



第七届“飞思卡尔”杯全国大学生 智能汽车竞赛

技 术 报 告

学 校：湖北汽车工业学院科技学院

队伍名称：独眼龙 4.0

参赛队员：陈慧超

于健博

高青鹏

带队教师：周 恒

柯 瀚

关于技术报告和学术论文使用授权的说明

本人完全了解第七届“飞思卡尔”杯全国大学生智能汽车竞赛关保留、使用技术报告和学术论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会和飞思卡尔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

目 录	II
第一章 引言	1
第二章 模型车整车结构设计与调校	2
2.1 车体的整体参数	2
2.2 主电路板安装	2
2.3 前轮定位参数的设定	3
2.4 舵机的安装	5
2.5 测速模块安装	6
2.6 差速机构的调整	6
第三章 系统电路设计	8
3.1 单片机功能概述	8
3.2 PCB 电路板	8
3.3 稳压电源	9
3.4 电池的合理使用	10
3.5 驱动电路	10
3.6 最小系统模块	13
第四章 视频采集和图像处理	14
4.1 视频采集	14
4.1.1 摄像头工作原理	14
4.1.2 摄像头的选择	15
4.1.3 信号分离电路	15
4.2 图像处理	17
4.2.1 目标指引线的提取	17
第五章 开发与调试	19
5.1 软件开发平台 CodeWarrior IDE	19
5.2 串口调试软件	20

第六章 模型车的主要技术参数说明	22
6.1 改造后的车模基本参数.....	22
6.2 电路功耗，所有电容总容量.....	22
6.3 传感器种类以及个数.....	22
6.4 除了车模原有的驱动电机、舵机之外伺服电机个数.....	22
6.5 赛道信息检测精度、频率.....	22
第七章 结论与展望	23
7.1 结论与总结.....	23
致 谢	24
参考文献	I
附录 A: 程序源代码.....	II

第一章 引言

本智能小车以飞思卡尔 16 位微控制 MC9S12G128 作为唯一的核控制单元, 采用摄像头作为传感器, 感应由赛道两边黑色引导线检测道路信息, 通过单片机处理优化, 把控制信号发送给电机和舵机。同时通过自制的转速传感器获取小车速度, 进行速度反馈处理, 最后利用 PID 控制方式作为电机驱动。本文对摄像头的彩像及对跑道位置的检测进行了简要的分析, 对小车的硬件与软件设计进行了介绍。

技术报告以智能小车的设计为主线, 包括小车的构架设计、软硬件设计, 以及控制算法研究等, 共分为七章。其中, 第一章为引言部分。第二章主要介绍了小车的总体方案的选取。第三章介绍了硬件设计, 主要介绍了电路的设计; 检测模块的设计。第四章对小车的视频采集和图像处理进行了详细的介绍。第五章描述了小车的软件调试过程。第六章主要介绍小车详细参数。第七章总结智能车经验和不足。

第二章 模型车整车结构与调校

2.1 车体的整体参数

比赛选用的模型车采用 1/10 的仿真车模，如图 2.1 所示，小车整体机构包括车模的底盘部分及转向和驱动部分。控制采用前轮转向，后轮驱动方案。A 车模已经延续七届，对此我们对 A 车模的调教已经有很多经验。

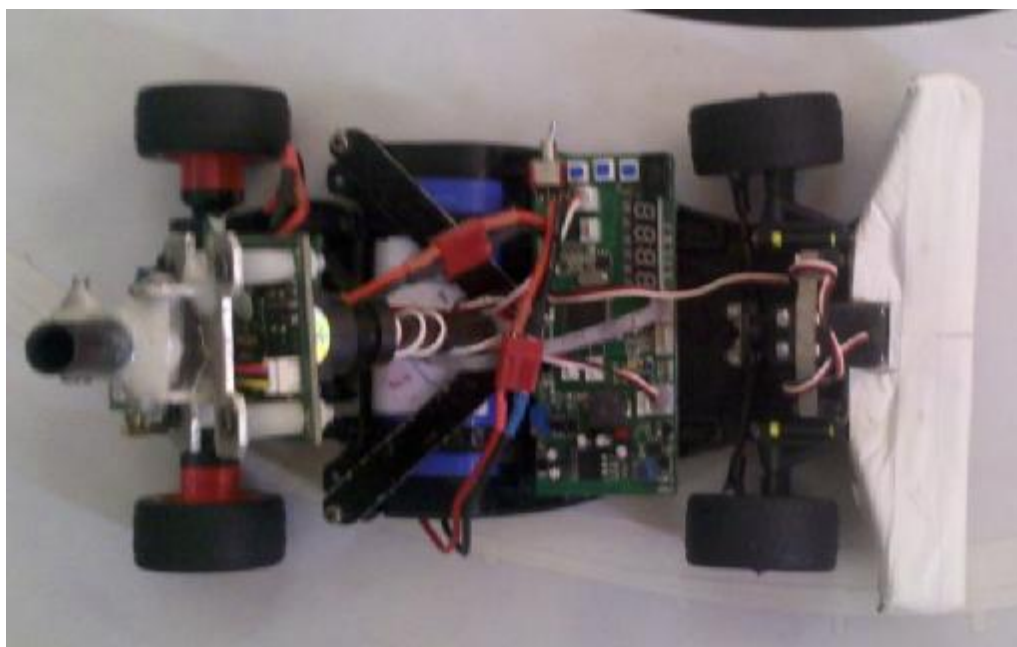


图 2.1 车模全貌

2.2 主电路板安装

为保证整车的紧凑性和整体质量的集中，在设计电路板时将其形状设计成大致与底盘相同的形状，利用底盘上原有的定位孔将其平铺在底盘上，固定好，这样做也是为了主板的拆装的方便快捷，也有利于主板的维修。因为考虑到 A 型车模，车模空间有限，又考虑到车体本身的重心高度，所以这次设计的主电路板的装配位置在底板上，故将电路板设计的大小比较窄。

由于电路板背面有一些裸露的铜线，所以为防止短路，特用胶枪固定防止短路。

2.3 前轮定位参数的设定

关于前轮定位参数的设定，我们可以通过改变转向横拉杆，前悬架横杆来改变前轮 **Caster** 角度。大度数的 **Caster** 可以增加车辆入弯时的转向，但是会增加车辆出弯时的转向，同时可以增加在颠簸的路面上行走的稳定性。适应比较小的 **Caster** 可以减少入弯时的转向，但会增加在弯道中间的转向，和出弯时候的转向。

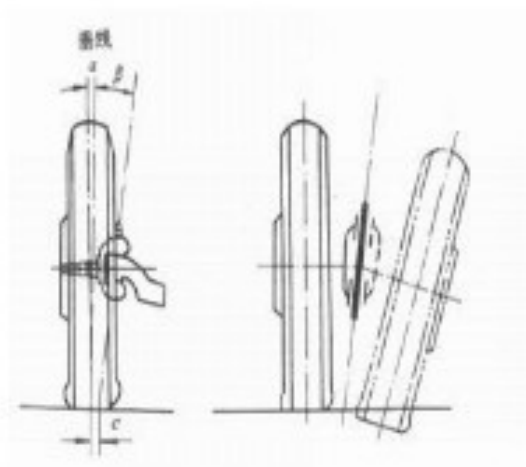
前轮束角（**Toe**）**Toe** 角度是当从模型车的正上方俯视底板，车轮的前端和车辆纵线的夹角。前轮 **Toe-in** 即前轮前端向内倾（内八字）；前轮 **Toe-out** 即前轮向外倾（外八字），称为 **Toe-out**。

通过改变转向拉杆的长度，可以改变车轮前束，如下图所示



调试中我们发现，在赛车过弯时，转向舵机的负载会因为车轮转向角度增大而增大。为了尽可能降低转向舵机负载，对前轮的安装角度，即前轮定位进行了调整。

前轮定位的作用是保障汽车直线行驶的稳定性，转向轻便和减少轮胎的磨损。前轮是转向轮，它的安装位置由主销内倾、主销后倾、前轮外倾和前轮前束 4 个参数决定，反映了转向轮、主销和前轴等三者在车架上的位置关系。



示意图

主销内倾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。

主销后倾是指主销装在前轴，上端略向后倾斜的角度。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，前轮稳定性也愈好。

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，因此高速时后倾的回正作用大，低速时内倾的回正作用大。

前轮外倾角对汽车的转弯性能有直接影响，它的作用是提高前轮的转向安全性和转向操纵的轻便性。前轮外倾角俗称“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。所以事先将车轮校准一个外八字角度，这个角度约在 1° 左右。

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。

经过与赛道的磨合，我们智能车前轮角度调整为前轮外倾角为 -2° ，其他皆为 0° 。

2.4 舵机的安装

舵机是一种位置伺服的驱动器，适用于那些需要角度不断变化并可以保持的控制系统，模型中使用的型号是：T170A。其工作原理是：控制信号由接收机的通道进入信号调制芯片，获得直流偏置电压。它内部有一个基准电路，产生周期为 20ms，宽度为 1.5ms 的基准信号，将获得的直流偏置电压与电位器的电压比较，获得电压差输出。最后，电压差的正负输出到电机驱动芯片决定电机的正反转。当电机转速一定时，通过级联减速齿轮带动电位器旋转，使得电压差为 0，电机停止转动。舵机的控制信号是 PWM 信号，利用占空比的变化改变舵机的位置。



舵机转向是整个控制系统中延迟较大的一个环节，为了减小此时间常数，通过改变舵机的安装位置，而并非改变舵机本身结构的方法可以提高舵机的响应速度。分析舵机控制转向轮转向的原理，可以发现在相同的舵机转向条件下，转向连杆在舵机的一端的连接点离舵机轴心距离越远，转向轮转向变化越快。这相当于增大力臂长度，提高线速度。针对上述特性，改变了原有的舵机的安装方法，对舵机如图 2.2 所示的安装。这样安装的优点是：①改变了舵机的力臂，使转向更灵敏；②舵机安装在正中央，使舵机左右转向时受力比较均匀，使小车能灵活的转向。

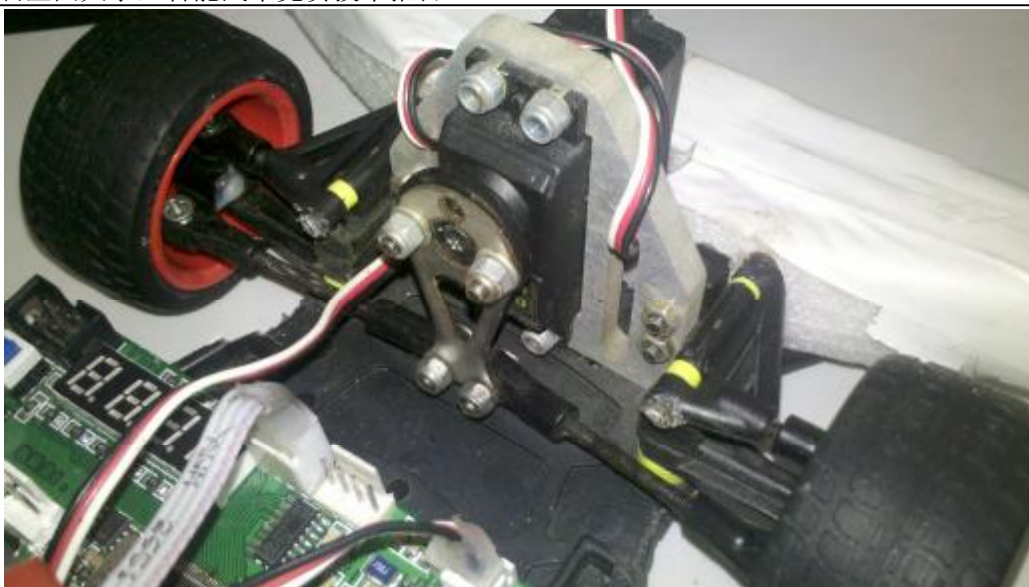


图 2.4 舵机安装图

2.5 测速模块安装

我们采用的是旋转编码器测速，编码器旋转一周回产生 100 个脉冲，已经充分满足的赛车的检测精度，从而提高了赛车高速时速度控制的稳定性，大大小了检测误差。



图 2.5 编码器的安装

2.6 差速机构的调整

差速机构的作用是在赛车转弯的时候，降低后轮与地面之间的滑动；并且还可以保证在轮胎抱死的情况下不会损害到电机。

差速器有差速作用和限滑作用：差速作用使两轮可以相对转动，并使两轮转动速度的平均值等于整驱动轴的转动速度；限滑作用在两轮相对转动时产生阻力，以限制两轮速度差，防止其中一个轮子发生过度打滑和空转。在转弯的过程中，外侧

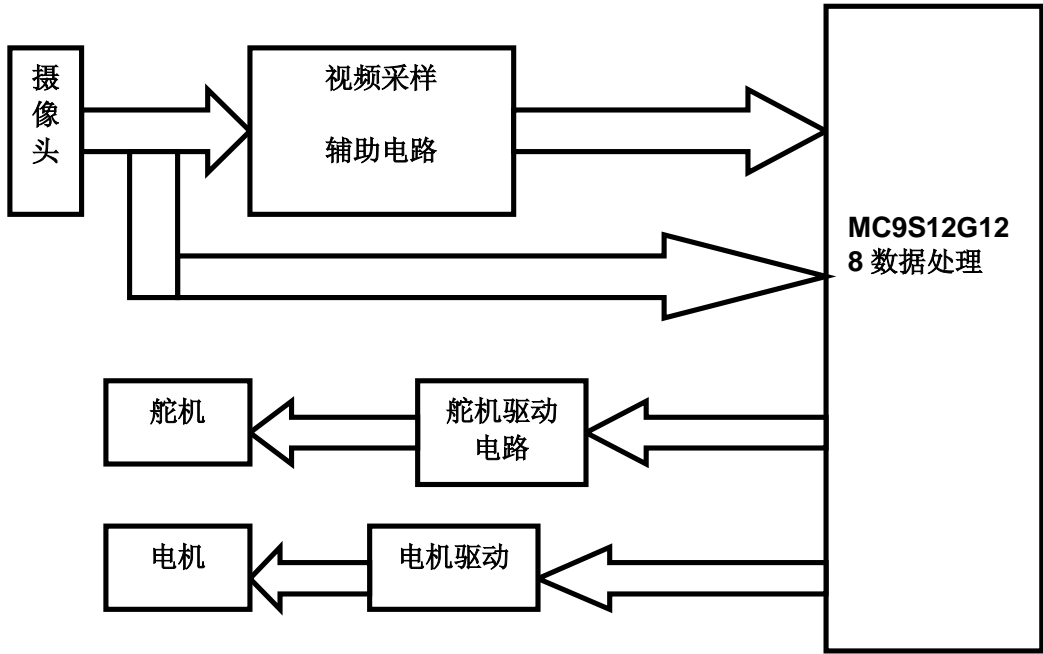
轮的转弯半径大于内侧轮的转弯半径，所以外侧轮需要比内侧转的更快。如果这时差速过紧（限滑作用过大），造成两个轮子互相角力以使自己得到适当的转速，从而使总体抓地力下降。另一方面，如果差速比较松，当一个轮子失去抓地力的时候(比如通过一个起伏)，会造成过度的空转，使另外一个轮子失去动力。这样会减少加速的反应。滚珠差速通常是由两块摩擦片，夹住装有差速珠的齿轮，并有调整压力的装置(弹簧/压力片和轴承)。在差速珠和摩擦片间涂有摩擦油。差速珠可以使摩擦片相对转动，而调整压力的装置和摩擦油可以控制差速珠在摩擦片上滚动时产生的阻力。对于滚珠差速来说，将差速拧紧，或者使用更浓的摩擦油，可以限制轮子相对转动；放松差速，或者使用更稀的摩擦油，可以使轮子更容易相对转动。一般来说，不宜使齿轮可相对于滚珠差速整体滑动，这样会损失动力，并且容易磨损差速珠和摩擦片。因此差速不能拧得过松。如果差速太紧，但齿轮仍能滑动，请更换更稀的摩擦油。

第三章 系统电路设计

3.1 单片机功能概述

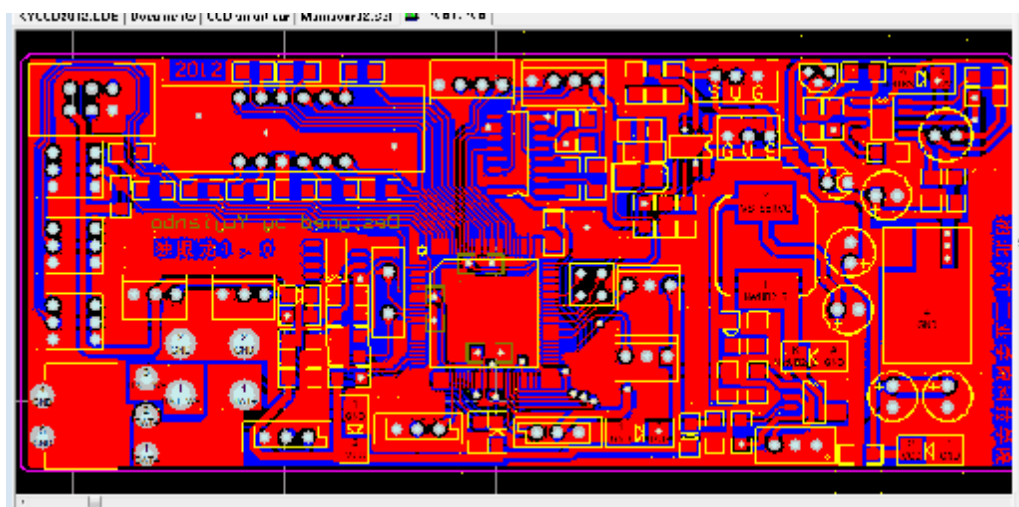
独眼龙 4.0 电路主要可以分为如下几个功能模块：电源电路、MC9S12G128，最小系统模块、速度反馈电路、电机驱动电路及其他辅助电路。通过 CCD 传感器和速度反馈采集小车当前的位置和速度信息，通过 MC9S12G128 最小系统模块对数据进行分析并操控电机驱动电路与转向舵机等达到对小车实时的闭环控制。

电路系统框架：



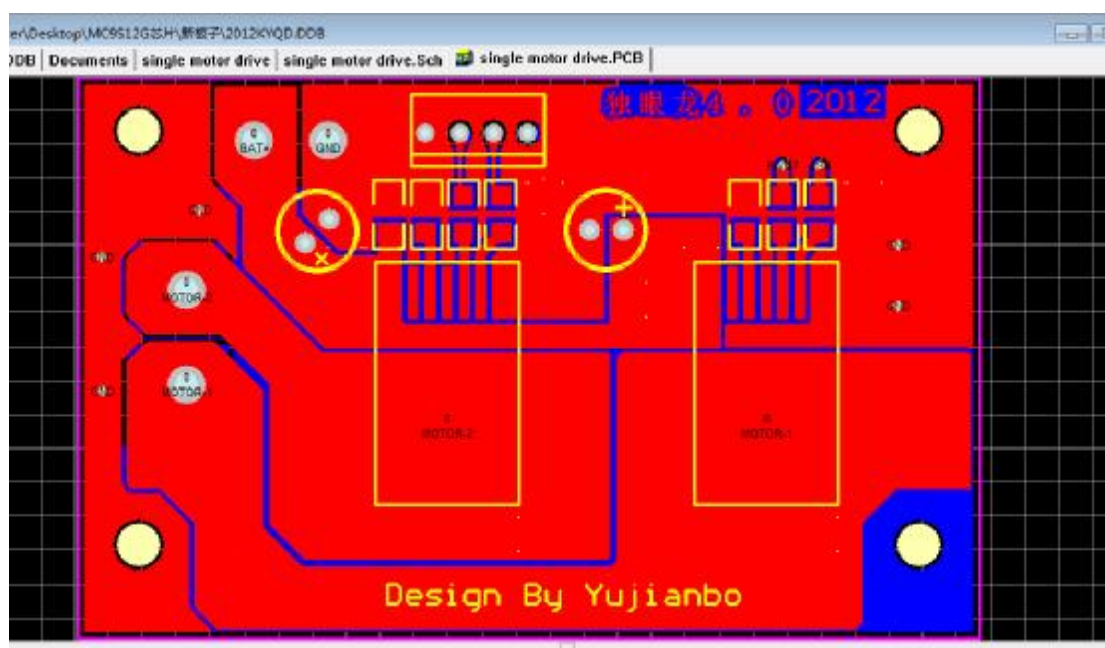
3.2 PCB 电路板

在最初的设计中主要考虑了实际需求、电路可行性、硬件布局及电路的简洁高效等方面。在 PCB 布局方面我们出于减少硬件所占车体面积与电路板重复利用的考虑。



系统板

由于空间限制，我们采用电机驱动和系统板分离，也达到了效果不错。



电机驱动 PCB

3.3 稳压电源

稳压电源的设计中我们选择了 LM2596 可调稳压芯片两路来为整个系统供电。LM2596 系列是美国国家半导体公司生产的 3A 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 (150KHZ)，和基准稳压器 (1.23v)，并具有完善的保护电路：电流限制、热关断电路等。该芯片已经完全能满足该车对负载的要求。在两路 LM2596 电路中我们采用一路以定值电阻调成 5V 电压给最小系统、数码管、按键等电路供电；令采用一路 LM2596 以可调压设计，对 CCD 以及转向舵机进行供电，根据舵机响应

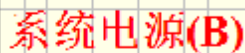


图 3.3 供电电源原理图

3.4 电池的合理使用

飞思卡尔智能车的电源为大赛组委会提供的 7.2v 镍-镉充电电池。该电池记忆性较强，因此合理的充放电是保护电池的必要措施。合理使用电源包括如下措施：

第一，实时检测电池电压，避免放电过度对电池造成损害，考虑到电路的简洁性和稳定性，最终投板时我们并没有设计低压警报模块。在日常调试中，通过观察车况来判断电池电量。

第二，统一管理合理充电，开始/停止充电登记管理，控制充电时间，充满电。

3.5 驱动电路

H 桥驱动电路图 3.51 中所示为一个典型的直流电机控制电路。电路得名于“H 桥驱动电路”是因为它的形状酷似字母 H。4 个三极管组成 H 的 4 条垂直腿，而电机就是 H 中的横杠（注意：图 3.51 及随后的两个图都只是示意图，而不是完整的电路图，其中三极管的驱动电路没有画出来）。

如图所示，H 桥式电机驱动电路包括 4 个三极管和一个电机。要使电机运转，必须导通对角线上的一对三极管。根据不同三极管对的导通情况，电流可能会从左至右或从右至左流过电机，从而控制电机的转向。

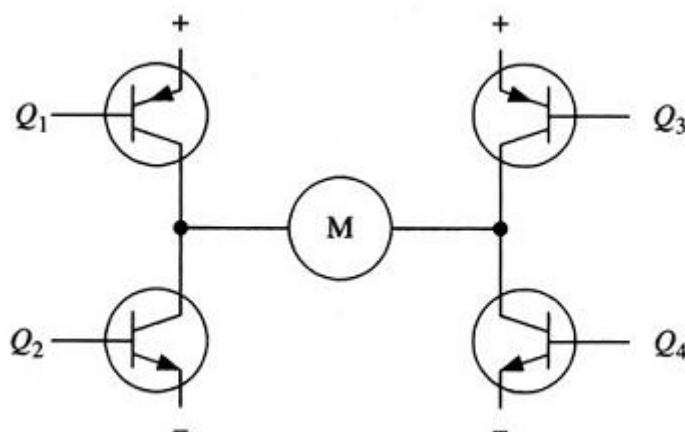


图 3.51 H 桥驱动电路

要使电机运转，必须使对角线上的一对三极管导通。例如，如图 3.52 所示，当 Q_1 管和 Q_4 管导通时，电流就从电源正极经 Q_1 从左至右穿过电机，然后再经 Q_4 回到电源负极。按图中电流箭头所示，该流向的电流将驱动电机顺时针转动。当三极管 Q_1 和 Q_4 导通时，电流将从左至右流过电机，从而驱动电机按特定方向转动（电机周围的箭头指示为顺时针方向）。

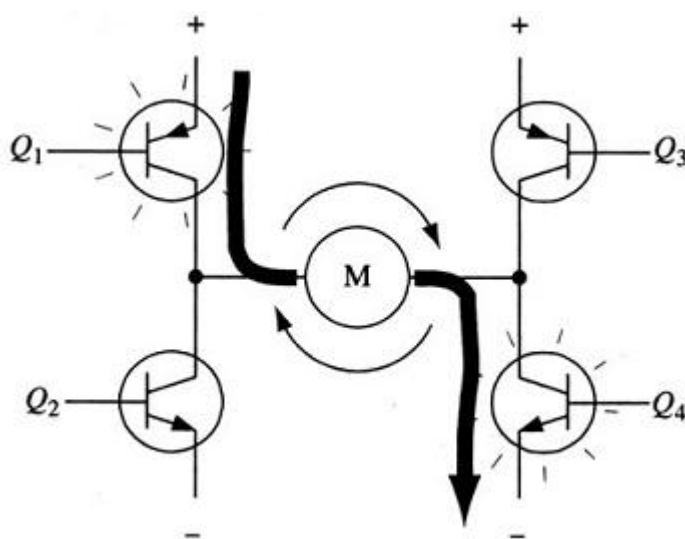


图 3.52 H 桥电路驱动电机顺时针转动

图 3.53 所示为另一对三极管 Q_2 和 Q_3 导通的情况，电流将从右至左流过电机。当三极管 Q_2 和 Q_3 导通时，电流将从右至左流过电机，从而驱动电机沿另一方向转动（电机周围的箭头表示为逆时针方向）。

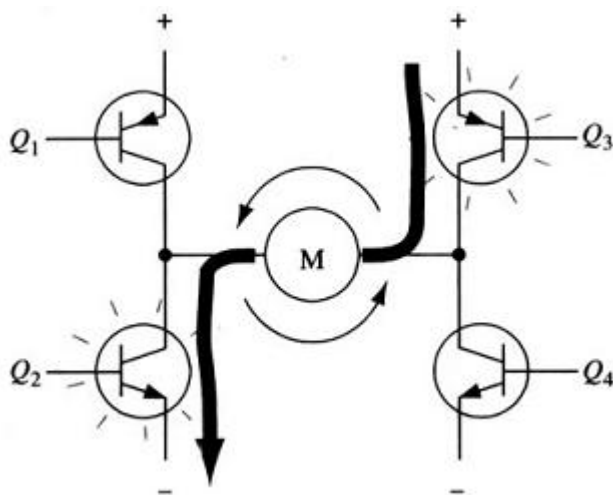
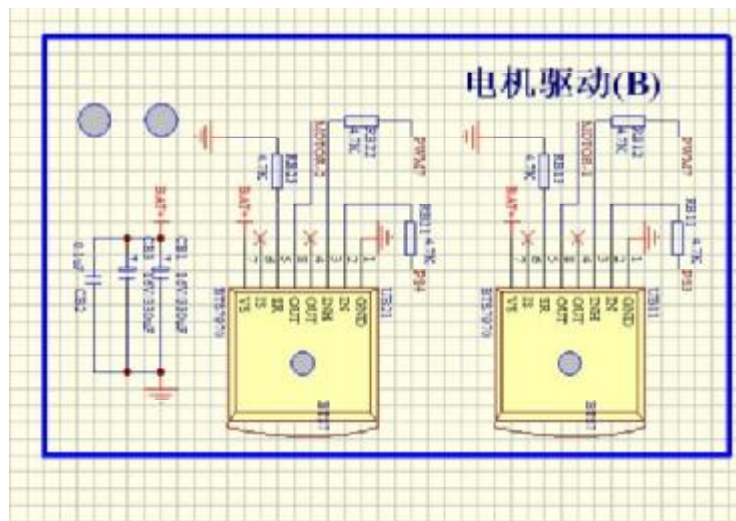


图 3.53 H 桥驱动电机逆时针转动

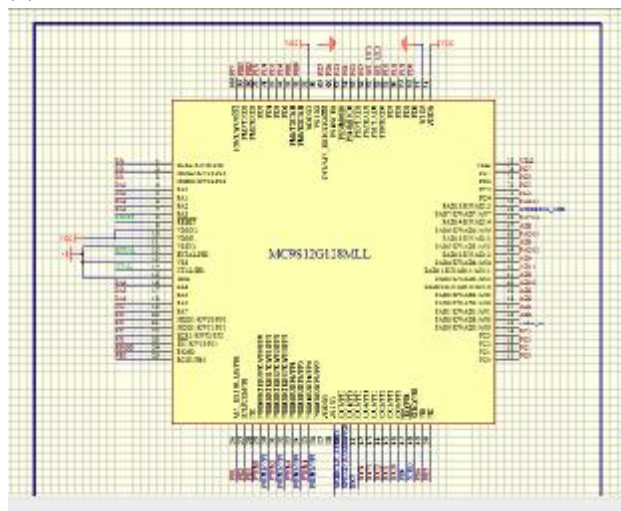
电池系统和电机驱动系统决定了智能车在运行过程中的加减速性能。从驱动能力上来说，MC33886 的驱动电流可达 5.0A，满足本系统的需要。在设计时，我们采用两片 MC33886 并联的方法来驱动电机，总的驱动电流可达 10A。但是 33886 的内阻、自身功耗和压降都较大。相对即将讨论的 BTS7960 而言，此芯片并不适合这种追求速度的比赛。

BTS7960 是半桥驱动芯片，就是说需要 2 个芯片来驱动一个电机，电流最高 43A。其标称驱动电流最大可达 43A，内阻为 7 至 9 毫欧，比 MC33886 高出一个数量级，其内阻小，功耗小，压降小，发热不是很厉害。所以我们最终决定使用 BTS7960 组成全桥来驱动电机。



3.53 驱动电路原理图

3.6 最小系统模块



3.61 最小系统原理图

第四章 视频采集和图像处理

4.1 视频采集

视频采集模块由摄像头、1881 视频采集信号分离芯片以及 S12 的 AD 模块构成。视频信号是 AD 采集的基础，下面先介绍视频信号的特征，然后再展开。

4.1.1 摄像头工作原理

摄像头分为黑白和彩色两种，根据赛道特点可知，为达到寻线目的，只要提取到画面的灰度信息，而不必要提取其色彩信息，所以本设计中采用的是黑白摄像头。摄像头主要由镜头、图像传感芯片和外围电路构成。图像传感芯片是其最重要的部分，但该芯片要配以合适的外围电路才能工作。将芯片和外围电路制作在一块电路板上，称为“单板”。若给单板配上镜头、外壳、引线和接头，就构成了通常所见的摄像头。摄像头的工作原理是：按一定的分辨率，以隔行扫描的方式采集图像上的点，当扫描到某点时，就通过图像传感芯片将该点处图像灰度转换成与灰度一一对应的电压值，然后将此电压值通过视频信号端输出。具体而言（参见图 5-1），摄像头连续地扫描图像上的一行，则输出就是一段连续的电压信号，该电压信号的高低起伏反映了该行图像灰度变化。当扫描完一行，视频信号端就输出一个低于最低视频信号电压的电平（如 0.3V），并保持一段时间。这样相当于，紧接着每行图像信号之后会有一个电压“凹槽”，此“凹槽”叫做行同步脉冲，它是扫描换行的标志。然后，跳过一行后（摄像头是隔行扫描的），开始扫描新的一行，如此下去，直到扫描完该场的视频信号，接着会出现一段场消隐区。该区中有若干个复合消隐脉冲，其中有个远宽于（即持续时间远长于）其它的消隐脉冲，称为场同步脉冲，它是扫描换场的标志。场同步脉冲标志着新的一场的到来，不过，场消隐区恰好跨在上一场的结尾和下一场的开始部分，得等场消隐区过去，上一场的视频信号才真正到来。摄像头每秒扫描 25 幅图像，每幅又分奇、偶两场，先奇场后偶场，故每秒扫描 50 场图像。奇场时只扫描图像中的奇数行，偶场时则只扫描偶数行。

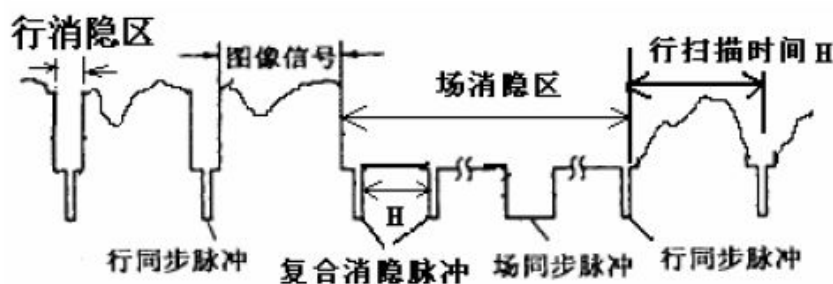


图 4-1 摄像头视频信号

4.1.2 摄像头的选择

因为 S12 单片机的 AD 转换时间在不超频的情况下最短为 $7\mu\text{s}$ ，所以如果选用一个分辨率为 320 的摄像头，则单行视频信号持续时间约为 $20\text{ms}/320=62.5\mu\text{s}$ ，AD 对单行视频信号采样的点数将不超过 $[62.5/7]+1=9$ 个。若使用分辨率越高的，则其采到的点数将更少。如前所述，摄像头的分辨率越高，尽管可提高纵向分辨能力，却会减少单片机 AD 采样单行信号的点数，削弱了横向分辨能力。现在市场上摄像头的分辨率通常都在 300 线以上，由此推得单行视频信号的持续时间至多为 $20\text{ms}/300=66\mu\text{s}$ ，AD 采样每行视频信号的点数至多为 $[66/7]+1=10$ 个（不超频的情况下），这对赛车定位来说是不够的，所以在选用摄像头时，应当尽量选择分辨率低的摄像头，这样做会降低摄像头的纵向分辨率能力（但降低都仍然远远够用），却可以增加单片机采样视频信号的点数，提高横向分辨能力。在本次设计过程中，在市场上我们找到了索尼的 CCD 摄像头，其分辨率为 200 万像素，以下章节中的视频采集工作就是基于此摄像头。

4.1.3 信号分离电路

要对有效地对视频信号进行采样，首先要处理好的问题是如何提取出摄像头信号中的行同步脉冲，消隐脉冲和场同步脉冲。这里有两种可行的方法。第一，直接通过单片机 AD 进行提取。因为行同步脉冲、消隐脉冲或场同步脉冲信号的电平低于这些脉冲以外摄像头信号的电平，所以据此可设定一个信号电平阈值来判断 AD 采样到的信号是否为上述三类脉冲。第二，就是给单片机配以合适的外围芯片，此芯片要能够提取出摄像头信号的行同步脉冲、消隐脉冲和场同步脉冲以供单片机作

控制之用。考虑到单片机的速度有限，而一些脉冲的间隔时间又较短，为了减轻其处理负担，采用了第二种方法进行信号的时序信息，如行同步脉冲、场同步脉冲和奇、偶场信号信息等，并将它们转换成 TTL 电平直接输出给单片的 I/O 口作控制信号之用。1881 的端口接线方式如图 5-2 所示。

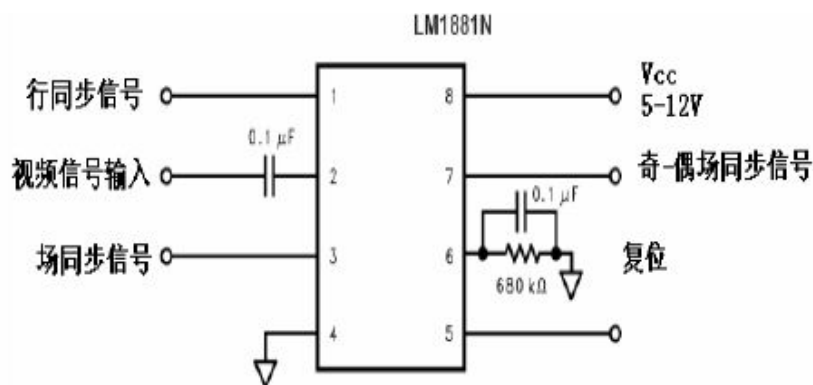


图 4-2 LM1881

其中，引脚 2 为视频信号的输入端，引脚 1 为行同步信号输出端（图 5-3 中的 b）。引脚 3 为场同步信号输出端，当摄像头信号的场同步脉冲到来时，该端将变为低电平，一般维持 230us，然后重新变回高电平（如图 5-3 中的 c）。引脚 7 为奇-偶场同步信号端输出端，当摄像头信号处于奇场时，该端为高电平，当处于偶场时，为低电平。事实上，不仅可以用场同步信号作为换场的标志，也可以用奇-偶场间的交替作为换场的标志。

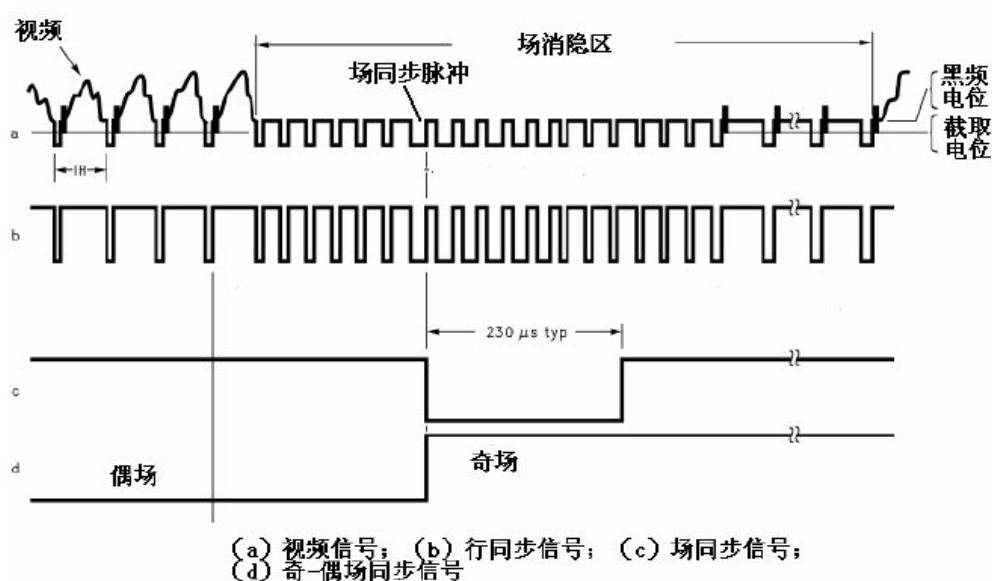


图 4-3 LM1881 信号时序图

4.2 图像处理

4.2.1 目标指引线的提取

智能车通过图像采样模块获得车前方的赛道图像信息，往下介绍如何分析此二维数组来提取黑线，我们采用边缘检测的方法。二维数组的行数和列数即为像素的图像坐标，我们若求出了黑线边缘的图像坐标，就知道了黑线的位置。黑线边缘的特点是其左、右两像素为一黑一白，两像素值的差的绝对值大于某阈值，大于可根据试验确定；而其余处的相邻两像素或全白，或全黑，像素值差的绝对值小于该阈值。这样，只要我们对两数组每行中任何相邻两点做差，就可以根据差值的大小是否大于该阈值来判断此两点处是否为黑线边缘，还可以进行根据差值的正负来判定边缘处是左白右黑，还是右黑左白。从最左端的第一个有效数据点开始依次向右进行阈值判断：由于实际中黑白赛道边缘可能会出现模糊偏差，导致阈值并不是个很简单介于两相邻之间，很可能要相隔两个点。因此：第 $line$ 为原点，判断和 $line+3$ 的差是否大于该阈值，如果是则将 $line+3$ 记为 i ，从 i 开始继续在接下的从 $i+3$ 到该行最末一个点之间的差值是否大于阈值，如果大于则将 $line+i/2+1$ 的坐标赋给中心给黑线中心位置值，如 5-4

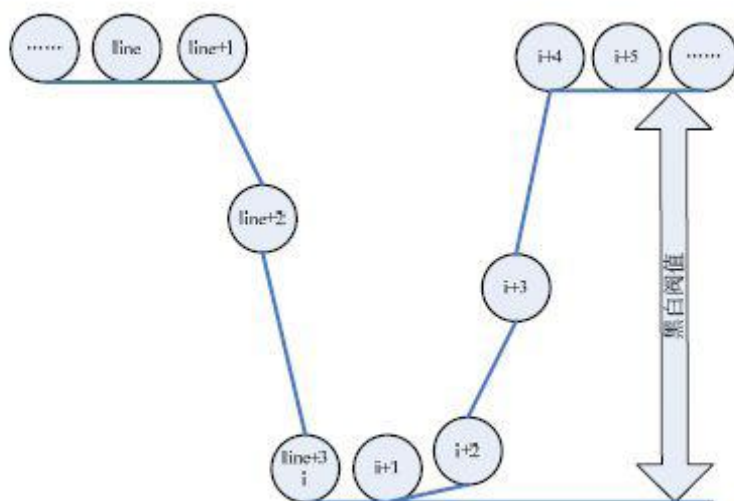


图4-4 单行黑线提取法

利用该算法所得到的黑线提取效果不仅可靠，而且实时性好；在失去黑线目标以后能够记住是从左侧还是从右侧超出视野，从而控制舵机转向让赛车回到正常赛

道。试验表明：只要阈值取得合适，该算法不仅可靠，而且实时性较好。如果更进一步可以设置阈值根据现场情况的变化而变化。在黑线引导线已经能够可靠提取的基础上，我们可以利用它来进行相应的弯、直道判定，以及速度和转向舵机控制算法的研究。

第五章 开发与调试

前面的章节中，已经对小车设计方案从硬件设计到软件构架都做了详尽的介绍和分析，但是这只是如同河的两岸——只能遥遥相对却无法相互沟通。所以在对算法进行开发和软硬件联调的过程中就需要一整套的软件开发与调试工具。在整个开发调试过程中，除汇编语言开发的方式以外，使用 Metrowerks 公司为 MC9S12 系列专门提供的全套开发工具（Metrowerks CodeWarrior IDE）。

5.1 软件开发平台 CodeWarrior IDE

CodeWarrior 是由 Metrowerks 公司提供的专门面向 Freescale 所有 MCU 与 DSP 嵌入式应用开发的软件工具，支持摩托罗拉汇编语言、ANSI C 语言和 C++ 语言。本系统在软件开发过程中使用的是 CodeWarrior V4.7。其主要包括集成开发环境 IDE、处理器专家、全芯片仿真、可视化参数显示工具、项目工程管理、专家生成系统、C 交叉编译器、汇编器、链接器以及调试器。

CodeWarrior IDE 能够自动地检查代码中的明显错误，它通过一个集成的调试器和编辑器来扫描你的代码，以找到并减少明显的错误，然后编译并链接程序以便计算机能够理解并执行你的程序。每个应用程序都经过了使用象 CodeWarrior 这样的开发工具进行编码、编译、编辑、链接和调试的过程。其主界面如图 5.1 所示。

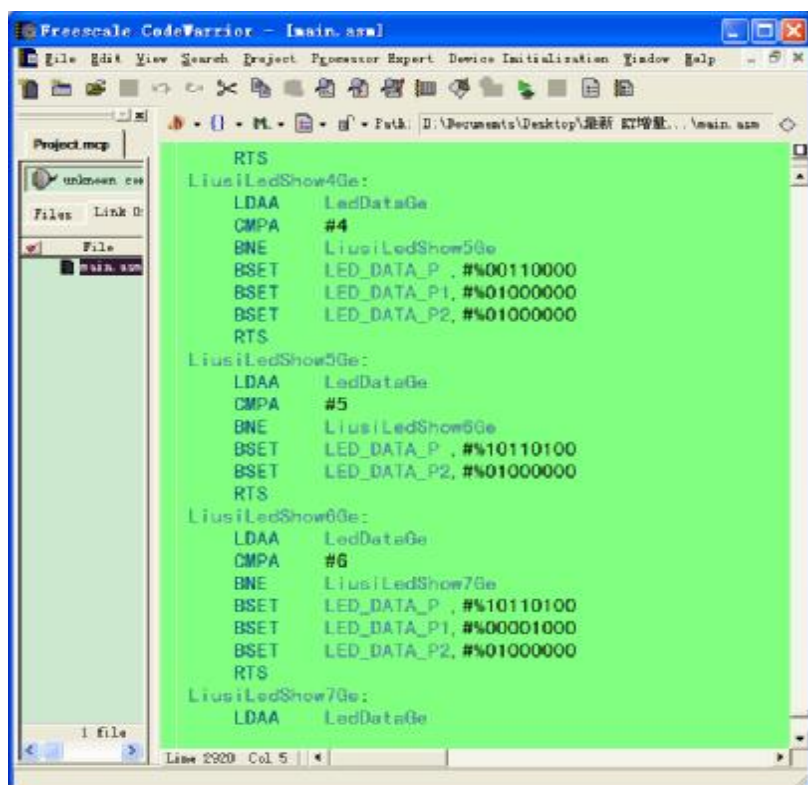


图 5.1 CodeWarrior 主界面

5.2 串口调试软件

为实时监控软件流程，可通过 SCI 将单片机 RAM 或 FLASH 中的数据发送到上位机，上位机通过串口调试软件接收单片机发送的数据。本系统中应用的串口调试工具为串口大师（Techcon），其主要有以下几个功能：

- ü 可灵活选择、设置串行端口及波特率
- ü 发送、接收 16 进制数或 ASCII 字符
- ü 测试串口设备/串口线路

串口调试软件的界面如图 5.2 所示。



图 5.2 串口调试软件 Techcon 界面

第六章 模型车的主要技术参数说明

6.1 改造后的车模基本参数

重量：1.5kg（包含电池）

车长：35cm

车宽：17cm

车高：40cm

6.2 电路功耗，所有电容总容量

电路总功耗：约 30W

所有电容总容量：1500uF

6.3 传感器种类以及个数

CCD 传感器：1 个

测速传感器：1 个

传感器总数：2 个

6.4 除了车模原有的驱动电机、舵机之外伺服电机个数

除了车模原有的驱动电机、舵机之外伺服电机个数为 0。

6.5 赛道信息检测精度、频率

赛道信息检测精度：7mm

赛道信息检测频率：5ms

第七章 结论与展望

7.1 结论与总结

本文从智能车的硬件结构、电路设计、软件实现方面进行了介绍。我们对这项比赛有很大的兴趣，对摄像头的算法还有改进。我们在整个过程中也碰到许多问题与困难，回顾近几个月的忙碌，从以下几个方面进行一下总结。

经过一年的细致工作，我们最终完成智能车的制作。在车模的制作中我们遇到很多困难，如小车在行驶的中途突然停车，起跑线误判，由于我们对小车的硬件和软件认识不够，所以我们阅读了大量的文献资料，制定了详细的工作计划，合理分配工作，提出了许多新的想法和设计思路。根据最终确定的硬件方案，完成了模型车的硬件方案的制作。并通过大量的跑道测试调节算法参数，不断完善控制算法，最终模型车在测试跑道上能稳定完成自主循迹行驶，最终速度也慢慢提升，并完成了技术报告的撰写。

7.2 展望

展望未来，如果以后从事智能汽车方面研究的话，在软硬件方面可以进行更多的尝试。因为实验和摸索是提高成绩的根本，只有经过反复实验才能得到可靠的算法。

(1) 硬件电路设计方面。在调试过程中，由于受到电源管理方面的影响，使控制器系统经常出现复位现象，经过改进后，这种现象消除了。这对于以后做关于控制器方面的工作，积累了很好的经验。

(2) 软件控制算法方面。参考文献有许多关于模糊控制和记忆算法的介绍，可以尝试研究一下。

(3) 智能车的应用前景方面。展望未来，关于智能车的应用可以工作在仓库、码头、工厂或危险、有毒、有害的工作环境里，此外还能担当起无人值守的巡逻、监视物料的运输、消防灭火等任务。

致 谢

在智能车制作期间，我们遇到了许多困难，从最开始的程序烧写到后来的传感器调试，从电路板设计到系统搭建，一个个问题的解决见证了在学习中不断成长整个团队。期间离不开导师的悉心指导。没有导师在思路上的指点，在资源方面的支持，我们将进展缓慢。几个月来，导师渊博的学识、严谨的治学态度、以及给予我们的支持和鼓励将使我们终生难忘，我们所取得的每一点成绩都与导师的热情关怀和精心指导是分不开的。至此报告完成之际，特别向导师致以衷心的感谢和崇高的敬意。

在智能车制作期间，我们涉猎控制、模式识别、传感技术、汽车电子、电气、计算机、机械等多个学科，这次磨练对我们的知识融合和实践动手能力的培养有极大的推动作用，在此要感谢清华大学，感谢他们将这项很有意义的科技竞赛引入中国；也感谢学院教务处对此次比赛的关注和支持，我们的成果离不开学校的大力支持。

参考文献

- [1] 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.
- [2] 刘慧银, 龚光华, 王云飞. Motorola(Freescale)微控制器 MC68HC08 原理及其嵌入式应用 [M]. 北京: 清华大学出版社, 2005. 1.
- [3] 邵贝贝. 嵌入式实时操作系统[uC / OS- II (第 2 版)[M]. 北京. 清华大学出版社. 2004
- [4] 王宜怀, 刘晓升. 嵌入式应用技术基础教程[M]. 北京: 华大学出版社, 2005. 7.
- [5] 余志生汽车理论. 第 4 版. 北京: 机械工业出版社[M], 2006. 5.
- [6] 吴晓燕, 张双选. MATLAB 在自动控制中的应用[M]. 西安: 西安电子科技大学出版社, 2006. 9.
- [7] 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. 5.
- [8] 卓晴, 黄开胜, 邵贝贝. 学做智能车: 挑战“飞思卡尔”杯[M]. 北京: 北京航空航天大学出版社, 2007. 3.
- [9] 陶永华等. 新型 PID 控制及其应用. [M] 机械工业出版社, 1998. 9
- [10] 王京起, 陈慧岩, 郑培. 应用模糊自适应PID和预瞄策略的自主车辆转向控制[J]. 《汽车工程》 2003. 25. 4
- [11] 陈宋, 李立国, 黄开胜. 智能模型车底盘浅析[J] 《电子产品世界》 2006. 11
- [12] 王明顺, 沈谋全. 基于 TPS7350 的智能车电源设计[R] 东北大学信息科学与工程学院
- [13] 智能车制作: <http://www.znczz.com/>
- [14] 第七届全国大学生智能车竞赛官方网站: <http://www.smartcar.au.tsinghua.edu.cn>

附录 A：程序源代码

```
. *****
/

. *****
/

; *    2012 年第七届“飞思卡尔”杯全国大学生智能汽车竞赛
;
; *                湖北汽车工业学院科技学院
;
; *                独眼龙 4.0
;
. *****
/

. *****
/

; #####

; *** Memory Map and Interrupt Vectors
;
. *****
/

RAMStart:          EQU    $00002000

RAMEnd:            EQU    $00003FFF

DFLASHStart:       EQU    $00000400

DFLASHEnd:         EQU    $000013FF

ROM_1400Start:     EQU    $00001400

ROM_1400End:       EQU    $00001FFF

ROM_4000Start:     EQU    $00004000

ROM_4000End:       EQU    $00007FFF

ROM_C000Start:     EQU    $0000C000

ROM_C000End:       EQU    $0000FEFF

PAGE_08Start:      EQU    $00020000

PAGE_08End:        EQU    $00023FFF

PAGE_09Start:      EQU    $00024000
```

PAGE_09End:	EQU	\$00027FFF
PAGE_0AStart:	EQU	\$00028000
PAGE_0AEnd:	EQU	\$0002BFFF
PAGE_0BStart:	EQU	\$0002C000
PAGE_0BEnd:	EQU	\$0002FFFF
PAGE_0CStart:	EQU	\$00030000
PAGE_0CEnd:	EQU	\$000313FF
PAGE_0C_A000Start:	EQU	\$00032000
PAGE_0C_A000End:	EQU	\$00033FFF
PAGE_0EStart:	EQU	\$00038000
PAGE_0EEnd:	EQU	\$0003BFFF
;		
Vsi :	EQU	\$0000FF80
Vportad:	EQU	\$0000FF82
Vatdcompare:	EQU	\$0000FF84
VReserved60:	EQU	\$0000FF86
Vapi :	EQU	\$0000FF88
VI vi :	EQU	\$0000FF8A
VReserved57:	EQU	\$0000FF8C
Vportp:	EQU	\$0000FF8E
VReserved55:	EQU	\$0000FF90
VReserved54:	EQU	\$0000FF92
VReserved53:	EQU	\$0000FF94
VReserved52:	EQU	\$0000FF96
VReserved51:	EQU	\$0000FF98
VReserved50:	EQU	\$0000FF9A

VReserved49:	EQU	\$0000FF9C
VReserved48:	EQU	\$0000FF9E
VReserved47:	EQU	\$0000FFA0
VReserved46:	EQU	\$0000FFA2
VReserved45:	EQU	\$0000FFA4
VReserved44:	EQU	\$0000FFA6
VReserved43:	EQU	\$0000FFA8
VReserved42:	EQU	\$0000FFAA
VReserved41:	EQU	\$0000FFAC
VReserved40:	EQU	\$0000FFAE
Vcantx:	EQU	\$0000FFB0
Vcanrx:	EQU	\$0000FFB2
Vcanerr:	EQU	\$0000FFB4
Vcanwkup:	EQU	\$0000FFB6
Vfl ash:	EQU	\$0000FFB8
Vfl ashfd:	EQU	\$0000FFBA
Vspi 2:	EQU	\$0000FFBC
Vspi 1:	EQU	\$0000FFBE
VReserved31:	EQU	\$0000FFC0
Vsci 2:	EQU	\$0000FFC2
VReserved29:	EQU	\$0000FFC4
Vcpmupl l l ck:	EQU	\$0000FFC6
Vcpmuocsns:	EQU	\$0000FFC8
VReserved26:	EQU	\$0000FFCA
VReserved25:	EQU	\$0000FFCC
Vportj :	EQU	\$0000FFCE

VReserved23:	EQU	\$0000FFD0
Vatd:	EQU	\$0000FFD2
Vsci1:	EQU	\$0000FFD4
Vsci0:	EQU	\$0000FFD6
Vspi0:	EQU	\$0000FFD8
Vtimpaie:	EQU	\$0000FFDA
Vtimpaovf:	EQU	\$0000FFDC
Vtimovf:	EQU	\$0000FFDE
Vtimch7:	EQU	\$0000FFE0
Vtimch6:	EQU	\$0000FFE2
Vtimch5:	EQU	\$0000FFE4
Vtimch4:	EQU	\$0000FFE6
Vtimch3:	EQU	\$0000FFE8
Vtimch2:	EQU	\$0000FFEA
Vtimch1:	EQU	\$0000FFEC
Vtimch0:	EQU	\$0000FFEE
Vrti:	EQU	\$0000FFF0
Virq:	EQU	\$0000FFF2
Vxirq:	EQU	\$0000FFF4
Vswi:	EQU	\$0000FFF6
Vtrap:	EQU	\$0000FFF8
Vcop:	EQU	\$0000FFFA
Vclmon:	EQU	\$0000FFFC
Vreset:	EQU	\$0000FFFE

```

. ****
/

. ****
/

. ****
/

. ****
/

; * 常量定义          MYRAM

. ****
/

KBUDown_D            EQU      DDRJ          ;发车键

KBI ncrease_D        EQU      DDRJ

KBDDecrease_D        EQU      DDRJ


KBUDown_P            EQU      PTJ           ;发车键

KBI ncrease_P        EQU      PTJ

KBDDecrease_P        EQU      PTJ


KBUDownBi te        EQU      %00100000    ; 5          ;位选

KBI ncreaseBi te    EQU      %01000000    ; 6          ;增加

KBDDecreaseBi te    EQU      %00010000    ; 4          ;减少

. ****
/

;          Variable in RAM

. ****
/

          ORG      RAMStart


LEDShowFlag          RMB      1

LED_Data1            RMB      1

LED_Data2            RMB      1

LED_Data3            RMB      1

```

LED_Data4	RMB	1	
LED_Counter	RMB	1	
LED_G6	RMB	1	
LedData	RMB	2	
BendSpeedLED	RMB	2	
HighSpeedLED	RMB	2	
LowSpeedLED	RMB	2	
STEERING_CENTER_LED	RMB	2	
UI tra_high_speedLED	RMB	2	

,*			

PIDTimer	RMB	1	
PIDCnt5ms	RMB	1	
;-----			
SpPIDInput	RMB	2	;采集到的反馈量
SpPIDaim	RMB	2	;调节的目标
SpPIDaim0	RMB	2	
SpPIDaim_Max	RMB	2	
SpPIDaim_Min	RMB	2	
SpPIDaim_Accumulate	RMB	2	
;-----			
SpKP	RMB	2	
SpKI	RMB	2	

SpKD	RMB	2	
ResponseTime		RMB	2
Speedupreaction		RMB	2
SpPIDOutput	RMB	2	
SpPIDLastValue		RMB	2
SpPIDDeltaU	RMB	2	
SpPIDDeltaUMAX		RMB	2
SpPIDDeltaUMIN		RMB	2
SpPIDOutputMAX		RMB	2
SpPIDOutputMIN		RMB	2
SpPIDError	RMB	2	
SpPIDError1	RMB	2	
SpPIDError2	RMB	2	
SpPIDErrorAbs		RMB	2
SpPIDErrorMax		RMB	2
SpPIDErrorMin		RMB	2
SpAI lowError		RMB	2
SHANGMainLinenumCorrectEnterFLAG		RMB	1
XIAMainLinenumCorrectEnterFLAG		RMB	1
SHANGNearErr		RMB	1
XIANearErr		RMB	1
SHANGArarryNUM		RMB	1
XIAArarryNUM		RMB	1
SHANGFirstPosition		RMB	1

XIAFirstPosition	RMB	1
SHANGArrayAverage	RMB	1
XIAArrayAverage	RMB	1
SHANGPathFlag	RMB	1
XIAPathFlag	RMB	1

```

, ****
/

```

```

, *
/

```

```

, ****
/

```

```

ORG ROM_4100Start

```

Entry:

```

LDS #RAMEnd+1

```

```

, ****
/

```

```

, *
/

```

```

, ****
/

```

```

*****

```

```

, *
/

```

```

*****

```

```

MOVB    #%00000000, ATDOCTL0

```

```

MOVB    #%00000000, ATDOCTL1

```

```

MOVB    #%01011000, ATDOCTL2

```

```

MOVB    #%10001000, ATDOCTL3

```

```

MOVB    #%00000000, ATDOCTL4

```

```

*****

```

, *
,

BSET PWMPOL, #%00000010

BSET PWMCLK, #%00000010

BSET PWMCTL, #%00010000

MOVB #5, PWMSCLA

MOVW #50000, PWMPER0

MOVW STEERING_CENTER, PWMDTY0

BSET PWME, #%00000010

MOVB #\$FF, LEDShowFlag

CLR LED_Counter

CLR LED_G6

CLR LED_Data1

CLR LED_Data2

CLR LED_Data3

CLR LED_Data4

MOVB #%11110000, DDRD

MOVB #%10000000, DDRJ

MOVB #%00001111, DDRM

MOVB #%11100000, DDRS

BSET Zhi Shi Deng_P, #Zhi Shi Deng_Pin

```
MOVW    #4321, LedData
LDD      #10000
JSR      CpuDelay100US
        JSR      System_Set ; 系统设置
LDD      #20000
JSR      CpuDelay100US
MOVB     #$00, LEDShowFlag; ///数码管显示结束标志位
```

```
InstrumentSetting_01:
```

```
; LBRA  InstrumentSetting_03
; MOVW  STEERING_CENTER_LED, LedData
```

MOVW #10, FarthestSteerLine

MOVW FarthestSteerLine, LedData

; LDAB Set_Num1

; SEX B, D

; STD LedData

CheckSetting_01:

BRCLR KBSHANGXIA_P, #KBSHANGXIABi te, InstrumentSetting_01End ; BRCLR

BRCLR KBI ncrease_P, #KBI ncreaseBi te, IncreaseSetting_01 ; BRCLR

BRSET KBDecrease_P, #KBDecreaseBi te, CheckSetting_01 ; BRSET

DecreaseSetting_01:

JSR LCD_del ay

BRSET KBDecrease_P, #KBDecreaseBi te, CheckSetting_01

JSR LCD_del ay100ms

JSR LCD_del ay100ms

JSR LCD_del ay100ms

LDD LedData

SUBD #1 ; ????????????????????????

STD LedData

; STD PWMDTY0 ; 转向中值

BRA CheckSetting_01

IncreaseSetting_01:

JSR LCD_del ay

BRSET KBI ncrease_P, #KBI ncreaseBi te, CheckSetting_01

JSR LCD_del ay100ms

```

JSR      LCD_delay100ms

JSR      LCD_delay100ms

LDD      LedData

ADDD     #1

STD      LedData

; STD      PWMDTY0          ;转向中值

BRA      CheckSetting_01

InstrumentSetting_01End:

JSR      LCD_delay

BRSET    KBSHANGXIA_P, #KBSHANGXIABite, CheckSetting_01

JSR      LCD_delay100ms

JSR      LCD_delay100ms

JSR      LCD_delay100ms

LDD      LedData

; STAB     Set_Num1

; MOVW     LedData, STEERING_CENTER_LED

STAB     FarthestSteerLine

```

```

*****

```

```

, *          , *
/          /

```

```

*****

```

```

InstrumentSetting_02:

; LBRA     InstrumentSetting_03

MOVW     STEERING_CENTER_LED, LedData

```

```
; LDAB    Set_Num1  
  
; SEX     B,D  
  
; STD     LedData
```

CheckSetting_02:

```
BRCLR     KBSHANGXIA_P, #KBSHANGXIABi te, InstrumentSetting_02End    ; BRCLR  
  
BRCLR     KBI ncrease_P, #KBI ncreaseBi te, IncreaseSetting_02      ; BRCLR  
  
BRSET     KBDecrease_P, #KBDecreaseBi te, CheckSetting_02           ; BRSET
```

DecreaseSetting_02:

```
JSR       LCD_del ay  
  
BRSET     KBDecrease_P, #KBDecreaseBi te, CheckSetting_02  
  
JSR       LCD_del ay100ms  
  
JSR       LCD_del ay100ms  
  
JSR       LCD_del ay100ms  
  
LDD       LedData  
  
SUBD     #1    ; ??????????????????????  
  
STD       LedData  
  
STD       PWMDTY0          ; 转向中值  
  
BRA       CheckSetting_02
```

IncreaseSetting_02:

```
JSR       LCD_del ay  
  
BRSET     KBI ncrease_P, #KBI ncreaseBi te, CheckSetting_03  
  
JSR       LCD_del ay100ms  
  
JSR       LCD_del ay100ms  
  
JSR       LCD_del ay100ms  
  
LDD       LedData
```

```

    ADDD    #1

    STD     LedData

    STD     PWMDTY0          ; 转向中值

    BRA     CheckSetting_02

InstrumentSetting_02End:

    JSR     LCD_delay

    BRSET   KBSHANGXIA_P, #KBSHANGXIABite, CheckSetting_02

    JSR     LCD_delay100ms

    JSR     LCD_delay100ms

    JSR     LCD_delay100ms

    LDD     LedData

; STAB     Set_Num1

; MOVW     LedData, STEERING_CENTER_LED

*****

*      ;;

*****

InstrumentSetting_03:

;   LBRA   InstrumentSetting_04

    MOVW    SpPIDaim_Max, LedData          ; DATA1=> A 表示最高速度

    JSR     LCD_delay100ms

    JSR     LCD_delay100ms

    JSR     LCD_delay100ms

    JSR     LCD_delay100ms

```

CheckSetting_03:

```
BRCLR    KBSHANGXIA_P, #KBSHANGXIABi te, InstrumentSetting_03End      ; BRCLR
BRCLR    KBI ncrease_P, #KBI ncreaseBi te, IncreaseSetting_03          ; BRCLR
BRSET     KBDecrease_P, #KBDecreaseBi te, CheckSetting_03              ; BRSET
```

DecreaseSetting_03:

```
JSR      LCD_del ay
BRSET     KBDecrease_P, #KBDecreaseBi te, CheckSetting_03
JSR      LCD_del ay100ms
JSR      LCD_del ay100ms
JSR      LCD_del ay100ms
LDD      LedData
SUBD     #2      ; ??????????????????????
STD      LedData
BRA      CheckSetting_03
```

IncreaseSetting_03:

```
JSR      LCD_del ay
BRSET     KBI ncrease_P, #KBI ncreaseBi te, CheckSetting_03
JSR      LCD_del ay100ms
JSR      LCD_del ay100ms
JSR      LCD_del ay100ms
LDD      LedData
ADDD     #2      ; ??????????????????????
STD      LedData
BRA      CheckSetting_03
```

InstrumentSetting_03End:

```
JSR      LCD_del ay
```

```

BRSET    KBSHANGXIA_P, #KBSHANGXIABi te, CheckSetting_03

JSR      LCD_del ay100ms

JSR      LCD_del ay100ms

JSR      LCD_del ay100ms

MOVW     LedData, SpPIDAi m_Max   ; ????????????????

*****

*

*****

InstrumentSetting_04:

;   LBRA   InstrumentSetting_05

MOVW     SpPIDAi m_Mi n, LedData

CheckSetting_04:

BRCLR    KBSHANGXIA_P, #KBSHANGXIABi te, InstrumentSetting_04End

BRCLR    KBI ncrease_P, #KBI ncreaseBi te, IncreaseSetting_04

BRSET    KBDecrease_P, #KBDecreaseBi te, CheckSetting_04

DecreaseSetting_04:

JSR      LCD_del ay

BRSET    KBDecrease_P, #KBDecreaseBi te, CheckSetting_04

JSR      LCD_del ay100ms

JSR      LCD_del ay100ms

LDD      LedData

SUBD     #1   ; ??????????????????????

STD      LedData

BRA      CheckSetting_04

IncreaseSetting_04:

JSR      LCD_del ay

```

```

BRSET    KBI ncrease_P, #KBI ncreaseBi te, CheckSetti ng_04

JSR      LCD_del ay100ms

JSR      LCD_del ay100ms

LDD      LedData

ADDD     #1      ; ??????????????????????

STD      LedData

BRA      CheckSetti ng_04

```

InstrumentSetting_04End:

```

JSR      LCD_del ay

BRSET    KBSHANGXI A_P, #KBSHANGXI ABi te, CheckSetti ng_04

JSR      LCD_del ay100ms

JSR      LCD_del ay100ms

MOVW     LedData, SpPI DAi m_Mi n ;

```

, *
/

InstrumentSetting_05:

```

MOVW     #0, CCDIF_I denti fy_startl ine

MOVW     CCDIF_I denti fy_startl ine, LedData

```

CheckSetti ng_05:

```

BRCLR    KBSHANGXI A_P, #KBSHANGXI ABi te, InstrumentSetti ng_05End      ; BRCLR

```

BRCLR KBI ncrease_P, #KBI ncreaseBi te, I ncreaseSetti ng_05 ; BRCLR

BRSET KBDecrease_P, #KBDecreaseBi te, CheckSetti ng_05 ; BRSET

DecreaseSetti ng_05:

JSR LCD_del ay

BRSET KBDecrease_P, #KBDecreaseBi te, CheckSetti ng_05

JSR LCD_del ay100ms

JSR LCD_del ay100ms

JSR LCD_del ay100ms

LDD LedData

SUBD #1 ; ????????????????????????

STD LedData

BRA CheckSetti ng_05

I ncreaseSetti ng_05:

JSR LCD_del ay

BRSET KBI ncrease_P, #KBI ncreaseBi te, CheckSetti ng_05

JSR LCD_del ay100ms

JSR LCD_del ay100ms

JSR LCD_del ay100ms

LDD LedData

ADDD #1

STD LedData

BRA CheckSetti ng_05

I nstrumentSetti ng_05End:

JSR LCD_del ay

BRSET KBSHANGXI A_P, #KBSHANGXI ABi te, CheckSetti ng_05

JSR LCD_del ay100ms

JSR LCD_delay100ms

JSR LCD_delay100ms

LDD LedData

STAB CCDIF_Identify_startline

System_Set_End:

LCD_delay100ms:

LDD #1000

JSR CpuDelay100US

RTS

LCD_delay:

LDD #500

JSR CpuDelay100US

RTS

LED_ShowMAP:

FDB LED_Show0
FDB LED_Show1
FDB LED_Show2
FDB LED_Show3
FDB LED_Show4
FDB LED_Show5
FDB LED_Show6
FDB LED_Show7
FDB LED_Show8
FDB LED_Show9
FDB LED_ShowA
FDB LED_ShowB
FDB LED_ShowC
FDB LED_ShowD
FDB LED_ShowE
FDB LED_ShowF

LED_Show0:

BSET PTM, #%00001011; PTM
BSET PTJ, #%10000000; PTJ
BSET PTS, #%00100000
BSET PORTD, #%10000000; PTD --->0(F)

```

    JMP    portHi ntEND

LED_Show1:

    BSET   PTS, %%00100000    ; PTS

    BSET   PORTD, %%10000000    ; PTD --->1

    JMP    portHi ntEND

LED_Show2:

    BSET   PTM, %%00001010    ; PTM

    BSET   PTS, %%00100000    ; PTS

    BSET   PTJ, %%10000000    ; PTJ

    BSET   PORTD, %%01000000    ; PTD --->2

    JMP    portHi ntEND

LED_Show3:

    BSET   PTM, %%00001010    ; PTM

    BSET   PTS, %%00100000    ; PTS

    BSET   PORTD, %%11000000    ; PTD --->3

    JMP    portHi ntEND

LED_Show4:

    BSET   PTM, %%00000001    ; PTM

    BSET   PTS, %%00100000    ; PTS

    BSET   PORTD, %%11000000    ; PTD --->4

    JMP    portHi ntEND

LED_Show5:

    BSET   PTM, %%00001011    ; PTM

    BSET   PORTD, %%11000000    ; PTD --->5

    JMP    portHi ntEND

LED_Show6:

```

```
BSET PTM, %%00001011 ;PTM

BSET PTJ, %%10000000 ;PTJ

BSET PORTD, %%11000000 ;PTD --->6

JMP portHi ntEND
```

LED_Show7:

```
BSET PTM, %%00000010 ;PTM

BSET PTS, %%00100000 ;PTS

BSET PORTD, %%10000000 ;PTD --->7

JMP portHi ntEND
```

LED_Show8:

```
BSET PTM, %%00001011 ;PTM

BSET PORTD, %%11000000 ;PTD

BSET PTS, %%00100000 ;PTS

BSET PTJ, %%10000000 ;PTJ --->8

JMP portHi ntEND
```

LED_Show9:

```
BSET PTM, %%00000011 ;PTM

BSET PTS, %%00100000 ;PTS

BSET PORTD, %%11010000 ;PTD --->9

JMP portHi ntEND
```

LED_ShowA:

```
BSET PTM, %%00000011 ;PTM

BSET PTS, %%00100000 ;PTS

BSET PORTD, %%11000000 ;PTD

BSET PTJ, %%10000000 ;PTJ --->A

JMP portHi ntEND
```

LED_ShowB:

```
BSET   PTM, %%00001011 ;PTM
BSET   PTS, %%00100000 ;PTS
BSET   PTJ, %%10000000 ;PTJ
BSET   PORTD, %%11000000 ;PTD --->B
JMP     portHi ntEND
```

LED_ShowC:

```
BSET   PTM, %%00001011 ;PTM
BSET   PTJ, %%10000000 ;PTJ --->C
JMP     portHi ntEND
```

LED_ShowD:

```
BSET   PTM, %%00001011;PTM
BSET   PTS, %%00100000;PTS
BSET   PTJ, %%10000000;PTJ
BSET   PORTD, %%10000000;PTD --->D
JMP     portHi ntEND
```

LED_ShowE:

```
BSET   PTM, %%00001011 ;PTM
BSET   PTJ, %%10000000 ;PTJ
BSET   PORTD, %%01000000 ;PTD --->E
JMP     portHi ntEND
```

LED_ShowF:

```
BSET   PTM, %%00001011 ;PTM
BSET   PTJ, %%10000000 ;PTJ
BSET   PORTD, %%01000000 ;PTD --->E
```

```
JMP    portHi ntEND
```

```
LED_ShowG:
```

```
    BSET    PTM, %%00001011    ;PTM
```

```
    BSET    PTJ, %%10000000    ;PTJ
```

```
    BSET    PORTD, %%01000000    ;PTD --->E
```

```
JMP    portHi ntEND
```

```
*****
```

```
    LDAA    LEDShowFl ag
```

```
    CMPA    #$FF
```

```
    LBNE    portHi ntEND
```

```
LED_Start:
```

```
;=====
```

```
    LDD    LedData
```

```
    LDX    #1000
```

```
    IDIV
```

```
    PSHD
```

```
    EXG    X, A
```

```
    STAA    LED_Data1
```

```
    PULD
```

```
    LDX    #100
```

```
    IDIV
```

```
    PSHD
```

```
    EXG    X, A
```

```

    STAA    LED_Data2

    PULD

    LDX     #10

    IDIV

    PSHD

    EXG     X, A

    STAA    LED_Data3

    PULD

    EXG     D, A

    STAA    LED_Data4

; =====

    INC     LED_Counter

    LDAA    LED_Counter

    CMPA    #$01

    LBNE    NotSeg1

Segment1: ;

    BCLR    PTM, #%00001111

    BCLR    PTJ, #%10000000

    BCLR    PTS, #%00100000

    BCLR    PORTD, #%11000000


    BCLR    PORTD, #%00010000

    BSET    PTS, #%10000000

    BSET    PTS, #%01000000

    BSET    PORTD, #%00100000

```

```
MOVB    LED_Data1, LED_G6

LDX      #LED_ShowMAP

LDAB     LED_G6

LSLB

LDY      B, X

JMP      Y

LBRA     SegmentEND0
```

NotSeg1:

```
LDAA     LED_Counter

CMPA     #02;

BNE      NotSeg2
```

Segment2:

```
BCLR     PTM, #%00001111

BCLR     PTJ, #%10000000

BCLR     PTS, #%00100000

BCLR     PORTD, #%11000000


BSET     PORTD, #%00010000

BCLR     PTS, #%10000000

BSET     PTS, #%01000000

BSET     PORTD, #%00100000
```

```
MOVW    LED_Data2, LED_G6

        LDX     #LED_ShowMAP

        LDAB    LED_G6

        LSLB

        LDY     B, X

        JMP     Y

        LBRA    SegmentEND0

NotSeg2:

        LDAA    LED_Counter

        CMPA    #3

        BNE     NotSeg3

Segment3:

        BCLR    PTM, %%00001111

        BCLR    PTJ, %%10000000

        BCLR    PTS, %%00100000

        BCLR    PORTD, %%11000000


        BSET    PORTD, %%00010000

        BSET    PTS, %%10000000

        BCLR    PTS, %%01000000

        BSET    PORTD, %%00100000


        MOVW    LED_Data3, LED_G6
```

```
LDX    #LED_ShowMAP
LDAB    LED_G6
LSLB
LDY     B, X
JMP     Y
LBRA     SegmentEND0
```

NotSeg3:

```
LDAA    LED_Counter
CMPA    #4
BNE     SegmentEND
BCLR    PTM, %%00001111
BCLR    PTJ, %%10000000
BCLR    PTS, %%00100000
BCLR    PORTD, %%11000000

BSET    PORTD, %%00010000
BSET    PTS, %%10000000
BSET    PTS, %%01000000
BCLR    PORTD, %%00100000
MOVB    LED_Data4, LED_G6
LDX     #LED_ShowMAP
LDAB    LED_G6
LSLB
LDY     B, X
JMP     Y
```

LBRA SegmentEND0

SegmentEND:

CLR LED_Counter

CLR LED_G6

SegmentEND0:

Fi l l Data:

PSHA

Fi l l Data1:

STAB 1,X+

DBNE Y,Fi l l Data1

PULA

RTS

Real time_Record_Ini ti on: ;

MOVW #\$00,Wai tTimeCOUNT

CLR RecordTemp

MOVW #\$00,RecordCount

CLR RecordCountFI ag

MOVW #\$00,Non_RecordCount

MOVW #RecordRAM,RecordRAMAddr

MOVW #RecordRAM+5,RecordRAMEndAddr

RTS

CCD_My_Fi l ter_Array_Ini ti on:

LDX #SHANG_CCDI nterpol ation_Aryy

LDAA #50

XXX

CCD_My_Filter_Array_Initiation_A:

```
        MOVB      #$FF, 1, X+  
        DBNE      A, CCD_My_Filter_Array_Initiation_A  
        RTS
```

CCD_Steer_Initiation:

```
        LDX      #OffsetDistanceArrayy  
        LDAA     #25
```

CCD_Steer_Initiation_A:

```
        MOVB     #22, 1, X+  
        DBNE     A, CCD_Steer_Initiation_A  
        RTS
```

, *
,

; =====

SCI_TX:

```
        BRCLR    SCI_OSR1, #$80, *  
        STAB     SCI_ODRL  
        RTS
```

, *
/

SCI_TX_W:

BRCLR SCI0SR1, #80, *

STAA SCI0DRL

BRCLR SCI0SR1, #80, *

STAB SCI0DRL

RTS

, *
/

SCI_RX:

BRCLR SCI0SR1, #20, *

LDAB SCI0DRL

RTS

, *
/

SCI_RX_W:

BRCLR SCI0SR1, #20, *

LDAA SCI0DRL

BRCLR SCI0SR1, #20, *

LDAB SCI0DRL

RTS
