

# 武汉科技大学

## 本科毕业设计

题目：基于摄像头的两轮自平衡车的设计

---

学 院：	信息科学与工程学院
专 业：	自动化
学 号：	200804134179
学生姓名：	徐 曙
指导教师：	潘 炼
日 期：	二〇一二年六月

## 摘 要

两轮自平衡小车是移动机器人的一个重要分支，具有体积小、运动灵活、可零半径转弯等诸多优点。并且相对于传统的四轮车，两轮车在直立行走、导航和定位方面对硬件设计、控制软件开发以及系统调试等提出了更高的要求。

本文设计了一种基于视觉导航的两轮自平衡小车系统，并实现了小车的直立行走，自主寻迹和定位三大功能。系统采用飞思卡尔十六位微处理器 MC9S12XS128 作为核心控制单元，通过增加各种传感器，设计稳压模块、最小系统模块、双轮测速模块、倾角测量模块、电机驱动模块和人机交互模块并编写相应程序以完成平衡控制，速度控制，转向控制和实时定位四大任务。

本设计通过对倒立摆进行动力学建模，类比得到小车平衡的条件。利用陀螺仪和加速度计获得车体的倾角和角速度，并进行卡尔曼滤波。对倾角进行 PD 控制实现小车的基本直立。通过光电编码器分别得到车子的线速度和转向角速度，对速度进行 PI 控制。将转速控制信号与平衡控制信号叠加加载到后轮两电机上，实现小车的静止和直立行走。用摄像头获得道路图像信息，进行图像预处理，提取出赛道中心线，获得车子的方向偏差控制量。通过对方向控制量和转向角速度进行 PD 控制来调节左右两轮的差速实现小车的转向和自主寻迹。建立里程计模型和 IMU 模型，对系统进行简单定位。最后对系统的误差进行分析与修正。

本设计开发了基于 MFC 的串口调试上位机对系统进行实时调试。实践证明，该两轮自平衡车具有很好的静态平衡性能和动态性能，能够在原地 4cm 范围以内保持静止，并能在道路上自主寻迹，流畅转弯。且该系统能有效抵抗外部扰动。

**关键词：** 自平衡； 倒立摆； IMU； 卡尔曼滤波； 航位推算

## Abstract

The self-balanced two-wheeled intelligent car is an important branch of the mobile robot, it has the advantages of small volume, flexible movement, zero turning radius etc. As compared to the traditional four wheel car, the vehicle walking upright, navigation and positioning put forward higher requirements in the hardware design, control software and system testing and other aspects.

This paper has designed a two wheeled self-balanced intelligent car system based on visual navigation, and finally achieved the goal that the car could walk upright stably, run along the racing track and positioning independently. This system uses the freescale 16 bit microprocessor MC9S12XS128 as the core control unit. By adding various sensors, designing the voltage regulator module, the minimum system module, double speed measure module, angle measurement module, motor drive module and man-machine interaction module, this system finally complete four complex tasks including balance control, speed control, steering control and real time positioning through designing corresponding control algorithms and other programs.

This design first obtain the balance conditions of the two wheeled self-balanced car through establishing dynamic model of the inverted pendulum. Then, this system uses gyroscopes and accelerometers to measure the dip angle and angular velocity of the car, and utilizing Kalman filter to fuse dip angle and angular velocity. The design uses PD controller in the dip angle control to realize basic upright. Then the design uses photoelectric encoders to obtain the speed of two wheels and steering angular velocity. The speed is controlled by PI controller. The car could keep stationary and upright by combining the speed control signal and dip angle control signal together. The design uses a CMOS image sensor to obtain current image information. Through image processing, extracting the track centerline and getting the direction migration information, we get the direction signal, the system could steer and tracing along the road independently by using PD controller. This paper also established odometer model and IMU model, combining the encoder and IMU sensor information through

adaptive weighted fusion and Kalman fusion to achieve the system positioning. Finally, this paper analysis the system's some errors and makes corrections.

This design develops MFC serial debugging host computer for system's parameters to debug. Through continuous practice and debugging, it improved that the self-balanced two-wheeled intelligent car system has very good static balance performance and dynamic performance, it can keep still in the same place within 4 cm distance, and it can also walk upright quickly, run along the path independently and smoothly. The system can also resist the external disturbance very well.

**Key words:** self-balanced; inverted pendulum; IMU; Kalman filter; dead reckoning

## 目 录

1	绪论	1
1.1	引言	1
1.2	课题研究背景	1
1.3	研究意义及国内外现状	2
1.3.1	两轮自平衡机器人的研究意义	2
1.3.2	国外两轮自平衡机器人技术的发展	2
1.3.3	国内两轮自平衡机器人技术的发展	3
1.3.4	两轮自平衡机器人技术的发展趋势	3
1.4	论文的主要内容	4
2	课题任务及方案论证	5
2.1	主要任务	5
2.2	传感器的选择	5
2.2.1	姿态传感器的选择	5
2.2.2	导航方式及寻迹传感器的选择	6
2.3	方案的论证	7
2.3.1	平衡控制的方案选择	7
2.3.2	双轮测速方式	7
2.3.3	系统定位方式	7
2.3.4	多传感器融合技术	8
3	系统原理概述	10
3.1	系统的任务分解	10
3.2	平衡控制原理	10
3.3	速度控制原理	12
3.4	摄像头的工作原理	13
3.5	转向控制原理	14
3.6	两轮驱动系统定位模型	14
3.6.1	里程计模型	14

3.6.2	IMU 模型 .....	16
4	系统硬件设计 .....	17
4.1	系统硬件组成 .....	17
4.2	MC9S12XS128 微处理器简介 .....	18
4.3	系统各硬件模块设计 .....	20
4.3.1	稳压电源模块 .....	20
4.3.2	倾角测量模块 .....	21
4.3.3	图像采集模块 .....	22
4.3.4	直流电机驱动模块 .....	23
4.3.5	双轮测速模块 .....	23
5	系统软件设计 .....	26
5.1	算法功能与框架 .....	26
5.1.1	单片机的硬件资源配置 .....	26
5.1.2	整个系统的软件框架 .....	27
5.2	卡尔曼滤波算法设计 .....	28
5.2.1	倾角传感器误差的分析 .....	28
5.2.2	卡尔曼滤波器简介 .....	29
5.2.3	陀螺仪和加速度计的卡尔曼滤波 .....	31
5.3	图像处理及黑线提取算法设计 .....	33
5.3.1	视频信号的采集 .....	33
5.3.2	图像的平滑滤波去噪 .....	34
5.3.3	图像的二值化 .....	34
5.3.4	赛道中心线的提取 .....	35
5.4	系统控制算法设计 .....	37
5.4.1	PID 控制算法简介 .....	37
5.4.2	平衡控制算法设计 .....	38
5.4.3	转向控制算法设计 .....	40
5.5	系统的定位算法设计 .....	41
5.5.1	编码器与加速度计的自适应加权数据融合 .....	41
5.5.2	编码器和陀螺仪的卡尔曼滤波器融合 .....	42

5.6 系统的误差分析与修正 .....	43
6 系统的安装与调试.....	44
6.1 系统机械的安装.....	44
6.1.1 测速编码器的安装.....	44
6.1.2 摄像头的安装 .....	44
6.1.3 倾角传感器的安装.....	44
6.2 系统的调试.....	45
6.2.1 开发平台 .....	45
6.2.2 无线串口调试 .....	46
6.2.3 系统的调试与参数整定.....	47
结束语 .....	50
参考文献.....	51
致 谢 .....	52
附录 A 整体设计实物图 .....	53
附录 B 部分硬件电路图 .....	55
附录 C 部分源代码 .....	56

# 1 绪论

## 1.1 引言

近几年来,移动机器人由于其广阔的应用前景,已经成为机器人研究领域的一个重要分支。伴随着移动机器人研究的不断深入,移动机器人的应用领域更加广泛,其所需适应的环境和执行的任务也更复杂。当要求机器人在狭窄的空间,仍然需要实现灵活机动转弯的时候,如何能够保持机器人灵活快捷的执行任务已成为现实应用上所面对的一个问题<sup>[1]</sup>。

两轮自平衡小车的概念就是在这样的需求下提出来的。两轮自平衡小车原理上是源于旋转驱动式倒立摆,它具有两轮共轴、独立驱动的特点,小车的重心位于轮轴上方,通过左右两个电机的运动保持动态平衡。由于两轮移动机器人具有特殊的结构,适应地形变化能力强,运动灵活,因此可以胜任一些复杂环境里的工作。与传统的轮式移动机器人相比,两轮移动机器人主要有如下优点:

- (1) 转向半径小,移动轨迹更为灵活易变,适宜狭窄空间的运动,可以在原地实现 360° 转弯;
- (2) 体积小,在场地面面积较小或要求灵活运输的场合十分适用;
- (3) 调速效果好:由 CPU 直接控制电机驱使机器人完成启动、加速、匀速、减速等动作,无需刹车、离合等装置,运动流畅;
- (4) 特殊的应用场合:可应用于步行街、广场、游乐场和大型会场等汽车无法通行、步行不便的场合,并且还环保为环保轻型车提供了一种新的思路。

## 1.2 课题研究背景

两轮移动机器人自面世以后,由于其不稳定的动态性能和系统所具有的强非线性,迅速吸引了世界各国机器人研究者的兴趣,成为验证各种控制理论的理想平台,具有重大的理论研究意义。两轮移动机器人所使用的控制方法主要有:PID 控制、状态反馈控制、极点配置控制、最优控制等,通常称为传统控制方法。但是传统控制方法的控制对象必须是一个已知的系统,即系统参数和状态都已知。由于两轮移动机器人的强非线性,对机器人的模型需进行线性化处理。这样得到的机器人模型不精确,控制效果不能达到最优。对于像两轮移动机器人这种工作环境需要经常变换的机器人来说,在对系统建模时采用了将非线性方程线性化的近似手段,使得线性化后的模型不再准确,如果仍然用传统的控制方法,将很难得到最佳控制效果。国内外也有一些研究者针对于两轮移动机器人采用了自适应控制、神经网络控制、模糊控制等技术进行了许多非线性控制方法研究。虽然有些控制方法控制效果还可以,但是这些方法普遍存在控制器设计难度大,参数难以调节、运算复杂度过大等局限性,往往难以在实际系统中真正应用<sup>[2]</sup>。

全国大学生飞思卡尔智能汽车竞赛作为自动化领域最重要的一项大学生科技竞赛,



大赛组委会今年也首次将电磁组的小车改用两轮行走方式。然而基于摄像头导航寻迹的两轮自平衡小车目前还是首例。

### 1.3 研究意义及国内外现状

#### 1.3.1 两轮自平衡机器人的研究意义

由于两轮移动机器人具有结构特殊、体积小、运动灵活、适应地形变化能力强、能够方便地实现零半径回转、鲁棒性强、适于在狭小和危险的工作空间内活动、能够胜任一些复杂环境里的工作。因此，两轮移动机器人有着广泛的应用前景，其典型应用包括代步工具、通勤车、紧急服务、空间探索、危险品运输、灭火、智能轮椅、高科技玩具、多机器人合作、机器人足球、控制理论测试平台等方面。目前在国际上，对于两轮移动机器人的应用研究也都处于刚刚起步阶段。所以两轮自平衡小车的研究很有意义<sup>[3]</sup>。

#### 1.3.2 国外两轮自平衡机器人技术的发展

最早提出两轮移动机器人构想的是日本 Electro-Communications 大学的 Kazuo Yamafuji 教授，他在 1985 年就萌生了制造一种自动站立机器的构思，并在 1987 年申请的一项“平行双轮机器人”的专利中使用了该技术。近几年，随着两轮移动机器人的研究不断开展，该项技术开始成为全球机器人控制技术的研究热点之一。

2002 年，瑞士联邦工业大学的 Felix 等研制出可遥控的两轮移动机器人 Joe，该装置如图 1.1 所示，其最大运动速度可以达到 1.5m/s。Joe 能稳定地做 U 型回转。Fellx 最初使用解耦状态空间控制器来保持系统的平衡，设计者认为使用自适应模糊控制算法能提高 Joe 的抗干扰能力。

2002 年，Leg 公司设计了两轮移动机器人 LegWay，该装置如图 1.2 所示的。Legway 是第一款玩具移动机器人，外形只有手掌大小，结构紧凑、控制灵活、模块化的结构设计，使得二次安装和拆卸都很方便。设计中引入了电机的差动驱动方式，机器人可以工作在斜面甚至不规则表面上，同时支持无线遥控操作。该机器人采用光电接近探测器来测量与地面之间的距离来实现平衡，并实现了对特定曲线轨迹进行跟踪。



图 1.1 机器人 Joe

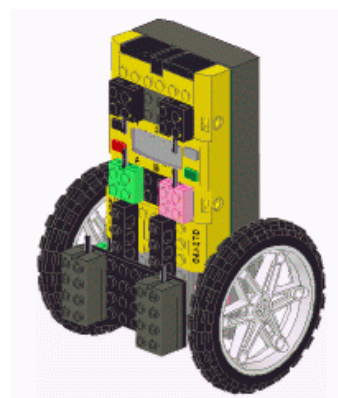


图 1.2 机器人 LegWay

### 1.3.3 国内两轮自平衡机器人技术的发展

目前，国内现阶段主要针对机器人建模与实验原型机的研制进行研究。

2004 年，中国科学技术大学成功研制了两轮自平衡电动代步车 Free Mover，如图 1.3 所示。它可以实现零半径回转，无刹车系统，整车质量约为 30 公斤。正常工作时，操作者经 10 分钟的训练后就可以驾驶其行走。身体前倾时车子自动加速，后倾时自动减速。通过控制手柄，该电动车可原地转弯任意角度，时速可达 10 公里。连续行驶里程约 30 公里<sup>[4]</sup>。

2005 年，哈尔滨工程大学的尹亮制作了一部如图 1.4 所示的双轮直立移动机器人 Sway，该机器人系统采用两块 8051 单片机分别作为机器人和上位机的核心控制器。车体倾斜角度由加速度计和反射式红外线距离传感器测量，提高了检测精度，采用差分算法提高系统的抗干扰能力。由 MODEM 完成机器人与上位机之间的数据通讯，液晶显示器作为人机交互界面。但该两轮自平衡机器人只能工作在理想的水平路面，没有对车轮打滑及地面不平整等异常情况的处理进行考虑。



图 1.3 两轮代步车 Free-Mover



图 1.4 机器人 Sway

### 1.3.4 两轮自平衡机器人技术的发展趋势

表 1.1 对几种典型的两轮自平衡机器人系统的结构特点，功能特点和传感器进行了综合比较。

表 1.1 典型两轮自平衡机器人的结构特点比较

	系统结构	感知系统类型	传感器
Legway	质量均布式	单传感器	光电探测器器
nbot	质量均布式	组合传感器	组合传感器 FAS-G
Joe	倒立摆式	单传感器式	陀螺仪
SegWay	倒立摆式	组合传感器式	1 个陀螺仪，2 个倾角计和光学传感器
Flathru	质量均布式	组合传感器式	3 个陀螺仪，1 个 3 轴加速度计

综上所述可以看出,为了实现两轮移动机器人的实用化,在机器人的系统设计导航以及控制方法等方面仍然有许多有待研究的问题,目前移动机器人的主要研究内容有路径规划、导航定位、控制、多移动机器人协调和协作等问题。具有重要的学术研究和实际应用价值。虽然国内外有很多研究机构对两轮移动机器人进行了大量研究,但是在机器人前进、后退、旋转等各种运动状态下,如何设计控制策略,保持车体系统的动态移动以及受干扰后能立即恢复平衡及导航等问题方面仍然有许多值得研究的地方。以下几方面是未来的两轮移动机器人的研究发展趋势。系统设计需要满足机器人动态移动的要求,既要有小巧、对称的机械结构,又要有简单、完善的控制不同,国内外现有原型系统在系统设计方面各具特色,体现了设计的多样性。机械结构从定质心倒立摆式结构、定质心质量均布式结构、变质心倒立摆式结构到变质心质量均布式结构;信息感知系统采用的传感器形式从单传感器、多传感器到组合传感器;控制系统从单微处理器到多微处理器结构<sup>[5]</sup>。

### 1.4 论文的主要内容

本论文主要叙述了基于视觉导航的两轮自平衡小车的设计与实现的整个过程。主要内容为两轮自平衡小车的平衡原理,直立控制,速度控制,转向控制,图像处理,小车的自主寻迹及系统定位算法的设计。整个内容分为六章,包括绪论、课题任务及方案论证、系统原理概述、系统硬件设计、系统软件设计和系统的机械安装及调试。第一章主要讲解了课题的研究背景、意义、国内外相关技术的发展现状和两轮自平衡机器人技术的发展趋势。第二章主要讲解了的设计的主要任务,传感器的选择和方案的论证。第三章主要讲解了两轮自平衡小车系统的任务分解,平衡控制原理,速度控制原理,摄像头的工作原理,转向控制原理及两轮驱动系统的定位模型。第四章主要讲解了系统的硬件设计,详细介绍了小车系统的硬件构成,主控芯片 MC9S12XS128 的结构及组成,以及稳压电源模块,倾角测量模块,图像采集模块,直流电机驱动模块和双轮测速模块的设计。第五章是系统的核心,主要讲解了软件设计的算法功能与框架,卡尔曼滤波器的简介及其算法的设计,图像处理及黑线提取算法设计,系统控制算法设计,系统定位算法设计及系统误差分析与修正。第六章主要讲解了系统机械部分的安装及系统的调试。最后总结与展望,对于其它各种问题,予以简单的说明和处理。

## 2 课题任务及方案论证

### 2.1 主要任务

本文研究并设计了一种基于视觉导航的两轮自平衡小车系统，实现了两轮小车的直立行走，自主寻迹和自主定位三大功能。系统采用飞思卡尔十六位微处理器 MC9S12XS128 作为核心控制单元，通过增加各种传感器，设计相应电路并编写相应程序分别完成平衡控制，速度控制，转向控制和实时定位四大任务。

系统需要利用角度传感器获得车体的倾角和角速度，并对测量角度和角速度进行卡尔曼滤波融合。通过编码器获取两轮的速度信息与正反转，并对速度进行闭环控制。通过数字摄像头获得道路实时图像信息，并进行图像预处理，提取出赛道中心线，通过调节左右两轮差速实现转向和自主寻迹。建立里程计模型和 IMU 模型对系统进行综合定位。通过 MFC 开发串口调试上位机对各种参数进行实时调试。

### 2.2 传感器的选择

基于视觉导航的两轮自平衡小车系统主要通过后轮两电机的前后运动来维持车体的平衡，利用陀螺仪和加速度计获得车体倾角和角速度，并通过卡尔曼滤波进行数据融合。采用摄像头作为传感器进行视觉导航，通过采集道路图像信息获取车身位置和偏移量，通过两电机的差速实现车子的转弯和自主寻迹。

#### 2.2.1 姿态传感器的选择

为实现两轮车的动态自平衡和运动控制，首先必须得到足够精确的车身倾角信息。根据机器人的应用环境，一般选择陀螺仪、加速度计、电子罗盘、超声波传感器组和光电接近探测器等传感器来采集两轮自平衡机器人的姿态信息。表 2.1 对这几种测量姿态的传感器的性能进行了比较。

表 2.1 常用测量姿态传感器性能比较

传感器	被测量	优点	缺点
倾角计	角度	静态、低速运动性能好，精度高，无累积误差	动态响应慢，不适合跟踪高速动态运动，频带窄，更新频率慢
加速度计	加速度	静态、低速运动性能精度高，更新频率快	动态响应慢，高速运动有误差，不适合跟踪高速动态运动
陀螺仪	角速度	动态响应性能好，频带宽，更新频率快	测量姿态时存在累积漂移误差，不适于长时间单独工作
电子罗盘	角度	静态、低速运动性能好，精度高，无漂移误差	动态响应慢，频带窄，受外界磁场影响大，不适于室内导航

由于该自平衡小车系统是大滞后的非线性系统，对动态性能和稳态精度的要求很

高。在此选用 IMU 传感器中的加速度计和陀螺仪测量倾角信息。

### 2.2.2 导航方式及寻迹传感器的选择

目前在移动机器人领域较为先进的导航方式有 GPS 导航, IMU 导航, 超声波导航, 激光雷达导航, 视觉导航, 光电导航, 电磁导航。由于我们的导航时需要小车在任意给定的赛道上自主沿着赛道中心线稳定行驶, 属于有轨迹导航。其中视觉导航, 光电导航, 电磁导航可以实现在任意轨迹上的自主寻迹导航。

表 2.2 三种有轨迹导航方式对比

导航方式	优点	缺点
光电导航	1)检测信息速度快 2)软件数据处理简单	1)道路参数检测精度低 2)检测前瞻距离短 3)占用 MCU 端口资源较多 4)容易受到外界光线影响
电磁导航	1)成本低 2)电路设计相对简单 3)检测信息速度快 4)受外界影响小	1)道路参数检测精度低 2)检测前瞻距离短
视觉导航	1)检测前瞻距离远 2)检测范围宽 3)检测道路参数多 4)占用 MCU 资源少	1)电路相对设计复杂 2)检测信息更新速度慢 3)软件处理数据较多

由于电磁导航的前瞻受到限制, 而光电导航硬件过于复杂, 不适合两轮自平衡小车的机械结构的优化设计。基于摄像头的视觉导航, 由于硬件简单, 大前瞻, 广视野, 能获得丰富的道路信息, 在此选择摄像头采集赛道的实时信息来实现小车的自主寻迹。

目前市面上常见的摄像头主要有 CCD 和 CMOS 两种: CCD 摄像头具有对比度高、动态特性好的优点, 但需要工作在 12V 电压下, 对于整个系统来说过于耗电, 且图像稳定性不高; CMOS 摄像头体积小, 耗电量小, 图像稳定性较高。因此, 经过实验论证之后我们决定采用 CMOS 摄像头。

对于 CMOS 摄像头分为数字和模拟两种。数字摄像头 OV7620 可以直接输出 8 路数字图像信号, 使主板硬件电路的简化成为可能, 且能够达到 60 帧/S 的帧速率, 最终我们选择了 CMOS 数字图像传感器的方案。

本设计选用 Omnivision 公司的 CMOS 图像传感器 OV7620 摄像头采集赛道信息, 输出为黑白全电视信号 PAL 制式模拟信号, 每秒 30 帧, 电视扫描线为 625 线, 奇场在前, 偶场在后。OV7620 摄像头最大分辨率为 640×480 像素, 内置 10 位双通道 A/D 转换器, 输出 8 位图像数据;具有自动增益和自动白平衡控制, 能进行亮度、对比度、饱和度、Y

校正等多种调节功能，其视频时序产生电路可产生行同步、场同步、混合视频同步等多种同步信号和像素时钟等多种时序信号，5V 电源供电，工作时功耗<120mW，待机时功耗<10uW。能够满足本系统道路识别的要求。

## 2.3 方案的论证

### 2.3.1 平衡控制的方案选择

方案一：假定系统的动力最初都来源于后轮的两个电机，通过电机的正反转来维持车子的平衡，通过电机的控制实现速度的控制，通过两电机的差速来实现转向的控制。最后将平衡控制信号，速度控制信号，转向控制信号一起叠加后加载到后轮两个电机上。可实现小车直立行走和自主寻迹。但是由于三个控制环都是耦合在一起的，方向控制和转速控制不能对平衡控制有太大的影响。且平衡控制的优先级最高。

方案二：由于平衡控制和速度控制两个控制环之间的耦合性很大，导致电机不能变化过大而对平衡造成破坏。考虑将直立控制和速度控制两个控制环进行解耦，直立控制仅依靠外力来维持平衡。在此可以在车子的中心位置安装一舵机，并在舵机的输出处安装一两端挂有重物的连杆，通过舵机带动连杆的移动实现自动调节车体前后重心的目的。让车体重心始终保持在正中心的位置便可保持车体的平衡。电机只负责前进和转向。这样一来，速度控制和转向控制可以有更大的稳定裕度。

经实践验证，第二种方案在平衡控制方面系统的响应速度很慢，平衡的稳定性和抗干扰性很难得到保障。最终选择方案一，系统的动力只依靠后轮的两个直流电机。通过控制后轮两个电机的转速来实现平衡控制，速度控制和转向控制。

### 2.3.2 双轮测速方式

双轮测速共有 3 中不同的方案，分别为：用数字模拟开关控制 IO 口对脉冲累加器来分时读取，一路内部脉冲累加器和一路外部计数器分别读取，两个外部计数器分别读取。3 种方案的对比如表 2.3 所示。

表 2.3 三种测速方式的比较

	数字模拟开关	脉冲累加器和外部计数器	两个外部计数器
读取方式	用 PT7 分时读取	用 IO 口和 PT7 分别读取	分别用 IO 口读取
优点	占用资源少	测速比较同步	测速几乎同步
缺点	测速不同步	无	占用资源多

综上所述，为了让两轮的测速既尽量保持同步又不占用过多资源，在此选择一路内部脉冲累加器和一路外部计数器的方式。

### 2.3.3 系统定位方式

定位的目的是检测系统当前的位置和姿态，常用于定位的传感器有 GPS，电子罗盘，光电编码器，加速度计，陀螺仪等等。定位技术分为两类：相对定位和绝对定位。相对定位是由编码器，IMU 传感器，里程计等传感器与机器人前一时刻的位姿来推出下一时刻的位姿。绝对定位是采用摄像头和激光超声波等传感器，然后根据已知位置的标志物进行定位。需要环境的先验知识，由于单传感器都有局限性。因此采用多个传感器进行融合以达到较准确的定位技术。下面分别介绍一下基于里程计的定位和基于 IMU 的航位推算算法。

本系统采用基于里程计和 IMU 传感器相结合的方式。

### (1) 里程计定位

基于里程计的定位可以分为基于速度的定位和基于运动轨迹的定位。由编码器可以得到轮子的里程或速度信息。利用不同轮上的里程或速度信息，可以计算出移动机器人的位置和姿态。如果能够较准确地得到移动机器人的轨迹，就可以实现较高精度的定位。利用的移动机器人，进行了实验可以发现，由于利用两轮驱动小车的编码器信息推算定位的累积误差比较大，因此不适合单独进行定位。

### (2) IMU 传感器定位

惯性传感器是移动机器人常用的一类内部传感器。较常见的有陀螺仪和加速度计。利用陀螺仪可以计算出移动机器人的姿态，利用加速度计可以计算出移动机器人的位置。但是由于陀螺仪和加速度计的漂移误差很大，也不能单独使用。

但是由于编码器和 IMU 传感器无法单独进行系统的准确定位，因此本系统采用双编码器与 IMU 结合的方式对车子进行准确定位。

## 2.3.4 多传感器融合技术

多传感器信息融合通过增加各类传感器信息的互补性，对环境变化的适应性，使移动机器人能够获取完成某一任务所需的完备信息，实现各种复杂、动态、不确定环境下的自主性，提高决策的正确性和导航控制的鲁棒性。水下机器人对周围环境感知的传感器有避碰声纳、水下摄像机和成像声纳等。具体融合算法如信息加权平均、低通平滑滤波、互补滤波、卡尔曼滤波、概率方法、模糊逻辑、信任度量、可能性度量、证据理论、神经网络、灰色系统理论等等，根据各自的特点和适当的场合采用不同的方法。

在测量车体倾角时，需要对加速度计和陀螺仪进行有效滤波得到精确角度和角速度。常用滤波算法有低通滤波，一阶互补滤波，二阶互补滤波，卡尔曼滤波。

低通滤波：对陀螺仪进行零漂低通滤波，取 100 次均值得到车体角速度，结合加速度计的角度值和由陀螺仪积分得到的角度值，对加速度低通滤波，20 次均值得到角度的准确值。

互补滤波：利用加速度计所获得的角度信息与陀螺仪积分后的角度进行比较，将

比较的误差信号经过比例放大之后与陀螺仪输出的角速度信号叠加之后再积分可以输出准确的角度。

卡尔曼滤波：卡尔曼滤波器是一种基于概率模型进行状态和参数估计的有效方法，由一系列递归数学公式描述。它们提供了一种高效可计算的方法来估计过程的状态，并使估计均方误差最小。卡尔曼滤波器应用广泛且功能强大，它可以估计信号的过去和当前状态，甚至能估计将来的状态。

低通滤波可以使角度输出更加平滑，但会使得信号无法实时反映车模倾角变化，从而减缓对于小车车轮的控制。另一方面也会将小车角速度变化的信息滤掉。互补滤波的平滑效果很好，响应也很灵敏，但是受陀螺仪温漂的影响比较大。卡尔曼滤波的收敛速度和滤波效果平衡得很好，只是运算耗的时间要长，而且卡尔曼滤波对陀螺仪的精度要求不是很高。在综合考虑之后，本设计在倾角的加速度计和陀螺仪的融合算法上采用卡尔曼滤波。



## 3 系统原理概述

### 3.1 系统的任务分解

相比四轮着地的小车，两轮平衡车的控制任务更为复杂。为了能够方便找到解决问题的办法，首先将复杂的问题分解成简单的问题进行讨论。由于车子运行的动力都来自于车模的两个后轮，后轮转动由两个直流电机驱动。因此从控制角度来看，车模作为一个控制对象，它的控制输入量是两个电机的转动速度。车模运动控制任务可以分解成以下三个基本任务：

- (1) 控制小车直立：通过控制两个电机正反向运动保持车模直立状态；
- (2) 控制小车速度：通过调节小车的倾角来实现车模速度控制，实际上最后还是演变成通过控制电机的转速来实现车轮速度的控制。
- (3) 控制小车转向：通过控制两个电机之间的转动差速实现小车转向控制。

小车直立和方向控制任务都是直接通过控制小车两个后轮驱动电机完成的。假设小车电机可以虚拟地拆解成两个不同功能的驱动电机，它们同轴相连，分别控制小车的直立平衡、左右方向。在实际控制中，是将控制小车直立和方向的控制信号叠加在一起加载电机上的。

三个分解后的任务各自独立进行控制。由于最终都是对同一个控制对象（小车的电机）进行控制，所以它们之间存在着耦合。为了方便分析，在分析其中之一时假设其它控制对象都已经达到稳定。比如在速度控制时，需要小车已经能够保持直立控制；在方向控制的时候，需要小车能够保持平衡和速度恒定；同样，在小车平衡控制时，也需要速度和方向控制也已经达到平稳。这三个任务中保持小车平衡是关键。三个任务执行的优先级为：平衡控制 > 速度控制 > 转向控制。由于小车同时受到三种控制的影响，从小车平衡控制的角度来看，其它两个控制就成为它的干扰。因此对小车速度、方向的控制应该尽量保持平滑，以减少对于平衡控制的干扰。以速度调节为例，需要通过改变小车平衡控制中小车倾角设定值，从而改变小车实际倾斜角度。为了避免影响小车平衡控制，这个小车倾角的改变需要非常缓慢的进行。

### 3.2 平衡控制原理

控制车模平衡的直观经验来自于人们日常生活经验。一般的人通过简单练习就可以让一个直木棒在手指尖上保持直立。这需要两个条件：一个是托着木棒的手掌可以移动；另一个是眼睛可以观察到木棒的倾斜角度和倾斜趋势（角速度）。通过手掌移动抵消木棒的倾斜角度和趋势，从而保持木棒的直立。这两个条件缺一不可，实际上就是控制中的负反馈机制。

该两轮小车的平衡控制也是通过负反馈实现，通过控制车子两个后轮的前后转动，抵消倾斜的趋势便可以保持车体直立。如图 3.1 所示，车体向前倾斜，车轮向前加速运

行。车体向后倾斜，车轮向右加速运行。

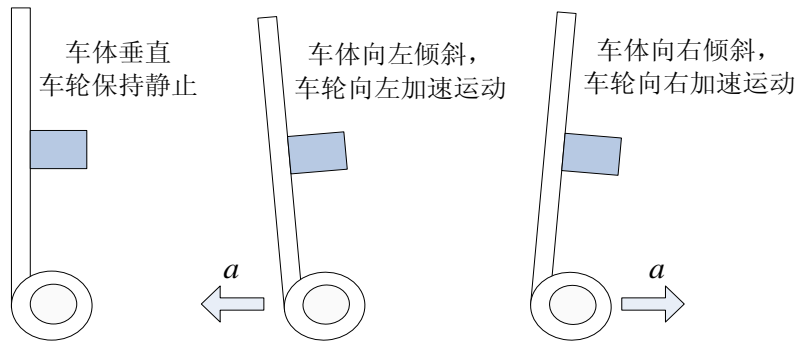


图 3.1 通过车轮的运动来保持车体的平衡

下面通过对倒立摆进行动力学建模来类比说明保持小车平衡的控制规律。

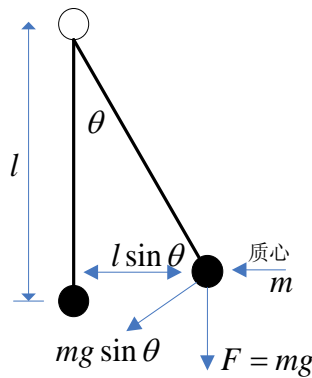


图 3.2 普通单摆受力分析

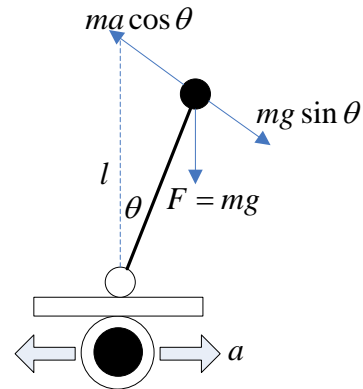


图 3.3 在车轮上倒立摆受力分析

对于普通单摆，当物体离开垂直的平衡位置之后，便会受到重力与悬线的作用合力，驱动重物回复平衡位置。在偏移角度很小的情况下，回复力与偏移的角度之间大小成正比，方向相反。在此回复力作用下，单摆便进行周期运动。在空气中运动的单摆，由于受到空气的阻尼力，单摆最终会停止在垂直平衡位置。空气的阻尼力与单摆运动速度成正比，方向相反。阻尼力越大，单摆越会尽快在垂直位置稳定下来。

倒立摆之所以不能像单摆一样可以稳定在垂直位置，就是因为它偏离平衡位置的时候，所受到的回复力与位移方向相同，而不是相反！因此，倒立摆便会加速偏离垂直位置，直到倒下。

如何通过控制使得倒立摆能够像单摆一样，稳定在垂直位置呢？要达到这一目的，只有两个办法：一个是改变重力的方向；另一个是增加额外的受力，使得恢复力与位移方向相反才行。显然能够做到的只有第二种方法。由于车子的动力全部来自于后轮的两电机，通过控制倒立摆底部车轮，使得它作加速运动。这样站在小车上（非惯性系，以车轮作为坐标原点）分析倒立摆受力，它就会受到额外的惯性力，该力与车轮的加速度方向相反，大小成正比。这样倒立摆所受到的回复力为：

$$F = mg \sin \theta - ma \cos \theta \approx mg \theta - mk_1 \theta \quad (3-1)$$

式中，由于 $\theta$ 很小，可进行线性化处理。假设负反馈控制中，车轮加速度 $a$ 与偏角 $\theta$ 成正比，比例为 $k_1$ 。如果比例 $k_1 > g$ ，那么回复力的方向便于位移方向相反了。

此外，为了使得倒立摆能够尽快地在垂直位置稳定下来，还需要增加阻尼力。虽然存在着空气和摩擦力等阻尼力，相对阻尼力比较小。因此需要另外增加控制阻尼力。增加的阻尼力与偏角的速度成正比，方向相反。因此式（3-1）可变为

$$F = mg\theta - mk_1\theta - mk_2\theta' \quad (3-2)$$

按照上面的控制方法，可把倒立摆模型变为单摆模型，能够稳定在垂直位置。因此，可得控制车轮加速度的控制算法为：

$$a = k_1\theta + k_2\theta' \quad (3-3)$$

式中， $\theta$ 为车模倾角， $\theta'$ 为角速度； $k_1$ ， $k_2$ 均为比例系数；两项相加后作为车轮加速度的控制量。只要保证在 $k_1 > g$ ， $k_2 > 0$ 的条件下，可以使得小车像单摆一样维持在直立状态。其中有两个控制参数 $k_1$ ， $k_2$ ， $k_1$ 决定了小车是否能够稳定到垂直平衡位置，它必须大于重力加速度； $k_2$ 决定了小车回到垂直位置的阻尼系数，选取合适的阻尼系数可以保证小车尽快稳定在垂直位置。

在角度反馈控制中，与角度成比例的控制量是称为比例控制；与角速度成比例的控制量称为微分控制（角速度是角度的微分）。因此上面系数 $k_1$ ， $k_2$ 分别称为P参数和D参数。其中D参数相当于阻尼力，可以有效抑制车模震荡。

总结控制小车直立平衡的条件如下：

- （1）能够精确测量车模倾角 $\theta$ 的大小和角速度 $\theta'$ 的大小；
- （2）可以控制车轮的加速度。

### 3.3 速度控制原理

小车运行速度和加速度是通过控制车轮速度实现的，车轮通过小车两个后轮电机经由减速齿轮箱驱动，因此通过控制电机转速可以实现对车轮的运动控制。

电机的运动控制有三个作用：

- （1）通过电机加速度控制实现小车平衡稳定。
- （2）通过电机速度控制，实现小车恒速运行和静止。通过速度控制，可以提高小车稳定性。
- （3）通过电机差速控制，可以实现小车方向控制。

通过安装在电机输出轴上的光电编码器来测量得到小车的两后轮速度。

假设车模在上面直立控制调节下已经能够保持平衡了，则只要通过控制车模的倾角就可以实现速度的控制。给定车模直立控制的角度设定值，在角度控制调节下，车体将会自动维持在一个角度。通过前面车模直立控制算法可以知道，车模倾角最终是跟踪重力加速度 Z 轴的角度。因此车模的倾角给定值与重力加速度 Z 轴角度相减，便可以最终决定车模的倾角。

为了简化分析，首先忽略了由于倾角控制引起的车模速度的变化。车模的速度最终由倾角所产生的加速度决定，速度的给定会影响到小车往前倾的角度给定。因此将倾角的进行积分便可以得到车模的运动速度。此时小车速度控制框图如图 3.4。

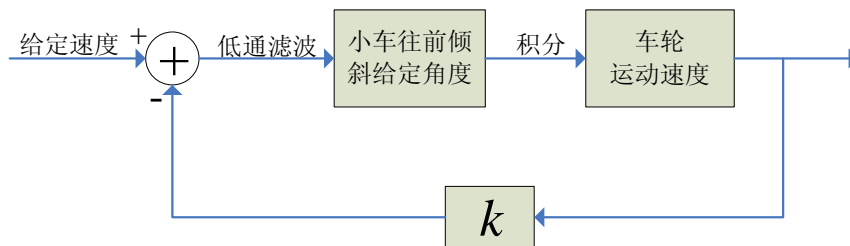


图 3.4 车模运动速度控制简化模型

结合小车角度的闭环控制，将角度控制闭环与速度控制闭环叠加在一起边可实现速度和角度的综合控制。但要注意，速度环的极性为“负”，原来的负反馈相当于正反馈。因为原来在直立控制下的车模速度与车模倾角之间传递函数具有非最小相位特性，在反馈控制下容易造成系统的不稳定性。

### 3.4 摄像头的工作原理

本设计的摄像头是用光荷耦合器件将光信号转换为电信，按一定的表示方法表示出来，再通过电视机或其它机器解读出来。输出为 PAL 制式信号，每秒 30 帧完整图像，每帧图像分奇偶场，有场同步脉冲信号，根据不同像素，每场有不同的行数，场、行脉冲附近部分都有无效的场、行消隐区。一帧图像的信号组成图如图 3.5 所示。当摄像头扫描完一行，视频信号端就输出一低于最低视频信号电压的电平，这称作行消隐区，紧接着每行图像对应的电压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（因为摄像头是隔行扫描的方式），开始扫描新的一行，如此下去，直到扫描完该场的视频信号，接着就会出现一段场消隐区。此区中有若干个复合消隐脉冲（简称消隐脉冲），在这些消隐脉冲中，有个脉冲它远宽于（即持续时间长于）其他的消隐脉冲，该消隐脉冲又称为场同步脉冲，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部分和下一场的开始部分，得等场消隐区过去，下一场的视频信号才真正到来。

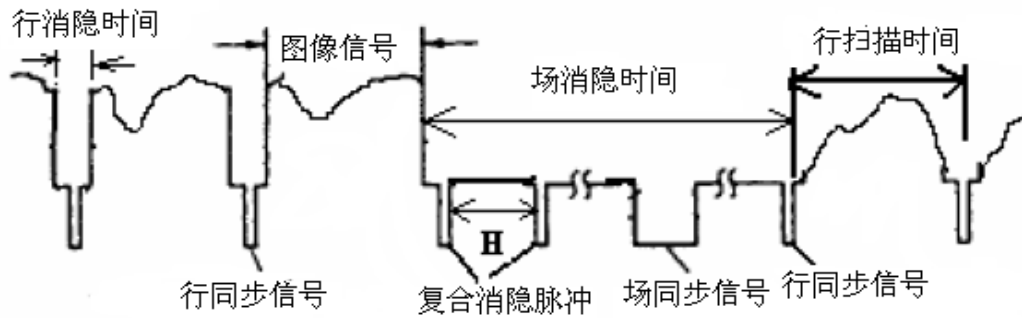


图 3.5 摄像头信号组成图

### 3.5 转向控制原理

给左右轮电机施加控制电压，驱动左右轮运动，就能实现机器人的前进和转向。小车的前进速度是两轮速度之和的一半，而转向速度则与两轮速度之差成正比。因此，两轮电压之和决定了机器人前进速度，两轮电压之差决定了机器人转向速度。不难发现机器人的前进状态与两轮电机电压之差无关，转向状态与左右电机电压之和的一半无关。

系统调试的环境是一铺有引导线的任意形状的跑道，包括直道，大弯，小弯，十字弯等等。跑道两边贴有两道宽度约为 2.5cm 的黑线，通过摄像头采集赛道的整幅图像信息，进行滤波去噪，提取出赛道两边的两条黑线，根据两黑线的位置计算出车体偏移赛道中心线的位置，并得到方向偏差控制量。

根据两编码器测得的速度之差，可得到车体的转向角速度。对方向控制量和车体转动角速度进行 PD 控制可得到转向控制信号。根据相应的转向控制信号调节左右两个轮子的差速来实现转弯和寻迹。

结合小车角度的闭环控制，速度的闭环控制，将角度闭环，速度闭环与差速闭环叠加在一起边可实现直立，速度和转向的综合控制。控制结构图如图 3.6 所示。

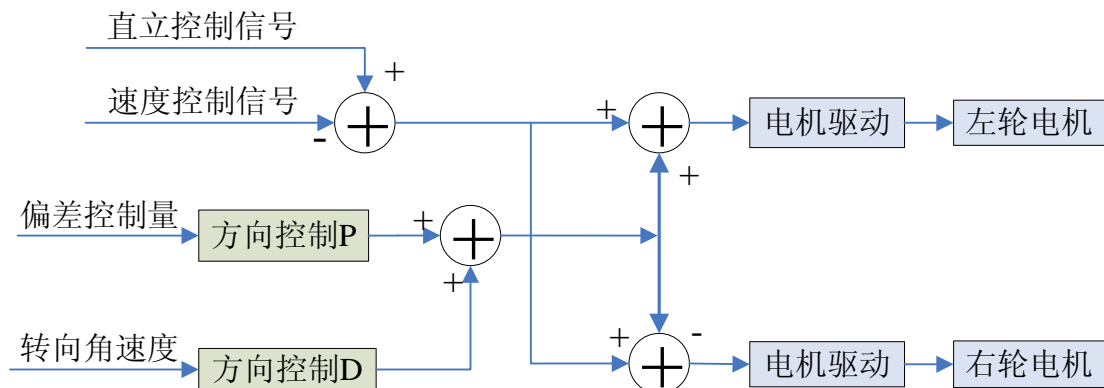


图 3.6 转向控制系统结构框图

### 3.6 两轮驱动系统定位模型

#### 3.6.1 里程计模型

在编码器采样时间足够小的情况下，将小车的运动轨迹看做圆弧型（直线运动时，运动半径为无穷大）。记  $(x_k, y_k, \theta_k)^T$  和  $(x_{k+1}, y_{k+1}, \theta_{k+1})^T$  分别为  $k, k+1$  时刻系统的位姿。 $v_L, v_R$  分别为车子左右两轮的线速度， $L$  为左右两轮之间的距离。里程计模型如图3.7所示：

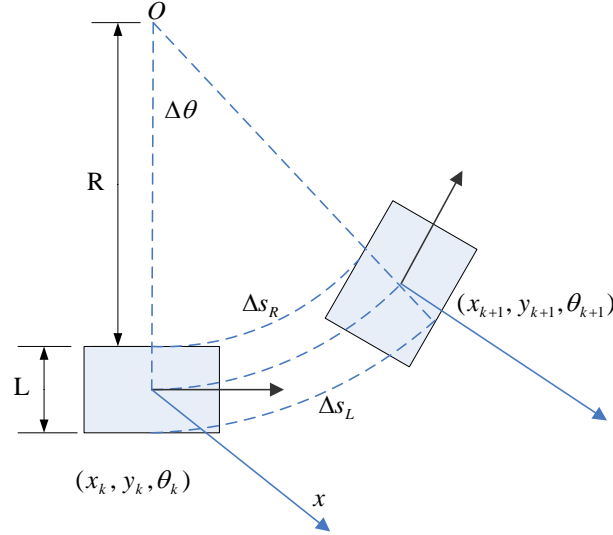


图 3.7 两轮驱动小车里里程计模型

通过光电编码器测量得到左右轮在测量时间  $\Delta t$  内的行进距离  $\Delta s_R$  和  $\Delta s_L$ ，相应地得到右轮速度  $v_R$  和左轮速度  $v_L$ ：

$$v_R = \frac{\Delta s_R}{\Delta t}, \quad v_L = \frac{\Delta s_L}{\Delta t}$$

得到车体的线速度为： $v = \frac{v_R + v_L}{2}$

记  $\Delta \theta$  为在单位采样时间  $\Delta t$  内车子转过的角度，

$$\text{又 } L\Delta \theta = \Delta s_R - \Delta s_L, \quad \text{故可得车身转向角速度 } \omega = \dot{\Delta \theta} = \frac{v_R - v_L}{L}$$

得到里程计定位模型如下：

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} v \cdot \cos(\theta_{k+1}) \Delta t \\ v \cdot \sin(\theta_{k+1}) \Delta t \\ \omega \Delta t \end{bmatrix} \quad (3-4)$$

$$\text{又 } \theta_{k+1} = \theta_k + \frac{1}{2} \Delta \theta = \theta_k + \frac{s_R - s_L}{2L}$$

$$\text{所以有: } \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \frac{s_R + s_L}{2} \cos(\theta_k + \frac{s_R - s_L}{2L}) \\ \frac{s_R + s_L}{2} \sin(\theta_k + \frac{s_R - s_L}{2L}) \\ \frac{s_R - s_L}{L} \end{bmatrix} \quad (3-5)$$

只要给出小车的初始位置和方位角，根据上述关系便可推算出小车的后轴中点任何时刻的位置和位姿。

由于实际小车的齿轮咬合不紧、机械结构间隙、双轮打滑和编码器采样时间长短等因素，小车系统的里程计模型并不能完全准确地反映其位姿变化。因此在算法中，小车系统的里程计模型只是用于预测车体的下一步位姿，为车体前后时刻的特征匹配提供启发信息。

### 3.6.2 IMU 模型

陀螺仪是一种惯性器件，用于测量其载体的转动角速度和瞬时转动角度。利用高精度陀螺仪测得车体航向角速度，角速度经积分得到系统的位姿角度。加速度计在机器人定位中的应用也很普遍，加速度计可以得到小车前进方向的运动加速度，将由运动加速度的变化量的不断累加得到前进的运动线速度。高精度的光电编码器能够准确测出机器人运动的瞬时位移，将编码器与加速度计和陀螺仪的值进行相互融合，消去各自的误差，得到准确的线速度  $v$ ，转动角速度  $\omega$ ，一个采样周期内车子转过的角度  $\theta$ ，从而能得到系统的位置和位姿。定位框图如图 3.8 所示。

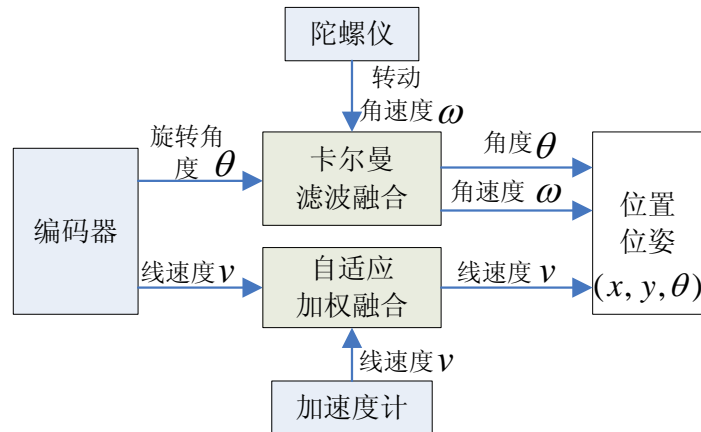


图 3.8 系统定位框图

## 4 系统硬件设计

### 4.1 系统硬件组成

系统的硬件电路中最重要的是系统控制电路的设计。设计小车控制系统的电路，首先需要分析系统的输入、输出信号，然后选择合适的核心控制嵌入式计算机（单片机），逐步设计各个电路子模块，最后形成完整的控制电路。为了实现车模直立行走，需要利用以下资源：

#### （1）AD 转换接口（至少 5 路）

摄像头：可以用于读取模拟摄像头的数据信号。

陀螺仪：两路。一路用于检测小车倾斜角速度，一路用于检测小车转动角速度。

加速度计：两路，测量加速度 Z 轴输出电压和 Y 轴输出电压。

辅助调试：（备用）1 到 3 路，用于车模调试、设置作用。

#### （2）PWM 接口（4 路）

控制左右两个电极双方向运行。由于采用单极性 PWM 驱动，需要四路 PWM 接口。如果采用双极性 PWM 驱动，可以使用两路。

#### （3）定时器接口（1 路）

测量两个电机转速，需要 1 个内部的脉冲累加器和 1 个外部定时器脉冲输入端口。

#### （4）通讯接口（备用）

SCI（UART）：一路，用于程序下载和调试接口；

I2C：（备用）如果选择飞思卡尔公司的数字加速度计，可以通过 I2C 接口直接读取加速度值。

#### （5）IO 接口（备用）

8 路输入：用于采集数字摄像头的 8 位灰度值。

4 到 8 路输入输出：用于测量车轮正反转获小车运行状态显示，功能设置等。

由于飞思卡尔半导体公司的 S12X 系列单片机采用复杂指令集计算机(CISC)架构，集成了中断控制器，具有丰富的寻址方式。中断有 7 个优先级，并且内核支持优先级的调度，最多可以有 117 个中断源。且内部集成有丰富的 AD 转换器和 PWM 发生器。本课题采用 S12X 系列的 16 位处理器 MC9S12XS128 作为核心的控制处理器进行设计。

系统的硬件部分主要由基于 S12 系列微处理器的控制模块、稳压电源模块、倾角传感器模块、直流电机驱动模块、测速模块、图像采集模块、人机交互模块和其它接口电路等系统组成。系统的主控芯片是 MC9S12XS128，由它的最小系统控制各个模块，使得各个模块之间的能够协调工作。通过单片机的内部 AD 采集各个传感器的信号，有控制器进行滤波和信号处理，对车身姿态和当前的赛道环境进行识别，最后通过一定的控制算法将控制量的输出通过 PWM 的形式送给后轮两个电机，以完成系统的所有任务。

车模控制电路整体框图如图 4.1 所示。



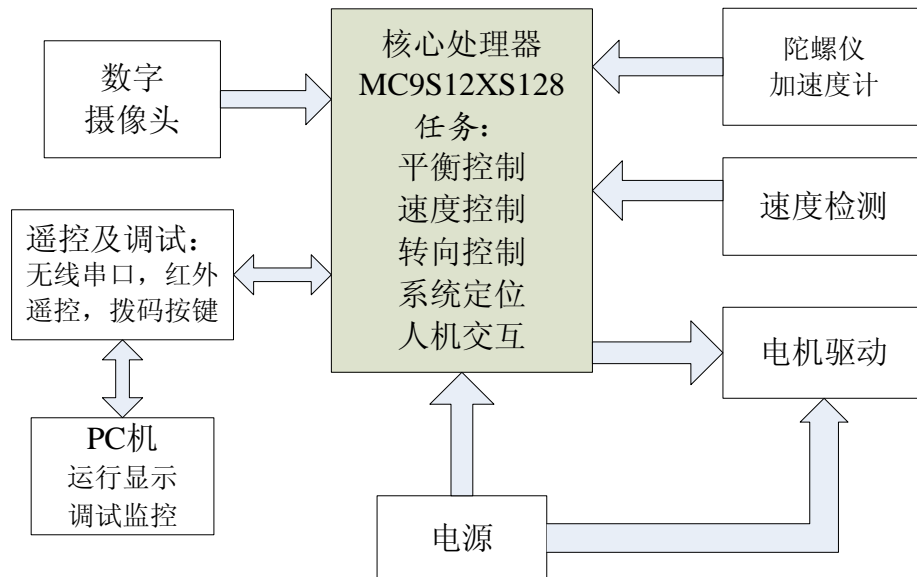


图 4.1 小车控制电路的整体框图

## 4.2 MC9S12XS128 微处理器简介

MC9S12XS128 只是 HCS12X 家族中的一个成员。该器件包括大量的片上处理器和外部 I/O。MC9S12XS128 是一个 16 位器件，有 16 位中央处理单元(128KB Flash)、128KB Flash、8KB RAM、2KB EEPROM 组成片内存储器。同时还包括 2 个异步串行通讯接口 (SCI)、2 个串行外设接口(SPI)和一个 8 通道输入捕捉/输出比较(IC/OC)增强型捕捉定时器(ECT)、1 个 16 通道 10 位 A/D 转换器(ADC)和一个 8 通道 PWM。图 4.1 为 MC9S12XS128 的 112 封装引脚图。

MC9S12XS128 包括 29 个独立的数字 I/O 口(A、B、K 和 E)，其中有 20 个独立的数字 I/O 口具有中断和唤醒功能。另外还具有 2 个 CAN 2.0A/B 标准兼容模块(MSCAN12)和一个 I2C 总线。

MC9S12XS128 具有内部 16 位数据通道，外部总线可以按照 8 位数据总线窄模式操作，所以可以和 8 位数据总线的存储器连接，以降低成本。PLL 电路允许根据系统需求调整电源功耗及性能。

MC9S12XS128 主要特性包括：

- 16 位 128KB Flash;
- 内部存储器：
  - ✧ 128KB Flash;
  - ✧ 8KB RAM;
  - ✧ 2KB EEPROM;
- 外围设备：
  - ✧ 2 个增强型串行通行接口(SCI);
  - ✧ 2 个串行外设接口(SPI);

- ◇ 1 个 I<sup>2</sup>C 总线接口；
- ◇ 2 个 MSCAN 模块；
- ◇ 1 个 16 通道 8/10/12 位 A/D；
- ◇ 8 通道 16 位增强型捕捉定时器(ECT)；
- ◇ 8 通道 8 位 PWM 或 4 通道 16 位 PWM。

由 MC9S12XS128 单片机的电源滤波电路、时钟电路、复位电路、串行通讯接口、BDM 接口和 SPI 接口构成单片机最小系统。各部分的功能如下：

- (1) 时钟电路为单片机提供一个外接的 16 Hz 的石英晶振。
- (2) 串口的 RS-232 驱动电路可实现 TTL 电平与 RS-232 之间的转换。
- (3) BDM 接口允许用户通过该接口向单片机下载和调试程序。
- (4) 供电电路主要是给单片机提供+5 V 的电源。
- (5) 复位电路是通过一个复位芯片给单片机一个复位信号。
- (6) 调试用 LED 灯和单片机的 PORTB 口相连，供程序调试使用。

MC9S12XS128 单片机的最小系统原理图如图 4.2 所示。

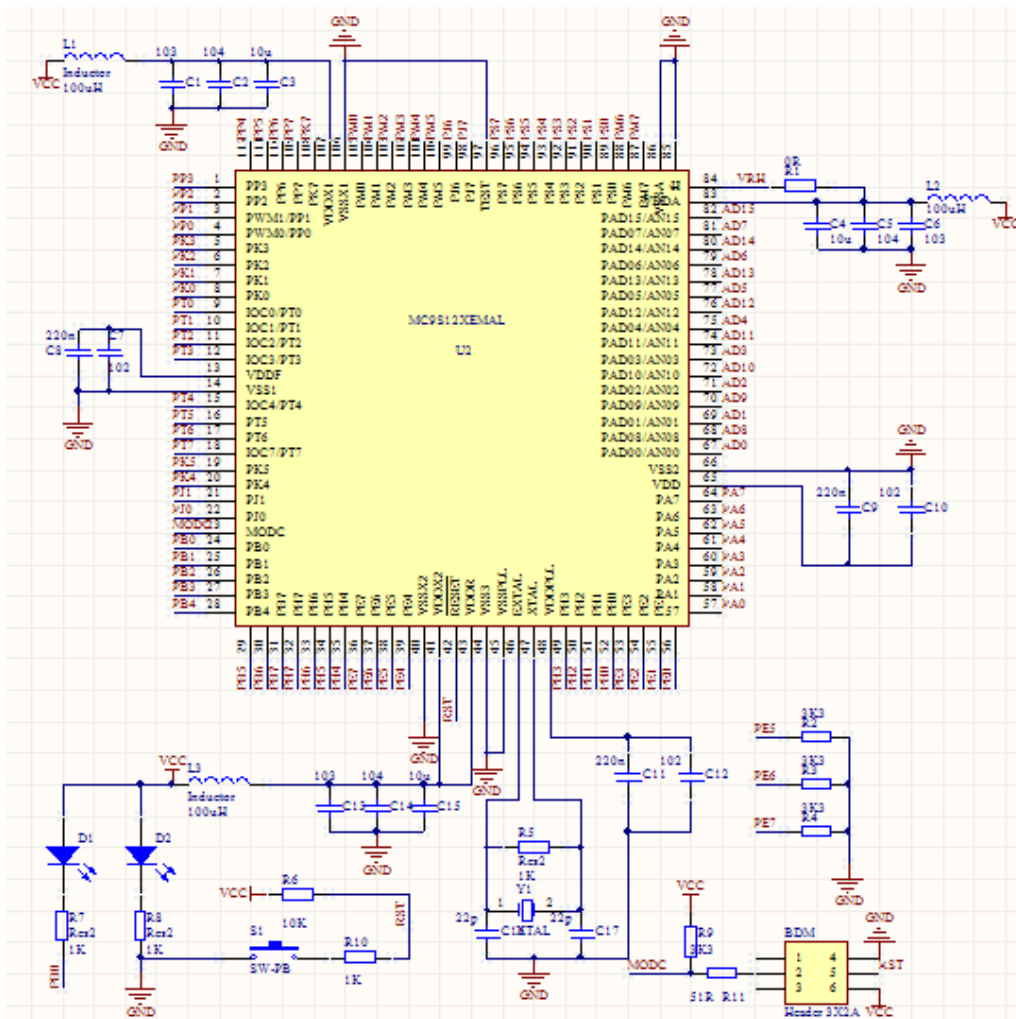


图 4.2 MC9S12XS128 最小系统原理图

### 4.3 系统各硬件模块设计

此小车的硬件系统结构主要由基于 MC9S12XS128 单片机的最小系统模块、稳压电源模块、测速模块、图像采集模块、倾角测量模块、直流电机驱动模块和人机交互模块等组成。

#### 4.3.1 稳压电源模块

为了能使系统能正常工作，就需要对电池电压进行调节。其中，单片机系统、车速传感器电路需要 5V 电压，摄像头电压工作为 5V、陀螺仪和加速度计工作电压为 3.3V。直流电机可以使用 7.2V 2000mAh Ni-cd 蓄电池直接供电。考虑到由于驱动电机引起的电压瞬间下降的现象，因此采用低压降的三端稳压器成为必然。我们采用 AMS1117 3.3 和 LM2940 5.0 作为稳压芯片。经试验电压纹波小，低压降，完全可以满足要求。整个系统的供电模块如图 4.3 所示。

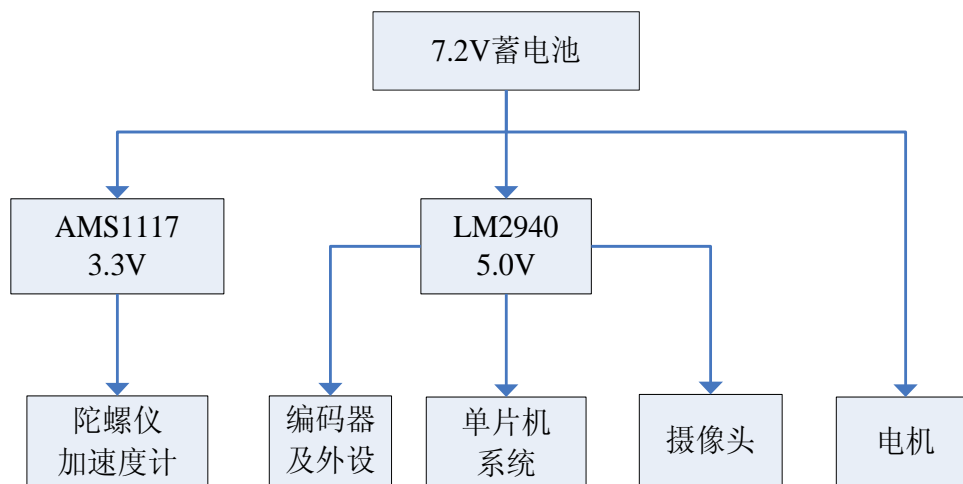


图 4.3 电源模块框图

LM2940CT-5.0 为低压差 5V 稳压管，该电源芯片最大输出电流为 3A，压差在最大输出电流时也不会超过 500mV。它具有外围电路简单、转换效率高、波形稳定、低压差电压、瞬态响应快的优点，是小功率微处理器理想的配套件。LM2940CT-5.0 的稳压电路原理图如图 4.4 所示。

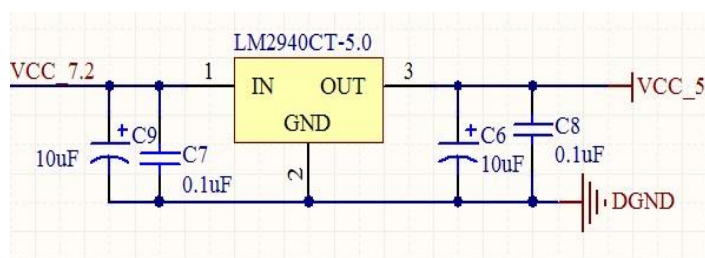


图 4.4 5V 稳压电路原理图

AMS1117-3.3 是一正向低压降三端稳压器，可以直接通过三端稳压输出固定电压 3.3V，精度为 1%，该电源芯片最大输出电流为 1A，低压差，纹波小，并内部集成过流和过热保护功能，满足系统要求。其稳压电路原理图如图 4.5 所示。

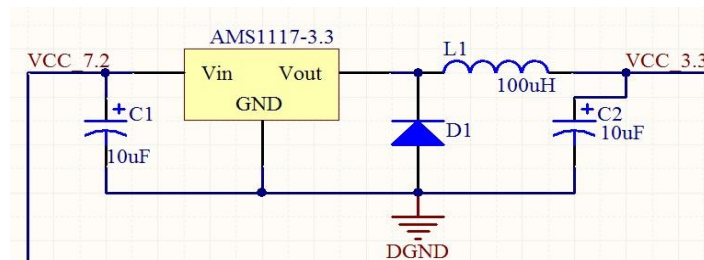


图 4.5 3.3V 稳压电路原理图

### 4.3.2 倾角测量模块

#### (1) 加速度传感器。

加速度传感器可以测量由地球引力作用或者物体运动所产生的加速度。

**MMA7260** 一款三轴低  $g$  半导体模拟加速度计，可以同时输出三个方向上的加速度模拟信号，最大输出灵敏度为  $800\text{mV/g}$ ，只需要测量其中一个  $Z$  轴方向上的加速度值，就可以计算出车模倾角。车模直立时，固定加速度器在  $Z$  轴水平方向，此时输出信号为零偏电压信号。当车模发生倾斜时，重力加速度  $g$  便会在  $Z$  轴方向形成加速度分量，从而引起该轴输出电压变化。当倾角较小时，输出电压的变化可以近似与倾角成正比。

**MMA7260** 是一款三轴低  $g$  半导体加速度计，可以同时输出三个方向上的加速度模拟信号，如图 4.6 所示。

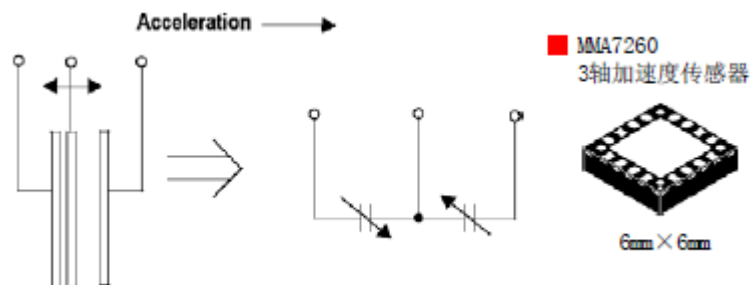


图 4.6 MMA7260 三轴加速度传感器

**MMA7260** 可以测量三个轴的加速度，本设计只需利用  $Z$  轴测量车体倾角，用  $X$  轴测量车子前进方向加速度即可。**MMA7260** 测量加速度的原理图如图 4.7 所示：

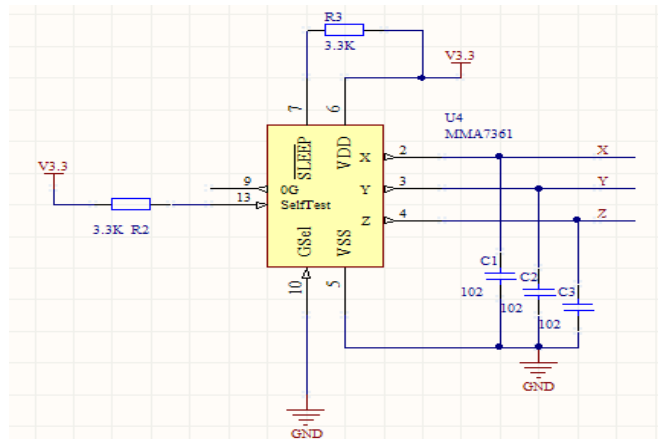


图 4.7 MMA7260 测量加速度的原理图

## (2) 角速度传感器-陀螺仪。

陀螺仪可以用来测量物体的旋转角速度。它利用了旋转坐标系中的物体会受到科里奥利力的原理，在器件中利用压电陶瓷做成振动单元。当旋转器件时会改变振动频率从而反映出物体旋转的角速度。

ENC-03 是一款可以用来测量物体的旋转角速度的模拟陀螺仪，输出的是车模前后方向的角速度，不会受到车体振动影响。因此该信号中噪声很小，在车模上安装陀螺仪，可以测量车模倾斜的角速度，将角速度信号进行积分便可以得到车模的倾角。

由于该款陀螺仪灵敏度为  $0.67\text{mv/deg/s}$ ，经测试，当小车的摆动幅度很大时，其输出的电压范围很小，导致测量的精度不高，而且受其温漂的影响很大。故需要将其低通滤波和放大处理，其硬件原理图如图 4.8 所示。

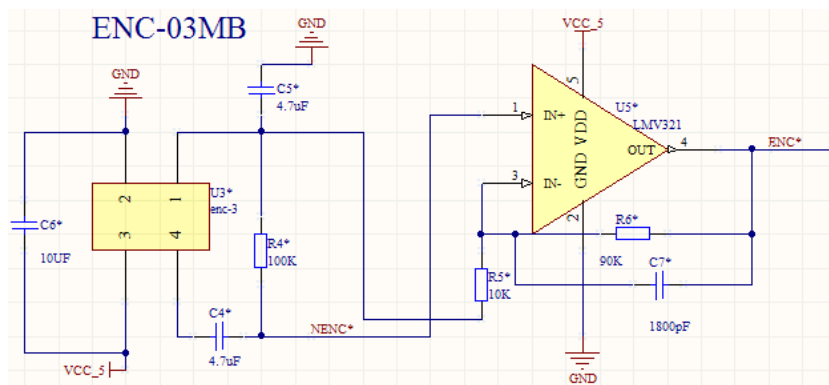


图 4.8 陀螺仪 ENC-03 的放大电路原理图

## 4.3.3 图像采集模块

OV7620 提供的 3 种数据制式，本系统选用 YUV16 位的数据制式且只提取其中的 Y 信号即亮度信息，生成黑白图像。使用 HREF-行同步信号、VSYNC-场同步信号为采集控制信号。飞思卡尔 S12 读取黑白图像的简单连接接口如图 4.9 所示。

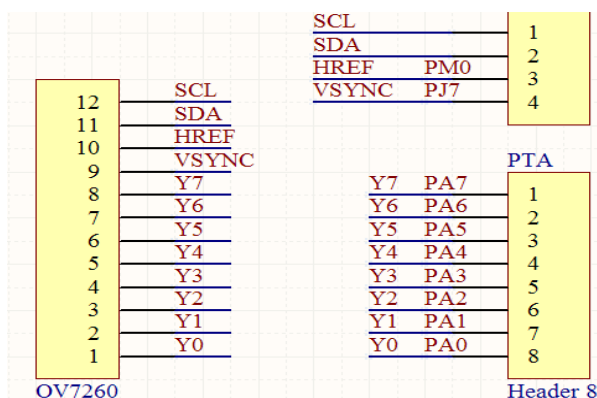


图 4.9: OV7620 摄像头与 S12 单片机的连接端口

#### 4.3.4 直流电机驱动模块

本系统选择了 IR2104 芯片驱动 H 桥来作为直流电机的驱动单元。采用 S12 单片机输出的 PWM 信号来控制直流驱动电机的转速和正反转。这样构成的 H 桥的驱动能力得到了明显的加强。并且在小车需要减速时可以通过自由停车或反向制动来实现。采用 H 桥全桥驱动时，采用 4 个 TO-200 封装的 MOS 管来搭建 H 桥驱动，其响应时间约为 0.05 秒，驱动能力要由于其它器件。而且 MOS 管的最大工作电流可达 16A，其散热性能很不错。同时，为了控制的精度，将 PWM0 和 PWM1，PWM2 和 PWM3，PWM4 和 PWM5，PWM4 和 PWM5 八个 8 位寄存器合成四个 16 位的寄存器 PWM23 和 PWM45。VCC 为电源电压 7.2 伏，通过两路 PWM 信号控制桥来工作。其输出端口 MOT1 和 MOT2 分别接驱动电机的两端。在此系统中需要采用两个相同的 H 桥对两电机分别进行控制。H 桥的硬件电路原理图如图 4.10 所示。

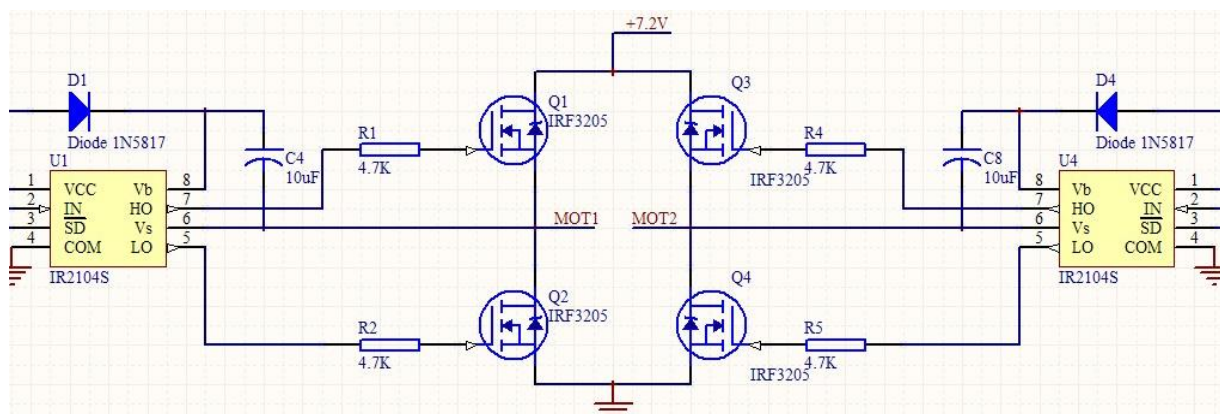


图 4.10 驱动桥电路原理图

#### 4.3.5 双轮测速模块

本系统车速检测单元采用日本 OMRON 公司的 E6A2-CW3C 型光电编码器作为车速检测元件。其精度达到车轮每旋转一周，旋转编码器产生 500 个脉冲，不仅硬件电路简单，而且信号采集速度快、精度高，满足控制精度要求。其硬件电路如图 4.11 所示。编码器的工作电压为 5—24 伏，输出为一系列脉冲。



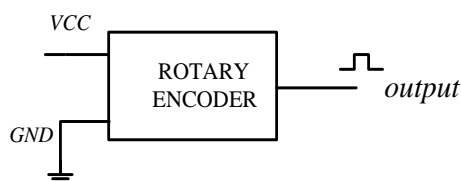


图 4.11 车速检测硬件电路图

E6A2-CW3C 型旋转编码器的输出方式为电平输出，且其输出为 5 线制，分别为电源线，地线和 3 相信号线。本系统将一个旋转编码器的输出接一个 2K 上拉电阻再与 MC9S12XS128 的 I/O 口 PT7 相连，PT7 采用 16 位输入脉冲累加模式对旋转编码器的输出脉冲进行累加计数。另一个编码器的输出接入一外部计数器。在旋转编码器的中轴上安装一个直径为 4cm，齿数 76，传动比 1:1 的齿轮，并将该齿轮与同轴于后轮的传动齿轮咬合。如此则后轮旋转的同时将通过传动齿轮带动旋转编码器一同旋转。此时只需要测量一定时间(10ms)旋转编码器输出的脉冲数就能准确计算出车速。车速的计算公式如公式 (4-1) 所示：

$$v = \frac{\pi d n}{200T} \quad (4-1)$$

式中 ( $d$ 为小车后轮直径, $n$ 为采样时间内旋转编码器产生脉冲数, $T$ 为采样周期)。

外部计数器测速原理图如图 4.12 所示。

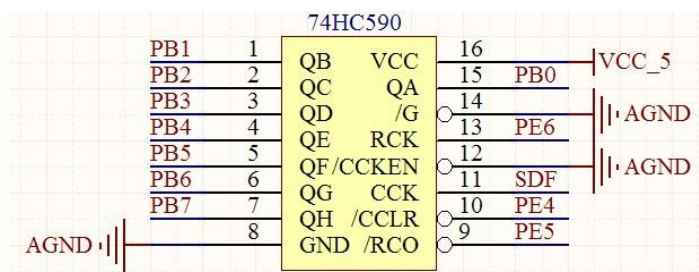


图 4.12 外部计数器芯片 74HC590 测速原理图

由于 500 线的精确对于需要在原点处保持静止的平衡车而言，其精度不够高，在此将 AB 两项的信号先相异或，再送入脉冲累加器或外部计数器进行读取，便可将 A 相的脉冲波形进行了分频，从而相当于使编码器的精度提高了两倍。

速度是一个矢量，不仅具有数值信息，还有方向信息。然而，用上述的脉冲计数法只能获取数值信息，却无法得知车体行驶的方向信息。为此，专门设计了一个鉴相电路来解决此问题。如图 4.13 所示。

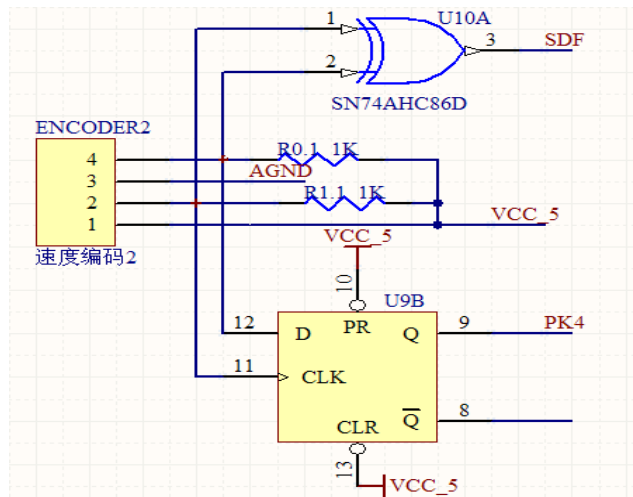


图 4.13 鉴相电路原理图

系统所采用的编码器提供两相输出，即 A 相和 B 相。当智能汽车向前行驶时，编码器从轴侧看是向左旋转的，正转时，A 相超前 B 相约为  $90^\circ$ ，反转时，A 相滞后 B 相约为  $90^\circ$ ，此时两相输出如图 4.14 所示。

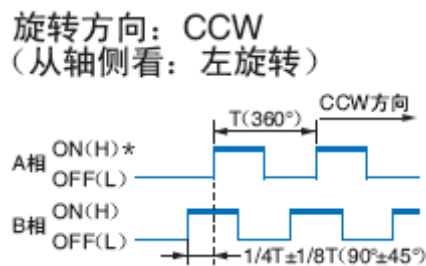


图 4.14 编码器左旋转时两相输出波形图

由波形图可知，此时 A 相滞后 B 相  $90^\circ$ 。本设计利用一片 D 触发器 74HC74 构建了一个鉴相器。在该鉴相器中，以 A 相输出作为输入时钟信号，B 相输出作为输入数据信号。当编码器左旋转时，由于 A 相滞后 B 相  $90^\circ$ ，相当于 A 相上升沿时 B 相总为高电平，根据 D 触发器的工作原理，74HC74 的 Q 脚输出恒为高电平。反之，在编码器右旋转时则输出低电平。这样，就实现了鉴相的功能，如图 4.15 所示。

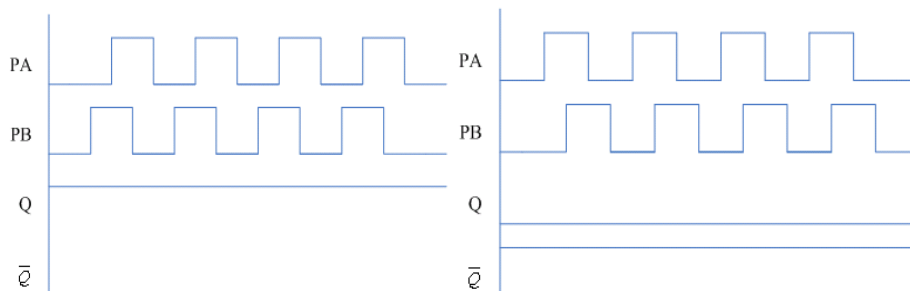


图 4.15 鉴相电路在编码器左、右旋转时的波形图



## 5 系统软件设计

在车子机械和硬件都设计好了之后，最重要的就是软件的设计和系统的调试。开发软件主要任务包括：

- (1) 建立软件工程，配置单片机资源，初步编写程序的主框架；
- (2) 编写上位机监控软件，建立软件编译、下载、调试的环境；
- (3) 编写实现各个子模块，并测试各个子模块的功能正确性；
- (4) 进行系统的综合调试，使小车获得更佳的稳定性和抗扰性能。

### 5.1 算法功能与框架

#### 5.1.1 单片机的硬件资源配置

**锁相环配置：**通过设置锁相环，可以改变单片机的时钟频率。在 MC9S12 单片机中，靠锁相环产生的时钟频率由公式（5-1）得到：

$$PLLCLK = 2 \times OSCCLK \times \frac{(SYNR+1)}{(REFDV+1)} \quad (5-1)$$

式中，OSCCLK 是外部晶体振荡时钟频率，一般为 16 MHz；SYNR 是时钟合成寄存器；REFDV 是时钟分频寄存器。在此将时钟频率设置为 48M。

**PWM输出控制：**脉宽调制模块有8路独立的可设置周期和占空比的8位PWM通道，每个通道配有专门的计数器。该模块有4个时钟源，能分别控制8路信号。通过配置寄存器可设置PWM的使能与否、每个通道的工作脉冲极性、每个通道输出的对齐方式、时钟源以及使用方式（八个8 位通道还是四个16位通道）。两个电机的PWM信号的频率均为10KHz。为了提高控制精度，我们将两路8位通道合并为一个16位通道来控制电机，这样可使电机的控制精度从1/255提高到1/65536。

**ADC 采集通道：**A/D 转换模块主要用于将模拟的视频电压信号转换成对应的数值，或读取陀螺仪和加速度计的电压值。

**ECT 模块：**定时中断及输入捕捉通道主要用于产生周期中断以进行速度采集，其初始化工作主要包括：设定预分频系数；定时器溢出中断使能；定时器使能。

**SCI模块：**串口通信模块设有两个串行通信接口SCI1 和SCI0。使用时，可以对波特率、数据格式（8 位或9 位）、发送输出极性、接收唤醒方式等进行选择。另外，发送和接收可分开使能，模块中还提供了多种避免传输错误的选项。在本次设计，利用串口SCI0，将所需要的信息通过串口传到上位机上进行处理和分析，以提高调试的效率。

**外部中断：**利用 PJ 口作为外部中断用上升沿来采集行中断信号。

**IO：**将 PA 口作为输入采集数字摄像头的视频数据信号的灰度值。PH 作为输入，用来过去拨码开关的状态。PB 口作为输出，通过 LED 用来显示车子实时的状态。PTIM\_PTIM0 作为 IO 输入采集场中断信号。

### 5.1.2 整个系统的软件框架

整个软件的功能模块包括：

- (1) 各传感器信号的采集；
- (2) 电机 PWM 输出；
- (3) 车模运行控制：直立控制，速度控制，方向控制；
- (4) 系统的定位
- (5) 图像的处理及黑线的提取；
- (6) 车模运行流程控制：程序初始化、车模启动与结束、状态监控；
- (7) 车模信息显示与参数设定：状态显示、上位机监控、参数设定等。

上述功能模块可以分成两大类：

第一类包括 1-4 功能，它们属与需要在精确的时间周期内完成，最好在一个周期定时中断里按照一定的时序完成。第二类包括 5-7 功能。它的执行不需要精确的时间周期。可以放在程序的主程序中完成，也可以放在定时器中断里完成。这两类任务之间可以通过全局变量实现相互的通讯。

在该系统中，由于数字摄像头的采用频率为 60HZ，周期为 16.7ms，故图像的采集由 PJ 口的外部中断来单独处理，且图像采集的优先级高于其它内部中断。倾角传感器信号的采集，电机转速的读取，小车直立控制，小车速度控制，小车方向控制等则放在一个 1ms 的定时器中断系统里面，用一个全局变量进行计数，则可实现系统各个子模块有序地执行。图像处理，黑线提取，方向偏差控制量的求取及无线串口数据的接收与发送则放在主程序中。

软件系统首先是对各个寄存器和各功能模块进行初始化。包括锁相环，PWM 通道，ADC 通道，ECT 设置，SCI 设置，IRQ 设置，及 IO 口的设置。功能模块包括 PID 参数的初始值，图像数组的初始化。

初始化完成后，读取人机交互模块的参数，包括拨码的状态和上位机传回的数据。然后开始进入无线循环。如果有中断来临，首先进入中断优先级最高的图像采集程序，再进入 1ms 的定时器中断，在定时器中断服务程序中，按照全局计数变量的取值依次按顺序执行各周期的任务。中断没有到来时，则执行图像处理和黑线提取程序。程序在主循环中不停发送监控数据，通过串口发送到上位机进行实时监控。

跌倒判断可以通过小车倾角是否超过一定范围进行确定，如果小车跌倒，则停止车模运行。图 5.1 是主程序流程图。

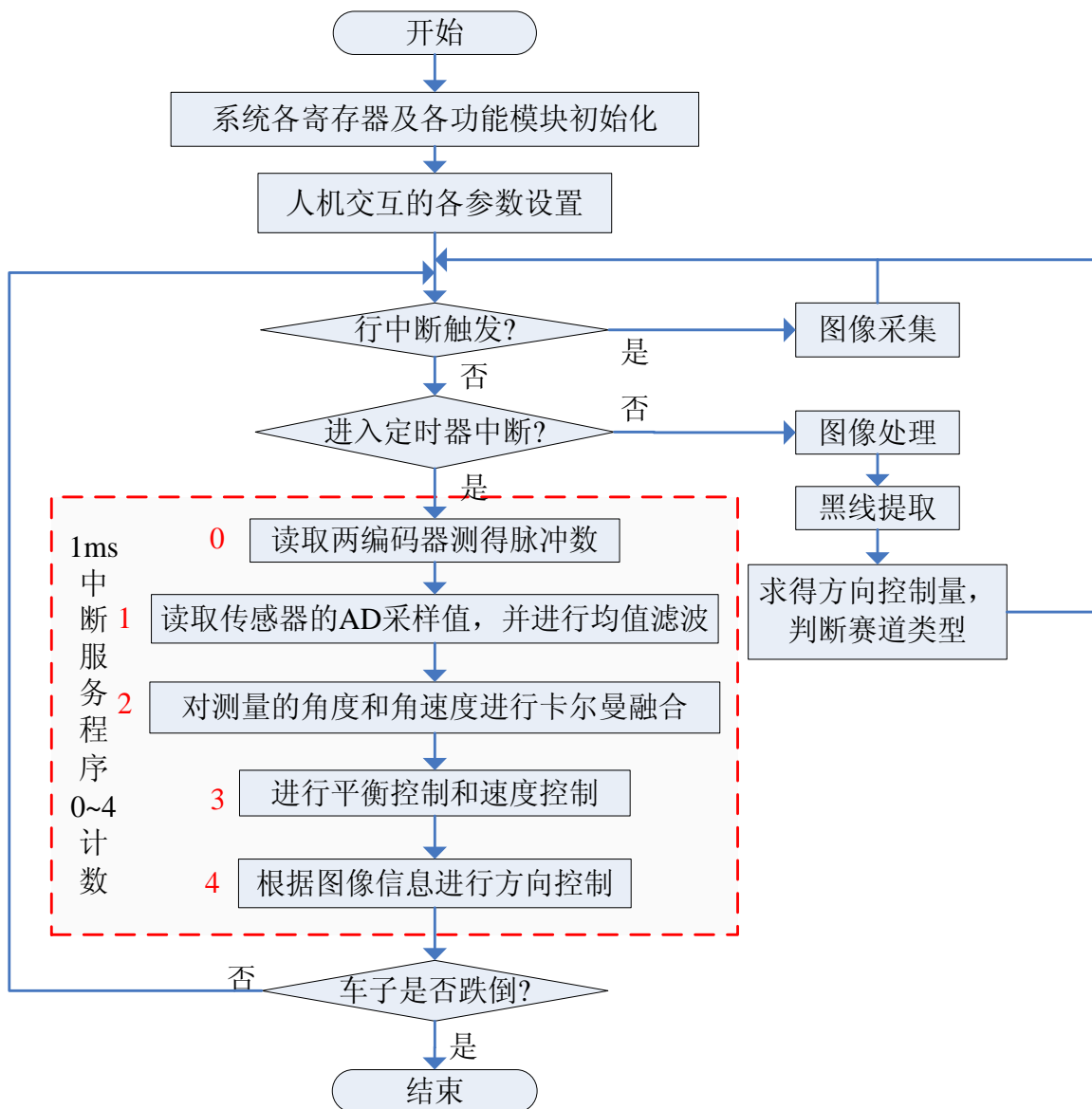


图 5.1 主程序流程图

## 5.2 卡尔曼滤波算法设计

### 5.2.1 倾角传感器误差的分析

加速度计：加速度计安装在待测对象上，加速度计将输出该轴方向的重力加速度分量值，根据分量值和反三角函数，很容易就得到倾角值。

陀螺仪：陀螺仪安装在待测对象上，将输出对象在其旋转方向上的角速度信号。通过对角速度进行积分便可以得到小车倾斜的角速度。

应该说，通过这两种传感器就能得到姿态信息，但是这两种传感器都有缺陷，该缺陷直接影响的所采集到的信息的可靠性。在对加速度计和陀螺仪的实际的性能的测试中，发现在实际小车运行过程中，由于车子本身的运动所产生的加速度会产生很大的干

扰信号叠加在上述测量信号上，使得输出信号无法准确反映车模的倾角。由于两轮自平衡小车对小角度的测量有较高要求，所以这个噪声已经影响了输出的信号的可靠性。陀螺仪的输出本身精度较好，但是对陀螺仪的长时间测试中，发现当输入固定时，陀螺仪输出有一定量的漂移。该漂移值如果不加处理，会严重影响信号的可靠性。而且角速度经过积分运算之后，变化形成积累误差。这个误差会随着时间延长逐步增加，最终导致电路饱和，无法形成正确的角度信号。

由此可见，加速度计和陀螺仪都不适宜单独的用于检测两轮自平衡小车的姿态信息。故采用卡尔曼滤波对并加速度计和陀螺仪的数据进行卡尔曼滤波有效融合，以得到准确的角度信息。

### 5.2.2 卡尔曼滤波器简介

1960 年，R.E.Kalman 在一篇论文中介绍了一种应用于离散线性滤波的迭代算法，这就是后来得到广泛应用的著名的卡尔曼滤波。

测量值混有噪声，而且系统的行为也可能受到随机干扰。卡尔曼滤波的目的就是要尽可能的减少噪声的影响，并且从含有噪声的测量值中得到系统状态的最优估计。同时，它的解是递归计算的，其状态的每一次更新估计都由前一次估计和新的输入数据计算得到，因此只需存储前一次估计，用计算机计算起来非常高效。这些优点使得卡尔曼滤波适合于本文的倾角及角速度状态估计。

卡尔曼滤波器解决了线性离散时间控制系统的状态估计问题。一般而言这个系统可用离散线性差分方程描述。

(1) 线性离散系统状态方程。

$$x(k) = Ax(k-1) + Bu(k-1) + w(k-1) \quad (5-2)$$

式中  $x(k) \in R^n$  ——  $k$  时刻系统的状态变量；

$u(k)$  ——  $k$  时刻系统的控制变量；

$w(k) \in R^n$  —— 过程噪声，可建模为零均值的白噪声过程，且其相关矩阵定义为：

$$E[w_1(n)w_1(k)^H] = \begin{cases} Q_1(n) & n = k \\ 0 & n \neq k \end{cases}$$

$A \in R^{n,n}$  —— 状态转移矩阵；

$B$  —— 输入控制矩阵；

(2) 传感器测量方程。

$$z(k) = Hx(k) + v(k) \quad (5-3)$$

式中  $z(k) \in R^n$  ——是  $k$  时刻传感器的测量向量;

$v(k) \in R^n$  ——观测噪声, 可建模为零均值的白噪声过程, 且其相关矩阵定义为:

$$E[v_1(n)v_1(k)^H] = \begin{cases} R_1(n) & \cdots n = k \\ 0 & \cdots n \neq k \end{cases}$$

$H$  ——观测矩阵;

### (3) 初始状态描述。

状态方程从初始状态  $x(t_0)$  开始传播, 对于所有真实系统的特定时刻而言, 这个初值总是一个具体的向量。然而, 由于事先可能并不知道这个具体值, 所以在建模时应把初始状态当成一个满足高斯分布的随机向量。故可用均值  $x_0$  和方差  $P_0$  表示  $x(t_0)$

$$\begin{cases} E\{x(t_0)\} = x_0 \\ E\{[x(t_0) - x_0][x(t_0) - x_0]^T\} = P_0 \end{cases}$$

$P_0$  是所有元素都是分布在对角线的正数的对称矩阵, 它给出了真实状态和估计状态之间的方差。对角线上的元素代表每个状态和真值的方差。

### (4) 卡尔曼滤波算法。

卡尔曼滤波算法利用反馈控制来实现对状态的估计, 它根据前一时刻的系统状态, 估计出当前的系统状态, 然后根据当前时刻的实际观测值作为反馈, 来修正估计的状态。因此, 卡尔曼滤波算法可分为两个步骤: 状态预测和测量修正。

状态更新方程及时地由当前系统状态和噪声方差估计出下一步的系统状态(先验估计); 而测量更新方程则负责反馈, 将新的测量信号加入已经在状态更新方程中得到的先验估计状态, 并最终得到系统状态的后验估计。状态更新方程也可以看成状态预计方程, 而测量更新方程则可当成状态修正方程。卡尔曼滤波算法递推过程见图 5.2 所示。

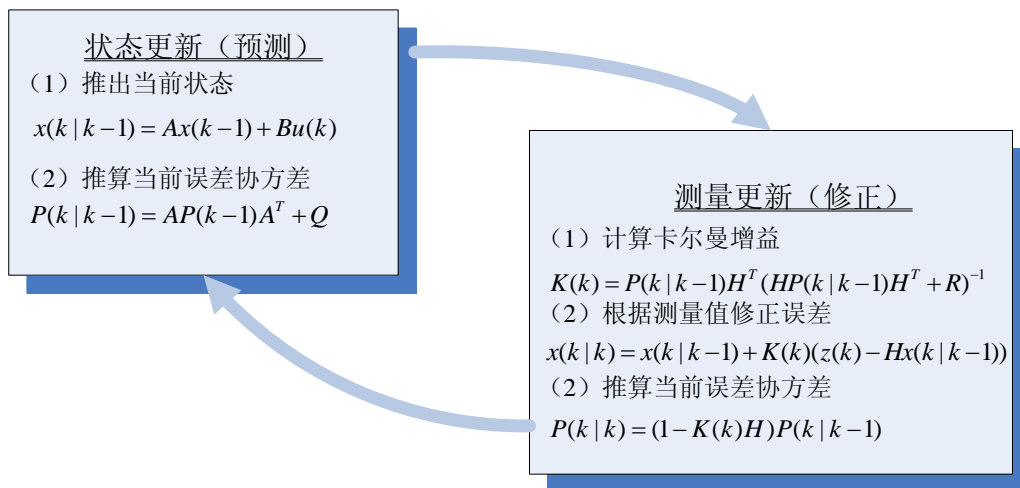


图 5.2 卡尔曼滤波“预测-修正”过程图

### 5.2.3 陀螺仪和加速度计的卡尔曼滤波

在本系统的设计过程中，状态向量的选取影响到整个状态方程的结构，是卡尔曼滤波器设计一个关键环节。考虑到倾角和倾角角速度存在导数关系，小车倾角真值  $\varphi$  可以用来做一个状态向量。然而倾角角速度却不适合作为状态向量，因为其导数无法给出。一个可行的选择就是不直接估计角速度真值，而采用加速度计估计出陀螺仪常值偏差  $b$ ，并以此偏差作为状态向量。相应的状态方程为：

$$\begin{aligned} \begin{pmatrix} \dot{\varphi} \\ \dot{b} \end{pmatrix} &= \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \varphi \\ b \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \omega_{gyro} + \begin{pmatrix} w_a \\ w_g \end{pmatrix} \\ \varphi_{acce} &= [1 \quad 0] \begin{pmatrix} \varphi \\ b \end{pmatrix} + v_a \end{aligned} \quad (5-4)$$

式中  $\omega_{gyro}$  ——包含固定偏差的陀螺仪输出角速度(作为卡尔曼滤波的输入信号)；

$\varphi_{acce}$  ——加速度计测量值

$w_a$  ——加速度计过程噪声

$w_g$  ——陀螺仪过程噪声

$v_a$  ——加速度计测量噪声

$b$  ——陀螺仪温漂误差

与相互独立，此处假设二者为满足正态分布的白噪声。令  $T$  为卡尔曼滤波器采样周期，则可得到离散系统的状态和测量方程为：

$$\begin{aligned} X(k) &= \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} X(k-1) + \begin{pmatrix} T \\ 0 \end{pmatrix} \omega_{gyro}(k-1) + \begin{pmatrix} w_a T \\ w_g T \end{pmatrix} \\ \varphi_{acce} &= [1 \quad 0] X(k) + v_a(k) \end{aligned} \quad (5-5)$$

要估算出  $k$  时刻的实际角度，就必须根据  $k-1$  时刻的角度值，再根据预测得到的  $k$  时刻的角度值得到  $k$  时刻的高斯噪声的方差，在此基础上卡尔曼滤波进行递归运算直至估算出最优的角度值。在此根据系统过程噪声协方差矩阵  $Q$  以及测量误差的协方差矩阵  $R$ ，对卡尔曼滤波器进行校正。

卡尔曼滤波器过程和测量的协方差矩阵的形式为：

$$Q = \begin{bmatrix} q_{-acce} & 0 \\ 0 & q_{-gyro} \end{bmatrix} \quad R = [r_{-acce}]$$

$q_{-acce}$  和  $q_{-gyro}$  分别为加速度计和陀螺仪测量的协方差，其不同的取值代表了卡

尔曼滤波器对不同传感器的信任程度。其值越小，表明信任程度越高。例如，如果更相信来自陀螺仪的数据，就可以通过设置比较小的  $q\_gyro$  值来将这样的信任体现到结果中去。 $R$  矩阵代表测量误差的协方差。如果  $R$  取值较大，那就意味着加速度测量误差有很大的噪声。 $Q$  和  $R$  矩阵的值通常可以用比较复杂的统计学方法计算获得。

融合卡尔曼滤波主要由以下 5 个方程来实现。

$$\begin{cases} x(k|k-1) = Ax(k-1) + Bu(k) \\ P(k|k-1) = AP(k-1)A^T + Q \\ K = P(k|k-1)[P(k|k-1) + R]^{-1} \\ x(k|k) = x(k|k-1) + K[z(k) - Hx(k|k-1)] \\ P(k|k) = (I - KH)P(k|k-1) \end{cases} \quad (5-6)$$

本设计经过大量实验后，最终采用  $q\_acce=0.001$ ， $q\_gyro=0.003$ ， $r\_acce=0.5$ ，

卡尔曼滤波器采用周期为 5ms， $P_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ，通过方程便可得到经过卡尔曼滤波之后的角度和角速度。经过卡尔曼滤波之前和之后的角度和角速度曲线图见图 5.3。

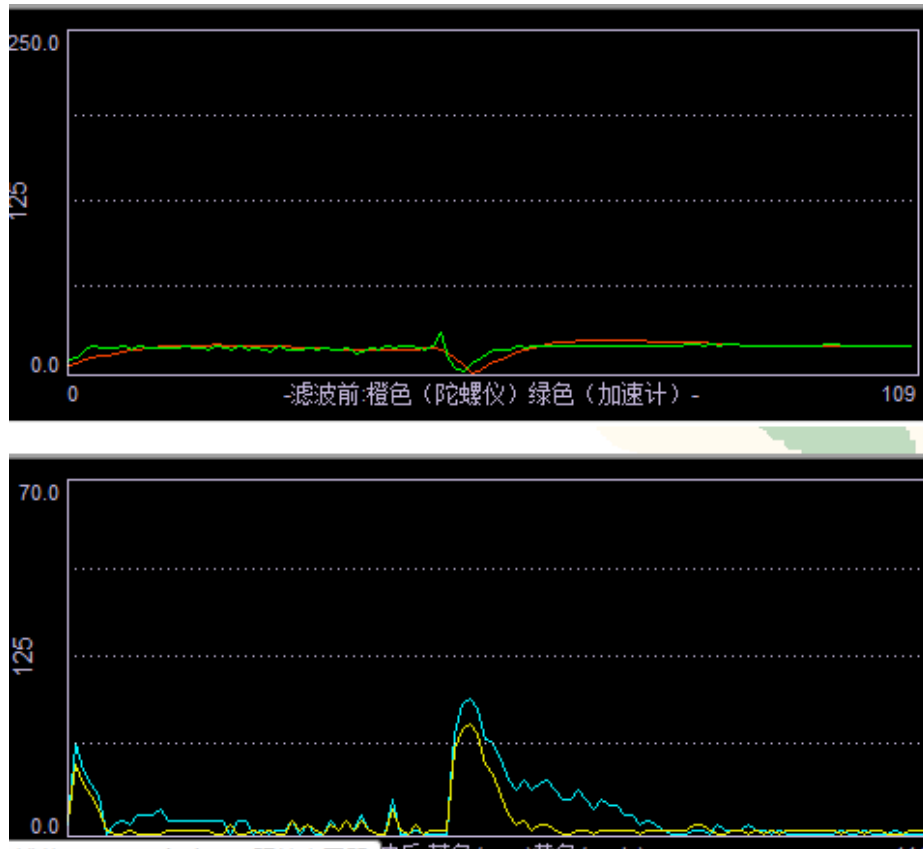


图 5.3 卡尔曼滤波前后效果对比图

### 5.3 图像处理及黑线提取算法设计

### 5.3.1 视频信号的采集

每场图像中有 320 个行信号，其中的第 23 行到第 310 行是视频信号，即每场中有效的视频信号有 288 行。第 311 行到下一场第 22 行是场消隐信号。行消隐区即为换行的标志。每幅又分奇、偶两场，先奇场后偶场。

该摄像头扫描的每场中有 320 行信号，其中第 23 行到 310 行是视频信号，即每场中有效的视频信号有 288 行。第 311 行到下一场的第 22 行是场消隐信号。在视频信号区，每行信号持续的时间相同，约为 62us；每行的行同步脉冲持续时间也相同，约为 4.7us。而在场消隐区，每行持续的时间会有所变化，每行对应的消隐脉冲持续时间，尽管大多数为 3.5us，但也有变化。场消隐区中第 320 行的消隐脉冲持续时间远长于其他消隐脉冲，此脉冲即为场同步脉冲。

通过数字摄像头捕获视频信号，省去了视频场行同步信号分离和 A/D 转换，可以采用查询方式或者中断方式对视频信号进行采样。主程序采用 I/O 查询方式判别奇偶场信号跳变，并用中断方式处理行同步引发的中断。当奇偶场信号发生跳变（即新的一场到来）时，对行同步信号计数器清零。

图像采集服务程序的流程为图 5.4 所示。

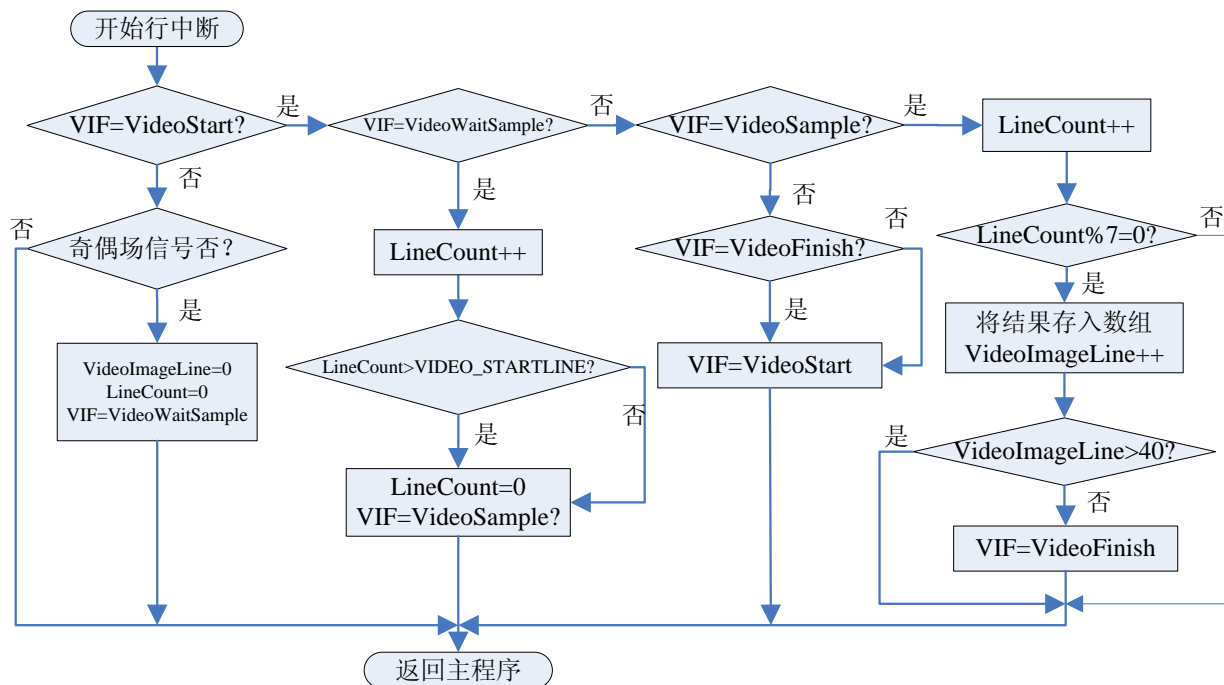


图 5.4 图像采集综合算法流程图

通过上位机与 CodeWarrior 编辑器进行数据传送将图像二维数组的数据传至上位机进行显示。在直道和弯道上采集到的图像信息如图 5.5 所示。



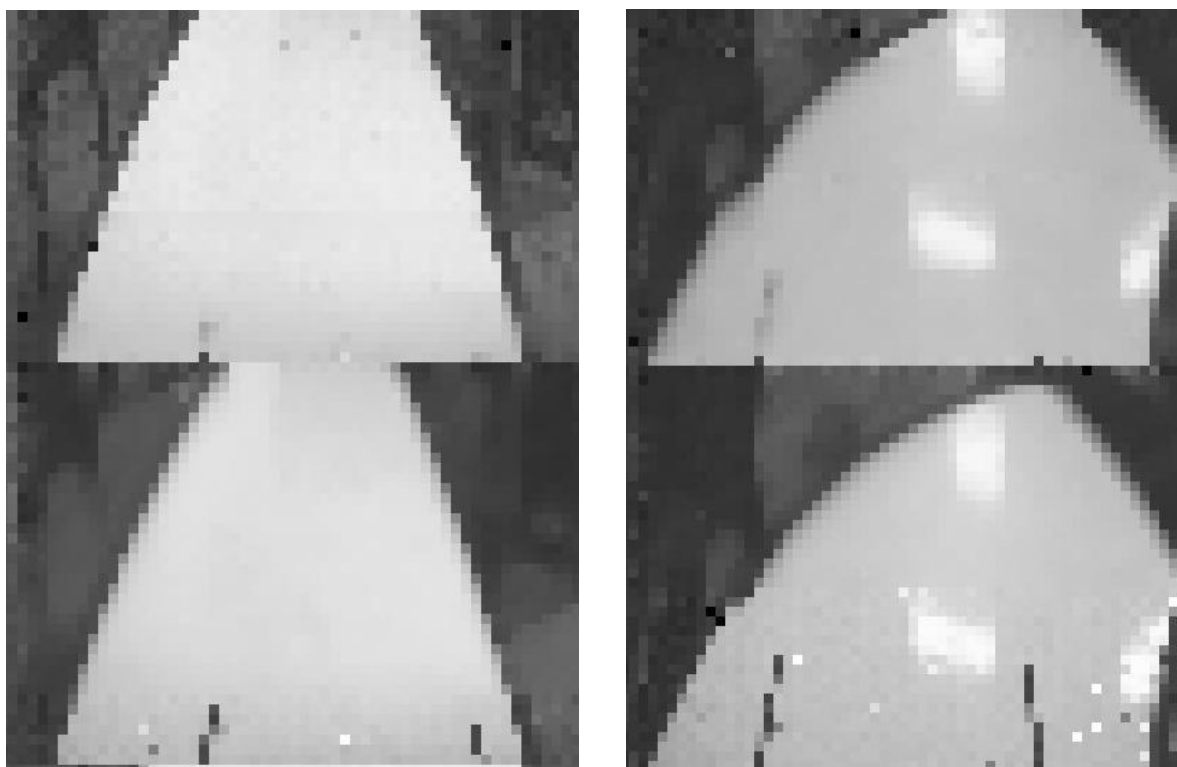


图 5.5 直道和弯道图像信息

由图 5.5 可以看出，直接采回来的图像仍有很多噪点，并且受外界光线的影响很严重。所以需要对图像进行滤波去噪。

### 5.3.2 图像的平滑滤波去噪

图像的平滑是一种实用的数字图像处理技术，主要的目的是为了减少图像的噪声。一般情况下，在空间域内可以用邻域平均来减少噪声；在频率域，由于噪声频谱通常多在高频段，依次可以采用各种形式的低通滤波的办法来减少噪声。

图像平滑滤波处理主要采用两类方法：空间域处理法和频域处理法。但由于频域处理法需要先把图像从空域变换到频域，在频率域处理后再变换回空间域，附加了图像正交变换的正变换和反变换，对于数据量较大的二维道路图像需要较大的内存，且运算时间也较长，不能满足视觉系统实时处理的要求。因此，本文采用空间域图像滤波处理方法。中值滤波是一种最为常用的图像滤波方法，它对异常值(与周围像素灰度值差别较大的像素的值)的敏感性比均值滤波的小，中值滤波可以在不减小图像对比度的情况下剔除这些异常值，使图像产生较少的模糊。因为它能够有效地去除尖峰噪声，还能对边缘起到很好的保护作用。在本设计中采用中值滤波对图像进行平滑去噪，窗口设为十字形，长度为 5。

依次读取窗口下各对应像素的灰度值，将这些灰度值从小到大排成一列，找出这些值里排在中间的一个，将这个中间值赋给对应模板中心位置的像素即可。

### 5.3.3 图像的二值化

图像的二值化可用硬件二值化或软件二值化实现。硬件二值化是采用一个比较器，将摄像头的的数据信号与参考电压进行比较，如果大于参考电压，则输出为 1，如果小于参考电压则输出为 0，从而得到图像的一系列二进制信号。软件二值化是确定一个阈值，大于该阈值的像素为白，小于该阈值的像素为黑。

二值化的关键是黑白阈值的确定。阈值的确定主要分为全局阈值法和局部阈值法。具体有平均灰度值法，跳变沿差分法，大津法，灰度直方图法等。

平均灰度值法：采样后先遍历一次图像，得到灰度值的平均值，用该灰度值乘以一个恰当的系数最为阈值。

边缘检测阈值法：找到像素的跳变沿，对跳变沿的前后两个像素灰度取平均值，使用一整行差分的平均值作为本行的阈值。实验证明，这种动态阈值能较好地适应各种光线情况，特别能适应光线不均匀的情况。

大津法（Otsu 法）：原理是取从 0 到最大灰度级之间的任一灰度值把图像分割成两类，分别计算这两类的像素点数和灰度平均值，然后计算它们的类间方差。当被分割的两类类间方差最大时，此灰度值就作为图像二值化处理的阈值。

直方图波谷阈值法：一幅物体与背景对比明显的图像一般具有包含双峰的灰度直方图。物体与背景的边界附近具有两个峰值之间的灰度级，其像素数目相对较少，从而产生了两峰之间的波谷。选择直方图的波谷作为灰度阈值将得到合适的二值化结果。但一般情况下，图像都会含有噪声，使得直方图的波峰和波谷的位置难以确定。

双峰直方图阈值法：适合目标与背景有较强对比的图像。由于一般的图像对应的直方图一定是双峰直方图，把阈值设定在相对于两峰的某个固定位置，这个结果将比直接寻找直方图波谷的位置更为可靠。其算法如下：

- (1) 计算像素灰度平均值，标准差。
- (2) 以像素平均值为分界点，分别求出左右部分的最大值的位置。
- (3) 若两峰值位置相距较近（说明直方图的双峰中有一个峰很低，需另寻波峰的位置，否则转至第 7 步）。
- (4) 求出像素灰度中值点位置。
- (5) 如果中值点位置大于灰度平均值，表明小峰在大峰左边，否则小峰在大峰右边。相应调整分界点的位置。
- (6) 重新求出大，小峰值的位置。
- (7) 以两峰值位置的中点作为所要求的阈值。

由于我们的道路环境不是很复杂，且单片机的运算能力有限，在此选用双峰直方图阈值法对灰度图像进行黑白二值化。

#### 5.3.4 赛道中心线的提取

黑线的提取有二值化算法，直接边缘检测算法，双峰边缘跟踪算法。最常用的是直

接边缘检测算法和双峰边缘跟踪算法。

直接边缘提取算法：边缘提取算法是最简单的黑线处理算法，即从左边的第一个有效像素点(usful\_line)开始依次对相的像素点进行灰度值相减(左减右)，当两者的差值的绝对值大于阈值(VALUE)时，判定下一个像素点(usful\_line+1)对应的为黑线的左边缘。继续将两像素点相减当差值再次大于阈值时，将此像素定位黑线的右边缘。白色灰度远大于黑色的灰度。

双峰边缘跟踪算法：双峰边缘跟踪算法是寻找出目标指引线的上边缘，仍然用上边缘的位置代表目标指引线的位置。但是双峰边缘跟踪算法是寻找目标指引线的上下边缘，求临近3个像素点的差值以提取特征点。若只有目标指引线一条黑线时，该算法能准确提取出目标指引线。该算法抗环境光强变化干扰的能力较强，同时能削弱或消除垂直交叉黑色指引线的干扰。

因为目标指引线是连续的，所以相邻两列的上边缘点比较接近。双峰边缘跟踪正是利用了这一特性，其主要思路是：当已寻找出某列的上边缘，若在该位置附近寻找下一列的上边缘，则只用花较少的步骤就可以找到。另外上下边缘同时做此步骤，提高了算法效率。这种方法的特点就是始终跟踪在每列上边缘的附近，去寻找下一列的上边缘，同时左右一起进行处理，所以就称这种方法为双峰边缘跟踪算法。实测该算法的运算时间平均为4.50ms。该算法能准确提取出目标指引线，而且具有较强的抗干扰性。该算法同边缘检测方法一样具有较强的抗光强变化的能力，而且该算法能更有效地削弱甚至消除垂直交叉黑色指引线的干扰，该算法还能排除画面内干扰图迹的影响，始终跟踪住目标指引线。

黑线提取算法的基本思想如下：

- (1) 直接逐行扫描原始图像，根据设定的阈值提取黑白跳变点；
- (2) 黑线宽度有一个范围，在确定的黑线宽度范围内提取有效黑块，这样可以滤除不在宽度范围内的黑点干扰；
- (3) 利用黑线的连续性，根据上一行黑线中心的位置来确定本行求出的黑线中心是否有效；
- (4) 图像是远处小近处大，所以黑线宽度范围和前后行黑线中心的位置差别都要动态调整；
- (5) 求黑线中心时，因为近处的黑线稳定，远处黑线不稳定，所以采用由近及远的办法；
- (6) 图像数据量大，全部扫描一遍会浪费很多时间，利用前面已经求出的黑线中心位置判断出黑线的趋势，从而推断出下一行的黑线大概位置，确定出扫描范围，避免整行逐点扫描，节省时间；
- (7) 提取出整场所有有效行的黑线中心后，根据预先设定好的权重计算出黑线中心的加权平均值，作为本场的黑线中心。

黑线提取算法的流程图如图 5.6 所示。

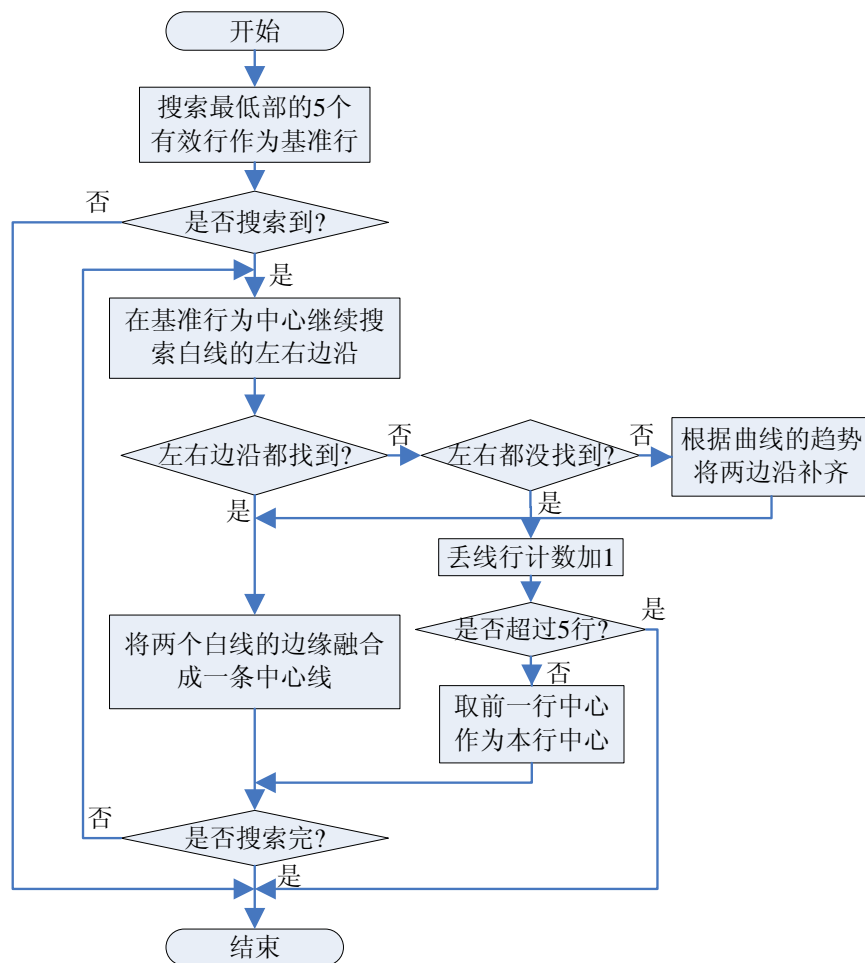


图 5.6 黑线提取算法流程图

## 5.4 系统控制算法设计

### 5.4.1 PID 控制算法简介

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 5.7 所示。

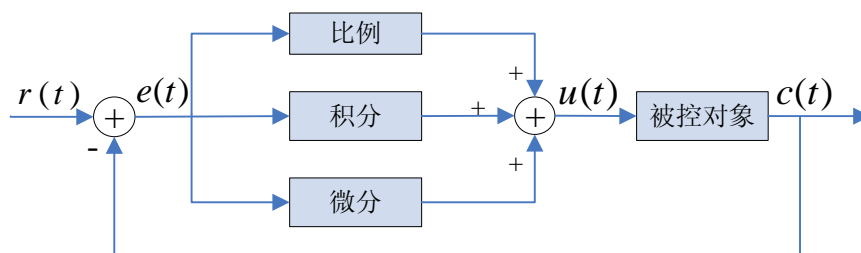


图 5.7 PID 控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

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

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_I} \sum_{j=0}^k e(j) + \frac{T_D}{T} [e(k) - e(k-1)] \right\} \quad (5-7)$$

#### (1) 增量式 PID 控制器

增量式的输出量只于上一次的相关，只需计算增量，当存在计算误差或精度不足时，对控制量计算的影响较小。

$$u(k) = u(k-1) + \Delta u$$

$$\Delta u = K_p \times [e(k) - e(k-1)] + K_i \times e(k) + K_d \times [e(k) - 2e(k-1) + e(k-2)] \quad (5-8)$$

增量式 PID 通常多用于速度。如将它用于直流电机控制，温度的控制。

#### (2) 位置式 PID 控制器

$$u(k) = K_p \times e(k) + K_i \times \sum_{i=0}^k e(i) + K_d \times [e(k) - e(k-1)] + u_0 \quad (5-9)$$

位置式 PID 算法每次输出与整个过去状态有关，计算式中要用到过去偏差的累加值，容易产生较大的积累误差。通常多用于跟随系统，控制量一般多跟角度，位置相关，如将它用于舵机的控制，步进电机的控制或平衡的控制。 $r(k)$  一般为 0。

### 5.4.2 平衡控制算法设计

#### (1) 直立控制：

直立控制首先需要根据加速度计和陀螺仪的值得到车体的倾角和角速度值。可知加速度计的量程为 1.5g，800mv。陀螺仪的灵敏度为 0.67mv/deg/s。由于该陀螺仪和加速度计均为模拟器件，故先将 AD 读到的电压值先减去零偏置电压，再通过一定的关系分别转换成角度和角速度。将该角度和角速度的值送入卡尔曼滤波器得到准确的角度和角速度。对经过卡尔曼滤波滞后的角度和角速度进行 PD 控制可使小车基本实现直立。只是由于系统相当于是开环控制，无法静止在某一点。系统抗击外界的干扰能力不强。

平衡结构框图如图 5.8 所示。

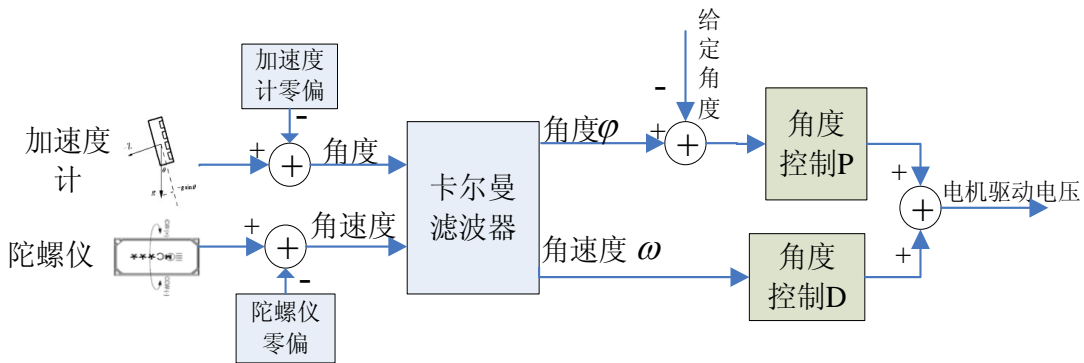


图 5.8 平衡控制的结构框图

角度控制 **P** 参数的作用：角度 **P** 参数相当于倒立摆的回复力，能克服重力的作用使小车保持直立，**P** 参数一般要比较大，至少大于重力加速度的等效数值。**P** 越大可以增加车模维持直立的倾向。但过大会引起小车的震荡。所以单靠比例控制难以使系统维持稳定。

角度控制 **D** 参数的作用：角度 **D** 参数相当于倒立摆的阻尼力，可以消除小车的震荡，克服震荡的自由能量的消失，但过大会引起小车的高频抖动。一般相对 **P** 参数来说都较小。

系统的平衡示意图如图 5.9 所示：

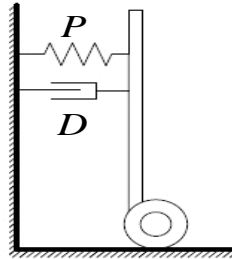


图 5.9 系统平衡控制示意图

## (2) 速度控制

通过光电编码器测量得到左右轮在测量时间  $\Delta t$  内的行进距离  $\Delta s_R$  和  $\Delta s_L$ ，相应地得到右轮速度  $v_R$  和左轮速度  $v_L$ ：

$$v_R = \frac{\Delta s_R}{\Delta t}, \quad v_L = \frac{\Delta s_L}{\Delta t}$$

得到车体的线速度为：  $v = \frac{v_L + v_R}{2}$ 。

首先需要对编码器的测量值进行地低通滤波。

```
LeftSpeed_filter*=0.8;           //车轮速度低通滤波 0.7304
LeftSpeed_filter+=LeftRealspeed*0.2; //0.2696
```

```
RightSpeed_filter*=0.8;          //车轮速度低通滤波 0.7304
RightSpeed_filter+=RightRealspeed*0.2; //0.2696
```

可知小车的速度最终由倾角所产生的加速度决定，速度通过 **PI** 控制器进行闭环控制。因此将倾角可知闭环与速度控制闭环叠加在一起加载到电机上便可实现对小车的运动速度的控制。此时小车速度控制框图如图 5.10 所示。

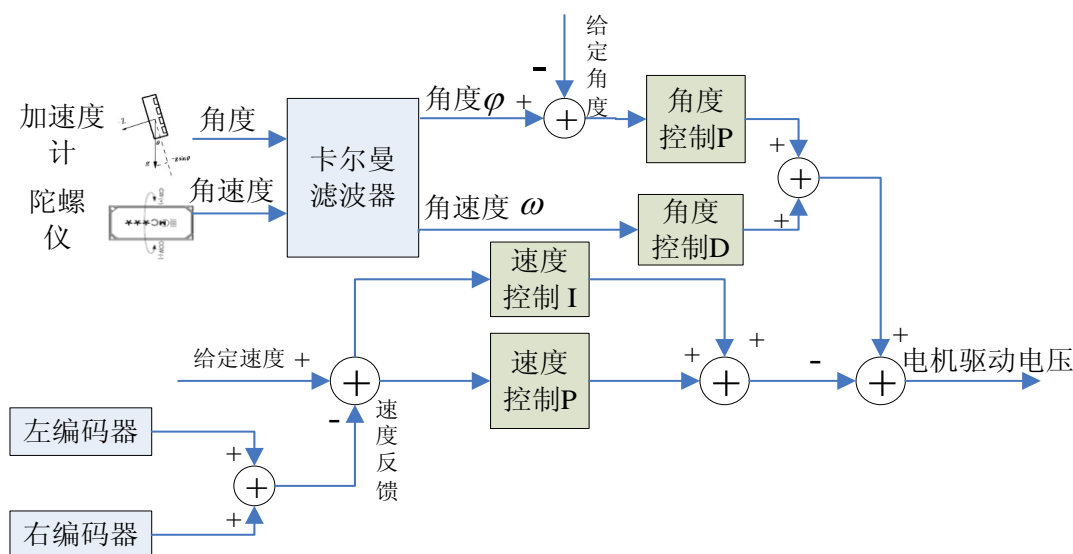


图 5.10 速度综合控制控制框图

速度控制P参数的作用：通过P参数可以消除来回摆动，从而使其静止在某一点，提高P参数可以提高系统抗外部干扰的能力。但P过大，会削弱角度P参数的作用，从而使角度产生震荡。

速度控制I参数的作用：通过速度和对速度进行积分，通过积分的误差来抵消它的运动倾向，阻碍其向一个方向，I参数越大，停止在某一点的速度越快，但I过大会在某一点来回摆动。

### 5.4.3 转向控制算法设计

#### (1) 车身转向角速度的获取

根据编码器获得车身转向角速度：
$$\omega = \frac{v_R - v_L}{L}$$

其中： $v_L$ ， $v_R$  分别为车子左右两个轮子的线速度， $L$  为左右两轮之间的距离

#### (2) 车身偏移道路中心的控制量

根据图像获得车子偏移道路中心的控制量，通过编码器得到车身转向角速度。对偏差控制量和转向角速度进行PD控制便可实现转向和自主寻迹。控制框图如图 5.11 所示。

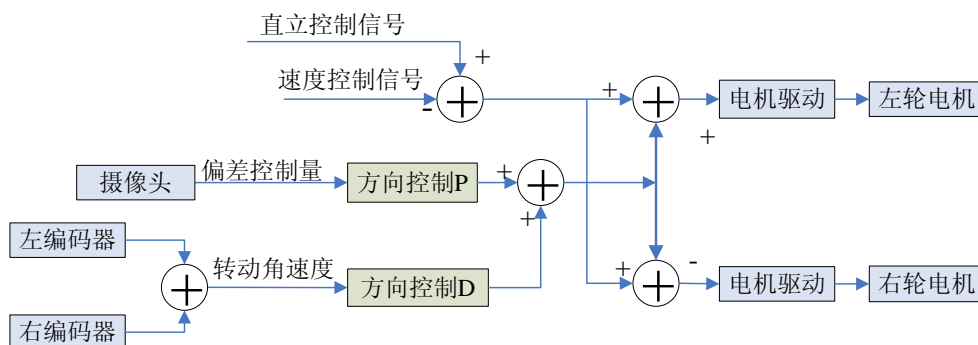


图 5.11 系统转向控制框图

方向控制  $P$  参数的作用：迅速减小偏差，使系统快速跟随给定，决定于系统的响应速度和快速性， $P$  参数太小，系统的控制作用太弱，不利于克服外界扰动，系统反应迟钝，延长调节时间长，转弯不及时。增加  $P$  参数可以尽快消除小车与中心线的偏差，但  $P$  过大系统容易产生超调。 $P$  与偏移量近似呈抛物线关系。

方向控制  $D$  参数的作用：微分控制能够按照偏差的趋势进行控制，提高系统的动态性能，适用于大滞后系统。能在偏差变化太大之前，在系统中引入一个有效的早期修正信号，起提前纠正的作用，且偏差变化越快，它的控制量越大，因而既可以加快系统响应速度又可以克服震荡，使系统更加稳定。能起预测前瞻和迅速调整车身方向的作用，从而加快响应速度，消除振荡，提高系统的稳定性，但  $D$  过大会导致系统频繁抖动，且对噪声很敏感。

将直立控制，速度控制，转向控制综合到一起便得到系统的综合控制框图，如图 5.12 所示。

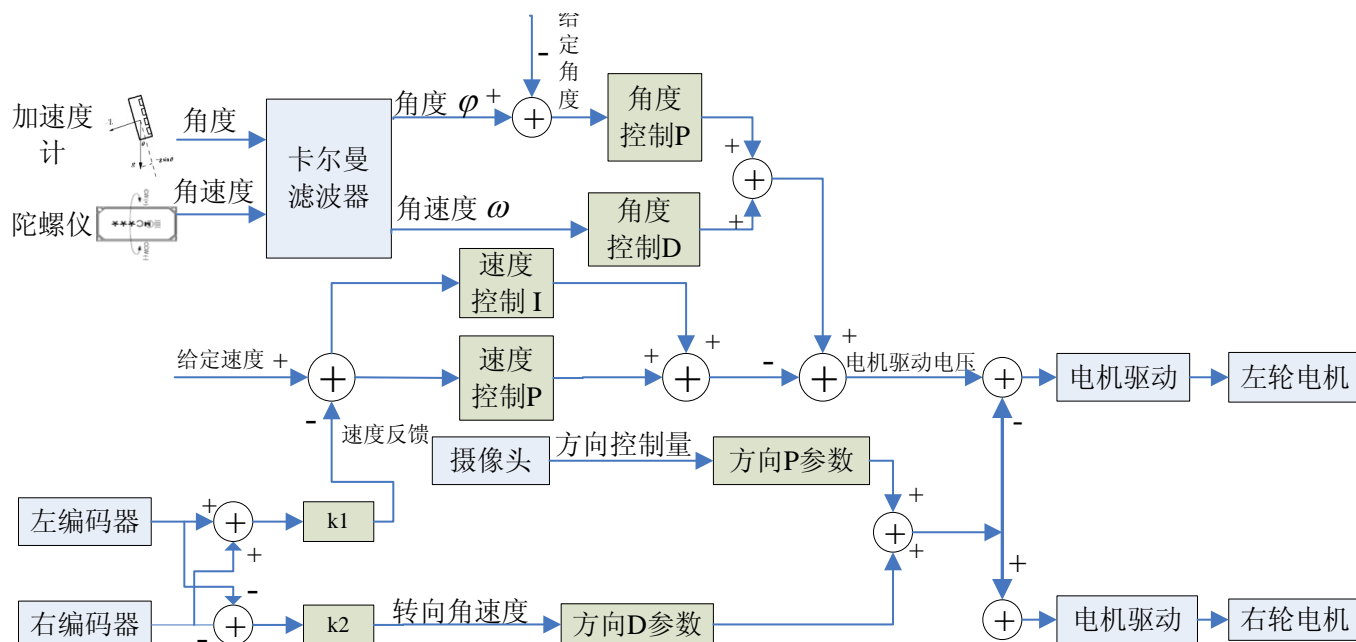


图 5.12 系统综合控制算法框图

## 5.5 系统的定位算法设计

### 5.5.1 编码器与加速度计的自适应加权数据融合

编码器的可以获得系统实时的速度信息，但是往往因轮子打滑等会产生一些系统误差。加速度计通过一定时间的积累可以获取系统的速度信息，但是会产生累积误差。在此将两线速度进行自适应加权数据融合，以弥补编码器的误差。

假定两个不同的传感器同时对某一变量  $x$  进行测量，其观测值分别为  $z_i$ ，观测时存在随即观测误差  $v_i$ ， $v \sim N(0, \sigma^2)$ ，观测相互独立，预设计一种处理方法，以得到  $x$  的



无偏估计值。有  $z_1 = x + v_1$ ,  $z_2 = x + v_2$ 。

假定  $x$  的估计值与测量值  $z_1, z_2$  呈线性关系, 则有  $\hat{x} = \omega_1 z_1 + \omega_2 z_2$ , 其中  $\omega_1, \omega_2$  为各传感器的权值。设估计误差为:  $\hat{x} = x - \hat{x}$ , 取的均方差最小为最优依据。根据估计值为无偏估计, 有  $E(\hat{x}) = E(x - \hat{x}) = E[x - \omega_1(x + v_1) - \omega_2(x + v_2)]$ , 又  $\omega_2 = 1 - \omega_1$ , 则:

$$\begin{aligned} E(\hat{x}^2) &= E\{[x - \omega_1(x + v_1) - \omega_2(x + v_2)]^2\} \\ &= E\{[\omega_1 v_1 + (1 - \omega_1)v_2]^2\} \\ &= E[\omega_1^2 v_1^2 + (1 - \omega_1)^2 v_2^2 + 2\omega_1(1 - \omega_1)v_1 v_2] \end{aligned}$$

又  $\because E(v_1^2) = \sigma_1^2, E(v_2^2) = \sigma_2^2$ , 且  $v_1, v_2$  相互独立, 所以  $E(v_1 v_2) = 0$

$\therefore E(\hat{x}^2) = \omega_1^2 \sigma_1^2 + (1 - \omega_1)^2 \sigma_2^2$ , 为使得  $E(\hat{x}^2)$  最小

$$\frac{\partial E(\hat{x}^2)}{\partial \omega_1} = 0 \Rightarrow \begin{cases} \omega_1 = 1 / \sigma_1^2 \left( \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) \\ \omega_2 = 1 / \sigma_2^2 \left( \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) \end{cases}$$

$$\text{最优估计量 } \hat{x} = \omega_1 z_1 + \omega_2 z_2 = z_1 / \sigma_1^2 \left( \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) + z_2 / \sigma_2^2 \left( \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right)$$

上式表明当两个传感器权值取合适值时, 可以通过两观测器已经获得的观测值融合得到最优的估计值  $\hat{x}$ 。可以把该结论推广到多个传感器的情况, 设多传感器组的方差分别为  $\sigma_1^2, \sigma_2^2, \sigma_3^2, \dots, \sigma_n^2$ , 各传感器的测量值分别为  $z_1, z_2, \dots, z_n$ , 彼此互相独立。真值的估计值为  $\hat{x}$ , 并且是  $x$  的无偏估计, 各传感器的加权因子分别为  $\omega_1, \omega_2, \omega_3, \dots, \omega_n$ , 根据多元函数求极值理论, 可求出总均方误差最小时所对应的加权因子为

$$\omega_k = 1 / \left| \sigma_k^2 \sum_{i=1}^n \frac{1}{\sigma_i^2} \right|。$$

最优加权因子决定于各个传感器的方差, 初始方差可根据各个传感器所提供的测量值, 依据相应的算法求出。

将编码器测得的线速度与加速度计累加得到的线速度进行自适应加权数据融合以得到准确的线速度。

### 5.5.2 编码器和陀螺仪的卡尔曼滤波器融合

以编码器和陀螺仪作为机器人的定位系统, 可建立机器人的运动学方程。编码器可

以得出系统的姿态角，陀螺仪可以得到系统当前的转动角速度。采用卡尔曼滤波器对两种传感器的数据进行融合，以减小编码器和陀螺仪的误差，获得机器人的位姿信息。

设状态变量为： $x = [\theta \ \omega]^T$ ，系统输入为 0

则：系统的状态方程为： $x(k) = Ax(k-1) + Bu(k-1) + w(k-1)$

测量方程为： $z(k) = Hx(k) + v(k)$

$w(k)$  是协方差为  $Q$  的系统过程噪声矩阵， $v(k)$  是协方差为  $R$  的测量噪声矩阵

根据式(5-6)中的方程组可得到系统的转动角速度  $\omega$  和单位周期内转过的角度  $\theta$ 。

将系统线速度  $v$ ，转动角速度  $\omega$  及一个采样周期内车子转过的角度  $\theta$  带入式（3-4）即可得到系统实时的位置和位姿，给出一个初始状态信息，便可实现系统的实时定位。

## 5.6 系统的误差分析与修正

### （1）里程计模型航位推算的误差

里程计是基于安装在驱动车轮上的编码器将车轮旋转转化为相对地面的线性位移这一前提，因而具有一定的局限性。其误差来源分为系统误差(systematic Errors)和非系统误差(Non-Systematic Errors)：系统误差包括左右驱动车轮半径的差异、车轮半径平均值与标称值的差异、车轮安装位置的差异、有效轮间距的不确定性、有限的编码器精度和采样速率，它对里程计误差的积累是恒定的；而非系统误差则包括运行地面的不平整、运行中经过意外物体、以及多种原因造成的车轮打滑，它对里程计误差的影响是随机变化的。

可以从三个方面对里程计模型进行修正：两轮之间的距离测量误差系数，左编码器的修正系数，右编码器的修正系数。

### （2）系统定位的误差

文中所采用的定位方法只是相对定位方法，随着时间的增加就会有误差的累积，因而可将激光超声波和电子罗盘等传感器加入机器人定位系统实现系统的绝对定位，采用相对定位和绝对定位相结合的方式消除经长时间运行而产生的累积误差。

## 6 系统的安装与调试

### 6.1 系统机械的安装

#### 6.1.1 测速编码器的安装

测速编码器的安装要求不影响智能车的重心和支点。本设计采用两个编码器分别测得两个后轮的转速，编码器的联轴器直接轻压轮胎，如图 6.1 所示。

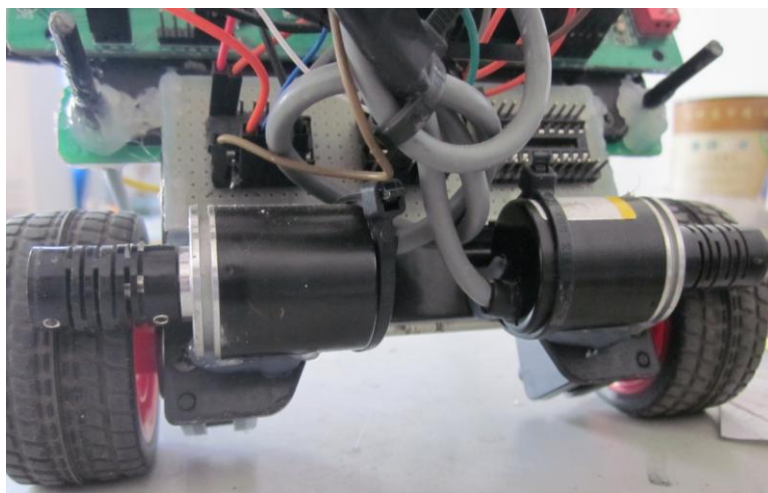


图 6.1 两测速编码器的安装图

#### 6.1.2 摄像头的安装

摄像头的安装位置应合适选取。安装位置太低，会导致视域不够广阔，影响寻线的有效范围；安装位置太高，导引线会变得过窄而无法被检测到。安装位置合适的一个标准是：在此位置的拍摄范围能满足控制的需要。如果控制的策略简单，则所需的拍摄范围就可以小一些；反之策略复杂，需获得的赛道信息较多，则拍摄范围就应大一些。该系统中摄像头的安装如图 6.2 所示。



图 6.2 摄像头安装示意图

#### 6.1.3 倾角传感器的安装

小车倾角传感器包括陀螺仪和加速度计。最好将带有陀螺仪和加速度计的电路板固

定在整个车子中间质心的位置，或者车子的底部，这样可以最大程度减少车子运行时前后振动对于测量倾角的干扰。安装角度传感器电路板时应该尽量保证陀螺仪传感器水平安装。该系统中倾角传感器的安装如图6.3所示。

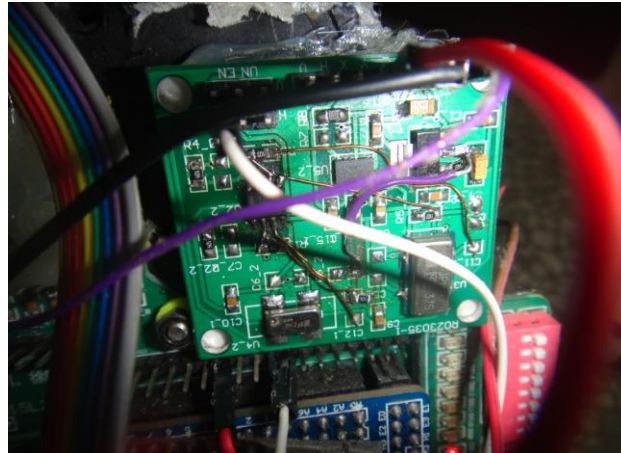


图 6.3 倾角传感器的安装

如果陀螺仪的安装不是水平的，那么这个转动就会在陀螺仪的 Z 轴方向存在一个分量。会影响小车过弯道时的速度，表现为过弯道时速度变快或变慢。根据陀螺仪倾斜的方向不同，这个分量有可能是正，有可能是负。从而会使得小车的控制“仿佛感觉到在上坡或者是在下坡”，引起车模的速度变慢或者变快，其示意图如图 6.4 所示。

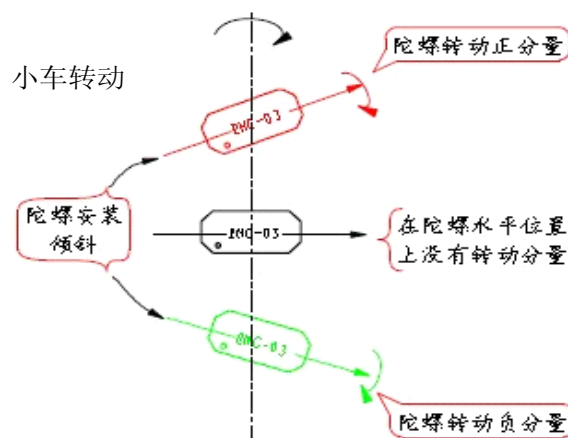


图 6.4 陀螺仪安装倾角对车子运行的影响

## 6.2 系统的调试

### 6.2.1 开发平台

MC9S12XS128单片机的开发主要使用Netrowerks公司Codewarrior软件进行了编写和调试。我们使用的是Codewarrior5.3版本，全新版本完美支持了如Windows Vista等系统，使用开发更加人性化。CodeWarrior是专门面向Freescale所有MCU与DSP嵌入式应用开发的软件工具，它包括构建平台和应用所必需的所有主要工具——IDE、编译器、调试器、编辑器、链接器和汇编程序等。一方面，CodeWarrior具有友好的界面，可以即时

改变目标单片机型号和开发调试场景，开发人员可以在线实时调试，实时修改寄存器的参数并观测变量的波形。另一方面，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，后编译并链接程序以便计算机能够理解并执行你的程序。CodeWarrior的界面如图6.5所示。

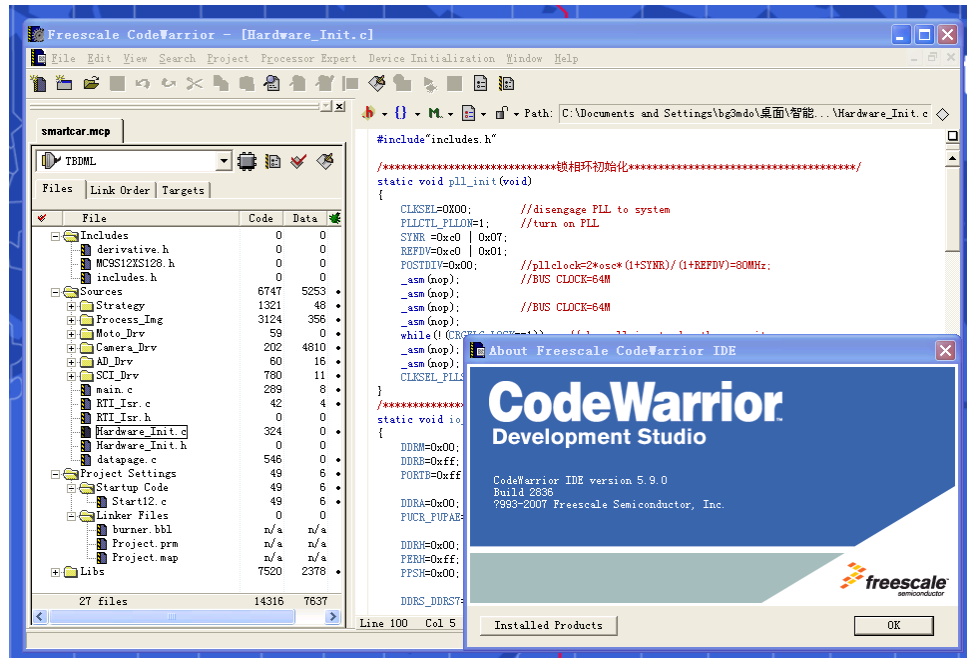


图 6.5 CodeWarriorIDE5.3 界面

CodeWarriorIDE集成开发环境中附带了强大的仿真和烧录工具Hiwave6.1工具配和BDM能实现程序的实时调试，观察变量，跟踪调试下载等功能。在对程序算法进行转译C语言的过程中可以调用全片仿真，了解程序实际工作状态，

## 6.2.2 无线串口调试

### (1) 串口调试工具。

本设计采用一对 XL105-232AP2 模块，如图 6.6 所示，一块连接 USB 转的串口与 PC 相连，另一块直接用于单片的串口。无线串口模块主要功能就是替代有线的串口通讯，使用简单，有效传输距离可达 15 米，可直接接单片机或 PC 串口。



图 6.6 无线串口 XL105-232AP2 模块



## (2) 无线串口调试上位机。

在调试的过程中,使用自己编写的上位机软件监控小车的运行参数。上位机软件用MFC编写,它既具有运行效率高、实时性好的优点,又结合小车的硬件设计的特点,符合系统的监控要求。实验表明上位机软件运行稳定、可靠。上位机软件图形界面如图6.7。

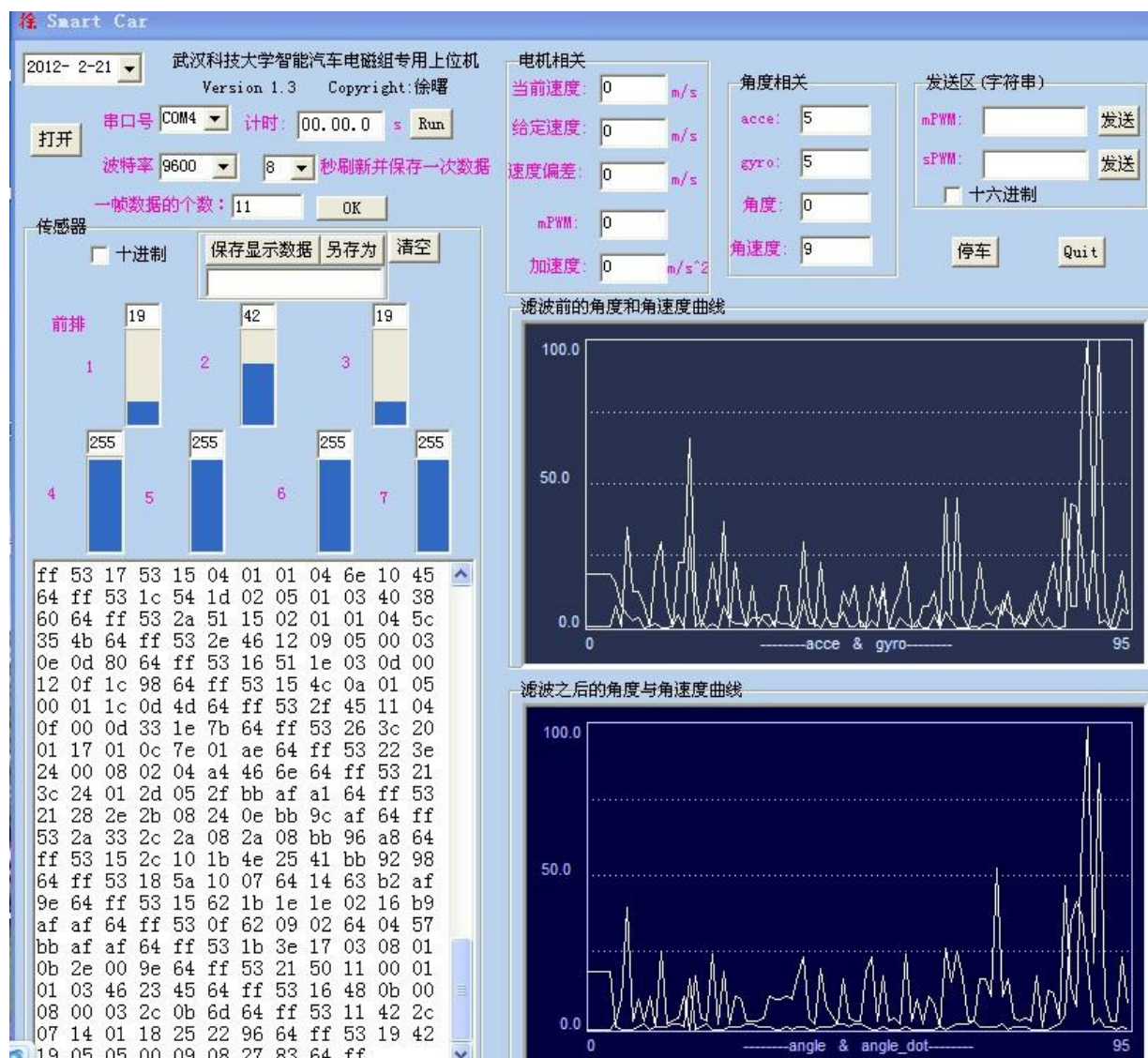


图 6.7 基于 MFC 的串口调试上位机

## 6.2.3 系统的调试与参数整定

小车拼装好后,就进入调试阶段。整车调试分为硬件调试,静态调试和动态运动调试三部分。

硬件调试过程中,先将电路加上 7.2V 电池电压之后,测量电路的工作电流和各个稳压电路的输出值是否在安全范围之内。主要测试软件信号是否正确,电路中的供电电路、PWM 输出、速度控制电路、编码器采样、IMU 传感器的值等是否正常。

静态参数调试:包括车模传感器参数包括了测量车模运动状态各传感器在车模静止时的零点偏移量以及单位换算的比例值

### (1) 角度参数的整定

角度控制参数包括比例和微分两个参数。比例P参数相当于倒立摆的回复力。这个参数需要大于重力加速度所产生的效果才能够使得小车保持直立。随着这个参数逐步增大，小车开始能够保持直立。该参数进一步加大，小车会开始出现来回摆动的现象。微分D参数相当于倒立摆中的阻尼力，它可以有效的抑制小车的摆动。当该参数过大时，会引起小车本身的震动。这是由于小车本身不是一个刚体，车体具有一定的共振频率。微分参数过大时会使得小车在电机的驱动下产生车体的共振。

调节这两个参数可以遵循先比例后微分的过程。首先调整比例P参数，使得小车能够保持直立并且开始来回摆动了。说明此时比例P参数已经可以克服重力的影响。然后逐步增加微分参数，小车逐步保持直立稳定。增大微分D参数直到小车开始共振，这样就可以确定微分参数的最大值了。然后适当减小微分D参数，然后逐步增加比例参数，直到小车又开始震荡，这样便可以确定比例参数的最大值。然后适当减小比例参数。在这些参数附近进行试验，直到寻找到一组小车直立的控制最优参数。

### (2) 速度参数整定

在角度参数初步确定之后，便可以进行速度控制参数的整定。速度控制参数包括比例P参数和积分I参数两个控制参数。由于是进行静态调整，故此先将速度控制闭环的速度设定值置为0。然后开始后面的比例和微分参数的调整。具体的调试过程可以首先逐步增加比例P参数，小车能够在一定平衡点附近来回运动。然后在逐步增加微分参数，车模就会很快停留在平衡点出。使用外力冲击小车，小车能够很快趋于静止。然后再逐步增加比例和积分控制参数，使得小车抵抗外部干扰冲击的能力逐步增强。

首先请注意速度的调制本质上是通过小车车模的角度来实现的。图 2-38 中显示了速度控制环节。比例控制 P 参数决定抑制速度误差的能力，但是只有比例控制，D 参数速度控制就会出现双积分负反馈控制。第一个积分是误差的积分，第二个积分是小车倾角所产生的加速度积分。因此小车速度就会出现震荡现象。通过微分控制可以消除这个震荡现象。而且由于小车的速度的控制对角度的控制来说是干扰，如果速度控制的参数调整的过大，就会削弱小车角度的控制作用。

### (3) 方向参数整定

方向控制包括两个参数：比例控制 P 参数和微分控制 D 参数。在调整方向参数时，比例控制 P 参数可以使得小车的方向恢复至正确位置。当该参数逐步增加时，小车的方向回复速度逐步加快。当比例 P 参数增加到一定数值之后，小车的方向回复由于过快，会出现方向过冲现象。此时通过增加微分控制 D 参数可以抑制这种方向过冲的现象。D 参数可以使小车在直道入弯道时迅速转弯，弯道进直道时迅速将车身摆正，并且在蛇形弯处能迅速调整车身的方向。通过几次组合测试，可以选择一组合适的比例控制和微分控制参数使得 D 参数方向控制即迅速又不会出现过冲现象。

由于小车方向和速度的控制对角度的控制来说都是干扰，因此，如果方向控制的参

数调整的过大，就会影响小车角度的控制。因此在此调试过程中，需要兼顾小车的方向和直立控制两方面的稳定性。

以上三个步骤调整了小车的角度、速度和方向控制。这些参数是在车模静止的时候进行调试的。它们之间具有一定的相互影响。在后面小车运动过程中根据前面调整所遵循的原则和它们之间相互影响的体验，逐步优化这些参数，使得小车在运动过程中保持稳定、快速的特性。

基本的静态测试完成后，就可以在赛道上进行动态测试了。动态调整一般是在静态调整所得到的各个参数基础之上，逐步提高小车速度的设定值，观察小车在不同的运行速度设定值下，它的角度、速度以及方向控制的性能。在保证小车的角度和方向稳定的条件下，逐步提高小车的速度，观察出现的各种现象，根据前面调整控制参数所得到它们之间的相互影响的关系，通过综合调整各参数，使得小车能够兼顾稳定性和速度。



## 结束语

两轮自平衡小车是基于倒立摆模型的复杂非线性系统，是传统四轮小车的进一步升华。具有重大的理论和实际意义。随着社会的发展，两轮自平衡机器人的研究与应用必将得到越来越多的重视。

本文以两轮自平衡小车为研究对象，围绕该两轮小车研究并实现了两轮小车的自平衡控制，速度控制，转向控制及自主寻迹。研究了两轮驱动小车的定位模型机及多传感器融合定位算法。对倒立摆动力学模型、传感器数据融合、里程计模型、多输入单输出PID控制器等展开了研究。本设计的关键技术为：小车倾角和角速度的滤波，自平衡控制的PID参数的整定，图像动态二值化和图像分割，道路的识别及小车转向的控制，角度闭环和速度闭环的相互协调。

本文具有以下创新点。

(1) 将卡尔曼滤波运用于加速度计和陀螺仪的融合之中，使倾角的收敛速度和滤波效果平衡得很好。角度的滤波是本设计的重点和难点所在。

(2) 设计并实现了将摄像头运用于两轮自平衡小车之中。利用了摄像头传感器的天然优势。

同时注意到，两轮移动机器人的研究仍然没有成熟。还有很多难题需要解决。

(1) 系统结构的解耦，考虑增加辅助执行结构，减轻电机的负担，从而增强平衡系统的稳定性。

(2) 现有的两轮自平衡小车当其倒下时，自身是无法站立起来的，需要靠外力的辅助才能重新再站起来，自主性和功能在很大程度上被减弱。

(3) 现有的两轮自平衡小车还无法实现上下坡，靠目前的硬件系统和机械结构式无法实现的，必须添加辅助执行结构和传感器。因此机器人的起步站立和上下坡功能将成为两轮自平衡机器人的下一发展目标。

(4) 系统定位可采取绝对定位和相对定位相结合的方式，利用电子罗盘，激光雷达等传感器获取系统的绝对角度和信息。

(5) 机器人系统在三维空间的定位。

## 参考文献

- [1] 吴怀宇, 程磊, 章政. 大学生智能汽车设计基础与时间[M]. 北京: 电子工业出版社, 2008.
- [2] 第七届全国大学生“飞思卡尔”杯智能汽车竞赛电磁组直立行走参考设计方案, 飞思卡尔智能汽车竞赛秘书处, 2012.
- [3] 阮晓钢, 任红格. 两轮自平衡机器人动力学建模及其平衡控制(北京工业大学电子信息与控制工程学院, 北京 100124).
- [4] 孔祥宣, 曹其新. 自主式双轮动态移动机器人的控制系统研究[D]. 上海: 上海交通大学, 2007:26 35.
- [5] ActivMEDIA ROBOTICS. Pioneer 2 Mobile Robot Operating Manual[ EB/ OL ] . [http://robots.activmedia.com/archives/pioneer users/p2opman2. pdf](http://robots.activmedia.com/archives/pioneer%20users/p2opman2.pdf), 1999. 10.
- [6] 付梦印, 邓志红. Kalman 滤波理论及其在导航系统中的应用[M] 北京: 科学出版社, 2003.
- [7] 毛克诚, 孙付平. 扩展卡尔曼滤波与采样卡尔曼滤波性能比较(解放军信息工程大学测绘学院, 河南郑州 450052).
- [8] An Introduction to the Kalman Filter. Greg Welch and Gary Bishop.TR 95-041.Department of Computer Science. University of North Carolina at Chapel Hill. Chapel Hill, NC 27599-3175.
- [9] 屠运武, 徐俊艳, 张培仁等. 自平衡控制系统的建模与仿真[J] 系统仿真学报, 2004, 16(4): 839 841.
- [10] 来清民. 传感器与单片机接口及实例 北京:北京航空航天大学出版社. 2008.1.
- [11] 黄智伟等. 32 位 ARM 微控制器系统设计与实践. 北京:北京航空航天大学出版社. 2010.3.
- [12] 潘炼. 检测技术及工程应用[M] 武汉:华中科技大学出版社. 2010.12.
- [13] Yamaguchi. Advanced Automated Fingerprint Identification system[C].32nd Annual 1998 International camahan conference on Security Technology,1998, 10:154-157.
- [14] Q.Fu, D.S.Mansjur, and B.H.Juang. Empirical System Learning for Statistical Pattern Recognition with Non-Uniform Error Criteria[J]. Signal Processing. 2010, 58(9): 4621-4633.
- [15] 王振宇. 两轮移动机器人平衡控制系统的研究[D]. 西南交通大学硕士论文. 2010
- [16] 袁泽睿. 两轮自平衡机器人控制算法的研究[D]. 哈尔滨工业大学硕士论文. 2008
- [17] 王晓宇. 两轮自平衡机器人的研究[D]. 哈尔滨工业大学博士论文. 2007
- [18] 邹细勇. 自主移动机器人的智能导航研究[D]. 浙江大学博士论文. 2006
- [19] 于咏生. 自主移动机器人导航中的若干问题研究[D]. 中国民航大学硕士论文. 2005

## 致 谢

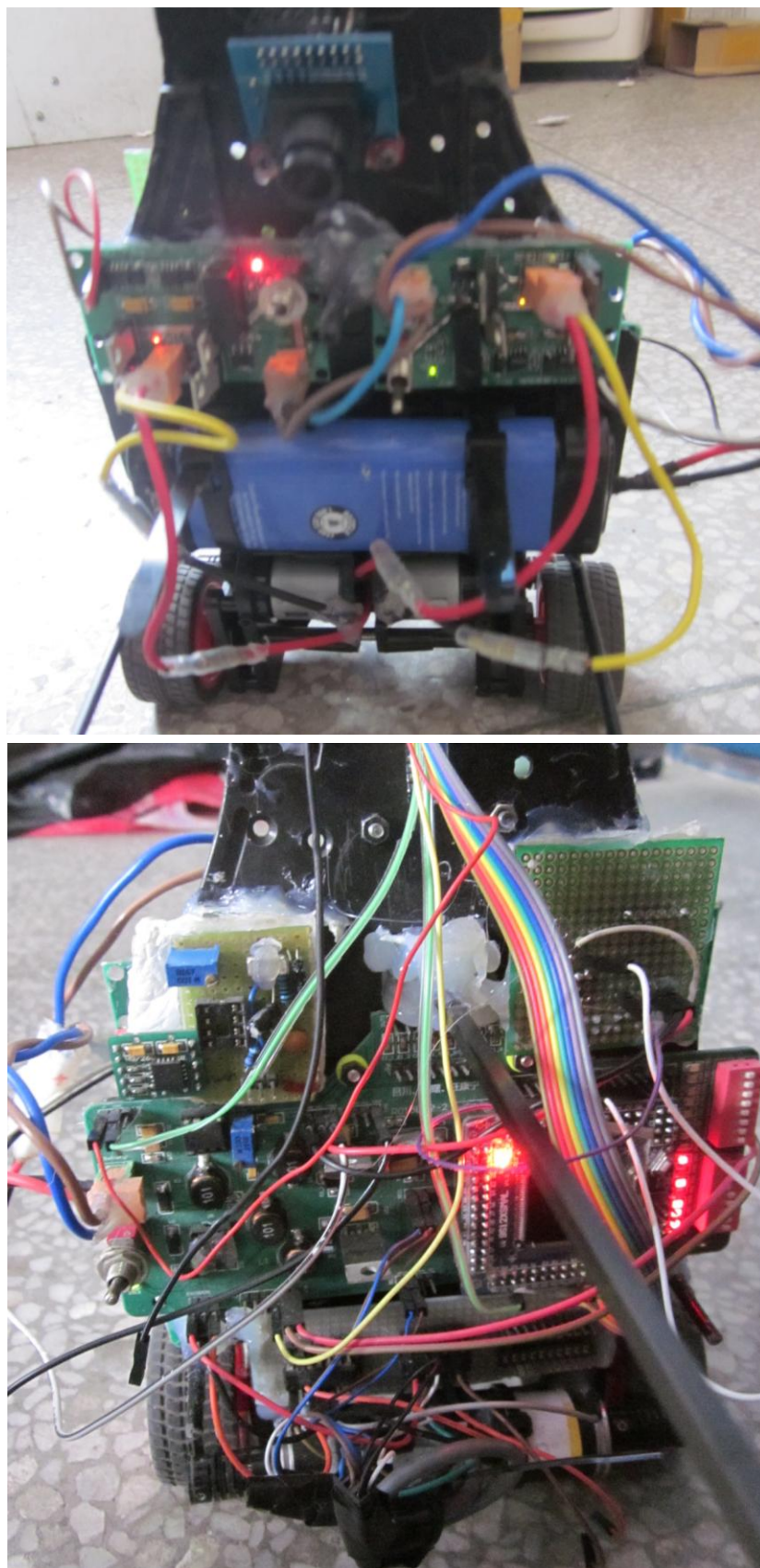
由于在智能汽车实验室积累了十几个月的项目经验，所以我选择了基于摄像头的两自平衡小车轮这个比较具有实用性和趣味性的课题，对我而言，这不仅给了我对智能控制领域和移动机器人领域有了一次更深层次的学习的机会，也是对我的大学四年所学知识的总结和综合检查。

刚接触这个课题的时候，我也是束手无策，但是经过和老师探讨，在电子开发网和各大技术论坛上和别人交流时，慢慢有了思路。其实这个项目也不是难的，在基本理论的学习过程中，感觉内容比较枯燥、繁多，但当我感觉自己学到了知识、有了进步的时候，我认为一切都是值得的。在进行具体的电路设计和程序设计的时候，我发现理论到实践的过程确实有相当的难度。但只要提前做好准备，并善于学习，勤于学习，没有什么事情是很困难的。并且在整个设计制作过程当中，我感觉收获也是非常大，不仅是理论和实践中的收获。更对我今后做的每一件有意义的事都会有很大的帮助和指导意义。毕业设计的完成意味着我的大学四年的学习生活即将结束，从此我将踏上一个新的学习，开始一段全新的生活。在此，我要感谢所有在我做毕业设计期间帮助过我的人。

首先我要感谢我的毕业设计导师潘老师的大力帮助和支持。我还要感谢在毕业设计期间帮助过我各位同学。从资料的查找，到电路板的制作，以及最后程序的调试，他们都给过我无私的帮助，在此，我要感谢他们的帮助。同时我还要感谢智能汽车实验室的章政老师，赵敏老师及实验室的每一位同学和师兄，是智能汽车实验室给我提供了我做项目的各种条件，是你们将我带入了技术之路。从你们每个人的身上我都能学到很多东西，是你们促进了我的成长。这次毕业设计的成功完成也是因为有了再智能汽车实验室十一个月的项目经验的积累，才能如此顺利。

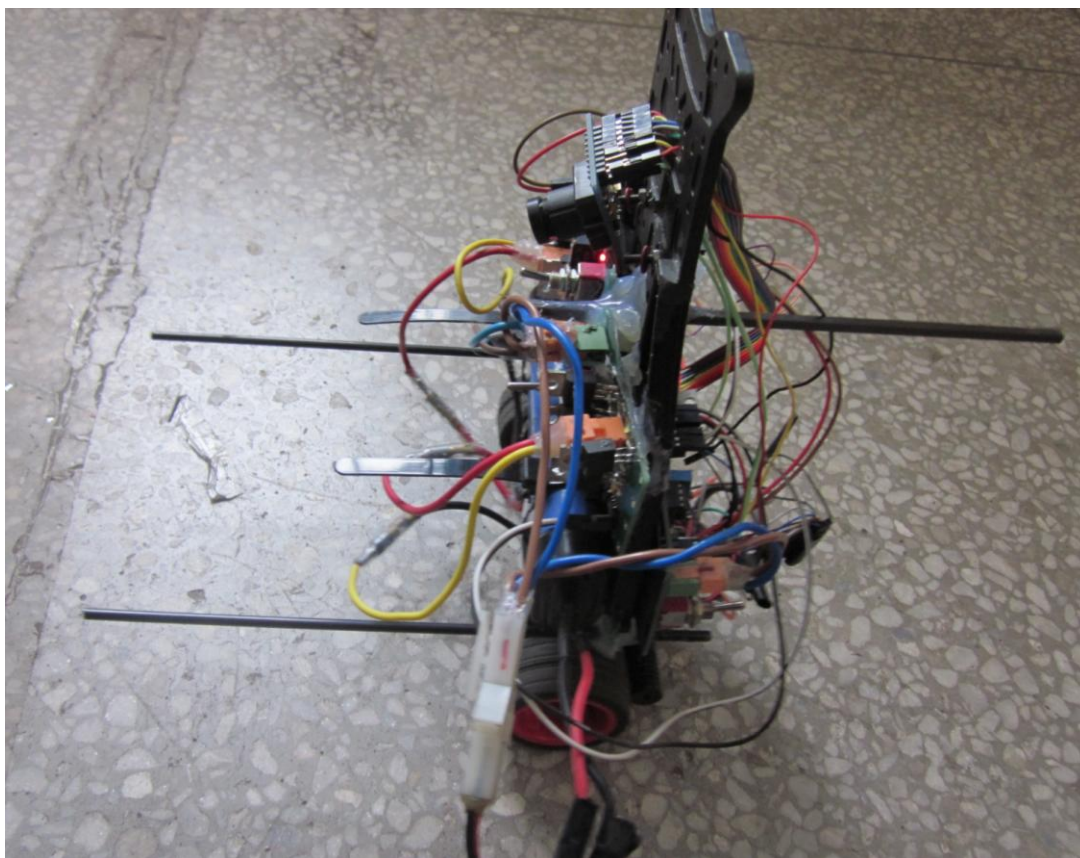
最后，我要感谢我的母校——武汉科技大学，大学四年，这里给我留下了美好的回忆。特别是在我即将踏上未来征程的同时，给了我这样一个锻炼的机会，使我加深了对以前知识的理解，拓宽了知识面，也提高了我对所学知识的综合的应用能力。祝愿母校明天更美好。

## 附录 A 整体设计实物图

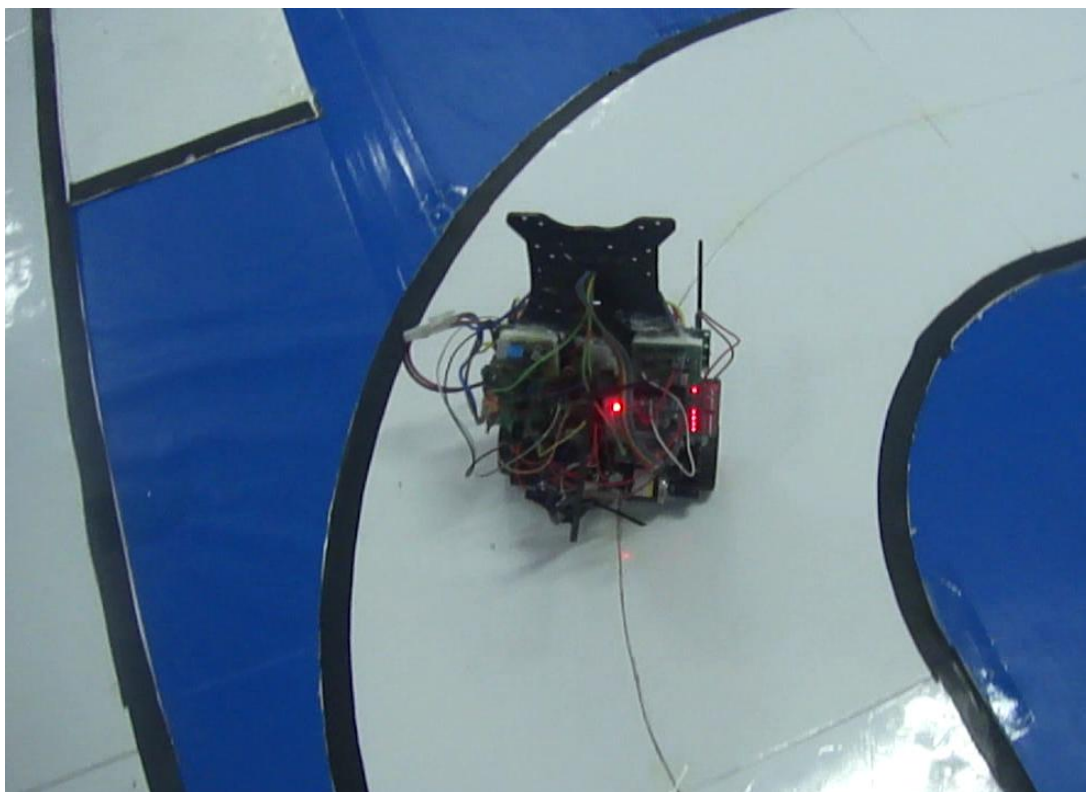




小车静态运行图

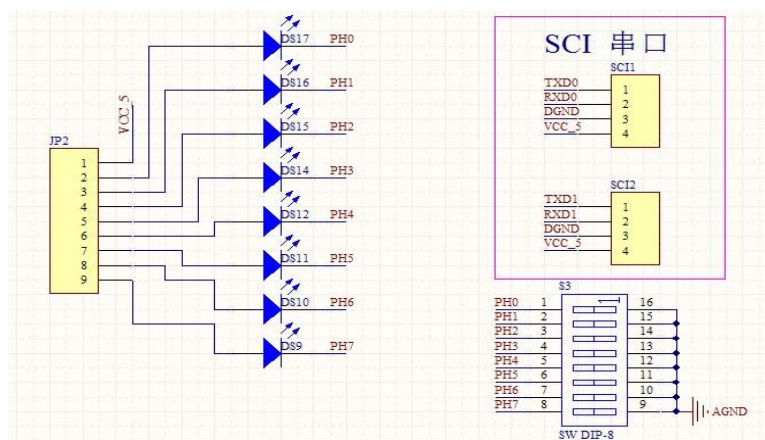


小车赛道动态效果

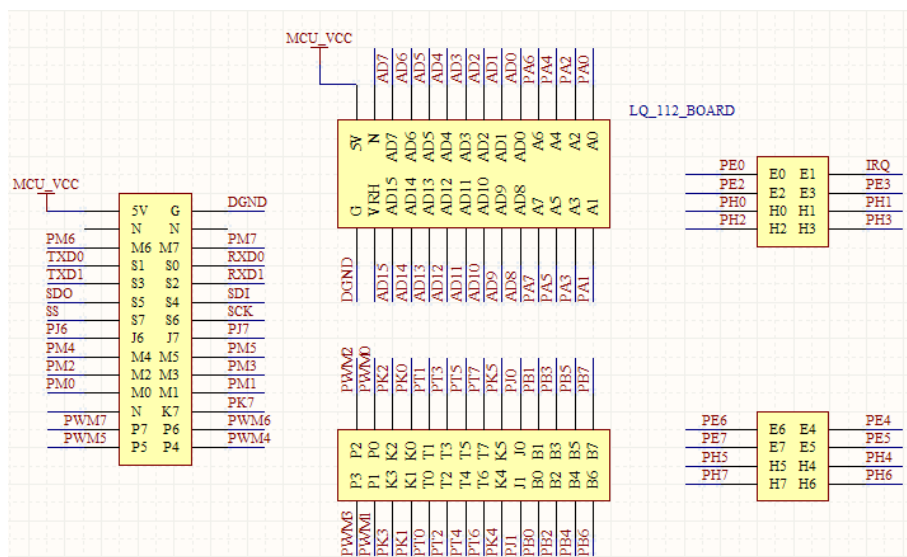


## 附录 B 部分硬件电路图

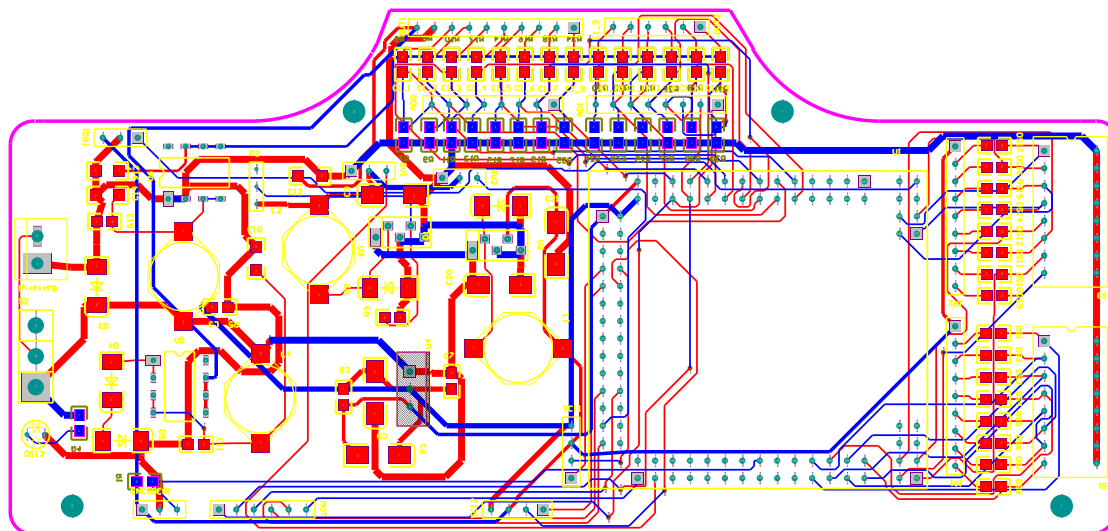
调试电路原理图:



主板部分电路原理图:



主板的 PCB 图:



## 附录 C 部分源代码

```
#include <hidef.h>          /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */
#include "math.h"

/*-----M A C R O S -----*/
#define VIDEO_POINT_NUMBER  61      //每行点数
#define VIDEO_LINE_NUMBER   40      //每场采集的行数
#define MIDLINE 19              //图像中间行
#define WHITE_BLACK_LATCH   140     //黑白电平阈值
#define VIDEO_MIDDLE_POINT  30      //采集图像中心值
#define ODD_EVEN_STATUS PTIM_PTIME //奇偶场信号
#define VIDEO_POINT_OUT     4       //行消隐区所占的点数
#define N_CHANNEL 2            //AD 采样通道数
#define N_SIGNAL 30           //每个周期 AD 采样的信号个数
#define N_CUN 30              //保存历史信息次数
#define CAR_SPEED_PERIOD 5
#define MIN_WHITELINE 5        //成为白线基准行的最小行数要求
/*-----行中断进程标志-----*/
#define VIF_WAITSAMPLE 1        //等待采样标志位
#define VIF_SAMPLELINE 2       //采样标志位
#define VIF_FINISH 3          //一场采样结束标志位
#define VIF_START 0           //采样开始标志位
#define VIF Videoclo_Flag
/*-----数组-----*/
byte Linelatch[ VIDEO_LINE_NUMBER ];          //阈值数组
byte VideoImage1[ VIDEO_LINE_NUMBER ][ VIDEO_POINT_NUMBER ]; //奇场
byte VideoImage2[ VIDEO_LINE_NUMBER ][ VIDEO_POINT_NUMBER ]; //偶场
byte VideoLine[ VIDEO_POINT_NUMBER + VIDEO_POINT_OUT ];
byte *VideoImage[ VIDEO_LINE_NUMBER ];        //图像处理数组
int Linepos[ VIDEO_LINE_NUMBER ];              //一场图像每行黑线中心存贮数组
int Rightlinepos[ VIDEO_LINE_NUMBER ];
int Leftlinepos[ VIDEO_LINE_NUMBER ];
int sto1[10];
int Steer_Out[10];
int AnglePwmMax=1000;

float angle,angle_dot;    //竖直方向的角度和角速度
float acce,gyro,w;
float Lastgyro;
float pre_Lastgyro;
float LeftRealspeed=0.0; //左轮速度实际值
float RightRealspeed=0.0; //右轮速度实际值
float LeftSpeed_filter=0.0;
```

```

float  RightSpeed_filter=0.0;
float  KPAngle=0.0;
float  AnglePID=0;
byte   Left_ZZ=1;
byte   Right_ZZ=1;
float  LeftSpeedPID=0.0;
float  RightSpeedPID=0.0;
float  Acce[N_CUN]={0};    //加速度计的历史值
float  Angle[N_CUN]={0};   //历史的角度值
byte   g_nCarSpeedCount=0; //中断服务程序计数
byte   g_nGetSpeedCount=0;

/*-----传感器-----*/
word   Pre_AD_Data[N_CHANNEL][N_SIGNAL]; //AD 采样值滤波之前存储的数组
float  AD_Data[N_CHANNEL];               //AD 过滤之后有效的值
float  Angle_Kp=100;
float  Angle_Kd=1.8;

/*-----电机-----*/
int SpeedMin=0;
int SpeedMax=1000;
int Expspeed=0;
int LeftSpeedPwm=0;
int RightSpeedPwm=0;
int LastLeftSpeedPwm=0;
int LastRightSpeedPwm=0;

/*-----标志位-----*/
Bool   image_ready;
Bool   OddEvenStatus;
Bool   Videoclo_Flag;
Bool   image_flag;
byte   g_ImageValidFlag,ddd_ValidFlag,dd_ValidFlag,d_ValidFlag ;
Bool   g_BaseValidFlag;

/*-----各变量-----*/
byte   latch; //黑白差值
byte   LineCount, Mid_Value;           //行中断采集行数计数
byte   VideoImageLine;                 //行中断实际采集行数计数
byte   g_validline_cnt;                 //有效行的数量
byte   g_Lvalidline_cnt;                //左有效行的数量
byte   g_Rvalidline_cnt;
byte   g_LineFlag;
byte   topline=0;                       //顶部黑线标志
byte   botline=VIDEO_LINE_NUMBER-1;    //底部黑线标志
char   Video_StartLine;                 //图像采集起始行
int     offset;                         //赛道弯曲程度
int     offset_dir;                     //偏移方向

```



```

int  servo_dir;
int  preoffset;
int  servo_pidout;           //偏移量
int  abs_servo_pidout;
int  SteerPwm=0;
int  servo_D=0;
int  averlinepos;           //平均值
word  g_BottomTenMiddle;    //底部六行的中心值
word  g_BaseLineMiddle;
word  g_preBottomTenMiddle;
word  timecount;
typedef struct PID{
    int  Kp_Left;
    int  Kp_Right;
    float Ki;
    int  Kd;
    int  error;              //当前速度偏差
    int  SumError;          //速度偏差的累积

    int  ExRightSpeed;       //右轮速度期望值
    int  ExLeftSpeed;        //左轮速度期望值

    float ek_Right;          //右轮速度偏差值
    float ek_Left;           //左轮速度偏差值

    float Sum_ek_Right;      //右轮速度偏差的累积
    float Sum_ek_Left;       //左轮速度偏差的累积
}PID;
PID  Vpid;
/*-----函数功能:锁相环初始化-----*/
void Init_PLL( void )
{
    CLKSEL = 0X00;           //disengage PLL to system
    PLLCTL_PLLON = 1;        //turn on PLL
    SYNRR = 0xD8;
    REFDV = 0x07;
    POSTDIV = 0x00;
    _asm( nop );             //BUS CLOCK=50M
    _asm( nop );
    while( !( CRGFLG_LOCK == 1 ) ); //when pll is steady ,then use it;
    CLKSEL_PLLSEL = 1;
}
/*-----函数功能:SCI 初始化-----*/
void Init_SCI0( void )

```

```

{
    SCI0BD = 0x147;
    SCI0CR1 = 0x00;
    SCI0CR2 = 0x3c;
}
/*-----函数功能:PWM 初始化-----*/
void Init_PWM(void)
{
    PWME=0x00;          //全关 8 个通道总的使能

    PWMCLK_PCLK1=1;      //选择时钟源 SA, 最大 65535, 0,1,4,5 通道可选
    PWMCLK_PCLK3=1;      //选择时钟源 SB, 最大 1150, 2,3,6,7 通道可选
    PWMCLK_PCLK5=1;      //选择时钟源 SA, 最大 65535
    PWMCLK_PCLK7=1;      //选择时钟源 SB, 最大 1150, 2,3,6,7 通道可选
    PWMCTL_CON01=1;       //通道 0、1 级联控制舵机
    PWMCTL_CON23=1;       //通道 2、3 级联用于控制驱动电机的正转
    PWMCTL_CON45=1;       //通道 4、5 级联用于控制驱动电机的反转
    PWMCTL_CON67=1;       //通道 6、7 级联用于控制尾部舵机
    PWMPRCLK=0;           // 预分频系数 0,A,B 使用总线时钟频率
    PWMSCLA=2;            //clockSA=clockA/(2*PWMSCLA)=12M
    PWMSCLB=2;            //clockSB=clockB/(2*PWMSCLB)=12M
    PWMPOL=0xff;          //初时输出高电平
    PWMCAE=0x00;          //左对齐方式
    PWMPER01=1200;        //设置周期为 1ms 频率=CLOCKSA/(PWMPERx)=1kHz
    PWMPER23=1200;        //设置周期为 1ms 频率=CLOCKSB/(PWMPERx)=1kHz    正转
    PWMPER45=1200;        //设置周期为 1ms 频率=CLOCKSA/(PWMPERx)=1kHz    反转
    PWMPER67=1200;        //设置周期为 1ms 频率=CLOCKSB/(PWMPERx)=1kHz
    PWMDTY45=0;           //Left ZZ
    PWMDTY23=200;         //Left FZ
    PWMDTY01=200;         //Right ZZ
    PWMDTY67=0;           //Right FZ
    PWME_PWME1=1;         //通道 1 输出允许    正转
    PWME_PWME3=1;         //通道 3 输出允许    反转
    PWME_PWME5=1;         //通道 5 输出允许    正转
    PWME_PWME7=1;         //通道 7 输出允许    反转
}
/*-----Init_ECT() 初始化 ECT-----*/
void Init_ECT(void)
{
    PERT_PERT7=1;         //上拉电阻
    TSCR2_PR = 7;         //128 分频
    TIOS = 0b00001111;    //PT7 acts as an input capture
    TCTL2 = 0b01010100;
    TCTL3 = 0x00;
}

```

```

    TCTL4 = 0x00;    //0 和 1 通道捕捉输入下降沿
    TFLG1_C2F = 1;   //通道 2 发生了中断事件
    TIE_C2I = 1;     //interrupt enable
    TC2 = 400;       // 1ms 中断一次  中断服务程序      */
    PACTL = 0x40;    //并成 16 位累加, //PT7 PIN,PACN32 16BIT,FALLing edge,NOT INTERRUPT
    PACNT = 0;
}
/*-----函数功能:IRQ 中断初始化-----*/
void Init_IRQ( void )
{
    PUCR = 0xD0;
    IRQCR_IRQE = 0;
    IRQCR_IRQEN = 0;    // Enaable IRQ
}
/*-----initialATD(void)初始化 AD-----*/
void Init_ATD(void)
{
    ATD0CTL1 = 0x20;
    ATD0CTL2 = 0x60;    //AD 模块不上电, 快速清零(每次读取结果自动清零)
    ATD0CTL3 = 0x88;    //右对齐(左对齐为 0x08), 转换序列长度为 1
    ATD0CTL4 = 0x22;    //10 位通道,4A/D 时钟周期,PRS=3
    ATD0CTL5 = 0x20;    //单通道采样, 连续转换
    ATD0DIEN = 0x00;    //禁止数字输入
}
/*-----GetADValue(void)读取传感器的采样值-----*/
void GetADValue(void) //查询方式,8 位精度, 右对齐, 数据存放在 ATDDRLx
{
    byte i;
    for(i=0;i<N_SIGNAL;i++)
    {
        ATD0CTL5_Cx=10;
        while(!ATD0STAT0_SCF);
        Pre_AD_Data[0][i]=ATD0DR0;
    }
    for(i=0;i<N_SIGNAL;i++)
    {
        ATD0CTL5_Cx=7;           //通道
        while(!ATD0STAT0_SCF);    //等待数据的转化
        Pre_AD_Data[1][i]=ATD0DR0; //结果
    }
}
/*-----函数功能:对 AD 的采样值进行均值滤波-----*/
void AD_Filter()           //AD 的滤波 可直接取平均值
{

```

```

int sum,i,j;
for(i=0;i<N_CHANNEL;i++)
{
    sum=0;
    for(j=0;j<N_SIGNAL;j++)    //采样点数
    {
        sum+=Pre_AD_Data[i][j];
    }
    AD_Data[i]=sum/N_SIGNAL;
}
}
/*-----函数功能:图像的综合处理-----*/
void ImageProcessing( void )
{
    //ImageInit();
    Mid_filter();
    GetLineLatch();
    Mid_filter();
    SearchBaseLine();
    GetBlackLine();
    GetRoadInformation();

    ddd_ValidFlag = dd_ValidFlag;           //递推赋值
    dd_ValidFlag = d_ValidFlag;
    d_ValidFlag = g_ImageValidFlag;
    g_preBottomTenMiddle=g_BottomTenMiddle; //基准行的中心值
}

void main(void) {
    /* put your own code here */
    Init_PLL() ;
    Init_ECT() ;
    Init_IRQ() ;
    Init_ATD() ;
    Init_PWM() ;
    Init_Time_flag();
    Video_distance() ;
    ImageInit() ;
    Init_Latch() ;
    OddEvenStatus = ODD_EVEN_STATUS ;
    VIF = VIF_START ;
    image_ready = 0 ;
    g_ImageValidFlag=1;
    g_BaseValidFlag=1;
}

```

```
Mid_Value = 30 ;

DDRB = 0xff; //输出
DDRH = 0x00; //输入
PORTB= 0b01010101;

//DDRS_DDRS0=0;
//DDRS_DDRS1=0;
DDRS_DDRS2=1; //输出 模拟数字通道计数
DDRS_DDRS3=1; //输出 模拟数字通道计数

DDRJ = 0x00; // PJ0 判断外部中断 行中断
PPSJ = 0x01; // Port J Polarity Select Register-rising EDGE
PIEJ = 0x01; // VIDEO SYNC INTERRUPT DISABLED,BUT NOT IN MAIN()
PERJ = 0xff;

EnableInterrupts;
TSCR1_TEN = 1 ; //定时器使能

for(;;) {
    _FEED_COP(); /* feeds the dog */

    if( image_ready == 1 )
    {
        if( image_flag == 1 )
        {
            ArrayCopy( 1 );
        }
        else if( image_flag == 0 )
        {
            ArrayCopy( 2 );
        }
        else{}
        GetPort();
        ImageProcessing();
        image_ready = 0 ;
    }
} /* loop forever */
/* please make sure that you never leave main */
}
#pragma CODE_SEG __NEAR_SEG NON_BANKED
void interrupt 10 TimerC2I(void) //通道 2 定时器中断
{
    TC2+=400; //1.024ms
```

```

TFLG1_C2F = 1; //清中断标志位
g_nCarSpeedCount++;
if(g_nCarSpeedCount >= CAR_SPEED_PERIOD) //读取电机脉冲数
{
    g_nCarSpeedCount=0;
    g_nGetSpeedCount++;
    if(g_nGetSpeedCount>=4)
    {
        GetRightSpeed();
        PTS_PTS2=1;
        PTS_PTS3=0;
        g_nGetSpeedCount=0;
    }
    else if(g_nGetSpeedCount==2)
    {
        GetLeftSpeed();
        PTS_PTS2=0;
        PTS_PTS3=1;
    }
}
else if(g_nCarSpeedCount == 1)
{
    GetADValue(); //AD 采样
    AD_Filter(); //AD 滤波
}
else if(g_nCarSpeedCount == 2)
{
    GetAngle(); //计算角度和角速度,卡尔曼滤波
}
else if(g_nCarSpeedCount == 3)
{
    Speed_control(); //直立控制和前进速度控制
}
else if(g_nCarSpeedCount == 4)
{
    DirectionControl(); //通过 PD 调节控制转向获得 SteerPwm
    Steer_control();
}
}
#pragma CODE_SEG DEFAULT

```