

Physical Computing

交互式系统 原理与设计

Dan O'Sullivan

Tom Igoe

著

张瑞萍 等 译



清华大学出版社

TP11
55

交互式系统原理与设计

Dan O'Sullivan & Tom Igoe 著

张瑞萍 等译

清华大学出版社

北 京

Dan O'Sullivan and Tom Igoe

Physical Computing

EISBN: 1-59200-346-X

Copyright © 2004 by Thomson Course Technology PTR, a division of Thomson Learning.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All Rights reserved.

本书原版由汤姆森学习出版集团出版。版权所有,盗印必究。

Tsinghua University Press is authorized by Thomson Learning to publish and distribute exclusively this Simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本中文简体字翻译版由汤姆森学习出版集团授权清华大学出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可,不得以任何方式复制或发行本书的任何部分。

981-254-903-X

北京市版权局著作权合同登记号 图字 01-2005-3315 号

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

交互式系统原理与设计/(美)奥沙利文(O'Sullivan,D.), (美)依葛(Igoe,T.)著;张瑞萍等译. —北京:清华大学出版社,2006.7

书名原文:Physical Computing

ISBN 7-302-12705-0

I. 交… II. ①奥… ②依… ③张… III. ①人-机系统-理论②人-机系统-系统设计 IV. TP11

中国版本图书馆 CIP 数据核字(2006)第 022078 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 冯志强

文稿编辑: 林晴佳

印 刷 者: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 24.5 字数: 609 千字

版 次: 2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷

书 号: ISBN 7-302-12705-0/TP·8110

印 数: 1~3000

定 价: 45.00 元

前言

计算机革命使得我们很多人都已经落伍。在建立 Apple Computer 公司时, Steve Jobs 有着类似的看法, 因此他希望为“其余的我们”构造计算机。这个想法使很多不是计算机专家的人们(比如艺术家、教育家和孩子)可以利用计算技术的力量。Apple 使用的图形用户界面(GUI)已经成功地广泛应用、复制, 并且是几乎所有个人计算机的标准界面。多亏了这个界面, 才使得各行各业的人们都可以使用计算机。

现在我们需要为其他人制造计算机。我们需要计算机响应身体的其余部位和世界。GUI 技术使你可以拖放, 但是它不会注意到你是否扭动或喊叫。它可以轻易地打开文件夹和启动程序, 但是我们想让计算机可以开门或启动汽车。办公室环境下的计算机已经发展到你坐着, 只需挪动手指就可以输入和接收意识中的信息。这不是你的全部生活, 也许不是最好的那部分。我们需要这样的计算机, 它们可以更多地感觉你的身体, 在更多地方为你服务, 除了传递信息外还可以传达肢体语言。

在纽约大学的 Tisch 艺术学院从事物理计算教学二十余年, 我们发现有着各种背景的很多人试图在现实与虚拟世界之间建立桥梁。也许你是个雕刻家, 希望通过触摸雕刻的不同部位, 可以显示不同的声音或图像, 或者你是个舞蹈家, 想要通过弯膝动作引起响铃。也许你是个社会学家, 需要自动记录经过某条街道的人数。或许你是位教师, 想要制造工具, 来使学生通过做而不只是读来了解世界。或者你只想在炎热的下午让你的窗帘自动落下。不管你有什么背景或技术经历, 本书都将帮你在物质世界和计算技术世界间建立更有趣的连接。

我们对计算机的认识

当绘制一台计算机时, 多数人绘制的元件都一样: 屏幕、键盘和鼠标。当我们想到计算机时, 这是我们脑海中的图像。为了全面探索计算技术的可能性, 你必须丢弃对计算机的老套看法。你必须想到计算技术, 而不是计算机。为了适应我们对计算的需要, 计算机应该具有相应的形状。那么什么是计算技术?

一个常见的答案是, 计算技术类似于人的思考。人工智能领域从一开始就在计算机科学中占据重要地位, 通过使用计算机模仿, 人工智能将来也许会取代人类。机器人科学是人工智能(AI)的物理实现。本书讲述的技术与机器人科学很相似, 但通常应用不同。在机器人科学中, 人们建造机器人来模仿人类的行为。我们并不反对机器人, 但是我们发现最有趣的机器人也不如最无聊的人。我们的方法来源于称为智能增强(IA)的不同计算技术领域。这种方法寻找人们提供对计算机的兴趣火花来捕捉和传达一个人的表达方式。我们试图支持人类的行为, 而不是模仿其行为。IA 把计算机作为人们之间通信的媒介。

那么, 作为媒介的计算机提供了什么? 它可以存储声音和图像, 但是以前的媒介(如磁

带和胶片)也可以做到这点。用胶片和磁带,必须根据位置按顺序调用声音和图像。因此,我们把磁带和胶片称为线形媒介。计算机提供了打破线形的方法。通过随机存取介质,可以随意调用计算机内存中的非连续部分,就好像它们彼此相邻一样。这样可以把思想记录在内存中,就像与其他思想相邻一样。当把随机存取和网络通信结合起来时,就可以显示存储在不同位置的信息和图像,就好像它们是相邻存储的。当重新梳理思想时,计算机减少了时间和空间的障碍。这样就可以更好地描绘思想之间关系的变动和多样性。在给予这些关系的多种版本声音时也会更公平。

即使你不想通过消灭时间和空间来拯救世界,计算机化的媒介也提供了很多好处。没有计算机,你可以连接某个按钮,通过按下该按钮来打开灯光。有了计算机,你可以使按钮与灯光的关系更复杂化。例如,你可以通过确定按下按钮的次数、按下的时间长度,或者是否与房间中的其他按钮一起按下来打开灯光。还可以改变这种关系,例如,在白天按两下按钮打开灯光,在晚上按一下打开灯光。为了使计算机理解这些关系,你需要编写计算机程序。智能增强方法依靠人类来制造最有趣的关系,所以物理计算技术的程序往往相对简单。

计算机对我们的认识

如果想让计算机支持人类(而不是以其他方式),必须观察这个人及其所处的环境,来确认需要的支持。那么,对于计算机,人看起来是什么样的?带着这个问题,你很可能觉得奇怪。为什么要在乎这个?计算机对人类的图像是由输入和输出设备决定的。对于多数台式机,这意味着鼠标、键盘、显示器和扬声器。对于这样的计算机,我们也许是一根手指,上面有一只眼睛和两个耳朵(参看图 0-1)。要想改变计算机对我们的反应,我们必须改变它对我们的认识。

通过计算机输入设备看到的人类是可怜的动物。Kurt Vonnegut 写的 *The Sirens of Titan* 中的 Tralfamadorians 看起来就很像这个,它们的侧面图与这个动物一样,与我们很不同。它不能走路、跳舞或跳跃;不能唱歌或尖叫。

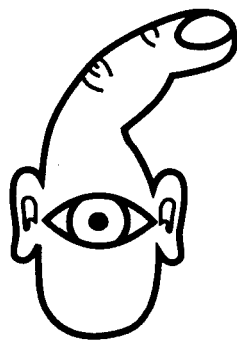


图 0-1 计算机对我们的认识

在发明新型计算机之前,需要理解为什么需要使用新形式。我们需要更好地认识自身来查看全部表情。这包括一切,从自发的面部表情到受过训练的演员的表情。即使是在站立这个过程中,每一秒的下意识调整动作都会暴露很多重要细节。即使这些表情持续的时间很短,人类也有惊人的洞察力来读取这些肢体语言中的内在含义。要使计算机成为表达媒介,你需要描述与计算机(或通过计算机)进行的对话。例如,在 Web 聊天室中,表达语境——就是用户姿态——是否符合聊天文本?你还需要检查环境。当你离开旋转椅时生活是否继续?计算机能否解释这个动作?人们喜欢用脚投票吗?如何记录他们的投票?完成了这些步骤后,你可以更多了解计算机的物理潜能,以及人类的潜能。

概 念

我们应该首先简要介绍整本书中反复出现的一些关键概念。物理计算技术是关于在物质世界与计算机虚拟世界之间建立对话的科学。变换就是把对话从一种能量形式转换为另一种,使这些对话成为可能的过程。你的工作是发现、学习并利用变换器把工程的物质能量转换为计算机使用的电能。为了更好地解决这个问题,首先把能量流动标识为输入和输出,然后单独处理每个问题。你将学习到能量流动中的信号可以显示为数字或模拟。标识出流动方式有助于弄清你建立的交互和缩小变换的范围。标识流动中事件发生的时间,不管是串行还是并行,有助于确定最好的交互计划。

交互:输入、输出和处理

当谈论计算机时,人们经常说计算机有用是因为它们使事情具有交互性。“交互”是个模糊的术语,经常被很多可笑的目的误用。作家和游戏程序员 Chris Crawford 的定义很好:交互是两个或多个演员之间听取、思考和交谈的往返过程。多数物理计算工程(包括多数通用计算机应用程序)可以分解为相同的三个阶段:听取、思考和交谈——或用计算机术语:输入、处理和输出。把你的工程按这个方式分解,可以使你更集中于某个挑战,甚至可以跳过本书的整节内容。在第 8 章,我们将讲述事件这三部分的循环,以令人满意的方法创建交互,就像交互对话。

输入

对于很多人而言,输入是他们想从物理计算学到的所有内容。他们满足于可以在计算机上表达自己的能力,不管是通过屏幕还是扬声器。但是用鼠标和键盘输入就感觉受限制。输入通常比输出容易,因为感觉活动比移动事物需要较少能量。

输出

最挑剔的计算工程是那些不仅仅感觉世界,而且还要改变世界的工程。总体来说,物理输出要比输入难,因为输出经常需要电流(与电子学不同)和机械技巧。本书的第 1 部分将讲解一些很容易使用的设备,如光、声音和移动方面的设备。与台式机连接也很容易,这些计算机有很大的存储声音和图像的容量。第 2 部分将深入讲解输出,通过使用电机和其他设备来移动物质世界中的物体。

处理

输入和输出是物理计算的物理部分。第 3 部分需要计算机读取输入,基于改动做出决定,然后激活输出或给其他计算机发送信息。这是需要编程的地方。

变换

物理计算的一个主要原则是变换,或者把一种形式的能量转换为另一种形式。麦克风是典型的变换器,因为它把空气中的声压转换为电压。扬声器以相反的方向转换相同的能量。变换器是任何物理计算系统的眼睛、耳朵、手、腿和嘴。

物理计算的很多挑战是把各种形式的能量,如光、热或压力转换为计算机能够理解的电能。有时可以容易找到这样的变换器;另外一些时候则需要设计交互来适应你所知道的变换器。

输入变换器(传感器)(如开关和各种电阻器)把热能、光能、动能和声音转换为电能。输出变换器(致动器,如电机和蜂音器)把电能转换为身体可以感觉到的各种能量。

数字和模拟

当描述一项活动时,从分析它可能的各种结果开始。有时我们看待事件是根据某个范围内的状态,有时只关心两个状态。当两个状态就够用时,我们称其为数字化状态。当考虑一系列状态时,我们称其为模拟状态。例如,当早上穿衣时你想知道外面的温度(模拟),而不是只想知道是热或冷(数字)¹。另一方面,当决定是否带伞时,只想知道是否下雨,而不关心雨是大是小。总体来说,数字输入和输出(I/O)比模拟输入和输出简单。因为计算机使用两个状态,或者二进制系统,但是模拟 I/O 更有趣。

你用来描述工程的语言将帮你确定 I/O 要求是模拟的还是数字的。例如,如果你可以用“是或否”或“两者”描述输入和输出,你可能是在说数字输入和输出。如果可以用“多少”描述输入,或用比较级形容词(如更强、更快、更亮等),你可能谈论的是模拟输入和输出。例如,数字输出可以决定灯光是开还是关,而模拟输出可以决定灯光是更亮还是更暗。

并行和串行

术语“数字”和“模拟”使我们清楚要听取(输入)和要说出(输出)的内容。我们还要知道说或听的方式。有时我们只表示思想,采用不连贯、一个接一个的方式。例如,乐器表演的简单旋律使我们关注旋律的结构,以及它改变我们情绪的方式。有时,我们一次表示很多思想,让它们互相补充。例如,交响乐的力量来源于同时弹奏的很多乐器的和谐美,每个乐器都很重要,但是它们的联合力量是我们最想体验的。

为了描述事件发生的顺序,我们可以说它们是一个接一个地按时间发生,或同时发生的。因为这个目的,我们把按时间顺序发生的事件称为串行事件,把同时发生的事件叫做并行事件。

我们用这些术语从广泛意义谈论组织事件的时间,还可以用它们指更科技化的工作方面。你将学到电能串行(一个接一个)流过元件的方式,或并行(同时几个)流过元件的方式。还将学到元件如何串行或并行地交换信息。

1 事实上,模拟和数字并不是最准确的术语。诸如多态与二态或者连续与二元之类的术语可能更好一些。但是数字和模拟是我们正在使用的工具的制造商当中普遍使用的术语。

实 践

物理计算是通过做而不是说来得到最好的理解,所以本书将主要集中讲解实践知识。下面是一些总体指南,帮助你理解本书的技术信息。如果你对细节不太理解,请返回这些指南,以它们做指导,重新概括你的整个项目。

开始:描述发生的情况

物理计算项目的第一步是描述你希望发生什么情况。如果你不能首先清晰地描述要发生的事件,就很难编写程序并构建电路来使之发生。从个人体验来描述项目的整个环境。描述所看到、听到和感觉到的信息,以及对环境可以改变的地方。描述事件展开的完整体验,当采取各种行动时会有什么改变,这些改变如何吸引注意力和改变行为。描述为什么会吸引这个人,以及事件发生的顺序。在实现这个项目时,你需要多次修订这些描述,所以不用担心会遗忘什么。另一方面,不要让实现的过程干扰你,而没有记下遗忘的细节。

集中描述要发生什么情况,而不是如何发生。避免描述具体的技术或使用的工具。这些细节会阻碍你的思维,也许会削弱你的观念。经常会有学生跳过技术,直接问如何使用某个难解和不易使用的传感器。我们的第一个问题总是,“你用这个干什么?”经常是他们描述完要发生的事,不用描述技术,就可以找到更简单的解决方案。

例如,如果你想在聚会上隆重地宣告客人的到来。当一个客人进入房间时,打开绚丽的帘布,用明亮的聚光灯照住这个人,并响起热烈的掌声。这些描述与技术没有丝毫关系,但是描述得很清楚,可以使你开始计划如何实现它。你知道需要帘布、聚光灯和掌声,还知道要能察觉到人们走进房间。

描述完项目并用清晰的语言反复陈述了你的观念后,不要考虑技术,而应该把项目分解为几个阶段:输入、输出和处理。例如,上述示例中的输入是某人进入房间,输出是灯光和掌声,而处理在人们进入房间时打开聚光灯和鼓掌。

接下来,标识你的输入和输出是数字的还是模拟的,并开始研究最好的变换器。同样使用上述示例,如果你想根据某人进入房间的距离远近来决定掌声的音量,就需要模拟输入和输出。如果想根据某个人是否在房间来决定是否需要掌声,就需要数字输入和输出。如果你的项目分解为如图 0-2 所示的种类,就可以使你关注本书最相关的部分。使用这个工作表或类似的工作表来填写你的项目的输入和输出需要。

数字输入	模拟输入	处理	数字输出	模拟输出

图 0-2 对你的物理计算挑战分类

另外,你应该描述事件发生的顺序。灯是在掌声前亮起吗?或者是同时发生?前者是串行事件,后者是并行事件。

参看图 0-3 的流程,可以明白你的项目的复杂程度和需要做的事情。

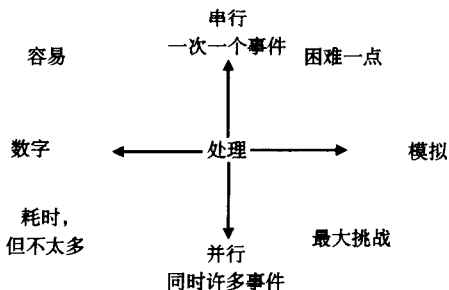


图 0-3 映射你的项目：模拟和数字，串行和并行

抽象(和分离)层次

对于任何技术实践,都不可避免决定你与工具间的抽象层次。更高级的工具使你处于从技术细节抽象的更高层次。²因此,它们易于使用,但并不总是让你随心所欲。开始于最高层次的方法仍可以完成工作。通过更高级的工具,可以快速尝试新观念,如果不起作用,可以在投入更多技术和精力前抽身而出。技术改变得很快,也许将来的更高级的工具相当于现在的低级工具。

在实践中却不是这么清晰。低级工具会使你走入岔道。例如,如果你是个美食爱好者,也许你从一开始就会被烹调吸引而不关注味道是否更好,因为你喜欢的是过程。小心被技术噱头分心,也许这些技术分散注意力而且费时,而结果却不会更好。另一方面,如果你知道一些烹调技术,但使用粗制的食物也不会制作出美味的菜肴。如果你想尝试很特殊、不同寻常的事情,自己动手有时比寻找、拼凑要容易。在最高层次合并工作,知道低层可用的东西,并知道何时向上或向下,就会产生最好的结果(参看表 0-1)。

表 0-1 抽象的层次

软件	食物	微控制器
较高层次(“Hello World!”)	较高层次(你好,请问需要些什么?)	较高层次(“Hello World”)
MAX	外卖	Teleo
LINGO/ACTIONSCRIPT	电视定餐	BASIC STAMP 2
PROCESSING	汉堡包自助	BX-24
JAVA	使用超市的柜台	BasicAtom Pro24
C	使用超市的加工工艺和屠夫	PIC
ASSEMBLY	自己种植食物,收获并从头开始准备	SX
较低层次(“1001001 0110110”)	较低层次(“亨利,去宰只鸡,今天晚上我们要吃煎饼。”)	较低层次(“1001001 0110110”)

2 如果你习惯于把“更高层次”看做是意味着在技术上更先进,那么这种考虑高层次和低层次的方式似乎是违反直觉的。相反,“更低层次”的想法意味着你和计算机硬件之间更低层次的联系。我们认为较少的联系会大有帮助。

工 具

我们将会给出不同层次的示例,但趋向于中间到高层次的工具。为了连接物质世界和数字,你将学习构建电路,把它们与计算机连接,为计算机编写软件并使计算机可以互相通信(参看图 0-4)。

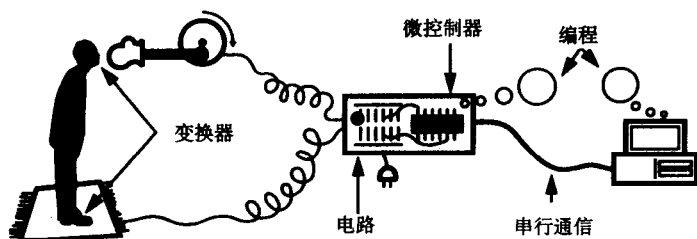


图 0-4 物理计算系统的组成部分

电路

必须构建一条电路来连接变换器和计算机。变换器用来感觉和控制世界,计算机用来解释发生的事件。对于多数常见电路,可以复制 4~5 条基本电路,我们将在本书开头几章中学到这些内容。构建这些电路很简单。只需连接一些电线和一到两个电子元件。

尽管这些电路有助于我们了解电流行为,但是我们的目的是用最少的工作把信息通过传感器从物质世界输入到计算机中。我们将讲解所需电路的基本知识,并指出获取更多信息的其他来源。从某种意义讲,计算机是所有常规电路之母,可以在软件中调试输入和输出的连接。有了对电流的基本认识,我们可以更好地理解物理计算。

通常用电路图描述电路,电路图显示了电子元件及其相互之间的连接方式。需要对电路图有足够的了解以读懂它们,但开始阶段不用画电路图或设计电路。

对变换器有了更多认识后,就涉及到能量转换。然后会学到更多有关电流行为和构建电路的知识,特别是处理像电机这样大功率的输出设备。

计算机

把“计算”与“物质”这两个词语放在一起似乎有些奇怪。计算机技术的主要优势在于它超出了物质世界的时间和空间。但物理计算仍然是关于人们所喜欢的物质世界的愉悦和限制的科学。在物理计算中,我们两者都想要:我们想使计算机给予的自由在人类喜欢的感官世界中也可以实现。因此,我们使用各种计算机,但总是竭力将它们放在后台中,以使我们着重关注前台中人与人之间的体验。

微控制器(见图 0-5)

在物理计算中使用的计算机主要是微控制器。这是个很小、很简单的计算机,擅长做三件事:从传感器接收信息,控制基本电机和其他引起物理变化的设备,以及给计算机和其他设备发送信息。它们充当物质世界和计算世界之间的大门。微控制器是复杂电子设备的中心,所以理解它们的工作方式将使你对拥有的电子设备有新的认识(图 0-5 显示了 4 种微控制器)。

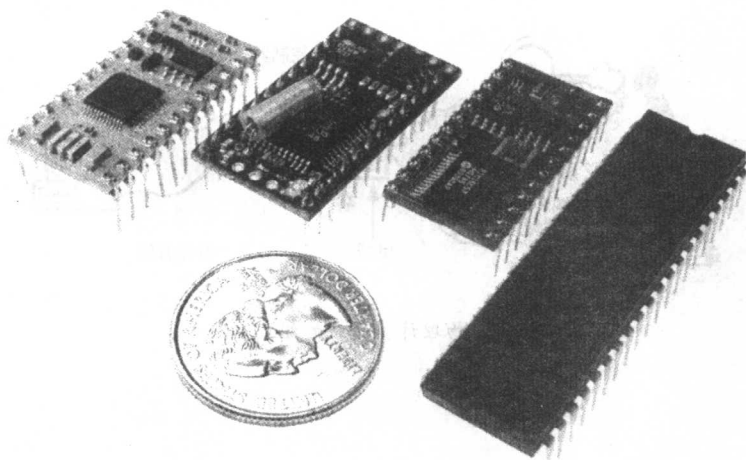


图 0-5 本书中讨论的 4 种微控制器。这些简单的计算机是许多物理计算应用的核心

微控制器很小,也便宜。这就使你可以探索特定位置的项目,把计算机安置于最不可能的地方,如灌木丛或脚底,或者开发把许多简单设备的动作集成更有趣的整体的项目。

微控制器可以用于很多事物上,如从洗衣机到电灯开关上。应该感谢这个无所不在的小东西,它降低了成本并改善了易用性。

多媒体计算机

在某种程度上,多媒体计算机(台式机或笔记本计算机)是我们在物理计算中使用的计算机。这些计算机假定使用它们的人除了手指和手之外,其他部位相对不活动,并且他们的眼睛和耳朵关注于一个方向。这些计算机在输出方面也许是多媒体的,但在输入方面不行。我们的一个主要目标是为人们勾画计算机,使人们认识到它不仅是桌上的一些大塑料盒;还要描述人和计算机间的交互,而不仅是打字和击键。我们想把计算机描述成超越多媒体计算机的原因是,它们非常有用,特别是在产生声音和图像,通过照相机和麦克风感觉物质活动方面。很多项目把微控制器有趣的输入和输出与多媒体计算机的输出能力结合起来。另一方面,如果你的项目没有涉及任何多媒体,如播放声音和视频,你也许根本不需要复杂地使用多媒体计算机及其带来的花销。与多媒体计算机相联系是本书与有关机器人学的书籍的区别。机器人学书籍趋向于有单独的微控制器。在多媒体计算机有助于人们之间的通信时,我们不会舍弃它的输出能力。多媒体计算机对项目的原型部分也有用,可以很小且便于携带,但是不会很快小型化。出于演示未来主义项目的目的,你也许会说“不要在乎幕后那

台巨大的计算机。”

中间计算机

在多媒体计算机和微控制器之间有广泛的计算机。例如,笔记本计算机、平板计算机、单片机、掌上型计算机和移动电话都是满足个人需要的计算机。这些中间平台使用的操作系统和开发环境多数与多媒体计算机相似,所以本书讲解的微控制器和多媒体计算机间的联系也会帮助理解这些平台。

编程

这个主题也许会使很多读者走开,因为他们已经学过编程并且失败了。事实上,物理计算是学习编程的好环境。本书中将抽象的编程概念(如位和字节)与实际的事物(如开关)联系起来。另外,微控制器的程序很小,也非常简单。在微控制器上你也许只想做很少的事情:读传感器,打开和关闭,向其他计算机发送信息。经常只有几行编码,而且很多是从其他程序借用来的,修改一下就可以符合你的目的。

有很多语言和微控制器可供选择,我们要推荐的是最友好的语言之一——BASIC。为微控制器编程只用在多媒体计算机上打出程序,然后下载到微控制器中即可。第5章“编程”针对有很少编程经验的人。如果你是个经验丰富的程序员,可以跳过示例直接学习语法部分。

而为多媒体计算机编程就是大项目了。编程的话题太广泛了,一本书不能概括。本书讲解计算机进行基本通信的方式。如果你有用 Director/Lingo、Max/MSP、Processing 或 Java 编程的经验,本书就非常适合你。因为我们将讲述用这些语言在微控制器和多媒体计算机间通信的方式。除了与微控制器通信外,出于项目的多媒体需要为多媒体计算机编程对我们来说过于特殊,而不能在这里作恰当介绍。如果你是编程新手,最好选择一个多媒体编程环境学习。我们的建议如上所述,并会提供一些示例来把信息从微控制器转到多媒体计算机中。

计算机间的通信

如果不谈论计算机网络,我们几乎就不必谈论计算机。即使不通过 Internet 发送信息,你也需要在本地不同类型的计算机间通信。例如,你的微控制器擅长监听开关,但是不擅长更高级的多媒体任务。它可以把信息发送给善于播放声音和视频的多媒体计算机。计算机通信有很多方法。我们将介绍一种称为串行通信的方法,它用最少的工作量提供最大的灵活性。我们还会讨论这种方法更具体的版本,如 MIDI 和 IP 协议。

你的概念:不要丢弃它

本书讲解从项目思想到你需要知道的实现它所使用的具体技术。从概念到实现很少是单行道。在开发的过程中,你的技能会改变这个概念。在你熟练掌握工具的使用方法之后,

同时会对该项目加深认识,而不一定都是在实施项目之前就具备足够的认识。但如果这是第一次使用这种技术,也许会迷失方向。

在通往物理计算的旅途上有两大陷阱。第一个也是比较令人高兴的陷阱是技术引诱。也许很高兴得到新技术能力,而去发掘不需要的技术细节或开始稀奇古怪的项目分支。事实上,难以区分技术诱惑何时会导致非常微妙和不需要的的项目,何时它又会导致自己变得孤芳自赏。当你在行进过程中时,让潜在的用户检查工作是好主意。如果用户没有注意到因为技术改动而引起的项目改进,那么就需要评估这个改动的必要性了。

第二个陷阱是启动过程太长,以至于你在某一部分受到挫折而放弃项目。有时回避技术问题需要完完全全重新开始项目,有时也许会对最终结果有很大损害。

有4件事可以使你关注于实现思想。首先,记录旅程日志。写下你的思想、遇到的问题 and 形成的解决方案。这有助于你记住在被技术或概念问题折磨得筋疲力尽时你的进展。事实上,最好的突破口也许是在此时找到的,记录在你迷失技术概念之前所进行的工作。一个好的处理方法是经常从实现细节中抽身而出,研究整体思想,而不要等待受到挫折而重新开始旅程。最好制作一份公共的 Web 日志,使其他人从你的进程受益。

第二,快速并站在较高层次工作。只要有可能,即可使用预制的技术解决方案检测事物。在证明整体概念前,不要浪费时间完善没完没了的细节。你实现某件事物的时间越长,投资就会越多,就越发不能客观地判断它对项目的实际价值。

第三,不要受到计划的制约。除非你是巫师,否则就要去亲自尝试一些事情,研究它的工作方式。如果第一个解决方案不起作用,再用另一个。每个变化都会提供新的想法,哪些对项目有好处,哪些没有。另外,对每个解决方案不要投资过多,这样在遇到障碍时,可以更容易找到迂回方法或不同的解决方案。

第四,与其他人合作。向你自己解释,特别是向那些不喜欢你的人进行解释,会使你保持诚实。当你已经检查过很多次,也想不出何处出了问题,换个角度看问题也许是最好的解决方案。

最后,经常休息,同时从事项目的多个部分。很多解决方案会出现在你的脑海里,所以经常休息或转换任务会有所帮助。

致 谢

本书是合作的成果,不仅有编者的努力,还有众多物理计算团体的专家的贡献。这里包含的材料是我们认为是最有用的材料的集合,来源于近十几年来我们自己、朋友、同事和学生的工作成果。

Red Burns 是物理计算及本书的教母。在 NYU 的 Tish 艺术院校的交互式远程通信编程中,她从一开始就是物理计算的先行者,满足了我们很多荒诞的请求,促使我们的主题广泛化,并鼓励了那些害怕科技的人们。Red 认为物理计算不仅仅属于技术人员,而且还是属于每个人的主题。

Geoff Smith 是教父。在从交互设计到微控制器,再到电子学和软件的一切方面,他的想法一直都帮助和激励我们。如果没有他,我们就无法写出本书。

Daniel Rozin 是我们重要的合作者和顾问。他的思想很多体现在本书中,以及我们的工作、研究和教学中。

有时需要个人来支持在 ITP 看到的一些非科学的方法。我们过去和现在的资助者有: NYNEX 的 Sergio Canetti; Apple Computer 的 Joy Mountford、Mike Mills 和 Linda Stone; Interval Research 的 Joy Mountford(再次表示感谢)和 Bob Adams; 美国 Networks and Oxygen 的 Sharleen Smith; Intel 的 Dana Plautz; 以及 Microsoft Research 的 Lili Cheng 和 Linda Stone(再次表示感谢)。

ITP 物理计算的全体教师和员工(过去的和现在的)在本书的编写过程中起了重要作用。Gary Schober 与 Rolf Levenbach 在电子学方面多年来提供了很多建议,并且为 ITP 的学生建立了物理计算世界和专业电子学工程之间的桥梁。Jeff Feddersen、Todd Holoubek、Greg Shakar 和 Michael Luck Schneider 作为教师和研究人員,在本书的编写过程中一直给予支持,并提供了很多示例。Jody Culkin、Cynthia Lawson、Jen Lewin、Andrew Milmoie、Camille Norment、Will Pickering、Joe Rosen、Ben Rubin、Joey Stein、Camille Utterback 和 Stere Weiss 集体把本教材教授给很多学生。ITP 社区的很多其他人都贡献了他们特有的专家技能,包括 Luke Dubois、Dan Palkowski、Amit Pitaru、Eric Singer、Leo Villarreal 和 Jaime Villarreal。James Tu 作为我们的技术编辑,在上述很多方面都做出贡献。

ITP 的全体教师和员工一起使这些成为可能。教师成员如 Pat O'Hara、Marianne Petit 和 John Thompson 支持了物理计算课程,提供了宝贵建议并帮助我们实现。员工成员(过去的和现在的)如 George Agudow、Edward Gordon、Midori Yasuda、Robert Ryan、Nancy Lewis、Marlon Erans、Ben Gonzalez、Gilles Bersier 和 Michael Wright 多年来一直容忍我们和学生的异想天开的想法,并使它们成为现实。

我们还汲取了姊妹院校的工作,包括 MIT 传媒实验室、皇家艺术院校、KTH 和瑞典交互式协会、IVREA 交互式设计协会和 UCLA 设计和传媒艺术院校。特别值得提到的是, Ben Fry 和 Casey Reas 帮助我们他们的处理编程环境示例加入本书。

最后是 ITP 的物理计算班级的学生推进了本书的编程工作。他们中的很多人为本书

提供了好建议。这些年来我们一直从上百名学生身上学习,在此表示感谢。现在和近期的学生 Jamie Allen、Mark Argo、Jason Babcock、John Bergren、Jonah Brucker-Cohen、Eric Forman、Sasha Harris-Cronin、Daniel Hirschmann、Rania Ho 以及其他很多人都直接提供了建议、补充和校正,它们都出现在本书的章节中。

同时还要感谢 Thomson 的编辑:Stacy Hiquet、Dan Foster、Danielle Foster、Kim Benbow、Michael Tanamachi,以及我们的代理:B 工作室的 Laura Lewin。我们要特别感谢技术编辑 James Tu 的深入见解和挑剔的反馈,他必须跨很多平台检查各种技术的示例。

当然,如果没有我们家庭和合作伙伴的支持、耐心和鼓励,也不会编写出本书。感谢 Kate、Lily、Terri 和我们的父母、兄弟和姐妹。

对于我们疏忽的人员,在此表示歉意,也感谢他们无声的支持。

目 录

第 1 部分 基 础

第 1 章 电流..... 2	2.15 USB-串行适配器 19
1.1 变换:电流基础知识..... 2	2.16 工具 20
1.2 电流和电子 3	2.17 购物清单 22
1.3 电流的流动方式 4	2.18 把一切都拿回家 26
第 2 章 购买零件..... 6	第 3 章 构建电路 27
2.1 无焊实验电路板 6	3.1 示意图..... 27
2.2 微控制器 6	3.2 连接符号..... 27
2.2.1 微控制器的特性(按优先顺序) 7	3.3 电源符号..... 28
2.2.2 高级微控制器模块..... 8	3.4 找到示意图..... 28
2.2.3 中级微控制器模块..... 9	3.5 实验电路板..... 29
2.2.4 低级解决方案..... 9	3.6 微控制器的安装位置..... 30
2.3 常用元件..... 10	3.7 把示意图转换成电路..... 30
2.3.1 开关 10	3.8 使用万用表..... 32
2.3.2 电阻器 11	3.9 焊接..... 34
2.3.3 可变电阻器 12	3.10 给实验电路板加电 35
2.3.4 电容器 13	3.10.1 用快速而拙劣的方法连接 35
2.3.5 二极管 13	3.10.2 用专业的方法连接 36
2.3.6 晶体管和继电器 15	3.10.3 电压调节器 37
2.4 电线..... 15	3.11 保持整洁 39
2.5 电源..... 16	第 4 章 微控制器 41
2.6 电源连接器..... 16	4.1 “Hello World!”是难点 41
2.7 电压调节器..... 17	4.2 微控制器的安装位置..... 41
2.8 RC 伺服电机 17	4.2.1 输入 41
2.9 串行连接器..... 17	4.2.2 输出 42
2.10 串行电缆 18	4.3 确定从输入到输出的路线..... 42
2.11 时钟晶体 18	4.4 标识微控制器的引脚..... 42
2.12 端板 18	4.5 低级微控制器:外部时钟 44
2.13 项目箱 18	
2.14 扎线带 19	

4.6 第一条基于微控制器的 电路.....	45	6.2 数字输出.....	79
4.7 把程序输入芯片.....	46	6.2.1 变换器	79
4.8 为类 Stamp 的模块编程	47	6.2.2 电路	82
4.8.1 类 Stamp 的编程硬件 连接	47	6.2.3 编程	84
4.8.2 类 Stamp 的编程软件 环境	48	6.3 模拟输入.....	85
4.9 为低级芯片编程.....	50	6.3.1 变换器	85
4.9.1 硬件编程器	50	6.3.2 电路	86
4.9.2 低级编程软件环境 ..	50	6.3.3 编程	89
4.10 调试	53	6.3.4 输入的脉冲宽度 调制	93
第 5 章 编程	54	6.4 模拟输出.....	93
5.1 好消息.....	54	6.4.1 输出的脉冲宽度 调制	93
5.2 流程控制:计算机读取程序的 方式.....	54	6.4.2 调暗 LED	95
5.3 循环.....	55	6.4.3 DC 电机速度控制 ...	96
5.4 If 语句.....	56	6.4.4 产生音调	97
5.5 变量.....	56	6.4.5 RC 伺服电机	102
5.6 内置例程:子例程和 函数.....	59	6.5 从模拟输入到模拟输出:换算 函数	107
5.7 自制的例程.....	62	6.6 结束语	114
5.8 高级循环:While-Wend 和 For-Next	63	第 7 章 在计算机之间通信.....	115
5.8.1 While-Wend 或 Do- While	64	7.1 物理协议	115
5.8.2 For-Next	65	7.2 定时协议	117
5.9 伪代码.....	66	7.3 电气协议	117
5.10 指令	68	7.4 数据包尺寸	118
5.11 调试	68	7.5 数字或字母:使用 ASCII 码	118
5.12 良好的调试习惯	70	7.6 适合微控制器的软件	120
5.13 坏消息	71	7.7 微控制器的串行输出	120
第 6 章 示意图、程序和变换器.....	72	7.8 用 LED 测试.....	126
6.1 数字输入.....	72	7.9 用终端软件测试	126
6.1.1 变换器:开关.....	72	7.10 微控制器的串行输入.....	127
6.1.2 数字输入电路	74	7.11 串行冻结和阻塞功能.....	129
6.1.3 编程	75	7.12 你的专用协议.....	131
		7.12.1 标点符号.....	131
		7.12.2 呼叫和应答.....	132
		7.13 发送更大的数.....	135
		7.13.1 用文本发送数.....	135