《微机原理与嵌入式系统》

基于Keil µVision5 Simulator的自主实验建议

（2020年特别定制版）

**2020年自主实验项目报告要求**

* **格式方面美观简洁即可，不需要严格按照常规实验报告撰写。**
* **必须写学号、姓名；电子版和书面报告均可。**
* **实验报告中要能显示上机内容已做（可附关键代码）；对于要求记录结果的问题，请在实验报告上记录结果；对于要求分析的结果，请给出文字分析。**
* **除标明为选作（\*）的思考题外，需要在实验报告上作答。**
* **实验内容和思考题均有编号，在实验报告中请保持编号信息。**
* **所有自主完成的实验项目撰写一份总的实验报告，在BB系统提交。**

中国科学技术大学 • 信息科学技术学院

目录

[实验1 基于ASM的Project 4](#_Toc40918253)

[1.1 实验目的 4](#_Toc40918254)

[1.2 实验内容 4](#_Toc40918255)

[1.2.1 Project的建立、编译、连接 4](#_Toc40918256)

[1.2.2 Project的调试（Debug） 6](#_Toc40918257)

[1.3 思考题（内容需要上机调试才能完成） 9](#_Toc40918258)

[实验2 基于C的Project 11](#_Toc40918259)

[2.1 实验目的 11](#_Toc40918260)

[2.2 实验内容 11](#_Toc40918261)

[2.2.1 建立基于C程序的Project 11](#_Toc40918262)

[2.2.2 代码功能验证 12](#_Toc40918263)

[2.2.3 代码分析技巧 12](#_Toc40918264)

[2.2.4 µVision联机资源使用 14](#_Toc40918265)

[2.2.5 Project的调试（Debug） 16](#_Toc40918266)

[2.3 思考题（内容需要上机调试才能完成） 17](#_Toc40918267)

[实验3 基于STM32库的GPIO与定时器 19](#_Toc40918268)

[3.1 实验目的 19](#_Toc40918269)

[3.2 实验内容 19](#_Toc40918270)

[3.2.1 下载ST公司STM32库及芯片手册 19](#_Toc40918271)

[3.2.2 建立基于STM32库的Project 20](#_Toc40918272)

[3.2.3 配置Project的头文件目录、预编译参数、Simulator 21](#_Toc40918273)

[3.2.4 Debug时使用外设仿真功能验证GPIO输入和输出 23](#_Toc40918274)

[3.2.5 Debug时使用Logic Analyzer观察GPIO输出 23](#_Toc40918275)

[3.2.6 使用外设仿真和Logic Analyzer验证GPIO输入和输出 25](#_Toc40918276)

[3.2.7 定时器的配置 26](#_Toc40918277)

[3.3 思考题（部分内容需要上机调试才能完成） 26](#_Toc40918278)

[实验4 基于STM32库的中断 28](#_Toc40918279)

[4.1 实验目的 28](#_Toc40918280)

[4.2 实验内容 28](#_Toc40918281)

[4.2.1 建立基于STM32库的Project 28](#_Toc40918282)

[4.2.2 配置Project的头文件目录、预编译参数、Simulator 29](#_Toc40918283)

[4.2.3 使用Logic Analyzer和外设仿真功能验证EXIT及GPIO输出 29](#_Toc40918284)

[4.2.4 观察NVIC寄存器组 29](#_Toc40918285)

[4.3 思考题（内容需要上机调试才能完成） 30](#_Toc40918286)

[附录1：相关资料下载 32](#_Toc40918287)

[ST公司STM32库及芯片手册 32](#_Toc40918288)

[ST公司芯片手册 32](#_Toc40918289)

[Keil5 手动安装STM32 芯片包 33](#_Toc40918290)

[附录2：µVision5 IDE常见错误 34](#_Toc40918291)

[使用Simulator调试程序时，access violation 34](#_Toc40918292)

[µVISION支持的Simulation 34](#_Toc40918293)

[FCARM - Output Name not specified 35](#_Toc40918294)

[Undefined symbol assert\_param 36](#_Toc40918295)

[帮助仅显示µVision User's Guide 36](#_Toc40918296)

# 基于ASM的Project

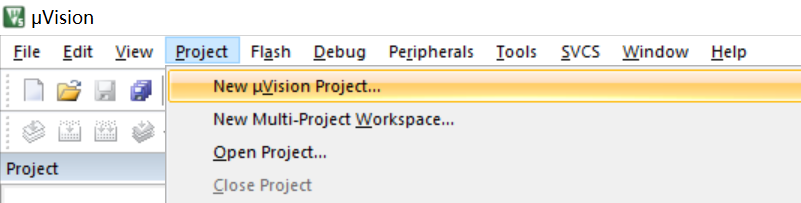
## 实验目的

1. 掌握µVision IDE基本使用、了解一个项目编译、连接、调试的工作过程
2. 汇编代码编写的一般语法，掌握编写子程序的方法
3. 掌握常规代码调试技巧
4. 理解编程者模型

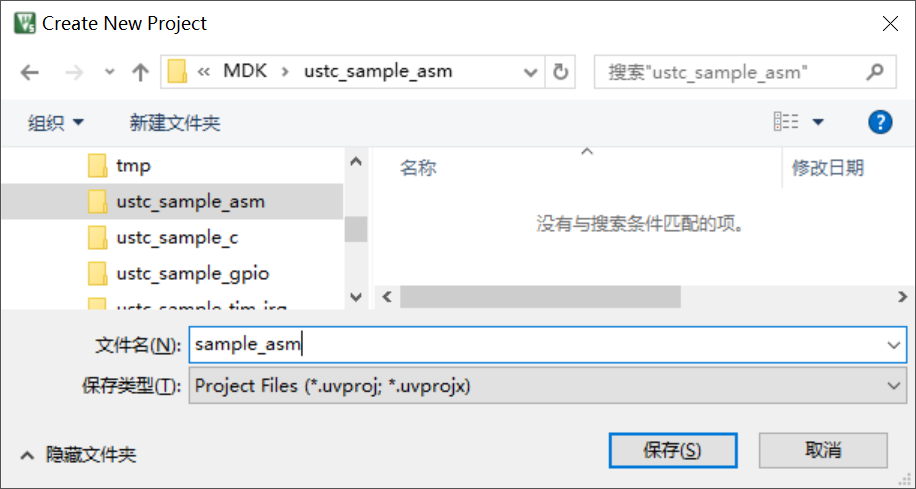
## 实验内容

### Project的建立、编译、连接

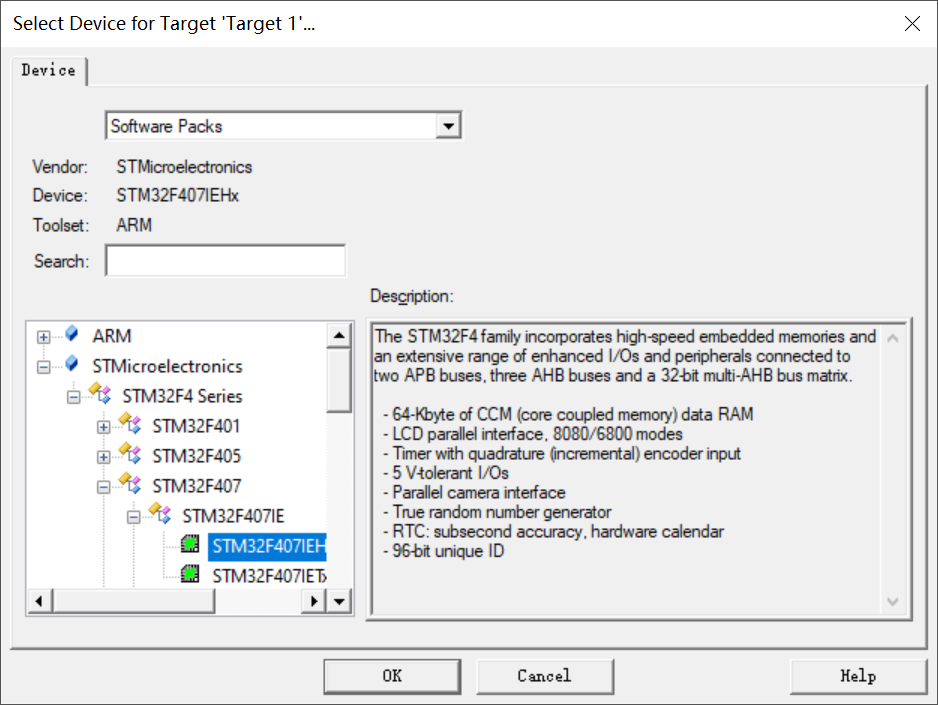
1. 通过IDE环境的主菜单“Project”🡪“New µVision Project”



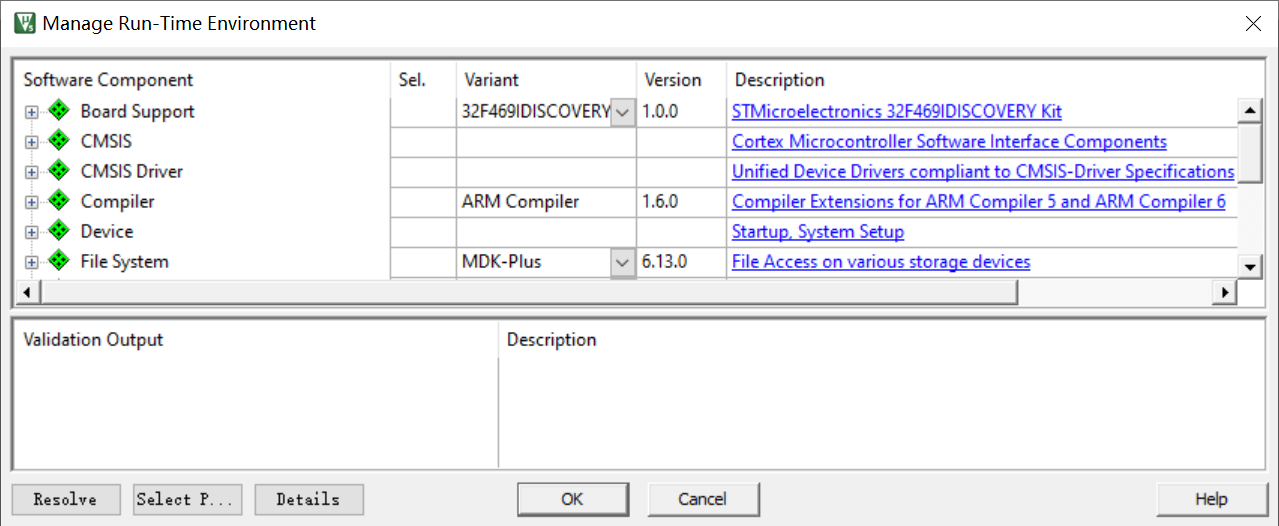
给新建的Project命名



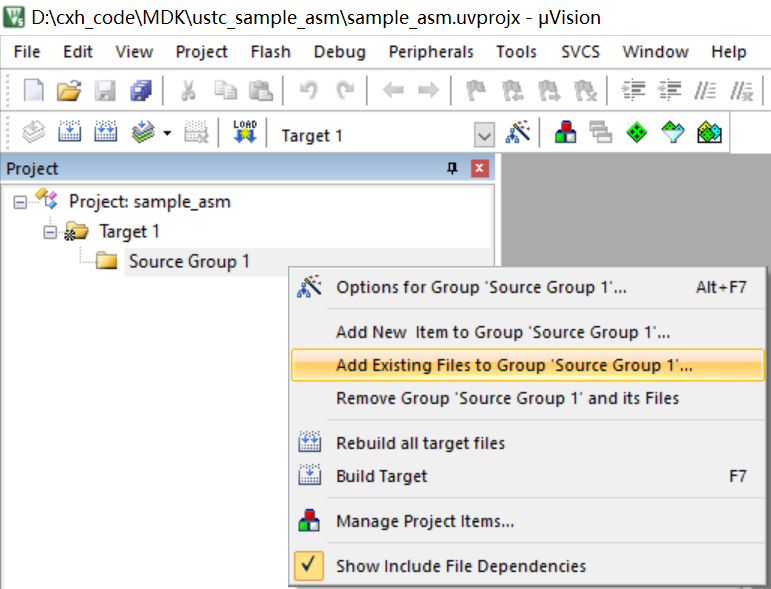
选择Device



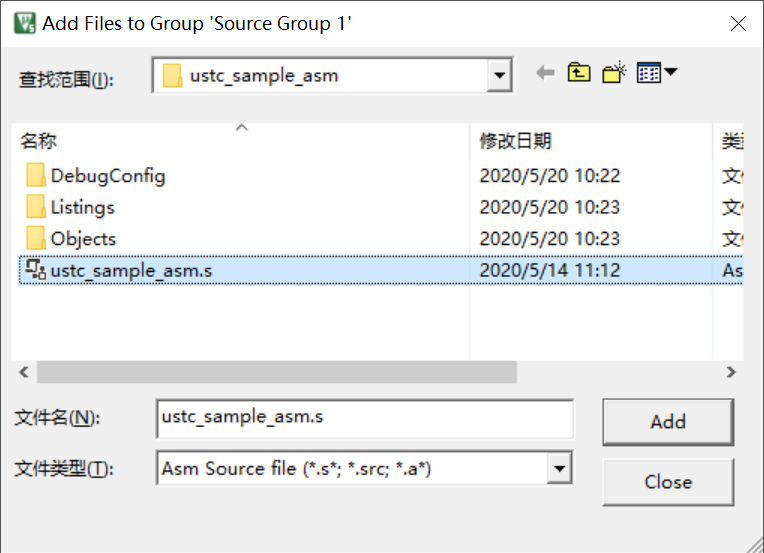
不对Project的软件库进行额外配置，单击“OK”



添加一个ASM文件至Project

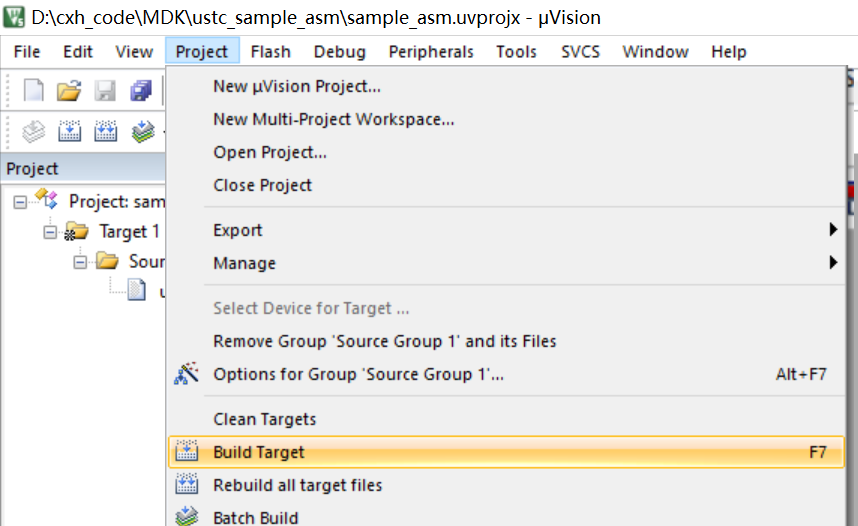


选择已有的ASM，“Add”



1. 编译、连接Project

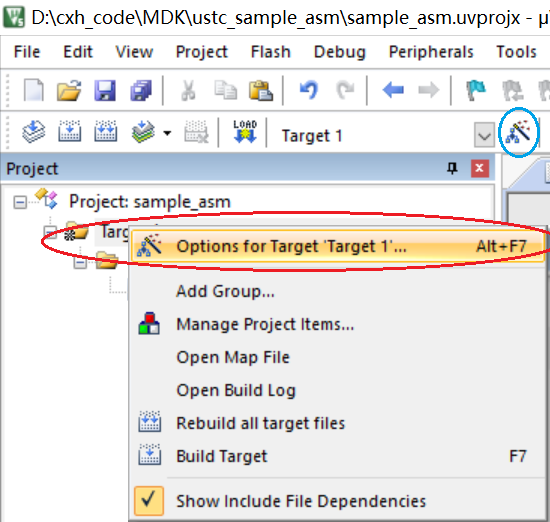
可从主菜单“Project”选择“Build Target”，或采用热键“F7”



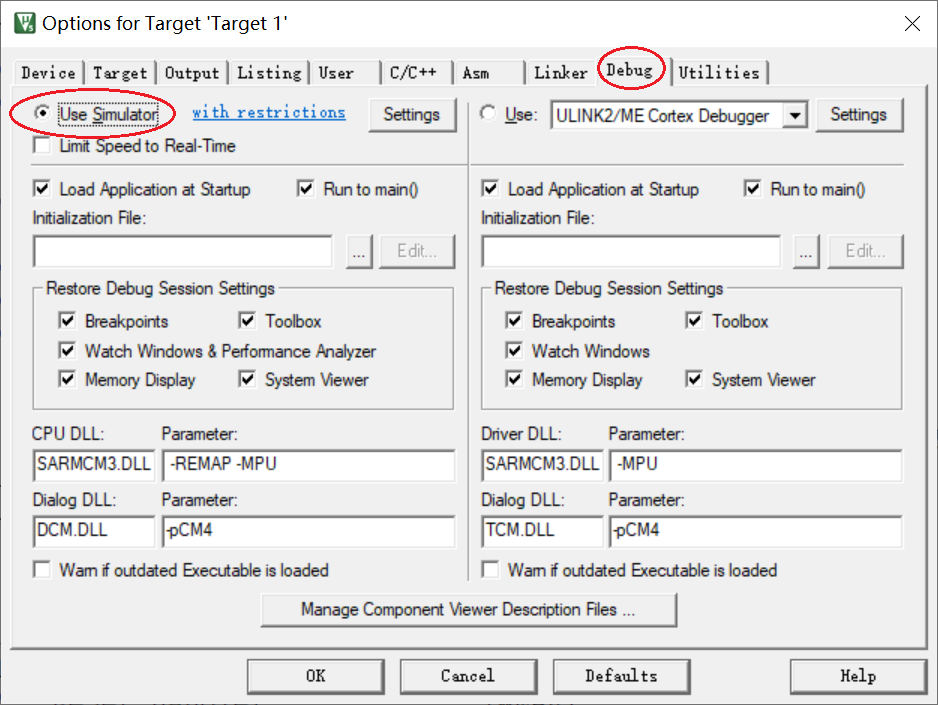
### Project的调试（Debug）

1. 调试

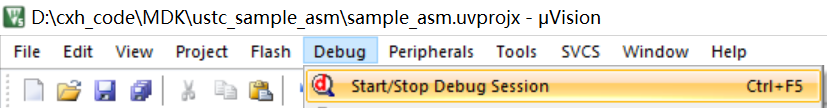
设置采用Simulator方式进行调试，先打开“Options for Target …”配置窗口



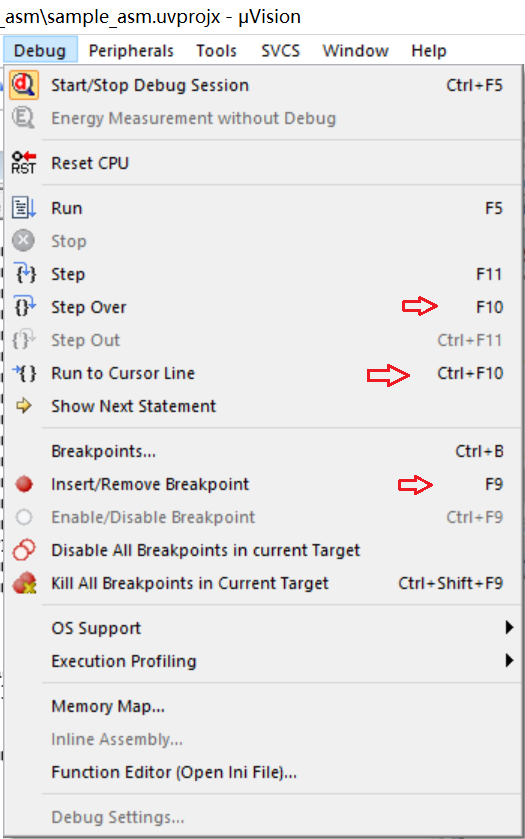
设置Debugger为“Simulator”



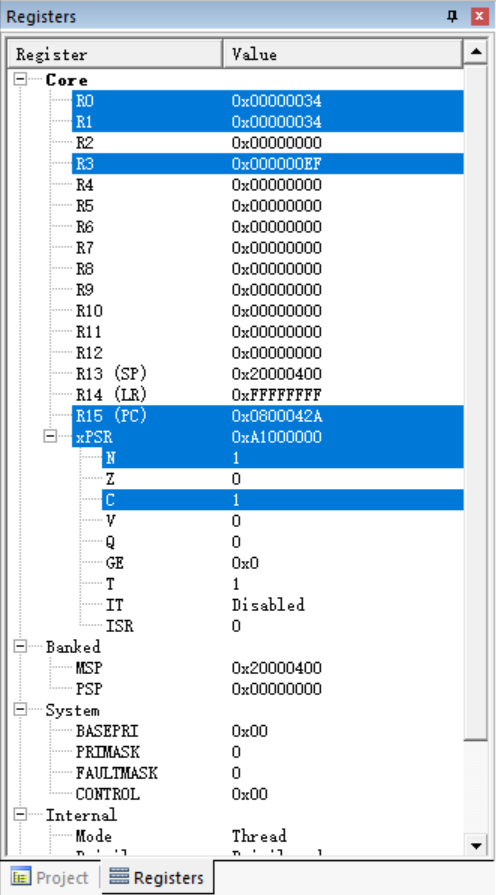
进入调试状态



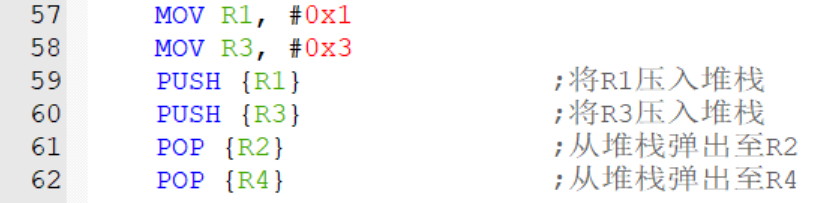
试验单步运行、运行到光标位置、设置断点等功能



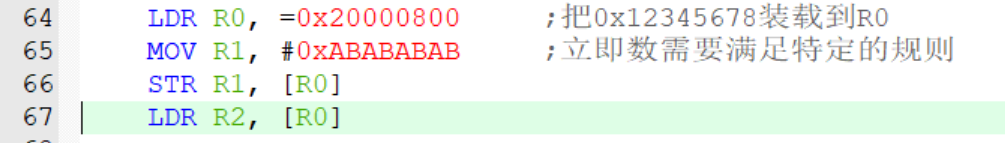
1. 观察调试过程中通用寄存器的变化



1. 观察如下代码执行前后栈指针的变化



1. 在如下代码执行前后观察存储器0x20000800地址的内容



## 思考题（内容需要上机调试才能完成）

1. 异常处理子程序Reset\_Handler的入口地址是？
2. 添加一行代码，使xPSR寄存器的Z标志位为1。
3. 示例代码中，为何使用的是MSP而不是PSP。
4. 请解释执行至Reset\_Handler中第一行代码时，为何MSP为“0x20000400”？
5. 请依据代码调试中观察到的机器指令解释伪指令“LDR R0, =0x20000800”被翻译为机器指令的执行过程？
6. 伪指令“LDR R0, =0x20000800”中数值“0x20000800”，被存放在哪个地址？
7. 解释指示符（伪指令）“EXPORT”和“DCD”的作用？
8. 观察代码执行过程中，PC变化的规律。并写出一条使PC递增2的指令，再写出一条使PC递增4的指令。

# 基于C的Project

## 实验目的

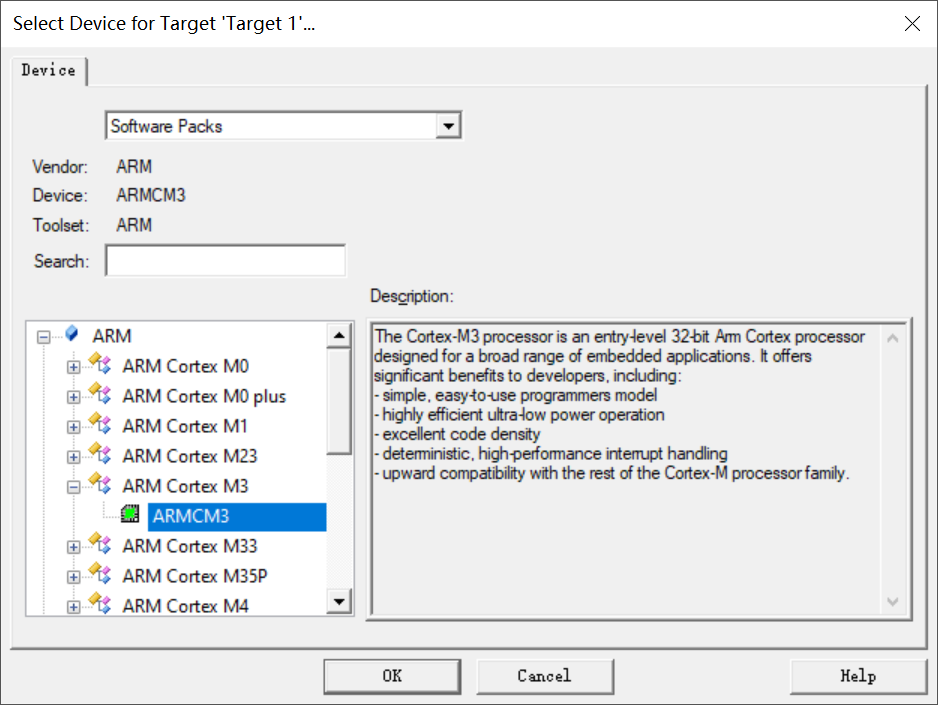
1. 掌握µVision IDE下创建C语言工程的基本步骤
2. 了解µVision IDE自带CMSIS库和device的启动文件
3. 掌握联机帮助查询技巧
4. 掌握代码分析技巧
5. 理解ARM汇编程序中的伪指令（指示符，Directive）
6. 掌握C和汇编混合编程方法

## 实验内容

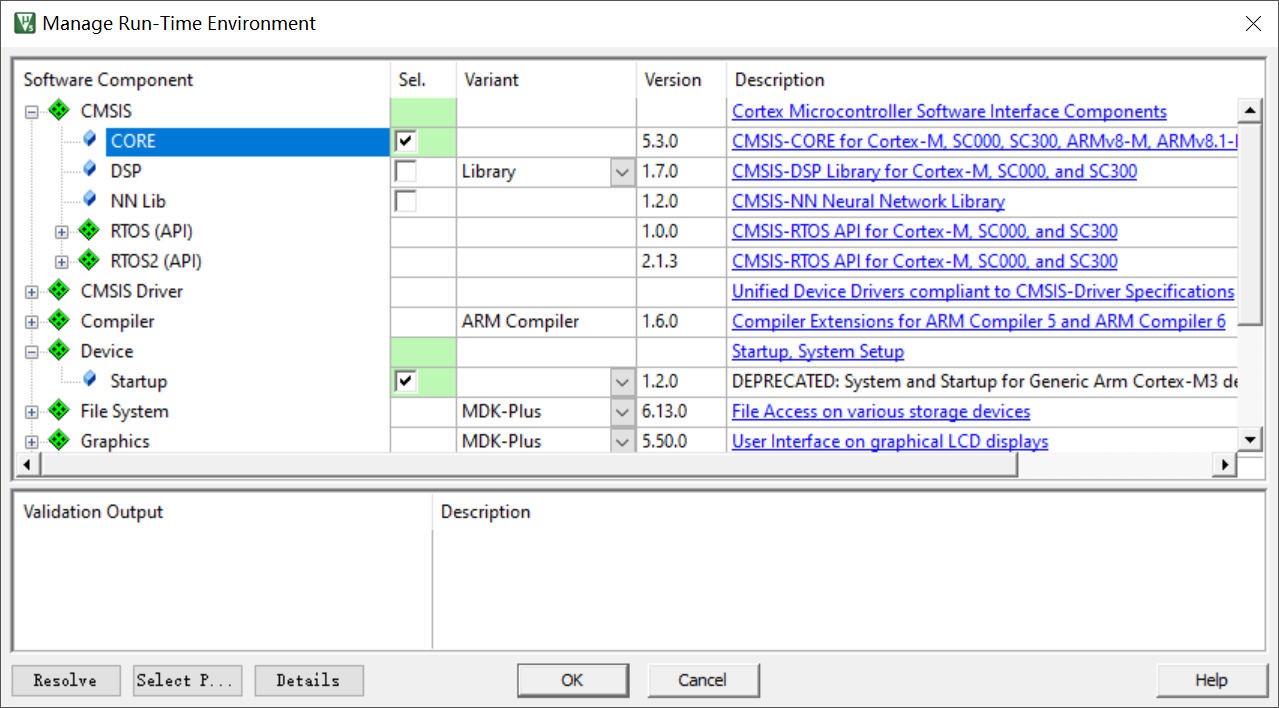
### 建立基于C程序的Project

1. 通过IDE环境的主菜单“Project”🡪“New µVision Project”创建

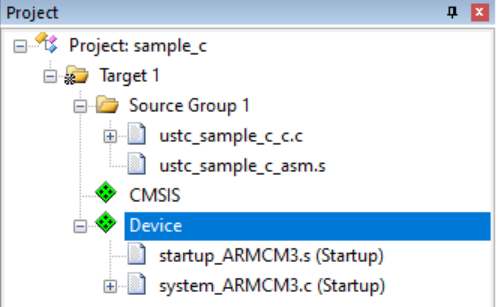
选择Device如下（如果选择了其他Device，后续界面显示会略有差异）



勾选Keil µVision自带的CMSIS库和对应Device的启动文件（Startup）



添加示例源代码“ustc\_sample\_c\_asm.s”和“ustc\_sample\_c\_c.c”至Project



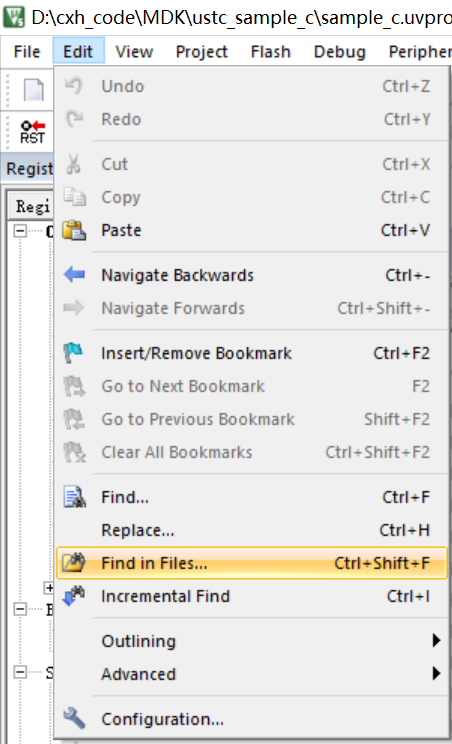
设置采用Simulator方式进行调试

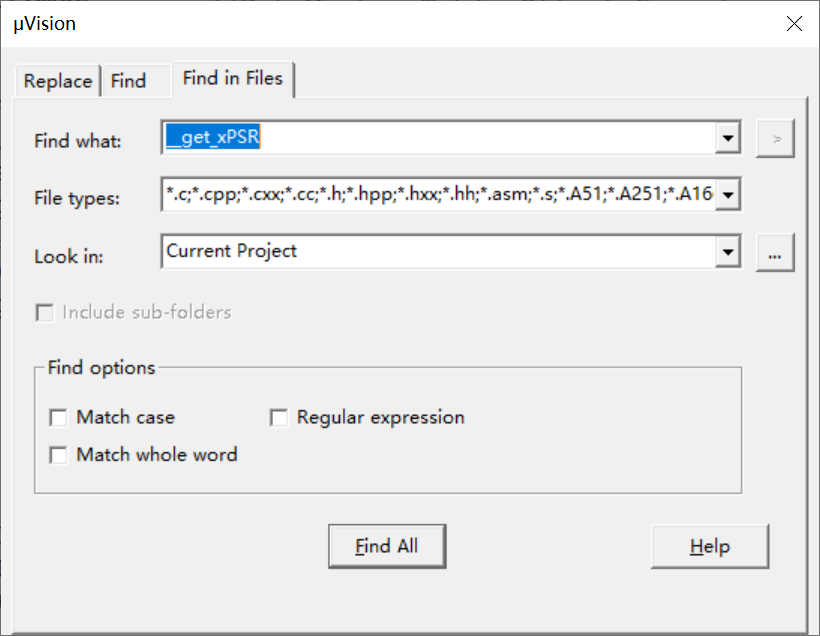
### 代码功能验证

1. 阅读启动文件“startup\_ARMCM3.s”和“startup\_ARMCM3.c”，分析启动的大致过程。
2. 分析示例代码中C调用汇编子程序的过程。
3. 分析示例代码中汇编调用C子程序的过程。

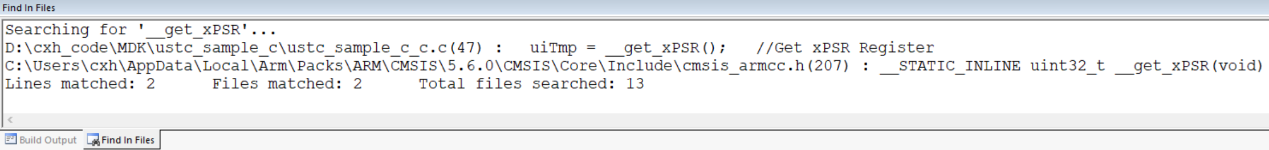
### 代码分析技巧

1. 在当前Project内所有文件中查找“\_\_get\_xPSR”关键字。

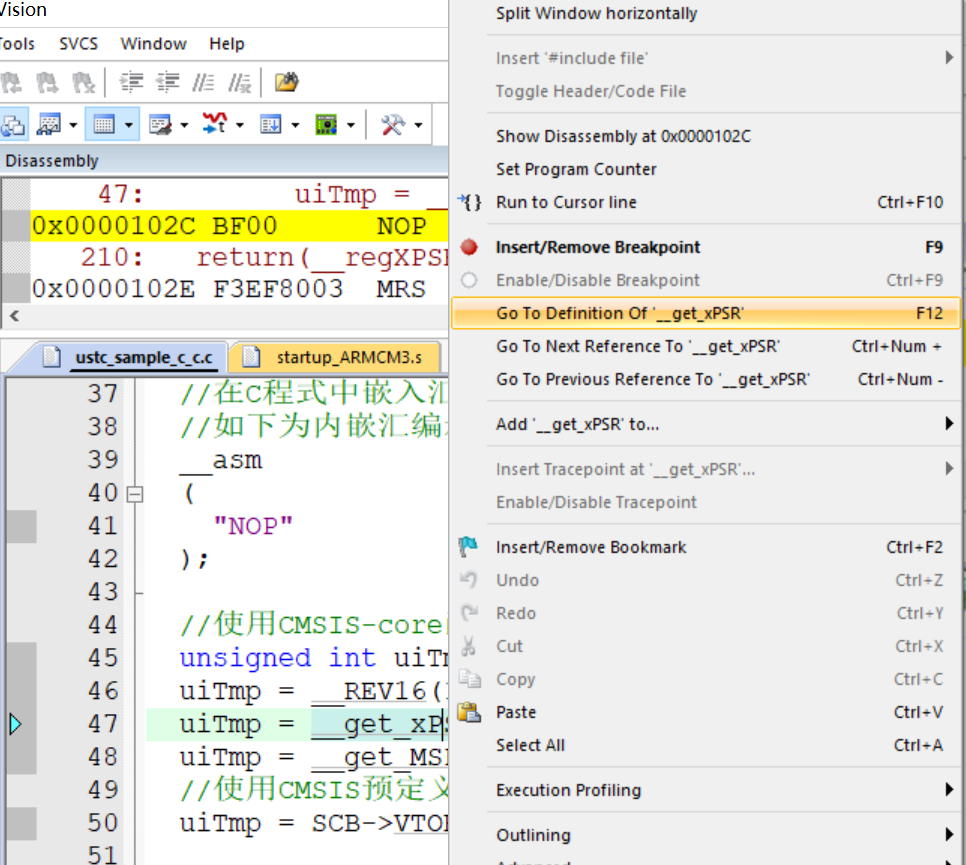




逐个文件查看关键字“\_\_get\_xPSR”出现的位置。

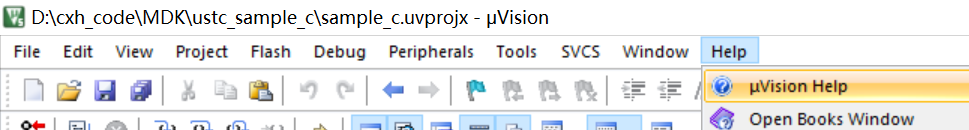


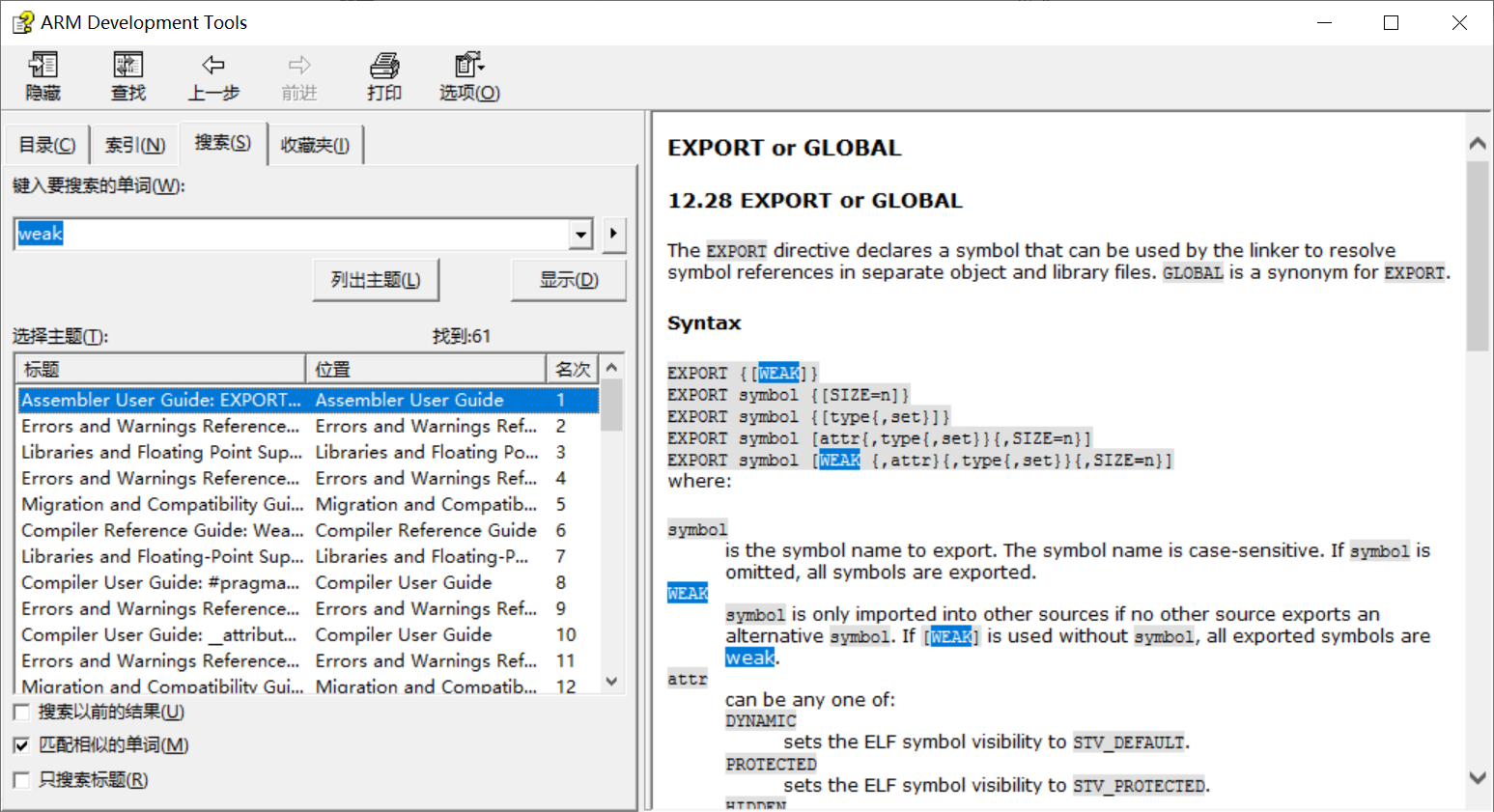
1. 分析示例代码中所调用的CMSIS-core函数的定义。如下图所示在函数位置点击鼠标右键（或热键F12），跳转至\_get\_xPSR()的定义文件。浏览所打开的“cmsis\_armcc.h”文件。



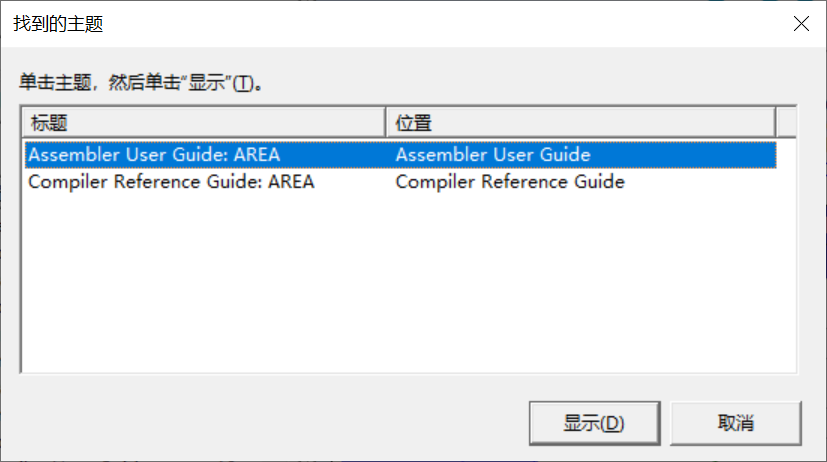
### µVision联机资源使用

1. 打开联机帮助文档，搜索指示符“WEAK”的作用。

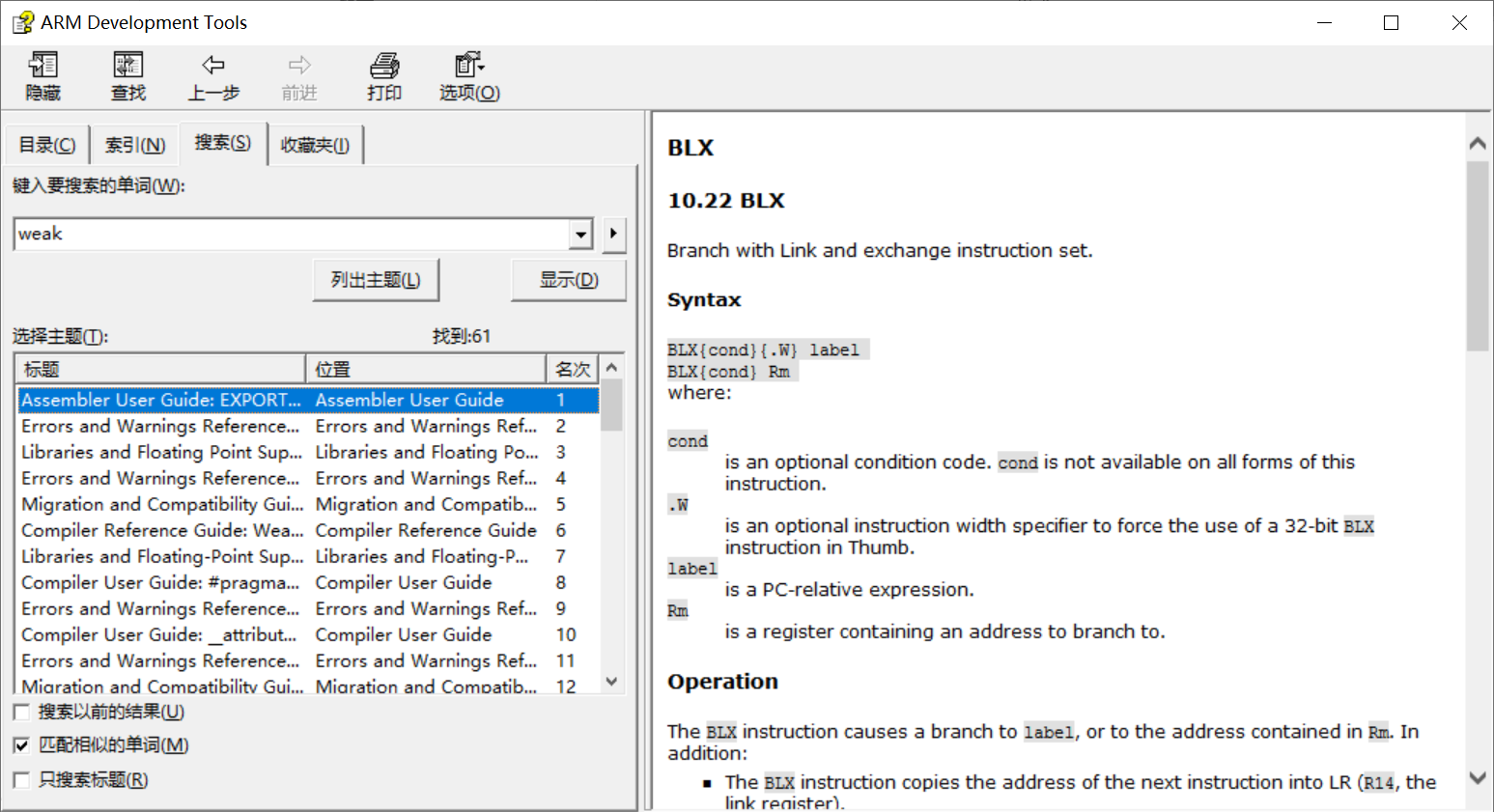




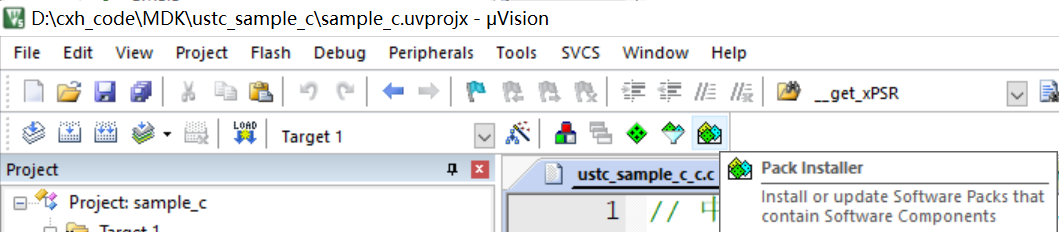
1. 将光标停留在某个指示符（伪指令关键字）上，如“AREA”，按键“F1”，查阅自动弹出的帮助信息。



1. 将光标停留在某条指令上，如“BLX”，按键“F1”，查阅自动弹出的帮助信息。

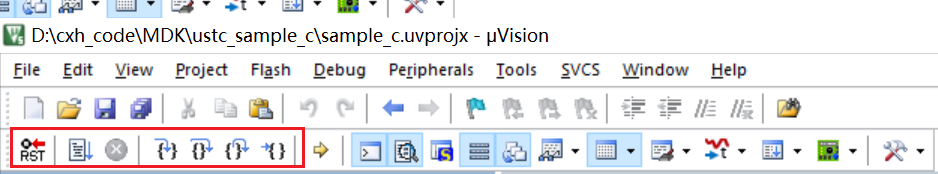


1. 了解包管理器“Pack installer”的基本功能。

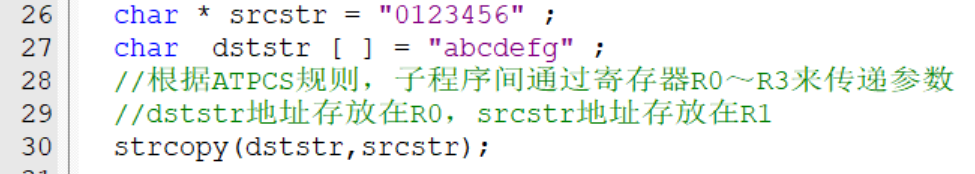


### Project的调试（Debug）

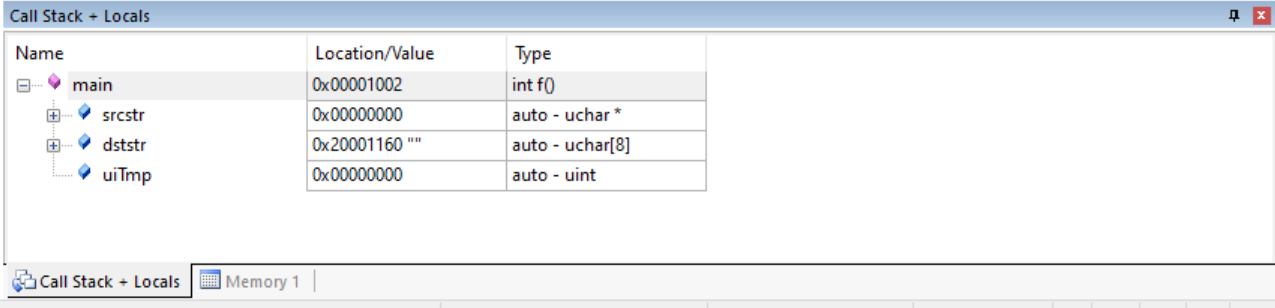
1. 通过试验分析下图所示Debug工具栏（红色框内）各个图标对应功能的区别。



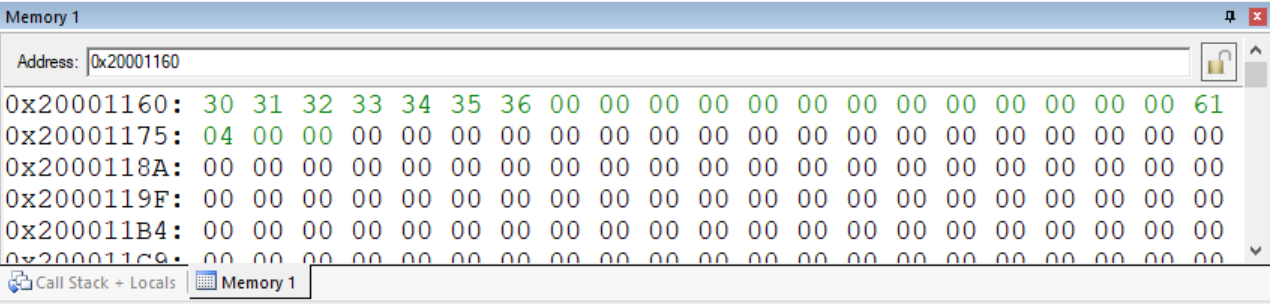
1. 通过Watch窗口观察一下代码执行前后变量值。



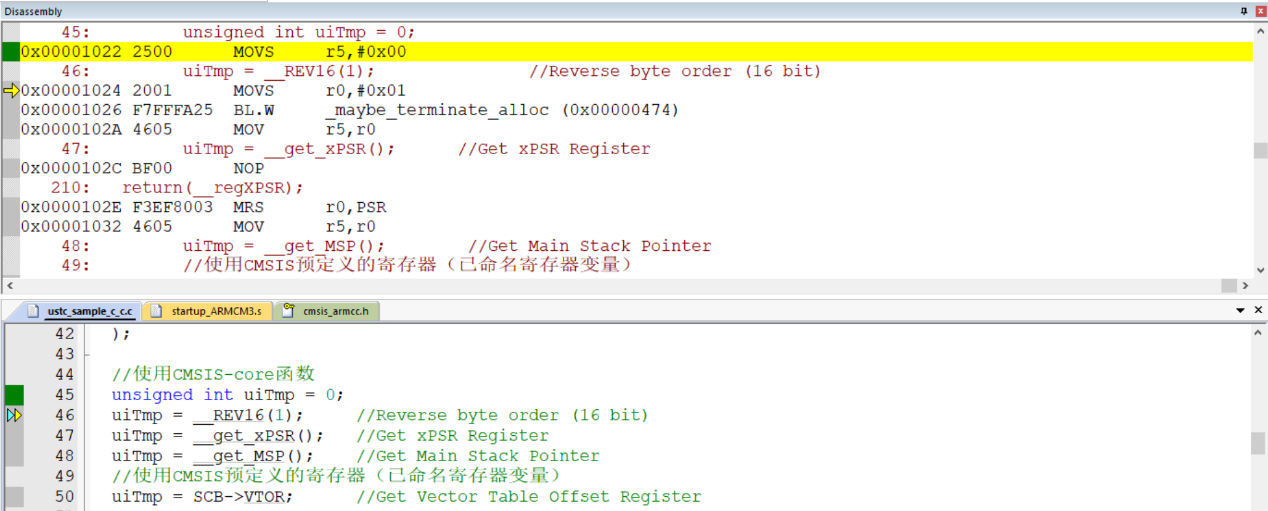
观察变量“srcstr”和“dststr”



同时观察Memory窗口对应地址的值

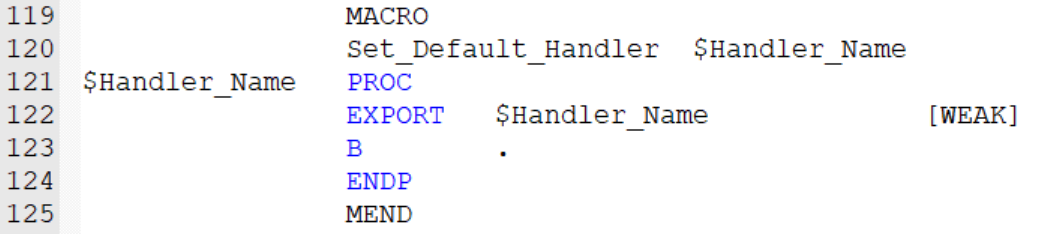


1. 分析下图所示反汇编窗口中机器指令与源代码窗口中C代码的对应关系。



## 思考题（内容需要上机调试才能完成）

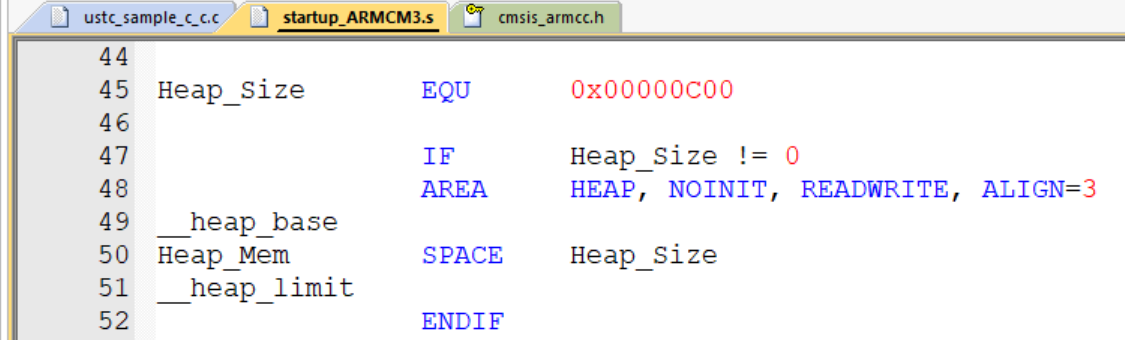
1. 分析启动文件“startup\_ARMCM3.s”中图下宏定义的含义？并写出$Handler\_Name等于NMI\_Handler时，该宏定义展开后的代码。



1. 依据调试结果，示例c程序中如下行中字符串"Hello USTCer\n"被保存在存储器的什么位置（写出存储器地址）？



1. 请结合“startup\_ARMCM3.s”文件中如下代码分析示例中“srcstr”的地址为何是“0x20001160”？



# 基于STM32库的GPIO与定时器

## 实验目的

1. 掌握µVision IDE中基于ST公司STM32库建立project的流程
2. 了解ST公司提供的TIM、GPIO相关库函数
3. 了解STM32F10X系列芯片定时器相关的寄存器功能。
4. 掌握µVision IDE中外设仿真模块（GPIO）的使用
   1. 学会利用外设仿真模块（GPIO）观察I/O引脚输出
   2. 学会利用外设仿真模块（GPIO）模拟I/O引脚的输入
5. 掌握µVision IDE逻辑分析模块（Logic Analyzer）的使用

## 实验内容

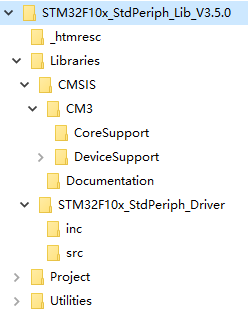
### 下载ST公司STM32库及芯片手册

1. 下载ST公司关于STM32F103系列芯片的库文件

<https://www.st.com/en/embedded-software/stm32-standard-peripheral-libraries.html>

详细参阅0说明。

库文件压缩包结构如下（其中libraries子目录即库文件）



1. 下载ST公司关于STM32F103系列芯片的文档

<https://www.stmcu.org.cn/document/detail/index/id-200272>

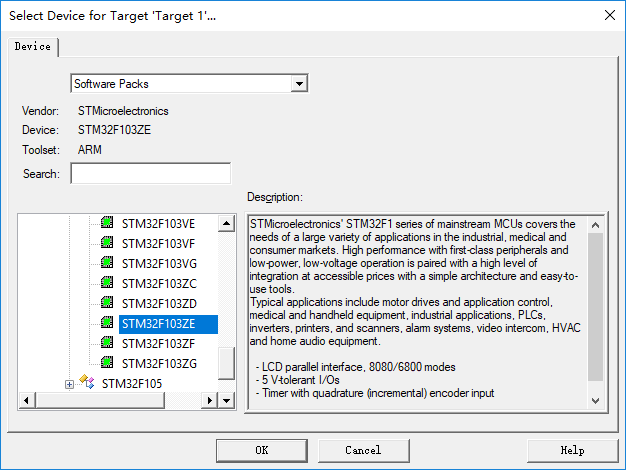
详细参阅0说明。

浏览芯片参考手册的目录，后续实验项目需要查阅本手册。

### 建立基于STM32库的Project

1. 通过IDE环境的主菜单“Project”🡪“New µVision Project”

选择STM32F10X系列的设备（图示STM32F103ZE为keil公司simulation支持的芯片），且不添加µVision自带的任何库函数。



1. 拷贝ST公司的库文件

将ST公司的库文件压缩包中下列内容拷贝至project目录下：

“Libraries”子目录；

“.\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\stm32f10x\_conf.h”文件；

“.\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\system\_stm32f10x.c”文件；

1. 添加ST公司的库文件

添加“system\_stm32f10x.c”文件至Project。

添加示例代码“sample\_gpio\_main.c”文件至Project。

添加如下ST库文件至Project：

“.\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm\startup\_stm32f10x\_hd.s”

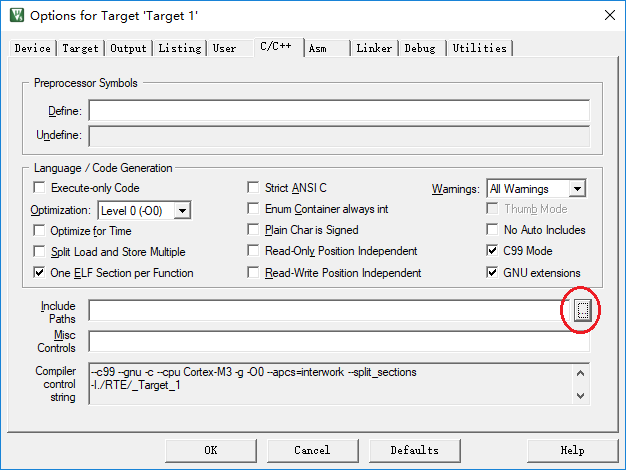
“.\Libraries\STM32F10x\_StdPeriph\_Driver\src\stm32f10x\_gpio.c”

“.\Libraries\STM32F10x\_StdPeriph\_Driver\src\stm32f10x\_rcc.c”

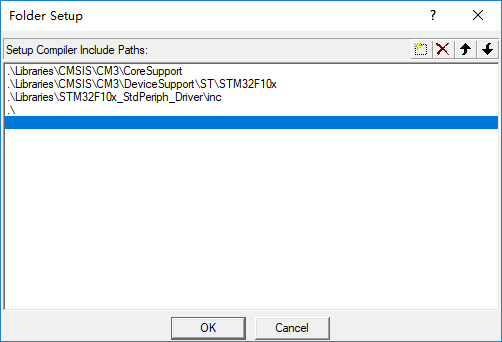
“.\Libraries\STM32F10x\_StdPeriph\_Driver\src\stm32f10x\_tim.c”

### 配置Project的头文件目录、预编译参数、Simulator

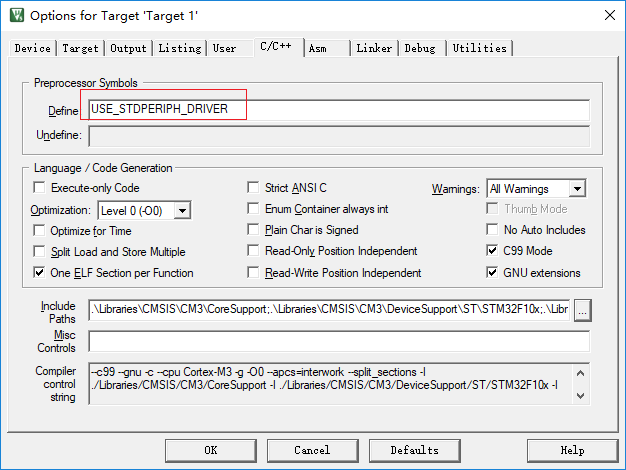
1. 配置“include file”路径执行ST库头文件目录



添加如下路径至project的头文件搜索路径



1. 配置预编译选项“Preprocessor Symbols”



备注：如果不配置此项，应该会出现错误

.\Objects\sample\_gpio.axf: Error: L6218E: Undefined symbol assert\_param (referred from stm32f10x\_gpio.o).

1. 设置采用Simulator方式进行调试，配置为STM32F103ZE芯片的simulator（如果不配置此项，虽然编译连接能通过，但是无法正常在simulator方式下Debug）



1. 编译、链接（Build Project）

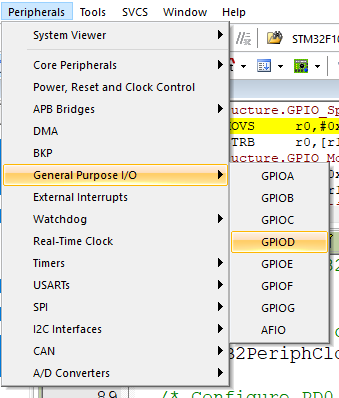
如果Build Project出现错误，

FCARM - Output Name not specified, please check 'Options for Target - Utilities'

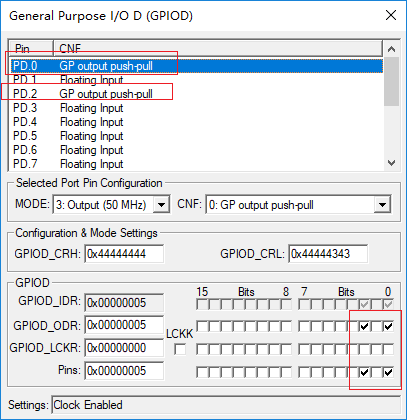
请参考0小节描述。

### Debug时使用外设仿真功能验证GPIO输入和输出

1. CTRL+F5进入调试状态，打开GPIO窗口。

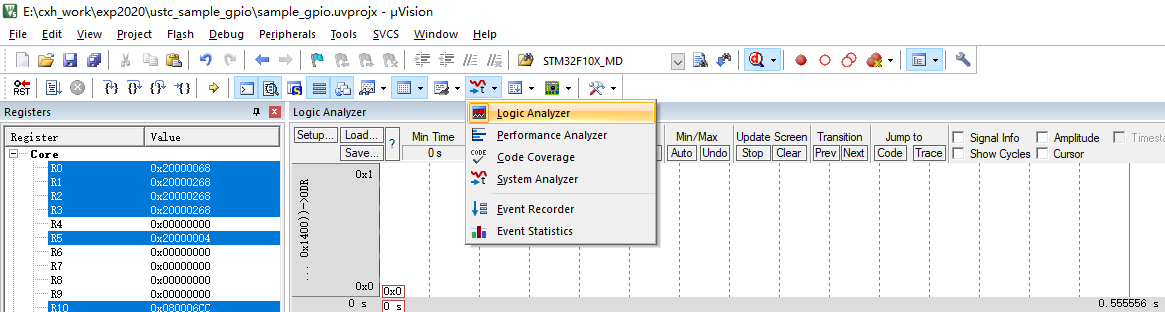


观察示例代码中一下代码行执行前后GPIOD的变化



### Debug时使用Logic Analyzer观察GPIO输出

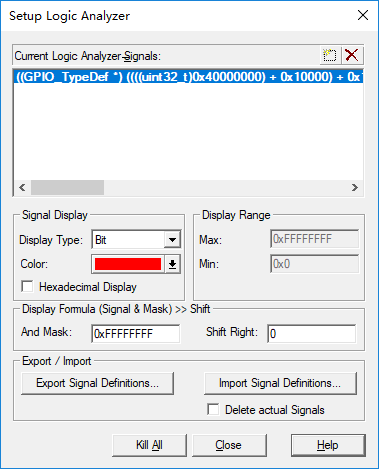
1. 通过µVision IDE自带的逻辑分析模块（Logic Analyzer）观察GPIOD的变化。



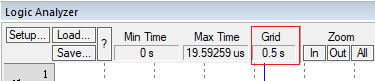
在上图Logic Analyzer窗口中单击“Setup”设置GPIOD->ODR为需要观察的信号



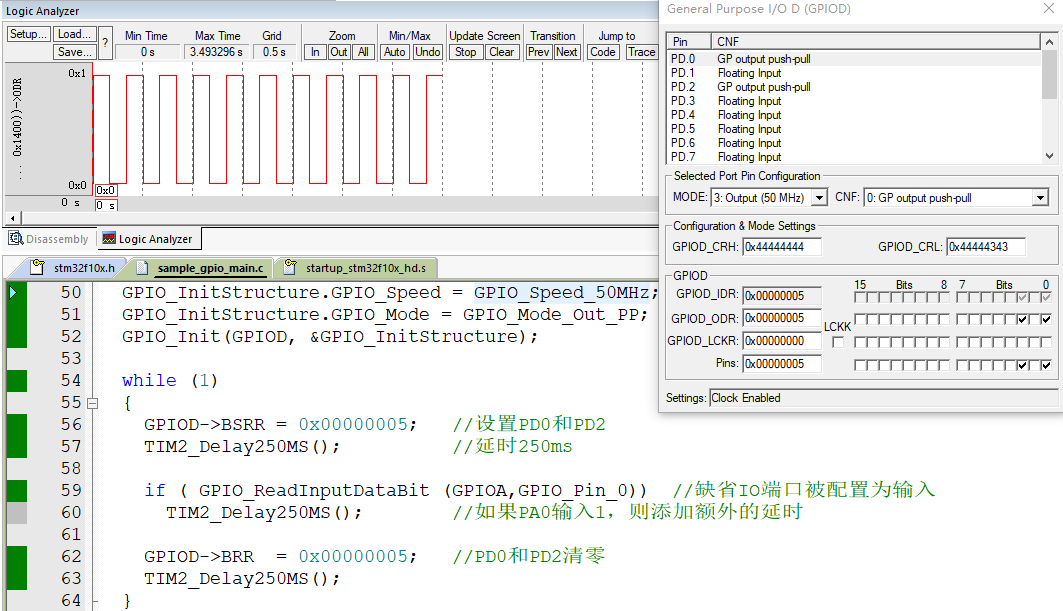
并设置GPIOD->ODR信号的显示类型（Display Type）为“Bit”



注意：Logic Analyzer的显示比例要合适（下图中Grid大小），可通过鼠标滚轮缩放。



F5运行示例代码，应该可以在Logic Analyzer窗口和GPIOD窗口同时观察到PD0和PD5位的变化。

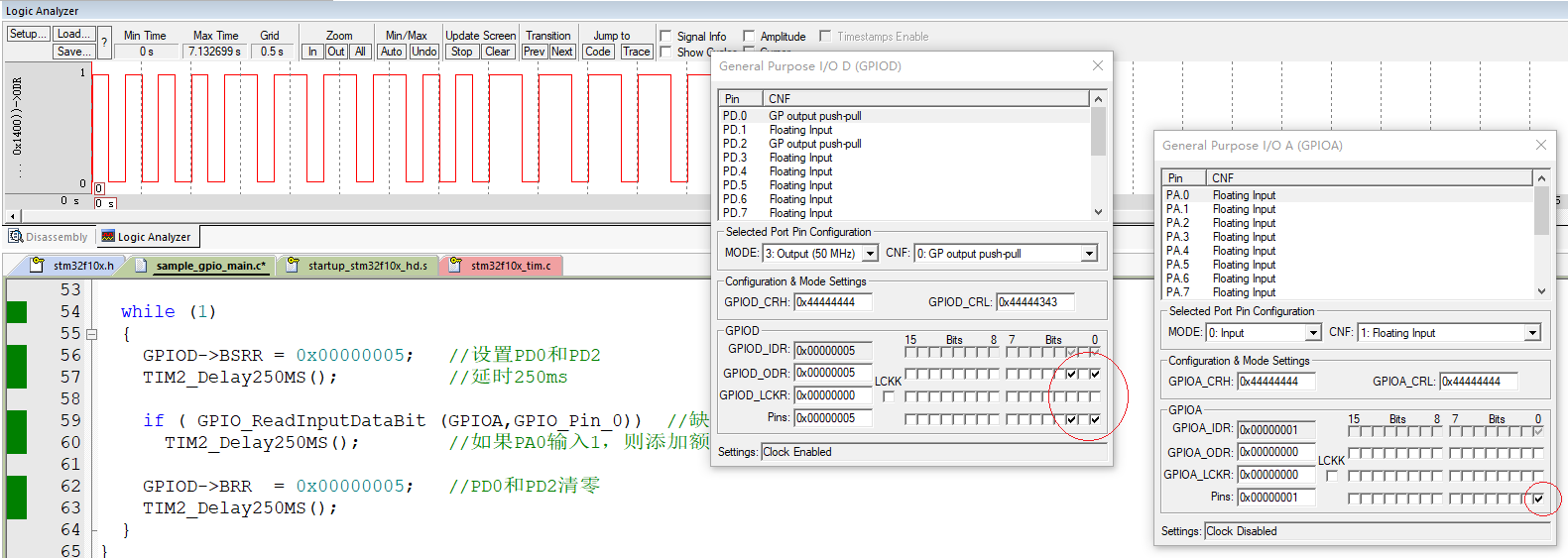


### 使用外设仿真和Logic Analyzer验证GPIO输入和输出

1. 再打开GPIOA的窗口（Peripherals🡪General purpose I/O🡪GPIOA）

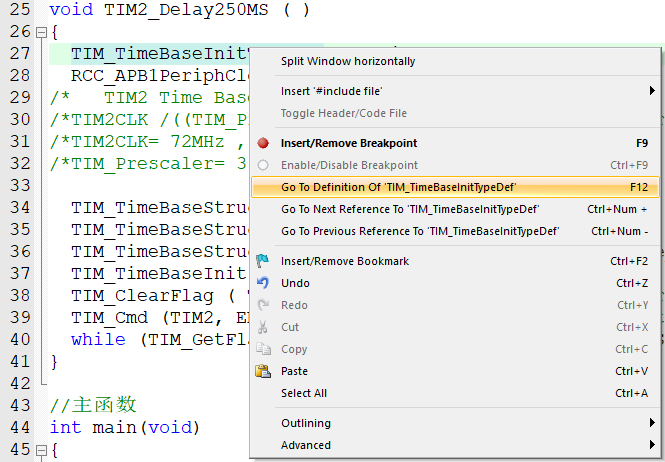


通过再GPIOA窗口单击PA0引脚位置，模拟PA0输入“1”，观察Logic Analyzer的变化，并与示例代码进行验证。



### 定时器的配置

1. 在代码中“TIM\_TimeBaseInitTypeDef”所在位置点击鼠标邮件（或F12）跳转到TIM\_TimeBaseInitTypeDef的定义文件。



阅读代码的注释信息，了解该结构体各成员的含义。

typedef struct

{

uint16\_t TIM\_Prescaler;

uint16\_t TIM\_CounterMode;

uint16\_t TIM\_Period;

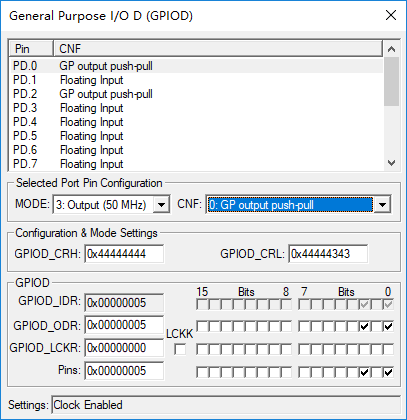
uint16\_t TIM\_ClockDivision;

uint8\_t TIM\_RepetitionCounter;

} TIM\_TimeBaseInitTypeDef;

## 思考题（部分内容需要上机调试才能完成）

1. 阅读讲义8.5.3小节，及STM32芯片手册，解释下图中“CRL、CRH、IDR、ODR、LCKR”几个寄存器的作用。



1. 解释代码“GPIOD->BSRR = 0x00000085; ”的作用。
2. 解释代码“GPIOD->BRR = 0x00000080;”的作用。
3. 解释库函数“GPIO\_ReadInputDataBit (GPIOD,GPIO\_Pin\_0)”的作用。
4. \*访问“http://www.keil.com/support/docs/3726.htm”，查询µVision IDE是否支持STM32F407ZG芯片的Simulation。
5. \*阅读STM32F103芯片手册，了解“stm32f10x.h”文件中如下结构体各成员所对应寄存器的功能。

typedef struct

{

\_\_IO uint32\_t CR;

\_\_IO uint32\_t CFGR;

\_\_IO uint32\_t CIR;

\_\_IO uint32\_t APB2RSTR;

\_\_IO uint32\_t APB1RSTR;

\_\_IO uint32\_t AHBENR;

\_\_IO uint32\_t APB2ENR;

\_\_IO uint32\_t APB1ENR;

\_\_IO uint32\_t BDCR;

\_\_IO uint32\_t CSR;

} RCC\_TypeDef;

# 基于STM32库的中断

## 实验目的

1. 掌握EXTI中断配置流程
2. 理解异常向量表
3. 理解异常优先级配置
4. 了解ST公司提供的TIM、NVIC相关库函数
5. 掌握µVision IDE中外设仿真模块（NVIC）的使用

## 实验内容

### 建立基于STM32库的Project

1. 通过IDE环境的主菜单“Project”🡪“New µVision Project”

选择STM32F10X系列的设备（图示STM32F103ZE为keil公司simulation支持的芯片），且不添加µVision自带的任何库函数。

1. 拷贝ST公司的库文件

将ST公司的库文件压缩包中下列内容拷贝至project目录下：

“Libraries”子目录；

“.\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\stm32f10x\_conf.h”文件；

“.\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\system\_stm32f10x.c”文件；

1. 添加ST公司的库文件

添加示例代码“sample\_nvic\_main.c”文件至Project。

添加示例代码“sample\_nvic\_exeption\_handler.c”文件至Project。

添加如下ST库文件至Project：

“.\system\_stm32f10x.c”

“.\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm\startup\_stm32f10x\_hd.s”

“.\Libraries\STM32F10x\_StdPeriph\_Driver\src\stm32f10x\_gpio.c”

“.\Libraries\STM32F10x\_StdPeriph\_Driver\src\stm32f10x\_rcc.c”

“.\Libraries\STM32F10x\_StdPeriph\_Driver\src\stm32f10x\_exit.c”

“.\Libraries\STM32F10x\_StdPeriph\_Driver\src\misc.c”

### 配置Project的头文件目录、预编译参数、Simulator

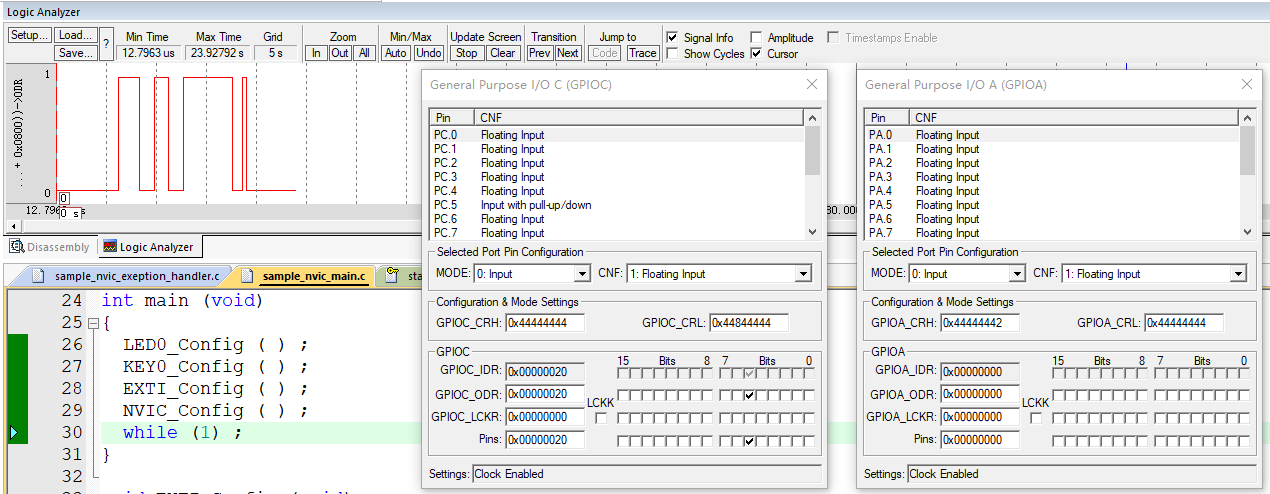
1. 配置“include file”路径执行ST库头文件目录

.\Libraries\CMSIS\CM3\CoreSupport;.\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x;.\Libraries\STM32F10x\_StdPeriph\_Driver\inc;.\

1. 配置预编译选项“Preprocessor Symbols”为：USE\_STDPERIPH\_DRIVER
2. 设置采用Simulator方式进行调试，配置为STM32F103ZE芯片的simulator，具体配置同上一个实验

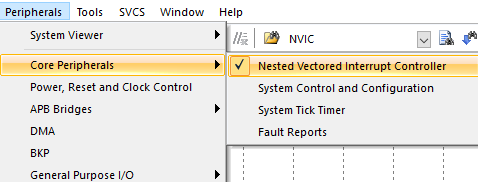
### 使用Logic Analyzer和外设仿真功能验证EXIT及GPIO输出

1. 打开µVision IDE自带的逻辑分析模块（Logic Analyzer），单击“Setup”设置GPIOA->ODR为需要观察的信号。并设置GPIOA->ODR信号的显示类型（Display Type）为“Bit”。
2. 打开GPIOA的窗口（Peripherals🡪General purpose I/O🡪GPIOA）。
3. 打开GPIOC的窗口（Peripherals🡪General purpose I/O🡪GPIOC）。
4. F5运行示例代码，在GPIOC窗口可以输入PC5（模拟中断信号）。随着PC5信号的变化，可以在Logic Analyzer窗口和GPIOA窗口同时观察到PA8的变化。

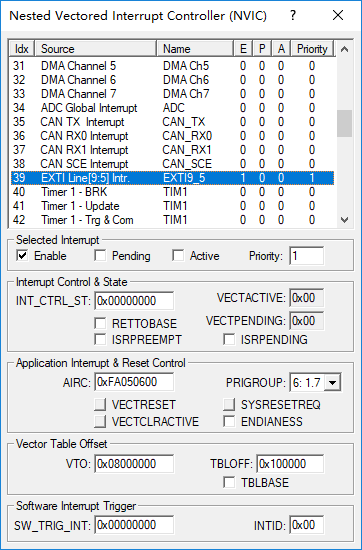


### 观察NVIC寄存器组

1. 打开（Peripherals🡪Core Periperals🡪Nested Vectored Interrupt Controller）

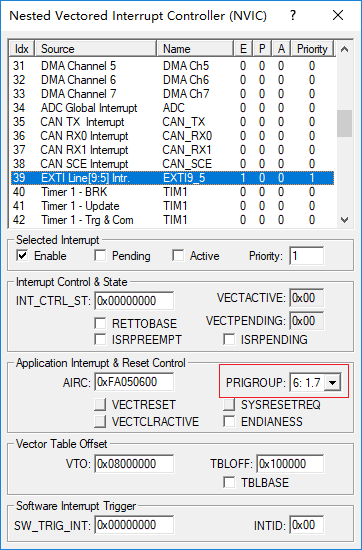


观察与外部中断#5有关的寄存器信息。



## 思考题（内容需要上机调试才能完成）

1. 为什么通过GPIOC的窗口（Peripherals🡪General purpose I/O🡪GPIOC）模拟PC5（中断信号）输入的时候，改变两次PC5后PA8才会发生变化？
2. 请通过调试获得EXTI9\_5\_IRQHandler()的入口地址（应该是0x800026A），这个地址保存在在异常向量表什么位置？
3. 解释下图中PRIGROUP的含义（请查阅讲义5.5.1小节），PRIGROUP和AIRC（Application Interrupt & Reset Control寄存器）是什么关系？

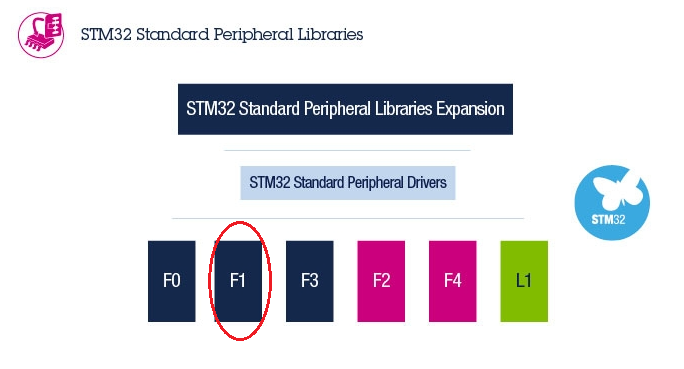


# 附录1：相关资料下载

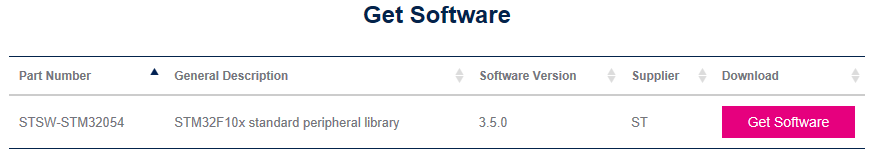
## ST公司STM32库及芯片手册

下载ST公司关于STM32F103系列芯片的库文件

<https://www.st.com/en/embedded-software/stm32-standard-peripheral-libraries.html>



跳转至<https://www.st.com/content/st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32-standard-peripheral-libraries/stsw-stm32054.html>



或者直接下载

<https://www.st.com/content/ccc/resource/technical/software/firmware/48/ab/e5/17/0d/79/43/74/stsw-stm32054.zip/files/stsw-stm32054.zip/jcr:content/translations/en.stsw-stm32054.zip>

## ST公司芯片手册

下载ST公司关于STM32F103系列芯片的文档

<https://www.stmcu.org.cn/document/detail/index/id-200272>



## Keil5 手动安装STM32 芯片包

<https://www.keil.com/dd2/Pack/>

可以参阅

[www.360doc.com/content/19/0923/23/66005518\_862827404.shtml](http://www.360doc.com/content/19/0923/23/66005518_862827404.shtml)

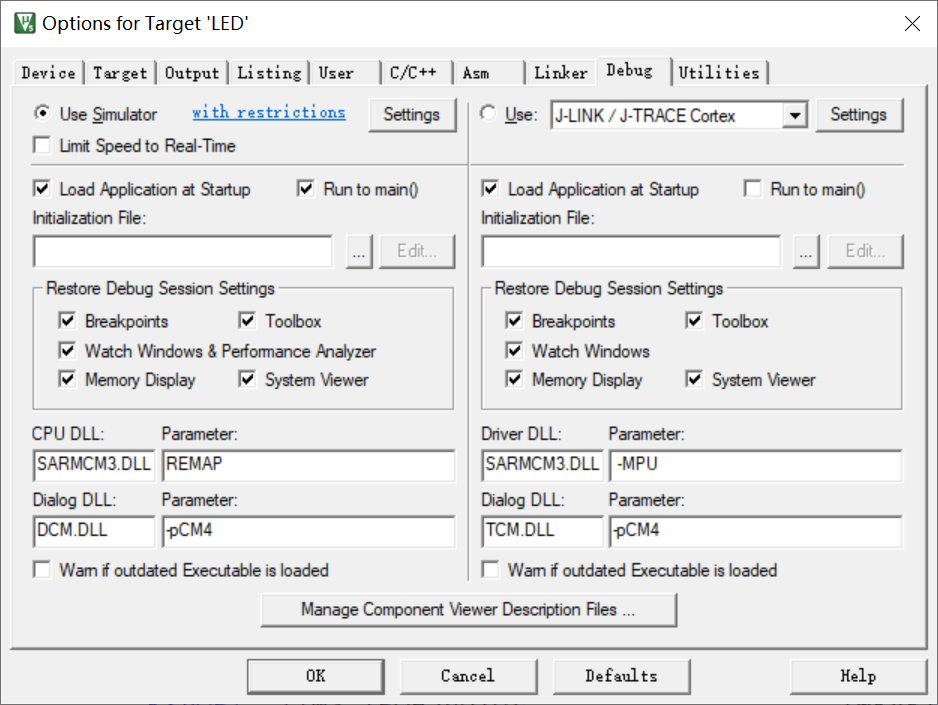
# 附录2：µVision5 IDE常见错误

## 使用Simulator调试程序时，access violation

\*\*\* error 65: access violation at 0x0000000C : no 'read' permission

往往是仿真时未配置具体芯片的simulator导致的错误

<https://blog.csdn.net/he__yuan/article/details/81712267>



DARMSTM.DLL

-pSTM32F407ZGTx

\*\*\* error 65: access violation at 0xFFFFFFF4 : no 'write' permission

<https://blog.csdn.net/Maple_Leaf_15/article/details/51057991>

## µVISION支持的Simulation

µVISION DEBUGGER: Simulation of Cortex-M Devices

<http://www.keil.com/support/docs/3726.htm>

**Peripheral Simulation for STMicroelectronics STM32F1 Series**

Dialog DLL | Parameter | Microcontroller Device

------------|-----------------|------------------------------------

DARMSTM.DLL | -pSTM32F101C6 | [STM32F101C6](http://www.keil.com/dd/chip/4250.htm)

DARMSTM.DLL | -pSTM32F101C8 | [STM32F101C8](http://www.keil.com/dd/chip/4249.htm)

DARMSTM.DLL | -pSTM32F101CBT6 | [STM32F101CBT6](http://www.keil.com/dd/chip/4248.htm)

DARMSTM.DLL | -pSTM32F101R6 | [STM32F101R6](http://www.keil.com/dd/chip/4247.htm)

DARMSTM.DLL | -pSTM32F101R8 | [STM32F101R8](http://www.keil.com/dd/chip/4246.htm)

DARMSTM.DLL | -pSTM32F101RB | [STM32F101RB](http://www.keil.com/dd/chip/4245.htm)

DARMSTM.DLL | -pSTM32F101T6 | [STM32F101T6](http://www.keil.com/dd/chip/4399.htm)

DARMSTM.DLL | -pSTM32F101T8 | [STM32F101T8](http://www.keil.com/dd/chip/4400.htm)

DARMSTM.DLL | -pSTM32F101V8 | [STM32F101V8](http://www.keil.com/dd/chip/4243.htm)

DARMSTM.DLL | -pSTM32F101VB | [STM32F101VB](http://www.keil.com/dd/chip/4242.htm)

DARMSTM.DLL | -pSTM32F103C6 | [STM32F103C6](http://www.keil.com/dd/chip/4236.htm)

DARMSTM.DLL | -pSTM32F103C8 | [STM32F103C8](http://www.keil.com/dd/chip/4235.htm)

DARMSTM.DLL | -pSTM32F103CB | [STM32F103CB](http://www.keil.com/dd/chip/4401.htm)

DARMSTM.DLL | -pSTM32F103R6 | [STM32F103R6](http://www.keil.com/dd/chip/4233.htm)

DARMSTM.DLL | -pSTM32F103R8 | [STM32F103R8](http://www.keil.com/dd/chip/4232.htm)

DARMSTM.DLL | -pSTM32F103RB | [STM32F103RB](http://www.keil.com/dd/chip/4231.htm)

DARMSTM.DLL | -pSTM32F103T6 | [STM32F103T6](http://www.keil.com/dd/chip/4402.htm)

DARMSTM.DLL | -pSTM32F103T8 | [STM32F103T8](http://www.keil.com/dd/chip/4403.htm)

DARMSTM.DLL | -pSTM32F103V8 | [STM32F103V8](http://www.keil.com/dd/chip/4224.htm)

DARMSTM.DLL | -pSTM32F103VB | [STM32F103VB](http://www.keil.com/dd/chip/4223.htm)

DARMSTM.DLL | -pSTM32F103ZE | [STM32F103ZE](http://www.keil.com/dd/chip/4216.htm)

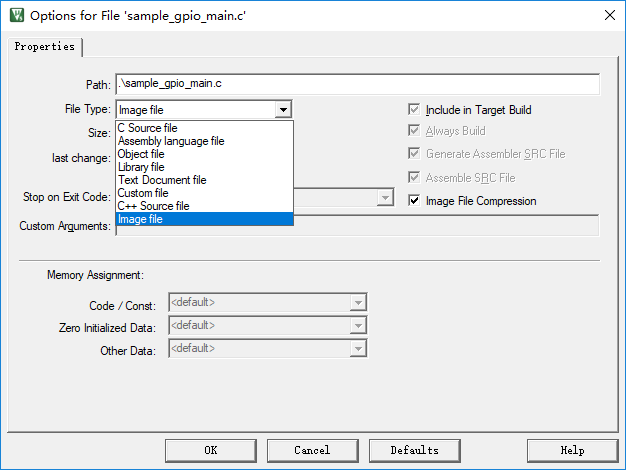
µVISION DEBUGGER: error 65: access violation...: no 'read' permission

<http://www.keil.com/support/docs/4045.htm>

## FCARM - Output Name not specified

FCARM - Output Name not specified, please check 'Options for Target - Utilities'

当keil 中出现：FCARM - Output Name not specified, please check ‘Options for Target - Utilities’这样的错误提示时，很有可能是加入文件时文件类型属性不正确（引起原因不明）。这时可点击每个文件查看其文件类型属性，如下图所示：



在file Type框中选择正确的属性，如果是C文件，选择”C source file”即可。

## Undefined symbol assert\_param

.\Objects\test2.axf: Error: L6218E: Undefined symbol assert\_param (referred from stm32f10x\_gpio.o).

在Options->C/C++->preprocessor Symboles有Define:项中输入USE\_STDPERIPH\_DRIVER，重新编译链接，无警告无错误。

## 帮助仅显示µVision User's Guide

在没有打开任何Project的情形下，帮助信息仅显示µVision User's Guide。如果打开了一个project，则可以显示完整信息如下。

