

硬件开源
电子设计平台

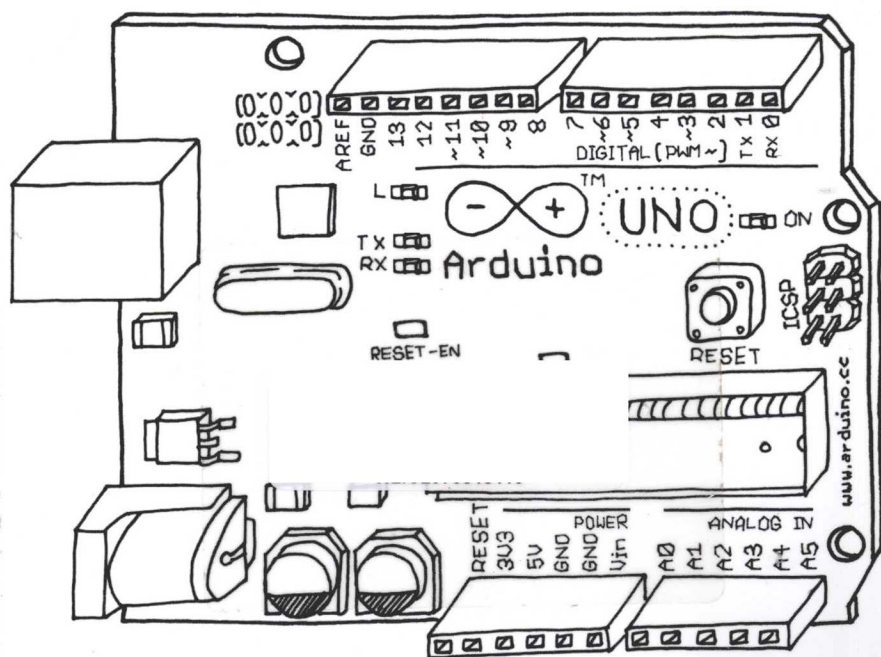
爱上 Arduino

(第2版)

Getting Started with Arduino (2nd Edition)

[美] Massimo Banzi 著

于欣龙 郭浩赞 译



人民邮电出版社
POSTS & TELECOM PRESS

O'REILLY

Make:
makezine.com

爱上Arduino

(第2版)

[美] Massimo Banzi 著

于欣龙 郭浩赞 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权人民邮电出版社出版

人 民 邮 电 出 版 社

北 京

图书在版编目 (C I P) 数据

爱上Arduino / (美) 班兹 (Banzi, M.) 著 ; 于欣龙
 , 郭浩赞译. — 2版. — 北京 : 人民邮电出版社,
2012.10

ISBN 978-7-115-28002-2

I. ①爱… II. ①班… ②于… ③郭… III. ①单片微
型计算机 IV. ①TP368.1

中国版本图书馆CIP数据核字 (2012) 第065923号

版权声明

Copyright ©2012 by Massimo Banzi

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom
Press, 2012. Authorized translation of the English edition, 2011 O'Reilly Media, Inc., the
owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2012。

简体中文版由人民邮电出版社出版 2012。英文原版的翻译得到 O'Reilly Media, Inc. 的
授权。此简体中文版的出版和销售得到出版权和销售权的所有者 —— O'Reilly Media,
Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

爱上 Arduino (第 2 版)

-
- ◆ 著 [美] Massimo Banzi
 - 译 于欣龙 郭浩赞
 - 责任编辑 宁茜
 - 执行编辑 马涵
 - 封面设计 Brian Scott 马冬燕
 - ◆ 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>
 - 北京隆昌伟业印刷有限公司印刷
 - ◆ 开本: 690×970 1/16
 - 印张: 7.25
 - 字数: 105 千字 2012 年 10 月第 2 版
 - 印数: 5 001 – 8 500 册 2012 年 10 月北京第 1 次印刷
 - 著作权合同登记号 图字: 01-2012-1723 号

ISBN 978-7-115-28002-2

定价: 38.00 元

读者服务热线: (010)67132837 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

目录

作者絮语	1
致谢	4
1/介绍	5
目标读者	6
什么是Physical Computing?	6
2/Arduino理念	8
原型	9
Tinkering	10
Patching	11
改装电路	13
改装键盘	15
我们喜欢变废为宝!	17
改装玩具	18
合作	19
3/Arduino工作平台	20
Arduino硬件	20
Arduino集成开发环境 (IDE)	23
在你的计算机上安装Arduino (IDE)	23

安装驱动程序：Macintosh操作系统下的方法	24
安装驱动程序：Windows操作系统下的方法	24
识别通信端口：Macintosh操作系统的情况	25
识别通信端口：Windows操作系统的情况	26
4/Arduino入门	28
解析互动装置	28
传感器与驱动器	29
LED闪烁	29
编写程序	31
给我个奶酪（Parmesan）	33
Arduino从不停止	34
真正的Tinker都写注释	34
代码，一步一步来	34
我们将会做什么？	37
什么是电？	38
使用按钮控制LED灯	40
它是如何工作的？	42
一个电路，一千种用法	43
5/ 高级的输入/输出控制方法	48
尝试其他开关类型传感器	48
使用PWM方式控制灯光亮度	51
使用光线传感器取代按钮	57
模拟输入	58
尝试其他模拟传感器	61
串行通信	62
驱动较大功率负载设备（直流电机、灯泡等）	63
复杂传感器	64
6/互动云	65
制订计划	67

编写程序源代码	68
组装电路	73
下面介绍如何安装.....	74
7/排疑解惑	76
测试板子	77
用面包板测试电路.....	78
将问题独立出来	79
开发环境（IDE）常见问题.....	79
利用网络资源解决问题	79
附录A/面包板	83
附录B/认识电阻和电容	85
附录C/Arduino语法参考	87
附录D/阅读电路简图	101

作者絮语

几年前我接受了一项非常有趣的挑战任务：教设计师们一些初步的电子入门知识，然后让他们为设计出的物件构建互动原型。

我下意识地根据我当年上学时老师教书的方式开始教他们电子电路知识。不久之后我就发现，教学效果并没有我希望的那么好。我开始回忆起我上学那会，坐在教室里感觉糟透了，所有理论批头盖脸向我灌来，但没有实际动手去操作。

实际上，我在入学之前已经通过自己实验的方式学到了很多电子电路的知识：虽然只有非常少的理论，却积累了很多动手的经验。

我开始思索我学到那些知识的过程。

- » 我拆开那些我能拿在手上的小型电子设备。
- » 我慢慢地学习认识那些电子元器件。
- » 我开始调试它们，改变一些内部连接，看它们会发生什么变化：通常不是爆炸就是冒烟。
- » 我开始搭建一些电子杂志销售的小套件。
- » 我组合那些我改装过的设备、套件以及其他的电路，让它们变成具有新的用途的东西。

作为一个小孩，我对于探索事物如何运作非常着迷。因此，我常拆分它们。这样的热情使我逐渐把目标定在那些家中闲置的东西，把它们拆成很小的部件。

最后人们纷纷把自家东西拿来给我拆卸。当时我最感兴趣的是一台洗碗机和从一家保险公司拿来的计算机，计算机附带一个巨型打印机，还有电子卡片、磁卡读

卡器和一些其他零件，要彻底拆开也是个巨大的挑战。

拆过很多东西以后，我基本知道了电子元器件是什么和它们的大概作用。加之我爸爸可能从20世纪70年代初就开始购买电子杂志，堆在家里到处都是，我每天都花几个小时去阅读上面的电路图，尽管似懂非懂。

一遍又一遍地读文章，又常常拆各种东西，这两者渐渐形成了良性循环。

有一年的圣诞节对于我又是一个巨大的飞跃。这一天我爸爸送给了我一个工具箱，里面有帮助青少年学习电子元器件的很多工具，每一个工具都装在小盒子里，小盒子上带有磁性小方块儿，能够与其他盒子相连接，顶端标着各自的电子符号。我当时不知道这是德国的标志性产品，是由Dieter Rams在20世纪60年代设计的。

有了这套新工具，我很快学会了怎样组装电路并试用它们，建立模型需要的时间也越来越短了。

之后我自己制作了收音机、扩音器，还有的电路能够发出巨大的噪声也能放出美妙的音乐，我还设计了雨天感应器、小机器人。

很久以来我都想找到一个英语单词来形容这种工作方式：没有特别的目的，从一个模糊的想法开始，得到一个完全意外的结果。后来发现了“tinker”这个词，我注意到这个词在很多其他领域都被用来描述某种操作方式，也用来描述那些探索的人们。例如，法国创造了“Nouvelle Vague”的那一代导演就被人们称为tinkers。我找到的tinker的最佳定义来自旧金山的探索展馆的一次展览：

“tinkering就是你开始做一件不怎么确定的事情的过程。只由灵感、创意、想象力和好奇心指引着，没有操作规则，也就没有失败，没有正确和错误。整个过程都是在观察事物的情况并不断修整它们。”

新发明、小装置，各种风马牛不相及的东西和谐地工作，这就是tinkering。

从最根本上来说，tinkering就是探索和娱乐的结合。

—— www.exploratorium.edu/tinkering

我从早年的实验中明白了能够用最基本的元件设计一个电路，让它按你的想法工作，需要积累多少经验。

1982年的夏天我迎来了另一个突破。在伦敦，我和父母一同参观了科学博物馆。那里新开了一个侧厅，展示计算机相关的展品。在有引导的实验中我大概了解了二进制数学和编程基础。

在那儿我明白了工程师已经不再用基础元器件设计电路，而是使用微处理器在很多产品中添加智能。软件为电子设计节约了大量时间，也使得tinker的过程周期越来越短。

回来以后我就开始攒钱，因为我想买台计算机学习编程。

我的第一个工程也是最重要的项目就是用我那台崭新的ZX81计算机控制一台焊接机。我知道这听上去没什么意思，但是因为当时有这个需要，而且因为我刚刚学习编程，这对我来说也着实是个挑战。这时我明白了写代码比不停地测试复合电路方便得多。

20多年过去了，我觉得这种经历可以教会那些连数学课也没上过的人把热情融入到tinker的过程中去，就像我少年时和那以后一直保持的一样。

—— Massimo

致谢

此书要献给 Luisa和Alexandra。

首先我要感谢Arduino团队的伙伴们：David Cuartielles、David Mellis、Gianluca Martino和Tom Igoe。与你们合作的经验真的非常棒。

Barbara Ghella可能并不知道，如果没有她宝贵的建议，也许就不可能有Arduino和这本书。

Bill Verplank教了我更多Physical Computing以外的知识。

Gillian Crampton-Simth给予我这个机会从她身上学到许多东西。

Hernando Barragan在Wiring上投入了许多精力。

Brian Jepson是位称职的编辑，也是本书一直以来热情的支持者。

Nancy Kotary、Brian Scott、Terry Bronson和Patti Schiendelman把我所写的内容更流畅地呈现出来。

我还想感谢更多人，不过Brian 说我已经没有多余的空间了。以下是一小部分我特别想感谢的人：Adam Somlai-Fisher、Ailadi Cortelletti、Alberto Pezzotti、Alessandro Germinasi、Alessandro Masserdotti、Andrea Piccolo、Anna Capellini、Casey Reas、Chris Anderson、Claudio Moderini、Clementina Coppini、Concetta Capecci、Csaba Waldhauser、Dario Buzzini、Dario Molinari、Dario Parravicini、Donata Piccolo、Edoardo Brambilla、Elisa Canducci、Fabio Violante、Fabio Zanola、Fabrizio Pignoloni、Flavio Mauri、Francesca Mocellin、Francesco Monico、Giorgio Olivero、Giovanna Gardi、Giovanni Battistini、Heather Martin、Jennifer Bove、Laura Dellamotta、Lorenzo Parravicini、Luca Rocco、Marco Baioni、Marco Eynard、MariaTeresa Longoni、Massimiliano Bolondi、Matteo Rivolta、Matthias Richter、Maurizio Pirola、Michael Thorpe、Natalia Jordan、Ombretta Banzi、Oreste Banzi、Oscar Zoggia、Pietro Dore、Prof Salvioni、Raffaella Ferrara、Renzo Giusti、Sandi Athanas、Sara Carpentieri、Sigrid Wiederhecker、Stefano Mirti、Ubi De Feo、Veronika Bucko。

1/介绍

Arduino是一个开源的、拥有简单输入/输出 (I/O)的电路板，它沿用了Processing语言的开发环境。Arduino可以用来开发独立运作互动装置，或者可以连接到你计算机上的软件（例如：Flash、Processing、VVVV或Max/Msp）。你可以自己动手组装这个电路板，或者直接购买套件；开源IDE（集成开发环境）可以免费从www.arduino.cc下载。

Arduino和目前市面上其他平台相比，有以下特点：

- » 支持多种操作系统：Windows、Macintosh、Linux。
- » 为了方便设计师以及艺术家的使用，采用Processing的集成开发环境。
- » 程序是通过USB而并非串行端口实现的。这一点非常实用，因为现今的多数计算机都没有串行端口。
- » 这是开源的硬件和软件——如果你愿意，可以直接下载电路图，购买所需的电子元器件自己制作，无须从Arduino制造商那里购买。
- » 硬件很便宜。USB电路板只要20欧元（译者注：差不多35美元，目前在中国所有的元器件成本仅需65元人民币左右），如果只是替换电路板上烧坏的微处理器，那么只需5美元（译者注：在中国是20元人民币左右），所以就算做错了，烧坏了部分元器件，也不用心疼。
- » 有一个活跃Arduino玩家的论坛，你可以随时找人帮助你。

» Arduino Project是以教育为宗旨而开发的，因此就算是初学者也能很快地入门。

这本书的目的是帮助初学者理解如何使用Arduino平台并培养自我学习精神。

目标读者

这本书一开始就是针对设计师和艺术家而编写的。所以，书中解释和叙述的方式肯定会让一些工程师无法接受。事实上，这章节的一个草稿版已经被某位工程师形容为“看得发毛”。不过那也正是作者我的目的。工程师自己也常常无法向其他工程师解释自己的逻辑和做法。让我们现在就开始深究所谓的“发毛”吧！

注意

这样的方式，以及某些“有争议”的设计，也可延伸为利用科技来创造原型，尤其是电子科技的原型。

当Arduino开始盛行时，我意识到有很多实验者、兴趣爱好者和黑客已经利用它来进行各式各样的非常漂亮和疯狂的创作了。我相信你们都是艺术家和设计师，所以这本书适合任何人。

Arduino是为了传授互动设计而诞生的，把设计原型的能力定为使用者的学习目标。关于互动设计有很多不同的说法，但我偏爱下面的定义：

互动设计就是设计互动体验。

如今，互动设计意味着人与物体间具有深意的体验。它是一种探索科技美以及人对现代科技之体验的创作。互动设计鼓励一种不断制作原型并由此使创作逐步完善的设计过程。这样的方式也可引申阐释为：利用科技来创造原型。更具体地说，利用Arduino做互动设计称为物理计算（Physical Computation）或者物理互动设计（Physical Interaction Design）。

什么是Physical Computing?

Physical Computing利用电子零件为设计师和艺术家的新素材制作原型。

这包括了用传感器和驱动器设计出与人互动的装置，并且通过微处理器（单片机）上的软件来控制整个互动装置。

在过去，一说到要处理有关电子方面的事情，总是会让人想到去把工程师找来，把多个小元器件拼凑成整个电路——这些问题使得做创意的人始终无法进入电子领域。大多数的工具都是为工程师设计的，并且想使用这些工具也需要很多相关知识。近几年来，微处理器变得更加便宜和更容易使用，也衍生出了很多实用的工具。

我们做的Arduino就是为了让初学者经过两三天学习就可以开始制作装置；而艺术家和设计师们，也可以通过Arduino很快地学习到电子电路以及传感器的基本知识，用很小的花费就可以制作出设计原型。

2/Arduino理念

Arduino的理念就是不要光说不练，要多动手做。我们不断追求更快更有利的方法来制作原型，利用双手探索更多的原型技术，并开发多方位的思维。

典型的程序思考模式偏向于单向思考，例如，如何从A到B；Arduino采用多方位思考模式，有时也许会迷失方向，也有可能发现新的目的地C。

把玩所有工具，找出意想不到的结果，这就是我们崇尚的Tinkering模式。在探索制作更好的原型的过程中，我们选择一些套件，以便我们随时调用整块硬件和软件。

接下来的几个章节会介绍由Arduino理念引出的想法、事件和开拓者。

原型

原型是Arduino学习的核心：

我们制作东西，与其他物件、人和网络进行互动。我们找寻更快、更简单、更快速的方式来制作尽可能便宜的原型。

很多电子知识的初学者以为他们必须从头学起，其实不必如此费力：其实你真正需要的是制作出能够快速看到成果的东西以激发你前进的动力，或者成果能激发他人投资你的产品。

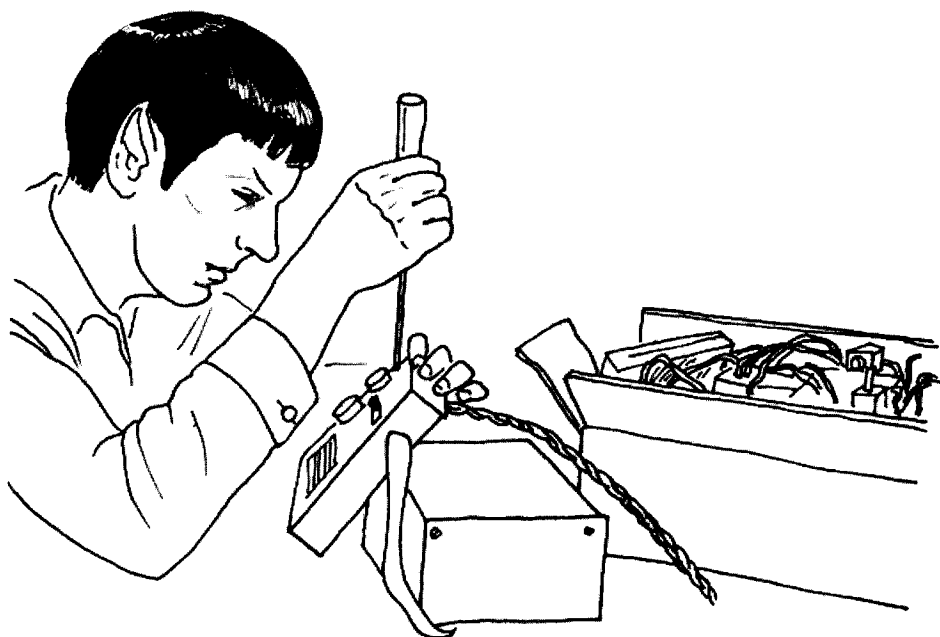
这就是我们要发明“取巧原型制作法”（opportunistic prototyping）的原因：当我们可以改装一个现成的由大公司和优秀工程师们开发的精密设备时，我们或许不必花费大量精力和时间从零开始。

譬如说，大师James Dyson尝试制作了5127种吸尘器原型，才做出满意的成品(www.international.dyson.com/jd/1947/asp)，我们可以从他的成果中学习到不少经验，而不需要再做那么多次。

Tinkering

我们相信把技术娱乐化是很有必要的，在这个过程中可以亲身探索硬件与软件的可用性——有时并没有非常明确的目的。

从现有的技术中汲取营养是Tinkering的最佳方式。我们可以使用一些便宜玩具或者废弃设备，尝试改造它们并做出全新的东西。



Patching

我对于使用简单的模块构建复杂的系统很感兴趣，最具代表性的作品就是Robert Moog制作的模拟电子音乐合成器。音乐家利用各种方式来连接不同的模块，从而编制出新奇的声音，这样的方式虽然让合成器看起来像是电话配线箱，但如果与一些旋钮结合，就能成为创新音乐平台。Moog形容这是一种“探索和见证”的过程。我相信大多数音乐家一开始都不懂这上百个旋钮的作用，但他们会不断地去尝试，找出适合自己风格的音乐。

减少创作过程的中断次数对于整个创作流程是非常重要的。没有太多障碍和开发过程将有利于激发创作的欲望。

这样的编程概念在软件中被称为“视觉化编程”环境，例如MAX、Pure Data 或者 VVVV。这些工具可以把不同的功能包装成“方块”，使用者可以按照自己不同的需求将方块连接起来。这些软件让使用者不须要被写代码所困扰，同时也能够体验到写代码的乐趣。如果采用传统编程方式，常常是一个令人受挫折的过程：编程，编译，该死！——出现错误，找错，重新编译，再执行。如果你的思考逻辑比较视觉化，建议你可以尝试使用上述工具。