中原传媒
CENTRAL CHINA MEDIA

普通高中教科书 通用技术 选择性必修

机器人设计与制作

绿色印刷产品

ISBN 978-7-5349-9649-8
9 787534 996498 >
定价：7.13 元

普通高中教科书 通用技术 选择性必修 机器人设计与制作

河南科学技术出版社

国家教材委员会专家委员会审核通过 2019

普通高中教科书

通用技术

机器人设计与制作

河南省基础教育教学研究室
河南科学技术出版社 组编

河南科学技术出版社

普通高中教科书

通用技术

机器人设计与制作

河南省基础教育教学研究室
河南科学技术出版社 组编

河南科学技术出版社
·郑州·

总 主 编：傅水根
本册主编：刘　荣
核心编者：王田苗　姚国才　高　凯
责任编辑：李晓慧　张春龙
美术编辑：张　伟
责任校对：徐小刚

普通高中教科书·通用技术（选择性必修）

机器人设计与制作

高中二年级

河南省基础教育教学研究室
河南科学技术出版社　组编

★

河南科学技术出版社出版发行
（郑州市郑东新区祥盛街27号）
邮政编码：450016　电话：（0371）65737028
河南日报报业集团有限公司彩印厂印刷
全国新华书店经销

★

开本：890mm×1 240mm　1/16　印张：6.25　字数：150千字
2020年3月第1版　2020年3月第1次印刷
ISBN 978-7-5349-9649-8
定价：7.13元

前言

尊敬的老师们，亲爱的同学们：

你们好！

新版的“通用技术”系列教材与大家见面了。这套新教材是在习近平新时代中国特色社会主义思想和社会主义核心价值观指导下，遵循教育部 2017 年新颁布的课程标准编写的。

高中阶段为什么要开设通用技术课程呢？

通用技术是与专业技术有所区别的技术，在当代技术体系中较为基础，在日常生活与生产中应用较为普遍。通用技术课程以立德树人、提高学生的技术学科核心素养为主旨，是一门来自生活与生产、面向全体学生、立足实践、注重创新、体现综合、科学技术与人文相统一的课程，着眼于培养德智体美劳全面发展的社会主义建设者和接班人。每本教材的编写，都有高中教师的积极参与。

纵观我国科技的发展，从群钻的发明、人工合成牛胰岛素，到治疗疟疾的青蒿素，再到为世界粮食安全做出重大贡献的超级水稻，以及为我国通信安全做出重大贡献的量子通信卫星，都说明我们中国人在科技领域开始走在世界的前列。要使我国由制造大国转变为制造强国，为中华民族的振兴和世界的繁荣做出更大的贡献，我们的基础教育和高等教育还需要深化改革，以培养出更多高素质、强能力和富于创造性的年轻一代。

当前，我国社会主义建设进入新时代。应用本套教材，我们将深刻理解技术，初识并感受设计的魅力，体验设计的创造乐趣；我们将认识设计中采用的 CAD/CAM 等软件和图样表达技术，在物化过程中采用的车工、铣工、钳工等常规制造工艺技术，先进的数控加工技术、激光雕刻技术、三维打印技术、机器人技术、无人机技术和智能家居技术等，会接触到互联网、大数据、云计算、物联网、人工智能和绿色生态技术。从难以忘怀的学习和历练中，同学们会受到创新意识、工程思维、工程素养和工匠精神的感染与熏陶，提高服务国家和人民的社会责任感，增强勇于探索的创新精神和解决复杂问题的能力。

通过情景导入、思维导图和设计任务引领，本教材充分展现“做中学”与“学中做”这一教育改革理念，并为此特意增添了“做中学”栏目。这里的“学”是在核心素养指导下，亲身经历将创意转化为设计的过程，培养学生在实践基础上的动手能力、实践能力或物化能力；而其中的“做”，就是“设计结合实践”。

这种“做”不是盲目的，而是在完成具体项目的复杂过程中，以学生为中心，以教师为主导，体现出团队的合作与交流，旨在实现从思维创意到设计，再到产品物化的不间断的、系统的、完整的迭代与优化。在学生的亲身经历和体验中，既有丰富、活跃、探究式的深度学习与能力转化过程，也有进一步思考与挖掘技术背后隐含的设计思想、思维方法和价值观等问题。

学生亲身经历的、与“项目”或“任务”密切关联的实践活动，在人才培养中具有非常重要的多种转化功能，即将知识转化为能力，将潜力转化为实力，将自疑转化为自信，将历练转化为素质，将聪明转化为智慧。那么，如何实现这些转化呢？那就是在实践中观察，在观察中思考，在思考中领悟，在领悟中成长。

本教材将“技术意识、工程思维、创新设计、图样表达、物化能力”这一核心素养贯穿始终，力求在实践中做到：符合现代科技发展的实际情况；体现学科交叉与融合的时代特征；与丰富的生活紧密联系，结构合理，满足学生多样化发展的需要；立足融合科学、工程、数学、技术、人文和社会的视野，体现劳动教育，突出创新精神、创新思维、实践能力和工程素养的培养。

在科技发展日新月异的今天，具备良好的核心素养、知识视野、实践能力和创新思维，是未来攻坚克难，成为国家栋梁的必备基础。我们会发现，身边到处充满着技术与设计的应用，到处展现着创造与发明的魅力，到处都有新时代青年施展才华的舞台。

本分册旨在帮助学生体会机器人设计与制作中软硬件协调、系统控制及路径规划的思想及方法，增强机械技术、电子技术、控制技术、计算机技术等的综合运用能力。全书共四章，介绍了机器人的发展历史、应用现状，基本构成和常用的机械结构；以开源硬件 Arduino 控制器为例，阐述了机器人的大脑——单片机的工作原理、开发流程及为控制器编程的方法；通过四个活动项目，比较和分析机器人常用的声音传感器、光敏传感器、红外线传感器和超声波传感器的特点及应用；结合巡线搬运机器人案例，介绍机器人常用的路径规划方法和控制策略。

尽管本套教材的编者们付出了极大努力，但囿于编者水平，仍会存在不足甚至错误之处，恳请广大师生在教与学的过程中，运用批判性思维方法，积极思考，发现问题，提出宝贵意见，以便在修订时加以改进与完善。

编　者

2019 年 3 月

目　录

第一章　机器人结构与传动机械……1
第一节　走进机器人世界……2
一、机器人的发展历史……2
二、机器人的应用……3
三、机器人的基本组成……6
第二节　简易机器人的设计……9
一、设计任务的提出……10
二、总体设计方案的制订……10
第三节　机器人常用的机械传动机构……11
一、齿轮传动机构……11
二、连杆传动机构……14
第四节　巡线搬运机器人机械结构的设计与制作……17
一、移动载体的设计与制作……17
二、搬运机械手的设计与制作……18
第二章　机器人控制器……24
第一节　单片机的基本构成与工作过程……25
一、单片机的基本构成……25
二、单片机的工作过程……26
第二节　认识 Arduino 控制器……28
一、Arduino 控制器简介……29
二、Arduino 扩展板简介……30
三、Arduino 控制器的开发流程……30
四、Arduino 控制器的开发环境……31
五、Arduino 控制器的开发语言……36
第三节　Arduino 控制器的应用一：LED 的编程控制……43
一、实验器件……43
二、实验内容……44

第四节　Arduino 控制器的应用二：移动机器人编程控制 …… 53
一、实验器件…… 53
二、实验内容…… 54
第三章　机器人感知与传感器……61
第一节　机器人常用传感器的种类和特点……62
第二节　声音传感器的原理及应用开发…… 63
一、声音传感器简介…… 63
二、声控机器人应用开发…… 64
第三节　光敏传感器的原理及应用开发…… 66
一、光敏传感器简介…… 66
二、光控机器人应用开发…… 66
第四节　红外线传感器的原理及应用开发…… 69
一、红外线传感器简介…… 69
二、红外线探路机器人应用开发…… 70
第五节　超声波传感器的原理及应用开发…… 72
一、超声波传感器简介…… 72
二、超声波避障机器人应用开发…… 73
第四章　机器人控制策略……77
第一节　机器人路径规划简介…… 78
一、机器人路径规划的定义…… 78
二、机器人路径规划的分类…… 78
三、送餐机器人的路径规划…… 79
第二节　机器人路径规划与运动控制实验…… 80
一、基础实验：机器人走正方形 …… 80
二、基础实验：机器人巡黑线 …… 81
三、基础实验：巡线中特殊路口的处理 …… 82
第三节　巡线搬运机器人的控制策略…… 84
一、控制程序的整体逻辑分析…… 85
二、抓取纸杯的运动控制…… 86
三、智能巡线的运动控制…… 86
四、程序的综合调试…… 91
附录　部分中英文词汇对照表……93

第一章
机器人结构与传动机械

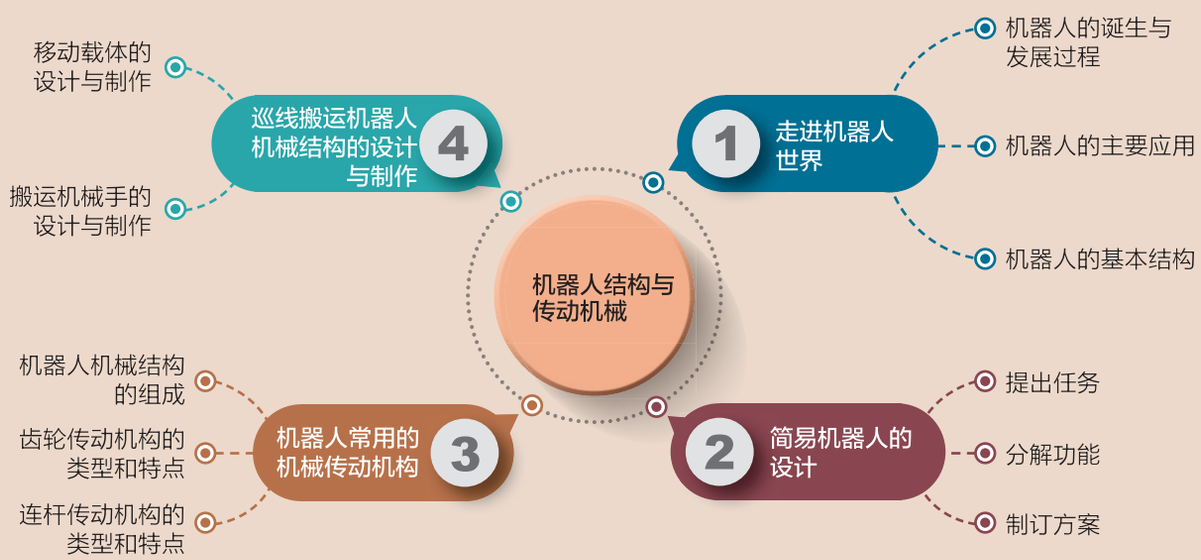
导　言

同学们，想象这样一个世界：你乘坐无人驾驶汽车飞驰在宽阔的马路上，下车后，你走进一家餐馆，机器人服务员热情地招待你，机器人厨师为你准备可口的饭菜，而扫地机器人正忙着打扫卫生。餐后，你还兴致勃勃地观看了机器人团体舞蹈队的精彩表演……这是科幻片？不，这些是已经出现在我们生活中的场景。

机器人作为一种能模仿人的某种活动的自动智能机械，已深入到人们生活的很多方面，在工业、农业、服务业、军事、航天等领域得到广泛的应用，在人们改造自然、促进社会发展的过程中发挥着极其重要的作用。

走近机器人，认识机器人，甚至亲手制作机器人，已成为很多青少年的憧憬。从本章开始，我们将一起走进机器人的世界，了解机器人的应用与发展，认识机器人的基本结构。通过一步步的学习，我们将为自己选择、设计、制作出一个机器人来。

思维导图



第一节　走进机器人世界

学习目标

1. 了解机器人的起源与发展。
2. 了解机器人的主要应用。
3. 思考并关注机器人发展过程中存在的伦理问题。
4. 知道机器人的基本组成。

相信同学们无论是在日常生活中，还是在影视作品中，都或多或少地接触过一些机器人。在 2016 年中央电视台春节联欢晚会上，540 台人形机器人（图 1.1）成方阵起舞，以整齐划一的优美舞姿引爆全场；在现代家庭生活中，扫地机器人（图 1.2）不用人管，十几分钟就可以打扫干净一间房屋。事实上，随着科技的发展，机器人早已成为我们人类的助手和朋友。那么，机器人是如何产生和发展的呢？机器人目前在人类和社会生活中扮演着什么角色呢？未来我们人类该如何与机器人相处呢？

图 1.1　人形机器人

图 1.2　扫地机器人

一、机器人的发展历史

机器人的英文是 robot，源于 1920 年捷克斯洛伐克作家恰彼克在科学幻想剧《万能机器人》中构思的一个人形机器罗伯特（源自 Robota，原意是劳役、苦工），他能说话，会走路，可以不知疲倦地进行连续工作。其实，人类对机器人的幻想和追求已有几千年的历史，如我国三国时期的指南车、欧洲 18 世纪的各种自动玩偶等。但真正意义上的机器人诞生在 20 世纪。表 1.1 给出了现代机器人发展的大事年表。

表 1.1　现代机器人发展的大事年表

年份	主要事件
1956 年	世界上第一家机器人公司 Unimation 在美国成立，并于 1961 年推出世界上第一台实用化工业机器人
1969 年	日本早稻田大学研发出世界上第一台用双脚走路的机器人
1978 年	美国 Unimation 公司推出一款通用工业机器人 PUMA，这标志着工业机器人技术已经进入成熟阶段
1996 年	世界上第一台自动扫地机器人“三叶虫”问世，并于 2001 年实现量产
2000 年	日本索尼公司推出机器狗“爱宝”，本田公司推出世界上第一台商业化人形机器人“阿西莫”
2003 年	火星探测机器人“勇气号”和“机遇号”登上火星执行科学考察任务
2012 年	我国“蛟龙号”水下机器人在马里亚纳海沟创造了下潜 7 062 m 的世界同类作业型机器人最大下潜深度纪录
2013 年	我国“嫦娥三号”探月机器人登陆月球，并创造了在月球工作时间全球最长纪录
2016 年	我国香港的汉森机器人公司推出人形机器人“索菲亚”，2017 年沙特阿拉伯授予她公民身份，她成为史上首个获得公民身份的机器人
2019 年	我国“嫦娥四号”成为世界上首次登陆月球背面开展科学考察任务的机器人

二、机器人的应用

（一）机器人为我们制造产品

现代工业品极大地丰富了人们的物质生活。当我们享用这些产品时，可曾想过其中不少产品都是出自机器人之手吗？现在，工业机器人已经在汽车、摩托车、家用电器等行业的生产线中广泛使用（图 1.3、图 1.4）。

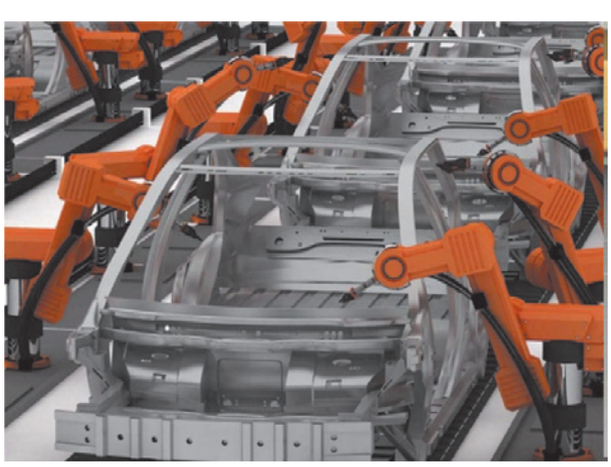


图 1.3　汽车生产线上的机器人

图 1.4　工厂中的物品搬运机器人

（二）机器人为我们提供服务

如何帮助人类摆脱繁重的劳动，并给人类提供更好的服务？不知疲倦、体贴入微的机器人当然是理想的选择。现在，已经有多种类型的机器人活跃在服务业的第一线，从事医疗、救援、家政等工作（图 1.5、图 1.6、图 1.7）。

图 1.5　手术机器人

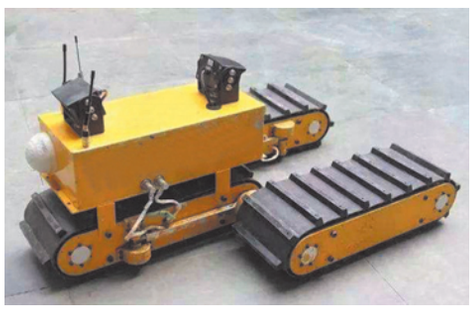


图 1.6　废墟搜救机器人

图 1.7　自动擦窗机器人

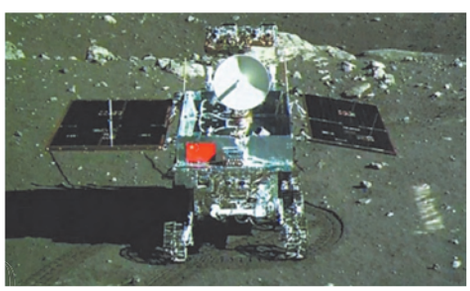


图 1.8　月球探测机器人

（三）机器人为我们探索未知世界

人类能生存的空间是有限的，而广袤的宇宙却是无限的，只有借助机器人才能把我们的探索能力延伸出去，拓展人类的活动空间。现在，机器人既能上九天揽月（图 1.8），又能下五洋捉鳖（图 1.9），甚至在炙热的火山口或者严寒的极地（图 1.10），都留下了它们的足迹。

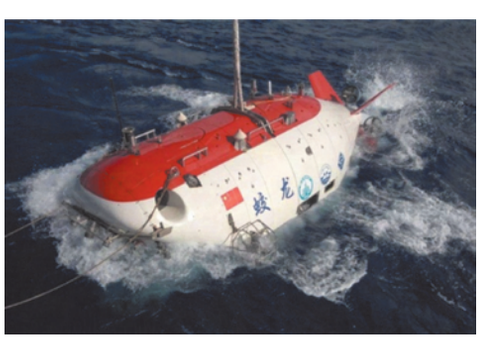


图 1.9　深海探测机器人

图 1.10　南极探测机器人

（四）机器人与人的关系

以上只是列举了机器人的几种典型应用，形形色色的机器人还有很多。从这些例子可

以看出，机器人是人类认识自然和改造自然的一种工具，它们并不一定需要具有人的外形和所有功能。我国科学家曾经对机器人下过这样的定义：机器人是一种自动化的机器，所不同的是这种机器具备一些与人或生物相似的视觉、听觉、嗅觉等感觉功能，可在人所不能适应的环境下代替人的工作。

需要指出的是，在发展智能机器人技术的过程中，我们也要认真思考机器人与人类的关系。一方面，机器人已经进入人类生活的方方面面，代替人类完成各种各样的工作，成为我们不可或缺的帮手和朋友。另一方面，我们也要看到机器人失控可能会给人类造成意想不到的伤害。据美国劳工部的统计数据，从 1987 年到 2017 年的 30 年间，美国发生了 30 多起与机器人相关的死亡事件。

2016 年 3 月，谷歌开发的围棋人工智能程序 Alpha Go 以 4 ∶ 1 的总比分战胜围棋世界冠军李世石，这让人们看到人类智慧的堡垒被机器人打破的可能性。2018 年 3 月在美国亚利桑那州发生首例无人驾驶汽车撞人事件，人们对责任归属问题展开了激烈的讨论。也许有一天，当机器人的智能达到一定程度后会产生自我意识，会因为自身的损坏而感到痛苦。到了那时，故意损害机器人，有可能会被认为是虐待劳工。机器人伦理学可能会走进课堂，而机器人劳动法可能会成为新闻热点。

总之，机器人的发展既有其对人类有利的一面，也暴露出其对人类潜在的威胁。因此，人类需要对机器人的研究与发展进行严格的约束。为了更好地使机器人与人类相处，一些科学家、政治经济学家制定了一些机器人与人类相处的原则，例如，著名的机器人三定律和英国工程和物理科学研究理事会 (EPSRC) 提出的机器人守则。

阅读材料

机器人三定律和 EPSRC 的机器人守则

为了避免机器人伤害人类，美国科幻作家阿西莫夫曾在其科幻小说《我，机器人》中提出机器人三定律：

第一定律——机器人不得伤害人，也不得见人受到伤害而袖手旁观；第二定律——机器人应服从人的一切命令，但不得违反第一定律；第三定律——机器人应保护自身的安全，但不得违反第一、第二定律。

对于阿西莫夫的机器人三定律，有些科学家持支持态度，而有些科学家则对其表示疑问与否定。EPSRC 的机器人守则中指出：设计和使用机器人的人类必须成为类似这种规则的遵守者及责任承担者。机器人的设计和操作必须遵守现行的各项法律法规、基本人权及包括隐私权在内的自由权。

探究与交流

结合机器人三定律及 EPSRC 的机器人守则，探讨一下无人驾驶汽车撞人事件的责任归属问题。

三、机器人的基本组成

机器人由于功能不同，形态也各异，但各类机器人在组成上存在共性，图 1.11 所示为一个典型机器人系统的组成。

图 1.11　机器人系统的组成

（1）机械结构：组成机器人的机械本体，如操作臂、末端执行器、行走机构等。

（2）驱动装置：驱动机器人关节的部件，如电动机、汽缸、液压缸等。驱动装置及机械结构组成机器人的机械系统。

（3）传感器：感知机器人内部状态及外界信息的元件，如关节旋转编码器、声音传感器、超声波传感器、气体传感器、光敏传感器、温度传感器等。

（4）控制器：负责接收传感器信息并指挥机器人的装置。

（5）动力源：为机器人各部分的正常工作提供能量的部件，如电源、气压源和液压源等。

机器人是由上述几个部分构成的一个有机整体。由传感器采集到的信息输入到控制器，后者对输入信号进行计算、判断和决策，然后将命令输出到驱动装置以便驱动机械结构。控制器中的控制程序是机器人控制的核心，它告诉机器人要做什么事情。通过编写不同的控制程序，机器人就可以执行不同的任务。

探究与交流

虽然大多数机器人在外形上与人类相差很大，但其结构组成与人相比还是具有相似性的，例如眼睛就是人体上相当于传感器的器官之一。试根据机器人各组成部分的特征，在人体上分别找出与之功能相似的部位或器官，完成表 1.2。

表 1.2　机器人结构与人体器官类比

机器人	机械结构	驱动装置	传感器	控制器	动力源
人体					

实践与体验

简易机器人小车的拼装及调试

【目的】体验简易机器人制作过程，感受机器人在程序控制下的自动运行。让同学们通过这一制作过程，进一步学习、理解机器人的系统组成。

【任务】移动是简易机器人最重要的功能之一，所以我们的制作就先从移动载体入手。同学们要精心完成这个机器人小车，因为后面的很多内容都离不开它。

【器材】可以购买市场上比较成熟的机器人套件，也可以单独购买控制器、电动机、车轮、传感器等器件。参考组装步骤如下：

第一步：按图1.12所列的零件和步骤组装机器人小车的电动机和驱动轮。

提示：电动机的两根固定长螺钉不要拧太紧，否则会影响电动机内部齿轮的转动，进而影响电动机的转动速度。

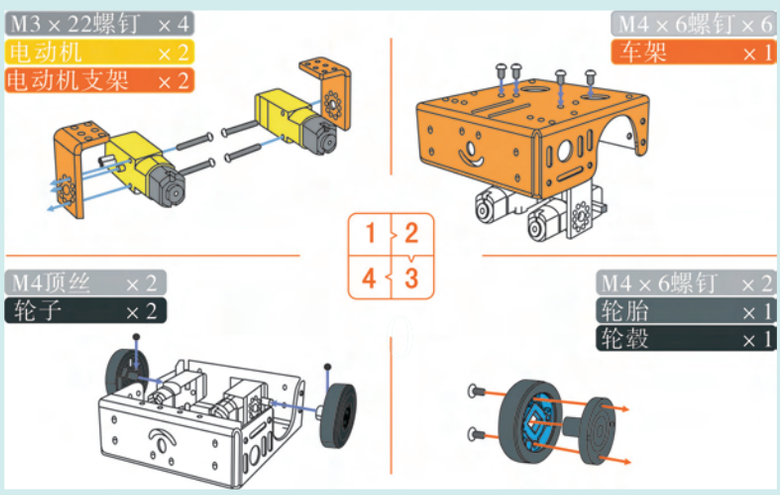


图 1.12　电动机和驱动轮的安装

第二步：按图1.13所列的零件和步骤组装机器人的万向轮和电池。

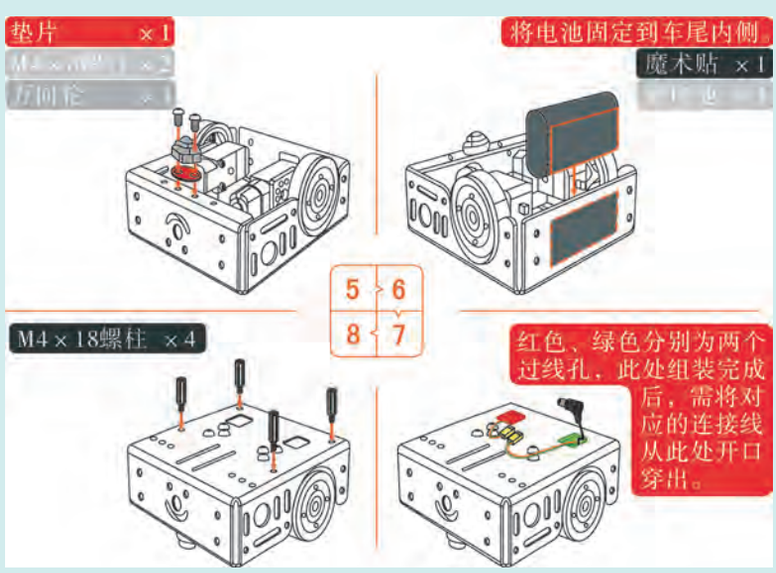


图 1.13　万向轮和电池的安装

第三步：按图1.14所列的零件和步骤组装机器人的控制器。此时小车已经初步组装好，但它不具备感知功能，还需为它装上各种传感器。

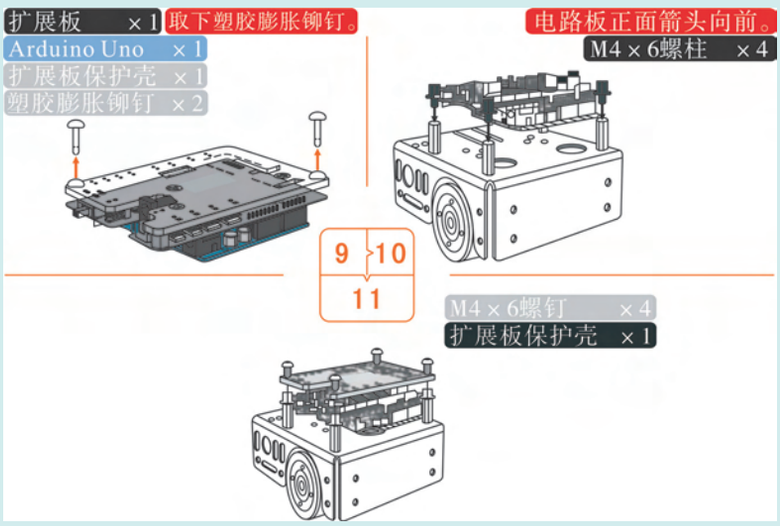


图 1.14　控制器的安装

第四步：按图1.15所列的零件和步骤组装机器人的巡线传感器模块。需要先卸下万向轮，再把巡线传感器模块夹在万向轮和机器人底盘支架之间。通过两个驱动轮的速度差实现转向，前端的万向轮起到支撑和导向的作用。

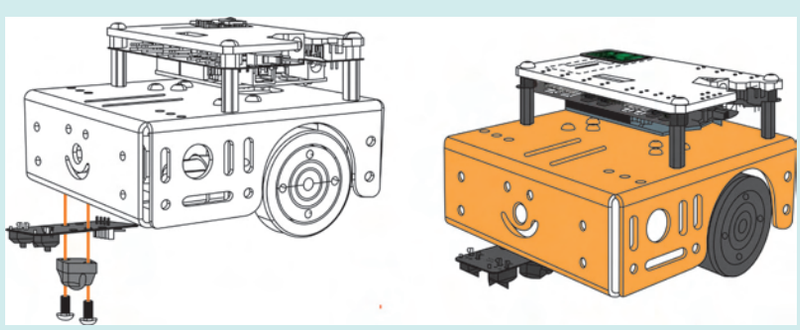


图 1.15　巡线传感器模块的安装

第五步：按图1.16所示，将电动机和传感器分别正确地连接到控制器上。

提示：巡线传感器模块接控制器的P14端口，两个电动机分别接控制器的P5和P6端口。

至此，一个完整的简易机器人小车就组装完成了。尽管我们还不理解它，但不妨先调用一下预存在控制器中的演示程序，试试它有什么能耐。先在一张白纸上画一条宽度约30 mm的黑线（或用黑胶带粘贴），然后将机器人小车放在黑线上，打开控制器的电源，观察机器人小车的运动（图1.17）。

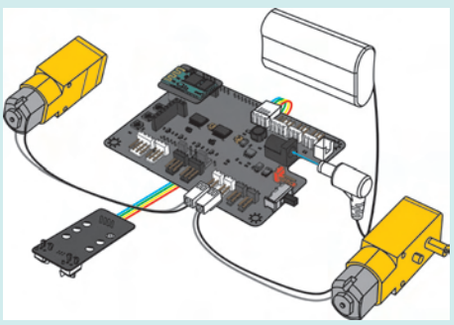


图 1.16　控制器与电动机和传感器的连接

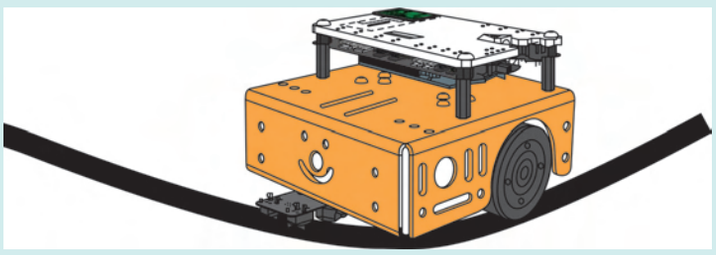


图 1.17　机器人小车自动寻迹

【交流评价】

1.机器人小车在直线上移动时，是否存在左右摆动的情况?

2.分析该机器人小车的系统组成，看看能否找出对应的零件材料。

第二节　简易机器人的设计

学习目标

1. 掌握设计、制作简易机器人的步骤。
2. 学会根据设计任务制订设计方案。

我们见过了各式各样的机器人，也亲手搭建了自己的机器人小车。这些机器人的功能和结构并不是凭空想象出来的，必须遵循一定的设计步骤和方法。

要制作自己的简易机器人，我们该如何设计呢?

在《技术与设计 1》中，我们已经对工程设计的基本过程有了认识，机器人的设计也遵循这一过程。图 1.18 给出了简易机器人设计与制作的流程。

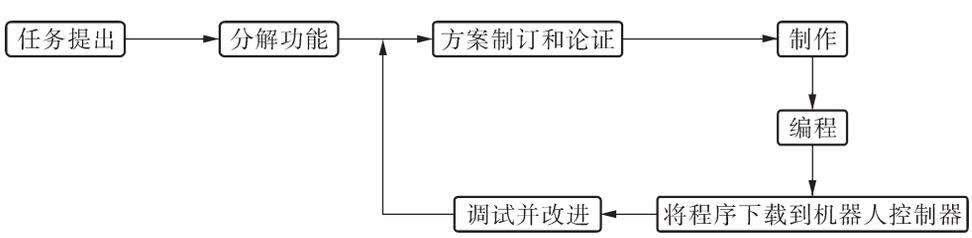


图 1.18　简易机器人设计与制作的流程

一、设计任务的提出

该项目模拟在车间内沿标记好的线路自动搬运货物的机器人。

【任务】

制作一台带机械手爪的移动机器人，它能够沿地面上的黑色引导线运动，并利用传感器自动检测放在外圈正六边形边上的货物（用纸杯代替），当检测到货物后，能够利用机械手爪将货物抓取离地并运送到仓库。最左边和最右边的黑线端点为机器人的出发区，一次只摆放一个货物，比赛开始时随机决定机器人出发区及货物摆放的位置，然后选手现场编写程序，机器人完成任务后结束计时。以用时长短决定选手的名次。

图 1.19　机器人巡线搬运货物的场地

【场地要求】

图 1.19 是机器人巡线搬运货物的场地示意图，场地尺寸为 1 200 mm × 2 000 mm，两个环线为标准的六边形，边长分别为 600 mm 和 300 mm，黑色引导线宽度为 10 mm。可以使用黑色绝缘胶带粘出比赛用场地。

【比宽规则】

（1）机器人一旦启动便不允许人为干预，即机器人必须由程序控制，而非人工控制。

（2）参赛选手必须现场从零开始编写程序，不允许在提前编写好的程序上进行更改。

（3）如果在作业中机器人偏离引导黑线，需将机器人取回至出发区重试，但是计时不中断。按照任务完成时间评估成绩。

二、总体设计方案的制订

设计任务明确之后，下面对巡线搬运机器人进行功能分解，并制订设计方案。表 1.3 归纳了该机器人的功能需求、解决方法、设计细节及具体方案。

表 1.3　巡线搬运机器人总体设计方案

功能需求	解决方法	设计细节	具体方案
自动巡线	可移动载体	移动平台类型：轮式、履带或步行 载体移动速度 载体转弯方式	轮式底盘三轮布局，双后轮驱动，电动机调速，双轮差速运动调整行进方向
	红外线传感器	个数、配置、灵敏度	用两个红外线传感器检测机器人相对黑线的位置，进而调节机器人的前进方向
识别纸杯	避障传感器	传感器类型：超声波、红外、接触式	用一个超声波传感器检测前方障碍物方位，采取相应的识别策略

续表

功能需求	解决方法	设计细节	具体方案
搬运纸杯	机械臂	驱动方式、结构类型	舵机驱动、串联结构，用平行四杆机构实现末端始终平行于地面
	机械手爪	驱动方式、抓取位置、对准纸杯方式	舵机驱动、抓住纸杯腰部、开口适当大一些
识别交叉路口	红外线传感器	个数、配置、灵敏度	用 2 个红外线传感器专门识别交叉路口

在后面的章节中，我们将对表 1.3 的总体设计方案做出详细的解说，论证方案的合理性和可行性。当然这只是一种参考方案，同学们可以发挥自己的聪明才智设计出其他的方案。

第三节　机器人常用的机械传动机构

学习目标

1. 了解机器人机械结构的组成。
2. 了解齿轮传动机构的类型、特点与应用。
3. 了解常见连杆传动机构的结构及其应用。
4. 动手组装基于连杆机构的多足机器人。

机器人的机械结构主要包括机体、传动机构和执行装置。图 1.20 所示的机械臂机器人，底部的基座是它的机体，中间所有杆状构件组成传动机构，末端的手爪属于执行装置。

传动机构在机器人机械结构中具有重要作用，它负责将驱动装置的动力和运动传递给执行装置。在机械中，常见的传动形式包括齿轮传动、链传动、连杆传动、带传动等。下面我们给同学们介绍机器人最常用的齿轮传动机构和连杆传动机构。

图 1.20　机械臂的机械结构

一、齿轮传动机构

（一）齿轮传动的概念

在机器人的机械系统中，齿轮传动的应用最广泛。齿轮是一种边缘有齿的轮子。齿轮传动是将一对齿轮安装在轴上，依靠均匀分布在圆周上的很多齿牙逐对啮合来传递动

力的传动方式。利用一系列齿轮之间的相互啮合组成变速箱，可以改变传送力的大小及方向。

齿轮传动的特点是紧凑、效率高、寿命长、传动比准确；但制造精度和安装精度要求较高，且不适于远距离传动。

（二）齿轮传动的类型

常见的齿轮传动类型如图1.21所示。

（1）直齿轮传动：最常见的齿轮传动形式。直齿轮呈圆柱状，轮齿均匀地分布于外圆柱面，方向平行于轴线。

（2）斜齿轮传动：齿轮也呈圆柱状，但轮齿方向与轴线成一定角度。与直齿轮传动相比，斜齿轮传动更平稳，可以传递更大的载荷并且可以承受更高的转速。斜齿轮传动广泛应用于汽车。

（3）锥齿轮传动：外形呈圆锥状的齿轮。它用来传递相交轴之间的运动。这种传动类型可以改变运动传递的方向。

（4）蜗轮蜗杆传动：蜗杆的外形像螺栓，一般作为主动件。从动件除了蜗轮外，还可用斜齿轮甚至直齿轮替代。蜗轮蜗杆用于交错轴间的传动，能获得很高的减速比。转动蜗杆，可轻巧地驱使蜗轮转动；反过来，转动蜗轮几乎无法驱使蜗杆转动（即具有自锁性）。

（5）齿轮齿条传动：齿条上所有的齿都排成直线。齿轮和齿条的运动形式分别为转动和平动。它可以将旋转运动转换为直线运动，或者互换。

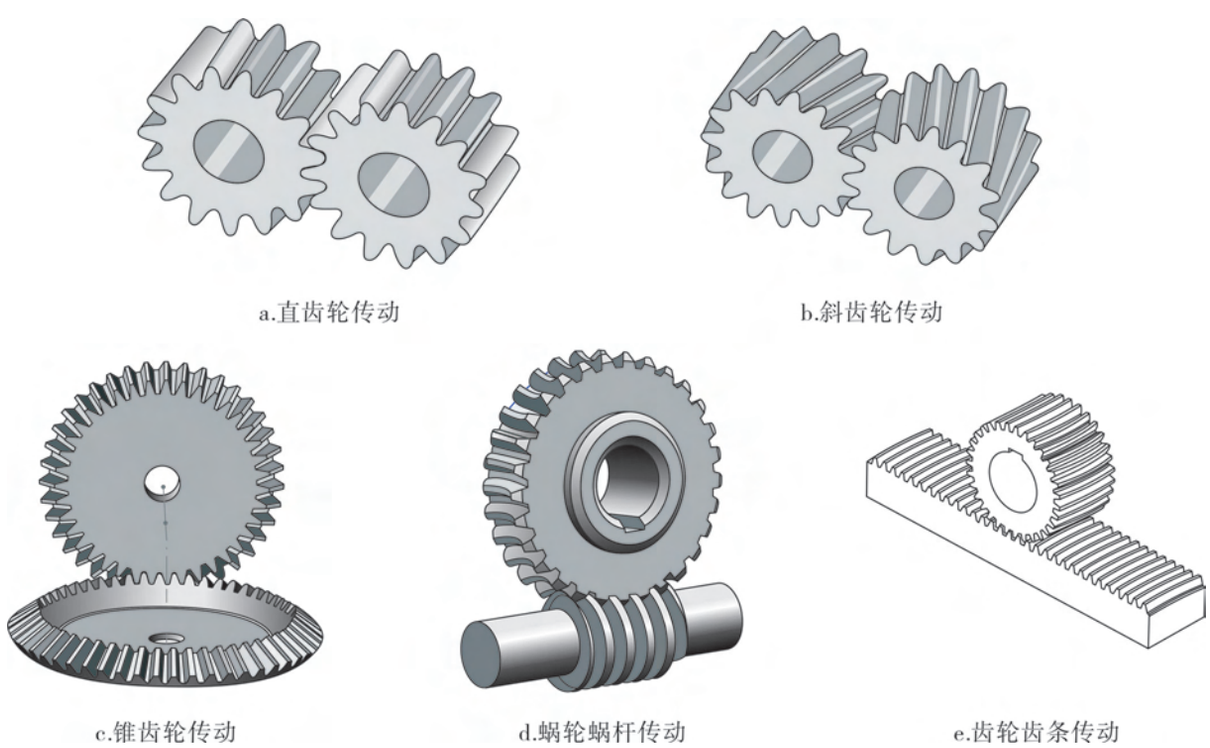


图 1.21　齿轮传动的类型

探究与交流

按图 1.22 所示搭建一套简易的齿轮传动装置，用手转动其中一个齿轮（被称作主动轮），观察另一个齿轮（被称作从动轮）的运动情况。

1．大齿轮齿数_____，小齿轮齿数_____。

2．当小齿轮顺时针转动时，大齿轮转动方向为_____。当大齿轮顺时针转动时，小齿轮转动方向为_____。

3．当大齿轮转一圈时，小齿轮转了_____圈。说明大齿轮转速比小齿轮_____。

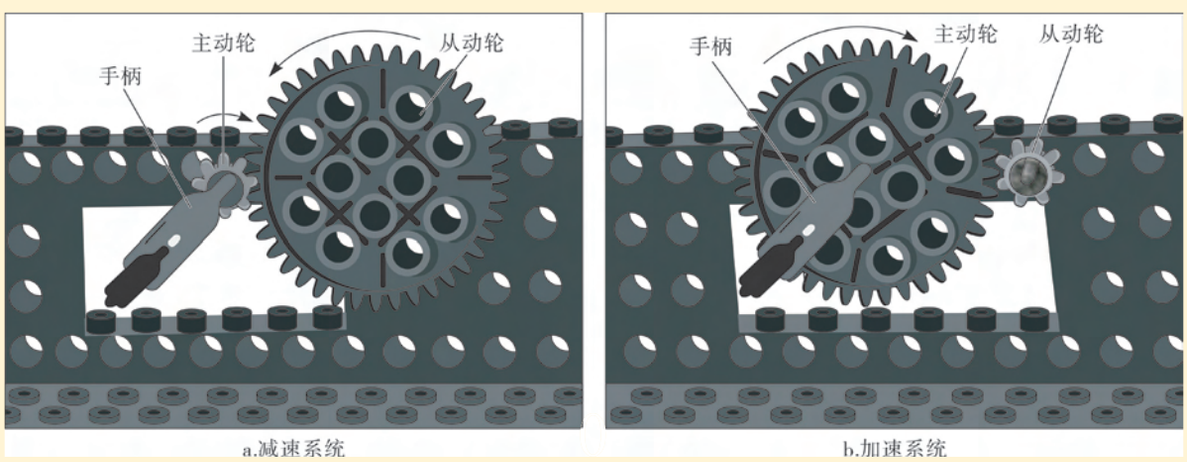


图 1.22　简易的齿轮传动装置

（三）传动比

一对相互啮合齿轮的角速度之比称为传动比。很容易想明白，主动轮转过一个齿，从动轮也必须转过一个齿，所以一对齿轮的转速与齿数成反比，即：

$$\omega_{主}：\omega_{从}=Z_2（从动轮齿数）：Z_1（主动轮齿数）$$

由此可知，利用不同齿数的齿轮相啮合，能够得到不同的传动比。若一个减速系统的传动比为 5∶1，则当小齿轮（主动轮）转过 5 圈时，大齿轮（从动轮）才转过 1 圈。

（四）轮系

在实际机械中，采用单对齿轮传动往往是不够的，为了满足不同的用途，需要采用多对齿轮。例如，在钟表里为保持时针、分针、秒针之间转速的定比关系，必须多对齿轮配合使用（图 1.23）；汽车后桥的差速齿轮机构（图 1.24），在弯道上将发动机传来的一种转速分解为两个后轮的不同转速，以适应不同的转弯曲率等。这种由一系列齿轮组成的齿轮机构统称为轮系。

图 1.23　钟表里的齿轮

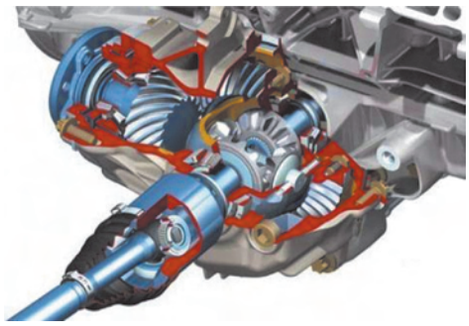


图 1.24　汽车的差速齿轮机构

轮系运转时，如果各齿轮轴线位置固定，则称之为定轴轮系。定轴轮系的传动比为系统运动输入齿轮转速与运动输出转速的比值，即：

$\omega_{主}$：$\omega_{从}$=各对齿轮从动轮齿数的乘积：各对齿轮主动轮齿数的乘积

轮系的作用和特点包括：

（1）能获得较大的传动比。一对啮合齿轮，考虑到尺寸的合理性，大小齿轮直径不宜相差太大。如果要求实现大传动比的传动，则需要采用轮系。

（2）实现变速和换向传动。

（3）实现相距较远的两轴之间的传动。

二、连杆传动机构

连杆机构是由若干个杆状或块状构件通过转动关节（也称铰链）或移动关节（也称滑块）连接而成的。所有构件在同一平面或相互平行的平面内运动的机构称为平面连杆机构，否则为空间连杆机构。

雨伞的骨架一般由若干条支路组成，每一支路都是一个简单的平面连杆机构，其原理可简化成图 1.25。图 1.20 所示的机器人机械臂则是一个空间连杆机构，其简图为图 1.26。

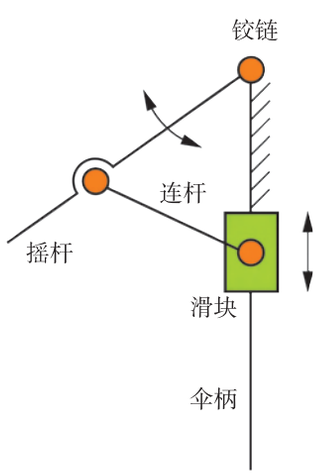


图 1.25　雨伞的收放机构

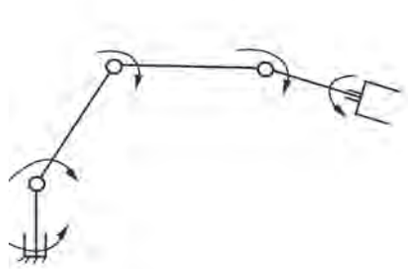


图 1.26　机械臂机构简图

最简单的连杆机构为平面四连杆机构，即用铰链将四个杆件连接而成。比较典型的平面四连杆机构有曲柄摇杆机构、曲柄滑块机构、双曲柄机构。

（1）曲柄摇杆机构：两个连架杆中一个为曲柄，另一个为摇杆的平面四连杆机构（图1.27）。与机架相连的杆件称为连架杆，能绕定轴做整周回转的连架杆称为曲柄，绕定轴做往复摆动的连架杆称为摇杆或摆杆，不与机架相连的杆件称为连杆。图1.28所示的雷达天线的俯仰机构就是一个典型的曲柄摇杆机构。

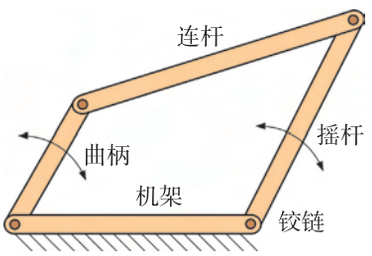


图1.27　曲柄摇杆机构

图1.28　雷达天线

（2）曲柄滑块机构：用曲柄和滑块来实现转动和移动相互转换的平面连杆机构（图1.29）。图1.30所示的内燃机是一个典型的曲柄滑块机构，内燃机的活塞就是滑块，内燃机的曲轴就是曲柄。燃烧气体膨胀，推动活塞再经连杆带动曲轴整周转动，结果滑块的直线输入运动被变换为曲柄的旋转输出运动。反过来，在曲柄滑块机构中，以曲柄为运动输入件，滑块为运动输出件，则曲柄的旋转运动变换为直线运动。

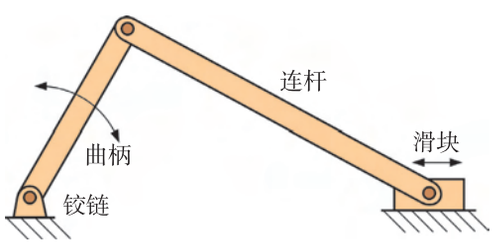


图1.29　曲柄滑块机构

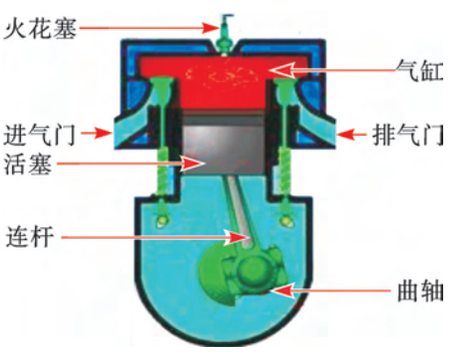


图1.30　内燃机

（3）双曲柄机构：若平面四连杆机构中与机架相连的两个杆件均能做整周转动，则称为双曲柄机构（图1.31）。平行四连杆机构的两组相对的杆件长度彼此相等，它是双曲柄机构中的一个特例，在日常生活中最为常见。图1.32所示的火车车轮联动机构即是双曲柄机构。

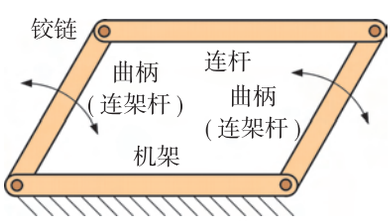


图1.31　双曲柄机构

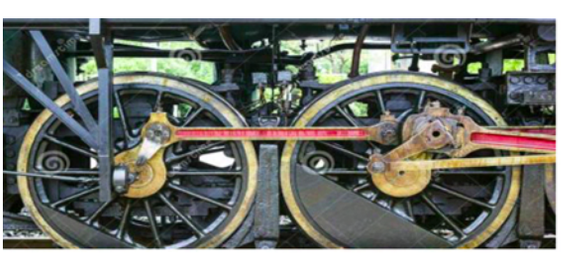


图1.32　火车车轮联动机构

实践与体验

多足机器人的组装

【目的】体验连杆机构在机器人的应用。

【任务】按照图 1.33 所示，将之前组装好的简易机器人小车去掉巡线传感器并安装上连杆，使之变成多足机器人。

【器材】组装好的机器人小车，连杆。

【场地】普通平整地面或者桌面。

参考组装步骤：按图 1.33 所示的零件和步骤组装多足机器人的腿部结构。

安装好的多足机器人如图 1.34 所示。

【交流与评价】

1. 完成的多足机器人的腿部是哪一种连杆传动机构？

2. 完成的多足机器人在运动上存在哪些问题？应该如何改进？

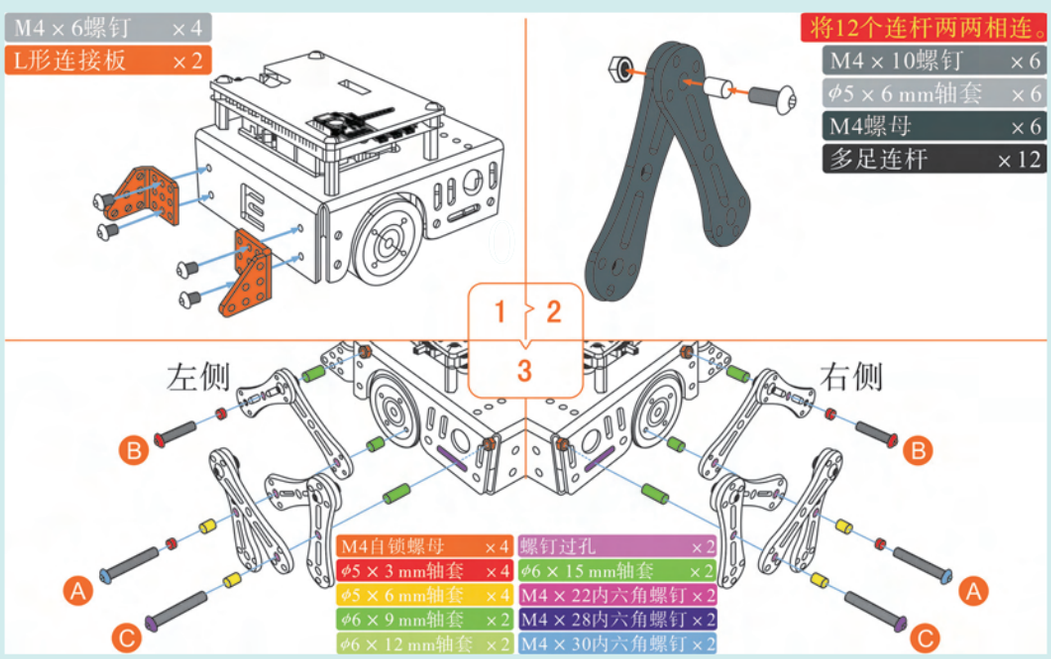


图 1.33　多足机器人组装步骤

图 1.34　组装完成的多足机器人

第四节　巡线搬运机器人机械结构的设计与制作

学习目标

1. 了解机器人移动载体的结构形式。
2. 能根据任务设计机器人的机械结构。
3. 动手组织基于连杆机构的多足机器人。

机器人的功能不同，其机械结构的形式也会有区别，因此，如何根据具体任务设计制作不同类型的简易机器人需要动一番脑筋。第二节提出了简易机器人的制作任务：巡线搬运机器人。在总体设计阶段，提出了以三轮小车为移动载体，由可抬起和放下的搬运机械手爪实现抓纸杯、运纸杯、放纸杯等功能。移动机构是对人类行走功能的模拟和扩展，抓纸杯机械手爪是对人类手臂动作和功能的模拟和扩展。在这一节，我们将讲解如何设计和制作满足操作要求的搬运纸杯机器人的机械结构。

一、移动载体的设计与制作

1. 移动载体的设计

机器人移动机构主要有三种形式：轮式、履带式、足式。

（1）轮式移动机构是最常见的机器人移动机构。常见的轮式移动机构有三轮和四轮两种。图1.35为三轮式移动机构的两种形式。第一节制作的移动小车（图1.17）正是运用了图1.35b所示的移动机构，即两个驱动轮和一个辅助轮，辅助轮起到支撑和导向的作用。图1.36为四轮式移动机构的两种形式，图1.36a的特点是可以原地转弯，图1.36b就是传统的汽车后桥驱动方式。

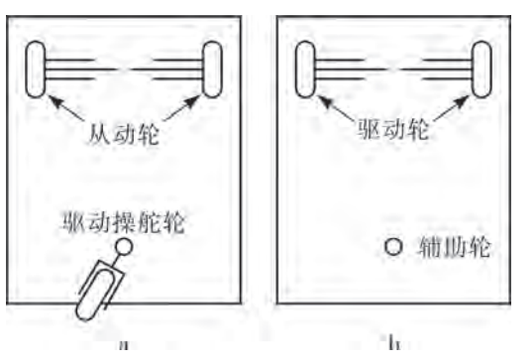


图 1.35　三轮式移动机构

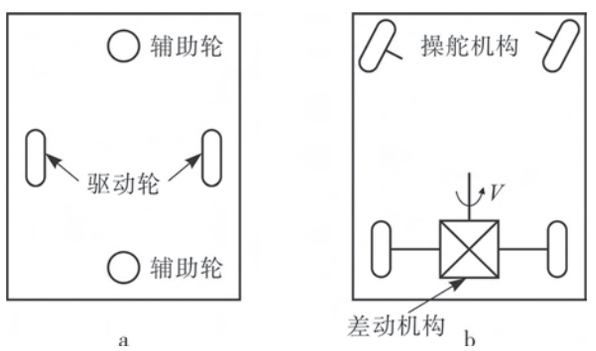


图 1.36　四轮式移动机构

（2）履带式移动机构的突出特征是将履带卷绕在多个车轮上，车轮不直接与路面接触，相当于一种为自己铺路的轮式移动机构。履带可以缓冲路面状态，因此可以在更苛刻的路面上行驶。最常见运用履带式移动机构的有坦克、挖掘机等。

（3）足式行走机构从外表上看更像人或动物，足式行走机器人也就是所谓的步行机器人。足式的特长是跨越，因此能在凹凸不平的地上行走，跨沟越豁，上下台阶等，具有较广的适应性。但是足式机器人的控制和稳定有难度。常见的足式移动机构有两足式、四足式、六足式等。第三节组装完成的多足机器人就是足式机器人，采用了曲柄摇杆机构，轮子的连续旋转运动带动六条腿有序地往复摆动，从而实现多足机器人的运动。机器人的轮子是其中的曲柄，与地面接触的前后四条腿是其中的摇杆，中间的两条腿是其中的连杆。

探究与交流

轮式、履带式和足式三种移动机构各有特点。试从结构复杂度、地面适应性、稳定性、可控性几个方面对三种移动方式进行比较，并完成表 1.4。

表 1.4　三种移动方式对比

移动机构形式		结构复杂度	地面适应性	稳定性	可控性
轮式	三轮				
	四轮				
履带式					
足式	两足				
	多足				

本设计任务要求机器人能够在平整的地面完成前进、后退、转弯等动作。通过前面的分析，我们知道了轮式结构是最简单的移动结构，也比较适合在平整地面上运动，因此是本设计任务移动载体结构形式的首选。1.35b 的结构又是轮式结构中最简单实用的一种，因此可以作为本设计任务的最终方案。

2. 移动载体的制作

我们在第一节完成搭建的轮式机器人，可以作为本设计任务的移动载体。

二、搬运机械手的设计与制作

人类的双手就是一个既灵巧又复杂的执行机构，可以完成抓、夹、握、拨、捏等操作，以至于目前还没有研制出完美的替代品。在工业自动化生产线中，机械手是一种常见的执行机构，用于将物体从甲地点搬到乙地点。机械手由机械手爪和机械臂两部分组成，下面分别进行介绍。

1. 搬运机械手爪的设计

机械手爪主要分为刚性机械手爪（图 1.37）和软体机械手爪（图 1.38）。

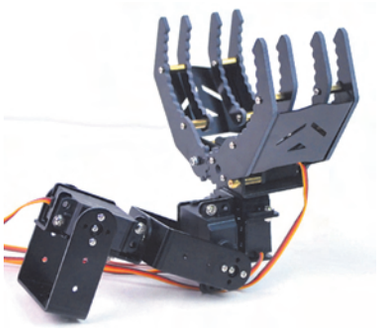


图 1.37　刚性机械手爪

图 1.38　软体机械手爪

刚性机械手爪是目前应用最普遍的机械手爪，已经有很多种成熟的结构形式，有的用减速电动机直接带动关节运动，有的使用舵机直接带动关节运动，有的则利用减速电动机带动钢丝、同步带、连杆来驱动关节运动。

软体机械手爪是最近几年新兴的机械手爪，采用软质材料制造成中空的气腔，通过往气腔里通气或抽气来实现手爪的张合。与传统的刚性机械手爪相比，它操作细小或易碎物体的能力更强。例如对不规则形状的易碎物体 (水果、蔬菜和生物组织等) 的分拣具有更高的适应性。

本设计任务要求机器人能够抓取纸杯。因为纸杯形状比较简单，体积也不大，重量也轻，因此使用舵机带动的刚性机械手爪就可以满足抓取纸杯的需求。

2. 搬运机械臂的设计

机械臂从机构学的角度可以分为串联机械臂（图 1.39）和并联机械臂（图 1.40）。

探究与交流

刚性机械手爪与软体机械手爪各有特点。试从结构复杂度、抓取适应性、结构稳定性、控制系统复杂度等几个方面进行比较，并完成表 1.5。

表 1.5　两种机械手爪对比

抓取机构形式	结构复杂度	抓取适应性	结构稳定性	控制系统复杂度
刚性机械手爪				
软体机械手爪				

图 1.39　串联机械臂

图 1.40　并联机械臂

串联机械臂是由一系列连杆通过转动关节或移动关节串联而成的，是一种开链机构，采用驱动器驱动各个关节的运动从而带动连杆的相对运动，使末端执行器达到合适的位置。串联机械臂也叫作关节机械臂，是当今工业领域中最常见的工业机器人的形态之一，适用于诸多工业领域的机械自动化作业。

并联机械臂一般由上下运动平台和两条或者两条以上运动支链构成，运动平台和运动支链之间以并联的方式构成闭链机构。并联机械臂具有结构紧凑、刚度高、速度快、承载能力大的特点，多用于需要高刚度、高精度、高速度、空间狭小的场合。

本设计任务要求机器人能够将纸杯抬高和放下，对运动速度和运动精度并没有提出要求，因此可以选择比较简单的串联机械臂的形式。采用一个舵机驱动关节运动，从而带动连杆的相对运动，就可以实现末端机械手爪的抬高和放下的功能需求。为了让末端机械手爪始终保持水平，我们可以引入平行四杆机构。

探究与交流

串联机械臂与并联机械臂各有特点。试从结构复杂度、结构紧凑度、结构稳定性、运动速度、控制精度、控制系统复杂度等几个方面进行比较，并完成表 1.6。

表 1.6　两种机械臂对比

机械臂分类	结构复杂度	结构紧凑度	结构稳定性	运动速度	控制精度	控制系统复杂度
串联机械臂						
并联机械臂						

3. 搬运机械手整体设计

分析完机械手爪跟机械臂的设计要点后，我们根据搬运机器人的具体要求，将机械手的设计问题一一细化（表 1.7）。

表 1.7　搬运机械手设计方案的讨论

功能需求	考虑因素	技术细节	具体方案
搬运纸杯	如何对准纸杯位置	机械手爪的安装位置要适合抓取运动方向上的纸杯；当运动方向有细微偏差的时候，要能够自适应地抓住纸杯	让手爪朝向机器人的运动方向，使机器人沿着黑线前进就能够抓到纸杯；增大手爪的开口大小，使手爪能更容易地包围住杯子
	如何防止纸杯变形	将手爪抓持面做成跟杯子侧壁吻合的形状；在手爪抓持面增加柔性缓冲材料	让手爪合上的时候，中间的抓持面正好对应高度纸杯腰身的圆环；手爪抓持面使用橡胶套
	如何防止纸杯掉落	增加手爪抓持面的材料摩擦系数；抬高或放下的时候要能够平行移动纸杯	手爪抓持面使用橡胶套；使用平行四杆机构实现手爪端面始终平行地面运动

续表

功能需求	考虑因素	技术细节	具体方案
搬运纸杯	高度要求	在抓取纸杯的时候让手爪处于合适的高度，太高了夹不到，太低了夹不住	手爪放下的时候高度为 40~60 mm
		在抬起纸杯的时候不要让纸杯遮挡住传感器，防止传感器误触发	手爪抬起的时候高度不低于 140 mm
	速度要求	抬起和放下的速度不要太快，否则容易让纸杯掉落	使用舵机带动平行四杆机构来实现手爪的升降

事实上，除了上面给出的方案，还有其他的解决途径。同学们可以随着自己经验的积累与解决问题能力的提高，不断寻找新的方式来创造性地设计出新的机械结构以满足任务需求，也可以对上述方案提出改进的意见。

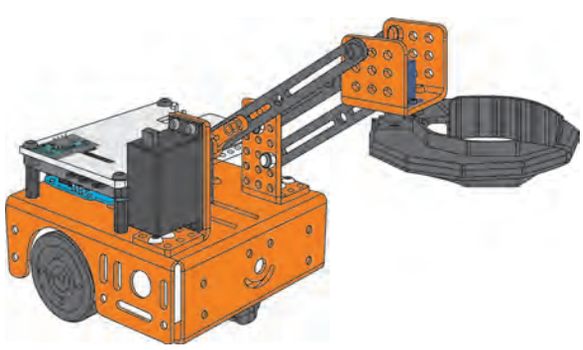


图 1.41　搬运机器人

图1.41给出了搬运机器人的具体机械结构，它是根据表1.5的分析结果，在第一节完成搭建的轮式机器人的基础上扩展了机械手爪的作品。它使用大舵机带动平行四杆机构运动，实现对纸杯的抬高和放下；使用小舵机带动手爪开合，实现对纸杯的抓取和放开。

4. 搬运机械手爪的制作

上面给出的搬运机器人的机械手爪动组装步骤如下：

第一步：按图1.42所列的零件和安装步骤组装机械手的舵机支架及大舵机。

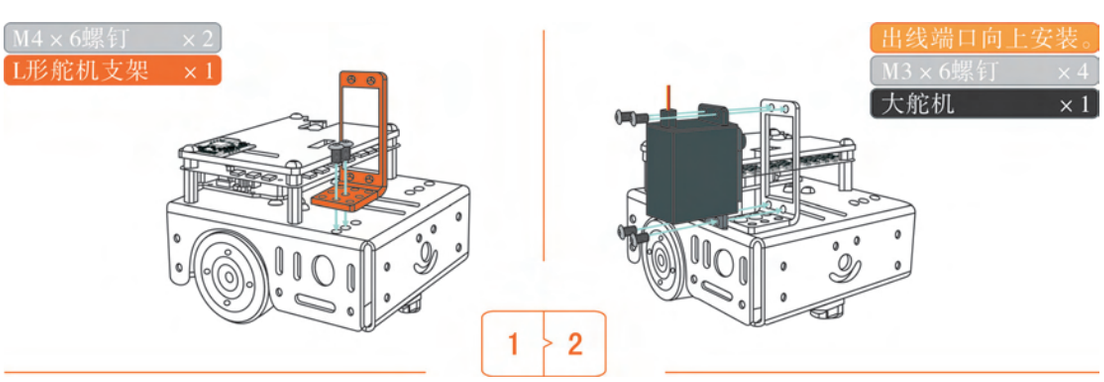


图 1.42　舵机支架及大舵机的安装

第二步：按图1.43所列的零件和安装步骤组装机械手爪的平行四杆机构。

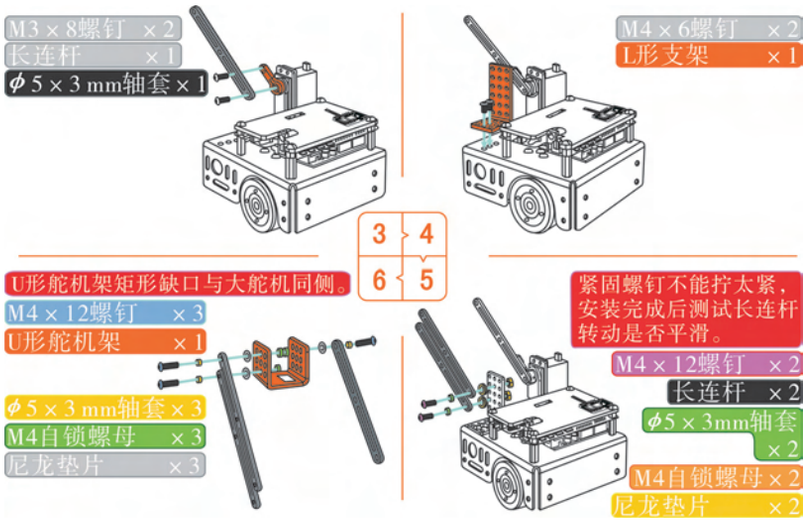


图 1.43　机械手的平行四杆机构的安装

第三步：按图1.44所列的零件和安装步骤组装机械手的机械手爪。至此，搬运机械手制作完成。

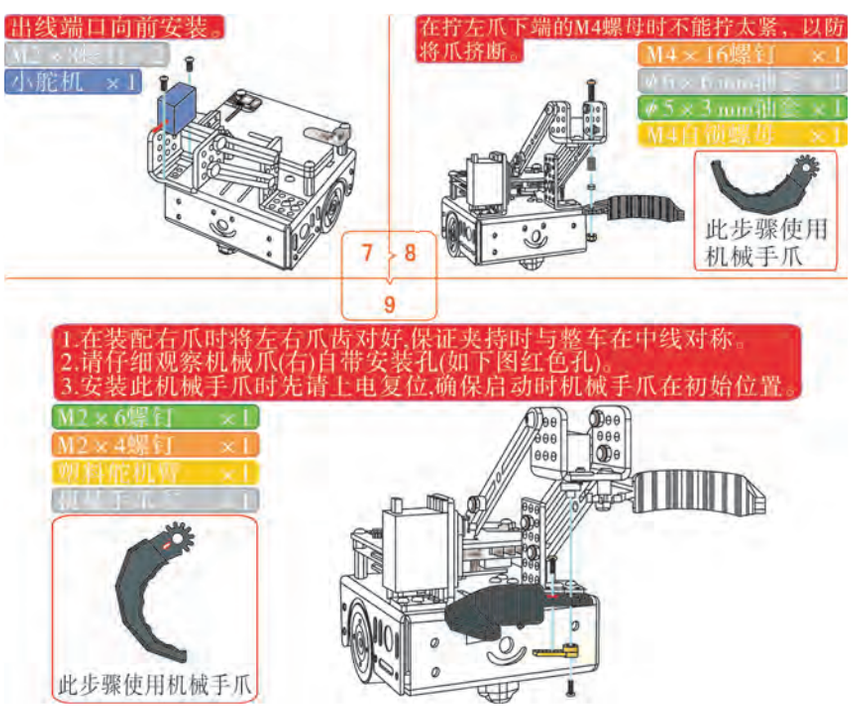


图 1.44　机械手爪的安装

探究与交流

我们总共用了两个舵机来完成搬运机械手的制作，你能准确描述大舵机是怎么将自己的旋转运动转变成为机械臂的抬起和放下运动的吗？小舵机又是如何将自己的旋转运动转变成为机械手爪的张开与闭合呢？它们都用到了我们前面学过的哪些机构？

本章小结

随着科学技术的不断进步，机器人的发展日新月异，并在越来越多的领域中为人类服务，进一步把人类从繁重的、危险的体力劳动中解放出来。根据不同的作业要求，机器人的结构形式也各异，但它们都包括了机械结构、驱动装置、传感器、控制器和动力源五个部分。

机器人的机械结构包括机体、传动机构和执行装置。传动机构既可以把单一运动转化成多种运动，又可以改变输出力及力矩的大小和方向。常见的机械传动形式有齿轮传动、链传动、连杆传动、带传动等，本章主要介绍了机器人上最常用的齿轮传动和连杆传动的特点及其应用。

机器人也遵从一般装置或系统的设计制作流程，首先需要根据任务和功能制订相应的设计方案，然后制作机器人本体、编写控制程序，接着开展作业测试，再根据测试结果进行改进。本章给出了一个简易机器人制作项目——巡线搬运机器人，通过本课程的学习，同学们将逐步完成这个项目，最终制作出自己的机器人。

学习评价

评价内容			评价方式		
			自我评价	小组评价	教师评价
过程评价	师生互动	能积极思考老师提出的问题			
		能基于已有经验构建新的知识			
		能积极参与课堂讨论			
	实践活动	能积极参与实践活动			
		与小组成员有效合作			
		简易机器人小车的拼装及调试			
		简易齿轮传动装置的搭建			
		多足机器人的组装			
结果评价	目标实现	了解机器人的发展历史及主要应用			
		知道机器人的基本组成			
		知道机器人机械结构的组成			
		了解巡线搬运机器人的整体设计方案			
		能根据任务要求设计巡线搬运机器人的机械结构			
	收获反思	收获与感悟			
		反思不足			

第二章
机器人控制器

导　言

人类的一切行为由大脑来支配，机器人也是如此。机器人的大脑就是它的控制器，计算机最适合这个角色。

在结构、功能、技术的复杂程度等方面互有区别的机器人，其控制器也不相同。先进的、动作复杂的机器人，通常采用功能强大的PC 机(个人计算机) 或专用运动控制器；而对于简易机器人，使用结构小巧、价格低廉的单片机控制器就完全可以了。下面就让我们从单片机入手，了解简易机器人的控制器吧。

思维导图

第一节　单片机的基本构成与工作过程

学习目标

1. 了解单片机的基本结构。
2. 能够描述单片机的工作过程。
3. 了解单片机的最小系统。

单片机就是集计算机基本功能于一体的集成电路（IC）封装芯片，也称为微型计算机或微控制器（microcontroller）。单片机具有体积小、集成度高、运行可靠、功耗低、易于开发、价格低廉等特点，因而应用非常广泛，在汽车、家用电器、智能玩具、自动售货机、手机等设备中都有它的身影。

单片机的种类繁多，图 2.1 所示是几种常用单片机，虽然它们的外形有所不同，但内部的结构都大同小异，因此熟悉了某一种单片机的基本结构，其他类型的产品可以举一反三，触类旁通。

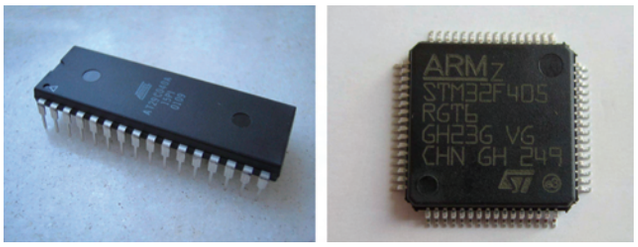


图 2.1　单片机

一、单片机的基本构成

单片机通常由中央处理器（CPU）、存储器、输入 / 输出（I/O）接口三大部分组成，它们通过内部总线连接起来，基本结构如图 2.2 所示。

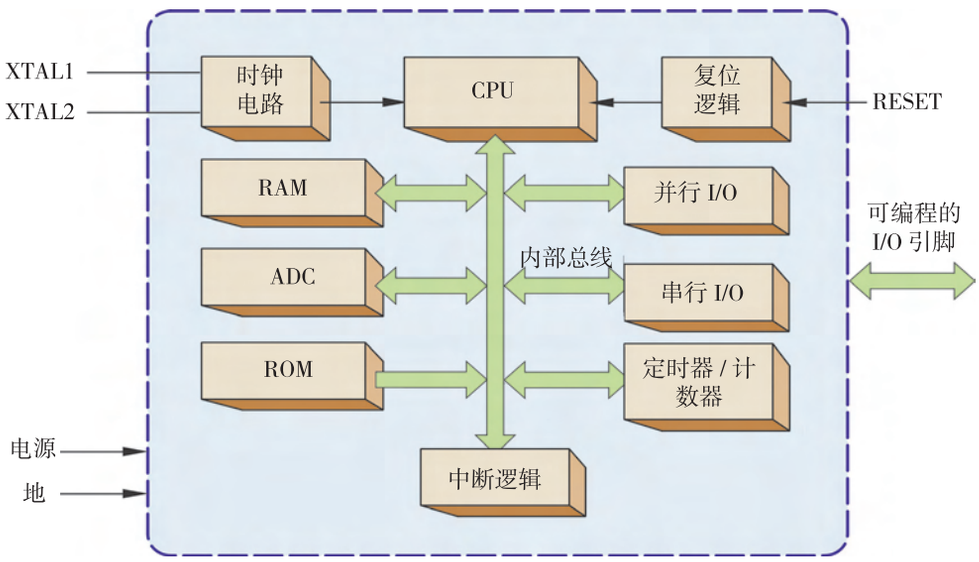


图 2.2　单片机的基本结构

CPU 也就是我们通常所说的微处理器，它负责管理程序的执行以及完成算术和逻辑运算。

存储器用来保存单片机要处理的信息，包括程序和数据。根据存储机制，存储器可分为只读存储器（ROM）、随机存储器（RAM）、可擦除可编程只读存储器（EPROM）、电可擦除可编程存储器（EEPROM）和闪速存储器（flash memory）等。

I/O 接口即输入 / 输出接口，是单片机与外部被控器件连接的桥梁。通常，I/O 接口包括并行 I/O 接口、串行 I/O 接口、特殊 I/O 接口。

单片机除了上述三个主要部分外，还有一些用于控制和监测的电路。例如，时钟电路决定了单片机执行指令的节奏快慢，定时器 / 计数器则使单片机能及时或按一定顺序执行指令。

另外，还有所谓的“中断逻辑”，它好比一个传达室，当控制对象的参数到达某个需要加以干预的状态时，经此传达室通报给 CPU，使 CPU 根据外部事态的轻重缓急采取适当的应对措施。

二、单片机的工作过程

前面介绍了单片机的结构，那么它是如何相互关联成为一个有机整体的？又是如何工作的呢？实际上，单片机内部有一条将它们连接起来的“纽带”，即所谓的“内部总线”。此总线有如城市的“主干道”，而 CPU、ROM、RAM、I/O 接口、中断逻辑等就分布在此“主干道”的两旁，并和它连通。一切指令、数据的传送都经过这条“主干道”。当然，指令和数据在内部总线中的传送并不是混乱无序的，它们是由 CPU 协调指挥的，从而使整个单片机有条不紊地工作。

单片机的内部工作过程是相当复杂的，为了让同学们更好地理解单片机的工作原理，我们来看看 163 × 156 这个算式在单片机中的执行流程。163 × 156 在单片机程序中对应 3 条指令，分别放置在程序存储器的 3 个连续地址中，如图 2.3 所示。首先，程序计数器（PC）指向的指令“LDS　R0，163”，于是单片机将该指令送入指令寄存器中，然后指令译码器翻译这条指令，并执行把操作数 163 送到数据寄存器 R0 中的操作（R0 位于单片机内部 RAM 中）。这条指令执行完后，PC 自动加 1，指向下一条指令“LDS　R1，156”，它的执行过程与前一条指令相同，不过结果是把操作数 156 送到数据寄存器 R1 中（R1 也位于单片机内部 RAM 中）。然后 PC 再自动加 1，指向乘法指令“MUL R0，R1”，经过指令译码器翻译，这条指令把 R0 和 R1 中的内容从内部 RAM 中取出来，在运算器中运算，再把执行结果 25 428 送回 R0 和 R1 中。至此，单片机就完成了整个乘法算式的操作。同学们是否觉得这个过程太烦琐、太慢了呢？其实，单片机正是靠这种单调重复但协调有序的操作保证了整个工作的正确无误。况且，单片机每秒能执行上百万条指令，这几条指令对它而言，用“瞬间”一词都嫌太长了。

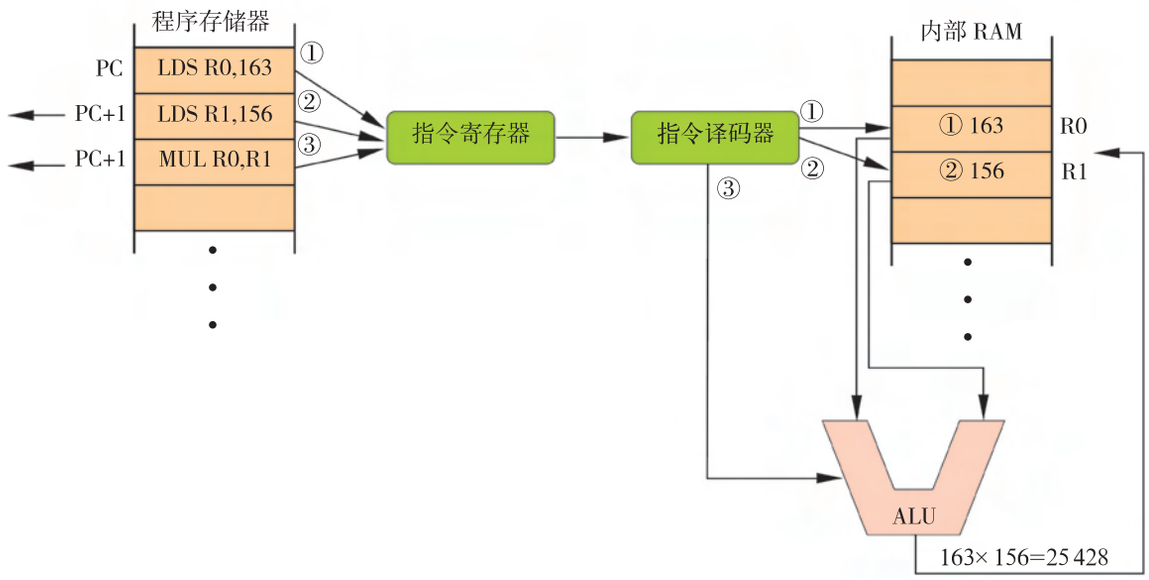


图 2.3　单片机的工作过程

阅读材料

单片机的最小系统

单片机只是一块将多种电路集成到一起的硅片，自身并不具备工作的能力，需要配备外围电路才能运行起来。用最少的元件组成的单片机可以工作的最简单电路叫作最小系统（图 2.4）。对最常见的 51 系列单片机来说，最小系统一般应该包括

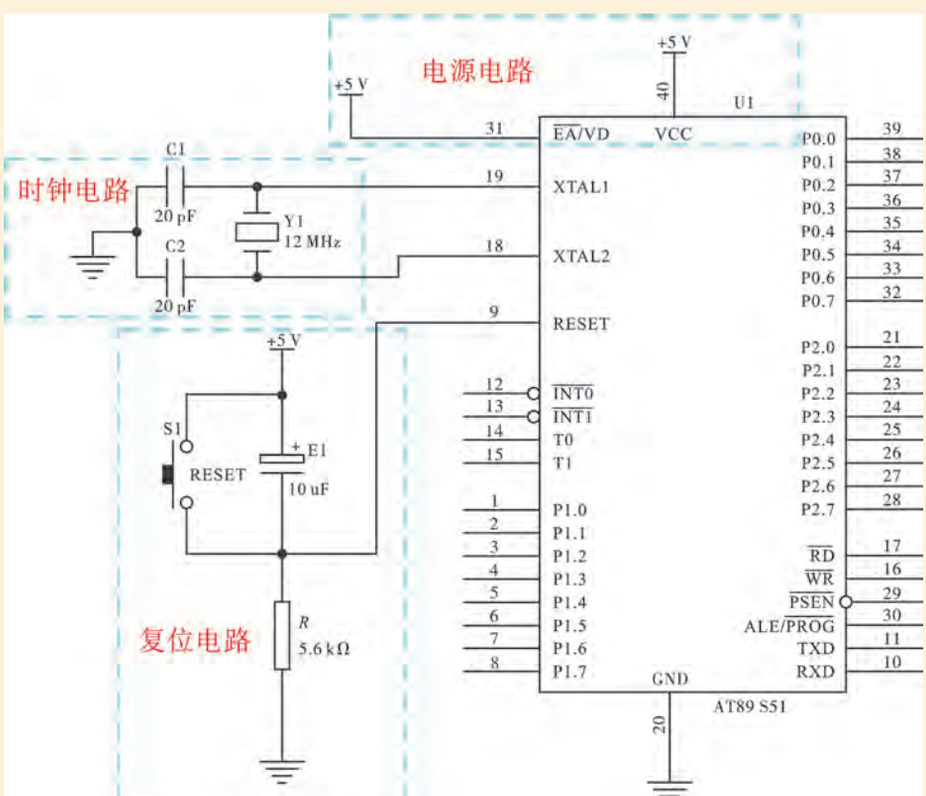


图 2.4　单片机最小系统

电源电路、时钟电路、复位电路等。

（1）电源电路：将输入电压稳定到适合单片机工作的5 V。当单片机接入更多外围设备的时候，还需要电源电路为其他器件提供对应的电压。电源管理是单片机系统中很重要的一个环节。

（2）时钟电路：时钟周期也称为振荡周期，定义为时钟频率的倒数。它是计算机中最基本的、最小的时间单位。在一个时钟周期内，CPU仅完成一个最基本的动作。单片机有内部时钟和外部时钟两种工作方式。利用单片机内部的振荡器，然后在引脚XTAL1和XTAL2两端接电容和晶振，就构成了内部时钟电路。如果接的是外部时钟芯片，则构成外部时钟电路。

（3）复位电路：单片机复位电路类似于计算机的重启功能，当计算机在使用中出现死机，按下重启按钮，计算机内部的程序从头开始执行。单片机也一样，当单片机系统在运行中，受到环境干扰出现程序跑飞的时候，按下复位按钮，内部的程序自动从头开始执行。

探究与交流

请同学们查找自己比较熟悉的家用电器的控制电路资料，看看它是否采用了单片机控制，如果有单片机，它是什么型号的？

第二节　认识 Arduino 控制器

学习目标

1. 了解 Arduino 控制器。
2. 了解 Arduino 控制器开发流程。
3. 了解 Arduino 控制器开发语言。

计算机系统是由硬件和软件两部分组成的，以单片机为核心的简易机器人控制器也是如此。单片机及其外围电路是控制器的硬件，固化在单片机中的程序是控制软件。对于初学者来说，直接使用单片机来设计简易机器人的控制系统会有一定的困难，而 Arduino 控制器很好地解决了这一难题。那么，Arduino 控制器具体是什么呢？

一、Arduino 控制器简介

Arduino 是目前使用比较广泛的开源硬件平台，它包含硬件 (各种型号的 Arduino 板) 和软件 (Arduino IDE 和 Visualino)，使用起来便捷灵活、方便上手。Arduino 控制器的核心是 AVR 单片机，常用的 Arduino 控制器主板有 Arduino Uno（图 2.5）、Arduino Nano、Arduino Mega2560 等。下面对最常见的 Arduino Uno 开发板（简称 Uno 板）进行介绍。

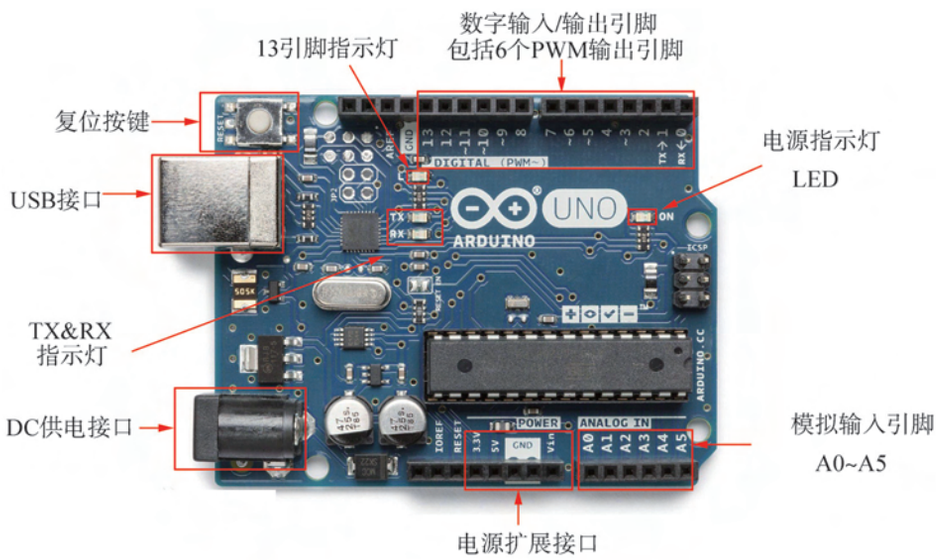


图 2.5　Arduino Uno 开发板

（1）数字引脚：位于 Uno 板顶边的一排插孔，标记为 D0~D13。D 代表数字信号，表示这类引脚的电平只有高（5 V）和低（0 V）两种状态。数字引脚既可输出也可输入，只有数字前带有“~”的引脚才能输出脉冲宽度调制（pulse width modulation，PWM）信号。

（2）模拟引脚：位于 Uno 板右下角的一排插孔，标记为 A0~A5。A 代表模拟信号，表示连续变化的量。模拟引脚只能输入，不能输出。输入信号在 0~5 V 内变化时，对应的整数范围是 0~1 023。

（3）电源扩展接口：位于 Uno 板底边中间的一排插孔，用于给外部器件供电，有 5 V 和 3.3 V 两种电压输出方式。GND 表示接地。

（4）USB 接口：位于 Uno 板左侧，在 PC 机上经过编译、上传的操作后，程序可由该接口上传到单片机中。上传程序前，需要将程序下载线一端连接该接口，另一端连接 PC 机的 USB 口，同时可给 Uno 板供电。

（5）DC 供电接口：除 USB 接口口供电外，还可以通过 DC 供电接口连接外部直流电源来给 Uno 板供电。

本教材就是基于 Uno 板进行简易机器人的开发的，同学们通过本课程的学习，以后可以自行开展更多有趣的设计与制作。

探究与交流

请同学们查阅相关资料，简单介绍一种 Arduino 的创意应用。

二、Arduino 扩展板简介

当我们需要在 Arduino 开发板上连接各种传感器和执行器的时候，可以在面包板上接插元件（图 2.6）。面包板上有很多小插孔，极像面包，因此得名。它用于连接各种电子器件，可以重复使用，可省去焊接过程且易于改变器件的连线，使接线变得更加灵活。

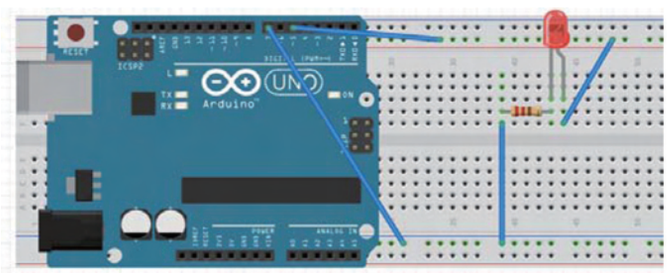


图 2.6　面包板在 Arduino 开发板中的应用

使用面包板进行复杂电路的搭建需要耗费很多时间和精力，而且接线容易松动，导致电路不稳定。而使用扩展板可以一定程度地简化电路搭建过程，更快速地搭建出整个控制系统。Arduino 扩展板是可以与 Arduino 拼接使用的特定功能的电路板，通常具有和 Arduino 开发板一样的引脚位置，可以堆叠按插到 Arduino 上，进而实现特定功能的扩展。如电动机驱动扩展板集成了电动机驱动芯片，可以直接驱动多路电动机和舵机（图 2.7）；网络扩展板集成了网络模块，可以让 Arduino 获得网络通信功能（图 2.8）。

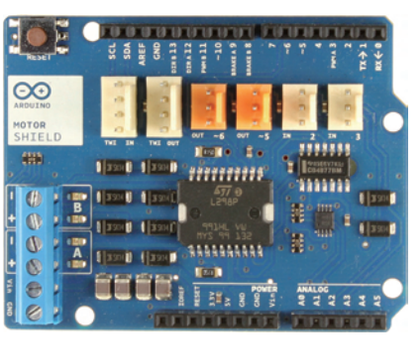


图 2.7　电动机驱动扩展板

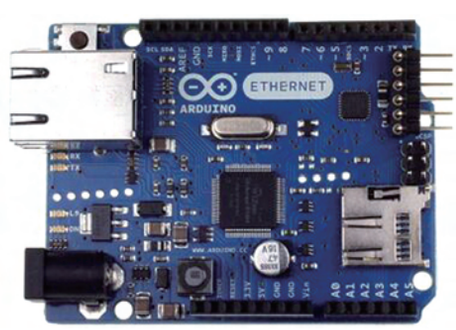


图 2.8　网络扩展板

三、Arduino 控制器的开发流程

开发单片机应用程序时，按照图 2.9 所示的流程进行。

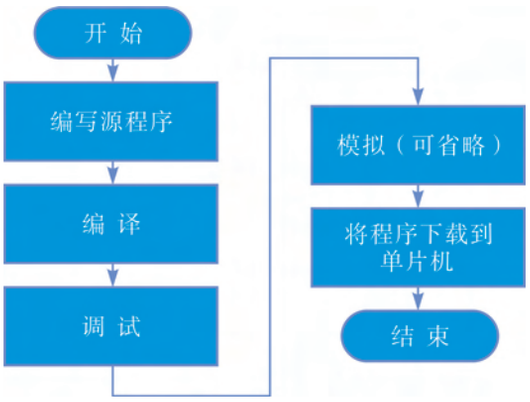


图 2.9　单片机程序开发流程

1. 编写源程序

在 PC 机上可以用图形化语言、C 语言或汇编语言等多种编程语言编写源程序。

2. 编译

编译的任务是检查程序是否存在语法错误，并把源程序转换成目标代码（机器代码）。不同单片机的编译软件互有差别。

3. 调试

利用编程环境中的程序调试功能，用户可以在程序中设置单步、断点、连续三种运行方式，从而可以详细了解程序运行的中间状态，便于找出程序中的错误。

4. 模拟

借助 PC 机，让编好的程序在一定的环境下运行，模仿程序的执行过程，看看机器人的动作是否达到预期的要求。所谓一定的环境指软件模拟环境或硬件模拟环境。如果程序很简单，也可以省略此步骤。

5. 将程序下载到单片机

不同的单片机有不同的下载方式。Arduino 控制器采用一线下载的方式，即在 PC 机上完成编程、编译和模拟等工作后，借助一根程序下载线就能将程序写入控制器。下载线的一端连接到 PC 机的打印机接口或串行通信口，另一端连接控制器的程序下载口。

如果在编译、模拟、下载的操作中出现错误，都应该返回源程序的编写环节进行修改，然后重试，直至顺利通过这个开发过程。

四、Arduino 控制器的开发环境

本教材将涉及两种Arduino开发环境：Arduino IDE（Arduino C语言编程软件）和 Visualino（图形化语言编程软件）。

（一）Arduino IDE

IDE 是 Integrated Development Environment 的英文缩写，也称为集成开发环境，是 Arduino 产品的软件编辑环境，简单地说就是用来编写代码、上传代码的地方。Arduino IDE 可以在 Windows 和 Mac 系统上安装。

1. Arduino IDE 主界面

Arduino IDE 的主界面非常简单明了，下面简单介绍 Arduino IDE 主界面中各部分的作用（图 2.10）。

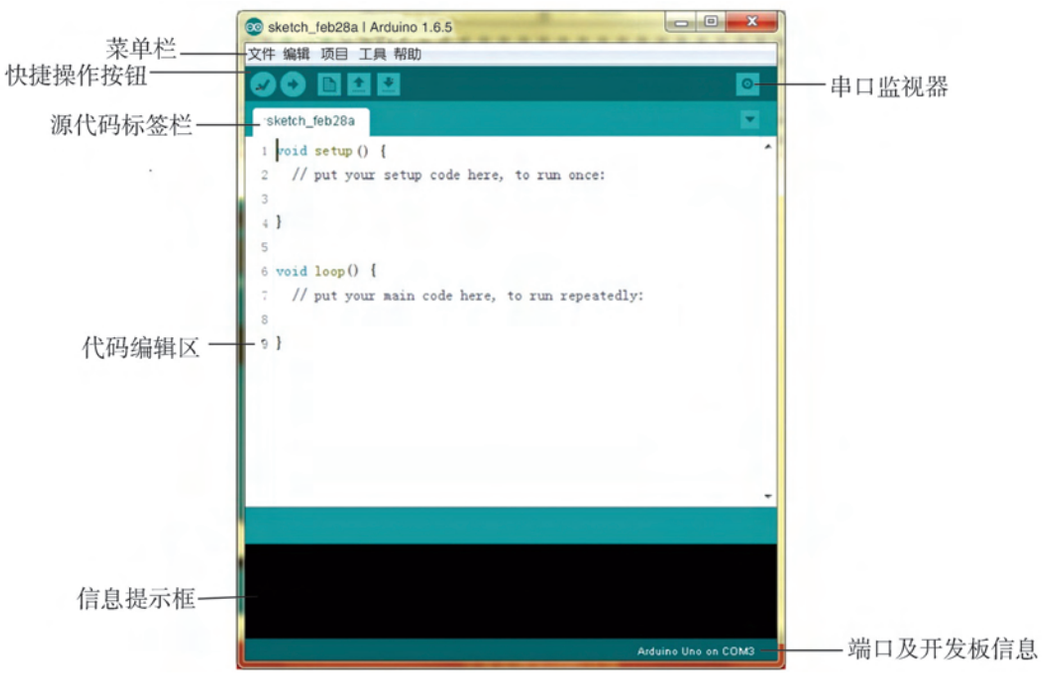


图 2.10　Arduino IDE 主界面

●菜单栏：将各类不同操作聚合到对应的菜单项中，用户可以根据自己想要操作的类别来选择不同的菜单项。

●快捷操作按钮：将用户最常用的操作以按钮的形式呈现出来，快捷操作按钮所实现的功能通过菜单栏同样可以实现。快捷按钮的名称及功能从左到右分别为验证、上传、新建、打开、保存、串口监视器。

●代码编辑区：编写及修改代码的地方。

●信息提示框：提示操作过程信息，例如编译过程和上传过程的操作信息。

●端口及开发板信息：用来提示开发板的类型及位于主机的哪个端口。

2. 运行闪烁 LED 示例程序

Arduino IDE 提供了大量现成的示例来供用户学习，本节将演示一个不需要其他拓展就可以运行的简单程序，既可以帮助我们熟悉如何上传程序，同时也可以测试一下板子的好坏。我们可以通过选择“文件”→“示例”→“01.Basics”→“Blink”选项打开该程序（图 2.11）。Uno 板上有个标注为“L”的发光二极管（LED），这段测试代码就是让这个 LED 循环闪烁的。

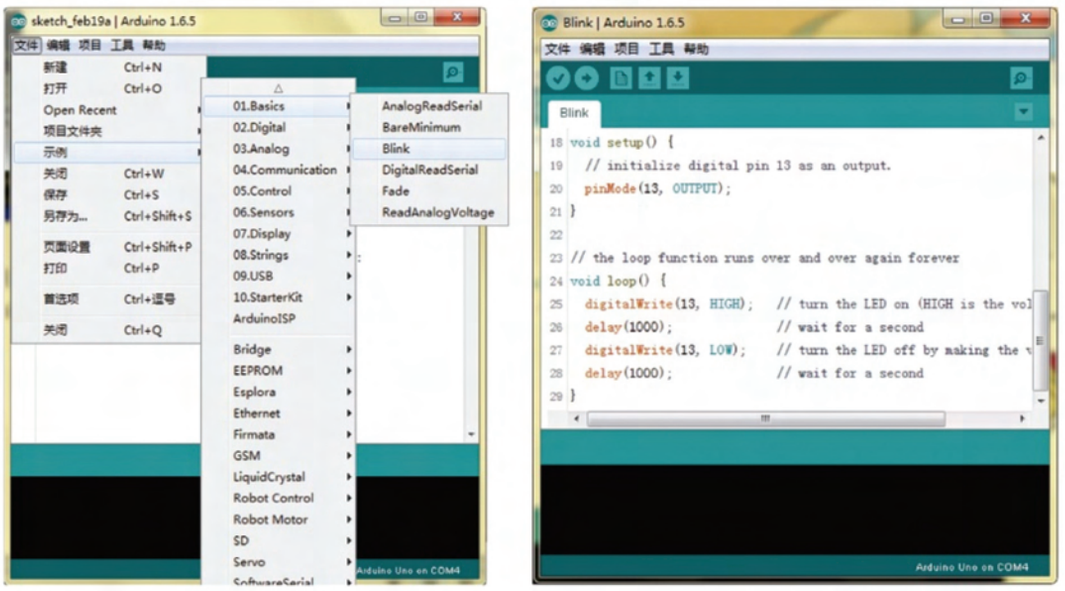


图 2.11　打开 Blink 示例程序

验证程序。通常写完一段代码后，我们都需要验证一下，看看代码有没有错误。单击“验证”，下方绿色进度条显示编译的进度（图 2.12）。

验证完成后，信息提示框显示“编译完成”（图 2.13）。由于是示例代码，所以校验不会有错误，不过在以后写代码的过程中，输入完代码都需要校验一下。

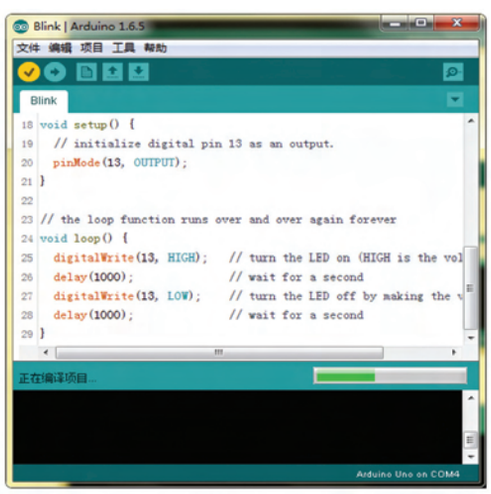


图 2.12　进度条显示验证程序的进度

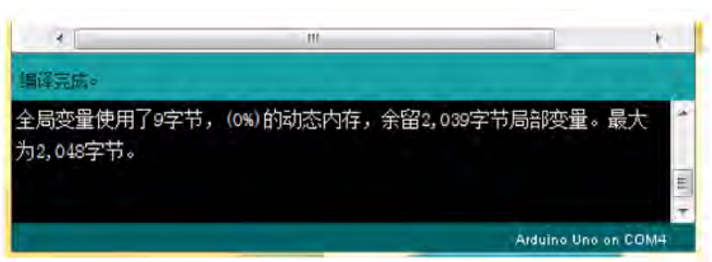


图 2.13　信息提示框显示编译完成

程序验证无误后可以开始将程序上传到开发板中。首先选择对应的开发板（图 2.14），然后选择与“设备管理器”里一致的端口号（图 2.15），再单击“上传”（图 2.16），将程序写入 Uno 板。

图 2.14　选择对应的开发板

图 2.15　选择端口号

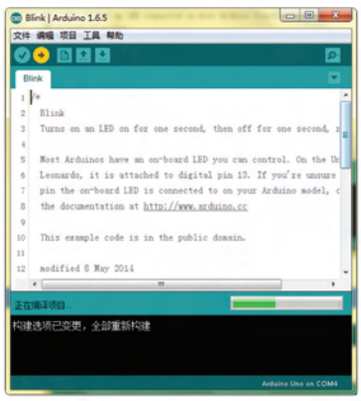


图 2.16　上传程序

信息上传成功后，信息提示框显示“上传成功”（图 2.17），然后你就会看到程序运行的效果：Uno 板上标注为“L”的 LED 在闪烁。

图 2.17　信息提示框显示上传成功

活动延伸

请同学们通过查找相关资料，自行完成 Arduino IDE 开发环境的安装，并简述安装过程及注意事项。

（二）Visualino

Visualino 是一款基于 Arduino 的集成模块化开发环境。作为一款图形化的编程软件，它把许多函数封装成图形模块，我们需要按照一定的规则来使用模块。编程就像拼图一样，我们只需要动动鼠标将图形拖拽到编程区中一块一块地拼好。但是，图形化编程软件只适用于逻辑相对简单的程序编写，如果遇到逻辑复杂一些的程序，Arduino IDE 编程软件可以更好地完成编程任务。

1. 下载并安装 Visualino

Visualino 开发环境安装包可以直接从其官方网站上免费下载。安装好后就可以开始图形化编程了。

2. Visualino 主界面

打开 Visualino 软件，其界面分为菜单栏、快捷操作按钮、图形模块菜单区、模块编程区、代码显示区、信息提示框、开发板和串口选择区等 7 个部分（图 2.18）。

图 2.18　Visualino 主界面

Visualino 主界面与 Arduino IDE 主界面类似，各部分功能也相似，请同学们参照 Arduino IDE 自行学习，此处不再赘述。

3. 上传程序

同 Arduino IDE 一样，在编译上传程序前，我们需要选择正确的开发板和端口号。程序验证成功后，即可上传到开发板中。

4. 注意事项

（1）Preferences 中对应存放 arduino_debug.exe 的地址链接（图 2.19）要与真实 arduino_debug.exe 存放地址一致。如果不一致，上传程序则无法成功。可单击右侧浏览按钮 ... ，搜索到 Arduino 安装保存时的根目录，选中 arduino_debug.exe 文件并确认。

（2）右下方的端口号 COMxx 要和 Arduino 的端口号一致（图 2.19），否则也会上传失败。有时候 PC 机连接了鼠标、键盘等，就会出现多个端口，根据前文图 2.15 的方法可以确认 Arduino 的端口号。

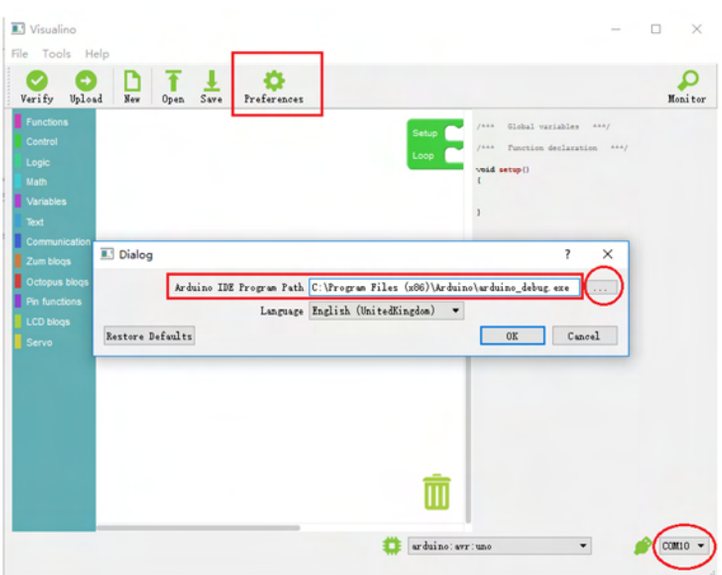


图 2.19　Visualino 注意事项

五、Arduino 控制器的开发语言

（一）Arduino C 语言

1. 基本语法

Arduino C 语言比标准 C 语言简单，但基本语法和程序结构是一样的。下面介绍注释符号、数据与数据类型、表达式、语句和语句块、函数等基本内容。

（1）注释符号。注释符号分单行注释符号和多行注释符号两种。

●单行注释符号“//”：单行注释以一对斜杠作为开头，两个斜杠之间不允许有空格（否则编译器会认为这是除法运算符）。通过这两个斜杠的提示，编译器知道其后面的内容直到本行末尾都是程序的注释，无须进行编译。

●多行注释符号“/* */”：多行注释以一个斜杠加一个星号（/*）作为开头，以星号加斜杠（*/）作为结束，同样星号与斜杠之间不能加空格，在开头和结尾之间的内容都会视为注释。

（2）数据与数据类型。数据是具有一定格式的数字或数值，C 语言中的数据有常量和变量之分。常量是指在程序运行过程中值保持不变的量。常量可以有不同的数据类型，如“0”“1”“2”“-3”等整型常量及“a”“b”等字符型常量。

变量指在程序运行中值可以改变的量。变量由两部分组成：变量名和变量值，它们的值由赋值语句提供。

数据类型指的是数据的不同格式。Arduino C 语言支持的数据类型有布尔型、字符型、整型、无类型等，如表 2.1 所示。

表 2.1　Arduino C 的常见数据类型

数据类型	中文名称	字节长度 (byte)	值范围
boolean	布尔型	1	逻辑 Ture 和 False
char	字符型	1	-128~127
int	整型	2	-32 768~32 767
void	无类型	0	没有返值

（3）表达式。表达式是由运算符和数据构成的，最终都会解析为一个数值。根据运算符的不同，可以简单地将表达式分为三类：算术表达式、关系表达式、逻辑表达式。

●算术表达式：用算术运算符和括号将运算对象连接起来的式子称为算术表达式。其中的运算对象包括常量、变量、函数等。

Arduino C 语言中的算术运算符号有 5 种，分别是：

+：加法运算符，或是正值符号；
-：减法运算符，或是负值符号；
*：乘法运算符；
/：除法运算符；
%：模（求余）运算符。

运算符的优先级规定：先乘、除、模，后加、减，括号的优先级最高。

●关系表达式：用关系运算符将两个或两个以上表达式连接起来的式子称为关系表达式。关系表达式产生的结果是一个逻辑值，即真或假。1 代表真，0 代表假。Arduino C 语言中的关系运算符有如下 6 种：

<:小于；
>:大于；
<=：小于或等于；
>=：大于或等于；
==：等于；
!=：不等于。

其中，前 4 种关系运算符的优先级相同，后 2 种的优先级也相同，但前 4 种的优先级高于后 2 种。

●逻辑表达式：是用逻辑运算符将关系表达式或逻辑量连接起来的式子。其值是逻辑量“真”或“假”。1 代表真，0 代表假。Arduino C 语言包含“与”“或”“非”3 种逻辑关系，其逻辑运算符为：

&& ：逻辑“与”；
||：逻辑“或”；
！：逻辑“非”。

“&&”和“||”是双目运算符，要求有两个运算对象，而“！”是单目运算符，只要求有一个运算对象。

下面举例说明逻辑运算符的用法。

若a=4，b=5，c=5，则
d=!(b>a) 为假，即d的值为0，因为b>a为真；
e=(b>a)||(a>c) 为真，即e的值为1，因为b>a为真，两者相或也为真；
f=(b>a)&& (a>c)为假，即f的值为0，因为a>c为假，两者相与也为假；
g=!(b>a)&&(c>a)为假，即g的值为0，因为！的优先级高于&&，故先执行!(b>a)，再执行&&的逻辑判断，故结果为假。

（4）语句和语句块。对计算机而言，语句是一条完整的 C 指令。所有的 C 语句都以一个分号（；）作为结束。下面是几条 C 语言语句的例子：

```
i=21;
c=a/9;
Drive (2,100);
```

语句块由一个或多个语句构成，语句块由一个左大括号“{”作为开始，一个右大括号“}”作为结束。位于左右大括号之间的所有语句构成语句块主体。大括号一定要成对出现。在编译器看来，一旦执行了一个语句块中的第一条语句，那么后面的语句便顺序执行完成。例如，如果机器人检测前方 1 m 内有障碍物，那么它将后退、再左转，否则前进。

```
if(distance<1)        // 如果距离小于 1 m
{
    backward( );    // 机器人后退
    turnleft( );    // 机器人左转
}
else                  // 否则距离不小于 1 m
{
    forward( );     // 机器人前进
}
```

（5）函数。Arduino C 语言中有两个必须存在的函数：setup( ) 函数和 loop( ) 函数。setup( ) 函数是所有程序的起点，对所用引脚进行状态等的初始化设置，在程序运行时只被调用一次。在 setup( ) 函数完成工作后，会自动执行 loop( ) 函数，并一直循环执行下去，直到外力（例如断电）终止它。

除了 setup( ) 和 loop( ) 函数外，Arduino C 语言中的函数分为两大类，一类是 Arduino 库函数，一类是自定义的函数。库函数是 C 语言软件提供的函数，用户按照函数格式直接使用。这样不仅使程序简洁，也节省了程序开发时间。自定义函数是用户自己根据实际需要编写的函数。编写自定义函数的格式如下：

```
函数类型   函数名（函数参数）
{          // 大括号内为函数体
    语句块；
    ……
}
```

2. 程序结构

在 C 语言程序中，一共有三种程序结构：顺序结构、选择结构（分支结构）和循环结构。

（1）顺序结构。程序从头到尾一句接着一句地执行下来，直到执行完最后一句。

（2）选择结构。程序执行到某个节点后，会根据一次判断的结果来决定之后向哪一个分支方向执行。Arduino C 语言中常用的选择结构主要有以下两种。

● if–else 语句：

```
形式 1：if–else
if ( 表达式 1)        // 如果表达式 1 为真，则执行语句块 1，否则执行语句块 2
{
    语句块 1;
}
else
{
    语句块 2;
}
形式 2：if– else if –else
if ( 表达式 1)          // 如果表达式 1 为真，则执行语句块 1，否则判断表达式 2 是否为真
```

```
{
    语句块 1;
}
else if ( 表达式 2)     // 如果表达式 2 为真则执行语句块 2，否则直接执行语句块 3
    {
        语句块 2;
    }
    else
    {
        语句块 3;
    }
```

开头的 if 和结尾的 else 都只能有一个，但是中间的 else if 可以有很多个。

● switch case 语句：

```
switch ( 变量 )
{                       // 执行到这一句时，变量的值是已知的
case 常数 1:            // 判断变量的值是否等于常数 1，若相等则执行语句块 1
    语句块 1;
    break;
case 常数 2:            // 判断变量的值是否等于常数 2，若相等则执行语句块 2
    语句块 2;
    break;
    ……
default:
    语句块 n;           // 如果变量的值不等于任何 case 后的常数，则执行语句块 n
    break;
}
```

（3）循环结构。Arduino C 语言提供了 3 种循环语句，for 语句、while 语句和 do-while 语句。它们可完成相同的功能。循环结构有一个循环体，循环体里是一段代码。对于循环结构来说，关键在于根据判断的结果来决定循环体执行多少次。

● for 语句：

```
for（表达式 1；表达式 2；表达式 3）
{       // 循环体
    语句 1;
    语句 2;
    ……
}
```

其中表达式 1 代表循环的初始条件；表达式 2 代表循环的终止条件；表达式 3 代表从初始条件到终止条件的步长增量。

● while 语句：

```
while( 表达式 )
{          // 循环体
   语句 1;
   语句 2;
   ……
}
```

while 语句的特点是先判断表达式，若表达式的值非 0，程序执行循环体，否则不执行循环。

示例程序：

```
while(1>0)
{                    // 循环体
        Forward ( ) ;       // 机器人前进 1 s
        delay(1000);
        TurnLeft ( ) ;      // 机器人左转 1 s
        delay(1000);
}
```

1>0 结果为真，所以一直循环执行循环体。

● do-while 语句：

```
do
{          // 循环体
   语句 1;
   语句 2;
    ……
}
while( 表达式 );
```

先执行一次循环体内的语句，然后判断表达式。若表达式的值为 1，则返回循环体重新执行该语句，如此反复，直到表达式的值为 0。

示例程序：

```
do                          // 循环语句
{        // 循环体
        Forward ( ) ;       // 机器人前进 1 s
        delay(1000);
        TurnLeft ( ) ;      // 机器人左转 1 s
        delay(1000);
}
while(1<0);
```

1<0 结果为假，所以只执行循环体一次。

（二）Visualino 图形语言

Visualino 图形化编程软件中的模块菜单包括 Functions（函数）、Control（控制）、Logic（逻辑）、Math（数学）、Variables（变量）、Pin functions（引脚函数）、Servo（舵机）等菜单。下面介绍几个常用模块菜单。

（1）Control（控制）菜单（图 2.20）中包含循环、条件判断、选择判断、延时设置等控制模块。

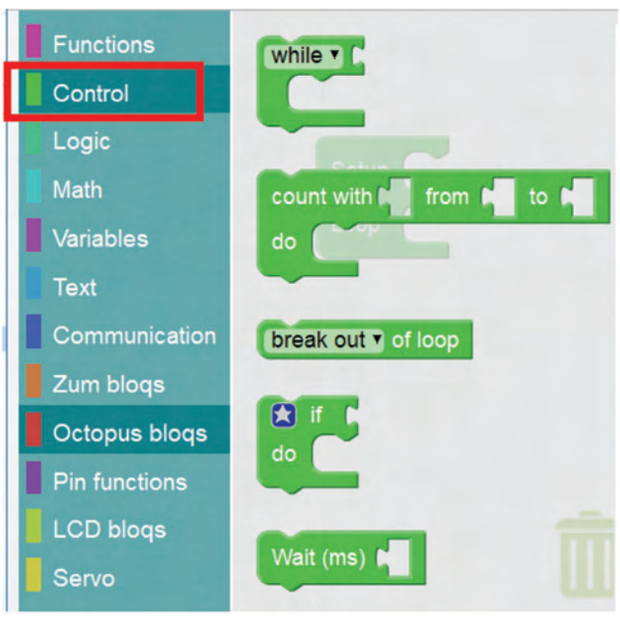


图 2.20　Control（控制）菜单

（2）Logic(逻辑)菜单（图 2.21）中包含与逻辑运算（&&、||、!）、数值比较运算（=、!=、>、<、>=、<=）和逻辑状态（真、假）。

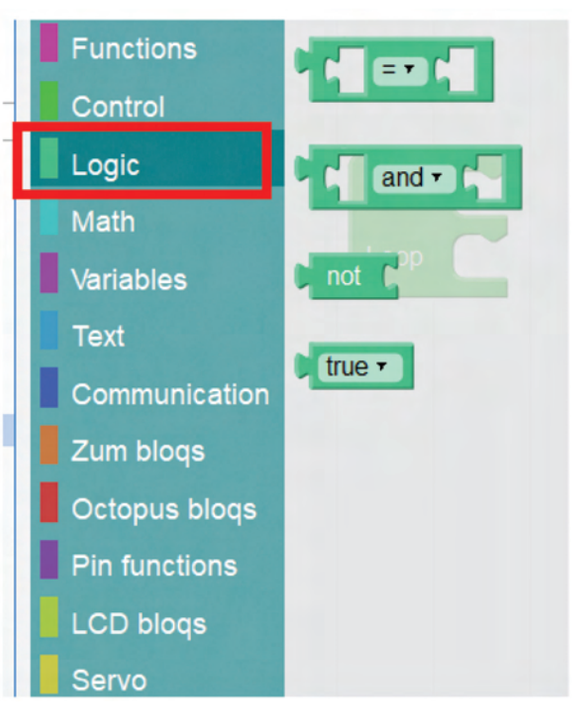


图 2.21　Logic（逻辑）菜单

（3）Math（数学）菜单（图 2.22）中包含数值设置、数字运算、映射、随机数生成等数学模块。

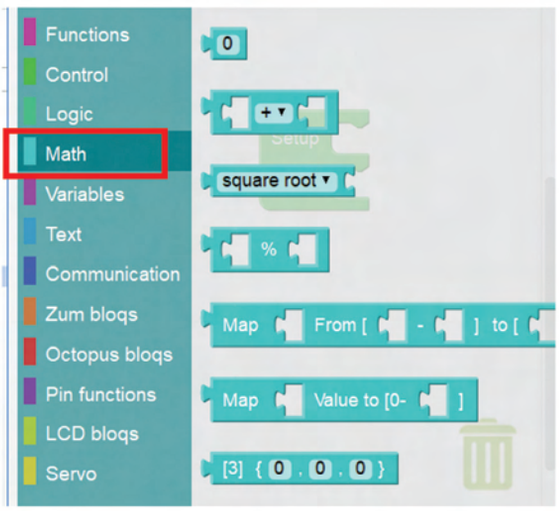


图 2.22　Math（数学）菜单

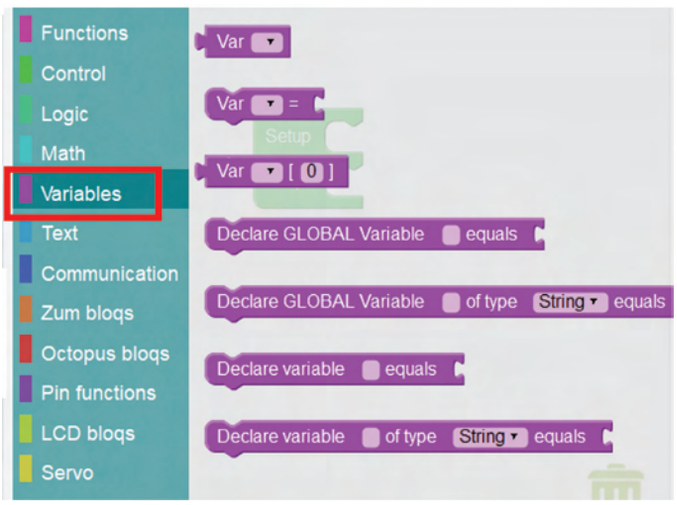


图 2.23　Variables（变量）菜单

（4）Variables（变量）菜单（图 2.23）中包含定义全局变量和局部整型变量模块，可定义的数据类型有整型（int）、字符串型（string）、长整型（long）、字节（byte）、浮点型（float）。

（5）Pin functions（引脚函数）菜单（图 2.24）中包含引脚号选择、引脚状态选择、数字引脚输入设置、数字引脚输出设置、模拟引脚输入设置、模拟引脚输出设置等模块。

（6）Servo（舵机）菜单（图 2.25）中包含舵机的引脚连接和角度控制模块。

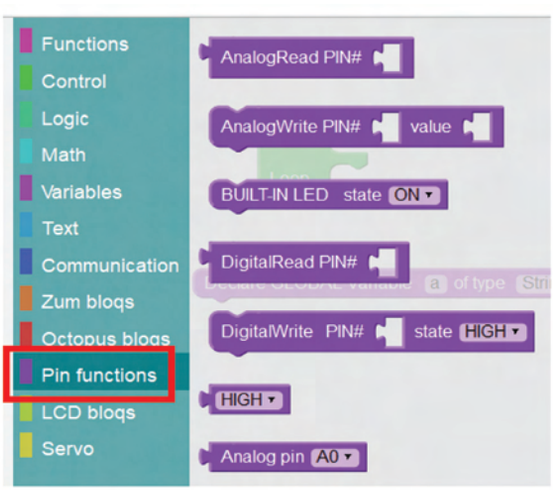


图 2.24　Pin functions（引脚函数）菜单

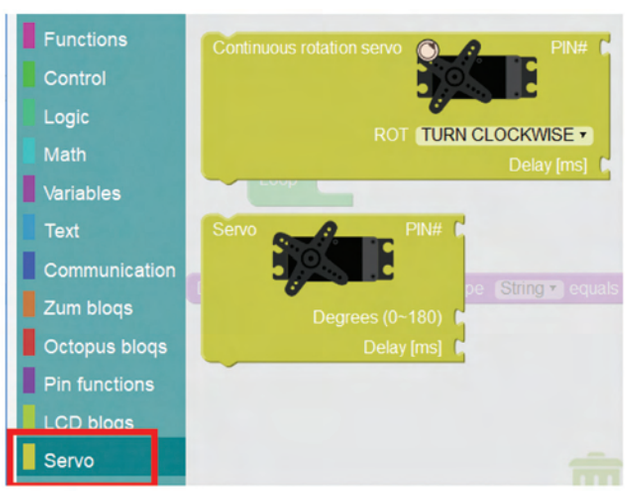


图 2.25　Servo（舵机）菜单

第三节　Arduino 控制器的应用一：LED 的编程控制

学习目标

1. 使用 Arduino 控制器完成简单程序设计。
2. 使用 Arduino 控制器控制 LED 灯。

对 LED 的控制是单片机学习的入门，本节通过 LED 实验让学生对单片机控制的开发流程有系统的认识，为下一章的传感器编程控制打下基础。

一、实验器件

Uno 板 1 块、USB 数据线 1 根、LED1 个、面包板 1 块、1 000 Ω 电阻 1 个、按键 1 个、电位器 1 个、导线若干。

1. 电阻

电阻（图 2.26）一般在电路中起到分压限流的作用，用来保护电路中的元器件。

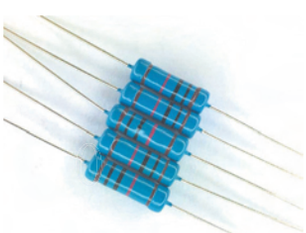


图 2.26　色环电阻

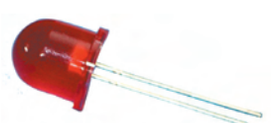


图 2.27　LED

2. LED

LED（图 2.27）有一长一短两个引脚，长引脚为阳极，短引脚为阴极。当电流从 LED 的阳极流入、阴极流出时，便会发光。使用时一般会将 LED 与电阻相连，避免 LED 因两端电压过大而被烧毁。

3. 面包板

面包板的结构分上、中、下三部分。上和下两部分是由两行插孔构成的窄条，中间部分由一条隔离槽和上下各 5 行的隔离插孔构成，如图 2.28 所示。

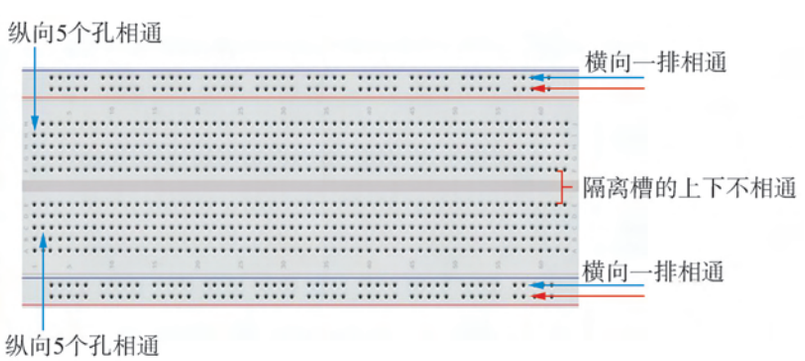


图 2.28　面包板

4. 按键开关

本实验所用按键开关（图 2.29）具有 4 个引脚，主要用于控制电路的接通与断开。当按键按下时电路接通，按键弹起时电路断开。在按键的底部有一个长形凹槽，将左右两侧的引脚隔开，意味着两侧不相通。从按键内部连接示意图（图 2.30）可清楚地观察到：1 和 4 两个引脚相通，2 和 3 两个引脚相通，按下按键时，4 个引脚便全部接通。实际拿到按键开关时，如果只需要连接两个引脚，为了方便起见，我们可直接从对角的两个点引线即可。

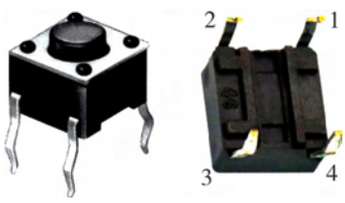


图 2.29　按键开关

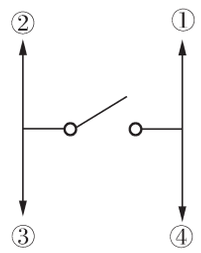


图 2.30　按键引脚示意图

5. 电位器

电位器是阻值可以变化的电阻元件（图 2.31）。它有 3 个引脚，两端分别为正极（5 V）引脚和负极（GND）引脚，中间的是输出引脚。使用时，旋转电位器上的旋钮，电位器的阻值发生变化，输出引脚的电压也会随之发生改变。电位器输出引脚连接到单片机的模拟输入引脚，就可以将电压信号量转换成离散的数字量。

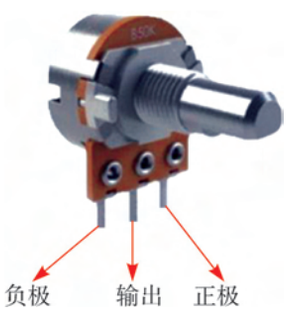


图 2.31　电位器

二、实验内容

本节课程我们通过 LED 实验走进奇光异彩的世界，分别学习数字输出、数字输入，同学们还可以通过挑战项目掌握模拟输入、模拟输出。

实验 1：闪烁的 LED

1. 实验要求

每隔 1 s 改变 1 次 LED 两端的电压，控制 LED 点亮和熄灭，使 LED 呈现闪烁的效果。

2. 程序流程分析

本实验要实现 LED 闪烁：亮 1 s，灭 1 s，共两种状态。而数字信号有两种状态，高电平或低电平：高电平为 5 V，低电平为 0 V。因此，可以将 LED 连接到 Uno 板上的数字引脚（D0~D13）。我们选取 D13 引脚作为这次实验中 LED 阳极的接口，LED 的阴极则接 GND。电路工作时，先通过 D13 引脚输出一个高电平，点亮 LED；1 s 后，再通过 D13 引脚输出一个低电平，熄灭 LED。重复这一过程，LED 便呈现闪烁的效果。

做中学

请同学们根据上述分析，完成程序流程图 2.32。

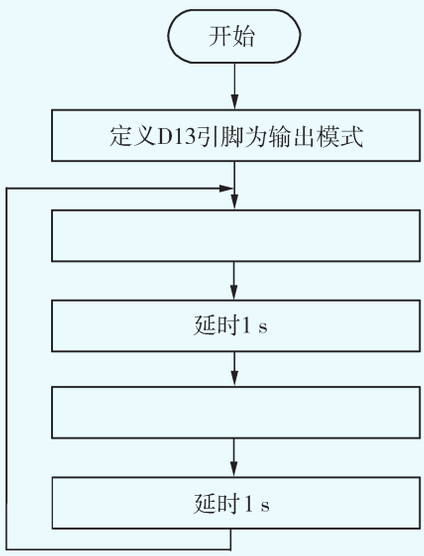


图 2.32　闪烁的 LED 程序流程图

小贴士

视觉暂留现象

视觉暂留现象又称“余晖效应”。人眼在观察景物时，光信号成像于视网膜上，并由视神经输入人脑，由此感觉到物体的像。但当物体移去时，视神经对物体的印象不会立即消失，而要延续 0.1~0.4 s 的时间，人眼的这种特性被称为“眼睛的视觉暂留”。

Arduino 执行一条指令的时间远远小于 1 ms，因此需要在数字引脚赋值语句后面增加延时函数，否则 LED 亮灭频率过高，由于余晖效应，人眼看不到 LED 的熄灭状态。同学们可以在本实验中改变延时时间，体会眼睛的视觉暂留现象。

3. 电路搭建

为了避免 LED 因两端电压过大而被烧坏，将 LED 与电阻相连（图 2.33）。

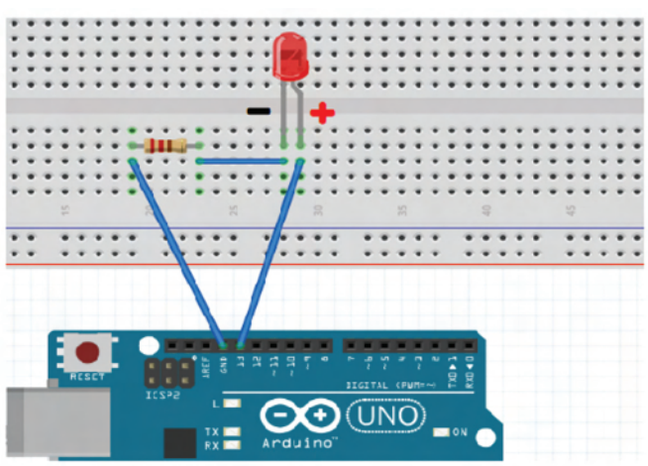


图 2.33　闪烁的 LED 电路接线图

4. 图形化语言编程

（1）图形模块介绍。

●数字引脚输出设置模块 DigitalWrite（图 2.34）在引脚菜单 Pin functions 中，用于定义数字引脚的高、低电平状态。

图 2.34　数字引脚输出设置模块 DigitalWrite

●数字引脚号选择模块 Digital pin（图 2.35）在引脚菜单 Pin functions 中，用于选择目标引脚，可以和数字引脚输入设置模块 DigitalRead 一起使用，完成对目标数字引脚的数据读取。

●延时设置模块 Wait(ms)（图 2.36）在控制菜单 Control 中，用于设置延时时间，单位是 ms，1 s=1 000 ms。

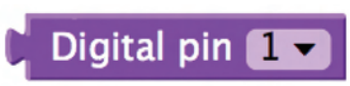


图 2.35　数字引脚号选择模块 Digital pin

图 2.36　延时设置模块 Wait(ms)

（2）根据程序流程图，完成图形化编程。

第 1 步：点亮 LED，即给 Uno 板的 D13 引脚写入高电平。在引脚菜单 Pin functions 中选择数字引脚输出设置模块 DigitalWrite，将其拖动到编程区的 Loop 中，并使两者的凹槽吻合，可以听到“咔哧”的声音（图 2.37）。

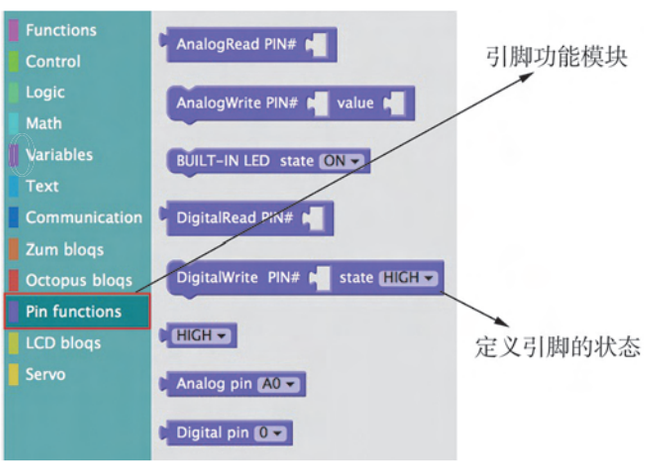


图 2.37　选择数字引脚输出设置模块 DigitalWrite

第 2 步：在 DigitalWrite 的第一个空槽处添加引脚号。单击引脚菜单，找到数字引脚模块 Digital pin，拖到凹槽中。单击 Digital pin 的黑色下拉三角，选择数字 13，然后将 state 选为“HIGH”，这样就完成了 D13 引脚输出高电平的定义，即点亮 LED（图 2.38）。

图 2.38　设置 D13 引脚为高电平

第 3 步：实现延时 1 s。在控制菜单 Control 中找到 Wait(ms) 延时模块，然后拖到 Loop 中（图 2.39）。在空槽处填入数字来表示延时的时长（如 Wait(ms) 2 000，表示延时 2 s）。可以通过在数学菜单 Math 中选择数值模块来实现数字的填入（图 2.40）。

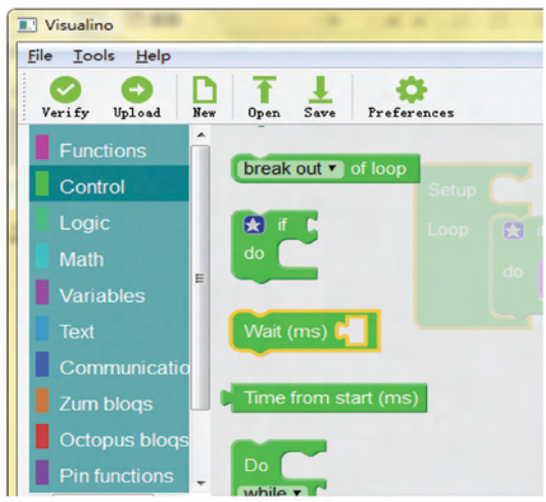


图 2.39　选择 Wait(ms) 延时模块

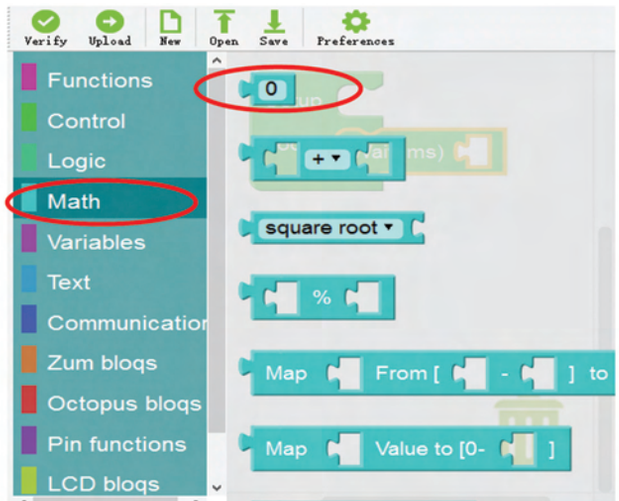


图 2.40　选择数值模块

第 4 步：实现 LED 熄灭 1 s，需要将 D13 引脚的 state 选为“LOW”，然后延时 1 s。接下来请同学们自己完成 LED 熄灭 1 s 的程序。

本实验的完整图形化程序见图 2.41。

第 5 步：用 USB 数据线将 Uno 板与 PC 机相连接，在右下角选择好开发板、端口号，然后编译、上传程序，观察实验现象是否为 LED 闪烁。

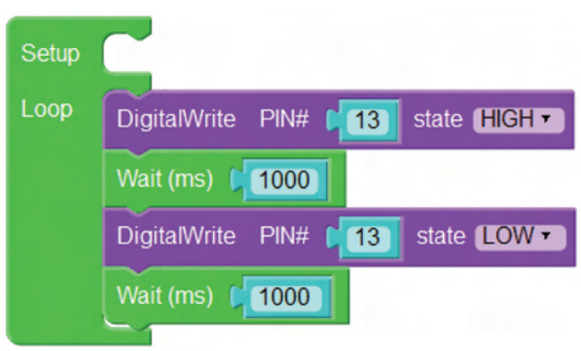


图 2.41　闪烁的 LED 的完整图形化程序

5. 代码编程

做中学

请同学们根据上述分析补全如下闪烁的 LED 代码程序。

```
void setup( )
{                                    // 初始化设置
   pinMode(13,_____);                // 设置 D13 引脚的输入输出模式
}
void loop( )
{                                    // 循环函数部分
   digitalWrite(13,_____);           // 点亮 LED
   delay(1000);                      // 延时 1 s，灯亮 1 s
   digitalWrite(13,_____);           // 熄灭点亮 LED
   delay(1000);                      // 延时 1 s，灯灭 1 s
 }
```

编程注意事项：①输入法设为英文模式；分号“;” 勿遗忘；“｛｝”大括号成对出现，关键字注意大小写。②本实验通过D13引脚输出信号给LED，因此调用pinMode( )函数定义D13引脚为输出模式。③先后调用digitalWrite( )函数和delay( )函数实现LED亮1 s、灭1 s，在loop函数中实现闪烁效果。

实验 2：按键控制 LED

1. 实验要求

按键按下时 LED 点亮，按键弹起时 LED 熄灭。

2. 程序流程分析

若要实现按键控制 LED，则需用 if( ) 语句不断地检测按键的状态。本次实验中 LED 只有两个状态（亮、灭），按键也只有两个状态（通、断）。所以，我们选择数字输入、输出引脚：LED 连接到 Uno 板的 D13 引脚，按键连接 D5 引脚。若按下按键，D5 引脚输入为低电平（对应的数字信号为 0）；若未按下，输入为高电平（对应的数字信号为 1）。所以检测到 D5 引脚为低电平时，令 D13 引脚输出高电平，即可完成点亮 LED，反之熄灭 LED。

做中学

请同学们根据上述分析完成程序流程图 2.42。

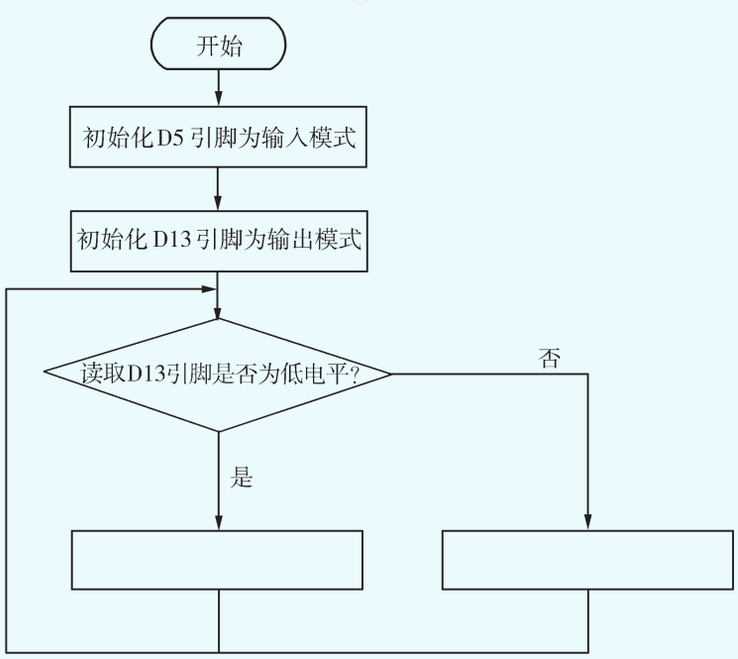


图 2.42　按键控制 LED 的程序流程图

3. 电路搭建

为了保证未按下按键时 D5 引脚向 Uno 板输入稳定高电平，需要接入上拉电阻，即在连接 D5 引脚的那端接入电阻，电阻另一端连接 5 V（图 2.43）。

4. 图形化语言编程

（1）图形模块介绍。

●数字引脚输入设置模块 DigitalRead（图 2.44）在引脚菜单 Pin functions 中，用来读取数字引脚的状态，得到数字 1 或 0。1 表示高电平，0 表示低电平。

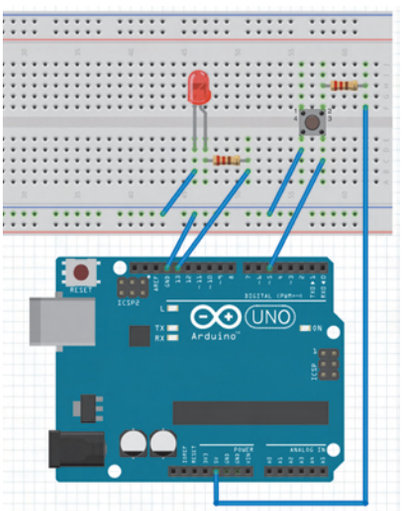


图 2.43　按键控制 LED 的电路接线

图 2.44　数字引脚输入设置模块 DigitalRead

●条件判断模块 if（图 2.45）在控制菜单 Control 中，用来实现条件判断。单击五角星，可以将 else 子模块拖到 if 下面的凹槽中。当 if 的假设条件成立时，执行 do 后面的语句；当 if 的假设条件不成立时，执行 else 后面的语句。

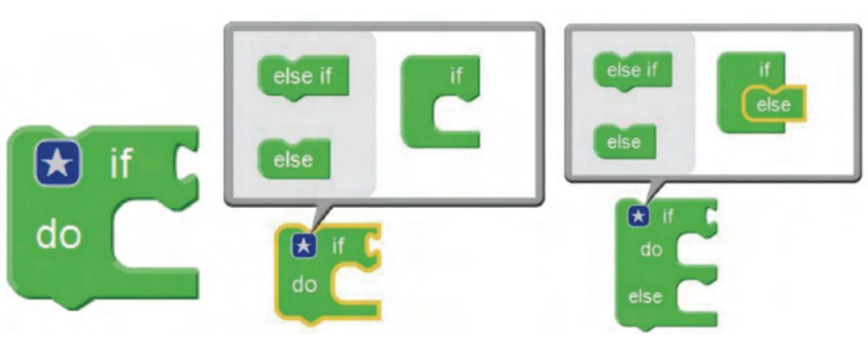


图 2.45　条件判断模块 if

（2）根据程序流程图，完成图形化编程。

第 1 步：在控制菜单 Control 中选择 if 条件判断模块，将其拖到 Loop 中。在 if 后面添加判断条件。条件为：读取数字引脚 D5 的状态为 0。

第 2 步：读取数字引脚 D5 的状态为 0 时，判断条件成立，执行 do 后的语句，利用 DigitalWrite 使 D13 引脚输出高电平。

第 3 步：读取数字引脚 D5 的状态为 1 时，判断条件不成立，执行 else 后的语句，利用 DigitalWrite 使 D13 引脚输出低电平。

本实验的完整图形化程序见图 2.46。

第 4 步：用 USB 数据线将 Uno 板与 PC 机相连接，在右下角选择好开发板、端口号，然后编译、上传程序，观察按键开关按下或者弹起时 LED 的状态。

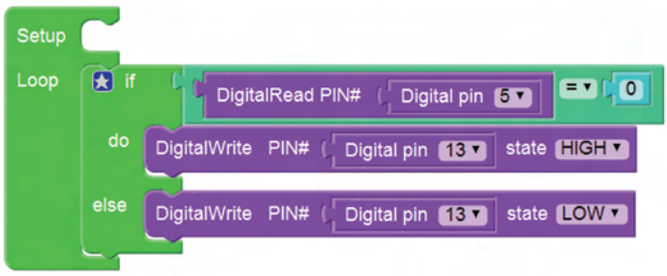


图 2.46　按键控制 LED 的完整图形化程序

做中学

请同学们根据上述分析，补全如下按键控制 LED 的代码程序。

```
void setup( )
{
  pinMode(5,_____);       // 设置 D5 引脚为输入模式

  pinMode(13,_____);     // 设置 D13 引脚为输出模式
}
void loop( )
{
  if (digitalRead(5)==_____)                    // 读取 D5 引脚的数字信号
   {
    digitalWrite(13,_____); // 点亮 LED
    }
else
    {
      digitalWrite(13,_____); // 熄灭 LED
     }
  }
```

挑战项目 1：LED 呼吸灯

请同学们通过改变 LED 阳极的电压，实现 LED 由熄灭状态逐渐变亮并逐渐熄灭的循环变化效果。

程序流程分析：

本次实验，我们要实现 LED 由熄灭状态逐渐变亮再逐渐熄灭的循环变化，这个过程中 LED 有许多亮度状态，我们需要让控制器输出不同的电压使 LED 呈现不同的亮度。

模拟信号有多种状态，0~5 V 的电压，对应输出的 PWM 值为 0~255。LED 的亮度由输出的 PWM 值决定：PWM 值越大，模拟电压值越大，则 LED 亮度越强；反之，LED 亮度变弱。所以，要将 LED 连接到 Uno 板上的模拟输出口（数字前带有 ~ 符号的引脚）。我们选取编号为 ~D5 的模拟输出口作为这次实验中 LED 的阳极接口。电路工作时，先通过

~D5 端口给 LED 一个 PWM 值为 0 的输出，然后再逐渐增加 PWM 值至 255，接着再逐渐降低至 0，如此往复，实现呼吸灯的效果。

请同学们根据上述分析完善 LED 呼吸灯程序流程图 2.47，并编写程序代码。

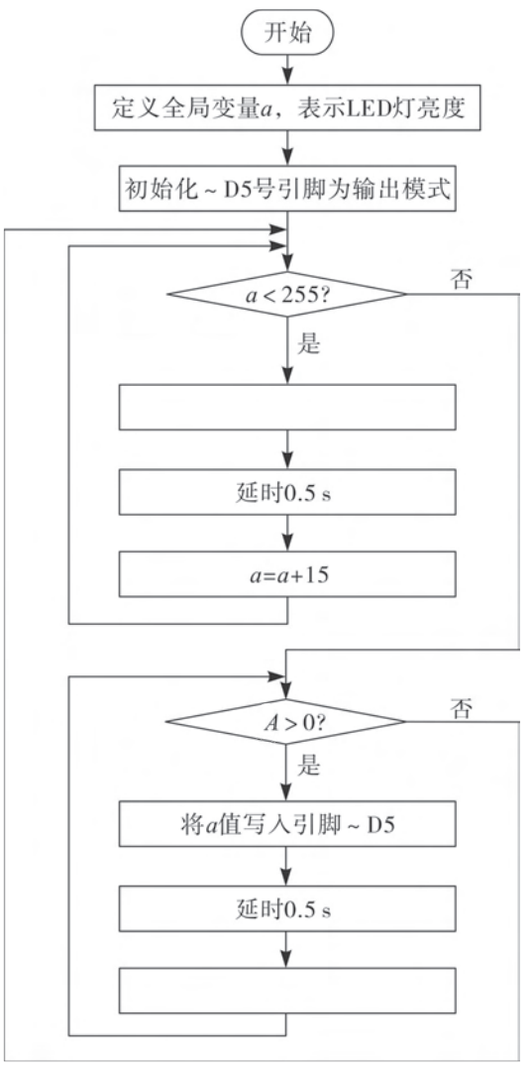


图 2.47　LED 呼吸灯的程序流程图

挑战项目 2：用电位器控制 LED 亮度

请同学们通过电位器来控制 LED 的亮度。

程序流程分析：

本次实验，我们将电位器的输出引脚与 Uno 板的 A0 口相连接，LED 与可以输出 PWM 的引脚 ~D9 连接，LED 的亮度由 PWM 的输出值决定。

Uno 板读取电位器的模拟电压值并通过映射函数转化为 PWM 输出。读取电位器的模拟电压值越大，PWM 输出值越大，则 LED 亮度越强；反之，LED 亮度变弱。

请同学们根据上述分析完善程序流程图 2.48，并完成编程代码。

请同学们根据上述分析完成实验。

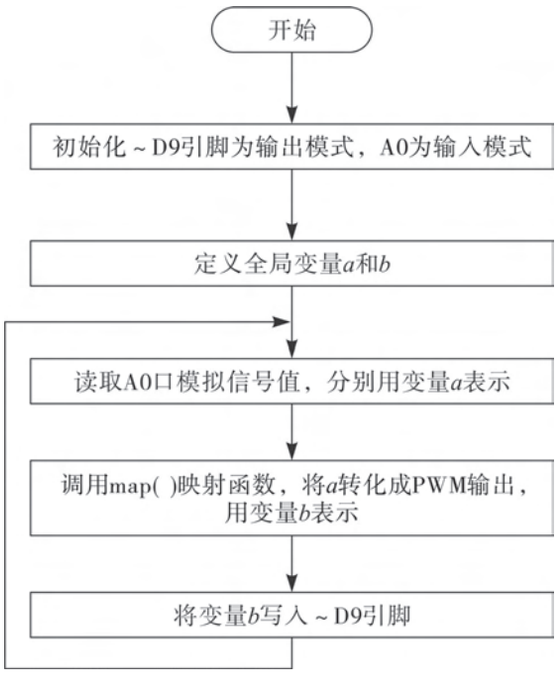


图 2.48　用电位器控制 LED 亮度的程序流程图

阅读材料

PWM 占空比输出和映射

PWM控制技术是pulse width modulation（脉冲宽度调制）的缩写，就是对脉冲的宽度进行调制的技术（图2.49），即按一定的规则改变脉冲的宽度，使输出端得到一系列高度相等而宽度不相等的脉冲，从而改变电路有效输出电压的大小。脉冲宽度越大，PWM值越大，有效输出电压越大。比如通过改变PWM值来改变电路有效输出电压，从而改变LED亮度；也可以通过改变PWM值来控制电动机转速。PWM值的计算方法如下：

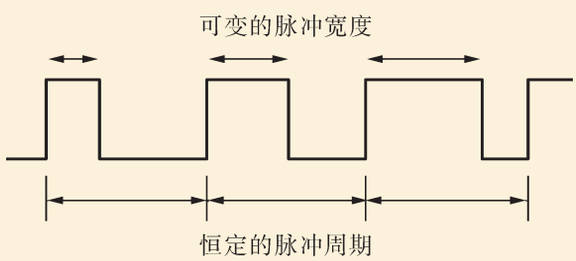


图 2.49　PWM 控制技术

（1）Uno 板引脚输出的最大电压为 5 V；

（2）PWM 就是把这 5 V 分成了 255 份；

（3）计算 PWM 值。

$$\text{PWM 值} = \frac{\text{模拟电压值}}{5\ \text{V}} \times 255$$

为了实现电位器的模拟输入信号(数字范围0~1 023)来控制LED的PWM输出(数字范围0~255)，我们引入映射这一概念。模拟输入转化的数字范围大于PWM输出的范围，直接把模拟输入当作PWM输出显然是不可以的，所以我们要把模拟输入转化为PWM输出：模拟输入的最小值0对应PWM输出的最小值0，模拟输入的最大值1 023对应PWM输出的最大值255。简单地说，就是将模拟输入等比缩小成PWM输出，这种可缩放的转化关系就叫作映射。因此，我们可以说将0~1 023映射到0~255。

第四节　Arduino控制器的应用二：移动机器人编程控制

学习目标

1. 使用Arduino控制器控制机器人运动。
2. 调用自定义函数编程控制机器人运动。

一、实验器件

机器人教具1套、USB数据线1根、导线若干。

1. 机器人教具

本教材的机器人教具采用典型的轮式移动机器人，通过左、右轮的差速运动实现转弯（图2.50）。在开展实验前，请自行完成机器人的组装，组装步骤参考第一章简易机器人小车的组装与调试实验。

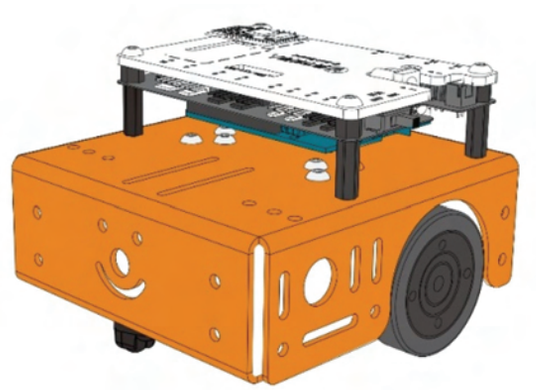


图2.50　轮式移动机器人教具

2. 机器人控制器

机器人控制器由Uno板和扩展板组成，Uno板是主控器；扩展板集成了电动机驱动及各种外设接口，可以很方便地与电动机及各种传感器连接（图2.51）。

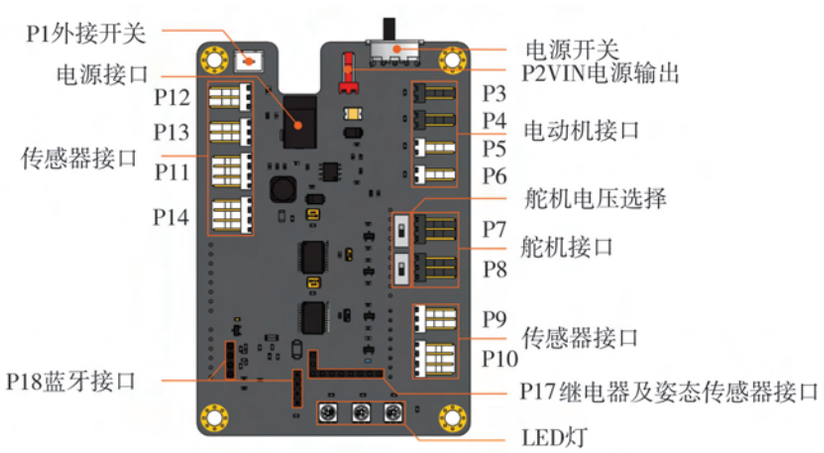


图 2.51　机器人控制器

电动机驱动方式：扩展板上引出的 4 路电动机接口输出的是电动机驱动芯片 TB6612 产生的电动机控制信号。在扩展板上每个端口对应标注了 Uno 板的控制引脚（例如 P6 端口对应标注 ~D6、D7），其中带有“~”的引脚为 PWM 输出引脚，通过输出不同的 PWM 值来控制电动机的转速，另外一个引脚通过数字输出 HIGH 或者 LOW 来控制电动机转动方向。如表 2.2 所示。

表 2.2　电动机接口引脚说明

端口	P3	P4	P5	P6
对应控制引脚	~D10、D12	D8、~D9	~D5、D4	~D6、D7

3. 直流电动机

我们可以看到直流电动机有两个接线引脚（图 2.52），只要给两接线引脚加上电压就能实现电动机转动。在电动机的耐压值范围内，电压越大，直流电动机转动越快；反之，电压越小，直流电动机转动越慢。我们可以通过改变 PWM 值来控制电动机转速。如果将两个接线引脚的接线对调一下，电动机将反向转动。

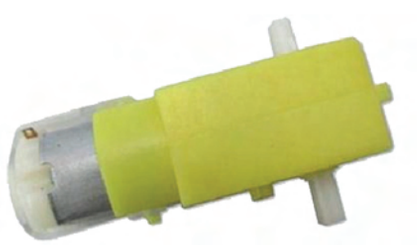


图 2.52　直流电动机

二、实验内容

本节课我们通过分别使用 Visualino 软件、Arduino IDE 软件、调用自定义函数编程控制机器人运动。

1. 实验要求

分别使用 Visualino 软件、Arduino IDE 软件、调用自定义函数编程控制机器人实现前进 2 s、后退 2 s、原地左转 2 s、原地右转 2 s、停止 2 s 的循环运动。

2. 电路搭建

机器人控制板上有 4 组端口（P3、P4、P5、P6）可以驱动电动机，本次实验使用 P3 和 P6 端口。将左电动机接在扩展板 P3 端口（~D10、D12），右电动机接在扩展板 P6 端口（~D6、D7）。~D10 用来控制左电动机的转速，D12 用来控制左电动机的转向；~D6 用来控制右电动机的转速，D7 用来控制右电动机的转向（图 2.53）。

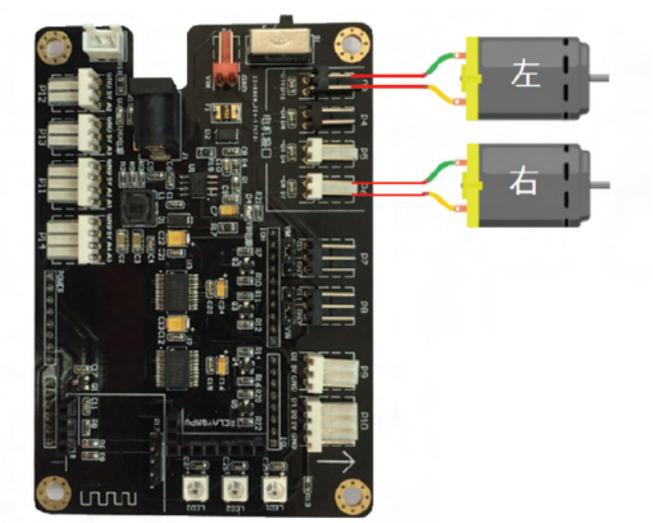


图 2.53　电动机与扩展板接线图

3. 程序流程分析

机器人由两个电动机分别控制其左右两侧轮子的运动，来改变运动状态：前进、后退、原地左转、原地右转、停止。机器人的五种运动状态见表 2.3，LOW 表示低电平（驱动电动机正转），HIGH 表示高电平（驱动电动机反转）。

表 2.3　机器人运动状态

引脚号		左轮		右轮	
		~D10	D12	~D6	D7
前 进	两个电动机以同样的转速同时正转	PWM 输出值为 0~255 数字信号	LOW	PWM 输出值为 0~255 数字信号	LOW
后 退	两个电动机以同样的转速同时反转		HIGH		HIGH
原地左转	左电动机反转，右电动机正转，转速一致		HIGH		LOW
原地右转	右电动机反转，左电动机正转，转速一致		LOW		HIGH
停 止	两个电动机的转速同时为 0		LOW/HIGH		LOW/HIGH

4. 图形化语言编程

第 1 步：参考机器人运动状态表，设计令机器人前进的图形化程序。首先，调用数字引脚输出设置模块 DigitalWrite 来设置 D7、D12 引脚输出低电平，使左、右电动机的转向为正转。然后，调用模拟引脚输出设置模块 AnalogWrite 来设置引脚 ~D6、~D10 输出的 PWM 值（0 ~ 255），示例中取 100，同学们也可以尝试输入不同的 PWM 值，再观察机器人运动速度的变化。

第 2 步：实现延时 2 s。在控制菜单 Control 中找到 Wait(ms) 延时模块，设置延时 2 s。

第 3 步：其他运动状态的编程方法与前进类似，请同学们自行编写完成。

本实验的完整图形化程序见图 2.54。

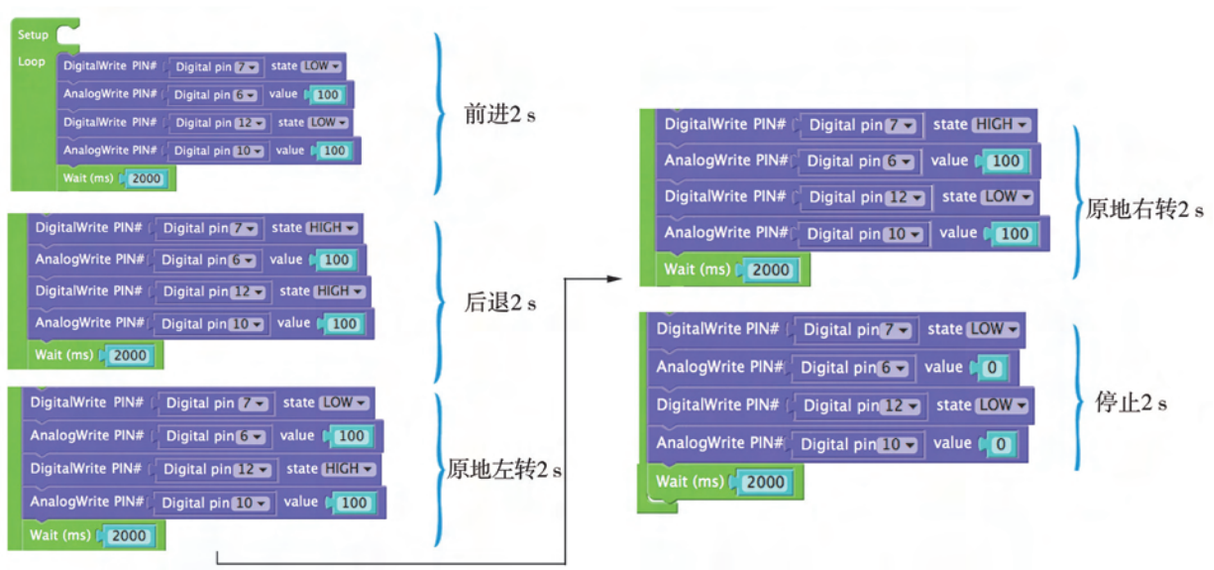


图 2.54　机器人运动的完整图形化程序

第 4 步：在右下角选择好开发板、端口号，然后编译、上传程序。观察机器人是否能顺利完成 5 个运动状态。

提示：由于实验用的简易电动机出厂后的性能参数会存在一定的差异，即使施加了相同的电压，也很难保证达到相同的转速，因此导致相同占空比电压驱动下的两个电动机转速不一致，使机器人产生差速运动而走不了直线，因此可以适当提高运动速度相对较慢的电动机的驱动电压占空比，或者适当降低运动速度相对较快的电动机的驱动电压占空比。

5. 代码编程

第 1 步：在 setup( ) 函数中初始化引脚状态，调用 pinMode( ) 函数定义引脚 ~D6、D7、~D10、D12 为输出模式。

第 2 步：使机器人前进 2 s。调用 digitalWrite( ) 函数让 D7、D12 号引脚输出低电平来实现车轮正转；调用 analogWrite( ) 函数让 ~D6、~D10 号引脚输出 PWM 值 100 来实现车轮保持一定转速。调用 delay( ) 函数实现延时 2 s（图 2.55）。

第 3 步：编程实现机器人后退 2 s、原地左转 2 s，原地右转 2 s，停止 2 s。

图 2.55　机器人前进 2 s 的图形化程序与代码程序的关系

做中学

请同学们根据上述分析补全如下机器人运动控制代码程序。

```
/********************** 初始化引脚设置 ******************/
void setup( )
{
  // 定义引脚 D7，D12，~D6，~D10 为输出
  pinMode(7, _____ );
  pinMode(6,_____ );
  pinMode(12,_____ );
  pinMode(10,_____ );
}
/************************** 主函数 **********************/
void loop( )
 {
  // 前进 2 s
  digitalWrite(7,_____ );
  analogWrite(6,_____);
  digitalWrite(12,_____ );
  analogWrite(10,_____);
  delay(2000);
  // 后退 2 s
  digitalWrite(7,_____ );
  analogWrite(6,_____);
  digitalWrite(12,_____ );
  analogWrite(10,_____ );
  delay(2000);
  // 原地左转 2 s
  digitalWrite(7,_____);
  analogWrite(6,_____);
  digitalWrite(12,_____);
  analogWrite(10,_____ );
```

```
    delay(2000);
    // 原地右转 2 s
     digitalWrite(7,_____);
     analogWrite(6,_____);
    digitalWrite(12,_____ );
    analogWrite(10,_____);
    delay(2000);
    // 停止 2 s
    digitalWrite(7,_____);
    analogWrite(6,_____);
    digitalWrite(12,_____);
    analogWrite(10,_____ );
    delay(2000);
}
```

第4步：运行程序。用USB数据线将Uno开发板与PC机相连接，选择正确的开发板、端口号后，对代码进行编译，上传代码。上传成功后，拔掉数据线并将扩展板的电源开关打开，观察机器人运动状态。

6. 利用自定义函数优化控制程序

根据上面两个程序，我们会发现机器人的一个运动状态需要写 5 条语句，如果机器人在各种运动状态中频繁切换，那么程序就会变得很冗长，需要不断重复很多语句。

庆幸的是，我们可以通过自定义函数来将实现某功能的语句打包，等当需要使用时直接调用自定义函数即可。所谓的自定义函数，就是用户自己定义的函数。比如我们可以用 void Drive(int state,int speed) 函数来实现机器人的运动状态控制，参数 state 表示运动方式，参数 speed 表示运动速度，而具体的控制程序则写在函数体中。

第 1 步：写出完整的自定义 Drive( ) 函数。

第 2 步：在 setup( ) 函数中进行初始化设置，定义引脚 D7，D12，~D6，~D10 为输出。

第 3 步：在 loop( ) 函数中调用 Drive( ) 函数来控制机器人运动，调用 delay( ) 函数延时 2 s，从而依次实现前进 2 s，后退 2 s，原地左转 2 s，原地右转 2 s，停止 2 s。

做中学

请同学们根据上述分析补全如下机器人运动控制代码程序。

```
/************ 控制机器人运动的 Drive() 函数 ***************/
void Drive(int state,int speed)
{                                          //state 运动状态，speed 运动速度
  analogWrite(6,_____);
  analogWrite(10,_____);          // 设定速度，当均为 0 时，机器人停止运动
  switch (state)
{
```

```
  case 1:                                          //机器人前进
        digitalWrite(7,LOW);
        digitalWrite(12,_____________);
        break;
  case 2:                                          //机器人后退
        digitalWrite(7,HIGH);
        digitalWrite(12,_____________);
        break;
  case 3:                                          //机器人左转
        digitalWrite(7,LOW);
        digitalWrite(12,_____________);
        break;
  case 4:                                          //机器人右转
        digitalWrite(7,HIGH);
        digitalWrite(12,_____________);
        break;
  default:break;                                   //跳出 switch 语句
   }
}
/********************** 初始化引脚设置 ******************/
void setup()
{                                                  //定义引脚 D7、D12、~D6、~D10 为输出
  pinMode(7,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(10,OUTPUT);
}
/************************** 主函数 **********************/
void loop()                                        //主循环程序
{
  Drive(_____________,100);                        //调用 Drive 函数
  delay(2000);                                     //前进 2 s，速度值 100
  Drive(_____________,100);
  delay(2000);                                     //后退 2 s，速度值 100
  Drive(_____________,100);
  delay(2000);                                     //左转 2 s，速度值 100
  Drive(_____________,100);
  delay(2000);                                      //右转 2 s，速度值 100
```

```
    Drive(0,_____);
    delay(2000);                          // 停止 2 s
}
```

第 4 步：运行程序。用 USB 数据线将 Uno 板与 PC 机相连接，选择好开发板、端口号，然后编译、上传程序，观察机器人运动状态是否符合实验要求。

本章小结

机器人的一切行为都是由它的控制器支配的，简易机器人控制器的核心一般都采用单片机。目前使用比较广泛是 Arduino 控制器，它简化了单片机工作流程，将复杂的单片机底层代码封装成简单易用的函数。

Arduino 控制器开发环境支持图形化语言编程和 Arduino C 语言编程两种编程方式。图形化编程方法与编制程序流程图在思路和形式上具有相似之处，只是将每个流程节点换成适当的图形化模块。Arduino C 语言的语法和程序结构与标准 C 语言基本形同，仅库函数有区别。

学习评价

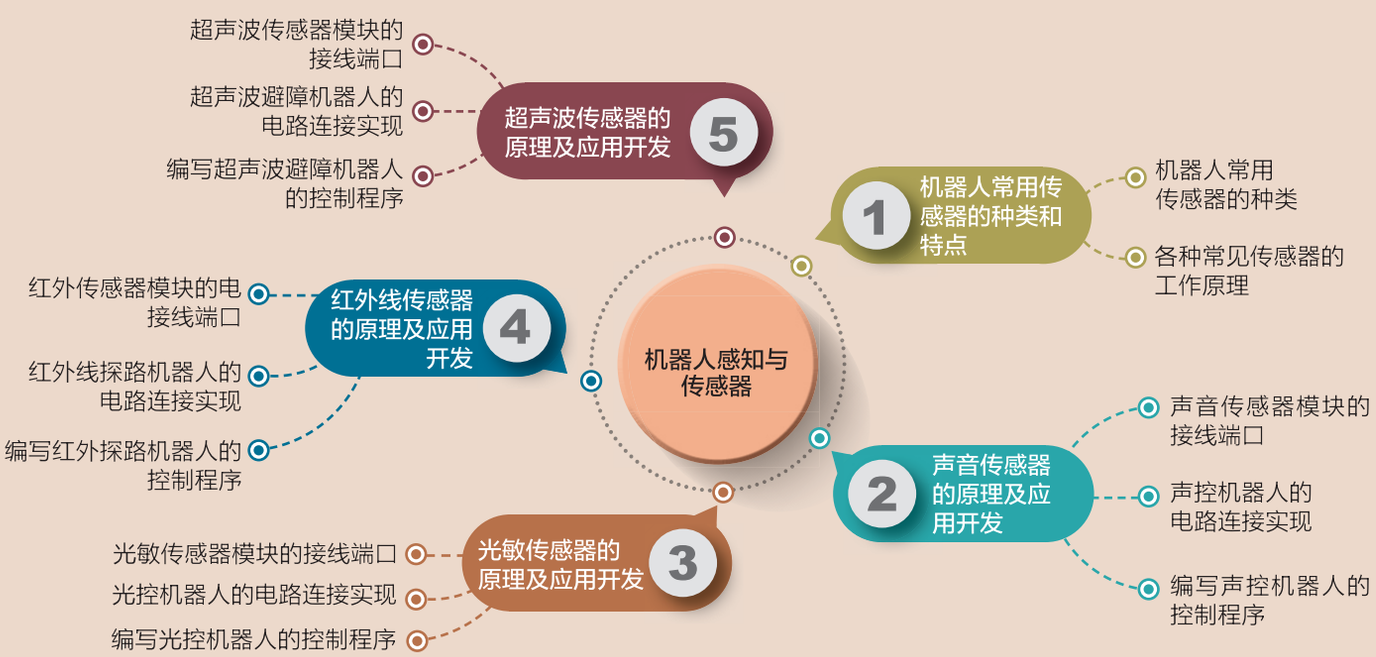
评价内容			评价方式		
			自我评价	小组评价	教师评价
过程评价	师生互动	能积极思考老师提出的问题			
		能基于已有经验构建新的知识			
		能积极参与课堂讨论			
	实践活动	能积极参与实践活动			
		与小组成员有效合作			
		LED 灯的编程控制			
		移动机器人编程控制			
结果评价	目标实现	掌握单片机的基本构成			
		认识 Arduino 控制器			
		基于 Arduino 控制器进行图形化编程和代码编程			
	收获反思	收获与感悟			
		反思不足			

第三章
机器人感知与传感器

导　言

机器人只有具备了感知能力才能够像智能生物一样根据环境的变化做出相应的反应和动作。给机器人装上具有不同功能的传感器，机器人就能获得视觉、听觉、嗅觉和触觉等各种“感觉”。本章重点介绍简易机器人中常用到的声音传感器、光敏传感器、红外传感器和超声波传感器，并分别利用这些传感器完成不同功能要求的机器人的应用开发。

思维导图



第一节　机器人常用传感器的种类和特点

学习目标

1. 认识生活中用到的传感器。
2. 掌握常见传感器的基本工作原理。
3. 能根据实际需求选择合适的传感器。

机器人获取外部环境信息依靠的是传感器。传感器是一种电子器件，它具有类似于人类五官的功能，能将外部环境信息转换成电信号。其基本组成如图 3.1 所示。

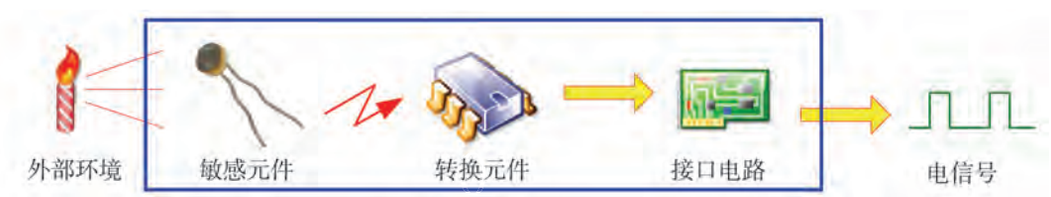


图 3.1　传感器组成

除了类似人的五官的功能之外，传感器还能感知流量、加速度、扭矩等物理量。在我们身边的摄像机、电视机、打印机、自动售货机、报警器等日常用品中，都有传感器的身影。路面上奔驰的汽车，更是通过多达几十种的传感器来保证驾驶的综合效率。可以说，传感器及传感技术的发展水平已经成为衡量一个国家科技实力的重要标志之一。

图 3.2 是传感器在拟人机器人中应用的实例，有温度传感器、视觉传感器、力传感器、触觉传感器等。

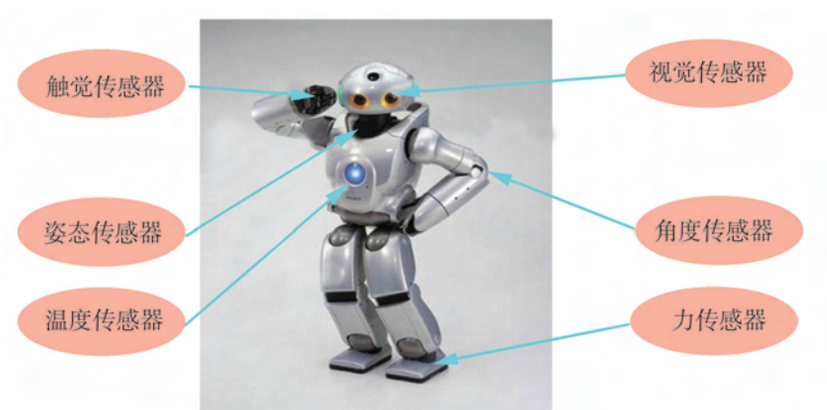


图 3.2　拟人机器人上的传感器

传感器是简易机器人必不可少的器件，本章将重点介绍四种简易机器人常用的传感器：声音传感器、光敏传感器、红外传感器和超声波传感器。

探究与交流

请同学们查阅有关资料，找出功能与人的五官功能类似的传感器并完成表 3.1。

表 3.1　人的五官功能与各种传感器的对应关系

感觉	传感器用例
视觉（眼）	
听觉（耳）	
嗅觉（鼻）	
味觉（舌）	
触觉（皮肤）	

第二节　声音传感器的原理及应用开发

学习目标

1. 掌握声音传感器的工作原理和使用方法。
2. 学会编写声音传感器的控制程序。
3. 能够使用声音传感器完成声控机器人的控制任务。

一、声音传感器简介

声音传感器（图 3.3）既能感受声音强度大小，也能记录声音的波形。它内置了一个对声音敏感的电容式驻极体话筒，声波能使话筒内的驻极体薄膜振动，导致电容变化，从而产生与之对应变化的微小电压。这一电压随后被转化成 0~5 V 的电压，经过 A/D 转换被数据采集器接收，并传送给计算机。

声音传感器让机器人可以“听到”声音。

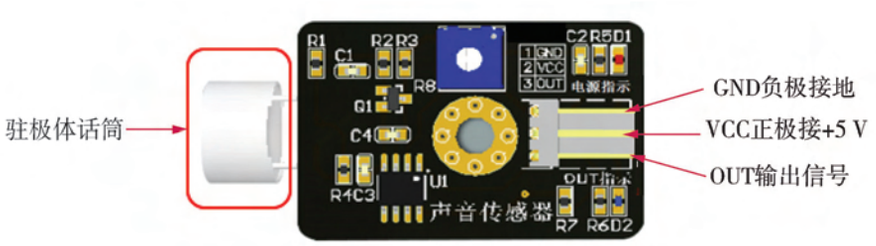


图 3.3　声音传感器模块

当声音传感器模块连接到机器人扩展板上的数字接口时，周边有声音时输出低电平，无声音时输出高电平。调节电位器的阻值可以改变声音传感器模块的灵敏度，用十字螺丝刀顺时针旋转到底时对声音最不敏感，很大的声音都不会让机器人启动运动；而逆时针旋转到底时对声音最敏感，非常微弱的声音（包括轮子转动的声音）都会让机器人持续运动下去；因此，需要把电位器调整到一个合适的位置，让机器人能被一定强度的声音启动，而又不会被轮子转动的声音影响。

二、声控机器人应用开发

1. 实验要求

当检测到声音时，机器人启动（图 3.4）：前进 1 s，左转 3 s，后退 1 s，速度 PWM 值为 100，可谓“闻鸡起舞”。

2. 实验器件

机器人套件 1 套、USB 数据线 1 根、声音传感器模块 1 个、导线若干。

3. 电路搭建

安装声音传感器：用 1 个 M4×6 螺钉、1 个 M4×6 塑料螺柱将声音传感器固定到机器人的正前方（图 3.5）。

图 3.4　机器人声音控制示意图

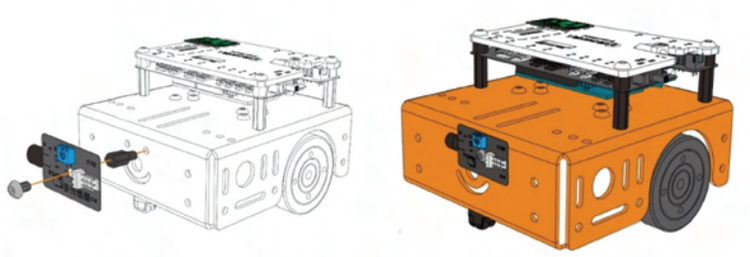


图 3.5　安装声音传感器模块

传感器电路接线：左电动机连接 ~D10、D12，也就是扩展板上的 P3 端口；右电动机连接 ~D6、D7，也就是扩展板上的 P6 端口；声音传感器模块接在 GND、5 V、D2，也就是扩展板上的 P9 端口（图 3.6）。

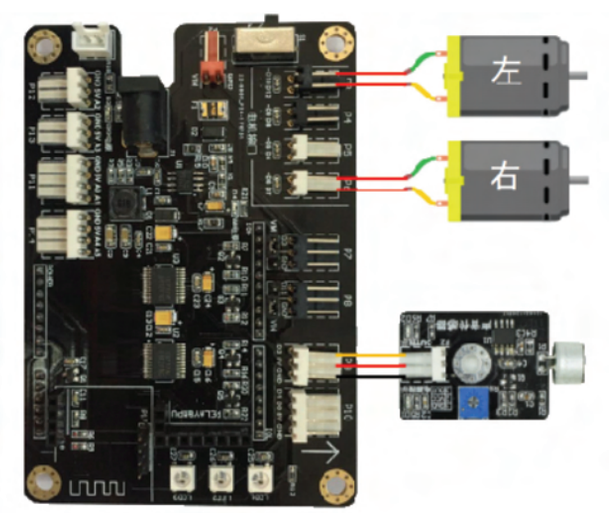


图 3.6　声音控制实验电路接线图

4. 程序流程分析

根据实验要求可知机器人运动状态可分为两种：周边无声音时保持静止状态，周边有声音时触发运动指令。所以，使用 if-else 语句可实现机器人在两种情况下的运动。

做中学

请同学们根据上述分析完成程序流程图 3.7。

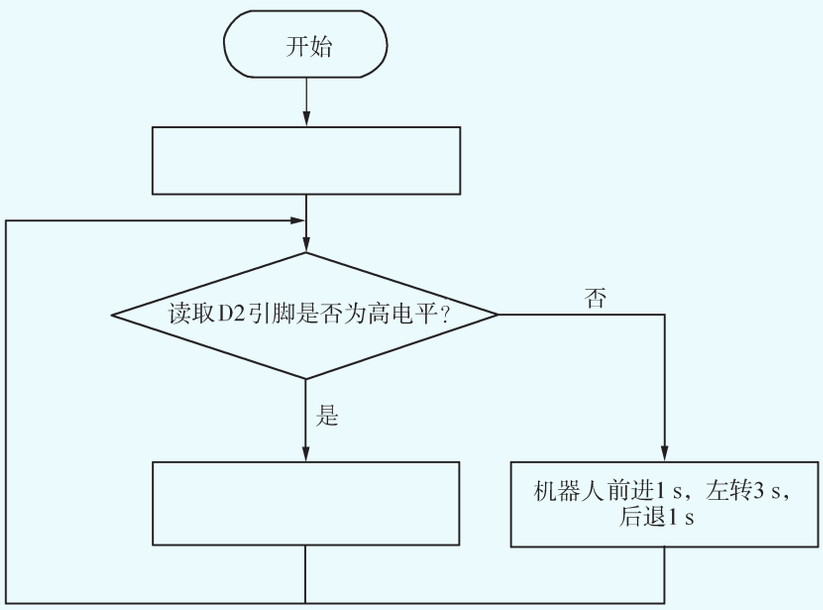


图 3.7　机器人声音控制的程序流程图

5. 根据流程图编写程序

第 1 步：在 setup( ) 函数和 loop( ) 函数外，编写好控制机器人运动的 Drive( ) 函数。

第 2 步：在 setup( ) 函数中初始化引脚状态，调用 pinMode( ) 函数定义 D2 引脚为输入模式。

第3步：　使用if–else语句。digitalRead(D2)作为条件，为1时（无声），机器人不动；为0时（有声），机器人启动。其中，机器人的运动状态通过调用Drive( )函数来实现。

做中学

请同学们根据上述分析自行完成程序的编写。

6. 运行程序

用 USB 数据线将 Uno 板与 PC 机相连接，选择正确的开发板、端口号后，对代码进行编译，编译成功后将代码上传到 Uno 板上。上传成功后，拔掉数据线并将扩展板的电源开关打开，测试机器人的运动状态。

探究与交流

请同学们寻找一下，看看生活中还有哪些物品用到声音传感器。

第三节　光敏传感器的原理及应用开发

学习目标

1. 掌握光敏传感器的工作原理和使用方法。
2. 学会编写光敏传感器的控制程序。
3. 能够使用光敏传感器完成光控机器人的控制任务。

一、光敏传感器简介

光敏二极管，又叫光电二极管，是将光信号变成电信号的半导体器件，它是光敏传感器的重要组成部分。它可以利用光照强弱来改变电路中的电流大小：无光照时，光敏二极管截止；受到光照时，形成光电流，并且随着入射光照强度的增大而变大。

在实验中，我们使用的是带有光敏二极管的光敏传感器模块，它有 4 个接线端口，如图 3.8 所示。

GND：接地端。

VCC：电源端，接入 5 V 电压。

SIG：信号输出端口（输出 0~5 V 电压值），通过 Uno 板模拟输入引脚后可得到离散的数字信号（0~1 023）。光照强度越大，输出的数字信号值越大。

NC：该引脚不连接。

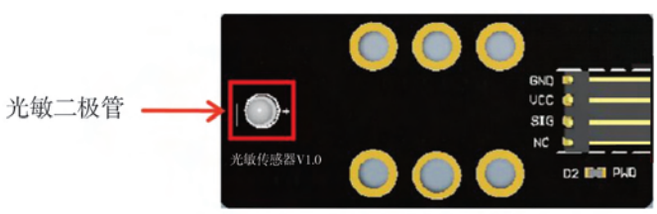


图 3.8　光敏传感器模块的接线端口

二、光控机器人应用开发

1. 实验要求

利用光敏传感器可以测量出光照的强弱，为机器人加上一对光敏传感器，使它向着有光照的地方前进，如图 3.9 所示。当机器人右边有光照时，则

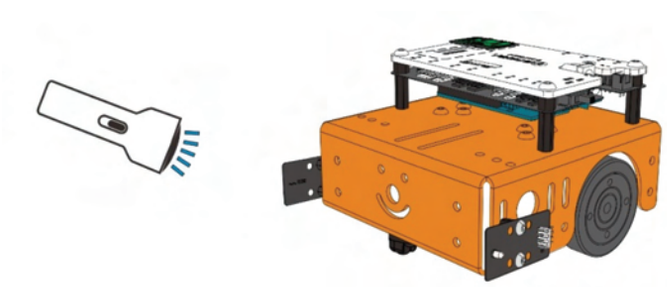


图 3.9　机器人光线控制示意图

右转；当左边有光照时，则左转；当左右两侧都有光照时，则直行；当左右两侧都没有光照时则停止。机器人会追着光线跑，犹如追日的“夸父”。

2. 实验器件

机器人套件 1 套、USB 数据线 1 根、光敏传感器模块 2 个、导线若干。

3. 电路搭建

用 4 个 M4 × 6 螺钉、4 个 M4 × 6 塑料螺柱、4 个 M4 螺母，将传感器固定到机器人的两侧（图 3.10）。

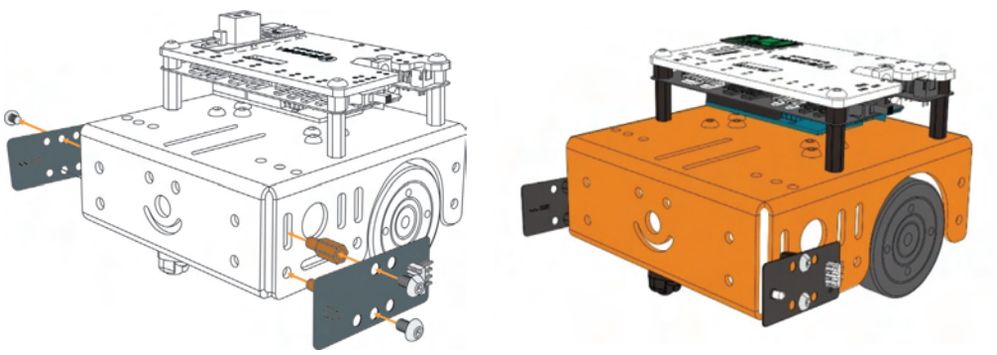


图 3.10　安装光敏传感器模块

传感器电路接线：左边电动机连接 ~D10、D12，也就是扩展板上的 P3 端口；右边电动机连接 ~D6、D7，也就是扩展板上的 P6 端口；左侧光敏传感器接在扩展板上的 P11 端口，其输出端口 SIG 连接 A0；右侧光敏传感器接在扩展板上的 P14 端口，其输出端口 SIG 连接 A4（图 3.11）。

4. 程序流程分析

依据实验要求分析得出 3 种情况：两侧都有光，机器人前进 Drive(1,100)；只有左侧有光，机器人左转 Drive(3,100)；只有右侧有光，机器人右转 Drive(4,100)；如以上 3 种情况都不满足，机器人停止 Drive(1,0)。所以我们使用 if–else if 语句区分 3 种情况。

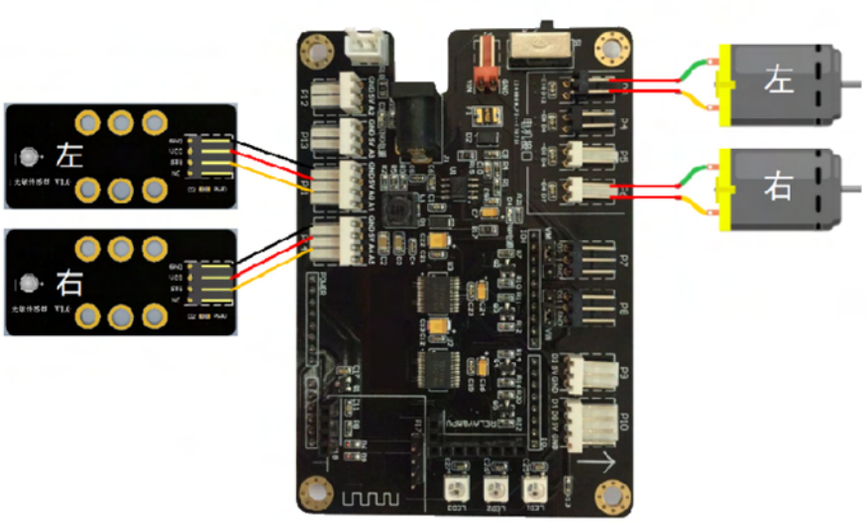


图 3.11　光线控制实验电路接线图

做中学

请同学们根据上述分析完成程序流程图 3.12。

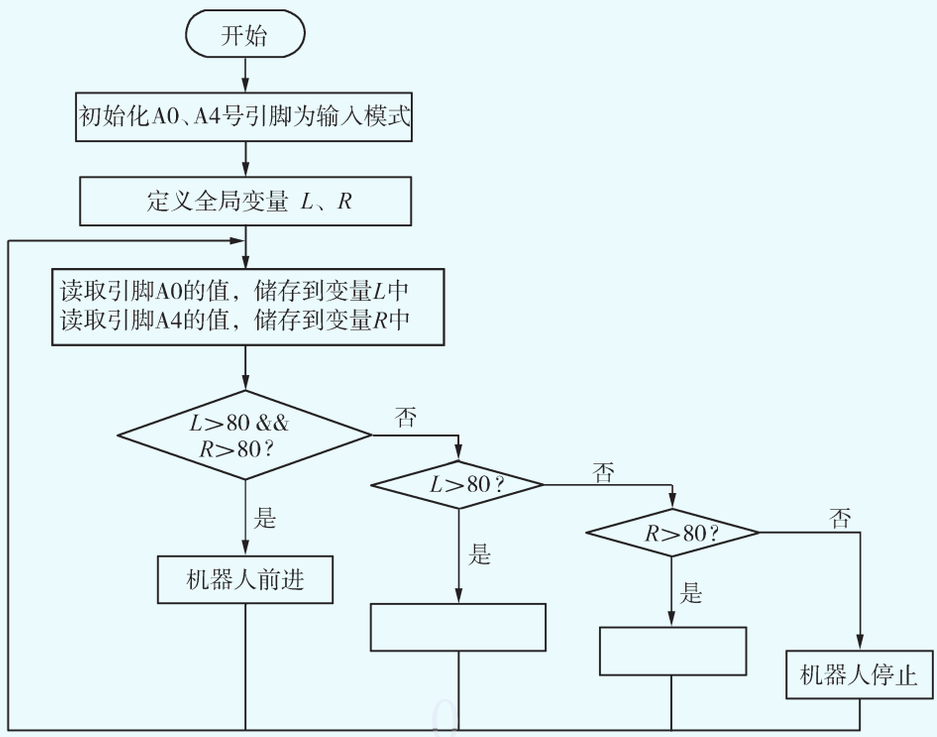


图 3.12　机器人光线控制的程序流程图

5. 根据流程图手编写程序

第 1 步：在 setup( ) 函数和 loop( ) 函数外，编写好控制机器人运动的 Drive( ) 函数。

第 2 步：定义两个全局变量，类型为整型，变量名分别为 *L* 和 *R*。

第 3 步：在 setup( ) 函数中初始化引脚状态，调用 pinMode( ) 函数定义 16 号和 17 号引脚为输入模式。

第 4 步：在 loop( ) 函数中编写主程序。调用 analogRead( ) 函数读取 A0、A4 引脚的状态，分别存入变量 *L* 和变量 *R* 中。

第 5 步：使用 if-else if 语句，如果（if）两个光敏传感器都检测到光线，那么机器人前进；如果（else if）只有左侧光敏传感器检测到光线，那么机器人左转；如果（else if）只有右侧光敏传感器检测到光线，那么机器人右转；否则（else）没有光敏传感器检测到光线，机器人停止运动。其中，机器人的运动状态通过调用 Drive( ) 函数来实现。

探究与交流

请同学们根据上述分析自行完成程序的编写。

6. 运行程序

用 USB 数据线将 Uno 板与 PC 机相连接，选择正确的开发板、端口号后，对代码进行编译，编译成功后将代码上传到 Uno 板上。上传成功后，拔掉数据线并将扩展板的电源开关打开，观察机器人运动状态。如果发现逐光效果不好，可以适当更改程序中 L 和 R 的比较值。

探究与交流

请同学们寻找一下，看看生活中还有哪些物品用到光敏传感器。

第四节　红外线传感器的原理及应用开发

学习目标

1. 掌握红外线传感器的工作原理和使用方法。
2. 学会编写红外线传感器的控制程序。
3. 能够使用红外线传感器完成机器人探路的控制任务。

一、红外线传感器简介

红外线有反射特性，而且对于不同颜色的物体有不同的反射率：深颜色对光的吸收比较强，所以反射比较弱；浅颜色对光的吸收比较弱，所以反射比较强。大家在冬天穿深颜色的衣服会感觉比较暖和，而在夏天穿浅颜色的衣服会感觉比较凉爽，就是利用这个特性。红外传感器就是以红外线为介质来帮助检测物体的传感器。当地面上有颜色分明的路径时，利用这个红外传感器可以控制物体巡线运动。

机器人上的巡线传感器模块工作电压为 5 V，集成了两路 TCRT5000 红外线传感器（图 3.13），检测反射距离为 1~25 mm，输出形式为数字开关量 0 和 1。当巡线传感器模块连接机器人扩展板的数字接口时，红外线传感器的发射端向外不断发射红外线。当检测为黑色时，发射出的红外线没有被反射回来或被反射回来但强度不够大，此时模块的输出端为高

电平；当检测为白色时，发射出的红外线被反射回来且强度足够大，此时模块的输出端为低电平。

巡线传感器模块有 4 个接线端口（图 3.14），分别为：

GND：接地端。

VCC：电源端，接入 5 V 电压。

DR：红外传感器 1 的信号输出引脚。

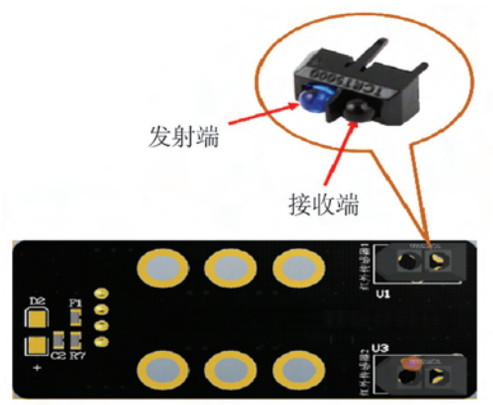


图 3.13　TCRT5000 巡线传感器模块

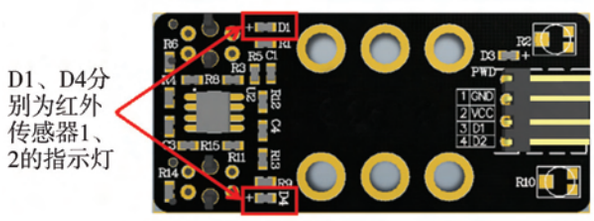


图 3.14　巡线传感器模块的接线端口

DL：红外传感器 2 的信号输出引脚。

二、红外线探路机器人应用开发

1. 实验要求

使机器人在白色场地内运动，白色场地周围有黑色标记线，一旦机器人走出白色场地能够立即后退，然后左转弯并继续行驶。也可以将机器人放在干净的白色桌面上运动，当机器人走到桌子的边缘时，能够立即做出同样的响应动作（图 3.15），防止从桌上掉落，可谓“悬崖勒马”。

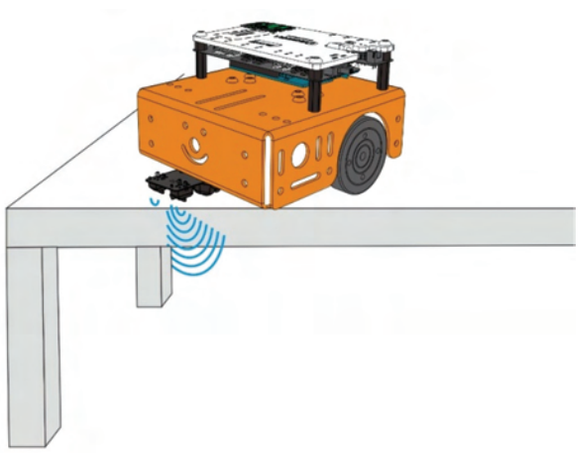


图 3.15　机器人红外线探路实验示意图

2. 实验器件

机器人套件 1 套、USB 数据线 1 根、巡线传感器模块 1 个、导线若干。

3. 电路搭建

用 2 个 M4 × 8 的螺钉将巡线传感器模块固定在机器人的底部与万向轮之间（图 3.16）。

传感器电路接线如图 3.17 所示：左边电动机连接 ~D10、D12，也就是扩展板上的 P3 端口；右边电动机连接 ~D6、D7，也就是扩展板上的 P6 端口；巡线传感器模块接 D1、D0、5 V、GND，也就是扩展板上的 P10 端口。其中，DR 连接 D0，DL 连接 D1。

4. 程序流程分析

本实验中，当两个红外线传感器均输出低电平时，表示机器人全部在白色场地内，可保持运动状态；当有一个红外线传感器输出高电平时，表示其中一个红外线传感器已经驶出了白色场地，需要立即返回；当两个红外线传感器均输出高电平时，表示两个红外线传感器都已经驶出了白色场地，需要立即返回。在这个程序中需要用到前面学到的逻辑运算表达式的相关知识。

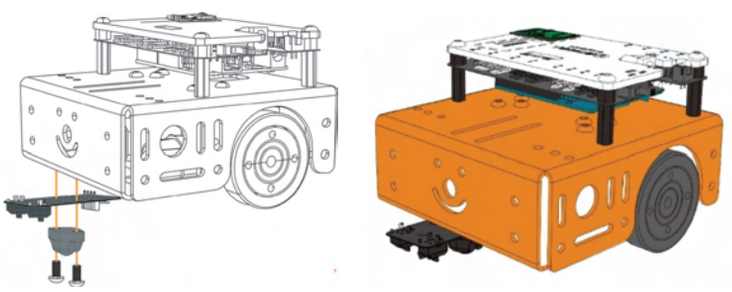


图 3.16　固定巡线传感器模块

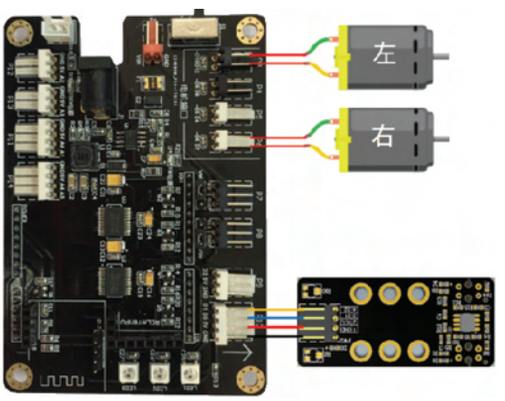


图 3.17　红外线探路实验电路接线图

做中学

请同学们根据上述分析完成程序流程图 3.18。

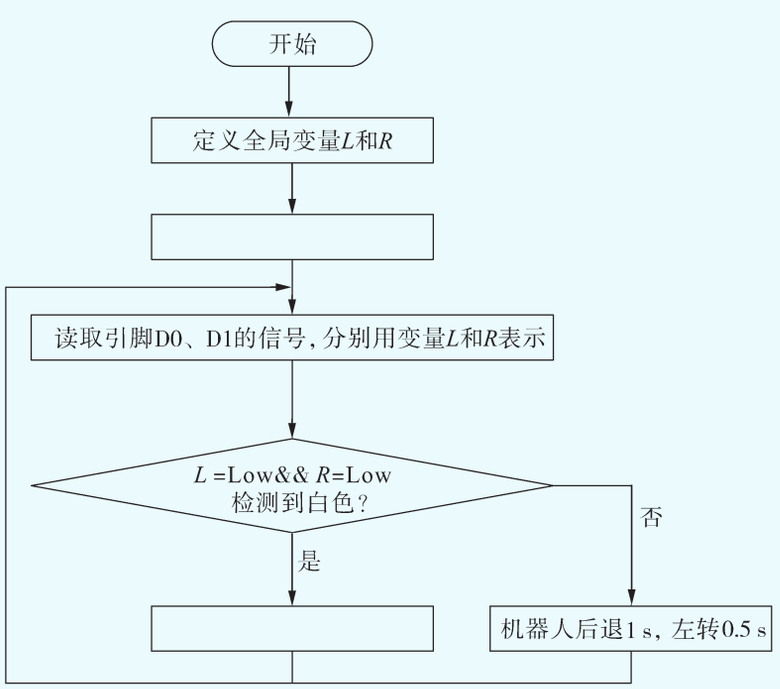


图 3.18　机器人红外线探路的程序流程图

5. 根据流程图，动手编写程序

第 1 步：在 setup( ) 函数和 loop( ) 函数外，编写好控制机器人运动的 Drive() 函数。

第 2 步：定义两个全局变量，类型为整型，变量名分别为 *L* 和 *R*。

第 3 步：在 setup( ) 函数中初始化引脚状态，调用 pinMode() 函数定义 D0 和 D1 引脚为输入模式。

第 4 步：开始在 loop( ) 函数中编写主程序。调用 digitalRead() 函数读取 D0 和 D1 引脚的状态，分别存入变量 *L* 和变量 *R* 中。

第 5 步：使用 if-else 语句，如果（if）两个红外线传感器均检测到白色，那么机器人前进；否则（else）机器人后退 1 s，左转 0.5 s。其中，机器人的运动状态通过调用 Drive( ) 函数来实现。

6. 运行程序

用 USB 数据线将 Uno 板与 PC 机相连接，选择正确的开发板、端口号后，对代码进行编译，编译成功后将代码上传到 Uno 板上。上传成功后，拔掉数据线并将扩展板的电源开关打开，观察机器人运动状态能否达到实验要求。

做中学

请同学们根据上述分析自行完成程序的编写。

第五节　超声波传感器的原理及应用开发

学习目标

1. 掌握超声波传感器的工作原理。
2. 学会编写超声波传感器的控制程序。
3. 能够使用超声波传感器完成机器人避障的控制任务。

一、超声波传感器简介

超声波是一种频率高于 20 000 Hz 的声波，因其频率的下限大于人听觉的上限而得名。蝙蝠在飞行时，会发出一种超声波，这些超声波碰到其他物体就会立刻反射回来，根据回声到来的方位和时间，蝙蝠在振翅之间就可以完成听、看、计算与绕开障碍物的全部过程。人类根据蝙蝠飞行识物的原理，制造出了雷达。

机器人上的超声波传感器模块工作电压为 5 V，感应角度不大于 15°，探测距离为 2~450 cm，精度可以达到 0.3 cm。超声波传感器模块由 4 个接线端口和两个探头组成（图 3.19），分别为：

VCC：电源端，接入 5 V 电压。

Trig：控制端，该引脚内部有 10 kΩ 的上拉电阻，用 Uno 板的 I/O 端口拉低 Trig 引脚，然后给一个 10 μs 以上的脉冲信号便可触发该传感器模块发射超声波信号。

Echo：接收端，当发射探头开始发射脉冲的时候输出高电平，当接收探头接收到反射回来的超声波信号时，开始输出低电平。通过计算该引脚保持高电平的时间 T，可以得到目标物体的距离 d=340 · T（m）。

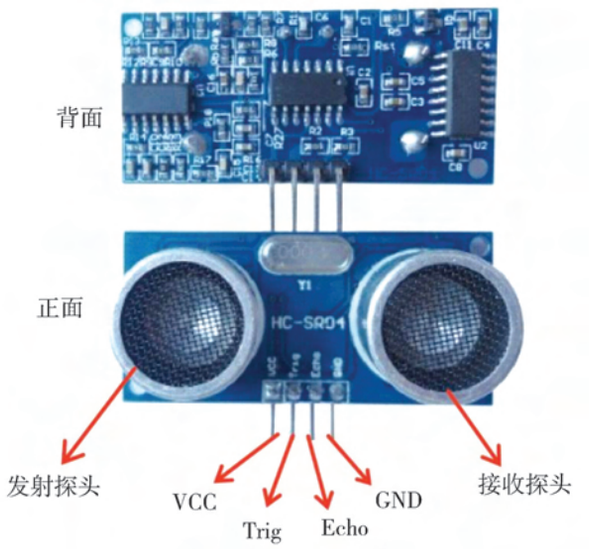


图 3.19　超声波传感器模块的接线端口

GND：接地端。

发射探头：用于发送超声波信号，当测距功能被触发后，发射探头将自动发送 8 个 40 kHz 的超声波脉冲。

接收探头：用于接收超声波信号。

二、超声波避障机器人应用开发

1. 实验要求

当机器人在运动过程中遇到前方有障碍物（图 3.20），可以立即返回并左转来躲避障碍物，可谓“迷途知返”。

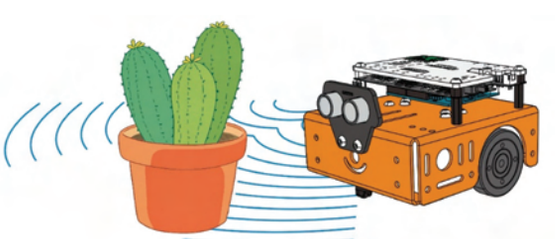


图 3.20　机器人避障示意图

2. 实验器件

机器人套件 1 套、USB 数据线 1 根、超声波传感器模块 1 个、导线若干。

3. 电路搭建

用 2 个 M4 × 6 螺钉将超声波传感器模块固定到机器人的正前方（图 3.21）。

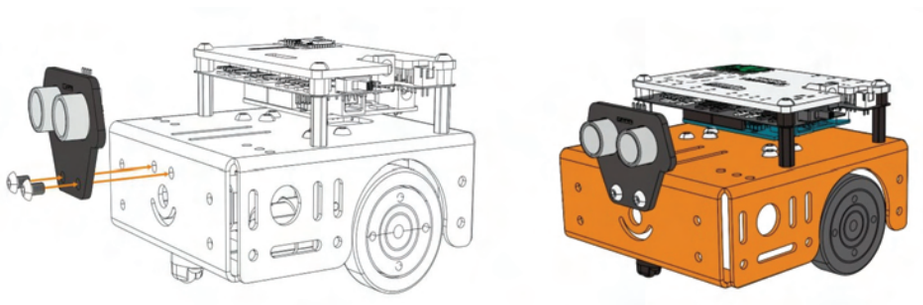


图 3.21　安装超声波传感器模块

超声波传感器电路接线：左边电动机连接 ~D10、D12，也就是扩展板上的 P3 端口；右边电动机连接 ~D6、D7，也就是扩展板上的 P6 端口；超声波传感器模块连接 D1、D0、5 V、GND，也就是扩展板上的 P10 端口。其中 Echo 连接 D0，Trig 连接 D1（图 3.22）。由于超声波传感器模块 4 个引脚的线序与扩展板上 P10 端口的线序不一致，因此机器人套件里的超声波传感器模块搭配了一块转接电路板。

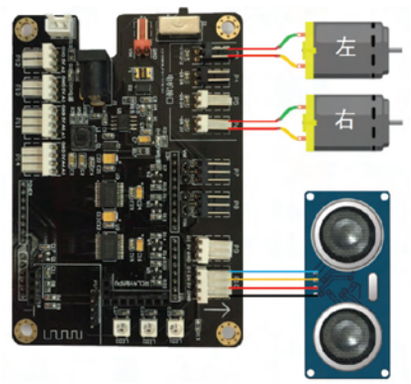


图 3.22　超声波避障实验电路接线图

4. 程序流程分析

超声波传感器模块测得前方障碍物的距离值，由此作为判断条件。当与障碍物的距离值小于 10 cm 时，机器人左转来躲避障碍物。为了减小机器人左转碰到障碍物的概率，我们在左转前加入后退的指令。当与前方障碍物的距离值大于 10 cm 时，机器人可以继续前进。

（1）delayMicroseconds()：延时函数，单位为微秒（μs）。

示例：delayMicroseconds(10); // 延时 10 μs。

（2）pulseIn(pin,value)：读取引脚为某个状态（高电平或低电平）的持续时间，单位为微秒（μs）。pin: 引脚编号。value:LOW（低电平）或 HIGH（高电平）。

做中学

请同学们根据上述分析完成程序流程图 3.23。

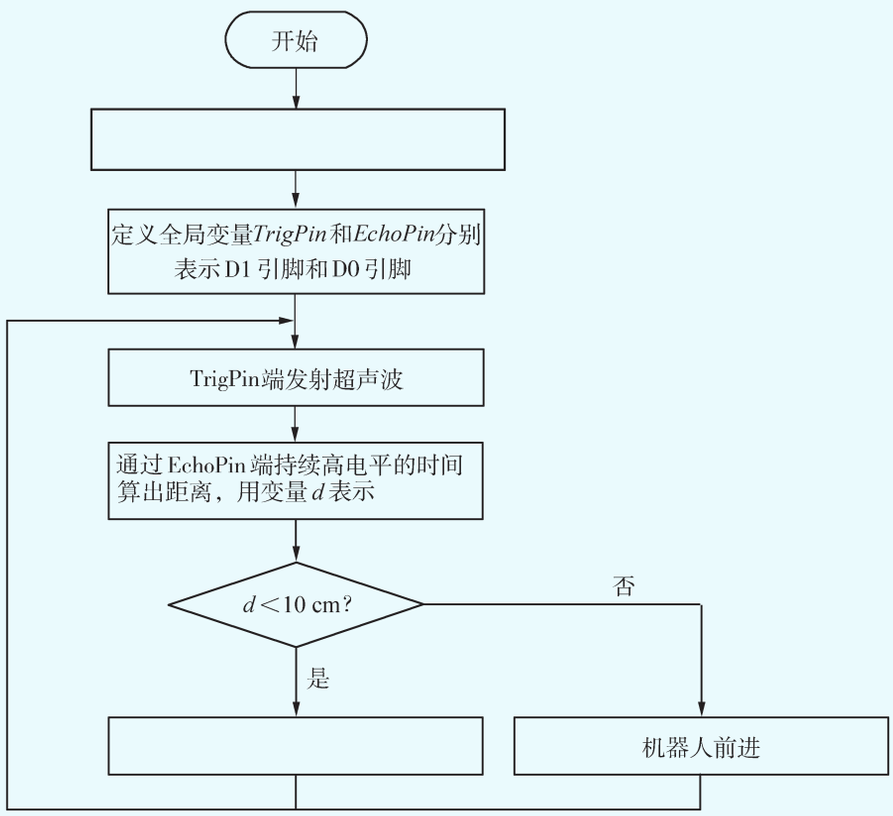


图 3.23　机器人实现避障的程序流程图

5. 根据流程图，动手编写程序

第 1 步：在 setup( ) 函数和 loop( ) 函数外，编写好控制机器人运动的 Drive( ) 函数。

第 2 步：定义全局变量 *TrigPin* 为 1，定义全局变量 *EchoPin* 为 0。

第 3 步：在 setup( ) 函数中初始化引脚状态，设置 D1 引脚为输出模式，D0 引脚为输入模式。

第 4 步：在 loop( ) 函数中编写主程序，触发超声波传感器模块发送超声波信号。首先通过 Uno 板向 TrigPin 引脚输入持续 2 μs 低电平（调用 delayMicroseconds( ) 函数实现），接着给 TrigPin 引脚输入持续 10 μs 的高电平，再给 TrigPin 引脚输入低电平。

第 5 步：调用 pulseIn( ) 函数测得 EchoPin 引脚持续输出高电平的时间，用变量 T 表示，再计算出障碍物的距离 $d=T/58$。

第 6 步：使用 if 语句来实现判断，如果距障碍物的距离不大于 10 cm，则机器人后退 1 s，左转 1 s；否则，如果距障碍物的距离大于 10 cm，则机器人前进。其中，机器人的运动状态调用 Drive( ) 函数来实现。

做中学

请同学们根据上述分析自行完成程序的编写。提示：在编写程序时，可能会用到以下函数。

示例：pulseIn(0,HIGH); // 读取 D0 引脚持续高电平的时间。

6. 运行程序

用 USB 数据线将 Uno 板与 PC 机相连接，选择正确的开发板、端口号后，对代码进行编译。编译成功后，在上传程序前需要确认超声波连接线与扩展板未连接，随后将代码上传到 Uno 板上。上传成功后，拔掉数据线，将超声波连接线与扩展板接口连接，并将扩展板的电源开关打开，观察机器人的运动状态。

注：超声波传感器模块的 Echo 端口连接数字引脚 D0，Trig 端口连接数字引脚 D1，而数字引脚 D0 和数字引脚 D1 是复用引脚，除了作为数字引脚，还作为通信接口。数字引脚 D0 的复用功能是 RX（Receive），即为 Arduino 接收来自计算机的指令信息；数字引脚 D1 的复用功能是 TX（Transmitter），即为 Arduino 发送指令信息给计算机。当上传程序到 Uno 板时，就使用了两引脚的复用功能，因此，为了保证程序顺利上传到 Uno 板上，需要在上传程序前拔掉与超声波连接的线，上传成功后再重新连接。

探究与交流

如果发现避障效果不好，我们可以怎么优化程序？

本章小结

机器人只有具备了感知能力才能够像智能生物一样根据环境的变化做出相应的反应。给机器人装上具有不同功能的传感器，机器人就能获得视觉、听觉、嗅觉和触觉等各种“感觉”。本章重点介绍了简易机器人中常用到的声音传感器、光敏传感器、红外线传线感器和超声波传感器，并分别利用这些传感器完成不同功能要求的机器人的应用开发。

学习评价

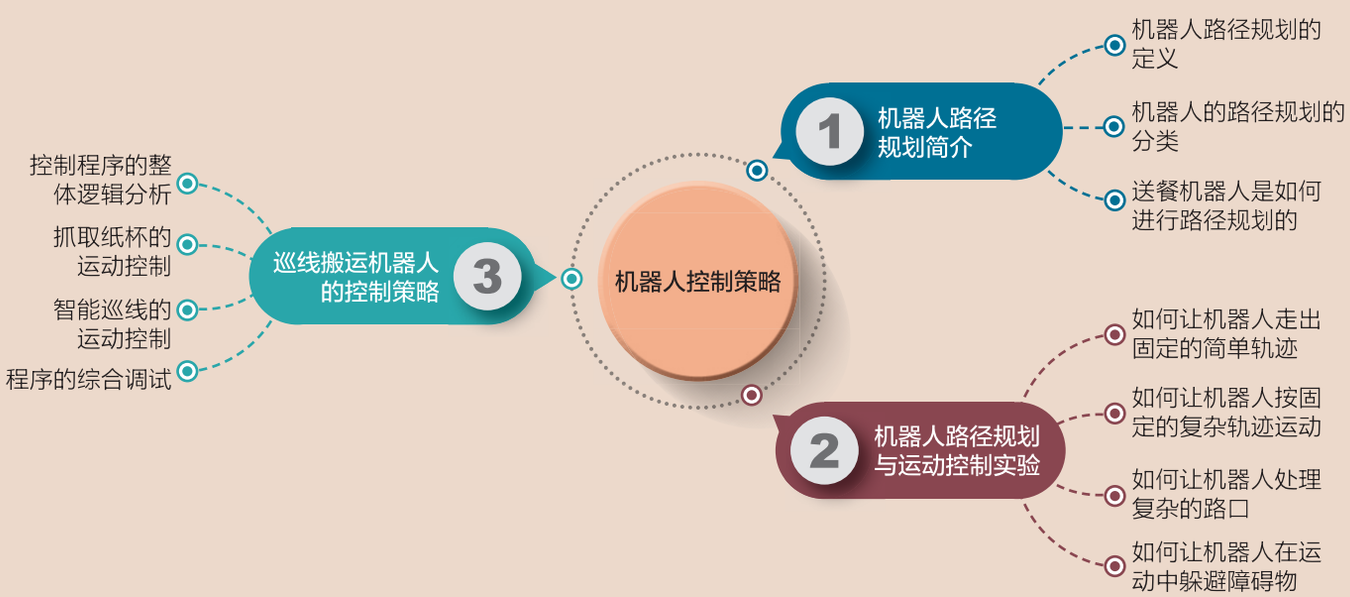
评价内容			评价方式		
			自我评价	小组评价	教师评价
过程评价	师生互动	能积极思考老师提出的问题			
		能基于已有经验构建新的知识			
		能积极参与课堂讨论			
	实践活动	能积极参与实践活动			
		与小组成员有效合作			
		完成声控机器人的任务			
		完成光控机器人的任务			
		完成机器人探路的任务			
		完成超声波避障机器人的任务			
结果评价	目标实现	了解常用传感器的工作原理			
		掌握声音传感器的使用方法			
		掌握光敏传感器的使用方法			
		掌握红外线传感器的使用方法			
		掌握超声波传感器的使用方法			
	收获反思	收获与感悟			
		反思不足			

第四章
机器人控制策略

导　言

同学们试想一个场景：现在家里有台能够在不同房间来回移动的服务机器人，当我们在客厅看电视的时候需要机器人去厨房帮忙取果盘，机器人是如何走到厨房的？如果路径上有拖鞋、宠物狗等障碍物，机器人是如何响应的？取到果盘之后，机器人又是如何返回客厅的？这些涉及的就是机器人的路径规划问题。扫地机器人、割草机器人、仓储机器人、送餐机器人等也会遇到类似的问题。本章我们将学习机器人的路径规划。

思维导图



第一节　机器人路径规划简介

学习目标

1. 了解机器人路径规划的定义。
2. 了解机器人路径规划的分类。
3. 了解送餐机器人的路径规划方法。

一、机器人路径规划的定义

路径规划技术是机器人研究领域的一个重要分支。机器人的最优路径规划问题，就是依据某个或某些优化准则（如工作代价最小、行走路径最短、行走时间最短等），在工作空间中找到一个从起始状态到目标状态能避开障碍物的最优路径。简单地说，应注意以下三点：明确起点和终点，躲避障碍物，优化路径。

通常，机器人路径规划需要解决四个问题：

（1）机器人如何从环境中获取周围的障碍物信息和其他相关信息。

（2）机器人如何根据内部及外部传感器来确定当前所处的位置。

（3）机器人如何根据其当前的位置和当前位置信息确定行动策略。

（4）如何产生合适的驱动信号使机器人运动在预定的轨迹上。

二、机器人路径规划的分类

机器人的路径规划根据其目的不同可以分为两种：一种是传统的点到点的路径规划，另一种是完全遍历路径规划。

点到点的路径规划：一种从起始点到终点的运动策略，它要求寻找一条从起点到终点的最优路径，使机器人能够在工作空间顺利地通行。例如，酒店送餐机器人、仓储机器人和自动导航物流小车等用的就是这种路径规划。

完全遍历路径规划：指在满足某种性能指标最优的前提下，寻找一条在指定区域内从起点到终点且经过所有可达到点的连续路径。如军事用的排雷机器人、家居及办公环境用的扫地机器人等都会用到完全遍历路径规划。

机器人的路径规划根据其对环境信息掌握程度的不同可以分为两种：一种是全局路径规划，另一种就是局部路径规划。

全局路径规划：一种环境信息完全已知情况下的路径规划，又称静态或离线路径规划。

局部路径规划：一种环境信息完全未知或部分未知情况下的路径规划，通过传感器对

机器人的工作环境进行探测，以获取障碍物的位置、形状和尺寸等信息，又称动态或在线路径规划。

三、送餐机器人的路径规划

图 4.1　送餐机器人

机器人路径规划的应用场景很多，送餐机器人（图 4.1）就是一个典型代表。它具有自动送餐、空盘回收、菜品介绍、自动充电等实用功能，能够代替或者部分代替餐厅服务员为顾客服务，能够减轻餐厅服务员的工作强度，为传统的餐饮服务行业注入新鲜的科技元素。

（一）工作任务分析

当客人需要点菜时，送餐机器人自动走到客户餐桌旁，客户可以通过机器人上的显示面板进行点餐，并通过语音交互让机器人对菜品进行介绍。当客人点完餐后，送餐机器人可以走到取餐处，由工作人员将菜品放到机器人的托盘上，然后自主移动回到对应餐桌旁，语音提示客户自取菜品，然后根据任务安排自行返回等待点或充电站。

（二）路径规划方法

送餐机器人可以在工作人员的操作下进行餐厅地图的创建与编辑，设置一系列的餐桌目标点。然后通过红外传感器巡地上黑线的方式进行导航。因为黑线能够吸收红外线，而普通白色地面不会吸收，因此可以通过判断不同红外传感器的反馈值来判断机器人相对黑线的运动位置。当红外传感器都在黑线上时，机器人保持前进；当红外传感器有一侧偏离黑线时，机器人及时向左或向右纠偏回正轨；当红外传感器都不在黑线上时，机器人后退寻找黑线。

此外，送餐机器人还要具备躲避路径中的障碍物的功能，因此需要用避障传感器来实时检测前方的障碍物，当有障碍物出现，能够及时停止运动，并语音播报避让信息，等障碍物消失后再继续前进。

目前，也有采用基于激光雷达的即时定位与地图构建技术来实现路径规划的送餐机器人，实现了无须铺设特定轨道情况下的路径规划。

探究与交流

请同学们查阅相关资料，介绍一下扫地机器人的路径规划方法。

第二节　机器人路径规划与运动控制实验

学习目标

1. 掌握机器人巡线的路径规划方法。
2. 掌握机器人特殊路口的处理方法。

在本节中，我们通过几种情况下机器人的运动路径编程实验，逐步掌握机器人路径规划方法。

一、基础实验：机器人走正方形

1. 任务说明

在不使用传感器的前提下，控制机器人沿着正方形地图（图 4.2）的边一直循环行走下去，正方形地图的边长为 50 cm。

图 4.2　正方形地图

2. 方案分析

首先按照地图规划机器人的行进路径，本实验的地图为正方形，机器人只需要两步动作： 第一步， 前进一个边长的距离；第二步，原地右转90° 。之后一直循环执行这两步即可。

3. 编程实现

我们通过调用 Drive() 和 delay() 函数来实现对机器人转动角度和位移的运动控制。调试中最关键的参数是前进速度、前进持续时间、右转速度、右转持续时间，在实验中需要依照地图对参数进行合理的调整。

做中学

部分程序清单如下所示，请同学们根据实验结果完善参数。

控制机器人运动的 Drive() 函数由同学们自行完成，参考上章节内容。

```
void loop()
{
Drive(1,_____ );          // 设置机器人前进速度
delay(_____ );            // 设置前进时间
Drive(4,_____ );          // 设置机器人右转速度
delay(_____ );            // 设置右转时间
}
```

二、基础实验：机器人巡黑线

1. 任务说明

利用红外线传感器，实现机器人巡黑色曲线的任务，黑线宽度为 10 mm，小于两红外线传感器之间的距离。

2. 方案分析

本方案采用两个红外线传感器跨在黑色线的两侧进行巡线，机器人在行驶过程中会出现向左偏离或者向右偏离的情况，遇到不同的情况，控制机器人执行不同的动作：当右边传感器检测到黑线说明机器人运动偏左了，需要控制机器人右转；当左侧传感器检测到黑线说明机器人运动偏右了，需要控制机器人左转；其余情况则直行。图4.3用两个蓝色圆点表示传感器位置。

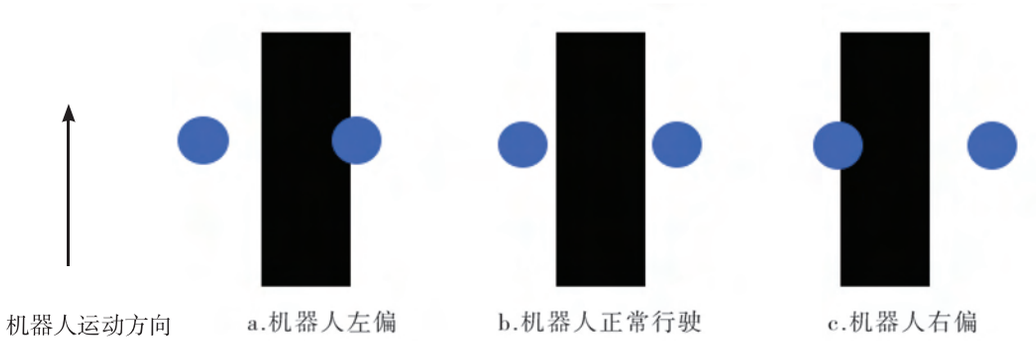


图 4.3　机器人巡线方案分析

做中学

请同学们根据上述分析完成下面的程序流程图（图 4.4）。

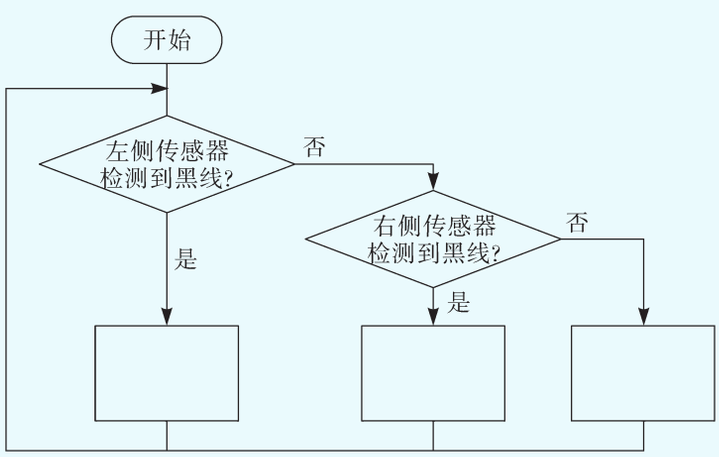


图 4.4　机器人巡黑线的程序流程图

3. 编程实现

我们根据程序流程图，动手编写程序，部分程序清单如下所示。

控制机器人运动的 Drive() 函数由同学们自行完成，参考上章节内容。

```
/*************** 定义全局变量，初始化引脚设置 ***************/
int Line_left = 1;              // 用变量 Line_left 表示左侧红外传感器输入端口
```

```
int Line_Right = 0;              //用变量 Line_Right 表示右侧红外线传感器输入端口
int R = 0;                       //定义变量 R，用于保存右侧传感器的状态值
int L = 0;                       //定义变量 L，用于保存左侧传感器的状态值
void setup( )
{
pinMode(Line_left,INPUT);        //设置左侧传感器为输入状态
pinMode(Line_Right,INPUT);       //设置右侧传感器为输入状态
}
void loop( )
{
R = digitalRead(Line_Right);     //读取右侧传感器输入引脚的状态，并赋值给 R
L = digitalRead(Line_left);      //读取左侧传感器输入引脚的状态，并赋值给 L
if (L == 1 &&R == 0)
    Drive(3, 100);               //如果左侧传感器检测到黑线，机器人向左转
 else if (L == 0 &&R == 1)
    Drive(4, 100);               //如果右侧传感器检测到黑线，机器人向右转
  else
    Drive(1, 100);               //如果两侧传感器均未检测到黑线，机器人前进
}
```

由于红外线传感器连接到了数字引脚 D0 和 D1，为了保证程序顺利上传到 Uno 板上，需要在上传程序前拔掉与红外线传感器连接的线，上传成功后再重新连接。

三、基础实验：巡线中特殊路口的处理

1. 任务说明

要求机器人在巡线的过程中按照规划好的路径顺利通过特殊路口（如图 4.5）。

图 4.5　巡线中常见的特殊路口

2. 方案分析

在特殊路口情况下，我们一般需要从红外线传感器的数量和安装位置上做文章。建议使用 4~6 路红外线传感器，并尽量将传感器安装在机器人底盘前方位置或者从车头探出。

假设路径要求在 T 形路口左转，我们可以做如下处理：使用 4 路红外线传感器，从左至右依次标号为 S1、S2、S3、S4，中间 2 路传感器 S2、S3 用于巡线。

第 1 步：正常巡线（图 4.6）的控制方法在前面的实验中已经讲过。

第 2 步：当 S1 或者 S4 检测到黑线的时候，表明遇到了 T 形路口（图 4.7），关闭巡线功能。为了给机器人留出左转的空间，我们要求机器人再前进一段距离，达到如图 4.8 所示的位置，具体前进多少，需要在具体实验中调节。

第 3 步：机器人开始左转，直到 S2 检测到黑线才停止左转，继续开启巡线（图 4.9）。

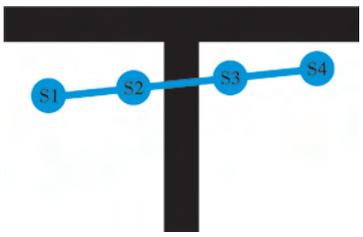


图 4.6　正常巡线

图 4.7　检测到 T 形路口

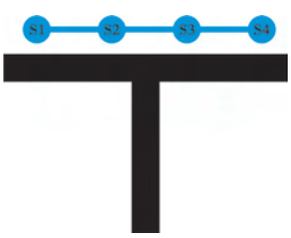


图 4.8　继续前进一段距离

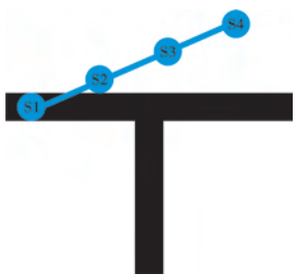


图 4.9　机器人左转

做中学

请同学们根据上面的分析，完善下面的程序流程图 4.10。

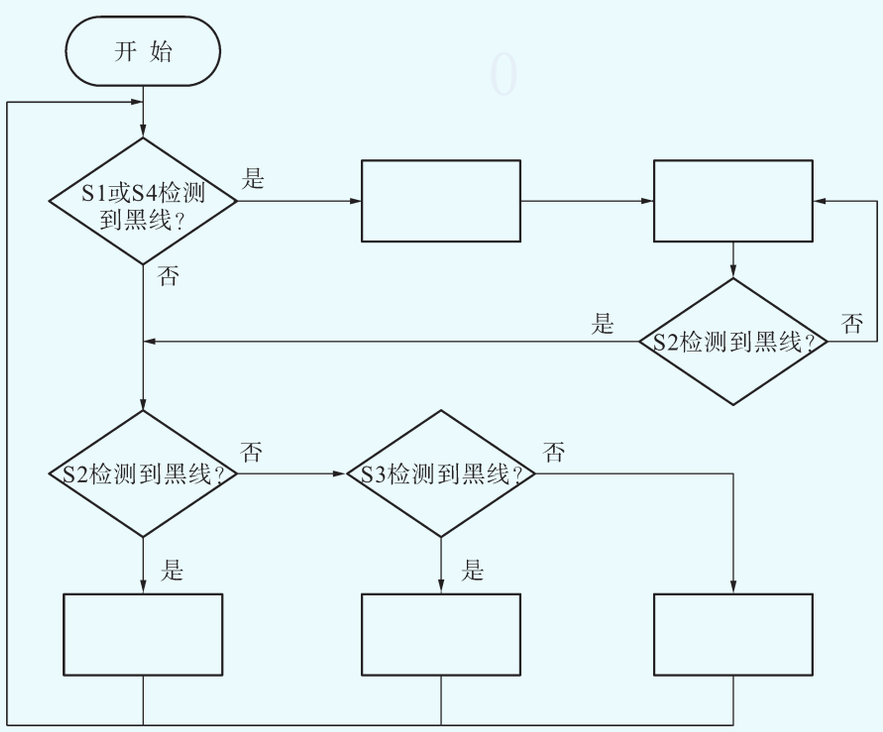


图 4.10　机器人 T 形路口运动的程序流程图

请同学们记录能顺利通过 T 形路口的运动控制参数，并完善下面的程序。

```
/****************************** 主函数 ******************************/
void loop()
{
```

```
if(digitalRead(S1)==1||digitalRead(S4)==1)
turn_Left();        // 当检测到 T 形路口的情况下调用左转弯函数
else
run_Forward();      // 没检测到 T 形路口的情况下调用正常巡线函数
}
/*************************** 左转弯函数 ***************************/
void turn_Left()
{
Drive(1, __________);        // 设置机器人前进速度
delay(__________ );          // 设置机器人前进时间
while (1)                    // 原地左转直到 S2 检测到黑线，然后继续巡线
{
Drive(3, __________ );       // 设置机器人左转速度
if (digitalRead(S2) == 1)
break ;                      // 如果 S2 检测到黑线，跳出 while 循环
     }
}
/*************************** 正常巡线函数 ***************************/
void run_Forward()
{
巡线程序块 ;                 // 正常巡线函数
}
```

第三节　巡线搬运机器人的控制策略

学习目标

1. 掌握自顶向下的程序设计方法。
2. 能够控制机器人通过交叉路口。
3. 能够控制机械手实现纸杯的抓取。

在第一章中我们提出了巡线搬运机器人的设计任务，并完成了移动载体跟搬运机械手的设计制作。在本节中，我们将综合前面学过的 Arduino 控制器编程、传感器检测、机器人路径规划等知识，完成巡线搬运机器人控制系统的开发。

在前面完成的机械结构基础上，我们根据巡线搬运机器人的具体任务要求，对其控制系统设计方案进行分析与讨论（表 4.1）。

表 4.1　巡线搬运机器人控制系统设计方案的讨论

功能需求	考虑因素	技术细节	具体方案
巡黑线运动	如何不脱线	当机器人运动方向偏左的时候，机器人向右调整方向 当机器人运动方向偏右的时候，机器人向左调整方向	使用 4 个红外线传感器来检测黑线，红外线传感器探头之间的距离大于黑线的宽度。若左侧传感器检测到黑线证明机器人运动方向偏右，若右侧传感器检测到黑线证明机器人运动方向偏左
交叉口检测	如何选择正确的分叉路	当不夹持纸杯的时候，机器人沿着外侧大六边形巡线运动 当夹持到纸杯的时候，机器人遇到交叉口后选择通往仓库的分叉路运动	使用标志位来区分是否夹持纸杯，标志位 0 表示未夹持纸杯，标志位 1 表示已经夹持纸杯。每次识别到交叉口后根据标志位决定选择哪条分叉路
搬运物检测	检测的距离	如何在距离纸杯 100 mm 以上就可以检测到纸杯	使用超声波传感器，不仅能检测到是否有障碍物，还能检测到与障碍物的距离
机械手控制	摆臂角度 运动速度 旋转扭矩	机械手臂放下与抬起的角度不超过90° 机械手爪打开与合上的角度不超过90° 搬运物体为纸杯，因此对控制手爪的驱动器的力矩要求不高 机械手臂的驱动器需要带动手臂连杆运动，因此力矩要求会比手爪的高	使用舵机作为驱动器，能够满足精确的小角度控制，机械手爪用小舵机带动，机械手臂用大舵机带动

事实上，除了上面给出的方案，还有其他解决问题的途径。同学们可以随着自己经验的积累与解决问题能力的提高，不断寻找新的控制系统设计方案，也可以对上述方案提出改进的意见。

一、控制程序的整体逻辑分析

在上述控制系统设计方案的基础上，我们还需要分析出控制程序的整体逻辑。本设计任务的巡线搬运机器人具有两种运动策略：巡线和搬运。我们可以通过读取超声波传感器的值来判断运动方向上是否有纸杯，以此选择机器人的运动策略。在达到仓库卸货之前，机器人运动控制程序的整体流程图4.11。

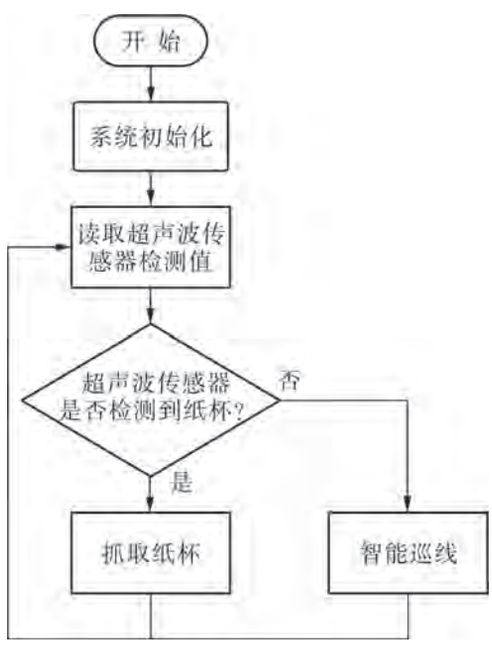


图 4.11　控制程序的整体流程图

如何利用超声波传感器来判断运动方向上是否有障碍物已经在本教材第三章的第五节进行了介绍，请同学们自行完成纸杯的检测，当纸杯处于机器人运动方向前方 180 ~ 240 mm 的时候，机器人停止前进。我们接下来重点完成的是抓取纸杯与智能巡线的控制程序。

二、抓取纸杯的运动控制

1. 舵机控制角度的分析与测试

当检测到运动方向上有纸杯时，机器人停止前进，并进入抓取纸杯的动作系列：张开手爪—放下手臂—合上手爪—抬起手臂。动作的切换对应的是手爪舵机和手臂舵机转动角度的变化，当我们掌握了各个运动对应的舵机角度，就可以实现对机械手抓取纸杯的运动控制。

探究与交流

请同学们编写程序控制机械手臂和机械手爪，找到各个动作系列对应的最合适的舵机角度组合，并完成表 4.2：

表 4.2　各个动作系列对应的最合适舵机角度

动作系列	张开手爪	放下手臂	合上手爪	抬起手臂
机械手臂舵机				
机械手爪舵机				

2. 舵机控制的编程实现

舵机主要是由驱动电动机、减速器与位置检测元件所构成的，通过 PWM 调速来控制转动角度的大小。为了方便对舵机进行控制，Arduino IDE 里面已经集成了控制舵机的库函数，我们可以直接加载库函数来进行舵机控制。加载了库函数后，我们需要给每个舵机都声明一个对象，然后就可以调用各个对象的库函数对舵机运动角度进行直接控制了。

```
#include<Servo.h>                    // 加载库函数
Servo myservo_Arm;                   // 声明手爪舵机的对象
Servo myservo_Hand;                  // 声明手臂舵机的对象
myservo_Arm.write(x);                // 让手爪舵机转动到角度 x
myservo_ Hand.write(y);              // 让手臂舵机转动到角度 y
```

我们使用的舵机的运动角度范围是 0~180°。为了保护舵机，一般尽量不让舵机在极限位置工作，因此舵机角度的参数输入范围一般是 10° ~170°。

三、智能巡线的运动控制

1. 巡线函数的控制逻辑分析

巡线函数相对比较复杂，我们首先需要区分的是正常直线还是交叉口。如果是正常直线，我们按本章第二节的机器人巡黑线基础实验的控制方法即可。如果是交叉口，我们需

要根据纸杯的摆放位置及交叉口序号来进行对应的决策。

如图4.12所示，假设机器人从最左侧出发、纸杯放在边L2上，机器人的移动动作系列是：前进直到交叉口P1—右转60°—前进直到交叉口P2—左转60°—前进直到纸杯并抓取纸杯—前进直到交叉口P3—左转120°—前进直到交叉口P9—卸载纸杯。

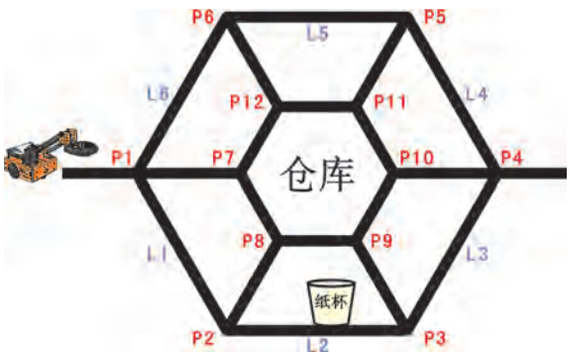


图 4.12　纸杯摆放位置示意图

探究与交流

如果改变机器人出发地点与纸杯摆放位置，请同学们分析对应的移动动作，并完成表 4.3：

表 4.3　不同出发点和纸杯摆放位置所对应的机器人移动动作

出发点	纸杯位置	移动动作
最左边	L6	
最左边	L3	
最右边	L2	
最右边	L1	

接下来，我们以机器人从最左侧出发、纸杯放在边 L2 上为案例继续分析。机器人运动过程中共遇到了 4 个交叉口，处理策略分别是：右转 60°、左转 60°、左转 120°、卸载纸杯，我们可以通过判断是第几个交叉口，来选择对应的运动策略。因此，可以得到图 4.13 的智能巡线子程序流程图。

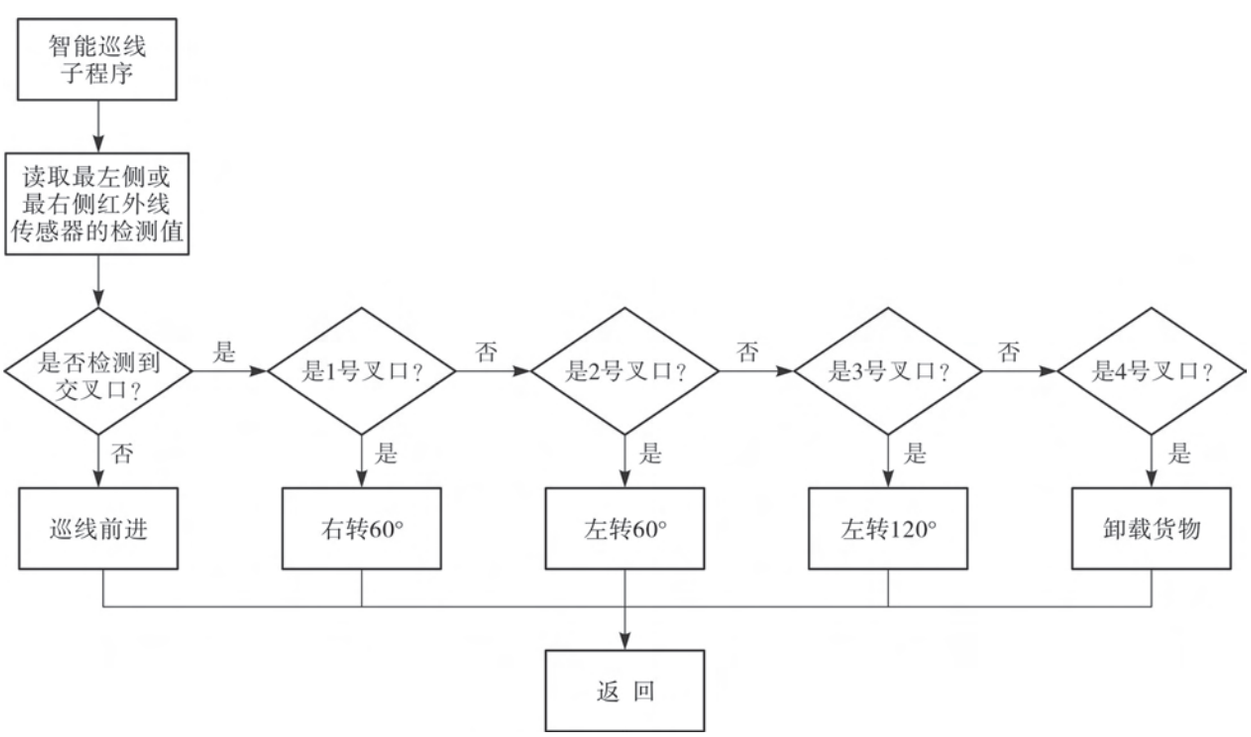


图 4.13　智能巡线子程序流程图

接下来需要解决的问题是如何判断是第几个交叉口，以及如何实现：右转60°、左转60°、左转120° 等交叉口转弯策略。

2. 交叉口序号判断

我们可以通过读取最左侧 S1 或最右侧 S4 红外线传感器的检测值来判断是否遇到交叉口，当这两个传感器有一个检测到黑线的时候，就证明遇到了交叉口。例如，我们使用变量 *C* 来标记是否遇到了交叉口，用变量 *N* 来记录检测到的交叉口数量。

```
if (digitalRead(S1) == 1||digitalRead(S4) == 1)          // 是否遇到交叉口
{
        C = 1;                                           // 将 C 赋值为 1，表示遇到交叉口
        N++;                                             // 交叉口数量增加 1
}
```

在判断是第几个交叉口的时候，我们可以使用 switch-case 条件判断语句，根据 *N* 的数值来选择不同的执行语句。

```
switch (N)
{
  case 1:   // 第一个交叉口
        语句块 1;
        break;
  case 2:  // 第二个交叉口
        语句块 2;
        break;
  case 3:  // 第三个交叉口
        语句块 3;
        break;
  case 4:  // 第四个交叉口
        语句块 4;
        break;
  default:
       break;
}
```

3. 交叉口转弯策略

（1）在 1 号交叉口实现右转 60°。在本节的基础实验中，我们已经学习了在 T 形交叉口实现左右转的控制方法。1 号交叉口共有四个分支，是非对称的十字交叉口，如果我们直接用 T 形交叉口的控制策略，会出现什么问题呢？机器人在检测到交叉口并前进一段距离后，有可能右侧传感器正好落在黑线上方，因此直接进入巡线前进的程序，导致机器人直接向前运动而没有实现右转弯。如何解决这个问题呢？我们可以让机器人在前进一段距离后先向右转一个小角度使得右侧传感器 S3 脱离黑线，然后再让机器人保持右转直到右侧传感器 S3 再次检测到黑线。

做中学

1. 请同学们完善下面的流程图（图 4.14）并进行编程调试。

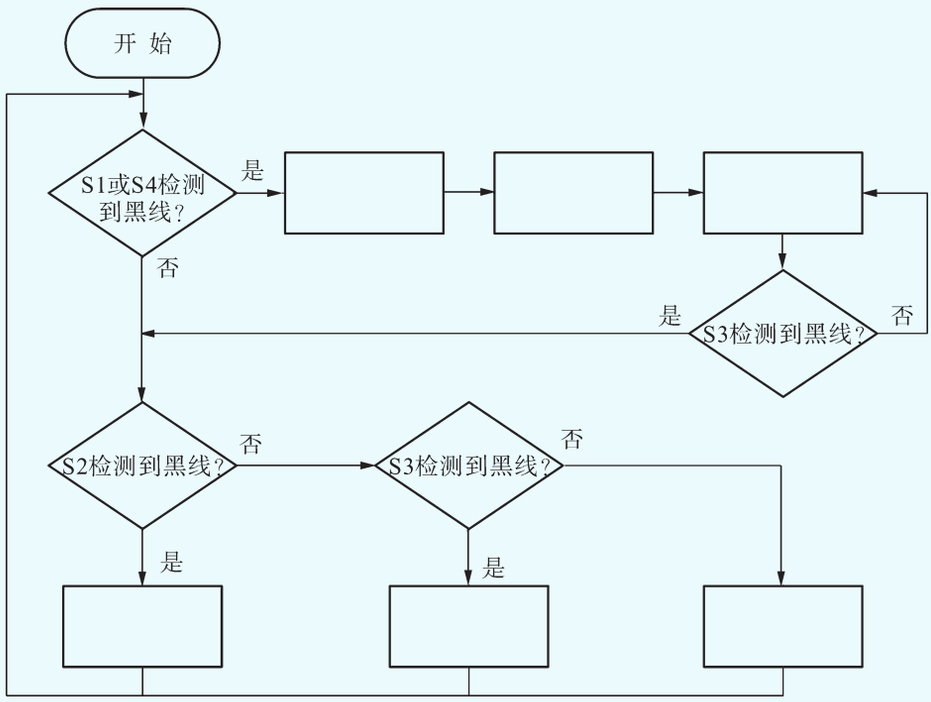


图 4.14　机器人交叉口右转 60° 运动的程序流程图

2. 请同学们记录能顺利通过 1 号交叉口的运动控制参数，并完善下面的程序。

```
/************************* 右转 60° 函数 ************************/
void turn_Right_60()
{
    Drive(1,____);              // 设置机器人前进速度
    delay(____);                // 设置机器人前进时间
    Drive(4,____);              // 设置机器人右转速度
    delay(____);                // 设置机器人右转时间
    while (1)
    {                           // 原地右转直到 S3 检测到黑线，然后继续巡线
      Drive(4,____);            // 设置机器人右转速度
      if (digitalRead(S3) == 1)
      break;                    // 如果 S3 检测到黑线跳出 while 循环
    }
}
```

（2）在 2 号交叉口实现左转 60° 。在机器人正常巡线前进的情况下，就能够顺利完成左转 60° 的任务目标，请同学们分析其中的原因。

（3）在 3 号交叉口实现左转 120° 。在 3 号交叉口，我们可以采用类似十字路口的运动控制策略：当检测到 3 号交叉口后，让机器人先前进一段距离，然后让机器人向左转一

个小角度使得左侧传感器 S2 脱离黑线，然后再保持左转直到左侧传感器 S2 再次检测到黑线。

做中学

1. 请同学们完善下面的流程图（图 4.15）并进行编程调试。

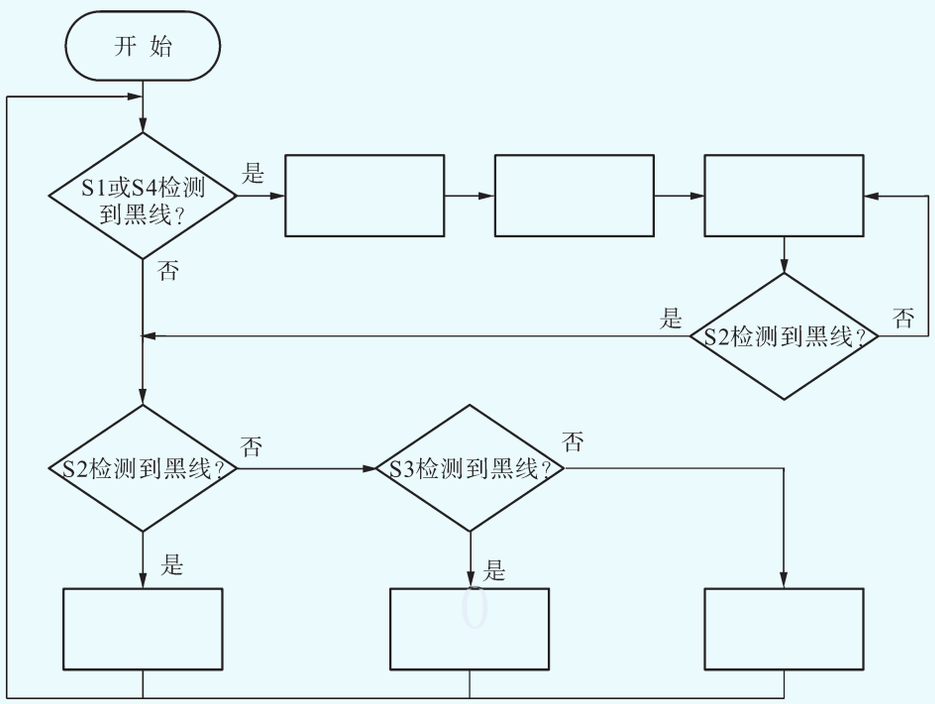


图 4.15　机器人交叉口左转 120° 运动的程序流程图

2. 请同学们记录能顺利通过 3 号交叉口的运动控制参数，并完善下面的程序。

```
/************************ 左转 120° 函数 ************************/
void turn_Left_120()
{
    Drive(1,_____);                  // 设置机器人前进速度
    delay(_____);                    // 设置机器人前进时间
    Drive(3,_____);                  // 设置机器人左转速度
    delay(_____);                    // 设置机器人左转时间
    while (1)
    {                                // 原地左转直到 S2 检测到黑线，然后继续巡线
        Drive(3,_____);              // 设置机器人左转速度
        if (digitalRead(S2) == 1)
        break;                       // 如果 S2 检测到黑线跳出 while 循环
    }
}
```

四、程序的综合调试

现在，我们已经一起将巡线搬运机器人涉及的整体控制逻辑跟各个子函数的实现都进行了分析，请同学们自行完成整体控制程序的编写，并在机器人载体上进行调试实验。请同学们将调试过程中遇到的问题及解决方法填到表 4.4。

表 4.4　巡线搬运机器人调试实验问题登记表

序号	实验现象	原因分析	解决方法
1			
2			
3			
4			
5			

实验成功后，请同学们尝试编写其他出发地点与纸杯摆放位置组合下的机器人运动控制程序，还可以尝试参加巡线搬运主题的国内外机器人大赛。

本章小结

路径规划是机器人研究领域的一个重要分支，是对巡线、避障、图像识别和运动控制等技术的综合运用。现实生活中，扫地机器人、送餐机器人、自动导航物流小车等都涉及路径规划问题。在本章，我们综合运用前三章所学的知识，完成了第一章提出的“巡线搬运机器人”项目控制程序的开发与调试。至此，我们的简易机器人已经搭建完成了。

学习评价

评价内容			评价方式		
			自我评价	小组评价	教师评价
过程评价	师生互动	能积极思考老师提出的问题			
		能基于已有经验构建新的知识			
		能积极参与课堂讨论			
	实践活动	能积极参与实践活动			
		与小组成员有效合作			
		机器人走正方形的编程控制			
		机器人巡黑线及巡线中特殊路口的处理			
		机器人抓取纸杯的运动控制			
结果评价	目标实现	了解机器人路径规划的定义、分类以及方法			
		掌握自顶而下的程序设计方法			
		制订巡线搬运机器人控制策略			
	收获反思	收获与感悟			
		反思不足			

附录　部分中英文词汇对照表

编程	programming
编译	compilation
变量	variable
常量	constant
齿轮	gear
齿轮传动	gear drive
传动装置	transmission device
传感检测系统	sensing monitoring system
代码	code
单片机	singlechip
电动机	electric motor
电平	electrical level
电阻（器）	resistor
动力源	power source
端口	port
发动机	engine
发光二极管	light emitting diode，LED
服务机器人	service robot
复位电路	reset circuit
工业机器人	industrial robot
光敏电阻	light-dependent resistor，LDR
函数	function
红外线传感器	infrared sensor
机器人	robot
机械臂	mechanical arm
机械传动	mechanical drive
机械结构	mechanical construction
开发环境	development environment
可编程逻辑控制器	programmable logic controller，PLC
控制系统	control system

履带机器人	tracked robot
脉冲宽度调制	pulse width modulation，PWM
模拟信号	analog signal
驱动	drive
曲柄摇杆机构	crank-rocker mechanism
软件	software
声音传感器	sound sensor
时钟电路	clock circuit
输出	output
输入	input
数字信号	digital signal
双曲柄机构	double-crank mechanism
双摇杆机构	double-rocker mechanism
随机存储器	random access memory，RAM
特种机器人	special type robot
调试	debug
通用串行总线	universal serial bus，USB
下载	download
巡线	line-following
循环	loop
引脚	pin
硬件	hardware
运算符	operator
只读存储器	read-only memory，ROM
执行装置	executive device
中央处理器	central processing unit，CPU