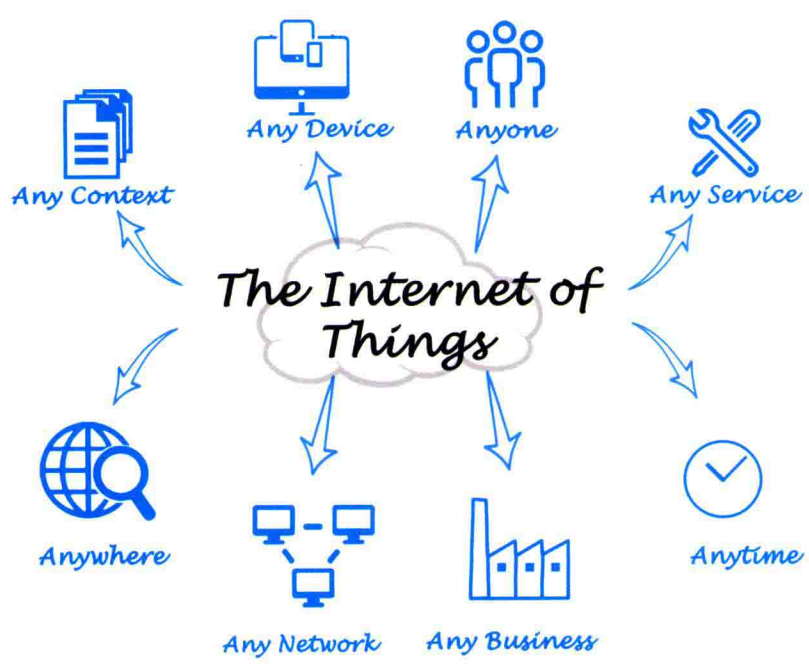




物联网Python 开发实战

安 翔 © 编著



非 外 借



扫码获取
项目源码+配套硬件

用Python实现物联网设备到云端全栈开发
通过项目实战深入物联网开发的所有细节

中国工信出版集团

电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

物联网 Python 开发实战

安翔 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

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

物联网 Python 开发实战/安翔编著. —北京: 电子工业出版社, 2018. 3
ISBN 978-7-121-33751-2

I. ①物… II. ①安… 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 036172 号

责任编辑: 富 军

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

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: 787×1092 1/16 印张: 20.25 字数: 518 千字

版 次: 2018 年 3 月第 1 版

印 次: 2018 年 3 月第 1 次印刷

印 数: 2 500 册 定价: 69.00 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 zlt@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)88254456。

前 言

本书内容写作安排：

第 1 章——物联网邂逅 Python

本章大致介绍物联网的组成、架构、发展现状及当前市面上典型的物联网应用方案，阐述 Python 这门编程语言的特性和优点，讲述用 Python 语言开发物联网终端设备、网关设备、Web 后台程序的具体方法及优势。

第 2 章——开启 Python 之旅

本章从零开始讲解如何使用 Python 语言进行编程，通过大量的实例代码带领 Python 初学者掌握基本的 Python 编程技巧。

第 3 章——Python 数据结构

Python 除了拥有比较通用的字符串、列表等数据结构外，还有元组、字典、集合等特有的数据结构。熟练掌握和运用 Python 的各种数据结构是编写高质量 Python 程序的基础。

第 4 章——Python 高级特性

Python 作为一门高级编程语言，拥有众多自身特有的高级特性，如生成器、迭代器、装饰器等。同时，Python 也可面向对象编程，只有掌握 Python 高级特性的使用方法，才能抓住 Python 编程的精髓。

第 5 章——物联网核心组件

本章介绍物联网的核心组成部分及关键技术点，首先介绍 WIFI、移动网络、Zigbee 通信、BLE、LoRa、NB-IoT 等网络通信方案，以及 HTTP、Websocket、XMPP、CoAP、MQTT 等网络通信协议；其次介绍常用的硬件设备种类，包括处理器、传感器、通信模块等；最后介绍几种市面上常用的物联网云平台，如中国移动的 OneNet 平台、AWS IoT 平台、IBM 的 Waston IoT 平台。

第 6 章——MicroPython 开发物联网终端

终端是物联网连接真实世界的第一层，是物联网产品中数量最为庞大的部件。本章重点讲解目前非常火爆的 MicroPython，可用于开发单片机程序，单片机是物联网终端设备的核心，除了介绍 MicroPython 的基本使用外，还会用 MicroPython 构建一个液晶屏显示程序实例。

第 7 章——构建物联网网关

网关是连接终端设备和后台的枢纽，同时还负责局域网的组建和维护、本地数据存储、通信规则制定、业务逻辑管理等。在实际项目中，网关通常是一个运行 Linux 操作系统并搭载物联网通信模块及其他外设的嵌入式设备。本章将介绍网关上嵌入式 Linux 的相关知识及 Python 环境的构建方法，为后续讲解网关 Python 的开发提供基本的软、硬件环境。

第 8 章——网关数据编码与处理

网关作为物联网应用中的通信枢纽，在通信过程中有多种格式的数据需要处理。其方式包括转发、编解码、解析等。本章讲述网关如何通过 Python 语言处理各种格式的数据，包括 CSV、JSON、XML、二进制数据、Base64 格式的数据。

第 9 章——网关多进程与多线程

物联网网关往往需要管理大量的终端设备，还要承担通信、运算、IO 操作等多种任务。本章介绍 Python 编程时实现多任务的方式及多进程、多线程、协程、异步 IO 等多种 Python 程序模型，并对它们做出对比，针对 CPU 密集型和 IO 密集型不同应用场景下方案的选择，使网关程序性能更优。

第 10 章——网关数据持久化

网关是运算中心与通信枢纽，也是局域网数据存储中心。数据持久化是一个系统稳定运行的必要条件。所有数据存放在内存中时，一旦设备发生软、硬件故障，将发生数据丢失。为了避免这种情况，某些关键数据需要固化到磁盘上。本章介绍两种数据固化到磁盘上的两种方式——普通文件和数据库，并介绍 Python 程序对这两种数据存储方式的实际运用。

第 11 章——Python 扩展

Python 在物联网行业的应用与其他行业的应用有很大不同：物联网涉及硬件操作且物联网设备的 CPU 处理能力通常非常有限。因此，在某些特殊场景，需要使用 C 语言编写操作硬件的程序从而提供 Python 调用接口以实现 Python 对硬件的操作，而且在某些效率要求非常高的场景，使用运行效率更高的 C 语言编写程序，再编译成 Python 扩展库，可提高程序执行效率。本章主要讲解使用 C 语言扩展 Python 的方法。

第 12 章——网关网络编程

网关负责局域网的组建和维护，同时需要与后台通信以完成数据、指令的传输。本章首先介绍使用 Python 进行基本的 Socket 编程，接着以实战项目为例，分别讲解基于 requests 模块的文件传输及基于 MQTT 协议实现的 hbmqtt 编程。

第 13 章——物联网后台 Web 开发

本章介绍基于 Python 的 Django Web 框架基础知识，包括视图与 URL 配置、模板、模型、表单、静态文件处理、用户注册与登录，主要介绍 Django 框架 Web 编程的基础知识

备，以便能够理解实战项目中的 Web 程序。

第 14 章——物联网 Python 项目实战

本章介绍一个基于物联网的智能种植项目，从项目架构、功能及项目所包含的终端设备、网关、后台程序等方面全面展示开发过程。该项目在 GitHub 提供了完整的源码，读者通过该实战项目能够了解一个完整物联网项目的全貌，掌握物联网 Python 开发的细节。

目 录

第 1 章 物联网邂逅 Python	1
1.1 物联网组成架构	1
1.2 物联网发展现状	3
1.2.1 终端设备	3
1.2.2 物联网操作系统	4
1.2.3 通信手段	4
1.2.4 网络建设	5
1.2.5 应用协议	5
1.2.6 物联网云平台	5
1.3 物联网典型应用	6
1.3.1 共享单车	6
1.3.2 智能家居	7
1.3.3 智慧农业	7
1.4 使用 Python 的理由	8
1.4.1 Python 特性	8
1.4.2 Python 无处不在	9
1.4.3 开发速度有时更重要	10
1.5 美丽的相遇	12
1.5.1 Python 与终端	12
1.5.2 Python 与网关	13
1.5.3 Python 与云平台	14
1.5.4 物联网 Python 全栈开发	14
第 2 章 开启 Python 之旅	16
2.1 Python 版本的选择	16
2.2 搭建开发环境	16
2.2.1 安装 VMware	17
2.2.2 在 VMware 上安装 Ubuntu	17
2.2.3 安装 VMware Tools 工具	18
2.2.4 虚拟机的网络配置	19
2.2.5 Samba 共享设置	22
2.2.6 修改 Python 版本	23
2.3 开始编写 Python 程序	24
2.3.1 交互式解释器	24

2.3.2	源代码执行	24
2.3.3	注释	24
2.3.4	代码块和缩进	25
2.3.5	继续和分隔	25
2.3.6	输入和输出	25
2.4	变量和数据类型	26
2.4.1	变量赋值	26
2.4.2	常量	27
2.4.3	数据类型	27
2.5	条件和循环	32
2.5.1	if 语句	32
2.5.2	while 循环	33
2.5.3	for...in 循环	34
2.5.4	跳出循环	35
2.5.5	pass 语句	36
2.6	函数	37
2.6.1	定义函数	37
2.6.2	函数调用	37
2.6.3	函数的参数	38
2.6.4	内置函数	41
2.7	变量进阶	42
2.7.1	全局变量与局部变量	42
2.7.2	global 关键字	42
2.7.3	nonlocal 关键字	42
2.7.4	变量作用域	43
2.8	模块与包	44
2.8.1	使用模块	44
2.8.2	包	45
2.9	异常处理	46
第3章	Python 数据结构	48
3.1	字符串	48
3.1.1	索引和切片	48
3.1.2	字符串运算符	49
3.1.3	字符串格式化	49
3.1.4	字符编码	50
3.2	列表	51
3.2.1	访问列表	51
3.2.2	更改列表	51
3.2.3	列表操作符	52

3.2.4 列表常用函数	53
3.3 元组	54
3.3.1 访问元组	55
3.3.2 元组操作符	55
3.3.3 元组内置函数	55
3.4 字典	56
3.4.1 访问字典	57
3.4.2 修改字典	57
3.4.3 字典键的特性	58
3.4.4 字典函数	58
3.5 集合	59
3.5.1 忽略重复元素	59
3.5.2 无序	59
3.5.3 常用操作	60
第4章 Python 高级特性	61
4.1 生成器	61
4.1.1 列表推导式	62
4.1.2 生成器表达式	62
4.1.3 生成器函数	63
4.2 迭代器	65
4.3 函数式编程	66
4.3.1 高阶函数	67
4.3.2 内置高阶函数	68
4.3.3 闭包	72
4.3.4 装饰器	73
4.3.5 匿名函数	78
4.3.6 偏函数	78
4.4 面向对象编程	79
4.4.1 类与对象	80
4.4.2 访问限制	81
4.4.3 继承	82
4.4.4 多重继承	83
4.4.5 多态	85
4.4.6 运算符重载	86
第5章 物联网核心组件	89
5.1 网络通信方案	89
5.1.1 WIFI 网络	89
5.1.2 移动网络	89
5.1.3 Zigbee	90

5.1.4	BLE	90
5.1.5	LoRa	90
5.1.6	NB-IoT	92
5.2	网络通信协议	92
5.2.1	HTTP	92
5.2.2	WebSocket	93
5.2.3	XMPP	93
5.2.4	CoAP	94
5.2.5	MQTT	95
5.3	硬件	96
5.4	物联网云平台	96
5.4.1	OneNet	96
5.4.2	AWS IoT	98
5.4.3	Waston IoT	98
第6章	MicroPython 开发物联网终端	100
6.1	MicroPython 简介	101
6.1.1	MicroPython 项目	101
6.1.2	MicroPython 特点	101
6.1.3	MicroPython 源码	102
6.2	MicroPython 开发板	103
6.2.1	官方开发板 PYBoard	103
6.2.2	国内开发板 TPYBoard	104
6.2.3	OpenMV	108
6.2.4	LoPy	109
6.3	第一个 MicroPython 程序	110
6.4	交互式解释器	112
6.5	按键中断与回调	114
6.6	定时器的使用	115
6.7	串口测试	118
6.8	SPI 接口驱动液晶屏	120
6.8.1	硬件连接	120
6.8.2	LCD 屏驱动	120
6.8.3	字库说明	122
6.8.4	主程序	123
6.9	源码编译与固件升级	124
6.9.1	准备编译环境	124
6.9.2	源码下载与编译	125
6.9.3	固件烧写	127
6.10	MicroPython 多线程	128

6.10.1	MicroPython 线程简介	128
6.10.2	MicroPython 线程使用	128
6.10.3	多线程中的锁	129
6.11	安全模式及恢复出厂设置	130
6.11.1	安全模式	131
6.11.2	恢复出厂设置	131
第7章	构建物联网网关	132
7.1	网关概述	132
7.1.1	网关的构成	132
7.1.2	常用物联网网关	133
7.2	自主构建网关	136
7.2.1	交叉编译	136
7.2.2	编译引导程序	138
7.2.3	内核配置与编译	140
7.2.4	制作文件系统	142
7.2.5	分区与下载	144
7.2.6	系统启动流程简介	146
7.2.7	移植 Python3	146
7.3	树莓派作网关	148
7.3.1	初次启动树莓派	148
7.3.2	串口登录树莓派	150
7.3.3	SSH 访问树莓派	153
7.3.4	升级 Python 版本	155
7.3.5	必备工具安装	157
7.3.6	板载 WIFI 配置	158
7.3.7	串口通信	161
第8章	网关数据编码与处理	164
8.1	读写 CSV 文件	164
8.1.1	读取 CSV 文件	164
8.1.2	写入 CSV 文件	167
8.2	JSON 解析	168
8.2.1	JSON 数据格式	168
8.2.2	编码	168
8.2.3	解码	170
8.3	XML 操作	171
8.3.1	XML 简介	171
8.3.2	解析 XML	172
8.3.3	创建与修改	174
8.4	二进制数据的读写	178

8.5	Base64 编解码	179
8.6	正则表达式	181
8.6.1	正则表达式简介	181
8.6.2	re 模块	182
8.6.3	贪婪匹配	187
第 9 章	网关多进程与多线程	188
9.1	多进程	188
9.1.1	multiprocessing 模块	189
9.1.2	进程同步	191
9.1.3	进程间通信	194
9.2	多线程	195
9.2.1	threading 模块	195
9.2.2	线程同步	196
9.2.3	线程间通信	198
9.3	多线程和多进程的思考	200
9.3.1	多核 CPU 利用率实验	200
9.3.2	GIL 全局锁	202
9.3.3	切换的开销	203
9.3.4	多线程与多进程的选择	203
9.4	异步 IO	204
9.4.1	协程	204
9.4.2	协程与多线程对比	205
9.4.3	asyncio	208
第 10 章	网关数据持久化	211
10.1	文件操作	211
10.1.1	读写文本数据	211
10.1.2	操作文件和目录	213
10.1.3	读写压缩文件	217
10.1.4	内存映射	219
10.1.5	临时文件与目录	220
10.1.6	序列化 Python 对象	221
10.2	SQLite 数据库	222
10.2.1	SQLite 简介	222
10.2.2	命令行操作 SQLite	223
10.2.3	创建数据库	224
10.2.4	创建表	225
10.2.5	SQL 语句	227

10.2.6 SQLite 的 Python 编程	230
第 11 章 Python 扩展	235
11.1 Python 扩展的原因	235
11.2 连接硬件的纽带	235
11.3 C 语言扩展 Python	236
11.3.1 简单的 C 语言 Python 扩展	236
11.3.2 传递整型参数	238
11.3.3 传递字符串参数	239
第 12 章 网关网络编程	242
12.1 网关网络通信方案	242
12.2 Socket 编程	243
12.3 requests	246
12.3.1 上传文件	246
12.3.2 下载文件	246
12.4 hbmqtt	246
12.4.1 安装	247
12.4.2 hbmqtt 命令操作	247
12.4.3 hbmqtt API 编程	250
12.5 LoRa 网络通信	252
12.5.1 LoRa 模块初始化	252
12.5.2 数据监听与接收	253
12.5.3 数据缓存	253
第 13 章 物联网后台 Web 开发	255
13.1 Django 简介	255
13.2 创建一个网站	256
13.2.1 Django 安装	256
13.2.2 创建项目	256
13.2.3 运行与访问	257
13.3 网站首页	258
13.3.1 源码文件结构	259
13.3.2 视图与 URL 配置	260
13.4 Django 模板	261
13.4.1 Django 模板系统	262
13.4.2 模板继承	263
13.5 Django 模型	266
13.5.1 安装 MySQL 数据库	266
13.5.2 创建模型	267
13.5.3 必要的配置	269
13.6 用户注册	270

13.7 账号登录	274
第 14 章 物联网 Python 项目实战	278
14.1 项目简介	278
14.2 终端设备程序开发	279
14.2.1 采集空气温湿度	281
14.2.2 获取环境光强度	286
14.2.3 雨量检测	288
14.2.4 水位检测	290
14.2.5 土壤湿度检测	291
14.2.6 水泵的控制	293
14.2.7 入侵检测	295
14.2.8 灯光控制	296
14.2.9 电量检测	297
14.2.10 LoRa 通信	297
14.2.11 JSON 消息	300
14.3 网关程序开发	301
14.3.1 终端通信	302
14.3.2 数据库管理	302
14.3.3 文件备份	303
14.3.4 服务器通信	303
14.3.5 2G 模块的使用	303
14.4 服务器端程序开发	304
14.4.1 与网关通信	305
14.4.2 环境数据可视化	305
14.4.3 滴灌控制	306
14.4.4 灯光控制	306
14.4.5 报警显示与设置	306
14.4.6 设备管理	307
14.4.7 备份文件管理	307

第1章

物联网邂逅 Python

物联网是新一代信息技术的重要组成部分，也是信息化时代的重要发展阶段。随着移动互联网的增速放缓，物联网无疑是当前发展最为火热的科技行业之一。

依靠简单的语法、丰富的库、高效的开发效率，Python 覆盖了越来越多的 IT 领域，如科学技术、服务器后端、网络爬虫、自动化运维等，成为目前上升势头非常强劲的编程语言。同时，Python 也在运行 Linux 系统的嵌入式设备上得到应用，如开源硬件树莓派上 Python 的使用不仅发挥其强大的库功能，而且通过扩展的 IO 库使得 Python 同样可以访问底层硬件，操作硬件不再是 C 语言的专利。计算能力弱、存储空间小的单片机领域也有多个可以运行的 Python 项目，包括历史悠久的 PyMite 及其衍生品 Pymbed。这其中的 MicroPython 迭代最为迅速。它拥有自己的 Python 虚拟机和解释器，能够在多款单片机上运行。

最火的编程语言结合最火的行业，当 Python 邂逅物联网，会发生怎样的故事？碰撞出怎样的火花呢？

1.1 物联网组成架构

自物联网概念诞生以来，各大芯片公司、运营商、互联网巨头均大力投身其中。经过多年的发展，2017 年，随着 NB-IoT、eSIM 等关键技术的成熟，共享单车等物联网应用爆发，真正的万物互联已经到来，物联网大大改变了人们的生活方式，正在成为下一个科技浪潮。

物联网作为一个系统网络，与其他网络一样，也有内部特有的架构。大体上来说，物联网由云、管、端三大部分组成。端，代表终端设备，负责真实世界的感知和控制，是物联网的最底层；管，即管道，是物联网的网络核心，一切数据和指令均靠管道来传输，是物联网的中间层；云，即云平台，负责真实世界数据的存储、展示、分析，是物联网的最上层，是中枢和大脑，也是连接人和物的纽带。

图 1.1 展示了物联网的组成架构。可以看出，终端是多种软、硬件的集合，是一个带有感知、控制、通信能力的智能硬件。

具体包含如下部分：

处理器。处理器是物联网终端设备的中枢，所有的外围设备均需要连接到处理器的 IO 上，数据采集、指令下发、数据传输等功能全部由处理器控制。根据应用场景复杂

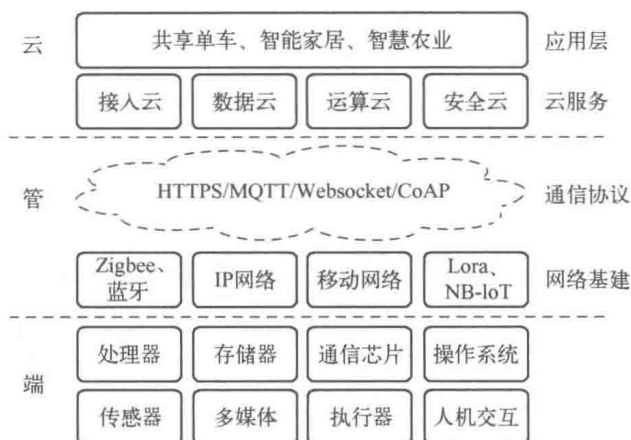


图 1.1 物联网组成架构

程度的不同，处理器可以是一个 8 位的单片机，也可以是含有多核的运算能力非常强劲的 CPU。

存储器。以单片机为核心的终端设备，存储器并不是必需的，因为单片机本身就是含有运算器、RAM、ROM 的集合，在 ROM 无法满足存储空间需求的情况下才需要外接存储器，而对于需要运行 Linux 等操作系统的终端，存储器是必不可少的部分，可用于存放操作系统的镜像、根文件系统、系统配置文件及用户数据。存储器的种类有 NAND Flash、NOR Flash、EEPROM、SD 卡、TF 卡等，具体的选择需要根据项目需求而定。

传感器。传感器是将真实世界的信息转化为计算机信号的转换器，是计算机能够感知世界的关键。传感器种类繁多，包含红外传感器、温湿度传感器、光照强度传感器、气体传感器、烟雾传感器等。

执行器。执行器是控制电路、机械的器件，如继电器、电机等，计算机通过执行器控制真实世界的物体。

多媒体。现实世界是丰富多彩的，除了具体的传感器数字之外，还可以是音频、图片、视频等，这些文件的采集就需要多媒体设备的支持，如声卡、摄像头等。

通信芯片。物联网终端上传数据、接收控制指令都需要与后台通信，通信芯片是提供该能力的基础。

操作系统。终端设备的低功耗、物联网特有的网络通信方式等特性决定了传统的 PC、移动操作系统无法有效满足物联网的需求，物联网需要针对物联网自身特性而设计的专用操作系统来提升开发和运行效率。

人机交互。终端设备有时需要与人直接交互，此时就需要屏幕等显示设备及按键、触摸屏、语音输入等输入设备的支持。

管道是物联网的通信基础，网络硬件设备提供真正的通信能力，网络基础建设是物联网管道的硬能力。除此之外，具体通信程序的实现需要合适的协议及规则，这些虚拟的软能力同样重要，可保证不同应用场景的通信效率和安全。物联网管道具体包含

如下内容：

网络基建。物联网设备的分散、应用场景的复杂性决定了一种单一能力的网络无法满足所有需求。它需要多种网络类型从功耗、时延、带宽、网络容量、覆盖面、稳定性等多方面来支持物联网的不同应用场景。

WIFI、有线网络具有速率和稳定性方面的优势，但覆盖率不高，在某些场景下，设备无法通过它接入网络；移动网络相比 WIFI、有线网络在覆盖率上有大幅提升，但在地下、偏远地区等特殊区域依然无法保证 100% 的覆盖。为了弥补覆盖问题，Zigbee、蓝牙 Mesh 以组网的方式构建局域网，通过网关的中继功能接入互联网，但通信距离、终端功耗等方面的能力并不是最优；Lora、NB-IoT 等低功耗广域网通信协议能够满足物联网设备对低功耗、通信距离长的要求，但通信速率很低，无法满足大数据量的传输。

可见，多种物联网通信方式互有所长，它们相互协作、弥补，共同构成物联网的网络基础建设，以满足物联网项目的不同应用场景。

通信协议。除了采用多种通信方式提供链路层、传输层的物联网网络之外，在应用层还需要不同的协议类型来满足不同的应用场景，以提高通信效率和保障通信安全。

HTTPS 是 HTTP 的安全版，基于 SSL，用于安全的 HTTP 数据传输；Websocket 替代 HTTP 轮询，可提供更加高效的全双工通信方式；MQTT 专为物联网设计，使用发布/订阅消息模式，解除应用程序耦合，对负载内容屏蔽消息传输，开销小、协议交换最小化，提供客户端异常中断机制，具有三种消息发布服务质量等级；COAP 是一种基于 REST 架构的协议，为资源受限的物联网设备而定制。

云平台是物联网的大脑和中心，大致可分为服务层和应用层。服务层提供基础、共有的服务，负责维护物联网终端设备的接入；存储和分析海量的传感器数据；提供物联网通信的安全保障；通过智能、快速地运算加速系统的运行效率。在服务层提供的基础功能之上，应用层可实现丰富多彩、五花八门的具体业务，如共享单车、智能家居系统、智慧农业系统等。

1.2 物联网发展现状

从上一节可以看出，物联网是一个非常复杂、庞大的体系，需要多方面共同构建实现，如需要半导体厂商提供处理、存储等芯片，需要运营商建设物联网网络，需要互联网企业提供后台服务，还需要集成商的整合。实际上，国内外多个科技巨头都加入了这场科技盛宴，不遗余力地促进物联网的发展，具体表现在如下方面。

1.2.1 终端设备

随着手机、平板电脑等移动设备量增速的放缓，ARM、NXP、TI、高通等半导体厂商均将发力点转移到物联网领域，打造针对物联网的专业芯片。

中国移动推出全球首款 eSIM 2G 基带芯片，更便于提供物联网海量设备的开卡服务。此