1

APP與微控器整合

2017中山大學電機實驗營

2

Outline

- App開發環境-Android Studio介紹
- App開發
- 實驗營App流程
- App與微控器整合

3

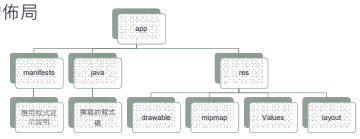
Android Studio介紹

- **Android Studio**是一個為Android平台開發程式的整合式開發環境，可供開發者免費使用，並在Windows、OS X、Linux平台上均可執行。

4

專案結構

- 一個專案下有：manifests、java、res三個資料夾
- Manifests：宣示說明檔案可向 Android 系統顯示應用程式的基本資訊，也就是系統在執行該應用程式的任何程式碼之前必須具備的資訊。
- Java資料夾中放的是所撰寫的程式碼
- Res資料夾下有：drawable、layout、mipmap、value
  - Drawable、mipmap、value放的是宣告的素材、圖片…等
  - Layout放的是app介面的佈局



5

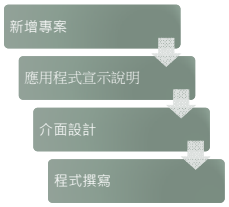
何謂Activity

- 最簡單的就是你可以把Activity看成一個User Interface Program，原則上它會提供使用者一個互動式的介面功能，那一個activity只有一個UI嗎？
- 當然不是囉，
  - 舉例來說：一個EMail程式，就可能包含三個activity
  - (1)郵件列表的activity
  - (2)顯示郵件內容的activity
  - (3)寫新郵件或回覆郵件的activity。

10

App開發流程

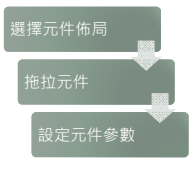
- 應用程式宣示說明：
  - 為應用程式的 Java 封裝命名。
  - 宣告應用程式必須擁有哪些權限
  - 描述應用程式的元件 — 組成應用程式的 Activity、服務、廣播接收器和內容供應程式。
  - 宣告應用程式要求的最低 Android API 級別。
  - 列出應用程式必須連結的程式庫。

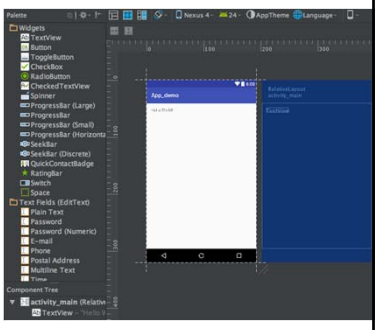


11

App介面設計

- 有兩種設計介面方法：
  - 拖拉元件、寫程式





12

程式撰寫流程

- 依照元件id宣告元件
- 不同元件有不同觸發條件
- 依照需求可執行多執行緒
- 只能在主程式的thread改變UI介面

13

程式撰寫

```
package mpd.app_demo;    //應用程式名稱

import android.support.v7.app.AppCompatActivity;    //引入所需library
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
    final Button front_Button = (Button) findViewById(R.id.btn_front); //宣告按鍵
    @Override
    protected void onCreate(Bundle savedInstanceState) {    //程式初始化
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        front_Button.setOnClickListener(new View.OnClickListener() {      //按鍵觸發才執行
            @Override
            public void onClick(View v) {
            /***do something***/
            }
        });
    }
}
```

14

App使用多Activity

- 開啟一個activity
  - Intent intent = new Intent(this, CarActivity.class);
    startActivity(intent);
- 關閉activity
  - finish();
- 在多activity之間共用socket
  - 將socket宣告成public static
    - public static Socket *control_socket* = null;
    - MainActivity.*connectedOutputStream*.write(result_L);

15

實驗營App流程-藍芽

```
if (!getPackageManager().hasSystemFeature(PackageManager.FEATURE_BLUETOOTH)){
    Toast.makeText(this,
            "FEATURE_BLUETOOTH NOT support",
            Toast.LENGTH_LONG).show();
    finish();
    return;
}
bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if (bluetoothAdapter == null) {
    Toast.makeText(this,
            "Bluetooth is not supported on this hardware platform",
            Toast.LENGTH_LONG).show();
    finish();
    return;
}
```

16

實驗營App流程-藍芽

- 如果藍芽沒有啟用就啟用

```
if (!bluetoothAdapter.isEnabled()) {
    Intent enableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(enableIntent, REQUEST_ENABLE_BT);
}
```

17

實驗營App流程-藍芽

```
Set<BluetoothDevice> pairedDevices =
bluetoothAdapter.getBondedDevices();
if (pairedDevices.size() > 0) {
    pairedDeviceArrayList = new ArrayList<BluetoothDevice>();

    for (BluetoothDevice device : pairedDevices) {
        pairedDeviceArrayList.add(device);
    }
    pairedDeviceAdapter = new ArrayAdapter<BluetoothDevice>(this,
            android.R.layout.simple_list_item_1, pairedDeviceArrayList);
    listViewPairedDevice.setAdapter(pairedDeviceAdapter);
}
```

18

實驗營App流程-藍芽

```
myThreadConnectBTdevice = new ThreadConnectBTdevice(device);
myThreadConnectBTdevice.start();

bluetoothSocket = device.createRfcommSocketToServiceRecord(myUUID);
bluetoothSocket.connect();
```

藍芽連線後執行ThreadConnected背景執行緒

```
startThreadConnected(bluetoothSocket);
```

19

實驗營App流程-藍芽

- 前面有介紹過什麼是Acticity，這裡就是一個在一個Activity開啟另一個Activity的範例
  - CarActivity是我們要開啟的Activity也就是我們車輛控制的地方

```
public void start_car_control(){
    Intent intent = new Intent(this, CarActivity.class);
    startActivity(intent);
}
```

20

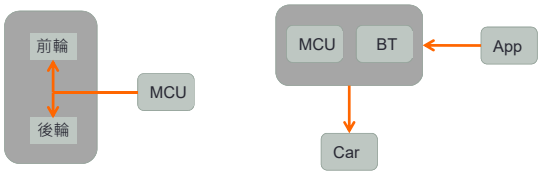
實驗營App流程-車輛控制

- 拖拉seekBar來調整速度、轉彎
- 按下“停止”按鈕馬上讓車輛停止
- 按下“置中按鈕”讓車輛方向置中
- 按下“右”按鈕、“左”按鈕來微調方向
- 按下“前”按鈕、“後”按鈕來微調速度

21

App與微控器整合

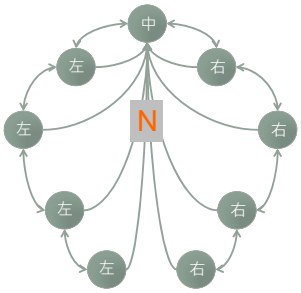
- App透過藍芽與微控器溝通
- 微控器依照收到的指令去控制車輛



22

App與微控器整合

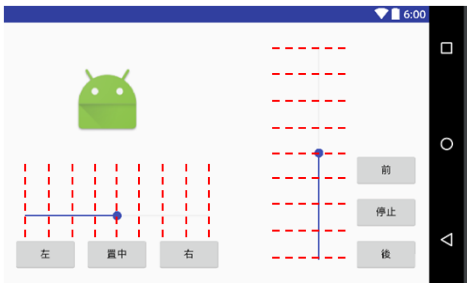
- 車輛控制指令：F、B、S、R、L、N
- 前輪
  - 向右一段：R
  - 向左一段：L
  - 置中：N
- 後輪
  - 加速一段：F
  - 減速一段：B
  - 立即停止：S



23

App與微控器整合

- 四段速度控制
- 四段方向角度控制
- 方向置中
- 一鍵停止



24

App與微控器整合-注意事項

- 藍芽模式的設定
- 微控器與App傳送的封包內容
- 微控器控制車輛的演算法要和App配合