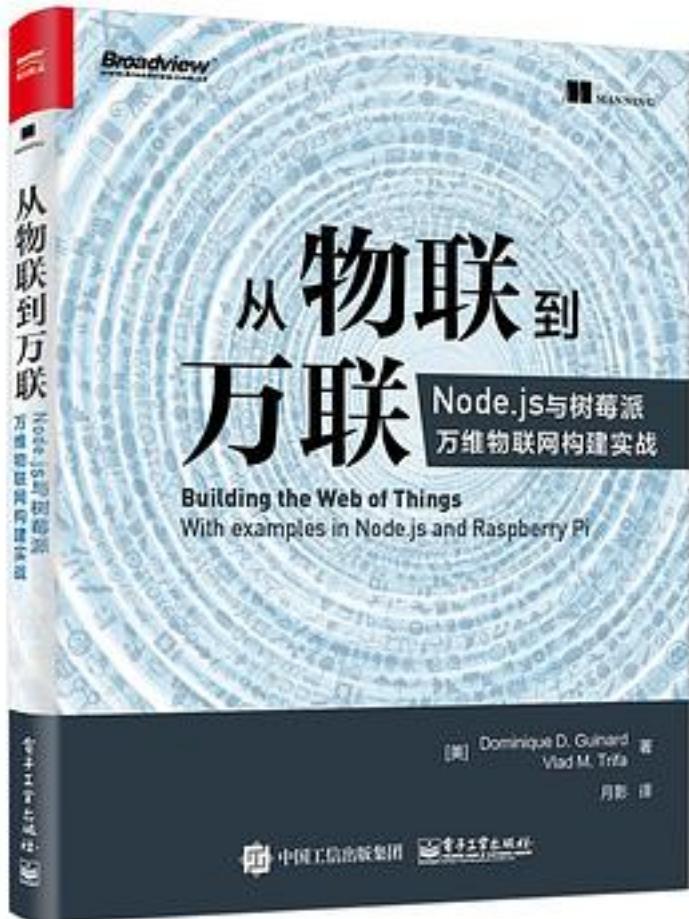


从物联到万联：Node.js与树莓派万维物联网构建 实战



[从物联到万联：Node.js与树莓派万维物联网构建实战 下载链接1](#)

著者: 【美】Dominique D. Guinard

出版者:电子工业出版社

出版时间:2018-1

装帧:

isbn:9787121327643

《从物联到万联：Node.js与树莓派万维物联网构建实战》是介绍万维物联网的入门教材。作者通过将树莓派作为物理设备网关，提出一种构建万物互联的可行方案——利用现有的万维网标准、HTTP协议，以及HTML、CSS和JavaScript技术，让智能产品终端成为开放的万维网的一部分，最终形成物联网和万维网的结合体——万维物联网。

《从物联到万联：Node.js与树莓派万维物联网构建实战》分为两个部分，第1章到第5章是第I部分，内容涵盖万维物联网基本概念、JavaScript和Node.js介绍，通过一个简单的例子来说明如何使用Node.js的Web框架与远程设备进行交互。第6章到第10章是第II部分，详细介绍万维物联网技术栈中的各层架构及如何用现有的Web技术实现各层架构，最终能够形成物理网聚合应用，能够快速创建复杂应用程序，整合各种设备和数据。

《从物联到万联：Node.js与树莓派万维物联网构建实战》涉猎的技术范围广泛，包括各种Web和物联网协议、Web标准及树莓派的原理与实践。适合之前没有丰富嵌入式开发经验，但希望探索物联网世界的Web开发人员阅读，也适合作为一本初级教程指导树莓派开发者和Node.js爱好者进行物联网开发实践。

作者介绍：

Dominique “Dom” Guinard 是 EVRYTHNG 的 CTO 和联合创始人，EVRYTHNG 是一个大规模的万维物联网云平台，通过连接产品到万维网让它们变得更智能化。Dom

拥有苏黎世联邦理工大学的博士学位，他在学校期间致力于万维物联网架构的早期研究：万维物联网架构是物联网的一个开放的应用层。早在 2012 年，他的博士论文被授予 ETH 勋章。在这之前，Dom 花了十年时间研究了许多物联网项目：与 Sun Microsystems 公司合作探索大规模 RFID

网络，研究移动电话作为物联网网关的作用，在苏黎世联邦理工大学的 Auto-ID 实验室参与诺基亚的一些研究，在麻省理工学院的 Auto-ID 实验室将 EPCglobal RFID 网络用于万维物联网，以及用四年时间在 SAP 的企业级软件中整合标记的对象和无线传感器网络。

Vlad Trifa 是 EVRYTHNG

的联合创始人、产品研发的执行副总裁。他是分布式嵌入传感器和交互式设备与使用 Web

技术的企业应用软件集成方面的公认专家。曾经他是麻省理工学院的感知城市实验室的研究员，在美国和新加坡研究城市与移动计算，在加州大学洛杉矶分校研究生生物和分布式信号处理，以及在日本东京的国际电器通信基础研究所（ATR）研究员与机器人交互和神经系统科学。他也曾经作为 SAP

的助理研究员，在工厂自动化与企业计算方面获得工业经验。Vlad

以博士学位毕业于苏黎世联邦理工大学计算机科学专业并拥有瑞士洛桑联邦理工学院机器人学、人工智能和机器学习方向的计算机科学硕士学位。

译者简介

吴亮（月影）奇虎360高级技术经理，360前端团队奇舞团

负责人，w3ctech顾问。曾先后在微软亚洲研究院做过访问学生、金蝶软件担任项目经理，百度电子商务事业部担任Web开发项目经理。多年来致力于Web标准和JavaScript技术的推广，活跃于国内各技术社区。

目录: 第I部分 物联网与万维物联网基础 1

1 从物联网到万维物联网 3

1.1 定义物联网 4

1.2 进入万维物联网	6
1.2.1 万维物联网场景：连接旅店	6
1.2.2 对比物联网和万维物联网	8
1.2.3 物联网简史	11
1.3 使用案例——为什么连接设备	13
1.3.1 无线传感器网络和分布式传感	13
1.3.2 可穿戴和自我量化	14
1.3.3 智能家居和建筑	16
1.3.4 智能城市和能源网络	16
1.3.5 智能制造与工业4.0	17
1.3.6 智能物流和供应链	18
1.3.7 营销	20
1.4 万维物联网——增强版的物联网	21
1.4.1 更容易编程	22
1.4.2 开放和可扩展标准	22
1.4.3 快速且易于部署、维护与集成	23
1.4.4 元素之间的松耦合	23
1.4.5 广泛使用的安全和隐私机制	24
1.4.6 万维物联网的弊端	25
1.5 总结	26
2 你好，万维物联网	27
2.1 初识万维物联网设备	28
2.1.1 主角登场：树莓派	29
2.2 练习1——在万维物联网上浏览一个设备	30
2.2.1 第1部分——Web 作为用户界面	30
2.2.2 第2部分——Web 作为API	34
2.2.3 小结	39
2.3 练习2——从一个WoT 传感器中轮询数据	40
2.3.1 第1部分——轮询当前传感器读数	40
2.3.2 第2部分——轮询和绘制传感器读数	41
2.3.3 第3部分——实时更新数据	42
2.3.4 小结	44
2.4 练习3——作用于现实世界	44
2.4.1 第1部分——使用一个表单来更新显示的文字	44
2.4.2 第2部分——创建你自己的表单来控制设备	46
2.4.3 小结	48
2.5 练习4——让别人知道你的设备存在	49
2.5.1 小结	52
2.6 练习5——创建你的第一个物理网聚合应用	53
2.6.1 小结	55
2.7 总结	55
3 Node.js 与万维物联网	.57
3.1 JavaScript 崛起：从客户端到服务端到智能产品	58
3.1.1 在智能产品中引入JavaScript	59
3.2 Node.js 简介	60
3.2.1 在电脑上安装Node.js	61
3.2.2 用Node.js 写一个Web 服务器	61
3.2.3 以JSON 格式返回传感器数据	63
3.3 Node.js 的模块化	64
3.3.1 npm——Node 的包管理器	64
3.3.2 通过package.json 和npm 完全管理依赖	65
3.3.3 你的第一个Node 模块	67
3.4 理解Node.js 事件循环	68
3.4.1 多线程Web 服务器	68

3.4.2 单线程、非阻塞Web 服务器	69
3.5 开始异步编程	71
3.5.1 匿名回调	72
3.5.2 具名回调	75
3.5.3 控制流库	76
3.6 总结和课外阅读	79
4 嵌入式系统入门	81
4.1 进入嵌入式设备的世界	82
4.1.1 业余爱好者的设备与工业级设备	82
4.1.2 实时操作系统与Linux	83
4.1.3 小结及超越树莓派	85
4.2 建立你的第一台WoT 设备——树莓派	86
4.2.1 入手树莓派	86
4.2.2 选择你的树莓派	88
4.2.3 购物清单	88
4.2.4 设置你的树莓派	89
4.2.5 连接你的设备	94
4.3 安装Node.js 到树莓派	95
4.3.1 在树莓派上使用Git 和GitHub	97
4.3.2 小结	97
4.4 连接传感器和执行器到树莓派	98
4.4.1 理解GPIO 端口	98
4.4.2 使用面包板连接电子元件	98
4.4.3 用Node.js 访问GPIO	100
4.4.4 课外阅读	105
4.5 总结	106
5 构建智能产品网络	107
5.1 连接智能产品	109
5.1.1 网络拓扑结构	109
5.1.2 网络分层模型	111
5.2 智能产品的网络层协议	112
5.2.1 从空间的角度考虑	113
5.2.2 网络协议和IoT	113
5.2.3 IoT 个人局域网	118
5.2.4 IoT 广域网	122
5.2.5 网络层协议的选择	124
5.3 智能产品的应用层协议	127
5.3.1 ZigBee 和蓝牙的应用层	128
5.3.2 Apple HomeKit 和Google Weave	129
5.3.3 消息队列遥测传输	130
5.3.4 资源受限的应用协议	133
5.3.5 应用层协议的选择	134
5.4 万维物联网架构	134
5.4.1 第1 层：接入层	135
5.4.2 第2 层：发现层	136
5.4.3 第3 层：共享层	136
5.4.4 第4 层：整合层	136
5.4.5 为什么WoT 是重要的	137
5.4.6 课外阅读	137
5.5 总结	138
第II部分 构建万维物联网	139
6 接入层：智能产品的API	141
6.1 设备、资源和Web 智能产品	142
6.1.1 表达性状态转移	142

6.1.2 为什么需要一致的接口	144
6.1.3 原则1：可访问资源	145
6.1.4 原则2：通过表述来操作资源	149
6.1.5 原则3：自描述信息	151
6.1.6 原则4：超媒体作为应用程序状态引擎	157
6.1.7 小结——Web 智能产品设计过程	160
6.2 超越REST：实时万维物联网	161
6.2.1 WoT 需要事件	161
6.2.2 发布/订阅	162
6.2.3 webhook——HTTP 回调	163
6.2.4 Comet——用hack 的方式让HTTP 支持实时Web	165
6.2.5 WebSocket	166
6.2.6 未来：从HTTP/1.1 到HTTP/2	170
6.3 总结	171
7 实现 Web 智能产品	173
7.1 连接设备到Web	174
7.2 直接集成模式——在设备上实现 REST	175
7.2.1 创建WoT 服务器	175
7.2.2 资源设计	177
7.2.3 表述设计	184
7.2.4 接口设计	187
7.2.5 通过WebSocket 实现pub/sub 接口	189
7.2.6 小结——直接集成模式	192
7.3 网关集成模式——CoAP	193
7.3.1 运行一个CoAP 服务器	194
7.3.2 通过网关代理CoAP	195
7.3.3 小结——网关集成模式	197
7.4 云端集成模式——EVRYTHNG 的MQTT	198
7.4.1 设置EVRYTHNG 账号	200
7.4.2 创建MQTT 客户端应用程序	204
7.4.3 使用action 来控制智能插座	206
7.4.4 创建一个简单的Web 控制应用	208
7.4.5 小结——云端集成模式	211
7.5 总结	212
8 发现层：描述和发现Web 智能产品	213
8.1 可发现性问题	214
8.2 发现智能产品	216
8.2.1 网络发现 (Network discovery)	216
8.2.2 Web 上的资源发现	219
8.3 描述Web 智能产品	222
8.3.1 Web 智能产品模型简介	224
8.3.2 元数据	226
8.3.3 属性	226
8.3.4 行为	228
8.3.5 智能产品	230
8.3.6 在树莓派上实现Web 智能产品模型	231
8.3.7 小结——Web 智能产品模型	237
8.4 语义化的Web 智能产品	238
8.4.1 关联数据和RDFa	238
8.4.2 约定的语义：schema.org	242
8.4.3 JSON-LD	243
8.4.4 课外阅读	245
8.5 总结	246
9 共享层：安全地分享Web 智能产品	247

9.1 保障智能产品安全性	248
9.1.1 加密基础	250
9.1.2 Web 安全与TLS：HTTPS 的S	251
9.1.3 在树莓派中启用TLS 以支持HTTPS 和WSS	253
9.2 授权和访问控制	258
9.2.1 通过 REST 和 API token 进行访问控制	258
9.2.2 OAuth：一个Web 身份验证框架	261
9.3 社交化万维物联网	264
9.3.1 社交化万维物联网认证代理	264
9.3.2 实现社交化万维物联网认证代理	267
9.4 课外阅读	274
9.5 总结	276
10 整合层：物理网聚合应用	277
10.1 构建一个简单的App——自动生成UI	278
10.1.1 Web 智能产品的通用用户界面	279
10.2 物理网聚合应用	286
10.2.1 使用Node-RED 工具创建物理网聚合应用	287
10.3 使用向导式工具来创建物理网聚合应用：IFTTT	293
10.3.1 将入侵者警报推送到 Google 电子表格上	294
10.3.2 使用自建通道发送请求	296
10.3.3 将入侵警报信息从Twitter 推送到Google 电子表格	297
10.4 课外阅读	299
10.4.1 从简单的聚合应用到大数据聚合应用	299
10.4.2 更好的用户体验	300
10.5 总结	300
附录A Arduino、BeagleBone、Intel Edison 与WoT	301
· · · · · (收起)	

[从物联到万联：Node.js与树莓派万维物联网构建实战 下载链接1](#)

标签

物联网

node

计算机

编程

树莓派

工程

lib

WoT

评论

消灭0评，这本书系统地讲了一遍WOT相关技术及实践。感觉WOT只不过是IOT的变种，主要优势是可以更充分地利用已有的web技术。另外，想抽点时间学下JavaScript了。

很有启发

[从物联到万联：Node.js与树莓派万维物联网构建实战 下载链接1](#)

书评

[从物联到万联：Node.js与树莓派万维物联网构建实战 下载链接1](#)