

中華大學資訊工程學系系統開發專題報告



智慧型手機遠端控制樂高機器設備之研究

專題組員：姚長宇、劉家瑋、謝博文、黃奕翔

指導老師：陳建宏老師

專題編號：PRJ2011-CSIE-10008

執行期間：100 年 06 月 至 101 年 06 月

目錄

壹、摘要.....	3
貳、簡介.....	4
參、專題進行方式.....	5
3-1 安裝 Eclipse & android.....	5
3-2 建立專案.....	6
3-3 介面.....	8
3-4 安裝 NXT 及 LeJOS.....	9
3-5 開發環境.....	12
3-6 樂高機器人的設計.....	13
3-7 機器人程式的運作.....	15
肆、專題分工.....	20
伍、時程規畫.....	20
陸、問題與解決方法.....	21
柒、主要成果.....	21
捌、結語.....	43
玖、銘謝.....	43
拾、參考文獻.....	43

壹、摘要

Android 是目前已成為各家智慧型手機業者競爭的角力場。Android 的當前發展及未來趨勢，根據今年十月 Android 所做的市場調查，Android 手機過去六個月在美國的市佔率排行第一，其次是 Apple 公司的 iPhone。而 Android 手機能夠有這麼大的市佔率，在於 Android 除了是手機作業系統，也是一種開放原始碼(open source)，這不只是代表它是免費的，還包括了其原始碼可自由下載。因此，各家廠商在開發手機的時候，都可用 Android 來設計它們的面板功能。

機器人一向能抓住大小朋友的目光，我們也期待機器人能以更多功能、更全面地出現在我們的生活中。在機器人平台方面，樂高機器人挾其既有大量通用零件以及多元程式開發環境兩大優勢，在教育性機器人市場可說是相當吃得開。可擴充的硬體周邊例如感測器、馬達以及各種擴充板等高達數十種。程式開發環境從文字式的 **C、C++、Java** 到圖形化的 **LabVIEW、MSRDS (Microsoft Robotics Developer Studio)、Google App Inventor**，支援之豐富也是其他機器人平台所未見。

貳、簡介

本專題目的是藉由Android開放手機平台作業系統，讓使用者能藉由藍芽介面，搭建Android手機與LEGO NXT機器人的平台，讓使用者可以用Android手機來操縱LEGO NXT機器人(遙控車)，LEGO NXT機器人也可以回傳資料給Android手機，來達成遠端操控遙控車，機器人也必須跟手機緊密的連繫才行。最基本的就是機器人一做完一組完整的動作後，他會告訴操控器可以再傳值過來，這樣我們就可以確認接收端(NXT機器人)有接收到資料。因此手機每次傳出一筆資料後就必須等待機器人的回應，有了正確的回應就可以在傳下一筆資料了。而控制方面則是在Android手機上，寫一個**方向感應器**感應的程式，來模擬實際的類比搖桿，讓使用者在玩遙控車的時候，多增添了許多樂趣。

參、專題進行方式

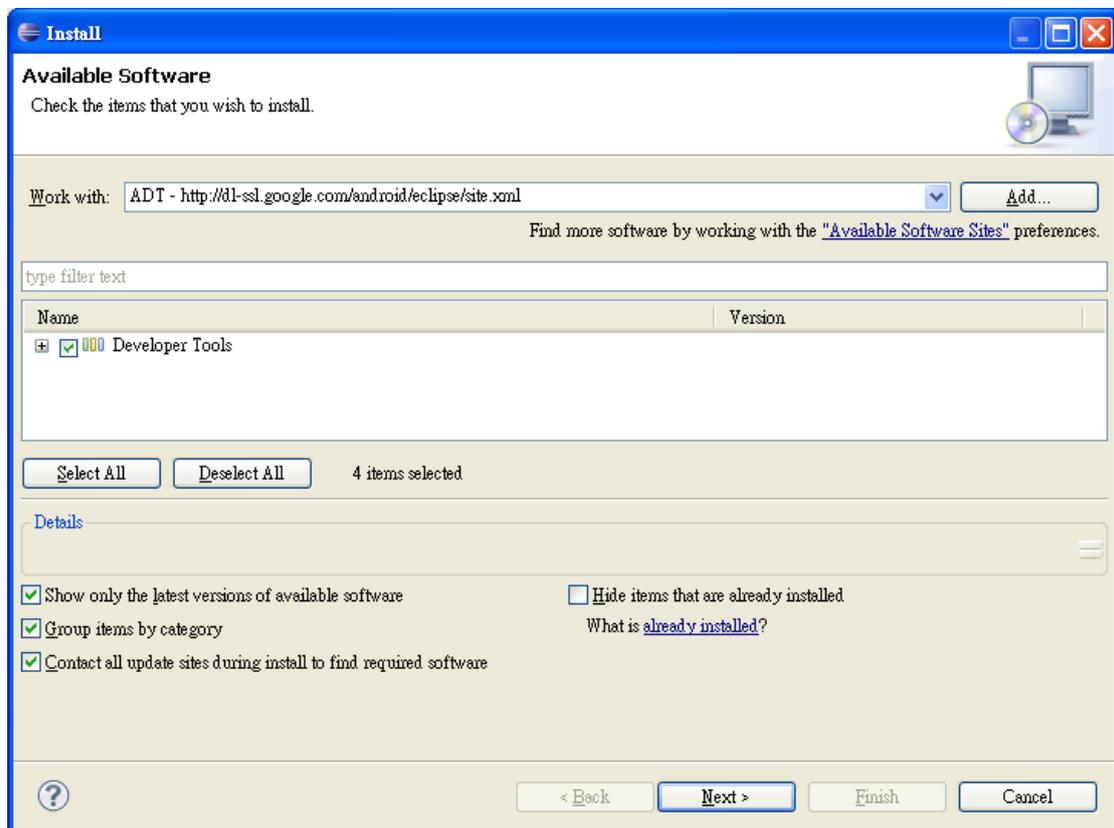
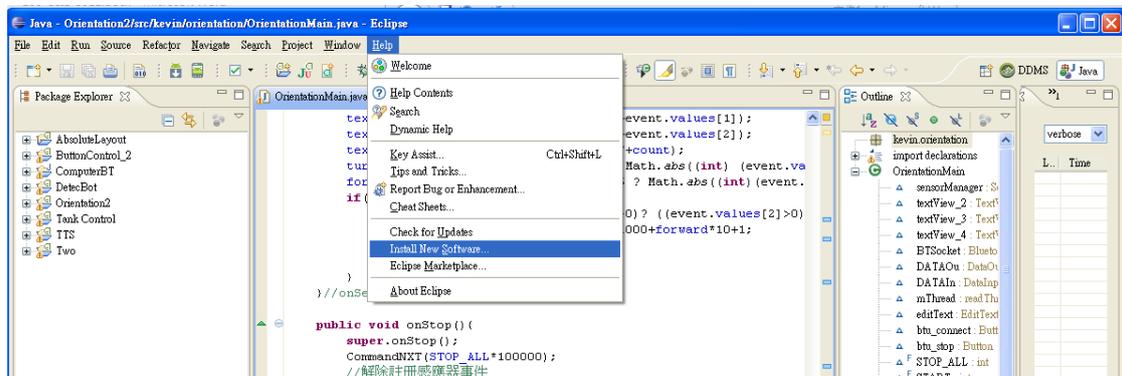
熟悉並安裝環境

首先從尋找資料開始，先在各個書店與圖書館內，尋找有關 android 跟 Lego NXT 的書籍，再練習裡頭的小程式，開始開發並熟悉接下來會用到的元件。

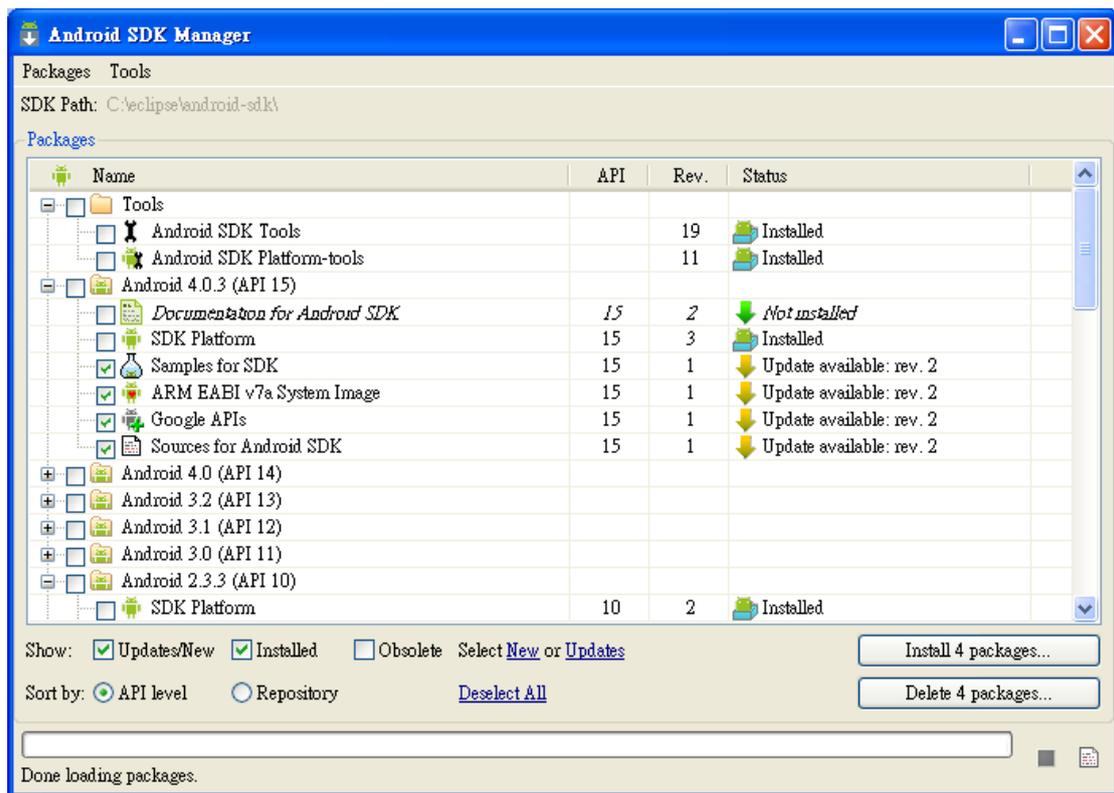
3-1 安裝 Eclipse & android

a. 先安裝 JDK、Eclipse、ADT(Android Development Tools)

ADT 是用 Eclipse 進行 Android 開發的開發工具，它本身不是 Android SDK
安裝方法和同其他 Eclipse 插件的方法一樣 (Help-> Install New Software...)



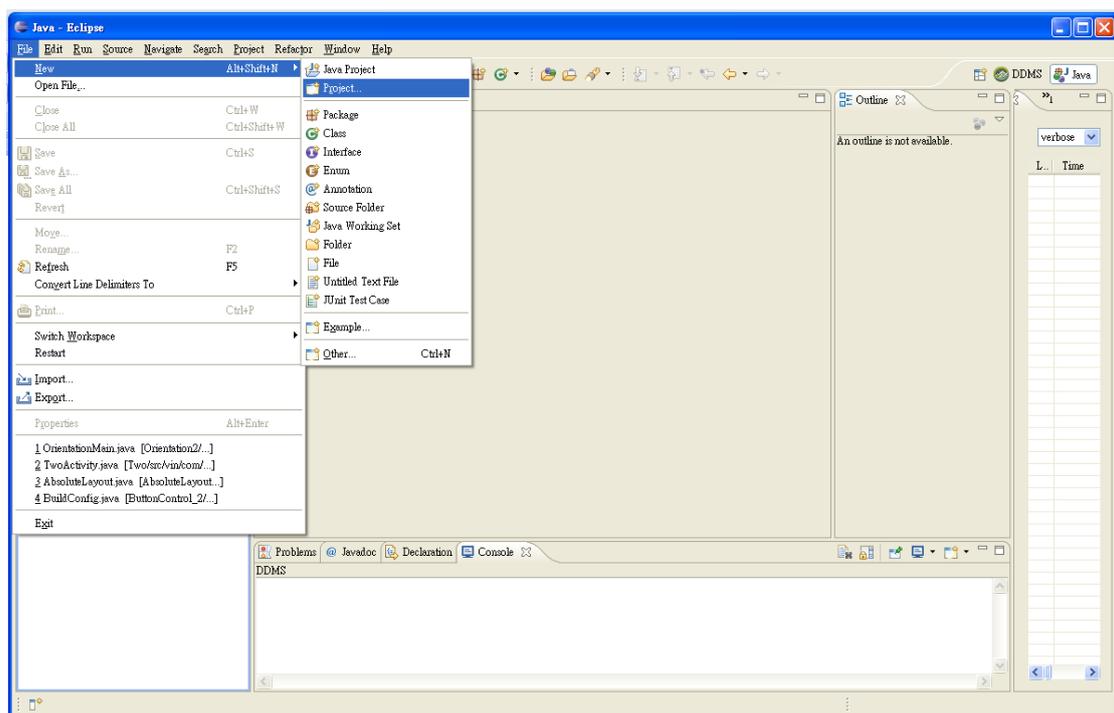
b. SDK Tools 的安裝

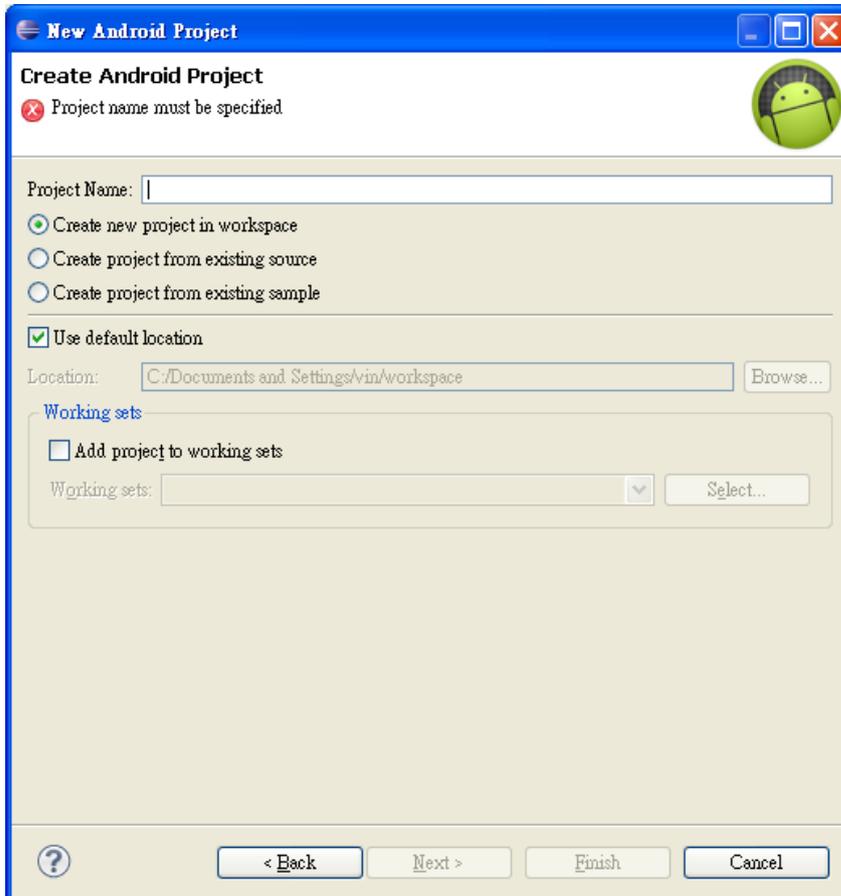
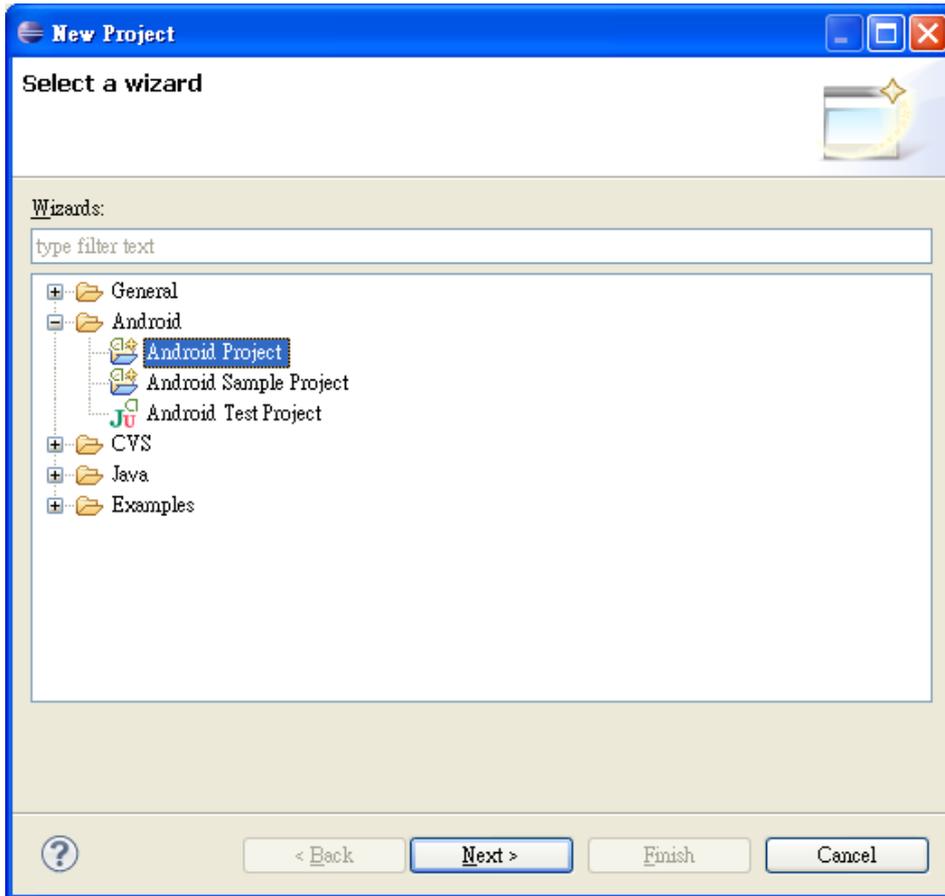


3-2 建立專案

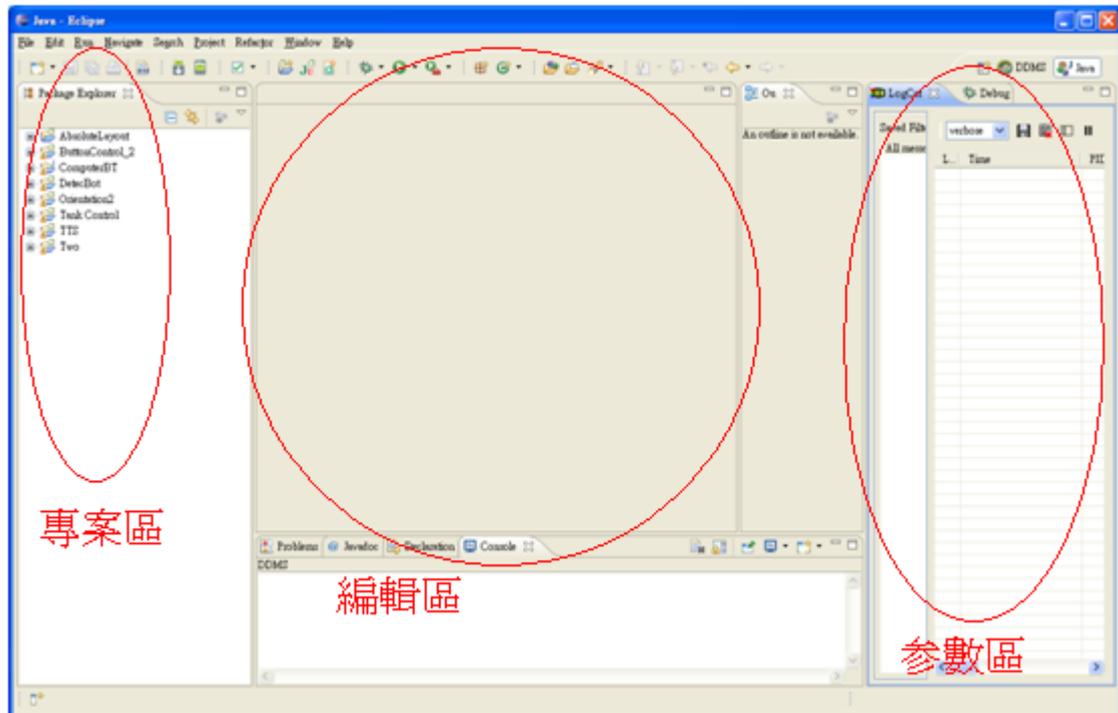
建立一個新專案

File→New→Project→Android Project





3-3 介面



專案區裡的目錄結構為

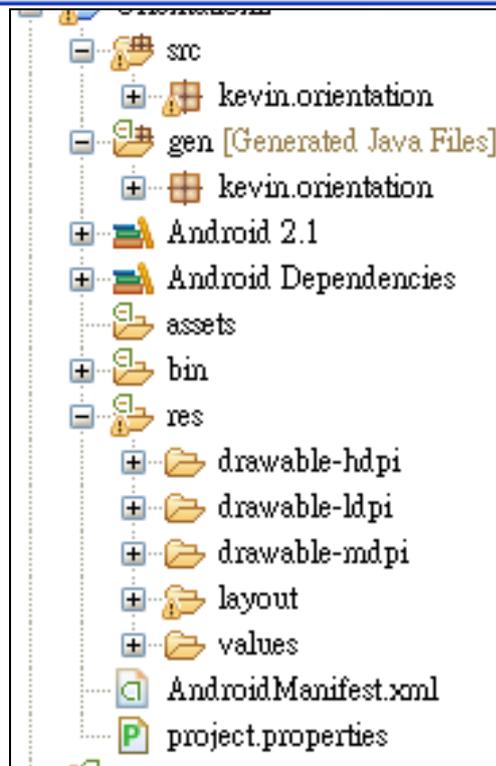
src 程式碼所在

gen 資源檔程式碼

res 是資料檔目錄

存放圖檔、音樂檔...等

畫面設計好後會存放在 main.xml 裡。

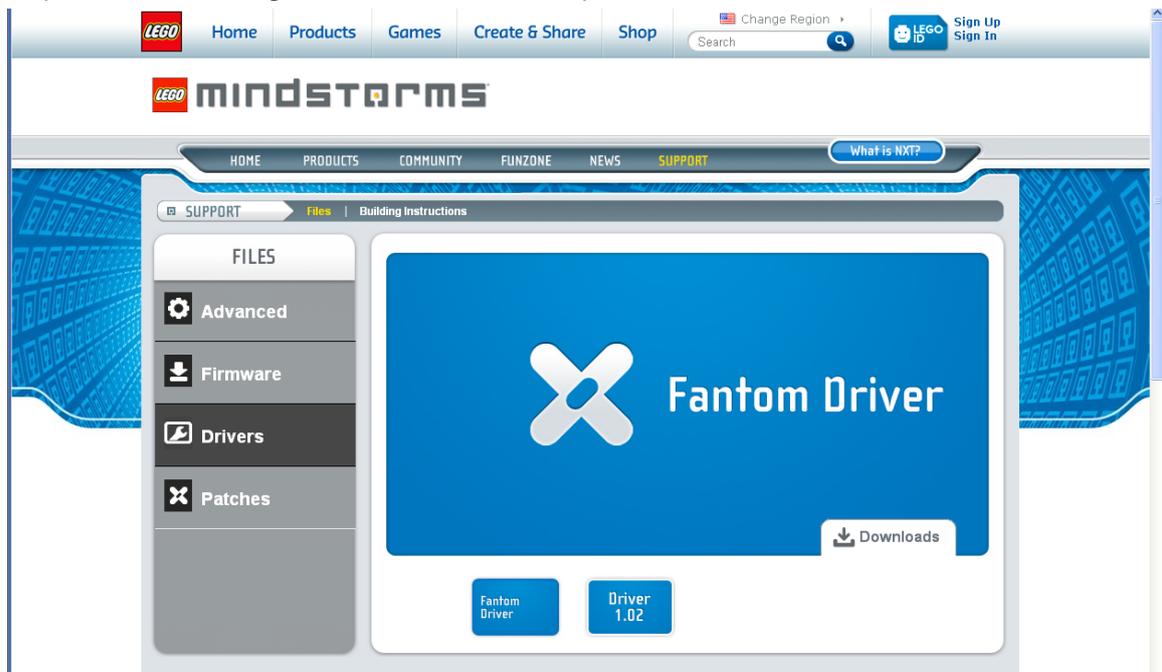


3-4 安裝 NXT 及 LeJOS

3-4-1 安裝 NXT 驅動

進入 LEGO 官網

<http://mindstorms.lego.com/en-us/Default.aspx>

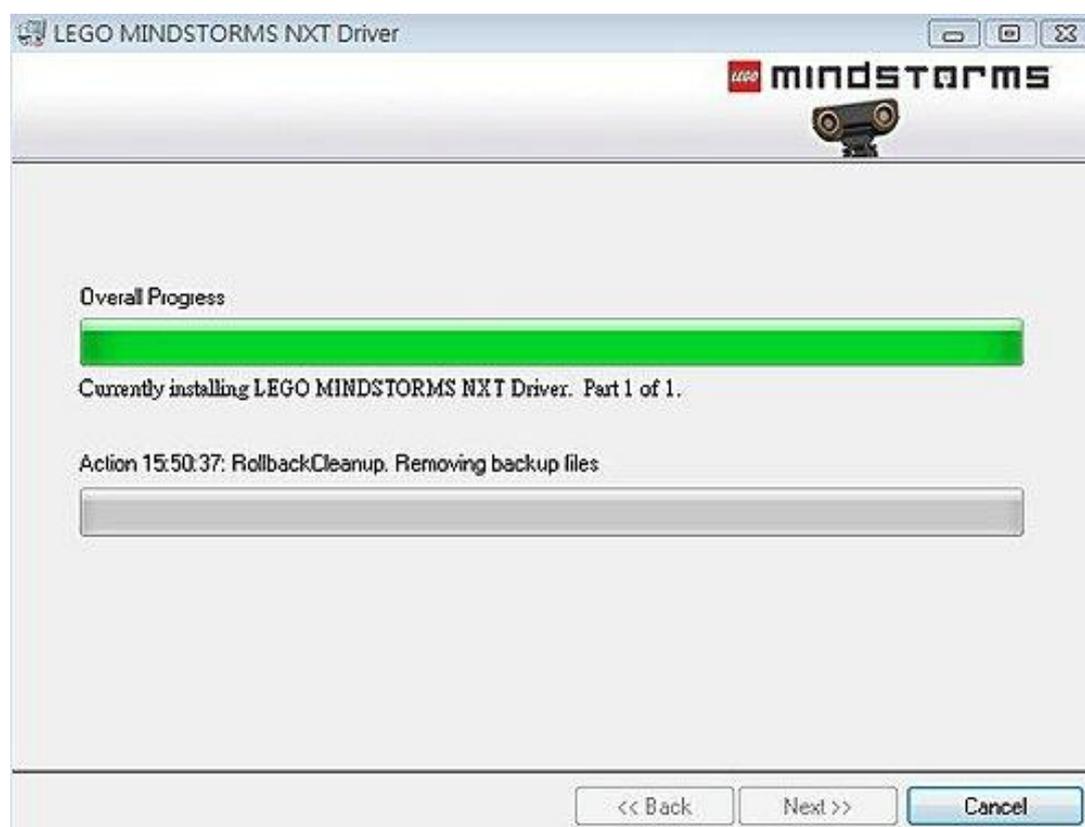


點選左側  Drivers

下載完選擇 Install NXT Driver

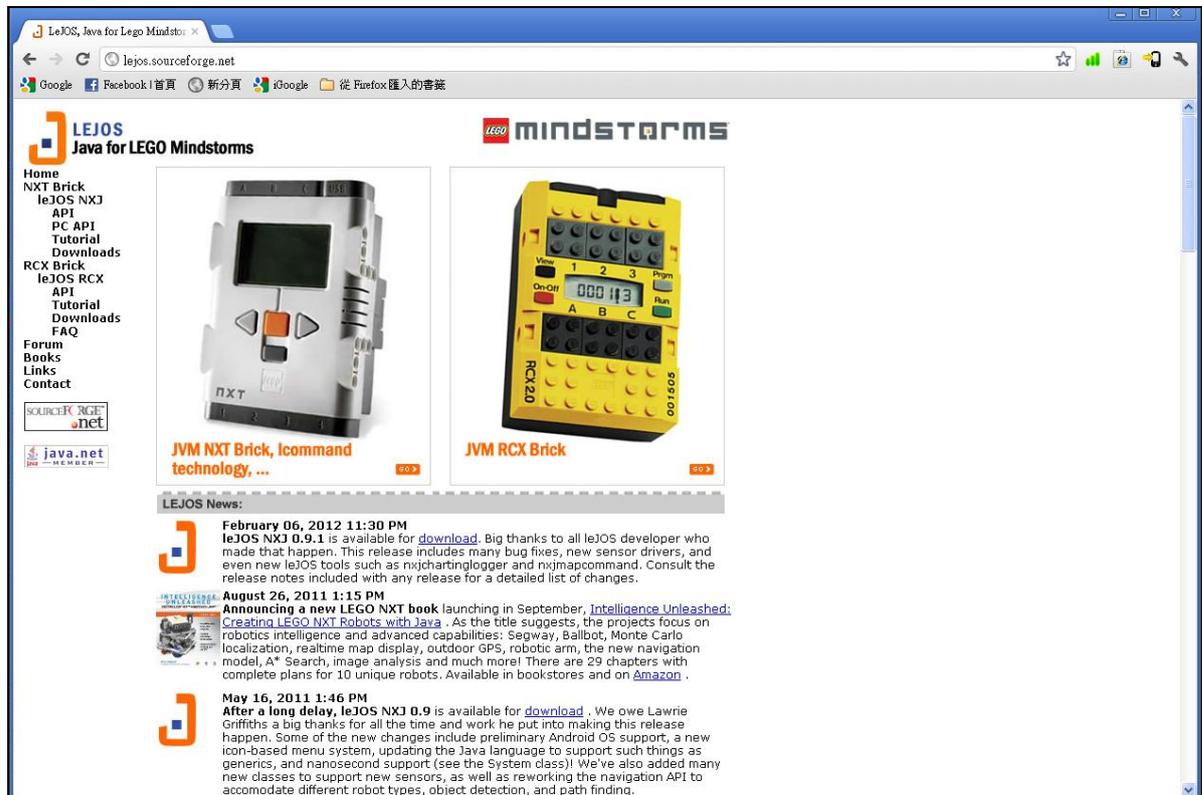


一直選擇 Next

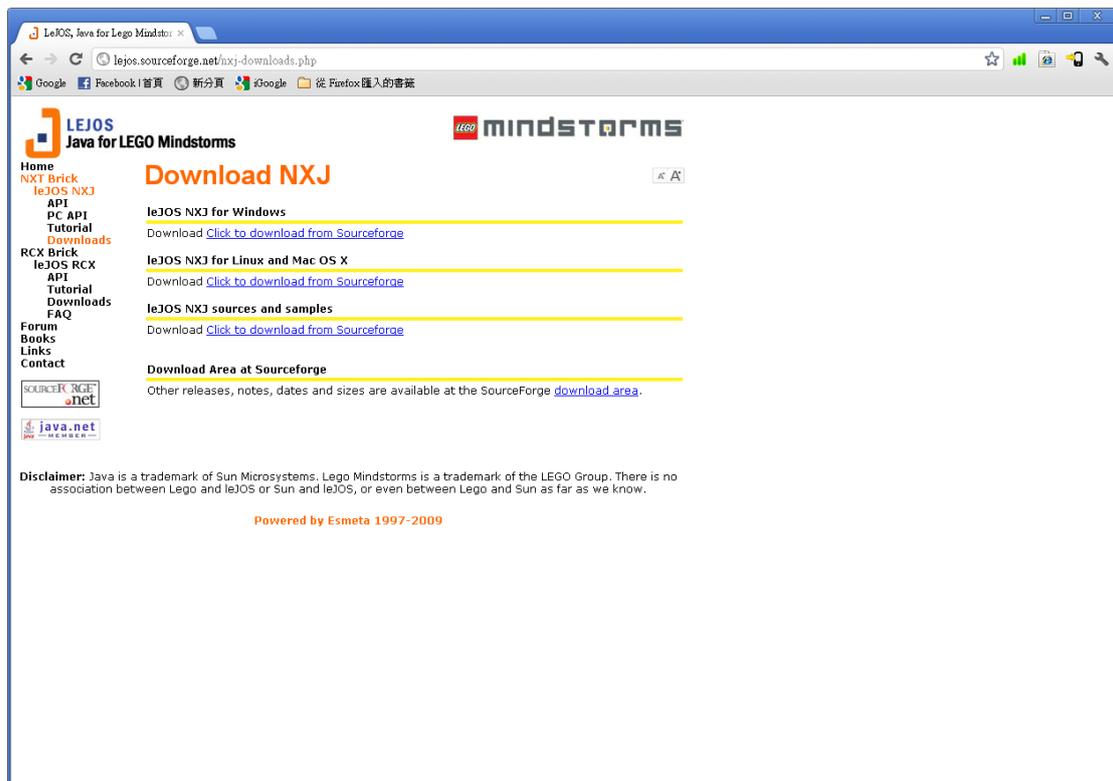


3-4-2 安裝 LeJOS

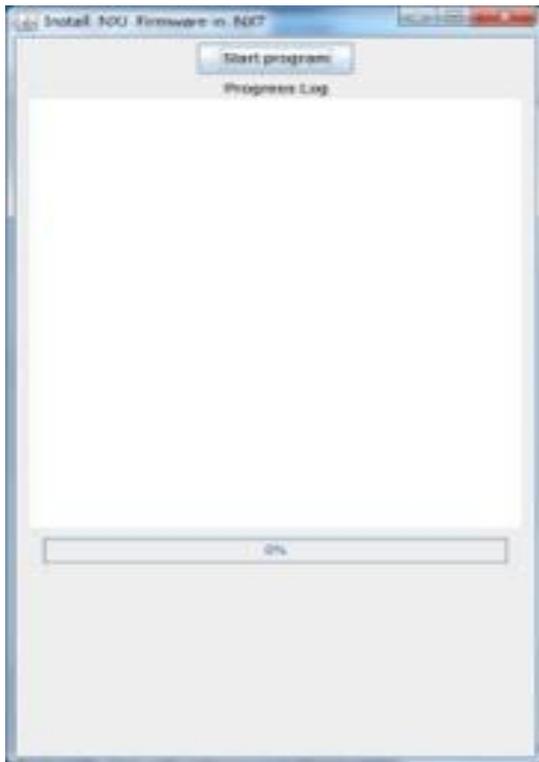
先上 <http://lejos.sourceforge.net/> 選左方的 Downloads



選 LeJOS NXJ for Windows



然後連接 NXT 主機更新軟體



3-5 開發環境

HTC 手機

Windows XP 電腦

JAVA JDK1.6

Eclipse 3.6.2

Android Development Tools



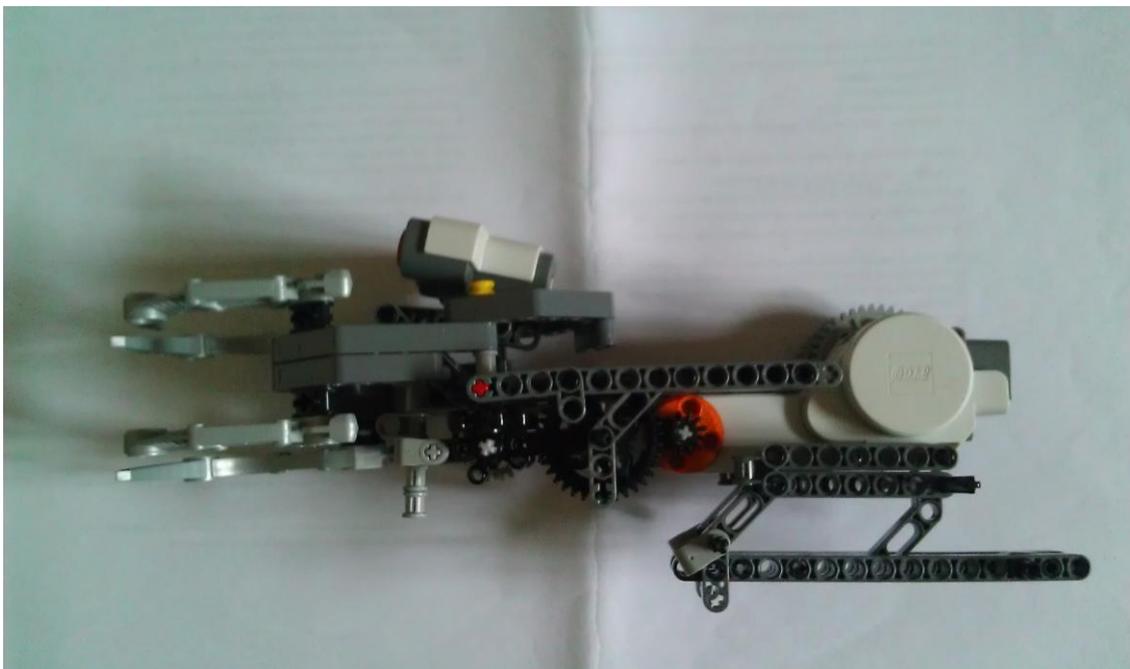
3-6 樂高機器人的設計



組裝成爪子(夾取物品)



加上超音波感應器(偵測距離)



組合完成之剖面圖



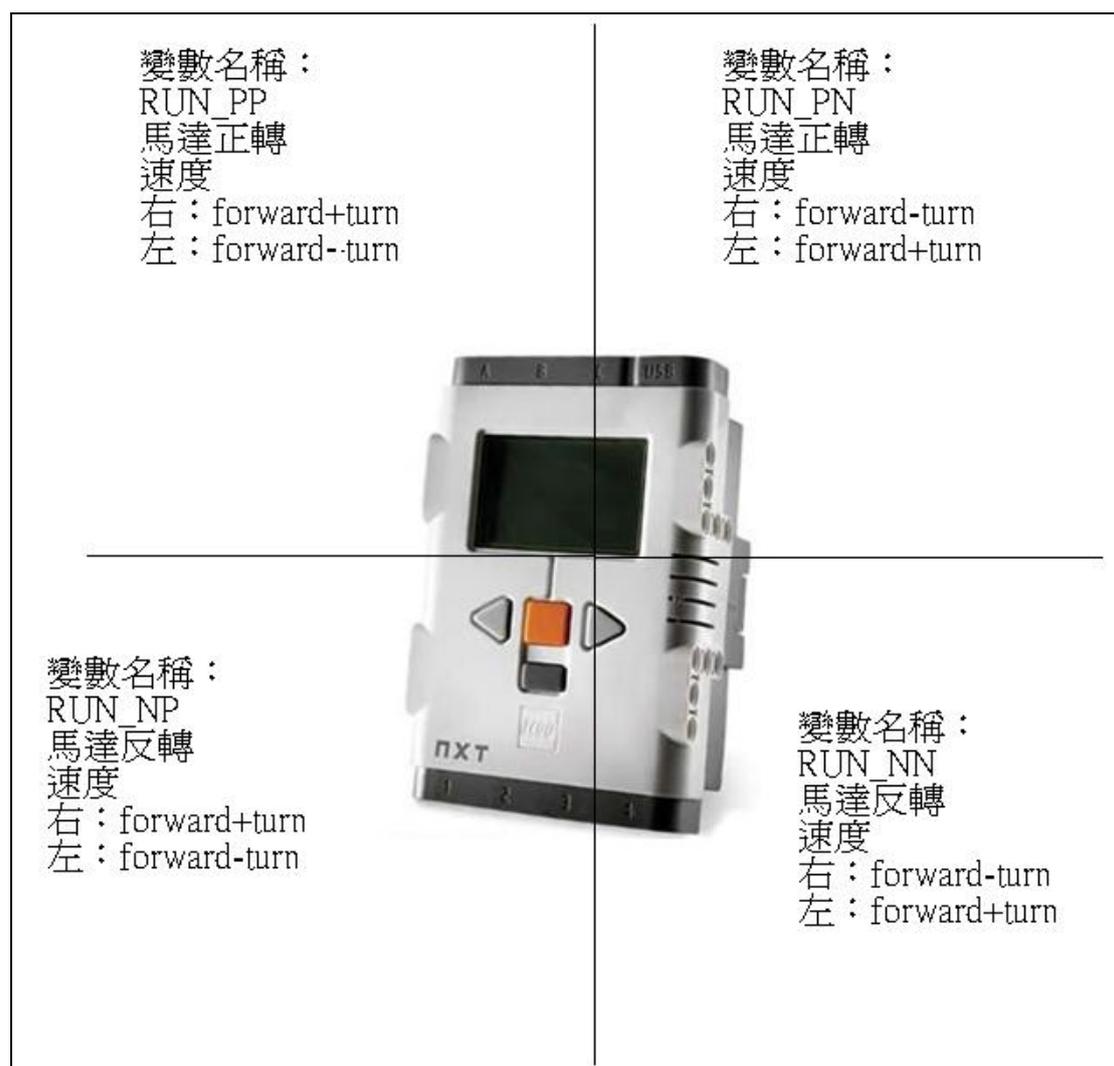
再加上輪子(馬達)及 NXT 主機即完成組裝



3-7 機器人程式的運作

Orientation 感應器：

Orientation 感應器可以讀取以手機為座標中心的三軸轉動角度，也就是 Direction、Pitch 和 Roll。Direction 是以手機正面向上法線(Z 軸)為軸旋轉，Pitch 是以手機中間橫向(Y 軸)為軸旋轉，Roll 是以手機中間直向(X 軸)為軸旋轉。我們在專題只用到 Pitch 和 Roll 的量值，在程式中 Pitch 的值代表 forward 的前進向量量值，Roll 代表的 turn 是轉彎向量量值，我們對於機器人的控制特別定義以 NXT 機器人為座標軸四個運動方向的分量。



我們定義 NXT 機器人的四個方位的指令(command)名稱和馬達速度的調配及轉向，在 NXT 中用 switch...case 就可以實做了，在手機中必須要用到 SensorEventListener 和 SensorManager 兩種物件，SensorManager 可以讓我們使用感應器系統服務、註冊感應器，SensorEventListener 處理感應器數值。

SensorEventListener 說明：

資料	意義	資料範圍
event.value[0]	Direction	0~359
event.value[1]	Pitch	-180~180
event.value[2]	Roll	-90~90

資料傳輸：

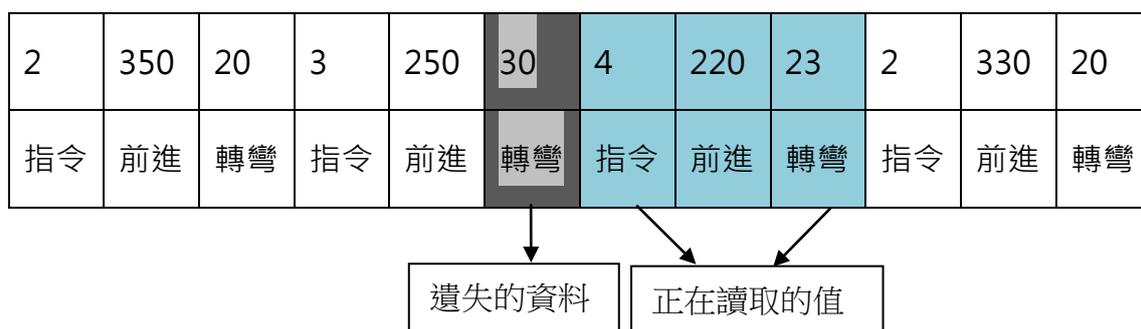
雖然我們使用藍芽對機器人發出控制得指令碼，機器人端只需要部隊的接收指令碼做出對應的動作就可以了，這樣的方式簡單，卻有危險，因為藍芽在傳送一連串的資料時，如果沒一筆獨立的資料間沒有確認機制，很有可能發生誤判情形。

我們對機器人同時發出多個指令：包含指令代碼(command)、前進分量(forward)和轉彎分量(turn)，機器人讀完這三個值才做出一個完整的動作，因此

我們將這三個參數歸類成一組、每一組都是獨立的。

如果每組資料都沒有確認機制，則很有可能因為資料的覆蓋或是遮蔽因而讀取錯誤。

假設我現在送出的資料：

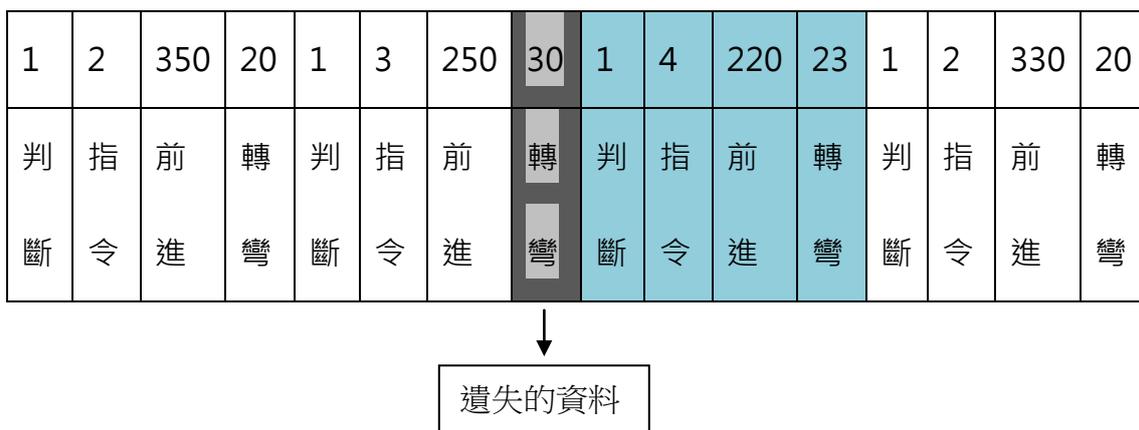


因為轉彎 30 的值在傳輸過程中遺失，真正讀到資料順序為：



可以發現傳輸串流中如果有資料遺失，會造成往後的資料傳輸錯誤。假設我們有定義 1~9 指令，但現在機器人讀到的指令碼為 20 的指令代號，這樣會造成機器人判斷上的錯誤。

但假設我們現在多一個“判斷”參數



設定機器人的邏輯判斷中設定“判斷”參數為 1 時代表該組資料正確，則機器人便可以發現自己讀到的值是錯誤的。

光是會自己發現讀到錯誤的值並不能叫回饋，機器人必須跟手機緊密的連繫才行。最基本的就是機器人一做完一組完整的動作後，他會告訴操控器可以在傳值過來，這樣我們就可以確認接收端(NXT 機器人)有接收到資料。

因此手機每次傳出一筆資料後就必須等待機器人的回應，有了正確的回應就可以再傳下一筆資料了。

資料合併傳輸：

藍芽傳輸中要如何同時傳輸多數資料，而我們卻只用一個判斷參數就可以了，我們使用的方法是將它們合成一筆資料，利用一些數學十進位法處理一下，

將整組資料合成一筆資料，再將該筆資料傳到 NXT 機器人中，機器人只要對該筆資料解密就可以得到全部的參數。

我們建立一個“ order” 變數，手機傳出 order，NXT 只要對 order 變數做除法來取位數就能分解出四個必要的變數。

例如:手機傳出 220531 變數

```
order = command/100000; //220531 除以 100000 取整數 2
```

```
command = command%100000; //餘數 20531
```

```
turn = (command/1000)*10; //20531 除以 1000 取整數等於 20 乘以
```

900/90(900 為馬達轉速最大值、90 為轉動手機最大角度)

```
command = command%1000; //餘數 531
```

```
forward = (command/10)*10; //531 除以 10 取整數乘以 900/90 等於 530
```

```
command = command%10; //餘數 1
```

```
check = command;
```

這樣機器人就能得到所需的四個參數了，中間的 900 除 90 為馬達的轉速比例調配，手機感應器以角度取得為值，最多 90 度，而 NXT 馬達最多 900，所以 $900/90=10$ 的參數就代表手機轉動量對應到馬達轉速的量。

肆、專題分工

工作	人員
報告與記錄	姚長宇
Android 程式設計開發	謝博文
機器人程式設計開發	劉家瑋
機器人組裝與實作	黃奕翔

伍、時程規畫

月份 工作內容	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	3月	4月	5月	6月
搜集資料	■	■	■	■												
學習相關軟體 應用	■	■	■	■												
程式開發		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
機器人設計與 分析				■	■	■	■	■	■	■	■	■				
機器人組裝與 應用				■	■	■	■	■	■	■	■	■	■	■		
實作與整合					■	■	■	■	■	■	■	■	■	■	■	■
測試與偵錯											■	■	■	■	■	■
預定進度累計百 分比	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	70%	80%	90%	100%

陸、問題與解決方法

Q1: android 撰寫不熟悉

Ans: 購買相關書籍以及上網搜尋資料，在多加練習。

Q2: NXT 的程式撰寫

Ans: 嘗試各種開發工具我們從 Microsoft Robotics Developer Studio 2008 到 LEGO MINDSTORMS，最後決定用 LeJOS 開發。

Q3: android 版本以極快得速度更新以及新開發工具不斷出現

Ans: android 系統的更新速度極快，最後我們決定以 android2.3 為主要開發版本，開發工具則是用 Eclipse 開發

Q4: LEGO 機器人需要的功能

Ans: 隨著開發的進度逐漸尋找可以增加的功能

Q5: 藍芽對機器人發出控制得指令碼，藍芽在傳送一連串的資料時，發生誤判情形

Ans: 再每一筆獨立的資料間建立確認機制

Q6: 藍芽傳輸中要如何同時傳輸多數資料

Ans: 將命令利用一些數學十進位法處理一下，將整組資料合成一筆資料，再將該筆資料傳到 NXT 機器人

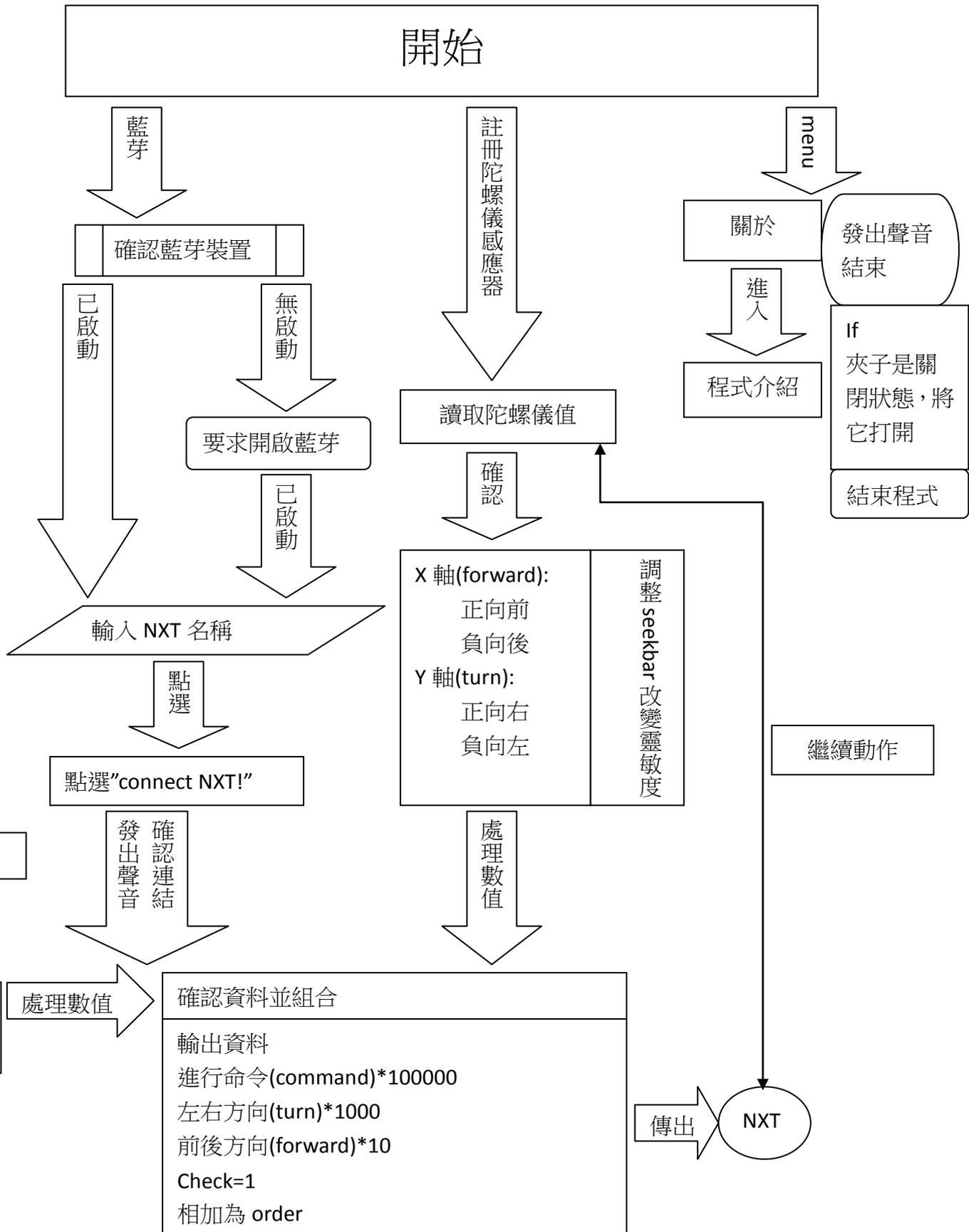
柒、主要成果

手機上的操作流程

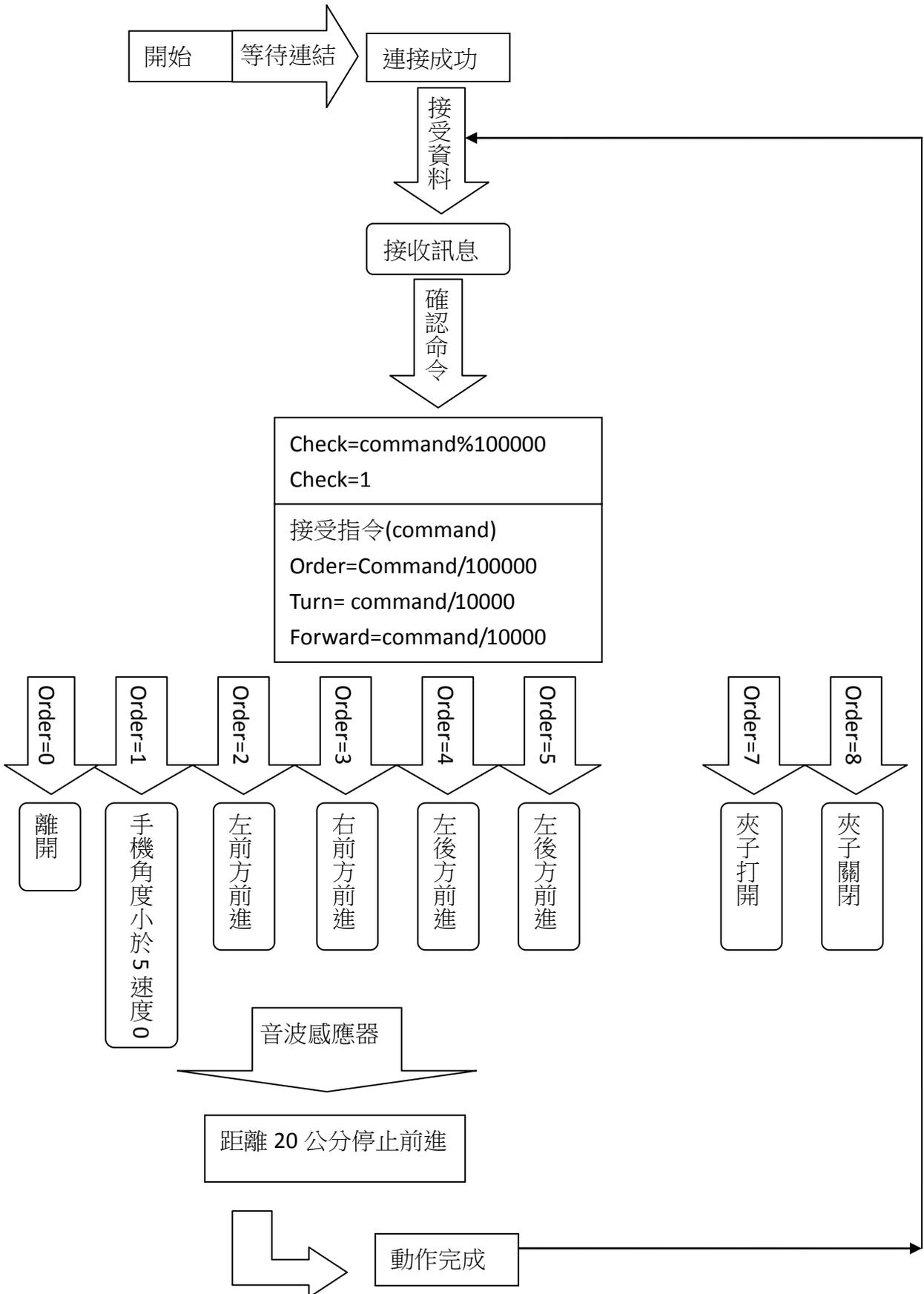


程式流程圖

Android 手機端



NXT 端



程式碼展示(Android 端)

Android 端程式

Main.xml

//Layout 畫面

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/widget0"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#FFFFFF0"
    android:orientation="vertical" >
```

//輸入NXT名稱的欄位，預設NXT

```
<EditText
    android:id="@+id/edtText"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/edit_test"
    android:textSize="18sp"
    android:inputType="none">
</EditText>
```

//啟動藍芽連線的按鈕

```
<Button
    android:id="@+id/btu_connect"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:gravity="center"
    android:text="@string/btu_connect"
    android:textColor="#ff006600"
    android:textStyle="bold" />
```

//確認指令傳遞狀態

```
<TextView
    android:id="@+id/textView_4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp" />
<TextView
    android:id="@+id/textView_3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp" />
<TextView
    android:id="@+id/textView_2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp" />
```

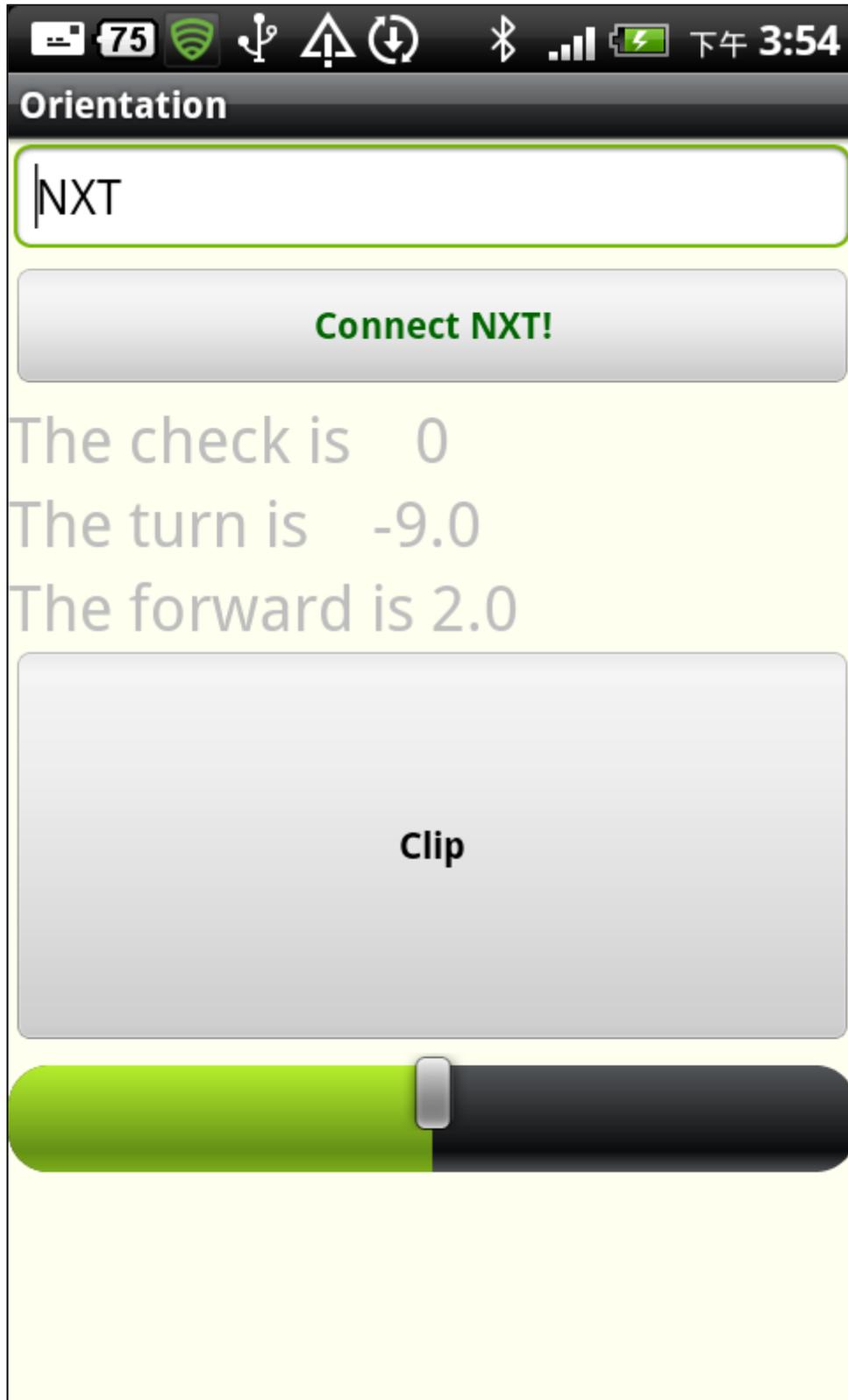
//夾子控制按鈕

```
<Button
    android:id="@+id/clip"
    android:layout_width="fill_parent"
    android:layout_height="154dp"
    android:text="@string/btu_clip"
    android:textStyle="bold" />
```

//NXT馬達轉速控制

```
<SeekBar
    android:id="@+id/stall"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:max="3"
    android:progress="2" />
</LinearLayout>
```

畫面



Activity 機動程式端製作

OrientationMain.java

//註冊各種事件，定義各項變數

```
@Override
```

//按下btu_connect鍵連線藍芽裝置，之後隱藏按鈕

```
btu_connect.setOnClickListener(new Button.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View arg0) {
```

```
        // TODO Auto-generated method stub
```

```
        creatNXTConnect(); //NXT連接
```

```
        btu_connect.setBackgroundColor(0xFFFFFFFF);
```

```
        btu_connect.setTextColor(0xFFFFFFFF);
```

```
    }
```

```
}); //setOnClickListener
```

//控制夾子開關

```
btu_clip.setOnClickListener(new Button.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View arg0) {
```

```
        // TODO Auto-generated method stub
```

```
        if(ckopen==0) {
```

```
            CommandNXT(7*100000+1);
```

```
            ckopen++;
```

```
        } else {
```

```
            CommandNXT(8*100000+1);
```

```
            ckopen--;
```

```
    }
```

```
};
```

```
});
```

//控制手機旋轉敏感度

```
seekbar.setOnSeekBarChangeListener(new  
SeekBar.OnSeekBarChangeListener() {  
    @Override  
    public void onStopTrackingTouch(SeekBar seekBar) {  
        // TODO Auto-generated method stub  
    }  
    @Override  
    public void onStartTrackingTouch(SeekBar seekBar) {  
        // TODO Auto-generated method stub  
    }  
    @Override  
    public void onProgressChanged(SeekBar seekBar, int progress,  
        boolean fromUser) {  
        // TODO Auto-generated method stub  
        @SuppressWarnings("unused")  
        stallnum = seekbar.getProgress();  
    }  
});  
} //OnCreate()結束
```

//離開程式時，若夾子為打開狀態，將其關上

```
    if(ckopen==1)  
        CommandNXT(8*100000+1);  
    finish();  
    break;  
}  
return super.onOptionsItemSelected(item);  
}
```

//傳資料給目標藍芽裝置

```
public void CommandNXT (int ORDER) {  
    if (DATAOu==null) {  
        return;  
    }  
    try {  
        DATAOu.writeInt (ORDER);  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace ();  
    }  
} //CommandNXT
```

//建立藍芽連線，

```
public void creatNXTConnect () {  
    try {  
        BluetoothAdapter BTAdapter =  
BluetoothAdapter.getDefaultAdapter ();  
        if (BTAdapter==null) {  
            Toast.makeText (this, "No Device  
found!", Toast.LENGTH_SHORT).show ();  
            finish ();  
        }  
  
        BluetoothDevice BTDevice = null;  
        Set<BluetoothDevice> BTList = BTAdapter.getBondedDevices ();  
        if (BTList.size ()>0) {  
            for (BluetoothDevice TempoDevice : BTList) {  
  
if (TempoDevice.getName ().equals (editText.getText ().toString ())) {  
                BTDevice = TempoDevice;  
            }  
        }  
    }  
}
```

```

    }
}
}
    BTSocket =
BTDevice.createRfcommSocketToServiceRecord(UUID.fromString("00001101-
0000-1000-8000-00805f9b34fb"));
    BTSocket.connect();
}
    //Link of successful
    String successTalk = "connect success";
    tts.speak(successTalk, TextToSpeech.QUEUE_ADD, null);
}
    DATAOu = new DataOutputStream(BTSocket.getOutputStream());
    CommandNXT(START*100000);
    mThread = new readThread();
    mThread.start();
} catch (IOException e) {
    Toast.makeText(this, "Wrong", Toast.LENGTH_LONG).show();
}
} //creatNXTConnect

```

//複寫感應器監聽事件

```

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {}
    // TODO Auto-generated method stub

@Override
    //對感應器的讀值做處理
    public void onSensorChanged(SensorEvent event) {
        // TODO Auto-generated method stub
        textView_2.setText("The forward is "+event.values[1]);
        textView_3.setText("The turn is "+event.values[2]);
        textView_4.setText("The check is "+count);
    }
}

```

//調整手機轉動向量，當手機讀值小於5度的時候前進 (forward) 和轉彎 (turn) 的量值為0。
//讀取感應器event的資料成員value[1]和value[2]就能得到轉動角度，再用TextView顯

//示在螢幕上做確認，count為0代表手機傳出，1代表從NXT傳出。

```
        turn = (Math.abs(event.values[2])>5 ? Math.abs((int)
(event.values[2])):0)*stallnum;
        forward = (Math.abs(event.values[1])>5 ?
Math.abs((int) (event.values[1])):0)*stallnum;
        if (BTSocket != null && count == 0) {
            command = ((event.values[1]>0)?
((event.values[2]>0)?RUN_PP : RUN_PN):((event.values[2]>0)? RUN_NP :
RUN_NN));
            order = command*100000+turn*1000+forward*10+1;
            CommandNXT (order);
            count=1;
        }
    } //onSensorChanged
}
```

//建立輸入串流，中間建立無限迴圈不斷處理NXT傳出的數值，check為1，
//Count變為1，單純的循環。

//開啟新的執行緒，屬於巢狀類別以便共用Activity的資料成員

```
class readThread extends Thread{
    int check=0,readData;
    public void run() {
        try {
            DATAIn = new
DataInputStream (BTSocket.getInputStream());
            while (true) {
                if (count == 1) {
                    check = DATAIn.readInt();
                    count = (check == 1)?0:1;
                    check=0;
                }
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

AndroidManifest.xml

//設置手機不會轉動

```
android:screenOrientation="portrait"
```

```
>
```

程式碼展示(NXT 端)

NXT 端程式

Orien2.java

//載入 lejos.nxt 類別、lejos.nxt.comm 類別來處理通訊、java.io 類別是用來處理串
//流。

//程式開始

//類別名稱 Orien2 必須跟檔案相同

//定義各項參數

//order 控制 NXT 機器人動作

//command 讀取來自 Android 手機的命令

//turn 轉彎數值

//forward 前進數值

//check 確認是否執行參數

//distance 讀取 NXT 音波感應器數值

//ckopen 夾子開闔確認

```
int order,command,turn,forward,check,distance,speed=0, ckopen=1;
```

//NXT 螢幕顯示連結中

```
System.out.println("Connecting");
```

//等待用戶端連線
//服務端一開始先等待客戶端發起連線要求，並設定連線方式為藍芽。使用
//waitForConnection(int timeout,int mode)方法可以建立與客戶端的連線，但必須
//傳入 2 個參數，time 為等待時間(ms)，如果設 0 則程式會一直等待客戶端連線，
//連線成功才往下執行。

```
BTConnection BTC = Bluetooth.waitForConnection(0,  
NXTConnection.RAW);
```

//連線成功後發出 BEEP 聲，在螢幕上顯示 Connect success!，定義 datain 從藍芽
//接收資料，定義 dos 發出資料。

```
Sound.beep();  
System.out.println("Connect success!");  
UltrasonicSensor l1 = new UltrasonicSensor(SensorPort.S1);  
DataInputStream datain = BTC.openDataInputStream();  
DataOutputStream dos = BTC.openDataOutputStream();  
try{  
    while(true){  
        //car control
```

//Command 收取資料，我們對 order 資料做除法來取位數就能分解出需要的變
//數，在令 distance 處理感應器變數，計算對 turn 和 forward 影響。

```
command = datain.readInt();  
distance = l1.getDistance();
```

//在 Android 手機內，利用一些數學十進位法處理一下，將整組資料合成一筆資
//料，再將該筆資料傳到 NXT 機器人中，機器人只要對該筆資料解密就可以得到
//全部的參數

```
order = command/100000;  
command = command%100000;  
turn = (command/1000)*10;  
command = command%1000;  
forward = (command/10)*10;
```

```
command = command%10;
```

```
check = command;
```

//判對 order 指令，控制夾子狀態，讓音波感應器可以做判斷

```
if(order==7){
```

```
ckopen=0;
```

```
}else if(order==8){
```

```
ckopen=1;
```

```
}
```

//根據感應器數值讓 NXT 有安全防護

//距離小於 20 公分會停止讓機器人前進，其餘不影響。

```
if( distance<=20 && order==2 && ckopen==1){
```

```
order=1;
```

```
}
```

```
else if(distance<=20 && order==3 && ckopen==1){
```

```
order=1;
```

```
}
```

```
else if(distance<=20 && ckopen==1)
```

```
{
```

```
if(order==2 || order==3) order=1;
```

```
}
```

//判讀指令，check 為 1 則執行 switch 內的各個 case 指令，讓機器人有不同動作。

//若感應器有反應，讓馬達減速。

```
if(check==1){
```

```
switch(order){
```

```
case 0:
```

```
Motor.B.stop();
```

```
Motor.C.stop();
```

```
System.exit(1);
```

```
break;
```

```
case 1:
```

```

        Motor.B.setSpeed(0);
        Motor.C.setSpeed(0);
        break;
//左前方前進
        case 2: //pp
            Motor.B.setSpeed(forward+turn);
            Motor.C.setSpeed(forward-turn);
            Motor.B.forward();
            Motor.C.forward();
            break;
//右前方前進
        case 3: //pn
            Motor.B.setSpeed(forward-turn);
            Motor.C.setSpeed(forward+turn);
            Motor.B.forward();
            Motor.C.forward();
            break;
//左後方前進
        case 4: //np
            Motor.B.setSpeed(forward+turn);
            Motor.C.setSpeed(forward-turn);
            Motor.B.backward();
            Motor.C.backward();
            break;
//右後方前進
        case 5: //nn
            Motor.B.setSpeed(forward-turn);
            Motor.C.setSpeed(forward+turn);
            Motor.B.backward();
            Motor.C.backward();
            break;
//夾子關閉
        case 7: //ClipClose
            Motor.A.setSpeed(300);
            Motor.A.forward();
            Delay.msDelay(830);
            Motor.A.stop();
            break;

```

//夾子打開

```
        case 8: //ClipOpen
            Motor.A.setSpeed(300);
            Motor.A.backward();
            Delay.msDelay(830);
            Motor.A.stop();
            break;
        default:
            System.out.println("No command");
            break;
    }
}
```

//NXT 回傳指令為整數 1 告知 android 可以在發送指令

```
        dos.writeInt(1);
        dos.flush();
        check=0;
    } //while
    } catch(IOException e){
        System.out.println("Wrong connection during
command"+e.toString());
        Sound.buzz();
        Delay.msDelay(4000);
    }
}
}
```

捌、結語

以上所說的功能 iPhone 當然也能勝任，因為控制原理是完全相同的。但由

於 Android 有其低進入門檻 (很多人會 Java) 以及高低價位機海戰術，甚至還有適合無程式開發的 Google App Inventor 搶佔初學者族群，加上 iPhone 程式必須在蘋果電腦上開發否則無法上架販售，故此相當看好 Android 在教學領域上的應用。

玖、銘謝

感謝陳建宏老師的指導，幫助我們對於專題的方向提供意見跟規劃，也向學校幫我們申請實體手機，讓我們在開發上更為方便，以及每一位組員的配合和協助，使此專題得以完成。

拾、參考文獻

Android2 手機應用程式設計入門 第 3 版

Android/NXT 機器人大戰-智慧型手機控制機器人

參考網站

 <http://developer.android.com/index.html>

LEGO MindStorms NXT

<http://www.pspwp.pwp.blueyonder.co.uk/science/robotics/nxt/index.html>

JWorld@TW

<http://www.javaworld.com.tw/jute/post/view?bid=26&id=275260>

nxtOSEK/JSP

<http://lejos-osek.sourceforge.net/>