目录

**[一． 综述 3](#_Toc18169)**

**[二．APP总览 4](#_Toc8760)**

**[三．功能介绍及核心代码解释 5](#_Toc5159)**

[1.蓝牙总开关 5](#_Toc22728)

[2. 蓝牙搜索 5](#_Toc2884)

[3. 蓝牙连接或断开 6](#_Toc17314)

[4. 任务4-摄像头图像传输 7](#_Toc30787)

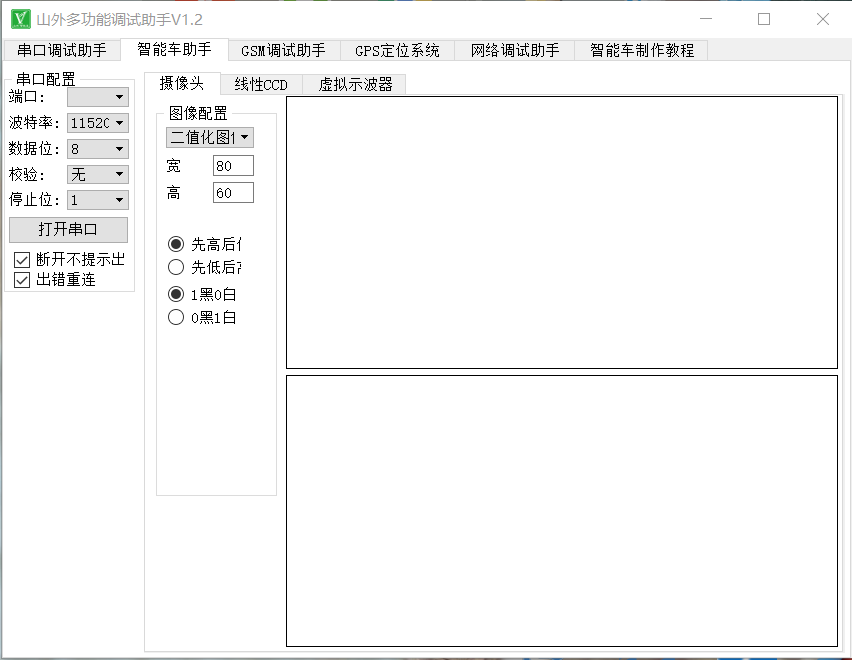
[5. 其余任务 7](#_Toc6558)

**[四． 结语 8](#_Toc11908)**

# 综述

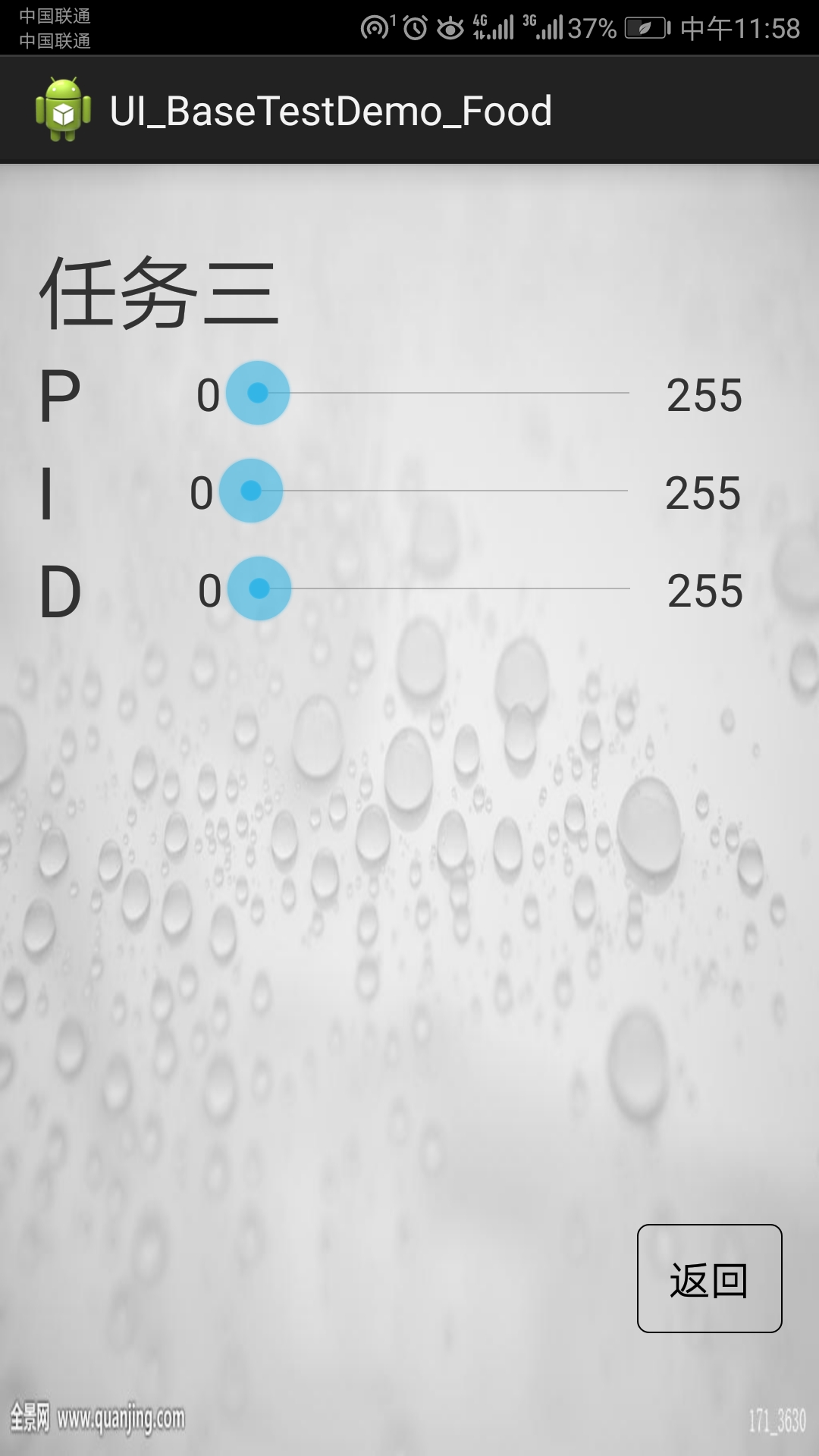
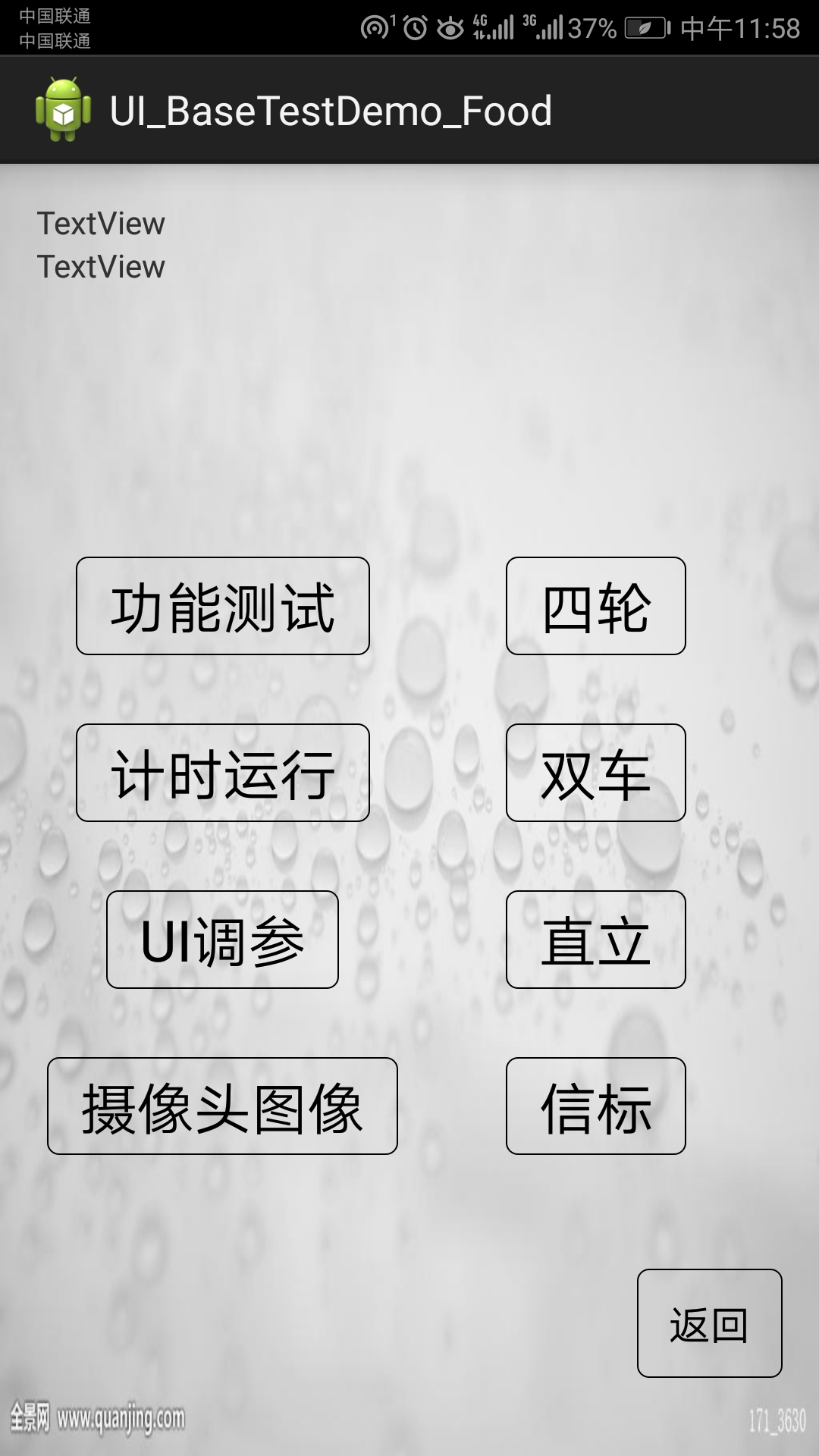
全国大学生“飞思卡尔”杯智能汽车竞赛起源于韩国，是韩国汉阳大学汽车控制实验室在飞思卡尔半导体公司资助下举办的以HCS12单片机为核心的大学生课外科技竞赛。组委会提供一个标准的汽车模型、直流电机和可充电式电池，参赛队伍要制作一个能够自主识别路径的智能车，在专门设计的跑道上自动识别道路行驶，最快跑完全程而没有冲出跑道并且技术报告评分较高为获胜者。其设计内容涵盖了控制、模式识别、传感技术、汽车电子、电气、计算机、机械、能源等多个学科的知识，对学生的知识融合和实践动手能力的培养，具有良好的推动作用。

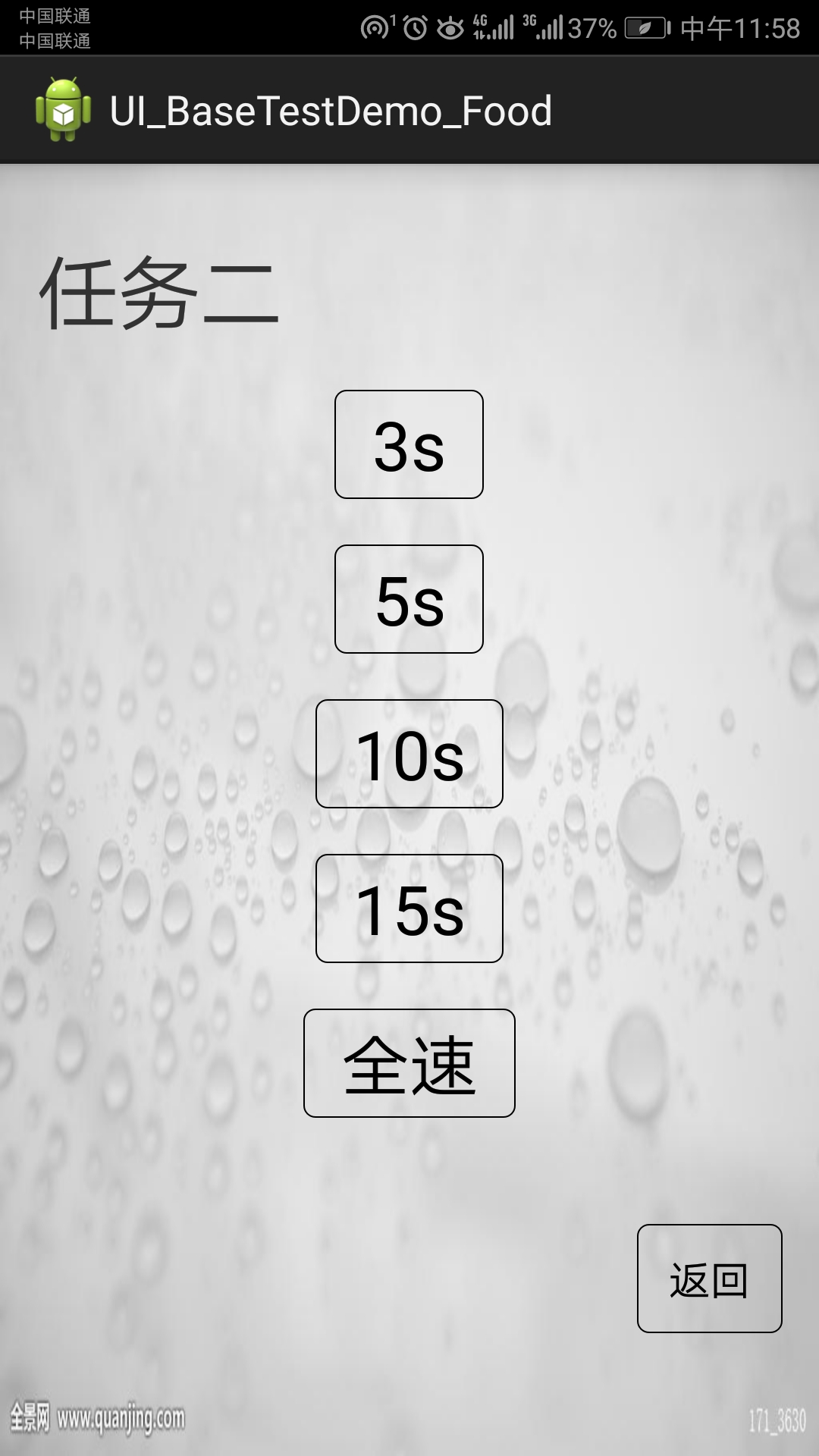
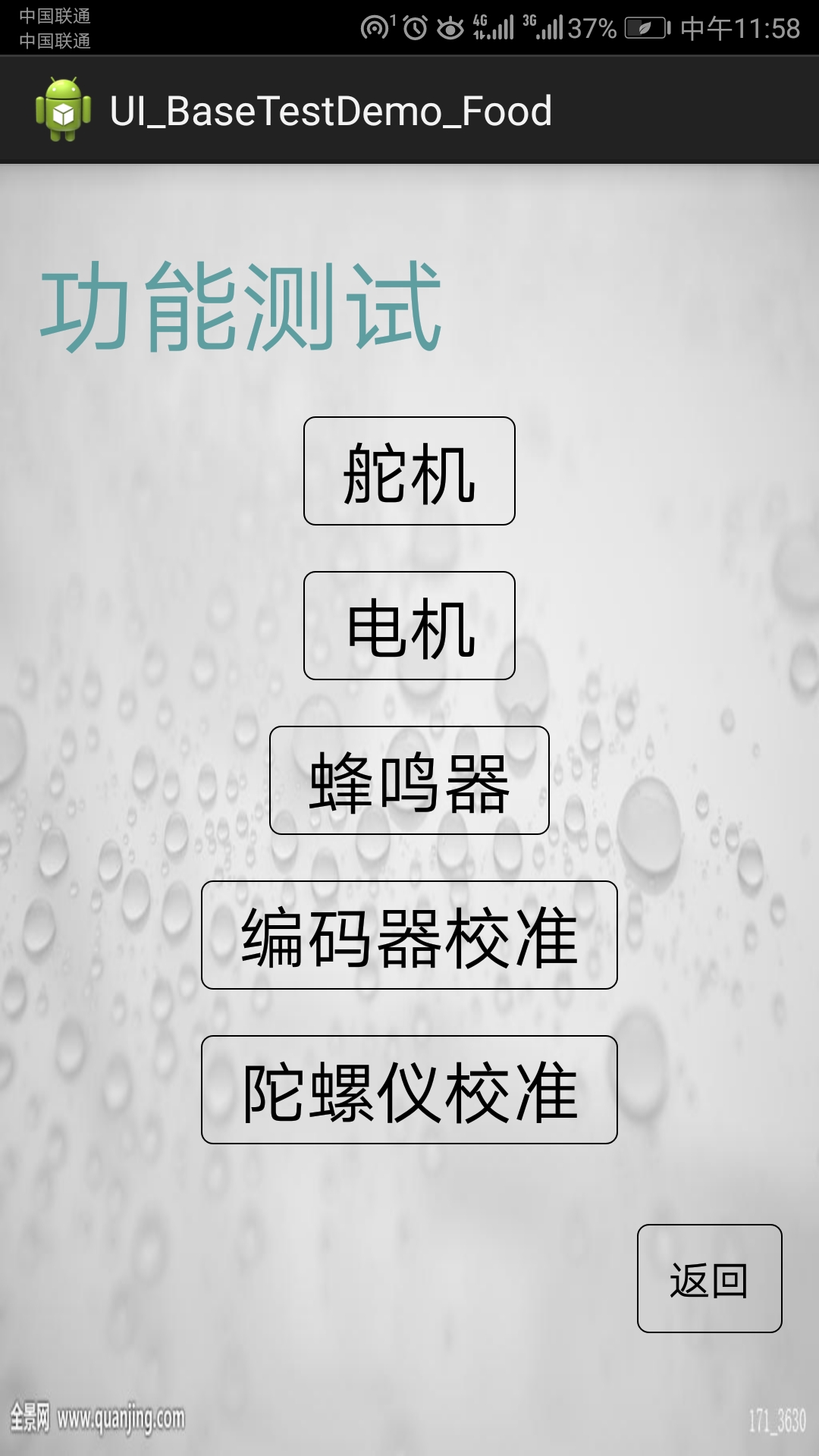
在智能车比赛中常常会用到各种单片机，以前是NXP公司，以后是英飞凌，STC等公司的单片机。跟单片机打交道会经常调试。因为智能车的单片机是脱手型，即使用时是运动状态。所以常常需要一些无线通信的上位机来与之通信，比较常见的是电脑的上位机如图。这种上位机功能较多，一般是公司型产品。



而根据上课所学java知识，并参考一些软件开源，便有了写一个手机智能车调试app的想法。有些功能还未实现如接收，有些原理不一定能通如蓝牙传输的带宽是否满足图像等。尽管如此还是先将一部分想法变为了现实。

# 二．APP总览





# 三．功能介绍及核心代码解释

## 1.蓝牙总开关

这是一个开关CompoundButton监听器，即检测蓝牙，若开就开，若没开就打开。

**if** (b) {

**if** (!bluetoothAdapter.isEnabled()) { //蓝牙未开启，则开启蓝牙

Intent enableIntent = **new** Intent(BluetoothAdapter.***ACTION\_REQUEST\_ENABLE***);

startActivity(enableIntent);

} **else** {

Toast.*makeText*(MainActivity.**this**, "蓝牙已开启", Toast.***LENGTH\_SHORT***).show();

}

} **else** {

bluetoothAdapter.disable();

Toast.*makeText*(MainActivity.**this**, "蓝牙已关闭", Toast.***LENGTH\_SHORT***).show();

}

## 蓝牙搜索

这是个View组件，点击搜索将已配对的蓝牙设备添加到列表中。

view.performHapticFeedback(HapticFeedbackConstants.***LONG\_PRESS***,HapticFeedbackConstants.***FLAG\_IGNORE\_GLOBAL\_SETTING***);//震动

**if** (bluetoothAdapter == **null**) {

Toast.*makeText*(MainActivity.**this**, "未发现蓝牙设备", Toast.***LENGTH\_SHORT***).show();

**return**;

} **else** **if** (!bluetoothAdapter.isEnabled()) {

Toast.*makeText*(MainActivity.**this**, "蓝牙设备未开启", Toast.***LENGTH\_SHORT***).show();

}

Set<BluetoothDevice> pairedDevices = bluetoothAdapter.getBondedDevices();

**if** (pairedDevices.size() > 0) {

**for** (BluetoothDevice device : pairedDevices) {

adapter.remove(device.getName());

adapter.add(device.getName());

}

} **else** {

//注册，当一个设备被发现时调用mReceive

Log.*i*("seach", "hhhhhh");

IntentFilter filter = **new** IntentFilter(BluetoothDevice.***ACTION\_FOUND***);

registerReceiver(mReceiver, filter);

}

## 蓝牙连接或断开

搜索设备在View监听器中点击设备，建立连接，建立成功后直接发送广播

**if** (strMacAddress == **null**) {

Toast.*makeText*(MainActivity.**this**, "请先搜索设备", Toast.***LENGTH\_SHORT***).show();

} **else** {

**if** (*booleanConnect* == **false**) {

Intent i = **new** Intent(MainActivity.**this**, MyService.**class**);

i.putExtra("Mac", strMacAddress);

startService(i);

bluetoothConnect.setEnabled(**false**);

}

**else** // 断开蓝牙

{

*booleanConnect* = **false**;

//stopService(new Intent(MainActivity.this, MyService.class));

bluetoothConnect.setText("连接");

Intent intent = **new** Intent();//创建Intent对象

intent.setAction("android.intent.action.cmd");

intent.putExtra("cmd", ***CMD\_STOP\_SERVICE***);

sendBroadcast(intent);//发送广播连接蓝牙

}

}

## 任务4-摄像头图像传输

这是一个比较难的设计，至今我还未完善好。具体的思路就是建立一个view的监听器，并当切换到任务4时，将蓝牙接收到的摄像头数据编码后显示到到view控件中。其中难点有两个：

一：蓝牙带宽，据我估算若此方法可行，显示帧率大约只有5帧左右。

二：编码技术，将收到的RGB565数据逐行逐列的编码，十分困难。

由于以上两个原因，此功能暂时不可行。

## 其余任务

由于其余任务不涉及到接收数据，基本模式相同如下：

1.创建返回按钮监听器

break1.setOnClickListener(**new** View.OnClickListener() {//创建事件监听器

@Override

**public** **void** onClick(View v) {

v.performHapticFeedback(HapticFeedbackConstants.***LONG\_PRESS***,HapticFeedbackConstants.***FLAG\_IGNORE\_GLOBAL\_SETTING***);

String s1="$0";

String s2="#";

String s=s1+s2;

SendBlueToothProtocol(s);

finish();

}

});

2.创建任务监听器，蓝牙广播

button\_steer.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View v) {

// **TODO** Auto-generated method stub

v.performHapticFeedback(HapticFeedbackConstants.***LONG\_PRESS***,HapticFeedbackConstants.***FLAG\_IGNORE\_GLOBAL\_SETTING***);

SendBlueToothProtocol("$1,1#");

}

});

**private** **void** SendBlueToothProtocol(String value ) {

Intent intent = **new** Intent();//创建Intent对象

intent.setAction("android.intent.action.cmd");

intent.putExtra("cmd", ***CMD\_SEND\_DATA***);

intent.putExtra("command", (**byte**)0x00);

intent.putExtra("value", value);

sendBroadcast(intent);//发送广播

}

}

# 结语

由于java水平，有限所做作品的水平亦有限，且借鉴了不少开源软件的地方。希望以后有机会进行完善。