第3章



# MDK 软件与工程模板创建

本章要点

- ▶ STM32 固件库概述
- ▶ STM32 固件库下载
- ▶ STM32 固件库目录结构
- ▶ Keil MDK 软件操作方法
- ▶ Keil MDK 工程模板的创建
- ▶ Keil MDK 软件模拟仿真调试

"工欲善其事,必先利其器",无论是基于寄存器方式还是基于库函数方式开发 STM32 应用程序,首先必须选择一个熟悉、完善的开发平台,建立方便、合理的程序工程模板。对于 51 单片机开发者来说,Keil-C 是再熟悉不过的。而 Keil 公司针对 32 位 ARM 嵌入式系统 推出的 Keil MDK 开发平台功能强大,基本操作又和 Keil-C 保持兼容,是 32 位嵌入式单片 机开发的首选。建立工程模板的核心内容包含两个方面:一是必须包含的文件;二是这些 文件对应的路径。

# 3.1 STM32 固件库认知



建立工程模板需要从指定路径找到必要文件,要想很好地完成这一任务, 我们需要首先认识 STM32 固件库。

### 3.1.1 STM32 固件库概述

意法半导体公司提供的 STM32F10x 标准外设库是基于 STM32F1 系列微控制器的固件库进行 STM32F103 开发的一把利器。可以像在标准 C 语言编程中调用 printf()一样,在 STM32F10x 的开发中调用标准外设库的库函数,进行应用开发。相比传统的直接读写寄存器方式,STM32F10x 标准外设库不仅明显降低了开发门槛和难度,缩短了开发周期,进

而降低开发成本,而且提高了程序的可读性和可维护性,给 STM32F103 开发带来了极大的便利。毫无疑问,STM32F10x 标准外设库是用户学习和开发 STM32F103 微控制器的第一选择。

STM32 固件库是根据 CMSIS 标准(Cortex Microcontroller Software Interface Standard, ARM Cortex 微控制器软件接口标准)而设计的。CMSIS 标准由 ARM 和芯片生产商共同提出,让不同的芯片公司生产的 Cortex M3 微控制器能在软件上基本兼容。

STM32F10x的固件库是一个或一个以上的完整的软件包(称为固件包),包括所有的标准外设的设备驱动程序,其本质是一个固件函数包(库),它由程序、数据结构和各种宏组成,包括了微控制器所有外设的性能特征。该函数库还包括每一个外设的驱动描述和应用实例,为开发者访问底层硬件提供了一个中间 API(APPlication Programming Interface,应用编程接口)。通过使用固件函数库,无须深入掌握底层硬件细节,开发者就可以轻松应用每一个外设。每个外设驱动都由一组函数组成,这组函数覆盖了该外设的所有功能。每个器件的开发都由一个通用 API 驱动,API 对该驱动程序的结构、函数和参数名称都进行了标准化。

### 3.1.2 STM32 固件库下载

意法半导体公司 2007 年 10 月发布了 V1.0 版本的固件库,2008 年 6 月发布了 V2.0 版的固件库。V3.0 以后的版本相对之前的版本改动较大,本书使用目前最为通用的 V3.5 版本,该版本固件库支持所有的 STM32F10x 系列。具体下载方法如下:

第一步: 输入 www.st.com 网址,打开意法半导体官方网站,在首页搜索栏输入 stm32f10x,其操作界面如图 3-1 所示。



#### 图 3-1 搜索资料操作界面

在图 3-1 中,单击右侧搜索图标开始搜索,结果如图 3-2 所示,其中 STM32F10x standard peripheral library 记录即为 STM32F1 的标准外设库。

Part Number	Status	Туре	Category	Description
STSW- STM32023	Active	Embedded Software	MCUs Embedded Software	How to inligrate from the STM32F10xxx firmware library V2.0.3 to the STM32F10xxx standard peripheral library V3.0.0 (AN2953)
STSW- STMJ2011	Active	Embedded Software	MCUs Embedded Software	Smartcard interface with the STM32F10x and STM32L1xx microcontrollers (AN2598)
STSW- STMJ2024	Active	Embedded Software	MCUs Embedded Software	Getting started with uClinux for STM32F10x high-density devices (AN3012)
STSW- STM32027	Active	Embedded Software	MCUs Embedded Software	Communication peripheral FIFO emulation with DMA and DMA timeout in STM32F10x microcontrollers (AN3109)
STSW- STM32054	Active	Embedded Software	MCUs Embedded Software	STM32F10x standard peripheral library
STSW- STM32008	Active	Embedded Software	MCUs Embedded Software	STM32F10xxx in-application programming using the USART (AN2557)

图 3-2 搜索结果页面

打开链接即可进入固件库下载页面,操作结果如图 3-3 所示,同时也可以看到,该固件 库版本为 3.5.0,该版本最为成熟和通用。

GET	SOFTWARE	

	2	La la construcción de		Landour Service	-
Part Number	General Description	Software Version	Supplier	Marketing Status	Download
STSW-STM32054	STM32F10x standard peripheral library	350	ST	Active	Gel Settiva)

图 3-3 固件库下载页面

单击图 3-3 右边的 Get Software 按钮,登录并确认著作权之后,即可将该固件库下载到的本机。

需要说明的是: 意法半导体官网资料需要登录才可以下载,如果没有账号还需要注册, 当然读者也可以直接在清华大学出版社网站下载本书的教学素材,里面包含 STM32 内核 固件库。

### 3.1.3 STM32 固件库目录结构

下载 STM32F10x 标准外设库并解压后,其目录结构如图 3-4 所示,由图可知固件库包 含四个文件夹和两个文件。





两个文件中,stm32f10x\_stdperiph\_lib\_um.chm 为已经编译的帮助系统,也就是该固件库的使用手册和应用举例,该文件很重要;而另一个文件 Release\_Notes.html 是固件库版本更新说明,可以将其忽略。

四个文件夹中,\_htmresc 文件夹是意法半导体公司的 LOGO 图标等文件,也可以将其 忽略,重要的三个文件夹是 Libraries、Project 和 Utilities,下面对其进行分别介绍。

#### 1. Libraries 文件夹

Libraries 文件夹用于存放 STM32F10x 开发要用到的各种库函数和启动文件,其下包括 CMSIS 和 STM32F10x\_StdPeriph\_Driver 两个子文件夹,如图 3-5 所示。



图 3-5 Libraries 文件夹

#### 1) CMSIS 子文件夹

CMSIS 子文件夹是 STM32F10x 的内核库文件夹,其核心是 CM3 子文件夹,其余可以 忽略。在 CM3 子文件夹下有 CoreSupport 和 DeviceSupport 两个文件夹,如图 3-6 所示。

40 🚽 ARM Cortex-M3嵌入式原理及应用——基于STM32F103微控制器



图 3-6 Libraries\CMSIS\CM3 目录

(1) CoreSupport 文件夹。该文件夹为 Cortex-M3 核内外设函数文件夹, Cortex-M3 内核通 用源文件 core\_cm3. c 和 Cortex-M3 内核通用头文件 core\_cm3. h 即在此目录下, 如图 3-7 所示。

		CoreSupport	85/10	-	• *;	檀黄 Ca	ore \$
组织 •		共享 *	刻录	*		• 🗆	0
core_cm3	core_cm3.						

图 3-7 CMSIS\CM3\CoreSupport 目录

上述文件位于 CMSIS 核心层的核内外设访问层,由 ARM 公司提供,包含用于访问内 核寄存器的名称、地址定义等内容。

(2) DeviceSupport 文件夹。该文件夹为设备外设支持函数文件夹,STM32F0x 头文件 stm32f10x.h 和系统初始化文件 system\_stm32f10x.c 即位于此目录下的 ST\STM32F10x 文件夹中,如图 3-8 所示。



图 3-8 DeviceSupport\ST\STM32F10x 目录

除了头文件和初始化文件,STM32F10x系列微控制器的启动代码文件,也位于此目录 下的 ST\STM32F10x\startup\arm 文件夹中,如图 3-9 所示。例如,本书配套开发板使用 的 STM32F103ZET6 微控制器属于 STM32F103 的大容量产品,因此,它对应的启动代码文 件为 startup\_stm32f10x\_hd.s。

2(件(上) 編編		工具(1) 梯間		-19	ET 6
組织 • 1	包含到库中 ▼	共学 •	刻家 »	•	
		1	1	1	1
startup_st.	startup_st	startup_st	startup_st	startup_st	startup_s
l.s	d.s	d_vl.s	d.s	d_vl.s	md.s
1	9				
startup st	startup st				
m32f10x_	m32f10x_x				

图 3-9 STM32F10x 启动代码文件目录

上述文件位于 CMSIS 核心层的设备外设访问层,由意法半导体公司提供,包含片上核 外设寄存器名称、地址定义、中断向量定义等。

2) STM32F10x\_StdPeriph\_Driver 文件夹

STM32F10x\_StdPeriph\_Driver 子文件夹为 STM32Fl0x 标准外设驱动库函数目录,包括了所有 STM32F10x 微控制器的外设驱动,如 GPIO、TIMER、SysTick、ADC、DMA、USART、SPI 和 I2C 等。STM32F10x 的每个外设驱动对应一个源代码文件 stm32f10x\_ppp.c 和一个头文件 stm32f10x\_ppp.h。相应地,STM32F10x\_StdPeriph\_Driver 文件夹下也有两个子目录: src 和 inc,如图 3-10 所示。特别地,除了以上 STM32F10x 片上外设的驱动以外,Cortex-M3 内核中 NVIC 的驱动(misc. c 和 misc. h)也在该文件夹中。



图 3-10 Libraries\STM32F10x\_StdPeriph\_Driver 目录

### 42 - ARM Cortex-M3嵌入式原理及应用——基于STM32F103微控制器

(1) src 子目录: src 是 source 的缩写,该子目录下存放意法半导体公司为 STM32F10x 每个外设而编写的库函数源代码文件,如图 3-11 所示。



图 3-11 STM32F10x\_StdPeriph\_Driver\src 目录

(2) inc 子目录: inc 是 include 的缩写。该子目录下存放 STM32F10x 每个外设库函数 的头文件,如图 3-12 所示。

文件(1)编辑(1)章看(1)	工具田 帮助田	
组织• ▲ 打开•	共享 ▼ 割景 >>	* - CI 0
🗟 misc.h	🖬 stm32f10x_exti.h	📾 stm32f10x_sdio.h
stm32f10x_adc.h	stm32f10x_flash.h	stm32f10x_spi.h
stm32f10x_bkp.h	stm32f10x_fsmc.h	stm32f10x_tim.h
stm32f10x_can.h	stm32f10x_gpio.h	stm32f10x_usart.h
stm32f10x_cec.h	stm32f10x_i2c.h	stm32f10x_wwdg.h
stm32f10x_crc.h	stm32f10x_iwdg.h	
stm32f10x_dac.h	stm32f10x_pwr.h	
stm32f10x_dbgmcu.h	stm32f10x_rcc.h	
stm32f10x_dma.h	stm32f10x_rtc.h	

图 3-12 STM32F10x\_StdPeriph\_Driver\inc 目录

#### 2. Project 文件夹

Project 文件夹对应 STM32F10x 标准外设库体系架构中的用户层,用来存放意法半导体公司官方提供的 STM32F10x 工程模板和外设驱动示例,包括 STM32F10x\_StdPeriph\_ Template 和 STM32F10x\_StdPeriph\_Examples 两个子文件夹,目录结构如图 3-13 所示。



图 3-13 STM32F10x\_StdPeriph\_Lib\_V3.5.0\Project 文件夹目录

1) STM32F10x\_StdPeriph\_Template 子文件夹

STM32F10x\_StdPeriph\_Template 子文件夹,是意法半导体公司提供的 STM32F10x 工 程模板目录,包括了 5 个开发工具相关子目录和 5 个用户应用相关文件,目录结构如图 3-14 所示。



图 3-14 Project\STM32F10x\_StdPeriph\_Template 文件夹目录

(1) 开发工具相关子目录:根据使用的开发工具的不同,分为 MDK-ARM、EWARM、 HiTOP、RIDE 和 TrueSTUDIO 这 5 个子目录,每个子目录分别存放对应开发工具下 STM32F10x 的工程文件。

(2) 用户应用相关文件:包括 main.c、stm32f10x\_it.c、stm32f10x\_it.h、stm32f10x\_conf.h 和 system\_stm32f10x.c这5个文件。无论使用5种开发工具中的哪一个构建STM32F10x 工程,用户的具体应用都只与这5个文件有关。这样,在同一型号的微控制器上开发不同应 用时,无须修改相关开发工具目录下的工程文件,只需要用新编写的应用程序文件替换这 5个文件即可。

### 44 - ARM Cortex-M3嵌入式原理及应用——基于STM32F103微控制器

2) STM32F10x\_StdPeriph\_Examples 子文件夹

STM32F10x\_StdPeriph\_Examples 子文件夹,是意法半导体公司提供的 STM32F10x 外设驱动示例目录。该目录包含许多以 STM32F10x 外设命名的子目录,囊括了 STM32F10x 所有外设,其目录结构如图 3-15 所示。



图 3-15 Project\STM32F10x\_StdPeriph\_Examples 目录

每个外设子目录下又包含多个具体驱动示例目录,而每个示例目录下又包含 5 个用户应用相关文件。意法半导体公司官方的外设驱动示例,不仅是了解和验证 STM32 外设功能的重要途径,而且给 STM32F10x 相关外设开发提供了有益的参考。

#### 3. Utilities 文件夹

Utilities 文件夹用于存放意法半导体公司官方评估板的 BSP(Board Support Package, 板级支持包)和额外的第三方固件。初始情况下,该文件夹下仅包含意法半导体公司各款官 方评估板的板级驱动程序(即 STM32\_EVAL 子文件夹),目录结构如图 3-16 所示。



图 3-16 Utilities 文件夹目录

用户在实际开发时,可以根据应用需求,在 Utilities 文件夹下增删内容,如删除仅支持 意法半导体公司官方评估板的板驱动包,添加由意法半导体公司及其第三方合作伙伴提供 的固件协议,包括各种嵌入式操作系统、文件系统、图形接口等,当然也可以不使用其参考模板,自行独立创建工程模板,本书采用的是后者。

# 3.2 工程模板创建

本节介绍 Keil MDK-ARM 软件的使用,并创建一个自己的 MDK 工程模板,该工程模板是后续学习的基础。

# 3.2.1 Keil MDK-ARM 软件简介

Keil MDK-ARM 是适用于基于 Cortex-M、Cortex-R4、ARM7 和 ARM9 处理器的设备 的完整软件开发环境。Keil MDK-ARM 是专为微控制器应用程序开发而设计的,它易于学 习和使用,同时具有强大的功能,适用于多数要求苛刻的嵌入式应用程序开发。Keil MDK-ARM 是目前最流行的嵌入式开发工具,集成了业内最领先的技术,包括 µVision4 集成开发 环境与 ARM 编译器,具有自动配置启动代码、集成 Flash 烧写模块、强大的 Simulation 设 备模拟、性能分析等功能。

目前 Keil MDK-ARM 的最新版本是 4.74。4.0 以上版本的 Keil MDK-ARM 的 IDE 界面有了很大的改变,并且支持 Cortex-M 内核的处理器。Keil MDK-ARM 4.74 界面简 洁、美观,实用性更强,对于使用过 Keil 的读者来说,更容易上手。Keil MDK-ARM 软件主 要特点如下:

(1) 完美支持 Cortex-M、Cortex-R4、ARM7 和 ARM9 系列器件。

- (2) 行业领先的 ARM C/C++编译工具链。
- (3) 确定的 Keil RTX,小封装实时操作系统(带源码)。
- (4) µVision4 IDE 集成开发环境,调试器和仿真环境。
- (5) TCP/IP 网络套件提供多种的协议和各种应用。
- (6) 提供带标准驱动类的 USB 设备和 USB 主机栈。
- (7)为带图形用户接口的嵌入式系统提供了完善的 GUI 库支持。
- (8) ULINKpro 可实时分析运行中的应用程序,且能记录 Cortex-M 指令的每一次执行。
- (9)关于程序运行的完整代码覆盖率信息。
- (10)执行分析工具和性能分析器可使程序得到最优化。
- (11) 大量的项目例程有助于快速熟悉 Keil MDK-ARM 强大的内置特征。
- (12) 符合 CMSIS (Cortex 微控制器软件接口标准)。

本书选择 Keil MDK-ARM 4.74 版本的开发工具作为学习 STM32 的软件。当然,读者 也可以到 Keil 公司网站下载或查看最新的 Keil MDK-ARM 软件版本。

### 3.2.2 工程模板的创建

工程模板是我们后续所有项目的基础,正确、合理的工程模板不仅使用起来得心应手,

### 46 - ARM Cortex-M3嵌入式原理及应用——基于STM32F103微控制器

而且有利于结构化程序设计。工程模板除了必须包含的框架体系结构,也有一部分个性化的因素,所以每个人创建的工程模板可能是不同的。

#### 1. 创建工程模板素材

创建工程模板素材主要是内核固件库 3.5 版,另外还有两个重要的预定义命令,也以文 本文档的形式给出来,相关素材均可以在清华大学出版社网站下载。

为了便于叙述,将原固件库的文件夹名 STM32F10x\_StdPeriph\_Lib\_V3.5.0 更改为简 短一些的 F10x\_Lib\_V3.5,以便在书中给出具体的文件路径。

#### 2. 工程模板创建步骤

第一步:创建或复制文件夹。

(1) 在桌面创建"工程模板"文件夹。

(2) 复制固件库中的 Libraries 文件夹到工程模板文件夹。

(3) 创建 Output 文件夹,用于存放输出文件。

(4) 创建 Startup 文件夹,用于存放启动文件,并复制 startup\_stm32f10x\_hd.s 文件到 该文件夹中,此文件为大容量芯片的启动文件,文件路径为:F10x\_Lib\_V3.5\Libraries\ CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm。操作结果如图 3-17 所示。

〇〇。),I程模板) startup		▼ +→ 標素 startup		-		×
文件(图编辑(图) 查看(图) 工具(图)	<b>转助(<u>H</u>)</b>	and other only	-	-	-	
组织 ▼ 包含到库中 ▼ 共享 ▼	刻录 新建文件夹			•		0
☆ 收藏夹	4					
🔊 下载						
1 桌面	startup_st					
3 最近访问的位置	m32f10x_h d.s					
詞库						
🖬 视频 🗉						
■ 图片						
■ 文档						
♪ 音乐						
1 计算机						
🏭 系统 (C:)						
👝 个人 (D:)						
📾 教学 (E:)						
💼 学习 (F:)						
1个对象						

#### 图 3-17 Startup 文件夹

(5) 创建 User 文件夹,并复制 main.c、stm32f10x\_conf.h、stm32f10x\_it.c、stm32f10x\_ it.h文件到该文件夹中。上述文件路径为:F10x\_Lib\_V3.5\Project\STM32F10x\_ StdPeriph\_Template,其操作结果如图 3-18 所示。

	I I I ISE GSET		
身助( <u>H</u> )		210 E	
刻录 新建又件夹			U W
main.c stm32f10x stm32f	10x stm32f10x		
_cont.h _it.c	_it.h		
	解助任) 鼓录 新建文件夹 main.c stm32f10x stm32f 	解助(1) 教授 新建文件夹 main.c stm32f10x stm32f10x stm32f10xjt.cjt.h	解助山

图 3-18 User 文件夹

(6) 创建 APP 文件夹,用于存放用户编写的外设驱动程序。 经过上述步骤创建的工程模板文件夹如图 3-19 所示。

2件(F) 编辑(E) 查看(V)	工具(T) 報	助(H)		-		And a second second second		-
组织 • 包含到库中 •	共享▼	刻录	新建文件夹				10 ×	
☆ 收藏夹 ▲ 下戦	Î	1	I	T	T	T		
桌面 30 最近访问的位置		APP	Libraries	output	startup	user		
二 库	=							
■ 100000								
全有								
<b>④'</b> 曰 <b>小</b>								
▶ 计算机								
🏭 系统 (C:)								
□ 个人 (D:)								
□ • 秋子 (C:)								

图 3-19 工程模板文件夹

第二步:建工程模板文件,建立文档分组。

(1) 在开始/程序或桌面快捷方式中启动 Keil µVision4 软件,其界面如图 3-20 所示。

▼ F:\STM32\STM32学习\电震培训STM32\实验项目\17 Infrared+PC\工程模板\Template.uvproj-	pVision4
Eile Edit View Project Flash Debug Peripherals Iools SVCS Window Help	the first state of the second state
DOID IN THE PROPERTY OF	SIPOB 🔄 🗟 🛹 🛛
· · · · · · · · · · · · · · · · · · ·	
Project D D	
E Target 1 *	
<ul> <li>user</li> <li>fmain.c</li> <li></li></ul>	
Build Output	a 🗈
	*
*	14
	J-LINK / J-TRACE Cortex

图 3-20 Keil µVision4 软件界面

(2) 依次单击菜单栏 Project→new µvision Project 命令,弹出如图 3-21 所示的窗口,表示新建一个工程文件,并需要选择保存路径。此时将保存路径选择为我们在桌面的创建的 工程模板文件夹,文件名为"工程模板"。

(3) 单击如图 3-21 所示的"保存"按钮,弹出选择芯片的对话框,由于开发板使用的是 STM32F103ZET6 芯片,故选择 CPU 为 STM32F103ZE。

(4) 单击图 3-22 中的 OK 按钮,弹出询问对话框,由于后面还需要专门添加此文件,故选择"否",其操作界面如图 3-23 所示。

(5) 建立分组并添加文件。

依次单击 Project→Manage→components Environment 或直接单击工具栏 ▲图标打开 Manage Project Items 对话框,其操作界面如图 3-24 所示。

在此对话框中的 Groups 项中, 先删除原来的 Source Group 1, 然后依添加 User、 Cmsis、Startup、ST Driver 和 APP 分组, 并为每个分组添加相应的源文件。

(1) User: main.c,stm32f10x\_it.c.

(2) Cmsis: core\_cm3.c,system\_stm32f10x.c.

(3) Startup: startup\_stm32f10x\_hd. s.

	程模板 →		<ul> <li> <li>     ◆</li></li></ul>	續板
组织 • 新建文	件夹			B • 0
☆ 收藏夹 下载 ■ 東面	APP Libra	ries output	startup	) user
<ul> <li>2 最近访问的位</li> <li>2 库</li> <li>2 限频</li> <li>2 図片</li> <li>2 文档</li> <li>3 音乐</li> </ul>	Ξ. F			
<ul> <li>計算机</li> <li>系统 (C:)</li> <li>个人 (D:)</li> </ul>				
文件名(N):	工程模板			
保存类型(工):	Project Files (*.uvproj)			
● 隠藏文件夹			保存(S)	取消

#### 图 3-21 保存新建工程文件

Vendor Device Toolset	STMicroelectronics STM32F103ZE ARM			
000000000	STM32F103VD STM32F103VE STM32F103VF STM32F103VG STM32F103ZC STM32F103ZD STM32F103ZE STM32F103ZF STM32F103ZF	1	ARM 32-bit Cottex-M3 Microcontroller, 72MHz, 5 Headble Static Memory Controller for SRAM, PSF PLL, Embedded Internal RC 8MHz and 32kHz, Nested Interrupt Controller, Power Saving Moder 4 Synch, There 3 SP/125, 212,5 USART US Synt Rk: There: 3 SP/125, 212,5 USART US CAN 2.0B Active, 3 12-bit 16-ch A/D Converter, SDIO, Fast I/O Ports	12kB Rasin, 54kB SRA AM, NOR and NAND F Real-Time Clock, s, JTAG and SWD, It Compare and PWM, TGbit Watchdog Time 8 20 Full Speed Intela 2 12bit D/A Converter
000	STM32F105R8 STM32F105RB STM32F105RC	-	*(	

图 3-22 CPU 芯片选择



图 3-23 复制启动文件选项

main.c stm32f10x_∎.c
Add Files

图 3-24 添加工程分组对话框

(4) ST Driver: stm32f10x\_gpio.c, stm32f10x\_rcc. c.

(5) APP: 此分组下面还没有文件,由用户编写。

建立分组和添加文件操作完成之后,Keil软件界面如图 3-25 所示,在左边的工程浏览 窗口,可以看到刚刚创建的分组和相应的文件信息。

第三步:设置输出文件夹,添加预编译变量,包含头文件路径。

(1) 依次单击菜单 Project→Options for target 命令或直接单击工具栏 ☎ 图标,可以打 开如图 3-26 所示的对话框。

(2) 在 Output 选项卡中勾选 Create HEX File,并选择输出文件夹为工程模板目录下的 output 文件夹,其操作界面如图 3-27 所示。

(3) 在 Listing 选项卡中单击 Select Folder for Listings,并选择输出文件夹为工程模板 目录下的 output 文件夹,操作界面如图 3-28 所示。

Eile Edit View Project Flash Debug Peripherals Iools SVCS M	/indow <u>H</u> elp
12338 143	E IF HE HE GIPOB
🖉 🕮 🥪 🕫 🛱 Target 1 💽 🔊 🛔 🕾 🚸	
Project 🛛 🗘 🖾	
<pre>rargef 1</pre>	
Build Output	7 🖸
	× ×
4	i i

图 3-25 建立分组和添加文件完成

I Microelectror	lics 5	TM32F1032	E	_	- Code C	Generation			
			Xtal (MHz): 8	10					
perating syste	m: [	None		*	FU	se Cross-I	Module Optimiza	tion	
ystem-Viewer	File (.S	Sfr):			FU	se MicroL	1 81	Big Ericker	
STM32F103/00	sfi								
Use Custor	n SVD	File		-					
Read/Only M	emon	Areas			- Read/	Write Men	nory Areas		
default off-cl	qir	Start	Size	Startup	default	off-chip	Start	Size	Nolnit
T ROM	n: [		1	- C	Г	RAM1:		1	Г
T ROM	12: [				Г	RAM2		-	
T ROM	13: 1		<u> </u>		F	RAM3:	-	1	- F
00-0	hip		ā.			on-chip	r.	i.	
IF IRON	11: 0	k8000000	0x80000	•	P	IRAM1:	0x20000000	0x10000	Г
	12: [		1	- r	F	IRAM2		1	- r
	-	-							

图 3-26 Options for Target 'Target 1'对话框

Options for Target Target 1 wice   Target Output   Listing   User	C/C++   Asm   L	inker   Debug:   V	tilities
Select Folder for Objects	Name of Executable	工程模板	
<ul> <li>Create Executable: \Output\工程模板</li> <li>▽ Debug Information</li> <li>▽ Create HEX File 1</li> <li>▽ Browse Information</li> <li>C Create Library: \Output\工程模板 ib</li> </ul>			Create Batch File
OK	Cancel	Defaults	Help

图 3-27 Output 选项卡设置

rice   Target   Dutput Listing User   C/C++   Assa   Linker   Dabug   Utilities   Select Folder for Listings. 2 Page Width 79 - Page Length: 66 - ✓ Assembler Listing: Output * Int ✓ Assembler Listing: Output * Int ✓ Browse for Folder ✓ Cor Folder / Dutput 3 + E C* 图* < 新作 #改日期 美型 大小 ▲ I程模板:build_log 2019/1/27 360 Chrom 0 KB ✓ Linke ✓ Fath:	ons for Targe	t /Target 1	
Select Folder for Listings       2       Page Width: 79 - Page Length: 66 -          Image: Assembler Listing: Output/*lat       Image: Assembler Listing: Output/*lat         Image: Folder       Image: Assembl	e   Target   Ou	tput Listing User   C/C++   Aum   Linker   Debug   1	Vtilities
▼ Assembler Listing: \Output\*lat       ▼ Assembler Listing: \Output\*lat       ▼ Browse for Folder       ▼ CCr       Folder       ● Output       3       ● E 音 目       ● State       ● TElevee       ● State       ● Televee       ● Televee <t< td=""><td>Select Folder</td><td>or Listings 2 Page Width: 79 Page Le</td><td>ngth:  66 🛨</td></t<>	Select Folder	or Listings 2 Page Width: 79 Page Le	ngth:  66 🛨
Image: Second seco	Assembler List	ng: \Output\*list	
F CCc Folder: Dutput 3 ★ 作 10 倍 回▼ F CPr 名称 常改日期 美型 大小 ★ I程標板.build_log 2019/1/27 360 Chrom 0 KB F Unk F Ath:	M Bro	wse for Folder	×
下 C Pr 名称 修改日期 美型 大小     ▲ 工程模板:build_log 2019/1/27 360 Chrom 0 KB     □	C Cc Folde	r: 0utput 3 + E 🕾	
■ 工程模板.build_log 2019/1/27 360 Chrom 0 KB	CPr 名称	*	
F Link   F path:   OF	(A.)	程模板.build_log 2019/1/27 360 Chrom	0 КВ
Fath: 000	Links		
F Fath: 000 1	F		
OK	F Path:	Ť.	
			OK
OK Cancel Defaults Hel		OK. Cancel Defaults	Help

图 3-28 Listing 输出文件目录

(4) 在 C/C++选项卡中,在 Define 区域添加两个重要的预编译命令: USE\_STDPERIPH\_ DRIVER。STM32F10X\_HD,这两个预编译命令存放在素材文件夹中的"两个重要的预编译 指令.txt"文件中,操作界面如图 3-29 所示。

need an feed on the f	Listing   User	L/C++ Asm   Linker   Debug	[ Utilities ]
Define: USE_STDPER	RIPH_DRIVER.ST	M32F10X_HD	
Undefine:			
Language / Code General Dotimization: Level D (-OC C Optimize for Time Split Load and Store M C One ELF Section per F Include Paths Misc Controls	ion ) <u>*</u> ultiple unction	Sinct ANSIC     Enum Container always int     Enum Container always int     Bain Char is Signed     Read-Only Position Independent     Bead-Write Position Independent	Warnings Canspecifieds Thereb Mode No Auto Includes C 99 Mode
Compiler control string	-M3 -g -00 -apcs= I\RV31\INC	interwork:	:

图 3-29 添加两个重要预编译命令

(5) 在如图 3-29 所示的 C/C++选项卡中,单击 Include Path 后面的 一按钮,打开包含文件夹路径设置对话框。将工程中可能需要用到的头文件所在路径全部包含进来,操作结果如图 3-30 所示。



图 3-30 包含路径添加结果

(6) 在 Debug 选项卡中选中 Use Simulator 单选按钮,其操作界面如图 3-31 所示。至此工程文件 Options for Target 选项已全部配置完成,单击 OK 按钮退出。

第四步: 创建 public. h 文件,重写 main. c 文件,编译调试。

Options for Target 'Target 1'	
Device   Target   Output   Listing   Vser   C/C++	Asm Linker Debug Utilities
C Use Simulator Settings	⊈se: ULINK2/ME Cortex Debugger → Settings
Ir Load Application at Startup Ir Run to main() Initialization File	I Load Application at Startup I Run to main() Initialization File:
Reatore Debug Session Settings I Breakpoints I Toolbox: I Watch Windows & Performance Analyzer I Memory Display I System Viewer	Restore Debug Session Settings Preskpoints      Toolbox Watch Windows Memory Display Wetch Viewer
CPU DLL. Parameter. SARMCM3.DLL REMAP	Driver DLL Parameter: SARMCM3.DLL
Dialog DLL: Parameter: DCM.DLL [-pCM3	Dialog DLL Parameter: TCM DLL CM3

图 3-31 Debug 选项卡设置

(1) 在 Keil µVision4 工程文件界面中,依次单击 File→New 新建一个空白文件,并将 其以文件名 public.h 保存到工程模板的 User 文件夹下,在 public.h 文件中输入以下代码。

```
# ifndef _public_H
# define _public_H
# include "stm32f10x.h"
# endif
```

(2) 将原 main. c 中的程序删除,写一个 main 的空函数,并包含公共头文件 public. h。

```
# include "public.h"
int main()
{
}
```

(3) 对整个工程进行编译,结果如图 3-32 所示,如果下面的编译输出显示为:"".\工程 模板.axf"-0 Error(s),0 Warning(s)."则表示工程模板创建成功,如有错误则需要返回上 面布骤,找出原因并进行更正,直至没有编译错误出现为止。

至此整个工程模板就创建完成了,工程模板对整个嵌入式系统的学习是至关重要的,后面项目学习都是在该模板的基础上进行扩展的,所以大家应该熟练掌握模板的创 建方法。

C:\Users\Administrator\Deskto	p\工程模板\工程模板.uvproj- µVision4	- E ×
File Edit View Project Flash	Debug Peripherals Tools SVCS Window Help	
PBBB	•	
· 프 프	n1 💽 🔊 🏝 🗟 👁 🚊	
Project 4	🖹 🖹 main.c* 🛄 public.h	+ ×
Guser     main.c     main.c	<pre>     1 #include "public.h"     2     3 int main()     4     {         5         6         7         8         }         9         10         11         *         *</pre>	
Build Output		¥ 🗉
compiling stm32fl0x_rcc.« linking Program Size: Code=840 R( ".\工程模慎.sxf" - 0 Brit	 D-data=336 RW-data=20 2I-data=1636 5±(≤), 0 Worning[≥].	
•		Simulation

图 3-32 编译输出界面

# 3.3 软件模拟仿真



第一步: 创建项目工程,并编译生成目标文件。

在上一节创建的工程模板中 main.c文件中输入图 3-33 中方框中的源程序,该程序用 于在 PC0 端口输出方波信号,该程序只是用于讲解 Keil MDK 软件仿真操作,其代码较为 简单,读者将在后续章节逐步学习,其操作过程如图 3-33 所示。

第二步:将调试方式设置为软件模拟仿真方式。

在 Keil MDK-ARM 的工程管理窗口中,选中刚才编译连接成功的 Target1工程并右击,在 右键菜单中选择 Option for Target 'Target 1'命令,打开 Option for Target 'Target 1'对话 框。在该对话框中,选择 Debug 选项卡,选中左侧的 Use Simulator 单选按钮,同时设置左 侧 CPU DLL 为 SARMCM3. DLL 和 Parameter 为空,并设置左侧 Dialog DLL 为 DARMSTM. DLL 和 Parameter 为-pSTM32F103ZE,然后单击 OK 按钮确定,其操作界面及顺序如图 3-34 所示。

第三步:进入软件模拟调试模式。

选择菜单 Debug→start/stop Debug session 命令或者单击工具栏中的 Debug 按钮 @,

🕼 C:\Users\Administrator\Desktop\工程模	版\工程模板.uvproj - µVision4	
File Edit View Project Hash Debug	Perpherais Tools SVCS Window Help	
● Ⅲ → 算 factobit Project 2编译链接 ■ ① □□□ Target 1		• ×
User     Imain.c     Imai	2 int main() 1 输入源程序 { u16 i; RCC->APB2ENR=0x0010; //开时钟 GPIOC->CRL=0x11111111; //工作模式 while(I) 9 10 10 11 12 GPIOC->ODR=0x0; //PC0为低电平 for(i=0;i<1000;i++); //延时 GPIOC->ODR=0x1; //PC0为高电平 for(i=0;i<1000;i++); //延时 } 14 15 16 7 m	E
Build Output	236 DM docardo 77 docarde36	9 E
romELF: creating hex file '.\Output\工程模板.axf" - 0 Err	ror(s), 0 Warning(s). 3 编译结果	
	Simulation	-

图 3-33 创建工程项目并编译

Options for Target Target 1'	
Device   Target   Output   Listing   User   C/C++	Asm   Linker Debug   Stilities
C Use Simulator Settings	ULINK2/ME Cortex Debugger   Settings
IV Load Application at Startup IV Run to main() Initialization File	Load Application at Startup     Run to main() Initialization File:
	- Ent
Restore Debug Session Settings	Restore Debug Session Settings
🖾 Breakpoints 🖾 Toolbox	I Breakpoints I Toolbox
Watch Windows & Performance Analyzer	🔽 Watch Windows
Memory Display V System Viewer	I Memory Diaplay I System Viewer
CPU DLL Parameter	Driver DLL Parameter
SARMCM3.DLL 2	SARMCM3.DLL
Dialog DLL: Parameter:	Dialog DLL. Parameter:
UNARMSTM.DLL IPSTM32F103ZE 3	TARMSTM.ULL PSTM32F1032E
05 0	ancel Defaults Help

图 3-34 软件模拟调试方式设置

进入软件模拟调试模式,其操作界面如图 3-35 所示。

File	Edit	View	Project	Flash	Deb	pug	Peripherals	Tools	SVCS	Window	Help	
R	31		1.0	3	Q	Start	/Stop Debug	g Session		Ctrl+F5	11: 11:	2
de.	171 6	-6	and ump	Trun	944	Deret	FOIL					

图 3-35 进入/退出模拟调试方式

第四步:打开相关窗口添加监测变量或信号。

选择菜单 View→Analysis Windows→Logic Analyzer 命令或者直接单击工具栏中的 Logic Analyzer 应按钮,打开逻辑分析仪窗口,如图 3-36 和图 3-37 所示。

Eile Edit	View Project Flash Debug	Peripherais Tools SVCS Window Help	
Registers Registers Core Ra Ra Ra Ra Ra Ra Ra Ra Ra Ra Ra Ra Ra	<ul> <li>✓ Status Bar Tgolbars</li> <li>■ Eroject Window</li> <li>④ Books Window</li> <li>④ Functions Window</li> <li>④ Functions Window</li> <li>◎ Source Browser Window</li> <li>◎ Build Output Window</li> <li>◎ Error List Window</li> <li>◎ Error List Window</li> <li>○ Command Window</li> <li>○ Command Window</li> <li>○ Disassembly Window</li> <li>○ Symbols Window</li> <li>○ Registers Window</li> <li>○ Call Stack Window</li> <li>○ Window</li> </ul>	<pre></pre>	AD_Value 願・  ジ ジ ・ ジ ・ ジ ・ ジ ・ ジ ・ ジ ・ ジ ・ ジ
Project	Memory Windows Segial Windows		
Command	Analysis Windows	Eggic Analyzer	9 E
A PORT	Tracg	E Performance Analyzer     ocation/Value	Тур
* []	System Viewer	Code Coverage 080001CC	in_
>	🛠 Toolbox Window	4	+
SSIGN B	TT BUILDE HAR DE HAR	Call Stack + Locals Memory 1	

图 3-36 打开逻辑分析仪



图 3-37 逻辑分析仪窗口

### 58 - ARM Cortex-M3嵌入式原理及应用——基于STM32F103微控制器

单击逻辑分析仪窗口的 Setup 按钮,打开 Setup Logic Analyzer 对话框,单击右上角的 New 按钮,在空白框中输入 PORTC.0 新增一个观测 信号,并在 Display Type 下拉列表框中选择 Bit, 单击 Close 按钮退出,如图 3-38 所示。这样,就在 Logic Analyzer 窗口中添加了一个观测信号 PORTC.0。在程序软件仿真运行过程中,可通过 观察该信号的波形图得到 STM32F103 微控制器 的引脚 PC0 上输入或输出的变化情况。

第五步:软件模拟运行程序,观察仿真结果。

选择菜单 Debug→Run 命令或者单击工具栏 中的 Run 按钮,开始仿真。让程序运行一段时间 后,再选择菜单 Debug→Stop 命令或者单击工具 栏中的 Stop 按钮,暂停仿真,其操作界面如 图 3-39 所示。

etup Logic Analyzer	
Current Logic Analyzer-Signals	±⊐ ×
+	- Diarday Barota
Dienlay Type:	Max: OxFFTF
Color:	Min: Ox0
Display Formula (Signal & Mask) :	>> Shft
And Mask Dx00000001	Shift Right: 0
Export / Import Export Signal Definitions	Import Signal Definitions
	T Delete actual Signals
KILAL	Gose Help

图 3-38 添加观测信号

C:\Users\Adn	ninistrator,Deskto	p(工程模板)工程模板.uvpro)。pVision4	×
File Edit View	v Project Flash	Debug Peripherals Tools SVCS Window	w Help
nead	1 1. 1 23	Q Start/Stop Debug Session Ctrl+F5	//= //= 🖄 AD_Value
ST EO	Pr 70 - 10 11	RST Reset CPU	■・回・圖・  次・
Registers	4 🙆	E Run F5	• ×
Register	Value 🔺	Stop	
E Core	0.000	(+) Step F11	
81	0x000	0 <sup>1</sup> Step Over F10	
82	D::400	()+ Step Out Ctri+F11	and Urrath
R4	Dx000	*{} Run to Cursor Line Ctrl+F10	10010; // 开时评
RS	0x200	Show Next Statement	1111111: //工作侯式, =
R7	0x000		
RS	Dx000	Breakpoints Ctrl+B	10· //PC0为低电平
-R10	0x080.	Insert/Remove Breakpoint F9	00:i++) : //延时
Ril	0x000.	Enable/Disable Breakpoint Ctrl+F9	1. /PC0为高电平
R13 (SP)	0x200	O Disable All Breakpoints	0;i++); //延时
R14 (LR)	0x080 *	Rill All Breakpoints Ctrl+Shift+F9	
El Project	egisters	OS Support	
ommana		Evention Profiling	9 E
A PORTC	a vice convice	Lacturer roming	Location/Value Type
LA (PORTC &	0x00000001)	Memory Map	0x080001CC int f0
		Inline Assembly	alanta da sere
SSTON Break	Disable Break	Function Editor (Open Ini File)	Patr Ditamon 1
aston preav	stonete predi	Debug Settings	Cimulation
-		and the second se	Simulation

图 3-39 开始/停止运行

然后,在 Logic Analyzer 窗口中可以看到程序仿真运行期间 PC0 的信号图,如图 3-40 所示。由图可以看出,PC0 端口输出信号为一方波,占空比为 50%,符合项目预期效果。如 果看不清信号波形图,单击 Zoom 中的 All 按钮可以显示全部波形,还可以通过 In 按钮放

大波形,Out 按钮缩小波形。

Logic Analyzer												
Setup	Load	Min Time	Max Time	Grid	Zoom	Min/Max	Update Sc	treen Tr	ransition	Jump to	Signal Info	Anplitude
A 0x0000001)	1											
0,9 672	134198 <u>0. (</u> 162243672	263294]				0, 934548 67287443	*		4			0,934898 ± 67312643

图 3-40 逻辑分析仪输出波形

第六步:退出模拟仿真调试模式。

选择菜单 Debug→Start/Stop Debug Session 命令或者单击工具栏中的 Debug 按钮 ④,即可退出模拟仿真调试模式。

上述例题只是 Keil MDK-ARM 模拟仿真最简单的应用,其观测信号范围十分广泛,包括 I/O 端口、逻辑数值、寄存器、存储器等,还可以设定外设的工作状态,包括 GPIO、ADC、 DMA、TIMER 等,还经常被用于时序分析,调试输出等典型应用,功能十分强大。用好模拟 仿真调试,可以提前发现错误,是硬件运行调试的有益补充。

### 本章小结

本章的主要任务是创建是一个正确的、合理的、适合自己的工程模板,这项工作也是本 课程后续学习的基础。本章首先对固件库进行认知,包括固件库概述、下载和目录结构等。 随后开始工程模板的创建,包括 Keil MDK-ARM 软件简介和工程模板创建具体步骤详解。 本章最后还介绍了基于 Keil MDK-ARM 软件的模拟仿真调试,该部分内容为选学内容,是 硬件运行调试的有益的补充。

# 思考与扩展

- 1. 什么是 STM32 固件库?
- 2. 目前通用的 STM32F10x 固件库版本是多少?
- 3. STM32 固件库下载网址是什么?并从该网站下载最新的固件库。
- 4. 请指出固件库中文件 core\_cm3. c 文件的路径。
- 5. 请指出固件库中文件 stm32f10x.h 文件的路径。
- 6. Keil MDK-ARM 软件的下载网址是什么?并从该网站下载最新的软件包。
- 7. Keil MDK-ARM 软件建立的工程文件的扩展名是什么?
- 8. 参照书中工程模板创建方法,建立自己的工程模板。