



第 1 章 了解 RISC-V	1
1.1 RISC-V 指令架构的演进历史	1
1.2 RISC-V 处理器家族	2
1.2.1 RISC-V 处理器内核	2
1.2.2 RISC-V SoC 平台	3
1.2.3 RISC-V SoC 芯片	4
1.3 RISC-V 嵌入式软件生态	5
1.3.1 开源 GNU 工具链软件	6
1.3.2 IAR Embedded Workbench	6
1.3.3 SEGGER Embedded Studio	7
1.3.4 嵌入式操作系统	9
1.4 RISC-V 内核、平台和芯片的选择	10
1.5 RISC-V 处理器与应用展望	11
1.6 本章小结	13
第 2 章 RISC-V 处理器芯片	14
2.1 GD32VF103 微控制器	14
2.1.1 芯片简介	14
2.1.2 芯片内核	15
2.1.3 GD32VF103 开发板	15
2.2 NXP RV32M1 微控制器	17
2.2.1 芯片简介	17
2.2.2 RV32M1 内核	17
2.2.3 RV32M1 开发板	17
2.3 WCH CH32V103 微控制器	19
2.3.1 芯片与内核简介	19
2.3.2 CH32V103 开发板	19
2.4 SiFive Freedom E310	20

2.4.1	E310 芯片和 E31 内核介绍	20
2.4.2	FE310 开发板	21
2.5	Kendryte K210	22
2.5.1	Kendryte K210 芯片	22
2.5.2	Kendryte K210 开发板介绍	23
2.6	CH2601 平头哥生态芯片	25
2.6.1	CH2601 MCU 简介	25
2.6.2	CH2601 开发板和开发环境简介	25
2.7	本章小结	27
第 4 章	RISC-V 软件开发工具	28
3.1	RISC-V 软件生态概述	28
3.2	RISC-V GNU 工具链	30
3.3	Nuclei Studio 开发环境	30
3.3.1	Nuclei Studio 简介	30
3.3.2	Nuclei Studio 安装	31
3.3.3	启动 Nuclei Studio	31
3.3.4	编译项目	33
3.3.5	运行和调试项目	33
3.4	SEGGER Embedded Studio 开发环境	35
3.5	IAR Embedded Workbench 开发环境	37
3.5.1	IAR RISC-V 评估套件	37
3.5.2	快速上手 IAR RISC-V 评估套件	38
3.6	Freedom Studio 开发环境	41
3.6.1	使用 Freedom Studio 创建工程	41
3.6.2	使用 Freedom Studio 编译和调试工程	43
3.7	本章小结	45
第 4 章	认识 RISC-V 内核	46
4.1	RISC-V 处理器架构	46
4.1.1	指令执行过程	46
4.1.2	RISC-V 概述	47
4.2	RV32I 指令集	51
4.2.1	RV32I 指令	51
4.2.2	寻址方式	58
4.3	RISC-V 异常和中断处理	60
4.3.1	RV32 特权模式与异常	61
4.3.2	机器模式异常管理寄存器	62

4.3.3 异常和中断响应过程	67
4.4 BumbleBee 内核	69
4.4.1 特 点	69
4.4.2 扩展指令集	69
4.4.3 CSR 寄存器	73
4.4.4 特权模式	75
4.4.5 中断控制器	76
4.4.6 定时器	79
4.4.7 内核低功耗机制	80
4.5 本章小结	80
第 5 章 RISC-V 软件开发	81
5.1 RISC-V 支持的软件环境	81
5.2 RISC-V 软件开发工具链	82
5.2.1 生成可执行程序	83
5.2.2 编译器	84
5.2.3 汇编器	85
5.2.4 链接器	87
5.2.5 代码生成示例	90
5.3 链接脚本	90
5.3.1 脚本格式	91
5.3.2 脚本示例	95
5.4 汇编语言	96
5.4.1 汇编程序格式	96
5.4.2 伪指令	98
5.4.3 汇编程序示例	98
5.5 函数调用规范(Calling convention)	102
5.5.1 函数调用过程	102
5.5.2 函数声明	103
5.5.3 调用约定	105
5.6 SEGGER Embedded Studio for RISC-V 示例	107
5.6.1 建立环境	107
5.6.2 创建工程	110
5.6.3 工程设置	113
5.6.4 工程源程序	116
5.6.5 程序生成与调试	118
5.7 本章小结	119

第 6 章 GD32VF103 微控制器	120
6.1 GD32VF103 结构	120
6.1.1 内核与总线结构	120
6.1.2 外设总线设备	121
6.1.3 外设内存映射	122
6.1.4 外设访问	124
6.2 GD32VF103 时钟系统	125
6.2.1 时钟管理	125
6.2.2 GD32VF103 复位和时钟单元	126
6.2.3 GD32VF103 时钟控制程序	127
6.3 GD32VF103 定时器	130
6.3.1 定时器原理	130
6.3.2 GD32VF103 定时器简介	131
6.3.3 定时器应用示例	133
6.4 GD32VF103 GPIO	136
6.4.1 GPIO 原理	136
6.4.2 GD32VF103 GPIO 简介	137
6.4.3 GPIO 编程示例	138
6.5 GD32VF103 串口	139
6.5.1 UART 原理	139
6.5.2 GD32VF103 USART	140
6.5.3 USART 通信示例	141
6.6 GD32VF103 I ² C 总线	145
6.6.1 I ² C 总线协议	145
6.6.2 GD32VF103 I ² C 接口	147
6.6.3 I ² C 总线数据传输	150
6.7 本章小结	155
第 7 章 GD32VF103 中断系统及应用	156
7.1 GD32VF103-EVAL	156
7.2 中断处理	157
7.2.1 中断原理	157
7.2.2 GD32VF103 中断系统	159
7.3 用按键控制 LED	162
7.3.1 电路原理	162
7.3.2 中断设置	163
7.3.3 中断服务程序	165

7.3.4 主程序	166
7.4 DMA 中断应用	167
7.4.1 DMA 原理	167
7.4.2 GD32VF103 DMA 控制器	168
7.4.3 GD32VF103 ADC	174
7.4.4 程序实现	180
7.5 触摸屏中断	185
7.5.1 扩展处理器外部设备	185
7.5.2 LCD 和触摸屏扩展	186
7.5.3 程序实现	188
7.6 本章小结	197
第 8 章 深入 RISC-V 程序开发	198
8.1 RISC-V 启动程序	198
8.1.1 启动过程	198
8.1.2 RISC-V 启动程序简介	199
8.2 巧用存储资源	205
8.2.1 C 语言存储修饰符	206
8.2.2 堆和栈	207
8.2.3 静态空间分配	210
8.3 C 语言程序优化	211
8.3.1 什么是优化	211
8.3.2 程序速度优化	211
8.3.3 存储资源优化	216
8.4 系统能耗优化	219
8.4.1 嵌入式系统能耗估计	219
8.4.2 嵌入式系统能耗优化	220
8.4.3 GD32VF103 功耗管理	222
8.4.4 看门狗处理	225
8.5 本章小结	229
第 9 章 嵌入式实时操作系统	230
9.1 嵌入式操作系统概述	230
9.1.1 什么是嵌入式操作系统	230
9.1.2 嵌入式操作系统的分类	231
9.1.3 嵌入式操作系统的应用	232
9.2 FreeRTOS 原理和功能	233
9.2.1 FreeRTOS 的发展历程	233

9.2.2 FreeRTOS 的基本功能	234
9.3 基于 RISC-V 的 FreeRTOS 移植	239
9.3.1 FreeRTOS 移植概述	239
9.3.2 基于 GD32VF103 的 FreeRTOS 移植	240
9.4 Tracealyzer 分析工具的应用	242
9.4.1 跟踪记录器的移植	242
9.4.2 运用 Tracealyzer 分析 FreeRTOS 应用	243
9.4.3 任务的时间量分析	244
9.5 本章小结	245
第 10 章 物联网操作系统及其应用	246
10.1 物联网操作系统的起源	246
10.2 物联网操作系统的基本功能	247
10.3 TencentOS tiny 简介	248
10.4 RT-Thread 简介	250
10.5 基于 RISC-V 和 TencentOS tiny 的空气质量检测终端应用	251
10.5.1 项目背景	251
10.5.2 项目方案	251
10.5.3 空气质量检测软件开发	256
10.5.4 腾讯物联网平台	257
10.6 基于 RISC-V 蓝牙开发板的 RT-Thread 多媒体应用	260
10.6.1 AB32VG1 开发板介绍	260
10.6.2 WAV 音频播放的应用	261
10.7 本章小结	268
第 11 章 基于 RISC-V 的电磁车设计	269
11.1 工作原理	269
11.1.1 电磁信号检测	269
11.1.2 PID 控制器	272
11.1.3 神经网络控制器	274
11.2 设计方案	275
11.2.1 总体方案	275
11.2.2 系统硬件结构	277
11.2.3 系统软件设计	281
11.3 系统软件实现	285
11.3.1 数据采集	285
11.3.2 算法实现	289
11.3.3 电机控制	292

11.3.4	DMA 中断服务程序	295
11.3.5	主程序	296
11.4	本章小结	296
第 12 章	高性能 RISC-V 处理器	297
12.1	处理器架构	297
12.1.1	C910 结构	297
12.1.2	内核系统	298
12.1.3	多核共享系统	299
12.2	编程模型	300
12.2.1	特权模式	300
12.2.2	寄存器	300
12.2.3	异常处理	302
12.2.4	数据格式	303
12.2.5	内存模式	305
12.3	指令集	306
12.3.1	浮点指令集	306
12.3.2	矢量指令集	307
12.4	虚拟内存管理	308
12.5	物理内存保护	309
12.5.1	PMP 控制寄存器	309
12.5.2	内存保护实现	310
12.6	中断控制器	311
12.6.1	内核局部中断控制器	311
12.6.2	平台中断控制器	314
12.7	本章小结	316
参考文献	317