

第五章 MDK-ARM软件的使用

主讲人：漆强

电子科技大学

ytqiqiang@163.com

本章内容



MDK-ARM软件的使用流程



MDK-ARM软件的实用功能



RTE环境与STM32CubeMX的联合使用

教学目标

- 熟练掌握MDK-ARM软件的使用流程
- 熟练掌握MDK-ARM软件的调试方法
- 了解MDK-ARM软件的RTE环境



电子科技大学
University of Electronic Science and Technology of China

5.1 MDK-ARM软件的使用流程

基本特点



1

支持源代码的编辑、编译、程序的下载和调试等多种功能

2

编译器和器件支持包分离，可根据需求安装，减小软件大小

3

提供多种中间件，如RTOS和GUI等，提高开发效率

4

强大的仿真功能，可以仿真微控制器片内的所有资源

使用步骤



步骤1

新建工程



步骤2

工程设置



步骤3


程序编译和下载



步骤4

程序调试

1 新建工程



Project Settings	
Project Name	
Demo	工程名称
Project Location	
C:\Users\qiqiang\Desktop\CubeMX	工程路径

在CubeMX软件的工程设置中，已经设置了工程名称和工程路径。软件将自动生成基于MDK集成开发环境的工程文件，因此用户不需要单独新建工程。

MDK工程文件存放在用户设置的工程路径，以上一章介绍的Demo工程为例，其文件路径为：

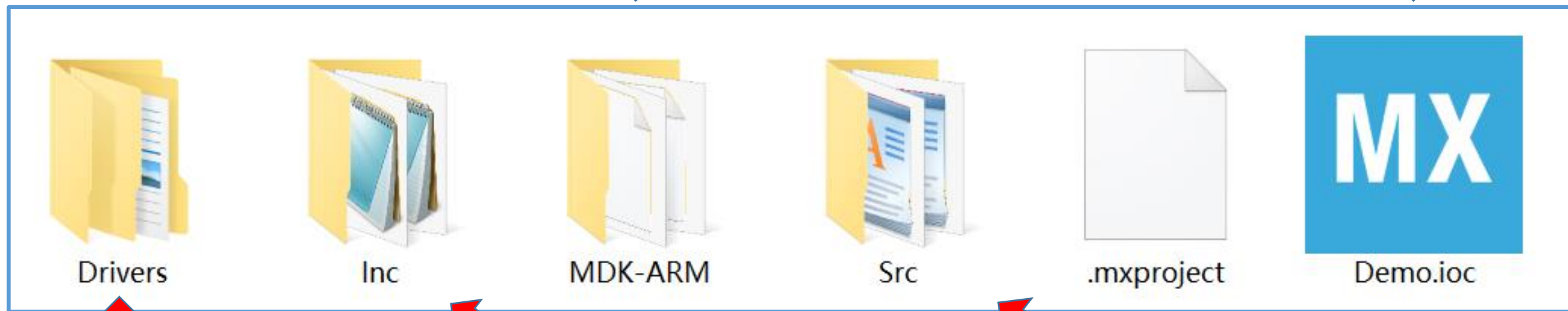
C:\Users\qiqiang\Desktop\CubeMX\Demo

MDK-ARM

工程文件内容

MDK工程文件及
`startup_stm32f411xe.s`

CubeMX
工程文件



HAL库驱动文件：
HAL_Driver文件夹存放
HAL库接口函数的源代码
CMSIS文件夹存放与内核
相关的文件

用户头文件：
`main.h`
`stm32f4xx_hal_conf.h`
`stm32f4xx_it.h`

用户源文件：
`main.c`
`stm32f4xx_hal_msp.c`
`stm32f4xx_it.c`
`system_stm32f4xx.c`

启动MDK工程



Drivers



Inc



MDK-ARM



DebugConfig



Demo



RTE



Demo\bin\qiang



Demo\avoptx



Demo.uvprojx



startup_stm32f411xe.lst



startup_stm32f411xe.s

MDK工程文件

C:\Users\qiqiang\Desktop\CubeMX\Demo\MDK-ARM\Demo.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

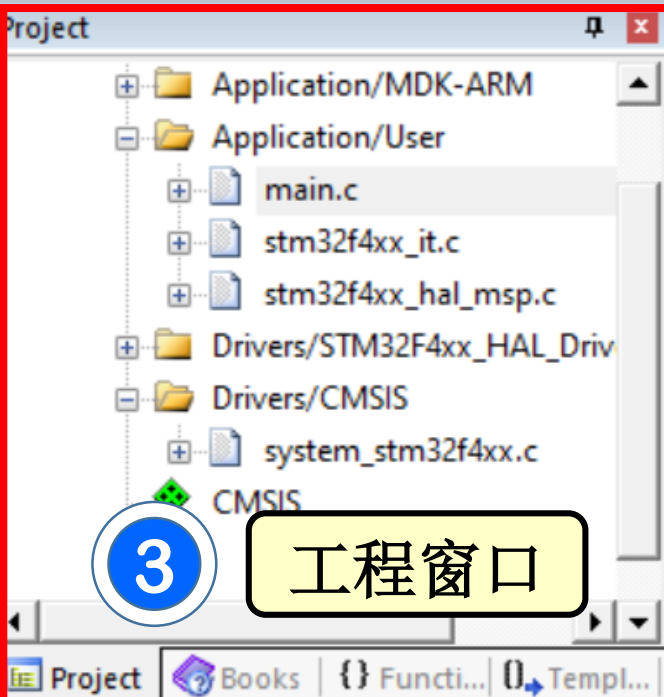


1

菜单栏

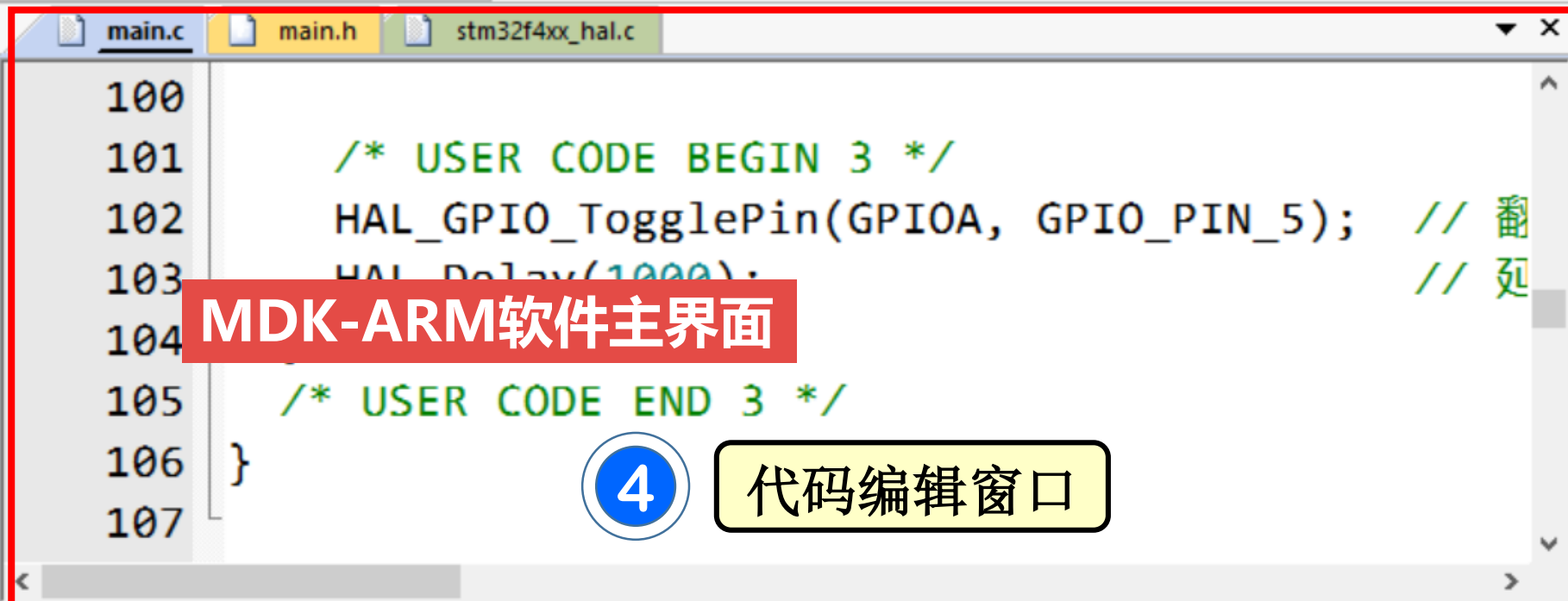
2

工具栏



3

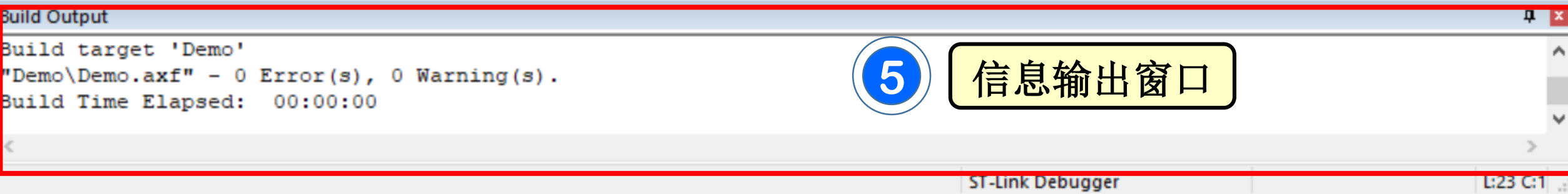
工程窗口



4

代码编辑窗口

MDK-ARM软件主界面



5

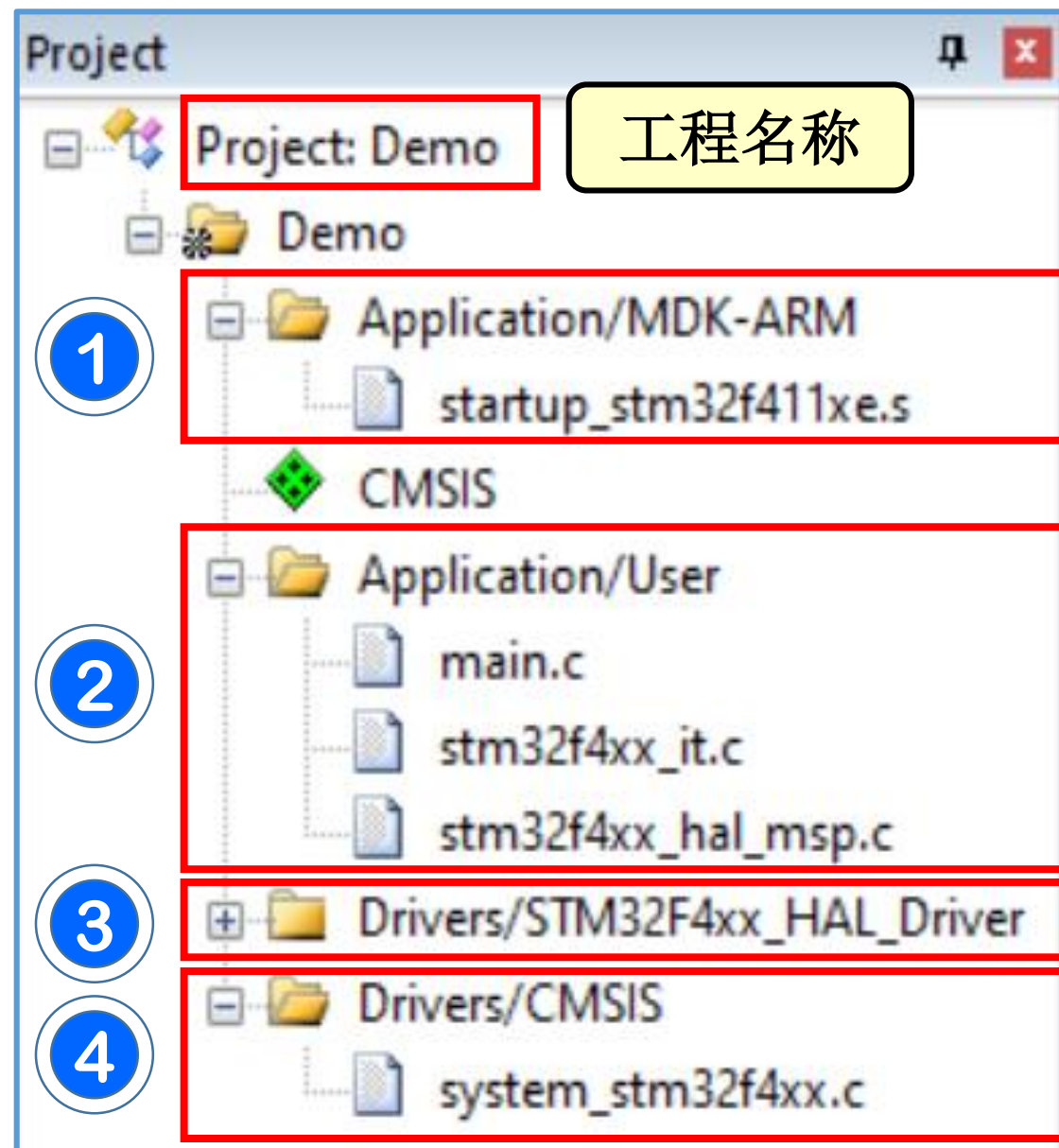
信息输出窗口

ST-Link Debugger

L:23 C:1

工程框架

- ① MDK-ARM组：启动代码文件
- ② User组：用户编程文件
- ③ HAL_Driver组：HAL库驱动文件
- ④ CMSIS组：系统初始化文件

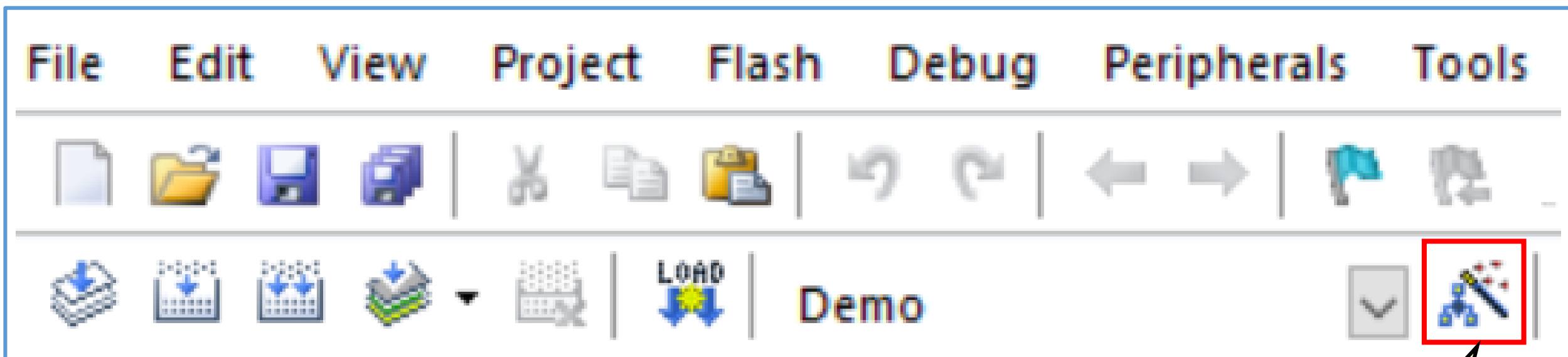




电子科技大学
University of Electronic Science and Technology of China

2 工程设置

工程设置

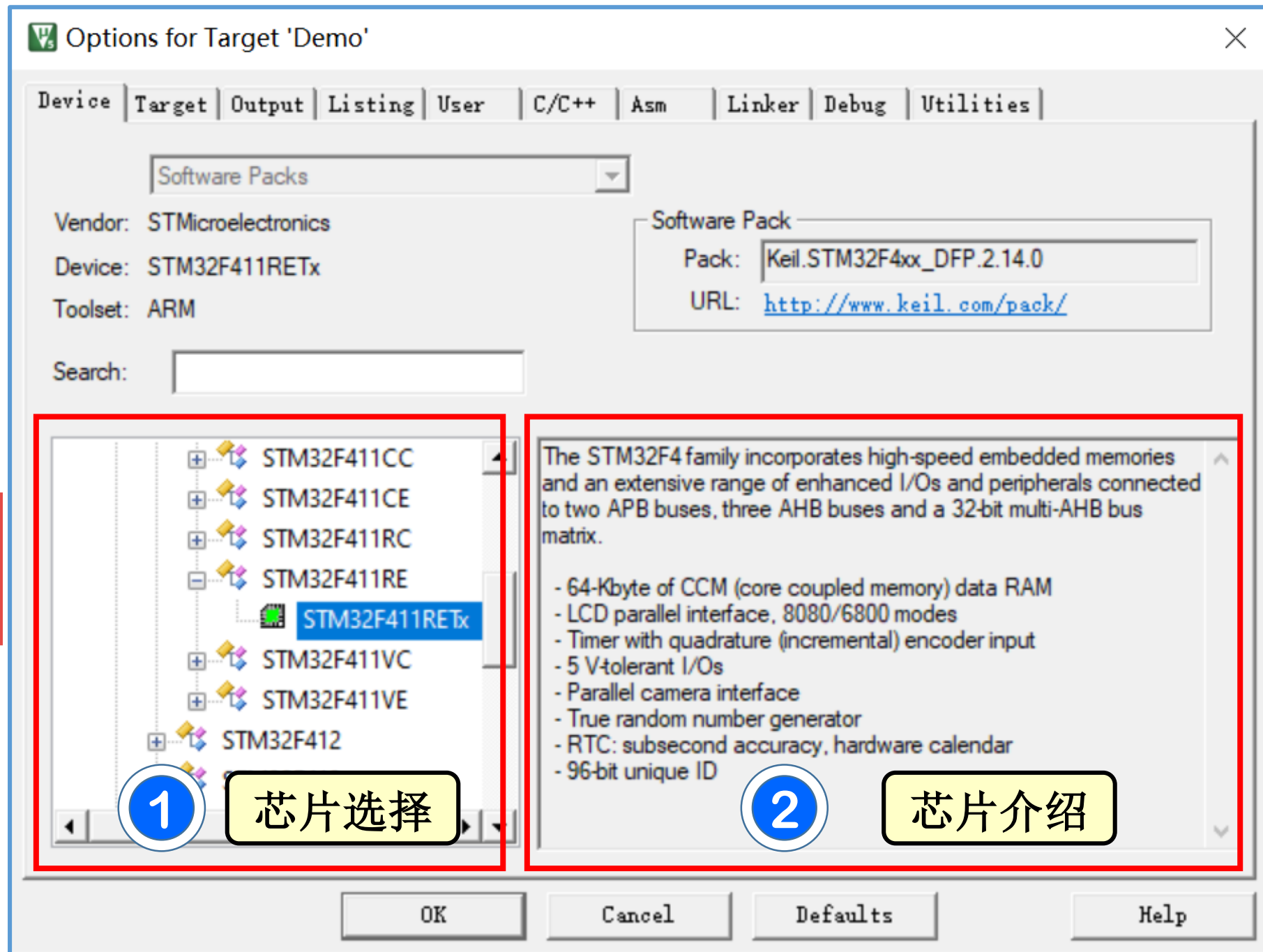


工程设置

Device标签页

主要功能

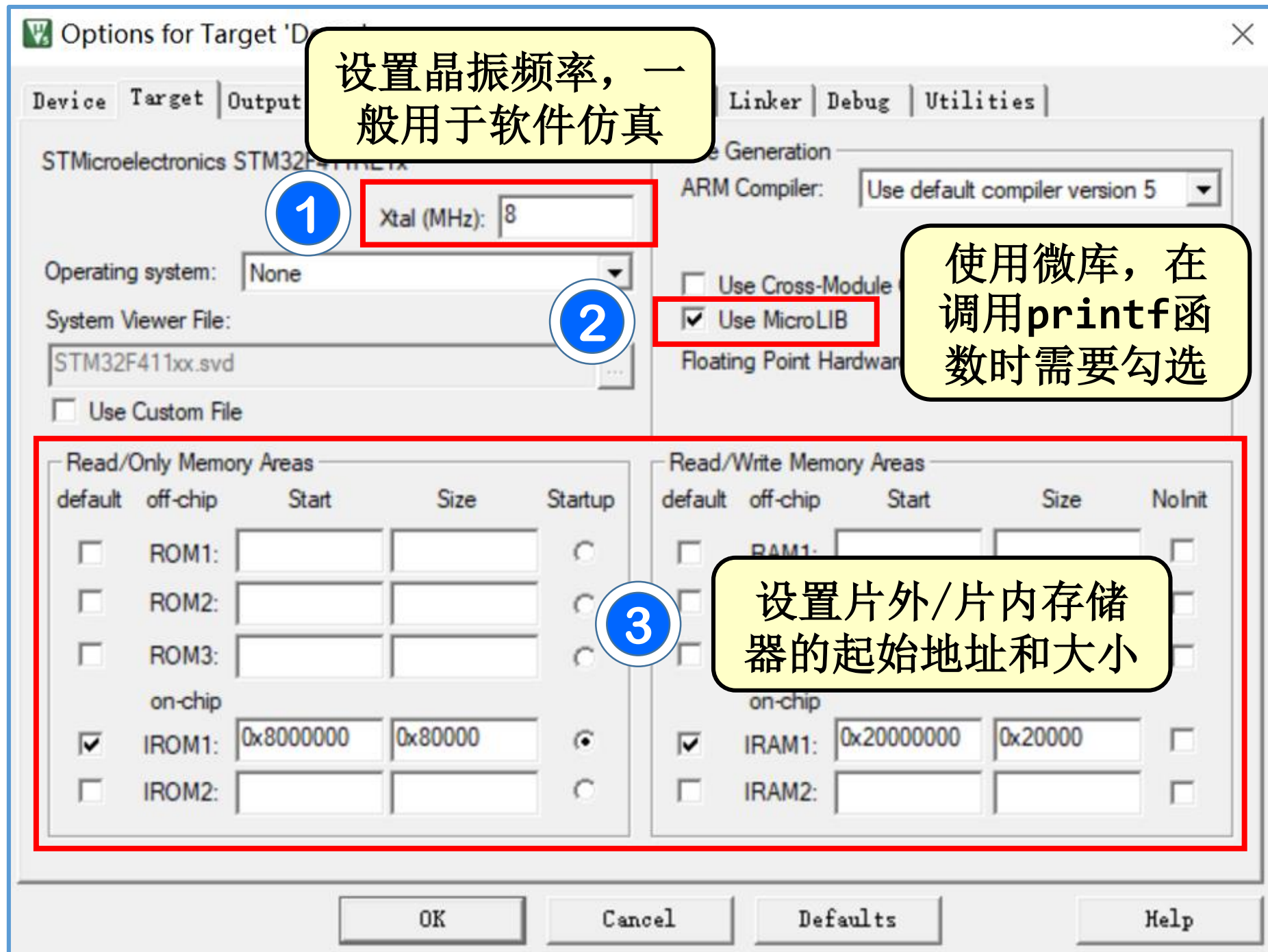
芯片型号的选择



Target标签页

主要功能

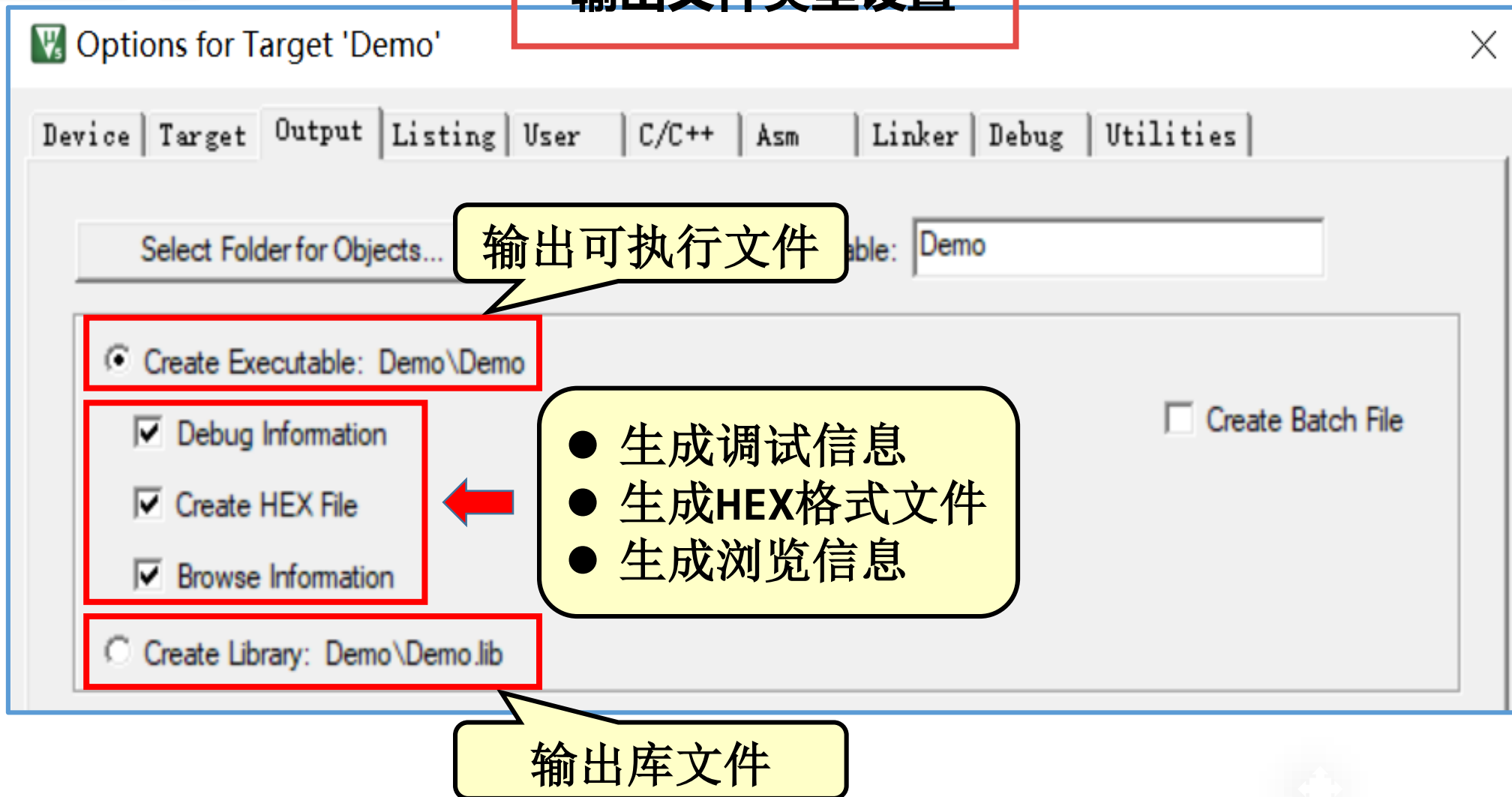
硬件设置



主要功能

Output标签页

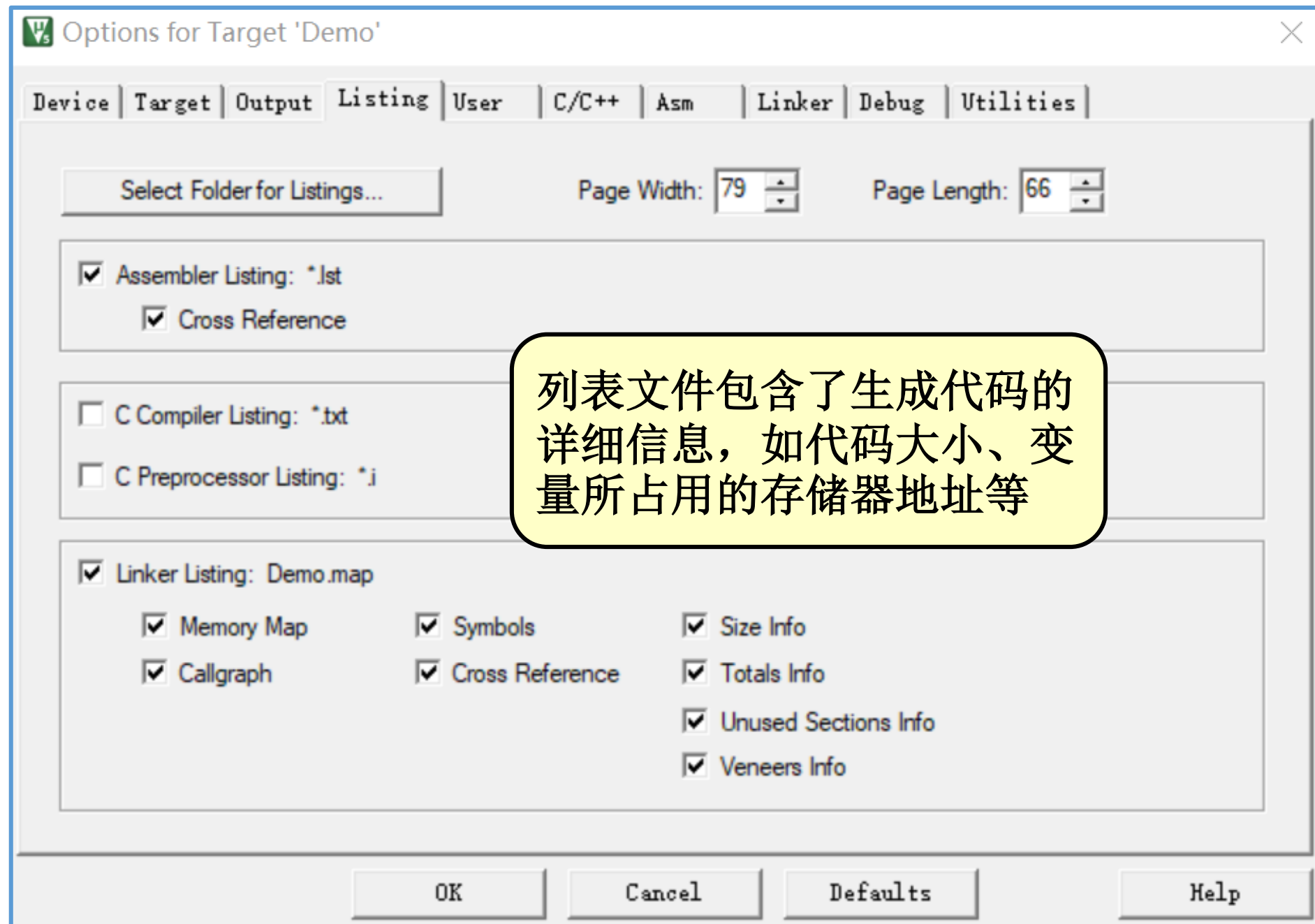
输出文件类型设置



Listing标签页

主要功能

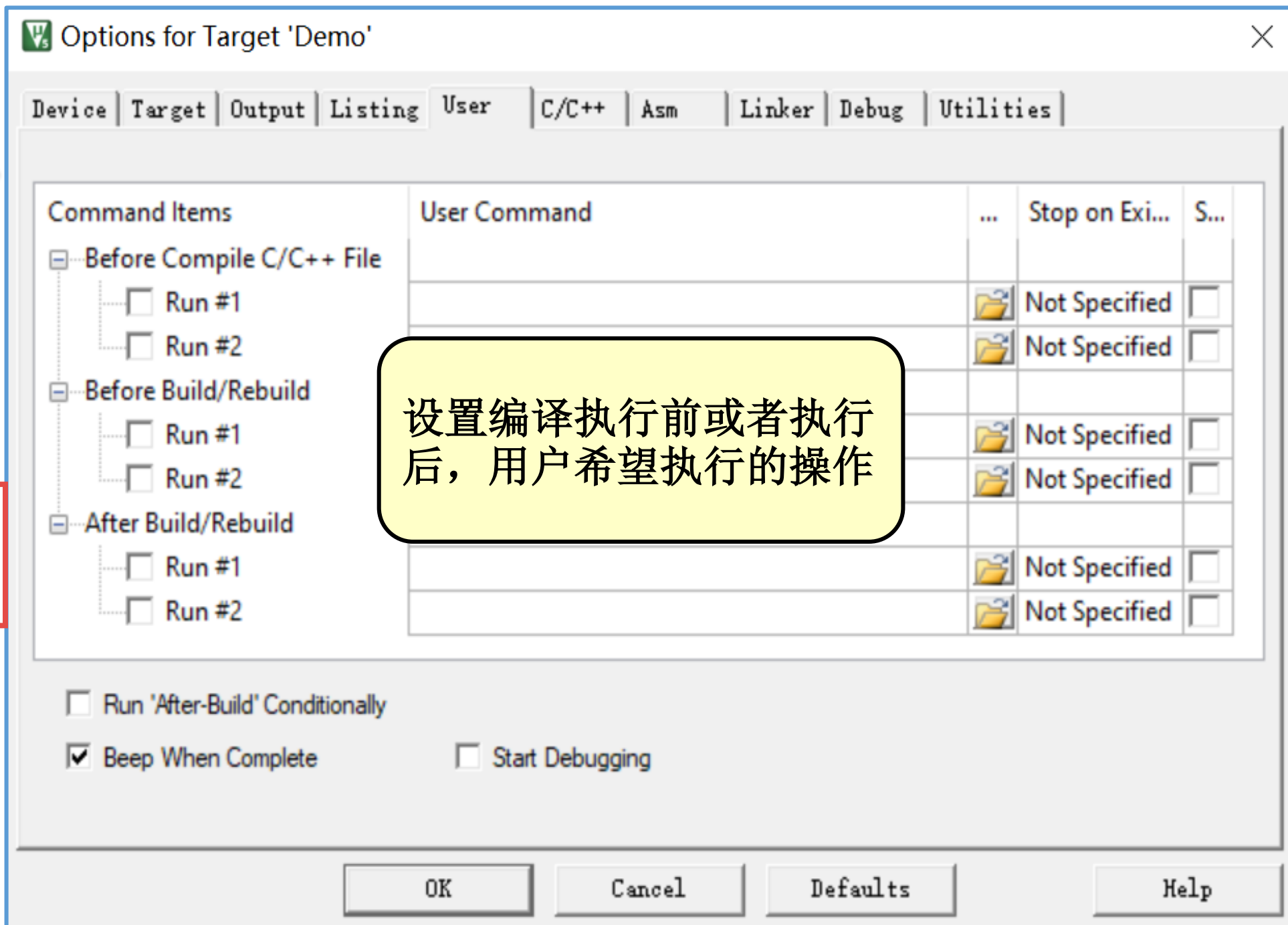
列表文件设置



User标签页

主要功能

用户功能设置



C/C++ 标签页

主要功能 与C语言编译相关设置

- USE_HAL_DRIVER表示使用HAL库
- STM32F411xE表示使用的芯片型号

1

预处理，相当于程序中的define定义

2

代码优化等级，建议初学者选择“Level 0”，便于观察程序中定义的全部变量以及设置断点

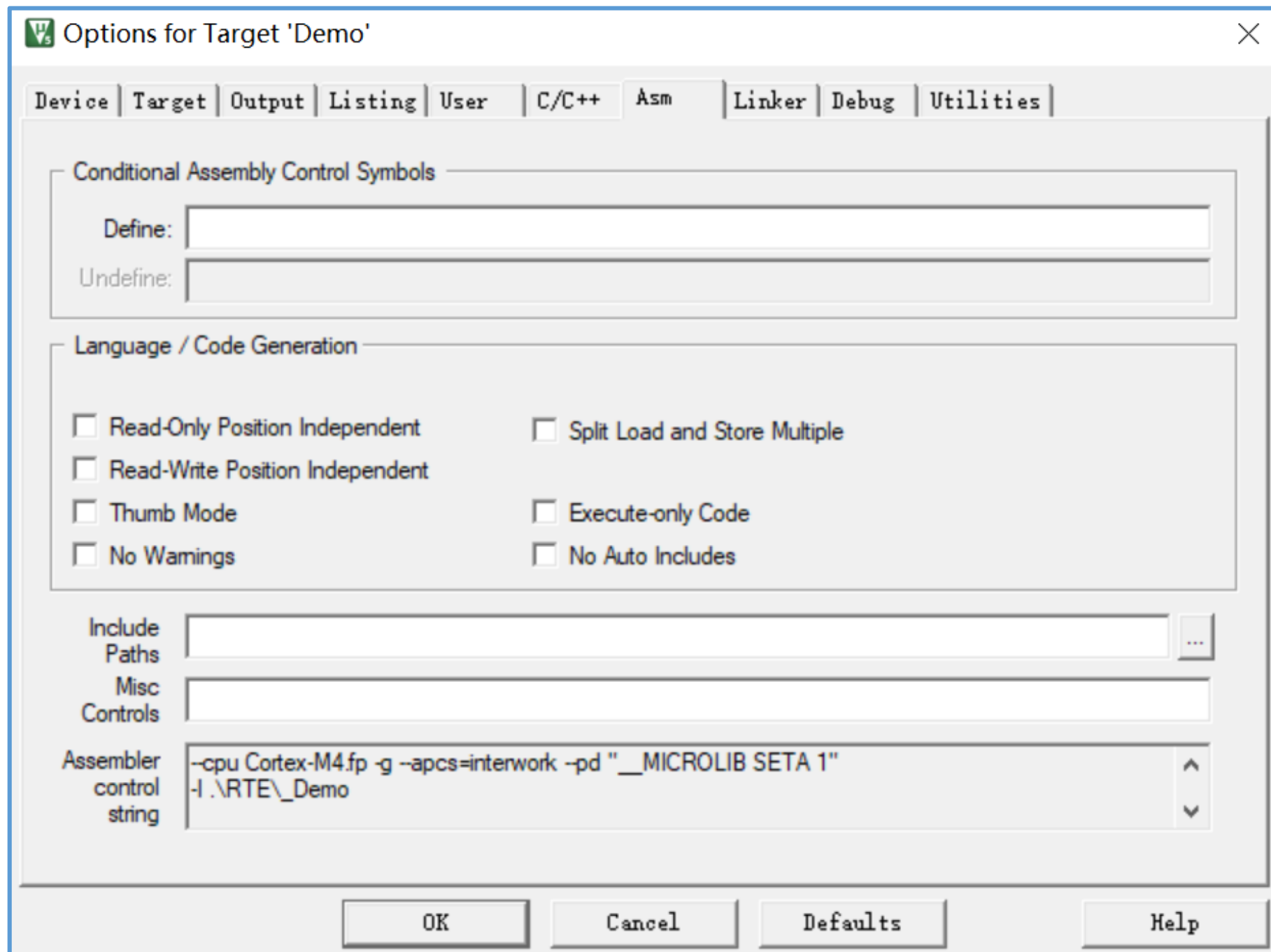
3

设置头文件的路径

ASM标签页

主要功能

与汇编语言编译 相关设置



Linker标签页

主要功能

链接器功能设置

Options for Target 'Demo'

DeviceTargetOutputListingUserC/C++AsmLinkerDebugUtilities

☒ Use Memory Layout from Target Dialog

☐ Make RW Sections Position Independent

☐ Make RO Sections Position Independent

☐ Don't Search Standard Libraries

☒ Report 'might fail' Conditions as Errors

X/O Base:

R/O Base:0x08000000

R/W Base:0x20000000

disable Warnings:

Scatter File

...

Edit...

Misc controls

Linker control string

--cpu Cortex-M4.fp *.o
--library_type=microlib --scatter "Demo\Demo.sct"

OK

Cancel

Defaults

Help

Debug标签页

主要功能
仿真功能设置

Option

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

软件仿真

硬件仿真

仿真器设置

仿真器选择

1

2

Use Simulator [with restrictions](#) Settings

Limit Speed to Real-Time

Load Application at Startup Run to m

Initialization File:

Restore Debug Session Settings

Breakpoints Toolbox

Watch Windows & Performance Analyzer

Memory Display System Viewer

CPU DLL: Parameter:

SARMCM3.DLL -REMAP -MPU

Dialog DLL: Parameter:

DCM.DLL -pCM4

Warn if outdated Executable is loaded

Use: ST-Link Debugger Settings

ULINKplus Debugger

CMSIS-DAP Debugger

J-LINK / J-TRACE Cortex

Models Cortex M Debugger

ST-Link Debugger

NU-Link Debugger

Pemicro Debugger

Stellaris ICDI

SiLabs UDA Debugger

Altera Blaster Cortex Debugger

TI XDS Debugger

Watch windows

Memory Display System Viewer

Driver DLL: Parameter:

SARMCM3.DLL -MPU

Dialog DLL: Parameter:

TCM.DLL -pCM4

Warn if outdated Executable is loaded

Manage Component Viewer Description Files ...

OK Cancel Defaults Help

仿真器信息

仿真器设置窗口

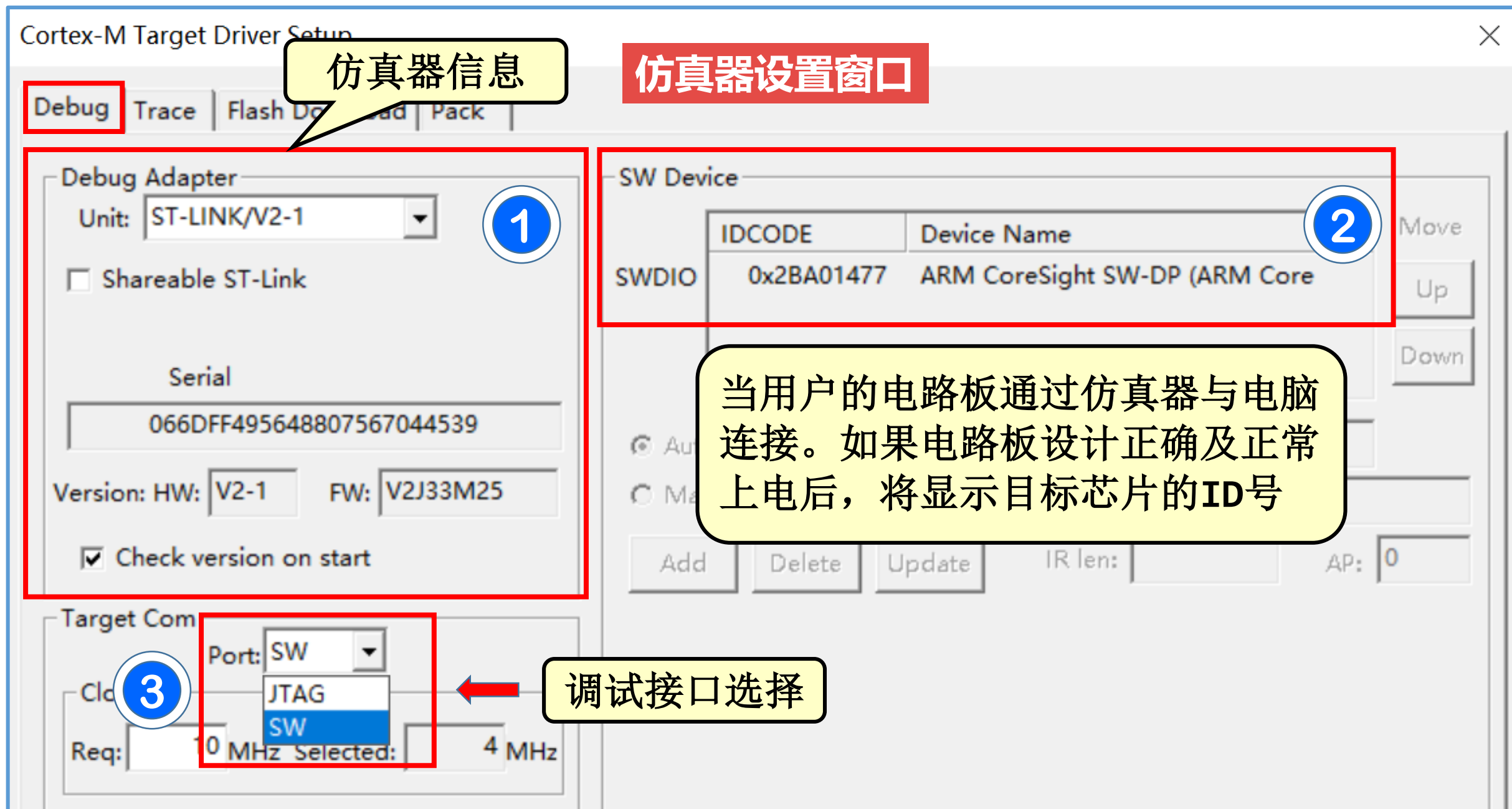
1

2

当用户的电路板通过仿真器与电脑连接。如果电路板设计正确及正常上电后，将显示目标芯片的ID号

3

调试接口选择



Cortex-M Target Driver Setup

仿真器设置窗口

Debug | Trace | **Flash Download** | Pack

Download Function



- ☐ Erase Full Chip
- ☒ Erase Sectors
- ☐ Do not Erase
- ☒ Program
- ☒ Verify
- ☒ Reset and Run

确保程序下载后立即执行

1

Start: 0x20000000

Size: 0x0800

MCU片内Flash的编程算法

Programming Algorithm

Description	Device Size	Device Type	Address Range
STM32F4xx 512kB Flash	512k	On-chip Flash	08000000H - 0807FFFFH

2

Start:

Size:

Add

Remove

Utility标签页

主要功能

下载功能设置

Options for Target 'Demo'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

Configure Flash Menu Command

☒ Use Target Driver for Flash Programming

Use Debug Driver

Settings

☒ Use Debug Driver

☒ Update Target before Debugging

Init File: ... Edit...

☐ Use External Tool for Flash Programming

Command: ...

Arguments:

☐ Run Independent

Configure Image File Processing (FCARM):

Output File:

Add Output File to Group: Application/MDK-ARM

Image Files Root Folder:

☐ Generate Listing

OK

Cancel

Defaults

Help

特别提示**利用CubeMX软件生成MDK工程只需要修改两处地方****1**

在**Debug**标签页中的选择所使用的仿真器，并在仿真器设置窗口中勾选 “**Reset and Run**”。

2

对于初学者，建议在**C/C++**标签页中的设置程序优化等级为 “**Level 0**”，以便于程序的仿真调试。

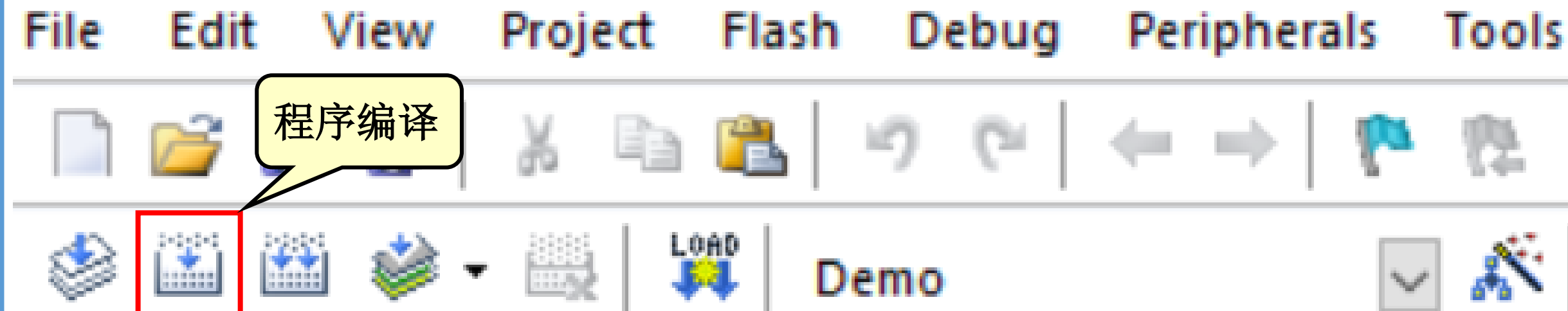


电子科技大学
University of Electronic Science and Technology of China

3 程序编译和下载

程序编译

程序编译



Build Output

```
linking...
Program Size: Code=5672 RO-data=472 RW-data=1184
FromELF: creating hex file...
"Demo\Demo.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:32
```

没有错误和警告

程序下载

如果程序下载后没有执行，可以按下复位按键，确保程序执行

程序下载

LOAD

Demo

Build Output

Erase Done.

Programming Done.

Verify OK.

Application running ...

Flash Load finished at 22:05:05

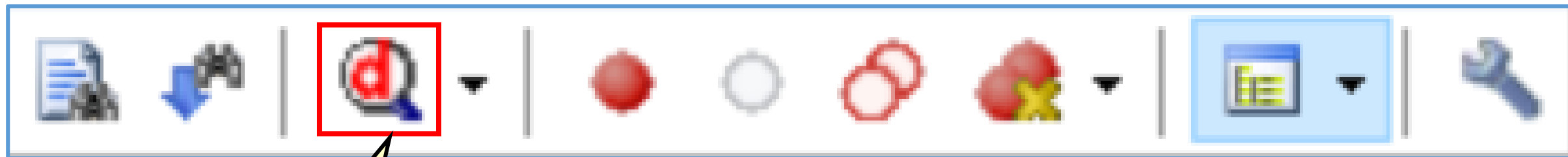


电子科技大学
University of Electronic Science and Technology of China

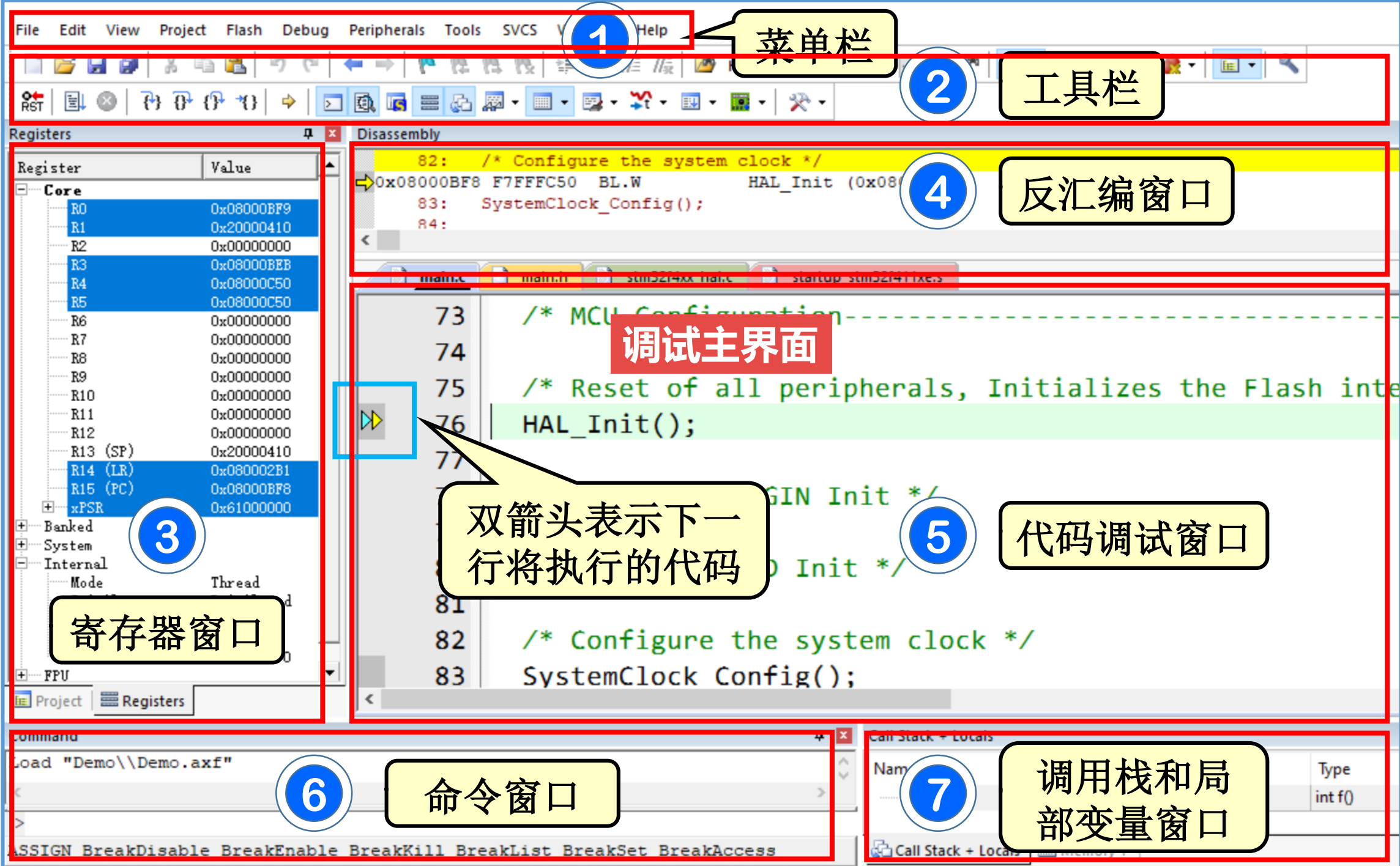
4 程序调试

程序调试

利用工具栏的Debug按钮进入程序调试界面

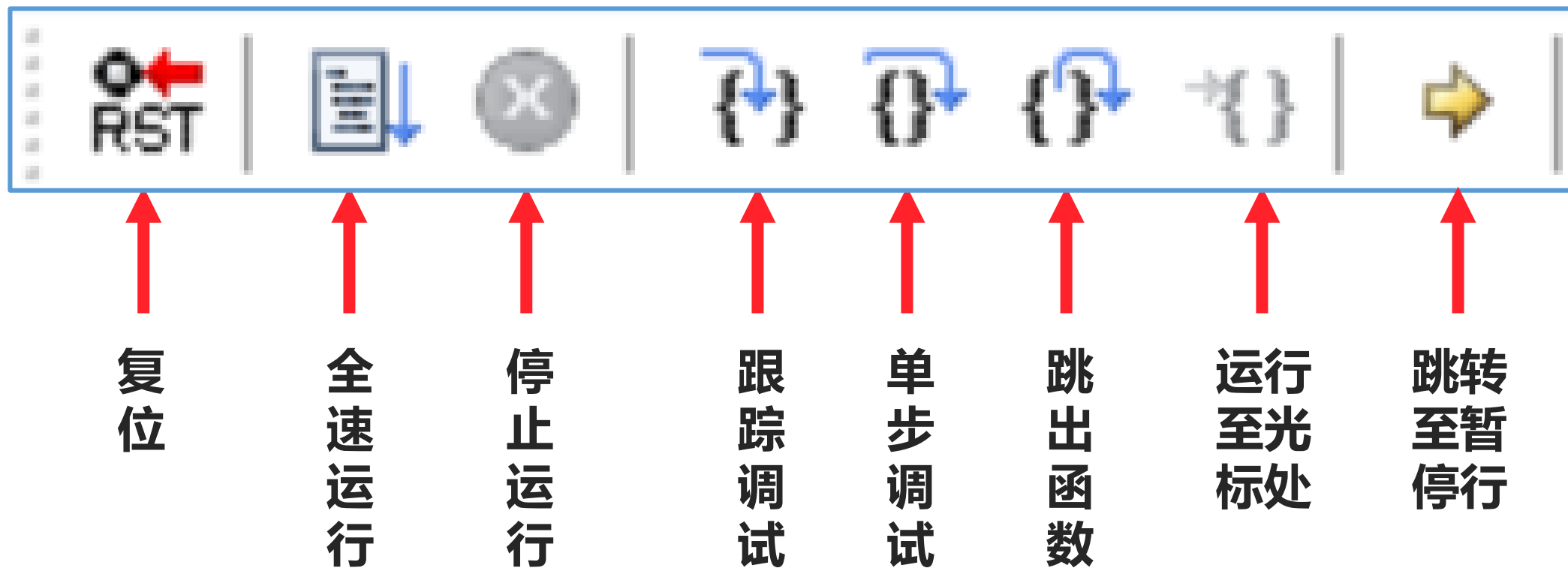


进入/退出
程序调试



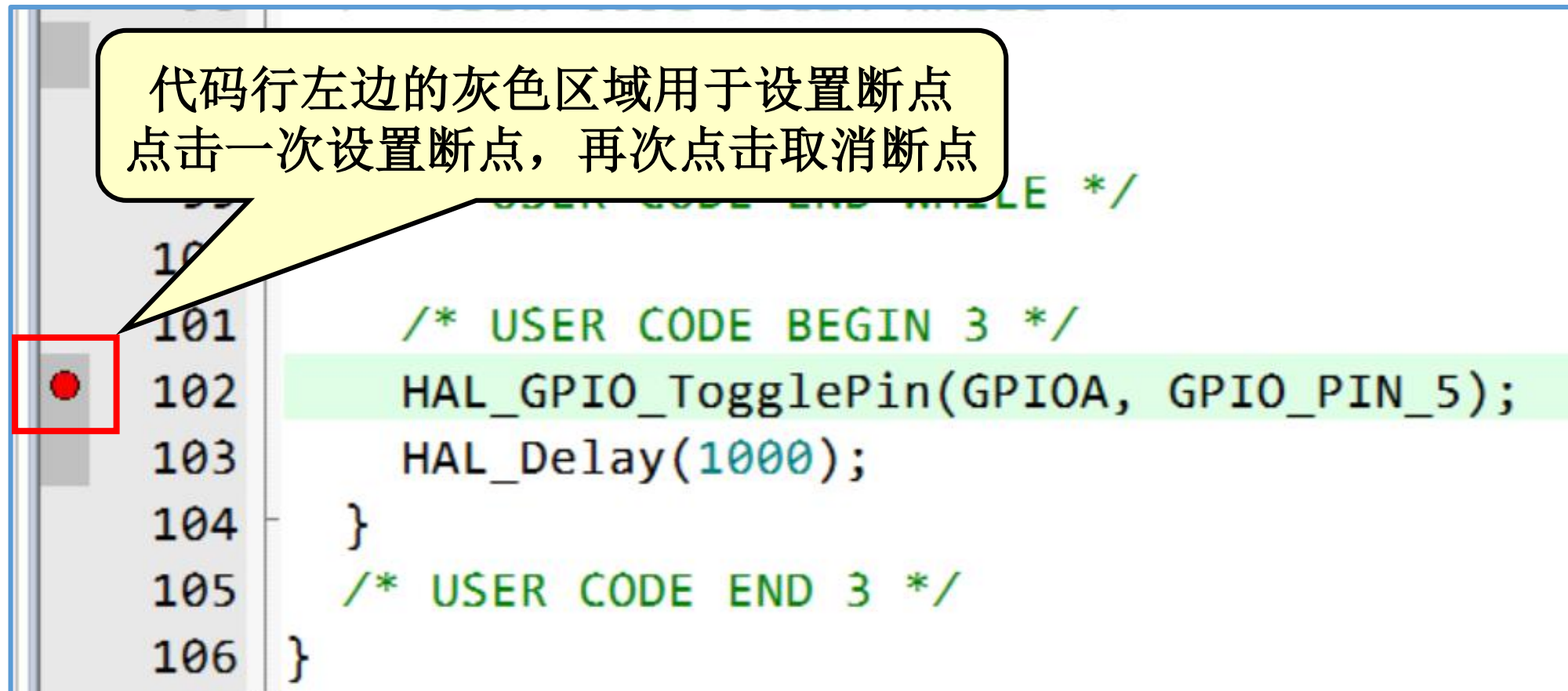
常规调试方法

- 跟踪调试：遇到函数调用将进入函数内部执行
- 单步调试：遇到函数调用，将函数当做一条语句执行，不进入函数内部



断点调试

代码行左边的灰色区域用于设置断点
点击一次设置断点，再次点击取消断点

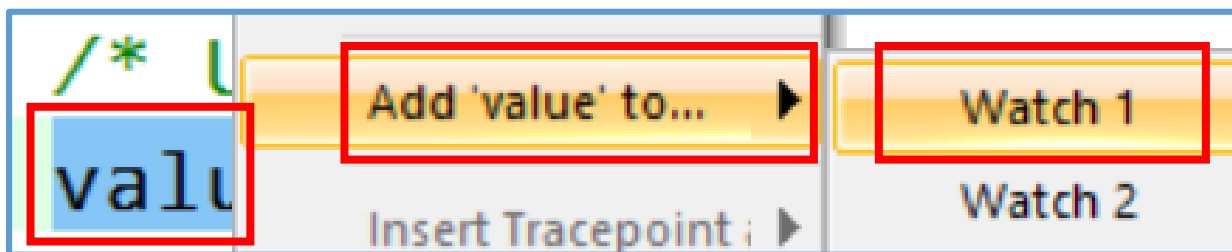


观察窗口

用于查看变量的值

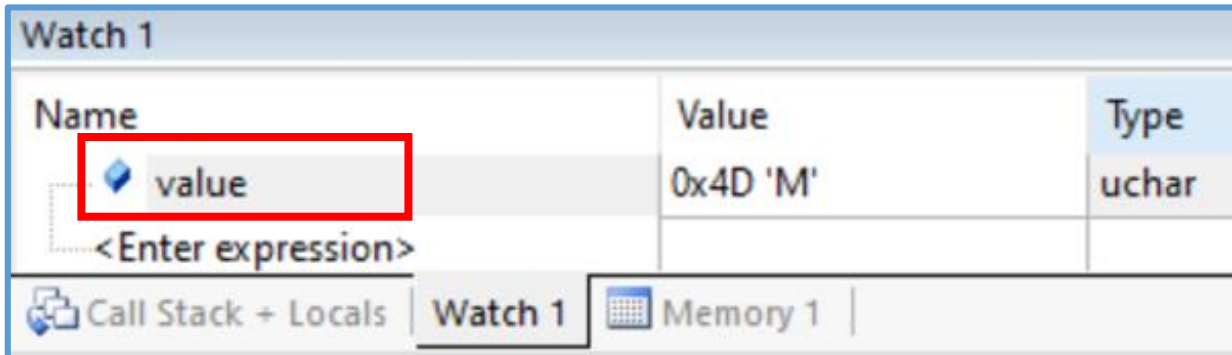
1

选中变量，鼠标右键



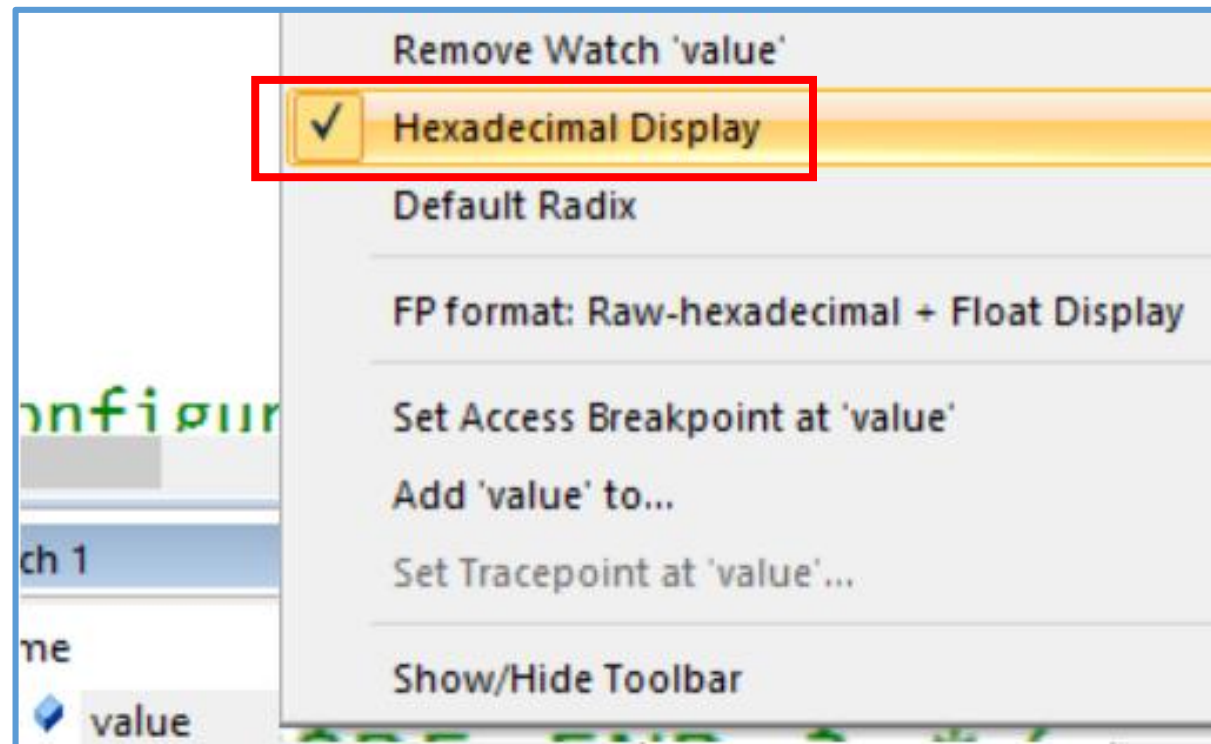
2

选中变量，鼠标右键

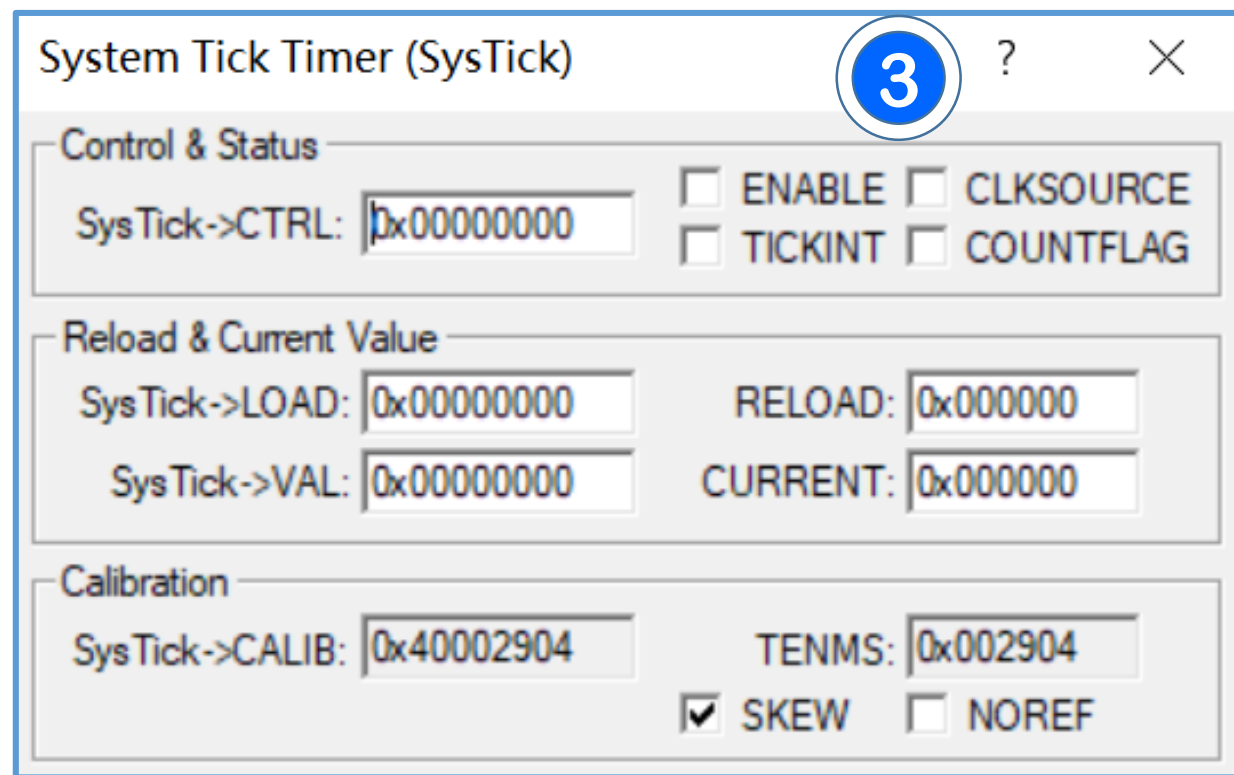
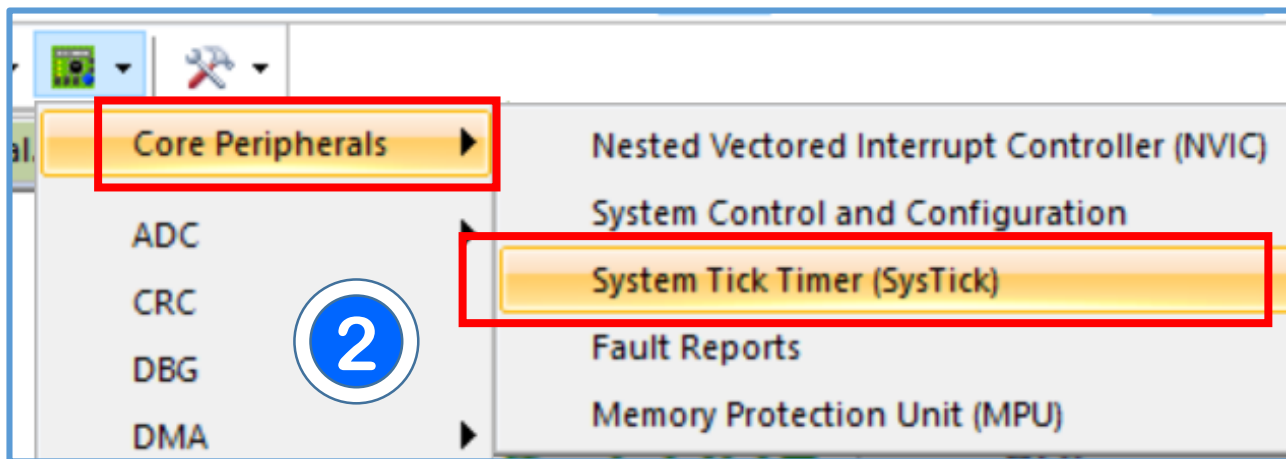
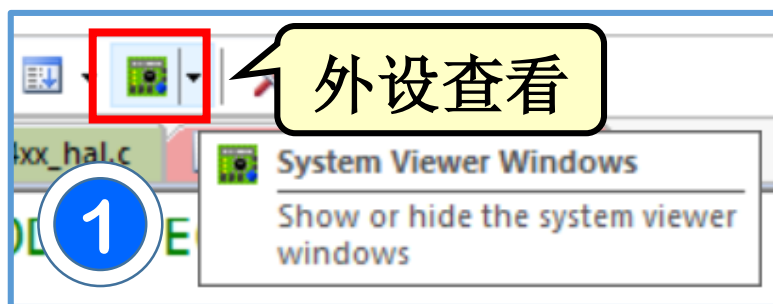


3

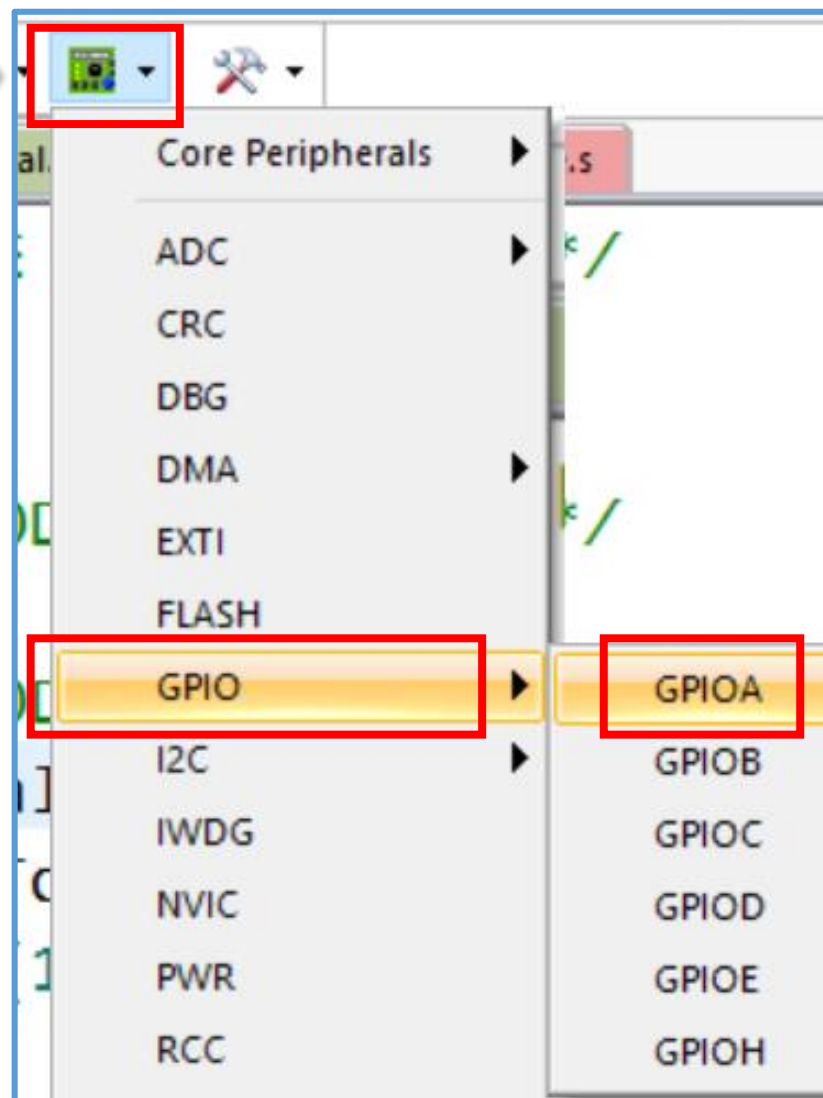
选择变量显示格式



外设查看功能可以查看与内核外设和片内外设相关的硬件寄存器的当前值



片内外设



The screenshot shows the 'GPIOA' register configuration window. The window displays a table of properties and their values. A red arrow points from the 'GPIOA' item in the peripheral tree to this window.

Property	Value
<input type="checkbox"/> MODER	0xA8000000
<input type="checkbox"/> OTYPER	0xA8000000
<input type="checkbox"/> OSPEEDR	0xA8000000
<input type="checkbox"/> PUPDR	0xA8000000
<input type="checkbox"/> IDR	0xA8000000
<input type="checkbox"/> ODR	0xA8000000
<input type="checkbox"/> BSRR	0
<input type="checkbox"/> LCKR	0xA8000000
<input type="checkbox"/> AFR1	0xA8000000
<input type="checkbox"/> AFRH	0xA8000000



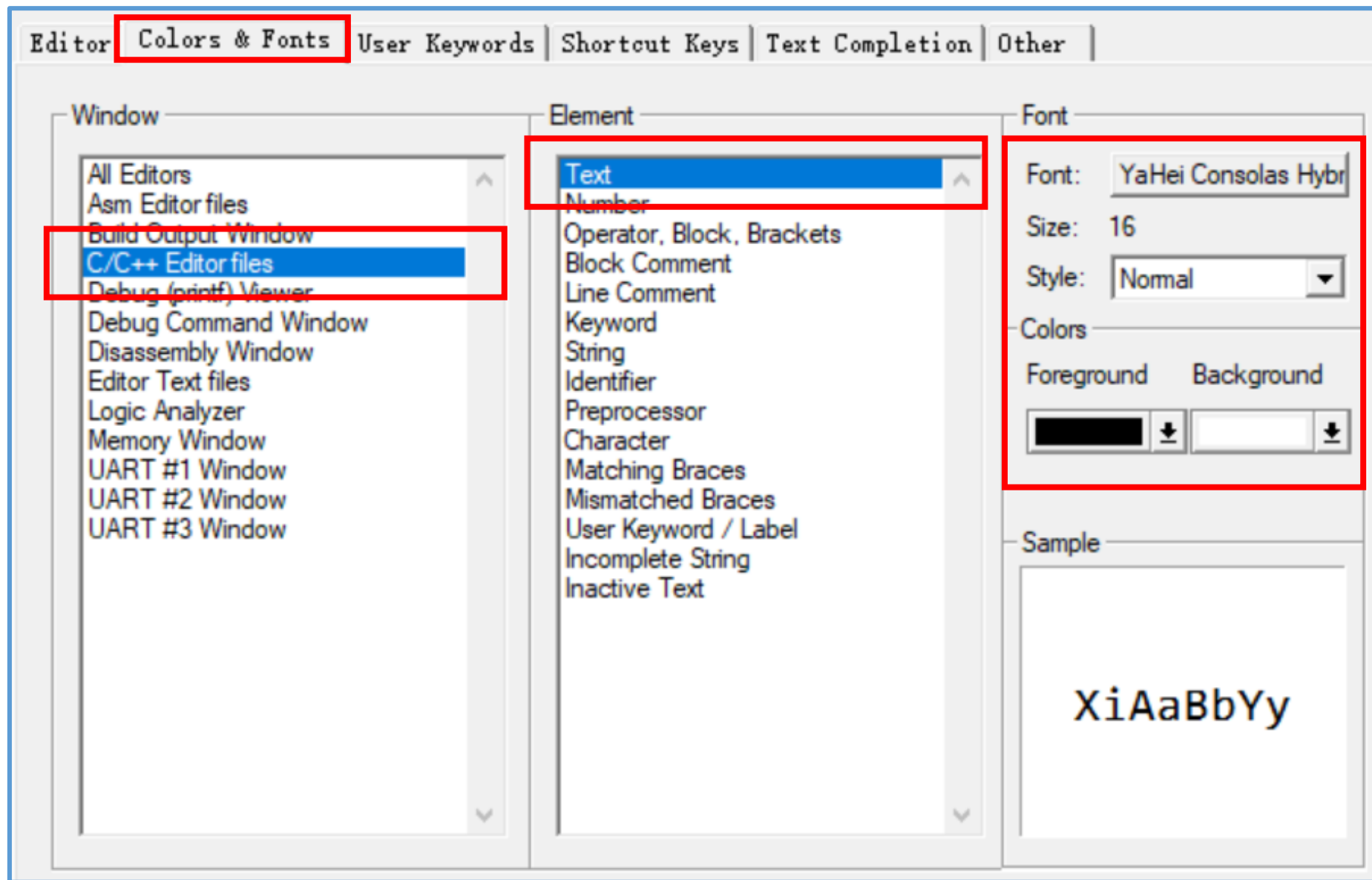
电子科技大学
University of Electronic Science and Technology of China

5.2 MDK-ARM软件的实用功能

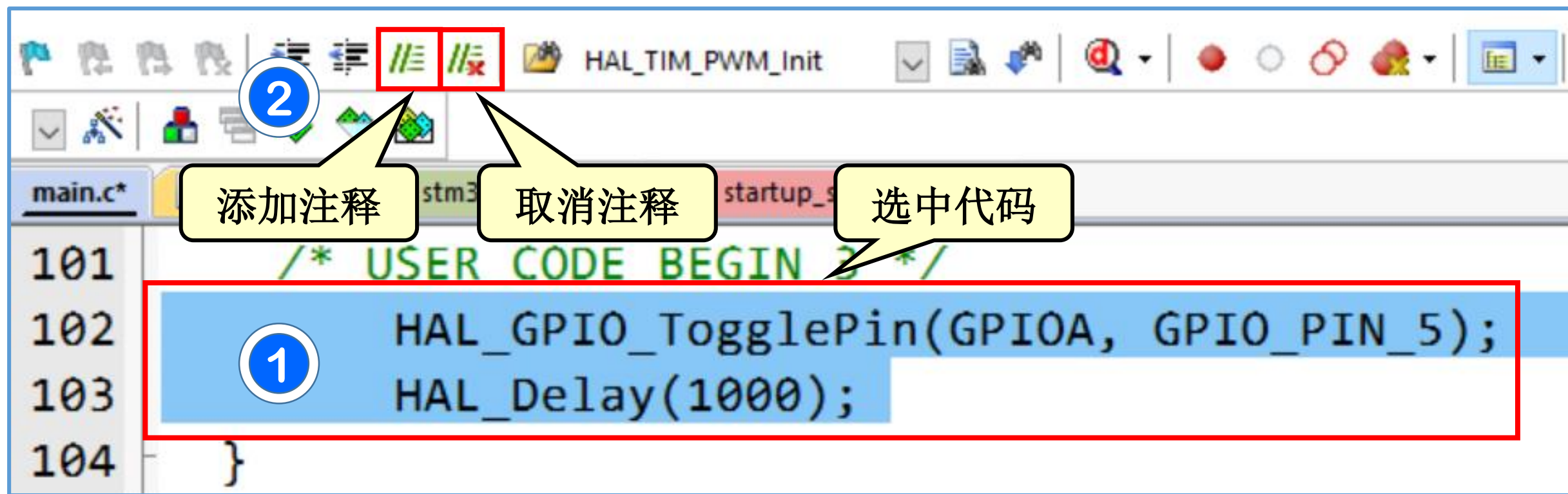
编辑器设置

软件配置

Configuration...
Configure µVision



代码注释及取消

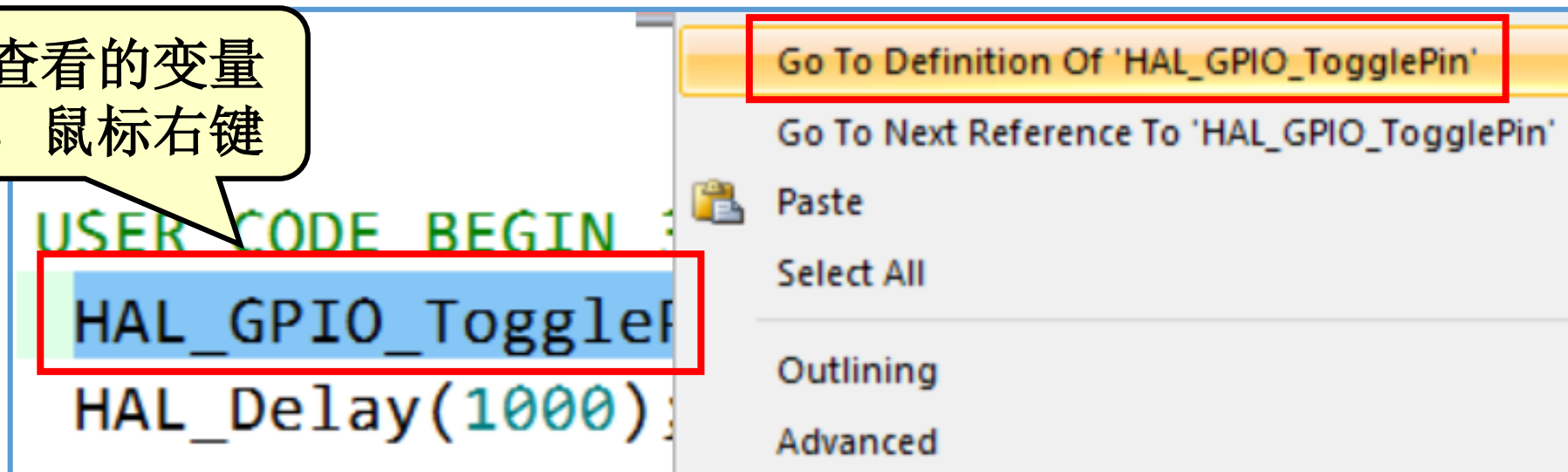


快速查看

1

查看变量和函数的原始定义

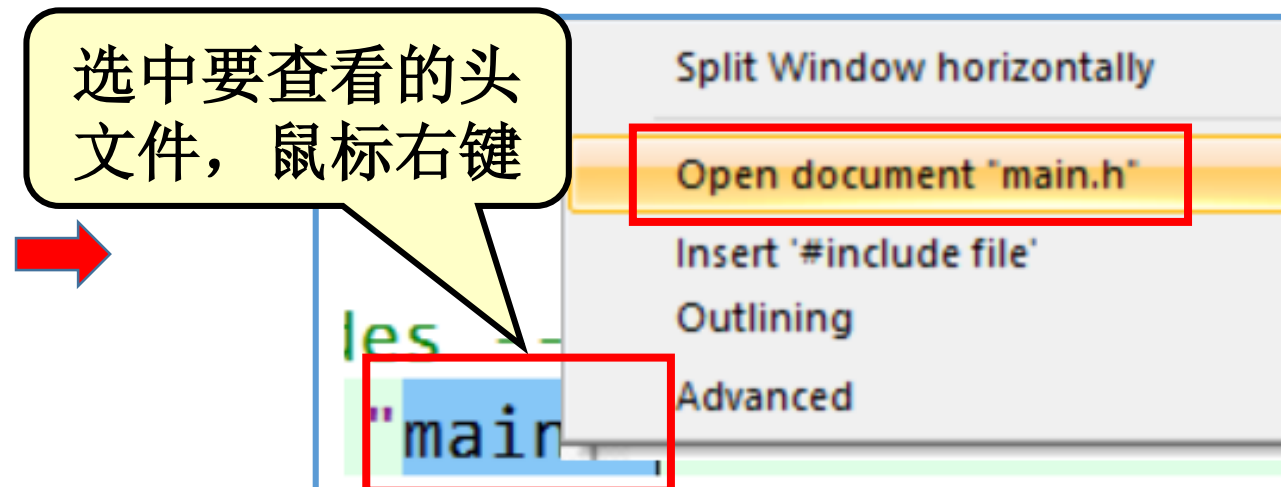
选中要查看的变量
或函数，鼠标右键



2

查看头文件

选中要查看的头
文件，鼠标右键





电子科技大学
University of Electronic Science and Technology of China

5.3 RTE环境和CubeMX联合使用

基本特点**RTE环境提供嵌入式系统设计的统一接口**

- 1 提供CMSIS软件接口和芯片启动文件
- 2 提供常用外设的驱动（GPIO、UART、I2C、SPI和USB等）
- 3 提供RTOS、GUI、文件系统和网络协议栈等中间件
- 4 可以借助CubeMX软件完成外设初始化设置

RTE使用步骤

在桌面上新建一个名为“RTE”的文件夹



Keil
uVision5



Project

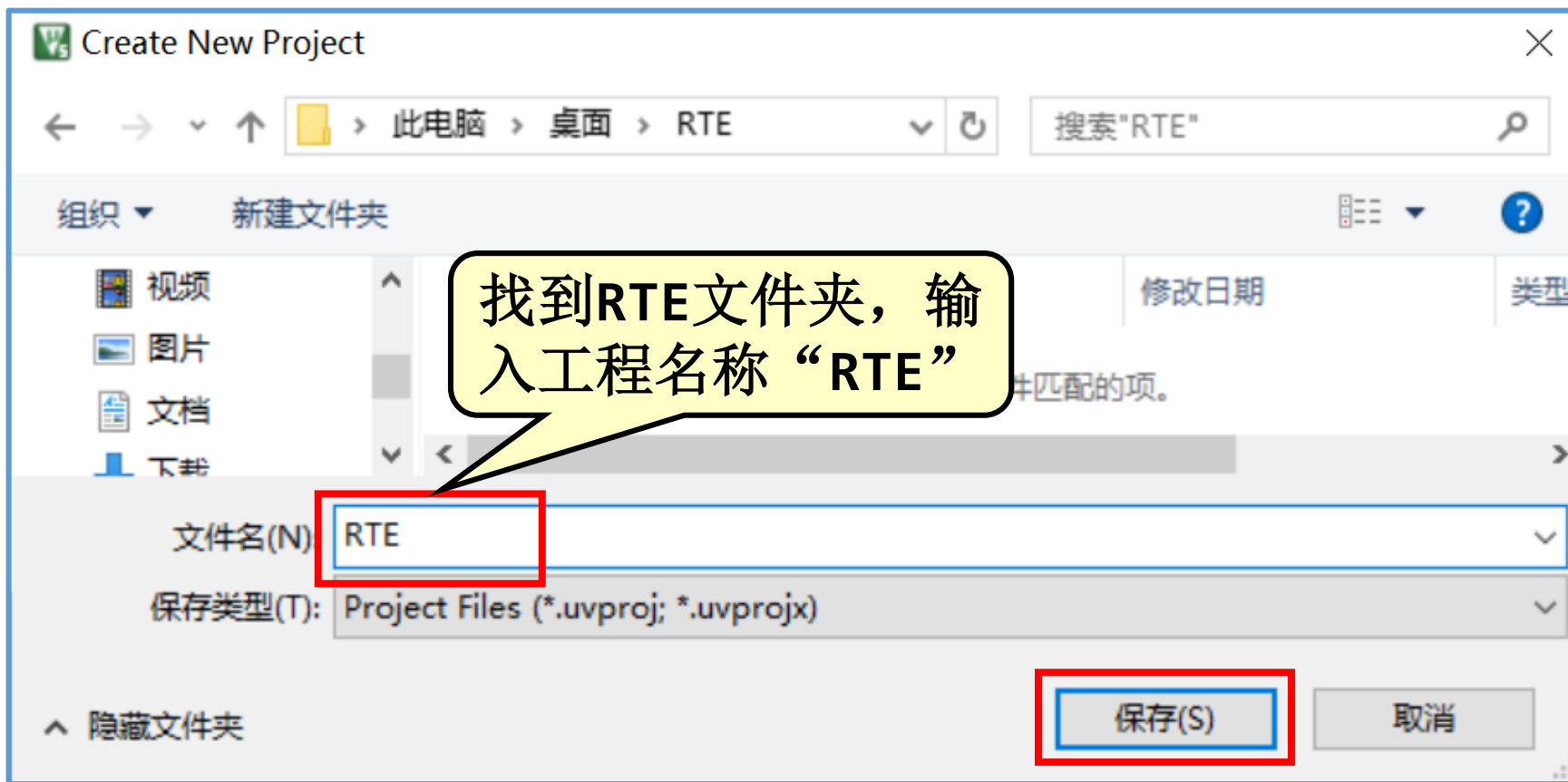
Flash

Debug

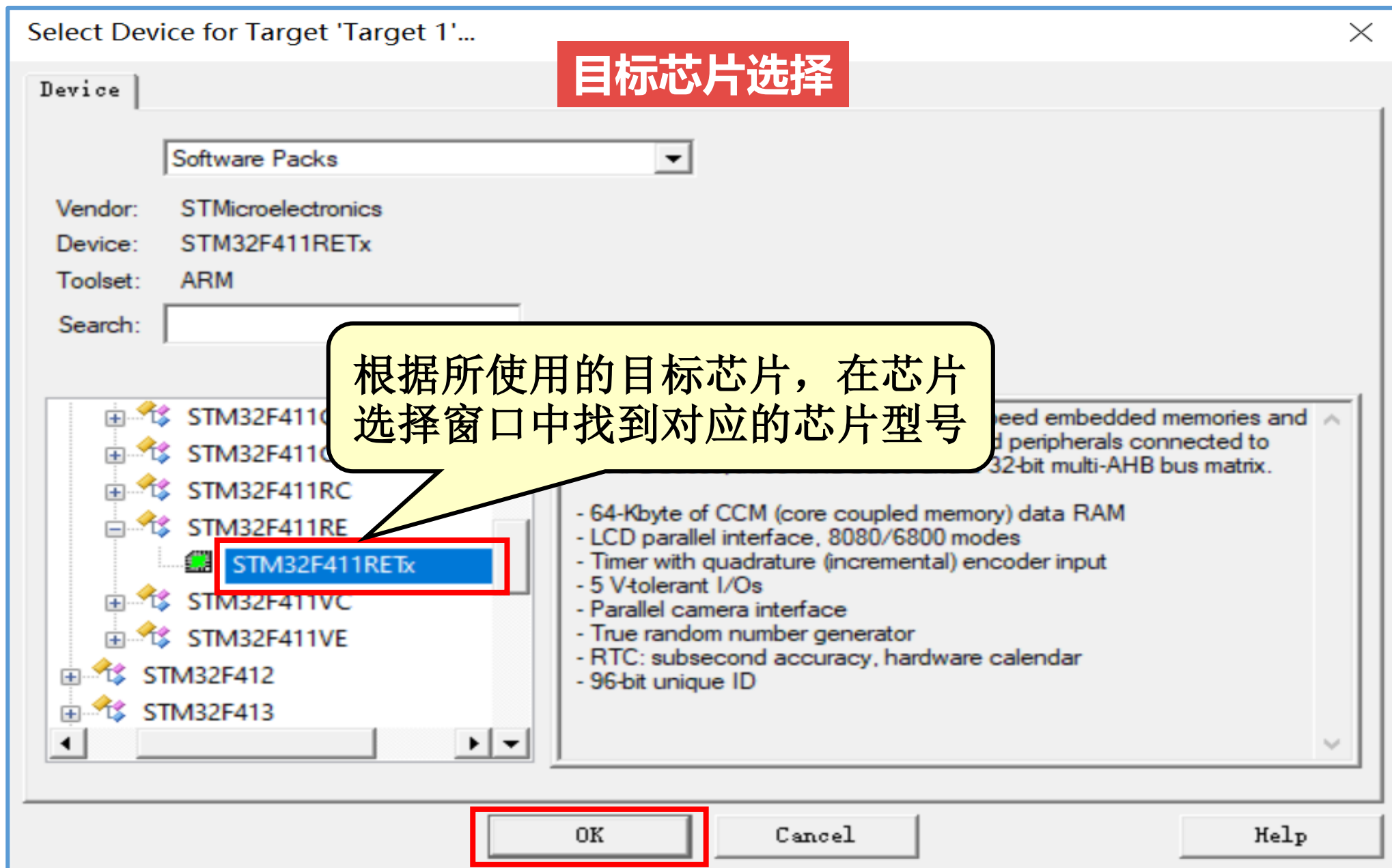
Peripherals

New μ Vision Project...

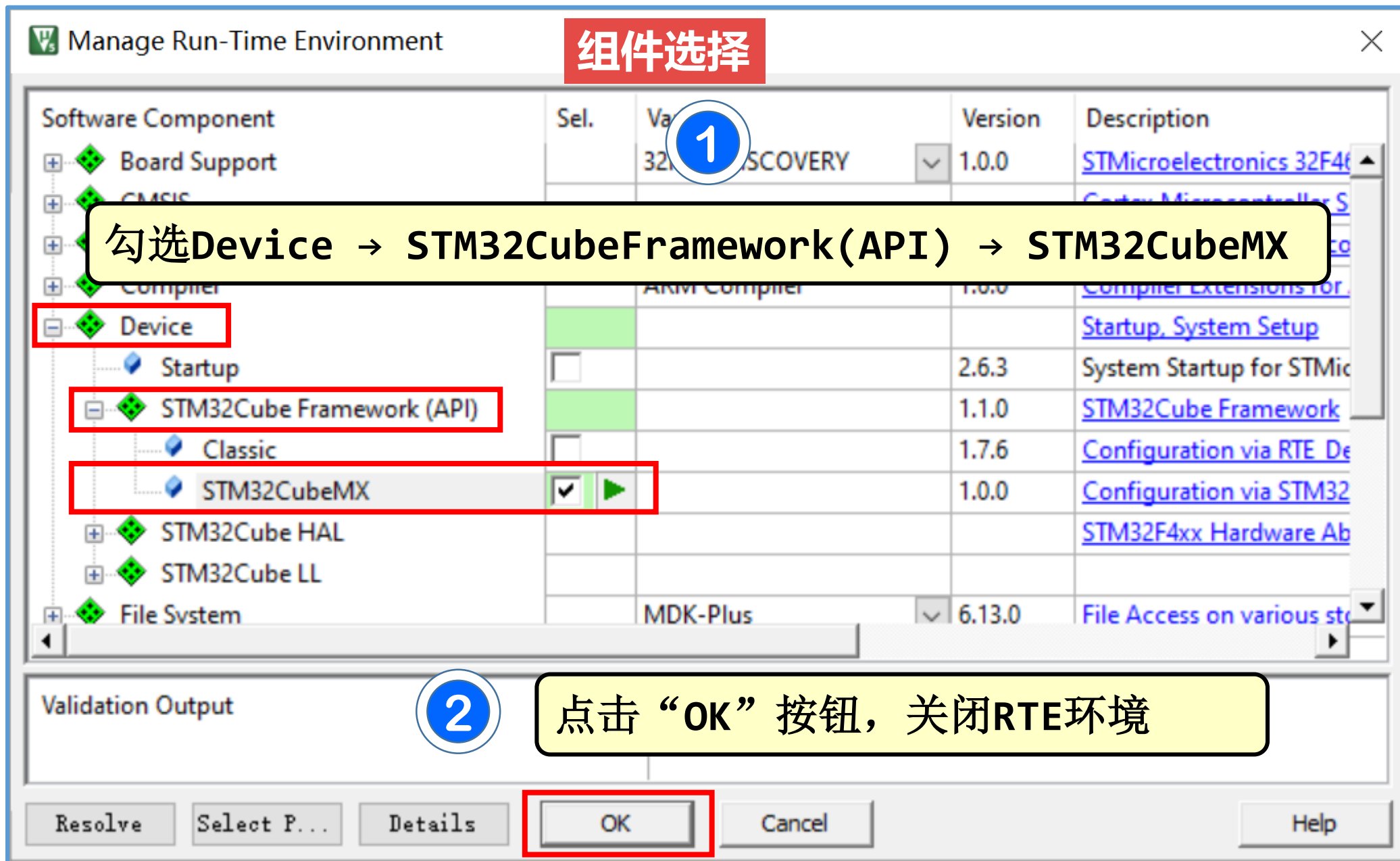
New Multi-Project Workspace...



RTE使用步骤



RTE使用步骤



RTE使用步骤

启动CubeMX

MDK: Selected Software Component Requires Code Generation by 'STM32CubeMX' X



Component:

Keil::Device:STM32Cube Framework:STM32CubeMX

Generator Program:

STM32CubeMX

Generates:

.\RTE\Device\STM32F411RETx\FrameworkCubeMX.gpdsc

3

将自动弹出启动CubeMX窗口，
点击“Start STM32CubeMX”
按钮，启动STM32CubeMX软件

Start STM32CubeMX

Cancel

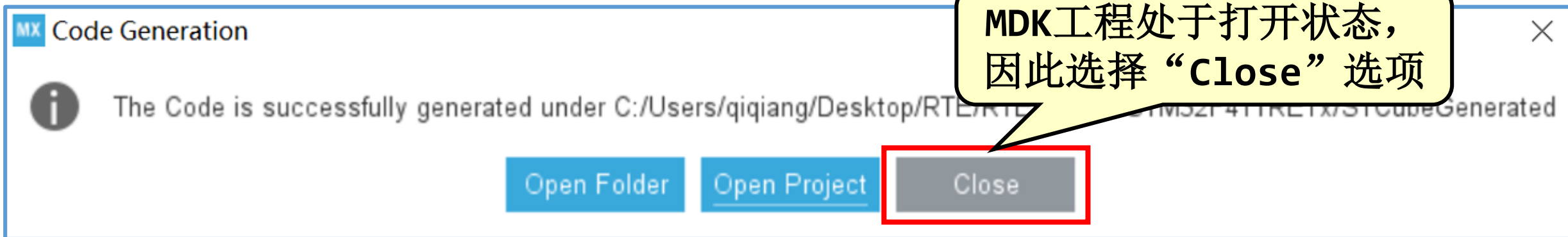
CubeMX软件使用步骤



只进行引脚分配、外设配置和时钟配置三个步骤，不需要进行目标选择和工程配置

联合使用

在CubeMX软件中完成相关设置，生成代码



在MDK软件中添加用户代码，完成工程设置，并进行程序的编译、下载和调试等后续步骤

本章结束

本章结束

