("Please enter numbers since 0 - 255");   //พิมพ์ข้อความใน " "
                                                          ออกทาง Serial Monitor
  ledcSetup(pinChannel, freq, resolution);  //ตั้งค่าเพื่อกำหนดรูปแบบของสัญญาณ PWM
  ledcAttachPin(LED1, pinChannel);          //กำหนดขาที่ใช้ในการควบคุมสัญญาณเอาต์พุต
  ledcAttachPin(LED2, pinChannel);
  ledcAttachPin(LED3, pinChannel);
}

void loop()
{
  if(Serial.available()>0)         //ตรวจสอบว่ามีข้อมูลส่งมาจาก Serial Monitor หรือไม่
  {
    value_PWM = Serial.parseInt();          //อ่านข้อมูลเลขจำนวนเต็มจาก Serial Monitor
                                            เก็บไว้ในตัวแปร
    ledcWrite(pinChannel, value_PWM);       //แสดงค่าตัวเลขหรือค่า Duty Cycle
                                            ที่ช่องสัญญาณ PWM
    Serial.println(value_PWM);   //พิมพ์ค่าตัวเลขหรือค่า Duty Cycle ใน Serial Monitor
    delay(3000);                 //หน่วงเวลารอ 3 วินาที
  }
}
```

บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 4 การปรับความสว่างของหลอดไฟ LED ด้วยตัวต้านทานปรับค่าได้

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

2. หลอดไฟ LED

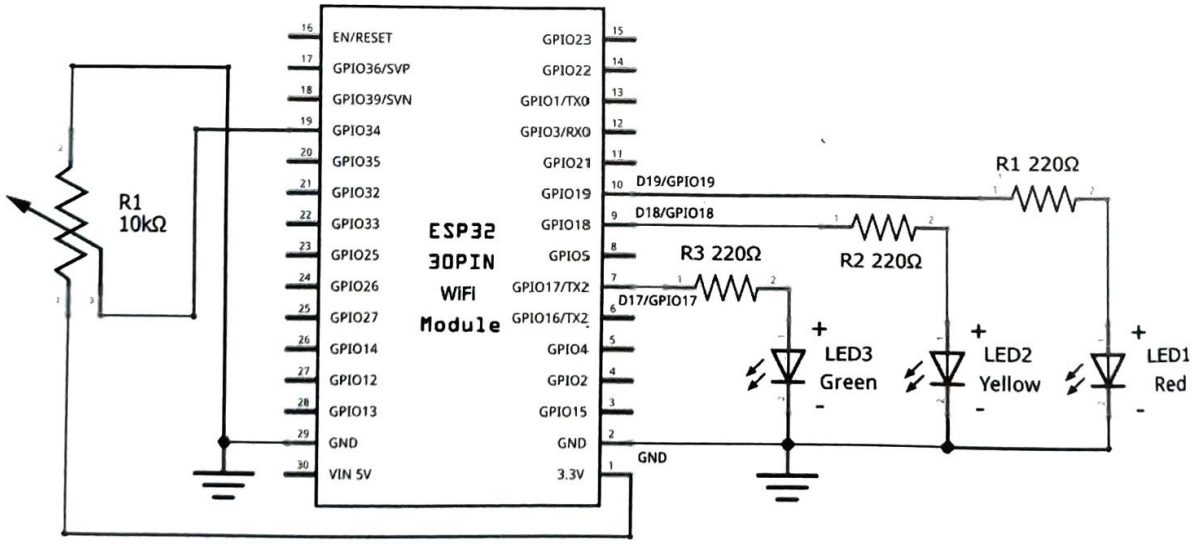
3.ตัวต้านทาน (Resistor)

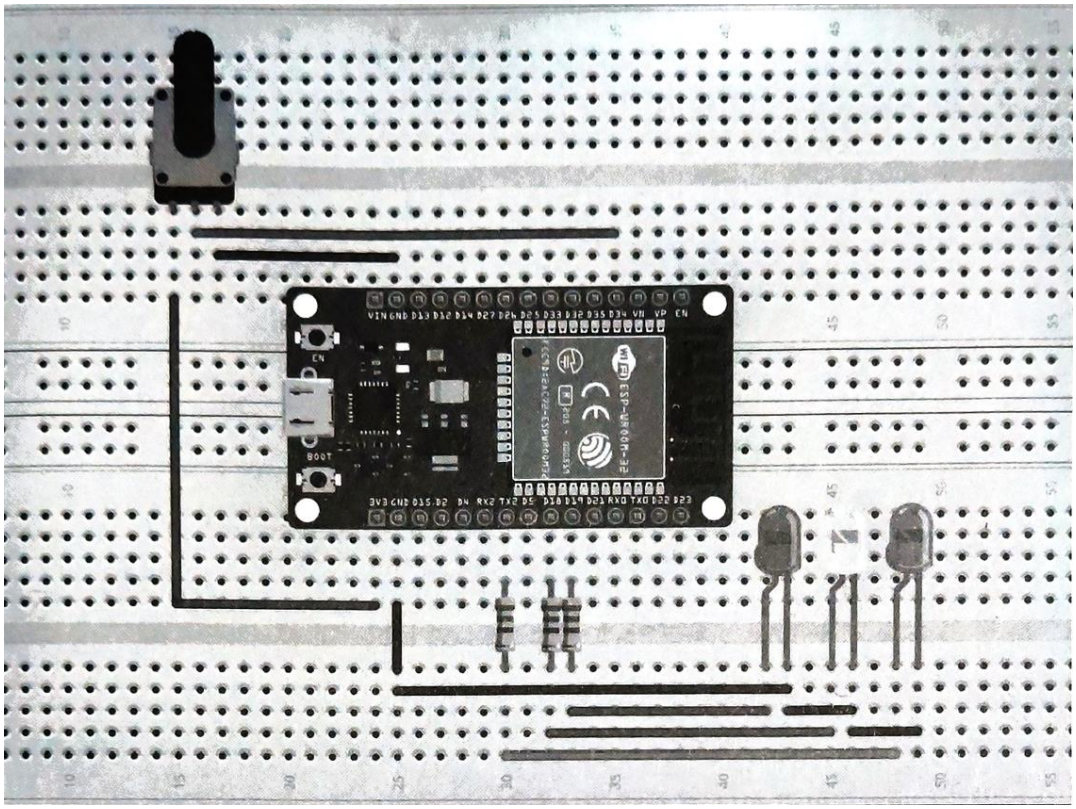
4.ตัวต้านทานปรับค่าได้ (POT)

5.แผงต่อวงจร

6.สายไฟต่อวงจร

ประกอบวงจรตามรูป





```
#define LED1 17                    //กำหนดให้ขา D17/GPIO17 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED1
#define LED2 18                    //กำหนดให้ขา D18/GPIO18 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED2
#define LED3 19                    //กำหนดให้ขา D19/GPIO19 เป็นเอาต์พุตที่ต่อกับหลอดไฟ LED3

const int analoglnPin = 34;        //ประกาศตัวแปรขาอินพุต Analog
int sensorValue = 0;               //ประกาศตัวแปรที่ใช้เก็บค่าตัวเลขอนาล็อก ค่าเริ่มต้นเป็น 0

const int pinChannel = 0;          //ประกาศตัวแปรที่ใช้เก็บข้อมูลช่องสัญญาณ PWM
const int freq = 3000;             //ประกาศตัวแปรที่ใช้เก็บข้อมูลความถี่สัญญาณ PWM
const int resolution = 12;         //ประกาศตัวแปรที่ใช้เก็บข้อมูลความละเอียดช่องสัญญาณ PWM
void setup() {
  Serial.begin(115200);
  ledcSetup(pinChannel, freq, resolution);   //ตั้งค่าเพื่อกำหนดรูปแบบของสัญญาณ PWM
  ledcAttachPin(LED1, pinChannel);           //กำหนดขาที่ใช้ในการควบคุมสัญญาณเอาต์พุต
  ledcAttachPin(LED2, pinChannel);
  ledcAttachPin(LED3, pinChannel);
}

void loop() {
  sensorValue = analogRead(analoglnPin);     //อ่านค่าอินพุต Analog จากขา 34
                                             ไปเก็บไว้ที่ตัวแปร
  ledcWrite(pinChannel, sensorValue);        //แสดงค่าตัวเลขหรือค่า Duty Cycle
                                             ที่ช่องสัญญาณ PWM
  Serial.println(sensorValue);               //แสดงค่าในตัวแปรออกทาง Serial Monitor
  delay(100);
}
```

บันทึกผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....