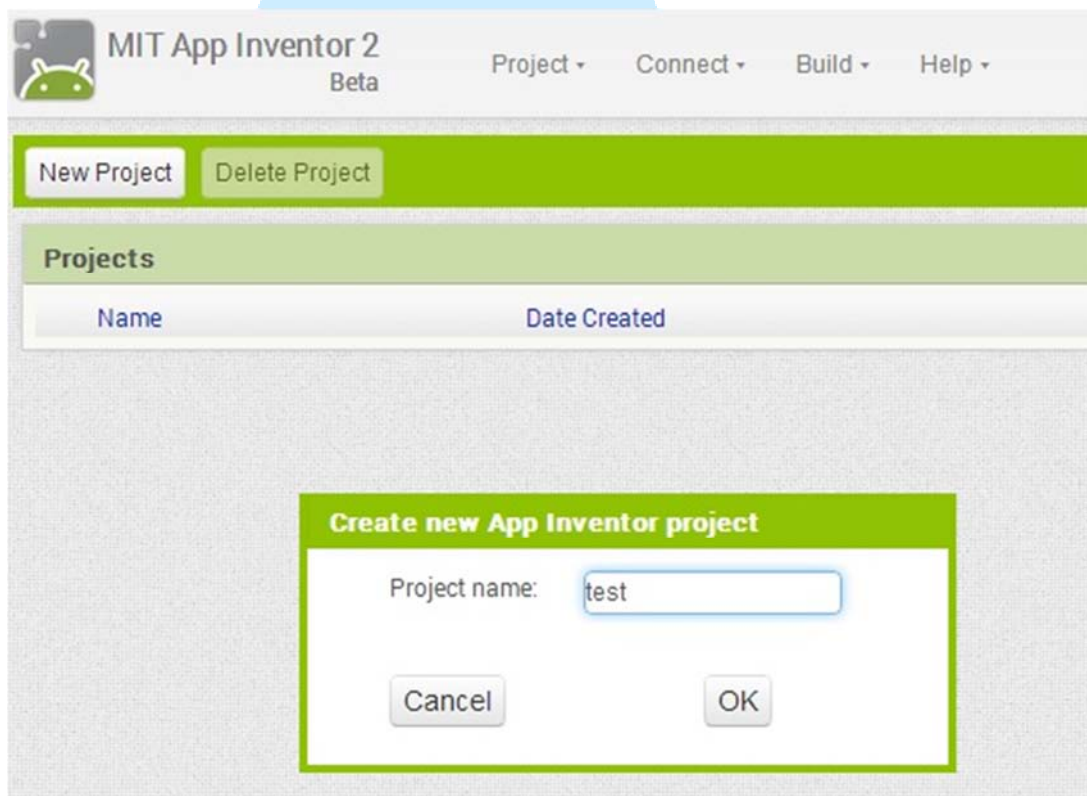


App Inventor 的基本介绍和蓝牙链接功能的使用

一、专案建置与基本介绍

我们 MIT App Inventor 2，它是图形化接口的开发环境，只要拉对象和方块图就能完成一个 APP，是不是很简单呢？

首先，进入网址 <http://ai2.appinventor.mit.edu/> 并登入 google 账号后点击 New Project 来新增专案，并打上专案名称点击 OK，如下图。



进入之后会看到开发接口，主要分为四项，Palette、Viewer、Components 和 Properties

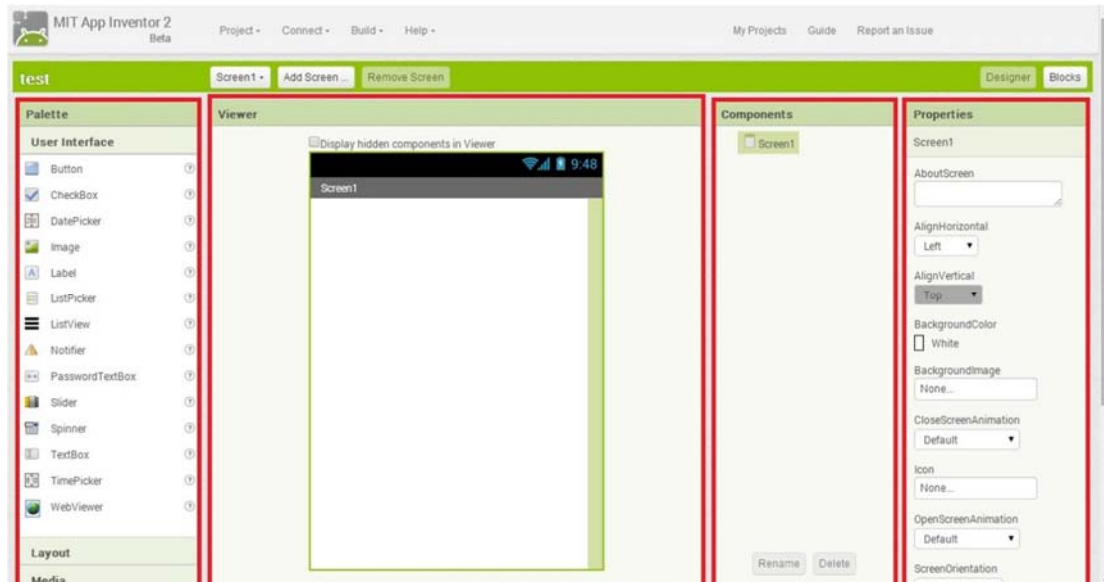
Palette: 所有组件的放置区，需要那些功能就把它拖到 Viewer 区的屏幕里，.net 玩家看了应该会觉得很亲切，很多功能都大同小异。

Viewer: 装置预览的画面，把需要的组件拖曳进来并做排版，但实际位置还是有出入，放到装置上看会比较准。

Components: 所有拖曳到预览画面的组件(包括隐藏的)都会出现在这里面，

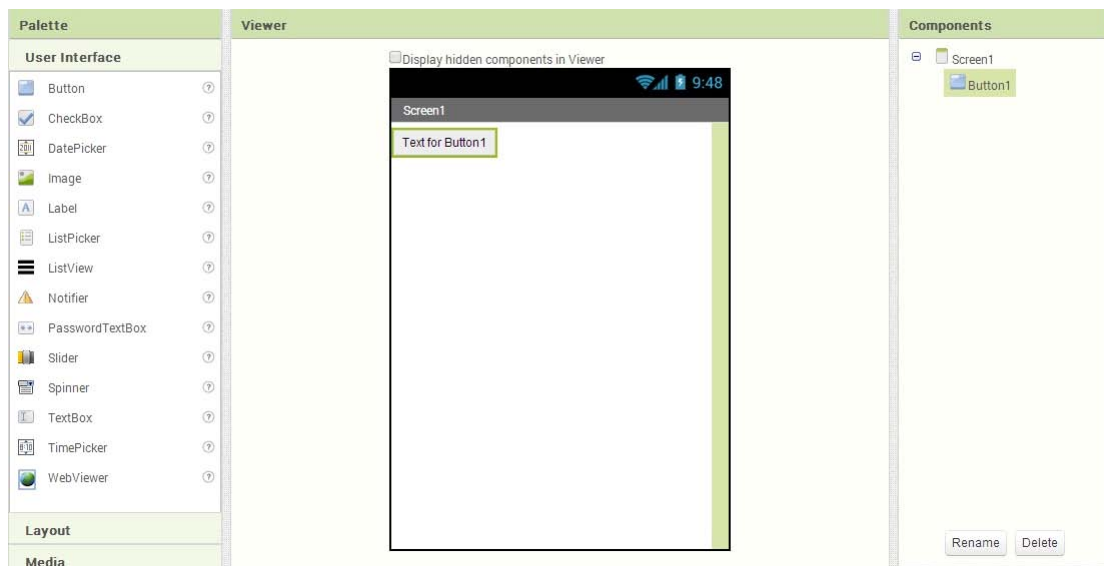
我们可以点击里面的组件到右边的 Properties 做修改。

Properties: 显示组件的属性，可以在预览画面中点击或 Components 里点击即可编辑组件的属性。



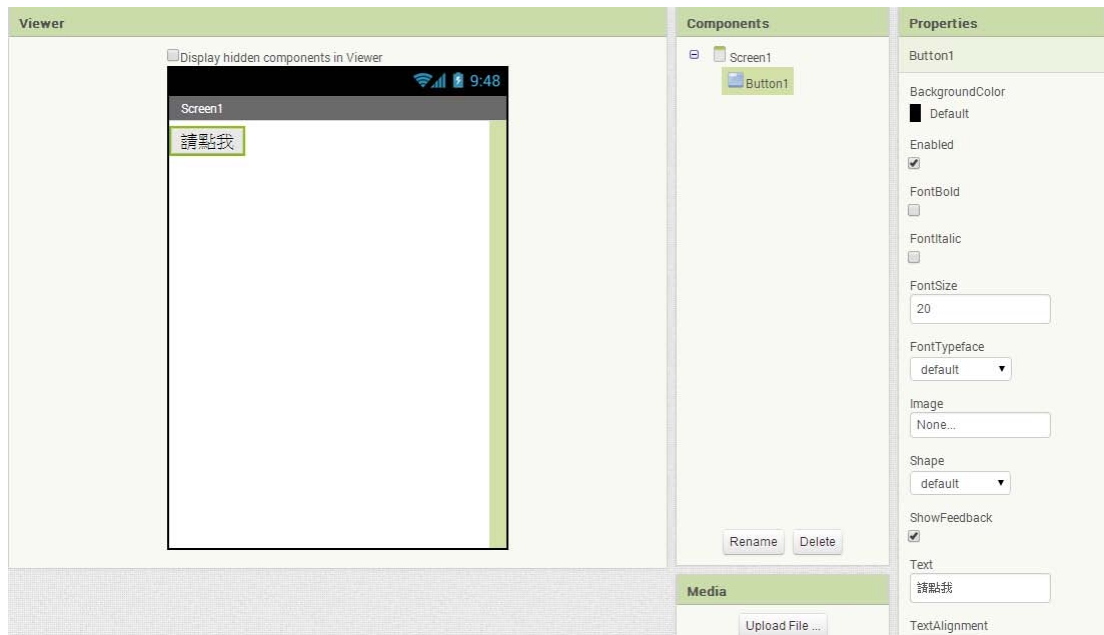
二、第一个程序

我们先来做一个按按钮会改变文字的程序，先把左侧的 button(按钮)点住并拖曳到预览画面中，如下图，如果拖错组件可在 Components 框框内点击 Delete 出现对话框按 OK 即可删除，另外 Rename 可对组件做更名，但在写程序也就是堆方块时也要找相同名字才能做调用的动作。

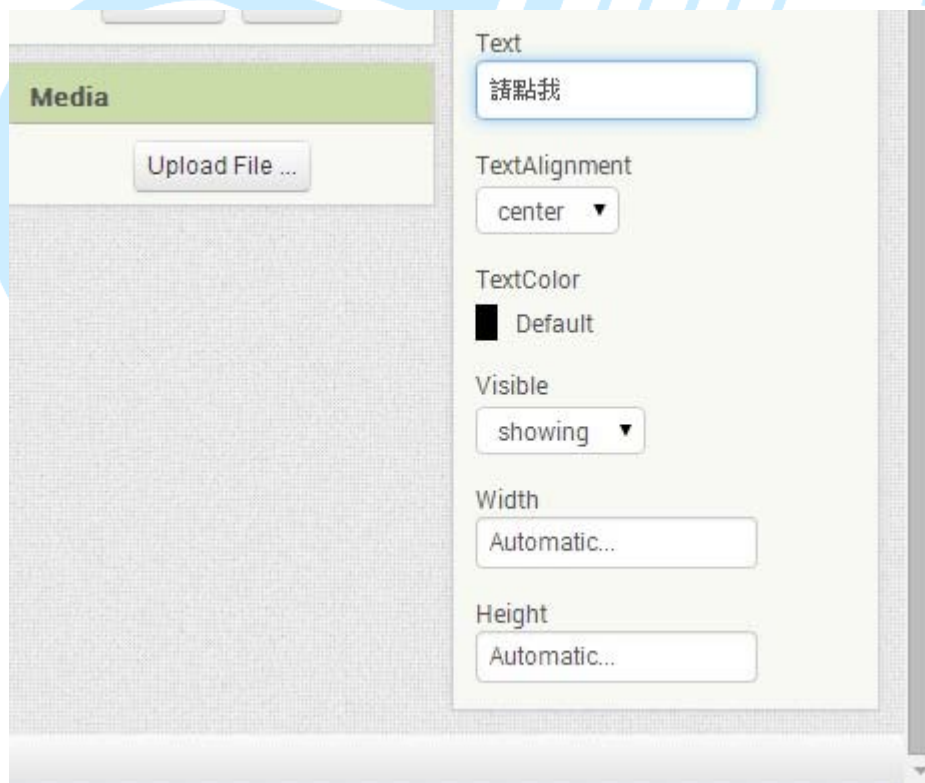


再来我们到 Properties(属性) 框框中的 FontSize (字号)改为 20，然后在

Text 选项中，改成想要的文字，例如：请点我，如下图。



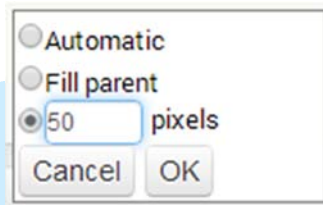
拉霸往下拉或鼠标往下滚到下面可以看到 Width(宽) 与 Height(高) 的选项。



点开可以看到三个控件: Automatic(自动)、Fill parent(充满)、pixels(自定义大小)，我们在 Width(宽) 的属性改为充满，如下图。



在 Height(高)的选项中选择自定义并输入 50，单位为 pixels，如下图。



完成图，记得属镜中的 Height(高)要比 FontSize(字号)大，不然会切到字喔。

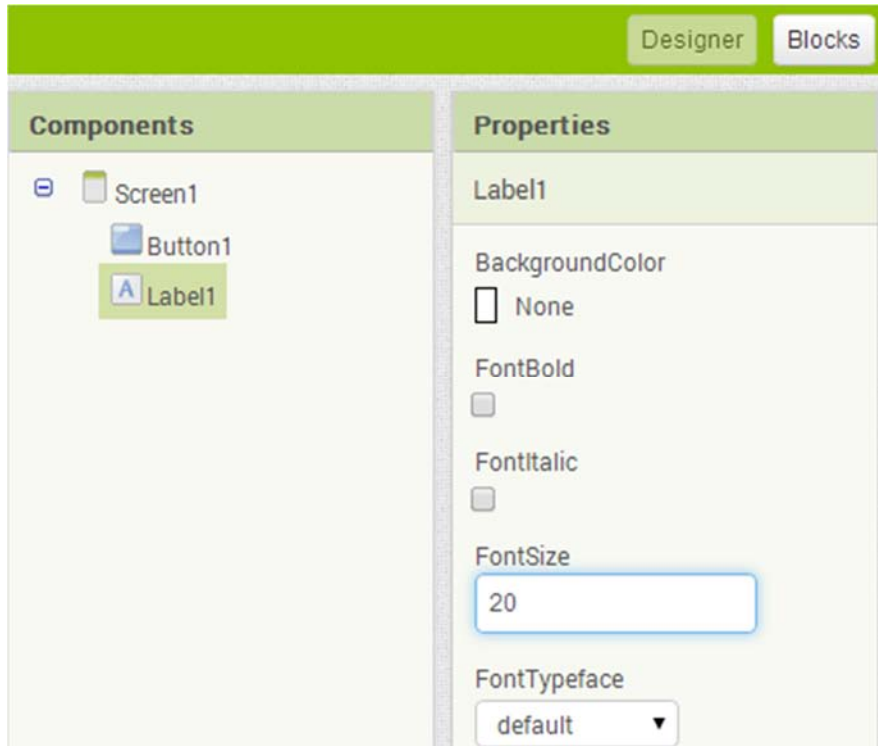


接着再把 Label 拖曳到预览画面中，如下图。

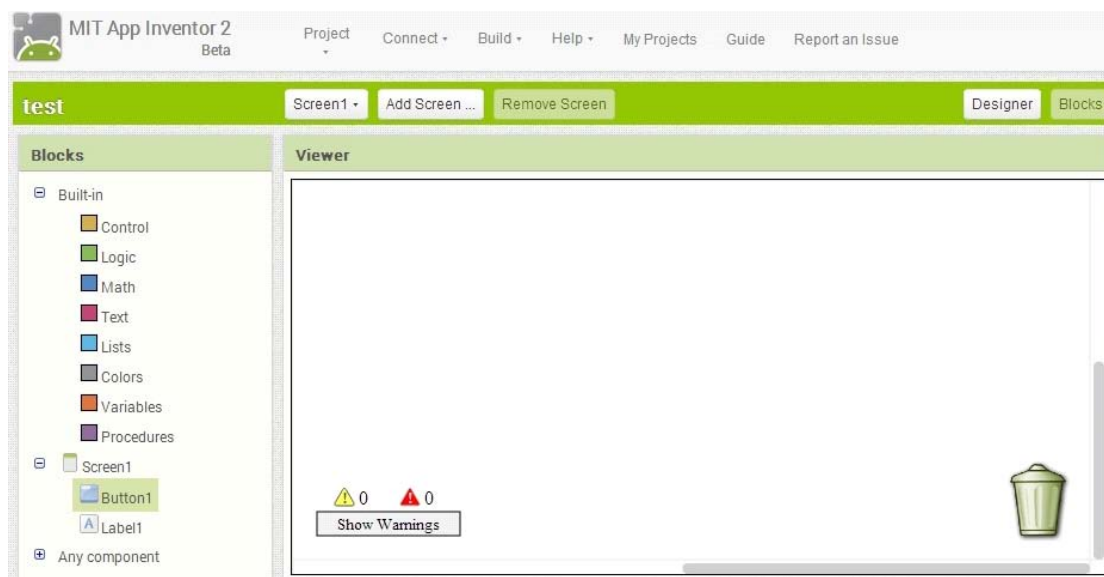


在再来我们将 Properties 框框中的 FontSize 一样改为 20，如下图，接着

看到右上角的 **Designer** 是设计程序 UI(接口)的地方也是目前所在的地方，而 **Blocks** 是积木区，也就是撰写程序的地方，我们程序接口也做的差不多了，接着来点击 **Blocks** 到积木区来一探究竟吧！

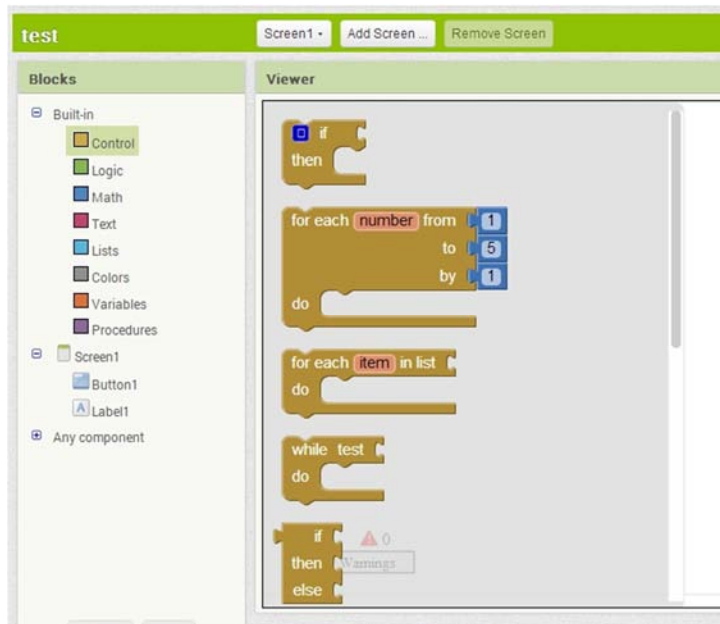


主要分为两大区域，分别是 **Blocks** 积木区和 **Viewer** 观看区也就是程序撰写的区域，我们先来介绍 **Blocks** 的部分吧。

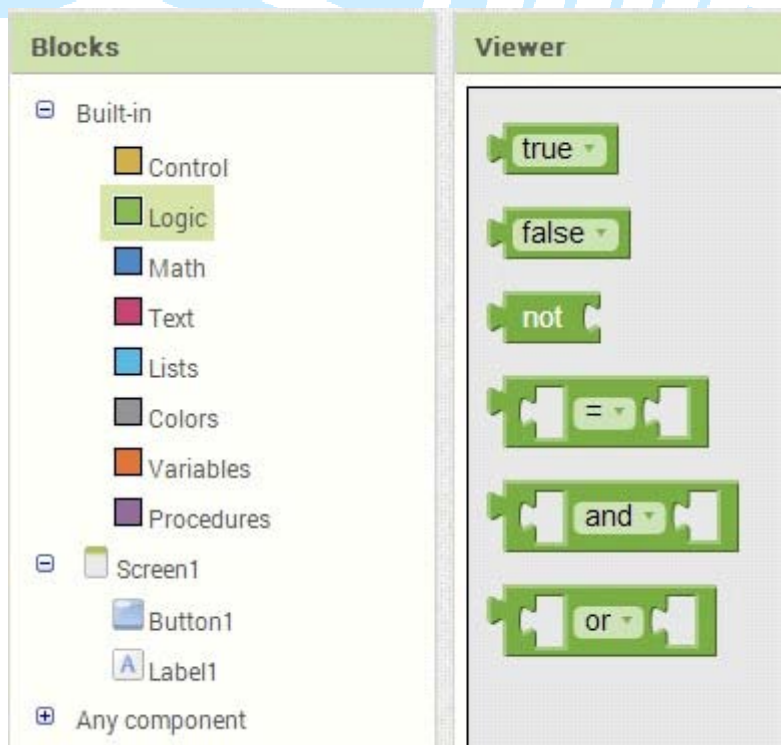


Blocks 积木区，主要会用到的部分是内建的 **Built-in** 及刚刚拉进预览窗口中

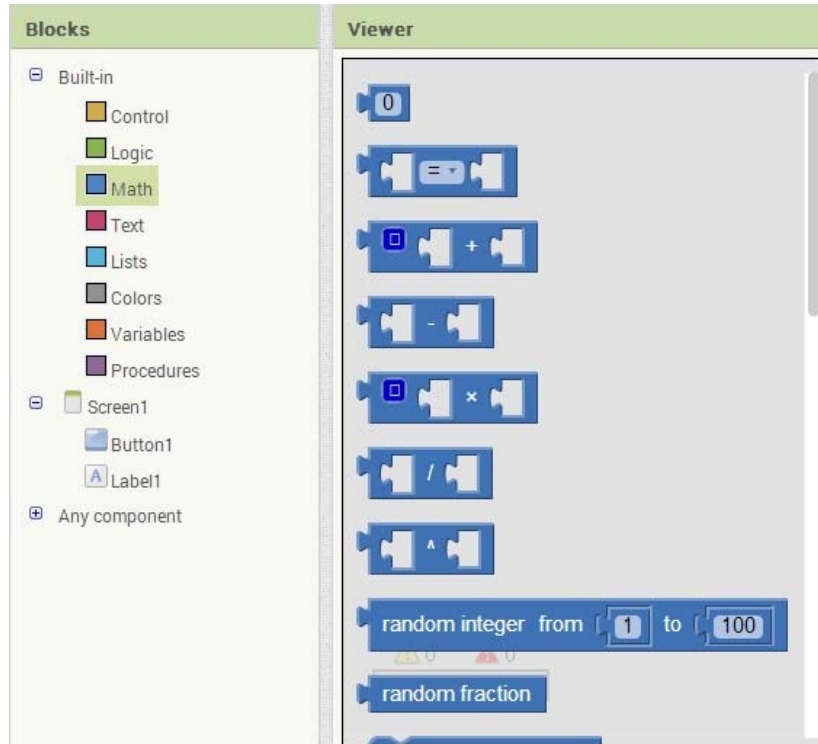
组件都会出现在 Screen1 中，我们现在点击 Built-in 中的 Control(控制)选项，如下图，有没有很熟悉呢？我们常用的判断式和循环都放在这里喔~



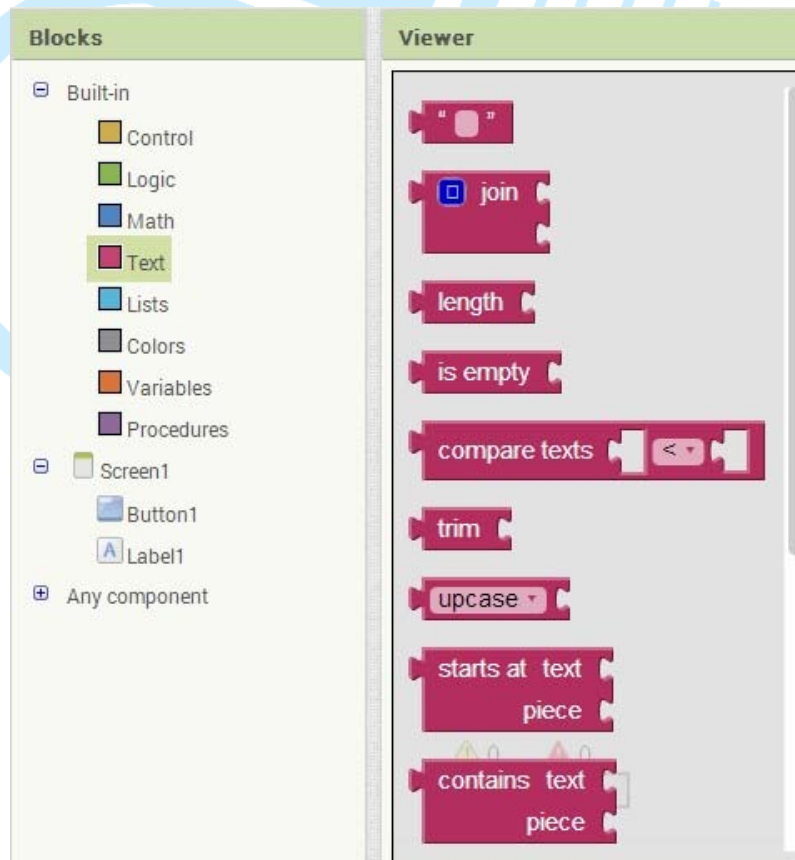
接下去是 Logic(逻辑)，布尔运算大家应该都不陌生，这里就不多做介绍了。



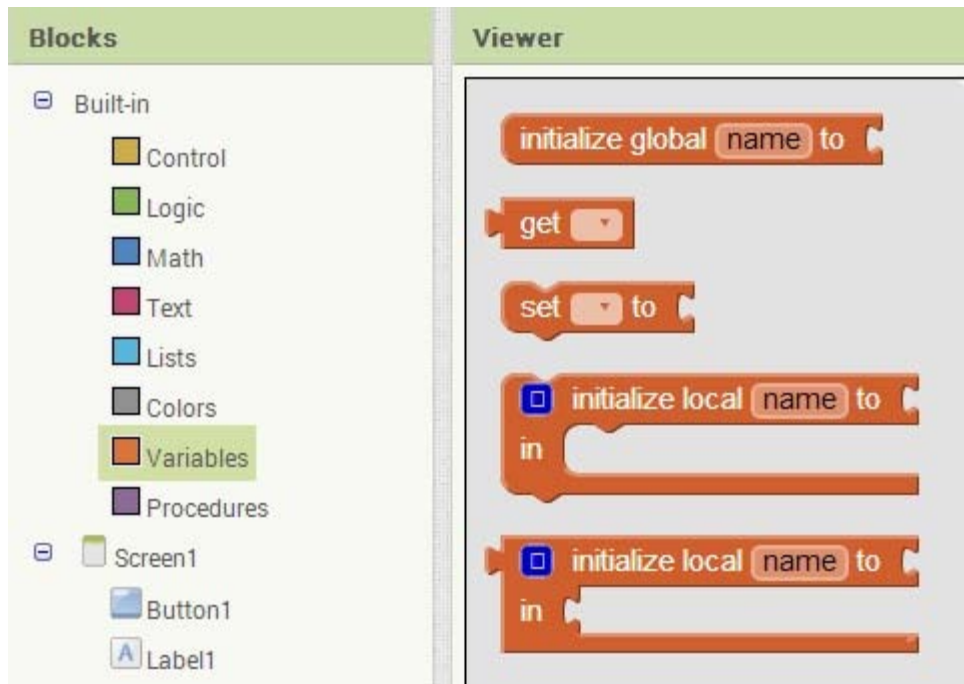
Math(数学)，里面包含了基本运算及随机变量，连三角函数都有喔。



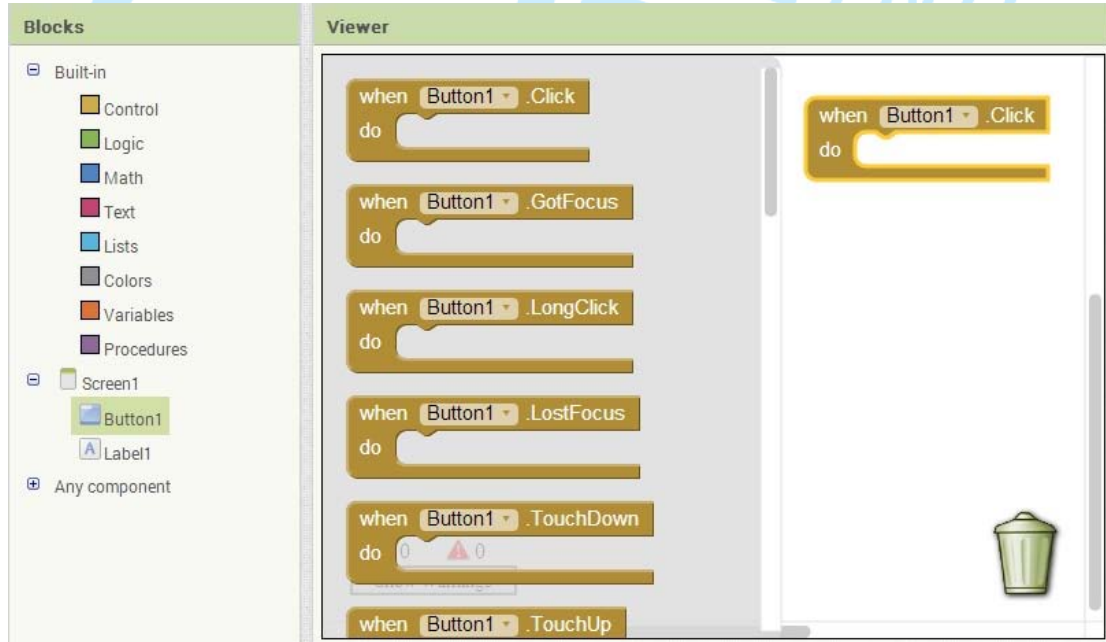
Text(文字), 包含字符串处理都在里面。



Variables(变量), 宣告、改变、取得变量的指令都在里面。



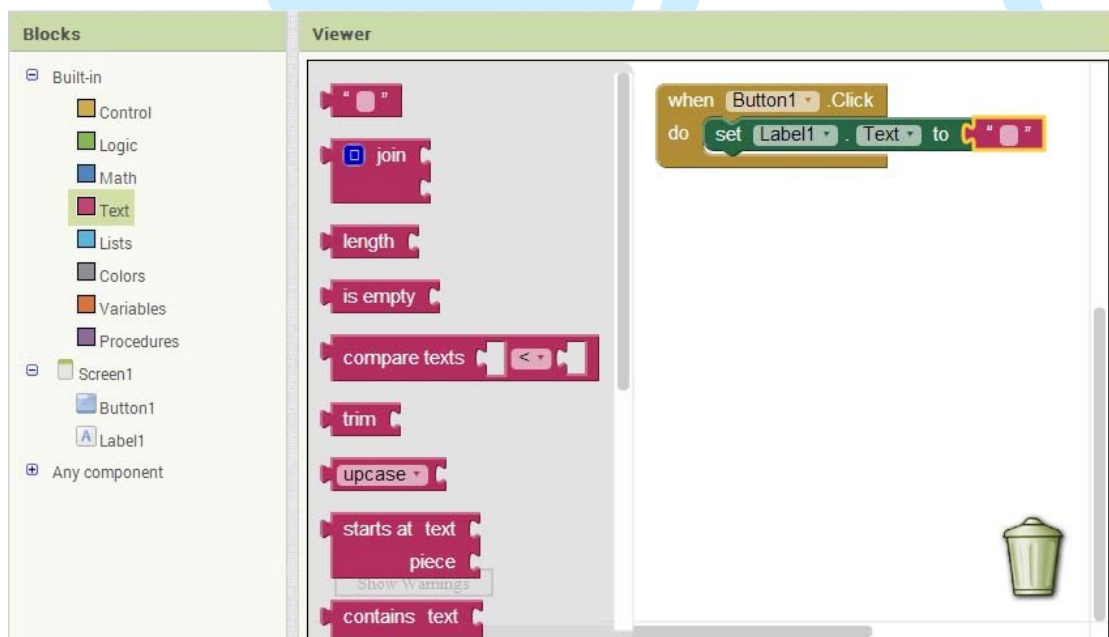
接着要进入正题啦，当 Button 被点击之后会触发 Click 的指令，我们就可以针对这个控件做一些事情，点击 Button1 可以看到控件(积木)跳出来，我们拖曳.Click(点击)这个积木出来到右侧，如果拖错东西出来可以拖到右下角的垃圾桶把它删除喔，如下图。



接着点击左侧的 Label1，把标有「set Label1.Text to」的积木拖曳到刚刚的积木里把它拼起来，如下图，它的功能为改变 Label1 显示的字。





接着点击左侧的 Text，把第一个积木拖到右边拚进去。

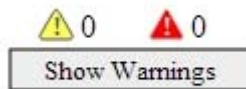


鼠标点击刚刚的积木中空白处把内容改为想要的字，例如：你好，这样程序就大致完成了，是不是很简单呢？

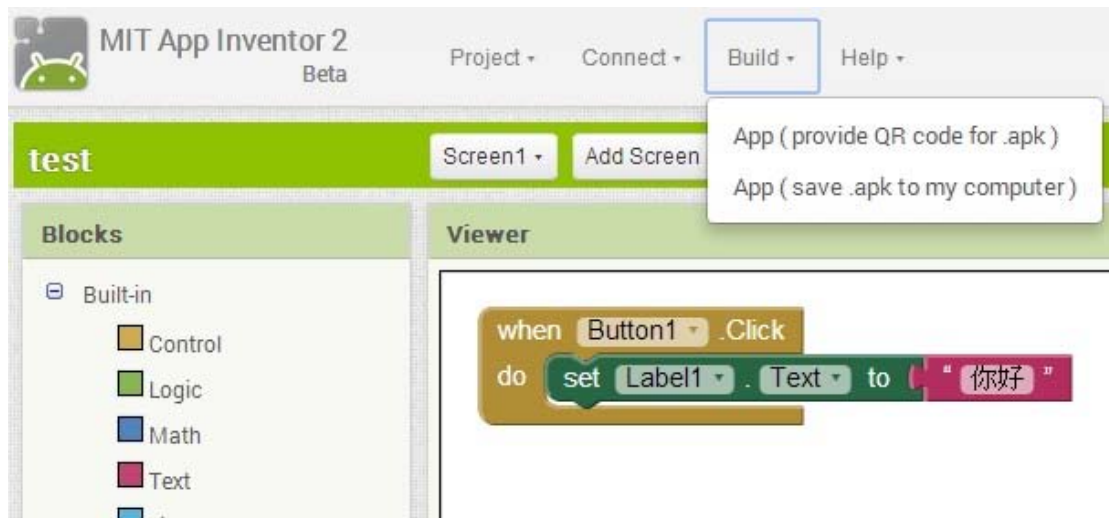


如果刚刚有开喇叭，在积木接合的瞬间会有"喀"一声，另外当积木拼不进去并弹开时，就代表你逻辑错误或拉错积木啦！在过程中观察一下会发现判断式和一些控制指令都是 ，而回传信息的积木都是 ，另外还要注意左下角有没有错误呢，如下图，黄色惊叹号是警告，通常都是积木没拼上去，可以不理它，

但是当有红色惊叹号出现，就表示程序严重错误，赶快找到错误并修正吧。



程序完成之后，目前有两种方案可以传到 Android 装置，方案一是藉由 WiFi 与计算机同一个局域网络可直接传档案，方案二是直接生成 apk 安装档，再丢到 Android 装置中安装。



将 APP 传到装置之前，先到 设定-> 安全性 把不明来源打勾，如有出现警告请按下确定。

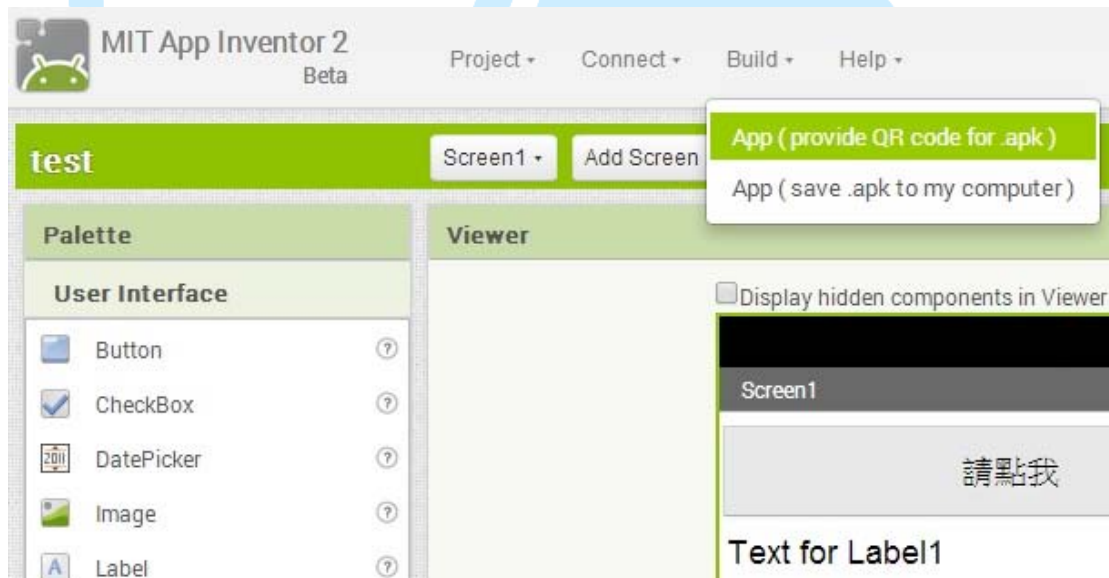


如果选择方案一，必须先安装 MIT AI2 Companion 这个 APP，开启

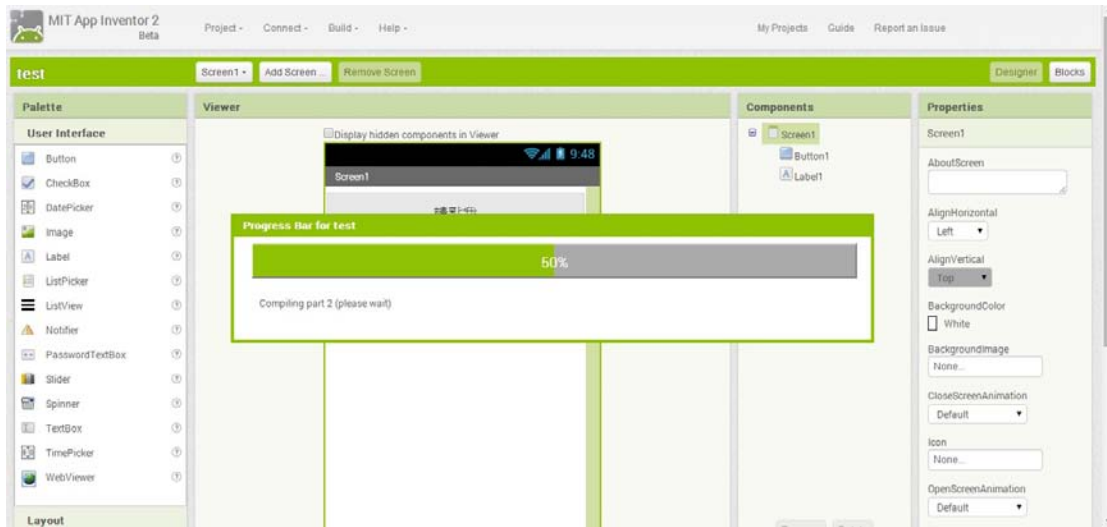
PLAY 商店搜寻打 MIT 就有关键词可以选择了，如下图，安装之后就把它开启吧。



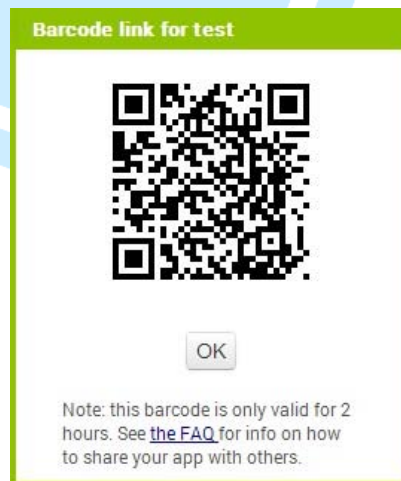
先回到计算机中，点击 Build-> App(provide QR code for .apk)，如下图。



接着会跑出进度条，正在编译 APP 中，如下图。



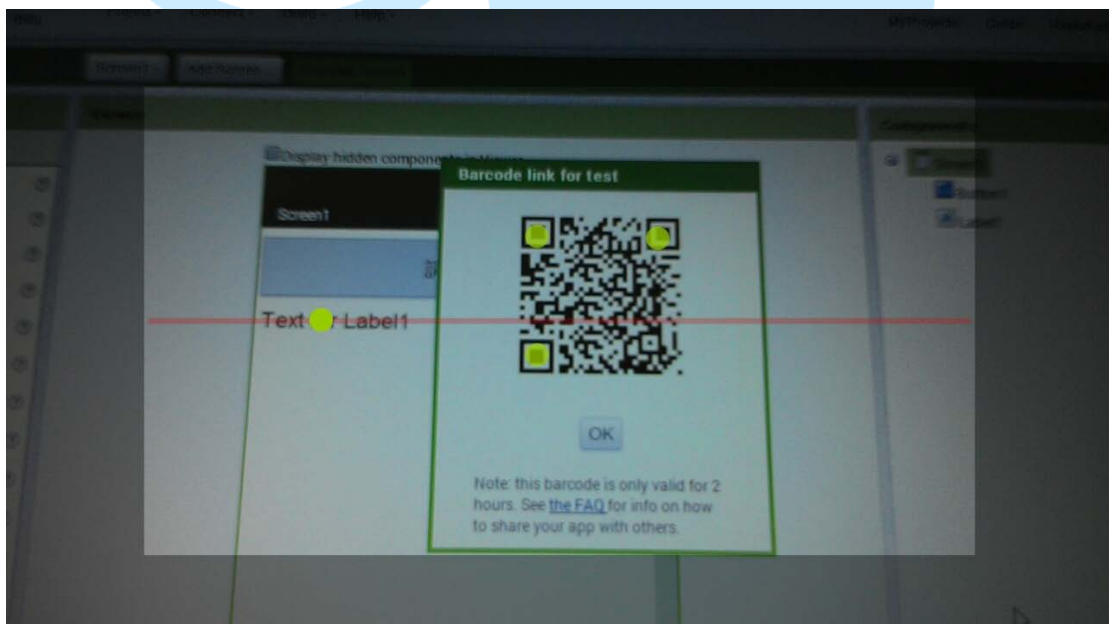
进度条跑完之后就会出现 QRcode 啦，如下图。



接着就是拿起装置开启刚刚安装的 MIT AI2 Companion 程序啦，如下图，然后点击 scan QR code。



之后对着屏幕扫描，如下图。

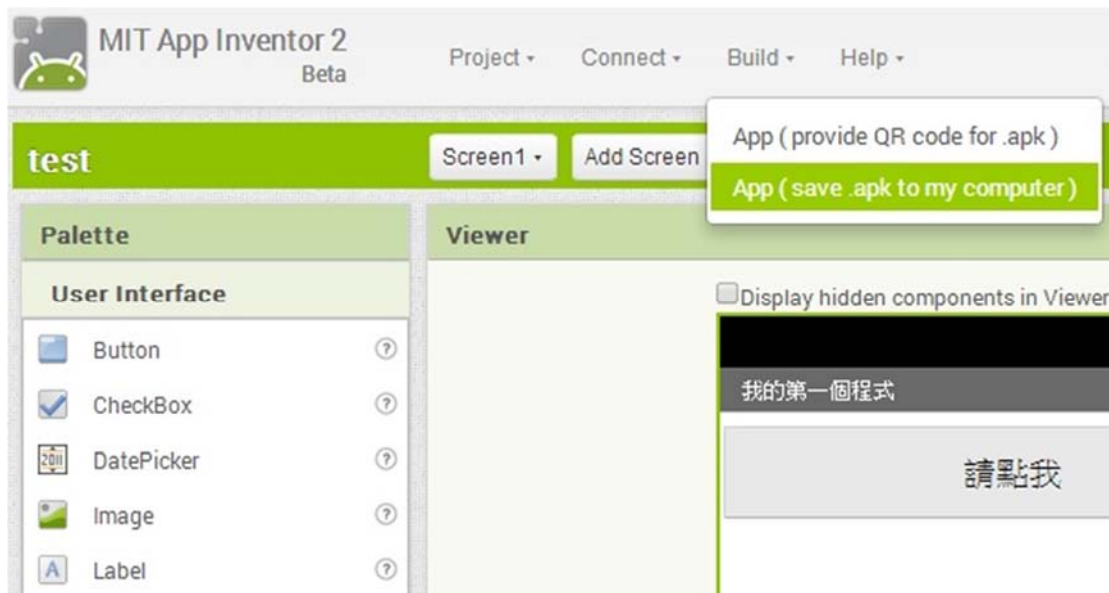


扫完 QR code 后会跳回原来的画面,这时不要紧张也不要乱按,其实是 APP 正透过网络从计算机传到手机,需等待一段时间,成功了便会出现程序安装画面,如下图,我们按点击安装。

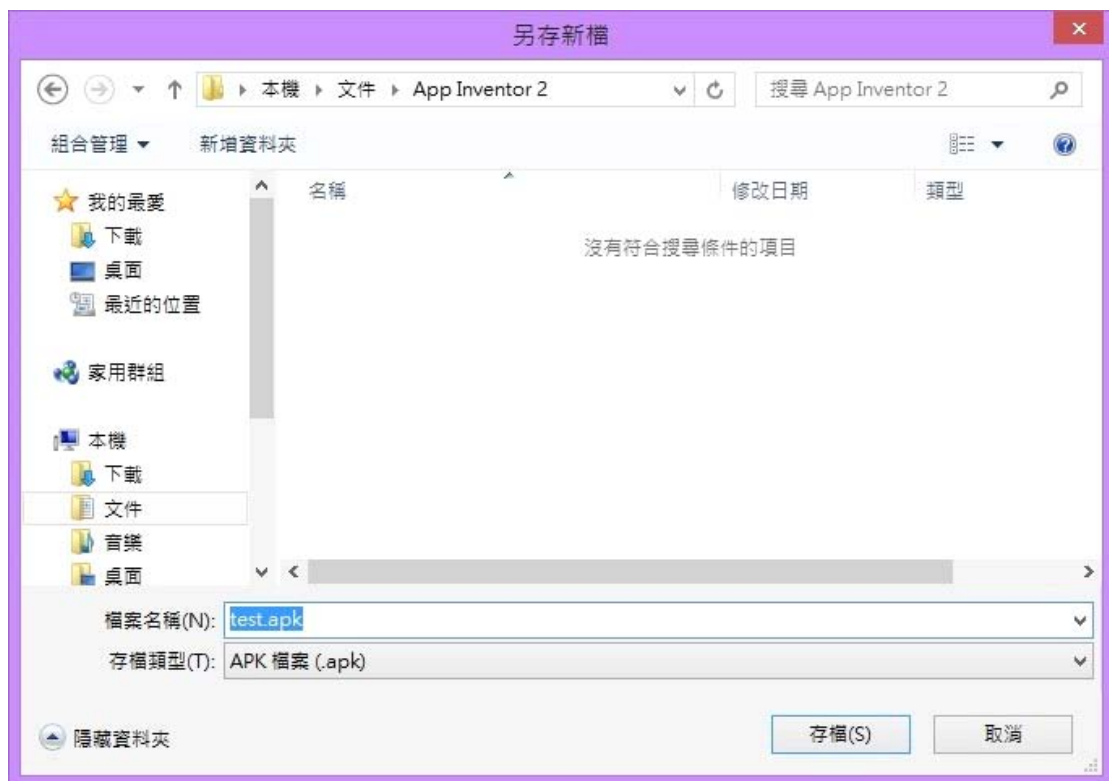


如果选择方案二, 点击 Build-> App(save .apk to my computer),

如下图。



进度条跑完之后就会跳出儲存窗口，如下图，储存之后再把它传进装置做安装，但装置必须安装档案管理 APP(有些装置会内建)，浏览至档案目录并点击安装，虽然方案二会比方案一还要费时，但多了安装档可以随时分享给其他人使用，方案一适合程序除错时使用。



安装完成后，我们就来验收结果啦，如下图，这是点击 Button 前的样子。

Screen1

請點我

Text for Label1

点击 Button 之后，Label 就变成 "你好" 了，如下图。

Screen1

請點我

你好

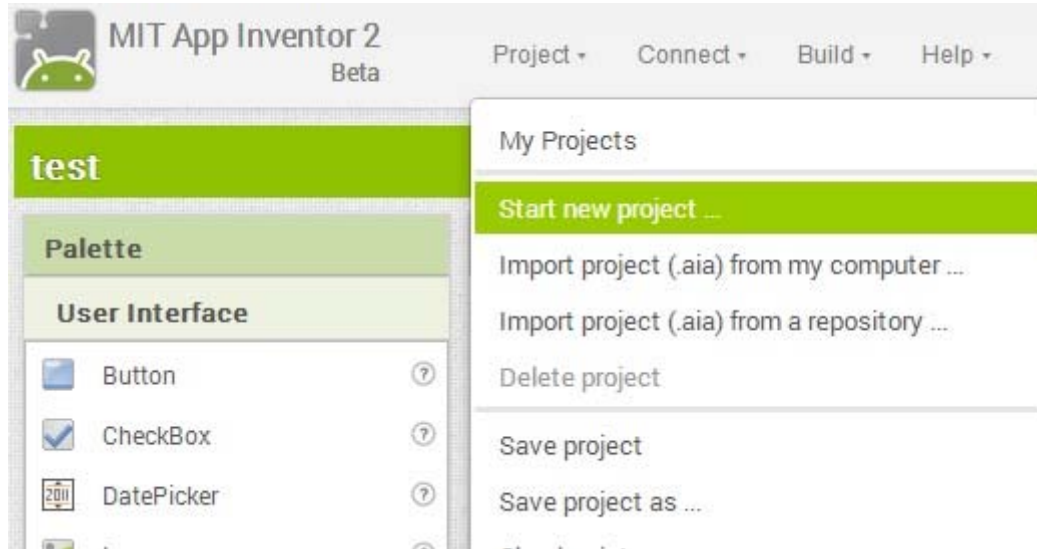
是不是觉得程序没有很完整呢？例如可以在 Screen1 属性中的 Title 项，把 "Screen1" 改成 "我的第一个程序"，还有在 Label1 属性中的 Text 项，把里面的文字去掉，这样只有在点击 button 之后才会出现文字，如下图，屏幕有没有干净许多呢？

我的第一個程式

請點我

三、第二个程序

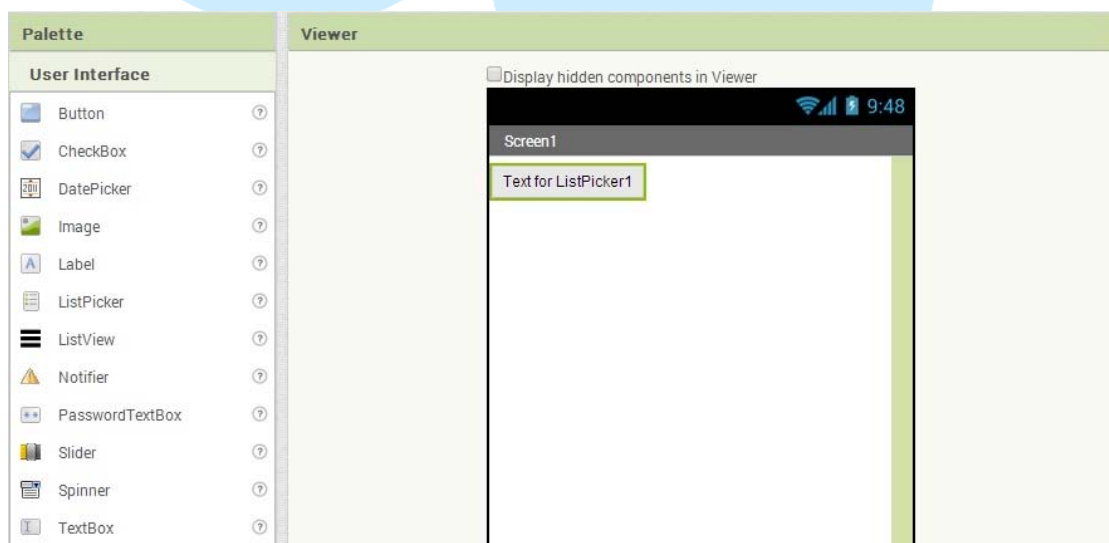
我们来做一个结合 EduCake 的程序吧，最简单方便的沟通方式就非蓝牙莫属啦～首先开新项目点击 Project-> Start new project



专案名称就取 "Bluetooth_Connect" 吧。

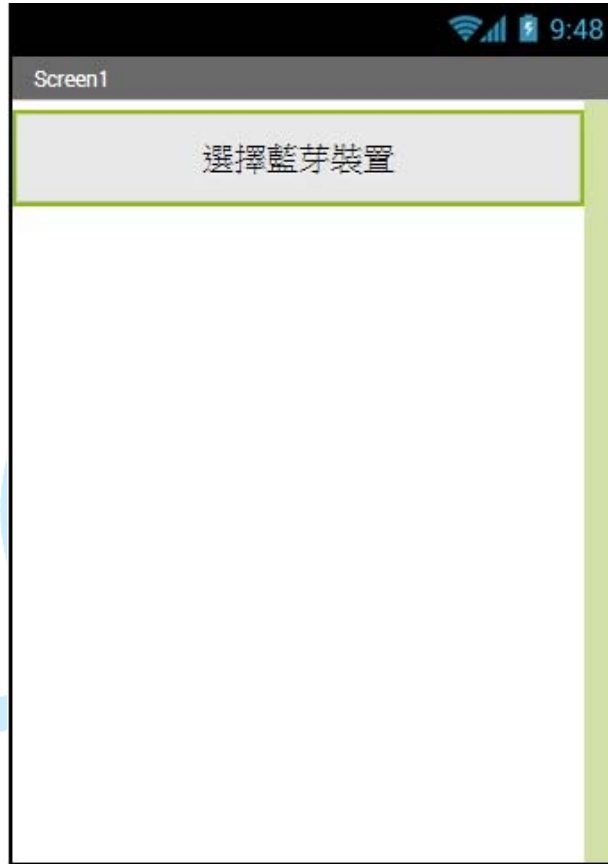


接着从左侧拖曳一个 ListPicker 组件到预览画面中。



在 Properties(属性)字段中的 FontSize 改为 20，Text 改为"选择蓝牙

装置", Width(宽) 改为 Fill parent(充满), Height(高) 改为 50 pixels, 改完之后会跟下图一样。



这里先介绍一下 ListPicker 的功能，它功能有点像 Button，不过它是具有选单的 Button，点击它就会跳出我们指定的选项，如下图，然后我们可以针对点击其中的选项再做后续的动作。

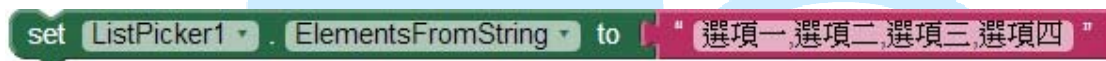


选单设定的方法有两种，一种是在 Properties(属性)字段中的

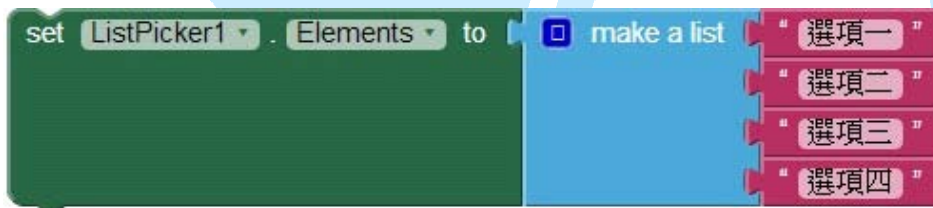
ElementsFromString 里来新增选项，如下图，选项跟选项之间用半角逗号隔开即可，使用的前提是有默认的选项才能预先新增，这里的步骤可以不用做，要测试也是可以的。



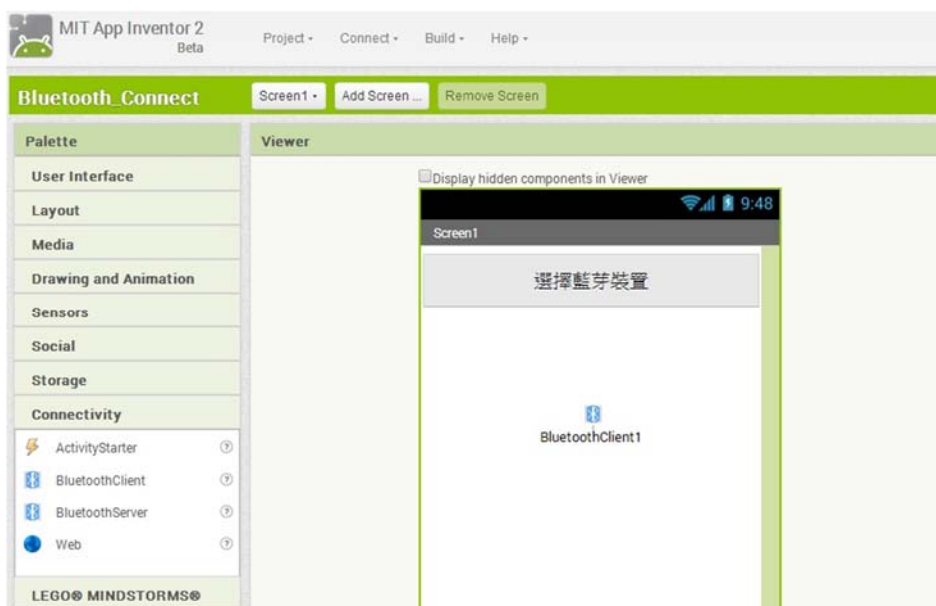
方法二，如下图，可用在程序中指定选项，弹性较高。



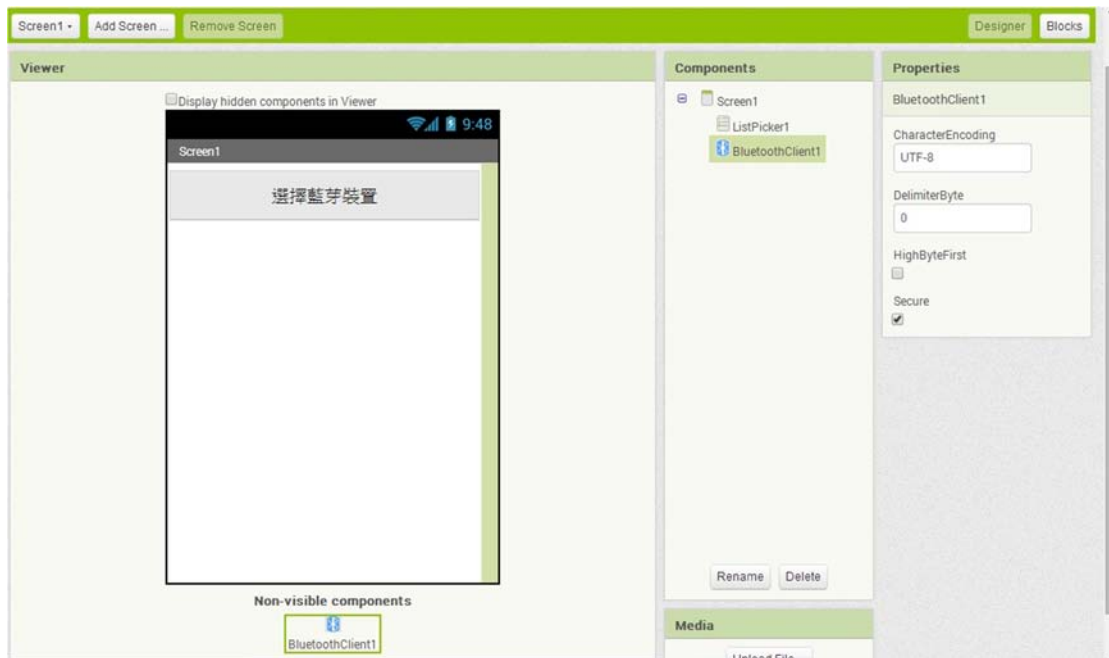
方法二的另一种表示法，一切以写程序方便就好。



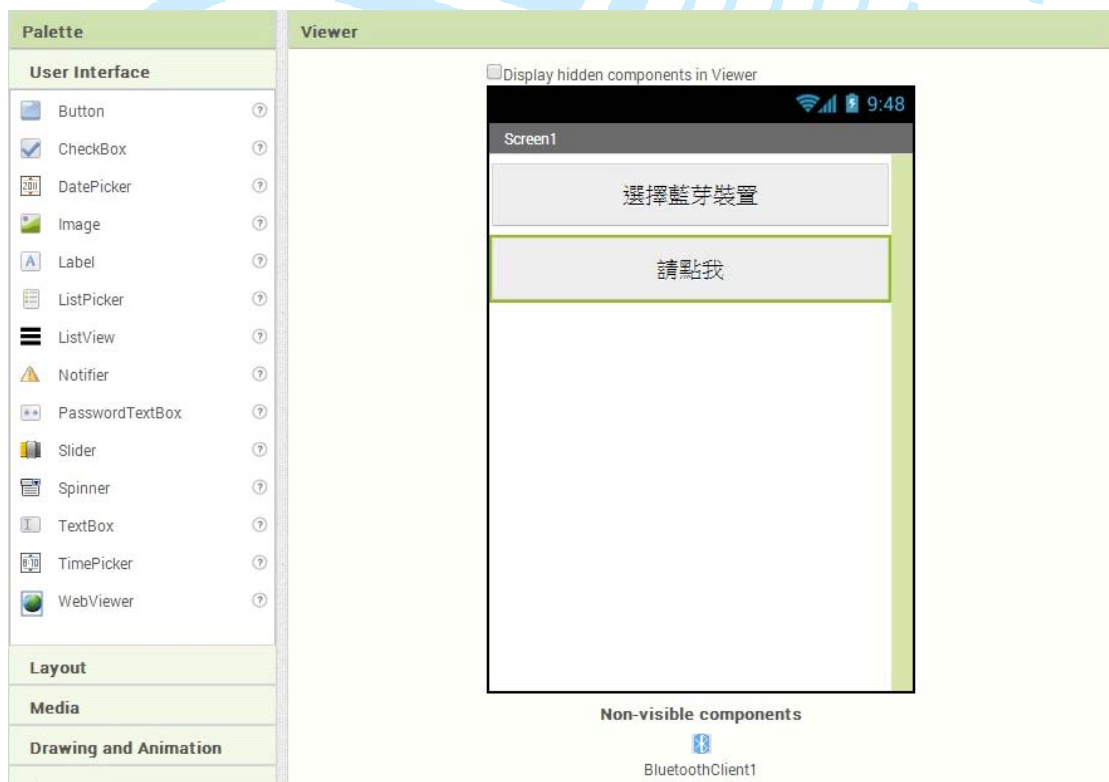
接下来进入正题，到 Palette(组件区)，点击 Connectivity 会跳出组件，接着把 BluetoothClient 拖曳到预览屏幕内，如下图。



拉进去之后，因为没有实体，所以只会显示在预览屏幕的下面，也不需要更改属性。



接下来从左侧拖曳一个 Button 组件到预览画面中，Properties(属性)字段中的 FontSize 改为 20，Text 改为"请点我"，Width(宽) 改为 Fill parent(充满)，Height(高) 改为 50 pixels，改完之后会跟下图一样。



在 Button1 的 Properties(属性)字段中 Visible 改为 hidden(隐藏)，用意是执行 APP 的时候先隐藏，等到蓝芽联机之后才做显现，接下来点击

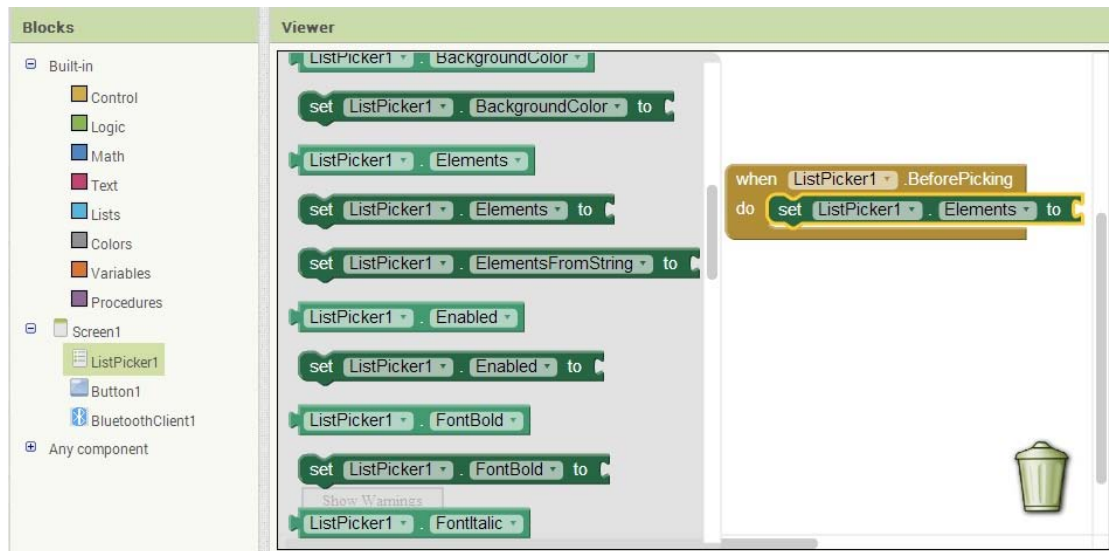
Blocks 到积木区吧。



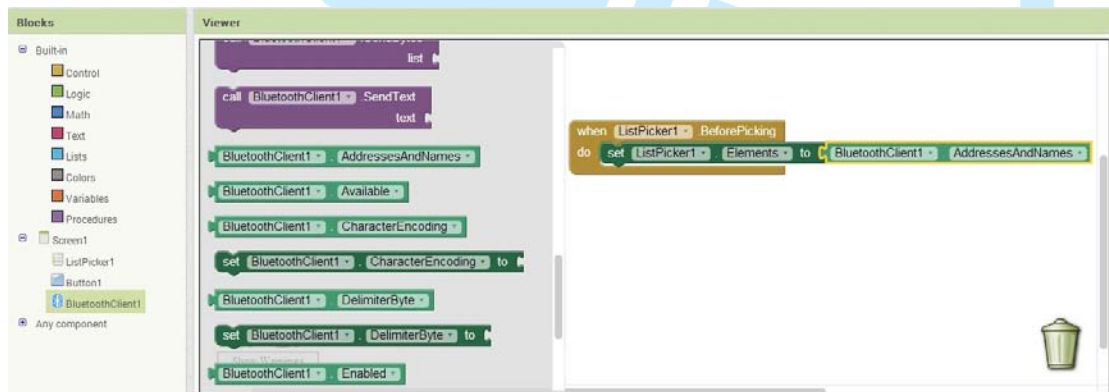
左侧点击 ListPicker1 出现控件，拖曳 BeforePicking 控件出来，如下图所示，它的触发条件是点击 ListPicker"前"会触发，我们可以派它做一些事情，例如：更改或新增 ListPicker 的选单。



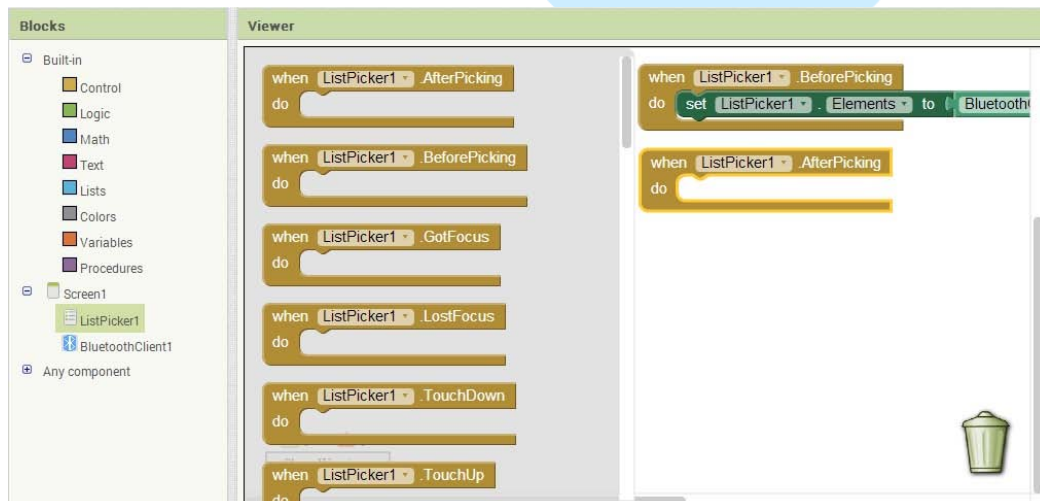
再次点击 ListPicker1 出现控件，滚动条往下拉才看的到 Elements 控件，把它拖曳出来组合进去，如下图。



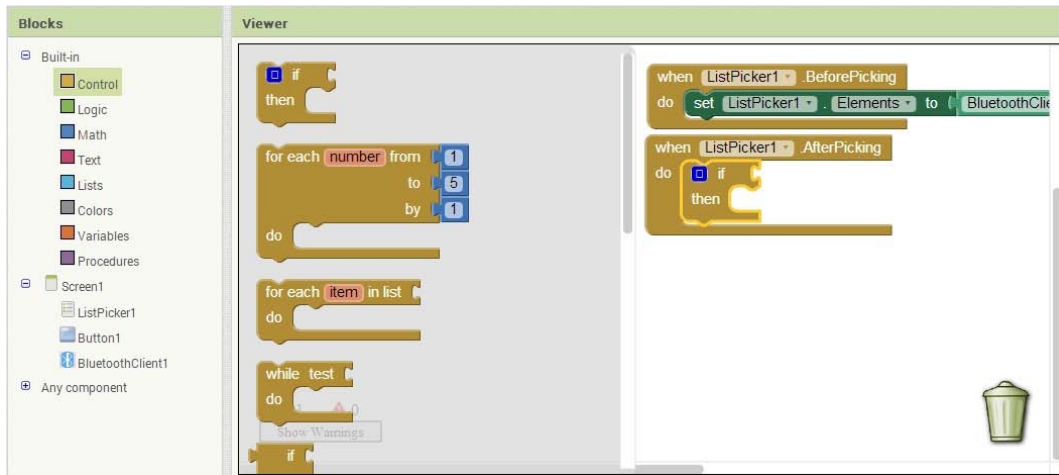
左侧点击 BluetoothClient1 出现控件，一样将滚动条往下拉，看到 AddressesAndNames，把它拖曳到右侧接上，如图，功能是将 Android 所配对过的蓝牙设备用 list 的方式列出来，我们这样做可以在程序刚开始执行的时候触发点击 ListPicker1 前的控件，然后做新增蓝牙列表的选项。



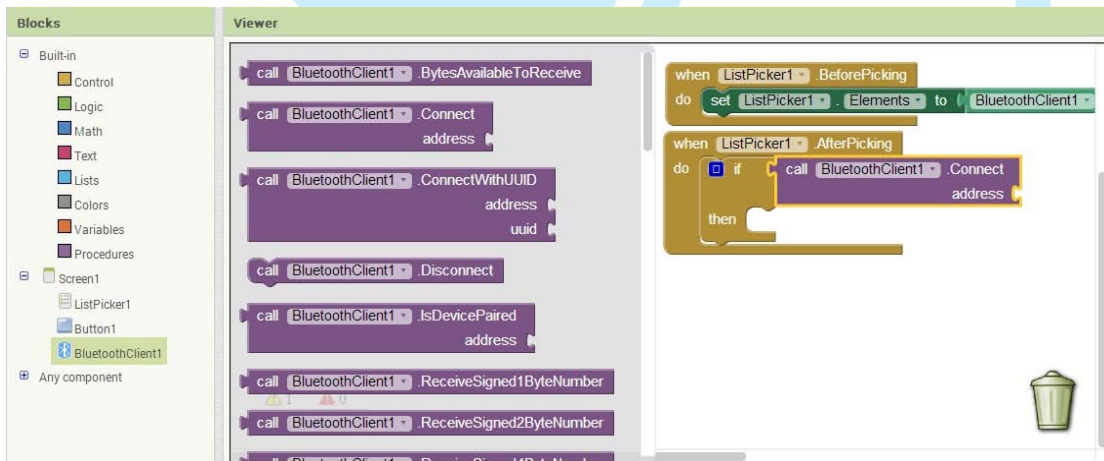
左侧点击 ListPicker1 出现控件，拖曳 AfterPicking 控件出来，如下图，它的触发条件是点击 ListPicker"后"就会触发 AfterPicking 控件。



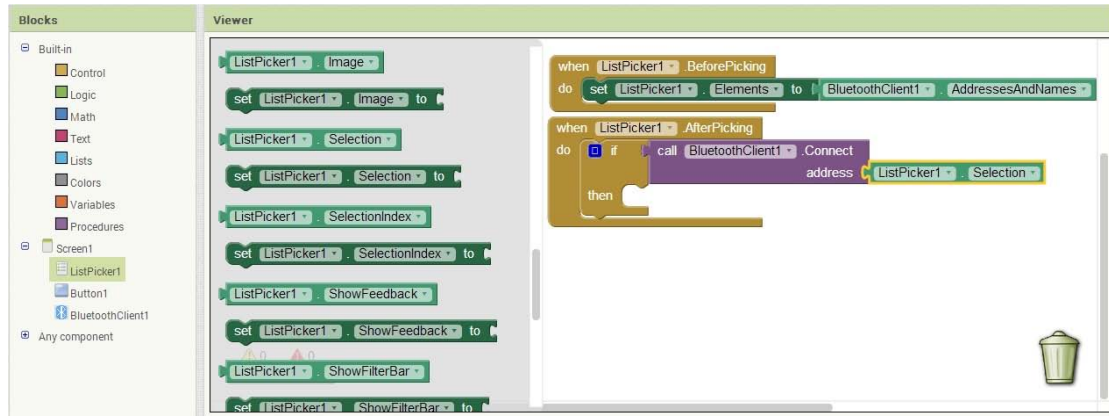
左侧点击 Control(控制), 把 if 判断式拖曳至右侧接合, 如下图。



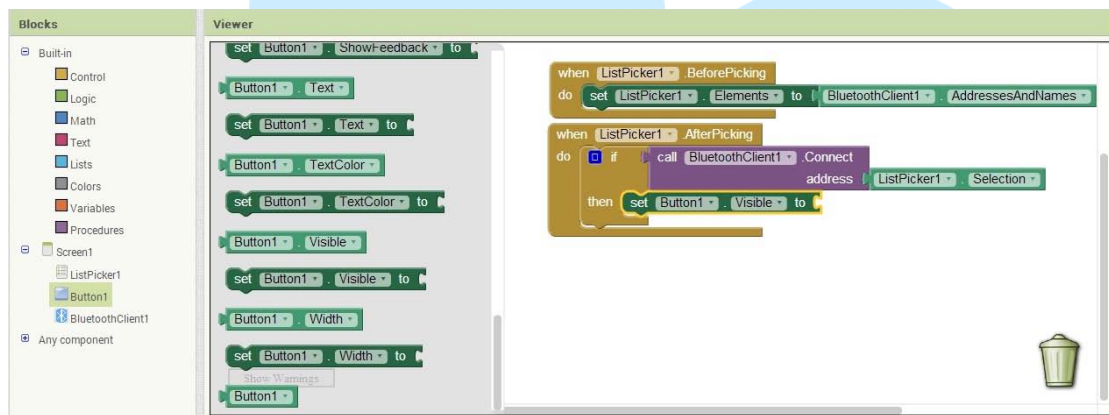
左侧点击 BluetoothClient1, 将 Connect address 控件拖曳到右侧跟判断式接合, 如下图, 功能是给它蓝牙的地址就可以联机, 成功回传 true, 失败回传 false。



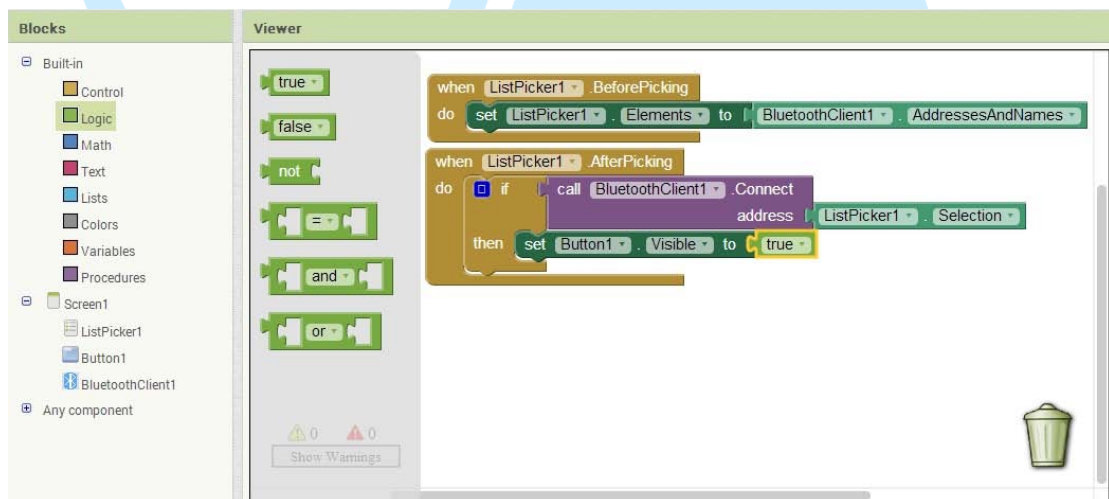
左侧点击 ListPicker1, 将 Selection 拖曳至右侧结合, 如下图, 它的功能是回传 ListPicker1 选单中我们选择的文字。



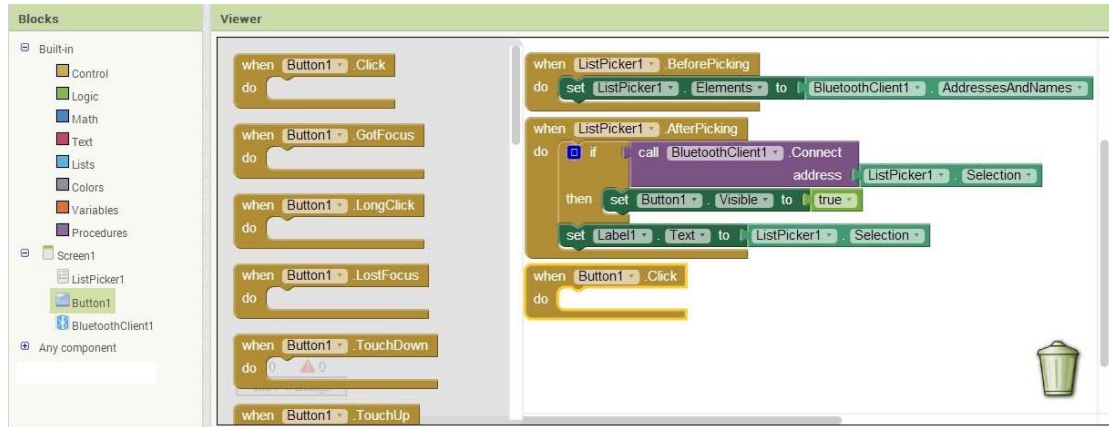
左侧点击 Button1, 将 .Visible 控件拖曳到右侧接合, 如下图, 因为我们刚开始是把 Button1 做隐藏, 当蓝牙联机成功会回传 true 去触发 if 的条件, 我们就可以把按钮显现出来了。



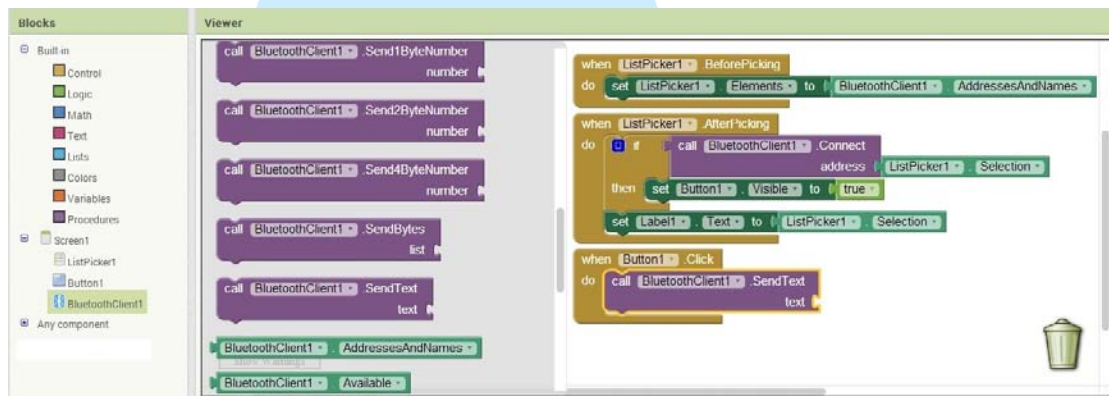
接着点击 Logic, 拖曳 true 到右侧接合, 如下图。



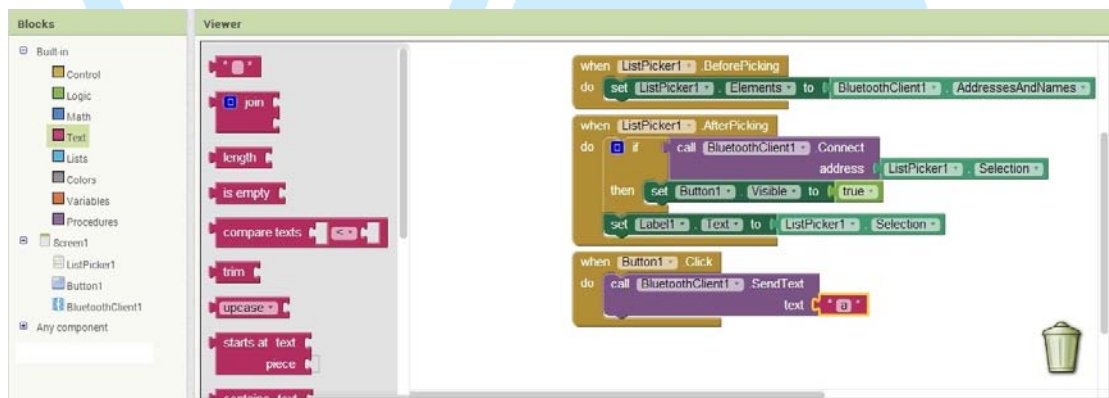
侧点击 Button1, 将 .Click 拖曳到右侧, 如下图。



接下来点击 BluetoothClient1, 将 .SendText text 拖曳至右侧接合, 如下图, 它的功能是可以传字符串给联机中的蓝芽装置。



接下来点击 Text, 我们拖曳一个空白字符串到右侧, 并在里面输入"a", 如下图, 这样每按一次 Button 就会传"a"的字符给联机中的蓝芽装置。



接下来介绍 EduCake 蓝芽模块的部分, 这里使用的蓝芽模块是较为大众化的: HC-06。

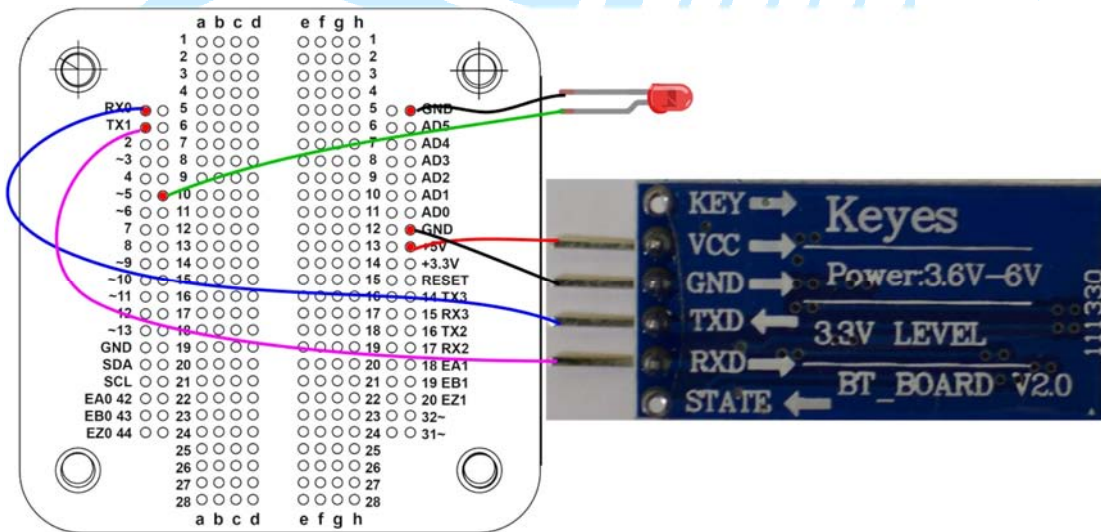
HC-06 正面



HC-06 反面



将 HC-06 接上 EduCake，再接上一颗 LED，接脚图如下：



接好电路以后，我们先来设定 HC-06 的一些参数


```
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());
}
```

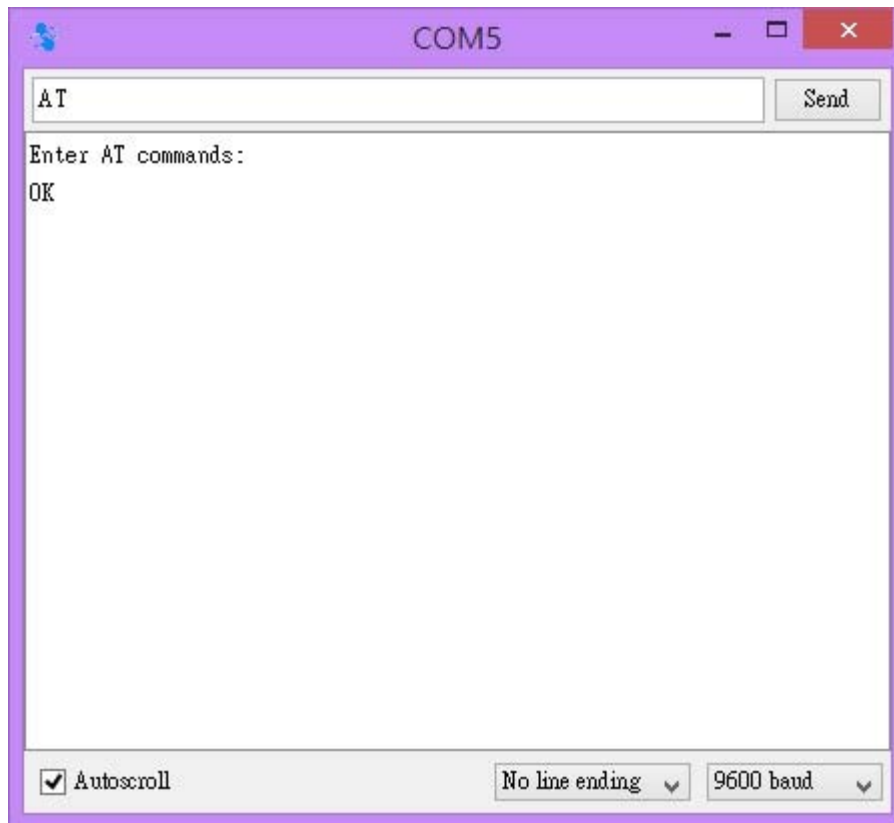
首先开启 IDE 接口，输入以下程序:程序上传好了以后，就可以利用 IDE 接口里面的 COM 监视器(Serial Monitor)来和蓝芽通讯，画面如下



到底要输入什么哩？其实 HC-06 在通电未联机情况下，灯光会闪烁并处于 AT command mode，当在这个状态的时候，我们就可以任意更改名称、密码，甚至连通讯速度(baud rate)都可以改，以下介绍几个常用的 AT 指令：

- AT : 测试是否正常工作，响应「OK」
- AT+NAMEaaa: 将装置名称改为「aaa」
- AT+PIN1234 : 将联机密码换为「1234」
- AT+VERSION : 响应韧体的版本
- AT+BAUD1 : 将 baud rate 换为 1200
- AT+BAUD2 : 将 baud rate 换为 2400
- AT+BAUD3 : 将 baud rate 换为 4800
- AT+BAUD4 : 将 baud rate 换为 9600
- AT+BAUD5 : 将 baud rate 换为 19200
- AT+BAUD6 : 将 baud rate 换为 38400
- AT+BAUD7 : 将 baud rate 换为 57600

上面三个 AT 指令比较常用到，如果要改名称为 abc，就输入 AT+NAMEabc，成功会显示 OKsetname，设定密码为 1234，就输入 AT+PIN1234，成功会显示 OKsetPIN，这里要注意的是更改名称之后要断电重开才会生效而且不能使用中文(会乱码)，而改完密码不需重开即生效。



刚刚做出蓝牙联机的 APP, 再加上设定完 HC-06, 终于要进入正题啦, 我们要做出用手机控制一个 LED 灯, 程序代码如下:

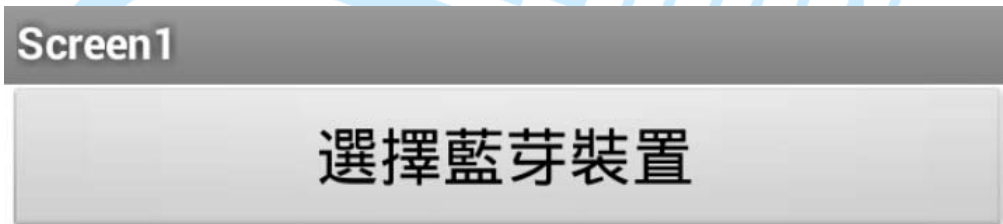
```
char ch;
int LED=0; // 儲存 LED 目前狀態
void setup()
{
    Serial.begin(9600); // 這行主要是設定和電腦的 COM 通訊
    的速度
    Serial1.begin(9600); // 這行主要是設定和藍芽 通訊的速度
    pinMode(5, OUTPUT); // 設定腳位 3 為輸出模式
}

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;
            }
        }
    }
}
```

程序上传之后，我们转移阵地到 Android 装置中吧，首先确认装置已经跟 HC-06 配对完成，如下图，本书以 HC-06 作为蓝芽名称方便辨识。



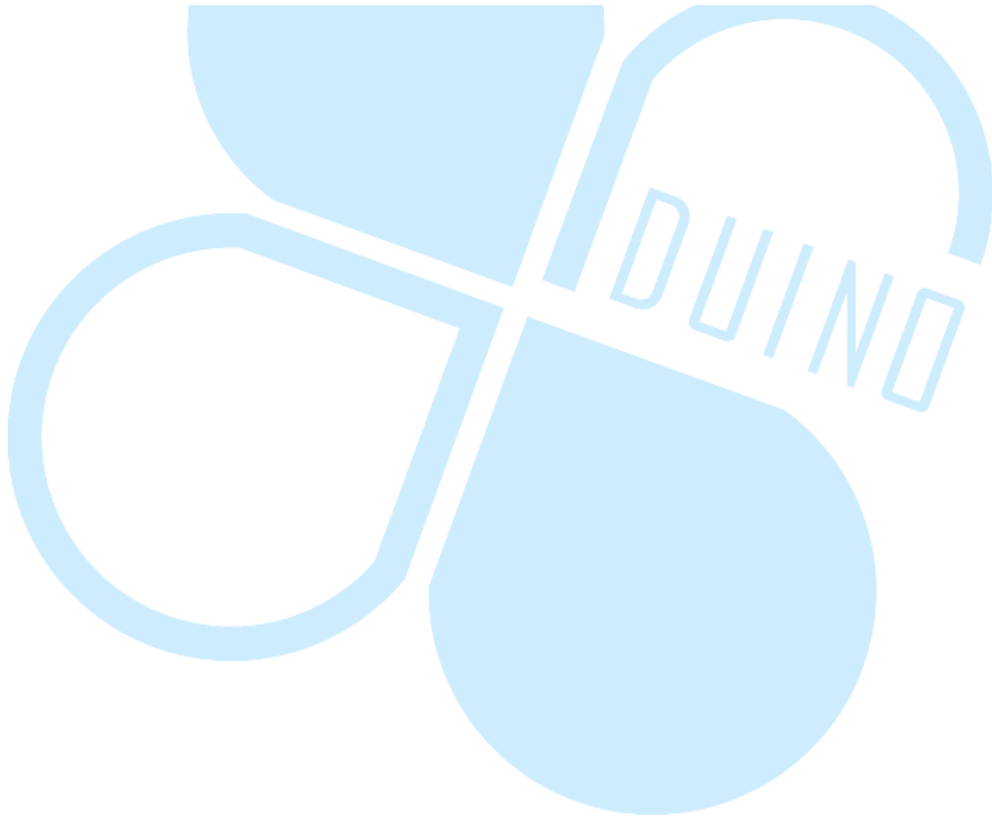
接着开启程序，如下图，点击"选择蓝芽装置"。



跳出蓝芽选单，如下图，请选择 HC-06 蓝芽模块联机。

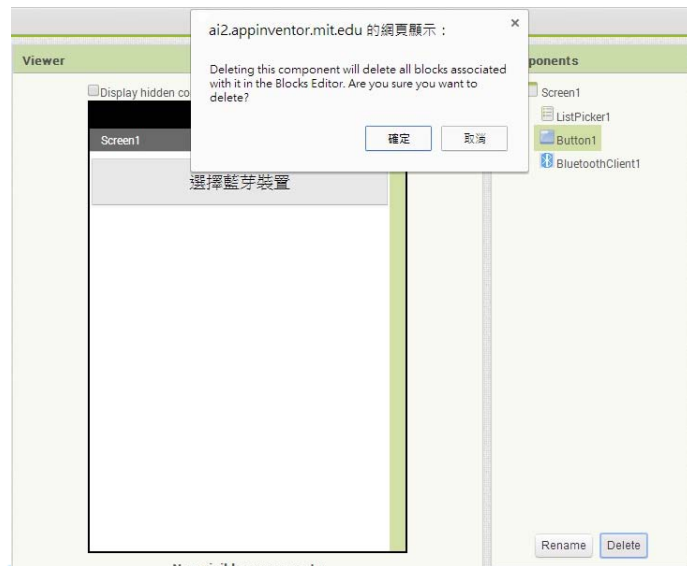


联机成功后会出现"请点我"的 Button，如下图，可以观察到 HC-06 从原本的闪烁变化为恒亮，接下来点击"请点我"，EduCake 上的 LED 灯是否亮起来了呢？再点一下就会熄灭了。

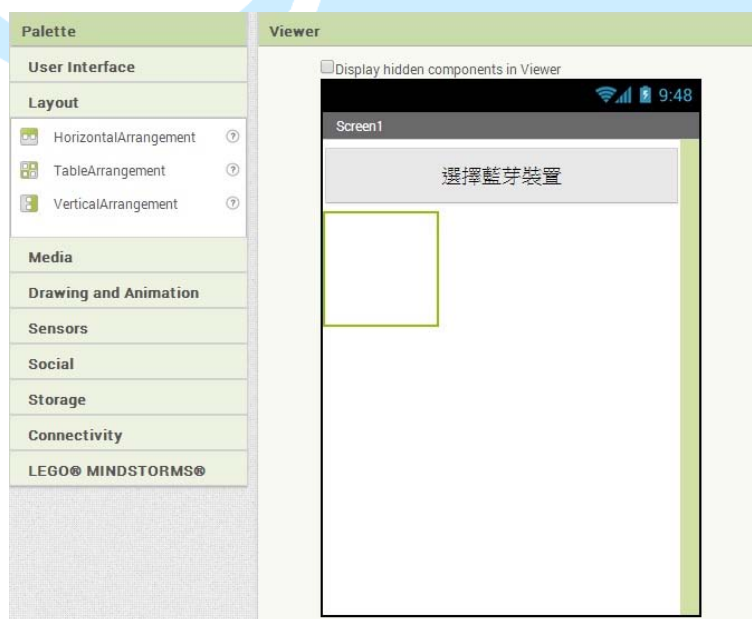


四、第三个程序

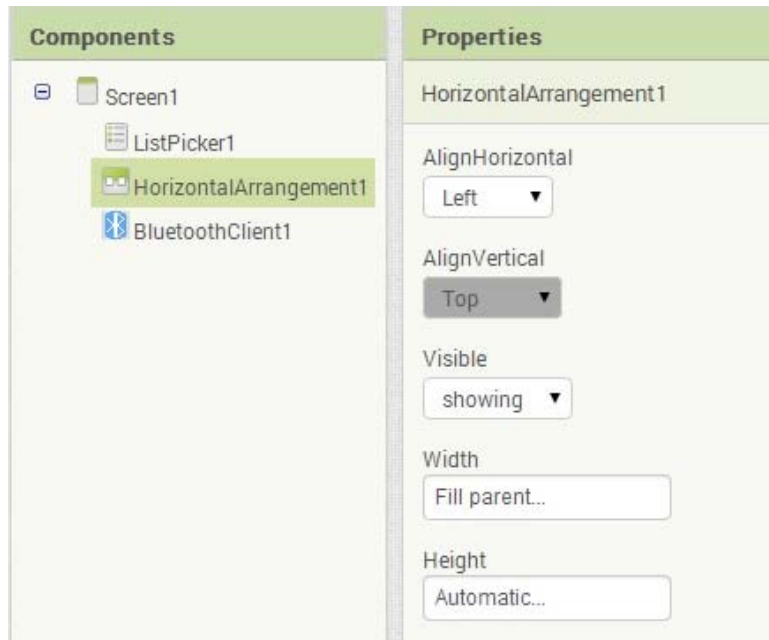
我们再度回到 App Inventor 中，延续刚刚的程序继续修改，如果想保留刚刚的作品，可以利用 Project-> Save project as ... 来另存一个新项目做编辑，接着删除原本的 Button1，如下图。



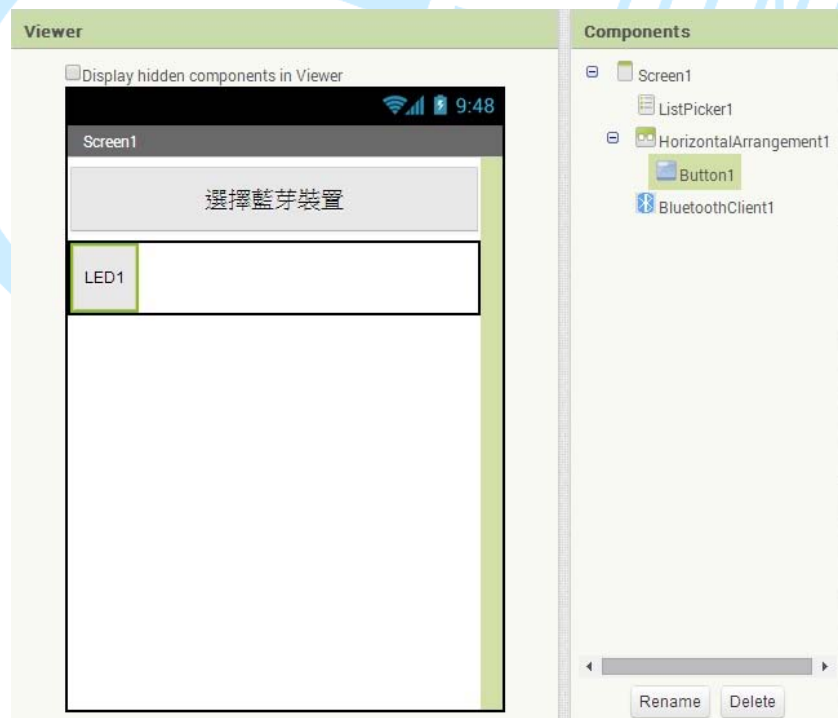
在刚刚的程序中只有控制一颗 LED，这次我们要一次控制五颗，并在 EduCake 上加入 3 颗按钮，当想要把五颗按钮摆成横的却没办法，这时候就需要用到 HorizontalArrangement，在 Layout 中，将它拖曳至预览屏幕中，如下图，这个组件的概念有点像图层管理，在框框里面的原件就只能在里面活动，并随着 HorizontalArrangement 隐藏而隐藏，是个方便管理接口的好工具。



在 HorizontalArrangement 的属性中, 将 Width(宽)改为 Fill parent, 如下图。



把 Button 拖曳到刚刚的框框内, 在属性中, 将 Text 改为 LED1, Height(高)改成 50 pixels, 如下图。



跟刚刚一样的动作, 再拉四颗按钮出来, 属性的部分除了名字之外其他

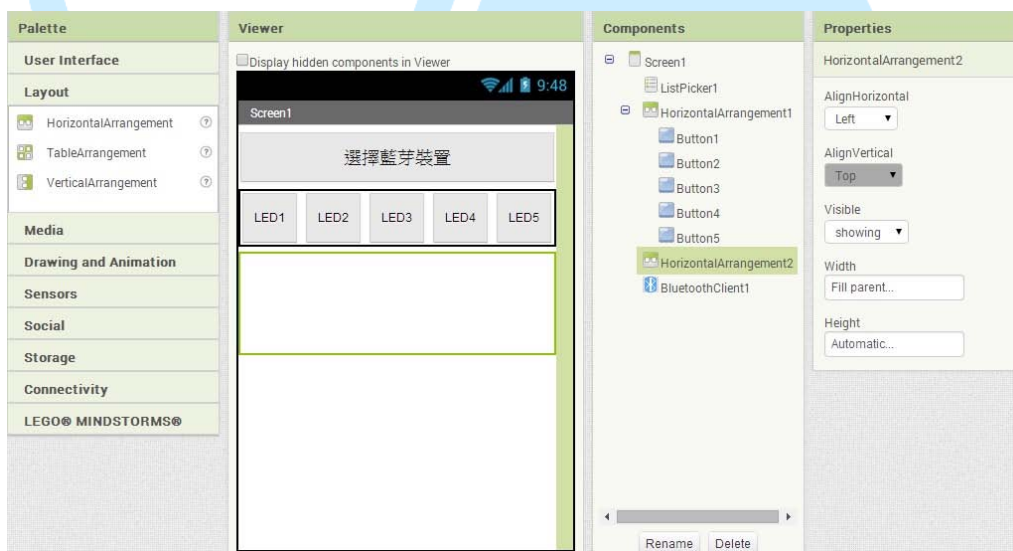
都相同，如下图。



感觉按钮是不是没有排正呢？再点击每个 button，在属性中的 Width(宽)改为 Fill parent，改完之后就会如下图一样。

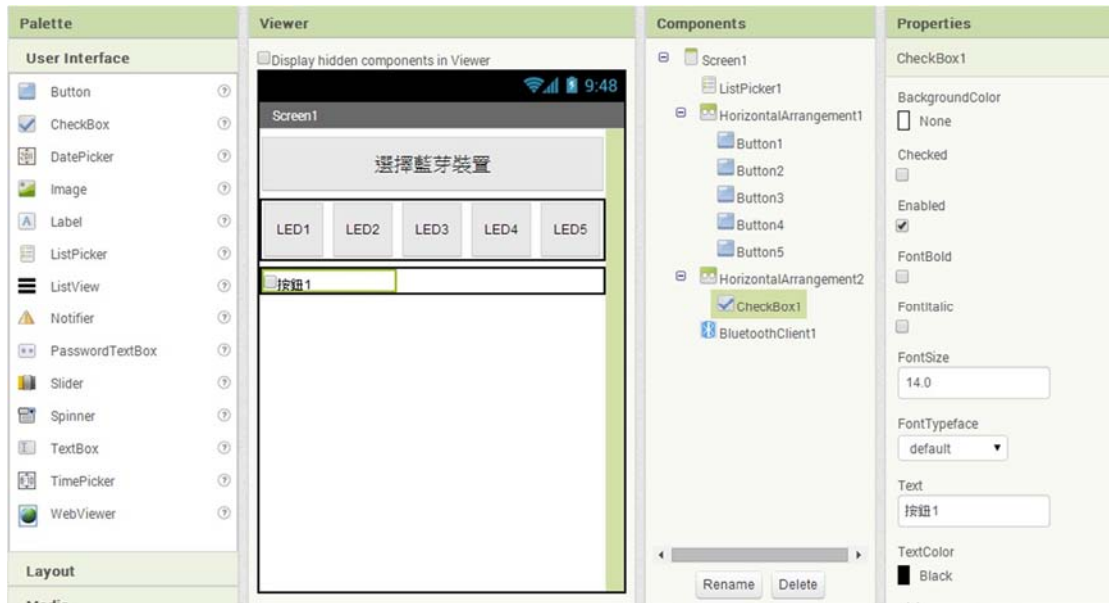


接下来再拖曳一个 HorizontalArrangement 到预览屏幕中，并在属性中的 Width(宽)改为 Fill parent，如下图。



在左侧点击 User Interface 接着拖曳一个 CheckBox 到刚刚的框框中，在属性中将 Text 改为按钮 1，如下图，它的功能是左侧的框框可以打勾也

可以取消勾选，我们这里是用来要观看 EduCake 的按钮是否有被按下。



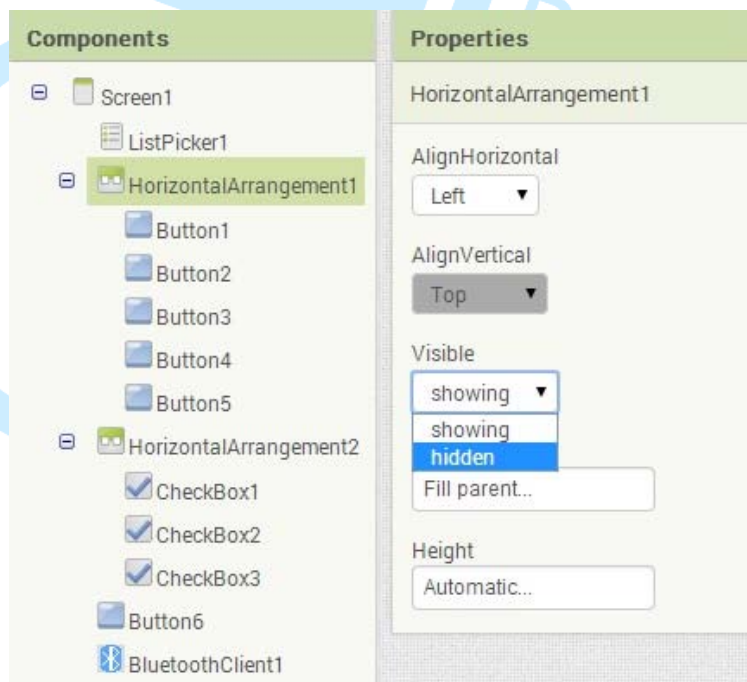
继续刚才的动作再拖曳两个 CheckBox, 属性中的 Text 分别改为按钮 2 及按钮 3, 最后再将三个 CheckBox 属性中的 Width(宽)都改为 Fill parent, 如下图, 此动作是为了要让组件与画面对齐。



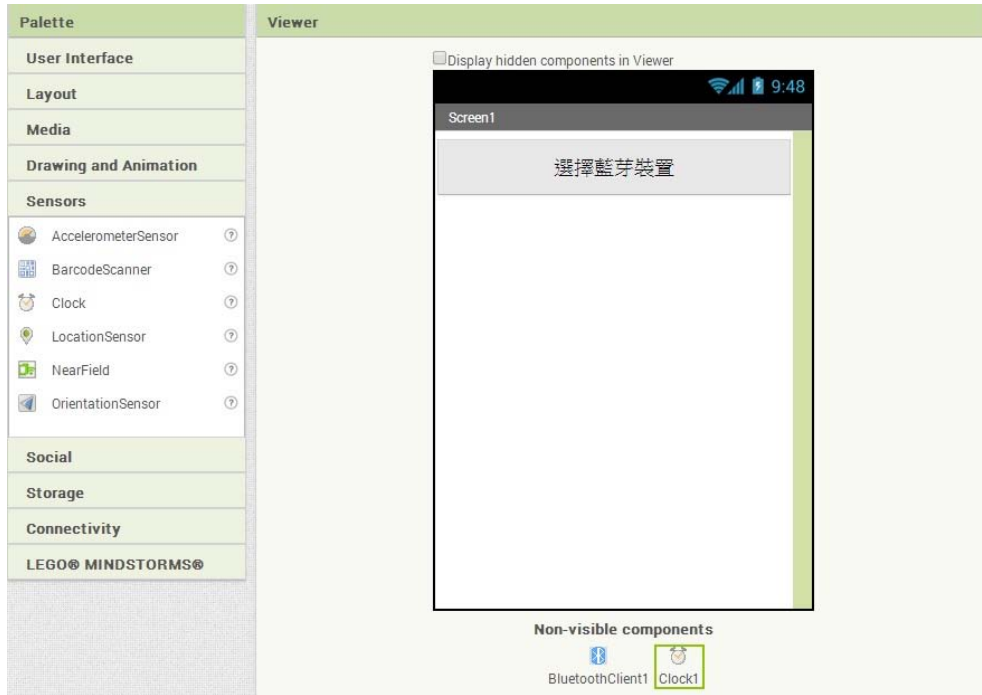
之后再拉一个 Button 到预览屏幕中, Text 改为中断联机, Width(宽)改为 Fill paren, 如下图。



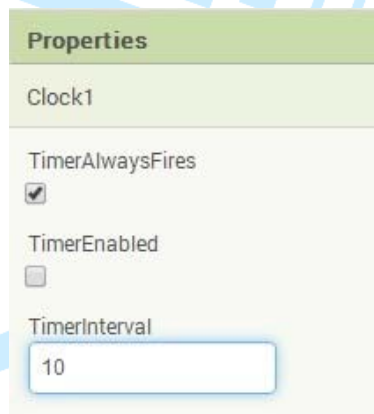
因为蓝牙在联机之前除了联机的 button 显示之外，其他的组件都要隐藏起来，所以我们将 HorizontalArrangement1、HorizontalArrangement2 及 Button6 都隐藏起来，将属性中的 Visible 改为 hidden，如下图。



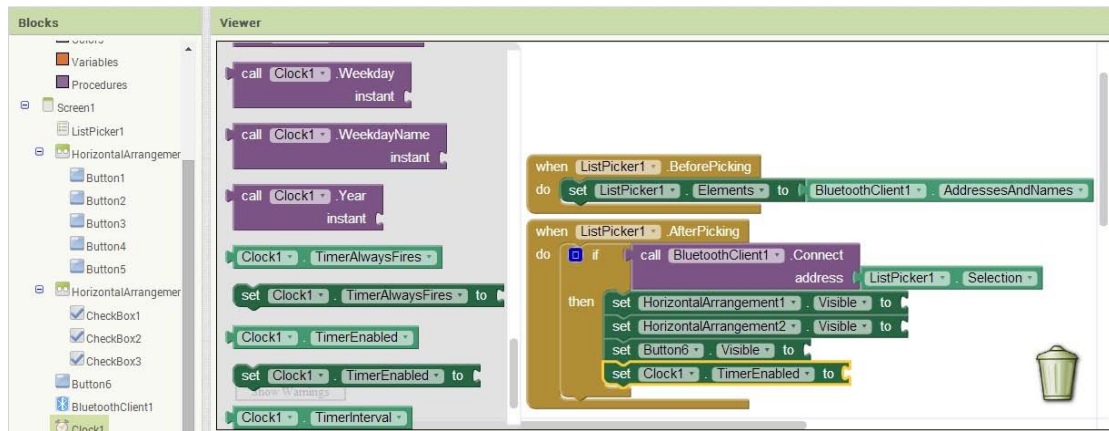
接着在左侧 Sensors 底下有 Clock 组件，将他拖曳至预览屏幕内，因为它也是没有实体所以只能显示在下面。



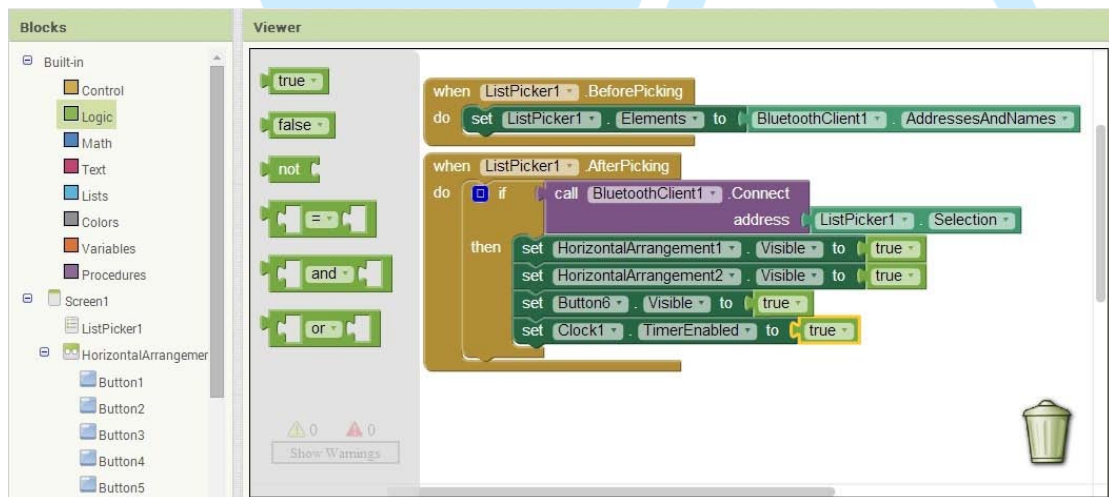
将属性中的 TimerEnabled 取消勾选，我们不让他一开始就自动执行，TimerInterval 是跑完一次之后的时间间隔，我们将它改为 10，单位为毫秒，如下图，接下来点击 Blocks 到积木区吧。



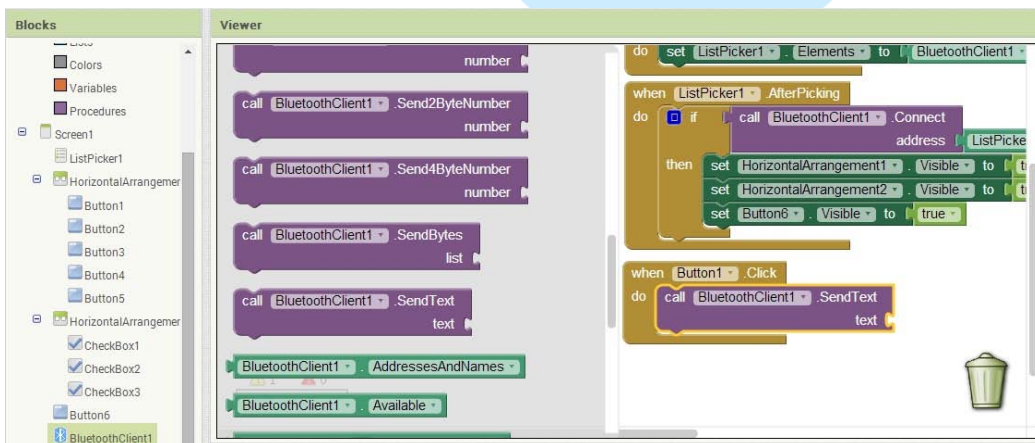
跟上一个程序一样，我们联机之后要显现刚刚隐藏的组件，所以我们把 HorizontalArrangement1、HorizontalArrangement2 及 Button6 的.Visible 控件拉出来，而 Clock1 是拉.TimerEnabled 出来，如下图。



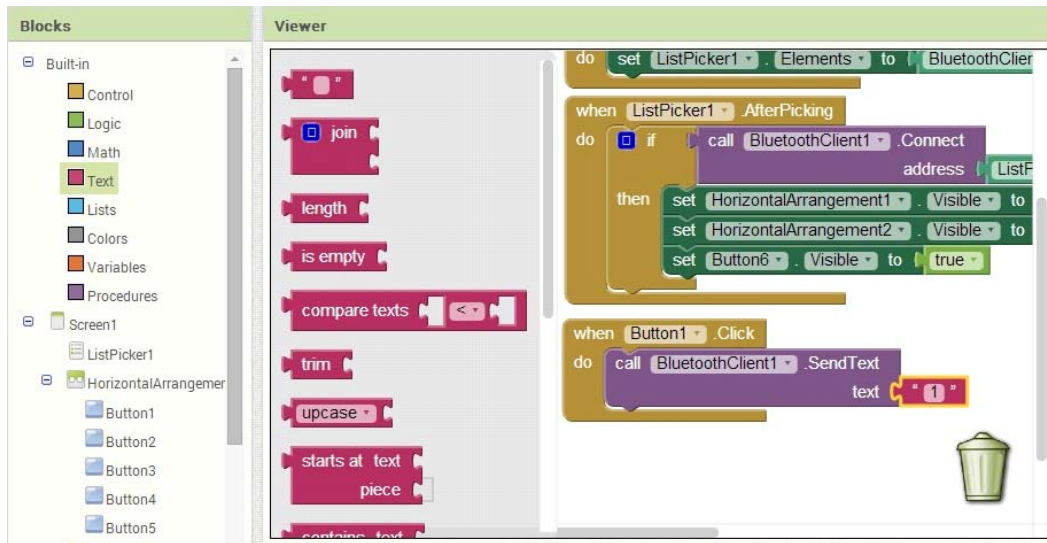
从左侧 Logic 中拖曳三个 true 过去接合，如下图。



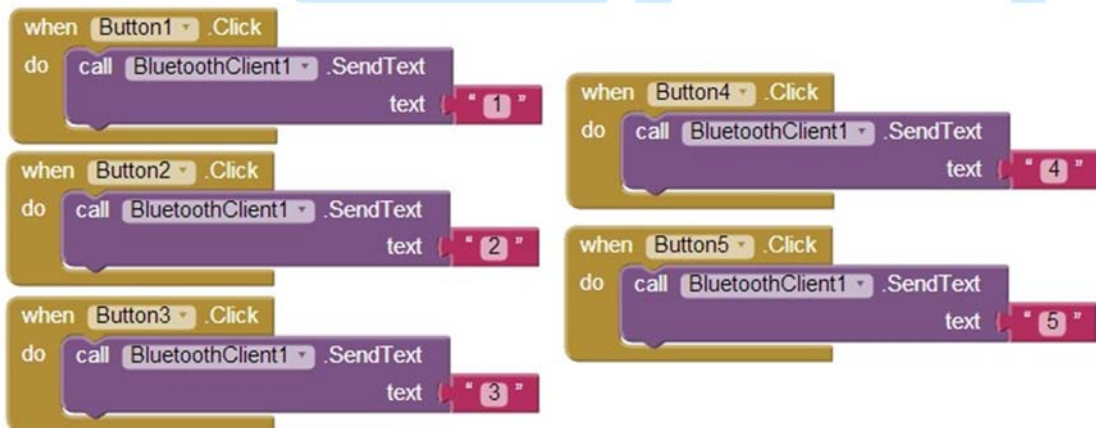
接下来要制作用五个按钮去控制 EduCake 上的 5 颗 LED，我们先拖曳 Button1 .Click 出来再拖曳蓝芽的 .SendText 出来接合，如下图。



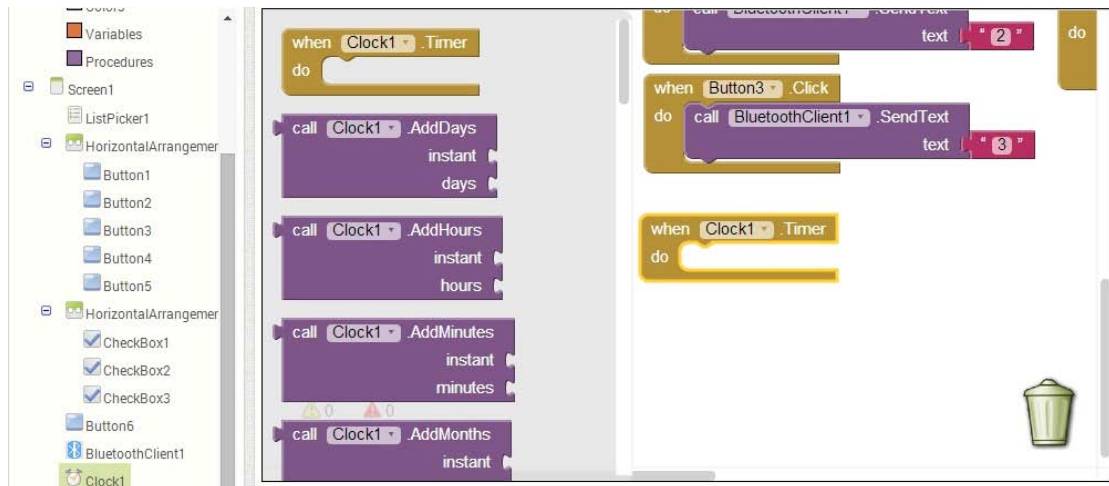
从左侧 Text 中拖曳空白的 text 过去接合，并在里面输入 1，如下图。



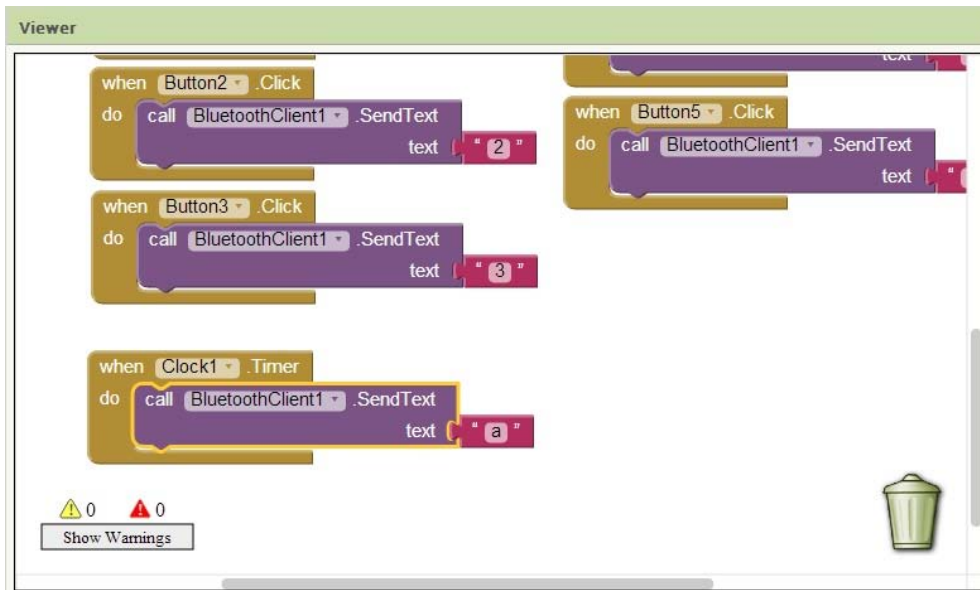
接下来就如法炮制出另外四个，并依顺序将 Text 控件改为 2、3、4、5，如下图，这样的用意是让等下撰写 EduCake 程序时能方便判断 Android 装置到底按了哪个按钮。



接下来要制作 Android 接收 EduCake 传回来按钮的讯号，左侧点击 Clock1 拉出.Timer 出来，如下图。



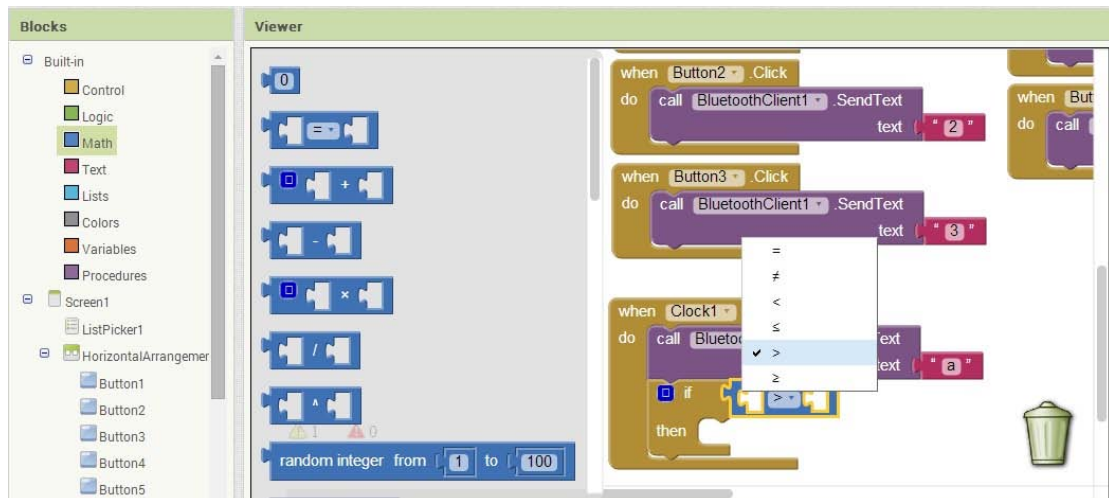
接下来我们可以点击上面的.SendText做 Ctrl + C 复制, Ctrl + V 贴上, 然后再拖曳进去 Timer 做接合, 再将 Text 的部分改为 a, 如下图。



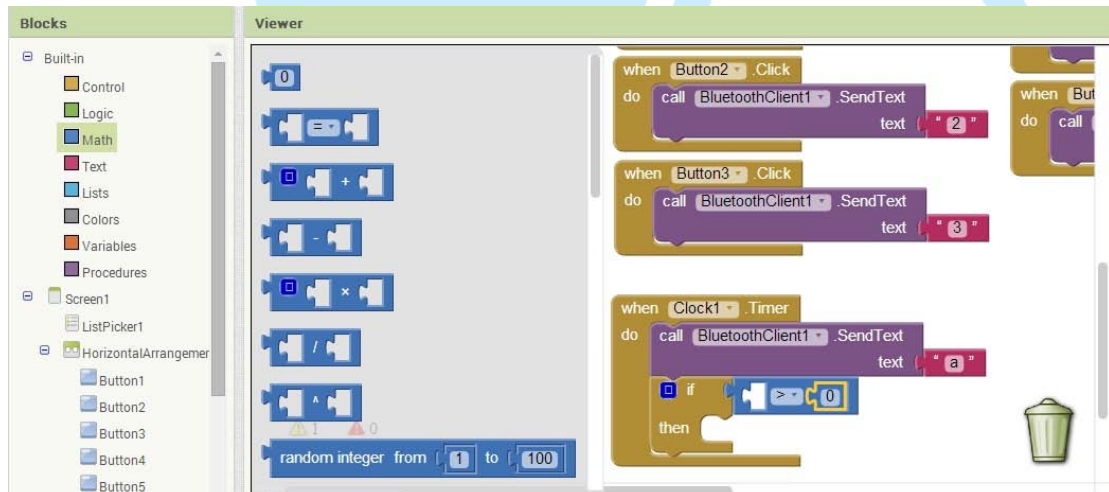
然后从左侧的 Control 中拉出一个 if 判断式接在下面, 如下图。



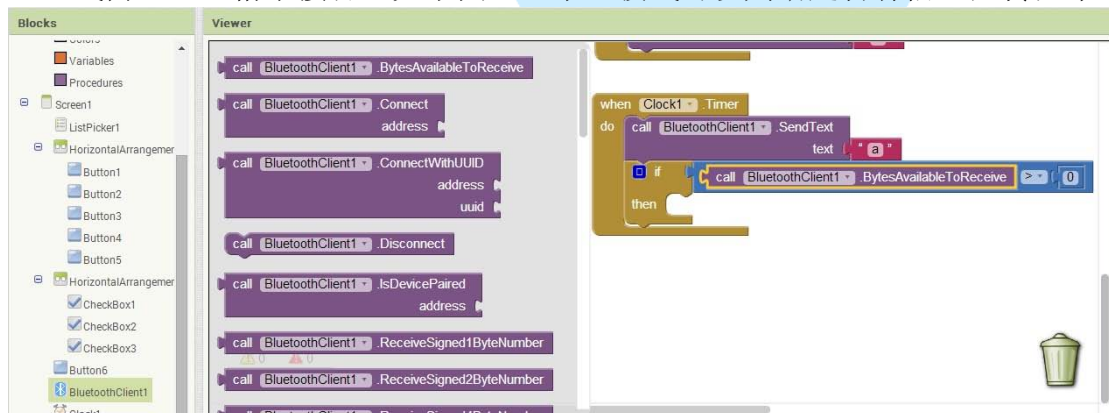
从左侧的 Math 中拉出一个比较式接在 if 的判断中，并点击中间改成 ">"，如下图。



从左侧 Math 拉出数字项，拖曳至比较式内右边空格中接合，如下图。

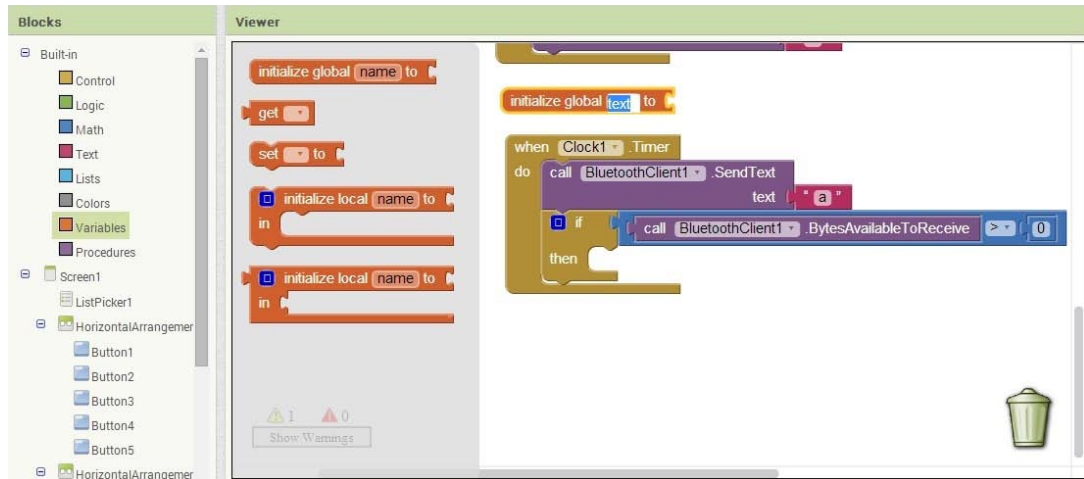


从左侧 BluetoothClient1 拉出.BytesAvaliableToReceive, 拖曳至比较式内左边空格中接合，如下图，这串比较式可以判断是否有信息回传回来。



当有数据回传时，我们就要定义一个空间去储存它，从左侧 Variables 中拖曳 initialize global name to 出来到右侧放哪里都可以，并把 name 改成 text，方

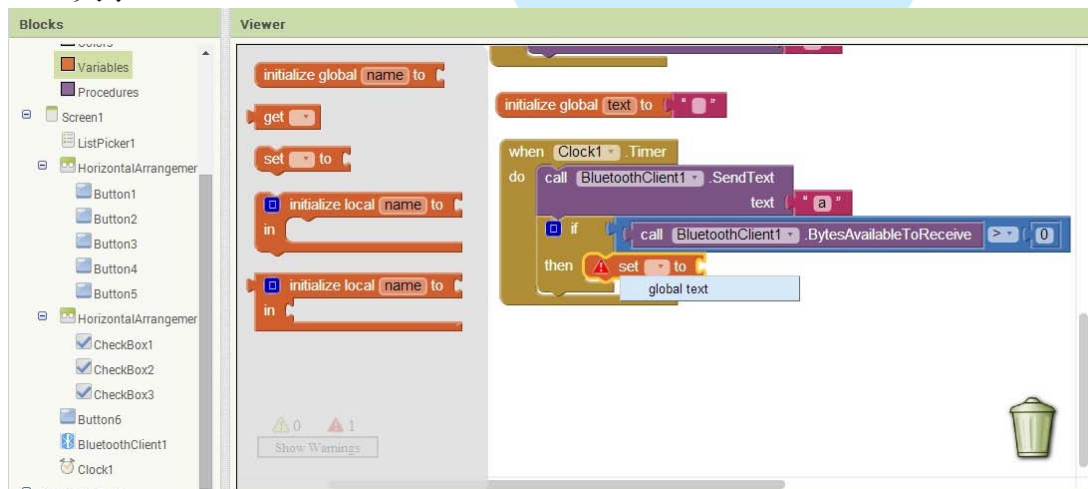
便程序阅读，如下图。



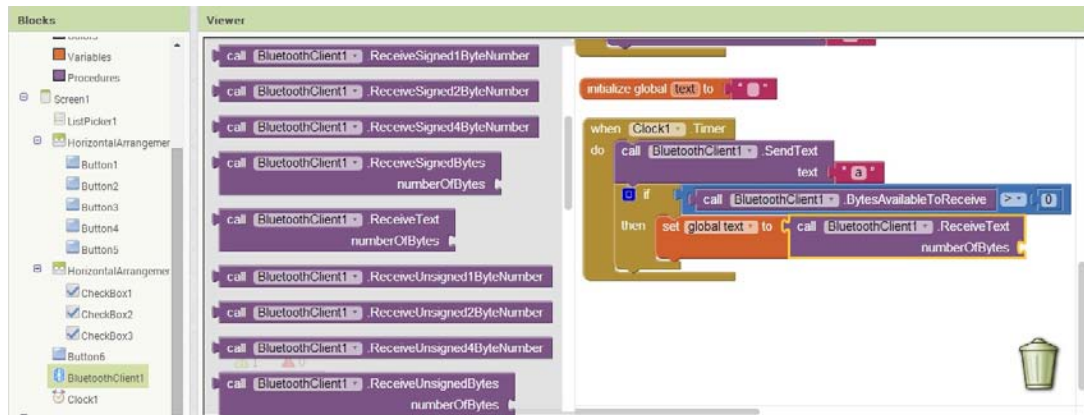
跟任写何程序一样，我们一定要赋予变量型别，我们就从左侧的 Text 拖曳一个空白的 Text 出来接在后面，如下图。



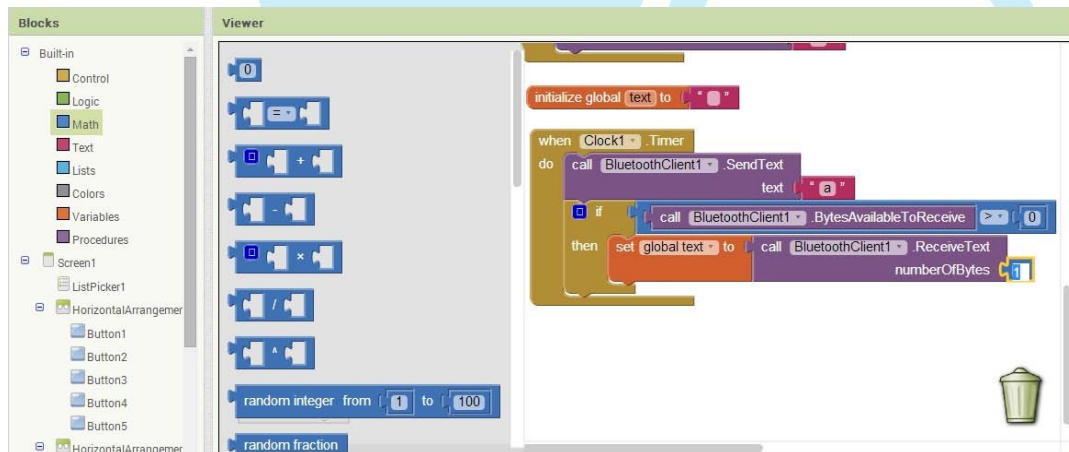
接着从左侧 Variables 中拖曳 set to 控件出来接在 if 判断式内，如下图，并点击中间空白的部分选择刚刚的字符串变量，这样红色惊叹号就会消失了。



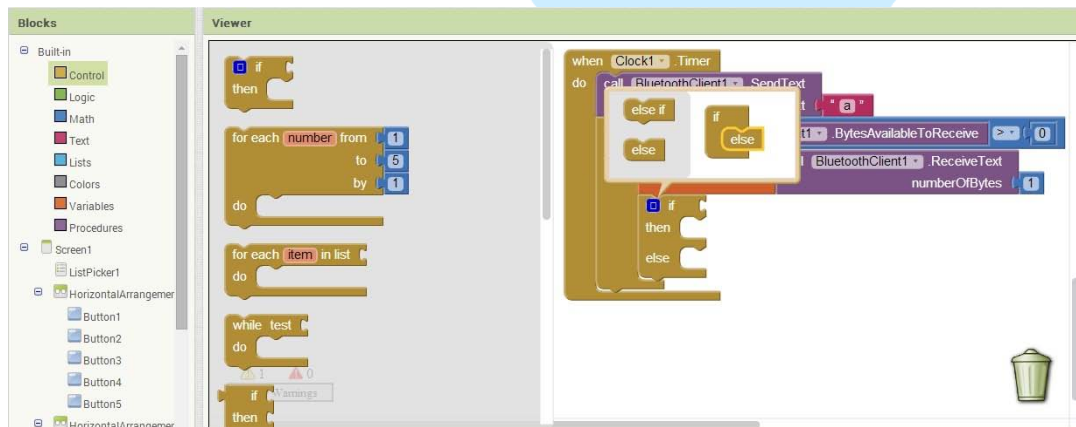
从左侧 BluetoothClient1 拉 .ReceiveText numberOfBytes 出来，接合在 set global text to 后面，如下图，它的功能是接收并回传所指定的字符串长度。



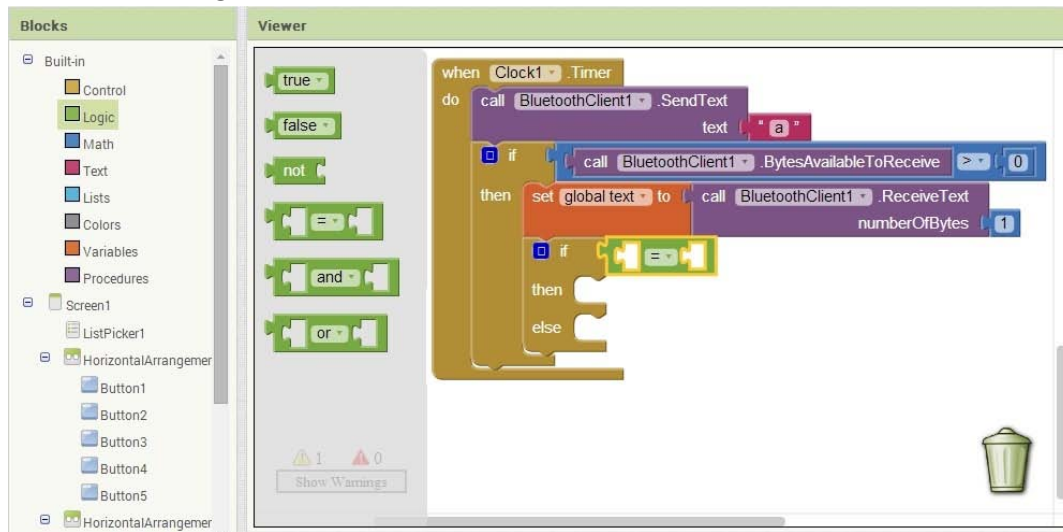
从左侧 Math 拉出数字接在.ReceiveText numberOfBytes 后面，并改为 1，我们只要接收一个字符即可



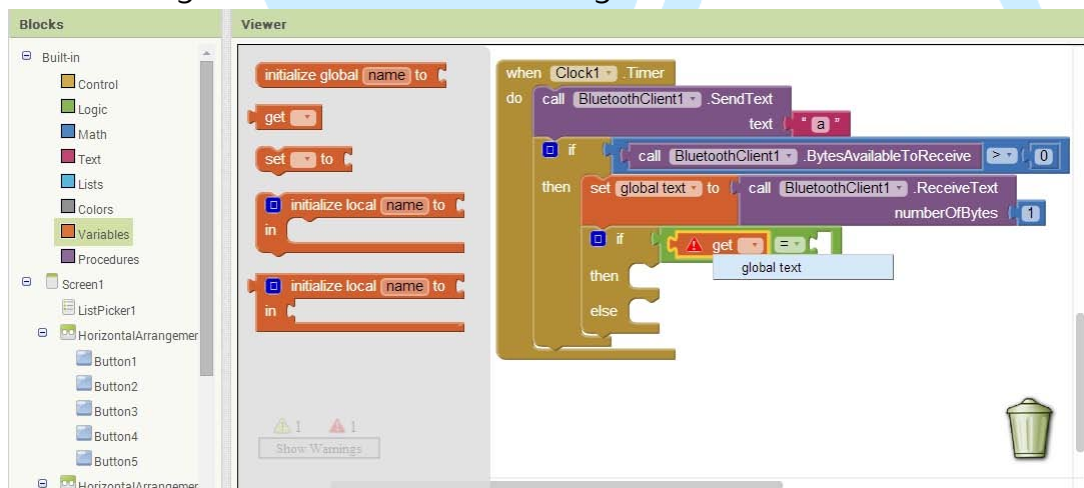
接下来从 Control 里拖出一个 if 判断式接在 set global text to 下面，如下图，这次的判断式比较特别我们要再加入例外的动作，点击判断式左上角蓝色四方形，把左侧的 else 拉进去到 if 里面，下面就会多出一个 else 了。



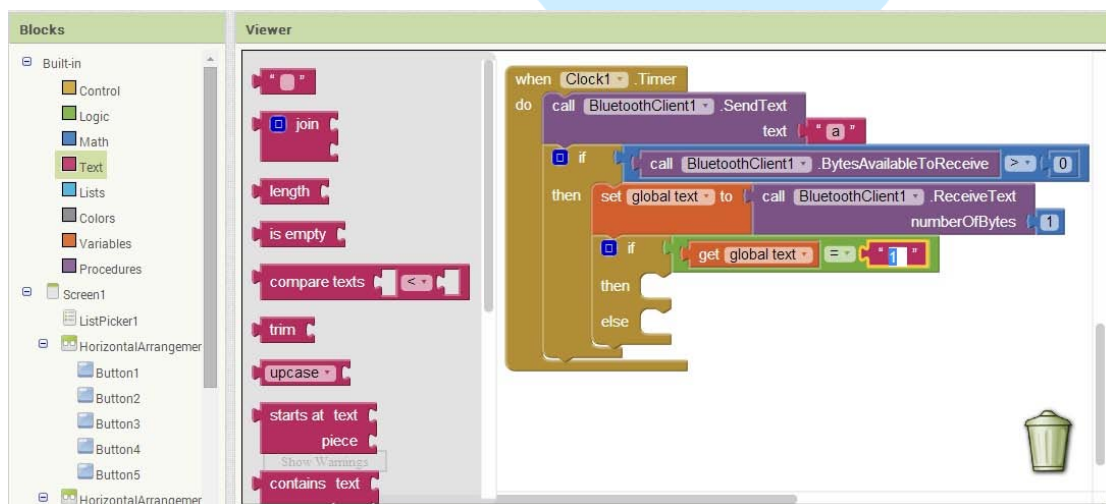
从 Logic 拖曳一个比较式出来，接在 if 的判断中，如下图。



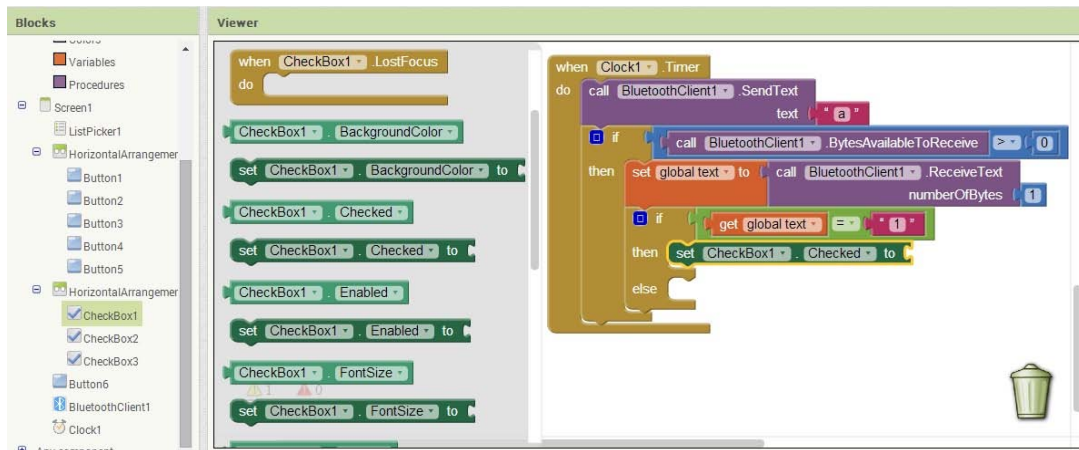
接着从左侧 Variables 中拉出 get 出来接在比较式左边空格中，如下图，并点击 get 空白处，选择刚刚定义的 global text 字符串变量。



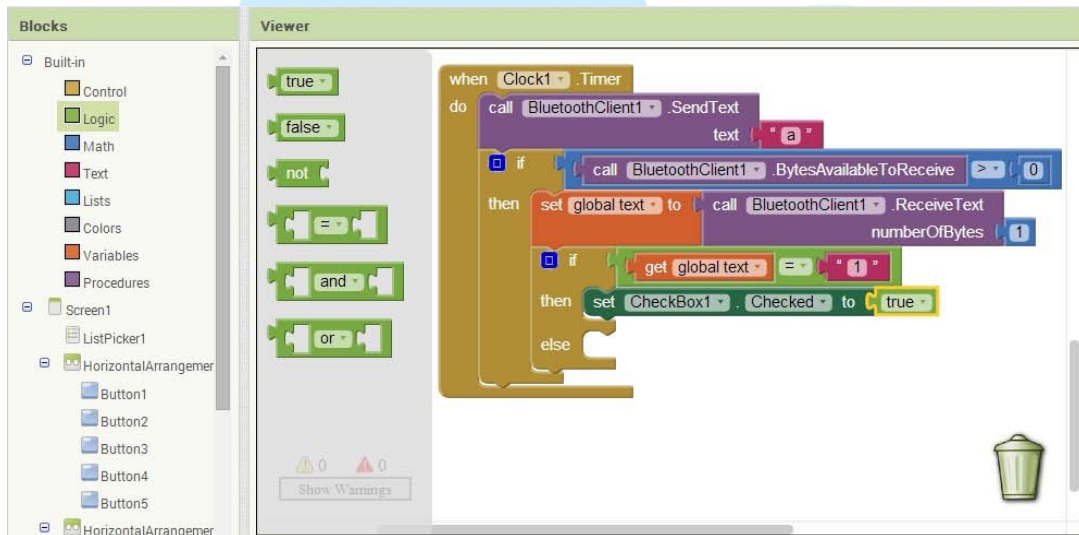
我们从左侧 Text 中拉出一个空白 text 出来接在比较式右边空白处，如下图，并在空白中填入 1。



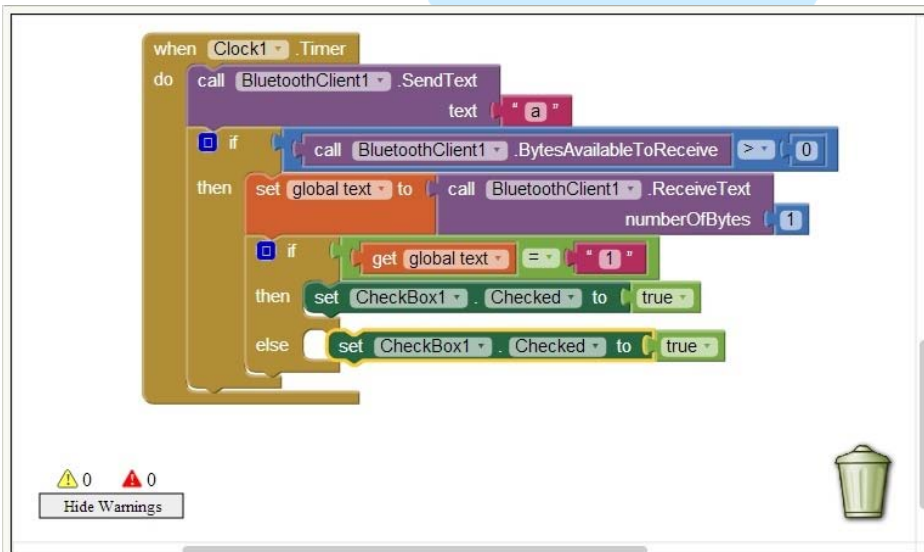
从左侧 CheckBox1 中拉出.Checked 出来接到 if 判断式内，如下图。



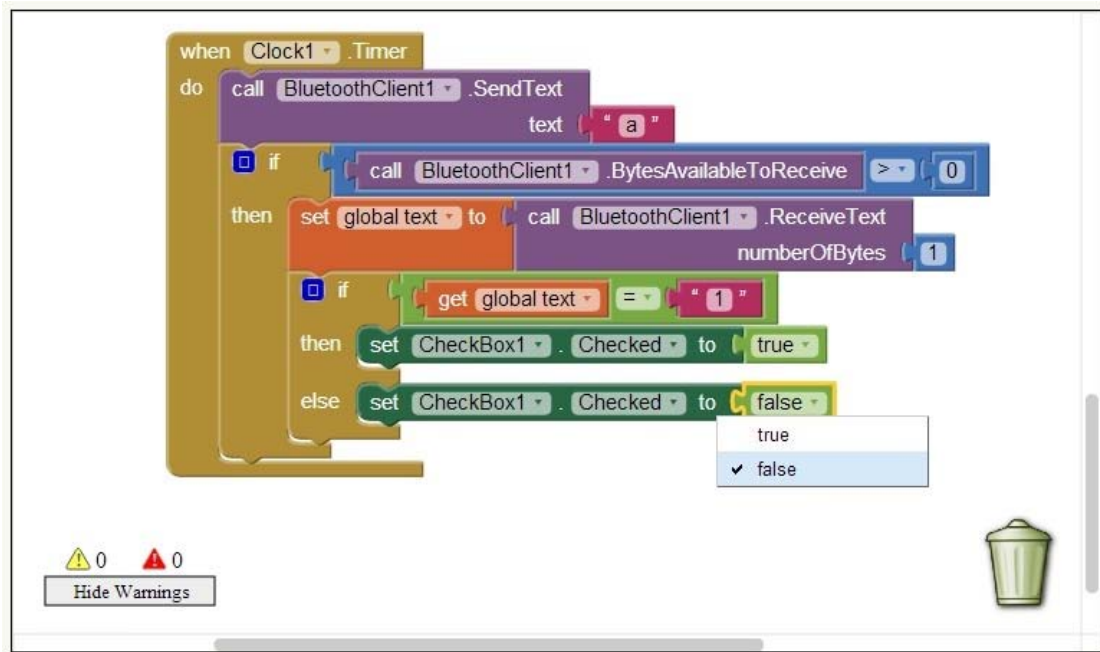
从 Logic 拖曳 true 到右边接合，如下图。



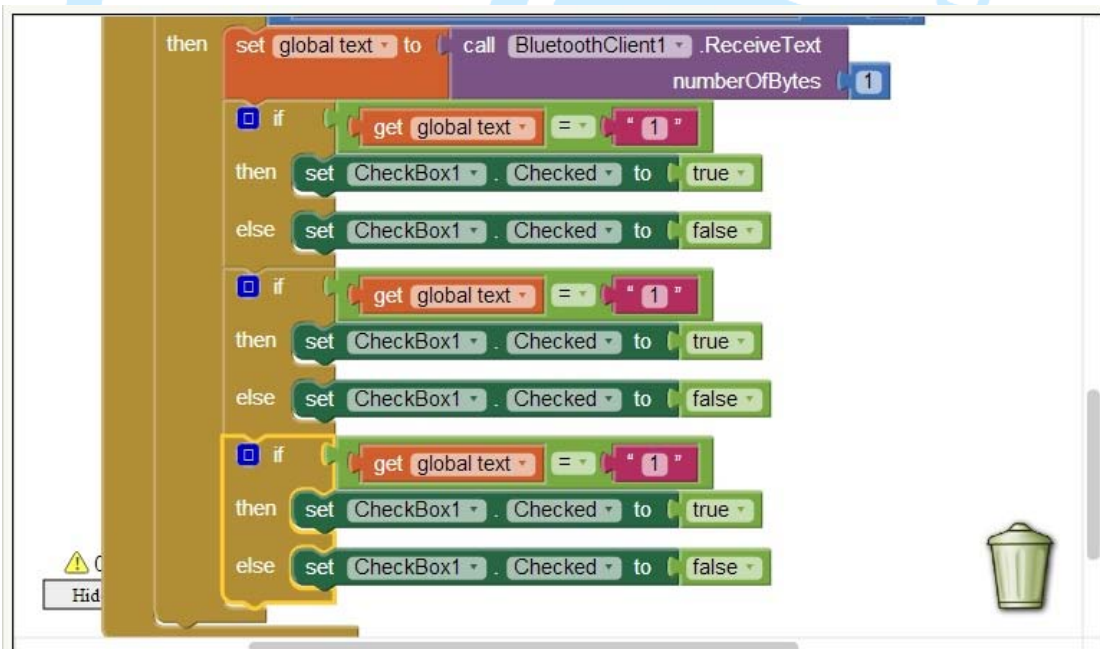
撰写程序就是要善用复制(Ctrl + C)与贴上(Ctrl + V)，我们点击 CheckBox1.Checked 之后按 Ctrl + C 复制，然后再按 Ctrl + V 贴上，就会再多出一组一样的组合，如下图，我们把它拖曳到 else 字段中吧。



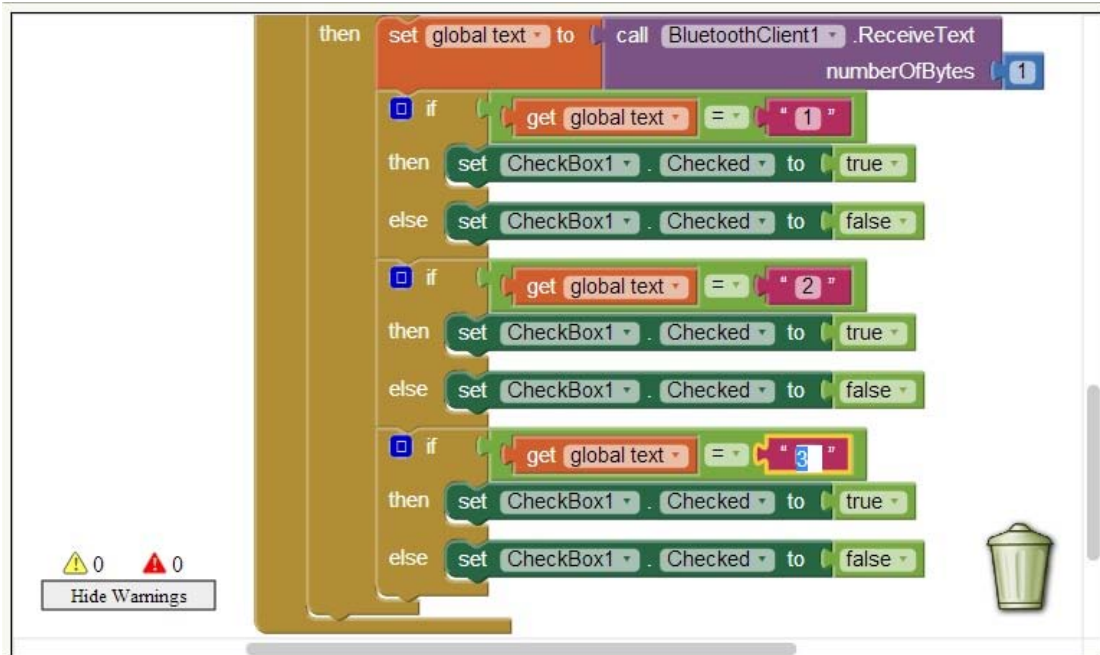
在 else 字段中，把后面的 True 改为 false，如下图，在 App Inventor 中，只要控件中有箭头都可以随时修改喔。



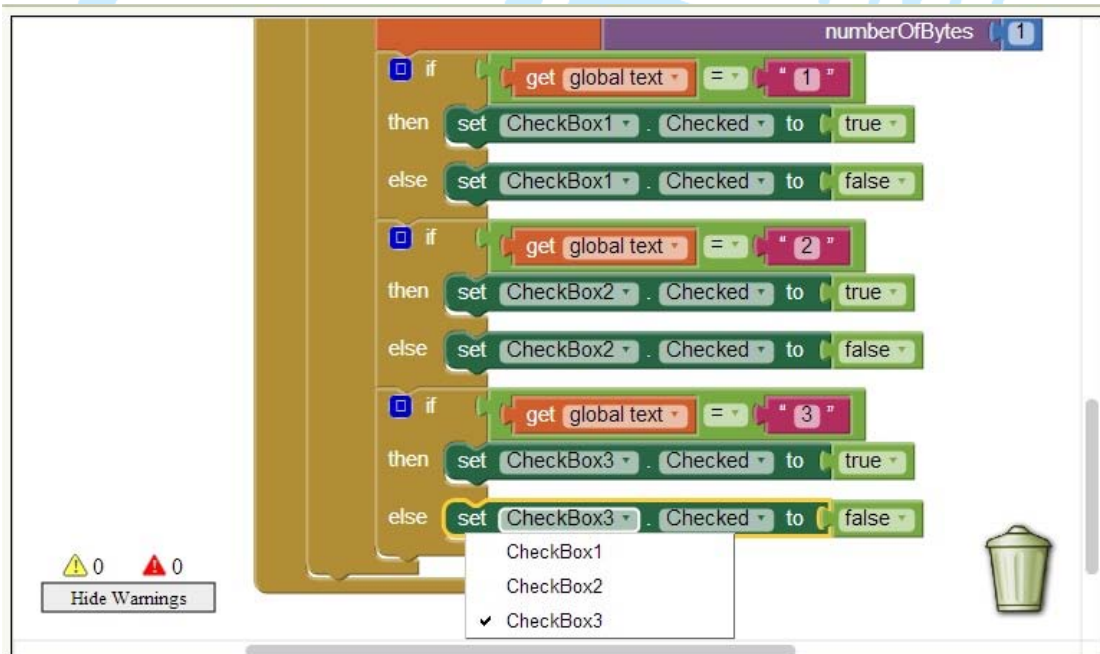
接下来就点击刚刚做的 if 判断式，复制(Ctrl + C)并贴上(Ctrl + V)，然后再移动到原本的 if 判断式下接合，之后再重复做一次，所以总共会有三个 if 判断式，如下图。



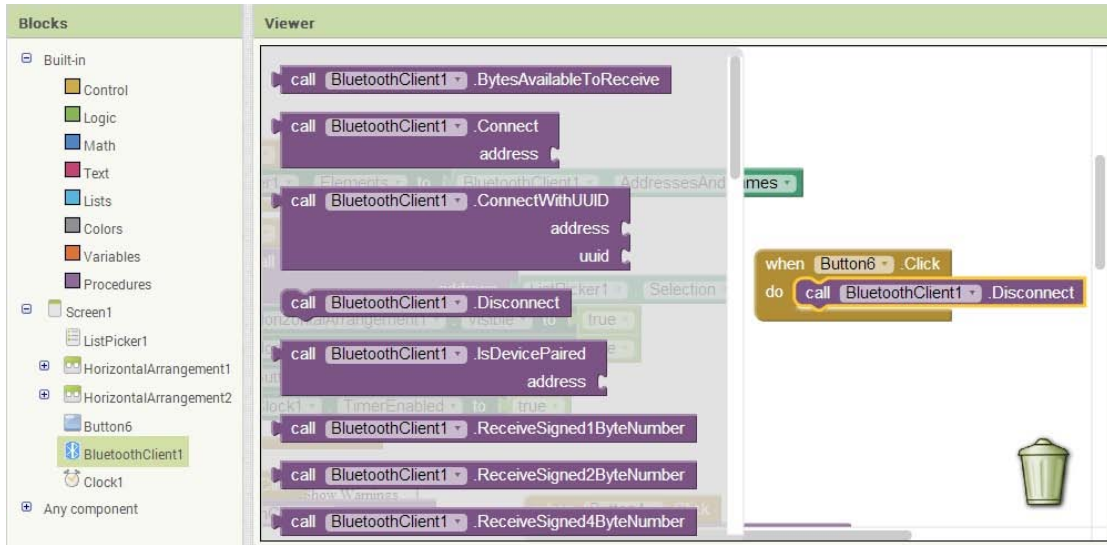
我们先来修改比较式中右边的数字，第一个不用改，第二和第三改为 2 和 3，如下图。



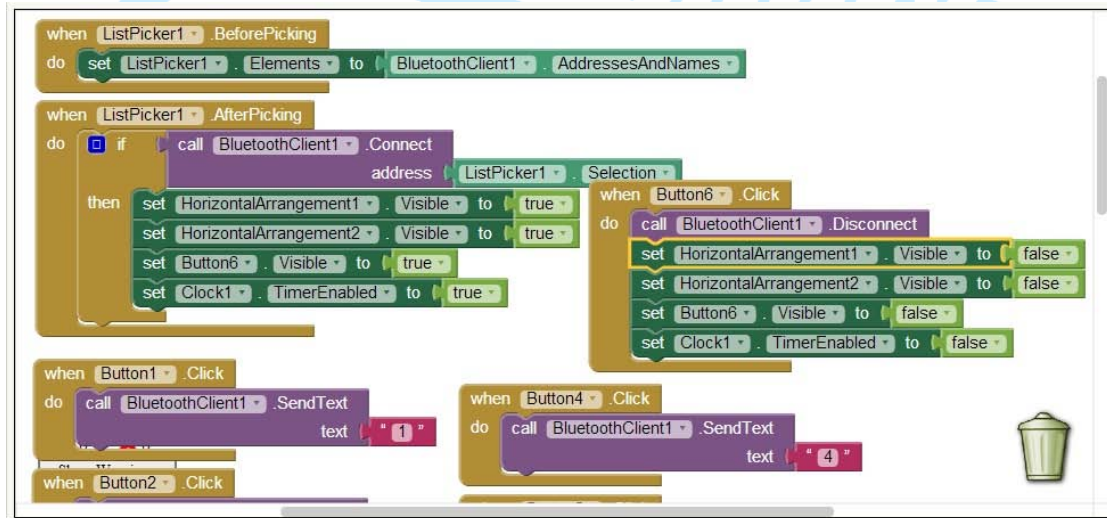
接着将第二和第三判断式内的 CheckBox1，改为 CheckBox2 及 CheckBox3，如下图。



接下来最后中断联机的按钮，也就是 Button6，拖曳.Click 事件出来，然后再从左侧 BluetoothClient1 中拖曳.Disconnect 出来接合，如下图。

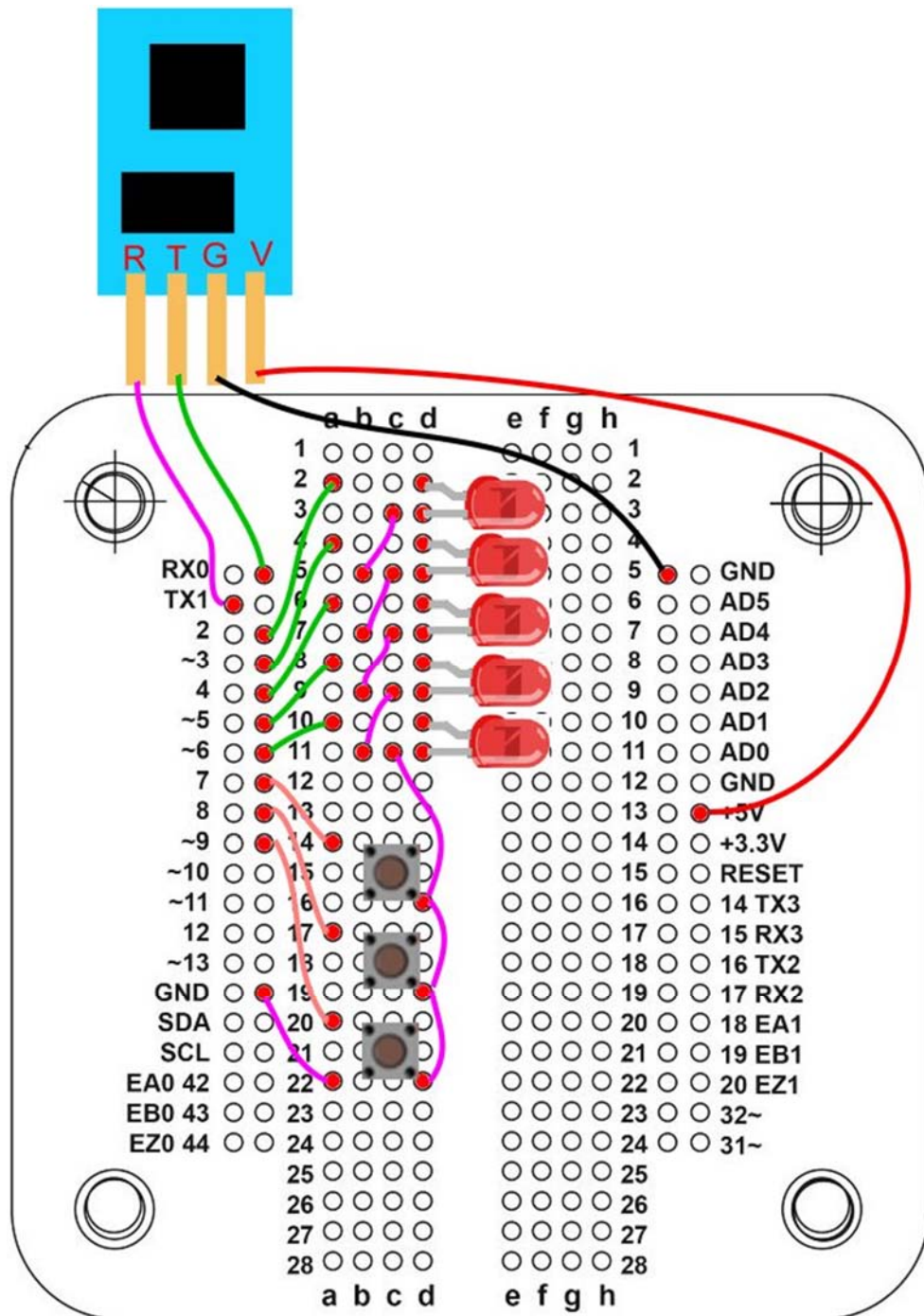


除了中断联机之外还要隐藏按钮和停止 Clock，就直接复制刚开始做的控件显示过去，再全部改成 false 就好了，如下图，做到这边就算完成了，接下来可以将程序传到 Android 装置中。



会不会觉得很奇怪，为什么要用 Clock 去接收处理 EduCake 所发送的讯息呢？因为我们需要实时的处理这些送过来的讯息，而必须隔一小段时间扫描一次看看有无讯息，而为什么要先送出"a"再做接收呢，因为 EduCake 的速度实在是太快了，一直传讯息会造成蓝牙反应不及，这时候我们就用 Arduino 牵制它，当接收到"a"的时候再送信息过去就好了，这样可以确保资料有确实的传递。

接下来我们将 EduCake 的 LED 从一个接到五个，再增加三个按钮，接脚图如下。



接着开启 IDE 接口，输入以下程序

```
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)
```

因为我们传输的是字符，我们要把它变成一半数字最简单的方法就是直接减去0的字符就好，至于减去1是因为数组刚刚定义5个实际是从0~4，所以要减去1，举个例子来说，当我们在Android装置中按下LED1，它就会传送一个"1"的字符回来，我们用if判断数组中的第0个位置自否为低电位，然后再做相应的动作。

接下来就是测试时间啦，联机之后，点击 LED1~5 会有相应的 LED 会亮起，而按下 EduCake 上的按钮会反应在按钮 1~3 上。

