

# 基于 Android 的蓝牙通信头盔应用<sup>①</sup>

张驰程, 周渊平

(四川大学 电子信息学院, 成都 610000)

**摘要:** 在驾驶摩托车的过程中, 为了保证安全, 都需要佩戴头盔. 而在行驶过程中, 会有手机来电或是查看时间等需求, 而在看时间或是接电话的过程中都存在安全隐患. 本文研究了利用蓝牙通信的方式使头盔连接 Android 手机, 通过语音指令的方式来操作手机和操作安装在头盔上的各个模块. 头盔上利用一块 51 单片机作为中央处理器对蓝牙模块以及各个模块进行操作和控制. 在 Android 手机端, 设计一款 APP, 使其能通过蓝牙连接到头盔, 并达到互相操作, 数据传输的功能; 同时还具备指令识别与语音合成的功能.

**关键词:** 蓝牙; 安卓; 语音识别; 串口通信; 单片机

## Application of Bluetooth Communication Intelligent Helmet Based on Android

ZHANG Chi-Cheng, ZHOU Yuan-Ping

(School of Electronic Information, Sichuan University, Chengdu 610000, China)

**Abstract:** In the process of driving a motorcycle, drivers have to wear helmets to ensure safety. However, there are some inconveniences and potential safety risks when drivers have a phone call or want to check the time while driving. In order to solve these problems, this paper studies the intelligent helmets which use Bluetooth connection to Android phone. The user can operate mobile phone and helmet to realize relevant functions through speech recognition. It uses 51 single chip microcomputer as the central processor to manipulate and control the Bluetooth module as well as other functional modules. Meanwhile, an APP is designed to enhance the user experience. It provides an easier interoperate between helmet and mobile phone via Bluetooth technology. This APP realizes not only data transmission function, but also instructions recognition and speech synthesis function.

**Key words:** Bluetooth; Android; speech recognition; serial port communication; SCM

随着交通工具的快速发展, 各种小型的机动和非机动载具越来越受人喜爱, 既有人们熟知的自行车、摩托车, 也有新兴的平衡车. 为了保障使用人员的安全, 往往需要佩戴头盔, 头罩虽然能有效的保护行驶时人的安全, 但却产生了许多问题: 佩戴时难以接听电话, 并且需手动控制, 此时手会脱离车把, 造成安全隐患. 考虑到这种情况, 需要一种能不用手操作就能完成对手机手操作, 并与使用者进行信息交互的新型智能头盔.

### 1 技术现状及研发目标

在智能骑行头盔市场, 技术最成熟的产品当属

SKULLY 公司的 skully AR-1 了, 该头盔主要采用了视频显示的交互方式. 该种方式虽然能很好的与用户进行信息交互, 但该种方式可能影响到用户视野, 分散驾驶者视觉注意力, 对于安全驾驶不利; 更重要的, 利用该种方式的交互也会使产品成本大大提高, 其高昂的售价令绝大多数骑行者望而却步.

而本文将使用单片机对蓝牙模块进行开发, 同时在 Android 手机端进行 APP 的开发. 最终达到头盔与 Android 手机之间能通过蓝牙进行语音交互, 只需说出语音指令, 手机端即可通过语音合成来反馈执行结果并执行相应操作. 本设计考虑充分利用手机的硬件资源进行一部分功能的实现, 如 GPS、加速传感器、

<sup>①</sup> 收稿时间:2016-05-24;收到修改稿时间:2016-06-30 [doi:10.15888/j.cnki.csa.005599]

方向传感器等,极大的降低了头盔的硬件成本.另外,在此基础上增加了自动温控风扇模块.主要目标是以为较低的硬件成本实现多功能控制,满足大多数骑行者的需求.

### 2 系统概述

本系统总体分为三部分:头盔部分, Android 手机部分, 以及连接两者的蓝牙无线通信.

#### 2.1 蓝牙协议

蓝牙协议体系结构分三个部分:底层硬件模块(Core Controller)该部分完成无线传输的硬件实现部分;中间协议层(Core Host)该层用于规定蓝牙设备需要实现的通用功能和协议层次;顶层应用框架(Profile)该层从应用场景的角度为蓝牙制定了不同的规范,这部分也是大众在生活中接触最多的部分.

本系统中会涉及到顶层应用框架里的两种 Profile, 即: SPP(Serial Port Profile)和 A2DP(Advanced Audio Distribution Profile). SPP 协议: 串行端口配置协议, 该协议能让蓝牙设备之间能创建串口进行数据传输. A2DP 协议: 蓝牙音频传输模型协议, 该协议是为了完成在蓝牙设备之间传输高清晰度的音频而制定的. 只要两个蓝牙设备都支持这两个协议规范, 那么, 两个蓝牙设备就既能传输串行数据, 也能传输高清晰度的音频了.

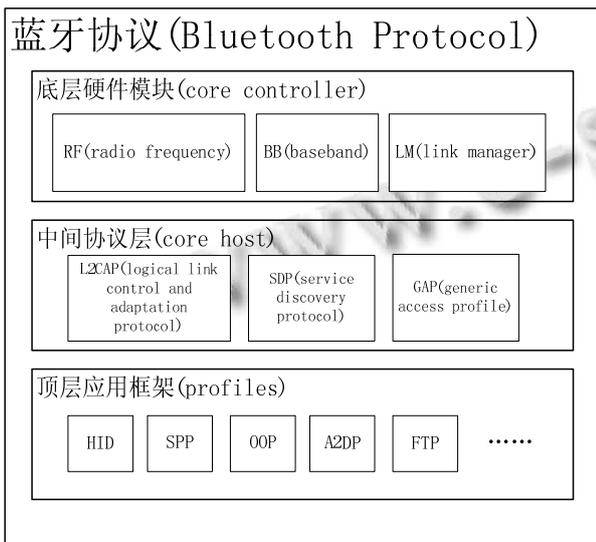


图1 蓝牙协议结构

#### 2.2 头盔结构

我们选用了一款常用的摩托车头盔, 并加装了以

下模块使其能完成相应功能: 单片机模块、蓝牙模块 (含麦克风和扬声器)、温度传感器、可变速风扇、语音放大模块 (含麦克风和扬声器)、充电电池.

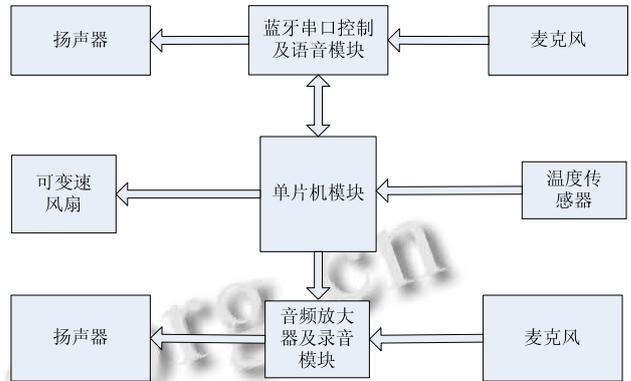


图2 头盔硬件结构框图

本文单片机采用了 STC89C51RC, 该单片机具有低功耗, 成本低, 特别是集成了串口引脚能供我们使用<sup>[2]</sup>. 蓝牙模块使用了 BK8000L, 该模块同时支持 SPP 协议和 A2DP 协议. 温度传感器采用的是 DS18B20 芯片, 该芯片具有单总线、体积小、分辨率高、抗干扰强等特点<sup>[3]</sup>. 音频放大采用了 PAM8406 模块, 该模块可调节音量, 具有优秀的抗噪性. 音频采集模块使用 ISD1820 模块, 该模块有直通模式和录音模式.

#### 2.3 Android 应用程序结构

Android 系统采用了分层的系统架构, 一共分为 4 层, 分别是: Linux 内核层 (Linux Kernel)、系统运行库层 (Android Runtime&Libraries)、应用框架层 (Application Framework)、应用层 (Application). 根据需求, 我们需要在应用层开发一款和头盔相匹配的应用程序. 该应用由两个主要部分组成:

**蓝牙通信:** 市面上的绝大多数 Android 手机都支持蓝牙功能, 并且都支持 SPP 协议和 A2DP 协议. 该部分完成与头盔蓝牙模块的配对, 与头盔进行 SPP 数据通信以及 A2DP 音频通信<sup>[4]</sup>.

**语音信号处理:** 当手机端接收到用户的语音指令数据后, 能解析出用户所说的指令内容并执行指令; 执行完成后, 将执行结果合成语音播放出来.

### 3 硬件模块的实现

硬件部分主要由 STC89C51RC 单片机以及蓝牙

模块 BK8000L 组成, 另外温度传感器以及风扇作为外围电路同单片机相连.

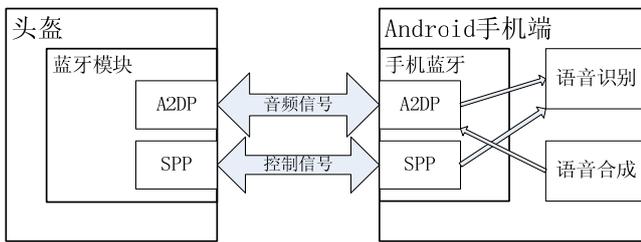


图 3 通信结构框架

### 3.1 电路连接

单片机作为头盔的中央处理器, 自身的 P3.0(RX)/P3.1(TX)与蓝牙串口模块的 RX/TX 交叉相连, 这样使单片机和蓝牙模块实现双向通信. 蓝牙模块需要将自身的 PREV, NEXT, PLY 连接到三个按钮; 同时蓝牙模块还需要连接麦克风以及扬声器用于语音信息的采集与播放.

表 1 按钮功能

按钮	功能
PREV	听歌上一曲
NEXT	听歌下一曲
PLY	播放/暂停, 接听电话/挂断电话

另外, 温度传感器连接单片机的 P1.6 口, 风扇连接单片机的 P1.0 口, 再将单片机以最小系统的方式连接好晶振、复位电路、以及电源.

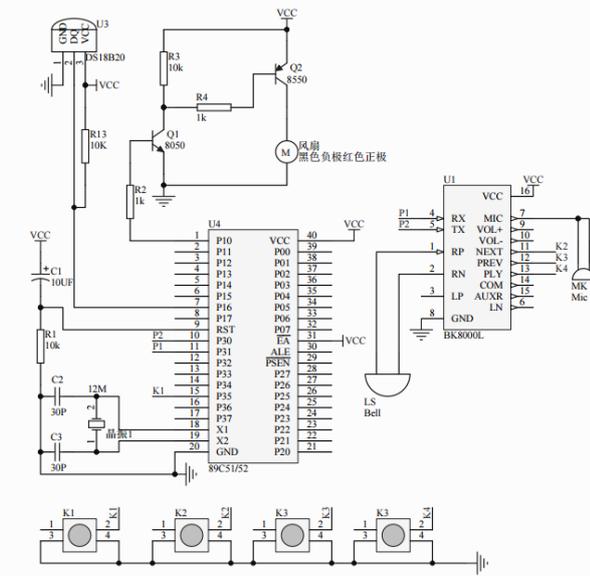


图 4 电路原理图

### 3.2 串行数据的创建与发送

#### 3.2.1 数据创建

串行数据的传输, 这里采用异步传输的方式, 所以在传输前, 需要确定好异步传输的数据格式, 需要指出的是异步传输的时候需要添加 2-3 个起始位以及结束位. 用 `uchar buffer[7]={ 0x68, 0x01, 0x00, 0x00, 0x00, 0x00, 0x16}` 来初始化数据格式.

起始字	起始字	温度高	温度低	指令触发	档位	结束字
-----	-----	-----	-----	------	----	-----

图 5 数据格式各字节意义

温度信息需要单片机从 DS18B20 的引脚处获取, 即 P1.6 引脚. 利用 `ReadTemperature()` 函数来获取温度信息, 并以高位和低位的方式存在 `buffer[2]` 和 `buffer[3]` 中. 当用户按下头盔上的 K1 按键的时候, 即代表准备说语音指令了, 这时将 `buffer[4]` 改变为 `0xff`, 当发送完毕时再置为 `0x00`.

#### 3.2.2 数据发送

单片机通过串口发送数据是将需要传输的数据缓存到自身的串口缓存 SBUF 中, 当发送完成之后会将 TI 置 1, 我们需要将发送串口中断 TI 置 0 就可发送下一个字节的数据了. 如果要一次性发送 7 字节的需要用到循环函数将 `buffer[7]` 的数据依次发出.

单片机的发送程序如下:

```
void send_char_com( unsigned char ch)
{
    SBUF=ch;
    while(TI==0);
    TI=0;
}
```

### 3.3 温控风扇

单片机通过 `ReadTemperature()` 函数接收到温度芯片的数据后, 可以根据温度情况来控制连接在单片机上的风扇转速. 这里控制风扇转速采用了脉冲式控制, 如全速时, 给风扇的全是高电平. 半速时, 是高低电平交换. 在通过不断的测试之后, 确定了 5 个转速档位, 分别对应 5 种不同的温度区间, 使其能在不同温度时自动切换档位.

## 4 软件部分的实现

Android 开发选择在 Eclipse 平台上进行. 语音识

别以及语音合成采用了讯飞开放平台提供的 API, 蓝牙相关功能采用了 Google 提供的 android.bluetooth 包里的相关类.

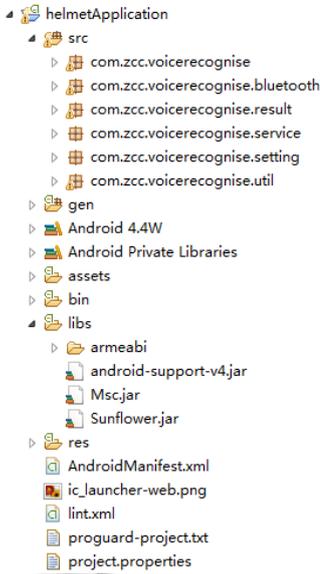


图6 Android 工程实例

整个应用程序主要由蓝牙部分和语音信号处理构成, 程序流程图如图7所示.

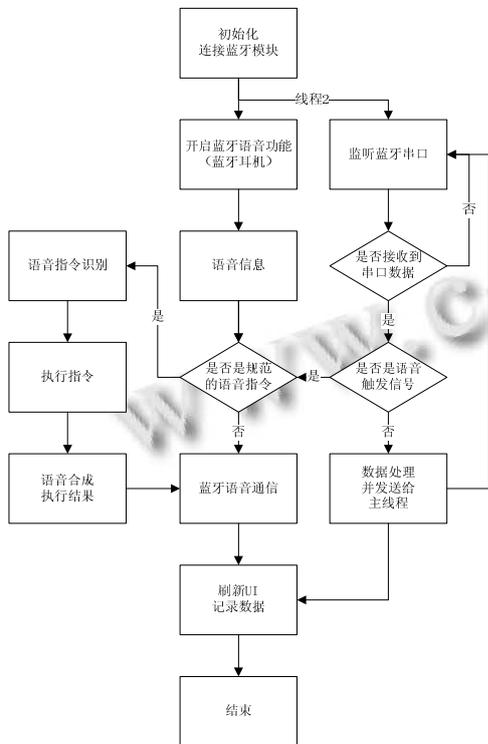


图7 Android 程序流程图

### 4.1 蓝牙连接流程

#### 1) 打开蓝牙开关

由于存在用户在使用 APP 之前没有打开蓝牙开关, 所以必须使用户在打开 APP 是询问用户打开蓝牙开关. 这里使用 BluetoothAdapter 类里面的一个静态变量: BluetoothAdapter.ACTION\_REQUEST\_ENABLE 初始化一个 Intent, 再调用 startActivityForResult()函数, 当回调结果为 RESULT\_OK 时, 代表用户已经开启了蓝牙开关. 反之如果未打开, 则退出程序.

#### 2) 连接头盔的蓝牙模块

在连接头盔前首先需要查看头盔蓝牙模块的 MAC 地址, 使用 getRemoveDevice(macAddress)函数得到远程蓝牙设备类 BluetoothDevice 的对象 device, 这时如果直接在主线程中连接远程蓝牙设备对象, 由于 Android 系统的 ANR(application not responding)机制, 可能导致程序崩溃强退, 于是此时需要单开一个线程来连接该远程蓝牙设备对象.

创建一个子线程类 connectThread, 在该线程里利用刚刚获取到的 device 对象中已经定义好的函数 listenUsingRfcommWithServiceRecord(name, UUID)来得到一个套接字 BluetoothSocket 与其连接, 这里需要特别注意 UUID 这个变量, 该变量决定你用什么样的协议来连接远程蓝牙设备.

表2 常用协议的 UUID

协议名称	UUID	数据类型
SPP	00001101-0000-1000-8000-00805F9B34FB	串行数据
A2	0000110A-0000-1000-8000-00805F9B34FB	音频数据
DP	0000110B-0000-1000-8000-00805F9B34FB	文档数据
FTP	00001106-0000-1000-8000-00805F9B34FB	对象数据
OPP	00001105-0000-1000-8000-00805F9B34FB	

获取到 Socket 套接字之后, 使用该套接字的函数 connect()即可进入阻塞状态, 直到成功连接或是连接失败. 这里只需要使用 SPP 协议的 UUID 进行连接就可以了, 那是因为 A2DP 是 Android 自带的协议, 一般开发者是不需要接触到它的 UUID 的, 但也是经过 UUID 去建立 Socket 接口的, 只是这个过程 Android 系统能自动完成. 所以当我们通过 SPP 协议的 UUID 连接到远程蓝牙设备的时候, 其实不仅连接了 SPP 协议部分, 还连接了 A2DP 协议部分.

#### 3) 数据传输

当连接成功后, 会涉及到两种数据的传输, 即串

行数据和音频数据。串行数据是头盔发送过来的温度信息和语音识别触发信号以及风扇档位信息，同样的，为了接收这些数据也需要在子线程中执行，保证程序一直处在侦听的状态。创建线程 `connectedThread`，在该线程里通过已经获取到的 `BluetoothSocket` 来获取 `InputStream` 和 `OutputStream`，通过这两个流对象就可以发送或是接收串行数据了。

但是这里存在一个问题，那就是音频数据和串行数据不能同时发送或同时接收。要达成这样的功能，需要手动调用 `BluetoothA2DP` 类的 `connect()` 函数，但由于该函数被隐藏起来了，我们可以使用 Java 的反射机制来调用该函数，经过这一步就可以让系统自动连接，并且自动管理音频，也就是说我们的 APP 连接了这个 `Socket` 之后，系统其他 APP 中播放音频文件时也能共用该 `Socket` 了。之所以能达到这样的功能，是因为这个 `connect()` 函数是调用了 `IBluetoothA2dp.aidl` 文件的接口，也就是说这个功能使通过 IPC (Inter Process Communication) 机制来实现的。

#### 4.2 语音指令的识别与执行

语音指令的识别和执行涉及到语音识别和语音合成，使用了讯飞开放平台提供的 API，所有的类打包在了 `Msc.jar` 的包中，`Sunflower.jar` 是统计移动流量统计的类。

##### 4.2.1 语音指令的获取

需要确定用户何时开始准备开始说语音指令，当程序在接收到头盔上的语音指令触发信号时，便开始将之后的一段通过蓝牙传输过来的语音信息作为语音指令，直到用户超过 2s (该时间可设置) 没说话为止。

##### 4.2.2 语音指令的识别与合成

主要功能是将语音信息转换成字符串类型，并将字符串中的指令信息提取出来执行。`SpeechRecognizer` 这个类就是用来听写的，使用该类的对象前应该先对该对象进行参数设置，执行该对象的 `setParameter()` 写入参数设置。然后当接收到语音指令触发信号的时候，调用 `startListening(RecognizerListener)` 函数就开始进行识别了；`RecognizerListener` 类有 4 个回调函数代表在识别的各个阶段，在 `onResult()` 这个函数中可以获取到识别的结果。在这个函数里可以将结果转换成字符串类型。之后对字符串进行拆分，拆分为指令类型和指令内容。

表 3 语音指令类型与内容

指令类型	指令内容
打电话给	联系人姓名
听音乐	音乐名称
风扇档位	档位数字
当前天气	NULL
头盔温度	NULL
当前时间	NULL

表 3 中，打电话给、听音乐、风扇档位三个指令执行时会直接产生效果，不会产生数据回复结果。而当前天气、头盔温度、当前时间都会产生字符串类型的数据结果，所以这三种指令还需要将字符串转换成语音信息通过蓝牙传输到头盔。

语音合成需要用到 `SpeechSynthesizer` 类，同样的，使用前需要通过 `setParameter()` 进行参数设置，之后只需要调用该类的 `startSpeaking(text, mTtsListener)` 函数就可以吧 `text` 的内容通过语音的方式播放出来了。`mTtsListener` 是一个 `SynthesizerListener` 类的对象，通过对该类回调函数的重写可以控制语音播放的各个过程。

## 5 系统测试

测试整个系统是否可以正常工作由以下几步组成：



图 8 头盔侧面图

- 1) 头盔各个模块能否正常上电，风扇能否正常转动，档位切换能否顺利进行。
- 2) 语音放大模块能否正常播放语音，录音功能能否正常工作。
- 3) 打开手机端安装好的 APP，打开蓝牙后能否成功连接头盔。

4) 戴上头盔,按头盔上的语音触发按键,并测试各个语音指令能否正常识别并执行。

经过以上测试过程之后各个部分均能正常工作也就达到了测试目标了。



图9 头盔正面图



图10 软件效果

## 6 结语

通过以上的分析及测试,该程序能成功的从头盔处获取数据,并进行分析及执行.实现了智能手机与可穿戴设备的互联,并且在操作过程中几乎可以不用到双手,利用这一特性,还能开发出更多的指令内容,

例如语音导航、跨应用操作程序等等.对于越来越多的可穿戴式设备,本系统提出了一种应用场景,能满足大多骑行人员的需求.既保证了安全,又能满足骑行者的功能需求。

## 7 展望

下一步,准备添加利用更多种的传感器,从而让骑行者进一步了解当前骑行状态并对骑行安全进一步保障.并考虑增加空气质量检测功能,同时添加一些保护部件使其能达到防雾霾的效果.同时对头盔外观进行进一步改善,使其更时尚,更容易被大众接受.同时,考虑将各组件整合成一个整体,并将其设计为可移植的模块,使其能在更广的范围,如:交警、工人、医生等职业人群进行应用。

## 参考文献

- 1 张群,杨絮,张正言,陆起涌.蓝牙模块串口通信的设计与实现.计算机技术应用,2012,31(3):79-82.
- 2 林慧勇.STC89C51RC/RD+系列单片机使用简介.电子报.2007-3-18(015).
- 3 何东坡,任贵波,韩春鹏,张鸿儒.DS18B20 在季冻土区公路路基温度场测量中的应用.电子技术应用,2010,36(4):48-50.
- 4 苏民生.蓝牙 A2DP 技术与音频应用测试.电子设计应用,2008,(11):1-3.
- 5 邓毅.面向矿山物联网的多功能智能头盔软件关键技术研究[学位论文].杭州:浙江大学,2014:128-131.
- 6 李鑫.基于 ARM、蓝牙通信的心电采集系统的设计[学位论文].西安:西北工业大学,2007:48-50.
- 7 申康.蓝牙技术的特点及实施.微电子技术,2001,29(5):58-60.
- 8 李元建.蓝牙与智能家居.电子技术,2007,34(2):5-7.
- 9 陈哲.智能家居语音控制系统的设计与实现[学位论文].成都:电子科技大学,2013:21-40.