

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。



工程模板创建

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。

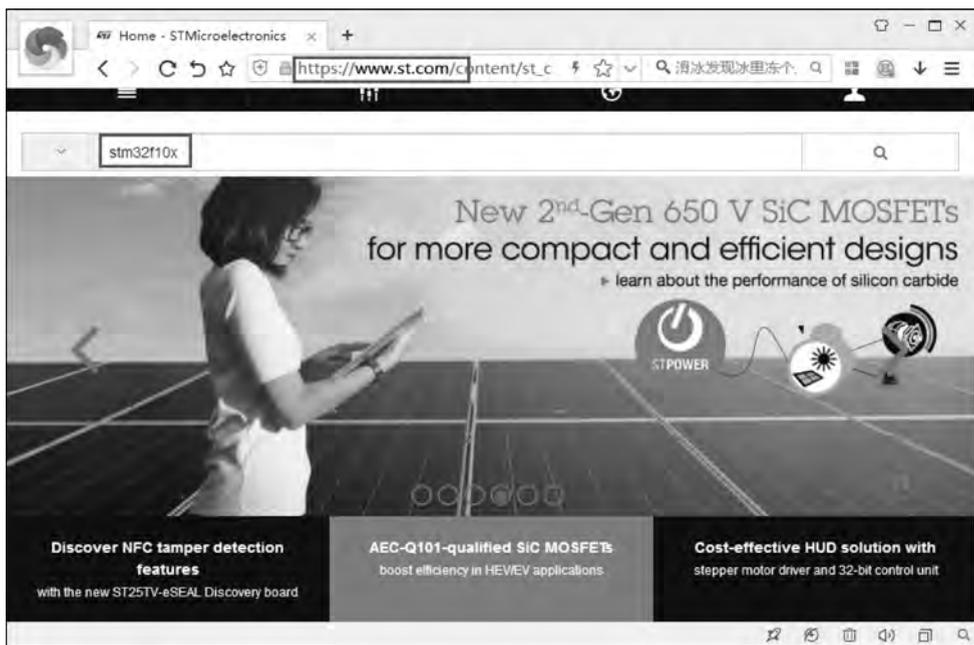


图 3-1 搜索资料操作界面

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	Type	Category	Description
STSW-STM32023	Active	Embedded Software	MCUs Embedded Software	How to migrate from the STM32F10xxx firmware library V2.0.3 to the STM32F10xxx standard peripheral library V3.0.0 (AN2953)
STSW-STM32011	Active	Embedded Software	MCUs Embedded Software	Smartcard interface with the STM32F10x and STM32L1xx microcontrollers (AN2598)
STSW-STM32024	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					
Part Number	General Description	Software Version	Supplier	Marketing Status	Download
STSW-STM32054	STM32F10x standard peripheral library	3.5.0	ST	Active	Get Software

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。



图 3-4 STM32 固件库目录结构

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。

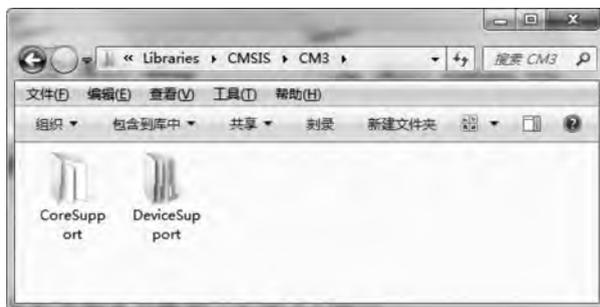


图 3-6 Libraries\CMSIS\CM3 目录

(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。



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`。



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 STM32F10x 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

(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。

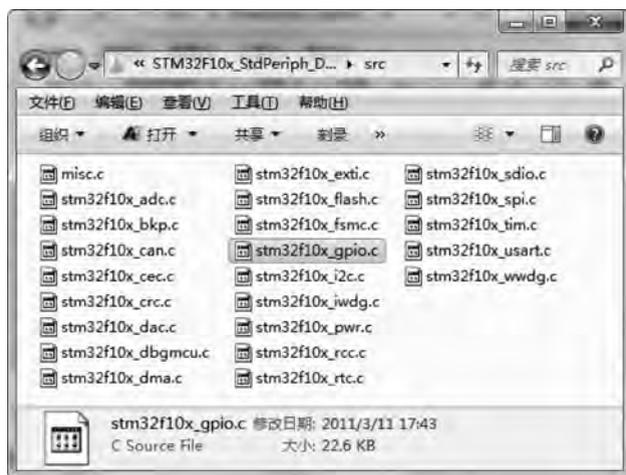


图 3-11 STM32F10x_StdPeriph_Driver\src 目录

(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。

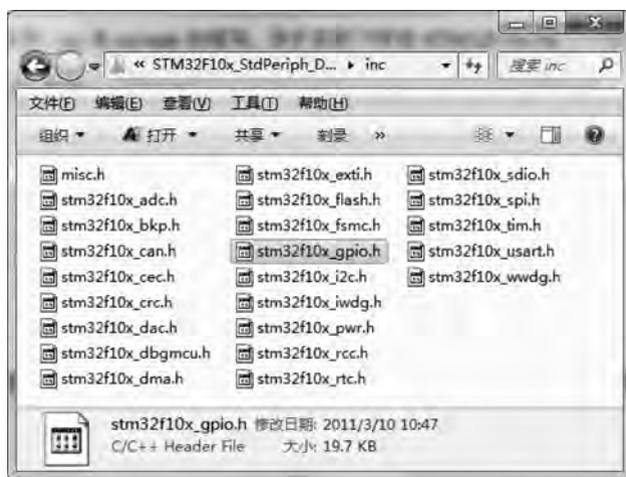


图 3-12 STM32F10x_StdPeriph_Driver\inc 目录

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。

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。

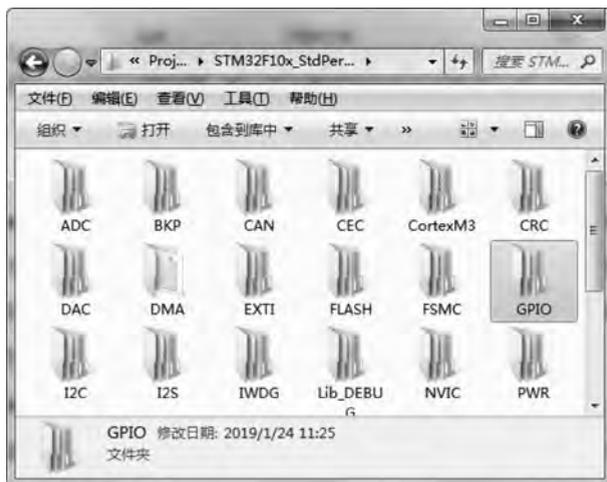


图 3-15 Project\STM32F10x_StdPeriph_Examples 目录

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。

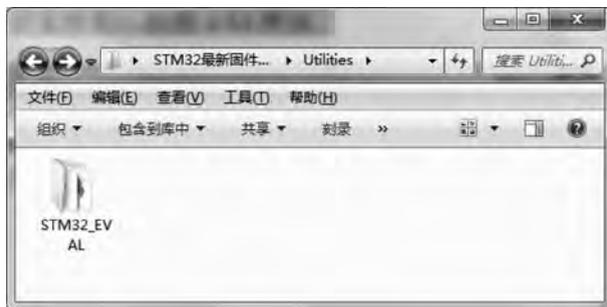


图 3-16 Utilities 文件夹目录

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 μ Vision4 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) μ Vision4 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,

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\CMS3\DeviceSupport\ST\STM32F10x\startup\arm。操作结果如图 3-17 所示。



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。



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。



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 μ Vision4 软件，其界面如图 3-20 所示。

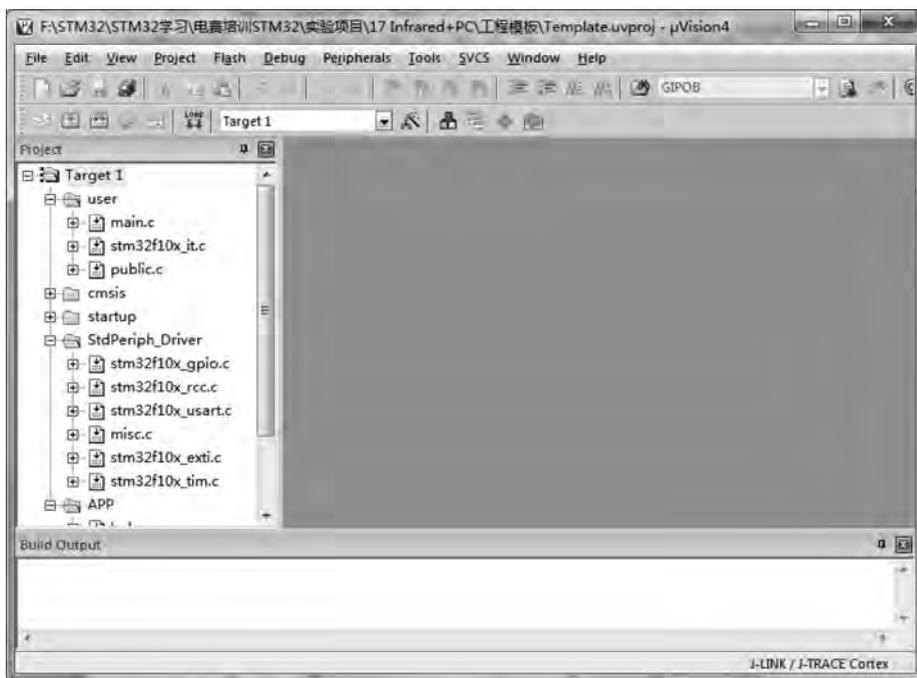


图 3-20 Keil μ Vision4 软件界面

(2) 依次单击菜单栏 Project→new μ vision 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。



图 3-21 保存新建工程文件

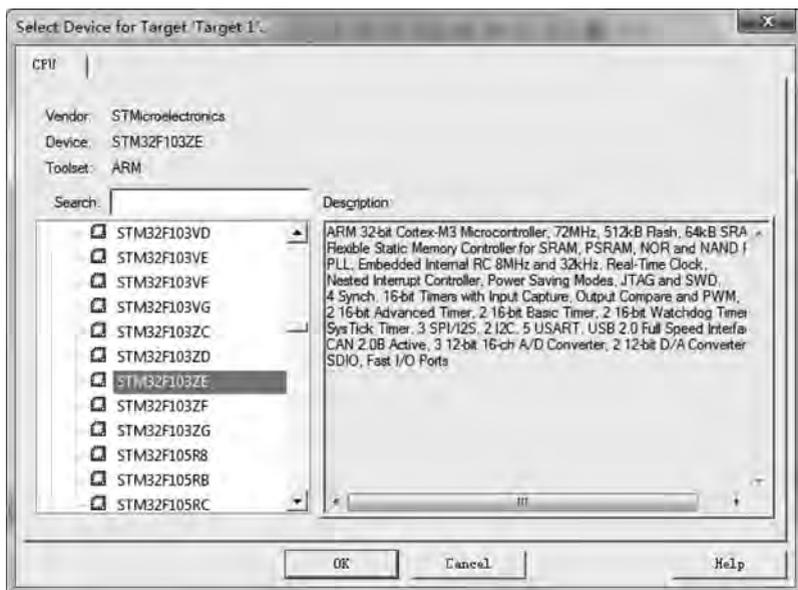


图 3-22 CPU 芯片选择



图 3-23 复制启动文件选项

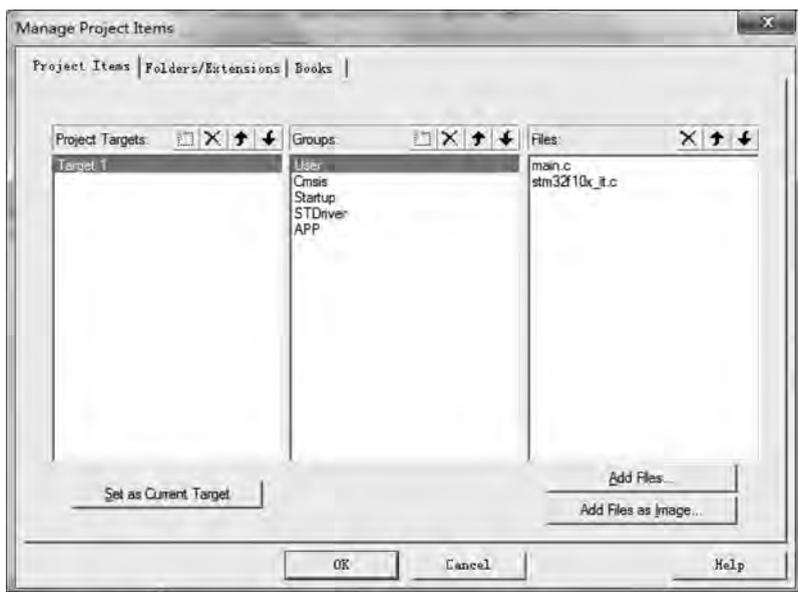


图 3-24 添加工程分组对话框

(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。

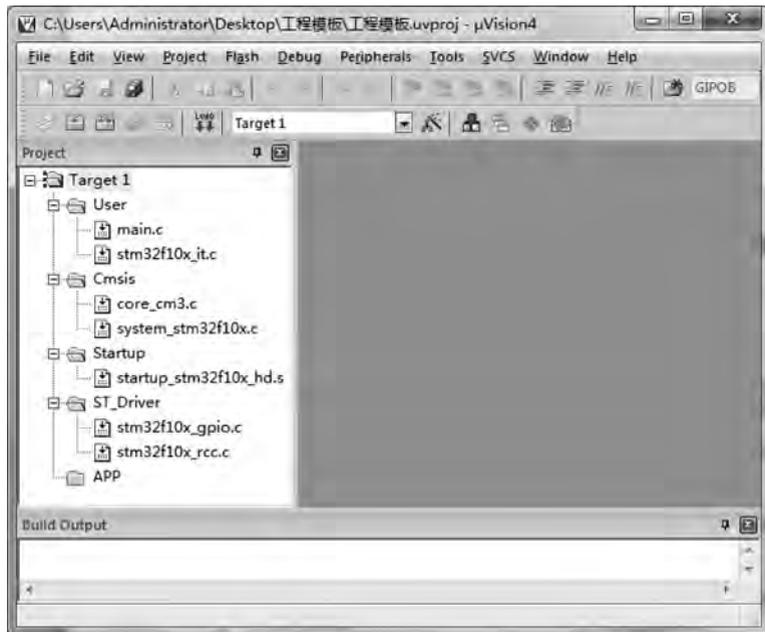


图 3-25 建立分组和添加文件完成

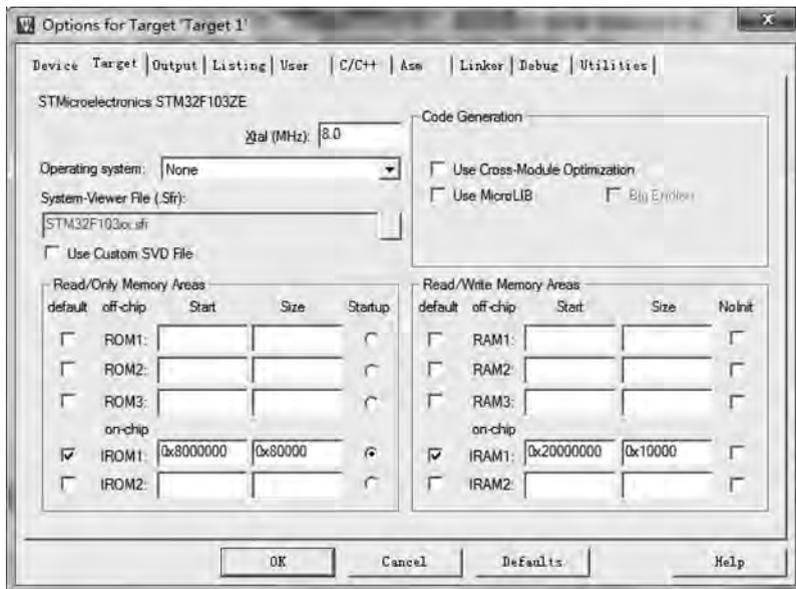


图 3-26 Options for Target 'Target 1'对话框



图 3-27 Output 选项卡设置



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。

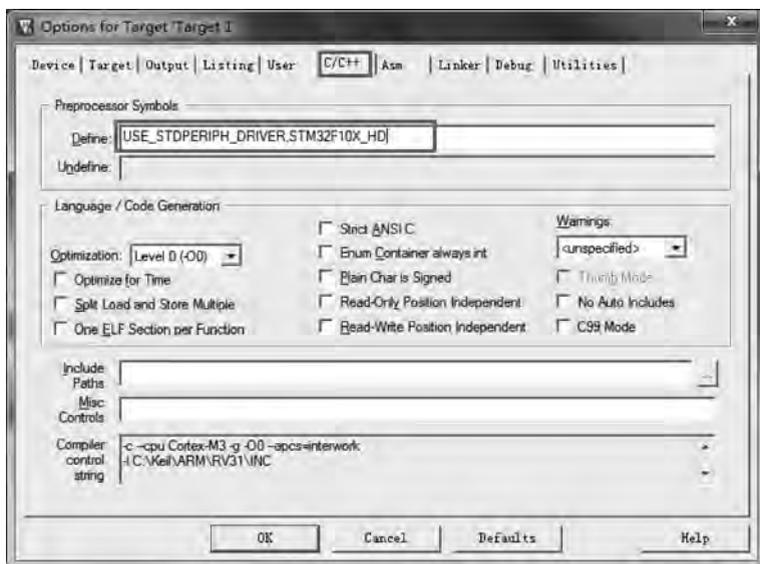


图 3-29 添加两个重要预编译命令

(5) 在如图 3-29 所示的 C/C++ 选项卡中,单击 Include Path 后面的  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。

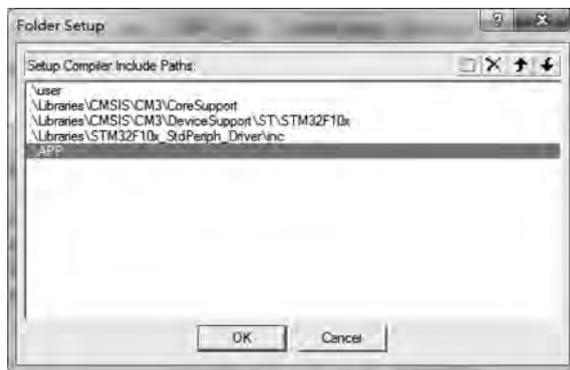


图 3-30 包含路径添加结果

(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。

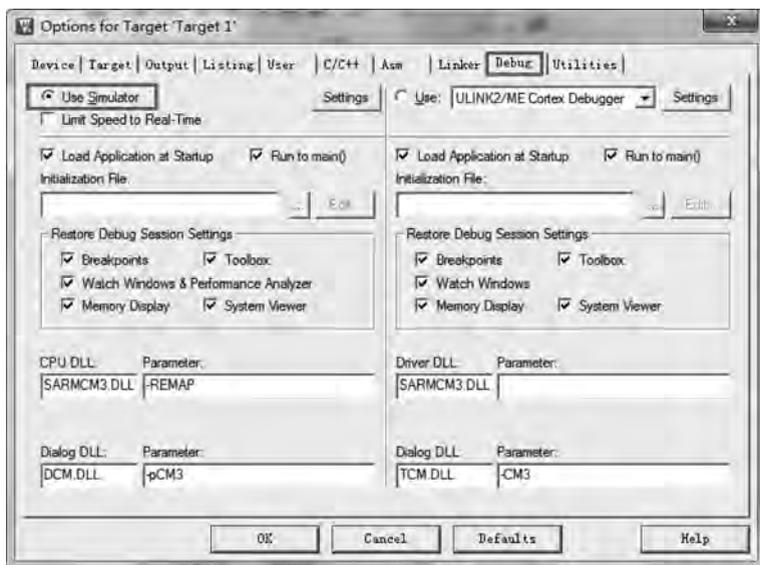


图 3-31 Debug 选项卡设置

(1) 在 Keil μ Vision4 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步骤,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。

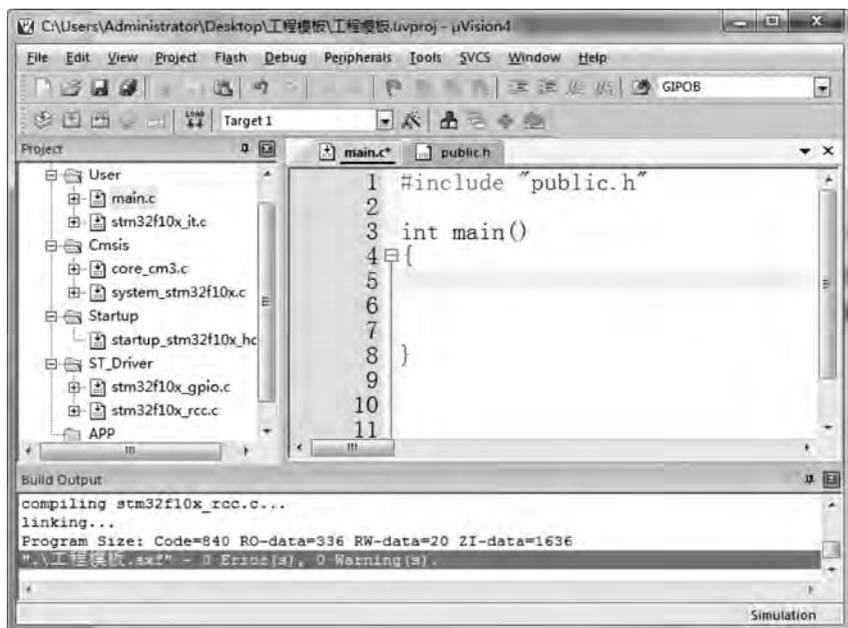


图 3-32 编译输出界面

3.3 软件模拟仿真

Keil MDK-ARM 同样提供了强大的软件模拟仿真功能,对于暂不具备硬件平台的学习者也可以通过模拟仿真学习嵌入式开发。下面以一个具体实例讲解软件模拟仿真方法。

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 ,



模拟仿真调试

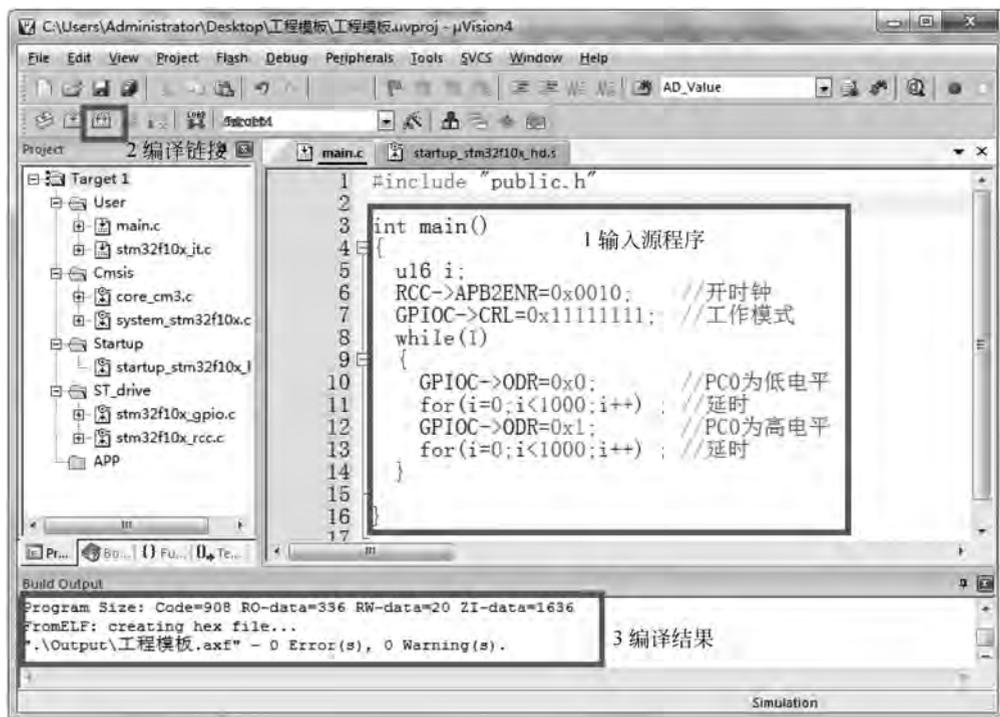


图 3-33 创建工程项目并编译

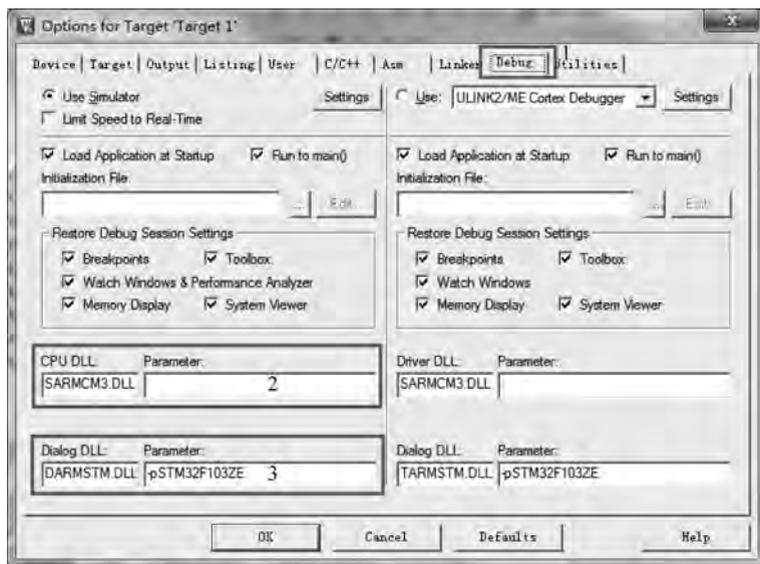


图 3-34 软件模拟调试方式设置

进入软件模拟调试模式,其操作界面如图 3-35 所示。

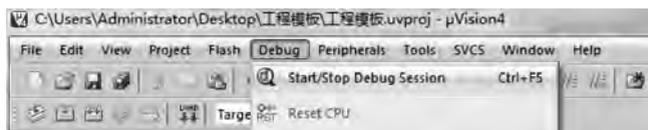


图 3-35 进入/退出模拟调试方式

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  按钮,打开逻辑分析仪窗口,如图 3-36 和图 3-37 所示。

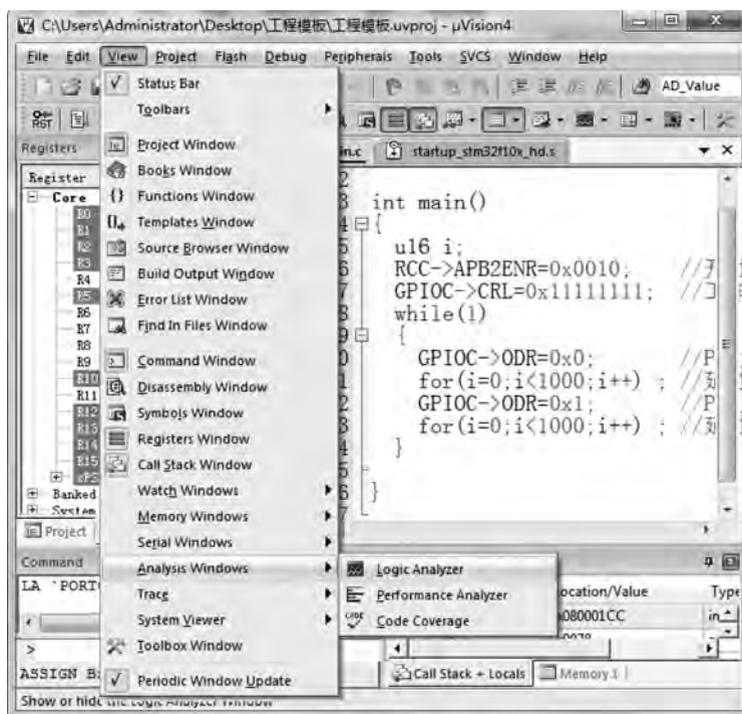


图 3-36 打开逻辑分析仪

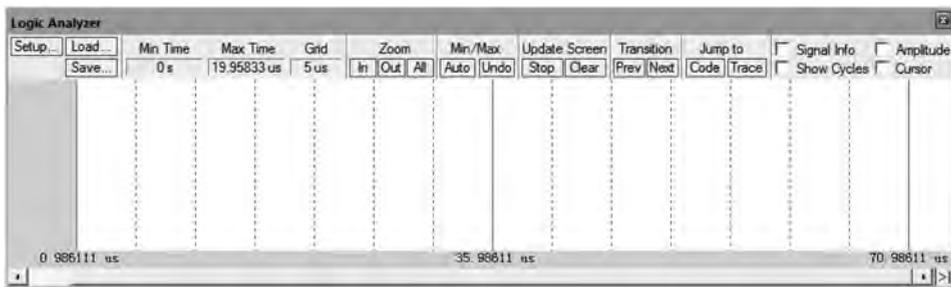


图 3-37 逻辑分析仪窗口

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。



图 3-38 添加观测信号

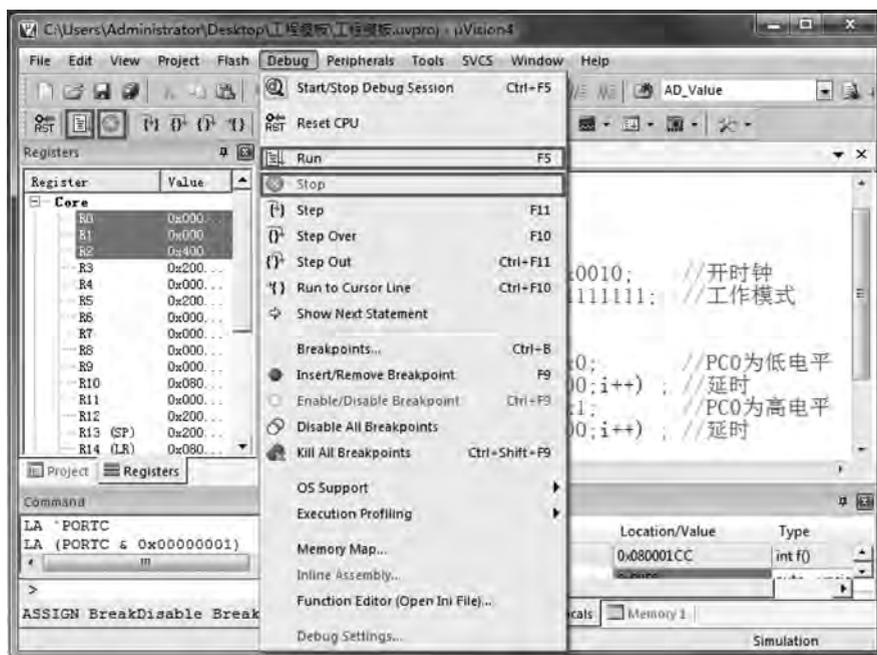


图 3-39 开始/停止运行

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。

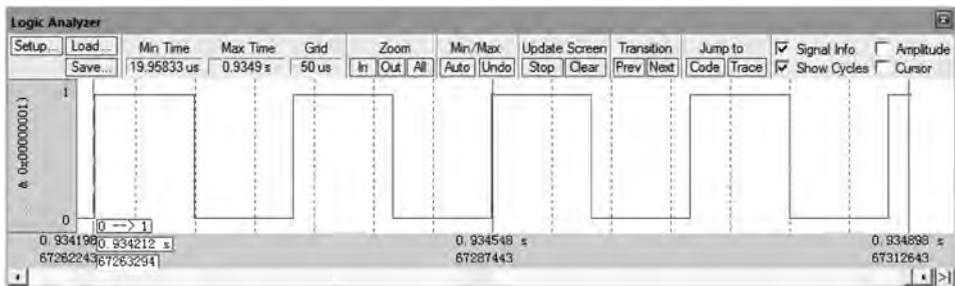


图 3-40 逻辑分析仪输出波形

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 ,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。