

Apress®

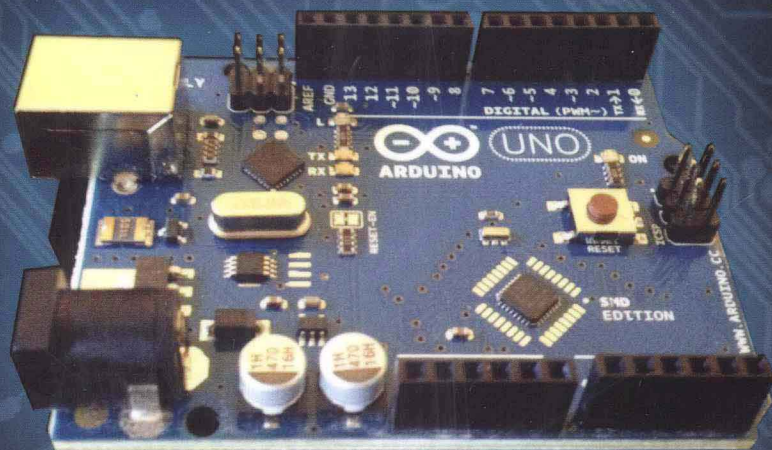
# Arduino

# C语言编程实战

## Beginning C for Arduino

### 用C语言的力量全面驾驭Arduino

【美】Jack Purdum 著  
麦秆创智 译



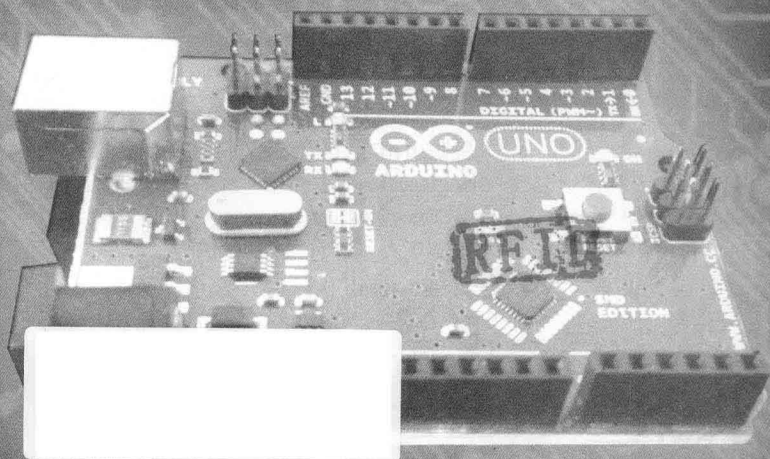
人民邮电出版社  
POSTS & TELECOM PRESS

# Arduino

## C语言编程实战

### Beginning C for Arduino

【美】Jack Purdum 著  
麦秆创智 译



人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

Arduino C语言编程实战 / (美) 普德姆  
(Purdum, J.) 著 ; 麦秆创智译. — 北京 : 人民邮电出  
版社, 2013. 10

ISBN 978-7-115-32933-2

I. ①A… II. ①普… ②麦… III. ①单片微型计算机  
—C语言—程序设计 IV. ①TP368.1②TP312

中国版本图书馆CIP数据核字(2013)第189962号

- 
- ◆ 著 [美] Jack Purdum
  - 译 麦秆创智
  - 责任编辑 刘 洋
  - 责任印制 彭志环 杨林杰
  
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京鑫正大印刷有限公司印刷
  
  - ◆ 开本: 800×1000 1/16  
印张: 15.25  
字数: 331 千字 2013 年 10 月第 1 版  
印数: 1—3 000 册 2013 年 10 月北京第 1 次印刷
- 
- 著作权合同登记号 图字: 01-2013-3880 号

定价: 49.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223  
反盗版热线: (010)67171154

# 版权声明

Beginning C for Arduino: Learn C Programming for the Arduino and Compatible Microcontrollers

By Jack Purdum, ISBN: 978-1-4302-4776-0

Original English language edition published by Apress Media.

Copyright © 2012 by Apress Media

Simplified Chinese-language edition copyright © 2013 by Post & Telecom Press

All rights reserved.

本书中文简体字版由 **Apress Media** 公司授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

## 内容提要

本书基于 **Arduino** 平台，介绍了 **Arduino** 编程所需的硬件、编程环境和 **Arduino** 上的 **C** 语言编程方法。从操作数和运算符开始，本书详细讲解了如何使用 **C** 语言构建程序语句，如何编写函数块和运行函数，如何引入函数原型，如何理解和使用指针，如何在程序中加入判断、循环和跳转，如何使用结构体数据类型，如何使用预处理指令，以及如何使用 **Arduino IDE** 附带的非默认库。附录中还介绍了如何选购 **Arduino** 编程硬件。

本书适合于对硬件编程感兴趣的技术人员阅读，尤其可作为 **C** 语言基础偏弱的读者的入门图书。

## 作者简介



杰克·普德姆 (Jack Purdum) 是普渡大学技术学院 (Purdue University's College of Technology) 的教授。普德姆博士在大学执教超过 25 年, 其间一共编写了 17 本程序开发和计算机相关的教材。如今, 他仍然孜孜不倦, 为杂志和期刊撰写文章, 为各种专业会议做演讲。他是 Ecosoft, Inc. 的创始人和 CEO。Ecosoft, Inc. 是一家专注于为 PC 研发编译器等程序开发工具的公司。普德姆博士持续地将精力投入到与面向对象程序开发分析和设计相关的线上培训和指导上, 他总结出了大量的教学经验和方法论 (例如: “右左右左法则”、“水桶比喻”、“程序设计五环节” 以及 Sideway Refinement 等), 还包括代码的评测机制 (Dhampstone) 等, 这些都成为了他多年教学生涯的硕果。他在穆斯静冈学院 (Muskingum College) 取得学士学位, 后在俄亥俄州立大学 (Ohio State University) 获得硕士和博士学位。

## 技术评审人简介



布拉德·利维 (Brad Levy) 是一位软件和硬件设计行业有超过 30 年经验的从业者。他开发的嵌入式系统广泛地用于能源管理、太阳能及航电测试设备。他为其参与创办的 Weather Metrics 公司开发了气象站及实时气象呈现软件。他拥有办公自动化、图像软件、系统库以及多平台下设备驱动程序的开发经验，同时掌握大型科学计算模拟程序的研发技术。布拉德从很早起就从事 C 语言的研发工作，甚至在堪萨斯大学 (University of Kansas) 时，还使用过 C 语言的前身：B 语言。

布拉德还熟悉 APL、Fortran、Pascal、C++、JavaScript、PHP、Python，以及多种处理器下的汇编语言。他还完成了编译器、连接器 (linker) 和运行时库的开发，并为一个嵌入式系统编写了多任务操作系统。他开发过各种方式进行的系统通信，包括直连通信、传统及蜂窝电话网络、卫星链路通信以及因特网。布拉德目前在从事与 Arduino 相关的工作，用于能源管理和铁路模型控制，以及“树莓派 (Raspberry Pi)”平台下的交互设计。

布拉德兴趣广泛，包括用户界面设计、图书馆学、未来信息和图书、Web 设计、摄影、图形设计、字体设计以及音频设备设计。他是美国计算机协会 (Association for Computing Machinery) 的长期会员。你可以访问他的网站：[www.bradlevy.com](http://www.bradlevy.com)。

## 译者序

我从事软件编程 10 余年，常常能够很深刻地感受到软件编程的局限性。大约这也就是为什么硬件编程在近年来变得越来越流行的原因。单片机是一个把“看不见摸不着”的软件通过“看得见摸得着”的硬件呈现在人们面前的工具，其重要性自无需我多言，从居家生活中的洗衣机、电冰箱，到谷歌研发的无人驾驶汽车，我们这个时代中处处都有单片机的身影。

在这之中，诞生了若干种开源机器人，利用硬件或软件的开源性，极大地方便了技术人员的技术交流和二次开发。Arduino 就是这样一个便捷灵活、方便易用的开源电子原型平台，在这个平台上，你可以自由地发展和实现很多自己的想法。拜其开源性所赐，世界上已经诞生了成千上万种优秀的应用和样例，无疑也为我们的科学研究和教学提供了极大的便利。Arduino 没有复杂的单片机基层代码，没有难懂的汇编，只有简单而实用的函数、丰富的接口、简单的编程环境、极大的自由度和可扩展性能，其高度标准化的接口为 Arduino 的持续发展奠定了坚实的基础。

本书从基于 Arduino 的 C 语言编程出发，讲解了 Arduino 开发过程中所需的硬件和软件基础。作者是一名极富教学经验的教授，在大多数人看来枯燥无味的语言基础能够被他讲解得生动有趣，每章结束后的思考习题很好地巩固了本章学到的知识。C 语言的指针历来是困扰 C 语言学习者的一大难题，本书关于指针部分的分析尤为细致且易于理解，并结合了许多具体的实例，相信能够为读者提供很好的帮助。在此过程中，也不难体会到作者独特的教学思路，不像高高的讲台上站着老师，而更像温和的与你面对面交流编程经验的朋友。学习编程语言的关键在于动手实践，书中提供了大量的实例可供读者用来学习与实践。

在本书的翻译过程中，如下人士亦有贡献：祝延彬、马煜坤、贾学来、陈勇、李科达、张皓博、王隼、罗一峰、褚彬池、王思敏、徐帆、张瑶等。在此一并致谢。

由于译者水平有限，匆忙之中难免疏漏，欢迎各位读者批评斧正。任何反馈内容可发送至责任编辑邮箱：[liuyang@ptpress.com.cn](mailto:liuyang@ptpress.com.cn)。



## 致 谢

没有人可以完全独立地完成一本书。很多人在并不知情的情况下为本书提供了大量的帮助，包括很多非常棒的教学案例，更好地阐述问题的方式，为各种正确或错误的观点提供反馈。很多不会编程的朋友，在我向他们讲解一个复杂的编程问题时为我的表达方式提供了很有用的建议。他们并不知道，在这个过程中他们帮助我打磨了我的教学方法。我必须要感谢他们，尤其是 Jane Holcer, Katie Mohr, John Purdum, Joe Kack, Bev Kack, John Strack 和 Mike Edwards。我想特别感谢我的技术编辑 Brad Levy，他为本书提供了大量细致入微的芯片和其他硬件的技术细节。同时，我还想感谢在本书附录 A 中提到的供应商能够参与到本书的编写过程中，他们在硬件和软件方面都为我提供了很多支持。除此之外，我还要感谢 Brigid Duffy, Michelle Lowman, Kim Burton，以及 Apress 出版社中所有在本书出版过程中帮助过我的人，在大家的共同努力下，本书才能够成为一本更好的书。

# 前言

我还记得第一次购买电子计算器时的情景。当时我在教授一门研究生的统计课程，必须要购买一个能够进行平方根运算的计算器。在 1960 年左右的年代，购买计算器是一个相当高级的需求。后来我在“教育优惠价”的活动中花 149.95 美元买到了！现在，我办公桌上放着一台 ATmega2560，体积比最初那台小一半，价格仅为它的 1/4，功能几乎已经完善到了无所不能的地步。我不禁惊讶，从烤面包机到汽车引擎，这一切到底是如何发生的？40 年前谁能想到现在的光景呢？

在进入微处理器领域之前，我的专业方向是软件工程，这个领域中有很多人都对微处理器技术感兴趣。我一直很喜欢电子产品，并且一直维护着我 50 年前获得的业余无线电执照。如果没有某种形式的软件参与，潜伏在 Atmel 系列微处理器中的计算潜能是无法被释放出来的。事实上，巧妙地结合硬件和软件这两个领域，你真的能看到一些非常令人兴奋的结果。

本书的主要目的是要教授你如何使用 C 语言编程。对于某些迷茫着双眼喃喃自语道“正是我想要的……一本 C 语言教程”的人，我要提醒你，这本书和其他 C 语言教材不一样。首先，很多教材似乎都喜欢把编程归为后台行为，转而注重微处理器开发过程这个方面。事实上，如果你读过一些关于单片机的书籍，可能会有这种感觉：软件，尤其是编程，是一个人们必须忍受的恶魔。也就是说，“真正的好东西”都在硬件中。然而，很棒的硬件上如果运行着马马虎虎的软件，那你就也只能得到一个马马虎虎的结果。制作优秀的软件所获得的回报一点儿也不比设计精良的硬件差。

另一个让本书与众不同的是我的教学经验。我有一名员工，他是我见过的最有天赋的程序员。在一个夏天，我为他指派了一名实习生，结果在不到一周的时间内，这名实习生哭着离职了，说这个人完全无法共事，更遑论从他那儿学到什么东西。你是一名很棒的程序员或工程师这一点完全不足以自动让你成为一名好的老师。直到有一天，当你站在讲台上，台下 150 双眼睛让你感觉自己就像聚光灯下的小鹿，这时你才会知道你那显而易见的想法是行不通的。超过 25 年不断的试错和纠正的过程帮助我培养出了大量的教学技巧，让我能够巧妙地指导学生从最有可能绊倒他们的绊脚石上跨越过去。

最后，教学并不是非得干巴巴，无聊又无趣。我尽可能地让这本书在阅读过程中就好像你和我在面对面交流编程经验一样。虽然你是最终的裁判，不过我希望你能够在这个过程中体会到编程的乐趣，就像我一样。编程可以成为最令人享受的消遣。

## 一些假设前提

首先，我要假设你不需要到下周就完全掌握 C 语言编程。这并不意味着你所需要的工具在本书中找不到。相反，我发现太多的学生在学习编程的过程中不花时间去真正敲键盘编写代码，自己尝试和调试程序。如果你采用的是这种办法，学习编程的过程实际上会被减慢。

## 2 前言

阅读一段叙述性文字很容易，你会对自己说：“对，我懂了。”然后继续进行下一个话题。只有当你把一段示例程序键入计算机，经过调试和测试，让它能够照你的设计正确地运行时，你才有资格被人拍着后背鞠躬谢幕。

其次，为了达到最佳的学习体验，你需要一台能够用于测试你的代码的硬件设备。我已经尽力精简了在学习本书的例程中所需要的设备（附录 B 中列出了所需的硬件，附录 A 中给出了一些推荐的供应商）。至少，你需要一个 Atmel 系列的微处理器电路板、一个面包板、几个 LED 灯和一些电线。这就够了。然而，还存在一种可能，你之所以会买下这本书，是因为你心中一开始就有一些想要去实现的想法。太棒了！买一些你所需要的硬件，然后开始通过一些程序片段来实现你整个设计中的某些小部分。带着目标去学习始终都是最好的。

最后，慢慢享受阅读的过程吧。停下来，设计一个你自己的程序，这远比我提供的任何一个例程更有用。你不能只是“阅读代码”——你必须实际参与进去，犯些错误，烧掉几个 LED，然后才能真正从你解决问题的过程中学到东西。每一章的最后，是你应该问自己的几个问题。我不是在拿它们凑字数。这些问题在那儿的原因在于让你思考这一章都学到了什么。在翻到书的最后看我的答案之前，花些时间自己想一想应该怎么回答。软件的魅力之一在于，对于同样的问题，解决的方式可以是多种多样的。

### 资源

还有很多地方可以为你提供额外的帮助，你很可能会用到它们。Arduino 拥有令人惊讶的健壮的环境，囊括了硬件和软件的各个方面。在互联网上搜索任何一个带有 Arduino 的关键词，搜索结果中总能为你提供一些有用的信息。Apress 出版过好几本非常优秀的硬件设计书籍，每一本都能很好地扩展你的知识基础。

附录 A 中给出了一些关于购买硬件组件的建议。如果你遇到了一个比较特别的问题，试着把你的问题发表在众多 Arduino 论坛之一上，互联网上有很多这样的论坛（例如 <http://arduino.cc/forum>）。你当地的高中物理教师也可能同样为你提供一些本地的资源。

最后，如果你确实在构建某个硬件设备或某个你觉得非常特殊的软件，通过这些论坛中的一个与别人共享它吧。Arduino 的 IDE 和所有与其相关的编程工具都是开源代码，这也就意味着，你所获得的所有帮助都是他人无偿提供给你的。如果你能为大家做出一些回报，那将是再好不过的事情。

好吧……先到这里。让我们开始挖掘 C 语言，让所有这一切旋转起来吧……

# 目 录

第1章 简介	1	2.4 小结	30
1.1 一些假设前提	2	2.5 练习	31
1.2 你需要什么	3	第3章 Arduino C数据类型	32
1.2.1 基于Atmel的单片机卡	3	3.1 boolean数据类型	33
1.2.2 面包板	5	3.2 char数据类型	34
1.2.3 零碎的小件	5	3.2.1 二进制数据	34
1.3 安装软件	6	3.2.2 char数据类型和字符集	35
1.4 检查硬件	7	3.2.3 生成ASCII字符表	36
1.4.1 连接USB线缆	7	3.3 byte数据类型	37
1.4.2 在集成开发环境中选择 $\mu$ C板	8	3.4 int数据类型	37
1.4.3 端口选择	9	3.5 word数据类型	38
1.5 载入并运行你的第一个程序	11	3.6 long数据类型	38
1.5.1 Blink程序	11	3.7 float和double数据类型	38
1.5.2 运行Blink程序代码	13	3.8 string数据类型	39
1.5.3 编译Blink程序代码	14	3.9 String数据类型	40
1.5.4 上传Blink程序	14	3.10 void数据类型	41
1.6 本章小结	16	3.11 array数据类型	42
第2章 Arduino C	17	3.12 变量的定义与声明	42
2.1 程序语言的通用构件	17	3.12.1 符号表	43
2.1.1 表达式	17	3.12.2 lvalue和rvalue	43
2.1.2 语句	19	3.12.3 “水桶”比喻	45
2.1.3 语句块	20	3.13 使用强制类型转换符	46
2.1.4 函数块	21	3.14 小结	47
2.2 程序的5个环节	22	3.15 练习	48
2.3 再来看看Blink程序	24	第4章 C语言的逻辑判断	49
2.3.1 程序注释	25	4.1 关系运算符	49
2.3.2 数据定义	26	4.2 if语句	50
2.3.3 setup()函数	28	4.3 升级版Blink程序	52
2.3.4 loop()函数	29		

## 2 目录

4.3.1 电路	52	5.5.2 continue语句	76
4.3.2 程序代码	54	5.6 一段完整的示例	77
4.4 升级版Blink程序的修改	55	5.6.1 环节1: 初始化	77
4.5 if-else表达式	56	5.6.2 环节2: 输入	77
4.6 if语句嵌套	57	5.6.3 环节3: 处理	77
4.7 自增和自减运算符	58	5.6.4 环节4: 输出	78
4.7.1 两种类型的自增运算符(++)	59	5.6.5 环节5: 结束	78
4.7.2 两种类型的自减运算符(--)	59	5.6.6 代码5-1是“傻瓜代码”	80
4.7.3 运算符的优先级	60	5.7 循环和编码风格	81
4.8 switch语句	61	5.8 小结	82
4.9 goto语句	62	5.9 练习	82
4.10 抛弃“魔数”	62	<b>第6章 C语言函数</b>	<b>83</b>
4.11 C语言的预处理指令	63	6.1 函数剖析	84
4.12 正面还是反面	65	6.1.1 函数类型说明符	84
4.12.1 初始化环节	65	6.1.2 函数名	84
4.12.2 输入环节	65	6.1.3 函数参数	85
4.12.3 处理环节	65	6.1.4 函数体	86
4.12.4 输出环节	66	6.1.5 函数签名	87
4.12.5 结束环节	66	6.2 “优质”函数所具备的条件	88
4.13 思考一下	68	6.2.1 使用面向任务的函数命名	88
4.14 小结	69	6.2.2 函数应该具有聚合力	89
4.15 练习	69	6.2.3 函数应避免耦合	89
<b>第5章 C语言的循环</b>	<b>70</b>	6.3 编写自己的函数	90
5.1 正常循环的特征	70	6.3.1 设计函数的考虑因素	90
5.1.1 条件1: 变量初始化	70	6.3.2 参数列表	91
5.1.2 条件2: 循环控制测试	71	6.3.3 函数体	92
5.1.3 条件3: 修改循环控制变量的 状态	71	6.4 逻辑运算符	92
5.2 使用for循环	71	6.4.1 逻辑与运算符(&&)	92
5.3 while循环	73	6.4.2 逻辑或(  )	93
5.4 do-while循环	74	6.4.3 逻辑非(!)	94
5.5 break和continue关键字	75	6.5 编写自己的函数	94
5.5.1 break语句	75	6.5.1 IsLeapYear()函数和编码 风格	95

6.5.2 实参 (Argument) 和形参 (Parameter) .....	96	8.7 练习 .....	134
6.5.3 为什么要坚持一种函数风格 .....	96	<b>第9章 高效使用指针</b> .....	136
6.6 闰年计算程序 .....	97	9.1 针对指针的关系判断 .....	136
6.7 函数中数据的传入和返回 .....	99	9.2 指针运算 .....	137
6.8 小结 .....	102	9.3 二维数组 .....	141
6.9 练习 .....	102	9.3.1 一点小改进 .....	144
<b>第7章 存储类型和作用域</b> .....	103	9.3.2 几维 .....	144
7.1 隐藏程序数据 .....	103	9.4 二维数组和指针 .....	144
7.2 语句块作用域 .....	103	9.5 函数指针 .....	147
7.3 本地作用域 .....	105	9.5.1 函数指针数组 .....	149
7.4 全局作用域 .....	108	9.5.2 enum数据类型 .....	149
7.5 作用域和存储类型 .....	110	9.6 右左右左规则 .....	153
7.5.1 auto存储类型 .....	110	9.7 小结 .....	154
7.5.2 register存储类型 .....	110	9.8 练习 .....	154
7.5.3 static存储类型 .....	111	<b>第10章 结构、联合和数据存储</b> .....	155
7.5.4 extern存储类型 .....	111	10.1 结构 .....	155
7.6 volatile关键字 .....	116	10.1.1 声明结构 .....	156
7.7 小结 .....	116	10.1.2 定义结构 .....	156
7.8 练习 .....	116	10.1.3 访问结构成员 .....	157
<b>第8章 初识指针</b> .....	117	10.1.4 从函数调用返回结构 .....	160
8.1 定义一个指针 .....	117	10.1.5 使用结构指针 .....	162
8.1.1 指针名 .....	117	10.1.6 初始化结构 .....	164
8.1.2 星号 .....	118	10.1.7 结构数组 .....	165
8.1.3 指针类型说明符和指针标量 .....	118	10.2 联合 .....	165
8.2 使用指针 .....	121	10.3 EEPROM内存 .....	166
8.2.1 取值运算符 (*) .....	122	10.4 数据日志 .....	167
8.2.2 使用取值运算 .....	123	10.5 其他存储选择 .....	175
8.3 指针用法小结 .....	125	10.5.1 Shield .....	175
8.4 指针为什么有用 .....	126	10.5.2 SD卡的其他功能 .....	176
8.5 指针和数组 .....	129	10.6 小结 .....	177
8.6 小结 .....	134	10.7 练习 .....	178

<b>第11章 C预处理和按位操作</b> .....	179	<b>附录A 供应商和资源</b> .....	208
11.1 预处理指令.....	179	A.1 供应商.....	208
11.1.1 #undef.....	180	A.1.1 Seeedino Studio.....	208
11.1.2 #line.....	181	A.1.2 Diligent Inc.....	208
11.1.3 #if条件指令.....	182	A.1.3 OSEPP.....	209
11.1.4 #include.....	183	A.1.4 Tinyos Electronics.....	210
11.2 带参宏.....	184	A.1.5 Cooking Hacks.....	210
11.2.1 按位操作符.....	184	A.2 资源.....	210
11.2.2 按位移动操作符.....	187	<b>附录B 实验用电子元器件</b> .....	212
11.2.3 使用不同进制的整型常量.....	188	B.1 开发板.....	212
11.2.4 带参宏(补充内容).....	189	B.2 面包板.....	212
11.3 小结.....	189	B.3 电子元件.....	212
11.4 练习.....	189	B.4 网购元件.....	213
<b>第12章 Arduino库</b> .....	191	B.5 实验.....	213
12.1 库.....	191	<b>练习答案</b> .....	215
12.1.1 Arduino库.....	191	第2章.....	215
12.1.2 其他库.....	196	第3章.....	215
12.2 编写自己的库.....	198	第4章.....	217
12.2.1 库头文件.....	199	第5章.....	219
12.2.2 库源代码文件(Dates.cpp).....	200	第6章.....	221
12.3 设置Arduino IDE以使用自己的库.....	203	第7章.....	222
12.4 使用Dates库实现的示例程序.....	203	第8章.....	222
12.4.1 向IDE添加Easter程序.....	205	第9章.....	223
12.4.2 keyword.txt文件.....	205	第10章.....	224
12.4.3 关键词颜色(theme.txt).....	206	第11章.....	226
12.5 小结.....	207	第12章.....	227
12.6 练习.....	207		

# 第 1 章

## 简介

这本书只有一个主要目的：教会你如何使用 C 程序语言来控制 Atmel 系列单片机。既然市面上有大量的 C 程序语言的书籍，你为什么要选择这一本呢？好问题，答案可能有很多种。不过，这里可以为你提供几个判断的依据，帮助你做决定：这本书是专门针对使用 Atmel 系列微处理芯片的 Arduino 系列单片机的。本书基于可从 Atmel 网站下载的综合开发环境（Integrated Development Environment, IDE）框架。这意味着你不必为学习 C 语言而购买其他的编程工具。不过，这个 IDE 所提供的 C 语言实现并不是美国国家标准学会（American National Standards Institute, ANSI）所定义的 C 语言，它实现的是 Arduino C，我们今后也会这样称呼它。Arduino C 是 ANSI C 的一个稳定的子集，所以，在学习时，可以略过其中不可用的某些 C 语言特性。虽然好处在于要学习的内容减少了，但这也意味着某些 ANSI C 中才有的功能需要我们变通地实现。

如果仅考虑上面两点的话，其实市面上也有许多其他讲解 Arduino C 的书籍，那么选择这本而非其他书的根本原因是什么呢？

首先，这是一本程序开发书籍，而且专注于此。虽然我们确实有很多小的硬件开发项目要完成，但真正的目的是学习 C 语言的开发——而非硬件本身。当你掌握了 C 语言之后，市面上有不少 Arduino 单片机书籍，你可以通过阅读它们来拓展硬件编程实践。

第二，我将带你进入 C 语言的本质，这样你可以获得对代码完成什么功能，以及怎样完成功能有更深入的认识。这种认识对于你在面对极小的内存环境时将非常有用。虽然有人说理解一门语言并不需要深入其内部细节，而且某些常用语似乎特别支持他们的观点，比如“不懂造车无碍开车（You don't have to know how to build a car to drive one）”。虽然这句话是对的，但如果你的车在加拿大西北地区的耶洛奈夫（Yellowknife）郊外 200 英里（1 英里≈1.61 千米）处抛锚了，那时你肯定会希望懂得哪怕是一点点的汽车原理。对于程序开发而言也是如此，你更好地了解一门语言的原理，那么就能更快速地发现、分析并解决问题。而且，对于一个问题，可能会有很多种解决的办法，选出更高效、更稳健甚至更优美的解决方案完全依赖于你对语言的认识深度。

第三，自从我在 1977 年第一次使用 C 语言到现在，已经积累了大量商用级的 C 语言开发经验。我的软件公司为早期的 PC 开发了 C 编译器和其他的开发工具。另外，三十几年前我



编写了第一本 C 语言教材，而对你最有用的还是我所拥有的教学经验。诚然，与我开发能力相当的程序员可能成千上万，我的一个雇员就是一个例子。但是，对于一个好的作者来说，他不仅需要拥有好的编程水平，还要有能力让周围的人也达到这样的水平。我拥有 30 多年的大学任教经验，所以我能清楚地知道，你可能会在什么地方卡住。虽然坏消息是你肯定会遇到很多困难，但好消息是在你之前，已经有成千上万的人遇到过了，所以我获得了大量的经验，来帮助他们和你战胜所有的这些困难。我相信你会发现这本书既实用又有趣。

最后，我的的确确非常着迷于 C 语言。我从 20 世纪 60 年代第一次开始编程到现在，使用过许许多多不同的语言，但 C 语言仍然是我的最爱。它简洁而强大，特别适于微控制器。所以我相信你也会喜欢上它。

这一章主要介绍为了更好地学习本书中的内容，你都需要做哪些准备，以及我对于你的一些基本要求。你将了解各种兼容 Arduino 的板子都有什么功能、大致的价格，以及可以在哪里购买（附录 A 中提供了一些供应商）。另外还介绍了其他一些可能会用到的硬件。然后，本章将告诉你如何下载 Arduino IDE 以及如何安装。最后，本章以一个简短的 C 程序作为结束，以检查 IDE 环境是否搭建成功。当你学完这一章后，会知道高效地学习这本书都需要哪些东西。

### 1.1 一些假设前提

显然，这本书有特定的读者群。所以，我需要读者有一定的假设：假设读者对 C 语言或者其他编程语言一无所知。我的确希望你什么都不懂，因为这样的话，可以从零开始。通常，具有一定的程序开发基础的人或多或少都有一些“坏习惯”需要纠正。所以，完全零基础地开始学习反倒是一件好事。

我还假设你对电子也一窍不通。不可否认的是，一些硬件方面的概念始终贯穿本书，但当遇到这些概念的时候，我将为你讲解，以便更好地学习后面的内容。如果你想要深入了解电子方面的知识，我建议你先学完这本书，然后再购买讲解你所感兴趣的某个具体硬件的图书。

我还假设你一定会做练习。当然，这意味着你需要在单片机和一些附加组件上小小地破费。但是我会尽可能地在保障学习的前提下，使用便宜的组件。附录 A 为你提供了一些供应商，你可以在那里买到价钱合理的各种配件。如果那里没有，也没关系，因为几乎所有的东西都可以在附近的无线器材铺中买到。附录 B 列出了许多细碎的配件，如果你要完整地实现本书的项目的话，就需要这些东西。不过，如果你打算略过某些实例，那么某些配件可能就不需要了。

我还假设你不会贪图速度，不会想要在这周末之前就把这本书搞定。也就是说，你会耐心地做练习，花时间仔细学习，并完全理解示例代码。C 语言的学习是一个循序渐进的过程，这一章学到的概念可能在后续章节中都会用到。而如果对某一章的概念一知半解，则可能导致后面的学习无法进行。所以，你要静下心来，慢一些，仔细思考你正在学习的内容，并且