**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：大连理工大学

队伍名称：守望者

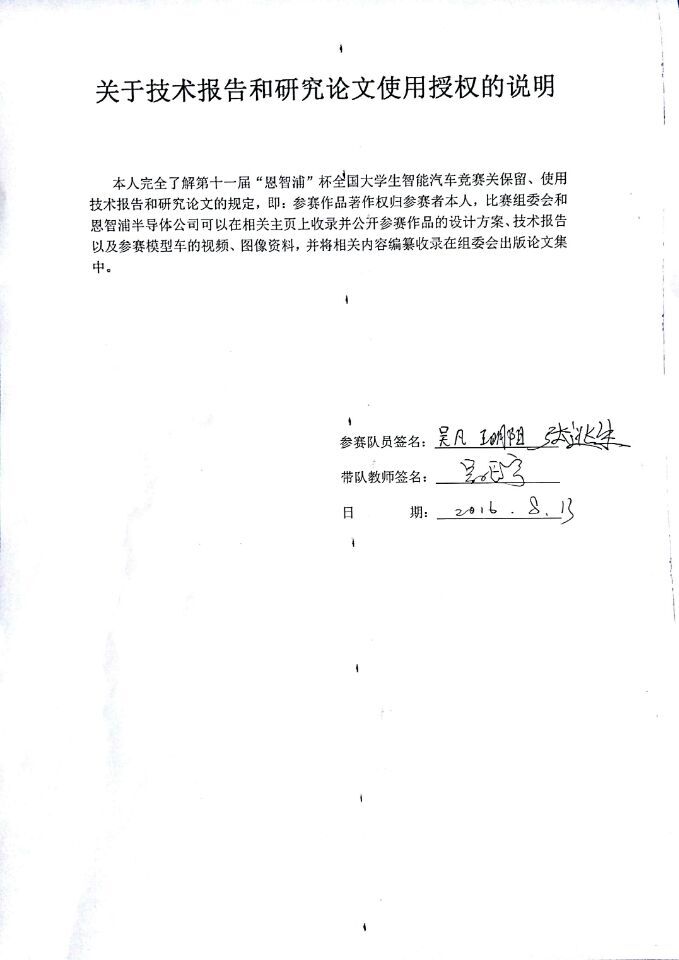
参赛队员：吴凡

张逃生

王明阳

带队教师：吴振宇

李胜铭

**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

[引 言 1](#_Toc458608890)

[第一章 方案设计 2](#_Toc458608891)

[1.1 系统总体方案的选定 2](#_Toc458608892)

[1.2 系统总体方案的设计 2](#_Toc458608893)

[1.3 小结 3](#_Toc458608894)

[第二章 机械设计制作及调整 4](#_Toc458608895)

[2.1 机械结构制作部分设计 4](#_Toc458608896)

[2.1.1摄像头的安装 4](#_Toc458608897)

[2.1.2车模倾角传感器的安装 4](#_Toc458608898)

[2.1.3测速传感器的安装 5](#_Toc458608899)

[2.1.4电路板的安装 5](#_Toc458608900)

[2.1.5电池安放 5](#_Toc458608901)

[2.1.6 舵机力臂的调整 5](#_Toc458608902)

[2.2 车模机械结构的调整与改装 6](#_Toc458608903)

[2.2.1 舵机改装 6](#_Toc458608904)

[2.2.2 底盘改装 6](#_Toc458608905)

[2.2.3 前轮定位 6](#_Toc458608906)

[2.2.4车轮外倾角 7](#_Toc458608907)

[2.2.5 前轮前束 7](#_Toc458608908)

[2.3智能车转向机构调整优化 8](#_Toc458608909)

[2.4智能车重心位置的调整 9](#_Toc458608910)

[2.5其它机械结构的调整 9](#_Toc458608911)

[2.6小结 9](#_Toc458608912)

[第三章 硬件电路方案设计 11](#_Toc458608913)

[3.1 单片机最小系统模块 11](#_Toc458608914)

[3.2 电源模块 12](#_Toc458608915)

[3.2.1 电池使用 12](#_Toc458608916)

[3.2.2 稳6V电源模块 12](#_Toc458608917)

[3.2.3 稳5V电源模块 13](#_Toc458608918)

[3.2.4稳3.3V电源模块 13](#_Toc458608919)

[3.3 电机驱动模块 13](#_Toc458608920)

[3.4 串口通信模块 14](#_Toc458608921)

[3.5 霍尔测速模块 14](#_Toc458608922)

[第四章 图像信息处理 16](#_Toc458608923)

[4.1 图像采集 16](#_Toc458608924)

[4.1.1 图像数据输出简介 16](#_Toc458608925)

[4.1.2 摄像头工作时序 16](#_Toc458608926)

[4.1.2.1 行中断时序 16](#_Toc458608927)

[4.1.2.2 场中断时序 17](#_Toc458608928)

[4.1.3 DMA简要介绍 17](#_Toc458608929)

[4.2 图像处理 18](#_Toc458608930)

[4.2.1 阈值的确定方法——硬件二值化处理 18](#_Toc458608931)

[4.2.2 采集图像思路 18](#_Toc458608932)

[4.2.3图像处理思路 18](#_Toc458608933)

[4.2.4图像校正 19](#_Toc458608934)

[4.3小结 19](#_Toc458608935)

[第五章 特殊赛道识别及处理 20](#_Toc458608936)

[5.1 正常循迹 20](#_Toc458608937)

[5.2 障碍识别及处理 21](#_Toc458608938)

[5.2.1 障碍识别 21](#_Toc458608939)

[5.2.2 障碍处理 21](#_Toc458608940)

[5.3 起跑线识别及处理 22](#_Toc458608941)

[5.3.1 起跑线识别 22](#_Toc458608942)

[5.3.2 起跑线处理 22](#_Toc458608943)

[5.4 十字识别及处理 23](#_Toc458608944)

[5.4.1 十字识别 23](#_Toc458608945)

[5.4.2 十字处理 23](#_Toc458608946)

[第六章 控制策略研究 24](#_Toc458608947)

[6.1 PID控制 24](#_Toc458608948)

[6.1.1 PID算法简介 24](#_Toc458608949)

[6.1.2 PID参数整定 25](#_Toc458608950)

[6.2 舵机PD控制 26](#_Toc458608951)

[第七章 开发工具、制作、安装、调试过程 27](#_Toc458608952)

[7.1 调试过程 27](#_Toc458608953)

[7.2 调试上位机 27](#_Toc458608954)

[第八章 车模主要参数 28](#_Toc458608955)

[8.1 智能汽车外形参数 28](#_Toc458608956)

[8.2 智能汽车技术参数 28](#_Toc458608957)

[结 论 29](#_Toc458608958)

[参 考 文 献 30](#_Toc458608959)

[附 录 31](#_Toc458608960)

引 言

随着电子科技的不断发展，越来越多的智能设备设备开始进入到人们的生产生活中，嵌入式的迅猛发展为智能研究提供了更广阔的平台。在工业生产、科学探索、救灾抢险、军事等方面，人工智能发挥着越来越重要的作用，国外的几家著名跨国公司都推出了自己的无人驾驶汽车，因此，智能控制策略变得尤为重要。

“恩智浦”杯全国大学生杯智能汽车竞赛是国家教学质量与教学改革工程资助项目，以恩智浦半导体公司生产的8、16、32位单片机为核心控制模块，通过增加道路传感器、电机驱动电路、伺服系统以及编写相应程序，制作一个能够自主识别道路的汽车模型。该竞赛是涵盖了智能控制、模式识别、传感技术、汽车电子、电气、计算机、机械等多个学科的比赛，对学生的知识融合和实践能力的提高，具有良好的推动作用。

本文采用第十一届“恩智浦”杯全国大学生智能车竞赛的汽车模型作为研究平台，以32位单片机MK60DN512Z作为主控制单元，运用IAR Embedded Workbench软件作为开发工具进行智能控制策略研究。

道路信息检测模块普遍采用简单、速度快的数字类摄像头。本届车模后置双电机，因此需要对两个电机分别进行速度、方向检测，以及PID控制，车模转向采用S-D5官方指定伺服器。

电源模块设计中，采用美国芯源系统有限公司的开关稳压电源芯片MP1482作为稳压5V输出；采用美国芯源系统有限公司的开关稳压电源芯片MP1482稳压输出6V电压给舵机供电，采用TI公司TLV1117作为稳压3.3V输出，为单片机、摄像头、串口通信模块供电。

单片机软件算法部分为本文论述的重点，主要体现了智能车的智能控制策略，运用PID控制理论完成对车模速度的控制，采用PD控制理论完成对舵机伺服器的控制，沿用采集赛道特征信息循迹算法，对新型赛道的识别及处理。

本篇技术报告将从智能车机械结构、硬件电路、图像处理、控制算法等方面详细介绍整个准备过程。

第一章 方案设计

本章主要介绍智能汽车系统总体方案的选定和总体设计思路。

# 1.1 系统总体方案的选定

本届智能汽车比赛，我队为摄像头组别。在循迹传感器方面，选用数字摄像头ov7725，相比模拟摄像头有：硬件上有相对简单、信号传输稳定、抗干扰能力强等优势。由于采用了B型车模，车身在循迹前进的过程中，要根据赛道信息进行相应的转向、减速、停车等处理。测速方面，考虑到B车模本身的结构，我们采用了专门用于测速的霍尔模块，易于固定，且精确度较高，用于速度闭环反馈。由于B型车模对车体姿态的控制要求不高，但是由于存在坡道等赛道因素，我们采用了MPU6050六轴加速度计陀螺仪模块，用于检测车体信息，帮助车更加稳定地前进。

# 1.2 系统总体方案的设计

遵照本届竞赛规则规定，智能汽车系统采用恩智浦的32位微控制器MK60DN512Z单片机作为核心控制单元用于智能汽车系统的控制。数字摄像头采集赛道信息，返回到单片机作为转向控制的依据。霍尔测速模块采集速度信息 ，用于速度的闭环控制；加速度计陀螺仪模块返回的数字信号作为车身当前姿态的信号；主控输出PWM波控制电机的转速以保持车速度的控制和以及舵机转向的控制。由于赛道存在很多角度比较大的弯道，因此为了保证车在转向的过程中能平稳度过，弯道减速是非常必要的，保持必要的差速也是稳定车体的必要因素；同时，在直道上必要的加速能减少总体的时间。

根据以上系统方案设计，赛车共包括六大模块：MK60DN512Z主控模块、传感器模块、电源模块、电机和舵机驱动模块、速度检测模块和辅助调试模块。各模块的作用如下：

MK60DN512Z主控模块，作为整个智能汽车的“大脑”，将摄像头、陀螺仪，加速度计和霍尔测速等传感器的信号，根据控制算法做出控制决策，驱动两个直流电机以及舵机完成对智能汽车的控制；

传感器模块，是智能汽车的“眼睛”，可以调整好一定的前瞻，提前感知前方的赛道信息，为智能汽车的处理器做出决策提供必要的依据和充足的反应时间，使用霍尔测速用于控制车行进过程中速度的闭环控制，使用加速度计陀螺仪模块用以保持车模稳定行进；

电源模块，为整个系统提供电压合适而又稳定的电源；

电机驱动模块，驱动直流电机以实现智能汽车的速度输出；

伺服器模块，控制车体的行进方向；

辅助调试模块，主要用于智能汽车系统的功能调试、赛车状态监控等。

# 1.3 小结

本章重点分析了智能汽车系统总体方案的选择，并介绍了系统的总体设计和总体结构，简要地分析了系统各模块的作用。在今后的章节中，将对整个系统的各个模块进行详细介绍。

# 第二章 机械设计制作及调整

智能车机械部分设计主要包括制作和调整两部分内容，制作部分的内容主要是对车模没有的部分进行设计，包括传感器支架、电路板固定、防撞、测速轮安装等。调整部分则主要是针对智能车车模本身已经有的机械部分，在规则允许范围内进行调整，改装，提高其运动性能，以适应高速行驶和快速控制，这部分主要包括舵机改装、底盘调整、避震调整、四轮定位等。本章内容主要对电路板安装、传感器支架以及机械调校部分进行介绍。

## 2.1 机械结构制作部分设计

由于大赛提供车模本身是运动型模型车通用车模，并没有提供专门为智能车安装电路、传感器等电路部分的部件，因此这部分机械结构需要自行设计制作并安装。制作部分主要原则为：轻、牢、简。

### 2.1.1摄像头的安装

为了降低整车重心，需要严格控制摄像头的安装位置和重量，我们采用3D打印技术自行设计了轻巧的夹持组件,并采用了碳纤维杆作为安装摄像头的主桅，这样可以获得最大的刚度质量比，整套装置具有很高的定位精度和刚度，使摄像头镜头便于拆卸和维修。摄像头镜头的安装如图2.1所示。



图2.1 摄像头的安装

### 2.1.2车模倾角传感器的安装

车模倾角传感器包括陀螺仪和加速度计。它们都是贴片元器件，单独固定在一块小电路板上，然后用铝架固定在车模上，从而保证检测数据的可靠性。

### 2.1.3测速传感器的安装

我们采用的测速传感器是霍尔测速模块，小巧轻盈，直接固定在电机齿轮外侧。

### 2.1.4电路板的安装

为了使小车具有较好的稳定性及转向性能，我们在搭建小车时尽量选择降低重心，因此我们选择将主控板和驱动板分别安装到车底盘的中部靠前的位置以及车的后方位置，从而实现降低重心，提高小车的稳定性。

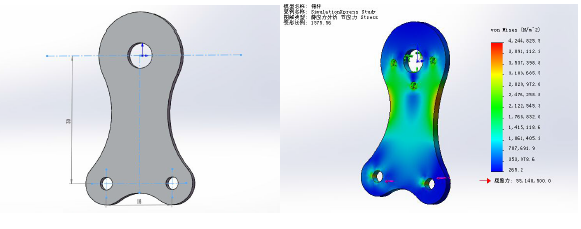
### 2.1.5电池安放

同样为实现降低重心，提高小车稳定性的目的，我们制作了高度很低的电池固定底座来达到以很低的高度牢固地固定电池的目的，最大程度的降低了小车的重心。

### 2.1.6 舵机力臂的调整

舵机的灵活转向控制则依赖于机械系统中各个零部件间协调运行。为提高智能车的整体协调性能，一定要把握好“车身简捷、底盘低稳、转向灵活、协调匹配”的设计与安装原则。舵机的输出转角通过连杆传动控制前轮转向，其转角精度直接影响到智能车模能否准确按赛道路线行驶，由于舵机从执行转动指令到响应输出需占用一定的时间，因而产生舵机实时控制的滞后。虽然车模在进入弯道时能够检测到黑色路线的偏转方向，但由于舵机的滞后性，极大限制车模在连续弯道上行驶的最大时速，为了减小舵机响应时间，我们利用杠杆原理，采用加长舵机力臂的方案来弥补这一缺陷。

我们通过运用Solidworks软件建模进行了一系列的运动仿真，通过仿真，最后确定了采用合适的舵机力臂，使舵机转弯灵活，轻盈，同时又能保证不对舵机损坏。

图2.2 舵机力臂的设计以及力学性能仿真

## 2.2 车模机械结构的调整与改装

车模本身的机械结构是通用结构，并不适合智能车竞赛的要求，因此要对这些部分进行改装，另外，为了提高车模的运动性能，对一些机械结构还需要调整，比如车轮前束等。这部分着重介绍舵机改装、底盘等部分的调整和改装。

### 2.2.1 舵机改装

为了提高舵机反应速度，在相同转角下，有尽可能大的线行程，因此需要延长舵机臂。另一方面，由于舵机扭矩和转角精度的限制，不能无限制延长舵机臂，这样就确定了舵机臂的长度，尺寸约为24mm×38mm。图2.3是舵机安装实物图，四个螺丝（上下各两个）将舵机牢固安装在支架上。



图2.3 舵机改装

### 2.2.2 底盘改装

由于赛道特性，底盘改装目标是尽可能低(能保证通过坡道)，这样可以最大程度保证智能车行驶的稳定性。因此我们降低了底盘高度。另外，由于赛道整体属于平坦路面，没有较大较多的颠簸，所以我们把后避震拆除，并紧固后桥连接件。

### 2.2.3 前轮定位

B型车模前轮可以调整的角度有主销前倾、内倾、前束等，这些角度的调整根据每个车的机械性能不同而不同调整，我们的智能车由于重心位置在中心偏后，因此前轮压力较小，转向负担不大，因此为了增加抓地力和稳定性，选择了主销内倾和负前束的调整。另外，由于车模本身的精度限制，这部分角度的调整并不是主要的，仅仅是为了避免负面影响以及修正车模本身的不对称和不平衡问题。图2.4所示



图2.4 前轮定位

主销内倾角是指在横向平面内主销轴线与地面垂直线之间的夹角，它的作用也是使前轮自动回正。角度越大前轮自动回正的作用就越强，但转向时也就越费力，轮胎磨损增大；反之，角度越小前轮自动回正的作用就越弱。通常汽车的主销内倾角不大于8°。

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，所以在调整时可以近似调整为0°3°左右，不宜太大。

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。

### 2.2.4车轮外倾角

前轮外倾角是指通过车轮中心的汽车横向平面与车轮平面的交线与地面垂线之间的夹角，对汽车的转向性能有直接影响，它的作用是提高前轮的转向安全性和转向操纵的轻便性。在汽车的横向平面内，轮胎呈“八”字型时称为“负外倾”，而呈现“V”字形张开时称为正外倾。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正外倾角度，一般这个角度约在1°左右，以减少承载轴承负荷，增加零件使用寿命，提高汽车的安全性能。

模型车提供了专门的外倾角调整配件，近似调节其外倾角。由于竞赛中模型主要用于竞速，所以要求尽量减轻重量，其底盘和前桥上承受的载荷不大，所以外倾角调整为0°即可，并且要与前轮前束匹配。

### 2.2.5 前轮前束

所谓前束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。像内八字那样前端小后端大的称为“前束”，反之则称为“后束”或“负前束”。在实际的汽车中，一般前束为012mm。

在模型车中，前轮前束是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，我们发现较小的前束，约束02mm可以减小转向阻力，使模型车转向更为轻便，但实际效果不是十分明显。

虽然模型车的主销后倾角、主销内倾角、车轮外倾角和前束等均可以调整，但是由于车模加工和制造精度的问题，在通用的规律中还存在着不少的偶然性，一切是实际调整的效果为准。

## 2.3智能车转向机构调整优化

理想的转向模型，是指在轮胎不打滑时，忽略左右两侧轮胎由于受力不均产生的变形，忽略轮胎受重力影响下的变形时车辆的的转向建模。在这种理想的模型下，车体的转向半径可以计算得到。

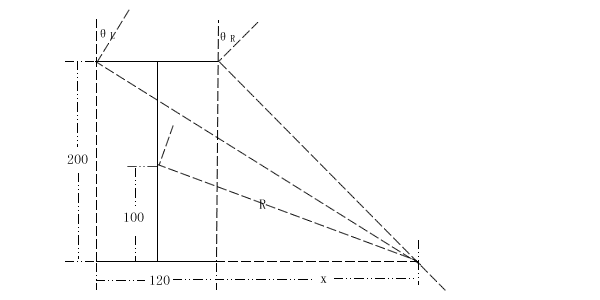


图2.5 智能车转向示意图

如图2.5，假设智能车系统为理想的转向模型，且其重心位于其几何中心。车轮满足转向原理，左右轮的轴线与后轮轴线这三条直线必然交于一点。

转向机构在车辆运行过程中有着非常重要的作用。合适的前桥和转向机构可以保证在车辆直线行驶过程中不会跑偏，能保证车辆行驶的方向稳定性；而在车辆转向时，合适的转向机构可以使车辆自行回到直线行驶状态，具有好的回正性。正是由于这些原因，转向系统优化设计成为智能车设计中机械结构部分的重点，直接关系到赛车能否顺利地完成比赛。在实际操作中，我们通过理论计算的方案进行优化，然后做出实际结构以验证理论数据，并在实际调试过程中不断改进。

在模型车制做过程中，赛车的转向是通过舵机带动左右横拉杆来实现的。转向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。经过最后的实际的参数设计计算，最后得出一套可以稳定、高效工作的参数及机构。

## 2.4智能车重心位置的调整

如果车重心特别靠前，后轮正压力不足会导致甩尾。为了使重心后移，我们尝试了很多搭建方式，使得保证结构稳定的前提下尽量减轻重量。同时，我们把舵机和电池均往后移，达到了预期的效果。

## 2.5其它机械结构的调整

另外，在模型车的机械结构方面还有很多可以改进的地方，比如说车轮、悬架、底盘、车身高度等。

模型车在高速的条件下（2.3m/s-3.5m/s），由于快速变化的加减速过程，使得模型车的轮胎与轮辋之间很容易发生相对位移，可能导致在加速时会损失部分驱动力。在实验中调试表明，赛车在高速下每跑完一圈，轮胎与轮辋之间通常会产生几个厘米的相对位移，严重影响了赛车的加速过程。为了解决这个问题，我们在实际调试过程中对车轮进行了粘胎处理，可以有效地防止由于轮胎与轮辋错位而引起的驱动力损失的情况。

此外，我们还对车身高度，以及底盘的形状和质量等，都进行了相应的改进和调整，均取得了不错效果。

## 2.6小结

本次的摄像头组使用了B车模，由于数字摄像头的使用以及车模的特点，车速普遍都比较快，因此，为了车在跑的过程中保证平稳性，机械结构的搭建变的非常重要，车的性能与机械结构有着非常密切的联系。良好的机械结构是模型车提高速度的关键基础。在同等的控制环境下，机械机构的好坏对其速度的影响十分显著。我们非常重视对智能汽车的机械结构的改进，经过大量的理论研究和实践，我们小车的大部分质量都集中在车的中间部位；对舵机结构的优化，使得车在转弯的时候变的更加平滑与稳定，经过坡道的时候有效地避免了车体过于剧烈的抖动，甚至冲出跑道。

对车轮胎的调整，使得车在行驶的过程中，保证了充足的摩擦力，为更快的速度提供了硬性的保障。

一个好的机械结构能够保障车平稳行驶的过程中获得最大的动力学性能，是进一步提速的关键。

第三章 硬件电路方案设计

本系统的硬件电路采用模块化设计方式。主要包括单片机最小系统模块、电源模块、图像采集模块、测速模块、串口通信模块、红外检测、显示模块等部分。

# 3.1 单片机最小系统模块

本设计的核心控制器为恩智浦公司生产的32位单片机MK60DN512Z。该单片机具有144个引脚。

MK60DN512Z具有丰富的系统资源和方便的外部电路接口，其中包括32位中央处理单元，UART模块，PIT定时中断模块，IIC模块，DMA模块，RAM存储器， FLASH存储器，EEPROM存储器，FTM PWM 模块等。系统模块电路如图3.1所示。

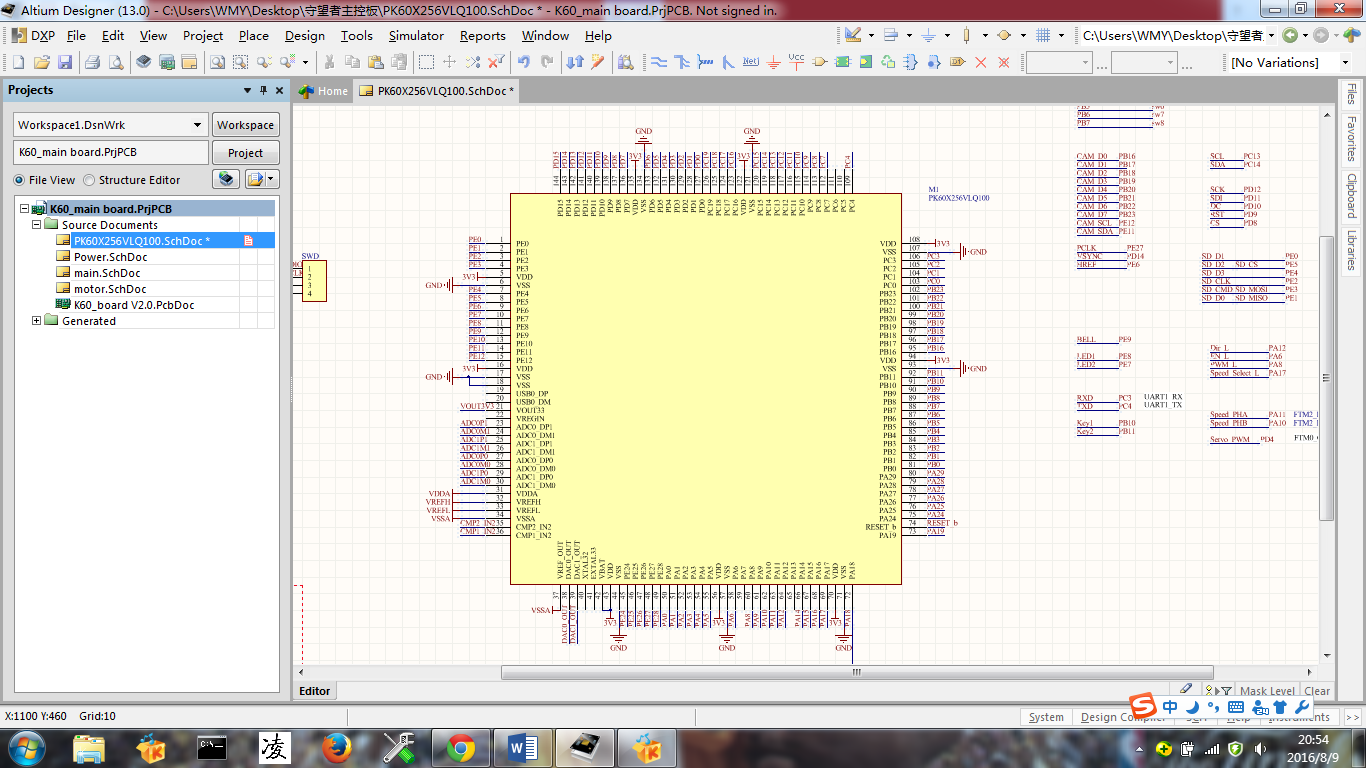


图3.1 单片机芯片部分原理图

# 3.2 电源模块

## 3.2.1 电池使用

电源模块是系统稳定工作的基础，因此，电源模块输出电压和电流的稳定性在整个智能车系统中起着非常重要的作用。

智能车的电池为Ni-Cd(镍镉)电池，该类型电池具有高效的利用率和稳定的性能，一直被作为各种航模、电动车等的供电设备。为了获得最高的性能和最长的寿命，该充电电池必须以正确的方法来使用。

对镍镉电池充电时，通常采用电池容量值的大约两倍来充电，当电池是1800mAh，那么用0.5到1.5安培来充电是安全的，而且充得比较饱满。通常电流高，电池的爆发力会强些，但未必如电流低时饱满。同时我们也必须注意充电电流不能过高，最好不要超过1.5A，当电流过高时，不仅不能提高电池性能，反而会损坏电池，严重时会导致电池起火、爆炸。根据经验，一般在0.5到1.5安培之间是效果较好。

电池充满电时，电压大约为8V。在电池压小于7V时，应注意及时充电，电池过放会对其造成不可逆转的损害，电压低于6V会对电池造成毁灭性伤害。

## 3.2.2 稳6V电源模块

由于舵机伺服器S-D5的额定工作电压为6V，MP1482系列是美国芯源系统有限公司生产的2A电流输出降压开关型集成稳压芯片，它内含固定频率振荡器（340KHZ）和基准稳压器（0.92v），并具有完善的保护电路、电流限制、热关断电路等。利用该器件只需极少的外围器件便可构成高效稳压电路。电路的设计原理如图3.2所示。

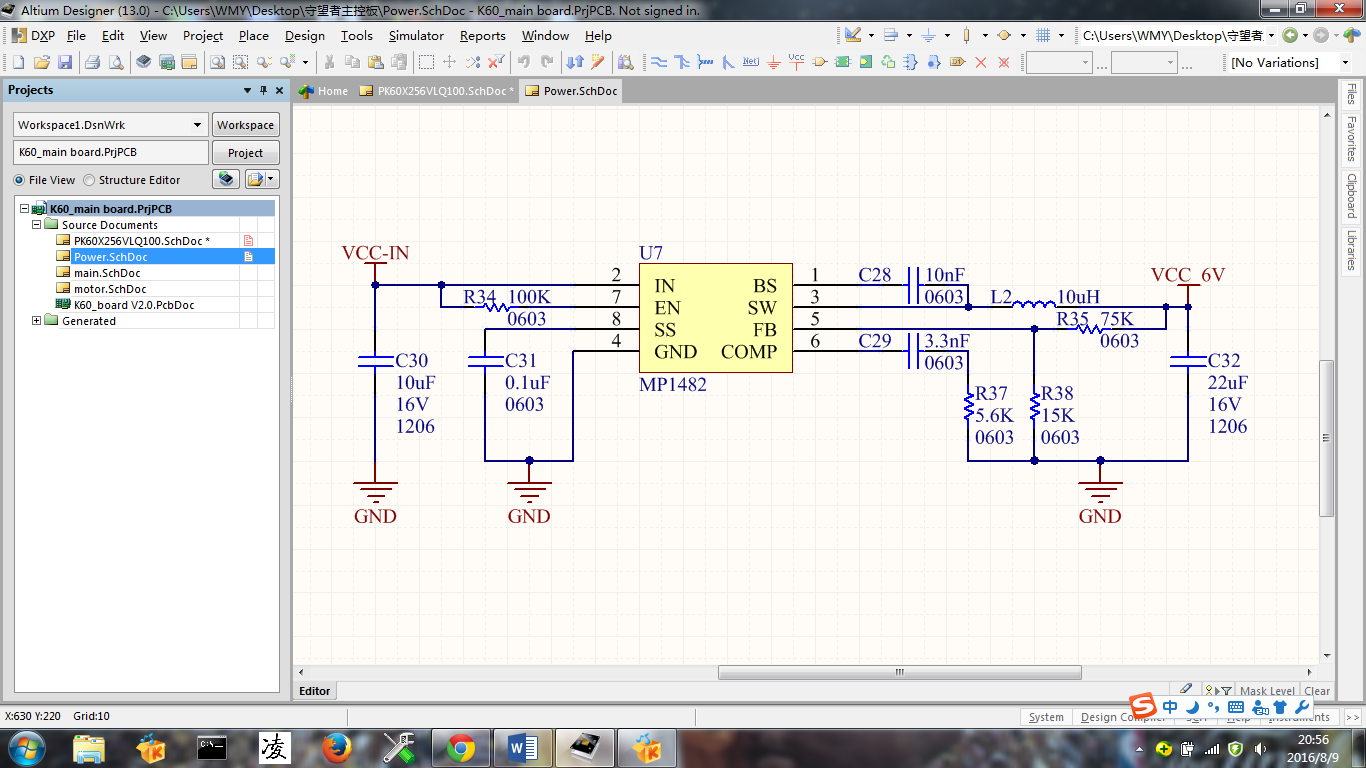


图3.2 舵机供电电路原理图

## 3.2.3 稳5V电源模块

由于系统串口通信等地方需要5V电源，同时为了更好得获得稳定的3.3V电源给MCU提供稳定的电源，我们仍然采用美国芯源系统有限公司生产的MP1482来达到稳定5V的目的。电路原理图如下：

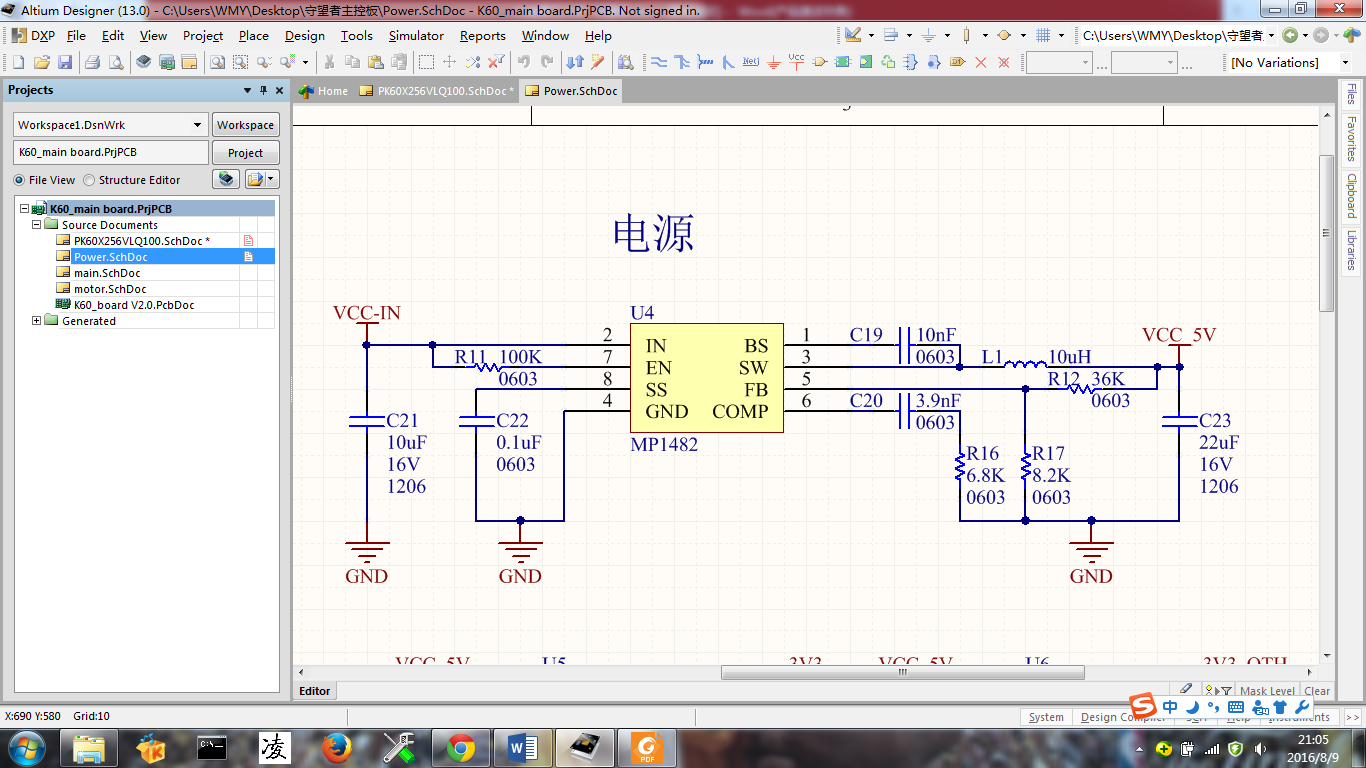


图3.3 稳5V电路原理图

## 3.2.4稳3.3V电源模块

由于单片机的额定电压为3.3V，摄像头供电3.3V，测速供电也为3.3V，因此，将电压稳定在3.3V并给各模块供电是必不可少的。电路的设计原理如图3.4所示。

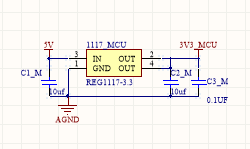


图3.4 稳3.3V电路原理图

# 3.3 电机驱动模块

本智能车系统车模的电机型号较大，对电机驱动的输出电流的要求较苛刻，因此本设计的驱动电路由2片IR2184S构成H桥。通过控制4个MOS管的导通和关断来实现正反转，并通过控制输入的PWM波的占空比来调节电机两端的平均电压，达到控制电机的转速的目的，具体电路图如图3.5所示。

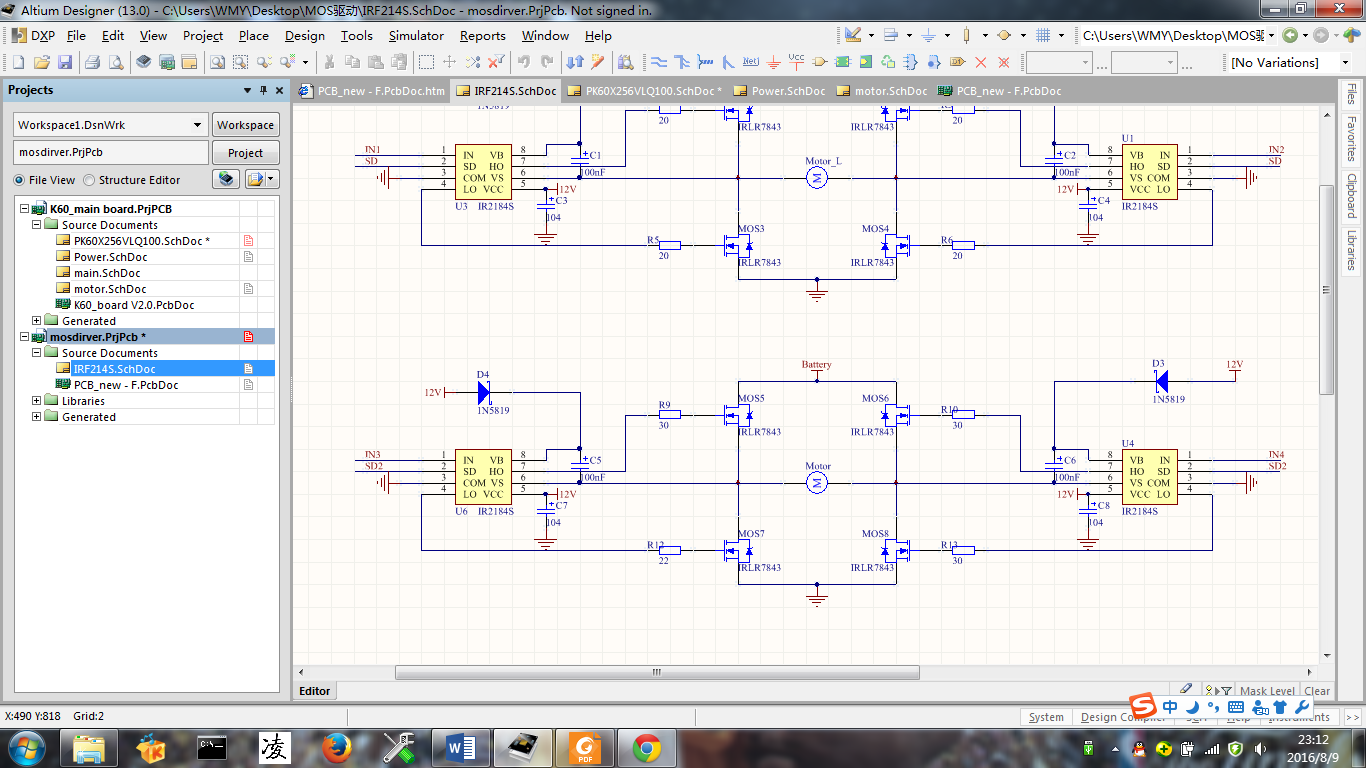


图3.5 电机驱动原理图

# 3.4 串口通信模块

调车的过程中，需要实时的观察变量变化情况，通过下载线在Live Watch 窗口中观察变量很受局限，所以采用蓝牙无线通信模块，波特率为115200，传输速度更快，配合上位机观察超调量，有助于调试PID参数，同时，也可以通过蓝牙发送赛道信息，便于对赛道元素的分析，具体电路图如图3.6所示。

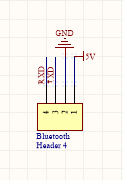


图3.6 串口通信原理图

# 3.5 霍尔测速模块

我们采用的霍尔测速模块，霍尔模块具有电路简单，测量精度高，体积小巧，安装简便的特点，电路的设计原理如图所示。

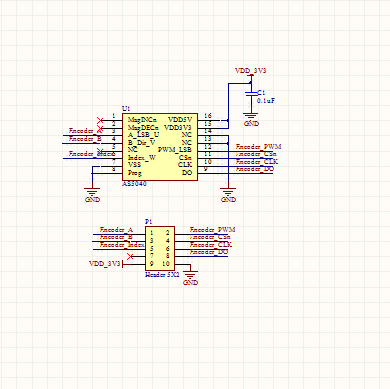


图3.7 霍尔测速电路原理图

第四章 图像信息处理

智能车采集图像信息的底层处理算法是整个上层控制策略的基础，图像采集的稳定性、图像处理方法、路径识别的准确性都决定着上层控制策略能否发挥作用，只有准确的识别出路径信息，智能车才能实现高速稳定行驶。我们使用ov7725数字摄像头来进行图像处理和识别。

# 4.1 图像采集

## 4.1.1 图像数据输出简介

摄像头的主要工作原理是[8]：将摄像头的像素点从上到下，从左到右逐个输出，虽然部分芯片可以配置输出顺序，但是对于智能车来说，正常输出即可，有些摄像头将一帧图像分成奇偶两场，ov7725并没有这项配置。

## 4.1.2 摄像头工作时序

## 4.1.2.1 行中断时序

图4.1 行中断时序图

首先需要明确的是HREF和HSYNC均为行中断信号，它们共用一个管脚，由寄存器配置使用选择什么信号输出。然而，它们的时序还存在着一定的区别，HREF在上升沿就会马上输出图像数据，而HSYNC则需要等待一段时间再输出，如果行中断里需要处理事情再开始采集，则显然用HREF的上升沿是很容易来不及采集第一个像素。

PCLK是整个控制摄像头指令的时序，在上升沿时MCU会采集图像，下降沿时摄像头输出图像。

行消隐区：如果不按照时序来采集，便会采集到值为0的消隐区，即黑色。行与行之间，场与场之间，都是存在消隐区的。里面存放的均为无效数据。

## 4.1.2.2 场中断时序

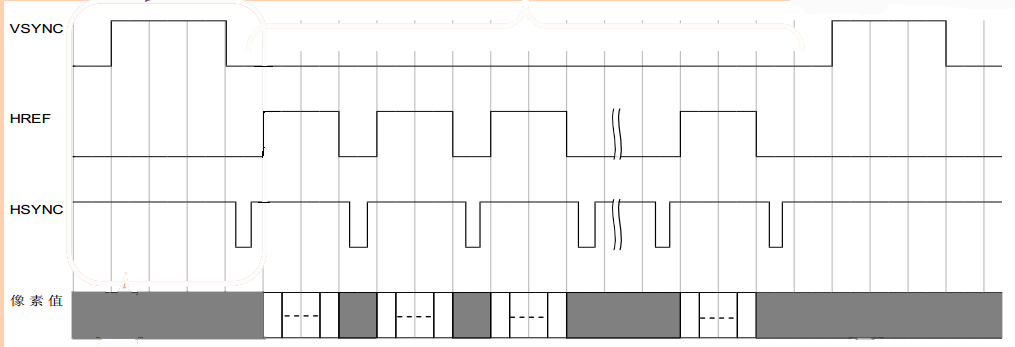


图4.2 场中断时序图

场中断信号VSYNC来了，便可以开始采集每帧的数据，可以在程序里选择上升沿或者下降沿触发中断。场中断和行中断的时序关系必须处理好，否则将出现异常。

## 4.1.3 DMA简要介绍

DMA(Direct Memory Access，直接内存存取) 允许不同速度的硬件装置来沟通，而不需要依赖于 CPU 的大量中断负载。否则，CPU 需要从来源把每一片段的资料复制到[暂存器](http://baike.baidu.com/view/238434.htm)，然后把它们再次写回到新的地方。在这个时间中，CPU 对于其他的工作来说就无法使用。

工作原理：DMA 传输将数据从一个地址空间复制到另外一个地址空间。当CPU 初始化这个传输动作，传输动作本身是由 DMA 控制器来实行和完成。典型的例子就是移动一个外部内存的区块到芯片内部更快的内存区。像是这样的操作并没有让[处理器](http://baike.baidu.com/view/50152.htm)工作拖延，反而可以被重新排程去处理其他的工作。DMA 传输对于高效能嵌入式系统算法和网络是很重要的。

在实现DMA传输时，是由DMA控制器直接掌管总线，因此，存在着一个总线控制权转移问题。即DMA传输前，CPU要把总线控制权交给DMA控制器，而在结束DMA传输后，DMA控制器应立即把总线控制权再交回给CPU。一个完整的DMA传输过程必须经过DMA请求、DMA响应、DMA传输、DMA结束4个步骤。

# 4.2 图像处理

由于图像处理是路径识别的基础，也是控制策略能否发挥作用的重要影响因素，因此，本文将对此部分进行详细介绍。

## 4.2.1 阈值的确定方法——硬件二值化处理

ov7725数字摄像头在硬件上做了二值化处理，所以阈值作为程序底层中一个十六进制的数值来体现，这里就不再赘述了。需要注意的是，如果想要获得效果明显清晰的图像，当环境改变时，阈值要做相应的调整来适应光线场地，这个是运行上层采线函数获得正确赛道信息的首要条件。

## 4.2.2 采集图像思路

图像采集的方案是不使用行中断信号，直接DMA模块计数来完成一场结束的判断。

1、需要采集图像时，开场中断

2、场中断来了，初始化 DMA 传输，并启动 DMA 传输

3、每个 PCLK 上升沿来了都触发 DMA 传输，把摄像头输出的值读取到内存数组里。当触发 n 次（n=图像像素数目）后就停止 DMA 传输。

4、DMA 停止传输时触发中断，中断里关闭场中断，图像采集完毕。

## 4.2.3图像处理思路

图像处理采用了赛道边沿提取的方法，基本思想如下：

1、逐行扫描原始图像，根据设定的阈值提取黑白跳变点；

2、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、利用赛道的连续性，根据上一行白块的位置和边沿的位置来确定本行的边沿点；

4、在弯道的时候赛道有回拐的情况，所以提取本行边沿之前还要向前搜索回拐的边沿点；

5、在弯道行进过程中，保持系统搜索最远有效行，为未来进一步打角做出判断，尤其是在速度比较快的时候，获取较远赛道信息变的非常重要；

## 4.2.4图像校正

由于摄像头是固定好的，因此，在制作机械结构的开始，就应该考虑到摄像头的前瞻等因素；初次之外，考虑到摄像头镜头的问题，摄像头采集到的图像存在失真的现象，摄像头拍照时为获取较大的视角势必要加入广角,然而加入广角便不可避免的发生图像桶形畸变,即使抓拍的图像产生鱼眼状中间凸起的形变,而这必将影响下一步的图像处理操作,为了准确的还原原始图像则需要利用各种技术对图像桶形畸变进行校正。 我们利用了数字图像处理技术对对拍摄到的含有畸变的图像进行有效校正获得了一定的效果。

## 4.3小结

对于摄像头组别来说，摄像头采集的图像是整个环节中最重要的一环，因此，保证了图像的稳定，才能让车判断出正确的赛道，并且到达一定的速度。得到正确的图像之后，如何巧妙的处理图像数据也是让车提速的关键，在处理图像数据的时候，硬件二值化以及图像的校正是非常重要的。对于微处理器来说，摄像头采集的数据量是比较大的，因此，恩智浦微处理器的DMA通道，提供了合适的解决方案。

第五章 特殊赛道识别及处理

本届比赛赛道类型主要为“十字”、“小S”形弯道、“楔形”障碍、起跑线，赛道较第十届难度有显著降低。所以，经考虑，我队在正常循迹的基础上着重对障碍、起跑线、十字进行图像识别，并采取正确的行驶策略。

# 5.1 正常循迹

在正常的赛道上时，尽管有弯道等特点，但是赛道仍然有明显的双边特征，而且背景和赛道区别特征明显，这就为摄像头辨别赛道提供了简便可靠的依据。

在循迹上，我们采用了多行加权平均判断，通过对摄像头采集的每一帧画面上进行多行加权优化，提取出最优化路径的赛道特征。

根据往届比赛以及本届东北赛的经验，赛车能否以最短的时间完成比赛，与赛车的速度和路径都有着密切的关系，因此，如何使赛车以一个最合理、最高效的路径完成比赛是提高平均速度的关键。

对于赛车路径的优化，我们从两方面来完成：

1）增加视场的长度和宽度

根据我们的分析，当赛车采集到的图像能够覆盖一个比较完整的S弯道时，通过加权算法计算出来的中心就会处于视场中央附近，此时赛车会以一个比较好的路径快速通过S弯道；相反，如果视场无法覆盖一个完整的S弯道，赛车就会误处理为普通的单向弯道，这样赛车的速度就会大大减慢。因此，尽量增大视场的长度和宽度就很有必要了。

2）优化加权算法

对整场有效行的中心求加权平均值的算法，在低速情况下可以有效地优化赛车路径，但在赛车速度提高到一定程度之后由于过弯时的侧滑，甚至侧翻，路径不是很好。而由于图像分布不均，三分之二的行分布于车体前方40cm的范围内，求出的加权平均值受车体近处的图像影响较大，相当于摄像头的前瞻不足。因此整场图像求加权的算法对于高速情况下的路径优化效果不是很明显。

为了解决这个问题，我们对于参与加权计算的图像行数及权重进行了处理，减小了车体前部40cm范围内的图像参与加权的行数和权重，同时增大视场前部图像的权重。在经过长期调试之后，得到了一套比较合适的参数，能够有效优化高速情况下的赛车路径。

# 5.2 障碍识别及处理

## 5.2.1 障碍识别

针对障碍，最开始采用识别其本身的方法，因为障碍颜色为黑色，在白色赛道上出现一块黑色区域，此为障碍，虽然表面上没有什么问题，但实际上存在许多误判，特别是在转弯时，会将蓝色底的部分误判为障碍，最终不得不放弃此方案，尝试其他方法。结合车模本身循迹方法，经过观察发现，障碍有如下有效信息：

1、障碍存在的地方，赛道宽度会大幅度减少

2、障碍存在的地方，中线偏移很大，且边的斜率会较直道部分斜率小

基于以上3条特点，加上范围阈值的实际调试，障碍的识别成功，误判率很低。

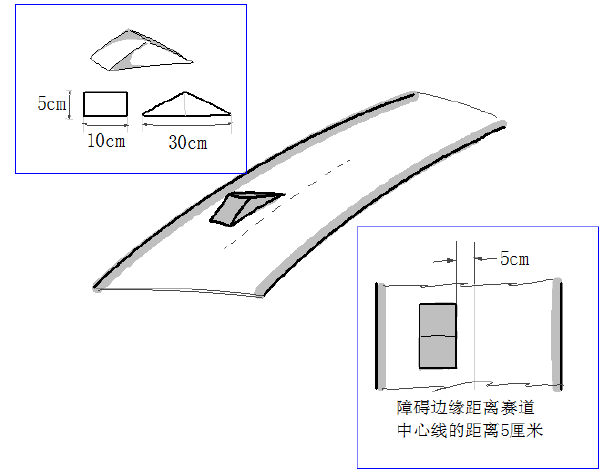


图5.2 障碍示意图

## 5.2.2 障碍处理

我队的车在正常循迹时会沿着拟合好的中线行驶，也就是说，在PID参数保持不变时，中线的位置决定了车模的轨迹。基于以上事实，我队在识别障碍后，在中线的位置添加了向左或向右的偏移量（根据障碍的位置而定），并延时打角，如此车模便会顺利绕过障碍。

# 5.3 起跑线识别及处理

## 5.3.1 起跑线识别

因为比赛规定要求，若车模无法在完成比赛后未能在通过起跑线后停住，则成绩加1秒，所以起跑线的识别变得尤为重要，直接决定车的档次。

结合车模本身循迹方法，经过观察发现，起跑线有如下有效信息：

1、起跑线前后1米内为直道

2、起跑线两道黑条长度有严格规定10厘米

3、两道黑条间距有严格规定9厘米

4、起跑线关于赛道成轴对称。

基于以上4条特点，加上范围阈值的实际调试，起跑线识别成功，误判率很低。

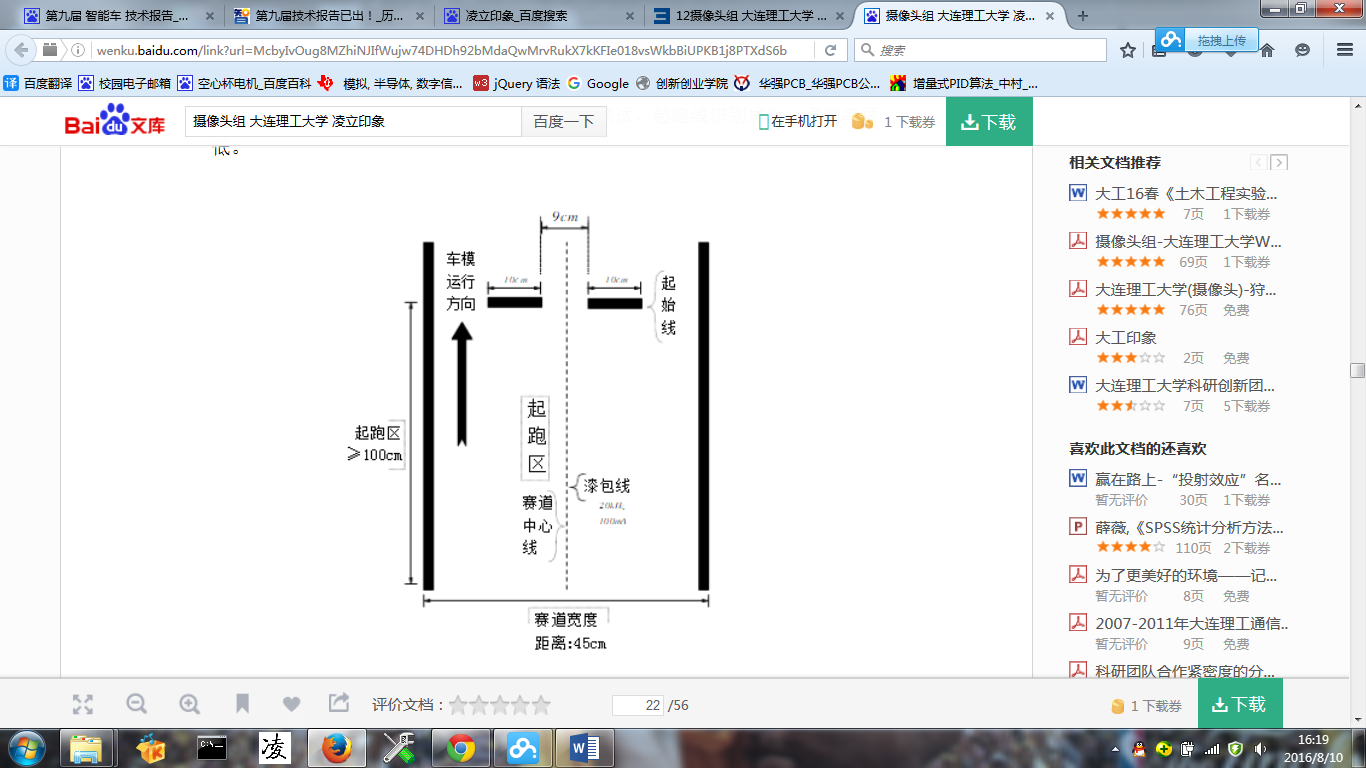


图5.3 起跑线示意图

## 5.3.2 起跑线处理

根据规定，要求车模通过起跑线后3米内停住，我队采用识别起跑线后，控制速度递减为零的方法，使车体能及时停住,并且减小了MOS反向输出突变对电机驱动的伤害。

# 5.4 十字识别及处理

## 5.4.1 十字识别

因为往届已经存在了十字路口，所以在算法识别上并无太多新意，十字的特点也特别明显、简单，就是在摄像头采集的图像中，存在多行全白，也可以说是没有黑边。

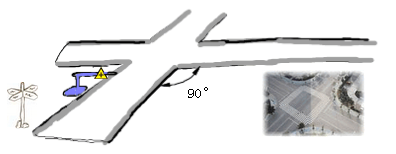


图5.4 十字示意图

## 5.4.2 十字处理

检测到十字，将中线设定为整幅图像的中间列，采取直走的方法。

第六章 控制策略研究

# 6.1 PID控制

## 6.1.1 PID算法简介

PID控制即比例、积分、微分控制，该方法在工程实际中应用相当广泛。PID控制算法具有结构简单、工作可靠、便于调整、性能稳定等优点。当被控对象的性能、结构等一系列参数无法得知，而且建模等控制方法亦无从下手时，我们便可以应用PID控制技术，但是，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。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控制器就是根据系统的误差，利用比例、积分、微分计算出控制量进行控制的。图6.1为PID控制系统原理图。



图6.1 PID控制系统原理图

PID函数的一般形式为：U(s)=kp(1+1/(TI\*s)+TD\*s)；其中kp为比例系数，TI为积分时间，TD为微分时间常数。

比例控制是一种简单实用的控制方式。该方法的输出与输入误差成比例关系。比例系数越大，调节作用亦越大，减少误差的过程则越迅速，但是比例系数如果过大，则系统的稳定性会大大下降。

对于积分控制，控制器的输出与系统的输入误差信号的积分成正比关系。积分项的引入是为了消除稳态误差，积分项对误差取决于时间的积分，随着时间的增加，积分项会增大。这样，即便误差很小，积分项也会随着时间的增加而加大，它推动控制器的输出增大使稳态误差进一步减小，直到等于零。

微分控制中，控制方法的输出与输入误差的变化率成正比关系。智能车控制系统由于存在有较大惯性，其变化总是落后于误差的变化自动控制系统，在克服误差的调节过程中可能会出现振荡，严重时会导致不稳定。为了充分解决此问题，我们需要提前抑制误差的变化作用，即在误差接近零时，抑制误差的作用就应该是零。这就是说，在控制器中仅引入“比例”项往往是不够的，比例项的作用仅是放大误差的幅值，而目前需要增加的是“微分项”，它能预测误差变化的趋势，这样，具有比例+微分的控制器，就能够提前使抑制误差的控制作用等于零，甚至为负值，从而避免了被控量的严重超调。

## 6.1.2 PID参数整定

PID控制算法中PID参数的整定是控制系统设计的核心内容。该过程是根据被控过程的特性确定PID控制器的比例系数、积分时间和微分时间的大小。

PID控制器参数整定的方法很多，概括起来有两大类：一是理论计算整定。该方法需要依靠一定的数学模型，即通过一定的模型进行理论计算，最后确定控制器参数。但是，理论计算的方法所得到的数据常常是不能够直接运用到实际中的，其必须通过实际工程进行验证，并进行调整和进一步修改。二是工程经验整定。该方法主要依赖工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。工程整定方法中的PID参数整定主要有临界比例法、衰减法、反应曲线法。三种方法各有其特点，每一种方法所得到的控制器参数，都需要在实际运行中进行最后调整与完善。此外，其共同点都是通过试验，然后按照工程经验公式对控制器参数进行整定。

进行 PID控制器参数的整定步骤一般如下：

1、预选择一个足够短的采样周期让系统工作；

2、仅加入比例控制环节，直到系统对输入的阶跃响应出现临界振荡，记下这时的比例放大系数和临界振荡周期；

3、在一定的控制度下通过公式计算得到PID控制器的参数；

4、根据实际运行情况对计算出的PID控制器的参数进行调整。

本系统的PID参数为通过上位机调试得来，具体调试方法为：先将PID参数设置为经典参数，然后通过上位机观察速度曲线，不断改变PID参数，直至观察速度曲线发现其加减速时间很短，超调量很少，则说明此时的PID参数已经基本比较合适，这样就确定出适合本系统的一组PID参数。

# 6.2 舵机PD控制

对于车体的转向控制，我队通过PD控制舵机的打角，实现弯道的过度，但是考虑到整个赛道的弯道曲率是有差异的，因此，我们选择用动态PD来控制舵机的打角，同时，根据赛道中心线来确定动态PD的范围，利用多次测量和用MATLAB拟合的方法，确定了控制舵机PD参数的函数关系，因此在过比较大的弯道的时候，车子也能有比较好的路径。

第七章 开发工具、制作、安装、调试过程

# 7.1 调试过程

我们用的是IAR编译下载软件，IAR Systems是全球领先的嵌入式系统开发工具和服务的供应商。公司成立于1983年，提供的产品和服务涉及到嵌入式系统的设计、开发和测试的每一个阶段，通过IAR编译软件强大的在线调试功能，可以得到大量的信息，为智能汽车的调试提供了很大的帮助。在智能汽车的调试过程中，有针对性的开发一个便于人机交互的上位机系统，通过简单明了的可视化界面直观的显示智能汽车的状态对调试有很大帮助。我们开发了用于监测智能汽实时状态的实时监测系统，大大提高了调试效率。

# 7.2 调试上位机

基于摄像头图像数据的复杂性，我们采用了往届学长自己编写的上位机软件，该上位机软件，包括了画曲线，显示摄像头图像等等功能，给我们调车模参数提供了很大的便利；下图是该软件部分界面：

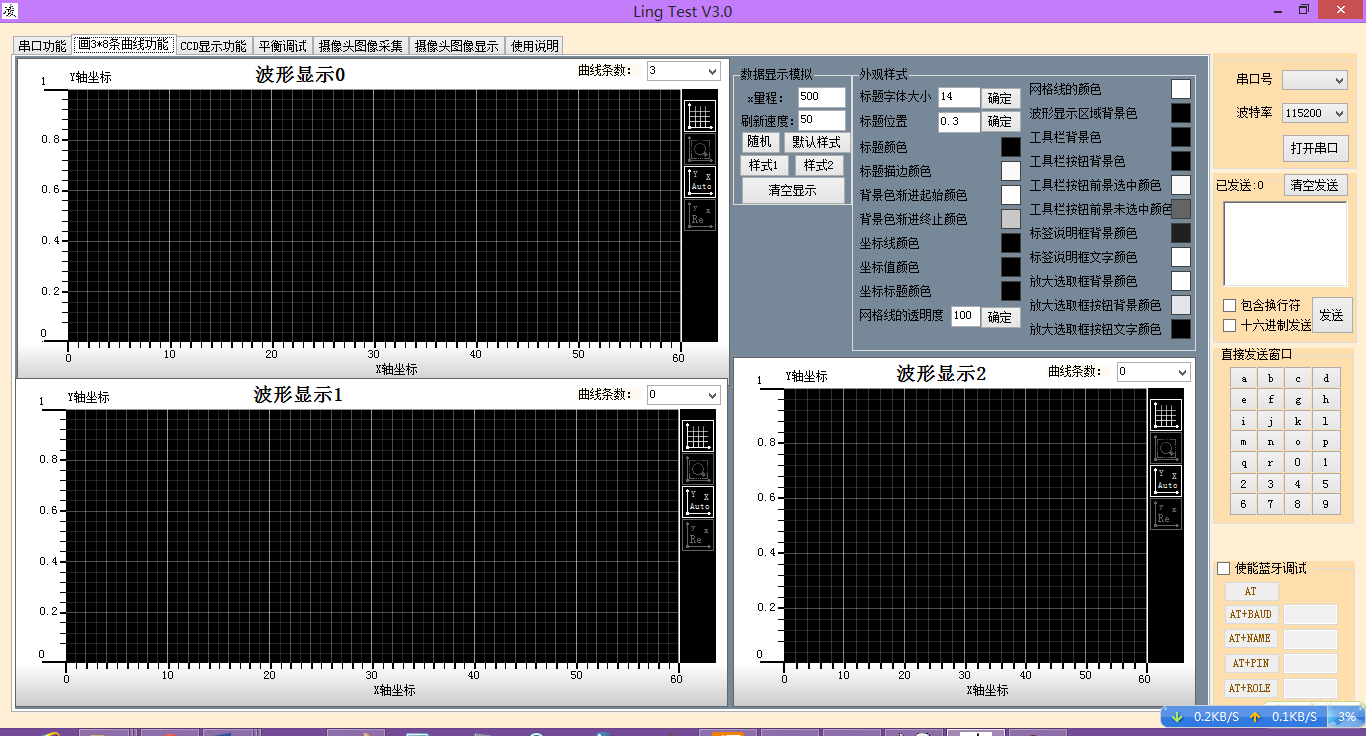
1.画图功能

图7.1 画图功能界面

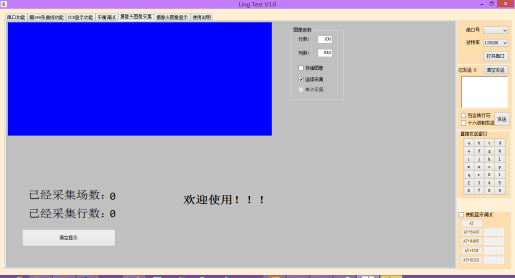
2. 摄像头图像采集功能

图7.2 摄像头图像采集功能界面

第八章 车模主要参数

# 8.1 智能汽车外形参数

经过改装后，智能汽车的外形参数为：

车长：280mm；车宽：250mm；车高：320mm；车重：1.3kg

# 8.2 智能汽车技术参数

智能汽车相关技术参数如表8.1所示：

表8.1 智能汽车技术参数

|  |  |
| --- | --- |
| 项目 | 参数 |
| 车模平均电流（匀速行驶）(毫安) | 300 |
| 电路电容总量（微法） | 800 |
| 传感器种类及个数 | 摄像头×1  霍尔传感器×1 |
| 新增加伺服电机个数 | 1 |
| 赛道信息检测空间精度（毫米） | 10 |
| 赛道信息检测频率（次/秒） | 150 |
| 主要集成电路种类/数量 | 2/3 |

**结 论**

在这份竞赛报告中，我们主要从智能车机械结构、硬件电路、图像处理、控制算法等方面介绍准备比赛的整个过程，大体上还是继承了前辈们的精髓，同时也针对新的比赛规则研究出应对策略，创新之处大体归为以下三点：

1、摄像头的选取：我队首次采用ov7725摄像头，且在硬件方面进行了二值化处理，只需通过软件调整阈值变量，大大降低了编程人员的工作量，可以将更多精力放在上层图像处理的程序上，而不是底层代码。主要精力放在了顶层设计上，这也是符合恩智浦杯智能汽车大赛的精神的。

2、特殊赛道识别：因为前人的循迹程序已基本完备，我们只是拿来稍微改动，并无太多创新、改良的地方，更多的是参数的调整；但是，每届比赛的规则都会发生变化，这些地方需要我们自行研究、突破。通过不断的改良代码、尝试新方法，终于研究出准确度高、误判率低的完备算法。希望未来参加智能车竞赛的同学能够在理解这些代码的基础上，推陈出新，将其焕发出新的生命。

3、运用多种手段进行人机交互，站在车的问题上思考出现的问题，有助于问题的解决。

恩智浦杯智能汽车大赛是培养大学生综合动手能力的一个很好的且成熟的平台，在这里，鼓励学生们进行创新，培养了我们的兴趣，让我们对嵌入式系统、图像识别与处理等方面进行了更多的认识，尤其是大大加深了我对ARM单片机的了解，以及对恩智浦单片机系统的深刻理解；于此同时，在这里每个人都会学有所得，做有所获，懂得团队的重要性，学会合作分工，找到自己擅长的一面，弥补自己的不足。

希望这篇技术文档会对未来参与智能车这项比赛的同学有所帮助！

参 考 文 献

[1] 童诗白，华成英．模拟电子技术基础[M]．北京. 高等教育出版社．2000

[2] 邵贝贝. 嵌入式实时操作系统[LC／OS-Ⅱ(第2版)[M]. 北京．清华大学出版社．2004

[3] 阎石.数字电子技术基础[M].北京:高等教育出版社,1998.

[4] 雷霏霖,梁志毅.基于CMOS传感器 OV7620 采集系统设计[J].电子测量技术,2008,12(31):110-112.

[5] 谭浩强.C程序设计[M].北京:清华大学出版社,2005.

[6] 郭芳,曹桂琴.数据结构基础[M].大连：大连理工大学出版社，1994.

[7] 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.

[8] 胡寿松.自动控制原理(第六版)[M].科学出版社,2014.

[9] 张文春．汽车理论[M]．北京．机械工业出版社．2005

[10] YUAN Quan,ZHANG YunZhou,WU Hao,et al. Fuzzy Control Research In The Courses Of Smart Car[C]. Machine Vision and Human-Machine Interface (MVHI), Kaifeng,China, 2010: 764-767.

[11] 侯虹.采用模糊PID控制律的舵机系统设计[J].航空兵器，2006,2(1):7-9.

[12] 孙浩,程磊,黄卫华,等.基于HCS12的小车智能控制系统设计[J].单片机与嵌入式系统应用,2007,03(16):51-54.

[13] LU Zhenlin,LI Jingjiao,Zhang Minghui.Intelligent Control Research based on the Smart Car[C]. Advanced Computer Control (ICACC),Shenyang,2010:292-297.

附 录

附录A：程序源代码

#include "MKL\_BME.h"

#include "include\_own.h"

long int Time\_1ms = 0;

struct CarMotor Motor = {0,0,0,0,0,0,0,0,0,0,0,0};

struct CarServo Servo = {0,0,0};

struct CarStartLine StartLine = {0,0,0,0,0};

struct CarBlock Block = {0,0,0,0};

void main()

{

DisableInterrupts;

delayms(100);

System\_Init(); //模块初始化

delayms(100);

EnableInterrupts;

while(1)

{

///////

ov7725\_get\_img();

Process\_image();

if(Switch\_6==1&&Switch\_8==1)

SCI\_Send\_fireImage(UART1);

if(Switch\_6==1&&Switch\_8==0)

SCISend\_to\_PIDDebug(UART1);

LCD();

}

}

void System\_Init()

{

/\*\*\*\*\*蜂鸣器\*\*\*\*\*/

gpio\_init(PORTE,9,GPO,LOW);

Buzzer\_ringms(100);

/\*\*\*\*LED:全灭\*\*\*\*/

gpio\_init(PORTE,8, GPO, HIGH);

gpio\_init(PORTE,7, GPO, HIGH);

/\*\*\*\*\*\*\*LCD\*\*\*\*\*\*/

LCD\_init(FALSE); //TRUE 为正着刷屏，

Disp\_single\_colour(Yellow);

/\*\*\*\*\*\*\*MPU6050\*\*\*\*\*\*/

Init\_MPU6050();

/\*\*\*\*\*\*boma\*\*\*\*\*\*/

Switch\_init();

/\*\*\*\*\*\*按键\*\*\*\*\*\*/

gpio\_init(PORTB, 10, GPI\_UP, LOW);

gpio\_init(PORTB, 11, GPI\_UP, LOW);

/\*\*\*\*串口初始化\*\*\*\*/

uart\_init(UART1,115200);

uart\_irq\_EN(UART1);//EN

uart\_irq\_DIS(UART1);//dis

uart\_putchar(UART1,'G');

uart\_putchar(UART1,'O');

/\*\*\*\*\*\*定时器\*\*\*\*\*/

pit\_init\_ms(PIT0, 1); //5ms定时器

set\_irq\_priority(68,2);

enable\_irq(68); //中断向量号68

/\*\*\*\*\*电机舵机PWM\*\*\*\*/

PWM\_init();

/\*\*\*\*霍尔测速初始化\*\*\*\*\*\*/

Decoder\_init();

/\*\*\*\*\*\*重要参数初始化\*\*\*\*\*\*/

Parameters\_init();

/\*\*\*\*鹰眼摄像头初始化,包括场中断和DMA\*\*\*\*\*\*/

Ov7725\_Init();

/\*\*\*\*\*\*\*结束初始化\*\*\*/

}

#include "include\_own.h"

void LCD\_PORT\_init()

{

gpio\_init(PORTD, 8, GPO, HIGH);//cs

gpio\_init(PORTD, 9, GPO, HIGH);//rst

gpio\_init(PORTD, 10, GPO, HIGH);//dc

gpio\_init(PORTD, 11, GPO, HIGH);//sdi

gpio\_init(PORTD, 12, GPO, HIGH);//scl

}

void LCD\_Reset()

{

reset = 0;

delayms(20);

reset = 1;

delayms(20);

}

void LCD\_init\_own(unsigned char direction)

{

reset=0;

delayms(20);

reset=1;

delayms(20);

write\_command(0xCB);

write\_data(0x39);

write\_data(0x2C);

write\_data(0x00);

write\_data(0x34);

write\_data(0x02);

write\_command(0xCF);

write\_data(0x00);

write\_data(0XC1);

write\_data(0X30);

write\_command(0xE8);

write\_data(0x85);

write\_data(0x00);

write\_data(0x78);

write\_command(0xEA);

write\_data(0x00);

write\_data(0x00);

write\_command(0xED);

write\_data(0x64);

write\_data(0x03);

write\_data(0X12);

write\_data(0X81);

write\_command(0xF7);

write\_data(0x20);

write\_command(0xC0);

write\_data(0x23);

write\_command(0xC1);

write\_data(0x10);

write\_command(0xC5);

write\_data(0x3e);

write\_data(0x28);

write\_command(0xC7);

write\_data(0x86); //--

write\_command(0x36);

if(direction)

write\_data(0xE8); //C8

else

write\_data(0x28);//0b00101000

write\_command(0x3A);

write\_data(0x55);

write\_command(0xB1);

write\_data(0x00);

write\_data(0x18);

write\_command(0xB6);

write\_data(0x08);

write\_data(0x82);

write\_data(0x27);

write\_command(0xF2); write\_data(0x00);

write\_command(0x26); write\_data(0x01);

write\_command(0xE0);

write\_command(0x11); //Exit Sleep

delayms(20);

write\_command(0x29); //Display on

write\_command(0x2c);

}

void write\_command(unsigned char c)

{

cs=0;

dcx=0;

sdi=(GET\_BITFIELD(c))->bit7;scl=0;scl=1;

sdi=(GET\_BITFIELD(c))->bit6;scl=0;scl=1;

sdi=(GET\_BITFIELD(c))->bit5;scl=0;scl=1;

sdi=(GET\_BITFIELD(c))->bit4;scl=0;scl=1;

sdi=(GET\_BITFIELD(c))->bit3;scl=0;scl=1;

sdi=(GET\_BITFIELD(c))->bit2;scl=0;scl=1;

sdi=(GET\_BITFIELD(c))->bit1;scl=0;scl=1;

sdi=(GET\_BITFIELD(c))->bit0;scl=0;scl=1;

cs=1;

}

void write\_data(unsigned char d)

{

cs=0;

dcx=1;

sdi=(GET\_BITFIELD(d))->bit7;scl=0;scl=1;

sdi=(GET\_BITFIELD(d))->bit6;scl=0;scl=1;

sdi=(GET\_BITFIELD(d))->bit5;scl=0;scl=1;

sdi=(GET\_BITFIELD(d))->bit4;scl=0;scl=1;

sdi=(GET\_BITFIELD(d))->bit3;scl=0;scl=1;

sdi=(GET\_BITFIELD(d))->bit2;scl=0;scl=1;

sdi=(GET\_BITFIELD(d))->bit1;scl=0;scl=1;

sdi=(GET\_BITFIELD(d))->bit0;scl=0;scl=1;

cs=1;

}

void write\_word(unsigned int dat)

{

write\_data(dat>>8);

write\_data(dat);

}

void RamAdressSet()

{

write\_command(0x2A);

write\_data(0x00);

write\_data(0x00);

write\_data(0xdb);

}

void LCD\_SetPos(unsigned int x0,unsigned int x1,unsigned int y0,unsigned int y1)

{

write\_command(0x2A);

write\_data(y0);

write\_data(y1>>8);

write\_data(y1);

write\_command(0x2c);

}

void LCD\_PutChar(unsigned int x, unsigned int y,unsigned char c, unsigned int fColor, unsigned int bColor)

{

LCD\_PutChar8x16( x, y, c, fColor, bColor );

}

void LCD\_PutString(unsigned int x, unsigned int y, unsigned char \*s, unsigned int fColor, unsigned int bColor)

{

unsigned char l=0;

while(\*s) {

if( \*s < 0x80)

{

LCD\_PutChar(x+l\*8,y,\*s,fColor,bColor);

s++;l++;

}

else

{

PutGB1616(x+l\*8,y,(unsigned char\*)s,fColor,bColor);

s+=2;l+=2;

}

}

}

void Disp\_single\_colour(unsigned int color)

{

unsigned int i,j;

LCD\_SetPos(0,320-1,0,240-1);

for (i=0;i<240;i++)

for (j=0;j<320;j++)

write\_word(color);

}

void Draw\_single\_line(unsigned char axis,unsigned int line,unsigned int color)

{

unsigned int i;

if(axis=='L')

{

LCD\_SetPos(line,line,0,240-1);

for (i=0;i<240;i++)

write\_word(color);

}

else if(axis=='H')

{

LCD\_SetPos(0,320-1,line,line);

for (i=0;i<320;i++)

write\_word(color);

}

}

void LCD\_Put\_Unsigned\_Int(unsigned int x,unsigned int y,unsigned char s0[],unsigned int pnum,unsigned int fColor, unsigned int bColor)

{

unsigned char l=0;

unsigned char s[30];

unsigned char count=0;

//copy

count=0;

while(s0[count])

{

s[count] = s0[count];

count++;

}

//strcat

if(pnum<10)

{ s[count]=pnum+'0';s[count+1]='\0';}

else if((10<=pnum)&&(pnum<100))

{ s[count]=pnum/10+'0';s[count+1]=pnum%10+'0';s[count+2]='\0';}

else if(pnum<=999)

{

s[count]=pnum/100+'0';

s[count+1]=pnum/10%10+'0';

s[count+2]=pnum%10+'0';

s[count+3]='\0';

}

else if(pnum<=9999)

{

s[count]=pnum/1000+'0';

s[count+1]=pnum/100%10+'0';

s[count+2]=pnum/10%10+'0';

s[count+3]=pnum%10+'0';

s[count+4]='\0';

}

else if(pnum<=99999)

{

s[count]=pnum/10000+'0';

s[count+1]=pnum/1000%10+'0';

s[count+2]=pnum/100%10+'0';

s[count+3]=pnum/10%10+'0';

s[count+4]=pnum%10+'0';

s[count+5]='\0';

}

count=0;

while(s[count])

{

if( s[count] < 0x80)

{

LCD\_PutChar(x+l\*8,y,s[count],fColor,bColor);

count++;

l++;

}

else

{

PutGB1616(x+l\*8,y,(unsigned char\*)s[count],fColor,bColor);

count+=2;

l+=2;

}

}

}

void LCD\_Put\_Int(unsigned int x,unsigned int y,unsigned char s0[],int pnum,unsigned int fColor, unsigned int bColor)

{

unsigned char l=0;

unsigned char s[30];

unsigned char count=0;

//copy

count=0;

while(s0[count])

{

s[count] = s0[count];

count++;

}

//strcat

if(pnum < 0)

{

s[count] = '-';

count++;

pnum = (-1) \* pnum;

}

if(pnum < 10)

{ s[count]=pnum+'0';s[count+1]='\0';}

else if((10<=pnum)&&(pnum<100))

{ s[count]=pnum/10+'0';s[count+1]=pnum%10+'0';s[count+2]='\0';}

else if(pnum<=999)

{

s[count]=pnum/100+'0';

s[count+1]=pnum/10%10+'0';

s[count+2]=pnum%10+'0';

s[count+3]='\0';

}

else if(pnum<=9999)

{

s[count]=pnum/1000+'0';

s[count+1]=pnum/100%10+'0';

s[count+2]=pnum/10%10+'0';

s[count+3]=pnum%10+'0';

s[count+4]='\0';

}

else if(pnum<=99999)

{

s[count]=pnum/10000+'0';

s[count+1]=pnum/1000%10+'0';

s[count+2]=pnum/100%10+'0';

s[count+3]=pnum/10%10+'0';

s[count+4]=pnum%10+'0';

s[count+5]='\0';

}

count=0;

while(s[count])

{

if( s[count] < 0x80)

{

LCD\_PutChar(x+l\*8,y,s[count],fColor,bColor);

count++;

l++;

}

else

{

PutGB1616(x+l\*8,y,(unsigned char\*)s[count],fColor,bColor);

count+=2;

l+=2;

}

}