

第十一届“恩智浦”杯全国大学生  
智能汽车竞赛  
技 术 报 告



学    校： 燕山大学

队伍名称： yanfei 队

参赛队员： 王雪翔

韩鹏飞

周永健

带队教师： 宋俊杰  呼忠权



## 关于技术报告和研究论文使用授权的说明

本人完全了解第十一届“恩智浦”杯全国大学生智能汽车竞赛关保留、使用技术报告和研究论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会和恩智浦半导体公司可以在相关主页上收录并公开参赛作品的设计方案、技术报告以及参赛模型车的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名：\_\_\_\_\_

带队教师签名：\_\_\_\_\_

日 期：\_\_\_\_\_

# 目 录

摘 要 .....	1
引 言 .....	2
第一章 系统总体设计 .....	4
1.1 车模的整体造型 .....	5
1.2 车模参数 .....	6
第二章 机械结构的安装及调整 .....	7
2.1 车模舵机位置的安装 .....	7
2.2 智能车前轮定位 .....	8
2.3 智能车电机定位 .....	10
2.4 智能车电池定位 .....	11
2.5 智能车底盘固定与重心调节 .....	11
2.6 智能车差速调节 .....	12
2.7 轮胎的保养与使用 .....	12
2.8 摄像头的安装 .....	12
2.9 编码器的安装 .....	13
2.10 其他机械结构 .....	13
2.11 小结 .....	13
第三章 电路设计说明 .....	15
3.1 电源管理模块 .....	15
3.2 主控板的设计 .....	17
3.3 电机驱动模块 .....	18
3.4 起跑线检测红外对管电路设计 .....	19
第四章 软件算法设计 .....	20
4.1 原始图像的特点 .....	20
4.2 赛道中心线提取及处理 .....	21
4.3 赛道边沿提取 .....	21
4.4 边线处理 .....	21
4.5 中断的调用 .....	23
4.6 电机控制策略 .....	23
4.6.1 PID 控制算法介绍 .....	23
4.6.2 位置式 PID .....	25
4.6.3 增量式 PID .....	25
4.6.4 舵机控制 .....	26
4.6.5 电机控制 .....	26
第五章 系统开发及调试工具 .....	27
5.1 开发工具 .....	27
5.2 上位机图像显示 .....	28
5.3 蓝牙调试模块与串口通信 .....	29
5.4 信号发生器以及示波器的使用 .....	30
第六章 总结 .....	32
致 谢 .....	33
参考文献 .....	34

附录 A：程序源代码 ..... 35



## 摘 要

本文以第十一届“恩智浦”杯全国大学生智能汽车竞赛为背景，详细的介绍了智能车的设计方案和原理。车模采用的是组委会统一规定购买的 B 车模，以 32 位微控制器 K60 为核心控制单元搭建硬件平台，在 IAR 开发环境中进行软件开发。通过 CMOS 摄像头（OV7620）采集图像，检测赛道信息，辨认障碍和其他赛道元素，并提取黑色引导线；舵机采用 PD 控制转向；通过光电编码器对车模运行过程中进行测速，构成速度反馈，用增量式 PID 来对模型车进行运行速度的闭环控制。

为了提高模型车的稳定性和速度的快速调节，我们建立了实际系统的基本模型，通过理论分析，在线仿真，并不断查阅相关资料后确定最终方案，最终完成智能车的制作。

关键字：MK60、OV7620、PID、智能车

## 引 言

为加强大学生实践、创新能力和团队精神的培养，促进高等教育教学改革。受教育部高等教育司委托，由教育部高等学校自动化专业教学指导委员会主办全国大学生智能汽车竞赛。该竞赛是以智能汽车为研究对象的创意性科技竞赛，是面向全国大学生的一种具有探索性工程实践活动，是教育部倡导的大学生科技竞赛之一。该竞赛以“立足培养，重在参与，鼓励探索，追求卓越”为指导思想，旨在促进高等学校素质教育，培养大学生的综合知识运用能力、基本工程实践能力和创新意识，激发大学生从事科学研究与探索的兴趣和潜能，倡导理论联系实际、求真务实的学风和团队协作的人文精神，为优秀人才的脱颖而出创造条件。

第十一届“恩智浦”杯全国大学生智能汽车竞赛由竞赛秘书处设计、规范标准硬软件技术平台，竞赛过程包括理论设计、实际制作、整车调试、现场比赛等环节，要求学生组成团队，协同工作，初步体会一个工程性的研究开发项目从设计到实现的全过程。该竞赛融科学性、趣味性和观赏性为一体，是以迅猛发展、前景广阔的汽车电子为背景，涵盖自动控制、模式识别、传感技术、电子、电气、计算机、机械与汽车等多学科专业的创意性比赛。该竞赛规则透明，评价标准客观，坚持公开、公平、公正的原则，力求向健康、普及、持续的方向发展。

本技术报告主要包括机械系统、硬件系统、软件系统等，详尽地阐述了我们的设计方案，具体表现在硬件电路的创新设计以及控制算法的独特想法，并重点描述我们在制作智能车的过程中遇到的困难和解决办法。这一年包含着我们的辛勤努力，这份报告凝聚了我们智慧，是我们团队共同努力的成果。

在智能车制作过程中，我们参考了大量的相关资料文献。从官方网站下载K60系列单片机开发技术手册，了解所有寄存器功能。同时也参照学习了历年的技术报告的算法思想，遇到困难在论坛上和知网上寻求帮助，结合去年赛道识别和转向的控制方式，通过边缘提取算法获得黑线位置，利用计算偏差的思想设计出适合本车的控制算法。

任何科研项目的启动都需要从查找文献开始，站在前人的肩膀上能让自己



少走很多弯路，也才能使相同方向的研究更快的取得进步。

# 第一章 系统总体设计

根据大赛的统一要求，须在组委会统一提供的车模平台上，根据规则要求选择传感器类型，设计系统硬件电路，开发软件算法。本系统主要包括三个大部分，分别为车模的机械结构、硬件电路系统、软件算法。每一个部分又由各个小模块构成。所以要构建一个完整的小车系统，必须先对各个模块进行论证和设计，在将整个系统组合成一个完整系统，进行系统整体的调试。

根据智能车系统的基本要求，我们设计了系统结构图，如图 1.1 所示。在满足比赛要求的情况下，力求系统简单高效，因而在设计过程中尽量简化硬件结构，减少因硬件而出现的问题。

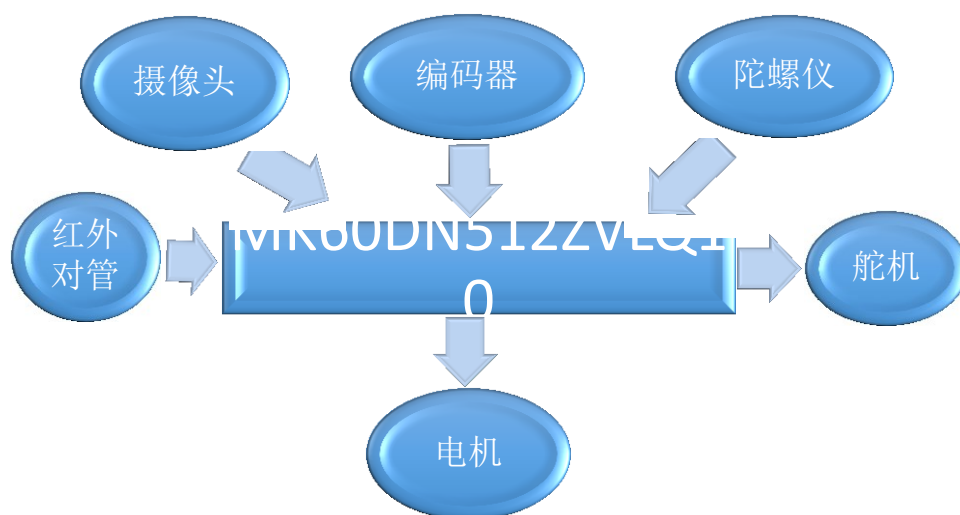


图 1.1 智能车整体结构

我们通过 CMOS 摄像头采集赛道上的图像，由 MK60DN512ZVLQ10 处理，计算出黑线位置，对偏差进行处理；用 PD 算法控制舵机转向；使用三轴数字陀螺仪用来检测坡道和转向的优化；通过光电编码器来检测速度，采用

MK60DN512ZVLQ10 的 DMA 采集功能进行脉冲捕捉获得速度和路程，而车速的目标值由默认值、运行安全方案和基于图像处理的优化策略进行综合控制；用 PID 控制电机，通过 PWM 控制驱动电路调整电机的功率；红外对管用来检测起跑线。调试过程通过辅助蓝牙模块 HC-05、键盘模块、拨码开关、LED 指示及编译器等完成。为使模型车在比赛中发挥出最佳性能，使其直线行驶稳定，入弯转向灵活，必须结合汽车理论相关知识对智能小车的运动特性进行分析，并对此智能小车的底盘结构进行相应的调整和参数优化。

### 1.1 车模的整体造型

车模的整体设计要简洁、轻便、可靠，在保证车能够稳定行驶的前提下，尽可能的降低车的重心，如图 1.2 是我们的小车的整体布局。

- (1)车模底盘采用硬连接，去除悬挂，使底盘降低，主板降低，以降低重心；
- (2)舵机竖直放置，方便控制；
- (3)用轻便坚固的碳纤杆作为摄像头杆的材料；
- (4)尽可能去除模型车上多余的线头，提高电路的稳定性能。



图 1.2 整车效果图

## 1.2 车模参数

项目	参数
路径检测方法（赛题组）	摄像头
车模几何尺寸长*宽*高(单位:毫米)	27.8*18.2*32
车模轴距/轮距（单位:毫米）	200/140
车模平均电流	3000mA
电路电容总量 单位:微发	1200
传感器种类及个数	一个 CMOS 摄像头、一个陀螺仪、两个光电管、一个光电编码器
新增加伺服电机个数	无
赛道信息检测空间精度（毫米）	10
赛道信息检测频率	50Hz
主要集成电路种类/数量	MK60DN256VMD10 最小系统、电机驱动桥
车模重量（带有电池）单位:千克	1.2

本章主要介绍了智能小车控制系统的工作原理和赛车整体结构设计框图。硬件、软件和机械部分的有效融合是赛车能否跑出好成绩的关键因素。根据系统的需求简要给出了系统的总体设计方案，并具体将整个系统划分成传感系统、控制决策系统和动力系统，并对系统的总体框架作了一个简单的介绍。只有在把握整体之后，才能进行细节的处理。

## 第二章 机械结构的安装及调整

通过不断地实验和测试，我们发现以下三个方面的机械结构对小车的运行影响很大：转向机构，传感器的固定，重心的调整。高速运行下舵机的转动速度对车转向的灵活程度起到了根本性的作用。摄像头是采集最原始赛道信息的最前端的传感器，良好的固定方案才能使其最大限度地采集到良好的赛道信息。而整车的重心越低，自然才能在控制时更加灵活稳定。

### 2.1 车模舵机位置的安装

舵机直接控制着小车的转向，因此舵机的安放位置和打孔精度至关重要，如果底盘打孔左右不对称就可能导致车模左转右转性能的差异，从而引起一些不必要的问题（侧滑或推轮）。

对于小车的转向，它是通过舵机带动左右横拉杆来实现的。由于舵机的转动速度和功率是一定，要想加快转向机构响应的速度，唯一的办法就是优化舵机的安装位置和其力矩延长杆的长度。由于功率是速度与力矩乘积的函数，过分追求速度，必然要损失力矩，力矩太小也会造成转向迟钝，因此设计时就要综合考虑转向机构响应速度与舵机力矩之间的关系，通过优化得到一个最佳的转向效果。

我们发现为了使转向轻便，要使舵机拉杆呈水平状态，可以让舵机打角迅速。



图 2.1 舵机摆臂

我们采用的舵机摆臂为图 2.1 所示，去除了车模原配的同轴连片，其安装方式为中置立式。

### 2.2 智能车前轮定位

现代汽车在正常行驶过程中，为了使汽车直线行驶稳定，转向轻便，转向后能自动回正，减少轮胎和转向系零件的磨损等，在转向轮、转向节和前轴之间须形成一定的相对安装位置，叫车轮定位，其主要的参数有：主销后倾、主销内倾、车轮外倾和前束。智能车竞赛模型车的四项参数都可以调整，但是由于模型车加工和制造精度的问题，在通用的规律中还存在一些偶然性。

主销后倾(Caster)是指主销装在前轴，上端略向后倾斜的角。它使车辆转弯时产生的离心力所形成的力矩方向与车轮偏转方向相反，迫使车轮偏转后自动恢复到原来的中间位置上。由此，主销后倾角越大，车速越高，转向轮稳定性也愈好，但过大的主销后倾角会导致转向沉重。

但是 B 车模是无法通过加减垫片的方式来调整主销后倾角的，如果要调整的话，只能通过反装图 2.2 的配件来完成。



图 2.2 主销后倾配件

主销内倾(Kingpin Inclination)是指主销装在前轴略向内倾斜的角度，它的作用是使前轮自动回正。角度越大前轮自动回正的作用就越强烈，但转向时也越费力，轮胎磨损增大；反之，角度越小前轮自动回正的作用就越弱。另外，除了这个作用之外，主销内倾还能够因为主销轴线延长线跟路面相交的点与车轮

接触地面的那个点的距离缩短，从而使转向力臂增长，让转向更轻便。

转向轮外倾角(Camber)对赛车的转弯性能有直接影响,它的作用是提高转向轮的转向安全性和转向操纵的轻便性。转向轮外倾角俗称“外八字”。如果车轮垂直地面一旦满载就易产生变形,可能引起车轮上部向内倾侧，导致车轮联接件损坏。所以事先将车轮校偏一个外八字角度，这个角度约在  $1^{\circ}$  左右。但是，对于注重车子转弯性能的赛车来说，为了提高车辆在转向时的极限能力，一般把前轮调成内倾。

对于模型车，通过调整前桥的螺杆的长度可以改变主销内倾角的大小，由于过大的内倾角也会增大转向阻力，增加轮胎磨损，所以在调整时可以根据实际情况进行角度的调整，近似为  $0^{\circ} \sim 3^{\circ}$  左右，不宜太大。出现侧滑现象可以适当增大。

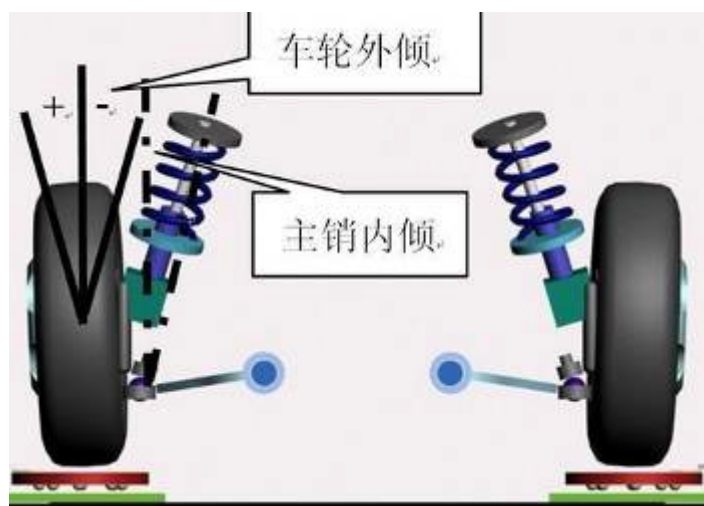


图 2.3 主销内倾角

主销内倾和主销后倾都有使汽车转向自动回正，保持直线行驶的功能。不同之处是主销内倾的回正与车速无关，主销后倾的回正与车速有关，因此高速时主销后倾的回正作用大，低速时主销内倾的回正作用大。

所谓前束(Toe)是指两轮之间的后距离数值与前距离数值之差，也指前轮中心线与纵向中心线的夹角。前轮前束的作用是保证汽车的行驶性能，减少轮胎的磨损。前轮在滚动时，其惯性力会自然将轮胎向内偏斜，如果前束适当，轮胎滚动时的偏斜方向就会抵消，轮胎内外侧磨损的现象会减少。一般来说，前轮会被调教成正前束(Toe-In)，因为正前束可以让车轮保持直线行驶。



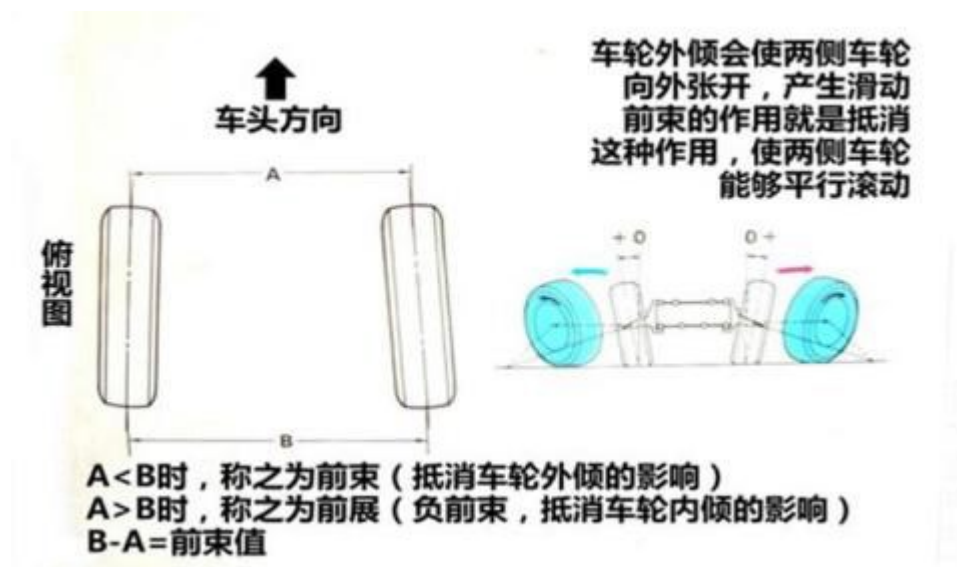


图 2.4 前轮前束示意图

在模型车中，前轮前束的调整是方便的，通过调整伺服电机带动的左右横拉杆实现的。主销在垂直方向的位置确定后，改变左右横拉杆的长度即可以改变前轮前束的大小。在实际的调整过程中，我们发现前束在摩擦力大的时候有明显的效果。但是一定不要太大，否则转向沉重，速度降低，适当的放开一两圈就够了。

另外，我们的车模的控制要求整个系统十分精确，所以车模本身存在的间隙就会对车在高速运行的情况下产生很大影响，尤其是转向桥部分。所以对于轴承与轴的间隙，主销与其配合的零件之间的间隙等，都应该尽量去减少，我们对此进行加薄垫片或垫纸等处理。

## 2.3 智能车电机定位

电机的定位主要影响前轮的齿轮动力传动。必须调整电机传动轴与轮轴平行，且要调整好合适的齿轮间距，否则会导致齿轮磨损严重，并发出刺耳的噪声。我们安装的时候发现齿轮间啮合要保有微小的裕度，并且不脱落，传动效果是比较不错的，要经常清除灰尘并且加上适当的润滑油也能让齿轮磨合更好。

整体原则：两传动齿轮轴保持平行，齿轮间的配合间隙要合适，过松容易打坏齿轮，过紧又会增加传动阻力，浪费动力；传动部分要轻松、顺畅，不能有迟滞或周期性振动的现象。判断齿轮传动是否良好的依据是，听一下电机带动后轮空转时的声音。声音刺耳响亮，说明齿轮间的配合间隙过大，传动中有撞



齿现象；声音闷而且有迟滞，则说明齿轮间的配合间隙过小，或者两齿轮轴不平行，电机负载变大。调整好的齿轮传动噪音很小，并且不会有碰撞类的杂音，后轮减速齿轮机构就基本上调整好了，动力传递十分流畅。

## 2.4 智能车电池定位

为了降低重心，我们将电池直接与底盘接触，并靠后放置，使用魔术扎带进行固定；关于电池的安装位置，其实是与前轮的调整有很大关系的，但必须要把智能车的质心质量分配到前/后轴，因而放在前面和后面都有各自的好处，放在前面可以平衡车重，出现的推轮现象是带有转向延迟的，而甩尾是能够高速响应的，需要配合前轮的角度进行调整则可以解决；放在后面则甩尾是带有延迟，反方向打角无法及时调整，造成转向过度，也可以通过调整前轮的角度来解决。所以前面提到的主销角以及前束角的调整也存在不少的偶然性，要以实际效果为准。

## 2.5 智能车底盘固定与重心调节

由于小车行驶道路比较平整，车模位置有限，我们没有使用悬挂装置。但是我们也没有采用彻底的硬性连接，而底盘连接要有一定的刚性裕度，过硬的连接会导致车模转向时前轮上下震动，转向困难，造成推轮现象。前轮是转向轮，为了让舵机转向更加灵活，在尽量降低重心的同时，使前轮部分地盘略高于后轮地盘，即让重心稍微偏后，达到既能使重心降低，又能使舵机转向灵活的目的。

底盘高度：对于后轮，通过调节后轮车轮的小部件即可调整后轮的高度；对于前轮，在前桥弹簧所处的位置适当增加垫片，或者是前轮与地盘所处位置增加垫片，均可有效的降低车的地盘高度。另外调节电池的位置，电路板的安放位置可使车的重心有效的向后方偏移。有时候可以认为加些配重来平衡，图 2.5 即我们的安放位置。

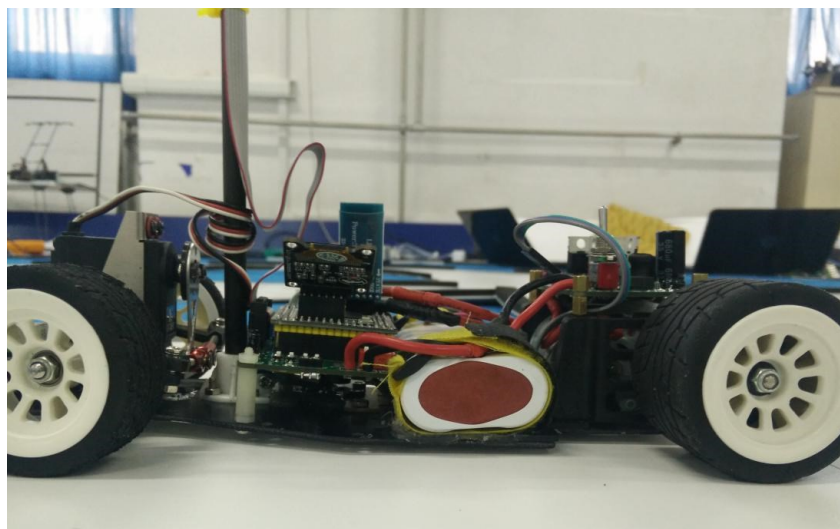


图 2.5 电路板 and 电池安装位置

### 2.6 智能车差速调节

差速是决定车模转向的灵活性和直线加速的迅猛性的重要环节。我们发现过紧的差速使转弯变的迟钝，在高速情况下容易侧翻。当然，差速过松的情况下转弯甩尾，变速困难。安装后测试效果，反复重复后，寻求最佳效果。最终我们的差速调整为最紧状态松一圈螺母而已，加减速效果明显，又能起到差速器功能。

### 2.7 轮胎的保养与使用

本届比赛 B 车模采用材料为橡胶的刻有花纹轮胎，在使用的过程中主要暴露出的问题有：轮胎材料强度不足，长期使用容易发粘、磨损严重。

规则中不允许对轮胎进行任何改变轮胎性能的处理，因此在调试过程中，我们尽量选择对称、圆的轮胎；尽可能改变前轮倾角，使轮胎与地面的接触面积在各个工况下综合最大；在不调试的时间里尽量避免四轮与地面接触，悬空放置，保证轮胎的形状不发生改变。

### 2.8 摄像头的安装

摄像头是整辆车的眼睛，它的安装是最重要的。经过多次安装实验，我们最后决定使用单杆结构来固定摄像头，并且使用配套的支架完成安装。对于安装的高度要注意，过高的话会使其抖动，另外过低则可能会使图像易受到赛

道反光影响，视野畸变严重。我们将摄像头安放在车模的前方位置。

## 2.9 编码器的安装

在安装编码器的时候要保证有合适的齿轮啮合。啮合过紧，或者两齿轮轴不平行，电机负载加大。调整好的齿轮传动噪音小，并且不会有碰撞类的杂音。编码器外壳要进行接地处理，起到静电屏蔽作用，防止丢线。

## 2.10 其他机械结构

另外，我们在调试过程中遇到好多困难，小车总是冲出赛道，因此撞坏了不少舵机，所以我们加装了防撞杆，并写了冲出赛道保护程序，效果明显；在模型车的机械结构方面还有很多可以改进的地方，比如说底盘、车身高度、悬架等。车的后轮部分有两个后轮轴固定座，不同的安装方式可使车的后体有不同的高度，下图 2.6 为固定轴实物图。



图 2.6 固定轴实物图

合理的底盘刚度和底盘高度调节会提高智能汽车的加速性能。但过低的底盘不仅容易使车架在坡道上而且很伤坡道，所以最终我们在不影响坡道的情况下只用了几个螺丝垫片适当的降低了一点底盘。

## 2.11 小结

通过对 B 车模的熟悉和了解，我们在比赛规则的允许范围内，制定了整车

机械系统的设计安装和改进方案，使其能够符合智能车的竞技要求。主要把原舵机固定去掉和后面的减震部分去除，在原舵机位置放置了电路板减震部件下面安装了电池。最终使车看起来更加简洁，分布更加合理。

## 第三章 电路设计说明

关于硬件设计，我们制定了一些目标原则：可靠、简洁。

可靠：电路进行单片化设计，将电源模块、调试模块、图像信号处理模块、速度反馈信息处理模块和各个接口电路设计在一块主板上。主板直接用螺丝螺母固定于底盘上。电路做好了模拟部分和数字部分的隔离、各个电源的滤波，以及对于干扰信号的屏蔽工作。各个接口连接可靠牢固。

简洁：主板外形设计根据车模尺寸、形状量身打造。器件排放整齐。减少了各个电路板之间的连线，使整车看上去简洁美观。

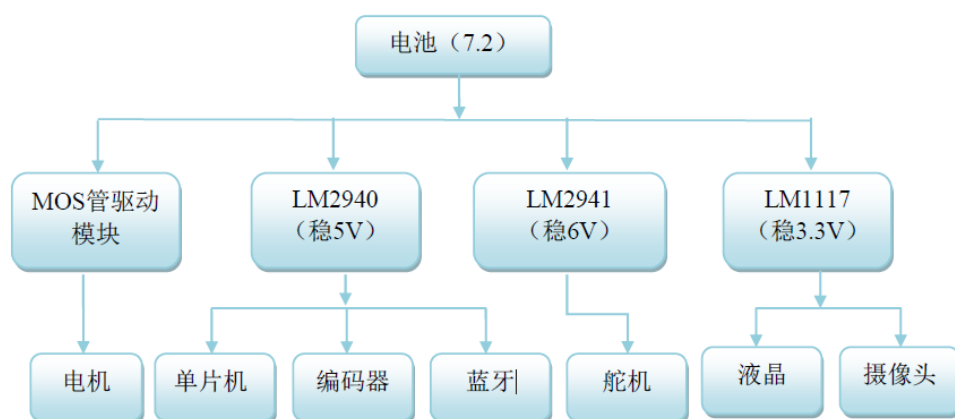


图 3.1 电源分配

### 3.1 电源管理模块

为了满足需求，我们的板子上面总共存在 5 种供电电压。

- 1) 智能车的系统电源由 7.2V 的镍镉可充电电池提供。
- 2) 5V 供电电源，主要给摄像头、红外对管、编码器、陀螺仪、LM1117 等器件进行供电，使用的稳压芯片为 TPS7350, 其电路图如图 3.2 所示。

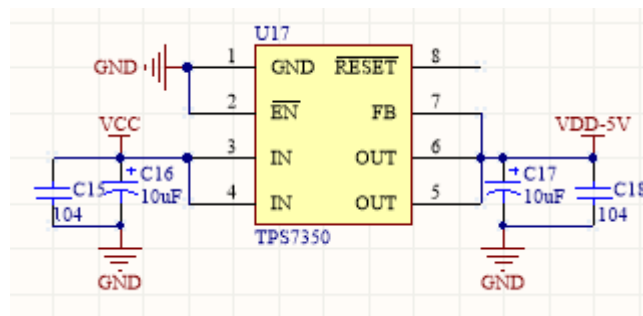


图 3.2 TPS7350 稳压芯片

- 3) 5.5V 舵机供电电压，稳压芯片为 LM1085ADJ，其特点是电压可调，可根据实际情况，改变电压值，来进行调试，保证舵机的灵敏性也能防止烧坏舵机。其电路图如图 3.3 所示。

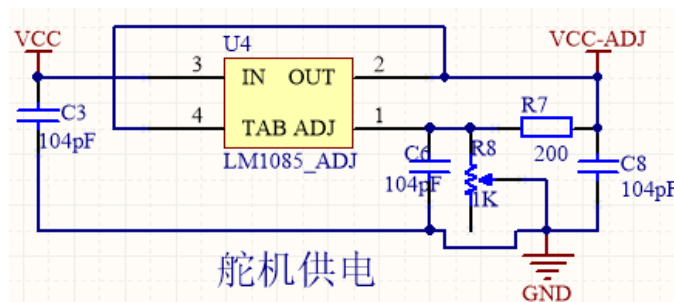


图 3.3 LM1085ADJ 稳压芯片

- 4) 3.3V 为 K60 供电，稳压芯片为 LM1117，其电路图如图 3.4 所示。

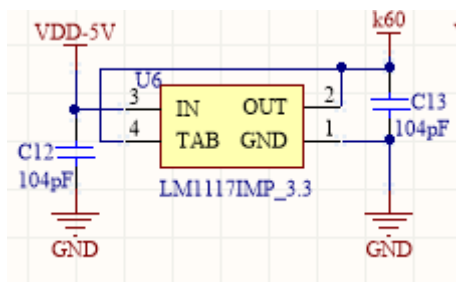


图 3.4 LM1117 稳压芯片

- 5) 12V 供电电压，为电桥驱动供电。采用 LM2577S-ADJ，其电路图如 3.5 所示。

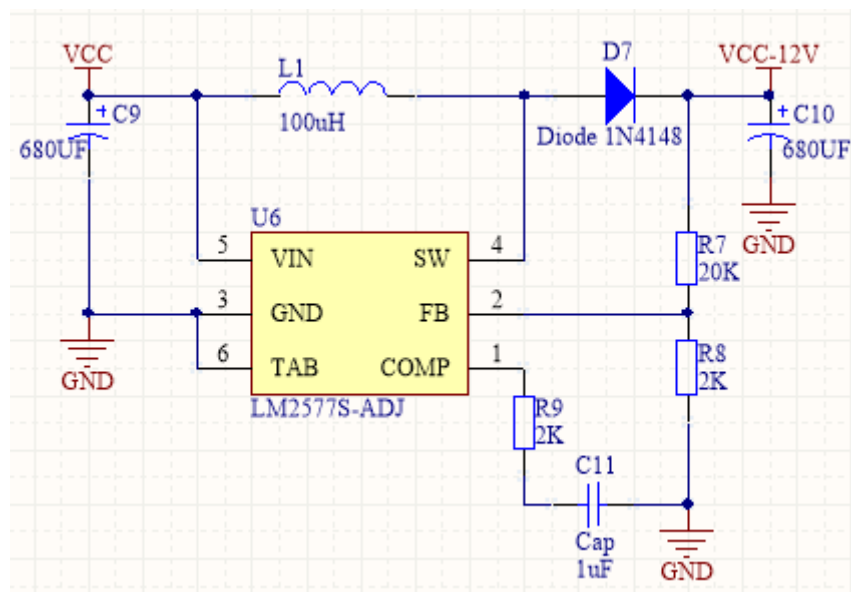
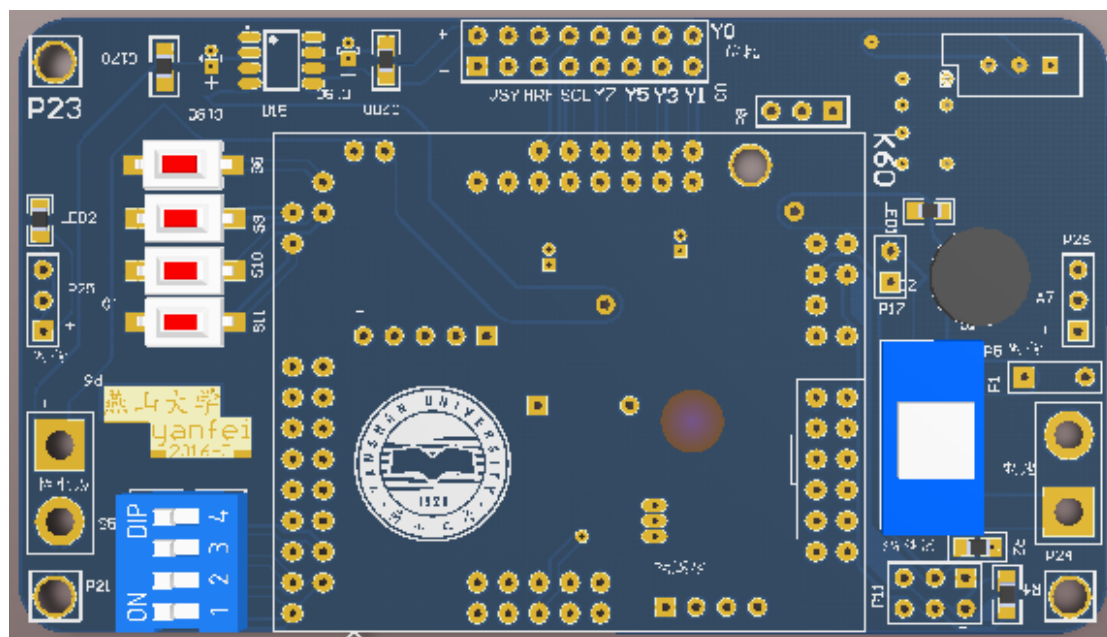


图 3.5 LM2577S-ADJ 稳压芯片

### 3.2 主控板的设计

我们本着可靠、高效、精致的原则进行设计，即是各个模块稳定，接地、屏蔽、滤波这几个方面做到位，不会发生意外。电路板上上面连线最短，面积小，利于安装。图 3.6 是我们设计的主控板。



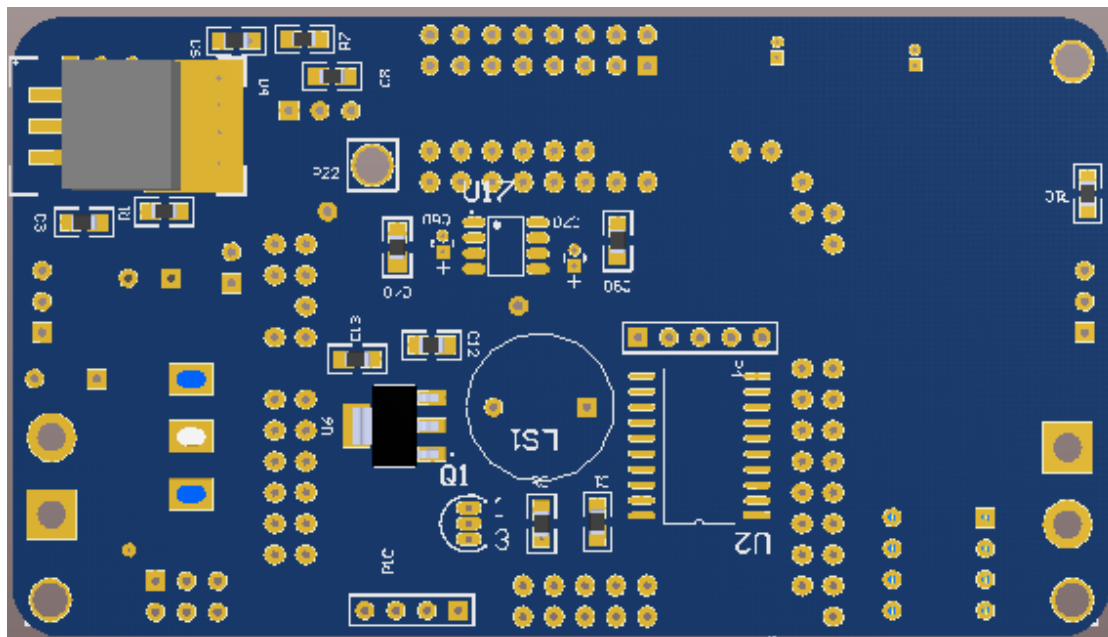


图 3.6 主控图

### 3.3 电机驱动模块

电机的驱动方案有 MOS 管，MC33886，BTS7960 或 BTN7971B 三种主要的方案。MC33886 芯片的内阻较大，发热明显，芯片较大引脚较多，占用空间较多最终导致 PCB 板太密。而 BTN 和 BTS 系列驱动芯片外围电路简单，但是内部集成度较高，芯片发热较厉害，并且一片 BTN 和 BTS 系列的驱动芯片价格非常昂贵，成本是 MOS 管和 MC33886 的 4 倍还多，这还是在考虑芯片烧毁的情况下。而 MOS 管的搭建效率较高，驱动电流很大，只需要驱动芯片因为 MOS 导通的电压为 12V，我们用的半桥驱动芯片为 IR2104 它的输入电压可达 60V 完全满足工作电压而且价格便宜，芯片即使烧毁所造成的损失也是可以接受的。控制桥电路图如 3.7 所示：



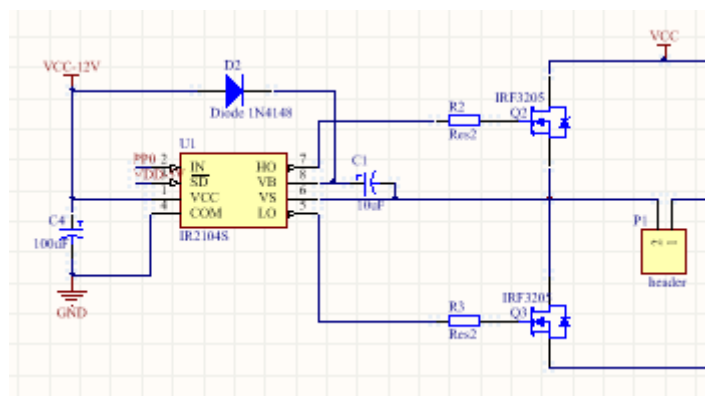


图 3.7 半桥驱动部分示意图

### 3.4 起跑线检测红外对管电路设计

检测起跑线，我们采用红外对管，原理如图 3.8 所示。当没有检测起跑线的时候，接收端导通，OUT 输出高电平，当检测到黑线的时候，接收端断路，OUT 为低电平，这样就可以区分出起跑线了。R12 的作用是分压，防止电压过大，导致单片机引脚击穿。

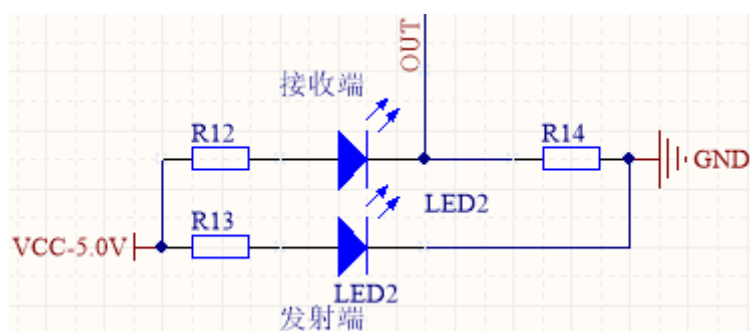


图 3.8 红外对管电路示意图

## 第四章 软件算法设计

高效的软件程序是智能车高速平稳自动寻线的基础。我们设计的智能车系统采用 CMOS 摄像头进行赛道识别，利用单片机的快速处理能力进行图像采集以及信息处理。配合采集行数达到失真矫正的效果，不断优化算法和路径。在智能车的转向和速度控制方面，我们使用经典高效的增量式 PID 算法，配合鲁棒控制，实现快速加减速，配合使用理论计算和实际参数补偿的办法，使智能车能够稳定快速寻线。

### 4.1 原始图像的特点

在图像信号处理中我们提取的赛道信息主要包括：赛道两侧边沿点位置、通过校正计算的赛道中心位置，中心点规划面积，赛道变化幅度，赛道类型判别。

在数字摄像头返回的这个矩阵中，每一个像素点都有一个从 0 至 255 的灰度值，值越大表示该像素点越亮，相反，越暗的像素点值就越小。同色间的灰度差很小，而不同色间的灰度差很大，通过给两个像素值作差得到的结果，就能够判断出当前像素位置有无发生灰度跳变以及灰度跳变的方向（从高灰度进入低灰度或相反）。我们所使用的方法与二值化的方法比较的优点在于适应性更强，避免不必要的运算，阈值选择更为方便且可靠性有所提高。

由于摄像头自身的特性和远小近大的视觉原理，图像会产生梯形式变形，这使得摄像头看到的赛道信息和实际真实的赛道信息有所区别。因此我们利用赛道进行测量和标定，得到一系列的参数，将摄像头采到的原始图像还原出真实赛道信息。远处的图像小，近处的图像大，黑线为梯形状。

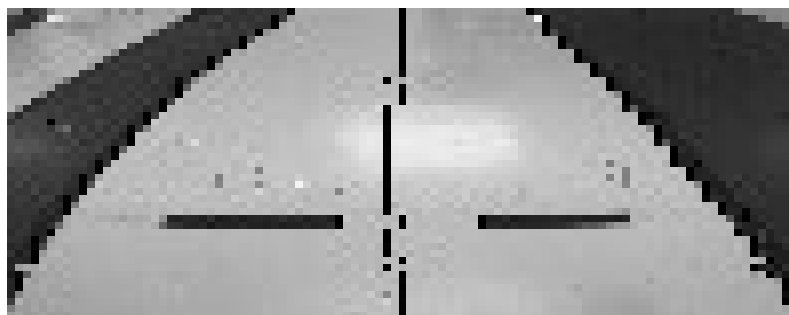


图 4.1 起跑线

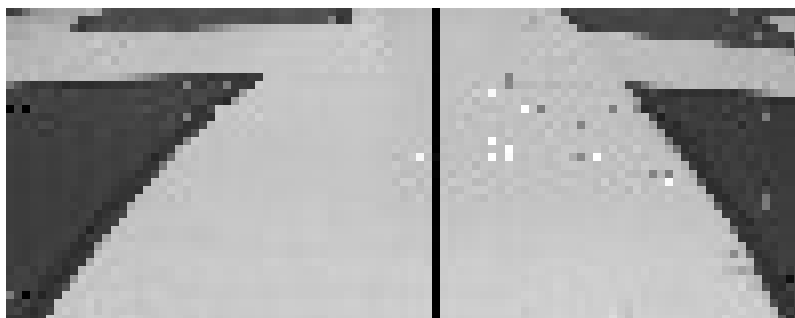


图 4.2 十字图像

## 4.2 赛道中心线提取及处理

在对赛道图像的处理过程中，主要运用边沿巡线法对赛道边线进行提取。由于使用数字摄像头，可以直接对边沿通过灰度值比较，进行跳变检测，提取中线并进行偏差处理。针对特殊赛道，比如十字，障碍等全部依托摄像头识别，对边线进行修正，但由于光线、杂点、赛道远处图像不清楚的干扰，图像效果会大打折扣。因此，借助陀螺仪检测坡道，光电管检测起跑线，稳定简单。

## 4.3 赛道边沿提取

经过多种方案的尝试和对赛道元素的考虑，我们采取边沿提取算法。边沿提取算法的基本思想如下：

（1）通过逐行扫描原始图像寻找最长白线两侧的左右边缘，根据阈值直接确定边界位置，提取黑白跳变点；

（2）确定图像全部信息，缺口位置，大小等，对于赛道宽度有一个范围，在确定的赛道宽度范围内提取有效赛道边沿，这样可以滤除不在宽度范围内的干扰；

（3）求边沿点时，因为近处的图像稳定，远处图像不稳定，所以采用由近及远的扫描办法；

（4）弯道的处理配合横向和纵向扫线，真实的反映赛道特征，减少摄像头丢线带来的失真；

## 4.4 边线处理

去往年相比，今年赛道元素简单，针对本届大赛的各类元素，十字，障碍，

坡道。我们利用图像边线的特征对其进行识别和补线。在补线的过程中，仅是对两边边线进行了校正。由于赛道元素的多样性以及赛车在赛道上姿态的不确定性，在寻找特征的过程中，处理下一个元素时，需要排除其对各类赛道的影响，准确的寻得其特征，避免误判。提取到中线后，我们利用小车的图像中心与赛道的实际中心的偏移量来控制舵机的转动。

(1) 十字的边缘并没有进行补线，而是通过按列统计法，选择寻找最长白线，确定中心位置，人为给定中线，即便车身倾角很大也可以顺利通过，适应性，稳定性很强，是传统的补线法远不及的；

(2) 障碍的处理同样采用按列统计法处理，稳定，可以达到通过障碍不会明显打角的效果；

(3) 小 S 采用加权平均的算法，转换成直线处理；

(4) 由于比赛坡道高度的不确定性，我们采用陀螺仪进行检测，程序中只用了 X 轴方向的数据。小车在水平方向行驶的时候，理想情况下，X 轴输出的电压值为零偏值（即小车静止时的值），当小车上坡道时，车身迅速由水平变为倾斜，陀螺仪检测到 X 轴上的角速度，输出大于零偏值的电压。通过与设定的阈值比较，若大于阈值则判断为坡道。在实际情况下，小车在行驶过程中由于加减速过猛和迅速过弯也会使陀螺仪的输出在零偏值上下波动，因此我们采取通过均值滤波来消除抖动。通过实践证明用陀螺仪检测坡道的方法可行，且当小车车身不正时仍可以判别出坡道。

下图是我们程序的整体控制策略：

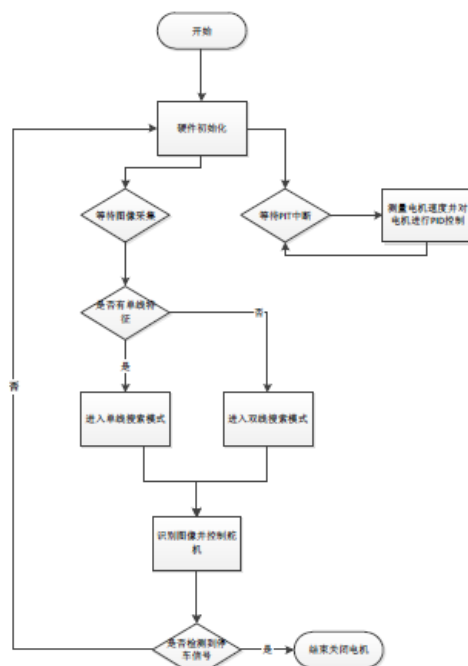


图 5.1 主程序框图

图 4.3 控制流程图

## 4.5 中断的调用

我们通过调用定时器中断对每个控制模块进行周期控制，用 2ms 的控制周期来检测起跑线，用 5ms 的控制周期来进行速度脉冲采集和速度控制，用 16.64ms 的控制周期来进行图像采集和方向控制，以确保主程序执行的实时性以及合理性。

## 4.6 电机控制策略

### 4.6.1 PID 控制算法介绍

在工程实际中，应用最为广泛的调节器控制规律为比例、积分、微分控制，简称 PID 控制，又称 PID 调节。PID 控制器问世至今已有近 70 年历史，它以其结构简单、稳定性好、工作可靠、调整方便而成为工业控制的主要技术之一。当被控对象的结构和参数不能完全掌握，或得不到精确的数学模型时，控制理论的其它技术难以采用时，系统控制器的结构和参数必须依靠经验和现场调试来确定，这时应用 PID 控制技术最为方便。即当我们不完全了解一个系统和被控对象，或不能通过有效的测量手段来获得系统参数时，最适合用 PID 控制技

术。PID 控制，实际中也有 PI 和 PD 控制。

PID 控制器是一种线性控制器，它根据给定值与实际输出值构成控制偏差。将偏差的比例(P)、积分(I)和微分(D)通过线性组合构成控制量，对被控对象进行控制，故称 PID 控制器，原理框图如图 4.4 所示。

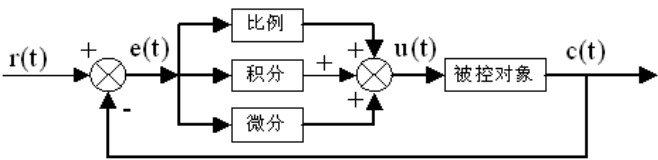


图 4.4 PID 控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

$$e(k) = r(k) - c(k) \tag{4.6.1}$$

(4.6.2)

式中：

- $k$  —— 采样序号， $k = 0, 1, 2 \dots$ ;
- $r(k)$  —— 第  $k$  次给定值;
- $c(k)$  —— 第  $k$  次实际输出值;
- $u(k)$  —— 第  $k$  次输出控制量;
- $e(k)$  —— 第  $k$  次偏差;
- $e(k-1)$  —— 第  $k-1$  次偏差;
- $K_p$  —— 比例系数;
- $T_i$  —— 积分时间常数;
- $T_d$  —— 微分时间常数;
- $T$  —— 采样周期。

简单说来，PID 控制器各校正环节的作用如下：

比例环节：及时成比例地反映控制系统的偏差信号，偏差一旦产生，控制器立即产生控制作用，以减少偏差。

积分环节：主要用于消除静差，提高系统的无差度。积分作用的强弱取决

于积分时间常数，越大，积分作用越弱，反之则越强。

微分环节：能反映偏差信号的变化趋势(变化速率)，并能在该偏差信号变得太大之前，在系统中引入一个有效的早期修正信号，从而加快系统的动作速度，减小调节时间。

数字 PID 控制算法通常分为位置式 PID 控制算法和增量式 PID 控制算法。

#### 4.6.2 位置式 PID

位置式 PID 中，由于计算机输出的  $u(k)$  直接去控制执行机构(阀门)， $u(k)$  的值和执行机构的位置(如阀门开度)是一一对应的，所以通常称公式(4.6.2)为位置式 PID 控制算法。

位置式 PID 控制算法的缺点是：由于全量输出，所以每次输出均与过去的状态有关，计算时要对过去  $e(k)$  进行累加，计算机工作量大；而且因为计算机输出的  $u(k)$  对应的是执行机构的实际位置，如计算机出现故障， $u(k)$  的大

幅度变化，会引起执行机构位置的大幅度变化，这种情况往往是生产实践中不允许的，在某些场合，还可能造成严重的生产事故。因而产生了增量式 PID 控制的控制算法，所谓增量式 PID 是指数字控制器的输出只是控制量的增量  $\Delta u(k)$ 。

#### 4.6.3 增量式 PID

当执行机构需要的是控制量的增量(例如：驱动步进电机)时，可由式(4.6.2)推导出提供增量的 PID 控制算式。由式(4.6.2)可以推出式(4.6.3)，式(4.6.2)减去式(4.6.3)可得式(4.6.4)。

$$u(k-1) = K_p \{e(k-1) + \frac{T}{T_I} \sum_{j=0}^{k-1} e(j) + \frac{T_D}{T} [e(k-1) - e(k-2)]\} \quad (\text{公式 4.6.3})$$

$$\begin{aligned} \Delta u(k) &= K_p \{[e(k) - e(k-1)] + \frac{T}{T_I} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)]\} \\ &= K_p \Delta e(k) + K_I e(k) + K_D [\Delta e(k) - \Delta e(k-1)] \end{aligned} \quad (4.6.4)$$

$$\text{式中： } \Delta e(k) = e(k) - e(k-1); \quad K_I = K_p \frac{T}{T_I}; \quad K_D = K_p \frac{T_D}{T}$$

公式(4.6.4)称为增量式 PID 控制算法，可以看出由于一般计算机控制系统

采用恒定的采样周期 $T$ ，一旦确定了 $K_p$ 、 $T_I$ 、 $T_D$ ，只要使用前后三次测量值的偏差，即可由式(4.6.4)求出控制增量。

增量式 PID 具有以下优点：

(1) 由于计算机输出增量，所以误动作时影响小，必要时可用逻辑判断的方法关掉。

(2) 手动/自动切换时冲击小，便于实现无扰动切换。此外，当计算机发生故障时，由于输出通道或执行装置具有信号的锁存作用，故能保持原值。

(3) 算式中不需要累加。控制增量 $\Delta u(k)$ 的确定仅与最近 $k$ 次的采样值有关，所以较容易通过加权处理而获得比较好的控制效果。

但增量式 PID 也有其不足之处：积分截断效应大，有静态误差；溢出的影响大。使用时，常选择带死区、积分分离等改进 PID 控制算法。

#### 4.6.4 舵机控制

在舵机的控制当中，根据图像处理的结果来得到一个合适的偏移量，然后通过这个偏移量反馈到舵机的打角上。在偏移量的计算当中，我们通过对整副图中线加权的方式来计算偏移量。给每一行设定一个权值，最后通过所有有效行的加权来计算出当前图像的偏移量。在小车跑动中，因为不需要考虑小车之前走过的路线，所以，我们舍弃了 I 控制，将小车舵机的 PID 控制简化成 PD 控制。PD 控制当中的偏差就是之前通过整幅图加权的方式得出。每一行加权的权重是经过长期整定的，最后得出了一套合理的，适应性强的权值。

#### 4.6.5 电机控制

速度控制对于小车的快速行进以及必要时候的减速起到了至关重要的作用，我们在尝试使用传统 PID 方法之后，发现并不能达到我们理想的速度控制需求，于是对其进行了自己的改进，加入了额外项进行修正，取得了不错的效果。



## 第五章 系统开发及调试工具

### 5.1 开发工具

#### (1) 软件开发工具

程序开放在 IAR Embedded Workbench IDE 下进行，Embedded Workbench for ARM 是 IAR Systems 公司为 ARM 微处理器开发的一个集成开发环境(下面简称 IAR EWARM)。比较其他的 ARM 开发环境，IAR EWARM 具有入门容易、使用方便和代码紧凑等特点。

EWARM 中包含一个全软件的模拟程序(simulator)。用户不需要任何硬件支持就可以模拟各种 ARM 内核、外部设备甚至中断的软件运行环境。从中可以了解 and 评估 IAR EWARM 的功能和使用方法。

在程序的编译和运行当中，遇到的大部分问题，都能够通过 IAR 的在线调试中解决。例如其中 IAR C-SPY Debugger 允许用户查看在源代码中查看变量或表达式，使用户可以在运行程序时跟踪其值的变化，并且查看变量的方式有许多种，我们使用 Live watch 功能，在小车实时运行中观察变量的变化，进而分析程序执行情况。

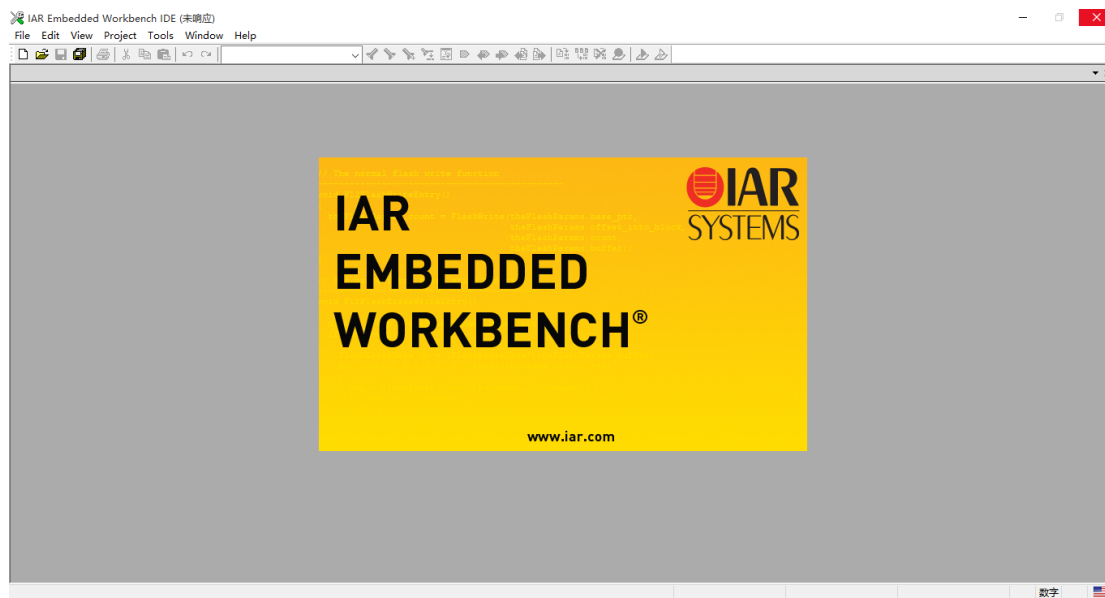


图 5.1 IAR 启动界面

IAR 自带的在线调试功能需要下载程序后才能进入相关功能。

## （2）硬件开发工具

硬件在 Altium Designer 中进行设计与布局，Altium Designer 是原 Protel 软件开发商 Altium 公司推出的一体化的电子产品开发系统，这套软件通过把原理图设计、电路仿真、PCB 绘制编辑、拓扑逻辑自动布线、信号完整性分析和设计输出等技术的完美融合，为设计者提供了全新的设计解决方案，使设计者可以轻松进行设计，复杂的电路板布局 Altium Designer 都可以非常很容易地设计出有效无误的 PCB 板。同时可以进行文档输出与分享，凭借强大的布线技术，以及丰富的元件库，使用者很方便的完成硬件方面的制作。

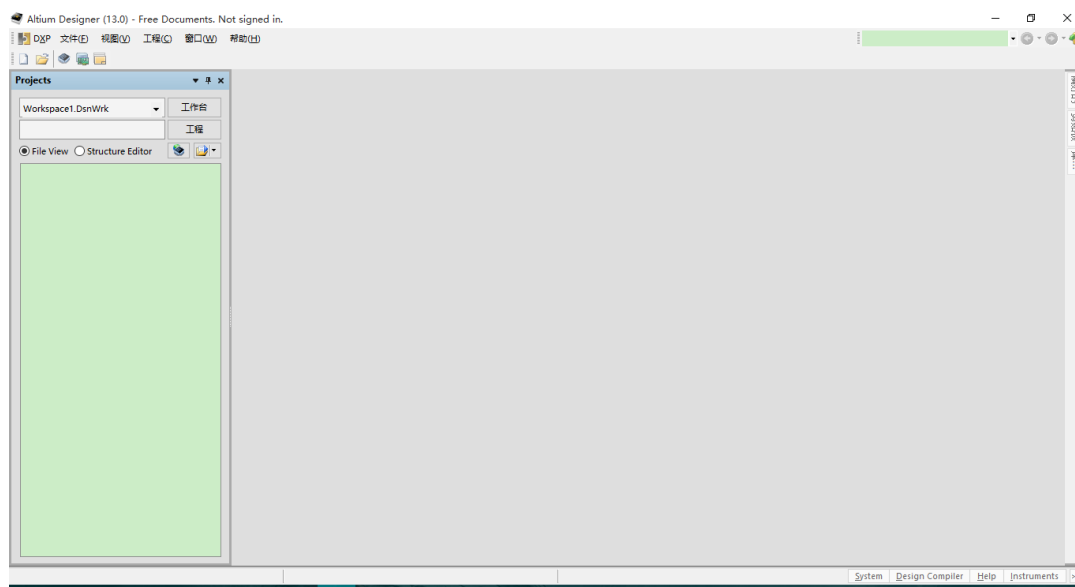


图 5.2 Altium Designer 工作界面

Altium Designer 有一个十分好用的功能就是文件变更自保存功能，强大的数据管理工具，掌控使用者的设计数据和 workflows。使用灵活的生命周期管理系统，可以精确地控制设计数据；使用强大的版本控制系统，可以跟踪每一个设计变更。

## 5.2 上位机图像显示

为了直观观察摄像头图像的效果，但出于时间的关系，我们并未编写上位机，转而直接采用了现成的上位机。配合山外上位机进行调试。主要在发送的图像上面加上图像中线显示以及 PID 整定方面以便分析，我们用到的上位机界面展示：

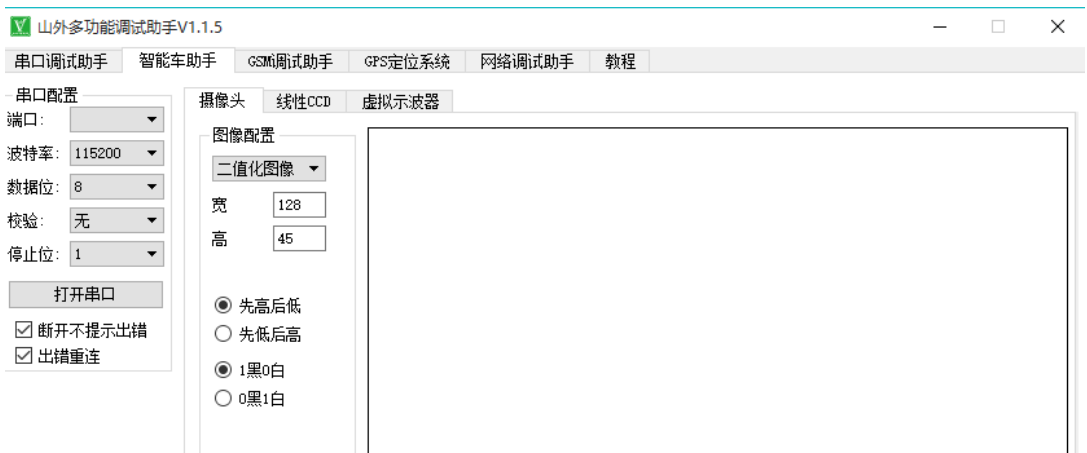


图 5.3 图像界面

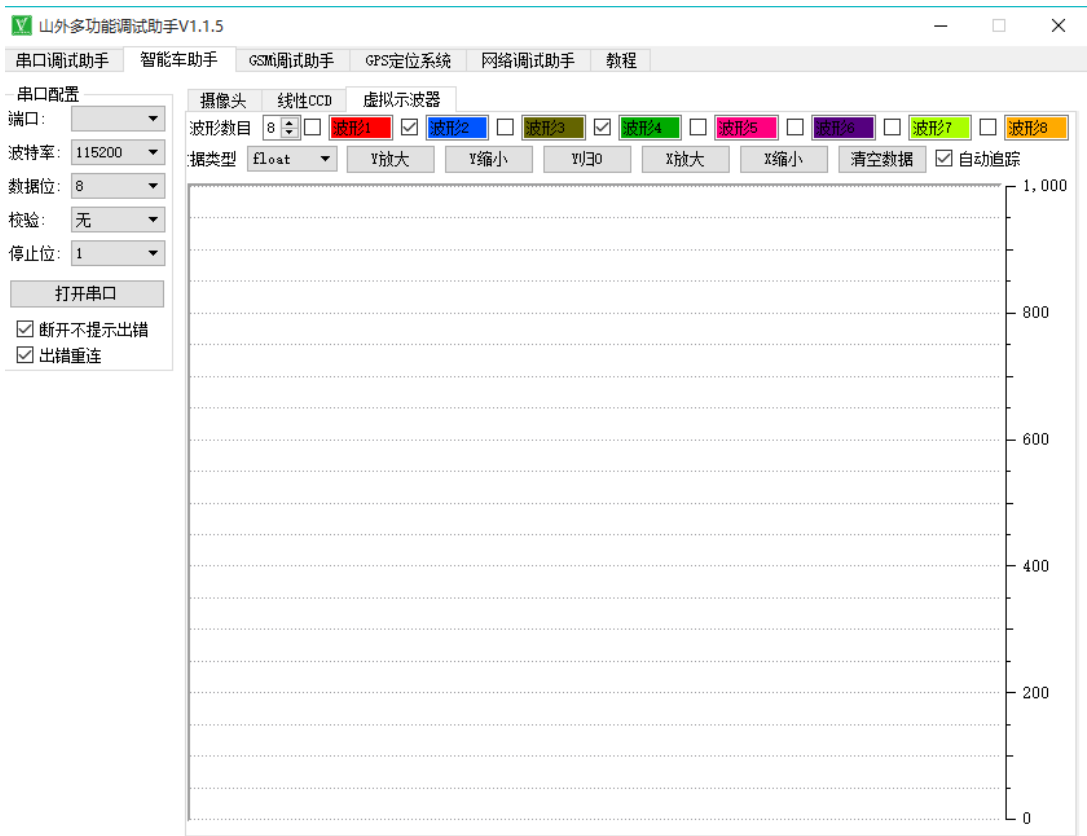


图 5.4 虚拟示波器界面

### 5.3 蓝牙调试模块与串口通信

在小车的调试过程，经常需要查看某些变量的值，与上位机之间进行通讯，于是给小车设计了蓝牙通讯模块与串口通信，使小车在行驶过程中将某些重要

变量通过串口发送到上位机进行观察、分析、处理。

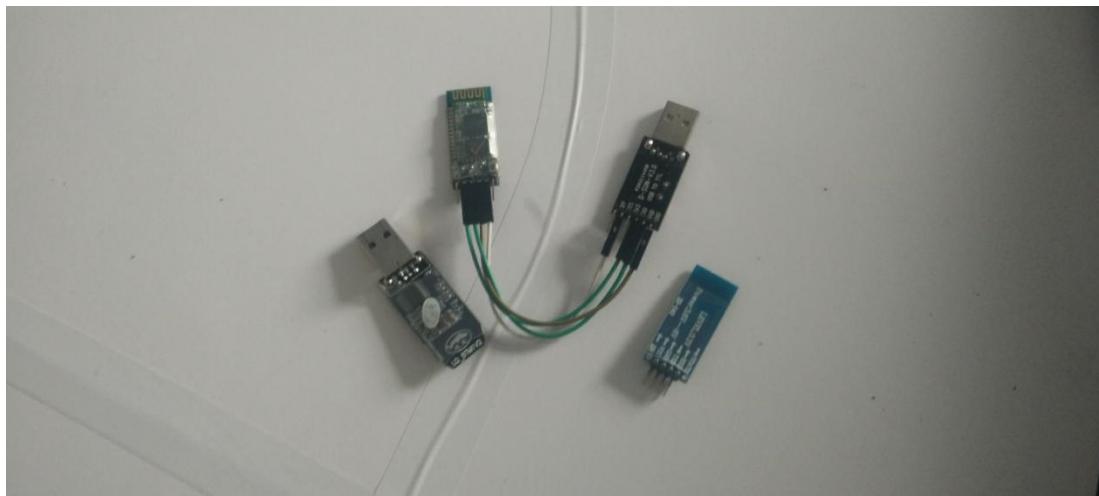


图 5.5 蓝牙与串口模块

#### 5.4 信号发生器以及示波器的使用

在电路板的设计过程中，前期需要不断在洞洞板上的进行测试，在没有任何把握或是理论基础的前提下，就提交 PCB 很容易造成时间和金钱上的浪费，尤其是在尝试新方案的时候。所以电路的设计和检测是离不开万用表和示波器的。



图 5.6 信号发生器与稳压电源

信号发生器与稳压源主要用来测试主控板与桥之间的连接性能，保证在输出很大占空比的情况下主控电压不会被拉低。

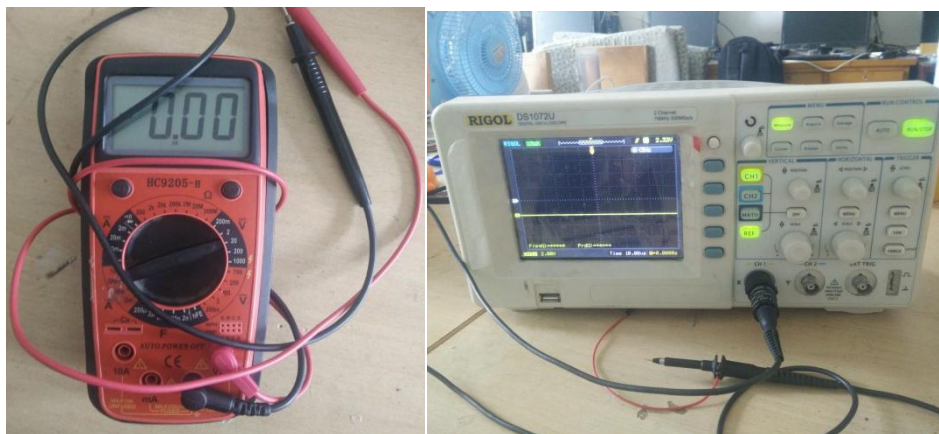


图 5.7 万用表与示波器

万用表与示波器是经常使用的工具，主要用于电压调节与测量，以及观察信号的发生状况。

## 第六章 总结

在这段日子中，我们遇到了各种问题，虽然大部分已经解决，但是我们仍有不少未解决的问题，比如电路板信号噪声很大，芯片发热严重，机械结构等仍有明显不足，从而限制了速度的提升；算法结构也需优化，细节方面也要重新处理，希望小学弟们能够解决这些问题，速度更快，过弯更流畅。

最后，我们预祝“恩智浦”智能车竞赛越办越好，希望燕山大学能在智能车大赛中取得更好的成绩！

## 致 谢

我们团队从二月到现在，从懵懂浅知到幸运地冲入决赛，这段日子想起来真是充满了各种回忆，走进实验室里仍交错着各种情感气息，有过伤心，也有欢笑，有争吵，也不少谈心。几个人从菜鸟开始，慢慢学起，上网查阅各种资料，请教学长，向大神学习，因为时间的紧迫，也少不了熬夜与逃课。在这段时间中我们懂得了如何学习，如何合作，也让我们理解到学的越多，要学的就越多。这是一个成长的过程，感谢宋老师与呼老师的支持，感谢鹏程师兄、胖师兄、杰哥、沛哥、晖哥的指导，德来学长、帅林学长、超哥的帮忙，感谢你们这些日子的陪伴，才让我们取得这样的成绩。我们相信：只要不放弃，并且向着目标一直努力着一定会有收获。

在这几个月的备战过程中，场地和经费方面都得到了学校和学院的大力支持，在此特别感谢一直支持和关注智能车比赛的学校和学院领导以及各位指导老师、指导学长，同时也感谢比赛组委会能组织这样一项有意义的比赛。

## 参考文献

- [1] 卓晴, 黄开胜, 邵贝贝. 学做智能车[M], 北京: 北京航空航天大学出版社, 2007:1-369.
- [2] 王宜怀. 嵌入式系统, 北京: 北京航空航天大学出版社.
- [3] 邵贝贝. 单片机嵌入式应用的在线开发方法[M], 北京: 清华大学出版社, 2004:1d8-63.
- [4] 谭浩强. C 程序设计[M], 北京: 清华大学出版社, 1999, 87-186.
- [5] 王晓明. 电动机的单片机控制[M], 北京:北京航空航天大学出版社. 2002.
- [6] 徐国华. 移动机器人的发展现状及其趋势[J]. 机器人技术与应用,2001(3):7-14.
- [7] 尹念东. 智能车辆的研究及前景[J]. 上海汽车, 2002(2):10-11.
- [8] 王荣本. 世界智能车辆研究概述[J]. 公路交通科技, 2001(10) :13-15.



## 附录 A：程序源代码

```
#include "main.h"
#define AMD      0
InteractionFac THK = {700,0,700}; //备参
#if 1// (defined (Main))
void main()//0811//新车
{
    DisableInterrupts
    Sth_Init();
    SetIrqPriority();    //重新赋值中断优先级
    DELAY_MS(1000);
    EnableInterrupts
#if (AMD)
    DELAY_MS(1000);
    LimitIntellective();//更新阈值
#endif
    //BangBangGo(200);
    //DELAY_MS(500);      //滞磁启动
    for(;;)
    {
        //L3G4200D_Init();//每次都初始化防止掉线或卡死
        if(UpdataPhoto)//已传递 Buff 数组
        {
            gpio_set (TIME_CHECK_PIN,0);//约 840us
            UpdataPhoto = 0;
            DataProcess();
            DisplayInfo();
            enable_irq (PORTC_IRQn);
            steer_go(SteeringPDCalc(&STEERING) + steer.median);
            //steer_go(steer.median);
            OutEdgeStop();//保护//为什么出去后关闭 pid 中断后疯转，需要关闭全部中断
            才行？可能是 FTM 模块一直使能吧？？？
        }
    }
}
#else
void main()
{
```

```
#if 0//对管测试
```

```
uint8 var1, var2;
OLED_Init();
adc_init(ADC0_SE10);
adc_init(ADC0_SE15);
while(1)
{
    var1 = adc_once(ADC0_SE10,ADC_8bit);
    var2 = adc_once(ADC0_SE15,ADC_8bit);
    OLED_Print_num_int(0,1,var1);
    OLED_Print_num_int(0,2,var2);
    DELAY_MS(500);
    LCD_CLS();
}
```

```
#elif 1//陀螺仪测试
```

```
IIC_init();
OLED_Init();
L3G4200D_Init();
for(;;)
{
    L3G4200D_Init();//每次都初始化防止掉线或卡死
    x_GyroValue = Get_Avx(>>3);
    //y_GyroValue = Get_Avy(>>3);
    //z_GyroValue = Get_Avz(>>3);
    OLED_Print_num_int(0,1,x_GyroValue);
    DELAY_MS(500);
    LCD_CLS();
}
```

```
#elif 0//新板功能测试
```

```
led_init(LED_MAX);
key_init(KEY_MAX);
Boma_Init();
OLED_Init();
Buzzer_Init();
while(1)
{
    if(1==BomaRead()) {
        OLED_P6x8Str(0,3,"2.Img_Change:");
        if
(!gpio_get(KEY5_PIN)||!gpio_get(KEY6_PIN)||!gpio_get(KEY7_PIN)||!gpio_get(KEY8_PIN)) {
```

```

        DELAY_MS(20);
        if
(!gpio_get(KEY5_PIN)||!gpio_get(KEY6_PIN)||!gpio_get(KEY7_PIN)||!gpio_get(KEY8_PIN)) {
            DELAY_MS(20);
            led(LED_MAX,LED_ON);
            buzzer_on
        }
    }
else {
    led(LED_MAX,LED_OFF);
    buzzer_off
}
}
}
#endif
}
#endif

#include "PidController.h"
float global_test = 0;
//short global_test1 = 0;
/*
*函数功能：舵机控制
*时间：2016_02_02
*/
float SteeringPDCalc(PD_Steering *pp)//__RAMFUNC 不知道写入 ram 会不会工作
{
    float p_out = 0, d_out = 0;

    pp->d_error = pp->error - pp->last_error;           //计算误差微分
    pp->last_error = pp->error;                           //更新 last_error

    p_out = pp->P * pp->error;
    d_out = pp->D * pp->d_error;
    //global_test = d_out;
    if (d_out > pp->D_l) {                                  //对微分相设置 Deadline
        d_out = pp->D_l;
    } else if (d_out < (- pp->D_l)) {
        d_out = - (pp->D_l);
    }
}

```

```
pp->out = p_out + d_out;

if (pp->out > steer.R_limit) {
    pp->out = steer.R_limit;
} else if (pp->out < steer.L_limit) {
    pp->out = steer.L_limit;
}
return pp->out;
}

/*
*函数功能: 计算给电机的占空比 (位置)
*时间: 2016_05_08
*/
short MotorPIDCalc(PID_Motor *pp)
{
    //static uint8 NE = 0;
    float p_out = 0, i_out = 0, d_out = 0;
    if (3 == Stop_Flag)//停车标志
    {
        //      if((pp->backpulse > 3) && (NE <= 1))
        //      {
        //          NE = 1;
        //          DesiredSpeed = 0;
        //      }
        //      else
        //      {
        //          NE = 2;
        //          return 0;
        //      }
        //      DisableInterrupts
        //      while(1) {
        //          steer_go(THK.steerfac+steer.median);//防止偏离
        //          BangBangBack(THK.speedfac);//反接制动
        //          DELAY_MS(THK.timefac);
        //          motor_stop();
        //          while(1);
        //      }
        DesiredSpeed = 0;
    }
}
```

```

    }
    //DesiredSpeed = 42;
    pp->setpulse = DesiredSpeed;
    //pid 参数计算
    pp->error = pp->setpulse - pp->backpulse;
    pp->d_error = pp->error - pp->last_error;
    pp->last_error = pp->error;           //更新 last_error

    //此处为积分分离，当偏差变化太大时不积分
    if (fabs(pp->error) < 5) {
        pp->i_error += pp->error;
    }

    // if (pp->error < pp->robustdown) { //鲁棒控制
    //     //gpio_set (PTE25,1);
    //     pp->out = -600;
    // } else if (pp->error > pp->robustup) {
    //     //gpio_set (PTE26,1);
    //     pp->out = 600;
    // }
    // else {                               //PID 控制
        //gpio_set (PTE27,1);
        p_out = pp->P * pp->error;
        i_out = pp->I * pp->i_error;
        d_out = pp->D * pp->d_error;

        if (i_out > pp->I_limit) {
            i_out = pp->I_limit;
        } else if (i_out < (- pp->I_limit)) {
            i_out = -(pp->I_limit);
        }
        if (d_out > pp->D_limit) {
            d_out = pp->D_limit;
        } else if (d_out < (- pp->D_limit)) {
            d_out = -(pp->D_limit);
        }
        //pp->out = (short)(pp->U_P*pp->setpulse + p_out + i_out + d_out); //顺反馈，加快反
应速度
        pp->out = (short)(p_out+ i_out/2 + d_out); //梯形积分
    //}

```

```
//global_test = d_out;
if (pp->out > 600) {
    pp->out = 600;
} else if (pp->out < (-600)) {
    pp->out = -600;
}
return pp->out;
}
/*
*函数功能：计算给电机的占空比（增量）
*时间：2016_08_03
*/
short PIDCalc(PID_Motor *pp)
{
    float iError,dError,ddError;

    float p_out=0,i_out=0,d_out=0,index=1;

    if (3 == Stop_Flag)//停车标志
    {
        DesiredSpeed = 0;
    }

    pp->setpulse = DesiredSpeed;

    //2 计算误差
    iError = pp->setpulse - pp->backpulse;// I 调节
    //与上次误差的误差
    dError = iError - pp->last_error; // P 调节
    //与上上次的误差
    ddError = iError + pp->prev_error - 2 * pp->last_error;
    if (iError < pp->robustdown)//鲁棒控制
    {
        pp->out = -700;
    }
    else if (iError > pp->robustup)
    {
        pp->out = 700;
    }
    else//3 增量计算
```

```
{
    p_out = pp->P * dError;
    i_out = pp->I * iError * index;
    d_out = pp->D * ddError;

    pp->out += p_out + i_out + d_out;
}

pp->prev_error = pp->last_error;
pp->last_error = iError;

if(pp->out > 700)
    pp->out = 700;
else if(pp->out < -700)
    pp->out = -700;

return pp->out;
}

//void Send_nihe(void)    //执行时间 10ms  40M 总线频率
//{
//    unsigned char SciBuf[40];
//    int temp0,temp1;
//    unsigned char DAT[4];
//    //
//    temp0=(int)(-Read_MMA8452_Z()*0.08785+90)*10; //加速度计
//    //
//    temp1=(int)((gf_Car_angle_I)+90)*10;
//    //
//    //temp0=(int)gf_CarSpeed;
//    //    temp1=(int)230;
//    //
//    DAT[0]=temp0;
//    DAT[1]=temp0>>8;
//    //
//    DAT[2]=temp1;
//    DAT[3]=temp1>>8;
//    //
//    SciBuf[0] = 0;    // 长度高字节，此设置为 0
//    SciBuf[1] = 12+4; // 16
```

```
//      SciBuf[2] = 0;    // 保留字节, 设置为 0
//      SciBuf[3] = 0;    // 保留字节, 设置为 0
//      SciBuf[4] = 0;    // 保留字节, 设置为 0
//      SciBuf[5] = 0;    // 保留字节, 设置为 0
//                               // 填充采集数据,共 12 个字节
//      SciBuf[6] = DAT[1]; // 通道 1 的高字节数据
//      SciBuf[7] = DAT[0]; // 通道 1 的低字节数据
//
//      SciBuf[8] = DAT[3]; // 通道 2 的高字节数据
//      SciBuf[9] = DAT[2]; // 通道 2 的低字节数据
//
//      SCI_SendData(&SciBuf[0]); // 发送到上位机
//
//
//}
//
//void SCI_SendData(unsigned char *data)
//{
//    int len;
//    unsigned char    lrc=0;
//    uart_putchar(UART4,'*'); // 发送帧头, 一个字节
//    len = (int)(data[0]<<8) | (int)(data[1]);
//    data += 2; // 调整指针
//    uart_putchar(UART4,'L'); // 发送帧类型, 共两个字节
//    uart_putchar(UART4,'D');
//    while(len--) // 发送数据的 ASCII 码, 含保留字节和 CCD 数据
//    {
//        SendHex(*data);
//        lrc += *data++;
//    }
//    lrc = 0-lrc; // 计算 CRC, 可以为任意值
//    SendHex(lrc); // 发送 CRC 校验 ASCII
//    uart_putchar(UART4,'#'); // 发送帧尾, 一个字节
//}
//
//void SendHex(unsigned char  data)
//{
//    unsigned char temp;
//    temp = data >> 4;
```



```
//    if (temp >= 10)
//    {
//        uart_putchar(UART4,temp - 10 + 'A');
//    }
//    else
//    {
//        uart_putchar(UART4,temp + '0');
//    }
//
//    temp = data & 0x0F;
//
//    if (temp >= 10)
//    {
//        uart_putchar(UART4,temp - 10 + 'A');
//    }
//    else
//    {
//        uart_putchar(UART4,temp + '0');
//    }
//}
```

```
/**
*****
***
```

函数功能：棒棒控制

入口参数：无

出口参数：无

时间：2016\_01\_29

```
*****
***/
```

void BangBangGo(short x)

```
{
    motor_go(x);
}
```

void BangBangBack(short y)

```
{
    motor_go(-y);
}
```