

第七届全国大学生“飞思卡尔”杯 智能汽车竞赛 技 术 报 告



常熟理工学院
CHANGSHU INSTITUTE OF TECHNOLOGY

学 校：_____常熟理工学院_____

队伍名称：_____闪电四队_____

参赛队员：_____宋长春、刘清、李荣伟_____

带队教师：_____李鑫、戴梅_____

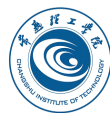
关于技术报告和学术论文使用授权的说明

本人完全了解第七届“飞思卡尔”杯全国大学生智能汽车邀请赛关保留、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：_____



摘要

本文以第七届“飞思卡尔”杯全国大学生智能汽车竞赛为背景介绍了常熟理工学院磁导航两轮自平衡智能小车的软硬件开发流程。本届电磁组比赛采用的是大赛组委会统一指定的 C 型车模。在 C 型车模的基础上，采用了以 freescale 半导体公司出产的 16 位 DSCMC56F8366 单片机为核心处理器，通过陀螺仪获得角加速度，加速度计获得角速度，两者经卡尔曼滤波之后得到的数据使得车模保持直立，最后以电感为主要的路径检测传感器。当检测到路径之后给车模以倾角，使得车模在寻求直立但无法直立的过程中保持一定的速度运行。整个智能车的设计涉及到了车模直立数学建模、车模机械结构的调整、传感器的布局、信号处理、控制算法等多个方面。整个车模的关键在于将车模直立，车模一旦直立基本上已经成功了一半。当然，智能车竞赛更加注重的是车模如何在硬件限制的条件下获得更高的速度，因此有必要对直立控制、速度控制、方向控制的 PID 参数进行整定。下文会介绍 PID 参数中哪些参数的改变对直立控制特别明显，哪些对速度控制明显，而又有哪些参数对方向控制明显。因为两轮直立车的特殊性，且时间非常紧凑，又且我们仅仅用了两只电感，所以没法检测赛道的路径，不能随意控制小车的速度，这不失为一次小小的遗憾。

关键词：两轮自平衡、电感、PID、卡尔曼滤波器

目录

摘要.....	3
第一章 引言.....	6
1.1 智能车研究背景.....	6
1.1.1 智能车研究历史与发展.....	6
1.1.2 两轮自平衡小车.....	6
1.2 大赛简介.....	7
1.3 章节安排及文献综述.....	8
第二章 车模原理.....	10
2.1 控制系统要求分析.....	10
2.2 直立控制.....	11
2.2.1 倒立摆原理.....	11
2.2.2 姿态检测系统.....	13
2.3 速度控制.....	18
2.4 方向控制.....	19
第三章 智能小车整体设计.....	20
3.1 系统结构框图.....	20
3.2 改造后车模结构参数（表）.....	21
第四章 硬件设计.....	21
4.1 C 型车模改造.....	21
4.2 各模块安装.....	23
4.2.1 姿态检测传感器安装.....	23
4.2.2 电磁传感器安装.....	24
4.2.3 编码器安装.....	24
4.2.4 电池安装.....	26
第五章 电路设计.....	27
5.1 核心底板.....	27
5.1.1 整体电路.....	27
5.1.2 电源电路的设计.....	28
5.2 驱动模块.....	28
5.3 磁导航模块.....	29



第六章 软件设计.....	30
6.1 DSC 片内资源及其配置.....	30
6.2 DSC 底层配置.....	32
6.3 直立控制算法.....	33
6.4 速度控制算法.....	35
6.5 方向控制算法.....	35
第七章 车模开发工具及调试过程.....	36
7.1 开发环境.....	36
7.1.1 Codewarrior IDE 功能介绍.....	36
7.1.2 ProcessorExpert 处理器专家.....	36
7.2 蓝牙模块.....	37
7.3 虚拟示波器.....	38
第八章 总结.....	40
参考文献.....	41
致谢.....	42
附录.....	43
1、PCB 底板图.....	43
2、程序.....	43

第一章 引言

1.1 智能车研究背景

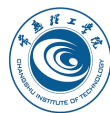
1.1.1 智能车研究历史与发展

汽车已经过了 100 多年的发展历史,从诞生的那一天起,它就从未停止过智能化发展的步伐。当前的信息技术革命正在推动汽车设计翻开新的一页,自动化、智能化、多功能已经成为了汽车发展的新趋势。电脑技术、自动控制技术以及现代信息融合技术和通信等高科技与汽车的结合,使未来汽车成为更安全、更方便、更舒适的交通工具。智能车系统是一个集环境感知、规划决策、多等级辅助驾驶等功能于一体的综合系统,是典型的、多学科的、综合性的高科技和高新技术的结合体,涉及传感器技术、信息融合技术、微电子技术、计算机技术、智能自动控制技术等。20 世纪 90 年代以来,随着汽车市场竞争激烈程度的日益加剧和智能运输系统(ITS)研究的兴起,国际上对于智能汽车及其相关技术的研究成为热门,一批有实力、有远见卓识的汽车行业大公司、研究院所和高等院校也正展开智能汽车的研究。目前它已成为世界众多发达国家重点发展的智能交通体系中的重要组成部分,也是世界车辆工程领域研究的热点和汽车工业增长的新动力。^[2]

智能车小车是研究智能车的一个缩影。智能小车上的许多想法、算法、结构在实际智能车上都可以有所借鉴。

1.1.2 两轮自平衡小车

今年新增的项目磁导航两轮自平衡小车有必要着重提一下。两轮自平衡小车是一个高度不稳定两轮机器人,其动力学方程是一多变量、严重不稳定、强



耦合、时变、具有参数不确定性的非线性高阶方程，加上运动学方程中的非完整性约束，要求完成的控制任务也具有多重性，即在完成系统平衡控制任务的同时，还要在复杂的环境下完成路径跟踪任务或自主移动的任务，因此，两轮自平衡小车是一个具体的、实现起来相对便宜的复杂系统，给控制理论提出了很大的挑战，是检验各种控制方法处理能力的典型装置，受到世界各国科学家的重视，成为具有挑战性的课题之一。两轮自平衡小车作为一种复杂系统的实验装置，其控制难度大，控制算法复杂，很适于理论研究、实验和仿真。在此实验系统上可进行解耦控制、不确定性系统控制、非线性系统控制、自适应控制、复杂系统分散协同控制等研究，且物理意义明显、观察方便，系统具备复杂系统的特点但其本身又不很复杂，价格低廉，占地面积小，是很有潜力的实验工具。其次建立在自平衡基础之上的两轮小车的其他性能，例如复杂环境下的导航、拐角的转弯等问题也极有可能解决在工业生产和社会生活中的许多挑战。^[1]

磁导航两轮自平衡小车的研究能够极大地锻炼学生对于控制理论的理解，是对控制理论软硬件的一次很好的检验。

1.2 大赛简介

飞思卡尔杯全国大学生智能车竞赛、全国大学生数学建模竞赛、电子设计大赛、机械设计大赛和结构设计大赛这些赛事是理工科院校的五大赛事。飞思卡尔杯全国大学生智能车竞赛参赛人数广，影响范围大，竞技难度高，专业知识强，对学生实践素质的要求高。该大赛与其它大赛不同的是，该竞赛融科学性、趣味性和观赏性为一体，是以迅猛发展、前景广阔的汽车电子为背景，涵盖自动控制、模式识别、传感技术、电子、电气、计算机、机械与汽车等多学科专业的创意性比赛。

全国大学生“飞思卡尔”杯智能汽车竞赛由竞赛秘书处设计、规范标准硬

软件技术平台，竞赛过程包括理论设计、实际制作、整车调试、现场比赛等环节，要求学生组成团队，协同工作，初步体会一个工程性的研究开发项目从设计到实现的全过程。

本届比赛已经是第七届。从该比赛的发展趋势来看，该比赛每届均有不同，每届均有拔高。尤其是本届比赛，竞赛规定赛车循一根黑线变为循两根，赛道与实际路面更加贴切。而新增加的两轮直立的电磁组则更是一次对控制算法的挑战。

1.3 章节安排及文献综述

第一章为引言，主要介绍了本次比赛的背景，着重强调了两轮自平衡车并引出下文。

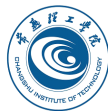
第二章介绍了车模原理，将车模的原理分为了三块，直立控制，速度控制和方向控制。其中两轮自平衡小车的发展参考了文献【1】。由于原理性的东西极难说清楚，又在直立控制、速度控制和方向控制参考了竞赛秘书处的【2】。其中还涉及到了PID算法，这参考了【3】。陀螺仪和加速度计两种参数的融合参考了【4】【5】【6】【7】。

第三章叙述了智能小车的整体设计，为下文模块设计做引文。

第四章为车模的硬件设计，介绍了车模如何改装，为什么这么改装；讨论了车模传感器、电池的安裝以及为什么安装在这个位置。其中编码器的叙述，我们参考了【8】。

第五章介绍了车模的电路设计。电路是承载整个系统运行的关键，我们贴出了自己的电路图，叙述了电路设计时的一些注意点。其中对一些芯片的叙述主要参考了生产商提供的 datasheet。如 BTS7970B 参考了【9】。

第六章是软件设计。简单叙述了本组使用的 MC56F8366 和我们小车相对应



的核心算法。MC56F8366 简介参考了【10】，而其底层资源的配置参考了【11】。

第七章介绍了车模开发工具及调试过程。叙述了车模开发环境，这包括 Codewarrior IDE 功能介绍，ProcessorExpert 处理器专家，调参工具，虚拟示波器以及函数信号发生器和计时器。Codewarrior IDE、ProcessorExpert 参考了【12】。

第八章为总结，总结本届比赛我们从做智能小车上学到的东西。

第九章为致谢，感谢那些人那些事，那些我们与小车相伴的日子。

第二章 车模原理

2.1 控制系统要求分析

本届电磁组比赛要求较为特殊，要求车模能够仅仅依靠后轮来沿着赛道比赛。相对于四轮着地的车来说，其控制算法是复杂的，控制过程是艰巨的。正因其特殊性，所以有必要对车模的原理做研究性的说明。

竞赛给出了电磁组的参考设计方案，按照参考设计方案，将车模运动的控制分为三个模块：1、直立控制模块；2、速度控制模块；3、方向控制模块。直立控制模块主要依靠电机的正反转使车模加速运动保持直立。在速度控制模块上面，通过给车模以倾角，使车模在保持直立但无法直立的状态中以一个较为恒定的速度运行。当然这个速度最终是要加到电机上面的。方向控制模块主要依靠了两个电机的差速。



图 2.1 直立控制优先

如何来平衡三个模块之间的关系呢？在直立控制、速度控制、方向控制中以直立控制最为重要。直立控制是方向控制和速度控制的基石。没有直立控制



就无所谓速度和方向控制。以直立控制的角度，速度控制和方向控制都是干扰。
速度

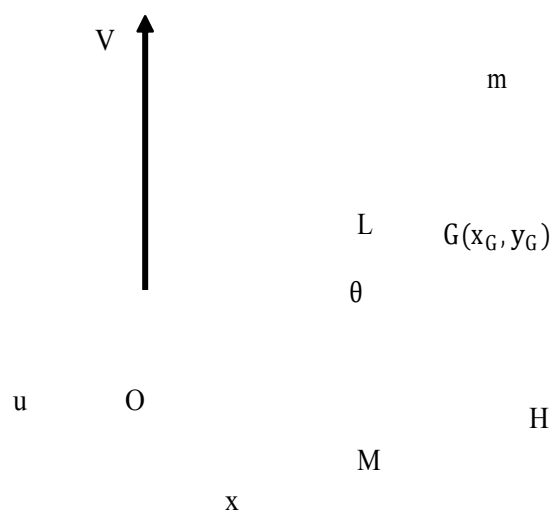
控制是给定倾角，方向控制是给定差速，这与保持直立的状态是矛盾的。而方向控制是在速度控制的基础上才能运行的。因此在后文的软件设计中要以直立为先。另外，速度和方向控制耦合在电机上面的值不是想给多少就直接给多少，而是计算出给的值之后分到多少个周期内慢慢给上去的。这保证了车模运行的稳定性。

在实际制作车模的三个月中，我们在直立控制上面投入了大量的精力，最后调到仅仅比例控制就能直立的程度，当然抗干扰性不会很好。但是，直立控制只有到这个水准再进入下两个步骤才是严谨的、科学的。^[2]

2.2 直立控制

2.2.1 倒立摆原理

上面已经说到直立控制是整个系统的关键，直立只有达到单单通过比例控制就能站立的程度，车模的速度才有可能提的上去。实际上，车模是一个单级



倒立摆模型。

图 2.2 倒立摆

倒立摆系统是控制研究中的一个典型问题。控制的目标是通过给小车底座施加一个控制量，使小车停留在预定的位置，并使杆子不倒下，即不超过一预先定义好的垂直偏离的角度范围。如图所示，小车质量为 M ，摆的质量为 m ，小车位置为 x ，摆的角度为 θ 。

设摆杆偏离垂直线的角度为 θ ，同时规定摆杆的重心坐标为，则

$$\begin{aligned} x_G &= x + l \sin \theta \\ y_G &= l \cos \theta \end{aligned} \quad (\text{公式 1})$$

则根据牛顿运动定律，建立水平和垂直运动状态方程。

摆杆围绕其重心的转动运动可以用力矩方程来描述

$$(I \text{ 为摆杆围绕其重心的转动惯量}) \quad (\text{公式 2})$$

摆杆重心的水平运动由下式来描述

$$(\text{公式 3})$$

摆杆重心的垂直运动由下式来描述

$$(\text{公式 4})$$

小车的水平运动由下式来描述

$$(\text{公式 5})$$

当 θ 很小时，。则以上各式就变为

$$(\text{公式 6})$$

$$(\text{公式 7})$$

$$0 = V - mg \quad (\text{公式 8})$$

$$(\text{公式 9})$$

将式(2.1)、式(2.2)、式(2.3)、式(2.4)求解可得到单级倒立摆方程如下

(公式 10)

(公式 11)

式中, $J=$ 。

从式(公式 10)和式(公式 11)就可以知道控制指标一共有 4 个, 即单级倒立摆的摆角、摆速。^[3]

实际上, 小车在运动的时候, 是不需要知道位置 x 的, 也就是说, 真正控制量只有三个: 1、摆角; 2、摆速 (车模倾角加速度)。那么, 车模倾角, 车模倾角加速度, 是需要测量的。在后文的硬件设计中, 将会对此详述。

2.2.2 姿态检测系统

姿态检测系统主要包括了三个部分: 1、陀螺仪; 2、加速度计; 3、卡尔曼滤波器

(1) 陀螺仪

陀螺仪在系统中可以用于测量车模倾角加速度。

陀螺仪是一种用高速回转体的动量矩敏感壳体相对惯性空间绕正交于自转轴的一个或二个轴的角运动检测装置, 可以用于检测角速度。竞赛允许使用的陀螺仪是日本村田公司基于压电陶瓷技术的单轴陀螺仪 ENC-03(如图 2.3 所示)。其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 体积小, 响应快, 功耗低, 成本低。采用模拟量输出, 检测范围可达 $\pm 300\text{deg/sec}$ (度每秒), 灵敏度为 0.67mv/deg/sec 。

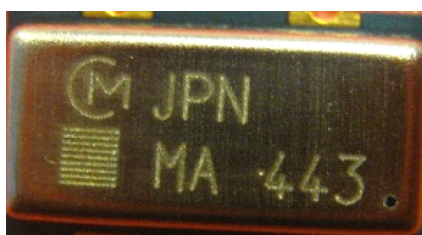


图 2.3 陀螺仪 ENC-03

陀螺仪直接输出角速度，将角速度进行积分便可以得到角度。陀螺仪输出数据噪声较少，短时误差较小。由于陀螺仪及其放大电路存在温漂，且需要经过积分运算，最终会导致误差累积，致使检测结果出错。实际情况确实如此。我们组曾经在有空调和无空调的环境下试验加速度的温漂。

那么是否需要抑制温漂呢？有一种解决方案是，在车模的顶部与底部同时安装两个陀螺仪，通过两个陀螺仪的差值求出角加速度。差值能够很有效的将温漂因素弱化掉。但是这无疑会增加机械结构的复杂性。另一种解决方案是利用加速度计获得的实际角度值进行校正。这是需要使用卡尔曼滤波器将积分得到的加速度和角加速度融合成为一个数据。

陀螺仪 1

陀螺仪 2

图 2.4 安装两个陀螺仪

(2) 加速度计

加速度计主要用于测量的值。

加速度计是一种利用检测质量块的惯性力来测量载体加速度的敏感装置，分为线加速度计和角加速度计。竞赛允许采用飞思卡尔公司利用 MEMS 微电子技术开发生产的三轴加速度计 MMA7260。MMA7260 是一种低 g 值、小量程线

性加速度传感器，在不运动或不受重力作用（0g）条件下输出为 1.65V，最大测量范围 0-6g，灵敏度最高可到 800mV/g，MMA7260 实物如图所示。

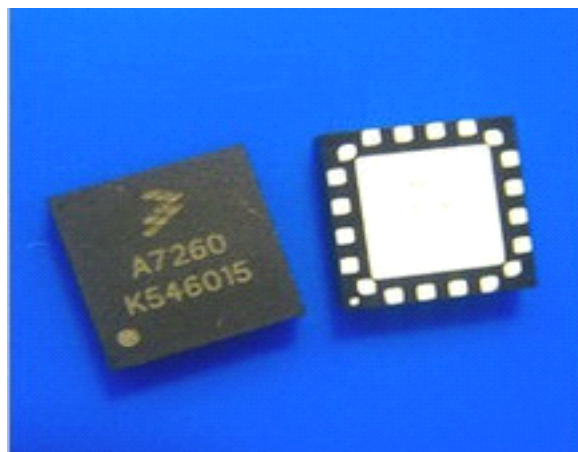


图 2.5 加速度计 MMA7260

加速度计可以直接通过反三角函数计算出小车倾斜角度，但是其对震动非常敏感，输出值中含有大量噪声，而且其输出的值是小车运动加速度与重力加速度的混合数据。因此不能直接使用。

（3）卡尔曼滤波器

虽然单一惯性传感器就可以单独进行姿态角度检测，但是其准确性主要取决于惯性器件的精度，单从改善硬件结构和生产工艺方面难以有很大幅度的提高，并且系统误差会随时间累积，不适用于长时间姿态检测。由于利用单一传感器（陀螺仪或加速度计）难以获得相对真实的小车姿态角度，出于对系统测量姿态角度准确性的考虑，本系统采用多传感器信号进行数据融合，以获得最佳姿态角度。^{[4][5]}

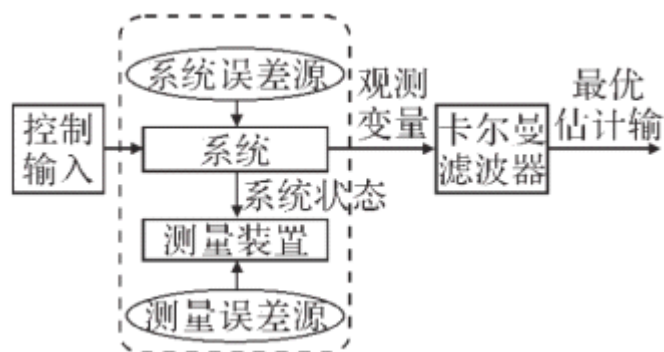
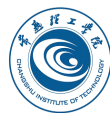


图 2.6 卡尔曼滤波

多传感器数据融合是一个非常重要的研究内容，只有采用最适合的融合方法才能获得最佳的效果。常用数据融合方法有加权平均法，神经网络法等。加权平均法是一种简单的融合方法，故其运算精度很差；神经网络法具有很好的非线性和有效的自学能力，但是其涉及的模型构建，参数优化非常复杂，不适用于本系统。

1960 年卡尔曼发表了著名的用递归方法解决离散数据线性滤波问题的论文。随着数字计算技术的进步，卡尔曼滤波器得到了越来越广泛的应用和推广，尤其是在自主或协助导航领域。卡尔曼滤波器与大多数滤波器不同之处，在于其是一种纯粹的时域滤波器，不需要像低通滤波器等频域滤波器那样，需要在频域设计再转换到时域实现。对于解决大部分的问题，是最优，效率最高甚至是最有用的。卡尔曼滤波器的广泛应用已经超过 30 年，包括机器人导航，控制，传感器数据融合甚至在军事方面的雷达系统以及导弹追踪等等。近年来更被应用于计算机图像处理，例如头脸识别，图像分割，图像边缘检测等等。国外有研究者根据加速度计与陀螺仪的互补特点研究出互补滤波算法，其简单明了并且具有较好的实时性与稳定性，能够较好的融合出姿态角度。考虑到本系统使用的惯性器件特性较差，互补滤波在本质原理上不能弥补器件特性缺陷，所以本系统以卡尔曼滤波器为主要数据融合方式。^{[6][7]}

卡尔曼滤波器是一种高效率的递归滤波器(自回归滤波器)，能够从一系列的不完全及包含噪声的测量中，估计动态系统的状态。卡尔曼滤波器不仅能估计



信号的过去和当前状态，甚至能估计将来的状态。

卡尔曼滤波器解决离散时间控制过程的一般方法，首先定义模型线性随机微分方程。假设卡尔曼滤波模型 k 时刻真实状态是从 $(k-1)$ 时刻推算出来，如

下

(公式 12)

式 (公式 13) 中， x_k 是 k 时刻状态； A 是 $k-1$ 时刻状态变换模型； B 是作用在控制器向量上的输入控制模型； w_k 是过程噪声，假设其均值为零，协方差矩阵符合多元正态分布

k 时刻对应真实状态的测量满足下式：

(公式 15)

式 2.9 中是观测模型，将真实控制映射为观测空间； v_k 为观测噪声，其均值为零，协方差矩阵符合正态分布：

(公式 16)

初始状态以及每一时刻的噪声都认为是互相独立的。

卡尔曼滤波器的操作主要包括两个阶段：预估与更新。在预估阶段，滤波器根据上一时刻状态，估算出当前时刻状态；在更新阶段，滤波器利用当前时刻观测值优化在预估阶段获得的测量值，以获得一个更准确的新估计值。

卡尔曼滤波器迭代过程如下：

1. 先验状态估计：

(公式 17)

2. 先验估计误差协方差

(公式 18)

3. 卡尔曼增益

(公式 19)

4. 后验状态估计

(公式 20)

5. 后验误差协方差

在上面各式中：

A: 作用在上的 n 阶矩阵；

B: 作用在控制向量上的 $n \times 1$ 输入控制矩阵；

H: $m \times n$ 观测模型矩阵，将真实状态空间映射为观测空间；

: $n \times n$ 先验估计误差协方差矩阵；

: $n \times n$ 后验估计误差协方差矩阵；

Q: $n \times n$ 过程噪声协方差矩阵；

R: $m \times m$ 过程噪声协方差矩阵；

I: n 阶单位矩阵；

: $n \times m$ 矩阵，称之为卡尔曼增益。

2.3 速度控制

直立车不同于四轮着地的车，它的速度并不是直接由供给电机的电压决定。车模需要保持平衡，一旦速度控制干扰了平衡，速度将无法继续维持下去。车模传感器的实际测量和车模真实的倾角是不一样的。在重力的作用下，车模就会朝倾斜的方向加速前进。控制车模的速度主要要解决三个问题：1、车模速度的测量；2、车模的直立控制实现车模倾角的改变；3、速度误差控制车模倾角

第一个问题可以通过安装在电机输出轴上的编码器来测量得到车模的车轮速度。第二个问题可以通过角度控制给定值来解决。给定车模直立控制的设定值，在角度控制调节下，车模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。第三个问题解释复杂。控制方法是，首先陀螺仪和加速度计要提供倾角和倾角加速度。车模通过编码器得到的速度和车模的给定值相合成，乘上系数后变为车模倾角给定量，最后就给电机。

2.4 方向控制

本系统主要采用了两只电感，感应赛道中心铺设的 100mA,20kHz 的交流导线，并将两只电感得到的值作差，乘上一个系数后便可以得到车模偏离中心的距离，通过 PID 算法，平滑差速控制车模的两只电机。（如图 2.7 所示为两只电感图）方向控制是和速度控制以及直立控制耦合的。那么，经过 PID 控制器之后的输出值，一个以正值的形式耦合到某一个电机上，另外一个则以负值形式耦合到另外一个电机上。在电机工作于线形区的时候，两只电机就可以实现差速控制了。方向控制中，最主要的是比例控制。后文软件设计会说明这些参数对车模的影响。

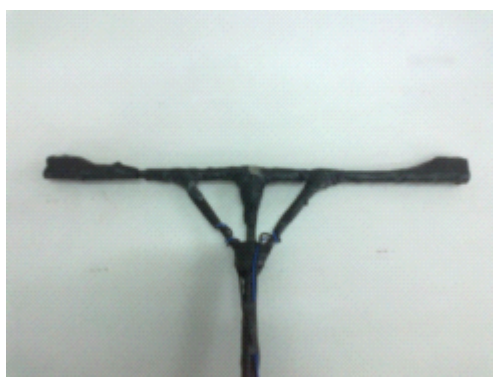


图 2.8 所示为两只电感图

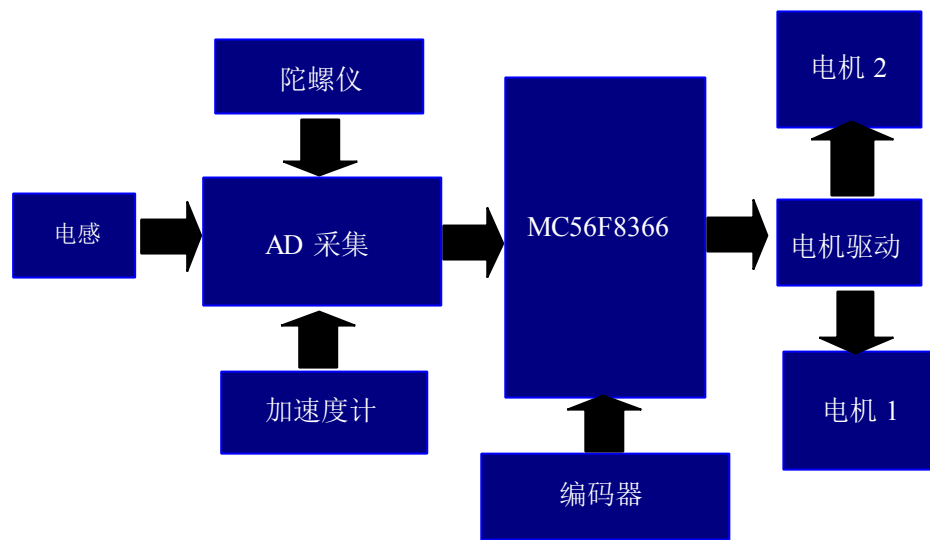
在逐步制作车模的过程中，我们发现，车模在仅仅比例的控制下，无论参数调得多么优越，总是无法解决车模转向过冲问题，转弯很不平滑，有卡顿现象。这时候，需要知道车模方向变化率来对转向进行修正。根据官网和实际的

摸索，在车模中间，有必要在中间安装转向陀螺仪。

第三章 智能小车整体设计

3.1 系统结构框图

本磁导航两轮自平衡智能小车是以陀螺仪、电感、加速度计、编码器为主要传感器，以 MC56F8366 为控制核心通过驱动电路控制两个电机正反运动的非线性控制系统。整个系统的结构框图如下：



智能车系统结构框图

图 3.1 系统整体框图

3.2 改造后车模结构参数（表）

类别	参数
----	----



车模总体重量(kg)	1.5
车长（带杆.mm）	440
车高(mm)	280
车宽(mm)	240
电路功耗(w)	3.8
电容重量（uf）	1500
传感器个数	5
传感器类型	陀螺仪，加速度计，工字型电感

第四章 硬件设计

4.1 C 型车模改造

本系统采用的是东莞市博思电子数码科技有限公司出产的C型车模。C型车模的特点是后轮有两个电机，适合于做电磁组的智能车。但是从官网上购买的车模有许多多余的物件，所以有必要对车模进行在竞赛规则允许范围内的改造。（如图所示为竞赛车模）

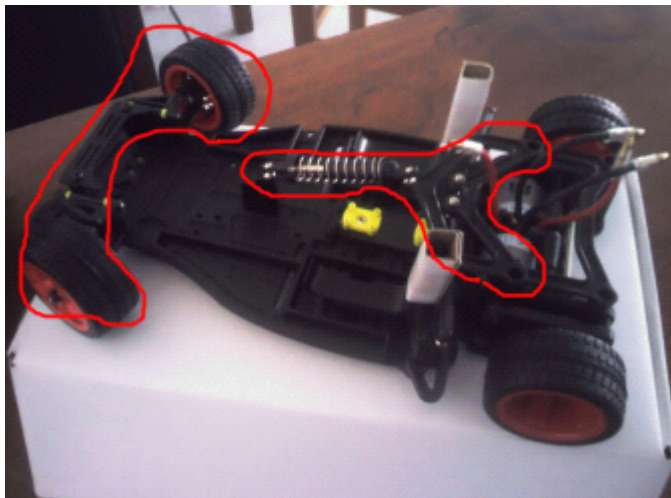


图 4.1 改装前的 C 车模

如图所示，将图片中圈出的部分去掉，将车模质量降低，使车模结构简洁。C 型车前后是活动的，这对直立是不利的。官网提供的方案是用热熔胶打死，但是热熔胶的机械性能是不牢靠的，对热的抵抗性是差劲的。当车模运动时，其两个电机产生的热量很容易熔化热熔胶，长期可能会导致车模变形甚至弯折。经过试验，如果用较大的力来回晃动固死的车模时，发现车模有明显弯折现象。所以只用热熔胶固死是不严谨的。

其实，C 车模的底板提供了大量的螺丝口，利用现有螺丝口和裁剪、打孔的万用板，可以将前后两个模块牢固拼接。



图 4.2 车模底板改装

4.2 各模块安装

两轮自平衡车比较特殊，单单靠两只轮子站立，这需要车体重心尽量下移。这和不倒翁的原理有相似之处。只不过不倒翁在受力时会重心升高，而智能车不会。但是上轻下重的物体比较稳定，也就是说重心越低越稳定。当车模直立保持稳定时，重心和接触点的距离最小。相对于重心较高的车模，车模的动力系统只需要经过微调就能够保持直立。

4.2.1 姿态检测传感器安装

(1) 前文已经提到，MMA7260 加速度计对震动非常敏感，输出有许多噪声，所以应该尽量往下装，并装于车模的重心。

(2) 本组的陀螺仪有两个。一个用于车模直立姿态的检测，一个用于转向。用于直立姿态检测的必须安装水平，如果不安装水平的话，会导致车模在入弯时过快或过慢。车模有两种运动，平动和转动。如果陀螺仪没有水平安装的话，车模转动时，会在 Z 轴上产生一个分量，造成车模误判。^[2]而转向陀螺仪的中轴只需要与车模转轴保持一致即可。

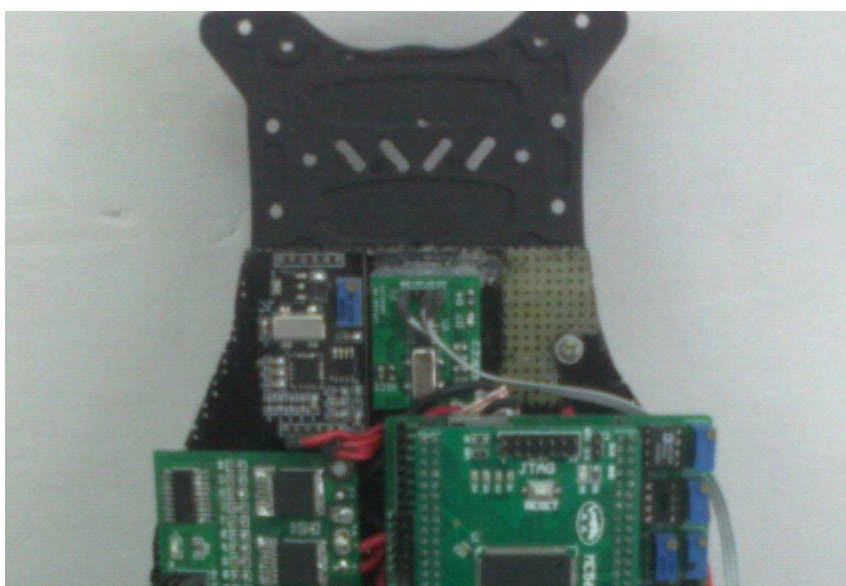


图 4.3 陀螺仪和加速度计的安装

4.2.2 电磁传感器安装

本系统仅仅采用了两个 10mH 的工字型电感。安装时如图所示向外平行放置。这样放置的原因是能够基本忽略十字弯，这对算法的简洁是一个极大地帮助。本次比赛并没有对车模的循迹模块长度进行限制。是不是长度越长越好呢？如果在电感仅有两个的情况下，增加杆长可以藐视一些弯道，尤其是小 S 弯。而且在没有改动算法的基础上就能够保持车模内切。这主要是因为，当杆子较长时，车模会提前知道自己要入弯，然后在还没有入弯时就会给电机差速，准备转弯，这样看起来车模就内切了。这是一种通过机械结构使路径优化的讨巧方法。

但是杆子并不是越长越好，杆子长了之后，车模在转弯时，循迹杆会完全伸出到跑道外，采集到的电感值就为零。有一种思路是当检测不到值时，左右两个电机分别保持前一个状态不变。车模杆子出了弯道一定是因为有些地方需要比较大的转弯角度，因此在转向时就给一个很大的差速，然后后面的弯道，车模保持前一状态就行了。另一种方法是，如果两边的电感应该保持对称用于循迹，那么应该多出一个电感来检测是不是出赛道了。可以使用奇数个电感。在赛场中，确实看到了有些队伍采用了奇数个电感。对于杆子不长的队伍，奇数个电感是无意义的。

4.2.3 编码器安装

官网方案在速度传感器上给出了一种粗糙的解决方式，采用光电码盘以及与之相对应的光电对管。这会使安装变得复杂。电机在高速运行中会产生两个影响：1、会产生高热，高热对安装的胶水提出了很高的要求，没有办法保证光电码盘不会因为时间久了而倾斜；2、会产生灰尘。灰尘极易卡在码盘内，肉眼

无法察觉，这会导致采到的脉冲值是不稳的。还有一个问题是，光电码盘在单位时间内给出的脉冲值太少。在后面的软件设计中，需要经常对速度控制周期做调整。如果一个速度控制周期内无法采集够速度脉冲，势必造成采样不精确。

【8】

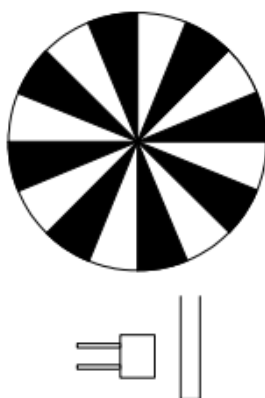


图 4.5 光电码盘

电磁组比赛采用结构紧凑的 C 型车模，经过论证，采用了 OME-NΦ18 编码器。该编码器为 157 线编码器，小巧，输出电压为正弦电压，配以转换电路可以将输出脉冲变为方波脉冲，直接送入单片机进行脉冲捕捉。通过支架和热熔胶，将其固定在轮的两侧上。如图所示，这样可以在节省空间，另外精度也有保证。

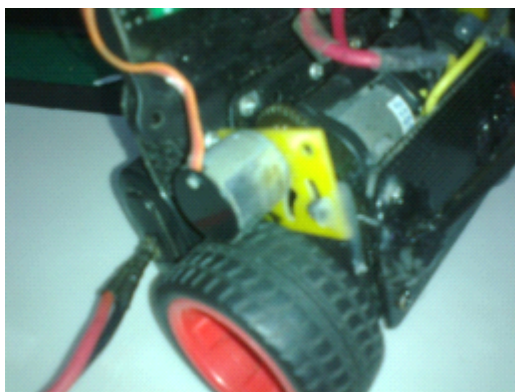


图 4.6 OME-NΦ18 编码器安装图

4.2.4 电池安装

三个组别的电池安装其实都应该好好研究。电池是全车中单重最大的物件。电池位置处理不当，对车模的转弯性能影响非常大。尤其对于直立的电磁组，电池安装位置合适可以降低 PID 参数整定的难度。下面将首先分析官网参考方案，然后提出两种方案。

官网参考方案中，将电池架设在车模中下部位。这样好像是降低了车模的重心，但事实上是不够的。重心离动力太远了，车模光是在调整直立时动力每次就需要很大的调整。于是有如下两种方案。

(1) 电池横向安装在后方与车轮保持基本水平的地方。这种安装已经使重心降低达到了极限，几乎与车轮轴线水平，重心离动力大约三到四厘米。这样的电池安装，车模直立较为容易，参数调试简单。而且车模在运行时会自然前倾，这样可以获得较高的速度。但是电池依然是横向放置的，车模在转向时，还是会有很大的转动惯量，不利于高速进入弯道。这种放置方式考虑重心下移多一点。这样后面的直立控制和速度控制 PID 比较好调。

(2) 电池垂直安装在车模背部中轴。之所以垂直安装是考虑车模转向的转动惯量。安装在中轴上使得转弯变得简单，车模方向控制好调一写，车模过冲现象不会很明显。

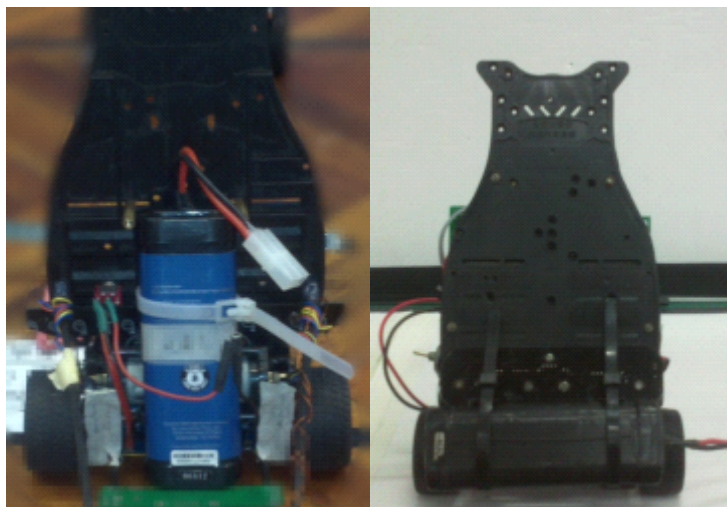


图 4.7 电池的装法

第五章 电路设计

5.1 核心底板

5.1.1 整体电路

直立车模对系统的稳定性提出了极高的要求。因此要尽可能地减少手焊和万用板的使用。所以我们用 Altium Designer 自己绘制了电路板。（如下图）

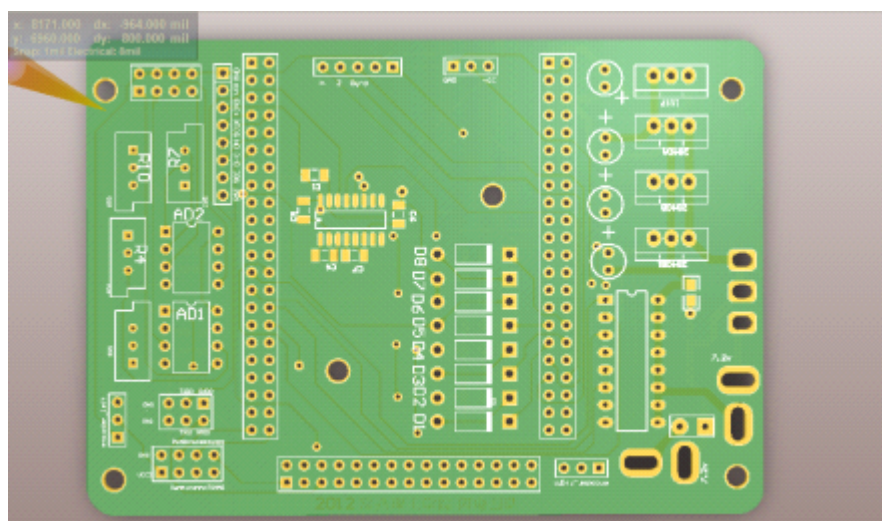


图 5.1 核心底板

核心底板除了驱动模块，几乎包括的所有的部分。主板的集成使得系统拥有足够的稳定性。而且，降低了重量，减少了焊接，减少了对于杜邦线的使用。

有些学校将单片机的最小系统直接画在 PCB 底板，由于单片机引脚且多且密，焊接失误在所难免。而且单片机属于高精的物件，不能够长时间承受焊接的高温，所以我们购买了单片机的核心底板。根据核心底板的尺寸做了上面的 PCB。

这样看来，整个板子简约、可靠而美观。

5.1.2 电源电路的设计

系统采用了统一的 7.2V、2000mA 可充电镍镉电池供电。由于系统的各个部分工作的电压不同，所以需要将充电电池电压转换为各个模块的所需的电压。具体的电压分配如下：

- 1、3.3V 给 MC56F8366 最小系统和 LCD5110 供电
- 2、5V 给编码器、MAX232、电感供电、驱动电路的逻辑输入电压
- 3、电源电压直接给电机（H 桥）供电。

MC56F8366 拥有极为丰富的片内资源。其两路输入脉冲捕捉口，极大地方便了电磁车对于两个轮子上编码器的计数。但是 MC56F8366 是 3.3V 供电，工作电流 200mA，计算得到的功耗为 660mW。在对单片机和 LCD5110 的供电上，我们采用了 LM1117，LCD5110 工作电流 200uA，其功耗可以忽略不计。而 8366 的功耗就较大了。MC56F8366 用 LM1117 稳下来供电，这样 LM1117 的功耗达到了 340mW，发热量会较大，对电池的损耗也比较严重。但是 LM1117 是 800mA 低压差线形稳压芯片，而且智能车是短距离高速度比赛，短时间的发热是可以接受的。实际证明，电池的损耗确实比较严重，在电压低于 7.8V 时系统运行地不够流畅。（大赛提供的电池充满电后可以达到 8.2V 左右）但是大体还是适合的。时间紧凑，本届比赛未能作出更改也是遗憾。



图 5.2 LM1117 稳压芯片

5.2 驱动模块

C 型车模的电机功率较小，转速不够快，功率不够大，加速不明显。经实验

测得，空载给电机 8.3V 电压时，其工作电流仅为 0.17A 左右。电机的性能直接决定了加减速性能，直接决定了驱动电路的参数性能。虽然说电机的性能不行，但是在飞思卡尔的赛场上，比的就是速度，所以我们还是采用了 BTS7970B，虽然它对电机性能的提高无明显优势，但我们认为 BTS7970B 可以完全发挥这个差电机的极限。

BTS7970B 拥有极强的电机驱动能力，最大驱动电流 68A，内部电阻最大 15 毫欧。BTS7970B 是半桥驱动芯片，两片芯片构成了 H 桥，这样就可以用来控制一个电机。C 型车模有两个电机，这样就需要使用 4 片 BTS7970。使用 7970 还有一个好处是 7970 的内阻很小，仅有 15 毫欧，那么它的发热量就就比较小，在大电流情况下利于这个电路的稳定。^[9]

考虑到 BTS7970B 的电流巨大，所以直接用单片机来控制并不是特别严谨，基于此，本组采用了 74LS244 锁存器。防止反向电流过冲进入 DSC，导致 DSC 烧坏。所以用 74LS244 缓冲保护。

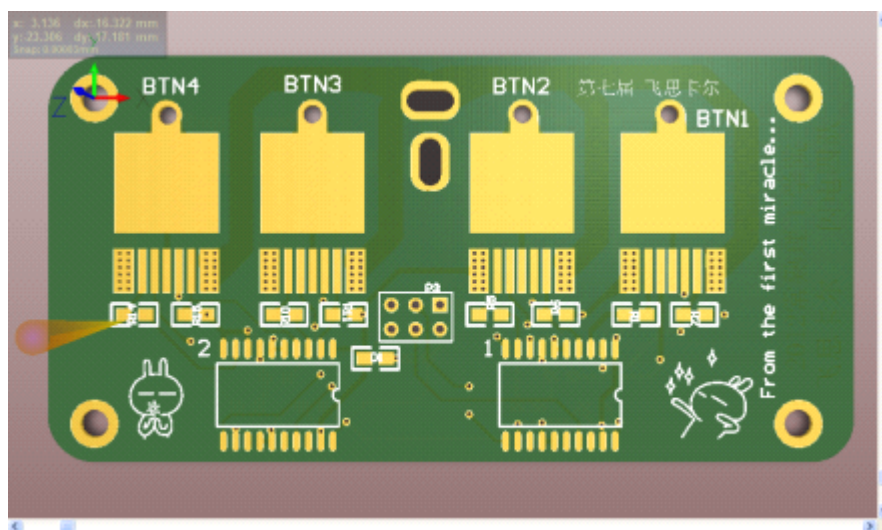


图 5.3 驱动电路

5.3 磁导航模块

本组最终采用了两只电感的方案。电感原先的电路图参考了官网方案。但

是我们发现，官网方案 AD 采集后的值并不令人满意。另外，官网方案由于直接拿三极管搭建，没有差分式放大电路，所以环境对 AD 采值的影响会非常大。所以我们自己用运放搭建了放大电路。如图 5.4

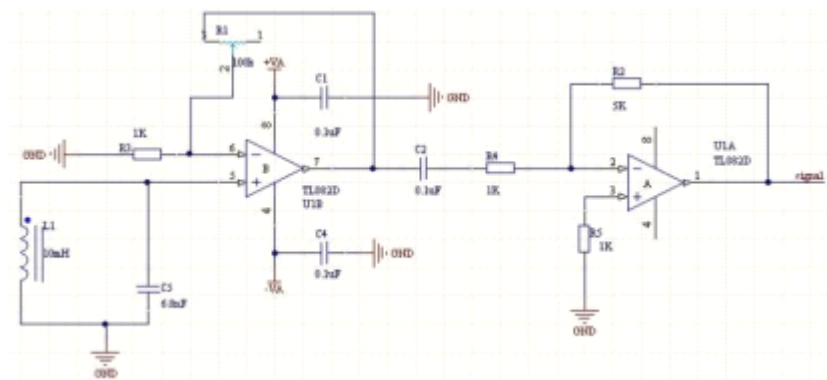


图 5.4 信号放大电路

第六章 软件设计

6.1 DSC 片内资源及其配置

DSC 具有非常丰富的片内资源。根据 freescale 半导体公司关于 DSC 的 datasheet: DSC 具有非常快的处理速度，在 60MHz 的总线频率下，每秒能运算六千万条指令；DSC 很好地将 MCU 和 DSP 联合在一个高效的、统一的架构中；它具有 1MB 的程序和数据存储空间；512KB 的程序 flash，4KB 的程序 RAM，32KB 的数据 RAM, 32KB 的数据 flash, 32KBd 的启动 flash, 最高两组六通道 PWM 脉宽调制模块，4 组 4 通道 12 位模数转换器。可以很惊奇的发现，DSC 还具有温度传感器，最多两个正交解码器，最多两个 FlexCAN 模块，两个异步串行通信接口，最多两个外围串行接口，最多 4 路通用正交计时器，看门狗，JTAG 实时调试接口，62 个通用 I/O 口。^[10]

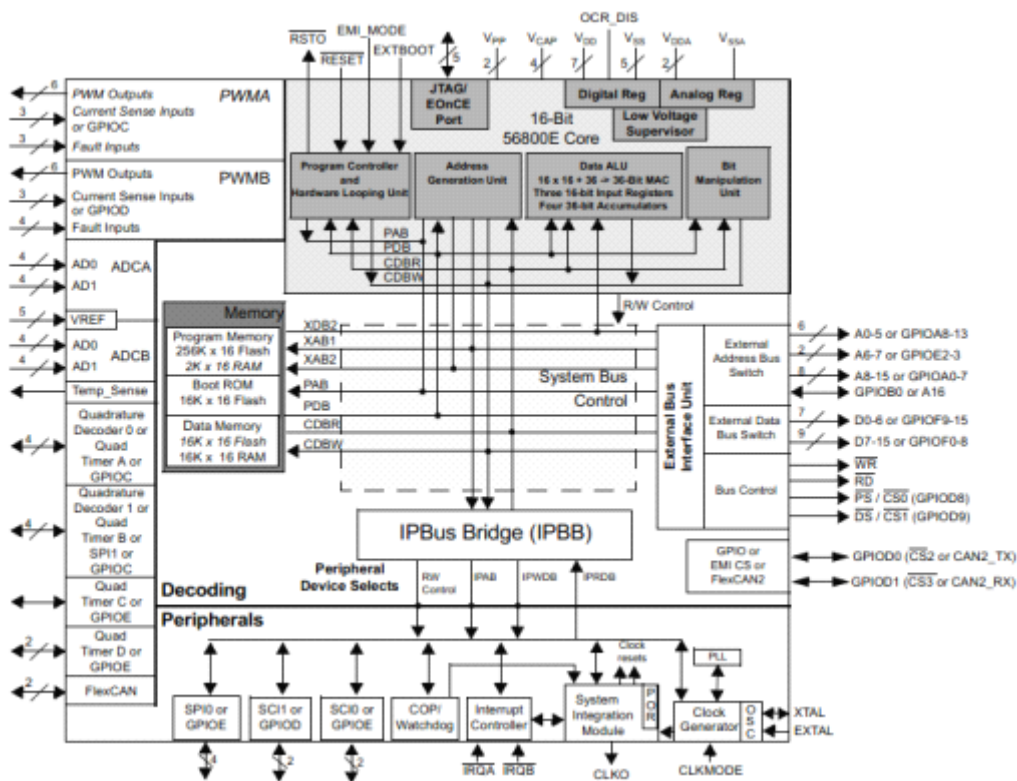


图 6.1 MC56F8366 片内资源

一方面，DSC 架构尤其适合类似于下述的典型控制操作：1、定期提供中断服务。2、从多个传感器和控制输入捕捉数据。3、向执行机构发送数据和控制脉冲，例如，发送占空比可变的 PWM 信号以合适的周期开关燃油喷射器或点火电路。4、与分布式系统中的其他控制器模块共享数据，例如，各种子系统周期性地发送状态数据到诊断模块或用户显示面板。另一方面，DSC 的 CPU 支持功能强大的一套 DSP 指令和灵活的寻址模式，因此能快速完成一系列精确的算术与逻辑运算。大多数信号处理算法以及许多一般数学计算，都包含有动态"乘积和"的计算。诸如 MAC(乘-累加)等特殊指令能够在一个指令周期内，求得两个 16 位数的乘积，将结果添加到累加器，然后从 RAM 预取一对数据值。因为有两个累加器，这种架构还能在回写数据到一个累加器的同时在另一个累加器中执行计算。这也就是说，它在运算浮点类型的数据时，执行的速度将会非常快。在电磁组的参数中，大多数都是浮点运算，那么 DSC 就非常适合这一类的运算

了。

相比较于大赛组委会推荐的 MC9S12XS128 单片机，DSC 具有明显的优势。首先，在单片机处理速度上，DSC 就已经略胜一筹。DSC 的总线处理速度为 60MHz，而 S128 单片机最高总线处理速度为 40MHz。而且刚才也已经叙述了，由于其结合了 DSP 的内核，这使得浮点运算速度极快，适用于计算。

电磁组使用 C 型车模，后轮采用两个独立的电机，为了对它们分别进行精确的速度控制，必然需要使用两个旋转编码器。然而，S12 只有一路 16 位硬件计数器，不能满足需求。采用中断进行软件计数会消耗大约 10% 的 CPU 资源，我们学校另外一组队伍采用的就是中断；采用外部计数电路会导致电路膨胀；采用时分复用计数端口的思路则会牺牲速度测量的准确性。

但是，选择 DSC 也有一些缺点。1、相较于资料满天飞的 S128 单片机，DSC 除了 datasheet 和往届其它学校的技术报告几乎就没有什么资料了。开发难度在刚开始的时候较大；2、DSC 在定义全局的浮点类型变量时，是不能够直接赋值的，一旦赋值将会出错。本小组就在此停顿了至少两天。

综合评估 DSC 的优缺点，可以认为选择 DSC 来开发是严谨的。

6.2 DSC 底层配置

利用 CodeWarrior 8.3 的 Processor Expert 系统来配置资源可以是开发过程简便高效。Processor Expert 是集成在 CodeWarrior 中的辅助开发系统。这一套系统使得复杂单片机的开发变得简单、高效、有趣。直接以代码方式配置 DSC 的资源，如果在没有完全了解 DSC 寄存器作用的时候，写出的代码是没有保证的。PE 基本不存在这样的问题。

下图是配置的资源：

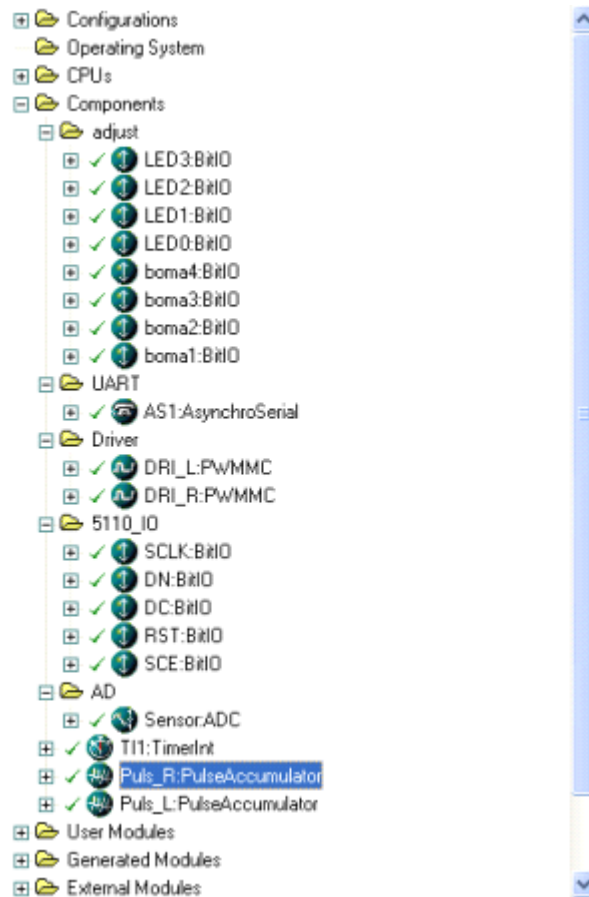


图 6.2 Processor Expert 来配置 DSC 的资源

上图包含了对 DSC 的一些底层基本配置。包括一个定时器，两个脉冲计数，两个 PWM 通道，5 个 AD 采集通道，以及若干个通用 I/O 口。^[11]

6.3 直立控制算法

直立控制是成败的关键。

首先是要将采集到的各种姿态数据进行处理。整个程序在运行时，一直处于死循环中。1ms 定时器中断服务程序主要是对各种传感器的数据进行更新。这包括：两个陀螺仪、一个加速度计、两个电感的 AD 值。将每次进入中断采集到的 AD 值分别减去各自静态值后相加，并且每 6 个中断就求一次平均，实质上就是进行均值滤波。6 不是固定的，是需要实验观测的，多长时间进行一次滤波要

看具体看每辆车。官网给出的方案说，零偏值总是存在一定的误差，这是可以通过后面的速度控制弥补的。这是一种很不严谨的说法。零偏值存在的误差必须通过直立控制调试解决，不然速度是上不去的。其实，解决这个问题并不是很难，只需要在直立函数中加入一个角度的修正就行了。

需要注意的是：将加速度计的值归一到-90度~90度有两种方案。一种方法是乘上 180°。但是这会出现一个很有趣的现象。利用这种方法调出来的车模，确实能够保持直立，但是鲁棒性极差。先将车模直立，然后用手在 -5°~5° 之内晃动，停止晃动后，发现车模立刻就倒下了。如果用反三角函数归一就不一样了。同样的实验，车模并不会倒下，依然能够保持直立。

然后，通过卡尔曼滤波器将加速度计和陀螺仪的值进一步优化，得到一个整合的数据送到直立函数中。这里还有一个关键之处是修正的值对实际值的追踪。这需要借助虚拟示波器来做。

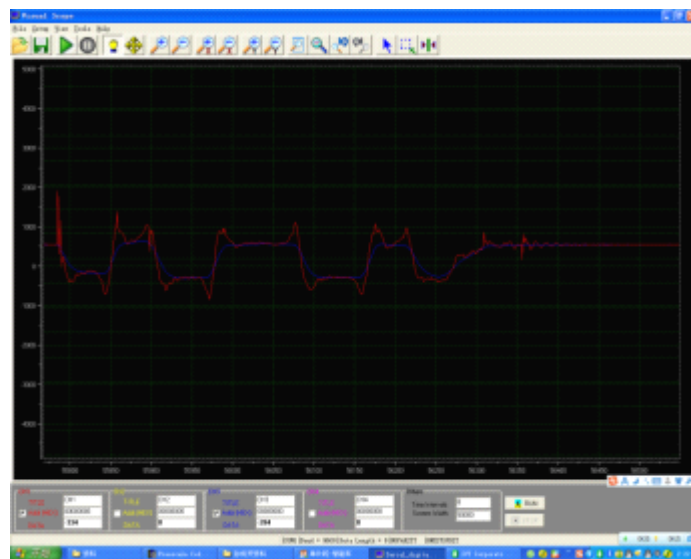


图 6.3 虚拟示波器波形图

在直立函数中主要有两个部分：1、PID 控制；2、bang-bang 控制。前文已经提到两轮自平衡车是一个多变量、严重不稳定、强耦合、时变、具有参数不确定性的非线性的系统。其实，在非线性、时变的系统时用 PID 来控制是比较困难的，但在不了解一种复杂算法而恣意使用是不严谨的。鉴于本系统所要达



到的功能有限，可以认为，PID 控制已经够了。实际证明，传统 PID 是够用的。通过 PID 控制器，就可以得到直立控制部分给电机的 PWM 控制值。直立控制中，三个参数对系统影响都较大。比例参数的调整能够使车基本直立，本组就单靠比例参数完成了直立，但是系统还不够稳定。通过微分参数，可以适当削弱系统的振荡，但是微分过大会带来震动。积分参数可以将车模静止，进一步平滑系统，但是过大会使系统晃动。当三个参数的整定都达到最优时，车模就能保持直立了。

6.4 速度控制算法

首先依然是采集速度信息。速度信息主要靠两个轮子上的编码器采集。速度采集和控制的时间可以与直立控制不一致。可以适当比直立慢一些。比如说，每 8 个中断进行一次速度控制。这也预示着，直立控制是优先的。将左右采到的两个值求差，求差之后的值减去你所需要的速度量。再经过位置式 PID 控制器，得出速度控制 PWM 量，并以一个正值另外一个负值分别加到左右两个电机上。

速度控制中，影响最大的是积分参数，积分参数非常影响车速。稍稍增加积分参数就会使车模速度整体变快。

6.5 方向控制算法

采集两个电感值，将它们的差除以它们的和，然后送入 PD 控制器。发现比例参数可以使车模很快转向，但是很容易引起车模的过冲。在没有添加转向陀螺仪微分参数之前，本组也试着整定微分参数，但发现这是徒劳的。于是，增加了陀螺仪微分参数，车模转向明显平滑了。

由于两只电感的限制，即使循迹模块再长也不能判断弯道，用算法也无法弥补。限于时间，本组没能提出更好的方案，这是一个缺憾。

第七章 车模开发工具及调试过程

7.1 开发环境

7.1.1 Codewarrior IDE 功能介绍

本组采用了 Codewarrior IDE 为开发环境来开发程序。Codewarrior 是由 Metrowerks 公司提供的专门面向 Freescale 所有 MCU 与 DSP 嵌入式应用开发的软件工具。其中包括集成开发环境 IDE、处理器专家、全芯片仿真、可视化参数显示工具、项目工程管理、C 交叉编译器、汇编器、链接器以及调试器。

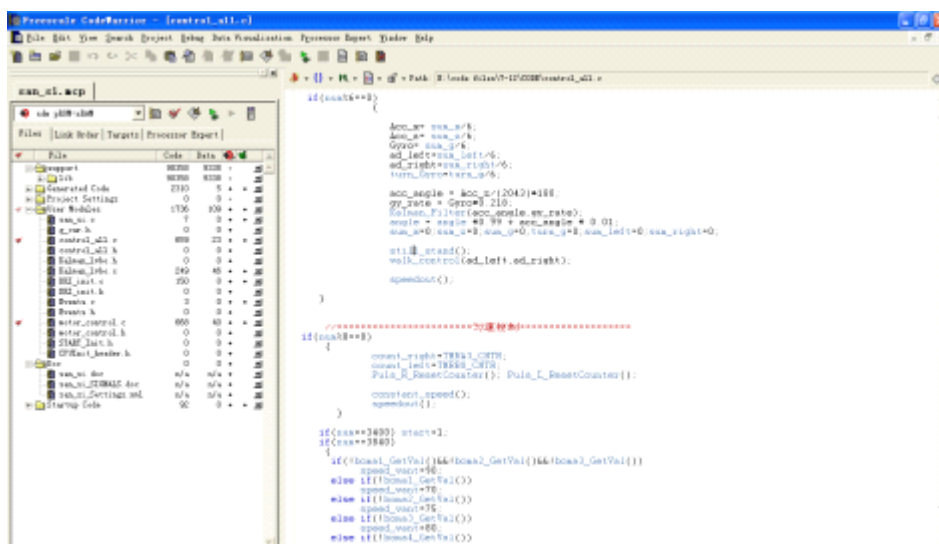


图 7.1 Codewarrior IDE 集成开发环境

7.1.2 ProcessorExpert 处理器专家

本组采用的是开发难度较大的 DSC 单片机。相对于 128 单片机，其底层资料网上较少。而且 DSC 是一种特殊的单片机，它是 MCU 和 DSP 的混合体。为了减少开发周期，把更多的精力投入到后期软件算法调试的过程中。我们采用 Processor Expert。

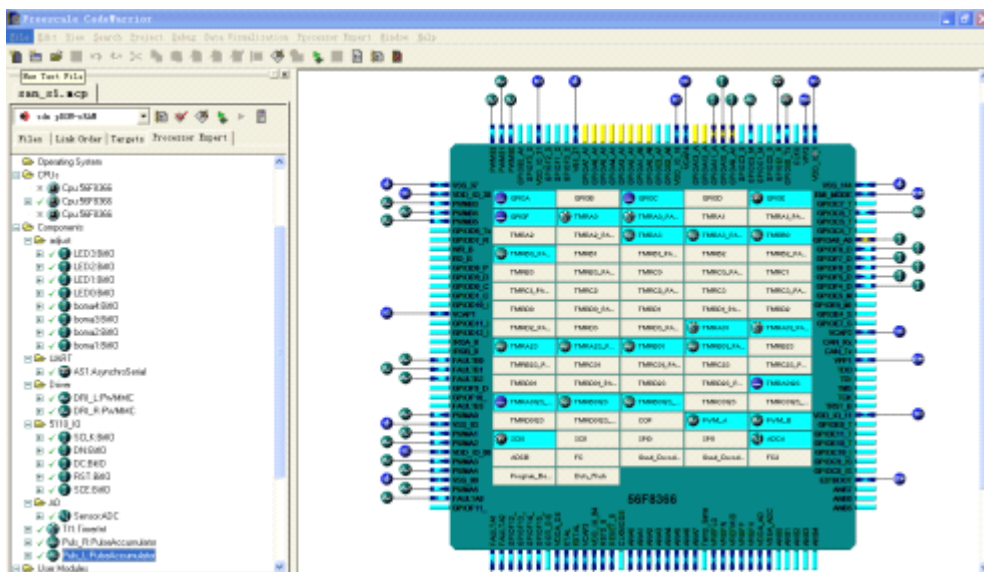


图 7.2 PE 使底层配置变得简单

ProcessorExpert 使开发 DSC 变的简单。DSC 所有的引脚定义，资源使用情况和底层函数的生成，都能够在上图的界面中体现。DSC 不同寻常的初始化代码也能够其中配置配置完成。^[12]

7.2 蓝牙模块

在初期调参的时候，本组调参直接上计算机，调一次，下载一次程序。这样做非常耗时间。而且 DSC 在烧写程序时发热相当严重，有一次连夜调车，烧写单片机过猛，直接将单片机擦坏了。主流的方案是在车体上安装一个显示屏配合按键调参。但是这样会使电路庞大，并且增加了多余的器件，摘取也不是很方便。所以我们选择了蓝牙模块结合 SPI 通信来调。



图 7.1 蓝牙模块

7.3 虚拟示波器

和大多数队伍一样，在处理参数时，需要借助虚拟示波器，将采集到的陀螺仪和加速度计的信息以及处理后的信息进行比对。看波形，调参数这是基础的方法。

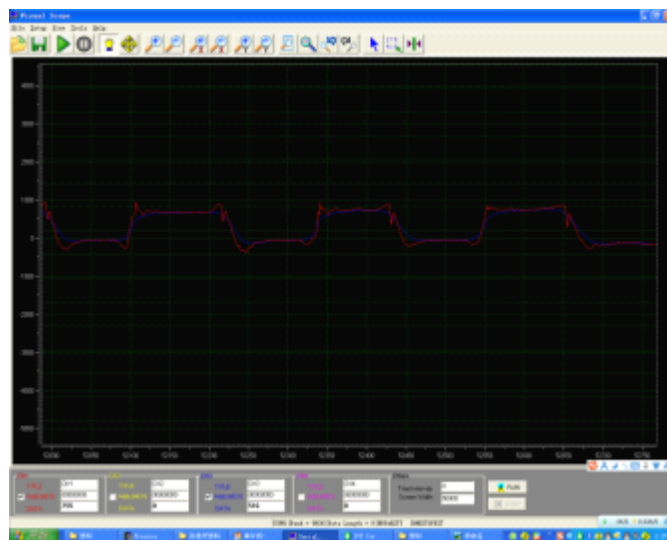


图 7.2 虚拟示波器

第八章 总结

本报告详细叙述了第七届“飞思卡尔杯”全国大学生智能汽车竞赛我校闪电四队两轮自平衡智能车制作情况。我们总结出了以下几点。

1、机械结构和电路设计的思考。机械结构的设计必须简洁、优雅、美观。稳定与速度是一种矛盾，速度上去了，稳定跟不上，稳定提高了速度有可能降下来。但是稳定是速度的前提与必要保证。在机械结构的设计上，应该尽量减少杜邦线和飞线等一切不稳定物件使用。另外，车模传感器的布局 and 安装也至关重要，安装在何处都要有周密的思考，详细的讨论。

2、软件设计思考。软件是实现小车整体运行的关键，是我们控制小车按照我们思路跑的重要载体。有的时候，软件设计可以弥补硬件设计的缺陷。比如说，两个电机不对称的问题。我们发现，小车直立控制很好，但是在匀速运行时，总是跑歪。我们查明了是电机或者驱动不对称的原因。修改硬件是危险的，因为一些参数有赖于硬件。所以针对某一个电机，我们给予了适当的补偿。这样虽然不能使车走得非常直，但也不至于打圈。

不做小车不知道，一做小车才发现，原来自己进步的空间还那么大。通过几个月来与小车的相伴，我们学会了如何用 DXP 制版，如何使用驱动，如何优化程序，如何上网搜资料，搭硬件。我们在很短的时间内，学会了一块新的单片机，我们在很短的时间内摸索出了磁导航两轮自平衡小车的制作方法和要求达到的工艺。智能小车还教会了我们如何团队合作，如何协作共赢，如何耐下性子在很细很细的问题上摸索。



参考文献

- 【1】王效杰.基于变结构控制的两轮自平衡小车系统设计与实现.学位论文.西安.西安电子科技大学学位论文.2006
- 【2】《电磁组参考设计方案第二版》，智能汽车竞赛秘书处，2012-3-1
- 【3】刘金琨.先进 PID 控制及其 MATLAB 仿真【M】.北京.电子工业出版社.2003
- 【4】王晓宇,闫继宏,臧希喆等.两轮白平衡机器人多传感器数据融合方法研究[J].传感器技术学报. 2007
- 【5】赵杰,王晓宇,秦勇等.基于 UKF 的两轮自平衡机器人姿态最优估计研究[J].机器人. 2006)
- 【6】耿延睿,崔中兴.组合导航系统卡尔曼滤波衰减因子自适应估计算法研究[J].中国惯性技术学报.2001.
- 【7】秦永元,张洪钺,汪叔华.卡尔曼滤波与组合导航原理[M].西安:西北工业大学出版社, 1998.
- 【8】第六届西北工业大学技术报告湛庐队.2011
- 【9】BTS 7970B.High Current PN Half Bridge. © Infineon Technologies AG .2006
- 【10】MC56F8366 Technical Data, Rev. 2.0,Freescale Semiconductor, Inc. 2005
- 【11】傅志强.CodeWarrior 集成开发环境使用简介.2005
- 【12】傅志强.CodeWarrior 集成开发环境使用简介.2005【13】第六届"飞思卡尔"杯全国大学生智能汽车竞赛技术报告

致谢

本次比赛我们组从五月份才开始准备，作为 10 届大二学生，我们获得了机会与学长一起奋斗在飞思卡尔的战线上。在这条战线上，我们深深感知不是我们一个队在战斗。

感谢学校教务处，电气与自动化工程学院的老师，是他们给我们提供了经费和良好的环境。

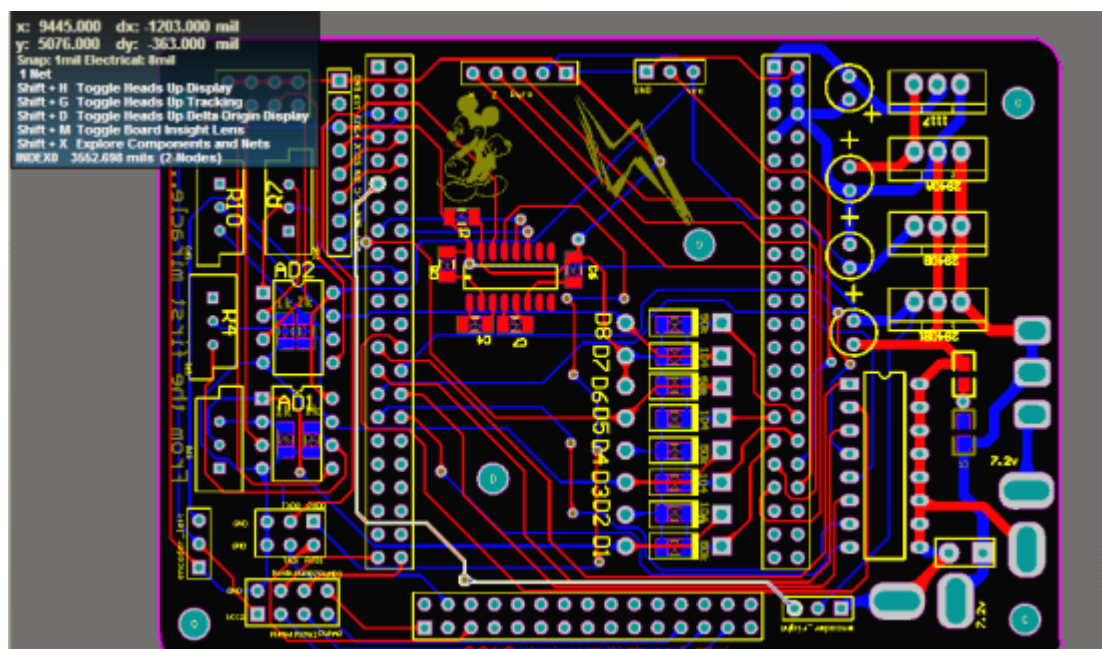
一定要感谢李鑫老师，是他的信任给了我们坚定不移的勇气。从进入实验室的那一刻，我觉得我们是幸运儿。大学飞奔在飞思卡尔赛场上的梦想就要完成。但是信任也是种压力，要我们转换为动力。做小车的路终究不是平坦的，需要我们跨过一个一个阶梯，通过一个一个瓶颈，每一次都会体验痛苦。

还要感谢张东林、孙传开、李超学长。张东林学长是实验室第一个做两轮自平衡小车的人，孙传开是第一个直立的人，他们为我们铺平了通往直立的道路。是他们的摸索才有了我们在速度上的提升。

最后要感谢组委会，正是这个比赛，让我们有机会和优秀的大学，优秀的对手一较高下。

附录

1、PCB 底板图



2、程序

```
//***** 上位机调试程序*****//
```

```
#include "CPUInit_header.h"
```

```
#include "START_Init.h"
```

```
float OutData[4] = { 0 };
```

错误！未定义书签。

```
unsigned short CRC_CHECK(unsigned char *Buf, unsigned char CRC_CNT)
{
    unsigned short CRC_Temp;
    unsigned char i,j;
    CRC_Temp = 0xffff;

    for (i=0;i<CRC_CNT; i++){
        CRC_Temp ^= Buf[i];
        for (j=0;j<8;j++) {
            if (CRC_Temp & 0x01)
                CRC_Temp = (CRC_Temp >>1 ) ^ 0xa001;
            else
                CRC_Temp = CRC_Temp >> 1;
        }
    }
    return(CRC_Temp);
}
```

```
void OutPut_Data(void)
{
    int temp[4] = {0};
    unsigned int temp1[4] = {0};
    unsigned char databuf[10] = {0};
    unsigned char i;
    unsigned short CRC16 = 0;
    for(i=0;i<4;i++)
```



```
{

temp[i]  = (int)OutData[i];
temp1[i] = (unsigned int)temp[i];

}

for(i=0;i<4;i++)
{
databuf[i*2]    = (unsigned char)(temp1[i]%256);
databuf[i*2+1] = (unsigned char)(temp1[i]/256);
}

CRC16 = CRC_CHECK(databuf,8);
databuf[8] = CRC16%256;
databuf[9] = CRC16/256;

for(i=0;i<10;i++)
    uart_putchar(databuf[i]);    //发送字符函数，此处可以换成你自己的字符
发送函数。
}

//*****卡尔曼滤波*****//
#include "CPUInit_header.h"
#include "START_Init.h"
float Q_angle=0.001, Q_gyro=0.003, R_angle=0.5, dt=0.01; //本程序此处为 10ms
//float Q_angle, Q_gyro, R_angle,dt; //本程序此处为 10ms
```

错误！未定义书签。

```
float  angle,angle_dot;
```

//在这个文件里只要改个采样时间即可，其他值最好不要改

//注意：dt 的取值为 kalman 滤波器采样时间;

```
float P[2][2] = {
                { 1, 0 },
                { 0, 1 }
                };
```

```
float Pdot[4] = {0,0,0,0};
```

```
float q_bias=0, angle_err=0, PCt_0=0, PCt_1=0, E=0, K_0=0, K_1=0, t_0=0, t_1=0;
```

```
const char C_0 = 1;
```

//这个地放方的卡尔曼滤波不知什么原因用的不是太好，希望这个以后能得到解决

```
//-----
```

```
void Kalman_Filter(float angle_m,float gyro_m)//angle_m: 为加速度测得角度。
```

gyro_m:陀螺仪的角速度

```
{//-----
```

```
Q_angle=0.001; Q_gyro=0.003; R_angle=0.5; dt=0.01;
```

//外部需要引用的变量：角

度，角速度

```
angle+=(gyro_m-q_bias) * dt;//先验估计
```




$\dot{P}[0] = Q_angle - P[0][1] - P[1][0]; // Pk'$ 先验估计误差协方差的微分

$\dot{P}[1] = -P[1][1];$

$\dot{P}[2] = -P[1][1];$

$\dot{P}[3] = Q_gyro;$

$P[0][0] += \dot{P}[0] * dt; // Pk$ 先验估计误差协方差微分的积分 = 先验估计误差协方差

$P[0][1] += \dot{P}[1] * dt;$

$P[1][0] += \dot{P}[2] * dt;$

$P[1][1] += \dot{P}[3] * dt;$

$angle_err = angle_m - angle; // zk$ -先验估计

$PCt_0 = C_0 * P[0][0]; //$

$PCt_1 = C_0 * P[1][0]; //$

$E = R_angle + C_0 * PCt_0; //$

$K_0 = PCt_0 / E; // Kk$

$K_1 = PCt_1 / E;$

$t_0 = PCt_0;$

错误！未定义书签。

```

t_1 = P[0][1]; //C_0 *

P[0][0] -= K_0 * t_0; //后验估计误差协方差
P[0][1] -= K_0 * t_1;
P[1][0] -= K_1 * t_0;
P[1][1] -= K_1 * t_1;

angle += K_0 * angle_err; //后验估计
q_bias += K_1 * angle_err; //后验估计
angle_dot = gyro_m - q_bias; //输出值（后验估计）的微分 = 角速度

}/**/
//*****pwm 输出*****//
void speedout(void)
{
    float pwm_left;
    float pwm_right;
    pwm_left = PWMOUT + turn_walk_left + pwm_apart; //pwm 输出的三个分
量，直立，巡线，实时速度
    pwm_right = PWMOUT + turn_walk_right + pwm_apart;

    pwm_left = adjust(pwm_left, ang);
    pwm_right = adjust(pwm_right, ang);

    if(pwm_left > 0) {b_left(pwm_left);}

```



```

else if(pwm_left<0) {f_left(-pwm_left);}

else {stop(); }

if(pwm_right>0) {b_right(pwm_right);}

else if(pwm_right<0) {f_right(-pwm_right);}

else {stop(); }/**/

}

//*****速度控制输出*****//

void constant_speed(void)

{

float KP,KI,KD;

speed_avg= (count_left+count_right)/2;

speed_p= speed_avg-speed_want;

speed_d=speed_old-speed_p;

speed_i=speed_i+speed_p;

speed_old=speed_p;

KP=0.05;KI=0.03;KD=0.02;

pwm_apart=-(KP*speed_p+speed_i *KI+speed_d *KD);

}

//*****方向控制输出*****//

void direction_out(void)

{

```

错误！未定义书签。

```
float turn_value;

turn_value=turn_walknew-turn_walkold;

turn_walk_left=turn_value*9/(9+1) + turn_walkold;

turn_walk_right=-(turn_value*9/(9+1) + turn_walkold);

if(turn_walk_left>-10 && turn_walk_left<10)  turn_walk_left=0;

if(turn_walk_right>-10 && turn_walk_right<10)  turn_walk_right=0;

if(turn_walk_left>turn_want_speed)

turn_walk_left=turn_want_speed;

else if(turn_walk_left<-turn_want_speed)

turn_walk_left=-turn_want_speed;

if(turn_walk_right>turn_want_speed)

turn_walk_right=turn_want_speed;

else if(turn_walk_right<-turn_want_speed)

turn_walk_right=-turn_want_speed
```