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。

| | pp Inventor 2 Beta | Proje | ct + | Connect + | Build + | Help + |
|-------------|-----------------------|--------------|---------|------------|---------|--------|
| New Project | Delete Project | | | | | |
| Projects | | | | | | |
| Name | | Da | te Crea | ted | | |
| | Creat | e new App I | nvent | or project | | |
| | P | roject name: | test | | | |
| | | Cancel | | OK | | |

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,

86DUIND www.86duino.com

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。

| м. | Screen | Add Screen Remove Screen | | Designer Bio |
|-----------------|--------|-------------------------------------|---------------|----------------------|
| ilette | Viewe | | Components | Properties |
| ser Interface | | Display hidden components in Viewer | Screent | Screen1 |
| Button | ۲ | ₹.1 🖹 9:4 | 8 | AboutScreen |
| CheckBox | 0 | Screen1 | | |
| DatePicker | 0 | | | AlignHorizontal |
| Image | 0 | | | Left • |
| Label | 0 | | | AlignVertical |
| ListPicker | 0 | | | Тор • |
| ListView | 0 | | | BackgroundColor |
| Notifier | 0 | | | White |
| PasswordTextBox | () | | | Backgroundimage |
| Slider | ۲ | | | NOT NO. |
| Spinner | 0 | | | CloseScreenAnimation |
| TextBox | 0 | | | |
| TimePicker | | | | None |
| WebViewer | 0 | | | OpenScreenAnimation |
| | | | | Default |
| ayout | | | Rename Delete | Screen Drientation |

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。

| Palette | Viewer | Components |
|---------------------|-------------------------------------|---------------|
| User Interface | Display hidden components in Viewer | 😑 🛑 Screen 1 |
| 📄 Button 📀 | 🛜 ் வி 🖻 9:48 | Button1 |
| CheckBox 🕐 | Screen1 | |
| DatePicker 🕜 | Text for Button 1 | |
| 🎽 Image 💿 | | |
| \Lambda Label 📀 | | |
| 📃 ListPicker 🕐 | | |
| ListView 🕐 | | |
| 🛕 Notifier 🕐 | | |
| PasswordTextBox (2) | | |
| 🛄 Slider 🕐 | | |
| 👕 Spinner 🕐 | | |
| TextBox 🕐 | | |
| TimePicker 📀 | | |
| WebViewer 💿 | | |
| | | |
| Layout | | Rename Delete |
| Media | | |

86DUIND www.86duino.com

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。

| Media | 請點我 | |
|-------------|---------------------------|----------------|
| Upload File | TextAlignment center 🔻 | |
| | TextColor Default | CHILD PLANTING |
| | Visible showing • | A DISTRICT |
| | Width | |
| | Automatic | |
| | Height | |
| | Automatic | |

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。

| Automa | tic nt |
|--------|-----------|
| D | pixels |
| Cancel | OK |

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

| Automa | tic |
|------------------------|--------|
| Fill pare | nt |
| 50 | pixels |
| Cancel | OK |

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。

| Orregal | | |
|---------|-----|--|
| Screen | | |
| | 請點我 | |

接著再把 Label 拖曳到預覽畫面中,如下圖。

| Palette | | Viewer |
|-----------------|---|-------------------------------------|
| User Interface | | Display hidden components in Viewer |
| Button | ٢ | جَرا ا 9:48 |
| CheckBox | 0 | Screen1 |
| DatePicker | ۲ | 清型印 |
| 🌌 Image | • | |
| A Label | 0 | Text for Label1 |
| ListPicker | ٢ | |
| ListView | ۲ | |
| A Notifier | 0 | |
| PasswordTextBox | 3 | |
| Slider Slider | ۲ | |
| Spinner Spinner | ۲ | |
| I TextBox | ۲ | |
| i TimePicker | ۲ | |
| WebViewer | 1 | |
| Layout | | |
| Media | | |

86DUIND www.86duino.com

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 !

| | Designer Blocks |
|--|-----------------|
| Components | Properties |
| Screen1 Button1 Label1 | Label1 |
| | BackgroundColor |
| | FontBold |
| | FontItalic |
| | FontSize |
| | 20 |
| | FontTypeface |
| | default 🔻 |

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。

| MIT App Inventor 2 Beta | Project Connect Build Help My Projects Guide Repo | ort an Issue |
|--|---|-----------------|
| test | Screen1 • Add Screen Remove Screen | Designer Blocks |
| Blocks | Viewer | |
| Built-in Control Logic Math Text Lists Colors Variables Procedures Screen1 | | |
| Button1 A Label1 | 1 0 🛕 0 Show Warnings | |

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





86DUIND www.86duino.com

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。





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。

| Blocks | Viewer | |
|---|---|---|
| Built-in Control Logic Math Text Lists | when Button1 .Click do when Button1 .GotFocus do | |
| Colors Colors Variables Procedures Screen1 Button1 | when Button1 .LongClick do when Button1 .LostFocus | |
| Any component | do when Button1 . TouchDown do 0 0 0 | 1 |

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。

86DUIND www.86duino.com Blocks Viewer Built-in Label1 . BackgroundColor Control when Button1 .Click Logic set [Label1]. Text] to 🕻 set Label1 . BackgroundColor . to Math Text Label1 . FontSize Lists set Label1 . FontSize to Colors Variables Label1 . Height Procedures 😑 📃 Screen1 set Label1 . Height . to Button1 A Label1 Label1 • . Text • Any component set Label1 . Text . to Label1 . TextColor

接著點擊左側的 Text,把第一個積木拖到右邊拚進去。

set Label1 . TextColor to

| Blocks | Viewer | |
|---|--|--|
| Built-in Control Logic Math Text Lists Colors Variables Procedures | when Button1 · Click do set Label1 · Text · to C * * * | |
| Creen1 C | compare texts the Compare text | |
| | starts at text piece Show warnings | |



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。

| Project + | Connect + | Build + | Help + |
|------------|----------------------------------|--|---|
| Screen1 • | Add Screen | App (pr | ovide QR code for .apk) we .apk to my computer) |
| Viewer | | | |
| when do | Button1 set Label1 | .Click | t to ("你好 " |
| | Project + Screen1 + Viewer | Project - Connect - Screen1 - Add Screen Viewer when Button1 - do set Label1 | Project • Connect • Build • Screen1 • Add Screen App (pr App (sa Viewer when Button1 • Click do set Label1 • Tex |

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。

| 〈 🔯 安全性 | |
|--|--|
| 裝置管理 | |
| 裝 置管理員 查看或撤銷裝置管理員 | |
| 不明的來源 允許安裝來源不明的應用程式 | |
| 驗證應用程式 不允許安裝可能造成系統損壞的應 用程式,或在安裝前提出警告 | |
| 憑證儲存空間 | |
| 儲存空間類型 ^{僅限軟體} | |
| 信任的憑證 | |

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。

| MIT App Invento | or 2 Beta | Project + | Connect + | Build + Help + |
|-----------------|--------------|-----------|------------|-------------------------------------|
| test | | Screen1 + | Add Screen | App (provide QR code for .apk) |
| Palette | | Viewer | | App (save .apk to my computer) |
| User Interface | | | | Display hidden components in Viewer |
| Button | 1 | | | |
| CheckBox | (?) | | | Screen1 |
| DatePicker | (?) | | | 請點我 |
| 🧧 🌌 Image | 0 | | | |
| A Label | 0 | | | Text for Label1 |

86Duind

www.86duino.com

接著會跑出進度條,正在編譯 APP 中,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。



您要安裝這個應用程式嗎?應用程式將取 得以下權限:



86DUIND www.86duino.com

如果選擇方案二·點擊 Build-> App(save .apk to my computer)·

如下圖。

| MIT App Inventor | 2 eta | Project • | Connect + | Build + | Help + |
|------------------|---------------------|-----------|------------|----------|-----------------------------|
| test | AN AMERICAN CONTROL | Screen1 • | Add Screen | App (pr | ovide QR code for .apk) |
| Palette | | Viewer | | App (sa | ave .apk to my computer) |
| User Interface | | | | Display | hidden components in Viewer |
| Button | 0 | | | | |
| CheckBox | ? | | | 我的第· | 一個程式 |
| DatePicker | 0 | | | | 請點我 |
| 🏭 Image | • | | | _ | |
| A Label | 1 | | | | |

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。

| | | | 另存 | 新檔 | | | × |
|--------------------------|------------|-----------|------------------|---------|--------------|----------------|-------|
| ⊛ 🤄 ד ↑ 🚺 | , 本 | 機 ▶ 文件 | ▶ App Inventor 2 | ~ d | , 搜尋, | App Inventor 2 | Q |
| 組合管理 ▼ 新 | 増資料 | 夾 | | | | | ≣ ▼ 🔞 |
| ☆ 我的最愛 ↓ 下載 ■ 桌面 ※ 最近的位置 | ^ | 名稱 | * | 沒有符合搜尋條 | 修改日期 件的項目 | 類型 | |
| 🜏 家用群組 | | | | | | | |
| □● 本機 | | | | | | | |
| ● 文件 | | | | | | | |
| | | | | | | | |
| ■ 桌面 | ~ | < | | | | | > |
| 檔案名稱(N): | test.a | pk | | | | | ~ |
| 存檔類型(T): | APK 1 | 昌宾 (.apk) | | | | | ~ |
| ▲ 陽藏資料夾 | | | | | 4 | 字檔(S) | 取消 |

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?



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

| est | | My Projects |
|----------------|--|--|
| | | Start new project |
| Palette | | Import project (.aia) from my computer |
| User Interface | | Import project (.aia) from a repository |
| Button | 0 | Delete project |
| CheckBox | 0 | Save project |
| DatePicker | 0 | Save project as |
| | | <u> </u> |
| 專案名稱就 C | 尤取 "Bluetoot reate new App Project name: | th_Connect"吧。 Inventor project Bluetooth_Connect |
| | | |

接著從左側拖曳一個 ListPicker 元件到預覽畫面中。

| Palette | | Viewer |
|-----------------|---|-------------------------------------|
| User Interface | | Display hidden components in Viewer |
| Button | 0 | ╤₄1 12 9:48 |
| CheckBox | 0 | Screen1 |
| DatePicker | • | Text for ListPicker1 |
| 🌌 Image | 0 | |
| A Label | 0 | |
| ListPicker | 0 | |
| ListView | 0 | |
| 🗥 Notifier | 0 | |
| PasswordTextBox | 0 | |
| Slider Slider | 0 | |
| Spinner Spinner | 0 | |
| TextBox | 0 | |

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 •

| Bluetooth_Connect |
|-------------------|
| 選項一 |
| 選項二 |
| 選項三 |
| 選項四 |

選單設定的方法有兩種,一種是在 Properties(屬性)欄位中的 ElementsFromString 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。

| | Properties |
|------------------------|---|
| | ListPicker1 |
| | BackgroundColor Default ElementsFromString 選項一,選項二,選項三,選 |
| 方法二·如下圖 |]·可用在程式中指定選項·彈性較高。 |
| set ListPicker1 . Elen | nentsFromString To 增置項一,選項二,選項三,選項四世 |
| 方法二的另一種 | 表示法·一切以寫程式方便就好。 |
| set ListPicker1 | . Elements to I make a list (選項一 " "選項三 " "選項三 " "選項回 " |

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。

| lewer | | Components | Properties |
|-------|--|--|--|
| | ■Display hidden components in Viewer Screen1 送择整艺学装置 | Screen1 ListPicker1 BluetoothClient1 | BluetoothClient 1 CharacterEncoding UTF-8 DelimiterByte 0 HighByteFirst Secure |
| | Non-visible components | Rename Delete | |

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 ·

| Palette | Viewer |
|-----------------------|-------------------------------------|
| User Interface | Display hidden components in Viewer |
| 📄 Button 💿 | ିଙ୍⊶ା 🖻 9:48 |
| CheckBox 📀 | Screen1 |
| DatePicker 🤊 | 選擇離芽裝置 |
| 🎽 Image 📀 | |
| A Label 🧿 | 請點我 |
| ListPicker 📀 | |
| ListView 🤊 | |
| 🛕 Notifier 🧿 | |
| PasswordTextBox 🤊 | |
| Slider 🧿 | |
| 🗑 Spinner 🤊 | |
| I TextBox 🧿 | |
| TimePicker 📀 | |
| WebViewer 🕥 | |
| Layout | |
| Media | Non-visible components |
| Drawing and Animation | 88 Distribution |
| Concore | BluetoothClient1 |

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。

| 請點我 | |
|---------------|--|
| | |
| TextAlignment | |
| center 🔻 | |
| | |
| TextColor | |
| Default | |
| Visible | |
| hidden 🔻 | |
| Width | |
| Fill parent | |
| Height | |
| neight | |
| 50 pixels | |

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。

| Blocks | Viewer |
|--|--|
| Built-in Control Logic Math Text Lists Colors Variables | when ListPicker1 AfterPicking do when ListPicker1 BeforePicking do when ListPicker1 GotFocus |
| Procedures Screen1 ListPicker1 Button1 BluetoothClient1 | do when ListPicker1 .LostFocus do |
| Any component | when ListPicker1 .TouchDown do A O when ListPicker1 .TouchUp |

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,功能是將 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。

| Blocks | Viewer |
|--|--|
| Built-in Control Logic Auth Text Lists Colors Variables Procedures Securit ListPictur1 | List List call Simetooth/Client1 call Simetooth/Client1 Bluetooth/Client1 AddressesAndNames b Bluetooth/Client1 call CharacterEncoding cet Bluetooth/Client1 cet CharacterEncoding cet Bluetooth/Client1 |
| BluetoothClient1 Any component | BluetoothClient I DelimiterByte D set (BluetoothClient I DelimiterByte D to M BluetoothClient I DelimiterByte D to M |

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。

86Duind

www.86duino.com

| Blocks | Viewer |
|--|---|
| Built-in Control Logic Math | when ListPicker1 AfterPicking do set ListPicker1 ListPicker1 BeforePicking do set ListPicker1 ListPicker1 BeforePicking |
| Text Lists Colors Variables | when ListPicker1 GotFocus |
| Screen1 ListPicker1 BluetoothClient1 Any component | when ListPicker1 LostFocus do |
| | when ListPicker1 TouchDown do |

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 ·

| Blocks | Viewer | |
|---|---|--|
| Builtin Control Logic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 Button1 BluetoothClient1 Any component | call BluetoothClient1 • BytesAvailableToReceive call BluetoothClient1 • Connect address call BluetoothClient1 • ConnectWithUUID address call BluetoothClient1 • ConnectWithUUID address call BluetoothClient1 • Disconnect call BluetoothClient1 • JsDevicePaired address call BluetoothClient1 • ReceiveSigned1ByteNumber 20 40 call BluetoothClient1 • ReceiveSigned2ByteNumber | when ListPicker1 BeforePicking do set ListPicker1 Elements to BluetoothClient1 when ListPicker1 AtterPicking do if call BluetoothClient1 Connect address then |
| | | |

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 ·

| Blocks | Viewer |
|--|--|
| Built-in Control Logic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 Button1 BluetoothClient1 Any component | <pre>when ListPicker1 @ BeforePicking do Set ListPicker1 @ BeforePicking do Set ListPicker1 @ BluetoothClient1 @ AddressesAndNames @ when ListPicker1 @ Elements @ to [BluetoothClient1 @ AddressesAndNames @ when ListPicker1 @ Selection set ListPicker1 @ SelectionIndex @ set ListPicker1 @ ShowFeedback @ then ListPicker1 @ ShowFeedback @ to [ListPicker1 @ ShowFietBar @ [ListPicker1 @ [L</pre> |
| | eet fiidPickerf v. ShowFilterBor 1 to |

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,我們就可以把按紐顯現出來了。

| Blocks | Viewer |
|--|--|
| Builtin Control Logic Math Text Lists Colors Variables Procedures Procedures ListPicker1 ListPicker1 Button1 | set Button1 · Extrements · to · Button1 · Text · to · Button1 · Text Color · Clements · to · Button1 · TextColor · Clements · to · Button1 · TextColor · Clements · Connect · Clements · Clemen |
| BluetoothClient1 Any component | Button1 Width to C Set Button1 . Width to C Button1 . Button1 . |

接著點擊 Logic,拖曳 true 到右側接合,如下圖。

| Blocks | Viewer |
|--|---|
| Builtin Control Logic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 Button1 BluetoothClient1 | <pre>when ListPicker1 BeforePicking do set ListPicker1 C. Elements to [BluetoothClient1 C. AddressesAndNames When ListPicker1 AfterPicking do if call BluetoothClient1 Connect address [ListPicker1 C. Selection then set Button1 C. Visible to C true C. Selection C.</pre> |
| Any component | A 0 A 0 Show Warnings |

86DUIND www.86duino.com

側點擊 Button1,將.Click 拖曳到右側,如下圖。

| Blocks | Viewer |
|---|---|
| Built-in Control Control Cogic Math Text Lists Colors Variables Procedures Coreen1 ListPicker1 Button1 ButtontClinet1 | when Button1 Click when ListPicker1 BeforePicking do set ListPicker1 Elements to BluetoothClient1 AddressesAndNames when Button1 GotFocus when ListPicker1 AtterPicking do if call BluetoothClient1 Connect address I ListPicker1 Steetion address I ListPicker1 Steetion when Button1 LongClick when Button1 LongClick when Button1 Click do when Button1 Click when Button1 Click |
| Any component | when Button1 TouchDown do 2 4 0 when Button1 TouchUp |

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

| Blocks | Viewer |
|--|---|
| Built in Control Logic Math Text Lists Cotors Variables Scrent Districtert Builtont Builtont Builtont Any component | cal SitetoothClentt: SendByteNumber number cal SitetoothClentt: SendByteNumber number cal SitetoothClentt: SendByteNumber number cal SitetoothClentt: SendBytes Est Set SitetoothClentt: SendBytes Est BitetoothClentt: SendBytes Est BitetoothClentt: AddressesAndNames BitetoothClentt: AddressesAndNames Cal SitetoothClentt: SendBytes Est BitetoothClentt: AddressesAndNames BitetoothClentt: AddressesAndNames |

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。

| Blocks | Viewer |
|--|--|
| Butt in Control Logic Logic Logic Logic Usts Colors Variables Procedures Screen1 LusPicken1 Button1 Button1Client1 Any component | <pre>viven [ListPicker] .BeforePicking do set [ListPicker] .Elements to (Elements t</pre> |

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。

86DUIND www.86duino.com

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



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 :回應靭體的版本
- ▶ 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。

| \$ | COM5 | - 🗆 🗙 |
|--------------------------|-------------|-------------------|
| AT | | Send |
| Enter AT commands: OK | | |
| 🗹 Autoscroll | No line end | ing 🗸 9600 baud 🗸 |

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:

86DUIND www.86duino.com



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。

| | く藍牙 開啟 | |
|------|---|---|
| | Android 装置 只向配對的裝置顯示 | |
| | 配對裝置 | |
| | HC-06 | |
| | | |
| | 搜尋裝置 | |
| 接 | 妾著開啟程式·如下圖·點擊"選擇藍芽裝置"。 | |
| Scre | een1 | |
| | 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 | 0 |
| | Bluetooth_Connect | |
| | 00:15:FF:F4:0B:72 JY-MCU | |
| | 98:D3:31:40:03:2F HC-06 | |
| | | |

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。



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。

| Viewer | Display hidden co | Deleting this component will dele with it in the Blocks Editor. Are yo delete? | ete all blocks associa u sure you want to | ted po | screen1 | |
|--------|-------------------|--|--|--------|-----------------------------|---|
| | Screen1 | | 確定 取消 | | Button1 BluetoothClient1 | |
| | 暹 | 醫羅藍芽裝置 | | | | 1 |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

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 顆按紐,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,是個方便管理介 面的好工具。

| alette | Viewer |
|--------------------------|-------------------------------------|
| User Interface | Display hidden components in Viewer |
| Layout | 🦻 🤹 🧐 🧐 🧐 😒 9:48 |
| HorizontalArrangement (? | Screen1 |
| TableArrangement 📀 | 選擇藍芽裝管 |
| VerticalArrangement | |
| Media | |
| Drawing and Animation | |
| Sensors | |
| Social | |
| Storage | |
| Connectivity | |
| LEGO® MINDSTORMS® | |
| | |
| | |
| | |
| | |

在 Horizontal Arrangement 的屬性中·將 Width(寬)改為 Fill parent· 如下圖。

| | Components | Properties |
|-------------|---|---|
| | Screen1 ListPicker1 HorizontalArrangement1 BluetoothClient1 | HorizontalArrangement1 AlignHorizontal Left AlignVertical Top Visible showing Width Fill parent Height Automatic |
| 把 Height | Button 拖曳到剛剛的框框内 t(高)改成 50 pixels · 如下圖 iewer Display hidden components in Viewer Screen1 選擇藍芽裝置 | S · 在屬性中 · 將 Text 改為 LED1 。 S · 在屬性中 · 將 Text 改為 LED1 S · · · · · · · · · · · · · · · · · · |

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。

| | | II. | P.al 📓 9:4 | 8 |
|---------|----------|------|------------|---|
| Screen1 | | | | |
| 選擇藍芽裝置 | | | | |
| LED1 LI | ED2 LED3 | LED4 | LED5 | |

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。

| Palette | Viewer | Components | Properties |
|---|--|---|---|
| User Interface Layout HorizontalArrangement | Display hidden components in Viewer Screen 1 | 9:48 □ Screen1 □ ListPicker1 □ MorizontalArrangement1 | HorizontalArrangement2 AlignHorizontal Left V |
| TableArrangement VerticalArrangement | ① 選擇藍芽裝置 ① | Button1 | AlignVertical Top |
| Media Drawing and Animation Sensors Social Storage | | Button4 | showing Width Fill parent Height Automatic |
| LEGO® MINDSTORMS® | | Rename Delete | |

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。

| Palette | Viewer | Components | Properties |
|--|---|---|---|
| User Interface | Display hidden components in Viewer | B Screen1 | CheckBox1 |
| Button CheckBox DatePicker Image | ◎ Screen1 ◎ 選擇藍芽裝置 ◎ | Image: System 2 Image: System 2 Image: System 2 Image: System 2 | BackgroundColor None Checked |
| ▲ Label ➡ ListPicker ➡ ListView ▲ Notifier ➡ PasswordTextBox ➡ Slider ➡ Spinner ➡ TextBox ➡ TimePicker ➡ WebViewer | 回 LED1 LED2 LED3 LE ① | ED4 LED5 Button4 Button5 Button5 Button5 CheckBox1 BluetoothClient1 | Enabled FontBold FontItalic FontSize 14.0 FontTypeface default Text 打按證 1 |
| Layout | | Rename Delete | TextColor Black |

繼續剛才的動作再拖曳兩個 CheckBox · 屬性中的 Text 分別改為按紐 2 及按紐 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 。

| DatePicker (7) | | | 選 | 擇藍芽裝 | 置 | | But | tton2 | | |
|----------------|---------------------|---|---------|-------|--------------|----------|----------|------------|---------------------------|--------------------|
| A | Label | 3 | LED1 | LED2 | LED3 | LED4 | LED5 | But | iton3 iton4 | Enabled |
| | ListPicker | 0 | teren 1 | | \$100 C | teen | | But | ton5 ontalArrangement2 | FontBold |
| | Notifier | 0 | 1×311 (| | <u>8802</u> | 1880 | <u> </u> | Che Che | eckBox1 | Fontitalic |
| | PasswordTextBox | 0 | | | | | | Che | eckBox3 | FontSize |
| | Slider | ۲ | | | | | | 🚯 Blueto | othClient1 | 14.0 |
| 8 | Spinner | 7 | | | | | | | | FontTypeface |
| 1 | TextBox | 0 | | | | | | | | default 🔻 |
| 810 | TimePicker | 0 | | | | | | | | Text |
| ۲ | WebViewer | 0 | | | | | | | | 按鈕3 |
| La | yout | | 3.4 | | | | | • Rename | ► Delete | TextColor Black |
| M | edia | | | Non-v | isible con | nponents | | | | Visible |
| Dr | awing and Animation | | | E | BluetoothCli | ent1 | | Media | | snowing • |
| Sensors | | | | | Upload File | | Width | | | |
| Social | | | | | | | | | Fill parent | |
| St | orage | | | | | | | | | o pixels |
| Co | onnectivity | | | | | | | | | Cancel OK |

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。

| | | | II. | P.al 📓 9:4 | | | |
|----------------|------|------|------|------------|--|--|--|
| Screen1 | | | | | | | |
| 選擇藍芽裝置 | | | | | | | |
| LED1 | LED2 | LED3 | LED4 | LED5 | | | |
| □按鈕1 □按鈕2 □按鈕3 | | | | | | | |
| | | 中斷連縛 | \$ | | | | |
| | | | | | | | |

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。

| Components | Properties |
|---|---|
| 🖯 🗖 Screen1 | HorizontalArrangement1 |
| EListPicker1 | AlignHorizontal |
| E HorizontalArrangement1 | Left |
| Button1 Button2 Button3 Button4 | AlignVertical Top Visible showing |
| Buttons HorizontalArrangement2 CheckBox1 CheckBox2 CheckBox3 Button6 | showing hidden Fill parent Height Automatic |

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。

| Palette | Viewer | |
|-----------------------|--------|-------------------------------------|
| User Interface | | Display hidden components in Viewer |
| Layout | | ╤₁∥ 12 9:48 |
| Media | | Screen1 |
| Drawing and Animation | | 選擇藍芽裝置 |
| Sensors | | |
| AccelerometerSensor | 0 | |
| BarcodeScanner | 0 | |
| 🔯 Clock | 0 | |
| LocationSensor | 0 | |
| DearField | 0 | |
| OrientationSensor | 0 | |
| Social | | |
| Storage | | |
| Connectivity | | |
| LEGO® MINDSTORMS® | » | |
| | | |
| | | Non-visible components |

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。

| Properties | |
|------------------|--|
| Clock1 | |
| TimerAlwaysFires | |
| TimerEnabled | |
| TimerInterval | |
| 10 | |

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。

| Blocks | Viewer |
|---|---|
| Builtin Control Cogic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 ListPicker1 HorizontalArrangemer Builton1 | <pre>when ListPicker1 BeforePicking do set ListPicker1 Elements to BluetoothClient1 AddressesAndNames when ListPicker1 AfterPicking do f call BluetoothClient1 Connect address (ListPicker1 Selection then set HorizontalArrangement1 Visible to true set Button6 Visible to true set Clock1 TimerEnabled to true</pre> |
| Button2 Button3 Button4 Button5 | A 0 A 0 Show Warnings |

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 ·



86DUIND www.86duino.com

| 從左側 Tex | t 中拖曳空白的 text ¾ | 過去接合,並在裡面輸入1.如下圖 |
|---|--|---|
| Blocks | Viewer | |
| Built-in Control Logic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 Screen1 UstPicker1 Button1 Button2 Button3 | <pre>i i i i i i i i i i i i i i i i i i i</pre> | do set ListPicker1 . Elements to BluetoothClier when ListPicker1 AfterPicking do if call BluetoothClient1 Connect address ListF then set HorizontalArrangement1 Visible to set HorizontalArrangement2 Visible to set Button6 Visible to frue v when Button1 Click do call BluetoothClient1 SendText text C 1 * |
| Button4 | piece | |

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。

| when Button1 . Click | |
|---------------------------------------|-------------------------------------|
| do call [BluetoothClient1] .SendText | |
| text (11 | when Button4 . Click |
| | do call BluetoothClient1 . SendText |
| when Button2 .Click | text (|
| do call BluetoothClient1 .SendText | |
| text (2 " | when Button5 . Click |
| | do call BluetoothClient1 .SendText |
| when Button3 . Click | text (|
| do call BluetoothClient1 .SendText | |
| text (3 1 | |

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。

| when Button2 .Click | |
|---|---|
| do call BluetoothClient1 .SendText text 1 2" | when Button5 .Click do call BluetoothClient1 .SendText text (|
| when Button3 . Click do call BluetoothClient1 . SendText text (3 " | |
| when Clock1 • .Timer do call BluetoothClient1 • .SendText | |
| ¹ 0 ▲ 0 Show Warnings | |

然後從左側的 Control 中拉出一個 if 判斷式接在下面,如下圖。



86Duind

www.86duino.com



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。

| Viewer | |
|--|--|
| call BluetoothClient1 BytesAvailableToReceive call BluetoothClient1 Connect address address address call BluetoothClient1 ConnectWithUUID address call BluetoothClient1 ConnectWithUUID address call BluetoothClient1 Disconnect | when Clock1 Timer do Call BluetoothGlient1 SendText text 1 a a Call BluetoothGlient1 BylesAvailableToReceive 2010 then |
| call (BluetoothClient1) ReceiveSigned1ByteNumber | |
| | Viewer |

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

86DUIND

www.86duino.com

| 便程式閱讀,如 | 口下圖。 | |
|--|---|--|
| Blocks | Viewer | |
| Builtin Control Logic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 White Markarangemer | initialize global name to get set to to in in in in in in in in in in | initialize global [ext] to when Clock1 Timer do Call EluctoothClient1 SendText text : a if Call EluctoothClient1 BytesAvailableToReceive |
| Button1 Button2 Button3 Button4 Button5 HorizontalArrangemer | ▲ 1 ▲ 0 ShowWarnings | |

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。

| Blocks | Viewer |
|--|---|
| Built-in Control Logic Math Text Lists Colors Variables Procedures Screen1 ListPricker1 ColorsArrangemer | <pre>initialize global lext to C * * * initialize global lext to C * * initialize global lext</pre> |
| Button2 Button3 Button4 Button5 | starts at text piece Show Warninga contains text |

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。 | | |
|---|--|---|
| Blocks | Viewer | |
| Variables Variables Procedures Screen1 ListPicker1 Button1 Button1 Button2 Button5 Button5 MorizontalArrangemer ✓ CheckBox1 | initialize global name to get set to in in in in the initialize local name to in in in the initialize local name to in in the initialize local name to in the initialize local name to in the initialize local name to in the initialize local name to in the initialize local name to in in the initialize local name to in in in the initialize local name to in in in in in initialize local name to in in in in in in in in in in | initialize global text to f * • • • • • • • • • • • • • • • • • • |
| CheckBox2 CheckBox3 Button6 BluetoothClient1 | A 0 A 1 Show Warnings | 1 |

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。

| Blocks | Viewer | |
|--|---|---|
| Built-in Control Logic Math Logic Math Lists Colors Variables Procedures ListPicker1 ListPicker1 MicroatalArrangemer Button1 | for each number from 1 1 to 1 5 by 1 do | when Clock1 Timer do call Fittemotificant1 Send Text else f felse f felse f felse f felse f felse f felse |
| Button2 Button3 Button4 Button5 | then the second | |

86Duind

www.86duino.com

| 從 Logi | c 拖曳一個比較式出來,接在 if 的判斷中,如下圖。 |
|---|---|
| Blocks | Viewer |
| Builtin Control Logic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 MetrizontalArrangemer Button1 | <pre>when Clock1 . Timer do call BluetoothClient1 . SendText text : a" if call BluetoothClient1 . BytesAvailableToReceive</pre> |
| Button2 Button3 Button4 Button5 HorizontalArrangemer | A 1 A 0 Show Warnings |

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。

| Blocks | Viewer |
|---|---|
| Builtin Control Control Cogic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 Screen1 Button1 Button1 Button1 Button2 Button3 | <pre>when Clock1 Timer do call EluetoothClient1 SendText text [*] * ength is empty compare texts [*] * tim tim starts at text </pre> |
| Button4 Button5 | piece Show Warnings Contains text |

86Duind

www.86duino.com

| 從左側 CheckBox1 中拉出.Checked 出來接到 if 判斷式內,如下圖。 | | | |
|---|---|--|--|
| Blocks | Viewer | | |
| Variables Variables Procedures Screen 1 UstPicker 1 Whorizontal Arrangemer Button 1 Button 2 Button 3 Button 3 Button 5 Sutton 5 | when CheckBox1 LostFocus do do call BluetoothClient1 SendText text (* 0 * text (* 0 * call BluetoothClient1 BluetoothClient1 Receive set CheckBox1 CheckBox | | |
| CheckBox1 CheckBox2 CheckBox3 Button6 BluetoothClient1 Clock1 | set CheckBox1 . Enabled to CheckBox1 . FontSize set CheckBox1 . FontSize to | | |

從 Logic 拖曳 true 到右邊接合,如下圖。

| Blocks | Viewer |
|--|--|
| Built-in Control Logic Math Text Lists Colors Variables Procedures Screen1 ListPicker1 MorizontalArrangement Button1 | <pre>when Clock1 Timer do call BluetoothClient1 BluetoothClient1 BluetoothClient1 BluetoothClient1 Receive Clock then Set global text to call BluetoothClient1 ReceiveText numberOfBytes 1 if get global text = if if then Set CheckBox1 Checked to C true is else</pre> |
| Button3 Button4 Button5 MorizontalArrangemer | ▲ 0 ▲ 0 Show Warnings |

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·第三和第三改為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.Dissconnect 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



86DUIND www.86duino.com



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。