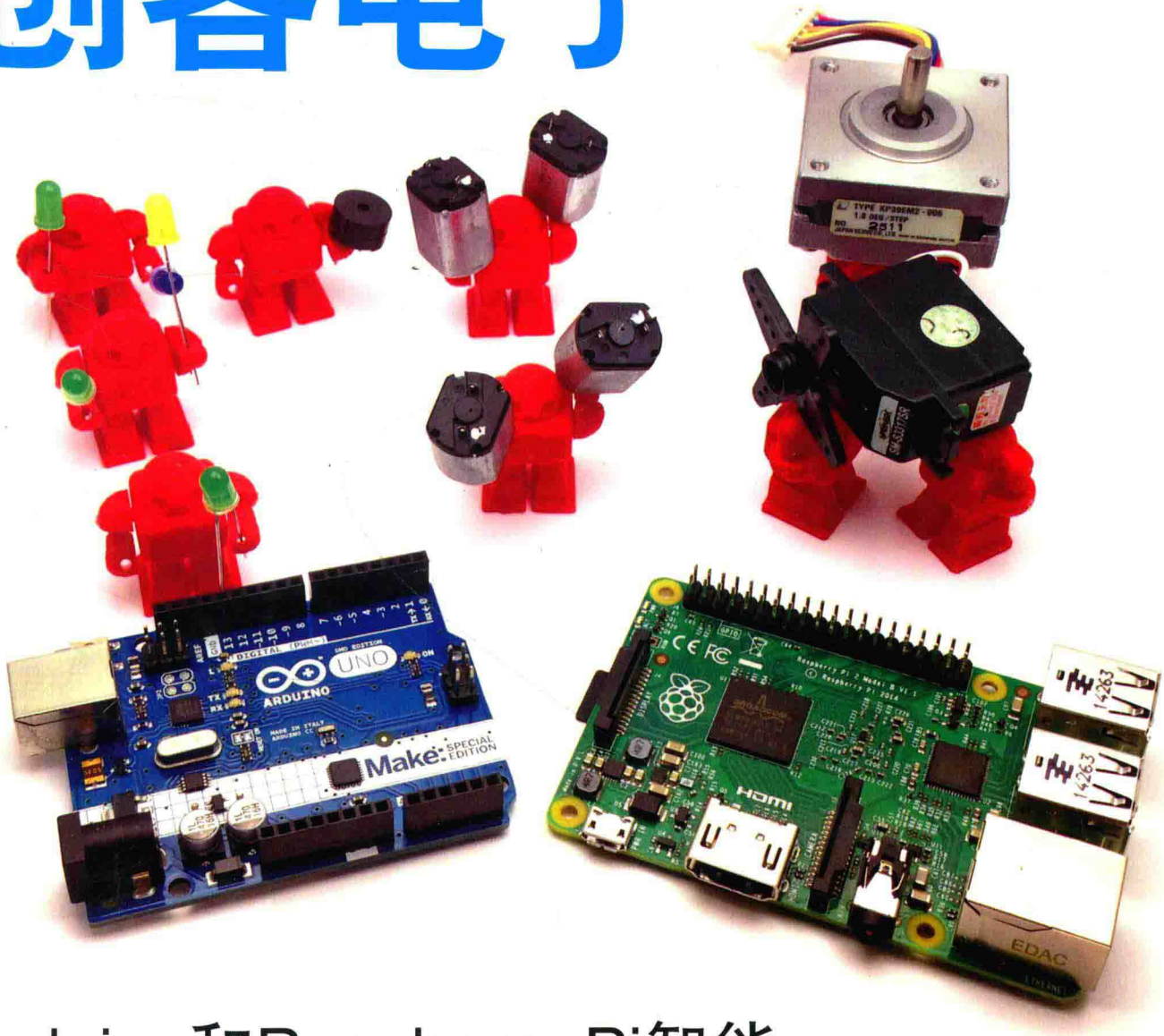
Make: Action

# 创客电子




## Arduino和Raspberry Pi智能制作项目精选

[英] Simon Monk 著

陈立畅 张佳进 马 瑛
陈 颖 毕成杰 任雨橦 译

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

MAKER MEDIA
SEBASTOPOL, CA

I'M MAKER 创客

# Make: Action

# 创客电子

## Arduino和Raspberry Pi智能制作项目精选


[英] Simon Monk 著
陈立畅 张佳进 马 瑛
陈 颖 毕成杰 任雨橦 译

人 民 邮 电 出 版 社
北 京

**图书在版编目（CIP）数据**

创客电子 : Arduino和Raspberry Pi智能制作项目精选 / (英) 西蒙.蒙克 (Simon Monk) 著 ; 陈立畅等译. -- 北京 : 人民邮电出版社, 2017.10
(i创客)
ISBN 978-7-115-46660-0

Ⅰ. ①创… Ⅱ. ①西… ②陈… Ⅲ. ①单片微型计算机－应用－电子器件－制作②软件工具－应用－电子器件－制作 Ⅳ. ①TP368.1②TP311.56③TN

中国版本图书馆CIP数据核字(2017)第199312号



## 内 容 提 要

本书主要介绍使用Arduino和Raspberry Pi控制身边的小物件，让其能够感知声、光，并控制它的动作。全书在讲述了基本知识的基础上，介绍了多个有趣的制作项目，图文步骤，让读者可以一步步跟着制作出来，通过实践进行学习。制作项目包括，用Arduino制作一个自动浇水器，设计一个LED交通信号灯，用Raspberry Pi制作会跳舞的小玩偶，等等。

◆ 著 [英] Simon Monk
译 陈立畅 张佳进 马 瑛 陈 颖 毕成杰 任雨橦
责任编辑 魏勇俊
责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京天宇星印刷厂印刷

◆ 开本：880×1230 1/16
印张：10.5 2017 年 10 月第 1 版
字数：395 千字 2017 年 10 月北京第 1 次印刷

著作权合同登记号 图字：01-2016-10030 号

定价：79.00 元

**读者服务热线：(010)81055339 印装质量热线：(010)81055316**
**反盗版热线：(010)81055315**
**广告经营许可证：京东工商广登字 20170147 号**

# 目录

# 简介 1

**Arduino 和 Raspberry Pi 使业余电子爱好者进入电子世界变得简单。也许你想设置一个 DIY 的家庭自动化系统，以便于你通过 WiFi 网络控制家庭的灯光和加热器，或者简单地控制一些电机的转动。**

**这本书将告诉你如何使用流行的 Raspberry Pi 和 Arduino 平台，这样你的 Pi 和 Arduino 设备就可以操控运动、光和声音。**

## Arduino 和 Pi

虽然 Arduino 和 Raspberry Pi 的体积都很小，其开发板都如同信用卡一般大小，但是实际上它们是差异性很大的两种设备。Arduino 是一种非常简单的微控制器板，且不能够运行任何操作系统。然而，Raspberry Pi 却是一种微型计算机，它可以运行 Linux 系统并且能够连接外部电子设备。

### Raspberry Pi

对于刚开始接触电子元器件，但是却又能熟练使用计算机的这类人而言，Raspberry Pi 将会是一种比较熟悉的设备。Raspberry Pi（图 1-1）和众多能够运行 Linux 系统的普通计算机相比，是非常小的一个版本。其拥有 USB 接口，方便人们连接到键盘和鼠标，同样可以连接到显示器或电视进行 HDMI 视频输出和音频输出。

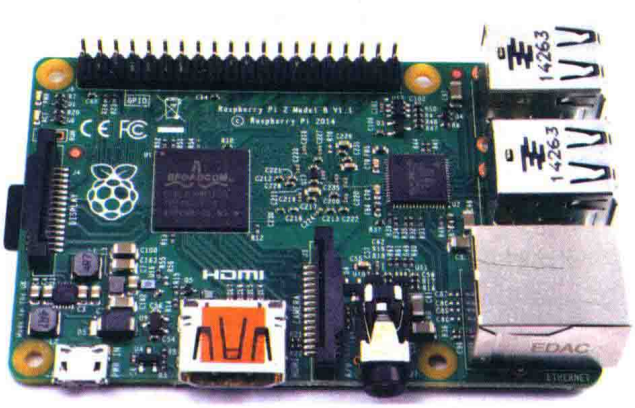



图 1-1 Raspberry Pi 2

Raspberry Pi 有一个以太网接口可以连接到你的网络，也可以连接 USB 无线适配器。其电源可以由 microUSB 插座供给。

其以一张 microSD 卡作为存储，而不是利用传统的磁盘驱动。这张卡里包含操作系统以及所有文档和程序。

Raspberry Pi 创立于英国，主要作为一种低成本计算机，用来帮助学校的孩子们学习计算机基础知识，尤其是 Python 编程。实际上，Pi 这个词据说是由 Python 中 Py 派生而来的。

以下几个方面使得 Raspberry Pi 与运行 Linux 的普通台式机或笔记本电脑不同。

- Raspberry Pi 的成本只有 40 美元（一款简装版的 Raspberry Pi 被称为 ModelA+，它只需要较低的价格。还有一种 Model zero 甚至更便宜）。
- Raspberry Pi 使用低于 5W 的功率。
- Raspberry Pi 有一排通用输入 / 输出（GPIO）引脚，允许直接连接电子元件（引脚在图 1-1 的左上部分。通过这些引脚，你可以控制 LED 灯、显示器、电机，以及之后在这本书中你会使用到的所有不同类型的输出设备）。

此外，Raspberry Pi 还可以使用 WiFi 或 LAN 电缆连接到 Internet，使其适用于物联网项目（第 16 章）。

Raspberry Pi 2（目前为止最新和最好的版本）的规格如下。

- ARM v7 900 MHz 四核处理器
- 1GB DDR2 内存
- 10.100 BaseT 以太网
- 4 个 USB 2.0 端口
- HDMI 视频输出
- 相机接口插座
- 40 针 GPIO 接头（所有引脚的工作电压为 3.3V）
- 如果是 Raspberry Pi 的新手，那么可以使用第 3 章中的 Python 编程语言来学习硬件和软件。

## Arduino

有很多不同型号的 Arduino 可以采用。本书主要讨论最广泛使用和流行的 Arduino 版本，Arduino Uno（图 1-2）。Arduino 比 Raspberry Pi 便宜一点——购买一个 Arduino Uno 大约需要 25 美元。

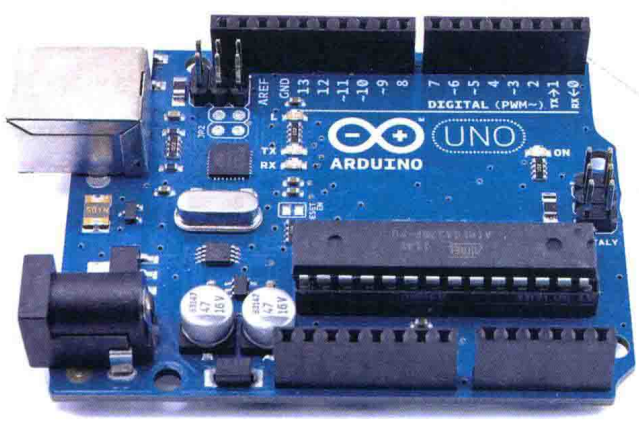



图 1-2 Arduino Uno 修订版 3

如果你习惯使用普通计算机，那么 Arduino 的性能就显得捉襟见肘，其内存只有 34KB（各种类型）。这意味着 Raspberry Pi 的内存是 Arduino 的三万倍，而且还不包括 Raspberry Pi 的 SD 闪存卡。此外，Arduino Uno 的处理器时钟频率只有 16MHz。Arduino 不能连接键盘、鼠标或显示器，也不能运行操作系统。

你可能很想知道 Arduino 实际上可以用作什么。它的使用非常简单，而且没有需要启动的操作系统甚至是在项目中你根本不需要用到的接口，那些接口只会增加成本和电力消耗。

可以说 Raspberry Pi 是一台通用计算机，而 Arduino 只需要做好一件事——连接和控制电子元件。

如果要对 Arduino 进行编程，那么你需要一台普通计算机（如果你愿意的话，你甚至可以使用 Raspberry Pi）。在你选择的计算机上，你需要运行集成开发环境（IDE），这允许你编写要下载到 Arduino 内置闪存的程序。

Arduino 一次只能运行一个程序，一旦它被编程，它会记住该程序，并在其上电后自动运行该程序。

Arduino 被设计成可以接受拓展板，其拓展板插在 Arduino 的输入 / 输出插座的顶部并且可以添加额外的硬件功能，如各种类型的显示器，以及以太网和 Wi Fi 适配器。

你可以使用 C 语言对 Arduino 进行编程（你可以在第 2 章中找到有关编程和使用 Arduino 的更多信息）。

## 选择设备：Arduino 或 Pi

本书介绍如何将电子产品连接到 Arduino 和 Raspberry Pi 的原因之一是，有些项目更适合于 Raspberry Pi，有些则适用于 Arduino。位于这两者之间的其他开发板通常十分类似于 Arduino 或 Raspberry Pi，这些开发板的使用方法也可以参考本书。

当开始一个新项目时，通常的经验是默认使用 Arduino。但是，如果项目有以下要求之一，那么 Raspberry Pi 可能是更好的选择。

- 互联网或网络连接
- 需要一个大屏幕
- 需要附加键盘和鼠标
- 需要 USB 外围设备，如网络摄像头

花费一些功夫和费用，可以对 Arduino 进行功能扩展，以满足前述的大部分要求。然而，如果你选择这样做，一切都变得异常困难，因为这些都不是 Arduino 的原本的特性，这些都是 Pi 的特性。

选择在 Raspberry Pi 上使用 Arduino 的理由包括：

成本

Arduino Uno 比 Raspberry Pi 2 便宜。

启动时间

Arduino 不需要等待操作系统启动。它只存在着一个小的延迟，大约一秒钟，检查是否有新的程序正在下载，然后它开始运行。

可靠性

Arduino 本质上是一个比 Raspberry Pi 更简单和更可靠的设备，并且不需要建立在操作系统之上。

电能消耗

Arduino 使用的电量大约只有 Raspberry Pi 的 1/10。如果你需要使用一个电池或太阳能作为电源，那么 Arduino 会是一个更好的选择。

GPIO 输出电流

Raspberry Pi 的 GPIO 引脚只能用于提供最大约 16 mA 的电流。另一方面，Arduino 引脚的额定值为 40 mA。所以，在某些情况下，你可以直接连接一个东西（如一个明亮的 LED）到 Arduino，但是不能将其直接连接到 Raspberry Pi 上。

Arduino 和 Raspberry Pi 都是用于项目开发的非常好的设备。并且，选择使用哪个设备在某种程度上也只是个人喜好的问题。

在将外部电子器件连接到 Raspberry Pi 时，必须要牢记它工作电压是 3.3V，而不是 Arduino 的 5V。如果将 5V 连接到 Raspberry Pi 的 GPIO 引脚，可能会损坏 GPIO 引脚甚至损坏整个 Raspberry Pi。

## 备选方案

Arduino Uno 和 Raspberry Pi 占据着一系列可用于控制事物的设备的两个极端。正如你所期望的，市场已经产生了许多其他设备，它们处于这两个极端之间，有时试图集两者的优点于一身。

新设备一直在层出不穷。Arduino 的开源性质导致了它有许多变化，因为设计是因人而异的，如控制无人机或与无线传感器连接。

图 1-3 显示了这一领域中一些最流行设备的传播。

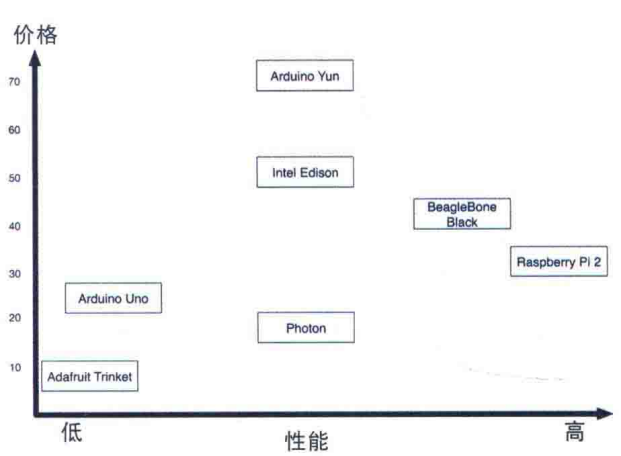



图 1-3 嵌入式平台

在 Arduino Uno 下面，Adafruit Trinket 是价格和性能的代表。这个有趣的开发板只有几个 GPIO 引脚，但在其他方面却与 Arduino 相当。Adafruit Trinket 非常适用于只有一个或两个输入或输出的项目。

有一类中间产品，包括 Arduino Yun，Intel Edison 和 Photon，它们都具有内置 WiFi 功能，旨在用于物联网（IoT）项目（见第 16 章）。其中，Photon 可能最具代表性。所有这三个设备都使用 Arduino C 编程，因此你了解到的使用 Arduino 的信息也适用于这些开发板。

BeagleBone Black 在概念上与 Raspberry Pi 非常相似。它也是一个单板计算机，从基本的计算性能来看，虽然目前的 BeagleBone Black 版本已经落后于 Raspberry Pi 2，但 BeagleBone Black 相比 Raspberry Pi 确实有一些优势。它有更多的 GPIO 引脚，还有一些引脚可以充当模拟输入，这是 Raspberry Pi 2 缺少的一个特性。BeagleBone Black 可以使用 Python 编程，类似于 Raspberry Pi 或 JavaScript 编程。

## 小结

本章简要介绍了 Arduino 和 Raspberry Pi。我们讨论了这些开发板的优点和缺点，并且还考虑了一些备选方案。接下来的两章将让你开始使用和编程 Arduino 和 Raspberry Pi。如果你以前使用过 Arduino 和 Raspberry Pi，你可以直接跳到第 4 章，使用 Arduino 和 Raspberry Pi 做一些操作！ 如果需要，你可以随时返回第 2 章和第 3 章。

# Arduino 2

**本章是 Arduino 入门书的修改版本，最初作为我的书“*The Maker's Guide to the Zombie Apocalypse from NoStarch Press*”的附录出版，并在此获得许可。**

**如果你是第一次接触 Arduino，本章将让你开始了解这个伟大的小开发板。**

## 什么是 Arduino？

Arduino 板有各种类型，但迄今为止最常见的和本书项目中常用的是 Arduino Uno（见图 2-1）。

Arduino Uno 经历了一些修订。图 2-1 中所示的 Arduino Uno 是一个 3 版本（R3）的开发板（写作时最新的）。

让我们用 USB 接口开始我们的 Arduino 之旅，它有几个用途：它可以用来为 Arduino 提供电源，从你的电脑对 Arduino 进行编程，还有就是作为通信链路。

USB 插座旁边的红色小按钮是复位按钮。当你按下此键时，Arduino 将重新启动并运行安装在其上的程序。


图 2-1 Arduino Uno

沿着 Arduino 的顶部和底部边缘，有可以连接电子器件的连接插座。在图 2-1 的顶部可以看到，数字输入与输出端口，上面有 0 到 13 之间的数字标识。你可以将开关连接到数字输入端，其将能够判断开关是否被按下。或者，你可以将 LED 灯连接至数字输出端，并通过改变数字输出端由低电平至高电平来点亮 LED 灯。事实上，在板上内置有一个 LED 灯，被称为“L”，其连接至引脚 13。

在数字 I / O 引脚下方，有一个电源 LED 指示灯，用来指示开发板已经通电。ICSP（在线串行编程）接头仅用于 Arduino 的高级编程并且不需要使用 USB 连接。绝大多数的 Arduino 用户永远不会使用 ICSP 接头。

这里值得强调的是 ATMega328（Arduino 的大脑）。ATMega328 是一个微控制器 IC（集成电路）。这个芯片存储 Arduino 将要运行的程序（它包含 32 KB 的闪存）。

在 ATMega328 下面，有一排标记为 A0 到 A5 的模拟输入引脚。数字输入只能指示一些开关量，而模拟输入则可以测量引脚上的电压（只要电压在 0 到 5V 之间）。这样的电压可以来自某种类型的传感器。如果缺少数字输入和输出，那么这些模拟输入引脚也可用作数字输入和输出。

## 安装 Arduino IDE

在这之后，有一排电源连接器，可以用作为 Arduino 供电的替代方法。它们还可为你所要连接的其他电子设备供电。

Arduino 还有一个直流电源插孔。这可以连接直流电压在 7V 到 12V 之间的任何设备，并可以使用电压调节器为 Arduino 提供 5V 的工作电压。Arduino 可以自动从 USB 接口获取电源或者根据连接的直流连接器获取电源。

Arduino 并不是如你所预期的计算机一样。它没有操作系统，不需要连接键盘，显示器或者鼠标。它

只运行一个程序，并且你需要通过计算机将程序下载到 Arduino 的闪存内。同时只要你愿意，还可以对 Arduino 重新编程多次（甚至数千次）。

为了对 Arduino 进行编程，你需要在计算机上安装 Arduino IDE。这个软件具有良好的跨平台特性——Arduino IDE 可以在 Windows，Mac 和 Linux 上运行——这是 Arduino IDE 非常受欢迎的原因之一。此外，Arduino IDE 允许你通过 USB 连接对 Arduino 进行编程，并且不需要特殊的编程硬件。

要为你的平台安装 Arduino IDE，请下载软件，然后按照 Arduino 网站（*http://arduino.cc/en/Guide/HomePage*）上的说明进行操作。

注意，Windows 和 Mac 用户将需要安装 USB 驱动程序，以便 Arduino IDE 可以与 Arduino 本身进行通信。

安装完毕并运行 Arduino IDE，就会弹出 Arduino IDE 的窗口，如图 2-2 所示。

按下下载按钮，便可下载程序至 Arduino 板。在此之前，Arduino IDE 会将编程代码编译为 Arduino 可以执行的代码。如果存在错误，则会将错误显示在日志区域之中。验证按钮执行的是相同的操作，但没有执行将程序下载至开发板的最后一步。

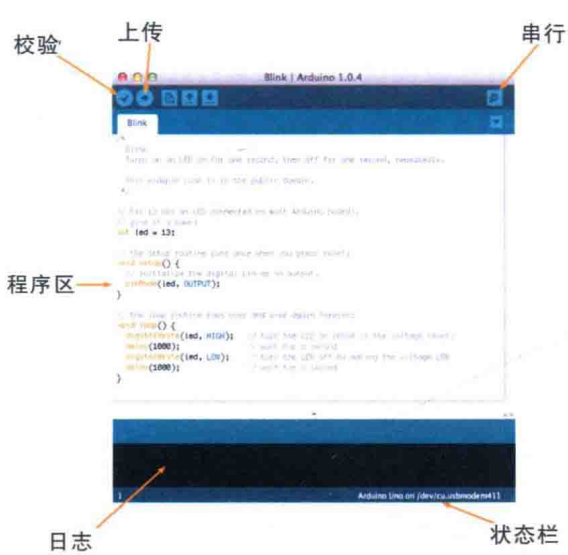



图 2-2　Arduino IDE

串行监视器按钮可以打开串行监视器窗口，用于与 Arduino 通信。在本书的许多实验中，你将使用串行监视器，因为它是从计算机向 Arduino 发送命令的一种很好的方法。串行监视器允许双向通信，这意味着你可以发送文本消息到 Arduino，并从它接收响应。

窗口底部的状态区域将告诉你 Arduino 的类型以及其相应的串行端口，当你按下“Upload”按钮时，将通过该端口对 Arduino 进行编程。图 2-2（*/dev/cu.usbmodem411*）中显示的端口是你在使用 Mac 或 Linux 计算机时所期望看到的类型。如果你使用 Windows 计算机来对 Arduino 进行编程，那么这将是 COM4 或 COM 后跟的一个数字，是 Windows 分配给 Arduino 的连接端口。

最后，但同样重要的是，Arduino IDE 的主要部分是程序区，你可以在其中键入要下载到 Arduino 的程序代码。

在 Arduino 的世界中，程序被称为 sketches（经 Arduino IDE 编译后的程序），Arduino IDE 的文件菜单允许你像在处理文字处理器中的文档一样打开和保存程序。文件菜单中还有一个 Examples 子菜单，你可以从中加载 Arduino 的内置示例程序。

## 上传程序

要测试你的 Arduino 板，并确保 Arduino IDE 已正确安装，请通过打开名为 Blink 的示例程序开始。你可以通过以下导航进行寻找，文件→示例→01. Basics.（Blink sketch 是图 2-2 种加载的示例）。

使用 USB 线将 Arduino 连接到要用来对 Arduino 进行编程的计算机上。你应该会看到 Arduino 的电源 LED 指示灯亮起，其他几个 LED 指示灯则在插入时会闪烁。

可以使用已连接的 Arduino，你需要告诉 Arduino 正在被编程的板子的型号（Arduino Uno）和它所连接的串行端口。通过导航到工具→板→Arduino Uno 设置板的类型。

通过导航到工具→串行端口设置串行端口。如果你使用的是 Windows 电脑，则不太可能有很多选择。实际上，你可能只能找到选项 COM4。在 Mac 或 Linux 计算机上，通常有列出的各种 USB 设备，可能很难确定哪一个是你的 Arduino 板。你应该在列表中看到一个启动 *dev / tty.usbmodemNNN* 的项目，其中 *NNN* 是数字。如图 2-3 所示，选择 Arduino 连接到我的 Mac 上。

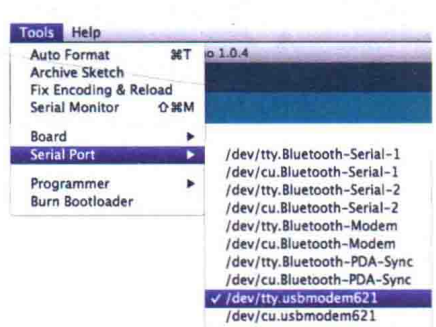



图 2-3　选择 Arduino 串口

如果你的 Arduino 没有出现在列表中，通常是因为 USB 驱动程序的问题。如果确实是 USB 驱动程序的问题，请重新安装。

如果已经准备好将程序上传至 Arduino，请按下上传按钮。在窗口的日志区域则会显示消息，等待几秒钟后，Arduino 上标有“TX”和“RX”的 LED 将会在程序上传至开发板时开始闪烁。

如果一切都按照计划进行，上传完成后，你应该会看到类似图 2-4 所示的消息。


图 2-4　成功上传

此消息告诉你程序已上传，并且已使用可用的 32,256 个字节中的 1,084 个字节。

程序完成上传后，你会注意到 Arduino 上的内置“L”LED 将慢慢闪烁。这就是你新上传的以 Blink 命名的程序。

## 本书代码

本书的所有软件——Arduino sketches（可以称为 Arduino 程序）和 Raspberry Pi 的 Python 程序——都可以从本书的 GitHub 页面（*https://github.com/simonmonk/make_action*）获得。

若要获取文件到你的 Mac，Linux 或 Windows 计算机，单击下载 ZIP 按钮，它位于 GitHub 页面的右下角。

你将下载一个 ZIP 文件，你可以将其保存在桌面或者其他方便的位置。当你打开 ZIP 压缩文件时，它将解压一个名为 *make_actionmaster/* 的目录。然后，你可以在名为 *arduino/* 的目录下找到 *Arduino* 的程序代码。在 *arduino/* 的目录下有两个子目录，*experiments/* 和 *projects/*。

每个实验或项目程序都保存在自己的目录中，通常情况下，其中有一个文件是实际的程序。例如，在目录 *experiments /* 中，你将找到目录 *ex_01_basic_motor_control /*，其中包含单个文件 *basic_motor_control.ino*。如果你已经安装了 *Arduino IDE*，点击打开此文件时将自动在 *Arduino IDE* 中打开它。

访问这些程序的另一种方法是将实验和项目文件夹复制到你的 Arduino sketchbook 目录（称为 *Arduino /*）中，同时也将复制在你的普通文档文件夹中（例如 Windows 电脑中的 *My Documents/* 或苹果电脑中的 *Documents/*）。

如果文件被复制到 *sketchbook /* 目录，那么你将能够通过导航到 Arduino IDE 中的 File → Sketchbook 来打开它们。

## 编程指南

本节包含主要编程命令的概述，以帮助你了解本书中使用的程序代码。如果你第一次接触此类编程，并且想学习 Arduino C，那么你应该考虑阅读我的关于 Arduino 编程的书：Arduino 编程入门。

### Setup 和 Loop

所有的 Arduino 程序必须包括一个 setup（）函数和一个 loop（）函数（函数是指用来完成某些动作命令的程序代码块）。到底 setup（）和 loop（）是如何工作的呢？让我们以上传至 Arduino 的 Blink 程序为例进行分析。

```
int led = 13;
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for 1 second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for 1 second
}
```

你会注意到，很多程序文本前面有一个 //。这表示 // 之后的文本将被视为注释。它不是程序代码，它只是文档，用以说明该程序具体命令动作。

正如注释所说，setup（）函数内的代码只运行一次（更确切地说，是每当给 Arduino 上电或者复位时，此函数内代码运行一次）。所以，当程序启动时，你只需要使用一次 setup（）函数就能完成所有你需要做的事情。在 Blink 的例子中，这里指明 LED 引脚为输出。

Loop（）函数中的命令将反复地运行，也就是说，一旦 loop（）中的最后一条命令执行完成后，它将再次在第一行重新开始。

我们跳过了 setup（）和 loop（）函数在 Blink 例子中的具体命令的解释，但是不需要担心，我们很快就会讨论它们。

## 变量

变量是名字赋值的一种方法。Blink 程序的第一行（忽略注释）如下：

```
int led = 13;
```

这里定义了一个名为 led 的变量，并赋予它一个初始值 13。赋值 13 是因为这是 L LED 连接到 Arduino 引脚的名称，int 是变量的类型。int 是整数的缩写，表示整型（无小数点）。

虽然你不必为每个引脚都设置变量名称，但这的确是一个非常好的方法，因为它更容易看清楚引脚用于什么。此外，如果你想使用不同的引脚，你只需要在一个定义变量的地方改变变量的值就可以了。

你也许已经注意到，在本书附带的程序中，在声明变量并定义引脚时，在该行前面有 const，如下所示：

```
const int led = 13;
```

关键字告诉 Arduino IDE 该变量不是一个会变的量，而是一个常量；换句话说，它的值永远是 13。这样不仅可以减少程序也能提高程序的运行速度，并且这通常被认为是初学编程的好习惯。

## 数字输出

Blink 程序是数字输出的一个很好的例子。引脚 13 被配置为 setup（）函数中的一个输出（该变量 led 之前被定义为 13）。

```
pinMode(led, OUTPUT);
```

这就是在 setup（）函数中的程序，它只需要被执行一次。一旦引脚被设置为输出，它将一直作为输出，直到我们改变它的定义。

在 Blink 中需要 LED 灯不断地闪烁，所以其代码为：

```
digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
delay(1000);               // wait for 1 second
digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
delay(1000);               // wait for 1 second
```

digitalWrite（）有两个参数（括号内，用逗号分隔）。第一个参数是要通过引脚写入 Arduino，第二个则是要给写入的引脚赋值。因此，HIGH 值会将输出设置为 5V（这将打开 LED 灯），而 LOW 值将会将引脚设置为 0V，从而关闭 LED 灯。

delay（）函数将程序暂停以指定参数的时间量（以毫秒为单位）。在 1 秒内有 1000 毫秒，因此这里显示的每个 delay（）函数将暂停程序 1 秒钟。

在 “实验：控制 LED” 章节中，你将使用外部 LED 的数字输出，并让其闪烁，而不是内置 LED。

## 数字输入

本书主要使用 digitalWrite（），关注于输出而不是输入。然而，你还应该了解数字输入，可用于将开关和传感器连接至 Arduino。

你可以使用 pinMode（）函数将 Arduino 引脚设置为数字输入。以下示例将把引脚 7 设置为输入。你也可以使用变量名代替。

正如输出一样，在 setup（）函数中定义一个输入引脚，因为程序运行时很少改变引脚的模式：

```
pinMode(7, INPUT)
```

将引脚设置为输入后，可以读取引脚，以确定引脚电压是否接近 5V（高电平）或更接近 0V（低电平）。在此示例中，如果输入端在测试时为低电平，则 LED 将打开（一旦点亮，LED 将保持点亮，因为代码中没有任何内容将其关闭）。

```
void loop()
{
  if (digitalRead(7) == HIGH)
  {
    digitalWrite(led, LOW)
  }
}
```

现在，代码将开始变得复杂，所以我们一次一行。

在第二行，我们有一个符号{。有时，这与loop（）放在同一行，有时在下一行。这只是个人喜好的问题；它对代码的运行没有影响。符号{标记代码块开始，而符号}标记代码块结束。它是一种将属于循环函数的所有代码行分组在一起的方法。

第一行使用if语句。紧接在单词“if”之后是条件。在这种情况下，条件是（digitalRead（7）== HIGH）。双等号（==）是一种比较两边的两个值的方法。所以，在这种情况下，如果引脚7为高，那么由{and}之后的if代码块将运行，否则它将不会运行。整齐排列{and}可以使得更容易分辨哪个符号{与符号}相匹配。

如果条件为真，我们就可以看到要运行的代码。这是我们用来打开LED的digitalWrite（）函数。

在本示例中，假设数字输入为高电平或低电平。如果使用连接到数字输入的开关，则该开关可以做的是关闭连接。通常，这意味着将数字输入连接到GND（0V）。如果开关断开，则数字输入将被认为是漂浮的。换句话说，电路上它没有牢固地连接到任何东西。输入将受到电噪并在高电平与低电平之间来回变化，为了防止这种不良编程指南现象，通常使用上拉电阻，如图2-5所示。

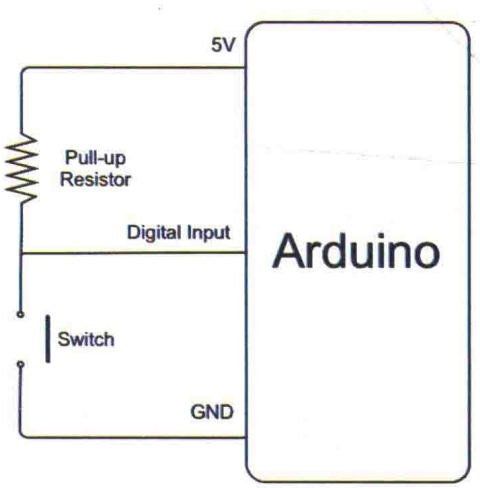



图2-5 数字输入端使用上拉电阻

Pull-upResistor：上拉电阻；Switch：开关；DigitalInput：数字输入；GND：接地

当开关未关闭时（如图2-5所示），电阻将输入引脚拉高至5V。当开关通过按下按钮关闭时，输入端连接到GND。

虽然可以使用外部的上拉电阻，但是如果将引脚模式设置为INPUT_PULLUP而不是INPUT，则Arduino输入具有大约40kΩ的内置上拉电阻。以下代码演示了如何设置数字输入的引脚模式以与开关一起使用，而无需使用图2-5中所示的外部上拉电阻：

```
pinMode(switchPin, INPUT_PULLUP);
```

## 模拟输入

模拟输入允许你在Arduino标记为A0至A5的任何特殊模拟输入引脚上测量0V和5V之间的电压。与数字输入和输出不同，在使用模拟输入时，不需要在设置中使用pinMode（）。

要读取模拟输入的值，请使用analogRead（）函数，提供要读取的引脚名称作为参数。与digitalRead（）不同，analogRead（）返回一个数字，而不仅仅是true或false。analogRead（）返回的数字是介于0和1,023之间的数字。结果为0表示0V，1,023表示5V。要将从analogRead（）返回的数字转换为实际电压，你需要将数字乘以5，然后除以1,023，或者可以将其除以204.6。这里是如何完成以上操作的Arduino C程序。

```
int raw = analogRead(A0);
float volts = raw / 204.6;
```

因为模拟输入的读数总是一个整数，所以变量raw的类型是一个int（整型）。但是，要将原始读数缩放为十进制数，变量需要为float（浮点）类型。

你可以将各种传感器连接至模拟输入端——例如，在本书的后面，你将学习如何使用光敏电阻组成的光传感器与模拟输入端连接（请参见章节“项目：Arduino家庭植物浇水器”）以及如何连接可变电阻（请参见“项目：一种恒温饮料冷却器”）。

## 模拟输出

数字输出允许你打开或者关闭某些东西（例如，LED）。但是，如果你想以分级的方式控制功率，你需要使用模拟输出。模拟输出对于控制LED的亮度或者电动机的速度是很有用的。正如你可能期望的，这本书在很多地方都会用到这个功能。

并非Arduino Uno的所有引脚都能够作为模拟输出——必须使用D3，D5，D6，D9，D10或者D11这些引脚。在Ardunio上，这些引脚的边上都有一个~标记。

要控制模拟输出，请使用analogWrite（）函数并且参数数字在0到255之间。值0表示0V（完全关闭），值255表示完全打开。

这里很容易将模拟输出看作是0V到5V之间的电压，如果在用作模拟输出的引脚和GND之间连接一个电压表，电压将确实是在0V和5V之间变化，该变化随着你改变analogWrite（）中所使用的值而变化。