

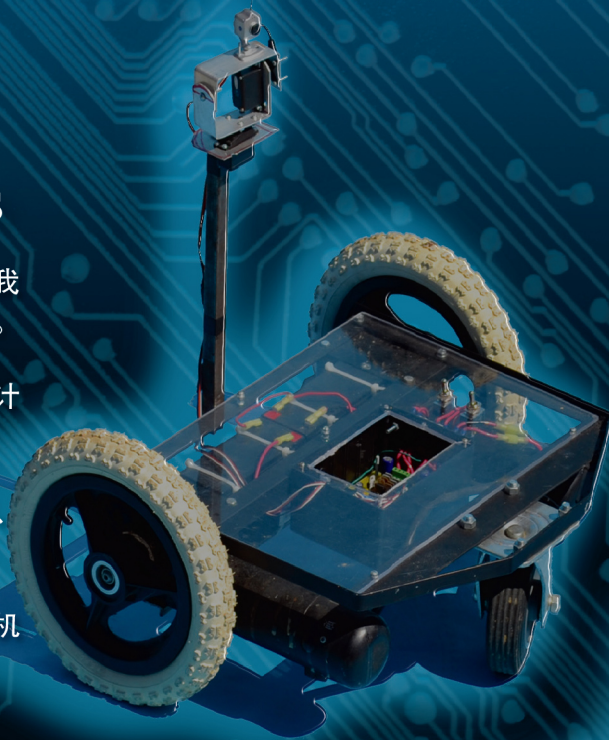
Arduino 机器人 权威指南

John-David Warren
【美】Josh Adams 著
Harald Molle

于欣龙 张阳 张岩 陈丽 译

Arduino Robotics

- 从基础机器人到高级机器人，我们都会提供详细的**指导与说明**。
- 教你熟练使用**EAGLE**软件设计制作**电路板**。
- 让你学会如何使用各种**传感器、电机、控制器**。
- 从**制作到测试**，轻松实现你的机器人作品展。



Apress®



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

译者简介:

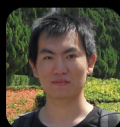


于欣龙，工学硕士，2013年毕业于哈尔滨工程大学机械设计及理论专业。大学期间获得国家、省级奖励及表彰共48项，申请发明专利2项、实用新型与外观专利6项，还曾荣获2007中国机器人大赛空中机器人组固定翼15公斤级冠军、5公斤级亚军。现任哈尔滨奥松机器人科技有限公司总经理，哈尔滨工程大学创业学院创业导师，中国第一本开源硬件书籍《爱上Arduino》译者，国内资深创客。



张阳，互动艺术设计师，热爱科技和艺术，致力于互动科技和雕塑的结合。做过小型四足机器人、检测人体喷水的互动佛像、用iPhone控制的可折叠轻型滑板车、尝试以产品标准设计研发的龙爪灯等。

个人网站: www.zhangyangdesign.com。

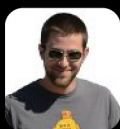


张岩，工程硕士，本科就读于哈尔滨工程大学英语专业，后被保送本校电子与通信工程专业攻读硕士研究生。目前在大唐电信从事智能卡研发工作，业余时间关注Linux和Contiki等开源项目，探索物联网技术在国内的应用模式和发展方向。



陈丽，博士研究生、讲师，IEEE及中国计算机学会会员，主要研究方向为无线移动感知网络及物联网等。参研国家自然科学基金及省部级科研项目5项，在《计算机学报》、《软件学报》等国家一级学术刊物及国外EI期刊发表论文6篇。

作者简介:



John-David Warren 毕业于阿拉巴马大学伯明翰分校，电子产品爱好者，实践过许多不同的项目，小到电动钓鱼竿，大到遥控割草机。2010年4月，他的作品登上了 *MAKE* 杂志封面。他目前和他美丽的妻子 Melissa 住在阿拉巴马州伯明翰市。



Josh Adams 毕业于阿拉巴马大学伯明翰分校，是软件开发人员，在设计优质软件 and 项目管理方面，拥有超过10年的专业经验。Josh 是 Isotope Eleven 公司的首席设计师，并负责监管架构决策。

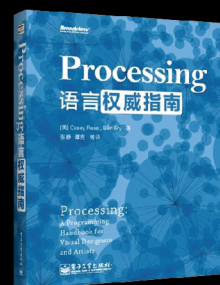


Harald Molle 是一位拥有30年经验的电脑工程师。他是一个专业的潜水员，为此甚至研究一款GPS控制机器人来勘测湖泊。他的妻子 Jacqueline 非常支持他。

同类图书:



ISBN 978-7-121-19201-2



ISBN 978-7-121-21352-6

Arduino机器人 权威指南

John-David Warren
【美】Josh Adams 著
Harald Molle

于欣龙 张阳 张岩 陈丽 译

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

你想制作有趣的机器人吗？你想让机器人沿着指定的路线行走吗？你想让机器人帮助你清扫庭院吗？你想让机器人载着你逛街吗？那么请选择《Arduino 机器人权威指南》吧！John-David Warren、Josh Adams 和 Harald Molle 会带你走进机器人的世界，教会你如何寻找配件、怎样制作机器人、怎样进行编程，甚至告诉你如何对机器人进行安全测试。

本书可以教会你如何用 Arduino 来控制各式各样的机器人，同时提供了每一步的详细指导。你不仅可以学会 Arduino 的基本使用方法，还可以了解各种电机的特性。同样，你会掌握其控制和排除故障的方法，将之应用到你的机器人项目中。本书从易到难，讲解了各种各样机器人的制作方法，其中包括 GPS 机器人、草地机器人、格斗机器人及赛格威机器人等。

无论你是只会摆弄 Arduino 的初学者，还是一个制作小工具的专家，《Arduino 机器人权威指南》都会帮助你制作出意想不到的机器人作品。

Arduino Robotics

By John-David Warren, Josh Adams, Harald Molle, ISBN:978-1-4302-3183-7

Original English language edition published by Apress Media.

Copyright ©2011 by Apress Media

Simplified Chinese-language edition copyright ©2014 by Publishing House of Electronics Industry

All rights reserved.

本书中文简体版专有版权由 Apress Media 授予电子工业出版社。专有版权受法律保护。

版权贸易合同登记号 图字：01-2013-4142

广告经营许可证号：京海工商广字第 0258 号

图书在版编目（CIP）数据

Arduino 机器人权威指南 / （美）沃伦（Warren, J.D.），（美）亚当斯（Adams, J.），（美）莫勒（Molle, H.）著；于欣龙等译. —北京：电子工业出版社，2014.3
书名原文：Arduino robotics
ISBN 978-7-121-22291-7

I. ①A… II. ①沃… ②亚… ③莫… ④于… III. ①单片微型计算机—应用—机器人—制作—指南
IV. ①TP242-62

中国版本图书馆 CIP 数据核字（2014）第 000482 号

策划编辑：林瑞和

责任编辑：李云静

特约编辑：顾慧芳

印 刷：

装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：36.5 字数：728 千字 彩插：1

印 次：2014 年 3 月第 1 次印刷

定 价：99.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlt@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

译者序

Arduino 的诞生可谓开源硬件发展史上的一个新的里程碑，本人见证着其在中国蓬勃发展的整个过程。尤其近几年，随着 Arduino 软硬件功能的不断提高，周边配套模块的不断完善，其应用日趋成熟，应用领域也日趋扩大，从环境监测、智能家居再到 3D 打印等多个领域都能看到它的身影，特别是在机器人和自动化领域，因为它的存在而生辉增色不少。Arduino 因其程序语法简单易懂、开发资料丰富，受到众多创客的追捧，从小就喜欢动手 DIY 的我也不例外。

正当本人忙于本书最后校稿之时，谷歌已完成对“大狗”(BigDog)机器人制造商 Boston Dynamics 公司的收购。它是谷歌近半年来收购的第八家机器人公司。这则爆炸性新闻，让“我和我的小伙伴们都惊呆了”，谷歌的这些举动足以说明未来机器人行业发展的巨大潜力。纵览国内现有解读 Arduino 和机器人制作的相关书籍，本人发现大多是围绕单一内容展开编写的，将两者结合，系统全面讲解 Arduino 在机器人上应用的书籍并不多。本人有幸先于广大读者读到本书英文版《Arduino Robotics》，通过阅读发现，本书在内容编排上由浅入深、循序渐进，力求通俗易懂，是一本由“理论入，实践出”的实用型工具书，为此本人决定联合国内几位资深机器人爱好者将其翻译出来，早日跟大家分享，共同学习。

内容编排

全书共分 13 个章节，各章节内容明确，重点突出，确保读者快速理解而又不浪费时间。如果你是一个完全没有电子基础却又对机器人制作感兴趣的初学者，可以从头开始章章通读，前两章从介绍理论知识、常用工具、电子元件和传感器入手，内容简单，容易理解；第 3 章主要介绍机器人常用的不同种类电机和驱动电路，让读者为后面制作机器人做准备。如果你对机器人制作和 Arduino 编程有一定基础，相信本书的各种机器人制作项目也会让你大开眼界，从第 4 章到 13 章作者通过亲自动手制作机器人项目的方式教授读者如何选择电子器件，焊接电路，制作 PCB，以及根据功能需要编写程序，到实际调试和后续

改进等过程，整个过程图文并茂，剖析准确到位，程序注释句句清晰、简明。其中第 6 章内容告诉读者如何使用目前国外最流行的 PCB 绘制软件 eagle 设计并制作电路板也是本书特色项目之一，填补了这款软件的国内教学空白。全书中的例子代码可通过作者创建的网站下载获取，关于 Arduino 其他硬件方面的创作，你也可以登录 Rebecca 的博客 <http://blog.sina.com.cn/arduino> 查找学习。

因本书涉及内容多、专业性强，故邀请不同译者分章翻译，最终统稿编排，因此书中难免会出现疏漏与错误。如果读者在阅读过程中发现任何问题，希望找到译者共同探讨，那么可以加入 Arduino Robotics 主题 QQ 群：334268182。在这个群里你会获得更多关于 Arduino 和机器人方面问题的解答。

致谢

首先要感谢作者为广大 Arduino 和机器人爱好者做出的巨大贡献。其次，本人因为 Arduino 有幸结识其他译者，在此特别感谢我的挚友张阳、张岩和陈丽，没有你们的辛勤付出，本书不可能顺利完成。当然，还要感谢本书编辑，因为你们对本书译稿的多次审阅和提出宝贵建议才促使本书更加完善并得以早日出版。另外，奥松机器人王非凡、朱新龙、马丽娜也为本书的校对和出版做了大量工作，在此对他们表示由衷感谢。

古人云：“授人以鱼，不如授之以渔”，本书不仅能够带着你一步一步去制作各种功能的机器人，还能教会你使用更多制作方法和小妙招。要想做一位名副其实的创客，《Arduino 机器人权威指南》是你的最佳选择，动手就从这里开始吧！

于欣龙

奥松机器人创始人、资深创客

关于作者

J-D (John-David Warren) 不仅是一位电子爱好者，他还曾做过建筑工人，经常做一些小发明。当他还是一个孩子的时候，他总是喜欢把自己手头的东西拆开，去研究这个东西到底是如何工作的。自那以来，他已经做了许多不同的项目，小到电动钓鱼竿，大到遥控制草机。2010 年 4 月，他的作品登上了 *MAKE* 杂志封面。在他做建筑工人的 8 年里，他曾从事过木工、管道工程、电气工程等行业。他的知识是建立在现实社会实践基础上的，而不是仅仅背诵书本知识。



除了制作机器人和遥控玩具外，他还喜欢自动化日常的事务，比如让发光二极管闪烁，设计腐蚀印刷电路板，以及其他相关的东西。他把大部分时间都花在研究、制作和测试各种电机控制器上面，让他的机器人动起来。他自封为“贫穷的机器人专家”，他总是试图找最便宜的零件去做某件事情——通常自己亲手制作。

J-D 毕业于阿拉巴马大学伯明翰分校 (UAB)，取得了企业管理学位。目前他和美丽的妻子 Melissa 住在阿拉巴马州伯明翰市，他们饲养了一群小动物。

Josh Adams 是一位软件开发人员，在开发优秀的软件 and 项目管理方面，拥有超过 10 年的专业经验。他为一个高中的科学项目做了一个特斯拉线圈（人工闪电制造器），人工闪电的范围大于 27 英寸。Josh 是 Isotope Eleven 公司的首席设计师，并负责监管架构决策，将顾客的要求演化为工作软件。Josh 毕业于阿拉巴马大学伯明翰分校 (UAB)，在数学和哲学专业获得了理学学士学位。当他不工作的时候，喜欢和家人在一起。另外，Josh Adams 还担任了本书技术审稿的工作。



自 1984 年以来，**Harald Molle** 一直从事电脑工程师的工作。早在嵌入式系统公司成立之前，他就在德国西南部的一个大学做研究工作，从此开始了他的职业生涯。**Harald Molle** 还是一名专业的潜水员，因为对潜水充满激情，所以他就开发了一款 GPS 控制机器人来勘测湖泊，这样就将个人爱好与自己的工作结合在了一起。他的妻子 **Jacqueline** 非常理解他，知道机器人爱好者是需要大量时间来进行研究工作的。另外，**Harald Molle** 还养了一只猫。



关于技术审稿人

本书的技术审稿人除了作者 Josh Adams 外，还有 Guilherme Martins。

Guilherme Martins 1977 年出生于里斯本。他对多种艺术形式感兴趣，很早就开始尝试各种媒介，比如摄影、录像、素描、油画等。自 2000 年以来，Guilherme Martins 曾在几家设计工作室和广告机构工作，一直从事自由职业的视觉项目，这跟动态影像设计、视觉效果、网页设计都相关。2007 年他开始与编舞家 Rui Horta 合作，在进行舞蹈表演、戏剧表演和歌剧表演时，将视觉内容投射到舞台上。

作为一个工匠和发明家，为了创造出新的互动体验，他对机器人技术和电子实验特别感兴趣。他大部分的研究成果在这个网站（<http://guilhermemartins.net>）里都可以查到。



鸣谢

合作作者

我要感谢我的朋友们，Josh Adams，感谢他对我的几个项目的帮助以及不断的支持和思想的援助，并把我介绍给了 Arduino。本书第 13 章的内容是他使用个人电脑写作的，利用游戏手柄和无线串行链路去控制一个机器人（Arduino+Processing）。正因为有了他超强的编码技巧，才使我的项目更加有吸引力。我的朋友，非常感谢你的帮助。

我还想感谢一位 Arduino 高手朋友，Harald Molle，他对 GPS 制导的机器船（第 9 章）——这一复杂的项目花费了很多的时间、倾注了大量的心血。本书第 9 章的内容是 Harald 用业余时间撰写的。在 Arduino 论坛上我看到他已解决了我曾遇到的同样问题之前，我做过三个很不成功的船体。他天才设计的双体船，很容易制作，并且在水中能保持直线行驶。他在这方面的知识要比我想象的多，我非常感谢他愿意与我分享他的经验。

技术编辑

特别感谢 Guilherme Martins 和 Josh Adams，他们花了大量的时间对这本书进行复审，对他们提出的建议与反馈，我非常感激。

编辑

非常感谢 Michelle Lowman 给我写这本书的机会；感谢 Anita Castro 的耐心，花费许多额外的时间完成这本书；同时，还要感谢 James Markham、Frank Pohlmann，以及 Dominic Shakeshaft，感谢他们的帮忙和指导；最后非常感谢 Apress 出版团队，是他们的帮助才使这本书得以出版——写第一本书不是那么容易的。

家庭

我要感谢我的妻子 Melissa，在写书的过程中非常理解我并给予我很大的帮助，没有她的支持，我是不可能写完这本书的。最后，我要感谢我的家人，感谢他们在这个项目中对我的支持和祝福。

Arduino 社区

本书中大多数项目的灵感都来源于 Arduino 社区。当我第一次开始学习 Arduino 和物理计算之前（2008 年），我从未接触过单片机和计算机编程。社区里 Arduino 的爱好者们都很热情，纷纷帮助我，给我代码例子、项目建议，提出一些新奇的想法。总之，正因为有了他们的支持，我才能坚定地走下去。在人人都想赚钱的社会里，你知道，不花一分钱就能参与，这是一件很棒的事（除了便宜的 Arduino 电路板本身）。

我要感谢一些帮助我学习的几个机器人爱好者、制造商和电子专家们。感谢 Massimo Banzi、Tom Igoe、Arduino 研发团队、Limor Fried (Ladyada.net)、Nathan Seidle (Spark Fun.com)、David Cood (Robotroom.com)、Jordi Munoz、Chris Anderson (DIYdrones.com)，感谢他们提供的 Arduino 相关教程、项目，以及他们对开源社区做出的贡献。还要感谢以下这几个人，他们提出的有关项目和配件的建议对我有很大的帮助，他们分别是：Larry Barello、John Dingley、Shane Colton 和 Bob Blick。

还要感谢我志同道合的朋友们，是他们倾听我疯狂荒谬的想法，检查我的技术原型，给我提供一些建议，甚至在我无钱购买零配件的时候，将零配件无偿地提供给我，他们是：Josh Adams、Anthony DiGiovanni 和 Laird Foret。

虽然在本书中上传了很多我自己拍摄的关于配件和程序的照片，但效果并没有我期望的好。下面列举的这些照片来源于我的零件供应商，非常感谢他们的帮助。

SparkFun.com: 1-10, 1-21, 1-30, 2-22, 2-24, 2-26, 2-27, 2-28, 2-29, 3-20

Digikey.com: 1-15, 1-16, 1-17

PartsForScooters.com: 10-4, 12-8, 12-18

DimensionEngineering.com: 10-20, 11-14

Adafruit.com: 2-16

Pololu.com: 10-22

Electronics-Tutorials.ws: 1-3

DannyG.com: 1-9

RobotMarketplace.com: 12-7

DFRobots.com: 5-11

HarborFreightTools.com: 10-3

——John-David Warren

非常感谢我的妻子 Kristen，对我摆弄机器人的耐心；我的儿子 Matthew，只想着要跟机器人玩；我的女儿 Gracie，即将出生。

——**Josh Adams**

谢谢来自 DIYdrones.com 网站的 Chris Anderson 和 Jordi Munoz，感谢他们在制作自动驾驶仪中的创举；感谢来自 <http://water.resist.free.fr> 网站的 Jean Margail，谢谢他对双体船的设计；感谢 Matthias Wolbert 制作的三维模型和 Robert Herrmann 提出的排除故障的建议。最后我把这一切归功于我的妻子，她知道为什么。

——**Harald Molle**

介 绍

这本书是为那些想了解更多关于 **Arduino** 和机器人的爱好者们写的。虽然有些项目是面向大学生和成人的，但涵盖机器人项目的初级章节也同样适合初中生和高中生阅读。因为我曾看到过一些很棒的项目，他们都是由老年人或者是年轻人制作的，所以我不会对这本书做年龄的限制。

预备知识

首先，你要能够使用基本的电动工具、手动工具、电压表和电烙铁。如果你没有这些方面的经验，也不用担心，因为你的第一次经验已经向你走来（你必须从某处开始），就像骑自行车一样，熟能生巧。

如果你是一名经验丰富的机器人制造者，通过学习我介绍的一些方法，能够使你进步得更快。如果你是一个初学者，你就再也不会把一些孔钻到错误的地方、把一个轮子安装得不直、把机器人做得如此丑陋。不要总是想第一次就把每一步都做得非常完美，第一次只要尽力做就行，然后回过头再去改善。手中有一个可以工作但不完美的机器人比什么都好（因为你太害怕去尝试）。

总之，这本书主要是为那些 **Arduino** 爱好者们提供有趣的项目。如果你在做项目的时候，没有寻找到乐趣，或是做错了；如果你困惑于一个项目，不得不停下来；那么请寻求帮助吧——没有人想让你感到沮丧，但学习新的东西，有时候会让你有一种“用头撞墙”的感觉；不要那样做，只要坚持住，相信你的问题最终会得到解决的。我建了一个谷歌网站，里面包含了本书每个项目的文件并且给大家提供了一个互相交流学习的平台。

网址：<https://sites.google.com/site/arduinorobotics/>

如果你想尝试一些其他的 **Arduino** 的项目，处理各种类型的传感器、发光二极管、家庭自动化，以及其他各种项目，你可以参考 **Apress** 出版公司关于 **Arduino** 的书籍。

书名: **Practical Arduino** 作者: **Jonathan Oser and Hugh Blemings** (2009)

书名: **Beginning Arduino** 作者: **Michael McRoberts** (2010)

John-David Warren

目 录

■ 第 1 章 基础	1
1.1 电学	2
1.1.1 电模拟	3
1.1.2 电的基础知识	4
1.1.3 电路	6
1.1.4 电信号测量	7
1.1.5 万用表	7
1.1.6 电压测量	8
1.1.7 电流强度测量	9
1.1.8 电容测量	9
1.1.9 电阻测量	10
1.1.10 使用欧姆定律计算电阻功率	11
1.1.11 示波器	12
1.1.12 负载	13
1.1.13 电路连接	14
1.2 电子学	15
1.2.1 半导体	16
1.2.2 技术手册	17
1.2.3 集成电路	18
1.2.4 通孔元件	19
1.3 Arduino 初级读本	20
1.3.1 Arduino 变体	21
1.3.2 Arduino 集成开发环境	24
1.3.3 草稿	24
1.3.4 信号	28

1.4	构建电路	38
1.4.1	电路设计	38
1.4.2	原理图	39
1.4.3	原型	41
1.5	搭建机器人	47
1.5.1	硬件	47
1.5.2	材料	49
1.5.3	工作区域	50
1.6	总结	51
■ 第 2 章	基于 Arduino 的机器人	52
2.1	Arduino 接口	52
2.1.1	继电器	53
2.1.2	晶体管	59
2.1.3	电机控制器的接口	66
2.2	用户控制	69
2.2.1	连线（有线）控制	70
2.2.2	红外控制（IR）	70
2.2.3	无线电控制系统	71
2.3	传感器导航	74
2.3.1	接触式感知	75
2.3.2	距离和反射式感知	75
2.3.3	方向（定位）	78
2.3.4	非自主传感器	81
2.4	总结	84
■ 第 3 章	让机器人动起来	85
3.1	电机	85
3.1.1	有刷直流电机（永久磁铁型）	86
3.1.2	无刷直流电机	87
3.1.3	步进电机	88
3.1.4	齿轮减速电机	90
3.1.5	伺服舵机	91
3.1.6	线性制动器	92

3.1.7	功率计算	93
3.1.8	驱动	93
3.1.9	寻找合适的电机	94
3.2	H 桥	95
3.2.1	产生制动	96
3.2.2	实现	97
3.2.3	H 桥集成电路	100
3.2.4	更改 PWM 频率	101
3.2.5	反电动势	103
3.2.6	电流检测	105
3.2.7	基于 H 桥的商品 (电机控制器)	106
3.3	电池	109
3.3.1	镍镉电池 (NiCad)	110
3.3.2	镍氢电池 (NiMH)	110
3.3.3	锂聚合物电池 (LiPo)	111
3.3.4	铅酸蓄电池	112
3.3.5	充电	114
3.4	材料	115
3.4.1	木材	115
3.4.2	金属	116
3.4.3	螺栓和螺母	116
3.4.4	塑料	117
3.4.5	链条和链轮齿	117
3.4.6	车轮	118
3.5	总结	118
■ 第 4 章	莱纳斯寻线机器人	120
4.1	莱纳斯的零件清单	121
4.2	如何使莱纳斯工作	123
4.2.1	轨道	123
4.3	制作红外传感器电路板	124
4.4	改装成连续旋转的伺服舵机	132
4.4.1	方法 1: 带有外部速度控制器的直接直流驱动	132
4.4.2	方法 2: 带有内部电机驱动电路的伺服脉冲驱动	135

4.5	合适的驱动轮	137
4.6	制作框架	141
4.7	制作连接	146
4.8	电池安装	147
4.8.1	安装电源开关	148
4.9	加载代码	148
4.10	制作轨道	155
4.11	测试	156
4.12	附加组件	157
4.12.1	LED 指示灯	157
4.12.2	喷漆	159
4.12.3	添加速度调节器（电位器）	162
4.13	总结	164
■ 第 5 章	墙追踪机器人 Wally	166
5.1	如何让 Wally 工作	167
5.2	Wally 的零件清单	169
5.3	电机控制器	171
5.3.1	高侧开关	171
5.3.2	低侧开关	171
5.3.3	电路制作	173
5.4	制作框架	178
5.5	安装传感器	182
5.6	安装电池和电源开关	185
5.6.1	电源开关安装	186
5.7	代码	187
5.7.1	代码目标	189
5.8	总结	198
■ 第 6 章	制作 PCB 板	199
6.1	PCB 基础	199
6.2	你需要什么来开始	200
6.3	电路设计	202
6.3.1	搜索开源设计	203

6.3.2	制作你自己的设计	204
6.3.3	使用 Eagle 原理图编辑器工作	207
6.3.4	使用 Eagle 电路板编辑器工作	211
6.4	转印设计	219
6.4.1	让我们制作一块 Arduino 克隆板——Jduino	219
6.4.2	开始转印	221
6.5	腐蚀	227
6.5.1	测量腐蚀液	227
6.5.2	腐蚀（方法 1）	228
6.5.3	腐蚀（方法 2）	230
6.5.4	清除墨粉	232
6.6	钻孔	234
6.7	焊接	235
6.7.1	制作 Arduino 克隆板	235
6.7.2	制作 BJT H-桥	238
6.8	测试	240
6.9	总结	243
■ 第 7 章	昆虫机器人	244
7.1	通过 Arduino 读取开关	245
7.2	如何使昆虫机器人工作	246
7.2.1	天线传感器	246
7.2.2	碰撞传感器	247
7.3	昆虫机器人的零件清单	247
7.4	电机	249
7.4.1	改装伺服舵机	249
7.4.2	控制伺服舵机	251
7.4.3	将脉冲值转换为角度值	252
7.4.4	将车轮安装到伺服舵机	253
7.5	制作框架	255
7.5.1	标记有机玻璃	255
7.5.2	切割有机玻璃	256
7.5.3	安装电机	258
7.5.4	安装脚轮	258

7.5.5	安装 Arduino	260
7.5.6	安装电池	260
7.6	制作传感器	262
7.6.1	前置天线传感器	262
7.6.2	后置碰撞传感器	263
7.7	制作连线	266
7.8	加载代码	267
7.8.1	创建一个延迟	267
7.8.2	变量	268
7.8.3	代码	269
7.9	制作一顶帽子	277
7.10	总结	280
■ 第 8 章	探险者机器人	281
8.1	如何使探险者机器人工作	282
8.1.1	R/C 控制	282
8.1.2	强大的电机	283
8.1.3	电流检测	283
8.1.4	启动视频	283
8.1.5	启动 Xbee	284
8.2	探险者机器人的零件清单	284
8.3	制作框架	286
8.3.1	规格	286
8.3.2	添加电池支架	287
8.3.3	切割底部框架支架	289
8.3.4	切割顶部框架支架	289
8.3.5	切割和弯曲主框架板	290
8.3.6	添加横杆和安装脚轮	291
8.3.7	有机玻璃甲板 (任选)	293
8.4	制作电机控制器	293
8.4.1	电流检测和限流	293
8.4.2	H 桥设计	294
8.5	设置 Arduino	298
8.5.1	连接 H 桥	299

8.6	设置 Xbee	300
8.6.1	测试 Xbee	302
8.7	添加摄像头	303
8.7.1	二自由度云台	304
8.7.2	制作第一个支架	305
8.7.3	制作第二个支架	306
8.8	加载代码	307
8.9	总结	316
■ 第 9 章	机器船	318
9.1	开场白	319
9.2	机器船的零件清单	319
9.2.1	聚苯乙烯泡沫塑料	321
9.2.2	环氧树脂	322
9.2.3	手套	323
9.2.4	玻璃纤维布	323
9.2.5	胶水	324
9.2.6	泡沫塑料切割机和美工刀	325
9.2.7	杂项	325
9.3	机器船设计	325
9.4	组装机器船	327
9.4.1	模板	327
9.4.2	将模板胶合到 EPS/ XPS 板上	329
9.4.3	切出分段	330
9.4.4	把分段胶合在一起	332
9.4.5	插入泡沫锚	333
9.4.6	涂层	334
9.4.7	涂抹成品	335
9.4.8	鳍	336
9.4.9	上色	336
9.4.10	甲板	336
9.4.11	完成组装	337
9.5	推进装置	338
9.5.1	底板	340

9.5.2	枢轴	340
9.5.3	管子	341
9.5.4	舵角	342
9.5.5	电机	343
9.5.6	舵机	344
9.5.7	推杆	344
9.6	电子设备	344
9.6.1	系统的核心——ArduPilot PCB	345
9.6.2	GPS 模块	346
9.6.3	电子调速器 (ESC)	346
9.6.4	电机	347
9.6.5	舵机	347
9.6.6	电池组	347
9.6.7	安装电子设备	348
9.7	软件和任务规划	351
9.7.1	GPS 接收器	351
9.7.2	软件	352
9.7.3	安装软件	372
9.7.4	任务规划	377
9.8	全部放在一起	380
9.8.1	集成系统	381
9.8.2	船, 欢呼吧!	383
9.9	故障排除	383
9.9.1	电机/螺旋桨的推力不够	384
9.9.2	电机不启动	385
9.10	总结	385
■ 第 10 章	草地机器人 400	386
10.1	如何使草地机器人 400 工作	387
10.1.1	割草机甲板	388
10.1.2	大容量电池	388
10.1.3	钢框架	389
10.1.4	卸料斗	389
10.1.5	充气轮胎	390

10.1.6	前灯	390
10.1.7	失效保护	390
10.2	工具和零件列表	391
10.2.1	割草机	391
10.2.2	零件清单	391
10.3	轮子	392
10.3.1	前脚轮	393
10.3.2	后驱动轮	393
10.3.3	安装链轮	394
10.4	框架	395
10.5	传动系统	402
10.5.1	安装电机支架	403
10.5.2	安装链条	406
10.6	电机控制器	408
10.6.1	选购一个电机控制器	408
10.6.2	散热风扇	410
10.6.3	电机控制器反馈	411
10.7	Arduino	413
10.7.1	固定好连接以防行驶颠簸	413
10.8	失效保护	417
10.8.1	无线电遥控拨动开关	418
10.8.2	功率继电器	420
10.8.3	避免无线电遥控本身的失效保护	421
10.9	连接电路	422
10.10	代码	423
10.11	美化和添加附件	429
10.11.1	喷漆	429
10.11.2	前灯	430
10.11.3	卸料斗	430
10.11.4	割草机安全开关	432
10.12	总结	432
■ 第 11 章	赛格威机器人	434
11.1	如何使赛格威机器人工作	435

11.1.1	惯性测量单元	435
11.1.2	转向和增益	436
11.1.3	啮合器	436
11.2	赛格威机器人的零件清单	436
11.3	选择合适的传感器	438
11.3.1	3.3V 电源	439
11.3.2	加速度计	440
11.3.3	陀螺仪	442
11.3.4	陀螺仪和加速度计的总结	443
11.3.5	角度滤波	444
11.4	制作惯性测量单元适配板	445
11.5	选择电机	446
11.5.1	卸掉电力制动器	448
11.5.2	电机安装位置	450
11.6	选择电机控制器	451
11.6.1	SoftwareSerial 库	452
11.6.2	Sabertooth 控制器的简化串口	453
11.7	电池	454
11.7.1	密封铅酸蓄电池	455
11.7.2	充电	456
11.7.3	12V 供电	456
11.8	框架	456
11.8.1	框架设计	458
11.8.2	制作框架	458
11.9	输入装置	462
11.9.1	转向	462
11.9.2	增益	462
11.9.3	啮合器	463
11.9.4	水平启动	463
11.9.5	安装输入装置到框架上	463
11.10	安装电子设备	467
11.10.1	焊接输入装置	470
11.10.2	连接线路	470
11.11	分析代码	471

11.11.1	sample_accel()函数	472
11.11.2	sample_gyro()函数	473
11.11.3	检查角度读数	474
11.11.4	calculate_angle() 函数	475
11.11.5	read_pots()函数	476
11.11.6	auto_level()函数	477
11.11.7	update_motor_speed() 函数	479
11.11.8	time_stamp()函数	482
11.11.9	serial_print_stuff()函数	482
11.11.10	完整代码	484
11.12	测试	491
11.13	总结	493
11.14	参考资料	493
■ 第 12 章	格斗机器人	494
12.1	机器人格斗的诞生	496
12.1.1	格斗机器人的规章制度	496
12.1.2	没有价格限制	497
12.2	格斗机器人零件清单	498
12.3	输入控制	500
12.3.1	Fly Sky CT-6: 32 美元、5 通道、2.4GHz 无线控制器备选方案	501
12.4	电子设备	504
12.4.1	Arduino	504
12.4.2	电机控制器	505
12.5	框架	507
12.5.1	是买，还是做	508
12.5.2	改装轮子	509
12.5.3	制作框架	510
12.6	传动系统	513
12.6.1	齿轮传动装置	514
12.6.2	链条张力调整螺母	515
12.7	电池	520
12.8	安装电子设备	522
12.8.1	保护好你的大脑	522

12.8.2 连接线路.....	524
12.9 代码.....	526
12.10 盔甲.....	532
12.11 武器.....	534
12.12 附加信息.....	540
12.13 总结.....	541
■ 第 13 章 其他控制方式.....	542
13.1 用 Processing 来解码信号.....	543
13.2 其他控制方式所用零件清单.....	543
13.3 选择输入设备.....	544
13.4 Processing 必备文件.....	545
13.5 遵照协议.....	546
13.6 检查 Processing 的代码.....	546
13.6.1 代码解析.....	549
13.6.2 测试 Processing.....	551
13.7 检查 Arduino 的代码.....	553
13.8 总结.....	560

第 1 章



基 础

Arduino 微控制器（如图 1-1 所示）就像一个小型指挥中心，随时等待你发出指令。只需要几行代码，就可以让你的 Arduino 打开或者关闭一盏灯，读取传感器数据并将其显示在计算机屏幕上，甚至可以用它建立一个自制电路来修复损坏的厨房家用电器。由于 Arduino 的多功能性和广大 Arduino 在线社区用户的有力支持，它已经吸引了新一代的电子爱好者，尽管这些人之前从未接触过微控制器，更不用说是编程了。

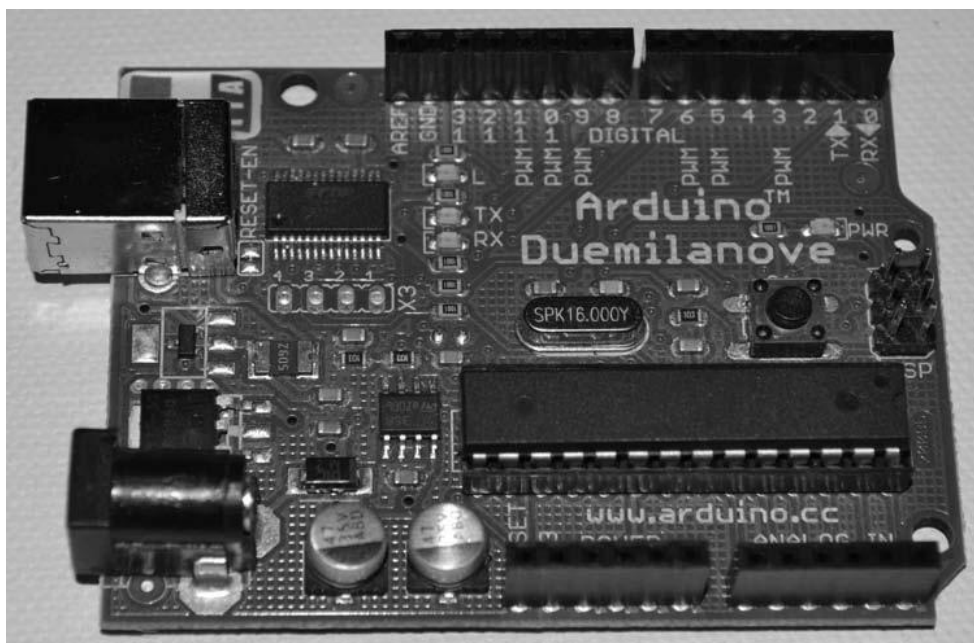


图 1-1 Arduino Duemilanove 微控制器

Arduino 主要的想法就是营造一种氛围，在这种氛围下任何感兴趣的人都可以通过预先支付少量的费用来参与并做出贡献。基本的 **Arduino** 板子可在网上买到，大约 20 美元，而对 **Arduino** 编程需要的所有软件都是开源的（免费使用和修改）。你只需要一台计算机和一条标准的 USB 数据线。除便宜之外，**Arduino** 的创建者还提出了一门易于学习的编程语言（来自 C++），将各种复杂的编程函数组合成简单的指令，使得初学者学习起来更加容易。

本书综合使用一些基本的机器人制作技术，加之 **Arduino** 制作机器人的简单性，便能根据你对工作的清楚理解对机器人进行修改和完善。本书不打算简单地向读者展示如何制作一个机器人，而是要培养初级的机器人制作者，并且可望激发出创造力，这样，读者就可以设计、制作并且修改自己的机器人。

在制作机器人的过程中，大多数人都会遇到一个不可避免的成本问题。很显然，我们可以花费数千美元添置顶级部件和昂贵的商业化产品，然而大多数的兴趣爱好制造者无论是从时间还是金钱上，都无力制造这样的机器人。正是考虑到这一点，本书利用一切机会从零开始向你展示如何制作机器人的一个部件——或者以尽可能低的成本完成制作机器人的工作。如果这些方法看起来都过于复杂，不要担心，因为有替代的零件清单供你去购买。

请理解，本书中的每一个项目在正常工作之前都需要经过多次尝试，其中一些甚至需要花费几周的时间来“调试”。根据以往的经验，我可以告诉你只要坚持下去，最终肯定会解决所有的问题，并且使你的经验更具价值。弄清楚为什么机器人不能正常运行往往需要做大量的故障排查工作，这就需要明白从开始到完成过程中的每一步，然后再检查每一步存在的错误。你动手实践的东西越多，你也就会理解得越好。

最后，如果本书涉及的一些内容你还无法理解，不要气馁，我们假定你是机器人制作和编程的新手，我们专门提供每个项目中用到的零部件和代码的实际工作知识，而不是给你罗列出一堆电气理论和复杂的指令。在开始之前，你最好是抱着“我能做到”的积极态度，这将会是你最好的工具。

为了更好地了解 **Arduino** 内部究竟发生了什么，我们首先应该讨论电气方面和其他一些基础知识（比如电学和电路等）。虽然 **Arduino** 的电压值（直流电压 5V）相对而言不会造成伤害，但是如果你不知道电如何工作，你也就不会知道它在什么时候会变得危险。事实证明，虽然本书提及的所有项目用的电压对于人体来说都不太高，但仍旧应该谨慎操作。

1.1 电学

电无非就是可利用的热，它可用来做各种各样不同的事情，比如照明灯泡、旋转电机，

或者就是为房间供暖。如果电可以很容易地穿过一个物体，则它就被称为“导体”（比如铜线）。每个导体对电流都有内阻，使得它无法传输 100% 的电能。甚至连铜导线也有一些减缓电流量的电阻，从而产生热。导体也存在最大功率值，可以在“过热”之前进行传输（如果导体是铜线，这意味着熔化）。关于电，总功率也可以被称为总热量，这就是为什么你看到的电灯泡或者微波炉的热功率都是以瓦特为热计量单位的。瓦特不仅仅用来衡量热能，而且用来衡量电功率。

一些电子设备（比如 **Arduino**）因为消耗电能较少，而产生的热量也较少，因此很少注意到散热问题。其他设备专门用来传输大量电流（比如电机控制器），必须使用金属散热片或者风扇来帮助散发设备中的热量。在任何情况下，能够确定电子设备产生的总热量对于我们知道如何正确地使用它是很有帮助的。

1.1.1 电模拟

电通常是看不见的（除非在雷雨天气有可能看见），因此很难理解当你打开一盏灯或者厨房电器时电线内部究竟发生了什么。为了易于描述，我们将电力系统认为是一个底部有出水管道的水箱（如图 1-2 所示）。

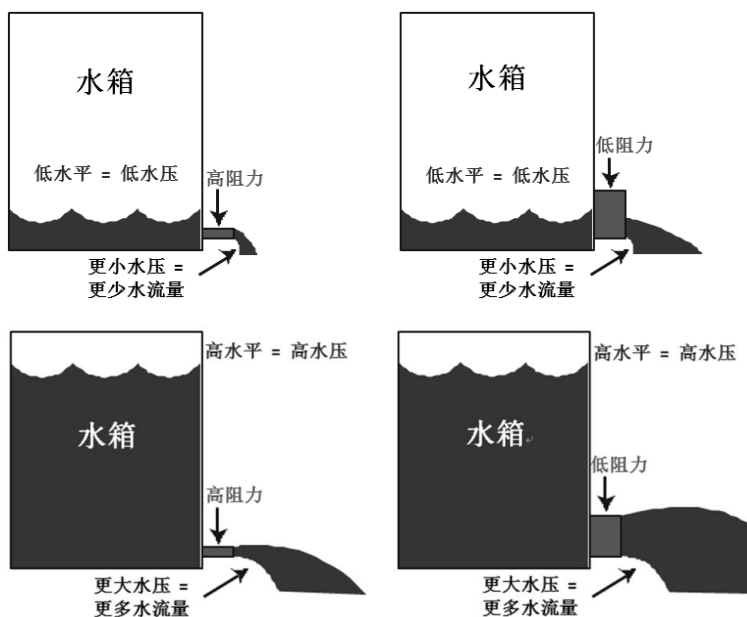


图 1-2 类似电力系统的水箱

以上四个图描述了水箱中阻力和压力如何影响水的输出。阻力越大流出的水量就越少，而压力越大流出的水量越多。你也可以观察到，当水流阻力降低时，即使施加的压力很小，也会有更多的水从水箱里流出来。

水箱中的水越多，从水箱出水管中流出的速度就越快（压力越高）。如果没有出水管，那水箱只是一个水库。实际上，水箱底部有一个让水流出的出水管，但是水流的速度受到出水管尺寸的影响。出水管的尺寸决定了水流出水箱的阻力，所以出水管尺寸的增加或者减少与水流出水箱阻力的增加或者减少成反比例（比如，更小的水管=更大的阻力=更小的水流）。

水的高度（或者压力）和阻力（或者出水管的尺寸）都可以测量出来，使用这些测量结果，你可以计算出在任意给定时刻流出水箱的水流总量。但水流和电流的不同之处在于，电流必须沿着其路径回到电源后才可以被使用。

1.1.2 电的基础知识

请注意，水压越高产生的水流输出就越多（保持阻力不变）。与之相同的是电的等效压力，称之为“工作电压”（ V ），这表示能够在电力系统中发现的潜在能量。系统的工作电压越高，就有越多的能量来驱动系统中的零部件。系统中发现的“电阻”（ R ）阻碍（减缓）电流，就像是出于出水管引起的阻力减缓了水从水箱中流出的速度。这意味着随着电阻的增大，工作电压（压力）也必须增大以保持相同的输出功率总量。每秒钟通过电气系统中的电荷总量（库仑）被称为“安培数”（ I ）或者“电流”，可以使用工作电压、电阻和欧姆定律计算出来。“瓦特”（ P ）是电功率的计量单位，它可通过工作电压乘以电流强度计算得到。本章我们将进一步讨论工作电压、电阻和电流强度。首先，让我们一起来看看它们之间的关系，欧姆定律。

根据维基百科描述（来源：http://en.wikipedia.org/wiki/Ohm's_law），欧姆定律表明，通过导体两端的电流直接正比于通过这两点的电位差或者工作电压。

在工作电压、电阻和安培数（电流）之间有一个简单的关系，可以用数学计算。给定其中的任意两个变量和欧姆定律，你就可以计算出第三个变量的值。瓦特是电功率的计量单位，它与欧姆定律有关，因为它也可以使用这几个同样的变量计算出来。请参阅图 1-3，其中 V 表示工作电压， R 表示电阻值， I 表示电流， P 表示电功率瓦特。

■ **注意：**图 1-3 中用的饼状图由 www.electronics-tutorials.ws 提供。如果你有兴趣学习更多关于电学的知识，应该直接访问这个网址——里面包含一些很有用的插图和描述。

欧姆定律的不同表示包含如下：

$$V = I \times R$$

$$I = V / R$$

$$R = V / I$$

使用下列公式计算总功率：

$$P = V \times I$$

$$P = I^2 \times R$$

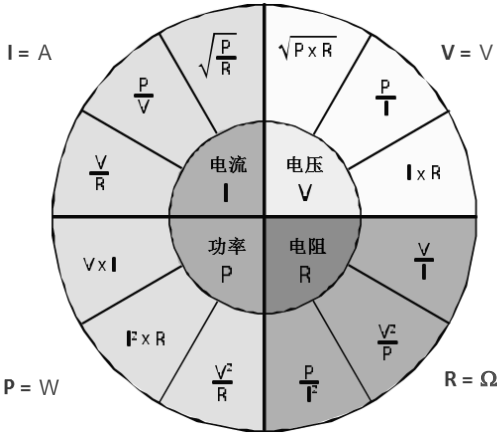


图 1-3 用欧姆定律计算电功率

在学习电气系统时，你可能会无意中发现一些别的术语，我们在此讨论其中的几个。也许你已经知道，电气系统通常使用一条“电源”线和一条“公共”线来构成电路。根据你看到的那样，这两方面对一些事物的叫法是不一样的。为了避免我学习时碰到过的概念混淆，表 1-1 提供了电气系统各种正极和负极名称的简短对比。

表 1-1 电气系统中涉及的正极和负极常用名

偏 压	极 性	电 流	示意图标记	常 用 名
正极	阳极	电源	VCC	电源
负极	阴极	汇点	VSS	接地（GND）

我们讨论了欧姆定律，以及用于描述各种各样电流属性的常用计量单位。表 1-2 提供了标准的电单位和相应符号的列表，我们将会在本书随后的各个章节中用到这些标识符，因此熟悉这些内容不失为一个好的主意。

表 1-2 常用电气计量项以及对应的符号

计 量 项	单 位	符 号
工作电压（能量）	伏特	V 或 E
安培数（电流）	安培	I 或 A
电阻	欧姆	R 或 Ω
电功率（电热量）	瓦特	P 或 W
电容	法拉	F
频率	赫兹	Hz

现在，让我们讨论电气系统中的不同部分。

1.1.3 电路

在电气系统中，电流的起点被称为“电源”，通常指的是电池或者电源供应器的正极。电流从源端开始，通过整个系统到达接收点，一般是电池的负极端或者接地线（GND）。对于电流，整个电路必须是“闭合”的，这就意味着电流可以返回到其起点。

术语“接地”来自将交流电路的返回路径使用铜棒直接连接到外面地上的习惯。你可能会注意到，大多数的电表也会在附近有一个通过导线连接到保险丝盒的接地棒，这条接地线给了返回电流一条退出系统的路径。即使直流电中与 GND 等效的是电池负极端，我们仍旧称之为 GND。

■ **注意：** 电流的电子流实际上是从负极流向正极的，但是除非你是一个物理学家，否则不用关注这点。出于学习的目的，我们假定传统的电流理论是说，在系统中电流是从正极（+）流向负极（-）的。

电气系统被称为“电路”，简单来说可以像一串插入电源插座的圣诞彩灯，或者复杂来说像你的计算机主板。我们现在仔细想一下，电路中，电子流只有在某些东西，被称为“负载”（如图 1-4 所示）构成电路时才能流动。一般情况下，电路中的负载指的是你打算提供用电流的设备。这可以是灯泡、电动机、加热线圈、扬声器、计算机 CPU，或者任何其他电路准备供电的设备。

通常情况下，有三种类型的电路：开路、闭路和短路。基本上，开路是被断开的电路，闭路是被接通的电路，短路是需要修复的电路（除非你使用了保险丝）。这是因为短路意味着电流存在一条路径，绕开了所有负载直接将电池的正极端连接到电池的负极端。这终归不是一件好的事情，通常会导致火花和一团烟雾，并偶尔伴随着较大的爆裂声。

在图 1-4 中，灯泡是该电路的负载，左边的开关决定电路是断开的还是接通的。左侧图片展示了没有电流通过负载的开路，右侧图片展示了向负载供电的闭路。

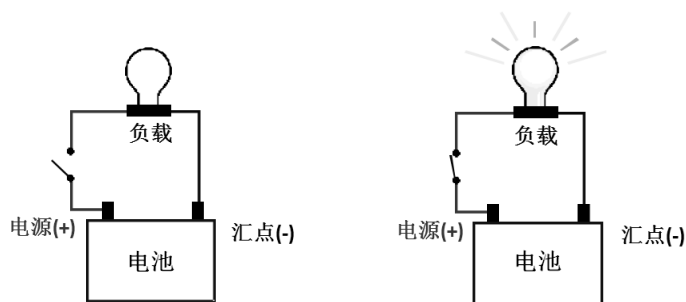


图 1-4 开路和闭路

1.1.4 电信号测量

如果没有一种方法来测量电信号的话，我们会像无头苍蝇一样——幸运的是，有一种叫做“万用表”的设备，它比较便宜，而且能够很容易地测量出工作电压、电阻和低量级的电流值。

1.1.5 万用表

有很多种不同类型的万用表，它们拥有不同的功能，但是我们所需要的只是一个能够测量直流工作电压可达 50V 的基本测量仪。

典型的万用表可以测量信号的电压值和部件或者负载的电阻。因为给定了电压和电阻，就可以计算出相应的电流，实际上你所要做的全部工作就是基本的电路测试。全功能的数字万用表如图 1-5 左图所示，虽然售价 50 美元左右，但是通常你可以找到一个售价低于 10 美元的简单模拟万用表如图 1-5 右图所示，它既可以测工作电压，也可以测量电阻。这两种万用表都可以完成基本测试，虽然数字万用表表现得要更好一些，但其实我想拥有一个便宜的模拟万用表来测量电阻，因为可以根据指针移动到测量值的速度来判断信号的强度。

标准万用表都有两个连接到基极的绝缘探针，用来连接到被测量的电子设备上。如果想测量电路或者电池的工作电压，那么应该将红色探针（连接到万用表“V, Ω , A”端）放到电池的正极端，再将黑色探针（连接到万用表“COM”端）放到电池的负极端或者接地。



图 1-5 华仪 MN16a 型数字万用表（左）可以测量交流电和直流电的电压、电阻、连续性（即电路通不通）、二极管测试、电容、频率、温度和高达 10A 的电流。本地五金店购买的廉价模拟万用表（右）可以测量交流电和直流电的电压、电阻（1k Ω ）和至多 150mA（0.15A）的电流。不管是诊断 Arduino，还是其他大多数的电路——你肯定需要一个万用表

1.1.6 电压测量

电压既可以测量家庭插座中那种类型的交流电，也可以测量电池中的直流电。万用表应该根据读取到的正确电压类型进行相应设置，一些万用表也有测量范围，在测量电压之前你需要设置。图 1-5 右图的模拟万用表被设置为 10VDC，指针的有效设置范围从 0~10VDC。

如果试图读取的电压值远高于所选择范围，就会导致万用表保险丝烧断，所以你应该始终使用高于实际测量电压的电压范围。如果你不确定要测试的是多大的电压电平，最好选择最大范围设置（该万用表上是 300VDC）。图 1-5 左图的万用表有直流电和交流电设置，只要保证所测电压不会超过万用表使用说明书上规定的最大额定电压，它就可以自动检测电压范围，然后将精确的电压值显示在屏幕上。

电信号的电压电平也决定了它是否能够将你的身体作为导体。根据个人的身材（水分含量、皮肤厚度等），强制要通过人体的电压电平也可能不尽相同，但是我可以证实，当

你站在地面上，如果不慎碰到了墙上 120V 交流电的电源插座，即便是穿着胶底鞋，也会引起肌肉痉挛。

■ **警示：**电压电平超过 40V 时会对人类和宠物造成伤害。时刻牢记，如果你不想最后躺在医院病床上，在电路上作业时请断开电源，并且使用绝缘工具（和橡胶柄）测试电路。

1.1.7 电流强度测量

大多数的万用表都可以测量少量交流电或者直流电的电流强度（250mA 甚至更小）。图 1-5（左）中的数字万用表在几秒钟的时间里可以测量大至 10A 的电流，然而功能弱一些的万用表只能测量最大值为 150mA 的电流。为了测量很大的电流（超过 10A），根据实际应用你需要一个电流传感器、电流表或者电压钳。

计量单位取决于工作电压和电路电阻。随着工作电压的降低（电池放电）或者电阻浮动，电流强度也会发生变化。一个大型机器人连续行走时，当它越过一块石头或者爬上一段斜坡，电流强度都会发生改变。这是因为直流电机的电阻值越大，其消耗的电流也就越多。从另一方面来说，LED 手电筒会消耗稳定的电流（每个 LED 大概 20~100mA），直到电池电量耗尽。

你也许已经注意到，电池容量的单位是安培/小时（Ah），表示其可以提供的电流和时间长短。其大概的意思是，规格为 6V、12Ah 的电池可以供应一个 6V 的电灯泡在 1A 的电流下工作 12 个小时，或者同样 6V 的电灯泡在 12A 电流下工作 1 个小时。你可能也注意到，更小一些电池（比如普通的 AA 电池）的容量单位是毫安/小时（mAh）。这样，一个 2200mAh 的电池就相当于 2.2Ah。

1.1.8 电容测量

电容表示一个设备中所能存储的电荷量，用法拉来计量，但是 1 法拉（F）是一个巨大的电容值，所以你会注意到，大多数项目中使用的电容值都是在微法（ μF ）级范围。电容器是一种电子设备，它能够存储电荷，并且根据需要向电路中其他元件提供电荷。虽然电容器很像电池，但是每秒钟它可以多次被完全放尽并且再充电——电容量的大小决定该电容器能多快地进行放电和充电。

一些万用表可以测量出电路中任意两点之间的电容值（或者电容器的电容值），比如图 1-5 中的华仪 MN16a。绝大多数万用表无法测量电容值，那是因为通常情况下多数电路

中测量它意义不大。当试图要获得一些特定的值或者测试一个电容器时，能够测试电容就会很有用，但是一般不会用到万用表的这项功能。

■ **警示：**更大的电容器可以保存有效电荷很长一段时间，接触带电电容器的引线会导致触电。CRT 计算机显示器或者电视机中的电容器、电机的启动电容器，甚至是一次性照相机中的小电容器都会产生电击，让你的手臂感觉刺痛几分钟，甚至灼伤你的皮肤。使用绝缘螺丝刀将电容器引脚连接在一起产生短路，释放掉存储电流，然后再去尝试处理它，这是一个不错的主意。

1.1.9 电阻测量

电阻使用欧姆作为计量单位，告诉我们一个导体导电性的好坏。电流和电阻成反比例关系，当电阻值增大时，电流值会降低。这样，电阻值较小的导体比电阻值大的要传输更多电流。每个导体都会有会一定的电阻——有些材料对电流有很大的抵抗作用，它们被称为“绝缘体”，这意味着它们不能导电。当电流通过导体受到阻力时，转化为热能。正是由于这个原因，我们尽可能使用电阻值最小的导体来避免产生热。

电阻器是一种电子设备，有已知的电阻值，单位为欧姆，主要用来限制通过它的电流（如图 1-6 所示）。

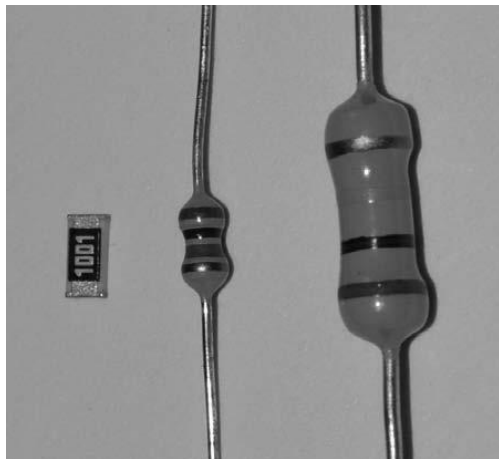


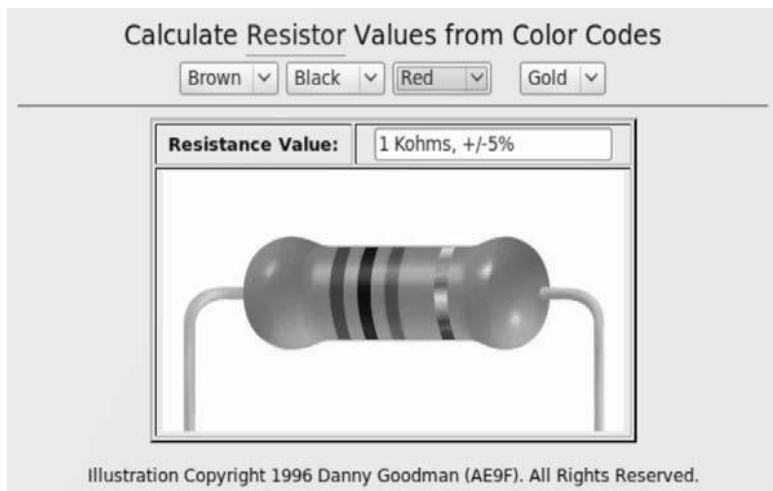
图 1-6 三个电阻：1/4W 表面贴装电阻（左），1/8W 通孔电阻（中）和 1/4W 通孔电阻（右）

请注意，1/4W 的表面贴装电阻（左）**比**等效 1/4W 通孔电阻（右）要小很多，但是两者消耗的能量相同。我通常使用 1/8W 通孔电阻，由于它们体积小，在工作中用起来很方便。

为了确保设备在安全的工作范围内运行，可以通过元件内置的电阻来限制传输到设备中的电流。

贴片电阻表面的数字标出了它的电阻值，单位为欧姆，而通孔电阻上的彩色编码条纹标出了它的电阻值。如果你想手动测量组件的电阻值，将万用表设置到**欧姆档**使用，极性无所谓，除非你测量的是二极管或者晶体管的电阻值。

我使用一个网页，只要你输入电阻上每个条纹的颜色，它就会告诉你电阻的欧姆值（如图 1-7 所示）。在设计或者确定大概电阻值时，它对快速参考很有帮助。请访问网址：<http://www.dannyg.com/examples/res2/resistor.htm>。



取自 Danny Goodman 使用许可的图片

图 1-7 这个缩略图是 Danny Goodman 设计的网页应用。我将这个网页保存在我浏览器的书签中，并且经常使用它来查看不熟悉电阻的色标

1.1.10 使用欧姆定律计算电阻功率

请记住，任何时候电路中都会有电阻，也就会产生热，为了选择一个可以有足够额定功率的电阻，计算出有多少热量将会通过电阻（取决于负载）总是一个很好的主意。电阻不仅以欧姆评定，也可以通过它们持续消散（去除）能量的多少来评定。常见的额定功率有 1/8W、1/4W、1/2W 等，其中较大的电阻的功耗值往往也比较大，除非采用表面贴装电阻（如图 1-6 所示）。

为了计算电阻中的能量消耗，你需要知道电路电压和电阻的欧姆值。首先，我们需要使用欧姆定律来确定将要通过电阻的电流，然后使用电阻和电流值计算总热量，也就是被电阻消耗的瓦特值。

例如，如果我们有一个 1000 欧姆的电阻（1k Ω ）和一个 12V 的电源，允许通过电阻的电流是多少呢？电阻的最小额定功率又是多少？

我们首先使用欧姆定律计算通过电阻的电流值：

$$V = I \times R$$

$$I = V / R$$

$$I = 12V / 1000\Omega$$

$$I = 0.012A \text{ 或 } 12mA$$

现在我们使用电流来计算总功率（热量）：

$$P = I^2 \times R$$

$$P = (0.012A \times 0.012A) \times 1000\Omega$$

$$P = 0.144W$$

计算出来的总功率是 0.144W，这意味着应该使用额定功率大于 0.144W 的电阻。而常用的电阻值通常是 1/8W（0.125W）、1/4W（0.25W）、1/2W（0.5W）等，我们应该使用一个额定功率至少为 1/4W（常见的大小）的电阻，并且仍旧能够安全地消耗 0.144W 功率。如果将大一点的电阻连接到电路中，那么使用一个 1/2W 的电阻不会烧坏任何东西——它只会简单地比同样电阻值的 1/4W 电阻更容易传输热量。

现在，应当能够计算出电阻的额定功率是否适合应用。接下来，让我们一起来谈谈其他类型的负载元件。

1.1.11 示波器

虽然万用表非常适合测量电压、电阻和电流，但是它只是有时候能有助于准确地查看电信号里发生的事情。还有另外一种被设计用来分析电信号的设备，称之为“示波器”。示波器可以探测电信号中的重复模式和振荡，然后在其屏幕上显示信号的波形图。对于电信号，它实际上是一个显微镜。这些机器都非常昂贵（500~5000 美元），直到最近，一些业

余爱好者用的示波器才进入市场，售价低于 100 美元。

开源数字示波器 DSO Nano(如图 1-8 所示)由 seeedstudio.com 研发,通过 Sparkfun.com (零件号 TOL-10244) 在美国出售。我已经拥有这样一台示波器接近一年了,而且使用比较频繁,这是因为它使用起来很简单,大小和重量只相当于一部手机,成本大约 89 美元。它含有可充电锂电池,能通过迷你 USB 数据线充电,还拥有可以存放所读数据的内存卡槽,以便于之后在计算机上查看。

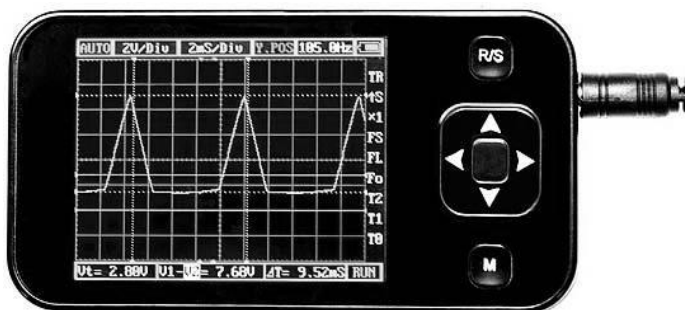


图 1-8 购自 seeedstudio.com 的 DSO Nano (在 Sparkfun.com 有售) 是个相当不错的选择, 因为别看它价格便宜, 却是个功能齐全的数字袖珍示波器

虽然在诊断电信号的时候示波器是一个很贵重的工具, 但是本书所涉及的项目都用不到它。你可以从一个简单的万用表上获得读数。Sparkfun.com 上也有其他廉价的示波器可供选择, 包括 DIY 工具套件, 售价 60 美元左右 (零件号 KIT-09484)。

1.1.12 负载

“负载”指的是电路中的用电设备。负载有很多不同的例子, 从直流电动机到发光二极管, 再到加热线圈, 每一个都会在电路中产生不同的反应。例如, 加热线圈 (存在于吹风机或者小型取暖器中) 是由金属制成的简易螺旋状电阻丝, 它加热时可以变得通红, 但是不会熔化。然而电动机使用电流可在线圈周围产生一个电磁场, 从而导致电动机轴的物理转动。我们所关注的负载有两种类型: 电感和电阻。

1. 电感性负载

如果向一个设备供电, 它就会产生活动的能量, 那么它可能就是一个电感性负载。这包括电机、继电器和螺线管。电感性负载被通电后会产生一个电磁场, 通常断电后一段时间电磁场才会消失。当使用开关断开电源之后, 磁场消退并且剩余电流转储回电源端, 这

种现象被称为反电动势（电动势），如果没有整流二极管的保护，它会损坏电路中的开关元件。

2. 电阻性负载

电阻性负载使用电流产生光，或者其他一些形式的热，而不是机械运动。这包括发光二极管、发热元件、电灯泡灯丝、焊接机、电烙铁和其他。电阻性负载使用的电流固定，因为它们的负载不会受外界因素的影响。

1.1.13 电路连接

在建立一个电路之前，应当确定所需要的工作电压，然后再选择连入电路的组件。虽然降低交流电电压的电平需要使用变压器，但是通过使用不同的布线法连接多个单独的电池组也能实现特定的直流电电压的电平。有两种不同的电路连接方式：串联和并联。

1. 串联

将电路布置成“串联”是指将多个设备连入电路中，首尾依次连接。经常把电池串联起来以获得更高的电压值。为了演示这个电路，我们使用两个 6V 10Ah 的电池，第一块电池的正极端连接到第二块电池的负极端。现在唯一打开的端点是第一块的负极端和第二块的正极端，这将产生一个 12V 的差值。

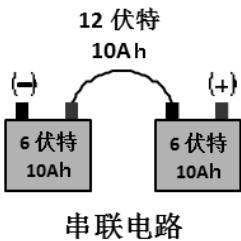


图 1-9 两块电池设置成串联电路会导致电压值加倍，而安培/小时容量值不变

当两块电池被设置成串联电路（如图 1-9 所示）时，电压值会加倍，但是安培/小时的容量保持不变。这样，两个 6V 10Ah 的电池一同工作会产生一个 12V 10Ah 的电池组。这一技术对于实现特定的电压电平很有帮助。

2. 并联电路

将电路设置成“并联”是指将所有的公共端放置在一起。这意味着所有正极端都连接起来，并且所有负极端连接起来。如果将两个 6V 10Ah 的电池按照前面的例子设置成并联电路（如图 1-10 所示），电压值会保持不变，但是安培/小时的容量会加倍，最后连接成一个 6V 20Ah 的电池组。

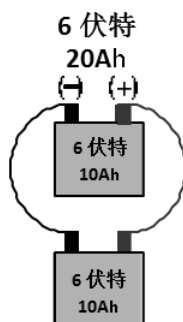


图 1-10 两个电池设置成并联电路会导致电压不变，但是安培/小时容量值加倍

3. 串并联

为了实现一个特定的电压值和安培/小时值，还可以将几个电池组同时按照串联和并联方式设置。请注意，两组 6V 10Ah 的电池被设置成串联以产生 12V 电压，然后两个串联的电池组被设置成并联以产生不变的电压值，但是容量为 20Ah。

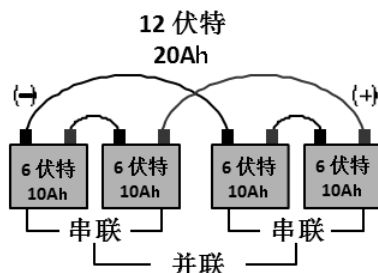


图 1-11 通过将串联的两个电池组并联起来的方法，可以使用四个 6V 10Ah 的电池组创建 12V 电流容量为 20Ah 的电池组

在创建一个电池组时，重要的是使用电压和 Ah 容量都相同的电池来创建更大的单元。这意味着，不可以将 12V 电池与 6V 电池组合在一起来实现 18V，而是要使用三个 6V 且容量相同的电池实现 18V，并且避免不均匀地充电/放电。

1.2 电子学

电子学领域主要的问题是控制电路中电流的流动，特别是使用电子开关。在电子开关发明之前，使用机械开关来打开和闭合电路，这就需要机械运动（比如用你的手控制开关向上或向下）来连接或者断开电路。虽然机械开关完全可以被使用，甚至在一些应用中优

先被选择，但是由于在切换过程中必须发生物理运动，故它们受限于切换速度到底有多快。就是机电开关（被称为继电器）也不合适用做电子设备，因为它利用电来产生激活开关的机械动作。

电子开关通过使用设备内部的电感应替代了机械的开关动作，这样一来就不需要活动的部件了。没有了物理运动，这些设备切换速度极快，并且可靠性更高。用来制造开关的材料只有在特定情况下才会导电。通常，向设备的输入端和输出端施加特定的电压或电流就能打开或者关闭它。当设备被打开时，它以一定量的电阻进行导电；当设备被关闭时，它就变成绝缘体，不再导电。这种类型的电子器件被称为“半导体”，因为它会根据相应的电气状况变成导体或者绝缘体。

1.2.1 半导体

半导体的应用取代了机械开关，使电路“电子化”，因为它能够以极高的速度切换电子信号，而这对于机械式电路根本是不可能的。有很多不同的半导体，但我们只讨论在大多数电路中常用的几种重要类型。

- 二极管：就像电流的单向阀，该设备只能够使电流在一个方向上通过——二极管非常有用，而且也是所有固态电子元件的基础。
- 发光二极管（LED）：当有电流通过它时，这种二极管发出微弱的光线。
- 光敏电阻（LDR）：这种类型半导体的阻值根据周围光线的亮度不断地变化。
- 双极面结型晶体管（BJT）：这是一个电流驱动的电子开关，主要使用了其快速切换的特性。
- 金属氧化物半导体场效应晶体管（moset）：这是一个电压驱动的电子开关，使用了其快速切换的特性，电阻低，并且能够在并联电路中工作。这是大多数功率放大器电路的基础。

这些器件都有多层携带正电荷和负电荷的硅，连到带有裸露金属导电引脚的芯片上，以便焊接到电路上。一些晶体管和场效应管内嵌二极管，以保护它们免受反向电压和反电动势的损害，所以再看一下你正在使用的元件的技术手册总是一个很好的主意。

1.2.2 技术手册

每个器件都应该有相应的技术手册，可以从制造商那里得到，通常是从其官方网站上下载的。技术手册中包含了关于器件的所有重要电气信息。上限一般被称为“绝对最大额定值”，告诉用户该器件什么时候会失效（如图 1-12 所示）。下限（如果可以）告诉你器件在什么电平时不再会对输入产生响应，这通常不会损坏器件，只是不工作。

还有一节被称为“电气特性”，它会告诉你设备在什么情况下正常运行。这一般会显示打开或者关闭设备的确切电压或电流电平。这些额定值有助于确定应选择其他什么样的元件值（比如电阻和电容），或者设备是否会按照预期目的正常工作。

技术手册通常会告诉你比“能做什么”更多的信息，并以图示和封装尺寸作为结束。一些技术手册甚至有电路布局的推荐规范，并给出了微控制器元件接口定义方法的建议。对于畅销或者常用的元件，也可以访问制造商的网址来获得进一步描述如何使用元件的补充文件，这些被称为“操作说明书”，能够加深你的认识，如图 1-12 所示。

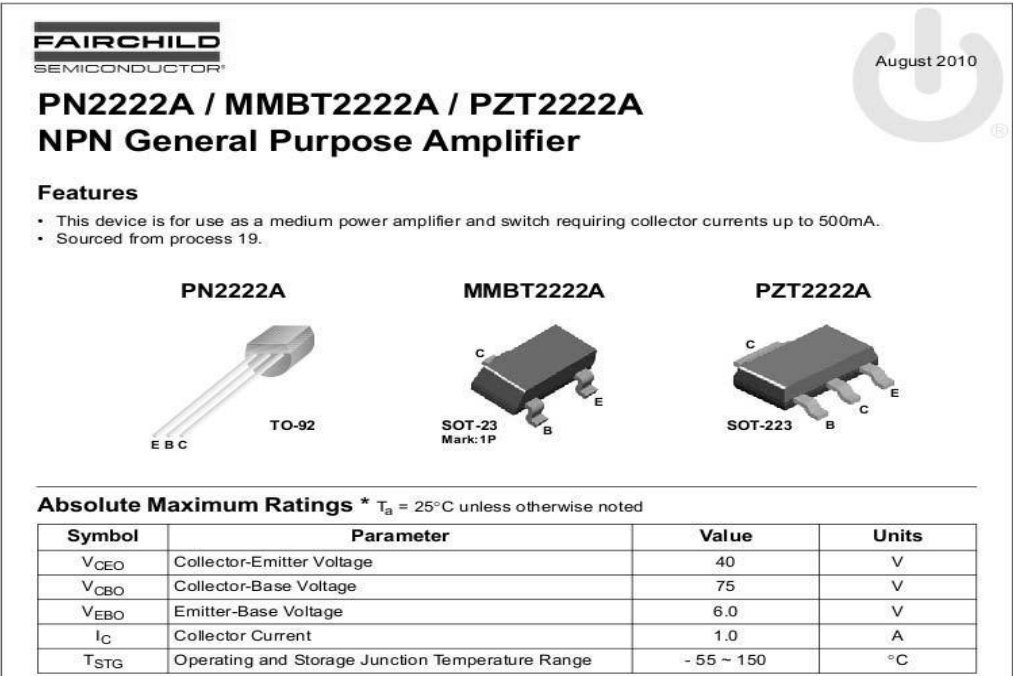


图 1-12 这里，你可以看到美国仙童半导体公司著名的 2n2222NPN 型半导体开关样例技术手册的第一页。它首先列出了可用的封装及引脚配置，然后是绝对最大额定值的简短列表

1.2.3 集成电路

一些包含多个元件的半导体被封装在一个芯片上，被称为集成电路（IC）。一个微型芯片上的集成电路往往能够容纳成千上万个晶体管、二极管、电阻和逻辑门（如图 1-13 所示）。这些组件可被用在大一些的“通孔”封装内，较新的型号则用于超小型的“表面贴装”芯片上。

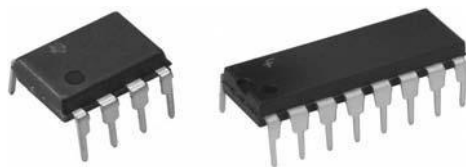


图 1-13 这里，你可以看见一个 8 引脚的双列封装（DIP）集成电路（左）和一个 16 引脚的 DIP 集成电路（右）。Arduino Atmega168/328 是一个 28 引脚的 DIP 集成电路（每侧有 14 个引脚）

封装

我们在不同的封装中使用不同类型的半导体。组件封装指的是已有的物理外观、尺寸和引脚配置，根据半导体类型，不同的封装考虑各种不同的散热性。如果选择大功率，那么一般容器越大，散热性也就越好。对于低功率电路，理想的设计通常是元件尽可能地紧凑，所以封装的尺寸越小，就越能吸引人的兴趣。我们使用的常见封装是 TO-92 和 TO-220（如图 1-14 所示），它可以封装从温度传感器到晶体管、再到二极管的任何东西。

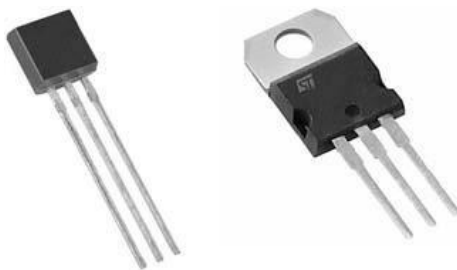


图 1-14 较小的 TO-92 型集成电路封装（左）适用于低功率的电压调节器、信号晶体管和传感器集成电路。大一些的 TO-220 型（右）适用于更大功率的电压调节器，功率场效应管开关和大功率二极管

TO-92 是一种较小的封装，通常适用于低功率晶体管开关和传感器。而 TO-220 通常适用于大功率应用，并且它是大多数功率场效应晶体管的基础，在芯片上的金属引脚失效之前其工作电流能接近 75A。TO-220 封装还包含一个内置的金属片来帮助散热，并且如果需要还允许连一个附加的散热器。

1.2.4 通孔元件

在这整本书中，都在寻找创建和修改我们项目的最简单方法。这通常意味着如果需要的话，可以很简单地更换使用的元件，同时也可以使用大一些的元件，足以让初学者感觉很方便地焊接到位。就半导体而言，术语“通孔”指的是任何元件，其引脚穿过钻在印刷电路板上的孔、可焊接到电路板底部的铜“垫”上。那些部件通常都很大，即使是一个初学者也很容易焊接到电路板上。很多通孔元件的引脚比需要的长很多，因此建议将元件焊接到合适的位置之后，剪断每个引脚底部多余的部分以避免任何电路板底部的短路。

1. 集成电路插座

“集成电路插座”是一个有金属触点的塑料基板，它往往被焊接到电路板上（如图 1-15 所示）。集成电路在焊接完成之后插到插座上，以减轻集成电路在焊接过程中过热的风险。如果问题是在电路出错，这也是有帮助的。由于不需要额外焊接，集成电路也很容易被替换。由于这些原因，所以我们总是使用集成电路插座。

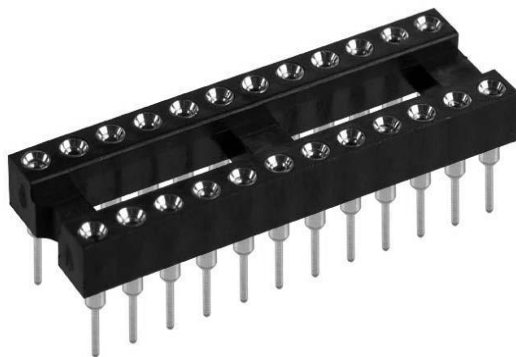


图 1-15 为了放置实际的集成电路，电路一建立就将集成电路插座焊接到 PCB 上。

这些插座通常每个不到 1 美元，所以只要可能我都尽量使用它们

2. 表面贴装元件（SMT 或者 SMD）

近些年，随着制造业的技术飞跃，较小元件的性能已变得越来越好。这导致了元件和集成电路尺寸的减小，以致能够创造出更小的设备，并能完成与大一些元件一样的事情。

虽然这些设备的内部是一样的，但是它们的引线管脚却要小很多，当初学者试着将它焊接到一个电路板上时会有些让人沮丧（如图 1-6 左图所示的表面贴装电阻）。这些元件和通孔元件之间最主要的区别是它们被焊接在电路板的上面，而且在电路板上不需要钻

孔。它们通常与电路板靠得很近，只需要很小的空间来安装，使它们适用于节省空间的应用。

一些表面贴装元件有裸露的端点，能够以正常方式进行焊接，但是另一些只是在芯片的底部有裸露端点，要求在表面贴装回焊炉中焊接它们。尽管可以使用烤箱模拟临时凑合的回焊炉，但本书中，打算在建立的电路中不使用表面贴装元件，以避免因使用表面贴装元件（SMD）元件而增加难度。

■ **注意：**在第 8 章中，由于找不到完成项目所需要的通孔元件，所以不得不使用表面贴装芯片。我寻找了一个最大的来用，那样焊接起来容易一些，而且比我想象的还要容易。

先避开一些电子学的术语和定义，我们该转到一些 Arduino 的特定主题上。

1.3 Arduino 初级读本

Arduino 是可编程的，使用基于功能强大的 AVR 微控制器，有 20 个 I/O 引脚，并且很便宜，每个组装电路板大约 30 美元。主 Arduino 使用标准的 USB 数据线连接到计算机，不仅提供了到 PC 的串行连接，还包括操作所需要的 5V 电源（使用 USB 数据线时不需要电池）。

Arduino 团队还开发了运行在计算机上的程序（可用于 Windows、Mac 和 Linux），用来编译你的代码，并且很容易上传至 Arduino 板。Arduino 板有一个 USB 适配器芯片（FTDI），插入时计算机可以把它当做串行设备来识别。编程所需要的大部分最新驱动程序和软件都可以从网站 www.arduino.cc 上免费下载。查看 Arduino 主页的“入门”一节，按照 Arduino 软件的安装指导一步一步地安装到你指定的操作系统中：<http://arduino.cc/en/Guide/HomePage>。

Arduino 软件被称为一个集成开发环境（IDE）。这是一个用于将代码上传至 Arduino 微控制器的开发软件。这个 IDE 包含一个文本编辑器和编译器，将简单的 Arduino 编程语言（我们编写的）编译成更复杂的能够直接被上传至微控制器的二进制文件。

Arduino 语言是 C++ 编程语言的一个变种，并且使用内置库文件来简化复杂的编码任务，使得初学者更容易上手。如果你之前没有任何编程经验，那么你将会从 Arduino 参考页中受益很多。这些页面展示每个 Arduino 命令，以及通过一个样例代码片段讲解如何使

用它。你可以访问 **Arduino** 网站查看这些页面，或者查看 **Arduino IDE** 的“帮助>参考”：
<http://www.arduino.cc/en/Reference/HomePage>。

因为 **Arduino** 语言是一个开源项目，所以它会不断地被改进和更新。经常会发布新版本的 **Arduino IDE**，所以最好使用最新的可用版本进行系统更新。本书中的大多数项目都是使用 IDE0019-0021 完成的，可以在 **Arduino** 主页上下载。

1.3.1 **Arduino** 变体

Arduino 包含许多不同形状和尺寸，但是只有两个型号使用完全不同的芯片：标准版和 Mega 版。标准版是基本的 **Arduino**，使用了 Atmega8/168/328 芯片，然而 Mega 版是不同的 **Arduino** 板，它拥有更多的 I/O 引脚，并且使用功能更强大的 Atmega1280 芯片。因为 **Arduino** 的设计是开源的，所以任何人都可以根据自己的意愿设计新版本的 **Arduino** 板，然后进行销售。正是出于这个原因，别的一些制造商已经制作出了 **Arduino** “克隆板”，可以当做标准版 **Arduino** 来工作，只是由第三方制造或者作为工具包提供作自行构建用。

也有一些 **Arduino** 板不具备板载 USB 转换器，所以必须使用一个特定的 USB (FTDI) 编程电缆对它们进行编程（如图 1-16 左图所示）。FTDI 编程电缆可以从 Sparkfun.com 上以大约 20 美元买到（零件号 EV-09718）。在单独的编程电缆上使用 FTDI 芯片，而不是使用 **Arduino** 板本身，这样做的好处是只要使用一个 Atmega328 芯片、16MHz 谐振器和其他一些很容易找到的元件就可以制作自己的 **Arduino** 板。如果增加几个头管脚，甚至能设计你自制 **Arduino** 板的内部电路（如图 1-16 所示）。

从 Sparkfun.com 上购买了 FTDI 编程电缆之后，我又去了一个官方授权的场所制作了 15 个不同的 **Arduino** 克隆板，它们有不同的引脚配置、螺丝接头、R/C 头、供电的伺服插头，甚至一些可分层的 **Arduino** 扩展板。虽然自制的电路板都没有板载 USB 功能，但是其中一些有 6 引脚 FTDI 编程的头来进行在线编程。这样一来，我制作每个电路板只要购买 8 美元的零部件。如果你偏爱原型设计，这是最省钱的实现方法。

在图 1-16 中，你可能注意到自制 **Arduino** 板部件很少。这是因为要让自制的 **Arduino** 板工作只需要三个必不可少的部件：Atmega168 芯片、16MHz 谐振器和+5V 电压的调压器。电容、LED 电源灯、头插脚和复原按钮都不是必需的，但是推荐使用，可提高可靠性，并且也易于集成到项目中。

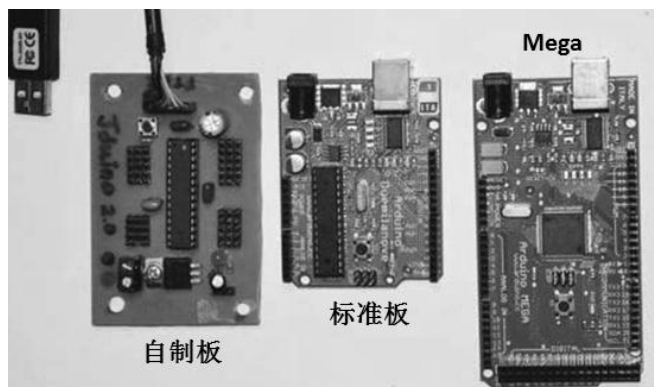


图 1-16 三种不同类型的 Arduino 板

需要注意的是，左边自制的变体电路板像标准版 Arduino 一样使用了 Atmega168 芯片，但是使用 FTDI 编程电缆来编程；中间的电路板是标准版 Arduino Doemilanove；右边最后一块电路板是 Arduino Mega。

1. 标准版 Arduino

标准版 Arduino 最初是基于爱特梅尔的 Atmega8 芯片制作的，采用 28 引脚的微控制器，总共 20 个可用输入/输出（I/O）。在 20 个可控制引脚中，6 个用作模拟输入，6 个用作脉冲宽度调制（PWM）输出，2 个可用于外部中断。标准版 Arduino 运行频率为 16MHz，包含三个可调计时器来允许修改 PWM 频率（将在本章随后讨论）。

有另外两个和这个芯片引脚兼容的变体，Atmega168 和 Atmega328 都包含了比之前更大的板载内存。新版本的标准版 Arduino 使用新的 Atmega328 芯片取代了旧的 Atmega8/168 芯片。如果你有一个旧型号 Arduino，并且想升级到拥有更大内存的新芯片，可以用 5.5 美元左右购买一个新的 Atmega328，然后只要将它插进你现有的 Arduino（这些芯片引脚兼容且物理上相同）。如你草稿中要用的内存比 Atmega8 现有的内存还要多的话，那就成问题了——这是一个针对高级用户和较大工程的问题。

这个芯片关键的优势之一是它可用于通孔封装集成电路中，该集成电路可以从 Arduino 板上去除，并且很容易安装在面包板（实验模型）中或者焊接在穿孔的原型板上，用来制作一个单独的 Arduino 克隆板，以便在项目中长久使用。通孔 Atmega328 芯片搭配一个 28-DIP 集成电路插座，特别适用于原型设计。

■ **注意：**如果你突然损坏了 **Arduino** 上的一个引脚，最可能的补救措施是使用一个新的来替换 Atmega168/328，它们每个大概 5.5 美元，你可以和 **Arduino** 预装引导装载程序一起从 **Sparkfun.com**（零件号 DEV-09217）购买。虽然这种事情已经发生好多次了，我仍然在使用我的第一块 **Arduino** 板。

2. **Arduino Mega**

Arduino Mega 是另一种型号，使用了功能更强大的 Atmega1280 芯片，就像一个增强型的标准版 **Arduino**，总共 70 个 I/O 引脚（如图 1-16 右图所示）。其中有 16 个模拟输入，12 个 PWM 输出和 6 个可用外部中断。同样的软件可用于 **Arduino** 所有的型号，**Arduino** 语言的每个命令在每个设备上都可以运行。

这个型号只适用于安装到电路板表面的 Atmega1280，无法去除，这样，与标准版 **Arduino** 相比就限制了它的通用性。这个电路板的开发成本在 75 美元左右，但是一些公司已经引进了大概 45 美元的 **Arduino Mega** 克隆板。如果你有能力购买特大号的 **Arduino**，在不改动任何硬件的情况下，根据需要有更多的 I/O 引脚也是很不错的。

3. 克隆板

虽然只有两种型号，它们使用不同的处理芯片，但很多情况下还有无数可供你制作和购买的 **Arduino** 克隆板在互联网上流传。**Arduino** 克隆板不是官方支持的 **Arduino** 板，然而每个克隆板可能有它自己特定的引脚设置、尺寸和预期目的。所有要求与 **Arduino** 兼容的都可以使用 **Arduino IDE** 软件上传 **Arduino** 代码。

甚至有克隆板偏离了标准的硬件规格，但仍可以被 **Arduino IDE** 所支持，比如使用 3.3V 和 8MHz 取代 5V 和 16MHz 作为标准的 **Arduino Pro Mini**。你可以在任何 **Arduino** 克隆板上使用 **Arduino IDE** 软件，但是必须在工具菜单中选择正确的电路板类型。

总而言之，在本书中你购买什么样的 **Arduino** 作为开始都不重要——只要是提及的 **Arduino**，它都应该很好地工作。在一些项目中我们特别指定使用标准版 **Arduino**，在一个项目中使用 **Arduino Mega**，其中在一个章节中使用 **Ardupilot**（带有 GPS 的 **Arduino**），并且还有一些自制的 **Arduino** 克隆板。现在，让我们来看一下 **Arduino IDE**，以便更好地了解它是如何工作的。

1.3.2 Arduino 集成开发环境

假设你已经按照指导下载并安装了 Arduino IDE，现在你需要打开程序。第一次打开计算机上的 Arduino IDE，它可能会问你想要将“草稿集”放在哪里（如果你使用 Windows 或者 Linux）。如果使用的是 Mac，你的草稿集会自动在 user/documents/Arduino 目录下创建。草稿集是指 IDE 将要存放所有你在 IDE 中创建草稿的文件夹。你选择了草稿集文件夹之后，它所有的内容将会显示在 File > Sketchbook 菜单里。

在打开的 IDE 窗口中，你会发现一个为你准备好输入代码的空白窗口，在 IDE 屏幕顶部有提供常用命令快捷键的蓝色工具栏（如图 1-17 所示）。表 1-3 提供了对每个快捷键的描述。







图 1-17 在 IDE 顶部有一个包含常见任务快捷键的工具栏。当使用 IDE 时，将鼠标光标悬停在每个按钮上就可以看见相应的描述




1.3.3 草稿

草稿仅仅是一组可以在 Arduino 上执行的指令的集合。使用 Arduino IDE 创建的草稿被存为.pde 格式的文件。为了创建一个草稿，你需要完成三个主要部分：变量声明、Setup 函数和主要的 Loop 函数。

表 1-3 Arduino IDE 工具栏按钮

工具栏按钮	描 述
	编译：该按钮用于检查“语法”或者代码的正确性。如果你有任何错误标记的东西或未定义过的变量，将会在 IDE 屏幕底部看见以红色字母标注的错误代码。但是，如果代码正确，会看到“编译完成”的消息和以千字节计的代码大小。这个按钮再按下时可检查代码中的错误
	停止：如果你正在运行与计算机进行通信的程序，按下这个按钮将会终止程序
	新建：这个按钮清除屏幕，并且让你在一个空白页开始工作
	打开：这个按钮让你从文件中打开一个已有的草稿，当需要打开一个你已经下载的或者以前使用过的文件时，会用到它

续表

工具栏按钮	描 述
	保存：选择这个按钮来保存你当前的工作
	上传：这是一个神奇的按钮，可以通过它将代码上传至 Arduino 。在 IDE 试图将代码上传至电路板之前先会对代码进行编译，但是我总是会在上传之前按编译键。如果从 Tools>Board 菜单中选择了错误的电路板类型，会得到错误的提示信息
	串口监控器：串口监控器是一个调试的工具（指出什么是错误的）。 Arduino 语言包含打印从 Arduino 的循环函数中收集的数值的命令，并将它们打印到计算机屏幕上，这样你就可以看见了。如果没有获得你预期的结果，这个功能会对你特别有帮助，因为它会告诉你到底是怎么回事。在安装到项目之前，我普遍使用这个功能来测试代码

1. 变量声明

变量声明是一个奇特的术语，意味着需要指定你想在草稿中使用的每个输入或者输出名字的类型。你可以使用任何名字来重命名 **Arduino** 的输入/输出引脚号码（比如 `led_pin`，`led`，`my_led`，`led2`，`pot_pin`，`motor_pin` 等），就能在整个草稿中通过名字调用引脚，而不是引脚的号码。你也可以为一个简单的值声明一个变量（没有连接到任何 I/O 引脚），使用该名字来引用该变量的值。这样，当你稍后想要在草稿中使用变量的值，记忆起来就比较容易。这些变量可以被声明为几种不同的类型，但是最常用的是整型（`int`）。在 **Arduino** 语言中，一个整型变量可以包含值的范围从-32768 到 32767。其他变量类型会在随后的例子中被用到（比如 `float`，`long`，`unsigned int`，`char`，`byte` 等），并且会在使用时进行解释。

下面就是一个变量声明的例子：

```
int my_led = 13;
```

我们将引脚 13 重命名为“`my_led`”，而不是向 **Arduino** 引脚号码（比如 13）发送命令。每当想要使用引脚 13 时，可以使用 `my_led` 代替。当你在整个草稿中对 `my_led` 有很多引用时，这样做很有帮助。如果你决定修改连到 `my_led` 的引脚号码（比如引脚 4），只要在变量声明时修改一次，然后所有 `my_led` 的引用都指向引脚 4，这意味着编码更容易。

2. Setup 函数

这个函数每次都是在 **Arduino** 加电的时候运行，并且只运行一次。在这里，我们通常使用 `pinMode()` 命令来决定所声明变量是作为输入还是输出。

`setup()`函数的例子：

```
void setup() {
```

```
pinMode(my_led, OUTPUT);  
}
```

我们只是使用 `setup()` 函数将变量 `my_led` 声明为输出端（`OUTPUT` 在代码中都需要大写）。虽说现在是这样，但你可以在 `setup()` 函数中做别的事情，比如打开 **Arduino** 串行端口。

3. Loop 函数

程序的主要代码都位于这个函数中，并且会一直重复不断地运行，直到切断 **Arduino** 电源。在这里我们要告诉 **Arduino** 应该在草稿中做什么。每当草稿执行到 `loop` 函数末尾时，就会返回到 `loop` 函数的开始。

在本例中，`loop` 函数通过 `delay(ms)` 函数方便地控制 LED 闪烁的开关。修改第一个 `delay(1000)` 会影响 LED 保持打开多久，而修改第二个 `delay(1000)` 则影响 LED 保持熄灭多久。

下面是一个 `loop()` 函数的例子：

```
void loop() {  
  
    digitalWrite(my_led, HIGH);    // loop 函数的开始，完成以下工作：  
    delay(1000);                  // 打开 LED  
    digitalWrite(my_led, LOW);    // 等待 1 秒钟  
    delay(1000);                  // 关闭 LED  
                                   // 等待 1 秒钟  
                                   // 循环结束，返回到循环的开始  
}
```

如果将这些代码段合并在一起，就会产生一个完整的草稿。你的 **Arduino** 应该有一个 LED 连接到数字引脚 13，因此该草稿中将那个引脚重命名为 `my_led`。LED 将会保持打开 1000ms（1 秒钟），然后保持关闭 1000ms，不断重复执行直到你拔下它为止。我鼓励修改清单 1-1 中 `delay()` 函数的时间值，再上传看看，你会发现什么。

清单 1-1 闪烁样例

```
//代码 1.1——闪烁样例  
// Blink the LED on pin 13  
  
int my_led = 13;                // 声明变量 my_led
```

```

void setup() {
    pinMode(my_led, OUTPUT);    // 使用 pinMode() 命令来设置 OUTPUT 模式
}

void loop() {
    digitalWrite(my_led, HIGH); // 设置 my_led 为 HIGH (打开)
    delay(1000);                // 等待 1 秒钟(1000ms)
    digitalWrite(my_led, LOW);  // 设置 my_led 为 LOW (关闭)
    delay(1000);                // 等待 1 秒钟
}

//返回到循环函数的开始

//代码结束

```

可以将这个代码样例复制到 **Arduino IDE** 屏幕中,然后按下编译按钮(如图 1-18 所示)。将 **Arduino** 插入到 USB 端口之后,就可以按下上传键将代码发送到 **Arduino**。如果是手动输入代码,就不必添加注释,因为注释是会被编译进代码里的。该代码上传之后不需要任何输入,但是可以修改 `delay()` 的时间,然后再上传观察其中的差别。

■ **注意:** 你将会在很多草稿中注意到,所有的注释都通过添加双反斜杠和一些文字表示。双反斜杠之后的任何文字都不会被编译到代码中,参考语句: `// This is a comment`,它不会被当作代码处理。

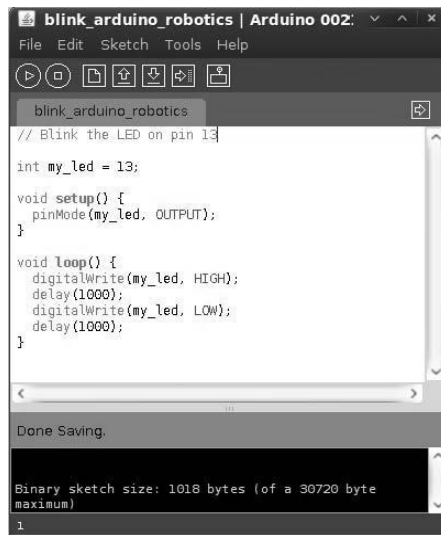


图 1-18 清单 1-1 闪烁样例草稿的 **Arduino IDE** 程序屏幕

1.3.4 信号

Arduino 可以同时进行读/写的信号有好几种类型，但是它们分为两大类：模拟的和数字的。数字信号只能是+5V 或者 0V，然而模拟信号可以是 0V 到+5V 之间的任何线性电压。也可以使用 Arduino 和各种库文件函数读取和写入数字脉冲信号和串行命令。

1. 数字信号

Arduino Uno/Diecimila/Duemilanove 有 14 个数字输入/输出引脚，分别标记为 D0~D13。Arduino 上的每个数字引脚都可以在 setup() 函数中通过使用 pinMode() 命令配置为 INPUT 或 OUTPUT。Arduino 的数字信号只能有两种状态：HIGH 或 LOW。不管数字信号是输入还是输出，都是这两种状态。当引脚是 5V 时被认为 HIGH，当引脚是 0V 或接地时被认为 LOW。

(1) 数字输入

如果你想要确定什么时候按钮被按下（比如碰撞传感器），开关是打开还是关闭，或者想要从传感器上读取脉冲来确定它的隐含值，数字输入会很有用。为了确定输入是 HIGH 还是 LOW，就会用到 digitalRead(pin) 命令。有时候数字输入信号不可能完全总是 5V，因此确定输入引脚为 HIGH 的临界值为 3V 左右，任何低于该临界值的都是 LOW。

用于玩具飞机、舰船和汽车的 R/C 接收器输出“伺服信号”，这是一种被调至 HIGH 状态的电子脉冲，在回退至 LOW 状态之前保持短暂而又固定时间长度的 HIGH 状态。脉冲的持续时间指定了 R/C 发射器控制摇杆的位置。如果你打算使用电压表测量这种类型的信号，将不会看到指针的移动。那是因为脉冲太短而无法在电压表上显示出来，但是 Arduino 上的任何一个数字输入都可以像伺服信号一样使用 pulseIn() 命令读取一个脉冲长度。

我们可以从数字输入中读取信息，不仅仅依据它是 HIGH 还是 LOW，还要依据它保持 HIGH 或者 LOW 多长时间。Arduino 擅长精确测量时间短到 10μs 左右的短电脉冲长度！这意味着相当多的信息可以通过脉冲或者串行命令的方式被编码成数字输入。

(2) 数字输出

数字输出同样很简单，也可以用于完成复杂任务。如果你有一个 Arduino，而且已经看见了 Hello World! 草稿，那就是使内置于电路板中引脚 D13 上的 LED 闪亮——这是最简单的数字输出应用。Arduino 上的每个引脚都能够提供或者发出大约 5V 40mA 的电流。

通常情况下，由 Arduino 引脚提供的电流无法向任何高于 LED 的元件供电，因此就要使用电压转换器或者放大器来增加通过 Arduino 转换 ON 和 OFF 的电流或电压，使得其可

以实现控制电机、灯泡或者继电器。数字引脚也是串行数据传输器的基础，通过单个数字输出可以发送多个命令（如清单 1-2 所示）。

清单 1-2 在同一个草稿中设置数字输入和输出

```
//代码样例——输入和输出
//本代码将 Arduino 引脚 2 设置为数字输入，引脚 13 设置为数字输出
//如果输入是 HIGH，那么输出到 LED 的会是 LOW
int switch_pin = 2;    //这条语句告诉 Arduino，我们想要将数字引脚 2 命名为
“switch_pin”
int switch_value;      // 我们需要一个变量来存储 switch_pin 值，所以定义了
“switch_value”
int my_led = 13;       // 告诉 Arduino 将数字引脚 13 命名为 “my_led”

void setup(){
    pinMode(switch_pin, INPUT); // 让 Arduino 知道使用 switch_pin (引脚 2) 作为输入
    pinMode(my_led, OUTPUT);    // 让 Arduino 知道使用 my_led (引脚 13) 作为输出
}

void loop(){
    switch_value = digitalRead(switch_pin); // 读取 switch_pin，然后保存到
switch_value
    if (switch_value == HIGH){              //如果 switch_value 值是 HIGH...
        digitalWrite(my_led, LOW);         // ...关闭 LED
    }
    else {                                  // 否则...
        digitalWrite(my_led, HIGH);        // ...打开 LED
    }
}
// 代码结束
```

该代码样例使用简单的 if 语句来测试 switch_pin 值。你可以将一根跨接线连接到 Arduino 的引脚 2 (switch_pin) ——跨接线的另一端连接到 GND 或者+5V，然后观察 LED 的变化值。如果输入值是 HIGH，Arduino 将 my_led 引脚设置为 LOW(Off)。如果输入值是 LOW，Arduino 将 my_led 引脚设置为 HIGH(On)。为了使用样例学习更多关于 if/else 语句的知识，请参阅 Arduino 的参考页：<http://arduino.cc/en/Reference/Else>。

(3) 特殊情况（外部中断）

当 Arduino 使用 digitalWrite()命令读取输入引脚时，只能接收到调用命令那一时刻的可

用值。但是，Arduino 完全可以不使用 `digitalRead()` 命令确定引脚的状态何时发生变化。这称之为“中断”。中断是一种输入方法，当某一个引脚的状态发生变化时，它会提醒你而不需要你检查。标准版 Arduino 有两个外部中断，分别是数字引脚 2 和 3。然而 Arduino Mega 有六个外部中断，分别是数字引脚 2、3、21、20、19 和 18。

中断必须在 `setup` 函数中被初始化一次，并且每次运行触发中断时都必须使用特定的函数调用中断服务程序（ISR）（见代码 1.3）。当引脚从 LOW 变为 HIGH（RISING），从 HIGH 变为 LOW（FALLING），或任何时候引脚只在任一方向上发生状态变化，都可以将中断设置为触发。

为了更好地解释这个过程，想象一下，午饭前你正在院子里修整草坪，知道午饭很快就会准备好，并且你不想错过它，但是你也不想每 5 分钟就停下割草机到里面去看午饭好了没有，你要求厨师午饭准备好了就出来告诉你一声。这样一来，你就可以继续修整草坪，而不用担心错过午饭了。

当午饭准备好了（引脚状态发生变化），你就会停下当前工作，并且吃完之后（中断服务程序），返回去接着修整草坪（主程序 `loop`）。

这样做很有用，因为定期检查并不经常改变状态的引脚的状态会减慢主程序 `loop` 中其他的函数。中断只是会暂停主程序 `loop`，花费一些时间去执行中断处理程序，然后马上返回 `loop` 中停止的确切位置。你可以使用一个中断引脚去检测机器人上的碰撞传感器，当检测到它被按下就需要立即停止电机，或者使用一个中断引脚去采集来自 R/C 接收器的脉冲，而无需停止其余的程序。

清单 1-3 要求使用玩具型 R/C 无线电系统。R/C 接收器使用 Arduino+5V 和 GND 供电，而 R/C 信号应当连接到 Arduino 引脚 2。如果你目前还没有 R/C 接收器，可以稍后测试该例子。

清单 1-3 使用中断引脚来采集 R/C 脉冲长度

```
// 代码样例——使用中断引脚来采集 R/C 脉冲长度
// 将来自 R/C 接收器的信号连接到 Arduino 数字引脚 2
// 当使用两个 Arduino 外部中断时，打开 R/C 发送器
// 如果接收到有效信号，你应该会看到引脚 13 上的 LED 打开
// 如果接收到无效信号，你会看见 LED 关闭
```

```
int my_led = 13;

volatile long servo_startPulse;
volatile unsigned int pulse_val;
```



```

int servo_val;

void setup() {
  Serial.begin(9600);
  pinMode(servo_val, INPUT);

  attachInterrupt(0, rc_begin, RISING);    // 初始化上升信号的中断
}

//设置上升沿中断
void rc_begin() {
  servo_startPulse = micros();
  detachInterrupt(0); // 关闭上升中断
  attachInterrupt(0, rc_end, FALLING); // 打开下降中断
}

//设置下降沿中断
void rc_end() {
  pulse_val = micros() - servo_startPulse;
  detachInterrupt(0); //关闭下降中断
  attachInterrupt(0, rc_begin, RISING); //打开上升中断
}

void loop() {
  servo_val = pulse_val; //记录中断服务程序计算的值
  if (servo_val > 600 && servo_val < 2400){
    digitalWrite(my_led, HIGH); // 如果数值在 R/C 范围之内，打开 LED
    Serial.println(servo_val);
  }
  else {
    digitalWrite(my_led, LOW); // 如果数值不在 R/C 范围之内，关闭 LED
  }
}

```

该 **Arduino** 代码搜索插入 **Arduino** 数字引脚 2 的 R/C 接收器的任何有效 R/C 伺服脉冲信号，**Arduino** “0 号外部中断” 位于 **Arduino** 的数字引脚 2。如果检测到一个有效脉冲（一定是长度在 600μs 和 2400μs 之间），数字引脚 13 上的 LED 会打开。如果没有检测到脉冲信号，LED 则会保持关闭。

由于清单 1-3 中使用了一个中断，它只会占用现有的 R/C 脉冲，而不用检查每个回路的脉冲。一些项目需要在每个回路周期执行许多不同任务（读取传感器，控制电机，发送

串行数据等），当中断引脚有情况发生变化，使用中断仅通过中断主程序 `loop` 就能节省宝贵的处理时间。

使用 **Arduino** 两个外部中断时，我遇到唯一的一个问题是它们仅适用于 **Arduino** 数字引脚 2 和 3，与用作 PWM 输出的数字引脚 3 产生冲突。

2. 模拟信号

我们已经知道，数字 I/O 信号必须是 LOW（0V）或者 HIGH（5V）。模拟电压可以是 0V 与 5V 之间的任何值（2V，3.4V，4.6V 等），并且 **Arduino** 有六个能够读取这些电压值的指定输入。这六个 10 比特的模拟输入（由数/模转换器）可以确定一个模拟电压的精确值。

（1）模拟输入

模拟输入是 0~5V 的直流电压电平，将该电压缩放成一个 10 比特的值，或从 0~1023 之间的值。这意味着，如果你将 0V 用于输入，会看到 0 的模拟值；使用 5V，会看见 1023 的模拟值；0V 到 5V 中间的任何值都会正比于输入的模拟值。

为了读取模拟引脚数值，必须使用 `analogRead()` 命令，并且指明你想要读取的模拟引脚（0~5）。**Arduino** 上关于模拟输入的一个有趣现象是，它们不需要在 `Setup` 函数中被声明为变量或者输入。通过使用 `analogRead()` 命令，**Arduino** 会自动知道你试图读取 A0~A5 其中的一个引脚，而不是数字引脚。

电位器（可变电阻器）充当分压器，用于输出弱电流模拟电压，它可以通过 **Arduino** 模拟输入来读取（如图 1-19 所示）。清单 1-4 提供了一个如何读取电位器数值的样例。



图 1-19 这个典型的旋转式电位器有三个端。外面的两个端应该分别连接到接地端和+5V（方向无所谓），而中间端应该连接到 **Arduino** 上的模拟输入引脚

清单 1-4 如何读取一个模拟输入

// 代码样例——模拟输入

```

// 从模拟的引脚 0 读取电位器
// 在串口监控器上显示 10 比特的值 (0~1023)
// 代码上传之后, 打开 Arduino IDE 上的串口监控器, 波特率为 9600b/s

int pot_val;    // 使用变 “pot_val” 来保存电位器的值

void setup(){
    Serial.begin(9600); // 开始时, Arduino 串口通信设置为 9600b/s
}

void loop(){
    pot_value = analogRead(0); // 使用 analogRead 读取模拟引脚 0 的值
    Serial.println(pot_val);    // 使用 Serial.print() 命令来向监控器发送数值
}

// 代码结束

```

将上面代码复制到 IDE 中, 然后上传到你的 **Arduino**。该草稿中使用 **Serial.begin()** 命令控制 **Arduino** 引脚 0 和引脚 1 的串行端口, 你就可以打开 IDE 中的串口监控器, 并查看从电位器的调整而转换成的模拟值。

(2) 模拟输出 (PWM)

从技术上说, 这并不是模拟输出, 而是等效于输出引脚现有模拟电压的数字输出。这个特性被称为脉冲宽度调制, 是一种传输电源和接地端之间电压电平的有效方式。

在电子学中, **PWM** 这个术语使用非常频繁, 因为它在微控制器中是一项很重要的有效功能。该术语代表脉冲宽度调制, 它是你通过电位器看到模拟电压的相应数字电压。**Arduino** 的数字引脚上有六个这样的输出, 例如引脚 3, 引脚 5, 引脚 6, 引脚 9, 引脚 10 和引脚 11。**Arduino** 任何时候都可以很容易地在草稿中使用 **analogWrite()** 命令修改频宽比或者输出。

为了使用 **analogWrite** (**PWM_pin**, **speed**) 命令, 你必须向 **PWM** 引脚 (引脚 3, 引脚 5, 引脚 6, 引脚 9, 引脚 10, 引脚 11) 写入。**PWM** 频宽比的范围从 0 到 255, 所以你不要将任何大于或者小于那个范围的值写入引脚。我通常增加一个过滤器以确保大于 255 或者小于 0 的速度值不会被写入到 **PWM** 引脚, 因为这会导致奇怪的不必要行为发生 (如清单 Listing 1-5 所示)。

清单 1-5 如何控制 PWM 输出

```
// 代码样例——模拟输入- PWM 输出
// 从模拟引脚 0 读取电位器值
// 引脚 3 上的 PWM 输出将正比于电位器输入 (与电压表相符) .

int pot_val;      // 使用变量 “pot_val” 来存放电位器的值
int pwm_pin = 3;  // Arduino PWM 引脚 3 命名为 “pwm_pin”

void setup(){
  pinMode(pwm_pin, OUTPUT);
}

void loop(){

  pot_value = analogRead(0);      // 使用 analogRead 读取模拟引脚 0 的值

  pwm_value = pot_value / 4;      // pot_value 最大值 = 1023/4 = 255

  if (pwm_value > 255){           // 过滤以保证变量 pwm_value 不超过 255
    pwm_value = 255;
  }
  if (pwm_value < 0){             // 过滤以保证变量 pwm_value does 不低于 0
    pwm_value = 0;
  }

  analogWrite(pwm_pin, pwm_value); // 将 pwm_value 值写到 pwm_pin
}
// 代码结束_
```

本代码同清单 1-4 一样，用来读取电位器，但是现在，它也控制一个与 Arduino 数字引脚 3 成正比的 PWM 输出信号。你可以使用电压表测量引脚 3 的输出，根据指针的位置它应该读取从 0~5V 的值。

如果你手边有一个 330Ω 电阻和一个 LED，你可以将电阻与 LED 串联起来（确保 LED 的极性是正确的），再连接到 Arduino 引脚 3 和接地端，使用数字 PWM 信号来查看 LED 从 0% 到 100% 的亮度变化。在该样例中，我们不能在引脚 13 上使用 LED，因为它没有 PWM 功能。

(3) 频宽比

在 1kHz 的 PWM 信号中，每秒钟有 1000 个 On/Off 周期，即每个周期 1ms。其中每个

1ms 周期期间，信号在一段时间里是 HIGH，剩下的一段时间里是 LOW。0%的频宽比表示整个 1ms 中信号都是 LOW，100%的频宽比则表示整个 1ms 中都是 HIGH。

对于每秒钟 1000 个周期，70%的频宽比表示 700 μ s 的 HIGH 和剩下 300 μ s 的 LOW，因此信号的整体效果是可用率 70%。

Arduino 上 PWM 输出的频宽比由 `analogWrite` (pin, duty-cycle) 命令来确定。频宽比的范围为 0~255，并且在编程期间任何时候都可以进行修改，重要的是保持频宽比的值不能超出 255 或者低于 0，因为这会在 PWM 引脚上导致不必要的影响。

大多数的电机转速控制器改变 PWM 信号的频宽比（保持频率固定不变），控制电机功率开关以改变电机的转速。这是控制电机速度的首选方法，因为相对而言，在开关过程中没有浪费热量。

（4）频率

频率以赫兹（Hz）为单位，反映每秒钟的（切换）周期数。切换周期是指在短暂的时间内，输出线从完全 HIGH 变为完全 LOW。PWM 信号通常包含一组频率和变化的频宽比，但是你可以通过为每一组 PWM 引脚添加一行代码来改变 Arduino PWM 频率，范围从 30Hz 到 62kHz（62000Hz）。

在 30Hz 时，输出线每秒钟从 HIGH 到 LOW 被切换 30 次，这将会对电阻负载有明显的影响，例如 LED 使它出现脉冲启动和关闭。使用 30Hz 正好适合于像直流电机这样的电感负载，它要花费比所允许切换周期更多的时间来切断电源，结果导致看似正常的运转。

频率越高，对于负载运转的切换影响就越不明显，但是过高的频率致使开关装置开始产生过多的热量。这是因为随着频率增加，切换周期的长度减少（如表 1-4 所示），如果切换周期太短，输出在返回到 LOW 之前就没有足够的时间完全从 HIGH 切换到 LOW。开关会保持在产生热能的跨导状态（也称之为“直通”），而不是开和关。

通过使用频率划分时间来确定每个频宽比的总长度比较简单。因为频率决定了 1s 的间隔内频宽比的数量，只要简单地用 PWM 频率除以 1s（或 1000ms）就可确定每个切换周期的长度。

为了便于参考，下面是一些常见的时间和速度转换：

- 1000ms = 1s
- 1000μs= 1ms
- 1,000,000μs = 1s
- 1000Hz=1kHz

表 1-4 展示了 **Arduino** PWM 引脚所有可用频率和每个频率上可用的引脚。

表 1-4 PWM 频率与周期时间表

PWM 频率（赫兹）	切换周期时间	Arduino PWM 引脚
30Hz	32ms	9 & 10, 11 & 3
61Hz	16ms	5 & 6
122Hz	8 ms	9 & 10, 11 & 3
244Hz	4 ms	5 & 6, 11 & 3,
488Hz	2 ms	9 & 10, 11 & 3
976Hz (1kHz)	1 ms (1000μs)	5 & 6, 11 & 3,
3,906Hz (4kHz)	256μs	9 & 10, 11 & 3
7,812Hz (8kHz)	128μs	5 & 6
31,250Hz (32kHz)	32μs	9 & 10, 11 & 3
62,500Hz (62kHz)	16μs	5 & 6

想要了解更多关于改变系统定时器来操作不同 PWM 频率的信息，请访问 **Arduino** 演练场的网址：<http://www.arduino.cc/playground/Main/TimerPWMCheatsheet>。

(5) 自制 PWM 的例子

为了模拟频率和频宽比，我们使用人工计时（用于学习和实验目的），结合样例 1-1（闪烁）和 1-4（电位器）来使你改变位于引脚 13（内置 LED）上**伪** PWM 输出频率和频宽比。你所需要做的就是将电位器连接到 **Arduino** 的模拟引脚 0。

使用人工计时和 **Arduino** 引脚 13 的内置 LED，我们可以模拟不同频率和从 0%到 100%不同频宽比的 PWM 信号，如清单 1-6 所示。

清单 1-6 伪 PWM 样例

```
// 代码样例——伪 PWM 样例（自制脉冲宽度调制代码）
// 用变化的频宽比闪烁引脚 13 上的 LED
// 频宽比是通过从模拟引脚 0 读取的电位器值确定的
// 通过将变量“cycle_val”降低到以下 ms 值来改变 PWM 的频率
```

```

// 10 ms = 100 Hz 频率 (快速切换)
// 16 ms = 60 Hz (正常照明频率)
// 33 ms = 30 Hz (中等切换)
// 100 ms = 10 Hz (慢速切换)
// 1000 ms = 1 Hz (极慢速切换) - 不可用, 但是无论如何要试一下

int my_led = 13; // 声明变量 my_led
int pot_val;     // 使用变量 “pot_val” 来存放电位器的值
int adj_val;     // 使用这个变量将那个 pot_val 调整成可变频率值
int cycle_val = 33; // 使用这个变量来手动调整伪 PWM 信号的频率

void setup() {
  pinMode(my_led, OUTPUT); // 使用 pinMode() 命令将 my_led 设置为 OUTPUT
}

void loop() {
  pot_val = analogRead(0); // 从 A0 读取电位器值 (返回的值从 0 到 1023)
  adj_val = map(pot_val, 0, 1023, 0, cycle_val); // 从 0 - cycle_val 映射到 0 -
1023 的模拟输入

  digitalWrite(my_led, HIGH); // 设置 my_led 为 HIGH (打开)
  delay(adj_val);             // 保持打开这个数量的时间
  digitalWrite(my_led, LOW);  // 设置 my_led 为 LOW (关闭)
  delay(cycle_val - adj_val);  // 保持关闭这个数量的时间
}

//代码结束

```

清单 1-6 展示了如何调整以 60Hz 闪烁的 LED 频宽比（每秒钟 16 个切换周期）。这个样例草稿只用于教学目的。因为 `cycle_val` 的值也表示 **LEDs** 衰减范围中有多少步，随着频率增加，你将会失去频宽比分辨率。我选择 60Hz 来示范与你家中灯泡一样的频率。在这种切换速度下，人眼不能轻易观察到脉动，LED 似乎稳定地发出正比于频宽比的光。

如果你想在先前的草稿中手动增加伪 PWM 信号的频率，你可以把 `cycle_val` 变量改得高一些（低频率）。为了将频率从 60Hz 改变至 30Hz，你可以通过修改变量 `cycle_val` 来改变循环时间，从 16ms 调整为 33ms。你也可以控制电位器来达到相同的频宽比，但是结果将明显不那么平滑。随着 PWM 频率降低到 60Hz 以下，你可以在任何频宽比（除了 100%）的 LED 上看到脉动的感觉。

现在，我们已经讨论了一些基本的 **Arduino** 函数，接下来我们讨论构建电路的基础知识。

1.4 构建电路

能够为 **Arduino** 编程并且测试电路是所做工作之一，但是如果你无法找到需要的适合电路，将会发生什么？对你来说，这可能是单独构建电路最简单的方式。首先你需要知道如何查看电路图纸，称之为电路图。电路图显示了每个电子元件（连同名字和数值）的通用符号，以及如何在电路中连接到其他元件的描述。

1.4.1 电路设计

电路设计可以在笔记本或者一页纸上完成，但是复制手工电路耗时并且枯燥。如果要在你的项目中投入少量的时间，你可以使用开源或者免费的程序来设计你的电路，创建原理图和电路图。现在我更喜欢使用计算机来完成我的电路设计，即使不打算根据设计制作印刷电路板，至少我想要给电路做出原理图。

有几个很好的计算机程序可以用来设计电路。对于初学者，我推荐名为 **Fritzing** 的开源程序，它使用给用户电路图视觉感受的一个很好的图解元件库，以及用于每个项目的一个恰当的原理图。在元件库中甚至有 **Arduino 板** 可用于你的原理图中——我使用该程序来生成一些较小的原理图和说明样例。

下载 **Fritzing**: <http://fritzing.org/>

对于更高级用户，**Eagle CAD** 是一个优秀的电路设计程序，可以使用免费版或者付费版，并且拥有大量元件库和专业设计工具。这个程序也会在多个章节中被用到，主要用来打开和打印 PCB 设计文件。

下载 **Eagle Cad**: <http://www.cadsoft.de/>

Eagle CAD 使你能够创建可靠、紧凑和具有专业水准的印刷电路板，并且根据你的确切需求进行定制。你将会在电路准备上花费稍微多一点的时间，但是之后你就可以按照你的意愿很轻松地复制出很多副本——一种使用较简单的点对点连线方式的枯燥任务。不要担心程序中所有这些按钮，如果你将鼠标移动到一个按钮上，它会告诉你它是做什么的。可将 **Eagle** 看做一个非常傻瓜的画图程序。

这个程序是一个印刷电路板（PCB）编辑器，并且有免费版本可供爱好者使用（限制电路板尺寸）。它使你能够打开、编辑和打印原理图与印刷电路板文件，多达两层和 3.2in×4.0in 的丝网印刷面积。不要被限制尺寸所蒙蔽，这足够你去创建本书中用到的任何


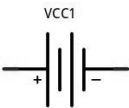

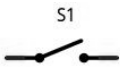

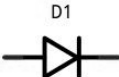

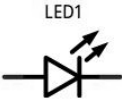


电路和其他很多电路。但是，如果你想要创建自己的印刷电路母版或者类似的东西，你就可能需要购买不限制印刷电路板尺寸的专业版。

我们将在第 6 章中进一步讨论使用设计软件来构建电路图。现在，我们集中于一些不同类型的组件以及它们的功能。尽管有很多现有的元件，但是整本书中的项目只用到了少量元件。让我们来看看一些图片、电气符号和其描述。

1.4.2 原理图

原理图是电路的图形化表示，对于每个电子设备使用一个标准符号，并且使用数字来表示其值。当设备被焊接到电路中时，这对于保证每个设备正确的极性和方向都很有帮助。示意图可以保持不变，即使电路中设备的值或者封装发生变化。表 1-5 列出的是一些原理图中常见的电子元件和符号。

表 1-5 阅读原理图时你可能会碰到的通用元件符号

元 件	原理图符号	描 述
		电源：电池电源的通用符号。电池是一种便携式的电源，用单元格来存储电荷。单元格可以以不同顺序排列来产生特定的电压电平。这里是一个 9V 的电池，通常用于遥控器或者烟雾探测器
		开关：简单的开关用于打开或者关闭电路。这种类型的开关被称为“瞬时按钮开关”，当按钮按下时电路关闭。释放按钮时，电路会被再次打开。这些按钮通常用在低功耗应用的电子电路中
		二极管：二极管的符号，三角形的背面（左侧）被称为“阳极”或者正极端，竖线端（右侧）被称为“阴极”或者负极端。二极管担当单向阀，有多种不同类型的二极管，并且它们也可以用来构建逻辑门、晶体管和几乎所有其他类型的半导体。你经常看到它的话，就会习惯这个符号了
		发光二极管（LED）：LEDs 通常作为指示灯用于电子电路中，因为它们价格便宜、功耗小、持续时间长，并且在尺寸下非常明亮。在我的工作台上有一口袋的各种颜色的 LEDs，因为我不可避免地要在每个项目中至少使用一个 LED
		电阻：电阻是具有特定阻值的线性元件，用于阻碍电流通过电路或者电路中的设备。电阻不是极化的，这意味着电流可在任意方向流入，并且它们流入的方向并不重要

续表

元 件	原理图符号	描 述
		<p>可变电阻器（电位器）：这是你通常所认为的音量按钮。可变电阻器使用滑动片原理来滑动一个横过线性可变电阻平面的导体。</p> <p>通常，电位器拥有三个引脚，最外面的两个连接至+5V 和 GND（以任意顺序），而中间突出部分是电位器可变模拟电压的输出，可连接到 Arduino 的输入引脚，或者其他控制电路</p>
		<p>NPN 型晶体管开关：晶体管是数字开关中最简单的一种。有几种不同的晶体管变种，但是我们只使用双极型晶体管和场效应晶体管。图为 2n3904NPN 型双极型晶体管，通常作为开关用于电子电路，电流电平可高达 200mA 左右。</p> <p>双极型晶体管通常用作高频、低功率电子开关，使用 Arduino 很容易控制。N 型晶体管当作低侧开关用于配合 P 型晶体管使用</p>
		<p>PNP 型晶体管开关：PNP 型晶体管类似于 NPN 型，但是只可以用作高侧开关。图为 2n3906PNP 型晶体管，也能够切换高达 200mA 的负载，并且设计用来配合 2n3904 NPN 型晶体管。</p> <p>通过将 PNP 和 NPN 型晶体管结合在一起，你可以创建一个放大电路或者信号缓冲器（更多信息见第 3 章）</p>
		<p>电容：电容是一种可以存储一定数量电荷的设备，主要用于向电路中其余部分提供电流，或者吸收电压峰值从而保证信号平滑。这种电容额定值为 100μF 和 25V。你应该始终选择额定电压至少高于系统工作电压 10V 的电容。超出电压限制会导致电容爆炸！</p> <p>有些电容是有极性的，并且有一个指定的接地端（通过条纹或较短引脚指示），而另一些是非极性的，可以在任意方向放置</p>
		<p>陶瓷谐振器：这种陶瓷谐振器代替了一个晶体和两个电容器，因为它有两个内置的电容器。你可以轻松地将中间引脚连接到接地端，外侧引脚连接到 Atmega168 芯片的 Xtal1 和 Xtal2（以任意顺序）。该设备为 Arduino 上所有的计时功能提供了一个基准，将它当作一个数字节拍器</p>
		<p>电机：该符号通常表示一个标准的双导线直流电机。连接齿轮箱的直流电机是一种小型的玩具型电机，可以用于机器人项目。通常有两根导线用于操作这种类型的电机，反转这两根导线的极性将会反转电机输出轴的方向</p>
		<p>调压器：LM7805 线性稳压器芯片对于将任何 6~25V 的直流电压输入转变为常规的+5V 输出电源都是非常有帮助的。它只可以提供 1A 左右的电流，所以你不要将它用于为机器人上的直流电机供电，但是对于面包板（即：手提式电子实验线路板）上原型设计或者向自制 Arduino 电路上的 Atmega168 供电，它都可以很好地工作</p>
		<p>接地（GND）：这个符号一般在电路中表示接地信号。每个电路都有一个接地信号，因为它完成电路的回路，电路中所有的接地信号都应该连接在一起，并且回溯到电源的负极端</p>

在图 1-20 中，我们使用表 1-5 中的一些符号创建了一个简单的电路原理图，包含电池（VCC1）、开关（S1）、限流电阻器（R1）和 LED 灯（LED1）。该原理图中，你可以看见用黑线连接的每个元件的符号，表示电气连接。连接后的原理图如图 1-21 所示。

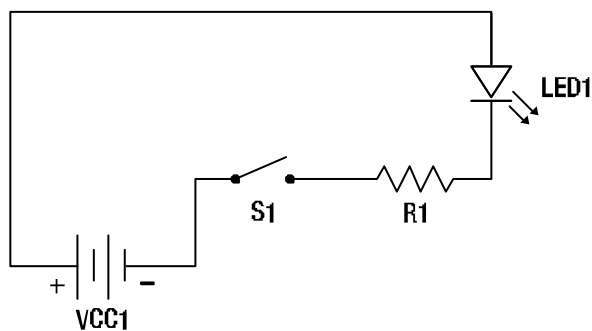


图 1-20 该原理图展示了一个简单的电路图中四个不同元件的电路符号

将如图 1-20 所示的原理图展示为如图 1-21 中所示硬件元件的电气连接。如果所有元件都是根据原理图所示进行连接，那么电路就会按预期工作。这使得用户可以在组装电路时不用考虑它们的物理尺寸或外观。

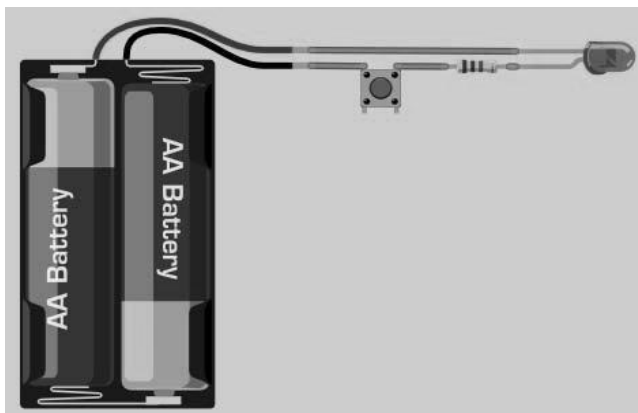


图 1-21 该图片展示了图 1-20 中原理图电路的插图。你可以看到电池组（VCC1）、开关（S1）、限流电阻器（R1）和红色 LED 灯（LED1）。按下开关按钮，电路闭合，LED 灯打开。释放按钮则会关闭 LED 灯

1.4.3 原型

原型以原始形式描述构建设计或者概念的艺术，它不是想要变得完美，而是测试一个

想法的可行性。即使你足够轻而易举地使用数学计算来确定机器人大概的体重和速度，实际上你在创建并且尝试过之前也不会知道它是如何工作的。这就是原型迟早被用到的地方。

你可以使用任何你感觉舒适的材料来创建一个临时的框架（用木头、聚氯乙烯、金属等）。只要它足够坚固，可以临时安装电机和电池，你便会获得有关机器人的实际速度和控制，调整驱动齿轮、电池容量或者相应系统电压的一个好的想法。

原型不仅仅是指安装电机和齿轮，也指设计、创建和测试电子电路。我们还会讨论一些可用的便利工具，使得测试和创建电路更加容易。

1. 面包板

面包板是一种塑料实验板，可以在大多数电子产品供应商店以低于 20 美元的价格买到（如图 1-22 所示）。对电子实验者来说，它是一个有价值的工具，因为你可以增加或者去除测试元件，只要简单地将它们放置在塑料网格上，而不需要焊接。面包板无法负载大量的电流，因此它们不应该用在功耗大的项目中，但是我推荐在创建固定模型之前使用面包板来测试你创建的任何电路。

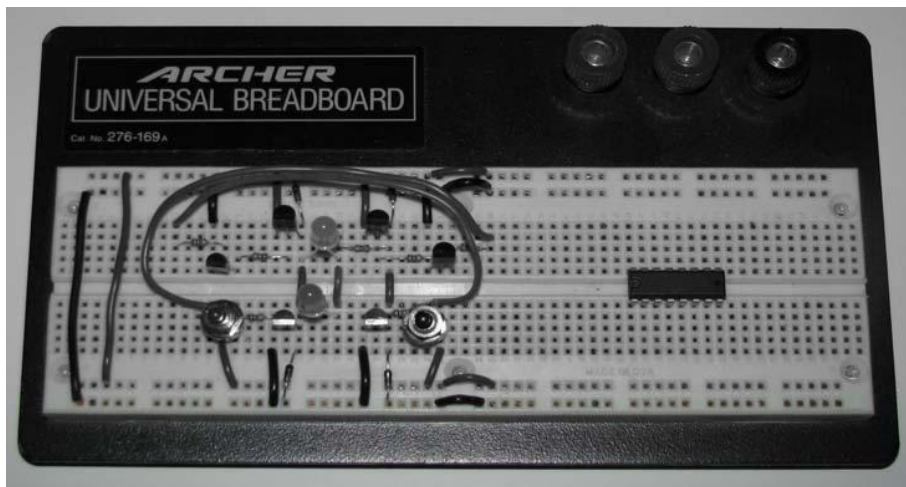


图 1-22 这是个常用的面包板，可在 Sparkfun.com（零件号 PRT-00112）或者任何电子器件供应商店找到

2. 穿孔原型板（穿孔板）

你的电路可以在面包板上正常工作之后，你要准备制作一块硬拷贝来当作原型使用。

有了穿孔板和电烙铁，这会变得相当容易。穿孔板是一种有 0.1 英寸孔间距的预穿孔 PCB（印刷电路板），易于集成大多数的穿孔元件（如图 1-23 所示）。穿孔板上的每个孔都有自己的铜焊垫供你焊接，并且每个铜焊垫都与其他焊垫相互分离（除特殊设计之外）。这种方法需要使用点对点导线接头，如果电路规模很大则会比较令人厌烦，在这种情况下一个更好的解决方案就是腐蚀你自己的印刷电路板（第 6 章中会再次遇到）。但是，对于构建电路的初学者来说，穿孔板会是一个很好的平台，可以测试各种各样的原型，而无须设计和腐蚀正经的印制电路板。

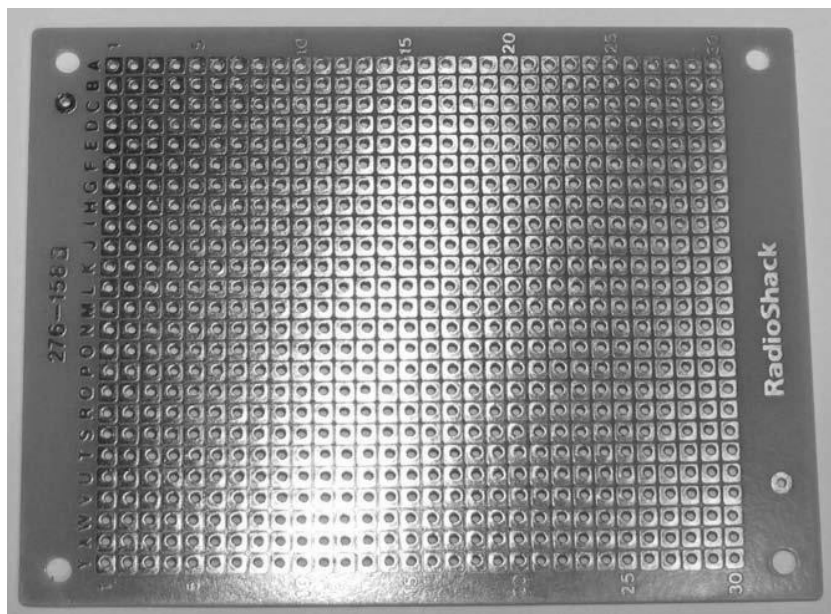


图 1-23 穿孔板的标准件，每个通孔单独有一个铜焊垫。你可以在这类板子上使用元器件、铜导线和电焊料来构建完整的电路。这种类型的电路板通常每个花费不到 5 美元，所以它对于原型很有用

3. 印刷电路板

确定了穿孔板原型上的电路可以按照预期正常工作之后，你可能想要制作 10 个电路板的复制品用来出售或者应用在其他项目中。手工连线 10 个这样的电路板不仅枯燥，而且这种用于点对点焊接项目中的导线可能会损坏或者产生故障，降低可靠性。

构建每一个电路板时，为了避免手工连线的枯燥过程，你可以制作“印刷电路板”，简称为 PCB。印刷电路板可以手工绘制，也可以在计算机上制作，但它就是在一片称为“铜

箔”的镀铜玻璃纤维板上创建一个电路设计，然后再溶解掉该设计周围的铜（如图 1-24 所示）。印制电路板上所有的导线都包含在从电路设计制作的铜线路中。

将铜电路腐蚀到板子上之后，你可以将多个元器件直接焊接到铜上，这称为电路板。Arduino 被印制在一块双面镀铜板上，并且涂有蓝色环氧树脂来防止铜线路短路。使用容易找到的材料，你可以在家里用短短的几个小时来制作你自己的印刷电路板（见第 6 章）。

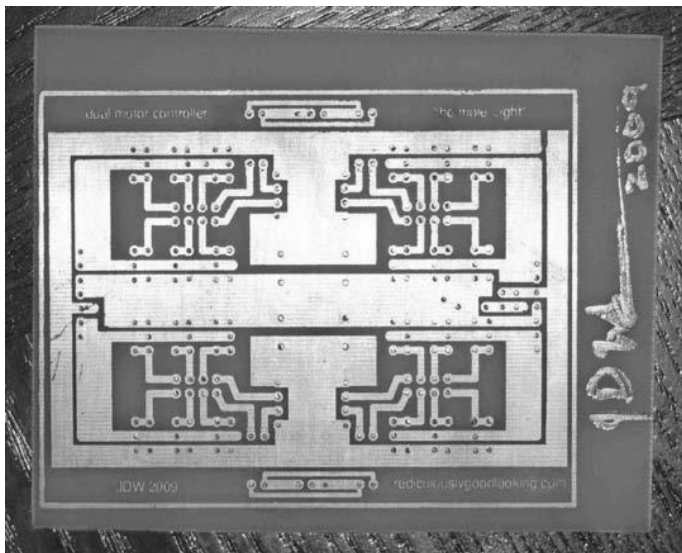


图 1-24 图中所示的印刷电路板是我第一批在计算机上设计的自制电机控制器之一。总共使用了 28 个场效应管开关来驱动两个直流电机，这是第 10 章中用在（200lb）草地机器人上最初的板子在完成电子电路之前，你需要将每个元件焊接到 PCB 上。

4. 焊接

电气焊接一般指的是使用烙铁和电焊锡将电子元件固定到印刷电路板上，它提供到 PCB 的安全连接。做法是让元件引脚和 PCB 铜焊垫足够热，这样，当焊锡接触到它们时就会熔化。同样应注意的是，你不应该用电烙铁加热焊锡丝，因为如果它们也是热的，它只会融合到元件引脚或者铜焊垫上。

可用的焊锡有许多不同类型，但是对于电气连接你应该使用松香芯型焊锡，如图 1-25 所示。通过使用直径更细的焊锡丝，它不需要非常高的热度就可融合到铜焊垫和元件引脚上。

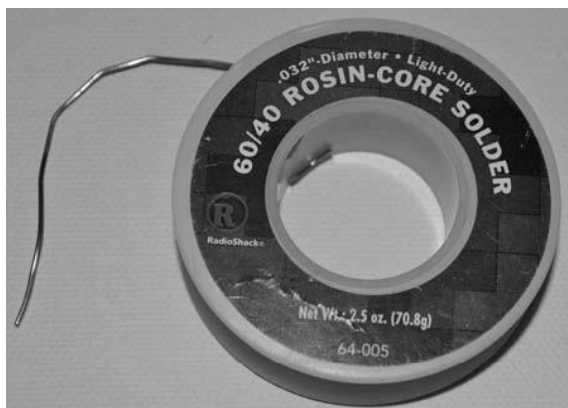


图 1-25 这是一卷用于电路构建的松香芯型焊锡

在试图焊接之前，最好让电烙铁完全加热。当电烙铁不够热，焊锡就熔化不了，焊接会变得令人沮丧！你只想在铜焊垫上使用足够的焊锡来完全填补元件引脚周围的任何缝隙，但是又不想使用过多的焊锡，因为它冒出来，就可能会触及另一个元件引脚或者铜线路。

你可以在很多五金店或者无线器材店以不到 10 美元的价格买到一个电烙铁。大多数的项目都要用电烙铁，它们需要一段时间加热（10 分钟左右），并且难以焊接成为结实的焊点，因为它们通常有一个大的烙铁头。

可调温电烙铁有多个加热部件，可以在 1 分钟内加热，通常有更小的烙铁头用于焊接小的项目或者紧凑的空间（如图 1-26 所示）。如果你买得起，我强烈推荐你买一个，它们的价格通常从 50~150 美元不等。



图 1-26 我用了廉价电烙铁多年，之后我妻子决定给我买一个“好”的电烙铁。Hakko 936 可能不是最好的，但是它比我之前浪费我时间的 7~10 美元的电烙铁要好很多。它可以在短短的 1 分钟内加热，而且比普通的电烙铁更热，让焊接成为一件轻而易举的事情

焊接可能需要花费一些时间来熟悉，所以我建议你在试图创建你的 PCB 之前买一些穿孔板然后用它去练习。你可以从不同的供应商那里购买到电子套件，它配备了所有需要的零部件、PCB 和说明，只要你要有一个电烙铁和一到两个小时的装配时间。在我学习焊接时，我买了几个套件，这些套件既提供一个有趣的项目又提供动手实践学习的宝贵经验。

5. 焊接捷径

当焊接穿孔板时，有时我们电路板的镀铜一侧有一条清晰的线路，将一个电气引线连接到另一个。为了使焊接更简单，以保证电路不受杂乱的导线影响，我们可以使用一些焊接捷径来简化连接（如图 1-31 所示）。

- 【选项 1—合并焊锡】你会发现，如果你加热相邻（**但分开**）的铜焊**垫**，并且使用焊锡，焊锡在回避两个铜焊**垫**之间空隙的时候，往往会向两个铜焊**垫**流动。这是因为没有任何镀铜的话，焊锡无法粘住玻璃纤维 PCB。如果你向两个焊**垫**添加了“过多”的焊锡，你会注意到，熔化的焊锡会试图跳过空隙到达另一个焊**垫**上另一组融化的焊锡。如果认真去做的话，你可以在两个焊**垫**之间**让**焊锡凝固，创建一个单一的焊接连接。这是在两个或三个相邻铜焊**垫**之间创建跳线的有用方法。但是对于大功率的连接，这是不可取的，因为焊锡不能够传输大量的电流。
- 【选项 2—**线迹**】你也可以将一段实心裸铜线（16~20**awg**），直接放置到你想要连接的铜焊**垫**上（如图 1-27 中的 A、B 和 D 所示）。如果该连接要跨越多个焊**垫**，最好是导线接触的每个焊**垫**都用少量的焊锡，以确保电路完成之后它不会移动。你还可以弯曲其他元件周围的导线来制作一条**板成弯状**或者倾斜的线。这种方法产生的结果类似于自制印制电路板的线迹。因为每根导线都直接从一个引线连接到另一个，所以 PCB 下面的其他元件不会有交叉导线。虽然可用于要传送的电流有一个相应的线表，但这种方法适合于大电流应用。

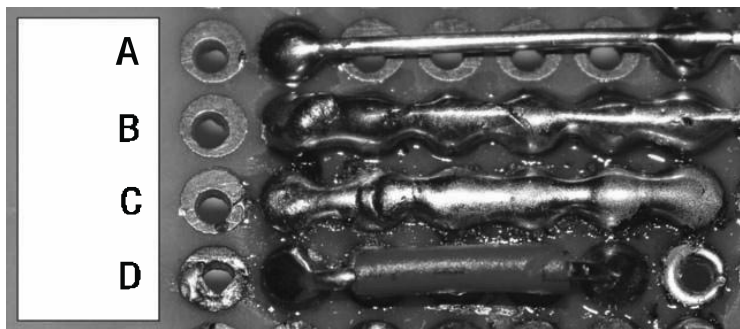


图 1-27 如何使用铜导线创建**线迹**

线迹 A 是一段没有绝缘的裸线，但是只焊接了裸线的两端。线迹 B 是一段裸线，但是在每个铜垫上都进行了焊接，使它远比线迹 A 安全。线迹 C 甚至没有导线，它只是横穿六个焊垫焊接起来。线迹 D 是一段有自己绝缘的导线，但是只是焊接了两端。线迹 C 很难完成横穿两个或者三个以上的焊垫，并且应用于大功率应用是不可取的。

1.5 搭建机器人

实际上动手搭建机器人是整个过程中我最喜欢的部分。在这里，你可以通过设计和构造任何你能想象的东西来表达自己的创意。这个过程通常开始于几块金属或木头和一些螺母、螺栓、螺丝、胶水、绝缘胶布和其他任何你可以找到的来让你的机器人动起来的材料。

首先你需要决定想要你的机器人做什么，并且设定好目标（即使他只是简单地四处游走）。你可以搭建一个自主机器人，使用传感器作引导或者使用你的输入作为控制的无线控制的机器人。如果你之前从未搭建过机器人，你可能应该从小处着手。本书中的几个章节使用玩具型伺服舵机来驱动机器人，这很容易直接连接到 **Arduino**，而不需要电机控制器。只要把一个项目的几个零件结合在一起，如果需要的话，组装和调整会更容易、更快。

不管你做什么，别试图第一次就做得完美。最好是有一个合适的想法和原型，而不是只有一大堆非常好的想法。一个好的想法可能在你的脑海中回响，但在你试过之前，你无法知道它是否会真正工作。在我找到一个可以正常工作并且喜欢的机器人之前，本书中的一些机器人已经过多次改进。如果你的机器人在第一次尝试时无法按照预期正常工作，那么做好笔记然后再试一次，这就是如何搭建好的机器人的经验。

建造、测试、修改、损坏、重新搭建，这就是机器人设计周期。

1.5.1 硬件

拥有合适的工具可以让搭建过程容易得多，但不是每个人都有工具齐全的工作台。因为好的工具比较昂贵，当你发现你需要它们时，你可能会想买一件。这样，你就不会有你从来不用的工具。

1. 基本的搭建工具

虽然通常很多的电动工具都是可选的，但下面的是基本工具，我建议在你开始制作机

机器人之前最好具备。你需要用时就可以用上了，这些工具都应该在你的清单中（如图 1-28 所示）：

- 锤子；
- 月牙扳手；
- 老虎钳（标准的和**针鼻的**）；
- 剥线钳和压线钳；
- 万能**钳**；
- 螺丝刀——十字头的和一字头的；
- 卷尺。



图 1-28 **基本的工具箱**：多功能 6 合 1 螺丝刀（上排中间）、25 **英尺**的卷尺（上排右侧），
（从左到右）锤子、活动扳手、边裁**钳**、尖嘴钳、剥线钳、压线钳、剪线钳和万能**钳**

有了基本的工具箱之后，你会开始需要更先进的工具，因为你需要它们（或者因为你能买得起）。你也可能会需要下面的工具来搭建本书中的每一个项目，只要可以用到它们，你就不一定要拥有每一个工具。

- **【计算机】**虽然你的工作台上通常不会有，但是你需要一台计算机来执行 **Arduino** 程序，上传代码。你没必要用最新、最好的计算机来运行 **Arduino IDE**，几乎任何有 **USB** 接口的计算机都可以做到。**Arduino** 软件和 Eagle Cad 都能够运行在 Windows、Linux 或者 Mac 上。

- **【电压表（万用表）】**这不一定要贵的，通常你能找到的最便宜的万用表都可以用来测量交直流电压、电阻和大约 250mA 的直流电流。我更喜欢模拟测量表，那样我就可以看见指针的每一个移动，以及它如何对我测试信号作出反应，数字测量表直接跳至读数处，可以轻松读取到精确值，但是要读变化的电压值较为困难。
- **【电钻】**如果你现在还没有，需要买一个。你几乎可以在任何一个五金店以不到 20 美元的价格买到一个电钻。如果你想要做一些好的东西，花 75 美元买一个 18V 的无线电钻工具包。如果你计划腐蚀你自己的印制电路板，拥有一台钻床也是很有用的。钻床通常可以用 60 美元买到。如果你打算使用金属，你还需要一些钻头。
- **【锯子】**你可能会需要多种锯子，但是它们的型号取决于你想要完成的工作量。你可以用来切割金属的最便宜的锯子就是钢锯，但是切割过厚的金属片需要一些耐心。往复锯（有时候被称为马刀锯或者全锯）对于切割任何东西都是一个不错的选择，从金属到木头，再到 PVC。如果你已经有一个竖锯，你会派上用场的。对于机器人项目来说，尽管它们功能稍微少一些，但是有一个竖锯的话工作会十分方便。
- **【电烙铁】**如果你计划制作本书中的全部电路，那么你就需要一个。记住，焊接时要用一块湿海绵或一个钢丝刷保持烙铁头干净。你可以用 7 美元从 Radio-Shack 买一个，但是如果你要经常焊接，我强烈推荐用可调节温度的可控烙铁。它们加热更快、更热，但是要花费从 50 到 150 美元不等。
- **【电焊机】**这个不需要，但是在更大的项目中，它会很有用。标准 110V 送丝式工作更好。时刻记住，要戴焊接面罩以避免伤害眼睛，并且永远不要直接看着焊接电弧。

1.5.2 材料

本书中，我们将会使用几种不同材料，包括木头、金属、塑料和玻璃纤维。在切割这些材料中的任何一种时，戴着防护目镜和手套始终是一个好主意。虽然我个人更喜欢金属，但是可以使用任何你感觉顺手的材料。

- **【木头】**木头是最简单、最便宜的材料，也牢固到足以支撑一个大机器人的重量。诱人的的是你可以从木场买一些 2x4s 用于制作机器人框架，它们有可能会变形和断裂，对于我们投入了大量时间的项目而言，这种方法便缺少吸引力。但是，它对原型设计会很有用。
- **【塑料】**在需要能看清楚的应用项目和小机器人底座上，我喜欢使用丙烯酸树脂玻

璃来取代玻璃，因为它很容易钻孔和挖掘，并且可以使用**竖**锯切割。PVC（管）会对需要轻量级、高强度的项目很有用。树脂玻璃、PVC 和其他大多数塑料都可以使用热风枪进行加工或定型。

- **【金属】**制作机器人框架要热处理金属是件困难的事情。但它非常坚固、耐用，并且通过焊接或者使用螺钉和螺母相连接。切割也是一件比较困难的事情，需要一把钢锯（和一些费力的工作）或者带有细齿金属刀片的往复**锯**。但一旦制作成，一个金属框架将会用很多年，并且不会弯曲或变形。大多数的五金店都销售 48 英寸中长款的各种角钢和金属棒，根据尺寸和厚度价格从 5~25 美元不等。
- **【玻璃纤维】**玻璃纤维是一种用来制作特定形状的很好的材料，而且这些形状几乎是不可能用金属和木头来实现的。一旦成型，它也是非常坚固和坚硬的，也防水。该过程包含铺设玻璃纤维布，然后在布上涂两层树脂。这大概只需要一个小时就可以变坚固，但是它会产生一些浓重的烟雾。1 加仑的玻璃纤维树脂 25 美元左右，大多数的五金店都能买到（能保存很长一段时间），特定玻璃纤维布（有时候被称为“垫子”）的价格为 8 平方英尺 5 美元左右。

1.5.3 工作区域

理想情况下，我们有无限的工作空间，但是通常你的空间取决于你的生活安排。当我居住在公寓里时，我将项目的一部分随处放置，并且将我们廊后面用作金属工作室，这令我的邻居们非常沮丧。既然我有一栋房子，那我尽量将项目的所有工作都限制在车库中完成，会在室外去做大部分的切割、磨削和产生噪声的工作，那里还有良好的通风。

当选择在哪里构建你的项目时，有几件事情你应该考虑。这些注意事项经常会被人们忽视，但是对你和周围人的安全都很重要。

- **【测试空间】**凡事不会总是按照计划进行，所以每当你正在测试一个会对别人造成身体威胁的动起来的机器人，有一个大的房间是个很好的想法。这本书中的几个机器人都很大，如果失去控制会严重伤害人和宠物。不要在室内或者人的周围测试大型机器人。
- **【通风设备】**吸入污染物会对你的肺部 and 大脑造成伤害。吸入锯屑可能只是简单地感觉不舒服，但是吸了酸性腐蚀剂的烟雾或焊接烟雾证明会对健康造成危害。总之应该在一个通风良好的地方或室外工作。如果你正在焊接、腐蚀、点焊，或者加工

玻璃纤维，要使用呼吸器面罩来保护你的肺部，再使用一个风扇将危险的烟雾从你的工作区中排出去，这是一个很好的主意。

- **【安全性】**时刻注意你的机器人。不用的时候切断机器人的电源线都是一个好的习惯，这样会防止意外的启动和可能的危险。不要低估你机器人对周围估计不到的物品的肆虐和毁坏的能力（即使这是他的目的）。
- **【小孩】**如果你的小孩可能会进入工作室，请务必确保你的电烙铁在小孩接触的范围之外，并且拔掉电源，使任何刀片或尖锐的东西都在够不着的地方，使小元件远离那些会将它误认为是新型糖果的人，并且确保任何会伤害小孩的机器人要放在远离关闭杀人开关（断开电源）的地方。这本书中的几个项目要使用电机来携带人。这些电机都很强大，如果它失去控制，足够对人造成身体伤害。出于这个原因，为了他们的健康，我建议你将动起来的机器人与人和宠物的距离保持在 20 英尺之外。

1.6 总结

回顾一下本章，我们首先讨论了电学的基本知识，包括电流的比喻、电的性质、电路和连接类型。讨论完电学之后，我们转移到电子学，并讨论了半导体、技术手册、集成电路和集成电路的封装。

接下来，我们简单介绍了 Arduino 微控制器，包括 Arduino IDE、两种主要的 Arduino 变体（标准版和 Mega 版）、一个草稿的元件，最后是在 Arduino 上不同类型可用的常见信号。

本章还简明扼要地讨论了电子电路设计，以及一些用于各种电子元件的不同类型的原理图符号。我们介绍了本书项目中你需要的基本工具和要用到的材料。

第 2 章，我们将讨论如何使用各种各样不同的设备连接到 Arduino 引脚。

第2章



基于 **Arduino** 的机器人

由于具备了一些关于电学、**Arduino** 和普通机器人制作方法的基本知识，所以我们直接进入完成本书项目所需的一些专用接口任务。在第 1 章中，代码样例使用了可以直接被连接到 **Arduino** 上的低功率元件（LED，电位器，R/C 接收器，按钮开关等）。本章将重点介绍如何将 **Arduino** 连接到机械、电子和光开关，以及一些不同的输入控制方法，最后再讨论一下传感器。

首先讨论将继电器、晶体管和电机控制器连接到 **Arduino** 上的基础知识。接下来，讨论控制 **Arduino** 的各种方法，重点介绍流行的无线控制方法。最后，会告诉你我个人对机器人可用的几种不同类型传感器的看点。

本章中没有代码样例，但是所提供的信息对于理解本书使用的接口方法、控制类型和传感器都非常有帮助。首先介绍一些能够让 **Arduino** 控制大功率设备的开关元件。

2.1 **Arduino** 接口

因为 **Arduino** 任何一个输出引脚只能够提供 40mA 左右的电流，这大大限制了其自身能够赋予动力的范围。一个典型的 5mm 红色 LED 大概需要 30mA 的电流，所以 **Arduino** 让它达到 100% 亮度没有什么问题——但是对于电流更大的设备，就会比较吃力。为了使用 **Arduino** 来控制大功率设备，就需要使用“放大器”。放大器也被称为“信号缓冲器”，简单地复制小功率输入信号，产生更大的输出功率来驱动负载。

基本的放大器有一个输入和一个输出——输入的是小功率信号（比如 **Arduino**），用于驱动更大的向负载供电的输出信号。放大器最好能够像 **Arduino** 切换小功率信号一样，尽

可能快地、高效地转换大功率信号。在现实中，放大器并不是完全高效的，一些热量会在切换过程中被消耗，往往需要在切换设备上使用散热器，也可能是用一个风扇来去除热量（比如计算机中的 CPU）。

根据所使用输出信号的类型，与 Arduino 连接的放大器电路具有不同类型。对于慢速切换信号，使用 `digitalWrite()` 命令，可以将 Arduino 连接至大功率继电器。对于快速切换 PWM 信号，使用 `analogWrite()` 命令，必须使用固态开关，它允许 0~100% 全数字输出控制。也可以购买预装配的电子速度控制器，并且使用 Arduino 来提供输入控制信号。

首先，介绍一下称为继电器的电子启动开关，它能够通过很大的电流，并且可使用 Arduino 进行控制。

2.1.1 继电器

继电器是一种电子开关，它使用电磁螺线管来控制机械电源触点的位置。螺线管与电机类似，因为它使用磁场来产生螺线管圆柱体的物理移动，但不是像电机输出轴一样旋转，螺线管圆柱体以直线运动方式来回移动。大多数的继电器都装在塑料或者金属外壳中，以保证移动部件不会受到外界干扰并防止灰尘（如图 2-1 所示）。

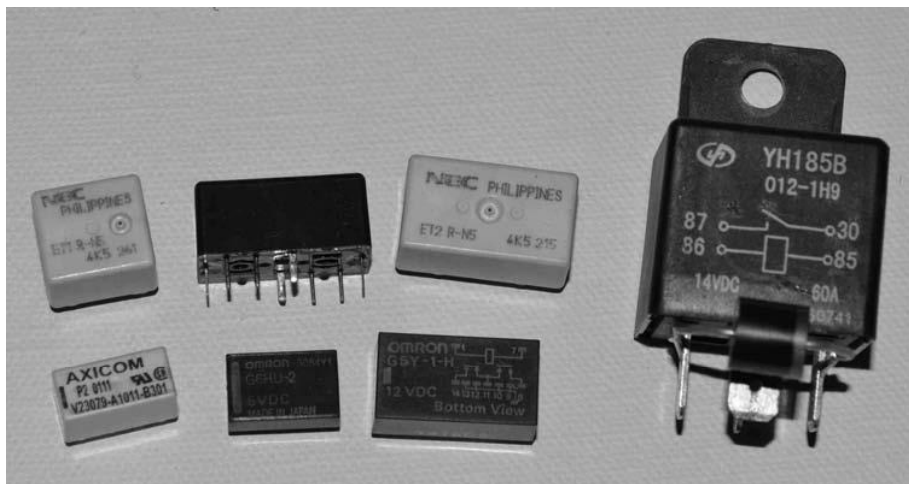


图 2-1 在这里，你可以看到各种继电器，尺寸从大到小。最下面一排三个小一些的继电器被称为“信号”继电器，意思是它们触点的额定电流不超过 2A。上面一排的三个继电器被称为“功率”继电器，范围从 5A 到 25A 的触点负载。最后，最右边那个庞大的继电器是汽车功率继电器，额定值为 60A

继电器包含两部分：螺线管和触点，并且相互之间电隔离。从本质上来说，这可以被当成一个电路中分开的两个部分（但有关联），因为每部分都有它自己的额定值。继电器内部的螺线管有一个带有磁性柱塞的电线圈，提供所需的活动来使触点开关闭合和断开。继电器线圈应当有线圈电阻和工作电压，这样就可以计算出它使用时会消耗多少电流。继电器中的触点是功率信号切换的地方。触点开关也拥有额定电压和额定电流，告诉你在触点失效之前预计继电器功率能导通多少功率。

1. 继电器种类

根据你的应用场景，继电器可用于多种不同的操作。因此了解每种继电器如何工作，对于确保你选择适合当前工作的继电器很有帮助。

- 常开接点 (NO)：这只是表示，当继电器线圈接通，继电器的两个电源触点被连接，线圈关闭时被断开。
- 常闭接点 (NC)：这与常开接点正好相反。当继电器关闭时电源触点被打开，继电器接通时被断开。
- 封锁：这意味着继电器中的触点开关不是弹簧承载的，并且停留在被放置到的任何位置，直到线圈极性发生改变才将触点开关返回到其起始位置。这相当于一个标准的家用照明开关，保持打开直到你关掉。
- 非封锁：这是“正常”类型的继电器，主要用于失效保护开关。此种继电器触点开关是弹簧负载性的，除非向线圈提供电源才返回到预设置的位置。这相当于瞬时按钮开关，通常保持打开，只有当你按下按钮，它才弹回至关闭位置。

2. 继电器配置

除了有不同的操作类型，根据用途可以将继电器的触点设置成各种配置。我们简要介绍如下 4 种常见的继电器类型。这些继电器中的每一个只有电磁线圈，但是电源触点的数量不同。如上所述，任何一个继电器配置既可以是常开接点或者常闭接点，也可以是封锁或非封锁的。

- 单刀单掷 (SPST)：这种类型的继电器使用一个线圈来控制有两个触点的开关——该继电器上总共有四个触点（如图 2-2 所示）。
- 单刀双掷 (SPDT)：这种类型的继电器使用一个线圈来操作有三个触点的开关（如

图 2-3 所示)。中间的触点用于负载，上面的触点用于电压 1，下面的触点用于电压 2（或者接地）。该继电器总共有五个触点，用于在两个不同的输入源（**掷点 1-1** 和 **掷点 1-2**）之间切换触点（电极 1）——也被称为三路开关。

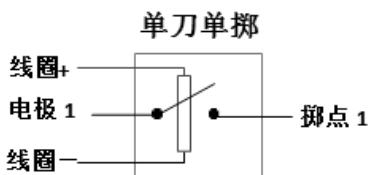


图 2-2 该单刀单掷继电器有一个电极，
一个触点（简单的开关）

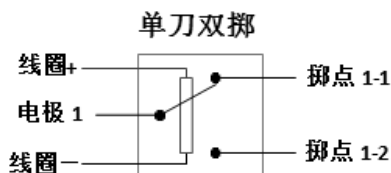


图 2-3 该单刀双掷继电器有一个电极，
两个触点（三路开关）

- 双刀单掷（DPST）：这种类型的继电器同一时刻使用一个线圈来操作两个独立的单刀单掷开关（如图 2-4 所示）。该继电器总共有六个触点，用于在同一时刻进行两个负载的切换——被切换的两个负载可以是有联系的（比如一组电机连接线），也可以是单独的（比如双电压电源开关）。
- 双刀双掷（DPDT）：这种类型的继电器同一时刻使用一个线圈来操作两个独立的单刀双掷开关（如图 2-5 所示）。该继电器总共有八个触点，可以被配置成 H 桥式电路，这将会在第 3 章介绍（用于控制负载的方向）。

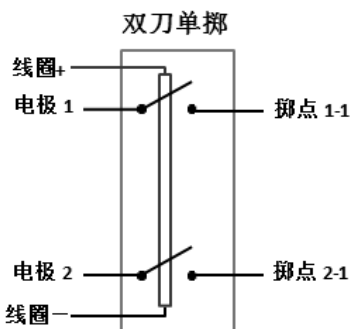


图 2-4 该双刀单掷继电器有两个电极，并且
每个电极对应一个触点（**双开关**）

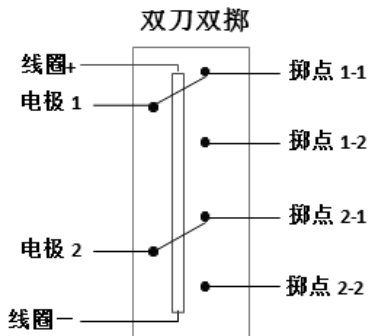


图 2-5 该双刀双掷继电器有两个电极，并且
每个电极有两个触点（**双三路开关**）

3. 用途

继电器比使用**厚铜**触点更具优势，所以它们可以轻而易举地使用相对较少的输入电流来切换大电流。因为螺线圈移动触点需要花费一些时间，故 PWM 不能使用继电器。PWM

信号作为模拟电压出现在继电器中，或者电压高到足以打开继电器线圈，或者只是保持继电器关闭——但是在通常情况下，将 PWM 信号用于继电器并不是一个好的想法。

但是，可以用继电器来切换使用 **Arduino** 的大功率负载——包括直流电和交流电灯泡、电机、加热器以及几乎所有的用电设备。在机器人中，继电器相当有用，因为它既可以切换大功率负载，也可以用于电子控制（以及远程），它的使用使很多想法成为现实。你可以将功率继电器作为应急电源断开与大型远程控制机器人的连接，或者作为电动马达和灯泡的遥控电源开关。

使用两个 SPDT（三路）继电器开关，我们能够控制直流电机的方向。在图 2-6 中，你可以看见，如果两个继电器线圈（控制 1 和控制 2）被激活，电机上端将会被连接到电源的正极、下端被连接到电源的负极，引起电机沿顺时针方向旋转。如果继电器线圈两端的电源被移走，电机上端将会被连接到电源的负极、下端被连接到电源的正极，则将引起电机沿逆时针方向旋转。

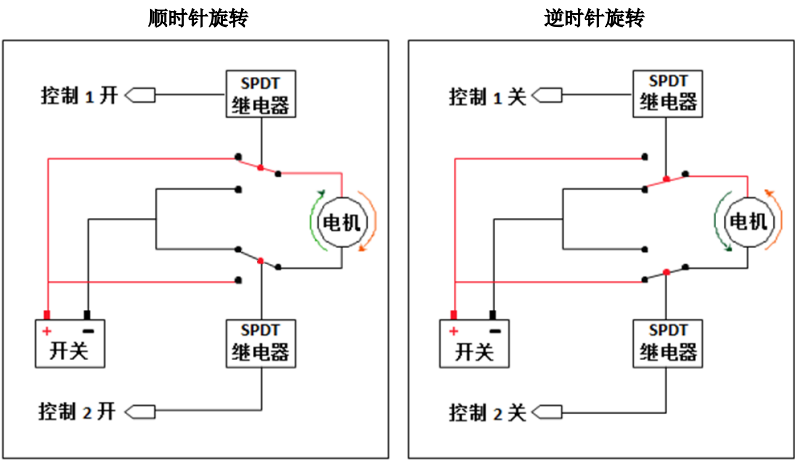


图 2-6 这些图片展示了如何使用两个单刀双掷继电器开关（或者一个双刀双掷继电器开关）控制直流电机

在使用继电器之前，要计算需要多少功率来驱动继电器线圈。如果继电器线圈消耗的电流超过 **Arduino** 可以提供的 40mA，将需要一个接口开关用 **Arduino** 来启动继电器线圈。

4. 计算电流消耗

为了确定继电器消耗的电流，首先必须通过查阅继电器数据手册来确定线圈电阻。如果无法查找到此类信息，可以使用万用表测量其电阻值。知道了线圈电阻和继电器的额

定电压，就能够使用欧姆定律来计算线圈的电流消耗。

在图 2-7 中，可以看到欧姆龙 G5-CA 系列继电器数据手册的例子。正如所见，继电器可提供三种不同的线圈电压（5V、12V，或者 24V）。下面列出的额定电流是每个模型的线圈电阻。5V 版本的继电器线圈的额定电流有 40mA，它电流很低，足以通过 [Arduino](#) 供电，而不使用接口电路。

Specifications

■ Coil Ratings

Item	Standard, high-capacity, or quick-connect terminals		
	5 VDC	12 VDC	24 VDC
Rated current	40 mA	16.7 mA	8.3 mA
Coil resistance	125 Ω	720 Ω	2,880 Ω
Must-operate voltage	75% of rated voltage (max.)		
Must-release voltage	10% of rated voltage (min.)		
Max. voltage	150% (standard)/130% (high-capacity, quick-connect terminals) of rated voltage (at 23°C)		
Power consumption	Approx. 200 mW		

Note: 1. The rated current and coil resistance are measured at a coil temperature of 23°C.
 2. The operating characteristics are measured at a coil temperature of 23°C.
 3. The "maximum voltage" is the maximum voltage that can be applied to the relay.

■ Contact Ratings

Item	Standard		
	Resistive load	Inductive load ($\cos\phi = 0.4$, $L/R = 7$ ms)	
Contact form	Single		
Contact material	Silver alloy		
Rated load	10 A at 250 VAC; 10 A at 30 VDC	3 A at 250 VAC; 3 A at 30 VDC	
Rated carry current	10 A		
Max. switching voltage	250 VAC, 125 VDC		
Max. switching current	10 A		
Max. switching power (reference value)	2,500 VA, 300 W	750 VA, 90 W	

图 2-7 这是一个继电器部分数据手册的样例，可以同时看到线圈额定值和触点额定值

尽管这个特殊的数据手册显示了继电器线圈的额定电流，但是一些继电器只列出了工作电压。这种情况下，我们必须使用万用表手动测量继电器线圈的电阻，然后使用欧姆定律来计算电流消耗。

根据图 2-7 中的数据手册，使用欧姆定律来验证 5V 继电器的电流消耗，其线圈电阻为 125 Ω 。

$$V = I \times R$$

$$I = V / R$$

$$I = 5\text{V} / 125\ \Omega$$

$$I = 0.040\ \text{A} \ (40\text{mA}) \text{ —— 数据手册结果正确!}$$

5. 反电动势注意事项

回顾第 1 章，继电器线圈（螺线管）是一种电感型负载，每当螺线管关闭时，都会产生逆电动势的波动。在没有标准电流整流二极管保护的情况下，这种逆电动势可能会严重损坏电子开关组件，例如图 2-8 中使用的 1n4004 型二极管。该二极管位于负载的两端（该例子中，继电器线圈）以避免逆电动势损坏 Arduino 的输出引脚。

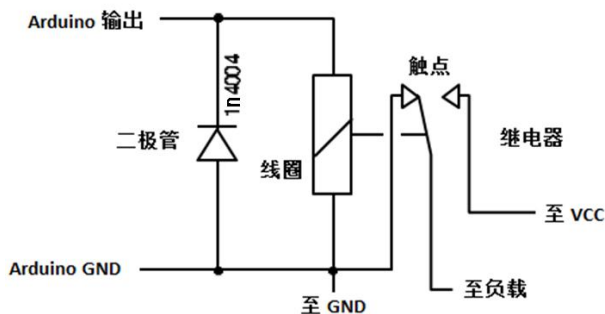


图 2-8 该原理图展示了继电器线圈周围二极管的应用，以保护 Arduino 的输出引脚或者其他开关设备免受继电器线圈产生的逆电动势损坏

尽管图 2-7 中列举的继电器可以被 Arduino 上有效的电流直接驱动，但是大多数功率继电器的开启电流需要略高于 40mA。该例子中，需要一个信号接口开关来控制 Arduino 向继电器线圈提供电源。为了实现这一切，首先需要讨论固态（电子）开关。

6. 固态开关

固态开关是一种使用没有移动部件的掺杂硅芯片来切换电子负载的设备。晶体管、场效应管、光敏晶体管和固态继电器都是固态开关的例子。由于固态电子设备没有移动部件，所以它们的切换速度远远高于机械开关。你可以查阅所使用元件的制造商数据手册，但通常情况下，在这些开关中可使用 PWM 信号向负载设备提供可变输出。

电路中，有两个可以放置开关的地方，用来控制负载电源。如果开关位于负载和电源正极之间，称之为高侧开关。如果开关位于负载和电源负极之间，则被称为低侧开关，如

图 2-9 所示。

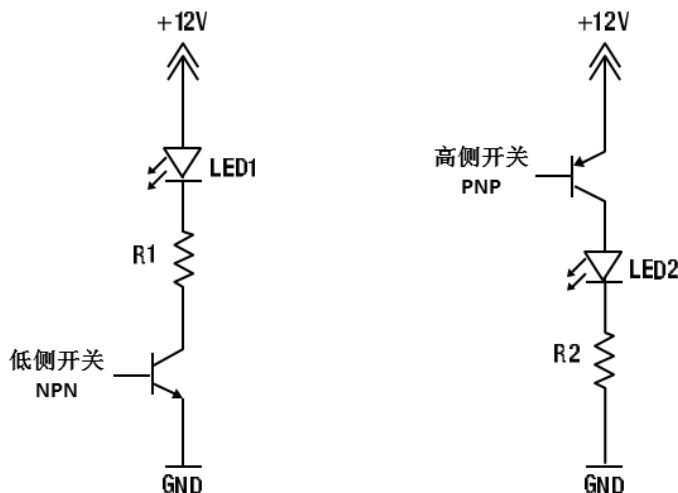


图 2-9 该图中，可以看到高侧开关和低侧开关的不同之处

2.1.2 晶体管

晶体管是一种电子开关，使用常见参考电压将小输入信号切换为大输出信号。晶体管开关不同于正常开关（例如继电器），因为它们不能随便放置在电路的任何地方。低侧开关必须使用负掺杂的晶体管，而高侧开关必须使用正掺杂的晶体管。

我们使用的常见类型的晶体管有三种：双极型晶体管（BJT）、金属氧化物半导体场效应晶体管（MOSFET）和光敏晶体管。所有这些晶体管都是晶体管（电子）开关并以同样的工作方式操作，但是每一种使用不同的方式激活。双极型晶体管通过向其基极引脚提供一定量的电流来激活，金属氧化物半导体晶体管类似于双极型晶体管，但不使用电流，必须向其栅极引脚提供一定量的电压电平（通常为 5V 或者 12V）。光敏晶体管是三个晶体管中最特殊的，因为该晶体管不是通过任何电信号来激活的，而是用光线激活的。我们可以将所有这些类型的晶体管直接连接到 Arduino。

在数据手册中，所有类型的晶体管都有电压和电流（安培）的额定值——应该严格遵守额定电压，因为超过这个限制可能会损坏晶体管。额定电流应当用作指导，以确定开关在什么场合变得特别热。如上所述，你可以安装一个散热片和冷却风扇来散发晶体管中的热量，从而增加其额定电流。

1. 双极型晶体管（BJT）

双极型晶体管是最常见类型的晶体管，是一种电流驱动的放大器/开关，其输出电流与输入电流有关，称之为“增益”。通常有必要在 **Arduino** 和双极型晶体管之间使用一个限流电阻器，防止它接收过大的电流和过热。在电感负载产生逆电动势的情况下，晶体管也没有二极管的保护，所以如果要驱动电机或者继电器的螺线管，应该使用保护二极管，如图 2-10 所示。如果没有使用逆电动势保护二极管，一旦晶体管开关损坏，那么 **Arduino** 输出引脚可能会被损坏。

一个基本的 BJT 拥有三个引脚：基极（输入）、集电极（输出）和发射极（通用）。发射极总是连接到电源的正电压或者负电压（极性的选择取决于晶体管的类型），集电极总是连接到负载。基极引脚用于激活将发射极和集电极连接到一起的开关。有两种类型的 BJT 晶体管，根据半导体芯片上三个掺杂硅层的排列来标记。

- **正负正（PNP）**：可用作高侧开关，PNP 晶体管的发射极连接到电源的正电压，集电极连接到负载，基极用于激活开关。为了关闭该晶体管，其基极引脚必须等于发射极引脚（正电压供电，或者干脆切断基极引脚电源）。由于必须要向基极引脚提供负极电流，或者 0V（接地）信号，所以打开该晶体管不是很直观。
- **负正负（NPN）**：可用作低侧开关，NPN 晶体管的发射极连接到电源的负电压（接地），集电极连接到负载，基极用于激活开关。为了关闭该晶体管，其基极引脚必须等于发射极引脚（负电压供电）。该晶体管通过向基极引脚提供正极电压来打开（见特定晶体管额定值的数据手册）。

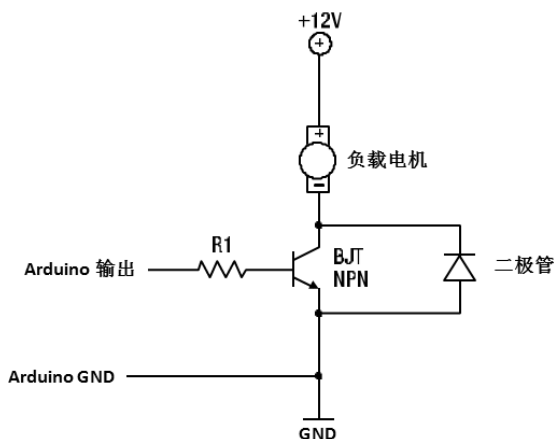


图 2-10 该原理图展示了 BJT 作为低侧开关，来驱动开关周围带有逆电动势保护二极管的电感负载（电机）。请注意，晶体管通过限流电阻器（R1）来驱动

大多数 BJT 需要逻辑电平信号（+5V），应用于基极引脚以便于激活开关。由于 BJT 属于电流驱动，所以当切断其基极引脚电源时，晶体管会迅速关闭。不同晶体管上用于切换所需的电流也是不同的，但是我们将只使用由 **Arduino** 提供的电平驱动的晶体管。普通的 2n2222a 型 NPN 晶体管完全可以使用几毫安的电流来启动，并且可以切换接近 1A 的电流，所以它可以被用作简单的低侧放大器开关。2n2907a 对应于 2n2222a，是一种 PNP 晶体管，通常用作简单的高侧开关。这些元件都可以在无线器材店、**Sparkfun.com** 和 **Digikey.com** 上买到，并且价格便宜（每个不到 1 美元）。

2. 场效应管

场效应管是一种用电压驱动的晶体管，而不是像 BJT 一样使用电流驱动。这种类型的开关也可以有非常高的 PWM 速度，并且通常具有很低的内部电阻，从而使得它们适合用于电机控制器。场效应管通常包括一个内部保护二极管（如图 2-11 所示），用于将输出电压与输入信号隔离开，并且免受负载产生的逆电动势损坏，因此它通常适用于将 **Arduino** 直接连接到场效应管开关，这是一个必须要添加到电路中的一个为数不多的元件。

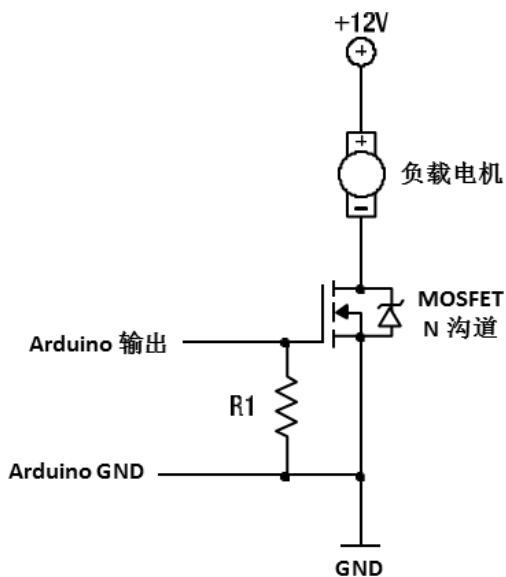


图 2-11 该原理图展示了一个用作低侧开关的场效应管开关（内置二极管），以驱动电感负载（电机）。注意：虽然没有所需的限流电阻器，但是不使用时用一个下拉电阻（R1）来使场效应管开关保持关闭

场效应管类似于 BJT 晶体管，因为它们都有相应的引脚和类型。场效应管引脚标记为栅极（输入）、漏极（输出）和源极（常用），分别对应于 BJT 晶体管中的基极、集电极和发射极（如图 2-12 所示）。另外，场效应管不会被标记成 NPN 或者 PNP，而是 N 沟道和 P 沟道，以表示其操作方式。出于实际目的，这些术语可以互换。因为场效应管开关由电压驱动，并且消耗很少的电流，所以不需要使用限流电阻器与场效应管栅极引脚串联（就像和 BJT 一样），但是在栅极引脚和源极引脚之间使用一个电阻器是个很好的做法（如图 2-11 中的 R1 所示），这样，不使用时可以完全关闭开关。

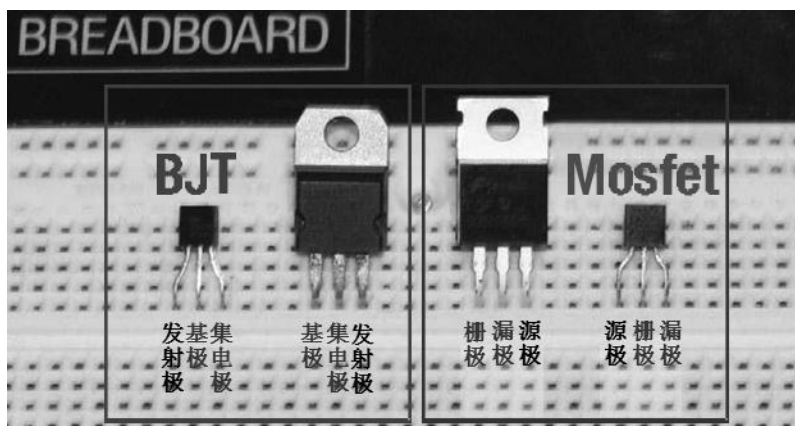


图 2-12 虽然它们物理上看起来是一样的，但是左侧的 BJT（晶体管）由电流驱动，右侧的场效应管（晶体管）由电压驱动。在每个晶体管下面，都标明了引脚——请注意，类似的晶体管封装有相应的引脚

3. 逻辑电平与标准的栅极场效应管

普通的场效应管如果完全打开，需要向基极引脚提供 10V 左右的电压。因为使用 Arduino 驱动任何超过 5V 的设备都需要使用电平移位器或者放大器，故我们使用所谓的逻辑电平场效应管来直接集成。逻辑电平场效应管可以使用 5V 的“逻辑电平”信号来打开，这能够很容易地连接到 Arduino 上。请记住，场效应管需要一个特定的电压电平来激活，但是电流很少。

场效应管也会对过多的栅极至源极电压比较敏感。如果超过范围，即使只是一秒钟，也会损坏场效应管，所以工作中应注意要将场效应管的电压保持在范围之内。施加到栅极引脚的最大电压在数据手册中罗列出来了，表示为“栅极至源极电压”或者“ V_{gs} ”——这个数字通常在 18VDC 和 25VDC 之间。

为了驱动标准的栅极场效应管，有很多不同的场效应管驱动集成电路，它使用逻辑电平输入信号和备用电源（通常为 12V）将输出信号放大发送到场效应管的栅极引脚。许多场效应管驱动器的目的是快速向栅极引脚提供大量电流，以达到高频 PWM 切换速度。更高的 PWM 频率适用于大电流驱动器，故我们将场效应管驱动集成电路应用到了本书中的几个项目中。

4. 场效应管电容

场效应管将微小的电容连接到栅极引脚，以保持栅极电压存在。即使切断栅极引脚的电源，电容电荷也使得场效应管可以保持激活状态。每次切换场效应管时，栅极电容必须满电荷，并且对其电路放电。由于这个原因，当 Arduino 不主动供电时，可通过使用“下拉”电阻耗尽该电容来保证栅极强制处于关闭状态（如图 2-11 中的 R1 所示），这是一个不错的想法。从栅极引脚到源极引脚使用一个 10kΩ 的下拉电阻（N 沟道上栅极到接地，P 沟道上栅极到电源）将足以在不使用时可使场效应管保持关闭。

随着施加到场效应管开关上 PWM 频率的增加，允许栅极电容充电和放电的时间不断减小。发生这种情况时，从驱动器到完全充电和放电，栅极电容将在较短的时间内需要大量电流。如果来自驱动器的电流无法满足切换周期之间的完全充电和放电，栅极将会停留在部分导通状态，最终导致产生过多热量。

据说，场效应管在快速切换时需要大量有效电流，这似乎很让人费解。因为场效应管需要特定电压来开启，并且电流通常都很小。虽然 Arduino 输出引脚能够提供的 40mA 电流对于慢慢地完全打开或者关闭场效应管来说是个很大的电流，但是在 PWM 频率情况下，场效应管电容每秒钟充放电次数要达到 10000~32000 次，这时 Arduino 提供的电流就无法使场效应管栅极电容完全充电或者放电了！

使用场效应管驱动集成电路（专用信号缓冲器）是驱动场效应管开关的最好办法，因为在每个切换周期中，它能够提供更比 Arduino 更多的电流。即使在很高 PWM 频率情况下，场效应管驱动器也能够向场效应管提供足够的电流以完成栅极电容的完全充放电，这对于无法有效驱动时减少开关产生热量是很重要的。当使用场效应管驱动器来控制场效应管时，可以忽略栅极引脚上的上拉或下拉电阻——反之，应该在场效应管驱动集成电路的每个引脚使用一个下拉电阻，该电路由 Arduino 的一个 PWM 输出引脚驱动。

5. 导通电阻— $R_{ds} (On)$

场效应管最重要的特性之一是开关打开时其漏极和源极引脚之间有内部电阻 (R_{ds})。这个特性很重要，因为开关的电阻决定了给定功率电平时所产生的总热量。通过查阅制造商数据手册可以知道最大 $R_{ds} (On)$ 值，然后使用 $R_{ds} (On)$ 电阻和将要通过开关的持续电流（安培）来确定消耗的最大功率。

6. 使用直流电机的 $R_{ds} (On)$ 和电流计算热量

场效应管的 $R_{ds} (On)$ 为 0.022Ω ($22m\Omega$)，并且持续电流为 10A，那么通过它的总功率是多少？使用第 1 章中图 1-3 的欧姆定律饼状图——我们想知道产生的瓦特数，那么要知道场效应管的电阻和通过电路的电流电平，就需要使用公式：瓦特=电流²× 电阻。

$$W = I^2 \times R$$

$$W = 10A^2 \times 0.022\Omega$$

$$W = 100A \times 0.022\Omega$$

$$W = 2.2W$$

这表示，如果试图通过开关的电流为 10A，那么 $R_{ds} (On) = 0.022$ 的场效应管将消耗 2.2W。根据我个人的经验，TO-220 封装内的场效应管消耗功率超过 2W 会导致场效应管过热。任何时候，如果想要降低操作温度，避免过多热量，良好的散热都是必需的，这时增加一个散热片或者散热风扇都是一个很好的想法。好的散热片和风扇能够极大地增加允许安全通过场效应管功率（或者热量）的总量。如果冷却方法无法很好地散热，你可以在并联电路中放置多个规格相同的场效应管来增加开关设备可流过的电流总量。假如你放置多个并联的场效应管，它仍然像是在操作一个开关，这是因为它们是同时打开和关闭的，并且它们的公共引脚是连接在一起的。

7. 并联场效应管

场效应管最有用的功能之一是能够将多个开关并联起来以增加电流量，并且减小电阻值。这是通过简单地将几个漏极端连接起来、几个源极端连接起来实现的（如图 2-13 所示）。几个栅极端应该使用同一个控制信号来驱动，但是每个场效应管都有自己的栅极电阻，以便分流相当于并联的每个场效应管总的有效电流——这些电阻值可能会非常低，阻值范围从 $10\sim 330\Omega$ 不等。

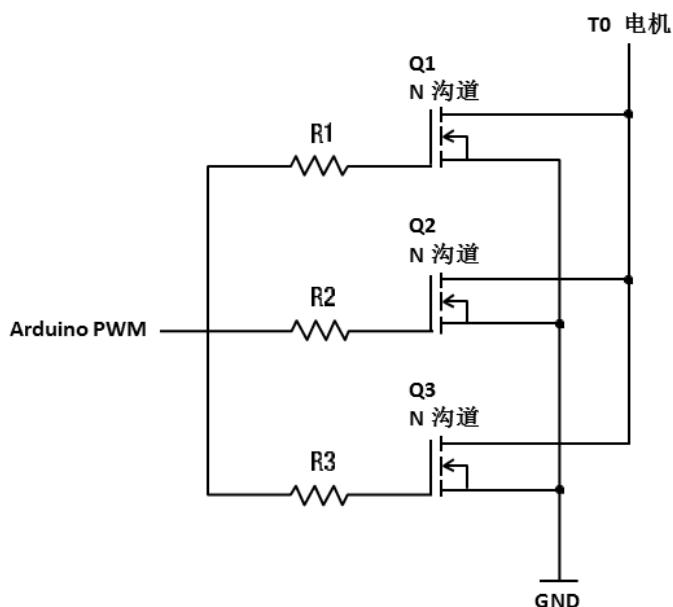


图 2-13 三个场效应管（Q1、Q2 和 Q3）构成并联电路（所有相似引脚绑定在一起）**当作一个**场效应管使用，允许 3 倍的电流通过，并且阻值只有一个场效应管的三分之一。电阻（R1、R2 和 R3）只要放置在均匀分配 **Arduino** 可用电流的地方，但也不是必需的

■ **注意：**即使使用并联方式，场效应管的电压范围也不会改变。如果超出了电压范围，你或许要烧坏连接的每个场效应管！

通过场效应管并联装置传输的总电流，等于通过单个场效应管的电流乘以并联的场效应管数目。除此之外，场效应管并联装置的总电阻值等于额定 $R_{ds(On)}$ 除以并联电路中场效应管的数目。这意味着，并联电路中使用两个场效应管，就可以减少一半的阻值——当电阻值减小，散热也随之减小。

8. 光敏晶体管

光敏电阻的操作类似于标准的 NPN 型晶体管，只是它使用来自 LED 的红外线来激活，而不是电流。这些晶体管通常用于 **寻线** 机器人来检测彩色表面不同的反射光。如果集成电路封装中包含光敏发射器和检测器，则该设备属于光隔离器，因为低功耗设备（红外线发射器）与大功耗开关（光敏晶体管）电隔离，故使得输入/输出电路相隔离（它们有不同的电源）。

该类型的开关就像是晶体管与继电器的混合体，像继电器一样有电隔离，但这种开关又被连接成像晶体管一样；你使用光敏晶体的独特功能是具有 PWM 切换能力的一个电隔离开关。图 2-14 中，使用连接到 **Arduino** 的红外线 LED 光来驱动基极（使用限流电阻器 R1），集电极（引脚 4）连接到负载负极端（作为低侧开关），发射极（引脚 3）连接到接地端。

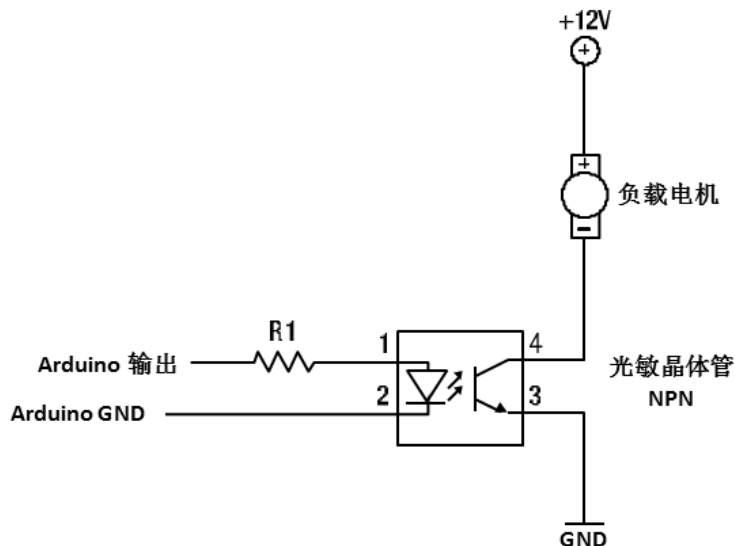


图 2-14 该光隔离器（红外线发光二极管与光敏电阻组对）被用作低侧开关来驱动电机。因为将 **Arduino** 连接到负载只是一束红外线光束（没有常用的接地信号），故你不必在开关上使用一个保护二极管（虽然推荐这么做）

2.1.3 电机控制器的接口

术语叫电机控制器的是一种放大器，旨在使用一组特定的信号命令控制电机的速度和方向。这里讨论的电机控制器有两种类型：电机控制集成电路和电子速度控制器（ESC）。

电机控制器集成电路是一种集成电路芯片，被设计成使用小功率输入信号产生大功率输出信号来控制直流电机的速度和方向。这通常需要一些额外的元件（几个电阻、一个电容和一个来自 **Arduino** 的+5V 电源），但是要注意电机控制。

电子速度控制器是一个纯粹的电机速度控制器电路，接收一种或者多种类型的输入信号，并且使用 PWM 向电机输出适当的速度和方向。通常有一些预封装单元花费成本较高，但是只需要很少的工作量就可以使用。如果你时间比较紧，这些都是很方便的。大多数的

电子速度控制器适合用在制作玩具飞机、汽车和舰船等设备，并且使用伺服脉冲输入信号。现在也有越来越多面向机器人应用的电子速度控制器，具有各种不同的接口选项。

1. 电机控制器集成电路

现有的几种封装集成电路芯片价格便宜，并且易于嵌入到电路中。常见的 L293D 型双电机控制器是一种 16 引脚双列直插封装集成电路芯片，包含两个受保护的驱动电路，在 36VDC 状态下能够向每个电机提供高达 600mA 的持续电流（如图 2-15 所示）。L298N 是一种类似的芯片，可以向每个电机提供 2A 的电流。这些芯片（以及其他一些）接收标准的 0~5V 输入信号，有内部逻辑门电路来防止意外过载或者命令控制器变为损坏状态。

请注意，在图 2-15 中有四组输入输出，标记为 1A（输入）和 1Y（输出）至 4A 和 4Y。这些输入引脚的数字状态决定了其相应的放大器输出引脚的数字状态。实际上，当你向引脚 1A 提供一个 5V 信号（VCC1）时，你会在引脚 1Y 获得一个 12V 的信号（VCC2）。每组输入/输出也有一个“使能”引脚。1-2 使能引脚同时控制 1Y 和 2Y 输出引脚的状态，3-4 使能引脚控制 3Y 和 4Y 输出。可以使用 **Arduino** 上的数字引脚来控制四个输入引脚，设置电机方向，而使用每个使能引脚上的 PWM 信号来设置每个电机的速度。

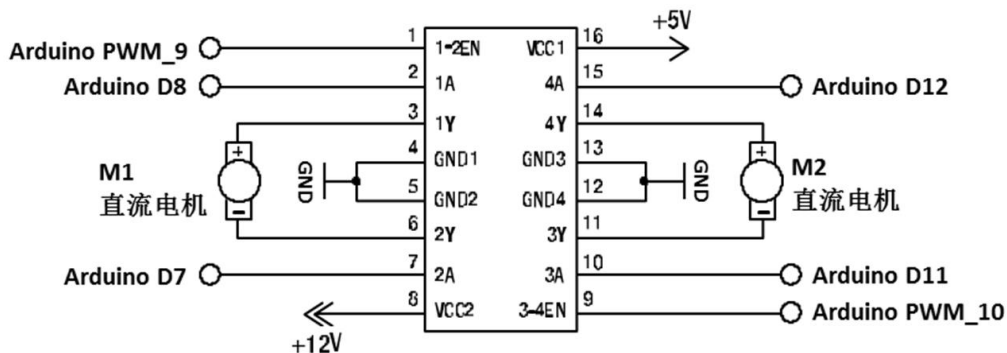


图 2-15 L293D 型 1A 双电机控制器集成电路原理图，以及如何连接到 **Arduino**

如果一个电机控制器集成电路符合你的需求，但你又不想创建自己的电路，那么可使用小型电机控制器集成电路的商业工具套件和现成的预组装电路。第 4 章中 **AdaFruit** 电机保护器的特点是两个 L293D 型电机控制器集成电路（如图 2-16 所示），每个能够向四个直流电机提供 600mA 的持续电流。只需将保护器插到 **Arduino** 上，即可运行——这个保护器不用来给四个直流电机提供持续电流的话，还能用导线将其连接成作为两个伺服舵机连接器使用（更多信息见第 3 章）。

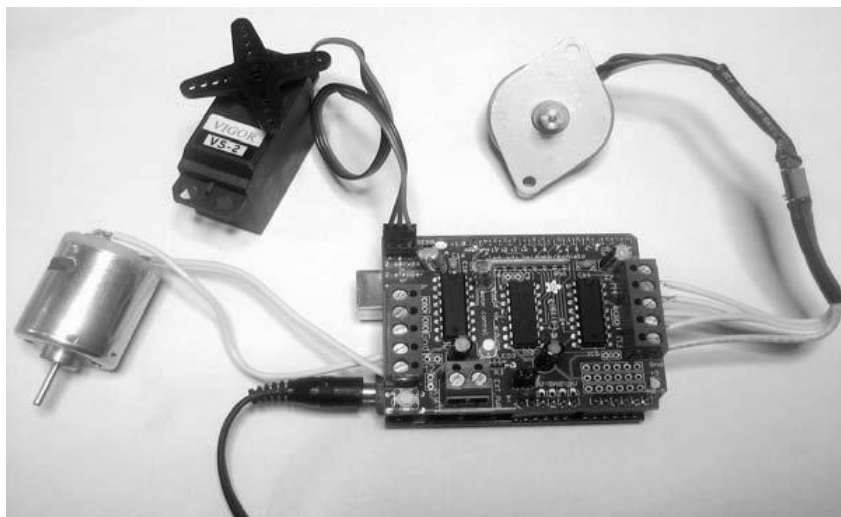


图 2-16 对于 **Arduino** 来说，**AdaFruit** 电机保护器是一个易用的电机驱动器，可以驱动各种不同的电机，从左到右分别是：直流电机、伺服舵机和步进电机

2. 电子速度控制器（**ESCs**）

电子速度控制器是一种电机控制器，具有自己的控制电路。这些控制电路都倾向于使用特定输入信号来驱动，并且不需要微控制器。有很多种不同的预组装的电子速度控制器，用于接收特定的输入信号并且很适于控制电机的玩具型车辆（汽车、舰船和飞机）和机器人。虽然很多这样的单元被设计用来接收来自标准玩具 **R/C** 系统的伺服脉冲信号，或者电位器模拟电压，但你可以使用 **Arduino** 来仿真伺服脉冲或者模拟电位器的值。这可以使你通过 **Arduino** 使用任何输入方法来控制专业的 **ESC** 电机控制器。

对于本书中用到的任何电机控制器，来自 **Dimension Engineering** 的 **Sabertooth 2x25** 型电机控制器（如图 2-17 所示）是一个通用的替代品。可以使用 **Arduino** 向 **Sabertooth** 发送控制信号来控制每个电机，使用的信号各不相同。电路板内部设置过量电流保护装置，所以当过热时它就可以进行关闭。

现有许多不同的速度控制器，每个都需要一个不同的控制接口；有些是 **PWM** 控制的，另一些使用串行命令，其他的则使用模拟电压。许多玩具电子速度控制器使用伺服脉冲作为输入信号来控制每个电机。通常使用的主要有四种电机控制接口方法，简单介绍一下其中的每一个。

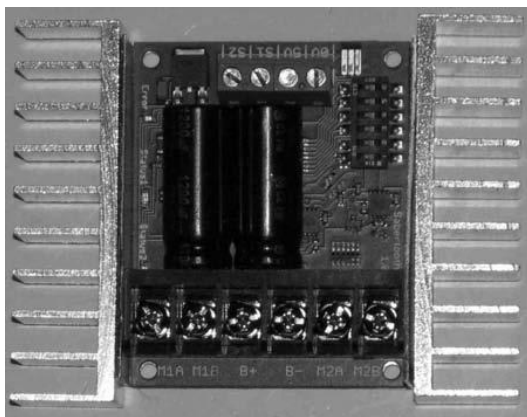


图 2-17 这就是 Sabertooth 2x25 25A 双电机控制器。虽然它看起来貌不惊人，但可以给两个直流电机提供足够的功率来以不错的速度搬运数百磅东西

- 简单的 PWM 控制：使用 PWM 信号来确定一个电机 0~100% 输出。PWM 信号的频宽比决定了电机单方向上按比例输出的速度。将自制电机控制器直接连接到晶体管开关，以及使用某种类型的场效应管驱动器集成电路，这是常见的控制方法。在两个方向上来驱动电机，需要两个 PWM 信号或者一个高速信号反相器。
- 双向模拟控制：使用 0~5V 模拟（或者 PWM）信号来决定控制一个电机的速度和方向。在这种模式下，2.5V 的中心位置是控制状态的中立点（静止状态），低于 2.5V 时电机反向旋转，随着电压降低按照比例速度递增，当达到 0V 时速度增加到反向 100%，而高于 2.5V 时电机正向旋转，达到 5V 时速度增加到正向 100%——这种信号称为“模拟”信号。
- R/C 控制：使用的是一种根据 R/C 发送器控制杆的对应位置解码产生特殊的“伺服”脉冲信号。该信号是一个电脉冲，其具有特定的时间基准，范围从 1ms 到 2ms，且中心位置会产生 1.5ms 脉冲。该类型的接口是为了直接连接到大多数爱好者的 R/C 无线系统设计的。
- 串行命令：使用串行数据线连接到计算机（USB）来接收一系列串行控制脉冲，用于控制电机。

2.2 用户控制

现在如何准确定义“机器人”存在不同的见解。有些人认为，它未必是个机器人，除

非具有一定的智能，例如能够根据所处周围环境做出决定。大多数工业机器人设备的目的是以一定的精确度完成指定的任务，周而复始——但是这些机器人都是由人来控制的，没有决策能力。有些人则认为，即使被人类控制，它也包括自动化过程。考虑到兴趣的多样性，我们扩大范围来将包含所有这些形式的都**当作**机器人。

机器人的控制方法根据其具体应用而有所不同。当用户想要进行一个特定的动作，固定的机器人可能只需要一个控制命令，而移动迅速的机器人每秒钟需要变更几种控制。自主机器人也可以有 R/C 控制，以允许有多种用途。**Arduino** 芯片拥有足够的空间来在同一个机器人上编写多种控制方法。这些控制方法可以简单到给机器人安装一组按钮，或者复杂到使用自身的 GPS 芯片将机器人引导到达一组经纬度坐标的位置（见第 9 章“机器船”中的相关内容）。虽然有许多不同的控制方法，但是我们只讨论一些常见的类型。

2.2.1 连线（有线）控制

连线控制很容易连接到 **Arduino** 上，因为你只要简单地用导线将每个按钮、开关或者电位器直接连接到 **Arduino** 输入引脚。你甚至可以构建自己的控制盒，只需要用几个电位器、一些按钮开关和几**英尺** 8 号恒温导线，这些都可以从五金店买到。使用一根 8 号导线和一个公共接地信号，你可以在一个有线控制器上轻松地拥有六七个独立的模拟信道或者数字信道。

医疗电动轮椅通常有一个摇杆控制器，操作者可以控制其移动，以及用几个按钮来调整从电机最高速度到座椅倾斜度的任何东西。这些控制器中的可变电阻器和按钮直接有导线连接到轮椅的微控制器上。第 11 章的项目是一个人可骑的赛格威式机器人，可以通过车把手来激活并控制——所有的连接都是硬连接到 **Arduino** 上的。

2.2.2 红外控制（IR）

一组红外线传感器可用来控制机器人，就像使用远程控制来改变电视频道一样。电视机遥控器对于每个按下的按钮都发送一个特定的红外线编码，这样电视机就会根据你所按下的按钮知道要做什么（比如音量增大/减小、频道向上/向下，等等）。按照同样的原理，可以在 **Arduino** 上使用 IR 接收集成电路读取的红外线编码（如图 2-18 所示），然后使用这些编码来控制机器人，按下每一个不同的按钮表示不同的机器人的动作。

这种类型的控制使用红外发射二极管发出的光，该二极管的外观和操作就像是一个 LED，不同的是，打开红外发射二极管时无法看见任何的可见光（用肉眼）。因为红外线

设备使用光线来传输信号，故这种类型的控制方法必须在发射器和接收器之间有清晰的视线。在传输路径清晰的情况下，IR 传感器的有效范围是 20 英尺左右。甚至有一些 R/C 汽车、室内微型直升机和玩具机器人都使用红外线无线连接，例如来自 WowWee 的玩具机器人（流行的 Robosapien）（www.wowwee.com）。

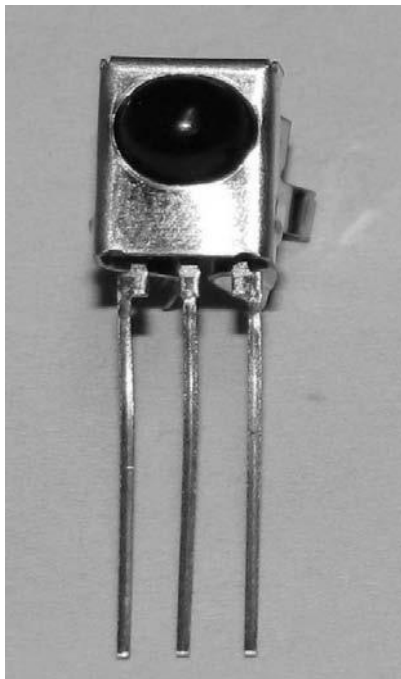


图 2-18 使用从一个旧的 VCR 拿来的红外线接收集成电路，你可以使用电视机的遥控器向 Arduino 发送信号

2.2.3 无线电控制系统

首先要声明的是，我绝不是一个无线电专家，因此没有资格详细介绍关于无线电信号传输的很多不同的可能方案——但是我已经尝试过几种不同的流行无线电控制方法，可以根据我的机器人使用经验告诉你哪一个方法效果最好。

无线电控制或许是人类控制机器人最常用的方法，因为它对于视力操作方面没有严格的限制，并且有效范围可达几英里。大部分的无线电控制设备旨在用于玩具飞机、舰船和汽车，但是 Arduino 拥有良好的接口来实现机器人的 R/C 控制。一个典型的玩具型无线电控制系统包含一个“发射器”，用于捕获你的输入，并且通过空气发送出去，还包括一个“接收器”，用于从空气中捕捉这些信息，并将它们转化成可用的电信号（如图 2-19 所示）。



图 2-19 这是个玩具型 2.4GHz 的无线电系统，包含一对六信道发射器和接收器，可以以不到 40 美元的价格从 HobbyPartz.com 上买到（零件号 79P-CT6B-R6B）

通常有三种可用于控制机器人的无线电系统：调幅（AM）、调频（FM），以及扩频（2.4GHz）。尽管这些类型信号中的每一种变化都是不同的，但是都实现了相同的目的：将发射器的值发送至接收器。对于更高级的项目，还会有 **Xbee**，这是一种 2.4GHz 的无线连接。

1. 调幅（AM）

AM 无线电系统很可能是应用最广泛的，但是主要用于远程控制玩具汽车和舰船（27MHz 和 49MHz 很流行）。这些无线电系统使用长导线天线来发送信号，会受到来自附近建筑、树木和其他无线电（处于同一频率）干扰的影响。AM 无线电系统通常拥有不超过三个的可用控制通道，使得它们很少用于机器人。这些系统可以用在小型机器人上，但是应该避免用在需要极高可靠性和范围的场合。这些系统能够连接到 **Arduino**，但是应该避免使用，除非你已经有了一个。

2. 调频（FM）

在美国，所有 FM 75MHz 的无线电系统都是为了用于地面车辆（比如汽车和舰船），并且所有的 FM 72MHz 无线电系统都是为了用于飞行器（比如飞机和直升机）。这些无线电系统通常都需要在发射器和接收器上放置一组匹配的晶体来确定无线电工作于哪个“通道”——这些晶体就是将无线电 Tx 和 Rx 调整到同样的特定频率，这样它们就可以无干扰地工作。在开阔地区，FM 无线电系统通常有良好的通信范围（高达 1 英里），并且不

会像 AM 系统那样容易被干扰，使得它们适用于机器人。

3. 扩频 2.4GHz

2.4GHz 的频率范围常常用于无线数据传输。它不仅频带宽，足以传输大量数据（例如音频、视频和 Web），还可以为免受干扰的 R/C 系统提供安全连接。有很多设备使用了该频段，例如无线摄像机系统、无线网络路由器、R/C 系统，甚至蓝牙无线协议。扩频在 2.4GHz 频率工作，通常超过来自于金属栅栏和电机噪声的干扰电平，并且没有使用限制，这使得它们非常适合用于任何类型的机器人。

2.4GHz 的 R/C 系统使用一种称为“绑定”的过程在发射器和接收器之间建立半永久连接。绑定是频率匹配的晶体对的替代品，用在传统的 R/C 系统中。扩频标准使用一种称为“调频”的过程来不断改变频率通道（发射器和接收器两者同时），以防止附近其他任何无线电设备的交叉信号。R/C 系统是获得 2.4GHz 安全连接的最便宜方式，可以直接连接到 **Arduino**。

4. Xbee

Xbee 作为无线串行数据连接（有可选择的数据速率），能够同时发送和接收数据。**Xbee** 无线电通信设备是由 **Digi International** (www.digi.com) 制造的，使用“**Zigbee**”无线通信协议。对于各种应用，这些无线电通信设备很容易连接到 **Arduino**，只需要 3.3V 的电源，并且连接到 **Arduino** 的串行 Tx 和 Rx 引脚（D0 和 D1）。

使用一组 **Xbee** 无线电模块，就可以创建你自己的自定义可编程 R/C 控制器或遥感连接，用来获得机器人在操作期间的有关信息（例如电池电压、电流和速度等），甚至可以使用一组 **Xbee** 无线电模块来为 **Arduino** 做无线编程。它可使用不同的功率电平来适应更长距离的传输，还可以提供与常见的 2.4GHz R/C 连接相当的有效的有效的工作范围。由于这种强大的无线连接的多功能性，故本书将会有多个章节来介绍使用一组 **Xbee** 进行机器人的控制和监测（如图 2-20 所示）。

有很多零售商都销售 **Xbee** 无线电模块和保护器适配器板，能够轻易地与 **Arduino** 进行连接。每个基本的 **Xbee** 无线电模块成本大约是 25 美元，每个适配器板的成本从 10 美元到 25 美元不等。适配器板是必需的，因为每个 **Xbee** 无线电模块的引脚间距是 0.05 英寸，它无法与使用 0.1 英寸间距的面包板或者穿孔原型板相兼容。在图 2-20 中，可以看见两个标准的 **Xbee** 无线电模块（Sparkfun.com 零件号 WRL-08665），一个连接到 **Arduino** 的 **Sparkfun Xbee Explorer Regulated** 分线板（Sparkfun.com 零件号 WRL-09132），另一个

连接计算机的 **Sparkfun Xbee Explorer USB** 分线板 (Sparkfun.com 零件号 WRL-08687)。**Xbee** 无线电模块的制造商也开发了一个称为 X-CTU 的软件程序, 连接到计算机时, 可以用来修改 **Xbee** 无线电模块的设置——X-CTU 软件免费使用, 但是当前只能在 Windows 下使用。

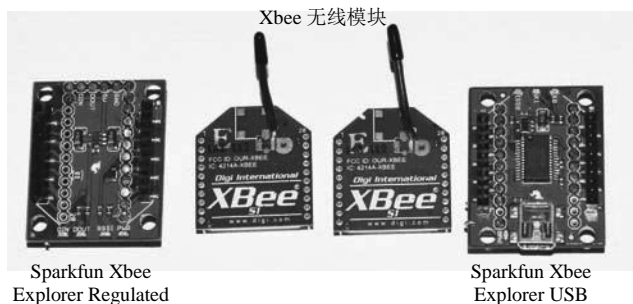


图 2-20 一对附带分线板的 2.4GHz**Xbee** 无线电模块 (中), 图片由 Sparkfun.com 提供

2.3 传感器导航

尽管创建一个将用户输入转换成机器人输出的控制连接会非常有用, 但是有一些任务需要机器人自己做决定, 而不需要人控制。本书中三分之一的机器人项目 (第 4、5 和 7 章) 使用某种类型的外部传感器来指导机器人到达指定地点, 而且无需任何的用户指导。

不管你相信与否, 你实际上是个自主 (自我控制) 的物体, 因为你使用几种不同的“传感器”来帮助确定你周围的环境。你的眼睛、耳朵、鼻子、双手和嘴, 每一个都有自己的触觉, 你的大脑会解释成某种形式的智能。通过这些传感器, 大脑能够做出明智的决定, 身体应该如何行进使你避免受到伤害。同样, 机器人传感器是一种机器人随身携带的设备, 用于收集其周围地区的信息。如果没有传感器, 机器人将无法知道自身周围有什么, 或者如何行进。这是给你的机器人添加智能的最简单方法。

传感器的类型有很多种, 每一种读取不同的环境信息, 因此最常见的是向机器人添加几种类型的传感器, 用来有效地绕过障碍物、收集重要的信息。传感器能够测量光线、距离、近物体、热、气体、加速度、角度、湿度和水分、接触, 以及旋转位置 (除了个别之外)。我们重点关注那些易于使用, 且性价比最高的传感器。

2.3.1 接触式感知

可以实现的最简单的一种传感器就是接触开关，它能告诉机器人是否触碰到某些东西。此种传感器通常被称为“凸点开关”，用在 iRobot 公司生产的 Roomba 机器人吸尘器中，以确定是否碰到墙壁或者其他别的物体。这种传感器类型的主要注意事项是，只有当机器人与一个物体产生物理接触才会被检测到。

凸点开关

凸点开关是一种简单形式的传感器，因为它仅仅包含两个电触头（如图 2-21 所示）。一旦触头被触碰，开关就会闭合；否则开关打开。我们使用这种形式的开关作为一种方法，当机器人碰撞到某东西时就会告诉它。如果我们在机器人身上放置多个触头，那么不仅会知道机器人何时碰到其他东西，而且还能够确定最好的行驶方向来避免再次碰到同一个物体。

这种传感器也可以用作行程开关。这些传感器通常被安装在车库开门器上。当门打开到某一确定位置时，会触碰到行程开关，然后主板接收到停止卷扬电机的命令。这就是它怎么知道在哪里将门关闭的原理。

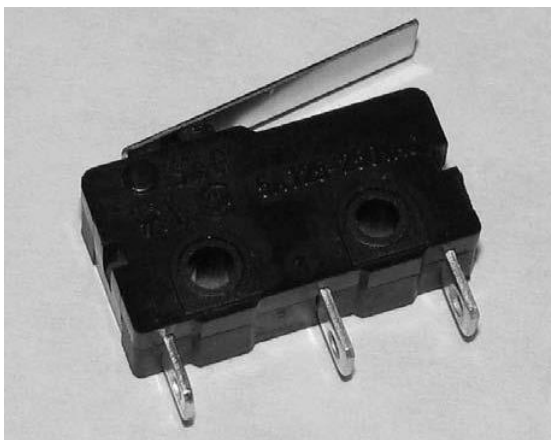


图 2-21 一个典型的带有控制杆的凸点开关

这类传感器（或者开关）被当作数字输入来读取，需要用到 `digitalRead()` 命令（见第 1 章中的清单 1-1）。

2.3.2 距离和反射式感知

当试图在无须触碰的情况下确定机器人附近是否有物体存在时，范围探测就会很有用。

一个好的范围探测器能够计算出与物体之间的距离，精度达到英寸左右。范围探测传感器使用反射的声波或者光波来测量一定范围内传感器和任何障碍物之间的距离。不同范围的探测方法有不同的有效范围、精度和价格。范围探测传感器的有效感知范围为 1 厘米到 25 英尺，价格从几美元的红外线测距仪到上千美元的激光测距仪。我们将使用第 4 章中用于莱纳斯的红外线传感器，以及第 7 章中用在 Wally 上的超声波测距仪。

1. 红外线传感器

红外线探测器使用红外线发射器发出红外线“光包”，并且使用探测器来检查从周围任何物体上反射回来的光。通过测量红外光返回探测器所花费的总时间，传感器能够确定与物体之间的距离。红外线测距仪能够探测到距离达 5 英尺远的物体——用超声波测距仪可探测更远的距离。夏普 GP2 系列的红外线近距离传感器可以从 Sparkfun.com (零件号 SEN-08958) 上以低于 15 美元的价格买到，可用来探测远至 5 英尺的短距离物体（如图 2-22 所示）。



图 2-22 这是一个夏普 GP2 红外线测距仪（Sparkfun.com 零件号 SEN-08958）

一对简单的红外线发射器和探测器可在近距离范围内（3 英尺以内），用来确定一个表面对红外光近似的反射率。这些简单的红外线发射器和探测器对是第 4 章中寻线追踪机器人的基础。发射器和探测器并排安装，并且朝向同一个方向。发射器不断地向地面发出一连串红外光，而探测器则不断地读取从地面反弹回来的反射光。

我们使用一条反光带作为机器人追踪的引导轨道。当机器人偏离了这条反光带，每一侧的红外线传感器接收到的红外光开始变少，因此调整电机的输出来保持机器人在反光带上前进。使用这种简单的引导方式，我们能够通过改变反光带路线轻易地修改机器人追踪的路径。现有好几种不同类型的红外线发射器和光敏晶体管封装可用于寻线追踪机器人（如图 2-23 所示）。



图 2-23 这是三种不同类型的红外线发射器和探测器对。最右边的红外线对用于莱纳斯，这是一种第 4 章将介绍的寻线追踪机器人。这些传感器的价格范围从 1 美元到 3 美元不等，可以从 Digikey.com 上买到

■ **注意：**很多使用遥控装置的家用电器都包含一个实用的红外线接收器，同你在 Radio Shack 或者 [Digikey](http://Digikey.com) 上可以找到的一样。如果你碰巧有一个损坏的 VCR、DVD 播放器、TV 或者立体音箱，而且也不介意拆开，那么你可以从 PCB 上拆下红外线传感器，还可以节省几美元。它们通常只有三个引脚：+5V、GND 和信号。

2. 超声波测距仪

超声波测距仪使用附近物体反射回来的高频声波来计算它们之间的距离。有些超声波传感器需要一个微处理器来发送和接收传感器信号，而另一些传感器则在传感器内部计算距离，并且产生一个易于被 [Arduino](http://Arduino.com) 读取的与距离或正比例的输出信号。

超声波测距仪可用在各种确定探测区域宽度的波束角中。窄波束角更适合探测距离较远的物体，而宽波束角则探测短距离物体更好一些。这些传感器的价格通常都在 30 美元到 50 美元之间，并且能够很容易地被 [Arduino](http://Arduino.com) 读取。

[MaxBotix](http://MaxBotix.com) 品牌的超声波测距仪有内置处理器，使得它能够在同一时刻输出独立的串行、模拟和 PWM 信号来提高连接的灵活性（如图 2-24 所示）。这些测距仪的精确测量距离约为 6 英寸到 25 英尺，非常适合于机器人的避障和探测。我更喜欢使用该品牌的超声波探测仪，因为它可靠性好，并且易于使用三个内置输出信号中的任何一个连接到 [Arduino](http://Arduino.com) 上。

3. 激光测距仪

这种类型的探测仪使用激光来扫描其周围的物体，就像是杂货店结账台的一台激光扫描仪。激光测距仪可以有宽达 240° 的视角，使它比其他传感器有更宽的周围视角。每次

激光产生一个旋转，它都以设置的间隔获取距离读数。当旋转完成时，该信号被编译，产生一张周围区域的快照。虽然这种传感器功能先进，但是最大探测范围只有约 15 英寸，而且价格昂贵（通常要花费 1000 美元左右）。在价格降下来之前，我们是不会测试任何一种这些单元的。



图 2-24 MaxBotix LV-EZ0 型号的超声波探测仪，有效探测范围为 6 英寸到 25 英尺（Sparkfun.com 零件号 SEN-08502）

2.3.3 方向（定位）

有几种不同的传感器，可以确定机器人的一个或多个位置和方向。GPS 传感器可以使用经纬度坐标告诉你传感器在地图上的哪个位置，而加速度传感器和陀螺仪传感器可以告诉你角的位置（倾斜）或者机器人的旋转角度。使用这些传感器，可以为赛格威（Segway）型机器人创建一个自动平衡的平台，或者向机器人上传一组 GPS 坐标来导航。

1. 加速度传感器

加速度传感器用来测量重力或者加速度。通过一个沿其测量轴倾斜的加速度传感器，可以读取相对于倾斜量的重力。加速度传感器有多达三个感知轴，可以使用模拟输出信号直接连接到 Arduino。目前，很多设备将加速度传感器用作输入控制、震动检测、稳定平台和自动平衡或倾斜接口——可以在这些产品中找到这些设备，例如数码相机、手机、笔记本电脑以及任天堂的 Wii 控制器等。

如今，大多数可用的加速度传感器都是小型表面贴装元件，但是也有一些，焊接了传感器及所需的全部滤波元件（电阻和电容）的不同分线板，这样你就可以很轻松地将它们连接到 Arduino 上。Sparkfun.com 上有大量可供选择的“Arduino 配备”传感器

电路板，价格范围从 20~50 美元不等。

能够被加速度传感器测量的有三个轴，被标记为 X 轴、 Y 轴和 Z 轴，分别对应于倾斜、俯仰角和偏航（如图 2-25 所示）。单轴加速度传感器测量 X 轴或者 Y 轴，双轴加速度传感器同时测量 X 轴和 Y 轴，三轴加速度传感器测量所有三个轴。每个测量轴表示传感器上一个独立的自由度（DOF）——因此，三轴加速度传感器也可被标记为 3DOF。

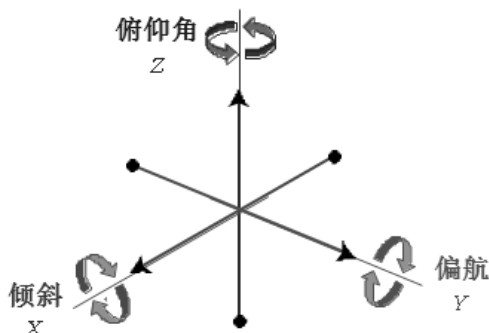


图 2-25 该图展示了三个转动轴：倾斜、俯仰角和偏航，以及对应的加速度传感器符号 X 、 Y 和 Z

加速度传感器用于测量重力的改变，但是它们也会对震动和突然移动或者冲击非常敏感，从而导致输出信号失真。如果加速度传感器用于测量角度近似值，则拥有另一台能够纠正震动造成短期错误的设备会是很有帮助的。“陀螺仪”传感器能够测量角度变化，但由于使用了不同的方法，对于突然的重力改变敏感度较低。

2. 陀螺仪传感器

陀螺仪是一种传感器，用于检测沿其测量轴的旋转速度变化。这些设备常常应用在自动平衡项目中，与加速度传感器协同使用来产生机器人稳定的角度近似。就像加速度传感器一样，陀螺仪传感器也通过所测轴来标记： X 轴、 Y 轴或 Z 轴。

陀螺仪传感器可以在短时间内获得良好的测量结果，但是由于具有长时间被称为“漂移”的机械错误，从而导致即使没有移动也会偏离起始点。为了纠正此漂移，我们需要使用一个长时间稳定角度读数的加速度传感器来作为陀螺仪传感器的参考点。

3. 惯性测量装置（IMU）

为了解决单独购买加速度传感器和陀螺仪传感器的麻烦，制造商将多个加速度传感器和陀螺仪传感器综合在同一个电路板上，以节省空间和成本。由多个相互之间密切合作的

传感器组成的电路板被称为惯性测量装置（或者 IMU）。如图 2-26 所示的 IMU 拥有两个陀螺仪传感器和一个加速度传感器来测量总共六个轴的数据。通过结合来自加速度传感器 X 轴和陀螺仪传感器 X 轴的读数，可以为本书第 11 章中的单轴自动平衡平台创建一种滤波角的测量方法。

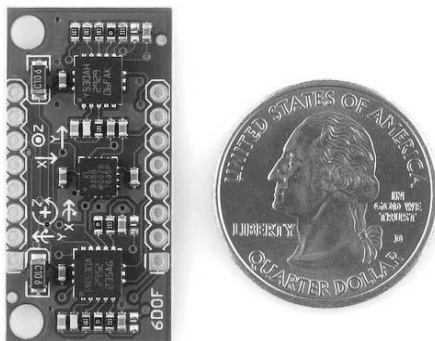


图 2-26 这是本书第 11 章中用到的 Sparkfun Razor 6DOF 惯性测量装置（零件号 SEN-10010）。IMU 包含两个陀螺仪传感器和一个加速度传感器，总共提供六个 IMU 方向的独立测量相位

Sparkfun.com 为加速度传感器、陀螺仪传感器和 IMU 提供很好的教程和购买指南，说明它们的特性和功能。如果你有兴趣了解更多关于这类传感器的信息，绝对值得读下列的书。

Sparkfun.com IMU 教程：<http://www.sparkfun.com/tutorials/167>

4. 全球定位卫星（GPS）

全球定位卫星（GPS）系统是一种传感器，使用来自太空中多个追踪卫星的信号来计算自身的位置，然后输出一组特定的经纬度坐标。虽然传感器同一时刻连接的卫星最多可达 20 个以获得最佳信号，但是要确定其自身位置必须能够至少获得三个 GPS 卫星的信号。该传感器可以以 50~150 美元的价格买到，提供位置、时间和速度的测量。大多数装置只需要一个电源和接地信号来开始输出串行数据流，它可以被 Arduino 读取到。

EM406 GPS 传感器（如图 2-27 所示）输出的信号是一个 NMEA 标准的字符串，必须通过解码来获得有效信息，例如经度、纬度、速度和方向。该信号使用标准串行连接发送至 Arduino，可与其他传感器结合用于使机器人从一个点自主导航到另一点。尽管 GPS 能提供关于机器人速度和方向的信息，但是它必须通过移动来计算这些值。借助于简单获得的两个位置读数之间的差异，可以确定机器人的行驶方向，并且相应地调整其功率，将其引导至正确的结束坐标处。



图 2-27 从 Sparkfun.com 购买的 EM406 GPS 传感器（零件号 GPS-00465）可以直接连接到 [Arduino](#)，用来制作自主导航车辆（例如本书第 9 章的机器船）

GPS 比较独特，几乎可以在世界上的任何位置访问卫星，但是由于它所连接的卫星在太空中（距离确实很远！），故地平面精度有所不同，从 10 英尺到 30 英尺不等。这意味着，每次只使用标准 GPS 导航，你不能指望机器人会在同一个点停止。机器人最终可能与你预想的目标点相距 10 英尺，并且路径可能与你期望的有几英尺的差异。GPS 也无法检测到其路径上移动的物体和树木，因此如果机器人在地面上行驶，安装其他目标检测传感器来协助 GPS 导航会是个好主意（例如超声波探测仪或者碰撞传感器）。

5. 实时活动（RTK）GPS

为了消除标准 GPS 系统的精度误差，有一种系统被称为实时活动（RTK）GPS，它使用已知位置的单独“基站”来计算 GPS 接收器的精确位置，误差降至 1 厘米！运行此种 GPS 通常需要更多设置，至于如何设置的问题已超出了本书介绍范围。但如果你需要 1 厘米精度的 GPS 单元，这也不是没有可能做到的。

2.3.4 非自主传感器

一些传感器并非用于自主控制，而是让用户可以去做其他不可能的事情。例如无线摄像头可以让操作者处于一个位置，而机器人在另一个不同的位置工作。还能够使用超声波测距仪、碰撞传感器、GPS 定位、温度传感器，或者硬件监控设备来协助你操控机器人。

在本书第 8 章中，我们将为大型动起来的机器人制作一个双电机控制器，并且安装了内置电流传感器来监测通过控制器的总电量。如果电流超过代码中预定义的电平，[Arduino](#) 立即关闭电机电源（在几秒钟之内），以防止过热。这种类型的保护对于任何工作量未知的机器人来说都是必不可少的。让机器人停止响应几秒钟来降温总比因过热而烧坏要好一些（需要维修）。

1. 摄像头

到目前为止，在机器人上放置的大多数即时反馈传感器都是无线摄像头。有好几种 100 美元以下的系统，包括无线彩色摄像头（带音频）和接收器，该接收器可以插入电视、计算机，或者 R/C 发射器上安装的小型 LED。这些系统通常基于 900MHz 或者 2.4GHz 的无线链路的创建，根据无线电强度的大小和发射器与接收器之间障碍物的不同，可以在从 300 英尺到 800 英尺的任何地方传输视频信号。

可以使用带有麦克风的 2.4GHz 无线摄像头（如图 2-28 所示），从机器人上将音频和视频信号传输到你可以看见并且控制的命令中心。让机器人去你需要它去的地方，这样会保证你的安全。再者，该传感器能够很好地与其他类型的传感器一起应用，以通知机器人是否碰到某些东西，或者是否正在靠近摄像头无法看到的物体。



图 2-28 从 [Sparkfun.com](#) 购买的 2.4GHz 无线摄像头（零件号 WRL-09189），大约 75 美元，是一种为机器人添加一对远程眼睛和耳朵的好方式

2. 电流传感器

电流传感器用于测量给定时刻通过某一给定点的电流。如果任一电机消耗过量电流，我们可以编程 [Arduino](#)（控制电机控制器），用来在指定时间内（1~2s）终止驱动电机，以避免过热。通过保护电机驱动器，就不大可能会导致失效。

虽然有好几种类型的电流传感器，但是我更倾向于使用电流传感器 IC，例如 Allegro MicroSystems 公司生产的 ACS-712（5A）或者 ACS-714（30A）（如图 2-29 所示）。尽管只有 8-SOIC（外形小的集成电路）表面贴装封装，但是它很容易同时连接到 [Arduino](#) 和电机

上。该集成电路只需要+5V 和 GND 信号在 VOUT 引脚开始输出模拟电压，而且很容易使用 Arduino 上的任何模拟输入引脚进行读取——甚至可以使用 Arduino 规定的+5V 向电流传感器集成电路提供电源。

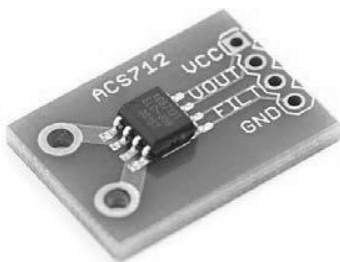


图 2-29 ACS-712 型双向电流传感器可用来与一个负载终端串联，以测量电流电平。它能够很容易通过 Arduino 上的模拟输入进行读取

在图 2-30 中，可以看到一个简单的原理图，说明如何将 ACS-714 电流传感器连接到 Arduino 上。电流传感器集成电路需要一个 5V 电源，以及一个连接过滤引脚和 GND 的旁路电容。然后，你必须通过传感器的电流感知引脚至少给一个电机的电源线进行路由。

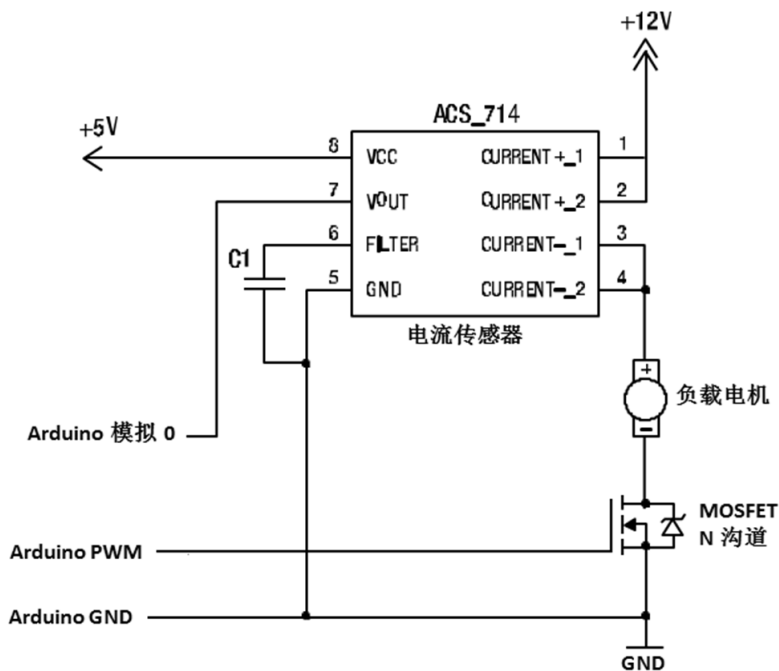


图 2-30 电路中 ACS-712/714 型电流传感器集成电路的简单原理图

在本书第 8 章 探险者机器人中,我们使用 ACS-714 型双向 30A 电流传感器(如图 2-29 所示 ACS-712 的兄弟型号)来感知通过任一驱动电机的电流。使用 Arduino 监控电流传感器的模拟输出电压,如果电流传感器的读数超出某一电平,可以告诉 Arduino 停止发送电机命令——这称之为“过流”保护,并且能使电机控制器避免自损坏。

2.4 总结

本章简明扼要地讨论了本书中使用的各种连接和控制方法,以及一些不同类型的传感器。

继电器可靠,并且连接容易,但是会产生逆电动势,必须使用保护二极管来从 Arduino 直接驱动。晶体管可以以多种形式提供使用,并且能够通过 Arduino 直接被驱动,但是必须作为高侧开关或者低侧开关专门摆放。ESC 适用于那些不想创建自己电路的人,但是其购买成本通常高于使用晶体管和继电器创建一个类似的电机控制器。

然后我们讨论了各种使用有线和无线连接的控制方法,包括红外线控制器和无线电控制。无线电控制使用各种不同方法来实现,其中最常用的就是 2.4GHz。我们使用 2.4GHz 玩具型无线电控制系统和 Xbee 无线串口链路来同时控制和监控机器人的运行。无线电设备控制用于本书中的探险者机器人、草地机器人和战斗机器人等项目。

最后,探讨了一些用于制作自主(自导)机器人的传感器类型。我们将接触式开关用于昆虫机器人,红外线探测器用于莱纳斯寻线机器人,超声波传感器用于墙追踪机器人 Wally,以及 GPS 用于机器船,用来引导这四种机器人沿着路线前进,而不需要任何用户输入。传感器也可以用来协助人们进行更容易或者改良的控制,如探险者机器人或赛格威机器人那样。

在第 3 章中,将讨论不同类型的电动装置、电池和轮子,这些东西在机器人项目中使用很普遍。



创新

创意

创造



Arduino UNO Rev3



HOT

Arduino Ethernet PoE



NEW

Arduino MEGA ADK



NEW

Arduino DUE



NEW

Arduino Yún



NEW

L6470 Stepper Driver



NEW

RoboClaw大功率驱动板



NEW

Serial Controlled Motor Driver



NEW

9 Degrees of Freedom



NEW

Big Easy Driver



NEW

CarDuino UNO SMD



HOT

CarDuino Leonardo



HOT

32路伺服舵机控制器



HOT

XBee继电器扩展板



NEW

双H桥直流电机驱动板



HOT

Raspberry Pi原型扩展板



HOT

Raspberry Pi直流电机驱动板



NEW

远距离无线视频传输套件



NEW

PicoBoard互动创新板



NEW

RB-330MG大扭矩伺服舵机



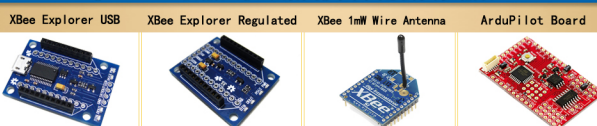
NEW

基于 Arduino 互动电子套件



HOT

《Arduino 机器人权威指南》图书配套实验器件



爱上 Raspberry Pi 入门套件



NEW

爱上 Arduino 互动入门套件



HOT

爱上 Processing 互动入门套件



HOT

基于 Arduino 互动电子积木套件



NEW

Arduino 与 Android 互动创客套件



HOT

基于 Arduino 传感器扩展板 金属万向轮 专用微型摄像头 7寸高清液晶显示器



SHARP 红外测距传感器 EM-406A GPS 模块 FT232 USB-TTL 模块 Mini 红外寻线传感器



长触型碰撞传感器 FS-C16B 无线遥控器 旋转角度电位计 按压式大按钮模块



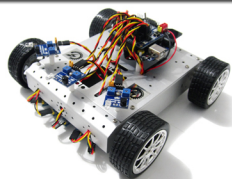
N20 直流减速电机 RB-65PG 伺服舵机 二自由度舵机云台 AS-RP5 履带机器人





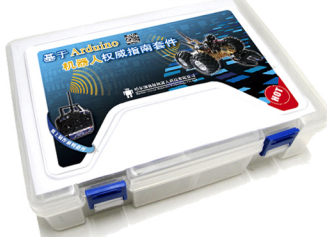
提供最新机器人资讯

AS-4WD寻线避障机器人



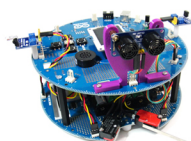
HOT

基于Arduino机器人权威指南套件



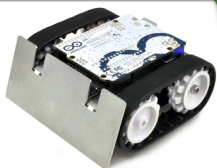
NEW

Arduino Robot机器人



NEW

基于Arduino相扑对抗机器人



NEW

m3pi Robot轮式机器人



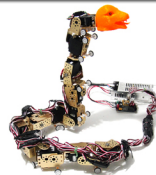
NEW

仿蝎式越野移动机器人



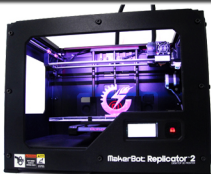
NEW

仿生机械蛇



HOT

MakerBot Replicator 2 3D 打印机



HOT



探索者机器人
Explorer-Bot



昆虫机器人
Bug-Bot



赛格威机器人
The Seg-Bot



水下机器人
RoboBoat



草地机器人
Lawn-Bot

AS-6D0F 双足竞步机器人



NEW

祝贺哈尔滨奥松机器人科技有限公司正式成为美国机器人零件巨头Pololu官方授权中国总代理

公司官网: www.robotbase.cn

公司电话: 18945688768、18946150895

公司邮箱: robotbase@aliyun.com

奥松机器人微信公众帐号: alsrobotbase

淘宝店网址: <http://robotbase.taobao.com>

树莓派专题网站: www.iraspberrypi.com

奥松机器人基地新浪微博: <http://weibo.com/robotbase>

机器人技术博客: <http://blog.sina.com.cn/robotbase>

Arduino教学博客: <http://blog.sina.com.cn/arduino>

优酷视频播客: <http://u.youku.com/robotbase>

资源

更多产品详情请登录网站或致电咨询



企业微信公众平台

Arduino带动的创客运动把互动电子带给了大众，让任何有兴趣的人可以快速入门，制作有趣的电子作品。机器人是在互动电子里面大家最感兴趣的一个主题。

《Arduino机器人权威指南》以每个章节一个机器人制作实例的方式带领读者进入有趣的机器人领域，详细的叙述让初学者可以按照一步一步的指示搭建自己的机器人。这是一本对机器人有兴趣的朋友不可或缺的好书。

李大维 上海新车间创始人

《Arduino机器人权威指南》一书通过一个个具体的实际机器人制作实例向大家展现了机器人制作过程中的种种奥妙。万事开头难，机器人的制作需要涉及电子学、计算机视觉、机器学、机电控制、光学等诸多领域的众多知识，这往往使希望开始制作机器人的初学爱好者觉得无从下手。而这本书通过介绍具体的机器人制作实例，可以带领读者在机器人制作中一步步深入，书中同时也不乏相关背景知识的介绍。本书的中译版在于欣龙先生的努力下问世了，它是广大机器人爱好者不可多得的参考书籍。

陈士凯 上海RoboPeak创始人

在机器人创新活动中，学生的视野会决定他们的未来，古人云：跋而望，不如登高之博见。只有了解世界同行在科研和教学领域中的成就，我们所教的内容才会融入世界发展的潮流之中，才能让学生接触世界最好的教育，培养他们具有国际的视野，这样我们的未来才会有希望。

Arduino的出现为广大的电子爱好者展示了一个广阔的天空，受到众多创客的追捧，如何将Arduino和机器人制作相结合，全面讲解Arduino在机器人上的应用，目前国内尚无这一方面的著作。在此我们应该感谢奥松机器人创始人于欣龙先生，他将*Arduino Robotics*介绍到中国，让中国的读者可以接触机器人创新的前沿，这是对中国机器人发展的重要贡献。

郑剑春 教育部“国培计划”专家、青少年科技创新丛书主编

你想制作有趣的机器人吗？你想让机器人沿着指定的路线行走吗？你想让机器人帮助你清扫庭院吗？你想让机器人载着你逛街吗？那么请选择《Arduino机器人权威指南》吧！John-David Warren、Josh Adams 和 Harald Molle 会带你走进机器人的世界，教会你如何寻找配件、怎样制作机器人、怎样进行编程，甚至告诉你如何对机器人进行安全测试。

这本书可以教会你如何用Arduino来控制各式各样的机器人，同时提供了每一步的详细指导。你不仅可以学会Arduino的基本使用方法，还可以了解各种电机的特性。同样，你会掌握其控制和排除故障的方法，将之应用到你的机器人项目中。本书从易到难，讲解了各种各样机器人的制作方法，其中包括GPS机器船、草地机器人、格斗机器人及赛格威机器人。

- ◆ 电机的类型及控制方法
- ◆ 自制传感器指导
- ◆ PCB的设计及制作
- ◆ 不同材料机器人的框架制作
- ◆ R/C 控制及解码
- ◆ 各种机器人的设计指导

无论你是只会摆弄Arduino的初学者，还是一个制作小工具的专家，《Arduino机器人权威指南》都会帮助你制作出意想不到的机器人作品。

Apress[®]
www.apress.com

支持单位： **奥松机器人**[®]
RobotBase
Robotics & Electronics



策划编辑：林瑞和
责任编辑：李云静
封面设计：吴海燕

上架建议：Arduino、机器人

ISBN 978-7-121-22291-7



9 787121 222917 >

定价：99.00元