

<<Arduino技术内幕>>

图书基本信息

书名：<<Arduino技术内幕>>

13位ISBN编号：9787115311924

10位ISBN编号：7115311927

出版时间：2013-4

出版单位：人民邮电出版社

作者：惠特

译者：翁恺

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Arduino技术内幕>>

内容概要

《Arduino技术内幕》摒弃了大量基础知识，全面深入揭秘Arduino的内部工作原理。其中，它通过“LED闪灯”、“LED照明”、“数字钟”等示例作品由浅入深、由易而难介绍Arduino开发，并最后以一个“自动机器人”示例总结全书，不仅介绍了可用的软硬件及软硬件间的协作，阐述了软硬件的选取与设计，而且分析了项目管理（包括文档化、团队工作与协作开发以及如何选择许可方式）方面的内容。另外，读者还可从《Arduino技术内幕》了解设计原则与优秀的编程技巧、编程风格，学会动手设计实现优秀的Arduino作品。《Arduino技术内幕》适合所有Arduino用户学习参考。

<<Arduino技术内幕>>

作者简介

作者：（美国）惠特（Dale Wheat）译者：翁恺 惠特（Dale Wheat），DPRG（Dallas Personal Robotics Group，世界上最早的个人机器人俱乐部）两届主席，专门研究电子学和嵌入式系统并传授电子学、单片机和焊接技术方面的知识，多次在技术类杂志和业余爱好者喜闻乐见的杂志（包括CircuitCellar、MAKE和Elektor）上发表文章。

另外。

他还在个人网站上销售自己设计和DIY的电子元件。

全职写作之前，他曾作为计算机程序员和系统分析师为IBM、MCI和DGTE（已并入Verizon）等公司提供咨询服务。

另外，他还著有Building Your Own Electronics Lab：A Guide to Setting Up Your Own Gadget Workshop一书。

。

<<Arduino技术内幕>>

书籍目录

第1章 硬件 1 1.1 什么是Arduino 1 1.2 ArduinoUno 2 1.2.1 处理器 3 1.2.2 串口 4 1.2.3 电源 5 1.2.4 扩展插座 6 1.2.5 盾板 8 1.3 ArduinoMega2560 11 1.4 之前的硬件 12 1.4.1 ArduinoSerial 12 1.4.2 ArduinoUSB 13 1.4.3 ArduinoExtreme 14 1.4.4 ArduinoNuovaGenerazione (新一代) 14 1.4.5 ArduinoDiecimila 14 1.4.6 ArduinoDuemilanove 15 1.4.7 ArduinoMega 15 1.5 谁在做Arduino 15 1.5.1 正式许可的产品 15 1.5.2 其他人 17 1.6 自己做Arduino 17 1.6.1 Arduino印刷电路板 17 1.6.2 面包板Arduino 18 1.7 小结 19 第2章 软件 21 2.1 主机和目标机 21 2.2 一步步来 21 2.2.1 第一步：写代码 22 2.2.2 第二步：编译代码 27 2.2.3 第三步：给芯片编程 28 2.2.4 第四步：测试和调试 29 2.2.5 第五步：重复 29 2.3 半自动化 29 2.4 更进一步 33 2.5 小结 33 第3章 AtmelAVR 34 3.1 起源 34 3.2 AVR芯片家族 34 3.3 若有疑问：芯片器件手册 35 3.4 芯片封装 35 3.4.1 双列直插 (DIP) 35 3.4.2 表面安装器件 (SMD) 36 3.4.3 多出来的引脚 37 3.5 管脚定义 37 3.6 AVR内核 43 3.6.1 时钟源 45 3.6.2 地址空间 46 3.6.3 指令集 49 3.7 片内外围设备 54 3.7.1 通用输入 / 输出 (I/O) 55 3.7.2 外部中断 55 3.7.3 定时器 / 计数器 56 3.7.4 USART 57 3.7.5 两线串行接口 (TWI) ，即I2C 57 3.7.6 模拟输入 58 3.8 小结 58 第4章 支持硬件 59 4.1 电路图 59 4.1.1 元件类型 60 4.1.2 元件编号 60 4.1.3 元件值 61 4.1.4 元件值误差 61 4.1.5 元件的其他参数 62 4.1.6 连接 62 4.2 给电路板供电 63 4.2.1 柱式电源插座 63 4.2.2 输入电源调节 63 4.2.3 稳压器 65 4.2.4 电源电路的演变 65 4.3 串口 67 4.4 处理器 69 4.4.1 功耗 69 4.4.2 I/O 驱动能力 70 4.4.3 —RESET信号 70 4.4.4 时间基准 70 4.4.5 去耦电容 71 4.4.6 闪灯 71 4.5 扩展空间 71 4.6 结构外形 72 4.7 通用串行总线 (USB) ：信号加电源 73 4.8 小结 74 第5章 Arduino软件 75 5.1 开源软件 76 5.2 多平台支持 76 5.3 Arduino的传承与发展 76 5.4 软件安装 77 5.5 实践 77 5.6 用户界面 78 5.6.1 File (文件) 菜单 79 5.6.2 Edit (编辑) 菜单和Edit关联菜单 80 5.6.3 Sketch (程序) 菜单 80 5.6.4 Tools (工具) 菜单 81 5.6.5 Help (帮助) 菜单 82 5.7 小结 82 第6章 优化 83 6.1 可行性 83 6.2 压缩闪灯程序 83 6.2.1 闪烁是如何实现的 84 6.2.2 衡量节省空间的优化措施 84 6.2.3 代码分析 84 6.2.4 没有pinMode () 的生活 85 6.2.5 缩写和简语 86 6.2.6 二进制写法 87 6.2.7 深入分析 87 6.2.8 轻松翻转 88 6.2.9 进一步精简 88 6.2.10 更有效地 “ 浪费时间 ” 89 6.2.11 更低级别的代码 89 6.3 用简单的串口通信来节省空间 90 6.3.1 “ Hello , world ! ” 做了什么 91 6.3.2 写配置寄存器 91 6.3.3 发送数据 93 6.3.4 一串字符 94 6.3.5 输出数字 95 6.4 节省SRAM 96 6.4.1 测量要用的SRAM 96 6.4.2 最小裸机 98 6.4.3 内存分区 99 6.4.4 变量的位置 100 6.4.5 使用恰当的数据类型 100 6.4.6 再探字符串 101 6.5 低功耗，还是高速度？ 102 6.6 电子测量 103 6.6.1 用Arduino做测试仪器 103 6.6.2 尽可能快 105 6.6.3 让它慢下来 108 6.6.4 进一步降低功耗 110 6.7 小结 112 第7章 硬件加软件 113 7.1 可用的外围设备 113 7.1.1 串口 113 7.1.2 通用数字输入输出 119 7.1.3 定时器和计数器 121 7.1.4 脉宽调制 (PWM) 输出 124 7.1.5 模拟输入 129 7.1.6 外部中断 133 7.1.7 中断手册 135 7.2 小结 137 第8章 示例作品 139 8.1 不只是闪烁的LED：从简单开始 139 8.1.1 足够慢 142 8.1.2 基本完成优化的六通道调光器 144 8.1.3 变暗了 147 8.2 闪烁LED的其他用处 154 8.2.1 红外遥控 154 8.2.2 TV—B—Gone 156 8.3 许多闪烁的LED 157 8.3.1 一个直接驱动的例子 157 8.3.2 用LED驱动器直接驱动 159 8.3.3 多路复用技术 160 8.4 数字钟 174 8.4.1 精度 179 8.4.2 用户界面 180 8.4.3 其他功能 180 8.5 小结 181 第9章 项目管理 182 9.1 文档 182 9.1.1 源代码注释 183 9.1.2 空格 184 9.1.3 按你的意思编程，按你的代码解释 185 9.1.4 自动文档编制 186 9.1.5 写给你的 “ 读者 ” 看 186 9.1.6 硬件文档 187 9.1.7 再进一步 188 9.2 团队工作和协同开发 189 9.2.1 博客 190 9.2.2 论坛 190 9.2.3 维基 191 9.2.4 修订控制系统 191 9.2.5 关于修订版号和版本号的说明 192 9.2.6 放作品的网站 192 9.3 选择许可方式 195 9.3.1 专利和商标 195 9.3.2 版权 196 9.3.3 开源 197 9.3.4 公共域 198 9.4 小结 199 第10章 硬件设计 200 10.1 了解硬件 200 10.1.1 必须要有的东西 201 10.1.2 想要有的东西 202 10.2 红外接近传感器 203 10.2.1 一个简易原型 205 10.2.2 一些简单的改进 207 10.2.3 印制电路板 210 10.2.4 PCB布局技术 211 10.2.5 第一次尝试 212 10.2.6 更小的版本 214 10.2.7 连接 215 10.3 自己定制的Arduino 216 10.3.1 兼容现有Arduino和盾板 216 10.3.2 电源的可能选择 217 10.3.3 处理器选择 218 10.3.4 还有什么？ 219 10.4 设计用的软件 219 10.4.1 CadSoftEAGLE 220 10.4.2 EAGLE贴士 220 10.5 小结 221 第11章 软件设计 222 11.1 Arduino高级话题 222 11.1.1 写Arduino库 222 11.1.2 备选内核 229 11.2 不用Arduino 239 11.2.1 再探BareMetal 239 11.2.2 其他开发环境 242 11.3 小结 245 第12章 组网 246 12.1 点对点组网 246 12.1.1 通过串口交谈 246 12.1.2 Arduino间的对话 248 12.2 MIDI：乐器数字接口 260 12.3 互联网 264 12.4 小结 268 第13

<<Arduino技术内幕>>

章 更多示例作品 269 13.1 一个自动机器人 269 13.2 电源 270 13.3 动作控制 270 13.4 传感器 276 13.4.1 光敏传感器 276 13.4.2 接触传感器 277 13.4.3 非接触传感器 278 13.4.4 声音传感器 279 13.4.5 指示器、控制器和其他形式的通信 279 13.5 控制系统 281 13.5.1 开环系统 281 13.5.2 闭环系统 282 13.6 示例机器人作品 283 13.6.1 一个练习机器人 284 13.6.2 下一个机器人 304 13.6.3 你的终极机器人 318 13.7 小结 319

<<Arduino技术内幕>>

章节摘录

版权页：插图：ATmega328和ATmega2560的时钟系统基本上是一样的，都有一个片内振荡器，其时钟频率由外部晶体或谐振器决定。

这个振荡器有两个运行模式：低功率模式消耗最小的功率，但是不能输出时钟信号到芯片外面；全幅模式则增加了功率消耗。

另外，两种芯片都可以从外部信号源获得时钟。

如果系统内已经有这样的信号了，芯片就不再需要专门的定时部件了。

片内还有两个RC（电阻电容）振荡器，频率分别是8.0 Mhz和128 KHz。

8 MHz的振荡器频率可以由软件校准（调整），但是没有石英晶体或陶瓷谐振器来得精确。

片内有一个系统时钟预分频器，将系统时钟以2的幂次分频，分频比可以从1：1到1：256。

降低系统时钟就降低了功率消耗。

如果主系统时钟是由两个校准过的电阻 / 电容（RC）振荡器之一产生的，还可以使用一个外部低频时钟晶体（一般是32 KHz）来驱动需要真实时钟的应用程序中的定时器 / 计数器2。

即使主处理器在睡眠模式或省电模式不工作了，这个定时器外围设备还会继续运行并保持计时的精确。

3.6.2地址空间 AVR内核要访问几个不同的内存和I / O设备阵列。

AVR体系结构本身是基于哈佛体系结构的，哈佛结构中，程序和数据存储空间是分离的，这与冯·诺依曼体系结构不同。

在后者中，两个存储空间是集成在一起、相互交叉、可互换的。

1.程序存储器 处理器要执行的实际机器语言指令存储在程序存储器（program memory）中。

在AVR中，这个存储器是片内的可编程16位宽flash阵列。

因为程序存储器是可以在程序运行中自己重写的，所以也可以存储一定数量的数据。

但是写入的时间比较长，可以擦写的次数也有限，这使得它不适合存储易失数据。

程序存储器有时候也被看做是ROM，即只读存储器（read—only memory）。

这又是一个穿越的术语，已经不再准确反映现在的存储器功能和性能了。

ROM这个术语通常用来描述系统内存中持久的或非易失的（non—volatile）部分，即使掉电也不会丢失数据。

过去用这个术语来和RAM作区分。

RAM的本意是随机存取内存（random—access memory），也是一个不当的用词，其表达的是可写的或易失的（volatile）内存器件。

Atmel在其AVR和其他半导体产品中使用的闪存（flash memory）技术具有非易失性的所有优点，同时又易于重新写入，而不依赖于复杂或昂贵的芯片编程硬件。

Arduino实现了一个引导装载程序，借此利用了AVR的可自编程特性。

引导装载程序是常驻的一小段固件，与主机PC通信，让编译好的程序可以从PC送过来保存在AVR的程序存储器里。

这样就不再需要任何形式的专用芯片编程硬件了。

<<Arduino技术内幕>>

编辑推荐

- (1) 教你动手制作机器人！
- (2) 全面深入理解Arduino工作原理，灵活进行Arduino开发
- (3) 依循设计原则，按需设计使用Arduino软硬件

<<Arduino技术内幕>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>