Android 連接 Bluetooth + UART

1.
開發工具介紹

Java 與 Kotlin

- Java 是編程 JVM 平台的程式語言之一，一種廣泛使用的電腦程式設計語言，擁有跨平台、物件導向、泛型程式設計的特性，廣泛應用於企業級 Web 應用開發和行動應用開發。
- Kotlin 是由 JetBrains 的開發團隊製作的程式語言，支援 JVM 、網頁、桌機等等多平台開發，且支援 Java 程式碼相互運作，也是目前 Android 的主要開發語言

Android Studio

- Android Studio 是 Android 應用程式的官方整合開發環境 (IDE)，以 IntelliJ IDEA 爲基礎，專供開發 Android 應用程式使用。除了 IntelliJ 強大的程式碼編輯器和開發人員工具，Android Studio 還提供更多功能，可提升建構 Android 應用程式的工作效率

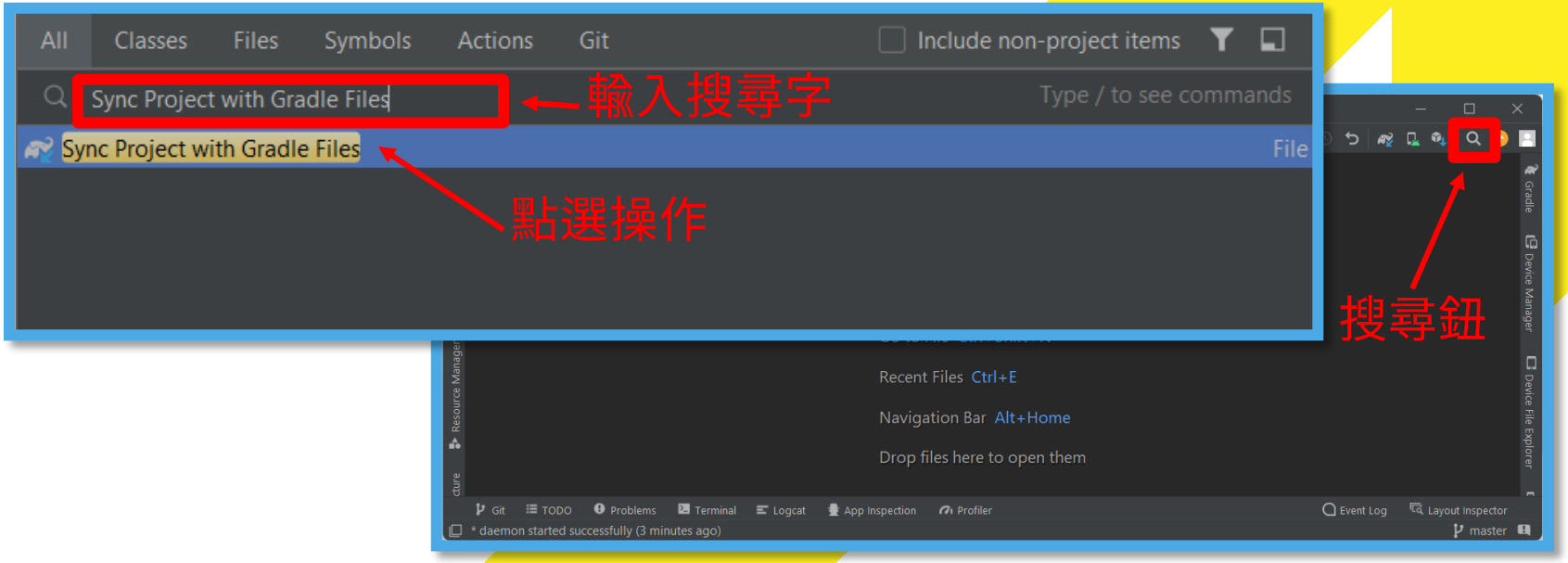
連結 (按我前往)

Android studio　Java　Kotlin 安裝指南

2.
專案介紹

前言

若不了解本文表達的 Android Studio 操作名詞，可點直接點選搜尋框搜尋



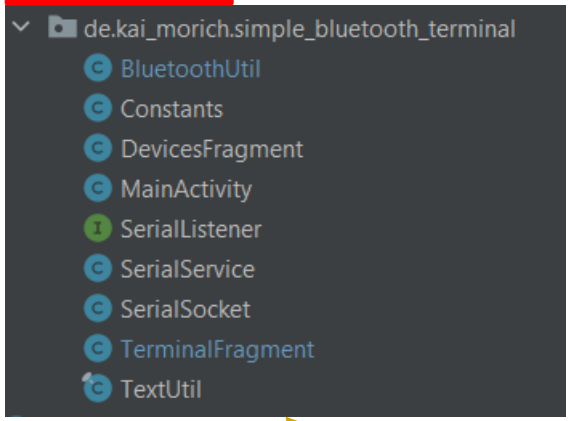
初步建置

1. 請使用 Android Studio 開啟資料夾中的project目錄
2. 執行「Sync Project with Gradle Files」
3. 執行「Make Project」確認檔案能正常執行
4. 依照你的需求修改程式

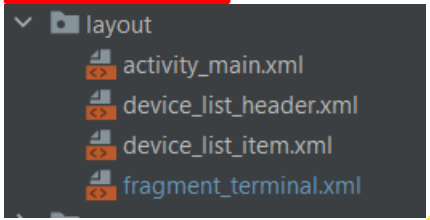
2-1.
檔案結構介紹

檔案結構

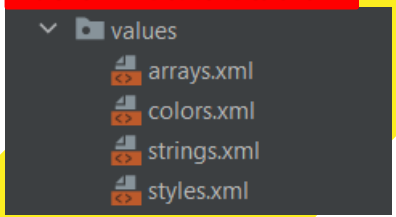
APP 邏輯



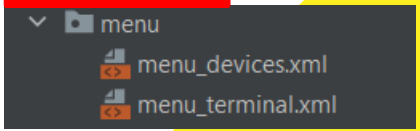
APP 介面



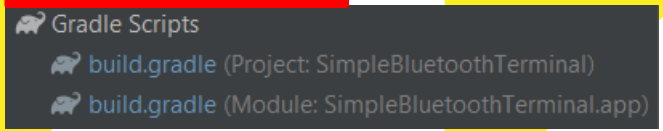
非程式相關檔案



APP 介面

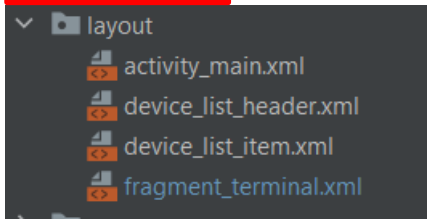


自動化建構工具

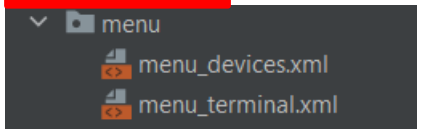


後續章節再介紹

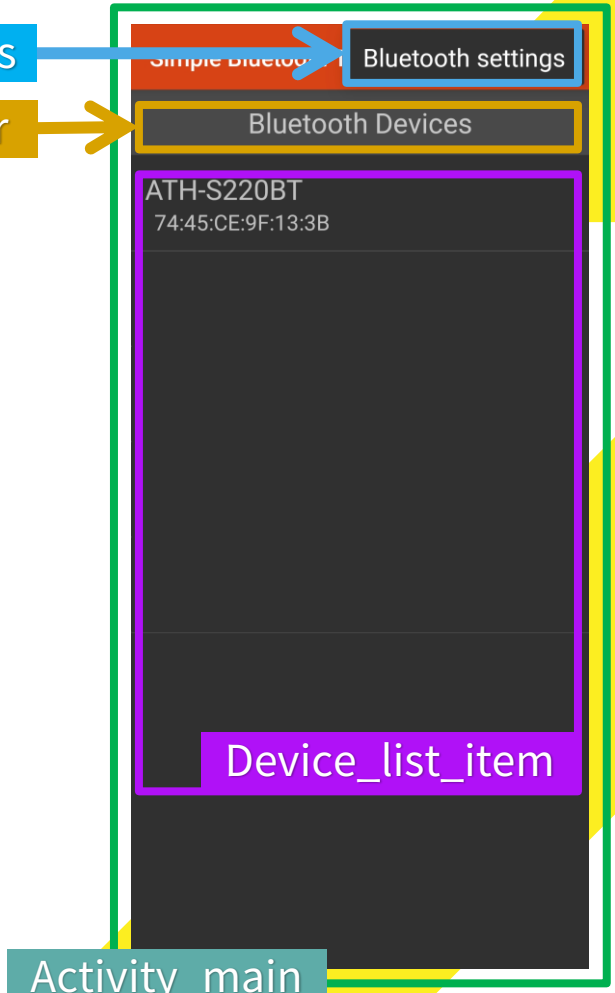
APP 介面

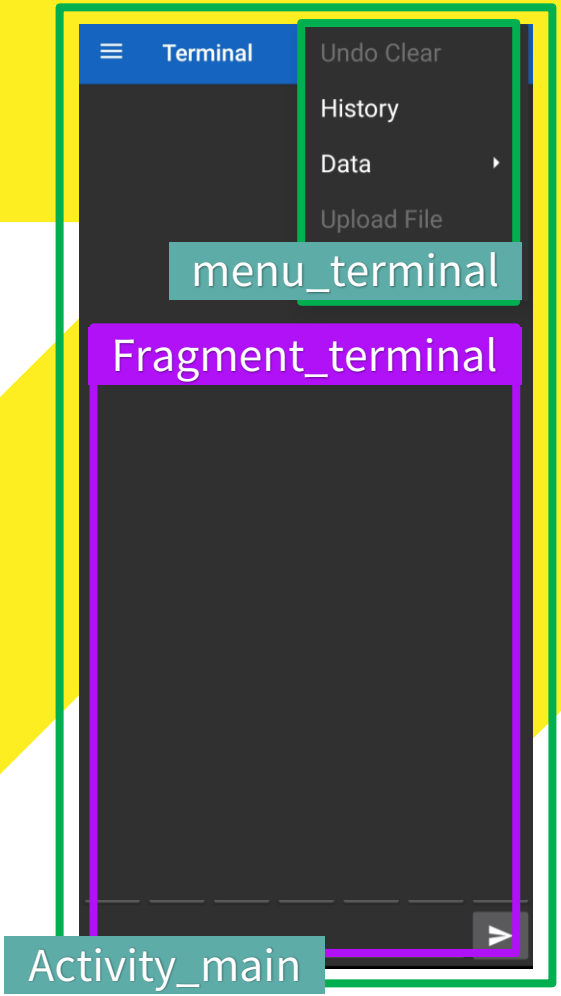


APP 介面

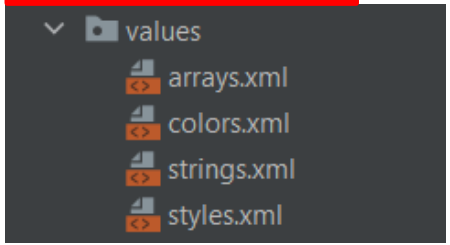


可藉由修改對應的文件
改變介面





非程式相關檔案



Color.xml 定義 APP 主題色彩，UART狀態、文字接收與發送的顏色。

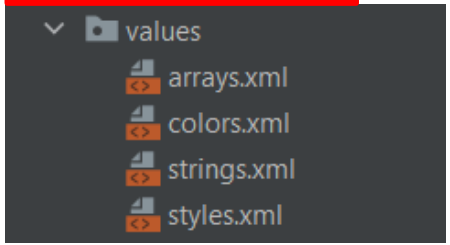
color.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="colorPrimary">#d84315</color>
4      <color name="colorPrimaryDark">#bf360c</color>
5      <color name="colorAccent">#ff6e40</color>
6
7      <color name="colorRecieveText">#00FF00</color>
8      <color name="colorSendText">#82CAFF</color>
9      <color name="colorStatusText">#FFDB58</color>
10 </resources>
11
```

主題色：
主題色(夜間模式下)：
強調色：

USRT 紀錄文字接收的顏色：
USRT 紀錄文字發送的顏色：
USRT 紀錄連接狀態的顏色：

非程式相關檔案



String.xml 用於統一 APP 中的文字

string.xml

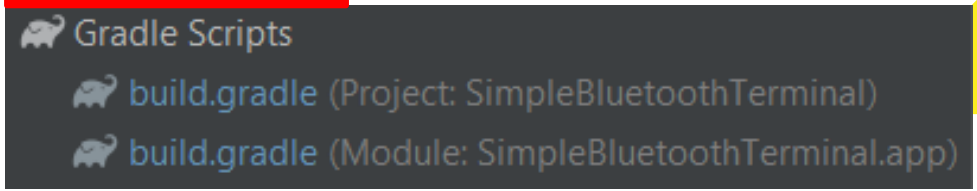
APP 名稱：
Device list 的標題：

存取藍牙權限標題
當接受存取藍牙權限
當被拒絕存取藍牙權限

```
<resources>
    <string name="app_name">Simple Bluetooth Terminal</stri
    <string name="devices">Bluetooth Devices</string>

    <string name="bluetooth_permission_title">Bluetooth pe
    <string name="bluetooth_permission_grant">Bluetooth pe
    <string name="bluetooth_permission_denied">Bluetooth p
</resources>
```

自動化建構工具



裡面設定許多參數用於建構 APP 使用，其中常用的設定有：

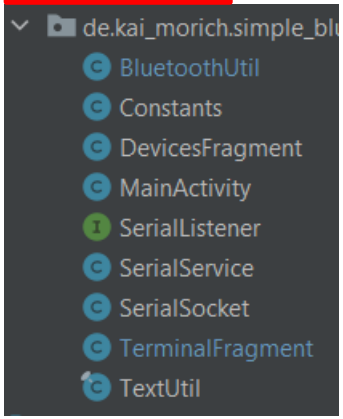
- repositories “函式庫來源”
- minSdkVersion “SDK 最低版本”
- applicationId “應用程式 ID”
- compileOptions “編譯器選項”
- dependencies “依賴套件”

大多參數已經都已經設定好了，
有其他需要請自行添加

2-2.
程式邏輯與介面互動

這裡解釋比較常使用的文件與方法，若有需求可自行鑽研

APP 邏輯



檔案	用途
BluetoothUtil	用於請求藍牙權限，決定裝置列表的排列順序
Devices Fragment	綁定Device_list_header和Device_list_item，並設計藍牙裝置頁面的交互邏輯
Terminal Fragment	綁定Fragment_terminal和menu_terminal，並設計終端機頁面的交互邏輯
SerialListener SerialService SerialSocket	定義串列傳輸介面與操作方法，並作爲伺服器常駐後台。
TextUtil	用於轉換文字格式

在 TerminalFragment 中，有幾個特別重要的參數

```
private TextView receiveText;
```

收集所有接收的文字

```
private void send(String str) {...}
```

UART 發送文字

```
private void receive(ArrayDeque<byte[]> datas) {...}
```

UART 接收文字

知道架構之後就可以開始實作 (後面章節會有範例展示)

2-3.
匯出成APK檔案

建置APK位置

1. 執行「Make Project」

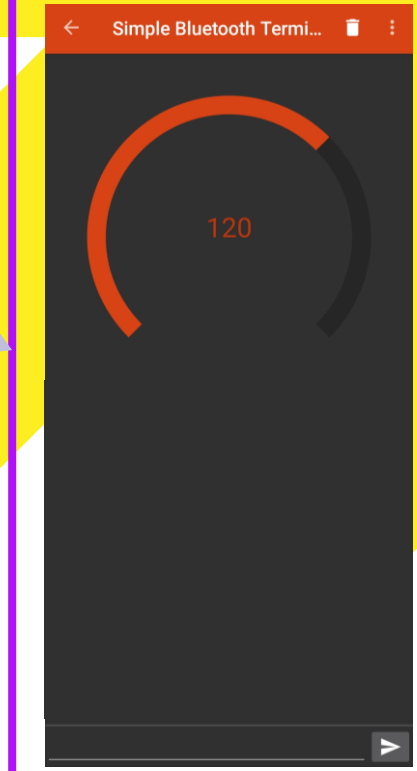
2. 請在專案中尋找
「 build/app/outputs/apk/debug/app-debug.apk 」

3.
範例展示 - 簡易伏特計

架構圖

顯示端

手機



藍牙無線模組

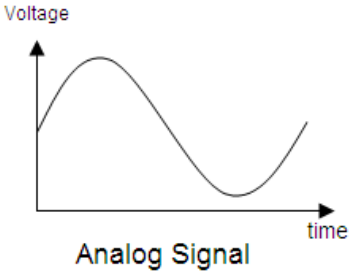
hc05

ADC

8051

測量端

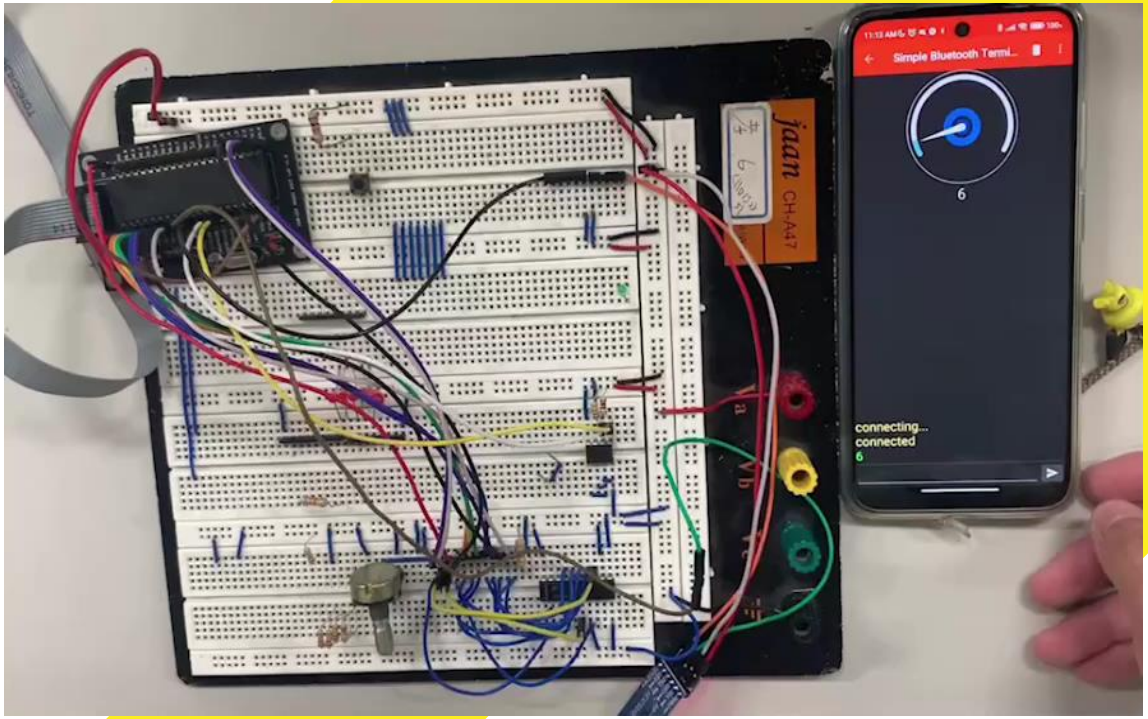
輸入訊號



影片範例

可變電阻輸出可變電壓，接上 ADC 之後交由 8051分析，並將電壓數值利用藍牙模組送入手機端顯示

請觀賞檔案中的影片「影片範例.mp4」



3-1.
製作教學

增加圓形表盤套件

Sildian/CircularSlider-Android

LINK

Get started

Implementing this view is really simple, you'll see !

First, add this line into your root build.gradle :

```
allprojects {
        repositories {
                ...
                maven { url 'https://jitpack.io' }
        }
}
```

Then add the dependency into your app build.gradle :

```
dependencies {
        implementation 'com.github.Sildian:CircularSlider-Android:1.0.0'
}
```

That's it ! You can now user the Circular Slider into your project. Just write this code into your layout :

```
<com.sildian.apps.circularsliderlibrary.CircularSlider
    android:id="@+id/activity_main_circular_slider_1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"/>
```

或是SuperKung/DashboardView也可以

LINK

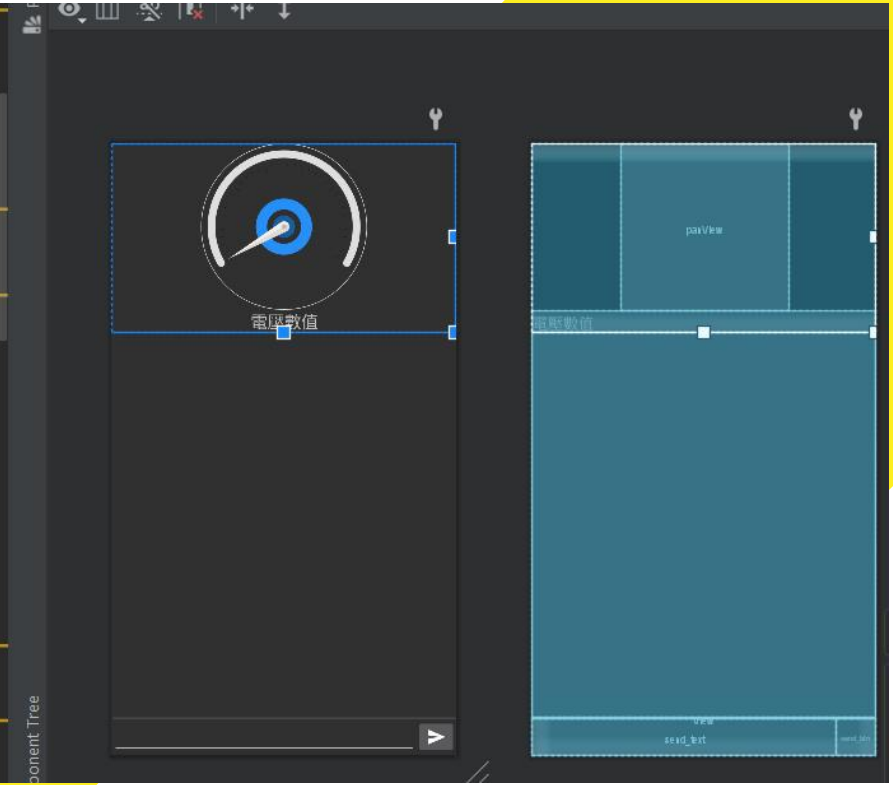
將圓形表盤加入畫面

Fragment_Terminal.xml

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <com.anderson.dashboardview.view.DashboardView
        android:id="@+id/panView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="当前速度" />


    <TextView
        android:id="@+id/panTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:text="電壓數值"
        android:textSize="20sp" />
```



綁定圓形表盤

TerminalFragment.java

```
public class TerminalFragment extends Fragment implements ServiceConnection, SerialListener {

    private DashboardView panView;
```

儲存圓形表盤的變數

```
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
        View view = inflater.inflate(R.layout.fragment_terminal, container, attachToRoot: false);
        receiveText = view.findViewById(R.id.receive_text);                          // TextView performance de
        receiveText.setTextColor(getResources().getColor(R.color.colorRecieveText)); // set as default color to
        receiveText.setMovementMethod(ScrollingMovementMethod.getInstance());

        panView = view.findViewById(R.id.panView);
```

在生命週期onCreateView的時候抓取圓形表盤

綁定圓形表盤

TerminalFragment.java

```
private void receive(ArrayDeque<byte[]> datas) {
    SpannableStringBuilder spn = new SpannableStringBuilder();
    for (byte[] data : datas) {...}
    receiveText.append(spn);
    String data = spn.toString();

    int a = Integer.parseInt(data);
    panView.setPercent(a);
    panView.setText(String.valueOf(a));
}
```

當接收來自8051的電壓數值時
連帶設定圓形表盤的指針

此時當8051不停送出測量端的電壓數值時，APP也會即時變化，達到類似電壓表的功能。

資料引用

爲了方便說明，在過程中會使用「Serial Bluetooth Terminal」（也就是謝欽旭教授的上課工具）簡化後的原始碼進行說明，專案授權延續原作者聲明。

Serial Bluetooth Terminal 原作者簡介

原作者姓名：kai-morich

原作者授權聲明 ：MIT License (聲明內容請查看「原作者 License.txt」)

原作者官網

專案引用來源

到這份檔案最後，偷偷給你看我之前在校園裡拍的貓咪。

一臉  呆呆的

圖片 by 虎库里儿 獸獸畫家

Thanks!

感謝您的觀賞！