

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

课程论文

THESIS OF CURRICULUM



工程实践与科技创新III-C:
基于安卓平台的小车控制系统

组 长: 高若彤 517030910148

组 员: 刘箬凡 517030910202

组 员: 程玉婷 517030910123

指导教师: 张士文

摘要: 随着互联网时代电子计算机和相关软件的发展, 安卓手机提供了一个很好的平台[1], 不仅可以令人们获得诸多便捷的服务, 还可以实现许多基于安卓手机的应用开发。智能小车的自动控制应用的实现便是其中之一, 所谓智能小车, 就是电子计算机等最新科技成果与现代汽车工业相结合的产物, 它兼具多种功能, 通常可以实现自主控制等操作。本项目中实现了基于安卓手机中所提供的多种功能对智能小车进行前进、后退、左转、右转的控制, 主要通过编译开发 Android APP 来达到控制小车的目的。其中软件采用 Android Studio 进行编写, 硬件部分则采用了 Energia 平台进行编译和烧写程序到单片机上。总体而言, 本项目功能多、用途广、架构清晰、效果良好, 最终很好地实现了基于蓝牙通信和安卓平台的智能小车控制驾驶。

关键词: 安卓平台; 手机控制; 视频回传; 应用开发; 智能小车; 程序烧写; 单片机

Abstract: With the development of the Internet era computer and related software, Android phones provides a good platform[1], not only can make people get a lot of convenient services, we can achieve a lot of application development Android phones based. One of them is the realization of the automatic control application of the smart car. The so-called smart car is the product of the latest technological achievements such as electronic computers and the modern automobile industry. It has multiple functions and can usually achieve autonomous control and other operations. In this project, the smart car is controlled based on various functions provided in the Android mobile phone to control forward, backward, left, and right turns. The purpose of controlling the car is mainly through compiling and developing the Android APP. The software is written in Android Studio, and the hardware is compiled and programmed on the microcontroller using the Energia platform. In general, this project has many functions, wide uses, clear architecture, and good results. In the end, the smart car control driving based on Bluetooth communication and Android Studio is well realized.

Key Words: Android Studio ; Mobile Phone Control ; Video Return ; Application Development ; Smart Car ; Programming ; Microcontroller.

1 项目介绍/Introduction

1.1 项目意义/Significance

安卓手机自 2008 年发行以来，其基于 Linux 的操作系统就在手机市场的世面上占有一席之地，历久弥新，与此同时，基于安卓手机的平台应用开发层出不穷，接连不断。而本项目主要实现的便是基于 Android Studio 的手机端 APP 开发，该应用主要用途是：通过调用手机的各种功能和数据库以期实现对智能小车的多种功能的操作控制，达到小车前进、后退、左转和右转的效果。本项目最终实现的主要功能如表 1 所示：

表 1 本项目实现的功能

工作任务	详细描述
蓝牙通信	与小车进行蓝牙的连接和通信
视频回传	两部手机实现 WiFi Socket 视频回传
按键控制	通过简单的按键控制小车
手势识别	通过识别手势控制小车运动
重力感应	重力感应控制小车运动
语音控制	通过语音口令控制小车运动

经过一个学期的努力与尝试，在本小组成员不断的学习与积累的过程中，最终成功开发出控制智能小车运动的手机端 APP 并完成了后续的测试与评估。其中软件部分分为小车端 APP 开发和手持端 APP 的开发，实现了双机的交互，控制小车行进。更进一步的，小车端应用主要是充当摄像头的角色，将小车的前方路况传回手持端，实现视频回传的功能；而手持端应用则用于处理发送至手机上的指令，并传输相应的单片机控制字符给小车，来达到控制小车的目的。控制小车的方式有多种多样，包括简单的前进后退左转右转的按键控制、手势识别、重力感应和语音控制。另一方面，硬件部分则是使用 Energia 编写代码并且将程序直接烧写在单片机的 MSP430 的板子上。

1.2 项目框架/Project Structure

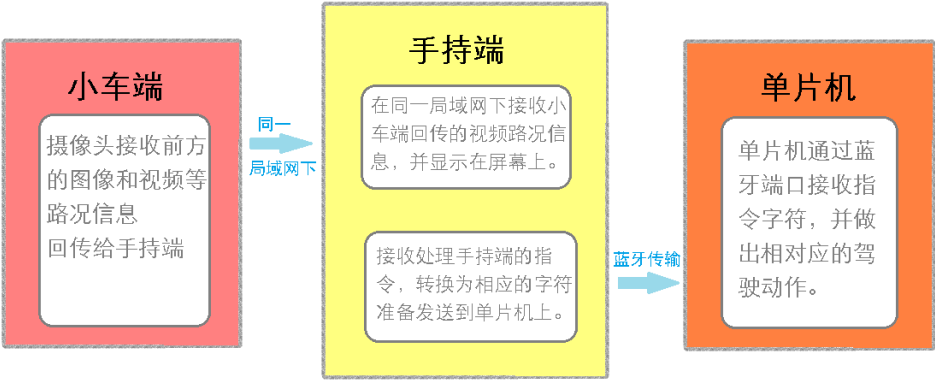


图 1 项目整体框架结构

本项目的整体框架如上图 1 所示。其中单片机部分的代码程序是使用 Energia 直接编译并烧写在板子上，主要实现的是通过蓝牙端口接收指令字符，并作出相应的前进、后退、左转和右转的驾驶动作。而软件部分的代码程序则分为小车端和手持端，小车端主要实现了摄像头接收正前方的图像和视频等路况信息，而后回传给手持端的功能；手持端则主要实现了在同一局域网下接收小车回传的视频路况信息，并显示在手机屏幕上，以及接收处理手持端的诸多不同类型的指令，转换为相应的字符通过蓝牙发送到单片机上。

2 项目内容与方法/Contents and Methods

2.1 单片机

单片机部分的代码编写相对来说较为简单，由于本项目中使用的小车单片机是 **MSP430**，蓝牙模块是 **HC-06**，使用的用于编译和烧写程序的是 **Energia**，而 **Energia** 是 **TI** 公司出品的、封装度极高的 **MSP430** 的开发环境，拥有非常丰富的库函数，同时可以让代码呈现惊人的简洁，无论写代码还是读代码，都相当直观且清晰[2]。因此本小组在编译和烧写单片机的时候选择了 **Energia** 作为开发环境。如图 2 所示是本项目所使用的单片机板子是 **MSP430 g2553**，从图上可以清晰地看出该板子有一个 **USB** 接口，用于和电脑连线进行程序的烧写，一些端口，用于连接蓝牙或其他调试设备，两个 **LED** 灯，用于显示该板子是否在正常工作等，以及该单片机的主体部分……图 3 所示则是该板子的系统电路连接图。

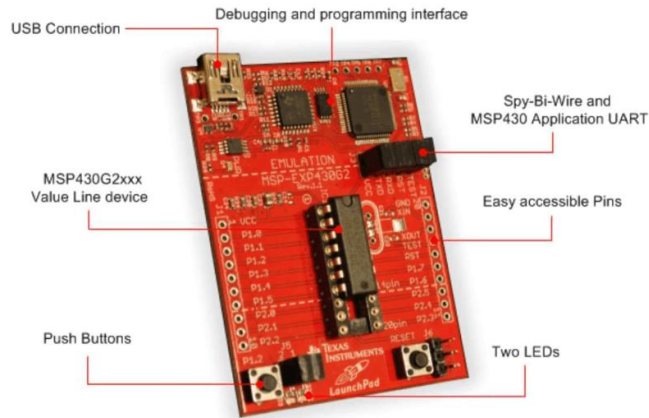


图 2 单片机 MSP430 g2553 板子示意图

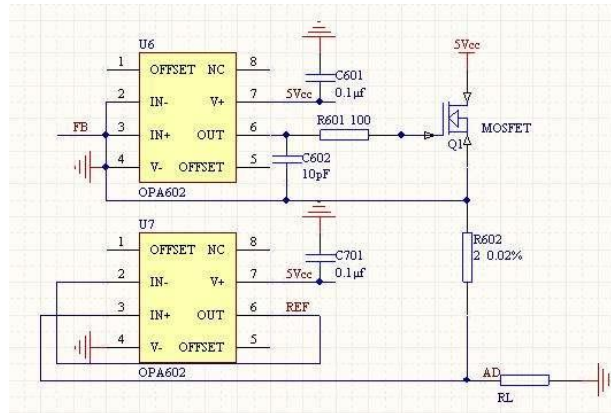


图 3 单片机 MSP430 g2553 电路连接图

另一方面，因为有了之前工程实践与科技创新 II -B 的基础，因此单片机部分的代码结构上与之课程的大致类似，在本项目中也只是在之前完成的内容上稍作修改，大部分的工作量在于不同小车的调试，包括舵机频率、小车蓝牙连接的稳定性等等。其中舵机频率，由于本小组使用的小车是两个后轮驱动，前轮是万向轮，因此只需要使动后两个轮子，但由于两个轮子驱动时会有差异，导致两个轮子虽然设置的是相同的频率，但实际上智能车走的路线并不是直线，因此在后期的调试中更多地关注于这一方面。如表 2 所示，是控制小车前进部分程序的伪代码：

表 2 控制小车前进的单片机代码

Algorithm 1: 控制小车前进 goAhead ()单片机端代码

```
Input: incomingByte=="G"
Output: 小车沿直线前进
1 stopBack();
  stopAhead();
2 digitalWrite(LEFT_BACK, HIGH);
  digitalWrite(RIGHT_AHEAD, HIGH);
3 analogWrite(LEFT_BACK,110); //set left tire's speed to 110
  analogWrite(RIGHT_AHEAD,90); //because it is slower than right tire
4 servo_pin 14.write(75);
```

2.2 APP 可视化页面设计

2.2.1 小车端

由于小车端需要完成的任务较简单，仅设计了两个界面：连接界面与视频界面。小车端手机与手持端手机接入同一无线局域网后，在小车端输入手持端手机 IP 地址（端口已经预设好），即可连接。连接成功后转入视频界面。

具体界面如图 3：



图 3 小车端 APP 界面一览

2.2.2 手持端

手持端页面设计采用了 Android Studio 自优化的侧滑模块 Navigation Drawer Activity，通过点击侧滑栏的对应项目唤醒相应 activity，执行指定任务，建立侧滑模块的选项则是如图 4 所示：

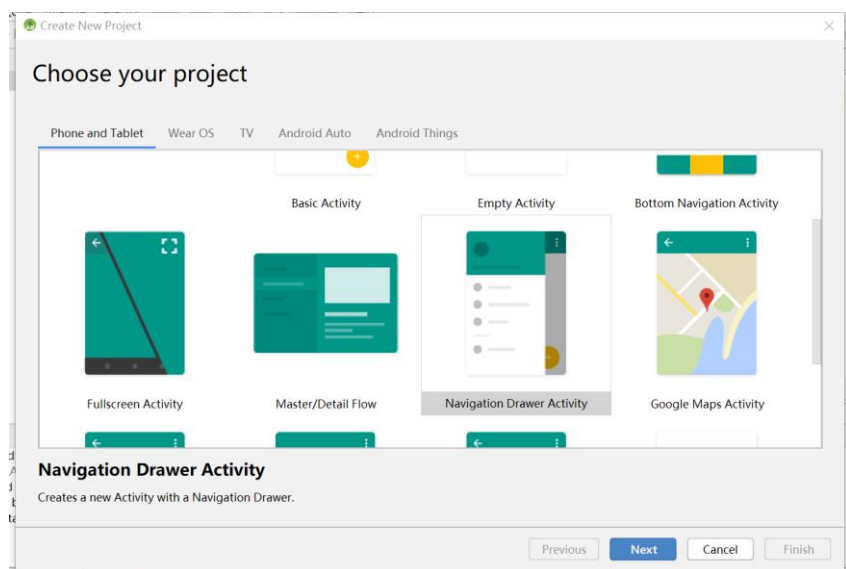


图 4 建立侧滑模块的项目

总共设置了五个任务：基本操作、蓝牙连接、手势识别、语音控制、重力感应。每个界面放置所需按钮及视频区域，从小车端接收的回传视频显示在上面。每个界面均采用 **Linear Layout** 布置。

具体界面如一系列的图 5 所示，其中图 5.1 是手持端 APP 的侧滑栏目显示时的主页面，图 5.2 是选择蓝牙连接的界面，图 5.3 是基本操作按键的界面，图 5.4 是手势识别的界面，图 5.5 是重力感应的界面，图 5.6 是语音识别的界面，所有的界面都尽量做的清晰明了且按键较少，为的是双机交互的时候可以用更大的空白空间来显示小车前方的视频路况信息：



图 5.1 手持端 APP 主页面

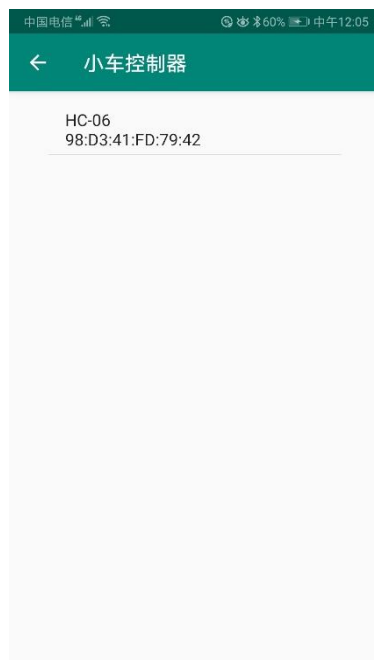


图 5.2 蓝牙连接界面



图 5.3 基本操作界面

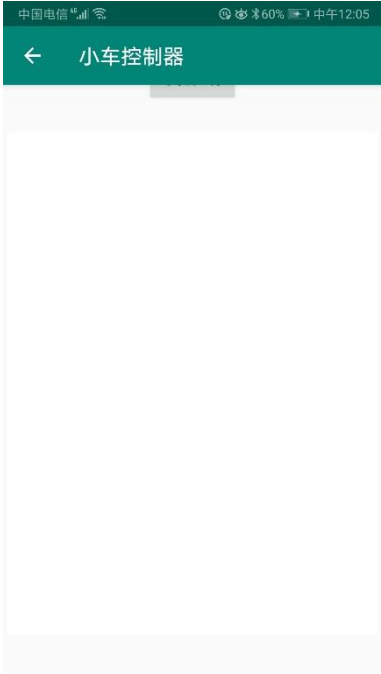


图 5.4 手势识别界面



图 5.5 重力感应界面



图 5.6 语音识别时的界面

2.3 APP功能实现

手持端实现多 Activity 切换，通过点击侧滑栏中的按钮唤醒相应 Activity，执行功能。全部功能封装实现，合并并在 package 中。

2.3.1 视频回传

建立连接：两部手机接入同一无线局域网，在手持端输入对应 IP 地址，建立 WiFi socket 连接，即可互传代表图片的比特流，如图 6 所示。

```
@Override
public void run() {
    try {
        sendSocket = new Socket();
        sendSocket.connect(new InetSocketAddress(IP, port), timeout: 10000); //端口号为30000
        sendSocket.setSoTimeout(200);
        ou = new DataOutputStream(sendSocket.getOutputStream());
        ou.write(bytes);
        ou.flush();
        Log.d(TAG, "Sender", msg: "Bytes Sent:" + bytes.length);
    } catch (Exception e) {
        Log.e(TAG, "Transmission failed:" + e.toString());
    } finally {
        lock.unlock();
    }
}
```

图 6 小车端 WiFi 连接

手持端接收视频：在网络连接线程中将接收到的代表图片的字节流进行解析，并传递给主线程在屏幕上显示，如图 7 所示。

```
@Override
public void run() {
    try {
        serverSocket = new ServerSocket(port);
        byte[] buffer = new byte[1024 * 1024];
        int bytesBuffered, bytesRead;
        while (receiving) {
            try {
                receiveSocket = serverSocket.accept();
                receiveSocket.setSoTimeout(500);
                inputStream = receiveSocket.getInputStream();
                bytesBuffered = 0;
                while ((bytesRead = inputStream.read(buffer, bytesBuffered, len: buffer.length - bytesBuffered)) > 0) {
                    bytesBuffered += bytesRead;
                }
                inputStream.close();
                receiveSocket.close();
                Log.d(TAG, msg: "Received:" + bytesBuffered + " bytes");
                lock.lock();
                bitmap = BitmapFactory.decodeByteArray(buffer, offset: 0, bytesBuffered);
            } catch (Exception e) {
                Log.e(TAG, msg: "Transmission failed:" + e.toString());
            } finally {
                lock.unlock();
            }
        }
    }
}
```

图 7.1 接收视频比特流


```

@Override
public void run() {
    if (receiveThread == null) {
        receiveThread = new Utils.ReceiveThread( port: 30000);
        receiveThread.start();
    }
    try {
        receiveThread.lock.lock();
        if (receiveThread.bitmap != null && mImageView != null) {
            mImageView.setImageBitmap(receiveThread.bitmap);
        }
    } finally {
        receiveThread.lock.unlock();
    }
    handler.post( r: this);
    // handler.postDelayed(this, 1);
}

```

图 7.2 解析比特流并传入 mImageView 中

小车端视频拍摄：申请相应的权限之后，利用 Android 自带的 Camera 类来操纵摄像头拍摄视频，然后将采集到的图像字节数组通过网络连接发送给控制程序进行显示，如图 8。

```

// 获取camera实例
public static Camera getCameraInstance() {
    Camera c = null;
    try {
        c = Camera.open();
    } catch (Exception e) {
        Log.d( tag: "TAG", msg: "camera is not available");
    }
    return c;
}

```

图 8 获取 Camera

2.3.2 蓝牙连接

蓝牙连接负责手机与小车间的指令传输，需要提前将手机和小车的蓝牙模块进行配对和连接。连接好 Server 与 Client 之后建立一个 Bluetooth socket，连接到该 socket 即可进行数据传输。

具体操作：先得到手持端蓝牙可以识别的设备列表，之后选中小车设备，建立连接，如图 9。

```

ConnectThread(connectableDevice device) {
    BluetoothDevice d = device.device;
    BluetoothSocket socket = null;
    UUID serviceUUID = d.getUuids()[0].getUuid(); // XXX
    Log.i(TAG, msg: "Using UUID:" + serviceUUID.toString());
    try {
        socket = d.createRfcommSocketToServiceRecord(serviceUUID);
    } catch (IOException e) {
        Log.i(TAG, msg: "Failed to create socket.", e);
    }

    mmDevice = device;
    mmSocket = socket;
}

@Override
public void run() {
    Instance mAdapterer.cancelDiscovery(); // just to make sure. Discovering consumes A LOT resources!
    try {
        BluetoothDevice d = mmDevice.device;
        Log.i(TAG, msg: "Connecting to " + d.getName() + "\n" + d.getAddress());
        mmSocket.connect(); // it blocks; which is the main reason this in a single thread.
        mmDevice.setConnected(mmSocket);
    } catch (IOException e) {
        Log.i(TAG, msg: "Client socket failed to connect", e);
        cancel();
    }
}

```

图9 选中蓝牙设备连接

2.3.3 指令传输

先注册与小车对应操作相同的 `public` 类，之后在各个操作中对应调用该类的函数，以实现向小车发送指令的功能。初始化指令为 `STOP`，如图 10 所示。

```

@Override
public void run() {
    if (mmDevice == null) {
        resetToast(text: "未找到任何蓝牙设备。");
    } else if (mmDevice.connected == null) {
        resetToast(text: "蓝牙设备尚未连接。");
    } else {
        try {
            if (mLastCommand != getMotionCommand()) {
                mLastCommand = getMotionCommand();
                Log.i(TAG, msg: "发送命令: " + getMotionCommand().toString());
                mmDevice.connected.write(serializeCommand(getMotionCommand()));
            }
        } catch (IOException e) {
            Log.i(TAG, msg: "发送命令失败: ", e);
        }
        handler.postDelayed(this, delayMillis: 20);
    }
}

```

图 10.1 向小车发送指令模块

```

public char toChar() {
    switch (this) {
        case STOP:
            return 'S';
        case FORWARD:
            return 'G';
        case BACKWARD:
            return 'B';
        case TURN_LEFT:
            return 'L';
        case TURN_RIGHT:
            return 'R';
        default:
            return 'S'; // XXX
    }
}

```

图 10.2 设定指令对应发送的 char

在基础操作模块中，使用前进、后退、左转、右转、停止五个按键，按下按键执行对应操作，向小车发送指令，如图 11 所示。

```

public void onResume() {
    super.onResume();
    mButtonStop.setOnClickListener((v) -> {
        setMotionCommand(MotionCommand.STOP);
    });
    mButtonForward.setOnClickListener((v) -> {
        setMotionCommand(MotionCommand.FORWARD);
    });
    mButtonBackward.setOnClickListener((v) -> {
        setMotionCommand(MotionCommand.BACKWARD);
    });
    mButtonTurnLeft.setOnClickListener((v) -> {
        setMotionCommand(MotionCommand.TURN_LEFT);
    });
    mButtonTurnRight.setOnClickListener((v) -> {
        setMotionCommand(MotionCommand.TURN_RIGHT);
    });
}
}

```

图 11 设置按键对应小车不同操作

2.3.4 手势识别

调用安卓手势库，利用 `onFling` 函数判断用户接触屏幕起始点的 x 轴与 y 轴差值，若大于某个标准则做出相应判定，如图 12 所示。

具体操作：当使用者接触到手机屏幕并向不同方位滑动时，程序判断手势方向向小车发出指令。

```
@Override
public boolean onFling(MotionEvent e1, MotionEvent e2, float vx, float vy) {
    float deltaX = e1.getX() - e2.getX();
    float deltaY = e1.getY() - e2.getY();
    if (deltaX > FLIP_DISTANCE) {
        setMotionCommand(MotionCommand.TURN_LEFT);
    } else if (deltaX < -FLIP_DISTANCE) {
        setMotionCommand(MotionCommand.TURN_RIGHT);
    } else if (deltaY > FLIP_DISTANCE) {
        setMotionCommand(MotionCommand.FORWARD);
    } else if (deltaY < -FLIP_DISTANCE) {
        setMotionCommand(MotionCommand.BACKWARD);
    } else {
        setMotionCommand(MotionCommand.STOP);
    }
    return true;
}
```

图 12 手势识别模块

2.3.5 语音识别

利用科大讯飞语音识别库，点击说话后识别出语句，将语句中的词语与提前设定好的方向词进行比对，若比对成功则发送对应指令，如图 13 所示。

具体操作：打开相应界面后说出包含“前、后、左、右、停”的语句，如“向左转”、“停止”等，即可做出对应操作。

```
public void onResult(RecognizerResult recognizerResult, boolean b) {
    String text = parseIatResult(recognizerResult.getResultString());
    if (text.isEmpty()) {
        return;
    }
    if (text.contains("停")) {
        setMotionCommand(MotionCommand.STOP);
    } else if (text.contains("前")) {
        setMotionCommand(MotionCommand.FORWARD);
    } else if (text.contains("后")) {
        setMotionCommand(MotionCommand.BACKWARD);
    } else if (text.contains("左")) {
        setMotionCommand(MotionCommand.TURN_LEFT);
    } else if (text.contains("右")) {
        setMotionCommand(MotionCommand.TURN_RIGHT);
    }
}
```

图 13 语音识别模块

2.3.6 重力感应

通过传感器管理器注册加速度计的监听器，当手机位置变化时，通过获取手机当前重力加速度在 x, y, z 轴上的分量，判断手机的倾斜方向，并与规定的四个方向比对，如图 14 所示。

具体操作：将手机水平放置，分别向前后左右倾斜，小车即可完成对应任务。

```

SensorManager.remapCoordinateSystem(mRotationMatrix, SensorManager.AXIS_X, SensorManager.AXIS_Y, mRotationMatrix);
SensorManager.getOrientation(mRotationMatrix, orientations);
orientations[0] = (float) Math.toDegrees(orientations[0]);
orientations[1] = (float) Math.toDegrees(orientations[1]);
orientations[2] = (float) Math.toDegrees(orientations[2]);
if (Math.abs(orientations[1]) > 30) {
    if (orientations[1] > 0) {
        setMotionCommand(MotionCommand.FORWARD);
    } else {
        setMotionCommand(MotionCommand.BACKWARD);
    }
} else if (Math.abs(orientations[2]) > 30) {
    if (orientations[2] > 0) {
        setMotionCommand(MotionCommand.TURN_RIGHT);
    } else {
        setMotionCommand(MotionCommand.TURN_LEFT);
    }
} else {
    setMotionCommand(MotionCommand.STOP);
}

```

图 14 重力感应模块

3 项目成果与自我评价/ Project achievements and self-evaluation

经过一个学期的共同努力与学习，本小组最终实现了基于安卓平台的智能小车控制系统。主要实现的功能是：通过按键控制小车前进、后退、左转、右转、停止等，通过指滑屏幕控制小车运动（其中向上滑就是前进，向下滑就是后退，向左滑是左转，向右滑是右转，停止是在屏幕中央画一个圆圈），通过语音口令控制小车运动（并对语音库进行了模糊处理，即“前进、向前走……”都可以让小车前进，“左拐、左转、向左……”都可以让小车左转等等），通过重力感应控制小车运动（手机向左倒就是左转，向后倒就是后退等等）。主要可视化的界面包括主界面的设计在内，本项目一共设计了六个页面，分别是：主页面、蓝牙连接页面、按键控制界面、手势识别界面、语音控制界面和重力感应界面，界面清晰直观并且用户友好性通过了后续的测试和评估，使用时也十分顺滑没有卡顿，总体而言，本项目的最终成果是优秀且丰富的。

而在项目进展的过程中，本小组的三位成员也有许多话想说：

高若彤：这次工科创课程还算圆满地完成了，因为之前有过工科创 II-B 的基础，所以开始以为这次课程不会很困难。但实际操作才发现，从蓝牙和 WiFi 传输，到 UI 界面布置，再到利用 Android Studio 平台进行 java 编程，最后编写出两个完整的 app，每一步对我来说都是很大的挑战。尤其是遇到关于讯飞语音的调用问题，按照官方资料配置好后软件却总是闪退，最后经过自己 debug 并求助于同学终于发现了问题所在并成功处理。在 app 编写过程中我查阅了相关书籍，得到了老师的指导，组员们也对我有很大的帮助，最终才得以完成这一任务。在这次课程中我收获良多，最终实现了用手机 app 控制蓝牙小车的功能。

刘箬凡：相对于工科创 II-B 的小车走黑线项目，这次的工科创 III-C 对我来说又是一次很大的挑战。由于我负责的是小车的单片机部分，因此前前后后跑了很多次的实验室，课程讨论群里也总能看到我在问老师什么时候在实验室。从一开始以为是蓝牙部分有问题去找了老师才知道根本没有拿到对的单片机的板子，到后来蓝牙模块供电出现不足而不稳定的问题，再到后来蓝牙可以使用但却无法传输指令（即在电脑上连接时可

以控制小车,但使用蓝牙串口调试工具时却无法使动小车),最后在实验室马上要测试的时候却发现程序烧录不到板子上去,最终又换了一块板子。在这一曲折的过程中,我学到了许许多多的关于单片机的知识,虽然最后都是在老师和助教的帮助下才解决的,但我在查阅资料的过程中也了解到了很多这方面的知识,不可不说是收获良多。

程玉婷:这次的科创综合了我们学的单片机知识和编程知识,经过一学期的努力我们终于完成了这个小车制作。在搭建 JAVA 平台和后期调试的时候都遇到了各种各样的困难,但是对我接下来的深入学习奠定了基础。感谢我的组长和组员非常认真负责地完成了自己的工作,遇到问题也会很耐心地讲解,然后一起解决,感谢老师和助教的辛苦付出,给了我们这样的机会来亲身体验安卓系统的开发,是非常宝贵的学习经历。我在其中起到的作用不算大,也和我自身的知识欠缺有关,希望以后可以学习到更多。感谢组内其他成员的努



力!



图 15 小组成员工作照



图 16 小车运行时的图像回传

图 17 本项目所用到的所有素材

参考文献:

- [1]冯向萍,张太红,白涛.“安卓应用程序开发”混合式教学模式探索与实践[J].计算机时代,2019(11):80-82+86.
- [2]侯瑞.MSP430 单片机实验课程教学探索与实践[J].教育现代化,2019,6(87):109-110+150.
- [3] https://blog.csdn.net/qq_30379689/article/details/52566339
- [4]许晓珊.基于 Android 应用的性能优化[J/OL].电子技术与软件工程,2019(23):57-58[2020-01-04].