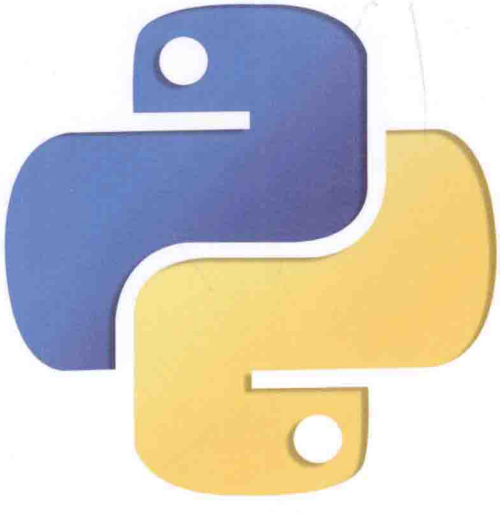
HZ BOOKS
华章科技

本土作者力作，应用驱动、接地气的Python速成攻略。
软硬件相结合，第一本基于开源硬件讲解Python的实用指南。
基于pcDuino的开发实例、完整的项目文件和源代码，可操作性极强。

数字匠人



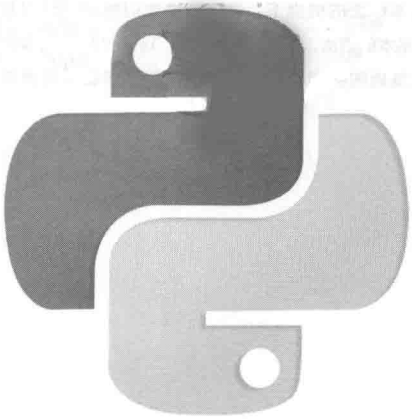
pcDuino

Python 硬件编程实战

李 茂 编著

机械工业出版社
China Machine Press

数字匠人



pcDuino

Python 硬件编程实战

李茂 编著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Python 硬件编程实战 / 李茂编著 . —北京：机械工业出版社，2015.1
（电子与嵌入式系统设计丛书）

ISBN 978-7-111-48774-6

I. P… II. 李… III. 软件工具 - 程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字（2014）第 287930 号

本书主要针对计算机基础比较薄弱的 Python 语言初学者，力图使用通俗易懂和深入浅出的语言风格阐述 Python 的基本概念。在对 Python 建立基本概念的前提下，循序渐进地引导读者学习 Python 版本的选择、不同平台下 Python 开发环境的搭建、Python 基本的语法，并最终使读者可以利用 Python 实现一些简单的应用开发。

本书主要适用于没有 Python 基础的初学者，包括但不限于具有硬件背景的工程师、非计算机专业的读者、Python 业余爱好者和学生等。

Python 硬件编程实战

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：李　燕　　责任校对：殷　虹
印　　刷：北京市荣盛彩色印刷有限公司　　版　　次：2015 年 2 月第 1 版第 1 次印刷
开　　本：186mm×240mm　1/16　　印　　张：12
书　　号：ISBN 978-7-111-48774-6　　定　　价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

版权所有 · 侵权必究
封底无防伪标均为盗版
本书法律顾问：北京大成律师事务所　韩光 / 邹晓东

前　言

笔者是一名有 6 年嵌入式软件研发经验的工程师，由于工作需要和自己的兴趣，在最近 3 年的时间开始接触上层软件方面的技术，其中就包括学习和使用 Python。越深入地学习，越让笔者觉得 Python 好用，于是陆续在博客上更新了很多 Python 的教程和资料，这本书的大多数素材均来自笔者的博客。

Python 语言从出现到如今已经有很长时间了。市面上关于 Python 的入门读物更是数不胜数，所以想再写一本入门的好书需要极大的勇气和决心。真正触动笔者下决心写这本书的原因是笔者发现对于很多国内读者，尤其是很多计算机基础相对薄弱的国内读者来说，市面上的教程往往很难看懂。因为这些 Python 图书多数都是国外作者写的，面对的读者是具有较强的计算机基础的国外用户。不仅如此，这些书被翻译成中文，在国内销售，还忽略了中外读者的差异，导致很多国内读者即便是看入门级的 Python 读物，学习起来仍然觉得很吃力。国外的作者往往会省略一些背景知识，而国内读者又恰恰缺少这些相应的背景知识，自然很多内容无法完全理解。举个例子，在解释选择 Python 的 IDE 开发环境时，如果直接让国内初学者使用某个界面相对好看但是功能复杂的 IDE，很多人会很难理解如何使用。初学者对于 IDE 和原始的命令行之间的关系本身就不是很清楚，在这种情况下，又怎么可能学会使用呢？笔者觉得应该换一种更好的方式去解释，即先分析透彻原始的开发环境和复杂的 IDE 之间的关系，然后解释为何要选择某个 IDE，再进一步介绍 IDE 中每个部分的功能和之前原始的命令行方式的对应关系。只有如此解释，初学者才能明白选择 Python 开发环境的真正含义，才能理解为何选择和使用某个 IDE。基于此，笔者才陆陆续续地编写了一系列针对 Python 初学者的帖子，并努力把每个知识点讲透。笔者觉得这样才能真正帮助读者学习和掌握 Python。

全书共分为 7 个章节：

- 第 1 章主要阐述一些与 Python 相关的基本概念，并介绍其应用领域；
- 第 2 章介绍如何下载和安装 Python；
- 第 3 章介绍如何在 Windows、Linux、Mac 等多种不同平台下开发 Python 应用，以及如何选择合适的开发环境；

- 第 4 章解释 Python 的基本知识；
- 第 5 章给出一些有趣的 Python 小实验；
- 第 6 章进一步给出 Python 在各种领域内的实际应用案例；
- 第 7 章介绍 Python 与开源硬件之间的关系，了解如何用 Python 搭配开源硬件以实现各种有趣的功能。

本书主要针对没有基础的初学者，包括但不限于具有硬件背景的工程师、非计算机专业读者、Python 业余爱好者、学生等。鉴于轰轰烈烈的创客运动，笔者把这本书的定位描述成创客的 Python 入门书。其实在笔者看来，只要是想动手，实现自己心中所想的人都应该是创客。从这个角度出发，笔者觉得任何一个想要学习 Python 的初学者都是创客。这是一本真真正正的入门书，读者只需具备计算机编程语言的基本知识，即可阅读本书。本书会带领读者从了解 Python 的概念开始，到搭建 Python 的开发环境，再到真正使用 Python 语言实现一些应用。

本书中所用到的标识及其含义如下：


提示。提示信息，如用于解释一些名词的含义、阐述相关知识背景。


注意。用于强调一些注意事项、心得体会等内容。


重要。提示某些重要的信息。


警告。用于指出严重的错误，提醒读者一定要避免这类错误。

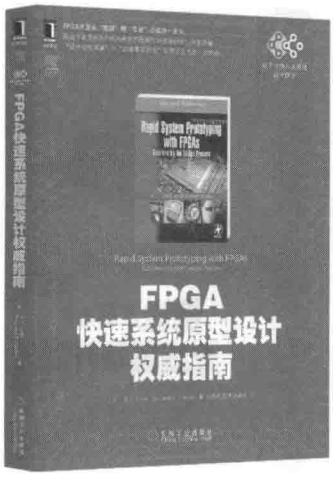
笔者首先感谢父母的教育，没有他们的悉心培养，笔者很难有今天的成就。笔者还要感谢朋友高静在此期间给予笔者的支持和无私的帮助，让笔者能够终成此书。由于笔者水平有限，书中错误在所难免，欢迎读者和同行们指出错误，提出您的宝贵意见或建议。

李茂

2014 年 11 月

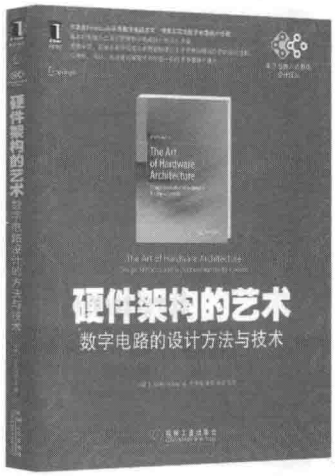
试读结束：需要全本请在线购买： www.ertongbook.com

推荐阅读



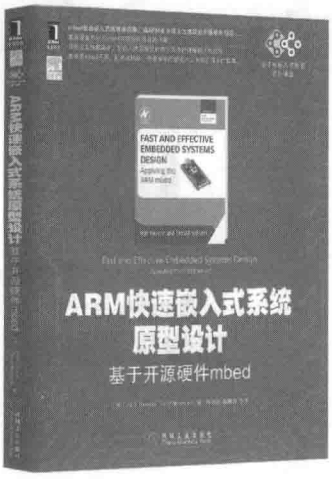
FPGA快速系统原型设计权威指南

作者：R.C. Cofer 等 ISBN：978-7-111-44851-8 定价：69.00元



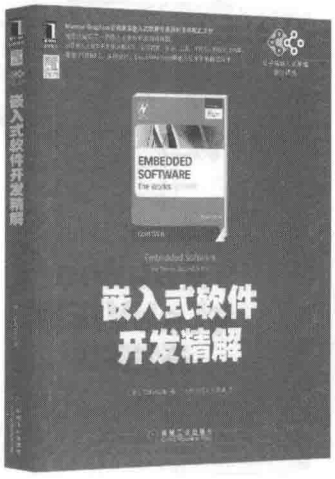
硬件架构的艺术：数字电路的设计方法与技术

作者：Mohit Arora ISBN：978-7-111-44939-3 定价：59.00元



ARM快速嵌入式系统原型设计：基于开源硬件mbed

作者：Rob Toulson 等 ISBN：978-7-111-46019-0 定价：69.00元



嵌入式软件开发精解

作者：Colin Walls ISBN：978-7-111-44952-2 定价：79.00元

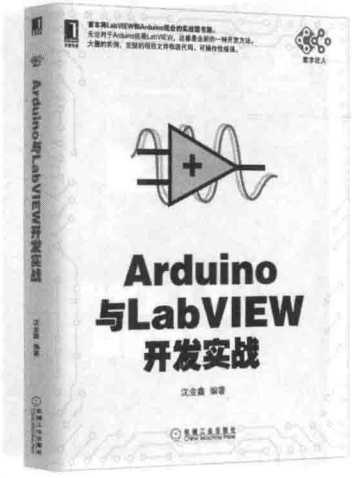
推荐阅读

Arduino高级开发权威指南（原书第2版）

作者：Steven F. Barrett ISBN：978-7-111-45246-1 定价：59.00元

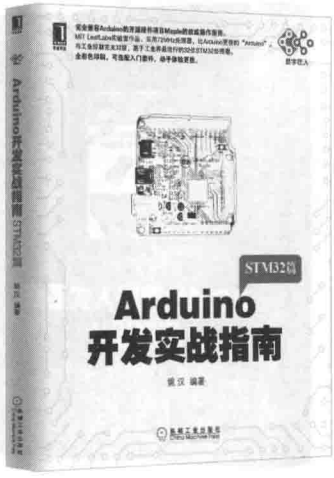
例说XBee无线模块开发

作者：Jonathan A. Titus ISBN：978-7-111-45681-0 定价：59.00元



Arduino与LabVIEW开发实战

作者：沈金鑫 ISBN：978-7-111-45839-5 定价：59.00元



Arduino开发实战指南：STM32篇

作者：姚汉 ISBN：978-7-111-44582-1 定价：59.00元

目　录

试读结束：需要全本请在线购买： www.ertongbook.com

第 1 章　Python 简介

1.1　Python 是什么

Python 作为一个英文单词，其本意是巨蟒、蟒蛇的意思。

Python 这个词在计算机语言领域内指的是一种计算机语言叫作 Python。

Python 语言名字的由来

我们之所以把蟒蛇——Python 用于命名计算机语言是有其历史典故的。

1989 年在阿姆斯特丹，Python 语言的创始人吉多·范罗苏姆（Guido van Rossum）为了打发圣诞节的无趣决心开发一个新的脚本解释程序，在给此新的计算机语言起名字时，由于其本人是巨蟒剧团[⊖]的忠实粉丝，所以就把此计算机语言的名字叫作了 Python。这就是 Python 作为一种计算机语言的名字的由来。

Python 语言被称为了蟒蛇，名字看似很凶猛，但并不代表 Python 语言本身很“凶猛”、“吓人”，反倒是谈论起 Python 语言功能的强大时，用“凶猛”来形容并不为过。

1.2　对 Python 的四种定义

对于某种计算机语言，根据其特点和语言本身侧重点的不同会有不同的分类和叫法。Python 作为计算机语言的其中一种也不例外。接下来就来详细解释 Python 的各种不同的分类和叫法的详细含义。

1.2.1　一种脚本语言

脚本的英文是 script。一般的读者看到 script 这个单词往往首先想到的是电影的剧本，我们大多数人都知道电影剧本其实就是由一段段的脚本所组成的，即电影剧本的脚本决定

⊖ 巨蟒剧团（Monty Python）是始于 20 世纪 70 年代而后风靡全球的一个英国六人喜剧团体。

了电影中的人和物要做哪些事情以及具体怎么做。

与之类似，计算机中的脚本决定了计算机中的操作系统和各种软件工具要做哪些事情以及具体怎么做。

此外，根据笔者的理解，脚本这个词还会让人有另外一种感觉：随性。

所谓的随性就像在现实中写电影脚本，直接拿张纸和一支笔就可以写了，写完之后就可以拿去使用，即拍电影了。而对应的计算机中的脚本也是类似的过程，用户想要实现一个功能，在构思了如何做之后，就可以直接找个文本编辑器写上对应的脚本，也就是普通的文本，接着让计算机去运行，从而实现想要的功能。当然计算机脚本与电影脚本也有不同之处：计算机中的脚本所运行的环境中有对应的脚本解析器，可以解释并执行对应的脚本。

例如，若创建 Linux 中的 shell 脚本，可以先创建一个普通文本文件，然后在其中添加 shell 脚本代码，保存文件，接着就可以在 Linux 的带有 shell 解释功能的环境中运行了。

那些相对“随性”的脚本语言来说不那么“随性”的非脚本语言叫作编译型语言，比如 C 语言，它们需要再加上编译这个步骤之后才能生成可以运行的程序，然后才能执行程序。

编译型语言和脚本语言的对比

（1）编译型语言

简单地说，编译性语言就是需要用该语言的编译器将源代码编译为可执行程序，然后才能运行可执行程序的语言。此编译过程需要先将源代码编译为目标文件，然后把目标文件加上必要的库文件，最后再链接为最终的可执行文件。

（2）脚本语言

脚本语言是无需用编译器编译源代码，可直接运行该源码形式脚本文件的语言。而直接能够运行脚本文件的背后，是当前环境中存在着此脚本的解析器。解析器负责读入此脚本源码，以及后续解析并执行的动作。

编译型语言和脚本语言的执行示例

（1）编译型语言的执行

编译型语言的执行过程为

源代码 ⇒（用编译器编译成）可执行程序 ⇒ 运行程序（可执行文件）

以在安装了 Visual Studio 2010 的 Windows 7 中编译运行 C 语言为例，其过程为

hello.c⇒（用编译器 cl 编译为）hello.exe⇒ 运行 hello.exe

其中，hello.c 是先被编译器 cl 编译为目标文件 hello.obj，然后再调用 link 将目标文件 hello.obj 链接为可执行文件 hello.exe 的。

（2）脚本语言的执行

脚本语言的执行过程为：

源代码 ⇒ 直接（在具有了解析器的环境中）运行程序（脚本源码）

以在安装了 Python 的 Windows 中运行 Python 脚本为例，其过程为：

demoPython.py⇒（直接在有了 Python 解析器 python.exe 的环境中）运行 demoPython.py

其中，python.exe 是安装了 Python 时就会被安装的 Python 解释器，负责解析并运行 Python 脚本文件。

1.2.2　一种解释型语言

我们有时候也会把 Python 称为一门解释型语言。这是因为脚本语言的天然特点之一就是解释性，而作为脚本语言的 Python 因此也可以被称为是一门解释型语言。

什么是解释型语言

脚本语言的特点是在具有解释器的环境中，以源代码的形式而无需编译就可以直接执行。脚本语言的内部执行过程是首先由解析器一行一行地读取脚本的源代码，同时解析每一行，然后给出执行后的结果。通俗地说就是：读一行，解释一行，执行一行。由此，脚本语言就天然地具有了解释性的特点，所以脚本语言也常被称为解释型语言。

1.2.3　一种高级语言

在解释 Python 为何是一种高级语言之前先来解释一下什么是高级语言。

什么是高级语言

事物的发展都是从低级走向高级的，计算机语言的发展也不例外。早期的计算机语言大多数都是针对硬件上的机器本身所开发的语言，一般叫作汇编语言。汇编语言之所以常被称为低级语言，是因为其语言本身直接和硬件打交道，而缺少对计算机细节的抽象，相对而言不是那么易于程序员理解和使用。

随着计算机语言的发展，出现了对计算机细节更宏观抽象的语言，这类语言更多地采用人类更容易理解的元素和概念，因此对于程序员来说，相对更容易理解、学习、掌握

和使用这类语言，而这类语言被称为高级语言。常见的一些高级语言有 Fortran、Pascal、Lisp、C、C++、Java、C#、Python、Ruby 等。

高级语言和低级语言的对比

高级语言在本身的设计层面会考虑到对计算机细节的封装和抽象。

比如一个普通的计算机语言概念中的数组，在常见的高级语言中有具体的实现，程序员可以很方便地直接拿来使用。而作为大多是汇编语言的低级语言，它们对于数组的实现和使用则很复杂，需要手动去写很多复杂的汇编代码才能使用。因为汇编语言等低级语言和底层硬件有关系，比如不同的 CPU 会有不同的指令集和不同的寄存器等各种内部资源，所以要具体实现数组这样的功能时，会有千差万别，需要考虑到很多细节和不同的实现方式，需要每个程序员对当前所使用的 CPU 的所有架构、内部资源等细节都很清楚，同时也需要对当前特定 CPU 的指令的语法都很清楚。在此前提下，才可以写出对应的汇编代码，才可以实现对应的数组。

由此可见，汇编语言会涉及太多物理层面上和硬件上的实现细节，不利于人类以普通的逻辑去理解，更难学习和掌握。而相对来说，高级语言涉及的基本元素和概念和人类的思维很相似，所以高级语言对于程序员来说更加容易理解、学习和掌握。

然后我们再来具体地解释为何 Python 是其中一种高级语言。Python 在语言设计的时候，和其他高级语言类似，也完全具有常用的各种基本元素，比如各种普通变量、列表、函数等内容，完全符合人类的逻辑，易于理解、学习和掌握，所以 Python 本身的确是一种高级语言。

1.2.4 一种面向对象的语言

在解释为何 Python 是一种面向对象的语言之前，先来简要介绍一下什么是面向对象的语言。

什么是面向对象的（高级）语言

首先要明确的是，面向对象的计算机语言这个概念是针对高级语言来说的。其次，在一堆高级语言里面，有些是在设计该语言本身时，对于语言本身的基本元素是以对象的方式设计的，而不同的对象之间的交互则成为整个程序运行的主要表现形式。

此处的对象往往和现实中的物体、概念、逻辑的整体等内容是一一对应的关系。由于

程序员用基于对象的编程语言设计程序时往往更加直观和易于理解，更加容易简化问题的处理逻辑，所以相对而言，面向对象的语言简化了使用该语言解决实际问题的复杂度，提高了处理事情的效率，所以程序员更容易理解、学习、掌握和使用。

而 Python 语言本身的设计也是基于对象的，也是面向对象的各种概念和逻辑的，所以说 Python 也是一种面向对象的编程语言。

有了前面介绍的对于不同 Python 称谓的理解之后，我们再来总结一下作为计算机编程语言的 Python 的常见且基本的定义：**Python 是一种面向对象的、解释型的计算机高级语言。**

至此可能很多读者才突然明白：哦，原来在各种资料或教程中所使用的 Python 的上述定义背后，还隐含着如此多方面的含义，原来 Python 的各种特点和称谓是这么回事。

如果读者会有此感悟，则才真正实现了笔者写本 Python 教程要实现的目的：希望学习技术的读者，不仅要知其然，也要知其所以然。将此逻辑应用到此处的含义就是：不仅要知道 Python 是如此定义的，更要知道为何会有这样的定义，即 Python 如此定义背后的真正完整的含义。

1.3　Python 的特点

1.3.1　作为脚本语言的优缺点

上面已经介绍过，Python 是一门脚本语言，也是一门解释型语言。下面就来简单解释一下作为解释型语言的 Python 有哪些特点。

1. 作为脚本语言的 Python 的优点

❑ 快速开发：不需要编译即可运行

正如前面的解释，写完 Python 脚本后直接就可以运行而省去编译的步骤，使用起来相对省事和高效。

2. 作为脚本语言的 Python 的缺点

❑ 性能相对不是特别强

Python 的性能相对一些其他语言（比如 C、C++ 等）来说不是特别强。对于性能要求比较苛刻的某些领域不太适合全部使用 Python 去实现所有的功能。现在已有的解决方法是：使用相对 Python 性能更好的其他语言去实现与性能相关的最核心部分的功能，然后再将此部分

试读结束：需要全本请在线购买：　www.ertongbook.com