技术说明

题 目：

学生姓名： 学号：

所在系院： 专业：

入学时间： 年 9 月

导师姓名： 职称/学位

导师所在单位：

完成时间： 2017 年 5 月

# 基于CMOS传感器的轮式机器人循迹算法的研究与设计

摘要：本项目是为参加第十一届全国大学生智能汽车而展开的。项目的内容是设计一个智能汽车，通过摄像头识别赛道。自主的在赛道上行驶。其中涉及到自动化专业的方方面面。其设计内容涵盖了控制、模式识别、传感技术、汽车电子、电气、计算机、机械、能源等多个学科的知识，对学生的知识融合和实践动手能力的培养，具有良好的推动作用。在本次比赛中， 本组使用大赛组委会统一提供的竞赛车模，采用恩智浦32位微控制器 MK60DN512ZVLQ10作为核心控制单元，自主构思控制方案及系统设计，包括传感器信号采集处理、控制算法及执行、动力电机驱动等，最终实现一套能够自主识别路线，并且可以实时输出车体状态的智能车控制系统。在制作小车的过程中，我们对小车的整体构架进行了深入的研究，分别在机械结构、硬件和软件上都进行过改进，硬件上主要是考虑并实践各种传感器的布局，改进驱动电路，软件上采用了多种算法结合的方式，小车的寻线方式采用适应性较强的优化的位置加权的方法。控制算法上，从PID到Bang-Bang,再到模糊PID都进行了一些研究。

关键词：摄像头传感器； PID控制器； 图像识别 ；寻迹机器人

**Research and design of wheeled robot tracking algorithm based on CMOS sensor**

**Abstract:** this project is to participate in the Eleventh National College Students intelligent vehicle launched. The content of the project is to design an intelligent car that identifies the track through the camera. Run autonomously on the track. It involves all aspects of automation. The content covers the control design, pattern recognition, sensor technology, automotive electronics, electrical, computer, machinery, energy and other disciplines of knowledge, to cultivate students' practical ability and knowledge fusion, has a good role in promoting. In this game, the use of the organizing committee to provide unified competition models, using NXP 32 bit microcontroller MK60DN512ZVLQ10 as the core control unit, independent design and system control scheme, including sensor signal acquisition and processing, control algorithm and implementation, motor driven power, finally realizes a set to be able to independently identify the route, and can intelligent vehicle body control system real-time output state. In the process of making cars in the frame of the car, we conducted in-depth research, in the mechanical structure, hardware and software have been improved, the hardware is considered and the practice of various kinds of sensor layout, improved driving circuit, the software uses a variety of algorithm combination method with position weighted line search the car is optimized with strong adaptability of the. In the control algorithm, some researches have been done from PID to Bang-Bang, and to fuzzy PID.

**Key words:** camera sensor; PID controller; image recognition; Tracking robot

**目 录**

[1 引言 1](#_Toc482884419)

[1.1 智能车大赛简介 1](#_Toc482884420)

[1.2 智能汽车制作要求及任务 1](#_Toc482884422)

[2 车模组装与机械调试 3](#_Toc482884426)

[2.1车模机械系统的组成 3](#_Toc482884427)

[2.2 车模底盘及转向机构调校 4](#_Toc482884428)

[2.2 各测控模块的安装 6](#_Toc482884429)

[2.2.1 摄像头的安装 . 6](#_Toc482884430)

[2.2.2编码器的安装 6](#_Toc482884431)

[2.2.3 PCB板的安装 6](#_Toc482884432)

[2.2.4 电池的安装 7](#_Toc482884433)

[2.2.5 整车设计效果图 7](#_Toc482884434)

[3 硬件电路设计 8](#_Toc482884435)

[3.1硬件电路整体架构 8](#_Toc482884436)

[3.2 电源管理模块 8](#_Toc482884437)

[3.3 图像采集模块 9](#_Toc482884438)

[3.4 电机驱动模块 9](#_Toc482884439)

[3.5编码器模块 10](#_Toc482884440)

[3.6 人机交互模块 11](#_Toc482884441)

[4 软件系统设计 12](#_Toc482884442)

[4.1 软件系统设计方案 12](#_Toc482884443)

[4.2 车模行驶信号获取与处理 13](#_Toc482884444)

[4.2.1赛道图像获取 13](#_Toc482884445)

[4.2.2弯道的补线 13](#_Toc482884446)

[4.2.3十字的补线 15](#_Toc482884447)

[4.2.4车模行驶偏差获取与处理 16](#_Toc482884448)

[4.2.3 障碍的识别 18](#_Toc482884449)

[4.2.3 起跑线的识别 19](#_Toc482884450)

[4.2.4 坡道的识别 20](#_Toc482884451)

[4.3 车模行驶状态控制 20](#_Toc482884452)

[4.3.1车模速度控制 20](#_Toc482884453)

[4.3.2车模转向控制 . 22](#_Toc482884454)

[4.4程序流程图 24](#_Toc482884455)

[5 系统调试与运行 25](#_Toc482884457)

[5.1 IAR Embedded Workbench IDE调试软件 25](#_Toc482884458)

[5.2 串口调试工具 25](#_Toc482884459)

[5.3 系统运行与分析改进 26](#_Toc482884460)

[6 结论 28](#_Toc482884461)

[参考文献 29](#_Toc482884462)

致 谢 [30](#_Toc482884463)

[附 录 31](#_Toc482884463)

# 1 引言

## 1.1 智能车大赛简介

全国大学生“恩智浦”杯智能汽车竞赛起源于韩国，是韩国汉阳大学汽车控制实验室在恩智浦半导体公司资助下举办的以恩智浦单片机为核心的大学生课外科技竞赛。组委会提供一个标准的汽车模型、直流电机和可充电式电池，参赛队伍要制作一个能够自主识别路径的智能车，在专门设计的跑道上自动识别道路行驶，最快跑完全程而没有冲出跑道并且技术报告评分较高为获胜者。[1]其设计内容涵盖了控制、模式识别、传感技术、汽车电子、电气、计算机、机械、能源等多个学科的知识，对学生的知识融合和实践动手能力的培养，具有良好的推动作用。[2]

## 1.2 智能汽车制作要求及任务

在本次比赛中，本组使用大赛组委会统一提供的竞赛车模，采用恩智浦公司的单片机作为处理器，通过摄像头识别赛道，使得小车自主的赛道上行驶一圈，速度最快者胜。比赛要求智能车使用的车模必须是大赛组委会指定的B型车模，不允许对车模的机械结构进行改动，只允许进行打孔，安装螺丝等操作。车模所使用的电路板必须为自行设计制作。比赛必须采用摄像头作为传感器识别赛道。

智能车制作过程中，主要任务分为机械、硬件电路、软件程序三大块。需要对车模的机械结构进行调校，新的车模机械结构可能不是最完美的，在高速运行下，稳定性较差。车模调校的结果直接影响着车模的运行状态。

硬件电路方面需要自行设计电路板，电路板需要包含电源单元，处理单元，电机控制单元。人机交互单元。必须在电路板铜层上刻上学校名称以及队伍名称。电路板上不允许使用成品模块。

软件方面需要自行设计软件方案，编写程序，控制小车在赛道上行驶。三大部分设计好后需要对智能车进行组装，各部件的安装位置需要经过测量，必须安装对称，否则造成车模重量分配不均，影响运行效果。最重要的是需要对车模进行调试。逐渐发现问题解决问题，直到小车达到最好的运行状态。

# 

# 2 车模组装与机械调试

## 2.1车模机械系统的组成

车模机械系统如图2.1所示。

车模机械系统

各个测控模块

车模底盘

转向机构

图2.1 车模机械系统结构图

智能车机械系统包含了测控模块、转向机构、车模的底盘三个部分。底盘是构成整个车模的主体，其材料由玻璃纤维构成。底盘上搭载着各个模块及零件，是整个车模最为重要的部件之一，底盘的质量直接影响着车模的质量。底盘上承载了车模的驱动机构。车模采用后轮驱动的形式，驱动电机和传动机构以及驱动轮通过机械连接的方式固定在底盘上。底盘前部承载着车模的转向机构，通过机械连接固定在底盘上。[3]同时底盘起着支撑整个车体的作用，是整个车模系统的主要承力机构。

转向机构包含了轮鞘、转向杯、轴承几个部分。转向机构安装在底盘的前部，驱动前轮运动，并控制车模的运动方向。其安装角度很有讲究，科学的安装角度直接影响着小车拐弯时的流畅性和车模整体的运行效率。机械上的问题往往都出现在转向机构上，这部分也是车模最脆弱的一部分。

测控模块包含了摄像头、PCB板、人机交互模块和编码器四个部分。摄像头是整个车模获取赛道信息的唯一器件，摄像头的性能决定了车模运行的速度，好的摄像头能够获取更加准确的信息，使得车模精准的运行。PCB板是控制电路的主要器件，包含了电源、电机驱动、处理电路单元。PCB板的稳定性决定了小车的稳定性。人机交互模块是整个小车和用户交流的一个器件，用户通过人机交互模块可以知道小车运行的信息，并且可以更改小车运行参数，做到脱机调试。

## 2.2 车模底盘及转向机构调校

本届智能车大赛摄像头组所使用的是大赛组委会指定的B车模，原装的B车模上安装有减震设备，为了提高小车的稳定性，我们拆除了减震，更改了舵机的安装位置，降低了底盘高度。在车模高速行驶过程中，底盘越低，车模重心越低，车模运转也会更加平稳。

舵机安装方式如图2.2所示。



图2.2 舵机的安装

舵机采取如下装法：舵机直立安装，并靠近转向桥。舵机的安装方式直接关系到是否能快速灵敏地转向，如果舵机调整不到位，将很大程度上限制转向角度和转向响应速度。舵机的安装有两种形式，一种是卧式安装，将舵机横放在底盘上，通过舵机输出轴驱动转向机构。但是卧式安装舵机的高度过低，需要增加一个机械连杆，这样稳定性会有所下降。而且舵机卧式安装的话，由于舵机宽度较大，会比较占空间，并且不好固定。所以我们采取了另一种立式的安装方式，立式安装就是将舵机站立着安装在底盘上。这种安装方式方便固定，舵机输出轴的转动方向和转向机构的运动方向在同一轴线上，这样减少了传动机构，增大了前轮转向的力矩。转向会更加灵敏。

舵机安装完成后，要想使车模更加稳定的运行，还需要对车模的前轮进行调校。为了保持汽车直线行驶的稳定性、转向的轻便性和减少轮胎与机件的磨损， 转向车轮、转向节和前轴三者与车架安装时保持一定的相对位置，这种具有一定相对位置的安装称为转向轮定位，也称前轮定位。前轮定位的内容包括：主销后倾、主销内倾、虚位调整、前轮外倾和前轮前束。[4]

1.主销后倾

主销装在前轴上，在汽车纵向平面内，其上端略向后倾斜，这种现象称为主销后倾。在纵向垂直平面内，主销轴线与垂线之间夹角γ叫主销后倾角。主销后倾角一般为 0.5°～3°。主销后倾作用是保持汽车直线行驶的稳定性，并力图使转弯后的前轮自动回正。[5]

2．主销内倾角

主销安装在前轴上，在汽车的横向平面内，其上端略向内倾斜，这种现象称为主销内倾。在横向垂直平面内，主销轴线与垂线之间的夹角β叫做主销内倾角。主销内倾角一般为 6°～9°。主销内倾的作用是使前轮自动回正、转向轻便，并减小汽车行驶时路面通过车轮传给转向机构的冲击力。

3.前轮外倾角

转向轮安装在车桥上，其旋转平面上方略向外倾斜，这种现象称为前轮外倾。 前轮旋转平面与纵向垂直平面之间的夹角α叫做前轮外倾角。前轮外倾角一般 为 0.5°～2°。 前轮外倾的作用是提高前轮工作的安全性和转向操纵轻便性。

4.虚位的消除

虚位是指车模前轮个转向轴之间存在着缝隙，导致前轮晃动。晃动的前轮会使得车模转向时出现前轮不稳的现象，所以需要消除前轮的虚位。当车模以较高的速度运行在赛道上，在弯道时车模所需要的向心力是完全依靠轮胎的摩擦力来提供。当前轮拐弯时晃动，容易破坏已经调好的前轮转向角，导致前轮各转向力矩不平衡，最终导致的结果是摩擦力不够，车模侧滑，从而整体上降低了车模的运行速度和流畅性。想要消除前轮虚位需要对各个部件进行处理。前轮的各个转向轴之间存在的缝隙使用锡箔纸包裹，这样就可以消除轴缝隙。各个球头之间的缝隙也会导致较大的虚位，需要用胶包裹住球头，减小球头活动范围，但是不可以将球头封死，否则会导致转向力矩不足。经过仔细的处理，虚位的影响降到了最低。[6]

## 2.2 各测控模块的安装

### 2.2.1 摄像头的安装 .

摄像头采用的是OV公司生产的OV7725型摄像头，这是一种30万像素带硬件二值化的黑白摄像头。最高支持150帧的帧率，可以提供稳定的画面。[7]为了保证摄像头拥有足够的视野，我们将摄像头尽可能的安装高，这样就可以保证以一定的角度俯视地面，一定程度上可以解决反光的影响。小车运行过程中会产生明显的震动，会引起摄像头的震动，摄像头一震动小车图像也会跟着震动，小车也会更加剧烈的震动，如此就会形成正反馈，造成不好的后果。因此需要减轻摄像头的震动，所以摄像头的安装支架采用三角形结构，大大增强了稳定性。

### 2.2.2编码器的安装

编码器采用的绝对式编码器，型号为泰庆公司生产的小钢炮系列迷你编码器。该编码器是一种磁电式的编码器，输出线数为512线。编码器安装位置不要贴近电机，以免造成干扰。编码器原装的接口是排座接口，为了减轻重量，我们使用排线接口作为编码器接口。编码器接口如图2.3所示。



图2.3 编码器接口

### 2.2.3 PCB板的安装

大赛要求车模的PCB板必须为自己设计制作。PCB板上包含电源，单片机最小系统，电机驱动和人机交互四个模块。[8]为了减小体积，我们将电机驱动和单片机最小系统以及电源电路集成到了一片板子上，安装时使用尼龙柱固定在底盘上。PCB板的安装如图2.4所示。

图2.4 pcb的安装

### 2.2.4 电池的安装

为了增大车模的稳定性，提高前轮对地面的压力，增强转向能力。我们将电池安装在了车模的前部，并用扎带固定好。电池和车模底盘之间的缝隙要尽可能的小，以降低车模的整体重心。

### 2.2.5 整车设计效果图

图2.5 整车效果图

# 

# 3 硬件电路设计

## 3.1硬件电路整体架构

硬件电路包含了电源管理，图像采集，电机驱动，编码器和人机交互五个模块，其整体架构如图3.1所示：

硬件电路架构

编码器模块

人机交互模块

电源管理模块

图像采集电路

电机驱动模块

图3.1 硬件电路架构图

## 3.2 电源管理模块

电源管理模块上我们有3.3v和5v两种电压芯片，分别使用tps7333和tps7350作为稳压芯片。TPS7333是TI公司的一款超低压差LDO，该芯片属于TPS73XX系列，后两位序号XX表示该芯片的输出电压值。如TPS7333表示3.3V，TPS7348表示4.85V，TPS7350表示5.0V，该系列还有一个输出电压可调的版本，后缀为01。[9]图3.2是tps7350的最小系统原理图。

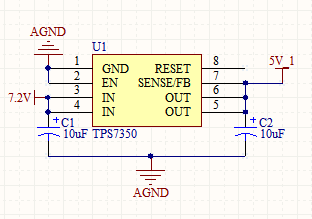


图3.2 电源管理电路

## 3.3 图像采集模块

图像采集模块我们使用ov7725摄像头采集赛道图像，为了减轻单片机工作压力，我们使用比较器对摄像头采集到的图像进行硬件二值化，这样可以大大减轻单片机的工作负担，提高程序执行速度。[10]摄像头接口和单片机之间以并口的方式传送数据，为了增大传输速度，图像数据通过单片机的DMA接口传输。图像采集模块电路如图3.3所示。

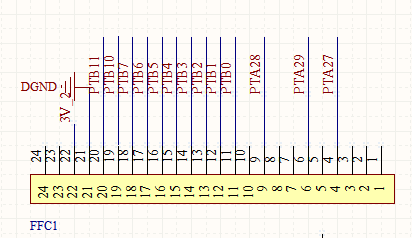


图3.3 摄像头接口电路

## 3.4 电机驱动模块

我们使用四个MOS管构成桥式电路，驱动电机。由于本届大赛摄像头组所使用的B车模电机为540电机，功率较大，工作电流也较大，选用了lr7843mos芯片作为驱动芯片。[11]为了防止电机驱动电路对单片机造成干扰，在驱动电路的输入级接入了逻辑门电路进行隔离。隔离电路如图3.4。

图3.4 隔离电路

## 3.5编码器模块

为了使小车更轻，我们选用了泰庆电子生产的512线迷你编码器。[12]该编码器特点是体积非常小巧，输出性能稳定，可以使用3.3v和5v供电。编码器可以输出两相脉冲，经过单片机正交解码，我们就可以得到具有正负的脉冲数。编码器安装在车轴正上方，如图3.5所示。



图3.5 编码器的安装

在编码器电路中，编码器的输出信号端接了10K欧姆的上拉电阻，以增强编码器信号的稳定性。编码器接口如图3.6所示。

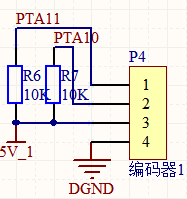


图3.6 编码器电路

## 3.6 人机交互模块

人机交互模块我们使用五向按键和0.92阴吋的oled液晶构成人机交互模块。该液晶为128\*64的分辨率，可以显示两种颜色。[13]五向按键和液晶模块构成类似于菜单栏的人机交互界面，人机交互界面的作用是显示小车运行的参数，并且可以通过一个按键更改小车运行的速度等参数，所更改的数据存入flash。人机交互模块如图3.7所示。

图3.7 人机交互模块

# 

# 4 软件系统设计

## 4.1 软件系统设计方案

软件是整个小车系统的核心。软件系统包括了摄像头的图像获取与处理、人机交互、车模控制三个部分。软件系统结构如图4.1所示：

软件系统

图像获取与处理

图像处理

图像采集

偏差提取

速度控制

人机交互模块

车模控制

方向控制

图4.1 软件系统结构

人机交互是人和小车之间交互的一个界面，控制着小车的液晶和按键，将小车运行的参数显示在液晶上，也可以显示小车图像处理后的画面，通过观察画面就可以查找出图像处理的问题，方便做出改进。人机交互中包含了一个五向按键。程序读出五向按键的状态，判断是否有按键按下，从而更改小车的参数，将更改的参数写入液晶显示并且同时保存在flash中。

图像获取与处理包含了图像采集、图像处理、偏差提取三个部分。图像采集是将摄像头拍摄的图像采集到内存中，并将其转换为坐标的形式。摄像头图像是通过单片机DMA通道采集的，通过摄像头采集程序采集图像数据，将采集到的图像保存在一个二维数组中，就可以将图像问题转换成数学问题进行处理。图像的处理主要是进行赛道中间线的提取与补线，赛道元素的识别，赛道弯曲程度的计算。图像处理是整个软件系统的核心。偏差提取是根据图像处理的结果提取车模对应于赛道的偏差，小车依据计算得到偏差要能够以最短的路径行驶。

车模控制包含了车模的速度控制和转向控制两个部分。速度控制函数是用来控制小车在直道加速弯道减速的，为了保证小车能够以最快的速度运行，需要保证小车在直道上立即加速，弯道上立即减速，在不同的弯道上以不同的速度行驶。转向控制是整个车模转向的控制函数，要根据图像处理的结果提取出小车的行驶偏差，控制小车在赛道上根据赛道的弯曲程度进行转向，并且保证小车运行流畅。

## 4.2 车模行驶信号获取与处理

### 4.2.1赛道图像获取

首先将采集到的图像解压为80\*60分辨率的图像，然后将整个图像存储在一个二维数组里，对赛道的分析最终转化为对数组的处理。为了提取赛道的偏差，就必须首先得到赛道的中线，综合分析后，我们采用了从中间往两边扫描的策略。当发现图像有从白跳变为黑的现象，就记录下此时的坐标，最后我们将右边界加左边界除以二，就是赛道的中间线。

### 4.2.2弯道的补线

当车模处于弯道时会出现丢边的情况，此时我们就要进行补线，补线是一项非常复杂的工作，在尝试过多种算法方案，经过十多天反复的实验后，终于取得了较好的效果。下面介绍一下采用过的几种较为有意义的方案：

方案一：此方案的思路是由于赛道的宽度是恒定的40cm，因此其对应的图像宽度也是恒定的。首先将小车摆在直道的正中间拍摄下一幅完整的照片，然后将每一行图像对应的宽度数出来，将他的宽度保存在一个数组中。当程序从图像的中间往两边扫描时，一直扫到图像的边界都没有扫到赛道的边界，则判定本行丢边，那么此时就要利用之前保存的每一行赛道对应的图像的宽度，如果丢了右边就将左边界加上整个赛道的宽度就得到了右边界，如果丢了左边界就将右边界减去整个赛道的宽度就得到了左边界。如此就能够大致将丢失的边界补回来，然后再将右边界加左边界除以2，就得到了中间线。但是我们预先保存的赛道宽度是在直道上的宽度，而在弯道时丢失边界处的图像宽度会变宽。图4.2是采用这种办法后的效果。

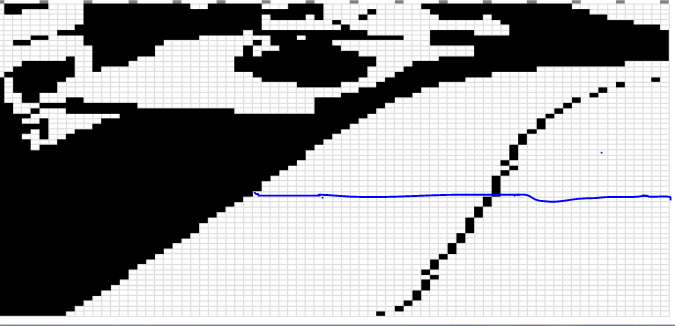


图4.2 弯道补线方案一效果

在图中可以看到画蓝线的那一条线是丢失了右边界的，因为此时赛道是弯曲的，这一行此时对应的宽度（也就是蓝线的长度）是明显大于在直道上这一行的宽度的，如果此时我们还按照上述方法补线，将左边界加上这一行对应直道上的宽度，那么明显我们补上的右边界到左边界的宽度是小于实际赛道宽度的，也就是没补够。此时的误差如果车速过快将会有不利的影响，为此我们尝试着将每一行补上去的宽度都加上一两个点，试图通过加上一个预估值来补偿一下误差，但是赛道错综复杂，每种赛道对应的误差都是不一样的，此时很难判断补偿量，补的效果也不理想，因此经过考虑决定尝试新的算法。

方案二：本方案可以很好的补齐图像。在上图中可以看到赛道往往只会丢失一边，比如上图中右边赛道边界丢失了但是左边依然在，因此我们考虑能否根据左边界变化的趋势来补上右边界。经过研究我们发现，赛道的两边界是平行的，在图像中也应该大致上平行。为此我们研究出了这种方法，当右边界从某一点开始丢失时，我们记录下丢失的这一个点的行坐标，然后计算出左边界下一行的点的相对于这一行点的偏移量，此时右边界丢失的这一点相对于上一点的偏移量应该和左边界是一样的，我们将右边界上一点没有丢失的坐标加上这个偏移量就得到了丢失点的准确坐标，以此类推，就可以得到整个边界丢失的坐标。如果丢了左边界也是如此处理。经过验证发现这种方法能够非常准确的推演出丢失的边界。效果非常好。图4.3是采用这种办法后的效果。

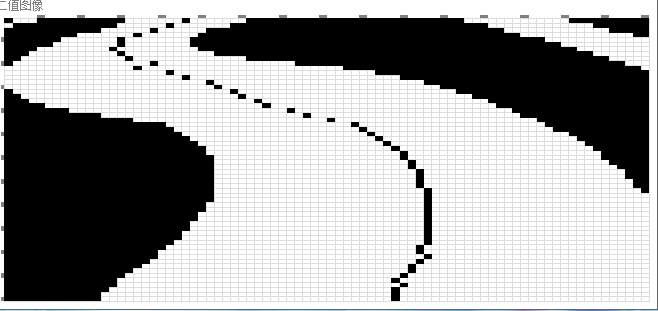


图4.3 弯道补线方案二效果

### 4.2.3十字的补线

赛道中会出现十字的情况，为此必须进行十字补线。要想进行十字补线，首先要判断出十字，十字的判断也是一个非常复杂的过程。当检测到有一行图像两边都扫不到边界，而且同一副画面中有两行及以上的行数出现了丢失一边边界的情况，并且要求这两行距离比较近，那么就可以判定为十字。判定为十字后，就需要对十字进行补线处理。十字补线涉及到非常复杂的推理和数学计算，关键要找准十字中关键的几个点。当我们发现中线有一点和边界接触了，就说明有可能是十字，此时就需要对整幅图像进行扫描。首先要找到图像中十字第一点的，然后记录下此点的坐标，将此点和图像底部的一点连接，并通过划线函数向上延伸，就可以补齐十字线。当车进入十字区域后，需要找到上十字第一点，然后将此点坐标和图像底部坐标连接，就补齐了十字图像。图4.4是十字补线的效果。

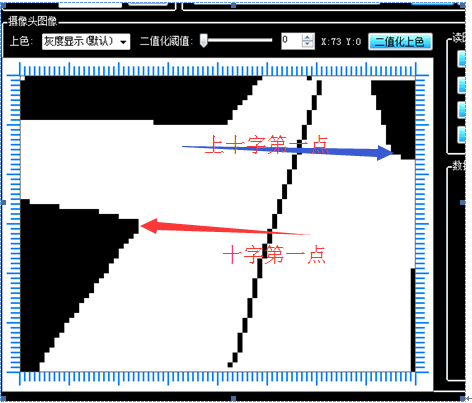


图4.4 十字补线效果

十字补线所用到的延伸函数：

void zhixian\_yanshen(int Y1,int X1,int Y2,int X2,int Y3)

{

int U,HY;

for(U=Y1;U>=Y3;U--)

{

HY=(U-Y1)\*(X2-X1)/(Y2-Y1)+X1;

currentzhongjian[U]=HY;

}

}

### 4.2.4车模行驶偏差获取与处理

偏差提取过程中，因为有弯道，车辆要拐弯，所以要提取车辆对于赛道的偏差量。为了跑的更快，车必须走最短路径所以要切内线走，这样可以以最短的路径走完全程。但是想走内切并不容易，首先得使小车沿着中间线走，这时我们选取某一行对应于中间线的偏差，调试小车直到能稳定沿着中间线走，至于选哪一行是经过多次试验总结得来的。如果小车能沿着中间线走了，我们在把中线偏差加上一个这一段中线上两点间的斜率，偏差和斜率都乘以一个权重，就是实际偏差了。至于斜率选哪两点的斜率实验出来的，对应的加权代码为my\_piancha=(int)(1.1\*currentzhongjian\_lk-0.9\*B)。加权的结果为1.1倍的点的偏差加上0.9倍的斜率，由于线性回归方程计算的斜率得到的值时反的，用减代替加，负负得正。

斜率的计算方面，传统的两点法算斜率虽然很简单，但是不可靠，因为图像是动来动去的，万一哪一点跑偏了，那斜率就错了。我们采用了线性回归方程算斜率，可以将图像中的每一点都纳入计算范围内。车模行驶过程中为了使车模用最短的时间跑完全程，要求车模走最短路径，为此，在偏差的提取上我们取离车身大约50cm处的一点的偏差和摄像头视野范围内一段中线的斜率，以一定的权重相加拟合成总的偏差，再将偏差进行PID运算即可控制舵机打角。斜率计算方法最简单的是取两点，根据两点横纵坐标直接计算出斜率，但是这种计算方法不可靠，最经典的斜率计算方法即是最小二乘法。在本程序中最小二乘法的函数是根据数学中的函数公式演化而来，其作用效果和数学公式是一样的。

最小二乘法数学公式如下：

（4-1）

在程序中累加我们使用的是循环的方式实现的，为了降低单片机执行的压力，我们没有采用浮点型计算数据,一旦遇到需要进行浮点运算的时候，就使用除法代替。

以下是小车最小二乘法计算斜率的c语言函数：

int regression(int startline,int endline)

{

if(endline>55)

endline=55;

int i;

int sumX=0,sumY=0,avrX=0,avrY=0 ;

int num=0,B\_up1=0,B\_up2=0,B\_up,B\_down;

for(i=startline;i<=endline;i++)

{

num++;

sumX+=i;

sumY+=currentzhongjian[i];

}

avrX=sumX/num;

avrY=sumY/num;

B\_up=0;

B\_down=0;

for(i=startline;i<=endline;i++)

{

B\_up1=(int)(currentzhongjian[i]-avrY);

B\_up2=i-avrX;

B\_up+=(int)(10\*(B\_up1\*B\_up2));

B\_up=B\_up/100\*100;

B\_down+=(int)(10\*((i-avrX)\*(i-avrX)));

}

if(B\_down==0)

B=0;

else

B=B\_up\*10/B\_down;

return B;

}

## 4.2.3 障碍的识别

由于赛道上存在障碍，所以需要对障碍进行处理，首先要根据图像判断出障碍。障碍的图像如图4.5：

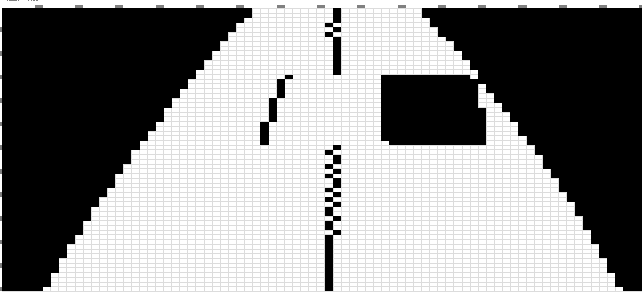


图4.5 障碍图像

当检测到车模处于直道上且整副图像呈宽窄宽变化时，就认定为障碍。检测到障碍后，只需要控制舵机打一定的角度，并保持一会，就可以绕过障碍。至于如何检测赛道变窄或者变宽，我们首先测取出车模在直道上检测到的每一行对应的宽度，作为标准宽度保存到数组里，只要与此数组进行对比就可以直道赛道是变宽还是变窄。检测到障碍后，只需要控制舵机打一定的角度，并保持一会，就可以绕过障碍。但是障碍出现在道路的右边还是左边都是有可能的，检测到障碍后还要对障碍所处的位置进行判断，由于程序已经检测到障碍部分的中线是突出的，所以只需要将突出部分的每个像素点的横坐标相加求平均值，然后和宽部分的横坐标平均值进行比较，如果为负就说明障碍在右边，如果为正就说明障碍在左边。经过验证发现这种方法很可靠，没有出现误判和漏判的状况。

## 4.2.3 起跑线的识别

赛道中存在起跑线，需要根据图像识别出起跑线起跑线的图像如4.6：

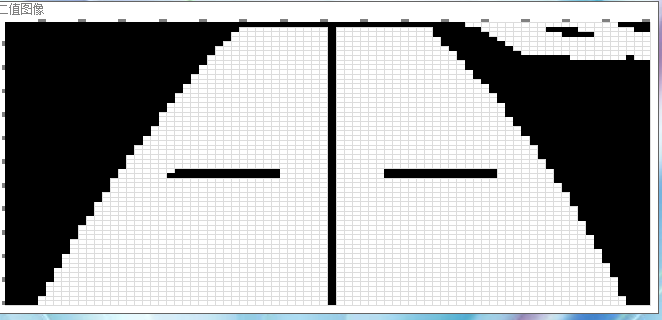


图4.6 起跑线图像

起跑线检测的思想用跳变沿的思想。检测的方法是从底部几行的中间位置开始，各往左和往右推13个像素点，然后从这个像素点的坐标开始，横坐标不变，纵坐标逐次减小，向上扫描，当发现像素点颜色出现从白变成黑的情况则记录下变化的状态，继续向上扫描，再次出现从黑变成白的跳变，且两边同时跳变，为了增大可靠性，允许两边跳变的出线一定的先后顺序，但是不允许出线较大时间间隔，则此时必然是起跑线。当检测到起跑线后会置位一个标志位，电机控制函数会控制电机反转刹车然后停车。由于车辆跑完一圈需要一定的时间，程序不需要每时每刻都检测起跑线，所以当车辆发车后10S才开始检测起跑线。这样可以降低在行驶过程中误判的几率，提高程序执行的效率。实现的方法是在定时器里增加变量，每隔5ms变量增加一，直到增加2000次，就开始执行起跑线检测程序。

## 4.2.4 坡道的识别

赛道中存在坡道，如果以过大的速度冲上坡道，有可能导致小车飞起，冲出赛道，所以检测到坡道需要减速，检测的依据是图像对称性检测。坡道图像如图4.7。

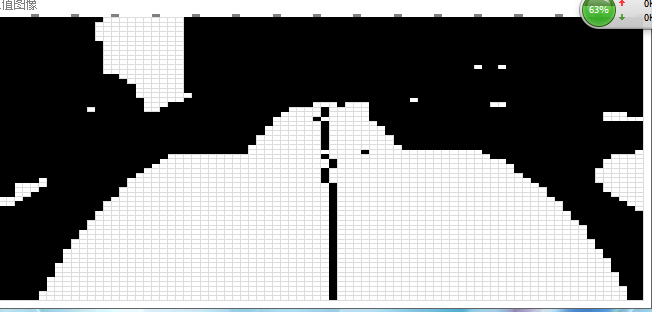


图4.7 坡道图像

从图像中可以发现图像的中心线基本上是直的，而图像赛道的边缘是非线性的，并且两边基本对称。在进行图像扫描时，每次检测到中间线是直的情况下，就开始检测两边的线性度。方法是每一个像素点相对于上一点的偏移量如果突然发生很大的跃变，就如图中一样，突然赛道收窄，并且两边的跃变程度类似，那么此时必然是坡道。电机控制函数就开始控制电机减速，并保持一段时间后，车辆恢复正常速度。中间线的线性度的检测凭借的是曲率，当曲率值很小，并且三段中间线的曲率都很小的情况下，中间线必然是直的。经过试验发现这种方法判断坡道没有出现误判的情况，判定的准确率较高。但是此种方法的适用性较低，一旦坡道的大小或者外形有所变化，就无法检测到了。

## 4.3 车模行驶状态控制

### 4.3.1车模速度控制

为了让车模整体上行驶速度更快，需要使用变速行驶的策略，既直道加速弯道减速。车模判断直道还是弯道的依据是赛道的曲率，曲率越大表示赛道越弯，根据不同的曲率给定不同的速度。由于使用数学上的曲率计算方法较为复杂，单片机执行起来较为耗时，我们简化了处理算法，即中线中某点相对于上一点每向右偏移一次，曲率加一，向左偏移一次减一，最后将整副图像的中线都判断一次就可以得到当前图像的曲率。根据曲率就可以判断赛道的缓急程度，不同的缓急程度给定不同的速度。

为了使速度控制精准，我们采用了PID控制车速。由于车模目标速度是连续变化的，是一个动态的系统，在这种情况下PID控制首先要保证的是跟随能力，所以静差的影响可以忽略，从而可以去除积分项，采用PD控制。图4.8是速度PID控制的架构图：

SV 偏差 速度

PD

电机

车轮

计算目标速度

曲率

-

编码器

图4.8 速度PID控制架构图

PID控制器数学公式为：

（4-2）

根据前人经验和调试现象发现，小车在运行过程中，速度静态误差非常小，往往不会对小车的运行产生影响，并且小车目标速度是不断不断变化的，因此在PID算法中我们将积分项系数设为0，只使用了PD控制。经过调试后发现小车能够很流畅的在道路上运行，精准的实现速度的切换。以下是PID控制车速的程序代码：

Void speed control()

{

piancha\_speed=maichongshu-speed\_now;

speed\_wucha=speed\_now-maichongshu;// bu

dwucha=speed\_wucha-lastspeed\_wucha2;

zhankongbi=last\_zhankongbi+speed\_kp\*speed\_wucha-speed\_ki\*lastspeed\_wucha+speed\_kd\*dwucha;//调速

if(zhankongbi>7000)

zhankongbi=7000;

if(zhankongbi<0)

zhankongbi=0;

if(piancha\_speed>menkan)

{

fanzhuan(dut);

}

else

zhengzhuan(zhankongbi);

last\_zhankongbi=zhankongbi;

lastspeed\_wucha3=lastspeed\_wucha2;

lastspeed\_wucha2=lastspeed\_wucha1;

lastspeed\_wucha1=lastspeed\_wucha;

lastspeed\_wucha=speed\_wucha;

}

### 4.3.2车模转向控制 .

车模转向控制是根据摄像头拍摄的赛道图像，处理得到车模相对于赛道中间位置（39）的偏差。偏差是将车模图像的中间坐标减去给定值39得到的,因为车模摄像头拍摄的图像横轴是有80个坐标点,其坐标从0到79.中间值为39.5.由于在程序中不允许出现浮点数据，所以采用39作为中间坐标,得到的偏差经过pid算法计算最终输出到舵机，控制舵机转向。最终小车的位置将通过摄像头反馈回来，进行下一周期的运算。

车模的转向控制我们采用了PD控制算法。同样由于小车转向是个不断变化的过程，是个动态的模型，主要保证控制的跟随性，无需考虑静态误差。既不需要积分项来消除静态误差，所以同样采用PD控制算法，不采用积分项。图4.9是PID控制的框图：

偏差 PV

车模位置

前轮

转向机构向

舵机

PD

给定位置

-

摄像头

图4.9 方向PID控制架构图

所采用的PID控制器数学形式为：

（4-3）

为了补偿B车模转向的抖动，我们使用前第四次的偏差量和当前的偏差的差作为微分项，使得抖动现象有了很大好转。为了防止舵机打到死角堵转，需要对舵机角度进行限幅。赛道情况是非常复杂的，同一组参数并不一定能够完全适应不同的赛道形式，比如KP过大，车模会出现抖动的现象，过小的话拐弯力度不够。**所以我们引入了模糊PID算法的概念，就是根据赛道的弯曲程度给定不同的KP值，在直道上以小一点的KP系数运行，提高稳定性，弯道上以较大的KP运行，增大拐弯力度。具体的分段方式是根据偏差大小来分的，偏差变大说明小车正在进入弯道，变小说明进入直道。**以下是方向控制的代码：

void servo()//舵机控制函数

{

if(abs(my\_piancha)<5)//直道上，偏差绝对值小于5

{

kp1=18;//直道上的P，小一点，防止直道抖动

}

else

kp1=20;//不是直道P就要大一点。以增大拐弯力度，切赛道内侧。

jiao=(int)(kp1\*duojijiaodu-kd\*(lastpiancha\_4-duojijiaodu));

if(jiao>=1000)

jiao=1000;

if(jiao<=-1000)

jiao=-1000;

ftm\_pwm\_duty(FTM1,FTM\_CH0,4210+jiao); //20

}

## 4.4程序流程图

程序采用的是循环执行的架构，其流程图如图4.10所示：

开始

各模块初始化

图像采集

结束

检测到起跑线？

# 

图像处理

速度控制

# 

# 

方向控制

# 否

是

停车

图4.10 程序流程图

# 5 系统调试与运行

## 5.1 IAR Embedded Workbench IDE调试软件

软件开发环境使用的是IAR开发软件。IAR是一个集成的开发环境，可以进行在线调试，具有非常友好的调试界面。在此开发环境下可以进行软件模拟仿真，硬件在线仿真。支持对更改程序的版本模式，可以将工程更改为发布版本和调试版本两种模式，非常方便。在软件中可以通过livewatch窗口实时监测到程序中各个变量的值，以及各个寄存器的状态，这对于项目的开发起到了很大的作用，缩短了项目开发的周期。

## 5.2 串口调试工具

串口调试工具在车模调试过程中起着非常要的作用，传统的蓝牙模块，传输距离短，速率慢，因此我们采用的是基于433M频率的无线通信串口模块，此模块传输距离可达1000米，最高波特率可支持194000b/s。

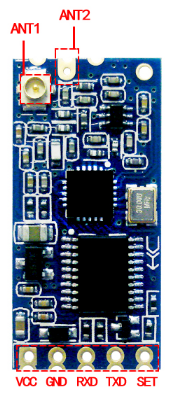


图5.1 无线调试模块

为了方便调试我们基于C++制作了上位机软件，在该上位机上可以查看摄像头图像，更改车模运行参数。

图5.2 上位机界面

## 5.3 系统运行与分析改进

经过上述方法的处理，小车在赛道上能够以较快的速度行驶，能够及时的切入弯道，并且以很好的路径在赛道上行驶，能够准确的判断出各个赛道元素，车速切换流畅，系统运行稳定可靠。最终在安徽省赛中，以31.187s的时间跑完了86米长的赛道，平均速度2.757m/s。位列安徽省第二名，成功晋级全国总决赛。但是也存在一些问题需要改进。

1．S弯道问题

在比赛过程中，会存在很长的一段S形弯道。如图3.5所示：



图5.3 小S弯道图

理论上S型弯道小车可以直接从中间直冲过去，但是在省赛中并没有进行处理，小车在S型弯道中是以抖动的形式缓慢行驶过去的，这样就整体拉低了小车运行的速度。如果能够在小S形弯道以直道的速度冲刺，那么将会在很大程度上提升车速。针对小S弯道存在的问题，省赛结束后进行了识别和处理，处理思路是一旦检测到赛道中有曲率正负交替出现的赛道，必然是S型弯道，此时小车以赛道的中间线为基准，全力加速，直接冲过小S弯道。

2.坡道检测策略的优化

在坡道检测方法上，省赛用的是摄像头识别的方法，但是在省赛中坡道样式有了变化，和程序里处理的样式不同，上坡道时小车全速冲了上去，导致小车飞了起来，但是最终巧合的落在了赛道上继续行驶。在全国总决赛中有多个坡道。为了降低风险，改变了坡道识别的策略，在硬件电路上增加了一个加速度传感器。当小车在平路上行驶时上下震动幅度是很小的，此时加速度传感器输出的值也很小，一旦冲上坡道，必然产生一个向上的加速度，此时小车就会减速上坡，当小车下坡时肯定也会震动一下，当在次检测到加速度值变大，说明小车已经下车了，此时开始恢复速度。经过测试，这种方法非常可靠，没有出现问题。

3.速度控制策略的优化

在省赛时速度控制策略不是很完美。比如在出弯道时，车辆只有在出了弯道以后才开始全力加速，等到速度提上来以后已经行驶了一大段距离，浪费了可以加速的路段。在后来准备全国总决赛过程中，针对省赛过程中速度控制策略上的一些不足进行了优化。针对出弯道加速不及时的问题，优化了赛道检测算法。检测到小车即将出弯道，就开始给电机很高的电压，电机开始全力加速，等到小车出了弯道以后，小车已经加到了很快的速度。在省赛中速度控制算法上也存在一些不足。省赛加速并没有使用bang-bang控制算法，速度的增加都是依靠PID算法一个个周期的累加完成的。为了使速度一步增加到位，在全国总决赛中，电机控制算法上结合了PID算法和bang-bang控制算法**。bang-bang控制算法的思路是一旦检测到小车给定的速度比当前实时速到大6以上，就让电机直接以满占空比全力加速，此时不经过PID算法提速，直到加速到和给定速度相同时，就切换到pid算法模式，通过pid算法调节速度**。经过这种处理，发现小车在直道上加速迅猛，能够很快的加速到给定速度，具有很大的爆发力。

经过上述改进，小车速度有了很大的提升，在全国总决赛中以2.97m/s的速度跑完全程。出弯加速非常及时，充分利用了每一段直道进行加速。在小S形弯道上，小车以直道的速度全速冲刺，拉高了整体的速度。最终，改进后的小车在全国近1000支参赛队伍中排名第13，获得了全国一等奖，是我省该项目所有参赛队伍中唯一一支获得该奖项的队伍，是我校参加该赛事以来获得过的最高奖项。

# 6 结论

小车制作过程中出现过很多值得思考的问题，在后续工作中值得引以为鉴。比如程序死机的问题，经过反复验证发现是程序编写时出现的漏洞。程序执行过程中数组超出范围，导致程序死机，经过改变数组长度，更改各个变量的作用域，有效地解决了死机的问题。**电机控制过程中存在着电机驱动板线路烧毁的现象，经过仔细推敲发现是因为车模刹车减速过程中反转力度过大，导致铜箔电流过大，烧毁铜箔。后期在设计第二块板子过程中将PCB板大电流的走线去除阻焊层，用焊锡加厚，有效地缓解了铜箔电流承受能力较弱的问题。**

整体来说整个车模的设计是比较严谨的，但也存在一些问题有待于进一步改进。车模摄像头所看到的视野决定了小车运行的提前量，如果摄像头看的范围越广，那么小车获取到的信息量也就越大，得到的信息也就更准确。在开始设计车模的时候，为了增大车模的视野，使用的是广角镜头，但是广角镜头存在着图像的畸变，需要进行逆透视变换，将畸变的图像还原。但是此过程是非常复杂的，为了减少难度，改用了80度的普通平角镜头。此镜头视野比较狭窄，只能看到110cm以内的路面，不能使小车提前做出反应。这方面在后期是很值得改进的。因此不管设计任何产品，是很难一次就做到完美的，但是对于发现的问题要及时的改正，不断地发现问题解决问题，就可以做到完美。

做车的过程是艰辛的，需要的是坚定的信念和浓厚的兴趣。考验的是队员的技术能力和专业功底以及心理素质。小车所用的技术并不是很高深的，重要的是过程，做车的过程就其实是一个产品开发的过程，全面锻炼了队员的专业知识应用能力，动手能力，学习能力。在以后的工作中更需要的是这种孜孜不倦的精神和敢于面对困难不放弃的勇气。

# 参考文献

[1] 杨恢先等.单片机原理及应用[J].国防科技大学出版社，2003.

[2] 胡健等.单片机原理及接口技术实践教程[J].机械工业出版社，2004.

[3]康华光.电子技术基础.高等教育出版社[J]，2006.

[4]阎石.数字电子技术基础[J].高等教育出版社，2006.

[5]李小坚.郝晓丽.Protel DXP电路设计与制版使用教材[M],人民邮电出版社，2015.

[6] 吴亦峰、陈德为.单片机原理与接口技术.北京:电子工业出版社，2005.

[7] 朱春妹. 无线话筒工作原理及其用途[J].才智,2010,10:45.[10].谢自美.《电子线路设计实验》[M]. 华中科技大学出版社，2011.

[8]邹华跃. 《数字集成电路基础学习参考》[M]. 南京大学出版社，2011.

[9]童诗白，华成英. 《模拟电子技术基础》[M]. 高等教育出版社，2014.

[10]宋拯.《移动通信技术》[M]. 北京理工大学出版社，2012.

[11]Andrea Goldsmith, 《Wireless Communications》, Cambridge University Press, 2012.

[12]David Tse, Pramod Viswanath，《Fundamentals of Wireless Communication》,

Cambridge University Press, 2014.

[13]Lars Ahlin, Jens Zander,《Principles of Wireless Communications》,fourth

edition, John Wiley & Sons, 2013.

# 致 谢

大学四年一晃而过，四年里参加很多的学术科技竞赛，依托于学校的实验室，参与了很多的创新训练项目，学到了很多的专业知识，为自己的发展打下了良好的基础。感谢学校四年来的培养，提供了良好的教学环境，感谢每位老师的辛勤付出。在此我也要感谢我的队友，我们在一起相互帮助，相互学习，共同解决问题，让我们知道了团队工作的乐趣。丰富了我们的课外知识，提高了我们的动手实践的能力。

在这即将毕业的时刻，我向这几年所有教我和指导我的老师表示深深的感谢，他们在生活上和学习上给我很大帮助。他们陪伴着我们度过了短暂的大学四年学习生活，他们改变了我们，让我们从他们身上学到了很多课本上不曾学到的知识，他们也让我们感觉到了他们那种无私奉献的精神。所以在以后的工作生活中，我们要学习这种精神，为社会做出贡献。

# 附 录

//作者：程祯 2016年智能车摄像头组国赛一等奖程序

//说明：saomiao文件下有一个lkcongzhongjiansaomiao（）函数是图像处理函数，偏差处理以及舵机控制均在此文件下

//duoji\_dianji文件下都是速度控制的相关程序，核心程序就是这两个文件

#include "common.h"

#include "include.h"

#include "math.h"

#include "saomiao.h"

uint8 imgbuff[CAMERA\_SIZE];

int currentzhongjian[60];

int right\_heixian[60];

int left\_heixian[60];

int quanjuCount;

int OutData[4],tuoluoyivar;

int star\_lineflag=0;

int LK\_jishi=0;

int LK\_jishi\_flag=0;

int LK\_yanshi=0,zhidao\_flag,yunsu\_flag,yuzhi;

extern int podao\_flag,zhangai\_right,zhangai\_left,qvlv\_quanju,duandian,mubiao\_speed,dawan\_speed,qulv\_jinduan;

extern void lukuangudge();

char go=0,S1,S,zhichong\_flag;

extern char lcd\_show\_enable1;

//s16 speed\_a;

extern int my\_piancha;

int zhangaijishiright\_flag=0,zhangaijishileft\_flag=0;

//int kp=22,kd=10,right\_heixian[60],lastpiancha\_1,duojijiaodu,left\_heixian[60],my\_lastzhongjian=40,currentzhongjianzhi=40,rightheixian\_flag,leftheixian\_flag,xielv;

//函数声明

void PORTA\_IRQHandler();

void DMA0\_IRQHandler();

void PIT0\_IRQHandler();

void Priority\_Set();

extern char s\_wan\_flag;

char pof;

//void PIT1\_IRQHandler();

/\*!

\* @brief main函数

\* @since v5.3

\* @note 山外 DMA 采集摄像头 实验

注意，此例程 bus频率设为100MHz(50MHz bus频率会太慢而导致没法及时采集图像)

此例程使用的上位机为【山外多功能调试助手】

具体资料请参考：山外多功能调试助手资料专辑

http://www.vcan123.com/forum.php?mod=collection&action=view&ctid=27

\*/

void hecheng()//虚拟中线函数

{

unsigned int ji;

for(ji=0;ji<=59;ji++)

{

imgyiwei[ji][currentzhongjian[ji]]=0 ;

}

}

void sendimg(uint8 \*imgaddr,uint32 imgsize)

{

img\_extract((u8 \*)imgyiwei, imgbuff,CAMERA\_SIZE); //解压图像

lkcongzhongjiansaomiao();

uint8 cmd[4] = {0, 255, 1, 0 }; //yy\_摄像头串口调试 使用的命令

hecheng();

uart\_putbuff(VCAN\_PORT, cmd, sizeof(cmd)); //先发送命令

uart\_putbuff(VCAN\_PORT, imgaddr, imgsize); //再发送图像

}

void star\_line\_judg()//起跑线检测

{

int kk,bai\_flag=0,hei\_flag=0,heibai\_flag=0,baihei\_flag=0;

for(kk=5;kk<=72;kk++)

{

if(imgyiwei[45][kk]>0)

bai\_flag=1;

else

if(bai\_flag&&imgyiwei[45][kk]==0)

{

baihei\_flag++;

bai\_flag=0;

}

if(imgyiwei[45][kk]==0)

hei\_flag=1;

else

if(hei\_flag&&imgyiwei[45][kk]>0)

{

heibai\_flag++;

hei\_flag=0;

}

}

if(baihei\_flag>=4&&heibai\_flag>=4&&baihei\_flag-heibai\_flag<=2)

star\_lineflag=1;

}

void lkzhongjian()

{

unsigned int i;

for(i=0;i<=59;i++)

{

imgyiwei[i][currentzhongjian[i]]=0;

}

}

/\*void sendimg(uint8 \*imgaddr,uint32 imgsize)

{

uint8 cmd[4] = {0, 255, 1, 0 }; //yy\_摄像头串口调试 使用的命令

//hecheng()；

uart\_putbuff(VCAN\_PORT, cmd, sizeof(cmd)); //先发送命令

uart\_putbuff(VCAN\_PORT, imgaddr, imgsize); //再发送图像

// uart\_putbuff(VCAN\_PORT, currentzhongjian, 60); //再发送图像

}\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* 功能说明：SCI示波器CRC校验

内部调用函数

\* 参数说明： 无

\* 函数返回：无符号结果值

\* 修改时间：2013-2-10

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

static unsigned short CRC\_CHECK(unsigned char \*Buf, unsigned char CRC\_CNT)

{

unsigned short CRC\_Temp;

unsigned char i,j;

CRC\_Temp = 0xffff;

for (i=0;i<CRC\_CNT; i++)

{

CRC\_Temp ^= Buf[i];

for (j=0;j<8;j++) {

if (CRC\_Temp & 0x01)

CRC\_Temp = (CRC\_Temp >>1 ) ^ 0xa001;

else

CRC\_Temp = CRC\_Temp >> 1;

}

}

return(CRC\_Temp);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

/\*

\* 功能说明：SCI示波器发送函数

\* 参数说明：

OutData[] 需要发送的数值赋予该数组

\* 函数返回：无符号结果值

\* 修改时间：2013-2-10

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void OutPut\_Data(void)

{

int temp[4] = {0};

unsigned int temp1[4] = {0};

unsigned char databuf[10] = {0};

unsigned char i;

unsigned short CRC16 = 0;

for(i=0;i<4;i++)

{

temp[i] = (int)OutData[i];

temp1[i] = (unsigned int)temp[i];

}

for(i=0;i<4;i++)

{

databuf[i\*2] = (unsigned char)(temp1[i]%256);

databuf[i\* 2+1] = (unsigned char)(temp1[i]/256);

}

CRC16 = CRC\_CHECK(databuf,8);

databuf[8] = CRC16%256;

databuf[9] = CRC16/256;

for(i=0;i<10;i++)

{

uart\_putchar (UART0,(char)databuf[i]);

}

}

void SendHex(unsigned char hex) {

unsigned char temp;

temp = hex >> 4;

if(temp < 10) {

uart\_putchar(UART0,temp + '0');

} else {

uart\_putchar(UART0,temp - 10 + 'A');

}

temp = hex & 0x0F;

if(temp < 10) {

uart\_putchar(UART0,temp + '0');

} else {

uart\_putchar(UART0,temp - 10 + 'A');

}

}

void SendImageData(unsigned char ImageData[][80])

{

int lll1,lll2;

unsigned char crc = 0;

lkcongzhongjiansaomiao();

lkzhongjian();

/\* Send Data \*/

uart\_putchar(UART0,'\*');

uart\_putchar(UART0,'L');

uart\_putchar(UART0,'D');

SendHex(0);

SendHex(0);

SendHex(0);

SendHex(0);

// imgyiwei[60][80];

for(lll2=0;lll2<80;lll2++)

{

for(lll1=0;lll1<60; lll1++)

SendHex(ImageData[lll1][lll2]);

}

/\* for(ll1=0;ll1<60;ll1++)

{

for(ll0=0;ll0<80;ll0++)

{

if(imgyiwei[ll1][ll0]==0)

Draw\_potL(ll0,ll1,0,1);

else

Draw\_potL(ll0,ll1,1,1);

}

}\*/

SendHex(crc);

uart\_putchar(UART0,'#');

}

#if 1

//编码器初始化

void FTM\_QUAD\_init()

{

/\*开启端口时钟\*/

SIM\_SCGC5 |= SIM\_SCGC5\_PORTA\_MASK;

/\*选择管脚复用功能\*/

// PORTA\_PCR12 = PORT\_PCR\_MUX(7);

// PORTA\_PCR13 = PORT\_PCR\_MUX(7);

PORTA\_PCR10 = PORT\_PCR\_MUX(6);

PORTA\_PCR11 = PORT\_PCR\_MUX(6);

/\*使能FTM1、FTM2时钟\*/

// SIM\_SCGC6|=SIM\_SCGC6\_FTM1\_MASK;

SIM\_SCGC3|=SIM\_SCGC3\_FTM2\_MASK;

// FTM1\_MOD = 65535; //可根据需要设置

FTM2\_MOD = 65535;

// FTM1\_CNTIN = 0;

FTM2\_CNTIN = 0;

// FTM1\_MODE |= FTM\_MODE\_WPDIS\_MASK; //禁止写保护

FTM2\_MODE |= FTM\_MODE\_WPDIS\_MASK; //禁止写保护

// FTM1\_MODE |= FTM\_MODE\_FTMEN\_MASK; //FTMEN=1,关闭TPM兼容模式，开启FTM所有功能

FTM2\_MODE |= FTM\_MODE\_FTMEN\_MASK; //FTMEN=1,关闭TPM兼容模式，开启FTM所有功能

// FTM1\_QDCTRL &= ~FTM\_QDCTRL\_QUADMODE\_MASK; //选定编码模式为A相与B相编码模式

// FTM1\_QDCTRL |= FTM\_QDCTRL\_QUADEN\_MASK; //使能正交解码模式

// FTM2\_QDCTRL &= ~FTM\_QDCTRL\_QUADMODE\_MASK; //选定编码模式为A相与B相编码模式 0x8u 0x00001000取反即 0x11110111

FTM2\_QDCTRL |= 0x08;//~0x00u;

FTM2\_QDCTRL |= FTM\_QDCTRL\_QUADEN\_MASK; //使能正交解码模式

//QUADMODE=1;

// FTM1\_SC |= FTM\_SC\_CLKS(3); //选择外部时钟

// FTM1\_CONF |=FTM\_CONF\_BDMMODE(3); //可根据需要选择

FTM2\_SC |= FTM\_SC\_CLKS(3);

// FTM2\_CONF |=FTM\_CONF\_BDMMODE(3);

}

#endif

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void main(void)

{

DisableInterrupts;

// SXT\_chunshuang\_flag=1;

// SXT\_chunshuang\_flag\_count=20;

// danxian\_chushizhongdian=20;

//zhongjian\_52hang=40;

//DSJ\_speed\_canshu();//没有

dianji\_canshu\_init();

// duoji\_dianji\_Init();

Priority\_Set();

camera\_init(imgbuff); //这里设定 imgbuff 为采集缓冲区！！！！！！

ftm\_pwm\_init(FTM0,FTM\_CH2,10000,0); //右电机初始化

ftm\_pwm\_init(FTM0,FTM\_CH5,10000,0);

ftm\_pwm\_init(FTM0,FTM\_CH4,10000,0); //左电机初始化

ftm\_pwm\_init(FTM0,FTM\_CH3,10000,0);

ftm\_pwm\_init(FTM1,FTM\_CH0,50,750);//舵机初始化

ftm\_quad\_init(FTM2);

lptmr\_pulse\_init(LPT0\_ALT2, 0xFFFF, LPT\_Rising);

pit\_init\_ms(PIT0,5);

my\_lcd\_init();//液晶初始化

set\_vector\_handler(PIT0\_VECTORn,PIT0\_IRQHandler); // 设置中断服务函数到中断向量表里

enable\_irq(PIT0\_IRQn); // 使能PIT中断

set\_vector\_handler(PORTA\_VECTORn,PORTA\_IRQHandler); //设置PORTA的中断服务函数为 PORTA\_IRQHandler

set\_vector\_handler(DMA0\_VECTORn,DMA0\_IRQHandler); //设置DMA0的中断服务函数为 DMA0\_IRQHandler

EnableInterrupts;

// adc\_init (ADC1\_SE4a);

gpio\_init(PTC18,GPO,1);//off

while(1)

{

camera\_get\_img();

// my\_lcd\_show();

if(LK\_jishi\_flag==1)

star\_line\_judg();

//sendimg((u8 \*)imgyiwei, CAMERA\_W \* CAMERA\_H);//我的上位机，不注释为向电脑发送图像

}

}

/\*!

\* @brief PORTA中断服务函数

\* @since v5.0

\*/

void PORTA\_IRQHandler()

{

uint8 n = 0; //引脚号

uint32 flag = PORTA\_ISFR;

PORTA\_ISFR = ~0; //清中断标志位

n = 29; //场中断

if(flag & (1 << n)) //PTA29触发中断

{

camera\_vsync();

}

#if 0 //鹰眼直接全速采集，不需要行中断

n = 28;

if(flag & (1 << n)) //PTA28触发中断

{

camera\_href();

}

#endif

}

/\*!

\* @brief DMA0中断服务函数

\* @since v5.0

\*/

//void PIT1\_IRQHandler()

//{

//shizi\_count++;

//}

void DMA0\_IRQHandler()

{

camera\_dma();

img\_extract((u8 \*)imgyiwei, imgbuff,CAMERA\_SIZE);

}

void PIT0\_IRQHandler(void)//定时器中断服务函数

{

lcd\_show\_enable1=0;//注释掉是调用液晶

PIT\_Flag\_Clear(PIT0);

if(lcd\_show\_enable1)//为0跳出液晶

{

my\_lcd\_show();

}

else

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*脉冲提取\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

get\_maichong();//获取电机转速

if(LK\_jishi\_flag==0)

LK\_jishi++;

if(LK\_jishi==300)//起步延时1.5秒

go=1;//小车前进

/\*if(quanjuCount<2000)

{

quanjuCount++;

quanjuCount=0;

}\*/

if(LK\_jishi>=2000)

{

LK\_jishi\_flag=1;

LK\_jishi=2000;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*边线提取\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// if(quanjuCount>20)

// {

lkcongzhongjiansaomiao();//图像处理

if(star\_lineflag==1)

LK\_yanshi++;

if(!star\_lineflag&&go)

{

if(yunsu\_flag==1)

mubiao\_speed=dawan\_speed;

else

lukuangudge();//路况判断

}

else

if(LK\_yanshi>30)//检测到起跑线

mubiao\_speed=0;

//if(pof)

//mubiao\_speed=0;

DSYJ\_dianji\_PID(mubiao\_speed); //控制电机转速

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*S弯判定\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(s\_wan\_flag)

{

S1=1;

//s\_wan\_flag=0;

}

else

{

S=0;

S1=0;

}

if(S1&&S<=25)

S++;

if(S>=25)

{

zhichong\_flag=1;

//gpio\_set(PTE1,1);

}

else

{

zhichong\_flag=0;

//gpio\_set(PTE1,0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*坡道处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// if(abs(my\_piancha)<=16&&qvlv\_quanju<=10&&duandian<12&&qulv\_jinduan<12)

// tuoluoyivar=adc\_once(ADC1\_SE4a,ADC\_8bit);

// else

// tuoluoyivar=104;

/\* if(abs(tuoluoyivar-104)>40&&!podao\_flag)

{

podao\_flag=1;

// gpio\_set(PTE1,1);

}

if(podao\_flag&&podao\_flag<800)

{

podao\_flag++;

}

else

{

podao\_flag=0;

//gpio\_set(PTE1,0);

}

if(podao\_flag>0&&podao\_flag<130)

pof=1;

else

pof=0;\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*障碍处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(zhangai\_right==1)

{

zhangaijishiright\_flag=1;

zhangai\_right=0;

}

else if(zhangai\_left==1)

{

zhangaijishileft\_flag=1;

zhangai\_left=0;

}

if(zhangaijishileft\_flag&&zhangaijishileft\_flag<35)

{

zhangaijishileft\_flag++;

// led(LED0, LED\_ON);

}

else

if(zhangaijishiright\_flag&&zhangaijishiright\_flag<35)

{

zhangaijishiright\_flag++;

}

else

zhangaijishiright\_flag=zhangaijishileft\_flag=0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*直道判定\*直道判定\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(abs(my\_piancha)<=13&&qvlv\_quanju<=13&&duandian<10)

{

zhidao\_flag++;

}

else

{

zhidao\_flag=0;

}

//gpio\_turn(PTA17);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*偏差处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

pianchachuli();

}

}

附录