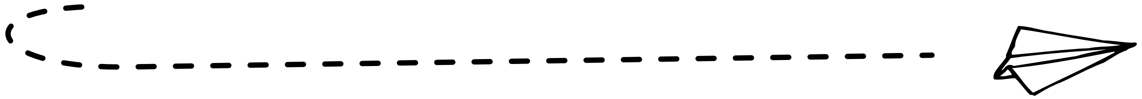
# 物聯網



第21組
B0844128鄭郁珊
B0844143陳亭妤

# 目錄

1.動 機

2.構 想

3.設備&流程圖

4.Arduino程式

5.程式方塊

6.操作影片

# 01

# 動機

安裝至門口玄關
回家開門馬上自動亮燈

安裝至廁所/廚房
半夜起床走路安全
不必摸黑開燈

安裝至走廊通道
人來即亮，人走自熄滅!

# 02

# 構想

# 構　想

☝ 偵測到屋內有人 → 判斷白天或晚上 → 打開燈(晚上)

紅外線　　感光　　RGB LED

✌ 偵測溫溼度 → 是否需開電風扇

溫濕　　電風扇

# 構　想

- 顯示溫濕度
- 屋內有幾個人
- 電燈是否打開
- 風扇是否打開

# 03

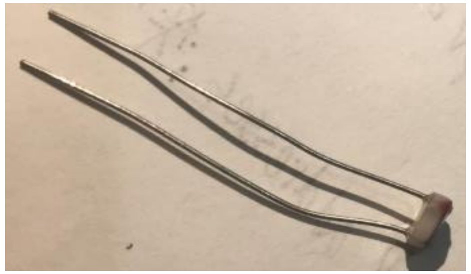
# 使用到的設備

# 使用的設備

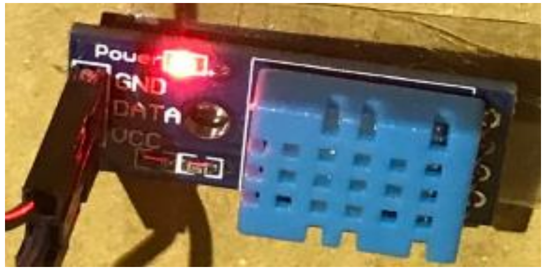
## 紅外線

## 電風扇

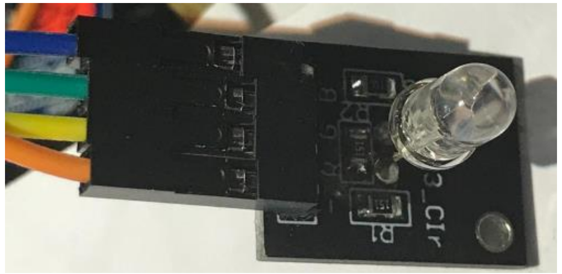
## 光敏電阻

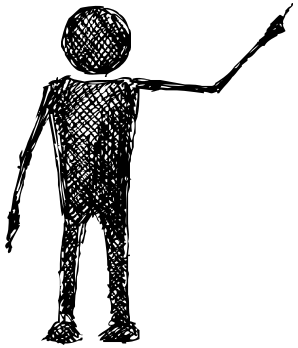


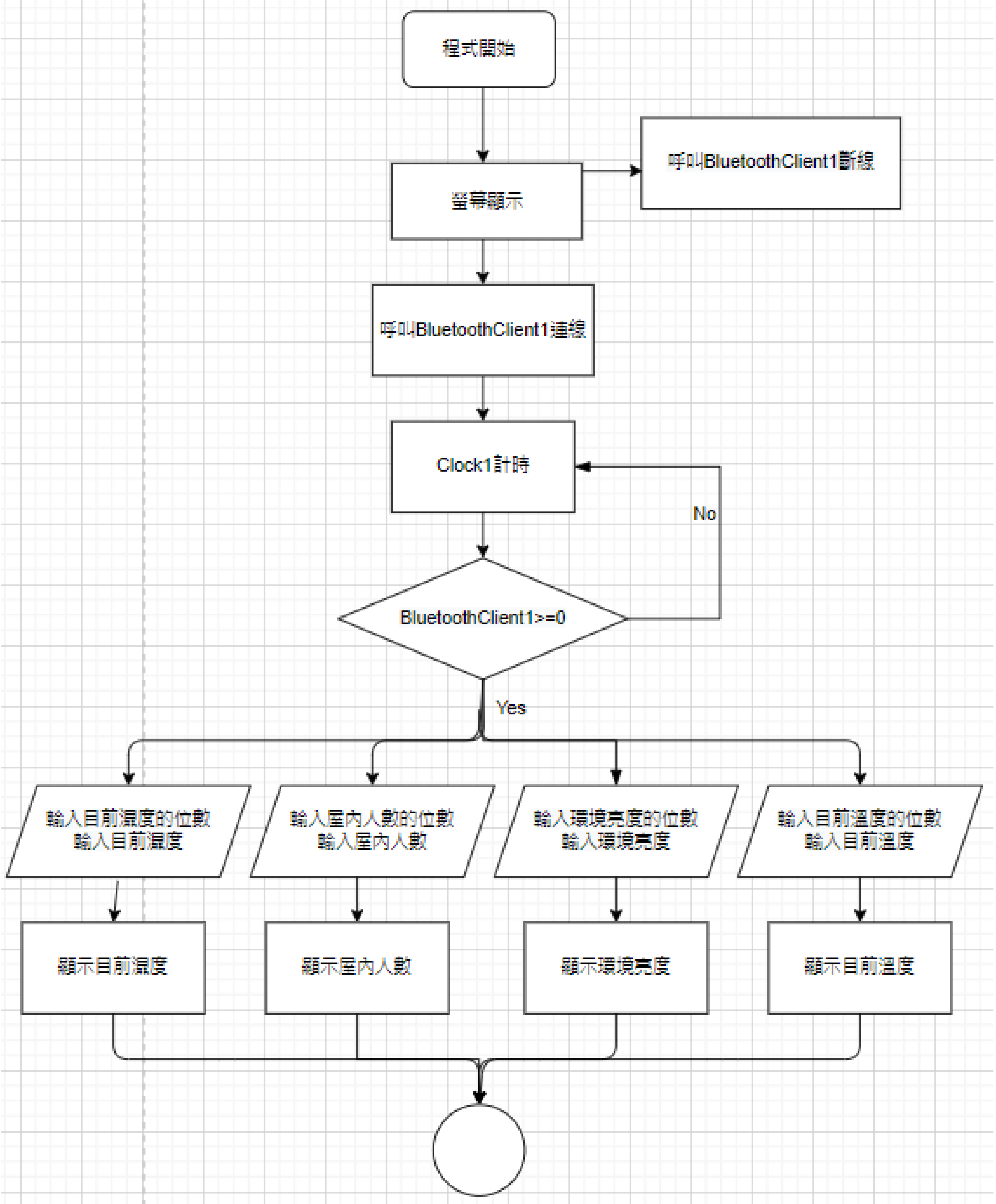
## 溫濕感測器

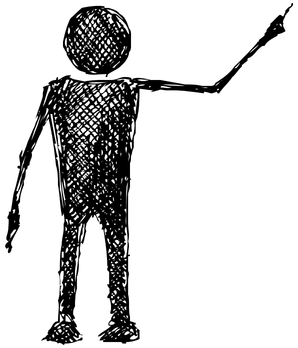


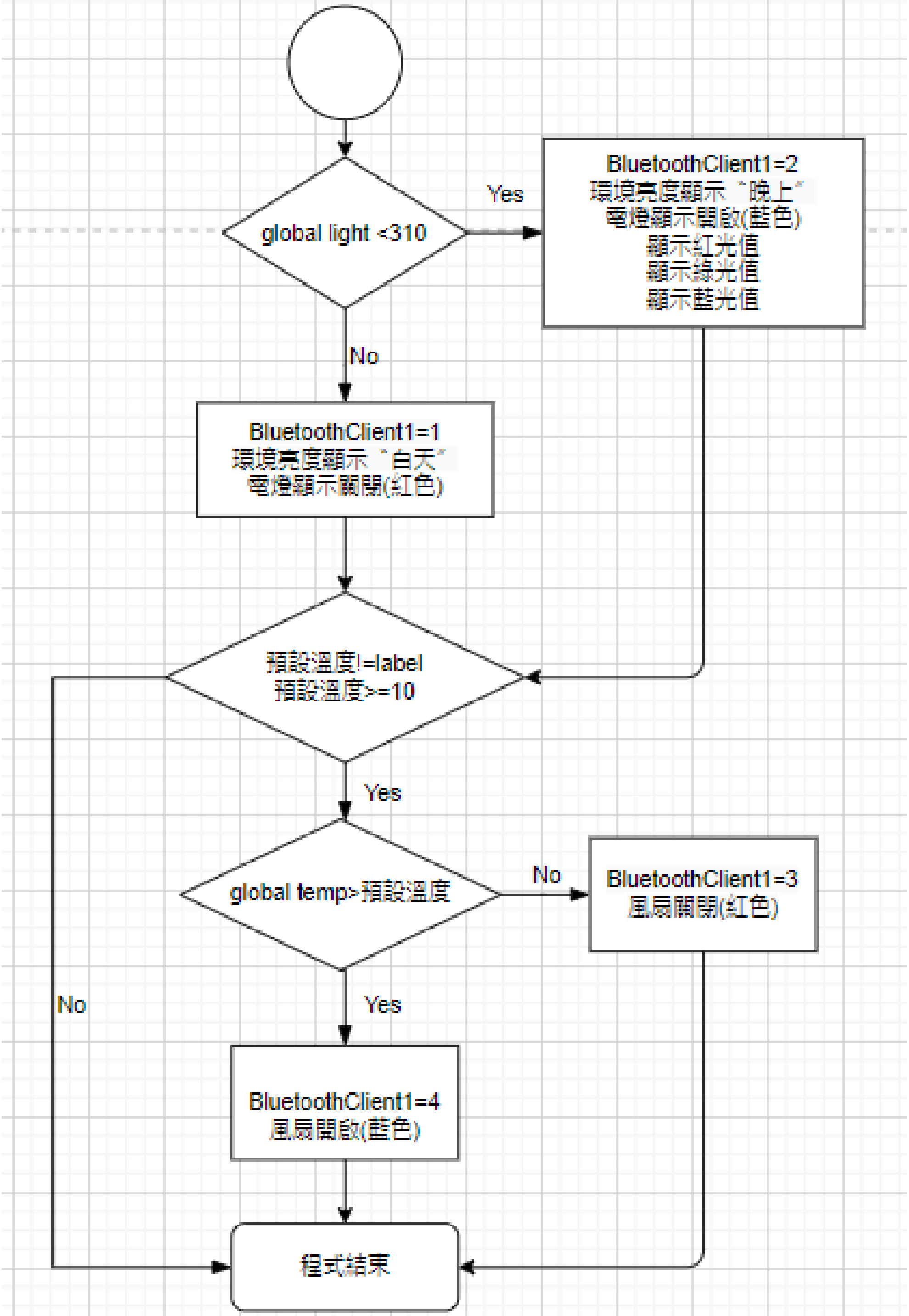
## RGB燈






程式開始
螢幕顯示
呼叫BluetoothClient1斷線
呼叫BluetoothClient1連線
Clock1計時
BluetoothClient1>=0
No
Yes
輸入目前濕度的位數
輸入目前濕度
輸入屋內人數的位數
輸入屋內人數
輸入環境亮度的位數
輸入環境亮度
輸入目前溫度的位數
輸入目前溫度
顯示目前濕度
顯示屋內人數
顯示環境亮度
顯示目前溫度





global light <310

Yes

BluetoothClient1=2
環境亮度顯示〝晚上〞
電燈顯示開啟(藍色)
顯示紅光值
顯示綠光值
顯示藍光值

No

BluetoothClient1=1
環境亮度顯示〝白天〞
電燈顯示關閉(紅色)

預設溫度!=label
預設溫度>=10

Yes

No

global temp>預設溫度

No

BluetoothClient1=3
風扇關閉(紅色)

Yes

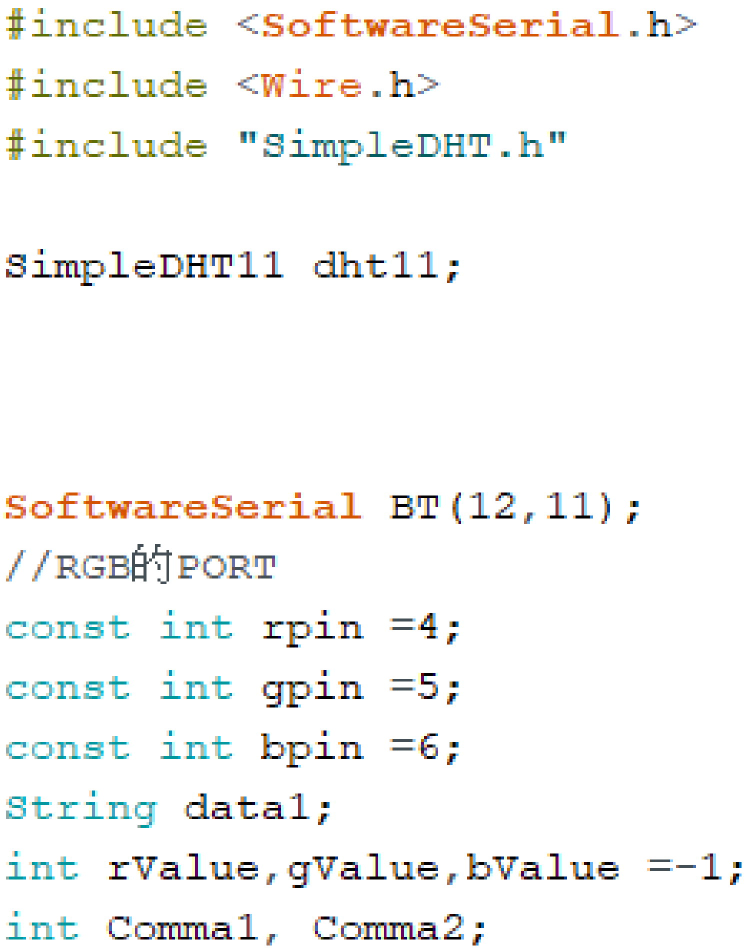
BluetoothClient1=4
風扇開啟(藍色)

程式結束

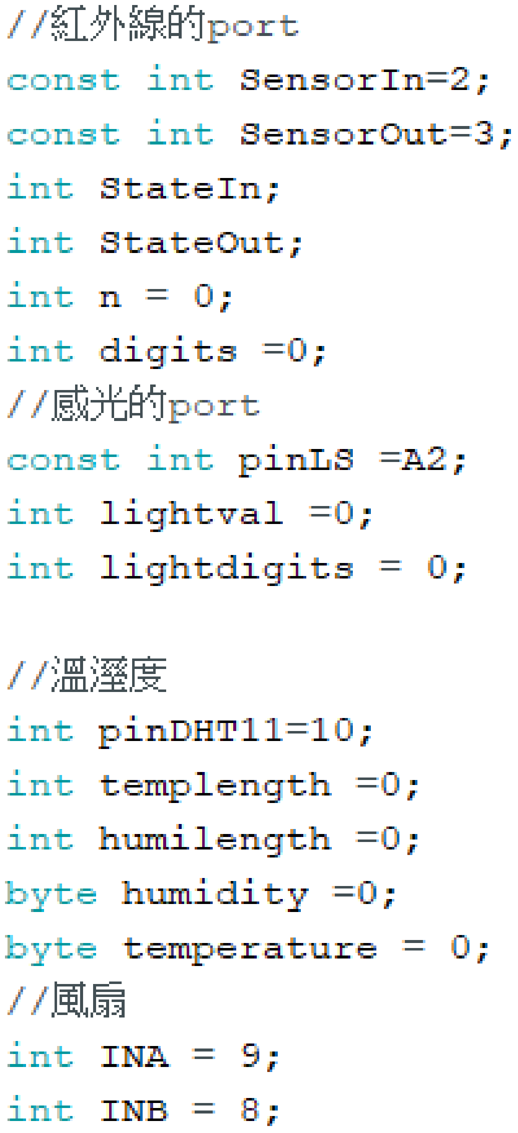
# 04

# Arduino 程式解說

# 全域變數

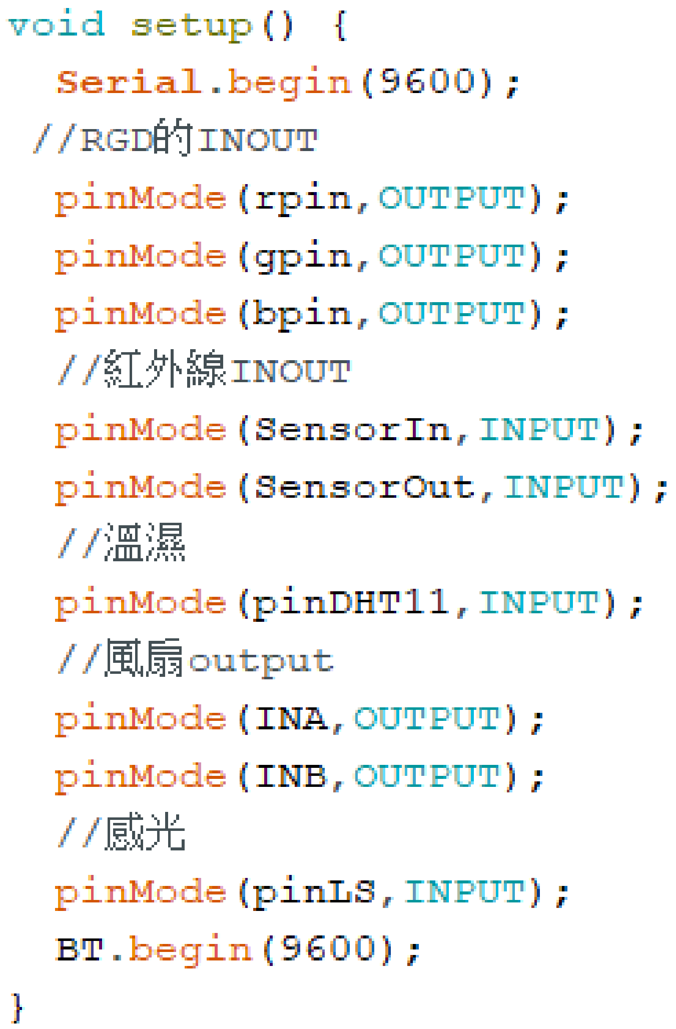


```
#include <SoftwareSerial.h>
#include <Wire.h>
#include "SimpleDHT.h"

SimpleDHT11 dht11;



SoftwareSerial BT(12,11);
//RGB的PORT
const int rpin =4;
const int gpin =5;
const int bpin =6;
String data1;
int rValue,gValue,bValue =-1;
int Comma1, Comma2;
```

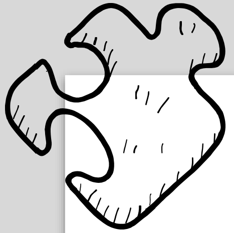


```
//紅外線的port
const int SensorIn=2;
const int SensorOut=3;
int StateIn;
int StateOut;
int n = 0;
int digits =0;
//感光的port
const int pinLS =A2;
int lightval =0;
int lightdigits = 0;

//溫溼度
int pinDHT11=10;
int templength =0;
int humilength =0;
byte humidity =0;
byte temperature = 0;
//風扇
int INA = 9;
int INB = 8;
```

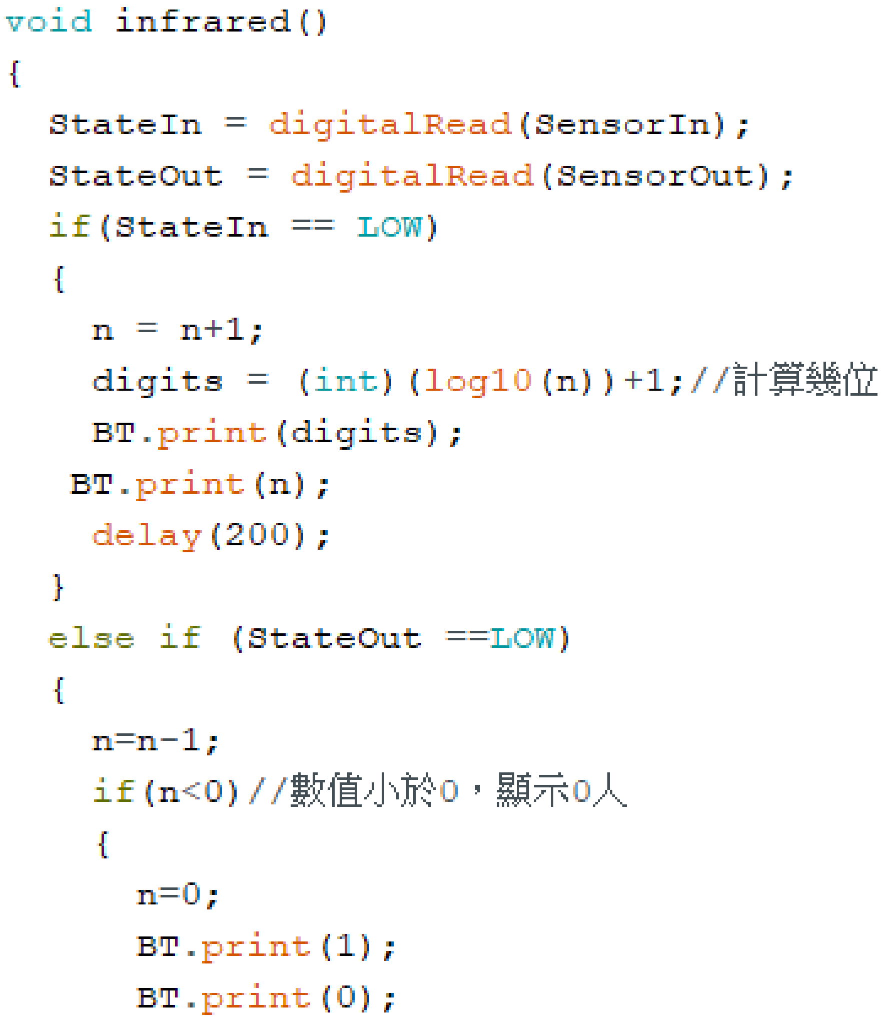
# Void setup()



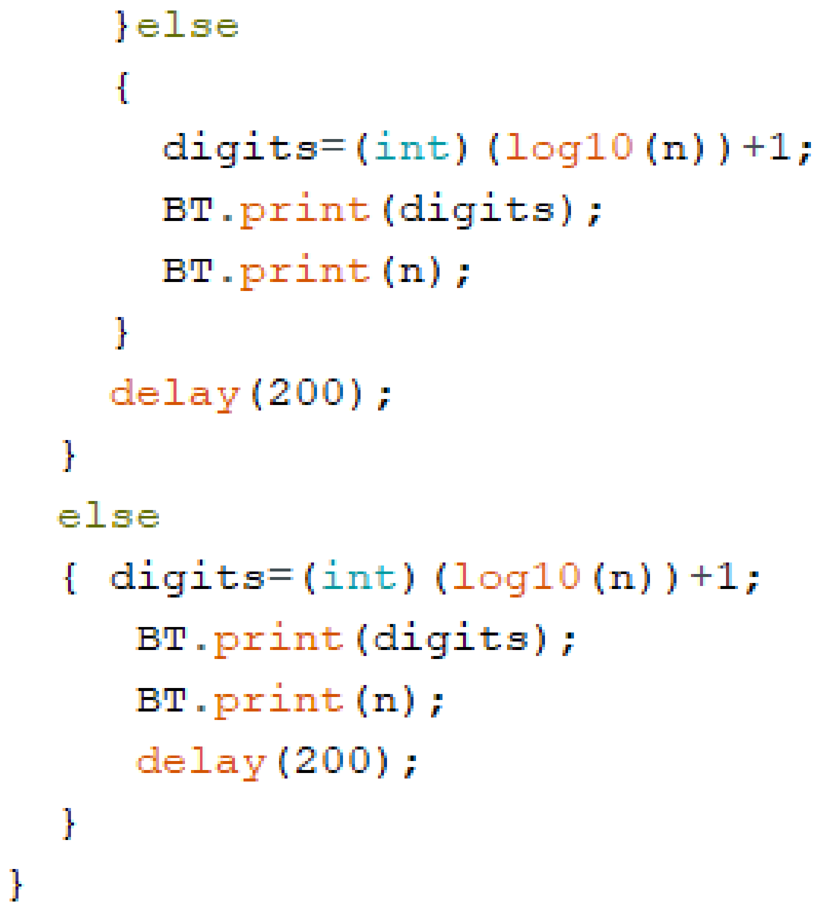
```
void setup() {
  Serial.begin(9600);
 //RGD的INOUT
  pinMode(rpin,OUTPUT);
  pinMode(gpin,OUTPUT);
  pinMode(bpin,OUTPUT);
  //紅外線INOUT
  pinMode(SensorIn,INPUT);
  pinMode(SensorOut,INPUT);
  //溫濕
  pinMode(pinDHT11,INPUT);
  //風扇output
  pinMode(INA,OUTPUT);
  pinMode(INB,OUTPUT);
  //感光
  pinMode(pinLS,INPUT);
  BT.begin(9600);
}
```



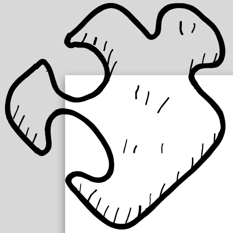
# 紅外線副程式



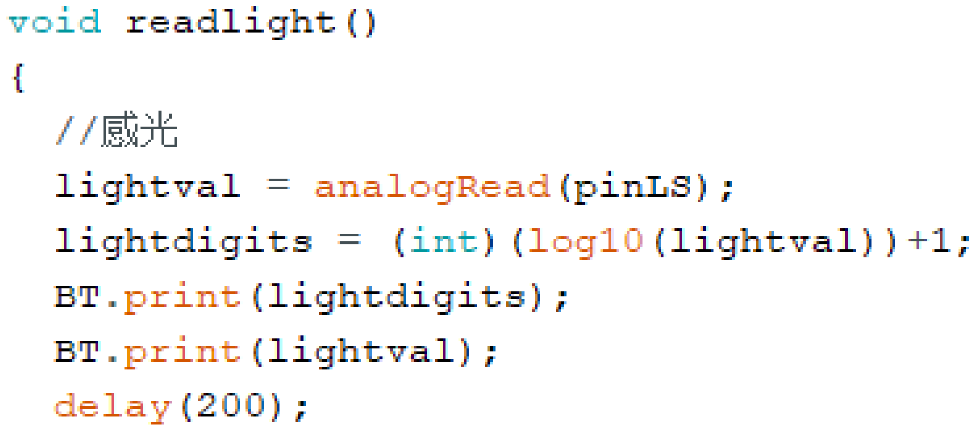
```
void infrared()
{
  StateIn = digitalRead(SensorIn);
  StateOut = digitalRead(SensorOut);
  if(StateIn == LOW)
  {
    n = n+1;
    digits = (int)(log10(n))+1;//計算幾位
    BT.print(digits);
   BT.print(n);
    delay(200);
  }
  else if (StateOut ==LOW)
  {
    n=n-1;
    if(n<0)//數值小於0，顯示0人
    {
      n=0;
      BT.print(1);
      BT.print(0);
```



```
    }else
    {
      digits=(int)(log10(n))+1;
      BT.print(digits);
      BT.print(n);
    }
    delay(200);
  }
  else
  { digits=(int)(log10(n))+1;
     BT.print(digits);
     BT.print(n);
     delay(200);
  }
}
```

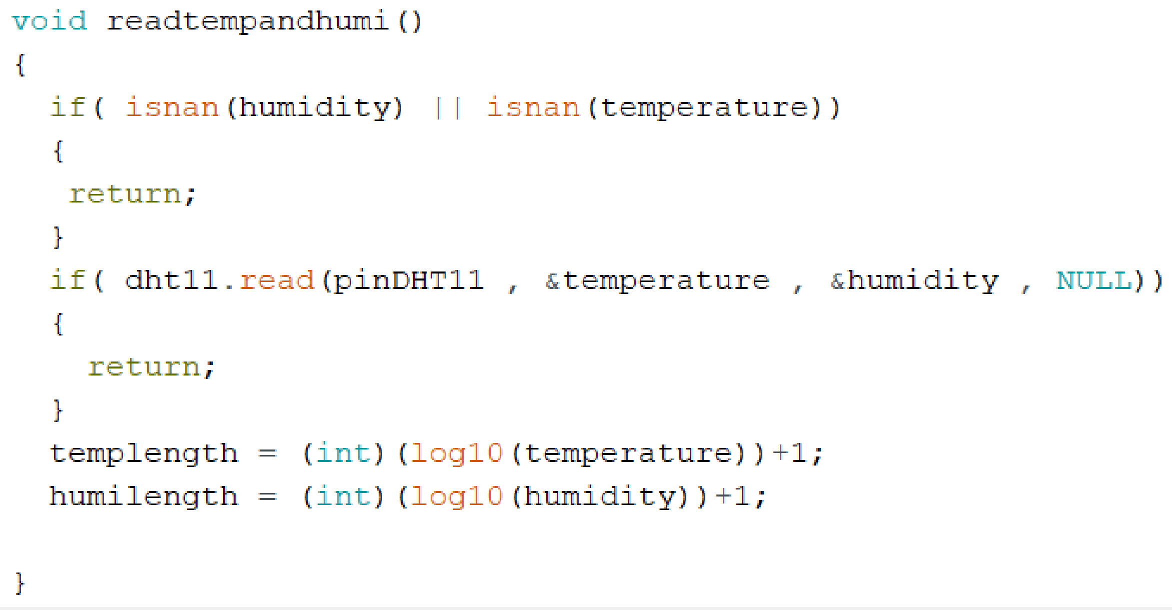


# 感光副程式



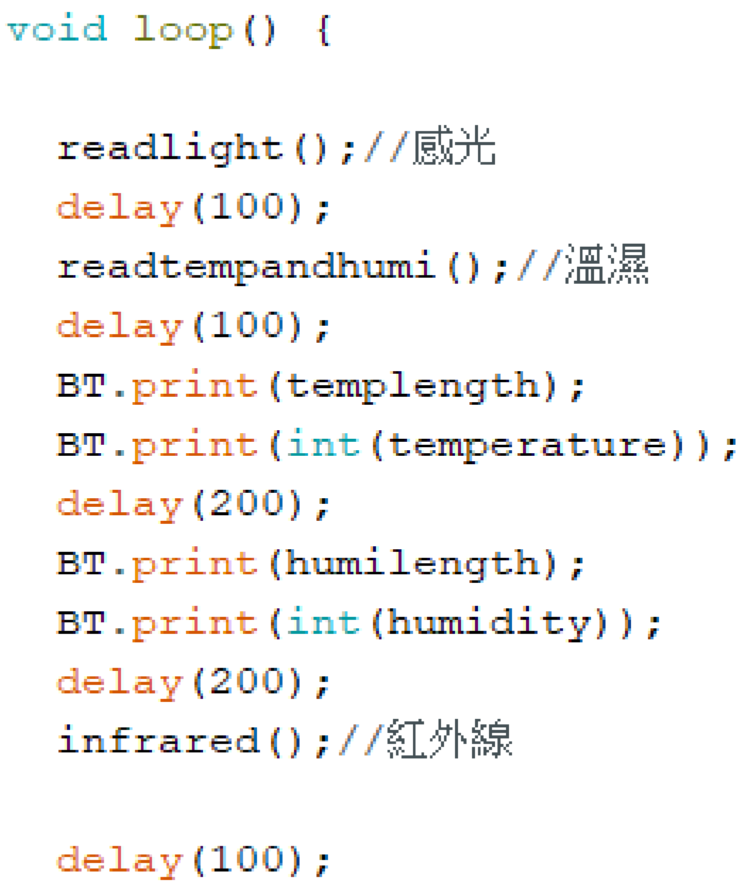
```
void readlight()
{
  //感光
  lightval = analogRead(pinLS);
  lightdigits = (int)(log10(lightval))+1;
  BT.print(lightdigits);
  BT.print(lightval);
  delay(200);
```

# 溫溼度副程式

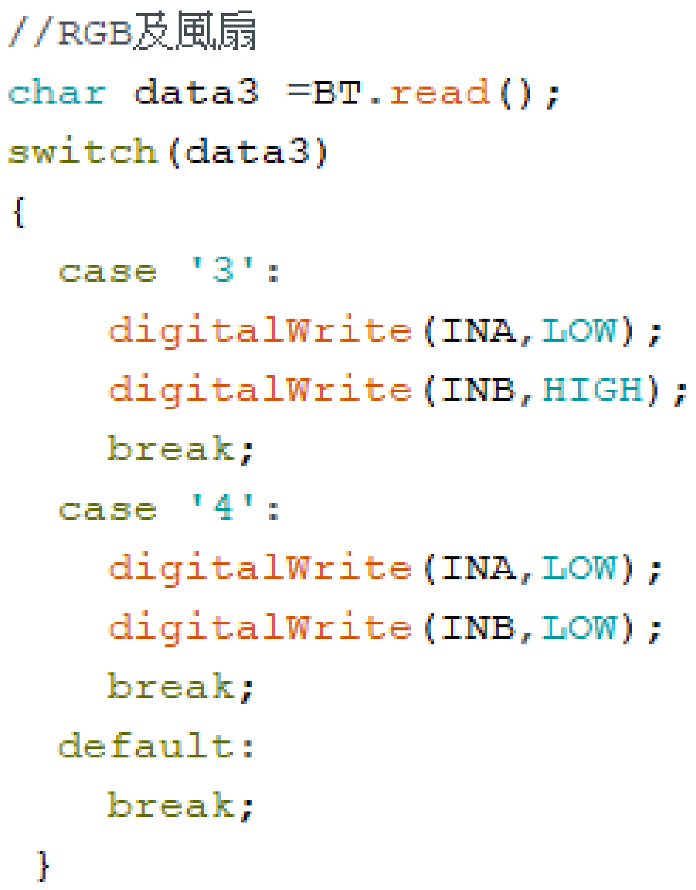


```
void readtempandhumi()
{
  if( isnan(humidity) || isnan(temperature))
  {
   return;
  }
  if( dht11.read(pinDHT11 , &temperature , &humidity , NULL))
  {
    return;
  }
  templength = (int)(log10(temperature))+1;
  humilength = (int)(log10(humidity))+1;

}
```

# 主程式

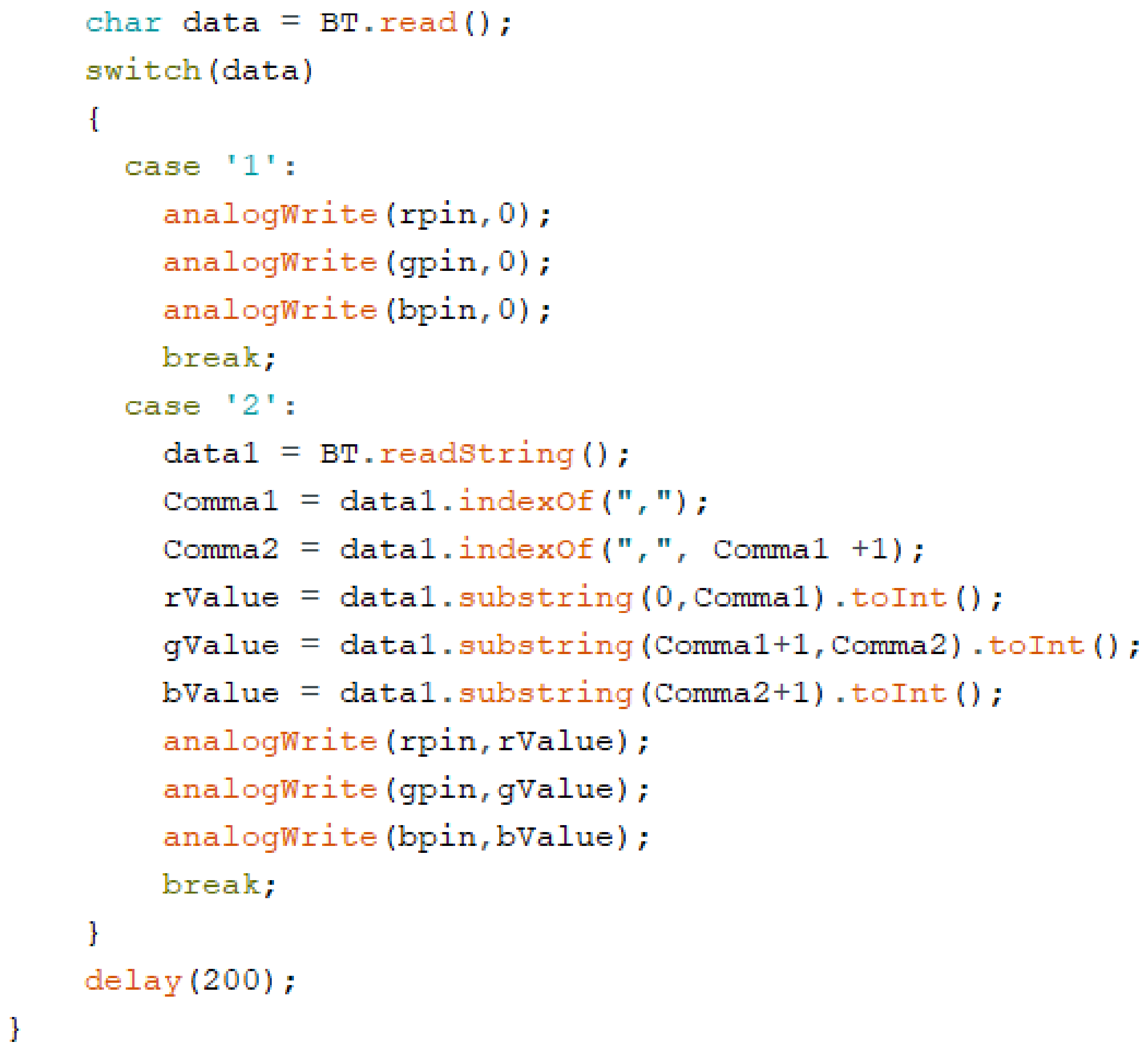


```
void loop() {

  readlight();//感光
  delay(100);
  readtempandhumi();//溫濕
  delay(100);
  BT.print(templength);
  BT.print(int(temperature));
  delay(200);
  BT.print(humilength);
  BT.print(int(humidity));
  delay(200);
  infrared();//紅外線

  delay(100);
```



```
//RGB及風扇
char data3 =BT.read();
switch(data3)
{
  case '3':
    digitalWrite(INA,LOW);
    digitalWrite(INB,HIGH);
    break;
  case '4':
    digitalWrite(INA,LOW);
    digitalWrite(INB,LOW);
    break;
  default:
    break;
 }
```

# 主程式

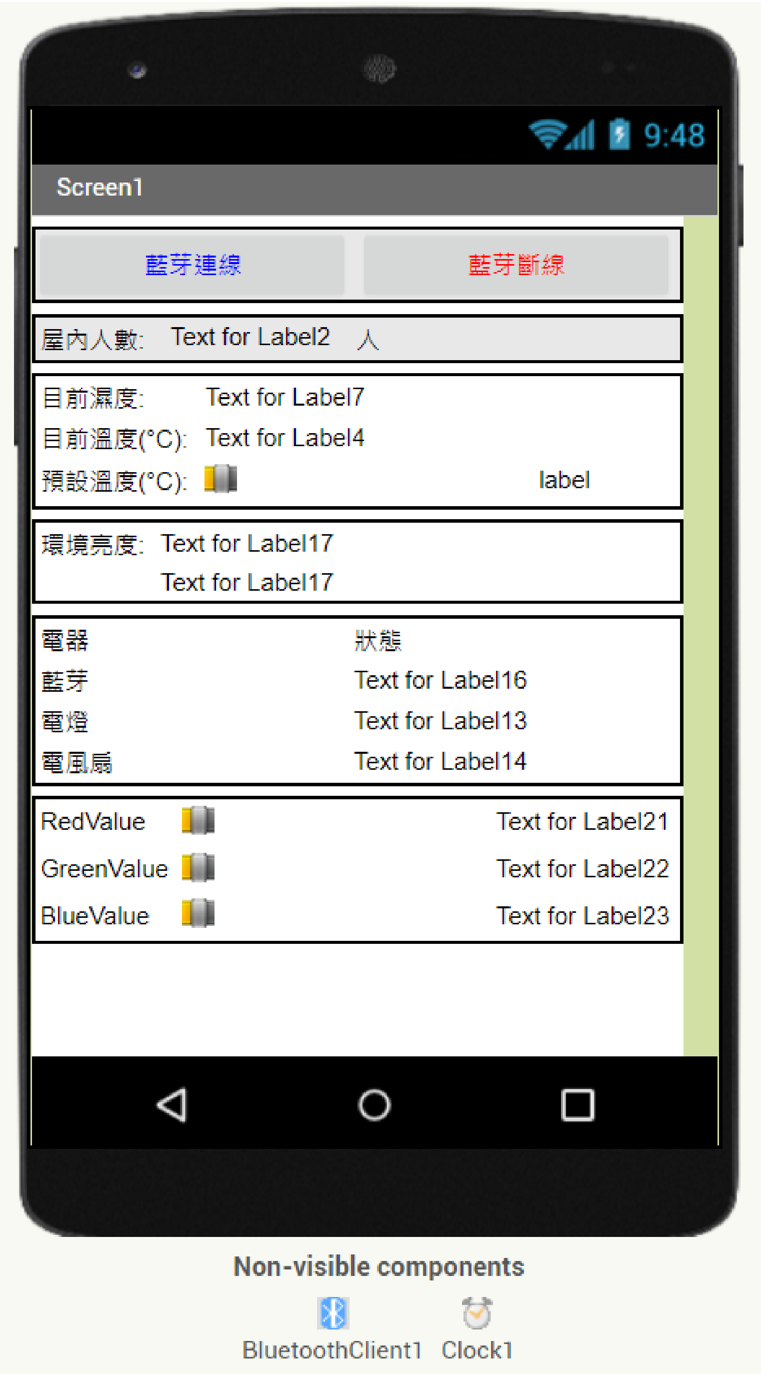


```
    char data = BT.read();
    switch(data)
    {
      case '1':
        analogWrite(rpin,0);
        analogWrite(gpin,0);
        analogWrite(bpin,0);
        break;
      case '2':
        data1 = BT.readString();
        Comma1 = data1.indexOf(",");
        Comma2 = data1.indexOf(",", Comma1 +1);
        rValue = data1.substring(0,Comma1).toInt();
        gValue = data1.substring(Comma1+1,Comma2).toInt();
        bValue = data1.substring(Comma2+1).toInt();
        analogWrite(rpin,rValue);
        analogWrite(gpin,gValue);
        analogWrite(bpin,bValue);
        break;
    }
    delay(200);
}
```

# 05

# 程式拼塊

# 手機操作介面



9:48

Screen1

藍芽連線 藍芽斷線

屋內人數: Text for Label2 人

目前濕度: Text for Label7
目前溫度(°C): Text for Label4
預設溫度(°C): label

環境亮度: Text for Label17
Text for Label17

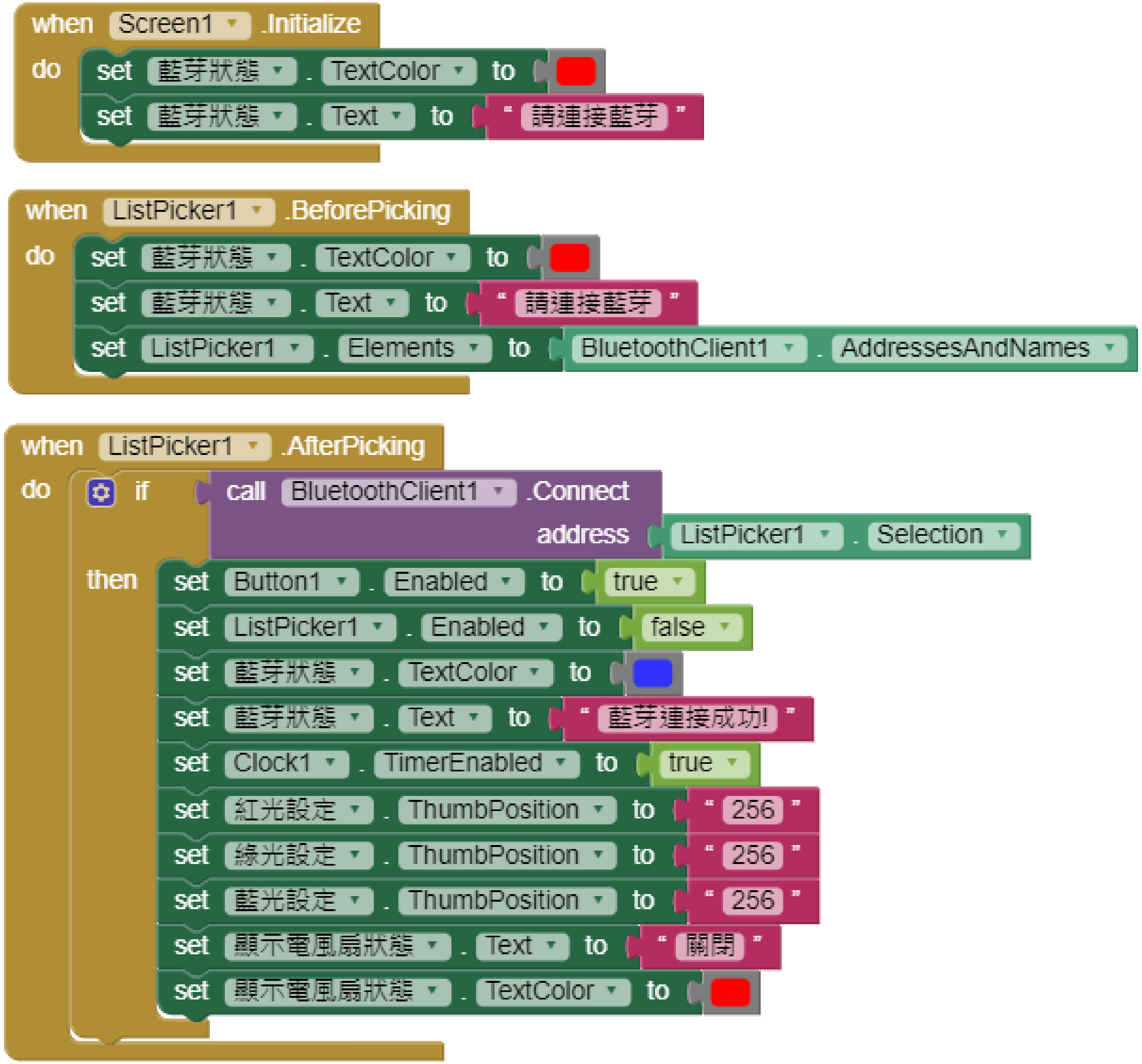
電器 狀態
藍芽 Text for Label16
電燈 Text for Label13
電風扇 Text for Label14

RedValue Text for Label21
GreenValue Text for Label22
BlueValue Text for Label23

Non-visible components

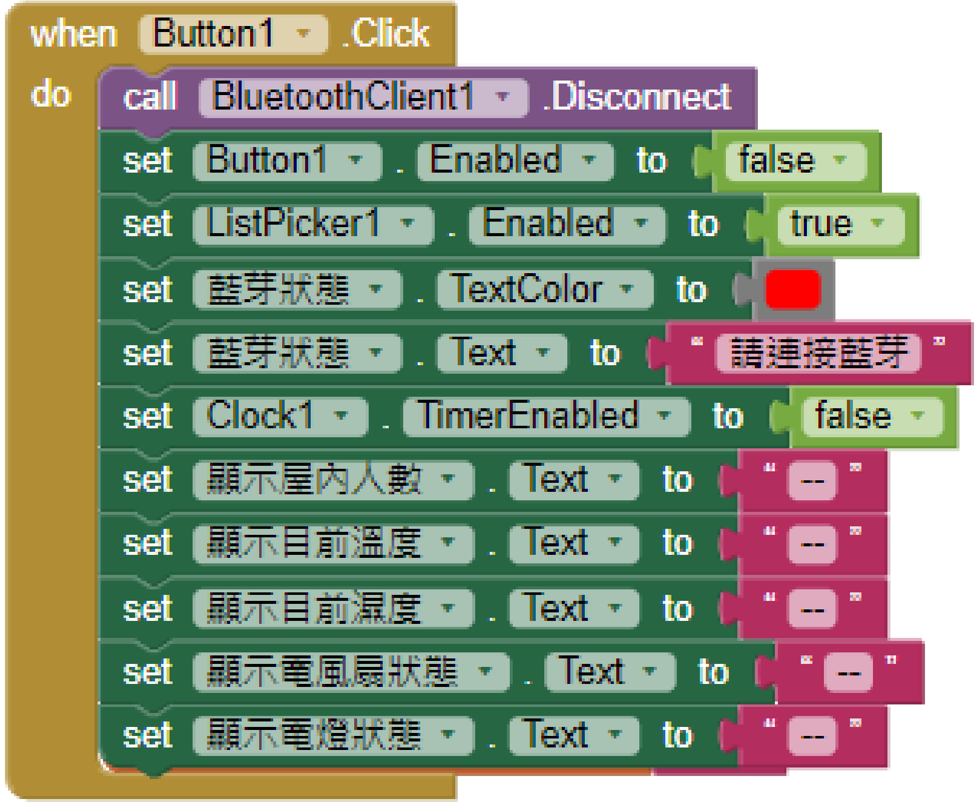
BluetoothClient1 Clock1

# 藍芽按鈕設定



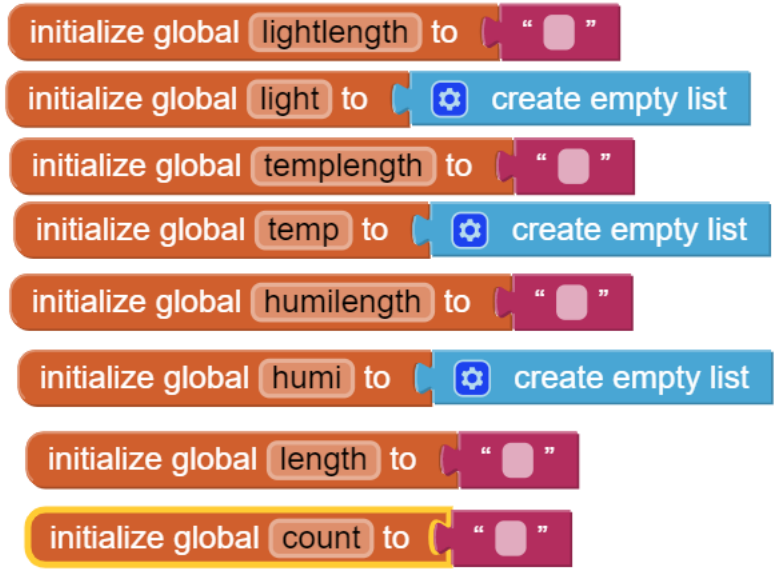
```
when Screen1 .Initialize
do  set 藍芽狀態 . TextColor to 🟥
    set 藍芽狀態 . Text to " 請連接藍芽 "

when ListPicker1 .BeforePicking
do  set 藍芽狀態 . TextColor to 🟥
    set 藍芽狀態 . Text to " 請連接藍芽 "
    set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do  if  call BluetoothClient1 .Connect
                              address ListPicker1 . Selection
    then set Button1 . Enabled to true
         set ListPicker1 . Enabled to false
         set 藍芽狀態 . TextColor to 🟦
         set 藍芽狀態 . Text to " 藍芽連接成功! "
         set Clock1 . TimerEnabled to true
         set 紅光設定 . ThumbPosition to " 256 "
         set 綠光設定 . ThumbPosition to " 256 "
         set 藍光設定 . ThumbPosition to " 256 "
         set 顯示電風扇狀態 . Text to " 關閉 "
         set 顯示電風扇狀態 . TextColor to 🟥
```

藍芽斷線按鈕設定



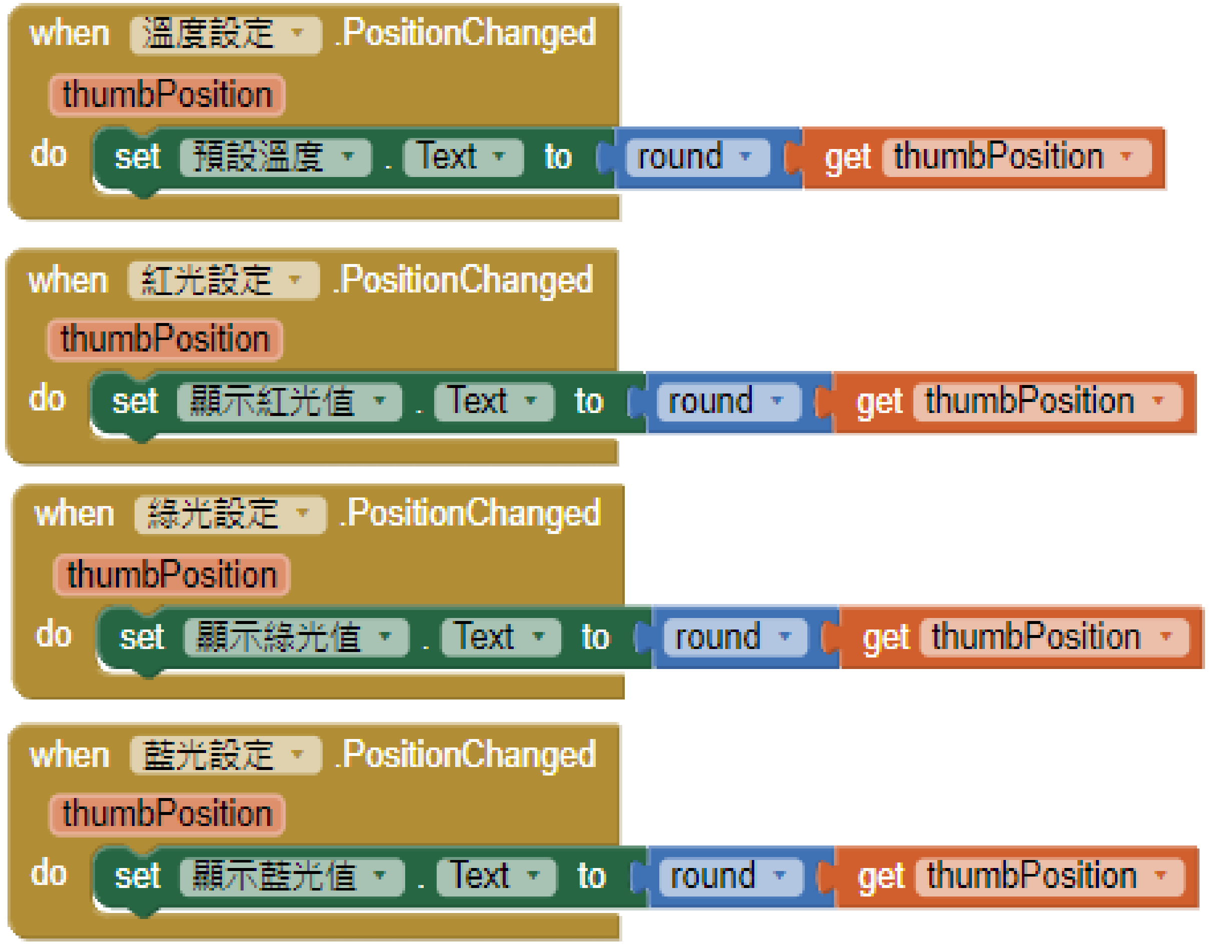
```
when Button1 .Click
do  call BluetoothClient1 .Disconnect
    set Button1 . Enabled to false
    set ListPicker1 . Enabled to true
    set 藍芽狀態 . TextColor to 🟥
    set 藍芽狀態 . Text to " 請連接藍芽 "
    set Clock1 . TimerEnabled to false
    set 顯示屋內人數 . Text to " -- "
    set 顯示目前溫度 . Text to " -- "
    set 顯示目前濕度 . Text to " -- "
    set 顯示電風扇狀態 . Text to " -- "
    set 顯示電燈狀態 . Text to " -- "
```

宣告全域變數



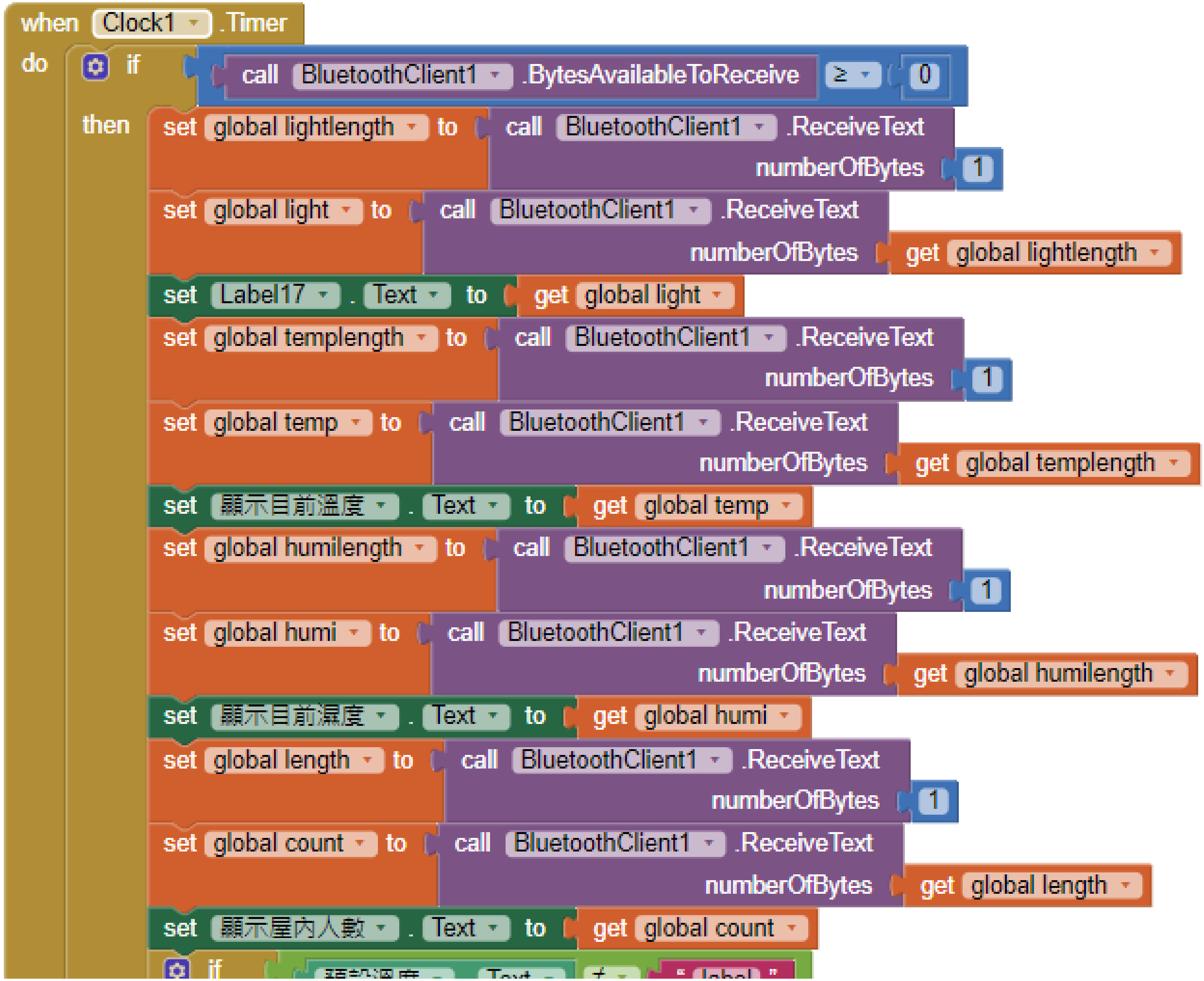
```
initialize global lightlength to " "
initialize global light to create empty list
initialize global templength to " "
initialize global temp to create empty list
initialize global humilength to " "
initialize global humi to create empty list
initialize global length to " "
initialize global count to " "
```

# Sliders 設置



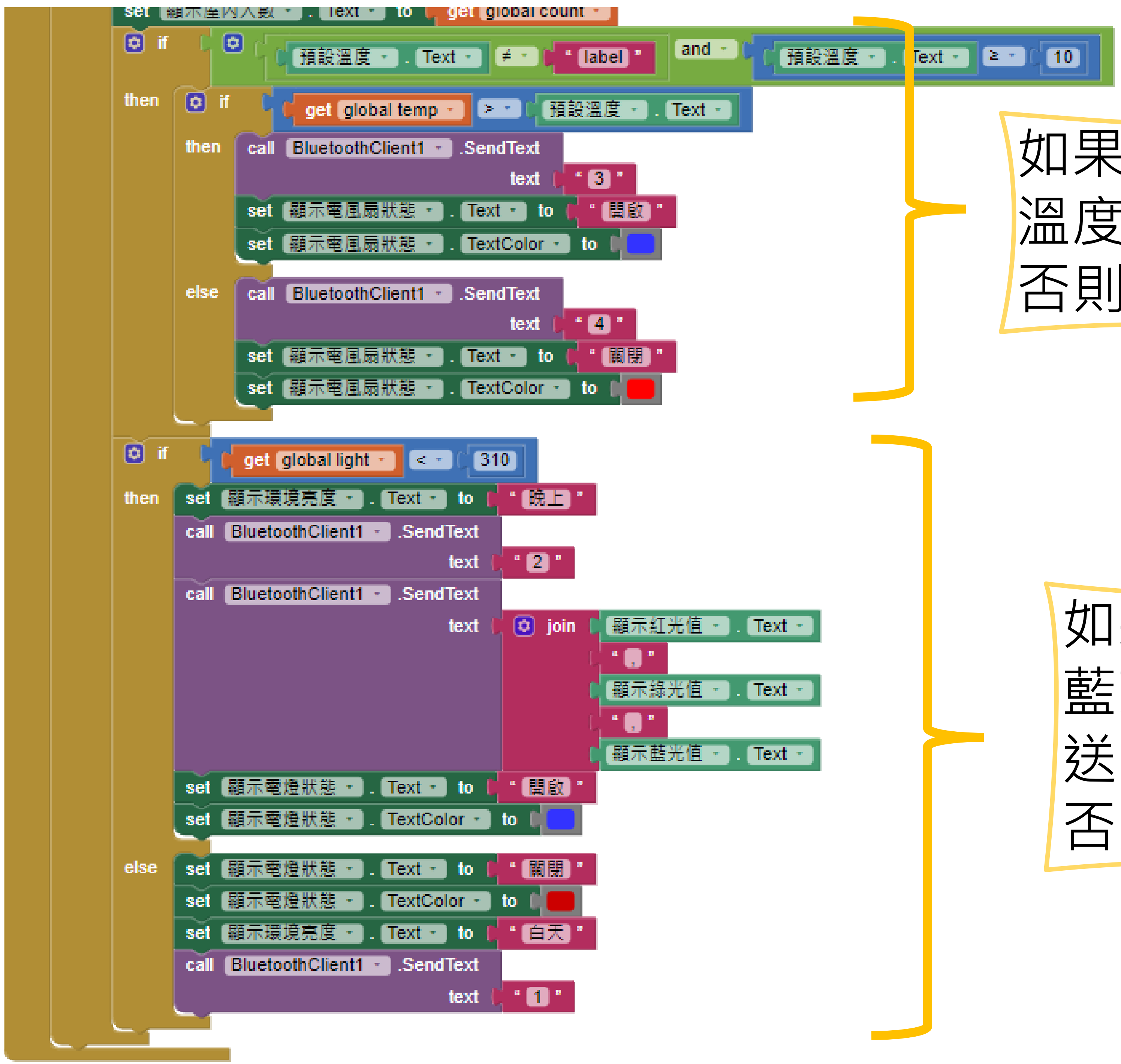
```
when 溫度設定 .PositionChanged
  thumbPosition
do  set 預設溫度 . Text to  round  get thumbPosition

when 紅光設定 .PositionChanged
  thumbPosition
do  set 顯示紅光值 . Text to  round  get thumbPosition

when 綠光設定 .PositionChanged
  thumbPosition
do  set 顯示綠光值 . Text to  round  get thumbPosition

when 藍光設定 .PositionChanged
  thumbPosition
do  set 顯示藍光值 . Text to  round  get thumbPosition
```

# 主程式



```
when Clock1 .Timer
do  if  call BluetoothClient1 .BytesAvailableToReceive ≥ 0
    then  set global lightlength to  call BluetoothClient1 .ReceiveText
                                              numberOfBytes 1
          set global light to  call BluetoothClient1 .ReceiveText
                                        numberOfBytes  get global lightlength
          set Label17 . Text to  get global light
          set global templength to  call BluetoothClient1 .ReceiveText
                                             numberOfBytes 1
          set global temp to  call BluetoothClient1 .ReceiveText
                                       numberOfBytes  get global templength
          set 顯示目前溫度 . Text to  get global temp
          set global humilength to  call BluetoothClient1 .ReceiveText
                                             numberOfBytes 1
          set global humi to  call BluetoothClient1 .ReceiveText
                                       numberOfBytes  get global humilength
          set 顯示目前濕度 . Text to  get global humi
          set global length to  call BluetoothClient1 .ReceiveText
                                         numberOfBytes 1
          set global count to  call BluetoothClient1 .ReceiveText
                                        numberOfBytes  get global length
          set 顯示屋內人數 . Text to  get global count
          if  預設溫度 . Text ≠ " label "
```

# 主程式



```
set 顯示室內人數 . Text to get global count
if  預設溫度 . Text ≠ "label"  and  預設溫度 . Text ≥ 10
then  if  get global temp > 預設溫度 . Text
      then  call BluetoothClient1 .SendText
                               text "3"
            set 顯示電風扇狀態 . Text to "開啟"
            set 顯示電風扇狀態 . TextColor to 🟦
      else  call BluetoothClient1 .SendText
                               text "4"
            set 顯示電風扇狀態 . Text to "關閉"
            set 顯示電風扇狀態 . TextColor to 🟥
if  get global light < 310
then  set 顯示環境亮度 . Text to "晚上"
      call BluetoothClient1 .SendText
                       text "2"
      call BluetoothClient1 .SendText
                       text  join  顯示紅光值 . Text
                                   ","
                                   顯示綠光值 . Text
                                   ","
                                   顯示藍光值 . Text
      set 顯示電燈狀態 . Text to "開啟"
      set 顯示電燈狀態 . TextColor to 🟦
else  set 顯示電燈狀態 . Text to "關閉"
      set 顯示電燈狀態 . TextColor to 🟥
      set 顯示環境亮度 . Text to "白天"
      call BluetoothClient1 .SendText
                       text "1"
```

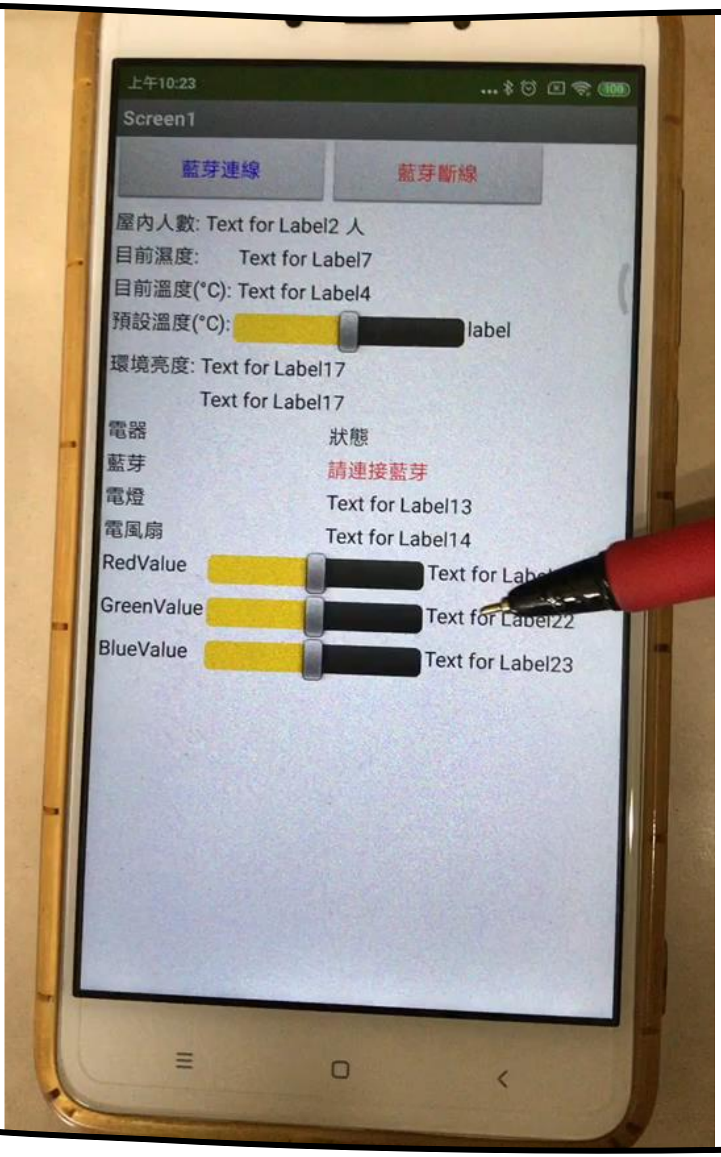
如果現在溫度>預設溫度，藍芽傳送3。否則傳送4。

如果光敏值<310，藍芽傳送4，並傳送RGB值。否則傳送1。

# 06

# 實際操作影片

# 影片



上午10:23

Screen1

藍芽連線 藍芽斷線

屋內人數: Text for Label2 人

目前濕度: Text for Label7

目前溫度(°C): Text for Label4

預設溫度(°C): label

環境亮度: Text for Label17

Text for Label17

| 電器 | 狀態 |
|---|---|
| 藍芽 | 請連接藍芽 |
| 電燈 | Text for Label13 |
| 電風扇 | Text for Label14 |

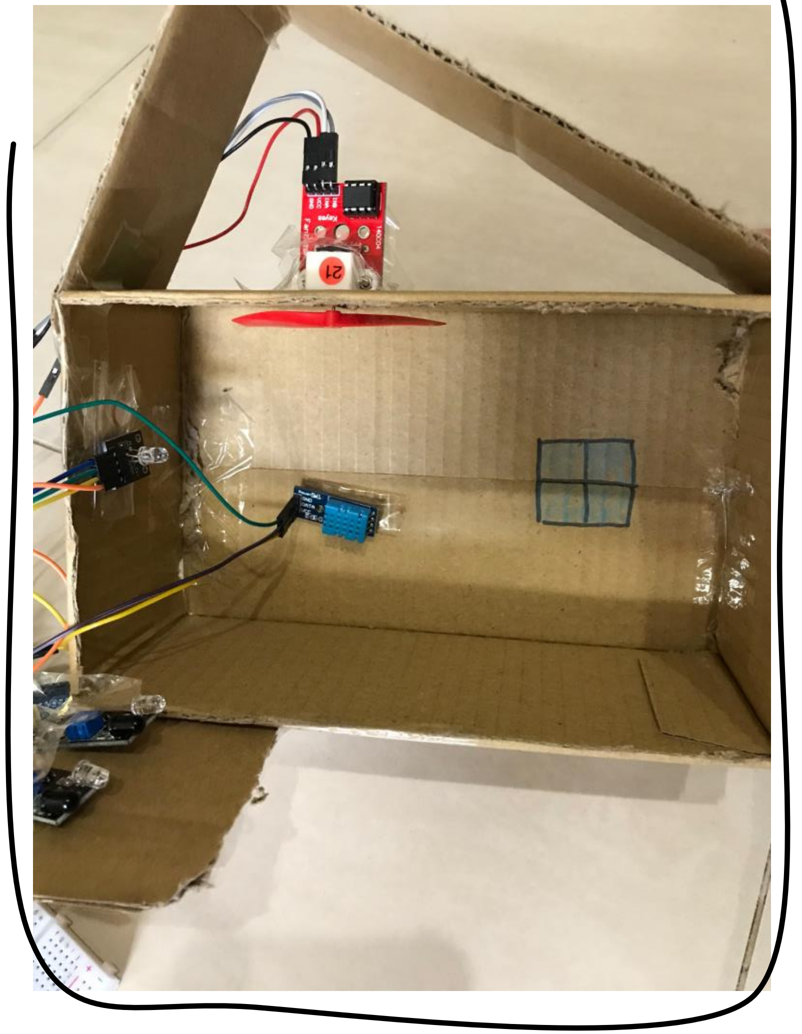
RedValue Text for Lab

GreenValue Text for Label22

BlueValue Text for Label23

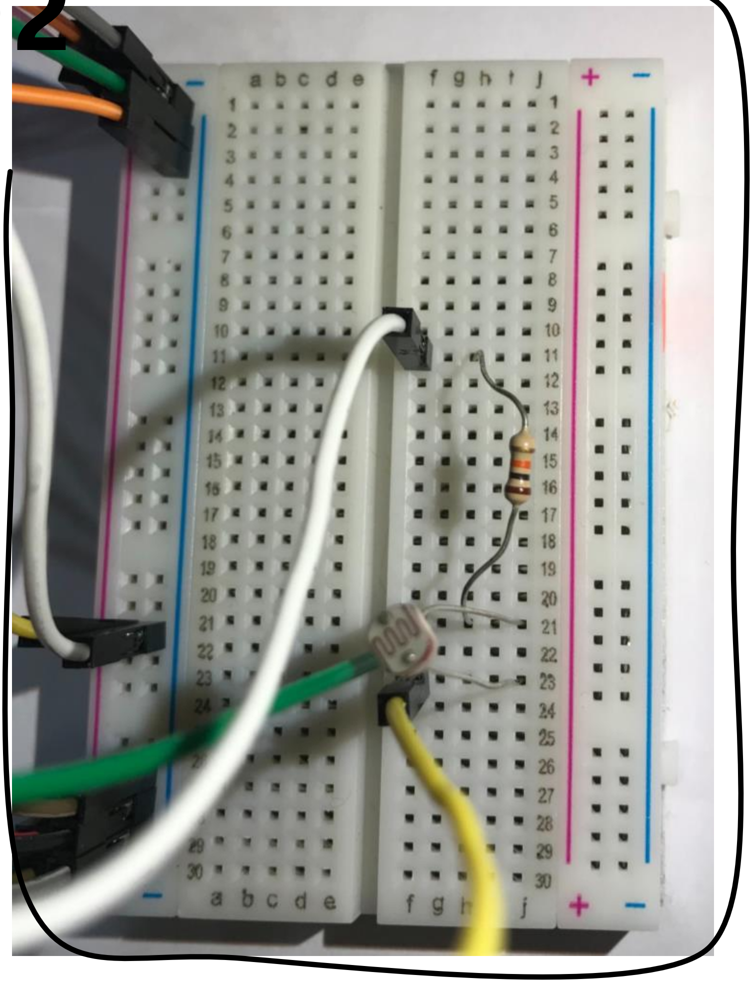
# 元件擺放

01



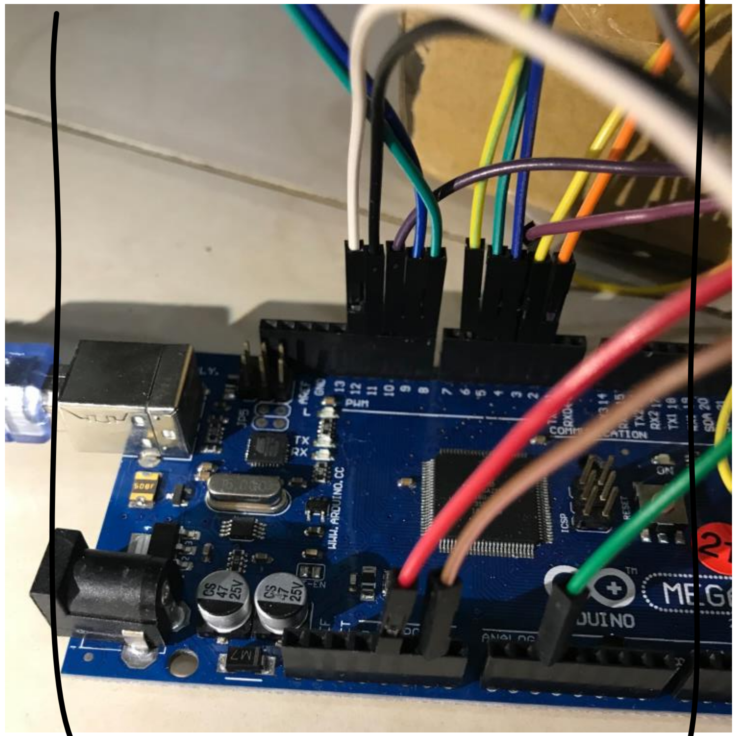
■ 房子

02



■ 麵包板

03

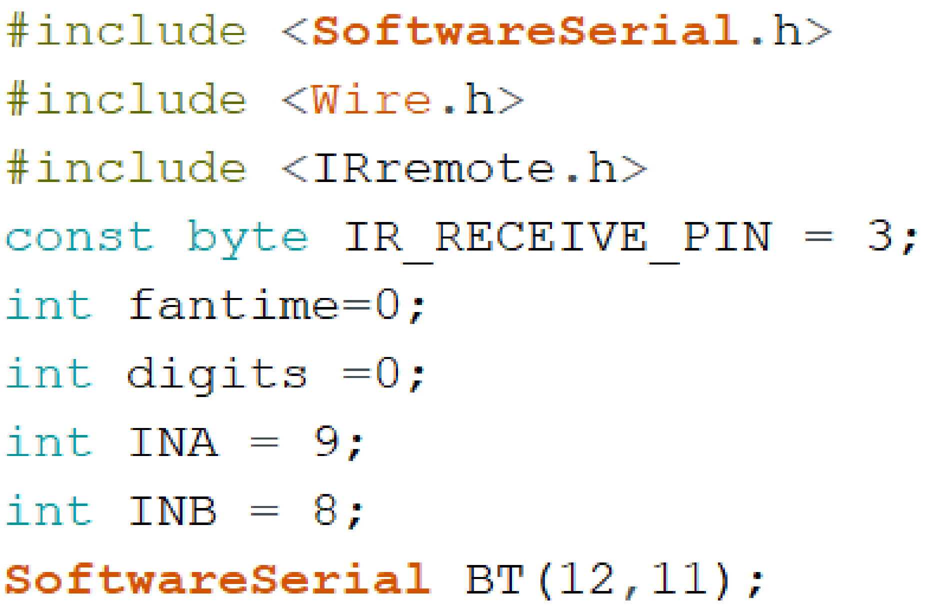


■ Mega板

# 07

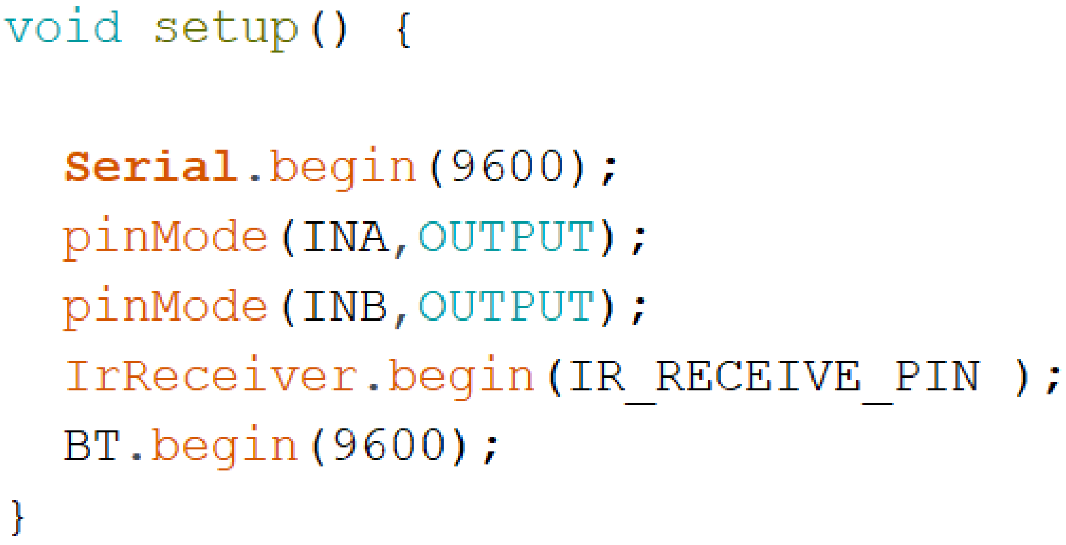
# 進階

## 全域宣告

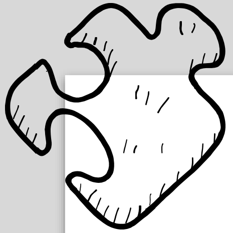


```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <IRremote.h>
const byte IR_RECEIVE_PIN = 3;
int fantime=0;
int digits =0;
int INA = 9;
int INB = 8;
SoftwareSerial BT(12,11);
```

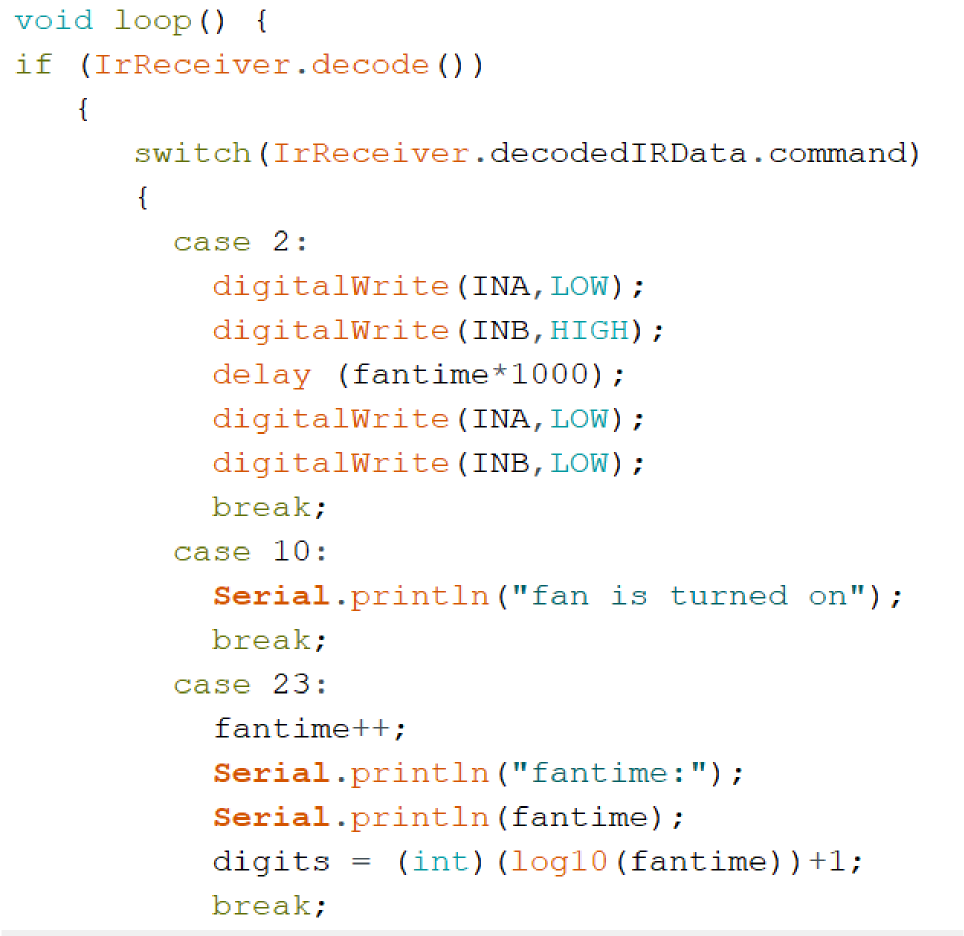
## setup



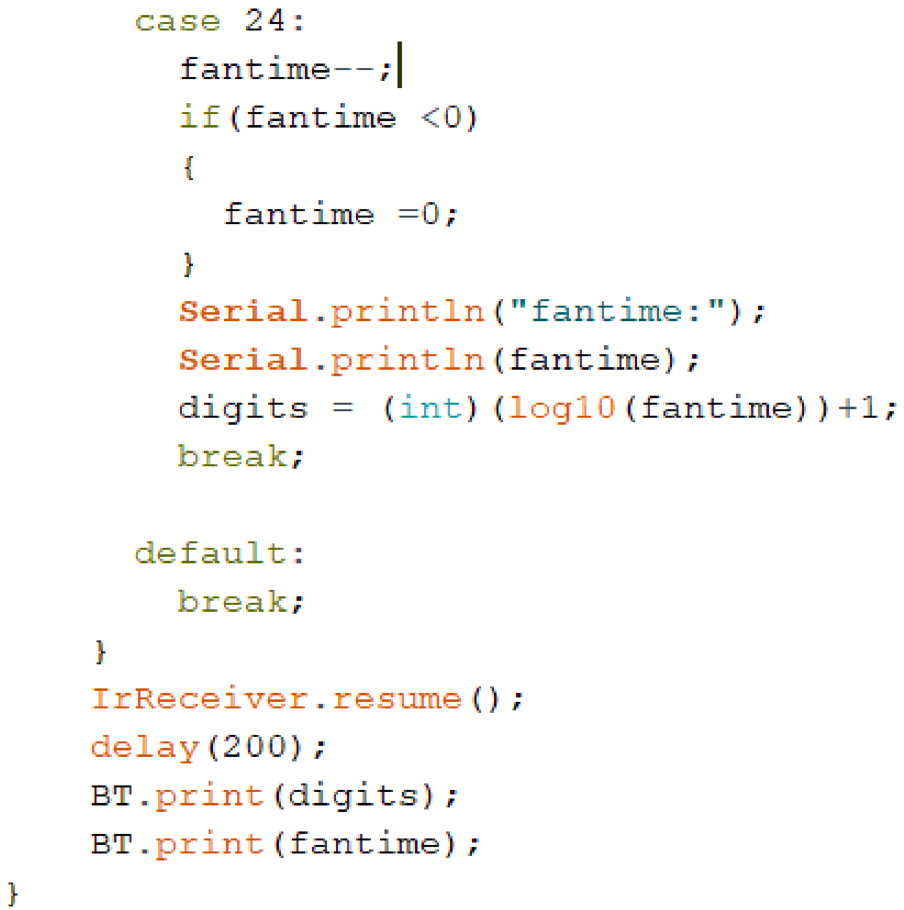
```
void setup() {

  Serial.begin(9600);
  pinMode(INA,OUTPUT);
  pinMode(INB,OUTPUT);
  IrReceiver.begin(IR_RECEIVE_PIN );
  BT.begin(9600);
}
```



# 主程式

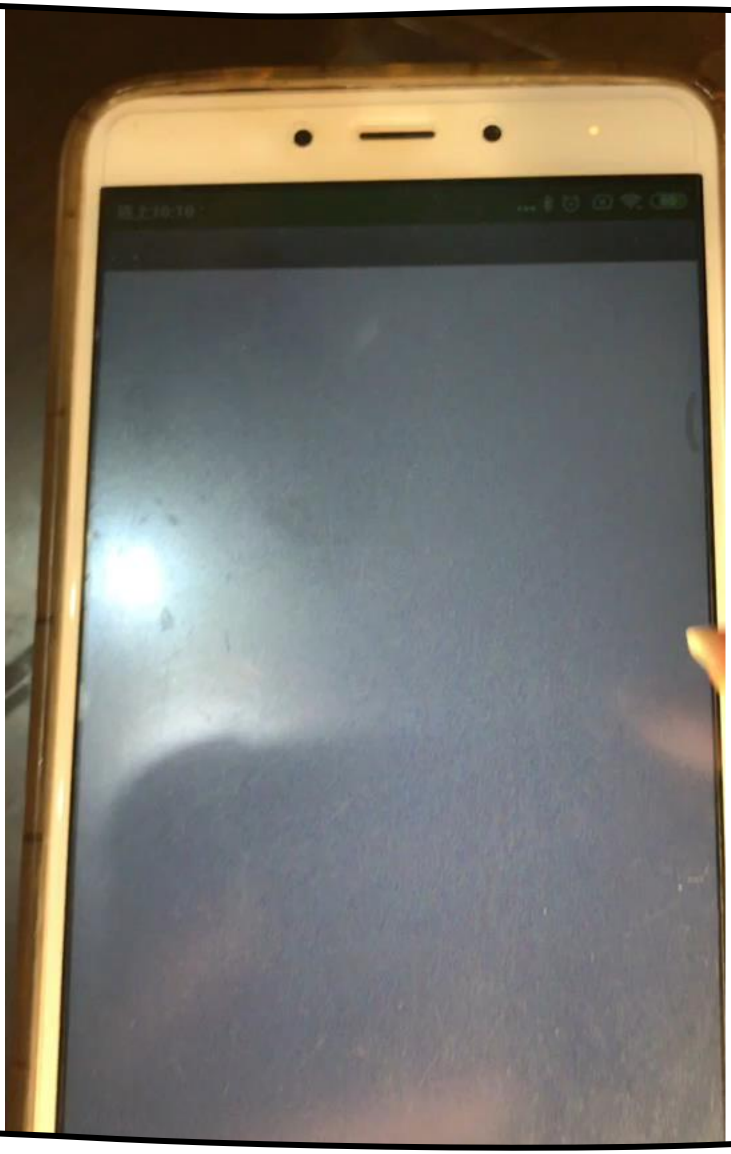


```
void loop() {
if (IrReceiver.decode())
   {
      switch(IrReceiver.decodedIRData.command)
      {
        case 2:
          digitalWrite(INA,LOW);
          digitalWrite(INB,HIGH);
          delay (fantime*1000);
          digitalWrite(INA,LOW);
          digitalWrite(INB,LOW);
          break;
        case 10:
          Serial.println("fan is turned on");
          break;
        case 23:
          fantime++;
          Serial.println("fantime:");
          Serial.println(fantime);
          digits = (int)(log10(fantime))+1;
          break;
```



```
        case 24:
          fantime--;
          if(fantime <0)
          {
            fantime =0;
          }
          Serial.println("fantime:");
          Serial.println(fantime);
          digits = (int)(log10(fantime))+1;
          break;

        default:
          break;
      }
      IrReceiver.resume();
      delay(200);
      BT.print(digits);
      BT.print(fantime);
  }
```

## 影片



# 謝 謝 聆 聽