

# 如何使用 STM32的软件库 在IAR的EWARM下 进行应用开发

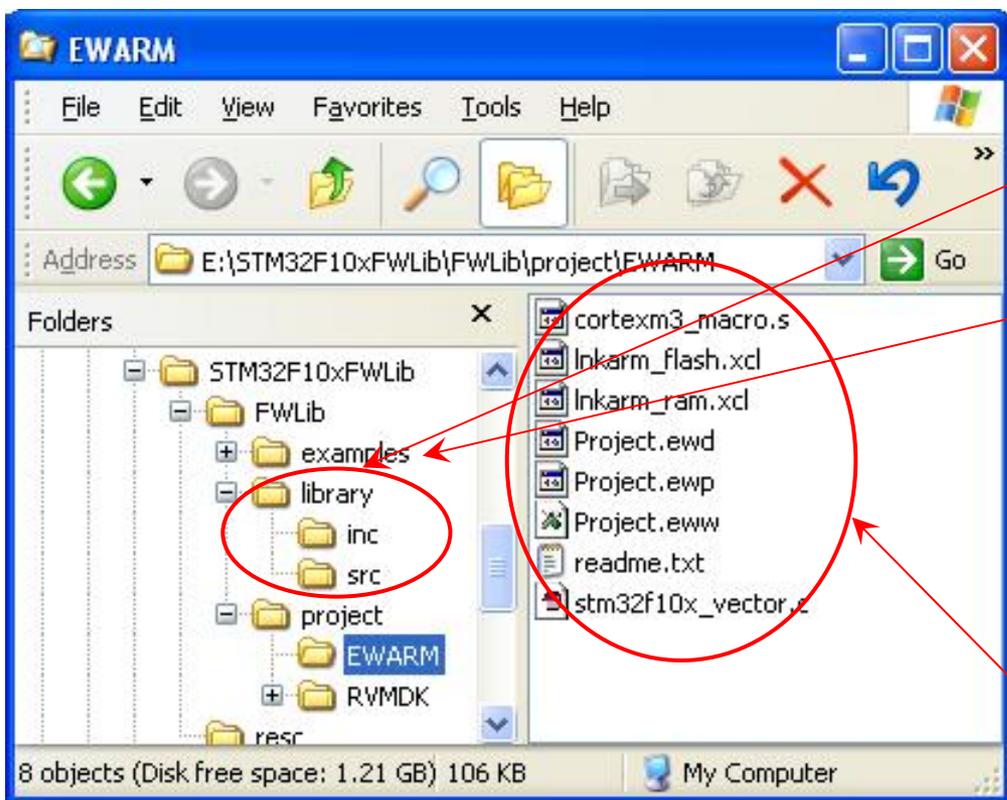


上海

2007年12月



# 步骤一：创建项目目录，拷贝公共文件

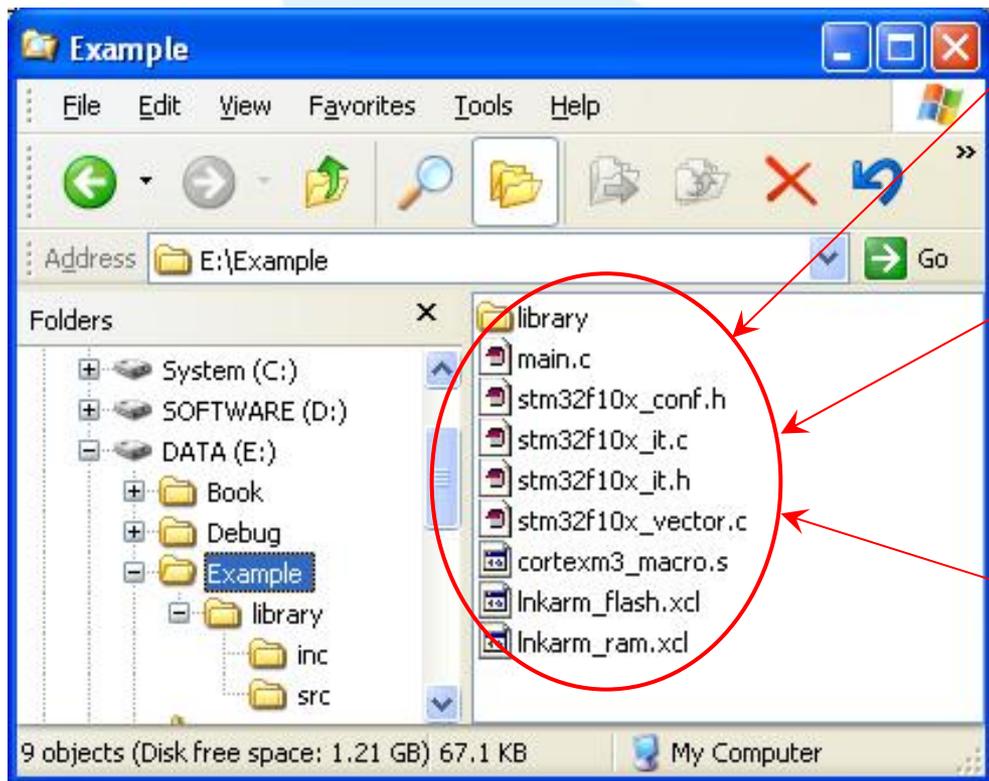


① 将STM32软件库中FWlib目录中的library目录拷贝到所建项目的目录中。

② 将软件库的Examples目录里的任一例程的stm32f10x\_conf.h、stm32f10x\_it.c、stm32f10x\_it.h和main.c拷贝到项目的目录中。

③ 拷贝软件库中\project\EWARM\目录中的stm32f10x\_vector.c、cortexm3\_macro.s、lnkarm\_ram.xcl和lnkarm\_flash.xcl到项目的目录中。

# 步骤一：创建项目目录，拷贝公共文件(续)



① 将STM32软件库中FWlib目录中的library目录拷贝到所建项目的目录中。

② 将软件库的Examples目录里的任一例程的stm32f10x\_conf.h、stm32f10x\_it.c、stm32f10x\_it.h和main.c拷贝到项目的目录中。

③ 拷贝软件库中\project\EWARM\目录中的stm32f10x\_vector.c、cortexm3\_macro.s、lnkarm\_ram.xcl和lnkarm\_flash.xcl到项目的目录中。

# 步骤二：修改应用文件(一)

## ◆ stm32f10x\_conf.h

```
#define _ADC
#define _ADC1
//#define _ADC2
...
#define _FLASH

#define _GPIO // include gpio.h file
//#define _GPIOA // not use GPIOA
//#define _GPIOB // not use GPIOB
#define _GPIOC // use GPIOC
//#define _GPIOD // not use GPIOD
//#define _GPIOE // not use GPIOE
#define _AFIO // use alternat function
...
#define _NVIC
#define _RCC
...
// Value of the External oscillator in Hz
#define HSE_Value ((u32)8000000)
```

修改设置文件  
stm32f10x\_conf.h

根据自己使用需求，打开相应外设的define：

◆ #define \_PPP (必须的)

◆ #define \_PPPx (可选的，根据外设决定)

根据自己电路板上的外部晶振频率来修改 HSE\_Value。

注：本文中使用STM32软件库中的例子。

## 步骤二：修改应用文件(二)

### ◆ main.c

```
#include "STM32F10x_lib.h"
int main() // main program(添加自己的代码)
{
    RCC_Configuration(); // Set System clock
    NVIC_Configuration(); // NVIC configuration
    GPIO_Configuration(); // GPIO configuration
    ADC12_Configuration(); // ADC configuration
    // Enable ADC1 reset calibration register
    ADC_ResetCalibration(ADC1);
    // Check ADC1 calibration
    while(ADC_GetResetCalibrationStatus(ADC1));
    ADC_StartCalibration(ADC1); // calibrate ADC1
    // Check the end of ADC1 calibration
    while(ADC_GetCalibrationStatus(ADC1));
    ADC_SoftwareStartConvCmd(ADC1, ENABLE);
    while(1); // Infinite loop
}
```

必须添加此行！

- 在main.c中编写自己的主程序
- 添加自己编写的代码。

## 步骤二：修改应用文件(三)

### ◆ **stm32f10x\_it.c**

```
#include "stm32f10x_it.h"

void ADC_IRQHandler(void)
{
    // Toggle PC.06 pin
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_SET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);

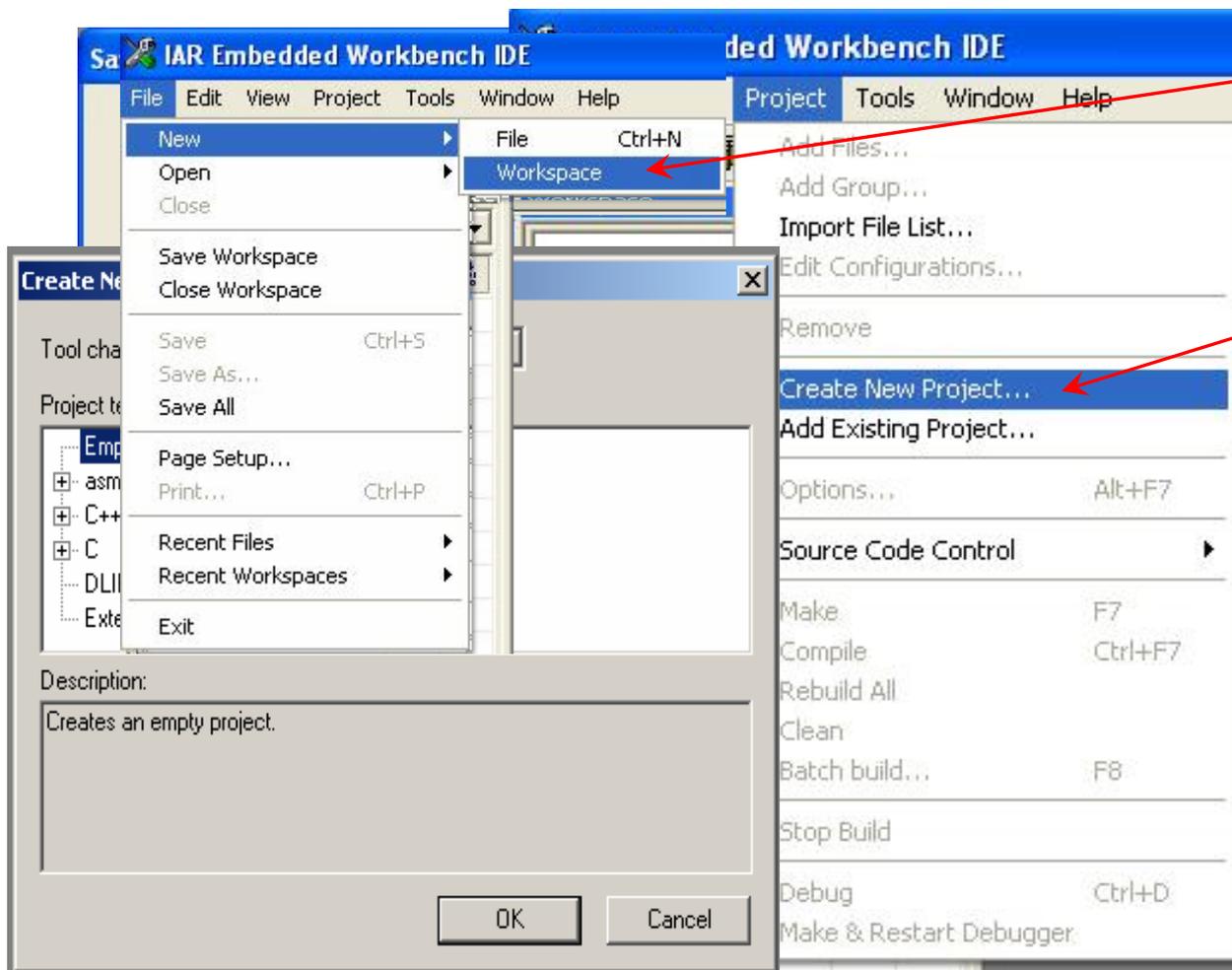
    // Clear ADC1 AWD pending interrupt bit
    ADC_ClearITPendingBit(ADC1, ADC_IT_AWD);
}

void TIM2_IRQHandler(void)
{
}
```

修改中断文件  
stm32f10x\_it.c

根据开启的中断，在对应的中断函数中添加代码。

# 步骤三：创建一个新的项目

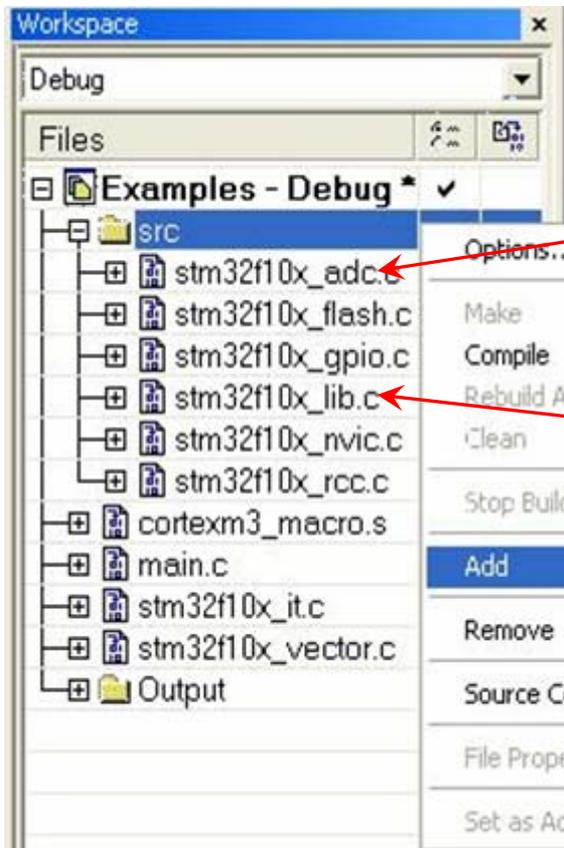


1. 点击菜单File-->New-->Workspace  
创建一个新的工程

2. 点击菜单Project-->Create New Project  
创建一个项目

3. 保存项目到设定的项目根目录中

# 步骤四：添加文件



向项目中添加文件：

根据stm32f10x\_conf.h中打开的外设，向项目中添加对应的PPP.c文件。

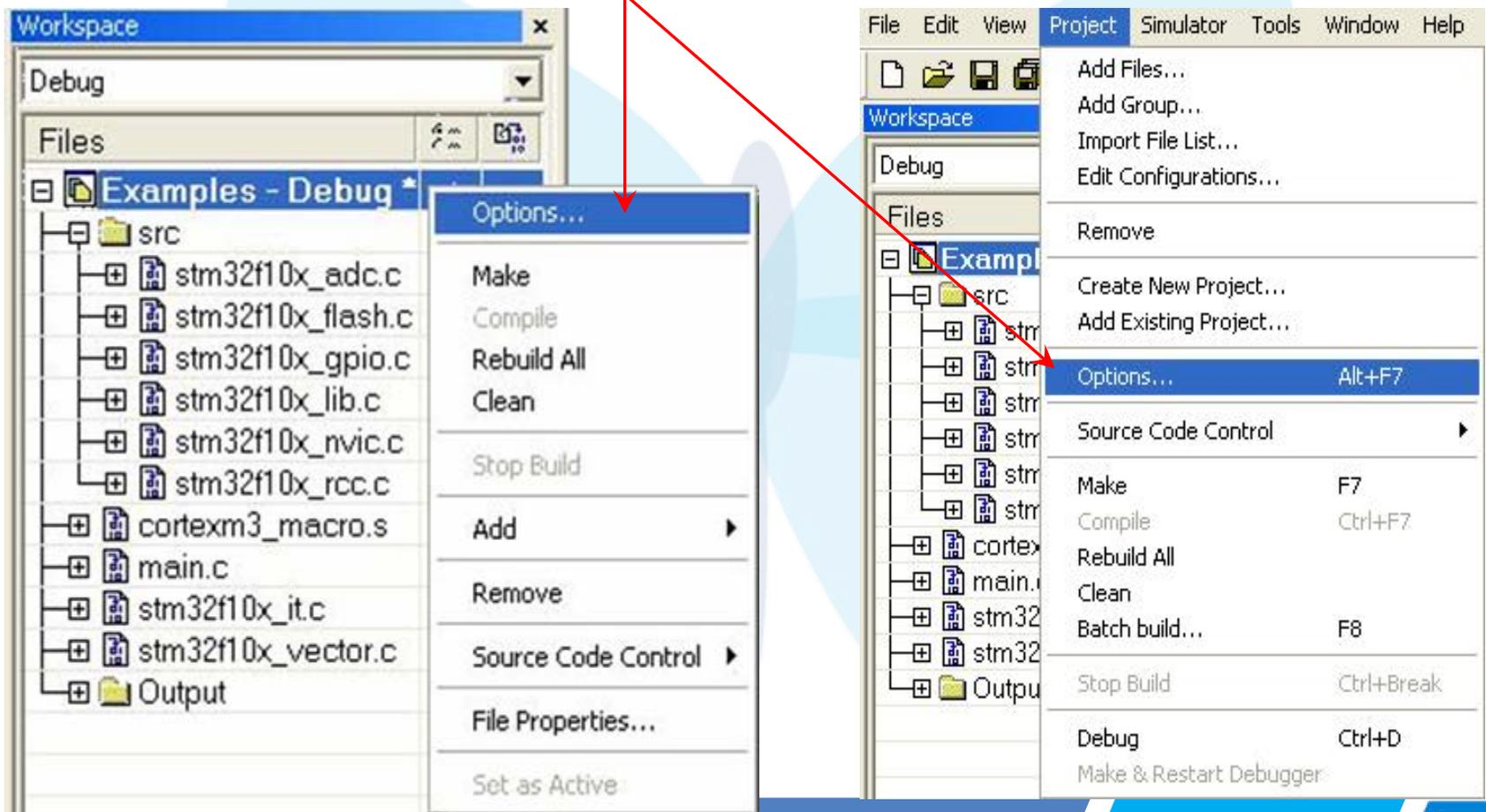
必须添加stm32f10x\_lib.c

用户可以向项目中添加\*.c文件；也可以添加代码组，然后将c文件添加到组里面：

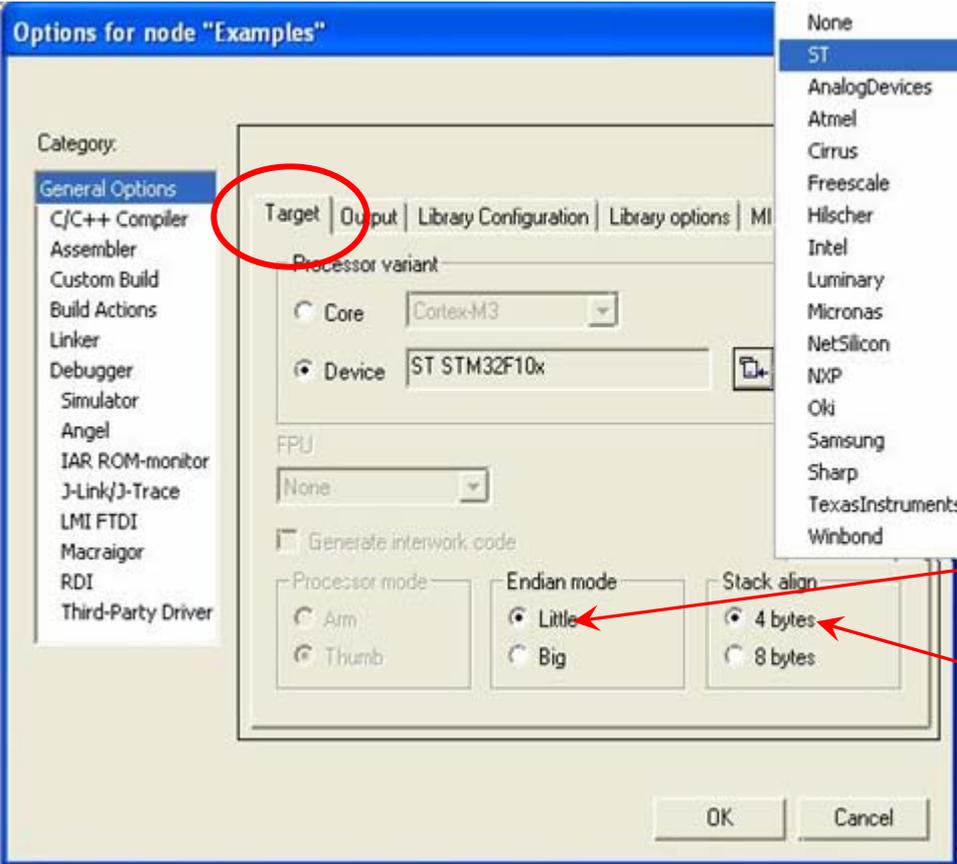
- 可以很好地进行源代码管理；
- 有助于生成较好的目标代码。

# 步骤五：Option设置(1)

在项目名称上单击右键选择Option，  
或点击菜单Project-->Option。



# 步骤五: Option设置(2): General Option



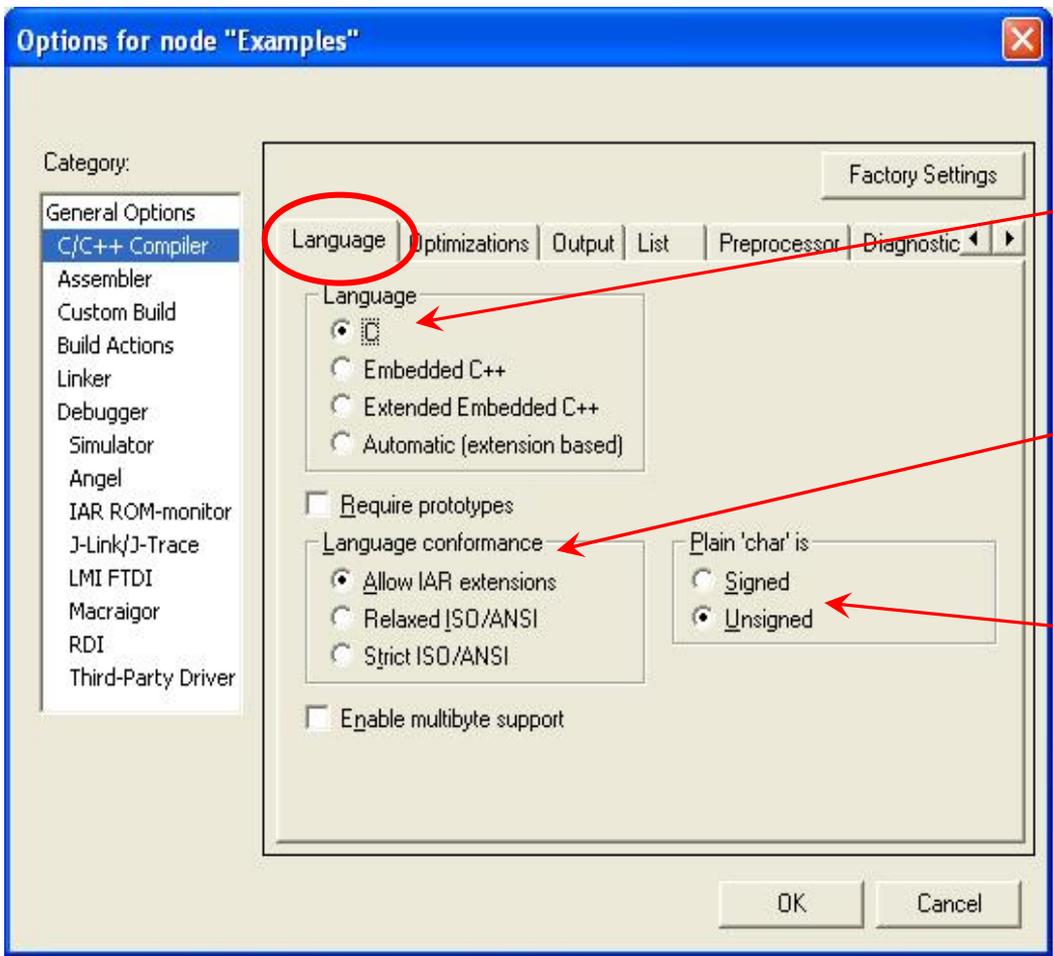
设置 Target

Processor Variant  
选择ST系列芯片:  
ST STM32F10x

Endian mode选择:  
Little

Stack align选择:  
4 bytes

# 步骤五: Option设置(3): C/C++ Compile A



设置 Language

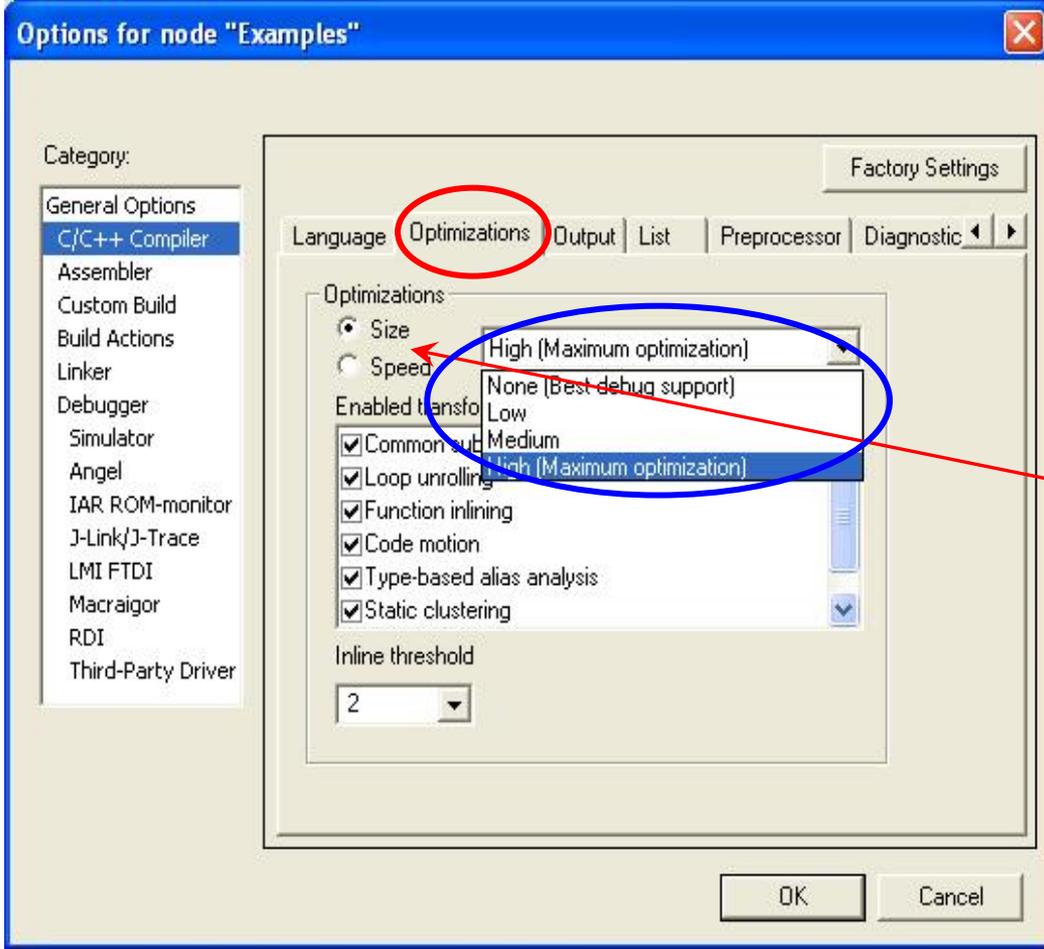
选择你使用的语言

选择使用语言的规范

Plain 'char' is:  
➤ Signed表明使用char定义的变量为有符号的;  
➤ Unsigned表明使用char定义的变量为有符号的。

# 步骤五: Option设置(3): C/C++ Compile B

 设置 Optimization

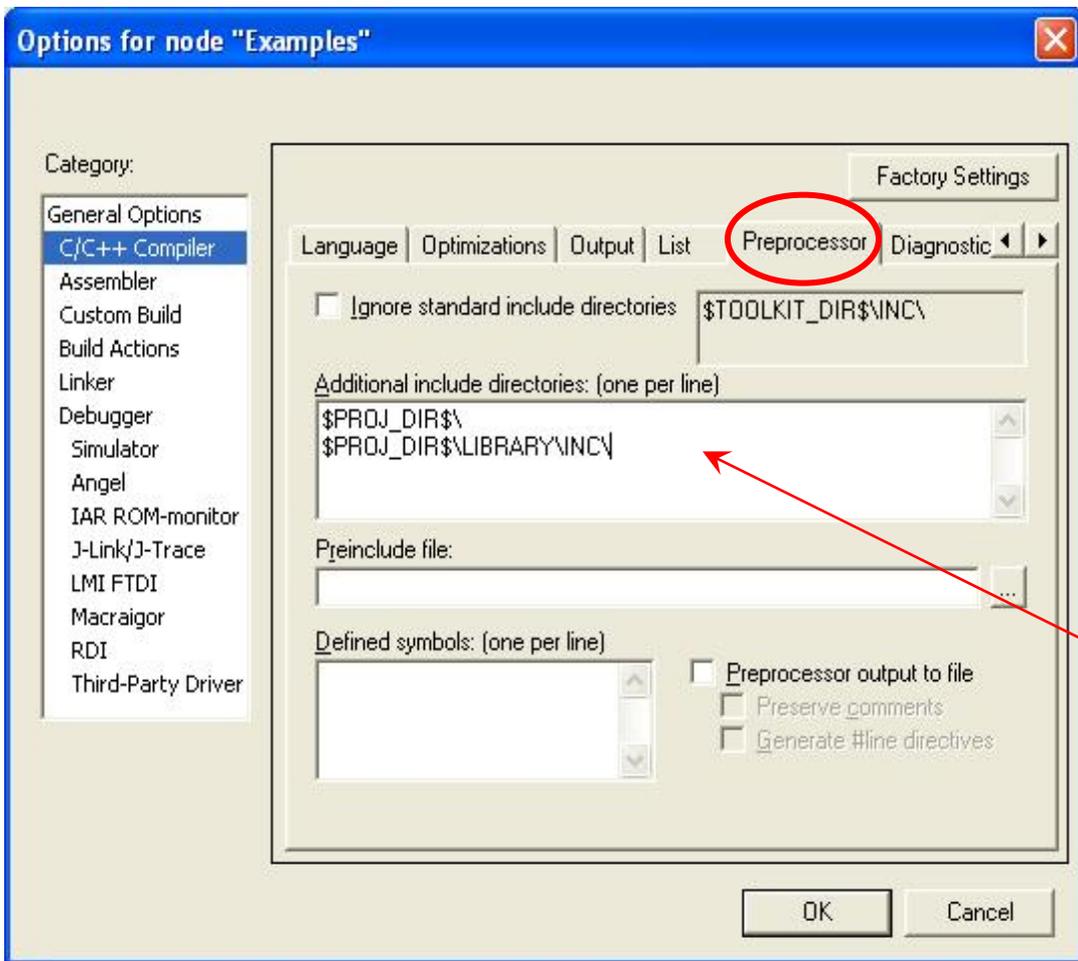


根据使用需求选择代码(Size)或运行速度(speed)优化:

1. None --- 用于调试
2. Low
3. Medium
4. High --- 用于最终代码

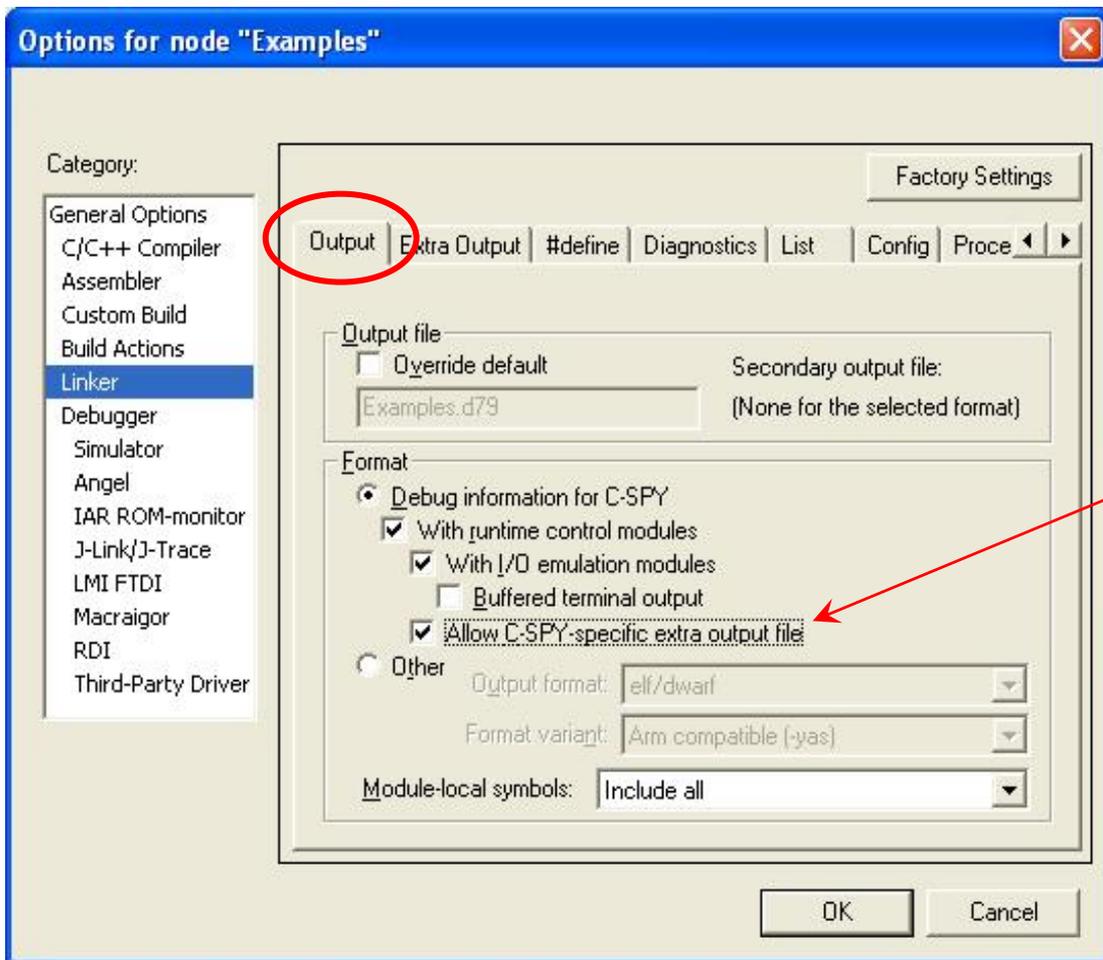
# 步骤五: Option设置(3): C/C++ Compile C

 设置 Preprocessor



设置\*.h文件所在的位置

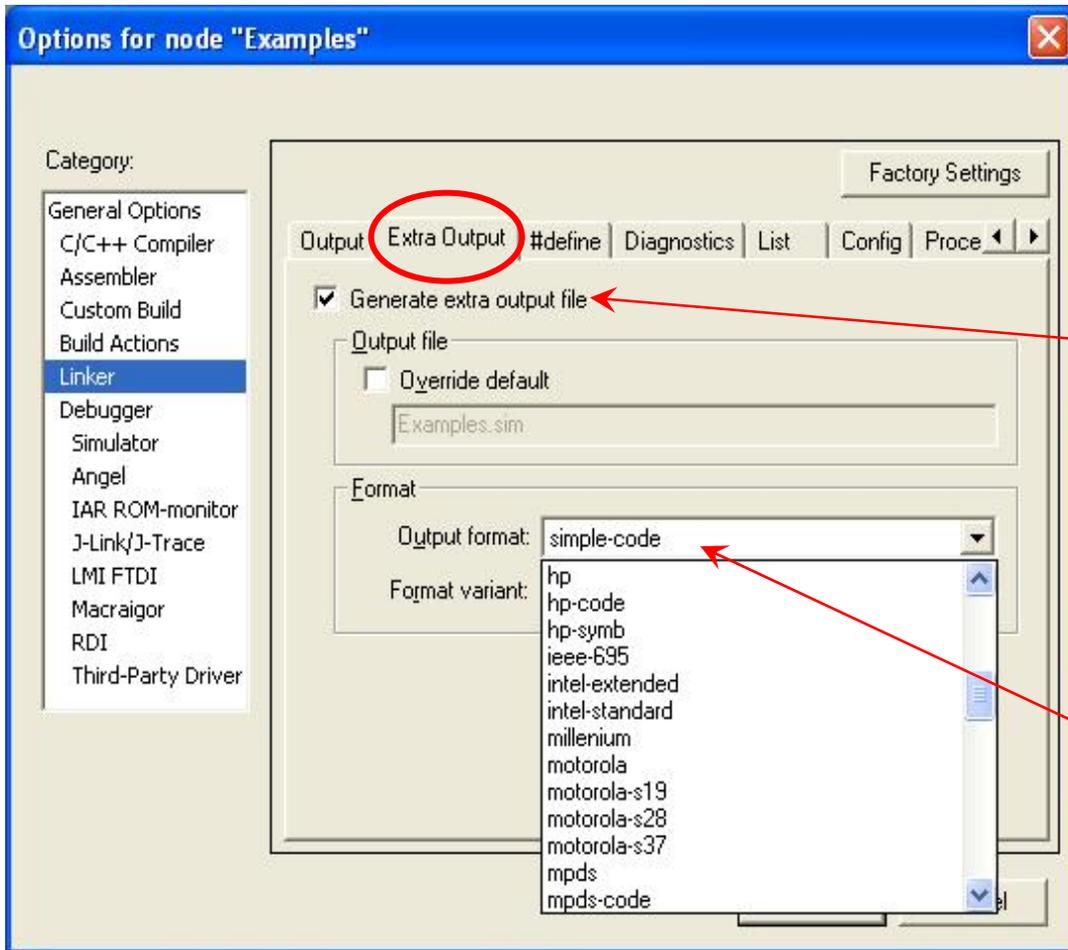
# 步骤五: Option设置(4): Linker A



设置: Output

开启C-SPY调试器

# 步骤五：Option设置(4)：Linker B

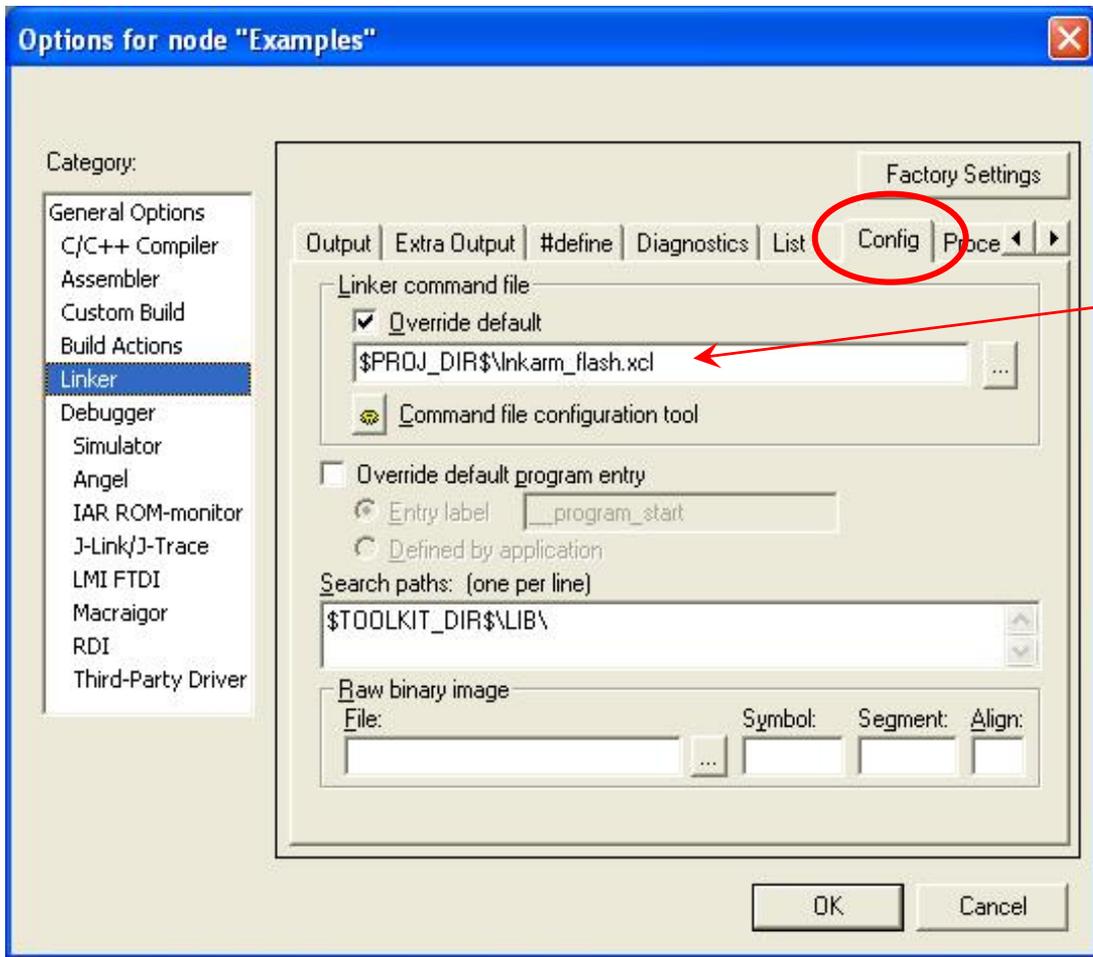


## 设置 Extra Output

打开“Generate the object file”选项；  
➤ 用于产生调试和目标文件

根据需求选择生成的目标文件；  
➤ 默认为IAR调试文件 simple-code。

# 步骤五: Option设置(4): Linker C



设置 Config

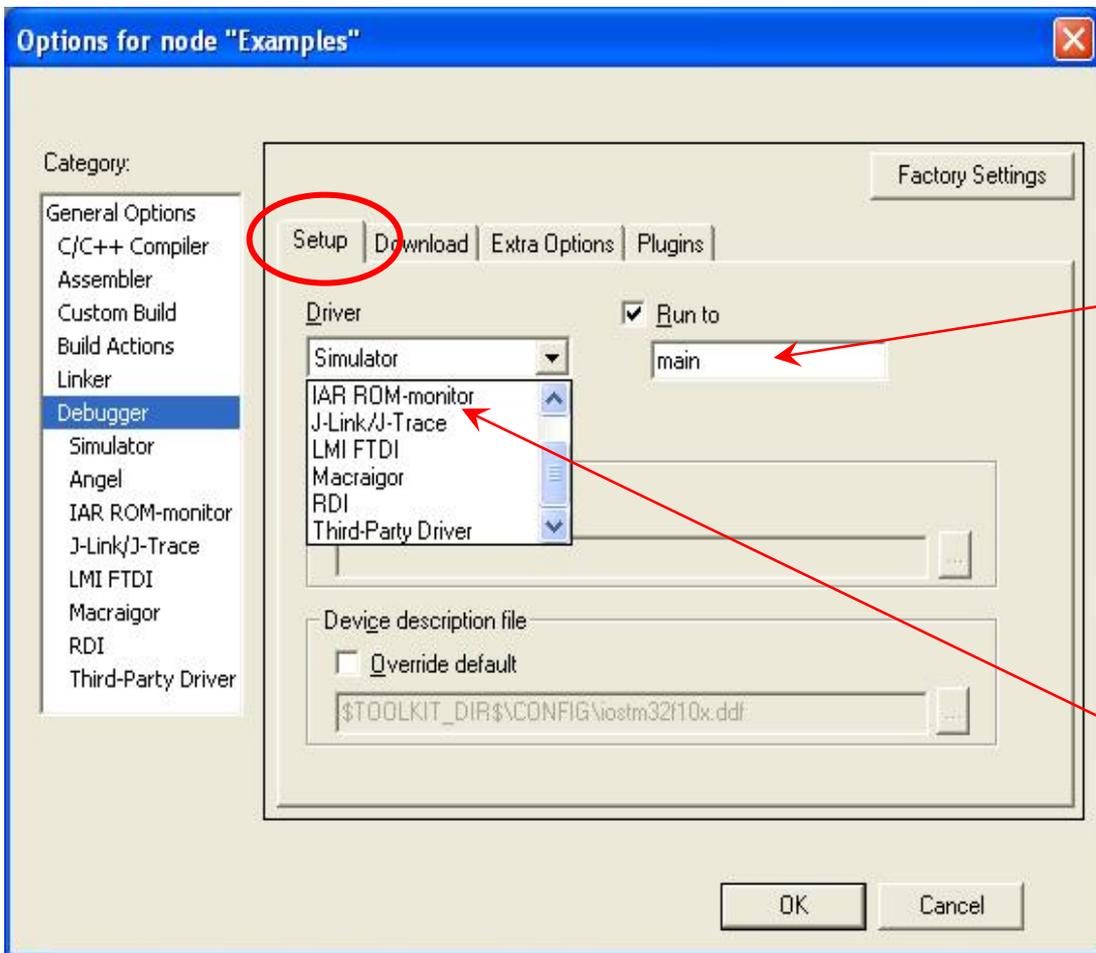
打开 Override default 选项:

- 如果在Flash中调试程序, 设置lnkarm\_flash.xcl;
- 如果在RAM中调试程序, 设置lnkarm\_RAM.xcl;

在进行该项设置时, 请确认电路板上的Boot0和Boot1引脚的跳线连接是否正确!

# 步骤五: Option设置(5): Debugger A

设置 Setup

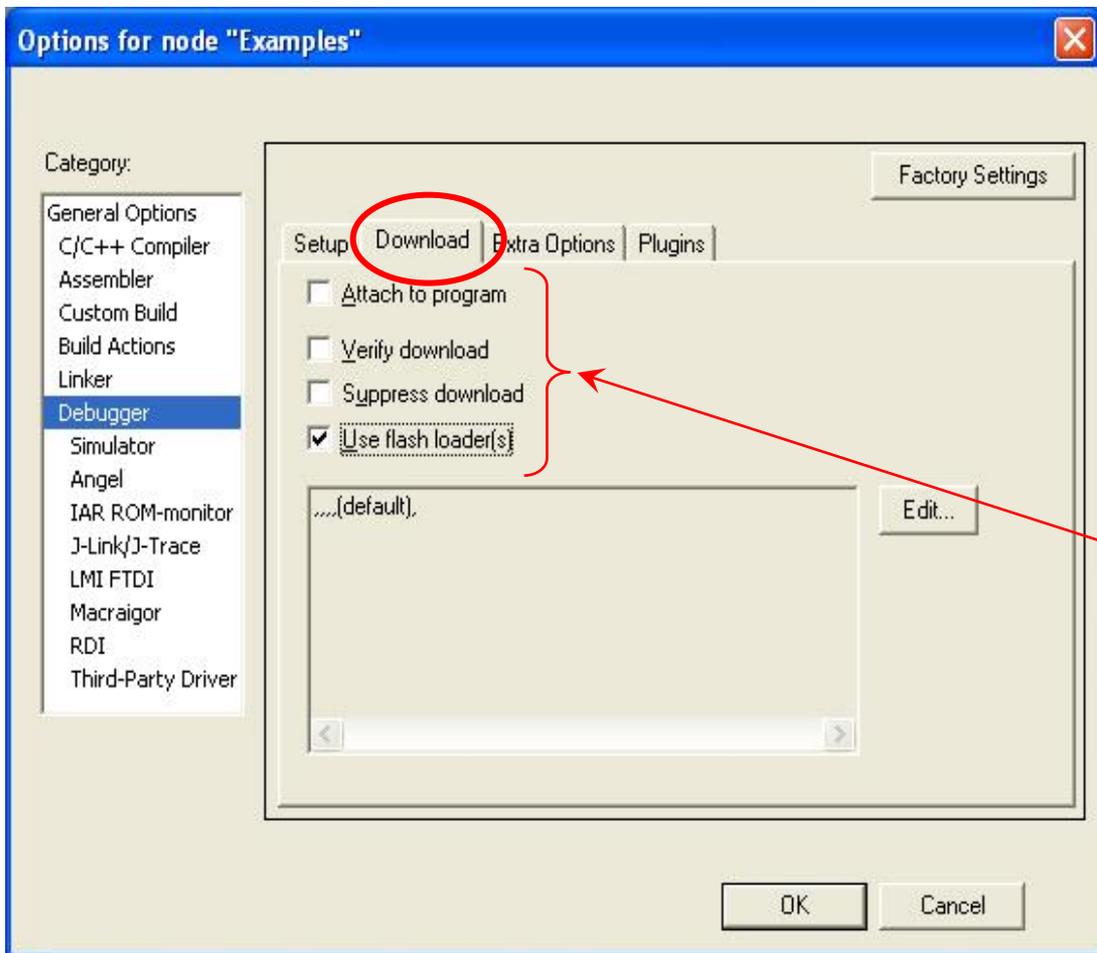


开启该选择, 可以  
设定调试起始点。

选择使用的调试工具

# 步骤五: Option设置(5): Debugger B

 设置 Download



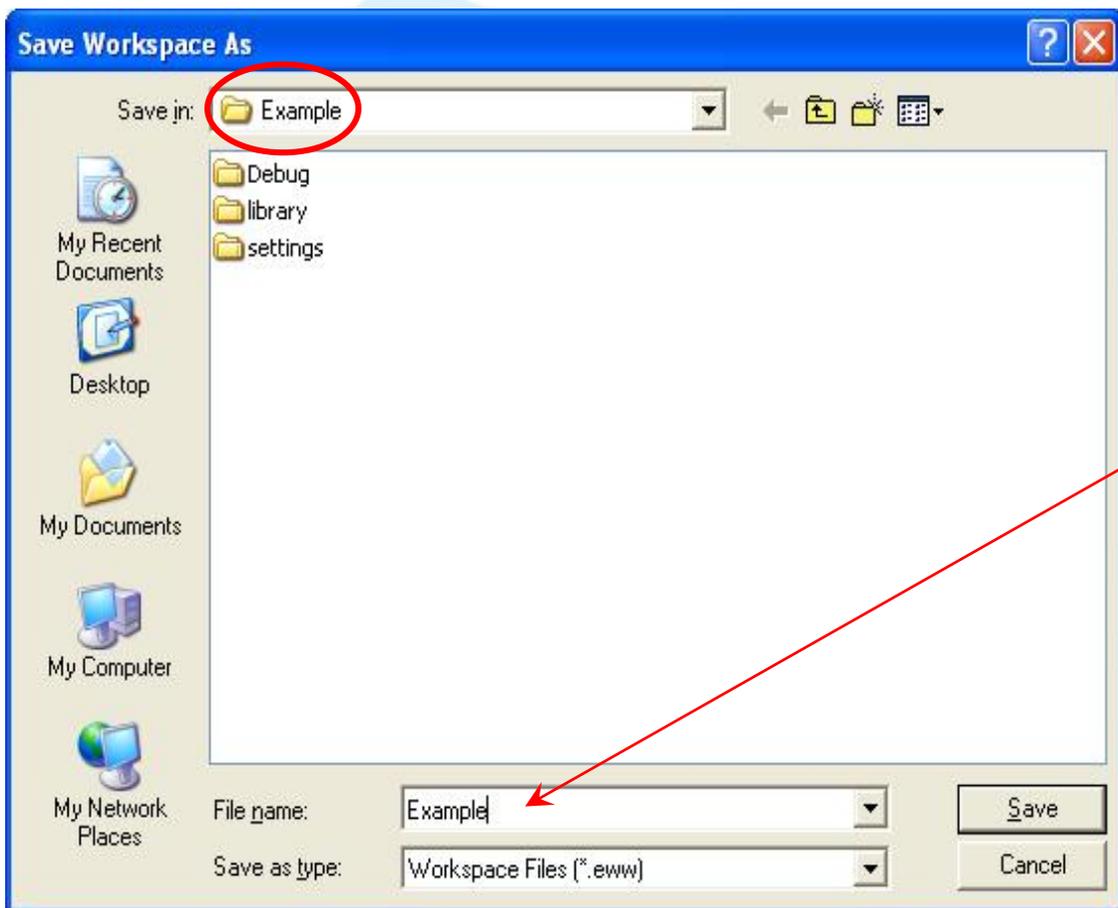
1. 已选择下载程序到Flash, 可选:

- Verify download
- Use flash loader(s) (必须开启!)

2. 已选择下载程序到RAM或使用模拟器: 无需选择。

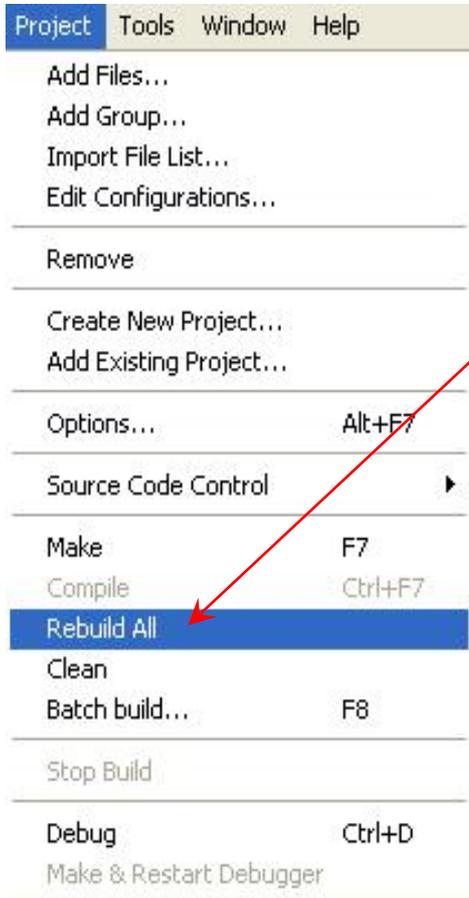
3. 如果程序已下载到Flash:  
➤ 开启Suppress download

# 步骤六：保存Workspace



保存工程文件  
到项目目录中

# 步骤七：编译项目



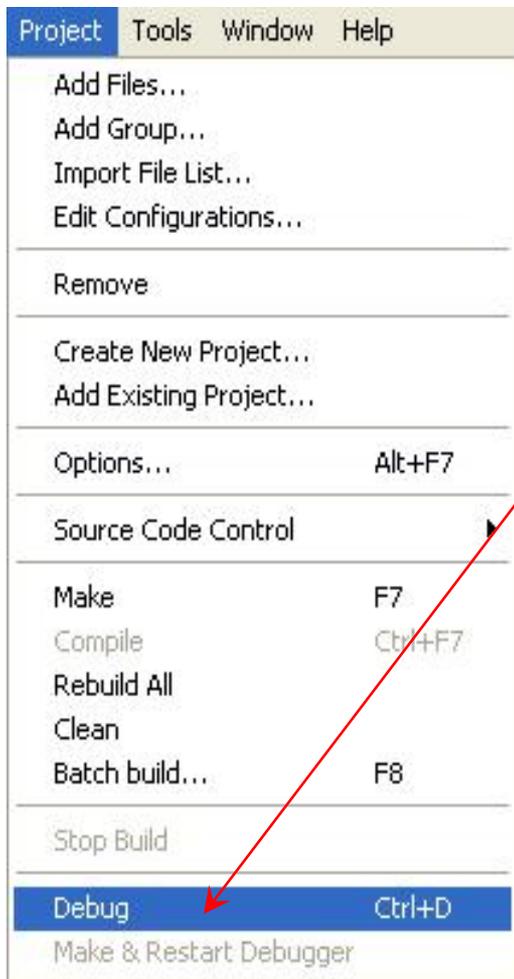
使用菜单Project-->Rebuild All编译项目



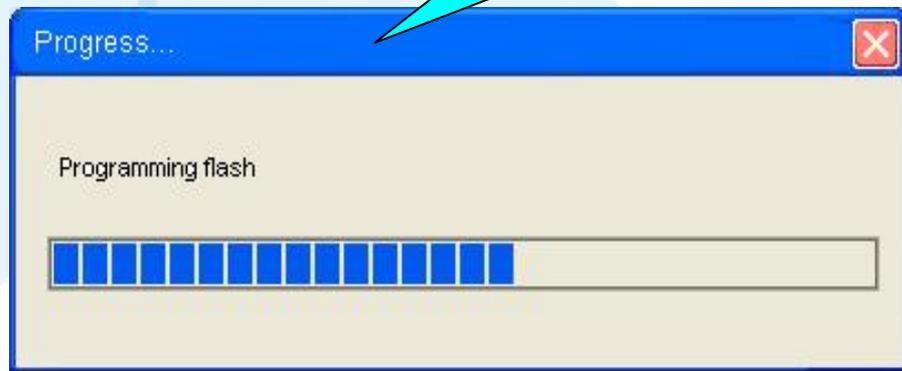
在Message窗口查看编译结果；如果有错误双击进行修改。

# 步骤八：调试(一)

使用菜单Project-->Debug或单击Debug按钮进入调试状态



如果在Flash中调试程序时，出现下载进程对话框



# 步骤七: Debug (二)

The screenshot displays the IAR Embedded Workbench IDE interface. On the left, the **Debug** menu is open, showing options such as Go (F5), Break, Reset, Stop Debugging, Step Over (F10), Step Into (F11), Step Out (Shift), Next Statement, Run to Cursor, Autostep..., Refresh, Set Next Statement, Macros..., and Logging. A red callout box with Chinese text points to the Debug menu and the workspace area. The main workspace shows the source code for `stm32f10x_gpio.c` with the `main` function highlighted. The **Disassembly** window shows the assembly code for the `main` function, and the **Register** window shows the state of the `ADC1` register.

使用菜单Debug中的调试菜单或工具条进行调试

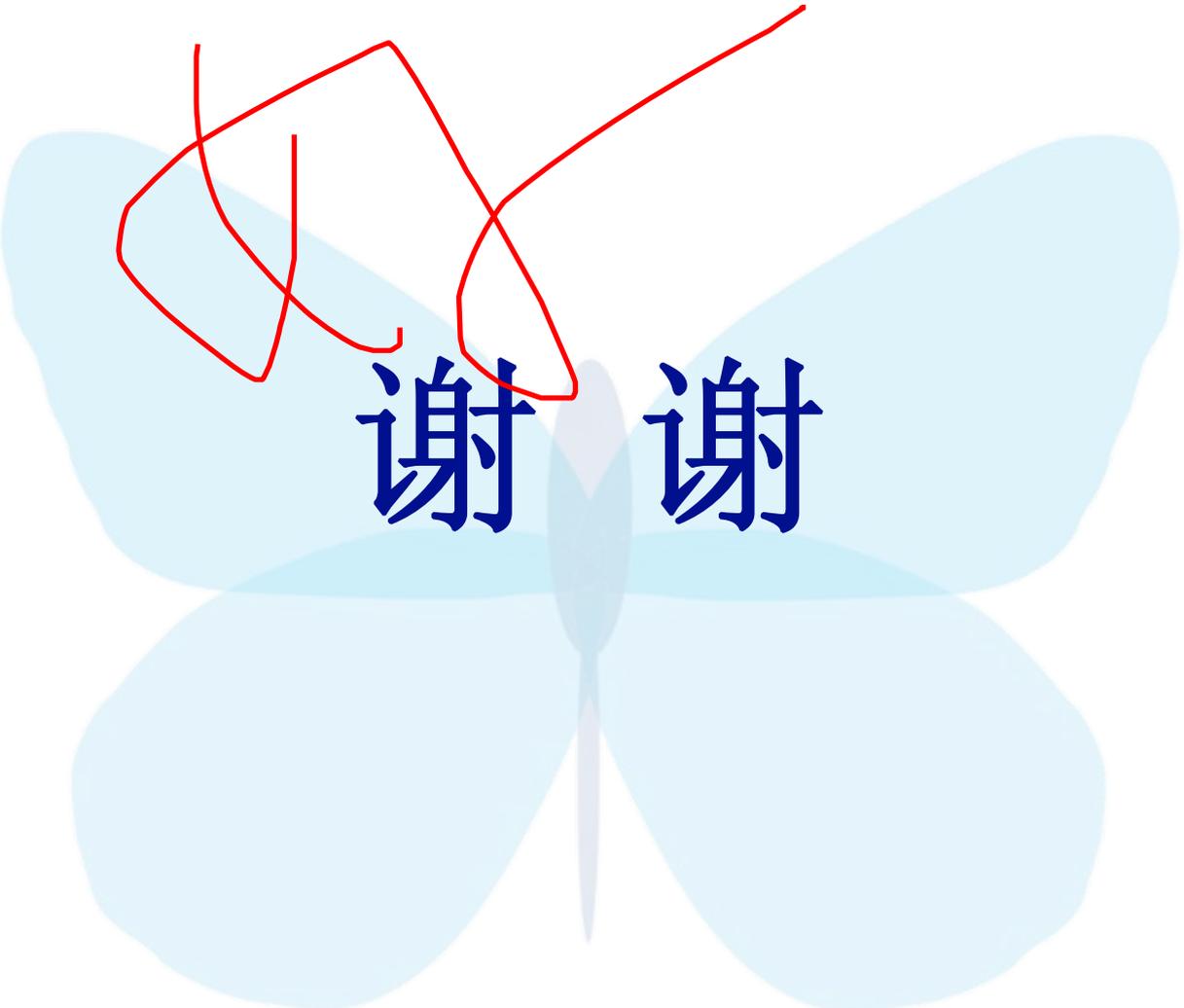
STM32 32位微控制器

STM32 释放您的创造力

2007年12月 22



[www.stmicroelectronics.com.cn/mcu](http://www.stmicroelectronics.com.cn/mcu)



谢谢

STM32 32位微控制器

2007年12月 23



STM32 释放您的创造力

[www.stmicroelectronics.com.cn/mcu](http://www.stmicroelectronics.com.cn/mcu)