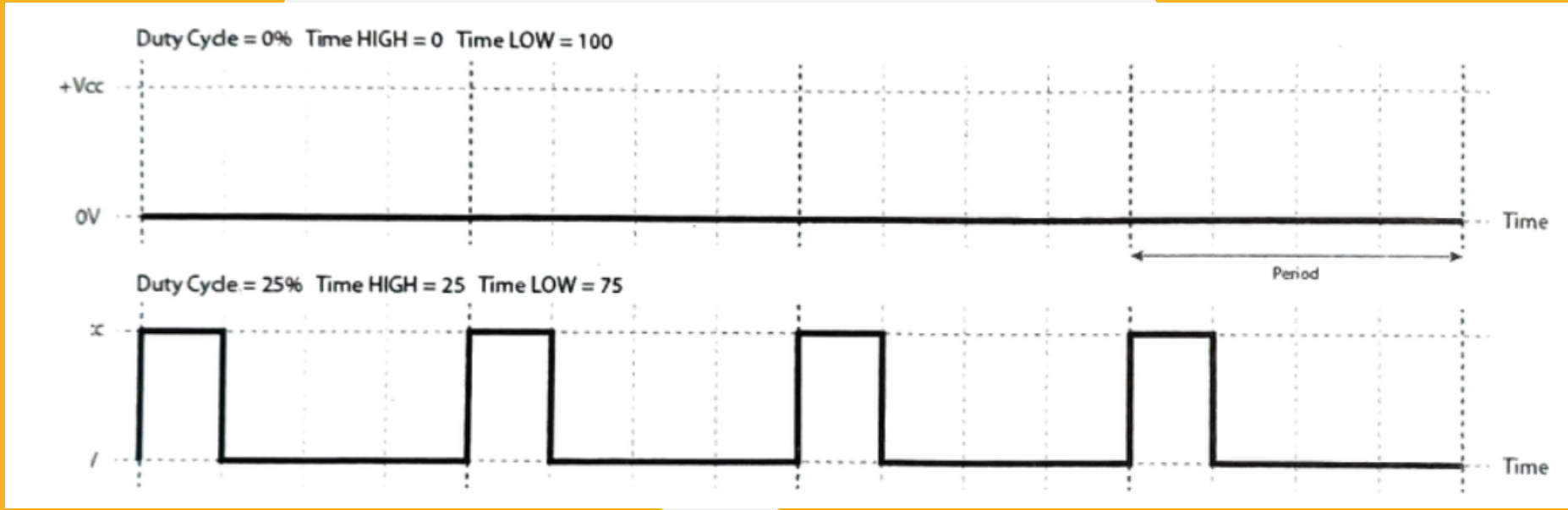
Pulse Width Modulation (PWM)



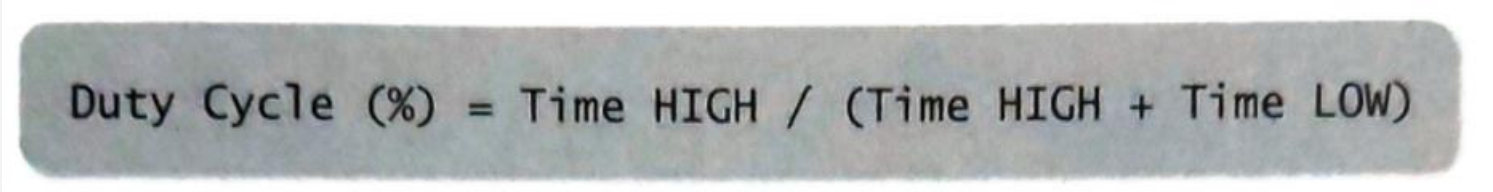
จุดประสงค์การเรียนรู้

1.ศึกษาการวัดแรงดัน Analog จาก PWM

2.ศึกษาการปรับเพิ่ม/ลดความสว่างของหลอดไฟ LED

3.ศึกษาการใช้ Serial Monitor ควบคุมความสว่างของหลอดไฟ LED

4.ศึกษาการปรับความสว่างของหลอดไฟ LED ด้วยตัวต้านทานปรับค่าได้



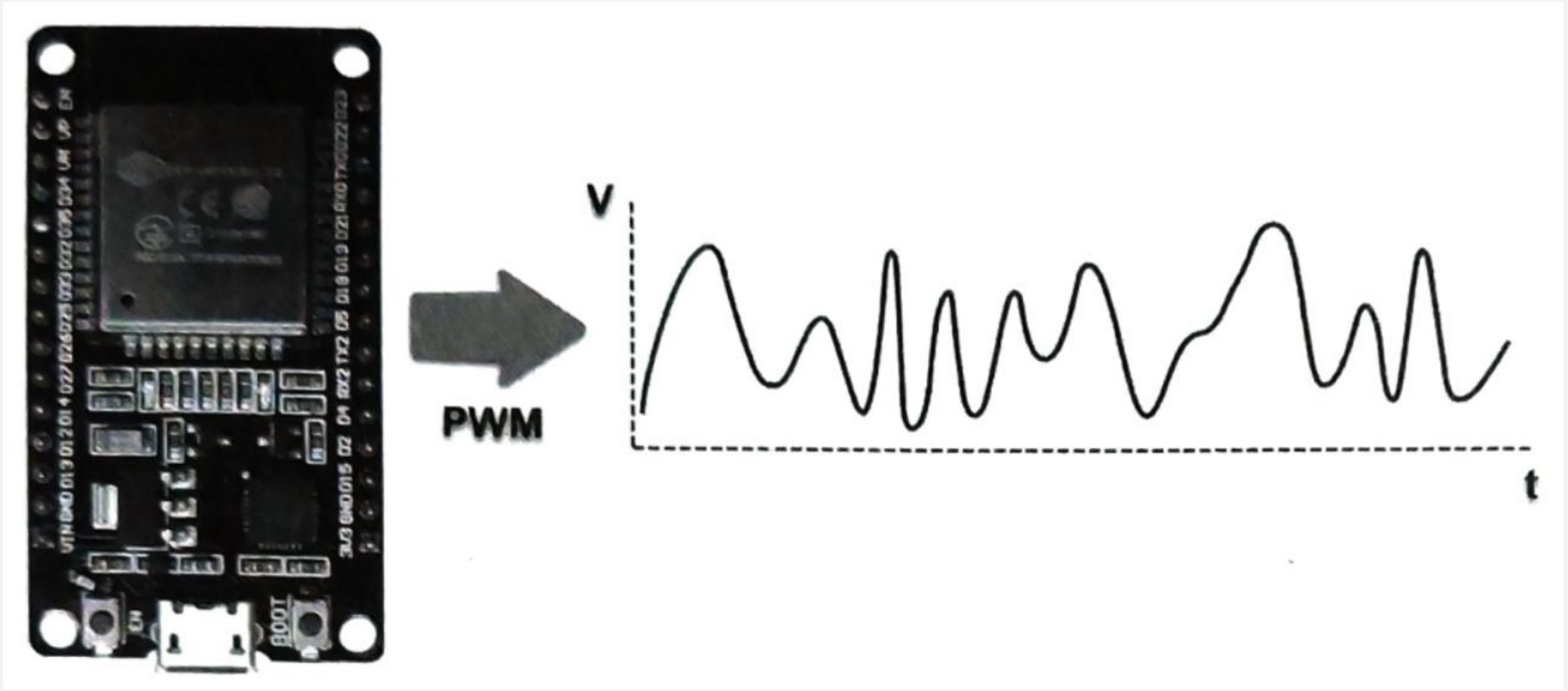
-Duty Cycle มีค่าเป็น 0%

-Duty Cycle มีค่าเป็น 25%

-Duty Cycle มีค่าเป็น 50%

-Duty Cycle มีค่าเป็น 75%

-Duty Cycle มีค่าเป็น 100%



$$V_{out} \text{ (avg.)} = \text{Duty Cycle} \times Vcc$$

คำสั่งการสร้างรูปแบบของสัญญาณ PWM

```
ledcSetup(pinChannel, freq, resolution);
```

การตั้งค่าให้กับรูปแบบของสัญญาณ PWM ที่เราจะสร้าง โดยเลือกกำหนด **pinChannel** หรือช่องที่ใช้สร้างสัญญาณ PWM ที่มีให้เลือกตั้งแต่ 0-15 รวมทั้งสิ้น 16 Channel จากนั้นเลือกกำหนดความถี่ (**freq**) ให้กับสัญญาณ PWM ซึ่งจะมากหรือน้อยขึ้นอยู่กับ resolution ที่เลือกใช้ สุดท้ายเลือกกำหนดความละเอียดหรือ **resolution** ให้กับสัญญาณ PWM ที่มีให้เลือกตั้งแต่ 1-16 bit หรือ 2^{16} = 65536 ระดับ แต่ที่นิยมใช้กันมักจะเป็น 8, 10 หรือไม่ก็ 12 bit ซึ่งถ้าเป็นความละเอียดที่ 10 bit หรือ 2^{10} = 1024 ระดับ ก็จะสามารถระบุค่าเป็นตัวเลขในแบบอนาล็อกได้ตั้งแต่ 0-1023 ตัวอย่างการตั้งค่า เช่น ledcSetup(1, 3000, 10);

```
ledcAttachPin(Pin, pinChannel);
```

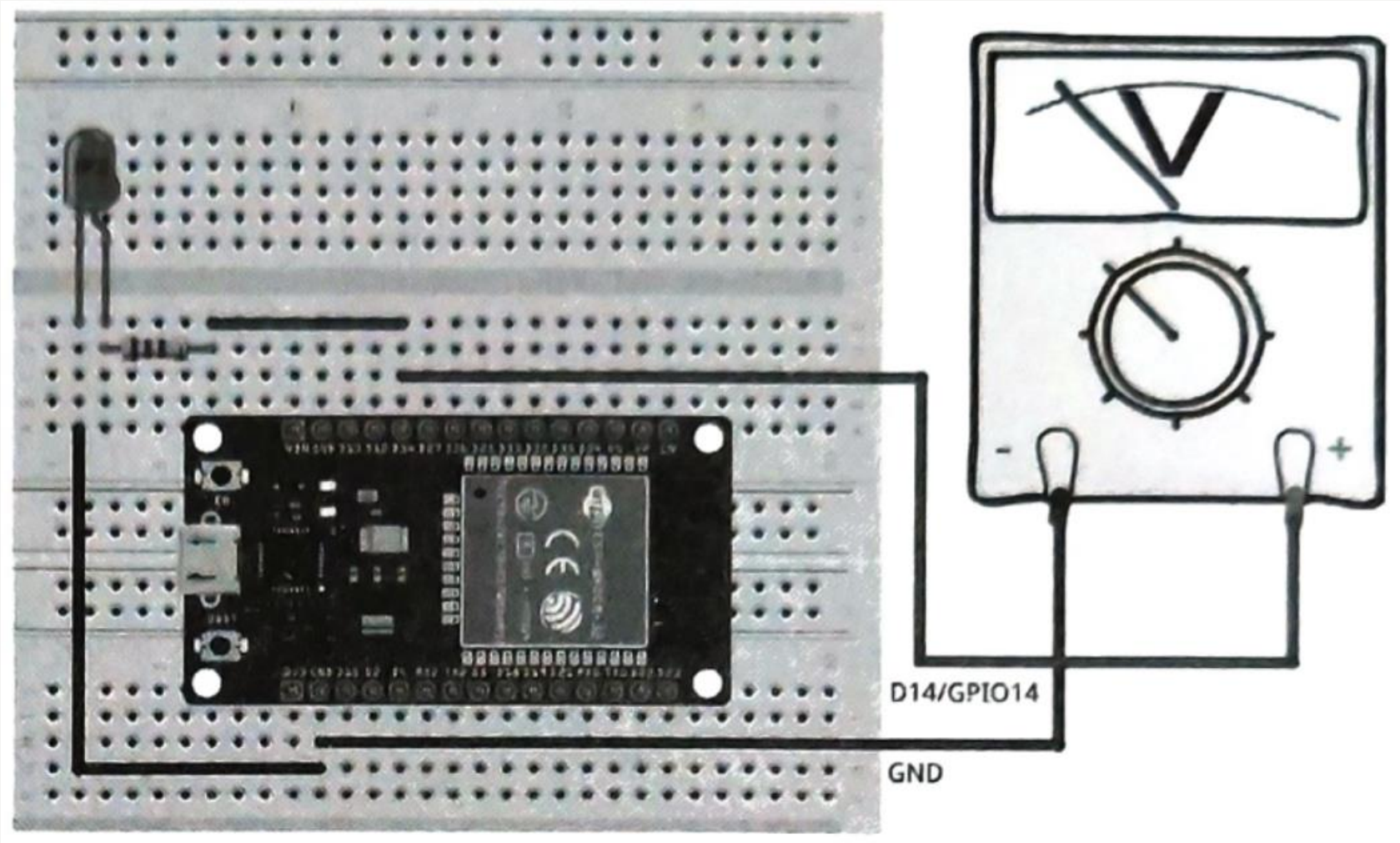
กำหนด Pin หรือขา GPIO ที่ใช้ในการควบคุมหรือสร้างสัญญาณ PWM และ **pinChannel** ที่เป็นช่องสัญญาณที่ได้เลือกกำหนดไว้ในการตั้งค่า ตัวอย่างเช่น ledcAttachPin(5, 1);

```
ledcWrite(pinChannel, value);
```

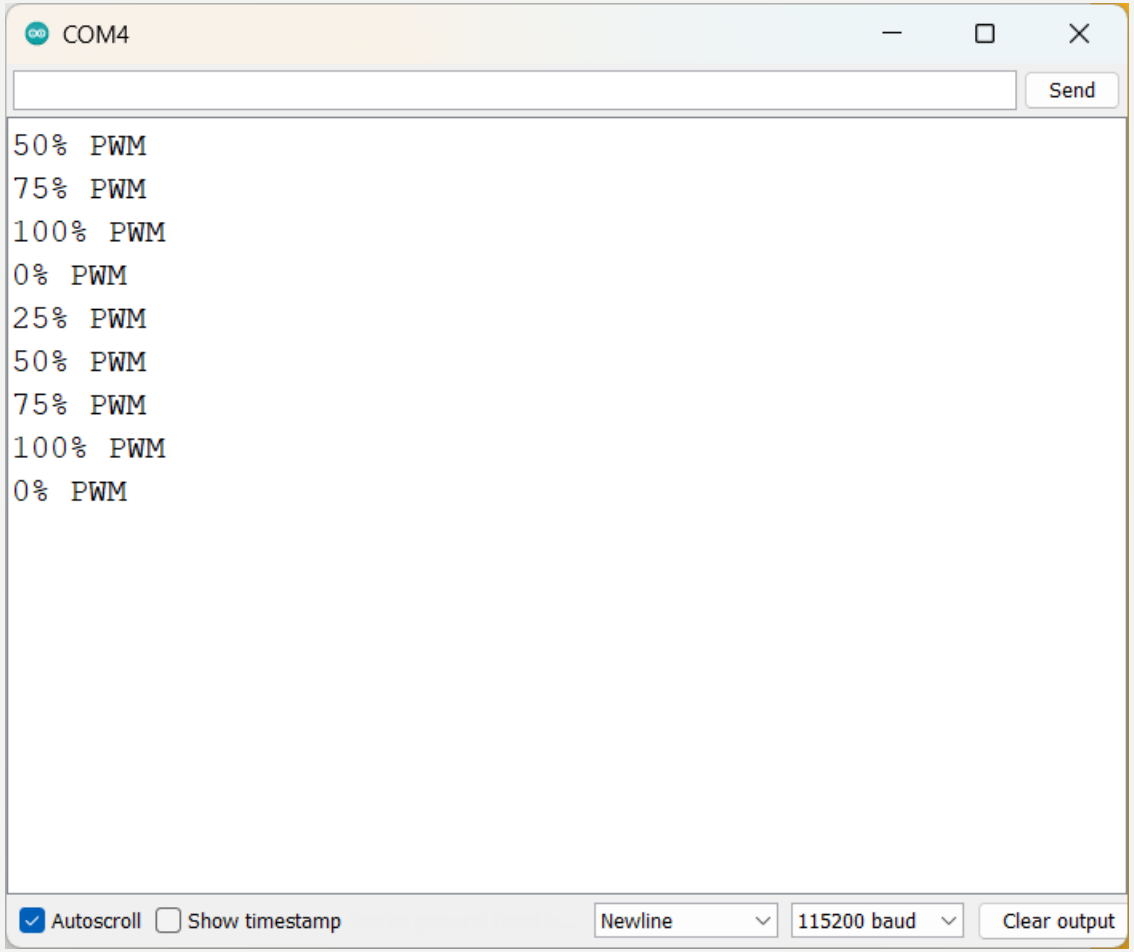
กำหนดให้แสดงค่าตัวเลขในแบบอนาล็อก (ขึ้นอยู่กับความละเอียดที่เลือกใช้) ที่ช่องสัญญาณ PWM หรือ **pinChannel** ที่ได้เลือกกำหนดไว้ จะเห็นว่าต้องมีการระบุค่า value ที่ซึ่งเป็นค่าตัวเลขของสัญญาณอนาล็อก เพราะฉะนั้นในการตั้งค่าให้กับรูปแบบของสัญญาณ PWM ถ้าเราได้เลือกกำหนดความละเอียดหรือ resolution เอาไว้ เช่น 10 bit หรือ 2^{10} = 1024 ระดับ ก็จะสามารถระบุค่าเป็นตัวเลขในแบบอนาล็อกได้ตั้งแต่ 0-1023 ซึ่งเทียบได้กับค่า Duty Cycle ที่อยู่ในช่วงตั้งแต่ 0-100% นั่นเอง

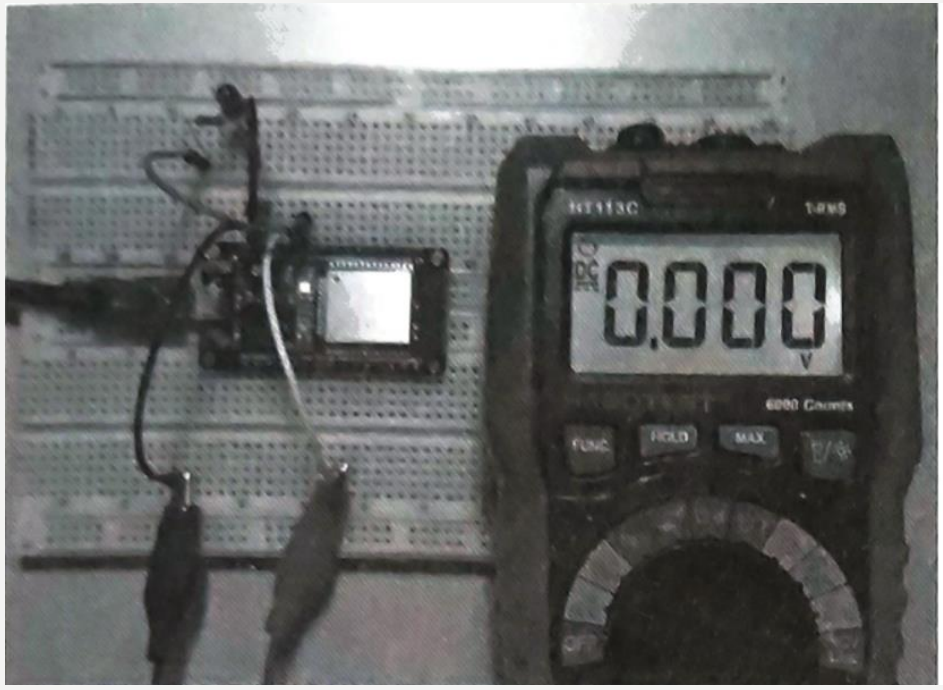
```
ledcWrite(pinChannel, 0);      //ได้เอาต์พุตที่มีแรงดันเฉลี่ย 0V โดยคิดจาก (0% x 3.3V)
ledcWrite(pinChannel, 255);    //ได้เอาต์พุตที่มีแรงดันเฉลี่ย 0.825V โดยคิดจาก (25% x 3.3V)
ledcWrite(pinChannel, 512);    //ได้เอาต์พุตที่มีแรงดันเฉลี่ย 1.65V โดยคิดจาก (50% x 3.3V)
ledcWrite(pinChannel, 767);    //ได้เอาต์พุตที่มีแรงดันเฉลี่ย 2.475V โดยคิดจาก (75% x 3.3V)
ledcWrite(pinChannel, 1023);   //ได้เอาต์พุตที่มีแรงดันเฉลี่ย 3.3V โดยคิดจาก (100% x 3.3V)
```

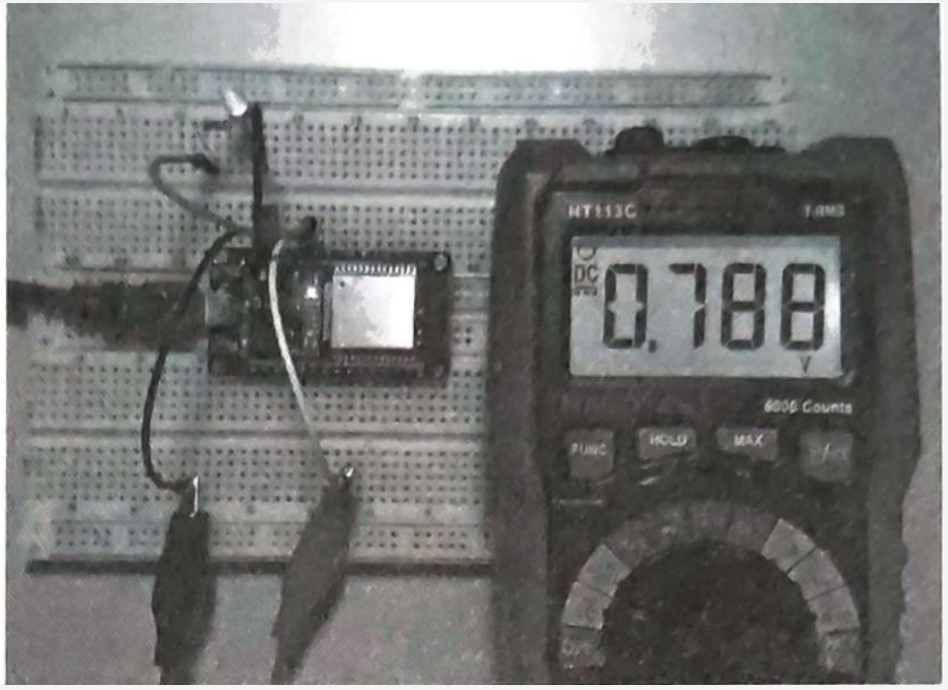
การทดลองที่ 1 การวัดแรงดัน Analog จาก PWM

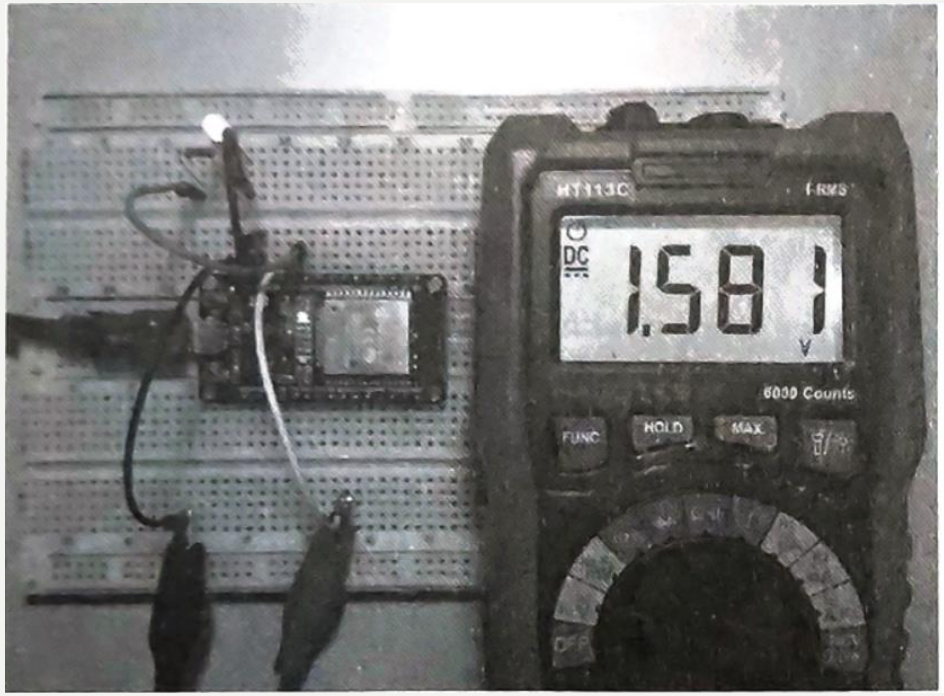


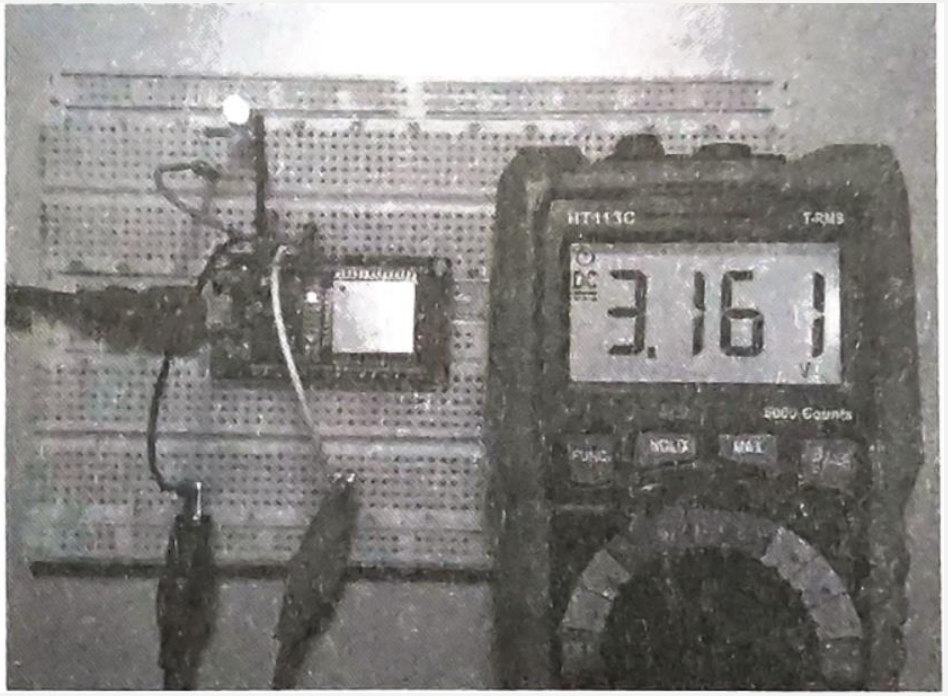
```
#define LED 14
const int pinChannel = 0;
const int freq = 5000;
const int resolution = 10;

void setup() {
  Serial.begin(115200);
  delay(100);
  ledcSetup(pinChannel, freq, resolution);
  ledcAttachPin(LED, pinChannel);
}

void loop() {
  Serial.println("0% PWM");
  ledcWrite(pinChannel, 0);
  delay(3000);

  Serial.println("25% PWM");
  ledcWrite(pinChannel, 255);
  delay(3000);

  Serial.println("50% PWM");
  ledcWrite(pinChannel, 512);
  delay(3000);

  Serial.println("75% PWM");
  ledcWrite(pinChannel, 767) ;
  delay(3000);

  Serial.println("100% PWM");
  ledcWrite(pinChannel, 1023);
  delay(3000);
}
```


COM4
Send
50% PWM
75% PWM
100% PWM
0% PWM
25% PWM
50% PWM
75% PWM
100% PWM
0% PWM
Autoscroll
Show timestamp
Newline
115200 baud
Clear output

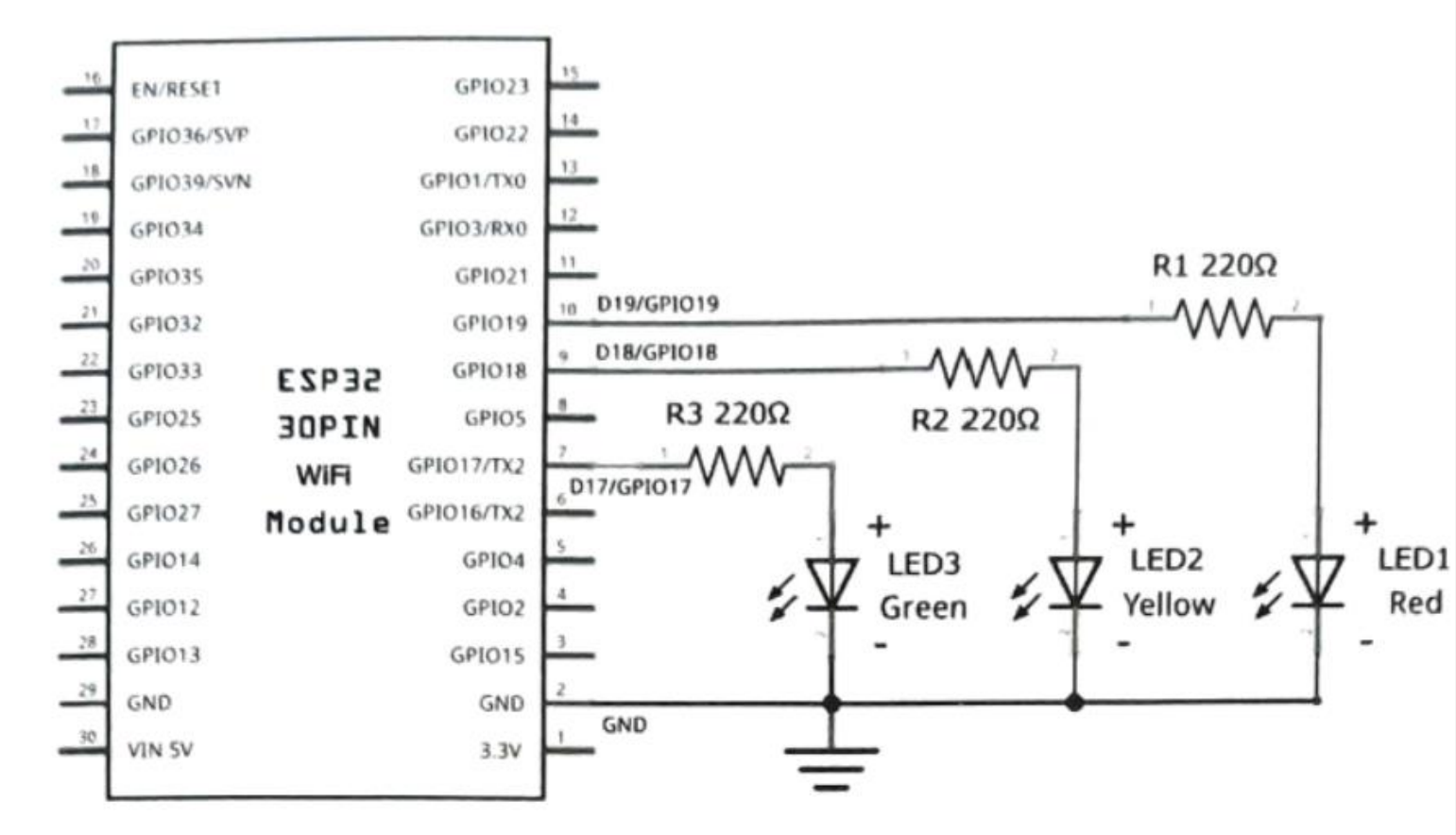

HT113C
T-RMS
0.000
V
6000 Counts
FUNC
HOLD
MAX

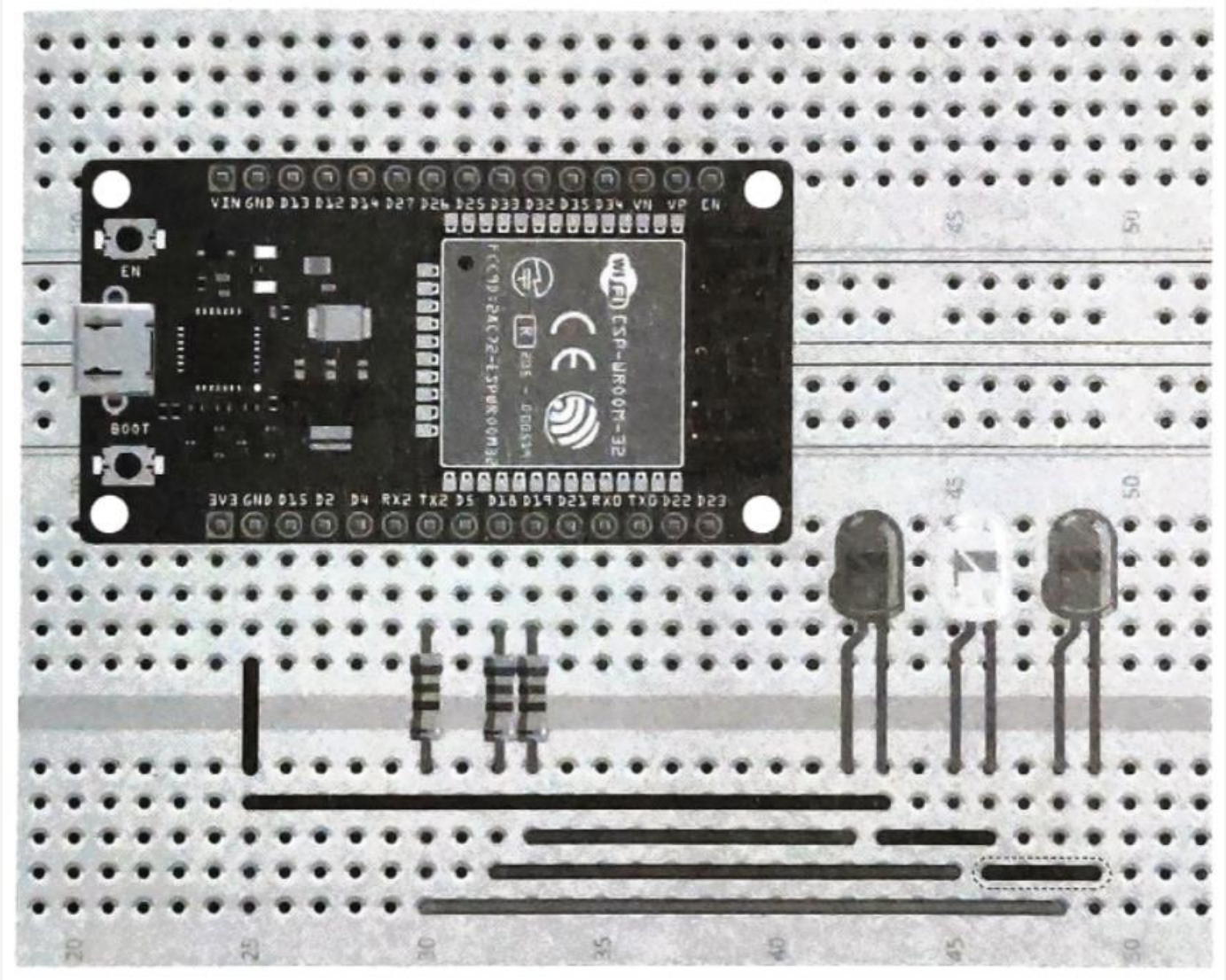

HT113C
T-RMS
0.788
V
6000 Counts
FUNC
HOLD
MAX


HT113C
T-RMS
1.581
V
6000 Counts
FUNC
HOLD
MAX


HT113C
T-RMS
3.161
V
6000 Counts
FUNC
HOLD
MAX

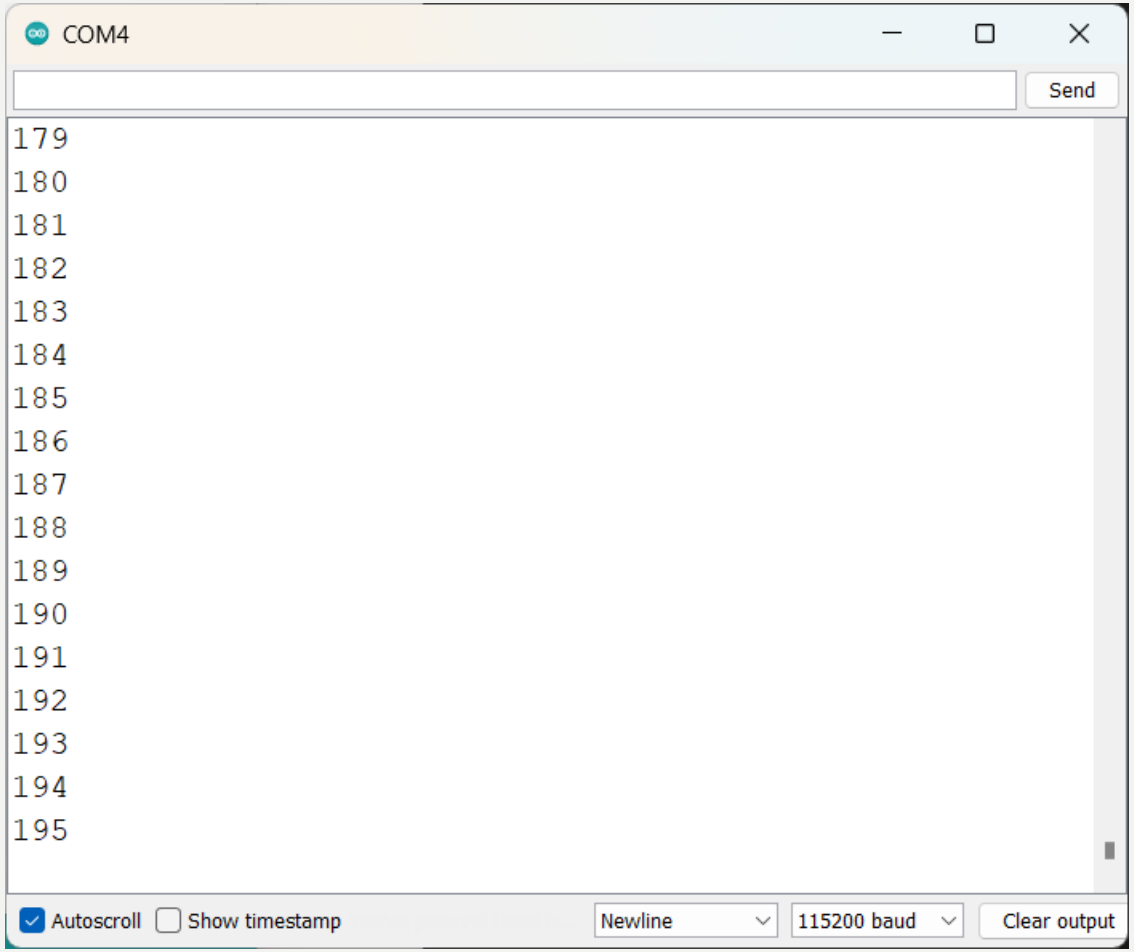
การทดลองที่ 2 การปรับเพิ่ม/ลดความสว่างของหลอดไฟ LED




VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
BOOT
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
ESP-WROOM-32

```
#define LED1 17
#define LED2 18
#define LED3 19
const int pinChannel = 0;
const int freq = 1000;
const int resolution = 8;
int value_PWM;

void setup() {
  Serial.begin(115200);
  ledcSetup(pinChannel, freq, resolution);
  ledcAttachPin(LED1, pinChannel);
  ledcAttachPin (LED2, pinChannel);
  ledcAttachPin (LED3, pinChannel);
}

```

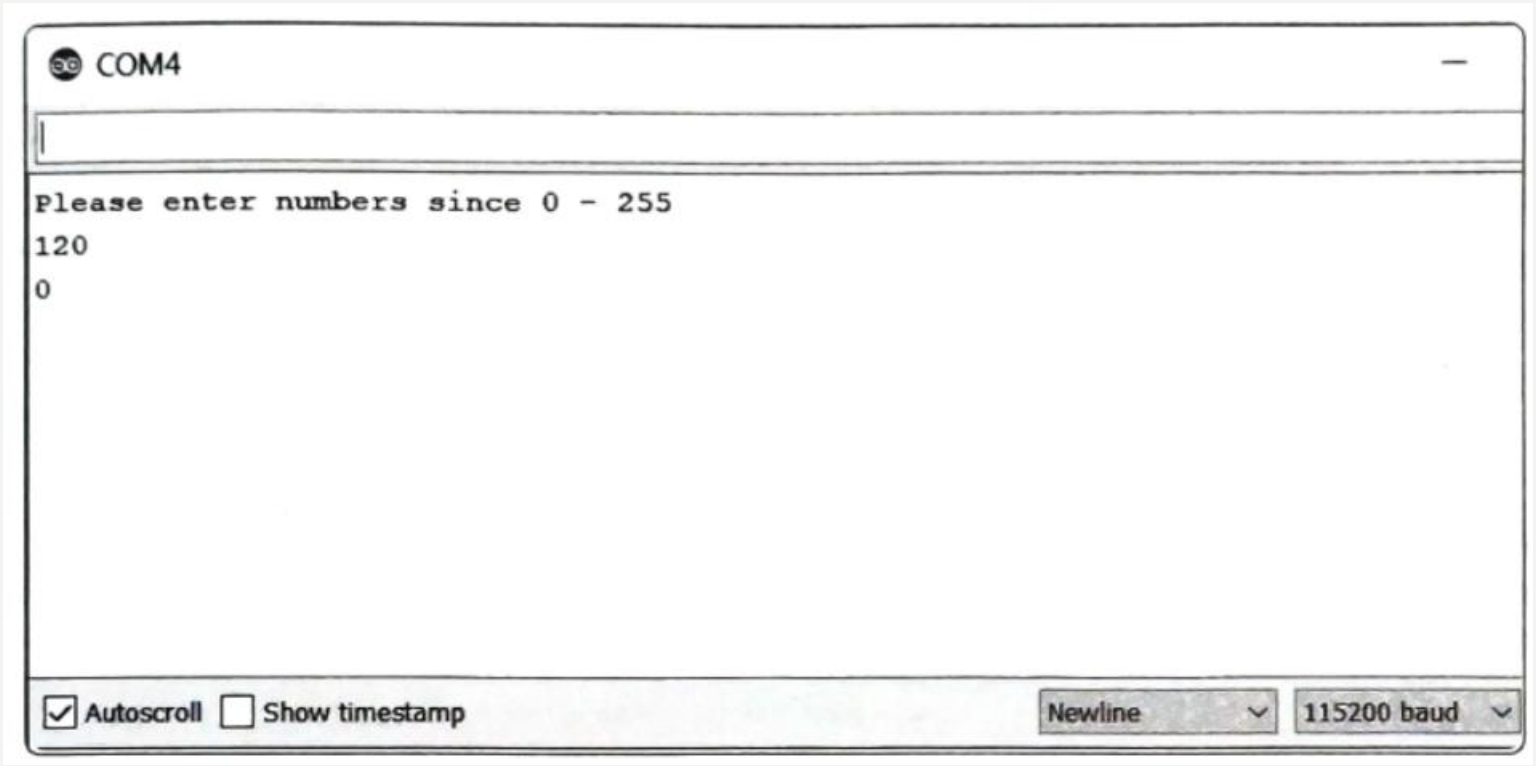
```
void loop()
{
  for (value_PWM = 0; value_PWM < 255; value_PWM++) {
    ledcWrite(pinChannel, value_PWM) ;
    Serial.println(value_PWM) ;
    delay(10);
  }
  for (value_PWM = 255; value_PWM > 0; value_PWM--) {
    ledcWrite(pinChannel, value_PWM);
    Serial.println(value_PWM) ;
    delay(10);
  }
}
```


COM4
Send
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
Autoscroll
Show timestamp
Newline
115200 baud
Clear output

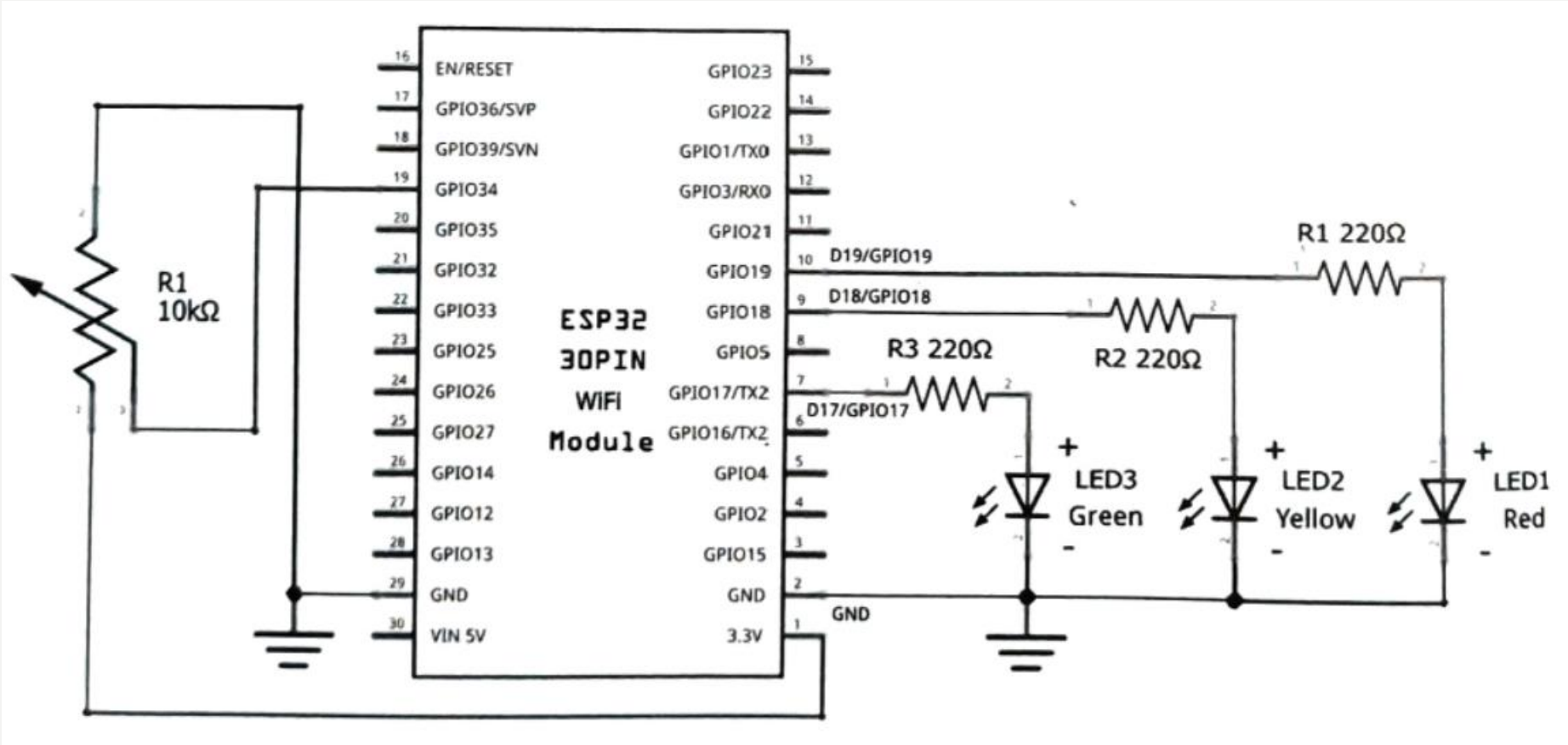
การทดลองที่ 3 การใช้ Serial Monitor ควบคุมความสว่างของ LED

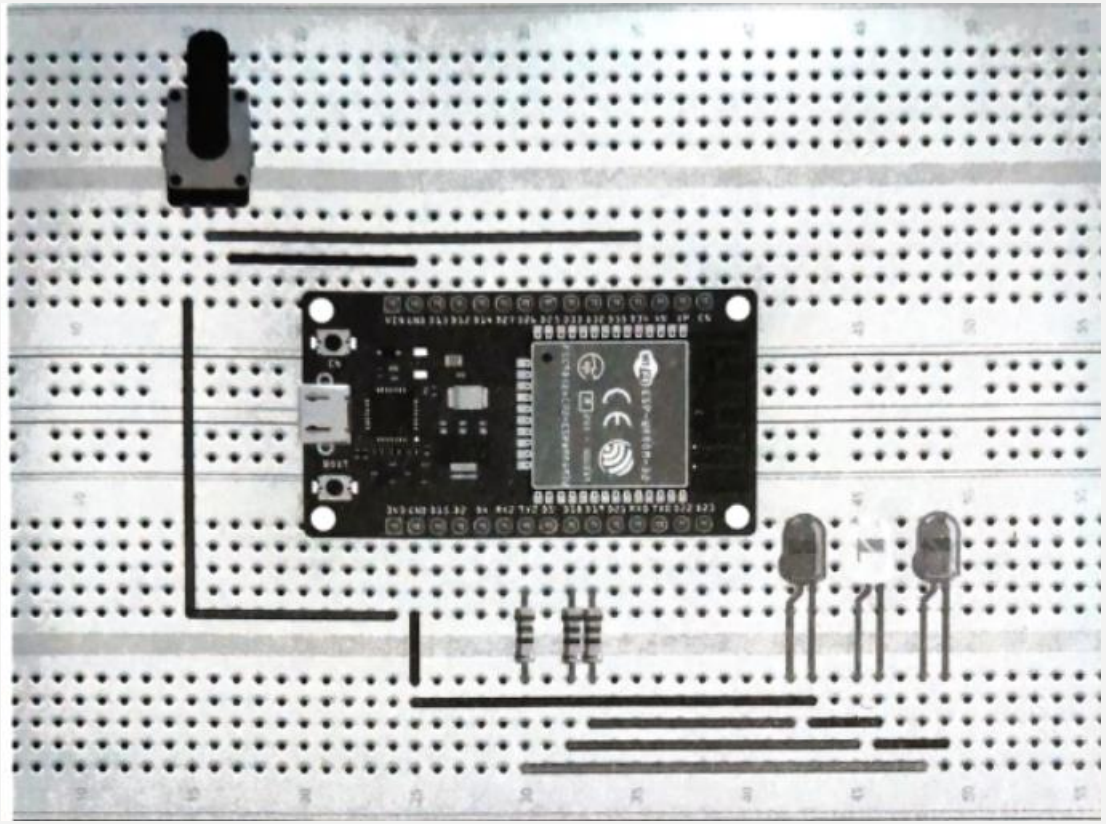
```
#define LED1 17
#define LED2 18
#define LED3 19
const int pinChannel = 0;
const int freq = 1000;
const int resolution = 8;
int value_PWM;

void setup() {
  Serial.begin(115200);
  delay(100);
  Serial.println("Please enter numbers since 0-255");
  ledcSetup(pinChannel, freq, resolution);
  ledcAttachPin(LED1, pinChannel);
  ledcAttachPin(LED2, pinChannel);
  ledcAttachPin(LED3, pinChannel);
}

```

```
void loop() {
  if (Serial.available() > 0)
  {
    value_PWM = Serial.parseInt();
    ledcWrite(pinChannel, value_PWM) ;
    Serial.println(value_PWM) ;
    delay(3000);
  }
}
```

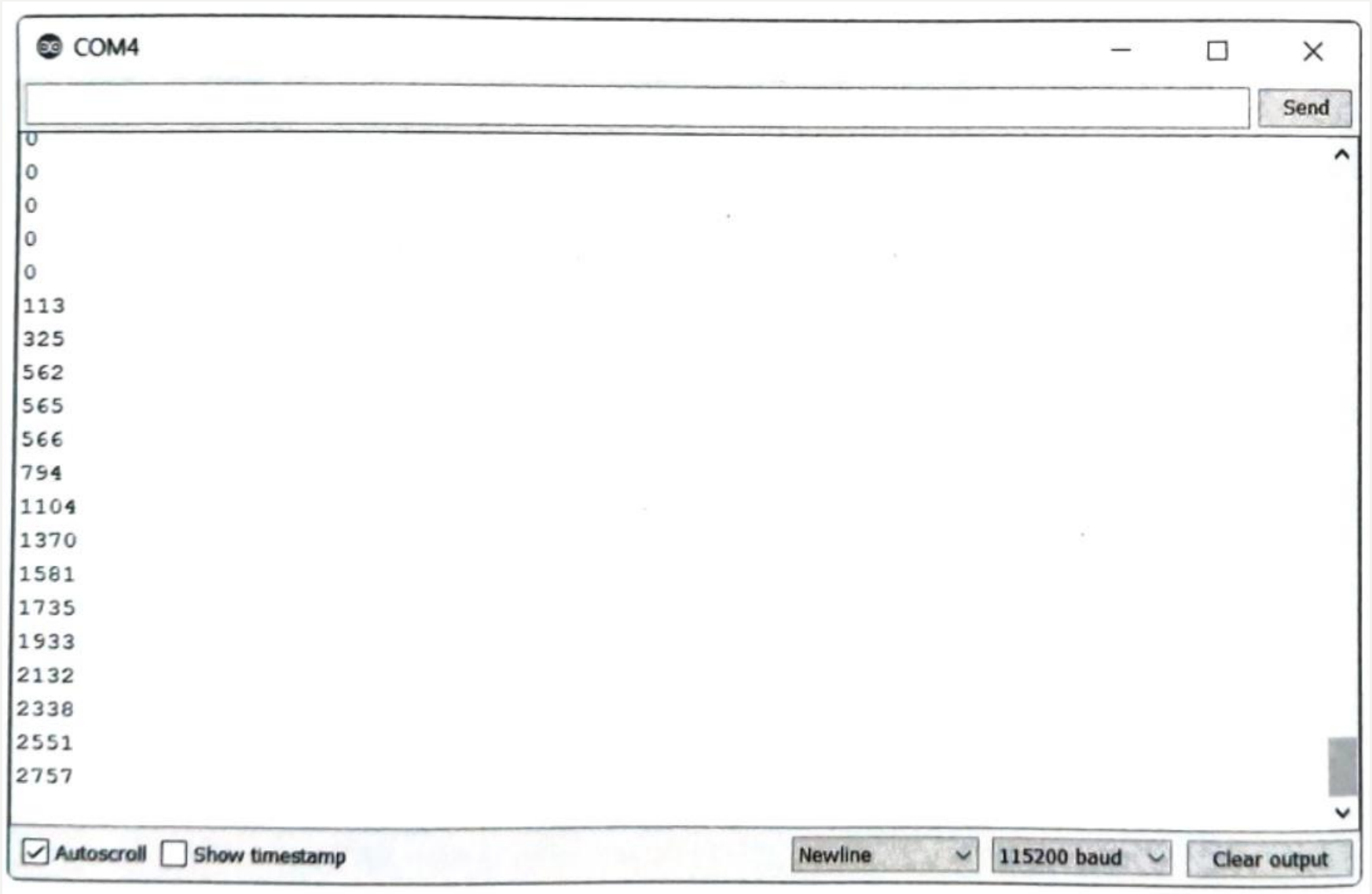

COM4
Please enter numbers since 0 - 255
120
0
Autoscroll
Show timestamp
Newline
115200 baud

การทดลองที่ 4 การปรับความสว่าง LED ด้วยตัวต้านทานปรับค่าได้





```
#define LED1 17
#define LED2 18
#define LED3 19

const int analogInPin = 34;
int sensorValue = 0;

const int pinChannel = 0;
const int freq = 3000;
const int resolution = 12;

void setup() {
  Serial.begin(115200);
  ledcSetup (pinChannel, freq, resolution);
  ledcAttachPin(LED1, pinChannel);
  ledcAttachPin(LED2, pinChannel);
  ledcAttachPin(LED3, pinChannel);
}

void loop() {
  sensorValue = analogRead(analogInPin);
  ledcWrite(pinChannel, sensorValue);
  Serial.println(sensorValue);
  delay(100);
}
```


COM4
Send
0
0
0
0
0
113
325
562
565
566
794
1104
1370
1581
1735
1933
2132
2338
2551
2757
Autoscroll
Show timestamp
Newline
115200 baud
Clear output