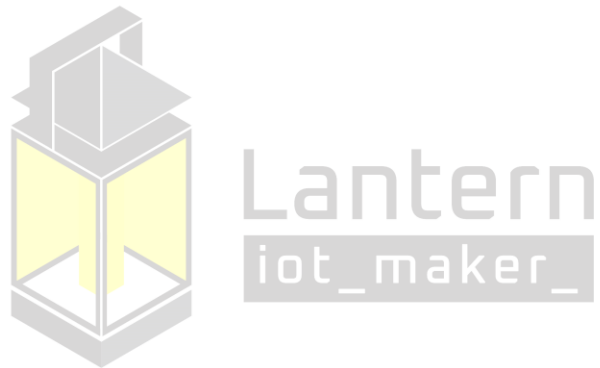


# Lantern IoT maker

Electronic component - Internet of Things - Computer System Services



LANTERN IOT MAKER  
LOGO DESIGN

**เอกสารประกอบการใช้งาน พร้อมตัวอย่างโปรแกรม**

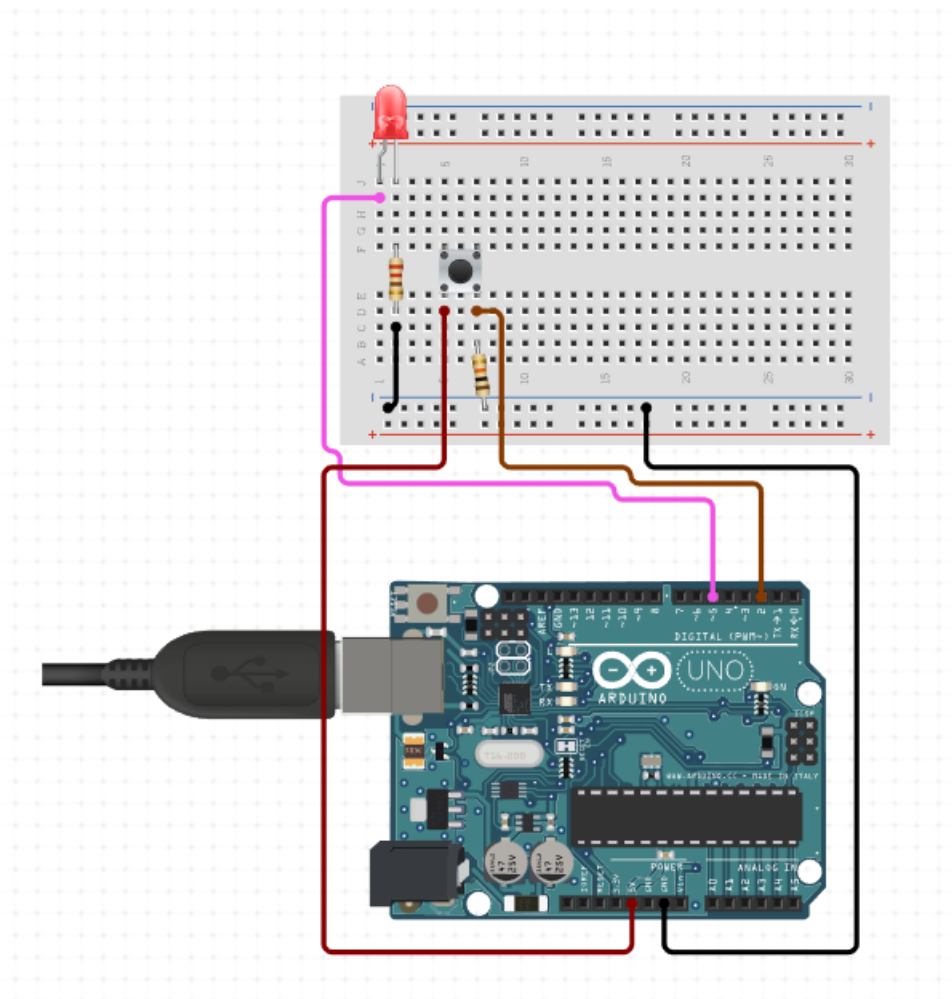
**ชุด Arduino starter kit set / IoT starter kit set**

## แบบฝึกหัดที่ 1 กดปุ่มแล้วไฟติด

### อุปกรณ์

1. Arduino UNO R3 SMD
2. Push button
3. LED
4. ตัวต้านทาน 220 โอห์ม
5. ตัวต้านทาน 10k โอห์ม

### การต่อวงจร



## โปรแกรมสำหรับ upload

Button

// constants won't change. They're used here to set pin numbers:

`const int` buttonPin = 2; // the number of the pushbutton pin

`const int` ledPin = 5; // the number of the LED pin

// variables will change:

`int` buttonState = 0; // variable for reading the pushbutton status

`void setup()` {

// initialize the LED pin as an output:

`pinMode`(ledPin, OUTPUT);

// initialize the pushbutton pin as an input:

`pinMode`(buttonPin, INPUT);

}

`void loop()` {

// read the state of the pushbutton value:

buttonState = `digitalRead`(buttonPin);

// check if the pushbutton is pressed. If it is, the buttonState is HIGH:

`if` (buttonState == HIGH) {

// turn LED on:

`digitalWrite`(ledPin, HIGH);

} `else` {

// turn LED off:

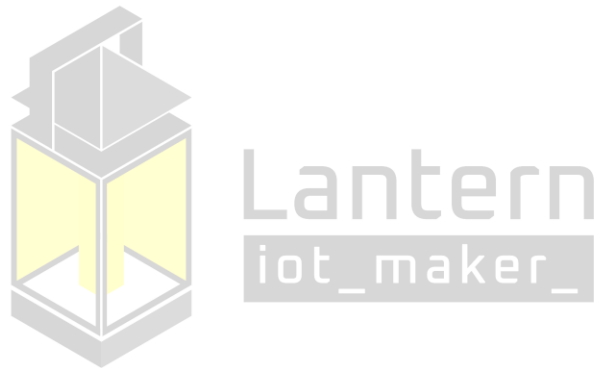
`digitalWrite`(ledPin, LOW);

}

}

## คำอธิบายโปรแกรม

1. เป็นโปรแกรมสำหรับศึกษาการรับค่าสัญญาณ digital โดยการ set ค่า pin ขา digital เป็น input และ output พร้อมทั้งกำหนด หมายเลข pin
2. ใช้คำสั่ง `digitalRead()` ในการอ่านค่าสัญญาณ digital
3. เช็คเงื่อนไขการรับค่าสัญญาณ digital เมื่อรับค่ามาเป็น 1 จะทำให้ไฟติด/ 0 จะทำให้ไฟไม่ติด
4. ใช้คำสั่ง `digitalWrite()` ในการแสดงผลค่าสัญญาณ digital ผ่าน LED



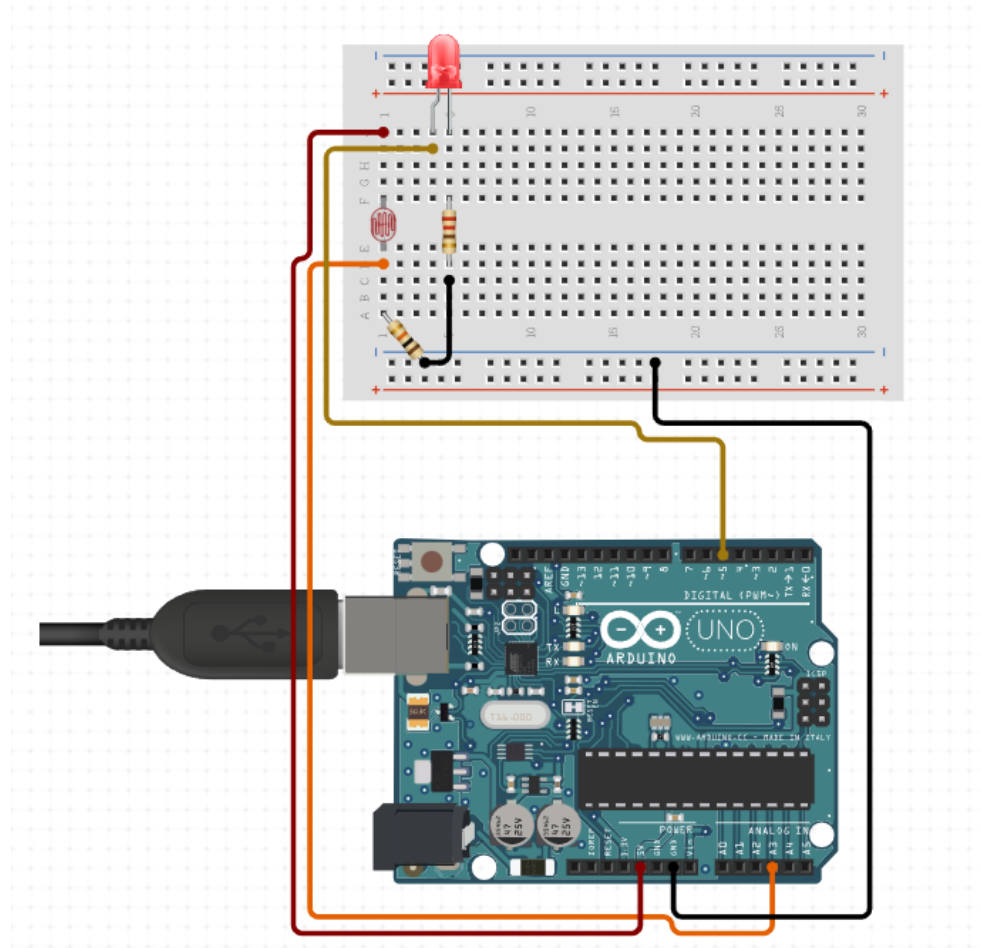
LANTERN IOT MAKER  
LOGO DESIGN

## แบบฝึกหัดที่ 2 วงจรเปิดปิดไปอัตโนมัติด้วย LDR

### อุปกรณ์

1. Arduino UNO R3 SMD
2. LDR
3. LED
4. ตัวต้านทาน 220 โอห์ม
5. ตัวต้านทาน 10k โอห์ม

### การต่อวงจร



## โปรแกรมสำหรับ upload

```
int ledPin = 5;

int analogPin = a3; //ประกาศตัวแปร ให้ analogPin

int val = 0;

void setup() {

  pinMode(ledPin, OUTPUT); // sets the pin as output

  Serial.begin(9600);

}

void loop() {

  val = analogRead(analogPin); //อ่านค่าสัญญาณ analog

  Serial.print("val = "); // พิมพ์ข้อความส่งเข้าคอมพิวเตอร์ "val = "

  Serial.println(val); // พิมพ์ค่าของตัวแปร val

  if (val < 50) { // สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ

    digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง

  }

  else {

    digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ

  }

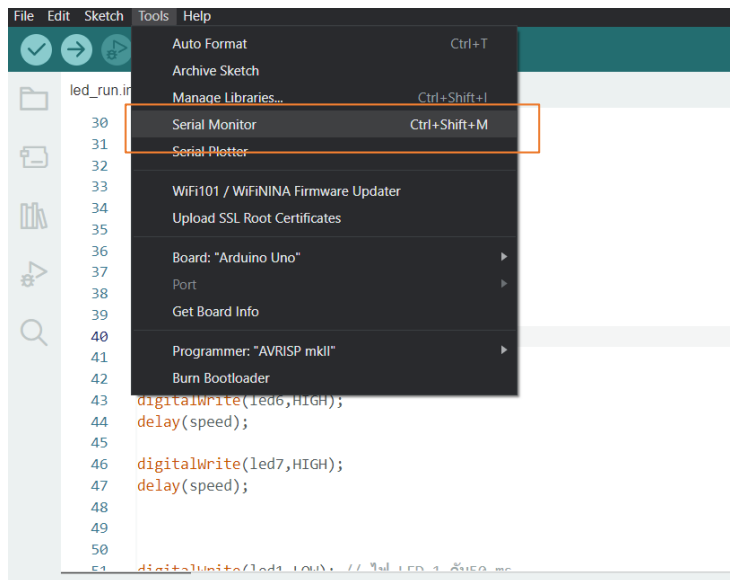
  delay(100);

}
```

LUEN DESIGN

## คำอธิบายโปรแกรม

1. เป็นโปรแกรมสำหรับการศึกษาการอ่านค่า Serial ซึ่งเป็นค่าสัญญาณ analog ที่ได้จาก LDR เป็นตัวรับค่าความสว่างแปลงเป็นสัญญาณไฟฟ้าแสดงค่าเป็น 0-1024
2. ใช้คำสั่ง Serial.begin() และ Serial.println ในการแสดงค่าที่พิมพ์ผ่าน Serial monitoring



3. ใช้คำสั่ง `analogRead()` ในการอ่านค่าสัญญาณไฟฟ้า ผ่าน analog pin a3



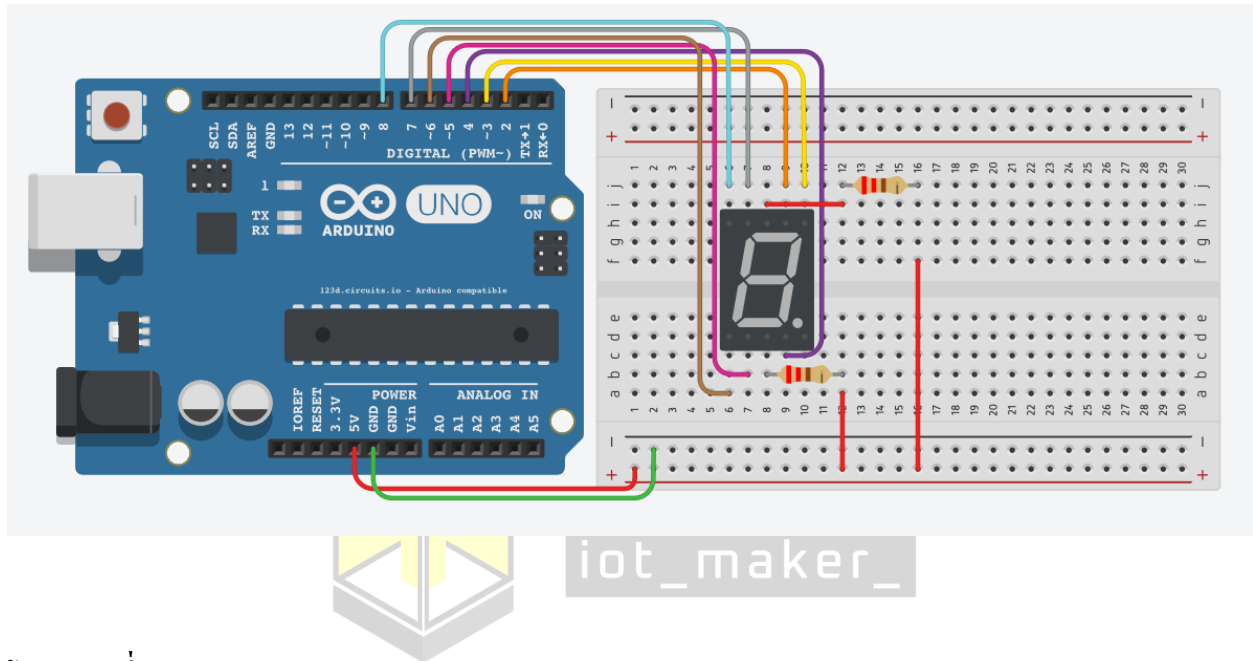
LANTERN IOT MAKER  
LOGO DESIGN

## แบบฝึกหัดที่ 3 โปรแกรม โดยใช้ Counter 7 segment

### อุปกรณ์

1. Arduino UNO R3 SMD
2. 7 segment 1 digit
3. ตัวต้านทาน 220 โอห์ม

### การต่อวงจร



### โปรแกรมที่ upload

```
#define segA 2//connecting segment A to PIN2
#define segB 3// connecting segment B to PIN3
#define segC 4// connecting segment C to PIN4
#define segD 5// connecting segment D to PIN5
#define segE 6// connecting segment E to PIN6
#define segF 7// connecting segment F to PIN7
#define segG 8// connecting segment G to PIN8
int speed = 500;
int COUNT=0;//count integer for 0-9 increment
void setup()
{
```

```

    for (int i=2;i<9;i++)
    {
        pinMode(i, OUTPUT);// taking all pins from 2-8 as output
    }
}

void loop()
{
    switch (COUNT)
    {
        case 0://when count value is zero show"0" on disp
            digitalWrite(segA, HIGH);
            digitalWrite(segB, HIGH);
            digitalWrite(segC, HIGH);
            digitalWrite(segD, HIGH);
            digitalWrite(segE, HIGH);
            digitalWrite(segF, HIGH);
            digitalWrite(segG, LOW);
            break;

        case 1:// when count value is 1 show"1" on disp
            digitalWrite(segA, LOW);
            digitalWrite(segB, HIGH);
            digitalWrite(segC, HIGH);
            digitalWrite(segD, LOW);
            digitalWrite(segE, LOW);
            digitalWrite(segF, LOW);
            digitalWrite(segG, LOW);
            break;

        case 2:// when count value is 2 show"2" on disp
            digitalWrite(segA, HIGH);

```

```
digitalWrite(segB, HIGH);
digitalWrite(segC, LOW);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, LOW);
digitalWrite(segG, HIGH);
break;

case 3:// when count value is 3 show"3" on disp
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);
digitalWrite(segF, LOW);
digitalWrite(segG, HIGH);
break;

case 4:// when count value is 4 show"4" on disp
digitalWrite(segA, LOW);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
break;

case 5:// when count value is 5 show"5" on disp
digitalWrite(segA, HIGH);
digitalWrite(segB, LOW);
digitalWrite(segC, HIGH);
```

```
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
break;

case 6:// when count value is 6 show"6" on disp
digitalWrite(segA, HIGH);
digitalWrite(segB, LOW);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
break;

case 7:// when count value is 7 show"7" on disp
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);
digitalWrite(segF, LOW);
digitalWrite(segG, LOW);
break;

case 8:// when count value is 8 show"8" on disp
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
```

```

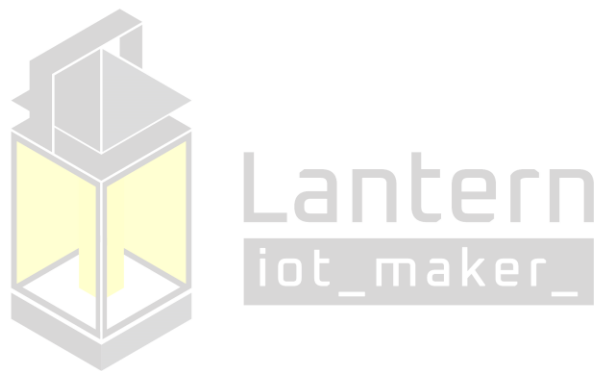
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
break;
case 9:// when count value is 9 show"9" on disp
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
break;
        default:
break;
}
if (COUNT<10)
{
    COUNT++;
    delay(speed);///increment count integer for every second
}
if (COUNT==10)
{
    COUNT=0;// if count integer value is equal to 10, reset it to zero.
    delay(speed);
}
}

```

### คำอธิบายโปรแกรม

1. เป็นโปรแกรมทดสอบการใช้งาน 7 segment โดคนการนับจาก 0 ถึง 9 แล้วกลับมาเริ่มต้นใหม่ วนไปเรื่อยๆ

2. ศึกษาในเรื่องของการวนลูป โดยใช้ for และ if ในการกำหนดเงื่อนไข
3. ศึกษาการใช้ switch / case ในการกำหนดเงื่อนไขแบบตัวเลือก



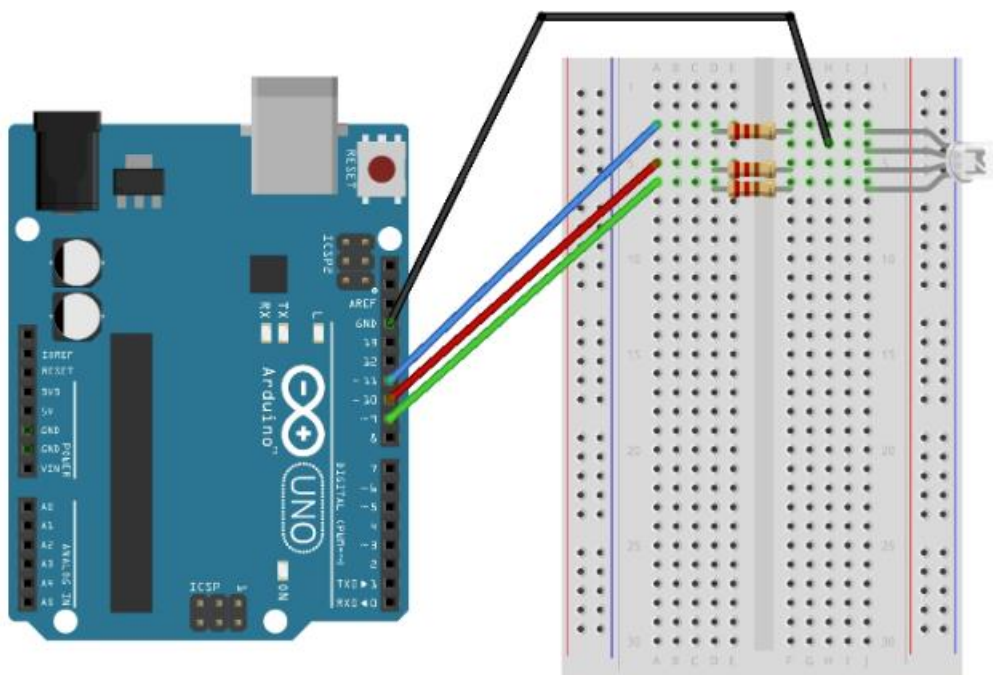
LANTERN IOT MAKER  
LOGO DESIGN

## แบบฝึกหัดที่ 4 RGB ไฟเปลี่ยนสี RGB 16 ล้านสี

### อุปกรณ์

1. Arduino UNO R3 SMD
2. 7 segment 1 digit
3. ตัวต้านทาน 220 โอห์ม

### การต่อวงจร



LANTERN IOT MAKER  
LOGO DESIGN

### โปรแกรมที่ upload

```
const int RED_PIN = 9;
const int GREEN_PIN = 10;
const int BLUE_PIN = 11;

int DISPLAY_TIME = 100; // In milliseconds
int speed = 500;
```

```
void setup()
{
  pinMode(RED_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(BLUE_PIN, OUTPUT);
}

void loop()
{
  mainColors();
  showSpectrum();
}

void mainColors()
{
  digitalWrite(RED_PIN, LOW);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BLUE_PIN, LOW);

  delay(speed);

  // Red (turn just the red LED on):

  digitalWrite(RED_PIN, HIGH);
  digitalWrite(GREEN_PIN, LOW);
  digitalWrite(BLUE_PIN, LOW);

  delay(speed);

  // Green (turn just the green LED on):
```

```
digitalWrite(RED_PIN, LOW);  
digitalWrite(GREEN_PIN, HIGH);  
digitalWrite(BLUE_PIN, LOW);  
  
delay(speed);
```

```
// Blue (turn just the blue LED on):
```

```
digitalWrite(RED_PIN, LOW);  
digitalWrite(GREEN_PIN, LOW);  
digitalWrite(BLUE_PIN, HIGH);  
  
delay(speed);
```

```
// Yellow (turn red and green on):
```

```
digitalWrite(RED_PIN, HIGH);  
digitalWrite(GREEN_PIN, HIGH);  
digitalWrite(BLUE_PIN, LOW);  
  
delay(speed);
```

```
// Cyan (turn green and blue on):
```

```
digitalWrite(RED_PIN, LOW);  
digitalWrite(GREEN_PIN, HIGH);  
digitalWrite(BLUE_PIN, HIGH);
```

```
delay(speed);
```

```
// Purple (turn red and blue on):
```

```
digitalWrite(RED_PIN, HIGH);
```

```
digitalWrite(GREEN_PIN, LOW);
```

```
digitalWrite(BLUE_PIN, HIGH);
```

```
delay(speed);
```

```
// White (turn all the LEDs on):
```

```
digitalWrite(RED_PIN, HIGH);
```

```
digitalWrite(GREEN_PIN, HIGH);
```

```
digitalWrite(BLUE_PIN, HIGH);
```

```
delay(speed);
```

```
}
```

```
void showSpectrum()
```

```
{
```

```
int x; // define an integer variable called "x"
```

```
for (x = 0; x < 768; x++)
```

```
{
```

```
showRGB(x); // Call RGBspectrum() with our new x
```

```
delay(10); // Delay for 10 ms (1/100th of a second)
```

```
}
```

```
}
```

```
void showRGB(int color)
```

```
{
```

```

int redIntensity;
int greenIntensity;
int blueIntensity;

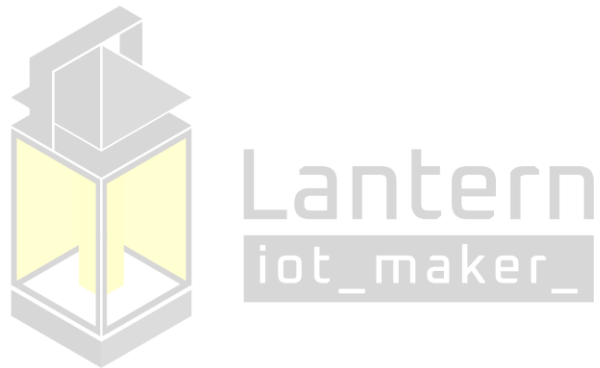
if (color <= 255)      // zone 1
{
    redIntensity = 255 - color;  // red goes from on to off
    greenIntensity = color;      // green goes from off to on
    blueIntensity = 0;          // blue is always off
}
else if (color <= 511)  // zone 2
{
    redIntensity = 0;          // red is always off
    greenIntensity = 255 - (color - 256); // green on to off
    blueIntensity = (color - 256); // blue off to on
}
else // color >= 512    // zone 3
{
    redIntensity = (color - 512); // red off to on
    greenIntensity = 0;          // green is always off
    blueIntensity = 255 - (color - 512); // blue on to off
}

analogWrite(RED_PIN, redIntensity);
analogWrite(BLUE_PIN, blueIntensity);
analogWrite(GREEN_PIN, greenIntensity);
}

```

## คำอธิบายโปรแกรม

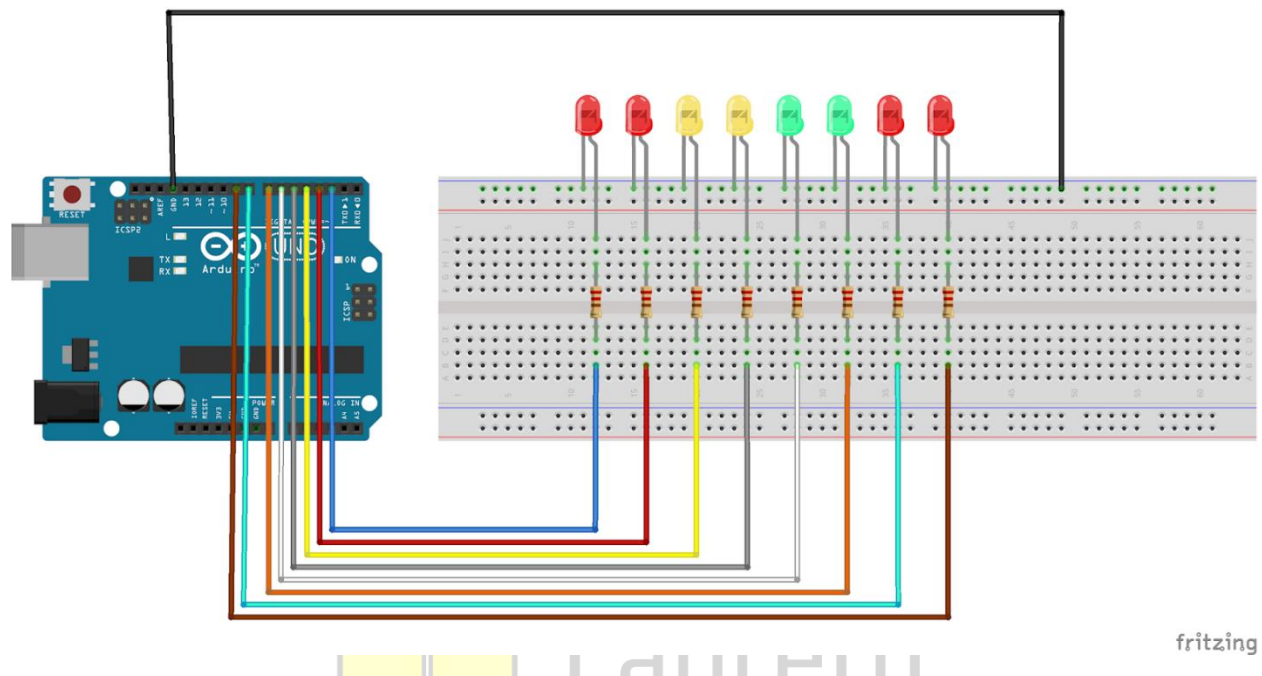
1. เป็นโปรแกรมสำหรับควบคุมการแสดงสี และการผสมสีของหลอด RGB โดยสามารถแสดงได้สูงสุดถึง 16 ล้านสี
2. ฝึกควบคุมความสว่างของหลอดไฟโดยใช้สัญญาณแบบ digital ด้วยคำสั่ง `digitalWrite ()`
3. ฝึกควบคุมความสว่างของหลอดไฟโดยใช้สัญญาณแบบ analog ด้วยคำสั่ง `analogWrite ()` โดยค่าสีที่แสดงออกมามีเริ่มต้นที่ 0 ถึง 255 ทั้งสีแดง เขียว น้ำเงิน
4. การเขียนโค้ดควบคุมการทำงานโดยใช้ ฟังก์ชัน เช่น `mainColors()` แสดงแม่สี 3 สี และการผสมสีอย่างง่าย, `showSpectrum()` แสดงการไล่สีด้วยสัญญาณแบบ analog



LANTERN IOT MAKER  
LOGO DESIGN

## แบบฝึกหัด เพิ่มเติม และแบบทดสอบ

### การต่อวงจร



```
int led1 = 2; // กำหนดขาใช้งาน
```

```
int led2 = 3;
```

```
int led3 = 4;
```

```
int led4 = 5;
```

```
int led5 = 6;
```

```
int led6 = 7;
```

```
int led7 = 8;
```

```
int led8 = 9;
```

```
int speed = 150;
```

```
void setup()
```

```
{
```

```
pinMode(led1, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา 2 เป็น OUTPUT
```

```
pinMode(led2, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา 3 เป็น OUTPUT
```

```
pinMode(led3, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา 4 เป็น OUTPUT
```

```
pinMode(led4, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา 5 เป็น OUTPUT
```

```
pinMode(led5, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา 6 เป็น OUTPUT
pinMode(led6, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา 6 เป็น OUTPUT
pinMode(led7, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา 6 เป็น OUTPUT
pinMode(led8, OUTPUT); // กำหนดขาทำหน้าที่ให้ขา 6 เป็น OUTPUT
digitalWrite(led1,LOW);
digitalWrite(led2,LOW);
digitalWrite(led3,LOW);
digitalWrite(led4,LOW);
digitalWrite(led5,LOW);
digitalWrite(led6,LOW);
digitalWrite(led7,LOW);
digitalWrite(led8,LOW);
}
void loop()
{
digitalWrite(led1,HIGH); // ไฟ LED 1 ติด 50 ms
delay(speed);
digitalWrite(led2,HIGH);
delay(speed);
digitalWrite(led3,HIGH);
delay(speed);
digitalWrite(led4,HIGH);
delay(speed);
digitalWrite(led5,HIGH);
delay(speed);
digitalWrite(led6,HIGH);
delay(speed);
digitalWrite(led7,HIGH);
delay(speed);
```

```

digitalWrite(led8,HIGH);
delay(speed);
digitalWrite(led1,LOW); // ไฟ LED 1 ดับ 50 ms
delay(speed);
digitalWrite(led2,LOW);
delay(speed);
digitalWrite(led3,LOW);
delay(speed);
digitalWrite(led4,LOW);
delay(speed);
digitalWrite(led5,LOW);
delay(speed);
digitalWrite(led6,LOW);
delay(speed);
digitalWrite(led7,LOW);
delay(speed);
digitalWrite(led8,LOW);
delay(speed);
}

```

คำถาม

1. วงจรนี้ใช้ทำอะไร
2. ลองต่อตาม แล้วผลที่ได้เป็นอย่างไร
3. วงจรนี้ใช้อุปกรณ์อะไรบ้าง
4. ตัวแปร speed คืออะไร เกิดอะไรขึ้น เมื่อเปลี่ยนค่า speed =50, speed=150, speed=300