

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： 哈尔滨工业大学

队伍名称： 终结者队

参赛队员： 苏坤岳

张焯然

陈东

带队教师： 王淑娟

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

第一章系统总体设计	5
1.1 系统概述	5
1.2 整车布局	5
第二章机械系统安装及调整	5
2.1 前轮倾角的调整	5
2.2 底盘高度的调整	6
2.3 舵机安装结构的设计和安装	6
2.4 摄像头的固定及安装	6
第三章硬件电路系统设计	6
3.1 硬件系统概述	6
3.2 电路系统设计	7
3.2.1xs128 最小系统板	7
3.2.2 电源模块.....	8
3.2.3 视频信号分离模块.....	9
3.2.4 A/D 转换模块.....	10
3.2.5 电机驱动模块.....	11
第四章程序设计及算法.....	12
4.1 黑线提取算法简介	12
4.2PID 控制算法简介.....	14
4.2.1 位置式 PID	16
4.2.2 增量式 PID	16
4.2.3PID 参数整定	17
4.3 舵机与电机控制算法	17
第五章系统开发软件及调试工具	18
5.1CodeWarrior	18
5.2 蓝牙通信模块	18
5.3 串口调试助手	18
第六章模型车的主要技术参数.....	19
结论	19

参考文献.....	20
附录.....	21

第一章 系统总体设计

1.1 系统概述

本文设计的智能车系统以 MC9S12XS128 微控制器为核心控制单元,通过 CMOS 摄像头检测赛道信息,使用模拟比较器对图像进行硬件二值化,提取黑色引导线,用于赛道识别;通过光电编码器检测模型车的实时速度,使用 PID 控制算法调节驱动电机的转速和转向舵机的角度,实现了对模型车运动速度和运动方向的闭环控制。为了提高模型车的速度和稳定性,使用 LabVIEW 仿真平台、无线模块、串口等调试工具,进行了大量硬件与软件测试。实验结果表明,该系统设计方案确实可行。

1.2 整车布局

本组车模现阶段的控制系统主要采用模糊控制与增量式 PID 相结合的控制策略,主要分为速度控制器和舵角控制器两部分。MCU 将摄像头得到的赛道信息处理,以赛道的弯曲程度为基础得到理论速度和理论舵角。码盘将实际速度以数字形式反馈给速度控制器。速度控制器综合理论速度和反馈速度计算出处理后的速度,通过 PWM 传输给 7960 模块,进而控制速度。舵角控制器对理论舵角进行处理得到舵角增量,通过 PWM 传给 S3010 伺服器, S3010 伺服器通过自带控制系统将舵角改变为控制值。

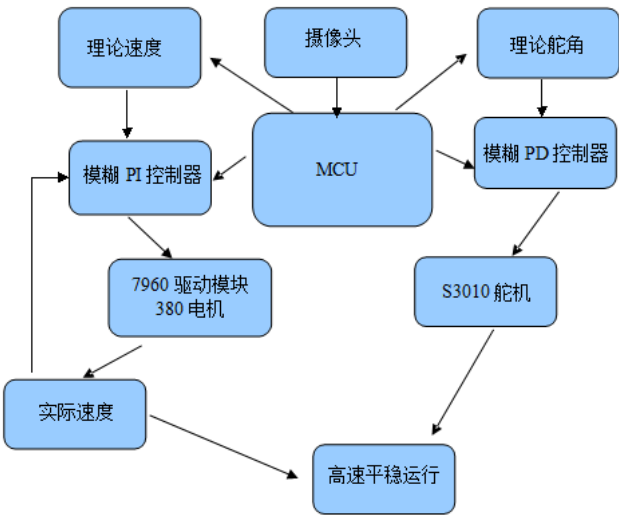


图1.2 控制系统简介

第二章 机械系统改装及调整

2.1 前轮倾角的调整

在调试过程中，我们发现由于前轮轴和车轮之间的间隙较大，对车高速时转向中心的影响较大，会引起高速转向下模型车的转向不足。然而这里是规则中严禁改动的部分，所以为了尽可能降低转向舵机负载，我们对前轮的安装角度，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等 4 个项目决定，反映了转向轮、主销和前轴等三者车架上的位置关系。在实际调试中，我们发现适当增大内倾角的确可以增大转弯时车轮和地面的接触面积，从而增大车了地面的摩擦程度，使车转向更灵活，减小因摩擦不够而引起的转向不足的情况。

2.2 底盘高度的调整

在保证顺利通过坡道的前提下，底盘尽量降低，从整体上降低模型车的重心，可使模型车转弯时更加稳定、高速。

2.3 舵机安装结构的设计和安装

原装车模的舵机为卧式安装，考虑到主板的安装方便以及车模转向性能，我们对舵机安装结构进行了较大的调整。比赛车模的转向是通过舵机带动左右横拉杆实现。舵机的转动速度和功率是一定，要想加快转向机构的响应速度，唯一的办法就是优化舵机的安装位置及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。利用实际参数经计算，我们得出了一套可以稳定高效工作的参数及结构。最终，我们设计了一套舵机连片(转向拉杆)，综合考虑了速度与力矩的关系，并根据模型车底盘的具体结构，简化了安装方式，实现了预期目标。

关于舵机的安装方式，我们实验室较为主流的有直立式安装和倒置式安装，与连片的啮合类似齿轮的啮合，由于连片长度以及齿宽的影响，很难调节到合适的舵机机械中值。为了弥补这种连片的缺陷，我们采用了可调中值的舵机连片。

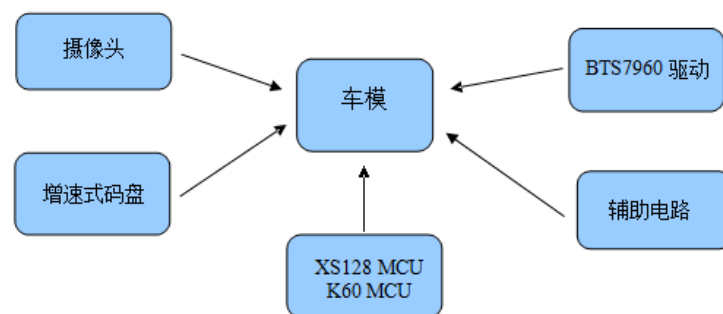
2.4 摄像头的固定及安装

为了降低整车重心，需要严格控制 CCD 摄像头的安装位置和重量，我们自行设计了轻巧的铝合金夹持组件并采用了碳纤维管作为安装 CCD 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，使摄像头便于拆卸和维修，具有赛场快速保障能力。

第三章 硬件电路系统设计

3.1 硬件系统概述

本组以博思数码公司生产的 A 型车模为基础，以摄像头为主要传感器，通过增速式码盘对速度进行精确测量，形成速度闭环控制。本组采用 freescale 公司生产的 S12 系列、K 系列单片机为系统处理器，进行信息采集、算法处理、PWM 控制。此小车以 380 电机为驱动，我们选用 BTS7960 芯片为电机驱动控制模块，加上电压升降模块、摄像头 AD 辅助模块实现赛车系统完整性。

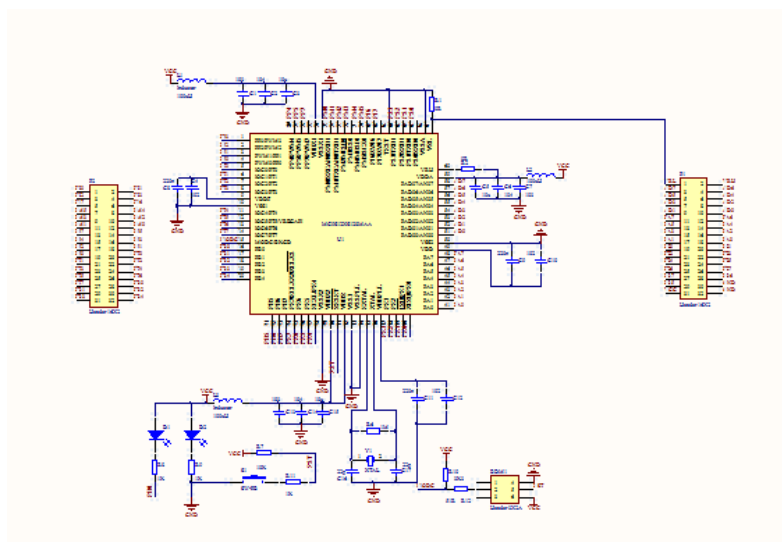


3.2 电路系统设计

3.2.1 xs128 最小系统板

Freescape16 位 HCS12 系列单片机也称 MC9S12 系列，简称 S12 系列。MC9S12X 系列是 HCS12 系列的增强型产品，基于 S12 CPU 内核，可达到 25MHz 的 HCS12 的 2~5 倍的性能。S12X 系列增加了 172 条额外指令，可以执行 32 位计算（共 280 条指令），总线频率最高可达 40MHz，并且具备完全的 CAN 功能，改进了中断处理能力。S12X 系列的 CPU 以复杂指令集 CISC 架构，集成了中断控制器，有丰富的寻址方式。中断有 7 个优先级并且内核支持优先级的调度，最多可有 117 个中断源，S12X 可访问最多 8M 的全部存储空间（包括片内和片外资源）。单片机最小系统板使用 MC9S12XS128 单片机，112 引脚封装，为减少电路板空间，板上仅将本系统所用到的引脚引出，包括 PWM 接口、计数器接口、外部中断接口、若干普通 IO 接口等。还包括电源滤波电路、时钟电路、复位电路、串行通讯接口、BDM 接口和 SPI 接口。

单片机最小系统板的原理图如图所示。



最小系统板原理图

3.2.2 电源模块

12V 电源模块：此模块采用 MC34063 芯片作为电压转换芯片，该器件本身包含了 DC / DC 变换器所需要的主要功能的单片控制电路且价格便宜。它由具有温度自动补偿功能的基准电压发生器、比较器、占空比可控的振荡器，R—S 触发器和大电流输出开关电路等组成。该器件可用于升压变换器、降压变换器、反向器的控制核心，由它构成的 DC / DC 变换器仅用少量的外部元器件，主要应用于以微处理器 (MPU) 或单片机 (MCU) 为基础的系统里。我们的应用电路里输入电压为电池电压 7.2V，经芯片转换为 12V 后为摄像头供电。

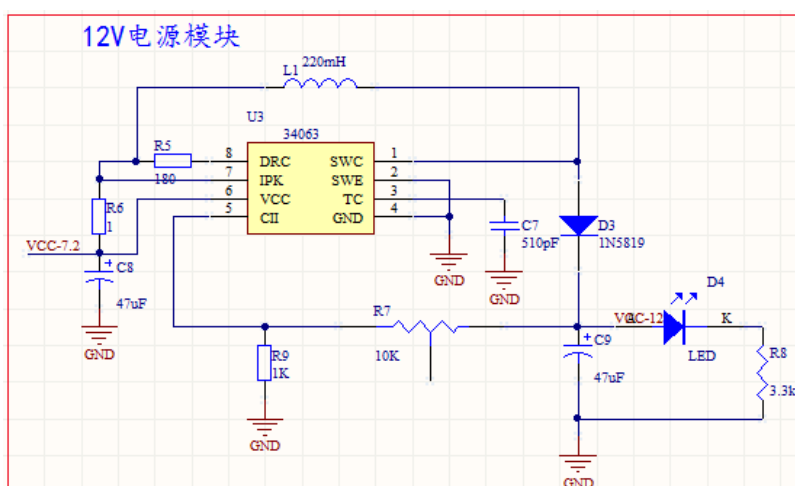


图 4.2 12V 电源模块

5V 电源模块：此模块采用 LM2940 电压转换芯片将电池电压 7.2V 转换为 5V，为单片机以及其他需 5V 电源的模块供电，对于单片机，需要提供稳

定的 5V 电源，由于 LM2940 的稳压的线性度非常好且转换压差较大，所以选用 LM2940 单独对其进行供电。

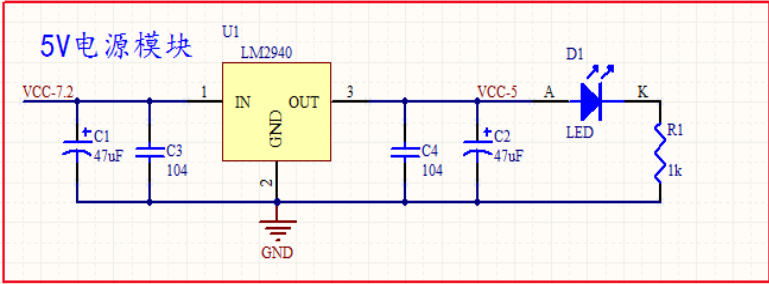


图 4.3 5V 电源模块

6V 电源模块：此模块采用 LM2941 电压转换芯片将电池电压 7.2V 转换为 6V，为舵机供电。

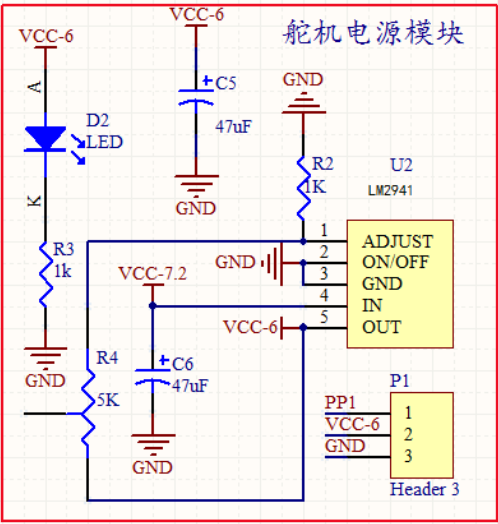


图 4.4 6V 电源模块

3.2.3 视频信号分离模块

视频信号分离模块：摄像头产生的信号为 PAL 制式的视频信号，PAL 电视标准为每秒 25 帧，每帧包括两场，电视扫描线为 625 线，奇场在前，偶场在后，每行结束有行中断信号，每场结束有场中断信号，场中断周期 20ms，行中断周期 64us，此模块采用 LM1881 芯片将摄像头的视频信号进行分离，分别产生单独的行中断，场中断以及奇偶场信号，再接到单片机的外部中断口产生中断，程序根据相应的中断进行相应的响应。

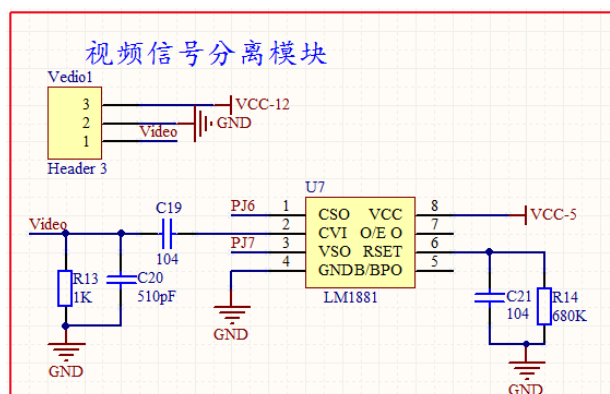


图 4.5 视频信号分离电路

3.2.4A/D 转换模块

AD 转换模块:此模块采用 TLC5510 高速 AD 采集芯片采集摄像头的信号,再输入单片机进行相应的处理, TLC5510 是 CMOS、8 位、20MSPS 模拟量转数字量的转换器(ADC),它采用半闪速结构(semi-flash architecture)。单 5V 工作电源且功耗只有 100mW(典型值)的功率。内含采样和保持电路,具有高阻抗方式的并行接口和内部基准电阻。与闪速转换器(flash converters)相比,半闪速结构减少了功率损耗和晶片尺寸。通过在 2 步过程(2-step process)中实现转换,可以极大地减少比较器的数目。转换数据的等待时间为 2.5 个时钟。内部基准电阻使用 VDDA 可产生标准的 2V 满度转换范围。为了实现此选项仅需外部跳线器,这样减小了对外部基准或电阻的需求。差分线性度在 25℃温度下为 0.5LSB,在整个工作温度范围内的最大值是 0.75LSB,用差分增益 1%和差分相位为 0.7%可以规定动态特性范围,其较快的采集频率以及响应速度正好适应我们更快的处理摄像头数据。

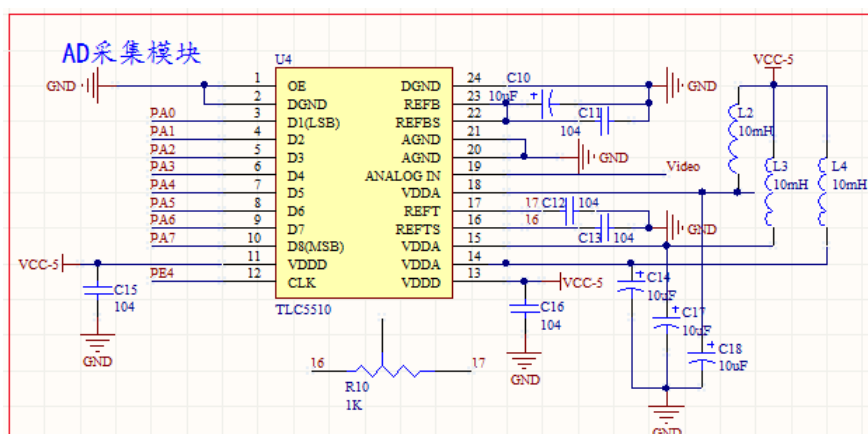
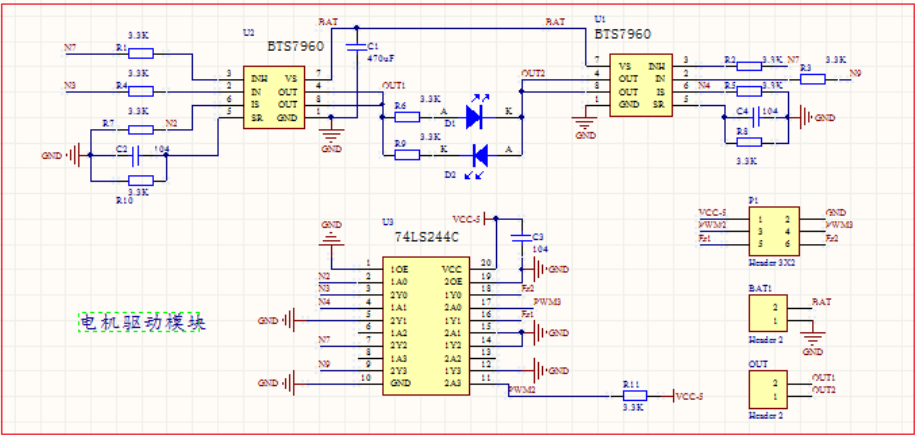


图 4.6 AD 采集模块

3. 2. 5 电机驱动模块



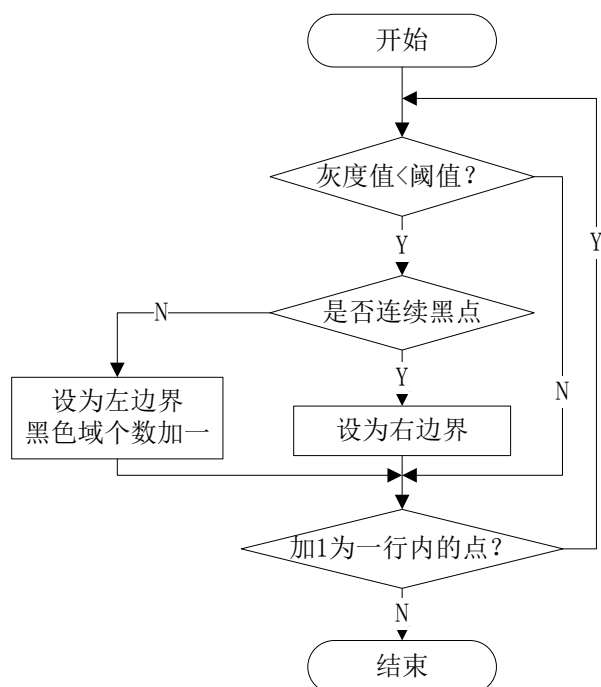
电机驱动模块

第四章 程序设计及算法

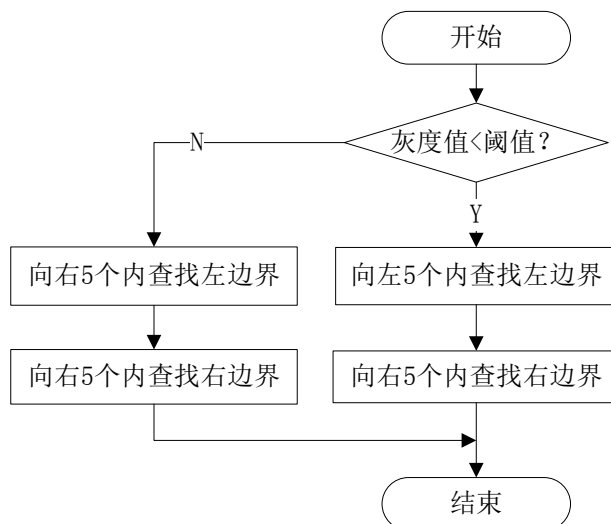
4.1 黑线提取算法简介

在不断的调试过程中，我们一共应用了三种赛道黑线提取的算法。GetBlackDots1, GetBlackDots2, GetBlackDots3, 此外我们还对视野范围内赛道是否有界做出了判断 BoundaryJudge。

GetBlackDots1 可称作轮询法（名字可能不太准确），即对一行图像所有点的灰度值与阈值进行比较，类似于二值化，但并不置 0 或置 1。而是判断出此行共有几个黑色区域，最后一个黑色区域中像素点的个数，以及其左边界和右边界的像素点。

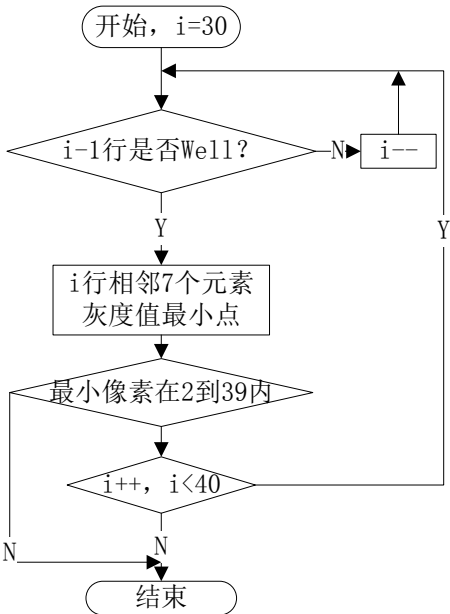


GetBlackDots2 可称作边沿提取法，此算法建立在第一种算法之上。若前一行数据只有一个黑色域，那么从 • 与前一行为左边界相邻的本行像素开始判断，分别判断相邻左右各五的像素查找本行的左边界，然后再查找右边界。



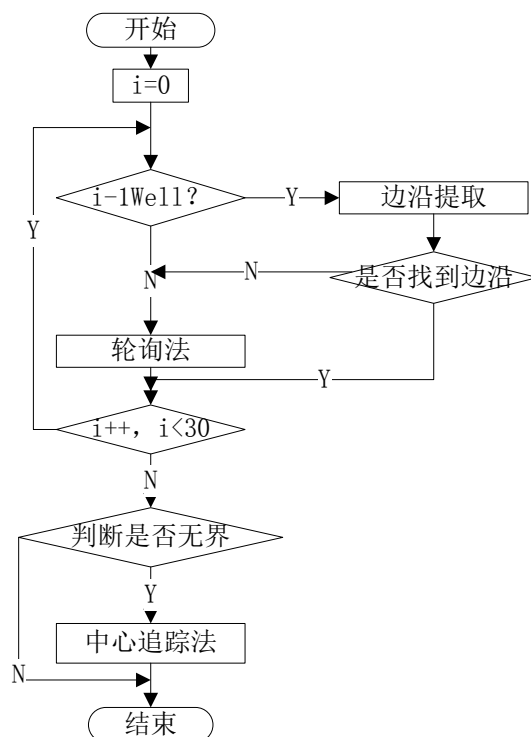
如果算法 2 能成功提取到左右边界则继续处理下一行，提取不到则对此行再次使用算法 1。

GetBlackDots3 可称作中心追踪法，主要处理 30 到 40 行图像。因为到最后黑线变的很细，且其灰度值变大，有可能大于之前设定的阈值。中心追踪法就是在前一行黑线中心的基础上，在下一行与其相邻的 7 个像素中，选择灰度值最小的像素做为其中心。这种算法的另一个优点是可以有效的避免赛道外界干扰。



一帧图像为 40*100 像素，我们将 40 行数据首先分为两部分，1 到 30 行为一部分，30 行道 40 行为一部分。首先判断 30 到 40 行图像，判断赛道是否在这一区域为无界的，即一直到最后每行都有赛道存在。如果有界，则指处理 1 到 30 行的图像，无界在处理全部，具体的处理流程图如图 4-8 所示。

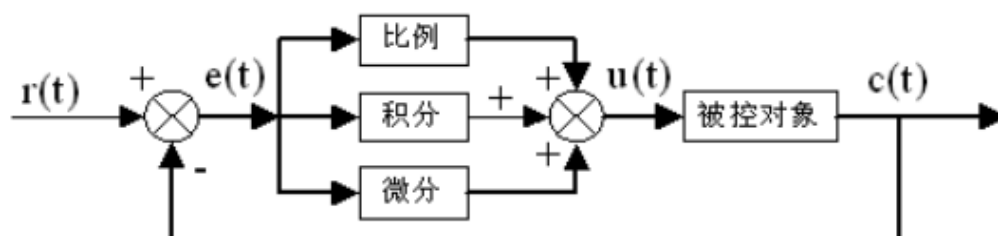
经过如上的处理过程，我们能得出每行的基本状况，主要分为 Well、Out、Right、Left、Polluted 等几种情况。若为 Well，还能得出黑线的左右边界，从而得出黑线的中心，以及中心值与图像中心 50 的偏差值。至此，赛道黑线信息提取的过程已经完成。

[illegible]

4. 2PID 控制算法简介

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 4.8 所示。



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：

$$e(k) = r(k) - c(k)$$

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_I} \sum_{j=0}^k e(j) + \frac{T_D}{T} [e(k) - e(k-1)] \right\}$$

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。

4.2.1 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(如阀门), $u(k)$ 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(4.2)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 $\Delta u(k)$ 。

4.2.2 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(4.2)推导出提供增量的 PID 控制算式。由式(4.2)可以推出式(4.3), 式(4.2)减去式(4.3)可得式(4.4)。

$$u(k-1) = K_p \left\{ e(k-1) + \frac{T}{T_I} \sum_{j=0}^{k-1} e(j) + \frac{T_D}{T} [e(k-1) - e(k-2)] \right\} \quad (4.3)$$

$$\Delta u(k) = K_p \left\{ [e(k) - e(k-1)] + \frac{T}{T_I} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)] \right\} \quad (4.4)$$

$$= K_p \Delta e(k) + K_I e(k) + K_D [\Delta e(k) - \Delta e(k-1)]$$

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

公式(4.4)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(4.4)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. 2. 3PID 参数整定

运用 PID 控制的关键是调整 KP、KI、KD 三个参数，即参数整定。PID 参数的整定方法有两大类：一是理论计算整定法。它主要是依据系统的数学模型，经过理论计算确定控制器参数；二是工程整定方法，它主要依赖工程经验，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。由于智能车系统是机电高耦合的分布式系统，并且要考虑赛道的具体环境，要建立精确的智能车运动控制数学模型有一定难度，而且我们对车身机械结构经常进行修正，模型参数变化较为频繁，理论计算整定法可操作性不强，最终我们采用了工程整定方法。此外，我们先后实验了几种动态改变 PID 参数的控制方法。

4. 3 舵机与电机控制算法

本队的速度调节采用 PID 调节，方向控制采用 PD 调节，控制策略不是很复杂。根据不同的赛道情况，分配不同的速度和方向控制的偏差值。如在直道情况下，我们的速度标准值设为 $SpdPID1.Ref = Model.spdZ - intErr$ ，其中 $Model.spdZ$ 为较大的常量，而我们的方向参考值设为 $StrPID1.Err = ErrAverage[1]$ ，以及第二区偏差值的均值为 PID 偏差值，第二区前瞻相对较远，可提前预判；在弯道情况下，我们的我们的速度标准值设为 $SpdPID1.Ref = Model.spdC - intErr$ ，其中 $Model.spdC$ 为较小常量，而我们的方向参考值设为 $StrPID1.Err = ErrAverage[0]$ ，及已第 1 区偏差值的均值为 PID 偏差值，第 1 区前瞻相对较近，因此时第 2 区内经常没有黑线信息。弯道相对直道来说，方向 PID 的参数较大，转向更加灵活。

控制策略也是需要进一步调节的部分，只有在实际调节过程中，才能进一步优化，是赛车性能进一步提高，速度不断加快，这也是飞思卡尔比赛的乐趣所在。

第五章 系统开发软件及调试工具

5.1 CodeWarrior

我们采用 CodeWarrior 5.1 for HCS12 作为推荐的程序编译软件。

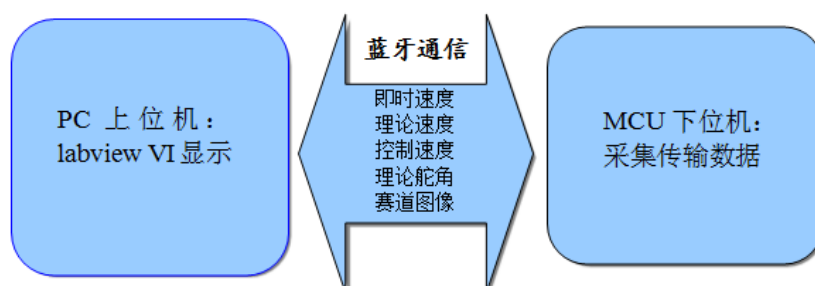
“CodeWarrior for S12” 是面向以 HC12 或 S12 为 CPU 的单片机嵌入式应用开发的软件包。包括集成开发环境 IDE、处理器专家库、全芯片仿真、可视化参数显示工具、项目工程管理器、C 交叉编译器、汇编器、链接器以及调试器。

5.2 蓝牙通信模块

为了便于调试，模型车在赛道上运行时的各个数据都需要在电脑上显示和保存。为实现这一目的，我们充分利用了 XS128 和 K60 单片机上的资源，利用外接蓝牙，是 MCU 与 PC 之间保持信息通信；并利用电脑上存储空间较大特点，在电脑上记录赛车运行时个参数。

MCU 编程时，在程序中设立节点，在节点处设立发送函数，将赛车运行的个参数传输个蓝牙模块，实现下位机功能。蓝牙模块将接收到的数字信息，以无线电波方式，传给电脑蓝牙适配器，蓝牙适配器将无线信号转换成数字，传给电脑 LabVIEW VISA 模块。

本组利用了 NI（国家仪器公司）的 LabVIEW 虚拟仪表编写了上位机，通过 LabVIEW 自带 VISA 模块在电脑上与蓝牙模块进行信息通信，并利用 LabVIEW 强度图将摄像头数据在电脑上实时显示，将赛车其他参数通过曲线、指示灯的方式同样直观的相识电脑上。我们通过 LabVIEW 自带文件保存功能将蓝牙通信传输的信息直接保存在电脑上，实现在 MCU 脱机以后，赛车所有运行信息参数的回放。



5.3 串口调试助手

对于数据的采集，我们还使用串口调试助手来向 PC 发送采集的数据，并显示出来，可以粗略的得到图像信息，且方便快捷。

第六章 模型车的主要技术参数

赛车基本参数	长	28.5cm
	宽	16.3cm
	高	30.1cm
车重	0.987kg	
功耗	空载	8.2W
	带载	大于 15w
电容总容量	1186.30612uf	
传感器	光电编码器	一个
	CCD 摄像头	一个
除了车模原有的驱动电机、舵机之外的伺服电机个数	0	
赛道信息检测	视野范围（近/远）	25/180cm
	精度（近/远）	2/9cm
	频率	50Hz

结论

自大二上学期初开始做车以来，我们小组成员从零做起，查阅往届技术报告，组装车模，调整机械，设计算法，调整参数，分析问题，最终完成了最初的目标。

上学期主要完成了硬件电路的制作，主控芯片的初始化，上位机写法的学习。下学期开始组装整车，研究图像采集算法和舵机、电机的控制策略，并不断调整参数，终于使车体姿态控制达到了预期的效果，直道加速，弯道减速，并使得车体在弯道内保持较好的路径，保证弯道所走路径最短，速度损失最小。

这一年的制作和调试时间中，在试验场地和经费中，我们都得到了学校和学院的大力支持，在此特别感谢一直关注和支持智能车竞赛的学校和学院的各位领导。

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
- [7] 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
- [8] 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
- [9] 张伟等. Protel DXP 高级应用[M]. 北京. 人民邮电出版社. 2002

附录：源代码

```
void SpeedDang(void)
{
    char speed_dang;
    ATD0DIENL=0xFF;
    DDR1AD0=0x00;
    PER1AD0=0xFF;
    speed_dang=PT1AD0&0x03;
    switch(speed_dang)
    {
        case 0:
            PORTB_PB5 = 0;
            //PORTB_PB0 = 1;
            LineHighSpeed=30;
            LineLowSpeed=20;
            BendHighSpeed=21;
            BendLowSpeed=18;
            break;
        case 1:
            PORTB_PB0 = 0;
            PORTB_PB5 = 1;
            LineHighSpeed=22;
            LineLowSpeed=20;
            BendHighSpeed=18;
            BendLowSpeed=14;
            break;
        case 2:
            break;
        case 3:
            break;
    }
}
```

```

        default:
            break;
    }
}

void ECT_speed_getting(void)
{
    SpeedDang();
    // char BendHighSpeed=20,BendLowSpeed=16;
    if(SmallSFlag==1)
    {
        num_off1=(offset[40] + offset[35] + offset[30]
            +offset[25]+ offset[20]+ offset[15] - 390)/7;

        ECT_speed = LineHighSpeed - num_off1*
num_off1*(LineHighSpeed-LineLowSpeed)/1020;
    }
    else
        ECT_speed = BendHighSpeed;
    if((BendFlag==1)&&(SmallSFlag==0))
    {
        if(PresFlag==1)
            ECT_speed = BendLowSpeed; // = BendHighSpeed - num_off1*
num_off1*(BendHighSpeed-BendLowSpeed)/1020;
        else
            ECT_speed = BendHighSpeed;// - num_off1* num_off1*12/1020;;
        num_off1=(offset[40] + offset[36] + offset[32]
            +offset[28]+ offset[24]+ offset[20] - 390)/7;
    }
    else /
    {
        num_off1=(offset[40] + offset[35] + offset[30]
            +offset[25]+ offset[20]+ offset[15] - 390)/7;

        ECT_speed = LineHighSpeed - num_off1*
num_off1*(LineHighSpeed-LineLowSpeed)/1020;
    }
}

```

```

    }
    if(StopFlag==1)
    {
        Stop_delay2++;
        if(Stop_delay2>0)
        {
            Stop_delay2=1;
        }
    }
}

void PID_drive(void)
{
    ECT_off = ECT_speed - ECT_num;
    //增量式 PID
    PID_num = PID_num + (ECT_off-ECT_off1)*P_num1/P_num2
                + ECT_off/I_num2           //I 环节//P 环节
                + (ECT_off-2*ECT_off2+ECT_off4)/D_num1; //D 环节

    if((Stop_delay2>0)||((Stop_delay<3000))
    {
        if(ECT_num>8)
            PID_num=-5;
        else
            PID_num=0;
    }

    if(PID_num>grade_num)
        PID_num = grade_num;
    else if(PID_num<-grade_num/3)
        PID_num = -grade_num/3;

```

```
Drive(PID_num);
```

```
ECT_off4 = ECT_off3;ECT_off3 = ECT_off2;
```

```
ECT_off2 = ECT_off1;ECT_off1 = ECT_off;
```

```
}
```