

完美图解物联网技术系列

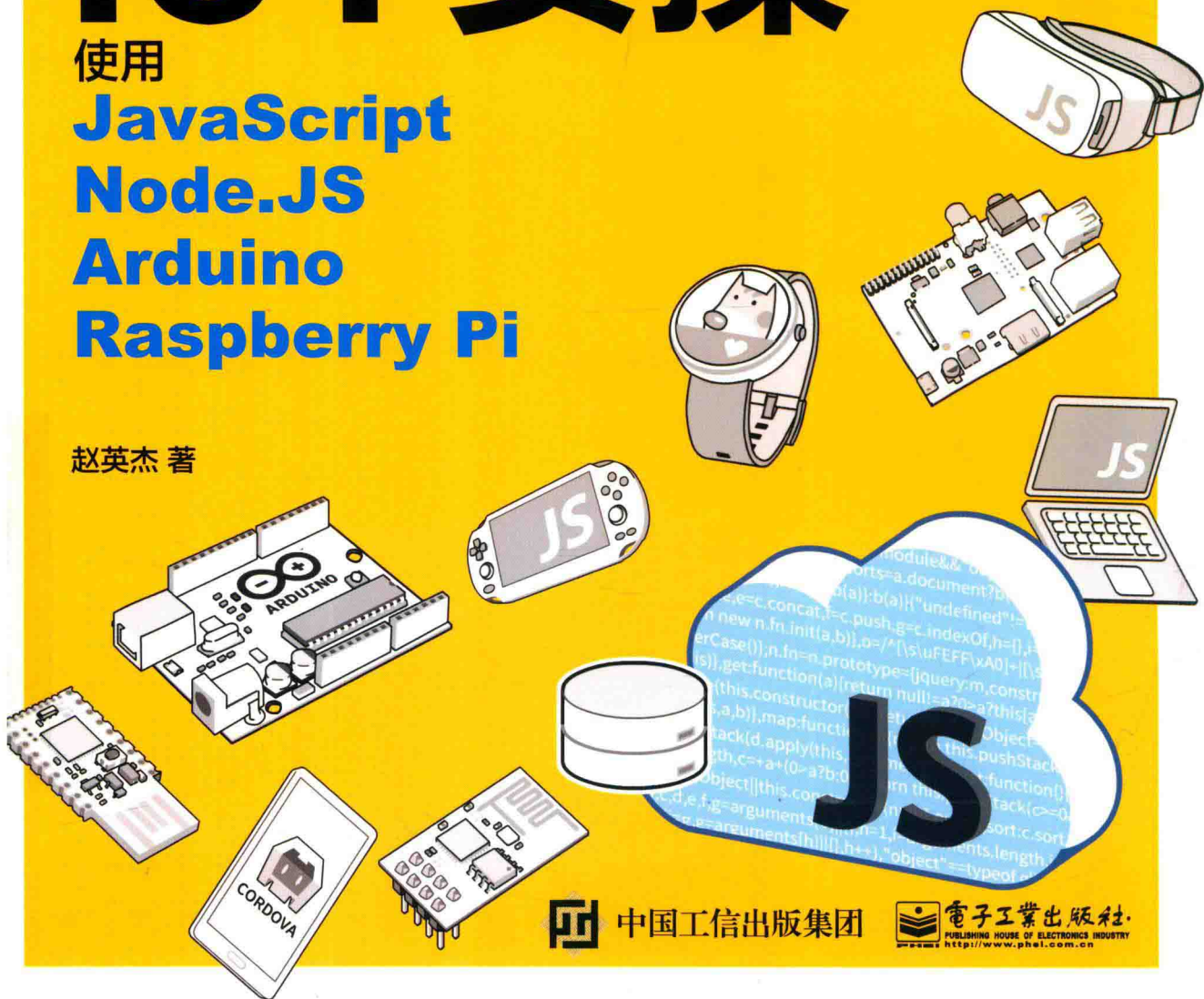
图文并茂，生动有趣，让软件工程师快速地进入物联网领域！

Broadview®
www.broadview.com.cn

完美图解物联网 IoT 实操

使用
JavaScript
Node.JS
Arduino
Raspberry Pi

赵英杰 著



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

完美图解物联网 IoT实操

使用

JavaScript

Node.JS

Arduino

Raspberry Pi

赵英杰 著

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

《完美图解物联网技术》系列图书是当前物联网开发技术应用的集大成者。本书内容强调串联网络软件和微电脑控制板,以 JavaScript 为主轴,开发网络应用程序、手机 App、互动网页、数据库程序和操控微电脑。书中包含网页式远程遥控 LED 灯、即时温湿度网站、自动感测拍照搜证、网页式远程 LED 绘图板、即时影像监控网站、影像辨识摇头控制摄像机云台、云端温湿度监控、即时信息动态图表等多元化范例。

希望本书能帮助读者了解物联网程序设计,进而开发出自己的云端物联网应用。

本书中文繁体字版《超图解物联网 IoT 实作入门:使用 JavaScript/Node.JS/Arduino/Raspberry Pi/ESP8266/Espruino》由台湾网昱多媒体出版社出版,著作权由赵英杰所有。

本书中文简体字版由赵英杰授权电子工业出版社,仅限于中国大陆地区出版发行。未经本书原出版者、原作者与本书出版者书面许可,任何单位和个人均不得以任何形式(包括任何数据库或存取系统)复制、传播、抄袭或节录本书全部或部分内容。

版权贸易合同登记号 图字:01-2016-9463

图书在版编目(CIP)数据

完美图解物联网 IoT 实操:使用 JavaScript, Node.JS, Arduino, Raspberry Pi / 赵英杰著. —北京:电子工业出版社, 2017.9

(完美图解物联网技术系列)

ISBN 978-7-121-31936-5

I. ①完… II. ①赵… III. ①JAVA 语言—程序设计—图解 IV. ①TP312.8-64

中国版本图书馆 CIP 数据核字(2017)第 139473 号

策划编辑:林瑞和

责任编辑:葛娜

印刷:三河市双峰印刷装订有限公司

装订:三河市双峰印刷装订有限公司

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:787×980 1/16 印张:23.5 字数:473 千字

版次:2017 年 9 月第 1 版

印次:2017 年 9 月第 1 次印刷

定价:99.00 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 zltts@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:010-51260888-819, faq@phei.com.cn。

序

本书探讨的核心主题是串联、汇流整合。互联网从最初的人际沟通桥梁，拓展成机器之间协同合作的交流管道。



本书强调的是串联网络软件和微电脑控制板，以 JavaScript 为主轴，开发网络应用程序、手机 App、互动网页、数据库程序和操控微电脑。笔者假设读者曾经使用 Arduino 开发工具写过 Arduino 程序。

本书使用在物联网应用中两种最根本的语言来创建应用程序。

- Arduino 的 C 语言：Arduino 系列控制板是电子互动 Maker 的最爱，许多非 Arduino 控制板，也强调跟 Arduino 的程序开发环境或者控制引脚相容，尽管控制板厂商不断推陈出新，但是 C 语言仍是当家主流。
- JavaScript：在全球信息网（World-Wide Web）中，最通行的语言是 JavaScript，也是所有网页/网站开发人员必知必会的程序语言。经过数年的发展，JavaScript 的应用也扩展到计算机应用程序、手机 App 开发，甚至操控微电脑控制板领域。

本书中采用的控制板类型比较多元化，不限于 Arduino，但它们的控制程序都是用 C 或 JavaScript 编写的。

当今市场上不乏大师级的 JavaScript 相关著作，像本书的每个章节，从互动网页、网站数据库程序设计、动态图表到手机 App 开发，都有专门的参考书籍，但它们都鲜

少提及 JavaScript 在物联网和微控制器方面的整合应用。所以本书的定位，并不是要取代这些书籍，而是一种延伸和扩充。

感谢电子工业出版社编辑林瑞和先生以及葛娜女士在出版过程中给予的宝贵意见和指正，让本书顺利问世。衷心期盼本书能帮助读者了解物联网程序设计，进而开发出自己的云端物联网应用。

赵英杰

2017 年 6 月 15 日于

中国台湾·糖安居

swf.com.tw

轻松注册成为博文视点社区用户 (www.broadview.com.cn), 扫码直达本书页面。

- **下载资源**：与本书相关的 Arduino 整合开发工具 (Windows 版和 Mac 版)、示例文件 (各章节的源代码以及文件对照说明)、Arduino 库 (搭配 Arduino 示例代码使用) 等内容，请在 <http://www.broadview.com.cn/31936> 的“下载资源”区域下载。
- **提交勘误**：您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分 (在您购买电子书时，积分可用来抵扣相应金额)。
- **交流互动**：在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/31936>



目录



无所不在的 JavaScript 与物联网设备 1

0.1 JavaScript 与物联网 3

0.2 MPU、MCU 与 SoC 介绍 6

0.3 章节导读 9



JavaScript 入门 13

1.1 JavaScript 入门 14

1.2 JavaScript 语言基础 19

1.3 函数定义与事件处理程序 29

1.4 定时产生随机数 32

1.5 Array (数组) 对象 34

1.6 认识 Object (对象) 类型 36

1.7 BOM (浏览器对象模型) 与 DOM (文档对象模型) 41

动手做 操作网页对象 43

1.8 在网页中嵌入 JavaScript 44

1.9 事件触发程序 46



认识 jQuery 库 53

2.1 jQuery 简介 54

2.2 jQuery 语法基础 59

2.3 AJAX 技术：动态更新 HTML 内容 64

动手做 从 Arduino 输出纯文本温度 66

2.4 网页消息交换格式 69

2.5 Arduino 输出 JSON 消息 73

2.6 调整灯光亮度的网页界面	76
动手做 接收调光值的 Arduino 网站程序	76
2.7 使用 jQuery UI 附加网页用户界面元素	79
动手做 制作滑块界面网页	82
2.8 AJAX 的安全限制	85



Node.js 入门

3.1 在电脑以及树莓派上安装 Node.js	94
3.2 Node 编程起步	99
3.3 使用 http 模块创建网站服务器程序	103
3.4 事件驱动、非阻塞 I/O 示范：读取文件	107
3.5 global（全局）对象与模块文件	113
3.6 安装与管理模块：使用 npm 工具程序	115
3.7 使用 Express 框架开发网站应用程序	118
3.8 接收 URL 数据	122
3.9 接收与处理 POST 数据	125
3.10 使用 package.json 管理 Node 项目的模块程序	129
动手做 从 Arduino 传递温湿度值给 Node 网站	131
3.11 使用 JSONP 格式跨网域访问信息	136



Node.js 串口通信与树莓派 GPIO 控制

4.1 安装编译 Node 模块所需的软件	144
4.2 安装 C/C++ 编译器	147
动手做 Node.js 串口通信	150
动手做 从 Node.js 发送串口数据	154
4.3 通过 Node.js 运行系统指令（运行 raspistill 指令拍照）	157
4.4 树莓派的 GPIO 简介	162
4.5 Node.js 程序 GPIO 控制：使用 onoff 模块	166
动手做 GPIO 输入 / 输出信号练习：按开关闪烁 LED	167
动手做 通过红外传感模块拍摄照片	170
4.6 树莓派 GPIO 集成 Arduino 控制板	172

4.7 5V 和 3.3V 电压准位转换	182
4.8 MOSFET 逻辑电位转换电路原理解说	185



使用霹雳五号操控 Arduino	189
------------------------	-----

5.1 霹雳五号：用 JavaScript 控制 Arduino	190
--	-----

动手做 用霹雳五号指挥 Arduino 闪烁 LED	192
---	-----

动手做 启用上拉电阻并读取开关信号	200
--------------------------------	-----

动手做 模拟输入与 PWM 输出程序实验	201
-----------------------------------	-----

动手做 检测温度	203
-----------------------	-----

动手做 控制舵机	207
-----------------------	-----

动手做 控制 LED 矩阵显示图像	209
--------------------------------	-----

5.2 使用 socket.io 创建实时连接	211
-------------------------------	-----

动手做 创建实时通信程序	214
---------------------------	-----

动手做 浏览器与矩阵 LED 作画	217
--------------------------------	-----



电子邮件、串流视频、计算机视觉与操控舵机	227
----------------------------	-----

6.1 通过 Node 发送电子邮件	228
--------------------------	-----

动手做 云搜证：拍照自动发送 E-mail	233
------------------------------------	-----

动手做 串流视频：推播实时影像	234
------------------------------	-----

6.2 使用 MJPG 压缩与串流视频	240
---------------------------	-----

6.3 控制舵机云台	246
------------------	-----

动手做 使用方向键操控舵机云台	247
------------------------------	-----

动手做 摄像头云台的 Node 服务器代码	249
------------------------------------	-----

6.4 使用触摸屏虚拟摇杆操控摄像头云台	250
----------------------------	-----

6.5 浏览器连接游戏控制器（电玩控制器）	254
-----------------------------	-----

动手做 使用 gamepad.js 库创建游戏器操作的网页程序	255
--	-----

6.6 计算机视觉（computer vision）应用	260
------------------------------------	-----

6.7 使用 tracking.js 侦测人脸	264
-------------------------------	-----

动手做 脸孔侦测与舵机联动	270
----------------------------	-----

动手做 侦测颜色	273
-----------------------	-----



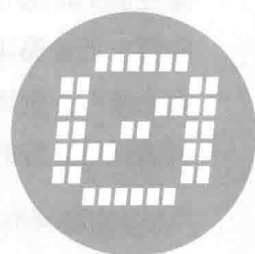
使用 MongoDB 数据库以及 ejs 模板引擎创建动态网页..... 279

7.1 数据库简介.....	280
7.2 安装 MongoDB.....	285
7.3 MongoDB 的基本操作.....	289
7.4 查询数据.....	294
7.5 更新与删除数据.....	298
7.6 使用 mongoose 包连接 MongoDB 数据库.....	299
7.7 连接 MongoDB 并读取数据的程序.....	303
动手做 存储 Arduino 上传的温湿度数据.....	307
7.8 认识模板引擎.....	309
动手做 在 ejs 模板中显示最近 10 笔温湿度数据.....	315
动手做 分页显示数据.....	318



数据可视化——使用 C3.js 与 D3.js 绘制图表..... 321

8.1 使用 C3.js 绘制图表.....	322
8.2 显示动态平移的实时线条图.....	327
动手做 使用数据库数据描绘折线.....	331
8.3 动态显示 Arduino 检测模拟值.....	334
动手做 使用图表动态显示传感器数据.....	335
8.4 简易数字滤波.....	340
8.5 使用 D3 库.....	344
8.6 认识 SVG.....	348
8.7 使用 D3 绘制 SVG 图像.....	353
8.8 使用动态数据绘制折线图.....	355
8.9 结合数据动态附加 HTML 元素.....	364

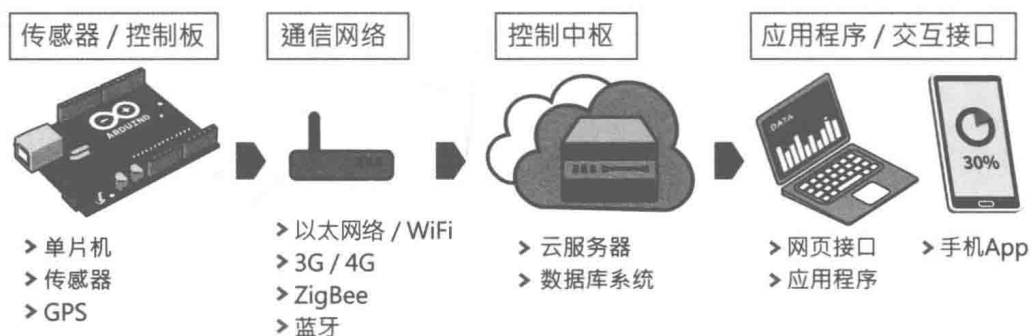


无所不在的JavaScript 与物联网设备



物联网 (Internet of Things, IoT) 一词, 是由美国麻省理工学院 Auto-ID 中心主任 Kevin Ashton, 于 1999 年进行 RFID 研究时所提出的概念。物联网是指为每个东西, 包括一般物品 (例如: 超商以及它所贩售的所有商品)、传感器, 甚至人类和动物, 都标上唯一识别码 (Unique Identifier, 如: 条形码、IP 地址、身份证号码等), 彼此之间通过约定好的通信协议, 利用互联网相连、分享数据。

物联网不仅是两个设备之间的互相连接, 它们通常也连接到负责收集数据, 以及协调这些设备运作的 (云) 服务器。此外, 控制器、传感器等物联网设备, 通常也都要具备让用户操作和监控的界面, 无论是网页还是 App 形式。下面是物联网的基本架构:



可连接互联网设备的数量, 在 2011 年时超越地球总人口数。这些设备包含消费性电子产品、工厂机器、家电、医疗器材、传感器等。以往, 数据通过人力输入到计算机 (例如, 抄录水、电、燃气用量), 现在多半则是由设备直接提供数据给另一个设备。



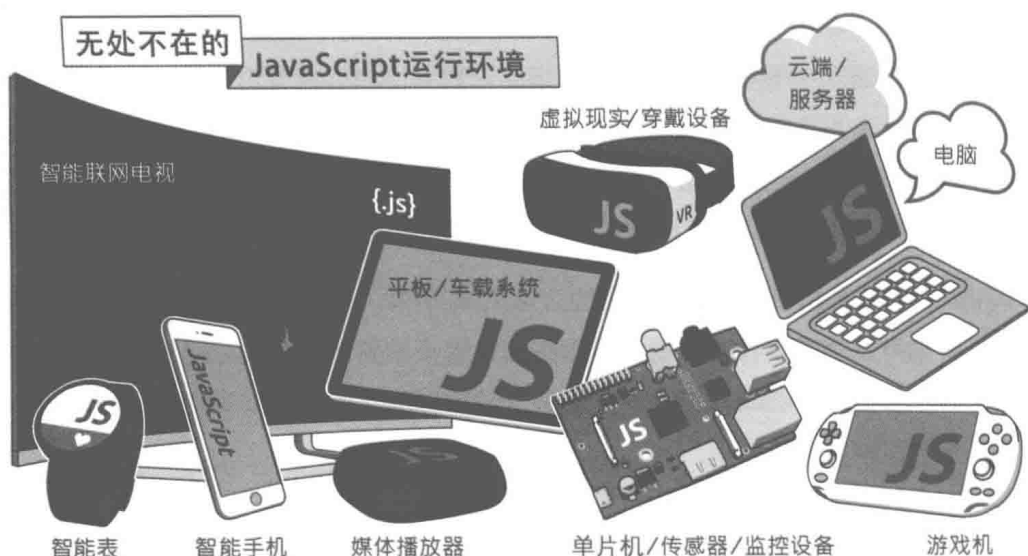
举例而言，你的手机认得“你”和“家人”，当你的小孩抵达校园时，“学校”会自动发送短信通知；手机可以记录你的运动习惯并协调智能型温控器，在你到家之前调整好室温；若是下班快抵达家门时，手机可以自动过滤公司电话，并通知住家的房子自动打开车库门。宅配的货车可实时更新网购商品位置和预计送达时间，也能读取送货路线的交通状况，并适时提醒或规划避开车流量大的路线。

所以，机器和机器相互连接、协同合作（Machine to Machine, M2M），就能完成代理人或者贴身秘书的工作。随着宽带网络基础建设的普及、各式云服务推陈出新，加上传感器、通信芯片和单片机技术的提升与价格下降，使得“物联网”从概念融入真实的日常生活中。

0.1 JavaScript与物联网

许多嵌入式系统采用的是运算性能、存储器容量相对较低的8位单片机，它们也通常采用专属的库和开发工具，并且多采用C/C++语言开发。就像电视机、空调等家电，逐渐脱离“单独”运作的时代，为了抢食物联网市场大饼，原本软硬件较为封闭的嵌入式控制器，也开始拥抱开放的互联网和Web标准。

这股风潮，也将原本称霸Web前台技术的JavaScript（注：在网页上提供交互功能的编程语言）顺势带入物联网。例如，Pebble智能表提供的开发工具PebbleKit和Pebble.js，让你用JavaScript来开发手表的App。又如，2014年被Google收购的Nest Labs公司推出的Nest温度控制器，也有供JavaScript代码使用的API（Application Interface，应用程序接口，也就是提供让外部程序控制的渠道），因此，程序员可用JavaScript读取或调控Nest温控器的状态。基本上，凡具备Web浏览器的设备，就能运行JavaScript。



在浏览器内操作的应用（像 YouTube, Google Docs 和 Office 365），叫作 Web 应用。Web 应用分成前台和后台（服务器端）两大部分，以电子邮件服务网站（如：QQ 和 Gmail）为例，显示在浏览器里电子邮件操作的界面（网页）是前台，提供用户填写信件的界面，负责发送邮件的则是后台。网页界面的架构通过 HTML 定义，外观样式由 CSS 定义，动态改变字体样式、大小、插入表情图案等交互功能则是用 JavaScript 写成的。



JavaScript 语言具有下列几个特色：

- 所有浏览器都支持，也是网页的标准语言。
- 语法简单，学习门槛低，容易入门。
- 不需要特殊的开发工具，在计算机、平板电脑或手机上，用文本编辑软件和浏览器就能开发和调试 JavaScript 代码。
- 有为数众多的开发者，以及书籍、网络等参考资源和示例。

用于服务器和单片的 JavaScript

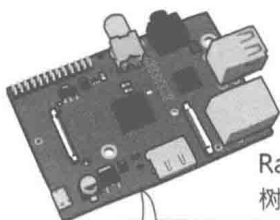
当 QQ 用户单击“发送”按钮之后，信件内容会从浏览器传给 QQ 服务器，而在服务器上运行的代码，将负责寄送信件和保存邮件等功能。不像浏览器，在服务器端运行的编程语言并没有统一，常见的语言和运行环境有 PHP、ASP.NET、Ruby 等。

其实，服务器端程序也能用 JavaScript 语言开发。可在浏览器之外独立运行 JavaScript 代码的 Node.js 于 2009 年问世，并相继被沃尔玛、PayPal（在线支付款项系统）和 Netflix（网飞，在线电视）等大型企业所采纳，用来建置可应付购物旺季高流量的高性能网站后台程序。

Node.js 并不是第一个可在浏览器外运行的 JavaScript 环境，但它是最成功的一个。



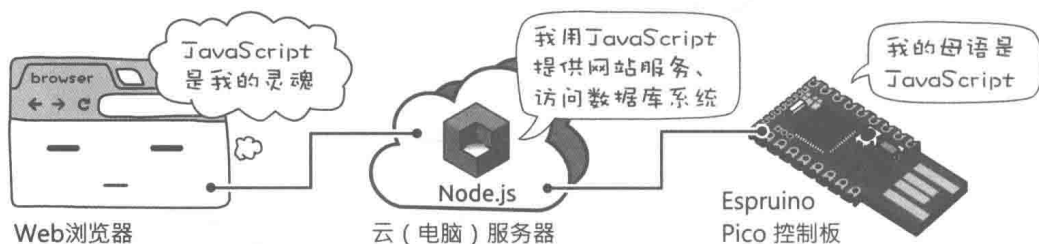
Node.js 能够在 Windows、Mac OS X 以及 Linux 系统中运行；不只是个人计算机，风靡创客的 Raspberry Pi（以下称“树莓派”）、BeagleBone Black（<http://beagleboard.org/BLACK>），以及 9 美元的开源微电脑 C.H.I.P.（<https://getchip.com/>），还有基于 x86 处理器架构的 Intel Galileo 控制板也都能运行 Node.js。



Raspberry Pi
树莓派控制板

我也可以运行 Node.js，
用 JavaScript 控制外设。

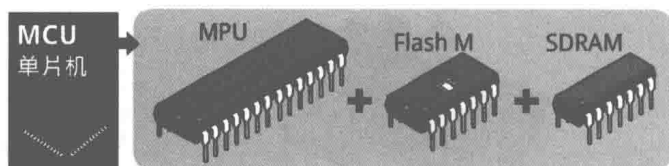
某些采用 32 位处理器的控制板，更直接把 JavaScript 语言当作“母语”，例如 Espruino (<http://espruino.com/>) 和 Tessel 2 (<http://tessel.io/>)，而三星电子公司在 2015 年 5 月发表了一个“瘦身”版的 JavaScript 语言，叫作 JerryScript (<https://samsung.github.io/jerryscript/>)，可运用在仅有 64KB 内存的单片机上。换句话说，在这些控制板上读取和设置 I/O 脚，都用 JavaScript。因此，从前台网页、后台服务器，到单片机和传感器终端，都能用相同的 JavaScript 语言编写；学习 JavaScript，也就能同时掌控前台、后台和感测端的软硬件！



0.2 MPU、MCU与SoC介绍

处理器芯片依照功能来区分，大致分成如下三种类型。

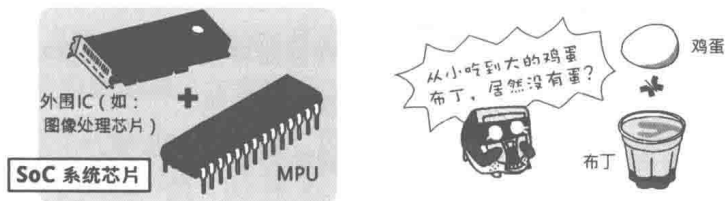
- **MPU:** 全名是 Microprocessor Unit (微处理器单元)，其实就是 CPU (中央处理器)，可运行指令，进行运算和逻辑处理。
- **MCU:** 全名是 Microcontroller Unit (单片机)，把微处理器 (MPU) 和快闪存储器、主存储器 (内存) 包在同一个芯片里，相当于一台微型计算机，耗电量低，但是处理器的性能不高 (时钟频率在 200MHz 以内)，而且存储器容量不大 (以 KB 为单位)。



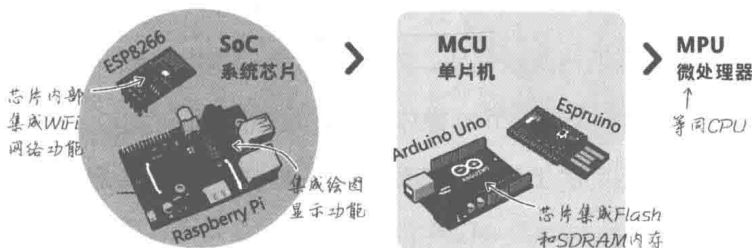
当今的 MCU 通常也会集成模拟 - 数字转换器 (模数转换)、USB 和其他外围界面

控制器。Arduino UNO 的 ATmega328 芯片属于 MCU。

- SoC : 全称为 System on a Chip (系统芯片), 集成微处理器和特定功能, 例如, 图像处理单元 (显示卡)、WiFi 网络、蓝牙、音效处理等。智能手机以及某些个人计算机的处理器, 都属于 SoC。采用这类处理器的设备, 通常需要较高速的运算性能 (时钟频率达数百 MHz 至数 GHz) 以及较大的存储器容量 (单位是 MB 或 GB), 所以存储器不在同一个芯片上, 耗电量也较大。



树莓派的处理器芯片属于 SoC。就功能而言, SoC 大于 MCU :



以穿戴设备应用来说, MCU 足以胜任普通数字手环所需, 而且能长时间运作; 具备多媒体功能的智能手表, 就必须采用 SoC。某些智能手表甚至同时搭载 SoC 和 MCU, 前者处理复杂的多媒体运算, 后者用于连接传感器以及电源管理。



本书采用的硬件设备

本书的示例采用下列控制平台和设备。

- 个人计算机 (Windows、Mac OS X 或 Linux) : 运行 Node.js 代码, 用 JavaScript 构

建网站服务器、实时信息交换平台、连接数据库以及控制串口连接的 Arduino 板。



- **Arduino UNO**：受限于单片机的性能和存储器容量，它无法直接运行 JavaScript 程序。Arduino UNO 板在本书中主要扮演两种角色：
 1. 搭配以太网网络卡，采用 C 语言当作网络控制节点，或者传感器节点，并且以被广泛使用的 JavaScript 数据交换格式（JSON），传递信息给前台或者网站服务器，详细请参阅第 1 章。
 2. 充当计算机的串口界面控制器，接收来自计算机（或者其他可运行 JavaScript 的高端微控制板，如：树莓派）的指令操控，请参阅第 5 章。
- **树莓派**：安装官方的 Raspbian 操作系统，运行以下工作：
 1. 个人计算机相同的工作（创建网站服务器、提供数据库服务、串口控制……）。
 2. 使用 JavaScript 访问与控制 GPIO 界面（参阅第 4 章）。
 3. 通过 JavaScript 运行 Linux 系统命令，配合专属摄像头拍摄照片以及串流视频（参阅第 6 章）。

