

第二十三章

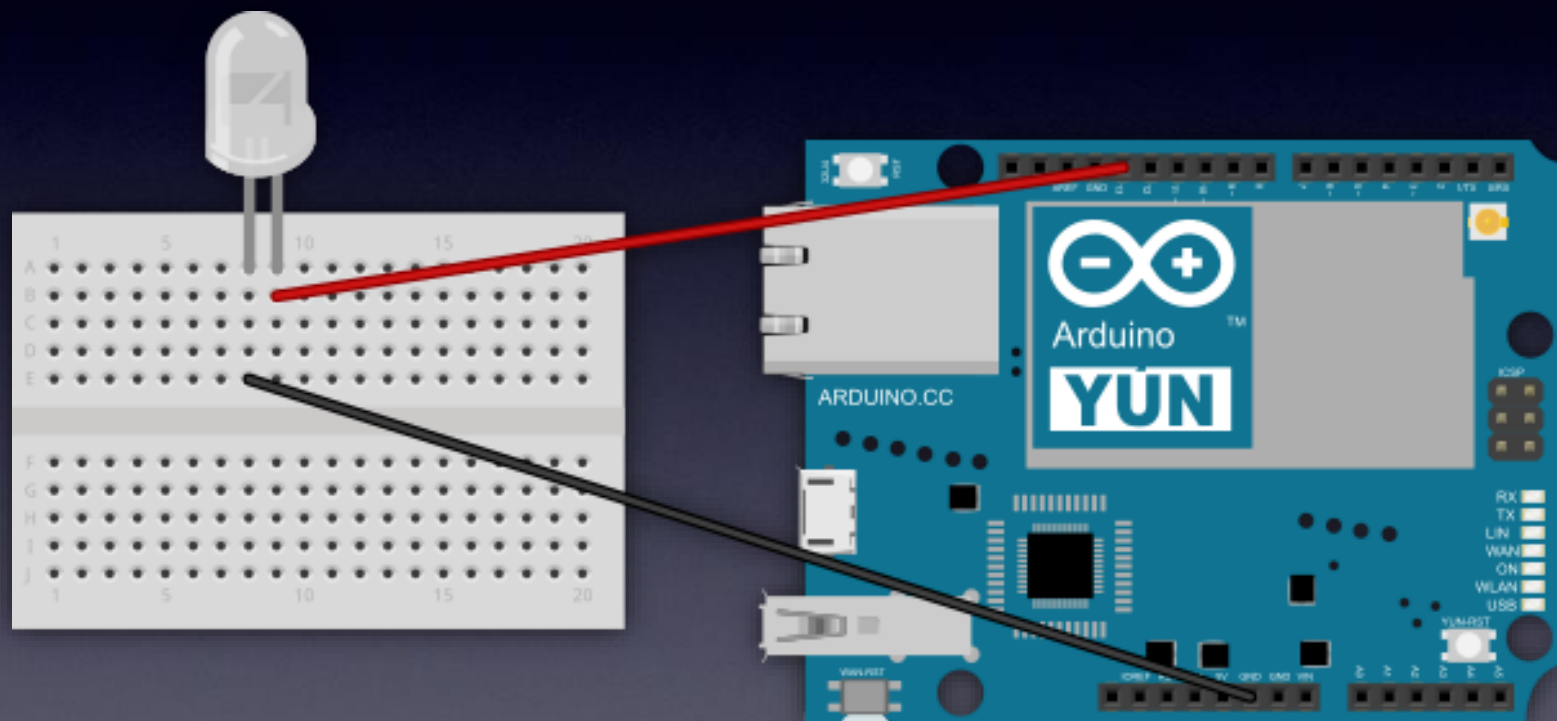
ARDUINO YUN SERVER

課程目的

- 使用Arduino Yun的預設功能，建立簡易的API Server
 - 建立API為：`http://arduino的ip/arduino/digital/1 or 0`
- Android藉由API的呼叫，可以開關與雲版相連的LED

ARDUINO端

線路圖



程式碼說明

BridgeServer server;

建立BridgeServer

```
void setup() {
    pinMode(13, OUTPUT);
    Bridge.begin();
    Console.begin();
    server.listenOnLocalhost();
    server.begin();
}

void loop() {
    BridgeClient client = server.accept();
    if (client) {
        String command = client.readStringUntil('/');
        if (command == "digital") {
            int onoff = client.parseInt();
            onoff = (onoff > 0) ? 1 : 0;
            digitalWrite(13, onoff);
            client.print("OK");
        } else {
            client.print("FAIL");
        }
        client.stop();
    }
}
```

程式碼說明

```
BridgeServer server;
```

Arduino初始化用的函式

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

準備Pin13作為輸出
讓LED亮起來

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

初始化Bridge和Console

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```


程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

設定Server聆聽Port 5555

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

啟動Server

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

進入Loop

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

藉由BridgeServer的accept()
來取得連線進入的client

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

若有取得client，表示有連線連入

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

讀取連線帶入的網址
讀取直到 '/' 出現為止

程式碼說明

```
BridgeServer server;  
  
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}  
  
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

本例子的API是arduino/digital/1
第一個arduino會被雲版處理掉
所以傳入的完整字串會是digital/1

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
  pinMode(13, OUTPUT);  
  Bridge.begin();  
  Console.begin();  
  server.listenOnLocalhost();  
  server.begin();  
}
```

```
void loop() {  
  BridgeClient client = server.accept();  
  if (client) {  
    String command = client.readStringUntil('/');  
    if (command == "digital") {  
      int onoff = client.parseInt();  
      onoff = (onoff > 0) ? 1 : 0;  
      digitalWrite(13, onoff);  
      client.print("OK");  
    } else {  
      client.print("FAIL");  
    }  
    client.stop();  
  }  
}
```

readStringUntil('/')表示將傳入的字串
讀入至/為止，所以代表取得的是digital

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

若取得的字串是digital

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
  pinMode(13, OUTPUT);  
  Bridge.begin();  
  Console.begin();  
  server.listenOnLocalhost();  
  server.begin();  
}
```

```
void loop() {  
  BridgeClient client = server.accept();  
  if (client) {  
    String command = client.readStringUntil('/');  
    if (command == "digital") {  
      int onoff = client.parseInt();  
      onoff = (onoff > 0) ? 1 : 0;  
      digitalWrite(13, onoff);  
      client.print("OK");  
    } else {  
      client.print("FAIL");  
    }  
    client.stop();  
  }  
}
```

接著處理readStringUntil()
後剩下的字串

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
  pinMode(13, OUTPUT);  
  Bridge.begin();  
  Console.begin();  
  server.listenOnLocalhost();  
  server.begin();  
}
```

```
void loop() {  
  BridgeClient client = server.accept();  
  if (client) {  
    String command = client.readStringUntil('/');  
    if (command == "digital") {  
      int onoff = client.parseInt();  
      onoff = (onoff > 0) ? 1 : 0;  
      digitalWrite(13, onoff);  
      client.print("OK");  
    } else {  
      client.print("FAIL");  
    }  
    client.stop();  
  }  
}
```

呼叫parseInt()表示將剩下的字串
試著轉換為整數

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
  pinMode(13, OUTPUT);  
  Bridge.begin();  
  Console.begin();  
  server.listenOnLocalhost();  
  server.begin();  
}
```

```
void loop() {  
  BridgeClient client = server.accept();  
  if (client) {  
    String command = client.readStringUntil('/');  
    if (command == "digital") {  
      int onoff = client.parseInt();  
      onoff = (onoff > 0) ? 1 : 0;  
      digitalWrite(13, onoff);  
      client.print("OK");  
    } else {  
      client.print("FAIL");  
    }  
    client.stop();  
  }  
}
```

若轉換失敗，就是0，成功就是將字串
轉變為數字

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

若轉換結果發現是大於0的數字
就以1來替代，不然就是0

程式碼說明

```
BridgeServer server;

void setup() {
  pinMode(13, OUTPUT);
  Bridge.begin();
  Console.begin();
  server.listenOnLocalhost();
  server.begin();
}

void loop() {
  BridgeClient client = server.accept();
  if (client) {
    String command = client.readStringUntil('/');
    if (command == "digital") {
      int onoff = client.parseInt();
      onoff = (onoff > 0) ? 1 : 0;
      digitalWrite(13, onoff);
      client.print("OK");
    } else {
      client.print("FAIL");
    }
    client.stop();
  }
}
```

針對Pin13
以onoff最後判斷的整數為第二個參數
若是1，LED就會亮起
若是0，則LED會熄滅

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

回覆連線的客戶端結果 OK

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

若上述判斷不合，回覆給連線客戶端FAIL

程式碼說明

```
BridgeServer server;
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    Bridge.begin();  
    Console.begin();  
    server.listenOnLocalhost();  
    server.begin();  
}
```

```
void loop() {  
    BridgeClient client = server.accept();  
    if (client) {  
        String command = client.readStringUntil('/');  
        if (command == "digital") {  
            int onoff = client.parseInt();  
            onoff = (onoff > 0) ? 1 : 0;  
            digitalWrite(13, onoff);  
            client.print("OK");  
        } else {  
            client.print("FAIL");  
        }  
        client.stop();  
    }  
}
```

呼叫stop停止連線

測試

- 找一台和雲板在**同一個無線網路環境下**的電腦
- 打開瀏覽器，在網址列輸入
<http://arduino.local/arduino/digital/0> 關閉LED
<http://arduino.local/arduino/digital/1> 開啟LED

ANDROID端

事前準備

- 找出雲板目前的IP位置
 - 登入<http://arduino.local>就可以取得
 - Android手機上無法直接支援Bonjour service，所以找不到<http://arduino.local>，故須使用ip來做連線(ios沒問題)
- Android與雲板的連線可以使用網際網路但必須從Router設定，詳細資料可參見下方連結
 - <http://forum.arduino.cc/index.php?topic=191344.0>

ANDROID MANIFEST

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.arduinocontrol">
```

```
    <uses-permission android:name="android.permission.INTERNET" />
```

```
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

一定要寫上使用Internet的權限

GRADLE

```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 23
    buildToolsVersion "23.0.3"
    defaultConfig {
        applicationId "com.example.arduinocontrol"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), ...
        }
    }
}
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
    compile 'com.android.volley:volley:1.0.0'
}
```

本範例使用volley作為
網路連線函式庫

LAYOUT

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/background"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFF"
    tools:context="com.example.arduinocontrol.MainActivity">

    <Switch
        android:id="@+id/switch1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

使用Switch作為LED的開關

ACTIVITY

設定找到的Arduino Yun IP

```
public class MainActivity extends AppCompatActivity {
    private static final String ARDUINO_YUN_IP = "192.168.43.169";
    private Switch mLightSwitch;
    private View mBackground;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mBackground = findViewById(R.id.background);
        mBackground.setBackgroundColor(Color.BLACK);
        mLightSwitch = (Switch) findViewById(R.id.switch1);
        mLightSwitch.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton button, boolean isChecked) {
                if (!isChecked) {
                    changeBackgroundColor(Color.WHITE, Color.BLACK);
                    sendLightOnOffRequest(false);
                } else {
                    changeBackgroundColor(Color.BLACK, Color.WHITE);
                    sendLightOnOffRequest(true);
                }
            }
        });
    }
    .....
}
```


ACTIVITY

```
public class MainActivity extends AppCompatActivity {
    private static final String ARDUINO_YUN_IP = "192.168.43.169";
    private Switch mLightSwitch;
    private View mBackground;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mBackground = findViewById(R.id.background);
        mBackground.setBackgroundColor(Color.BLACK);
        mLightSwitch = (Switch) findViewById(R.id.switch1);
        mLightSwitch.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton button, boolean isChecked) {
                if (!isChecked) {
                    changeBackgroundColor(Color.WHITE, Color.BLACK);
                    sendLightOnOffRequest(false);
                } else {
                    changeBackgroundColor(Color.BLACK, Color.WHITE);
                    sendLightOnOffRequest(true);
                }
            }
        });
    }
    .....
}
```

使用findViewById()找到Switch
並且設定監聽器來確認Switch是開啟或關閉

ACTIVITY

```
public class MainActivity extends AppCompatActivity {
    private static final String ARDUINO_YUN_IP = "192.168.43.169";
    private Switch mLightSwitch;
    private View mBackground;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mBackground = findViewById(R.id.background);
        mBackground.setBackgroundColor(Color.BLACK);
        mLightSwitch = (Switch) findViewById(R.id.switch1);
        mLightSwitch.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton button, boolean isChecked) {
                if (!isChecked) {
                    changeBackgroundColor(Color.WHITE, Color.BLACK);
                    sendLightOnOffRequest(false);
                } else {
                    changeBackgroundColor(Color.BLACK, Color.WHITE);
                    sendLightOnOffRequest(true);
                }
            }
        });
    }
    .....
}
```

當Switch被開關時，onCheckedChanged
就會被呼叫，參數2就是Switch目前狀態

ACTIVITY

```
public class MainActivity extends AppCompatActivity {
    private static final String ARDUINO_YUN_IP = "192.168.43.169";
    private Switch mLightSwitch;
    private View mBackground;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mBackground = findViewById(R.id.background);
        mBackground.setBackgroundColor(Color.BLACK);
        mLightSwitch = (Switch) findViewById(R.id.light_switch);
        mLightSwitch.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton button, boolean isChecked) {
                if (!isChecked) {
                    changeBackgroundColor(Color.WHITE, Color.BLACK);
                    sendLightOnOffRequest(false);
                } else {
                    changeBackgroundColor(Color.BLACK, Color.WHITE);
                    sendLightOnOffRequest(true);
                }
            }
        });
    }
    .....
}
```

當Switch是開關時，呼叫
changeBackgroundColor()更改背景顏色及
sendLightOnOffRequest()與arduino溝通

ACTIVITY

ObjectAnimator是專門做Android物件動畫的工具

```
private void changeBackgroundColor(int startColor, int endColor) {  
    ObjectAnimator colorFade =  
        ObjectAnimator.ofObject(  
            mBackground,  
            "backgroundColor",  
            new ArgbEvaluator(),  
            startColor, endColor);  
    colorFade.setDuration(3000);  
    colorFade.start();  
}
```

ACTIVITY

```
private void changeBackgroundColor(int startColor, int endColor) {  
    ObjectAnimator colorFade =  
        ObjectAnimator.ofObject(  
            mBackground,  
            "backgroundColor",  
            new ArgbEvaluator(),  
            startColor, endColor);  
    colorFade.setDuration(3000);  
    colorFade.start();  
}
```

使用ObjectAnimator.ofObject()建立物件
參數1是動畫要使用在哪個物件上
參數2是該物件的哪個屬性要改變
參數3是動畫撥放時該如何改變，範例使用預設的ArgbEvaluator()
參數4與5 表是動畫開始狀態和結束狀態

ACTIVITY

```
private void changeBackgroundColor(int startColor, int endColor) {  
    ObjectAnimator colorFade =  
        ObjectAnimator.ofObject(  
            mBackground,  
            "backgroundColor",  
            new ArgbEvaluator(),  
            startColor, endColor);  
    colorFade.setDuration(3000);  
    colorFade.start();  
}
```

使用setDuration設定動畫撥放的時間場短
呼叫start()開始撥放動畫

ACTIVITY

依照傳入的參數決定要呼叫雲板的哪個API

```
private void sendLightOnOffRequest(boolean isOn) {
    String url = (isOn) ?
        "http://" + ARDUINO_YUN_IP + "/arduino/digital/1" :
        "http://" + ARDUINO_YUN_IP + "/arduino/digital/0";
    StringRequest request =
        new StringRequest(Request.Method.GET, url,
            mOnSuccessListener, mOnErrorListener);
    NetworkManager.getInstance(MainActivity.this).request(null, request);
}

private Response.Listener<String> mOnSuccessListener =
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d("NetworkResponse", response);
        }
    };

private Response.ErrorListener mOnErrorListener =
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("NetworkError", "", error);
            Toast.makeText(MainActivity.this, "Error!!", Toast.LENGTH_SHORT).show();
        }
    };
};
```


ACTIVITY

```
private void sendLightOnOffRequest(boolean isOn) {
    String url = (isOn) ?
        "http://" + ARDUINO_IP + "/arduino/digital/0",
        "http://" + ARDUINO_IP + "/arduino/digital/0";
    StringRequest request =
        new StringRequest(Request.Method.GET, url,
            mOnSuccessListener, mOnErrorListener);
    NetworkManager.getInstance(MainActivity.this).request(null, request);
}

private Response.Listener<String> mOnSuccessListener =
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d("NetworkResponse", response);
        }
    };

private Response.ErrorListener mOnErrorListener =
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("NetworkError", "", error);
            Toast.makeText(MainActivity.this, "Error!!", Toast.LENGTH_SHORT).show();
        }
    };
};
```

建立Volley的String Request
使用Get的方式
成功結果回報到mOnSuccessListener
失敗結果回報到mOnErrorListener

ACTIVITY

```
private void sendLightOnOffRequest(boolean isOn) {
    String url = (isOn) ?
        "http://" + ARDUINO_YUN_IP + "/arduino/digital/1" :
        "http://" + ARDUINO_YUN_IP + "/arduino/digital/0";
    StringRequest request =
        new StringRequest(Request.Method.GET, url,
            mOnSuccessListener, mOnErrorListener);
    NetworkManager.getInstance(MainActivity.this).request(null, request);
}

private Response.Listener<String> mOnSuccessListener =
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d("NetworkResponse", response);
        }
    };

private Response.ErrorListener mOnErrorListener =
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("NetworkError", "", error);
            Toast.makeText(MainActivity.this, "Error!!", Toast.LENGTH_SHORT).show();
        }
    };
};
```

藉由NetworkManager的request()方法
發送出StringRequest

ACTIVITY

```
private void sendLightOnOffRequest(boolean isOn) {
    String url = (isOn) ?
        "http://" + ARDUINO_YUN_IP + "/arduino/digital/1" :
        "http://" + ARDUINO_YUN_IP + "/arduino/digital/0";
    StringRequest request =
        new StringRequest(Request.Method.POST, url,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    Log.d("NetworkResponse", response);
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Log.e("NetworkError", "", error);
                    Toast.makeText(MainActivity.this, "Error!!", Toast.LENGTH_SHORT).show();
                }
            });
    NetworkManager.getInstance().addToRequestQueue(request);
}
```

Response.Listener負責接收成功結果
當雲板回傳結果時 onResponse()就會被呼叫到

```
private Response.Listener<String> mOnSuccessListener =
new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        Log.d("NetworkResponse", response);
    }
};

private Response.ErrorListener mOnErrorListener =
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Log.e("NetworkError", "", error);
        Toast.makeText(MainActivity.this, "Error!!", Toast.LENGTH_SHORT).show();
    }
};
```

ACTIVITY

```
private void sendLightOnOffRequest(boolean isOn) {
    String url = (isOn) ?
        "http://" + ARDUINO_YUN_IP + "/arduino/digital/1" :
        "http://" + ARDUINO_YUN_IP + "/arduino/digital/0";
    StringRequest request =
        new StringRequest(Request.Method.GET, url,
            mOnSuccessListener, mOnErrorListener);
    NetworkManager.getInstance(MainActivity.this).request(null, request);
}

private Response.Listener<String> mOnSuccessListener =
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d("NetworkResponse", response);
        }
    };

private Response.ErrorListener mOnErrorListener =
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("NetworkError", "", error);
            Toast.makeText(MainActivity.this, "Error!!", Toast.LENGTH_SHORT).show();
        }
    };
};
```

Response.ErrorListener負責接收失敗結果

補充

- 要讓Android能夠藉由Bonjour Service直接找到Arduino Yun，可以使用Network Service Discovery API，詳情可見以下連結
 - <http://developer.android.com/intl/zh-tw/training/connect-devices-wirelessly/nsd.html#discover>
 - <http://spirit-blog.logdown.com/posts/211216-bonjour-ios-and-android>
- 範例程式碼
 - <https://github.com/dbi1463/BonjourDemo>

Q & A