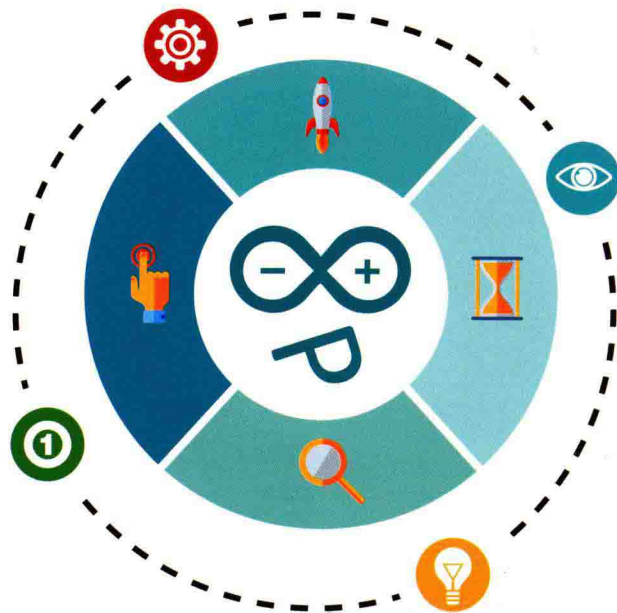


首本完全针对Processing和Arduino互动编程的书籍。  
配合传感器和Arduino平台，在Processing上实现各种有趣的  
案例和游戏开发例程。



电子与嵌入式系统  
设计丛书



# Processing与Arduino 互动编程

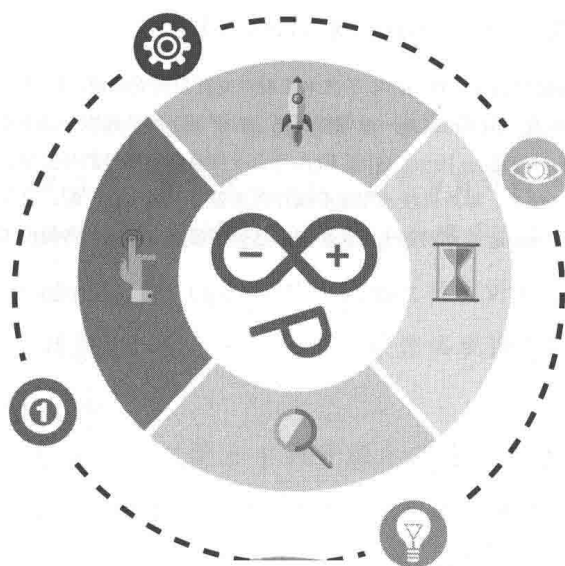
黄文恺 吴羽 编著



机械工业出版社  
China Machine Press



电子与嵌入式系统  
设计丛书



# Processing与Arduino 互动编程

黄文恺 吴羽 编著



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

Processing 与 Arduino 互动编程 / 黄文恺, 吴羽编著. —北京: 机械工业出版社, 2016.4  
(电子与嵌入式系统设计丛书)

ISBN 978-7-111-53526-3

I. P… II. ①黄… ②吴… III. ①程序设计 ②单片微型计算机—程序设计 IV. ①TP311.1  
②TP368.1

中国版本图书馆 CIP 数据核字 (2016) 第 073003 号

Processing 是一种开源编程语言, 专门为电子艺术和视觉交互设计而创建, 其目的是通过可视化的方式辅助编程, 并在此基础上表达数字创意。Arduino 是一款便捷灵活、方便上手的开源电子原型平台, 包含硬件 (各种型号的 Arduino 板) 和软件 (Arduino IDE)。Processing 和 Arduino 都是简单易学的开发平台, 前者是软件平台, 后者是硬件平台, 两者结合, 把电子技术、单片机技术和图形编程技术更好地融合在一起, 可以实现更多有趣的应用。本书提供了一些 Arduino 与 Processing 互动的例子, 包括读取各类传感器, 以及与各类传感器或控制装置结合的互动小游戏。

## Processing 与 Arduino 互动编程

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: 缪 杰

责任校对: 殷 虹

印 刷: 三河市宏图印务有限公司

版 次: 2016 年 4 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 11.5

书 号: ISBN 978-7-111-53526-3

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) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@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

Processing 是一种具有革命性和前瞻性的新兴计算机语言，该语言设计的初衷是为了使编程实现交互式图形更容易。该语言是以数字艺术为背景的程序设计语言，是 Java 语言的延伸，支持许多现有的 Java 语言架构，但在语法上简易许多。它具有跨平台的特性（支持 Windows、iOS 和 Android），对 OpenCV 和 Kinect 有良好的支持，除了可以很方便地创作震撼的视觉表现及互动媒体作品以外，还可以快速实现诸如图形处理和人工智能等高级应用。

Arduino 是一款便捷灵活、方便上手的开源电子原型平台，包含硬件（各种型号的 Arduino 板）和软件（Arduino IDE）。Arduino 能通过各种各样的传感器来感知环境，通过控制灯光、电机和其他的装置来反馈、影响环境。Arduino 同样适用于互动设计，来自全球的爱好者逐渐拓展其应用范围，使其在开源的无人机和 3D 打印机上都发挥了良好的作用。

Processing 和 Arduino 都是简单易学的开发平台，两者的结合把电子技术、单片机技术和图形编程技术更好地融合在一起，使编程设计更加人性化。传统的电子工程师想将获取的信号转换为图形界面显示是比较困难的，需要掌握较多的编程知识，并熟悉开发界面。而借助 Processing，可以非常简单地创建各种图形界面显示。机器人爱好者也可以借助 Processing 实现实时操作或远程控制机器人的电机或关节，借助 Kinect 或者 Xtion 可以实现机器人与操纵者的肢体动作同步。

本书提供了一些 Arduino 与 Processing 互动的例子，既包括读取各类传感器，也包括用各类传感器或控制装置结合 Processing 的互动小游戏。本书侧重于通过实例让读者更好地掌握 Arduino 和 Processing 的编程技巧，若读者想了解 Arduino 和 Processing 的入门知识，请参考笔者的《Arduino 开发实战指南·机器人卷》<sup>①</sup>。

---

① 本书已由机械工业出版社出版，书号 ISBN 978-7-111-46798-4。——编辑注

## 本书的主要内容及读者对象

本书是侧重于搭载硬件并进行实例编程的书籍，适合有一定基础的读者阅读，如电子信息、机械电子、互动技术、通信工程和计算机等专业的学生，也同样适合机器人爱好者阅读。全书分为入门篇、互动篇和游戏开发篇。第一篇介绍了 Processing 与 Arduino 的入门及通信方式，通过串口编程实现两者的互通。第二篇主要介绍如何读取一些传感器并在 Processing 上进行显示，也讲述了如何使用摇杆、重力传感器等来控制物体，还涉及如何使用 Processing 来控制电机的转速等知识。第三篇介绍利用一些传感器来开发互动小游戏的方法，如使用敲击传感器的“击鼓大师”，利用水银倾斜开关控制弹珠台，采用声音传感器来控制火柴人跨越障碍，以及用 PS2 摇杆来开发“太空大战小蜜蜂”的射击类游戏。

## 致谢

首先要感谢刘嘉杰、黄海锋、潘强、陈思强，他们牺牲了节假日时间帮助我整理书稿内容，并对每一个程序进行了验证。特别感谢韩杰威为本书制作了游戏所需要的各种图片。

其次要感谢“广州市教育局青少年科技教育计划”对本书的撰写、器材的购置提供的资助。

最后要感谢读者朋友，感谢您肯花费时间和精力阅读本书。

由于时间有限，书中难免存在疏漏与错误，敬请批评指正。希望有更多志同道合的朋友加入到互动技术的应用和机器人的制作与开发中来！

黄文恺

2016年1月于广州大学

# 目 录

## 前言

## 第一篇 入门篇

第 1 章 Processing 与 Arduino 快速入门 .....	2
1.1 Processing 快速入门 .....	2
1.2 Arduino 快速入门 .....	4
第 2 章 Processing 与 Arduino 通信 .....	8
2.1 串口简介 .....	8
2.2 Processing 串口编程 .....	10
2.3 Arduino 串口编程 .....	11
2.4 Processing 与 Arduino 通信编程 .....	13

## 第二篇 互动篇

第 3 章 声音振幅读取与显示 .....	22
3.1 声音传感器简介 .....	22
3.2 Arduino 读取声音传感器 .....	22
3.3 Processing 绘制振动条 .....	24
3.4 声音振幅显示 .....	25
第 4 章 旋转编码器控制播放音乐 .....	28
4.1 旋转编码器简介 .....	28

4.2	Arduino 读取编码器值	29
4.3	Processing 读取音乐文件	31
4.4	调节音量大小	32
<b>第 5 章</b>	<b>跟随环境光变化的太阳</b>	<b>35</b>
5.1	光敏传感器简介	35
5.2	Arduino 读取光敏传感器	35
5.3	Processing 绘制太阳	37
5.4	根据亮度调节太阳的颜色	39
<b>第 6 章</b>	<b>超声波测距传感器的读取与显示</b>	<b>41</b>
6.1	超声波测距传感器简介	41
6.2	Arduino 读取超声波传感器	42
6.3	Processing 绘制距离值与提示	43
6.4	超声波读取值显示	45
<b>第 7 章</b>	<b>控制彩色 LED 灯</b>	<b>47</b>
7.1	RGB 彩色 LED 模块简介	47
7.2	Arduino 控制 LED 灯亮度和颜色	47
7.3	Processing 进度条绘制	48
7.4	调节彩色 LED 灯	50
<b>第 8 章</b>	<b>温湿度检测提示</b>	<b>54</b>
8.1	温湿度传感器简介	54
8.2	Arduino 读取温湿度传感器	54
8.3	Processing 绘制温湿度显示计	57
8.4	温湿度检测显示	58
<b>第 9 章</b>	<b>液位检测与提示</b>	<b>65</b>
9.1	液位传感器简介	65

9.2	Arduino 读取液位传感器	65
9.3	Processing 绘制液位高度与阈值	66
9.4	液位检测与液位阈值提示	67
<b>第 10 章</b>	<b>摇杆控制坦克</b>	<b>69</b>
10.1	PS2 摇杆简介	69
10.2	Arduino 读取摇杆数值	70
10.3	Processing 绘制坦克和键盘控制移动	70
10.4	摇杆控制坦克移动	73
<b>第 11 章</b>	<b>平衡球</b>	<b>78</b>
11.1	MPU6050 传感器简介	78
11.2	Arduino 读取 MPU6050 传感器	79
11.3	Processing 绘制平衡球和边界	82
11.4	控制平衡球	83
<b>第 12 章</b>	<b>电机控制</b>	<b>89</b>
12.1	L298N 电机驱动模块简介	89
12.2	Arduino 电机转速控制	90
12.3	Processing 绘制速度控制条	92
12.4	控制电机转速	95
<b>第三篇</b>	<b>游戏开发篇</b>	<b>101</b>
<b>第 13 章</b>	<b>击鼓大师</b>	<b>102</b>
13.1	设计思想	102
13.2	物料清单	102
13.3	电路接线	103
13.4	软件设计	103
13.4.1	功能分析与实现	103



13.4.2	程序流程图	104
13.4.3	难点与技巧	105
13.4.4	界面设计	106
13.5	游戏使用说明	107
13.6	源代码	107
<b>第 14 章</b>	<b>变脸弹珠台</b>	<b>124</b>
14.1	设计思想	124
14.2	物料清单	124
14.3	电路接线	124
14.4	软件设计	125
14.4.1	功能分析与实现	125
14.4.2	程序流程图	126
14.4.3	界面设计	127
14.5	游戏使用说明	128
14.6	源代码	128
<b>第 15 章</b>	<b>奔跑的火柴人</b>	<b>141</b>
15.1	设计思想	141
15.2	物料清单	141
15.3	电路接线	141
15.4	软件设计	142
15.4.1	功能分析与实现	142
15.4.2	程序流程图	143
15.4.3	难点与技巧	143
15.5	界面设计	145
15.6	游戏使用说明	146
15.7	源代码	146
<b>第 16 章</b>	<b>太空飞船大战小蜜蜂</b>	<b>153</b>
16.1	设计思想	153

16.2	物料清单 .....	153
16.3	电路接线 .....	153
16.4	软件设计 .....	154
16.4.1	功能分析与实现 .....	154
16.4.2	程序流程图 .....	155
16.4.3	难点与技巧 .....	155
16.5	界面设计 .....	160
16.6	游戏使用说明 .....	161
16.7	源代码 .....	161

# 第一篇 入门篇

第 1 章 Processing 与 Arduino 快速入门

第 2 章 Processing 与 Arduino 通信

# 第 1 章

## Processing 与 Arduino 快速入门

### 1.1 Processing 快速入门

Processing 是专门为开发面向图形的应用程序（Visually Oriented Application）而设计的一种具有革命性和前瞻性的新兴编程语言。Processing 的创造者——麻省理工学院（MIT）媒体实验室的 Casey Reas 和 Benjamin Fry 将它看作是一个代码素描本。该语言特别擅长算法动画和即时交互反馈。近年来，越来越多的开发者将其用于交互动画、复杂数据可视化、视觉设计、原型开发和制作等方向。

Processing 是 Java 语言的延伸，它使用了更简化的语法和图形编程模型，增加了更多贴心和人性化的设计。由于其简单易用的特性，吸引了众多非专业的编程人员，包括数字艺术、电子互动、机器人控制等专业。与其他复杂的软件开发环境相比，Processing 的开发环境非常简洁明了，如图 1-1 所示。

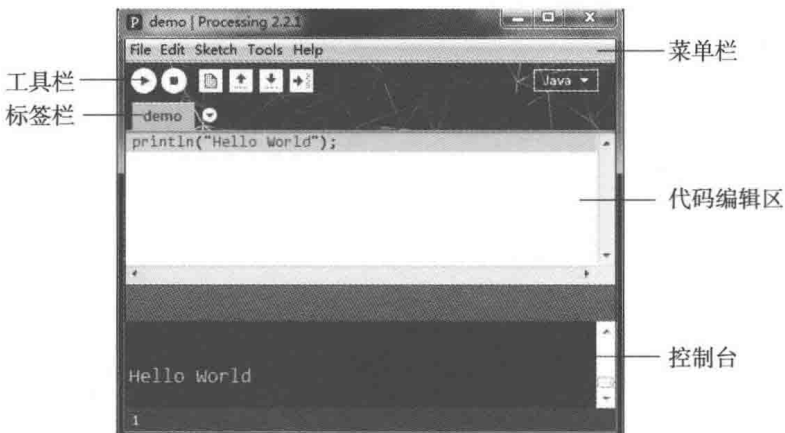



图 1-1 Processing 开发界面


打开 IDE (Processing 开发环境), 最大面积的空白区就是文本输入框, 上面有一排功能按钮, 具体说明如下。

: 程序的检查和编译, 并运行。

: 停止运行。

: 新建一个 Processing 程序。

: 打开文件。

: 保存文件。

: 输出程序。

Processing 简单易用的特点, 以及其丰富的拓展, 可让学习者把精力放在创造很多富有想象力的作品上, 而不是耗费大量的时间在基础的、枯燥无味的编程学习上。

Processing 和 Arduino 的搭配可以让图形化界面和硬件产生互动。添加 OpenCV 库, 可以实现人脸识别等各种高级的图形处理功能, 添加 Kinect 库, 可以识别人体的肢体动作并进行交互。因为源自 Java, 所以大量的 Java 库都可以添加进来直接调用, 比如 Box2D、Unity 等引擎都可以在开发时调用。除此之外, 还可以借助于人工智能库 (AI for 2D Games) 开发各种游戏。这些丰富的拓展让使用 Processing 设计出的作品充满了魅力, 富有想象力。

一个标准的 Processing 主程序结构格式如下所示。

```
void setup()  
{  
  // 用于程序初始化设置, 只执行一次  
}  
void draw()  
{  
  // 主程序, 添加用户的代码。默认每秒执行 60 次  
}
```

setup() 函数中的程序代码在程序启动后只执行一次, 一般用来进行初始化设置、配置一些变量的初始参数等。

Processing 执行完一次 `setup()` 函数后就会不间断地运行 `draw()` 函数。在 `draw()` 函数里可绘制要显示的内容或执行各种操作，默认以每秒 60 帧（60fps）的速度不断更新并重绘，直到执行“停止”的命令或关闭窗口为止。`draw()` 每运行一次相当于在窗口绘制一个新的帧，运行一次后再重新执行下一次。

关于 Processing 的语法，读者可以访问 <https://Processing.org> 的参考 (Reference)，本书不对其语法进行详述。

---

**注意** 由于 Processing 基于 Java，请确定您的计算机已经配置 Java 环境，否则 Processing 不能正常工作。

---

## 1.2 Arduino 快速入门

Arduino 是一个开放源代码硬件项目平台，由意大利的 Massimo Banzi 设计。该平台面世之后，很快就风靡全球。它的开发环境可以在 Windows、Macintosh OSX、Linux 三大主流操作系统上运行。该平台的开发板基于 AVR 指令集的单片机作为处理核心，是一块具有 USB 接口 Simple I/O 接口板。该平台的软件开发使用类似 Java、C 语言的 IDE 集成开发环境。

Arduino 开发板虽然使用了 AVR 的单片机，但它简化了单片机工作的流程，对 AVR 库进行了二次编译封装，将复杂的单片机底层代码封装成简单实用的函数，使用者无需关心单片机编程的烦琐细节，如寄存器、地址指针等，从而大大降低了单片机系统开发难度。相比 ICCAVR、WINAVR 和 CAVR 等专业且复杂难懂的开发平台，Arduino 特别适合初学者、学生、艺术家和一些业余爱好者使用。随着 Arduino 的不断完善和强大，也有专业技术人员用它来设计开发产品原型。

以 Arduino UNO 板（见图 1-2）为例，它拥有以下的资源：

- 数字引脚：0 ~ 13
- 串行通信：0 作为 RX，接收数据；1 作为 TX，发送数据
- 外部中断：2, 3

- PWM 输出: ~ 3, ~ 5, ~ 6, ~ 9, ~ 10, ~ 11
- SPI 通信: 10 作为 SS, 11 作为 MOSI, 12 作为 MISO, 13 作为 SCK
- 板上 LED: 13
- 模拟引脚: A0 ~ A5 (在引脚号前加 A, 以与数字引脚区分)
- TWI 通信: A4 作为 SDA, A5 作为 SCL

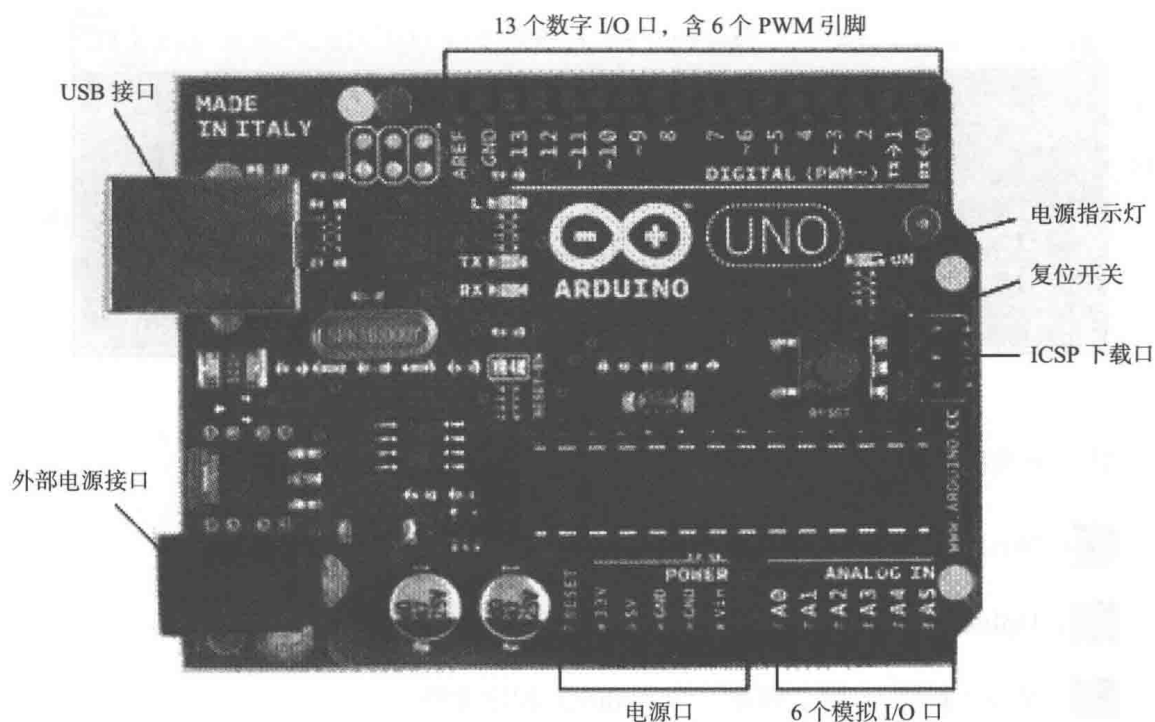


图 1-2 Arduino UNO 的资源


Processing 是纯软件平台, 无法直接读取各种传感器的值, 更无法直接控制各种机电设备。而借助 Arduino, Processing 可以读取各种传感器的数值, 控制各种机电装置, 可扩展用于控制智能家居、无人机、机器人等各种硬件实体。


Arduino 的开发环境与 Processing 非常相似, 如图 1-3 所示。它除了包含 File (文件)、Edit (编辑)、Sketch (概述)、Tools (工具)、Help (帮助) 这 5 类菜单外, 在菜单栏下方还提供了 5 个常用的快捷菜单按钮。





图 1-3 Arduino 开发界面


这 5 个快捷菜单按钮的具体功能如下。

: Verify (校验), 用于完成程序的检查和编译。

: Upload (上传), 用于将编译完成的程序上传到 Arduino 控制板。

: New (新建), 用于新建一个 Arduino 程序文件。

: Open (打开), 用于打开一个已存在的 Arduino 程序文件, 其文件后缀名为 .pde, 1.0 版本以后的文件后缀名为 .ino。

: Save (保存), 用于保存当前的程序文件。

一个标准的 Arduino 主程序结构格式如下所示:

```
void setup()  
{  
  // 初始化, 设置各种变量参数等, 只执行一次
```



```
}  
void loop()  
{  
  // 添加主程序代码，执行完毕后继续循环执行  
}
```

与 Processing 类似，Arduino 分为两部分 `setup()` 函数和 `loop()` 函数。`setup()` 函数主要用于程序初始化，设置各种变量的初始值等，单片机启动后只执行一次；`loop()` 函数是主函数，单片机执行完 `setup()` 后会一直循环的运行该函数，直到断电为止。和 Processing 不同，单片机没有绘制图形的概念，`loop()` 函数不会按照一定的速率运行，该函数执行完里面的代码后会立刻进入下一次循环。

关于 Arduino 的基本语法，读者可以访问 [www.Arduino.cc](http://www.Arduino.cc) 的参考 (Reference)，或阅读笔者撰写的另一本基础教程《Arduino 开发实战指南·机器人卷》，本书不再对 Arduino 的基本语法进行详述。