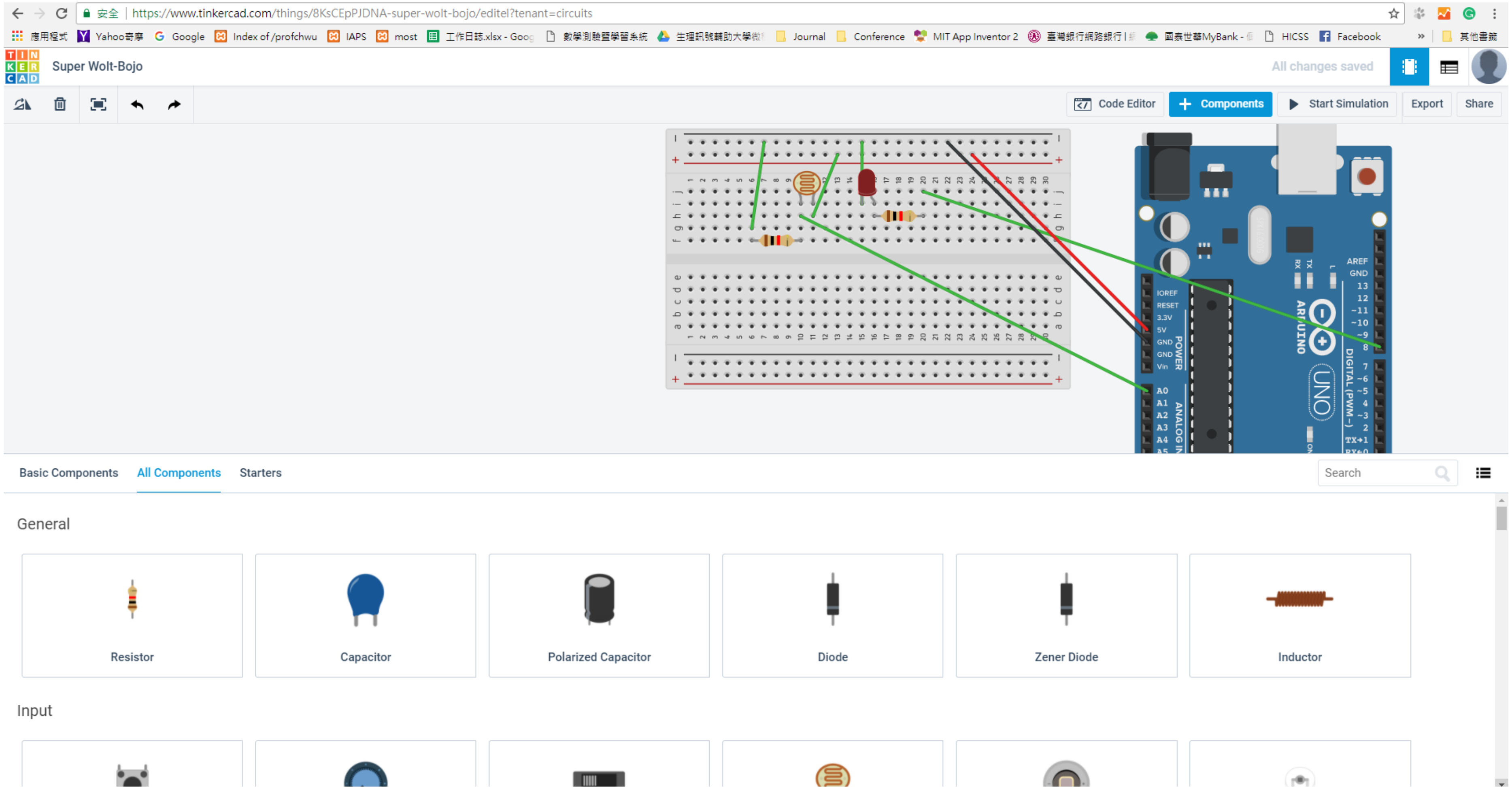
Arduino範例教學
三色RGB LED燈

吳智鴻

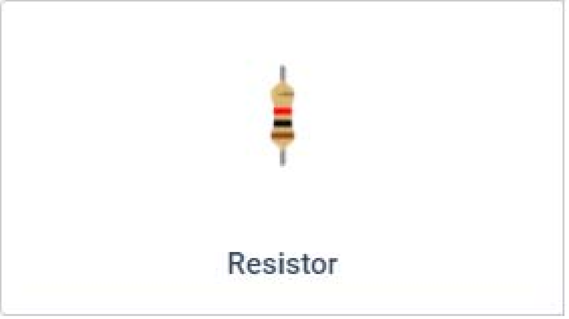
教學網站：HTTP://120.108.221.55/PROFCHWU/COMPUTER

2017/09/18

線上學習Arduino
www.thinkcard.com

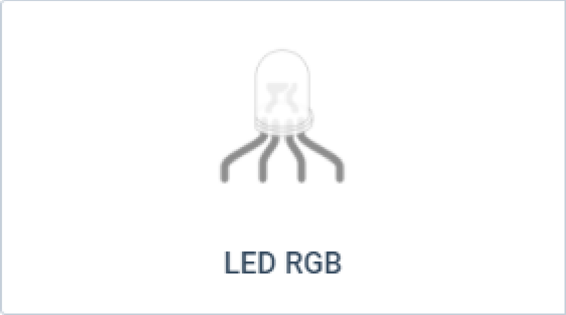


設計一個光敏電阻控制的LED燈
拉出所需元件



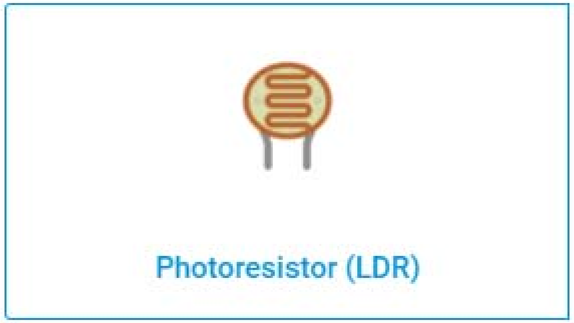
Resistor

電阻



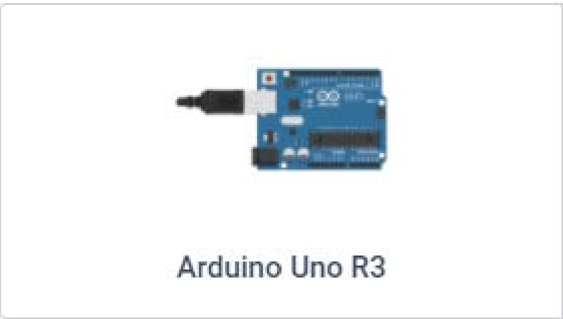
LED RGB

RGB LED燈



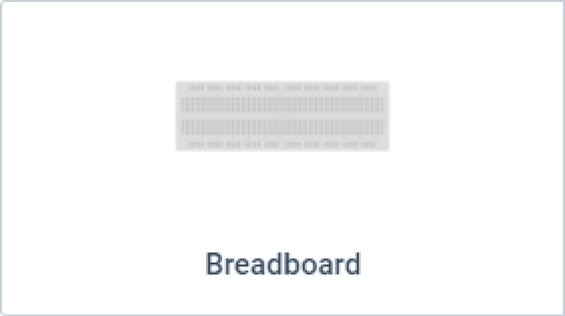
Photoresistor (LDR)

光敏電阻



Arduino Uno R3

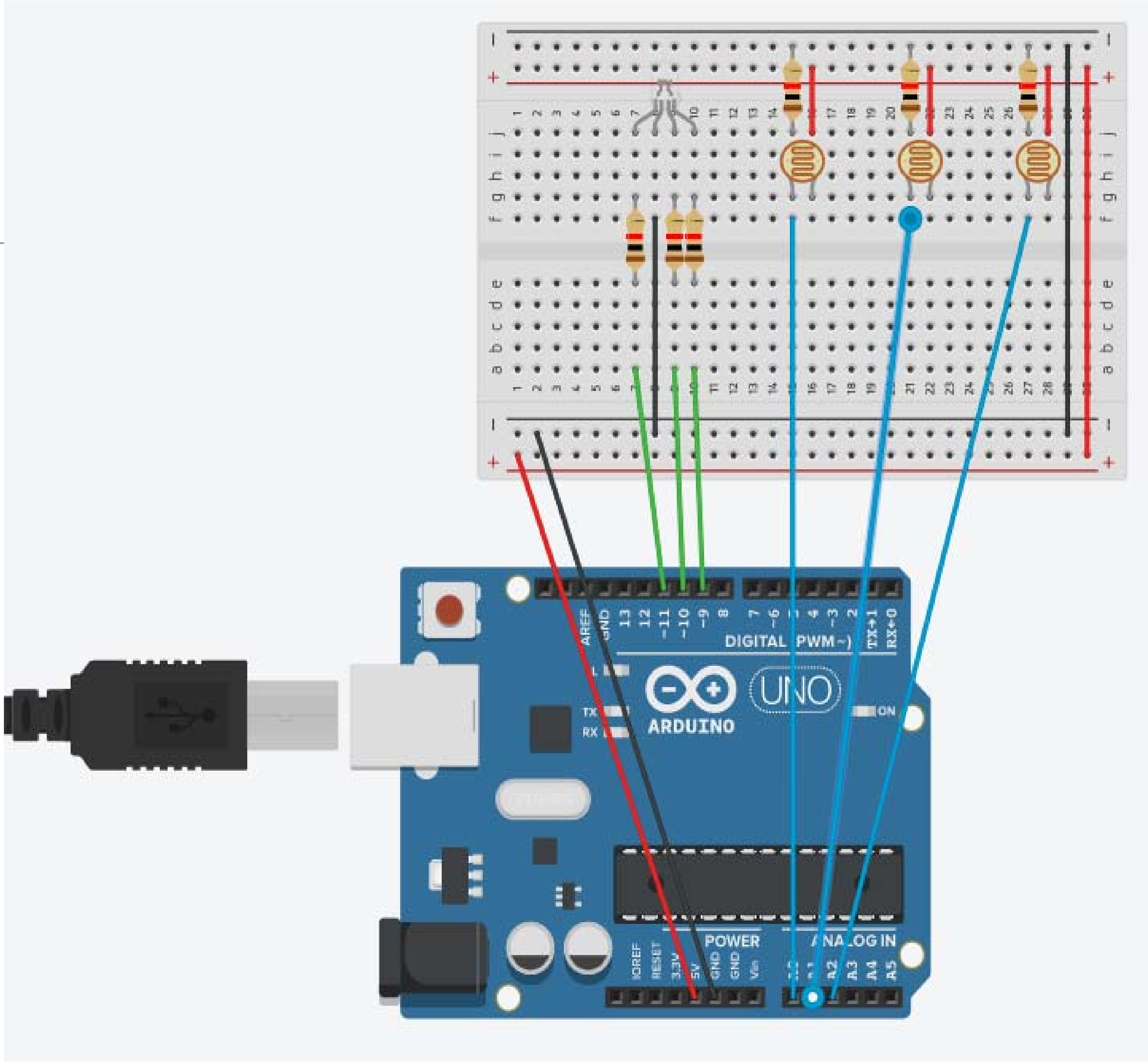
Arduino主機板



Breadboard

麵包板

電路圖


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
UNO
ARDUINO
TX
RX
L
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN

設定輸入與輸出腳位

```
const int greenLEDPin = 9;             //定義綠色的腳位為9
const int blueLEDPin = 10;             //定義藍色的腳位為10
const int redLEDPin = 11;              //定義紅色的腳位為11

const int redSensorPin = A0;           //定義紅色光敏電阻的腳位為A0
const int greenSensorPin = A1;         //定義綠色光敏電阻的腳位為A1
const int blueSensorPin = A2;          //定義藍色光敏電阻的腳位為A2
```

注意！ 程式設定的腳位數值需要與你的電路圖接腳一致。不然程式無法運作。

設定RGB與RGB sensor的初始數值

```
int redValue = 0;

int greenValue = 0;

int blueValue = 0;


int redSensorValue = 0;

int greenSensorValue = 0;

int blueSensorValue=0;
```

設定RGB與RGB sensor的初始數值

```
int redValue = 0;

int greenValue = 0;

int blueValue = 0;


int redSensorValue = 0;

int greenSensorValue = 0;

int blueSensorValue=0;
```

void setup()為開始設定與Arduino做序列通訊，通訊速度9600 bits per second

```
void setup(){
 Serial.begin(9600);
 pinMode(greenLEDPin, OUTPUT);
 pinMode(redLEDPin, OUTPUT);
 pinMode(blueLEDPin, OUTPUT);
}
```

以9600 bps速度與Arduino通訊

定義三個顏色輸出的腳位

In the loop() function, read the sensor values on A0, A1, and A2 with analogRead() and store the values in the appropriate variables.

```
void loop(){
        redSensorValue = analogRead(redSensorPin);
        delay(5);
        greenSensorValue = analogRead(greenSensorPin);
        delay(5);
        blueSensorValue = analogRead(blueSensorPin);
        }
```

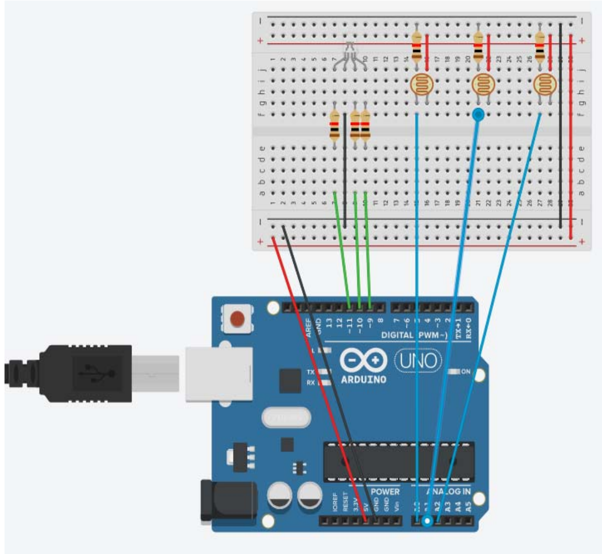
A0

A1

A2

讀取腳位的感測器數值

設定延遲 5 milliseconds －以便能等待接收值



Analog sensor readings always take on values between 0 and 1023. Later, you will convert this to a more useful value.

Next print out the sensor values on one line. The “\t” is the equivalent of pressing the “tab” key on the keyboard.

```
//report raw data to serial port
Serial.print("Raw SensorValues \t Red: ");
Serial.print(redSensorValue);
Serial.print("\t Green: ");
Serial.print(greenSensorValue);
Serial.print("\t Blue: ");
Serial.println(blueSensorValue);
```

將讀取到的感測器數值（光敏電阻）印出

螢幕上顯示會像這樣

```
Raw Sensor Values Red:1000 Green: 560 Blue:400
```

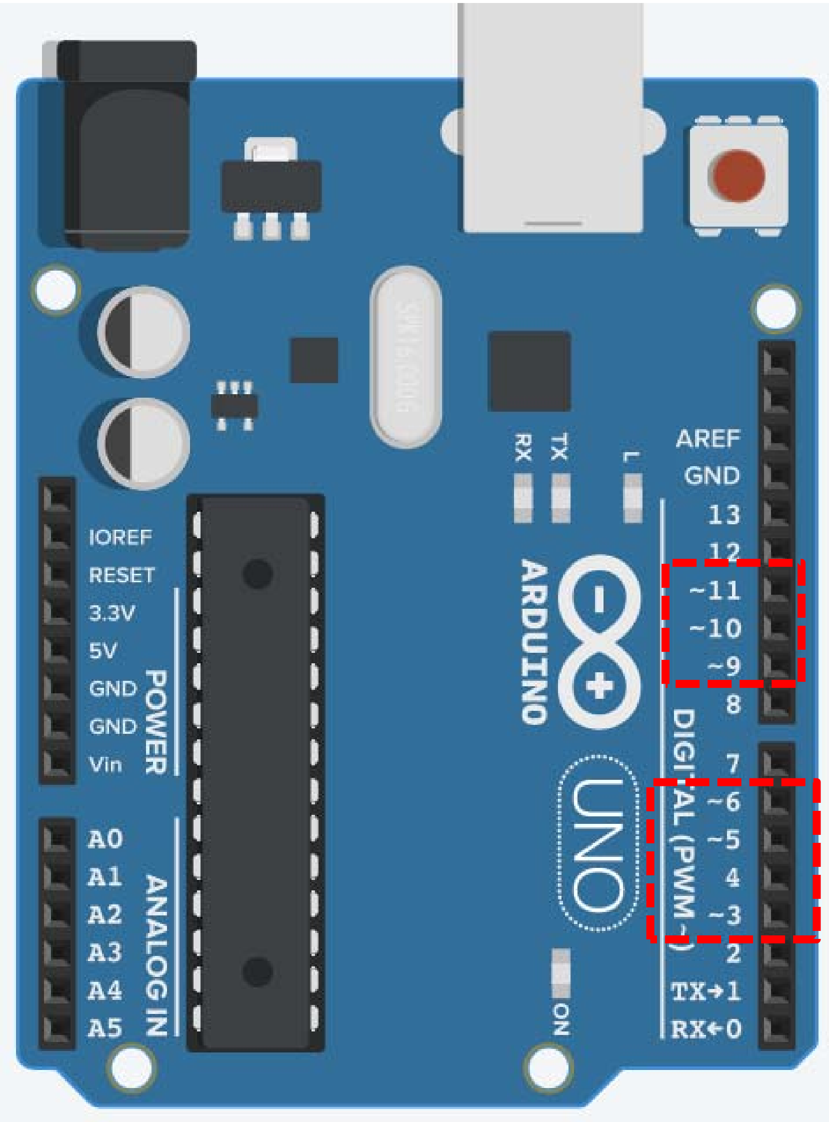
In the serial monitor, your data will look like this:
Raw SensorValues Red: 1000 Green: 560 Blue: 400

Arduino PWM腳位

電壓輸出控制其實不複雜。

Arduino提供PWM輸出，分別是pin 3,5,6,9,10,11 ，就是板子上有~符號的，都能用PWM輸出。

美中不足的是只有8bits，電壓是0-5V，從0-255，每增加1，電壓增加0.019607843 V。



The function to change the LED’s brightness via PWM is called analogWrite(). analogWrite() needs two arguments: the pin to write to, and a value between 0-255.

```
//convert sensor values to PWM duty cycle #

redValue = redSensorValue/4;

blueValue = blueSensorValue/4;

greenValue = greenSensorValue/4;
```

光敏電阻的值為０－１０２３

ＰＷＭ的值為０－２５５

0 :暗
127: 50%亮
255: 100%亮

所以把光敏電阻的值/4 可以轉換成PWM腳位所需要的電壓數值

A value of 255 will set the pin HIGH 100% of the time, making the attached LED as bright as it can be.

A value of 127 will set the pin HIGH for 50% of the period, making the LED dimmer.

A value of 0 would set the pin LOW 100% of the time, turning the LED off.

To convert the sensor reading from a value between 0-1023 to a value between 0-255 for analogWrite(), simply divide the sensor readings by 4.

Next, print out the new mapped values on their own line.

```
//report PWM values to serial port
Serial.print("Mapped SensorValues \t Red: ");
Serial.print(redValue);
Serial.print("\t Blue: ");
Serial.print(blueValue);
Serial.print("\t Green: ");
Serial.println(greenValue);
```

螢幕上顯示會像這樣

```
Raw Sensor Values Red:1000 Green: 560 Blue:400 Mapped SensorValues Red: 250 Green:140 Blue 100
```

Along with the previous information sent through the serial port, the information in the serial monitor will look like this (for example):

```
Raw SensorValues Red: 1000 Green: 560 Blue: 400 Mapped SensorValues Red: 250 Green:140 Blue: 100
```

Finally, use the converted sensor data to set the brightness of the LED.

```
analogWrite(redLEDPin, redValue);
```

將redValue數值寫入redLEDPin腳位（控制紅色是否發亮）

```
analogWrite(greenLEDPin, greenValue);
```

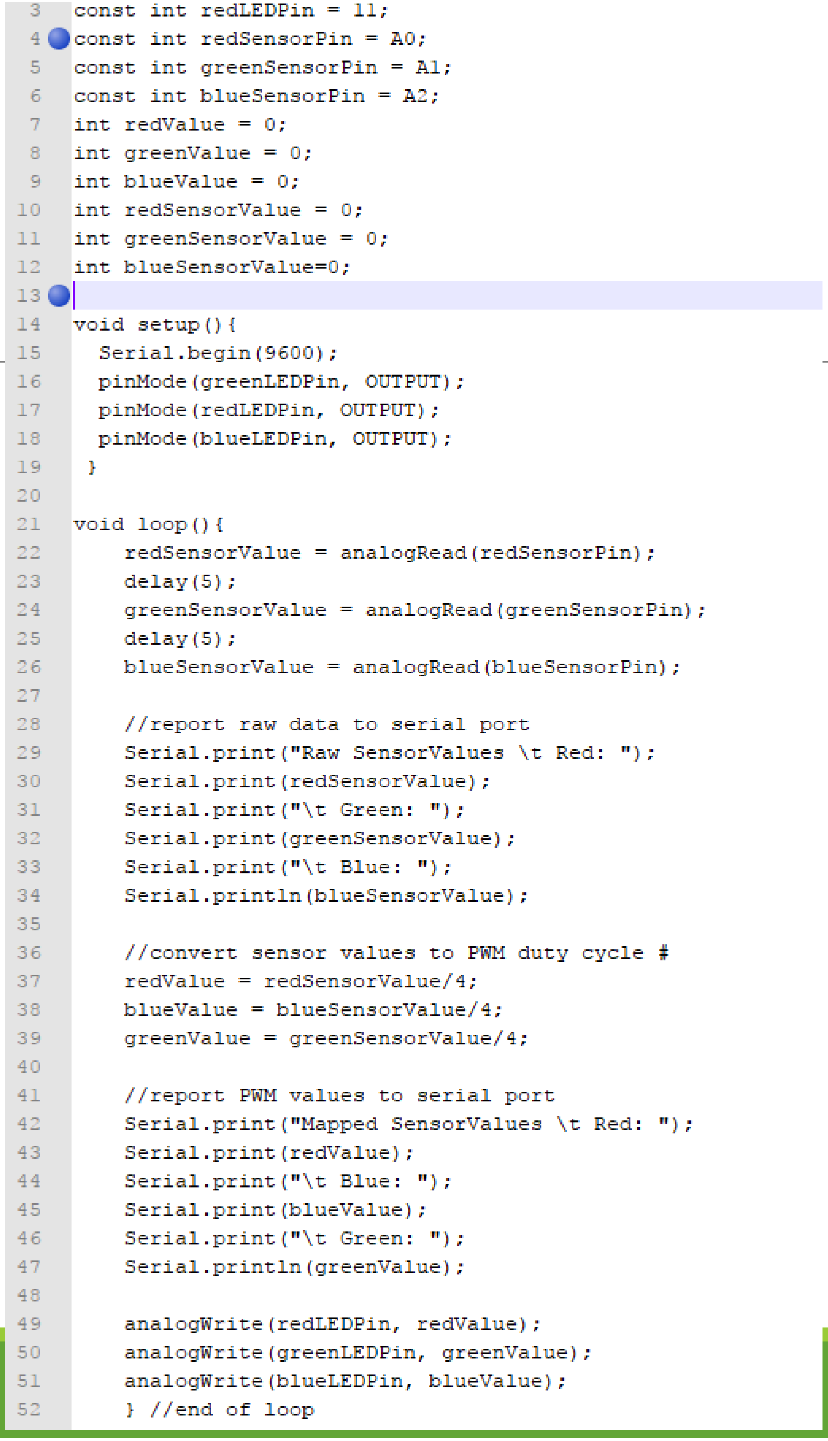
將greeValue數值寫入greenLEDPin腳位（控制綠色是否發亮）

```
analogWrite(blueLEDPin, blueValue);
} //end of loop
```

將blueValue數值寫入blueLEDPin腳位（控制藍色是否發亮）

You will put the analogWrite() function into practice! Remember, analogWrite() needs two arguments: the pin to write to, and a value between 0-255 (here given by the variable redValue, etc).

全部的程式



```
const int redLEDPin = 11;
const int redSensorPin = A0;
const int greenSensorPin = A1;
const int blueSensorPin = A2;
int redValue = 0;
int greenValue = 0;
int blueValue = 0;
int redSensorValue = 0;
int greenSensorValue = 0;
int blueSensorValue=0;

void setup(){
  Serial.begin(9600);
  pinMode(greenLEDPin, OUTPUT);
  pinMode(redLEDPin, OUTPUT);
  pinMode(blueLEDPin, OUTPUT);
 }

void loop(){
    redSensorValue = analogRead(redSensorPin);
    delay(5);
    greenSensorValue = analogRead(greenSensorPin);
    delay(5);
    blueSensorValue = analogRead(blueSensorPin);

    //report raw data to serial port
    Serial.print("Raw SensorValues \t Red: ");
    Serial.print(redSensorValue);
    Serial.print("\t Green: ");
    Serial.print(greenSensorValue);
    Serial.print("\t Blue: ");
    Serial.println(blueSensorValue);

    //convert sensor values to PWM duty cycle #
    redValue = redSensorValue/4;
    blueValue = blueSensorValue/4;
    greenValue = greenSensorValue/4;

    //report PWM values to serial port
    Serial.print("Mapped SensorValues \t Red: ");
    Serial.print(redValue);
    Serial.print("\t Blue: ");
    Serial.print(blueValue);
    Serial.print("\t Green: ");
    Serial.println(greenValue);

    analogWrite(redLEDPin, redValue);
    analogWrite(greenLEDPin, greenValue);
    analogWrite(blueLEDPin, blueValue);
    } //end of loop
```