



# CW32F030x6 / x8 用户手册

---

CW32 系列微控制器软件开发工具入门

---

版本号: Rev 1.1



# 引言

本文档将指导初学者如何构建和运行应用程序示例，以及构建和调试自己的应用程序。其结构如下：

- 第一章描述软件和硬件要求（一些支持 CW32 系列的工具链、CW-LINK 安装和固件包内容）。
- 第二章提供关于如何使用一些工具链构建和调试应用程序的分步指南：
  - IAR systems® 公司开发的 IAR Embedded Workbench® for ARM®（EWARM）
  - Keil® 公司开发的 Microcontroller Development Kit for ARM®（MDK-ARM）

本手册并未涵盖与软件开发环境相关的所有主题，但是演示了编译器 / 调试器入门所需的基本步骤，并提供了全面理解每个步骤所需的文档的链接。



# 目录

引言 .....	1
1 系统要求 .....	3
1.1 固件包 .....	4
2 支持 CW32F 系列的 IDE .....	5
2.1 EWARM 工具链 .....	5
2.1.1 EWARM V9.30 或更高版本 .....	5
2.1.2 EWARM V9.30 以下版本 .....	13
2.2 MDK-ARM 工具链 .....	19
3 版本信息 .....	24

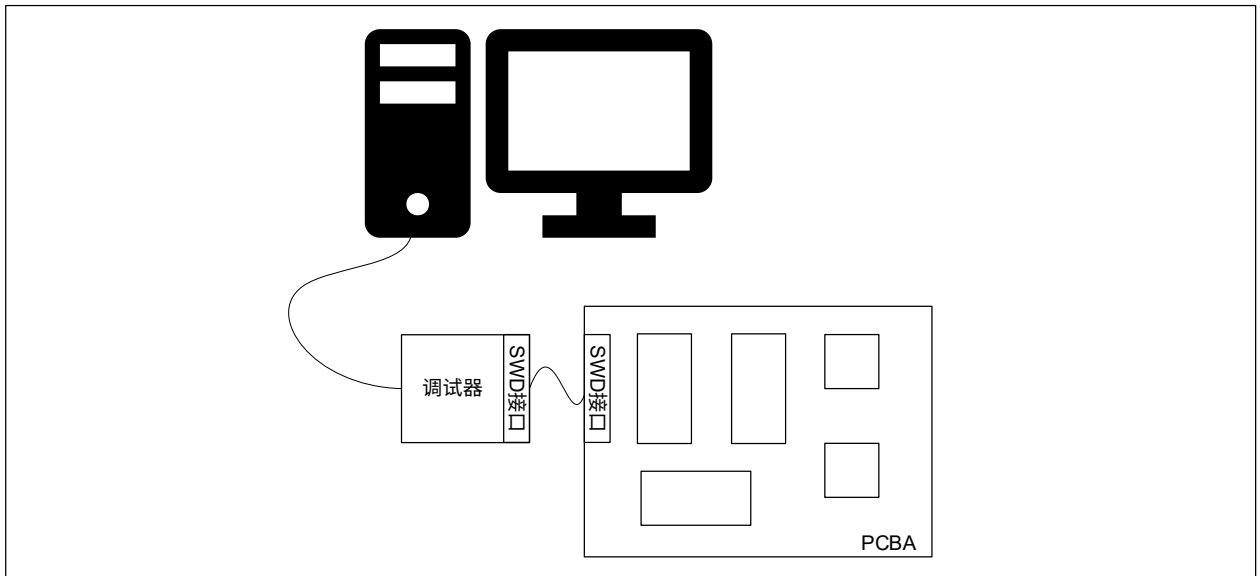


# 1 系统要求

开始前，用户应：

1. 安装首选的 Integrated Development Environment (IDE) 。
2. 安装调试器 CW-LINK 的驱动程序。
3. 下载 CW32F 系列固件。
4. 为了与 CW32F 开发板建立连接，将开发板的 SWD 接口与调试器 CW-LINK 连接，并将调试器 CW-LINK 与 PC 连接，如下图：

图 1-1 调试硬件连接

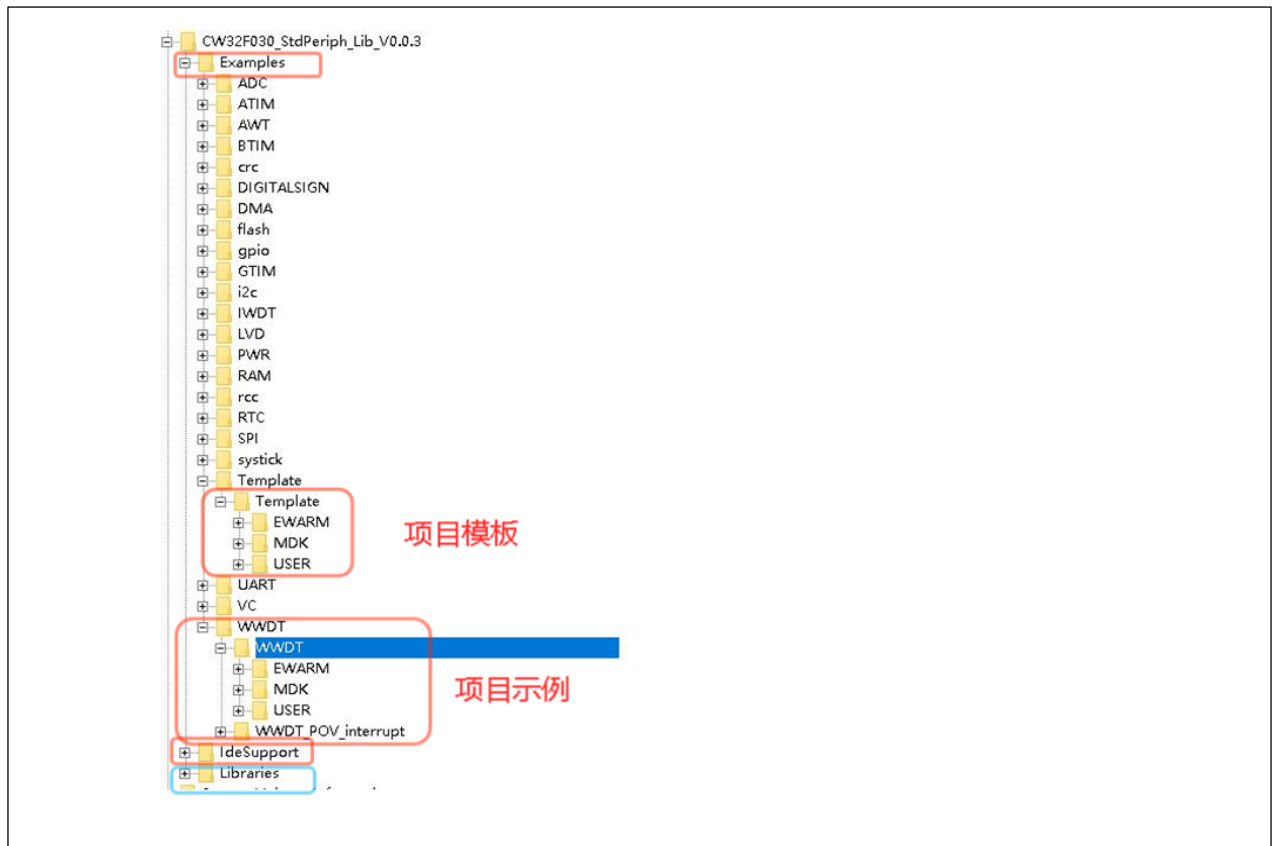




## 1.1 固件包

CW32F 系列的驱动、示例和 IDE 支持文件包含在同一个 .zip 文件中。解压缩 .zip 文件生成一个文件夹，即 CW32FXXX\_StdPeriph\_Lib\_VX.Y.Z，其中包含以下子文件夹（参见下图）：

图 1-2 固件包结构图



其中：

- Examples 包含每个可以直接运行的外设示例的工程。
- Template 是一个预配置项目，其主要功能为空，将由用户自定义。它可以根据外设驱动程序创建应用程序。
- IdeSupport 包含对 KEIL 和 IAR 的 IDE 软件的支持的插件。
- Libraries 包含一组外设的驱动程序。

## 2 支持 CW32F 系列的 IDE

支持 CW32F 系列的工具链：

- EWARM v7.70 或更高版本
- MDK-ARM v5.17 或更高版本

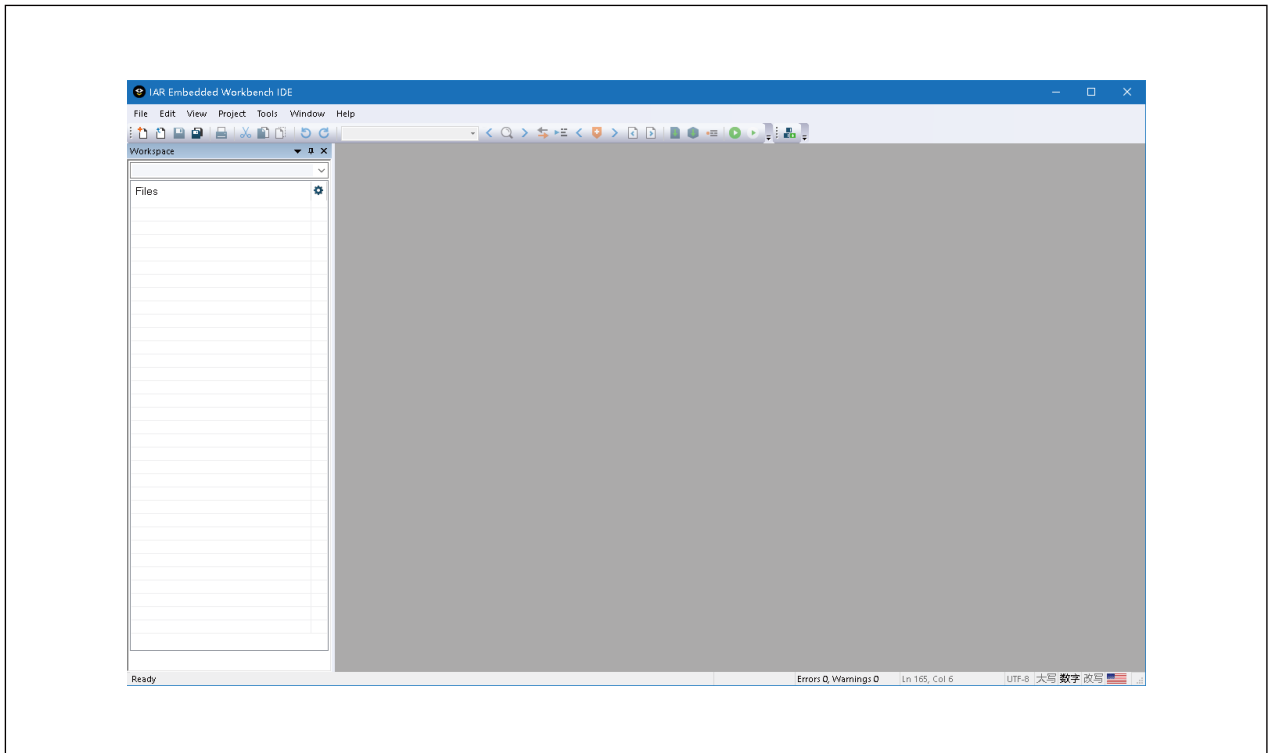
### 2.1 EWARM 工具链

#### 2.1.1 EWARM V9.30 或更高版本

从 EWARM V9.30 版本开始，IAR 已全面支持芯源半导体的 MCU。通过 EWARM 可以快速的对 CW32 系列 MCU 进行开发。步骤如下：

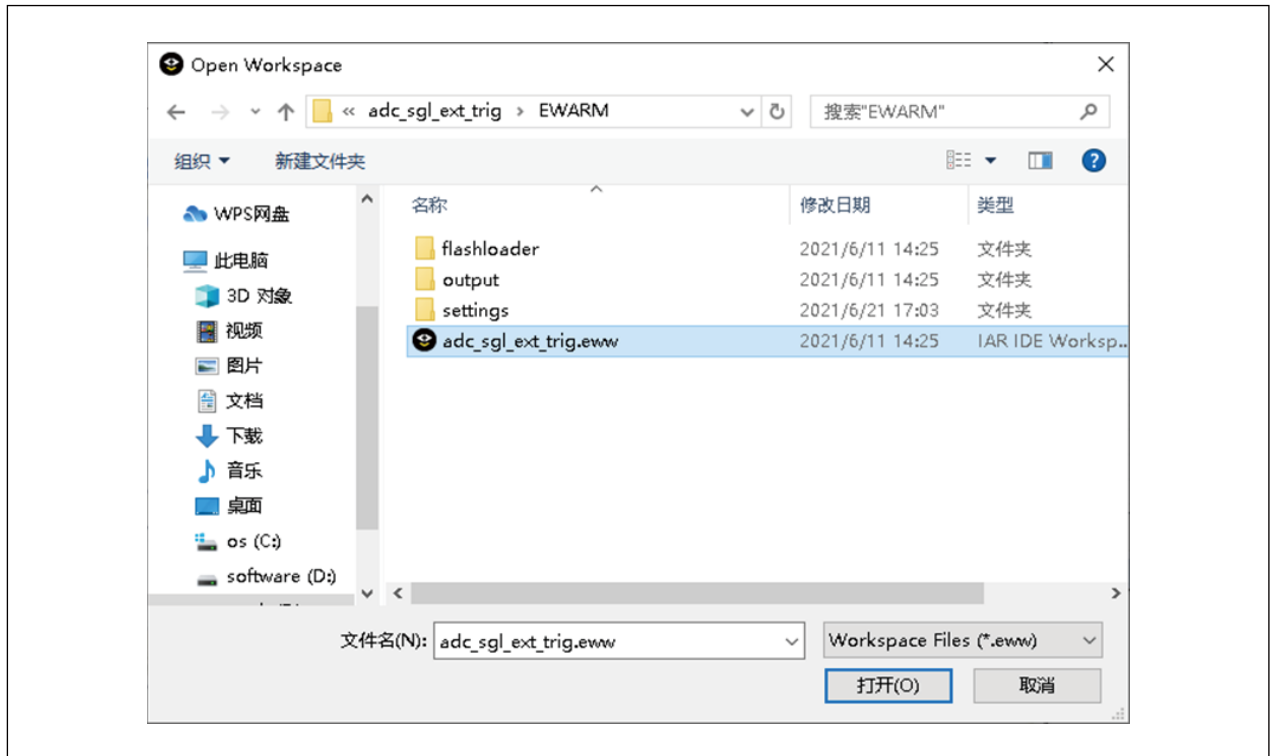
1. 启动 IAR Embedded Workbench for ARM. 如图 2-1:

图 2-1



2. 在“File”菜单中，选择“Open Workspace”以显示“Open Workspace”对话框。浏览并选中一个示例工作区文件，点击“打开”启动它，如图 2-2：

图 2-2



- 在“Project”菜单中，选择“Options”以显示“Option.”对话框，并选择“GeneralOptions”项，如下图 2-3；选择“Device”项，并从“WHXY”器件目录下选择相应的器件，如图 2-4：

图 2-3

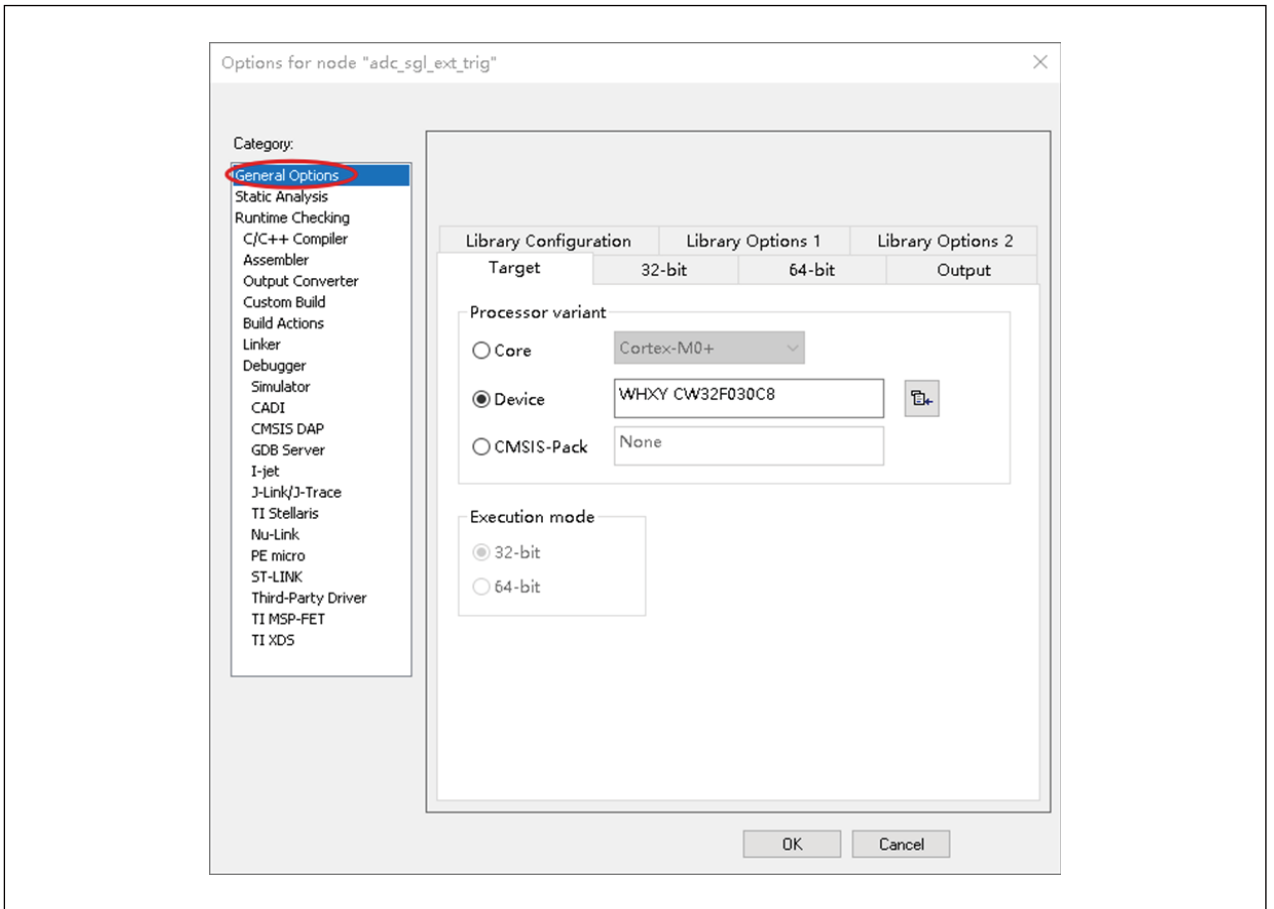
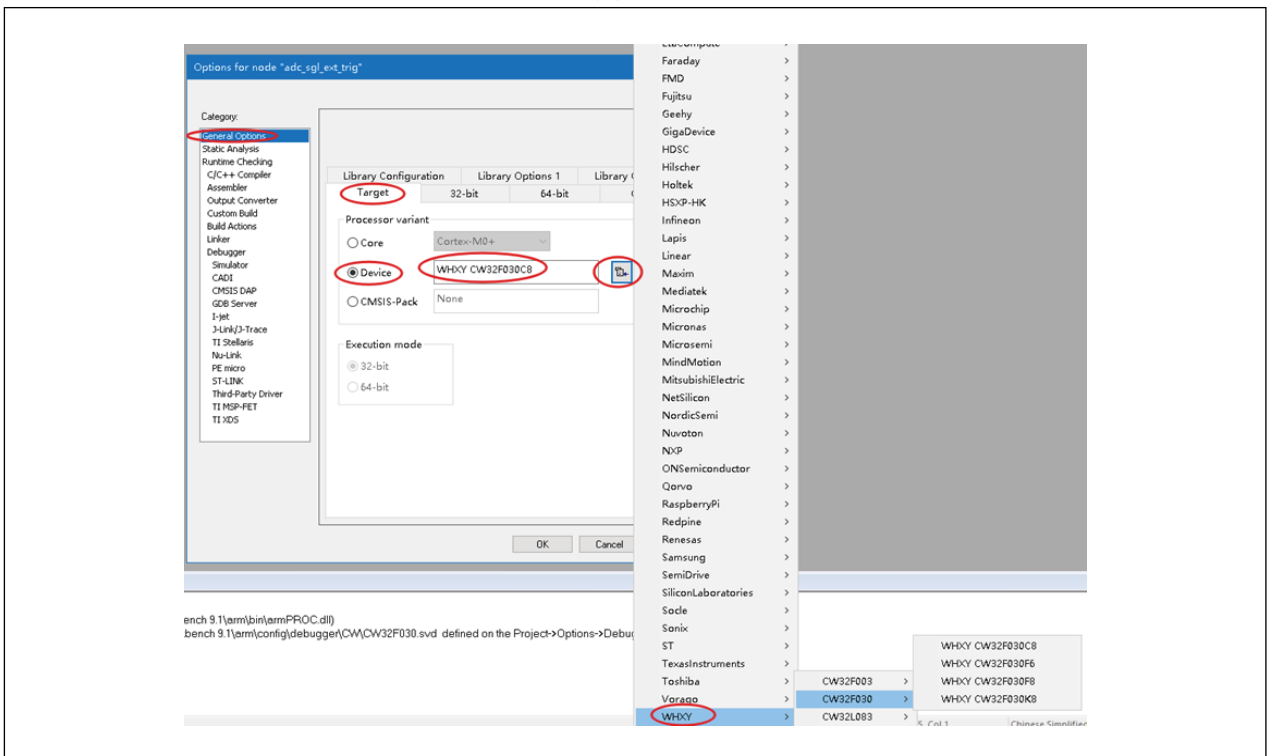
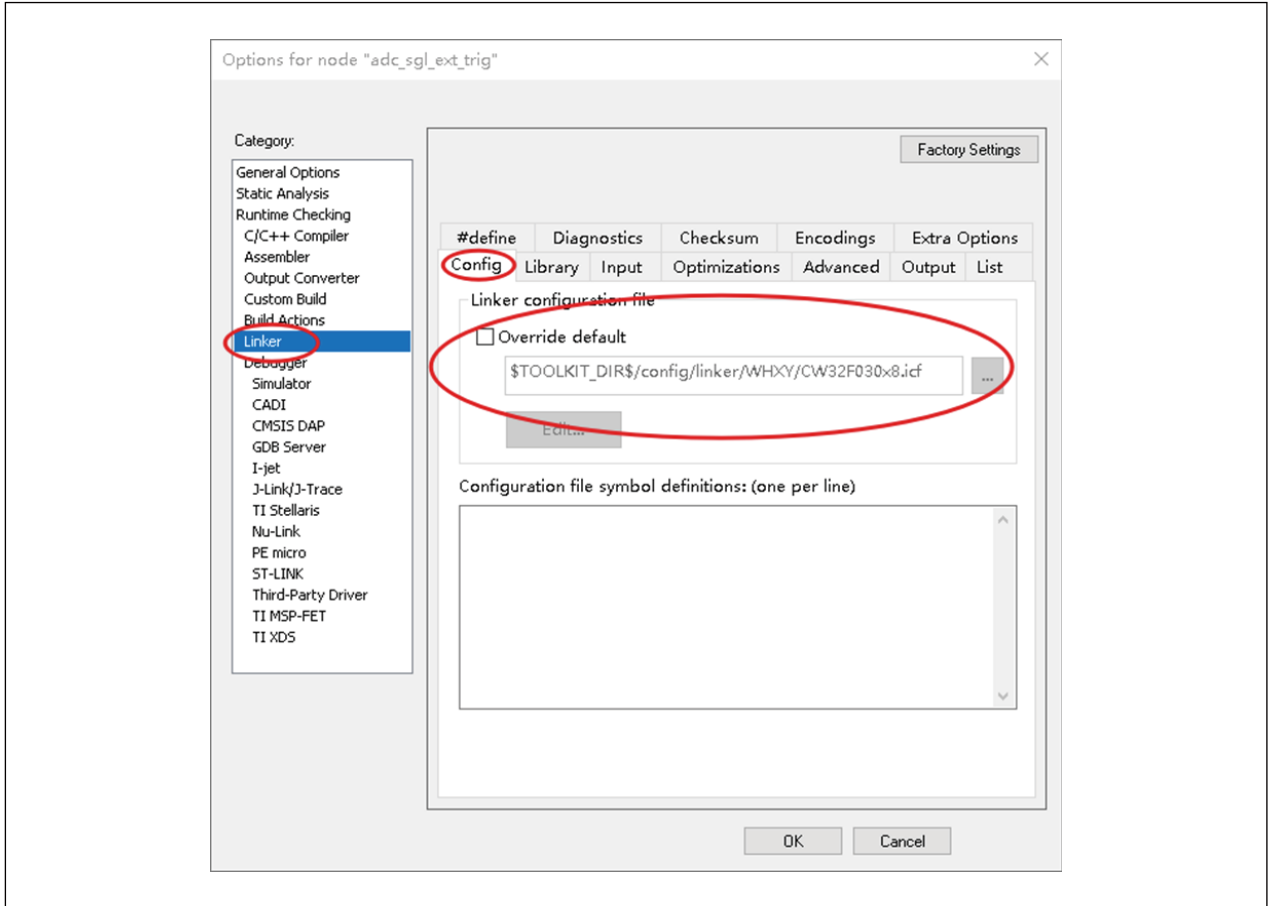


图 2-4



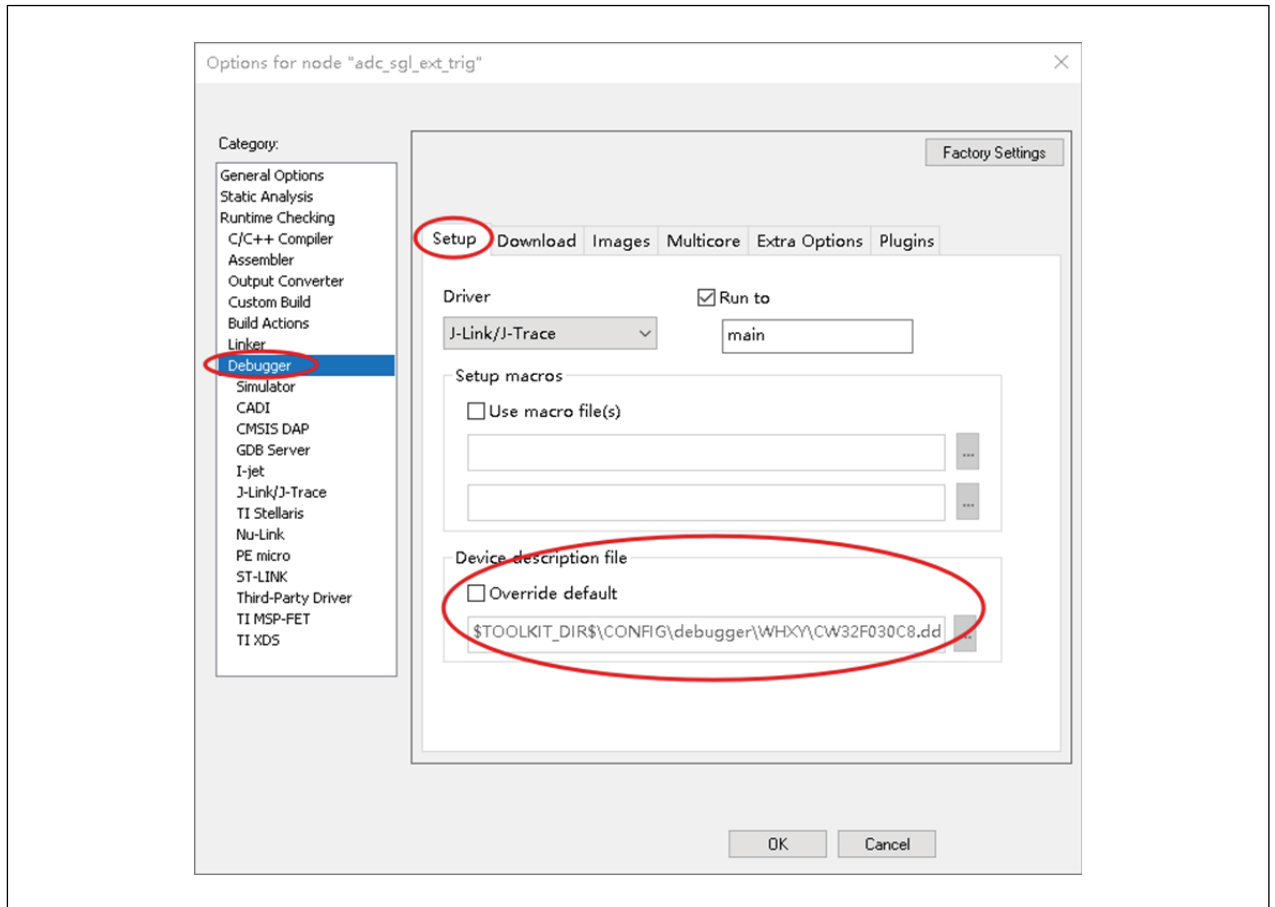
4. 在“Project”菜单中，选择“Options”以显示“Option.”对话框，并选择“Linker”项，确认“Config”标签中“Linker configuration file”为默认值，如图 2-5：

图 2-5



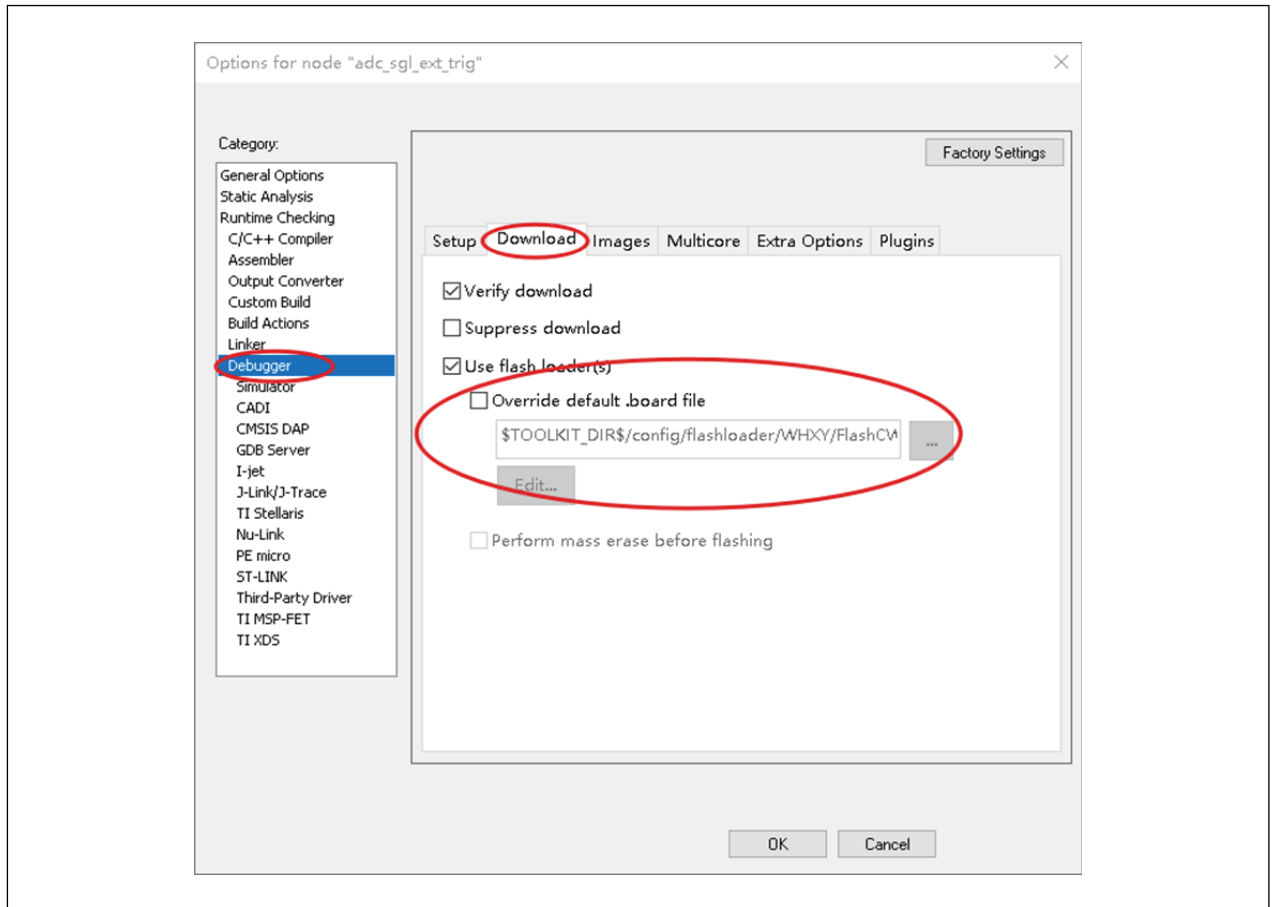
5. 在“Project”菜单中，选择“Options”以显示“Option.”对话框，并选择“Debugger”项，确认“Setup”标签中“Device description file”为默认值，如图 2-6：

图 2-6



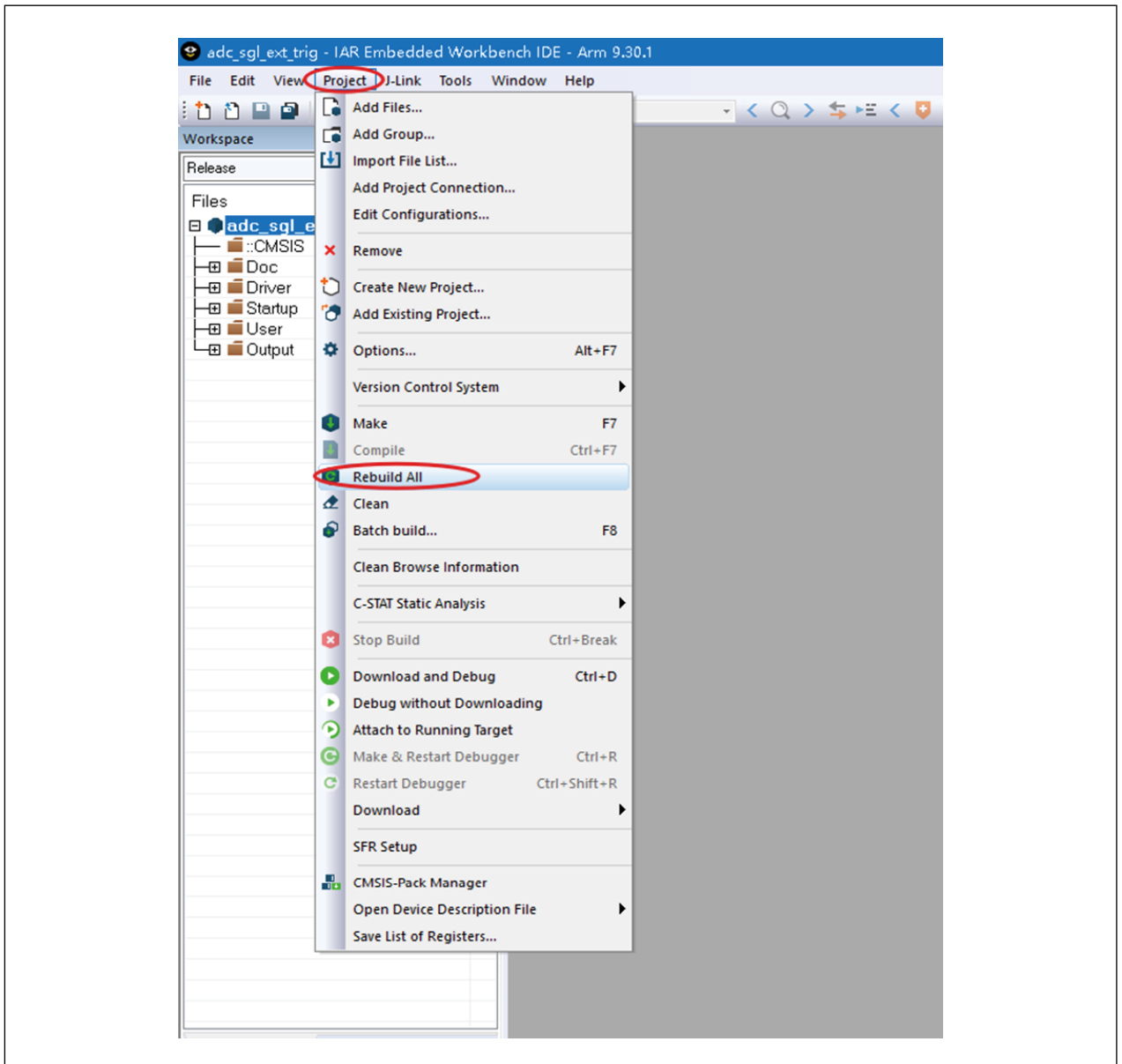
- 在“Project”菜单中,选择“Options”以显示“Option.”对话框,并选择“Debugger”项,确认“Download”标签中“Override default .board file”为默认值,如图 2-7:

图 2-7



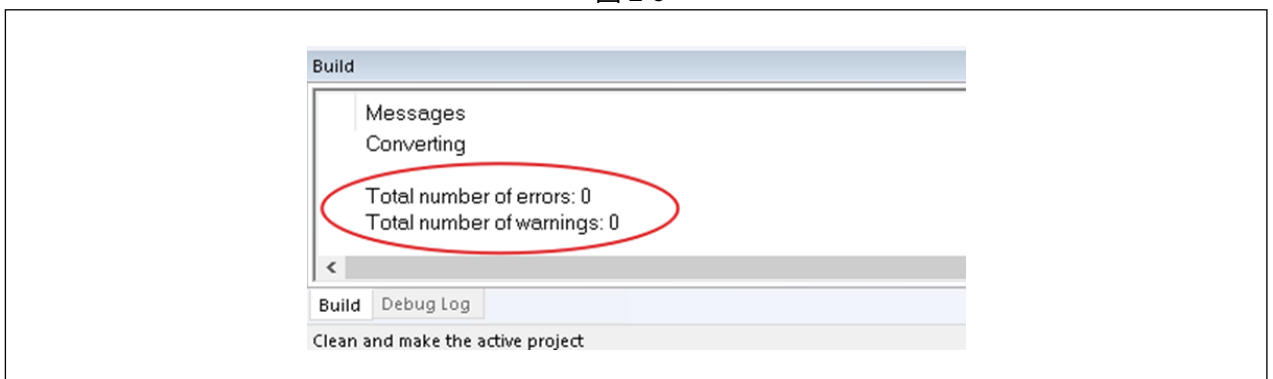
7. 在“Project”菜单中，选择“Rebuild All”以编译项目，见图 2-8：

图 2-8



8. 如果编译成功，将显示如图 2-9 所示的内容：

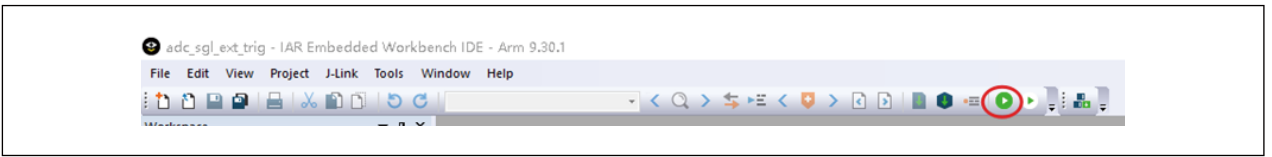
图 2-9





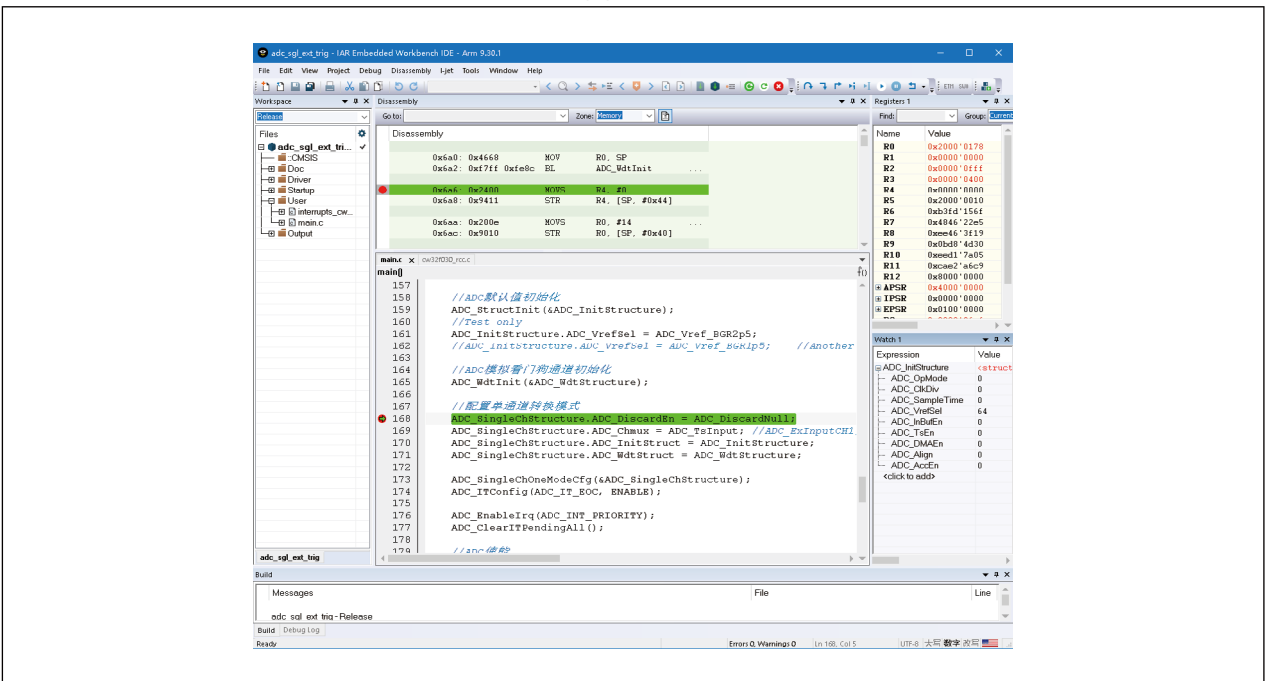
- 在 IAR Embedded Workbench IDE 中, 从项目菜单中选择下载和调试或点击工具栏中的下载和调试按钮, 以对 Flash 存储器进行编程并开始调试, 如图 2-10:

图 2-10



- IAR Embedded Workbench 中的调试器可用于用 C 语言和汇编语言调试源代码, 设置断点, 以及监控各个变量和代码执行过程中发生的事件, 如图 2-11:

图 2-11



### 2.1.2 EWARM V9.30 以下版本

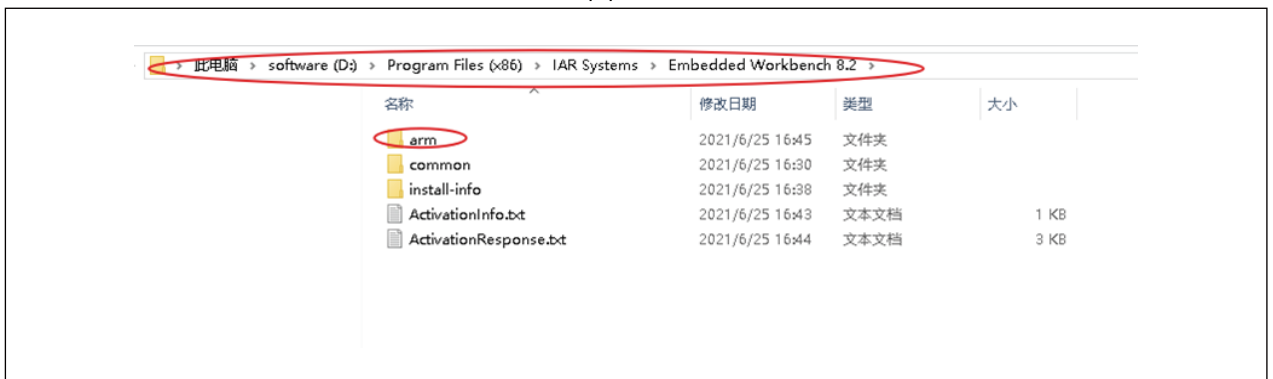
(以 CW32F030 系列为例。)

1. 复制固件包“IdeSupport\EWARM\”目录下的 arm 文件夹（如图 2-12）到 EWARM 的安装路径下。  
如 EWARM 安装路径为 D:\Program Files (x86)\IAR Systems\Embedded Workbench 8.2，则将 arm 文件夹拷贝到 D:\Program Files (x86)\IAR Systems\Embedded Workbench 8.2\ 目录下，覆盖同名的 arm 文件夹，如图 2-13：

图 2-12

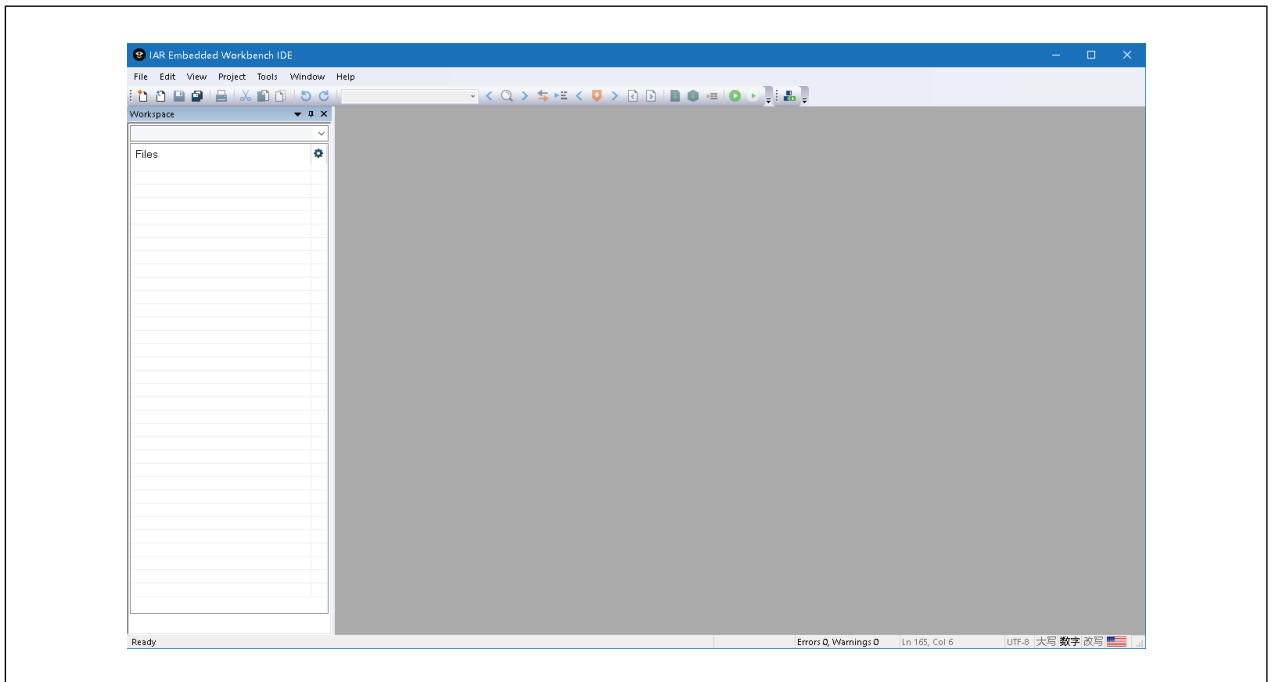


图 2-13



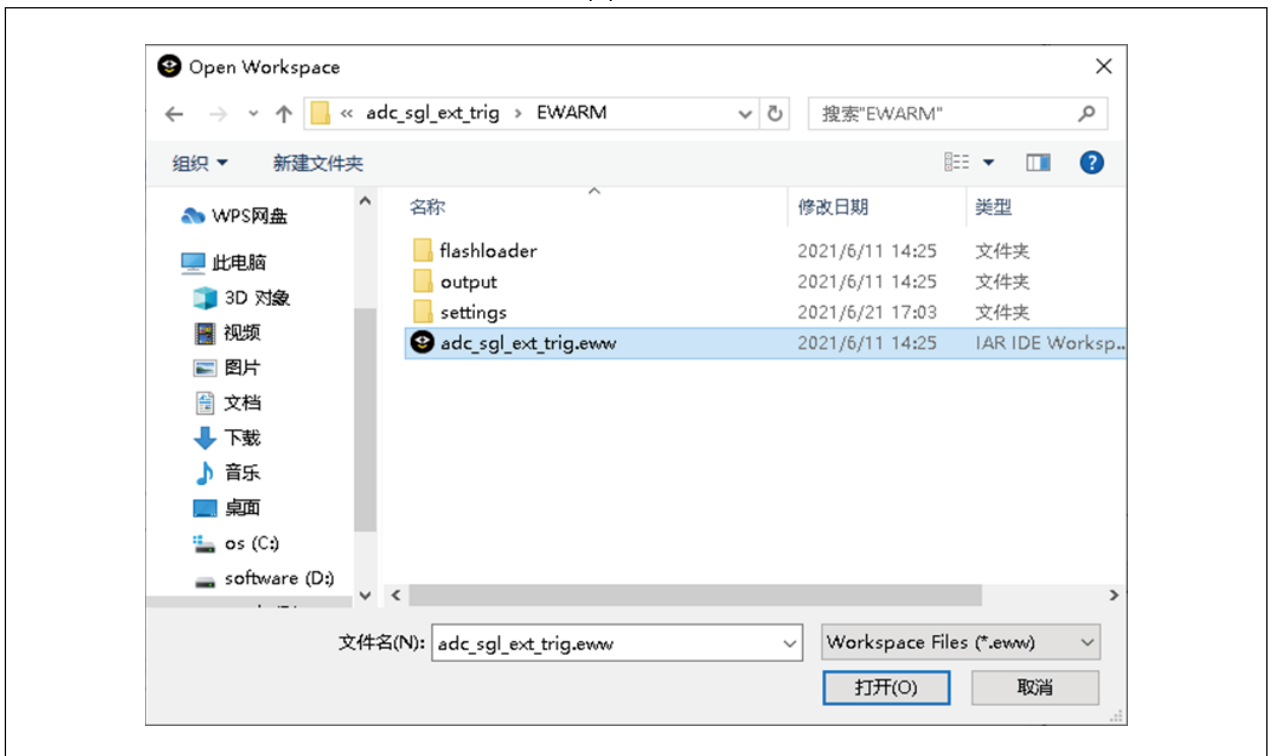
2. 启动 IAR Embedded Workbench for ARM，如图 2-14：

图 2-14



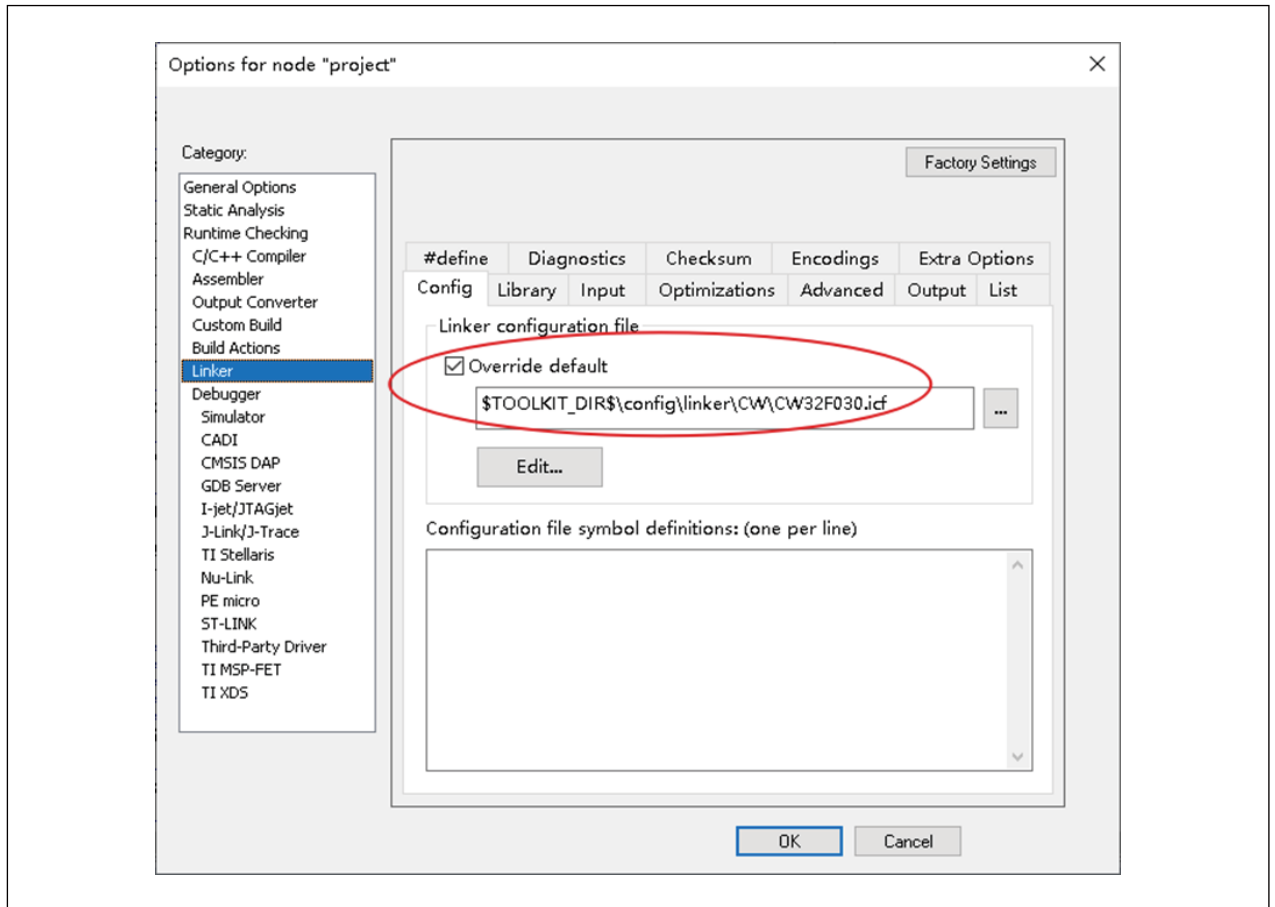
3. 在“File”菜单中，选择“Open Workspace”以显示“Open Workspace”对话框。浏览并选中一个示例工作区文件，点击“打开”启动它，如图 2-15：

图 2-15



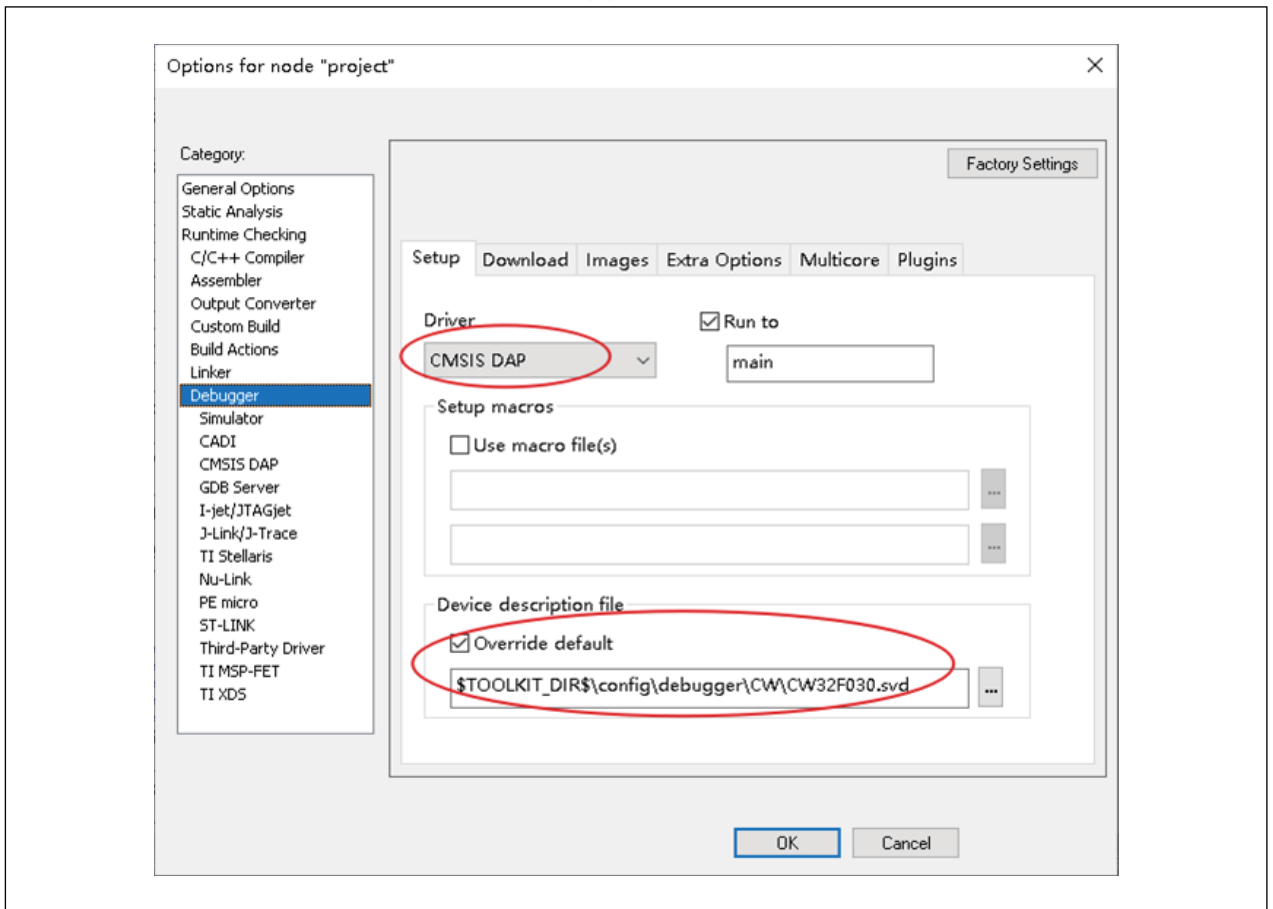
4. 在“Project”菜单中，选择“Options”以显示“Option .”对话框，并选择“Linker”项，修改 Linker 的配置文件为默认目录下的“CW”文件夹下的“CW32F030.icf”文件，如图 2-16：

图 2-16



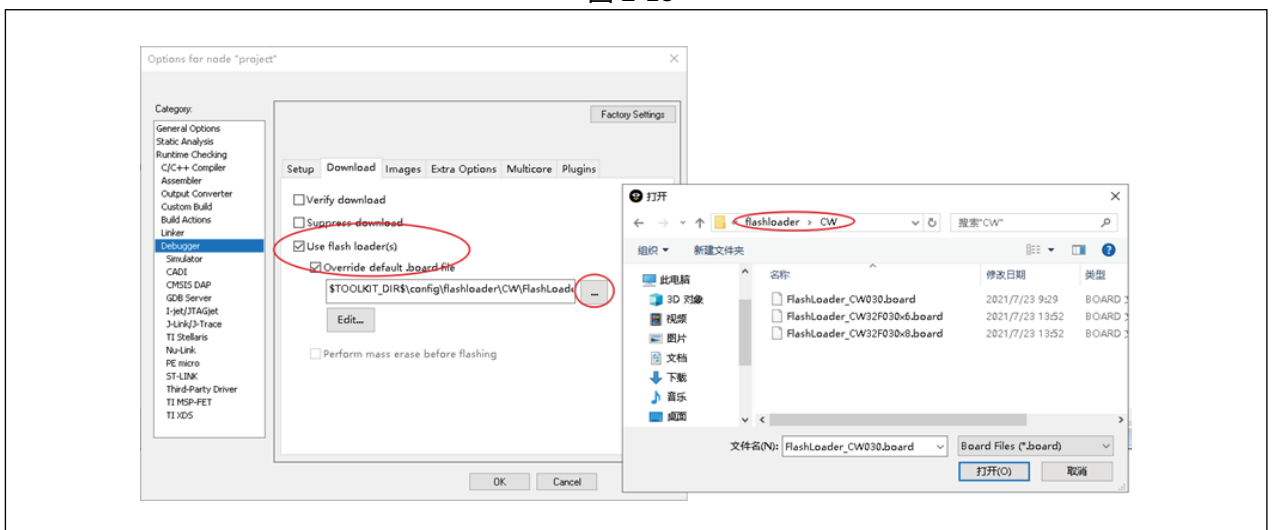
5. 在“Project”菜单中，选择“Options”以显示“Option.”对话框，并选择“Debugger”项，将“Driver”选为“CMSIS DAP”，并修改“Device description file”为默认目录下“CW”文件夹下的“CW32F030.svd”文件，如图 2-17:

图 2-17



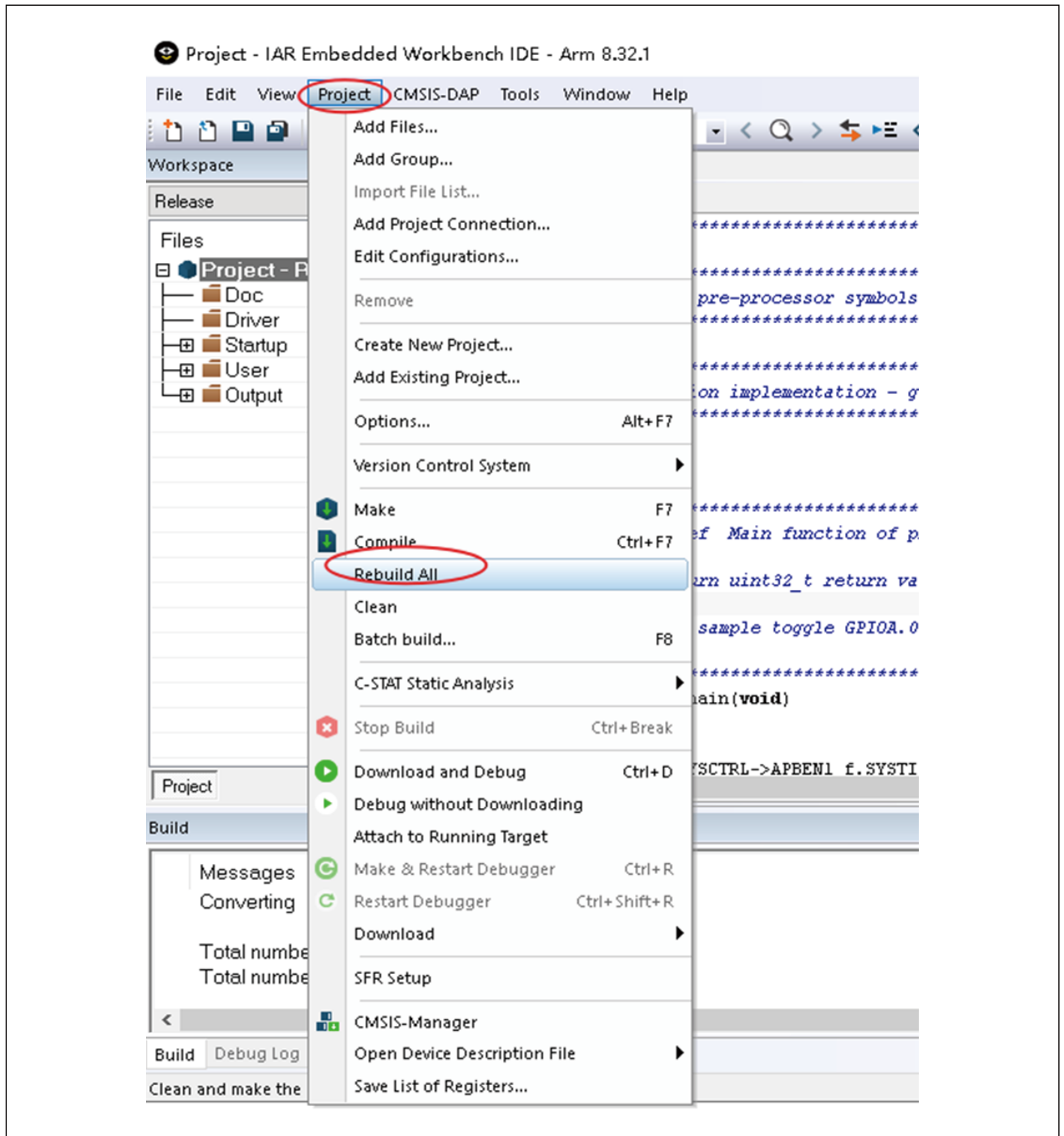
6. 在“Options”对话框中选择“Download”，复选“Use flash loader(s)”和“Override default .board file”，根据所选用芯片的 RAM 大小，将 .board 文件选择为默认目录下“CW”文件夹下的相应“FlashLoader\_CW32F030xx.board”文件，如图 2-18:

图 2-18



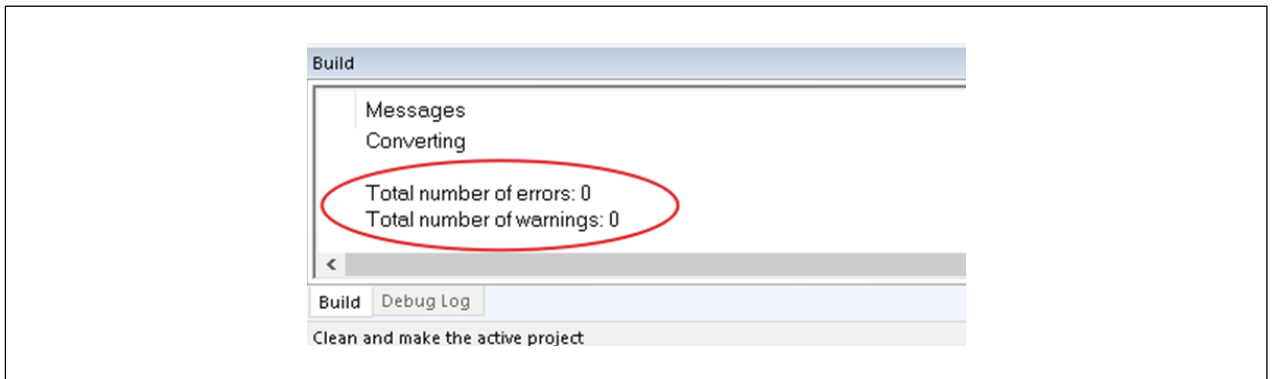
7. 在“Project”菜单中，选择“Rebuild All”以编译项目，如图 2-19：

图 2-19



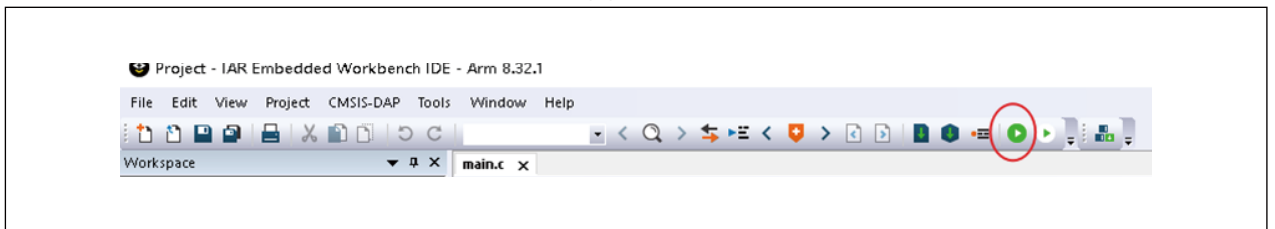
8. 如果编译成功，将显示如图 2-20 所示的内容：

图 2-20



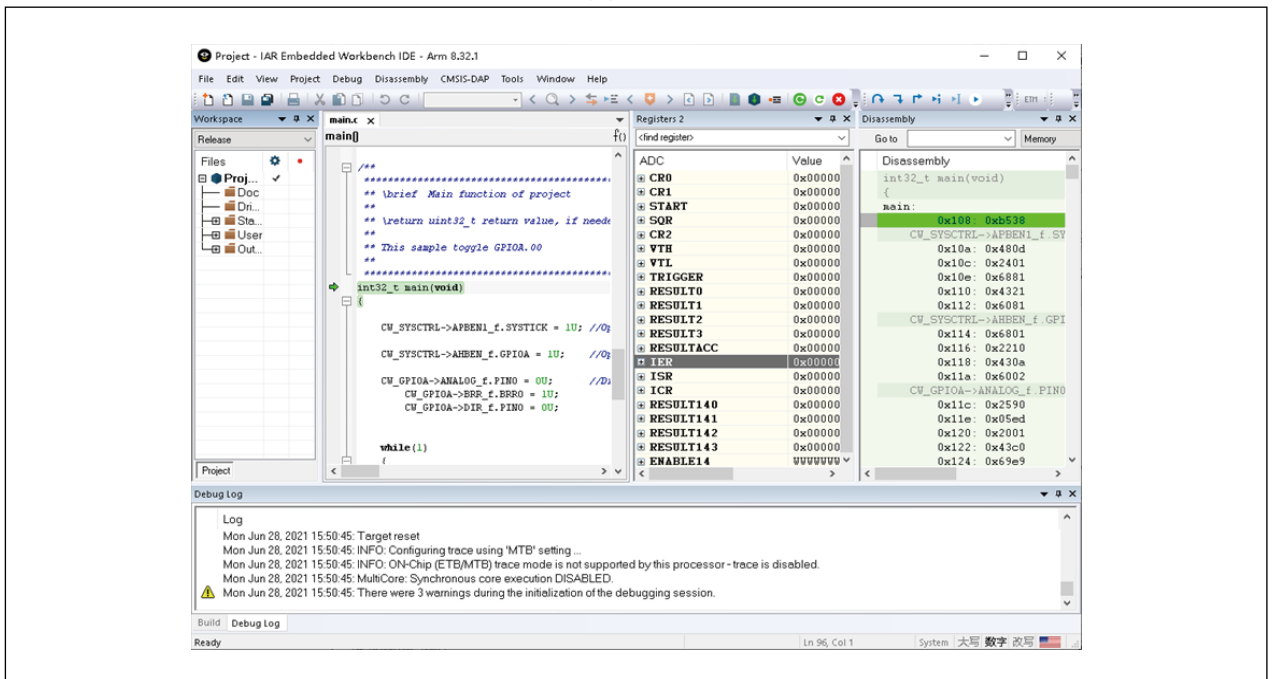
9. 在 IAR Embedded Workbench IDE 中，从项目菜单中选择下载和调试或点击工具栏中的下载和调试按钮，以对 Flash 存储器进行编程并开始调试，如图 2-21：

图 2-21



10. IAR Embedded Workbench 中的调试器可用于用 C 语言和汇编语言调试源代码，设置断点，以及监控各个变量和代码执行过程中发生的事件，如图 2-22：

图 2-22



## 2.2 MDK-ARM 工具链

(以 CW32F030 系列为例。)

1. 双击固件包 “IdeSupport\MDK” 目录下的 WHXY.CW32F030\_DFP.x.y.z.pack 文件 (如图 2-23), 在弹出的界面下点击 “Next” 进行安装 Pack, 如图 2-24:

图 2-23

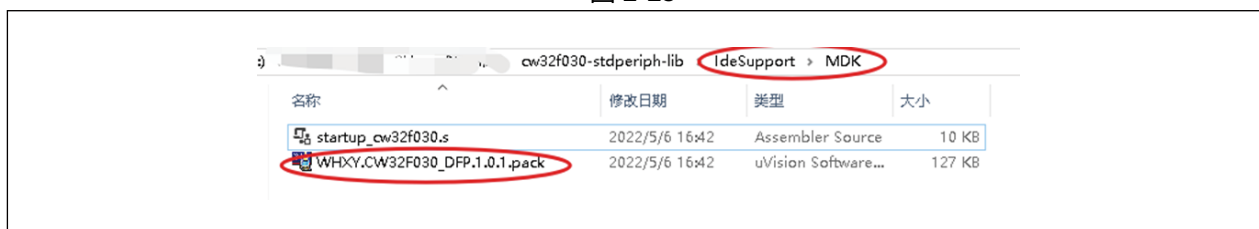
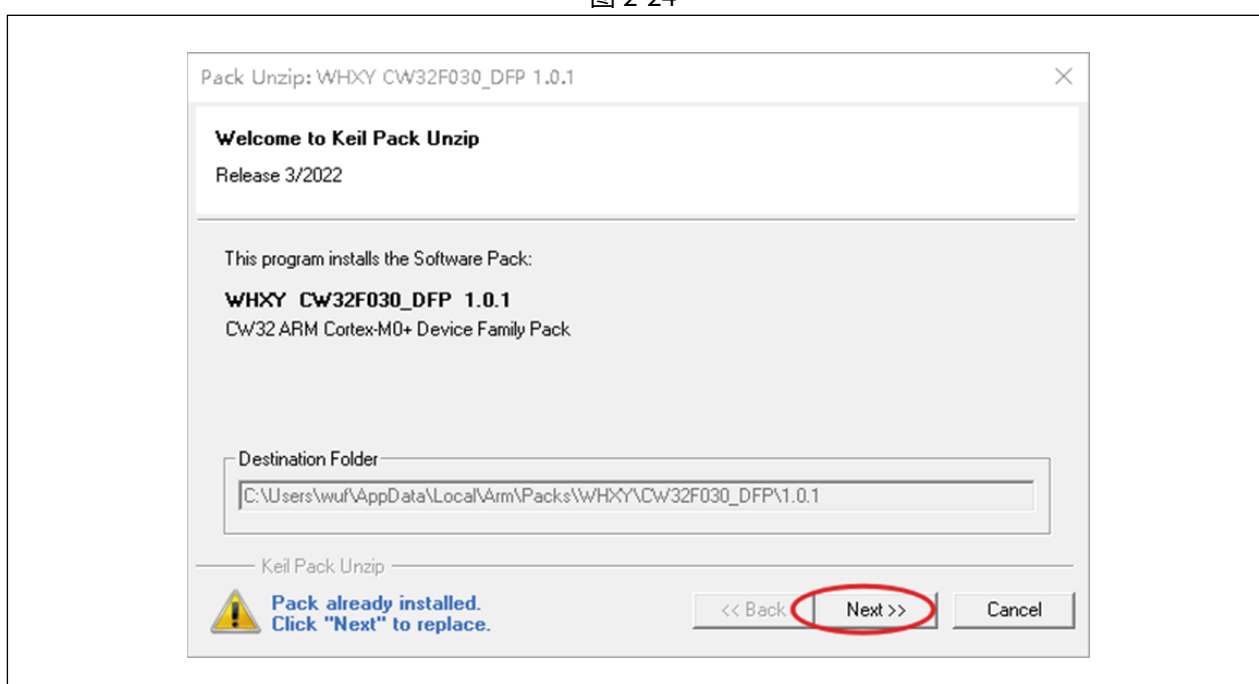
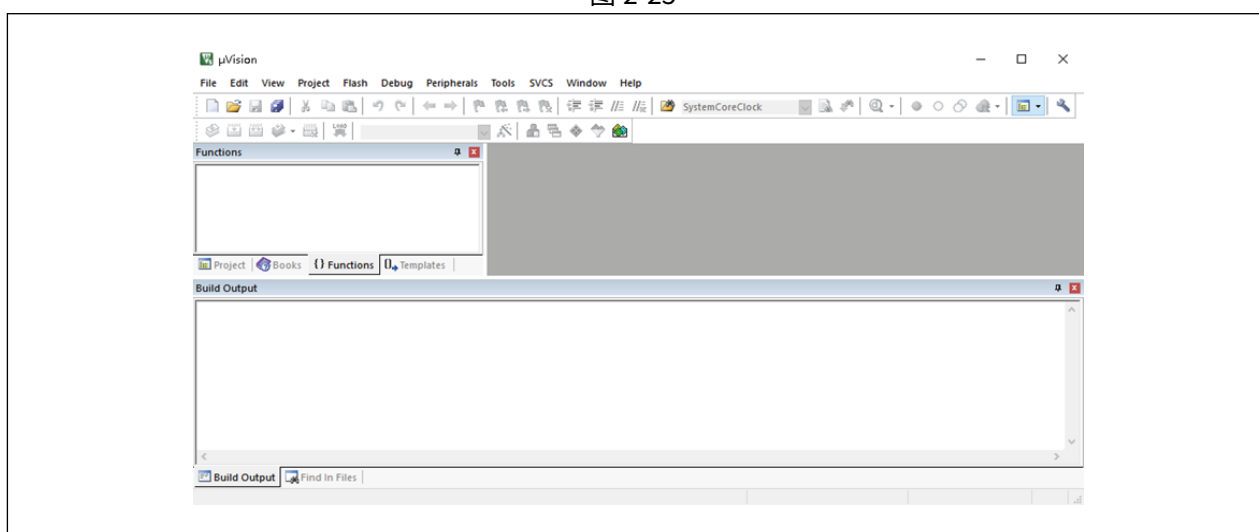


图 2-24



2. 打开 Keil MDK-ARM 微控制器集成开发环境, 如图 2-25:

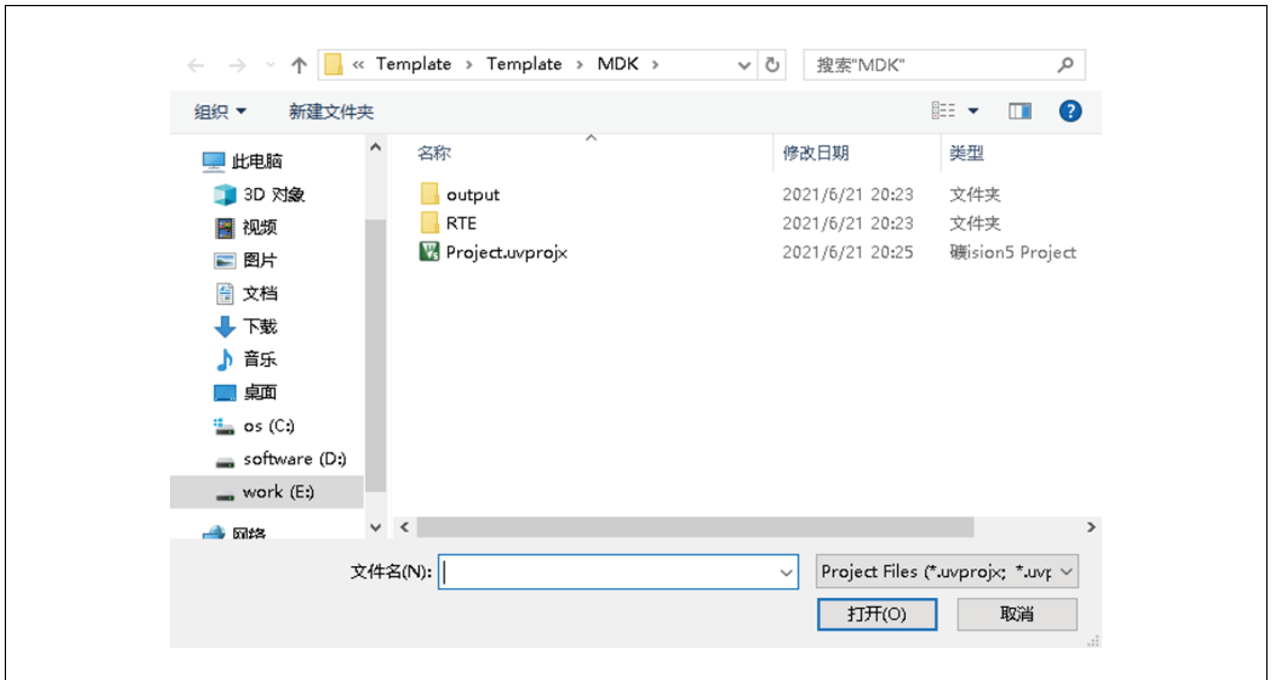
图 2-25





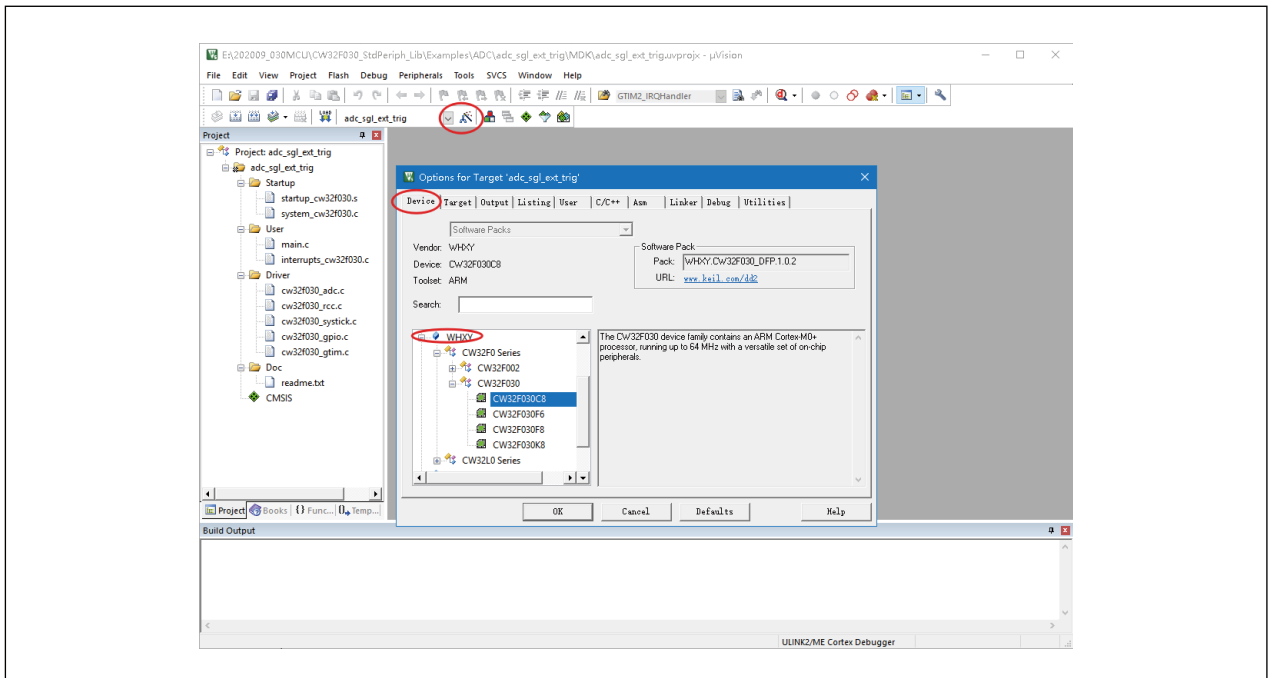
3. 在“Project”菜单中，选择“Open Project”以显示“Select Project File”对话框。浏览并选中一个示例项目文件，点击“打开”启动它，如图 2-26：

图 2-26



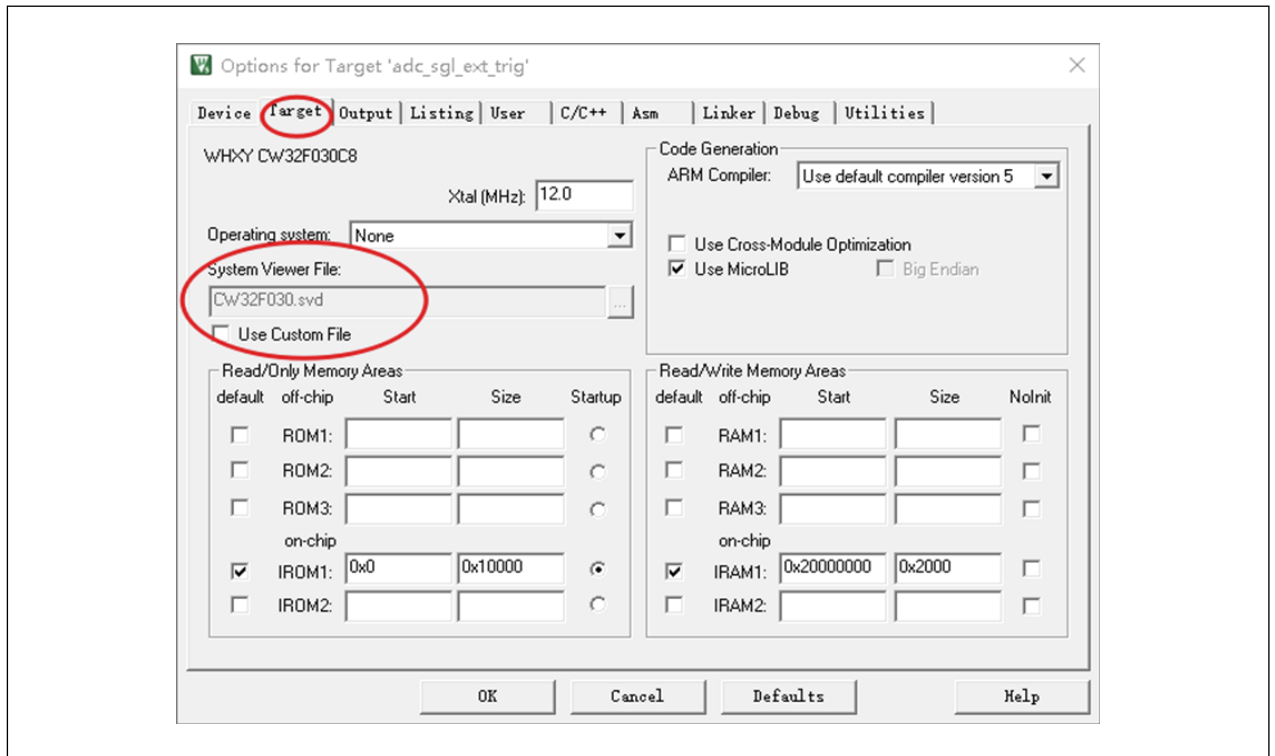
4. 选择工具栏的“Option”按钮，以显示“Option”对话框，选择“Device”标签，在器件列表中选择“WHXY”目录下的相应器件（以 CW32F030C8 为例），并点击“OK”确定，如图 2-27：

图 2-27



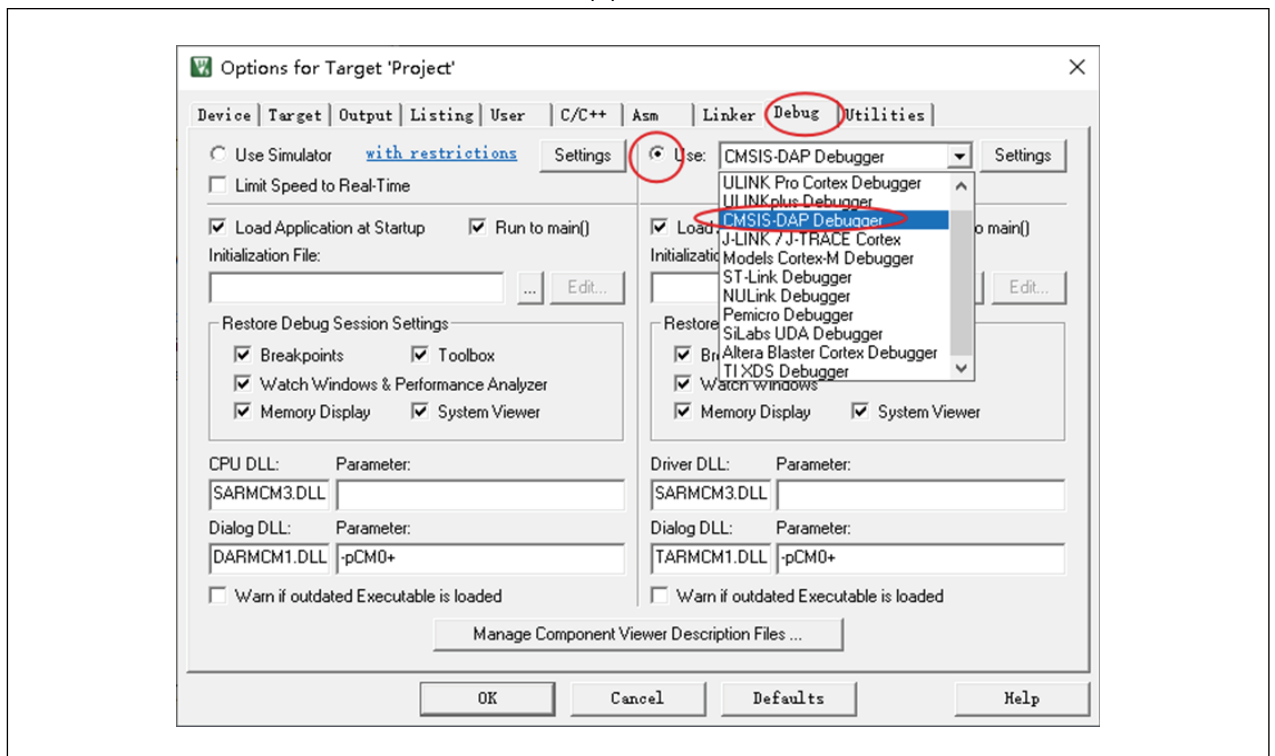
- 在“Option”对话框中，选择“Target”标签，确认“System Viewer File”为默认值，如图 2-28:

图 2-28



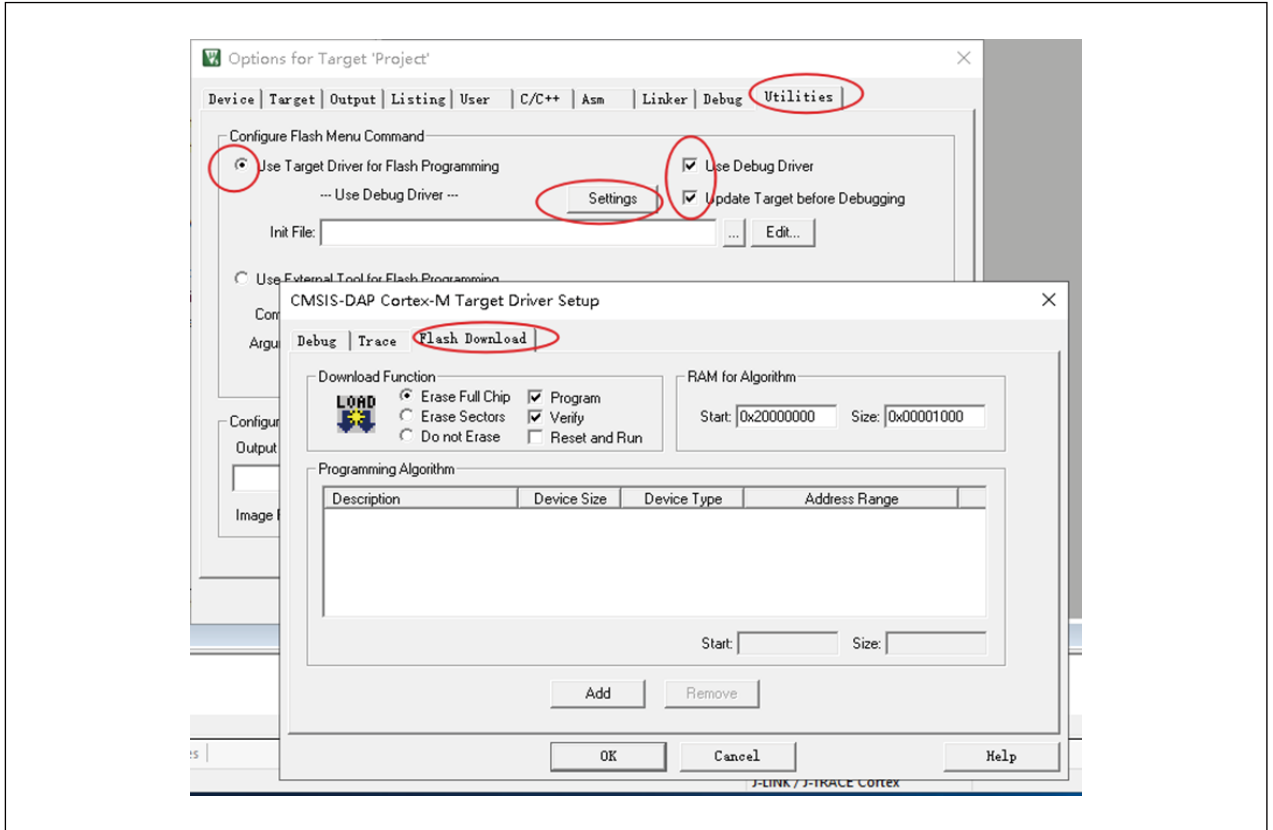
- 在“Option”对话框中，选择“Debug”标签，修改调试器为“CMSIS-DAP Debugger”，如图 2-29:

图 2-29



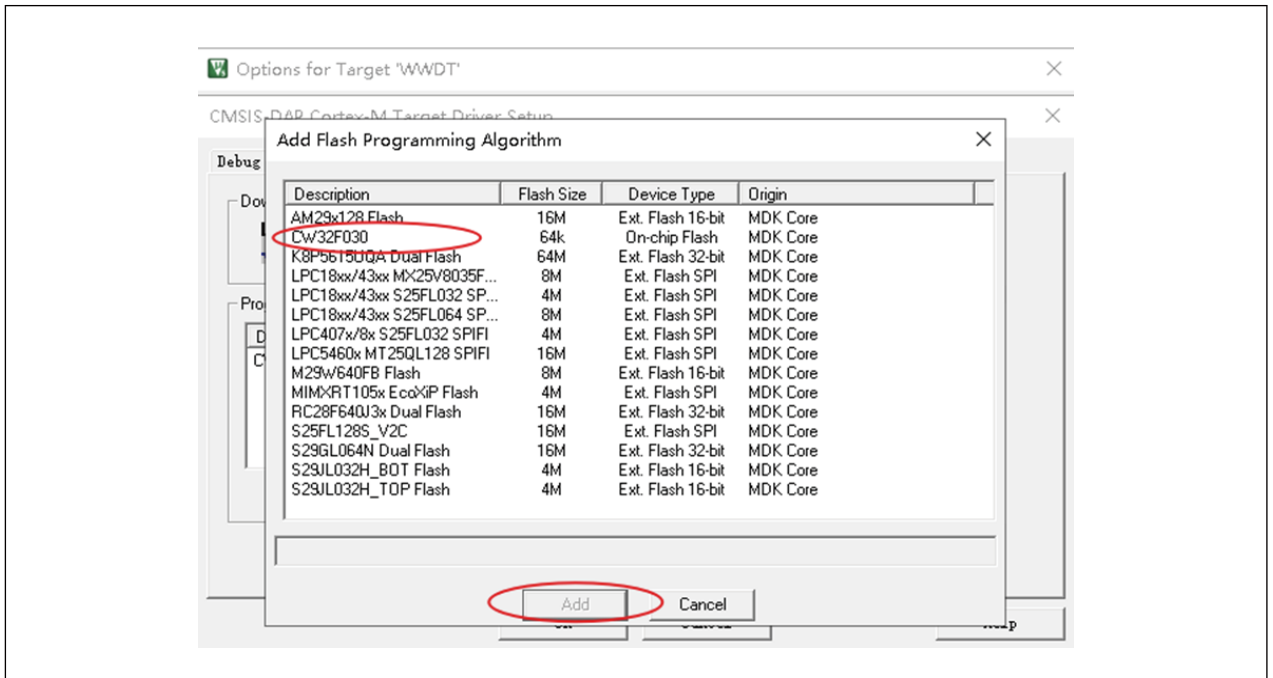
- 在“Option”对话框中，选择“Utilities”标签，点击“Settings”按钮，在弹出的对话框中选择“Flash Download”项，如图 2-30：

图 2-30



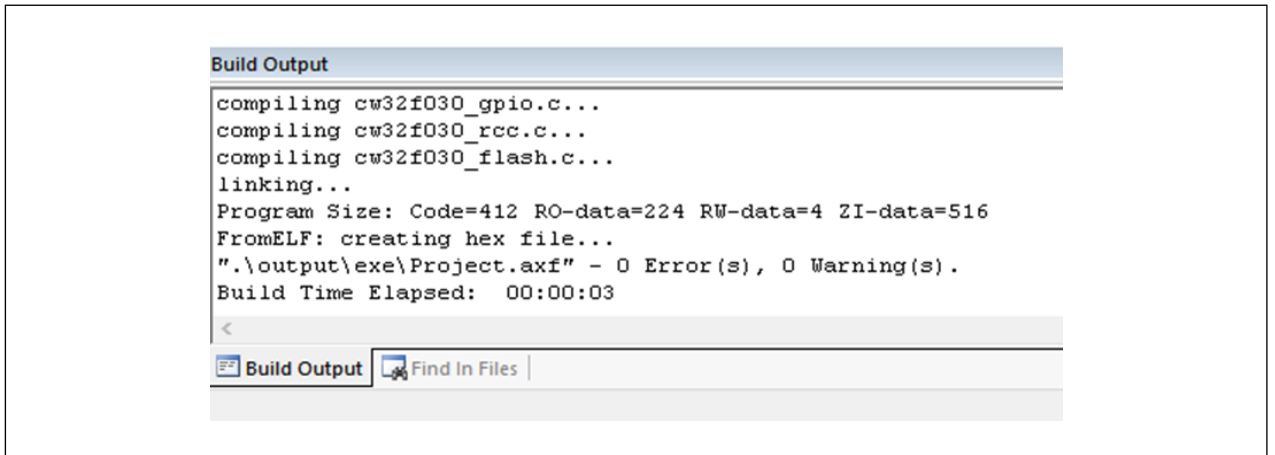
- 点击“Add”按钮，找到“CW32F030”的 Flash 烧写算法，点击“Add”按钮添加，如图 2-31：

图 2-31



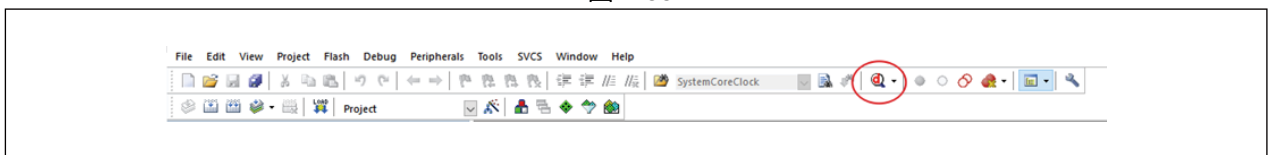
5. 在“Project”菜单中选择“Rebuild all target files”以编译项目。
6. 如果项目编译成功，将显示以下窗口，如图 2-32：

图 2-32



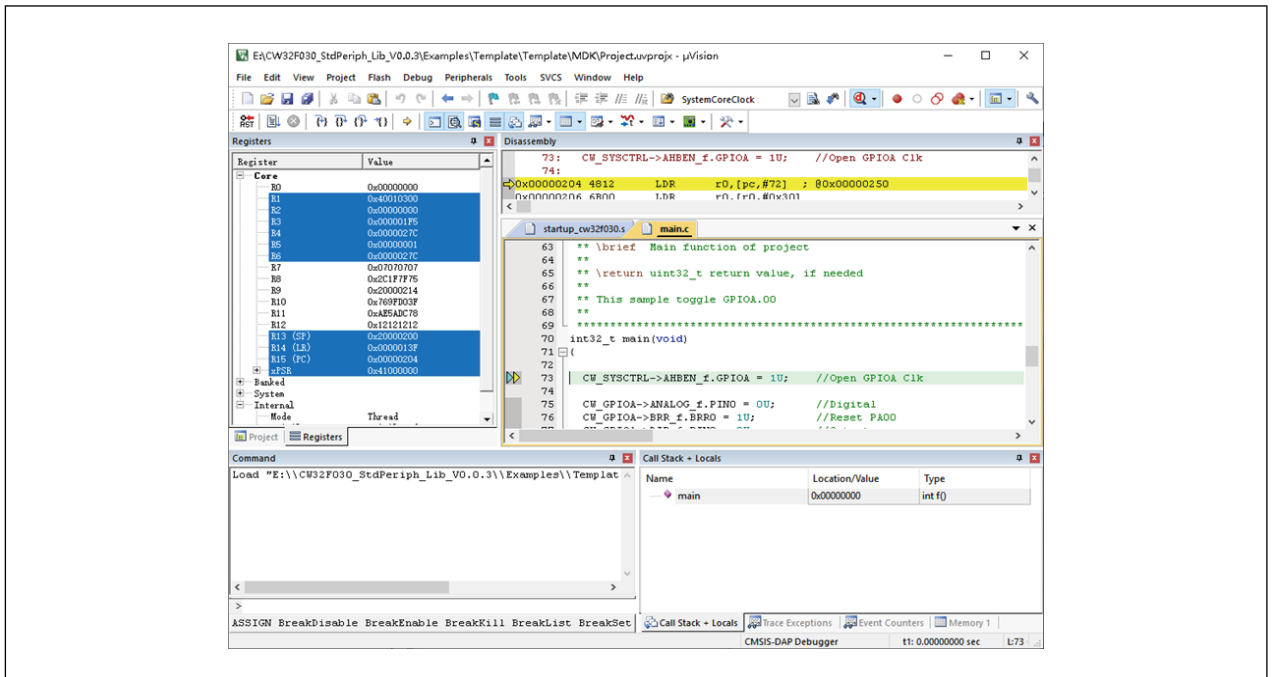
7. 在 MDK-ARM IDE 中，从调试菜单中选择开始 / 停止调试会话或点击工具栏中的开始 / 停止调试会话按钮，以对 Flash 存储器进行编程并开始调试，如图 2-33：

图 2-33



8. MDK-ARM 中的调试器可用于用 C 语言和汇编语言调试源代码，设置断点，以及监控各个变量和代码执行过程中发生的事件，如图 2-34：

图 2-34



### 3 版本信息

表 3-1 文档修订信息

日期	版本	变更信息
2021-10-14	Rev 1.0	初始发布
2022-06-22	Rev 1.1	增加 EWARM V9.30 或更高版本相关内容