微電腦介面電路設計

專題報告I

組別 : D組

班級 : 電機3B

姓名 : 林啟源

學號 : 103501027

1. **主題：**App藍芽遙控車

* 軟體使用 : Arduino、Android Studio
* 硬體使用 : Arino duino/GenuUno、Arduino Shield、伺服馬達 X 2、光敏電阻X 1、

LED燈 X 4、電阻 X 4、藍芽模組(HC-06) X 1、蜂鳴器 X 1、

紅外線感、測模組 X 1

* 功能 :

1. 手機透過藍芽控制arduino(物聯網應用)，使車輪轉動。
2. 光感電阻偵測光線，當光線不足時自動開啟LED燈。
3. 紅外線感測距離，當後方距離太接近時會自動停止後退，並啟動蜂鳴器警示。
4. 轉彎時會自動打方向燈，可避免駕駛忘記打方向燈。
5. **程式碼 + 程式說明：**
6. **Arduino code**

#include<Servo.h> //引用Servo Libary

#include <SoftwareSerial.h> //引用藍芽相關 Libary

Servo servoRight; //定義伺服馬達右邊輪子

Servo servoLeft; //定義伺服馬達左邊輪子

SoftwareSerial BT(8, 9); //接收腳(RX), 傳送腳(TX)

char val; //設定字元，用於接收資料的變數

int PinLed0=10; //設定前LED腳位

int PinLed3=7; //設定左後LED腳位

int PinLed4=6; //設定右後LED腳位

int photoresisterPin=5; //設定光敏電阻腳位

int light\_on; //光敏電阻控制LED

int light\_press=1; //開關LED

int rLed=2; //設定紅外線腳位

int BeepPin=4; //設定蜂鳴器腳位

void setup()

{

servoRight.attach(12); //右輪連結servo的12腳

servoLeft.attach(13); //左輪連結servo的13腳

Serial.begin(9600); //設定鮑率為9600

BT.begin(9600); //設定藍牙模組的連線速率

pinMode(PinLed0,OUTPUT); //設定前LED輸出腳位

pinMode(PinLed3,OUTPUT); //設定左後LED輸出腳位

pinMode(PinLed4,OUTPUT); //設定右後LED輸出腳位

pinMode(photoresisterPin,INPUT); //設定光敏電阻

pinMode(2,INPUT); //設定redLED

pinMode(4,OUTPUT); //設定蜂鳴器

tone(BeepPin,3000,1000); //tone(腳位,頻率,時間)

}

void loop()

{

//LED

light\_on = digitalRead(photoresisterPin); //由光敏電阻感測是否開燈

if (light\_press==1)

{

if(light\_on==0) //前方LED燈開

digitalWrite(PinLed0,HIGH);

else if(light\_on==1) //前方LED燈關

digitalWrite(PinLed0,LOW);

}

else

digitalWrite(PinLed0,HIGH);

//紅外線感應

int irLed = digitalRead(rLed); //紅外線感應偵測到物體會輸出0，

Serial.println(irLed); //未偵測到物體會輸出1

delay(100);

//蜂鳴器

if (irLed==0) //當紅外線感應器輸出值為0時，代表接近物體，

{ //因此使蜂鳴器發出聲音

tone(BeepPin,3000,500);

bestop();

}

//偵測到藍芽

if (BT.available())

{

val = BT.read(); //讀取藍芽的值

Serial.println(val); //顯示藍芽的值

//控制車子方向

if(val=='B') //前進

{

Serial.println("Forward"); //在模擬時顯示出"Forward"字串

forward();

}

else if(val=='F') //後退

{

Serial.println("Backward"); //在模擬時顯示出"Backward"字串

backWard();

}

else if(val=='L') //左前轉

{

Serial.println("Left"); //在模擬時顯示出"Left"字串

turnLeft();

digitalWrite(PinLed3,HIGH); //使左後LED亮

digitalWrite(PinLed4,LOW); //使右後LED暗

}

else if(val=='R') //右前轉

{

Serial.println("Right"); //在模擬時顯示出"Right"字串

turnRight();

digitalWrite(PinLed3,LOW); //使左後LED暗

digitalWrite(PinLed4,HIGH); //使右後LED亮

}

else if(val=='I') //原地右轉

{

Serial.println("RightUp"); //在模擬時顯示出"RightUp"字串

turnRightUp();

digitalWrite(PinLed3,LOW); //使左後LED暗

digitalWrite(PinLed4,HIGH); //使右後LED亮

}

else if(val=='G') //原地左轉

{

Serial.println("LeftUp"); //在模擬時顯示出"LeftUp"字串

turnLeftUp();

digitalWrite(PinLed3,HIGH); //使左後LED亮

digitalWrite(PinLed4,LOW); //使右後LED暗

}

else if(val=='J') //右後轉

{

Serial.println("RightDown"); //在模擬時顯示出"RightDown"字串

turnRightDown();

}

else if(val=='H') //左後轉

{

Serial.println("LeftDown"); //在模擬時顯示出"LeftDown"字串

turnLeftDown();

}

else //停止

{

bestop();

digitalWrite(PinLed3,LOW);

digitalWrite(PinLed4,LOW);

}

}

}

//紅外線

int irDetect(int irLedPin,int irReceiverPin,long frequency)

{

tone(irLedPin,frequency,8);

delay(10);

int ir=digitalRead(irReceiverPin);

delay(10);

return ir;

}

//方向 透過給予不同頻率使輪子轉動方向、速度不同

int forward(){ //前進

servoRight.writeMicroseconds(1700);

servoLeft.writeMicroseconds(1300);

}

int turnLeft(){ //左

servoRight.writeMicroseconds(1300);

servoLeft.writeMicroseconds(1300);

}

int turnLeftUp(){ //左上

servoRight.writeMicroseconds(1500);

servoLeft.writeMicroseconds(1300);

}

int turnLeftDown(){ //左下

servoRight.writeMicroseconds(1500);

servoLeft.writeMicroseconds(1700);

}

int turnRight(){ //右

servoRight.writeMicroseconds(1700);

servoLeft.writeMicroseconds(1700);

}

int turnRightUp(){ //右上

servoRight.writeMicroseconds(1700);

servoLeft.writeMicroseconds(1500);

}

int turnRightDown(){ //右下

servoRight.writeMicroseconds(1300);

servoLeft.writeMicroseconds(1500);

}

int backWard(){ //後退

servoRight.writeMicroseconds(1300);

servoLeft.writeMicroseconds(1700);

}

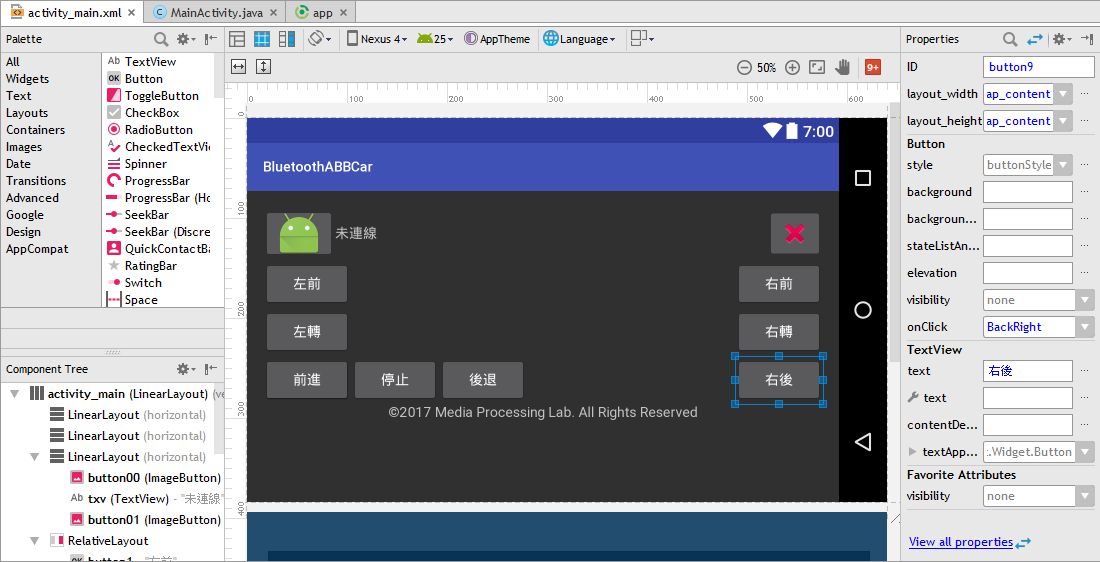
int bestop(){ //停止

servoRight.writeMicroseconds(1500);

servoLeft.writeMicroseconds(1500);

}

1. **App code**
2. **Xml code**

****

1. **Java code**

package com.j204\_1.han.bluetoothabbcar;

import android.content.pm.ActivityInfo;

import android.os.Bundle;

import android.os.Message;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.TextView;

import tw.com.flag.api.FlagBt;

import tw.com.flag.api.FlagTank;

import tw.com.flag.api.OnFlagMsgListener;

public class MainActivity extends AppCompatActivity implements OnFlagMsgListener {

FlagBt bt;

TextView txv;

FlagTank tank;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

this.setRequestedOrientation( //讓手機螢幕保持橫向模式

ActivityInfo.SCREEN\_ORIENTATION\_LANDSCAPE);

txv = (TextView)findViewById(R.id.txv);

bt = new FlagBt(this); // 建立藍牙物件

tank=new FlagTank(bt); // 建立Tank物件

}

public void onDestroy() {

bt.stop(); // 確保程式結束前會停止藍牙連線

super.onDestroy();

}

public void connect(View v) {

if(!bt.connect()) // 選取已配對裝置進行連線

txv.setText("找不到任何已配對裝置");

}

public void quit(View v) {

bt.stop();

finish();

}

@Override

public void onFlagMsg(Message msg) {

switch (msg.what) {

case FlagBt.CONNECTING: // 嘗試與已配對裝置連線

txv.setText("正在連線到：" + bt.getDeviceName());

break;

case FlagBt.CONNECTED: // 與已配對裝置連線成功

txv.setText("已連線到：" + bt.getDeviceName());

break;

case FlagBt.CONNECT\_FAIL: // 連線失敗

txv.setText("連線失敗！請重連");

break;

case FlagBt.CONNECT\_LOST: // 目前連線意外中斷

txv.setText("連線中斷!請重連");

break;

}

}

// 定義藍芽資料傳送

public void ForwardLeft(View v){bt.write("G");}

public void Forward(View v){bt.write("F");}

public void ForwardRight(View v){bt.write("I");}

public void Left(View v){bt.write("L");}

public void Stop(View v){bt.write("S");}

public void Right(View v){bt.write("R");}

public void BackLeft(View v){bt.write("H");}

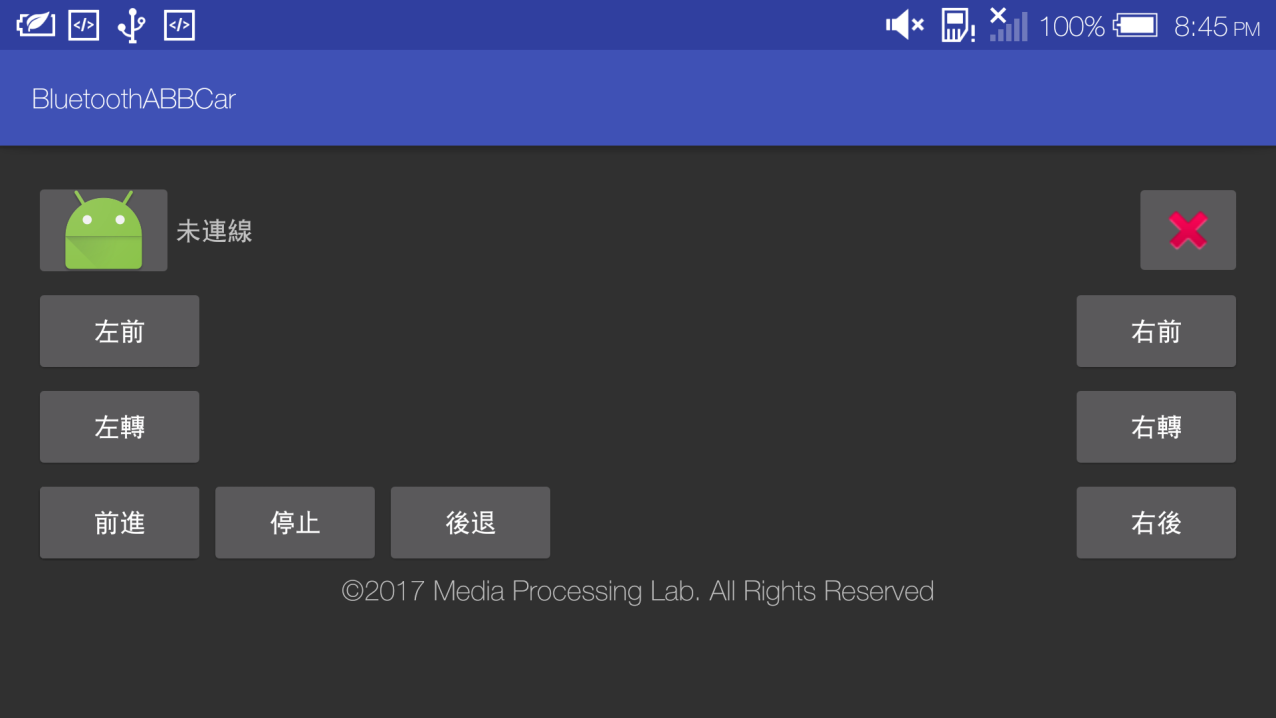
public void Back(View v){bt.write("B");}

public void BackRight(View v){bt.write("J");}

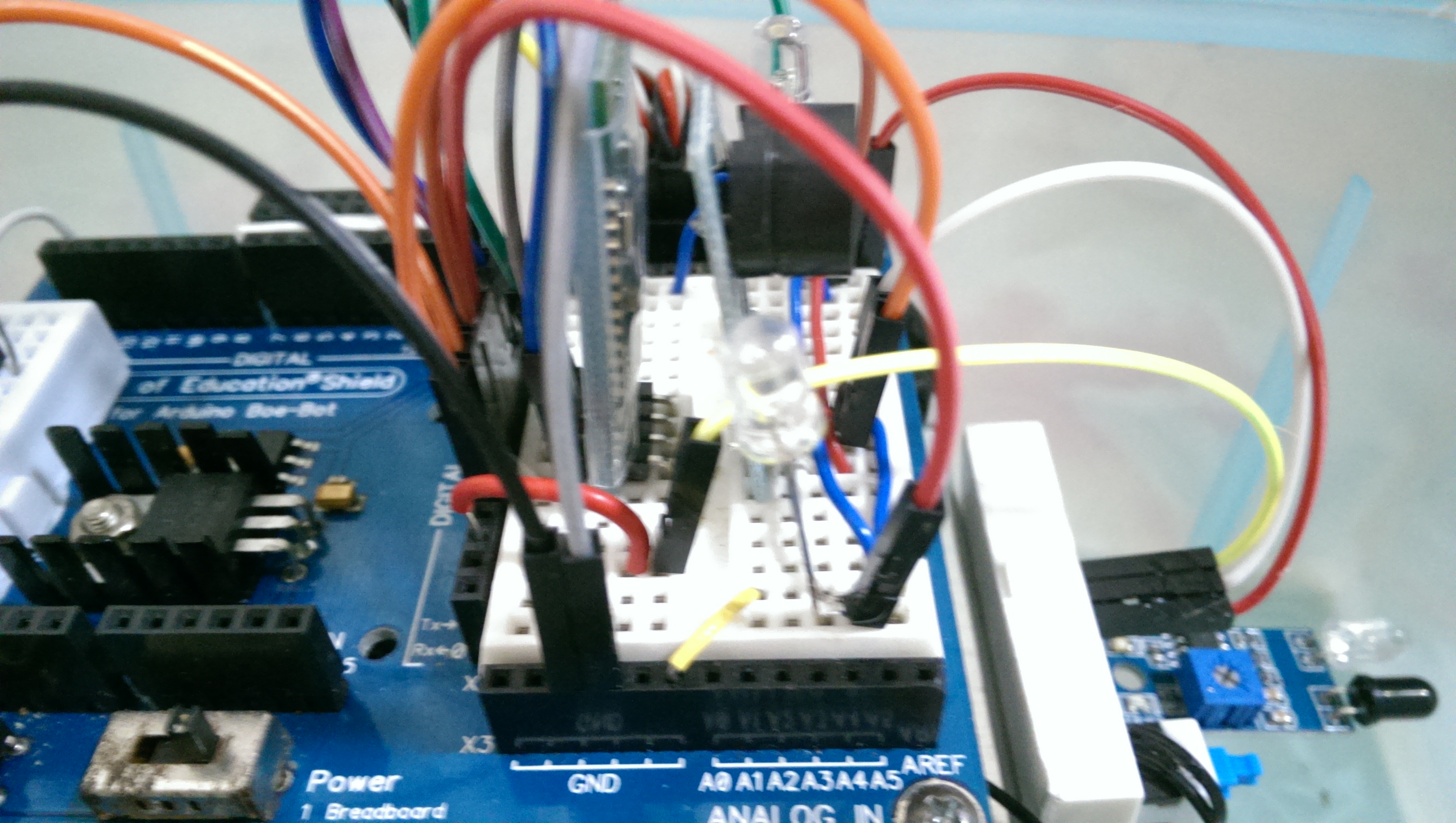
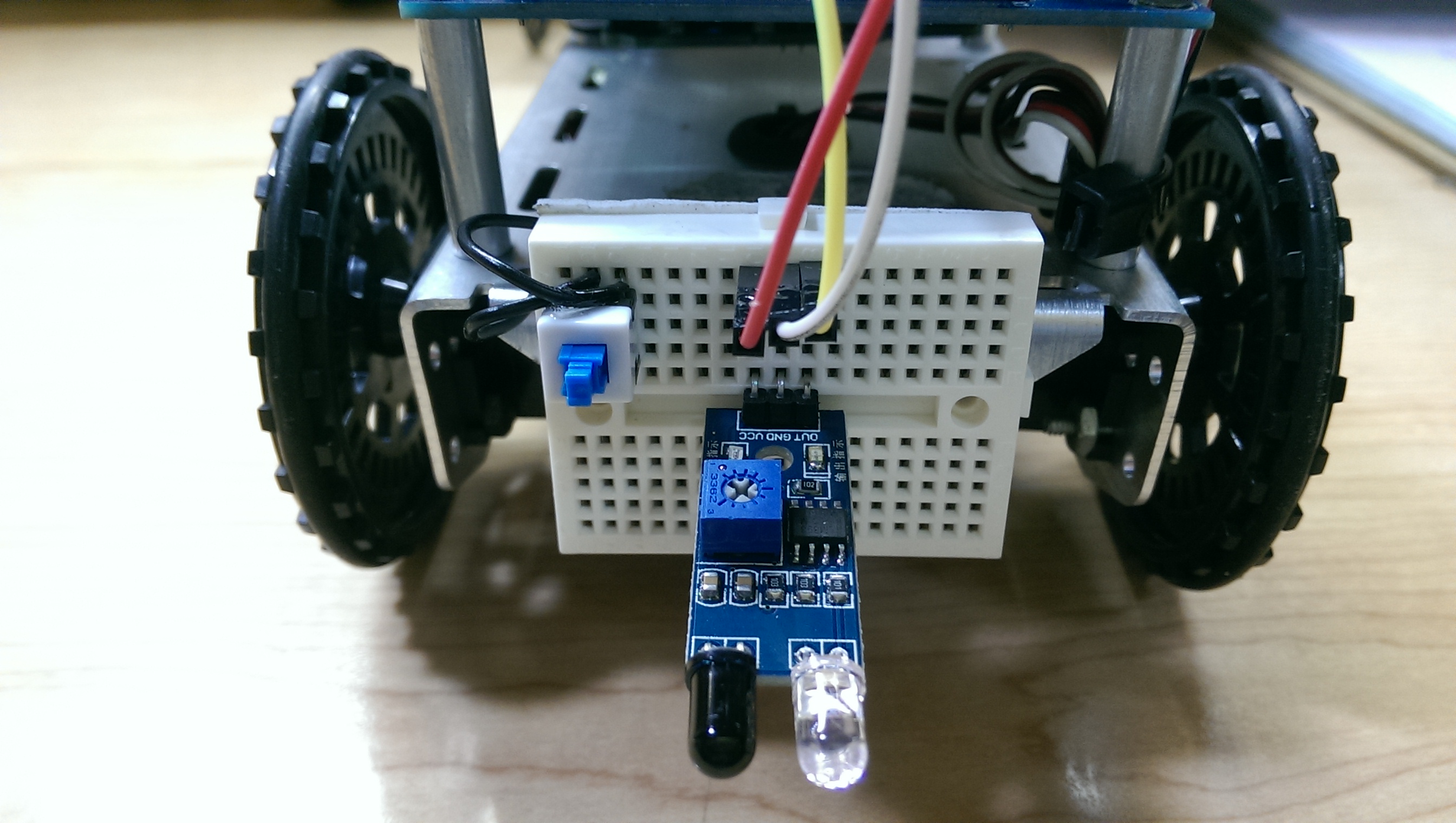
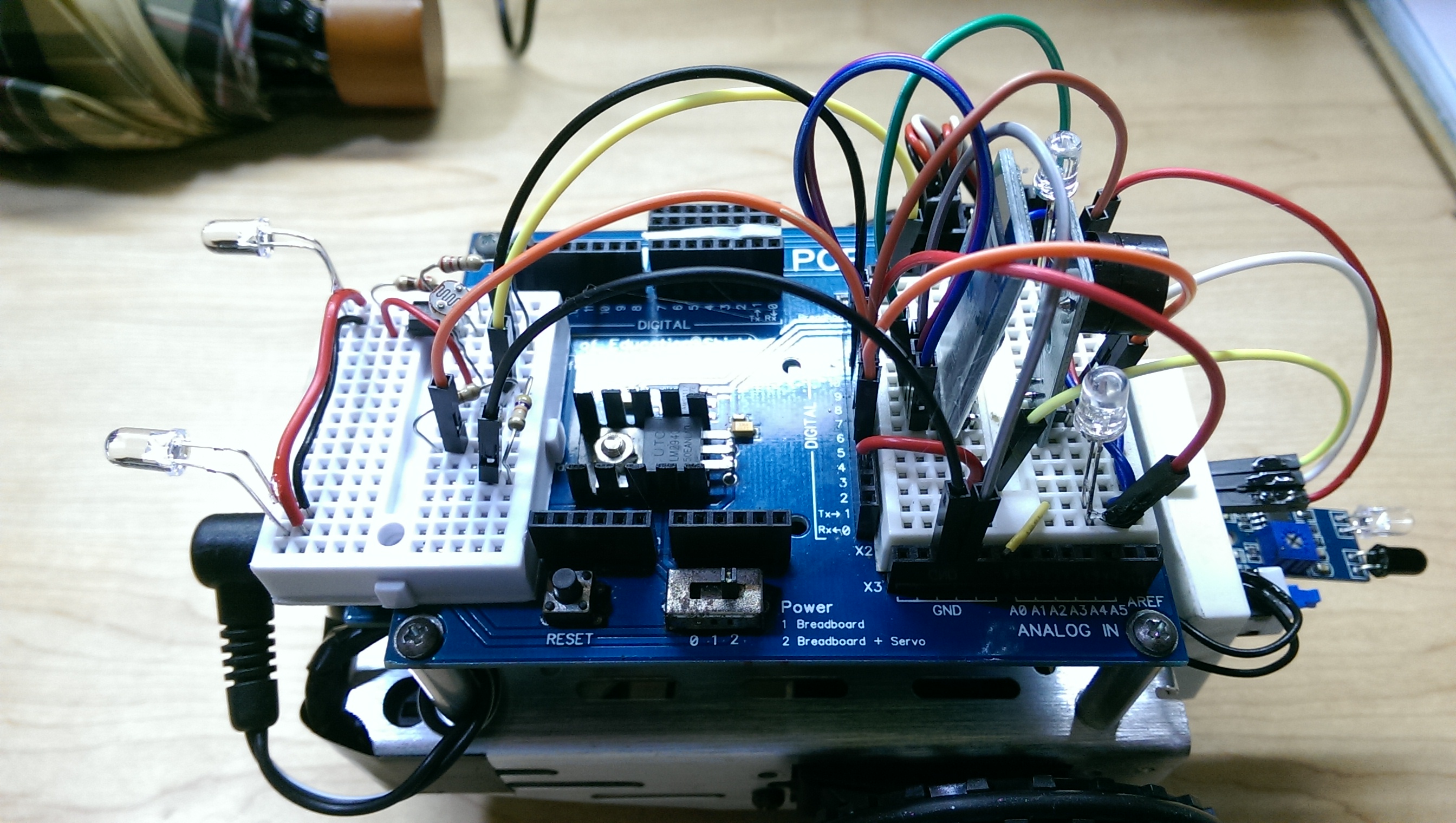
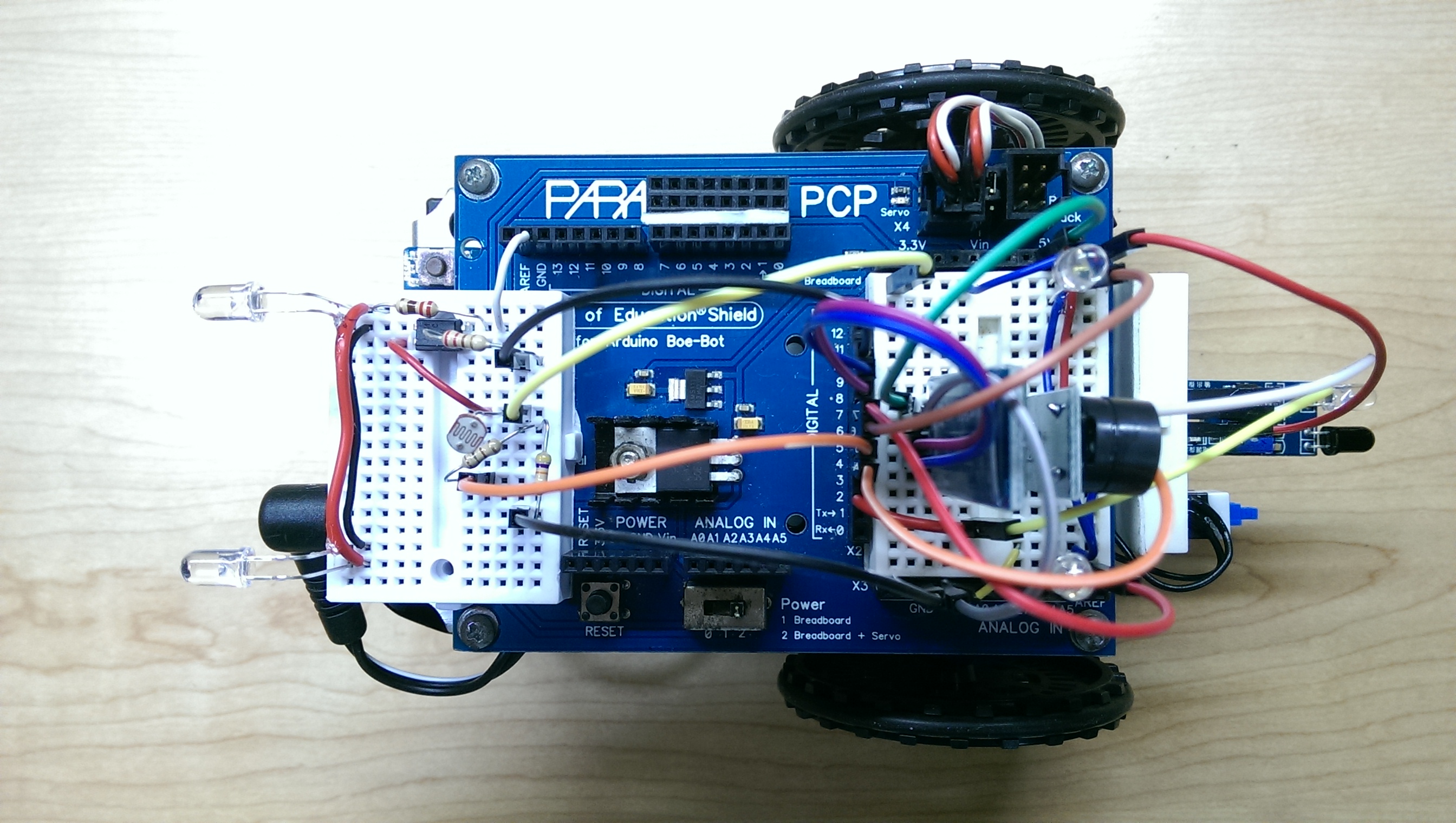
}

1. **介面：**

手機app截圖 -

****

實作照片 -



1. **運作說明：**

首先啟動電源開關🡺藍芽模組在未連線時燈會閃爍🡺與手機連線🡺啟動電源後若燈光不足，車子前端的LED會自動亮起(燈光的控制是由光敏電阻自動開關)🡺使用app控制車子

🡺若車子轉彎，會自動閃爍方向燈，提醒後方來車。

🡺若車子後方有障礙物且車子還未停止，會啟動蜂鳴器警示並使車子停止後退。



藍芽遙控車流程圖(available判斷是否有按按鈕)

1. **心得：**

由這次的專題作業讓我學習了arduino這個語言及app的製作，arduino是目前很熱門的程式語言，且較C、C++簡易，因此在學習方面速度較快。另外透過藍芽模組嚷我們能使用手機與arduino連線，這也是近來熱門的物聯網應用之一。App的製作讓我學習到簡單的app其內容可分為操作介面及內部控制程式，手機的普及使得越來越多app的出現，透過學習android studio讓我了解了一些app的內容設計及寫法。

經由課堂上助教所教的一些元件使用方式，讓我了解到有哪些元件可以增加到我的遙控車上，使作品的豐富度增加不少。我們將可以成果的功能擴充並將其應用在日常生活中。目前新出的汽車多配備倒車雷達、倒車顯影及自動點燈系統，這些功能可以讓我們的生活更加安全，因此我們能夠從簡易的arduino做起，研發更多設計能應用在生活上。