

第五届“飞思卡尔”杯全国大学生 智能汽车竞赛

技 术 报 告

学 校： 乐山师范学院

队伍名称： 明星电缆 3 队

参赛队员： 李富钢

李升飞

赵小平

带队教师： 童强 祝家雄

关于技术报告和学术论文使用授权的说明

本人完全了解第五届“飞思卡尔”杯全国大学生智能汽车邀请赛关保留、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

第一章 引言.....	1
1.1 概述.....	1
1.2 本文结构.....	1
第二章 系统设计.....	2
2.1 系统方案介绍.....	2
2.2 智能车系统总体结构.....	2
2.2.1 中央处理单元.....	3
2.2.2 光电检测模块.....	3
2.2.3 电源模块.....	3
2.2.4 速度检测模块.....	3
2.2.5 舵机驱动模块.....	3
2.2.6 电机驱动模块.....	3
第三章 机械结构设计.....	4
3.1 车模选择.....	4
3.2 车模安装.....	4
3.2.1 后轮调整.....	4
3.2.2 前轮调整.....	4
3.2.4 车身重心调整.....	4
3.2.5 智能车重量调整.....	4
3.2.6 其它机械结构的调整.....	5
3.3 传感器的设计安装.....	5
3.3.1 传感器的选择.....	5
3.3.2 光电传感器的原理.....	6
3.3.3 光电传感器的设计.....	7

3.4 智能车建模及尺寸参数.....	7
第四章 电路设计.....	9
4.1 电源模块设计.....	9
4.2 电机驱动模块设计.....	10
4.3 舵机控制模块设计.....	10
4.4 车速检测模块设计.....	10
4.4.1 测速方法研究.....	10
4.4.2 M/T 法测速测速原理.....	11
4.5 坡道控制模块设计.....	11
第五章 软件系统设计.....	13
5.1 软件系统总体设计.....	13
5.2 软件各功能模块设计.....	14
5.2.1 时钟模块.....	14
5.2.2 ECT 模块.....	14
5.2.3 AD 转换模块.....	15
5.3 路径识别与算法.....	15
5.4 弯道控制及策略.....	16
5.5 速度与转角控制策略与算法.....	16
第六章 开发制作与调试.....	18
6.1 系统开发工具.....	18
6.2 调试过程.....	18
6.2.1 硬件调试.....	18
6.2.2 软件调试.....	19
第七章 智能车主要技术参数说明.....	20
7.1 智能车基本参数.....	20
7.2 电路部分参数.....	20

7.3 除车模原有的驱动电机、舵机外伺服电机数量.....	20
第八章 总结.....	21
参考文献.....	22
附录：	I

第一章 引言

1.1 概述

自“飞思卡尔”杯全国大学生电子设计竞赛 2006 年举办第一届以来，越来越多地受到学校和企业各界的关注。今年三月报名参加第五届智能汽车竞赛，本小队成员学习了“飞思卡尔”单片机控制器、传感技术、汽车机械构造、模式识别等各方面知识，积累了大量理论知识与实践经验。

本智能车系统采用 MC9S12XS128 单片机作为控制核心，接收并处理激光传感器检测赛道反馈信息，采用两个舵机分别对赛车转向进行控制及实现光电传感器跟踪赛道“摇头”。并采用旋转编码器对智能车速度进行实时检测，同时运用 PID 算法实现赛车速度闭环控制。另外，智能车系统采用独立安装在车前端的 ST188 光电传感器检测坡道路径。小车速度提升的关键是转角要和速度匹配的非常好，我们采用现场整定的办法获取智能小车转角和速度的最佳匹配参数，实现小车的最优控制。在核心控制单元的协调控制下，该智能车能够快速稳定的跑完全程。

1.2 本文结构

本文共八个章节，第一章为引言，简明的讲述全国大学生智能汽车竞赛的情况并对本技术报告的结构作简单介绍；第二章至第七章为本文的主体，详细介绍了本小组的参赛车模的机械、硬件电路以及软件的设计与实现，并对智能车的开发和调试作了说明；第八章为总结，对本技术报告的总体进行简要说明，以及本小队在智能车竞赛准备过程中的一些心得体会。

第二章 系统设计

2.1 系统方案介绍

本参赛小队的智能车系统采用“飞思卡尔”16位单片机MC9S12XS128作为核心控制单元，由安装在智能车上部的数个光电传感器接收从比赛赛道反馈回来的信号并传到核心控制单元单片机中，由智能车“大脑”接收并处理从传感器反馈的信号后，PWM模块产生相应的PWM波形，通过输出不同占空比控制智能车的转向舵机与直流电机，以实现控制智能车按照规定赛道行驶。

为了使智能车能够更加快速稳定的行驶，智能车“大脑”S12单片机必须把对赛道路径的判断、转向舵机的角度控制以及对直流电机的控制紧密的联系在一起。不论是某一路采集信号的误判还是单片机对转向伺服电机控制的失当，都会引起智能车在行驶过程中产生抖动甚至偏离赛道。所以，对智能车总系统设计的合理性和科学性就显得尤为重要。

2.2 智能车系统总体结构

智能车系统总体结构如图 2.1

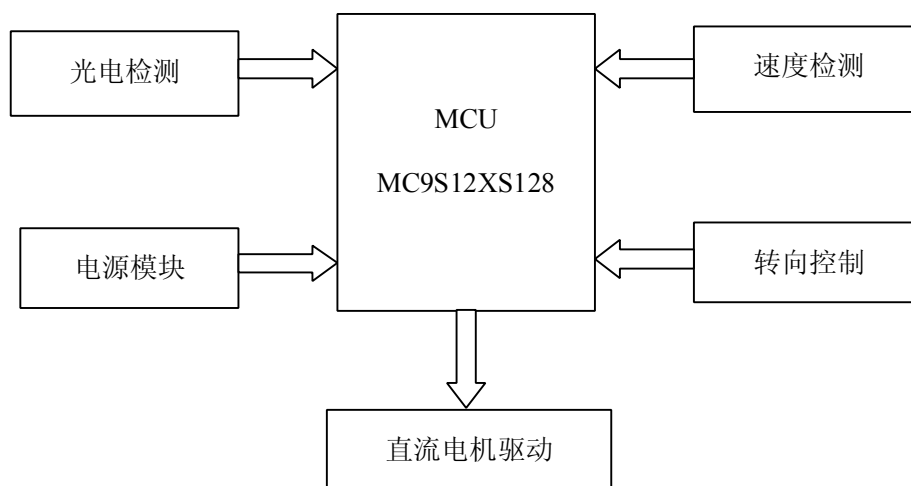


图 2.1 系统总体结构框图

2.2.1 中央处理单元

本智能车系统采用 **freescale** 单片机 **MC9S12XS128** 作为核心控制芯片,对采集的信号进行分析和处理,再根据信号对直流电机和转向舵机进行有效控制。**MC9S12XS128** 单片机 CPU 属于增强型 16 位 CPU,片内总线时钟最高 40 MHz;其片内具有 8 KB RAM、128 KB 程序闪存、2 KB 数据闪存;脉宽调制模块(PWM)可设置成 4 路 8 位或者 2 路 16 位,逻辑时钟选择频率宽;具有 1 个 16 路 12 位精度 A/D 转换器;另外片内还具有增强型捕捉定时器。完全可以胜任大量的数据处理和控制工作。

2.2.2 光电检测模块

利用光电方式对比赛赛道信息进行采集并反馈给中央处理单元,由中央处理单元对采集数据进行处理。

2.2.3 电源模块

为各个电路模块提供稳定的直流电源,保证各个模块稳定工作。

2.2.4 速度检测模块

对智能车的实时速度检测并反馈,以实现智能车速度的闭环控制,以便调整智能车在直到和弯道上的行驶速度,使其能够平稳快速的跑完全程。

2.2.5 舵机驱动模块

对智能车上的舵机进行驱动,以达到快速准确的控制智能车的行驶方向及传感器准确随赛道路径“摇头”。

2.2.6 电机驱动模块

对直流电机进行驱动,控制赛车的速度。

第三章 机械结构设计

3.1 车模选择

根据大赛的要求，我们选择 B 型车模作为智能车制作的基础车模。B 型车模采用的是国内厂商生产的 1:16 的电动越野遥控车的底盘部分，其特点为四轮驱动，四轮独立悬挂，相比于往年比赛采用的 A 型车模结构上复杂程度有所增加。同时可调整参数也有所增加，而通过调整这些参数，车辆的性能提升有明显的效果。

3.2 车模安装

在严格遵守比赛对车模所做的要求的前提下，按照说明书所说的步骤对车模进行安装，并根据实际条件进行了一定的改装。

3.2.1 后轮调整

按照车辆运动学理论，增大后轮距，能够提高智能车稳定性并减小侧滑。因此我们增了后轮距使智能车更稳定的跑完全程。

3.2.2 前轮调整

前轮的中心位置对智能车转向有极其重要作用。为了保证智能车在前轮不人为打偏的情况下按照直线行驶，将前轮调整成外“八”字型，保证赛车在直线赛道上不跑偏。

3.2.4 车身重心调整

车身重心位置的前后调整，对智能车行驶性能有很大的影响。按照车辆运动学理论，车身重心前移会增加转向，但会降低转向的灵敏度，同时降低后轮的抓地力；重心后移会减少转向，但会增大转向灵敏度，后轮抓地力也会增加。因而调整合适的车体重心，让智能车更加适应跑道是很关键的。根据实际调试经验，鉴于当前舵机响应迟缓，因此，需要将车的中心前移，增加转向性能。

3.2.5 智能车重量调整

车模在整个竞赛过程中，加速加速度和减速加速度的大小将直接决定车模

在全程中的平均速度，所以，提高加速度对全程平均速度的提高有重要作用。在硬件系统确定的情况下，欲提高车模的加速度，最直接的办法就是降低车模的重量。我们组在减轻重量方面的操作原则为在不违反比赛规则和不影响正常行驶的情况下，减轻所用材料的重量，拆除多余零件。在电路板的设计过程中，为了减少车体的重量，在保证正常工作的情况下，采用最简单的电路和最少的元器件。在布局上，使用的双面紧凑布局的方法，最大限度减小电路板的面积，减轻重量。

3.2.6 其它机械结构的调整

另外，在 B 型模型车的机械结构方面还有很多可以改进的地方，比如说悬架、底盘、车身高度等。B 型车模的底盘材质较柔软，对于复杂多变的赛道以及高速行驶的智能车，容易发生车体自身的弹性形变。所以本小队在 B 型车模的底盘下加了一层硬质塑料板对车模底盘进行加固。

此外，我们还对模型车的前后悬架弹簧的预紧力进行调节，选用不同弹性系统的弹簧等方法进行了改进，并且对车身高度，进行了相应的改进和调整，均取得了不错效果。

3.3 传感器的设计安装

3.3.1 传感器的选择

方案一：采用 CCD/CMOS 摄像头，CCD/CMOS 摄像头的优点是检测前瞻距离大、检测范围宽、检测道路参数多；缺点是电路设计复杂，需要视频信号同步分离，且工作电压高于电池电压，需要升压电路，加大电源的损耗，而且本系统采用的处理器是 25MHz 的单片机，处理速度有限，并且 CCD 采集一帧图像的时间最快为 50ms，对于一个速度要求很高的系统，采集时间过长，还易受外界干扰，软件计算量大。

方案二：采用电磁线磁场传感器检测电磁感应磁场的方法。竞赛车模需要能够通过自动识别赛道中心线位置处由通有 100mA 交变电流的导线所产生的电磁场进行路径检测。采用此方法检测赛道的方法的优点是传感器设计电路实现简单，体积小。但缺点是磁场传感器检测抗干扰能力小，环境中存在诸多电磁干扰。对智能车的平稳行驶有一定影响。

方案三：使用光电传感器。通过发光管发射激光照射跑道，跑道表面与中心黑线具有不同的反射强度，利用光电接收管可以检测到反射回来的不同强度

的光信息。这种检测方法具有较高的可靠性和稳定性，且电路设计比较简单，检测信息快，而且单片机处理信息更加方便。但这种方法对道路参数检测分辨率，易受到外界光线的干扰。

光电传感器分两种：

(1) 利用红外发光管发出红外光线，接收管接收不同材质反射的不同信息。例如单光束反射取样式 ST188。红外发光管缺点是其收发受光线影响大，前瞻小，导致控制核心对赛道的预处理能力较差。

(2) 利用激光发光管发出激光，接收管接收不同材质反射的不同信息。其优点是前瞻较大，接收管有效接收激光管发出的光线距离较红外线更长，传感器的视野更加广阔，控制核心对赛道的预处理能力更强。

通过对上述三个方案的比较，结合自身的实际情况，我们采用方案三第二种，即利用光电方式，通过激光发光管出激光，接收管接收反射回来的不同强度的光信息来检测路径的方案。对于坡道赛道而言，较长的发射光光线容易使智能车因丢失黑线而导致“眼盲”。所以在坡道赛道上，本智能车采用 ST188 光电传感器对坡道赛道进行独立检测，减少软件算法的复杂性。

3.3.2 光电传感器的原理

光电传感器检测路面信息的原理是由发射管发射一定波长的激光，经地面反射到接收管。如图 3.1，由于在黑色和白色上反射系数不同，在黑色上大部分光线被吸收，而白色上可以反射回大部分光线，所以接收到的反射光强是不一样的，进而导致接收管的特性曲线发生变化程度不同，而从外部观测可以近似认为接收管两端输出电阻不同，进而经分压后的电压就不一样，就可以将黑白路面区分开来。

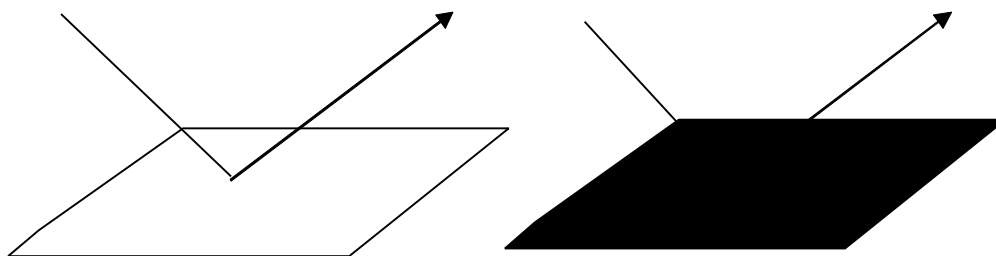


图 3.1 光电传感器原理

3.3.3 光电传感器的设计

激光传感器与普通的光电传感器原理都是一样，但是其前瞻能力远大于普通的光电传感器，可以达到 40-50 厘米。

在竞赛中，规则规定传感器最多不能超过 16 个，选用 12 个激光传感器，所有的传感器呈“一”字排布。如图 3.2，激光传感器由两部份构成，一部份为发射部份，一部分为接收部份。发射部份由一个振荡管发出 300KHz 频率的振荡波后，激光管发光；接收部份由一个相匹配 300KHz 的接收管接收返回的光强，经过电容滤波后直接接入 S12 单片机，由于激光传感器使用了调制处理，接收管只能接受相同频率的反射光，因而可以有效防止可见光对反射激光的影响。为了更好的接收反射回来的光，在接收管上套一个透镜，可以起到一定的滤光作用，使接收管接收的信号更加强，便于单片机对数据进行处理。

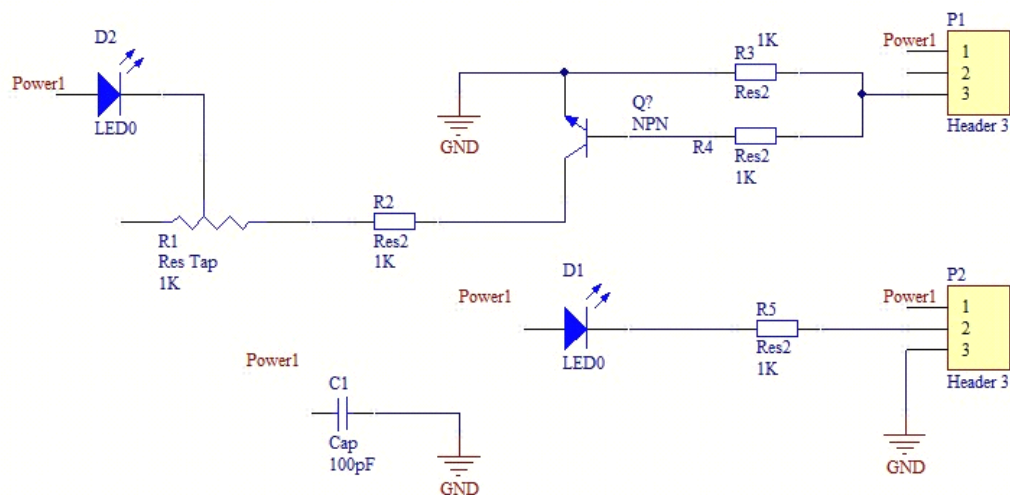


图 3.2 激光传感器发射与接收电路

为了简化 12 路激光传感器的控制，减少激光传感器相互之间的干扰，传感器的发光控制采用了分时点亮的策略，使用 74LS154 作为分时控制器，由 S12 的 4 个 I/O 口来控制传感器的开断。

3.4 智能车建模及尺寸参数

改装后的模型车如图 3.3

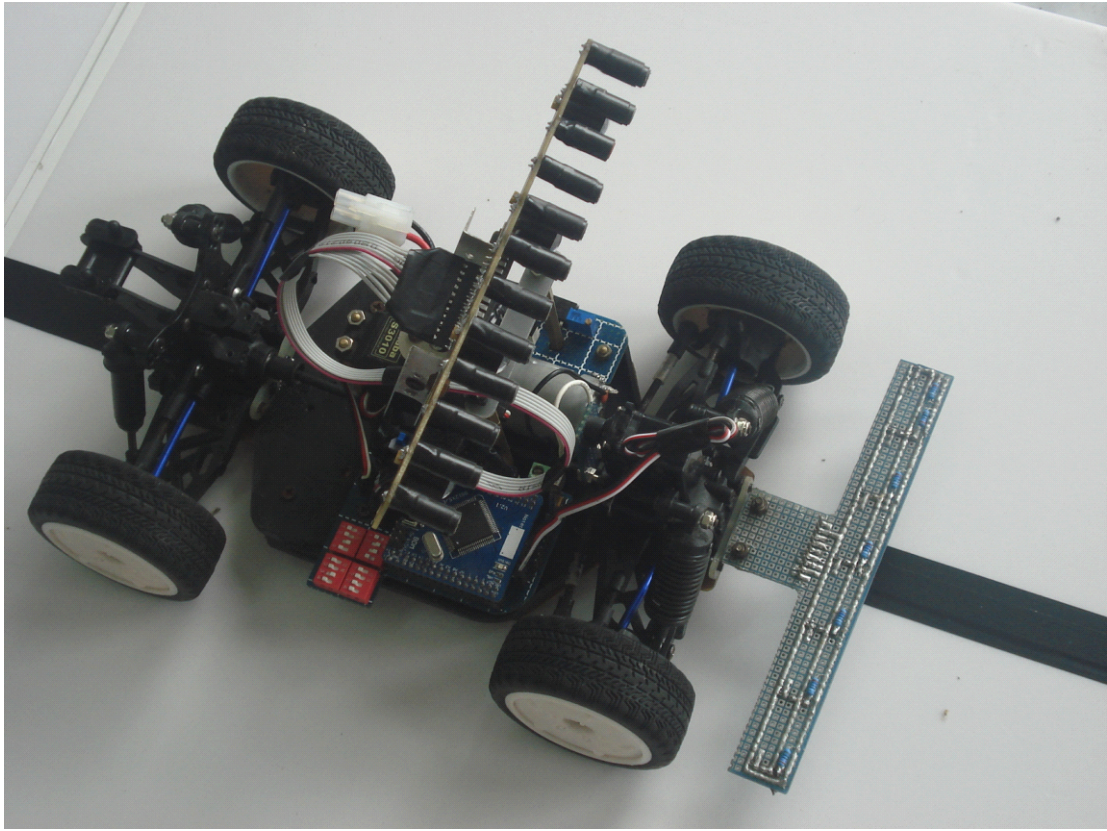


图 3.3 改装后的模型车

改装后的模型车基本尺寸参数如表3. 1

表 3.1 模型车基本尺寸参数

项目	尺寸
轴距	7cm
前轮距	15cm
后轮距	15cm
车轮直径	6cm
总重量	1.5kg
外形尺寸	28 x18x 16

第四章 电路设计

4.1 电源模块设计

电源是智能车系统中极其重要的部分，关系到整个系统是否能够正常工作，因此在设计控制系统时应选好合适并且电压稳定的电源。

根据竞赛规则，本次比赛 B 型车模的电源统一由 6 节串联共 9V 的电池提供，由于电路中的不同电路模块所需要的工作电压和电流容量不相同，因此电源模块应该包含多个稳压电路，将充电电池电压转换成各个模块所需要的电压。智能车对电路功率消耗有较严格的要求，如果稳压芯片选用 L7805，其转换压差较高，且 7805 直接输入不接输出的情况下，其内部还会有 3mA 的电流消耗（静态电流）。而 LM2940 是低压差线性稳定器，比 7805 转换效率更高。所以本智能车系统选用 LM2940 为其他模块进行供电，功耗低，供电稳定。

（1）经过 3 片稳压芯片 LM2940 稳压后，输出 5V 电压，分别对单片机、传感器和速度检测编码器供电。

（2）经过稳压芯片 LM2941 稳压后，输出 6V 电压，对转向舵机供电。

其中电机驱动模块的电源由电池直接提供。

电源稳压电路如图 4.1

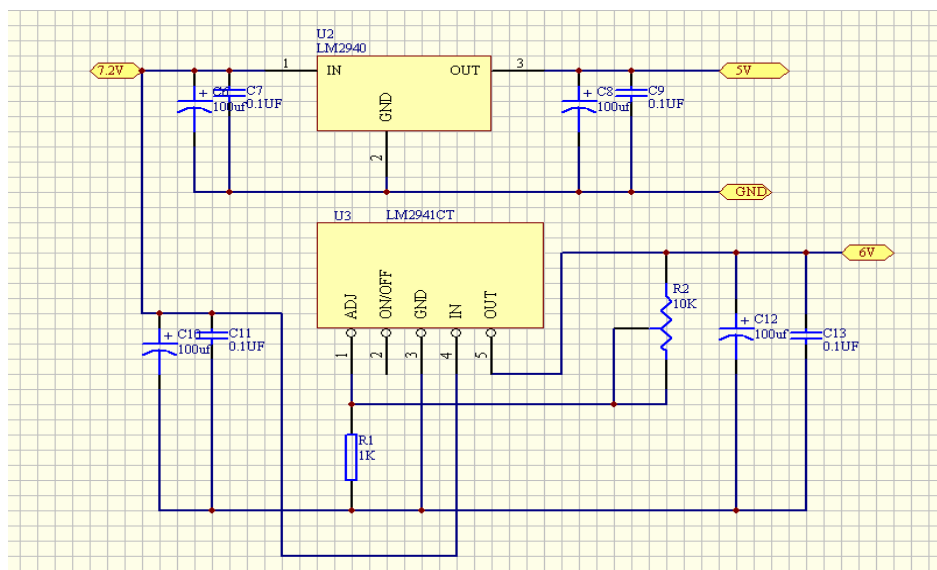


图 4.1 电源模块原理图

4.2 电机驱动模块设计

本智能车系统通过智能功率芯片 BTS7960 驱动直流电机转动，BTS7960 的接口电路如图 4.2 所示

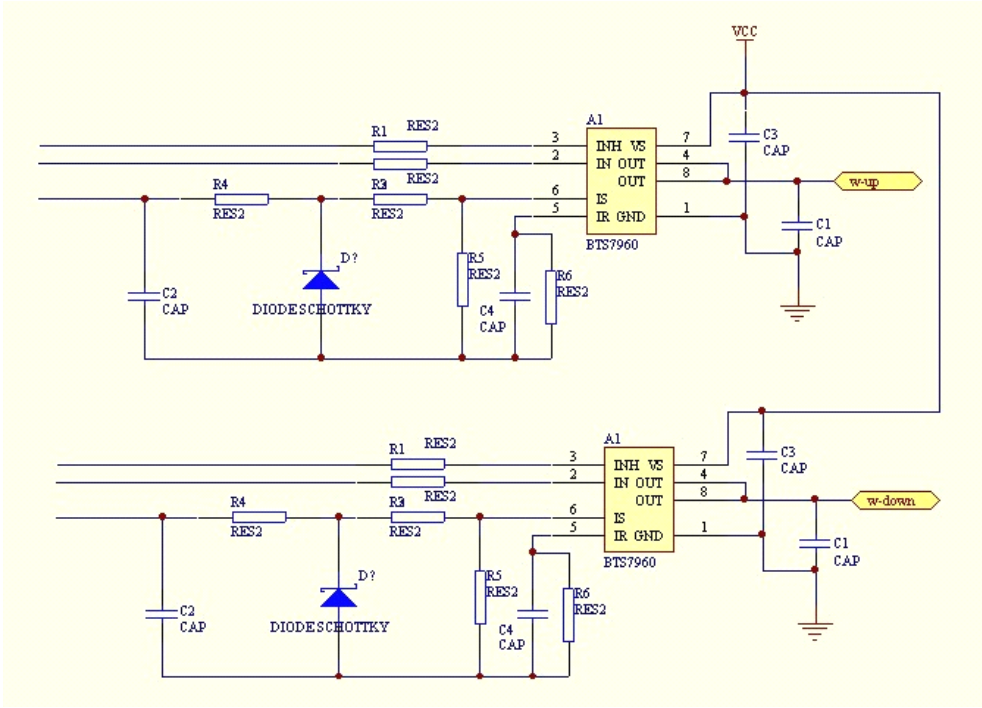


图 4.2 电机驱动电路

4.3 舵机控制模块设计

对舵机的控制分为“摇头”舵机与转向舵机，由于“摇头”舵机对转向的控制精度和转向灵敏度要求更高，本智能车的“摇头”舵机采用 A 型舵机，可以完成软件对它的控制要求，“摇头”舵机供电直接接 9V 电池。而转向舵机采用 B 型舵机，电源接 LM2941 稳压后的 6V 稳压直流电源。

4.4 车速检测模块设计

4.4.1 测速方法研究

车速检测采用编码器输出脉冲来测量智能车在行进过程中的速度。编码器测速方法主要有 3 种：M 法，T 法以及 M/T 法，综合对比 3 种测速方法，本智能车测速采用第三种 M/T 法测速，M/T 法兼有 M 法和 T 法的优点，在高速和低

速段都可以获得较高的分辨率。

4.4.2 M/T 法测速测速原理

同时测量检测时间和在此检测时间呢脉冲发生器发送的脉冲数来确定被测转速，原理如图 4.3 所示。它是用规定时间间隔 T_g 以后的第一个测速脉冲去终止时钟脉冲计数器，并由此计数器值 m_2 来确定检测时间 T 。检测时间为：

$$T = T_g + \Delta T \quad (\text{公式 1})$$

设电机再 T (s) 时间内转过的角度位移为 X (rad)，则其时间转速值为

$$Nm = \frac{60X}{2\pi T} = \frac{60X}{2\pi(T_g + \Delta T)} \quad (\text{公式 2})$$

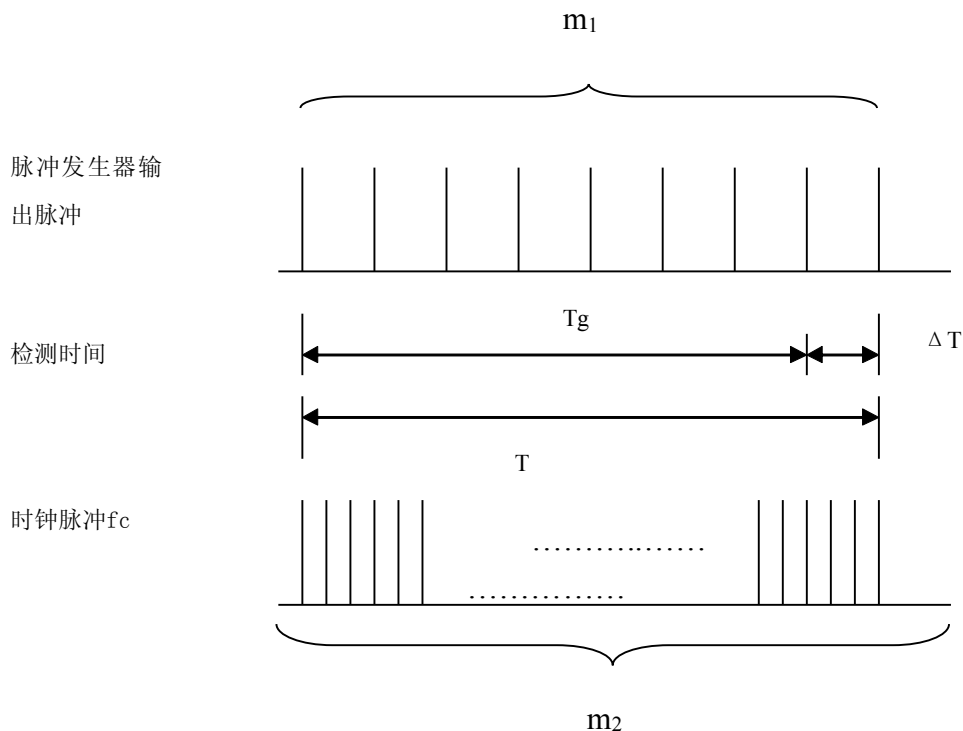


图4.3 M/T法检测原理

4.5 坡道控制模块设计

根据大赛规则，在决赛比赛中增加坡道赛道，智能车制作采用光电方式独立检测坡道路径，综合考虑用使用了 7 个安装在车前端的单光束反射取样式光电传感器 ST188 独立检测坡道路径。单个 ST188 检测电路如图 4.4。

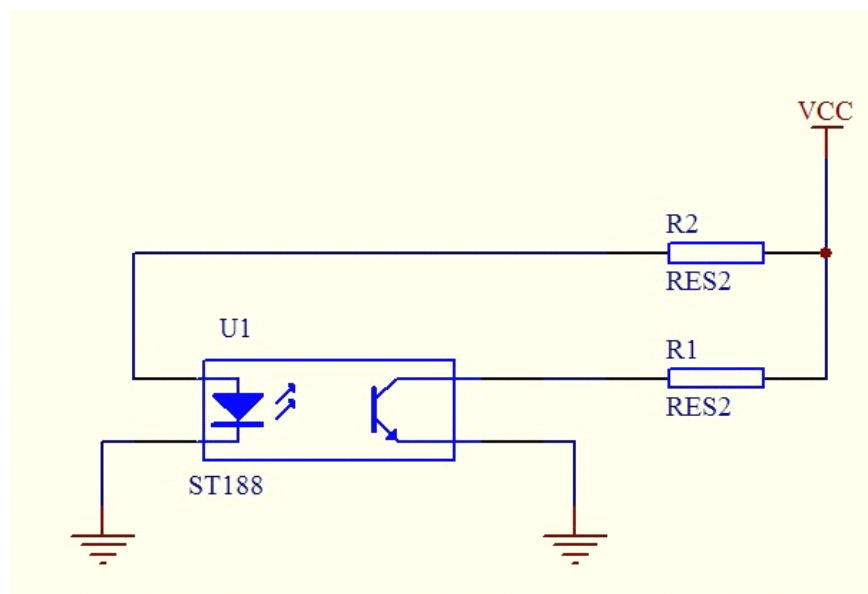


图4.4 单个ST188检测电路

该传感器检测电路简单，在硬件的外部我们没有加任何的信号滤波电路，由环境光引起的环境干扰不能通过 RC 滤波网络来简单的滤除，只能通过软件来滤除。

第五章 软件系统设计

5.1 软件系统总体设计

在整个系统设计中，用到了 4 个单片机基本功能模块：时钟模块、PWM 输出模块、ECT 模块、AD 转换模块。通过编写程序先对所用到的模块进行初始化，并通过对相应数据寄存器或状态寄存器的读写，实现期望的功能。

系统通过在主程序内循环调用信号检测、信号处理、路径计算和赛车控制等功能子模块，程序的执行为先对各个模块进行初始化，然后在主程序中完成相应的功能。

首先对各功能模块进行初始化，初始化流程图如图 5.1 所示。

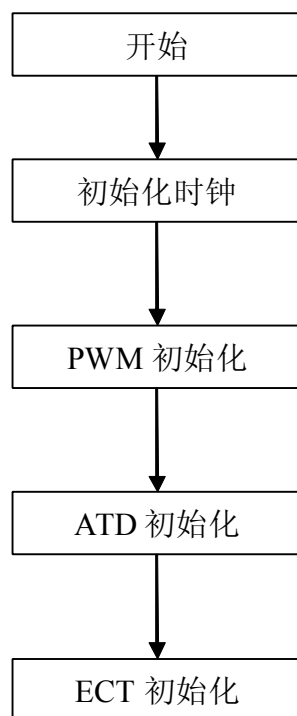


图 5.1 系统初始化流程图

初始化完成以后，就在主程序中编写程序，完成智能车沿赛道快速平稳行驶任务。

5.2 软件各功能模块设计

5.2.1 时钟模块

时钟基本脉冲是 CPU 工作的基础。MC9S12XS128 微控制器的系统时钟信号，由时钟振荡电路或专用时序脉冲信号提供。MCU 内部的所有时钟信号都来源于 EXTAL 引脚，也为 MCU 与其他外接芯片之间的通信提供了可靠的同步时钟信号。

S12 的总线时钟是整个 MCU 系统的定时基准和工作同步脉冲，其频率固定为晶体频率的 1/2。对于 S12，可以利用寄存器 SYNR、REFDV 来改变晶振频率 FOSCCLK，可以选用 8MHz 或 16MHz 外部晶体振荡器作外时钟。将 SYNR 设为 2，REFDV 设为 1，可以得到 24MHz 的总线频率，接近 S12 单片机的上限内部总线频率 25MHz。而锁相环产生的时钟频率为：

$$FPLLCLK=2*FOSCCLK*(SYNR+1)/(REFDV+1)$$

设计中我们将 SYNR 设为 2，REFDV 设为 1，因此，总线时钟为 24MHz。

5.2.2 ECT 模块

S12 的 ECT 具有 8 个输入(IC)/输出(OC)比较通道，可以通过设置 TIOS 寄存器选择输入或输出比较功能。ECT 既可以作为一个时基定时产生中断，也可以用来产生控制信号。

模数递减计数器（MDC）是 S12 微控制器 ECT 特有，它是一个 16 位计数器。其外围配备了常数寄存器 MCCNT 和控制寄存器 MCCTL，分别为 MDC 提供定时常数和时钟信号。通过寄存器 TCTL4 设定各个引脚的各种动作，初始化设置过程如下所示：

```
TCTL4=0x01;          // Set the rising endge for PT0.
PACN10=0x0000;
PBCTL=0x40;          //pt0 and pt1 级联成 16 位计数器
MCCNT=60000;         //60000*24M/16=40ms
MCCTL=0xC7;
TSCR1=0x10;
```

通过 ECT 模块，我们实现了对脉冲进行计数，检测智能车的速度，对速度

进行闭环控制。

5.2.3 AD 转换模块

AD 转换模块由模拟量前端的 8 选 1 多路转换开关，采样缓冲器及放大器，逐次逼近式模拟量转换、控制部分及转换结果存储部分等组成。

AD 转换所需要的时间周期是固定不变的，但采样时间和时钟频率可以通过寄存器 ATDxCTL4 (x 为 0 或 1) 在一定范围内选择，其公式为：

$$ATDClock = BusClock * 0.5 / (PRS + 1) \quad (\text{公式 4})$$

从公式和总线频率的取值范围可以得出 A/D 时钟频率范围满足：

$$500\text{kHz} \leq ATDClock \leq 2\text{MHz}$$

通过寄存器 ATDxCTL2 (x 为 0 或 1) 控制 ATDx 的启动、状态标志以及上电模式；寄存器 ATDxCTL3 (x 为 0 或 1) 用于控制结果寄存器的影射，设置转换长度；寄存器 ATDxCTL5 (x 为 0 或 1) 用于选择转换方式，选择转换通道，设置单/多通道转换和单次/连续转换模式以及选择对齐方式，写寄存器 ATDxCTL5 将会启动一次新的转换，如果写该寄存器时 ATD 正在进行转换，则转换操作将被中止。

5.3 路径识别与算法

要想很好的控制小车在规定的跑到上稳定而快速的行驶，路径的识别，小车准确的定位，非常的关键。

算法的依据是返回的二值化数据，通过这组数据算出小车的重心偏移量，公式如下：

$$position = v * 2 / black_number - 11 \quad (\text{公式 5})$$

Position 表示重心偏移量；v 表示传感去坐标；black_number 表示在黑线上的传感器个数。

在小车快速行驶的过程中，会有丢失数据，或是采集到噪点，所以要分别对丢失处理，和滤波。

若是丢失，我们就在附近的点进行搜索，同时保持先前的 pre_position 不变。

滤波算法是，如果数据的跳变超过 2 个传感器。这样的数据我们就认为是无效数据。

通过以上的算法，小车就可以稳定而快速的行驶了。

5.4 弯道控制及策略

车辆在弯道行驶时，需要对三个参数进行设定：切弯路径、转向角度、入弯速度。

其中，切弯路径主要决定了车辆是选择内道过弯还是外道过弯。切内道，路径最短，但是如果地面摩擦系数过小会导致车辆出现侧滑，原因是切内道时，曲率半径小，速度快，智能车需要的向心力很大，而赛道本身是平面结构，向心力将全部由来自地面的摩擦力提供，因此赛道表面的摩擦系数将对赛车的运行状态有很大影响。切外道，路径会略长，但是有更多的调整机会，同时曲率半径的增加会使得模型车可以拥有更高的过弯速度。

转向角度决定了车辆过弯的稳定性。合适的转向角度会减少车辆在转弯时的调整时间，不仅路径可以保证最优，运动状态的稳定也会带来效率的提高，减少时间。舵机的转角直接影响到车辆转向角度的大小，转角过大会导致前轮打死，赛车在高速行驶过程中前轮转向打死而影响速度甚至发生侧滑。转角过大则失去了过弯的最佳路径甚至冲出赛道。

对于入弯速度的分析，应该综合考虑路径和转向角度的影响。一般赛车采取入弯减速，出弯加速的方案，这样理论上可以减少过弯时耗费的时间。如果不考虑路径和转向角度，只是单纯地分析过弯速度，会造成控制策略的局限甚至错误。所以现在本系统参考实际驾驶时的一些经验，对过弯速度的处理方式确定为：入弯时急减速，以得到足够的调整时间，获得正确的转向角度；在弯道内适当提速，并保持角度不变，为出弯时的加速节约时间；出弯时，先准确判断标志，然后加速，通过消耗掉一定的时间，保证行驶状态的稳定性，而且弯道内的有限加速对之后赛车在直道上的提速也有很大的帮助。

5.5 速度与转角控制策略与算法

舵机的程序控制，分为“摇头”舵机和转向舵机。为了增加小车寻迹跟踪黑线的范围和稳定性，我们特别将光电收发模块装在一个可以“摇头”的舵机上。“摇头”舵机作为一个黑线跟踪的随动系统，要求响应快，控制可靠。对于“摇头”舵机的算法，我们使用的是 PD 算法，要到达的目标是使传感器紧跟黑线中心，其实这里就形成了一个闭环控制系统，这个系统中 P 参数影响系统的反应快慢，D 参数用于减小超调。既要很快的跟上黑线，又要防止振荡。

转向舵机的参数来源与“摇头”舵机 PWM 值，在加上传感器的重心偏移量。这些参数，我们经过多次反复的调试和现场整定，最终确定。

速度电机的控制使用的是 PID 算法，他能很好的对小车进行快速而准确的控制，做到响应快、振荡小、超调量小，稳定性好。

要想让小车跑的更快，特别是转角要和速度匹配的非常好，这是决定小车速度极限的关键和核心。我们采用现场整定的办法获取智能小车转角和速度的最佳匹配参数。方法是首先将小车的速度开环控制，给定一个较大的速度 PWM 值，然后在此恒定速度下获取并调整转角的控制曲线和参数值，最后再将速度闭环控制，进一步调整速度和转角的控制参数值，使两者达到一个最优控制。经过反复调试比较，转角的最终控制曲线接近于一个二次或三次函数曲线。

传感器使用了分时点亮的算法，每次点亮距离相隔最远的一个，这样就有效的减少和防止了的传感器间的互相干扰，能准确的返回数据。

第六章 开发制作与调试

6.1 系统开发工具

S12 系列微控制器具有一个由片内仿真、触发和跟踪硬件构成的单线背景调试模式(BDM)，因此它可以通过使用两种开发工具：简单串行电缆或低成本的 BDM，来完成调试功能。CodeWarrior for S12 是面向以 HC1 和 S12 为 CPU 的单片机嵌入式应用开发软件包。包括集成开发环境 IDE、处理器专家库、全芯片仿真、可视化参数显示工具、项目工程管理器、C 交叉编译器、汇编器、链接器以及调试器。

6.2 调试过程

智能车系统的调试是智能车制作过程中一个很重要的环节。在完成系统硬件和软件设计后，要进行系统调试，以检查系统的完整性和有效性。系统调试主要包括硬件调试和软件调试两部分。

6.2.1 硬件调试

对智能车系统调试，首先分别对直流电机驱动、舵机驱动、传感器、电源、速度检测等模块进行测试。特别是对传感器模块的测试，传感器就像是智能车的“眼睛”，对智能车沿规定赛道行驶有非常重要的作用。

传感器接收效果受激光发射管所发射光的距离影响，理论上是光照越近，接收管接收效果越好，但是较小前瞻对智能车对赛道的预知能力有所影响，导致智能车对整体控制效果的局限。后经过调试，激光发射管发射的光在 40-45cm 时接收管接收效果较好且方便控制芯片对信号进行处理。

各模块在调试过程中都出现了不同类型的问题，不过经过本小队队员的努力检查与改进，各模块均能正常工作。

单个模块调试完成后，就可以对智能车整体系统进行调试。将各个模块按照预先设计的位置安装在车模上，进行整体硬件调试。

首先不开电机与舵机，使前后轮都保持直线方向，用手轻推智能车，查看之前在机械机构设计过程中对车轮的改动是否达到预期的目标，调试初期车辆并不能保持直线方向行驶，而是稍稍向一边偏，这样赛车在直道行驶控制策略

当中就会影响控制精度。经过一定的改动过后，赛车直线偏向问题得到解决。给电机一个相对较小的速度，设置一个预测的转角，让赛车在赛道上慢跑一圈，检测各模块对系统整体的作用及对整体的误差影响。

经过不断地调试，系统各模块构成的整体逐步提高了控制精度和稳定性。

6.2.2 软件调试

同硬件调试过程一样，首先对软件 ECT 、 ATD 、 PWM 等模块进行调试。软件调试是一个漫长而艰辛的过程：传感器信号的接收和处理、速度控制算法、舵机转角对赛道的适应能力、起跑线的检测以及弯道控制策略的参数调整等，都是比较复杂而困难的历程。

软件调试的关键是看智能车采集的数据，使用在线仿真调试，我们可以方便的看到变量和寄存器的数值变化，为我们调试减轻了不少的负担，也可以使用无线收发模块，进行数据的传输实时的观察智能车动态情况，为我们进一步分析和提高车速以及加强稳定性有很大的意义。在反复的调试中寻找合适的参数，分析和解决所遇的的各种各样的问题。

经过长时间的调试以及对参数的调整，智能车达到对不同赛道的较强适应能力并能平稳快速的跑完全程。

第七章 智能车主要技术参数说明

7.1 智能车基本参数

车长：28cm

车宽：18cm

车高：16cm

车重：约 1.2kg

传感器个数：12

7.2 电路部分参数

本智能车采用一块比赛标准 9V 电池供电，电路中一共有 6 个电容，容量总计约 1110 μ F。当模型车全功率运行时，功耗约为 50W。

7.3 除车模原有的驱动电机、舵机外伺服电机数量

除车模原有的电机、舵机外还用了一个舵机。

第八章 总结

在这份技术报告中，我们分模块的介绍了智能车机械硬件方面、软件方面以及电路的设计，最重要的是我们的控制策略及算法。

在传感器的排法和位置选择上，经过综合考虑，我们把传感器按照“一”字型排列安装在智能车的中部，这样防止在赛道调试过程中车辆冲出赛道而撞坏模型车。另外我们注意到：激光发射管发射的光如果不在同一直线上，即发射光照出的距离不同，接收管接收信号会有很大差距，所以我们严格控制发射光必须控制在同一直线，这样对整个系统的稳定性有很大作用。

在软件设计和算法方面，我们先分模块介绍智能车所用到的模块的初始化流程，再详细的介绍路径识别、弯道控制以及车辆行驶速度的控制算法。通过不断地调试与参数的修改，赛车能够比较平稳的跑完整个赛道。

另外，机械结构相当重要。当模型车的速度遇到一个新的瓶颈，在尝试了新的控制策略、提高舵机响应等各种方法都没有效果时，就不要忽视了机械结构方面的因素，应及时检查运动部件是否出现松动，两车轮转角是否符合，并作出相应的改进。

一个稳定的系统必然要经历不断的调试与修改，我们小队成员都在不断努力，使模型车速度能够达到更高！

经过接近半年的努力，我们相信，在即将到来的比赛中我们会取得成功，也许我们的知识还不够丰富，思维还不够严谨，硬件技术还不够精湛，但是我们坚信，通过我们不断的学习和团队合作，一定会取得更高的成就，也认识到要做好一件事情，只有兴趣和激情是完全不够的，还要有持之以恒的毅力和不辞辛苦的精神，特别是软件部分的调试，没有冷静的头脑和缜密的思维，是不可能编写出结构严谨的程序的。这一部智能车和这一份技术报告，凝结了我们小队每一位成员的汗水和智慧。我相信它们会成为我们珍贵的记忆。

参考文献

- [1] 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
- [2] 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
- [3] 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
- [4] 臧杰, 阎岩. 汽车构造[M]. 北京. 机械工业出版社. 2005
- [5] 安鹏, 马伟. S12 单片机模块应用及程序调试[J]. 电子产品世界. 2006. 第 211 期. 162-163
- [6] 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
- [9] 沈长生. 常用电子元器件使用一读通[M]. 北京. 人民邮电出版社. 2004
- [10] 宗光华. 机器人的创意设计与实践[M]. 北京. 北京航空航天大学出版社. 2004
- [11] 张伟等. Protel DXP 高级应用[M]. 北京. 人民邮电出版社. 2002
- [12] 张文春. 汽车理论[M]. 北京. 机械工业出版社. 2005

附录:

程序源代码:

main.c:

```
/*
*****
COPYRIGHT (c) 2010 BY LSTC.
-- ALL RIGHTS RESERVED --

main.h
Leshan Teachers College Smartcar Lab
Designed by LiFuGang
Description:
*****/
#include <hidef.h> /* common defines and macros */
#include <MC9S12XS128.h> /* derivative information */
#include "sys_init.h"
#include "variable.h"
#include "function.h"
#include "driver.h"
#include "isr.h"
#pragma LINK_INFO DERIVATIVE "mc9s12xs128"
//-----main-----
void main(void)
{
    DisableInterrupts;
    system_init();
    key_BCD();
    led_show();
    for(;;)
    {
        EnableInterrupts;
        deal_sample();
        car_angle();
        speed_pid();
        set_wdt();
    }
}
```

第五届全国大学生智能汽车邀请赛技术报告

```
}
```

Driver.h:

```
/******  
COPYRIGHT (c) 2010 BY LSTC.  
-- ALL RIGHTS RESERVED --
```

main.h

Leshan Teachers College Smartcar Lab

Designed by LiFuGang

Description:

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

```
//-----deal_sample-----
```

```
void deal_sample(void)
```

```
{  
    uchar v,i;  
    uchar black_number;  
    v=0;  
    black_number=0;  
    if(pre_position>=7)  
    {  
        for(i=8;i<=13;i++)  
        {  
            if(laser_n[i])  
            {  
                v+=i;black_number++;  
            }  
        }  
    }  
    else if(pre_position<=-7)  
    {  
        for(i=0;i<=5;i++)  
        {  
            if(laser_n[i])  
            {  
                v+=i;black_number++;  
            }  
        }  
    }  
}
```



```

else
{
    for(i=3;i<=10;i++)
    {
        if(laser_n[i])
        {
            v+=i;black_number++;
        }
    }
}
if(!black_number)
{
    LED=1;
    if(absolute(pre_position)<5)
    {
        for(i=5;i<=9;i++)
        {
            if(laser_n[i])
            {
                v+=i;black_number++;
            }
        }
    }
    if(!black_number)
    {
        if(pre_position>=5)        position=13;
        else if(pre_position<=-5)  position=-13;
        else                        position=pre_position;
    }
    else
    {
        position=v*2/black_number-13;
    }
}
else
{
    position=v*2/black_number-13;
}
}
//-----car_angle-----
void car_angle(void)

```

第五届全国大学生智能汽车邀请赛技术报告

```
{
    if(position>0)    angle_pwm=angle_center-angle_data;
    else             angle_pwm=angle_center+angle_data;
    if(position>=13)    angle_pwm=LEFT;
    else if(position<=-13)    angle_pwm=RIGHT;
    PWMDTY01=angle_pwm;
    pre_position=position;
}
//-----car_driver-----
void car_driver(sint x)
{
    if(x>8000)        x=8000;
    if(x<-10000)      x=-10000;
    if(x>=0)
    {
        PWMDTY23=0;PWMDTY67=x;
    }
    else
    {
        PWMDTY23=-x;PWMDTY67=0;
    }
}
}
Function.h:

/*****
COPYRIGHT (c) 2010 BY LSTC.
-- ALL RIGHTS RESERVED --

function.h
Leshan Teachers College Smartcar Lab
Designed by LiFuGang
Description:
*****/
void set_wdt(void)
{
    ARMCOP = 0X55;
    ARMCOP = 0XAA;
}

//-----延时1us-----
```

```

void delay_us(uint i)
{
    while(i--)    //24MBus
    {
        _asm(nop); _asm(nop);
        _asm(nop); _asm(nop);
        _asm(nop); _asm(nop);
        _asm(nop); _asm(nop);
        _asm(nop); _asm(nop);
        _asm(nop); _asm(nop);
        _asm(nop); _asm(nop);
        _asm(nop); _asm(nop);
        set_wdt();
    }
}
//-----延时1ms-----
void delay_ms(uint i)
{
    while(i--)    //24MBus
    {
        delay_us(1000);
    }
}
//-----LED-----
void led_show (void)
{
    LED=0;
    delay_ms(500);
    LED=1;
    delay_ms(500);
    LED=0;
}

//-----取绝对值-----
uint absolute(sint x)
{
    return(x>0?x:-x);
}
//-----封装采样-----
void packet_sample(void)

```

```
{
    laser_n[0]=laser[0];
    laser_n[1]=laser[1];
    laser_n[2]=laser[2];
    laser_n[3]=laser[3];
    laser_n[4]=laser[4];
    laser_n[5]=laser[5];
    laser_n[6]=laser[6];
    laser_n[7]=laser[7];
    laser_n[8]=laser[8];
    laser_n[9]=laser[9];
    laser_n[10]=laser[10];
    laser_n[11]=laser[11];
    laser_n[12]=laser[12];
    laser_n[13]=laser[13];
}
//-----get_speed-----
void get_speed(void)
{
    pulse_count=PACNT;
    PACNT=0;
}
//-----key_BCD-----
void key_BCD(void)          //T2-1  T4-2  T6-3
{
    if(key==0)      {speed_max=20;speed_min=12;}
    else if(key==1) {speed_max=25;speed_min=20;}
    else if(key==2) {speed_max=25;speed_min=20;}
    else if(key==3) {speed_max=38;speed_min=20;}

    else if(key==4) {speed_max=40;speed_min=20;}
    else if(key==5) {speed_max=42;speed_min=22;}
    else if(key==6) {speed_max=44;speed_min=26;}
    else if(key==7) {speed_max=46;speed_min=28;}
}
```

Isr.h:

/*****

COPYRIGHT (c) 2010 BY LSTC.

-- ALL RIGHTS RESERVED --

isr.h

Leshan Teachers College Smartcar Lab

Designed by LiFuGang

Description:

Sys_init.h:

```

/*****
COPYRIGHT (c) 2010 BY LSTC.
-- ALL RIGHTS RESERVED --
sys_init.h
Leshan Teachers College Smartcar Lab
Designed by LiFuGang
Description:
*****/

#include <hidef.h>
#include <mc9s12xs128.h>
//-----crg_init-----
void crg_init(void)
{
    CLKSEL=0;                                //bus=2*16M*(2+1)/(1+1)
    PLLCTL_PLLON=1;
    SYNCR=0x00|0x02;
    REFDV=0x80|0x01;
    POSTDIV=0X00;
    while(!(CRGFLG_LOCK));                    //总线频率为24m
    CLKSEL_PLLSEL=1;
}
//-----pwm_init-----
void pwm_init(void)
{
    //pwm01
    PWMCTL_CON01=1;                          //01 级联
    PWMCLK_PCLK1=1;                          //通道1时钟源为 clock SA
    PWMPRCLK=0x12;                          //时钟周期4分频 舵机:预分频以后colck_sa为
24mhz/4=6MHZ colck_sb为24mhz/4=12MHZ
    PWMPOL_PPOL1=1;                          //极性选择, 先高后低
    PWMCAE_CAE1=0;                          //通道1左对齐
    PWMSCLA=0x01;                          //3MHZ

```

```
PWMPER01=60000;      //pwm输出为3000KHZ/60000=50HZ
PWCNT01=0;           //从0开始计数
PWMDTY01=2575;
PWME_PWME1=1;

//pwm4
PWMCLK_PCLK4=1;      //通道4时钟源clock SA
PWMPOL_PPOL4=0;      //极性选择，先高后低
PWMCAE_CAE4=0;       //通道3左对齐
PWMSCLA=0x01;        //6MHZ/2=3000KHZ
PWMPER4=5;           //3000KHZ/5=600khz
PWMDTY4=2;           //1/2占空比
PWCNT4=0;             // 5 2
PWME_PWME4=1;

//pwm23
PWMCTL_CON23=1;      //23 级联
PWMCLK_PCLK3=1;      //通道4时钟源clock SB
PWMPOL_PPOL3=0;      //极性选择，先低后高
PWMCAE_CAE3=0;       //通道3左对齐
PWMSCLB=0x01;        //12MHZ/2=6000KHZ
PWMPER23=10000;      //6000KHZ/10000=600HZ
PWMDTY23=0;
PWCNT23=0;
PWME_PWME3=1;
//pwm67
PWMCTL_CON67=1;      //67 级联
PWMCLK_PCLK7=1;      //通道4时钟源clock SB
PWMPOL_PPOL7=0;      //极性选择，先低后高
PWMCAE_CAE7=0;       //通道3左对齐
PWMSCLB=0x01;        //12MHZ/2=6000KHZ
PWMPER67=10000;      //6000KHZ/10000=600HZ
PWMDTY67=0;
PWCNT67=0;
PWME_PWME7=1;
}
//-----ect_init-----
void ect_init(void)
{
    TIOS=0X01;    //定时器通道0设置为输出比较
```

```

    TSCR1=0X80;
    TSCR2=0X06;
    TC0=0X0088;
    TCTL2=0X00;
    TIE=0X81;
    PACTL=0X50;    //PT7 PIN,PACN32 16BIT,FALLing edge,NOT
INTERRUPT0
}
//-----io_init-----
void io_init(void)
{
    DDRT=0X00;
    PERT=0xff;
    PPST=0xff;
    DDRS=0XFF;
    PERS=0xff;
    PPSS=0xFF;
    DDRB=0X04;
    PUCR_PUPBE=1;
    DDRA=0X00;
    PUCR_PUPAE=1;
    DDRJ=0x00;
    IRQCR=0xc0;
    PPSJ=0X00;
    PIEJ=0X80;
}
//-----wdt_init-----
void wdt_init(void)
{
    COPCTL = 0X07;
}
//-----syetem-----
void system_init(void) {
    DisableInterrupts;
    crg_init();
    pwm_init();
    ect_init();
    wdt_init();
    io_init();
}

```

第五届全国大学生智能汽车邀请赛技术报告

}

Variable.h:

/******

COPYRIGHT (c) 2010 BY LSTC.

-- ALL RIGHTS RESERVED --

variable.h

Leshan Teachers College Smartcar Lab

Designed by LiFuGang

Description:

*****/

#define angle_center 2575

#define SERVE_KP 12

#define SERVE_KD 180

#define LEFT (angle_center-550)

#define RIGHT (angle_center+550)

#define LED PORTB_PB2

//-----laser-----

uchar laser_num=0;

uchar laser[14];

uchar laser_temp[14]={13,6,12,5,11,4,10,3,9,2,8,1,7,0};

uchar laser_n[14];

//-----angle-----

uint angle_data;

uchar jump;

uchar time;

sint angle_pwm;

uchar position_flag=0;

uchar right_flag,left_flag;

uchar e_angle_k0,e_angle_k1,e_angle_k2,e_angle;

uchar finish_flag=0

//-----speed-----

uint pulse_count;

uint straight_count;

ulong countdown_timer;

uchar pulse_time;

uchar speed_max;

uchar speed_min;

//0 1 2 3 4 5 6 7 8 9 10


```

11      12      13
      //67  7    78  8    8 9    9    9 10  10    10 11  11    11 12
12      12 13    13
      //0 1 2 3 4 5  67  8 9 10 11 12 13
//-----pid-----
uint  MOTOR_KP;
uint  MOTOR_KI;
uint  MOTOR_KD;
sint  e_k0,e_k1,e_k2;
sint  speed;
uint  ideal_speed;
uint  pre_ideal_speed;
ulong time_count=0;
uchar key;
//-----position-----
schar position,pre_position,position_1;
//-----hongwai-----
uchar key_num=0;
uchar flag=0;
//-----start_line-----
uchar black_num=0;
uchar stop_flag;
uchar check_flag;

```