



WILEY

Arduino编程

实现梦想的工具和技术

Arduino Sketches Tools and Techniques
for Programming Wizardry

快速精通Arduino编程的实战指南
深入透彻讲解Arduino的编程语言
快速掌握编程技能与作品实践操作

好“吃”好玩的
创客“水果书”



[法] 詹姆斯 A. 兰布里奇 (James A. Langbridge) 著
黄峰达 王小兵 陈福 译

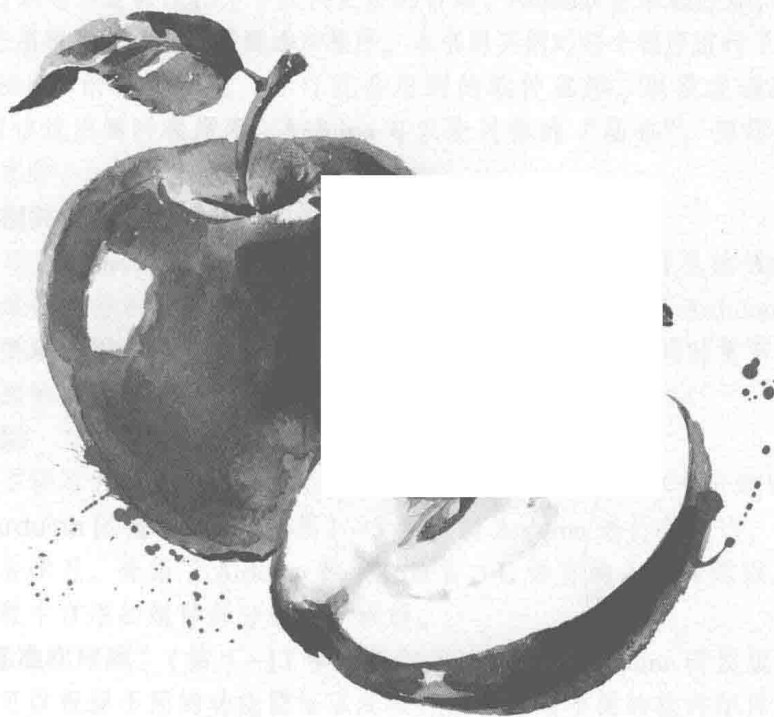
 机械工业出版社
CHINA MACHINE PRESS



WILEY

Arduino编程

实现梦想的工具和技术



[法] 詹姆斯 A. 兰布里奇 (James A. Langbridge) 著
黄峰达 王小兵 陈福 译



机械工业出版社
CHINA MACHINE PRESS

Copyright© 2015 by John Wiley & Sons, Inc.

All Rights Reserved. This translation published under license. Authorized translation from the English language edition, entitled Arduino Sketches: Tools and Techniques for Programming Wizardry, ISBN: 978-1-118-91960-6, by James A. Langbridge, Published by John Wiley & Sons, Inc. No part of this book may be reproduced in any form without the written permission of the original copyrights holder.

本书中文简体字版由 Wiley 授权机械工业出版社独家出版, 未经出版者书面允许, 本书的任何部分不得以任何方式复制或抄袭。

版权所有, 翻印必究。

北京市版权局著作权合同登记 图字: 01-2015-1283 号。

图书在版编目 (CIP) 数据

Arduino 编程: 实现梦想的工具和技术/ (法) 詹姆斯·A. 兰布里奇 (James A. Langbridge) 著; 黄峰达, 王小兵, 陈福译. —北京: 机械工业出版社, 2017. 8

(创客+)

书名原文: Arduino Sketches: Tools and Techniques for Programming Wizardry
ISBN 978-7-111-57482-8

I. ①A… II. ①詹… ②黄… ③王… ④陈… III. ①单片微型计算机—程序设计 IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2017) 第 175063 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 林 桢 责任编辑: 闫洪庆

责任校对: 张晓蓉 封面设计: 鞠 杨

责任印制: 常天培

北京京丰印刷厂印刷

2017 年 9 月第 1 版第 1 次印刷

184mm × 240mm · 20.5 印张 · 469 千字

标准书号: ISBN 978-7-111-57482-8

定价: 89.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

服务咨询热线: 010-88361066

机工官网: www.cmpbook.com

读者购书热线: 010-68326294

机工官博: weibo.com/cmp1952

010-88379203

金书网: www.golden-book.com

封面无防伪标均为盗版

教育服务网: www.cmpedu.com

本书涵盖了学习 Arduino 所需的各方面知识。介绍了目前市场上常见的 Arduino 开发板，解释了如何下载并运行软件以及如何编程；最关键的是，本书解析了 Arduino 的编程语言，以及在根据设计需要添加程序库之后我们可以获得哪些额外的功能。同时贯穿整本书的大量实例对电子电路方面的知识也进行了入门级的讲解。

通过阅读本书，Arduino 可以变成你的“画布”，而你所编写的程序可以成就你的数字杰作。本书主要适用对象为创客，他们知道能够如何创造出令人惊讶的应用程序，如何是日常任务智能化。当然本书也同样适用于对 Arduino 编程感兴趣的开发者。

原书前言

Arduino 为我们开辟了一个新世界，无论是业余爱好者还是专业人员都能够利用 Arduino 系统创造出一些精彩复杂的设备来帮助他们完成一个个引人入胜的小玩意。按下按钮点亮一盏灯，这样的简单设备可以驱动 3D 打印，你可以把 Arduino 应用于很多设备中。

要开启这一切，Arduino 需要使用一款编程软件来进行设计以帮助你完成程序。它们在与外界通信时会注意项目本身的逻辑性。为了提供更多的帮助，Arduino 自带程序库，可以根据需求安装软件，也根据应用程序以及硬件需要添加程序。本书用实例对每个程序进行了详解。

本书介绍了 Arduino 的编程环境，你可能会用到的软件程序，以及在遇到各种不同的 Arduino 开发板时应该选用哪种程序库。Arduino 可以变成你的“画布”，而你所编写的程序可以成就你的数字杰作。

关于本书以及相关技术的概述

本书涵盖了学习 Arduino 所需的各方面知识。介绍了目前市场上常见的 Arduino 开发板，解释了如何下载并运行软件以及如何编程；最关键的是，本书解析了 Arduino 的编程语言，以及在根据设计需要添加程序库之后我们可以获得哪些额外的功能。同时贯穿整本书的大量实例对电子电路方面的知识也进行了入门级的讲解。

本书的结构框架

本书为帮助想要学习 Arduino 编程的人提供了尽可能多的信息，本书分成四个部分。

第 1 部分，“Arduino 的基本知识”（第 1~3 章）对 Arduino 进行了概述，包括它们的来源以及为什么需要去学习。介绍了 Arduino 的编程语言、C 语言的入门知识以及电子方面的基本知识，也对编程中常用的组件部分进行了讲解。

第 2 部分，“标准程序库”（第 4~17 章）详细阐述了每种 Arduino 开发板可能用到的程序库，也就是说你可以根据不同的功能需要以及硬件支持选用不同的软件组件。每个程序库按章呈现，其使用方法会结合具体的实例进行解释，有助于理解其用途。

第 3 部分，“特殊设备程序库”（第 18~23 章）主要介绍特殊类型的 Arduino 开发板所用的程序库，可以将其加载到新的软件中与硬件设备连接或者完成指定任务。此外，每一个库中程序的使用都提供了相应的示例。

第 4 部分，“用户程序库和扩展板”（第 24~26 章），这一部分是关于 Arduino 的进一步研究，说明了如何导入用户的程序库以及如何设计、分享自己的程序。介绍了如何将一个电路板加到你的板子中以增加一些功能，并形成自己的扩展板。

IV Arduino 编程：实现梦想的工具和技术

谁应该读本书

本书主要适用对象为创客，他们知道如何能够创造出令人惊讶的应用程序，如何使日常任务智能化。当然本书也同样适用于对 Arduino 编程感兴趣的开发者。

需要的工具

每一章都有示例，每一章的开头部分都准确地列出了学习该章所需要的工具。根据本书中的示例介绍，我们需要以下的硬件：

- 计算机
- USB 数据线以及微型 USB 数据线
- 5V 电源
- 面包板和与其相适应的导线
- 几种 Arduino 开发板：两个 Arduino Uno, Arduino Due、Arduino Mega 2560、Arduino Esplora、Arduino Robot、Arduino 各一个
- SainSmart LCD 扩展板
- SainSmart 以太网扩展板
- LM35 温度传感器
- SD 存储卡
- Adafruit ST7735 TFT 接口板
- Adafruit MAX31855 接口板
- K 型热电偶丝
- Adafruit 的 SI1145 UV 传感器板
- SainSmart 的 Wi-Fi 无线扩展板
- DHT11 湿度传感器
- HC-SR04 超声波测距传感器
- HYX-S0009 或者伺服电动机
- L293D
- 5V 双极步进电动机
- 红绿蓝 LED 灯
- 10k Ω 电阻
- 4.7k Ω 电阻

关于网站

书中的案例的源代码都可以从 www.wiley.com/go/arduinosketches 网站下载。

总结

Arduino 开发是一个有趣的话题，它为人们打开一个拥有无数可能的新世界。Arduino 不仅非常适合嵌入式开发的学习，也适用于将日常生活智能化，或者是用它做出一些夺人眼球的小玩意。在本书中你会看到许多关于如何制作一个简单设备的示例讲解，详细地从硬件原理图开始直到你能够组装设备并运行程序。

对每个程序的介绍以及对不同功能的解释能够帮助你获得最想要的运行效果。每个程序库中都提供了相应的示例，逐行显示的代码帮助你理解整个程序在做什么。希望在你启动新项目的时候，本书能够为你提供一些参考，帮助你玩得开心！

目 录

原书前言

第 1 部分 Arduino 的基本知识

第 1 章 Arduino 的简介 2

1.1 Atmel AVR	3
1.2 Arduino 项目	4
1.3 ATmega	5
1.3.1 ATmega 系列	5
1.3.2 ATtiny 系列	6
1.3.3 其他系列	6
1.4 不同的 Arduino	6
1.4.1 Arduino Uno	7
1.4.2 Arduino Leonardo	7
1.4.3 Arduino Ethernet	8
1.4.4 Arduino Mega 2560	8
1.4.5 Arduino Mini	9
1.4.6 Arduino Micro	9
1.4.7 Arduino Due	9
1.4.8 LilyPad Arduino	10
1.4.9 Arduino Pro	10
1.4.10 Arduino Robot	11
1.4.11 Arduino Esplora	11
1.4.12 Arduino Yún	12
1.4.13 Arduino Tre	12
1.4.14 Arduino Zero	12
1.4.15 你自己的 Arduino	13
1.5 扩展板	13
1.5.1 什么是扩展板	13
1.5.2 不同的扩展板	14
1.5.3 Arduino 电动机扩展板	14

1.5.4 Arduino 无线 SD 扩展板	14
1.5.5 Arduino 以太网扩展板	14
1.5.6 Arduino WiFi 扩展板	14
1.5.7 Arduino GSM 扩展板	14
1.5.8 你自己的扩展板	15
1.6 你能用 Arduino 做什么	15
1.7 你需要为本书准备什么	15
1.8 小结	16

第 2 章 Arduino 编程 17

2.1 安装环境	17
2.1.1 下载软件	18
2.1.2 运行软件	19
2.1.3 使用自己的 IDE	20
2.2 你的第一个程序	20
2.3 理解你的程序	24
2.4 编程基础	26
2.4.1 变量和数据类型	26
2.4.2 控制结构	27
2.4.3 函数	30
2.4.4 库	31
2.5 小结	31

第 3 章 电子基础 32

3.1 电子入门	32
3.2 电压、电流和电阻	33
3.2.1 电压	33
3.2.2 电流	34
3.2.3 电阻	34

3.3 基本元器件	35	3.5 输入和输出	40
3.3.1 电阻	35	3.6 连接发光二极管	40
3.3.2 电容	37	3.6.1 计算	40
3.3.3 二极管	37	3.6.2 软件	41
3.3.4 发光二极管	38	3.6.3 硬件	41
3.3.5 晶体管	38	3.6.4 接下来呢	42
3.4 面包板	39	3.7 小结	42

第2部分 标准程序库

第4章 Arduino 语言	44	第5章 串口通信	55
4.1 I/O 函数	44	5.1 串口通信的简介	55
4.1.1 数字 I/O	44	5.2 UART 通信	56
4.1.2 模拟 I/O	45	5.2.1 波特率	57
4.1.3 生成音调	46	5.2.2 数据位	58
4.1.4 读取脉冲	47	5.2.3 奇偶校验位	58
4.2 时间函数	47	5.2.4 停止位	58
4.2.1 delay()	48	5.3 调试和输出	58
4.2.2 delayMicroseconds()	48	5.4 启动一个串行连接	59
4.2.3 millis()	48	5.5 写数据	60
4.2.4 micros()	48	5.5.1 发送文本	60
4.3 数学函数	49	5.5.2 发送数据	62
4.3.1 min()	49	5.6 读数据	62
4.3.2 max()	49	5.6.1 开始通信	62
4.3.3 constrain()	49	5.6.2 是否阻塞	63
4.3.4 abs()	50	5.6.3 读取一个字节	63
4.3.5 map()	50	5.6.4 读取多个字节	63
4.3.6 pow()	50	5.6.5 数据预览	64
4.3.7 sqrt()	51	5.6.6 解析数据	64
4.3.8 random()	51	5.6.7 清除	65
4.4 三角函数	51	5.7 示例编程	65
4.4.1 sin()	52	5.8 软件串口	68
4.4.2 cos()	52	5.9 小结	69
4.4.3 tan()	52	第6章 EEPROM	70
4.4.4 常量	52	6.1 EEPROM 的简介	70
4.5 中断	52	6.2 Arduino 上的不同存储器	71
4.5.1 attachInterrupt()	53	6.3 EEPROM 库	72
4.5.2 detachInterrupt()	53	6.3.1 读取和写入字节	72
4.5.3 noInterrupt()	54	6.3.2 读取和写入位	73
4.5.4 interrupts()	54	6.3.3 读取和写入字符串	75
4.6 小结	54	6.3.4 读取和写入其他值	75

6.3.5 示例程序	76	9.3.2 IP 地址	108
6.4 准备 EEPROM 存储	80	9.3.3 DNS	109
6.5 扩展非易失性存储器	80	9.3.4 端口	109
6.6 小结	81	9.4 Arduino 上的以太网	109
第 7 章 SPI	82	9.4.1 导入 Ethernet 库	110
7.1 SPI 的简介	82	9.4.2 开始构建以太网	110
7.2 SPI 总线	83	9.5 Arduino 作为客户端	112
7.2.1 与 RS-232 比较	83	9.5.1 发送和接收数据	113
7.2.2 配置	83	9.5.2 示例程序	115
7.2.3 通信	84	9.5.3 Arduino 作为服务器	117
7.3 Arduino SPI	84	9.5.4 示例程序	118
7.4 SPI 库	85	9.6 小结	120
7.5 Arduino Due 上的 SPI	87	第 10 章 WiFi	121
7.6 示例程序	88	10.1 引言	121
7.6.1 硬件	89	10.2 WiFi 协议	122
7.6.2 程序	90	10.2.1 拓扑	122
7.6.3 练习	93	10.2.2 网络参数	123
7.7 小结	93	10.3 Arduino WiFi	124
第 8 章 Wire	94	10.3.1 导入库	124
8.1 Wire 的简介	94	10.3.2 初始化	125
8.2 连接 I ² C	95	10.3.3 状态	125
8.3 I ² C 协议	95	10.3.4 扫描网络	126
8.3.1 地址	95	10.3.5 连接和配置	127
8.3.2 通信	96	10.3.6 无线客户端	128
8.4 进行通信	97	10.3.7 无线服务器	128
8.4.1 主机通信	98	10.4 示例应用	129
8.4.2 从机通信	99	10.4.1 硬件	130
8.5 陷阱和缺陷	104	10.4.2 程序	130
8.5.1 不同电压	104	10.4.3 练习	136
8.5.2 总线速率	105	10.5 小结	137
8.5.3 I ² C 扩展板	105	第 11 章 LiquidCrystal	138
8.6 小结	105	11.1 引言	138
第 9 章 Ethernet	106	11.2 LiquidCrystal 库	140
9.1 以太网的简介	106	11.2.1 写入文本	141
9.2 以太网	107	11.2.2 光标命令	141
9.2.1 以太网电缆	107	11.2.3 文本方向	142
9.2.2 交换机和集线器	107	11.2.4 滚动	142
9.2.3 以太网供电	108	11.2.5 自定义文本	143
9.3 TCP/IP	108	11.3 示例程序	144
9.3.1 MAC 地址	108	11.3.1 硬件	145

VII Arduino 编程：实现梦想的工具和技术

11.3.2 软件	145	14.2.2 转动伺服电动机	176
11.3.3 练习	148	14.2.3 断开	177
11.4 小结	148	14.2.4 准确性与安全性	177
第 12 章 SD	149	14.3 示例程序	177
12.1 引言	149	14.3.1 电路图	179
12.2 SD 卡	151	14.3.2 程序	179
12.2.1 容量	152	14.3.3 练习	180
12.2.2 速率	153	14.4 小结	181
12.3 在 Arduino 中使用 SD 卡	153	第 15 章 Stepper	182
12.3.1 公认的 SD 卡	153	15.1 电动机的简介	182
12.3.2 限制	154	15.2 控制步进电动机	183
12.4 SD 库	154	15.2.1 硬件	183
12.4.1 导入库	154	15.2.2 单极性与双极性步进电动机	183
12.4.2 连接 SD 卡	154	15.3 Stepper 库	184
12.4.3 打开和关闭文件	155	15.4 示例项目	184
12.4.4 读取和写入文件	155	15.4.1 硬件	185
12.4.5 文件操作	157	15.4.2 程序	185
12.4.6 SD 卡操作	157	15.5 小结	186
12.4.7 高级用法	158	第 16 章 Firmata	187
12.5 示例程序	158	16.1 Firmata 的简介	187
12.6 小结	161	16.2 Firmata 库	188
第 13 章 TFT	162	16.2.1 发送消息	188
13.1 引言	162	16.2.2 接收消息	189
13.2 技术	163	16.2.3 回调	189
13.3 TFT 库	164	16.2.4 SysEx	191
13.3.1 初始化	164	16.3 示例程序	192
13.3.2 屏幕准备	165	16.4 小结	194
13.3.3 文本操作	165	第 17 章 GSM	195
13.3.4 基本图形	166	17.1 全球移动通信系统 (GSM) 的简介	195
13.3.5 上色	167	17.2 移动数据网络	196
13.3.6 图形图像	167	17.2.1 GSM	196
13.4 示例程序	168	17.2.2 3G	197
13.4.1 硬件	168	17.2.3 4G 和未来	197
13.4.2 程序	169	17.2.4 调制解调器	197
13.4.3 练习	173	17.3 Arduino 和 GSM	198
13.5 小结	173	17.4 Arduino GSM 库	199
第 14 章 Servo	174	17.4.1 GSM 类	200
14.1 伺服电动机的简介	174	17.4.2 SMS 类	201
14.2 控制伺服电动机	175	17.4.3 VoiceCall 类	202
14.2.1 连接伺服电动机	175		

17.4.4 GPRS	204	17.5 应用示例	205
17.4.5 Modem	205	17.6 小结	209

第3部分 特殊设备程序库

第18章 Audio	211	20.5.2 鼠标	241
18.1 音频的简介	211	20.6 示例程序	242
18.2 数字语音文件	212	20.6.1 硬件	243
18.3 Arduino 上的音乐	213	20.6.2 源代码	243
18.4 Arduino Due	213	20.7 小结	245
18.4.1 数-模转换器	214	第21章 Esplora	246
18.4.2 数字音频到模拟音频	214	21.1 Esplora 的简介	246
18.4.3 创建数字音频	214	21.2 Arduino Esplora 库	247
18.4.4 存储数字音频	215	21.2.1 RGB LED	247
18.4.5 播放数字音频	215	21.2.2 传感器	248
18.5 示例程序	216	21.2.3 按键	249
18.5.1 硬件	217	21.2.4 蜂鸣器	250
18.5.2 程序	218	21.2.5 TinkerKit	250
18.5.3 练习	221	21.2.6 LCD 模块	251
18.6 小结	221	21.3 示例程序和练习	251
第19章 Scheduler	222	21.4 小结	253
19.1 调度器的简介	222	第22章 Robot	254
19.2 Arduino 多任务	224	22.1 Robot 库的简介	255
19.3 调度	224	22.2 Arduino 机器人	256
19.3.1 多任务合作	225	22.3 Robot 库	257
19.3.2 非合作函数	227	22.3.1 控制板	257
19.4 示例程序	228	22.3.2 电动机板	263
19.4.1 硬件	229	22.4 示例程序和练习	264
19.4.2 程序	231	22.5 小结	266
19.4.3 练习	233	第23章 Bridge	267
19.5 小结	234	23.1 Bridge 库的简介	267
第20章 USBHost	235	23.2 Bridge	268
20.1 USBHost 的简介	235	23.2.1 Process	269
20.2 USB 协议	236	23.2.2 FileIO	271
20.3 USB 设备	237	23.2.3 YunServer	272
20.3.1 键盘	237	23.2.4 YunClient	273
20.3.2 鼠标	238	23.3 示例应用程序	273
20.3.3 集线器	238	23.3.1 硬件	273
20.4 Arduino Due	238	23.3.2 程序	273
20.5 USBHost 库	239	23.3.3 练习	277
20.5.1 键盘	240	23.4 小结	277

第4部分 用户程序库和扩展板

第24章 导入第三方库	279	第26章 创建你自己的库	298
24.1 库	279	26.1 库	298
24.1.1 查找库	280	26.1.1 库基础	298
24.1.2 导入一个库	280	26.1.2 初级库	298
24.1.3 使用扩展库	282	26.1.3 高级库	302
24.2 示例应用程序	284	26.1.4 添加注释	305
24.3 练习	288	26.1.5 添加示例	306
24.4 小结	288	26.1.6 须知	306
第25章 创建你自己的扩展板	289	26.1.7 编码风格	307
25.1 创建一个扩展板	289	26.1.8 发布你的库	308
25.1.1 想法	289	26.1.9 闭源库	308
25.1.2 必备硬件	290	26.2 示例库	308
25.1.3 必备软件	290	26.2.1 库	308
25.2 你的第一个扩展板	291	26.2.2 示例	314
25.2.1 第1步：面包板	292	26.2.3 须知	316
25.2.2 第2步：原理图	294	26.2.4 收尾	317
25.2.3 第3步：PCB	296	26.3 小结	317
25.3 小结	297		

第1部分

Arduino的基本知识

第1章 Arduino 的简介

第2章 Arduino 编程

第3章 电子基础

第1章

Arduino的简介

电子发烧友存在已经很久了。几十年来，他们一直在创造有趣的设备和产品。业余无线电爱好者通过模仿杂志或自己设计简单的电路图来制作自己的收音机。我们中有多少人也是通过构建一个无线电系统来探索电子学，直到上瘾？只需要很少钱的元器件，你就可以创建自己的收音机，并且通过一个小的低品质扬声器来收听长波信号，它可能比在商店中买的要更好一些，因为它是自制的。如果你想要有更棒的声音，你可以买一个更好的扬声器。如果想要更大的音量，工具包中也有为这些准备的材料。发烧友按自己的需要制作自己的放大器和配件。也有为各个层级提供的相应书籍，从初学者到专家。工具包也能用于从最小元件到整个计算机系统。可以毫不夸张地说，你可以走进一家电子商店，购买 DIY 计算机，花费几个小时焊接内存、芯片到印制电路板上。我就是这样开始的。

在 20 世纪 90 年代，情况稍微有些变化。大多数爱好者的办公桌上都有自己的计算机，而且可以用它们来创建电路图，模拟系统的一部分，甚至用透明布局来印制电路板，让整个过程更加容易。然而，某些东西缺失了。几乎所有能制作出来的设备都是不可编程的。虽然可以使用微处理器，但是它们要么太昂贵，要么太复杂。处理器本身是无用的，它必须连接外部存储器。为了在每次启动的时候都能运行一个程序，它也必须有一个只读存储器。如果你想要中断，你也需要在设计中添加相应的芯片。最终它的复杂度超出一些爱好者的能力范围。为了做到这些但是又不这么复杂，爱好者倾向于使用已经在他们办公桌上的编程设备：一台个人计算机（PC）。

大多数 PC 在当时使用的是 ISA 总线，如图 1-1 所示。ISA 是一个简单的总线，允许组件添加到处理器和通用计算机系统中。这是个允许用户往他们的计算机中插入扩展卡的简单系统，要使用它也很容易。创建一个可以插入到 ISA 插槽的电路板不是很难，并且已经存在完成的原型板，爱好者和工程师可以在印制他们的电路板之前测试解决方案。其中的一些电路板甚至包含试验板，一个简单的系统不需要焊接就可以让用户把他们的部件和导线放进去。这引起了一场小革命，众多发烧友用这种类型的电路板来做以前不能做的事情：创建可编程系统。一个 ISA 板可以有数字输入和输出、模拟输入和输出、无线电、通信设备——什么都是有可能的。所有的这些可以用计算机来控制，使用简单的编程语言，如 C 或者 Pascal。我的 ISA 卡一直在通过读取温度计中的数据，开启电加热器，像一个恒温器一样保持着我的学生公寓的

温暖；它还能充当闹钟，而这取决于我第二天的课程安排。虽然有时我不想上一些一大早的课，平心而论这通常来说是我的错；ISA 卡在预算紧张的时候，是很完美的选择。

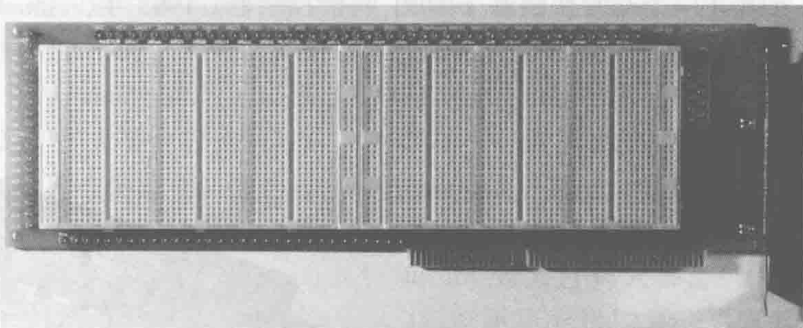


图 1-1 ISA 原型板

计算机变得越来越快，系统也一直在演变。行业变了，所以有了扩展端口。当爱好者成为 ISA 总线专家的时候，业界发明了一种新的系统：VESA 局部总线（VLB）。VLB 是一种扩展 ISA，只需要为内存映射 I/O 和直接存储器存储（DMA）添加第二个连接器，但是它宣告了一个变化。计算机确实越来越快，而且有些计算机总线系统无法跟上。即使是 VLB 也无法跟上，仅仅过了一年，PCI 就变成了参考。PCI 总线是一种先进的总线，但是需要组件和逻辑来识别自己。它突然让创建自制板变得越来越困难。一些爱好者决定使用其他工业标准接口，诸如并行端口或者 RS-232，但大部分人都停止了创建这样的系统。这些人继续使用模拟系统或不可编程的数字系统。取代可编程序微控制器的系统是使用逻辑门设计的。例如，如果 A、B 输入都为真或者 C 输入为假时，灯就打开。但是当输入量增加时，这些事情就变得越来越复杂。

模拟系统，如收音机和放大器，都没有编程的形式。它们在“脑子”里已经被设计成一个特定的任务。配置是模拟的，用小螺丝刀（规范术语为螺钉旋具，本书采用习惯用语），设计者可以“微调”电位器值、可变电阻值。这是不可能通过一个特定的值乘以一个输入信号来为设备编程的。相反，在组件中加入电位器以对抗元器件公差的影响。因此，设计需要一个额外的阶段——校准。特定的输入信号输入到装置中，同时期望一个特定的输出值。

处理器是存在的，而且能被使用，有一些项目中会使用它们，但是在设计中集成一个处理器，通常意味着需要使用几个组件。存储器芯片、I/O 控制器，或者不得不使用的总线控制器，经过十年间的技术进步，电路也变得越来越复杂。即使当这些设计工作时，对它们进行编程也是一种挑战。大多数的编程是通过 EEPROM 器件（电可擦编程只读存储器）来完成的。它们可以存储一个计算机程序以及可以通过附加在计算机上的外部编程器进行编程。它们被称为可擦写只读存储器是因为它们存储的内容确实可以擦除和更换，但是要擦除上面的内容，需要从电路中移除并且用紫外光照射 20min。程序中的一个小错误可能经常需要 30min 或者更长时间来纠正。

1.1 Atmel AVR

Atmel 公司是一家成立于 1984 年的美国半导体公司，Atmel 是先进的存储器和逻辑技术

(Advanced Technology for Memory and Logic) 的英文缩写。从一开始, Atmel 公司就设计了比同类产品更低功耗的内存芯片, 但它很快就决定创建可编程器件。1994 年, Atmel 公司进入微处理器市场, 创造了一个非常快的基于 8051 的微控制器。1995 年, Atmel 公司是第一批 ARM 架构授权公司之一, 这让它获得先进的处理器技术。

Atmel 公司并不只是使用 ARM 技术, 在 1996 年它也创造了自己的处理器——AVR (见图 1-2)。AVR 代表了什么? 这也是 Atmel 公司的众多谜团之一。由 Alf-Egil Bogen 和 Vegard Wollan 设计, 也有人说它代表 Alf 和 Vegard 的 RISC 处理器。我们将永远不知道, 而且在当时, 人们并没有兴趣去了解名字, 人们更感兴趣的是开始接触这个先进的技术。

在这之前, 为只读存储器设备编程需要一些繁琐的步骤, 如使该芯片暴露于紫外线下, 或者其他复杂的擦除方法。随着 Atmel 8 位

AVR 的出现, 这一切都改变了。AVR 是第一个采用片上闪存用于程序存储的微控制器系列。它也直接在芯片上包含了随机存取存储器 (RAM), 基本上包含了在一个芯片上运行微控制器所需要的一切。突然间, 所有的复杂设计可以替换成单个部件。更妙的是, 为芯片编程可以在几分钟内使用最小的硬件来完成。Atmel 公司设计的一些微控制器可以直接允许用户将其插入 USB 端口, 并且采用 Atmel 公司的软件进行编程。从编译程序到执行可以在一分钟内完成。

有一些学习平台已经存在了, Parallax's BASIC Stamp 和 PIC 设备都在使用, 但是 Atmel AVR 的出现为电子爱好者提供了另一个选择。此前, 在数字系统中, 逻辑是在印制电路板之前定义的。输入和输出分别连接至逻辑门, 且功能已经被设计到产品中。现在, 随着 AVR 系列的出现, 爱好者和工程师们有一个新的可能性。取代了电子设计的功能, 系统可以用计算机编程与外界交互。这简化了电子产品; 替换了多个逻辑门的使用, 一切都直接连接到了微控制器, 这样就可能对与外界发生的事件做出反应进行编程。程序可以被编写以及再次编写, 设备可以被编程以及重新编程, 这为电子世界打开了一个新的大门。从理论上来说, 一个设备可以应对每一种可能发生的情况。该技术已经存在; 唯一剩下的事情就是有人来创建设备。

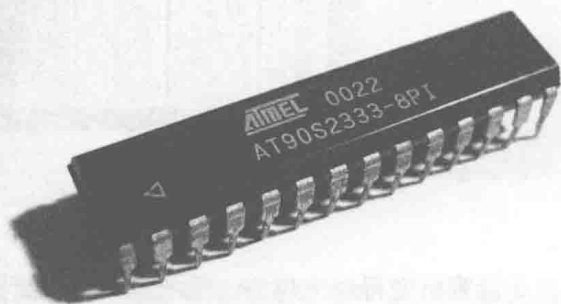


图 1-2 Atmel AVR 微处理器

1.2 Arduino 项目

Arduino 项目开始于 2005 年, 项目是为了位于意大利 Ivrea 的 Interaction Design Institute Ivrea 学院的学生而设计的。学生被告知要使用一个 BASIC Stamp, 它是一个小的 PBASIC (BASIC 编程语言的一个分支) 可程序微控制器, 但是这个设备的价格 (大概 75 美元), 对于学生来说过于昂贵, 不仅仅是购买价格偏贵, 而且替换损坏的单元也很昂贵。

Arduino 一开始就是为学生而设计的项目, 有针对性地去替换 BASIC Stamp。因为 Atmel

8 位 AVR 简单而且价格低,而且还有一个优点是需要的外部元器件少,所以便选择了这个芯片。它还有大量的输入和输出,这让它成为未来设计的完美选择。

学生和老师一起为这个新的设计而工作,其中一个是让 Atmel AVR 可以很容易地接受扩展卡。当完成原始版本时,研究人员尽力使它更轻、更便宜,更易于学生、爱好者和工程师们使用。第一个 Arduino 板诞生了。改进了 Arduino 的原始设计,如用 USB 代替 DB-9 串口连接器,这有助于扩大平台的吸引力。

每一个 Arduino 都有两方面。当然硬件只是 Arduino 项目的一部分。每个 Arduino 使用的 Atmel 微控制器都有特定的固件,每个设备都嵌有一个小程序来查找运行的程序,或者帮助使用串口来安装程序。

最后的设计以开源的形式发布,并且由 Arduino 设计和销售。Arduino 作为一个开源硬件项目发布是一个有趣的举动。因为它是开源的,它吸引了越来越多的用户寻找到这个项目。因为 Arduino 已经有不错的输入和输出设计,用户开始创造能添加到原始 Arduino 上的电路板。当 Arduino 设计了一个新的开发板时,它保留了原来的输入和输出设计,这样可以实现现有的插件可以在新的设计中使用。

刚开始的时候 Arduino 是为教育用途而设计的,随着在电子爱好者中名声的上涨,越来越多的经销商开始出售 Arduino 电路板。

Arduino 不仅创造了硬件——一个嵌入式硬件如果没有相应的软件也是很难使用的,而且也花费了很多时间在开发自己的语言和集成开发环境 (IDE)。最终的结果是一个很好的 IDE,可以运行在 Windows、Mac OS 和 Linux 操作系统上,并且可以转换 Arduino 语言 (C/C++ 的高级变体) 到 AVR 代码。

Arduino 开发环境隐藏了所有的链接到嵌入式系统和混合软件——如建立环境、连接器、讨厌的命令行的难题,并允许开发者在 Arduino 编程语言中使用简单的 C 语言函数来编程。

1.3 ATmega

Atmel 已经根据多种因素将它的 AVR 设计成不同的群体。它们有很多 AVR 微控制器,并知道对于项目来说哪些是必不可少的。有些 ATmega 设备有更多的内存,或者更多的数字与模拟的输入和输出,或者特定的封装尺寸。

1.3.1 ATmega 系列

Atmel megaAVR 是 AVR 系列中最强劲的部分。它们是为那些需要写大量代码的应用程序而设计的,并有着从 4KB 一直到 512KB 的闪存,这足以让最苛刻的方案运行。Atmel megaAVR 设备有多种尺寸,从 28 引脚封装到 100 引脚封装。这些设备具有的嵌入式系统数量惊人,仅举几例,模-数转换器、多个串行模式和看门狗定时器。它们也有大量的数字输入和输出线,使它们的设备非常适合与多种组件通信。