ใบงานที่ 3

การเชื่อมต่อ ADC และ DAC

จุดประสงค์การเรียนรู้

1.ศึกษาการวัดสัญญาณ Analog

2.ศึกษาการปรับความถี่ในการกะพริบของหลอดไฟ LED

3.ศึกษาการปรับแต่งค่าช่วงของเอาต์พุตด้วยฟังก์ชั่น map

4.ศึกษาการวัดค่าแรงดันเอาต์พุต

5.ศึกษาการปรับเพิ่ม/ลดความสว่างของหลอดไฟ LED

เครื่องมือและอุปกรณ์การทดลอง

1.เครื่องไมโครคอมพิวเตอร์

2.บอรด์ NodeMCU ESP32

3.โปรแกรมการทดลอง

4.อุปกรณ์อิเล็กทอรนิกส์สำหรับทดลอง

การทดลองที่ 1 การวัดสัญญาณ Analog

ตัวอย่างนี้จะเป็นการทดลองวัดสัญญาณ Analog ที่ได้รับจากการหมุนปรับตัวต้านทาน ปรับค่าได้ (Potentiometer) โดยตัวมันสามารถที่จะป้อนระดับแรงดันอินพุตหรือสัญญาณ อนาล็อก Analog ได้ตั้งแต่ OV ไปจนถึงระดับ VCC ซึ่งคล้ายคลึงกับการทำงานของเซ็นเซอร์ Analog ทั่วๆ ไป โดยเมื่อเราหมุนปรับแกนกลางของตัวต้านทานไปทางขาด้านใด ความ ต้านทานระหว่างขาด้านนั้นกับขากลางจะค่อยๆ ลดน้อยลง จนเหลือ 09 ส่งผล ให้ระดับแรงดันอินพุตที่ขา ADC ของบอร์ดจะค่อยๆ ลดน้อยลงจนเหลือ OV หรือก็คือจะได้ค่าตัวเลขในแบบอนาล็อกออกมาเป็น 0 นั่นเอง ในทางกลับกัน ถ้าหมุนปรับแกนกลางไปทางขาอีกด้าน ความต้านทานจะค่อยๆ เพิ่มสูงขึ้น จนทำให้ระดับแรงดันอินพุตที่ขา ADC สูงขึ้นจนถึงระดับ VCC หรือก็คือจะได้ค่าตัวเลขในแบบอนาล็อกออกมาเป็น 4095 นั่นเอง

อุปกรณ์ในการทดลอง

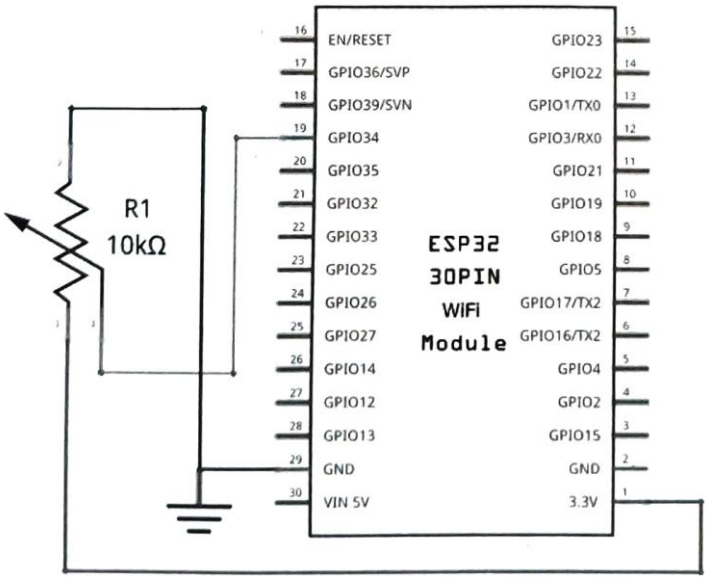
1.บอรด์ทดลอง ESP32

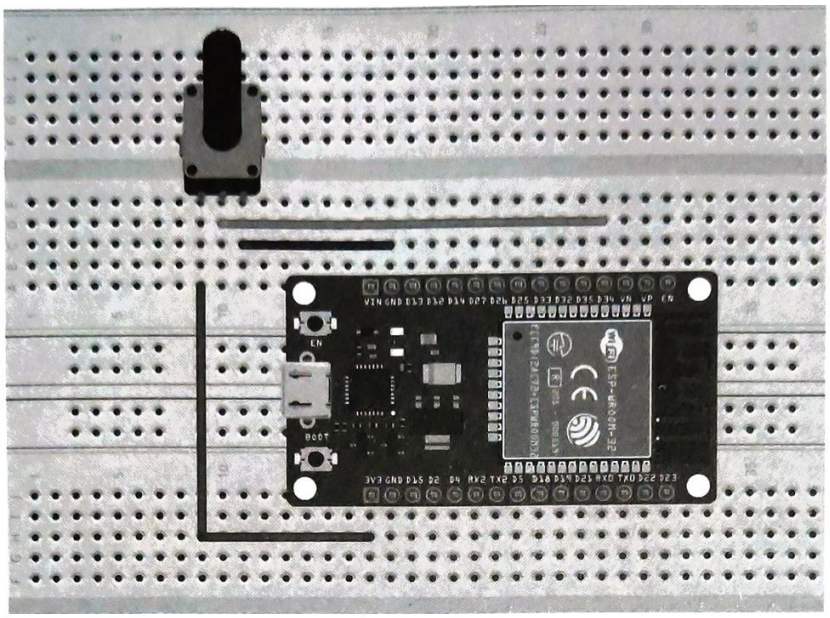
2.ตัวต้านทานปรับค่าได้ (POT)

3.แผงต่อวงจร

4.สายไฟต่อวงจร

ประกอบวงจรตามรูป



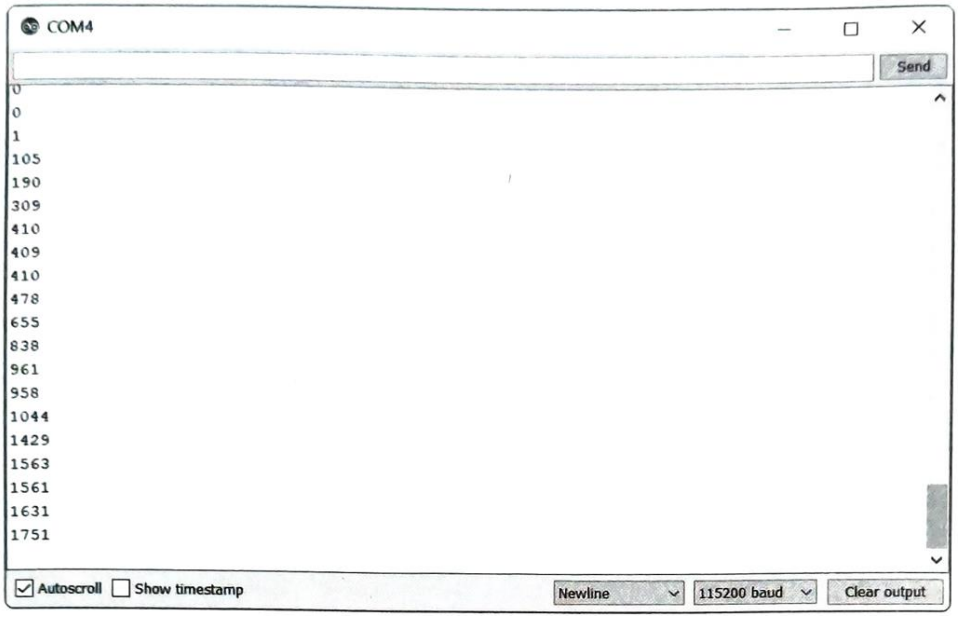


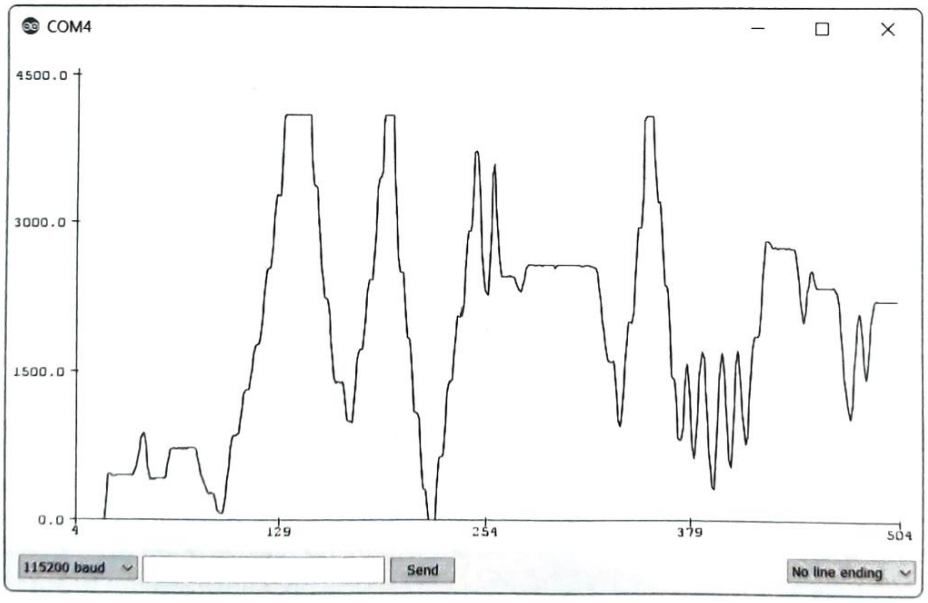
```
const int analoglnPin = 34;       //ประกาศตัวแปรขาอินพุต Analog
int sensorValue = 0;              //ประกาศตัวแปรที่ใช้เก็บค่าตัวเลขอนาล็อก ค่าเริ่มต้นเป็น 0

void setup() {
  Serial.begin(115200);
}

void loop() {
  sensorValue = analogRead(analoglnPin); //อ่านค่าอินพุต Analog จากขา 34 ไปเก็บไว้ที่ตัวแปร
  Serial.println(sensorValue);           //แสดงค่าในตัวแปรออกทาง Serial Monitor
  delay(100);
}
```

เปิดหน้าต่าง Serial Monitor แล้วเปิดหน้าต่าง Serial Plotter ขึ้นมาดู เมื่อทดลองปรับแกนหมุนของตัวต้านทานปรับค่าได้ จะเห็นว่าการพล็อตกราฟก็ให้ผลลัพธ์ที่เป็นไปในแนวทางเดียวกัน ดังรูป





บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 2 การปรับความถี่ในการกะพริบของหลอดไฟ LED

อุปกรณ์ในการทดลอง

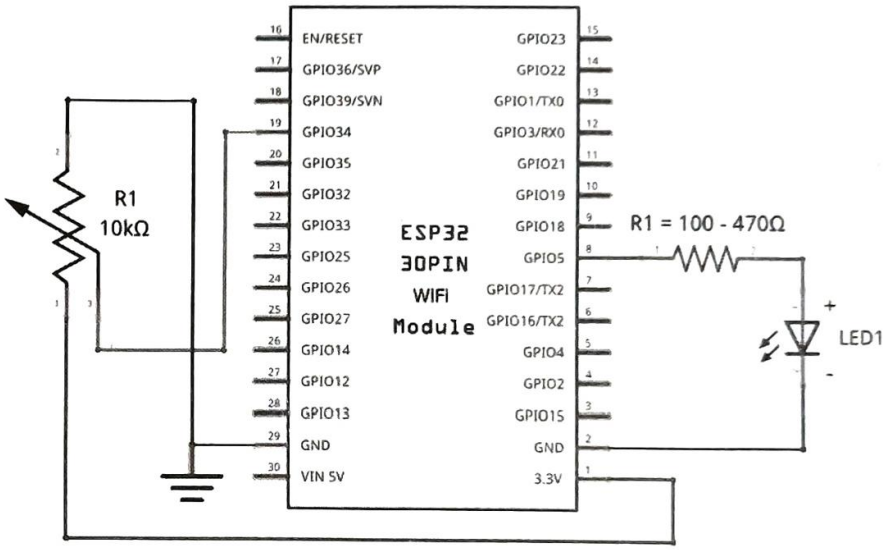
1.บอรด์ NodeMCU ESP32

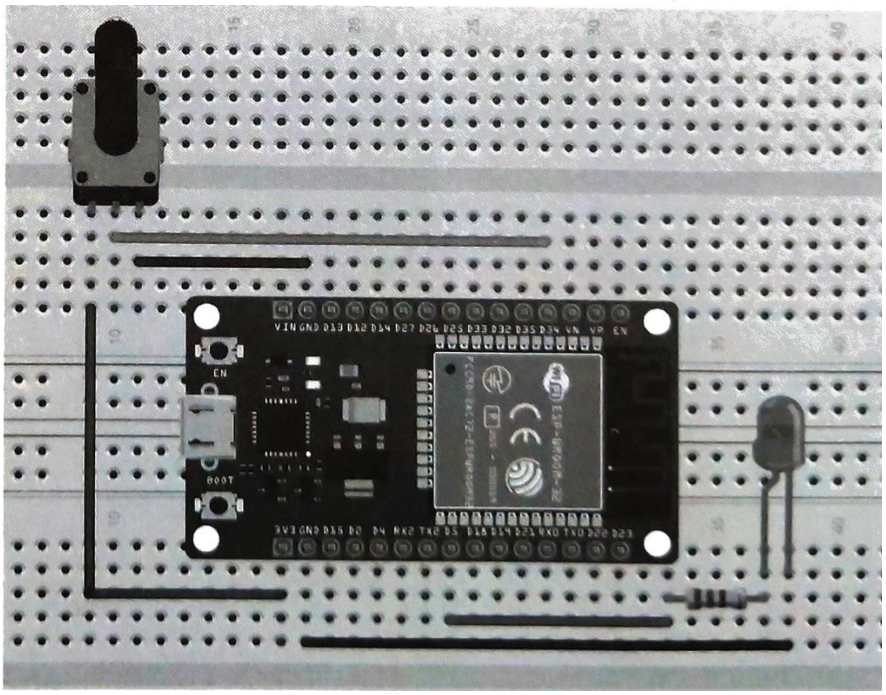
2.ตัวต้านทานปรับค่าได้ (POT)

3.แผงต่อวงจร

4.สายไฟต่อวงจร

ประกอบวงจรตามรูป





เมื่อต่อวงจรเสร็จแล้วก็ลงมือเขียนโค้ดและอัพโหลดโปรแกรม โดยเราจะเอาโค้ดจากตัวอย่างก่อนหน้านี้ มาแก้ไข ซึ่งมีรายละเอียดดังนี้

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

void setup() {
  Serial.begin(115200);
  pinMode(5, OUTPUT);                //กำหนดให้ขา D5/GPIO5 เป็น Output
}

void loop() {
  sensorValue = analogRead(analoglnPin);   //อ่านค่าอินพุต Analog จากขา 34 ไปเก็บไว้ที่ตัวแปร
  Serial.println(sensorValue);             //แสดงค่าในตัวแปรออกทาง Serial Monitor

  digitalWrite(5, HIGH);             //กำหนดให้ Output ที่ขา D5/GPIO5 เป็น HIGH หลอดไฟติด
  delay(sensorValue);                //นำค่าอินพุต Analog จากขา 34 มาเป็นค่าหน่วงเวลา
  digitalWrite(5, LOW);              //กำหนดให้ Output ที่ขา D5/GPIO5 เป็น LOW หลอดไฟดับ
  delay(sensorValue);
}
```

บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 3 การปรับแต่งค่าช่วงของเอาต์พุตด้วยฟังก์ชั่น map

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

2.ตัวต้านทานปรับค่าได้ (POT)

3.แผงต่อวงจร

4.สายไฟต่อวงจร

จากตัวอย่างก่อนหน้านี้ที่เมื่อปรับแกนหมุนของตัวต้านทานปรับค่าได้ เพื่อส่งสัญญาณอนาล็อก (Analog) ที่อยู่ในรูปของระดับแรงดันอินพุตไปที่ขา ADC ของบอร์ด ก่อนจะถูกแปลงสภาพด้วยการเปรียบเทียบค่า แรงดันอ้างอิง แล้วส่งเป็นสัญญาณดิจิตอลเอาต์พุตหรือเป็นค่าตัวเลขแบบอนาล็อกในช่วงตั้งแต่ 0-4095 ออกมา ถ้าหากเราต้องการที่จะเปลี่ยนแปลงหรือกำหนดค่าตัวเลขในช่วงดังกล่าวเอง เช่น ถ้าต้องการกำหนดให้เป็นค่า ตัวเลขในช่วง 100-2500 ก็สามารถทำได้ โดยใช้ฟังก์ชั่นหรือคำสั่ง เพื่อช่วยในการปรับแต่งช่วงค่าของ เอาต์พุตให้อยู่ในช่วงที่ต้องการ ซึ่งรูปแบบและรายละเอียดของคำสั่ง map คือ

map(value, from Low, fromHigh, toLow, toHigh);

•value : ค่าตัวเลขในช่วงเดิม หรือค่าที่ต้องการแปลง

• fromLow : ค่าตัวเลขต่ำสุดของช่วงเดิม (ก่อนแปลง)

• fromHigh : ค่าตัวเลขสูงสุดของช่วงเดิม (ก่อนแปลง)

• toLow : ค่าตัวเลขต่ำสุดของช่วงใหม่ (หลังแปลง)

• toHigh : ค่าตัวเลขสูงสุดของช่วงใหม่ (หลังแปลง)

ผลลัพธ์หลังจากการใช้ฟังก์ชั่น map) เราก็จะได้ค่าของตัวเลขเอาต์พุตที่อยู่ในช่วงใหม่ตามที่เราต้องการ เช่น จากตัวอย่างก่อนหน้านี้ถ้าเราต้องการเปลี่ยนค่าของตัวเลขเอาต์พุตในช่วง 0-4095 ให้เป็น 100-2500 ก็สามารถกำหนดรายละเอียดลงในฟังก์ชั่น map) ได้ดังนี้

newValue = map(sensorValue, 0, 4095, 100, 2500) ;

ใช้วงจรเดิมและโค้ดเดิมจากตัวอย่างที่แล้วหรือ การทดลองที่ 2 โดยนำโค้ดมาแก้ไขแล้วอัพโหลดโปรแกรม ซึ่งมีรายละเอียดดังนี้

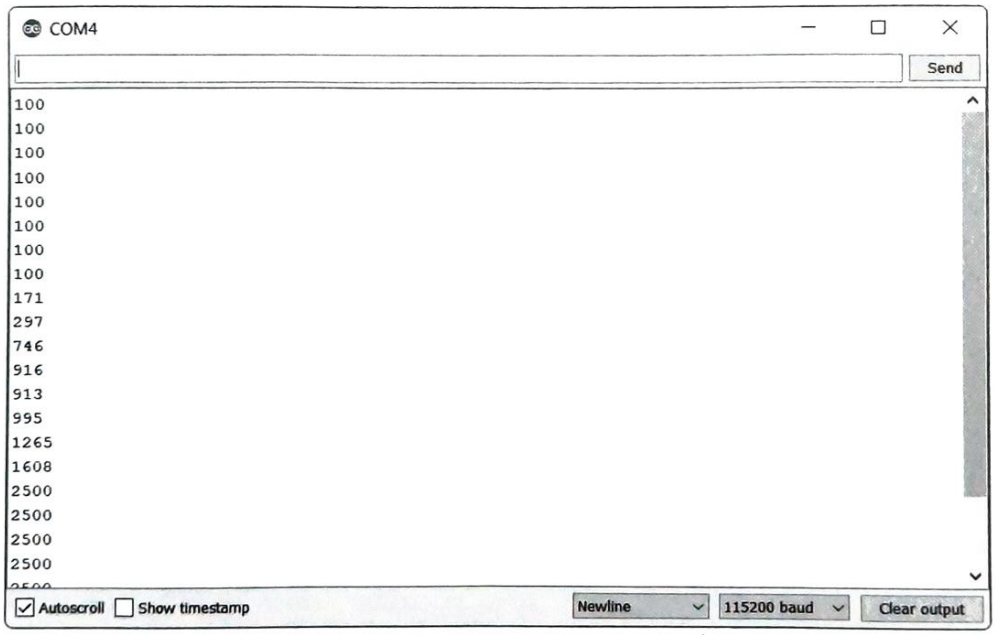
```
const int analogInPin = 34;          //ประกาศตัวแปรขาอินพุต Analog
int sensorValue = 0;                 //ประกาศตัวแปรที่ใช้เก็บค่าตัวเลขอนาล็อก ค่าเริ่มต้นเป็น 0
int newValue = 0;                    //ประกาศตัวแปรที่ใช้เก็บค่าตัวเลขอนาล็อกในช่วงใหม่
                                     โดยให้ค่าเริ่มต้นเป็น 0

void setup() {
  Serial.begin(115200);
  pinMode(5, OUTPUT);                //กำหนดให้ขา D5/GPIO5 เป็น Output
```

```
}

void loop() {
  sensorValue = analogRead(analoglnPin);  //อ่านค่าอินพุต Analog จากขา 34 ไปเก็บไว้ที่ตัวแปร
  newValue = map(sensorValue, 0, 4095, 100, 2500);  //ปรับให้เป็นค่าตัวเลขเอาต์พุต
                                                    ที่อยู่ในช่วงใหม่
  Serial.println(newValue);          //แสดงค่าในตัวแปรออกทาง Serial Monitor

  digitalWrite(5, HIGH);             //กำหนดให้ Output ที่ขา D5/GPIO5 เป็น HIGH หลอดไฟติด
  delay(newValue);                   //นำค่าตัวเลขเอาต์พุตที่อยู่ในช่วงใหม่มาเป็นค่าหน่วงเวลา
  digitalWrite(5, LOW);              //กำหนดให้ Output ที่ขา D5/GPIO5 เป็น LOW หลอดไฟดับ
  delay(newValue);
}
```



บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 4 การวัดค่าแรงดันเอาต์พุต

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

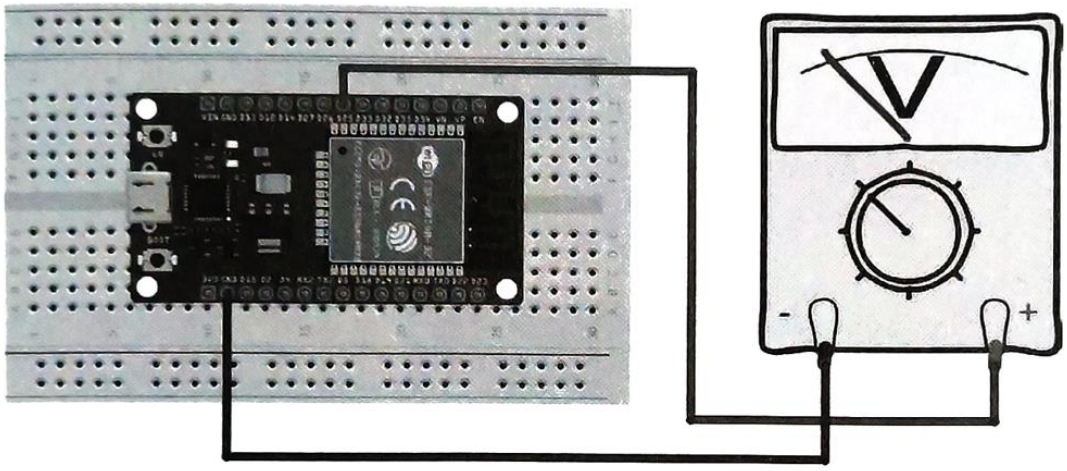
2.ดี.ซี.โวล์ทมิเตอร์

3.แผงต่อวงจร

4.สายไฟต่อวงจร

ตัวอย่างนี้จะเป็นการทดลองวัดค่าแรงดัน ที่ได้จากการแปลงข้อมูลที่เป็นสัญญาณดิจิตอลในรูปของตัวเลข ที่มีค่าตั้งแต่ 0-255 ให้เป็นค่าแรงดันในช่วงตั้งแต่ 0-3.3V ผ่านทางขา DAC โดยในที่นี้เราจะทดลองป้อนข้อมูล ที่เป็นค่าตัวเลข 128 ซึ่งจากการคำนวณเมื่อแปลงไปเป็นข้อมูลหรือสัญญาณ Analog ในรูปของค่าแรงดันผ่าน ทางขา DAC ค่าแรงดันที่ได้จะเป็น 1.65V ซึ่งในที่นี้เราจะใช้มัลติมิเตอร์เป็นตัววัดแรงดันที่ขา DAC (ในที่นี้เลือก ใช้ขา D25/GPIO25) เทียบกับ GND

ประกอบวงจรตามรูป

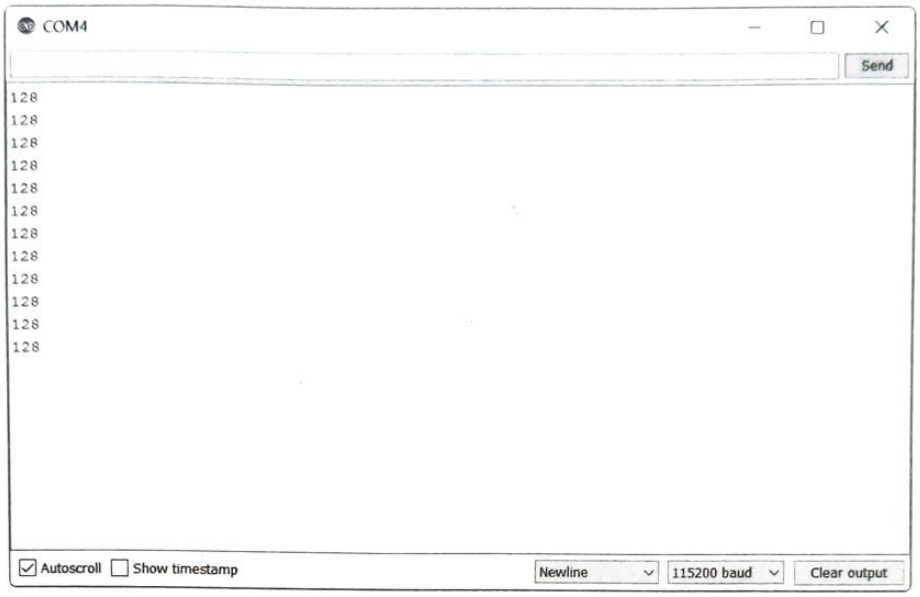


ตัวอย่างนี้ไม่ต้องต่อวงจรหรืออุปกรณ์อิเล็กทรอนิกส์ใดๆ เพิ่มเติมให้กับบอร์ด เพียงแค่หาดิจิตอลมัลติมิเตอร์ มาใช้วัดค่าแรงดัน Analog ที่ได้จากขา DAC โดยในขั้นแรกให้เราเอาสายวัดสีแดงที่เป็นขั้วบวก (+) ของมัลติมิเตอร์ แตะที่ขาเอาต์พุตของสัญญาณ DAC ซึ่งในที่นี้ใช้เป็นขา D25 หรือ GPIO25 ส่วนสายวัดสีดำที่เป็นขั้วลบ (-) ให้แตะ ที่ขากราวด์ (G) จากนั้นปรับฟังก์ชั่นการวัดของมัลติมิเตอร์ไปที่ Voltage (V) เลือกให้มีการปรับย่านวัดแบบ Auto สุดท้ายก็ลงมือเขียนโค้ดและอัพโหลดโปรแกรม ดังนี้

```
#define DAC1 25                    //กำหนดให้ขา D25/DGIO25 เป็นขา DAC1
int Value = 128;                   //ประกาศตัวแปรที่ใช้เก็บข้อมูลตัวเลขที่เป็นสัญญาณดิจิตอล

void setup() {
  Serial.begin(115200);
}

void loop() {
  dacWrite(DAC1, Value);           //กำหนดให้ขา DAC1 มีค่าเท่ากับข้อมูลที่เก็บไว้ในตัวแปร Value
  Serial.println(Value);           //แสดงค่าในตัวแปรออกทาง Serial Monitor
  delay(1000);
}
```



บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 5 การปรับเพิ่ม/ลดความสว่างของหลอดไฟ LED

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

2.หลอดไฟ LED

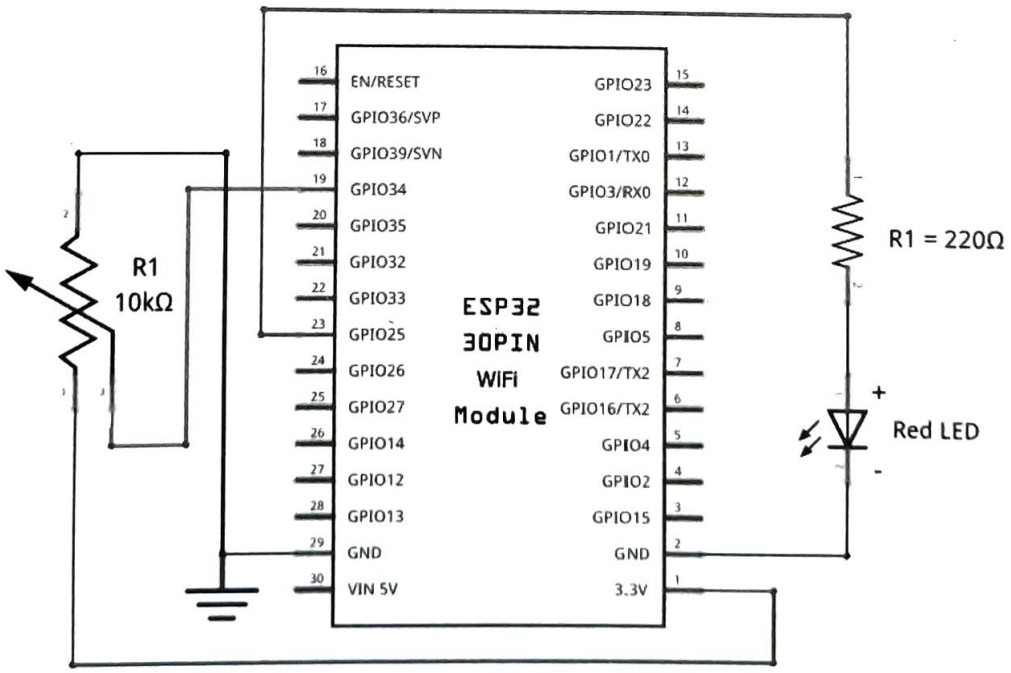
3.ตัวต้านทาน (Resistor)

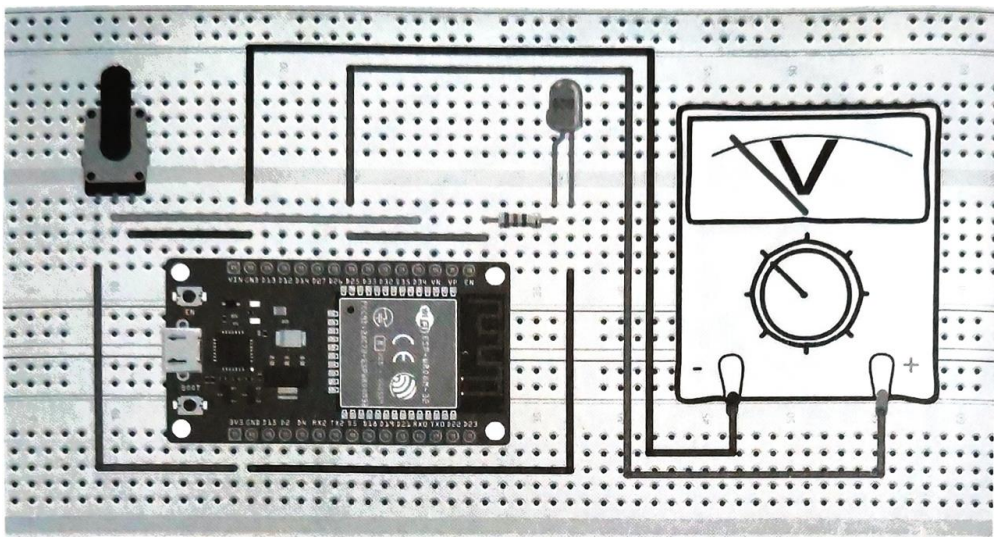
4.ตัวต้านทานปรับค่าได้ (POT)

5.แผงต่อวงจร

6.สายไฟต่อวงจร

ประกอบวงจรตามรูป





ลงมือต่อวงจรตามรูป จากตัวอย่าง การทดลองที่ 1 ที่มีตัวต้านทานปรับค่าได้ (Potentiometer) เชื่อมต่ออยู่ กับขา ADC ของบอร์ด (ในที่นี้คือขา D34/GPIO34 หรือ ADC1_6) อยู่แล้ว เพียงแค่เพิ่มหลอดไฟ LED สีแดง ต่อขาบวก (+) เข้ากับตัวต้านทาน 2209 และเชื่อมต่อไปยังขา DAC ของบอร์ด ซึ่งในที่นี้คือขา D25/GPIO25 หรือ DAC1 ส่วนขาลบ (-) ของหลอดไฟ LED ให้ต่อเข้ากับขา GND จากนั้นเพื่อวัดค่าแรงดันไฟที่ขา DAC หรือ ที่หลอดไฟ LED ได้รับ ให้นำเอาสายวัดสีแดงที่เป็นขั้วบวก (+) ของมัลติมิเตอร์ แตะที่ขา D25/GPIO25 ส่วน สายวัดสีดำที่เป็นขั้วลบ (-) ให้แตะที่ขากราวด์ (G) จากนั้นปรับฟังก์ชั่นการวัดของมัลติมิเตอร์ไปที่ Voltage (V) เลือกให้มีการปรับย่านวัดแบบ Auto สุดท้ายก็ลงมือเขียนโค้ดและอัพโหลดโปรแกรม โดยเราจะเอาโค้ดจาก ตัวอย่าง การทดลองที่ 1 และ 4 มาแก้ไข ซึ่งมีรายละเอียดดังนี้

```
const int analoglnPin = 34;      //ประกาศตัวแปรขาอินพุต Analog
int sensorValue = 0;             //ประกาศตัวแปรที่ใช้เก็บค่าตัวเลขอนาล็อก ค่าเริ่มต้นเป็น 0
int newValue = 0;                //ประกาศตัวแปรที่ใช้เก็บค่าตัวเลขอนาล็อกในช่วงใหม่
                                 โดยให้ค่าเริ่มต้นเป็น 0
#define DAC1 25                  //กำหนดให้ขา D25/DGIO25 เป็นขา DAC1

void setup() {
  Serial.begin(115200);
}

void loop() {
  sensorValue = analogRead(analoglnPin);  //อ่านค่าอินพุต Analog จากขา 34 ไปเก็บไว้ที่ตัวแปร
  newValue = map(sensorValue, 0, 4095, 116 ,255);    //ปรับให้เป็นค่าตัวเลขเอาต์พุต
                                                     ที่อยู่ในช่วงใหม่
  dacWrite(DAC1, newValue);      //กำหนดให้ขา DAC1 มีค่าเท่ากับข้อมูลที่เก็บไว้ในตัวแปร newValue
  Serial.println(newValue);      //แสดงค่าในตัวแปรออกทาง Serial Monitor
  delay(100);
}
```

บันทึกผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....