



[瑞典]皮特·瓦厄 (Peter Waher) 著 黄峰达 王小兵 译

物联网实战指南

Learning Internet of Things

实战角度介绍现有协议、通信模式、构架和物联网安全
通过吸引人、富有启发性的Raspberry Pi实例，
去了解和探索物联网



机械工业出版社
CHINA MACHINE PRESS

物联网实战指南

[瑞典] 皮特·瓦厄 (Peter Waher) 著

黄峰达 王小兵 译



机械工业出版社

本书从探讨流行的 HTTP、UPnP、CoAP、MQTT 和 XMPP 等物联网协议开始, 并从实战角度介绍了现有的协议、通信模式、构架以及物联网安全的重要性。

本书适合那些对物联网感兴趣的开发者和工程师阅读。那些对电子学、树莓派 (Raspberry Pi) 等有基本的了解 (高中水平) 以及有一些代码托管的编程经验的人, 通过本书将会很快学到当前先进的物联网解决方案。

Peter Waher: Learning Internet of Things (ISBN: 978-1-78355-353-2).

Copyright © 2015 Packt Publishing. First published in the English language under the title “Learning Internet of Things”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2016 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记 图字: 01-2015-3078 号。

图书在版编目(CIP)数据

物联网实战指南/(瑞典)皮特·瓦厄(Peter Waher)著;黄峰达,王小兵译. —北京:机械工业出版社,2016.7

书名原文:Learning Internet of Things

ISBN 978-7-111-54307-7

I. ①物… II. ①皮…②黄…③王… III. ①互联网络—应用—指南
②智能技术—应用—指南 IV. ①TP393.4-62②TP18-62

中国版本图书馆 CIP 数据核字 (2016) 第 164135 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:林 桢 责任编辑:林 桢

责任校对:王琳琳 封面设计:赵颖喆

责任印制:常天培

北京圣夫亚美印刷有限公司印刷

2016 年 9 月第 1 版第 1 次印刷

169mm×239mm·11.75 印张·208 千字

0001—3600 册

标准书号:ISBN 978-7-111-54307-7

定价:59.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:010-88361066 机工官网:www.cmpbook.com

读者购书热线:010-68326294 机工微博:weibo.com/cmp1952

010-88379203 金书网:www.golden-book.com

封面无防伪标均为盗版

教育服务网:www.cmpedu.com

作者简介

Peter Waher 是 Clayster 公司的创始人之一，这家公司从斯堪的纳维亚半岛发展到了四大洲。Clayster 公司专注于开发物联网应用，并为快速物联网应用开发提供了一个平台。现在，Peter 是 Clayster 公司智利子公司 Clayster Laboratorios Chile S. A. 的 CEO，他们向合作伙伴公司提供开发相关的专业知识，并促进研究机构提高物联网技术。其已经在计算机和通信设备领域工作了 20 年，拥有包括了从使用汇编语言进行开发的资源受限设备到高级系统设计、架构的经验。当前，其正努力参与各种物联网的标准化工作，包含 IEEE、UPnP 和 XSF。其开发的物联网智能 IPTV 应用赢得了全球智慧生活联盟奖（Living Labs Global Award）。Peter 的 Linkedin 是 <http://linkedin.com/in/peterwaher/>。

技术审阅者简介

Fiore Basile 是一个程序员、系统管理员、创新者、企业家和创客。从 1996 年开始，其在一些意大利以及欧洲范围内的工业项目和研究项目担任项目经理、咨询师、技术人员。现在其正在研究可穿戴技术、更有效率的计算和智能连接设备。

Dominique “Dom” Guinard 是 EVERYTHNG 公司的 CTO 兼联合创始人，Dom 拥有苏黎世联邦理工学院的博士学位，在校期间其主要研究定义物联网构架、互联对象的全球性网络（传感器网络、设备、机器和标记物）。其还是 Web of Things. org 的联合创始人和物联网系列会议的创始人。2011 年，Dom 在世界 100 大物联网思想家中立列第五。2012 年年初，其物联网博士研究被授予苏黎世联邦理工学院奖章。

黄峰达（Phodal Huang）目前是 ThoughtWorks 公司的一名软件工程师。其是最小物联网系统（<https://github.com/phodal/iot>）项目的创建者，同时也是电子书《一步步设计物联网》（<http://designiot.phodal.com>）的作者。其喜欢设计、画画、写作、旅行以及 Hacking，在其个人网站（<http://www.phodal.com>）可看到更详细的信息。

Joachim Lindborg 是一名资深的系统工程师，在这个领域其对于所有技术都有着相当多的经验，目前是一名关注持续创新的 CTO，主要工作内容是对于分布式物联网的不断思考和通过高级数据分析实现能源效率及可持续发展的社会。

Ilesh Patel 拥有通信工程学士学位及超大规模集成电路（VLSI）和嵌入式系统设计的硕士学位。其具有很好的调试技巧，掌握高级 C/C++ 编程、脚本语言 Python 以及 VHDL。还有开发 Python 自动化测试框架、基于 FPGA 系统设计开发以及微控制器等的知识和实践经验。

序 言

物联网是当前最热门的技术术语之一。许多大公司认为它在接下来几年会有数以万亿的市场价值，因此开始投资数十亿美元研究与开发物联网。在此之上，有数以万计的发布计划在同一时期与数十亿的设备相连。

此外，似乎没有人就物联网实际是什么达成一致意见。他们肯定的是不管物联网是什么，它都很值钱。物联网能赚很多钱，但也会存在很大的竞争，这意味着许多的困惑。为了做行业的领头羊，一些公司突出其卓越的知识，发明新的流行语。在这些获取读者注意力的争夺战中，我们在全世界看到了各种各样的定义，一个更比一个好，例如，“万物互联（Internet of Everything, IoE）”“物体的网络（Web of Things）”“人与物体的互联网（Internet of People and Things）”等。无独有偶，在一些易重叠和混淆的术语中也有提及，如“大数据”“机器对机器（M2M）”“信息物理系统（CPS）”等。

物联网实际是什么，物联网究竟意味着什么，这些都没有形成一种共识，这也使得写一本关于物联网这个主题的书变得有些困难。并非在技术方面存在什么困难，而是你需要去界定你即将介绍物联网的哪些方面，而哪些方面是不需要讲的。在某种程度上，你需要以一种简单的、有效的、建设性的，同时也应尽量减少争议的方式去定义物联网（IoT）。

物联网定义

为了能够定义它，让我们首先看看这个词是如何被创造的。Kevin Ashton 曾指出，因特网中的大多数数据最初进入系统或者被系统捕获都是人为控制的。从系统的角度来看，人的作用无非是相当于对数据的质量和数量有所限制的一种缓慢的、易出错的、效率低的路由器，有时能够解释数据或者改正数据。相反，如果这些系统能够连接传感器并直接监测现实世界的事件或者性能，那将会很高效。因此，系统避开人这个中间层直接连接传感器，再将传感器连接到互联网来捕获真实世界的数据。

这一定义的问题在于，它不仅是一种定义还是一种愿景，尽管这是很重要的一点。如果系统可以访问直接由传感器捕获的数据，当然，这些数据将会更加丰富、准确。传感器网络几十年前就已经被知晓并且有其专门的研究领域。这两者之间的真正区别在哪儿？在海量数据的高效存储处理方面，物联网和大

数据的区别在哪儿？在讨论物体之间的通信方面，物联网为何不同于机器对机器（M2M）或者设备对设备（D2D）？或者说，考虑到它和系统通过传感器和执行器与真实世界交互，与信息物理系统（CPS）的区别在哪儿？物联网和刚才提到的研究领域的真正差别在哪儿？

因此，我们有一个非常简单的定义，那就是：



物联网就是当我们连接物体时所获得的东西，它由互联网控制，而非人。

互相竞争的定义

物联网不同于传感器网络，物体不需要传感器，传感器网络也不需要连接到互联网。物联网也不同于大数据，因为物体不需要捕获或产生数据，也不需要在大数据库存储云端集中存储数据。物联网不属于 M2M，因为在互联网上意味着人类可以（并且想要）直接访问这些物体。此外，后者以及 CPS 还关系到非互联网协议，在封闭和控制环境下的网络中的机器和/或设备之间的消息传输以及自动化。连接到互联网远不止简单地连接和信息传输。互联网是开放的，意味着任何人都可以向里面添加物体。这也意味着他们想要物体以一个松散的耦合方式进行互操作。互联网不仅是开放的，也是世界上最大的网络，当然它也是世界上最污秽的地方。物体连接到互联网，你可以知道的是，如果可以的话，有人会只为了乐趣就试图利用它甚至摧毁它。将物联网与 M2M 通信就好像假设在一个受控的实验室环境中你让一帮 3 岁的俏皮孩子拿着高咖啡因含量的饮料和锤子，并许诺他们砸坏一切所有看得见的东西就将得到冰淇淋奖赏，而你的实验还将继续下去。

然而有些人担心物联网是有一些局限的，于是发明了新的术语，包括人和物体的互联网，这已经包含在我们刚才看到的定义里，我们注意到，当我们连接物体时，人们已经通过计算机连接到互联网。这样的定义其实是没有必要的。其他人讨论的物体的网络（WoT）是物联网（IoT）的一个子集，其通信仅限于网页技术例如 HTTP、浏览器、脚本等。这种看法可能源于万维网（WWW）就等于互联网，接入互联网需要通过浏览器和 URL。尽管我们将在本书中讨论 Web 技术，但是我们仍然认为 Web 技术有太多限制条件。

有一些更像商业流行用语的误导性的定义，而不是技术术语，例如万物互联，令人产生一种比物联网更宏大的想法。但是，万物互联所包含的是不是物联网已经包含了的？所有相连的物体已经包含在物联网里。物体不能直接连接（空气或者水）或者间接连接（真空或者快乐），也不能访问，尽管这个名字的字面意思就是这样。万物需要一个物体或一个人连接到互联网。有一种说法称

万物互联包含过程，但是这样将与我们刚才看到的定义不符，这样的过程只是简单的推论，不需要新的定义。

直接的结果

现在我们已经对物联网有了一个明确的定义，我们将物体接入互联网获得一些东西，非人为控制。现在我们开始我们的研究，定义包含四个重要的组成部分：

- 连接，其中涉及通信协议的研究；
- 物体，涉及传感器、执行器和控制器的研究，其他物体除外；
- 非人为操作相关的配置；
- 互联网相关的安全性，如标识、认证和授权以及互操作。

本书将同一时间使用简单实用的例程介绍这些概念，以及怎样在 Raspberry Pi 平台说明并实现这些关键概念。

本书涵盖的内容

第 1 章，物联网项目准备，介绍我们即将用到的项目，如基本项目架构、开发环境，怎样准备我们的 Raspberry Pi，以及如何基于其实现基本输入输出操作，并将贯穿整本书。

第 2 章，HTTP，说明基本的 HTTP 以及怎样应用于物联网，同时也描述怎样与请求/响应和事件订阅通信模式相关。

第 3 章，UPnP 协议，说明 UPnP 协议基础知识，怎样使用 UPnP 协议用于发现特定局域网设备，并且还讨论了如何调用设备上的服务和订阅活动。另外，UPnP 协议也描述了如何构建发布易发现的服务和事件设备。

第 4 章，CoAP，说明 CoAP 基础知识，怎样应用于有限通信带宽设备上，并且会告诉你如何发布内容、如何订阅事件、如何使用块传输大容量项目以及如何发现设备上的现有资源。

第 5 章，MQTT 协议，介绍 MQTT 协议和展示物联网应用怎样使用发布/订阅通信模式和消息代理绕过防火墙。

第 6 章，XMPP，介绍 XMPP 以及如何使用联合组消息代理提供全球身份和如何为物联网开发者提供一系列丰富的通信模式。

第 7 章，使用物联网服务平台，说明一些专为物联网设计的促进物联网服务的快速应用程序开发服务平台的原因，同时关注一些反复出现的问题，例如安全性、互操作性、可扩展性、管理和监视等。

第 8 章，创建协议网关，向你展示如何使用好的抽象模型去方便协议桥梁的创建，允许系统之间的互联和基于不同技术的服务，这将使你为基于物联网

的智能城市设计一个安全和互操作的基础设施。

第9章，安全和互操作性，概述现有的威胁和常见的攻击模式以及怎样建立反制措施去保护你的物联网解决方案。它也显示了互操作性在物联网中的重要性和怎样避免限制一个有利于其他方面的方案。

附录A，控制台应用程序，展示控制台应用程序的基本结构，这个示例适用于整本书。

附录B，采样和历史，展示如何采样本书中出现的传感器项目的传感器数值和保存历史记录。

附录C，对象数据库，展示如何只通过类定义使数据保存在对象数据库中。

附录D，控制，展示本书的项目中是如何实现对一个执行器进行控制操作的。

附录E，HTTP基础，提供基本的HTTP概述。

附录F，传感器数据查询参数，提供我们可以调用的一组查询参数，以限制从设备到有用数据的读出请求。

附录G，HTTP安全，讨论使用HTTP的多种方式实现应用安全。

附录H，HTTP响应延迟，说明如何修改HTTP使用的请求/响应通信模式去模仿事件订阅通信模式。

附录I，UPnP协议基础，提供UPnP协议基本概述。

附录J，UPnP数据类型，列举常用的UPnP数据类型。

附录K，网络摄像头接口，展示一个简单地使用摄像头拍摄并发布照片的接口。

附录L，网页中的文本编码，讨论网页中的文本编码以及可能的冲突。

附录M，使用邮件发送快照，展示如何发送包含使用摄像头拍摄的快照的邮件。

附录N，追踪摄像头，展示控制应用怎样追踪网络中的摄像头。

附录O，证书和验证，简述证书如何工作，如何在树莓派上安装。

附录P，聊天界面，展示如何为你的设备增加一个聊天接口，让使用基于XMPP的标准聊天应用聊天成为可能。

附录Q，QR码，展示使用一种简单的方式产生和显示QR码。

附录R，材料清单，包含本书中示例使用到的开发板元件和材料清单。

本书附录未提及的东西可以在以下链接下载：https://www.packtpub.com/sites/default/files/downloads/3494_3532OT_Appendices.pdf

你需要为本书准备什么

除了一台安装有Windows、Linux或者Mac OS系统的计算机，还包括四块或

者五块信用卡大小的 Raspberry Pi B 型计算机，当然这些 Raspberry Pi 需要 SD 卡并安装好 Raspbian 操作系统。

本书使用到的软件免费，可以从以下网站获取：

- ❑ 基于 C# 的开发环境，可以是 Xamarin、MonoDevelop 或者 VisualStudio。Xamarin、MonoDevelop 免费可用，VisualStudio 有免费试用版。
 - Xamarin，在 Windows 和 Linux 平台可以免费使用，可以从这里下载：<http://xamarin.com/download>。
 - MonoDevelop，只有 Linux 系统（Debian，Ubuntu，Fedora，Red Hat，openSUSE）可用，可以从这里下载：<http://www.monodevelop.com/download/>。
 - VisualStudio Windows 系统免费试用版可以从这里下载：<http://www.visualstudio.com/downloads/download-visual-studio-vs>。
- ❑ 第 4 章 CoAP，我们将使用火狐浏览器的 Copper 插件验证 CoAP 调用，可以从这里免费下载：<https://addons.mozilla.org/en-US/firefox/addon/copper-270430/>。
- ❑ 第 7 章使用物联网服务平台和第 8 章创建协议网关，我们将使用 Clayster 物联网平台调用 ClaysterSmall，无论是私人使用、测试、学校使用或者 Clayster 管理工具都可以免费获得执照，可以从以下链接获取：<http://www.clayster.com/downloads/>。
- ❑ 本书展示的源代码可在 GitHub 上下载，请参见“下载示例代码”，这将有助于了解详细信息。

本书适合人群

本书适合那些对物联网感兴趣的开发者和工程师。只需对电子学、Raspberry Pi 等有基本了解（高中水平）或有一些代码托管的编程经验，例如 C# 或者 Java 或者 C++ 这种面向对象语言，你将会很快学到当前最先进的物联网解决方案。

说明

在本书中，你将会看到很多能区分不同信息的文本风格，下面是一些风格的示例以及它们的解释。文本中的代码字、数据库表名、文件夹名、文件名、文件扩展名、路径名、虚假 URL、用户输入和推特用户定位说明如下：“例如，数字信号输出使用 DigitalOutput 类处理”。

代码块设置如下：

```
private static DigitalOutput executionLed =  
    new DigitalOutput (23, true);  
private static DigitalOutput measurementLed =  
    new DigitalOutput (24, false);  
private static DigitalOutput errorLed =  
    new DigitalOutput (25, false);  
private static DigitalOutput networkLed =  
    new DigitalOutput (18, false);
```

任何命令行输入或输出写入如下：

```
$ sudo apt-get update  
$ sudo apt-get upgrade  
$ sudo apt-get install mono-complete
```

新的术语或者关键字，例如“物联网（IoT）”将会加粗显示，你在屏幕上菜单或者对话框看到的关键字在文本中显示如下：为了使屏幕按照我们希望的方式显示，我们进入“Setting”菜单，单击“Layout”菜单项目，然后选择“No Menu 5x4”选项。



这里给出重要的注意事项。



提示和技巧则会在这里出现。

读者反馈

我们很高兴能获得读者的反馈，让我们知道你对本书的真实想法——你喜欢什么或者不喜欢什么。你的反馈至关重要，可以让我们开发出更多让你受益的内容。给我们发送一般的反馈，可以发送英文 e-mail 到邮箱：feedback@packtpub.com，然后通过你的邮箱主题说明书的标题。

如果你精通本书的某一个主题，或者对某一个主题有兴趣又或者也想写这样一本书，为本书出力，可以通过 www.packtpub.com/authors 查看作者指南。

客户支持

现在你成了 Packt 出版社的一名尊敬的用户，为使你的购买物超所值，我们也为你准备了一系列的东西。

下载示例代码

本书示例项目中所展示的代码都可以从 GitHub 免费下载，下表显示了每一章内容对应的链接：

章节	项目名称	URL
1 ~ 2	Learning-IoT-HTTP	https://github.com/Clayster/Learning-IoT-HTTP
3	Learning-IoT-UPnP	https://github.com/Clayster/Learning-IoT-UPnP
4	Learning-IoT-CoAP	https://github.com/Clayster/Learning-IoT-CoAP
5	Learning-IoT-MQTT	https://github.com/Clayster/Learning-IoT-MQTT
6	Learning-IoT-XMPP	https://github.com/Clayster/Learning-IoT-XMPP
7	Learning-IoT-IoTPlatform	https://github.com/Clayster/Learning-IoT-IoTPlatform
8	Learning-IoT-Gateway	https://github.com/Clayster/Learning-IoT-Gateway

查看或下载在线附录

你可以从你的 Packt 出版社个人账户查看或下载本书的附录 A ~ R，链接地址是 <http://www.packtpub.com/>，如果你从其他地方购买了本书，你可以访问 <https://www.packtpub.com/books/content/support>，注册之后，会收到一封邮件。注意上面提到附录下载链接地址是 https://www.packtpub.com/sites/default/files/downloads/3494_3532OT_Appendices.pdf

勘误

尽管我们竭尽所能提高内容的准确性，但错误仍在所难免。如果您在书中发现了错误，也许是文字的错误，也许是代码的错误，如果您能将错误反馈给我们，我们将不胜感激。通过这种方式，可以减少其他读者的困惑，同时也能帮助我们提高后续版本的质量。如果您发现了任何错误，请访问链接：<http://www.packtpub.com/submit-errata>，选择书名，单击 errata submission form 链接，输入您发现的错误细节并反馈给我们。一旦您发现的错误被修正，您的提交将被接受，然后错误将上传到我们的网站上或者增加到相应书名的勘误栏目的勘误列表里面。所有已经存在的勘误可以通过链接 <http://www.packtpub.com/support> 选择相应的书名来查看。

盗版

互联网里盗版的存在是所有媒体一直存在的问题。在 Packt 出版社，我们对版权和许可证的保护非常严格，如果您发现我们的作品在互联网上以任何非法形式传播或复制，请立刻告知我们地址或者网站名称以便我们可以寻求补救。

如果您发现了可疑盗版材料，请通过邮箱 copyright@packtpub.com 联系我们。非常感谢您对保护我们的作者版权所给予的帮助，使我们有能力给您奉献更有价值的内容。

答疑

如果您对本书任何方面有疑问，可以通过邮箱 questions@packtpub.com 联系我们，我们会竭尽所能去解决它。

目 录

作者简介	
技术审阅者简介	
序言	

第1章 物联网项目准备	1
1.1 创建传感器项目	2
1.1.1 准备 Raspberry Pi	2
1.1.2 Clayster 库	3
1.1.3 硬件	4
1.1.4 与硬件交互	5
1.1.5 硬件接口	6
1.1.6 传感器值内部显示	6
1.1.7 持久化数据	7
1.1.8 传感器值的外部显示	8
1.1.9 导出传感器数据	8
1.2 创建执行器	10
1.2.1 硬件	11
1.2.2 硬件接口	11
1.3 创建控制器	13
1.3.1 表示传感器值	13
1.3.2 解析传感器数据	13
1.3.3 计算控制状态	14
1.4 创建摄像头	15
1.4.1 硬件	15
1.4.2 在 Raspberry Pi 上访问 串口	17
1.4.3 硬件接口	17
1.4.4 创建持久的默认设置	17
1.4.5 添加可配置的属性	18
1.4.6 持久化设备	19

1.4.7 在当前设置下工作	19
1.4.8 初始化摄像头	20
1.5 总结	21
第2章 HTTP	22
2.1 HTTP 基础	22
2.2 让 HTTP 支持传感器	24
2.2.1 搭建基于传感器的 HTTP 服务器	25
2.2.2 在传感器上安装 HTTPS 服务器	27
2.2.3 添加根菜单	27
2.2.4 在 HTML 页面显示测量 信息	29
2.2.5 动态生成图形	31
2.2.6 创建传感器数据资源	36
2.2.7 解释读出请求	36
2.2.8 测试数据导出	37
2.2.9 用户认证	37
2.2.10 为增强网络性能添加 事件	38
2.3 添加 HTTP 支持的执行器	38
2.3.1 创建 Web 服务资源	38
2.3.2 访问单个输出	39
2.3.3 获取全部输出	40
2.3.4 访问报警输出	40
2.3.5 使用测试形式	41
2.3.6 访问 WSDL	41
2.3.7 使用 REST Web 服务接口	42
2.4 为控制器添加 HTTP 支持	42
2.4.1 订阅事件	43

2.4.2 创建控制线程	44	3.7.4 接收事件	65
2.4.3 控制执行器	45	3.7.5 执行动作	66
2.5 总结	46	3.8 总结	67
第3章 UPNP 协议	47	第4章 CoAP	68
3.1 UPNP 介绍	47	4.1 生成 HTTP 二进制文件	68
3.1.1 提供服务体系结构	47	4.1.1 寻找开发工具	69
3.1.2 设备和服务能力的 文档化	48	4.2 为传感器添加 CoAP	70
3.2 创建设备描述文档	48	4.2.1 定义第一个 CoAP 资源	70
3.2.1 选择设备类型	49	4.2.2 手动触发事件通知	71
3.2.2 更加友好	50	4.2.3 注册数据输出资源	72
3.2.3 给设备提供标识符	50	4.2.4 返回 XML	73
3.2.4 添加图标	50	4.2.5 返回 JSON	74
3.2.5 为服务添加索引	51	4.2.6 返回纯文本	74
3.2.6 提供一个用于顶层网页 展示的 URL	52	4.2.7 发现 CoAP 资源	75
3.3 创建服务描述文档	52	4.2.8 测试 CoAP 资源	75
3.3.1 添加操作	52	4.3 为执行器添加 CoAP	75
3.3.2 添加状态变量	53	4.3.1 定义简单控制资源	76
3.3.3 添加唯一设备名	53	4.3.2 在 CoAP 中解析 URL	77
3.4 提供一个 Web 接口	54	4.3.3 使用 CoAP 控制输出	78
3.5 创建 UPNP 接口	54	4.4 在控制器中使用 CoAP	79
3.5.1 注册 UPNP 资源	55	4.4.1 监测观测到的资源	79
3.5.2 替换占位符	56	4.4.2 接收通知	80
3.5.3 添加 SSDP 支持	57	4.4.3 执行控制操作	81
3.5.4 通知网络	58	4.5 总结	81
3.5.5 响应搜索	59	第5章 MQTT 协议	82
3.6 实现静止图像服务	61	5.1 发布和订阅	83
3.6.1 初始化事件触发状态量	61	5.2 给传感器添加 MQTT 支持	84
3.6.2 提供 Web 服务属性	61	5.2.1 控制线程的生命周期	85
3.6.3 添加服务属性	62	5.2.2 标记重要事件	85
3.6.4 添加动作	63	5.2.3 连接到 MQTT 服务器	86
3.7 使用摄像头	64	5.2.4 发布内容	88
3.7.1 设置 UPNP	64	5.3 给执行器添加 MQTT 支持	89
3.7.2 发现设备和服务	64	5.3.1 初始化主题内容	89
3.7.3 订阅事件	65	5.3.2 订阅主题	89
		5.3.3 接收发布内容	90

5.3.4 解码和解析内容	91	6.3.10 处理请求删除不友好 的人	111
5.4 给控制器添加 MQTT 支持	92	6.3.11 搜索配置服务器	111
5.4.1 处理传感器发来的事件	92	6.3.12 提供注册信息	113
5.4.2 解码和解析传感器值	93	6.3.13 保持连接	113
5.4.3 订阅传感器事件	93	6.3.14 友好关系协商	114
5.4.4 控制执行器	94	6.3.15 处理存在的订阅请求 ..	115
5.5 总结	96	6.3.16 持续的中断协商	116
第 6 章 XMPP	97	6.3.17 为传感器添加 XMPP 支持	116
6.1 XMPP 基础知识	97	6.3.18 添加传感器的服务接口 ..	116
6.1.1 联合全局的可扩展性	97	6.3.19 更新事件订阅	117
6.1.2 提供全局的身份标识	98	6.3.20 发布合约	117
6.1.3 授权通信	99	6.4 为执行器添加 XMPP 支持 ..	118
6.1.4 在线存在的感知	99	6.4.1 添加控制器服务接口 ..	119
6.1.5 使用 XML 通信	99	6.5 为摄像头添加 XMPP 支持 ..	120
6.1.6 通信模式	100	6.6 为控制器添加 XMPP 支持 ..	120
6.1.7 扩展 XMPP	100	6.6.1 设置传感器客户端接口 ..	120
6.1.8 连接到服务器	101	6.6.2 设置控制器客户端接口 ..	122
6.1.9 附加安全的配置	102	6.6.3 设置摄像头客户端接口 ..	123
6.2 为对象添加 XMPP 支持	102	6.6.4 通过 XMPP 获取摄像头 图像	124
6.2.1 连接到 XMPP 网络	103	6.6.5 识别同等功能	124
6.2.2 监控连接状态事件	104	6.6.6 对同类表示应答	125
6.2.3 通知你的朋友	104	6.6.7 检测规则的变化	126
6.2.4 在 XMPP 之外处理 HTTP 请求	104	6.7 串联全部	127
6.3 提供额外的安全层	105	6.8 总结	128
6.3.1 配置的基础知识	105	第 7 章 使用物联网服务平台 ..	129
6.3.2 初始化对象注册接口 ..	107	7.1 选择物联网平台	129
6.3.3 注册对象	107	7.1.1 Clayster 平台	130
6.3.4 更新公开对象	108	7.1.2 下载 Clayster 平台	130
6.3.5 声名对象	109	7.1.3 创建一个服务项目	130
6.3.6 从注册表中移除对象 ..	109	7.1.4 添加引用	131
6.3.7 否认对象	109	7.1.5 制作 Clayster 模块	131
6.3.8 初始化配置服务器 接口	110	7.1.6 执行服务	132
6.3.9 处理友好的推荐	111	7.1.7 使用包清单	132

7.1.8	从 Visual Studio 执行	133
7.1.9	配置 Clayster 系统	133
7.1.10	使用管理工具	134
7.1.11	浏览数据源	134
7.2	使用 XMPP 接入我们的设备	136
7.2.1	为我们的传感器创建类	136
7.2.2	找到最好的类	137
7.2.3	订阅传感器数据	137
7.2.4	解读传入的传感器数据	138
7.2.5	为我们的执行器创建一个类	139
7.2.6	自定义控制操作	139
7.2.7	为我们的摄像头创建类	140
7.3	创建我们的控制应用	140
7.3.1	理解渲染	140
7.3.2	定义应用程序类	141
7.3.3	初始化控制器	141
7.3.4	添加控制规则	142
7.3.5	理解应用索引	143
7.3.6	定义 brieflet	143
7.3.7	显示计量表	144
7.3.8	显示二进制信号	145
7.3.9	推送更新到客户端	146
7.3.10	完成应用	148
7.3.11	配置应用	149
7.3.12	查看 10ft 界面应用	149
7.4	总结	150

第 8 章 创建协议网关 151

8.1	理解协议桥接	151
8.2	使用抽象模型	152
8.3	Clayster 抽象模型基础知识	153
8.3.1	理解可编辑数据源	154
8.3.2	理解可编辑对象	154

8.3.3	使用通用数据源	155
8.3.4	覆盖关键属性和方法	156
8.3.5	处理设备间通信	157
8.4	理解 CoAP 网关架构	157
8.5	总结	159

第 9 章 安全和互操作性 160

9.1	理解风险	160
9.1.1	重新发明一个轮子，但是是颠倒的	160
9.1.2	了解你的邻居	161
9.2	攻击模式	162
9.2.1	拒绝服务	162
9.2.2	猜测凭据	162
9.2.3	访问存储凭据	162
9.2.4	中间人	163
9.2.5	嗅探网络通信	163
9.2.6	端口扫描和网络爬行	164
9.2.7	搜索功能和通配符	164
9.2.8	密码破解	165
9.3	实现安全的工具	165
9.3.1	虚拟专用网络	165
9.3.2	X.509 证书和加密	166
9.3.3	身份授权	166
9.3.4	用户名和密码	167
9.3.5	使用消息中间人和配置服务器	167
9.3.6	集中与分散	168
9.4	互操作性的需要	168
9.4.1	解决复杂性	168
9.4.2	降低成本	169
9.4.3	允许各种新的服务和重复使用设备	169
9.4.4	结合安全和互操作性	169
9.5	总结	170