

3.1 实验目的

- 熟悉在 Keil 工具中编写 C 语言程序；
- 熟悉 C 语言程序的调试。

3.2 实验设备

1. 硬件

PC 一台。

2. 软件

- (1) Windows 7/8/10 系统；
- (2) Keil μ Vision5 集成开发环境。

3.3 实验内容

在 Keil 工具中通过 C 语言完成 $\text{sum} = 1 + 2 + \dots + 10$ 的程序编写,并在仿真模式下进行调试。

3.4 实验预习

- 复习 C 语言编程；
- 学习在 Keil 工具下如何建立工程及配置。

3.5 实验原理

在调试该 C 语言程序时,配置本实验教材所使用的开发板,或以仿真模式运行。

3.6 实验步骤

3.6.1 创建工程

在 Keil 工具中创建一个新工程,在实验 2 创建工程的基础上在 USR 中添加 main 函数。

3.6.2 修改配置

单击如图 3.1 所示方框中的按钮,打开配置界面。

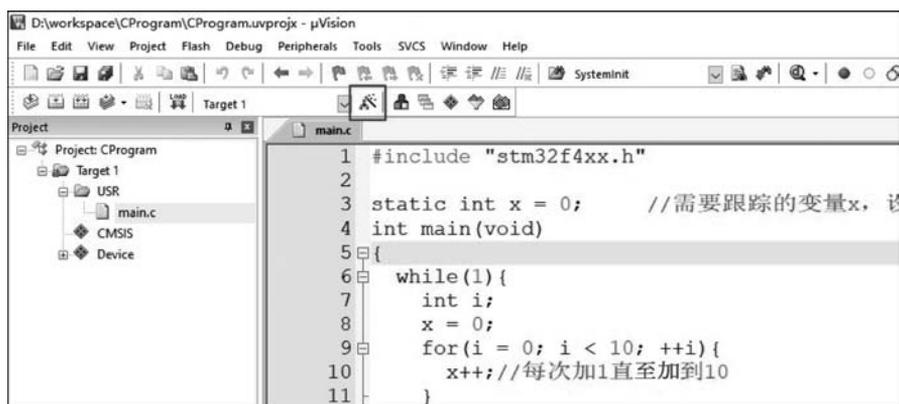


图 3.1 打开配置界面按钮

切换到 Debug 选项卡,如图 3.2 所示。

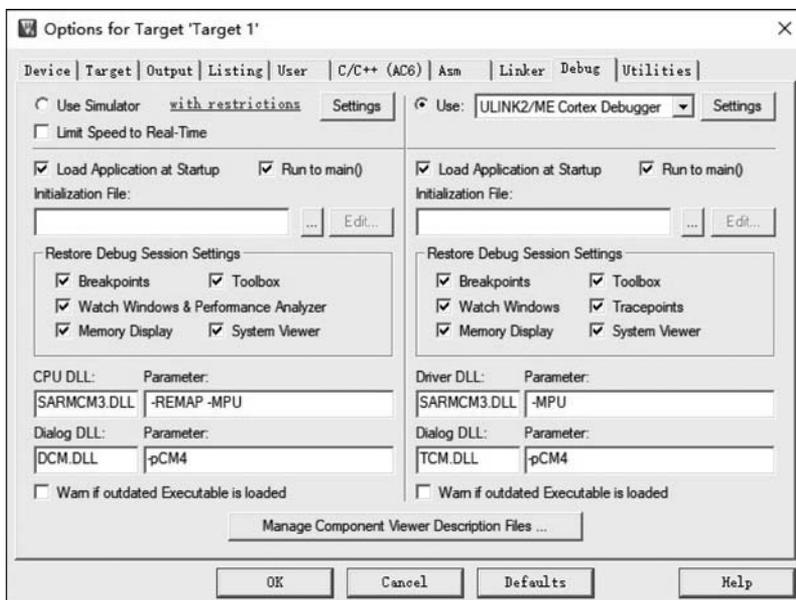


图 3.2 Debug 选项卡

将选项卡中的内容改为如图 3.3 所示,注意粗黑框标注的内容,具体修改见下文。

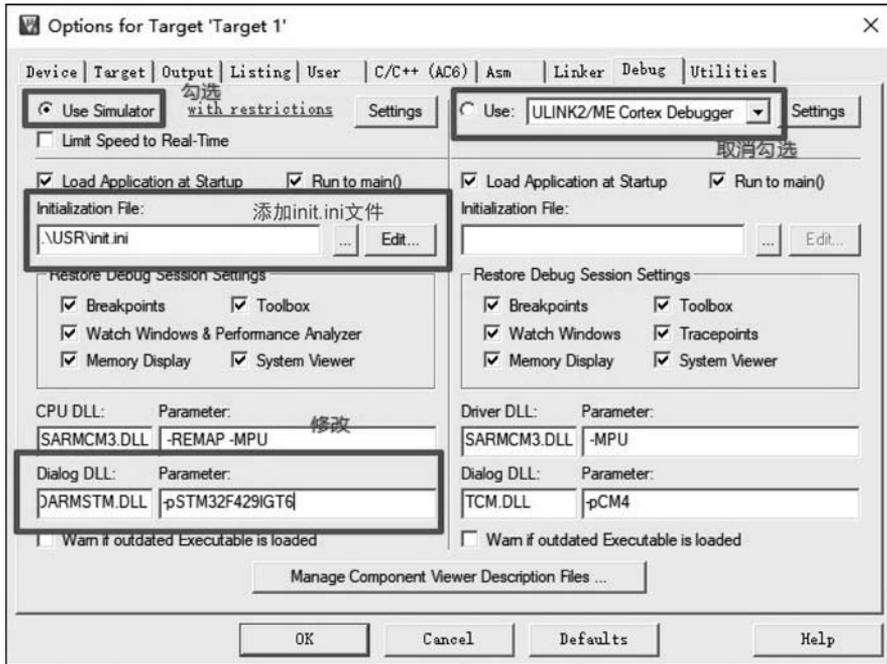


图 3.3 需要修改 Debug 选项卡的内容

其中:

Use Simulator 代表要使用软件模拟仿真。

Dialog DLL 的内容是 DARMSTM.DLL,表示使用 STM 系列。

Parameter 的内容是“-pSTM32F429IGT6”,这里匹配的是当前开发板的型号。

Initialization File,该选项可以配置映射一些地址,从而实现 STM32 开发板不同功能的模拟仿真。一个典型的 Initialization File 内容如下:

```
map 0x40000000,0x4000ffff read write
map 0x40010000,0x4001ffff read write
map 0x40020000,0x4002ffff read write
map 0x48000000,0x4800ffff read write
```

在工程文件夹下新建 init.ini 文件,将上述 4 行内容添加到这个文件中,在 Debug 选项卡的 Initialization File 选项下添加该文件。

在该文件中,我们设置了 4 段地址映射,使得可以实现模拟某些功能。一个大致的地址功能映射表如图 3.4 所示。

上面只是截取了部分存储器的映射情况,这是 AHB1 的地址。例如,如果需要使用 RCC 功能,那么可以看到 0x40021000 至 0x400213FF 这部分地址需要有读写属性,可以在 Initialization File 中添加相应映射地址的字段来启用读写。注意:该地址对于不同型号的开发板会有变化。

Debug 选项卡中的内容修改完毕后,单击 OK 按钮完成配置。

总线	编址范围	大小	外设
AHB1	0x4002 4000 - 0x4002 43FF	1 KB	TSC
	0x4002 3400 - 0x4002 3FFF	3 KB	Reserved
	0x4002 3000 - 0x4002 33FF	1 KB	CRC
	0x4002 2400 - 0x4002 2FFF	3 KB	Reserved
	0x4002 2000 - 0x4002 23FF	1 KB	FLASH 接口
	0x4002 1400 - 0x4002 1FFF	3 KB	Reserved
	0x4002 1000 - 0x4002 13FF	1 KB	RCC

图 3.4 部分存储器的映射

3.6.3 跟踪变量

配置完成后,就可以通过 Debug 进行仿真了。本次实验中,我们通过一个简单的累加来仿真,验证程序的正确性。main.c 文件内容如下:

```
#include "stm32f4xx.h"

static int x=0;           //需要跟踪的变量 x,设置为全局变量,局部变量无法用分析器跟踪
int main(void)
{
    while(1){
        int i;
        x=0;
        for(i=0; i<10; ++i){
            x++;           //每次加 1 直至加到 10
        }
    }
}
```

在编写好程序后,单击 Keil 左上角的 Build 或 Rebuild 按钮进行编译,如图 3.5 所示。

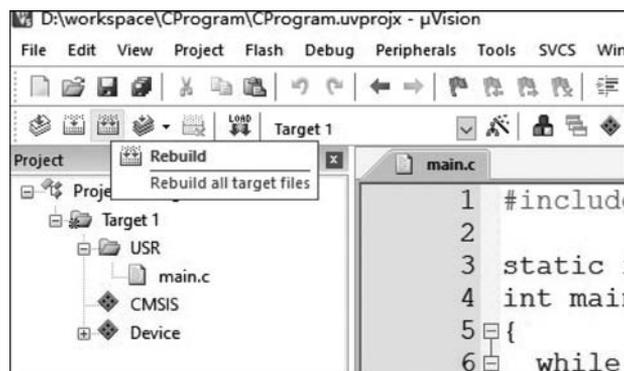


图 3.5 重新编译

在如图 3.6 所示的 Output 框中看到无错误(0 Error(s))提示后即编译成功。编译完成后,单击 Debug 按钮,然后单击分析器按钮,如图 3.7 所示。

```

Rebuild started: Project: CProgram
*** Using Compiler 'V6.15', folder: 'C:\Keil_v5\ARM\ARMCLANG\Bin'
Rebuild target 'Target 1'
compiling main.c...
assembling startup_stm32f429xx.s...
compiling system_stm32f4xx.c...
linking...
Program Size: Code=376 RO-data=444 RW-data=0 ZI-data=1632
".\Objects\CProgram.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:00

```

图 3.6 重新编译成功的 Output 框

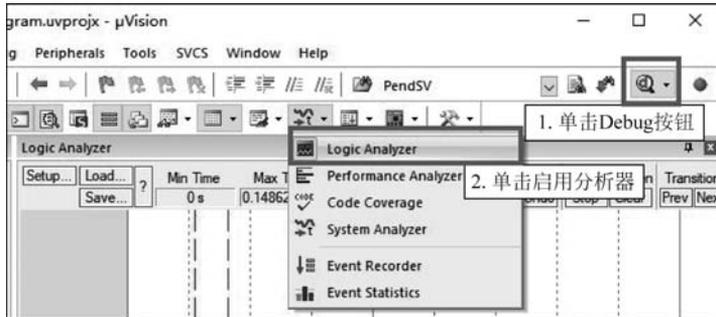


图 3.7 打开 Debug 窗口并启用分析器

单击分析器左上角的 Setup 按钮,如图 3.8 所示。

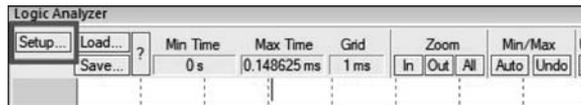


图 3.8 单击 Setup 按钮

在如图 3.9 所示界面中单击右上角标识处添加变量 x(如果已经有则不用添加,关闭此框即可),添加完毕后单击 Close 按钮。



图 3.9 添加变量

在分析器中左侧看到变量 x,即添加成功,如图 3.10 所示。请注意,需要在分析器中跟踪的变量必须为全局变量,若设置为局部变量,则可能无法添加到分析器中。



图 3.10 添加变量成功

因为分析器展示的默认范围非常广(和变量类型上下限有关),因此需要设置自动调整范围以更好地展现值的变化。在分析器窗口中空白位置右击,选中 Adaptive Min/Max 选项,如图 3.11 所示。

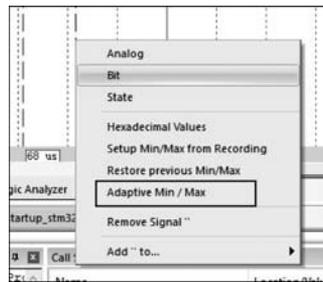


图 3.11 设置仿真变量的最大/最小值

单击 Keil 左上角的 Run 按钮,可以看到分析器中出现波形,如图 3.12 所示。

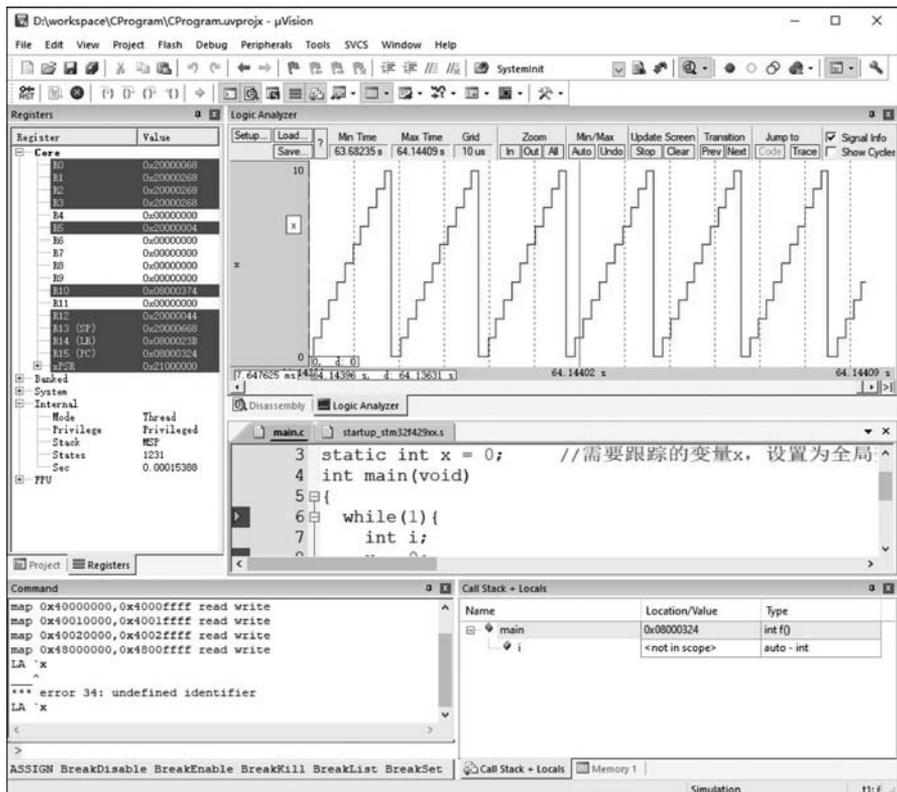
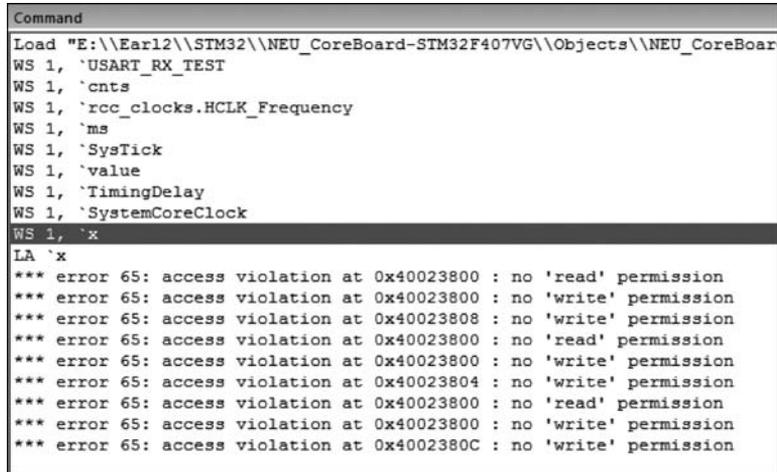


图 3.12 观察变量的波形

可以看到,波形从 0~10 以阶梯形式上升,符合程序设定,每次加 1 直到 10 的结果是一致的。

请注意,如果 Initialization File 没有被加载或者没有正确配置地址,那么在 Command 窗口会出现如图 3.13 所示的错误信息。此时需要重新配置 Initialization File。



```

Command
Load "E:\Earl2\STM32\NEU_CoreBoard-STM32F407VG\Objects\NEU_CoreBoar
WS 1, `USART_RX_TEST
WS 1, `cnts
WS 1, `rcc_clocks.HCLK_Frequency
WS 1, `ms
WS 1, `SysTick
WS 1, `value
WS 1, `TimingDelay
WS 1, `SystemCoreClock
WS 1, `x
LA `x
*** error 65: access violation at 0x40023800 : no 'read' permission
*** error 65: access violation at 0x40023800 : no 'write' permission
*** error 65: access violation at 0x40023808 : no 'write' permission
*** error 65: access violation at 0x40023800 : no 'read' permission
*** error 65: access violation at 0x40023800 : no 'write' permission
*** error 65: access violation at 0x40023804 : no 'write' permission
*** error 65: access violation at 0x40023800 : no 'read' permission
*** error 65: access violation at 0x40023800 : no 'write' permission
*** error 65: access violation at 0x4002380C : no 'write' permission
  
```

图 3.13 初始文件配置错误信息

此外,还有一种在设置变量时失败的情况,是在最新版 Keil μ Vision5 中,默认的 ARM 编译器会使用版本 6 的编译器,在该版本的编译器进行编译后,在 Debug 选项卡的 Setup 里无法设置变量,提示 Unknown Signal。解决方案是在 Target 选项卡中将 ARM Compiler 切换到版本 5 或以下的版本,如图 3.14 所示。

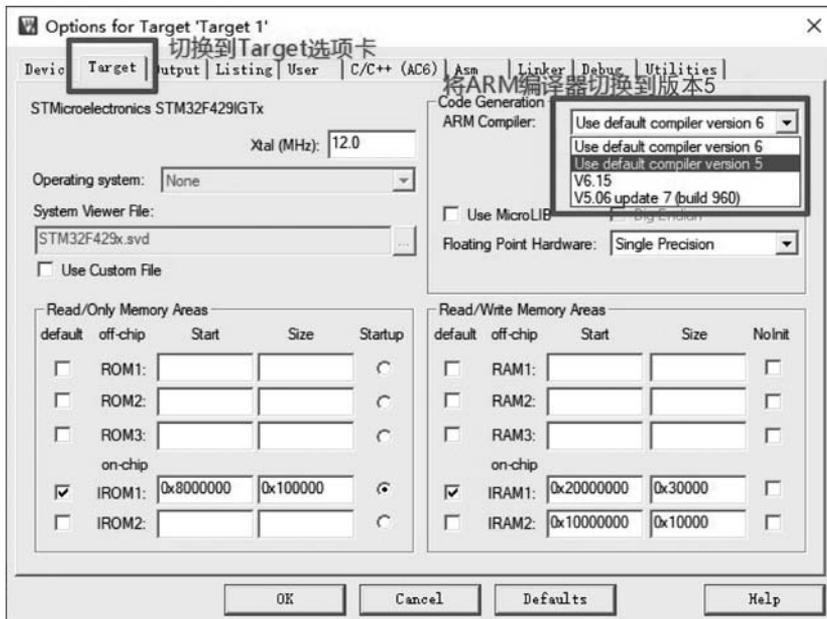


图 3.14 切换 ARM 编译器版本

3.7 实验参考程序

```
#include "stm32f4xx.h"

static int x = 0;           //需要跟踪的变量 x, 设置为全局变量, 局部变量无法用分析器跟踪
int main(void)
{
    while(1){
        int i;
        x = 0;
        for(i = 0; i < 10; ++i){
            x ++;           //每次加 1 直至加到 10
        }
    }
}
```

3.8 实验总结

本实验给出了如何使用 Keil 软件进行仿真的例子：从 Keil 软件仿真器中模拟运行一个简单的累加计算。初学者经过本节的实验理解和体验 STM32 的仿真过程，熟练掌握 Keil 软件的 STM32 系列开发板的仿真器使用方法。

3.9 思考题

- (1) 尝试通过配置其他映射地址，启用 GPIO 引脚，观察 GPIO 引脚高低电平变化。
- (2) 使用 Debug 模式下的 Watch 功能观察变量的变化，并比较和仿真分析器的相同和不同点。
- (3) 查阅资料了解不同型号的 STM32 的地址映射配置。