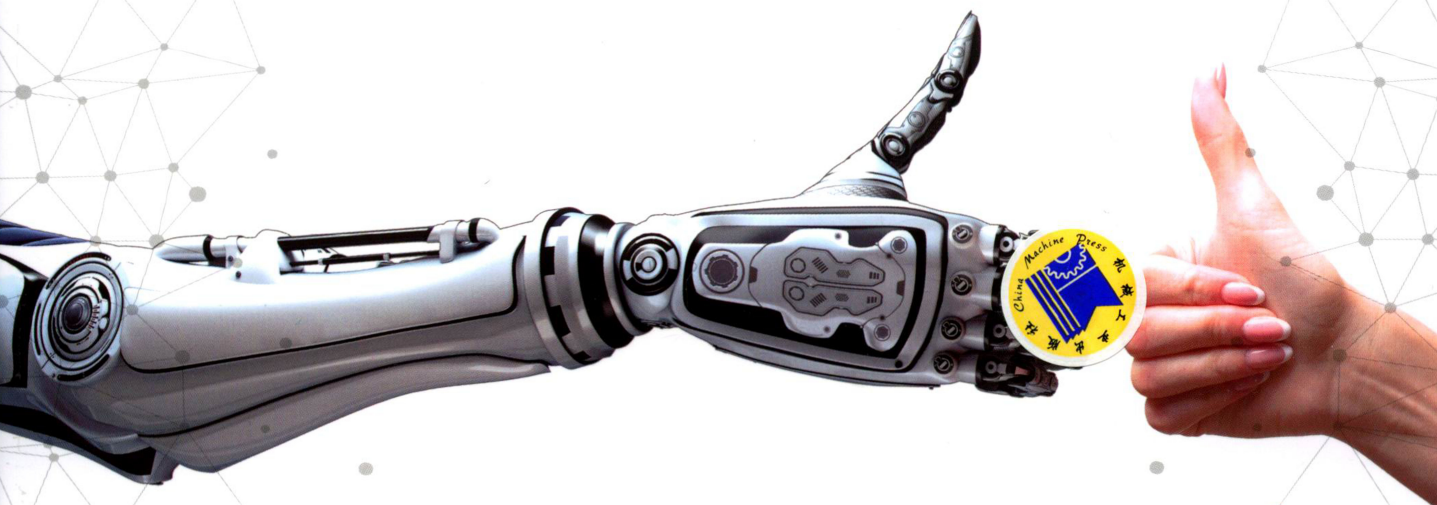


COMPETITION ROBOT

竞赛机器人

王志良 解仑 戎豫 曹策 / 编著

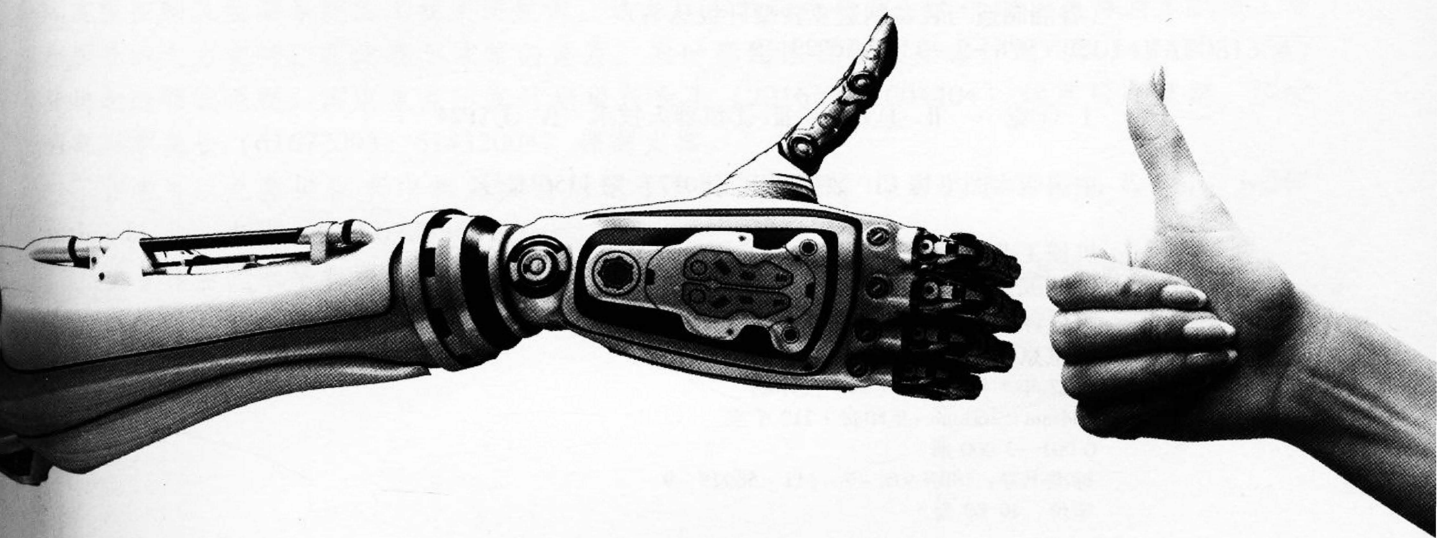


竞赛机器人是青少年学习掌握机器人的重要途径
本书包括竞赛机器人的国际赛事、控制器、开发平台
结构与部件、编程、运动控制算法、工程化应用实例

COMPETITION
ROBOT

竞赛机器人

王志良 解仑 戎豫 曹策 / 编著



本书通过实验操作的方法向读者传授竞赛机器人的相关技术,在介绍多种竞赛机器人的功能及其国内外赛事和竞赛机器人的控制器技术、软件开发平台使用方法、整体结构设计、硬件电路设计与制作、软件模块的编程方法、机器人走迷宫的数学算法以及迷宫场地制作之后,进行了竞赛机器人工程化应用的研究分析,以门座式起重机控制系统为例,分析多自由度协调控制及在线监测等工程应用应用案例。

本书可以作为大学生、相关专业研究生的学习参考书,也适宜从事计算机、机械电子工程、自动化、智能科学、数字娱乐的科技人员阅读。

图书在版编目 (CIP) 数据

竞赛机器人/王志良等编著. —2 版. —北京:机械工业出版社, 2017. 6

(智能制造与装备制造业转型升级丛书)

ISBN 978 - 7 - 111 - 56929 - 9

I. ①竞… II. ①王… III. ①机器人技术 IV. ①TP24

中国版本图书馆 CIP 数据核字 (2017) 第 115618 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 罗 莉 责任编辑: 翟天睿

责任印制: 常天培 责任校对: 任秀丽

北京京丰印刷厂印刷

2017 年 7 月第 2 版·第 1 次印刷

184mm×260mm·9 印张·212 千字

0 001—3 000 册

标准书号: ISBN 978 - 7 - 111 - 56929 - 9

定价: 49.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线: 010-88361066 机工官网: www.cmpbook.com

读者购书热线: 010-68326294 机工官博: weibo.com/cmp1952

010-88379203 金书网: www.golden-book.com

封面防伪标均为盗版 教育服务网: www.cmpedu.com

前 言

人工心理的重要技术基础与应用领域之一是机器人学科与技术。尤其是与人工智能和机器人情感密切相关的家用机器人和教育机器人。竞赛机器人是教育机器人的一种重要形式，是青少年学习掌握机器人技术的一种主要途径，它主要涉及人工智能、计算机控制、传感器技术和竞技技术。如何更好、更容易地让青少年学习掌握这门技术，尤其是通过实验制作的方法，由浅入深地向读者传授机器人技术，是我们撰写本书的主要目的。

本书以大学生、研究生为读者对象，详细讲述竞赛机器人相关技术与方法。全书共分为8章。第1章主要介绍各种竞赛机器人的各种功能以及其国内外赛事；第2章主要介绍竞赛机器人的控制器技术；第3章介绍一种竞赛机器人软件开发平台——MPLABIDE 的使用方法；第4章介绍竞赛机器人平台的整体结构设计；第5章介绍竞赛机器人平台的硬件电路设计与制作；第6章介绍竞赛机器人软件模块的编程方法；第7章介绍竞赛机器人的控制方法；第8章介绍了机器人工程化应用实例。

本书由王志良、解仑、戎豫、曹策编著。全书由以上作者们制定主要内容、章节划分并统稿。第1、2、5、6章主要由王志良、解仑编写；第3章和第4章的遥控器制作部分主要由王志良、解仑、戎豫编写；第8章由解仑、曹策编写；第4章和第7章主要由曹策、解仑编写。本书第8章主要分析了作者所在课题组设计制作的机器人大型化工程装备——门座式起重机控制及监测系统的工程实用案例。这是本书的特色所在。本书的出版得到了机械工业出版社的大力支持，在此表示诚挚的谢意。同时感谢国家科技支撑计划（2014BAF081304）先进制造领域课题，国家重点研发计划重点专项（2016YFB1001404）信息领域课题，国家自然科学基金（61672093，61432004）课题支撑。

作者可以免费提供书中相关实验的电子文档及程序。作者的电子信箱 E-mail: wzl@ustb.edu.cn，电话：010-62332641。

由于作者的水平有限，书中肯定有不少的缺点与疏漏之处，敬请读者批评指正。

目 录

前言

第 1 章 绪论 1

1.1 机器人概述 1

1.2 竞赛机器人的国际赛事 2

1.2.1 RoboCup 比赛 2

1.2.2 FIRACup 比赛 4

1.2.3 迷宫机器人比赛 4

1.2.4 寻线机器人比赛 6

1.2.5 灭火机器人比赛 7

1.2.6 舞蹈机器人比赛 7

1.2.7 相扑机器人比赛 8

1.2.8 搏斗机器人 9

1.3 国内的机器人比赛 9

1.4 竞赛机器人平台的主要功能 15

参考文献 16

第 2 章 竞赛机器人的控制器 18

2.1 引言 18

2.2 机器人控制器类型 18

2.2.1 串行处理结构 18

2.2.2 并行处理结构 20

2.3 三种机器人控制器的比较 21

2.3.1 51 系列单片机 21

2.3.2 PIC 系列单片机 23

2.3.3 AVR 系列单片机 24

2.4 PIC16F877 (A) PIC 系列单片机 25

2.4.1 PIC 系列单片机性能特点 25

2.4.2 PIC16F87X 单片机的结构与性能特点 27

2.4.3 单片机 C 语言编程 29

2.5 机器人控制器的发展趋势 30

参考文献 31

第 3 章 竞赛机器人的软件开发平台 33

3.1 MPLAB 概述 33

3.1.1 MPLAB 集成开发环境的组成 33

3.1.2 MPLAB 运用方式 34

3.1.3 MPLAB 对硬件与软件的要求 34

3.2 MPLAB 的安装和启动 35

3.2.1 完整的 MPLAB 安装 35

3.2.2 MPLAB 的启动 35

3.3 MPLAB 的使用 35

3.3.1 启动 MPLAB IDE 35

3.3.2 创建源文件 36

3.3.3 创建项目 38

3.3.4 给项目添加文件 38

3.3.5 选择器件 40

3.3.6 设置配置位 41

3.3.7 选择 MPLAB ICD2 作为调试器 41

3.3.8 通过向导完成调试器的设置 42

3.3.9 建立 PC 与 MPLAB ICD2 仿真下载器之间的通信联系 43

3.3.10 更新 MPLAB ICD2 固件 (操作系统) 46

3.3.11 生成目标文件 (编译) 46

3.3.12 下载目标代码 48

3.3.13 运行和调试 50

3.3.14 在编程器模式下载目标代码 50

3.3.15 文件保存 52

3.4 软件编程基础 52

3.4.1 C 语言概述 53

3.4.2 整型量 53

3.4.3 符号常量 54

3.4.4 简单赋值运算与赋值表达式 54

3.4.5 控制语句 55

参考文献 61

第 4 章 竞赛机器人的结构与部件 62

4.1 平台的机械结构 62

4.2 平台的行走机构 64

4.3 舵机 65

4.4 将舵机改装成执行机构 67

4.5 竞赛机器人平台的组装 72

4.6 竞赛机器人专用遥控器的制作 74

参考文献 76

第 5 章 竞赛机器人的电子电路 77

5.1 硬件电路组成	77	7.2.1 临界比例度法	111
5.2 执行机构驱动电路	77	7.2.2 衰减曲线法	111
5.3 传感器检测电路	80	7.2.3 试凑法	112
5.3.1 起跑线的检测	81	7.3 数字 PID 控制算法改进	112
5.3.2 迷宫隔栅检测部分	81	7.3.1 积分项改进	112
5.3.3 传感器的特性曲线	83	7.3.2 微分项改进	114
5.3.4 沿跑道中线的运行	85	7.4 PID 控制在竞赛智能车上的实现	116
5.4 无线发射接收模块	85	7.4.1 PID 控制器输入标准值的 设定	116
5.5 数据存储模块	88	7.4.2 PID 控制器被控对象控制参量 的设定	116
参考文献	90	7.4.3 智能汽车车速 PID 控制器的工作 原理	116
第 6 章 竞赛机器人的编程技术	91	7.4.4 舵机的 PD 控制	117
6.1 竞赛机器人的控制	91	7.5 模糊控制算法	117
6.1.1 CPU 引脚资源分配	91	7.5.1 模糊控制基本原理	117
6.1.2 初始化模块	92	7.5.2 模糊控制器的设计	118
6.1.3 运动模块	93	7.6 赛道记忆算法	121
6.1.4 A-D 转换模块	97	7.6.1 赛道记忆算法前提	122
6.2 迷宫智能算法的实现	98	7.6.2 赛道记忆算法描述	122
6.2.1 沿跑道中线前进的判断程序	98	参考文献	123
6.2.2 无记忆功能迷宫算法的 编程实现	99	第 8 章 工程化应用实例	124
6.2.3 有记忆功能的迷宫算法分析	100	8.1 交流电动机矢量控制理论	124
6.3 上位机软件	104	8.1.1 感应电动机的空间矢量	124
参考文献	105	8.1.2 感应电动机矢量变换控制	125
第 7 章 竞赛机器人的运动控制 算法	106	8.2 大型起重机控制系统	130
7.1 PID 控制算法	106	8.3 大型起重机安全监控管理信息 系统	133
7.1.1 PID 控制简介	106	参考文献	136
7.1.2 PID 控制算法分类	107		
7.2 PID 参数整定	111		

第 1 章 绪 论

1.1 机器人概述

机器人一词最早出现于 1920 年捷克作家 Karel Capek 的剧本《罗萨姆的万能机器人》中，在该剧中“Robota”这个词的本意是苦力，是剧作家笔下的一个具有人的外表、特征和功能的机器，是一种人造的劳动力。我国采用的是意译，实际机器人的外形大多很少与人有相似之处。

机器人一词的出现和第一台工业机器人的问世都是近几十年的事。然而人们对机器人的幻想与追求却已有 3000 多年的历史。人类希望制造一种像人一样的机器，以便代替人类完成各种工作。西周时期，我国的能工巧匠偃师就研制出了能歌善舞的机器——伶人，这是我国最早记载的机器人。后汉三国时期，蜀国丞相诸葛亮成功地创造出了“木牛流马”，并用其运送军粮，支援前方战争。公元前 2 世纪，亚历山大时代的古希腊人发明了最原始的机器人——自动机，它是以水、空气和蒸汽压力为动力的会动的雕像，它可以自己开门，还可以借助蒸汽唱歌。

现代机器人的研究始于 20 世纪中期，其技术背景是计算机和自动化技术的发展，以及核能的开发利用。自 1946 年第一台数字电子计算机问世以来，计算机技术取得了惊人的进步，向高速度、大容量、低价格的方向发展。大批量生产的迫切需求推动了自动化技术的进展，其结果之一便是 1952 年数控机床的诞生。与数控机床相关的控制、机械零件的研究又为机器人的开发奠定了基础。另一方面，核能实验室的恶劣环境要求用某些操作机械代替人处理放射性物质。在这一需求背景下，美国原子能委员会的阿尔贡研究所于 1947 年开发了遥控机械手，1948 年又开发了机械式的主从机械手。

1954 年美国戴沃尔最早提出了工业机器人的概念，并申请了专利。该专利的要点是借助伺服技术控制机器人的关节，利用人手对机器人进行动作示教，机器人能实现动作的记录和再现。这就是所谓的示教再现机器人。现有的机器人差不多都采用这种控制方式。

作为机器人产品最早的实用机型（示教再现）是 1962 年美国 AMF 公司推出的“VER-STRAN”和 UNIMATION 公司推出的“UNIMATE”。这些工业机器人的控制方式与数控机床大致相似，但外形特征迥异，主要由类似人的手和臂组成。1965 年，MIT（美国麻省理工学院）的 Robots 演示了第一个具有视觉传感器的、能识别与定位的、简单积木的机器人系统。

1973 年，瑞典的 ABB 与德国的 KUKA 分别将各自研发的最新一代工业机器人推向市场，IRB6 是 ABB 公司设计研发的世界第一款微处理器控制的全电动工业机器人；而 KUKA 推出的第一代自主研发机器人 FAMULUS 拥有多达 6 个驱动轴来完成复杂机械动作^[1]。

如今机器人发展的特点可概括为：从横向看，应用面越来越宽，由 95% 的工业应用扩展到更多领域的非工业应用，像做手术、采摘水果、剪枝、巷道掘进、侦查、排雷，还有空间机器人、潜海机器人；从纵向看，机器人的种类会越来越多，像进入人体的微型机器人，

已成为一个新方向，可以小到像一个米粒般大小。最重要的是机器人智能化得到加强，机器人会更加聪明。具有感知能力和思维能力的机器人是我们追求的目标。归纳起来，机器人的发展除了受到人们想象力的限制外，还受到计算机技术、传感技术、材料的限制。

反过来，机器人的发展促进相关技术的进步。从1959年第一台机器人诞生到20世纪80年代初，机器人技术经历了一个长期缓慢的发展过程。到了90年代，随着计算机技术、微电子技术、网络技术等的快速发展，机器人技术也得到了飞速发展。除了工业机器人水平不断提高之外，各种用于非制造业的先进机器人系统也有了长足的进展。

现代机器人在处理器、远程控制与通信、信息采集系统、自动化设备等方面都有了显著的提升。同时，现代机器人的研发重心也在从工业机器人向服务型机器人转移，服务型机器人的成本相对更低，受众面也更广，随着人们对居家生活质量越来越高的要求，服务型机器人走入千家万户已成为机器人产业的新趋势。

随着计算机技术和人工智能技术的飞速发展，使机器人在功能和技术层次上有了很大的提高，移动机器人和机器人的视觉和触觉等技术就是典型的代表。由于这些技术的发展，推动了机器人概念的延伸。20世纪80年代，将具有感觉、思考、决策和动作能力的系统称为智能机器人，这是一个概括的、含义广泛的概念。这一概念不但指导了机器人技术的研究和应用，而且又赋予了机器人技术向深广发展的巨大空间，水下机器人、空间机器人、空中机器人、地面机器人、微小型机器人等各种用途的机器人相继问世，许多梦想成为了现实。将机器人的技术（如传感技术、智能技术、控制技术）扩散和渗透到各个领域，形成了各式各样的新机器——机器人化机器。当前与信息技术的交互和融合又产生了“软件机器人”“网络机器人”的名称，这也说明了机器人所具有的创新活力。

1.2 竞赛机器人的国际赛事

在世界科技和经济大发展的情况下，为了普及机器人知识，促进机器人技术的研究发展，世界上很多国家和地区纷纷开展了各种类型的机器人比赛。机器人比赛除了需要采用比较先进的技术外，还有几个特点：一是对抗性强的比赛时间短，有很强的观赏性；二是能够充分地发挥参赛者的创造性思维；三是比赛策略和临场发挥非常重要；四是比赛规则和题目变化较大。

目前，日本、美国、英国等发达国家的机器人竞赛发展很快。在这些国家，从小学生到大学生，从学校到社会，都有各种不同的机器人比赛，日本每年都有很多机器人比赛，如走迷宫机器人比赛（老鼠机器人）、寻白线机器人、大学生机器人大赛。在英国有：ROBOT WARS、techno games，全民参加比赛，政府监督，电视台主办，形成了较为完整的机器人竞赛体系。因为有了这样的比赛，人们更加了解机器人，机器人的发展也得到很大的促进。

下面就简要介绍一下国际上比较有名的机器人竞赛：RoboCup 竞赛、FIRACup 竞赛、迷宫机器人竞赛、寻线机器人竞赛、迷宫操作机器人竞赛、灭火机器人竞赛、舞蹈机器人竞赛、相扑机器人竞赛等。

1.2.1 RoboCup 比赛

RoboCup（Robot World Cup，机器人世界杯）比赛的前身是日本机器人足球赛，是由日

本的北野宏明和浅田埤等学者于 1993 年 6 月发起创办的。它是通过提供足球比赛这样一个标准问题来促进人工智能、机器人以及相关领域的研究而建立的国际组织。其最终目标是经过 50 年左右的研究，使机器人足球队能战胜人类足球队。

1997 年 8 月 23 日，首届机器人足球世界杯赛在日本名古屋举行，来自美国、日本、欧洲和澳洲的 40 多支机器人足球队参加了比赛。众多国际知名学府成为 RoboCup 的首批参赛者。在举办第一届 RoboCup 的同时，1997 国际人工智能联合大会（IJCAI97）也在名古屋举行，在这之后，1999 年和 2003 年，RoboCup 和 IJCAI 都在同时同地举办，这也无疑表明了 RoboCup 在人工智能界的地位。图 1-1 所示为足球机器人比赛的两幅图片。

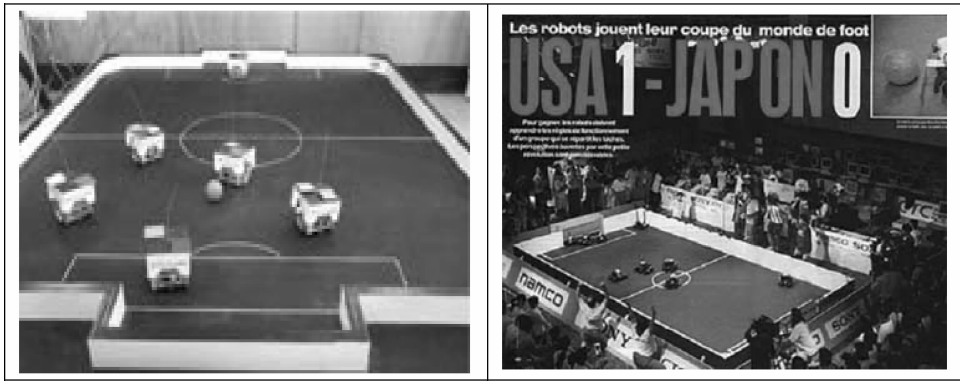


图 1-1 足球机器人比赛

RoboCup 至今已举办了 18 届，目前 RoboCup 的活动包括有：技术会议、机器人比赛、挑战计划、教育计划、基础发展等，而机器人比赛是所有活动的核心。目前机器人比赛的类型包括：

(1) 足球比赛 (RoboCupSoccer)

仿真组 (Simulation Team)；

小型组 (Small Size Team)；

中型组 (Middle Size Team)；

四腿组 (Sony Legged Team)；

类人组 (Humanoid League)。

(2) 营救比赛 (RoboCupRescue)

仿真组 (Rescue Simulation Team)；

机器组 (Rescue Robot Team)。

(3) 青少年比赛 (RoboCupJunior)

足球机器人 (Soccer Challenge)；

舞蹈机器人 (Dance Challenge)。

RoboCup 以其独特的魅力，成为各类国际机器人竞赛中最具水平和影响力的赛事之一。许多世界名校包括美国卡耐基-梅隆大学、康奈尔大学、斯坦福大学，澳大利亚新南威尔士大学，日本东京大学，新加坡南洋理工大学，荷兰阿姆斯特丹大学，德国多特蒙德大学以及

中国的清华大学、中国科技大学等都积极参与。由机器人联合会组织的 1999 年第四届机器人世界杯足球赛于 8 月 4~8 日在巴西的坎皮那斯举行。东北大学的“牛牛”队代表我国参加了比赛，并取得了 3 对 3 对抗赛的第五名和标准动作比赛的第一名的好成绩，为祖国争得了荣誉。第四届 RoboCup 足球赛及学术大会于 2000 年 8 月 25 日~9 月 3 日在澳大利亚墨尔本举行。来自约 30 个国家的 154 支球队参加了预选，约 100 支球队参加了正式比赛。在 40 支球队参赛的仿真组比赛中，中国科技大学“蓝鹰”队攻入 20 个球，“蓝鹰”队是第一支进入 RoboCup 的中国球队。最近两年的仿真组八强中，来自中国的球队在 RoboCup 比赛中表现突出，清华风神队更是连续夺得两次冠军和一次亚军。

1.2.2 FIRACup 比赛

FIRA 微机器人足球比赛最早是由韩国高等技术研究院（Korea Advanced Institute of Science and Technology, KAIST）的金钟焕（Jong-Hwan Kim）教授于 1995 年提出，并于 1996 年在 KAIST 所在的韩国大田（Daejeon）举办了第一届国际比赛。1997 年 6 月，第二届微机器人足球比赛（MiroSot97）在 KAIST 举行期间，国际机器人足球联盟（Federation of International Robot-soccer Association, FIRA）宣告成立。此后 FIRA 在全球范围内每年举行一次国际机器人足球联盟杯（FIRA Cup）比赛，同时举办学术会议（FIRA Congress），供参赛者交流他们在机器人足球研究方面的经验和技能，这成为了 FIRA 的惯例。

在 FIRA 比赛蓬勃开展的同时，有关机器人足球系统和机器人足球比赛的理论研究也取得了长足进展。每一届世界杯比赛之前，主办者都会举行培训和研讨班，并在比赛举行的同时召开机器人足球专题的国际学术会议，这些论文集中介绍了与机器人足球相关的视觉系统、运动规划、动作设计、策略选择等领域的最新研究成果。这些学术研讨和交流活动，极大地促进了相关学科的理论研究。

理论研究的成果使得机器人足球比赛的水平不断提高。在 1996 年的第一届 MiroSot 比赛中，大多数参赛队使用的视觉系统的采集/处理速度仅为 10 帧/s，机器人速度也不过 50cm/s。仅过两年，来自韩国的 Keys 队，凭借他们高达 60 次/s 的视频采集/处理速度和机器人 2m/s 的运动速度，在法国巴黎举行的 FIRA'98 世界锦标赛中一举夺魁，其足球机器人的表现让人惊叹不已。这些进步得益于电子和计算机技术的发展带来的硬件性能的飞速提高。

另一方面，有关足球机器人动作和策略的研究也成绩显著。早先的比赛当中，机器人之间缺乏合理的分工协作，很容易挤作一团。现在这种现象已不存在，随着策略研究的不断成熟，比赛的精彩程度也在不断增加。MiroSot 机器人的设计制造技术取得重大的进展，随着新的单片机技术的发展，机器人的控制更加可靠，运动速度成倍增加，同时也形成了共轴两轮独立驱动这种一直延续至今的小车布局。足球机器人如图 1-2 所示。

1.2.3 迷宫机器人比赛

第一次参赛的迷宫机器人（MicroMouse）是机械的。1972 年，《Machine Design》杂志发起了一场比赛，用弹簧做动力的迷宫机器人沿着跑道，比拼毅力，看谁跑的最远。最后获得第一名的迷宫机器人跑了 251.6m（825.3ft）。

迷宫机器人（MicroMouse）是一种小型自主探测迷宫的机器人。1977 年，IEEE（美国电气与电子工程师学会）会刊 Spectrum 杂志提议并在世界范围内征集 MicroMouse 比赛，参

赛的迷宫机器人应在规定的迷宫区域内看谁最快地识别并走出迷宫。在 1977 年 5 月，Spectrum 宣布第一届美国迷宫机器人竞赛于 1979 年 6 月在纽约举行。

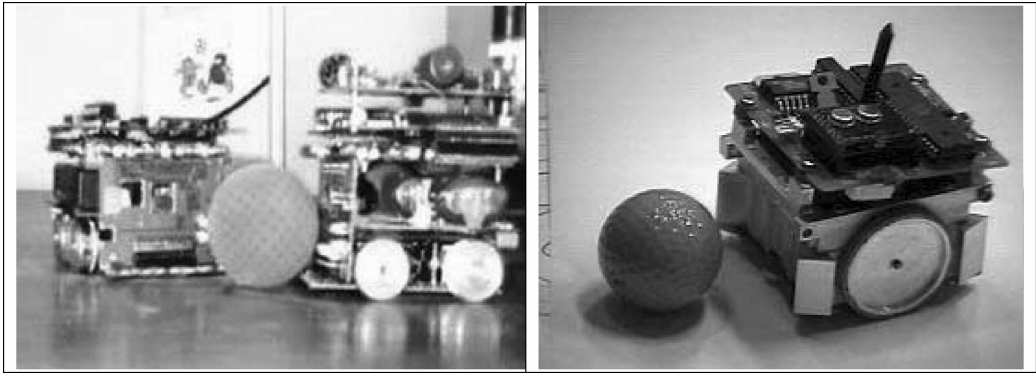


图 1-2 足球机器人

1978 年夏，6000 只迷宫机器人在美国经历了三轮比赛，其中的 15 只迷宫机器人一路过关斩将，于 1979 年最终获得参加 NCC（全美计算机会议）纽约年会决赛的资格。结果 Boland and Dikkeknback 的“月光号迷宫机器人”以全程 29.78s 的成绩夺冠。此后，此项赛事在一片褒奖声中一发不可收拾。

最著名的迷宫机器人是：Nick Smith’s Sterling Mouse、Alan Dibley’s Mitee Mouse I and Mitee Mouse II、Dave Otten’s Mitee Mouse I and Mitee Mouse II，如图 1-3 所示。

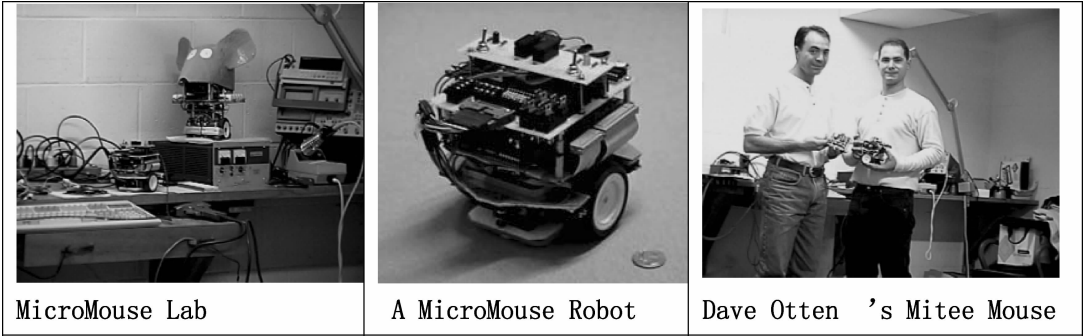


图 1-3 最著名的迷宫机器人

1980 年，欧洲各国效仿美国，每年在 Euromicro（欧洲微型计算机学会）年会会场举办类似的比赛。1980 年，欧洲人在伦敦举行了第一届欧洲迷宫机器人大赛，18 个迷宫机器人没有到达终点。从日本科学基金会派去的参观者把竞赛规则带回了东京，并举办了第一届全日本迷宫机器人竞赛。

1985 年 8 月，来自全欧洲和美国的迷宫机器人参加了在东京举行的第一届世界迷宫机器人竞赛。所用的传感器有红外线传感器、超声波传感器、电荷耦合器件（CCD）传感器等。驱动装置有步进电动机、直流伺服电动机。日本的 Noriko-1 队包揽了所有的比赛奖牌。日本从 1980 年开始举办全日本的机器鼠比赛，每年有大约 100 台机器鼠参加比赛，是知名度较高、基础雄厚、普及性较广的一项赛事。迷宫机器人比赛分三个项目：走迷宫、寻线、

操作。

从第 13 届日本机器人比赛开始设置操作项目，任务是将迷宫路径上事先放置的圆罐反转倒置，在规定的时间内倒置的圆罐越多，得分越多。为此，机器人需要加强作业功能，如装备探测圆罐、翻转倒置圆罐的手抓等。

机器人走迷宫比赛是要求机器人从比赛场地中的起始区出发，在规定的时间内，寻找最好的路线走出迷宫，一般比赛是以机器人的俯视中心离开迷宫场地为准。这就要求不论迷宫形式怎样变化，机器人都能通过不断探测找到终点。一般机器人由行走机构、导向及障碍探测器、控制系统组成。它由蓄电池供电，可完成前进、后退、斜行、横移、转弯等动作。机器人穿越迷宫的整个过程，展示了全方位行走和路线识别技术。

迷宫机器人一旦起动，必须在没有参赛选手的干预下自动控制，机器人必须是由计算机程序控制，而非人工现场控制。机器人在运行过程中可以碰撞或接触墙壁，但是不能故意标记和破坏墙壁。机器人经过比赛场地时不能把任何东西留在后面，并且不能在比赛场地留下任何可以帮助它运行的标记。如果裁判认为机器人故意破坏了比赛场地（包括墙壁），该机器人将被取消参赛成绩或资格。

1.2.4 寻线机器人比赛

从第 7 届迷宫机器人比赛开始设立寻线比赛项目。迷宫机器人在黑底色比赛场地上巡着白色条带标志路线行进（从第 13 届起，路径可实时改变），条带宽度为 1.9cm，路线由若干直线和圆弧形成的复杂曲线，可以近似直角地连接或交叉，如图 1-4 所示。所有的圆弧曲率半径大于 30cm，在曲线拐点处也有标志，且将其邻近长 1m、宽 0.4m 的区域划分为出发区。迷宫机器人的尺寸限制在 25cm × 25cm × 20cm（长 × 宽 × 高）内，重量不限。在起点，允许在 3min 的限定时间内环绕场地试走 3 圈，每圈终了必须在出发区内自动停车后再起动。因此，可以在环绕策略上进行规划。比如第一圈探测直线曲线、熟悉和记忆场地等。第二圈尝试和选择加减速度的合适数值、改变战略等。第三圈应当全速前进。寻线一周所用时间最短的机器人获胜。比赛路线每年变动，机器人需要现场靠标志物的特征实时识别路线，因此对机器人的智能有一定的要求。

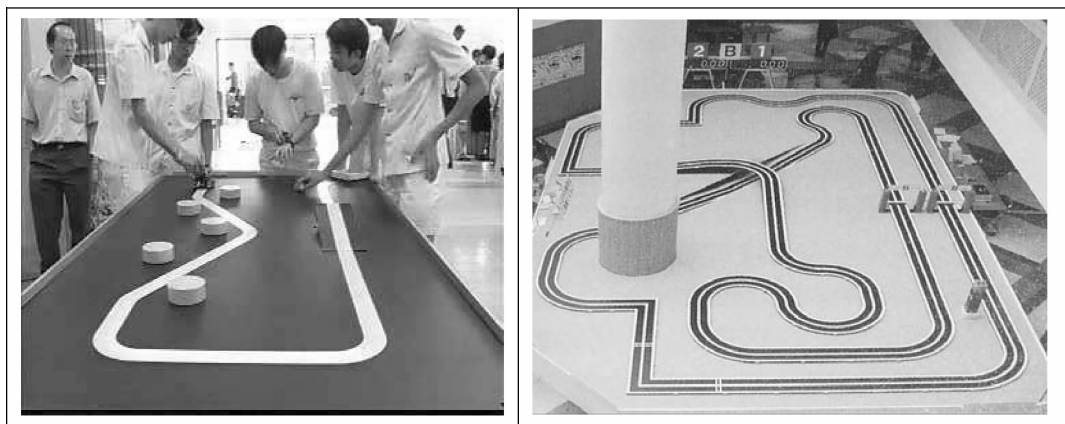


图 1-4 寻线机器人及场地

1.2.5 灭火机器人比赛

灭火机器人需要主动寻找蜡烛火焰，它可以运用类似水、空气、CO₂、Halon（哈龙——属于卤代烷的化学品，主要用于灭火）等物质，或者使用机械方式扑灭蜡烛火焰，但是禁止使用任何危险的或可能破坏比赛场地的方法或物质。比如，不能通过燃放爆竹产生冲击波来使蜡烛熄灭，也不能通过碰倒蜡烛而使蜡烛熄灭。蜡烛在燃烧状态时不允许被撞倒。如果机器人在灭火过程中碰撞蜡烛，机器人的运行仍然有效。如果蜡烛被撞倒，按未完成灭火任务处理。机器人扑灭蜡烛的过程中所产生的杂物，例如水、发酵粉、生奶油等遗留物，将在竞赛间歇被裁判员清理干净。

机器人在试图扑灭火焰前，必须到达距离火焰 30cm 以内。在距离火焰 30cm 的圆上有一条 2.5cm 宽的白线，机器人在扑灭火焰之前，必须有一部分在圆圈内。为了使围绕蜡烛的圆圈不会碰到代表门的线，蜡烛必须离门口足够远。蜡烛将随机地放在比赛场地的任意一个房间里。在机器人所经历的 3 次测试中，蜡烛至少放在两个以上的房间里测试。蜡烛放在一间房子的任意位置，但不能放在走廊上。机器人的前端在进入房间碰到蜡烛前应至少可以移动 33cm。两款灭火机器人如图 1-5 所示。

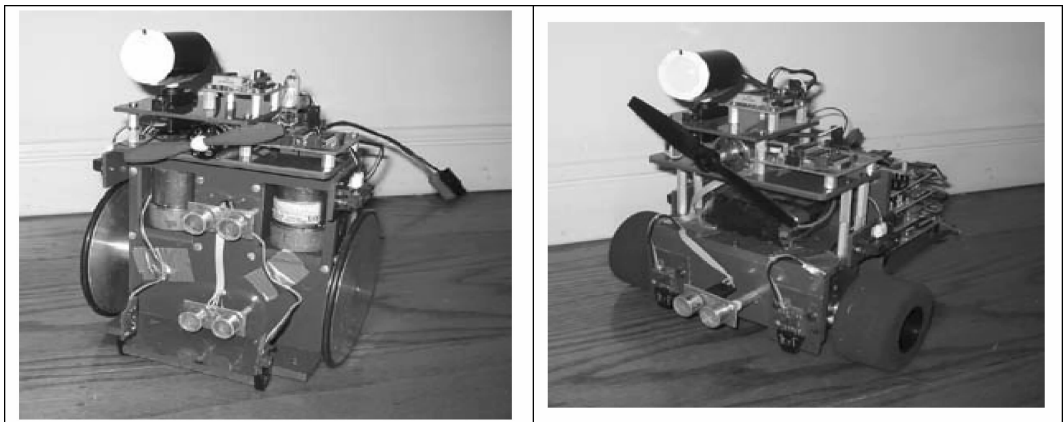


图 1-5 灭火机器人

1.2.6 舞蹈机器人比赛

舞蹈机器人比赛是一项观赏性很强的比赛。机器人舞蹈既具有极强的观赏性和趣味性，更是一个系统化的工程设计。舞蹈机器人涉及了机电一体化技术、检测和传感技术、精密加工和机械密封技术、现代化控制技术和管理技术、光电感应技术、计算机程序控制技术等多个方面，是集成了多学科前沿技术的运动机器人的一种。舞蹈机器人比赛需要多位评委为机器人的表现打分，是一项融技术与艺术于一体的比赛。在世界重大的 RoboCup 比赛中，RoboCup Junior 比赛的一个最吸引人的项目就是舞蹈机器人比赛。另外，在 2001 年由日本发起并举办的“国际机器人竞赛节（RoboFesta）”中，舞蹈机器人比赛也由最初的表演性比赛发展到现在的正式比赛项目。两个比赛的基本规则类同，都需要考虑机器人的技术含量和艺术表现力两个方面。图 1-6 所示为入选美国科技杂志推出“机器人明星榜”的舞蹈机器人“梦露”。

舞蹈机器人的脸庞脱胎于玛丽莲·梦露，长长的裙摆藏起了它的三个轮子。跳舞机器人高 1.6m，能像人一样跳华尔兹。安装在机器人上身的传感器能令其“预测”舞伴的下一个舞步。跳舞机器人不仅仅会跳舞。一位设计者说：“机器人可以通过用户的运动，预测他们的需求，能够为体弱多病的老人或残疾人提供更到位的服务，这些人只要发出口头指令便可以。”

1.2.7 相扑机器人比赛

出于对相扑运动的喜爱，日本于 1990 年 3 月举行了第一届相扑机器人大会，大会举办得相当成功，同年 12 月又举行了第二届相扑机器人大会。自次年起，相扑机器人大会定于每年的 12 月举行。“相扑”是一个体育竞技比赛，目标是通过机器人的短刃接触把一方推出界外，是一项有趣而激动人心的活动，它不仅仅是一个展示机器人技术的好机会，更能体现一个学生应用传感技术对信息的综合处理能力和创新思维。

机器人相扑比赛的规则要求机器人的长和宽不得超过 20cm，重量不得超过 3kg，对机器人的身高没有要求，如图 1-7 所示。机器人的比赛场地是高 5cm、直径为 154cm 的圆形台面。台面上敷以黑色的硬质橡胶，硬质橡胶的边缘处涂有 5cm 宽的白线。这种以黑白两色构成边界线的比赛场地便于相扑机器人利用低成本的光电传感器进行边界识别。相扑机器人使用的传感器有：超声波传感器、触觉传感器等，成本也都不高。正是由于费用不太高，所以发展很快，到 1993 的第 4 届比赛，参赛机器人已超过 1000 台。由于竞技过程是双方机器人“身体”的直接较量，气氛紧张、比赛激烈。



图 1-6 舞蹈机器人“梦露”

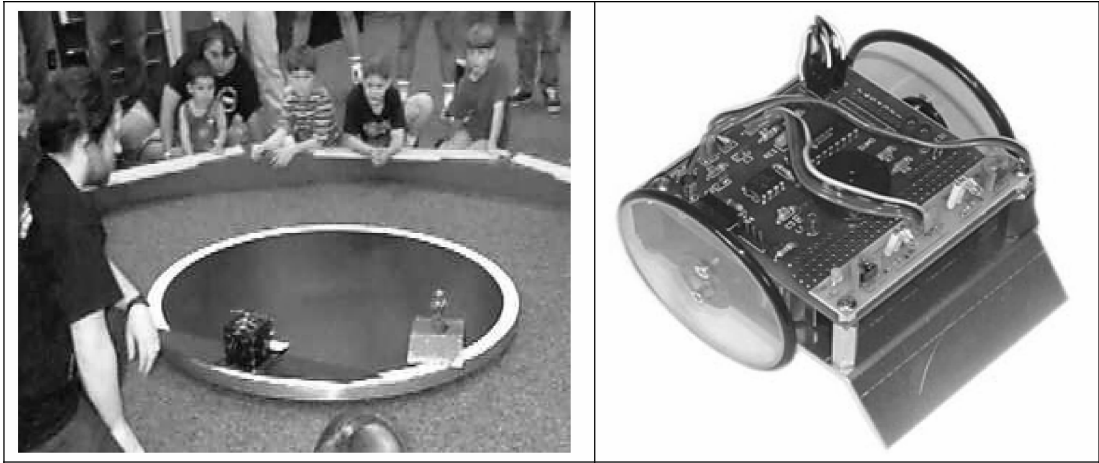


图 1-7 相扑机器人

相扑机器人比赛的规则比较宽松，给参赛者留有较大的发挥空间。比如，为了防止被对手推下赛台，有的相扑机器人采用了必要时可将自己的底部吸附在比赛场地的方法，并靠这

种策略多次赢得了胜利。

1.2.8 搏斗机器人

《机器人大擂台》是由英国 TNN 电视台发起并组织的世界规模最大的科普类竞赛节目，1998 年在英国电视台首次公演。截至 2008 年，它的搏斗系列已经覆盖了全球 27 个国家，包括英国、美国、瑞典、意大利、荷兰等。其中的很多国家现在也发展了自己的机器人大战系列。参赛的机器人选手也由最初的 30 多部机器人发展到每年有 1700 多部机器人参赛，比赛的科技含量和激烈程度逐年增加，观众的参与兴趣也与日俱增。同时，通过比赛现场的讲解，该活动也丰富了世界各地青少年的相关科技知识，加深了他们对机器人技术的兴趣，极大地增强了他们的动手能力。

在该节目中，参赛选手设计并制造无线电控制的格斗机器人以角斗士的方式进行战斗，竞争者依赖于机器人闪电般快速的反应和驱动能力来击败对手。这些精锐的机器人不仅相互之间战斗，并且要与主办方的机器人周旋。图 1-8 所示为第五届《机器人大擂台》冠军，也是进入格斗机器人名人堂的“利箭”与主办方设置的主场机器人“机器骑士”。主场机器人负责在场上巡逻，摧毁并分解进入禁区的参赛机器人。该比赛采取类似足球联赛的形式，组织者将机器人分成若干小组，进行淘汰赛，直至决出一个总冠军。在淘汰赛中机器人在障碍物之间穿行，试图击倒目标。挑战越大，所能得的分越高。胜者直接进入半决赛。在半决赛中，机器人在格斗场上相互格斗，直到其中一个机器人被击败，如果出现平局，则专家委员会将根据四条标准（①破坏程度；②战斗风格；③攻击性；④控制），决定哪个机器人取得胜利。后来又经过改进，障碍物被取消，成为纯机器人格斗比赛，与半决赛规则一样，彻底摧毁对方或是四项评分较高的机器人将获胜。



图 1-8 格斗机器人

1.3 国内的机器人比赛

机器人竞赛是近年来国际上迅速开展起来的一种高科技对抗活动，它涉及人工智能、自动控制、机械电子、通信、传感及机构等多个领域的前沿研究和技术融合。它集高科技、娱乐和比赛于一体。同时机器人技术水平也是一个国家科技水平的重要标志之一。近年来，在

我国大学，机器人作为机械电子学、计算机技术、人工智能等的典型载体被广泛地用来作为工科本科生的讲授课程之一；在中学，模型机器人则逐渐成为素质教育和技能实践的选题之一，各种机器人比赛方兴未艾。

下面就简要介绍一下国内主要的机器人竞赛：

1. 中国青少年电脑机器人比赛

中国青少年电脑机器人比赛是由中国科学技术协会和其他相关部门共同主办，由中国科学技术协会青少年部和省、自治区、直辖市科学技术协会，省及省级青少年电脑机器人竞赛组织单位等共同承办。大赛接受社会各界赞助和协办。

青少年电脑机器人活动，是一项综合多种学科知识和技能的青少年科技活动，孩子们通过计算机编程、工程设计、动手制作与技术构建，结合孩子们的日常观察、积累，去寻求自己最完美的解决方案，发展自己的创造力。中国青少年电脑机器人竞赛是以弘扬科学技术、突显创造与创新、强化团队贡献、培养科学素质、关联当今世界面临的问题与机遇为宗旨。中国青少年电脑机器人活动将现实世界的事物与“动手做”活动相结合，放手让青少年通过活动，去发现解决问题的方法，并获得那些当今科学家和工程师们所面对的机会。同时，它也是中国科学技术协会在 21 世纪青少年科技创新活动中着力创建和打造的一个崭新品牌，是青少年科技竞赛活动探索走向社会化的一种尝试。

赛事包括 FLL（FIRST LEGO League 机器人世锦赛）和 IROC（国际机器人奥林匹克委员会）的比赛项目等。图 1-9 所示为比赛中的一款机器人。

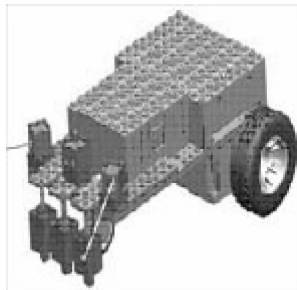


图 1-9 中鸣简易版
积木机器人套件

2. 亚太大学生机器人电视大赛

作为亚洲广播联盟（ABU，简称亚广联）成员国，中央电视台积极参与亚广联组织的亚太地区大学生机器人电视大赛，并依据亚广联当年公布的比赛主题与规则，组织全国大学生机器人电视大赛，并建立了专题网站——全国大学生机器人电视大赛网站（<http://www.robocon.com.cn>）。全国大学生机器人电视大赛由中央电视台主办，教育部、科技部、中国科学院、中国科学技术协会共同协办。该项大赛的目的是培养和开发全国大学生的聪明才智与创新精神，展示当代大学生机器人制作能力与高新技术应用水平。比赛的各项规程由中央电视台本着“是否有利于学生的科技创新，是否有利于发展机器人技术，是否有利于比赛的精彩性与可观赏性，是否有利于机器人大赛的健康发展”的原则，与各高校代表队商议制定。各高校根据当年的比赛主题，并结合本校实际情况报名参赛，一校一队。比赛设冠军 1 名、亚军 1 名、季军 2 名以及优胜奖 12 名，其中冠、亚、季军可获得由组委会颁发的奖金；比赛同时还设立最佳创意、最佳技术、最佳设计、最佳策略和最佳风格等奖项，由大赛评审委员会根据比赛表现评选决定。

ABU Robocon（Asia Broadcast Unit Asia-Pacific Robot Contest，亚广联亚太地区机器人大赛）是由亚广联节目部发起倡导的。此活动的前身是日本广播协会的机器人比赛，该项赛事从 1988 年开始，于 1989 年成为日本广播协会（NHK）每年的赛事，命名为“全日本机器人大赛”。1990 年第一次邀请国外代表队参赛，成为一项国际性比赛，历时十年。中国清华大学、北京航空航天大学上海交通大学都曾参加过该项赛事，其中清华大学曾于 1998 年获得第三名的好成绩。

2000 年 3 月“亚广联亚太地区机器人大赛”第一次筹备会议在日本举行，成立了“亚广联亚太地区机器人大赛”筹备委员会，并选举了六个常任理事机构（中国 CCTV、日本 NHK、韩国 KBS、新加坡 TCS、泰国、印尼），筹备委员会的秘书处设在日本 NHK 的总部。2001 年 2 月份，在马来西亚召开了第一届“亚广联亚太地区机器人大赛”的预备会议。

2002 年 8 月 31 日，由亚广联主办、日本 NHK 承办的第一届亚太地区机器人大赛的总决赛在日本东京成功落下了帷幕。参加此次比赛的有来自包括中国科技大学和澳门大学在内的 19 个国家和地区的 20 支代表队，竞相参与题为“攀登富士山顶”的角逐。中国科技大学在比赛中一路过关斩将，在决赛中由于场地太滑，机器人无法接近得分最多的“富士山顶”，以两分之差惜败给越南胡志明市科技大学队，获得亚军。图 1-10 所示为中国队与越南队的比赛。



图 1-10 中国队与越南队比赛

每届大赛的主题由承办国提出。第一届大赛由日本提出“攀登富士山”，第二届大赛由泰国提出“藤球太空占领者”，第三届大赛由韩国提出“鹊桥相会”，都力图跟自己的民族文化联系起来。第四届大赛由中国主办，提出的主题是：“登长城点圣火”，这也跟我国的历史文化和举办奥运密切相关。机器人大赛和足球比赛不一样，在足球比赛中，观众的喝彩能够增强运动员的斗志，鼓舞士气。但在机器人大赛中，机器人听到喝彩、助威不会激动，反而有可能影响操作机器的人正常发挥。有时决赛上场时机器人发生故障，到最后一刻被淘汰，非常可惜。机器人大赛的主要看点：一是机器人高智能的表演，二是选手高超的战术，三是双方高水平的“智斗”。实际上，机器人比赛，最好看、最精彩、最激烈的往往就在关键的三分钟里。因此，可以说是“精彩三分钟，感动三分钟。”

除了每年的承办国之外，其他国家或地区每年只能有一只参赛队代表国家出征 ABU - Robocon。我国中央电视台为了选拔能够代表中国参赛的代表队，自 2002 年开始举办中央电视台机器人电视大赛（简称 CCTV-Robocon），以选拔队伍参加亚广联亚太地区大学生机器人大赛。

目前,全国大学生机器人电视大赛已经成功举办了 14 届,北京科技大学代表队表现优异,共获得 2 次冠军,4 次亚军,3 次季军。在第四届全国大学生机器人比赛中,北京科技大学获得国内选拔赛的冠军,并与亚军哈尔滨工程大学一起代表中国队,参加在北京举办的主题为“登长城点圣火”的亚太地区大学生机器人大赛。在小组赛的第一轮北京科技大学代表队就与卫冕冠军越南队相遇,经过激烈的争夺,北京科技大学代表队以 13:9 击败对手,使后者未能进入下一轮比赛。之后又一路闯关,在决赛中与日本的东京大学代表队相遇,经过激烈的对抗,最终以 37:42 的比分惜败于对手,取得了我国参加同类比赛的最好成绩。取得 ABU 总决赛亚军的北京科技大学代表队如图 1-11 所示。



图 1-11 2005 年 ABU 总决赛亚军——北京科技大学代表队

3. 中国机器人大赛

中国机器人大赛是由中国自动化学会机器人竞赛工作委员会和科技部高新技术研究发展中心主办的一次全国性竞赛活动,大赛每年举行一次,截止到 2005 年底已举办了五届比赛。大赛是国际 Robo Cup、FIRA 等组织在中国惟一的授权组织和联系通道,由一批全国重点大学专家、教授担任评委,戴汝为、蔡鹤皋等 10 多位院士、20 多位“863”机器人专家担任顾问。

全国机器人大赛展示了我国在研究智能机器人上所做的不懈努力和突出成就,展现了中国高校大学生的创新意识和能力。机器人大赛以科技含量高、学科跨度宽、参与面广和展示性强等特点在国际上有着很强的影响力。它涉及人工智能、图像处理、通信传感、精密机构和自动控制等多领域的前沿研究和技术集成。专家预测,机器人产业在 21 世纪将成为和汽车、计算机并驾齐驱的主干产业。从庞大的工业机器人到微观的纳米机器人,从尖端的仿人机器人到孩子们喜爱的机器人宠物,机器人正日益走近我们的生活,成为人类最亲密的伙伴。目前已经形成了一个国际联盟的人工智能和机器人项目开发目标,被世界各国科研机构 and 众多高等院校所重视。国内的教育和科研机构也日益关注机器人事业的发展,清华大学、中国科技大学等著名高校基本形成了完整的课程体系,对推动高校的科技创新和产学研一体

化产生了积极作用，也为提高我国在机器人领域的国际地位做出了积极贡献。

机器人大赛以足球机器人（RoboCup 和 FIRA）为主要项目。在足球机器人系统开发过程中，不仅要遇到机器人学、机电一体化、通信与计算机技术等，而且涉及图像处理、传感器数据融合、决策与对策、模糊神经网络、人工生命与智能控制等学科的内容。此外，还有机器人灭火、机器人舞蹈、机器人走迷宫、机器人越野、机器人创意等，其中 RoboCup 比赛以其独特的魅力，已成为各类国际机器人竞赛中最具水平和影响力的赛事之一。Robocup 中型组比赛如图 1-12 所示。



图 1-12 Robocup 中型组比赛

4. 中国青少年机器人比赛

中国青少年机器人竞赛活动，是 20 世纪末由中国科学技术协会青少年工作部积极创意并组织开展的一项青少年科技活动，是中国科学技术协会在 21 世纪青少年科技创新活动中着力创建和打造的一个崭新品种。随着电子、信息技术的应用与迅速普及，青少年电脑机器人活动如雨后春笋，在我国已有 30 多个省、自治区、直辖市蓬勃兴起。研究、创意、动手制作机器人活动已成为新世纪中小学校青少年科技创新活动的新亮点和新领域。中国青少年机器人竞赛以丰富多彩、形式多样的机器人探究项目，激发广大青少年的创新思维和实践能力。这项活动最具创新精神、创新意识与创新能力的培养，孩子们通过电脑机器人活动可以进行计算机编程、工程设计、动手制作与技术构建，同时也可以结合他们的日常观察、积累，去寻求自己最完美的解决问题的方案，发展自己的创造力。

自 2001 年起，这项活动已成为我国青少年每年一度的重大科技赛事。目前，全国百余万青少年电脑机器人爱好者通过各种形式投身于电脑机器人的高科技竞技与机器人科学研究和工程研究活动中。

中国科学技术协会已相继在广东、河南、广西等地成功举办了 16 届中国青少年机器人竞赛。第 16 届中国青少年机器人竞赛由中国科学技术协会、中国科学院主办，中国科协青少年科技中心与国防科技大学承办，北京青少年科技中心和怀柔区人民政府协办。共有来自全国 30 个省市自治区、新疆生产建设兵团、香港、澳门及中国儿童中心选派的 497 支参赛队的 1412 名中小学生报名参加机器人综合技能比赛、FLL 机器人工程挑战赛、VEX 机器人工程挑战赛及机器人足球比赛四个类别。第 16 届青少年机器人竞赛是中国科协首次与中国科学院联合主办。国科大作为承办高校，把丰富的科研教育资源融入到竞赛活动中。赛会期间，四场一线科学家的科普报告和“两弹一星”纪念馆等特色参观活动，让参赛学生和辅导教练了解到不同科研领域的研究历程和前沿方向，带给学生和教练们很多人生和治学的启迪。竞赛中涌现了很多创意独特的机器人作品和多样化的解决方案。这也说明越来越多的学校、家庭和校外教育机构对科学和工程技术教育越来越重视，促进了学生机器人科技活动水平的提高，极大地丰富了青少年的课外活动，让学生的学习变成了充满碰撞、游戏、尝试和反思

的过程。

机器人技术综合了多学科的发展成果,代表了高新技术的发展前沿及其在人类生活应用领域的不断扩大。通过组织丰富多彩的青少年机器人竞赛,必将促进我国机器人科技水平的提高与活动的普及,激发广大青少年对科技的兴趣,提高青少年的科学素质,为我国机器人研究和开发储蓄后备人才。

5. “飞思卡尔”杯全国大学生智能车竞赛

作为全球最大的汽车电子半导体供应商,飞思卡尔半导体公司一直致力于为汽车电子系统提供全范围应用的单片机、模拟器件和传感器等器件产品和解决方案。汽车电子应用已涵盖了从汽车电子控制装置到车载汽车电子装置的几乎所有系统。电子控制即通过电子装置控制汽车发动机、底盘、车身、制动防抱死及动力转向系统等,而车载汽车电子装置包括汽车信息娱乐系统、导航系统、汽车音响及车载通信系统等。

在这样的行业背景下,教育部为了加强大学生实践、创新能力和团队精神,培养面向未来的科技尖端人才,在已举办的全国数学建模、电子设计、机械设计、结构设计等四大专业竞赛的基础上,设立了第五项大学生设计竞赛——全国大学生智能汽车竞赛。

“飞思卡尔”杯全国大学生智能汽车竞赛由高等学校自动化专业教学指导分委员会主办,飞思卡尔半导体公司协办,采用邀请赛方式。比赛以迅猛发展的汽车电子为背景,涵盖了控制、模式识别、传感技术、电子、电气、计算机、机械等方面的技术,是多个学科交叉的科技创意性比赛,智能车集所有机器人技术为一身是一项重要的机器人赛事。旨在培养大学生对知识的把握和创新能力,以及从事科学研究的能力。

比赛以飞思卡尔汽车用 MC9S12DG128 16 位微控制器作为智能模型车的主控芯片。组委会提供统一智能车竞赛车模、飞思卡尔的 HCS12 单片机开发板、CodeWarrior 开发软件和在线调试工具等。要求参赛队伍在车模平台基础上,自主构思控制方案及系统设计,包括传感器信号采集处理、控制算法及执行、动力电动机驱动、转向舵机控制等,完成智能车工程制作及调试。制作出一个能够自主识别路线的智能车,在专门设计的跑道上自动识别道路行驶。比赛名次将主要取决于赛车单圈最短时间。

第一届(2006年):承办:清华大学

特等奖:清华大学

第二届(2007年):承办:上海交通大学

特等奖:上海交通大学

第三届(2008年):承办:东北大学

摄像头组 冠、亚、季军:东北大学、北京科技大学、上海交通大学

光电组 冠、亚、季军:武汉科技大学、北京科技大学、东北大学

第四届(2009年):承办:北京科技大学

摄像头组 冠、亚、季军:北京科技大学、上海交通大学、上海大学

光电组 冠、亚、季军:北京科技大学、清华大学、杭州电子科技大学

第五届(2010年):承办:杭州电子科技大学

摄像头组 冠、亚、季军:北京科技大学、杭州电子科技大学信息工程学院、南京师范大学

光电组 冠、亚、季军:杭州电子科技大学、杭州电子科技大学信息工程学院、乐山师

范学院

电磁组 冠、亚、季军：广东技术师范学院、清华大学、杭州电子科技大学

第六届（2011 年）：承办：西北工业大学

摄像头组 冠、亚、季军：湖南大学、北京科技大学、山东大学

光电组 冠、亚、季军：西北工业大学、电子科技大学、乐山师范学院

电磁组 冠、亚、季军：杭州电子科技大学、北京科技大学、西北工业大学

第七届（2012 年）：承办：南京师范大学

摄像头组 冠、亚、季军：北京科技大学、常熟理工学院、电子科技大学

光电组 冠、亚、季军：北京科技大学、山东大学、乐山师范学院

电磁组 冠、亚、季军：中南民族大学、浙江大学、华中科技大学

第八届（2013 年）：承办：哈尔滨工业大学

摄像头组 冠、亚、季军：北京科技大学、武汉科技大学、西安交通大学

光电组 冠、亚、季军：北京科技大学、厦门大学、厦门大学嘉庚学院

电磁组 冠、亚、季军：电子科技大学、北京科技大学、东北大学秦皇岛分校

第九届（2014 年）：承办：电子科技大学

摄像头组 冠、亚、季军：南京师范大学、北京科技大学、电子科技大学

光电组 冠、亚、季军：北京科技大学、中南民族大学、华中科技大学

电磁组 冠、亚、季军：北京科技大学、浙江大学、电子科技大学

第十届（2015 年）：承办：山东大学

摄像头组 冠、亚、季军：厦门大学、北京科技大学、山东大学

光电组 冠、亚、季军：重庆大学、中南大学、北京科技大学

电磁组 冠、亚、季军：北京科技大学、天津大学、中南民族大学

1.4 竞赛机器人平台的主要功能

竞赛机器人是人工智能技术、人工心理、机器人学、机电一体化技术、通信与计算机技术、视觉与传感器技术、智能控制与决策、多智能体系统实时模式识别与行为系统、智能体结构设计、实时规划和推理、基于网络的三维图形交互、多智能体技术（Multi-Agent）、图像识别技术的综合应用；在现实中，竞赛机器人的组织者始终奉行研究与教育相结合的根本宗旨。比赛与学术研究的巧妙结合更激发了青年学生的强烈兴趣，通过比赛培养了青年学生严谨的科学研究态度和良好的技能，为普及科技、提高学生兴趣、加强我国科技实力做出了一定贡献。

通过对竞赛机器人平台技术进行细致研究，制作了一个竞赛机器人试验平台，重点是设计了平台的共轴两轮独立驱动的合理化结构，合理布置了红外发射接收传感器、色差传感器，大大增强了试验平台的功能，提高了传感器的抗干扰能力，提高了传感器检测距离的精度，降低了机器人的功耗。基于这个竞赛机器人平台，可以很容易地组装起迷宫机器人、灭火机器人、寻线机器人、滑雪机器人、足球机器人等，可以快速组队参加比赛，同时减少了试验时间，减少了寻找合适的配件的时间，把重点放在解决特定任务的算法上，大大地提高了工作效率。

同时基于这个平台，可以很容易地开展人工智能各种算法的试验、单片机试验、电动机驱动试验、传感器的试验、C 语言编程的试验等，可以说也是一个很好的综合试验平台，适合于大学自动化专业、计算机专业、电子信息等专业的各类试验。该机器人平台也是一款适合于中小学开展科普活动的有效教具，通过竞赛机器人平台可以完成许多有创意的功能及比赛，如会走 S 形的迷宫机器人、追人机器人、追光机器人、滑雪机器人、救援机器人等。

竞赛机器人平台技术主要涉及了人工智能、自动控制、机械电子、通信、传感及机构等多个领域的前沿研究和技术融合。这里以不同的竞赛机器人为例，分别来介绍其中涉及的支撑技术，见表 1-1。

表 1-1 各种竞赛机器人的主要支撑技术

	类 型	主要支撑技术
1	RoboCup 竞赛	分布式人工智能技术、智能机器人技术、机器人学、机电一体化技术 通信与计算机技术、视觉与传感器技术、智能控制与决策、多智能体系统、实时模式识别与行为系统、智能体结构设计、实时规划和推理、基于网络的三维图形交互、多智能体技术、图像识别技术
2	FIRA 竞赛	机器人足球竞赛的理论研究、机器人足球相关的视觉系统、运动规划、动作设计、策略选择 视频采集/处理、电子和计算机技术、检测及传感器技术、多智能体技术、单片机技术、人工智能技术、神经网络技术
3	ABU 竞赛	多电动机控制技术、人体运动学、伺服电动机控制技术、机电一体化技术 检测及传感器技术、单片机技术、人工智能技术、电子和计算机技术
4	迷宫机器人	伺服电机控制技术、机电一体化技术、检测及传感器技术、单片机技术、人工智能技术、神经网络技术、电子和计算机技术
5	寻线机器人竞赛	伺服电动机控制技术、机电一体化技术、检测及传感器技术、单片机技术、人工智能技术
6	灭火机器人竞赛	伺服电动机控制技术、机电一体化技术、检测及传感器技术、单片机技术、人工智能技术
7	舞蹈机器人竞赛	多电动机控制技术、人体运动学、伺服电动机控制技术、机电一体化技术 检测及传感器技术、人工智能技术
8	相扑机器人竞赛	伺服电动机控制技术、机电一体化技术、检测及传感器技术、单片机技术、人工智能技术、电子和计算机技术

参 考 文 献

[1] 计时鸣, 黄希欢. 工业机器人技术的发展与应用综述 [J]. 机电工程, 2015, 01: 1-13.

[2] 王志良. 人工心理学——关于更接近人脑工作模式的科学 [J]. 北京科技大学学报, 2005 (5): 478-481.

[3] 森正弘, 铃木泰博, 等. 机器人竞赛指南 [M]. 北京: 科学出版社, 2002.

- [4] 国家 863 计划智能机器人专家组. 机器人博览 [M]. 北京: 中国科学技术出版社, 2001.
- [5] 森山和道 (サイエンスライター). ドラえもんザロボット誕生 [J]. ROBOCON Magazine 2004 (4).
- [6] 神永拓. ゼロからはじめるヒューマノイドロボット [J]. ROBOCON Magazine 2003 (12).
- [7] 刘召, 陈恳, 刘莉, 等. 清华大学 MOS 队仿人足球机器人硬件系统的设计与实现 [C]//2005 全国先进制造装备与机器人技术高峰论坛论文集, 2005.
- [8] 刘进长, 辛健成. 机器人世界 [M]. 郑州: 河南科学技术出版社, 2000.

第 2 章 竞赛机器人的控制器

2.1 引言

从世界上第一台遥控机械手的诞生至今已有 50 年了，在这短短的几年里，伴随着计算机、自动控制理论的发展和工业生产的需要及相关技术的进步，机器人的发展已经历了三代：

- 1) 可编程的示教再现型机器人；
- 2) 基于传感器控制具有一定自主能力的机器人；
- 3) 智能机器人。

简单地说，所谓机器人的控制器就是机器人的大脑，就像人的大脑支配人的四肢来完成动作一样，机器人的控制器控制机器人的各个部分来实现机器人的功能。作为机器人的核心部分，机器人控制器是影响机器人性能的关键部分之一。它从一定程度上影响着机器人的发展。目前，由于人工智能、计算机科学、传感器技术及其他相关学科的长足进步，使得机器人的研究在高水平上进行，同时也对机器人控制器的性能提出更高的要求。

对于不同类型的机器人，如竞赛机器人与工业机器人，控制系统的综合方法就有较大差别，控制器的设计方案也不一样。这里仅讨论竞赛机器人控制器问题。

2.2 机器人控制器类型

机器人控制器是根据指令以及传感信息控制机器人完成一定的动作或作业任务的装置，它是机器人的心脏，决定了机器人性能的优劣。

从机器人控制算法的处理方式来看，控制器可分为串行、并行两种结构类型。

2.2.1 串行处理结构

所谓的串行处理结构是指机器人的控制算法是由串行机来处理。对于这种类型的控制器，从计算机结构、控制方式来划分，又可分为以下几种：

1. 单 CPU 结构、集中控制方式

用一台功能较强的计算机实现全部控制功能。在早期的机器人中，如 Hero-I、Robot-I 等，就采用这种结构，但控制过程中需要许多计算（如坐标变换），因此这种控制结构速度较慢，实时性差、难以扩展，其构成框图如图 2-1 所示。

2. 两级 CPU 结构、主从式控制方式

一级 CPU 为主机，担当系统管理、机器人语言编译和人机接口功能，同时也利用它的运算能力完成坐标变换、轨迹插补，并定时地把运算结果作为关节运动的增量送到公用内存，供二级 CPU 读取；二级 CPU 完成全部关节位置数字控制。这类系统的两个 CPU 总线之

间基本没有联系，仅通过公用内存交换数据，是一个松耦合的关系。对采用更多的 CPU 进一步分散功能是很困难的。日本于 20 世纪 70 年代生产的 Motoman 机器人（5 关节，直流电动机驱动）的计算机系统就属于这种主从式控制方式，其构成框图如图 2-2 所示。主从控制方式系统实时性比较好，适用于高精度、高速度控制，但其系统扩展性较差，维修困难。

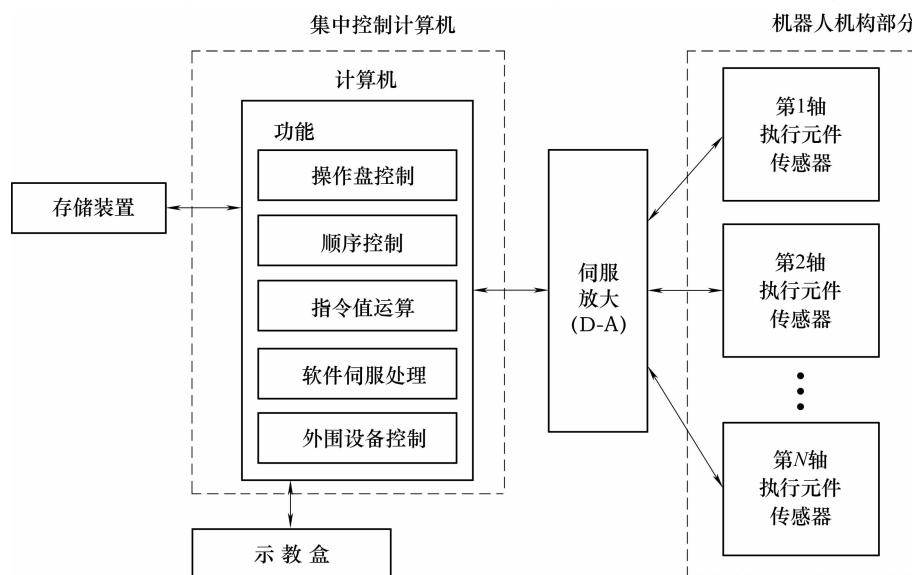


图 2-1 集中控制方式结构框图

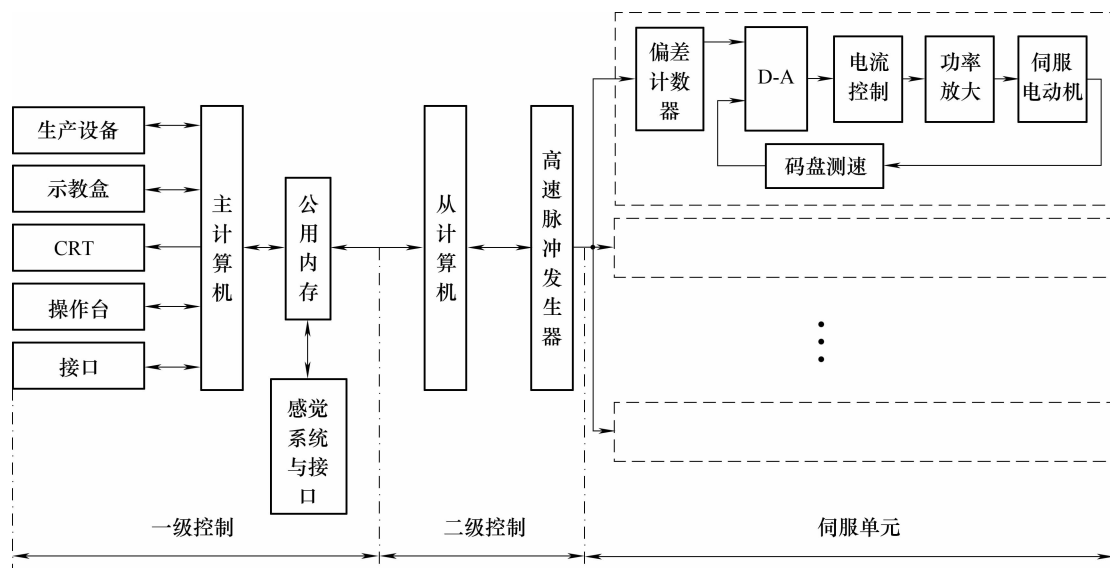


图 2-2 主从控制方式结构框图

3. 多 CPU 结构、分布式控制方式

目前，普遍采用这种上、下位机两级分布式结构，上位机负责整个系统管理及运动学计算、轨迹规划等。下位机由多 CPU 组成，每个 CPU 控制一个关节运动，这些 CPU 和上位机联系是通过总线形式的紧耦合。这种结构的控制器工作速度和控制性能明显提高。但这些多

CPU 系统共有的特征都是针对具体问题而采用的功能分布式结构，即每个处理器承担固定任务。各模式之间可以是主从关系，也可以是平等关系。这种方式实时性好，易于实现高速、高精度，易于扩展，可实现智能控制，其结构如图 2-3 所示。目前世界上大多数商品化机器人控制器都是这种结构。

以上几种类型的控制器都是采用串行机来计算机器人控制算法。它们存在一个共同的弱点：计算负担重、实时性差。所以大多采用离线规划和前馈补偿解耦等方法来减轻实时控制中的计算负担。当机器人在运行中受到干扰时，其性能将受到影响，更难以保证高速运动中所要求的精度指标。

由于机器人控制算法的复杂性以及机器人控制性能的亟待提高，许多学者从建模、算法等多方面进行了减少计算量的努力，但仍难以在串行结构控制器上满足实时计算的要求。因此，必须从控制器本身寻求解决办法。方法之一是选用高档次微机或小型机；另一种方法就是采用多处理器作并行计算，提高控制器的计算能力。

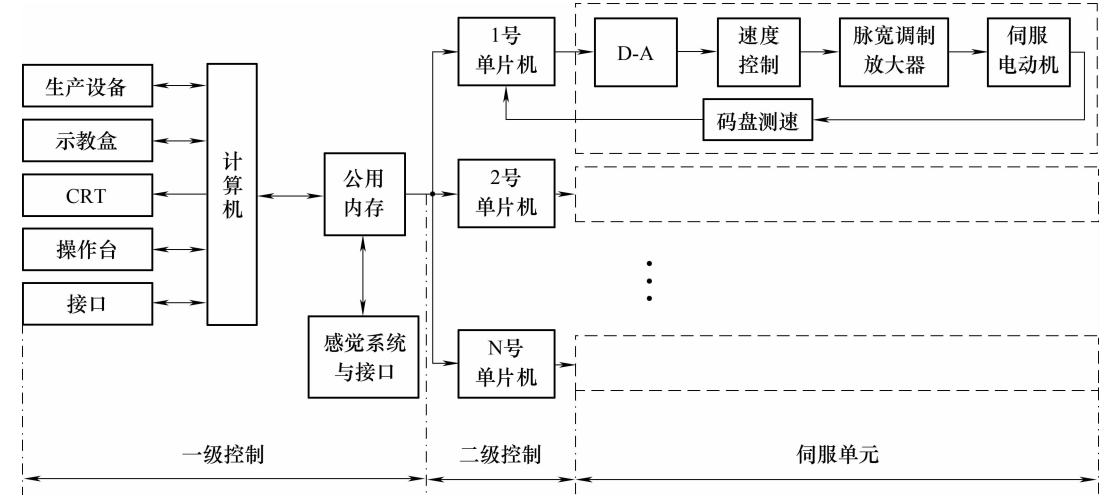


图 2-3 分布式控制方式结构框图

2.2.2 并行处理结构

并行处理技术是提高计算速度的一个重要而有效的手段，能满足机器人控制的实时性要求。关于机器人控制器并行处理技术，人们研究较多的是机器人运动学和动力学的并行算法及其实现。1982 年，J. Y. S. Luh 首次提出机器人动力学并行处理问题，这是因为关节型机器人的动力学方程是一组非线性强耦合的二阶微分方程，计算十分复杂。提高机器人动力学算法计算速度也为实现复杂的控制算法（如计算力矩法、非线性前馈法、自适应控制法等）打下基础。开发并行算法的途径之一就是改造串行算法，使之并行化，然后将算法映射到并行结构。一般有两种方式：一是考虑给定的并行处理器结构，根据处理器结构所支持的计算模型，开发算法的并行性；二是首先开发算法的并行性，然后设计支持该算法的并行处理器结构，以达到最佳并行效率。

构造并行处理结构的机器人控制器的计算机系统一般采用以下方式：

1. 开发机器人控制专用超大规模集成电路（VLSI）

设计专用超大规模集成电路（VLSI）能充分利用机器人控制算法的并行性，依靠芯片

内的并行体系结构易于解决机器人控制算法中大量出现的计算，能大大提高运动学、动力学方程的计算速度。但由于芯片是根据具体的算法来设计的，当算法改变时，芯片则不能使用，因此采用这种方式构造的控制器不能通用，更不利于系统的维护与开发。

2. 利用有并行处理能力的芯片式计算机（如 Transputer、DSP 等）构成并行处理网络

Transputer 是英国 Inmos 公司研制并生产的一种并行处理用的芯片式计算机。利用 Transputer 芯片的 4 对位串通信的链接对，易于构造不同的拓扑结构，且 Transputer 具有极强的计算能力。利用 Transputer 并行处理器，人们构造了各种机器人并行处理器，如流水线型、树型等。

随着数字信号处理芯片速度的不断提高，高速数字信号处理器（DSP）在信息处理的各个方面得到广泛应用。DSP 以极快的数字运算速度见长，并易于构成并行处理网络。

3. 利用通用的微处理器

利用通用微处理器构成并行处理结构，支持计算，实现复杂控制策略在线实时计算。

从前面的介绍可以知道，串行处理结构的优点是硬件结构比较简单，处理方便，但对于计算量比较大的系统来说负担较重、实时性差；并行处理结构相对来说结构较为复杂，因具有数据的并行处理能力，实时性较好。竞赛机器人平台的功能相对较为简单，要求的运算量也不大，串行处理结构是可以满足要求。

2.3 三种机器人控制器的比较

单片机亦称单片微型电脑或单片微型计算机，国际上统称为微控制器，就是将中央处理单元（CPU）、程序存储器（ROM）、数据存储器（RAM）以及输入输出（I/O）端口等部件集成在一片大规模或超大规模集成电路上的超微型计算机。现代社会中，单片机有着极为广泛的应用领域，如自动控制、智能仪表、数据采集系统、外部设备控制系统、家用电器等。在竞赛机器人的控制系统中，就是以单片机为核心的。

8 位单片机由于内部构造简单、体积小、成本低廉，在一些较简单的控制器中应用很广。由于 8 位单片机种类繁多，这里仅将常用的几种单片机在性能上作一个简单的比较。

2.3.1 51 系列单片机

应用最广泛的 8 位单片机首推 Intel 的 51 系列，由于产品硬件结构合理，指令系统规范，加之生产历史“悠久”，有先入为主的优势。世界有许多著名的芯片公司都购买了 51 芯片的核心专利技术，并在其基础上进行性能上的扩充，使得芯片得到进一步的完善，形成了一个庞大的体系，直到现在仍在不断翻新，把单片机世界炒得沸沸扬扬。有人推测，51 芯片可能最终形成事实上的标准微控制器（MCU）芯片。

51 系列优点之一是它从内部的硬件到软件有一套完整的按位操作系统，称作位处理器或布尔处理器。它的处理对象不是字或字节而是位。它不光能对片内某些特殊功能寄存器的某位进行处理，如传送、置位、清零、测试等，还能进行位的逻辑运算，其功能十分完备，使用起来得心应手。虽然其他种类的单片机也具有位处理功能，但能进行位逻辑运算的实属少见。51 系列单片机在片内 RAM 区间还特别开辟了一个双重功能的地址区间，有 16 字节，单元地址为 20H ~ 2FH，它既可作字节处理，也可作位处理（作位处理时，合 128 位，相应

位地址为 00H ~ 7FH)，使用极为灵活。这一功能无疑给使用者提供了极大的方便，因为一个较复杂的程序在运行过程中会遇到很多分支，因而需建立很多标志位，在运行过程中，需要对有关的标志位进行置位、清零或检测，以确定程序的运行方向。而实施这一处理（包括前面所有的位功能），只需用一条位操作指令即可。

【例 2-1】 如对 21H 的第 0 位（相应位地址为 08H）置位，只需用一条位指令 SETB08H，对周围的其他位不会产生影响。

有的单片机并不能直接对 RAM 中的位进行操作，如 AVR 系列单片机中，若想对 RAM 中的某位置位时，必须通过状态寄存器 SREG 的 T 位进行中转。

【例 2-2】 如对 RAM 中的 R0 寄存器的第 4 位置位，则：

BSET6；状态寄存器 T 置位

BLD R0, 4；将 T 位复制到 R0 的第 4 位

显然，AVR 系列单片机比 51 系列单片机要复杂。

51 系列单片机的另一个优点是乘法和除法指令，这给编程也带来了便利。8 位除以 8 位的除法指令，商为 8 位，精度嫌不够，用得不多。而 8 位乘 8 位的乘法指令，其乘积为 16 位，精度还是能满足要求的，用得较多。作乘法时，只需一条指令就行了，即 MPLAB（两个乘数分别在累加器 A 和寄存器 B 中。积的低字节在累加器 A 中，高位字节在寄存器 B 中）。很多 8 位单片机都不具备乘法功能，作乘法时还得编一段子程序调用，十分不便。

在 51 系列单片机中，还有一条二进制转换成十进制调整指令 DA，能将二进制变为 BCD 码，这对于十进制的计量十分方便。而在其他的单片机中，则也需调用专用的子程序才行。

Intel 公司 51 系列单片机的典型产品是 8051，片内有 4KB 的一次性可编程（OTP）存储器。Atmel 公司就将其改为电可改写的闪速（Flash）存储器，允许改写 1000 次以上，这给编程和调试带来极大的便利，其 AT89C51、AT89C52 等产品成为了当今最流行的 8 位单片机。

51 系列单片机的 I/O 引脚的设置和使用非常简单，当该引脚作为输入引脚使用时，只需将该引脚设置为高电平（复位时，各 I/O 引脚均置高电平）。当该引脚作为输出引脚使用时，则为高电平或低电平均可。低电平时，吸入电流可达 20mA，具有一定的驱动能力；而高电平时，输出电流仅数十微安甚至更小（电流实际上是由引脚的上拉电流形成的），基本上没有驱动能力。其原因是高电平时，该引脚也同时作为输入引脚使用，而输入引脚必须具有高的输入阻抗，因而上拉的电流必须很小才行。作为输出引脚使用，欲进行高电平驱动时，得利用外电路来实现（见图 2-4），I/O 引脚不通，电流经 R 驱动发光二极管（LED）发光；低电平时，I/O 引脚导通，电流由该引脚入地，LED 熄灭（I/O 引脚导通时对地的电压降小于 1V，LED 的阈值为 1.5 ~ 1.8V）。

51 系列单片机 I/O 引脚使用简单，但高电平时无输出能力，可谓有利有弊。故其他系列的单片机（如 PIC 系列、AVR 系列等）对 I/O 端口进行了改进，增加了方向寄存器，以确定输入或输出，但使用也变得复杂。

一些简装的 51 系列单片机产品也相应出现，如 Atmel 公司的 AT89C1051、AT89C2051、AT89C4051 等（闪速存储

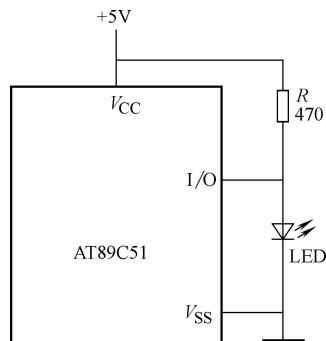


图 2-4 I/O 脚的高电平输出电路

器分别为 1KB、2KB、4KB 等，但不能外接数据存储器），指令系统与 AT89C51 完全兼容，但引脚均为 20 个，不光体积小，而且价格低廉，这使得其他的公司竞相仿造。

不过，原 51 系列单片机也有许多值得改进之处，例如运行速度过慢等。当晶体振荡频率为 12MHz 时，机器周期达 $1\mu\text{s}$ ，显然适应不了现代高速运行的需要。华邦（Winbond）公司生产的单片机产品型号有 W77 系列和 W78 系列。W78 系列与 AT89C 系列完全兼容。W77 系列为增强型，对原有的 8051 的时序作了改进，每个机器周期从 12 个时钟周期改为 4 个时钟周期，使速度提高了三倍，同时，晶体振荡频率最高可达 40MHz。W77 系列还增加了看门狗（Watchdog）定时器、两组通用异步收发器（UART）、两组数据指针（DPTR）、在线系统可编程（ISP）等多种功能。

特别是两组数据指针，能给编程带来很大的便利。在 51 系列单片机中，数据指针（DPTR）是片内与片外的数据存储器打交道的主要途径（由片外数据存储器读入片内累加器 A 或由片内累加器 A 写入片外数据存储器），也是程序存储器与累加器 A 之间的数据传送的必由之路。由于频繁的交换数据，特别是数据块的搬运和比较，数据指针非常吃紧，它需要不断地实施现场保护与还原，不光编程变得复杂，而且运行速度也减慢。而当采用两组数据指针时，可以各负其责，互不相扰，轻松地完成上述过程。两组数据指针的选取取决于特殊功能寄存器 AUXR1 的第 D0 位 DPS。当 DPS 为 0 时，选中数据指针 DPTR0（复位时 DPS 也为 0）；DPS 为 1 时，选中数据指针 DPTR1。DPS 位不能位寻址，故不能进行布尔操作，但由于 AUXR1 的 D1 位被强制为逻辑“0”，不可能发生由 D0 位向 D1 位的进位，因而可以通过对 AUXR1 进行增 1 来使 D0 位由 0 变为 1 或由 1 变为 0，从而达到两组数据指针的快速切换的目的。

ISP 功能能实现在系统可编程，可以省去通用的编程器，单片机在用户板上即可下载和烧录用户程序，而无需将单片机从生产好的产品上取下。未定型的程序还可以边生产边完善，加快了产品的开发速度，减少了新产品因软件缺陷带来的风险。由于可以将程序下载并观看运行结果，故也可以不用仿真器。

单片机的提速运行、两组 DPTR 及 ISP 功能并非是 W77 系列所特有的，一些新型的 51 系列产品大都有该功能，如 Philips 公司的 51LPC 系列、AT89 系列中的某些型号、STC89C 系列等。有的单片机还附有 A/D 转换、D/A 转换、片内数据存储器（EEPROM）、PWM 输出、I²C 总线、上电复位检测、欠电压复位检测等，这些新系列的单片机，它们都兼容 8051 的指令系统。增强功能的实现，大都是由片内新增的特殊功能寄存器来进行设置，这些寄存器被安排在片内特殊功能寄存器区间（80 ~ FFH）的预留地址上。

比较有代表性的产品还有 STC89C51RC、C8051F331/330 等。可以这么说，新的 51 系列单片机产品几乎可以涵盖所有新的功能。由于新型号的芯片种类太多，此处不可能一一列举，读者可根据使用的需求查阅相关的资料。

2.3.2 PIC 系列单片机

PIC 系列单片机是美国微芯公司（Microship）的产品，是当前市场份额增长最快的单片机之一。CPU 采用精简指令集计算机（RISC）结构，分别有 33、35、58 条指令（视单片机的级别而定）。而 51 系列单片机有 111 条指令，AVR 单片机有 118 条指令，都比 PIC 系列单片机复杂。采用哈佛（Harvard）双总线结构，运行速度快（指令周期约为 160 ~ 200ns），

它能使程序存储器的访问和数据存储器的访问并行处理,这种指令流水线结构,在一个周期内完成两部分工作:一是执行指令;二是从程序存储器取出下一条指令,这样总的看来每条指令只需一个周期(个别除外),这也是高效率运行的原因之一。此外,它还具有低工作电压、低功耗、驱动能力强等特点。PIC 系列单片机共分三个级别:基本级、中级、高级。其中,又以中级的 PIC16F873 (A)、PIC16F877 (A) 用得最多,本书以这两种单片机为例进行说明。这两种芯片除了引出脚不同外(PIC16F873 (A) 为 28 脚的塑料双列直插式封装(PDIP)或小外廓集成电路(SOIC)封装;PIC16F877 (A) 为 40 脚的 PDIP 或 44 脚的塑料有引线片式载体(PLCC)封装、方形扁平封装(QFP)),其他的差别并不很大。

PIC 系列单片机的 I/O 端口是双向的,其输出为 CMOS 互补推挽输出电路。I/O 引脚增加了用于设置输入或输出状态的方向寄存器($TRIS_n$, 其中 n 对应各端口,如 A、B、C、D、E 等),从而解决了 51 系列 I/O 引脚为高电平时同为输入和输出的状态。当置位 1 时,为输入状态,且不管该引脚呈高电平或低电平,对外均呈高阻状态;置位 0 时,为输出状态,不管该引脚为何种电平,均呈低阻状态,有相当的驱动能力,低电平吸入电流达 25mA,高电平输出电流可达 20mA。相对于 51 系列单片机而言,这是一个很大的优点,它可以直接驱动数码管显示,并且外电路比较简单。它的 A/D 为 10 位,能满足精度要求。具有在线调试及编程(ISP)功能。

该系列单片机的特殊功能寄存器(SFR)并不像 51 系列单片机那样都集中在一个固定的地址区间内(80 ~ FFH),而是分散在 4 个地址区间内,即存储体 0 (Bank0: 00 ~ 7FH)、存储体 1 (Bank1: 80 ~ FFH)、存储体 2 (Bank2: 100 ~ 17FH)、存储体 3 (Bank3: 180 ~ 1FFH)。只有 5 个专用寄存器 PCL、STATUS、FSR、PCLATH、INTCON 在 4 个存储体内同时出现。在编程过程中,少不了要与专用寄存器打交道,得反复地选择对应的存储体,也即对状态寄存器 STATUS 的第 6 位(RP1)和第 5 位(RP0)置位或清零。这多少给编程带来了一些麻烦。对于上述的单片机,它的位指令操作通常限制在存储体 0 区间(00 ~ 7FH)。

数据的传送和逻辑运算基本上都得通过工作寄存器 W (相当于 51 系列单片机的累加器 A)来进行,而 51 系列单片机还可以通过寄存器相互之间直接传送(如:MOV 30H, 20H;将寄存器 20H 的内容直接传送至寄存器 30H 中),因而 PIC 单片机的瓶颈现象比 51 系列单片机还要严重,这在编程中很有感受。

2.3.3 AVR 系列单片机

AVR 系列单片机是 Atmel 公司推出的较为新颖的单片机,其显著的特点为高性能、高速度、低功耗。它取消机器周期,以时钟周期为指令周期,实行流水作业。AVR 系列单片机指令以字为单位,且大部分指令都为单周期指令。而单周期指令既可执行本指令功能,同时完成下一条指令的读取。通常时钟频率用 4 ~ 8MHz,故最短指令执行时间为 250 ~ 125ns。该系列的型号较多,但可用下面三种为代表:AT90S2313 (简装型)、AT90S8515、AT90S8535 (带 A-D 转换)。

通用寄存器一共有 32 个(R0 ~ R31),前 16 个寄存器(R0 ~ R15)都不能直接与立即数打交道,因而通用性有所下降。而在 51 系列单片机中,它所有的通用寄存器(地址为 00 ~ 7FH)均可以直接与立即数打交道,显然要优于前者。

AVR 系列单片机没有类似累加器 A 的结构,它主要是通过 R16 ~ R31 寄存器来实现 A

的功能。在 AVR 系列单片机中，没有像 51 系列单片机那样的数据指针（DPTR），而是由 X（由 R26、R27 组成）、Y（由 R28、R29 组成）、Z（由 R30、R31 组成）三个 16 位寄存器来完成数据指针的功能（相当于有三组 DPTR），而且还能作后增量或先减量等的运行。

在 51 系列单片机中，所有的逻辑运算都必须在 A 中进行；而 AVR 系列单片机中却可以在任两个寄存器之间进行，省去了在 A 中的来回折腾，这些都比 51 系列单片机强。

AVR 系列单片机的专用寄存器集中在 00~3F 地址区间，无须像 PIC 系列单片机那样先进行选存储体的过程，使用起来比 PIC 系列单片机方便。AVR 系列单片机的片内 RAM 的地址区间为 0060~00DF（AT90S2313）和 0060~025F（AT90S8515、AT90S8535），它们占用的是数据空间的地址，这些片内 RAM 仅仅是用来存储数据的，通常不具备通用寄存器的功能。当程序复杂时，通用寄存器 R0~R31 就显得不够用；而 51 系列单片机的通用寄存器多达 128 个（为 AVR 系列单片机的 4 倍），编程时就不会有这种感觉。

AVR 系列单片机的 I/O 引脚类似 PIC 系列单片机，它也有用来控制输入或输出的方向寄存器，在输出状态下，高电平输出的电流在 10mA 左右，低电平吸入电流为 20mA。虽不如 PIC 系列单片机，但比 51 系列单片机强。

以上的三种 AVR 系列单片机型号其引脚与对应的 51 系列单片机兼容，如 AT90S2313 与 51 系列的 AT89C2051 的引脚兼容（PDIP-20 脚），AT90S8515、AT90S8535 与 51 系列的 AT89C51 兼容（PDIP-40 脚）等。

通过对几种单片机性能的比较，并且考虑到 PIC 系列芯片的稳定性能和抗干扰能力较强，最终选用了 Microchip（微芯）公司生产的 PIC16F877（A）芯片作为本竞赛机器人平台的控制系统，下面具体介绍一下该芯片。

2.4 PIC16F877（A）PIC 系列单片机

Microchip 公司推出的 PIC 8 位单片机系列是业内率先采用精简指令集计算机（Reduced Instruction Set Computer, RISC）结构的高性能价格比的嵌入式控制器（Embedded Computer）。其高速度、低工作电压、低功耗、较大的输入输出直接驱动能力、一次性可编程（One Time Program, OTP）、低价位、小体积等，都体现了微控制器工业发展的新趋势。

2.4.1 PIC 系列单片机性能特点

1. 哈佛（Harvard）双总线结构

这样的存储结构是基于哈佛双总线概念，数据和指令传输总线完全分开，以避免典型的普通复杂指令集计算机（CISC）设计中经常出现的瓶颈问题。传统的冯·诺伊曼结构的计算机是在同一个存储空间取指令和数据（即普林斯顿结构），两者不能同时进行，故限制了工作带宽。而在哈佛结构的计算机中，指令和数据空间是完全分开的，一个用于指令，另一个用于数据。由于可以对程序和数据同时进行访问，所以提高了数据吞吐率。正因为 PIC 系列单片机中采用了哈佛双总线结构，所以与常见的微控制器不同的一点是：程序和数据总线可以采用不同的宽度，数据总线都是 8 位的，但低档、中档和高档系列的指令总线位数分别为 12、14 和 16 位。图 2-5 所示是两种不同结构的计算机示意图。

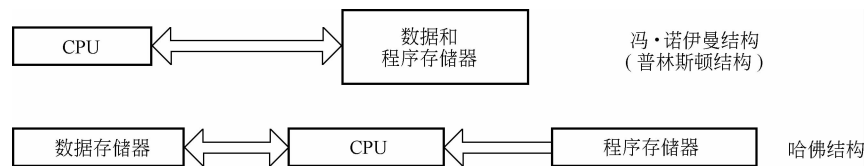


图 2-5 计算机的两种不同结构示意图

2. 指令单字节化

因为数据总线和指令总线是分离的，并且采用了不同的宽度，所以程序存储器（ROM）和数据存储器（RAM）的寻址空间（即地址编码空间）是相互独立的，而且两种存储器宽度也不同。这样设计，不仅可以确保数据的安全性，还能提高运行速度和实现全部指令的单字节化。

3. 类 RISC（RISC-like）结构

这为 8 位单片机市场设立了一种事实上的新的性能标准。为了达到独一无二的高速性能，PIC 单片机采用了小型机设计结构。

先进的类 RISC 结构体现在每一条高效率和功能强大的指令上。三个系列微控制器的指令都是单字的宽字位指令：低档、中档和高档系列的指令位数分别为 12、14 和 16 位，且分别只有 33、35 和 38 条指令，它们向上兼容。其指令系统除了程序分支指令是单字双周期指令外，其他指令都是单字单指令周期。在这些指令中，没有功能交叉的指令，使得所有的指令具有简洁性。单字宽指令提供了软件编码的效率，减少了所需要的程序存储单元，使系统具有最高处理效率和突出性能。另外，由于所用指令数较少和较简洁，编程任务和调试任务相对就比较容易，而且学习和实现都非常快。在相同情况下，PIC 单片机所需要的编码比一般微处理器要少一半，其指令的高效率又可以使得开发时间节约 30%。

4. 寻址方式简单

PIC 系列单片机只有 4 种寻址方式（即寄存器间接寻址、立即数寻址、直接寻址和位寻址），容易掌握。

5. 代码压缩率高

1KB 的存储空间，对于像 51 系列这样的单片机，大约只能存放 600 条指令，而对于 PIC 系列单片机，则能够存放的指令条数多达 1024 条。

6. 运行速度快

由于采用了哈佛总线结构，以及指令的读取和执行采用了流水作业方式，使得运行速度大大提高。

7. 功耗低

PIC 系列单片机的功率消耗极低，其中有些型号，在 4MHz 时钟下工作时耗电不超过 2mA，在睡眠模式下耗电可以达到 1μA 以下。

8. 驱动能力强

I/O 端口驱动负载的能力较强，每个 I/O 引脚吸入和输出电流的最大值可以分别达到 25mA 和 20mA，能够直接驱动发光二极管（LED）、光耦合器或者微型继电器。

9. I²C 和 SPI 串行总线端口

用这两种串行总线技术可以实现芯片间同步串行数据传输。

10. 寻址空间设计简洁

PIC 系列单片机的程序、堆栈、数据三者各自采用互相独立的寻址（或抵制编码），而且前两者的地址安排不需要用户操心。

11. 保密性好

PIC 以保密熔丝来保护代码，用户在烧入代码后熔断熔丝，别人再也无法读出，除非恢复熔丝。目前，PIC 采用熔丝深埋工艺，恢复熔丝的可能性极小。

12. 外界电路简洁

该系列单片机内集成上电复位电路，I/O 端口上拉电路，看门狗定时器等电路，符合真正意义上的“纯单片”应用。

2.4.2 PIC16F87X 单片机的结构与性能特点

本竞赛机器人平台具体选用的单片机型号为 PIC16F877，芯片的外形如图 2-6 所示。

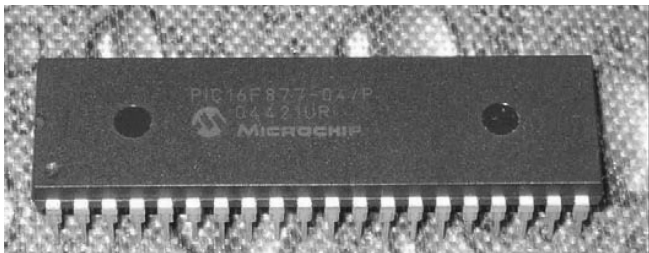


图 2-6 PIC16F877 芯片的外形

PIC16F877 引脚排列如图 2-7 所示。

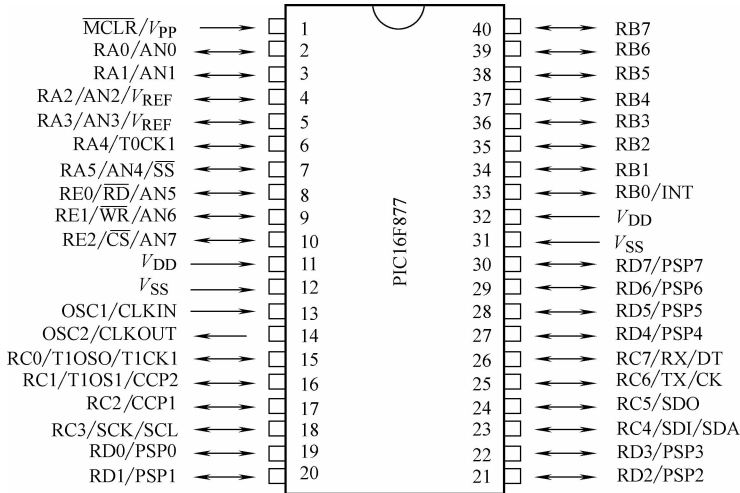


图 2-7 PIC16F877 单片机的引脚排列

PIC16F87X 系列单片机内部机构如图 2-8 所示。

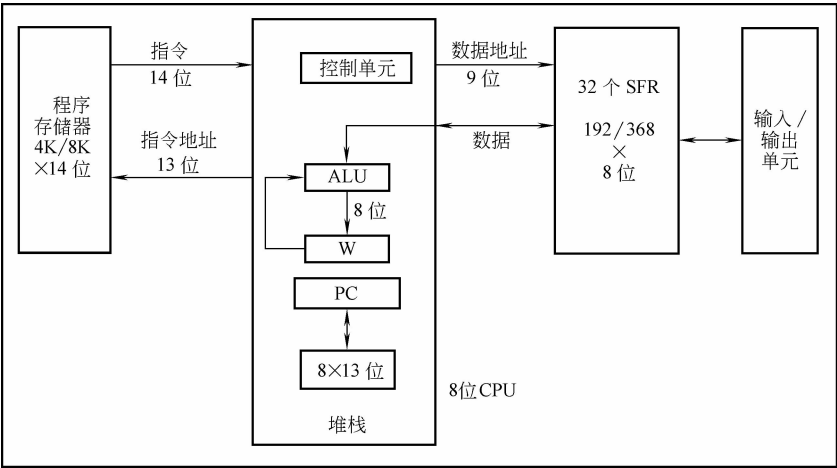


图 2-8 PIC16F87X 系列单片机内部机构

PIC16F87X 系列单片机主要有 4 个不同型号：PIC16F873/874/876/877。

PIC16F87X 系列的最高工作频率可达到 20MHz，具有上电复位（Power-on Reset，POR）和掉电锁定复位（Brown-out Reset）两种重置功能，以及上电定时器和晶体振荡启振定时器。除了一个看门狗定时器之外，另外还有三个定时器和两个 CCP 模块，串行通信模式方面共支持 USART（通用同步/异步收发器）、SPI 和 I²C。PIC16F87X 系列单片机的寻址方式有：直接寻址、间接寻址、立即寻址（常数寻址）和相对寻址。

PIC16F87X 系列单片机性能特点如下：

1. 高速度

由于 PIC 系列单片机所用的单字宽周期指令、哈佛双总线和 RISC 结构，其数据吞吐量最高可达 6MIPS，即每秒钟可执行 600 万条指令，这几乎是其他大多数 8 位单片机速度的 8 倍。

实时执行对时间要求苛刻的应用，例如电动机控制、高速 I/O 或者串行数据位流操作等，PIC 系列单片机可以提供一种新的低成本解决方案，利用 PIC 高性能算法处理能力的实时性可以取代低效率的存储操作和精度不高的查表法。

2. 低功耗

由于采用全静态 CMOS 设计，电源操作能耗很低，但是非常可靠。因为片内集成了上电复位电路（POR）、监视定时器电路和 RC 型振荡器电路等选择，所以就不需要增加价格较高的片外支持的功能元器件，从而降低了系统成本和系统功耗。

3. 用户可选择振荡器

片内集成有带 RC 型振荡器的监视定时器，而且引进了“用户可选择振荡器”的新设计概念，它允许用户在 4 种从 0 ~ 20MHz 的频率范围内选择一种振荡器形式，让芯片执行指令，以优化系统的功耗。这 4 种振荡器方式是：低成本的 RC 型振荡器，XT 型标准的石英晶体振荡器，HS 型高速石英晶体振荡器和 LP 型低功耗低频石英晶体振荡器。

4. 具有程序代码保护功能

可通过对片内 EEPROM 保密设置进行加密。

5. 功能强大的 I/O 性能

可以用来作为高速 I/O 控制器。数据输入线允许有 25mA 的倒灌电流，可以直接驱动 LED。

6. 具有多种功能模块

具有 PWM、A/D 转换、D/A 转换器、内部集成的温度传感器、基准电源、振荡器、多路复用 LCD 驱动器、异步串行通信接口、数据存储器（EEPROM）等。

7. 中断特性

PIC16F87X 系列单片机有中断功能，根据要求可以选用多个不同数目的内部和外部中断源。中断控制可以进行全局控制和分别控制。外部接口的中断优先级用软件确定。可以用几个不同的中断源激活处理器从休眠状态中苏醒。有固定的中断开销时间，同步中断是 3 个周期。用户可以根据需要存储/恢复寄存器。

2.4.3 单片机 C 语言编程

PIC16F877 单片机支持 C 语言编程，称之为 PICC。C 语言是一种源于编写 UNIX 操作系统的语言，它是一种结构化的语言，可产生紧凑代码。C 语言的结构是括号 { } 而不是字和特殊符号。C 语言可以进行许多机器级函数控制而不用汇编语言。

与汇编语言相比，有如下优点：

对单片机的指令系统不要求，仅要求对单片机的存储器结构有初步的了解；寄存器分配、不同寄存器的寻址及数据类型等细节可由编译器管理；程序有规范的结构，可分为不同的函数，这种方式可使程序结构化；具有将可变的选择与特殊操作组合在一起的能力，改善了程序的可读性；关键字及运算函数可用近似人的思维过程方式使用；编程及程序调试时间显著缩短，从而提高效率；提供的库包含许多标准子程序，具有较强的数据处理能力；已编好程序可容易地植入新程序中，因为它具有方便的模块化编程技术；C 源程序的书写格式自由度较高，灵活性很强，有较大的任意性。

C 语言作为一种非常方便的语言而得到广泛的支持，C 语言程序本身不依赖于机器硬件系统，基本不做修改就可以根据单片机的不同较快地移植过来。

PIC 程序的编程要点如下：

1) C 语言是由函数构成的。一个 C 源程序至少包括一个函数（main 函数），也可以包含一个 main 函数和若干其他函数。因此，函数是 C 语言程序的基本单位。被调用的函数可以是编译器提供的库函数，也可以是用户根据需要自己编制设计的函数。

2) 一个函数由两部分组成：函数说明部分。包括函数名、函数类型、函数属性、函数参数名、形式参数（形参）类型，一个函数名后面必须跟一个圆括号，函数参数可以没有，如 main（）函数。如果一个函数有多个大括号，则最外层的一对 {} 为函数的范围。函数体一般包括：变量定义和执行部分，由若干语句组成。当然，在某些情况下也可以没有变量定义部分，甚至可以既无变量定义也无执行部分。

3) 一个 C 语言程序总是从 main 函数开始执行的，而不论 main 函数在整个程序中的位置如何。

4) C 语言程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上。C 语言程序无行号。

5) 每个语句和数据定义的最后必须有一个分号。分号是 C 语句的必要组成部分。分号不可少,即使是程序中最后一个语句也应包含分号。

6) C 语言本身没有输入输出语句。输入和输出的操作是由库函数 `scanf` 和 `printf` 等函数来完成的。C 对输入输出实行“函数行”。

7) 可以用 `/* */` 对 C 程序中的任何部分作注释。一个好的、有使用价值的源程序都应当加上必要的注释,以增加程序的可读性。

2.5 机器人控制器的发展趋势

随着现代科学技术的飞速发展和社会的进步,对机器人的性能提出更高的要求。智能机器人技术的研究已成为机器人领域的主要发展方向,如各种精密装配机器人、力/位置混合控制机器人、多肢体协调控制系统以及先进制造系统中的机器人的研究等。相应地,对机器人控制器的性能也提出了更高的要求。

但是,机器人自诞生以来,特别是工业机器人所采用的控制器基本上都是开发者基于自己的独立结构进行开发的,采用专用计算机、专用机器人语言、专用操作系统、专用微处理器。这样的机器人控制器已不能满足现代工业发展的要求。

从前面提到的两类机器人控制器来看,串行处理结构控制器的结构封闭,功能单一,且计算能力差,难以保证实时控制的要求,所以目前绝大多数商用机器人都是采用单轴 PID 控制,但也难以满足机器人控制的高速、高精度的要求。虽然分布式结构在一定层次上是开放的,可以根据需要增加更多的处理器,以满足传感器处理和通信的需要,但它只是在有限范围内开放。

并行处理结构控制器虽然能从计算速度上有了很大突破,能保证实时控制的需要,但必须看到还存在许多问题。目前的并行处理控制器研究一般集中于机器人运动学、动力学模型的并行处理方面,是基于并行算法和多处理器结构的映射特征来设计的,即通过分解给定任务,得到若干子任务,列出数据相关流程图,实现各子任务在对应处理器上的并行处理。由于并行算法中通信、同步等内在特点,如程序设计不当,则易出现锁死与通信堵塞等现象。

随着机器人控制技术的发展,针对结构封闭的机器人控制器的缺陷,开发“具有开放式结构的模块化、标准化机器人控制器”是当前机器人控制器的一个发展方向。近几年,日本、美国和欧洲一些国家都在开发具有开放式结构的机器人控制器,如日本安川公司基于 PC 开发的具有开放式结构、网络功能的机器人控制器。我国 863 计划中智能机器人主题也已对这方面的研究立项。

开放式结构机器人控制器是指:控制器设计的各个层次对用户开放,用户可以方便地扩展和改进其性能。其主要思想是:

1) 利用基于非封闭式计算机平台的开发系统,如 Sun、SGI、PC's。有效利用标准计算机平台的软、硬件资源为控制器扩展创造条件。

2) 利用标准的操作系统(如 Unix、Vxwork)和标准的控制语言(如 C、C++),从而可以改变各种专用机器人语言并存且互不兼容的局面。

3) 采用标准总线结构,使得为扩展控制器性能而必需的硬件(如各种传感器、I/O 板、

运动控制板), 可以很容易地集成到原系统中。

4) 利用网络通信, 实现资源共享或远程通信。目前, 几乎所有的控制器都没有网络功能, 利用网络通信功能可以提高系统变化的柔性。

我们可以根据上述思想设计具有开放式结构的机器人控制器。而且设计过程中要尽可能做到模块化。模块化是系统设计和建立的一种现代方法, 按模块化方法设计, 系统由多种功能模块组成, 各模块完整而单一。这样建立起来的系统, 不仅性能好、开发周期短而且成本较低。模块化还使系统开放, 易于修改、重构和添加配置功能。

新型的机器人控制器应有以下特色:

(1) 开放式系统结构 采用开放式软件、硬件结构, 可以根据需要方便地扩充功能, 使其适用于不同类型机器人或机器人化自动生产线。

(2) 合理的模块化设计 对硬件来说, 根据系统要求和电气特性, 按模块化设计, 这不仅方便安装和维护, 而且提高了系统的可靠性, 系统结构也更为紧凑。

(3) 有效的任务划分 不同的子任务由不同的功能模块实现, 以利于修改、添加、配置功能。

(4) 实时性、多任务要求 机器人控制器必须能在确定的时间内完成对外部中断的处理, 并且可以使多个任务同时进行。

(5) 网络通信功能 利用网络通信的功能, 以便于实现资源共享或多台机器人协同工作。

(6) 形象直观的人机接口 另外, 机器人控制器中, 运动控制板是必不可少的。由于机器人性能的不同, 对运动控制板的要求也不同。美国 Delta Tau 公司推出的 PMAC (Programmable Multi-axies Controller) 在国内外引起重视。PMAC 是一种功能强大的运动控制器, 它全面地开发了 DSP 技术的强大功能, 为用户提供了很强的功能和很大的灵活性。借助于 Motorola 公司的 DSP56001 数字信号处理器, PMAC 可以同时操纵 1~8 轴, 比起其他运动控制板来说, 有很多可取之处。

由于适用于机器人控制的软、硬件种类繁多和现代技术的飞速发展, 开发一个结构完全开放的标准化机器人控制器存在一定困难, 但应用现有技术, 如工业 PC 良好的开放性、安全性和联网性, 标准的实时多任务操作系统, 标准的总线结构, 标准接口等, 打破现有机器人控制器结构封闭的局面, 开发开放性结构、功能模块化的标准化机器人控制器是完全可行的。

随着机器人技术的发展, 机器人应用领域的不断扩大, 对机器人的性能提出了更高的要求, 因此, 如何有效地将其他领域 (如图像处理、声音识别、最优控制、人工智能等) 的研究成果应用到机器人控制系统的实时操作中, 是一项富有挑战性的研究工作。而具有开放式结构的模块化、标准化机器人控制器的研究无疑对提高机器人性能和自主能力、推动机器人技术的发展具有重大意义。

参 考 文 献

- [1] Microchip Technology Inc., PIC16F87X 数据手册——28/40 脚 8 位 FLASH 单片机 [M]. 刘和平, 黄开长, 严利平, 等, 译. 北京: 北京航空航天大学出版社, 2001.

- [2] 史伟. PIC 系列单片机在蓄电池模拟器中的应用 [J]. 中国电子商情 (基础电子), 2011, 08: 39-41.
- [3] 李学海. PIC 单片机实用教程——基础篇 [M]. 北京: 北京航空航天大学出版社, 2002.
- [4] 刘豹. 现代控制理论 [M]. 2 版. 北京: 机械工业出版社, 2000.
- [5] 王文治, 吕强中, 于常涛. MCS-51 与 PIC 单片机指令系统的比较 [J]. 微处理机, 2002 (4).
- [6] 丁元杰. 单片微机原理及应用 [M]. 北京: 机械工业出版社, 1998.
- [7] 窦振中. PIC 系列单片机原理和程序设计 [M]. 北京: 北京航空航天大学出版社, 1998.

第3章 竞赛机器人的软件开发平台

上一章介绍了竞赛机器人平台的核心单元——控制器，控制器是机器人的“大脑”，负责对外界信息进行处理，并且做出下一步的决策。大脑要工作自然离不开它的载体，同样控制器也需要有一个可以运行承载的环境，这个环境就是控制器的软件开发平台。为了更好地理解什么是软件开发平台，我们看一个例子：在邮局，工作人员把信件按照收信人的地址分类——寄往北京的信件放入一号箱子、寄往天津的信件放入二号箱子、寄往上海的信件放入三号箱子——这个分类的规则由邮局的管理者制定，通常这个规则要用汉字书写，要写在纸上，并经过反复思考修改，最后告知工作人员。同样，机器人也需要一套规则——什么时候前进、什么时候转向、什么时候后退，这套规则由机器人的开发人员制定，通常这套规则是用C语言、汇编语言等编程语言制定的，利用软件开发平台书写程序，并要在软件开发平台上经过反复修改调试，之后，软件开发平台中的编译器将C语言或者汇编语言转换成机器语言，告知控制器。基于这个软件开发平台进行软件编程是竞赛机器人技术开发的主要内容。在这个环境下，才能够进行对硬件的编程操作，才能实现竞赛机器人“大脑”的功能。下面就来介绍一下这个软件开发平台——MPLAB。

3.1 MPLAB 概述

MPLAB 是适用于 Microchip Technology 公司 PIC16/17 系列单片机的、基于 Windows 的、易学易用的集成开发环境（Integrated Development Environment，IDE）。这种集成开发环境可以方便又灵活地编辑、汇编、仿真、开发和调试查错。

3.1.1 MPLAB 集成开发环境的组成

MPLAB 集成开发环境软件集成了下列开发工具：

1. 工程项目管理程序（Project Manager）

工程项目管理程序用于创建工程项目，并用指定的与工程项目有关的文件工作。MPLAB 工程项目管理程序也使操作者能通过单击鼠标建立应用程序，并把代码下载到仿真器或模拟程序中。

通过建立工程项目可以实现符号调试（Symboic Debugging）。工程项目管理程序是 MPLAB 的核心部分。不创建工程项目就不能进行符号调试。要进行源代码符号调试需要进行下列操作：

- 1) 创建工程项目；
- 2) 把源代码文件加入到工程项目之中；
- 3) 汇编或编译源代码；
- 4) 编辑源代码；
- 5) 重建（Rebuild）所有源文件，或编译单个文件；

6) 调试源代码。

2. MPLAB 编辑程序 (MPLAB Editor)

MPLAB 编辑程序用于编写录入和编辑应用程序源文件, 或其他用于 PIC16/17 单片机控制器的文本文件。

3. MPASM 汇编程序 (MPASM Assembler)

MPASM 通用 PIC16/17 单片机汇编程序, 可以提供完善的宏调用功能、条件汇编以及几种不同的源文件与列表文件格式。在使用 MPASM 过程中, 不退出 MPLAB, 便可生成各种目标代码, 以支持 Microchip 开发工具与其他编辑工具。

4. MPLAB-SIM 软件模拟程序 (MPLAB-SIM Software Simulator)

MPLAB-SIM 软件模拟程序可以提供模拟 PIC16/17 单片机的指令执行和输入/输出的环境, 不需要任何硬件, 在 PC 上就可以用软件仿真全系列单片机, 对用户应用系统程序的逻辑进行调试, 查找代码错误。这是一种非实时、非在线的软件模拟器, 对于那些实时性要求不高的程序, 采用这种方法调试可以降低开发成本。

5. PICMASTER 硬件在线仿真器 (PICMASTER Emulator)

PICMASTER 硬件仿真器是高性能的在线仿真器。对于有实时性要求的系统就必须使用这种硬件在线仿真器进行调试。

6. 其他工具

MPLAB IDE 支持 Microchip 开发工具以及其他开发工具, 诸如编程器、编译程序以及其他可用产品。

MPLAB 工具是按功能组织的, 这有助于很容易地找到下拉式选单以及可定制的快捷键并使用它们。MPLAB 工具可以用于调试源程序代码, 在编辑源文件时可以自动查错, 用基于内部寄存器值的断点进行调试, 用模拟程序检查程序流程或用 PICMASTER 硬件在线仿真器进行时序检查, 可以在不同窗口内跟踪变量的值, 并随时可从 MPLAB 在线帮助中快速地找到问题的解决方法。

3.1.2 MPLAB 运用方式

MPLAB 可在三种不同方式下运行:

1. MPLAB-SIM 软件模拟程序方式 (MPLAB-SIM Simulator Mode)

在这种方式下, 提供用于调试应用系统设计源程序的快速工具。

2. 编辑程序方式

不用软件模拟程序或硬件仿真器便可编写、编译/汇编等, 并能从源代码中删除错误。

3. PICMASTER 硬件在线仿真器方式

能用 PICMASTER 硬件在线仿真器实时运行用户应用系统的目标程序, 同时还可以对硬件进行调试。

3.1.3 MPLAB 对硬件与软件的要求

在 Windows 操作环境下, 利用其多任务特性可以同时运行多个程序, 例如 E-mail、电子表格以及 MPLAB IDE 等。因为 MPLAB IDE 具有多处理器仿真能力, 并用客户机程序支持动态数据交换 (Dynamic Data Exchange, DDE), 所以由 MPLAB IDE 所采集的数据可被其他程

序（例如 Microsoft Excel）所共享。

如果要求利用仿真器系统的特性，安装 MPLAB 软件（MPLAB.EXE）所需要的 PC 系统最小配置要求如下：

- 1) 386 或以上档次的处理器的个人计算机，推荐 CPU 使用奔腾处理器；
- 2) 4MB 内存可用，推荐 16MB 以上内存；
- 3) VGA（视频图形阵列）或 Super VGA 显示器；
- 4) Microsoft Windows 3.1 或以上版本。

3.2 MPLAB 的安装和启动

3.2.1 完整 MPLAB 安装

在 MPLAB Setup 开始复制文件之前，它将进行检查，以确保所选驱动器具有可供 MPLAB 安装用的空间。在完整的安装中，MPLAB Setup 将安装以下一些应用程序：

- 1) MPLAB 程序文件（包括编辑程序）；
- 2) MPLAB—SIM；
- 3) 工具（MPASM for Windows）；
- 4) 帮助（Help）。

详细安装过程请参见第 8 章实验一。

3.2.2 MPLAB 的启动

MPLAB 可以用以下几种不同方式启动：

- 1) 用鼠标双击 Microsoft MPLAB 程序组内的图标；
- 2) 用鼠标单击 Windows 中 File→Run 命令；
- 3) 通过其他任何可启动该程序的应用程序；

4) 在 MPLAB 命令行中列出包含 MPLAB 将要打开的文本文件。MPLAB 编辑程序（MPLAB Editor）将打开在命令行上所列出的每一个文件。命令行上给出的文件名允许包括常用的 DOS 通配符。

3.3 MPLAB 的使用

3.3.1 启动 MPLAB IDE

以使用 C 语言为例，用以下任意一种方式启动软件：

- 1) 选择“开始→所有程序→Microchip MPLAB IDE→MPLAB IDE”；
- 2) 双击桌面上的 MPLAB IDE 图标。

MPLAB IDE 桌面显示如图 3-1 所示。

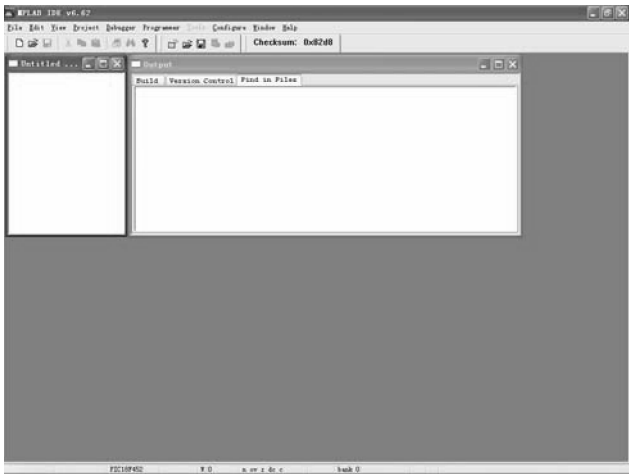


图 3-1 MPLAB IDE 启动桌面

3.3.2 创建源文件

选择 File（文件）→new（新建），出现一个文本编辑窗口，在该窗口中输入 C 语言源程序，如图 3-2 所示。

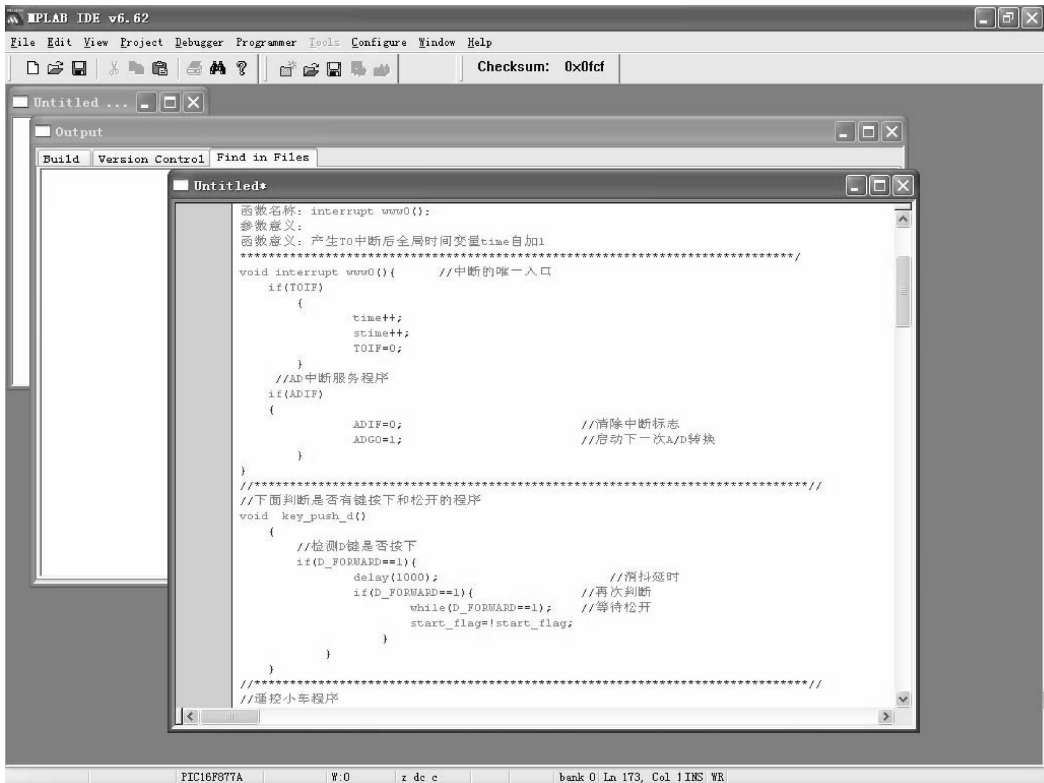


图 3-2 输入 C 语言源程序的窗口

先创建文件夹，名字可以任意取，但是不要有汉字，这里以取名为 MyPrj 为例，然后将

上面输入的源程序保存该文件夹，注意文件的扩展名为 .c，这里将它保存为 exam1.c，如图 3-3 所示。

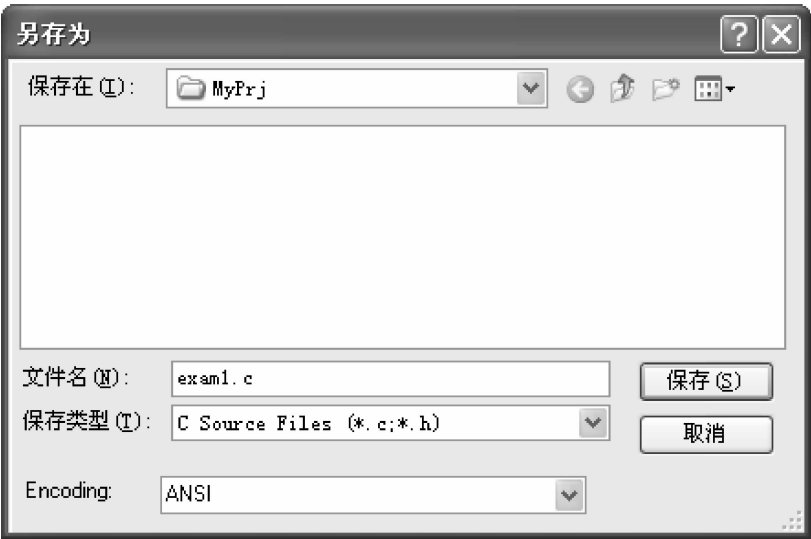


图 3-3 另存为对话框

文件保存后，源程序编辑窗口中的语句被着上了各种颜色，这些颜色能帮助我们更好地阅读源程序，快速发现输入有误的指令，如图 3-4 所示。

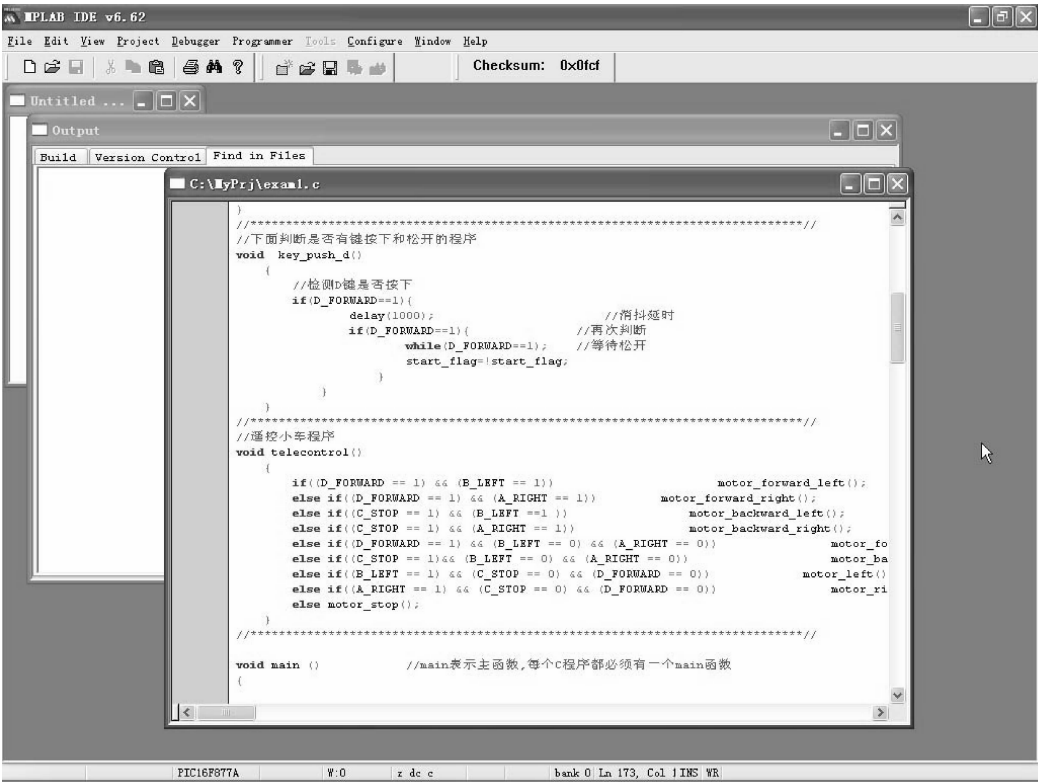


图 3-4 着色后的语句窗口

3.3.3 创建项目

选择 Project(项目)→new(新建)，出现 New Project(新项目)对话框，如图 3-5 所示。

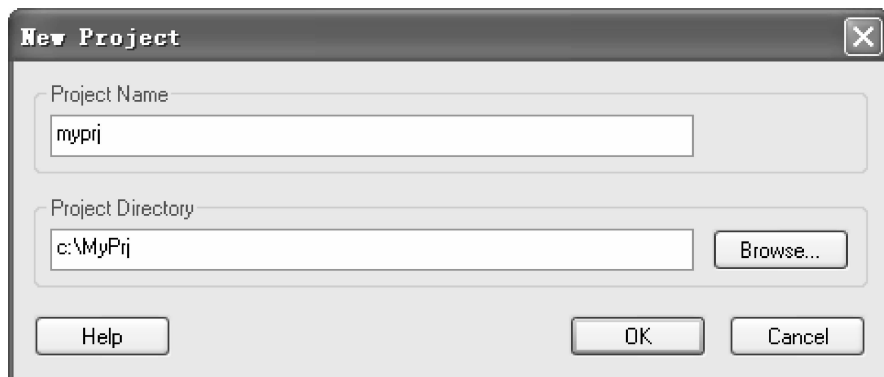


图 3-5 New Project（新项目）对话框

在 New Project（新项目）对话框中，我们将这个示例项目命名为 myprj，使用“Browse”按钮，将项目放在名为 c:\MyPrj 的文件夹中。单击“OK”按钮，在 MPLAB IDE 界面上会看到已创建项目的项目窗口，如图 3-6 所示，如果项目窗口未打开，请选择 View→Project。

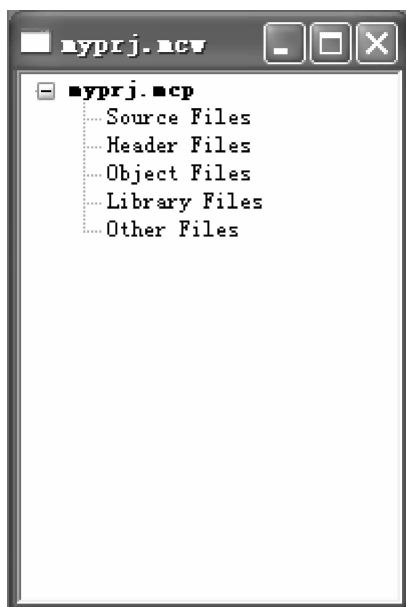


图 3-6 项目窗口

3.3.4 给项目添加文件

源文件是必须添加的。其他文件，如头文件、库文件、链接描述文件，视项目的具体情况可加可不加。在本例中，只添加一个源文件 exam1.c，在左边的项目窗口中找到 Source Files，在其上用鼠标右键单击，在弹出的快捷菜单中选 Add Files，如图 3-7 所示。

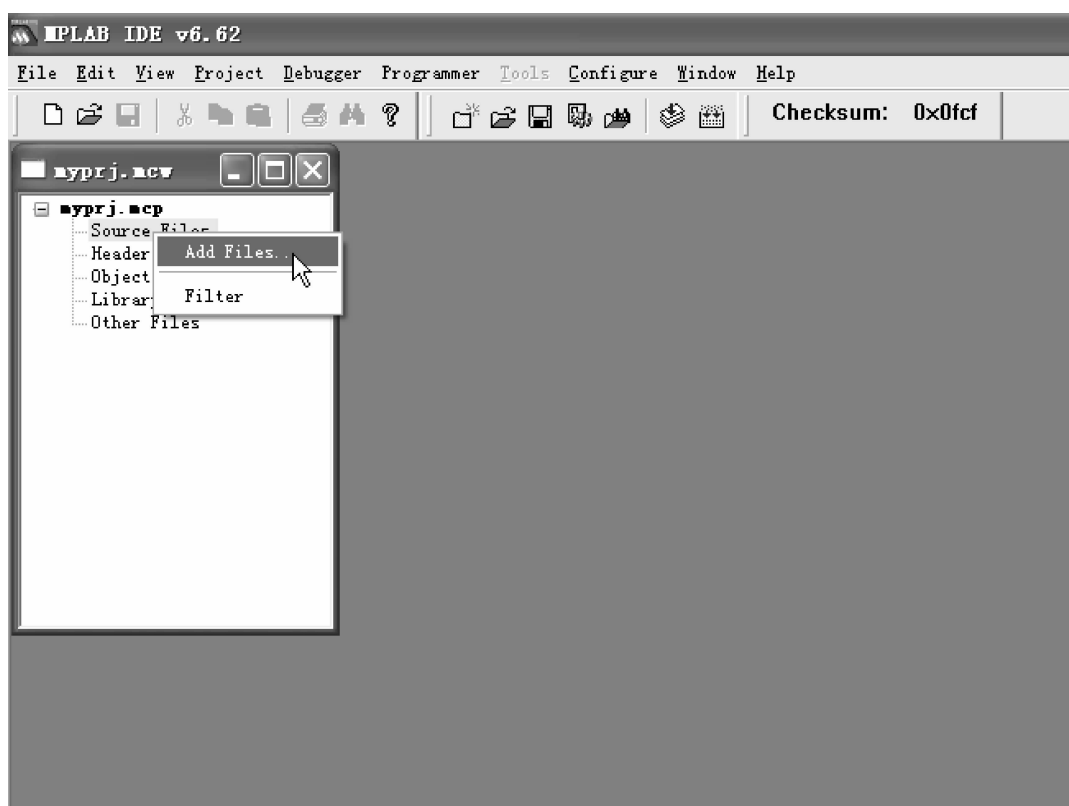


图 3-7 在项目窗口添加源文件

双击要添加的文件 exam1.c，或单击要添加的文件 exam1.c，然后单击“打开”按钮，如图 3-8 所示。

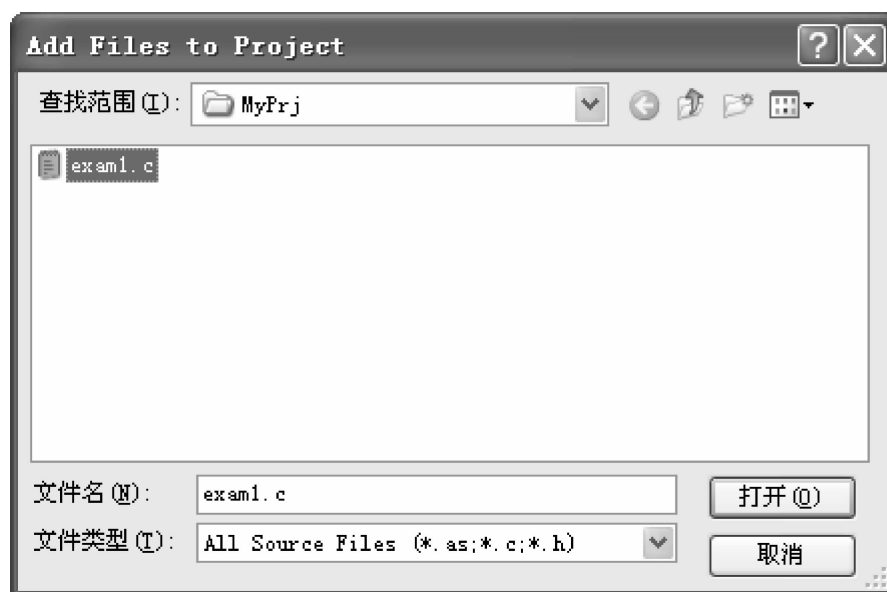


图 3-8 添加的文件对话框

文件添加成功后的项目窗口的 Source Files 下已经添加了文件 exam1.c，如图 3-9 所示。

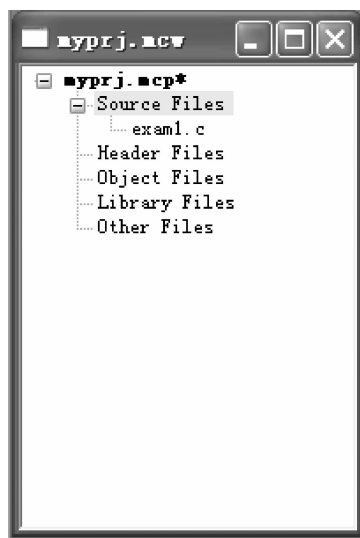


图 3-9 已添加文件后的项目窗口

3.3.5 选择器件

选择 Configure→Select Device 后会弹出 Select Device 对话框，单击 Device 栏右边的下拉箭头，在下拉菜单中既含有 PIC16F877 又含有 PIC16F877A，在本书中使用的是 PIC16F877A 芯片，一定要选择器件 PIC16F877A！（提醒：PIC16F877A 与 PIC16F877 是两个内部结构不

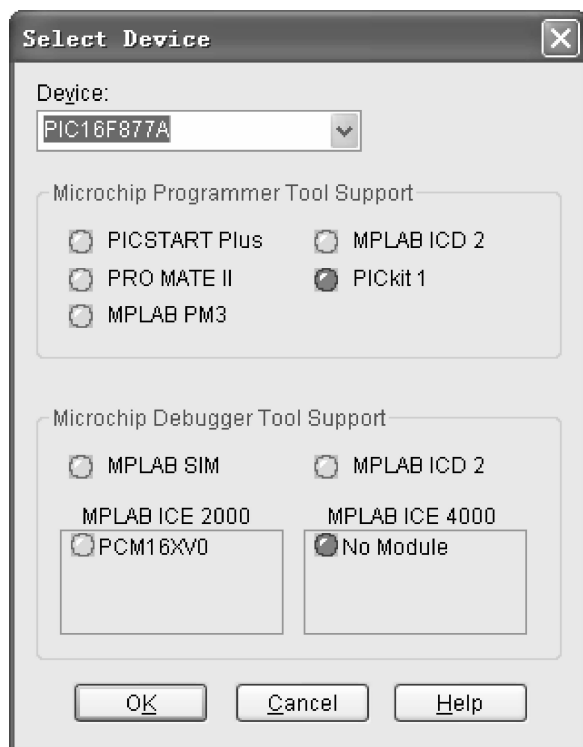


图 3-10 选择器件窗口

同的芯片，两个芯片不能直接相互替换，所以千万不要误选成 PIC16F877)，在 Microchip Programmer Tool Support 部分中，MPLAB ICD 2 旁边的“灯”图标应为绿色，如图 3-10 所示。选择完毕后，单击“OK”按钮。

说明：MPLAB ICD 2 支持的器件会在 Microchip Programmer Tool Support 下 MPLAB ICD 2 旁边有一个“绿灯”图标。不支持的器件则有一个“红灯”图标。而“黄灯”图标表明可以选择这个器件，但可能有某些限制，可能不适于量产，但它可以让 MPLAB ICD 2 做试用。

3.3.6 设置配置位

设置要下载到器件中的配置位，选择 Configure→Configuration Bits。通过单击“Settings”栏中的文本，可以更改这些配置位，如图 3-11 所示。

Oscillatot: 振荡方式选择，这取决于目标板上的晶体振荡器。

Watchdog Timer: 看门狗定时器使能位，视是否使用看门狗而定，如果使用了看门狗则选为 on，未使用则选择 Off。

Power Up Timer: 上电延时使能位，一般选择为 on，以有利于芯片起振，也可以选 Off。

Brown Out Detect: 掉电检测使能位，ICD2 作调试工具时，一定要选择为 Off。

Low Voltage Program: 低压编程使能，必须选择为 Off。

Flash Program Write: 写 Flash 使能，选择为 Write Protection Off（写保护关闭）。

Data EE Read Protect: 读内部 EEPROM 保护位，用 ICD2 作调试工具时，一定要选择为 Off。

Code Protect: 加密位，用 ICD2 作调试工具时一定要选择为 Off。

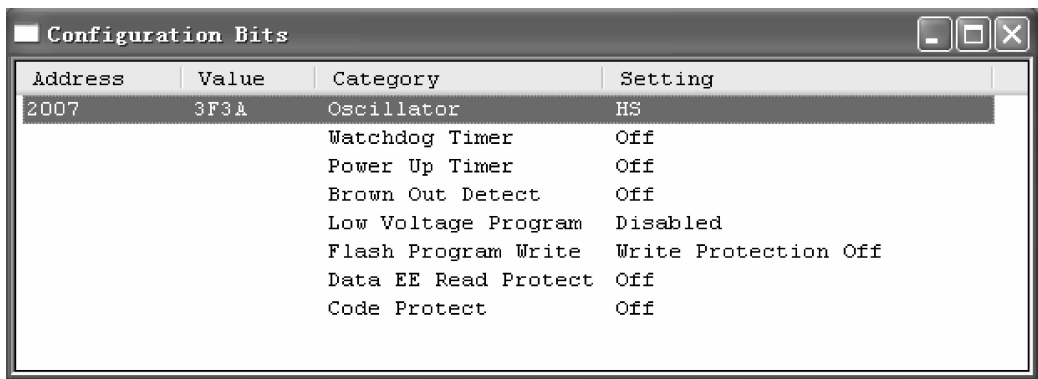


图 3-11 配置位窗口

3.3.7 选择 MPLAB ICD2 作为调试器

选择 Debugger→Select Tool →MPLAB IDE2。调试器菜单将会显示可供使用的其他调试选项。同时，输出窗口会打开显示连接信息。如果 MPLAB IDE 在启动时试图连接到 MPLAB ICD2，但因为 USB（通用串行总线）端口或一个非 COM1 的 COM 端口被使用而失败，那么允许失败，继续第二步的通信端口设置。

注意：MPLAB ICD2 作为调试工具时所下载的程序只能用于仿真调试，不能够离线使用，若要下载能离线使用的程序，就要选择 Programmerr→Select Programmer→MPLAB ICD2，

将 MPLAB ICD2 作为编程工具，如图 3-12 所示。

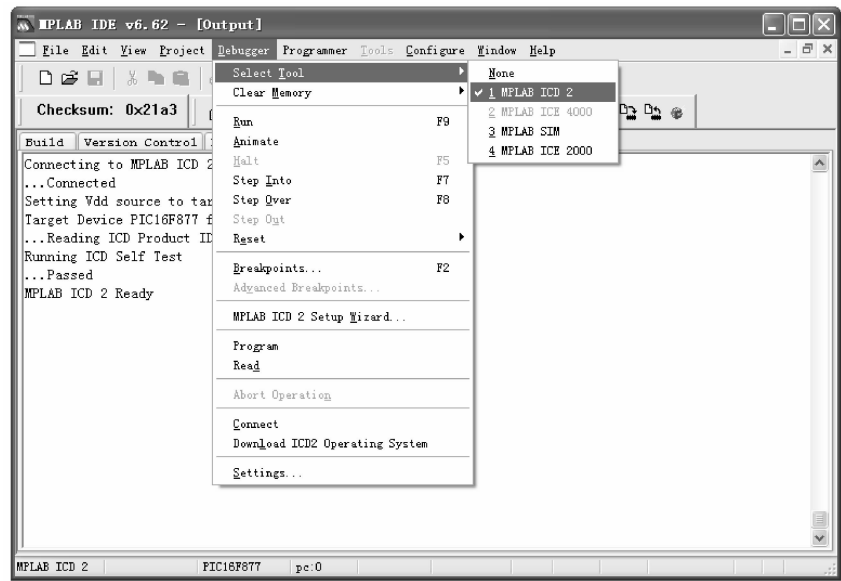


图 3-12 选择 MPLAB ICD2 作为调试器窗口

3.3.8 通过向导完成调试器的设置

选择 Debugger→MPLAB ICD2 Setup Wizard，启动 ICD2 调试器设置向导，如图 3-13 所示。

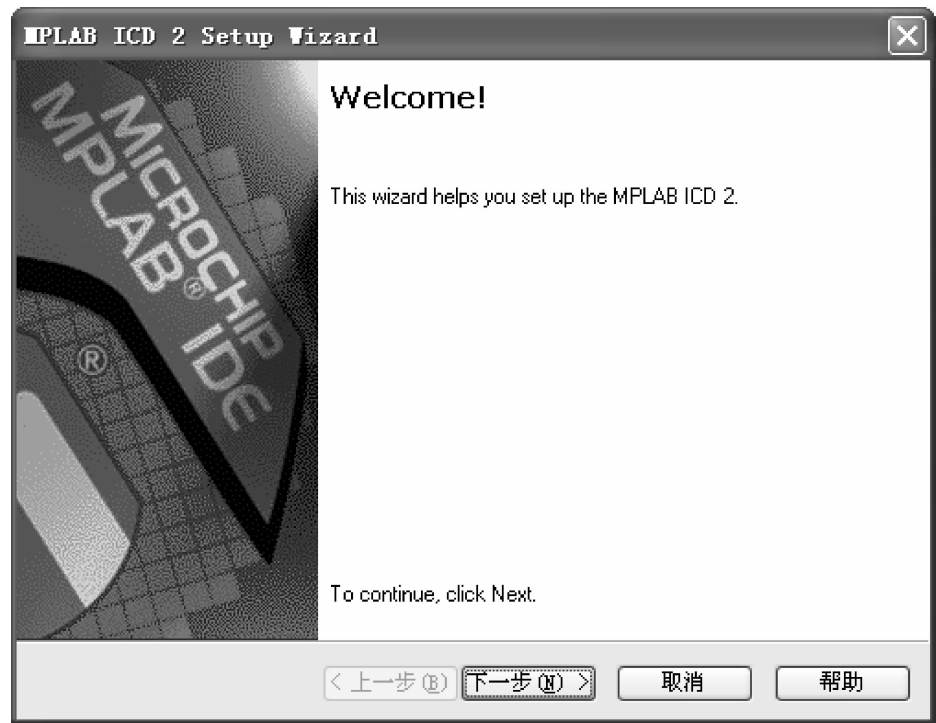


图 3-13 调试器设置向导

选择 USB 或 COM 端口/波特率。因为我们使用的“MPLAB ICD2 仿真下载器”为 USB 接口，COM 串行端口数据传输太慢，因此通信端口选 USB 即可，当然还要看 MPLAB ICD2 仿真下载器上提供什么接口，如果 USB 和 COM 端口都有，则建议使用 USB 端口，如图 3-14 所示。

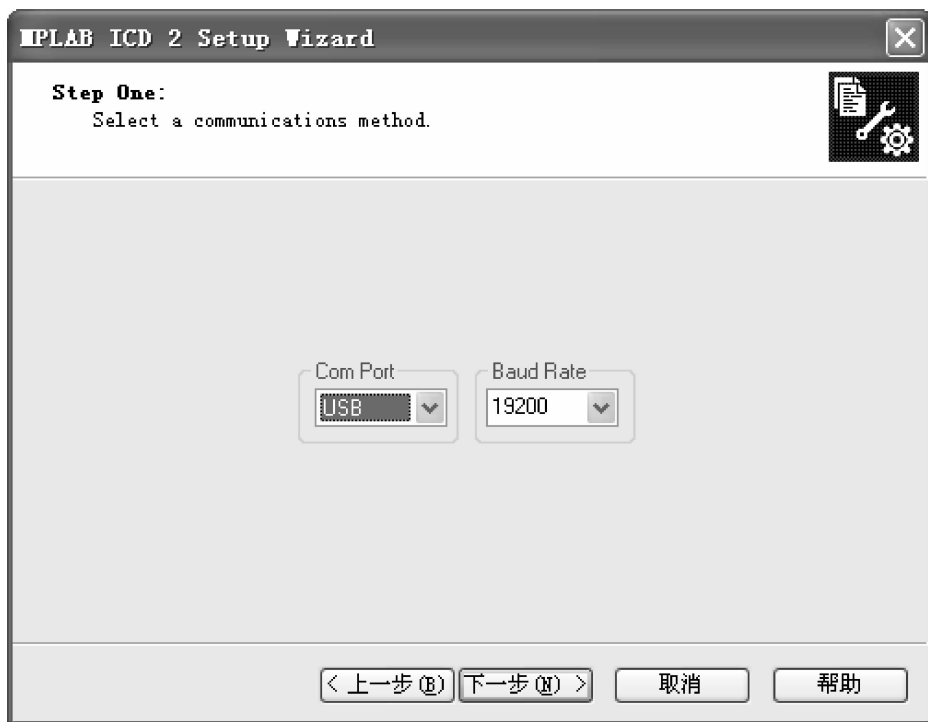


图 3-14 端口/波特率选择窗口

确定是否由“MPLAB ICD2 仿真下载器”对实验板供电。由“MPLAB ICD2 仿真下载器”向目标板供电，则选中“Power target from the MPLAB ICD2”；由独立的电源向目标板供电，则选中“Target has own power supply”，如图 3-15 所示。

自动连接前面最好打上钩，这样不用手动连接，比较方便，如图 3-16 所示。

自动下载 OS（操作系统）前面最好打上钩，这样不用手动下载 OS（操作系统），比较方便，如图 3-17 所示。

最后出现完成窗口，如图 3-18 所示。查看设置无误后，单击“完成”按钮，若发现有错误，则可单击“上一步”按钮，退回重新设置。

3.3.9 建立 PC 与 MPLAB ICD2 仿真下载器之间的通信联系

手动连接：选择 Debugger→Connect，连接到 MPLAB ICD 2。

自动连接：如果 ICD 2 设置中选择了“Automatically connect at startup”，那么系统会自动连接，输出窗口提示“…Connected”，并且没有出现红色的警告或错误信息，表示已经正常连接，如图 3-19 所示。

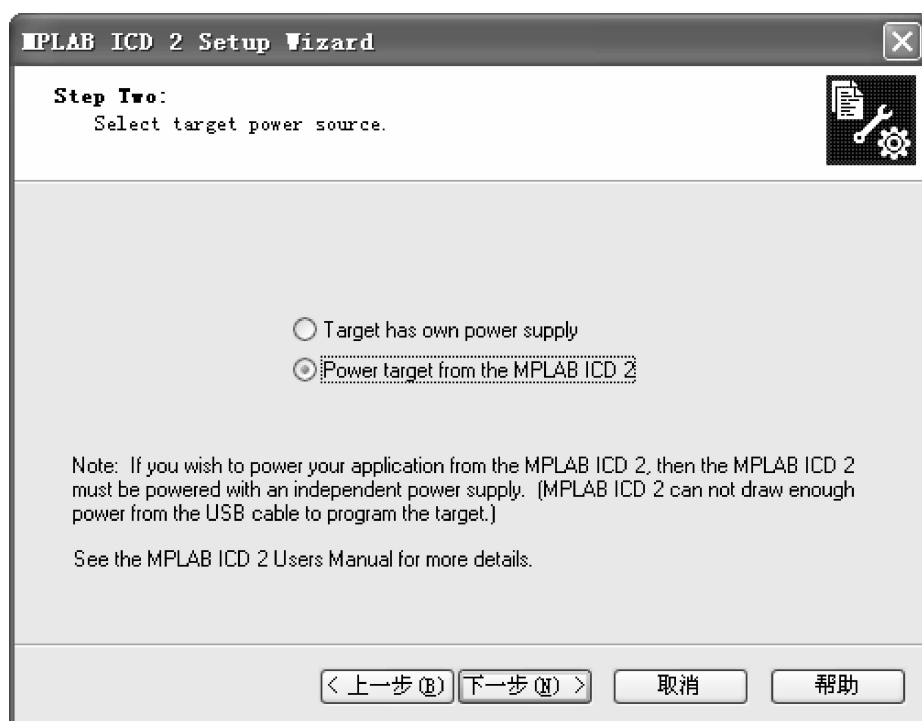


图 3-15 供电选择窗口

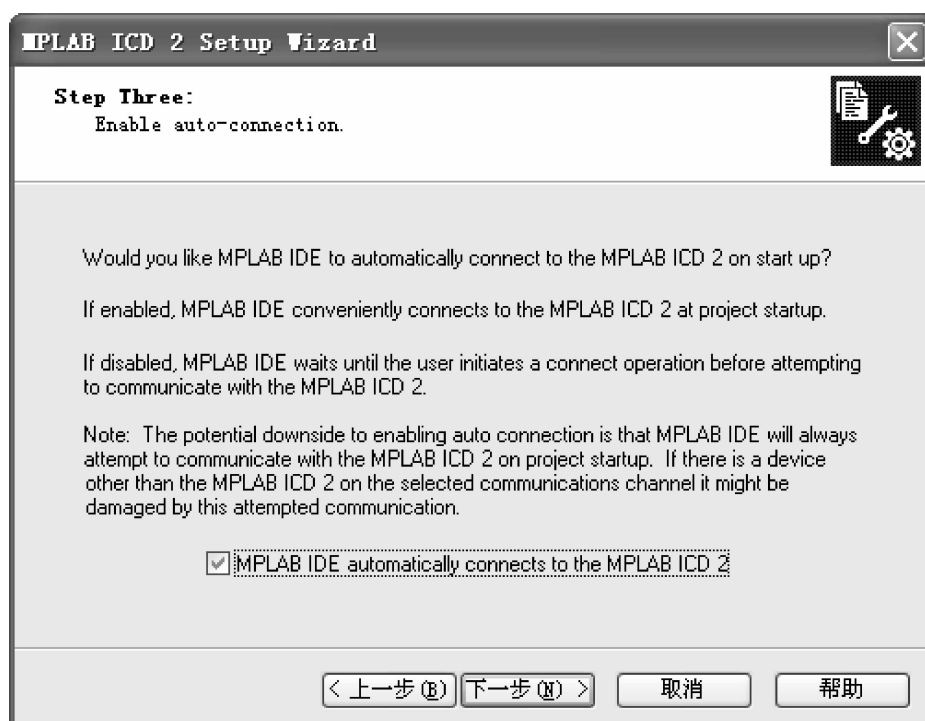


图 3-16 自动连接询问窗口

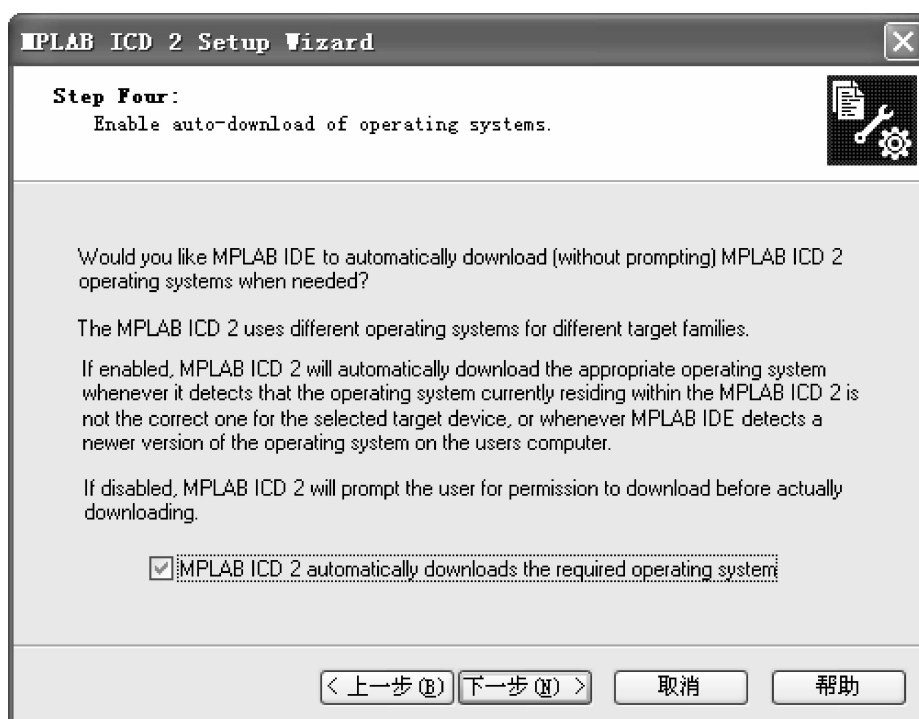


图 3-17 自动下载 OS 询问窗口

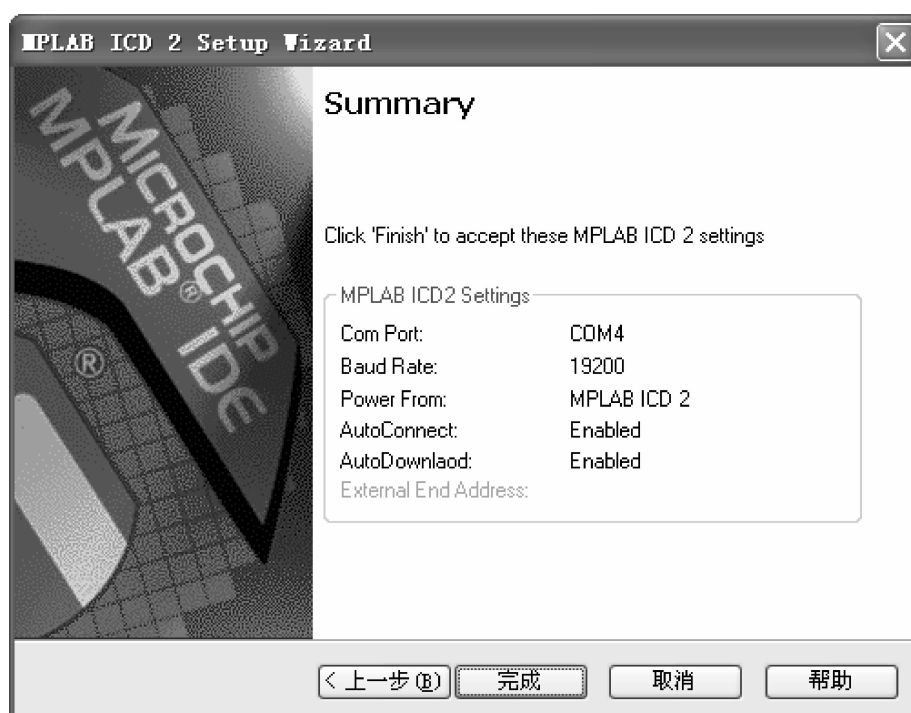


图 3-18 完成窗口

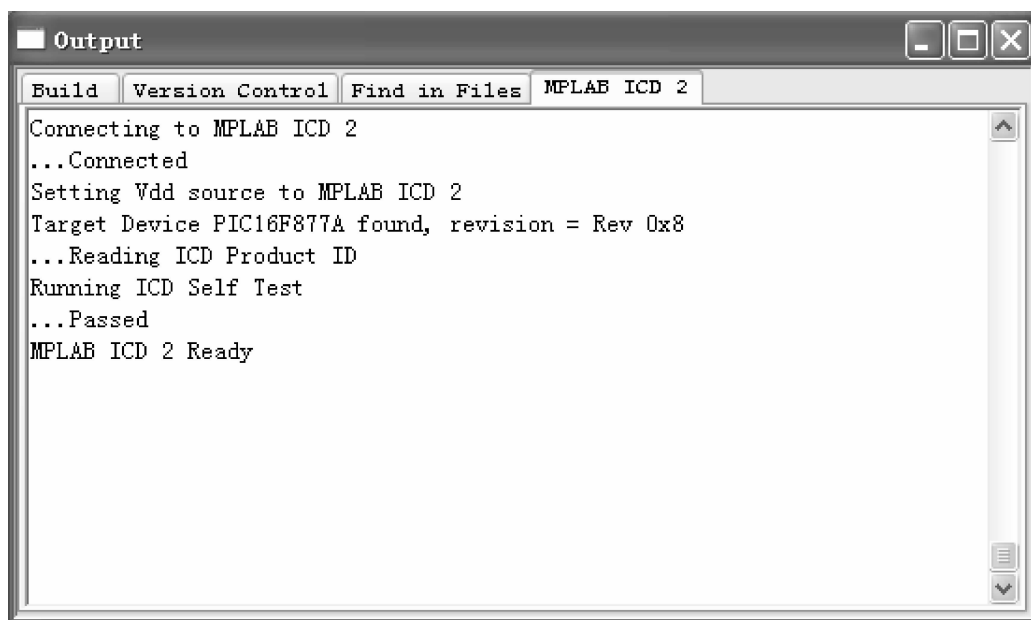


图 3-19 显示连接正常的输出窗口

3.3.10 更新 MPLAB ICD2 固件（操作系统）

根据 MPLAB IDE 的版本或器件的选择，会提示一条信息说明固件需要更新。MPLAB IDE 将自动安装新的固件。

同时，由于不同的 MPLAB ICD2 固件用于不同的 PIC 系列单片机，当切换不同的 PIC 系列单片机时也可能出现这个对话框。

固件可以按照下面的步骤手动更改：

选择 Debugger→Download ICD2 Operating System。选择 ICD2 固件文件对话框会打开。

从裂变中或浏览器中选择要下载的固件文件。文件名的组成为 icdxxxxxx.hex，其中 xxxxxx 为版本号。关于 MPLAB IDE 提供最新版本的信息可以查看 README 文件或 MPLAB ICD2.txt。根据选择的 PIC 器件版本可能会变动。

单击“Open”按钮，MPLAB IDE 将下载新的操作系统到 MPLAB ICD2。

3.3.11 生成目标文件（编译）

在编译之前，先确认编译器选择正确，选择 Project→Select Language Toolsuite，出现语言选择窗口，如图 3-20 所示。如果程序用汇编语言编写，在 Active Toolsuite 中应选择 Microchip MPASM Toolsuite；如果程序用 C 语言编写，在 Active Toolsuite 中应选择 HI-TECH PICC Toolsuite。当然还有很多编译器可供选择，此处不一一列举。

选择 Project→Build All 或在项目窗口中的项目名称上单击右键，并从弹出菜单中选择“Build All”。

编译器总会生成文件名与源文件（.c 文件）名相同的目标文件（.hex 文件）。编译时会打开一个状态窗口，显示编译的进度和最后的结果。编译完成后，自动关闭该窗口。同时

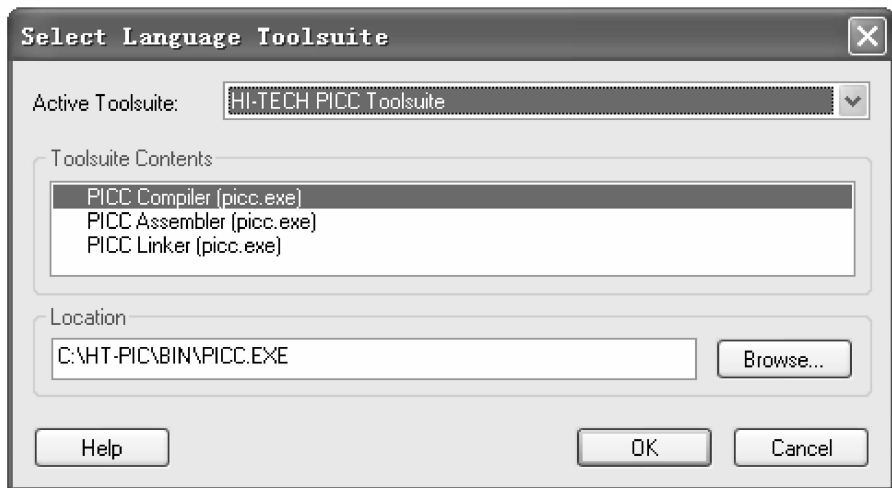


图 3-20 语言选择窗口

还将打开 Output 窗口，Output 窗口没有出现红色警告或错误信息，而且最后一句显示“BUILD SUCCEEDED”，就表示编译成功了，如图 3-21 所示。

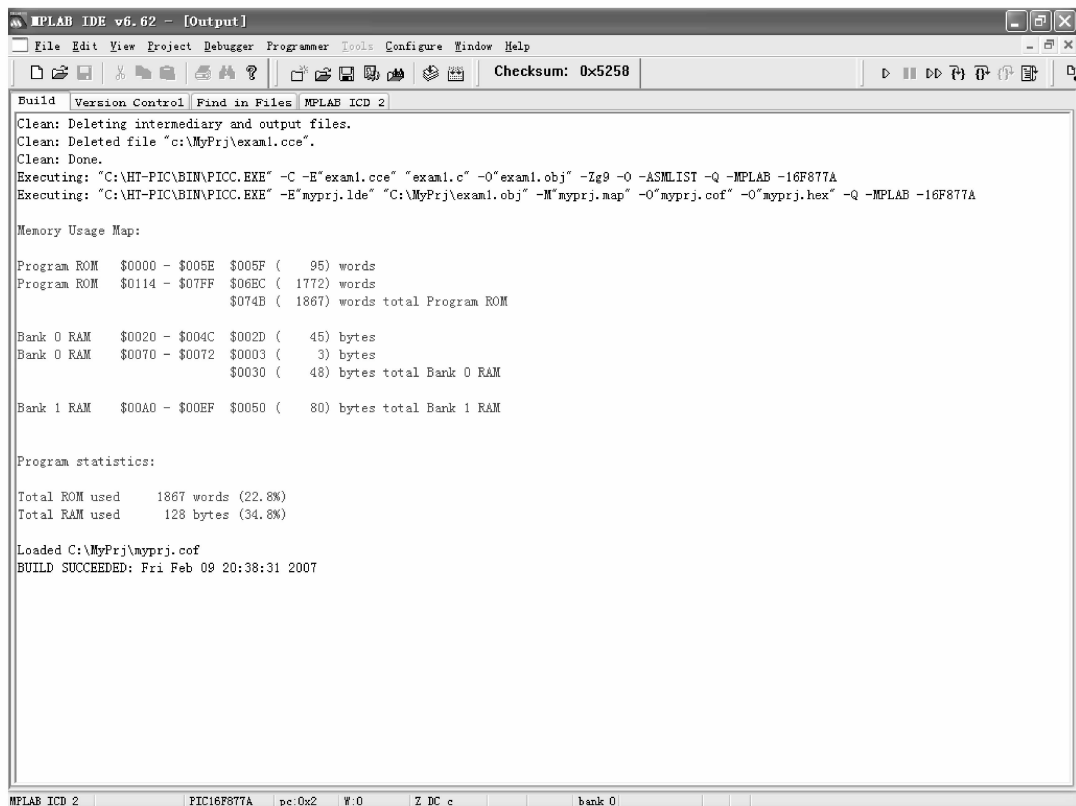


图 3-21 编译窗口

注意：为调试生成目标文件（. HEX 文件）之前，必须已经选择 ICD 2 作为调试器。如果编译失败，请检查以下各项，然后重新编译项目：

- 1) 检查在编辑器窗口中输入的所有代码的拼写和格式。如果在 Output 窗口中报告有错误，双击该错误就会在源代码窗口左侧的灰色区域中以绿色箭头指出源代码中出错的行。
- 2) 检查用于该项目及其项目文件的语言工具是否正确。

为初学者解释几个概念：

C 语言是一种高级编程语言，用 C 语言进行编程写出的代码，保存为 .c 文件，称为 C 语言的源程序，但单片机无法识别和运行源程序，单片机只能看懂由 0 和 1 组成的机器码，因此这里需要一个“翻译”的过程，称为“编译”，编译的作用就是把单片机无法识别的汇编源程序翻译成单片机能识别的机器码，如图 3-22 所示，由于机器码是最终在单片机上运行的代码，因此也称为目标代码，其后缀是 .hex。

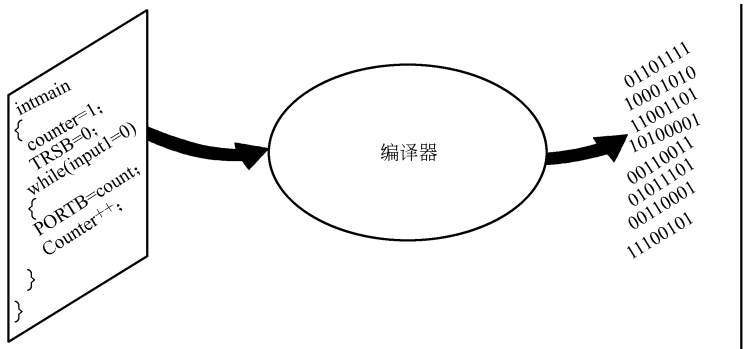


图 3-22 编译器将源代码转换为机器指令

另外，整个“翻译”的过程是由一个 PICC 工具软件完成的。当选择 Project→Build All 时，系统会自动调用 PICC 编译器，自动完成整个编译过程。关于 PICC 的安装与使用请参照第 8 章实验一。

3.3.12 下载目标代码

在调试之前，必须把目标代码下载到目标单片机中。选择 Debugger→Program，把目标文件 exam1.c（目标文件中包括的就是目标代码）下载到实验板上的 PIC 中。下载可能需要几分钟时间。在下载过程中，Output 对话框的 MPLAB ICD 2 选项卡下会显示目前的操作阶段。Output 窗口没有出现红色的警告或错误信息，而且倒数第 3 句显示“…Programming succeeded”，就表示下载成功了，如图 3-23 所示。

注意：进入调试模式的一个必要条件是，开发板或应用系统已经加了晶体振荡器。如编程器模式下下载没问题，但调试器模式下下载不成功，则通常是开发板或应用系统忘了装上晶体振荡器。注意：初学者容易将 Debugger（调试器）下拉菜单下的“Program（编程）”命令和 Programmer（编程器）下拉菜单下的“Program（编程）”命令混为一谈，其实这是两个作用完全不同的命令，切记：在调试器模式下，必须使用调试器下拉菜单下的“Program”命令，同样，在编程器模式下，必须使用编程器下拉菜单下的“Program”命令。同时也不要以为：在编程器模式下已经执行“Program”命令了，因此切换到调试器模式时，

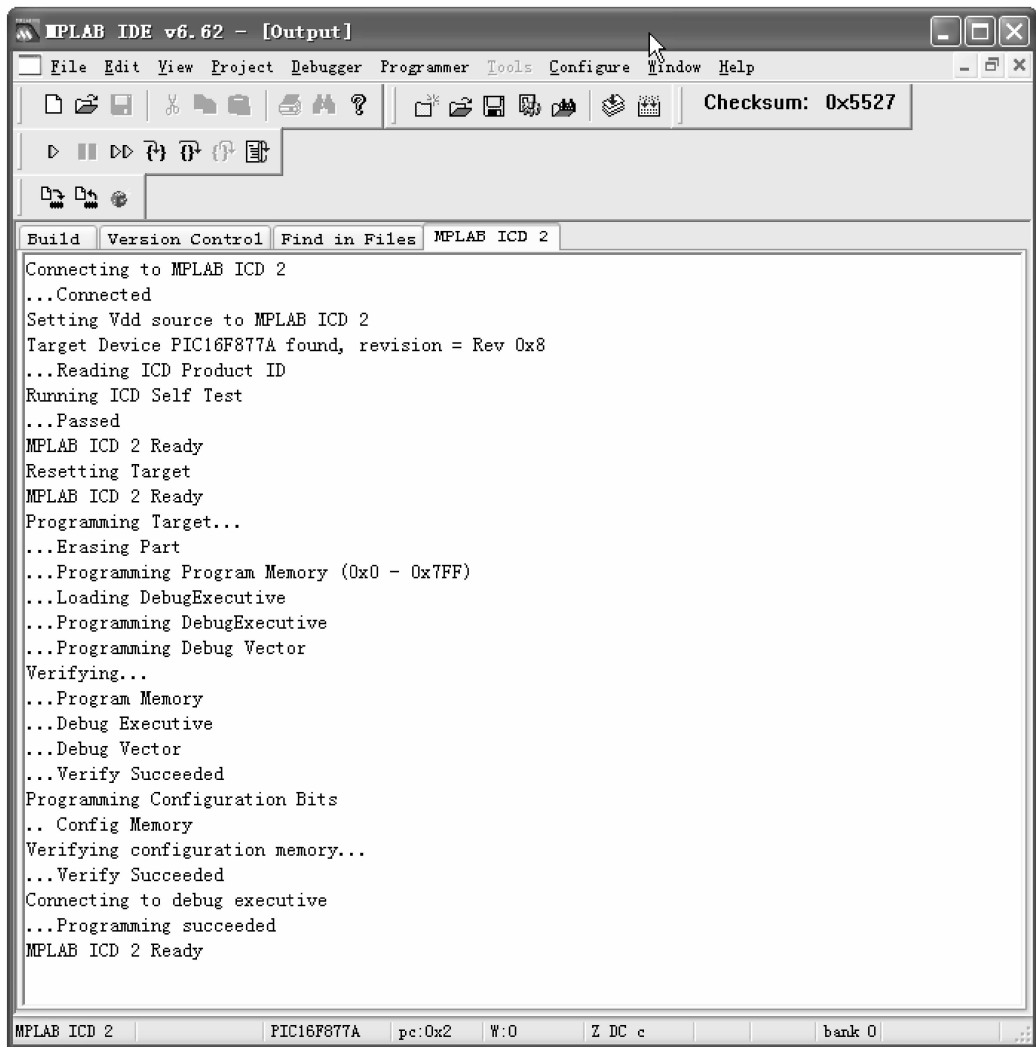


图 3-23 下载目标代码窗口

就可以不用执行调试器模式下的“Program”命令了，这是大错特错的，反之也是一样。

Debugger（调试器）下拉菜单下和 Programmer（编程器）下拉菜单下还有许多命令在字面上是一样的，但实际功能是不一样的，注意不要混淆。下面将作进一步的解释，初学者可以跳过不读。

执行 Debugger（调试器）下拉菜单下的 Program（编程）命令时，系统会自动在用户编写程序的目标代码中加入调试执行代码，调试执行代码会自动下载到程序存储器的高端地址，以便使用“MPLAB ICD2 仿真下载器”来进行调试，这种状态下，MPLAB IDE 可以通过 MPLAB ICD2 仿真下载器来监视 PIC 系列单片机的各个存储器所存储的内容；执行 Programmer（编程器）下拉菜单下的 Program（编程）命令时，系统只将用户编写程序的目标代码下载到程序存储器中，MPLAB IDE 无法监视 PIC 系列单片机存储器中的存储内容。由此可见，这两个编程命令的作用是完全不同的。

3.3.13 运行和调试

在调试器模式下运行 exam1 进入调试模式后，工具栏会多出调试器快捷工具条，如图 3-24 所示。

从左到右分别是：

- 1) run——全速运行；
- 2) halt——暂停；
- 3) animate——自动单步（或称为慢速运行）；
- 4) step into——单步运行（子程序内部也单步运行）；
- 5) step over——单步运行（子程序内部代码被全速执行，整个子程序被作为单步运行中的一步来执行）；
- 6) step out ——跳出子循环（当程序运行在子程序中时，可以使用它直接运行完该子程序）；
- 7) reset——复位。



图 3-24 快捷工具条

MPLAB ICD 2 在调试器模式下可以实时或单步执行代码，实时执行代码又称为“全速运行”，当然调试器模式下的“全速运行”与单片机脱开发系统独立运行时的“全速运行”还是有区别的，单步执行代码又称为“单步运行”。

首先执行调试器模式下的“全速运行”。选择 Debugger→Run（或单击 Run 工具栏按钮）。

如果编写的程序没有错误，则可继续下一步——在编程器模式下下载目标代码；如果有错误，则先选择 Debugger→Halt（或单击 Halt 工具栏按钮）来中止程序执行，然后利用一些调试方法：如设置断点对程序进行修改，直到无误为止。需要注意的是，当对程序作了修改以后，记得保存之后选择 Project→Build All 来重新编译并重新下载源代码。

3.3.14 在编程器模式下下载目标代码

当程序成功调试并运行后，下一步是在编程器模式下下载 PIC 系列单片机，以便它能脱开发系统而独立工作。进行这一步时，为 MPLAB ICD2 保留的资源被释放。

按照以下步骤进行下载：

1) 选择 Debugger → Select Tool → None，禁止 MPLAB ICD 2 作为调试器，如图 3-25 所示。

2) 在 Programmer→Select Programmer 菜单中，选择 MPLAB ICD 2 作为编程器，如图 3-26 所示。

3) 选择 Programmer → MPLAB ICD2 Setup Wizard…，通过向导完成编程器的设置，与 3.3.9 节中的设置方法相似。

4) 选择 Programmer→Verify，检测硬件连接。如果检测无误，则在输出窗口出现提示，如图 3-27 所示。

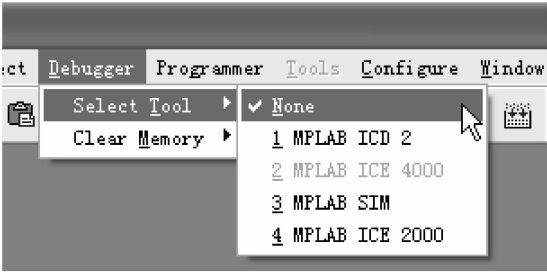


图 3-25 禁止 MPLAB ICD 2 作为调试器

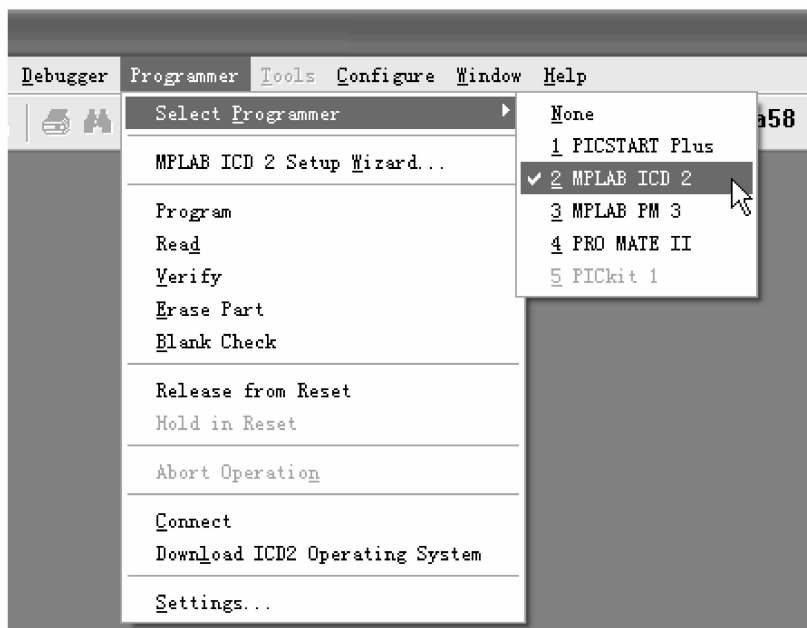


图 3-26 选择 MPLAB ICD2 作为编程器

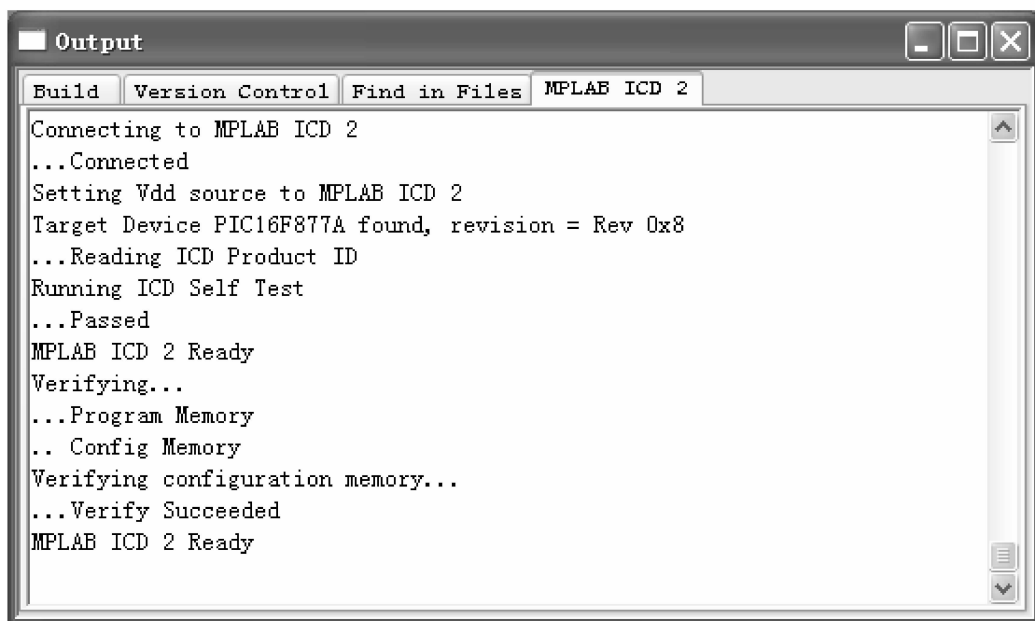


图 3-27 显示硬件连接无误的输出窗口

5) 选择 Programmer→Program, 下载程序到目标板。如果下载成功, 则在输出窗口出现提示, 如图 3-28 所示。

此后, 目标板可以独立运行了。

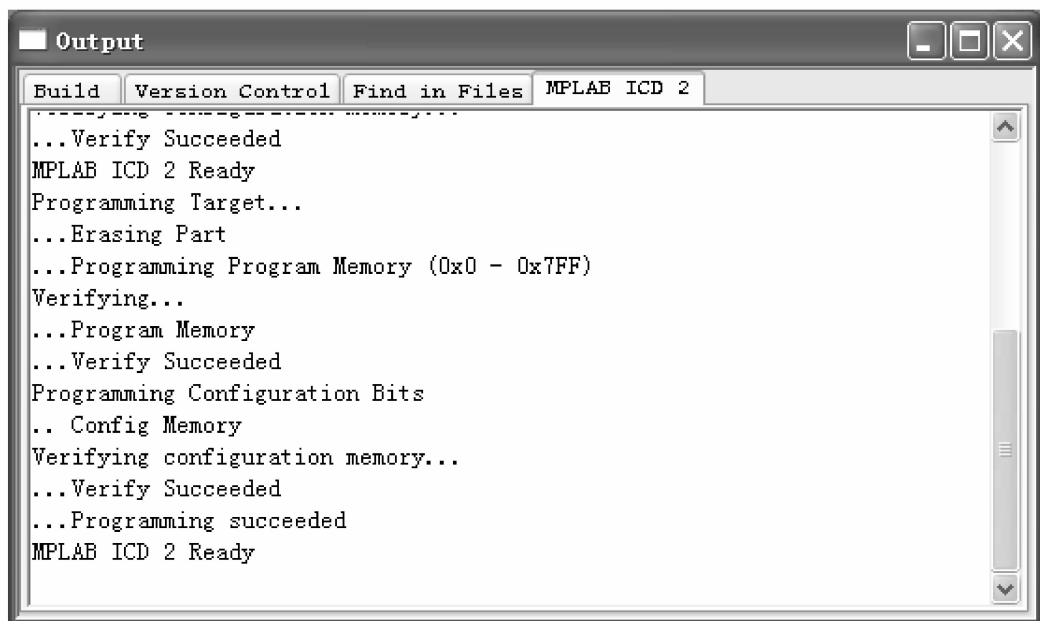


图 3-28 显示下载成功的输出窗口

3.3.15 文件保存

开发工作中途及完成后，要及时进行文件的保存，可以选择 File→Save Workspace，保存为工作空间文件或选择 Project→Save Project，保存项目文件。

下面对这两种文件类型及其区别进行说明：

Project（项目文件）包括了编译所需的源文件、连接文件、头文件……及这些文件间的组织结构，同时还包括了芯片设置、配置位设置等。

Workspace（工作空间文件）在项目文件包含的内容基础上，还进一步包含了调试器或编程器的选择、已打开的窗口等。

简单地说：项目文件就是程序员辛辛苦苦做出来的“产品”，工作空间文件比项目文件多包括了一些内容，多包括的内容其实就是做“产品”过程中用到的各种工具，例如当前正在调试程序，为此选择了“MPLAB ICD2 仿真下载器”，还打开了变量观察窗口，这些在工作空间文件中也会被保存。如果项目还没有最终完成，那么在保存时应该保存为工作空间文件类型，下一次打开时，应该打开这个项目文件，这样出现在您面前的一切和您上次看到的一模一样，您可以方便地继续您未完成的工作。

3.4 软件编程基础

本书主要采用 C 语言进行编程，故本节介绍一些 C 语言的常用语句，其他语句和用法，请读者根据需要参阅其他相关书籍。

3.4.1 C 语言概述

C 语言是在 20 世纪 70 年代初问世的。1978 年由美国电话电报公司 (AT&T) 贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的《THE C PROGRAMMING LANGUAGE》一书。通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的标准 C 语言，后来由美国国家标准学会在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称之为 ANSI C。

C 语言是一种结构化语言。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构。它还可以直接访问内存的物理地址，进行位 (bit) 一级的操作。由于 C 语言实现了对硬件的编程操作，因此 C 语言集高级语言和低级语言的功能于一体。它既可用于系统软件的开发，也适合于应用软件的开发。此外，C 语言还具有效率高、可移植性强等特点。因此广泛地移植到了各类各型计算机上，从而形成了多种版本的 C 语言。这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

3.4.2 整型量

整型量包括整型常量、整型变量。整型常量就是整常数。在 C 语言中，使用的整常数有八进制、十六进制和十进制三种。

1. 整型常量

(1) 八进制整常数 八进制整常数必须以数 0 开头，即以数 0 作为八进制数的前缀。数码取值为 0~7。八进制数通常是无符号数的。

以下各数是合法的八进制数：

015 (十进制为 13) 0101 (十进制为 65) 0177777 (十进制为 65535)

以下各数不是合法的八进制数：

256 (无前缀 0) 03A2 (包含了非八进制数码) -0127 (出现了负号)

(2) 十六进制整常数 十六进制整常数的前缀为 0X 或 0x。其数码取值为 0~9, A~F 或 a~f。

以下各数是合法的十六进制整常数：

0X2A (十进制为 42) 0XA0 (十进制为 160) 0XFFFF (十进制为 65535)

以下各数不是合法的十六进制整常数：

5A (无前缀 0X) 0X3H (含有非十六进制数码)

(3) 十进制整常数 十进制整常数没有前缀。其数码为 0~9。

以下各数是合法的十进制整常数：

237 -568 65535 1627

以下各数不是合法的十进制整常数：

023 (不能有前导 0) 23D (含有非十进制数码)

2. 整型变量

整型变量可分为以下几类：①基本型，类型说明符为 int，在内存中占 2 字节，其取值

为基本整常数。②短整型，类型说明符为 `short int` 或 `short`。所占字节和取值范围均与基本型相同。③长整型，类型说明符为 `long int` 或 `long`，在内存中占 4 字节，其取值为长整常数。④无符号型，类型说明符为 `unsigned`。

无符号型又可与上述三种类型匹配而构成：①无符号基本型，类型说明符为 `unsigned int` 或 `unsigned`。②无符号短整型，类型说明符为 `unsigned short`。③无符号长整型，类型说明符为 `unsigned long`。各种无符号类型量所占的内存空间字节数与相应的有符号类型量相同。但由于省去了符号位，故不能表示负数。

整型变量的说明：变量说明的一般形式为：类型说明符 变量名标识符，变量名标识符，…；

例如：

```
int a, b, c; (a, b, c 为整型变量)
```

```
long x, y; (x, y 为长整型变量)
```

```
unsigned p, q; (p, q 为无符号整型变量)
```

3.4.3 符号常量

在 C 语言中，可以用一个标识符来表示一个常量，称之为符号常量。符号常量在使用之前必须先定义，其一般形式为：

```
#define 标识符 常量
```

其中，`#define` 也是一条预处理命令，称为宏定义命令，其功能是把该标识符定义为其后的常量值。一经定义，以后在程序中所有出现该标识符的地方均代之以该常量值。习惯上符号常量的标识符用大写字母，变量标识符用小写字母，以示区别。

```
#define PI 3.14159
```

```
void main ()
```

```
{ float s, r;
```

```
  r = 5.0;
```

```
  s = PI * r * r;
```

```
  printf ("s = %f\n", s);
```

```
}
```

程序运行结果如下：

```
s = 78.539749
```

程序中用 `#define` 命令行定义 `PI` 代表一串字符 3.14159，在对程序进行编译时，凡本程序中出现 `PI` 的地方，编译程序均用 3.14159 这一字符串来代替，在这里，`PI` 是一个用户自己选择的符号名，本程序中，可以把 `PI` 视为 3.14159 的替身。

3.4.4 简单赋值运算与赋值表达式

在 C 语言中，“=” 符号称为赋值运算符，由赋值运算符组成的表达式称为赋值表达式，其一般形式为：

```
变量名 = 表达式
```

例如：


```
a = 10          /* 把常量 10 赋给变量 a */
b = a          /* 把 a 中的值赋给变量 b, a 中的值不变 */
```

赋值表达式的功能是把计算表达式的值再赋予左边的变量。赋值运算符具有右结合性。因此 $a = b = c = 5$ 可理解为 $a = (b = (c = 5))$ 。

在其他高级语言中，赋值构成了一个语句，称为赋值语句。而在 C 语言中，把 “=” 定义为运算符，从而组成赋值表达式。凡是表达式可以出现的地方均可出现赋值表达式。例如，式子 $x = (a = 5) + (b = 8)$ 是合法的。它的意义是把 5 赋予 a，8 赋予 b，再把 a、b 相加，和赋予 x，故 x 应等于 13。

在 C 语言中也可以组成赋值语句，按照 C 语言规定，任何表达式在其末尾加上分号就构成为语句。因此如：

```
x = 8;
a = b = c = 5;
```

都是赋值语句，如果赋值运算符两边的数据类型不相同，系统将自动进行类型转换，即把赋值号右边的类型换成左边的类型。

3.4.5 控制语句

1. if 语句

用 if 语句可以构成分支结构。它根据给定的条件进行判断，以决定执行某个分支程序段。C 语言的 if 语句有三种基本形式。

1) 第一种形式为基本形式 if (表达式) 语句；其语义是：如果表达式的值为真，则执行其后的语句，否则不执行该语句。

例如：

```
void main()
{ int a,b,max;
  printf("\n input two numbers: ");
  scanf("%d%d",&a,&b);
  max = a;
  if (max < b) max = b;
  printf("max = %d",max);
}
```

本例程序中，输入两个数 a，b。把 a 先赋予变量 max，再用 if 语句判别 max 和 b 的大小，如 max 小于 b，则把 b 赋予 max。因此 max 中总是大数，最后输出 max 的值。

2) 第二种形式为 if-else 形式。

if (表达式)

语句 1;

else

语句 2;

其语义是：如果表达式的值为真，则执行语句 1，否则执行语句 2。

例如：同样完成功能：输入两个整数，输出其中较大的数，改用 if-else 语句判别 a、b

的大小，若 a 大，则输出 a，否则输出 b。

```
void main ( )
{ int a, b;
  printf ( " input two numbers: " );
  scanf ( " %d%d", &a, &b);
  if ( a > b)
    printf ( " max = %d \ n", a);
  else
    printf ( " max = %d \ n", b);
}
```

3) 第三种形式为 if-else-if 形式。

前两种形式的 if 语句一般都用于两个分支的情况，当有多个分支选择时，可采用 if-else-if 语句，其一般形式为：

```
if ( 表达式 1)
  语句 1;
else if ( 表达式 2)
  语句 2;
else if ( 表达式 3)
  语句 3;
...
else if ( 表达式 m)
  语句 m;
else
  语句 n;
```

其语义是：依次判断表达式的值，当出现某个值为真时，则执行其对应的语句，然后跳到整个 if 语句之外继续执行程序。如果所有的表达式均为假，则执行语句 n，然后继续执行后续程序。

例如：判别键盘输入字符的类别。可以根据输入字符的 ASCII 来判别类型。由 ASCII 表可知，ASCII 值小于 32 的为控制字符。在“0”和“9”之间的为数字，在“A”和“Z”之间的为大写字母，在“a”和“z”之间的为小写字母，其余则为其他字符。这是一个多分支选择的问题，用 if-else-if 语句编程，判断输入字符 ASCII 所在的范围，分别给出不同的输出。例如输入为“g”，输出显示它为小写字母。

```
#include "stdio. h"
void main( )
{ char c;
  printf( "input a character: " );
  c = getchar( );
  if(c < 32)
    printf( " This is a control character \n" );
```

```

else if( c >='0' && c <='9' )
    printf( " This is a digit\n" );
else if( c >='A' && c <='Z' )
    printf( " This is a capital letter\n" );
else if( c >='a' && c <='z' )
    printf( " This is a small letter\n" );
else
    printf( " This is an other character\n" );
}

```

使用 if 语句时需注意以下几个问题：

1) 在三种形式的 if 语句中，在 if 关键字之后均为表达式。该表达式通常是逻辑表达式或关系表达式，但也可以是其他表达式，如赋值表达式等，甚至也可以是一个变量。例如：if (a=5) 语句；if (b) 语句；都是允许的。只要表达式的值为非 0，即为“真”。如在 if (a=5) …；中，表达式的值永远为非 0，所以其后的语句总是要执行的，当然这种情况在程序中不一定会出现，但在语法上是合法的。

2) 在 if 语句中，条件判断表达式必须用括号括起来，在语句之后必须加分号。

3) 在 if 语句的三种形式中，所有的语句应为单个语句，如果要想在满足条件时执行一组（多个）语句，则必须把这一组语句用 {} 括起来组成一个复合语句。但要注意的是在 {} 之后不能再加分号。例如：

```

if ( a > b )
{
    a ++ ;
    b ++ ;
}
else
{
    a = 0 ;
    b = 10 ;
}

```

4) if 和 else 子句中可以是任意合法的 C 语句，因此当然也可以是 if 语句，通常称此为嵌套的 if 语句。内嵌的 if 语句既可以嵌套在 if 子句中，也可以嵌套在 else 子句中。例如：根据输入的学生成绩，给出相应的等级。90 分以上的等级为 A，60 分以下的等级为 E，其余每 10 分为一个等级。

```

main( )
{
    int g;
    printf( " Enter g: " );
    scanf( " %d" , &g );
    printf( " g = %d:" , g );
    if( g >= 90 )
        printf( " A" \n );
    else if( g >= 80 )

```

```

    printf( " B" \n );
else if( g >= 70 )
    printf( " C" \n );
else if( g >= 60 )
    printf( " D" \n );
else
    printf( " E" \n );

```

2. switch 语句

C 语言还提供了另一种用于多分支选择的 switch 语句，其一般形式为：

```

switch ( 表达式 ) {
case 常量表达式 1: 语句 1;
case 常量表达式 2: 语句 2;
...
case 常量表达式 n: 语句 n;
default: 语句 n + 1;
}

```

其语义是：计算表达式的值。并逐个与其后的常量表达式值相比较，当表达式的值与某个常量表达式的值相等时，即执行其后的语句，然后不再进行判断，继续执行后面所有 case 后的语句。如表达式的值与所有 case 后的常量表达式均不相同时，则执行 default 后的语句。例如：

```

void main( )
{
    int a;
    printf( " input integer number: " );
    scanf( " %d" ,&a);
    switch ( a )
    {
        case 1:printf( " Monday\n" );
        case 2:printf( " Tuesday\n" );
        case 3:printf( " Wednesday\n" );
        case 4:printf( " Thursday\n" );
        case 5:printf( " Friday\n" );
        case 6:printf( " Saturday\n" );
        case 7:printf( " Sunday\n" );
        default:printf( " error\n" );
    }
}

```

本程序是要求输入一个数字，输出一个英文单词。但是当输入 3 之后，却执行了 case3 以及以后的所有语句，输出了 Wednesday 及以后的所有单词。这当然是不希望的。为什么会出现这种情况呢？这恰恰反映了 switch 语句的一个特点。在 switch 语句中，“case 常量表达式”只相当于一个语句标号，表达式的值和某标号相等则转向该标号执行，但不能在执行完

该标号的语句后自动跳出整个 switch 语句，所以出现了继续执行所有后面 case 语句的情况。这是与前面介绍的 if 语句完全不同的，应特别注意。为了避免上述情况，C 语言还提供了一种 break 语句，专用于跳出 switch 语句，break 语句只有关键字 break，没有参数。在后面还将详细介绍。修改例题的程序，在每一个 case 语句之后增加 break 语句，使每一次执行之后均可跳出 switch 语句，从而避免输出不应有的结果。

```
void main()
{
    int a;
    printf("input integer number: ");
    scanf("%d",&a);
    switch (a)
    {
        case 1:printf("Monday\n");break;
        case 2:printf("Tuesday\n"); break;
        case 3:printf("Wednesday\n");break;
        case 4:printf("Thursday\n");break;
        case 5:printf("Friday\n");break;
        case 6:printf("Saturday\n");break;
        case 7:printf("Sunday\n");break;
        default:printf("error\n");
    }
}
```

在使用 switch 语句时还应注意以下几点：

- 1) 在 case 后的各常量表达式的值不能相同，否则会出现错误。
- 2) 在 case 后，允许有多个语句，可以不用 {} 括起来。
- 3) 各 case 和 default 子句的先后顺序可以变动，而不会影响程序执行结果。
- 4) default 子句可以省略不用。

3. while 语句

while 语句的一般形式为：

while (表达式) 语句；

其中，表达式是循环条件，语句为循环体。while 语句的语义是：计算表达式的值，当值为真（非 0）时，执行循环体语句。

例如：统计从键盘输入一行字符的个数。

```
#include <stdio.h>
```

```
void main()
{
    int n=0;
    printf("input a string:\n");
    while(getchar()!='\n') n++;
    printf("%d",n);
}
```

本例程序中的循环条件为 `getchar () != '\n'`，其意义是：只要从键盘输入的字符不是

回车就继续循环。循环体 $n++$ 完成对输入字符个数计数。从而程序实现了对输入一行字符的字符个数计数。

使用 while 语句应注意以下几点：

1) while 语句中的表达式一般是关系表达或逻辑表达式，只要表达式的值为真（非 0）即可继续循环。

2) 循环体如包括有一个以上的语句，则必须用 `{}` 括起来，组成复合语句。

3) 应注意循环条件的选择，以避免死循环。

4. do-while 语句

do-while 语句的一般形式为：

do

语句；

while（表达式）；

其中，语句是循环体，表达式是循环条件。

do-while 语句的语义是：先执行循环体语句一次，再判别表达式的值，若为真（非 0）则继续循环，否则终止循环。do-while 语句和 while 语句的区别在于 do-while 是先执行后判断，因此 do-while 至少要执行一次循环体。而 while 是先判断后执行，如果条件不满足，则一次循环体语句也不执行。while 语句和 do-while 语句一般都可以相互改写。

对于 do-while 语句还应注意以下几点：

1) 在 if 语句和 while 语句中，表达式后面都不能加分号，而在 do-while 语句的表达式后面则必须加分号。

2) do-while 语句也可以组成多重循环，而且也可以和 while 语句相互嵌套。

3) 在 do 和 while 之间的循环体由多个语句组成时，也必须用 `{}` 括起来组成一个复合语句。

4) do-while 和 while 语句相互替换时，要注意修改循环控制条件。

5. for 语句

for 语句是 C 语言所提供的功能更强、使用更广泛的一种循环语句。其一般形式为：

for（表达式 1；表达式 2；表达式 3）

语句；

其中，表达式 1 通常用来给循环变量赋初值，一般是赋值表达式。也允许在 for 语句外给循环变量赋初值，此时可以省略该表达式。表达式 2 通常是循环条件，一般为关系表达式或逻辑表达式。表达式 3 通常可用来修改循环变量的值，一般是赋值语句。这三个表达式都可以是逗号表达式，即每个表达式都可由多个表达式组成。三个表达式都是任选项，都可以省略。

一般形式中的“语句”即为循环体语句。for 语句的语义是：

1) 首先计算表达式 1 的值。

2) 再计算表达式 2 的值，若值为真（非 0），则执行循环体一次，否则跳出循环。

3) 然后再计算表达式 3 的值，转回第 2 步重复执行。在整个 for 循环过程中，表达式 1 只计算一次，表达式 2 和表达式 3 则可能计算多次。循环体可能多次执行，也可能一次都不执行。

例如：计算 $s = 1 + 2 + 3 + \dots + 99 + 100$

```
void main ( )  
{ int n, s = 0;  
  for ( n = 1; n <= 100; n ++ )  
    s = s + n;  
  printf ( " s = %d \n" , s );  
}
```

另外，for 语句也可与 while 语句、do-while 语句相互嵌套，构成多重循环。

参 考 文 献

- [1] Microchip Technology Inc. PIC16F87X 数据手册——28/40 脚 8 位 FLASH 单片机 [M]. 刘和平, 黄开长, 严利平, 等, 译. 北京: 北京航空航天大学出版社, 2001.
- [2] 李学海. PIC 单片机实用教程——基础篇 [M]. 北京: 北京航空航天大学出版社, 2002.
- [3] 陈建新. PIC 单片机开发应用与实验工具制作 [M]. 北京: 北京航空航天大学出版社, 2006.
- [4] 微芯科技股份有限公司. PIC 单片机宏汇编与集成开发环境 [M]. 北京: 清华大学出版社, 2002.
- [5] 刘和平. PIC16F87X 单片机实用软件与接口技术 [M]. 北京: 北京航空航天大学出版社, 2002.
- [6] 刘笃人. PIC 软硬件系统设计——基于 PIC16F87X 系列 [M]. 北京: 电子工业出版社, 2005.
- [7] 李学海. PIC 单片机实践 [M]. 北京: 北京航空航天大学出版社, 2004.

第 4 章 竞赛机器人的结构与部件

机器人竞赛是近年来国际上迅速开展起来的一种高科技对抗活动，它涉及人工智能、自动控制、机械电子、通信、传感及机构等多个领域的前沿研究和技术融合。本章所述的竞赛机器人平台是为青少年参加各种机器人大赛而搭建的一个硬件平台，在这个硬件平台基础上，可以方便地实现迷宫机器人、寻线机器人、滑雪机器人、灭火机器人等的创建。同时在这个平台可以开展各种单片机编程练习，完成单片机的各种功能。也可以练习各种算法的实现，如深度优先搜索法、广度优先搜索法、A* 搜索法等。下面就来介绍一下机器人平台的整体设计。

4.1 平台的机械结构

竞赛机器人平台的机械结构为层次结构，如图 4-1 所示。层与层之间用标准铜柱支撑，电气连接采用标准插针插座连接，便于拆装和维护。其电路板分为两层：下层是驱动板；上层是控制板。

竞赛机器人驱动板结构如图 4-2 所示。它主要包括左右车轮、两个改造后的舵机、四个色差传感器、万向轮、电气连接插座、H 桥驱动芯片等。迷宫机器人转弯的中心与驱动板的中心一致，这样小车转弯半径最小，行动灵活。

竞赛机器人控制板结构如图 4-3 所示。

竞赛机器人控制板主要包括 CPU 单元、传感器调理电路、传感器驱动电路、遥控器无线接收单元、RS232 通信电路、串行口编程座等。传感器外形如图 4-4 所示。

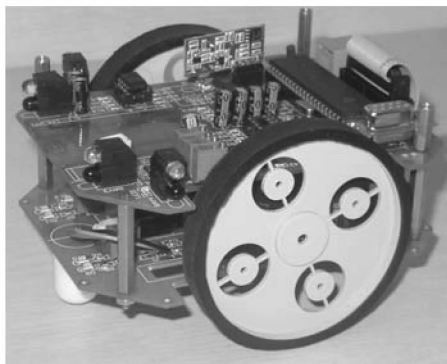


图 4-1 竞赛机器人平台的整体外观

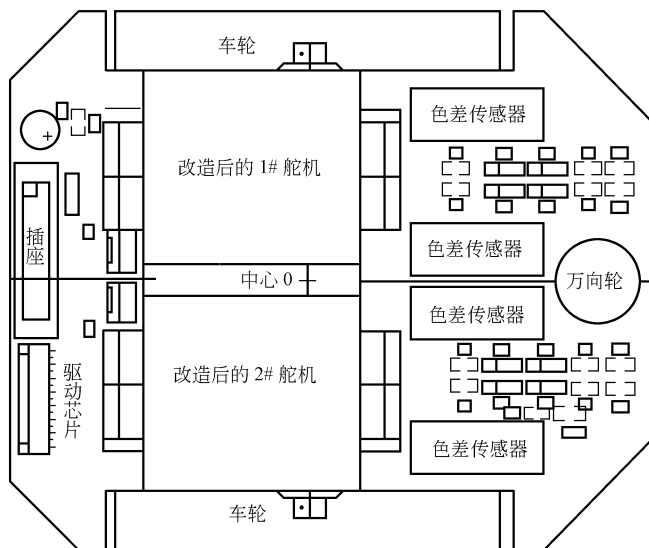


图 4-2 驱动板结构图

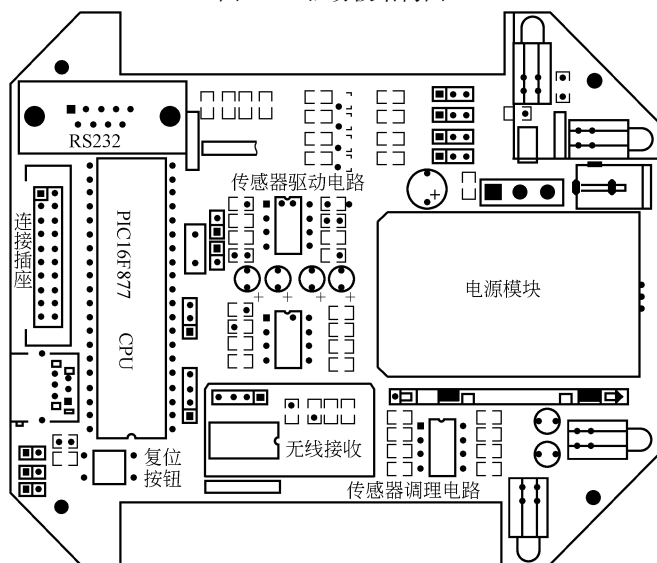


图 4-3 竞赛机器人控制板结构

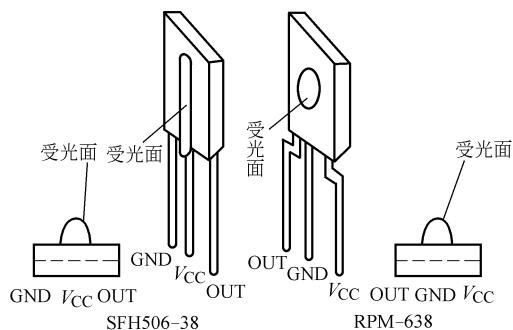


图 4-4 红外传感器外形

4.2 平台的行走机构

行走机构，也称为机器人的下肢，是机器人的关键部分，主要用来承受体重和完成位移，它决定着机器人能否迅速灵活地移动，能否准确地按照操作者的意愿到达指定点。

机器人的行走机构首先要体现稳定性，其次是灵活性。很多工业机器人都不具备行走机构，它们在固定的位置完成固定的动作，因此只需要一个机械臂，也就是我们说的执行机构。但是当生产过程需要一个机器人在较大的范围内工作时，如果不具备行走机构，那么它的体积将非常庞大；如果安置了行走机构，可以使它做适当的位移，在工作范围的不同区间工作，可以节约空间、提高效率。对于竞赛机器人平台来说，行走机构是必不可少的。

机器人的行走方式主要有三种：足式行走、履带式行走和轮式行走。轮式行走机构由滚动摩擦代替滑动摩擦，主要的特点是效率高，适合在平坦的路面上移动，定位准确，而且重量较轻，制作简单。在各类机器人竞赛中，场地通常比较固定，路面状况好，而且又对参赛机器人重量上有限制，所以轮式机器人可以发挥出它高效率的特点，在机器人竞赛中使用得最多。下面就轮式行走方式进行讨论。在机器人的设计中一般常用的轮式结构布置有以下几种：

1. 三轮式车体

三轮式车体的两个驱动轮安装在前边的两侧，只在后端装有一个从动轮，位于两驱动轮的中垂线上，构成三点布置型式，如图4-5所示。使用三个轮的优点是所有的车轮都会着地，不会产生悬空现象，控制稳定。但当机器人重心偏移时，如转弯、相撞等，由于只有三个轮子支撑车体，稳定性不好。

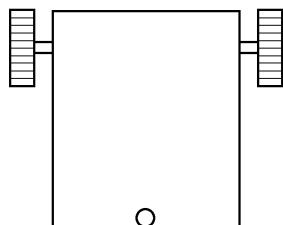


图4-5 三轮式

2. 四轮式车体

四轮式车体四个轮子位于车体的四个角，前边是驱动轮，后边是从动轮，如图4-6所示。较三轮车体来说，四个轮子可以更好地支撑车体，行走稳定，当重心偏移时，左右晃动小。但如果某个车轮的中心不正，或是安装高度有误差，可能导致四个车轮不能同时着地，使控制困难。所以在安装车轮时，一定要保证驱动轮绝对着地。

3. 对称四轮式车体

对称四轮式如图4-6所示，与一般四轮式车体相比，两个驱动轮前移，安装在前后两侧的中线上，两个从动轮前后各一个，四个轮子安装呈“十”字形，如图4-7所示。由于四个轮不是安装在车体的四个角，支撑的稳定性不如图4-6所示的四轮式车体。但是，车体是前后对称的，某些机器人只要求底盘是前后对称的（如要求原地转弯），可采用此驱动方式。由于小车是靠驱动轮与地面产生附着力改变运动状态，因此不论何种布置形式，在小车的运动过程中的某一瞬时状态只能是三点支撑。即要么前一个被动支撑轮与两个驱动轮同时着地，要么后一个被动支撑轮与两个驱动轮同时着地。虽然，由于驱动轮上轮胎的弹性会使得在小车的运动过程中的某一瞬时呈现四轮着地的现象，但这种情况应尽量避免出现，多一个被动支撑轮着地只会增加小车与地面的摩擦力，从而消耗更多的能量（由于小车尺寸受到严格限制，车载能量是有限的）。在小车的运动过程中，绝对不允许出现两个驱动轮离开地面的现象。竞赛机器人平台是集多种功能于一身的，要求机器人平台可以原地左转或者原地

右转，为了保证机器人在转弯过程中的灵活性，这里选择了对称四轮式车体结构。

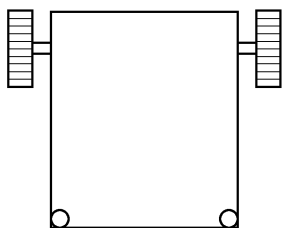


图 4-6 四轮式

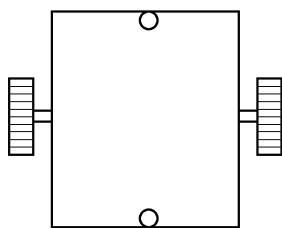


图 4-7 对称四轮式

4.3 舵机

舵机最早出现在航模运动中。在航空模型中，飞行器的飞行姿态是通过调节发动机和各个控制舵面来实现的，舵机因此得名。控制舵面的伺服电动机，不仅在航模飞机中，在其他的模型运动中都可以看到它的应用。船模上用来控制尾舵，车模中用来转向等。由此可见，凡是需要操作性动作时都可以用舵机来实现。

1. 舵机的结构

一般来讲，舵机主要由舵盘、减速齿轮组、位置反馈电位器（5k）、直流电动机、控制电路板等组成，如图 4-8 所示。实际的舵机又有许多区别，例如直流电动机就有有刷和无刷之分，齿轮有塑料和金属之分，输出轴有滑动和滚动之分，速度有快速和慢速之分等，组合不同，价格也千差万别。例如，其中小舵机一般称作微舵，同种材料的条件下价格是中型的一倍多，金属齿轮价格是塑料齿轮的一倍多。需要根据需要选用不同类型。常见的舵机厂家有：日本的 Futaba、JR、SANWA 等公司，国产的有北京的新幻想、吉林的振华等公司。

舵机的瞬时运动速度是由其内部的直流电动机和变速齿轮组的配合来决定的，在恒定的电压驱动下，其数值是恒定的。但其平均运动速度可通过分段停顿的控制方式来改变，例如，把动作幅度为 90° 的转动细分为 128 个停顿点，通过控制每个停顿点的时间长短来实现 $0^\circ \sim 90^\circ$ 变化的平均速度。对于多数舵机来说，速度的单位是“度/秒”。

标准的舵机有三条引线，分别为：电源线 VCC、地线 GND 及控制线 CON，如图 4-9 所示。电源线与地线用于提供内部的直流电动机及控制线路所需的能源，电压通常介于 $4 \sim 6V$ 之间，该电源应尽可能地与 CPU 系统的电源隔离（因为舵机会有噪声）。甚至小舵机在重负载时也会拉低放大器的电压，所以整个系统的电源供应的比例必须合理。

2. 舵机的工作原理

舵机是一个典型闭环反馈系统，其工作原理如图 4-10 所示。

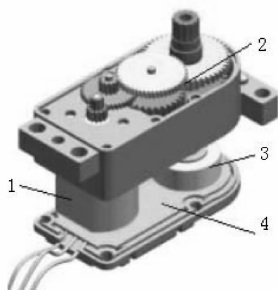


图 4-8 舵机的结构
1—直流电动机 2—变速
齿轮组 3—反馈电位器
4—控制电路板

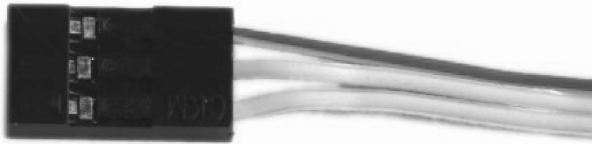


图 4-9 舵机的引线

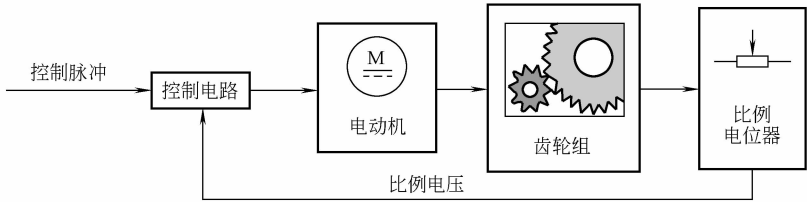


图 4-10 舵机工作原理框图

减速齿轮组由电动机驱动，其齿轮组的输出轴带动一个线性的比例电位器，该电位器把输出轴转过的角度 θ 转换成比例的电压反馈给控制电路，控制电路将其与输入的控制脉冲信号比较，产生纠偏脉冲，并驱动电动机正向或反向地转动，使齿轮组的输出位置与期望值相符，令纠偏脉冲趋于 0，从而达到使舵机精确定位的目的。

给舵机输入一个周期性的正向脉冲信号，这个周期性脉冲信号的高电平时间通常在 1 ~ 2ms 之间，而低电平时间应在 5 ~ 20ms 之间，并不很严格，表 4-1 所示为舵机的输入正脉冲宽度与舵机输出轴的转角的关系。

表 4-1 输入脉冲宽度与输出轴的转角的关系

输入正脉冲宽度（周期为 20ms）	直流伺服电动机输出臂位置
	$\approx -90^\circ$
	$\approx -45^\circ$
	$\approx 0^\circ$
	$\approx 45^\circ$
	$\approx 90^\circ$

3. 使用舵机时的注意事项

普通的模拟舵机不是一个精确的定位器件，即使是使用同一品牌型号的舵机产品，它们

之间的差别也是非常大的，在同一脉冲宽度驱动时，不同的舵机输出位置存在 $\pm 10^\circ$ 的偏差也是正常的。

特别注意，绝不可加载让舵机输出位置超过 $\pm 90^\circ$ 的脉冲信号，否则会损坏舵机的输出限位机构或齿轮组等机械部件。

由此可见，舵机具有以下一些特点：体积紧凑，便于安装；输出力矩大，稳定性好；控制简单，便于和数字系统接口。

正是因为舵机有很多优点，所以现在不仅仅应用在航模运动中，已经扩展到各种机电产品中来，在机器人控制中应用也越来越广泛。

4.4 将舵机改装成执行机构

众多的舵机生产厂商生产形状、尺寸和输出力矩各异的舵机，从市场上可以购得的舵机数以百计。在选取舵机时一般应该考虑以下问题：

- 1) 被选择的舵机能否改装为连续转动；
- 2) 输出力矩的大小；
- 3) 是否允许超过额定电压的限度；
- 4) 转速的高低；
- 5) 如何控制舵机。

一些舵机即使改装之后也无法变为连续旋转的，因为其花键齿轮并非是在整个圆周上都有齿轮。因此选择了性价比较高，同时也是比较常见的 Futaba S3003 舵机作为行走电动机。

在 Futaba S3003 舵机中有一个限位装置，使得舵机只能旋转 180° ，而竞赛机器人平台的行走电动机需要连续旋转，需要将 Futaba S3003 舵机进行改装。

1. 所需器件

- 1) 导线；
- 2) 若干焊锡丝；
- 3) Futaba S3003 舵机。

2. 改装工具

- 1) 刻刀；
- 2) 吸锡器；
- 3) 60/40W 电烙铁；
- 4) 小螺钉旋具；
- 5) 尖嘴钳。

3. 改装步骤

舵机外形如图 4-11 所示，按以下步骤改装 Futaba S3003 舵机：

1) 要拆掉舵机的顶盖，需要首先用小螺钉旋具拆卸舵机的舵盘。舵盘是用螺钉连接在花键轴上的，用小螺钉旋具拧松螺钉以后舵盘就和花键轴分离了。注意在松开螺钉时一定要抓紧舵盘，不然的话可能会损坏舵机。拆卸后的舵盘和舵机如图 4-12 所示。

2) 将舵机翻转，看到舵机的底部，如图 4-13 所示。使用小螺钉旋具拧松底面的螺钉。

3) 抓紧外壳，并将外壳按向花键轴，拆下顶盖。注意保护舵机顶部外露的其他齿轮。



图 4-11 Futaba S3003 舵机外形图



图 4-12 拆掉舵盘的舵机和小螺钉旋具



图 4-13 舵机底部

舵机内部的齿轮组如图 4-14 所示。



图 4-14 舵机齿轮组

4) 如图 4-15 所示，拆除黑色的花键齿轮以及上方的白色齿轮。将舵机内部齿轮组取下来以后，可以看到一个金属矩形块，它是反馈电位器的调整片，可以通过它来调整和设定舵机的中心点。



图 4-15 拆掉的花键齿轮

5) 如图 4-16 所示，将花键齿轮放在一个平的硬质表面上，小心地将刻刀放到齿轮上，保持与齿轮面平行。切除止动齿一直切到主轴，但不要切得过深，以免影响主轴的转动。不要用偏口钳之类的切割器切除止动齿，否则齿轮会因为受力不均而破碎，这样做的后果是操作者很难预料哪些齿轮在切除止动齿后仍安然无恙，而哪些会破损。

6) 切除止动齿后需要使用简易刀具修整齿轮轮廓，修正后的花键齿轮如图 4-17 所示。

7) 舵机顶盖如图 4-18 所示，为了使舵机可以作圆周运动，需要刀具切除顶盖两侧的限位齿。在切除限位齿的过程中应该特别小心，注意保持顶盖表面的平整。图 4-19 为切除限位齿的顶盖内部结构。



图 4-16 舵机的止动齿



图 4-17 拆除限位的舵机止动齿



图 4-18 舵机顶盖



图 4-19 拆除限位齿的舵机顶盖

8) 舵机变成行走电动机以后不再需要原有的控制电路。完整的舵机一共有三根线，其中两根为电源线，一根为信号线。将这三根线全部剪断，然后将其中的两根电源线直接接到电动机的两个端子上，如图 4-20 所示。



图 4-20 拆除底盖后的舵机

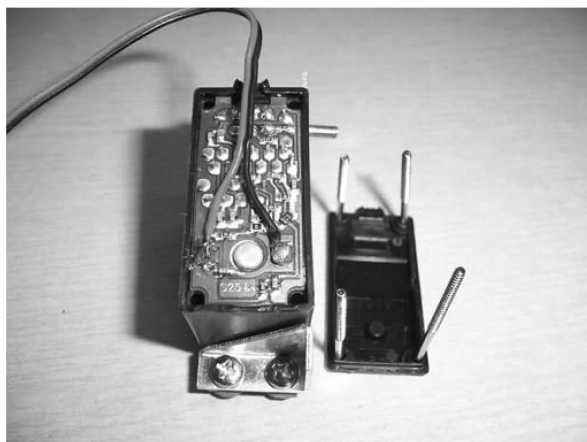


图 4-21 修改电路后的舵机

9) 重装舵机之前需要确定花键齿轮是否可以连续转动。如果在转动过程中有摩擦, 或者有轻微的上下的振动, 说明在转动过程中舵机的止动齿与顶盖的限位齿之间有干涉, 需要再重新修整一下舵机的止动齿和顶盖的限位齿, 直到可以连续平滑地转动花键齿轮为止。

10) 小心地合上舵机的顶盖, 拧紧螺钉, 直到舵机壳体与顶盖紧密贴合。至此, 改装 Futaba S3003 舵机的工作即告完工, 如图 4-21 所示。

4.5 竞赛机器人平台的组装

前面已经介绍了将舵机改装成直流电动机的方法, 这一节将要介绍如何搭建起竞赛机器人平台。考虑到机器人所要实现的功能, 这里将竞赛机器人平台的结构设置为双层结构, 每层结构都有一层印制电路板 (PCB) 构成。这样既完成了电路部分的设计制作, 又搭建起了竞赛机器人的平台。PCB 分为两层, 下层是驱动板, 上层是控制板。

竞赛机器人驱动板结构图如图 4-22 所示, 主要包括: 左右车轮、两个改造后的舵机、四个色差传感器、万向轮、电气连接插座、H 桥驱动芯片等。驱动板的功能是: 接受控制板来的控制信号, 驱动两台直流电动机, 完成竞赛机器人的前进、原地左转、原地右转、停止、后退。

驱动板层与舵机通过角钢来连接, 在固定时需要在舵机与螺栓之间放两个垫片。垫片一般用在连接件中, 一个是软质地的, 另一个是硬质地较脆的, 其主要作用是增大接触面积, 分散压力, 防止把质地软的压坏。这里垫片还有一个作用就是防止松动。但若安装不当, 则不会起到防止螺栓松动的作用。正确的安装方向应该是将弹簧垫片的凹面朝向连接件, 凸面朝向螺母。这样螺母拧紧后, 由于垫片受力变形, 支撑表面受到反作用力而起到锁紧的作用。

舵机与轮子的连接是通过一个黑色的连接件组装起来的, 如图 4-23 所示。连接件相对较小, 将连接件与轮子直接组装起来并不容易, 因此在组装的过程中先将连接件装在舵机的舵盘上, 然后再将舵机上的连接件装在轮子上。安装好的轮子如图 4-24 所示。

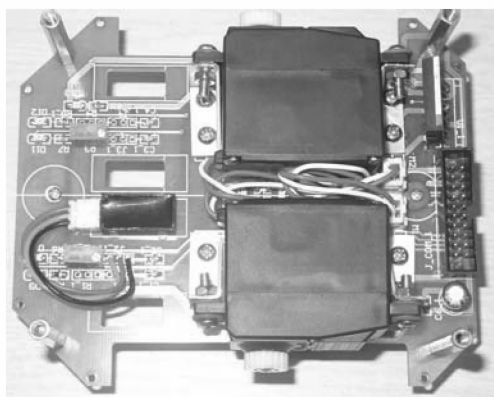


图 4-22 竞赛机器人驱动板



图 4-23 舵机、轮子与连接件

在安装好轮子以后，将整体固定在底层驱动板上，固定的过程中仍然要注意在螺母与电路板的连接处加一个垫片，防止将电路板压坏。图 4-25 为组装以后的底层驱动板全貌。

竞赛机器人控制板结构图如图 4-26 所示，控制板层主要包括 CPU 单元、传感器调理电路、遥控器无线接收单元、RS232 通信电路、串行口编程座等部分。

控制板的功能如下：

- 1) 调理四个红外传感器来的信号。
- 2) 处理四个红外传感器来的信号，通过智能判断程序，决定竞赛机器人的运动状态。
- 3) 把电源电压稳压成单片机可以正常工作的电压以及给电池进行充电。
- 4) 遥控接收模块接收到遥控器发射出来的信号，执行遥控器的命令。



图 4-24 安装好的轮子

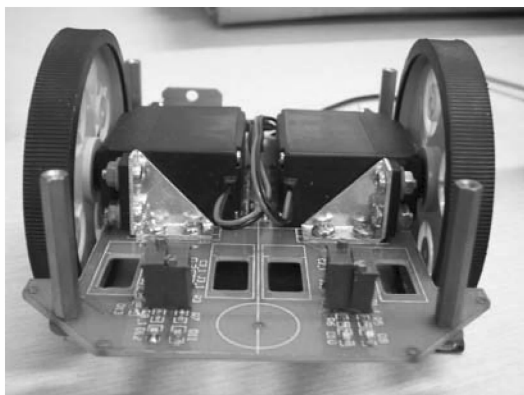


图 4-25 底层驱动板全貌

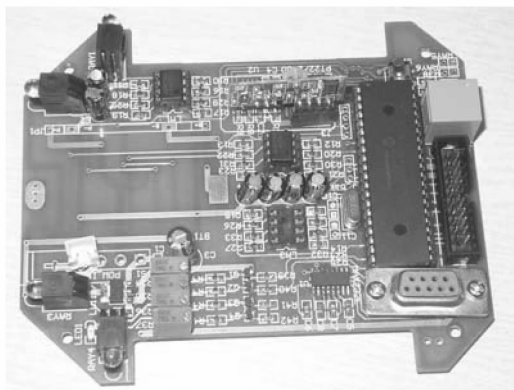


图 4-26 竞赛机器人控制板结构

5) 接收上位机来的信号，执行上位机的命令（包括迷宫的指定路径，前后左右等方向信息），具体命令参见表 4-2。

表 4-2 上位机来的命令

	功能	1#电动机（左侧）	2#电动机（右侧）	备注
1	前进	正转	正转	相同的速度
2	左转	减速正转	加速正转	
3	右转	加速正转	减速正转	
4	180°	正转	反转	车的中心不变
5	停车	给 0V 电压	给 0V 电压	制动

如图 4-1 所示。层与层之间用标准铜柱支撑，电气连接采用标准插针插座连接，便于拆装和维护。

4.6 竞赛机器人专用遥控器的制作

迷宫机器人的运动方式共分为前进、倒退、左转和右转，故选用手柄式的遥控器最为人性化，便于操作。但是市面上没有手柄式的遥控器，因此需要对从市场上买到的遥控器进行改装。下面就介绍一下如何将它装到手柄中。

1. 所需器件

- 1) 与接收器配套的普通遥控器（待改装的遥控器），外形如图 4-27 所示；

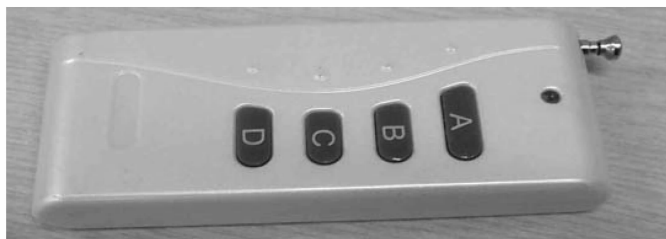


图 4-27 待改装的遥控器外形

- 2) 手柄，外形如图 4-28 所示；
- 3) 导线；
- 4) 若干焊锡丝。



图 4-28 手柄外形

2. 改装工具

- 1) 小螺钉旋具；
- 2) 60/40W 电烙铁；
- 3) 万用表；
- 4) 吸锡器。

3. 改装步骤

- 1) 用螺钉旋具将待改装的遥控器背面的螺钉松开，打开遥控器后盖，取出其中的 PCB（印制电路板）。可发现在 PCB 上有四个按键，如图 4-29 所示。

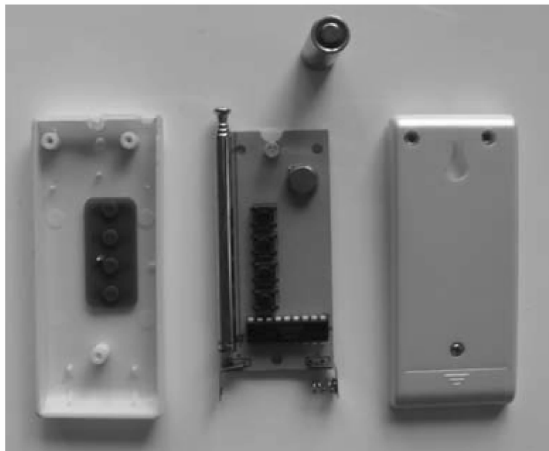


图 4-29 遥控器内部视图

2) 每个按键有四个引脚, 利用万用表可检测出当按键按下时, 哪两个引脚由断开变为接通, 将这两个引脚记住, 以备后面所用, 之后将按键除去。对四个按键都按此步处理, 注意不要损坏 PCB。

3) 用螺钉旋具将手柄背面的螺钉松开, 打开手柄后盖, 取出其中的 PCB, 如图 4-30 所示。注意打开后盖时要小心按键掉落。

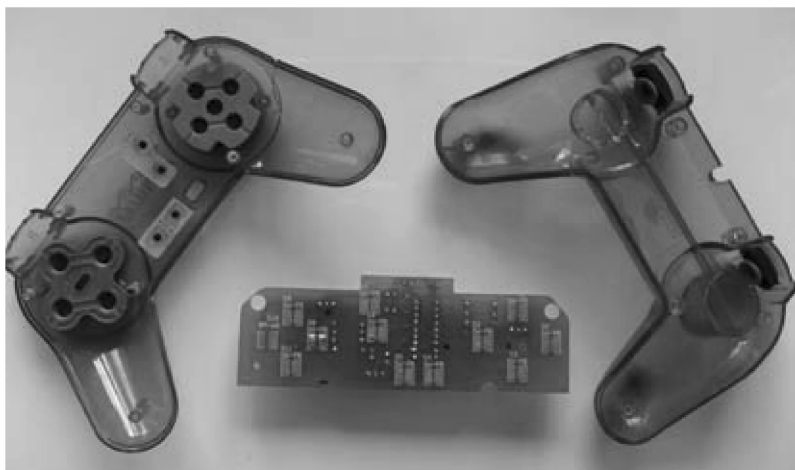


图 4-30 手柄内部视图

4) 将手柄中 PCB 上的全部芯片以及导线除去。

5) 若希望手柄左侧的方向键与右侧的四个键功能同步, 即有相同的功能。可将手柄内的 PCB 左右两侧对应的触板用导线连通。

6) 将遥控器中的按键上所作标记的引脚与手柄内的 PCB 上的按键触板对应相连。即作标记的引脚之一与触板左板片相连, 另外一个引脚与触板右板片相连。注意遥控器上的按键与手柄的方向键的功能一一对应, 如图 4-31 所示。

7) 将电池焊在对应位置, 将改装好的 PCB 装回手柄内, 拧紧螺钉, 改装完成。

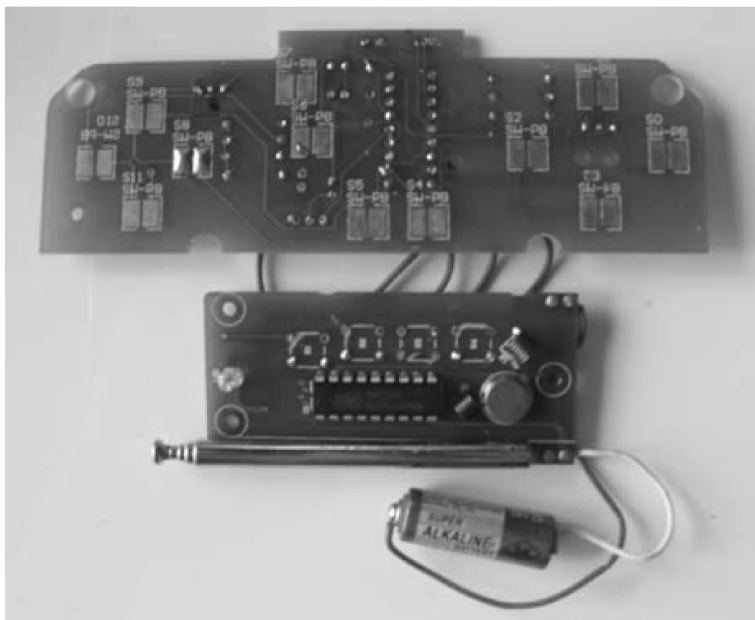


图 4-31 手柄 PCB 与遥控器 PCB 间的连接

参 考 文 献

- [1] 张文增, 孙振国, 曹一鹏, 等. 机器人微型电机驱动器的设计与实现 [J]. 微特电机, 2004 (2).
- [2] 陈定明. 电机与控制 [M]. 北京: 高等教育出版社, 2004.
- [3] 曾峰, 侯亚宁, 曾凡雨. EDA 工具应用丛书——印刷电路板 (PCB) 设计与制作 [M]. 北京: 电子工业出版社, 2002.
- [4] 王晓明. 电动机的单片机控制 [M]. 北京: 北京航空航天大学出版社, 2002.
- [5] 柯南. 非常电路板设计 Protel 99 之 PCB [M]. 北京: 中国铁道出版社, 2000.

第 5 章 竞赛机器人的电子电路

上一章介绍了竞赛机器人的整体结构的设计，也就是机器人有了自己的“骨架”，但是还不能运动起来，还需要将“骨架”连起来组合成一个整体，从而实现对机器人的控制。硬件电路就是实现这样的功能，下面就来介绍一下机器人平台的硬件电路设计。

5.1 硬件电路组成

竞赛机器人的硬件主要包括：执行机构驱动电路、色差传感器检测原理、迷宫隔栅检测、无线发射接收模块、串行口传输模块、数据存储模块、迷宫场地。

5.2 执行机构驱动电路

经过改造后的舵机相当于带减速箱的直流电动机。由于舵机的性能优越，舵机的一致性很好，当两个电动机加同样的电压时，电动机的转速基本一致，误差很小。直流电动机的物理模型如图 5-1 所示。其中，固定部分有磁铁，这里称作主磁极；固定部分还有电刷。转动部分有环形铁心和绕在环形铁心上的绕组。图 5-1 表示一台最简单的两极直流电动机模型，它的固定部分（定子）上，装设了一对直流励磁的静止的主磁极 N 和 S，在旋转部分（转子）上装设电枢铁心。定子与转子之间有一气隙。在电枢铁心上放置了由导体连成的电枢线圈，线圈的首端和末端分别连到两个圆弧形的铜片上，此铜片称为换向片。换向片之间互相绝缘，由换向片构成的整体称为换向器。换向器固定在转轴上，换向片与转轴之间亦互相绝缘。在换向片上放置着一对固定不动的电刷 B1 和 B2，当电枢旋转时，电枢线圈通过换向片和电刷与外电路接通。

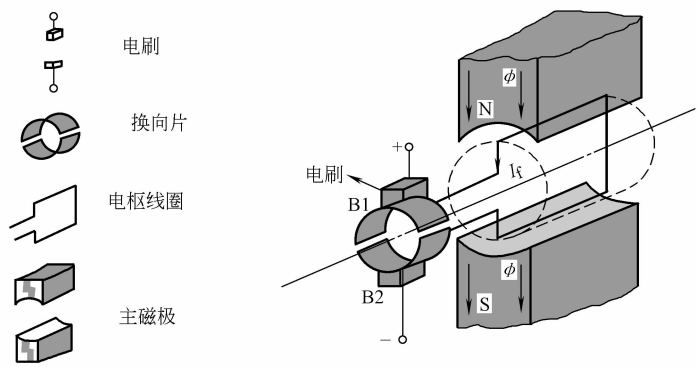


图 5-1 直流电动机的物理模型

设原动机拖动转子转速为 n ，电动机内部的固定部分要有磁场。这个磁场是由图 5-1 所示的永久磁铁和磁极铁心上绕套的线圈，再通过直流电产生的。其中， I_f 为励磁电流。这种

线圈每个磁极上有一个，也就是，电动机有几个磁极就有几个励磁线圈，这几个线圈串联（或并联）起来就构成了励磁绕组。这里要注意各线圈通过电流的方向不可出错。在以上条件下环外导体将感应电动势，其大小与磁通密度 B 、导体的有效长度 l 和导体切割磁场速度 v 三者的乘积成正比，其方向用右手定则判断。对某一根转子导体的电动势性质是交流电，而经电刷输出的电动势却是直流电了。这便是直流电动机的工作原理。

竞赛机器人的电动机需要正反转控制，为此这里采用 L298 双通道直流电动机驱动芯片来驱动。L298 是一块单片集成电路，有两种封装形式：15 脚直立封装（Multiwatt15）和 20 脚贴片封装（PowerSO20），如图 5-2 所示。L298 是双 H 桥高电压大电流功率集成电路，可以用来驱动 2 个直流电动机，也可以用来驱动感性负载，如继电器、电磁阀、步进电动机等。采用 L298 作为电动机驱动电路，可靠性高，可以方便地控制电动机的正反转。

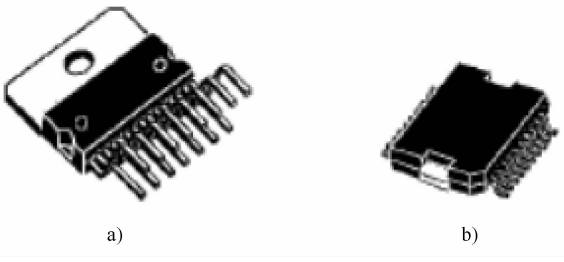


图 5-2 L298 的两种封装形式
a) Multiwatt15 b) PowerSO20

L298 有两个独立的使能端，其内部框图如图 5-3 所示。

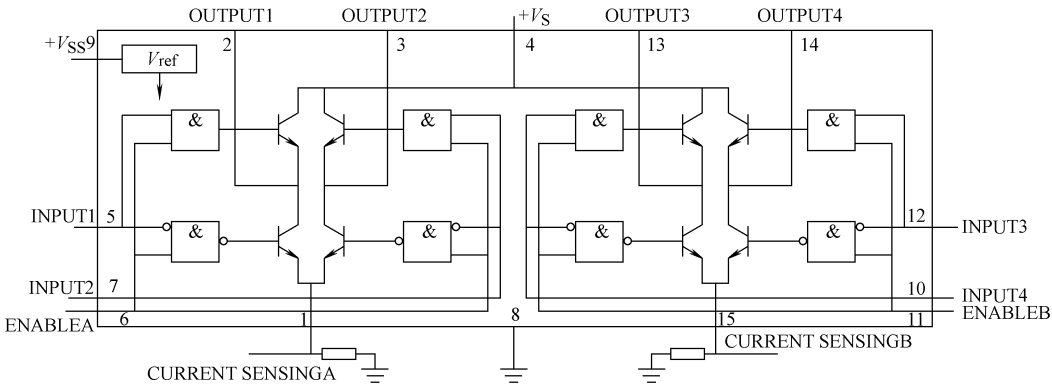


图 5-3 L298 内部框图

L298 引脚排列如图 5-4 所示。特别要注意图中看起来是短的引脚，实际是弯的引脚。

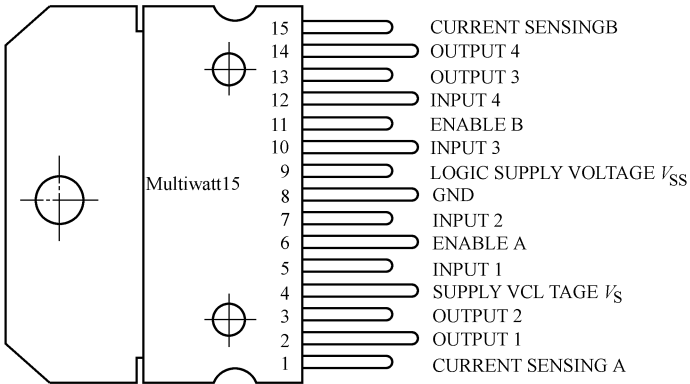


图 5-4 L298 的引脚排列

L298 引脚名称和功能见表 5-1。

表 5-1 L298 引脚名称和功能

MW. 15	名称	功 能 说 明
1; 15	CURRENT; SENSINGA; CURRENT; SENSINGB	与地之间接检测电阻，用来控制负载电流
2; 3	OUTPUT1; OUTPUT2	AH 桥的输出端子，这两个端子流过负载电流
4	V_s	功率输出的电源电压端子
5; 7	INPUT1; INPUT2	AH 桥的 TTL 电平输入
6; 11	ENABLEA; ENABLEB	使能端子
8	GND	电源地
9	V_{ss}	逻辑电源电压端子
10; 12	INPUT3; INPUT4	BH 桥的 TTL 电平输入
13; 14	OUTPUT3; OUTPUT4	BH 桥的输出端子，这两个端子流过负载电流

L298 内部设计了两个可以接受标准的 TTL 逻辑电平的高工作电压、高工作电流的全 H 桥驱动电路。功率电源电压 V_s 范围为 2.5 ~ 46V，逻辑电源电压 V_{ss} 工作范围：5 ~ 7V。最大静态工作电流是 70mA（ $V_s = 42V$ ， $V_{ss} = 5V$ ， $T_j = 25^{\circ}C$ ）。

L298 的主要参数：

- 1) 允许驱动电压： $V_s = 50V$ ；
- 2) 允许控制电压： $V_{ss} = 50V$ ；
- 3) 直流允许输出电流： $I_{O(MAX)} = 2A$ ；
- 4) 直流最大允许尖峰电流： $I_{O(peak)} = 3A$ （ $t = 100\mu s$ ）；
- 5) 允许采样电压： $V_{SENSE} = -1 \sim 2.3 V$ ；
- 6) 允许功耗： $P_{tot} = 25W$ （ $T_{CASE} = 75^{\circ}C$ ）。

注意：L298 每一路输出正常可以提供 2A 的电流，峰值电流（100 μs 之内）可达 4A。
竞赛机器人的驱动电路如图 5-5 所示。

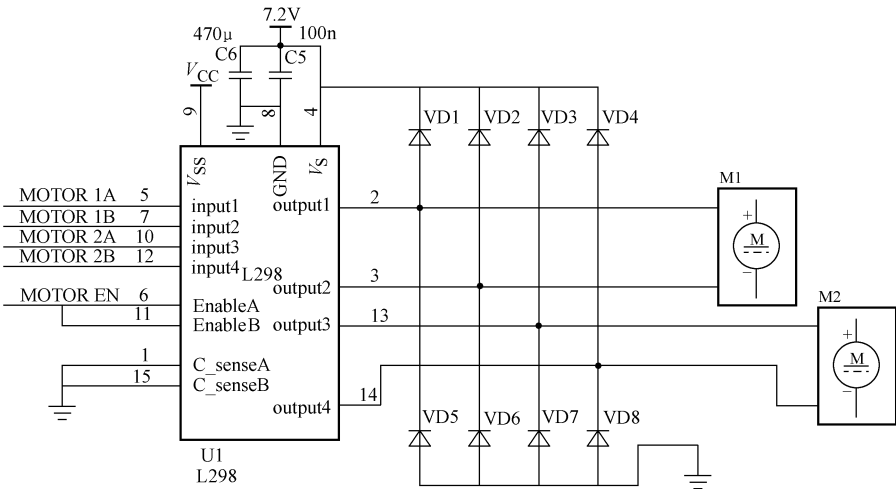


图 5-5 竞赛机器人的驱动电路

其中，M1、M2 是直流电动机；VD1 ~ VD8 是整流二极管。当 Enable A 和 Enable B 使能端为高电平、输入端 input1、input2 分别为 1、0 时，M1 正转；input1、input2 分别为 0、1 时，M1 反转。同理 input3、input4 分别为 1、0 时，M2 正转；input3、input4 分别为 0、1 时，M2 反转。竞赛机器人的运动状态见表 5-2。

表 5-2 竞赛机器人的运动状态

ENABLE A ENABLE B	INPUT1	INPUT2	INPUT3	INPUT4	左轮	右轮	状态
均为 0	×	×	×	×	静止	静止	静止
均为 1	0	0	0	0	静止	静止	停止
均为 1	0	1	1	0	正转	正转	前进
均为 1	0	1	0	0	正转	翻转	右转
均为 1	1	0	1	0	反转	正转	左转

注：0 为低电平，1 为高电平，×为任意电平。

5.3 传感器检测电路

传感器是以一定的准确度把被测量转换为与之有确定对应关系的、便于应用的另一种量的一种测量装置。换一种说法，传感器是把非电量转换成电量的装置。传感器通常由敏感元件、转换元件和测量电路三部分组成。敏感元件是指能直接感受（或响应）被测量的部分，即将被测量通过传感器的敏感元件转换成与被测量有确定关系的非电量或其他量。转换元件则将上述非电量转换成电参量。测量电路的作用是将转换元件输入的电参量经过处理，转换成电压、电流或频率等可测量，以便进行显示、记录、控制和处理。测量电路中，较多使用电桥电路。应该注意的是，有些敏感元件可以直接输出电参量，也就是说，没有敏感元件与转换元件之分。

我们可以这样理解：传感器是测量装置，能完成检测任务；传感器的输入量是某一被测量，比如物理量、化学量、生物量等；传感器的输出量是某种被测量，这种量要便于传输、转换、处理、显示等，例如气、光、电量等；传感器的输入和输出有对应的关系，且有一定的准确度。

如图 5-6a 所示，假如开关闭合，灯就会被点亮。反过来，把开关断开，灯自然会被熄灭。如图 5-6b 所示，如果这个开关不是用手来控制开或者关的，而是由外部的因素决定，比如光、声等，那么这个开关就变成了一个传感器。

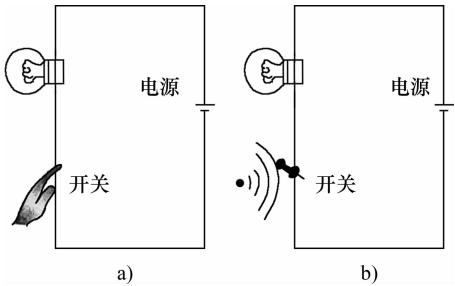


图 5-6 传感器示意图

a) 用手控制开关电路 b) 用传感器控制电路通断

5.3.1 起跑线的检测

常用的竞赛智能车巡线方案有光电传感器阵列和 CCD 探测法等，这里只介绍光电传感器阵列的设计方案。光电传感器是由红外发射管和红外接收管组成的传感器，如图 5-7 所示。



图 5-7 红外传感器地面颜色识别原理

红外发射管发射的红外线具有一定的方向性，当红外线照射到白色表面上时会有较大的反射，如果距离 D_1 取值合适，则红外接收管可接收到反射回的红外线，再利用红外接收管的电气特性，在电路中处理红外线的接收信息；如果反射表面为黑色，则红外光会被表面将其大部分吸收，红外接收管就难以收到红外线。这样，就可以利用红外收发管组成的光电传感器检测区分赛道上的黑白线，实现智能车的巡线方案。

5.3.2 迷宫隔栅检测部分

用于竞赛机器人比赛的场地如图 5-8 所示，竞赛机器人需要准确地检测迷宫隔栅的有无及距隔栅的距离。

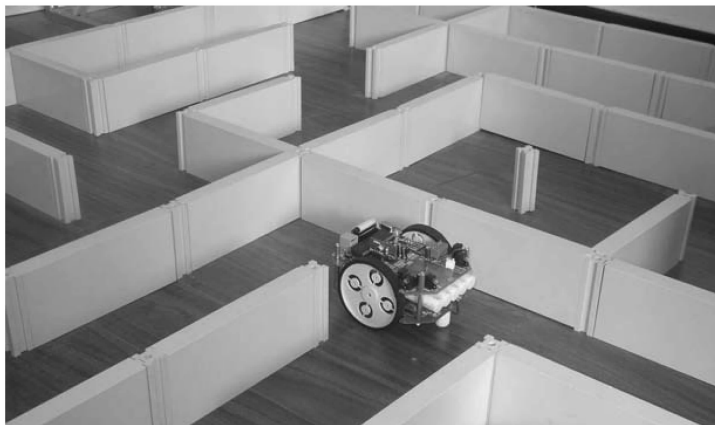


图 5-8 竞赛机器人走迷宫的比赛场地

1. 迷宫隔栅检测电路

迷宫隔栅检测电路如图 5-9 所示，晶体管 V1 用来驱动红外发射管，红外接收管接收迷宫隔栅反射回来的红外光，并转化为对应的电压信号 V_{in} ，经过电压跟随器放大隔离，最后

输出与测量距离 x 对应的电压 V_{out} 。

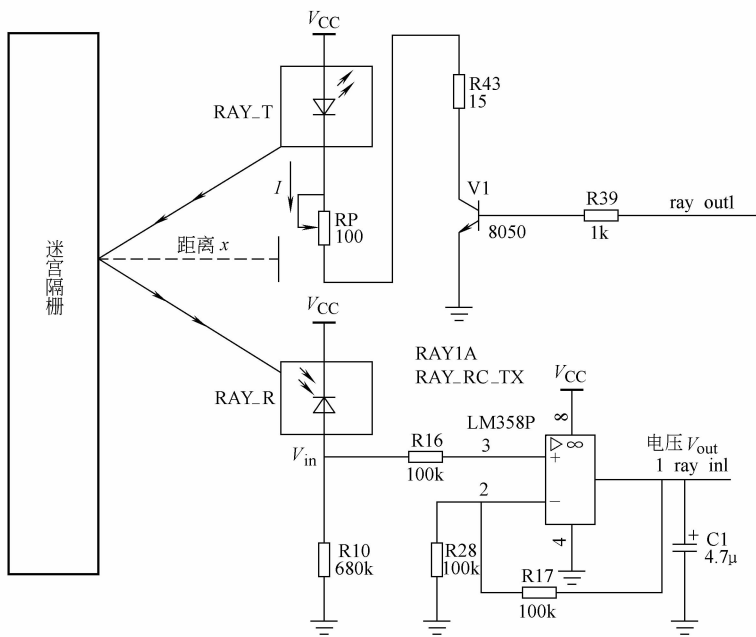


图 5-9 迷宫隔栅检测电路

传感器采用红外光调制解调技术，大大增强抗干扰能力，能准确检测出有无隔栅。其工作原理是：通过红外发射管发射特定频率的红外线，当照射到隔栅上，就会发生反射，红外接收管不断检测反射光，从而实现对物体的检测。红外信号需要调制到 38kHz 的载波上，这样，一方面与生活中常见的普通红外线区别开来，另一方面可以增大检测距离。红外接收电路采用型号为 TSOP1738 的红外线接收管，其内部已经集成了专用电路，具有调谐、放大和检测信号的功能。TSOP1738 的红外线接收管的输出是解调出来的电平信号，可以直接输入单片机，使用非常方便。参考电路如图 5-10 所示。

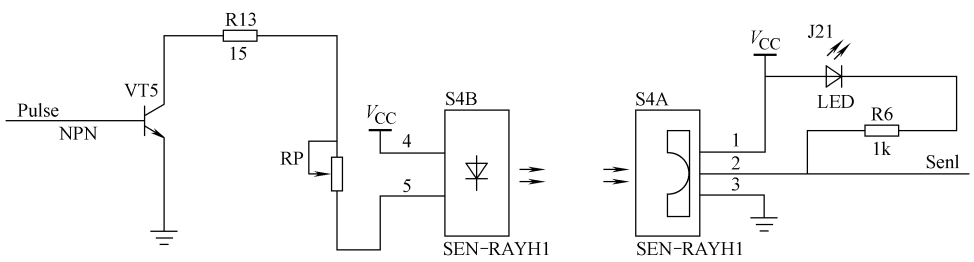


图 5-10 红外传感器的调制发射和接收

红外线是一种只有少数生物才能感觉到的光波，其波长为 1 ~ 1000nm，具有定向传播和反射能力。移动机器人上的红外传感器探测原理为：红外传感器发出红外信号并检测返回的信号，通过返回信号的强度并依据环境物体的反射系数估计传感器本身到目标物体的距离，红外传感器的探测特性与超声传感器相反，即角度分辨率高，而距离分辨率低。当然，它同样具有灵敏度高、响应速度快、结构简单、成本低等优点，其缺点是探

测距离依赖于物体反射的光强度。在环境未知的情况下，环境中的障碍物的反射系数是无法估计和确定的。

2. 驱动红外发射管的脉冲信号

红外发射管发射的是光脉冲信号，如图 5-11 所示。四个红外发射管轮流发射，提高了抗干扰能力。脉冲的频率为 40kHz，高电平时间是 5ms，低电平时间是 20ms。采用窄脉冲驱动红外发射管，可以提高检测的可靠性。

驱动红外发射管的窄脉冲发射电流的幅值是： $I = (V_{cc} - V_{ce} - V_{ray_out}) / R = 50\text{mA}$ 。脉冲的占空比是 $q = 1/4$ 。平均发射电流是 $50/4\text{mA} = 12.5\text{mA}$ 。

3. 检测隔栅的四个传感器的布置

竞赛机器人检测隔栅的四个传感器的布置如图 5-12 所示。其中，1 号用来检测左侧隔栅，2 号、3 号用来检测前方隔栅，4 号用来检测右侧隔栅。测距方面采用模拟传感器，解决探测墙壁的有无和沿着跑道中线前进的问题。

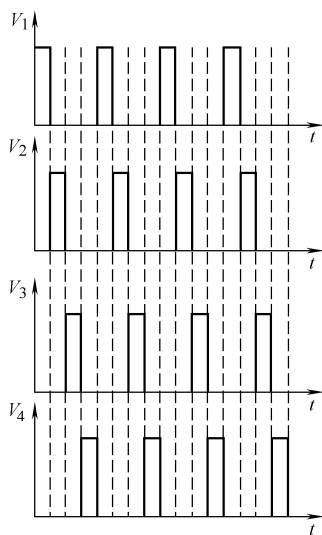


图 5-11 红外发射管发射的脉冲信号

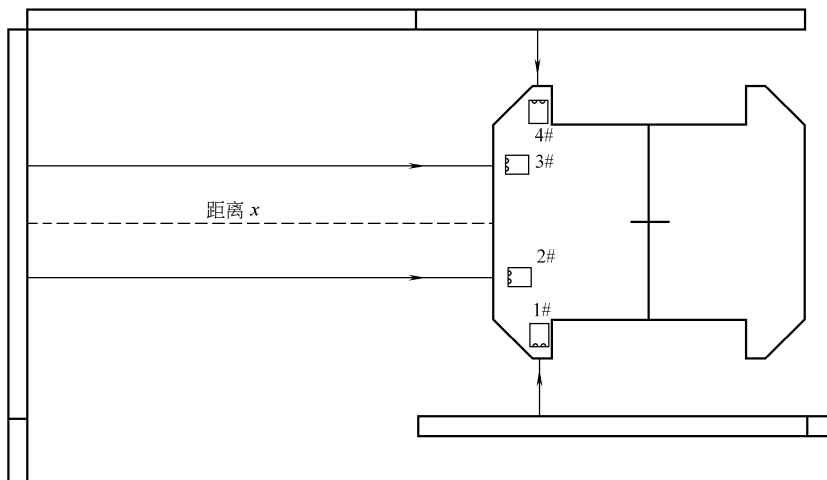


图 5-12 检测隔栅的四个传感器的布置

5.3.3 传感器的特性曲线

为了了解传感器输出电压 V_{out} 与检测距离 x 之间的关系，做了大量的试验。首先需要校准基准，依次将竞赛机器人的传感器放置于距迷宫隔栅 35mm 处，通过示波器观察输出电压 V_{out} 的波形，如图 5-13 所示，仔细调整可调电位器，使输出 V_{out} 为 2.0V。调整传感器与迷宫隔栅的间距，每改变 5mm 的距离，记录一次输出电压 V_{out} ，试验数据见表 5-3。

表 5-3 传感器输出电压 V_{out} 与检测距离 x 之间的关系

传感器编号	距离/cm	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
1#	输出电压/V	3.3	3.3	3.3	3.3	3.2	2.9	2.4	2.0	1.75	1.5	1.35
2#		3.3	3.3	3.3	3.3	3.2	2.9	2.4	2.0	1.75	1.5	1.4
3#		3.3	3.3	3.3	3.2	3.2	2.75	2.4	2.0	1.8	1.55	1.4
4#		3.3	3.3	3.25	3.25	3.15	2.7	2.4	2.0	1.8	1.7	1.5
传感器编号	距离/cm	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	20
1#	输出电压/V	1.2	1.1	1.05	0.95	0.8	0.8	0.75	0.7	0.7	0.65	
2#		1.5	1.3	1.25	1.15	1.1	1.05	1	0.95	0.9	0.85	
3#		1.3	1.2	1.1	1.0	0.9	0.9	0.8	0.75	0.75	0.75	
4#		1.45	1.3	1.25	1.2	1.1	1.1	1.0	0.9	0.9	0.9	0.7

传感器输出电压 V_{out} 与检测距离 x 之间的关系曲线如图 5-14 所示。

从图 5-14 可以看出，每组发射接收传感器的变化趋势基本一致。由于迷宫的跑道宽约为 17.2cm，小车宽为 10cm，竞赛机器人在跑道的中线时，左右传感器距跑道边的距离约是 3.5cm。以左侧传感器（1#）为例，从表 5-3 及图 5-14 的曲线可以看出：

- 1) 当检测距离 $x > 8.5\text{cm}$ 时，检测电压 $V_{out} < 1.0\text{V}$ ，说明左侧没有隔栅。
- 2) 当检测距离 $x < 3\text{cm}$ 时， $V_{out} > 2.4\text{V}$ ，说明竞赛机器人偏离中线，向左偏了。
- 3) 当检测距离 $x > 4\text{cm}$ 时， $V_{out} < 1.75\text{V}$ ，说明竞赛机器人偏离中线，向右偏了。

以 2# 传感器为例，当检测距离 $x < 2\text{cm}$ 时，检测电压 $V_{out} > 3.0\text{V}$ ，说明传感器前方有隔栅了（该停车或转弯了，否则要碰隔栅了）。

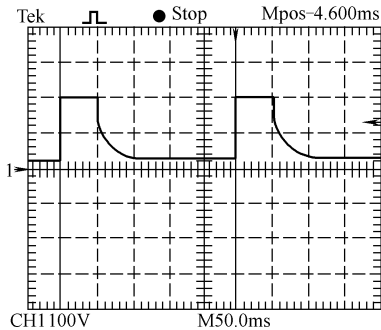


图 5-13 校基准的波形

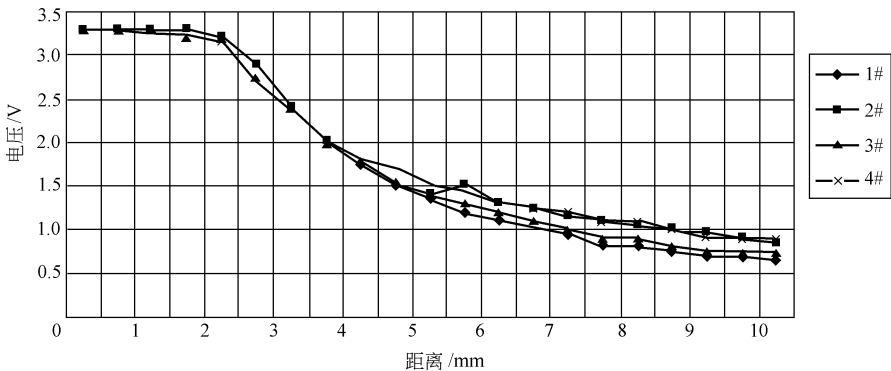


图 5-14 传感器输出电压 V_{out} 与检测距离 x 之间的关系曲线

5.3.4 沿跑道中线的运行

赛道中的竞赛智能车负责沿中线前进的方向控制是根据车模检测到的电磁感应电压来生成电动机差动控制量的。通过左右电动机速度差驱动车模转向消除车模距离道路中心的偏差。为了逐步消除车模距离道路中心线的距离差，可通过车模方向的调整及车模的前行运动来完成。该过程是一个积分过程，因此简单的比例控制就可以进行车模差动控制，并从而完成车模的方向控制。但考虑到车模自身安装的电池等比较重的物体，会产生较大的转动惯量，所以车模在调整过程中容易出现转向过冲现象，此现象如果不加以抑制，则车模在很大程度上会冲出赛道。为了能更准确反映车模重心距离电磁线的距离差别，避免角度的影响，在进行方向控制时，使用左右两个线圈感应电动势之差除以左右两个线圈感应电动势之和得一比值，使用该比值进行方向控制，可以消除检测线圈对传感器测量角度的影响。

5.4 无线发射接收模块

通过无线遥控器可以控制机器人的起动、停止、前进、左转、右转、功能设定等，竞赛机器人的遥控器如图 5-15 所示。面板上有 A、B、C、D 四位按键及一个发射指示灯。内部采用进口声表谐振器稳频，频率一致性非常好，稳定度极高，工作频率为 315MHz，频率稳定度优于 $10^{-3}\%$ ，使用中无须调整频点（如果需要也可以），使用极为方便。

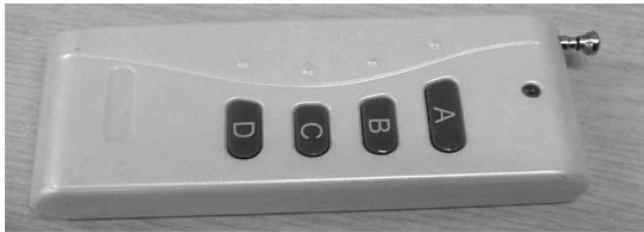


图 5-15 遥控器外形图

PT2262 编码芯片和 PT2272 解码芯片是台湾地区普城公司生产的一对 COMS 专用编码电路芯片，其引脚图如图 5-16 所示，振荡频率通过一个外接电阻进行调节。PT2262 的引脚名称见表 5-4，PT2272 的引脚名称见表 5-5，PT2262/PT2272 的特点见表 5-6。

表 5-4 PT2262 引脚名称和功能说明

名称	引脚	功 能 说 明
A0 ~ A11	1 ~ 8、10 ~ 13	用于进行地址编码，可置为“0”、“1”、“f”（悬空）
D6 ~ D11	7、8、10 ~ 13	数据输入端，有一个为“1”，即有编码发出
V_{DD}	18	电源正端（+）
V_{ss}	9	电源负端（-）
TE	14	编码启动端，用于多数据的编码发射，低电平有效
OSC1	16	振荡电阻输入端，与 OSC2 所接电阻决定振荡频率
OSC2	15	振荡电阻振荡器输出端
Dout	17	编码输出端（正常时为低电平）

表 5-5 PT2272 引脚名称

名称	管脚	说 明
A0 ~ A11	1 ~ 8、10 ~ 13	用于地址编码，可为“0”、“1”、“f”（悬空），须与 2262 一致
D0 ~ D5	7、8、10 ~ 13	地址或数据引脚
V _{DD}	18	电源正端（+）
V _{ss}	9	电源负端（-）
DATAIN	14	数据信号输入端，来自接收模块输出端
OSC1	16	振荡电阻输入端，与 OSC2 所接电阻决定振荡频率
OSC2	15	振荡电阻振荡器输出端
VT	17	解码有效确认（正常时为低电平），解码有效变成高电平（瞬态）

表 5-6 PT2262/PT2272 的特点

1	CMOS 工艺制造，低功耗	4	工作电压范围宽：2.6 ~ 15V
2	外部元器件少	5	数据最多可达 6 位
3	RC 振荡电阻	6	地址码最多可达 531441 种

编码电路 PT2262 和 PT2272 的第 1 ~ 8 引脚为地址设定脚，可以选择悬空、接正电源、接地三种状态，3 的 8 次方为 6561，所以地址编码不重复度为 6561 组。只有 PT2262 和 PT2272 的地址编码完全相同，才能配对使用。

遥控器中的编码电路如图 5-17 所示，当有按键按下时，PT2262 得电工作，其第 17 脚输出经调制的串行数据信号，第 17 脚为高电平期间，315MHz 的高频发射电路起振并发射等幅高频信号，第 17 脚为低平期间，315MHz 的高频发射电路停止振荡，所以高频发射电路完全受控于 PT2262 的第 17 脚输出的数字信号，从而对高频电路完成幅度键控（ASK）调制，相当于调制度为 100% 的调幅。

竞赛机器人主板上的无线接收电路如图 5-18 所示。其中，PT2272 是解码芯片。J3 指 DVS-9921 无线接收模块。DVS-9921 的外观如图 5-19 所示。其工作电压：4.85 ~ 5.5V；工作电流：2.3mA 左右；接收距离：120m 左右；长 × 宽 × 厚：31.3mm × 14.4mm × 8mm。PT2262/PT2272 的电气参数见表 5-7。

在具体的应用中，外接振荡电阻可根据需要进行适当的调节，阻值越大，振荡频率越低，编码的宽度越大，发码一帧的时间越长。网站上大部分产品都是用 2262Hz/1.2MHz/2272Hz/200kHz 组合的，少量产品用 2262Hz/4.7MHz/2272Hz/820kHz。

PT2262 编码芯片发出的编码信号由地址码、数据码、同步码组成一个完整的码字，PT2272 解码芯片接收到信号后，其地址码经过两次比较核对后，VT 脚才输出高电平，与此同时相应的数据脚也输出高电平。

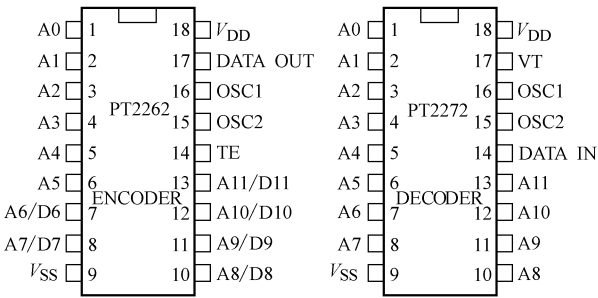


图 5-16 编码芯片 PT2262 与解码芯片 PT2272 的引脚图

一个字码由 12 位 AD 码（地址码 A 加数据码 D，如 8 位地址码加 4 位数据码）组成，地址码和数据码都用宽度不同的脉冲来表示：两个窄脉冲表示“0”；两个宽脉冲表示“1”；一个窄脉冲和一个宽脉冲表示“F”，也就是地址码的“悬空”，如图 5-20 所示（注：图 5-20 中 a 代表两个时钟振荡周期）。同步位的周期是 $128a$ ，脉宽是 $4a$ ，如图 5-21 所示。

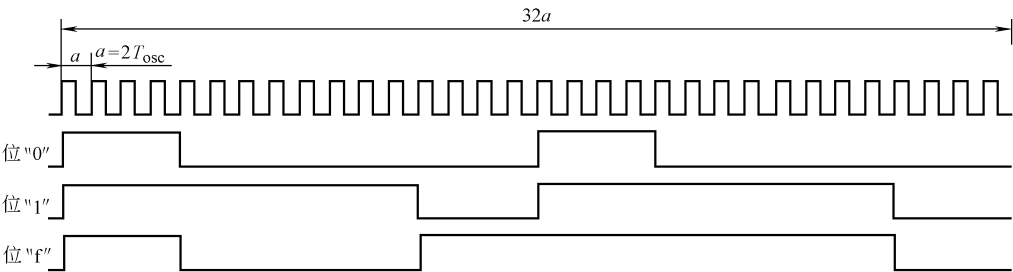


图 5-20 表示地址码和数据码的脉冲

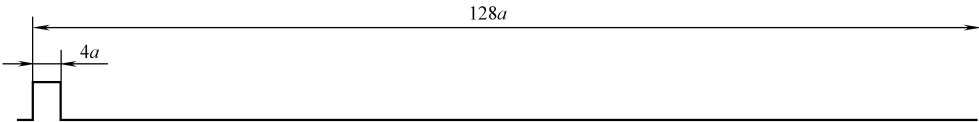


图 5-21 同步位示意图

每组字码之间由同步码隔开，用单片机软件解码时，程序只要判断出同步码，然后对后面的字码进行脉冲宽度识别即可。PT2262 每次发射时至少发射 4 组字码，PT2272 只有在连续两次检测到相同的地址码加数据码时，才会把数据码中的“1”所对应的数据输出端设置为高电平，同时把 VT 端设置为高电平。因为无线发射的特点是第一组字码非常容易受零电平干扰，往往会产生误码，所以程序可以丢弃处理。

PT2272 解码芯片的后缀有 L4/M4/L6/M6 之分，其中 L 表示锁存输出，M 表示非锁存输出，类似点动的控制。后缀的 6 和 4 表示有几路并行的控制通道：当采用 4 路并行数据（PT2272-M4）时，对应的地址编码应该是 8 位；如果采用 6 路的并行数据（PT2272-M6），对应的地址编码应该是 6 位。

5.5 数据存储模块

24C512 是串行 EEPROM 读写芯片，是通过 I^2C 总线方式通信的。 I^2C 总线是一种用于 IC 器件之间的两线制总线。两根线分别是：SDA（串行数据）线及 SCL（串行时钟）线，连接到总线上的 I^2C 器件之间可以传送信息，并根据地址识别每个 I^2C 器件，不管是单片机、存储器、LCD 驱动器还是键盘接口。

1. I^2C 总线发展

串行扩展总线技术是新一代单片机技术发展的一个显著特点。其中 PHILIPS 公司推出的 I^2C 总线（INTEL IC BUS）最为著名。与并行扩展总线相比，串行扩展总线有突出的优点：电路结构简单，程序编写方便，易于实现用户系统软硬件的模块化、标准化等。目前 I^2C 总

线大量应用在视频、音像系统中，PHILIPS 公司推出的近 200 种 I²C 总线接口器件，主要是视频、音像、单片机器件。I²C 总线最显著的特点是：规范的完整性、结构的独立性和用户使用时的“傻瓜”化。I²C 总线有严格的规范，如接口的电气特性、信号时序、信号传输的定义、总线状态设置、总线管理规则及总线状态处理等。

由于 I²C 总线系统中，各个节点的电气特性及地址给定都具有较强的独立性，因此，在应用系统中采用 I²C 总线结构就有可能实现用户梦寐以求的器件及功能单元的软、硬件标准化和模块化设计。

2. I²C 总线的基本结构

采用 I²C 总线结构的单片机或 IC 器件，其内部不仅有 I²C 接口电路，而且将内部各个单元电路功能划分为若干相对独立的模块，通过软件寻址实现片选，减少了器件片选线的连接。CPU 不仅能通过指令将各个功能单元电路挂起或摘离总线，还可对该单元的工作状况进行检测，从而实现对硬件系统的既简单又灵活的扩展与控制。I²C 总线接口电路结构如图 5-22 所示。

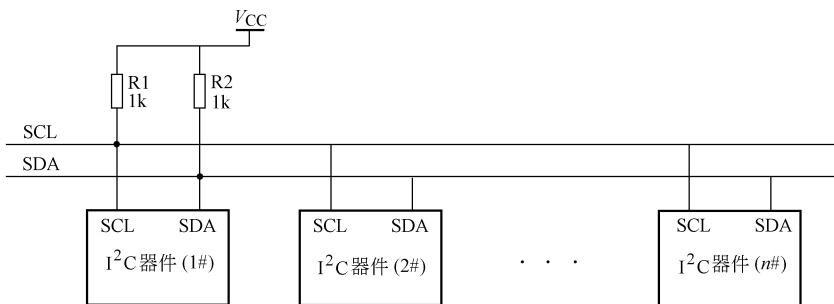


图 5-22 I²C 总线接口电路结构

在硬件结构上，任何一个具有 I²C 总线接口的外围器件，不论其功能差别有多大，都具有相同的电气接口；除了总线外，各器件节点没有其他电气连接，甚至各节点的电源都可以单独供电；在各器件节点上没有并行扩展时所必需的片选线，器件地址给定完全取决于器件类型与单元电路结构。在软件上，不论何种器件，其 I²C 总线的数据传送都具有相同的操作模式，而且每个器件操作时都与其他器件节点无关。在实际使用中，总线节点上的器件甚至可在总线工作状态下撤除或挂上总线。

3. 双向传输的接口特性

I²C 总线在传送数据过程中共有三种类型信号，它们分别是：开始信号、结束信号和应答信号。

1) 开始信号：SCL 为高电平时，SDA 由高电平向低电平跳变，开始传送数据。

2) 结束信号：SCL 为低电平时，SDA 由低电平向高电平跳变，结束传送数据。

3) 应答信号：接收数据的 IC 在接收到 8bit 数据后，向发送数据的 IC 发出特定的低电平脉冲，表示已收到数据。CPU 向受控单元发出一个信号后，等待受控单元发出一个应答信号，CPU 接收到应答信号后，根据实际情况做出是否继续传递信号的判断。若未收到应答信号，由判断为受控单元出现故障。

目前有很多半导体集成电路上都集成了 I²C 接口。带有 I²C 接口的单片机有：CYGNAL 的 C8051F0XX 系列、PHILIPSP87LPC7XX 系列、MICROCHIP 的 PIC16C6XX 系列等。很多外围器件如存储器、监控芯片等也提供 I²C 接口。

4. 数据的传送

I²C 规程运用主/从双向通信。器件发送数据到总线上，则定义为发送器，器件接收数据，则定义为接收器。主器件和从器件都可以工作于接收和发送状态。总线必须由主器件（通常为微控制器）控制，主器件产生串行时钟线（SCL），控制总线的传输方向，并产生起始和停止条件。SDA 线上的数据状态仅在 SCL 为低电平期间才能改变，SCL 为高电平期间，SDA 状态的改变被用来表示起始和停止条件。在竞赛机器人中 PIC16F877 作为 I²C 主控制器，AT24C512 作为从控制器，具体电路如图 5-23 所示。

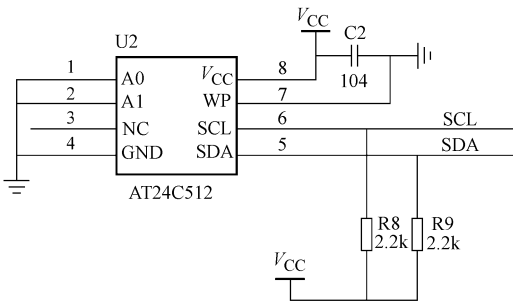


图 5-23 竞赛机器人的数据存储芯片

AT24C512 是 Atmel 公司生产的 64KB 串行电可擦的可编程存储器，内部有 512 页，每一页为 128 字节，任一单元的地址为 16 位，地址范围为 0000 ~ 0FFFFH。它采用 8 引脚封装，具有结构紧凑、存储容量大等特点，可以在 2 线总线上并接 4 片芯片，特别适用于具有大容量数据存储要求的数据采集系统，因此在测控系统中被大量采用。AT24C512 芯片的具体参数请参考其数据手册。

参 考 文 献

[1] 赵明岩. 竞赛机器人本体制作教程 [M]. 杭州：浙江大学出版社，2013.

[2] 陈万米. 竞赛机器人的创新与实践 [M]. 上海：上海大学出版社，2012.

[3] 郭洪红. 竞赛机器人设计与实践 [M]. 北京：科学出版社，2012.

[4] 黄文恺，伍冯洁，陈虹. Arduino 开发实战指南·机器人卷 [M]. 北京：机械工业出版社，2014.

[5] 赵国冬，陈东良，郭峰. 陆上机器人运动 [M]. 哈尔滨：哈尔滨工程大学出版社，2012.

[6] 杨林权. 机器人足球竞赛与设计技术 [M]. 武汉：华中科技大学出版社，2009.

第 6 章 竞赛机器人的编程技术

机器人的核心是控制系统。机器人的先进性和功能的强弱通常都直接与其控制系统的性能有关。机器人的控制是一项跨多学科的综合性技术，涉及自动控制、计算机、传感器、人工智能、电子技术和机械工程等多学科的内容。前面的章节已经介绍了机器人的结构，已经打好了一个竞赛机器人的平台，本章就以竞赛机器人走迷宫为例，来介绍竞赛机器人的软件设计与程序实现。

竞赛机器人平台的软件采用了 C 语言编程，采用模块化编程思想，易于理解和移植。竞赛机器人的软件主要由初始化模块、前进模块、左转模块、右转模块、加速模块、减速模块、停车模块、走定长模块等组成。通过改变机器人的软件，可以方便地修改机器人的各种功能和性能，实现各种算法。

6.1 竞赛机器人的控制

6.1.1 CPU 引脚资源分配

PICmicro 系列芯片典型的 I/O 引脚既能作为输入，又能作为输出，在输出模式下，这些引脚大约可以提供或者接受 20mA 的电流。根据竞赛机器人平台所要实现的功能，对单片机引脚资源进行分配，PIC16F877A 单片机引脚排列如图 6-1 所示。

16F877					
V_{pp}	1	MCLR/ V_{pp}	PGD/RB7	40	PGD
ray in1	2	RA0/AN0	PGC/RB6	39	PGC
ray in2	3	RA1/AN1	RB5	38	right
ray in3	4	RA2/AN2/VREF	RB4	37	left
ray in4	5	RA3/AN3/VREF	PGM/RB3	36	PGM
FAN	6	RA4/T0CK1	RB2	35	back
ray in5	7	RA5/AN4/SS	RB1	34	forward
			INT/RB0	33	MOTOR EN
ray in6	8	RE0/ \overline{RD} /AN5		32	V_{cc}
ray in7	9	RE1/ \overline{WR} /AN6	V_{DD}	31	GND
ray in8	10	RE2/ \overline{CS} /AN7	V_{SS}		
V_{CC}	11	V_{DD} PIC16f877DIP	RD7/PSP7	30	MOTOR 2B
GND	12	V_{SS}	RD6/PSP6	29	MOTOR 2A
X1	13	OSC1/CLKIN	RD5/PSP5	28	MOTOR 1B
X2	14	OSC2/CLKOUT	RD4/PSP4	27	MOTOR 1A
line in1	15	RC0/T1OSO/T1CK1	RC7/RX/DT	26	877RXD
line in2	16	RC1/T1OS1/CCP2	RC6/TX/CK	25	877TXD
line in3	17	RC2/CCP1	RC5/SDO	24	line in4
SCL	18	RC3/SCK/SCL	RC4/SD1/SDA	23	SDA
ray out1	19	RD0/PSP0	RD3/PSP3	22	ray out4
ray out2	20	RD1/PSP1	RD2/PSP2	21	ray out3
PIC16F877					

图 6-1 PIC16F877A 单片机的引脚排列

1. 竞赛机器人控制系统中的输入信号

- 1) 遥控器来的四个按键 A、B、C、D 信号: forward、back、left、right (开关量)。
- 2) 驱动板来的四个色差传感器信号: line_in1、line_in2、line_in3、line_in4 (开关量)。
- 3) 控制板上的四路红外传感器信号: ray_in1、ray_in2、ray_in3、ray_in4 (模拟量)。
- 4) 四路预留的火焰传感器信号: ray_in5、ray_in6、ray_in7、ray_in8 (模拟量)。

2. 输出信号

- 1) 四路控制红外发射管的信号: ray_out1、ray_out2、ray_out3、ray_out4 (开关量)。
- 2) 五个控制电动机的信号: motor_en、motor_1a、motor_1b、motor_2a、motor_2b (开关量)。

- 3) 预留的灭火风扇驱动信号: fan (开关量)。

3. I²C 总线信号

SDA (串行数据) 信号及 SCL (串行时钟) 信号。

4. 程序下载及在线调试端口

PGD、PGC、PGM 端口。

5. RS232 串口通信信号

TX、RX。

说明: 所谓开关量就是对应 0、1 两种状态, 即低电平和高电平。

6.1.2 初始化模块

该模块完成了单片机各端口定义、变量定义及赋初值、定时器初始化、中断初始化、A/D 转换初始化、串行口初始化等内容。各语句的详细说明请参看 PIC16F877 使用手册。

1. 系统初始化模块

系统初始化模块的任务是完成单片机各端口的定义及赋初值、中断初始化。该模块程序如下:

```
void initial_sys ( )
{
    TRISA = 0b00101111;    //定义 A 口 (6 位) 的 RA4 为输出端口, 其他为输入端口
    TRISB = 0XFE;          //定义 B 口 (8 位) 的 RB0 为输出端口, 其他为输入端口
    TRISD = 0X00;          //定义 D 口 (8 位) 的各端口为输出
    TRISE = 0X07;          //定义 E 口 (3 位) 的各端口为输入
    PORTD = 0X00;          //赋 D 口的初值为 0
    PORTE = 0X00;          //赋 E 口的初值为 0
    INTCON = 0X00;         //全局中断禁止
    GIE = 1;               //允许没有屏蔽的中断向 CPU 请求
    PEIE = 1;              //允许各外设中断
}
```

2. 定时器 Timer 0 初始化模块

定时器 Timer0 初始化模块的任务是完成与定时器 Timer0 相关的设置。该模块程序如下:

```
void initial_T0 ( ) {
    INTCON = 0x00;        //全局中断禁止
```

```

GIE = 1;           //允许没有屏蔽的中断向 CPU 请求
TMR0 = 0X7f;       //置定时器初值
TOIE = 1;          //Timer0 溢出中断允许
TOIF = 0;          // Timer0 溢出中断标志位清零
T0CS = 0;          //使用内部指令周期时钟
PSA = 0;           //预分频器分配给 Timer0 模块
PS0 = 1;
PS1 = 1;
PS2 = 1;           //PS0: PS2 = 111, 预分频器比选择
}

```

3. A/D 转换初始化模块

A/D 转换初始化模块的任务是把连续变化的模拟量转换为离散的数字量。该模块程序如下：

```

void adinitial ( ) {
    ADCS1 = 1;
    ADCS0 = 0; //ADCS1: ADCS0 = 00, 时钟选择位  $f_{osc}/32$ , 即  $T_{ad}$  不小于  $1.6\mu s$ 
    CHS2 = 0;
    CHS1 = 0;
    CHS0 = 0; //CHS2: CHS0 = 000, A/D 模拟通道选择位, 选择了通道 0
    ADCON1 = 0X00; //转换结果左移, ADRESL 低 6 位为 0, 8 个通道 A/D
    PIE1 = 0X70; //A/D 转换中断使能, 其他外设中断禁止
    PIE2 = 0X00; //CCP2、SSP、EEPROM 写操作中中断禁止
}

```

6.1.3 运动模块

运动模块又包括如下子模块：

1. 前进模块

前进模块的任务是控制竞赛机器人沿着直线前进。从竞赛机器人的原理图 5-5 可见，单片机驱动直流电动机的信号为分别为 RB0、RD4、RD5、RD6、RD7。前进模块由 motor_forward () 函数完成。

```

void motor_forward ( )
{
    EN_MOTOR = 1; //使能 L298
    RD4 = 0;       //驱动左轮正转（向前转）
    RD5 = 1;
    RD6 = 1;       //驱动右轮正转（向前转）
    RD7 = 0;
}

```

2. 原地右转模块

右转模块的任务是控制竞赛机器人通过左轮正转或右轮反转来实现原地向右转弯。从竞

赛机器人的原理图（见图 5-5）可见，单片机驱动直流电动机的信号分别为 RB0、RD4、RD5、RD6、RD7。右转模块由 motor_right（）函数完成。

```
void motor_right ( )
{
    EN_MOTOR = 1;           //使能 L298
    RD4 = 0;                 //驱动左轮正转（向前转）
    RD5 = 1;
    RD6 = 0;                 //驱动右轮反转（向后转）
    RD7 = 1;
}
```

3. 原地左转模块

左转模块的任务是控制竞赛机器人通过右轮正转或左轮反转来实现原地向左转弯。从竞赛机器人的原理图（见图 5-5）可见，单片机驱动直流电动机的信号分别为 RB0、RD4、RD5、RD6、RD7。左转模块由 motor_left（）函数完成。

```
void motor_left ( )
{
    EN_MOTOR = 1;           //使能 L298
    RD4 = 1;                 //驱动左轮反转（向后转）
    RD5 = 0;
    RD6 = 1;                 //驱动右轮正转（向前转）
    RD7 = 0;
}
```

4. 左转 90°模块

左转 90°模块的功能是控制竞赛机器人按逆时针方向原地左转 90°。从驱动板结构图（见图 4-2）可以看出，左轮反转（向后转），同时右轮正转（向前转）一定的时间 turn_time_set，就可以实现竞赛机器人左转 90°。turn_time_set = 30，大约为 0.2s。左转 90°模块由函数 void motor_90left（）完成。

```
void motor_90left ( )
{
    time = 0;
    EN_MOTOR = 1;           //使能 L298
    RD4 = 1;                 //驱动左轮反转（向后转）
    RD5 = 0;
    RD6 = 1;                 //驱动右轮正转（向前转）
    RD7 = 0;
    while (time != turn_time_set); //等待
    motor_stop ( );
    time = 0;                //时间清零
    stime = 0;
}
```

5. 右转 90°模块

右转 90°模块的功能是控制竞赛机器人按逆时针方向原地右转 90°。从驱动板结构图

(见图 4-2) 可以看出, 左轮正转 (向前转), 同时右轮反转 (向后转) 一定的时间 `turn_time_set`, 就可以实现竞赛机器人右转 90° 。`turn_time_set = 30`, 大约为 0.2s。右转 90° 模块由函数 `void motor_90right ()` 完成。

6. 行走模块

在竞赛机器人沿跑道中线前进时, 需要不断根据传感器信号调整行走方向, 使得竞赛机器人不会撞到墙上, 这时的左转右转只是在不断地调整, 左转功能模块具体由左轮停止转动、右轮继续向前运动来实现, 即通过 (`RD5 = 0; RD6 = 1;`) 完成, 同理右转模块具体由函数 (`RD5 = 1; RD6 = 0;`) 完成。在前进过程中, 会遇到四种情况:

- 1) 左侧有墙壁, 右侧无墙壁;
- 2) 左侧无墙壁, 右侧有墙壁;
- 3) 左侧和右侧都有墙壁;
- 4) 左侧和右侧都无墙壁。

因此需要根据不同情况来设计行走过程中的方向调整方案。当只有一侧有墙壁时, 需要根据这一侧墙壁的传感器反馈信号进行纠偏; 当两侧都有墙壁时, 根据两侧的传感器纠偏; 当两侧都没有墙壁时, 继续前进。沿左侧墙壁纠偏由函数 `alongleft ()` 具体实现, 沿右侧墙壁纠偏由函数 `alongright ()` 具体实现, 沿两侧墙壁纠偏由函数 `alongbothsideside ()` 具体实现。当两侧传感器探测距离大于 7cm 时, 认为该侧没有墙壁。

//右侧无墙壁, 依据左侧传感器沿着左侧墙壁行走

```
void alongleft( ) {
    if( ( adresult_an[3] <= y5cm_left) && ( adresult_an[3] >= y7cm_left) ) { RD5 = 0; RD6 = 1; }
```

//左侧传感器与墙壁的距离大于 5cm 小于 7cm 时, 向左转

```
else if( adresult_an[3] >= y2cm_left) { RD5 = 1; RD6 = 0; }
```

//左侧传感器与墙壁的距离小于 2cm 时, 向右转

```
else { RD5 = 1; RD6 = 1; }
```

//继续前进

```
}
```

//左侧无墙壁, 依据右侧传感器沿着右侧墙壁行走

```
void alongright( ) {
```

```
if( ( adresult_an[0] <= y5cm_right) && ( adresult_an[0] >= y7cm_right) ) { RD5 = 1; RD6 = 0; }
```

//右侧传感器与墙壁的距离大于 5cm 小于 7cm 时, 向右转

```
else if( adresult_an[0] >= y2cm_right) { RD5 = 0; RD6 = 1; }
```

//右侧传感器与墙壁的距离小于 2cm 时, 向左转

```
else { RD5 = 1; RD6 = 1; }
```

//继续前进

```
}
```

//两侧都有墙壁,沿两侧墙中间走

```
void alongbothsideside() {  
    if(adresult_an[0] >= y3cm_right) {RD5 = 0;RD6 = 1;} //右侧距离小于 3cm,向左转  
    else if(adresult_an[3] >= y3cm_left) {RD5 = 1;RD6 = 0;} //左侧距离小于 3cm,向右转  
    else {RD5 = 1; RD6 = 1;} //继续前进  
}
```

在函数中左轮控制电压为 PWM 波,波形如图 6-2 所示。可见加在左轮电动机的平均电压减小,从而减小了其速度,而右轮的驱动电压仍保持不变,右轮全速前进,从而实现了左转。

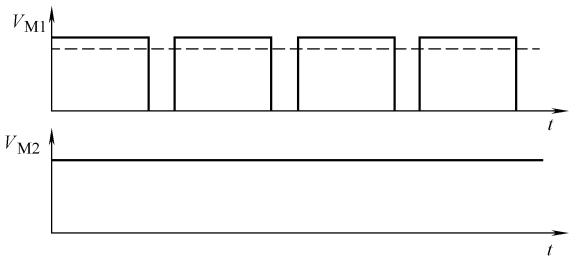


图 6-2 PWM 的波形

7. 停车模块

停车模块的功能是控制竞赛机器人准确停车,这里采用了能耗制动,将左轮和右轮驱动电压置零,则由反电动势产生的制动电流由放电回路释放,如图 6-3 所示,从而使竞赛机器人尽快停止运动,具体由函数 motor_stop () 完成。

```
motor_stop ( )  
{  
    EN_MOTOR = 0; //禁止 L298  
    RD4 = 0; //左轮电动机电压为零  
    RD5 = 0;  
    RD6 = 0; //右轮电动机电压为零  
    RD7 = 0;  
}
```

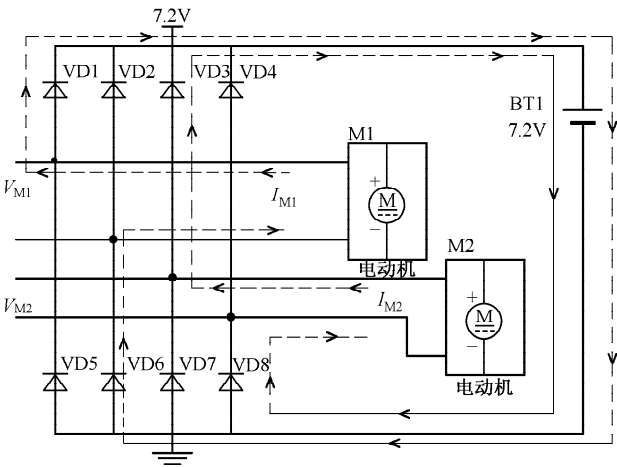


图 6-3 驱动电动机的能耗制动的放电回路

8. 走定长模块

竞赛机器人经常需要走固定长度的距离,如半个跑道宽度的距离,具体由 motor_halfgrid () 完成。竞赛机器人前进一定的时间 halfgrid_time_set 就停车。

```

void motor_halfgrid( )
{
    halfgrid_time = 0;
    EN_MOTOR = 1;           //使能 L298
    RD4 = 0;
    RD5 = jiupian_pwm;
    RD6 = 1;
    RD7 = 0;
    while( halfgrid_time! = halfgrid_time_set );
    motor_stop( );
    flag = STOP;
}

```

6.1.4 A-D 转换模块

A-D 转换模块由函数 ad() 完成。A-D 转换的程序框图如图 6-4 所示。

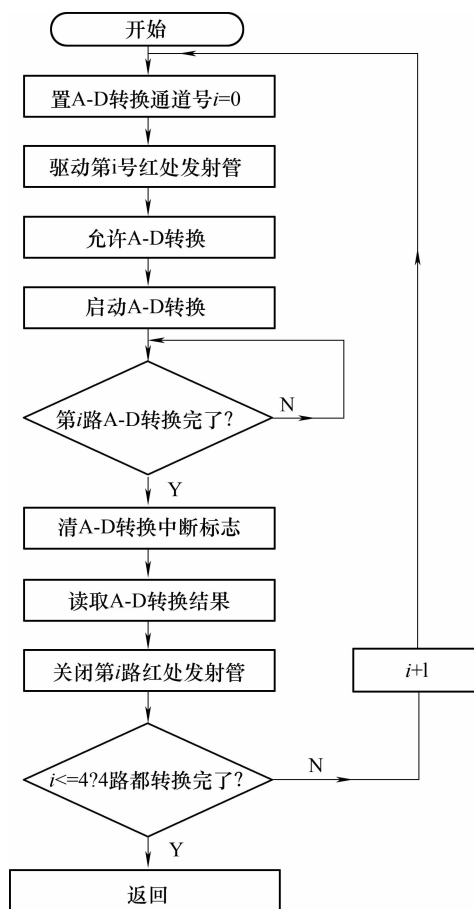


图 6-4 A-D 转换的程序框图

```

void ad4( )
{ unsigned char i;

```

```

for(i=0; i<4; i++) {
    switch(i)          //置 A-D 转换通道号, 并驱动红外发射管
    { case0: { CHS1=0; CHS0=0; RAY_OUT1 = 1;}; break;
      case1: { CHS1=0; CHS0=1; RAY_OUT2 = 1;}; break;
      case2: { CHS1=1; CHS0=0; RAY_OUT3 = 1;}; break;
      case3: { CHS1=1; CHS0=1; RAY_OUT4 = 1;}; break;
    }
    delay(1000);
    ADON=1;              //允许 A-D 转换
    ADGO=1;              //启动 A-D 转换
    while(! ADIF);       //等待转换完毕
    ADIF=0;              //清除中断标志
    adresult_an[i].adre[0]=ADRESL; //读取并存储 A-D 转换结果
    adresult_an[i].adre[1]=ADRESH;
    delay(250);
    switch(i)            //关闭红外发射管
    { case 0: RAY_OUT1 = 0; break;
      case 1: RAY_OUT2 = 0; break;
      case 2: RAY_OUT3 = 0; break;
      case 3: RAY_OUT4 = 0; break;
    }
}
}

```

6.2 迷宫智能算法的实现

迷宫智能算法模块的主要功能是:

- 1) 优化设置模块 (优化设置竞赛机器人最佳加减速度, 控制策略);
- 2) 解决迷宫问题算法的实现;
- 3) 经历路径的记忆;
- 4) 竞赛机器人速度的控制 (加速、减速的平滑, 转弯等);
- 5) 迷宫策略的优化。

6.2.1 沿跑道中线前进的判断程序

由于迷宫的跑道较窄, 曲折变化, 机器人经常要转 90° 、 180° 等, 这就需要竞赛机器人沿着跑道中线前进, 否则就会撞到迷宫。

迷宫场地中每个小格的标准长度是 180mm, 为了可以方便地搬运、组装迷宫场地, 这里采用了标准的木地板组合, 木地板的标准宽度是 192mm, 迷宫的隔栅宽度是 20mm, 故迷宫的跑道宽度是 172mm, 与木地板标准宽度几乎一样。竞赛机器人的宽度是 100mm, 故竞

赛机器人在跑道中线时，左右边距离各是 36mm。

从图 5-14 所示的传感器输出电压 V_{out} 与检测距离 x 之间的关系曲线中可以看出，当检测距离 $x < 3\text{cm}$ 时，检测电压 $V_{out} > 2.4\text{V}$ ，说明竞赛机器人偏离中线，向左偏了。当检测距离 $x > 4\text{cm}$ 时， $V_{out} < 1.75\text{V}$ ，说明竞赛机器人偏离中线，向右偏了。根据曲线的对应关系，可以实现竞赛机器人沿着跑道中线前进。

首先实现溜边走的程序。竞赛机器人溜边前进的示意图如图 6-5 所示。当竞赛机器人左侧的传感器测得距离 $L_1 < 3\text{cm}$ 时，竞赛机器人右拐，当左侧传感器测得左侧距离 $L_2 > 4\text{cm}$ 时，竞赛机器人左拐。如图 6-5 所示，路径 S 是竞赛机器人的实际前进路线，整体地看，相当于竞赛机器人沿着跑道中线 M 前进。溜边前进的程序框图如图 6-6 所示。溜边走的功能由 liubian() 函数完成。

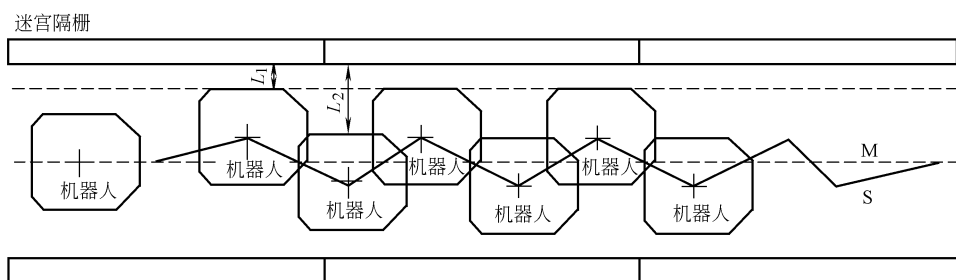


图 6-5 竞赛机器人溜边前进的示意图

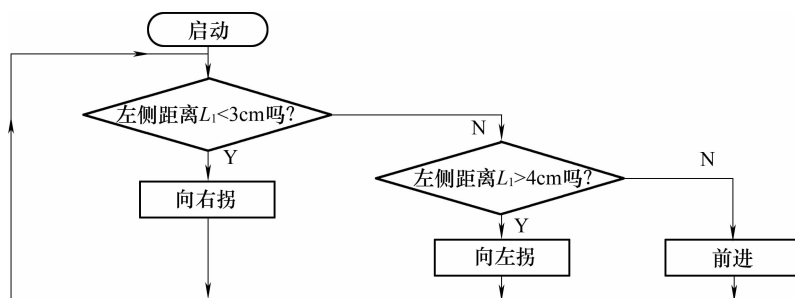


图 6-6 溜边前进的程序框图

```

liubian( ) {
    if( adresult_an[0].y >= y24) turntright( );    //小于 3cm, 右拐
    else if( adresult_an[0].y <= y17) turntleft( ); //大于 4cm, 左拐
    else turntoforward( );                        //前进
    if( adresult_an[1].y >= y24) motor_stop( ); //前方距离小于 3cm(有隔栅), 停
}
  
```

6.2.2 无记忆功能迷宫算法的编程实现

无记忆功能的迷宫算法的程序框图如图 6-7 所示，在迷宫中自由行走的程序由函数 pandeduan() 完成。

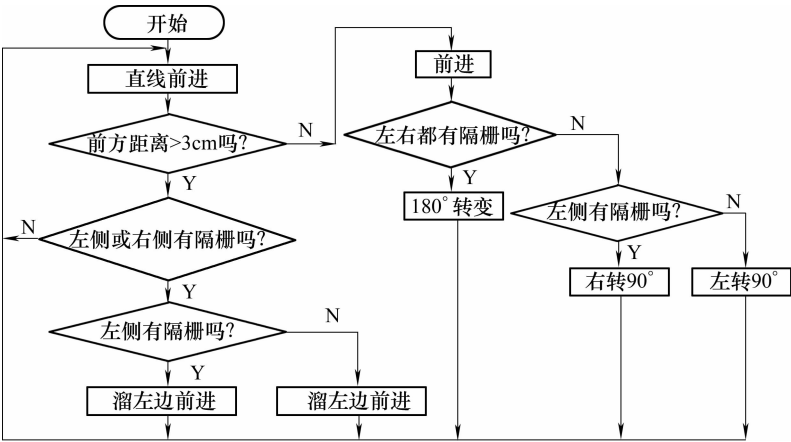


图 6-7 无记忆功能的迷宫算法的程序框图

//直走动作

```
void motor_along( ) {
    EN_MOTOR = 1;           //使能 L298
    RD4 = 0;
    RD7 = 0;
    if( ( adresult_an[0] >= y7cm_right) &&( adresult_an[3] >= y7cm_left) )
    {
        alongbothside( );    //两侧都有墙壁，双边纠偏
    }
    else if( ( adresult_an[0] <= y7cm_right) &&( adresult_an[3] >= y7cm_left) )
    {
        alongleft( );        //只有左侧有墙壁，左单边纠偏
    }
    else if( ( adresult_an[0] >= y7cm_right) &&( adresult_an[3] <= y7cm_left) )
    {
        alongright( );       //只有右侧有墙壁，右单边纠偏
    }
    else { RD5 = 1; RD6 = 1; } //继续前进
}
```

6.2.3 有记忆功能的迷宫算法分析

竞赛机器人有了记忆功能，才能表现出智能。

1. 迷宫的状况

设在迷宫中的竞赛机器人如图 6-8 所示，竞赛机器人具有三个方向的传感器：前方传感器 F、左方传感器 L、右方传感器 R。如果前方有隔栅，则传感器来的信号为“1”，无隔栅则为“0”。竞赛机器人在迷宫中运动时，三个传感器 F、L、R

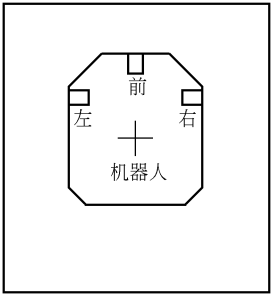


图 6-8 在迷宫中的竞赛机器人

会传来迷宫的 8 种状况，即 111、110、101、100、011、010、001、000，如图 6-9 所示。即死胡同 111、右转弯 110、左转弯 101、直丁字路口 100、直路 011、右丁字路口 010、左丁字路口 001、十字路口 000。

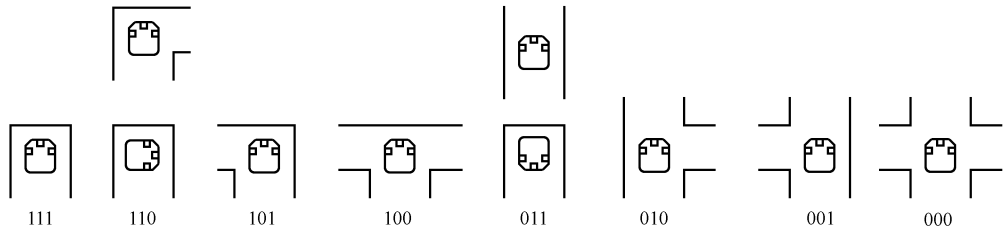


图 6-9 迷宫的 8 种状况

2. 记忆迷宫的状态寄存器

设可位操作的路况寄存器 `road_flag` 用来记忆迷宫的路况，每遇到变化的分叉的路况，置 `road_flag.0 = 1` 或 `0`，且左移一位，把路况保存起来。“1”表示可前进，“0”表示可右转（对于十字路口，右转优先），如图 6-10 所示。

设死胡同寄存器 `dead_flag` 用来记忆遇到的死胡同。每遇到一个死胡同，置 `dead_flag.0 = 1`，且左移一位。

设转向寄存器 `turn_flag` 用来记忆竞赛机器人的向左转或向右转的操作状态，如图 6-11 所示。

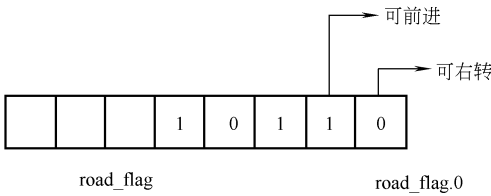


图 6-10 可位操作的路况寄存器 `road_flag`

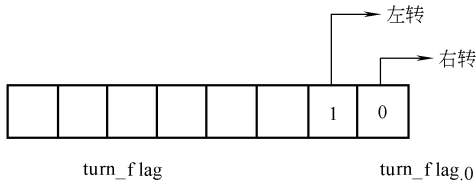


图 6-11 转向寄存器 `turn_flag`

3. 迷宫问题的分析

一般的迷宫有上面的 8 种状况，当竞赛机器人遇到这 8 种状况时，做出正确的选择才能走出迷宫。下面具体来研究竞赛机器人遇到这 8 种状况时应采取的措施。

(1) 死胡同 111 状况 当竞赛机器人遇到死胡同 111 状况时，应采取的处理方法是：原地 180°调头，同时置死胡同标志，如图 6-12 所示，由 `state_111()` 函数实现。

(2) 右转弯 110 状况 当竞赛机器人遇到右转弯 110 状况时，应采取的处理方法是：原地右转 90°前进，如图 6-13 所示，由 `state_110()` 函数实现。

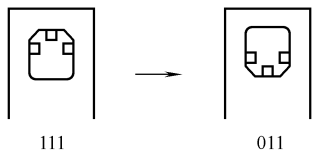


图 6-12 死胡同 111 状况

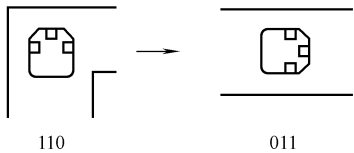


图 6-13 右转弯 110 状况

(3) 左转弯 101 状况 当竞赛机器人遇到左转弯 101 状况时,应采取的处理方法是:原地左转 90°前进,如图 6-14 所示,由 state_101()函数实现。

(4) 直路 011 状态 当竞赛机器人处于直路 011 状态时,应采取的处理方法是:继续保持前进,由 state_011()函数实现。

(5) 十字路口 000 状态 当竞赛机器人遇到十字路口 000 状态时,应采取的处理方法是:向右转 90°前进,如图 6-15 所示,由 state_000()函数实现。

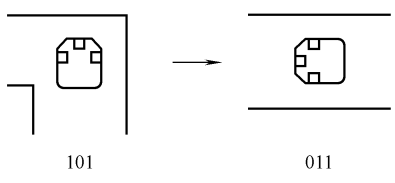


图 6-14 左转弯 101 状况

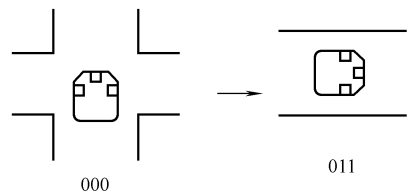


图 6-15 十字路口 000 状况

(6) 直丁字路口 100 状态 当竞赛机器人遇到直丁字路口 100 状态时,如图 6-16 所示。应采取的处理方法的程序框图如图 6-17 所示,由 state_100()函数实现。

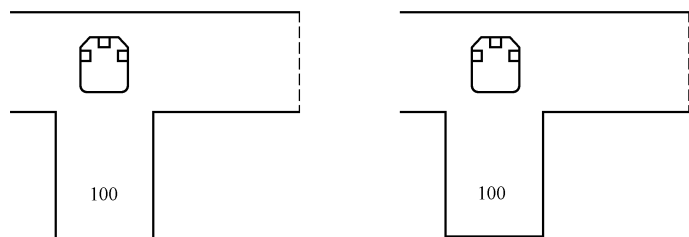


图 6-16 遇到直丁字路口 100 状况

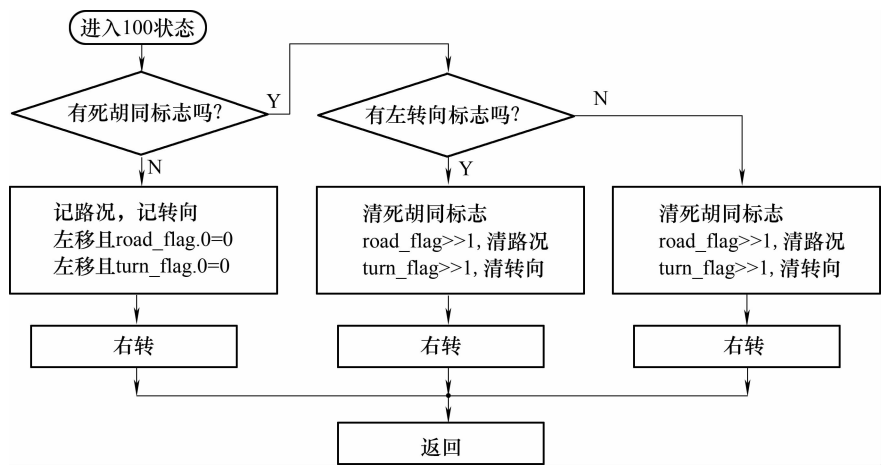


图 6-17 遇到直丁字路口 100 状况时的程序框图

(7) 右丁字路口 010 状态 当竞赛机器人遇到右丁字路口 010 状态时,如图 6-18 所示。应采取的处理方法的程序框图如图 6-19 所示,由 state_010()函数实现。

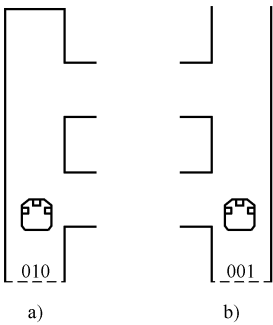


图 6-18 丁字路口状况

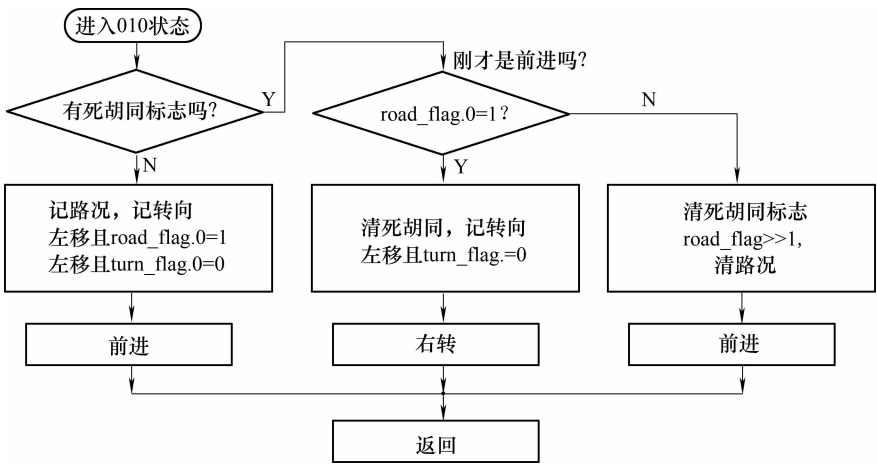


图 6-19 遇到右丁字路口 010 状况时的程序框图

(8) 左丁字路口 001 状态 当竞赛机器人遇到左丁字路口 001 状态时，如图 6-18b 所示。应采取的处理方法的程序框图如图 6-20 所示，由 state_001() 函数实现。

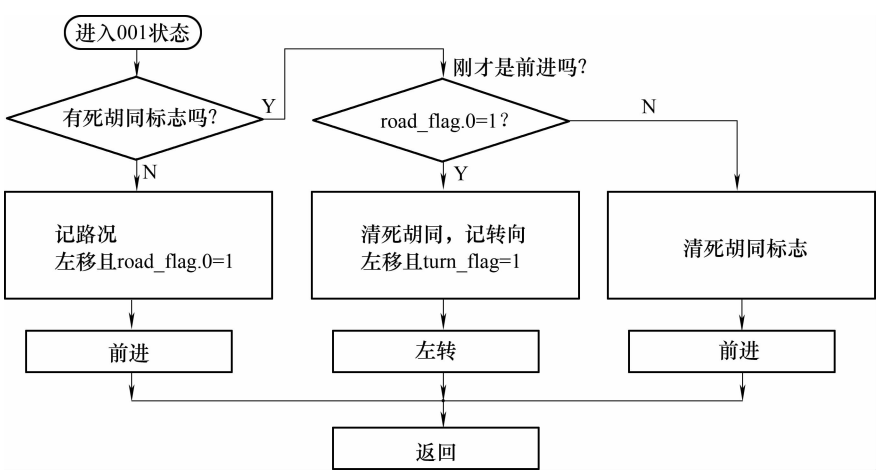


图 6-20 遇到左丁字路口 001 状况时的程序框图

4. 迷宫问题的解决

竞赛机器人在遇到每种分叉口状况时，记忆路况，转向。当遇到死胡同时返回，当返回时遇到刚才的分叉口时，根据上次的转向，决定当前的运动，或前进或左转或右转。其程序框图如图 6-21 所示。

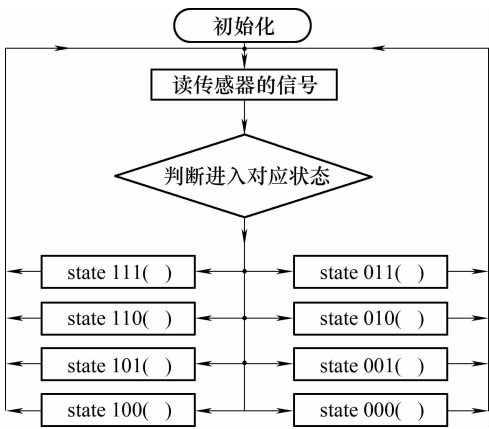


图 6-21 具有记忆功能的迷宫程序框图

6.3 上位机软件

上位计算机一般是指以个人计算机为硬件的基于高级语言进行软件开发的计算机系统。它主要完成复杂功能的软件设计与程序实现。上位机主要目的是为操作者提供一个友好的控制界面，方便操作者的控制。下位计算机一般是指单片计算机，主要完成实时性要求较高的控制与计算功能。下位机通过各种通信手段来与上位机通信，通信得到的数据，作控制还是作数据用，这是依据程序的设计而定的，下位机主要处理一些直接的控制任务，如电动机的控制等，而上位机则主要从宏观上考虑控制的策略(比赛的策略)，分析场地情况。由于有了这样的结构，即骨架，我们就可以在这样的骨架上添加肌肉，让我们的机器人控制系统更加健壮。经过努力，我们初步地实现了这个控制结构，从设计到实施，都表现出了这种竞赛机器人的控制结构的优越性。控制结构的好坏直接影响到控制的效果和控制的目的。

这里，上位机主要功能是：利用上位机软件界面，设置与实际迷宫一致的模拟迷宫，人为给出具体路径，经无线下载到竞赛机器人中，竞赛机器人就按照上位机的设置路径精确运行。上位机软件界面如图 6-22 所示。上位机主要用来完成如下功能：

- 1) 解决迷宫问题算法的实现；
- 2) 经历路径的记忆；
- 3) 竞赛机器人速度的控制(加速、减速的平滑，转弯等)；
- 4) 迷宫策略的优化。

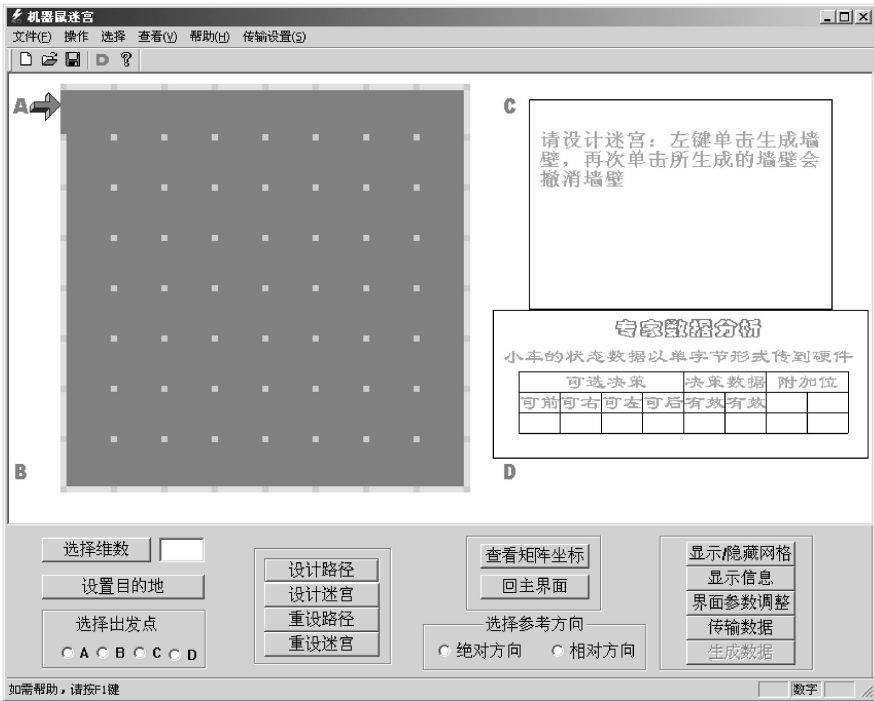


图 6-22 上位机软件界面

参 考 文 献

[1] 谭浩强. C 程序设计[M]. 北京: 清华大学出版社, 1988.

[2] 苏素珍, 土屋喜一. 使用遗传算法的迷宫学习[J]. 机器人, 1994, 16(5): 286-289.

[3] 陈慧南. 算法设计与分析——C++语言描述[M]. 北京: 电子工业出版社, 2006.

[4] 克莱因伯格. 算法设计[M]. 北京: 清华大学出版社, 2006.

[5] 范策, 周世平, 胡潇琨. 算法与数据结构——C语言版[M]. 北京: 机械工业出版社, 2004.

[6] 胡建平. C语言程序设计[M]. 天津: 天津大学出版社, 2005.

第 7 章 竞赛机器人的运动控制算法

7.1 PID 控制算法

7.1.1 PID 控制简介

在工程实际中，应用最为广泛的调节器控制规则就是比例（Proportion）、积分（Integral）、微分（Derivative）控制，简称 PID 控制。PID 控制器问世至今已有 70 余年，它以结构简单、稳定性好、工作可靠、调整方便而成为工业控制主要技术之一。PID 控制技术主要应用于系统控制器结构和参数不能通过理论计算，必须依靠经验和现场调试来确定的情况。即当不完全了解一个系统的控制过程和被控对象的有效参数时，适用 PID 控制技术。PID 控制实际中也分 PI 控制和 PD 控制。PID 控制器就是利用比例、积分、微分三种控制来尽量减小被控对象产生的系统误，以及加强系统稳定性与抗干扰能力。图 7-1 所示为 PID 控制系统原理图。

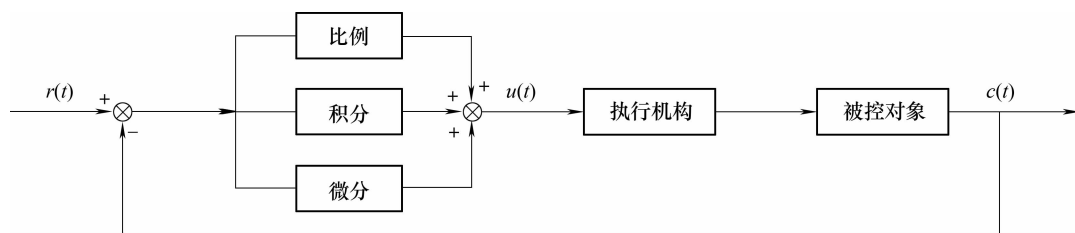


图 7-1 PID 控制系统原理图

下面将各种常用控制规律的控制特点进行简单的总结。

(1) 比例控制规律（P）。采用比例控制规律能较快地克服扰动的影响，比例控制的主要作用是减小偏差，比例作用越大，调节速度越快。但比例调节不能很好地稳定在一个理想的数值，会有余差出现。因此比例控制仅适用于控制要求不高、负荷变化不大、控制滞后对被控对象影响不大，或者被控参数允许在一定范围内有余差的场合。

(2) 比例积分控制规律（PI）。在工程中，比例积分控制规律是应用最广泛的一种控制规律。积分项是对控制时间的积分，随着时间的增加，积分项也会随之增大，从而推动控制器的输出增大，使稳态误差进一步减小，直到等于零。积分项的引入可以帮助比例控制消除系统余差。

(3) 比例微分控制规律（PD）。微分调节具有预见性，系统可以预见偏差的变化趋势并产生超前控制作用，在偏差形成之前将其消除。对于具有滞后的控制通道，引入适当的微分调节参与控制，可以有效减少系统超调量与调节时间，对于提高系统的动态性能指标有着显著效果。因此，对于控制通道的时间常数或时间滞后较大的场合，为了提高系统的稳定性，

减小动态偏差等可选用比例微分控制规律。与此同时，微分作用也会放大外界的干扰，使得系统抗干扰能力下降，故不能施加过大的微分作用。

(4) 例积分微分控制规律 (PID)。PID 控制规律是一种较理想的控制规律，它在比例控制的基础上引入积分，可以消除余差，再加入微分作用，又能提高系统的稳定性。它适用于控制通道时间常数或时间滞后较大、控制要求较高的场合。

竞赛机器人的结构与各个模块的工作参数往往会受到场地、灯光、环境等因素的影响，导致控制系统无法构造精准的数学模型，所以机器人的控制参数及控制器结构通常需要开发人员在现场依靠经验进行调试，因此 PID 控制技术的灵活应用成为机器人竞赛选手制胜的关键。

7.1.2 PID 控制算法分类

1. 模拟 PID 调节器

PID 调节器是一种线性调节器，它将给定值 $r(t)$ 与实际输出值 $c(t)$ 偏差的比例(P)、积分(I)、微分(D)通过线性组合构成控制量，对被控对象进行控制。PID 调节器的微分方程如下：

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \tag{7-1}$$

$$e(t) = r(t) - c(t) \tag{7-2}$$

PID 调节器的传递函数对应如下：

$$G(s) = \frac{U(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} + T_d s \right] \tag{7-3}$$

模拟 PID 控制系统框图如图 7-2 所示。

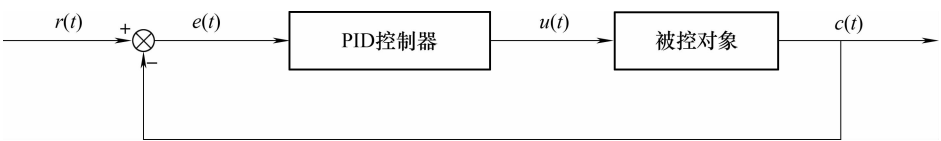


图 7-2 模拟 PID 控制系统框图

2. 数字 PID 调节器

为了便于计算机实现 PID 控制算式，必须把连续函数微分方程改写成为离散的差分方程，模拟 PID 控制规律的离散化见表 7-1。

表 7-1 模拟 PID 算子离散化对照表

模拟形式	离散化形式
$e(t) = r(t) - c(t)$	$e(n) = r(n) - c(n)$
$\frac{de(t)}{dt}$	$\frac{e(n) - e(n-1)}{T}$
$\int_0^t e(t) dt$	$\sum_{i=0}^n e(i) T = T \sum_{i=0}^n e(i)$

根据式(7-1)结合表 7-1 可以得出差分方程。

$$u(n) = K_p \{ e(n) + \frac{T}{T_i} \sum_{j=0}^n e(j) + \frac{T_D}{T} [e(n) - e(n-1)] \} \quad (7-4)$$

式中, $u(n)$ 是第 n 时刻的系统离散控制量, 如果控制周期 T 比被控对象时间常数 K_p 小, 则该离散控制与连续控制十分接近。由于在模拟 PID 调节器控制过程中很难实现理想微分 $\frac{de(t)}{dt}$, 所以通过计算机对差分方程式(7-4)进行近似计算, 得到控制参数的方法称为理想微分 PID 数字控制器。

在计算机控制系统中, 数字 PID 控制算法又分为位置式 PID 和增量式 PID。

(1) 位置型 PID。模拟 PID 调节器的调节动作是连续的, 任一瞬间的输出控制量 u 都对应一个被控对象的位置状态, 由式(7-4)可知数字 PID 调节器每一个采样时刻的输出控制量 $u(n)$ 也与被控对象的位置对应, 故称式(7-4)为位置型 PID 算式(位置式 PID)。由于计算机在实现位置型算式的过程中需要不断累加偏差 $e(j)$, 不仅占用了大量存储单元, 而且编程复杂, 因此需要对原算式进行改进。

(2) 增量型 PID。根据式(7-4)不难写出第 $(n-1)$ 时刻的控制量 $u(n-1)$, 即

$$u(n-1) = K_p \{ e(n-1) + \frac{T}{T_i} \sum_{j=0}^{n-1} e(j) + \frac{T_D}{T} [e(n-1) - e(n-2)] \} \quad (7-5)$$

式(7-4)减去式(7-5)后得到 n 时刻控制量的增量 $\Delta u(n)$

$$\Delta u(n) = K_p \{ e(n) - e(n-1) + \frac{T}{T_i} e(n) + \frac{T_D}{T} [e(n) - 2e(n-1) + e(n-2)] \} \quad (7-6)$$

由于式(7-6)中的 $\Delta u(n)$ 对应于第 n 时刻被控对象位置的增量, 故称此式为增量算式。因此, n 时刻的实际控制量为

$$u(n) = u(n-1) + \Delta u(n) \quad (7-7)$$

式中, $u(n-1)$ 为第 $(n-1)$ 时刻的控制量。

综上所述, 计算 $\Delta u(n)$ 和 $u(n)$ 要用到第 $(n-1)$, $(n-2)$ 时刻的历史数据 $e(n-1)$, $e(n-2)$ 和 $u(n-1)$, 这 3 个历史数据也已在前时刻存于内存储器。通常采用平移法保存这些历史数据。比如, 计算完 $u(n)$ 后, 首先将 $e(n-1)$ 存入 $e(n-2)$ 单元, 然后将 $e(n)$ 存入 $e(n-1)$ 单元, 以及把 $u(n)$ 存入 $u(n-1)$ 单元, 如图 7-3 所示。这样就为下时刻计算机做好准备。

由此可见, 采用增量型算式当存在计算误差或精度不足时, 对控制量计算的影响很小; 另外, 增量式易于实现 PID 调节器手动到自动的无冲击切换。位置型 PID 与增量型 PID 控制算法的流程如图 7-4 所示。

3. 不完全微分 PID

将微分环节引入智能车的方向和速度控制, 能明显改善系统的动态性能, 但同时也会使系统对于外界干扰及信号突变特别敏感, 影响系统的稳定性。为了克服上述缺点, 本文在 PID 算法中加入了一阶惯性环节。不完全微分 PID 算法结构如图 7-5 所示。

由于普通数字 PID 调节器的微分作用只在第一个周期内起作用, 而不能按照偏差变化的趋势在整个调节过程中起作用, 因此第一个周期内的微分作用很强, 对于时间常数较大的系

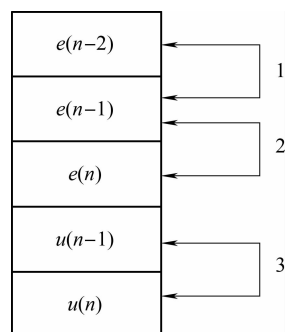


图 7-3 增量式保存数据方法

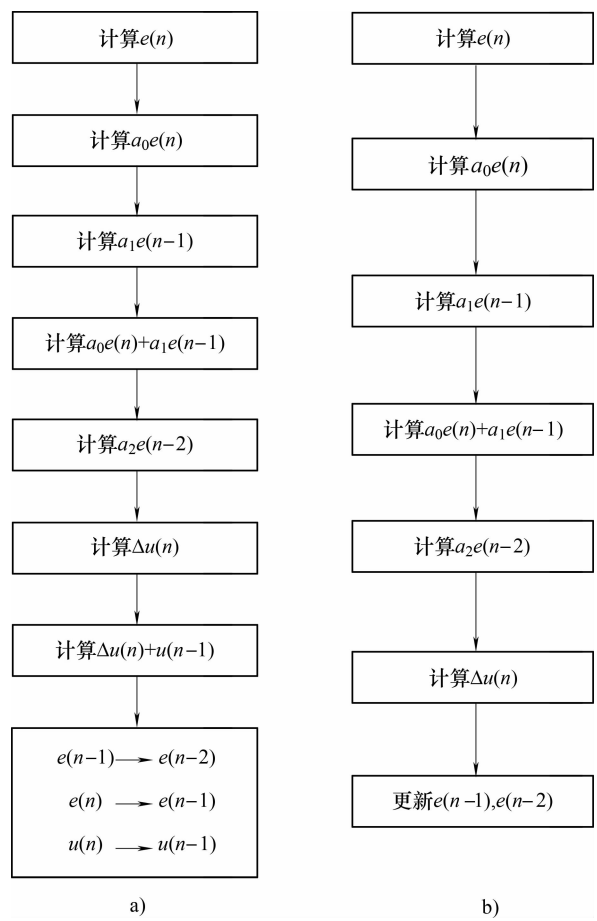


图 7-4 位置型 PID 与增量型 PID 控制算法的程序流程图
a) 位置型 PID b) 增量型 PID

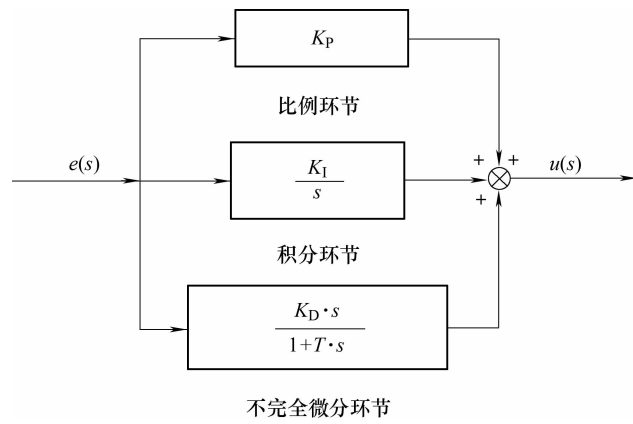


图 7-5 不完全微分 PID 算法结构图

统，其调节作用很小，不能达到超前控制误差的目的，容易产生溢出。为了克服上述缺点，通常情况下，为了避免纯微分运算的出现，可以在微分项上加一个一阶惯性环节 $\frac{1}{1 + T \cdot s}$ ，

此时可以得到系统的传递函数为

$$U(s) = \left(K_P + \frac{K_I}{s} + \frac{K_D \cdot s}{1 + T \cdot s} \right)$$

(7-8)

由拉普拉斯变换的微分定理可将频域部分方程转化为时域微分方程并离散化,可求得算式为

$$u_D(k) = K_D \cdot (1 - \alpha) \cdot [e(k) - e(k-1)]$$

$$+ \alpha \cdot u_D(k-1)$$

(7-9)

式中, α 为系统时间常数与一阶惯性环节时间常数决定的一个常数。分析式(7-9)可知,引入不完全微分后,微分输出在第一个采样周期有所减小,此后又按照一定比例衰减,由此可见,不完全微分数字 PID 调节器不但能抑制高频干扰,而且克服了普通数字 PID 调节器的缺点。不完全微分数字 PID 调节器输出的微分作用能在各个周期内按照偏差的变化趋势均匀地输出,真正起到了微分的作用,改善了系统的性能。由于在第一个采样周期内不完全微分数字调节器的输出比完全微分数字调节器的输出幅度小得多,有效克服了智能机器人在复杂环境下受到的偏差干扰给速度控制带来的不良影响,因此不完全微分数字 PID 调节器具有比较理想的调节性能。图 7-6 所示为不完全微分 PID 算法的程序流程图。

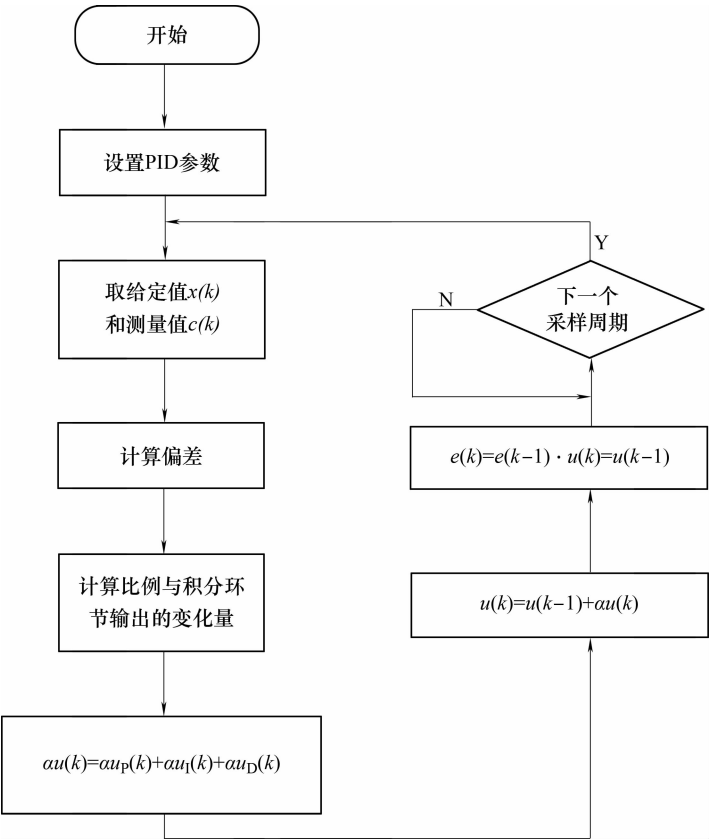


图 7-6 不完全微分 PID 算法的程序流程图

4. 微分先行 PID

微分先行是把对偏差的微分改为对被控量的微分,这样在给定值变化时输出不会产生大

幅度变化。而且由于被控量一般不会突变，因此即使给定值已发生改变，被控量也是缓慢变化的，从而不致引起微分项的突变。微分项的输出增量为

$$\Delta u_D(n) = \frac{K_p T_D}{T} [\Delta c(n) - \Delta c(n-1)] \quad (7-10)$$

由于竞赛的智能车在场地行驶过程中经常会遇到变速、转弯、爬坡等情况，因此竞赛机器人的速度设定值和方向设定值都会频繁地发生变化，导致控制系统产生振荡影响稳定性。为了消除设定值频繁变化造成的系统

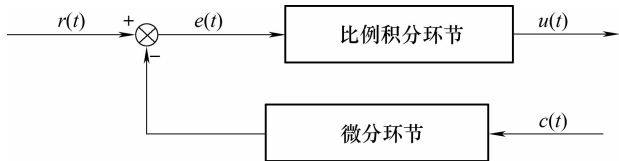


图 7-7 微分先行 PID 控制的结构图

振荡，我们在竞赛机器人的速度控制与方向控制上引入了微分先行 PID 算法，该算法只对输出量（机器人的速度测量值与舵机偏转量）进行微分。这样使得设定值的变化并不会影响到输出量，而被控量的变化相对比较缓和，这就很好地避免了设定值频繁变化给控制系统造成的振荡，有效改善了系统的动态性能。图 7-7 所示为微分先行 PID 控制的结构图。

7.2 PID 参数整定

PID 控制器的参数整定是控制系统能否正常运行的关键，它根据被控对象在控制过程中的动态特性来确定 PID 控制器的比例系数、积分时间及微分时间。PID 参数整定方法主要分为两大类。一是理论计算整定法，它主要依据系统的数学模型，经过理论计算确定控制器参数。这种方法得到的计算数据不能直接用，还必须通过工程实际进行调整和修改。二是工程整定方法，它主要依赖于工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。下面简单介绍几种主要的参数整定方法。

7.2.1 临界比例度法

首先将 PID 调节器的积分时间 T_I 置于最大，微分时间 T_D 置零，比例度（PID 调节器放大倍数的倒数， $\delta = \frac{1}{K_p}$ ）设置适当；当系统稳定后，将比例度逐渐减小，等得到等幅振荡过程时，记下临界比例度 δ_k 和临界振荡周期 T_k ；然后根据 δ_k 与 T_k 的值查询经验公式（见表 7-2），计算出 PID 调节器的主要参数 δ 、 T_I 、 T_D 的值；最后按照“先 P 后 I 最后 D”的操作程序微调调节器整定参数，使控制系统的稳定性、快速性、准确性达到最佳。

表 7-2 临界比例度法经验公式

	$\delta(\%)$	T_I/min	T_D/min
P	$2\delta_k$		
PI	$2.2\delta_k$	$T_k/1.2$	
PID	$1.6\delta_k$	$0.5T_k$	$0.25T_I$

7.2.2 衰减曲线法

将 PID 控制器设置为纯比例控制，比例度 δ 由大到小进行调节，同时观察系统控制过程

中出现衰减比为 4:1 时, 记录此时的比例度 δ_s 及振荡周期 T_s , 根据经验公式(见表 7-3)计算出对应的积分时间 T_i 与微分时间 T_d 。

表 7-3 衰减曲线法经验公式

	$\delta(\%)$	T_i/min	T_d/min
P	δ_s		
PI	$1.2\delta_s$	$0.5T_s$	
PID	$0.8\delta_s$	$0.3T_s$	$0.1T_s$

7.2.3 试凑法

凑试法是通过模拟运行观察系统的响应(例如阶跃响应)曲线, 然后根据各调节参数对系统响应的大致影响, 反复凑试参数, 直至达到满意的响应, 从而确定 PID 的调节参数。增大比例系数 K_p 一般将加快系统的响应, 这有利于减小静差。但过大的比例系数会使系统有较大的超调, 并产生振荡, 使稳定性变坏。增大微分时间 T_d 有利于加快系统响应, 使超调量减小, 稳定性增加, 但对于干扰信号的抑制能力将减弱。在凑试时, 可参考以上两种参数整定过程, 对参数进行先比例, 后积分, 再微分的整定步骤, 其具体步骤如下:

首先整定比例部分。将比例系数由小调大, 并观察相应的系统响应, 直至得到反应快、超调小的响应曲线。如果系统没有静差或静差小到允许的范围之内, 并且响应曲线已属满意, 那么只需要用比例调节器即可, 且最优比例系数可由此确定。

当仅调节比例调节器参数, 系统的静差还达不到设计要求时, 则需要加入积分环节。整定时, 首先将积分常数 T_i 设置为一个较大值, 经第一步整定得到的比例系数会略为缩小(如减小 20%), 然后减小积分常数, 使系统在保持良好动态性能的情况下, 静差得到消除。在此过程中, 可根据响应曲线的好坏反复修改比例系数和积分常数, 直至得到满意的效果和相应的参数。

若使用比例积分器能消除静差, 但动态过程经反复调整后仍达不到要求, 则可加入微分环节。在整定时, 先置微分常数 T_d 为零, 在第二步整定的基础上增大 T_d , 同时相应地改变 K_p 与 T_i , 逐步凑试, 以获得满意的调节效果和参数。

7.3 数字 PID 控制算法改进

为了解决机器人实际运行过程中遇到的问题以便提高 PID 控制性能, 我们对数字 PID 控制算法进行一些改进, 本节将主要就改进积分作用和微分作用进行介绍。

7.3.1 积分项改进

1. 积分分离

积分分离 PID 算法是在偏差较大时, 为避免超调量过大而导致系统响应时间延长, 暂时取消积分作用; 当偏差值小于设定阈值时, 再重新投入积分作用以保证系统的控制精度。

积分分离 PID 控制算法的表达式为

$$u(k) = K_p e(k) + \beta K_i \sum_{i=0}^k e(i) + K_D \frac{e(k) - e(k-1)}{T_s}$$

(7-11)

式中， T_s 为采样周期； β 为积分分离的开关系数，根据公式 $\beta = \begin{cases} 1 & |e(k)| \leq \varepsilon \\ 0 & |e(k)| > \varepsilon \end{cases}$ ， ε 值的确定要根据实际系统来定。若 ε 值过大，则达不到积分分离的目的；若 ε 值过小，则会导致系统无法进入积分区，从而使系统出现余差。

采用普通 PID 控制的阶跃响应曲线如图 7-8 所示。采用积分分离 PID 的阶跃响应曲线如图 7-9 所示。

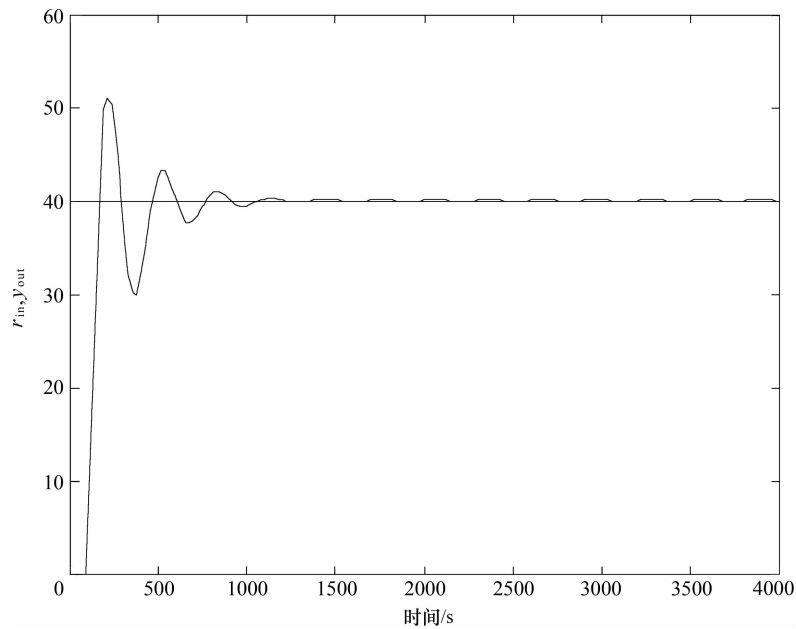


图 7-8 普通 PID 控制的阶跃响应曲线

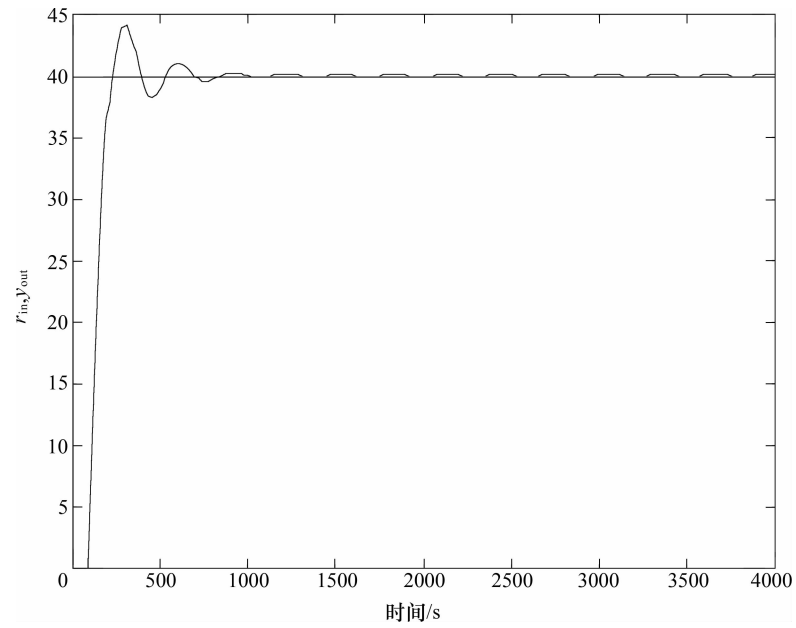


图 7-9 采用积分分离 PID 的阶跃响应曲线

从仿真结果可以看到系统的精度很好,采用积分分离 PID 控制算法,系统的控制效果有很大的改善。因此本系统可以应用于各种对精度要求较高的控制场合。

2. 变速积分

变速积分 PID 算法的积分项表达形式如下:

$$u_i(k) = T \left\{ \sum_{j=0}^{k-1} e(j) + f[e(k)] e(k) \right\} k_i \quad (7-12)$$

式中,系数 $f[e(k)]$ 为 $e(k)$ 的函数;当 $|e(k)|$ 增大时, $f[e(k)]$ 减小,反之则增大。

变速积分 PID 算法的基本思路是,根据偏差的大小,相应改变积分项累加的速度,偏差越大,积分越慢;偏差越小,积分越快。变速积分 PID 算法能较好地解决标准 PID 算法中积分系数取大会产生超调,甚至积分饱和;取小则又延长了消除静差的时间的矛盾。采用普通 PID 控制的阶跃响应曲线如图 7-10 所示,积分速率采用变速积分控制的阶跃响应曲线如图 7-11 所示。

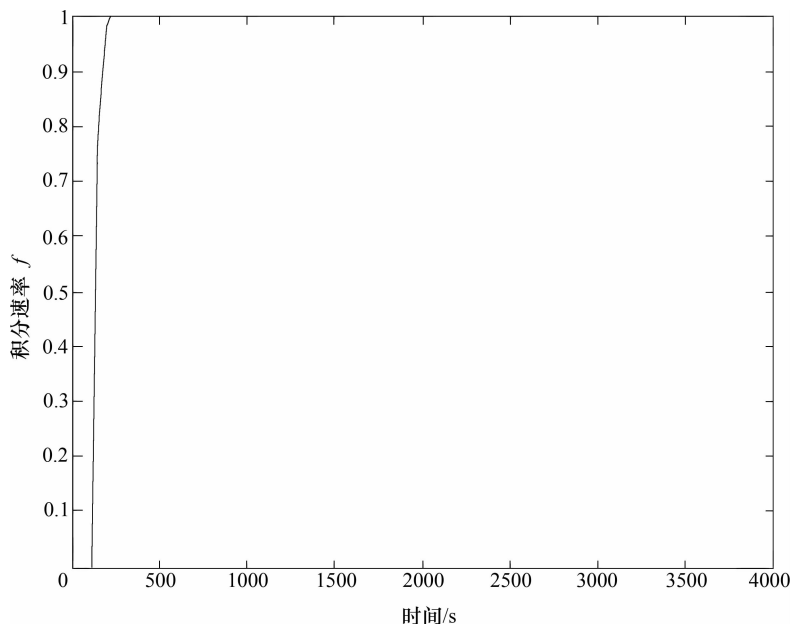


图 7-10 普通 PID 控制的阶跃响应曲线

由图 7-10 和图 7-11 的仿真结果可以看出,变速积分使得系统的调节时间有所减小,稳定精度有所提高。

7.3.2 微分项改进

由于微分作用是在相邻的采样周期内进行的,因此它的强弱不仅与微分时间 T_D ,放大系数 K_p 有关,而且与采样周期 T 也有明显关系。当 T 太小时,两次采样之间被控参数的变化一般也不会太大,因而微分作用较弱。所以通过增加微分时间 T_D 和放大系数 K_p 来加强采样周期内的微分作用,这样一来,系统的稳定性及抗干扰能力就会减弱,所以对微分项的改进应该从减少干扰与降低数据误差下手。

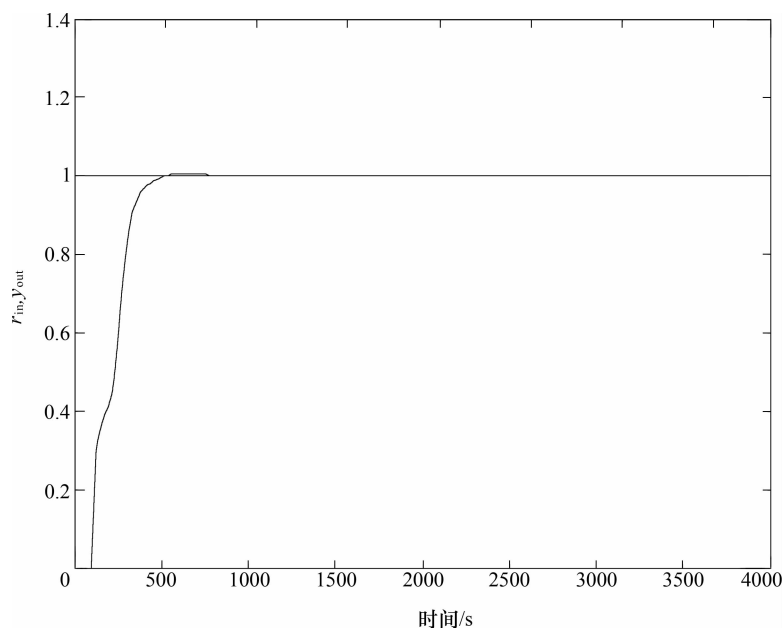


图 7-11 采用变速积分控制的阶跃响应曲线

1. 偏差平均

偏差平均的公式为

$$\bar{e}(n) = \frac{1}{m} \sum_{j=1}^m e(j) \quad (7-13)$$

式中，平均项数 m 取决于被控对象的特性。一般流量信号 m 取 10，压力信号 m 取 5，温度等缓慢变化的信号 m 取 2 或 1。

2. 测量值微分

当控制系统的给定值 $r(n)$ 发生阶跃变化时，微分动作将导致控制量 $u(n)$ 的大幅变化，这样不利于生产的稳定操作。因此，在微分项中不考虑给定值 $r(n)$ ，只对测量值 $y(n)$ （即被控量）进行微分。考虑到在正反作用下，偏差的计算方法不同，即

$$e(n) = y(n) - r(n) \quad (\text{正作用}) \quad (7-14a)$$

$$e(n) = r(n) - y(n) \quad (\text{反作用}) \quad (7-14b)$$

参照式(7-6)中的微分项

$$\Delta u_D(n) = K_D [e(n) - 2e(n-1) + e(n-2)] \quad (7-15)$$

改进后的微分项算式为

$$\Delta u_D(n) = K_D [y(n) - 2y(n-1) + y(n-2)] \quad (\text{正作用}) \quad (7-16a)$$

$$\Delta u_D(n) = K_D [r(n) - 2r(n-1) + r(n-2)] \quad (\text{反作用}) \quad (7-16b)$$

必须注意，对串级控制的副级控制的副调节器而言，因给定值是主调节器的输出控制量，故上述仅对测量值进行微分的做法并不适用，仍应按原微分项算式(7-15)对偏差进行微分。测量值微分也称微分先行。

7.4 PID 控制在竞赛智能车上的实现

智能车的速度取决于车模自身驱动电动机的转速，驱动电动机的转速受制于电动机驱动模块输出电流的大小，电动机驱动模块输出电流的大小受制于智能车控制核心——单片机输出的脉冲频率和占空比。当单片机输出的脉冲频率为定值时，脉冲占空比越大，驱动电动机转速越高；反之，转速就低。在智能车行驶过程中，如果赋予智能车驱动电动机以较高速度，则车模在直道行驶中不会有任何问题；但车模在通过弯道时，由于车模方向控制的机械系统具有惰性，车辆自身也具有惯性，因此车模极易从弯道的入口处冲出赛道。所以欲保证智能车在不冲出赛道的前提下达到一定的平均速度，则智能车的速度必须是可变的，即直道速度要快，弯道速度要慢，且速度控制系统的响应时间要短。因此，参赛选手在智能车速度控制系统的选取上多半采用由 PID 控制器组成的闭环控制系统。

7.4.1 PID 控制器输入标准值的设定

因为智能汽车在运行过程中，车速必须随赛道的变化而变化，所以车的速度对于某个路段来说是恒定的，而对于整个赛道来讲又是变速的。因此该系统中的 PID 控制器所要输入的标准值一定要受控于路径传感器的输出电压。基本方法是：对路径传感器输出的有效电压取绝对值，然后再乘以固定系数，用得到的积作为输入标准电压。

7.4.2 PID 控制器被控对象控制参量的设定

智能汽车中的驱动电动机是 PID 控制器的受控对象，电动机的速度由单片机输出的 PWM 脉冲频率和脉宽决定，在实际操作过程中采用频率固定、脉宽可调的方法进行控制。驱动电动机理想速度还受赛道材料的摩擦力、赛道路径变化等条件制约，所以驱动电动机的标准控制参量一般预设 3 个固定值，一个为直道时的控制参量，其余两个为不同角度弯道时的控制参量。预设的控制参量是否合适，还需要在车模试跑中予以调整，以保证车模在不冲出赛道时保持尽可能快的速度。

7.4.3 智能汽车车速 PID 控制器的工作原理

智能汽车车速 PID 控制器系统结构框图如图 7-12 所示。路径传感器在系统中负责检测赛道的变化情况，并完成将赛道位置信息转换为电压信息的任务。滤波器用以滤除路径传感器输出电压中的干扰成分，并负责对路径传感器输出电压进行一定的调整，然后将此电压作为车速 PID 控制器系统中的标准输入电压。采样电路用来检测电动机转速，并负责将速度信息转换为电压信息。反馈电路将采样电路输出的电压变换后形成送往比较器，与通过比较器求代数和，然后将值送往 PID 控制单元。PID 控制单元对该值进行处理，处理后的结果送往 PWM 控制电路，用以控制 PWM 电路输出的脉冲宽度，最终起到控制驱动电动机的作用。

普遍应用于工业自动化控制领域的 PID 控制器是一个具有反馈环节的自动化控制系统。在智能汽车车速控制中应用 PID 控制器可以有效控制智能汽车在不同赛道上的速度，且具有速率变化迅速及运行平稳的特点。需要注意的是，此系统中的输入标准值和输出标准值需要根据不同赛道参数和不同类型的电动机在实践中予以设定和调整。

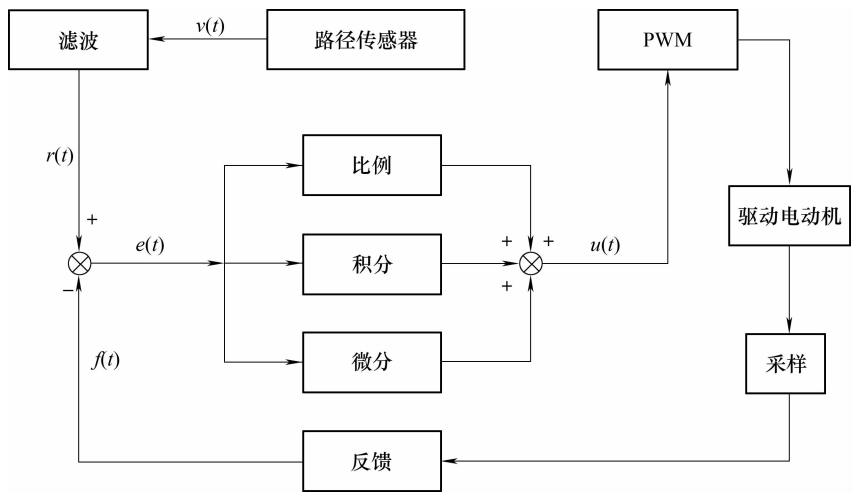


图 7-12 智能汽车车速 PID 控制器系统结构框图

7.4.4 舵机的 PD 控制

对舵机采用 PD 控制是各类竞赛智能车常用的控制方法，其表达式为

$$\text{output} = K_p e + K_i \sum e + K_d \Delta e \tag{7-17}$$

式中， e 为车身与跑道的偏移量； K_p 为比例系数； K_i 为积分系数； K_d 为微分系数；output 为舵机 PWM 输出量； Δe 为上一次偏移量与当前偏移量之差。在该 PD 控制中，起主要作用的是比例项。由于积分项会导致舵机转向延迟，故常将积分系数 K_i 置零。在实际比赛过程中适当调节微分系数 K_d 能使智能车优化路径，达到更好的转向效果。

7.5 模糊控制算法

7.5.1 模糊控制基本原理

竞赛机器人在场地运动的过程中，路径规划控制系统具有高度不确定性，是一个多输入多输出 (Multiple-Input Multiple-Output, MIMO) 系统。对于这种具有高度不确定性的 MIMO 系统，传统的控制方法不能达到很好的控制效果。模糊推理控制方法将人类的驾驶经验融入系统控制之中，因此可以较好地满足系统自适应性、鲁棒性和实时性的要求。模糊控制方式借助模糊数学这一工具，通过推理来实现控制。模糊逻辑模拟了人类思维的模糊性，它采用与人类语言相近的语言变量进行推理，因此借助这一工具可以将人类的控制经验融入系统控制之中，使得系统可以像有经验的操作者一样去控制复杂系统。总的说来模糊控制具有以下特点：①不依赖于被控对象的精确数学模型，易于对不确定性系统进行控制；②是一种易于控制、易于掌握的较理想非线性控制器，是一种语言控制器；③抗干扰能力强，响应速度快，并对系统参数的变化有较强的鲁棒性。模糊控制器的基本结构由模糊输入接口、模糊推理及模糊输出接口三个模块组成。模糊输入接口的主要功能是实现精确量的模糊化，即把物理量的精确值转换成语言变量值。语言变量的分档根据实际情况而定，一般分为 3~7 档，档数越多，控制精度越高，计算量也越大。模糊推理决策机构的主要功能是模仿人的思维特征，

根据总结人工控制策略取得的语言控制规则进行模糊推理，并决策出模糊输出控制量。模糊输出接口的主要功能是把模糊输出量转化为精确量，施于被控对象。

模糊控制系统结构如图 7-13 所示，其与传统控制系统最大的区别在于采用了模糊控制器。

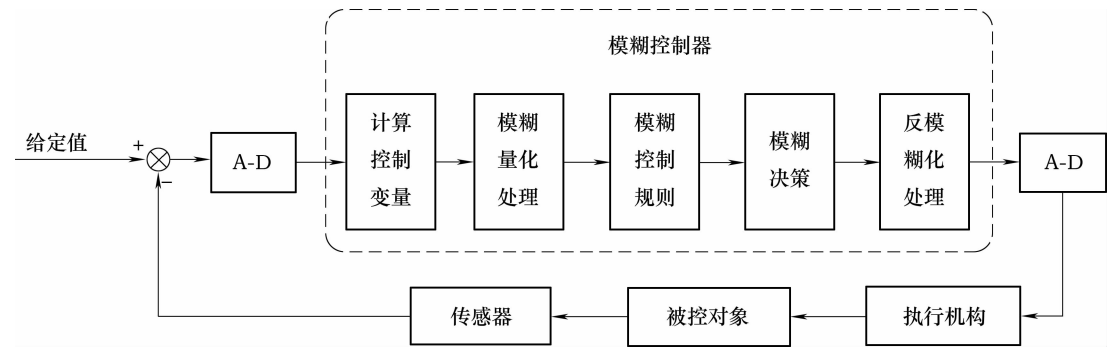


图 7-13 模糊控制系统结构图

模糊控制器主要由模糊化、知识库、模糊推理和清晰化这四部分组成，如图 7-14 所示。首先把输入的精确量转换成模糊量；知识库包含了具体应用领域的知识和要求的控制目标和规则等，由数据库和模糊规则库两部分组成；模糊推理是模糊控制器的核心，它具有模拟人的模糊概念推理能力，输入量经由模糊推理得到模糊控制量；再经过解模糊变换为用于实际控制的清晰量，作用于被控对象。

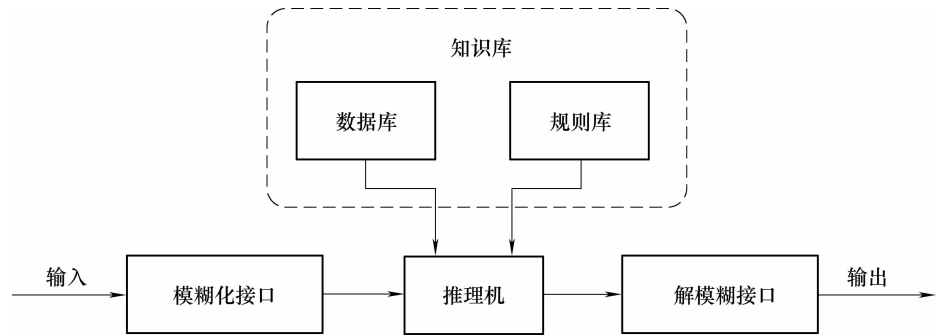


图 7-14 模糊控制器主要组成结构

7.5.2 模糊控制器的设计

模糊控制器输入输出语言变量的选取应该尽量简单，同时又要考虑机器人行驶和避让障碍物的实际情况，较好地反应机器人避障的过程，这里以避障机器人为例，设 d_r 为右侧障碍物距离， d_c 为前方障碍物距离， d_l 为左侧障碍物距离；方位传感器测得航向与目标点到机器人连线的夹角为 tg 。因此模糊控制器的四个输入变量分别是 d_r 、 d_c 、 d_l 和 tg ；模糊控制器的输出变量有一个，即机器人转向角 sa 。

模糊语言的确定包括由语法规则生成适当的模糊语言值，根据语义规则确定隶属度函数及模糊变量的论域等。这里使用连续型论域，采用简单线性化处理方法，对每个输入量进行模糊语言描述，如下：

距离输入变量： $d = \{\text{近}, \text{远}\} = \{\text{near}, \text{far}\}$

目标方位输入变量： $tg = \{\text{左大}, \text{左中}, \text{左小}, \text{前}, \text{右小}, \text{右中}, \text{右大}\}$
 $= \{\text{lb}, \text{lm}, \text{ls}, \text{zo}, \text{rs}, \text{rm}, \text{rb}\}$

输出转向角变量： $sa = \{\text{左转}, \text{左中转}, \text{左稍转}, \text{直走}, \text{右稍转}, \text{右中转}, \text{右转}\};$
 $= \{\text{tlb}, \text{tlm}, \text{tls}, \text{tz}, \text{trs}, \text{trm}, \text{trb}\}$

模糊分割是指模糊语言对所在论域的划分，决定了模糊控制的精细化程度。模糊分割的越多，控制越精细，但对应控制规则的数目也就越多；若模糊分割太粗，则难以对控制性能进行细致调整。所以，模糊分割的数目并无指导性的方法，主要根据具体情况和经验来划分。例如上面对距离输入变量的划分，通常情况下对距离变量划分为 $\{\text{近}, \text{中}, \text{远}\}$ ，但如此一来，模糊规则的数量共有 $3^3 \times 7 = 189$ 条，较难实现。因此将距离变量简单划分为 $\{\text{近}, \text{远}\}$ ，模糊规则数减少到 $2^3 \times 7 = 56$ 条。模糊规则数大幅减少，既容易实现，又未过多降低控制精度。

模糊语言值只是一个模糊子集，语言值要通过隶属函数来描述。在论域连续的情况下，隶属度常用函数的形式来进行描述，常见的隶属函数有三角形、梯形、高斯型等。通常，隶属函数的形状越陡，分辨率越高，控制灵敏度也就越高；相反，若隶属函数变化缓慢，则控制特性也平缓，对应的系统稳定性好。因此，在选择隶属度函数时，一般在误差为零的区域附近采用分辨率较高的隶属函数，而在误差较大的区域，可采用分辨率较低的隶属函数，以获得较好的鲁棒性。此外，大多数情况下，隶属函数的选取具有较大的随意性，对同一模糊集不同的人可以选取不同的隶属函数去表述，并无绝对的对错。因此，可以通过改变隶属函数来观察模糊控制器输出的变化。各输入输出变量隶属函数如图 7-15、图 7-16、图 7-17 所示。

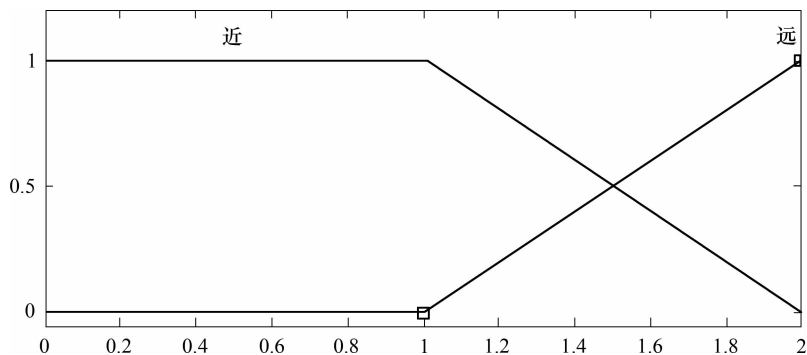


图 7-15 dl 、 dc 和 dr 的隶属函数

模糊规则的建立是模糊控制的核心问题。模糊控制系统是用一系列基于专家知识的语言来描述的，用一系列模糊条件描述的模糊控制规则就构成了模糊控制规则库。根据模糊控制理论，控制规则反映了输入输出变量间的关系。机器人转向的基本原则是：当探测到机器人左/右和前方有障碍物靠近时，机器人应及时转向右/左方。转向角度的大小则视障碍物距离和航向与目标点到机器人连线的夹角而定，障碍物距离越近，转向角越大；夹角越大，转向角越小。根据前面所确定的输入输出集，模糊规则的一般表述形式为“if(条件) then(结果)”。机器人一侧有障碍物是最基本的情况，这里仅以右侧有障碍物的情况来说明。当机

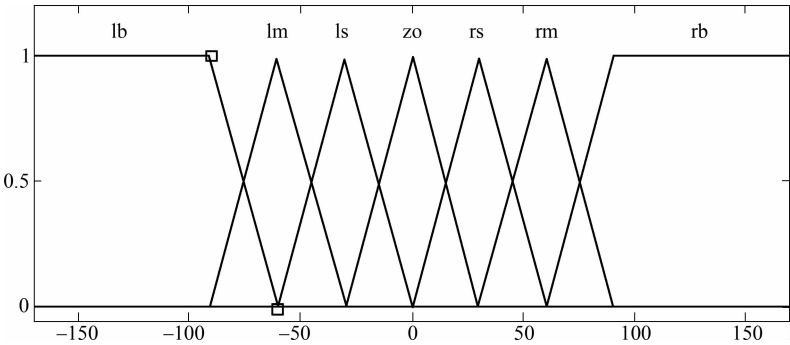


图 7-16 夹角 tg 的隶属函数

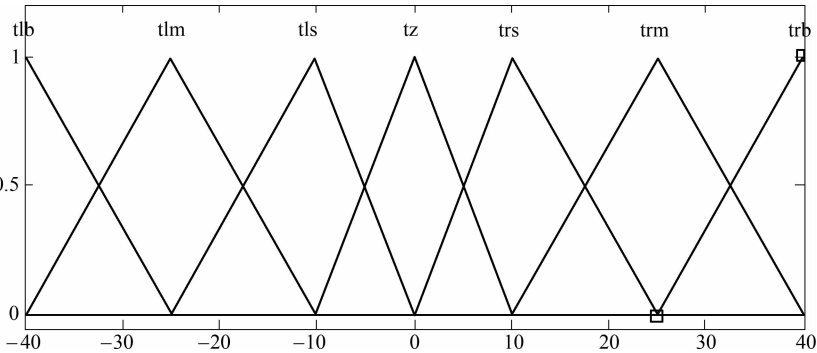


图 7-17 输出转向角 sa 的隶属函数

器人右侧近距离内有障碍物，而左侧和前方近距离内均无障碍物时。当左右障碍物对称时，根据基本规则，机器人将无法确定行进方向而在两障碍物间来回转向，形成死锁。在这种情况下，目标方位角 tg 则会决定机器人的转向，这就是局部最优选择。用模糊逻辑表示如下：

如果 dc 为近， dr 为远， dl 为远，且目标点在机器人左侧，则机器人左转；if (dr is near) and (dc is near) and (dl is near) and (tg is ls), then (sa is tlb)。

如果 dc 为近， dl 为远， dr 为远，且目标点在机器人右侧，则机器人右转。

if (dr is near) and (dc is near) and (dl is near) and (tg is rs), then (sa is trb)。

按照同样的方法，针对不同方位的障碍物和目标方位角，可制定完备的控制规则。这样的规则制定方法是基于控制器行为特征的方式，将动作分为若干的基本行为，对于较复杂的行为，则分由几个较简单的行为依次构成。这样可简化模糊规则的制定，减少模糊规则的条数，避开被控对象特征的建模。模糊控制规则经简化后，放入模糊规则库，供模糊控制器查询。

竞赛智能车多采用两轮驱动，通过对电动机转速与智能车位置的反馈，左、右轮驱动电路主要受 PID 调节，而对 PID 控制进行修正则可以使用模糊控制器。智能车运行过程的系统结构图如图 7-18 所示。

在转向控制算法中，如果仅采用 PID 控制算法，则为了使机器人行进顺畅，需要在直线运动偏差较小时缓慢调节甚至不调节；在遇到弯道产生大的偏差时，舵机能够迅速动作。这就使得偏差量比例系数 P 是一个变量，即该参数呈非线性特征，由于 P 是一个离散的数值，因此将导致转向调节的阶跃式变化，一方面机器人对路径变化反应不灵敏，另一方面也使得

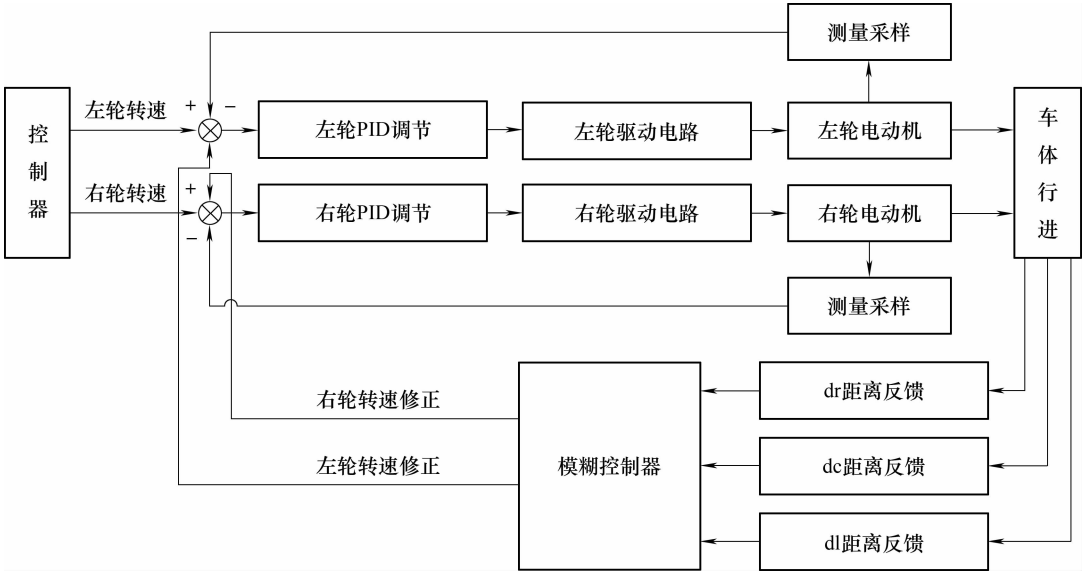


图 7-18 智能车运行过程的系统结构图

控制系统易产生超调及振荡现象。对于追求高车速、短决策周期控制策略的智能机器人来说，通过在 PID 调节的基础上辅助采用模糊控制算法，可以有效增强系统控制响应速度及对不确定因素的适应性。在寻迹与避障类竞赛中对机器人舵机及电动机控制多采用模糊控制算法。

7.6 赛道记忆算法

全国大学生智能汽车竞赛的规则为记忆算法在智能汽车控制系统中的应用提供了条件。比赛中智能车在赛道上连续跑两圈，并记其中最好圈成绩为比赛成绩。如图 7-19 所示，赛道记忆算法在第一圈以最安全的速度缓慢驶过一圈，并将赛道信息保存下来，第二圈根据保存下来的信息进行车速和转角决策的相应最优化，从而在第二圈取得好成绩。无论智能车的传感器前瞻距离有多远，在跑圈时它都只能预测在一段有限距离内赛道的情况。而采用赛道记忆算法的智能车，在第二圈时已对整个赛道有了全面的认识，从而在相同条件下，将比不使用赛道记忆的智能车更具优势。

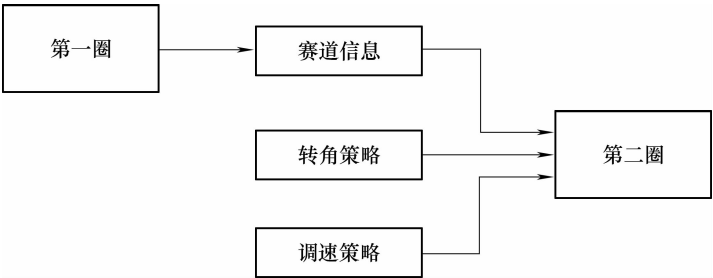


图 7-19 记忆算法原理

7.6.1 赛道记忆算法前提

要想成功实现赛道记忆算法，必须满足以下 4 点要求：

1) 赛车在第一圈必须安全走完全程。在第一圈，赛车的最主要目的不是为了快速跑完全程，而是采集赛道信息，如果智能汽车在第一圈出现冲出赛道的情况，则赛道记忆功能也会将错误的信息记录下来，导致第二圈在此处控制算法无法矫正偏差。因此使用赛道记忆算法的智能车，在第一圈一般都会采用较缓慢的匀速走完全程。

2) 赛车必须能明确分辨出赛道起点线和十字交叉线的区别。根据“飞思卡尔”杯智能车比赛的比赛规则，在赛道计时起点两边各有一个长度为 0.1m 的黑色计时起跑线，把智能汽车前端通过起始线作为比赛的开始或结束时刻。同时，由于赛道可以交叉，交叉角为 90° ，因此采用赛道记忆算法的智能车需要判断赛道起始线与交叉线，从而分辨出自己正在进行第几圈。

3) 赛车必须能够有足够记录一圈赛道数据的内存空间。由于单片机芯片的 RAM 容量有限，因此算法研究者需要对算法进行优化处理，改变保存的格式以节省内存空间。

4) 在跑第一圈时，赛车需即时对当前赛道数据进行处理，避免第二圈时还在进行数据处理而给赛车的控制实时性带来不良的影响。

7.6.2 赛道记忆算法描述

采用赛道记忆算法的智能车行驶过程如图 7-20 所示，第一圈的主要目的就是将赛道的重要特征按特定的格式记录下来，作为第二圈的参考。初圈最终要记录的数据的横坐标一般为行驶的路程，而纵坐标则为赛道此点的曲率半径。对于横坐标，由于初圈一般匀速驶完，因此可以直接用时间来表示，也可以用车速传感器的当前脉冲数来表示。而对于纵坐标则无法第一时间取得当前点的曲率半径，它需要对采集到的原始数据进行相关处理后才能得出。在初圈中，我们能记录的只是当前的转角设定值，可用舵机占空比或直接转换过来的角度值表示。

在第二圈，由于智能车本身的某些局限性，如舵机响应时间、赛道摩擦系数、轮胎抓地力、传感器布局等因素的影响，因此对于某种特定弯道，智能车有它所谓的极限速度。如果我们利用初圈得到弯道的曲率半径后，在过各弯道时将车速调至它相应的极限速度，那么理论上这就是它所能跑出的最优成绩了。对于直道，特别是长直道，更能发挥赛道记忆算法的优势。例如，如果在初圈记忆后得到某段直道为 2800 个脉冲点长，那么在决策中就可在前 2500 个脉冲时起动电动机用最大占空比加速，超过 2500 脉冲时直接减速，将速度调在下一个弯道的极限速度进行过弯。由于事先已经知道直道的长度，因此调速策略更能有的放矢；对于小 S 弯道，可以采取高速直线过弯的策略。为了安全起见也可以采用之后过弯的策略让智能车采取较小的转角高速过弯；对于圆形弯道，需要在弯道的曲率和过弯的速度之间做一个权衡，当行驶曲率较大时，智能车可以以较大的速度行驶，当进入内圈后，由于半径减小，离心力增大，此时宜降低车速平稳通过。由于智能车保有对初圈的记忆，因此跑第二圈时可以直接得出下个弯道是左拐还是右拐。这样智能车就可以按比规定更小的曲率半径通过赛道，从而减少了经过路径，也节约了时间。同样，一般赛车在直道上高速行驶时经常会出现来回振荡的现象。如果经过初圈记忆，则赛车判断自己正处于直道时可以进行相应的措施，便能很好地解决这个问题。

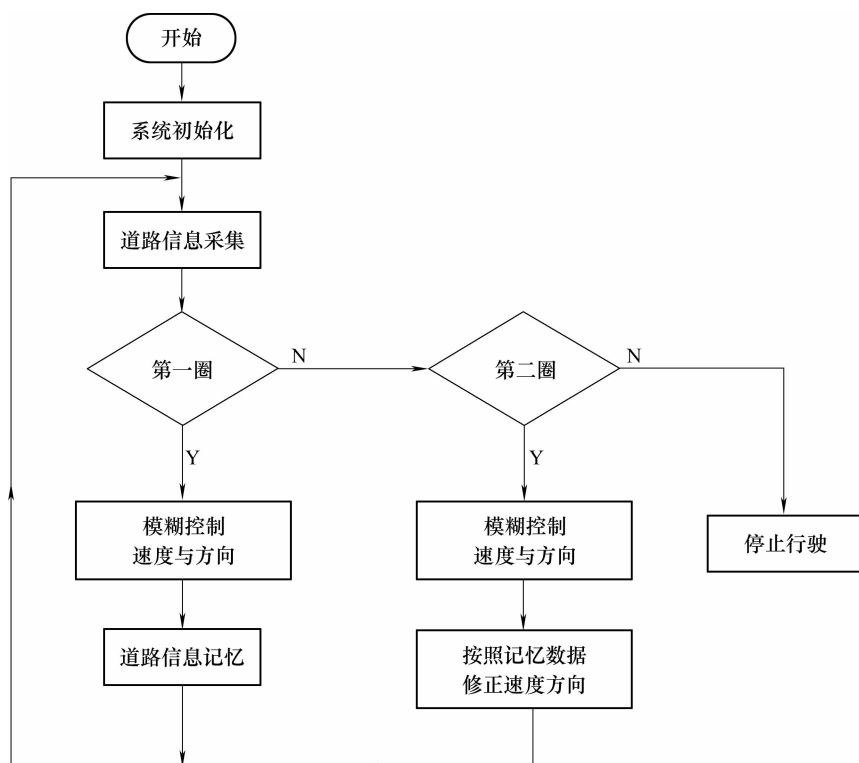


图 7-20 记忆算法软件控制流程图

虽然赛道记忆算法可以大大提高赛车的成绩，但目前来讲，在实际使用过程中想取得好的成绩，还需要做更深入的研究工作。原因主要有以下两点：

第一，目前大学生智能车比赛的赛道有着越来越复杂、越来越长的趋势，十字交叉线也将不只出现一个，这对初圈记忆和数据处理都产生了很大的挑战；

第二，由于决赛之前赛道是不公布的，因此赛道记忆算法存在很大的风险，较保守和安全的算法在决赛中可能更有利。

虽然赛道记忆算法存在一定的风险，但是它独特的思路却是值得继续发展下去的，有着很大的发展潜力。

参 考 文 献

- [1] 郭天祥. 新概念 51 单片机 C 语言教程——入门、提高、开发、拓展[M]. 北京：电子工业出版社，2009.
- [2] 霍顿. C 语言入门经典[M]. 北京：清华大学出版社，2008.
- [3] 刘金锟. 先进 PID 控制 MATLAB 仿真[M]. 3 版. 北京：电子工业出版社，2011.
- [4] 邴守东，李国林. 不完全微分 PID 控制算法研究与仿真实验[J]. 电子工业专用设备，2013，42(1)：46-50.
- [5] 周斌，刘旺，林辛凡，等. 智能车赛道记忆算法的研究[J]. 电子产品世界，2006，15：160-161.
- [6] 隋金雪，杨莉，张岩. “飞思卡尔”杯智能汽车设计与实例教程[M]. 北京：电子工业出版社，2014.
- [7] 席爱民. 模糊控制技术[M]. 西安：西安电子科技大学出版社，2008.

第8章 工程化应用实例

(竞赛)机器人的工程化装备应用,越来越受到业界的高度重视,其市场应用也越来越广阔。因此本章以目前智能制造中的典型装备——大型起重机控制系统,作为工程化应用实例进行说明和分析,同时将其作为信息物理系统(CPS)的经典案例剖析以飨读者。

8.1 交流电动机矢量控制理论

在大型起重装备中,从系统的可靠性、安全节能等方面考虑,都应用笼型感应电动机作为驱动装置。本节主要分析和研究笼型感应电动机的数学模型及其矢量控制系统。

8.1.1 感应电动机的空间矢量

任何在空间按正弦形式分布的物理量都可以用空间矢量表示。下面以感应电动机定子绕组为例,感应电动机的定子绕组均采用对称的三相绕组,当三相交流电流 i_A, i_B, i_C 流入三相绕组后,就形成旋转的绕组磁势。若略去空间谐波磁势,则各相绕组产生的基波磁势可分别用每相磁势空间矢量 $\vec{F}_A, \vec{F}_B, \vec{F}_C$ 表示,其长度代表基波磁势的大小,分别与各相电流的瞬时值成正比,其方向代表基波磁势幅值所在的方向,分别处于各相绕组的轴线上^[1]。基波磁势空间矢量如图 8-1 所示。

在图 8-1 中,若取轴线 A 作为实轴 Re , 虚轴 Im 领先于实轴 Re , 则在此电动机剖面的空间复平面内,有下列关系式成立:

$$\vec{F}_A = F_A e^{j0^\circ} = F_A; \vec{F}_B = F_B e^{j120^\circ} = a F_B; \vec{F}_C = F_C e^{j240^\circ} = F_C e^{-j120^\circ} = a^2 F_C$$

式中, F_A, F_B, F_C 分别表示各相基波磁势的幅值。

$$a = e^{j120^\circ} = \cos 120^\circ + j \sin 120^\circ = -\frac{1}{2} + j \frac{\sqrt{3}}{2}$$

根据矢量合成法则,将各相磁势空间矢量 $\vec{F}_A, \vec{F}_B, \vec{F}_C$ 相加,可得到三相合成磁势空间矢量。取合成磁势的 $2/3$ 定义为磁势空间综合矢量(简称为定子磁势空间矢量),用 \vec{F}_S 表示,则有

$$\vec{F}_S = \frac{2}{3}(\vec{F}_A + \vec{F}_B + \vec{F}_C) \quad (8-1)$$

磁势空间矢量可表示为

$$\vec{F}_S = \frac{2}{3}(F_A + a F_B + a^2 F_C) \quad (8-2)$$

由于磁势的大小和电流成正比,因此在三相系统中,若各相的相电流瞬时值分别用 $i_A,$

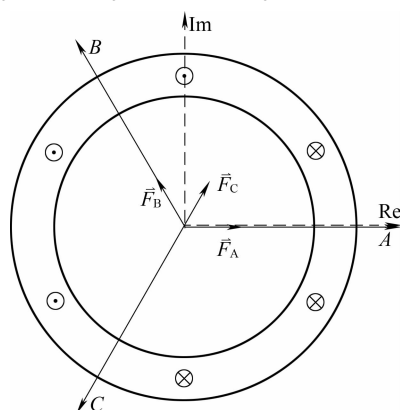


图 8-1 基波磁势空间矢量

i_B, i_C 表示, 则这三个瞬时值电流构成一个定子电流空间矢量, 用 \vec{I}_s 表示。

$$\vec{I}_s = \frac{2}{3}(\vec{I}_A + \vec{I}_B + \vec{I}_C) = \frac{2}{3}(i_A + ai_B + a^2i_C) \quad (8-3)$$

电流空间矢量在复平面上的表示如图 8-2 所示。一般说来, 电流空间矢量的长度和位置都是随时间变化的。

当三相电流没有零序分量(当电动机三相绕组是星形联结而中点不外接时, 此条件可以满足), 即 $i_A + i_B + i_C = 0$ 时, 可以证明, 磁势空间矢量 \vec{F}_s 在 A, B, C 三相轴线上的投影正好等于 $\vec{F}_A, \vec{F}_B, \vec{F}_C$, 且此时 \vec{F}_s 和电流 i_A, i_B, i_C 之间出现了一一对应关系。即任意一组的 i_A, i_B, i_C 将唯一地确定一个磁势空间矢量 \vec{F}_s , 反之亦然。实际上, 考虑到三相电流没有零序分量时的约束条件 $i_A + i_B + i_C = 0$, 一组电流 i_A, i_B, i_C 只包含两个独立变量, 而一个空间矢量 \vec{F} 也包含着实部的和虚部两个独立变量, 所以两者之间有可能一一对应。

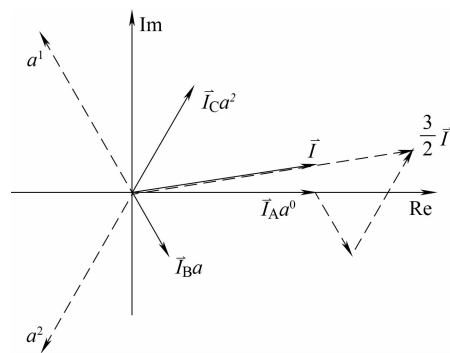


图 8-2 电流空间矢量

将三相电动机中各相的物理量综合起来, 用空间矢量表达, 具有下列优点:

- 1) 电流空间矢量代表三相合成电流, 磁链空间矢量代表三相合成磁场, 直观且形象化;
- 2) 由某物理量的空间矢量在某一相绕组轴线上的投影, 可得到该物理量的瞬时值。所以, 根据空间矢量的变化情况, 很容易求出各相有关物理量的时间波形图;
- 3) 可以证明, 电动机的转矩正比于磁链空间矢量和电流空间矢量的矢积, 所以在空间矢量图上能直观地确定电动机转矩的大小和方向;
- 4) 由于引入空间矢量概念时并不要求各相的物理量随时间作正弦变化, 因此, 变流装置输向电动机绕组的非正弦电流或非正弦电压, 也可以用空间矢量加以分析^[2]。

8.1.2 感应电动机矢量变换控制

1971 年德国学者 F. Blaschke 提出了感应电动机磁场定向型矢量变换控制理论, 实现了能找出两个分别决定磁通和电磁转矩的独立控制量, 并且求出了这两个控制量与直接测量及控制的定子坐标变量之间的关系式, 使得感应电动机的电磁转矩能像直流电动机一样解耦独立控制。因此在感应电动机的动态分析和控制中, 坐标系统的选择是关键, 采用合适的坐标系能使问题简化, 控制简单, 并准确获得良好的动态性能^[3]。

感应电动机矢量变换控制就是将用静止坐标系所表示的电动机矢量变换到磁场定向的坐标轴系, 这里包括转子磁场、定子磁场、气隙磁场。三相电流(可以是转子电流或定子电流)经过由三相静止坐标系到两相垂直静止坐标系, 再由两相垂直静止坐标系到两相旋转坐标系的变换, 并使 m 轴沿着磁链的方向, 则感应电动机就变成了由励磁电流分量 i_m 和转矩电流分量 i_t 分开控制的直流电动机。按照直流电动机的控制方法, 求得控制量后, 再经过坐标反变换, 就能控制感应电动机, 对感应电动机的控制转为对磁链参照系下的直流电动机的

控制。

矢量变换是一组矩阵表达式。矢量变换包括三相静止坐标系到两相静止坐标系的变换(称 Clarke 变换或 3/2 变换), 两相静止坐标系到两相旋转坐标系的变换(称 Park 变换), 以及直角坐标到极坐标的变换(称 K/P 变换)。

图 8-3 所示为三相静止坐标到两相静止坐标的变换, 图中的三相坐标系 A-B-C 变换到两相坐标系 α - β 。

从矢量合成的角度推导, 一个矢量从一个三相坐标系 A-B-C 变换到两相坐标系 α - β 也称为 3/2 变换, 反之为 2/3 变换。

现以电流矢量的变换来简单说明其变换关系式, 矢量 \vec{I} 在三相坐标系下可表示为 $\vec{I} = i_A + ai_B + a^2i_C$, 在 α, β 两相坐标轴系下可表示为 $\vec{I} = i_\alpha + ji_\beta$ 。根据坐标变换保持矢量 \vec{I} 不变的原理, 在矢量图中, 可以得到沿 α 轴分解的分量 $i_\alpha = i_A - \frac{1}{2}i_B - \frac{1}{2}i_C$, 以及沿 β 轴分解的分量 $i_\beta = \frac{\sqrt{3}}{2}i_B - \frac{\sqrt{3}}{2}i_C$ 。写成矩阵形式为

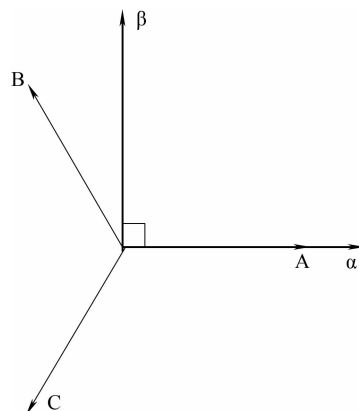


图 8-3 三相静止坐标系到两相静止坐标系的变换

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{3}{2} \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \quad (8-4)$$

则 3/2 变换矩阵 $C_{3/2}$ 为

$$C_{3/2} = \frac{3}{2} \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \quad (8-5)$$

如果三相平衡, 即 $i_A + i_B + i_C = 0$, 则可以推出

$$\begin{cases} i_\alpha = \frac{3}{2}i_A \\ i_\beta = \frac{3}{2}\left(\frac{1}{\sqrt{3}}i_A + \frac{2}{\sqrt{3}}i_B\right) \end{cases} \quad (8-6)$$

则简化后的矩阵形式为

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{3}{2} \begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_A \\ i_B \end{bmatrix} \quad (8-7)$$

由式(8-7)可以看出, i_α 的幅值是 i_A 幅值的 $\frac{3}{2}$ 倍, 这是保证变换前后合成矢量幅值不变的前提, 即三相和两相产生的合成电流矢量相同。同理, 可以推导出逆变换为

$$\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (8-8)$$

式中, 从两相静止 α - β 坐标系变换到三相静止 A-B-C 坐标系, $2/3$ 变换矩阵 $C_{2/3}$ 为

$$C_{2/3} = \frac{2}{3} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad (8-9)$$

图 8-4 所示为两种坐标系, α - β 为两相静止坐标系, m - t 为两相旋转坐标系。如果让 m - t 坐标以同步角速度 ω_s 旋转, 则产生的磁动势与 α - β 坐标系等效。 m 轴和 α 轴的夹角 γ (亦即 m 轴和 A 轴夹角) 是一个变量, 随负载、转速而变, 不同时刻有不同的值。

由此得变换关系式为

$$\begin{bmatrix} i_m \\ i_t \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (8-10)$$

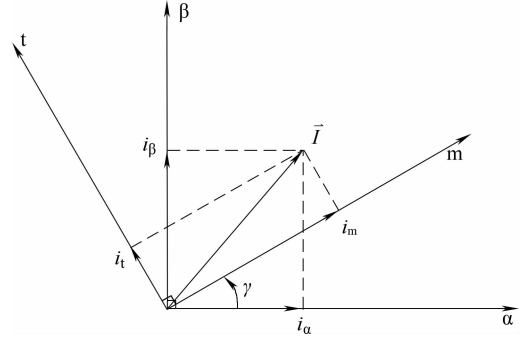


图 8-4 m - t 轴系与 α - β 轴系关系图

由图 8-4 可见, 从两相静止坐标系 α - β 到

磁链坐标系 m - t 的夹角即磁链位置角 φ_s , α - β 坐标系变换到 m - t 坐标系的变换矩阵为

$$C_{2s/2r} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \quad (8-11)$$

同理, 可以推出反变换矩阵为

$$C_{2r/2s} = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \quad (8-12)$$

如果直接由三相坐标系直接变换到旋转坐标系, 则将旋转变换式(8-12)与式(8-5)相乘, 得到 A-B-C 坐标系直接变换到旋转坐标系 m - t 的变换矩阵

$$C_{3s/2r} = \frac{2}{3} \begin{bmatrix} \cos\gamma & \cos(\gamma - 120^\circ) & \cos(\gamma + 120^\circ) \\ -\sin\gamma & -\sin(\gamma - 120^\circ) & -\sin(\gamma + 120^\circ) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (8-13)$$

同理求得 m - t 轴系变换到 A-B-C 轴系的变换矩阵为

$$C_{2r/3s} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 1 \\ \cos(\gamma - 120^\circ) & -\sin(\gamma - 120^\circ) & 1 \\ \cos(\gamma + 120^\circ) & -\sin(\gamma + 120^\circ) & 1 \end{bmatrix} \quad (8-14)$$

在计算磁动势时, 采用极坐标系能简化计算和分析。设合成矢量与 m 轴的夹角为 φ , 则

直角坐标与极坐标其变换关系式

$$i_s = \sqrt{i_m^2 + i_t^2} \quad (8-15)$$

$$\varphi = \arctan \frac{i_t}{i_m}, \text{ 由于 } \tan \varphi \text{ 幅值变化过大, 因此常用公式 } \varphi = 2 \arctan \frac{i_t}{i_s + i_m} \quad (8-16)$$

由于在理论分析和实际运行过程中, 双馈交流感应电动机的磁链、电压、电流等物理量都是以其基波形式来进行研究的, 并且电动机内部绕组采用了分布、短距、斜槽等措施, 其气隙磁场具有很小的谐波, 因此将其视为“理想感应电动机”来进行研究和分析, 如图 8-5 所示。“理想绕线式感应电动机”的假定如下^[4]:

1) 忽略铁磁饱和、磁滞及涡流等的影响, 不计铁心和导线的集肤效应;

2) 转子为圆柱形, 气隙均匀, 忽略定子、转子齿槽的影响, 认为定子、转子都有光滑的表面;

3) 忽略温度变化对于电动机有关参数的影响;

4) 定、转子三相绕组对称, 每相绕组均在气隙中产生正弦形分布的磁通密度。

在三相静止坐标系下, 异步电动机的定子三相对称绕组 A, B, C 在空间是固定的, A 相绕组的中轴线是该坐标系的参考坐标轴, 转子三相对称绕组 a, b, c 以转子角速度 ω_r 逆时针旋转, 转子 a 轴和定子 A 轴夹角为 θ 。

并规定各绕组电压、电流、磁链的正方向符合电动机惯例和右手螺旋定则。这时感应电动机的数学模型由电压方程、磁链方程、转矩方程和运动方程组成^[5]。

(1) 三相坐标系下的电压方程。

三相定子电压平衡方程为

$$\begin{cases} u_A = i_A R_A + p\psi_A \\ u_B = i_B R_B + p\psi_B \\ u_C = i_C R_C + p\psi_C \end{cases} \quad (8-17)$$

相应的三相转子折算到定子侧后的电压方程为

$$\begin{cases} u_a = i_a R_a + p\psi_a \\ u_b = i_b R_b + p\psi_b \\ u_c = i_c R_c + p\psi_c \end{cases} \quad (8-18)$$

式中, $u_A, u_B, u_C, u_a, u_b, u_c$ 分别为定、转子各相电压; $i_A, i_B, i_C, i_a, i_b, i_c$ 分别为定、转子各相电流; 设 R_s, R_r 分别为定、转子相电阻, 由于定、转子绕组对称, 因此 $R_A = R_B = R_C = R_s, R_a = R_b = R_c = R_r$; $\psi_A, \psi_B, \psi_C, \psi_a, \psi_b, \psi_c$ 为各绕组磁链; $p = \frac{d}{dt}$ 为微分算子。

写成矩阵形式为

$$[U] = [R][I] + p[\psi]$$

即

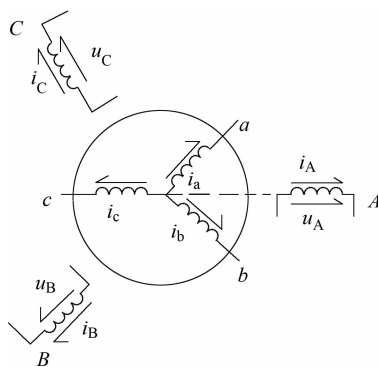


图 8-5 绕线式三相感应电动机示意图

$$\begin{bmatrix} u_A \\ u_B \\ u_C \\ u_a \\ u_b \\ u_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 & 0 & 0 \\ 0 & 0 & R_s & 0 & 0 & 0 \\ 0 & 0 & 0 & R_r & 0 & 0 \\ 0 & 0 & 0 & 0 & R_r & 0 \\ 0 & 0 & 0 & 0 & 0 & R_r \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \\ i_a \\ i_b \\ i_c \end{bmatrix} + p \begin{bmatrix} \psi_A \\ \psi_B \\ \psi_C \\ \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} \quad (8-19)$$

式中,

$$\begin{aligned} [U] &= [u_A \quad u_B \quad u_C \quad u_a \quad u_b \quad u_c]^T \\ [I] &= [i_A \quad i_B \quad i_C \quad i_a \quad i_b \quad i_c]^T \\ [R] &= \text{diag}[R_s \quad R_s \quad R_s \quad R_r \quad R_r \quad R_r]^T \\ [\psi] &= [\psi_A \quad \psi_B \quad \psi_C \quad \psi_a \quad \psi_b \quad \psi_c]^T \end{aligned}$$

其中对角线元素 L_{AA} , L_{BB} , L_{CC} , L_{aa} , L_{bb} , L_{cc} 是各有关绕组的自感, 其余各项则是绕组间的互感。

(2) 磁链方程。

$$[\psi] = [L][I]$$

其中每个绕组的磁链是它本身的自感磁链和其他绕组对它的互感磁链之和。

以定子为例, 每相绕组的全自感系数包括两部分, 一部分是只与该绕组交链但不穿过气隙的漏感 $L_{s\delta}$, 还有一部分是穿过气隙且与其自身绕组相交链的主磁链对应的主电感 L_{sh} , 因此定子每相全电感

$$L_{AA} = L_{BB} = L_{CC} = L_{sh} + L_{s\delta}$$

同理, 转子每相全电感

$$L_{aa} = L_{bb} = L_{cc} = L_{rh} + L_{r\delta}$$

两相绕组之间只有互感。定子和转子的三相彼此之间位置都是固定的, 它们的轴线在空间的相位差是 $\pm 120^\circ$, 假定气隙磁通为正弦分布的条件下, 互感值为 $L_{sh} \cos 120^\circ = L_{sh} \cos(-120^\circ)$

$= -\frac{1}{2}L_{sh}$, 于是

$$\begin{cases} L_{AB} = L_{BC} = L_{CA} = L_{BA} = L_{CB} = L_{AC} = -\frac{1}{2}L_{sh} \\ L_{ab} = L_{bc} = L_{ca} = L_{ba} = L_{cb} = L_{ac} = -\frac{1}{2}L_{rh} \end{cases} \quad (8-20)$$

由于定子任一相与转子任一相之间的位置是变化的, 因此互感是转子位置角 θ 的函数

$$\begin{cases} L_{Aa} = L_{aA} = L_{Bb} = L_{bB} = L_{Cc} = L_{cC} = L_{sh} \cos \theta \\ L_{Ab} = L_{bA} = L_{Bc} = L_{cB} = L_{Ca} = L_{aC} = L_{sh} \cos(\theta + 120^\circ) \\ L_{Ac} = L_{cA} = L_{Ba} = L_{aB} = L_{Cb} = L_{bC} = L_{sh} \cos(\theta + 120^\circ) \end{cases} \quad (8-21)$$

式中, s 代表定子侧, r 代表转子侧; θ 为转子轴和定子轴夹角。

(3) 运动方程。

电动机的机械运动方程为

$$T_e = T_L + \frac{J}{n_p} \frac{d\omega_r}{dt} \quad (8-22)$$

式中, T_e 为电动机额定输出转矩; T_L 为负载转矩; J 为电动机转轴上总的转动惯量; n_p 为电动机极对数。

(4) 电磁转矩方程。

根据机电能量转换的理论可知, 极对数为 n_p 的感应电动机电磁转矩 T_e 处理后的表达式^[6]为

$$T_e = n_p L_h [(i_A i_a + i_B i_b + i_C i_c) \sin \theta + (i_A i_b + i_B i_c + i_C i_a) \sin(\theta + 120^\circ) + (i_A i_c + i_B i_a + i_C i_b) \sin(\theta - 120^\circ)] \quad (8-23)$$

8.2 大型起重机控制系统

交流电动机的气隙磁链定向控制框图如图 8-6 所示, 控制系统由以下几部分组成: 速度环控制器、电流环交、直流调节器、电压前馈、电压模型、坐标变换、转子励磁控制、SVPWM 模块和逆变模块等。给定的速度指令信号 n^* 与检测到的转子速度信号 n 比较, 经速度控制器的调节输出 T 轴电流控制器给定信号 i_{st}^* , M 轴电流控制器的给定信号 i_{sm}^* 设定值为零。电流采样得到的三相定子电流通过 $3/2$ 变换和旋转变换后分别为 M 轴和 T 轴电流调节器的反馈输入。电流的给定和反馈相比较, 经过电流交、直流控制器调节, 并与前馈单元输出电压相加作为三电平 SVPWM 的输入信号。三相输入信号调制 SVPWM 模块输出 12 路 PWM 波, 产生频率和幅值可变的电动机所需供电的三相正弦电压。

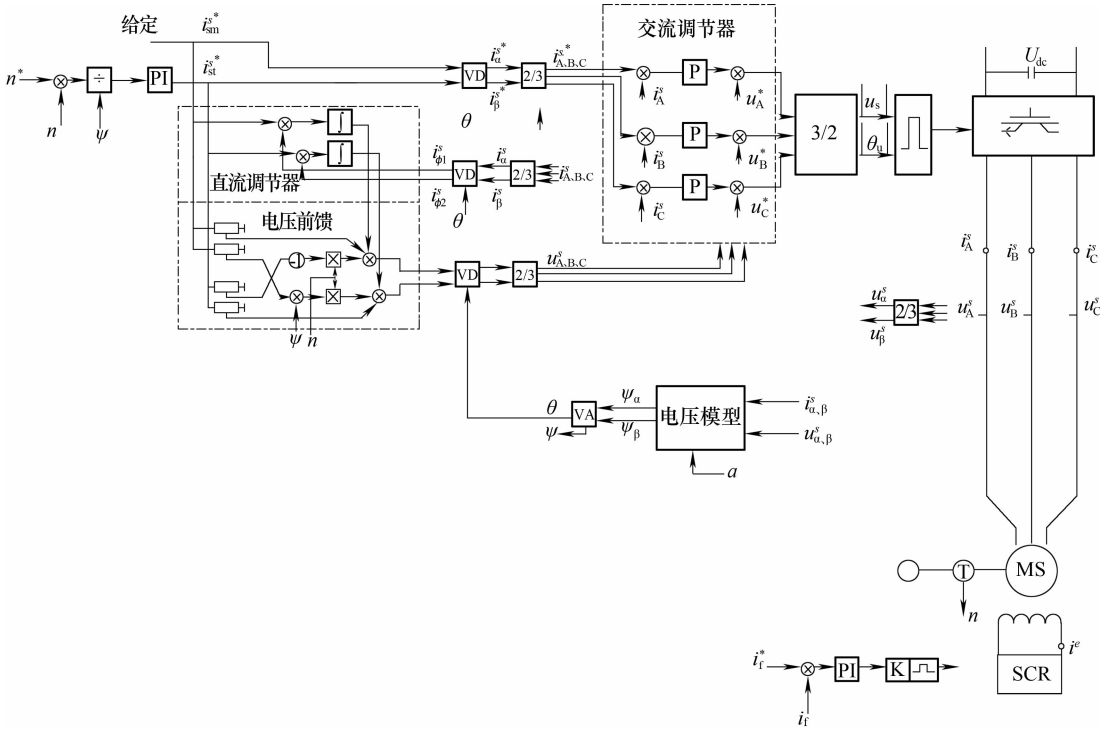


图 8-6 交流电动机矢量控制系统框图

针对异步电动机应用的高性能低压两电平交—直—交变频器(位能型负载、四象限), 对

整个控制系统主要由多个核心控制器、一个高性能工控机及显示器构成。它们之间通过通用现场总线通信协议构成以工控机为主站，其他设备为从站的以太网结构。倍福工控机实现对所有核心控制器的编程与控制，对它们的信息进行采集和汇总，实现对多台电动机之间的逻辑控制和协同控制，并实现对核心控制器的调试、监控功能。此外工控机可以实现与外部自动化系统的以太网连接。黑心控制器是自主开发的电力电子控制器，用于实现核心控制算法。每个核心控制器可以控制一台整流回馈变流装置和一个逆变装置，并通过工业以太网卡实现网络通信。除供电电源与网络接口外，核心控制器与外部设备的连接全部采用光纤通信，提高了系统的抗干扰能力，并且增强了可靠性。用于数据采集的接口模块，包括电压电流采样、数字输入、模拟输入、编码器输入等均通过光纤将信息传输至核心控制器，其发出的指令和变量也通过光纤传递给数字输出、模拟输出等模块。核心控制器通过光纤与变频器的功率接口模块连接，将 PWM 脉冲指令传递给变频器的功率模块。功率模块的电压电流、电动机转速反馈等信号经过采集和转换后经光纤反馈给核心控制器。监控系统或显示屏用于对整个系统的运行状态进行监控，显示各种运行变量和故障信息。控制器主要由电源供电模块、核心控制模块、光纤通信模块及网络通信模块等构成。电源供电模块为控制器各种元件提供不同电源供电。核心控制模块包括执行主要控制算法的数字信号处理器及实现大量数据处理和逻辑处理的可编程门阵列，且二者之间通过并行的数据地址总线进行数据通信，并通过中断的形式对执行动作的进程进行控制。光纤通信模块实现电信号与光信号的相互转换，使核心控制器与外部的数据交换和信息交流全部通过光纤来实现。网络通信模块采用工业以太网卡，使核心控制器可以向网络传输数据，也可以从网络接收数据。接口模块根据其功能主要分为数字输入、数字输出、模拟输入、模拟输出、采样模块、编码器输入等，采用自主研发设计的功能模块，用以实现外部设备与核心控制器之间的数据交换。根据实际需要，可以自由灵活配置。

(1) 数字输入模块。每个数字输入模块可同时实现多路开关量输入，用于各种开关型器件的状态采集。该模块通过光纤将采集到的开关量基于自行开发的通信协议传输给核心控制器。除了多路开关状态指示灯之外，该模块还具有电源状态和信号处理状态两个状态指示灯。

(2) 数字输出模块。每个数字输出模块可以同时实现多路开关量输出，用于输出各种开关型控制信号或状态信号。核心控制器通过光纤将开关量输出传输给该模块。同样，除多路开关输出状态指示灯外，该模块还具有一个电源指示灯和一个信号处理状态指示灯。

(3) 模拟输入模块。每个模拟输入模块可以同时实现多路模拟信号的输入，接线灵活，输入信号可以是电压信号、电流信号或热电偶信号。通过光纤将信息传输给核心控制器。具有电源指示灯和信号处理状态指示灯。

(4) 模拟输出模块。每个模拟输出模块可以同时实现多路模拟信号的输出，可以输出电压信号或者电流信号。核心控制器通过一路光纤将模拟输出信号传递给该接口模块。该模块同样具有电源指示灯和信号处理状态指示灯。

采用高性能工控机作为工业以太网控制网的主站，与网络中的各核心控制器（作为从站）、I/O 模块等构成主从站网络模式。工控机的主要作用是完成对不同核心控制器的编程控制，包括工作模式的选择、参数的设置及设备运行的监控等。核心控制器作为变频器控制系统的核心，需实现以下软件功能：变频器系统的控制逻辑；变频器控制所需电压，电流信

号的 A-D 采样；速度编码器的脉冲信号的处理，获取转速信息；系统保护功能，包括过欠电压保护、过电流保护，以及因其他各类故障引起的保护；核心控制算法，实现异步电机的三种控制方式，即 V/F 开环控制方式、无速度编码器的频率闭环矢量控制方式、有速度编码器的速度闭环矢量控制方式；不带速度编码器的 V/F 开环控制适用于简单应用场合和成组传动高水平同步运转，调速范围为 1: 10。磁场定向矢量控制用于高动态性能传动系统的频率闭环控制（无编码器）和转速闭环控制（有编码器）。实现准确地调节转矩电流分量和磁通电流分量，其调节频率为 2. 5kHz，在调速范围 1: 100 以内。具有外部附加力矩输入接点，外部电流给定输入接点，外部力矩给定输入接点；具有力矩给定输出接点，电流给定输出接点；具有系统自动辨识和优化功能和抱闸控制功能。PWM 脉冲产生模块根据由控制算法得出的控制指令产生相应的脉冲指令；自由功能模块配备具有不同功能的软件模块，分为信号模块、信号转换模块、计时器模块、逻辑模块、计算模块、控制模块等，使传动系统能够适用于各种不同的使用场合；可编程的输入输出模块配置有可编程的输入输出接口，即多个可编程开关量输入点、多个可编程开关量输出点、多个可编程模拟量输入点及多个可编程模拟量输出点的设置；网络通信功能模块配备有开发基于工业以太网卡通信程序，用以实现核心控制与工控机的网络通信；故障记录功能有核心控制器作为控制系统的核心，需对发生的所有故障尤其是重要故障进行记录，并通过本地存储或远程传输存储的方式进行保留，以备分析故障原因。

系统的性能指标如下：触发脉冲的最小脉冲频率为 1. 7kHz，设定频率为 2. 5kHz，最大设定频率为 16kHz；电流控制性能指标：基本负载电流为 $0.91 \times$ 额定输出电流；短时电流为 $1.36 \times$ 额定输出电流（对于过载时间 60s）；过载周期时间为 300s；额定输出频率为 10 ~ 60Hz；输出频率范围为 0 ~ 75Hz；变频器过载能力为 200%，1min；静态速度精度为 $\leq 0.01\%$ ；速度控制响应时间为 110ms；电流控制响应时间 $< 8\text{ms}$ ；动态指标为 $0.25\% \cdot \text{s}$ ；装置额定效率 $\geq 97\%$ ；输入谐波畸变率 $\leq 5\%$ ；输出谐波畸变率 $\leq 5\%$ 。速度控制性能指标见表 8-1。

表 8-1 速度控制性能指标

运行模式	压频比特性	闭环频率控制	闭环速度控制
数字式给定分辨率	0. 001Hz 31 位 + 标志		
模拟式给定分辨率	$f_{\text{max}}/2048$		
内部频率分辨率	0. 001Hz 31 位 + 标志		
速度准确率			
在 $n > 10\%$	0. 20%	0. 10%	0. 05%
在 $n < 5\%$	1. 00%	1. 00%	0. 1%
弱磁运行		0. 15%	0. 1%
速度上升时间		25ms 若 $n > 5\%$	20ms
恒定磁场范围		$< 2.5\%$ 若 $n > 5\%$	$< 2.5\%$ 若 $n > 1\%$
弱磁范围		$< 5\%$	$< 5\%$
转矩上升时间		$\approx 5\text{ms}$ 若 $n > 5\%$	约 5ms
转矩脉动		$< 2\%$	$< 2\%$

8.3 大型起重机安全监控管理信息系统

大型起重机安全监控管理信息系统是经典的信息物理系统 (Cyber Physical System, CPS), 本节以门座式大型起重机的 CPS 系统为例, 进行相关的说明和分析。

大型港口门座起重机安全管理监控系统设计和实施中要求采用成熟、先进的技术, 系统主要性能指标要达到国内先进水准。以经济实用为原则, 充分考虑初始一次性投资的规模, 并使系统今后运行时的成本最小化, 达到较高的性价比。系统配置要适合使用要求, 从而创造高效、安全、准确的环境, 达到安全生产的目的。

项目遵循总体设计、分步实施的原则, 整个系统要具有较高的可扩展性、安全性、可靠性和准确性。要求监控系统通过与起重机 PLC 控制系统通信, 达到对事件的快速处理、响应和存储功能。

大型港口门座起重机主要是为了对起重机进行实时监控管理而开发的一套监控系统。本项目安全管理系统设计需考虑如下子系统工程, 即实时状态监控系统、故障报警系统、历史故障查询系统、操作记录查询系统。监控系统可进行扩展, 留有足够的接口, 以满足整个监控系统的要求。

系统的安全防范系统的设计需满足以下国标中的各项要求: GB/T 28264—2012《起重机械安全监控系统》; GB 6067.1—2010《起重机械安全规程 第1部分: 总则》; GB 5226.2—2002《机械安全 机械电气设备 第32部分: 起重机械技术要求》。门座式起重机安全监控系统拓扑图(司机室触摸屏远程操作工控机)如图 8-7 所示。

显示本系统要求显示的起重机实时信息, 包括起重量力矩信息、环境风速信息、制动器状态信息、起升高度下降深度信息、其他限位信息等。工程浏览器左栏中单击“数据字典”, 在数据字典中新定义变量, 系统中各个限位器故障点接入 PLC 均为干接点开关量信号, 在组态王中选择开关类型为 I/O 离散型, 连接设备选择在前面所配置的 C200HE, 寄存器选择 IR 寄存器, 地址对应 PLC 梯形图中的地址。读写类型选择只读, 数据类型为位读取。按照上述步骤建立开关量采集点数据字典。模拟量设置: 系统中起重量传感器、风速传感器、幅度传感器等接入 PLC 模拟量模块 AD003, 均为模拟量信号, 在组态王中选择开关类型为 I/O 实数型, 连接设备选择在前面所配置的 C200HE, 寄存器选择 DM 寄存器, 地址对应采集点 PLC 梯形图中采集数据的存储区域。读写类型选择只读, 数据类型为 USHORT。AD003 模块的分辨率为 1/4000, 将变量最小原始值设置为 0, 最大原始值设置为 4000。最小值和最大值分别表示实际需要测量的量程, 按照如上步骤建立模拟量采集点数据字典。现场环境下风速仪通过 RS232 串口与工控机进行通信, 组态王中提供有串口数据发送的驱动, 在风速仪接入的串口新建“串口数据发送”设备。接收数据的处理过程代码如下, 自定义函数 transform() 实现串口数据的读取和显示。

1) 首先提取串口数据中的有效数据, 即风速字符。

\\本站点\提取的风速字符 = StrMid(\\本站点\接收的风速字符串, 6, 1);

2) 对提取的字符进行处理。StrASCII 返回某一指定的字符串变量首字符的 ASCII 值, 减去 48 即可得到整数值。代码如下:

\\本站点\风速字符整数值 = StrASCII(\\本站点\提取的风速字符) - 48;

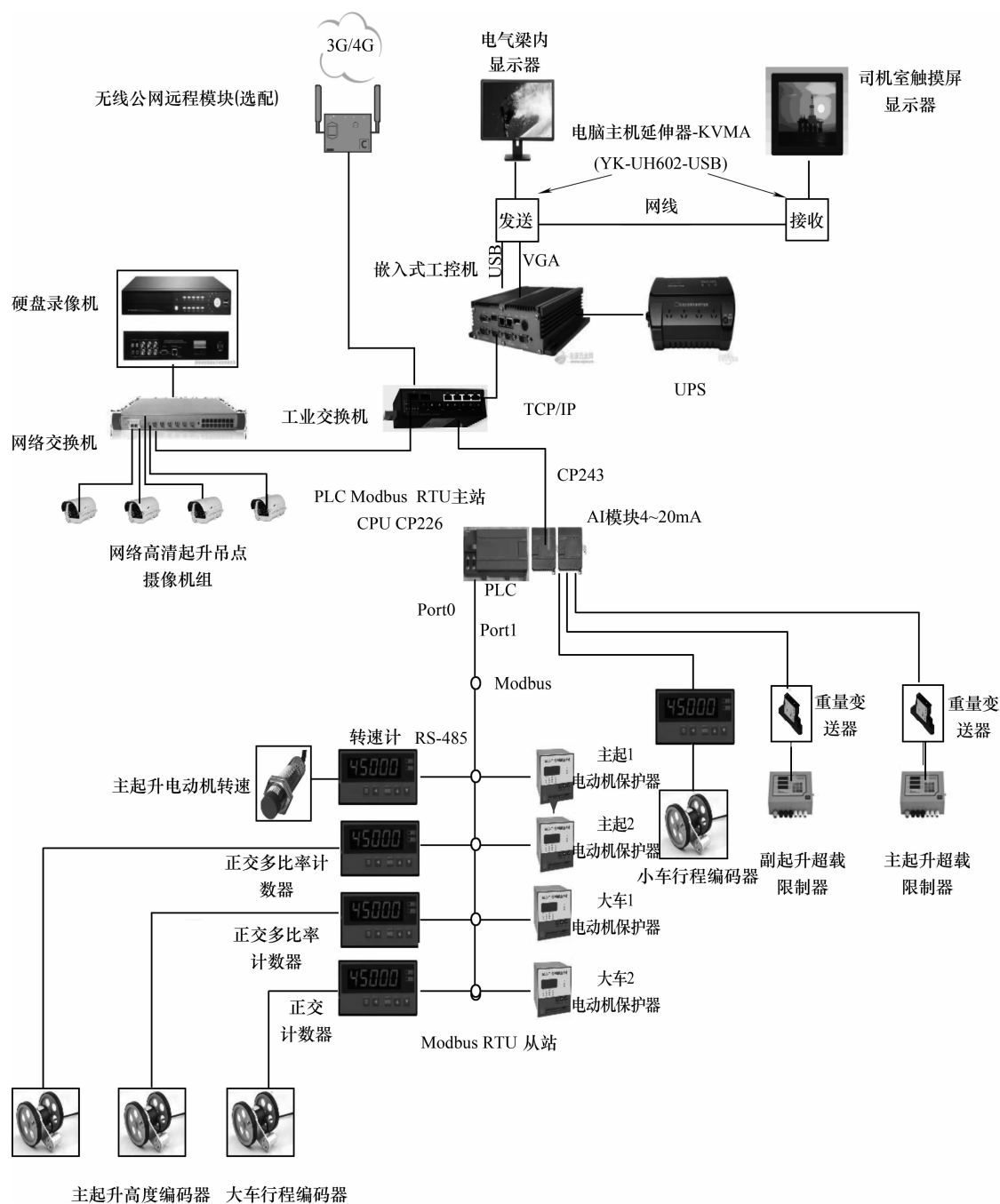


图 8-7 门座式起重机安全监控系统拓扑图(司机室触摸屏远程操作工控机)

3) 对获得的十六进制整数进行运算,代码如下:

```
float temp1;
```

```
if(\\本站点\风速字符整数 >= 0 && \\本站点\风速字符整数 <= 9) { temp1 = \\本站点\风速字符整数 * 16 * 16; }
```



```
else{\本站点\风速字符整数=本站点\风速字符整数-7;temp1=\本站点\风速
字符整数*16*16;}
\本站点\提取的风速字符=StrMid(\本站点\接收的风速字符串,7,1);
\本站点\风速字符整数=StrASCII(\本站点\提取的风速字符)-48;
float temp2;
if(\本站点\风速字符整数>=0&\本站点\风速字符整数<=9){temp2=\本站
点\风速字符整数*16;}
else{\本站点\风速字符整数=本站点\风速字符整数-7;temp2=\本站点\风速
字符整数*16;}
\本站点\提取的风速字符=StrMid(\本站点\接收的风速字符串,8,1);
\本站点\风速字符整数=StrASCII(\本站点\提取的风速字符)-48;
float temp3;
if(\本站点\风速字符整数>=0&\本站点\风速字符整数<=9){temp3=\本站
点\风速字符整数;}
else{\本站点\风速字符整数=本站点\风速字符整数-7;
temp3=\本站点\风速字符整数;}
\本站点\风速=(temp1+temp2+temp3)*0.1;
实时监控画面采集源包括的部分见表 8-2。
```

表 8-2 系统数据采集源

序号	数据采集源	传感器类型
1	起重量限制器	开关信号
2	力矩限制器	开关信号
3	起升高度下降深度限制器	凸轮限位开关
4	运行行程限制器	模拟值显示
5	幅度限制器	限位开关信号
6	门限位	限位开关
7	机构之间运行连锁	PLC 程序运行连锁
8	抗风防滑装置	行走锚定限位开关
9	风速仪装置	风速仪 RS-232 通信方式
10	同一轨道防碰撞装置	行走防碰撞限位开关
11	超速保护装置	模拟值显示速度
12	供电电缆安全限位	限位开关
13	起升机构制动器	制动器限位开关

注：因现场设备为 360 度全回转运行，所以不设置回转锁定装置。

系统数据采集源主要分成两类，一类是通过 PLC 采集，另一类是风速仪通过串口 RS-232 直接进入工控机。PLC 无源开关量信号主要由 C200HE 系列 PLC 的 IA222 开关数字量模块采集。开关量传感器开关信号直接连入 PLC 的 IA222 模块，PLC 模拟量信号主要由 C200HE 系列 PLC 的 AD003 模拟量模块采集。模拟量传感器开关信号直接连入 PLC 的

AD003 模块。系统可对所采集的数据源进行实时信息处理,系统可以实时监控 PLC 采集模块的采集量,对表中监控的参数进行实时显示。对表中起重机运行状态及故障信息进行实时记录,系统可设置超限监视的上下限,进行超限监视和报警处理,如超上、下限报警;报警方式采用声音和画面同时报警,显示报警画面、声音、报警灯变红色;报警确认可以关闭报警声音和将报警灯复位。运行数据统计和计算,对历史报警可以进行查询,方便管理。

工作循环记录采用主、副起升机重量记录;起始和终止时间;工作时长;系统可以对起重机累计工作时间进行记录,当起重机左右手柄都不在零位,即处于运行状态时,记录累积载荷时间;对操作员的操作进行记录,可以记录操作手柄的操作状况,包括起始时间、终止时间、操作名等信息,方便事后故障分析和认定。

参 考 文 献

- [1] 李崇坚. 交流同步电机调速系统[M]. 北京: 科学出版社, 2006.
- [2] 王君艳. 交流调速[M]. 北京: 高等教育出版社, 2003.
- [3] 高景德, 王祥珩, 李发海. 交流电机及其系统分析[M]. 北京: 清华大学出版社, 2001.
- [4] 许实章. 电机学[M]. 北京: 机械工业出版社, 1995.
- [5] 符曦. 感应电动机的矢量控制及应用[M]. 北京: 机械工业出版社, 1986.
- [6] T A Lip. "Dual Stator Winding Induction Machine Drive"[J]. IEEE Trans. on Industry Applications, 2000, 36(5): 1369-1379.

同系列书推荐

双足步行机器人

机器智能：人工情感

机器智能：人工心理

机器智能：人脸工程



机工微信



E世界



科技电眼

电话服务

服务咨询热线：010-88361066

读者购书热线：010-68326294

010-88379203

网络服务

机工官网：www.cmpbook.com

机工官博：weibo.com/cmp1952

金书网：www.golden-book.com

教育服务网：www.cmpedu.com

封面无防伪标均为盗版

上架指导 机器人

ISBN: 978-7-111-56929-9

策划编辑◎罗莉/封面设计◎MX DESIGN STUDIO

0-1765628429

ISBN 978-7-111-56929-9



9 787111 569299 >

定价：49.00元

为中华崛起传播智慧

地址：北京市百万庄大街22号

邮政编码：100037