

第十届“飞思卡尔”杯全国大学生 智能汽车竞赛技术报告



学 校： 重庆大学

队伍名称： 不慢队

参赛队员： 李浩蜜

耿孝谨

刘 统

带队教师： 宋焱翼

甘 平

关于技术报告和研究论文使用授权的

说明

本人完全了解第十届“飞思卡尔”杯全国大学生智能汽车竞赛关保留、使用技术报告和研究论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会和飞思卡尔半导体公司可以在相关主页上收录并公开参赛作品的设计方案、技术报告以及参赛模型车的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名：李浩蜜

耿孝谨

刘统

带队教师签名：宋

焱翼

甘平

日期：2015 年 8 月 4 日

目录

摘 要.....	3
引 言	4
第一章 方案设计.....	5
1.1 制作情况概述.....	5
1.2 系统方案的选定.....	5
1.3 硬件电路设计方案.....	6
1.4 软件设计方案.....	7
1.5 小结.....	7
第二章 智能汽车机械结构调整.....	8
2.1 车模的机械结构.....	8
2.2 光电码盘的安装.....	9
2.3 线性 CCD 支架安装.....	10
2.4 加速度计与陀螺仪的安装.....	11
2.5 齿轮咬合与车胎处理.....	11
第三章 硬件系统设计与实现.....	12
3.1 硬件系统总体设计.....	12
3.1.1 可靠.....	12
3.1.3 简洁.....	12
3.1.3 美观.....	13
3.2 电源模块.....	13
3.3 电机驱动.....	14
3.4 线性 CCD.....	15
3.5 灯塔起跑模块.....	16
3.6 陀螺仪与加速度计.....	17
第四章 软件系统的设计及实现.....	19
4.1 软件系统总体设计.....	19
4.2 主要控制算法.....	20

4.2.1 直立控制.....	20
4.2.2 速度控制.....	21
4.2.3 转向控制.....	22
4.3 赛道元素识别与处理策略.....	23
4.3.1 CCD 光线自适应算法.....	23
4.3.2 坡道的识别.....	25
第五章 系统的开发环境与车模调试.....	29
5.1 开发环境.....	29
5.2 上位机软件.....	29
第六章 模型车主要参数.....	31
6.1 智能汽车外形参数.....	31
6.2 智能汽车技术参数.....	31
附录 1.....	32
附录 2	42

摘 要

本文主要介绍了直立循迹智能小车系统的原理、软硬件设计以及小车制作过程，对小车的系统介绍包括车模机械结构的调整、模块电路的设计、传感器信号的处理、控制算法以及整车调试的方法等。

设计以飞思卡尔公司的 32 位单片机 MKL26Z256VLL4 为控制核心，采用线性 CCD 采集赛道元素信息，光电码盘获取小车的速度，并利用加速度计 MMA8451 以及陀螺仪 MPU3050 进行角度融合，获取车模姿态，完成平衡直立。为了提高小车的稳定性，对传感器采集的数据进行滤波处理并且采用 PID 算法来实现小车的匀速行驶与转向。使用 IAR 集成编译环境编写代码，自行编写了上位机软件对小车进行调试并且采用 SD 卡记录数据、蓝牙回传数据和 OLED 液晶调参作为辅助调试手段。经过大量的软硬件测试，实现了基于线阵 CCD 的直立循迹智能小车。

关键词：MKL26Z256VLL4，PID，线阵 CCD，上位机

引 言

“飞思卡尔”杯全国大学生智能汽车竞赛是由教育部高等学校自动化专业教学指导委员会主办全国大学生智能汽车竞赛。该竞赛以“立足培养，重在参与，鼓励探索，追求卓越”为指导思想，旨在促进高等学校素质教育，培养大学生的综合知识运用能力、基本工程实践能力和创新意识。智能车竞赛涉及自动控制、模糊识别、传感技术、电子、电气、计算机、机械与汽车等多个学科，为大学生提供了一个充分展示想象力和创造力的舞台，吸引着越来越多来自不同专业的大学生参与其中，激发了大学生的创新思维，对于其实践、创新能力和团队精神的培养具有十分最重要的价值。该竞赛分竞速赛、创意赛和技术方案竞赛三类比赛。竞速赛是在规定的模型汽车平台上，使用飞思卡尔半导体公司的 8 位、16 位、32 位微控制器作为核心控制模块，通过增加道路传感器、电机驱动电路以及编写相应软件，制作一部能够自主识别道路模型汽车，按照规定路线行进，以完成时间最短者为优胜。创意赛在统一比赛平台上，充分发挥参赛队伍想象力，以特定任务为创意目标，完成研制作品，由竞赛专家组观摩作品现场展示、质疑、现场观众投票等环节，最终决定比赛名次。技术方案比赛是第八届新设定比赛组，针对参加全国总决赛的队伍出现的优秀创新方案进行评选，通过现场方案交流展示、专家答辩以及现场参赛队员的投票评出技术方案名次。技术方案组比赛目标是提高参赛队员创新能力，鼓励队员之间相互学习交流。

本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。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。本竞赛规则透明，评价标准客观，坚持公开、公平、公正的原则，保证竞赛向健康、普及，持续的方向发展。

第一章 方案设计

1.1 制作情况概述

根据第十届智能汽车竞赛的规则，本组选用比赛方提供的 E 型车模，采用飞思卡尔 32 位微控制器 MKL26Z256VLL4 作为核心控制单元，自主设计主控制板和其他相应的功能模块电路，综合线性 CCD 采集的赛道信息和加速度及、陀螺仪融合的姿态信息完成对小车的直立控制、速度控制以及转向控制，从而小车的自主循迹和对不同赛道环境的适应。

本组着重研究了两轮平衡小车直立原理，分析了车体机械结构对小车性能的影响，根据规则对车模进行了合理的改造，选择比较轻便、稳固的机械结构。深入理解 PID 参数在小车控制中所发挥的作用，通过不断调试优化各个参数，对弯道、直角、单线等赛道元素选择合适的处理策略，以增强小车的适应性与稳定性。

本文通过对小车系统的整体方案、机械结构、硬件设计、软件算法等方面的介绍，阐述了本组在智能车制作中的方法与思路。在将近一年的时间里，小组成员查阅了大量资料，学习了通信、自动化、计算机、机械等多个学科的知识，培养了我们电路设计、软件编程、系统调试等方面的能力，以系统的角度看待问题，锻炼了我们知识融合、实践动手并付诸创新的能力，对今后的学习工作都有着重大的实际意义。

1.2 系统方案的选定

为了满足比赛要求，需采用比赛规定范围内芯片，单一电源供电，实现平衡车，本组采用了如图所示的小车系统结构。线性 CCD 用于采集赛道信息，可以将传感器采集到的亮度信息转换为电压值；光电编码盘用于测量小车行进的距离，根据行进的时间转化为小车当前速度；MPU3050 是三轴陀螺仪，而 MMA8451 是加速度计，通过融合加速度与角速度可以控制小车直立。这些传感器采集的数据通

过 I2C 总线传输到 KL26 中进行处理后，微控制器将直立控制量、速度控制量以及转向控制量一并加在电机上，控制小车完成相应操作。单片机通过蓝牙与上位机进行串口通信，方便检查系统故障和调试参数，根据比赛规则要求，系统还设计有灯塔接收检测模块，用于发车以及停车，整个系统由竞赛提供的标准镍铬电池作为供电模块。

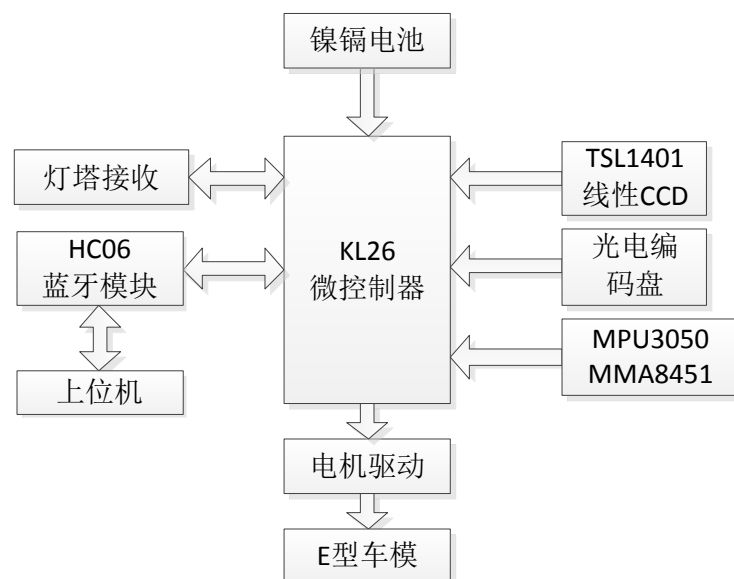


图1.1 系统结构框图

1.3 硬件电路设计方案

系统的硬件是整个系统的基础，是软件平台得以稳定运行的基础，所以硬件电路的设计是非常重要的。在硬件系统的设计过程中，本着系统安全、稳定的原则，电路设计尽量简单化、模块化。为了使整个电路设计的更加简化，考虑后将驱动电路与控制电路整合在一块板上，只要做好隔离，就不会造成信号串扰，这样减轻了复杂度，提高了稳定性。在主板之外还有灯塔模块和线性 CCD 模块和光电码盘，这些模块通过排线连接到主板预留的插槽上。下图为硬件电路设计结构框图。



图1.2 硬件设计框图

1.4 软件设计方案

智能车系统通过图线性 CCD 获取前方赛道的信息，同时通过速度传感器实时获取智能车的速度。程序检测灰度信息的跳变沿来检测黑线，计算智能车距离中线的偏差，从而获取准确的行驶偏移量，接着采用 PID 算法对电机的差速比进行反馈控制，在 PID 算法的基础上，加入模糊控制，有利于对智能车的非线性特性的控制。最终智能车根据检测到的速度，结合线性 CCD 获取到的偏差量，采用电机差速比闭环控制、陀螺仪加速度及对电机直立的闭环控制、以及编码器对电机转速的闭环控制，达到对智能车的三个单独的闭环控制不断进行恰当的控制调整，使智能车在符合比赛规则情况下根据赛道信息比赛。

1.5 小结

本章重点分析了智能汽车系统总体方案的选择，并介绍了系统的总体设、总体结构，以及简要地分析了系统各模块的作用。在今后的章节中，将对整个系统的其他模块进行详细介绍。

第二章 智能汽车机械结构调整

车模的机械结构在某种程度上限制着小车的最高速度，在速度较低的情况下各种机械结构可能没有太大差别，当速度逐渐加快时，相同的控制算法在不同的机械结构上就会有不同的表现。对模型车的机械结构有一个全面清晰的认识尤为重要，先建立相应的数学模型，再针对具体的设计方案来调整赛车的机械结构，并在实际的调试过程中不断的改进优化和提高结构的稳定性。本章内容为 E 型车模的结构设计，电路板和各个传感器的安装。

2.1 车模的机械结构

本组选用 E 型车模，其长度为 255mm，宽度为 155mm。相比之前的 D 车模，E 车轮距更大，在转向方面更有优势，且选用的电机型号为 RS-380，功率更大，没有了减速齿轮组，齿轮比更小，车轮转速更快，车速提升空间更大。



图 2.1 E 型车

对于平衡车来说，整个车子的重心高低十分重要。过高的重心会使车子在转向的时候容易侧翻、抬轮子，容易产生系统的不稳定；对于速度控制，很小的倾角就会产生较大的加速度，不利于精确地控制速度。但是车子的重心也不能过低，

首先过低的重心会使车子在过坡时车底碰到坡道，其次过低的重心不利于车子迅速加速。

车模上的两个电机位置不能改变，可以改变位置的结构只有电池和主板的位置，CCD 支撑杆的长度不仅影响着前瞻，也对车体的平衡有很大影响。小组在经过多次讨论与尝试后，采用了下图所示的机械结构，电池和电路板置于车体之前，使车子前倾时可以更快的达到设定速度；CCD 支撑杆长约 30cm，正常行进时前倾 20 度左右；截取车模尾部，这样做的好处是时车子转弯时更加灵活，较小的控制量就可以完成转弯，可以平稳匀速的过弯，减轻跳轮的问题，但出现的问题是稳定性变差，尤其是在以较快速度过坡道时，由于车体过于轻巧，车子会脱离赛道，在空中的小车没有着力点而失控，为此在小车底部添加配重来实现坡道的稳定。



图 2.2 小车机械

2.2 光电码盘的安装

小车系统需要外部的速度反馈来完成控制，本组没有采用个头较大的编码器，而是采用圆片状的光栅编码器，这样的优点是可以让车体更加轻便，无需外加齿

轮咬合，动力损失小，同时测量精度能够满足该设计；缺点是不易安装，而且容易损坏和进灰。光电码盘的原理与其他测速常用的编码器相同，也是将检测到的遮光次数转化为脉冲信号从而实现速度检测。我们采用的是 WTK-20 型光电编码器作为速度检测模块。并做了适当的改装和调节使得其便于安装、拆卸和维护



图 2.3 光电码盘

2.3 线性 CCD 支架安装

为了尽量减轻伸出车体外支架的重量，本组选用碳纤维杆作为支架材料，用尼龙夹将 CCD 固定在支架上，底部通过与支架垂直的碳纤维杆加固支架，防止小车运行中支架的晃动对传感器采集造成影响。搭建这样的结构可以用很轻的质量获得较好的稳定度，支架的刚度也有所保证，便于拆卸与维修。CCD 的俯仰角度影响小车的前瞻，角度低会使前瞻变小，不利于对赛道信息的预判，相反则会使前瞻变大，但是也会带来误判等问题。



图 2.4 CCD 的安装

2.4 加速度计与陀螺仪的安装

为了最大程度减少车模运行时振动对于测量倾角的干扰，将转向陀螺仪和加速度传感器放在同一块电路板上，并且将此模块安装在车体上而非主板上，这样传感器获得的数据才是车子自身的倾角信息，而主板会由于震动引入噪声。整个模块安装应注意与车体运行时处于与水平垂直的位置，安装的位置一定要正，尽量靠近中间，否则会使小车在转向时造成左右转向不均。

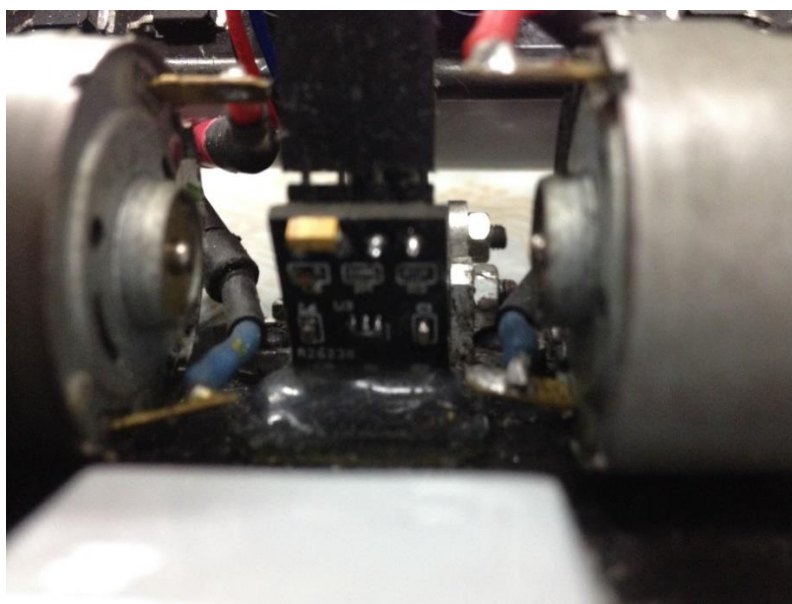


图 2.5 陀螺仪和加速度计

2.5 齿轮咬合与车胎处理

齿轮咬合要求传动流畅又不易打齿，使车子在行进时不会发出很大噪声，否则会加大传感器的噪声，不利于小车的控制。仔细调整塑料齿轮和铜齿的间隙，经常涂用润滑油，及时清理齿轮间的异物，使得控制和提速更容易。在速度较快的情况下，车胎的摩擦力是限制智能车速度提升的一个因素，对车胎进行适当的打磨可以增强车胎的摩擦力，并且经常擦拭，小车过弯时不易打滑，但过大的摩擦力也不利于车子流畅的转弯。

第三章 硬件系统设计与实现

3.1 硬件系统总体设计

可靠、简洁、美观是我们对硬件电路的设计方案既定的系统设计目标。

3.1.1 可靠

可靠是系统设计的第一要求，我们对电路设计的所有环节都进行了如下的改进和设计：

电源系统的稳压性能、纹波噪声的去处、功率的充足提供的可靠性等；主控系统的电路优化，减少电路元件的体积和用量，减少不稳定因素；动力系统的动力性能可靠性，如加速、制动等性能会有大幅度的改善，为整车的性能提升提供充足的保证；做好各部分的接地、屏蔽、滤波等工作，将高速数字电路与模拟电路分开，使本系统工作的可靠性达到了设计要求；传感器的性能稳定，保证整车数据流的准确可靠；

3.1.3 简洁

为了尽量减轻整车重量，降低模型车的重心位置，应使电路设计尽量简洁，尽量减少元器件使用数量，缩小电路板面积，使电路部分重量轻，易于安装。我们在对智能车系统进行分析后可以通过以下几点来进行对系统简洁处理：

智能车机械结构的简化，这需要我们设计出更加精巧、灵活的机械构件，提高智能车的精度，提高智能车的制作工艺，简化智能车的机械复杂程度。在对智能车硬件电路系统进行了详细分析后，通过认真筛选所用器件、简化电路、合理设计元件排列、电路走线，最终的硬件电路板采用贴片封装形式，使本系统硬件电路部分轻量化和机械强度部分的指标达到设计要求。

3.1.3 美观

一个好的设计应该是能给人以美感的。除了满足基本的设计要求之外，我们还对元器件的摆放位置做了大量的尝试，使得整个电路较为美观

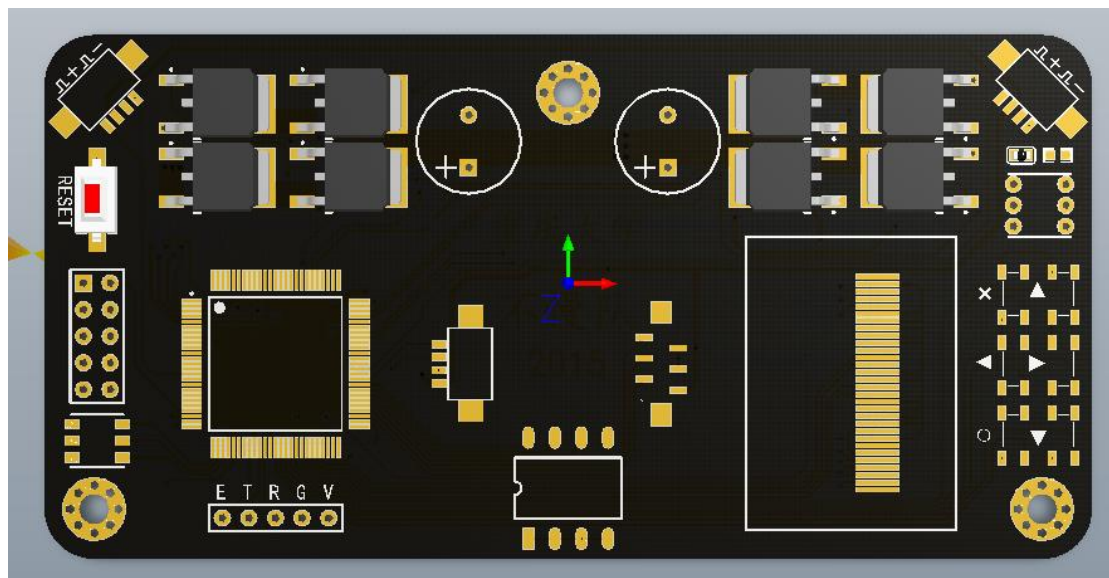


图 3.1 PCB 板

3.2 电源模块

整车的硬件电路电源由可充电镍镉电池提供（7.2V、2000mAh）。但是系统中的各个电路模块所需要的工作电压和工作电流各不相同，所以设计了多种稳压电路，将电池电压转换成各个模块所需要的电压。电机驱动电路需要 12V 电压，单片机、SD 卡、OLED 以及陀螺仪模块需要 3.3V 电压，光电码盘、CCD 模块和蓝牙串口模块需要 5V 供电。12V 电压由 MC34063 升压后得到的，5V 由 TPS7350 得到，再由 TPS7333 得到到 3.3V 电压。

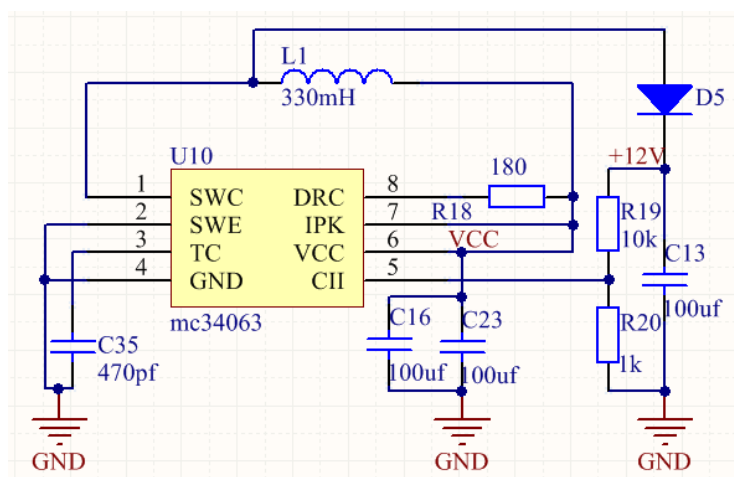


图 3.2 12V 电压产生电路

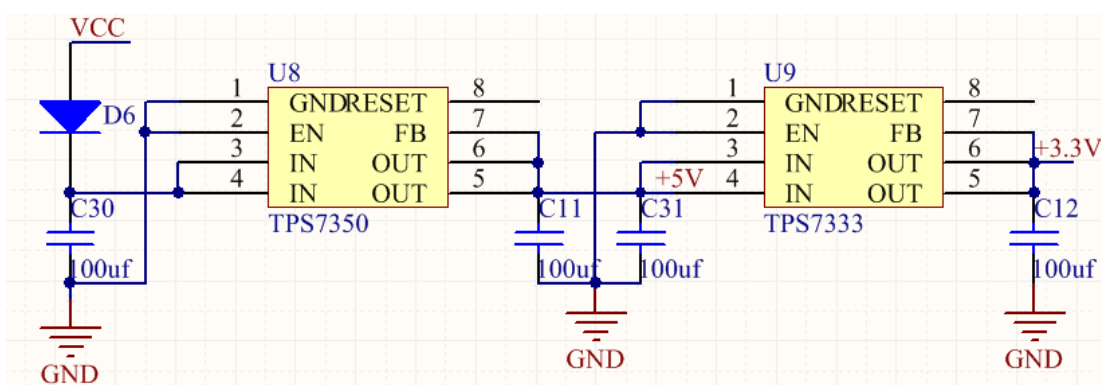


图 3.3 5V、3.3V 电压产生电路

3.3 电机驱动

电机驱动电路和数字电路部分集成在一块板上，在绘制 PCB 板时要注意隔离模拟与数字，否则会使电路不稳定。电机驱动板为一个由分立元件制作的直流电动机可逆双极型桥式驱动器，其功率元件由四支 N 沟道功率 MOSFET 管组成，额定工作电流可以轻易达到 100A 以上，而且内阻很小，大大提高了电动机的工作转矩和转速。

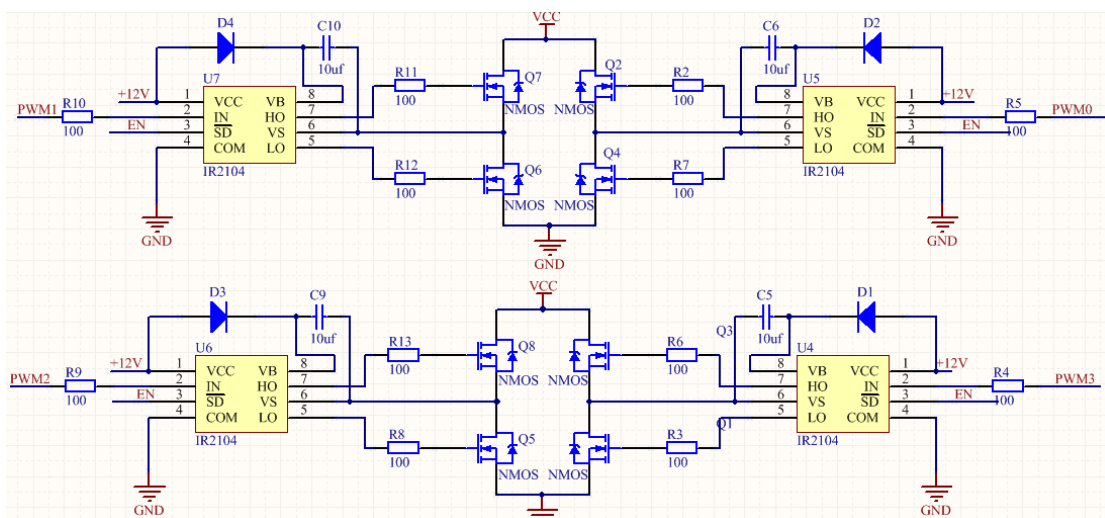


图 3.4 电机驱动电路

3.4 线性 CCD

线性 CCD 是采集赛道元素信息的传感器，电路的优劣直接影响到软件中对赛道的判断和决策。线性 CCD 传感器为 TSL1401，接一级运放调理信号，做好对电源的滤波来降低噪声。本组自己绘制了 CCD 传感器的 PCB 板，在稳定的前提下符合自己的需求。

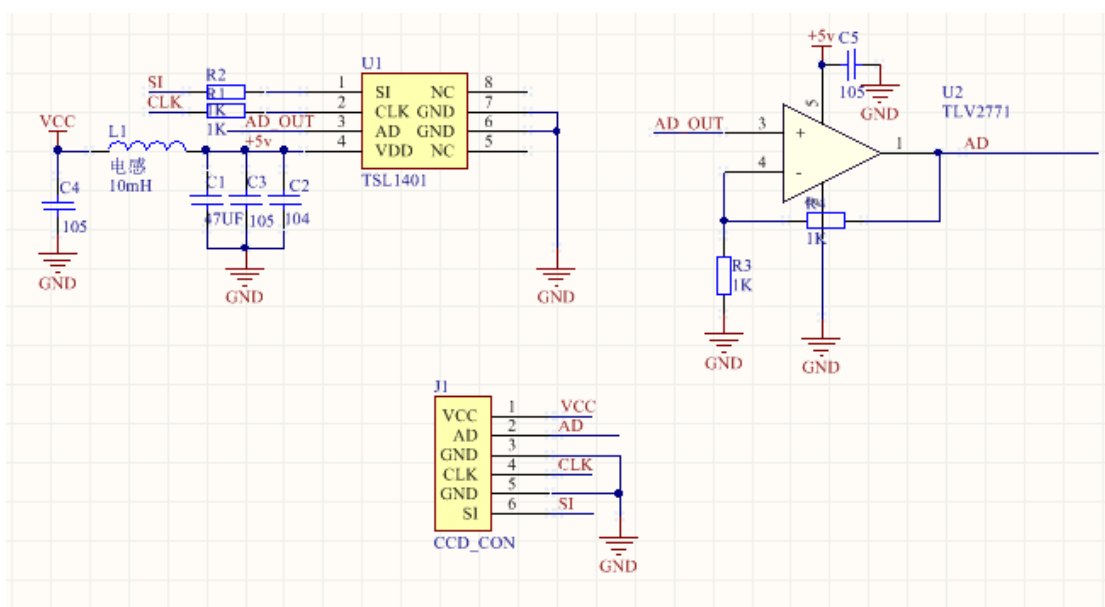


图 3.5 线性 CCD 电路

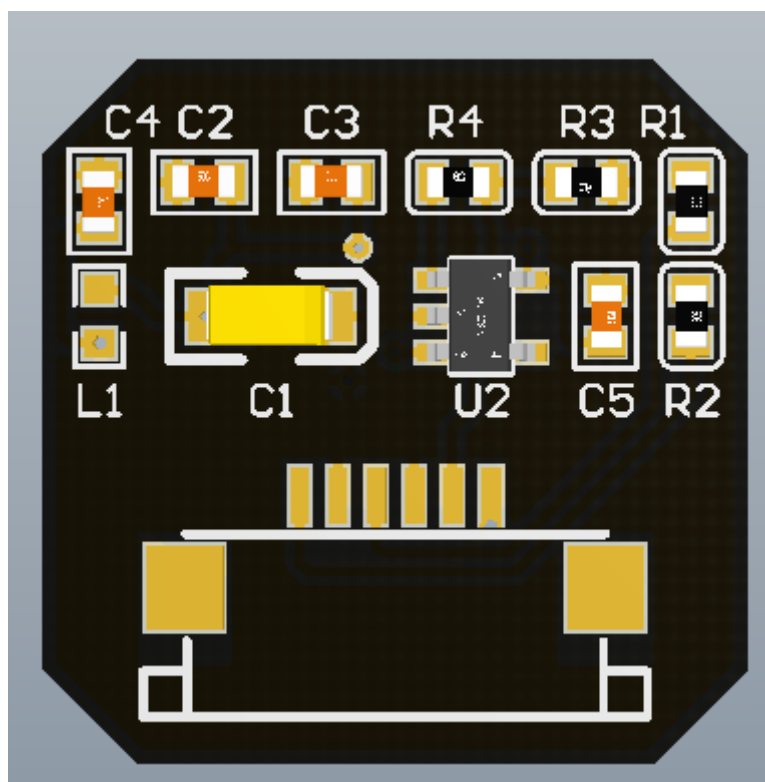


图 3.6 线性 CCD PCB 板

3.5 灯塔起跑模块

本届光电直立组比赛规则中新增加了起跑灯塔，在比赛开始后智能车需要检测灯塔发射的红外线信号起跑，在跑完一圈后同样需要检测红外信号来完成停车。针对比赛规则，我们自行设计了灯塔模块，配合软件相应的检测程序，完成灯塔信号的接收。

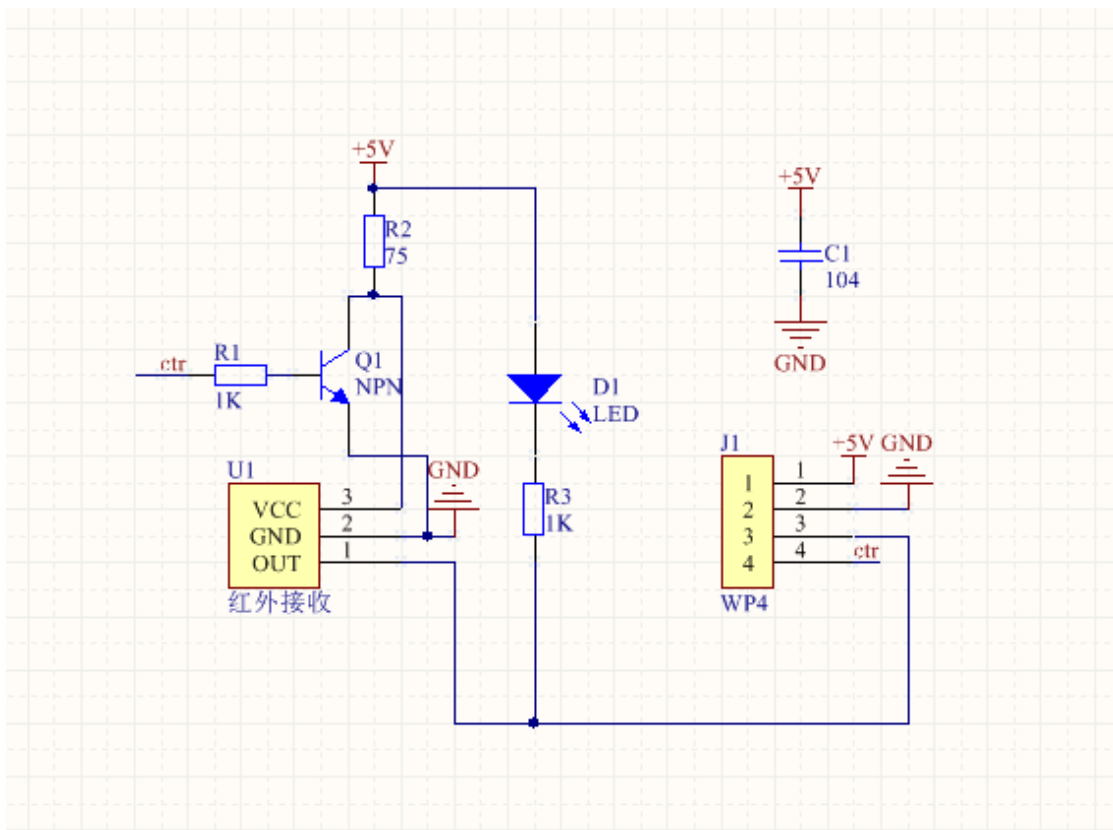
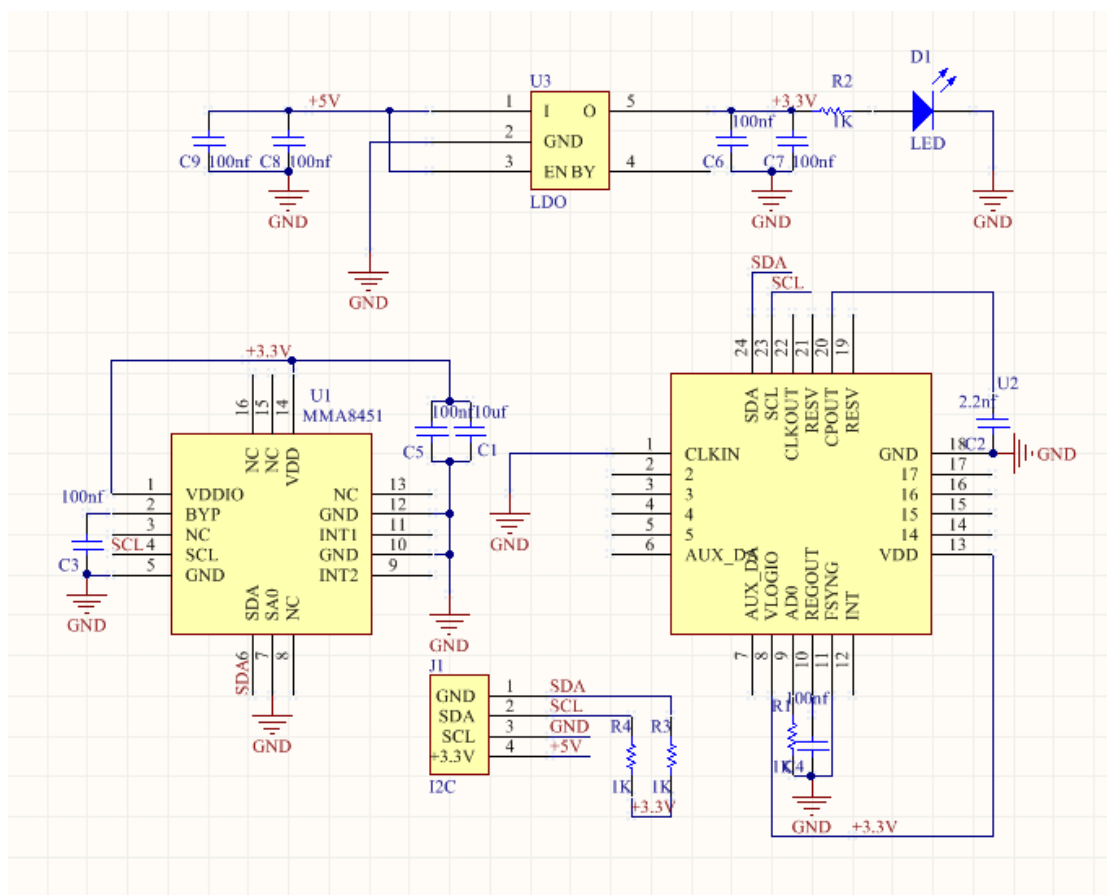


图 3.7 灯塔检测电路

3.6 陀螺仪与加速度计

加速度计和陀螺仪是车体直立必须的信息反馈器件，根据竞赛规则要求，本组选用的是 Freescale 公司的 MMA8451 三轴加速度传感器和 MPU3050 陀螺仪。原理图如图所示



第四章 软件系统的设计及实现

4.1 软件系统总体设计

主程序只负责初始化和数据发送，OLED 显示等任务，在定时中断中进行传感器数据、采集 CCD 曝光、控制量计算以及控制量输出等操作。

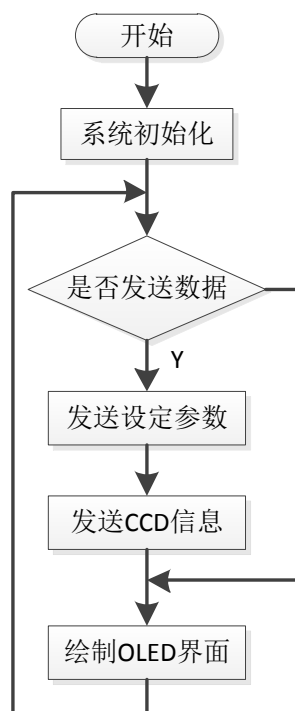


图4.1 主程序流程图

我们以 5ms 为周期对直立车进行控制。

软件的主要功能包括有：

- (1) 各传感器信号的采集、处理；
- (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) 车模运行流程控制：程序初始化、车模启动与结束、车模状态监控；
- (5) 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。

第一类包括 1-3 功能，它们属于需要精确时间周期执行，因此可以在一个周期定时中断里完成。第二类包括 4-5 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。这两类任务之间可以通过全局变量实现相互的通讯。

4.2 主要控制算法

4.2.1 直立控制

分别读取加速度和陀螺仪的值，可以计算车模的倾角。对于读取的陀螺仪和加速度计的数值需要减去零偏值。这个数值需要通过实验确定。在车模保持直立静止时，读出两个通道的数值，便是相应的零偏值。这个数值会带有一定的误差，往往会使得车模往一个方向加速行驶。这个误差可以通过后面的速度控制加以消除。

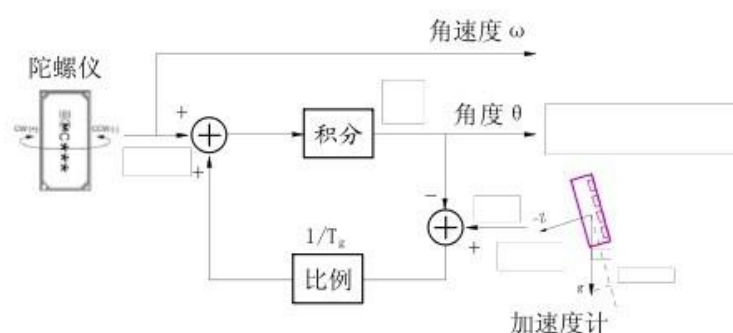


图 4.2 加速度计与陀螺仪的反馈原理图

对于加速度计所得到的数值，通过一个比例系数将它归一化到-90 至 90 之间。同样，后面的陀螺仪的读出的数据也需要通过一个比例系数进行归一化。

利用加速度计所获得的角度信息 θ_g 与陀螺仪积分后的角度 θ 进行比较在上述方案中，将比较的误差信号经过比例 $1/T_g$ 放大之后与陀螺仪输出的角速度信号叠加之后再进行积分。对于加速度计给定的角度 θ_g ，经过比例、积分环节之后产生的角度 θ 必然最终等于 θ_g 。

第一项是重力加速度 Z 轴对应的角度，第二项是陀螺仪加速度经过惯性环节的数值。系统不再存在纯积分环节，所以陀螺仪的角速度微小的偏差不会形成积累误差。其中参数 T_g 决定了这两个惯性环节的时间常数。 T_g 越大，角度输出跟踪 Z 轴输出越慢，但可以有效抑制重力加速度计上的噪声。 T_g 过大，就会放大陀螺仪输出误差。这个参数需要根据实际车模运行效果进行调整，详见后面“车模调试篇”中的方法。

为了避免输出角度 θ 跟踪时间过长，可以采取以下两个方面的措施：

(1) 仔细调整陀螺仪的放大电路, 使得它的零点偏置尽量接近于设定值, 并且稳定。

(2) 在控制电路和程序运行的开始, 尽量保持车模处于直立状态, 这样一开始就使得输出角度 θ 与 θ_g 相等。此后, 加速度计的输出只是消除积分的偏移, 输出角度不会出现很大的偏差。使用加速度计来矫正陀螺仪的积分漂移只是其中一种方法。

4.2.2 速度控制

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和微分两个控制参数, 这里的比例和微分定义是指反馈速度而言。如果相对于车模的反馈位置 (速度的积分量), 这两个参数也可以分别对应称作积分和比例参数。

比例控制参数决定抑制速度误差的能力, 但是只有比例控制, 车模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, 第二个积分是车模倾角所产生的加速度积分。因此车模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。在调整时, 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比例参数, 车模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。使用外力冲击车模, 车模能够很快趋于静止。然后再逐步增加比例和微分控制参数, 使得车模抵抗外部干扰冲击的能力逐步增强。当比例和微分参数达到一定的数值后, 车模开始出现摆动现象。至此可以知道比例和微分的大体范围。在此范围内通过几次调整测试, 可以初步确定速度控制参数的最佳组合。

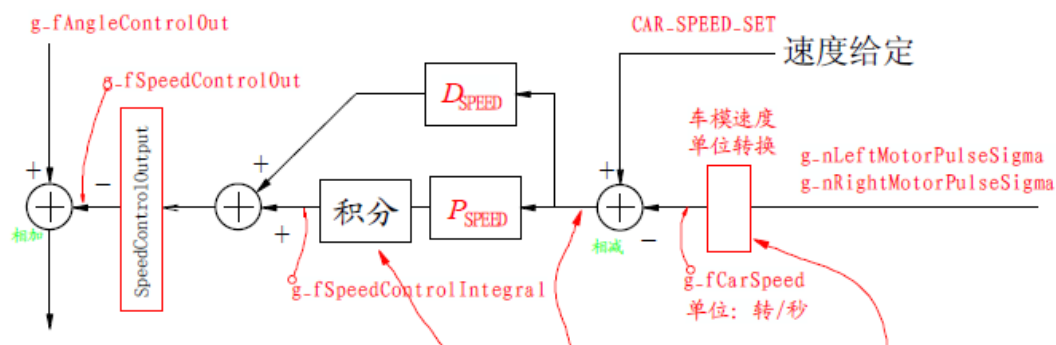


图 4.3 速度控制原理

4.2.3 转向控制

方向控制包括两个参数：比例控制参数和微分控制参数。比例控制参数可以使得车模方向恢复正确位置。当该参数逐步增加时，车模方向回复速度逐步加快。当比例参数增加到一定数值之后，车模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参数可以抑制这种方向过冲的现象。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车模方向控制即迅速又不会出现过冲现象。

通过电机差速控制，可以实现车模方向控制。差速的控制方法参见后面“车模方向控制”。电机运动控制是通过改变施加在其上的驱动电压大小实现的。对于电机的电磁模型、动力学模型以及车模的动力学模型进行分析和简化，可以将电机转速与施加在其上的电压之间的关系简化成如下的一阶惯性环节模型。

电机运动明显分为两个阶段：第一个阶段是加速阶段；第二个阶段为恒速阶段。其中，在加速阶段，电机带动车模后轮进行加速运动，加速度近似和施加在电机上的电压成正比，加速阶段的时间长度取决于时间常数 T_1 。该常数由电机转动惯量、减速齿轮箱减速比、车模的转动惯量决定，一般在十几到几百个毫秒。在恒速阶段，电机带动车模后轮进行恒速运行，运行速度与施加在电机上的电压成正比。

4.3 赛道元素识别与处理策略

本届智能汽车竞赛中新添加了直角和单线元素，对赛道元素的处理必须建立在准确识别的基础上，虽然线性 CCD 采集的信息相比摄像头少了很多，但通过观察边线的变化规律，总结各个元素的特点，也可以做到准确识别各个元素。

4.3.1 CCD 光线自适应算法

线阵 CCD 只能获取赛道原始的亮度/. 灰度信息, 无论是在实验室，还是在比赛场地，赛道环境的亮度差异很大，如果智能车对环境的适应性不高，就很可能折戟沙场，因此有必要让智能车适应各种光线。由于在不同环境下单片机 12 位 AD 值采集回来的白色赛道亮度值在 200-1200 左右，跨度很大，如果只是简单地采用阈值计算黑白边线，遇到赛场光线不均匀的情况下就很容易出错。考虑到在不同光线下，赛道的亮度和边线/背景的亮度比值是近似确定的, 于是我们对采集回来的 CCD 信号作归一化处理—将所有值归一化到 0-127 之间，这样既便于计算，也便于存储和显示，并且对光线的适应能力大大增强。相关代码如下：

```
void CCD_Capture()
{
    uint8 i=0;
    CCD_SI=1;
    delay200ns();
    CCD_CLK=1;
    delay200ns();
    CCD_SI=0;
    delay200ns();
    //采集第一个点
    CCD_Buff[0]=adc_once(ADC0_SE8, ADC_12bit);
    Max_Value=CCD_Buff[0];
    CCD_CLK=0;
    //采集第 2~128 个点
```

```

for(i=1; i<128; i++)
{
    delay200ns();
    CCD_CLK=1;
    delay200ns();
    CCD_Buff[i]=adc_once(ADC0_SE8, ADC_12bit)
    if((Right-Left>20)&&Leftlastfind&&Rightlastfind)
//上次左右边线都找了那么应该在中间搜索最高亮度，把其他亮度抑制掉，防止
反光
    {
        if((i>Left)&&(i<Right))            if(Max_Value<CCD_Buff[i])
            Max_Value=CCD_Buff[i];
    }
    else
    {
        if(Max_Value<CCD_Buff[i])    Max_Value=CCD_Buff[i];
    }
    CCD_CLK=0;
}
//发送第 129 个 clk 作为结束
delay200ns();
CCD_CLK = 1;
delay200ns();
CCD_CLK = 0;
}
//做归一化处理(将采样值限制在 0-127)
void CCD_Normalization()
{
    uint8 i=0;

```

```

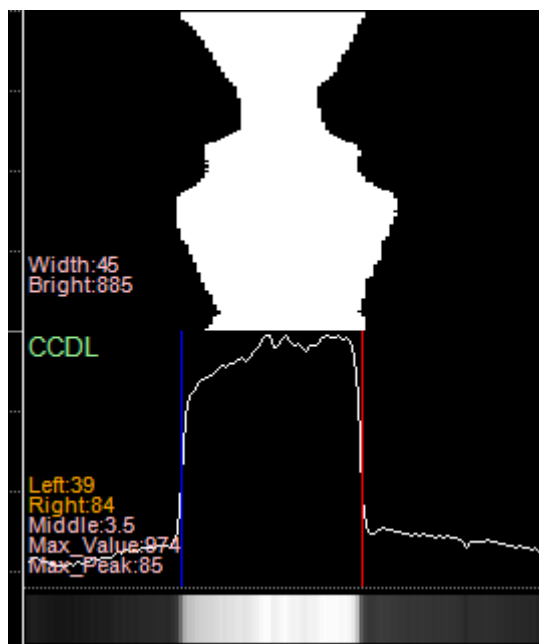
float ratio;
int temp;
if(Max_Value>CCD_Offset)
ratio=127.0/(Max_Value-CCD_Offset);
else ratio=0;
for(i=0; i<128; i++)
{
    if(CCD_Buff[i]>CCD_Offset)
temp=CCD_Buff[i]-CCD_Offset;
    else temp=0;
    CCD_Buff[i]=(int) (temp*ratio);
    if(CCD_Buff[i]>126) CCD_Buff[i]=126;
    CCD_Draw_Buff[i]=CCD_Buff[i];
}
CCD_Draw_Buff[130]=BYTE0(Max_Value);
CCD_Draw_Buff[131]=BYTE1(Max_Value);
}

```

4.3.2 坡道的识别

由于只有两个轮子着地，直立车如果不经特殊处理过坡道很容易导致从坡道上摔下来或者从坡道上飞起，而准确稳定地识别坡道又是对坡道特殊处理的前提，

我们通过坡道宽度变化来识别坡道。坡道宽度变化的典型图像如图所示：



//坡道检测相关代码：

```
void Is_Hill()
{
    static int Width_Increase_Cnt=0, Straight_Cnt;
    float slope;
    float Angle_Change=0;
    int i;

    if((Stop==0)&&(Distance>1)&&(ABS(Middle_Err)<=5)&&(RoadType!=5)&&Rightlastfind&&Leftlastfind)    //偏差小于一定值 在直道，用于判定坡道。跑
    动距离大于 1，再判断值，防止起跑误判
    {
        Straight_Cnt++;
        if(Straight_Cnt>14)    //计算连续 15 次的值
        //前 15 个状态一直是直道
        {
            Width_Increase_Cnt=0;
```

```

Angle_Change = Previous_Angle[0]-Previous_Angle[14];

slope=10* Slope_Calculate(0, 8, Previous_Error);

for(i=14;i>0;i--)                                //统计宽度增加
的次数和变化的角度值
{
    if(Previous_Width[i]>Previous_Width[i-1])
        Width_Increase_Cnt--;
    if(Previous_Width[i]<Previous_Width[i-1])
        Width_Increase_Cnt++;
}

if((Width_Increase_Cnt>=6)&&(Angle_Change<4)&&(ABS(slope)<2))    //
只要宽度增加的次数大于 5 且变化的角度大于-3 度则认为找打了坡道 排除由
于车模角度减小带来的干扰
{

    RoadType=5;
    RoadTypeConfirm=0;
    Character_Distance=Distance;
    Hill_Distance=Distance;
}
}

else
{

```

```
        Angle_Change=0;  
        Straight_Cnt=0;  
    }  
}
```

第五章 系统的开发环境与车模调试

5.1 开发环境

在对程序进行开发和软硬件联调的过程中需要一整套的软件开发与调试工具。程序的开发是在 IAR Embedded Workbench 下进行的，包括源程序的编写、编译和链接，并最终生成可执行文件。包括集成开发环境 IDE、处理器专家库、全芯片显示工具、项目工程管理器、C 交叉编译器、汇编器、链接器以及调试器。使用 J-LINK 来下载程序，把编译好的程序下载到单片机里运行。

IAR Embedded Workbench 开发软件使用界面如图 5.1 所示。

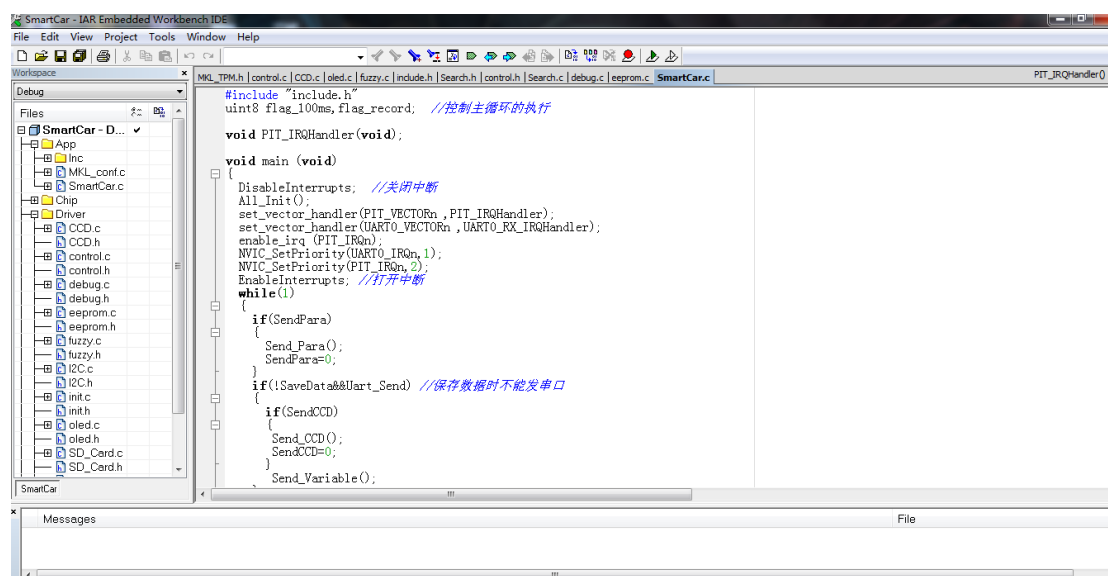


图 5.1 IAR 开发环境界面

5.2 上位机软件

智能车调试的过程中上位机软件功不可没，一个好的上位机可以查看和修改小车的各个参数，查看重要变量的曲线并且记录小车在行进过程中 CCD 的图像信息，并可以进行逐帧分析。我们根据自己的实际需要用了 VB.NET 编写了这套上位机。

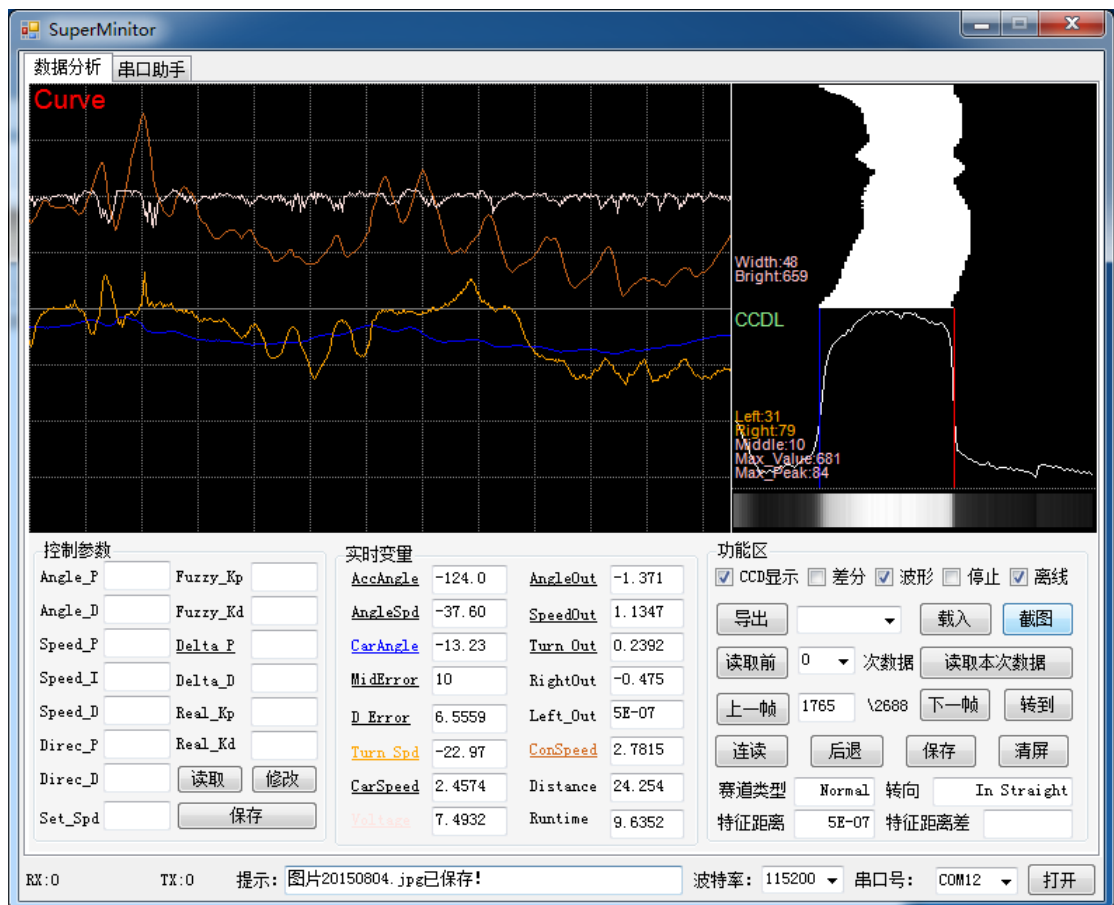


图 5.2 上位机界面

第六章 模型车主要参数

6.1 智能汽车外形参数

经过改装后，智能汽车的外形参数为：

车长：250mm；车宽：250mm；车高：350mm；车重：1.2kg

6.2 智能汽车技术参数

智能汽车相关技术参数如表 6.1 所示：

表 6.1 智能汽车技术参数

项目	参数
车模轴距/轮距（毫米）	200/140
车模平均电流（匀速行驶）（毫安）	3000
电路电容总量（微法）	1500
传感器种类及个数	线性 CCD/1 个
新增加伺服电机个数	0
赛道信息检测空间精度（毫米）	5
赛道信息检测频率（次/秒）	200
主要集成电路种类/数量	13/20

附录 1

主程序:

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

```
uint8 flag_100ms,flag_record; //控制主循环的执行
```

```
void PIT_IRQHandler(void);
```

```
void main (void)
```

```
{
```

```
    DisableInterrupts; //关闭中断
```

```
    All_Init();
```

```
    set_vector_handler(PIT_VECTORn ,PIT_IRQHandler);
```

```
    set_vector_handler(UART0_VECTORn ,UART0_RX_IRQHandler);
```

```
    enable_irq (PIT_IRQn);
```

```
    NVIC_SetPriority(UART0_IRQn,1);
```

```
    NVIC_SetPriority(PIT_IRQn,2);
```

```
    EnableInterrupts; //打开中断
```

```
    while(1)
```

```
    {
```

```
        if(SendPara)
```

```
        {
```

```
            Send_Para();
```

```
            SendPara=0;
```

```
        }
```

```
        if(!SaveData&&Uart_Send) //保存数据时不能发串口
```

```
        {
```

```
            if(SendCCD)
```

```
            {
```

```

        Send_CCD();

        SendCCD=0;

    }

    Send_Variable();
}

if(SendSD&&Stop)
{
    SendSD=0;

    Send_SD();
}

if(flag_100ms)
{
    Read_Switch();

    if(OLED_Refresh)
    {
        Check_BottonPress();

        if(Stop==true|SaveData==false) OLED_Draw_UI();

    }

    else Check_StartPress();

    flag_100ms=0;
}

if(flag_record&&SaveData&&SD_OK)
{
    if(Stop!=true)                //如果小车正在运行，则记录
    {
        if (Starting) Block_Index=1;

        else Record();                //存储数据
    }
}

```

```

else if(CarStopedJustNow) //小车刚停止
{
    CarStopedJustNow=false;
    Write_Information();
}

flag_record=false;
}
}

}

void PIT_IRQHandler(void)
{
    if(PIT_TFLG(PIT0) == 1 )
    {
        static uint8  part=0;
        static uint8  cnt100ms;
        //灯塔相关
        static uint8 light_tower_cnt;
        static uint8 light_tower_delay;
        static int pulse_cnt;
        static uint8 light_tower_on=0;

        part++;
        cnt100ms++;
        SpeedCount++;
        //循环计数控制
        switch(part)
        {
            case 1:
                //直立控制

```

```

        Get_Attitude();
        Angle_Calculate();
        Angle_Control();
        //速度控制
        Get_Speed();    //5ms 采集一次速度
        SpeedCount++;
        if(SpeedCount>= Speed_Filter_Times)
        {
            Speed_Control();
            SpeedCount=0;
        }
        Speed_Control_Output();
        break;
case 2: //CCD 采集
        CCD_Capture(); //采集 CCD
        break;
case 3:
        CCD_Normalization();
        read_buff[0]= Get_Y_Gyro();
        break;
case 4:
        RoadType_Distinguish();

        if((RunTime<0.1)&&(Stop==0)) //防止起步的时候乱飘
        {
            Rightlastfind=0;
            Leftlastfind=0;
            Middle_Err=0;
        }

```

```

        Direction_Control();

        DirectionCount=0;

        flag_record=1;                //开记录标志位

        SendCCD=1;

        break;

case 5:

        part=0;

        read_buff[1]= Get_Y_Gyro();


        break;

default:

        break;

}

DirectionCount++;

Direction_Control_Output();

Moto_Out();

//////////////////////灯塔测试

if(Stop)

{

    light_tower_delay++;

    if(light_tower_on&&light_tower_delay==100)

    {

        light_tower_delay=0;

        pulse_cnt=lptmr_pulse_get();

        if(pulse_cnt<10)

        {

            if(Light_Tower_Index!=0) Light_Tower_OFF;

            light_tower_on=0;


```

```

    }

    lptmr_pulse_clean();
}

if(light_tower_on==0&&light_tower_delay==10) //关断灯塔 10ms 后再打开
{
    light_tower_delay=0;
    Light_Tower_ON;
    light_tower_on=1;
    lptmr_pulse_clean();
}
}

//////////////////////LED 状态指示

if(cnt100ms>=100)
{
    if(Stop)
    {
        LED_BLUE_TURN;
        LED_RED_OFF;
        LED_GREEN_OFF;
    }
    else
    {
        if(Starting)
        {
            LED_GREEN_TURN;
            LED_RED_OFF;
            LED_BLUE_OFF;
        }
        else

```

```

    {
        LED_RED_TURN;

        LED_GREEN_OFF;

        LED_BLUE_OFF;
    }
}

cnt100ms=0;

flag_100ms=1;
}

//////////////////////启动
if(Starting)
{
    if(Light_Tower_Index==0)        //不起用灯塔检测，倒计时启动
    {
        Start_Cnt--;

        if(Start_Cnt==0)

        {
            Starting=false;
        }
    }

    else                                //灯塔起跑

    {
        light_tower_delay++;

        if(light_tower_detected==0)

        {
            if(light_tower_delay==200)

            {
                pulse_cnt=lp_tmr_pulse_get();

                if(pulse_cnt>150)

```



```

    {
        light_tower_detected=1;
        light_tower_delay=0;
        Light_Tower_ON;
    }
    else    Light_Tower_OFF;
    lptmr_pulse_clean();
}

if(light_tower_delay>=210)
{
    light_tower_delay=0;
    Light_Tower_ON;
}
}

else
{
    if(light_tower_delay==10)                //起跑时距离灯塔很近所以接收到的
    以接收到的是连续波，一旦断掉就认为可以起跑了
    {
        pulse_cnt=lptmr_pulse_get();
        if(pulse_cnt<5)
        {
            Starting=false;
            light_tower_cnt=0;
            light_tower_on=0;
        }
        light_tower_delay=0;
        lptmr_pulse_clean();
    }
}

```

```

    }
}
}

//////////////////////停车

if(((RunTime>29&&Light_Tower_Index==1)|| (Light_Tower_Index==3))&&Stop==0&&Starting==0)
//灯塔停车
{
    light_tower_delay++;
    if(light_tower_on&&light_tower_delay==50) //50ms 扫描
    {
        light_tower_delay=0;
        pulse_cnt=lpmtmr_pulse_get();
        if(pulse_cnt>35)           //说明遇到强信号了
        {
            light_tower_cnt++;
        }
        else
        {
            if(light_tower_cnt>4)
            {
                ForceStop=true;//防止干扰的产生
            }
            light_tower_cnt=0;
        }
        Light_Tower_OFF;
        light_tower_on=0;
        lpmtmr_pulse_clean();
    }
}

```

```

        if(light_tower_on==0&&light_tower_delay==10) //关断灯塔 10ms 后再打开
        {
            light_tower_delay=0;
            Light_Tower_ON;
            light_tower_on=1;
            lptmr_pulse_clean();
        }
    }

    //////////////////////////////////////统计时间及速度
    if(Stop==false&&Starting==false)
    {
        RunTime=RunTime+0.001;
        AverageSpeed=Distance/RunTime;//求得运行时间
    }
}

PIT_Flag_Clear(PIT0);        //清中断标志位
}

```

附录 2

主板总图:

