微控制器編程課程（基礎）課程
智能亮度調節枱燈
編程範例

```
int light_sensor = A0;    //設定 light sensor 於 arduino 的位置
int light_value;                //建立一個盛載光感應器數值的變數
int light_value_map;      //建立一個盛載光感應器數值(改變比例)的變數
int Blue_light = 4;      //設定藍燈於 arduino 的位置
int Green_light = 5;    //設定綠燈於 arduino 的位置
int Red_light = 6;        //設定紅燈於 arduino 的位置
int led = 3;                    //設定 led 於 arduino 的位置

void setup() {
   Serial.begin(9600);
   pinMode(light_sensor, INPUT);    //設定光感應器是輸入裝置
   pinMode(Red_light, OUTPUT);        //設定 RGB 是輸出置
   pinMode(Blue_light, OUTPUT);      //設定 RGB 是輸出裝置
   pinMode(Green_light, OUTPUT);    //設定 RGB 是輸出裝置
   pinMode(led, OUTPUT);                    //設定 LED 是輸出裝置
}

void loop() {
   light_value = analogRead(light_sensor);    //以 analog 模式讀取 light sensor 數據並載入數據 light_value 變數中
   light_value_map = map(light_value, 1, 1023, 255, 1); //將 light_value 數值由 1-1023 反比地下降至 255-1
   analogWrite(Red_light, light_value_map);          //以 light_value_map 數值控制亮度
   analogWrite(Green_light, light_value_map);      //以 light_value_map 數值控制亮度
   analogWrite(Blue_light, light_value_map);        //以 light_value_map 數值控制亮度
   analogWrite(led, light_value_map);
   if (light_value >= 700) {
      analogWrite(led, 0);
   }
   Serial.print("light_value = ");
   Serial.println(light_value);
```

```
    Serial.print("mapped = ");
    Serial.println(light_value_map);                          //  在監視器可視化
light_value_map 數值
    delay(50);
}
```

微控制器編程課程（基礎）課程

溫濕度感應器

編程範例

```
#include <DFRobot_DHT11.h>
#include <SoftwareSerial.h>
SoftwareSerial sSerial(3, 2);
DFRobot_DHT11 DHT;
int tempSensor = 13;
int light_sensor = ;
int light_value;
int tempValue;
int temperature;
int humidity;

void setup() {
   sSerial.begin (57600);
   pinMode(light_sensor,);
}

void loop()
{
   DHT.read(tempSensor);
   temperature = DHT.temperature;
   humidity = DHT.humidity;
   light_value = analogRead(light_sensor);
   sSerial.print();
   sSerial.print();
   delay(2000);
   sSerial.print(temperature);
   sSerial.print("t");
   delay(2000);
   sSerial.print(humidity);
   sSerial.print("H");
   delay(2000);
}
```

微控制器編程課程（基礎）課程
洗手倒數機械人
編程範例

```
#include <Servo.h>
Servo myServo;
// Define the pins for the ultrasonic sensor and LED
const int echoPin = 9;
const int trigPin = 10;
const int ledPin = 13;

// Define variables to keep track of time and distance
long duration;
int distance;

// Define a boolean variable to keep track of whether an object has been detected
and whether the countdown is in progress
bool objectDetected = false;
bool countdownInProgress = false;

void setup() {
  // Set the ultrasonic sensor and LED pins as inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  myServo.attach(3);
  myServo.write(1);

  // Begin serial communication for debugging purposes
  Serial.begin(9600);
}

void loop() {
  // Send a pulse to the ultrasonic sensor to initiate a reading
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Measure the time it takes for the pulse to return to the sensor
  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance in centimeters
  distance = duration * 0.034 / 2;

  // Print the distance for debugging purposes
  Serial.print("Distance: ");
  Serial.println(distance);

  // Check if an object is detected
  if (distance < 10) {
    if (!objectDetected) {
      // If an object has just been detected, set the boolean to true and start a
countdown of 5 seconds
      objectDetected = true;
      int countdown = 5;
      while (countdown > 0) {
        // Print the countdown for debugging purposes
        Serial.print("Countdown: ");
        Serial.println(countdown);

        // Wait for 1 second
        delay(1000);

        // Measure the distance again to check if the object has left
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
        duration = pulseIn(echoPin, HIGH);
        distance = duration * 0.034 / 2;

        if (distance >= 10) {
```

```
        // If the object has left, stop the countdown and set the boolean to false
        countdownInProgress = false;
        objectDetected = false;
        break;
      }

      // Decrement the countdown
      countdown--;
    }

    if (countdown == 0) {
      // If the countdown was successful, set the boolean to true and start a new
countdown of 20 seconds
      countdownInProgress = true;
      countdown = 20;
      while (countdown > 0) {
        // Print the countdown for debugging purposes
        Serial.print("Countdown: ");
        Serial.println(countdown);

        // Wait for 1 second
        delay(1000);

        // Turn on the LED and servo to 179
        myServo.write(179);
        digitalWrite(ledPin, HIGH);
        Serial.print("light on ");

        // Decrement the countdown
        countdown--;
      }

      // Set the boolean back to false turn off the LED and servo to 1
      countdownInProgress = false;
      myServo.write(1);
      digitalWrite(ledPin, LOW);
      Serial.print("light off ");
    }
```

```
        }
    } else {
        // If no object is detected, set the boolean to false
        objectDetected = false;

        // If the countdown was in progress, stop it and turn off the LED
        if (countdownInProgress) {
            countdownInProgress = false;
            digitalWrite(ledPin, LOW);
            Serial.print("light off");
        }
    }
}
```