

深入剖析STM32官方库及其使用的详细指南。

从什么是寄存器，怎么用寄存器编程开始讲起，再到如何构建库函数雏形，最后到如何熟练地使用固件库编程。全书由浅入深，步步为营，配套秉火STM32开发板，提供完整源代码，极具操作性。



单片机与嵌入式



YZLI0890372691

标准库

STM32库 开发实战指南

基于STM32F4

刘火良 杨森 编著



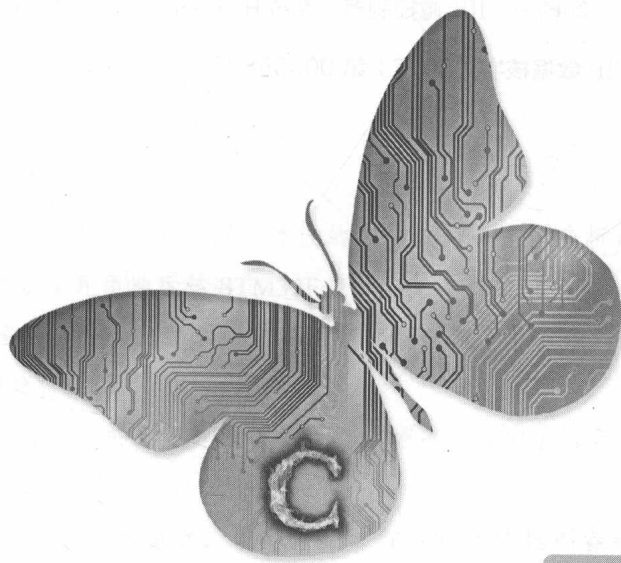
配套视频



机械工业出版社
China Machine Press



单片机与嵌入式



标准库

STM32库 开发实战指南

基于STM32F4

刘火良 杨森 编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

STM32 库开发实战指南：基于 STM32F4 / 刘火良, 杨森编著. —北京：机械工业出版社, 2017.2

(电子与嵌入式系统设计丛书)

ISBN 978-7-111-55745-6

I. S… II. ①刘… ②杨… III. 微控制器—系统开发—指南 IV. TP332.3-62

中国版本图书馆 CIP 数据核字 (2017) 第 005558 号

STM32 库开发实战指南：基于 STM32F4

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：缪 杰

责任校对：董纪丽

印 刷：三河市宏图印务有限公司

版 次：2017 年 2 月第 1 版第 1 次印刷

开 本：186mm×240mm 1/16

印 张：57.75

书 号：ISBN 978-7-111-55745-6

定 价：129.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88379426 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

前 言

本书的编写风格

本书着重讲解 STM32F429 的外设以及外设的应用，力争全面分析每个外设的功能框图和使用方法，让读者可以零死角地玩转 STM32F429。

基本每个章节对应一个外设，每章的主要内容大概分为 3 个部分，第 1 部分为简介，第 2 部分为外设功能框图分析，第 3 部分为代码讲解。

外设简介则是用作者自己的话把外设概括性地介绍一遍，力求语句简短，通俗易懂，避免照抄数据手册中的介绍。

外设功能框图分析是每章的重点，该部分会详细讲解功能框图各部分的作用，是学习 STM32F429 的精髓所在，掌握了整个外设的框图则可以熟练地使用该外设，熟练地编程，日后学习其他型号的单片机会得心应手。即使单片机的型号不同，外设的框图基本也是一样的。这一步的学习比较枯燥，但是必须下功夫钻研，方能学有所成。

代码分析则是讲解使用该外设的实验过程，主要分析代码流程和一些编程注意事项。在掌握了框图之后，学习代码部分则会轻而易举。

本书的学习方法

本书第 3 ~ 11 章连贯性非常强，属于单片机底层知识的讲解，对后面章节的学习起着“千斤顶”的作用，读者需要按照顺序学习，不可跳跃阅读。学完这部分之后，能力稍强的用户基本可以入门 STM32。其余章节连贯性较弱，可根据项目需要选择阅读。另外本书配套 200 集手把手教学视频和大量的 PPT，观看视频辅助学习，效果会更好。相关视频请到秉火论坛^①下载。

本书的参考资料

本书的参考资料为《STM32F4xx 中文参考手册》和《Cortex-M4 内核参考手册》，这两本是 ST 官方的手册，属于精华版，内容面面俱到，无所不包。限于篇幅问题，本书着重于 STM32F429 的功能框图分析和代码讲解，有关寄存器的详细描述则略过，在学习本书的时候，涉及寄存器描述部分还请参考上述两本手册，这样学习效果会更好。

本书的配套硬件和程序

本书配套的硬件平台为秉火 STM32F429 挑战者开发板，见图 0-1。如果配合该硬件平台做实验，必会达到事半功倍的学习效果，省去中间移植时遇到的各种问题。书中提到的配套工程程序可以在秉火论坛 (www.firebbs.cn) 下载。

本书的技术论坛

如果在学习过程中遇到问题，可以到秉火论坛 (www.firebbs.cn) 发帖交流，开源共享，共同进步。

鉴于作者水平有限，本书难免存在纰漏，热心的读者也可把勘误发到论坛，以便我们改进。祝你学习愉快！M4 的世界，秉火与您同行！

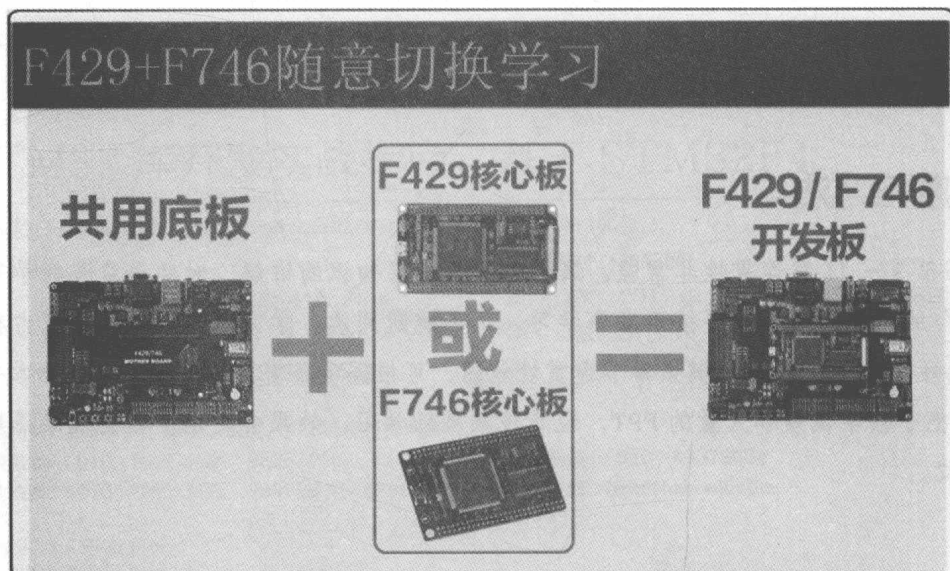


图 0-1 秉火 STM32F429 挑战者硬件资源

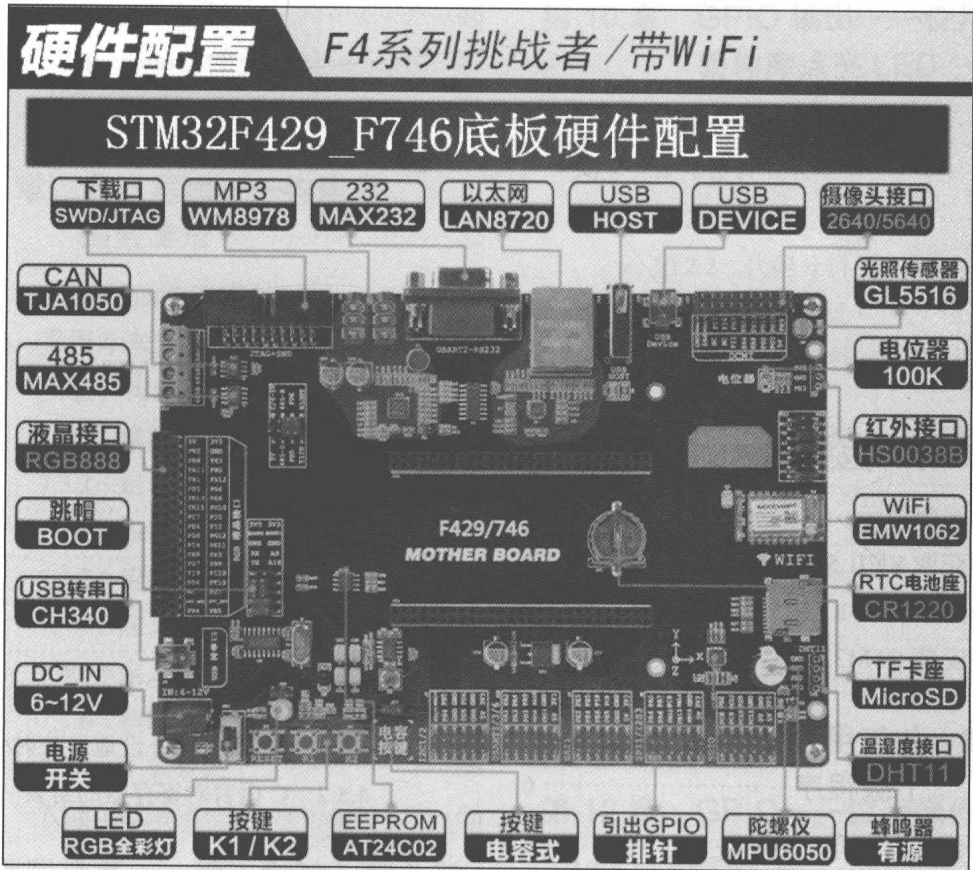


图 0-1 (续)

目 录

前 言

第 1 章 如何安装 KEIL5 1

- 1.1 温馨提示 1
- 1.2 获取 KEIL5 安装包 1
- 1.3 开始安装 KEIL5 1
- 1.4 安装 STM32 芯片包 3

第 2 章 如何用 DAP 仿真器 下载程序 6

- 2.1 仿真器简介 6
- 2.2 硬件连接 6
- 2.3 仿真器配置 6
- 2.4 选择目标板 9
- 2.5 下载程序 9

第 3 章 初识 STM32 10

- 3.1 什么是 STM32 10
- 3.2 STM32 能做什么 10
 - 3.2.1 智能手环 11
 - 3.2.2 微型四轴飞行器 12
 - 3.2.3 淘宝众筹 12
- 3.3 STM32 选型 13
 - 3.3.1 STM32 分类 13
 - 3.3.2 STM32 命名方法 14

- 3.3.3 选择合适的 MCU 14

第 4 章 寄存器 17

- 4.1 寄存器简介 17
- 4.2 STM32 的外观 17
- 4.3 芯片里面有什么 18
- 4.4 存储器映射 21
- 4.5 寄存器映射 23
 - 4.5.1 STM32 的外设地址映射 23
 - 4.5.2 C 语言对寄存器的封装 26

第 5 章 新建工程——寄存器版 31

- 5.1 新建本地工程文件夹工程 31
 - 5.1.1 新建本地工程文件夹 31
 - 5.1.2 新建工程 32
- 5.2 下载程序 38

第 6 章 使用寄存器点亮 LED 39

- 6.1 GPIO 简介 39
- 6.2 GPIO 框图剖析 39
 - 6.2.1 基本结构分析 39
 - 6.2.2 GPIO 工作模式 42
- 6.3 实验：使用寄存器点亮 LED 43
 - 6.3.1 硬件连接 44
 - 6.3.2 启动文件 44

6.3.3	stm32f4xx.h 文件	46	第 10 章	GPIO 输出——使用 固件库点亮 LED	88
6.3.4	main 文件	47	10.1	硬件设计	88
6.3.5	下载验证	51	10.2	软件设计	88
			10.2.1	编程要点	89
			10.2.2	代码分析	89
			10.2.3	下载验证	94
			10.3	STM32 标准库补充知识	94
第 7 章	自己写库——构建库 函数雏形	52	第 11 章	GPIO 输入——按键 检测	98
7.1	STM32 函数库简介	52	11.1	硬件设计	98
7.2	采用库来开发及学习的原因	53	11.2	软件设计	99
7.3	实验：构建库函数雏形	53	11.2.1	编程要点	99
7.3.1	修改寄存器地址封装	54	11.2.2	代码分析	99
7.3.2	定义访问外设的结构体 指针	55	11.2.3	下载验证	102
7.3.3	定义初始化结构体	59	第 12 章	GPIO——位带操作	103
7.3.4	定义引脚模式的枚举类型	60	12.1	位带简介	103
7.3.5	定义 GPIO 初始化函数	62	12.1.1	外设位带区	103
7.3.6	使用函数点亮 LED	64	12.1.2	SRAM 位带区	104
7.3.7	下载验证	66	12.1.3	位带区和位带别名区 地址转换	104
7.3.8	总结	66	12.2	GPIO 位带操作	105
第 8 章	初识 STM32 标准库	67	第 13 章	启动文件	108
8.1	CMSIS 标准及库层次关系	67	13.1	启动文件简介	108
8.1.1	库目录、文件简介	68	13.2	查找 ARM 汇编指令	108
8.1.2	各库文件间的关系	74	13.3	启动文件代码讲解	109
8.2	使用帮助文档	75	第 14 章	RCC——使用 HSE/HSI 配置时钟	116
8.2.1	常用官方资料	75	14.1	RCC 主要作用——时钟部分	116
8.2.2	初识库函数	76			
第 9 章	新建工程——库函数版	78			
9.1	新建本地工程文件夹	78			
9.2	新建工程	79			
9.3	配置魔术棒选项卡	82			
9.4	下载器配置	85			
9.5	选择 Flash 大小	86			

14.2	RCC 框图剖析——时钟树	116	17.2	SysTick 寄存器介绍	145
14.2.1	系统时钟	116	17.3	SysTick 定时实验	146
14.2.2	其他时钟	121	17.3.1	硬件设计	146
14.3	配置系统时钟实验	122	17.3.2	软件设计	146
14.3.1	使用 HSE	122			
14.3.2	使用 HSI	122	第 18 章 通信的基本概念	152	
14.3.3	硬件设计	123	18.1	串行通信与并行通信	152
14.3.4	软件设计	123	18.2	全双工、半双工及单工通信	153
14.3.5	下载验证	129	18.3	同步通信与异步通信	153
			18.4	通信速率	154
第 15 章 STM32 中断应用概览	130		第 19 章 USART——串口通信	155	
15.1	异常类型	130	19.1	串口通信协议简介	155
15.2	NVIC 简介	131	19.1.1	物理层	155
15.2.1	NVIC 寄存器简介	131	19.1.2	协议层	158
15.2.2	NVIC 中断配置固件库	132	19.2	STM32 的 USART 简介	159
15.3	优先级的定义	132	19.3	USART 功能框图	160
15.3.1	优先级定义	132	19.4	USART 初始化结构体详解	166
15.3.2	优先级分组	133	19.5	USART1 接发通信实验	167
15.4	中断编程	134	19.5.1	硬件设计	168
			19.5.2	软件设计	168
			19.5.3	下载验证	173
第 16 章 EXTI——外部中断 / 事件控制器	136		19.6	USART1 指令控制 RGB 彩灯实验	174
16.1	EXTI 简介	136	19.6.1	硬件设计	174
16.2	EXTI 功能框图	136	19.6.2	软件设计	174
16.3	中断 / 事件线	138	19.6.3	下载验证	179
16.4	EXTI 初始化结构体详解	139			
16.5	外部中断控制实验	139	第 20 章 DMA	180	
16.5.1	硬件设计	140	20.1	DMA 简介	180
16.5.2	软件设计	140	20.2	DMA 功能框图	180
16.5.3	下载验证	144	20.3	DMA 数据配置	184
			20.4	DMA 初始化结构体详解	188
第 17 章 SysTick——系统定时器	145				
17.1	SysTick 简介	145			

20.5 DMA 存储器到存储器模式 实验	190	22.4.3 下载验证	234
20.5.1 硬件设计	190	第 23 章 SPI——读写串行 Flash	
20.5.2 软件设计	190	23.1 SPI 协议简介	235
20.5.3 下载验证	195	23.1.1 SPI 物理层	235
20.6 DMA 存储器到外设模式实验	195	23.1.2 协议层	236
20.6.1 硬件设计	195	23.2 STM32 的 SPI 特性及架构	238
20.6.2 软件设计	195	23.2.1 STM32 的 SPI 外设 简介	238
20.6.3 下载验证	199	23.2.2 STM32 的 SPI 架构 剖析	239
第 21 章 常用存储器介绍		23.2.3 通信过程	241
21.1 存储器种类	200	23.3 SPI 初始化结构体详解	242
21.2 RAM 存储器	200	23.4 SPI——读写串行 Flash 实验	243
21.2.1 DRAM	201	23.4.1 硬件设计	243
21.2.2 SRAM	202	23.4.2 软件设计	244
21.2.3 DRAM 与 SRAM 的 应用场合	202	23.4.3 下载验证	264
21.3 非易失性存储器	202	第 24 章 串行 Flash 文件系统 FatFs	
21.3.1 ROM 存储器	202	24.1 文件系统	265
21.3.2 Flash 存储器	203	24.2 FatFs 文件系统简介	266
第 22 章 I²C——读写 EEPROM		24.2.1 FatFs 的目录结构	266
22.1 I ² C 协议简介	205	24.2.2 FatFs 帮助文档	266
22.1.1 I ² C 物理层	205	24.2.3 FatFs 源码	267
22.1.2 协议层	206	24.3 FatFs 文件系统移植实验	268
22.2 STM32 的 I ² C 特性及架构	209	24.3.1 FatFs 程序结构图	268
22.2.1 STM32 的 I ² C 外设简介	209	24.3.2 硬件设计	269
22.2.2 STM32 的 I ² C 架构剖析	210	24.3.3 FatFs 移植步骤	269
22.2.3 通信过程	212	24.3.4 FatFs 底层设备驱动 函数	271
22.3 I ² C 初始化结构体详解	213	24.3.5 FatFs 功能配置	276
22.4 I ² C——读写 EEPROM 实验	215		
22.4.1 硬件设计	215		
22.4.2 软件设计	216		

24.3.6	FatFs 功能测试	277	26.1.1	液晶显示器	317
24.3.7	下载验证	280	26.1.2	LED 和 OLED 显示器	318
24.4	FatFs 功能使用实验	281	26.1.3	显示器的基本参数	319
24.4.1	硬件设计	281	26.2	液晶屏控制原理	319
24.4.2	软件设计	281	26.2.1	液晶面板的控制信号	320
24.4.3	下载验证	286	26.2.2	液晶数据传输时序	321
			26.2.3	显存	323
第 25 章	FMC——扩展外部		26.3	LTDC 液晶控制器简介	323
	SDRAM	287	26.3.1	图像数据混合	323
25.1	SDRAM 控制原理	287	26.3.2	LTDC 结构框图剖析	324
25.1.1	SDRAM 信号线	288	26.4	DMA2D 图形加速器简介	327
25.1.2	控制逻辑	289	26.5	LTDC 初始化结构体	329
25.1.3	地址控制	289	26.6	LTDC 层级初始化结构体	331
25.1.4	SDRAM 的存储阵列	289	26.7	DMA2D 初始化结构体	334
25.1.5	数据输入输出	289	26.8	LTDC/DMA2D——液晶显示	
25.1.6	SDRAM 的命令	290		实验	336
25.1.7	SDRAM 的初始化流程	295	26.8.1	硬件设计	336
25.1.8	SDRAM 的读写流程	296	26.8.2	软件设计	338
25.2	FMC 简介	297	26.8.3	下载验证	358
25.3	FMC 框图剖析	298			
25.4	FMC 的地址映射	300	第 27 章	LTDC——液晶显示	
25.5	SDRAM 时序结构体	302		中英文	359
25.6	SDRAM 初始化结构体	303	27.1	字符编码	359
25.7	SDRAM 命令结构体	304	27.1.1	ASCII 编码	359
25.8	FMC——扩展外部 SDRAM		27.1.2	中文编码	362
	实验	305	27.1.3	Unicode 字符集和编码	365
25.8.1	硬件设计	305	27.1.4	UTF-32	365
25.8.2	软件设计	305	27.1.5	UTF-16	365
25.8.3	下载验证	316	27.1.6	UTF-8	366
			27.1.7	BOM	367
第 26 章	LTDC/DMA2D——		27.2	字模简介	367
	液晶显示	317	27.2.1	字模的构成	368
26.1	显示器简介	317	27.2.2	字模显示原理	368

27.2.3	如何制作字模	370	29.2.1	ADC 功能	431
27.2.4	字模寻址公式	371	29.2.2	电压转换	437
27.2.5	存储字模文件	372	29.3	ADC 初始化结构体详解	437
27.3	LTDC——各种模式的液晶显示 字符实验	372	29.4	独立模式单通道采集实验	438
27.3.1	硬件设计	373	29.4.1	硬件设计	439
27.3.2	显示 ASCII 编码的 字符	373	29.4.2	软件设计	439
27.3.3	显示 GB2312 编码的 字符	382	29.4.3	下载验证	443
27.3.4	显示任意大小的字符	391	29.5	独立模式多通道采集实验	443
27.3.5	下载验证	398	29.5.1	硬件设计	443
			29.5.2	软件设计	443
			29.5.3	下载验证	449
			29.6	三重 ADC 交替模式采集实验	449
			29.6.1	硬件设计	449
			29.6.2	软件设计	450
			29.6.3	下载验证	455
第 28 章	电容触摸屏—— 触摸画板	399	第 30 章	TIM——基本定时器	456
28.1	触摸屏简介	399	30.1	TIM 简介	456
28.1.1	电阻触摸屏检测原理	399	30.2	基本定时器	456
28.1.2	电容触摸屏检测原理	401	30.3	基本定时器功能框图	458
28.2	电容触摸屏控制芯片	402	30.4	定时器初始化结构体详解	460
28.2.1	GT9157 芯片的引脚	403	30.5	基本定时器定时实验	461
28.2.2	上电时序与 I ² C 设备 地址	404	30.5.1	硬件设计	461
28.2.3	寄存器配置	404	30.5.2	软件设计	461
28.2.4	读取坐标信息	406	30.5.3	下载验证	464
28.3	电容触摸屏——触摸画板 实验	408	第 31 章	TIM——高级定时器	465
28.3.1	硬件设计	408	31.1	高级控制定时器	465
28.3.2	软件设计	409	31.2	高级控制定时器功能框图	466
28.3.3	下载验证	430	31.3	输入捕获应用	476
			31.3.1	测量脉宽或者频率	476
			31.3.2	PWM 输入模式	477
			31.4	输出比较应用	478
第 29 章	ADC——电压采集	431			
29.1	ADC 简介	431			
29.2	ADC 功能框图剖析	431			

31.5	定时器初始化结构体详解	480	33.7	SDIO 命令初始化结构体	522
31.6	PWM 互补输出实验	483	33.8	SDIO 数据初始化结构体	523
31.6.1	硬件设计	484	33.9	SD 卡读写测试实验	523
31.6.2	软件设计	484	33.9.1	硬件设计	524
31.6.3	下载验证	488	33.9.2	软件设计	524
31.7	PWM 输入捕获实验	489	33.9.3	下载验证	549
31.7.1	硬件设计	489			
31.7.2	软件设计	489			
31.7.3	下载验证	494			
第 32 章 TIM——电容按键检测					
32.1	电容按键原理	495			
32.2	电容按键检测实验	496			
32.2.1	硬件设计	497			
32.2.2	软件设计	497			
32.2.3	下载验证	504			
第 33 章 SDIO——SD 卡读写测试					
33.1	SDIO 简介	505			
33.2	SD 卡物理结构	506			
33.3	SDIO 总线	507			
33.3.1	总线拓扑	507			
33.3.2	总线协议	508			
33.3.3	命令	510			
33.3.4	响应	512			
33.4	SD 卡的操作模式及切换	514			
33.4.1	SD 卡的操作模式	514			
33.4.2	卡识别模式	514			
33.4.3	数据传输模式	516			
33.5	STM32 的 SDIO 功能框图	516			
33.6	SDIO 初始化结构体	521			
第 34 章 基于 SD 卡的 FatFs 文件系统					
34.1	FatFs 移植步骤	550			
34.2	FatFs 接口函数	552			
34.3	FatFs 功能测试	557			
第 35 章 I²S——音频播放与录音输入					
35.1	I ² S 简介	561			
35.1.1	数字音频技术	561			
35.1.2	I ² S 总线接口	562			
35.1.3	音频数据传输协议标准	562			
35.2	I ² S 功能框图	565			
35.3	WM8978 音频编译码器	567			
35.4	WAV 格式文件	569			
35.4.1	RIFF 文件规范	570			
35.4.2	WAV 文件	570			
35.4.3	WAV 文件实例分析	571			
35.5	I ² S 初始化结构体详解	571			
35.6	录音与回放实验	572			
35.6.1	硬件设计	573			
35.6.2	软件设计	573			
35.6.3	下载验证	601			
35.7	MP3 播放器	601			
35.7.1	MP3 文件结构	602			

35.7.2	MP3 解码库	605	37.1.1	CAN 物理层	680
35.7.3	Helix 解码库移植	606	37.1.2	协议层	684
35.7.4	MP3 播放器功能实现	606	37.2	STM32 的 CAN 外设简介	690
35.7.5	下载验证	614	37.3	CAN 初始化结构体	698
第 36 章 ETH——LwIP 以太网通信			37.4	CAN 发送及接收结构体	700
36.1	互联网模型	615	37.5	CAN 筛选器结构体	701
36.2	以太网	616	37.6	CAN——双机通信实验	703
36.2.1	PHY 层	616	37.6.1	硬件设计	703
36.2.2	MAC 子层	617	37.6.2	软件设计	704
36.3	TCP/IP 协议栈	618	37.6.3	下载验证	713
36.3.1	需要协议栈的原因	619	第 38 章 RS-485 通信实验		
36.3.2	各网络层的功能	619	38.1	RS-485 通信协议简介	714
36.4	以太网外设	620	38.2	RS-485——双机通信实验	715
36.4.1	SMI 接口	621	38.2.1	硬件设计	715
36.4.2	MII 和 RMII 接口	623	38.2.2	软件设计	716
36.4.3	MAC 数据包发送和接收	624	38.2.3	下载验证	723
36.4.4	MAC 过滤	626	第 39 章 电源管理——实现低功耗		
36.5	PHY: LAN8720A	626	39.1	STM32 的电源管理简介	724
36.6	LwIP: 轻型 TCP/IP 协议栈	629	39.1.1	电源监控器	724
36.7	ETH 初始化结构体详解	629	39.1.2	STM32 的电源系统	726
36.8	以太网通信实验: 无操作系统 LwIP 移植	635	39.1.3	STM32 的功耗模式	727
36.8.1	硬件设计	635	39.2	电源管理相关的库函数及命令	729
36.8.2	移植步骤	635	39.2.1	配置 PVD 监控功能	729
36.8.3	下载验证	661	39.2.2	WFI 与 WFE 命令	729
36.9	基于 μ COS-III 移植 LwIP 实验	663	39.2.3	进入停止模式	730
第 37 章 CAN——通信实验			39.2.4	进入待机模式	731
37.1	CAN 协议简介	680	39.3	PWR——睡眠模式实验	732
			39.3.1	硬件设计	732
			39.3.2	软件设计	732

39.3.3	下载验证	735	41.1.1	数字摄像头与模拟摄像头 的区别	766	
39.4	PWR——停止模式实验	735	41.1.2	CCD 与 CMOS 的区别	767	
39.4.1	硬件设计	735	41.2	OV5640 摄像头	767	
39.4.2	软件设计	735	41.2.1	OV5640 传感器简介	769	
39.4.3	下载验证	739	41.2.2	OV5640 引脚及功能 框图	769	
39.5	PWR——待机模式实验	739	41.2.3	SCCB 时序	771	
39.5.1	硬件设计	740	41.2.4	OV5640 的寄存器	772	
39.5.2	软件设计	740	41.2.5	像素数据输出时序	773	
39.5.3	下载验证	743	41.3	STM32 的 DCMI 接口简介	773	
39.6	PWR——PVD 电源监控实验	743	41.3.1	DCMI 整体框图	774	
39.6.1	硬件设计	743	41.3.2	DCMI 接口内部结构	775	
39.6.2	软件设计	745	41.3.3	同步方式	775	
39.6.3	下载验证	747	41.3.4	捕获模式及捕获率	776	
第 40 章 RTC——实时时钟			748	41.4	DCMI 初始化结构体	776
40.1	RTC 简介	748	41.5	DCMI——OV5640 摄像头实验	777	
40.2	RTC 功能框图解析	748	41.5.1	硬件设计	777	
40.3	RTC 初始化结构体讲解	751	41.5.2	软件设计	779	
40.4	RTC 时间结构体讲解	752	41.5.3	下载验证	797	
40.5	RTC 日期结构体讲解	753	第 42 章 MDK 的编译过程及 文件类型全解			798
40.6	RTC 闹钟结构体讲解	753	42.1	编译过程	798	
40.7	RTC—日历实验	754	42.1.1	编译过程简介	798	
40.7.1	硬件设计	754	42.1.2	具体工程中的编译 过程	799	
40.7.2	软件设计	754	42.2	程序的组成、存储与运行	800	
40.7.3	下载验证	760	42.2.1	CODE、RO、RW、 ZI Data 域及堆栈空间	800	
40.8	RTC—闹钟实验	760	42.2.2	程序的存储与运行	801	
40.8.1	硬件设计	760	42.3	编译工具链	802	
40.8.2	软件设计	760	42.3.1	设置环境变量	803	
40.8.3	下载验证	765				
第 41 章 DCMI——OV5640 摄像头			766			
41.1	摄像头简介	766				

42.3.2	armcc、armasm 及 armlink	804	43.4	实验：在内部 SRAM 中调试 代码	869
42.3.3	armar、fromelf 及用户 指令	807	43.4.1	硬件设计	869
42.4	MDK 工程的文件类型	808	43.4.2	软件设计	870
42.4.1	uvprojx、uvoptx、uvguix 及 ini 工程文件	809	43.4.3	下载验证	877
42.4.2	源文件	811	第 44 章 读写内部 Flash	878	
42.4.3	Output 目录下生成的 文件	811	44.1	STM32 的内部 Flash 简介	878
42.4.4	Listing 目录下的文件	831	44.2	对内部 Flash 的写入过程	881
42.4.5	sct 分散加载文件的格式 与应用	837	44.3	查看工程的空间分布	882
42.5	实验：自动分配变量到外部 SDRAM 空间	846	44.4	操作内部 Flash 的库函数	884
42.5.1	硬件设计	846	44.5	实验：读写内部 Flash	887
42.5.2	软件设计	847	44.5.1	硬件设计	887
42.5.3	下载验证	853	44.5.2	软件设计	887
42.6	实验：优先使用内部 SRAM 并把 堆区分配到 SDRAM 空间	853	44.5.3	下载验证	893
42.6.1	硬件设计	854	第 45 章 设置 Flash 的读写 保护及解除	894	
42.6.2	软件设计	854	45.1	选项字节与读写保护	894
42.6.3	下载验证	864	45.1.1	选项字节的内容	894
第 43 章 在 SRAM 中调试 代码	865		45.1.2	RDP 读保护级别	896
43.1	在 RAM 中调试代码	865	45.1.3	PCROP 代码读出保护	898
43.2	STM32 的启动方式	865	45.2	修改选项字节的过程	898
43.3	内部 Flash 的启动过程	867	45.3	操作选项字节的库函数	899
			45.4	实验：设置读写保护及解除	901
			45.4.1	硬件设计	902
			45.4.2	软件设计	902
			45.4.3	下载验证	908

用 fromelf 文件反汇编代码，不过它不能还原出 C 代码。由于 lib 文件的主要目的是为了保护 C 源代码，也算是达到了它的要求。

6. htm 静态调用图文件

在 Output 目录下，有一个以工程文件命名的后缀为 *.bulid_log.htm 及 *.htm 文件，如“多彩流水灯 .bulid_log.htm”及“多彩流水灯 .htm”，它们都可以使用浏览器打开。其中 *.build_log.htm 是工程的构建过程日志，而 *.htm 是链接器生成的静态调用图文件。

在静态调用图文件中包含了整个工程各种函数之间互相调用的关系图，而且它还给出了静态占用最深的栈空间数量以及它对应的调用关系链。

如图 42-43 所示是“多彩流水灯 .htm”文件顶部的说明。

Static Call Graph for image .. \.. \Output\多彩流水灯.axf
#<CALLGRAPH># ARM Linker, 5060020: Last Updated: Tue Mar 15 16:55:24 2016
Maximum Stack Usage = 56 bytes + Unknown(Cycles, Untraceable Function Pointers)
Call chain for Maximum Stack Depth:
main => LED_GPIO_Config => GPIO_Init

图 42-43 静态占用最深的栈空间说明

该文件说明了本工程的静态栈空间最大占用 56 字节 (Maximum Stack Usage:56bytes)，这个占用最深的静态调用为“main->LED_GPIO_Config->GPIO_Init”。注意这里给出的空间只是静态的栈使用统计，链接器无法统计动态使用情况，例如链接器无法知道递归函数的递归深度。在本文件的后面还可查询到其他函数的调用情况及其他细节。

利用这些信息，我们可以大致了解工程中应该分配多少空间给栈，有空间余量的情况下，一般会设置比这个静态最深栈使用量大一倍，在 STM32 中可修改启动文件改变堆栈的大小；如果空间不足，可从该文件中了解到调用深度的信息，然后优化该代码。

注意：查看了各个工程的静态调用图文件统计后，我们发现本书提供的一些比较大规模的工程例子，静态栈调用最大深度都已超出 STM32 启动文件默认的栈空间大小 0x00000400，即 1024 字节，但在当时的调试过程中却没有发现错误，因此我们也没有修改栈的默认大小（有一些工程调试时已发现问题，它们的栈空间就已经被我们改大了）。虽然这些工程实际运行并没有错误，但这可能只是因为它使用的栈溢出 RAM 空间恰好没被程序其他部分修改而已。所以，建议在实际的大型工程应用中（特别是使用了各种外部库时，如 Lwip/emWin/Fatfs 等），要查看本静态调用图文件，了解程序的栈使用情况，给程序分配合适的栈空间。

42.4.4 Listing 目录下的文件

在 Listing 目录下包含了 *.map 及 *.lst 文件，它们都是文本格式的，可使用 Windows 的