

# 路口號誌燈

作者：葉富豪

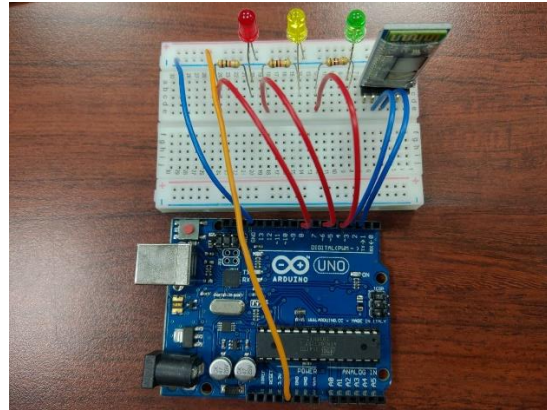
## 路口號誌燈

本單元是透過路口號誌燈實作，讓同學了解 APP Inventor 與 Arduino 的整合應用，並學習到相關的程式邏輯概念。主要是以 Arduino 電路模擬號誌燈情境，經由 APP Inventor 製作藍芽遙控程式，並執行藍芽連接、控制訊號傳送及接收、燈號 LED 控制、中斷藍芽連接，總結以上我們將 APP 的需求整理成列表如下：

需求名稱	需求說明	對應元件	
需求 1	狀態	顯示藍芽連線狀態	Label
需求 2	藍芽裝置選單	讓使用者選擇周遭可連線的藍芽裝置	ListPicker
需求 3	轉換號誌	依使用者指令轉換路口號誌燈	Button
需求 4	中斷連線	讓使用者中斷藍芽連接	Button
需求 5	藍芽連線	將 APP 連線至藍芽裝置	BluetoothClient

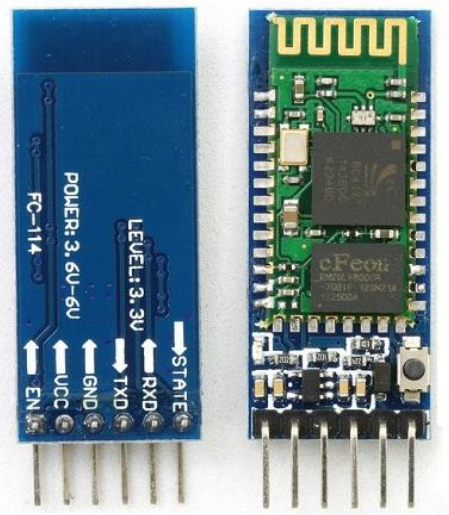
## 需要材料:

材料名稱	數量
Arduino Uno 板	1
麵包板	1
藍芽模組(HC-05)	1
1K 電阻(棕黑紅金)	3
發光二極體(紅)	1
發光二極體(黃)	1
發光二極體(綠)	1



[Arduino 板線路圖]

- 紅色 LED 連接 Pin 8
- 黃色 LED 連接 Pin 6
- 綠色 LED 連接 Pin 4



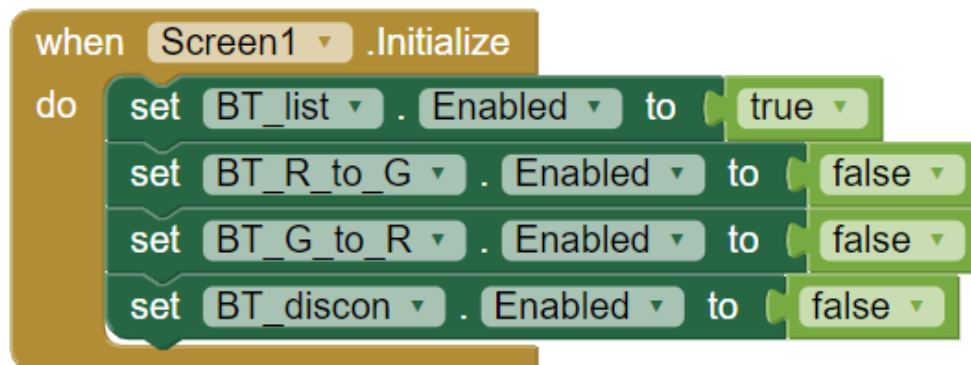
[Arduino 藍芽模組 HC-05 圖]

- 藍芽模組 VCC 連接正極
- 藍芽模組 GND 連接負極
- TXD (設備發送端，即輸出)連接 Pin 2
- RXD (設備接收端，即輸入)連接 Pin 3

元件架構完整圖:



程式初始時只有藍芽選單(BT\_list)需要啟動，因此將啟動(Enabled)的初始值設為 True。其餘功能須等藍芽連線成功才能夠使用，因此將啟動(Enabled)的初始值設定為 False。



藍芽選單(BT\_list)被選擇之前(BeforePicking)，設藍芽選單的成員(Elements)為藍芽裝置的位置與名稱(AddressAndNames)，在使用 APP 連線藍芽以前，請先將藍芽開啟並先與 HC-05 配對(配對金鑰通常為 0000 或是 1234)。



藍芽選單被點選後(AfterPicking)，呼叫藍芽連線選中的藍芽裝置，若有成功連線，則將所有按鈕的功能啟動，因此將啟動(Enabled)設為 True，並且將藍芽選單功能關閉，因此將啟動(Enabled)設為 False，且文字標籤(Con\_text)顯示連線狀態為連線中。

```

when BT_list .AfterPicking
do
  if call BluetoothClient1 .Connect
      address BT_list . Selection
  then
    set BT_list . Enabled to false
    set con_text . Text to "連線中"
    set BT_R_to_G . Enabled to true
    set BT_G_to_R . Enabled to true
    set BT_discon . Enabled to true
  
```

依據點選按鈕發射控制訊號，當 BT\_R\_to\_G 被點選後，則發送“1”為訊號，當 BT\_G\_to\_R 被點選後，則發送“2”為訊號，讓 Arduino 依據訊號來做出相對應的動作。

```

when BT_R_to_G .Click
do
  call BluetoothClient1 .SendText
  text "1"

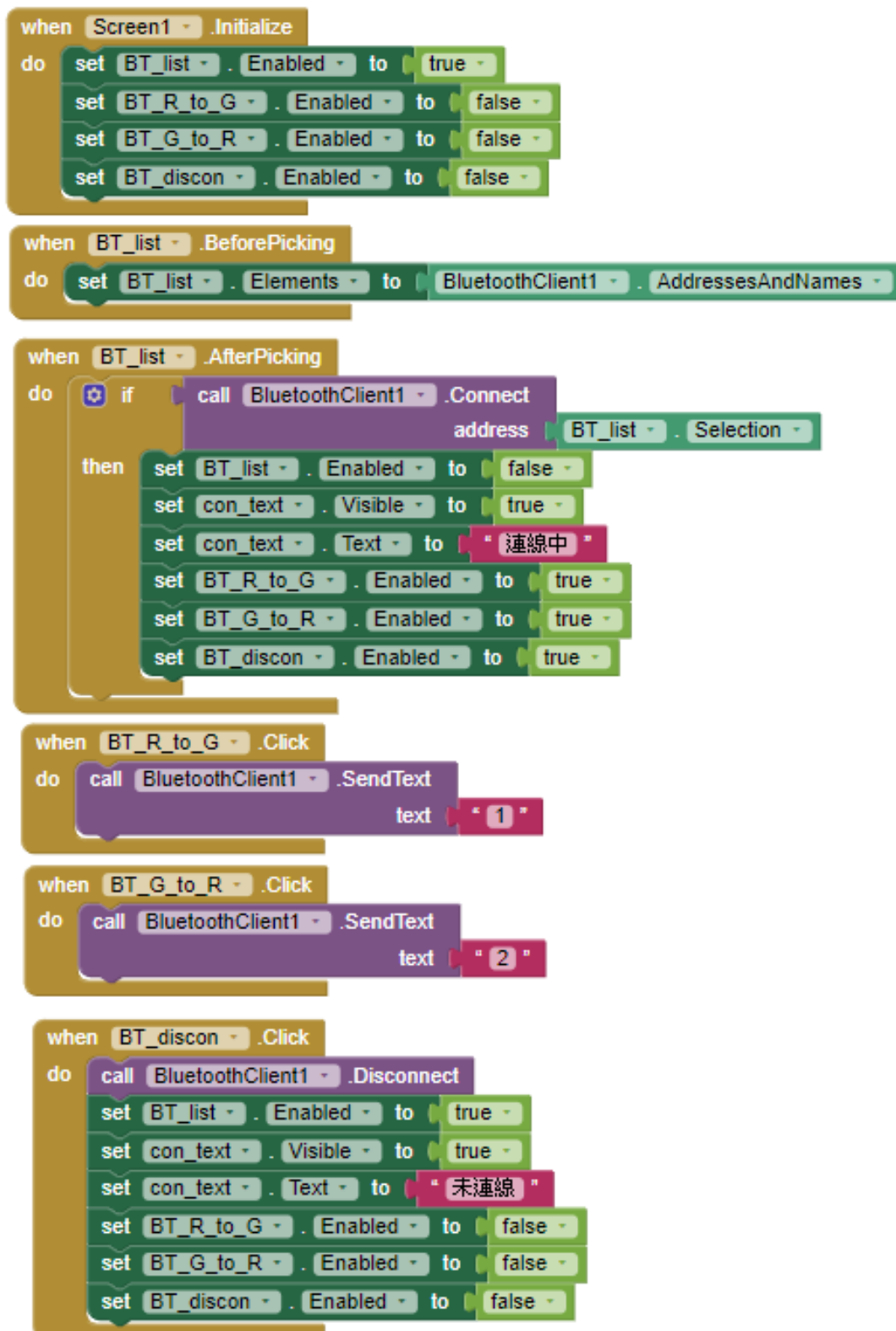
when BT_G_to_R .Click
do
  call BluetoothClient1 .SendText
  text "2"
  
```

當藍芽斷線(BT\_discon)被點選後，則藍芽連線中斷，將所有功能關閉，因此將啟動(Enabled)設為 False，文字標籤(Con\_text)顯示連線狀態為未連線，並且將藍芽選單重啟，因此將啟動(Enabled)設為 True。

```

when BT_discon .Click
do
  call BluetoothClient1 .Disconnect
  set BT_list . Enabled to true
  set con_text . Text to "未連線"
  set BT_R_to_G . Enabled to false
  set BT_G_to_R . Enabled to false
  set BT_discon . Enabled to false
  
```

APP 程式完整圖:



Arduino 程式完整圖:

```

#include<SoftwareSerial.h> // 匯入SoftwareSerial程式庫
SoftwareSerial BT(2,3);    // 接收腳, 傳送腳
const int Rpin = 8;       // 宣告紅色LED腳位
const int Ypin = 6;       // 宣告黃色LED腳位
const int Gpin = 4;       // 宣告綠色LED腳位
int BT_receive;           // 宣告藍芽接收器變數
void setup()
{
  BT.begin(9600); // 設定藍芽模組的連線速率
  pinMode(Rpin,OUTPUT); // 設定紅色LED腳位的模式
  pinMode(Ypin,OUTPUT); // 設定黃色LED腳位的模式
  pinMode(Gpin,OUTPUT); // 設定綠色LED腳位的模式
}
void loop()
{
  if(BT.available())      //判斷藍芽可否接收
  {
    BT_receive = BT.read(); //如果藍芽可接收, 則藍芽變數等於藍芽接收值
    /*如果藍芽變數等於1, 紅色LED亮滅1次, 黃色LED亮滅3次, 綠色LED持續發光*/
    if(BT_receive == '1')
    {
      digitalWrite(Rpin,HIGH);
      delay(2000);
      digitalWrite(Rpin,LOW);

      digitalWrite(Ypin,HIGH);
      delay(200);
      digitalWrite(Ypin,LOW);
      delay(200);
      digitalWrite(Ypin,HIGH);
      delay(200);
      digitalWrite(Ypin,LOW);
      delay(200);
      digitalWrite(Ypin,HIGH);
      delay(200);
      digitalWrite(Ypin,LOW);
      delay(200);

      digitalWrite(Gpin,HIGH);
      delay(2000);
      digitalWrite(Gpin,HIGH);
      delay(200);
    }

    /*如果藍芽變數等於2, 綠色LED亮滅1次, 黃色LED亮滅3次, 紅色LED持續發光*/
    if(BT_receive == '2')
    {
      digitalWrite(Gpin,HIGH);
      delay(2000);
      digitalWrite(Gpin,LOW);

      digitalWrite(Ypin,HIGH);
      delay(200);
      digitalWrite(Ypin,LOW);
      delay(200);
      digitalWrite(Ypin,HIGH);
      delay(200);
      digitalWrite(Ypin,LOW);
      delay(200);
      digitalWrite(Ypin,HIGH);
      delay(200);
      digitalWrite(Ypin,LOW);
      delay(200);

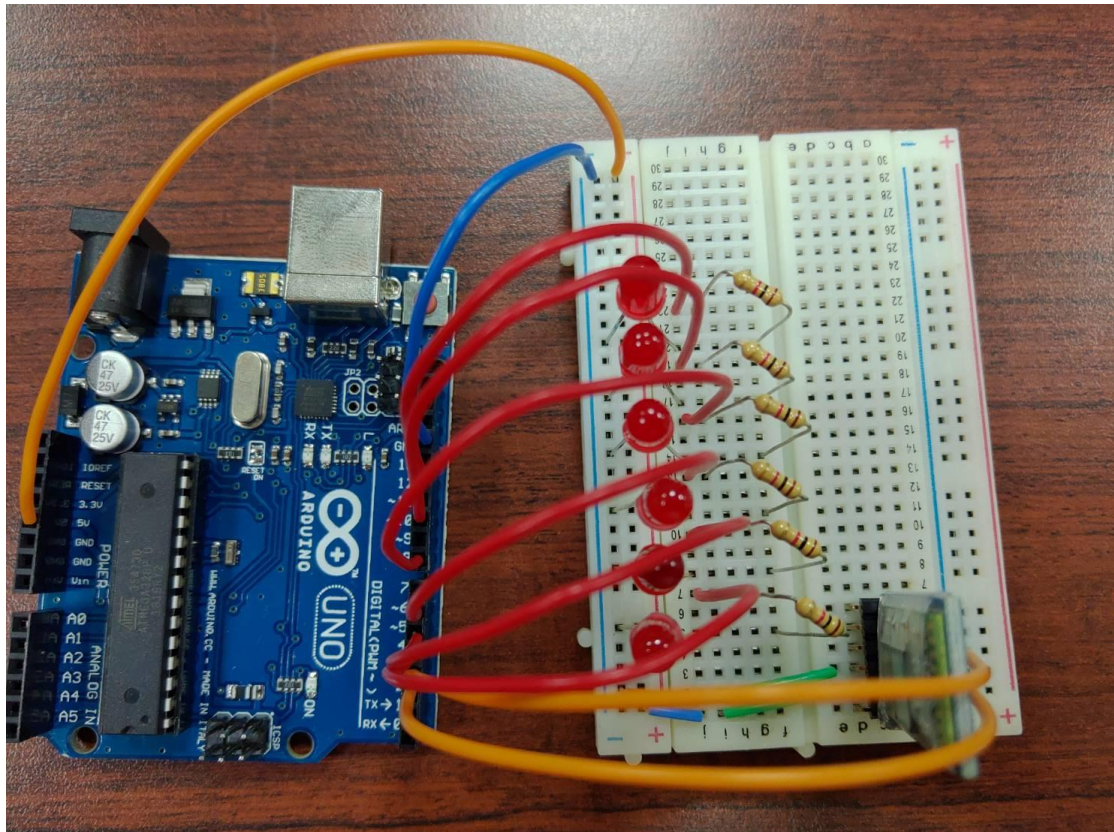
      digitalWrite(Rpin,HIGH);
      delay(2000);
      digitalWrite(Rpin,HIGH);
      delay(200);
    }
    delay(100);
  }
}

```



### 進階練習:

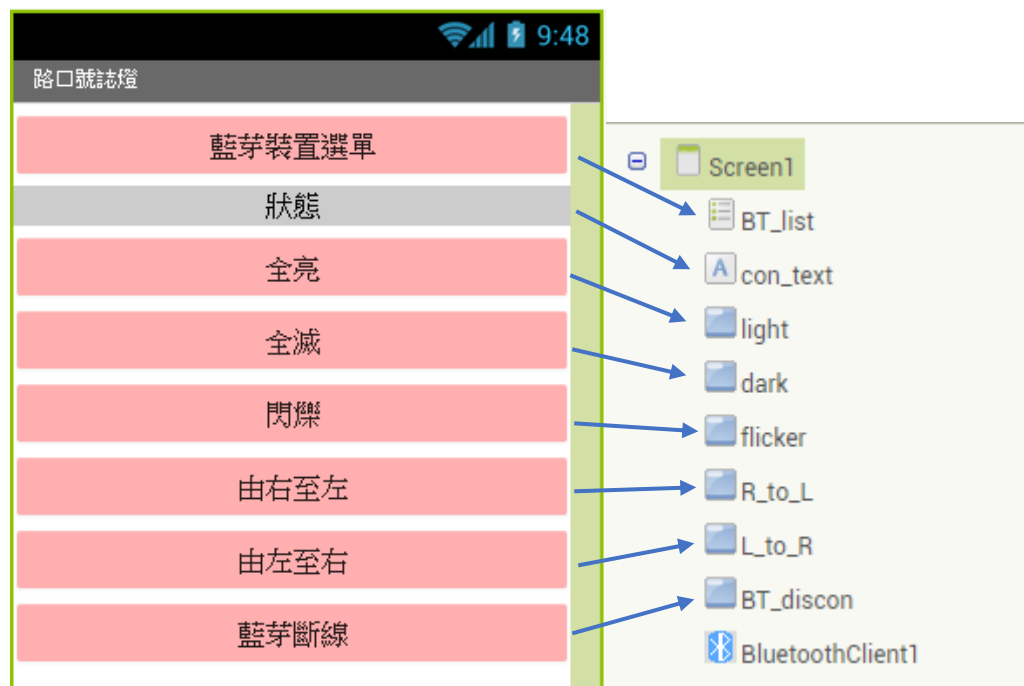
讓手機 APP 控制 LED 五種發光方式，分別為全亮、全滅、閃爍 3 次(每次閃爍間隔為 0.2 秒)、由右至左閃爍 1 次(每次閃爍間隔為 0.2 秒)、由左至右閃爍 1 次(每次閃爍間隔為 0.2 秒)。



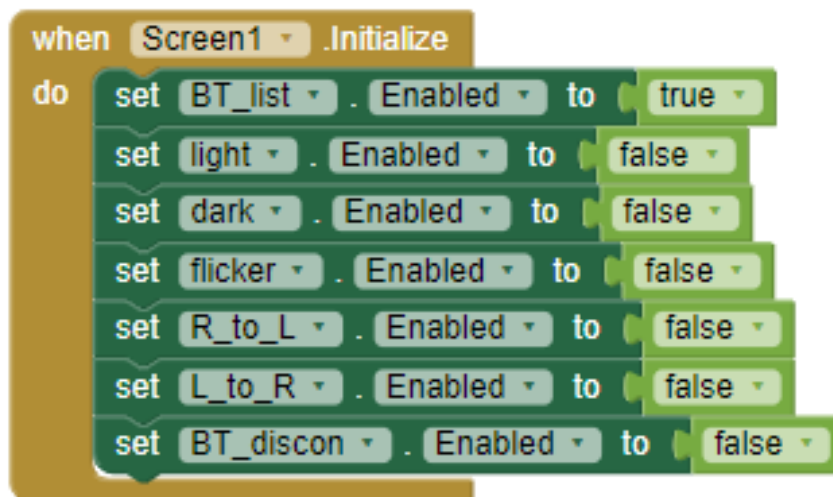
[Arduino 板線路圖]

- 紅色(R1)LED 連接 Pin 4
- 紅色(R2)LED 連接 Pin 5
- 紅色(R3)LED 連接 Pin 6
- 紅色(R4)LED 連接 Pin 8
- 紅色(R5)LED 連接 Pin 10
- 紅色(R6)LED 連接 Pin 12
- 藍芽模組 VCC 連接正極
- 藍芽模組 GND 連接負極
- TXD (設備發送端，即輸出)連接 Pin 2
- RXD (設備接收端，即輸入)連接 Pin 3

元件架構完整圖:



依據進階練習的功能，將按鈕增加至五個，並且將藍芽選單以外的按鈕設定啟動(Enabled)初始值為 False。





當藍芽選單被點選後，呼叫藍芽連線選中的藍芽裝置，將按鈕啟動(Enabled)的初始值設為 True，藍芽選單啟動(Enabled)的初始值設為 False，並且依據所點選的按鈕分別發送“1”，“2”，“3”，“4”，“5”為訊號，讓 Arduino 依據訊號來做出相對應的動作。

```

when BT_list .AfterPicking
do
  if call BluetoothClient1 .Connect
     address BT_list . Selection
  then
    set BT_list . Enabled to false
    set con_text . Text to "連線中"
    set light . Enabled to true
    set dark . Enabled to true
    set flicker . Enabled to true
    set R_to_L . Enabled to true
    set L_to_R . Enabled to true
    set BT_discon . Enabled to true

when light .Click
do
  call BluetoothClient1 .SendText
  text "1"

when dark .Click
do
  call BluetoothClient1 .SendText
  text "2"

when flicker .Click
do
  call BluetoothClient1 .SendText
  text "3"

when R_to_L .Click
do
  call BluetoothClient1 .SendText
  text "4"

when L_to_R .Click
do
  call BluetoothClient1 .SendText
  text "5"

```

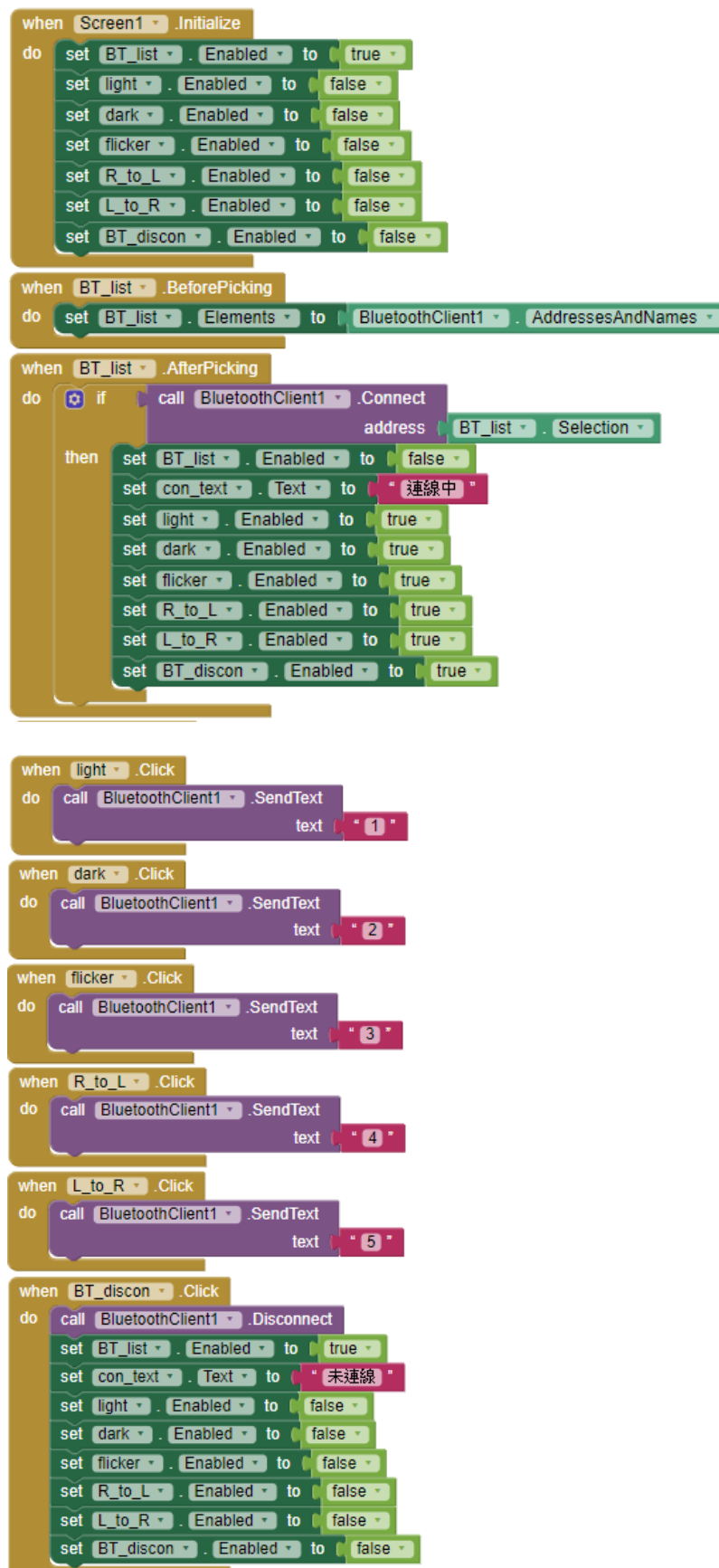
當藍芽斷線被點選後，中斷藍芽連線，並將藍芽選單啟動(Enabled)初始值設為 True，其餘按鈕的啟動(Enabled)初始值設為 False。

```

when BT_discon .Click
do
  call BluetoothClient1 .Disconnect
  set BT_list . Enabled to true
  set con_text . Text to "未連線"
  set light . Enabled to false
  set dark . Enabled to false
  set flicker . Enabled to false
  set R_to_L . Enabled to false
  set L_to_R . Enabled to false
  set BT_discon . Enabled to false

```

APP 程式完整圖:



Arduino 程式完整圖:

```

#include<SoftwareSerial.h> // 匯入SoftwareSerial程式庫
SoftwareSerial BT(2,3); // 接收腳 , 傳送腳
const int R1pin = 4 ; // 宣告紅色LED腳位
const int R2pin = 5 ; // 宣告紅色LED腳位
const int R3pin = 6 ; // 宣告紅色LED腳位
const int R4pin = 8 ; // 宣告紅色LED腳位
const int R5pin = 10 ; // 宣告紅色LED腳位
const int R6pin = 12 ; // 宣告紅色LED腳位
int BT_receive; // 宣告藍芽接收器變數
void setup()
{
  BT.begin(9600); // 設定藍芽模組的連線速率
  pinMode(R1pin,OUTPUT); // 設定紅色LED腳位的模式
  pinMode(R2pin,OUTPUT); // 設定紅色LED腳位的模式
  pinMode(R3pin,OUTPUT); // 設定紅色LED腳位的模式
  pinMode(R4pin,OUTPUT); // 設定紅色LED腳位的模式
  pinMode(R5pin,OUTPUT); // 設定紅色LED腳位的模式
  pinMode(R6pin,OUTPUT); // 設定紅色LED腳位的模式
}
void loop()
{
  if(BT.available()) //判斷藍芽可否接收
  {
    BT_receive = BT.read(); //如果藍芽可接收，則藍芽變數等於藍芽接收值
    /*如果藍芽變數等於1，紅色LED全亮*/
    if(BT_receive == '1')
    {
      digitalWrite(R1pin,HIGH);
      digitalWrite(R2pin,HIGH);
      digitalWrite(R3pin,HIGH);
      digitalWrite(R4pin,HIGH);
      digitalWrite(R5pin,HIGH);
      digitalWrite(R6pin,HIGH);
    }

    /*如果藍芽變數等於2，紅色LED全滅*/
    if(BT_receive == '2')
    {
      digitalWrite(R1pin,LOW);
      digitalWrite(R2pin,LOW);
      digitalWrite(R3pin,LOW);
      digitalWrite(R4pin,LOW);
      digitalWrite(R5pin,LOW);
      digitalWrite(R6pin,LOW);
    }
  }
}

```

```

/*如果藍芽變數等於3，紅色LED閃爍3次*/
if(BT_receive == '3')
{
  /*for迴圈，重複的程式使用for迴圈重複執行*/
  for(int i = 0 ; i < 3 ; i++)
  {
    digitalWrite(R1pin,HIGH);
    digitalWrite(R2pin,HIGH);
    digitalWrite(R3pin,HIGH);
    digitalWrite(R4pin,HIGH);
    digitalWrite(R5pin,HIGH);
    digitalWrite(R6pin,HIGH);
    delay(200);
    digitalWrite(R1pin,LOW);
    digitalWrite(R2pin,LOW);
    digitalWrite(R3pin,LOW);
    digitalWrite(R4pin,LOW);
    digitalWrite(R5pin,LOW);
    digitalWrite(R6pin,LOW);
    delay(200);
  }
}

/*如果藍芽變數等於4，紅色LED由右至左依序閃爍*/
if(BT_receive == '4')
{
  digitalWrite(R1pin,HIGH);
  delay(100);
  digitalWrite(R1pin,LOW);
  delay(100);
  digitalWrite(R2pin,HIGH);
  delay(100);
  digitalWrite(R2pin,LOW);
  delay(100);
  digitalWrite(R3pin,HIGH);
  delay(100);
  digitalWrite(R3pin,LOW);
  delay(100);
  digitalWrite(R4pin,HIGH);
  delay(100);
  digitalWrite(R4pin,LOW);
  delay(100);
  digitalWrite(R5pin,HIGH);
  delay(100);
  digitalWrite(R5pin,LOW);
  delay(100);
  digitalWrite(R6pin,HIGH);
  delay(100);
  digitalWrite(R6pin,LOW);
  delay(100);
}

```

```
.  
/*如果藍芽變數等於5，紅色LED由左至右依序閃爍*/  
if(BT_receive == '5')  
{  
  digitalWrite(R6pin,HIGH);  
  delay(100);  
  digitalWrite(R6pin,LOW);  
  delay(100);  
  digitalWrite(R5pin,HIGH);  
  delay(100);  
  digitalWrite(R5pin,LOW);  
  delay(100);  
  digitalWrite(R4pin,HIGH);  
  delay(100);  
  digitalWrite(R4pin,LOW);  
  delay(100);  
  digitalWrite(R3pin,HIGH);  
  delay(100);  
  digitalWrite(R3pin,LOW);  
  delay(100);  
  digitalWrite(R2pin,HIGH);  
  delay(100);  
  digitalWrite(R2pin,LOW);  
  delay(100);  
  digitalWrite(R1pin,HIGH);  
  delay(100);  
  digitalWrite(R1pin,LOW);  
  delay(100);  
}  
}  
delay(100);  
}
```

---