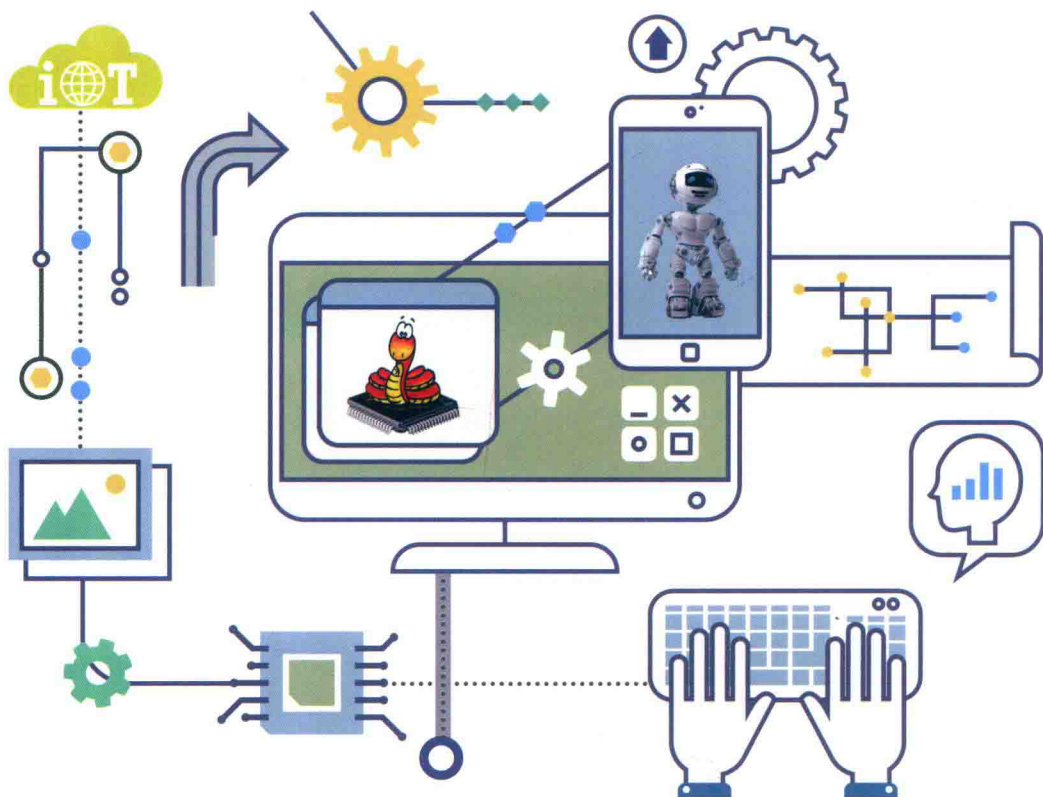
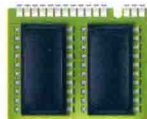




MicroPython

入门指南

邵子扬 编著

智能硬件设计丛书

MicroPython 入门指南

邵子扬 编著

電子工業出版社

Publishing

Industry

内 容 简 介

MicroPython 是近年国外开源硬件（也是智能硬件）范围最热门的主题之一，它使用 Python 语言在嵌入式中编程，不需要了解底层寄存器、数据手册、厂家的库函数，大部分外设和常用功能都有自己的库，使开发和移植变得容易和快速。MicroPython 已经可以真正用于开发产品，这是其迅速吸引智能硬件开发者的原因。本书带领初学者逐步了解 MicroPython 的基础知识、硬件平台、开发技巧；采用大量浅显易懂的实例，使读者在实践中快速入门，学习如何使用 MicroPython 做出很酷的东西，学会简单、有趣、强大的硬件和软件开发。

这是一本实用的书，即使你没有任何编程知识，这本书也将带你踏上从第一步到高级项目的旅程。

本书的读者可以是电子工程师、大中专院校信息类专业学生、电子爱好者，甚至中学生。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

MicroPython 入门指南/邵子扬编著. —北京：电子工业出版社，2018.1

（智能硬件设计丛书）

ISBN 978-7-121-32846-6

I. ①M… II. ①邵… III. ①软件工具—程序设计—指南 IV. ①TP311.561-62

中国版本图书馆 CIP 数据核字（2017）第 247847 号

策划编辑：曲 昕

责任编辑：曲 昕

印 刷：北京季蜂印刷有限公司

装 订：北京季蜂印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：720×1 000 1/16 印张：18 字数：294 千字

版 次：2018 年 1 月第 1 版

印 次：2018 年 1 月第 1 次印刷

定 价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254468，quxin@phei.com.cn。

前言

PREFACE

MicroPython 是近年开源社区中最热门的项目之一，它功能强大，使用简单，是创客、DIY 爱好者、工程师最好的工具，也可以用在专业开发中。

相比另一个创客神器 Arduino，MicroPython 使用更加简单、方便，入门更快，性能也更好，更加适合初学者。它无须复杂的设置，不需要安装特别的软件和额外的硬件，也不用编译和下载，只要一个 USB 线，使用任何文本编辑器就可以进行编程。大部分硬件的功能，使用一个命令就能驱动，不用了解硬件底层就能快速开发，对于产品原型设计、软件移植非常有好处，让开发过程变得轻松，充满乐趣。和传统开发方法相比，MicroPython 开发产品原型的速度更快，程序也更容易实现模块化，更方便进行维护。和其他类似软件相比，MicroPython 是可以真正用在产品开发中的软件。

MicroPython 以其开放的架构和 MIT 授权方式，在很短时间内就风靡世界，全世界有很多爱好者用它做出各种产品和有趣的应用。现在 MicroPython 已经被移植到了多种硬件平台上，如 STM32、ESP8266、ESP32、dsPIC33、RTL8195、CC3200 等，让我们有很多选择。

本书可以作为学习 MicroPython 的入门读物，也可以作为工具书，查看各种 API 的用法。本书先介绍 MicroPython 的起源，然后介绍 MicroPython 的基础知识和硬件平台，再重点介绍 STM32 和 ESP8266 上 MicroPython 的应用方法以及 API，最后介绍了几个有趣的应用。通过本书读者可以快速了解并掌握 MicroPython。

本书作者作为 MicroPython 中文社区站长，长期致力于 MicroPython 的研究和推广。本书的内容是由中文社区活动资料、社区的中文教程、官网英文社区、作者翻译的官网文档等整理而来，希望通过这本书，让广大爱好者和工程师可以了解到国外最新的技术，也希望和更多的爱好者一起交流，共同进步。书中难免存在不足和错误，请广大读者指正。

编著者

2017 年 10 月

目 录

CONTENTS

第 1 章	MicroPython 简介	001
1.1	MicroPython 是什么.....	001
1.2	MicroPython 的历史.....	001
1.3	MicroPython 的特点.....	004
1.4	授权.....	005
第 2 章	基础知识	006
2.1	Python3 和 MicroPython.....	006
2.2	MicroPython 的系统结构.....	007
2.3	安装驱动.....	008
2.4	常用终端软件.....	010
2.5	MicroPython 的 REPL.....	014
2.5.1	REPL 的快捷键.....	014
2.5.2	使用 help() 函数.....	015
2.5.3	查看模块包含的变量和函数.....	019
2.6	MicroPython 官方固件.....	020
第 3 章	硬件平台介绍	023
3.1	pyboard.....	023
3.2	ESP8266.....	028
3.3	其他可以运行 MicroPython 的硬件.....	032

第 4 章

pyboard 的使用

041

4.1

快速指南

041

4.2

从闪灯开始

047

4.3

按键的使用

049

4.4

GPIO 的使用

050

4.4.1

GPIO 的其他函数

052

4.4.2

GPIO 的别名

055

4.4.3

直接用端口名称

057

4.5

ADC 模数转换

057

4.6

DAC 数模转换

058

4.7

定时器 (Timer)

063

4.8

UART

068

4.9

I2C

071

4.10

SPI

074

4.11

外中断

076

4.12

USB_VCP

078

4.13

CAN

080

4.14

文件操作

084

4.15

小结

085

第 5 章

pyboard 的 Python 标准库和微库

086

5.1

内置函数

087

5.2

数组 (array)

088

5.3

复数运算 (cmath)

089

5.4

垃圾回收 (gc)

091

5.5

数学计算 (math)

092

5.6

数据流事件 (select)

095

5.7	系统功能 (sys)	096
5.8	binary/ASCII 转换 (ubinasii)	098
5.9	集合和容器类型 (ucollections)	099
5.10	哈希算法库 (uhashlib)	100
5.11	堆队列算法 (uheapq)	101
5.12	输入/输出流 (uio)	102
5.13	JSON 编码解码 (ujson)	103
5.14	“操作系统”基本服务 (uos)	103
5.15	正则表达式 (ure)	108
5.16	socket 模块 (usocket)	110
5.17	压缩和不压缩原始数据类型 (ustruct)	114
5.18	时间函数 (utime)	115
5.19	zlib 解压缩 (uzlib)	120

第 6 章 | MicroPython 特别库..... 121

6.1	简化的 BTree 数据库 (btree)	121
6.1.1	函数	121
6.1.2	方法	122
6.1.3	常量	123
6.1.4	例程	123
6.2	帧缓冲区操作 (frambuf)	124
6.2.1	类	124
6.2.2	构造函数	125
6.2.3	绘制基本形状	126
6.2.4	绘制文字	126
6.2.5	其他方法	126
6.2.6	常量	127
6.3	硬件相关函数 (machine)	127
6.3.1	复位函数	127

6.3.2	中断相关函数	128
6.3.3	功率管理	128
6.3.4	常量	129
6.3.5	Class	129
6.4	访问和控制 MicroPython 内部 (micropython)	138
6.5	使用网络 (network)	140
6.5.1	class CC3K	141
6.5.2	class WIZNET5K	143
6.6	结构化访问二进制数据 (ctypes)	144
6.6.1	定义数据结构层	145
6.6.2	模块内容	146
6.6.3	结构说明和实例化结构对象	147
6.6.4	结构对象	147
6.6.5	限制	148
6.7	线程	148
6.7.1	基本函数	149
6.7.2	使用方法	149

第 7 章 | pyboard 专用模块 (pyb)

7.1	时间相关功能	151
7.2	复位功能	152
7.3	中断相关函数	153
7.4	功耗管理	153
7.5	其他函数	154
7.6	类	158
7.6.1	加速度传感器 (Accel)	158
7.6.2	方法	158
7.6.3	例程	159
7.7	ADC	159

7.7.1	构造函数	159
7.7.2	方法	160
7.7.3	ADCA11	161
7.8	数模转换 (DAC)	163
7.8.1	构造函数	163
7.8.2	方法	163
7.9	三线舵机驱动 (servo)	164
7.9.1	构造函数	165
7.9.2	方法	165
7.10	pyb 和 machine 的区别	166

第 8 章 | ESP8266 167

8.1	快速指南	168
8.2	ESP8266 专用模块 esp	176
8.3	MicroPython 标准模块和 machine 模块	179
8.4	网络配置 (network)	179
8.4.1	函数	180
8.4.2	class WLAN	180
8.4.3	构造函数	180
8.4.4	方法	181
8.4.5	连接网络	184
8.4.6	作为 http 服务器	186
8.5	文件管理软件	189
8.5.1	WebREPL	189
8.5.2	uPyLoader	193
8.5.3	MicroPython File Uploader	200
8.5.4	uPyCraft	201

第 9 章	使用技巧和常见问题	206
9.1	不能正确识别出 PYBFlash 磁盘	206
9.2	安装虚拟串口失败	206
9.3	PYBFLASH 磁盘中文件损坏或乱码	207
9.4	恢复出厂设置	207
9.5	怎样升级 pyboard 的固件	208
9.5.1	Linux 下升级固件	210
9.5.2	Nucleo 开发板升级固件	211
9.6	升级 ESP8266 的固件	212
9.6.1	需要的软件	212
9.6.2	固件文件	212
9.6.3	进入升级模式	213
9.6.4	esptool.py	213
9.6.5	Flash Download Tools	215
9.6.6	uPyLoader	216
9.7	访问寄存器	217
9.7.1	方法	217
9.7.2	使用 STM32 的寄存器控制 LED	218
第 10 章	MicroPython 应用	219
10.1	计算圆周率	219
10.2	驱动 OLED 模块	222
10.3	温度传感器 DS1820	225
10.4	温湿度传感器 DHT11	228
10.5	气压传感器 BMP180	229
10.6	用热敏电阻测温度	237
10.7	在 SensorTile 上使用 MicroPython	242

10.7.1	移植 MicroPython 固件	244
10.7.2	传感器驱动	246
10.7.3	使用 SensorTile 制作开源智能怀表项目	261
10.8	其他 MicroPython 项目链接	270
附录		272

第 1 章

MicroPython 简介

1.1 MicroPython 是什么

可能很多读者还不太清楚 MicroPython 是什么？也有一些读者听说了 MicroPython，但是还不太清楚它到底有什么用。从单词的组成看，它是由 Micro 和 Python 两个部分组成，Micro 是微小的意思，而 Python 是一种编程语言，两者合起来的字面意思就是微型的 Python。实际上 MicroPython 就是用于嵌入式系统上的 Python，可以用 MicroPython 在嵌入式系统中编程，做各种应用。

在嵌入式编程中，我们传统都是使用 C/C++、汇编等软件，而 Python 是一种脚本语言，为什么要去用 Python 编程呢？它有什么特点呢？让我们先从 MicroPython 的历史说起吧。

1.2 MicroPython 的历史

现在开源硬件中最热门的 MicroPython 是由英国剑桥大学的教授 Damien George（达米安·乔治）发明的，Damien George 是一名计算机工程师，他每天

都要使用 Python 语言工作，同时也在做一些机器人项目。有一天，他突然冒出了一个想法：能否用 Python 语言来控制单片机，进行实现对机器人的操控呢？

可能很多读者都知道，Python 是一款非常容易使用的脚本语言，它的语法简洁、使用简单、功能强大、容易扩展。而且 Python 有强大的社区支持，有非常多的库可以使用，它的网络功能和计算功能也很强，可以方便和其他语言配合使用，使用者也可以开发自己的库，因此 Python 被广泛应用于工程管理、网络编程、科学计算、人工智能、机器人、教育等许多行业，Python 语言也长期在编程语言排行榜上处于前五的位置。更重要的是 Python 也是完全开源的，不像 Windows、Java 那样受到某些大公司的控制和影响，它完全是靠社区在推动和维护，所以 Python 受到越来越多的开发者青睐。不过遗憾的是，因为受到硬件成本、运行性能、开发习惯等一些原因的影响，前些年 Python 并没有在通用嵌入式方面得到太多的应用。

随着半导体技术和制造工艺的快速发展，芯片的升级换代速度也越来越快，芯片的功能、内部的存储器容量和资源不断增加，而成本却在不断降低。特别是随着像 ST 公司和乐鑫公司高性价比的芯片和方案应用越来越多，这就给 Python 在低端嵌入式系统上的使用带来了可能。

Damien 花费了六个月的时间开发了 MicroPython。MicroPython 本身使用 GNU C 进行开发，在 ST 公司的微控制器上实现了 Python3 的基本功能，拥有完善的解析器、编译器、虚拟机和类库等。在保留了 Python 语言主要特性的基础上，他还对嵌入式系统的底层做了非常不错的封装，将常用功能都封装到库中，甚至为一些常用的传感器和硬件编写了专门的驱动。我们使用时只需要通过调用这些库和函数，就可以快速控制 LED、液晶、舵机、多种传感器、SD、UART、I2C 等，实现各种功能，而不用再去研究底层模块的使用方法。这样不但降低了开发难度，而且减少了重复开发工作，可以加快开发速度，提高了开发效率。以前需要较高水平的嵌入式工程师花费数天甚至数周才能完成的功能，现在普通的嵌入式开发者用几个小时就能实现类似的功能，而且要更加轻松和简单。

为了宣传 MicroPython，2014 年 Damien 在 Kickstarter（国外最著名的众筹网站之一）上进行了一次众筹（如图 1.1 所示），众筹的内容就是我们后面将要介绍的 pyboard (PYB V10)。PYB V10 是专门为 MicroPython 而设计，它使用

了 STM32F405RG 微控制器，开发板上内置了 4 个不同颜色的 LED 指示灯、一个三轴加速传感器、一个 microSD 插座，可以通过 USB 下载用户程序和升级固件，使用非常方便。PYB V10 在 KickStarter 上的众筹非常成功，一推出就受到全世界的工程师和爱好者的广泛关注和参与，获得很高的评价，并很快被移植到多个硬件平台上，很多爱好者用它做出各种有趣的东西。



图 1.1 pyboard 众筹

MicroPython 最早是在 STM32F4 微控制器平台上实现的，现在已经移植到 STM32L4、STM32F7、ESP8266、ESP32、CC3200、dsPIC33FJ256、MK20DX256、micro: bit、MSP432、XMC4700、RT8195 等众多硬件平台上，此外还有不少开发者在尝试将 MicroPython 移植到更多的硬件平台上，更多的开发者在使用 MicroPython 做嵌入式应用，并将它们分享在网络上。

MicroPython 并不是在单片机/微控制器上唯一尝试使用 Python 编程的，更早还有像 PyMite 这样的开源项目，但是它们都没有真正完成，而 MicroPython 首先在嵌入式系统上完整实现了 Python3 的核心功能，并可以真正用于产品开发。

除了 MicroPython，在嵌入式系统上还有像 Lua、Javascript、MMBasic 等脚本编程语言。但是它们都不如 MicroPython 的功能完善，性能也没有 MicroPython 好，在可移植性、使用的简便方面都不如 MicroPython，可以使用的资源也很少，因此影响并不是太大，只是在创客和 DIY 方面有所应用。

MicroPython 在 KickStater 上众筹的网址和说明为:

<https://www.kickstarter.com/projects/214379695/micro-python-python-for-microcontrollers/>

1.3 MicroPython 的特点

MicroPython 并没有带来一种全新的编程语言,但是它的意义却超过了一种新式的编程语言。它为嵌入式开发带来了一种新的编程方式和思维,就像以前的嵌入式工程师从汇编语言转到 C 语言开发一样。它的目的不是要取代 C 语言和传统的开发方式,而是让大家可以将重点放在应用层的开发上,嵌入式工程师可以不需要每次从最底层开始构建系统,可以直接从经过验证的硬件系统和软件架构开始设计,减少了底层硬件设计和软件调试的时间,提高了开发效率。同时它也降低了嵌入式开发的门槛,让一般的开发者也可以快速开发网络、物联网、机器人应用。

随着硬件的高速发展,传统的嵌入式开发方式逐渐显露出一些问题。现在的芯片越来越复杂,更新换代也越来越快,几乎每隔半年到一年,各硬件厂家都会推出新型号的芯片,包含了新的功能,或者提高了性能。以前的嵌入式开发以 8 位单片机为主,芯片虽然也很复杂,但是寄存器不多,用法也简单,用不长的时间就能初步掌握。而现在的 ARM 和其他 32 位、64 位芯片,寄存器非常多,使用上也非常复杂,很少有工程师还能停留在寄存器级别进行复杂的软件开发了。如果说以前的工程师通过几天就可以熟悉一种单片机、几个星期就能初步掌握它、几个月就能熟练应用开发产品,现在就很少有工程师能够跟上芯片更新的步伐了。而且现在的环境一般也不会再允许大家先去学习几个月到一两年的时间,很多时候都是边用边学。以前的工程师只要深入掌握了一两种单片机就可以在很长时间内应对大部分的应用,现在就需要使用硬件厂家提供各种函数库和辅助开发工具,才能充分利用控制器的各种新功能,如:ST 公司的 CubeMX、NXP 公司 CodeWarrior 中的 PE、Silabs 公司 Simplicity Studio 的 Hardware configurator、Microchip 公司的 MCC 等。虽然这些工具也可以带来很大的方便,但是各厂家的工具都不相同,库函数也是互不兼容,使用这些工具开发的程序很难直接移植,给我们的系统设计和维护带来许多不便。

而 MicroPython 让我们在应用层级移植程序有了可能，就像 PC 上的程序那样。以前使用 C/C++ 编程时，虽说 C/C++ 也是高级编程语言，可以方便程序移植，但是实际开发中由于 C/C++ 语言本身的复杂性、硬件平台的多样性、开发工具的依赖性等因素，造成在实际产品中程序移植变得很困难，很多时候做系统移植还不如重新开发来得简便。而 MicroPython 是脚本语言，本身和硬件的相关性就比较小，加上 Python 语言本身简洁的特性，所以程序移植也变得容易了。

在物联网时代，嵌入式系统开发的要求和以前有所变化，需要程序能够灵活多变、快速响应用户的需求。传统开发时，需要修改整个项目，然后提供新的二进制文件升级，功能测试、现场维护和远程升级都比较复杂。而 MicroPython 只需要考虑用户功能，只要升级用户程序，简单方便。

MicroPython 的特点是简单易用、移植性好、程序容易维护，但是采用 MicroPython 和其他脚本语言（如 Javascript）开发的程序，其运行效率肯定没有采用 C/C++、汇编等编译型的工具高。MicroPython 并不会取代传统的 C/C++ 语言，但是在很多情况下，硬件的性能是过剩的，降低一点运行效率并不会有太大影响，而 MicroPython 带来的开发效率整体提升，才是最大的好处。

如果说 Arduino 将一般电子爱好者、DIYer、创客带入了嵌入式领域，让他们不再畏惧硬件的开发和使用。那么 MicroPython 完全就可以作为工具去开发真正的产品，让普通工程师和爱好者可以快速开发嵌入式程序，让嵌入式开发和移植变得轻松和简单。

1.4 授权

虽然 MicroPython 的功能非常强大，用户也很多，但是 MicroPython 和 pyboard 都使用了非常宽松的 MIT 授权方式，而不是大公司常用的 GPL 授权。这意味着任何人都可以使用、修改、发行它，并可以将它免费用在商业产品中。MicroPython 的开放性让它在短短几年时间，就获得了很大发展，全世界有很多工程师和爱好者在学习并使用它，移植到了很多系统中，并分享了众多的成果。

第2章 基础知识

MicroPython 虽然很容易使用，但是对于初学者（特别是对于 Window 平台下的开发者），为了更好地学习和使用 MicroPython，避免一些因为不同平台差异带来的影响，我们需要先做一些准备工作，了解它的基础知识和使用方法、交互式命令环境、常见的问题，以及开发常用的软件等等。

2.1 Python3 和 MicroPython

因为本书不是专门讲解 Python 语言的，所以 Python 语言相关的知识大家可以通过专门的书籍和教程去学习和了解。好在 Python 语言非常简单易学，即使没有学习过 Python 语言，大部分读者也可以在几天时间就能初步掌握。这里假设读者已经对 Python 语言有了初步的了解。

读者即使没有学习过 Python 语言，掌握起来也非常快。Python 的参考书非常多，大家可以灵活选择。推荐以下两本书：

- 《笨办法学 Python（第3版）》，（美）Zed A. Shaw 著，人民邮电出版社，2014年10月。
- 《Python+Cookbook（第3版）》，（美）比斯利，琼斯著，人民邮电出版社，