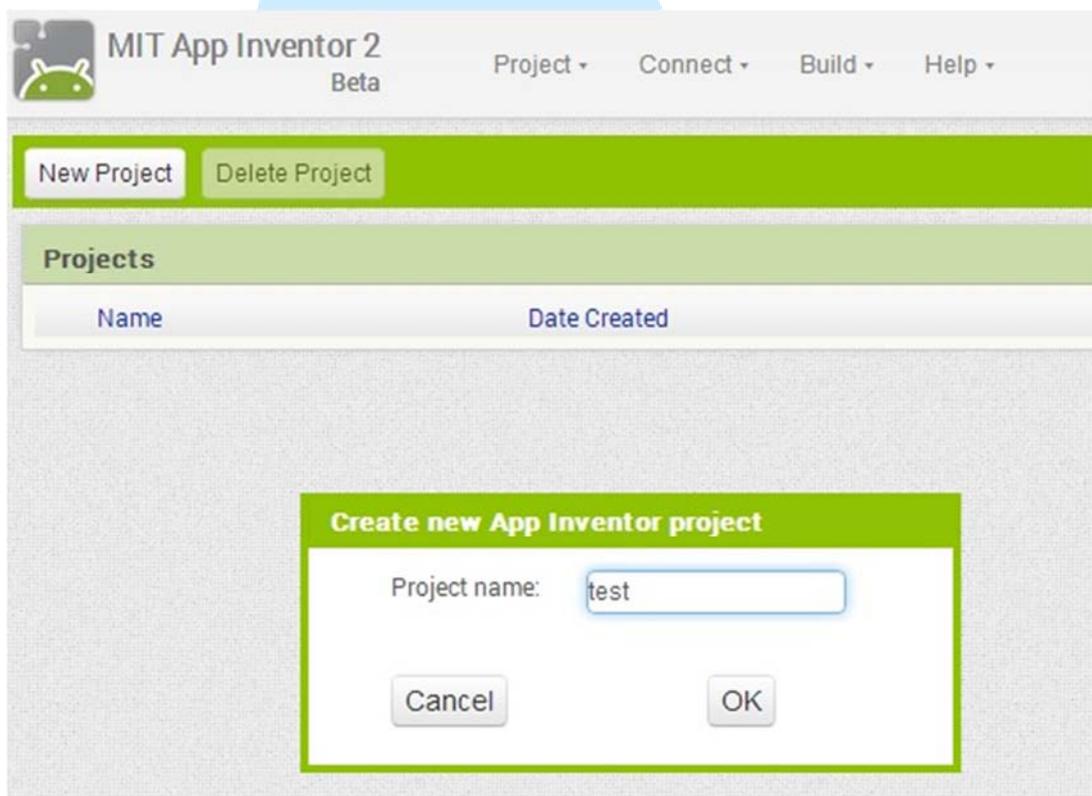


App Inventor 的基本介紹和藍芽連結功能的使用

一、專案建置與基本介紹

我們 MIT App Inventor 2，它是圖形化介面的開發環境，只要拉物件和方塊圖就能完成一個 APP，是不是很簡單呢？

首先，進入網址 <http://ai2.appinventor.mit.edu/> 並登入 google 帳號後
點擊 New Project 來新增專案，並打上專案名稱點擊 OK，如下圖。



進入之後會看到開發介面，主要分為四項，Palette、Viewer、Components 和 Properties

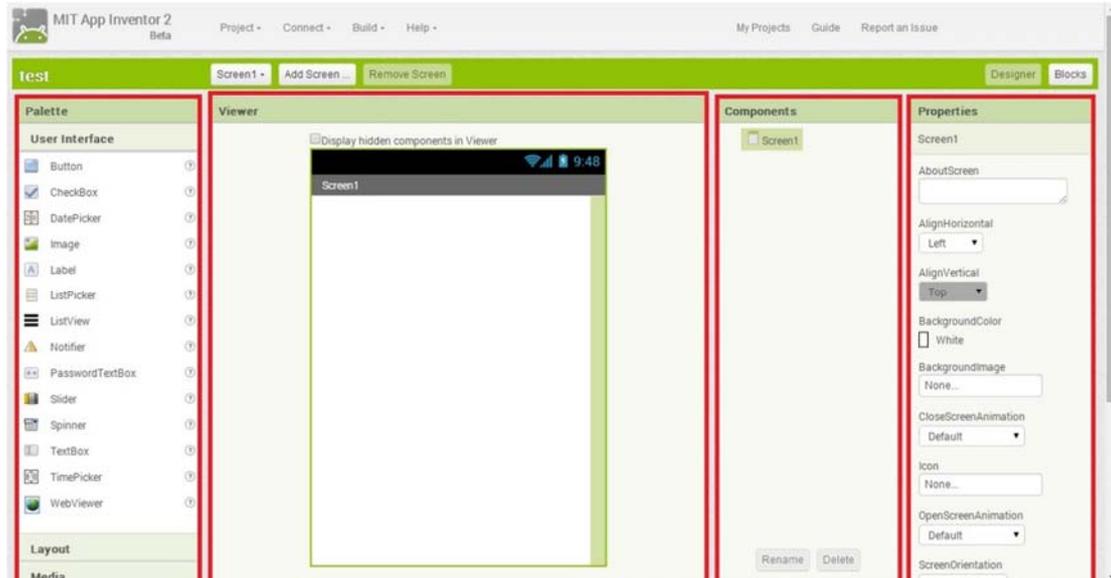
Palette: 所有元件的放置區，需要那些功能就把它拖到 Viewer 區的螢幕裡，.net 玩家看了應該會覺得很親切，很多功能都大同小異。

Viewer: 裝置預覽的畫面，把需要的元件拖曳進來並做排版，但實際位置還是有出入，放到裝置上看會比較準。

Components: 所有拖曳到預覽畫面的元件(包括隱藏的)都會出現在這裡面，

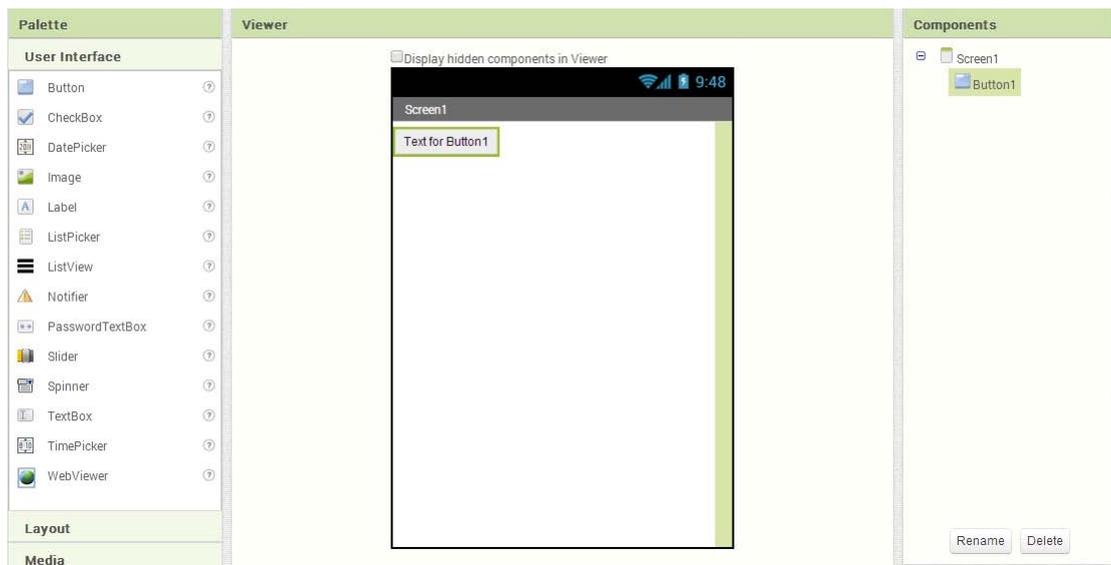
我們可以點擊裡面的元件到右邊的 Properties 做修改。

Properties: 顯示元件的屬性，可以在預覽畫面中點擊或 Components 裡點擊即可編輯元件的屬性。

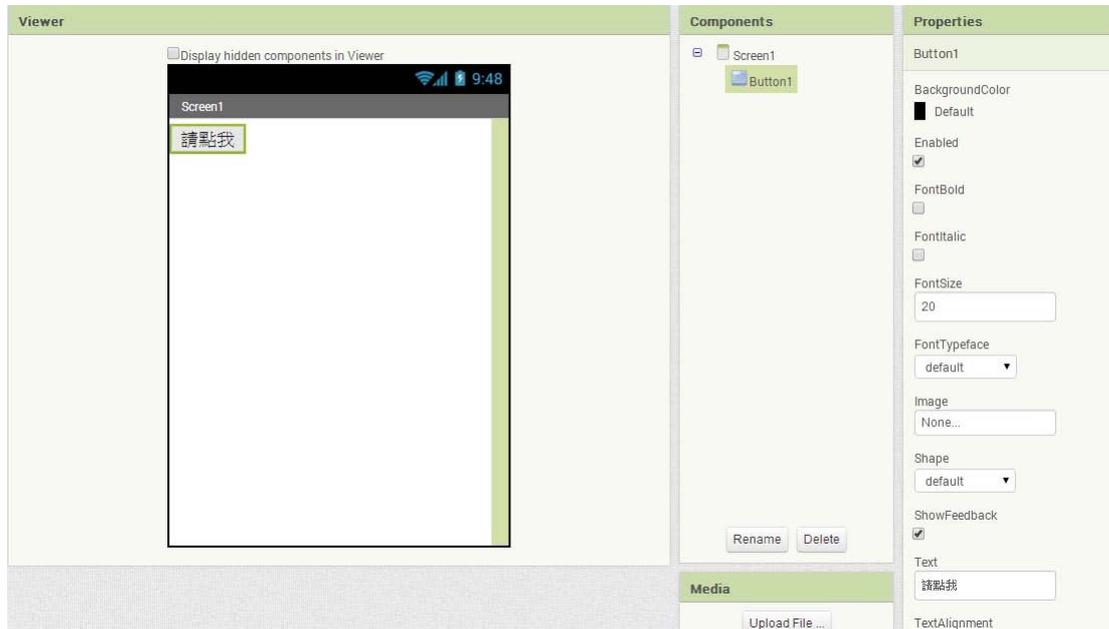


二、第一個程式

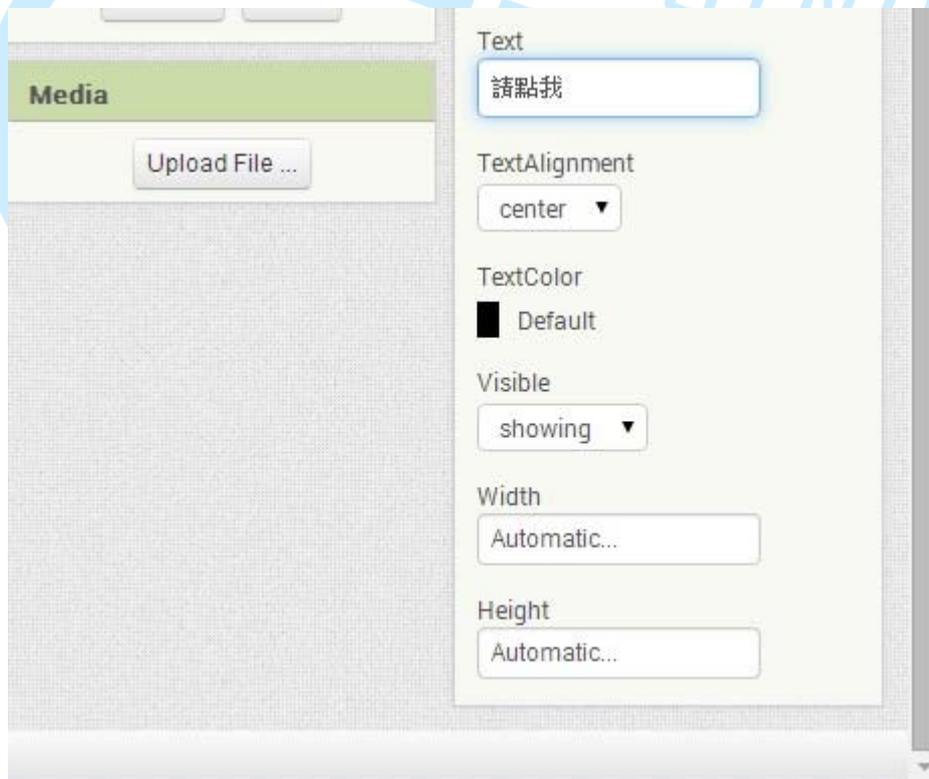
我們先來做一個按按鈕會改變文字的程式，先把左側的 button(按鈕)點住並拖曳到預覽畫面中，如下圖，如果拖錯元件可在 Components 框框內點擊 Delete 出現對話框按 OK 即可刪除，另外 Rename 可對元件做更名，但在寫程式也就是堆方塊時也要找相同名字才能做調用的動作。



再來我們到 Properties(屬性) 框框中的 FontSize (字型大小)改為 20，然後在 Text 選項中，改成想要的文字，例如：請點我，如下圖。



拉霸往下拉或滑鼠往下滾到下面可以看到 Width(寬) 與 Height(高) 的選項。



點開可以看到三個控制項: Automatic(自動)、Fill parent(充滿)、pixels(自訂大小)，我們在 Width(寬) 的屬性改為充滿，如下圖。



在 Height(高)的選項中選擇自訂並輸入 50，單位為 pixels，如下圖。



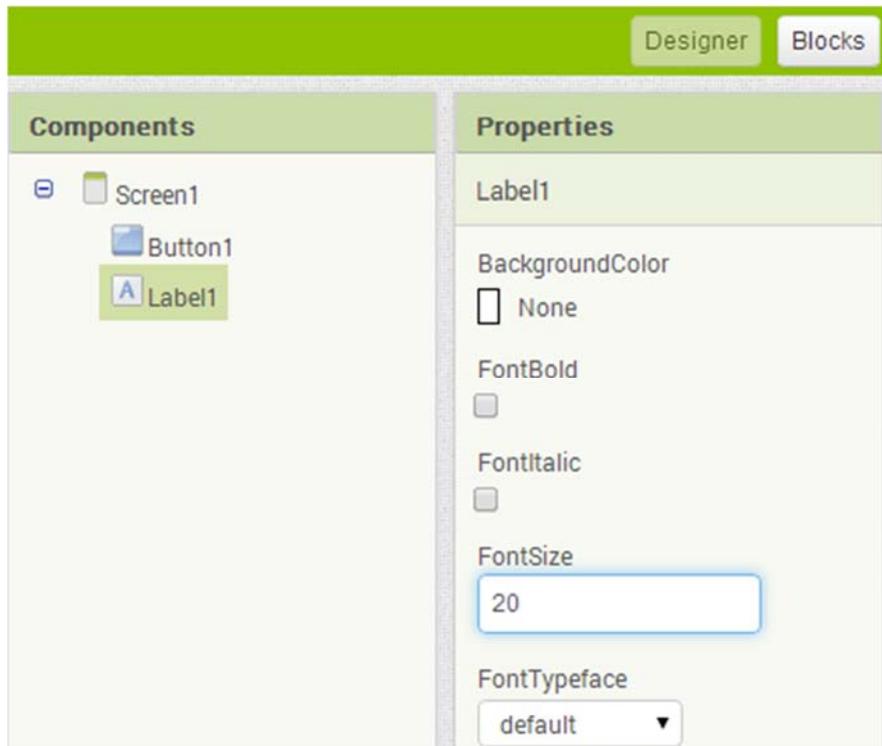
完成圖，記得屬鏡中的 Height(高)要比 FontSize(字型大小)大，不然會切到字喔。



接著再把 Label 拖曳到預覽畫面中，如下圖。



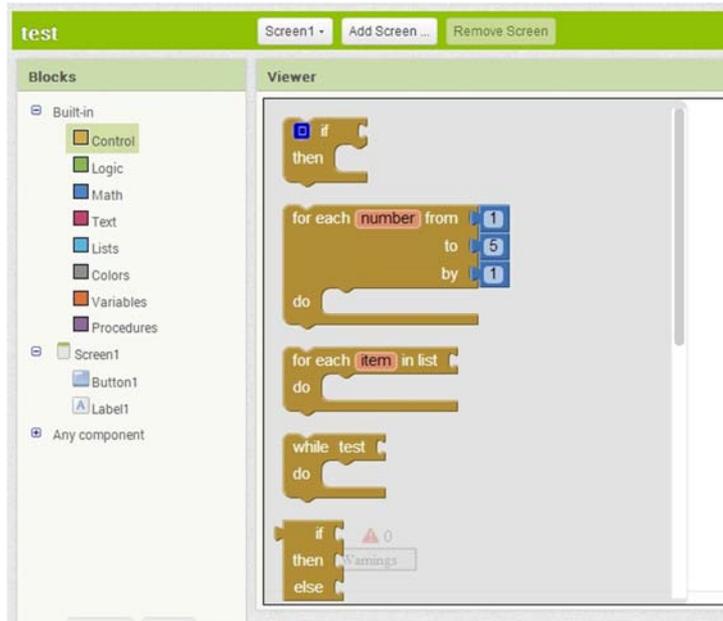
在再來我們將 Properties 框框中的 FontSize 一樣改為 20，如下圖，接著看到右上角的 Designer 是設計程式 UI(介面)的地方也是目前所在的地方，而 Blocks 是堆積木區，也就是撰寫程式的地方，我們程式介面也做的差不多了，接著來點擊 Blocks 到堆積木區來一探究竟吧！



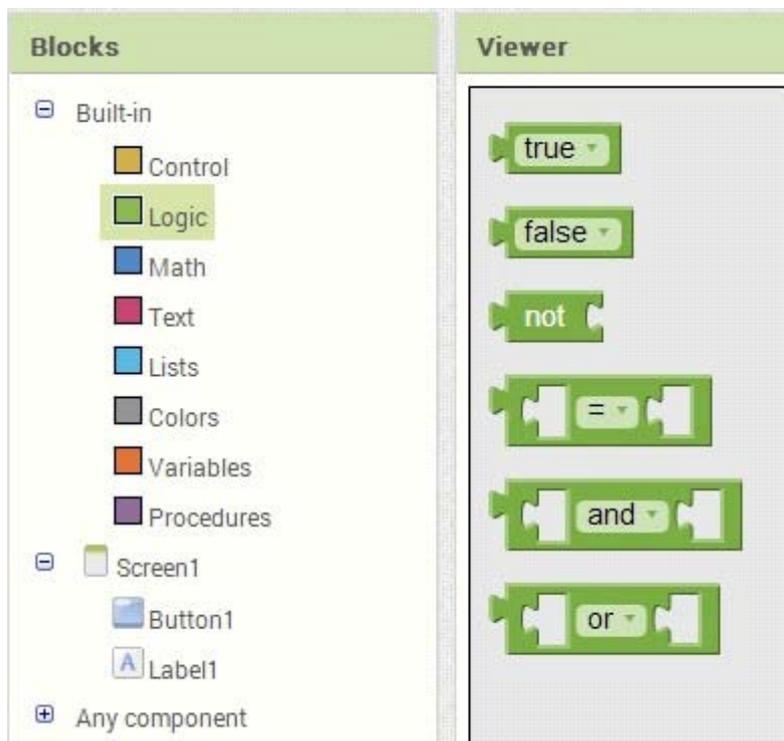
主要分為兩大區域，分別是 Blocks 積木區和 Viewer 觀看區也就是程式撰寫的區域，我們先來介紹 Blocks 的部分吧。



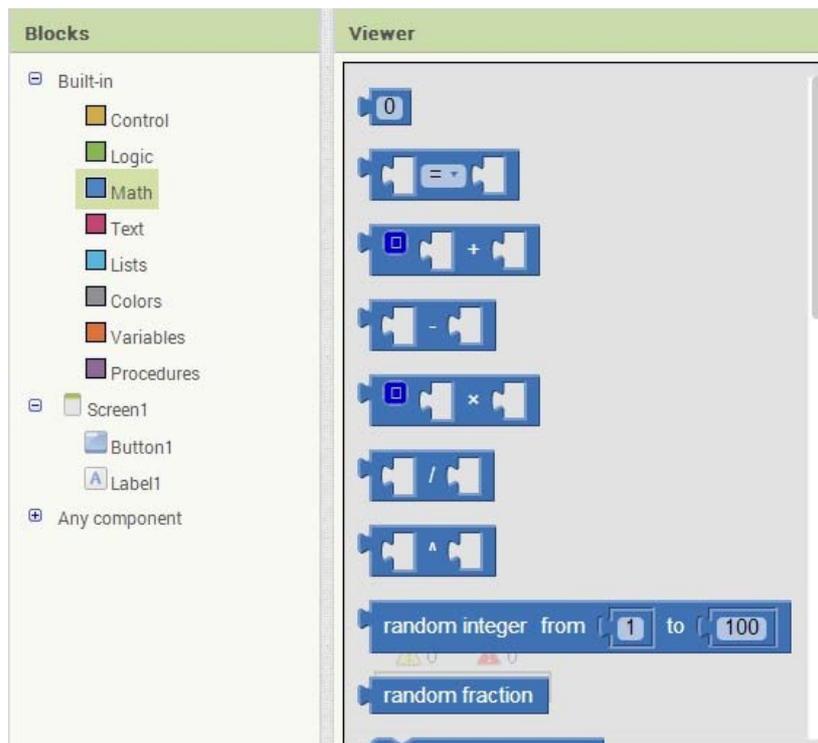
Blocks 積木區，主要會用到的部分是內建的 Built-in 及剛剛拉進預覽視窗中元件都會出現在 Screen1 中，我們現在點擊 Built-in 中的 Control(控制)選項，如下圖，有沒有很熟悉呢？我們常用的判斷式和迴圈都放在這裡喔~



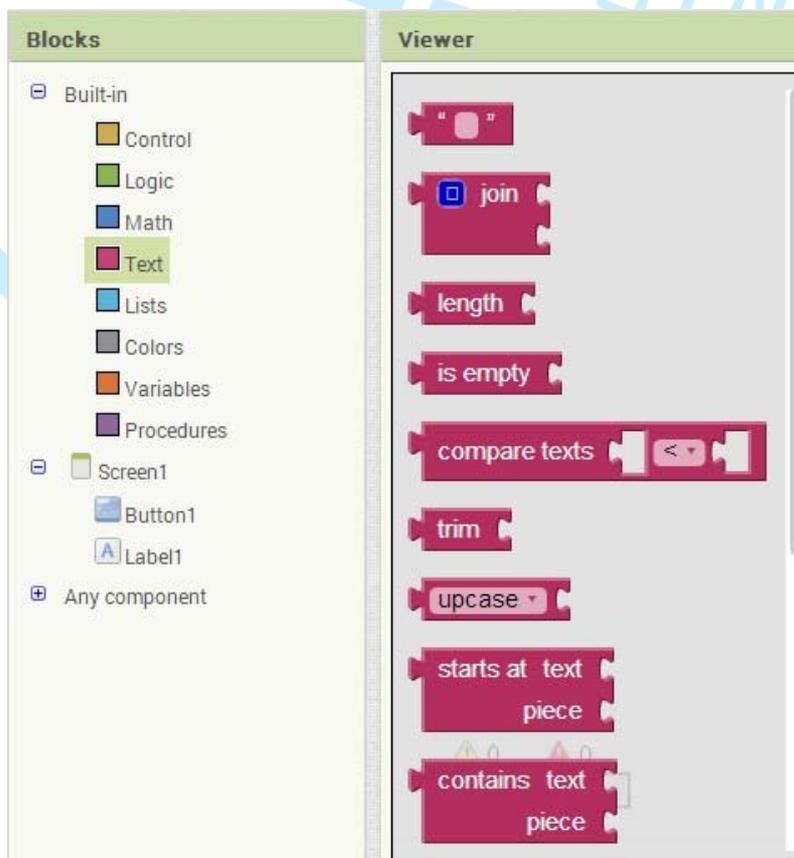
接下去是 Logic(邏輯)，布林運算大家應該都不陌生，這裡就不多做介紹了。



Math(數學) · 裡面包含了基本運算及隨機變數 · 連三角函數都有喔。



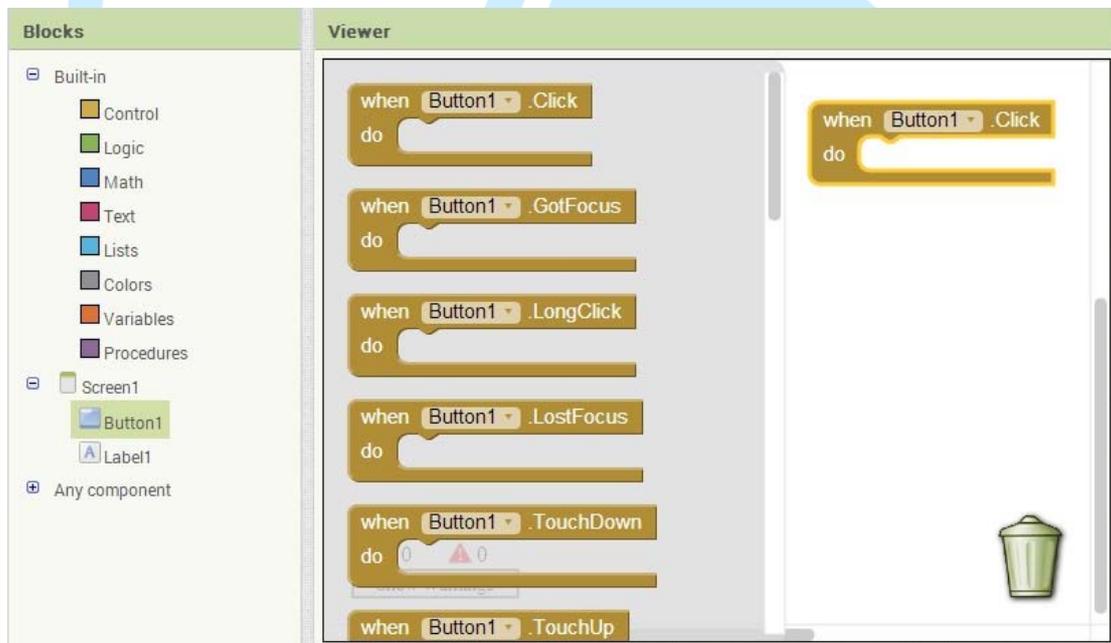
Text(文字) · 包含字串處理都在裡面。



Variables(變量) · 宣告、改變、取得變量的指令都在裡面。



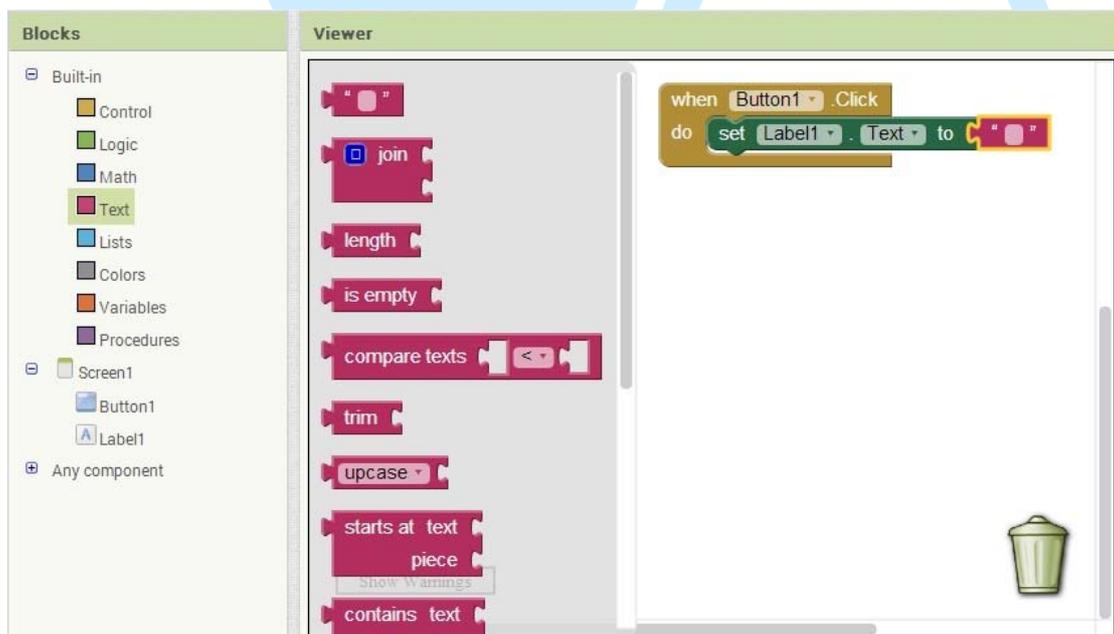
接著要進入正題啦，當 Button 被點擊之後會觸發 Click 的指令，我們就可以針對這個控制項做一些事情，點擊 Button1 可以看到控制項(積木)跳出來，我們拖曳 Click(點擊)這個積木出來到右側，如果拖錯東西出來可以拖到右下角的垃圾桶把它刪除喔，如下圖。



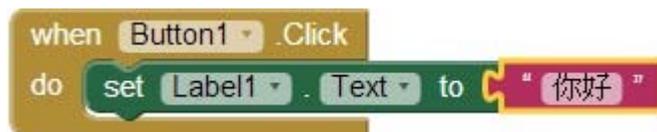
接著點擊左側的 Label1，把標有「set Label1.Text to」的積木拖曳到剛剛的積木裡把它拼起來，如下圖，它的功能為改變 Label1 顯示的字。



接著點擊左側的 Text，把第一個積木拖到右邊拚進去。

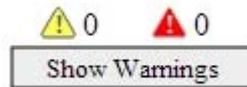


滑鼠點擊剛剛的積木中空白處把內容改為想要的字，例如：你好，這樣程式就大致完成了，是不是很简单呢？

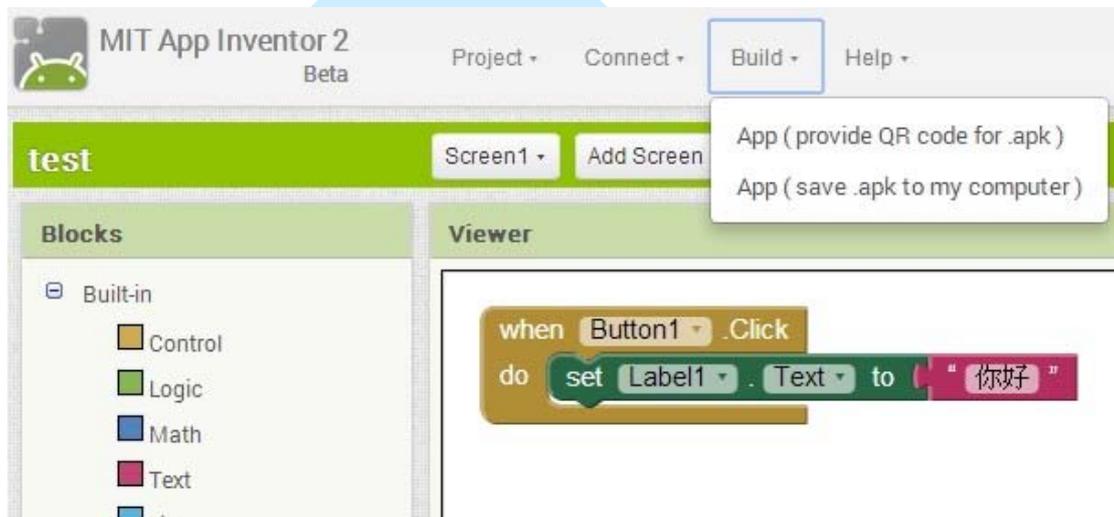


如果剛剛有開喇叭，在積木接合的瞬間會有"喀"一聲，另外當積木拼不進去並彈開時，就代表你邏輯錯誤或拉錯積木啦！在過程中觀察一下會發現判斷式和一些控制指令都是 ，而回傳資訊的積木都是 ，另外還要注意左下角有沒

有錯誤呢，如下圖，黃色驚嘆號是警告，通常都是積木沒拼上去，可以不理它，但是當有紅色驚嘆號出現，就表示程式嚴重錯誤，趕快找到錯誤並修正吧。



程式完成之後，目前有兩種方案可以傳到 Android 裝置，方案一是藉由 WiFi 與電腦同一個區域網路可直接傳檔案，方案二是直接生成 apk 安裝檔，再丟到 Android 裝置中安裝。



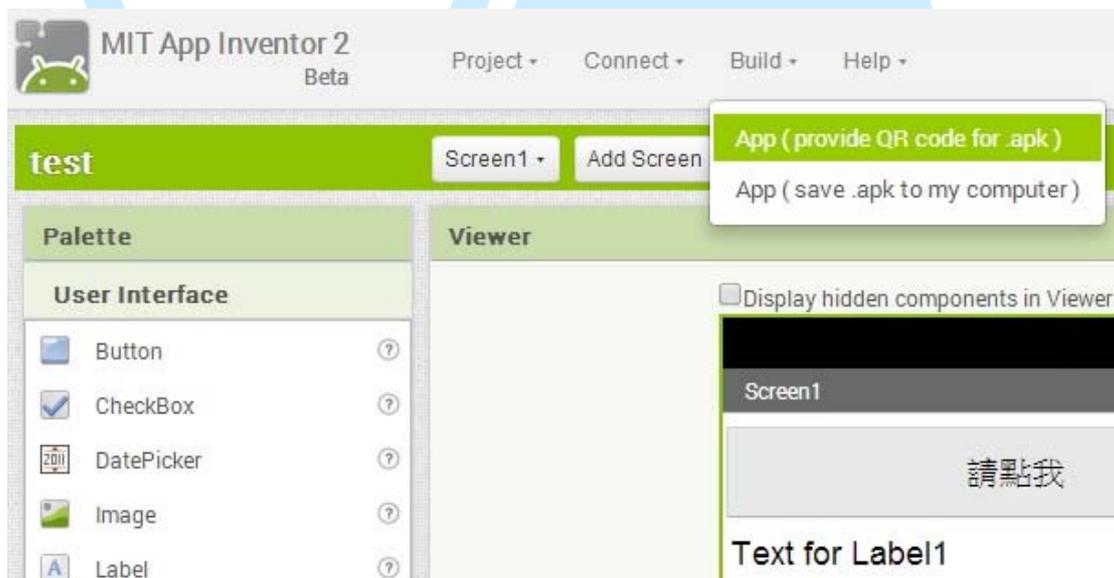
將 APP 傳到裝置之前，先到 設定-> 安全性 把不明來源打勾，如有出現警告請按下確定。



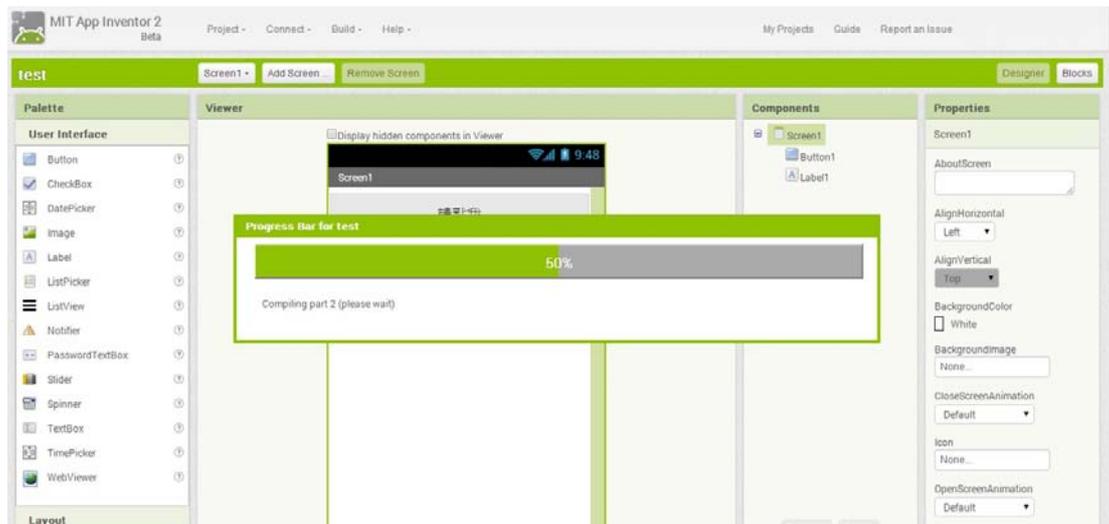
如果選擇方案一，必須先安裝 MIT AI2 Companion 這個 APP，開啟 PLAY 商店搜尋打 MIT 就有關鍵字可以選擇了，如下圖，安裝之後就把它開啟吧。



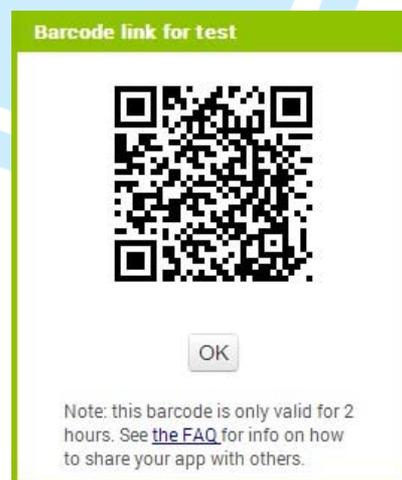
先回到電腦中，點擊 Build-> App(provide QR code for .apk)，如下圖。



接著會跑出進度條，正在編譯 APP 中，如下圖。



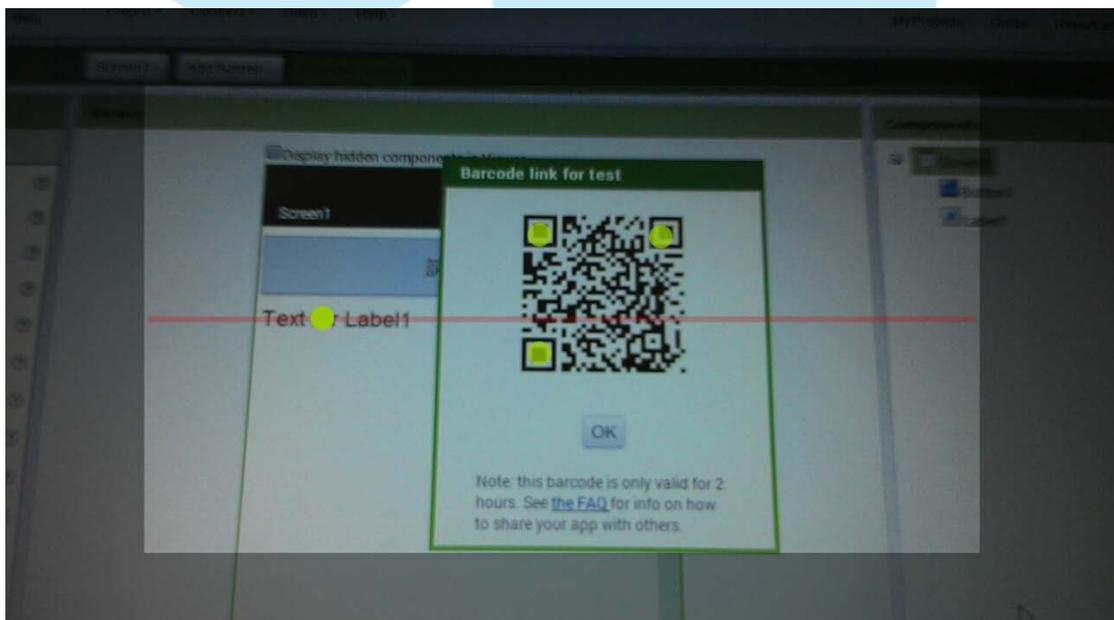
進度條跑完之後就會出現 QRcode 啦，如下圖。



接著就是拿起裝置開啟剛剛安裝的 MIT AI2 Companion 程式啦，如下圖，然後點擊 scan QR code。



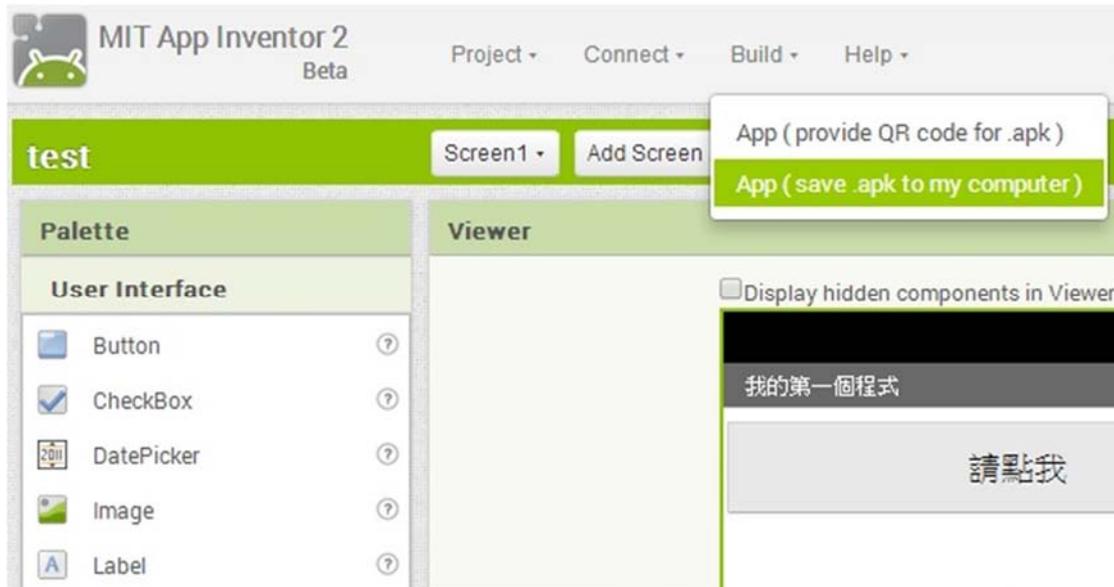
之後對著螢幕掃描，如下圖。



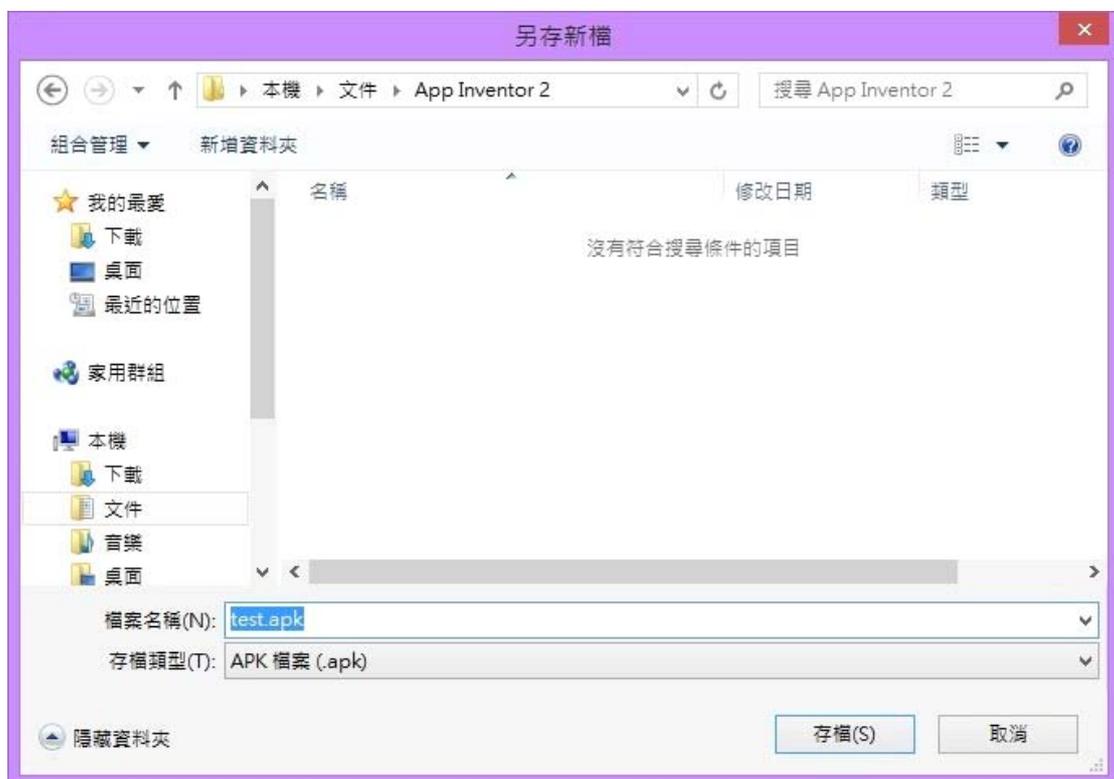
掃完 QR code 後會跳回原來的畫面，這時不要緊張也不要亂按，其實是 APP 正透過網路從電腦傳到手機，需等待一段時間，成功了便會出現程式安裝畫面，如下圖，我們按點擊安裝。



如果選擇方案二，點擊 Build-> App(save .apk to my computer)，如下圖。



進度條跑完之後就會跳出儲存視窗，如下圖，儲存之後再把它傳進裝置做安裝，但裝置必須安裝檔案管理 APP(有些裝置會內建)，瀏覽至檔案目錄並點擊安裝，雖然方案二會比方案一還要費時，但多了安裝檔可以隨時分享給其他人使用，方案一適合程式除錯時使用。



安裝完成後，我們就來驗收結果啦，如下圖，這是點擊 Button 前的樣子。

Screen1

請點我

Text for Label1

點擊 Button 之後，Label 就變成 "你好" 了，如下圖。

Screen1

請點我

你好

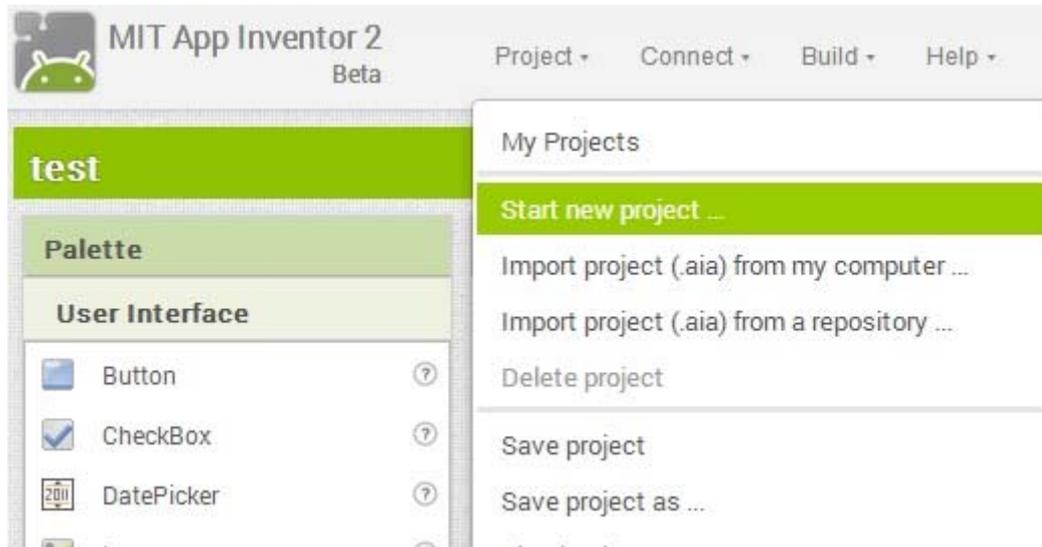
是不是覺得程式沒有很完整呢？例如可以在 Screen1 屬性中的 Title 項，把 "Screen1" 改成 "我的第一個程式"，還有在 Label1 屬性中的 Text 項，把裡面的文字去掉，這樣只有在點擊 button 之後才會出現文字，如下圖，螢幕有沒有乾淨許多呢？

我的第一個程式

請點我

三、第二個程式

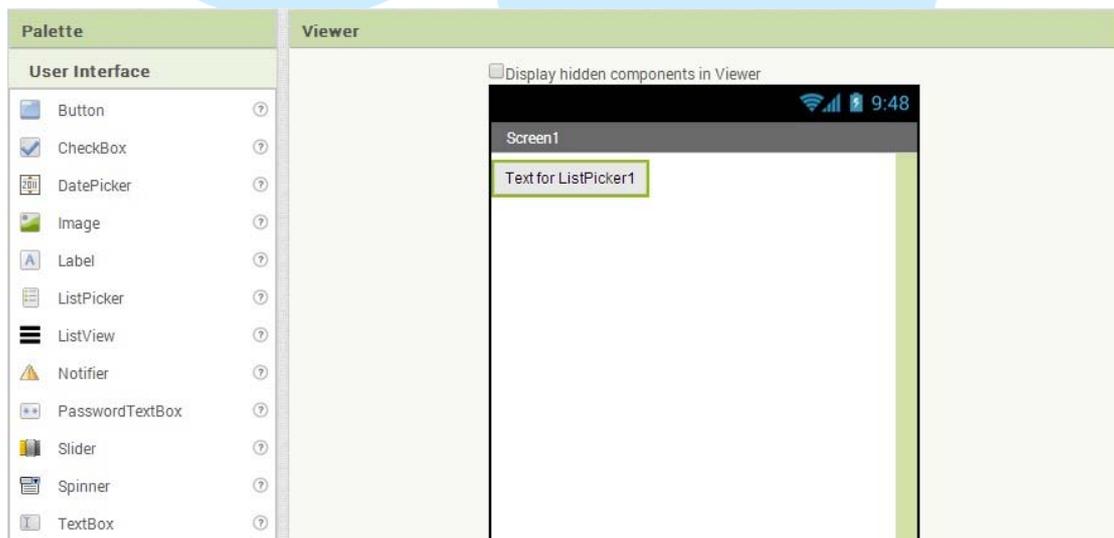
我們來做結合 EduCake 的程式吧，最簡單方便的溝通方式就非藍芽莫屬啦～首先開新專案點擊 Project-> Start new project



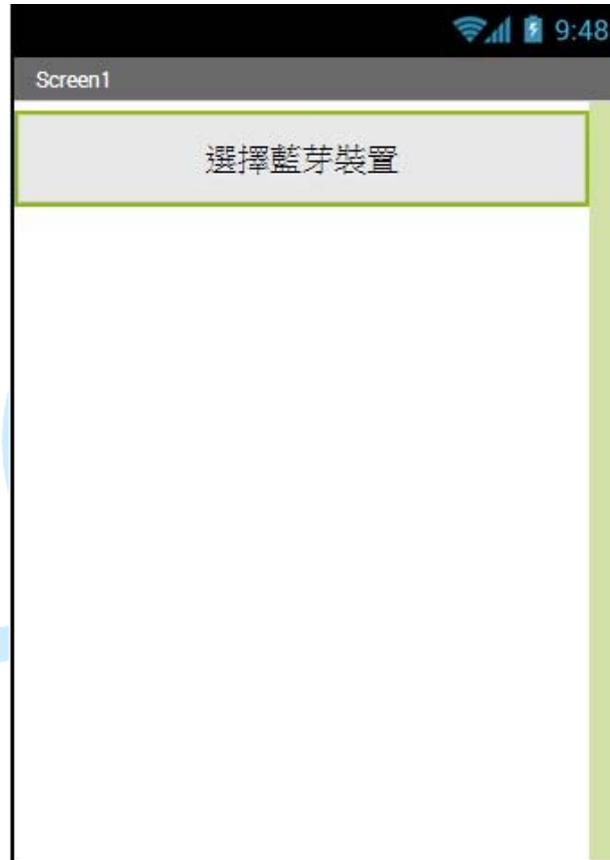
專案名稱就取 "Bluetooth_Connect" 吧。



接著從左側拖曳一個 ListPicker 元件到預覽畫面中。



在 Properties(屬性)欄位中的 FontSize 改為 20 · Text 改為"選擇藍芽裝置" · Width(寬) 改為 Fill parent(充滿) · Height(高) 改為 50 pixels · 改完之後會跟下圖一樣。



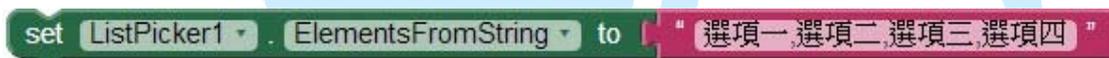
這裡先接紹一下 ListPicker 的功能，它功能有點像 Button，不過它是具有選單的 Button，點擊它就會跳出我們指定的選項，如下圖，然後我們可以針對點擊其中的選項再做後續的動作。



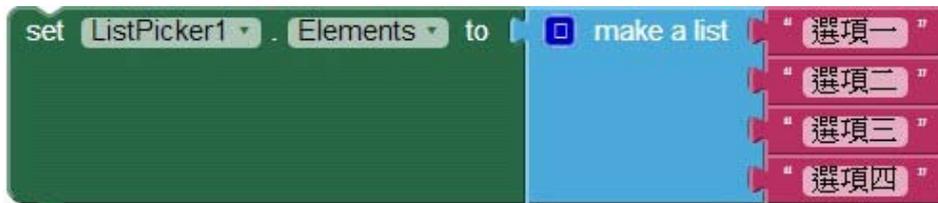
選單設定的方法有兩種，一種是在 Properties(屬性)欄位中的 ElementsFromStrings 裡來新增選項，如下圖，選項跟選項之間用半形逗號隔開即可，使用的前提是有預設的選項才能預先新增，這裡的步驟可以不用做，要測試也是可以的。



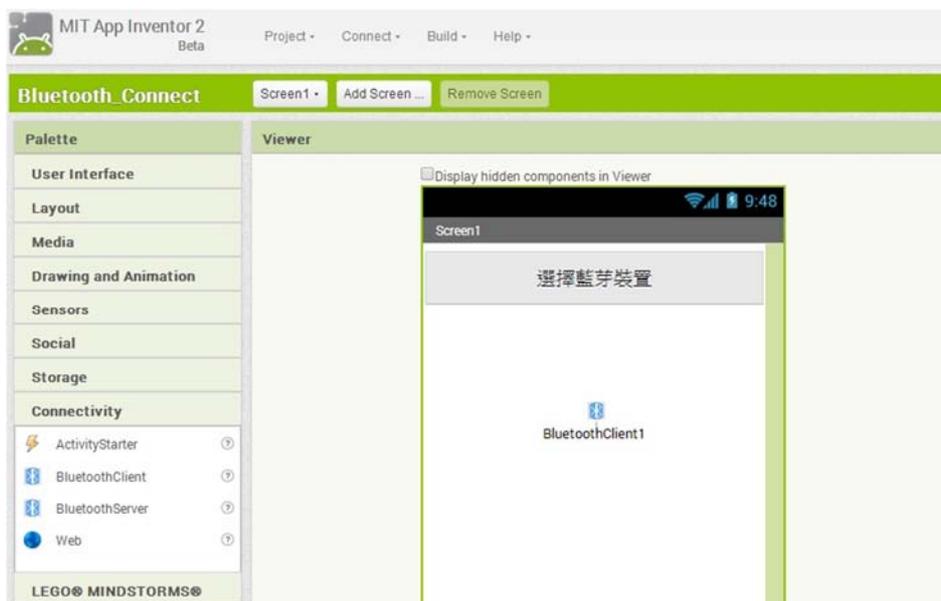
方法二，如下圖，可用在程式中指定選項，彈性較高。



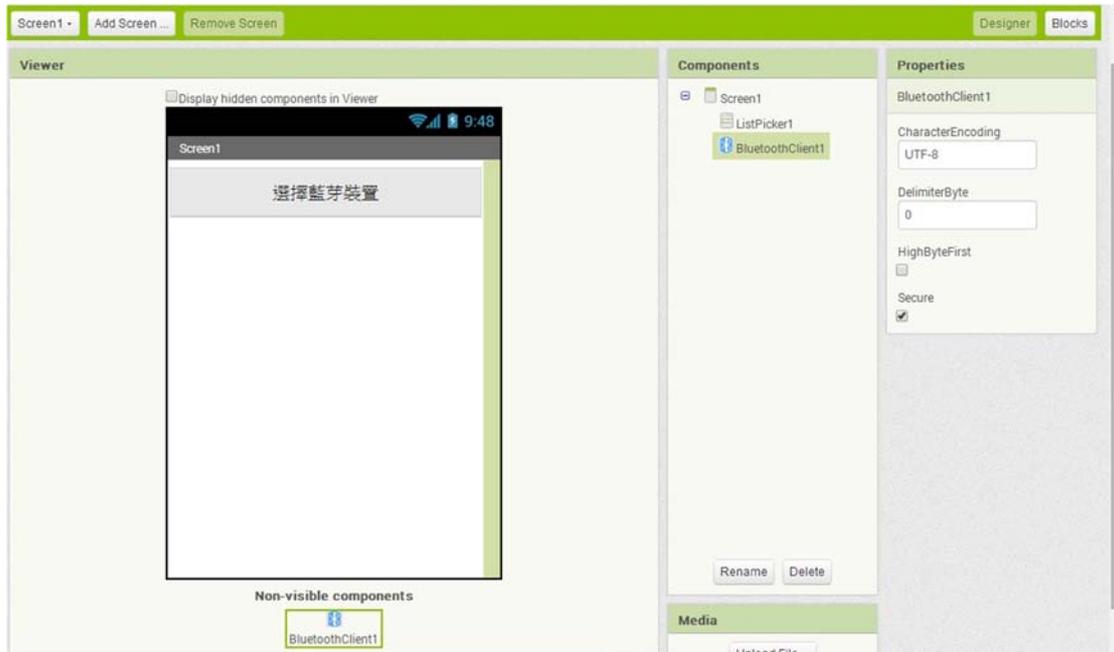
方法二的另一種表示法，一切以寫程式方便就好。



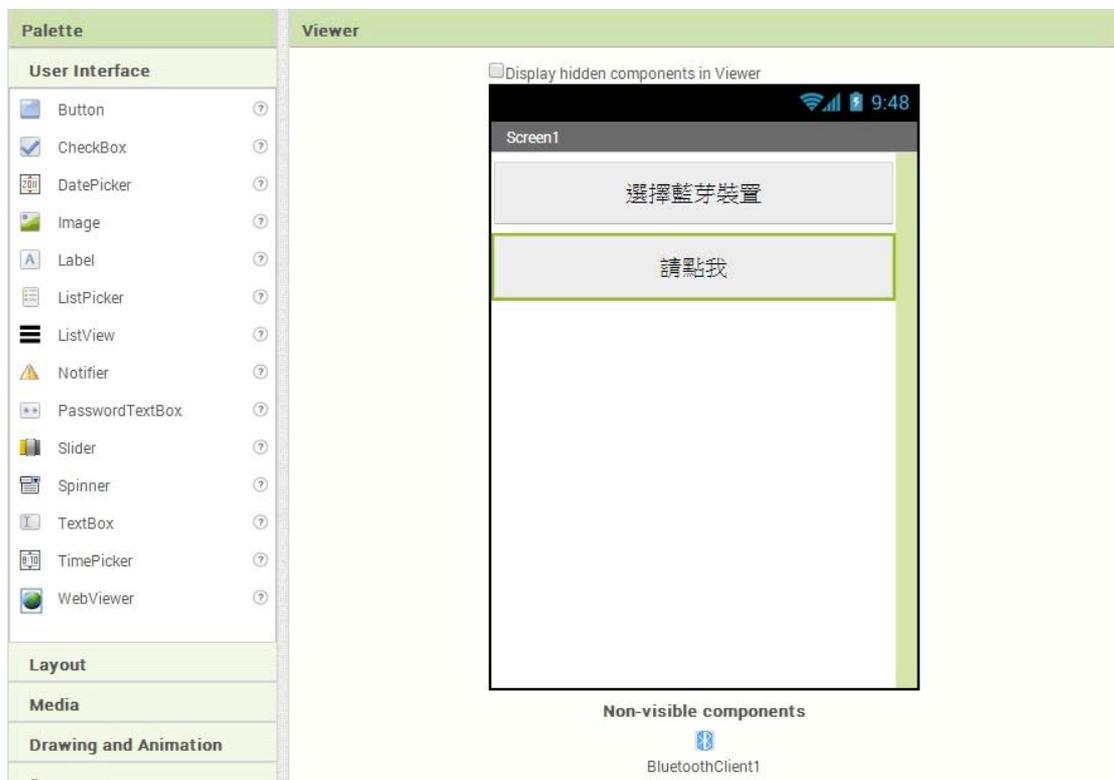
接下來進入正題，到 Palette(元件區)，點擊 Connectivity 會跳出元件，接著把 BluetoothClient 拖曳到預覽螢幕內，如下圖。



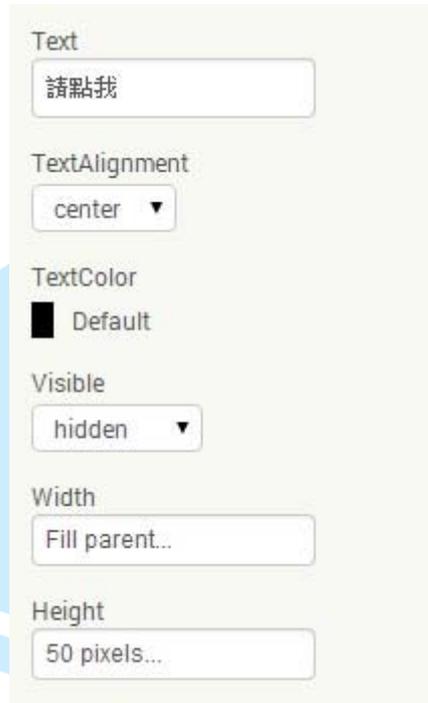
拉進去之後，因為沒有實體，所以只會顯示在預覽螢幕的下面，也不需要更改屬性。



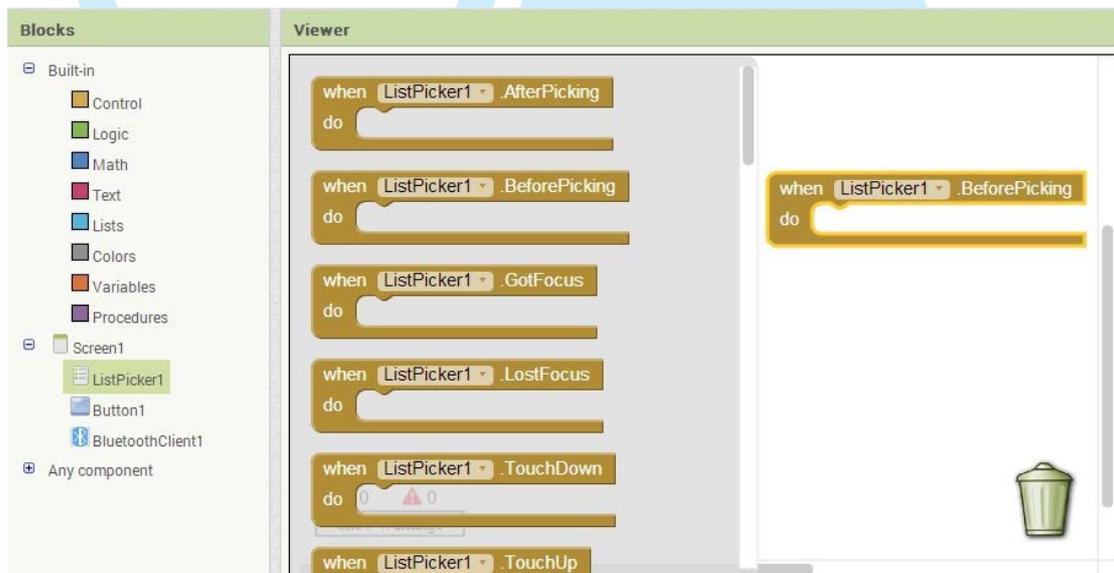
接下來從左側拖曳一個 Button 元件到預覽畫面中，Properties(屬性)欄位中的 FontSize 改為 20，Text 改為"請點我"，Width(寬) 改為 Fill parent(充滿)，Height(高) 改為 50 pixels，改完之後會跟下圖一樣。



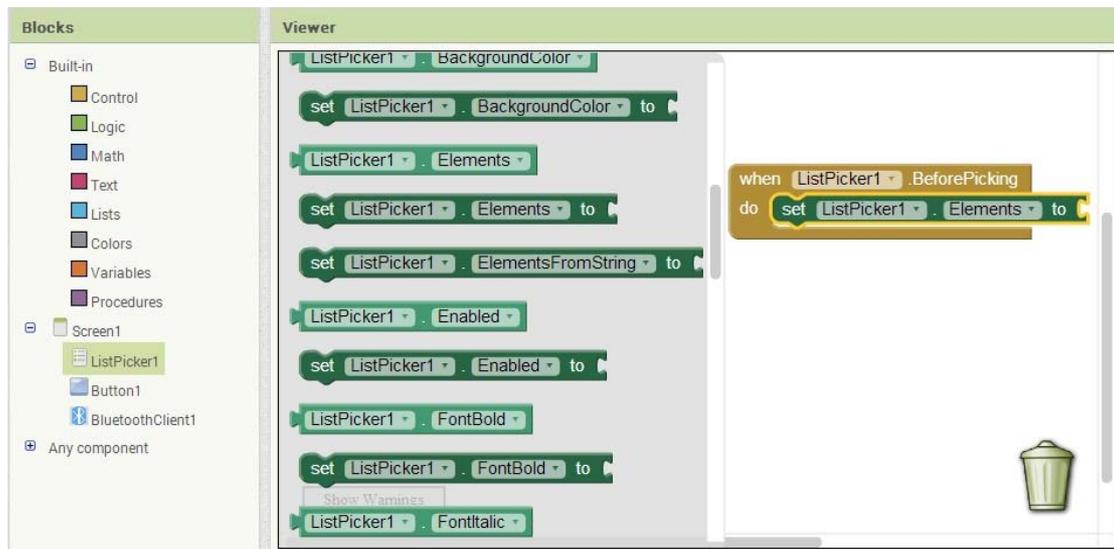
在 Button1 的 Properties(屬性)欄位中 Visible 改為 hidden(隱藏)·用意是執行 APP 的時候先隱藏·等到藍芽連線之後才做顯現·接下來點擊 Blocks 到積木區吧。



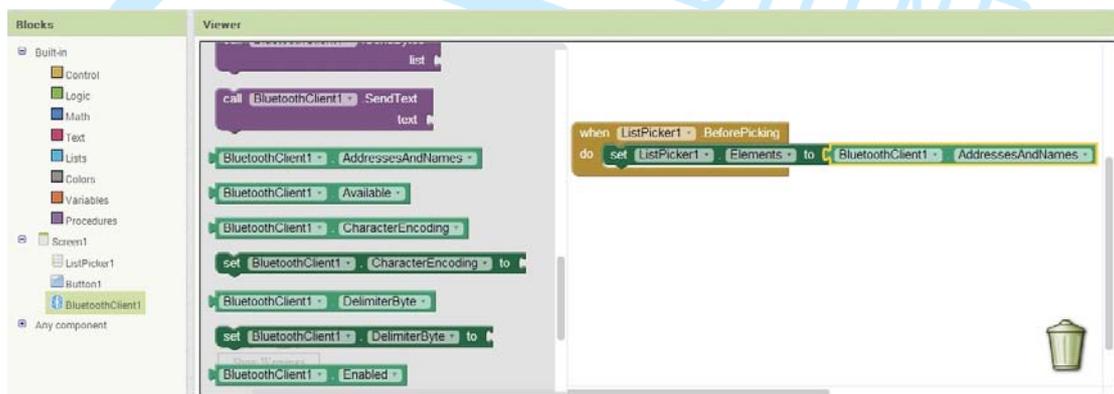
左側點擊 ListPicker1 出現控制項·拖曳 BeforePicking 控制項出來·如下圖·它的觸發條件是點擊 ListPicker"前"會觸發·我們可以派它做一些事情·例如: 更改或新增 ListPicker 的選單。



再次點擊 ListPicker1 出現控制項，滾動條往下拉才看的到 Elements 控制項，把它拖曳出來組合進去，如下圖。



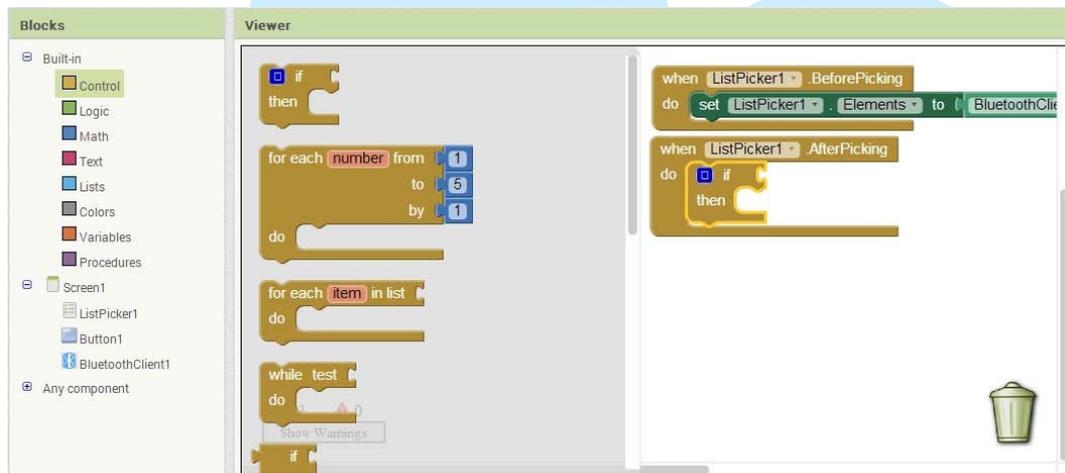
左側點擊 BluetoothClient1 出現控制項，一樣將滾動條往下拉，看到 AddressesAndNames，把它拖曳到右側接上，如下圖，功能是将 Android 所配對過的藍芽設備用 list 的方式列出來，我們這樣做可以在程式剛開始執行的時候觸發點擊 ListPicker1 前的控制項，然後做新增藍芽清單的選項。



左側點擊 ListPicker1 出現控制項，拖曳 AfterPicking 控制項出來，如下圖，它的觸發條件是點擊 ListPicker"後"就會觸發 AfterPicking 控制項。



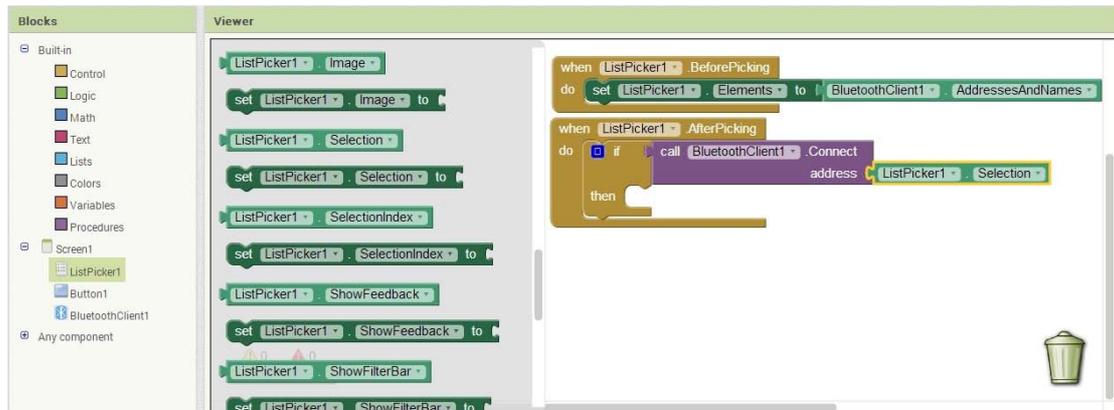
左側點擊 Control(控制)·把 if 判斷式拖曳至右側接合·如下圖。



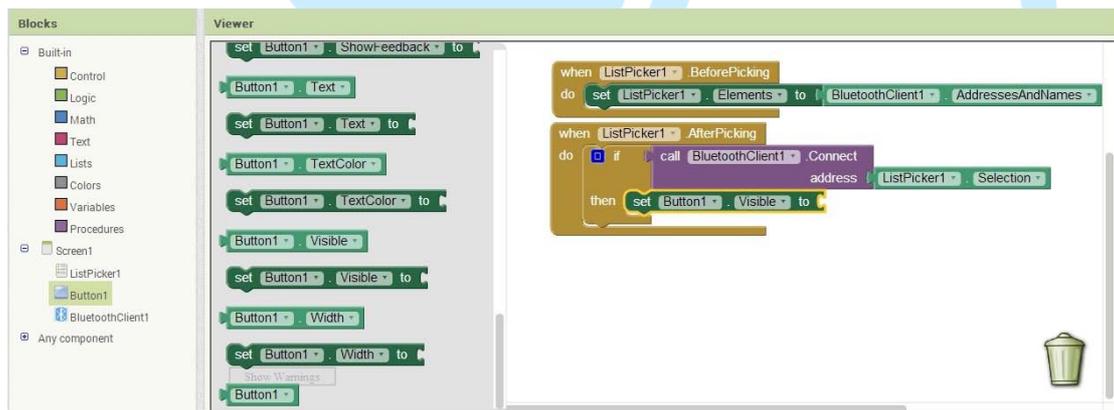
左側點擊 BluetoothClient1·將 Connect address 控制項拖曳到右側跟判斷式接合·如下圖·功能是給它藍芽的位址就可以連線·成功回傳 true·失敗回傳 false。



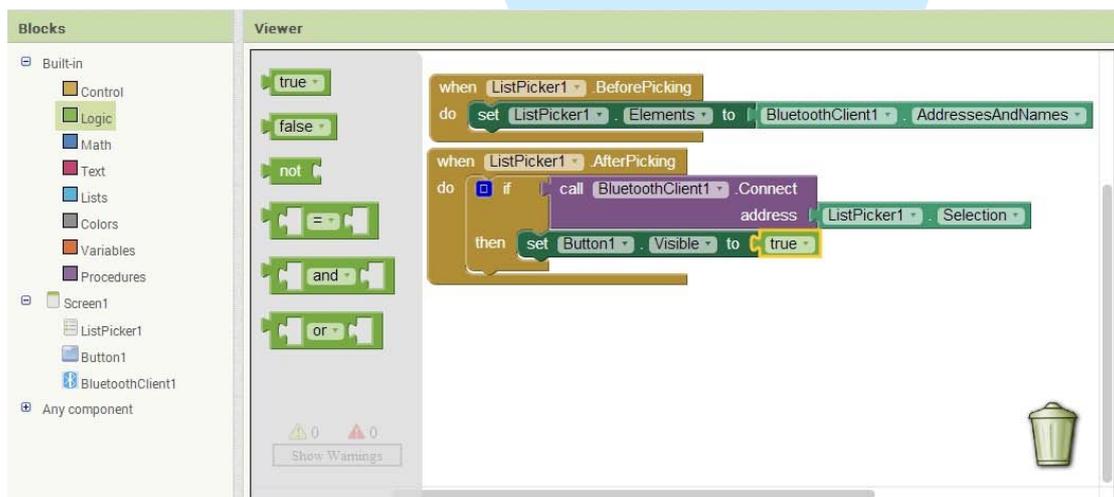
左側點擊 ListPicker1，將 Selection 拖曳至右側結合，如下圖，它的功能是回傳 ListPicker1 選單中我們選擇的文字。



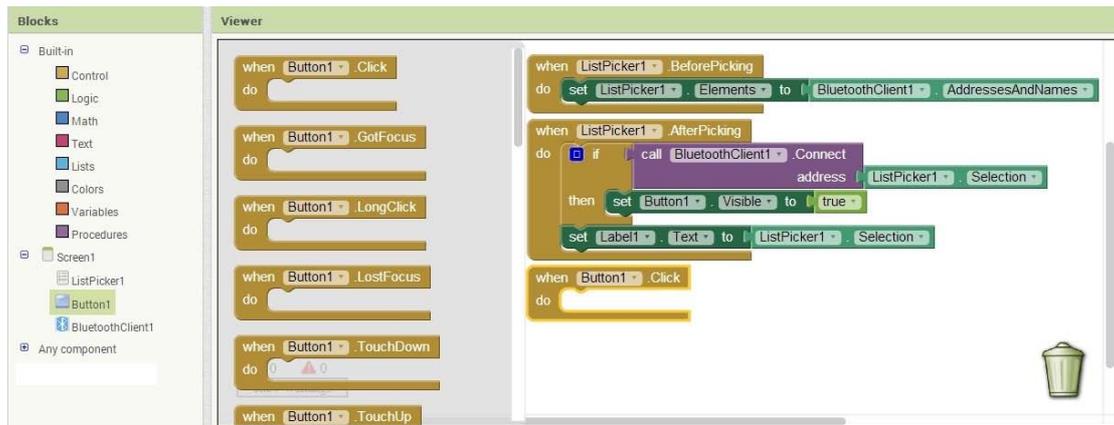
左側點擊 Button1，將 .Visible 控制項拖曳到右側接合，如下圖，因為我們剛開始是把 Button1 做隱藏，當藍芽連線成功會回傳 true 去處發 if 的條件，我們就可以把按鈕顯現出來了。



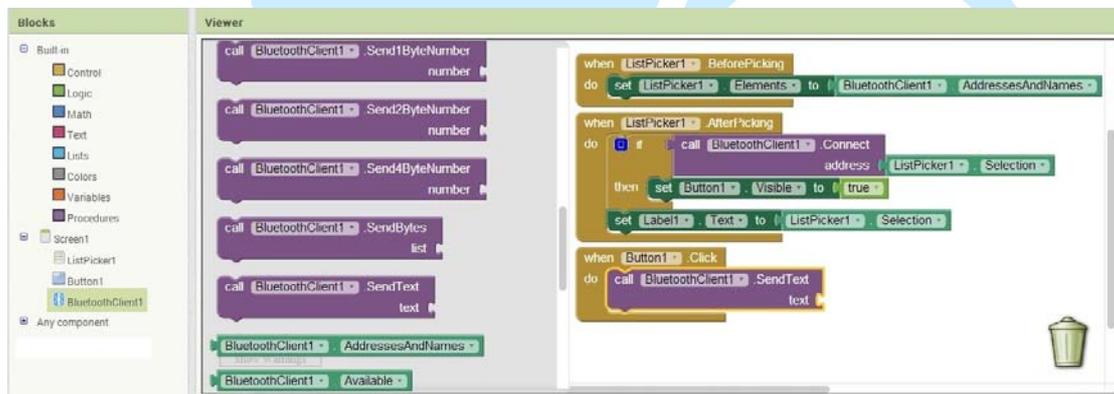
接著點擊 Logic，拖曳 true 到右側接合，如下圖。



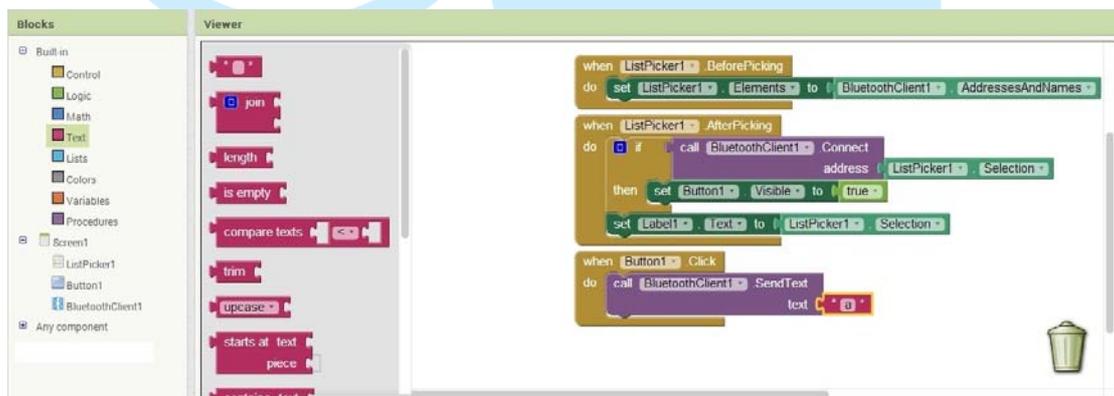
側點擊 Button1，將 .Click 拖曳到右側，如下圖。



接下來點擊 BluetoothClient1，將 .SendText text 拖曳至右側接合，如下圖，它的功能是可以傳字串給連線中的藍芽裝置。

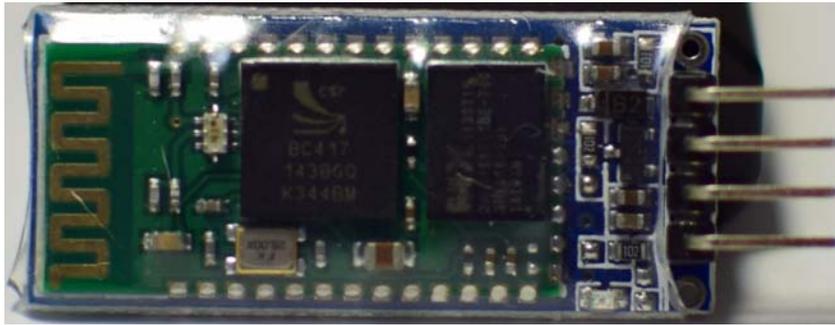


接下來點擊 Text，我們拖曳一個空白字串到右側，並在裡面輸入"a"，如下圖，這樣每按一次 Button 就會傳"a"的字元給連線中的藍芽裝置。



接下來介紹 EduCake 藍芽模組的部分，這裡使用的藍芽模組是較為大眾化的: HC-06。

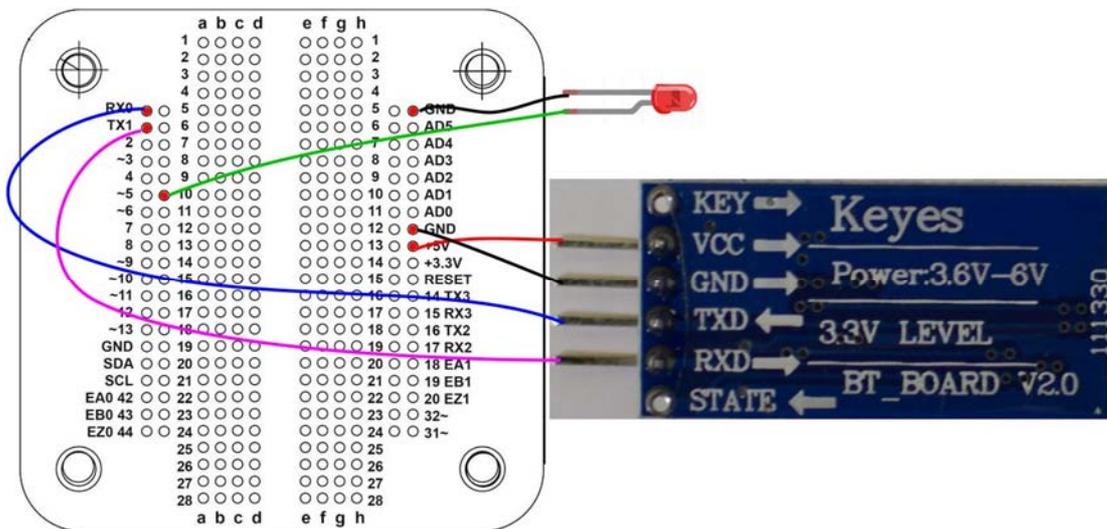
HC-06 正面



HC-06 反面



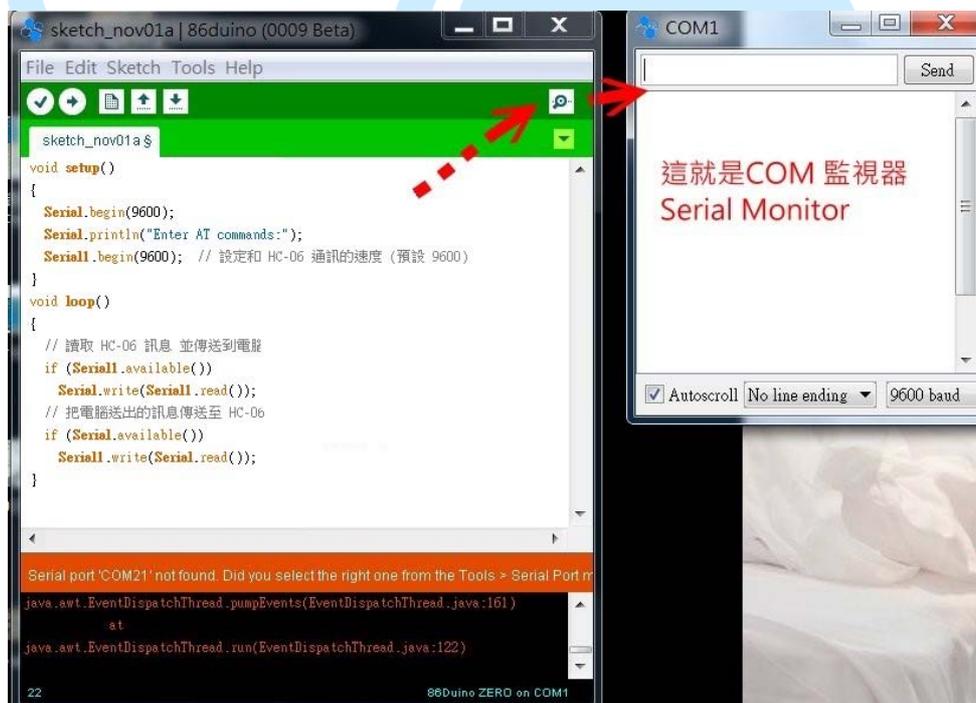
將 HC-06 接上 EduCake，再接上一顆 LED，接腳圖如下：



接好電路以後，我們先來設定 HC-06 的一些參數

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  Serial1.begin(9600); // 設定和 HC-06 通訊的速度 (預設 9600)
}
void loop()
{
  // 讀取 HC-06 訊息 並傳送到電腦中
  if (Serial1.available())
    Serial.write(Serial1.read());
  // 把電腦送出的訊息傳送至 HC-06
  if (Serial.available())
    Serial1.write(Serial.read());
}
```

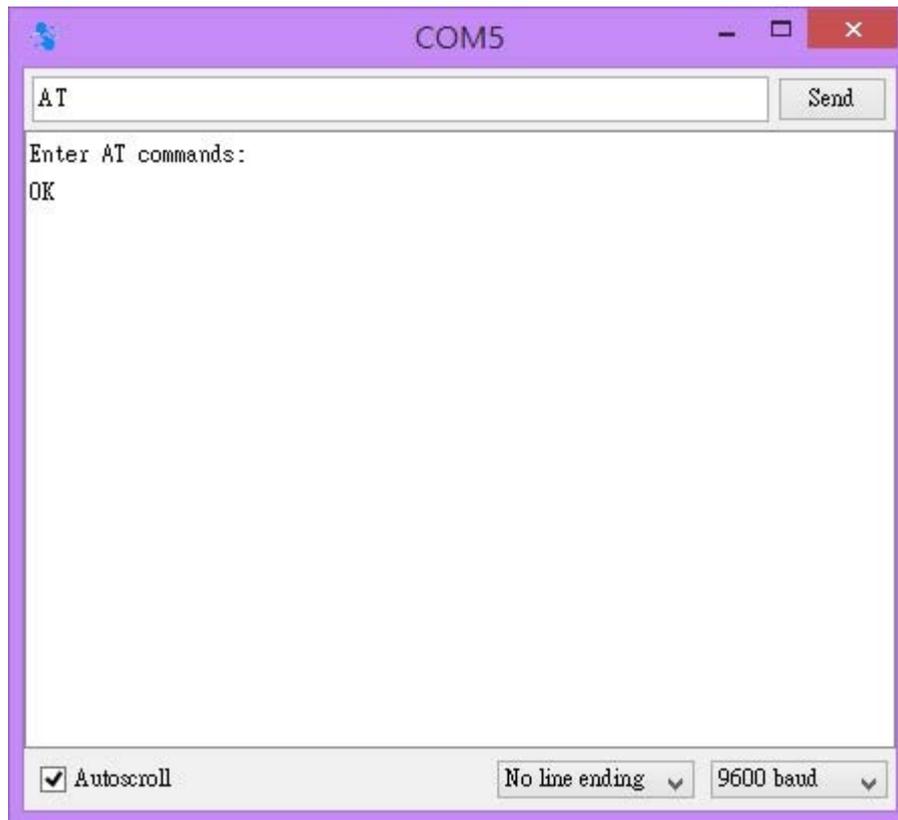
首先開啟 IDE 介面，輸入以下程式。程式上傳好了以後，就可以利用 IDE 介面裡面的 COM 監視器(Serial Monitor)來和藍芽通訊，畫面如下



到底要輸入什麼哩？其實 HC-06 在通電未連線情況下，燈光會閃爍並處於 AT command mode，當在這個狀態的時候，我們就可以任意更改名稱、密碼，甚至連通訊速度(baud rate)都可以改，以下介紹幾個常用的 AT 指令：

- AT : 測試是否正常工作，回應「OK」
- AT+NAMEaaa : 將裝置名稱改為「aaa」
- AT+PIN1234 : 將連線密碼換為「1234」
- AT+VERSION : 回應韌體的版本
- AT+BAUD1 : 將 baud rate 換為 1200
- AT+BAUD2 : 將 baud rate 換為 2400
- AT+BAUD3 : 將 baud rate 換為 4800
- AT+BAUD4 : 將 baud rate 換為 9600
- AT+BAUD5 : 將 baud rate 換為 19200
- AT+BAUD6 : 將 baud rate 換為 38400
- AT+BAUD7 : 將 baud rate 換為 57600

上面前三個 AT 指令比較常用到，如果要改名稱為 abc，就輸入 AT+NAMEabc，成功會顯示 OKsetname，設定密碼為 1234，就輸入 AT+PIN1234，成功會顯示 OKsetPIN，這裡要注意的是更改名稱之後要斷電重開才會生效而且不能使用中文(會亂碼)，而改完密碼不需重開即生效。



剛剛做出藍芽連線的 APP，再加上設定完 HC-06，終於要進入正題啦，我們要做出用手機控制一個 LED 燈，程式碼如下：

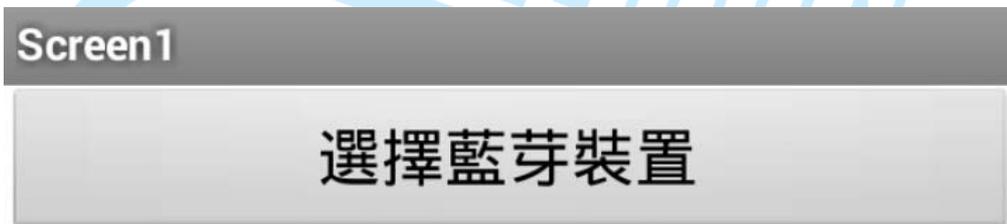
```
char ch;
int LED=0; // 儲存 LED 目前狀態
void setup()
{
    Serial.begin(9600); // 這行主要是設定和電腦的 COM 通訊
    的速度
    Serial1.begin(9600); // 這行主要是設定和藍芽 通訊的速度
    pinMode(5, OUTPUT); // 設定腳位 3 為輸出模式
}

void loop()
{
    if (Serial1.available()) // 檢查電腦端是否有訊息來
    {
        ch = Serial1.read(); // 若有訊息, 一次讀取一個 byte
        if (ch == 'a') // 判斷是否為 a 的訊息
        {
            if (LED==LOW) //判斷腳位 5 的燈是否有亮(預設是
            LOW)
            {
                digitalWrite(5, HIGH); // 點亮腳位 5 的燈
                LED=HIGH; // 變更 LED 狀態為 ON
            }
            else
            {
                digitalWrite(5, LOW);
                LED=LOW;
            }
        }
    }
}
```

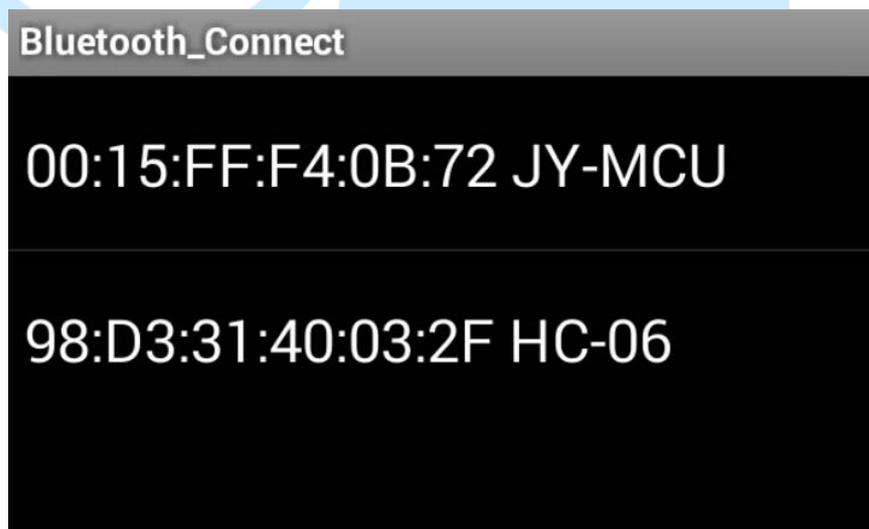
程式上傳之後，我們轉移陣地到 Android 裝置中吧，首先確認裝置已經跟 HC-06 配對完成，如下圖，本書以 HC-06 作為藍芽名稱方便辨識。



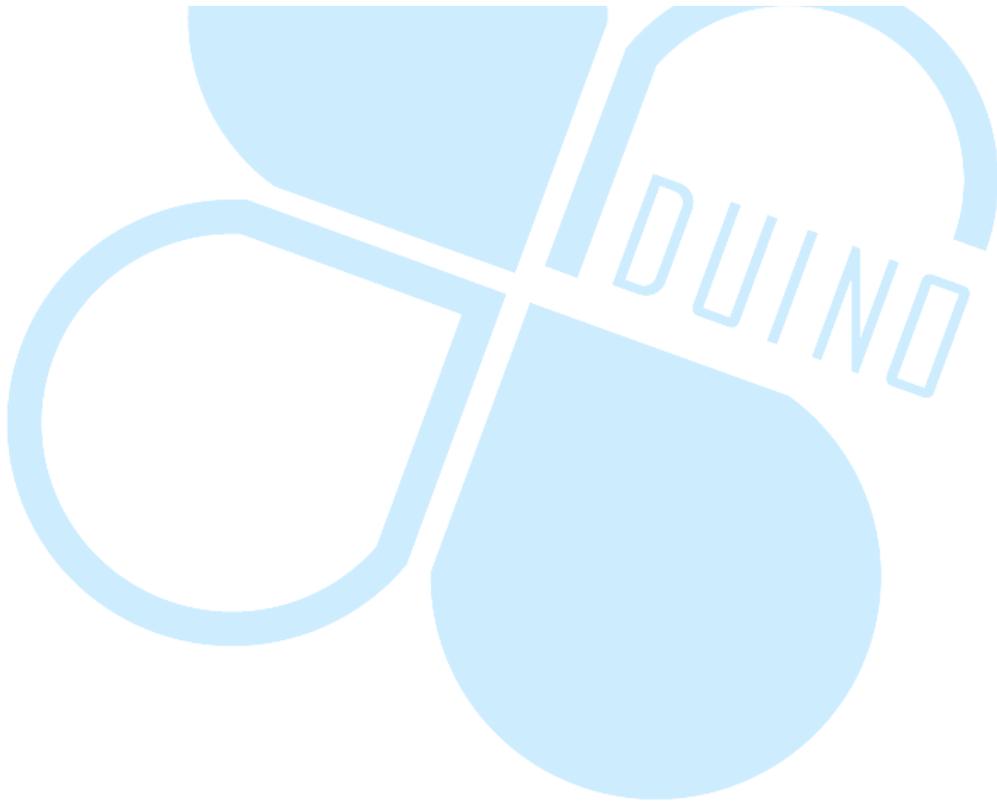
接著開啟程式，如下圖，點擊"選擇藍芽裝置"。



跳出藍芽選單，如下圖，請選擇 HC-06 藍芽模組連線。

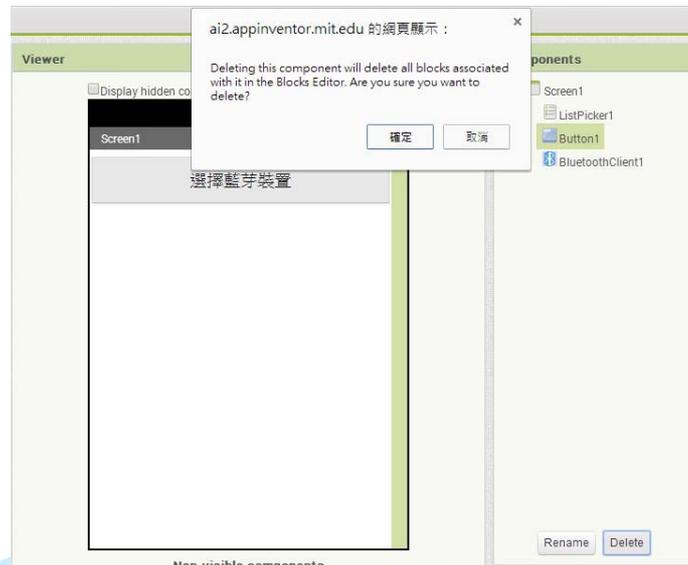


連線成功後會出現"請點我"的 Button，如下圖，可以觀察到 HC-06 從原本的閃爍變化為恆亮，接下來點擊"請點我"，EduCake 上的 LED 燈是否亮起來了呢？再點一下就會熄滅了。

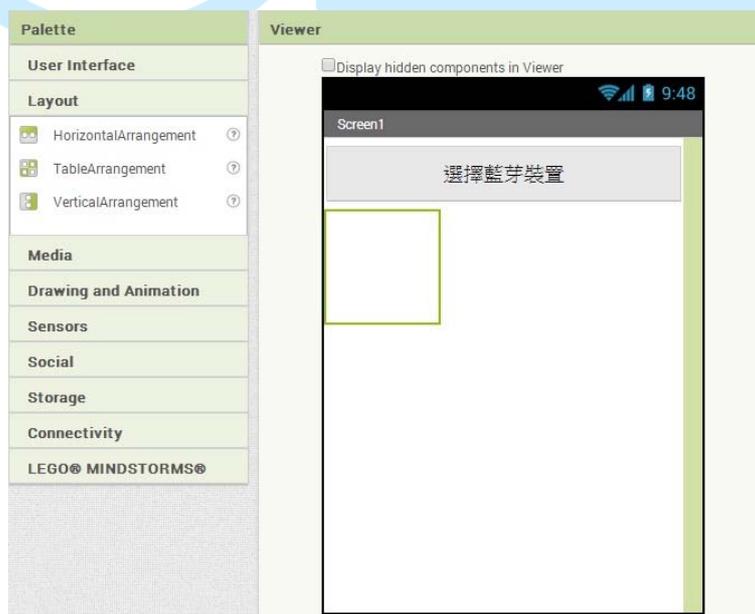
Screen1**選擇藍芽裝置****請點我**

四、第三個程式

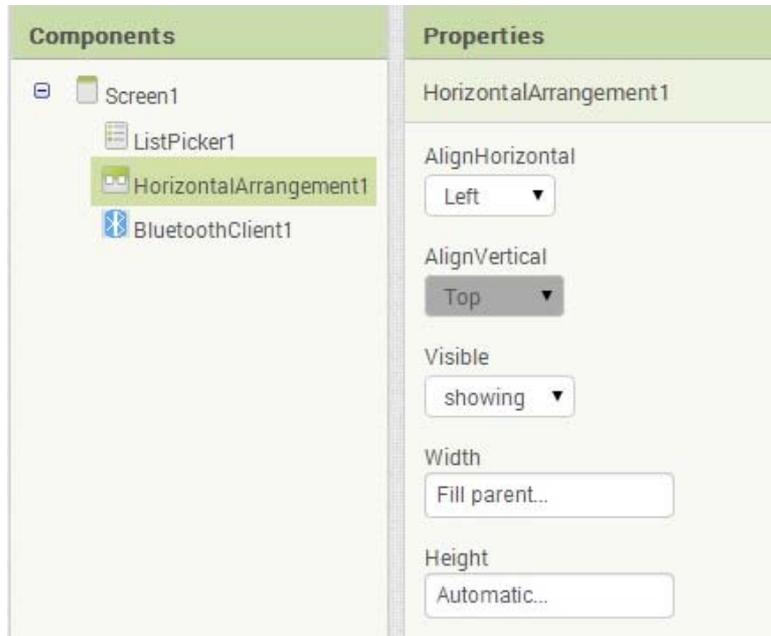
我們再度回到 App Inventor 中，延續剛剛的程式繼續修改，如果想保留剛剛的作品，可以利用 Project-> Save project as ... 來另存一個新專案做編輯，接著刪除原本的 Button1，如下圖。



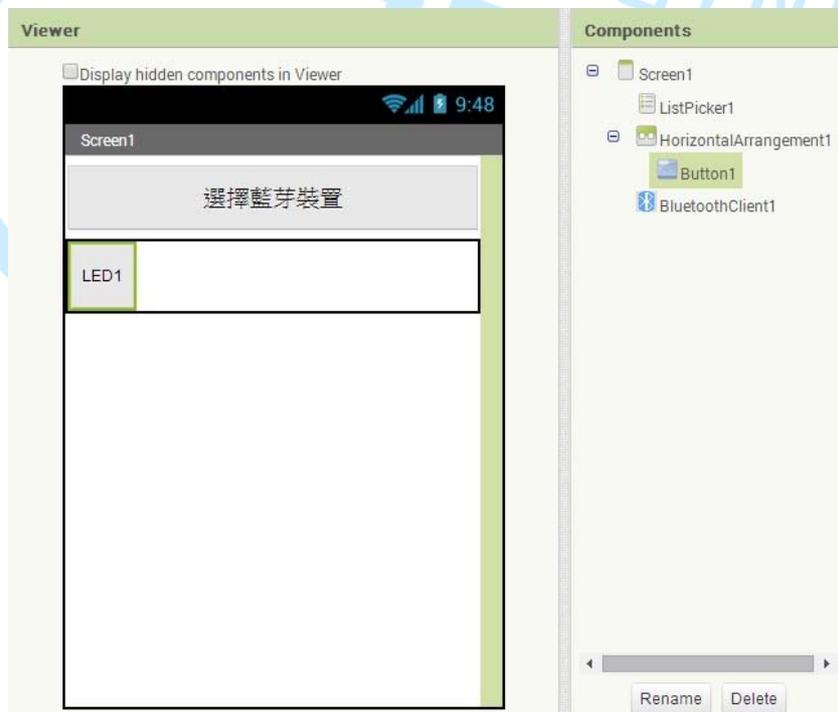
在剛剛的程式中只有控制一顆 LED，這次我們要一次控制五顆，並在 EduCake 上加入 3 顆按鈕，當想要把五顆按鍵擺成橫的卻沒辦法，這時候就需要用到 HorizontalArrangement，在 Layout 中，將它拖曳至預覽螢幕中，如下圖，這個元件的概念有點像圖層管理，在框框裡面的原件就只能在裡面活動，並隨著 HorizontalArrangement 隱藏而隱藏，是個方便管理界面的好工具。



在 HorizontalArrangement 的屬性中，將 Width(寬)改為 Fill parent，如下圖。



把 Button 拖曳到剛剛的框框內，在屬性中，將 Text 改為 LED1，Height(高)改成 50 pixels，如下圖。



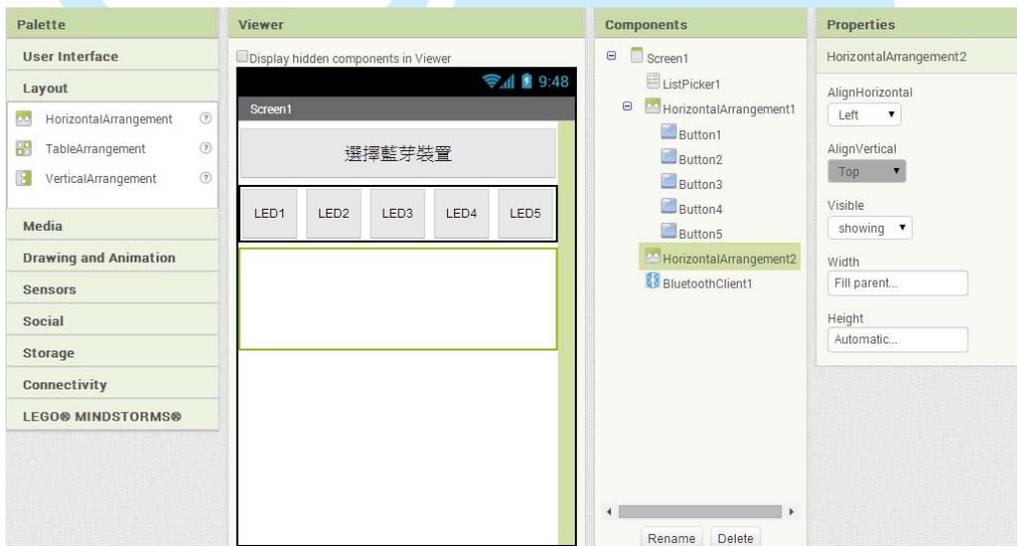
跟剛剛一樣的動作，再拉四顆按鈕出來，屬性的部分除了名字之外其他都相同，如下圖。



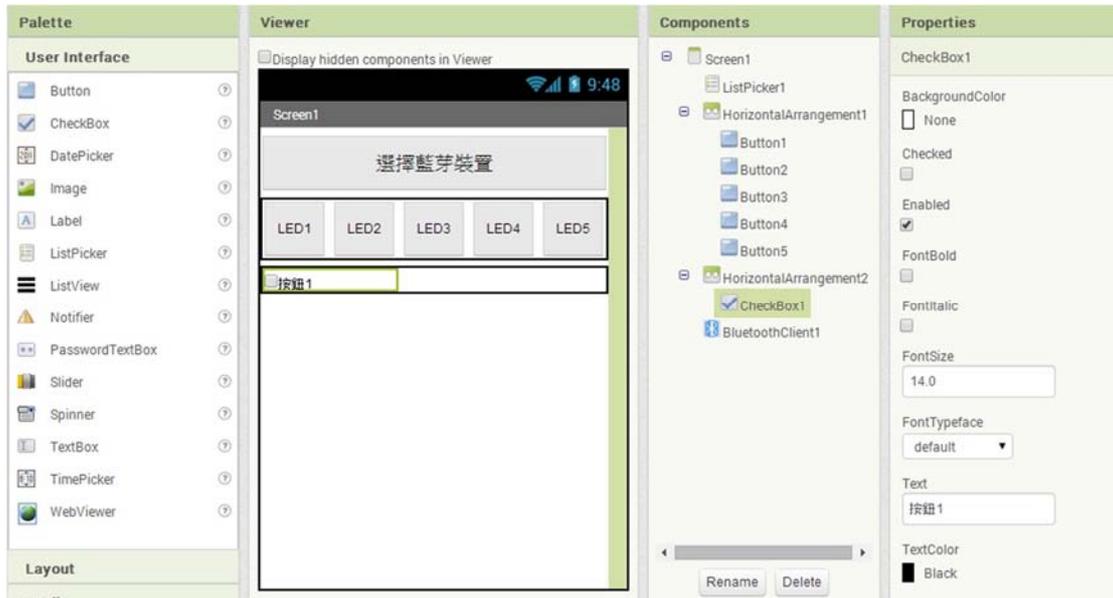
感覺按鈕是不是沒有排正呢？再點擊每個 button，在屬性中的 Width(寬)改為 Fill parent，改完之後就會如下圖一樣。



接下來再拖曳一個 HorizontalArrangement 到預覽螢幕中，並在屬性中的 Width(寬)改為 Fill parent，如下圖。



在左側點擊 User Interface 接著拖曳一個 CheckBox 到剛剛的框框中，在屬性中將 Text 改為按鈕 1，如下圖，它的功能是左側的框框可以打勾也可以取消勾選，我們這裡是用來要觀看 EduCake 的按鈕是否有被按下。



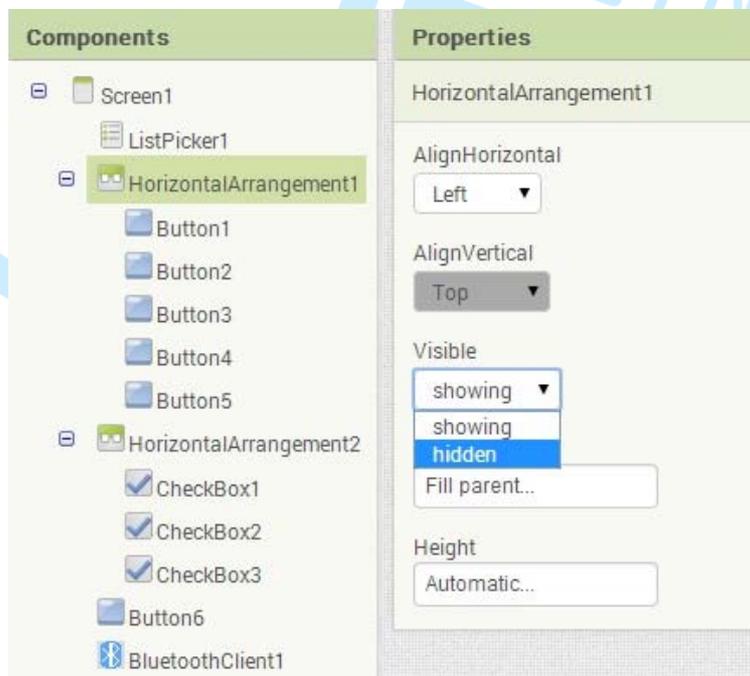
繼續剛才的動作再拖曳兩個 CheckBox，屬性中的 Text 分別改為按鈕 2 及按鈕 3，最後再將三個 CheckBox 屬性中的 Width(寬)都改為 Fill parent，如下圖，此動作是為了要讓元件與畫面對齊。



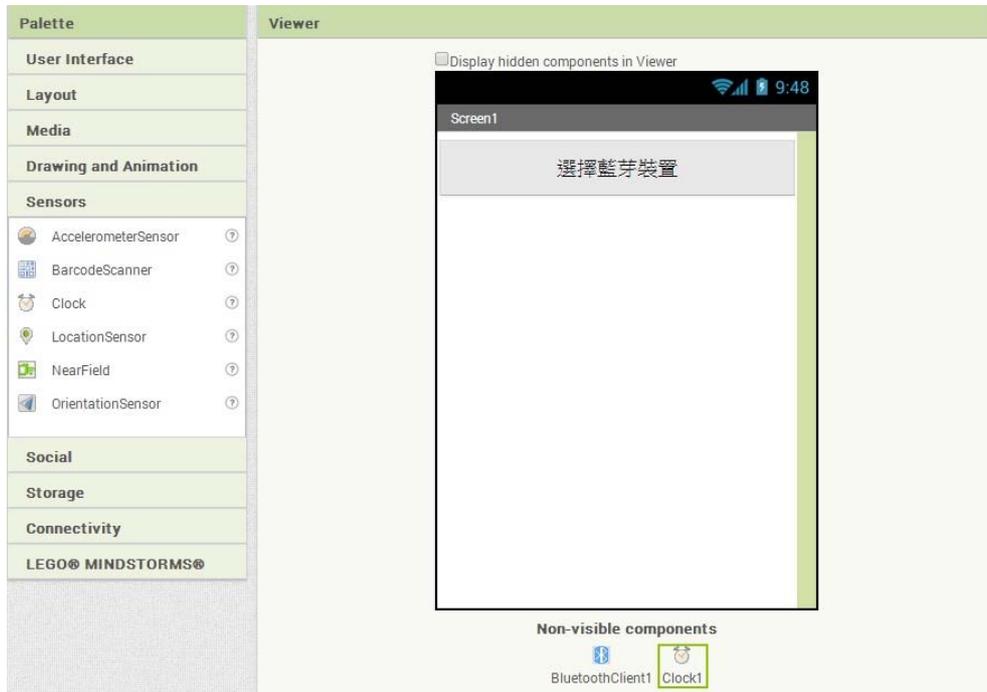
之後再拉一個 Button 到預覽螢幕中，Text 改為中斷連線，Width(寬) 改為 Fill paren，如下圖。



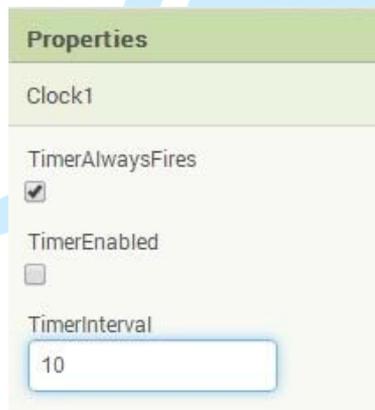
因為藍芽在連線之前除了連線的 button 顯示之外，其他的元件都要隱藏起來，所以我們將 HorizontalArrangement1、HorizontalArrangement2 及 Button6 都隱藏起來，將屬性中的 Visible 改為 hidden，如下圖。



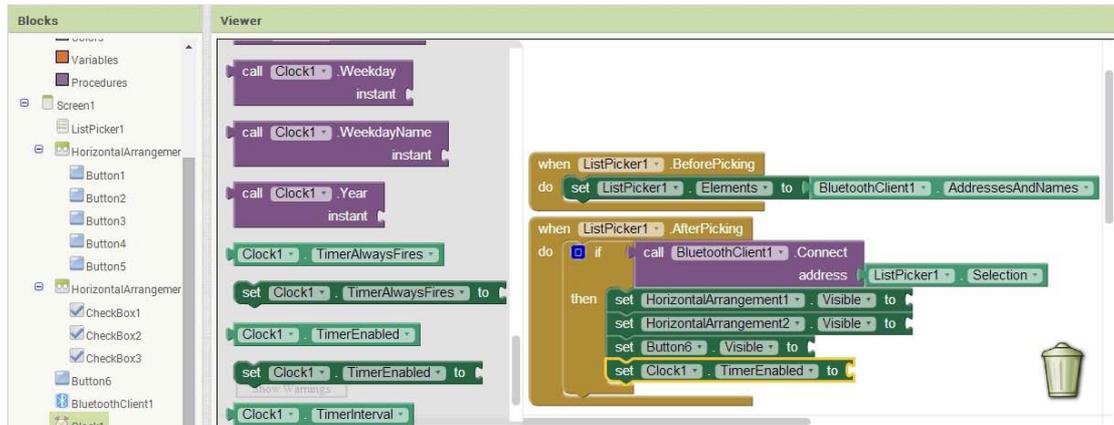
接著在左側 Sensors 底下有 Clock 元件，將他拖曳至預覽螢幕內，因為它也是沒有實體所以只能顯示在下面。



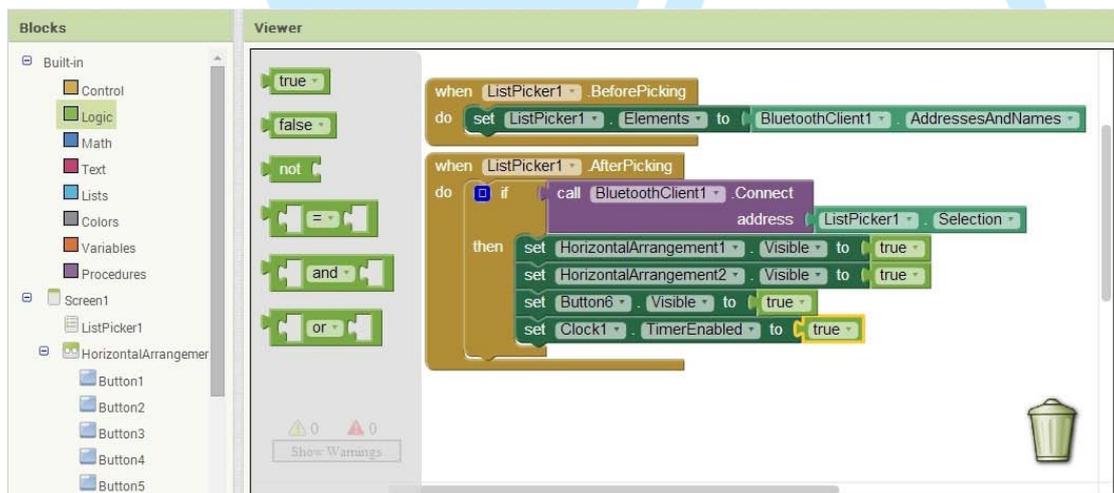
將屬性中的 TimerEnabled 取消勾選，我們不讓他在開始就自動執行，TimerInterval 是跑完一次之後的時間間隔，我們將它改為 10，單位為毫秒，如下圖，接下來點擊 Blocks 到積木區吧。



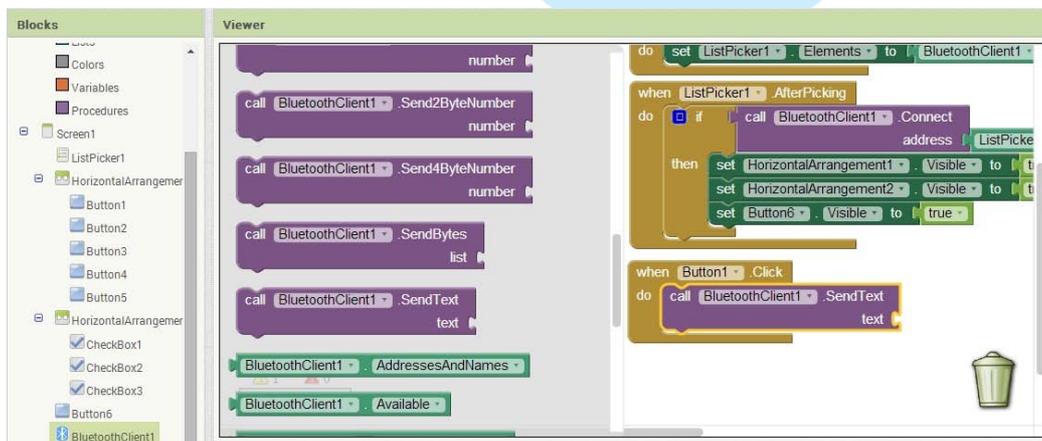
跟上一個程式一樣，我們連線之後要顯現剛剛隱藏的元件，所以我們把 HorizontalArrangement1、HorizontalArrangement2 及 Button6 的.Visible 控制項拉出來，而 Clock1 是拉.TimerEnabled 出來，如下圖。



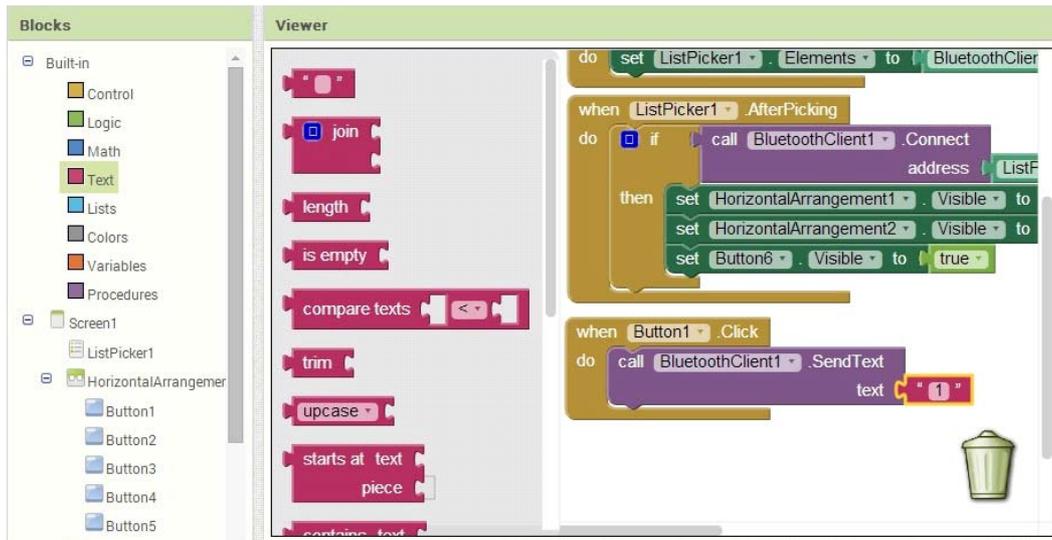
從左側 Logic 中拖曳三個 true 過去接合，如下圖。



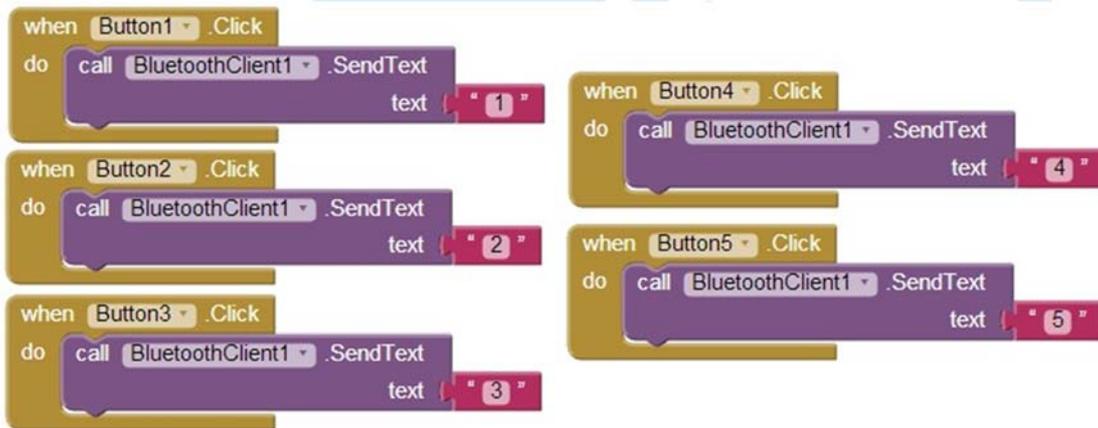
接下來要製作用五個按鈕去控制 EduCake 上的 5 顆 LED，我們先拖曳 Button1 .Click 出來再拖曳藍芽的 .SendText 出來接合，如下圖。



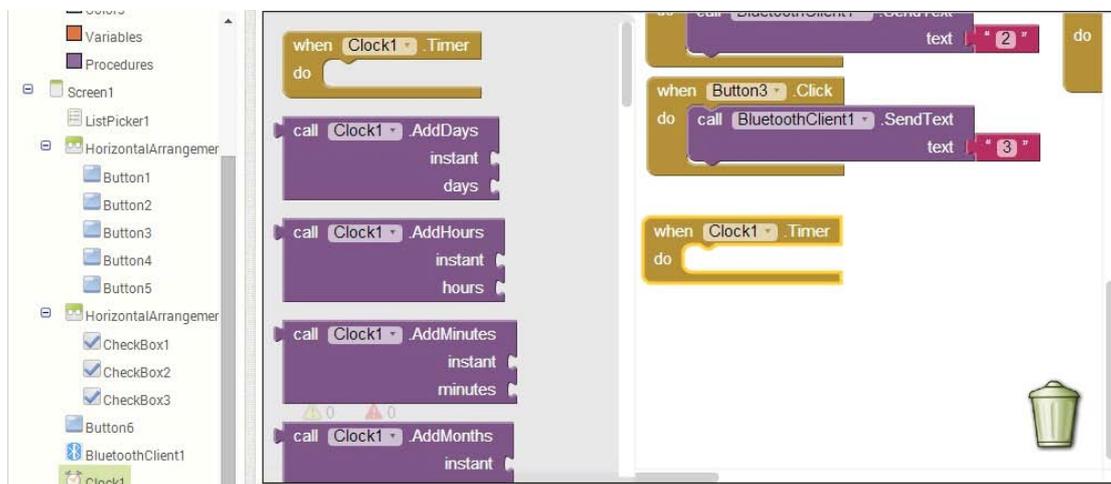
從左側 Text 中拖曳空白的 text 過去接合，並在裡面輸入 1，如下圖。



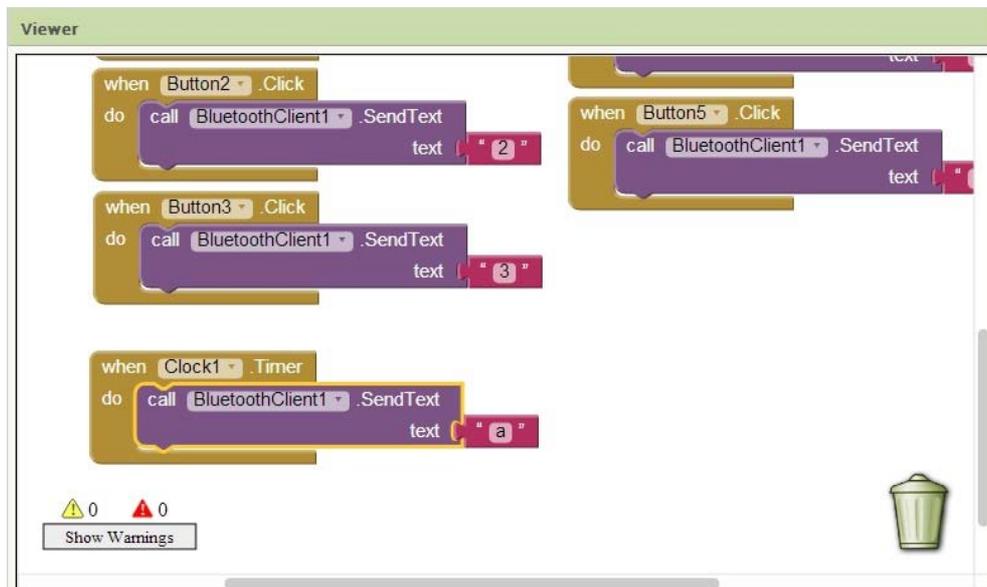
接下來就如法炮製出另外四個，並依順序將 Text 控制項改為 2、3、4、5，如下圖，這樣的用意是讓等下撰寫 EduCake 程式時能方便判斷 Android 裝置到底按了哪個按鈕。



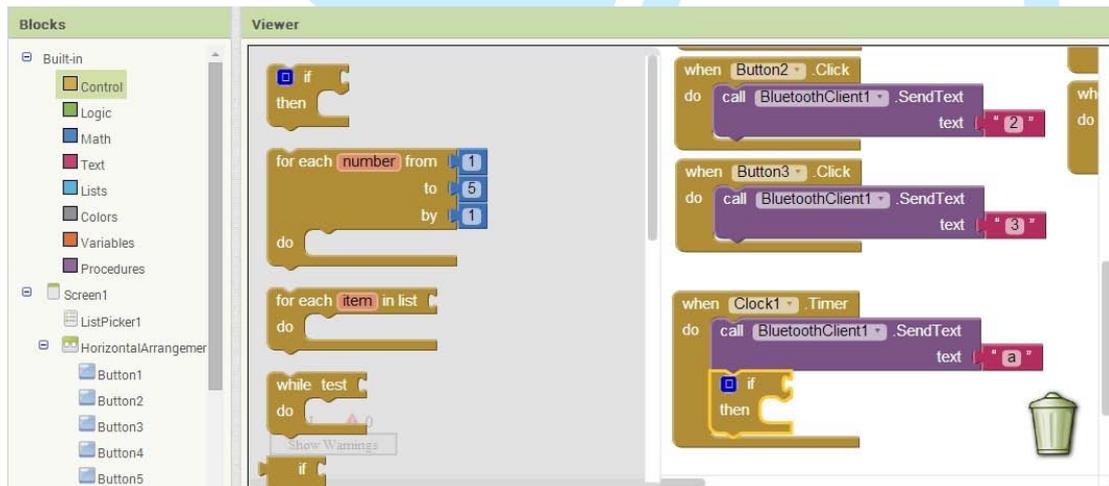
接下來要製作 Android 接收 EduCake 傳回來按鈕的訊號，左側點擊 Clock1 拉出 Timer 出來，如下圖。



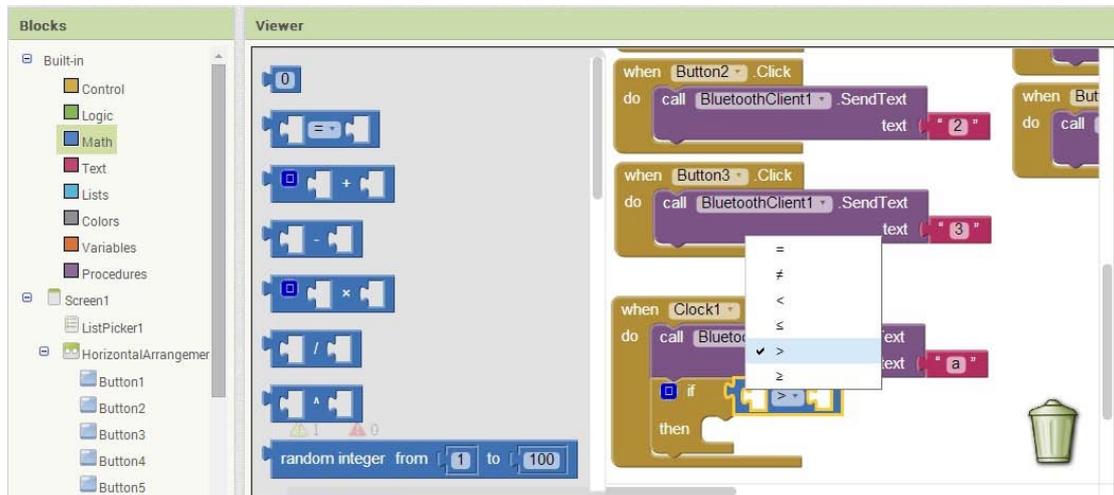
接下來我們可以點擊上面的.SendText 做 Ctrl + C 複製, Ctrl + V 貼上, 然後再拖曳進去 Timer 做接合, 再將 Text 的部分改為 a, 如下圖。



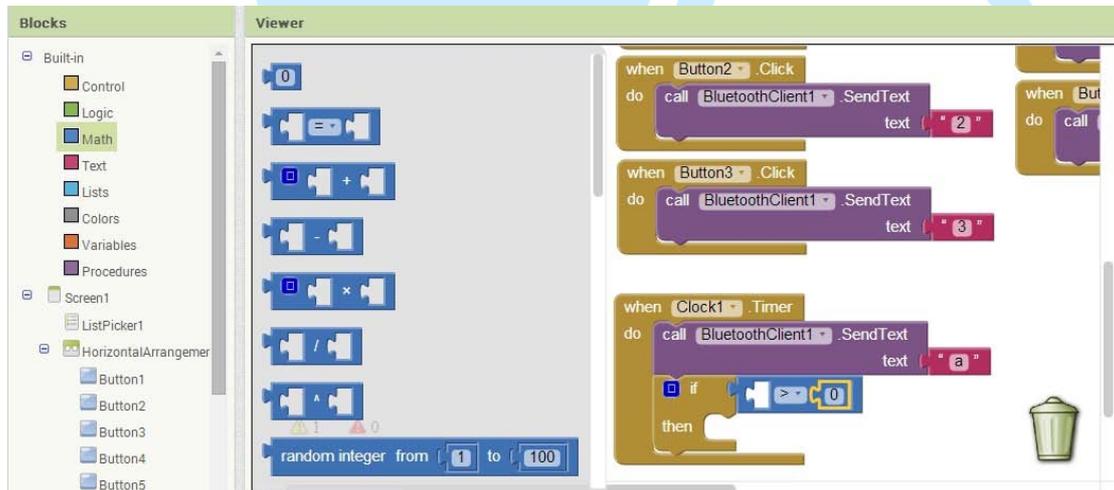
然後從左側的 Control 中拉出一個 if 判斷式接在下面, 如下圖。



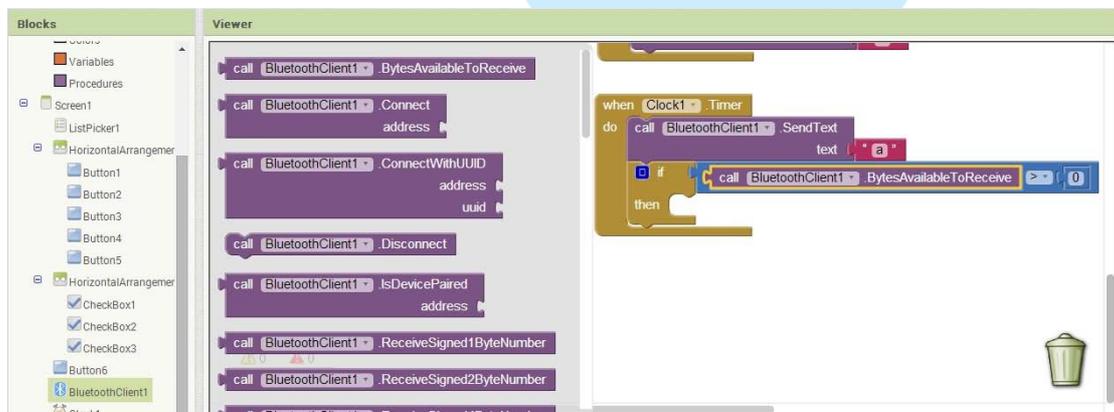
從左側的 Math 中拉出一個比較式接在 if 的判斷中，並點擊中間改成 ">"，如下圖。



從左側 Math 拉出數字項，拖曳至比較式內右邊空格中接合，如下圖。

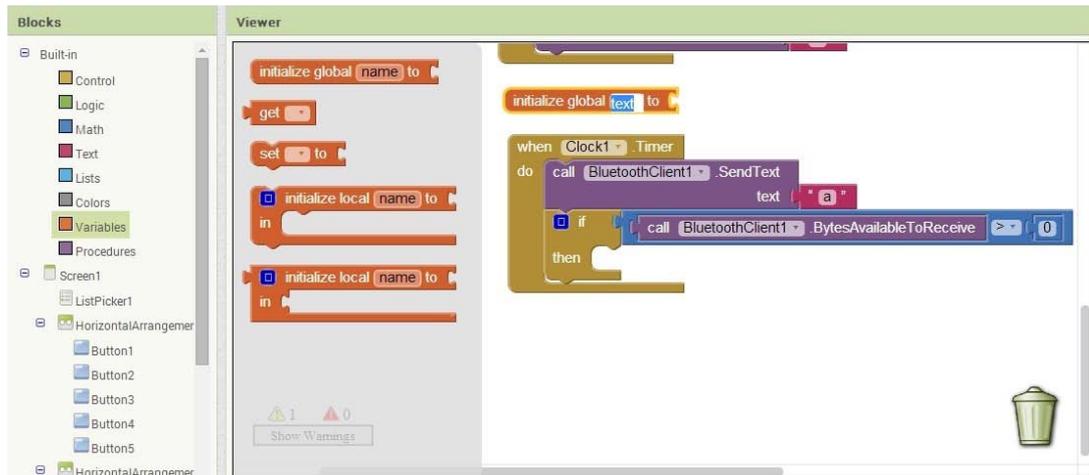


從左側 BluetoothClient1 拉出.BytesAvaliableToReceive，拖曳至比較式內左邊空格中接合，如下圖，這串比較式可以判斷是否有資訊回傳回來。

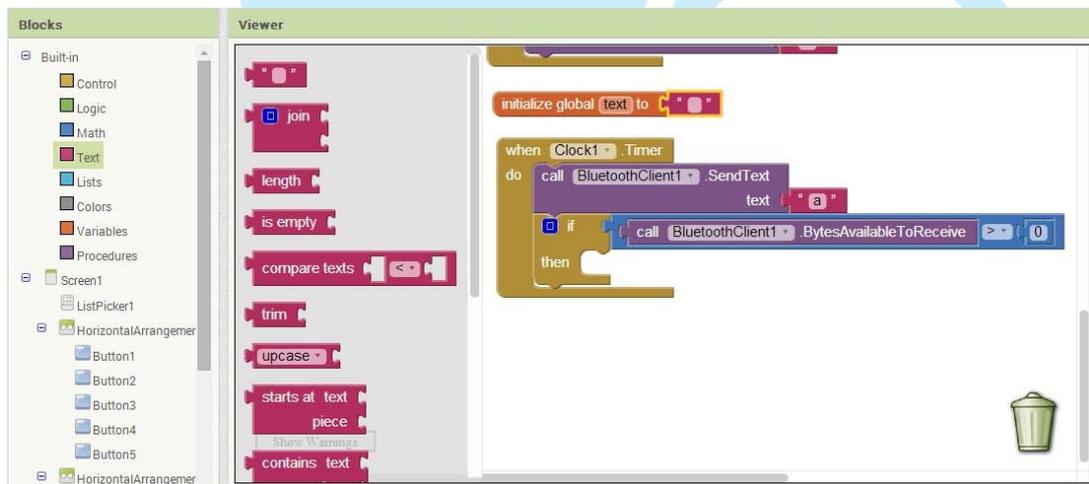


當有資料回傳時，我們就要定義一個空間去儲存它，從左側 Variables 中拖曳 initialize global name to 出來到右側放哪裡都可以，並把 name 改成 text，方

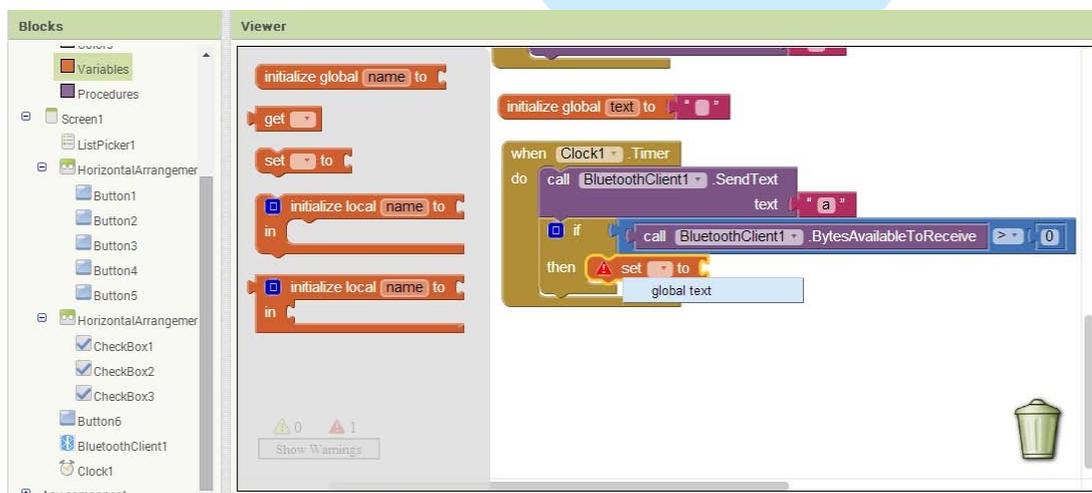
便程式閱讀，如下圖。



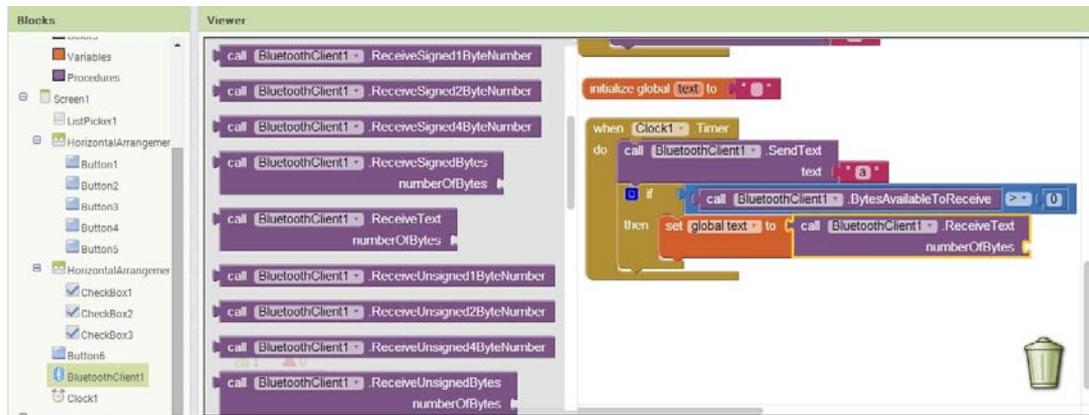
跟任寫何程式一樣，我們一定要賦予變數型別，我們就從左側的 Text 拖曳一個空白的 Text 出來接在後面，如下圖。



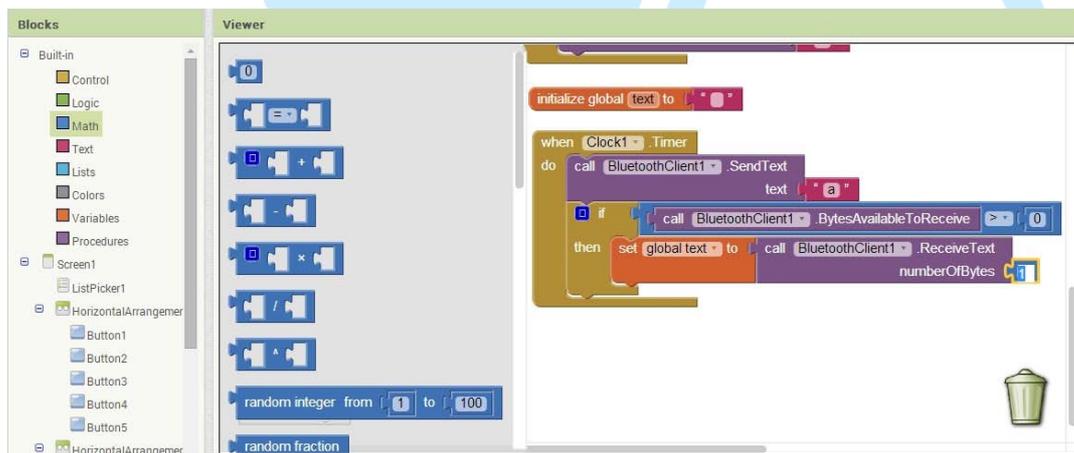
接著從左側 Variables 中拖曳 set to 控制項出來接在 if 判斷式內，如下圖，並點擊中間空白的部分選擇剛剛的字串變數，這樣紅色驚嘆號就會消失了。



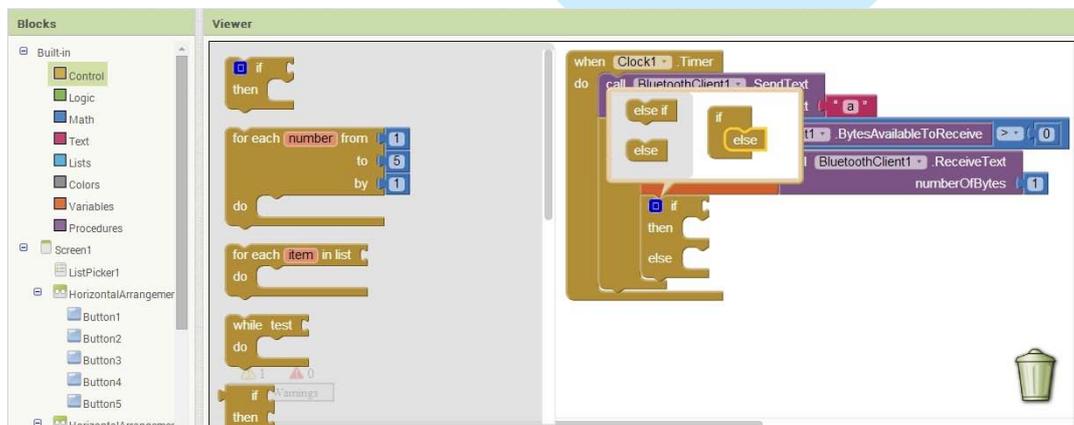
從左側 BluetoothClient1 拉 .ReceiveText numberOfBytes 出來，接合在 set global text to 後面，如下圖，它的功能是接收並回傳所指定的字串長度。



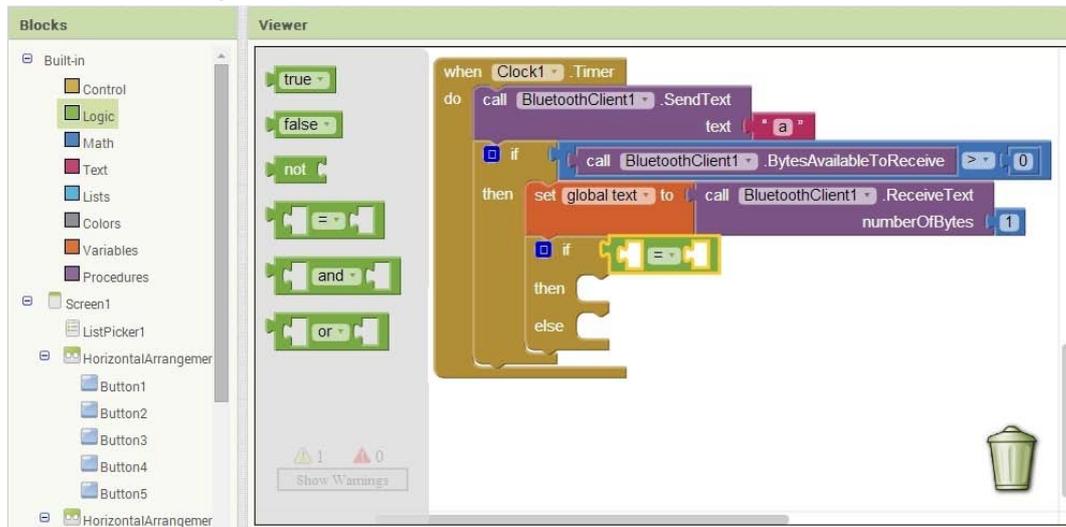
從左側 Math 拉出數字接在.ReceiveText numberOfBytes 後面，如下圖，並改為 1，我們只要接收一個字元即可



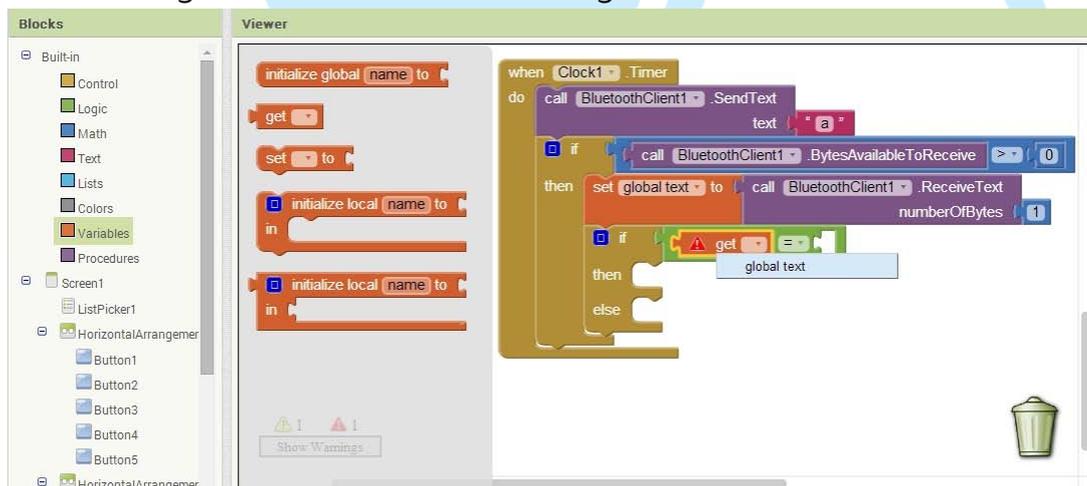
接下來從 Control 裡拖出一個 if 判斷式接在 set global text to 下面，如下圖，這次的判斷式比較特別我們要再加入例外的動作，點擊判斷式左上角藍色四方形，把左側的 else 拉進去到 if 裡面，下面就會多出一個 else 了。



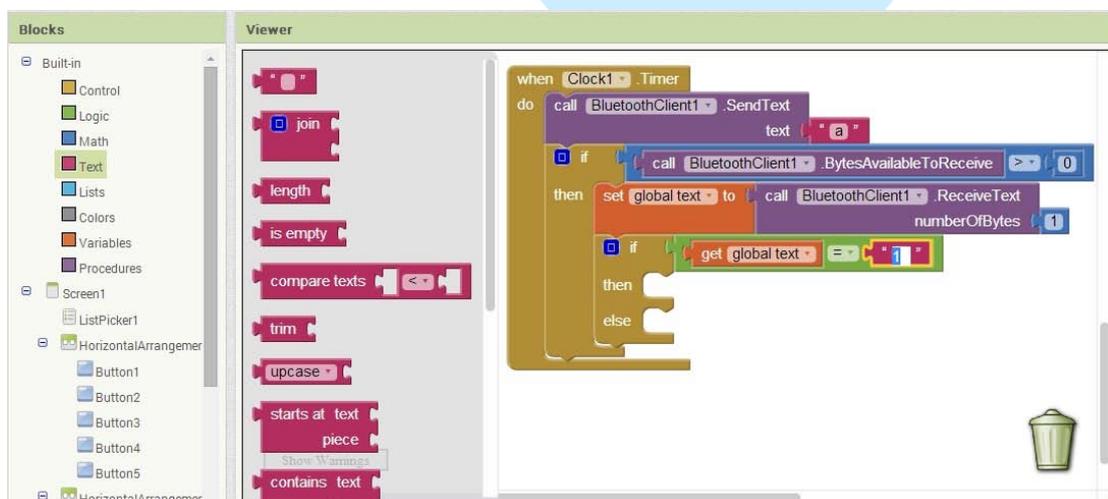
從 Logic 拖曳一個比較式出來，接在 if 的判斷中，如下圖。



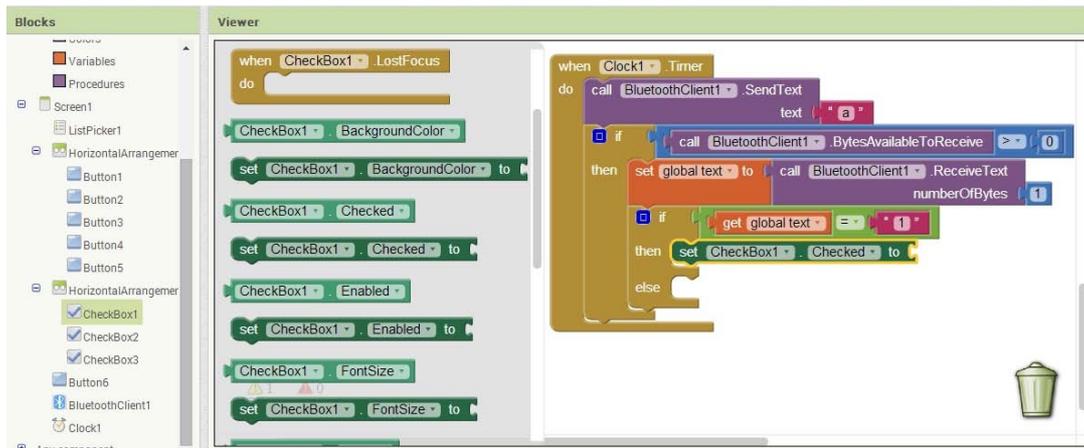
接著從左側 Variables 中拉出 get 出來接在比較式左邊空格中，如下圖，並點擊 get 空白處，選擇剛剛定義的 global text 字串變數。



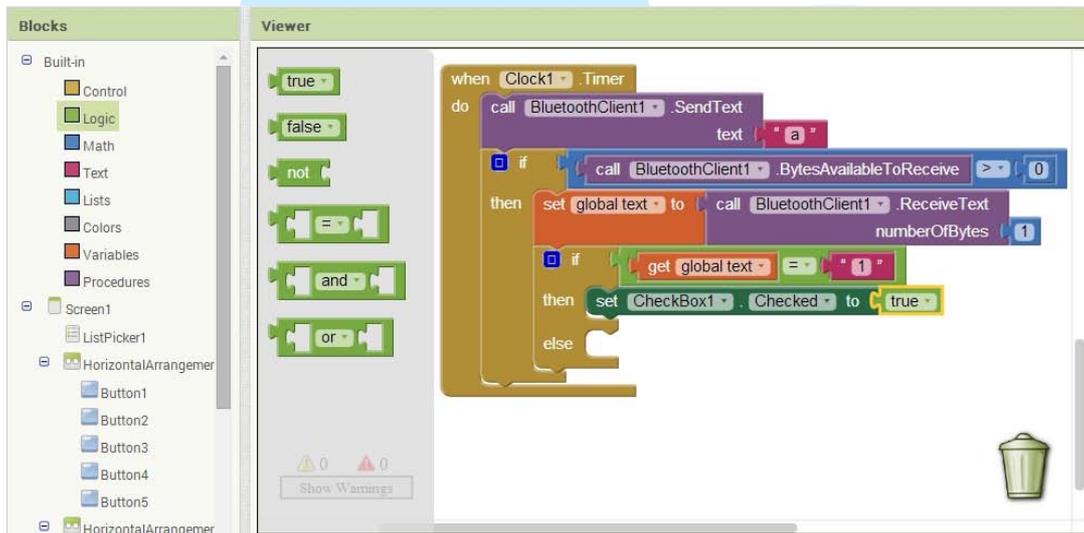
我們從左側 Text 中拉出一個空白 text 出來接在比較式右邊空白處，如下圖，並在空白中填入 1。



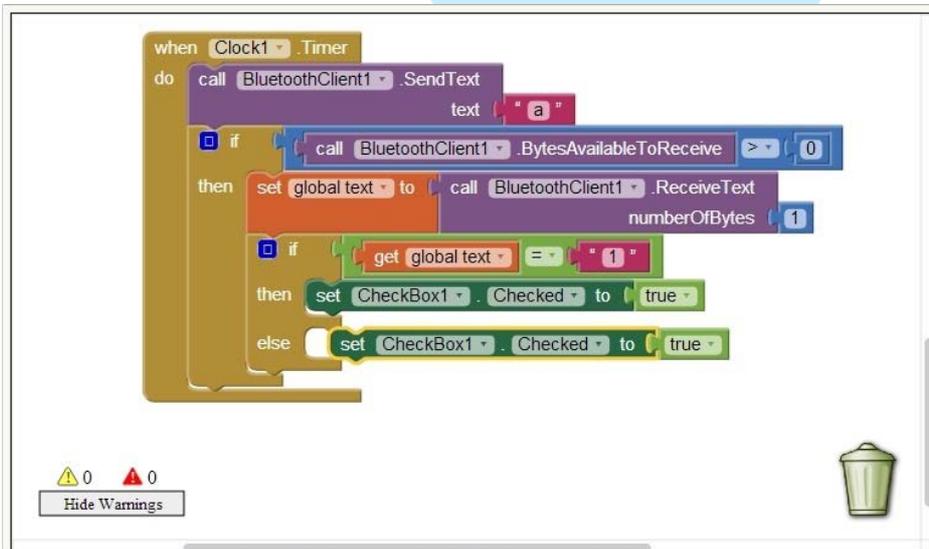
從左側 CheckBox1 中拉出.Checked 出來接到 if 判斷式內，如下圖。



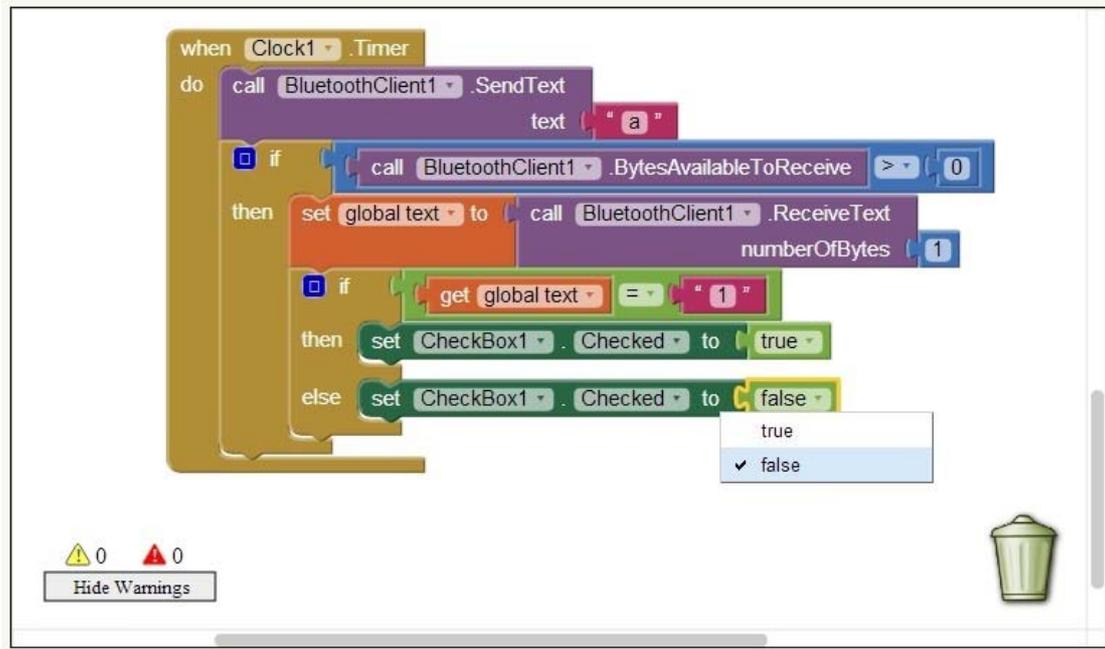
從 Logic 拖曳 true 到右邊接合，如下圖。



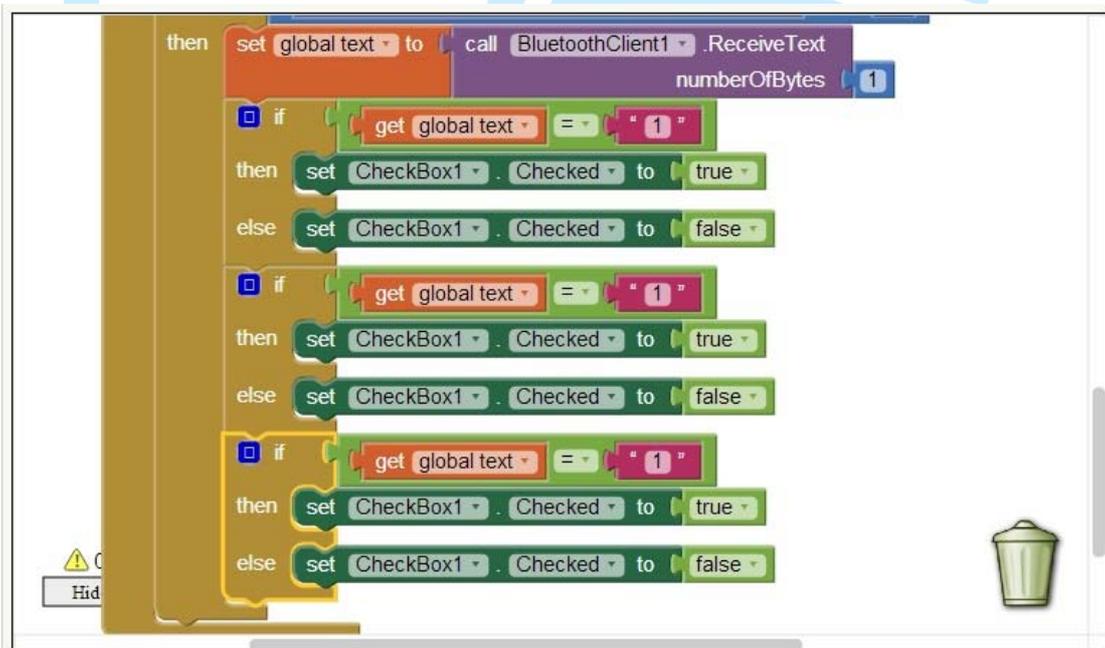
撰寫程式就是要善用複製(Ctrl + C)與貼上(Ctrl + V)，我們點擊 CheckBox1.Checked 之後按 Ctrl + C 複製，然後再按 Ctrl + V 貼上，就會再多出一組一樣的組合，如下圖，我們把它拖曳到 else 欄位中吧。



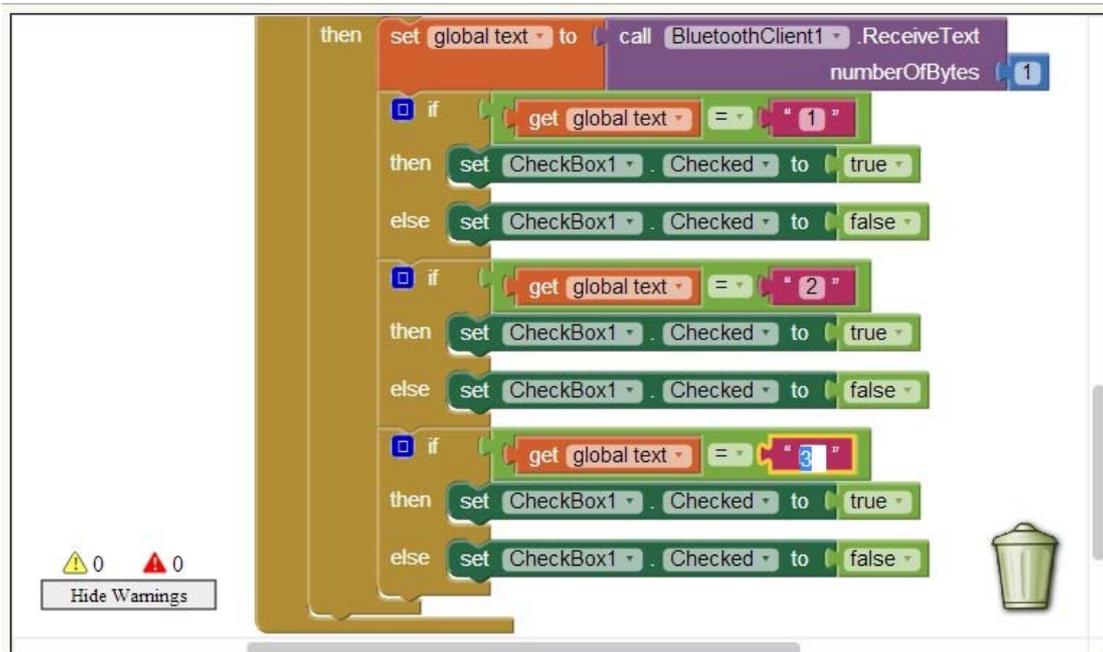
在 else 欄位中，把後面的 True 改為 false，如下圖，在 App Inventor 中，只要控制項中有箭頭都可以隨時修改喔。



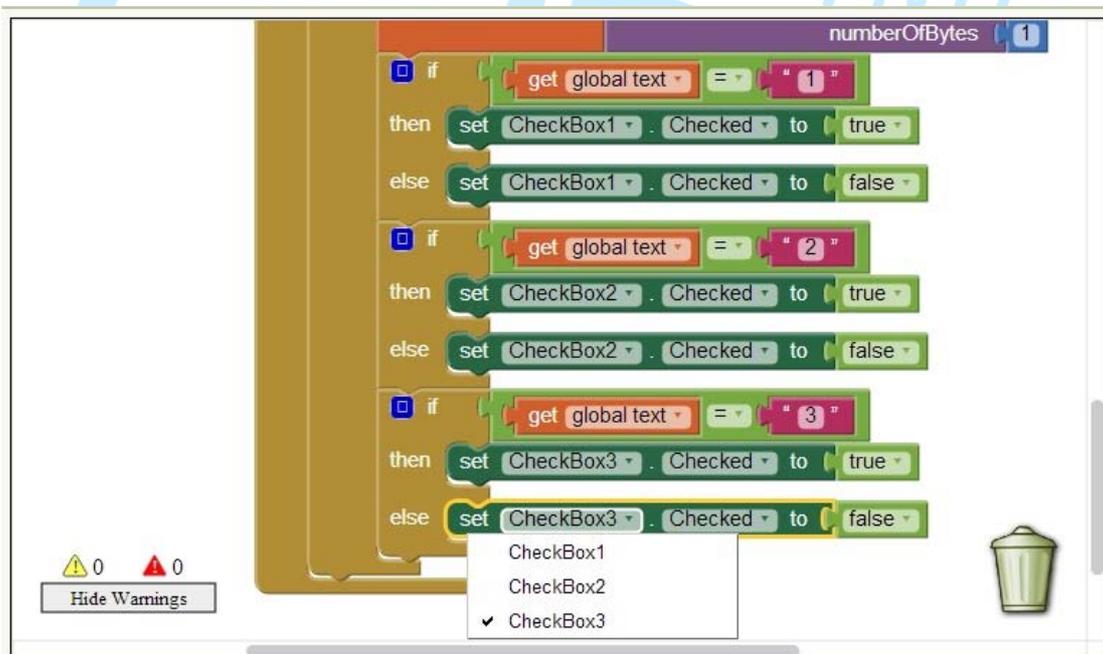
接下來就點擊剛剛做的 if 判斷式，複製(Ctrl + C)並貼上(Ctrl + V)，然後再移動到原本的 if 判斷式下接合，之後再重複做一次，所以總共會有三個 if 判斷式，如下圖。



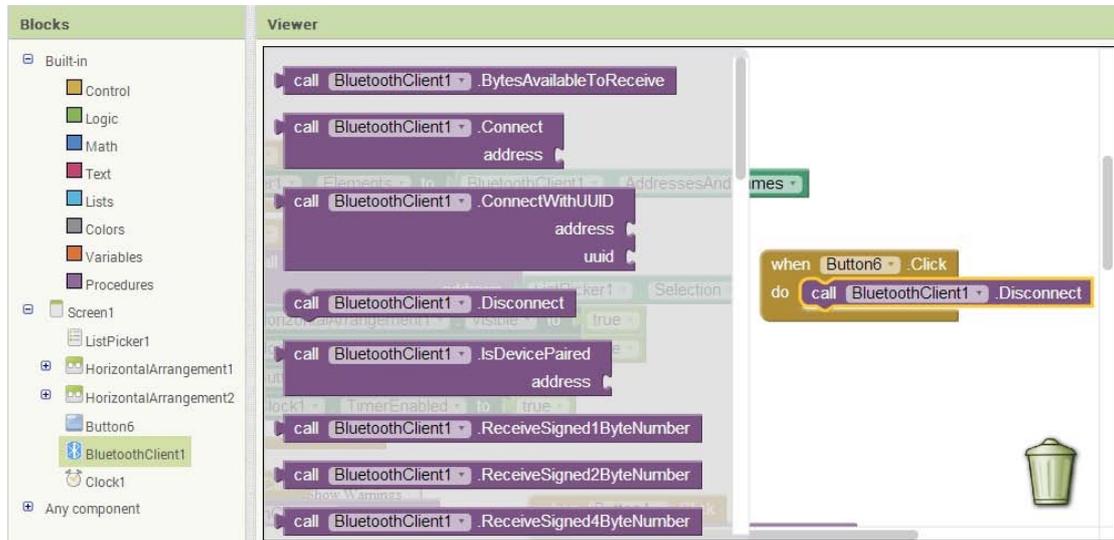
我們先來修改比較式中右邊的數字，第一個不用改，第二和第三改為 2 和 3，如下圖。



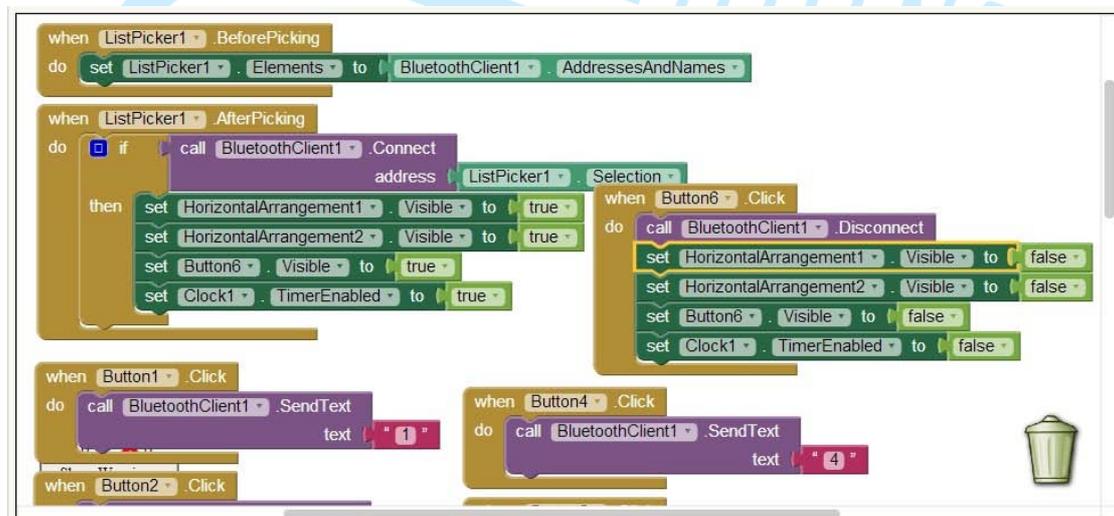
接著將第二和第三判斷式內的 CheckBox1，改為 CheckBox2 及 CheckBox3，如下圖。



接下來最後中斷連線的按鈕，也就是 Button6，拖曳.Click 事件出來，然後再從左側 BluetoothClient1 中拖曳.Disconnect 出來接合，如下圖。

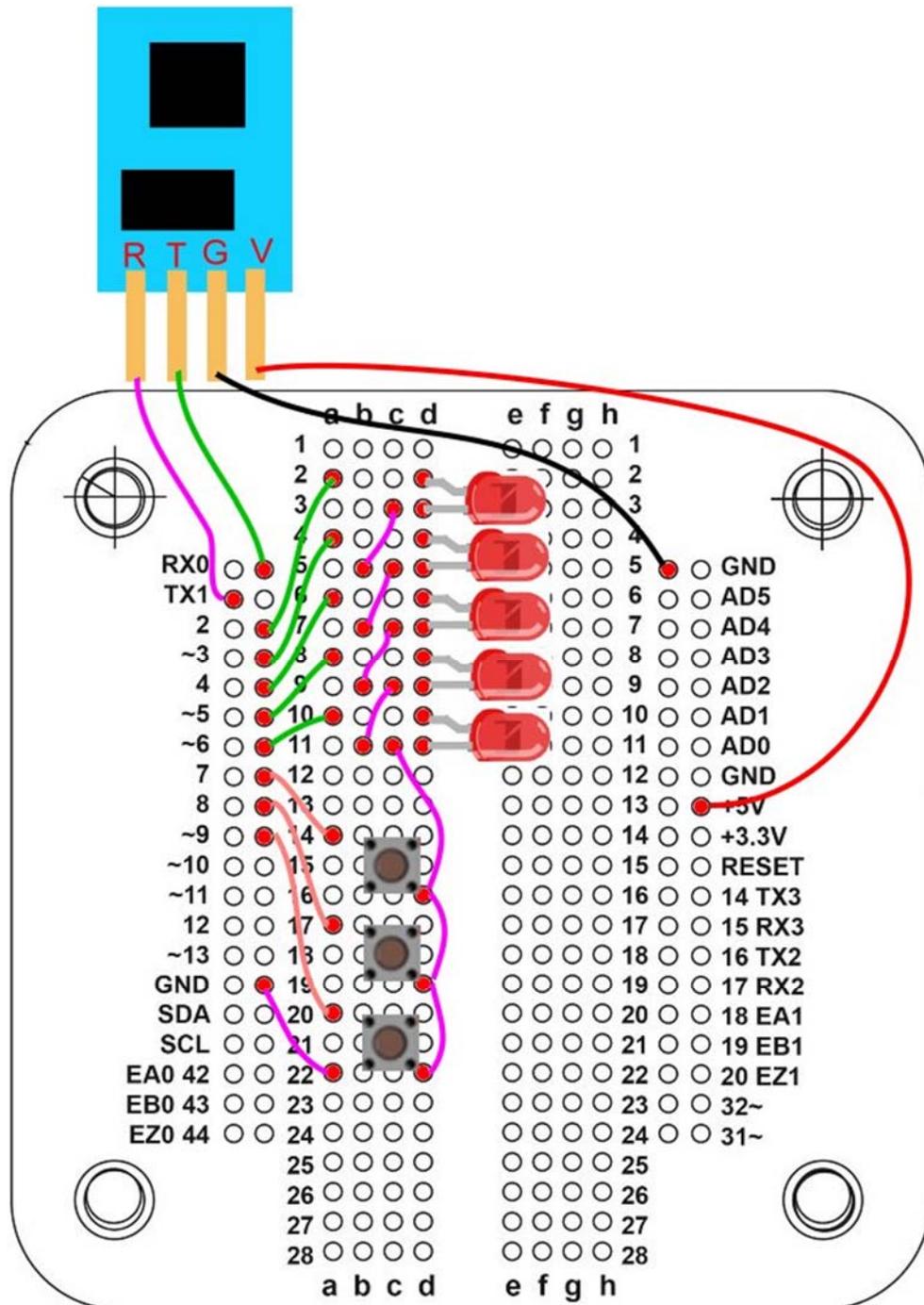


除了中斷連線之外還要隱藏按鈕和停止 Clock，就直接複製剛開始做的控制項顯示過去，再全部改成 false 就好了，如下圖，做到這邊就算完成了，接下來可以將程式傳到 Android 裝置中。



會不會覺得很奇怪，為什麼要用 Clock 去接收處理 EduCake 所發送的訊息呢？因為我們需要即時的處理這些送過來的訊息，而必須隔一小段時間掃描一次看看有無訊息，而為什麼要先送出"a"再做接收呢，因為 EduCake 的速度實在是太快了，一直傳訊息會造成藍芽反應不及，這時候我們就用 Arduino 牽制它，當接收到"a"的時候再送資訊過去就好了，這樣可以確保資料有確實的傳遞。

接下來我們將 EduCake 的 LED 從一個接到五個，再增加三個按鈕，接腳圖如下。



接著開啟 IDE 介面，輸入以下程式

```
char ch;
int LED[5]={0,0,0,0,0}; // 把 5 個腳位狀態先存到陣列裡面

void setup() {
    Serial.begin(9600); // 這行主要是設定和電腦的 COM 通訊的
    速度
    Serial1.begin(9600); // 這行主要是設定和藍芽 通訊的速度
    for(int a=2;a<7;a++) // 先設定這五個腳位為輸出模式
        pinMode(a, OUTPUT);
    for(int a=7;a<10;a++) // 先設定這三個腳位為輸入模式
        pinMode(a, INPUT);
}

void loop() {
    if (Serial1.available() > 0)
    {
        ch = Serial1.read();
        if(ch == 'a') // 等待 Android 端傳送"a"過來
        {
            if(digitalRead(7)) // 偵測 pin7 按鈕是否按下
                Serial1.print("1");
            else if(digitalRead(8)) // 偵測 pin8 按鈕是否按下
                Serial1.print("2");
            else if(digitalRead(9)) // 偵測 pin9 按鈕是否按下
                Serial1.print("3");
            else
                Serial1.print("0");
        }
    }
}
```

```
    }  
    else  
    {  
        if(LED[ch-'0'-1]==LOW)  
        {  
            LED[ch-'0'-1]=HIGH; // 儲存 LED 的狀態  
            digitalWrite(ch-'0'+1, HIGH);  
        }  
        else  
        {  
            LED[ch-'0'-1]=LOW;  
            digitalWrite(ch-'0'+1, LOW);  
        }  
    }  
}  
}
```

其中

```
if(LED[ch-'0'-1]==LOW)
```

因為我們傳輸的是字元，我們要把它變成一半數字最簡單的方法就是直接減去0的字元就好，至於減去1是因為陣列剛剛定義5個實際是從0~4，所以要減去1，舉個例子來說，當我們在Android裝置中按下LED1，它就會傳送一個"1"的字元回來，我們用if判斷陣列中的第0個位置自否為低電位，然後再做相應的動作。

接下來就是測試時間啦，連線之後，點擊 LED1~5 會有相應的 LED 會亮起，而按下 EduCake 上的按鈕會反應在按鈕 1~3 上。

