

```
class Scheduler:
```

```
def __init__(self):
```

```
# initialize with 0 event
```

```
self.events = []
```

Python 微控制器编程

从零开始 使用MicroPython

```
# execute function after a
```

```
def schedule(self, delay, fu
```

```
if delay <= 0:
```

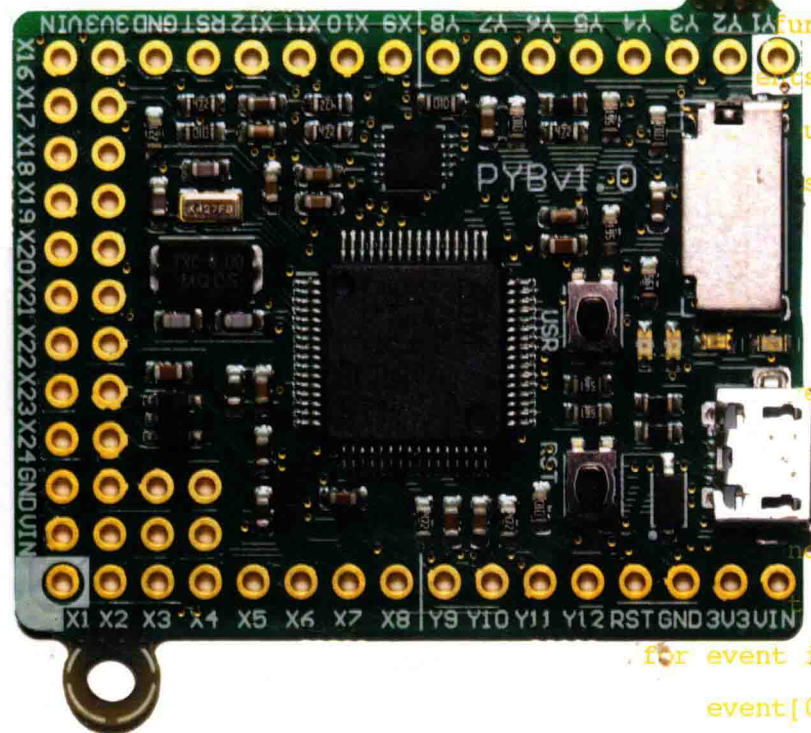
```
# execute function i
```

```
function()
```

[美] 唐纳德·诺里斯(Donald Norris)
别志松 李 杨

著
译

```
else:
```



```
function
```

```
self.events.append([
```

```
value with sm
```

```
self.events.sort(key
```

```
events still
```

```
self.events >
```

```
event
```

```
function = se
```

```
delay tim
```

```
for event in self.ev
```

```
event[0] -= dela
```

```
# wait required dela
```

```
self.schedule(delay)
```

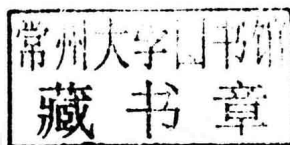
```
self.execute function
```

```
self.execute function()
```

清华大学出版社

Python 微控制器编程从零开始 使用 MicroPython

[美] 唐纳德·诺里斯(Donald Norris) 著
别志松 李 杨 译



清华大学出版社

北 京

Donald Norris

Python for Microcontrollers: Getting Started with MicroPython

EISBN: 978-1-25-964453-5

Copyright © 2017 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education and Tsinghua University Press Limited. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Translation copyright © 2018 by McGraw-Hill Education and Tsinghua University Press Limited.

版权所有。未经出版人事先书面许可，对本出版物的任何部分不得以任何方式或途径复制或传播，包括但不限于复印、录制、录音，或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和清华大学出版社有限公司合作出版。

此版本经授权仅限在中国大陆地区销售、不能销往中国香港、澳门特别行政区和中国台湾地区。

版权©2018 由麦格劳-希尔(亚洲)教育出版公司与清华大学出版社有限公司所有。

北京市版权局著作权合同登记号 图字：01-2018-0336

本书封面贴有 McGraw-Hill Education 公司防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

Python微控制器编程从零开始 使用MicroPython/(美)唐纳德·诺里斯(Donald Norris)著,别志松,李杨译. —北京:清华大学出版社,2018

书名原文: Python for Microcontrollers: Getting Started with MicroPython

ISBN 978-7-302-50649-2

I. ①P… II. ①唐… ②别… ③李… III. ①软件工具—程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2018)第 155746 号

责任编辑:王军于平

装帧设计:思创景点

责任校对:牛艳敏

责任印制:沈露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:148mm×210mm 印 张:8.875 字 数:248千字

版 次:2018年8月第1版 印 次:2018年8月第1次印刷

定 价:49.80元

产品编号:078160-01

译者序

MicroPython 语言是一种简洁高效的编程语言，它源于 Python 3 语言，被优化后用在微控制器上运行。该语言既遵循了常规的 Python 语言的规范，易于上手且有强大的库的支撑，又可以让你轻松地将代码移植到微控制器上，更易于实现底层的操作。MicroPython 语言正越来越受到国内外嵌入式系统开发者和研究人员的青睐。

本书作者 Donald Norris 具有数十年的电子电路开发经验，编写并出版了多本有关电子工程开发方面的著作。本书是 Norris 的倾心之作，也是第一本介绍使用 MicroPython 语言进行微控制器编程的书籍。不管是 MicroPython 语言的初学者，还是微控制器的资深开发者，都会从中领略到使用 MicroPython 语言的乐趣。

本书言简意赅、细致全面，既有 MicroPython 编程开发的基本原理和概念，也有实用性的开发示例。读者在阅读的过程中，不妨着手一试！

在这里要感谢清华大学出版社的编辑们，在本书的翻译和出版过程中得到他们的大力支持和帮助。本书由别志松、李杨翻译，参与翻译的还有于飞、王迎、李争等人。

对于本书的翻译，译者始终保持着“敬畏之心”，但难免还会存在纰漏，恳请广大读者批评指正。

译者

作者简介

Donald Norris 拥有电子工程学位和生产管理专业的 MBA 学位。目前他在南新罕布什尔大学从事 IT 相关的本科和研究生教学工作。他也在这里创立并教授机器人方面的课程。作为一名兼职教授，他在多所学院和大学拥有超过 36 年的教学经验。

Norris 先生从美国海军文职人员的岗位退休，他专注于核潜艇声呐相关的研究和高级数字信号处理相关的工作。退役后，他从事了 22 年的专业软件开发工作，使用 C、C#、C++、Python、Node.js 和 Java 语言进行程序设计，并从事过 5 年 IT 安全咨询工作。

Norris 先生创办了一家名为 Norris Embedded Software Solutions 的咨询公司，专注于开发基于微处理器和微控制器的应用程序解决方案。他认为自己是一个不断创新的技术爱好者和极客，并一直热衷于通过实验去验证自己的新想法。他拥有飞行执照、热爱摄影、热爱业余无线电、酷爱跑步。

除了本书，Norris 先生也是 *Raspberry Pi Projects for the Evil Genius*、*Programming the Intel Edison: Getting Started with Processing and Python* 以及其他 3 本 TAB 书籍的作者。

致 谢

首先我要感谢我的妻子 Karen，她对我所有的实验充满耐心，并坚持“讨论”了本书所有的工程。

感谢编辑 Michael McCabe 的大力支持，以及编辑助理 Amy StoneBraker 的支持。

另外，感谢项目经理 Anju Joshi 所做的细致工作。

最后，我要感谢 www.micropython.org 网站的 Damien George 博士，他的杰出和努力使他创造了 MicroPython 语言，并使 Pyboard 市场化。

前 言

这是第一本为介绍 MicroPython 语言编写的书。尽管在网络上，特别是 www.micropython.org 网站也有一些优秀的教程，但是我发现对于不愿意就在线教程投入时间和精力和精力的读者来说，读书是一种引人入胜的学习形式。另外，不需要非难在线的作者，一本编写和组织得很好的书通常对读者理解新的创新性技术，如 MicroPython，是一种好的途径。

在本书中我也展示了一些工程，应该能吸引大部分读者。一个是机器人车，另一个是组建全球定位系统(Global Positioning System, GPS)语句解析器，两个关于如何使用 Pyboard 开发板的工程，以及一个为打网球的读者提供的擦网球探测器。所有这些工程的主旨都是学习如何使用外部设备和传感器与 MicroPython 微控制器接口联系。

本书包含 3 个运行 MicroPython 的独立模块。第一个是 Pyboard，由 Damien George 博士设计，他也是 MicroPython 语言的创造者。所有的介绍脚本和本书的工程都在 Pyboard 上运行。所讨论的另外两种开发板是 ESP8266 和 WiPy。这些开发板运行 MicroPython，但是它们也有内置的无线电系统，这是 Pyboard 所不具备的。实际上，WiPy 有 4 个独立的无线电系统，如果需要一个连接各种无线协议的模块，它非常适合。在第 10 章我们将学习关于它的所有内容。

如果阅读了所有这些介绍材料，并且完成了大部分工程，读者在 MicroPython 的知识和经验方面应该会有很大进步。我个人在设计并最终实现了工程后，通常收获颇多。工程往往进展得很好，但

VIII Python 微控制器编程从零开始 使用 MicroPython

是有时也会问题重重。然而，这就是我所理解的实验的乐趣。著名的爱因斯坦教授曾经说过：“没有犯过错误的人也没尝试过新的事物”。

我想提醒所有经验丰富的 Linux 开发者，在深入学习 MicroPython 之前至少要回顾本书开始的章节，因为虽然 MicroPython 是在 Python 3 之后创建的，但是有一些理应认识到的“陷阱”。我尝试在整本书提供很多有用的提示和技术来帮助读者学习 MicroPython。

因此，直接开始 MicroPython 之旅吧！

Donald Norris

目 录

第 1 章 概述	1
1.1 MicroPython 简介	1
1.2 设计思想	2
1.3 MicroPython 探究	2
1.4 面向对象编程和一些 Python 基础	4
1.4.1 面向对象的概念	4
1.4.2 建模一个机器人	7
1.4.3 Python 基础	9
1.4.4 Robot 类	12
1.4.5 子类	13
1.4.6 使用接口	14
1.4.7 集成的 Robot 工程设计和测试	16
1.4.8 动态绑定	19
1.5 使用基于 Pyboard 的 MicroPython	22
1.6 bare-metal 方法	25
1.7 小结	26
第 2 章 Python 简介	29
2.1 Pyboard 硬件	29
2.2 通用微控制器程序开发	33

2.2.1	中断	33
2.2.2	控制 Pyboard	36
2.2.3	Python 测试程序	39
2.3	闪烁 LED	40
2.3.1	PyBlink	40
2.3.2	PyBlink_MP	41
2.4	硬件中断演示	42
2.5	计时器	45
2.6	其他 Pyboard 硬件	48
2.6.1	ADC	49
2.6.2	DAC	50
2.7	小结	56
第 3 章	接口、文件和库	57
3.1	脉冲宽度调制	57
3.2	位串行端口	59
3.2.1	UART 串行协议	59
3.2.2	I2C 串行协议	65
3.2.3	SPI 串行协议	67
3.3	目录结构	69
3.4	导入模块	70
3.4.1	使用文件导入	71
3.4.2	从 PYBFLASH 子目录导入模块	72
3.4.3	使用 SD 卡导入	74
3.5	启动过程	75
3.6	基于 LED 的错误报告	76
3.7	库	76
3.7.1	标准库	77
3.7.2	自定义 Python 库	77

3.7.3	Pyboard 专用库和类	78
3.7.4	MicroPython 库	79
3.8	小结	80
第 4 章	擦网球探测器	81
4.1	初始设计	82
4.2	工程需求	82
4.2.1	原型	84
4.2.2	加速度计	85
4.2.3	低级 I2C 命令演示	90
4.2.4	倾角计	92
4.3	硬件设计	94
4.4	在网球场试验	102
4.5	最终程序	104
4.6	小结	105
第 5 章	LCD 和触摸传感器开发板	107
5.1	LCD 板规格说明	108
5.2	初始 LCD 模块操作	109
5.3	LCD 图形演示	114
5.4	对 LCD 控制器使用外部命令	115
5.5	触摸控制器	116
5.5.1	电容感应	117
5.5.2	LCD 模块触摸传感器原理图和 MPR121 寄存器	118
5.5.3	MPR121 驱动软件	120
5.6	触摸传感器初步测试	122
5.6.1	由触摸板控制的 LED	123
5.6.2	LCD 和触摸传感器示例	126
5.6.3	使用 LCD 模块 GPIO 管脚	128
5.7	MPR121 PWM	133

5.8	小结	135
第 6 章	AMP 音频开发板	137
6.1	组装 AMP 开发板	137
6.2	AMP 开发板电路	139
6.2.1	音频输入电路	141
6.2.2	音频输出电路	145
6.3	初步测试	148
6.4	播放 MP3 文件	154
6.5	音频输入演示	155
6.6	小结	157
第 7 章	自动机器人车	159
7.1	组建机器人车平台	159
7.2	一个标准模拟伺服	160
7.3	CR 伺服	163
7.4	机器人车电源	164
7.5	电路布线说明	166
7.6	机器人车软件	169
7.7	初步测试运行	171
7.8	避障演示	178
7.9	小结	180
第 8 章	基于 Pyboard 的 GPS	181
8.1	概述	181
8.2	GPS 简史	181
8.3	GPS 工作基本原理	182
8.4	Ultimate GPS 接收机	188
8.5	初步 GPS 接收机测试	189
8.6	GPS 接收机 UART 通信	192

8.7	NMEA 协议	197
8.7.1	纬度和经度格式	199
8.7.2	解析 GPS 消息	199
8.8	MicroPython GPS 解析器	200
8.9	小结	204
第 9 章	ESP8266	205
9.1	ESP8266 SMT ESP-12E 模块	205
9.2	ESP8266 HUZAZH 开发板	208
9.3	ESP8266 软件	212
9.3.1	在 HUZAZH ESP8266 开发板上安装 MicroPython	213
9.3.2	探索基于 ESP8266 HUZAZH 开发板的 MicroPython	215
9.4	实验	218
9.4.1	模拟-数字转换器	218
9.4.2	1 引线温度传感器	221
9.4.3	NeoPixel 演示	225
9.4.4	APA102 演示	228
9.5	webREPL	232
9.6	小结	233
第 10 章	WiPy	235
10.1	WiPy 规范说明	235
10.2	Wi-Fi 模式	237
10.2.1	站点	237
10.2.2	接入点	237
10.2.3	直连	238
10.3	WiPy 扩展板	238
10.4	创建初始 WiPy 网络连接	243
10.5	FileZilla	249
10.6	站点操作	251

XIV Python 微控制器编程从零开始 使用 MicroPython

10.7	启动过程和恢复文件系统	253
10.8	Pymakr	255
10.9	小结	257
第 11 章	MicroPython 的现在和未来	259
11.1	MicroPython 语言	259
11.2	硬件平台	261
11.2.1	LoPy	261
11.2.2	LoRa 无线电系统	262
11.2.3	SiPy	264
11.2.4	Sigfox 与 LoRa	265
11.3	小结	265

第 1 章

概 述

本书是关于 MicroPython 语言的技术书籍，它包括使用 Pyboard、ESP8266 或 WiPy 开发板进行开发的 8 个工程。在后续章节将详细介绍使用这些开发板开发的各工程；不过，我将使用 Pyboard 介绍 MicroPython 语言，因为在 Pyboard 开发板中安装了 MicroPython 语言，并且可以使用了。

1.1 MicroPython 简介

MicroPython 是 Python 3 语言的一个变种，专门用于在存储器受限的微控制器上运行。其思想来自于剑桥大学的 Darrien P. George 博士，他与一个众筹团队一起于 2013 年开发了 Pyboard 产品的第一个版本。该版本是一个专门设计的硬件开发板，作为操作系统(OS)运行 MicroPython。George 博士在 2014 年的 Python 大会(PyCon)上提出了 MicroPython 1.0。George 博士也建立了 micropython.org 网站，来支持 MicroPython 语言和 Pyboard。在该网站上可以找到关于 MicroPython 和 Pyboard 的最新信息。该网站上也有几个非常活跃的用户论坛，我发现它们对于回答在 MicroPython 和 Pyboard 中遇到的问题很有用。

1.2 设计思想

首先，我希望介绍我自己关于正确使用 MicroPython 和 Pyboard 的设计思想。应当注意，这里的介绍只代表我个人的思想，而不是 Damien George 的方法，虽然我认为二者并无太大的差别。

基于 Pyboard 实现的 MicroPython 被设计用于快速有效地创建嵌入式系统控制程序。嵌入式系统可以简单定义为任何需要使用微控制器以满足工程需求的工程，不管有无人为干预。典型的嵌入式系统通常使用传感器，有时使用电子机械传动器与微控制器接口。它们可能附接有人机接口设备，便于人类交互，但是更多是自动的，为单机运行或连接至云进行远程访问。

嵌入式系统通常不需要通用应用程序，如终端程序或数据库程序。MicroPython 轻易不支持添加此类高级应用程序。如果工程需要使用这些类型的应用程序，很可能是使用了不合适的开发板。Raspberry Pi 和 BeagleBone Black 或许更适合你的工程，因为它们都支持强大的 Linux 发行版，易于使用，且能够配置成满足高级需求。

总而言之，MicroPython 和 Pyboard 最适合使用高级语言快速开发嵌入式应用程序。这些应用程序在运行时非常高效，且比传统开发方法使用更少的存储器。

1.3 MicroPython 探究

在开始深入探究 Python 和 MicroPython 错综复杂的细节之前，我想应该展示二者的徽标(logo)。图 1-1 显示了官方 Python 语言的徽标，它只能与 python.org 网站上的 Python 组织所承认和支持的 Python 语言实现的程序关联。非正式场合下，该徽标俗称双蛇徽标。



图 1-1 官方 Python 徽标

而 MicroPython 的徽标如图 1-2 所示, 看起来更不正式, 同时也通过徽标中的物理芯片突出反映了其对微处理器的使用。

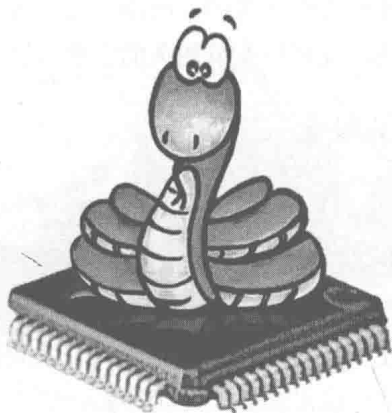


图 1-2 MicroPython 徽标

Python 名称有意思的一点是它并不是基于蟒蛇(python)这一物种, 而是来源于 Python 创造者 Guido van Rossum 对英国喜剧组合 Monty Python 的兴趣。

1.4 节将介绍面向对象(Object-Oriented, OO)的概念, 并且讨论 Python 语言的主要特点和初衷。这对于理解为何创建 MicroPython, 以及为何它对于 Python 社区是一个真正重要的补充来说非常必要。如果你使用 Python 和面向对象技术非常得心应手, 只是希望开始使用 MicroPython, 可以随意跳过 1.4 节。