

2016 年英特尔杯大学生电子设计竞赛嵌入式系统专题邀请赛

20146Intel Cup Undergraduate Electronic Design Contest

- Embedded System Design Invitational Contest

作品设计报告

Final Report



Intel Cup Embedded System Design Contest

报告题目：护航天使——基于机器视觉的车载机器人

学生姓名：张乐纯 刘畅 陈超

指导教师：戴志涛

参赛学校：北京邮电大学

2016 年英特尔杯大学生电子设计竞赛嵌入式系统专题邀请赛

参赛作品原创性声明

本人郑重声明：所呈交的参赛作品报告，是本人和队友独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果，不侵犯任何第三方的知识产权或其他权利。本人完全意识到本声明的法律结果由本人承担。

参赛队员签名：

日期： 年 月 日

护航天使——基于机器视觉的车载机器人

摘要

护航天使是一个多功能的车载机器人，旨在为有车一族们提供驾车时的安全保障——消除由于在路上紧急停车时人为放置紧急停车牌而导致的危险隐患。本系统以 Intel Minnowboard Turbot 嵌入式平台为主体，通过机器视觉实现机器人的自动巡航，集成了 GPS、打电话、语音合成等模块，完成了一个功能多样、小巧实用的车载机器人。采用了 Genuino 101 模块控制机械及传感器部分，使用基于 OpenCV 库的优化道路线识别算法和驾驶控制算法实现自动巡航。陀螺仪模块实时监控机器人的行驶过程，一旦发生意外，会自动拨打 110 报警，汇报自己的事故地点。不仅如此，其摄像头还会完整记录下事故前的道路影像。另外，护航机器人提供易用健壮的用户交互方式，用户可通过打电话、发短信与机器人交互。测试结果表明，机器人整体运行稳定，达到了预期效果。

关键词：车载机器人，机器视觉，GPS，OpenCV，道路线识别算法

Angel Escort - Vehicular Robot Based on Machine Vision

ABSTRACT

Vehicle mounted robot is a multifunctional vehicle robot designed for a safe driving. It could drive automatically and raise the emergency triangle sign, which could prevent the danger of human going backward along the road and raising the sign.

Vehicle mounted robot, built in Intel Turbot embedded platform, contains GPS unit, cellphone unit, speech unit. It can auto-driving for the computer vision and robot control. It is indeed a elegant, useful robot with a great many functions.

It uses Genuino 101 unit to control machine and sensor, meanwhile utilize lane detection algorithm and driving control algorithm to drive automatically. During the process, gyroscope unit always keep an eye on the robot. And once it detects the acute change of accelerated speed, which is very likely to be an accident, it will call the police and tell the police the exact location. Moreover, it records the video of the accident. It is easy for user to interact with vehicle mounted robot by just texting and making a call. And, it is suitable for car, for the reason that it can be easily folded while it is not in use.

Key words: vehicle robot, computer vision ,GPS,OpenCV, robot control

目 录

第一章 护航天使系统概述.....	1
1.1 选题背景	1
1.1.1 我国道路交通事故现状.....	1
1.1.2 紧急停车警示牌摆放的相关法规	1
1.1.3 自动驾驶技术的发展现状.....	1
1.1.4 机器人的发展现状.....	2
1.2 护航天使机器人的意义.....	2
1.3 护航天使机器人的功能.....	2
1.4 护航天使机器人的特点.....	3
1.5 护航天使机器人系统软件工具集.....	3
1.6 护航天使机器人系统硬件工具集.....	3
第二章 护航天使系统需求分析.....	4
2.1 整体需求分析.....	4
2.2 各个模块需求分析.....	4
2.2.1 外设模块	4
2.3 突发状况及应对措施.....	5
2.3.1 机器人电池耗尽	5
2.3.2 与 Turbot 主板的串口通信意外断开	5
2.4 系统整体流程.....	5
第三章 基于机器视觉的护航天使机器人的设计.....	7
3.1 系统硬件设计.....	7
3.1.1 硬件的组成与连接.....	7
3.1.2 系统硬件的控制部分的设计与实现.....	9
3.2 系统算法设计.....	10
3.2.1 Minnowboard 主程序	10
3.2.2 道路线识别算法.....	11
3.2.3 语音合成模块	15
3.2.4 GPS 模块算法	16
3.3 扩展外设的设计实现.....	19
3.3.1 Genuino 101 模块	19
3.3.2 电机驱动模块	20
3.3.3 稳压模块	20
3.3.4 警示牌升降模块	21
3.3.5 舵机转向模块	21
3.3.6 GPS+GSM 模块	21
3.3.7 语音合成模块	21
3.4 护航机器人实物效果.....	22
第四章 系统测试	26
4.1 硬件测试	26

4.1.1 四杆结构测试。	26
4.1.2 Genuino 101 开发板的串口控制功能	26
4.1.3 语音合成模块测试	27
4.1.4 SIM808 模块测试	27
4.2 软件测试	28
4.2.1 道路线识别算法测试	28
4.3 联合测试	30
4.3.1 测试环境.	30
4.3.3 测试结果与结论.	31
第五章 总结	32
附录	36
Minnowboard Turbot 代码:	36
Genuino 101 代码:	82

第一章 护航天使系统概述

1.1 选题背景

1.1.1 我国道路交通事故现状

随着我国道路交通事业的飞速发展，交通事故猛增已成了交通管理所面临的严重。汽车交通作为人类文明的标志，彻底地改变了人类发展的进程，给人类以舒适和便捷等正面效应的同时也给人类生活带来一些负面效应，交通事故就是其中最严重、危害最大的负面效应之一。正因为交通事故造成的严重后果，世界各国尤其是发达国家都对事故预防及对策倾注了大量人力、物力和财力，制定了较为完善的道路交通管理、法规和相关政策，许多国家从 20 世纪 60 年代起开始实行了综合治理交通和减少交通事故的措施，虽然每年因交通事故造成的损失很高，但交通事故上升的势头已趋于平缓，欧美、日本等发达国家交通事故尤其是近几年交通事故处于稳定趋势，而我国的道路交通事故所造成的损害后果却依然严重，而且一直处于上升的趋势，每年交通事故死亡人数居于世界首位。近年来在我国机动车数量快速增长的情况下，交通事故及伤亡人数呈不断上升趋势。

1.1.2 紧急停车警示牌摆放的相关法规

我们国家的《道交法》五十二条明确地规定，机动车在道路上发生故障需要停车排除故障时，驾驶人应当立即开启危险报警闪光灯，并且如果机动车能移动的尽快移动到安全地带，不能移动的应当开起报警闪光灯，这个灯不能停止，并且在来车的方向要设置警告标志，就是我们平常说的安全的三角架，扩大示警的距离，必要时的话还要报警。在我们国家《道交法》实施条例的六十条也明确地规定，安全的距离一般是在 50 到 100 米，因为一般的道路考虑到车速、刹车，如果在高速公路上通常行驶的速度比较快，这时候安全的距离设置应该是在 150 米以外，有明确的规定，主要是为了避免二次事故的发生，后面来车有一个安全的警示。

1.1.3 自动驾驶技术的发展现状

自动驾驶汽车又称无人驾驶汽车、电脑驾驶汽车，是一种通过电脑系统实现无人驾驶的智能汽车。自动驾驶技术依靠人工智能、视觉计算、雷达、监控装置和全球定位系统协同合作，让电脑可以在没有任何人类主动的操作下，自动安全地操作机动车辆。自动驾驶技术在 20 世纪就已经有数十年的研发历史，21 世纪初呈现出接近实用化的趋势。除沃尔沃、通用等汽车厂商外，一些 IT 行业巨头也涉足这一领域。谷歌自动驾驶汽车于 2012 年 5 月获得了美国首个自动驾驶车辆许可证，预计于 2017 年前进入市场销售。百度与德国宝马汽车公司合作开发自动驾驶汽车，计划于 2017 年正式推出自动驾驶车辆。

人工智能、自动驾驶技术的蓬勃发展，为护航机器人的自动驾驶提供了技术可行性方案。另外，道路线识别算法的研究工作也已经进行了非常多年头，相对成熟。

1.1.4 机器人的发展现状

智能机器人则是一个在感知-思维-效应方面全面模拟人的机器系统，外形不一定像人。这种机器人带有多种传感器，能够将多种传感器得到的信息进行融合，能够有效的适应变化的环境，具有很强的自适应能力、学习能力和自治功能。

智能机器人涉及到许多关键技术，这些技术关系到智能机器人的智能性的高低。这些关键技术主要有以下几个方面：多传感信息耦合技术，导航和定位技术，机器人视觉技术，智能控制技术，人机接口技术。

现代智能机器人基本能按人的指令完成各种比较复杂的工作，如深海探测、作战、侦察、搜集情报、抢险、服务等工作，模拟完成人类不能或不愿完成的任务，不仅能自主完成工作，而且能与人共同协作完成任务或在人的指导下完成任务，在不同领域有着广泛的应用。

1.2 护航天使机器人的意义

设计护航机器人的初衷是为了帮助人们减少在行车过程中的隐患危险。众所周知，在行车过程中，难免有汽车抛锚等突发状况，尤其是在高速公路上。此时不得不在道路旁边的紧急停车道停车，并进行检修或者报警。根据我国道路交法规，此时故障车辆的车主需要在距离车后方 150m 到 200m 的距离范围内摆放停车警示牌。但是此时就需要人在紧急停车带逆行去摆放紧急停车牌，这在高速公路上是十分危险的行为。

所以护航机器人的意义就是通过一个功能齐全、小巧实用的车载机器人来代替人完成摆放紧急停车牌的工作，并且通过机器视觉实现全自动巡航。不仅如此，还有翻车报警功能，当车载机器人被后来车辆撞倒后，立刻向车主报警。若车主在一定时间内没有解除警报，则立刻拨打 110 报警，并通过 GPS 模块和语音合成模块获得并播报自己的事故地点，实现了一套完整的安全警报流程。

1.3 护航天使机器人的功能

护航机器人在用户车子发生故障之后，能够自动驾驶至车辆后方摆放紧急停车警示牌，并且在遇到危险时，通过向用户手机拨打电话进行警示，同时可自动拨打 110 报警。

(1) 自动驾驶示警功能：启动护航机器人后，护航机器人能够沿当前道路自动驾驶，到达目的地后，举起紧急停车警示牌。

(2) 报警功能：当护航机器人检测到巨大加速度变化后，会向用户手机拨打电话，若用户挂掉了电话，则护航机器人认为用户安全；若用户迟迟未接电话，护航机器人认为用户发生了危险，会自动拨打 110 报警，播报当前的经纬度和相关报警信息。

(3) 用户注册：用户通过手机向护航机器人发送“HHTS”短信息，成为护航机器人的使用者之一。使用者可以通过打电话控制护航机器人。用户通过手机向护航机器人发送“HHTS0”短信息，成为护航机器人的拥有者(拥有者只能有一个)。拥有者在护航机器人检测警报后，会接到电话提醒。

(4) 远程控制：护航机器人的注册用户可以通过拨打电话控制机器人。第一次拨打电话，使机器人停止前进，第二次拨打电话让机器人倒车回到出发点。

1.4 护航天使机器人的特点

(1) 自动驾驶：护航机器人摄像头距地仅有 40 厘米，现如今用于自动驾驶的算法和成熟的道路识别算法几乎都无法直接应用，护航机器人中集众之所长，有自己的一套自动驾驶算法。

(2) 交互方式简单健壮：用户通过短信和电话两种方式与护航机器人进行交互，无论事故发生在什么地区，什么天气等外界环境下，2G 通信网络相较于其他通信方式最为稳定可靠，所以健壮性较好。另外，短信和电话是手机的两个最基本的功能，用户只要有一台普通的手机，即可与护航机器人进行交互，十分简单。

(3) 一体化系统：护航机器人上集成了自动驾驶、测险报警、远程控制等功能，是一个高度一体化的系统，同时拥有较好的扩展性。

1.5 护航天使机器人系统软件工具集

表 1-1 系统软件工具集

名称	简介	作用
OpenCV	跨平台开源计算机视觉库	用于图像识别算法的构建
Gnome-Session	Ubuntu 自带的开机启动编辑器	开机启动主程序
Qt Creator	跨平台 C++应用程序开发框架	开发主程序的 IDE
Arduino	开源电子原型平台	Genuino101 的开发

1.6 护航天使机器人系统硬件工具集

表 1-2 系统硬件工具集

名称	接口	作用
Intel® Minnowboard Turbot	USB	运行主程序、控制外设
Genuino 101	USB、GPIO	控制机械、传感器
MG996R 舵机	信号线	控制机器人转向、立警示牌
USB-TTL 模块	USB、串口	连接串口到 Minnowboard
SYN6228 语音合成模块	串口	语音合成
SIM808 模块	串口	外接 GPS 天线，接打电话，收发短信
电机	JOHNSON540	驱动机器人前进
摄像头	USB	捕获外部图像
麦克风	音频接口	接收语音
LM2596S 5V 稳压模块	GPIO	将电源稳压至 5V，为 Minnowboard Turbot 平台和扩展外设供电

第二章 护航天使系统需求分析

2.1 整体需求分析

护航天使车载机器人作为车载机器人，需要独立供电。与用户交互过程需要无线通讯功能，主要功能为摆放紧急停车牌和报警功能，需要实时监测路况信息并且进行分析，对机器人的行进策略进行指导。

机器人选用 Minnowboard Turbot 平台作为中控，装有 Ubuntu 14.04 系统，运行道路识别主程序，Genuino 101 模块负责大部分的硬件扩展。

2.2 各个模块需求分析

2.2.1 外设模块

2.2.1.1 摄像头

由于护航机器人需要在紧急停车带自动巡航，故需要通过摄像头获取路面信息，包括是否有车辆以及通过对道路线的分析判断自己是否驶出了紧急停车带。

2.2.1.2 GPS

当护航机器人在车后 150m 摆放紧急停车牌时，如果有后面的车辆装上机器人，机器人需要报警，在报警时需要通过 GPS 进行定位，以准确播放出当前的地理位置，使机器人提供的报警信息更具有参考价值。同时，在机器人行驶过程中，需用 GPS 对它行走的方向进行二次修正。

2.2.1.3 语音合成

在护航机器人报警过程中，获得了 GPS 定位后，需要转换成语音，通过喇叭播放出来，完成报警过程。

2.2.1.4 陀螺仪

在护航机器人行走过程中，需要陀螺仪检测机器人是否被翻到，或者是否被后面的车辆撞上。这些需要陀螺仪检测机器人的三轴加速度，是否短时间内的巨大变化。

2.2.1.5 拨打电话

当陀螺仪检测到巨大变化后，需要向车主进行汇报。而通知车主最便捷最直接的方式就是拨打车主的电话。所以，护航机器人还需要装有 sim 卡，在紧急时刻以电话的方式通知车主。

2.2.1.6 稳压设备

因为护航机器人的主板规格是 5V 供电的，为了主板的安全性和机器人的健壮性，需要有一个 5V 的稳压模块保证主板的供电不会过高，从而确保机器人的正常运行。

2.2.1.7 转向设备

在护航机器人拍摄并分析完成

2.2.2 控制模块

一级模块为 Minnowboard Turbot 平台，二级控制模块为 Genuino 101 模块。

2.3 突发状况及应对措施

2.3.1 机器人电池耗尽

本系统对小车的两块车载电源都有完整的监控芯片，当电池电压略低于电池原始电压后，立刻使蜂鸣器发声报警。这样能在电池完全耗尽前让用户得知机器人的电量情况，及时充电。

2.3.2 与 Turbot 主板的串口通信意外断开

Minnowboard Turbot 主板不会立刻断开与拓展外设的连接，而是会启动延时程序，遍历所有的串口，直到重新连接上并且恢复串口通信后再继续运行。

2.4 系统整体流程

本系统的系统整体流程如图 2-1 所示。

当机器人开机后，主板 Minnowboard Turbot 和副板 Genuino101 同时供电，主板 Minnowboard Turbot 与副板通 Genuino101 过串口进行通信。主板 Minnowboard Turbot 开机后给副板 Genuino101 发送直走信息，同时通过摄像头获取当前的道路信息，经过道路线识别算法后得到预判道路线。获取到预判的道路线后，结合自身姿态，判断车体是否偏离了道路线的中间。如果向左偏移，则通过串口给 Genuino101 发出右转信息，反之发出左转信息。当 Genuino101 收到相应的信息后，改变转向舵机的 PWM 波信号，从而改变机器人的行走方向。

在 Genuino101 板内置陀螺仪，用于实时监控机器人的三轴姿态角。如果机器人有剧烈震动或翻转，那么就会启动自动报警：当判断机器人翻车后，Minnowboard Turbot 板会立刻通过串口给打电话模块发送信息，立刻给车主报警，若车主查看后排除危险，只需挂断电话即可。若车主没有注意危险情况，则机器人未得到车主相应，就会立刻拨打 110 报警，并通过 GPS 模块获取当前的经纬度信息，再由语音模块反馈给麦克风，实现报警功能。若车主在机器人到达目标位置之前就已经排除了危险，可以继续车程，则只需要给机器人拨打电话，则机器人接收到车主的电话后会自动挂断，然后原地倒车再返回。返回路上同样是通过机器视觉实现全自动巡航。

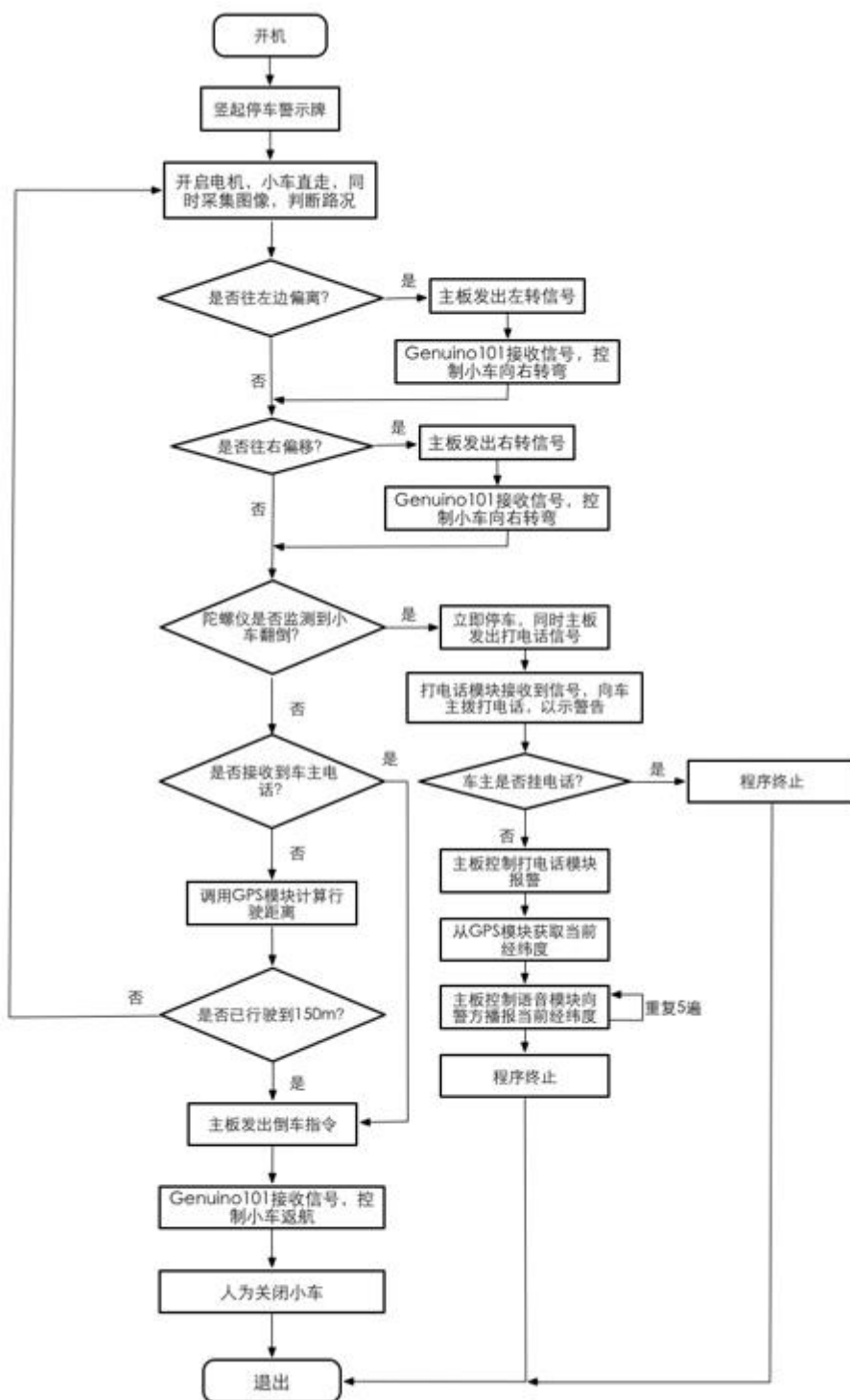


图2-1系统整体流程

第三章 基于机器视觉的护航天使机器人的设计

3.1 系统硬件设计

本机器人主要由两部分组成，Minnowboard 及其附属外设，Genuino 及其连接的舵机、电机等。机器人的类型为车型机器人，硬件部分主要集中在机器人上，机器人的硬件设计分为三层，具体见系统硬件设计。机器人的软件集中在 Minnowboard Turbot 和 Genuino 101 上，其中 Minnowboard Turbot 主要负责采集摄像头等各种外设发送的数据并加以处理，使机器人完成各种各样的动作。Genuino 101 负责舵机、电机的具体控制。

3.1.1 硬件的组成与连接

硬件的整体布局如图 3-1 所示。

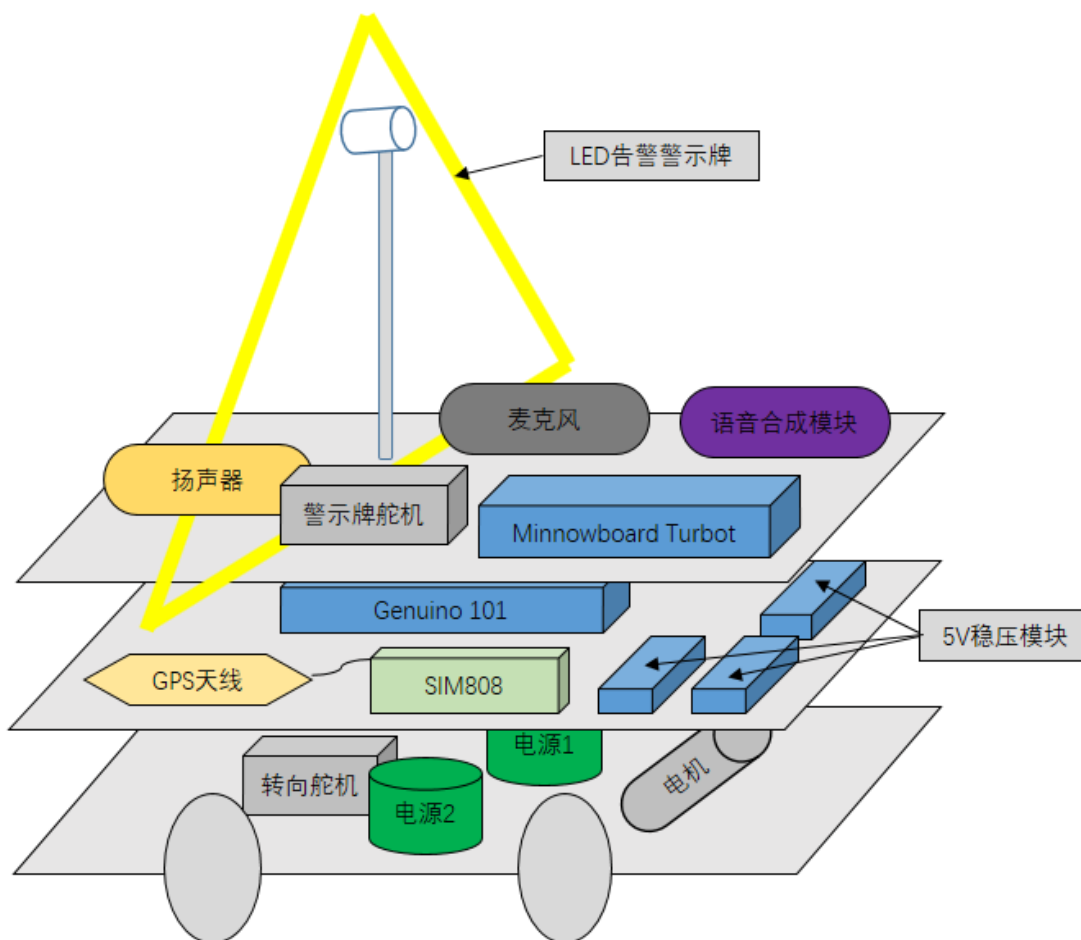


图3-1 硬件布局示意图

本系统的硬件部分可以根据主控板分为两个部分。Minnowboard Turbot 板计算能力较强，用来完成主程序，与摄像头、语音合成模块、Genuino 101 开发板、Sim808

模块连接；Genuino 板的嵌入式控制能力较强，因此连接舵机、电机。硬件连接如图 3-2 所示。下面对个硬件的功能和作用进行详细介绍：

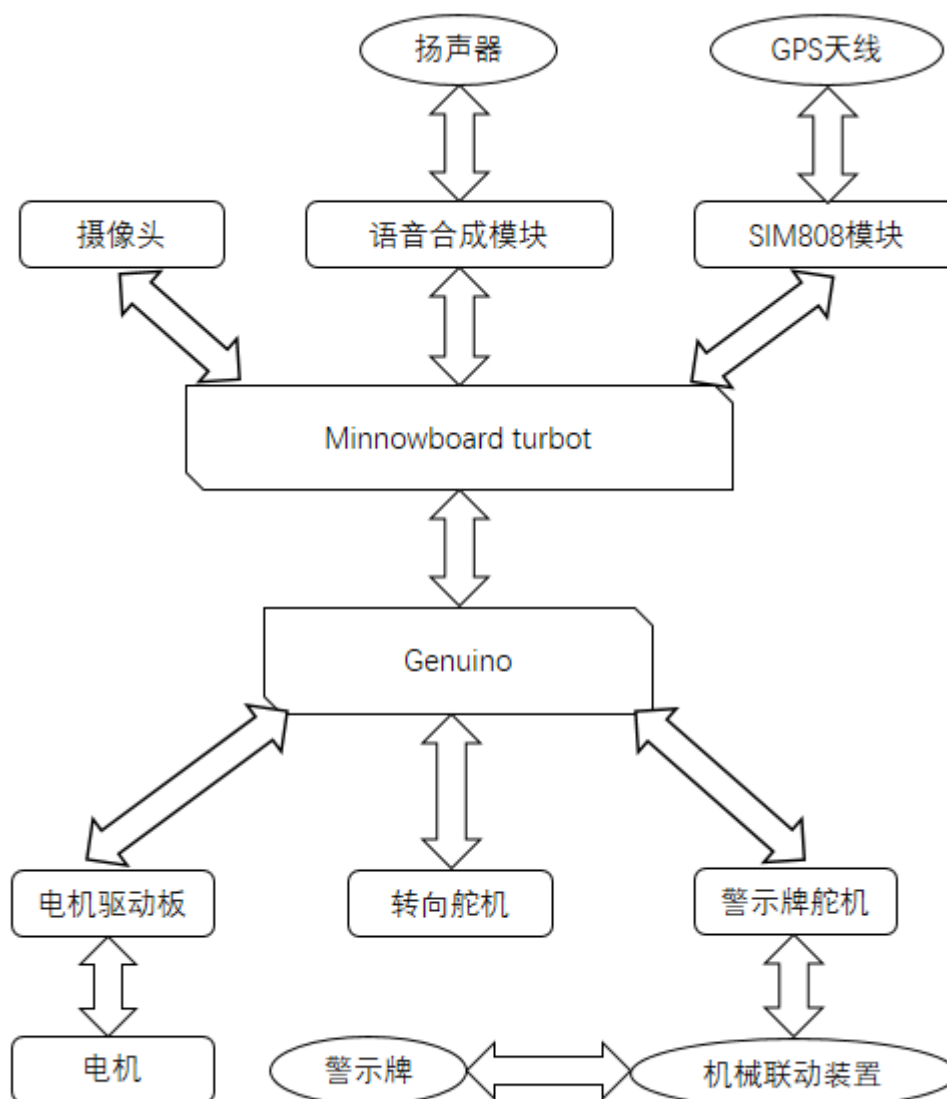


图3-2 系统硬件连接整体示意图

- (1) 语音合成模块：Minnowboard Turbot 与其使用 USB-TTL 进行连接，Minnowboard Turbot 向其发送 16 进制的 GBK 编码，外接扬声器将合成好的语音播放出来。
- (2) 摄像头：固定在机器人最高的位置，用于采集待处理的图像，经 USB 串口与 Minnowboard Turbot 交换信息。
- (3) GPS、GSM 集成模块：集成了 GPS，GPRS 等功能，外接 GPS 天线可以接收 GPS 信息，同时具有接打电话、收发短信功能，经 USB 串口与 Minnowboard Turbot 交换信息。
- (4) 5V 稳压模块：输入电压为一个足够大的电压区间，输出电压经过调节变压器控制在 5V，用于给 Minnowboard Turbot，电机驱动板，两个舵机供电。
- (5) 电机驱动板：用于接收 Genuino 发出的 PWM 波来控制电机的转速及方向。
- (6) 电机：电机为直流电机。
- (7) 舵机：转向舵机和警示牌舵机都是通过 PWM 波由 Genuino 101 直接控制。转向舵机控制机器人转向，警示牌舵机通过机械联动装置控制警示牌的起落。
- (8) 机械联动装置：机械联动装置为典型四杆结构，如图 3-3 所示。

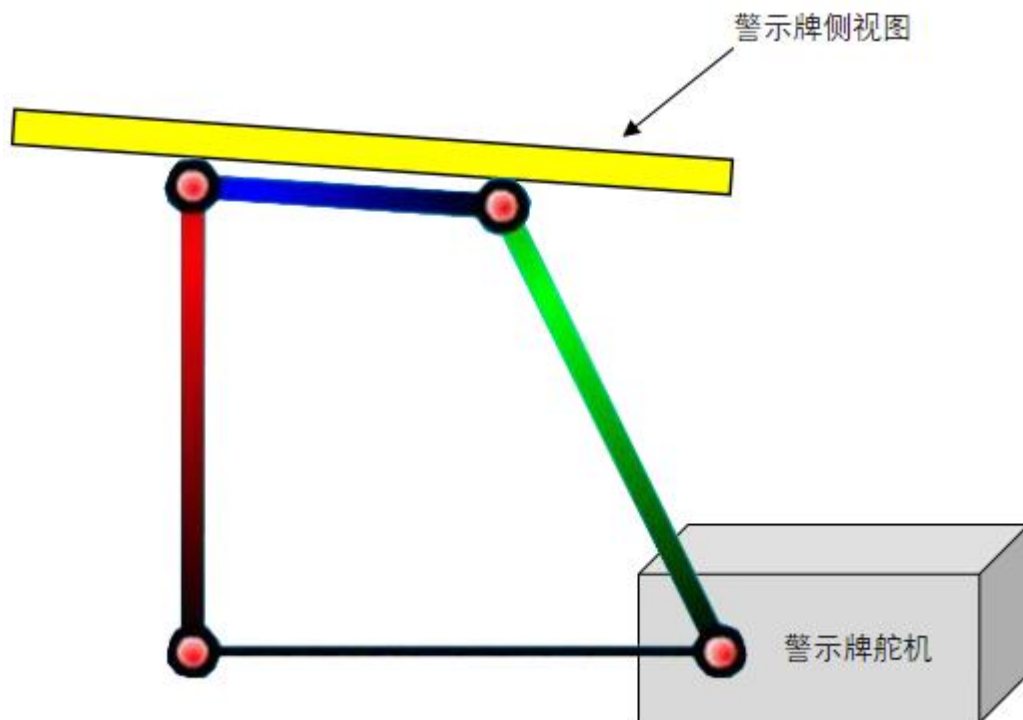


图3-3 四杆结构示意图

3.1.2 系统硬件的控制部分的设计与实现

硬件系统的控制部分集中在 Genuino 101 开发板上。首先，Genuino 101 开发板通过串口接收到 Minnowboard Turbot 平台的控制命令，同时还要回传相应的数据，需要设计一个基于串口的通信协议。其次，Genuino 101 开发板通过操作 GPIO 接口控制电机和舵机，并且控制开发板内部的陀螺仪模块。

在串口通信协议的设计上，要具有一定的容错性。由于串口通信的偶发不稳定性，接收端可能将偶然的脉冲捕捉，形成非法字符，干扰系统运行，这就要求系统在正常运行情况下接收到非法字符后，本条指令可以作废，但不能影响后续指令的识别，否则将导致死机。本系统中，Genuino 101 和 Minnowboard Turbot 平台间的通信主要有三类指令：速度指令，方向指令，倒车指令，外设模块指令等等。这些指令可以全部用小写英文字母表示。

因为串口错误字符被误认为字母的可能性极小，因此，可以认定除字母外的其它一切字符均为指令结束符。如果正常传输，系统使用换行符‘\n’终结。如果出现非法字符，则本次传输立即结束，本条指令被作废，由 Minnowboard Turbot 在下一个周期进行重发，不会导致死机。

所以，要对串口信息进行灵活判断。在 Genuino 101 代码的每次循环中，首先获取陀螺仪的三轴信息，然后与上一次的结果进行差值对比，然后判断差值是否过大，如果差值过大，则认为机器人已经翻倒，需要通过串口告知主板，如果不过大，则接收串口信息并且对串口信息进行判断并处理，处理过程要兼顾灵活性和健壮性，如果不符合预设的指令中任何一条，则要将相应指令及时丢弃，如果收到了正确指令，要对指令进行正确处理，处理后将收到的串口信息回发给 Minnowboard Turbot。

对于 GPIO 部分, Genuino 101 开发板通过 GPIO 口控制电机和舵机, 通过输出 PWM 波给电机驱动板间接控制电机, 通过输出 PWM 波给舵机直接控制舵机, 为了保证供电稳定和电路安全, 电机和舵机的供电均不使用板载电源。

3.1.3 硬件的整合

各个硬件模块中摄像头和 Minnowboard Turbot 使用 USB 接口直接进行连接; 语音合成模块和 SIM808 模块使用 USB 转串口模块连接到 Minnowboard Turbot, Minnowboard Turbot 与 Genuino 101 使用 USB 进行连接, 为解决 Minnowboard Turbot 的 USB 口不够用的问题, 外接一个 HUB 集线器以扩展 USB 接口。

对于 Genuino 101 控制的部分, 都是通过 GPIO 对各种外设进行控制。

3.2 系统算法设计

3.2.1 Minnowboard 主程序

程序流程图如图 3-4 所示。

具体流程:

(1) 从本地初始化数据: 从本地初始化用户信息、控制信息等。

(2) 道路线识别: 基于 OpenCV 的道路线识别算法, 识别出当前行驶车道的道路线。

(3) 行车控制: 根据识别出来的道路线, 对护航机器人的位置信息进行判断, 并根据控制算法, 对行驶方向、速度进行控制。

(4) 外设监控: 对 GPS 和电话模块, 对 Genuino 模块等外设进行监控, 若发现断开, 则重新初始化。若有数据, 则读取。

(5) 检测到巨大加速度变化: Genuino 模块上的陀螺仪模块若检测到巨大加速度变化, 则通过串口通信的方式发送通知, 接收到此信息后, 进入测险报警过程。

(6) 拨打用户电话: 通过电话模块拨打用户电话

(7) 用户及时响应: 若用户及时挂掉电话, 则回到正常过程 (即下一步是读取 GPS 信息), 若

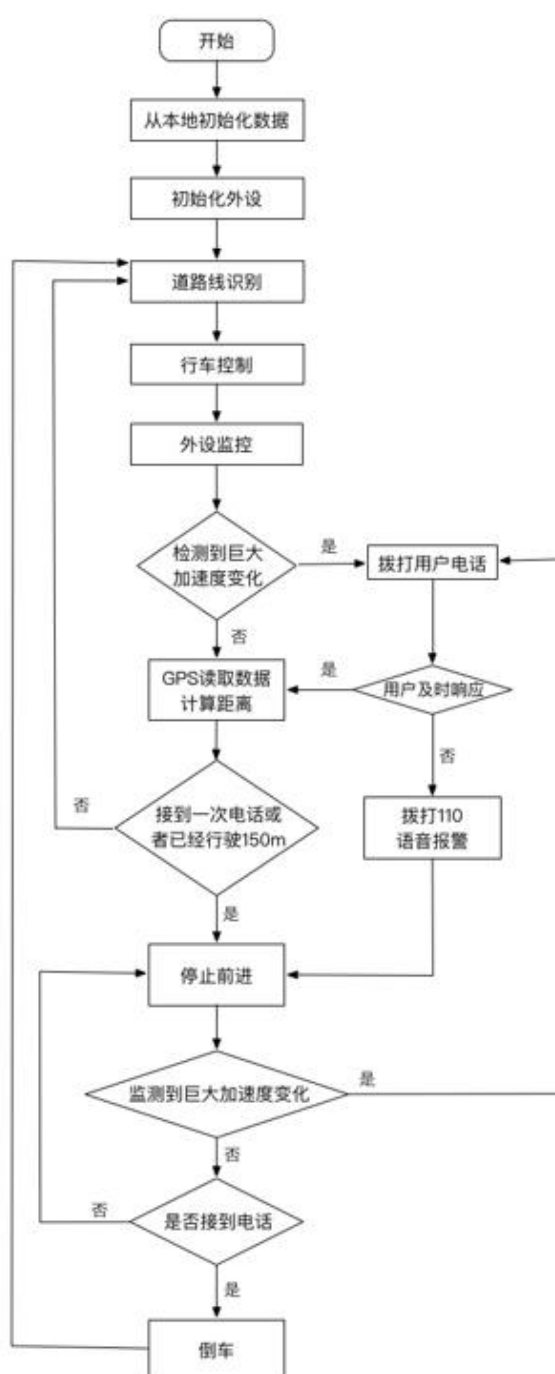


图3-4 Minnowboard程序流程图

用户未及时响应，则拨打 110 报警。

(8) 拨打 110 语音报警：通过电话模块拨打 110，通过语音模块播放声音，声音内容包含由 GPS 模块提供的经纬度。

(9) GPS 读取数据计算距离：使用 GPS 模块获取 GPS 信息，通过经纬度的变化计算行驶的距离。

(10) 接到一次电话或已走 150 米：通信模块接到电话后，会通知主程序，主程序记录下，转入停车状态；根据 GPS 信息计算出的距离，若大于等于 150 米，转入停车状态。

(11) 停止前进：停止行车，继续监视外设，检测是否有巨大加速度变化，等待第二次电话。

(12) 倒车：将小车行驶方向转变 180 度，利用 GPS 信息中的磁偏角辅助倒车过程。

3.2.2 道路线识别算法

3.2.2.1 道路线识别算法流程图

如图 3-5 所示。

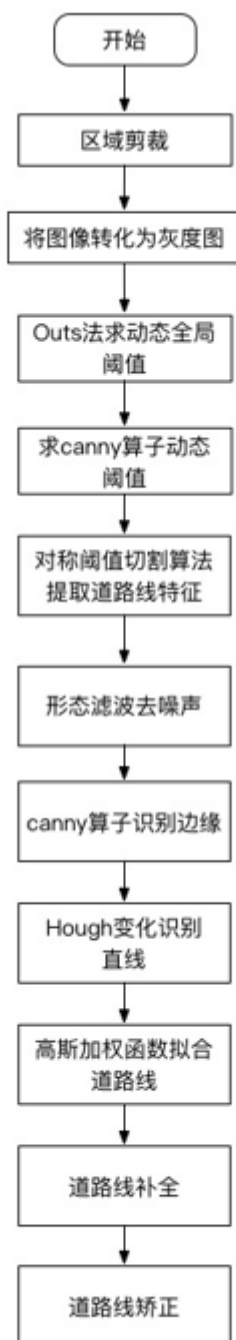


图3-5 道路线识别算法流程图

3.2.2.2 处理过程

- (1) 灰度化图像：对每一像素点的 RGB 值，以某种方式线性叠加，将 3 维信息降到 1 维。
- (2) 计算对称阈值切割算法中使用的全局阈值以及 canny 算子中使用的高低阈值
- (3) 对称阈值切割：利用图像信息计算出的全局阈值和道路先验，进行道路线特征提取。
- (4) canny 算子识别边缘

识别后的图像：



图3-6 识别后的图像

(5) hough 变换识别直线

(6) 高斯加权函数拟合道路线：经过 hough 变化识别直线后，会识别出多条直线，利用高斯加权函数从多条直线中拟合出道路线。

(7) 道路线补全

(8) 道路线校正

处理后图像(与图图 3-6 的三张图片依次对应)

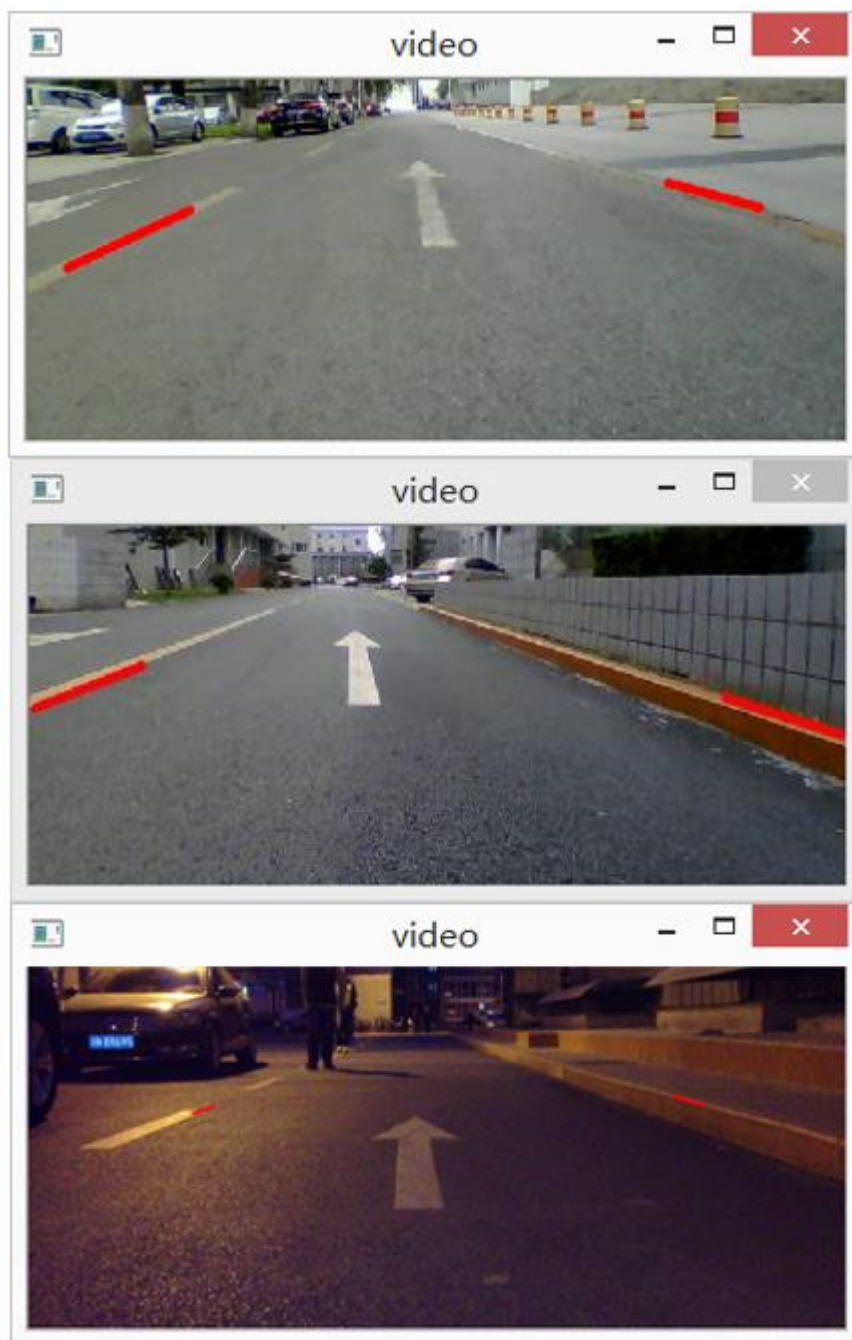


图3-7 处理后的图像

3.2.2.3 对称阈值切割算法

对称阈值切割算法利用了道路先验，利用了图像信息中计算出的全局阈值。其优点是能够去除掉许多噪声，将非道路线的直线特征滤去。另外，对由阴影的、破损的道路线和光线不足的阴天情况鲁棒性较好。

算法流程如图 3-8 所示。

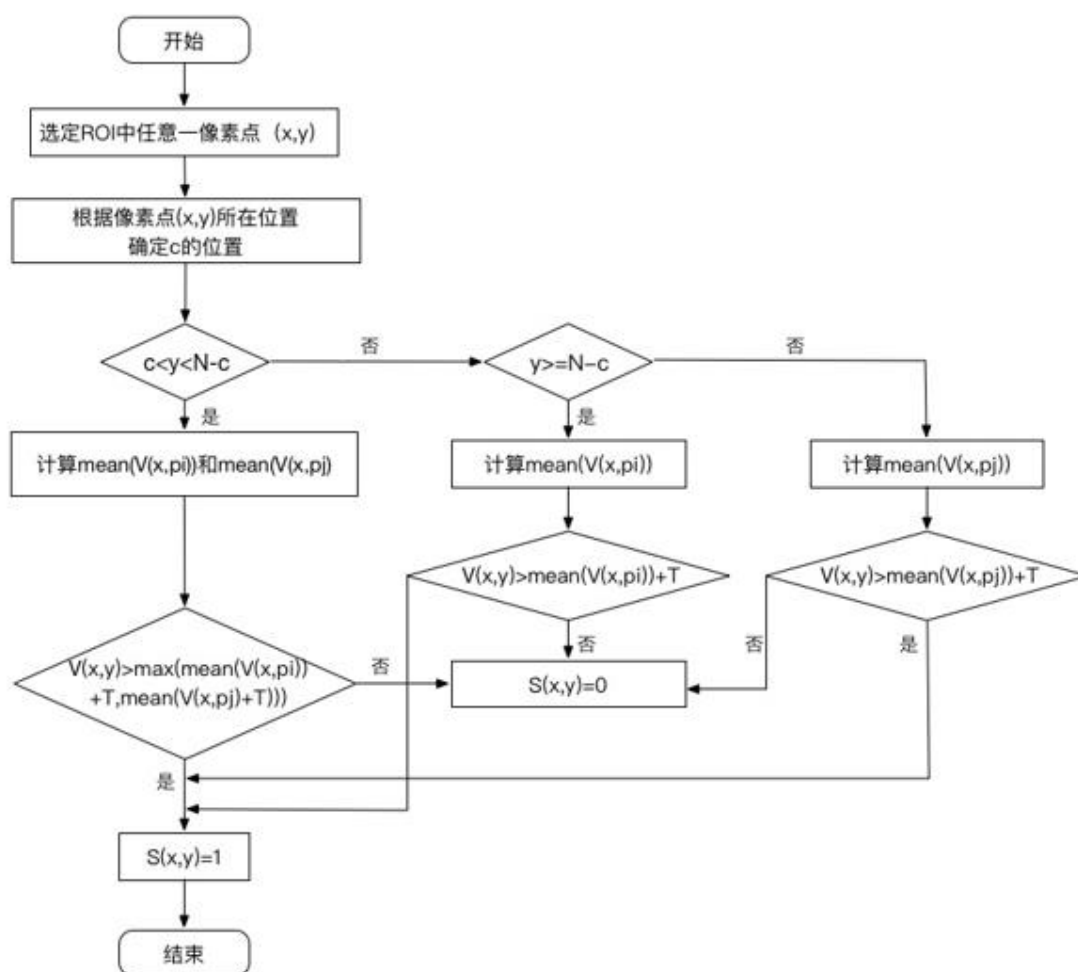


图3-8对称阈值切割算法流程图

注：ROI 为裁剪后的图像区域；T 是全局阈值；

$c = [k * (0.5 * L) / (M - 1)] + 0.5 * L$, $k = 0, 1, 2, \dots, M - 1$ (M 为图像像素高度)

3.2.3 语音合成模块

流程图如下：

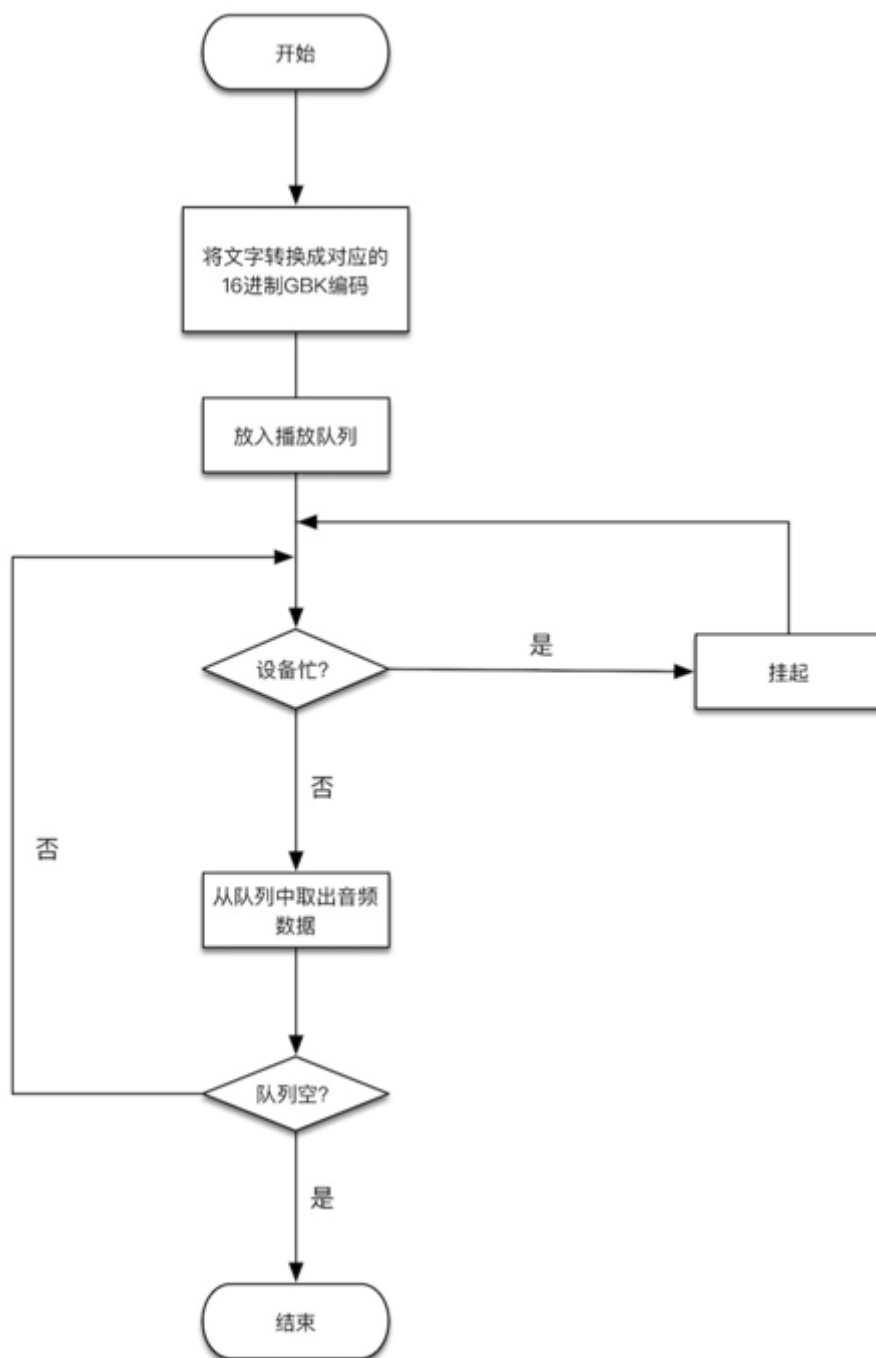


图3-9 语音合成模块流程图

3.2.4 GPS 模块算法

3.2.4.1 GPS 通信协议

该协议为 NMEA 0183 2.0 版，此协议是为了在不同的 GPS 导航设备中建立统一的 RTCM 标准。下面仅介绍我们使用到了的部分：

推荐定位信息 (RMC)

\$GPRMC,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11>,<12>*hh

<1> UTC 时间，hhmmss(时分秒)格式

- <2> 定位状态, A=有效定位, V=无效定位
- <3> 纬度 ddmm. mmmm(度分)格式(前面的0也将被传输)
- <4> 纬度半球 N(北半球)或 S(南半球)
- <5> 经度 dddmm. mmmm(度分)格式(前面的0也将被传输)
- <6> 经度半球 E(东经)或 W(西经)
- <7> 地面速率(000.0~999.9 节, 前面的0也将被传输)
- <8> 地面航向(000.0~359.9 度, 以真北为参考基准, 前面的0也将被传输)
- <9> UTC 日期, ddmmyy(日月年)格式
- <10> 磁偏角(000.0~180.0 度, 前面的0也将被传输)
- <11> 磁偏角方向, E(东)或 W(西)
- <12> 模式指示(仅 NMEA0183 3.00 版本输出, A=自主定位, D=差分, E=估算, N=数据无效)

3.2.4.1 算法流程图

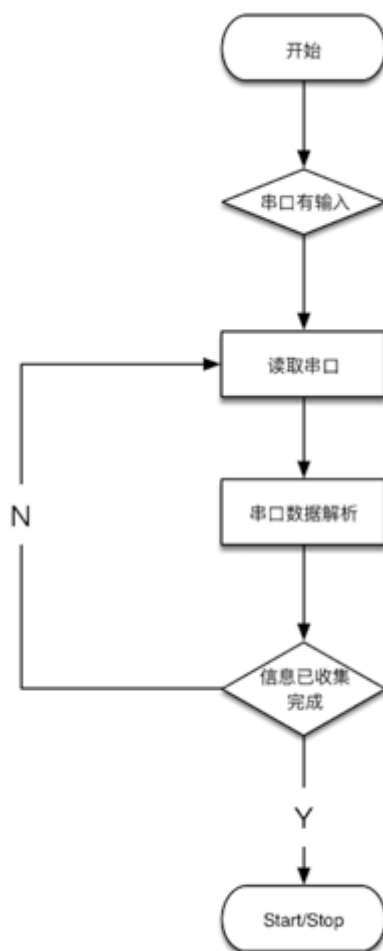


图3-10 GPS算法流程图

串口数据分析基于自动机模型, 与编译器的实现的相同。接受串口数据输入, 根据当前状态, 对数据进行解析、保存, 并转移到下一个状态, 直到所有信息收集完毕转移到终止态。

3.2.5 Genuino101 算法

算法流程图如下：

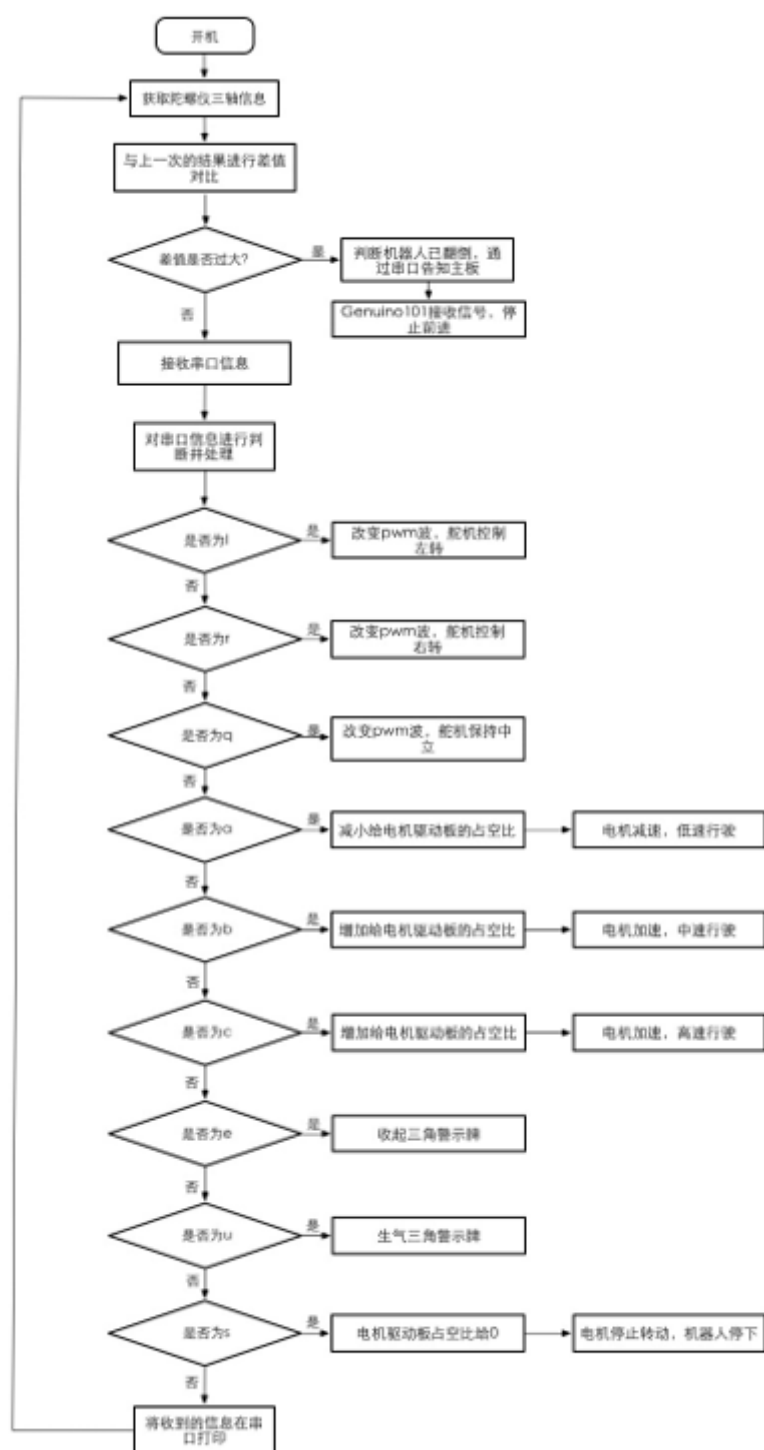


图3-11 Genuino 101算法流程图

Genuino101 的算法实现部分主要分为两个模块，一个是三轴陀螺仪模块，一个是逻辑算法模块。

在三轴陀螺仪模块主要是调用了 Arduino 的 CurieIMU 库，此库包含了对 Genuino101 板中自带的 BMI160IMU 模块的驱动，此模块可以通过传感器读取自身的三轴姿态角。通过相邻两次的三轴姿态角的差值即可确定在短时间内机器人是否有巨大的晃动。若差值较小，则将当前的陀螺仪读数覆盖到上一次的读数中，并继续读取下一次

结果，重复比较过程。一旦此差值较大，则极大可能是机器人已经被撞翻或驶向了非正常道路。 所以就会立刻拨打用户的电话，发出警告。

逻辑算法模块主要实现的是对串口数据的读写。当 Genuino101 开发板从串口读入数据时，判断是否为控制命令，如“r”命令表示右转，“l”命令表示左转 等。如果不是控制命令，则进入下一次循环。如果是控制命令，则对相应的控制部件进行控制。控制命令如下：

表 3-1 控制命令表

命令	功能
l	向左转
r	向右转
a	以低档速度前进
b	以中档速度前进
c	以高档速度前进
d	倒车
e	升起停车警示牌
u	收起停车警示牌
s	停车
q	直行

3.3 扩展外设的设计实现

对于 Minnowboard Turbot 来说，有以下几个扩展外设：Genuino 101 模块、电机驱动模块、稳压模块、警示牌升降模块、舵机转向模块、GPS 模块、语音播报模块。

下面将依次介绍上述模块。

3.3.1 Genuino 101 模块

Genuino 101 模块是一个十分适合硬件扩展的模块，如图所示。

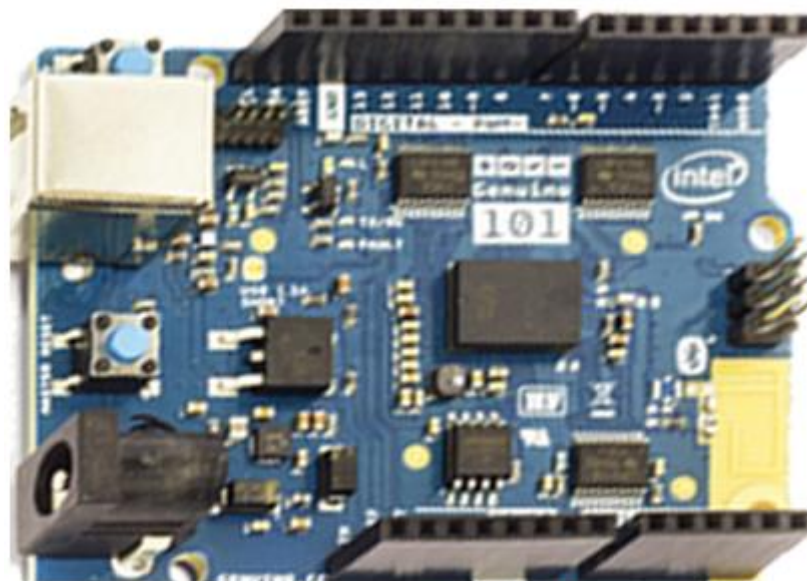


图3-12 Genuino 101模块

在 Genuino101 开发板上，使用了 Arduino 进行开发，主要通过 USB 转串口从主板 Minnowboard Turbot 获得控制信息，然后在本地找到相应的外设模块发出信号进行控制。控制完成后，会向串口写入得到的消息，由此告知主板 Minnowboard Turbot，此次操作已经完成。Genuino101 开发板的供电口和串口是分开的，因为若合在一起则可能出现所供功率不够的情况。

在控制舵机和电机方面，Genuino101 开发板主要是通过库函数 Servo 对舵机进行控制，其优点是控制简单，方便读和写数据。缺点是只能用开发板上的几个特定的标有波浪线“~”符号的 GPIO 口产生 PWM 波，选取其中两个 GPIO 口连接电机的驱动板，再选取两个分别连接两个舵机的信号线。在电机的控制上，是根据获取的不同速度指令，从 GPIO 口给电机驱动板发出不同的占空比，从而实现速度的调节。

3.3.2 电机驱动模块

电机驱动模块一个有 5 个引脚，分别是 VCC, GND, EN, 和两个信号线。电机驱动板的 VCC 和 GND 由 Genuino101 供电，是电机驱动模块和 Genuino101 开发板实现共地。使驱动板正常工作并且用一个普通 GPIO 口连接电机驱动板的使能端口 EN，需要使用电机驱动时，发出高电平使其有效即可。再选取 Genuino101 两个专门带有波浪线“~”的 GPIO 口连接电机驱动板的信号线。两根信号线不可以同时都给出占空比，任何时刻都要保证其中一根信号线的占空比为 0。当为 0 的信号线反向后，电机反向转动。

驱动板需要接收 Genuino 101 开发板发出的 PWM 波，根据获取的占空比调节供给电机的输出电压，电机在不同电压下的转速不同，在额定范围内，转速随电压升高而增加。

3.3.3 稳压模块

稳压模块使用 LM2596S 模块，如图所示。

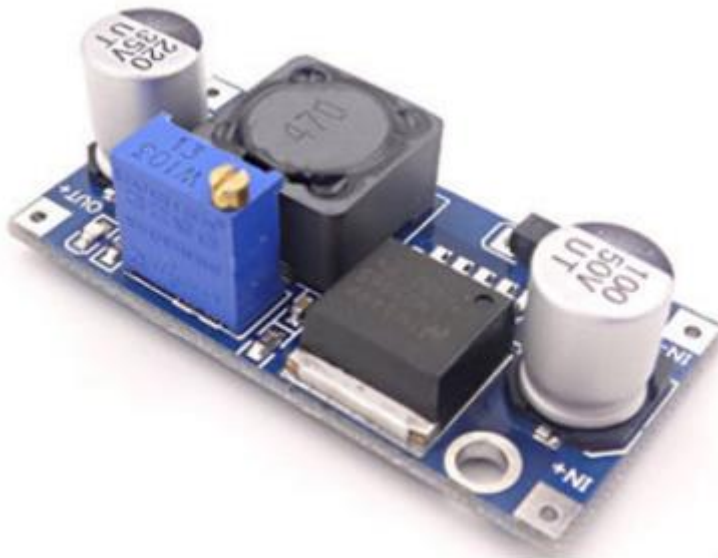


图3-13 稳压模块

稳压模块能用 12.7 的电池作为输入端，经过调整电位计后，使得输出电压稳定在 5.0V，给舵机和主办的散热风扇进行供电。稳压模块的最大支持电流为 3A，最大功率能达到 15W，其精度保证输出误差在正负 4%以内。

本机器人的稳压模块一共有 3 个，均被调节成 5V 稳压模块。每个的功率都要求在 10W 以下，否则稳压模块会过热。给舵机供电的稳压模块功率需求较低，只需要一个。给 Minnowboard Turbot 需要 5V 3A 的电源，使用一个 5V 稳压模块会导致稳压模块过热，所以由两个相同规格的 5V 稳压模块并联平分电流来平分功率，这样可以大大缓解稳压模块发热的情况。

3.3.4 警示牌升降模块

警示牌升降模块的机械核心是四杆结构，控制核心是警示牌舵机。警示牌舵机由 5V 稳压模块直接供电，由 Genuino 101 开发板的 PWM 波控制。Genuino 101 开发板通过发出不同占空比的 PWM 波，控制警示牌舵机旋转的角度，进而带动四杆结构，实现升起、降落警示牌的功能，如图 3-3 所示。

3.3.5 舵机转向模块

转向模块是由舵机带动连杆，从而控制左右两个轮子同时向左或者向右转。由于机械结构的限位，转向舵机的 PWM 波信号也控制在合理的范围内。首先需要对转向舵机的中间位置进行确定，然后将中间位置设置为机器人直走的方向，最后通过控制将机器人左转和右转的幅度变为一致。

3.3.6 GPS+GSM 模块

GPS 和 GSM 模块集成于 SIM808 芯片上，如图所示。



图3-14 SIM808模块

SIM808 芯片外接的 GPS 天线可以接收 GPS 信息，经 USB 串口与 Minnowboard Turbot 进行交互。

3.3.7 语音合成模块

语音播报模块为 SYN6288 模块，如图所示。



图3-15 SYN6288模块

此模块经 USB 串口与 Minnowboard Turbot 交换信息，Minnowboard Turbot 以 16 进制 GBK 编码向语音合成模块发送需要播报的内容，语言合成模块通过扬声器将内容播报。

3.4 护航机器人实物效果

整个系统样机的实物图如图 4-XX，机器人为车型机器人，底板为弹簧减震底板，能够适应凹凸不平的路面，使得摄像头的采集效果稳定。摄像头将采集到的数据通过 USB 接口传送给 Minnowboard Turbot 并且被暂存。摄像头虽然被安装在最高的位置，但是警示牌竖起时不会撞到摄像头。在机器人行进过程中，用户可以拨打机器人的电话对机器人进行召回。各个模块上自带的指示灯足以判断各个模块的工作状态。

用户紧急停车后，将机器人放在车道中间，按要求接上电源，机器人就会开始正常启动。当系统成功启动，所有模块均被上电，各个模块指示灯全部正常发光，可以以此判断系统是否启动完成，如图 3-16 所示。

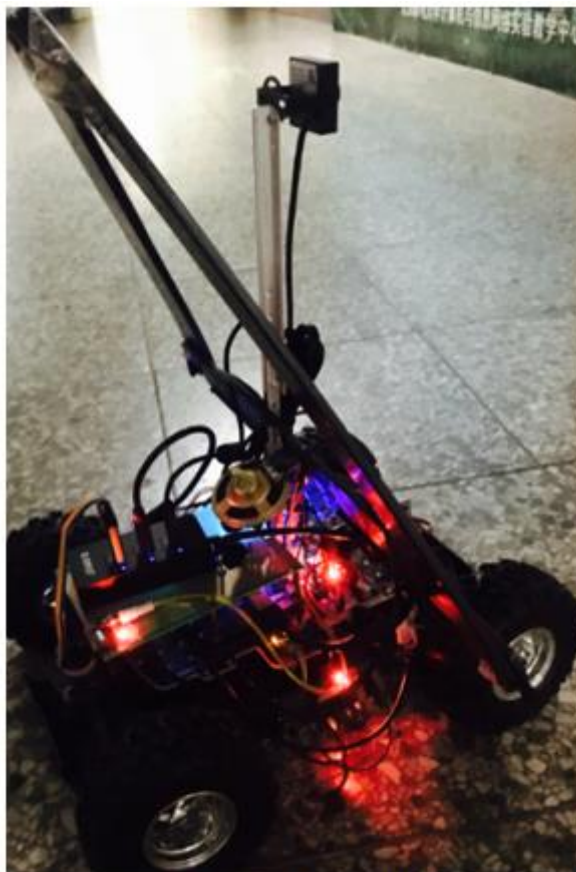


图3-16 启动后的机器人

系统成功启动后，警示牌会自动竖起来，并且机器人会举着警示牌根据检测到的道路路线自动行进，因为汽车所停的道路可能是弯道，而且最初的前进方向并不一定是平行于道路的方向，所以机器人行进的路线是动态调整的。这种动态调整会使机器人在车道内不断微调方向，防止机器人跑出车道。警示牌自动竖起来的优势在于，在机器人行进过程中也可以保持警示状态，这方面相对于人手动放警示牌的行进过程而言更加科学，且更加安全。

当机器人走到预设的距离，就会停止行进，并且保持警示牌竖立的状态，如图 3-17 所示。



图3-17 机器人竖起警示牌

当机器人遇到意外，比如被其他车子撞翻，就会使 Genuino 101 上面的陀螺仪给出意外信号，当 Minnowboard Turbot 接收到此意外信号，就会给 SIM808 模块发出拨打用户电话的指令，如果用户没有发生事故，可以在 10 秒钟之内挂机，则机器人忽略此次意外信号；如果用户 10 秒内没有挂机，说明用户发生了危险(比如已经昏迷)，那么机器人就会给 110 拨打求救电话，从 SIM808 模块的 GPS 部分获得位置信息，转成 GBK 格式的 16 进制编码，传输到语音合成模块，语音合成模块通过扬声器给接通后的 110 电话汇报位置信息，例如：发生事故，经度 116.3715，纬度 39.9263。经纬度精确到小数点后四位，十几米的精度足以帮助警方发现事故现场。

如果一切正常，用户结束紧急停车，给机器人拨打电话，机器人就会进行 180° 的转弯然后原路返回。考虑到紧急停车道的宽度可能无法满足机器人的转弯，机器人的转弯其实是转动一定角度后进行倒车，倒车一段距离后继续进行转弯，这样反复多次，可以在有限的道路宽度进行 180° 的转弯。在原路返回过程中，机器人的行进方向同样是根据道路线检测得到的结果而动态调整的，如图 3-18 所示。



图3-18 机器人转向过程

第四章 系统测试

系统测试分为硬件测试、软件测试、联合测试。硬件测试主要针对警示牌和舵机相连的四杆结构部分，这是比较关键的机械联动装置。需要测试 Minnowboard Turbot 和 Genuino 101 的串口通信功能。另外，还需单独测试语音合成模块，SIM808 模块等需要串口通信的模块。软件测试主要针对道路线识别算法是否合理，边界情况，算法的正确性和运算时间以及应急情况的处理速度。联合测试主要测试系统整体运行，各模块之间的通信、接口是否出现问题并及时调整。

4.1 硬件测试

4.1.1 四杆结构测试。

(1)测试内容：测试四杆结构是否可以正常升降警示牌。在不影响摄像头的情况下(不可以撞击到摄像头(可以实现警示牌的正常升起，可以实现警示牌的正常降落(同样的，不能撞击到摄像头(，测试完成后，舵机的角度不错位。

(2)测试方法：让四杆结构交替升起、降落警示牌 10 次，观察每次起落的结果，以确保四杆结构可以正常升降警示牌。

(3)测试环境：自然环境为普通的道路环境，微风。软件环境为 Genuino 101。

(4)预期结果：四杆结构经过警示牌舵机控制可以正常升降警示牌，并且不会影响到摄像头。

(5)实际结果：十次测试中，警示牌均被升起、降落到位，并且没有撞击到摄像头；并且经过十次测试后，舵机的角度并未错位。

(6)测试结论：四杆结构是否可以正常升降警示牌。

4.1.2 Genuino 101 开发板的串口控制功能

(1)测试内容：测试 Genuino 101 开发板的 USB 串口是否可以与 Minnowboard Turbot 进行双向收发数据。

(2)测试方法：改编 Arduino IDE 中自带的串口通信例程，分别用 Arduino 中的串口监视器、Minnowboard Turbot 中运行的串口调试助手(Minicom(进行测试。测试程序功能为 Minnowboard Turbot 通过串口向 Genuino 101 发送字符串，Genuino 101 接收到串口信息后向 Minnowboard Turbot 回发此字符串并且在字符串末尾加上”ok”。

(3)测试环境：Minnowboard Turbot + Ubuntu 14.04，Arduino IDE。

(4)预期结果：Minnowboard Turbot 通过串口向 Genuino 101 发送字符串，Genuino 101 向 Minnowboard Turbot 回发此字符串并且在字符串末尾加上”ok”,测试数据及对应的预期结果如下表所示。

表 4-1 串口测试数据表

Minnowboard Turbot 发送数据	Genuino 101 回发数据
Hello	Hellook

World	Worldok
ok	okok
ko	kook
111	111ok

(5)实际结果：世界接轨与预期结果基本一致，Minnowboard Turbot 发送任意字符串，Genuino 101 均会回发对应的加工后的字符串。

(6)测试结论：Genuino 101 开发板的 USB 串口可以与 Minnowboard Turbot 进行双向收发数据。

4.1.3 语音合成模块测试

(1)测试内容：测试语音合成模块是否可以正常完成语音合成的工作。

(2)测试方法：将 5 句由数字、汉字、小数点等组成的中文翻译成 16 进制的 GBK 编码，并且通过串口调试助手发送给语音合成模块，观察语音合成模块是否可以通过扬声器发出预期的声音。

(3)测试环境：Minnowboard Turbot + Ubuntu 14.04

(4)预期结果：语音合成模块可以通过扬声器发出 5 句中文对应的声音。

(5)实际结果：测试数据如下表所示。

表 4-2 语音模块测试数据

编号	测试用例	16 进制 GBK 编码
1	经度：10122.2222	0xFD 0x00 0x13 0x01 0x01 0xBE 0xAD 0xB6 0xC8 0xA3 0xBA 0x31 0x30 0x31 0x32 0x32 0x2E 0x32 0x32 0x32 0x32 0x84
2	发生事故，请求帮助	0xFD 0x00 0x15 0x01 0x01 0xB7 0xA2 0xC9 0xFA 0xCA 0xC2 0xB9 0xCA 0xA3 0xAC 0xC7 0xEB 0xC7 0xF3 0xB0 0xEF 0xD6 0xFA 0xD1
3	纬度：22.2222 发生事故	0xFD 0x00 0x18 0x01 0x01 0xCE 0xB3 0xB6 0xC8 0xA3 0xBA 0x32 0x32 0x2E 0x32 0x32 0x32 0x32 0xB7 0xA2 0xC9 0xFA 0xCA 0xC2 0xB9 0xCA 0x8C
4	1234.4321	0xFD 0x00 0x0C 0x01 0x01 0x31 0x32 0x33 0x34 0x2E 0x34 0x33 0x32 0x31 0xDF
5	请求支援	0xFD 0x00 0x0B 0x01 0x01 0xC7 0xEB 0xC7 0xF3 0xD6 0xA7 0xD4 0xAE 0xE5

语音合成模块可以通过扬声器发出 5 句中文对应的声音，与预期结果一致。

(6)测试结论：测试语音合成模块可以正常完成语音合成的工作。

4.1.4 SIM808 模块测试

(1)测试内容：测试 SIM808 模块是否可以完成正常的接打电话、收发短信及是否可以按照格式要求返回 GPS 数据。

(2)测试方法：使用 SIM808 模块的 AT 指令逐一对 SIM808 模块的各个功能进行单独测试。

(3) 测试环境: Minnowboard Turbot + Ubuntu 14.04

(4) 预期结果: SIM808 模块可以通过相应 AT 指令控制完成正常的接打电话、收发短信功能,使用 AT 指令将 GPS 电源开关和数据开关打开后,GPS 模块源源不断地以 nmea0183 协议向串口发送 GPS 信息,测试数据如下表所示。

表 4-3 SIM808 模块测试

功能	AT 指令
发送短信	AT+CSCS=" GSM" AT+CMGF=1 AT+CMGS=" 10086" >HaHa 1A(十六进制发送(接收短信 AT+CMGR=1 打电话 ATD10086; 接电话 ATA GPS AT+CGNSPWR=1 AT+CGNSTST=1

(5) 实际结果: 使用 10086 进行测试,模块可以正常收发短信和拨打电话,当使用手机向 SIM808 模块拨打电话的时候,模块可以接电话。当使用 AT+CGNSPWR=1 和 AT+CGNSTST=1 命令打开 GPS 部分的电源开关和数据开关,模块开始向串口以 nmea0183 协议向串口源源不断地发送 GPS 信息,直到执行 AT+CGNSTST=0 或者 AT+CGNSPWR=0 指令中止发送。

(6) 测试结论: SIM808 模块可以完成正常的接打电话、收发短信及可以按照格式要求返回 GPS 数据。

4.2 软件测试

4.2.1 道路线识别算法测试

(1) 测试内容: 实际道路下道路线识别的准确率和速度

(2) 测试方法: 护航机器人多处实际道路视频采样,采样后作为识别的算法的输入,观察道路线识别的情况

(3) 测试环境: Ubuntu14.04

(4) 输入图像与识别情况:



图4-1 识别情况

(5) 效果分析：分别在白天和夜间进行了五次测试，每次测试约为 10 分钟，实验的统计数据如下：

白天：

表 4-4 白天道路线识别算法测试

实验编号	测试时间(秒)	帧数	平均每帧时间(秒)
1	600	137	4.37
2	602	133	4.53
3	600	133	4.51
4	601	133	4.52
5	600	136	4.41

夜间：

表 4-5 白天道路线识别算法测试

实验编号	测试时间(秒)	帧数	平均每帧时间(秒)
1	602	134	4.49
2	601	133	4.52
3	600	130	4.62
4	600	131	4.58
5	601	134	4.49

可见，无论是在白天还是夜间，道路线识别算法都能有较稳定的输出，识别的速率约在 4.5 帧每秒。

4.2.1 Genuino 101 代码整体测试

(1)测试内容：测试 Genuino 101 开发板是否可以根据串口指令完成指定的任务和功能。

(2)测试方法：使用 Minnowboard Turbot 通过串口向 Genuino 101 开发板通过发送一系列设计好的指令，包括一些不正确的干扰指令，观察 Genuino 101 是否可以正常、稳定地工作。指令集及其实现的功能如下表：

表 4-6 指令集测试

命令	功能
l	向左转
r	向右转
a	以低档速度前进
b	以中档速度前进
c	以高档速度前进
d	倒车
e	升起停车警示牌
u	收起停车警示牌
s	停车
q	直行
其他	非法指令

(3)测试环境：Minnowboard Turbot + Ubuntu 14.04

(4)预期结果：测试数据如上表，测试数据中，非法指令 1 和非法指令 2 无回显，其他指令均正常执行。

(5)实际结果：与预期结果基本一致，各个指令都执行成功，并且实现了预期的功能，而且非法指令 1 和非法指令 2 都没有回显，并且非法指令后面的指令都能正常执行，达到了容错的目的。

(6)测试结论：Genuino 101 可以和 Minnowboard Turbot 平台进行正常串口通信，并且 Genuino 101 的串口接收功能性能稳定，可以满足主程序运行过程中的各种指令需求，并且可以进行有效的控制。

4.3 联合测试

4.3.1 测试环境

硬件环境：

Minnowboard Turbot 平台

Genuino 101 开发板

高清网络摄像头

电机驱动板

语音合成模块

GPS+GSM 模块(SIM808 模块)

5V 稳压模块

5.3.2 测试内容

(1)程序及系统框架

系统能够执行所有模块，与之相关的数据结构需要被成功构建，Minnowboard

Turbot 平台通过串口发送的指令 Genuino 101 和其他外设能够正确处理。

(2) 机器人竖立自动行进到目的地

在给机器人打电话让其提前停止行进的状况下，机器人需要顺利完成竖立警示牌，并且自动行进到目的地的任务。

(3) 通过给机器人打电话让机器人停止前进

在机器人的行进过程中，给机器人打电话，机器人在接到电话后会停止前进，并且保持竖立牌子的状态。

(4) 通过给机器人打电话让机器人原路返回

机器人在非行进状态下，接收到电话，能够顺利原路返回。

(5) 通过将机器人推倒启动报警流程

将机器人推倒，陀螺仪的中断需要被触发，机器人给用户的手机打电话，若接通电话则取消报警，若十秒内未接通电话则报警，报警时可以正确播报 GPS 信息及发生事故等提示语句。

4.3.3 测试结果与结论

(1) 程序及系统框架

机器人在测试中的表现均与预期情况基本吻合，Minnowboard Turbot 平台通过串口发送的指令 Genuino 101 和其他外设能够正确处理，测试成功。

(2) 机器人竖立自动行进到目的地

系统能够正确的获取道路线的图像信息，确认摄像头前无遮挡后，摄像头能够正常拍摄道路，并进行识别与存储，同时指导机器人行进的方向，机器人最终顺利到达目的地，测试成功。

(3) 通过给机器人打电话让机器人停止前进

系统可以通过 SIM808 模块的串口顺利完成接打电话的任务，并且指导机器人停止前进，测试成功。

(4) 通过给机器人打电话让机器人原路返回

机器人在非行进状态下，接收到用户的电话，顺利原路返回，测试成功。

(5) 通过将机器人推倒启动报警流程

将机器人推倒，机器人则给用户的手机打电话，接通电话则取消报警，十秒内未接通电话则报警，报警时可以正确播报 GPS 信息及发生事故等提示语句，系统可以通过 SIM808 模块的 GPS 部分接收到 GPS 信息，并且正确解析，测试成功。

第五章 总结

做这个机器人的想法在比赛之前就已经萌生了。起因是其中一个队员的朋友在紧急停车道上停车，回头去放警示牌，在放警示牌的路上发生了车祸。于是，她把自己的想法和我们进行交流，恰逢 intel 嵌入式比赛这片肥沃的土壤，我们三个会心一笑，一个高度一体化、融合了机器人控制、机器视觉等技术的护航机器人就这样生长起来了。

机器人要有眼睛，摄像头站得最高；机器人要有嘴巴，扬声器被固定；机器人要有耳朵，麦克风加入了……就这样一点一滴的思路，不计月日的积累，机器人被逐渐完善，忘记是哪一天，因为希望那样的悲剧不再发生，想让机器人像天使一样保护停车人，于是，“护航天使”成为了我们讨论组的名字，最后，也变成了我们作品的名字。

青青园中葵，朝露待日晞。在我们的共同努力下，机器人像小苗一样茁壮成长，最终的丑小鸭终于长大了，最初的闪念，终于被具现化，队员们都抑制不住内心的喜悦，拥抱我们可爱的护航天使。

我们在思维发酵的过程中进行了以下几点创新：

- (1)道路线识别：如今的道路线识别算法基本是为自动驾驶汽车而设计的，算法利用了道路先验知识，摄像头高度约在 2 米左右，护航机器人的摄像头高度约为 40 厘米。由于情况差别过大，所以即使改变了算法中的参数，也无法达到很好的识别效果。护航机器人的道路线识别算法适于多种天气状况，并能接受由于摄像头距地面高度过小而带来的拍摄道路情况不完整，不清楚等问题。
- (2)出险检测与报警：陀螺仪检测巨大加速度变化，判断是否发生事故，同时通过电话模块向用户发出警示，若用户未及时响应，则拨打 110 报警，并播报当前 GPS 位置。
- (3)简单健壮的交互：用户与护航机器人通过电话和短信进行交互，简单方便。同时 2G 移动通信网覆盖广，质量稳定，奠定了交互的健壮性，适应复杂的行车情况。
- (4)在系统启动的时候竖立警示牌，使其在行进过程中也起到警示效果。

有限的时间，无尽的可能。除此以外，我们还跳出了很多更新的点子，这意味着还有更多更实用的功能可以体现在护航天使上：

如车道繁忙程度测试，可以考虑增加一个图像处理的线程，专门用来统计车流量，具体实现同样依赖 OpenCV 库。这样，当车流密集的时候，机器人就可以向用户提出警告，尽量减少临时停车时间。

如充分利用 2.4G wifi 模块实现局域网通信，这样用户和机器人之间就可以有更频繁的交互，控制功能就会更加丰满。

此外，护航天使还有很大的改进空间：

如摄像头高度的改进，为了适应各个高度的不同路宽的道路，摄像头的支撑杆将改进为可调高度的活动杆，针对较宽的道路可以提升摄像头的高度，从而满足摄像头对两侧道路线的充分捕捉。

如停车警示牌的固定改进：由于停车警示牌本身较轻，所以在本项目直接利用四杆机构将其升起。为了应对各种大风的天气状况，将对警示牌的支撑进行加固处理。

机器人防撞击改进：当机器人在紧急停车带逆行时被冲撞后，需要保持一段时间的正常功能，并且尽全力保留下被冲撞前的视频录像。故需要对主板 Minnowboard Turbot 进行最大化的

保护。考虑后期为主板加上缓冲材料，并再用坚固的外壳将其保护起来。

相信我们的小天使会在以后的日子里更加丰富，更加强壮！

谢辞

首先，感谢戴志涛老师在整个过程中对我们的关心、支持和指导。在作品设计，项目、人员安排，和人力、资金支持等等方面，戴老师给予了我们最大的帮助，在此对戴老师表示感谢！

感谢计算机学院的张俊喆学长和电子信息工程学院的罗子朦学长，你们丰富的嵌入式经验和卓越的想象力帮助我们解决了非常多的难题，没有你们，我们这一路不知道要多少艰辛。感谢电子信息工程学院的孔德伟同学，谢谢你分享的竞赛用板的使用经验。

同时，需要感谢两位 intel 工程师，李冬冬和李明。感谢二位在我们的竞赛用板出现问题时提供的帮助，谢谢！

最后，感谢曾经辛勤奋斗的我们自己。作品设计报告是我们辛勤工作的成果，但是时间的仓促以及我们自身水平的不足，报告中难免会遗留了些许问题。恳请阅读此篇作品设计报告的老师、同学，多予指正，不胜感激！

参考文献

- [1] 樊超,狄帅,侯利龙等.一种阴影及破损车道线识别方法研究[J].计算机应用研究,2012,29(10):3968-3971,3989.DOI:10.3969/j.issn.1001-3695.2012.10.098.
- [2] A. A. Assidiq, O. O. Khalifa, R. Islam, and S. Khan, “Real time lane detection for autonomous vehicles,” in Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on, 2008, pp. 82–88.
- [3] H. Li and F. Nashashibi, “Lane Detection (Part I): Mono-Vision Based Method,” 2013.
- [4] M.Fang, G.X.Yue and Q.C.Yu, “The Study on An Application of Otsu Method in Canny Operator”, in Proc.ISIP'09, 2009, pp. 109-112, Aug. 2009
- [5] Y.L.Zhang and L.Yan, “Edge Detection Based on Canny Adaptive Method” M.Sc. thesis, Norstwest University, Xi'an , China, Jun. 2009
- [6] K. L. R. Talvala, K. Kritayakirana, and J. Christian Gerdes. Pushing the limits: From lanekeeping to autonomous racing. Annual Reviews in Control, 2011.
- [7] Levinson, Jesse, et al. ”Towards fully autonomous driving: systems and algorithms.” Intelligent Vehicles Symposium, 2011.
- [8] Daniel Chi Kit Ngai and Nelson Hon Ching Yung, Senior Member,IEEE,”A Multiple-Goal Reinforcement Learning Method for Complex Vehicle Overtaking Maneuvers”,IEEE Transactions on Intelligent Transportation Systems, VOL. 12, NO. 2, JUNE 2011.
- [9] XUE Guoxin, SUN Yuqiang, SHI Guodong, Wang Yue,”One-lane Traffic Flow Simulation Model based on Imaginary Slope with Changing Tail Length”, IEEE 2011.
- [10] Wei SHI, Hua KUANG, Li-yun DONG and Shi-qiang DAI,” Characteristics of Lane Changing Induced by Bus Stop and Deceleration Area”,IEEE 2011,Fourth International Joint Conference on Computational Sciences and Optimization.
- [11] Jazar, Reza N., Mathematical Theory of Autodriver for Autonomous, Vehicles, Journal of Vibration and Control, 16(2), 253-279, 2010.

附录

Minnowboard Turbot 代码:

```
#include "intel.h"

#define NCC 10

using namespace std;
using namespace cv;

struct termio
{
    unsigned short  c_iflag;      /* 输入模式标志 */
    unsigned short  c_oflag;      /* 输出模式标志 */
    unsigned short  c_cflag;      /* 控制模式标志*/
    unsigned short  c_lflag;      /* local mode flags */
    unsigned char   c_line;        /* line discipline */
    unsigned char   c_cc[NCC];    /* control characters */
};

struct stateOfLine
{
    int line[3][4]; // 3 frame in a row, 2 parameters for each line:  b1, k ;  first
    left line ,and then right line
    double p[3][2];
    int point; //determine which one is the present one
    int num;
};

struct GPSInfo{
    double latitude;//jingdu
    double longtitude;//weidu
    char dla;
```

```
char dlo;
double angel;//jiaodu

bool completed;
};

//声明 IplImage 指针
IplImage* pCutFrame = NULL;
IplImage* pCutFrImg = NULL;
//声明 CvMemStorage 和 CvSeg 指针
CvMemStorage* storage = cvCreateMemStorage();
CvSeq* lines = NULL;
CvSeq* lines2 = NULL;
//裁剪的天空高度
int CutHeight = 210;
int CutWidth = 1;//只是为了实验的时候观察方便
//缩放
CvSize size;
double fScale =0.75;//0.7;//0.85
//车道线的实际像素宽度
int l1;
//离散高斯分布数组
double Gussian_left[20] = { 1.0000, 0.9802, 0.9231, 0.8353, 0.7261, 0.6065, 0.4868,
0.3753, 0.2780,
0.1979,0.1353,0.0889,0.0561,0.0340,0.0198,0.0111,0.0060,0.0031,0.0015,0.0007};/
/10 段
double Gussian_right[20] = { 1.0000, 0.9802, 0.9231, 0.8353, 0.7261, 0.6065, 0.4868,
0.3753, 0.2780, 0.1979, 0.1353, 0.0889, 0.0561, 0.0340, 0.0198, 0.0111, 0.0060,
0.0031, 0.0015, 0.0007 };// 0.01, 0.03, 0.034, 0.044, 0.05, 0.058, 0.063, 0.07,
0.077, 0.1,
double Gussian_backup[20] = { 1.0000, 0.9802, 0.9231, 0.8353, 0.7261, 0.6065, 0.4868,
0.3753, 0.2780,
0.1979,0.1353,0.0889,0.0561,0.0340,0.0198,0.0111,0.0060,0.0031,0.0015,0.0007};/
/10 段
int u1 = 4;//左车道 ~~~~~!!!!!!!!!!!!!!!!!!!!~
int u2 = 15;//右。。。
~~~~~!!!!!!!!!!!!!!!!!!!!~
double k1=100;//左车道初始斜率
double k2=-100;//右车道初始斜率

//for display
IplImage *temp1;
IplImage *temp2;
```

```
//for canny
CvMat* mat;
CvMat* mat1;

//1 放得是 y 值小的那个 2 是 y 值大的那个
CvPoint l_line1, l_line2;
CvPoint r_line1, r_line2;
struct stateOfLine staLine;

CvVideoWriter* writer = 0; //保存就加上这句

//time-record for state-change
QTime timespan;
QTime backspan;

//car's direction
enum directionState{ lEft, rIght, sTraight, bAck, sTop} ;
enum directionState carS = sTraight;

//ground truth for line of road
CvPoint gr_l_point1, gr_l_point2;
CvPoint gr_r_point1, gr_r_point2;

//for go back to where
bool prespot=false;
bool spot=false;
bool onlyOne=true;

//color area cut
int iLowH = 11;
int iHighH = 25;

int iLowS = 43;
int iHighS = 255;

int iLowV = 46;
int iHighV = 255;

//-----
int speed_arr[] = { B38400, B19200, B9600, B4800, B2400, B1200, B300,
                   B38400, B19200, B9600, B4800, B2400, B1200, B300, };
```

```
int name_arr[] = {38400, 19200, 9600, 4800, 2400, 1200, 300, 38400,
                  19200, 9600, 4800, 2400, 1200, 300, };

//-----for broadcast
char
sound_announcement[24]={0xFD, 0x00, 0x15, 0x01, 0x01, 0xBB, 0xA4, 0xBA, 0xBD, 0xBB, 0xFA,
0xC6, 0xF7, 0xC8, 0xCB, 0xD7, 0xD4, 0xB6, 0xAF, 0xB1, 0xA8, 0xBE, 0xAF, 0x91}; //护航机器人
自动报警
char sound_latitude[10]={0xFD, 0x00, 0x07, 0x01, 0x01, 0xBE, 0xAD, 0xB6, 0xC8, 0x97}; //
经度
char
sound_longitude[10]={0xFD, 0x00, 0x07, 0x01, 0x01, 0xCE, 0xB3, 0xB6, 0xC8, 0xF9}; //纬度
char
sound_accident[14]={0xFD, 0x00, 0x0B, 0x01, 0x01, 0xB7, 0xA2, 0xC9, 0xFA, 0xCA, 0xC2, 0xB9
, 0xCA, 0xAB};
char
sound_number[10][7]={ {0xFD, 0x00, 0x04, 0x01, 0x01, 0x30, 0xC9}, {0xFD, 0x00, 0x04, 0x01,
0x01, 0x31, 0xC8}, {0xFD, 0x00, 0x04, 0x01, 0x01, 0x32, 0xCB},

{0xFD, 0x00, 0x04, 0x01, 0x01, 0x33, 0xCA}, {0xFD, 0x00, 0x04, 0x01, 0x01, 0x34, 0xCD}, {0xFD
, 0x00, 0x04, 0x01, 0x01, 0x35, 0xCC},

{0xFD, 0x00, 0x04, 0x01, 0x01, 0x36, 0xCF}, {0xFD, 0x00, 0x04, 0x01, 0x01, 0x37, 0xCE}, {0xFD
, 0x00, 0x04, 0x01, 0x01, 0x38, 0xC1},

{0xFD, 0x00, 0x04, 0x01, 0x01, 0x39, 0xC0}}; //number 0~9
char sound_point[7]={0xFD, 0x00, 0x04, 0x01, 0x01, 0x2E, 0xD7}; //point

int Otsu(IplImage* src);
void ccImagePreProcess(IplImage* pCutFrImg, int T, int l1); // T为Otsu算法计算出的
ROI全局阈值 l1为测量出的实际车道像素宽度
void AdaptiveFindThreshold(CvMat *dx, CvMat *dy, double *low, double
*high); //CV_IMPL
void ccVideoLineDetect(IplImage *pFrame, int nFrmNum);
void probabilityOfLine(int x1, int y1, int x2, int y2, int imgWidth, double
*lweight, double *rweight, int picHalfHeight);
void fixSingleLine(int type, double *k, double *x, double k1);

void ccControl();
void ccStop();
void ccControlInit();
void ccTurnLeft();
void ccTurnRight();
```

```
void ccGoStraight();
void ccSetSpeed(int speed);
void ccBackward();
void ccRaiseFlag();
void ccDownFlag();
void ccControlClear();

//phone number
QString ownerPhone = "18911961729";

QString phoneNum;
QStringList phoneData;
void ccPhoneDataInit();
bool ccVerifyNumber(QString num);
void ccPhoneHangUp();
void ccDialPhone(QString num);

void ccAlarm(QString num);

QTime aCount;

//speech systhesyszier
void ccSpeechInit();
void ccCallPolice(QString num);

//supervision of chuan kou of gpsAndPhone
char supervibuf[100];
int statespv=0;
int indexsbuf=0;
void supervisionCK();

//supervision of chuan kou of Geniuno101
char geniunobuf[100];
int stategeniuno=0;
void ccGeniunoSupervision();

//GPS Control
struct GPSInfo gpsinfoForCall;
struct GPSInfo gpsinfo0;//gps info
struct GPSInfo gpsinfo1;
int state=0;
int indexgbuf=0;
char gpsBuf[100];
void ccGPSAndPhoneInit();
```

```
void ccGPSRead(struct GPSInfo *gpsinfo);
void ccGPSStart();
void ccGPSStop();

//-----chuan kou-----
int OpenDev(char *Dev);
void set_speed(int fd, int speed);
int set_Parity(int fd,int databits,int stopbits,int parity);

fd_set rfd;
struct timeval tv;

int fd;// genoniu 101
int gpsfd;//gps
int speechfd;//speaker

char  buffer[100];//write
int   Length;
int   nByte;

int nread=0;
char buf[1024];//read
//-----for tiao shi-----
void writeToTxt(QString command);
QString videoname;
QString txtname;

//帧数
int nFrmNum = 0;

int main() {
    //声明 IplImage 指针
    IplImage* pFrame = NULL;
    //声明 CvCapture 指针
    CvCapture* pCapture = NULL;

    //窗口命名
    cvNamedWindow("video", 1);
    cvNamedWindow("BWmode", 1);
    //调整窗口初始位置
    cvMoveWindow("video", 300, 0);
    cvMoveWindow("BWmode", 300, 520);
```



```
//initialize staLine
staLine.num=0;
staLine.point=0;

//initialize gpsinfo
gpsinfo0.completed=false;
gpsinfo1.completed=false;
gpsinfoForCall.completed=false;

//不能打开则退出

if (!(pCapture = cvCaptureFromFile("test 周三 6月 8 15:20:26 2016.avi"))){
    fprintf(stderr, "Can not open video file\n");
    return -2;
}

/*

VideoCapture cap1;
Mat frame1;
//int fps = 50;
bool stop=false;
cap1.open(0);
if (!cap1.isOpened())
{
    cap1.open(1);
    if (!cap1.isOpened()){
        cap1.open(2);
        if (!cap1.isOpened())
            return -1;
    }
}

//cap1.set(CV_CAP_PROP_FRAME_WIDTH, w);
//cap1.set(CV_CAP_PROP_FRAME_HEIGHT, h);
//cap1.set(CV_CAP_PROP_FPS, fps);

int isColol=1;
int frameW=640;
int frameH=480;
videoname = "test"+QDateTime::currentDateTime().toString()+".avi";
txtname = "test"+QDateTime::currentDateTime().toString()+".txt";
```

```
writer = cvCreateVideoWriter(videoname.toString().data(), CV_FOURCC('F',
'L', 'V', '1'), 2, cvSize(frameW, frameH), isColor); //fps ==2 CV_FOURCC('F', 'L',
'V', '1')

int abandon=0;
//ccControlInit();
*/
/*
ccPhoneDataInit();
ccGPSAndPhoneInit();
//ccSpeechInit();

//get gps info for call
ccGPSStart();
while(gpsinfoForCall.completed==false)
    ccGPSRead(&gpsinfoForCall);
ccGPSStop();
*/
//ccCallPolice("18911961729");
//ccAlarm("18911961729");

while(pFrame = cvQueryFrame(pCapture) ){ //pFrame =
cvQueryFrame(pCapture)    !stop
    nFrmNum+=4;

    if(nFrmNum<4400)
        continue;
/*
    nFrmNum++;
    abandon--;
    if(abandon>=0){
        cap1 >> frame1;
        continue;
    }
    cap1 >> frame1;
    abandon=4;
    IplImage a = IplImage(frame1);
    pFrame = &a;
*/
    ccVideoLineDetect(pFrame, nFrmNum);
    //ccControl();
```

```
/*          //record the video
int o;
if(o=cvWriteFrame(writer, pFrame) == 0) {
    printf("\n\n\nwrong wrong wrong wrong wrong\n\n\n");
}
*/
/*
supervisionCK();
//ccGeniunoSupervision();
*/

//按键事件，空格暂停，其他跳出循环
int temp = cvWaitKey(2);
if (temp == 32){
    while (cvWaitKey() != 's');
}
else if (temp == 'q'){
    break;
}
}

//销毁窗口
cvDestroyWindow("video");
cvDestroyWindow("BWmode");
//释放图像
cvReleaseImage(&pCutFrImg);
cvReleaseImage(&pCutFrame);
cvReleaseCapture(&pCapture);
cvReleaseVideoWriter(&writer);

ccControlClear();
return 0;
}

int Otsu(IplImage* src)
{
    int height = src->height;
    int width = src->width;

    //histogram
```

```
float histogram[256] = { 0 };
for (int i = 0; i < height; i++)
{
    unsigned char* p = (unsigned char*)src->imageData + src->widthStep * i;
    for (int j = 0; j < width; j++)
    {
        histogram[*p++]++;
    }
}
//normalize histogram
int size = height * width;
for (int i = 0; i < 256; i++)
{
    histogram[i] = histogram[i] / size;
}

//average pixel value
float avgValue = 0;
for (int i = 0; i < 256; i++)
{
    avgValue += i * histogram[i]; //整幅图像的平均灰度
}

int threshold;
float maxVariance = 0;
float w = 0, u = 0;
for (int i = 0; i < 256; i++)
{
    w += histogram[i]; //假设当前灰度 i 为阈值, 0~i 灰度的像素(假设像素值在
    //此范围的像素叫做前景像素) 所占整幅图像的比例
    u += i * histogram[i]; // 灰度 i 之前的像素(0~i)的平均灰度值: 前景像素
    //的平均灰度值

    float t = avgValue * w - u;
    float variance = t * t / (w * (1 - w));
    if (variance > maxVariance)
    {
        maxVariance = variance;
        threshold = i;
    }
}

return threshold;
}
```

```

void ccImagePreProcess(IplImage* pCutFrImg, int T1,int roadwidth){
    //uchar* data = (uchar *)pCutFrImg->imageData;
    //int step = pCutFrImg->widthStep / sizeof(uchar);
    int T = 0.5* T1;
    uchar* tmp;
    int l = roadwidth;//车道像素宽度
    int c = 0;
    int N = pCutFrImg->height;
    int M = pCutFrImg->width;
    double mean1=0, mean2=0;//mean1 for pi    mean2 for pj
    for (int y = 0; y<N; y++)
        for (int x = 0; x < M; x++)
        {
            mean1 = 0; mean2 = 0;
            tmp = &((uchar *) (pCutFrImg->imageData + y*pCutFrImg->widthStep))[x];
            //tmp = cvGet2D(pCutFrImg, y, x).val[0];
            c = y*0.5*l / (M - 1) + 0.5*l;

            if (y<N-c && y>c){
                for (int i = y - c; i < y;i++){//计算 mean(V(x,pi))
                    if (i<N && i>=0)
                        mean1 += ((uchar *) (pCutFrImg->imageData +
i*pCutFrImg->widthStep))[x];//cvGet2D(pCutFrImg, i, x).val[0];
                }
                mean1 /= c;
                for (int i = y; i < y + c; i++){//计算 mean(V(x,pj))
                    if (i<N && i>=0)
                        mean2 += ((uchar *) (pCutFrImg->imageData +
i*pCutFrImg->widthStep))[x];//cvGet2D(pCutFrImg, i, x).val[0];
                }
                mean2 /= c;
                if (*tmp>mean1+T && *tmp>mean2+T){
                    *tmp = 255;
                }
                else
                {
                    *tmp = 0;
                }
            }
        }
    else if (y>=N-c){
        for (int i = y - c; i < y; i++){//计算 mean(V(x,pi))
            if (i<N && i>=0)
    
```

```

        mean1 += ((uchar *) (pCutFrImg->imageData +
i*pCutFrImg->widthStep))[x]; //cvGet2D(pCutFrImg, i, x).val[0];
    }
    mean1 /= c;
    if (*tmp>mean1+T) {
        *tmp = 255;
    }
    else
    {
        *tmp = 0;
    }
}
else
{
    for (int i = y; i < y + c; i++) { //计算 mean(V(x, pj))
        if (i<N && i>=0)
            mean2 += ((uchar *) (pCutFrImg->imageData +
i*pCutFrImg->widthStep))[x]; //cvGet2D(pCutFrImg, i, x).val[0];
        }
        mean2 /= c;
        if (*tmp>mean2 + T) {
            *tmp = 255;
        }
        else
        {
            *tmp = 0;
        }
    }
}
}
}

```

```

void AdaptiveFindThreshold(CvMat *dx, CvMat *dy, double *low, double
*high) //CV_IMPL
{
    CvSize size;
    IplImage *imge = 0;
    int i, j;
    CvHistogram *hist;
    int hist_size = 255;
    float range_0[] = { 0, 256 };
    float* ranges[] = { range_0 };
    double PercentOfPixelsNotEdges = 0.7;
    size = cvGetSize(dx);
    imge = cvCreateImage(size, IPL_DEPTH_32F, 1);
}

```

```
// 计算边缘的强度, 并存于图像中
float maxv = 0;
for (i = 0; i < size.height; i++)
{
    const short* _dx = (short*)(dx->data.ptr + dx->step*i);
    const short* _dy = (short*)(dy->data.ptr + dy->step*i);
    float* _image = (float*)(imge->imageData + imge->widthStep*i);
    for (j = 0; j < size.width; j++)
    {
        _image[j] = (float)(abs(_dx[j]) + abs(_dy[j]));
        maxv = maxv < _image[j] ? _image[j] : maxv;
    }
}
// 计算直方图
range_0[1] = maxv;
hist_size = (int)(hist_size > maxv ? maxv : hist_size);
hist = cvCreateHist(1, &hist_size, CV_HIST_ARRAY, ranges, 1);
cvCalcHist(&imge, hist, 0, NULL);
int total = (int)(size.height * size.width * PercentOfPixelsNotEdges);
float sum = 0;
int icount = hist->mat.dim[0].size;

float *h = (float*)cvPtr1D(hist->bins, 0);
for (i = 0; i < icount; i++)
{
    sum += h[i];
    if (sum > total)
        break;
}
// 计算高低门限
*high = (i + 1) * maxv / hist_size;
*low = *high * 0.4;
cvReleaseImage(&imge);
cvReleaseHist(&hist);
}

//for detect
bool firstFrame=true;
//for control
bool firstTime=true;

void ccVideoLineDetect(IplImage *pFrame,int nFrmNum){
    //用于动态计算 canny 阈值
    double low = 0, high = 0;
```



```
QTime timedif;

timedif.start();

//设置 ROI 裁剪图像
cvSetImageROI(pFrame, cvRect(CutWidth, CutHeight, pFrame->width - CutWidth,
pFrame->height - CutHeight));

//第一次要申请内存 p
if (firstFrame){
    size.width = (pFrame->width - CutWidth) * fScale;
    size.height = (pFrame->height - CutHeight) * fScale;
    pCutFrame = cvCreateImage(size, pFrame->depth, pFrame->nChannels);
    //cvCopy(pFrame, pCutFrame, 0);
    cvResize(pFrame, pCutFrame, CV_INTER_NN);
    pCutFrImg = cvCreateImage(cvSize(pCutFrame->width, pCutFrame->height),
IPL_DEPTH_8U, 1);
    //转化成单通道图像再处理
    cvCvtColor(pCutFrame, pCutFrImg, CV_BGR2GRAY);

    //for display memory apply
    templ = cvCreateImage(cvSize(size.width*1.0, size.height*1.0),
pCutFrame->depth, pCutFrame->nChannels);
    temp2 = cvCreateImage(cvSize(size.width*1.0, size.height*1.0),
pCutFrImg->depth, pCutFrImg->nChannels);

    //for canny detector
    mat = cvCreateMat(pCutFrImg->height, pCutFrImg->width, CV_32SC1);
    mat1 = cvCreateMat(pCutFrImg->height, pCutFrImg->width, CV_32SC1);

    //计算 l1
    l1 = size.width
*11/12;//~~~~~!!!!!!!!!!!!!!!!!!!!!!!!!!!!~::~:

    //caculate ground truth of road
    gr_l_point1=cvPoint( size.width*2/5,0);
    gr_l_point2=cvPoint(0,size.height/2);
    gr_r_point1=cvPoint(size.width *3/5,0);
    gr_r_point2=cvPoint(size.width ,size.height/2);

    firstFrame=false;
}

//获得剪切图
```

```
//cvCopy(pFrame, pCutFrame, 0);
cvResize(pFrame, pCutFrame, CV_INTER_NN);

//begin~~~~~main detect
//前景图转换为灰度图
cvCvtColor(pCutFrame, pCutFrImg, CV_BGR2GRAY);

//Ostu 求 ROI 全局阈值
int T = Otsu(pCutFrImg);
//求 canny 动态阈值
cvConvert(pCutFrImg, mat);
cvConvert(pCutFrImg, mat1);
AdaptiveFindThreshold(mat, mat1, &low, &high);

//图像预处理：基于对称阈值分割的车道线特征提取
ccImagePreProcess(pCutFrImg, T, 11);

//二值化前景图
//cvAdaptiveThreshold(pCutFrImg, pCutFrImg, 255,
CV_ADAPTIVE_THRESH_GAUSSIAN_C, CV_THRESH_BINARY);
//cvThreshold(pCutFrImg, pCutFrImg, T, 255.0, CV_THRESH_BINARY); //80, 150
//进行形态学滤波，去掉噪音
cvErode(pCutFrImg, pCutFrImg, 0, 1); //2
cvDilate(pCutFrImg, pCutFrImg, 0, 1); //2
//canny 变化
cvCanny(pCutFrImg, pCutFrImg, low, high); // (50, 120)

//hough 变换
lines = cvHoughLines2(pCutFrImg, storage, CV_HOUGH_PROBABILISTIC, 1, CV_PI /
180, 50, 15, 5); // (50, 15, 15)
//end~~~~~main detect

/*
//begin~~~~~color detect
Mat imgOriginal(pCutFrame, 0);
Mat imgHSV;
vector<Mat> hsvSplit;
cvtColor(imgOriginal, imgHSV, COLOR_BGR2HSV); //Convert the captured frame
from BGR to HSV

//因为我们读取的是彩色图，直方图均衡化需要在 HSV 空间做
split(imgHSV, hsvSplit);
```

```
equalizeHist(hsvSplit[2],hsvSplit[2]);
merge(hsvSplit,imgHSV);
Mat imgThresholded1;

    inRange(imgHSV, Scalar(iLowH, iLowS, iLowV), Scalar(iHighH, iHighS, iHighV),
imgThresholded1); //Threshold the image

//开操作（去除一些噪点）
Mat element = getStructuringElement(MORPH_RECT, Size(5, 5));
morphologyEx(imgThresholded1, imgThresholded1, MORPH_OPEN, element);

//闭操作（连接一些连通域）
morphologyEx(imgThresholded1, imgThresholded1, MORPH_CLOSE, element);

//imshow("Thresholded Image", imgThresholded1); //show the thresholded image

IplImage imgThresholded3 = IplImage(imgThresholded1);
IplImage *imgThresholded = & imgThresholded3;
//canny 变化
cvCanny(imgThresholded, imgThresholded, low, high); //(50,120)

//hough 变换
lines2 = cvHoughLines2(imgThresholded, storage, CV_HOUGH_PROBABILISTIC, 1,
CV_PI / 180, 50, 15, 5); //(50,15,15)
//end~~~~~color detect
*/

//画出直线
l_line1=cvPoint(0, 0); //1 is the smaller one, 2 is the bigger one
l_line2 =cvPoint(0, 0);
r_line1 =cvPoint(0, 0);
r_line2 =cvPoint(0, 0);
int l_num=0,r_num=0;
double l_total_weight = 0;
double r_total_weight = 0;
double maxLP=0;
double maxRP=0;

//main detected line
for (int i = 0; i<lines->total; i++){
    CvPoint* line = (CvPoint*)cvGetSeqElem(lines, i);
    double k = ((line[0].y - line[1].y)*1.0 / (line[0].x - line[1].x));
    double lweight = 0;
    double rweight = 0;
```

```
//计算这条线段是车道线的概率
probabilityOfLine (line[0].x, line[0].y, line[1].x, line[1].y,
size.width,&lweight,&rweight,size.height/2);

cout << "nFrmNum " << nFrmNum << " 's k = " << k << " left weight: " <<
lweight << " " << "right weight: " << rweight << endl;
//to filter the lines
//cvLine(pCutFrame, line[0], line[1], CV_RGB(255, 0, 0), 6, CV_AA);
if (lweight>0){
    //cvLine(pCutFrame, line[0], line[1], CV_RGB(255, 0, 0), 6, CV_AA);
    if(maxLP<lweight)
        maxLP=lweight;
    l_total_weight += lweight;
    l_num++;
    if (line[0].y < line[1].y){
        l_line1.y += line[0].y *lweight;
        l_line1.x += line[0].x *lweight;
        l_line2.y += line[1].y *lweight;
        l_line2.x += line[1].x *lweight;
    }
    else
    {
        l_line1.y += line[1].y *lweight;
        l_line1.x += line[1].x *lweight;
        l_line2.y += line[0].y *lweight;
        l_line2.x += line[0].x *lweight;
    }
}
}else if (rweight>0){
    //cvLine(pCutFrame, line[0], line[1], CV_RGB(255, 0, 0), 6, CV_AA);
    if(maxRP<rweight)
        maxRP=rweight;
    r_total_weight += rweight;
    r_num++;
    if (line[0].y < line[1].y){
        r_line1.y += line[0].y *rweight;
        r_line1.x += line[0].x *rweight;
        r_line2.y += line[1].y *rweight;
        r_line2.x += line[1].x *rweight;
    }
    else
    {
        r_line1.y += line[1].y *rweight;
        r_line1.x += line[1].x *rweight;
```

```

        r_line1.y += line[0].y *rweight;
        r_line1.x += line[0].x *rweight;
    }
}
}

/*
//color detected line
for (int i = 0; i<lines2->total; i++){
    CvPoint* line = (CvPoint*)cvGetSeqElem(lines2, i);
    double k = ((line[0].y - line[1].y)*1.0 / (line[0].x - line[1].x));
    double lweight = 0;
    double rweight = 0;

    //计算这条线段是车道线的概率
    probabilityOfLine (line[0].x, line[0].y, line[1].x, line[1].y,
size.width,&lweight,&rweight,size.height/2);

    //cout << "nFrmNum " << nFrmNum << " 's k = " << k << " left weight: " <<
lweight << " " << "right weight: " << rweight << endl;

    //to filter the lines
    //cvLine(pCutFrame, line[0], line[1], CV_RGB(255, 0, 0), 6, CV_AA);
    if (lweight>0){
        //cvLine(pCutFrame, line[0], line[1], CV_RGB(255, 0, 0), 6, CV_AA);
        if(maxLP<lweight)
            maxLP=lweight;
        l_total_weight += lweight;
        l_num++;
        if (line[0].y < line[1].y){
            l_line1.y += line[0].y *lweight;
            l_line1.x += line[0].x *lweight;
            l_line2.y += line[1].y *lweight;
            l_line2.x += line[1].x *lweight;
        }
        else
        {
            l_line1.y += line[1].y *lweight;
            l_line1.x += line[1].x *lweight;
            l_line2.y += line[0].y *lweight;
            l_line2.x += line[0].x *lweight;
        }
    }
    }else if (rweight>0){
        //cvLine(pCutFrame, line[0], line[1], CV_RGB(255, 0, 0), 6, CV_AA);
    }
}

```

```

        if(maxRP<rweight)
            maxRP=rweight;
        r_total_weight += rweight;
        r_num++;
        if (line[0].y < line[1].y) {
            r_line1.y += line[0].y *rweight;
            r_line1.x += line[0].x *rweight;
            r_line2.y += line[1].y *rweight;
            r_line2.x += line[1].x *rweight;
        }
        else
        {
            r_line1.y += line[1].y *rweight;
            r_line1.x += line[1].x *rweight;
            r_line1.y += line[0].y *rweight;
            r_line1.x += line[0].x *rweight;
        }
    }
}

*/

if (l_total_weight > 0) { //归一化
    l_line1.x /= (l_total_weight);
    l_line1.y /= (l_total_weight);
    l_line2.x /= (l_total_weight);
    l_line2.y /= (l_total_weight);
}

if (r_total_weight>0) { //归一化
    r_line1.x /= (r_total_weight);
    r_line1.y /= (r_total_weight);
    r_line2.x /= (r_total_weight);
    r_line2.y /= (r_total_weight);
}

//begin----- for test
    if (l_total_weight > 0)
        cvLine(pCutFrame,l_line1, l_line2, CV_RGB(255, 0, 0), 10, CV_AA);
    if (r_total_weight>0)
        cvLine(pCutFrame,r_line1, r_line2, CV_RGB(255, 0, 0), 10, CV_AA);
    printf("lines2 number: %d   time: %dms   u1:%d u2:%d \n" ,
lines->total,timedif.elapsed() ,u1,u2);
//end-----for test

//re-initialize the gussian-array
    
```

```

    for(int e=0;e<20;e++){
        Gussian_right[e]=Gussian_backup[e];
        Gussian_left[e]=Gussian_backup[e];
    }
//-----

    bool f1=false,f2=false;
    double k,k1;
    double b, b1;
    double x01=0, x02=0,centerx=0;
    double distance=0;
    bool sign=true;

    //fittable parameter
    int turnValue = 11/10;
    int adjustValue = 11 *3/2;
    int axisValue = 11*5/16;
    int abanValue = 11/4;
    double kValue =0.2;
    double kForEmergency = 0.9;

    //calculate line that detected
    if (l_total_weight>0){
        k = (l_line1.y *(1.0) - l_line2.y) / (l_line1.x *(1.0) - l_line2.x);
        b = l_line1.y *1.0 - k*l_line1.x;
        x01 = (size.height/2 -b)/k;
        f1 = true;
    }
    if (r_total_weight>0){
        k1 = (r_line1.y *(1.0) - r_line2.y) / (r_line1.x *(1.0) - r_line2.x);
        b1 = r_line1.y *1.0 - k1*r_line1.x;
        x02 = (size.height/2-b1 )/ k1;
        f2 = true;
    }
    //to make two line
    if (!f1){
        if (f2){
            //x01 = x02 - adjustValue;
            //k = -k1;
            fixSingleLine(1,&k,&x01,k1);
            b = size.height/2 - k * x01;
            centerx = (b1-b)/(k-k1);

            distance = (size.width/2 - centerx) ;

```



```

    }
    else{
        //go straight
        sign=false;
    }
} else{
    if (!f2){
        //x02 = x01 + adjustValue;
        //k1=-k;
        fixSingleLine(0,&k1,&x02,k);
        b1 = size.height/2 - k1 *x02;
        centerx = (b1-b)/(k-k1);

        distance = (size.width/2 - centerx) ;
    }else{
        if( x02-x01-l1 > axisValue || x02-x01-l1 <- axisValue){// use
probability to adjust when two line detected
            if(maxLP>maxRP){//l_total_weight/l_num > r_total_weight/r_num
                x02 = x01 + adjustValue;
                k1=-k;
            }
            else{
                x01 = x02 - adjustValue;
                k=-k1;
            }
        }
        centerx = (x01 + x02)/2;
        distance = (size.width/2 - centerx);
    }
}

/*
// throw the line away if the change bewteen last two frame and it is too huge
if(staLine.num<2 && sign ){
    staLine.num++;
    staLine.line[staLine.point][0]=x01;
    staLine.line[staLine.point][1]=k;
    staLine.line[staLine.point][2]=x02;
    staLine.line[staLine.point][3]=k1;
    staLine.p[staLine.point][0]=l_total_weight/l_num;
    staLine.p[staLine.point][1]=r_total_weight/r_num;
    staLine.point = (staLine.point+1)%3;
}else if(sign){
    int past1=(staLine.point+1)%3, past2 = (staLine.point+2)%3;

```

```

        if(l_total_weight>0 &&
            ( ((x01+x02)/2 -
(staLine.line[past1][0]+staLine.line[past1][2])/2) > abanValue ||
            ((x01+x02)/2 -
(staLine.line[past1][0]+staLine.line[past1][2])/2) < -abanValue )&&
            ( (staLine.p[past1][0]+staLine.p[past2][0])/2 >
( l_total_weight/r_num -0.15 ) ) ) {&&
            ( (staLine.p[past1][0]+staLine.p[past2][0])/2 > ( l_total_weight/r_num
-0.15 ) )
                sign=false;
            }else if(r_total_weight>0 &&
                ( ((x01+x02)/2 -
(staLine.line[past1][0]+staLine.line[past1][2])/2) > abanValue ||
                ((x01+x02)/2 -
(staLine.line[past1][0]+staLine.line[past1][2])/2) < -abanValue )&&
                ( (staLine.p[past1][1]+staLine.p[past2][1])/2 >
(r_total_weight/r_num -0.15 ) ) ) {
                    sign=false;
                }
            else {
                staLine.num++;
                staLine.line[staLine.point][0]=x01;
                staLine.line[staLine.point][1]=k;
                staLine.line[staLine.point][2]=x02;
                staLine.line[staLine.point][3]=k1;
                staLine.p[staLine.point][0]=l_total_weight/l_num;
                staLine.p[staLine.point][1]=r_total_weight/r_num;
                staLine.point = (staLine.point+1)%3;
            }
        }

    */

    //-----for test
    if(sign) {
        if(k==0 || k1==0 )
            sign=false;
        else{
            printf("-----%f-----%d-----\n", centerx, size.width / 2);
            cvLine(pCutFrame, cvPoint( centerx,100), cvPoint(size.width /
2,100 ), CV_RGB(255, 0, 0), 3, CV_AA);

            b=size.height/2- k*x01;
            b1=size.height/2 -k1*x02;

```

```

        cvLine(pCutFrame,cvPoint(0,(int)b), cvPoint((int)(-b/k),0),
CV_RGB(150,10,10), 5, CV_AA);
        cvLine(pCutFrame,cvPoint(size.width,(int)(size.width*k1+b1)),
cvPoint((int)(-b1/k1),0), CV_RGB(150,10, 10), 5, CV_AA);
    }
    printf("-----left:-----%f-----%d-----\n", k ,b);
    printf("-----right:-----%f-----%d-----\n", k1, b1);
}
//for test-----

//control
if(!spot){
    if(!sign || (timespan.elapsed()<=400 && !firstTime ) ){
        if(carS!=sTraight){//new@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
            ccGoStraight();
            ccSetSpeed(1);
            carS=sTraight;
        }
    }else if(distance>turnValue){
        if( carS!=lEft){
            ccTurnLeft();
            ccSetSpeed(2);
            carS=lEft;
        }
        if(k1>=kForEmergency){
            firstTime=false;
            timespan.start();
            printf("!!!!!!!!!!!!!!!!!!!!!!!!!!!! keep go
left  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
        }
    }else if(distance<=-turnValue){
        if(carS!=rIght){
            ccTurnRight();
            ccSetSpeed(2);
            carS=rIght;
        }
        if(k<=-kForEmergency){
            firstTime=false;
            timespan.start();
            printf("!!!!!!!!!!!!!!!!!!!!!!!!!!!! keep go
right  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
        }
    }else if(carS!=sTraight){

```

```

        ccGoStraight();
        ccSetSpeed(2);
        carS=sTraight;
    }

    if(prespot){
        if(carS==sTraight){
            spot=true;
            ccStop();
            carS=sTop;
        }
    }

} else if(spot){ // recieve the call to go back
    ccGPSStart();

printf("!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\n!!!!!!!!!!!!!!!!!!!!!!!!\nturn--back\n!!!!!!!!\n");
    while(gpsinfo0.completed!=true)
        ccGPSRead(&gpsinfo0);
    for(int i=0;i<5;i++){
        backspan.start();
        ccTurnRight();
        ccBackward();
        carS=rIght;
        while(backspan.elapsed()<1100)
            ;
        ccStop();
        carS=sTop;
        while(backspan.elapsed()<1100+500)
            ;
        backspan.start();
        ccTurnLeft();
        ccSetSpeed(2);
        carS=lEft;
        while(backspan.elapsed()<1000)
            ;
        ccStop();
        carS=sTop;
        while(backspan.elapsed()<1000+500)
            ;
        while(gpsinfo1.completed!=true)

```

```

        ccGPSRead(&gpsinfo1);
        gpsinfo1.completed=false;
        if(gpsinfo1.angel-gpsinfo0.angel >150
||gpsinfo1.angel-gpsinfo0.angel <-150) {
            printf("oh stop by angel,now turn time is %d",i);
            break;
        }
    }
    ccStop();
    ccGPSStop();
    prespot=false;
    spot =false;
}

//-----
//缩放显示
cvResize(pCutFrame, temp1, CV_INTER_NN);
cvResize(pCutFrImg, temp2, CV_INTER_NN);

//显示图像
cvShowImage("video", temp1);
cvShowImage("BWmode", temp2);
}

void fixSingleLine(int type,double *k, double *x, double k1){//type==0, know left
line; type==1, know right line
    //ground truth of right line
    double krt = (gr_r_point1.y*1.0 - gr_r_point2.y)/(gr_r_point1.x*1.0 -
gr_r_point2.x); //ground truth's k
    //double brt = gr_r_point1.y - krt * gr_r_point1.x;
    //ground truth of left line
    double klt = (gr_l_point1.y*1.0 - gr_l_point2.y)/(gr_l_point1.x*1.0 -
gr_l_point2.x); //ground truth's k
    //double blt = gr_l_point1.y - klt * gr_l_point1.x;

    if(type==1){//detected right line

        //calculate the rotated angle
        double theate =-( qAtan(krt) - qAtan(k1) );
        //translation to origin
        double tempx = gr_l_point2.x-gr_l_point1.x;

```

```

double tempy = gr_l_point2.y-gr_l_point1.y;
//translate
double transx = tempx*qCos(theate) - tempy*qSin(theate);
double transy = tempx*qSin(theate) + tempy*qCos(theate);
//translation back to what it was
double backx = transx + gr_l_point1.x;
double backy = transy + gr_l_point1.y;
//result
*k = (backy - gr_l_point1.y)/(backx - gr_l_point1.x);
double tempb = gr_l_point1.y - *k * gr_l_point1.x;
*x = (size.height/2 - tempb) / *k;
}else if(type==0) { //detected left line

//calculate the rotated angle
double theate = -( qAtan(klt) - qAtan(kl) );
//translation to origin
double tempx = gr_r_point2.x-gr_r_point1.x;
double tempy = gr_r_point2.y-gr_r_point1.y;
//translate
double transx = tempx*qCos(theate) - tempy*qSin(theate);
double transy = tempx*qSin(theate) + tempy*qCos(theate);
//translation back to what it was
double backx = transx + gr_r_point1.x;
double backy = transy + gr_r_point1.y;
//result
*k = (backy - gr_r_point1.y)/(backx - gr_r_point1.x);
double tempb = gr_r_point1.y - *k * gr_r_point1.x;
*x = (size.height/2 - tempb) / *k;
}
}

void probabilityOfLine(int x1,int y1,int x2,int y2,int imgWidth,double
*lweight,double *rweight,int picHalfHeight){
double k = ((y1 - y2)*1.0 / (x1 - x2));
double b = y1 - k*x1;
double b1 = (picHalfHeight-b)/k;    //y==half height 时，x 的坐标，用于权值判
断
double maxX = x1 > x2 ? x1 : x2;
double minX = x1<x2? x1: x2;
int margin;

b1 = b1>0?b1:-b1;
int d =(int)(b1 / imgWidth *20);

```

```

        if (k<-0.17 && k>-2) {
            *rweight = 0;
            margin=d > u1 ? d - u1 : u1 - d;
            if(margin>19){
                margin=19;
            }
            *lweight = Gussian_left[margin];
            Gussian_left[margin]=0;
            return;
        }
        else
        {
            *lweight = 0;
        }

        if (k>0.17 && k<2) {
            *lweight = 0;
            margin=d > u2 ? d - u2 : u2 - d;
            if(margin>19){
                margin=19;
            }
            *rweight = Gussian_right[margin];
            Gussian_right[margin]=0;
        }
        else
        {
            *rweight = 0;
        }
    }

//1 放得是 y 值小的那个 2 是 y 值大的那个
void ccControl() {
    double k = (l_line1.y *(1.0) - l_line2.y) / (l_line1.x *(1.0) - l_line2.x);
    double b = l_line1.y *1.0 - k*l_line1.x;
    double x0l = -b / k;

    double k1 = (r_line1.y *(1.0) - r_line2.y) / (r_line1.x *(1.0) - r_line2.x);
    double b1 = r_line1.y *1.0 - k1*r_line1.x;

```



```
double x02 = -b1 / k1;

double distance = size.width > (x01 + x02) ? (size.width - (x01 + x02)) / 2 : ((x01
+ x02) - size.width) / 2;
printf("-----%f-----%d-----", (x01+x02)/2, size.width/2);

}

void ccControlInit() {
    char *dev0 = "/dev/ttyACM0";
    char *dev1 = "/dev/ttyACM1";          //   "/dev/ttyACM1"; //串口二
    char *dev2 = "/dev/ttyACM2";
    char *dev3 = "/dev/ttyACM3";
    char *dev4 = "/dev/ttyACM4";
    if((fd = OpenDev(dev0)) == -1)
        if((fd = OpenDev(dev1)) == -1)
            if((fd = OpenDev(dev2)) == -1)
                if((fd = OpenDev(dev3)) == -1)
                    if((fd = OpenDev(dev4)) == -1)
                        exit(0);

    printf("port open succeed!!!!");
    set_speed(fd, 19200);
    if (set_Parity(fd, 8, 1, 'N') == false) {
        printf("Set Parity Error\n");
        exit (0);
    }

    tv.tv_sec=3;
    tv.tv_usec=0;
    FD_ZERO(&rfd);
    FD_SET(fd, &rfd);
}

void ccControlClear() {
    close(fd);
    close(gpsfd);
    close(speechfd);
}

void ccTurnLeft() {
    writeToTxt("left");
}
```

```

        printf("\n-----turn
left-----");
        buffer[0]='l';
        Length=1;
        nByte = write(fd, buffer ,Length);
    }

void ccGoStraight() {
    writeToTxt("straight");
    printf("\n-----go
straight-----");
    buffer[0]='q';
    Length=1;
    nByte = write(fd, buffer ,Length);
}

void ccTurnRight() {
    writeToTxt("right");
    printf("\n-----turn
right-----");
    buffer[0]='r';
    Length=1;
    nByte = write(fd, buffer ,Length);
}

void ccSetSpeed(int speed) {
    QString cmd = "speed"+QString::number(speed);
    writeToTxt(cmd);

    printf("\n-----speed  %d-----
-----", speed);

    if(speed==1)
        buffer[0]='a';
    else if(speed==2)
        buffer[0]='b';
    else if(speed==3)
        buffer[0]='c';
    Length=1;
    nByte = write(fd, buffer ,Length);
}

void ccBackward() {
    writeToTxt("Back");

```

```

        printf("\n-----go
back-----");
        buffer[0]='d';
        Length=1;
        nByte = write(fd, buffer ,Length);
    }

void ccRaiseFlag() {
    writeToTxt("raise flag");
    printf("\n-----raise
flag-----");
    buffer[0]='u';
    Length=1;
    nByte = write(fd, buffer ,Length);
}

void ccDownFlag() {
    writeToTxt("down flag");
    printf("\n-----down
flag-----");
    buffer[0]='e';
    Length=1;
    nByte = write(fd, buffer ,Length);
}

void ccStop() {
    printf("stop-----");
    buffer[0]='s';
    Length=1;
    nByte = write(fd, buffer ,Length);
}

int OpenDev(char *Dev)
{
    int fd = open( Dev, O_NONBLOCK|O_RDWR| O_NOCTTY | O_NDELAY );           //|
    O_NOCTTY | O_NDELAY|O_RDWR
    if (-1 == fd)
    {
        perror("Can't Open Serial Port");
        return -1;
    }
    else
        return fd;
}
    
```

```
void set_speed(int fd, int speed){
    int i;
    int status;
    struct termios Opt;
    tcgetattr(fd, &Opt);
    Opt.c_iflag &= ~(INLCR|ICRNL |IGNCR); //new
    Opt.c_oflag &= ~(ONLCR|OCRNL); //new
    Opt.c_iflag &= ~(IXON); //new

    for ( i = 0; i < sizeof(speed_arr) / sizeof(int); i++) {
        if (speed == name_arr[i]) {
            tcflush(fd, TCIOFLUSH);
            cfsetispeed(&Opt, speed_arr[i]);
            cfsetospeed(&Opt, speed_arr[i]);
            status = tcsetattr(fd, TCSANOW, &Opt);
            if (status != 0) {
                perror("tcsetattr fd");
                return;
            }
            tcflush(fd, TCIOFLUSH);
        }
    }
}
```

```
int set_Parity(int fd,int databits,int stopbits,int parity)
{
    struct termios options;
    if ( tcgetattr( fd,&options) != 0) {
        perror("SetupSerial 1");
        return(false);
    }
    options.c_cflag &= ~CSIZE;
    options.c_lflag &= ~ (ICANON|ECHO|ECHOE|ISIG); // new
    options.c_oflag &= ~OPOST;
    //new
```

```
    switch (databits) /*设置数据位数*/
    {
        case 7:
            options.c_cflag |= CS7;
            break;
        case 8:
            options.c_cflag |= CS8;
```

```
        break;
    default:
        fprintf(stderr, "Unsupported data size\n"); return (false);
    }
switch (parity)
{
    case 'n':
    case 'N':
        options.c_cflag &= ~PARENB;    /* Clear parity enable */
        options.c_iflag &= ~INPCK;     /* Enable parity checking */
        break;
    case 'o':
    case 'O':
        options.c_cflag |= (PARODD | PARENB); /* 设置为奇效验*/
        options.c_iflag |= INPCK;             /* Disable parity checking */
        break;
    case 'e':
    case 'E':
        options.c_cflag |= PARENB;          /* Enable parity */
        options.c_cflag &= ~PARODD;         /* 转换为偶效验*/
        options.c_iflag |= INPCK;           /* Disable parity checking */
        break;
    case 'S':
    case 's': /*as no parity*/
        options.c_cflag &= ~PARENB;
        options.c_cflag &= ~CSTOPB; break;
    default:
        fprintf(stderr, "Unsupported parity\n");
        return (false);
    }
/* 设置停止位*/
switch (stopbits)
{
    case 1:
        options.c_cflag &= ~CSTOPB;
        break;
    case 2:
        options.c_cflag |= CSTOPB;
        break;
    default:
        fprintf(stderr, "Unsupported stop bits\n");
        return (false);
    }
/* Set input parity option */
```

```
if (parity != 'n')
    options.c_iflag |= INPCK;
tcflush(fd, TCIFLUSH);
options.c_cc[VTIME] = 0; /* 设置超时 15 seconds*/ //150
options.c_cc[VMIN] = 0; /* Update the options and do it NOW */
if (tcsetattr(fd, TCSANOW, &options) != 0)
{
    perror("SetupSerial 3");
    return (false);
}
return (true);
}

void writeToTxt(QString command) {
    QFile file(txtname);
    if(!file.open(QIODevice::Append | QIODevice::Text))
        return;
    QTextStream out(&file);
    out << command
    <<QDateTime::currentDateTime().toString()<<"--frame:"<<QString::number(nFrmNum,
    10)<<"\n";
}

void ccGPSAndPhoneInit() {
    char *dev0  = "/dev/ttyUSB0";
    char *dev1  = "/dev/ttyUSB1";
    char *dev2  = "/dev/ttyUSB2";
    char *dev3  = "/dev/ttyUSB3";
    char *dev4  = "/dev/ttyUSB4";
    if((gpsfd = OpenDev(dev0) )== -1)
        if((gpsfd = OpenDev(dev1) )== -1)
            if((gpsfd = OpenDev(dev2) )== -1)
                if((gpsfd = OpenDev(dev3) )== -1)
                    if((gpsfd = OpenDev(dev4) )== -1) {
                        printf("gps&phone port open failed!!!!");
                        writeToTxt("gps&phone port open failed!!!!");
                    }

    printf("gps port open succeed!!!!");
    set_speed(gpsfd, 19200);
    if (set_Parity(gpsfd, 8, 1, 'N') == false) {
```

```
        printf("Set Parity Error\n");
        exit (0);
    }

    strcpy(buffer, "AT+CLIP=1");// set coming call display
    buffer[9]='\n';
    Length=10;
    nByte = write(gpsfd, buffer ,Length);
}

void ccGPSStart() {
    strcpy(buffer, "AT+CGNSPWR=1");// init gps
    buffer[12]='\n';
    Length=13;
    nByte = write(gpsfd, buffer ,Length);

    strcpy(buffer, "AT+CGNSTST=1");// init gps
    buffer[12]='\n';
    Length=13;
    nByte = write(gpsfd, buffer ,Length);
}

void ccGPSStop() {
    strcpy(buffer, "AT+CGNSTST=0");// init gps
    buffer[12]='\n';
    Length=13;
    nByte = write(gpsfd, buffer ,Length);
}

void ccGPSRead(struct GPSInfo *gpsinfo) {
    if( (nread=read(gpsfd, buf, 1023))>0 ) {
        buf[nread+1]='\0';
        printf("\n\n----return:::----- %s \n\n", buf);
    }
    for(int i=0;i<nread;i++) {
        switch(state) {
            case 0:
                if(buf[i]=='$') {
                    state=1;
                }
                break;
            case 1:
                if(buf[i]=='G') {
                    state=2;
                }
            }
        }
    }
}
```

```
    }else{
        state=0;
    }
    break;
case 2:
    if(buf[i]=='P'){
        state=3;
    }else{
        state=0;
    }
    break;
case 3:
    if(buf[i]=='R'){
        state=4;
    }else{
        state=0;
    }
    break;
case 4:
    if(buf[i]=='M'){
        state=5;
    }else{
        state=0;
    }
    break;
case 5:
    if(buf[i]=='C'){
        state=6;
    }else{
        state=0;
    }
    break;
case 6:
    if(buf[i]==',' ){
        state=7;
    }else{
        state=0;
    }
    break;
case 7: //<1>
    for(;i<nread;i++){
        if(buf[i]==',' ){
            state=8;
            break;
        }
    }
}
```



```
    }  
    }  
    break;  
case 8: // <2>  
    for(; i < nread; i++) {  
        if(buf[i] == ',' || buf[i] == '\n') {  
            state = 9;  
            break;  
        }  
    }  
    break;  
case 9: // <3>  
    for(; i < nread; i++) {  
        if(buf[i] == ',' || buf[i] == '\n') {  
            // get longitude as double  
            if(indexgbuf > 1) {  
                gpsBuf[indexgbuf] = '\0';  
                char tempbuf[3] = {gpsBuf[0], gpsBuf[1], '\0'};  
                QString temp1(tempbuf);  
                double a = temp1.toDouble();  
                QString temp2(gpsBuf + 2);  
                double b = temp2.toDouble();  
                a += b / 60;  
                gpsinfo->longitude = a;  
            } else {  
                gpsinfo->longitude = 0;  
            }  
            // gpsinfo.longitude = 0;  
            // reinit the buf  
            indexgbuf = 0;  
  
            // go to next step  
            state = 10;  
            break;  
        } else if( (buf[i] <= '9' && buf[i] >= '0') || buf[i] == '.' ) {  
            gpsBuf[indexgbuf++] = buf[i];  
        }  
    }  
    break;  
case 10: // <4>  
    if(buf[i] == 'N' || buf[i] == 'S') {  
        gpsinfo->dla = buf[i];  
        state = 11;  
    } else if(buf[i] == ',' || buf[i] == '\n') { // nothing
```

```
        gpsinfo->dla='p';
        state=12;
    }else{// error handling
        state=0;
    }
    break;
case 11:
    if(buf[i]=='(',')'){
        state=12;
    }else{//error handling
        state=0;
    }
    break;
case 12://<5>
    for(;i<nread;i++){
        if(buf[i]=='(',')'){
            //get latitude as double
            if(indexgbuf>1){
                gpsBuf[indexgbuf]='\0';
                char tempbuf[3]={gpsBuf[0],gpsBuf[1],'\0'};
                QString temp1(tempbuf);
                double a = temp1.toDouble();
                QString temp2(gpsBuf+2);
                double b = temp2.toDouble();
                a +=b/60;
                gpsinfo->latitude = a;
            }else{
                gpsinfo->latitude = 0;
            }
            //gpsinfo.latitude = 0;
            //reinit the buf
            indexgbuf=0;

            //go to next step
            state=13;
            break;
        }else if( (buf[i]<='9' && buf[i]>='0') || buf[i]=='.' ){
            gpsBuf[indexgbuf++]=buf[i];
        }
    }
    break;
case 13://<6>
    if(buf[i]=='W' || buf[i]=='E' ){
        gpsinfo->dlo=buf[i];
    }
```

```

        state=14;
    }else if(buf[i]==' '){ //nothing
        gpsinfo->dlo='p';
        state=15;
    }else{//error handling
        state=0;
    }
    break;
case 14:
    if(buf[i]==' '){
        state=15;
    }else{//error handling
        state=0;
    }
    break;
case 15: // <7>
    for(;i<nread;i++){
        if(buf[i]==' '){
            state=16;
            break;
        }
    }
    break;
case 16:// <8>
    for(;i<nread;i++){
        if(buf[i]==' '){
            //get latitude as double
            gpsBuf[indexgbuf]='\0';
            QString temp(gpsBuf);
            gpsinfo->angel = temp.toDouble();
            gpsinfo->completed=true;

            //begin-----for test
            printf("\n\ninformation:  %f%c, %f%c, angel:%f\n\n",gpsinfo->latitude,gpsinfo->dla,gpsinfo->longtitude,gpsinfo->dlo,gpsinfo->angel);

            QString aaa = QString("\n\ninformation:  %1%2, %3%4, angel:%5\n\n").arg(gpsinfo->latitude).arg(gpsinfo->dla).arg(gpsinfo->longtitude).arg(gpsinfo->dlo).arg(gpsinfo->angel);
            writeToTxt(aaa);
            //end-----for test

            //reinit the buf
            indexgbuf=0;

```

```

        //go to start (remaining is not important)
        state=0;
        break;
    }else if( (buf[i]<='9' && buf[i]>='0') || buf[i]=='.' ) {
        gpsBuf[indexgbuf++]=buf[i];
    }
}
break;

}

}

void supervisionCK() {
    if( (nread=read(gpsfd, buf, 1023))>0 ) {
        buf[nread+1]='\0';
        printf("\n\n----return:::----- %s \n\n", buf);
    }
    for(int i=0;i<nread;i++) {
        switch(statespv) {
            case 0:
                if(buf[i]=='+') {
                    statespv=1;
                }else if(buf[i]=='N') {
                    statespv=10;
                }
                break;
            case 1:
                if(buf[i]=='C') {
                    statespv=2;
                }else {
                    statespv=0;
                }
                break;
            case 2:
                if(buf[i]=='L') {
                    statespv=3;
                }else {
                    statespv=0;
                }
                break;
            case 3:

```

```

        if(buf[i]=='I'){
            statespv=4;
        }else{
            statespv=0;
        }
        break;
    case 4:
        if(buf[i]=='P'){
            statespv=5;
        }else{
            statespv=0;
        }
        break;
    case 5:
        if(buf[i]==':'){
            statespv=6;
        }else{
            statespv=0;
        }
        break;
    case 6:
        if(buf[i]==' '){
            statespv=7;
        }else{
            statespv=0;
        }
        break;
    case 7:
        if(buf[i]=='""){
            statespv=8;
        }else{
            statespv=0;
        }
        break;
    case 8: // read the phone number
        for(;i<nread;i++){
            if(buf[i]!='""){
                supervibuf[indexsbuf++]=buf[i];
            }else{
                supervibuf[indexsbuf]='\0';
                printf("\n-----phone number: %s-----\n",supervibuf);
                phoneNum = QString(supervibuf);
                if(ccVerifyNumber(phoneNum) && onlyOne ){
                    prespot=true;
                }
            }
        }
    }
}

```

```

        printf("\n-----number
valid-----\n\n-----number
valid-----\n\n-----number
valid-----\n\n-----number valid-----\n");
        onlyOne=false;
    }
    ccPhoneHangUp();

    indexsbuf=0;
    statespv=0;
    break;
}
}
break;

case 10:
    if(buf[i]=='0'){
        statespv=11;
    }else{
        statespv=0;
    }
    break;
case 11:
    if(buf[i]==' '){
        statespv=12;
    }else{
        statespv=0;
    }
    break;
case 12:
    if(buf[i]=='A'){
        statespv=13;
    }else if(buf[i]=='C'){
        statespv= 18;
    }
    break;
case 13:
    if(buf[i]=='N'){
        statespv=14;
    }else{
        statespv=0;
    }
    break;
case 14:

```

```
        if(buf[i]=='S') {
            statespv=15;
        }else{
            statespv=0;
        }
        break;
case 15:
    if(buf[i]=='W') {
        statespv=16;
    }else{
        statespv=0;
    }
    break;
case 16:
    if(buf[i]=='E') {
        statespv=17;
    }else{
        statespv=0;
    }
    break;
case 17:
    if(buf[i]=='R') {
        statespv=0;
        ccCallPolice("110");
    }else{
        statespv=0;
    }
    break;
case 18:
    if(buf[i]=='A') {
        statespv=19;
    }else{
        statespv=0;
    }
    break;
case 19:
    if(buf[i]=='R') {
        statespv=20;
    }else{
        statespv=0;
    }
    break;
case 20:
    if(buf[i]=='R') {
```

```
        statespv=21;
    }else{
        statespv=0;
    }
    break;
case 21:
    if(buf[i]=='I'){
        statespv=22;
    }else{
        statespv=0;
    }
    break;
case 22:
    if(buf[i]=='E'){
        statespv=23;
    }else{
        statespv=0;
    }
    break;
case 23:
    if(buf[i]=='R'){
        statespv=0;
        if(aCount.elapsed()>25000){
            //ccCallPolice("110");
        }
    }else{
        statespv=0;
    }
    break;
    }
}

}

void ccPhoneDataInit(){
    phoneData << "13141284196" <<"18911961729";
}

void ccPhoneHangUp(){
    strcpy(buffer,"ATH");// set coming call display
    buffer[3]='\n';
    Length=4;
    nByte = write(gpsfd, buffer ,Length);
}
```



```

void ccDialPhone(QString num) {
    strcpy(buffer, "ATD");// set coming call display
    int i=0;
    for(;i<num.length();i++) {
        buffer[i+3]=*(num.mid(i,1).toLatin1().data() );
    }
    buffer[i+3]=' ';
    buffer[i+4]='\n';
    Length=i+4+1;
    nByte = write(gpsfd, buffer ,Length);
}

bool ccVerifyNumber(QString num){
    QList<QString> :: Iterator it = phoneData.begin(), itend = phoneData.end();
    int i=0;
    for(;it != itend;it++,i++){
        if(*it == num) {
            return true;
        }
    }
    return false;
}

void ccAlarm(QString num) {
    ccDialPhone(num);
    aCount.start();
}

void ccSpeechInit() {    //remain to alter~~~~~
    char *dev0  ="/dev/ttyUSB0";
    char *dev1  ="/dev/ttyUSB1";
    char *dev2  ="/dev/ttyUSB2";
    char *dev3  ="/dev/ttyUSB3";
    char *dev4  ="/dev/ttyUSB4";
    if((speechfd = OpenDev(dev0) )== -1)
        if((speechfd = OpenDev(dev1) )== -1)
            if((speechfd = OpenDev(dev2) )== -1)
                if((speechfd = OpenDev(dev3) )== -1)
                    if((speechfd = OpenDev(dev4) )== -1) {
                        printf("speech port open failed!!!!");
                        writeToTxt("speech port open failed!!!!");
                    }

    printf("speech port open succeed!!!!");
}
    
```

```
set_speed(speechfd, 9600);
if (set_Parity(speechfd, 8, 1, 'N') == false) {
    printf("Set Parity Error\n");
    exit (0);
}
}

void ccGeniunoSupervision() {
    if( (nread=read(fd, geniunobuf, 20))>0 ) {
        geniunobuf[nread+1]='\0';
        printf("\n\n----return:::----- %s \n\n", buf);
    }
    for(int i=0;i<nread;i++){
        switch(stategeniuno) {
            case 0:
                if(geniunobuf[i]=='o') {
                    stategeniuno=1;
                }
                break;
            case 1:
                if(geniunobuf[i]=='v') {
                    stategeniuno=2;
                }else{
                    stategeniuno=0;
                }
                break;
            case 2:
                if(geniunobuf[i]=='e') {
                    stategeniuno=3;
                }else{
                    stategeniuno=0;
                }
                break;
            case 3:
                if(geniunobuf[i]=='r') {
                    stategeniuno=0;
                    ccAlarm(ownerPhone);// receive the failing alarm and call the
owner
                }else{
                    stategeniuno=0;
                }
                break;
        }
    }
}
```

```
}  
}  
  
void ccCallPolice(QString num) {  
    QTime delay;  
    printf("11111111111111111111111111111111\n");  
    //ccDialPhone(num);  
  
    //护航机器人自动报警  
    nByte = write(speechfd, sound_announcement ,24);  
    delay.start();  
    while(delay.elapsed()<4000);  
  
    for(int k=0;k<2;k++){  
        //经度  
        nByte = write(speechfd, sound_latitude ,10);  
        delay.start();  
        while(delay.elapsed()<2000);  
  
        //value  
        double xx=230.11;  
        QString latitude = QString::number(xx);  
        for(int l=0;l<latitude.length();l++){  
            char c= *(latitude.mid(l,1).toLatin1().data() );  
            if(c !='.')  
                nByte = write(speechfd, sound_number[c-'0'] ,7);  
            else  
                nByte = write(speechfd, sound_point ,7);  
            delay.start();  
            while(delay.elapsed()<1000);  
        }  
        delay.start();  
        while(delay.elapsed()<500);  
  
        //纬度  
        nByte = write(speechfd, sound_longitude ,10);  
        delay.start();  
        while(delay.elapsed()<2000);  
  
        //value  
        double xxx=320.11;
```

```
QString longitude = QString::number(xxx);
for(int l=0;l<longitude.length();l++){
    char c=(longitude.mid(l,1).toLatin1().data() );
    if(c !='.')
        nByte = write(speechfd, sound_number[c -'0' ] ,7);
    else
        nByte = write(speechfd, sound_point ,7);
    delay.start();
    while(delay.elapsed()<1000);
}
delay.start();
while(delay.elapsed()<500);

//发生事故
nByte = write(speechfd, sound_accident ,14);
delay.start();
while(delay.elapsed()<3000);
}
}
```

Genuino 101 代码:

```
#include <CurieIMU.h>
#include "CurieTimerOne.h"
#include <Servo.h>
Servo myservo; //创建一个舵机控制对象
Servo myservo2;
                // 使用 Servo 类最多可以控制 8 个舵机
int pos = 0;    // 该变量用与存储舵机角度位置
char val;//读取串口值
int pin3 = 0;
int pin6 = 0;

//用于陀螺仪判断
int calibrateOffsets = 1; // int to determine whether calibration takes place or
not
float ax0 = 0.00;
float ay0 = 0.00;
```

```
void turn(int turn){

    //转向舵机
    myservo.write(turn);

}

void keep_going() {
    analogWrite(6, pin6);
    analogWrite(3, pin3);    //调节电机占空比
}

void flag_up() {
    for(pos = 115; pos>50; pos-=1)    // goes from 180 degrees to 0 degrees
    {
        myservo2.write(pos);          // tell servo to go to position in
variable 'pos'
        delay(15);
    }
    /*
    int i =240;
    for (i;i>145;){
        i -=3;
        delay(60);
        analogWrite(2, i);
    }*/
    //analogWrite(3, 130);
    //delay(2000);
}

void flag_down() {
    //舵机转到最低
    for(pos = 50; pos<=115; pos+=1)    // goes from 180 degrees to 0 degrees
    {
        myservo2.write(pos);          // tell servo to go to position in
variable 'pos'
        delay(15);
    }

    /*
    int i =145;
    for (i;i<240;){
        i +=3;
        delay(60);
        analogWrite(2, i);
    }
    */
}
```

```
}  
*/  
// analogWrite(3, 220);  
//delay(2000);  
}  
  
//陀螺仪数据转换  
float convertRawAcceleration(int aRaw) {  
    // since we are using 2G range  
    // -2g maps to a raw value of -32768  
    // +2g maps to a raw value of 32767  
  
    float a = (aRaw * 2.0) / 32768.0;  
  
    return a;  
}  
  
//转换绝对值函数  
float convert (float abc) {  
    if (abc<0){  
        return -abc;  
    }  
    else{  
        return abc;  
    }  
}  
  
void setup() {  
    pinMode(3, OUTPUT); //电机线 1  
    pinMode(6, OUTPUT); //电机线 2  
    pinMode(5, OUTPUT); //转向舵机  
    pinMode(2, OUTPUT); //三脚架舵机  
    pinMode(12, OUTPUT); //电机驱动低使能  
    myservo.attach(5); // 该舵机由 arduino 第 10 脚控制  
    myservo2.attach(2); //三脚架舵机  
    Serial.begin(19200);  
    //analogWrite(6, 0);  
    //analogWrite(9, 0);  
    digitalWrite(12, LOW);  
    flag_up();  
    keep_going();  
    turn(117); //turn 中间为 77 左边为 60 140 右边为 90
```

```
/*
 * 陀螺仪初始化
 */
while (!Serial);    // wait for the serial port to open

// initialize device
Serial.println("Initializing IMU device...");
CurieIMU.begin();

if (calibrateOffsets == 1) {
    // use the code below to calibrate accel/gyro offset values
    Serial.println("Internal sensor offsets BEFORE calibration...");
    Serial.print(CurieIMU.getAccelerometerOffset(X_AXIS)); Serial.print("\t");
    Serial.print(CurieIMU.getAccelerometerOffset(Y_AXIS)); Serial.print("\t");
    Serial.print(CurieIMU.getAccelerometerOffset(Z_AXIS)); Serial.print("\t");
    Serial.println("");

    Serial.print("Starting Acceleration calibration...");
    CurieIMU.autoCalibrateAccelerometerOffset(X_AXIS, 0);
    CurieIMU.autoCalibrateAccelerometerOffset(Y_AXIS, 0);
    CurieIMU.autoCalibrateAccelerometerOffset(Z_AXIS, 1);
    Serial.println(" Done");

    Serial.println("Internal sensor offsets AFTER calibration...");
    Serial.print(CurieIMU.getAccelerometerOffset(X_AXIS)); Serial.print("\t");
    Serial.print(CurieIMU.getAccelerometerOffset(Y_AXIS)); Serial.print("\t");
    Serial.print(CurieIMU.getAccelerometerOffset(Z_AXIS)); Serial.print("\t");
    Serial.println("");
}
// Set the accelerometer range to 2G
CurieIMU.setAccelerometerRange(2);

}

void loop() {

    /*
     * 陀螺仪函数
     */
    int axRaw, ayRaw, azRaw;        // raw accelerometer values
    float ax, ay, az;
    float xxx,yyy;

    // read raw accelerometer measurements from device
```

```
CurieIMU.readAccelerometer(axRaw, ayRaw, azRaw);
```

```
// convert the raw accelerometer data to G's  
ax = convertRawAcceleration(axRaw);  
ay = convertRawAcceleration(ayRaw);  
az = convertRawAcceleration(azRaw);
```

```
xxx = convert(ax-ax0);  
yyy = convert(ay-ay0);
```

```
if (xxx>0.6||yyy>0.6) {  
    Serial.println("over");  
    Serial.println();  
}  
ax0 = ax;  
ay0 = ay;
```

```
// put your main code here, to run repeatedly:  
if (Serial.available())  
{
```

```
val=Serial.read(); //读取串口值  
if (val=='l'){  
    //turn (95); //左转  
    turn (130);  
    Serial.print(val);  
    //delay(500);  
}
```

```
else if (val=='r'){  
    //turn (61); //右转  
    turn(100);  
    Serial.print(val);  
    //delay(500);  
}
```

```
//直行  
else if (val == 'q'){  
    //turn(80);  
    turn(117);  
    Serial.print(val);  
}
```

```
//放下牌子  
else if(val == 'e'){
```



```
flag_down();
Serial.print(val);
}

else if (val=='u') {
    flag_up();
}
else if (val=='s') {
    //从直行到停车, 需要先减速
    if (pin3>39) {
        int i = 4;
        for(i;i>0;i--) {
            pin3 -=10;
            keep_going();
            delay(100);
        }
    }
    //从倒车倒停车, 要先减速
    if(pin3<1&&pin6>0) {
        int i = 4;
        for (i;i>0;i--) {
            pin6 -=10;
            keep_going();
            delay(100);
        }
    }

    pin3 = 0;
    pin6 = 0;
    keep_going();
    Serial.print(val);
}
//以一挡速度前进
else if(val == 'a') {
    //如果是从倒车直接道之行, 那么需要先给倒车减速
    if (pin3<1&&pin6>0) {
        int i = 5;
        for (i ; i>0; i--) {
            pin6 -=10;
        }
        keep_going();
        delay(100);
    }
}
```

```
pin3 = 40;
pin6 = 0;
keep_going();
Serial.print(val);
}
//以二档速度前进
else if (val == 'b'){
    //如果是从倒车直接道之行, 那么需要先给倒车减速
    if (pin3<1&&pin6>0){
        int i = 5;
        for (i ; i>0; i--){
            pin6 -=10;
        }
        keep_going();
        delay(100);
    }
    pin3 = 50;
    pin6 = 0;
    keep_going();
    Serial.print(val);
}
//以三档速度前进
else if (val == 'c'){
    //如果是从倒车直接道之行, 那么需要先给倒车减速
    if (pin3<1&&pin6>0){
        int i = 5;
        for (i ; i>0; i--){
            pin6 -=10;
        }
        keep_going();
        delay(100);
    }
    pin3 = 60;
    pin6 = 0;
    keep_going();
    Serial.print(val);
}
else if (val == 'd'){
    //如果是从直行直接变为倒车, 则需要 0.4s 减速时间.
    if (pin3>39){
        int i = 3;
        for(i;i>0;i--){
            pin3 -=10;
            keep_going();
        }
    }
}
```

```
        delay(100);  
    }  
}  
  
pin3 = 0;  
pin6 = 50;  
keep_going();  
  
Serial.print(pin3);  
Serial.print(pin6);  
}  
else{  
    keep_going();  
}  
//delay(1000);  
}  
}
```