<<真实世界的Python仪器监控>>

图书基本信息

书名：<<真实世界的Python仪器监控>>

13位ISBN编号：9787121186592

10位ISBN编号：7121186594

出版时间：2013-1

出版时间：电子工业出版社

作者：约翰·休斯

译者：OBP Group

前言

本书介绍自动化仪器及其自动化控制。
我们将探讨如何运用Python 语言快速轻巧地构建自动化仪器的控制系统。
从研究实验室到工业厂房，自动化仪器无所不在。
一旦人们意识到收集随时间变迁的数据很有用，自然就需要某种手段来捕捉并完成数据记录。
当然，人们可以取叠纸拿个时钟，盯着温度计、刻度盘或是其他仪表，定期记录数值，但是很快就会受不了这种乏味的工作。
如果这一记录过程可以自动化，无疑将更加可靠和易行。
幸运的是，技术的进步早已超越了手写日志及发条驱动的带状图记录的时代！
如今，人们可以购买各种便宜的物理仪器并使用计算机来获取数据。
一旦计算机被连接到仪器，数据收集、分析和控制等等功能就可以自由扩展，唯一受限的只是实现者自身的创造力了。
本书的主要目的是向读者展示如何创建一个有能力同用户友好交互的仪器或控制应用程序软件，并使用最低成本运行起来。
为此，我们仅基于最必需的步骤来创建程序，包括怎么使用不同类型的输入/输出硬件接入现实世界的底层接口。
我们也将研究一些行之有效的方法，用以指导创建强大且可靠的程序。
特别提醒，应该为数据处理所必需的算法支付设计费用。
最终，我们将体验如何为用户设计命令的输入以及结果展示。
如果读者能从本书中发现一些想法，并创造性地运用在各种仪器设备上，满足自己的需要，那么我的愿望也就达成了，善哉。
本书的目标读者本书专为需要或是自制仪器控制器（也称为数据采集和控制系统）的人准备的。
你可能是名研究员、软件开发者、学生、项目主管、工程师，或一个业余爱好者。
想实现的应用系统，可能只是在实验过程中需要的自动化电子测试系统，或是其他类型的自动化设备。
本书要完成的目标软件将是跨平台的。
我假定你至少在Windows 平台特别是XP 平台玩得很顺。
而我会使用Ubuntu 发行版本的Linux系统，不过书中讨论的程序将在各种兼容发行版中良好运行，同时我也假定你知道如何使用csh 或是bash 命令行脚本。
由于本书是关于如何通过物理硬件同现实世界交互的，其中自然涉及了一系列相关电气产品。
但是，并不要求读者是名有足够背景知识的电气工程师。
在第2 章，包含了基本电子理论知识的介绍，虽然事实上不必理解深层次的电子学知识也可以令计算机与现实世界交互，不过，知道多点相关领域知识绝对没坏处，万一首次遇到意外，我们可以从中获得思路。
不论读者的工作类型或场所怎样，最关键的，我假定你需要通过某些硬件接口捕获一些数据，或是产生控制信号。
更加重要的是，需要轻便又精确且可靠地构建出这些仪器的控制软件来。
本书所用编程语言我们将使用Python作为主要的编程语言，仅仅嵌入一点点的C程序。
在本书中，我将假定你有一些编程经验，并对Python或C（理想情况下，两者都）熟悉。
如果不是这样，有Perl 或Tcl/Tk 或如Matlab 或IDL 分析工具的经验，也是一个合理的起点。
本书坚定地回避Python 语言更深奥的知识，配合大量的实例代码、图表注释和截屏来引导理解。
对C 涉及得很少，只用来说明如何创建和使用Python 应用的底层系统扩展。
第3 章覆盖了Python 语言的基础介绍，第4 章介绍了C 语言的基础知识，对以上语言进一步的探究可通过阅读建议自行学习。
为什么选用PythonPython 是Guido van Rossm 在80 年代末开发的解释型语言。
因其是种即时编译的脚本语言，故而用户可以在Python 命令行环境中直接创建并执行。

语言本身很容易学习和理解，只要一开始别理会过多的高级功能（装饰器，自省，列表推导，等等）就行。
因此，Python 提供了快速构建原型及易懂的双重好处，这反过来又有利于快速为不同的设备创建各种不同应用，没有开发者通常需要应对的学习曲线以及传统的编译语言依赖的特定供应商提供的编程环境。
Python是高度可移植的，几乎运行在所有现代计算平台中。
在项目中坚持只使用常用的接口方法，应用程序就很可能在安装Windows的PC中编写，但是不用修改一行代码也可以在Linux操作系统中运行良好。
甚至于可以在Sun的Solaris机器和Apple的OSX系统中运行，即使书中没有特意提及这一点。
一旦Python必须配合特定平台的特定扩展或驱动程序，便失去了可移植性，所以在这些情况下，我将提供分别适用于Windows 和Linux 的替代品。
本书包括了完整的可用示例代码，并配合框图和流程图来说明关键点，操作一些现成的、低成本的接口硬件。
系统我们将探究的是那种既可以用在实验室，也可以直接用于工业环境的仪器设备。
比如说，应用于电子实验室、风洞中的设备，或是进行气象数据收集的设备。
而所说的系统，可能只是一个简单的温度记录仪，也可能是个复杂的真空控制系统。
一般来说，本书描述的技术可以作用于任何可以连接到PC的硬件，当然总是有些设备是使用封闭标准的特殊硬件，但是我们不会处理这些，也不会深入到复杂的数据处理领域，比如说：炼油厂的自动工程方案，核电厂，或机器人飞船。
系统在这些领域通常是配合同样精密和复杂的软件，并通过专用硬件来实现极其复杂的控制。
我们只关注最通常的设备、驱动和接口，以及使用一些通用界面方法轻松构建出可用的系统。
方法论我们通过现实世界的实例一步步地理解如何定义设备应用，选择合理的接口以及硬件，建立可能需要的底层驱动以便配合Python接口与完成硬件控制。
我们还将探索TkInter和wxPython 的图形界面，以及curses 的图形化文字界面。
本书包含的主要内容如下。
如何封装一个硬件供应商的y DLL 驱动，以便Python 扩展使用。
如何与以USB 为基础的I/O 设备通信。
如何使用类似RS-232 和RS-485 或是GPIB 工业标准接口。
追加上一个什么样的硬件类型才可能发现并使用接口。
本书还提供了参考，可以索引到现成的开源工具和库，以便即使从零开始，也可以用最短的时间完成一个可用的硬件控制系统。
本书的内容组织本书分为14章和2节附录。
第14章将开头12 章的所有实践集中应用为一系列现实世界的例子。
第1 ～ 6章引入了基础概念，读者可以选择跳过。
每章重点内容主要如下。
第1 章 仪器介绍从整体上来说什么是仪器，如何控制系统的工作，以及这些概念如何在实际世界使用。
涵盖的例子包含自动户外灯，电器仪表在工程中的测试，在实验室中控制化学过程和热度批处理。
第2 章 基本电子学作为手册书，必须对物理硬件接口以及如何完成一个自动化工程从整体上涉及的各个方面有描述。
本章对电子以及电气产品进行了简介，然后探讨了内置功能模块以及数字控制接口、模拟接口、计数器和计时器。
最后，介绍并评论了作为幕后技术的串行和并行接口。
如果你已经熟悉电子电路原理和装置，可以跳过本章。
不过，建议至少要关注一下，以便为今后可能的利用留个印象。
第3 章 Python 编程语言这本书不是Python 教程，本章提供了Python 的基本知识以及核心概念，以便读者快速上手将Pytho最常用的功能在本书讨论情景中应用起来。

本章还提供了一系列工具的概述，以便协助大家更加轻松地进行编程。
第4章 C 语言本章从较宏观的层次介绍了C语言知识。
目标是提供足够的资料，以便理解书中实例代码，并不会引入C语言神秘的细节。
幸运的是，C其实是相对简单的语言，而在这一章的信息应足以让你开始创建自己的驱动扩展以供给Python 脚本使用。
第5章 Python 扩展本章介绍了Python 扩展是如何创建的，以及通常有哪些种类的扩展。
提供的例子，无论是在本章，或是在后面的章节，都可以作为你自己应用的模板起点。
第6章 硬件：工具与耗材虽然读者可能从来没有触摸过仪器设备以及电烙铁，但是很有可能用过螺丝刀、钢丝钳和数字万用表（DMM），这也足够开始了。
在这一章我提供了一个清单，来说明开发仪表系统需要什么样的基本工具，以便可以按图索骥填充到书架上一个小盒子里，为你将来可能真的动手创造自动系统时使用。
在最后，充分讨论了两件可以帮助你免除猜测之苦快速定位问题的设备：示波器和逻辑分析仪。
本章还就应该准备哪些类型的工具提供了一 些可行性建议，以便读者参考决定购置或是升级。
第7章 物理接口列出了一系列用Python 进行数据采集或是控制时最常见的物理接口。
RS-232 和RS-485，就仪器控制而言，这就是最常见的两个串行接口。
本章还涵盖了一些讨论场景中会遇见的USB 和GPIB/IEEE-488 接口的基础知识。
最后，我们将关注PC 的可插入式I/O 硬件，即通常的PCI 型电路板，以及通常可从硬件供应商得到支持的API规范。
第8章 入门本章描述了一个对软件开发行之有效的过程。
将这一内容设置在这里是因为，不论用何种语言来开发自控系统，至关重要的是计划，明确核心功能，比照预期数据来观察测试结果，并持续改进这一过程。
通过开发Python 扩展，我们将击穿模糊和不确定之门，通过设备控制软件进入真实世界。
第9章 控制系统概念要对现实世界进行数据采集和控制，必须至少掌握部分控制和系统论。
本章扩展了在第1章中介绍的控制系统概念，详细介绍了检查与共同控制系统概念和模型，包括了如反馈、“ Bang-Bang ”控制器和比例积微分（PID）控制等议题。
还提供了一个基本控制系统案例分析，并给出了如何选择合适模型的一些指导方针。
最后，介绍如何应用数学控制系统转化为实际的Python代码。
第10章 建立和使用模拟器建立和使用模拟器，可以帮助我们加快开发过程，通过提供一个可以安全试错的环境来检验思路，不仅针对控制软件，还可以针对可模拟的硬件提供一些宝贵的经验（非侵入式的）。
无论是因为目标硬件暂时未能提供，或太昂贵无法承担损坏的风险，都可以通过模拟器简便地令软件可以运行，进行测试改进，获得足够的信心，令它正常在现实世界中工作。
第11章 设备I/O在这一章中，我们将看看如何使用在第7章中介绍的物理接口，在真实世界和你的应用间搬运数据。
我们将从接口协议格式开始讨论一些基本概念，然后引入一些软件实例以介绍几个常用协议解析包：pySerial，pyParallel 和PyVISA。
最后，会展示一些技术来说明如何读取和写入数据到设备。
我们将对比阻塞与非阻塞的I/O 事件，以及如何应对潜在的数据I/O 错误，以便使应用程序更加健壮。
第12章 文件读/写本章谈及如何实施对设备文件的读/ 写以及审查，并对比了文件格式，从普通的ASCII 和CSV 文件到二进制文件数据库都进行了简述。
我们还将研究Python对配置文件的处理能力，看看它通过库方法有多么容易存储和检索配置参数。
第13章 用户界面除非应用程序是嵌入硬件或是专门设计作为后台运行过程的，否则可能都会需要某种用户界面。
本章考察命令行界面，或是用Python的curses模块生成的文字控制界面，以及如何使用ANSI 的终端仿真器程序显示数据，接受输入。
本章还包含如何用Python内置的TkInter模块来生成图形界面，并另外阐述了wxPython GUI 包。
第14章 真实实例我们将考察几种不同类型的设备如何进行数据采集和控制。

本章首先是一个捕捉连续的数据输出的数字万用表实例。
然后，考察一个采用普通的串行接口进行命令和数据交换的数据采集装置。
最后，我们将详细分析硬件供应商提供的一个API 接口，通过DLL 对USB 设备进行数据I/O 操作，借鉴较早章节的例子对常见设备完成几乎所有的关键操作，以展示如何将理论付诸实践。
最后是两个附录，包含了其他实用信息：附录A 自由及开源软件资源附录B 仪器设备资源本书的格式本书使用下列排版约定。
斜体Italic用来表示新术语、URL、电子邮件地址、文件名和文件扩展名。
等宽字体Constant width用来表示计算机代码片段，或者在文中引用Python 模块，以及变量名、函式名、数据类型、语句、关键字等程序元素。
等宽加粗字体Constant width bold用来表示命令或者需要由用户输入的文本。
等宽斜体Constant width italic用来表示应该由用户提供或特定语境确定的值替换的文本。
中文版书中切口处的“ ”表示原书页码，便于读者与原英文版图书对照阅读，本书的索引所列页码为原英文版页码。
代码示例的使用本书的宗旨就是帮助你完成工作。
一般而言，除非要原样引用大量代码，否则你可以在自己的程序和文档中随意使用书中的代码，而无须与我们联系以取得授权。
例如，在编写程序时引用本书若干代码片段是无须授权的。
然而销售或分发O ’Reilly 图书示例光盘则是需要授权的。
引用本书内容以及代码来答疑解难是不需要授权的，但是将书中的代码大量引入到你的产品和文档中则需要授权。
如果你在引用书中内容时注明出处，我们将不胜感激，虽然这不是必需的。
引用声明通常包含标题、作者、出版商和ISBN 编码。
Safari Books OnlineSafari Books Online 是一个按需出版数字图书馆，在这里可以轻松搜索到超过7500 种技术及创意类参考书和视频，快速得到要找的答案。
通过订阅，可以从这一在线图书馆浏览任何网页，观看任何视频；可以在手机等移动设备上阅读书籍；可以在产品付印前获得新书信息，优先一睹创作中的手稿并给作者提交反馈；可以复制并粘贴代码示例，组织收藏夹，下载章节，对重点部分添加书签，创建笔记，打印页面，以及从许多其他的省时功能中受益。
译者序Zoom.Quiet 代序本书是“ 侠少 ”（张春雨编辑）推荐给我翻译的。
只是因为之前组织过几本Python 相关技术图书的翻译工程，就成为所谓资深人士，进而被编辑盯上了。
可是拿到书一看，不是单纯讲Python 开发技巧，而是如同TBBT（《生活大爆炸》）片头曲背景MV 那般，内容包罗万象：从电子到仪表到线缆再到高端的软件工程都有所包含，内容的推进也是高速但清晰的，而且完全是根据学习的自然路径组织的，一步一步，从简入繁，自然而然，明明白白，很有Pythonic的感觉！
所以，俺就无耻地心动了。
再说，华蟒用户组（CPyUG）订阅人数近一万，其中肯定有软硬兼修的高人，俺只要作好大妈的本职工作，就可以向中国Python 社区贡献第一本硬件相关的好书了！
于是，革命的乐观主义精神主导了俺的情绪。
终于，在2011 年春节前，俺接下了本书的翻译组织任务！
最终图书署名为OBP Group（开放图书工作组），原因有三：本书的翻译出版过程沿用了OBP（Open Book Project，y 开放图书计划）的协同流程；大家的主要协同场景是在Google Group（邮件列表）中；工作组专门为本书成立，翻译完成后就地解散，转为通过Group 的形式存在，继续支持图书内容的讨论。
其中OBP 的工作流程细节请参考
：http://code.google.com/p/openbookproject/wiki/HowToBuildBookOnline目前，OBP 是作为一种开放的分布式协作流程而存在的。

任何人都可以使用OBP 实践检验过的在线分布式团队的组织形式来完成任意作品。
其实就是将软件工程管理的思想及工具，组合应用在图书创译方面而已。
相比其他翻译团队，特殊在协同的工具链。
1. 在Bitbucket.org 上使用Mercurial 分布式仓库，或是在Github/Gitcafe 中使用分布式仓库。
2. 基于Sphinx 来组织新结构化文本（rST）。
3. 通过readthedocs.org 随时编译为html 格式的图书式网站。
果然，通过邮件列表，快速报名上来十多位有意向的译者。
根据以往经验，三个和尚没水吃，人多时，大家都以为有人在翻译，自个儿就不动手了。
于是，以C语言/Python语言/硬件进行领域划分，俺单独负责所有Python及软件工程相关的内容，另外邀请四位译者分别承担其他两部分内容，再加上责任编辑，实际上形成了一个有专项目标的迷你社区。
那么什么是社区呢？
简单的说就是相对固定的成员在相对固定的场所对固定的目标进行固定的行动。
所以，技术图书的翻译团队，如果是分布式的，其实就形成了一个确切的社区。
只是，一般OBP专项社区随着图书的出版会快速转入静默，不是图书内容没有值得持续讨论的地方，而是大家不愿意回忆那些熬夜翻译赶进度的苦吧。
虽然原书有600多页，翻译为中文，也就500页的样子，其中还包含大段的代码，平均到每个人也就100多页，几万字而已。
相比动辄几百万字的网络小说来，大家的工作量不算大。
但是，技术文字的翻译，不是口水化的玄幻小说，至少要做到：原文内容自个儿看得懂翻译出来的中文自个儿看得明白翻译出的文字任何人也看得明白而且，近20个月，大家并不是全职翻译，都是从学习/工作/生活中挤出时间来义务翻译的。
常常要花很大的精力，重新熟悉翻译过的内容再跟上作者思路，找到合适的表述方法来。
这是个反复的，不断自我折腾的过程……所以，进度的控制就完全失控了，好在此失控是K.K所说的能自我调节的“ 失控 ”。
有负责的催生婆——责任编辑一直在执行田间管理，译者们只要根据协同约定，调整好状态，按时完成章节的翻译，hg push到封闭仓库中，所有相关成员就可以同步到变化继续了。
从意动到交付初版的这20个月里，个人来说发生了很多事儿，很chaos。
但是，能在社区还有编辑的支持下拿出成书来，真的是极大的安慰。
只是，大家心里又随着不断的review中发现的各种小问题，而酝酿出怀疑是否将书中丰富的软硬件知识翻译到位了的惶恐。
然而，只要有好书，值得翻译的书，硬着头皮也还是愿意继续惶恐的！
最后要强调的一点是，本书的OBP Group所有成员，从来没有坐在一起交流过，直到最后签约阶段，才在北京同其中一半的成员面对面吃了次准庆功宴。
感谢网络以及开源工具的力量，更要感谢所有成员给力的大爱，使我们能又见到一本好书的面市！
以下是所有工作组成员的简述，再次感谢所有成员，如果内容方面有什么问题，责任肯定在我们这群自不量力的行者身上，欢迎大家通过任意方式告诉我们，我们会坚持修订书中的内容的！
开源电子硬件和自由软件爱好者。
几年前得知外婆打电话不便，遂决定制作一部特殊的照片拨号电话机，经过不断自学和实践，最后做出facephone 解决了问题，从此成为一名硬件hacker。
浏览过图书的目录后，就发现这是一本难得的好书，它的内容和我的工作息息相关，于是决定加入翻译团队。
我比较熟悉的是书中的电子学部分，于是一开始就着手相关的翻译，然而在翻译的过程中发现自己原本以为懂得的东西实际上很多都不懂，有时为了弄清楚一个知识点，竟然会费几天的时间去学习。
这样做是因为我自己就是一个技术书籍的读者，如果翻译得不准确对读者是不负责的，对这本O'Reilly的图书更是不敢怠慢，我想要把自己翻译的部分翻译得尽量准确易懂。
在边学边译的过程中，我重新学到很多知识，这是参与翻译过程的一个很大的收获。

由于是业余时间翻译，在时间上会很不够用。
有时要等到老婆孩子睡觉之后才有时间坐在电脑前一边读一边译。
有一段时间我感觉自己对翻译工作失去了信心，后来多亏“大妈”和张编辑的号召才又回到工作状态，完成了自己的翻译任务。
书译完了，女儿也从几个月长到了两周岁。
如果说要感谢，我要感谢我的女儿艾达和妻子小霞，很多本应要陪她们的周末和夜晚都被我用来翻译了。
我是一个码农，最早是在网上认识“侠少”的，后来在组织珠三角技术沙龙时和他有过一次合作。
这次是作为“救火队员”被拉到这个团队的。
本书是关于使用Python开发现实世界中的仪器软件的，我对Python本身不算擅长，只是入门级别，好在自己翻译的这章涉及语言本身的内容很少，这也是我敢应承下来的原因之一。
翻译所谓“信达雅”，达到任何一个都不容易，反复读，仔细揣摩作者的意思，然后翻成中文，又要反复读，反复改，让语言更加通顺，更加贴合作者的原意。
此中辛苦，确实不经历过就没法体会。
我是最后阶段才加入，所以对于团队感觉并不是很深切，基本上也和“大妈”、“侠少”有过交流。
这种方式应该说有其优点，但是如何找到合适的人，如何协调好，还是件不容易的事情。
在SHLUG邮件列表上看到“大妈”吼，然后发现这本书很有意思，并且自己也了解一部分相关知识，于是申请加入了翻译团队。
这本书的内容还是不错的，所以翻译的过程也是一个学习的过程，学习就有快乐。
而且有时为了一句话反复思考半天，最后忽然想到“啊，原来可以这样翻译！
”的时候更是快乐……其实个人一直很向往Github那种异步的工作方式，所以加入到OBP后还是很兴奋的。
但是现实是残酷的，大家都分布在全国各地，时间和精力上并不完全一致，而且又都是利用空余时间做翻译，翻译又不是自己的专长。
虽然大家都是怀了一颗奉献的心来的，但是有时毕竟会出现心有余而力不足的状况。
所以整个翻译的进度就受到了影响。
在此仅代表个人向编辑道个歉，感谢编辑长久以来的宽容和支持。
OBP也可以总结这次合作的经验教训，为以后的项目积累经验。
同时，在这本书的翻译，尤其是最后的审校过程中还是暴露出了现在传统出版行业和“互联网公司”型团队的衔接问题，包括版本控制流程、译文格式排版等。
个人对出版行业了解不深，所以这个话题还是留给专业人士展开吧。
这次是我第一次参与翻译工作，出现各种疏漏在所难免，还请大家多多批评指正。
主要从事Linux下的嵌入式软件开发工作，4年工作经验。
由于本人经验和时间所限，发布之际还是诚惶诚恐，担心工作中有不足和失误之处，万望大家发现问题多多批评，指正，谅解！
sa xiao远程松散的协作方式好处是很灵活，每个成员可以根据自己的情况自由安排工作时间，加上主导者“侠少”、“大妈”制订的比较完善的流程，使得这样的工作方式可行。
但大家都是在有本职工作或者学业的情况下自觉自愿地工作，且没有硬性的外在约束，这也加大了协调的难度，延长了工期。
一方面，工作规则和制度需要更完善；另一方面，随着更多项目的完成，成员的磨合度提升，效率自然会提高。
关于书的内容，切入点很好，翻译起来有一定的难度，其他的让市场说话吧。
孙伟 学习使用C语言很多年以后，喜欢Python也有一两年了，有幸参与到本书的翻译当中，感觉是一个非常好的学习和锻炼。
C语言靠近底层系统，简洁直接，长久不衰；Python语法清晰明快没什么花活：这两门语言都是我喜欢的。
本书介绍用Python 进行工业控制方面的编程，是Python 书籍中比较少见的，感兴趣的朋友可以读读看

。
第4 章和第5 章的翻译，断断续续花了我将近两个多月时间。
虽然这两章不是很复杂，也没有什么生僻的内容，但是当自己把一句句英文翻译成中文的时候，我才发现这个事情不是想象中那么容易。
有些句子要反复斟酌，有些术语英文好懂，翻译成中文后却有点怪怪的，需要多花时间找到恰当的表述方式，对我自己算是一个很好的锻炼。
由于水平所限，想必还是有很多问题，欢迎大家指正。
xiaoma开源软件爱好者和程序员。
小学时在少科站老师的指导下组装过收音机，中学时开始接触Apple II 电脑上的Logo编程，后逐渐走上了软件开发这条不归路。
这已经不是我第一次参与技术图书的翻译，虽然少了些许初夜般的兴奋，但仍然保持着诚惶诚恐的态度，唯恐糟蹋了这一本好书。
由于本人才疏学浅，因此难免挂一漏万，不足之处请大家多多指正。
想到本书的出版，能为读者提供学习和工作上的帮助，所有的吐糟都化为了青烟。
在此要感谢“ 大妈 ”的奔走张罗和译者、编辑们的辛苦工作，使得本书的中译本能顺利出版。
Zoom.QuietPython中文社区创始人/管理员之一，热心于Python社区的公益事业，大家熟知的社区“ 大妈 ”，OBP及蟒营工程设计者/主持人，参与/组织/主持各种线上/线下活动，主持编撰了《可爱的Python》，坚持用Pythonic 感化国人进入FLOSS 社区进行学习/分享/创造！
本书是从纯软件跨入软硬天师的最好引导，强烈建议准备或是已经在玩开源软件的行者置备！

<<真实世界的Python仪器监控>>

内容概要

《真实世界的Python仪器监控:数据采集与控制系统自动化》介绍借助Python编程来实现对数据的自动化收集和处理，书中介绍的技巧除了可以应用于软件项目管理、办公自动化和科学研究，也可以用来对工业生产中对电子设备、机械设备进行自动化管理和监控。
《真实世界的Python仪器监控:数据采集与控制系统自动化》属于Python应用的中高端选题，市面上找不到同类书。

<<真实世界的Python仪器监控>>

书籍目录

章节摘录

版权页： 插图： 为什么需要需求 没有需求或没有工作说明书，就无法对最终的结果形成清晰的思路，也无法明确如何行事。
当然，有时系统应该做什么只是一些模糊的概念，模糊的概念终究是模糊的，肯定是不和其他人的模糊概念相同的。
午餐期间，餐巾纸上的涂鸦不是需求；会议期间，草草记在白板上的也不是。
类似的东西都不是明确的需求，不仅是因为它们不符合可核查的标准，而且还因为它们只是些浪花儿，而实际上其下面可能是非常深的海沟。
即使是最小的需求集合，也足以引导我们定义出明确和具体的实现要求。
那种可以令所有的人认同的就是好需求，同时，好需求还可以对成品的成败给出明确的测量。
一则真实的需求故事 很久以前，我们负责为一个大型的射电望远镜系统的滤波器阵列建立实时数据采集系统。
该滤波器阵列是旧设备，其存储单元甚至要人工拖出来，而且有640个独立通道，各自对应到一个特定的频率。
我们的工作是将高速模拟数字转换器和旧的滤波器阵列系统整合起来，通过实时操作系统获得每个通道的数据，射频工程师（RFengineer）也时刻准备好了，一切都很靠谱，直到我们向滤波器阵列负责人咨询有什么特殊需要时，开启了一段奇幻之旅……我们问：“每个通道的转换时间最大可容忍多少？
”回答：“嗯嗯嗯，尽可能快”，这明显是不靠谱的，所以我们尝试换种方式再问，仍然收到模糊的回答。
一来一去，就这样至少30分钟，我们越来越沮丧，而其他人越来越恼火，认为我们只是企图制造一些数字。
最终我们总算挤出了一点儿基本的数值来开展工作，但仍然不得不用乱猜的方式来弥补一些“失踪”的数值。

媒体关注与评论

“O’Reilly Radar 博客有口皆碑。
”——Wired“O’Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。
”——Business 2.0“O’Reilly Conference 是聚集关键思想领袖的绝对典范。
”——CRN“一本O’Reilly 的书就代表一个有用、有前途、需要学习的主题。
”——Irish Times“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。
”——Linux Journal

<<真实世界的Python仪器监控>>

编辑推荐

《真实世界的Python仪器监控:数据采集与控制系统自动化》可帮助读者了解如何通过自行开发应用程序来监视或控制仪器硬件。
无论是想从设备中采集数据，还是进行自动化控制，这本实用图书都将向你展示——如何利用Pyhton的快速开发能力，实现从接线到建立接口，直到完成可用软件的整个过程。
《真实世界的Python仪器监控:数据采集与控制系统自动化》提供逐步讲解、清晰实例，以及将PC连接到各种设备的实践技巧。

<<真实世界的Python仪器监控>>

版权说明