

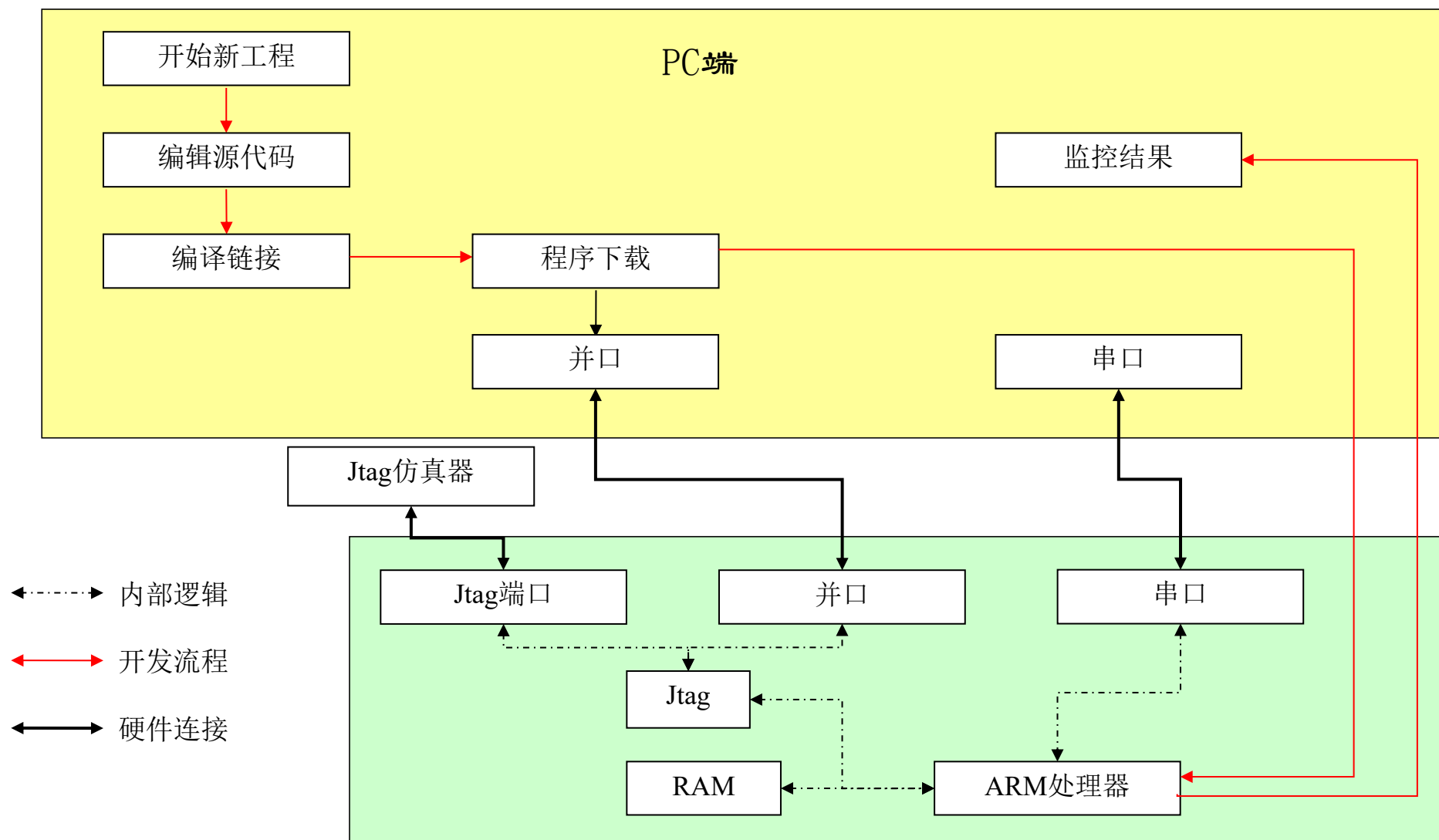


5. Setting Up the Development Environment of ARM — (1)

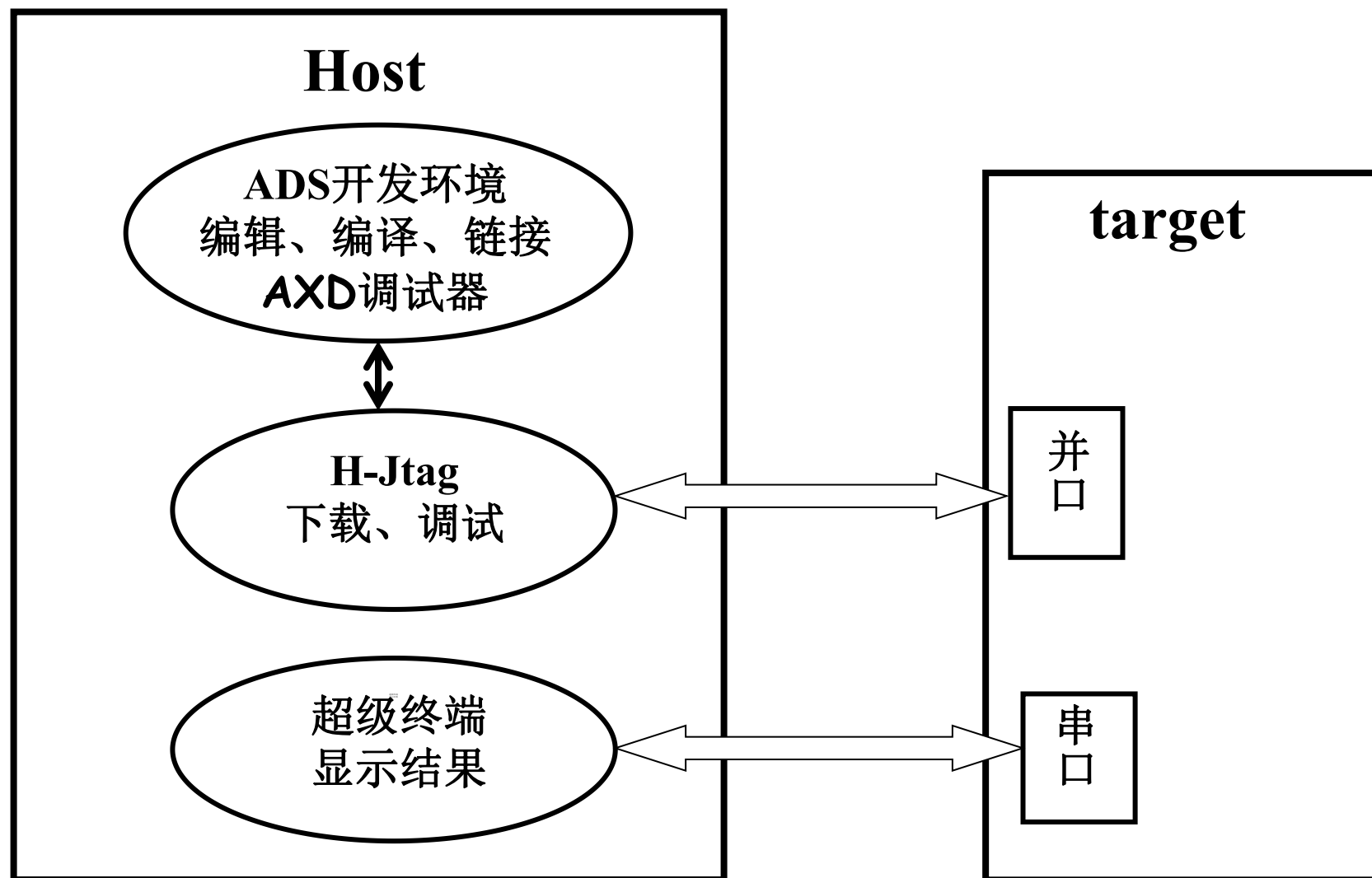
content

- **Using ADS**
- Using Linux
 - Host Configuration
 - Building Cross-compiler tool chain
 - download vivi kernel root
 - Application Development
 - Download using the NFS

Using ADS (ARM Developer Suite)



Using ADS



Using ADS

➤ ADS集成开发环境组成介绍

- ❑ ARM ADS全称为ARM Developer Suite, 是ARM公司推出的ARM集成开发工具。
- ❑ ADS由命令行开发工具, ARM运行时库, GUI集成开发环境(Code Warrior), 调试工具(AXD), 实用程序和支持软件组成。

- SoundMAX
- ThinkVantage
- Adobe Reader 7.0
- Internet Explorer
- Outlook Express
- Windows Media Connect
- Windows Media Player
- Windows Messenger
- Windows Movie Maker
- Microsoft Office
- WinRAR
- Symantec Client Security
- Symantec 系统中心控制台
- UltraISO
- Keil uVision2
- 附件
- 启动
- 远程协助
- 金山词霸 2005
- 天网防火墙个人版
- 新东方英语软件系列

- 迅雷
- Google Earth
- 腾讯软件
- QQ游戏
- 闪联通用自动更新
- ARM Developer Suite v1.2
- WebZIP

- AXD Debugger
- CodeWarrior for ARM Developer Suite
- License Installation Wizard
- Online Books
- ReadMe for ARM Developer Suite v1.2
- Setup for ARM Developer Suite v1.2

程序 (P)

Using ADS

➤ ADS集成开发环境组成介绍

□ 命令行开发工具完成将源代码编译，链接成可执行代码的功能

- armcc, armcpp (ARM C, C++编译器)

- tcc (Thumb C 编译器)

- armasm (ARM和Thumb的汇编器)

- armlink (ARM连接器)

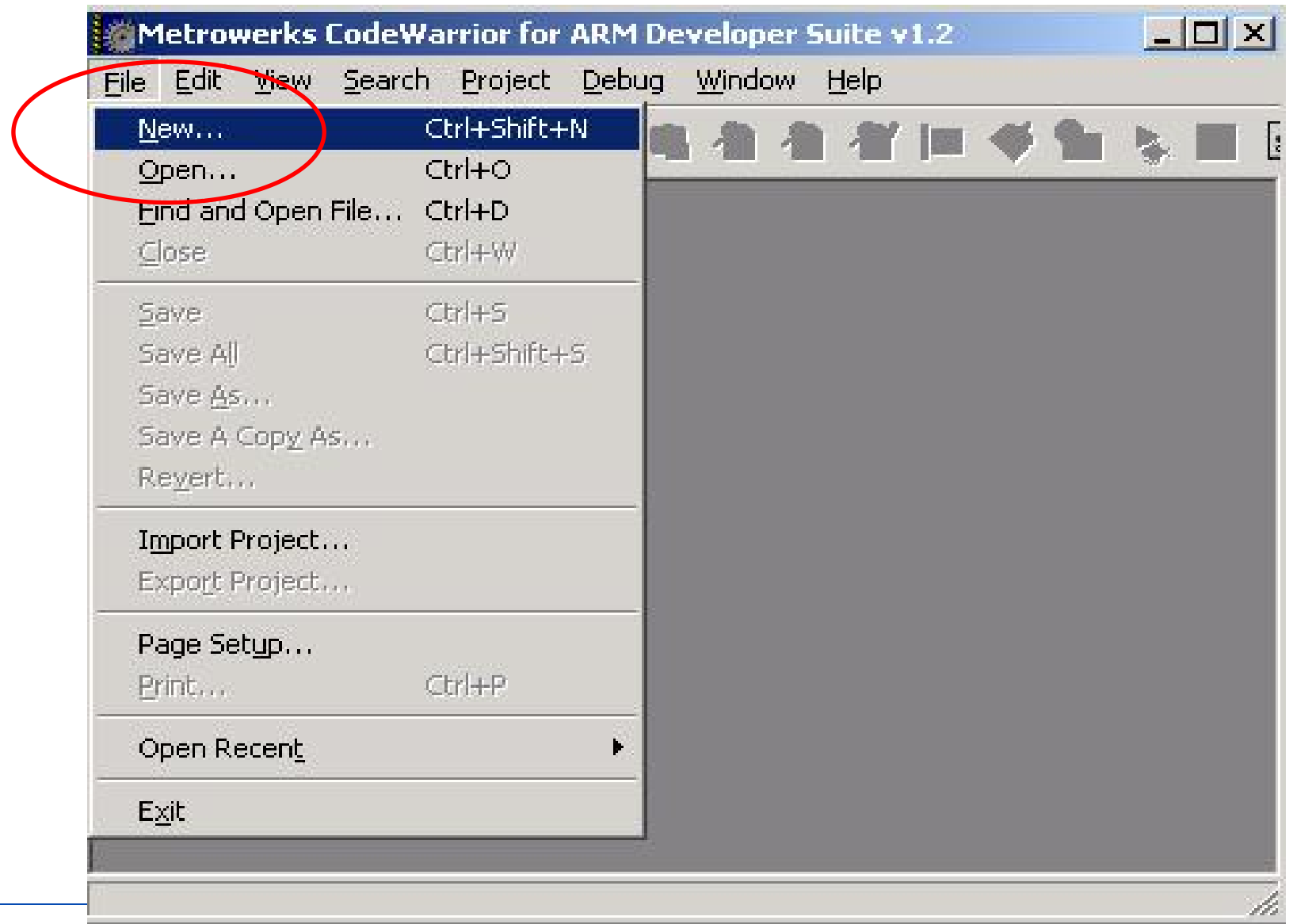
□ ANSI C库函数, C++库函数

Using ADS

- GUI开发环境——CodeWarrior集成开发环境
 - ❑ CodeWarrior(IDE)为管理和开发项目提供了简单多样化的图形用户界面。
 - ❑ CodeWarrior(IDE)为ARM处理器用C, C++或ARM汇编语言进行程序代码开发。
 - ❑ 可以在CodeWarrior(IDE)为ARM配置在前面介绍的各种命令工具, 实现对工程代码的编译, 汇编和链接。

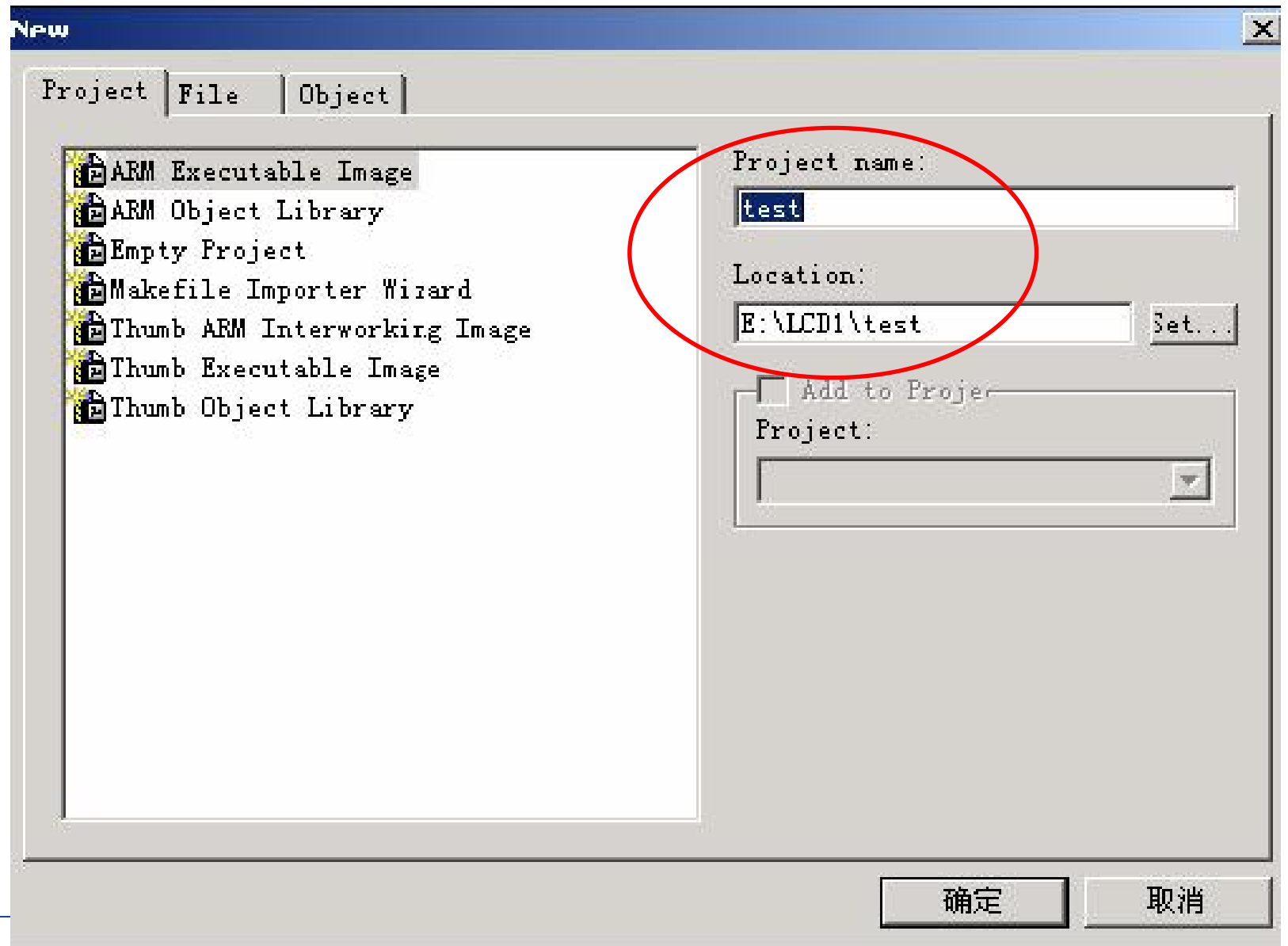
Using ADS

➤ CodeWarrior(IDE)



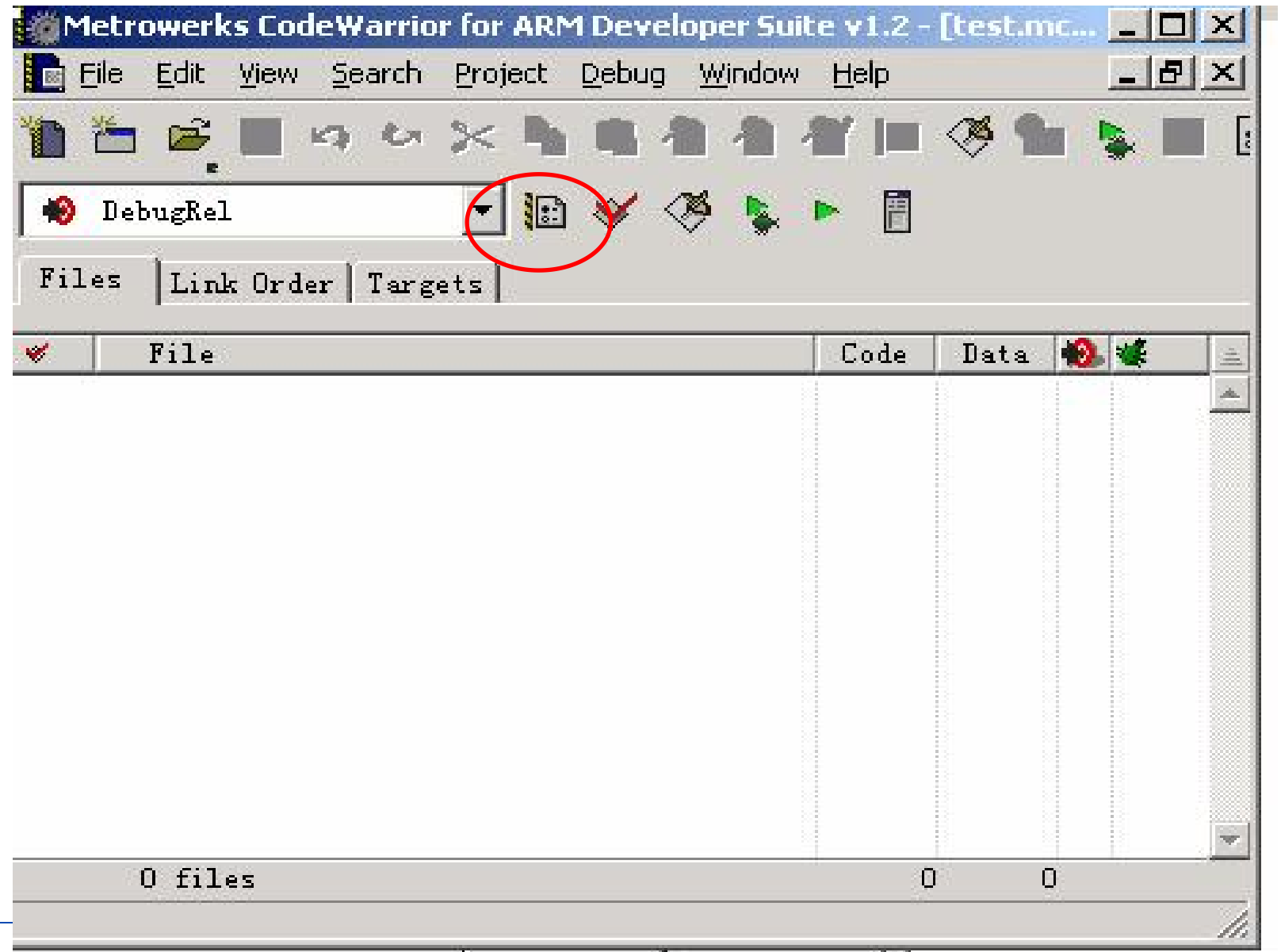
Using ADS

➤ CodeWarrior(IDE)

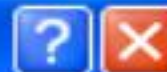


Using ADS

➤ CodeWarrior(IDE)



DebugRel Settings



Target Settings Panels

- [-] Target
 - Target Settings**
 - Access Paths
 - Build Extras
 - Runtime Settings
 - File Mappings
 - Source Trees
 - ARM Target
- [-] Language Settings
 - ARM Assembler
 - ARM C Compiler
 - ARM C++ Compiler
 - Thumb C Compiler
 - Thumb C++ Com...
- [-] Linker
 - ARM Linker
 - ARM fromELF
- [-] Editor

Target Settings

Target Name:

Linker:

Pre-linker:

Post-linker:

Output Directory:

☐ Save project entries using relative paths

Using ADS

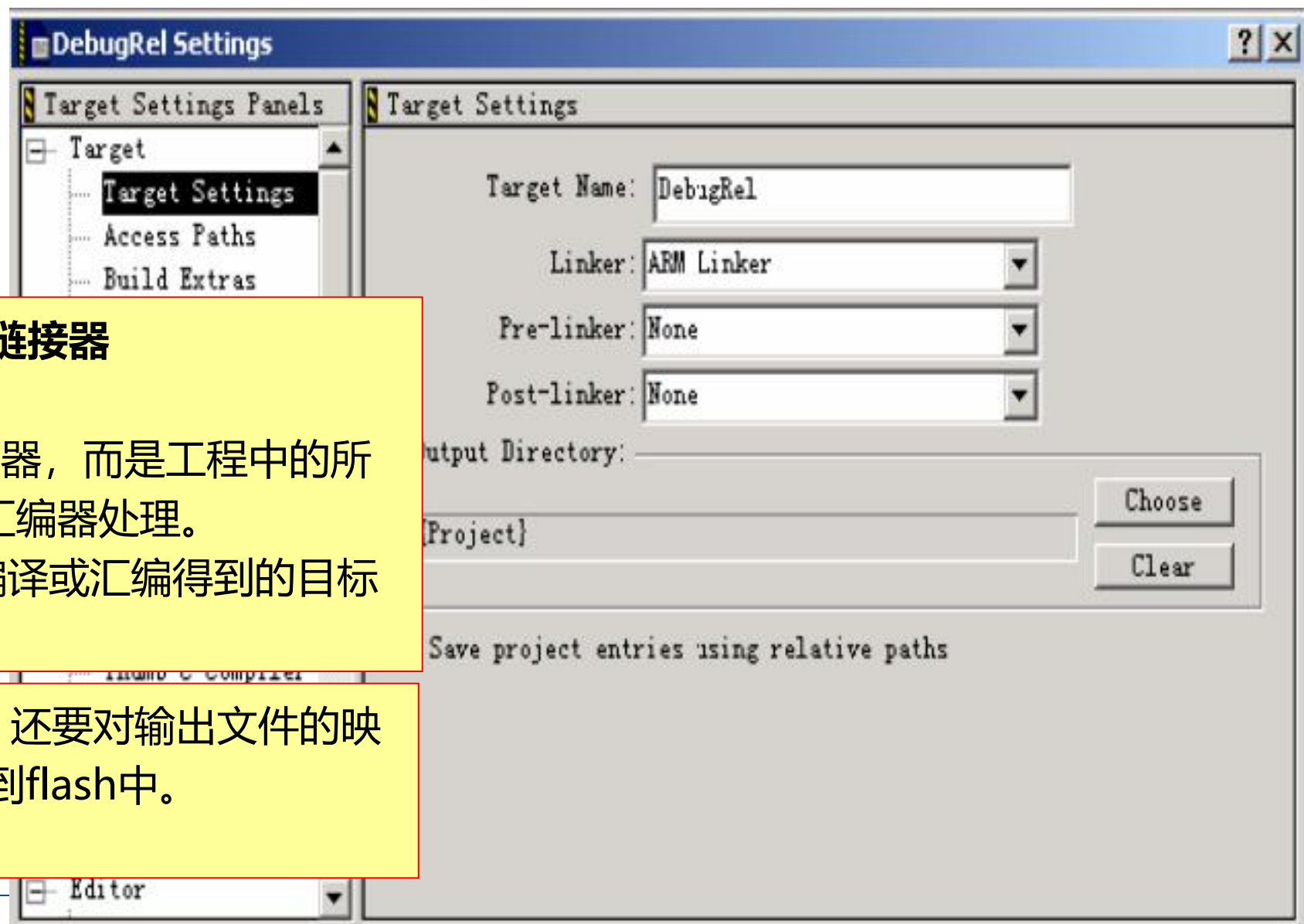
➤ CodeWarrior(IDE)

- **Linker:** 供用户选择要使用的链接器

- ARM Linker: 默认选项。
- None: 不是不用任何链接器, 而是工程中的所有文件都不会被编译器或汇编器处理。
- ARM Librarian: 表示将编译或汇编得到的目标文件转换为ARM库文件。

- **Post-Linker:** 在链接完成后, 还要对输出文件的映像文件进行转换, 以便能烧写到flash中。

- FromELF: 默认选项



- Target Settings Panels
- [-] Target
 - Target Settings
 - Access Paths
 - Build Extras
 - Runtime Settings
 - File Mappings
 - Source Trees
 - ARM Target
 - [-] Language Settings
 - ARM Assembler
 - ARM C Compiler
 - ARM C++ Compiler
 - Thumb C Compiler
 - Thumb C++ Com...
 - [-] Linker
 - ARM Linker
 - ARM fromELF
 - [-] Editor

ARM Assembler

Target | ATPCS | Options | Predefines | Listing Control | Extras

Architecture or Processor

ARM7TDMI

Floating Point

Pure-endian softfp

Byte Order

☒ Little Endi

☐ Big Endian

Initial State

☒ ARM

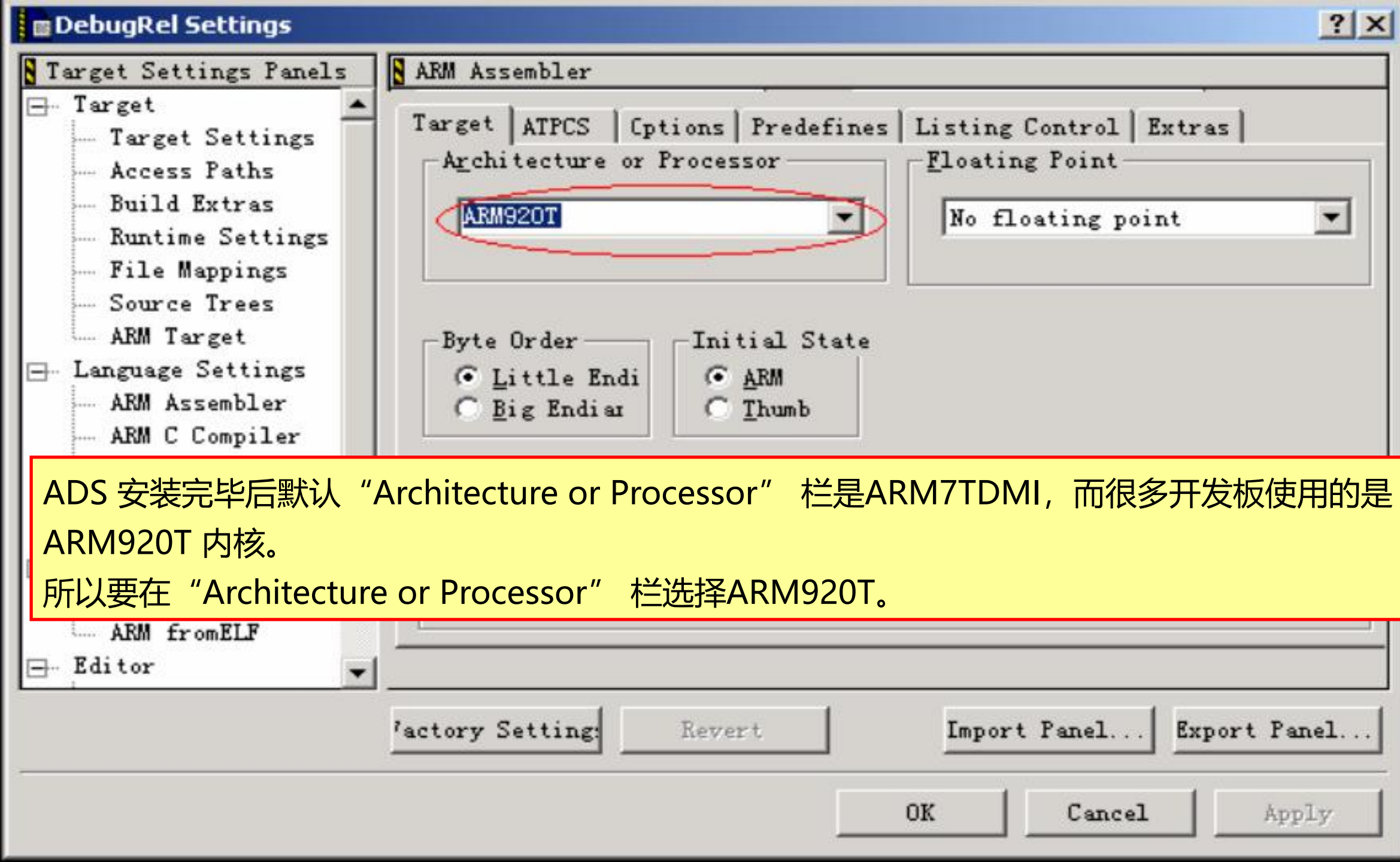
☐ Thumb

Equivalent Command Line

-keep -g

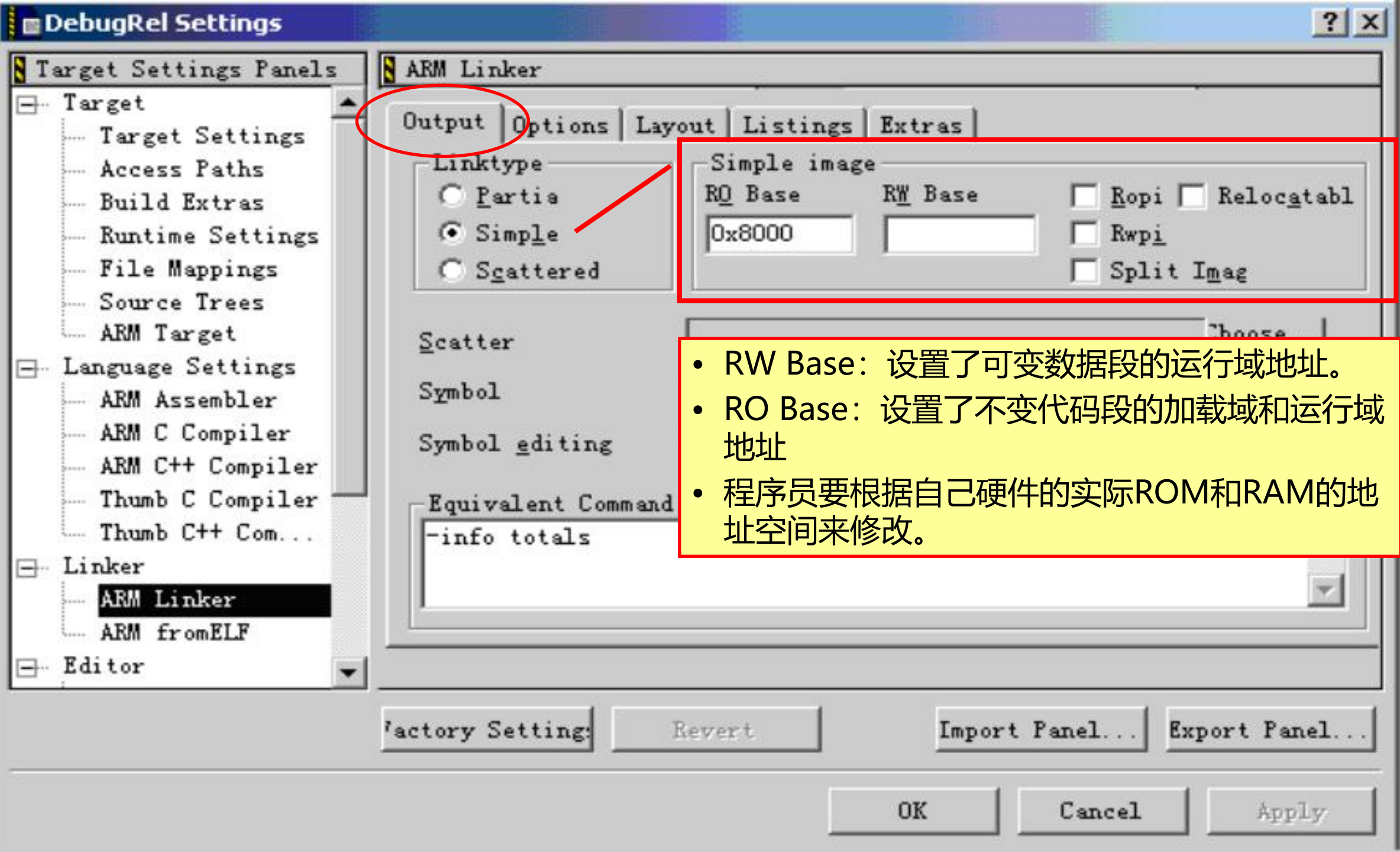
Factory Settings Revert Import Panel... Export Panel...

OK Cancel Apply



ADS 安装完毕后默认 “Architecture or Processor” 栏是ARM7TDMI，而很多开发板使用的是ARM920T 内核。

所以要在 “Architecture or Processor” 栏选择ARM920T。



- RW Base: 设置了可变数据段的运行域地址。
- RO Base: 设置了不变代码段的加载域和运行域地址
- 程序员要根据自己的硬件的实际ROM和RAM的地址空间来修改。

DebugRel Settings



Target Settings Panels

- [-] Target
 - Target Settings
 - Access Paths
 - Build Extras
 - Runtime Settings
 - File Mappings
 - Source Trees
 - ARM Target
- [-] Language Settings
 - ARM Assembler
 - ARM C Compiler
 - ARM C++ Compiler
 - Thumb C Compiler
 - Thumb C++ Com...
- [-] Linker
 - ARM Linker**
 - ARM fromELF
- [-] Editor

ARM Linker

Output | Options | Layout | Listings | Extras

Place at beginning of image

Object/Symbol

2440init.o

Section

Init

Place at end of image

Object/Symbol

Section

Equivalent Command Line

-info totals -ro-base 0x30000000 -first 2440init.o(Init)

Factory Settings

Revert

Import Panel...

Export Panel...

Using ADS

➤ CodeWarrior集成开发环境(IDE)

- ❑ ADS除了有集成了大多数ARM工具链的CodeWarrior IDE以外，还包含和调试相关的功能，如ARM调试器(AXD)，虽然可能没有集成到CodeWarrior IDE中。
- ❑ 用户通过调试器软件，如AXD，再使用“debug agent”（调试代理），就可以对包含有调试信息的，正在运行的可执行代码进行比如变量的查看，断点的控制等调试操作。

Using ADS

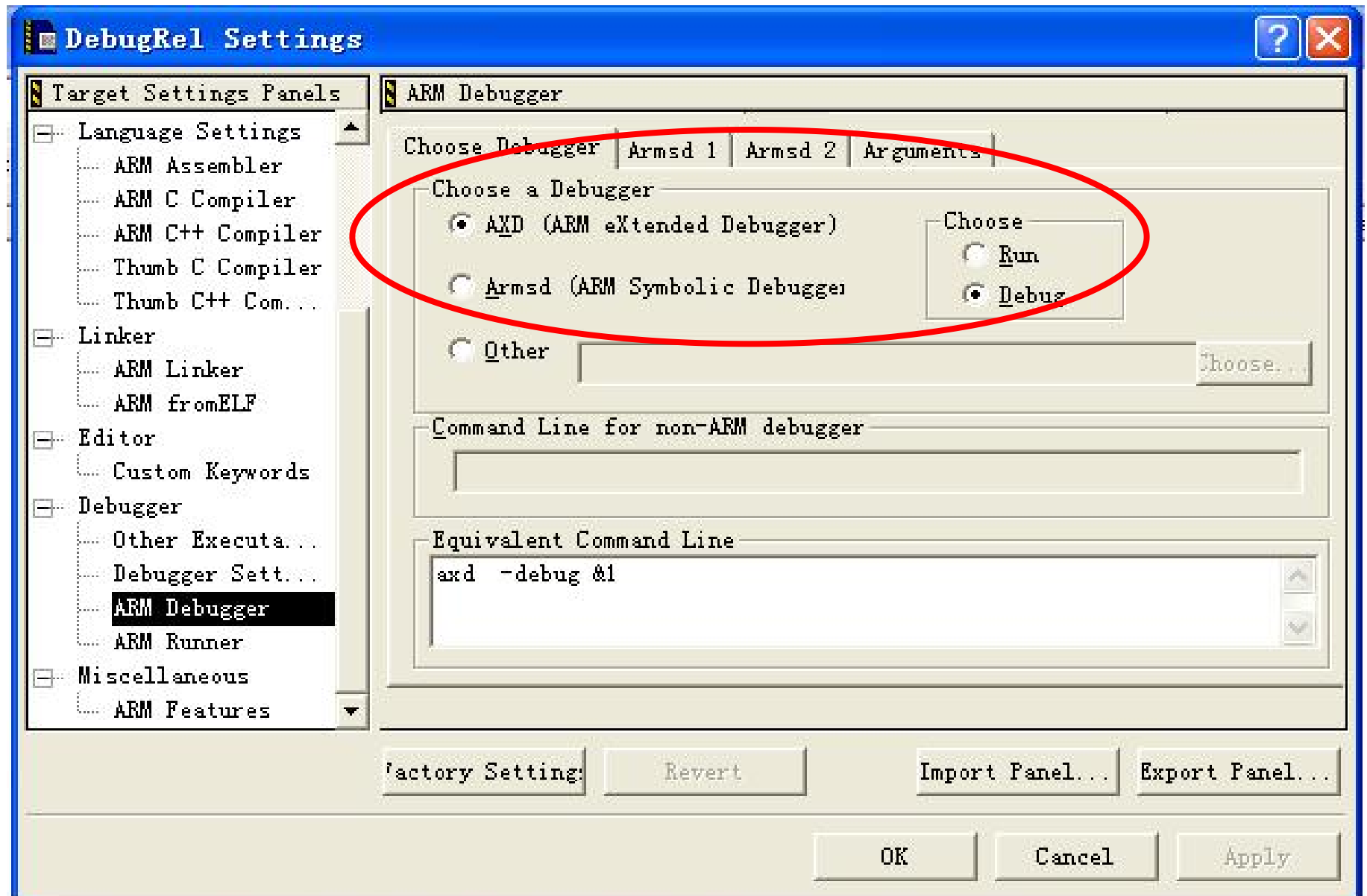
➤ ADS Debugger

▣ ADS中包含有3个调试器:

- AXD(ARM eXtended Debugger): ARM扩展调试器;
- armsd(ARM Symbolic Debugger): ARM符号调试器
- 与老版本兼容的Windows或Unix下的ARM调试工具, ADW/ADU(Application Debugger Windows/Unix)。

Using ADS

➤ ADS Debugger



Using ADS

➤ ADS调试器 (Debug target)

- ❑ 在软件开发的最初阶段，可能还没有具体的硬件设备。如果要测试所开发的软件是否达到了预期的效果，这可以由软件仿真来完成。即使调试器和要测试的软件运行在同一台PC上，也可以把目标当作一个独立的硬件来看待。
- ❑ 当然，也可以搭建一个PCB板，这个板上可以包含一个或多个处理器，在这个板上可以运行和调试应用软件。

Using ADS

➤ ADS调试器

调试器能够发送以下指令：

1. 装载映像文件到目标内存；（和固化烧写不同）
2. 启动或停止程序的执行；
3. 显示内存，寄存器或变量的值；
4. 允许用户改变存储的变量值。

Using ADS

➤ AXD调试器配置

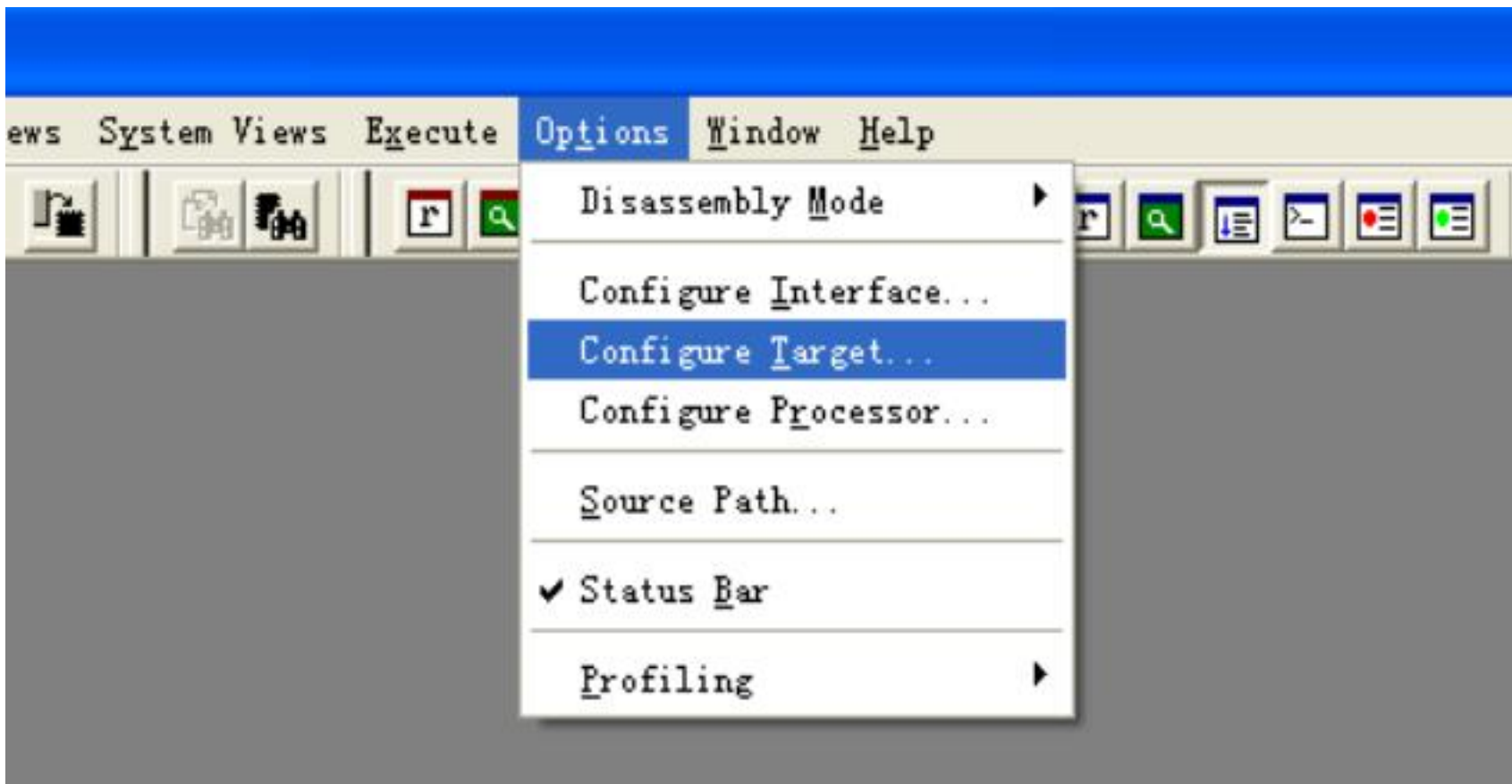
- ❑ 如果打开过程中出现错误不必理会，连续确定或取消就可以



Using ADS

➤ AXD调试器配置

- ❑ 选择AXD菜单中的->OPTIONS->CONFIGURE TARGET->ADD->

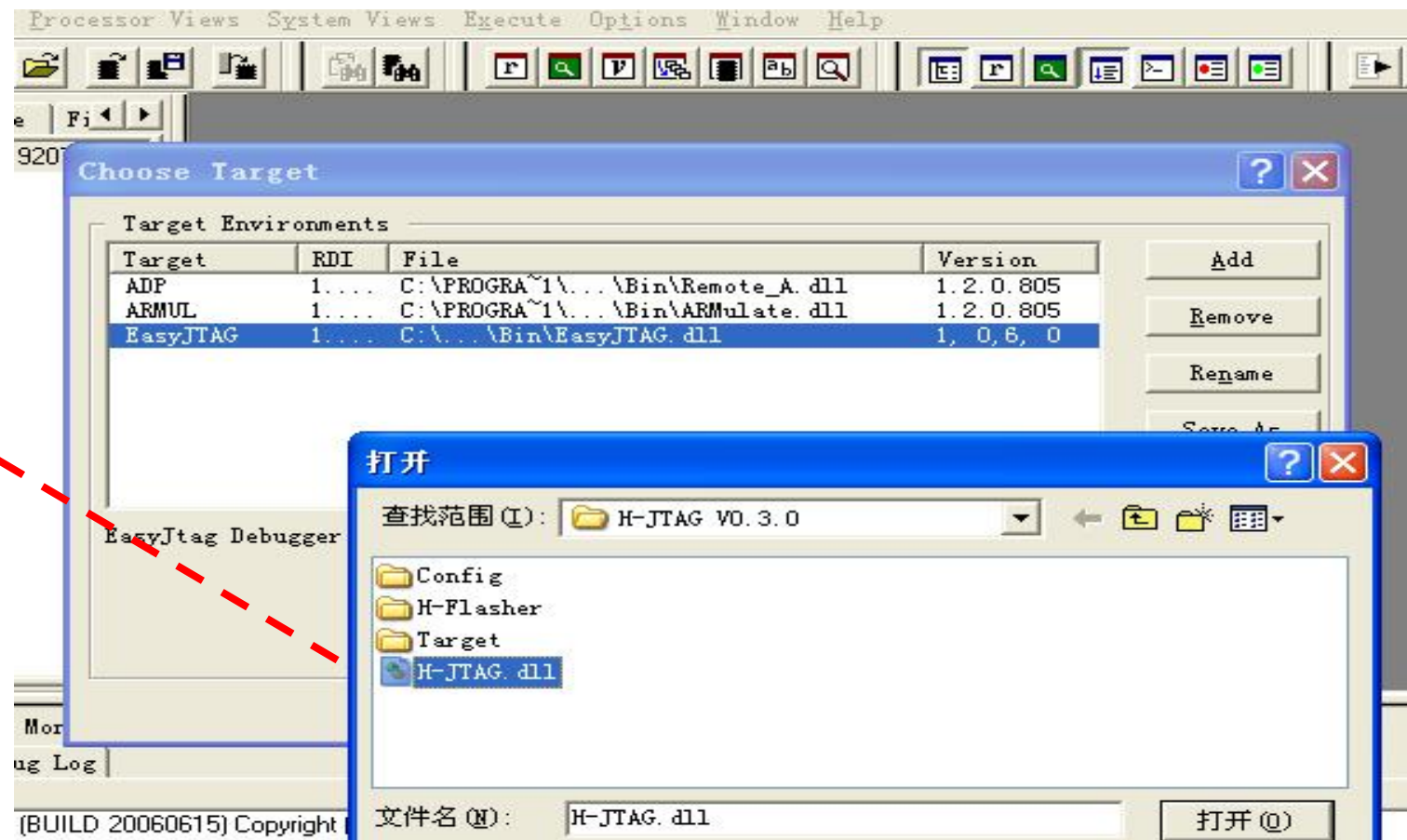


Using ADS

➤ AXD调试器配置

- ❑ 选择C:\Program Files\H-JTAG V0.3.0目录下的H-JTAG.DLL.->选择OK就完成了配置。

添加调试代理 软件

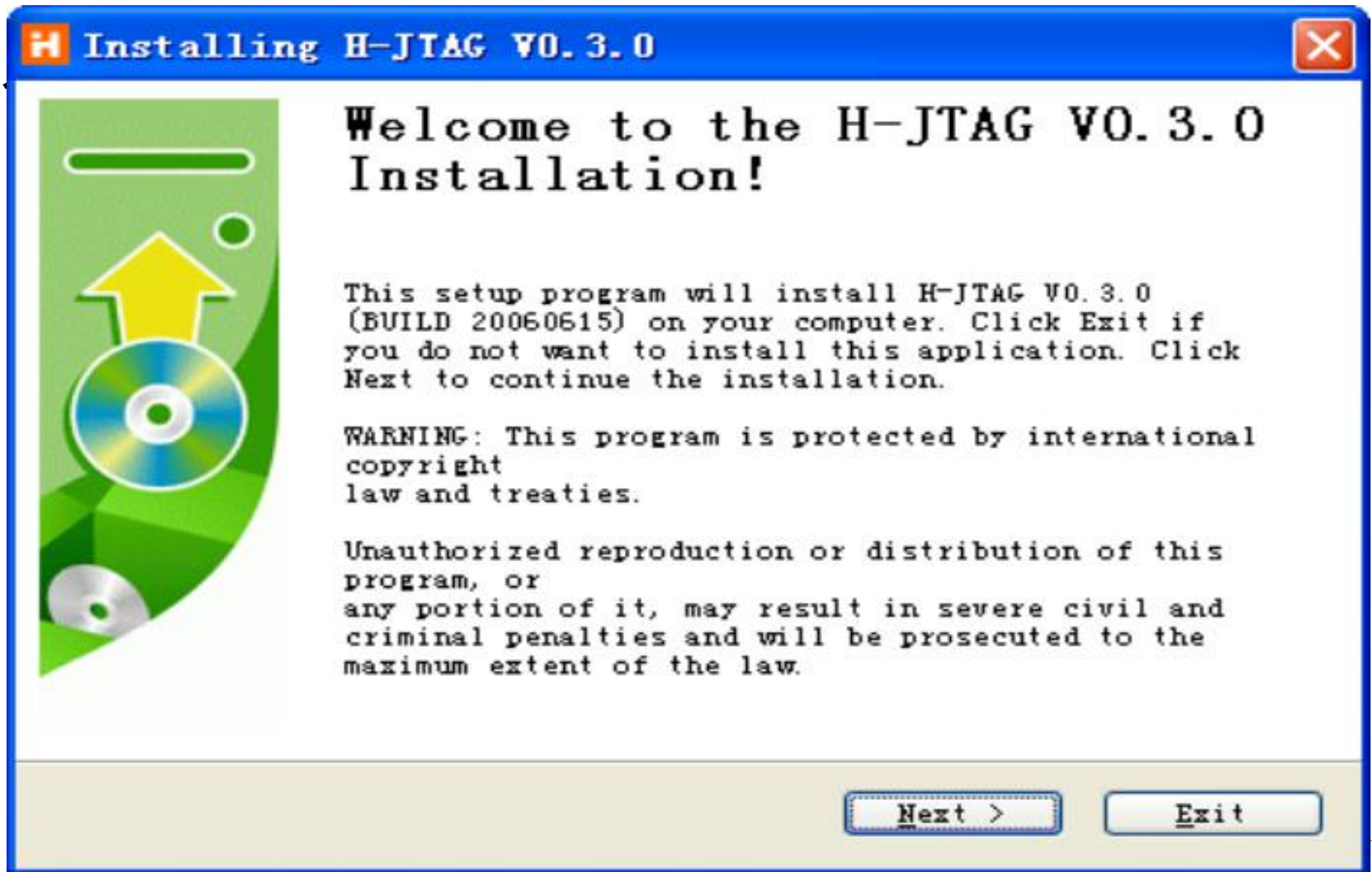


Using ADS

- H-JTAG介绍：是一个免费的ARM调试代理。
 - ① 支持ARM7/ARM9，支持自动检测和手动指定内核.
 - ② 使用RDI接口，支持SDT，ADS，REALVIEW和IAR.
 - ③ 支持个硬件断点或者数量不限的软件断点
 - ④ 支持ARM/THUMB模式
 - ⑤ 支持LITTLEENDIAN & BIGENDIAN
 - ⑥ 支持自定义接口

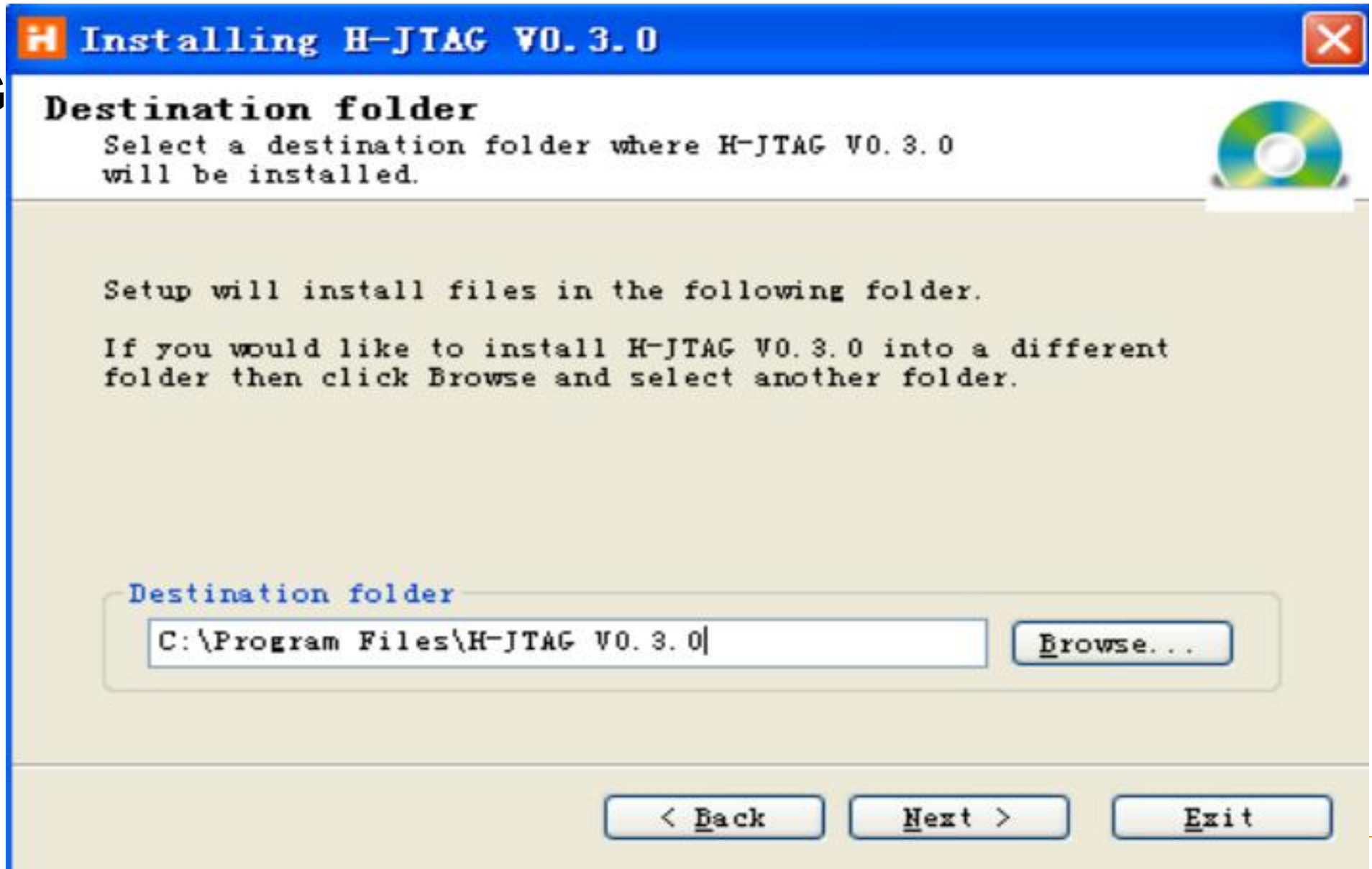
Using ADS

➤ H-JTAG



Using ADS

➤ H-JTAG



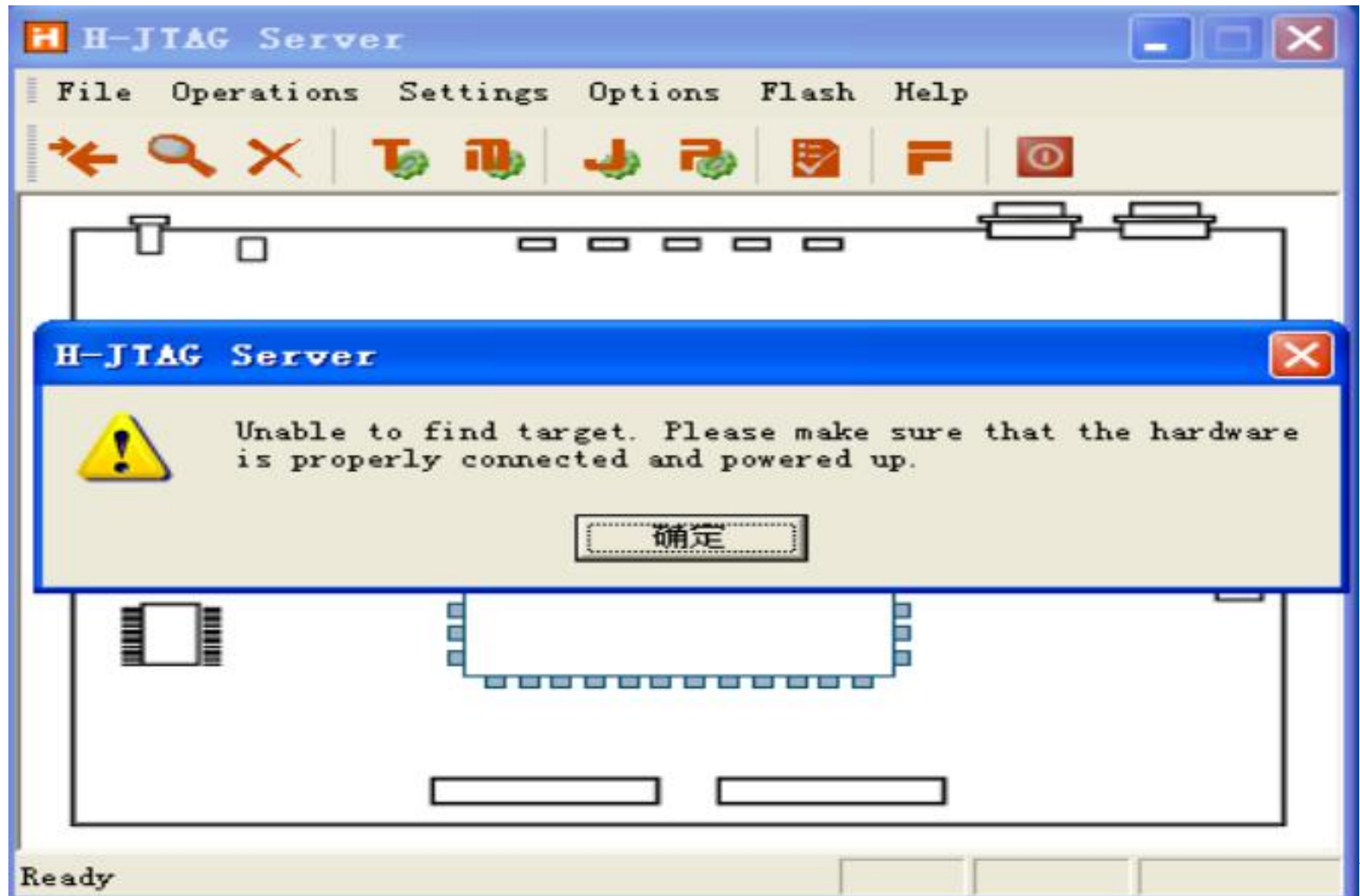
Using ADS

➤ H-JTAG



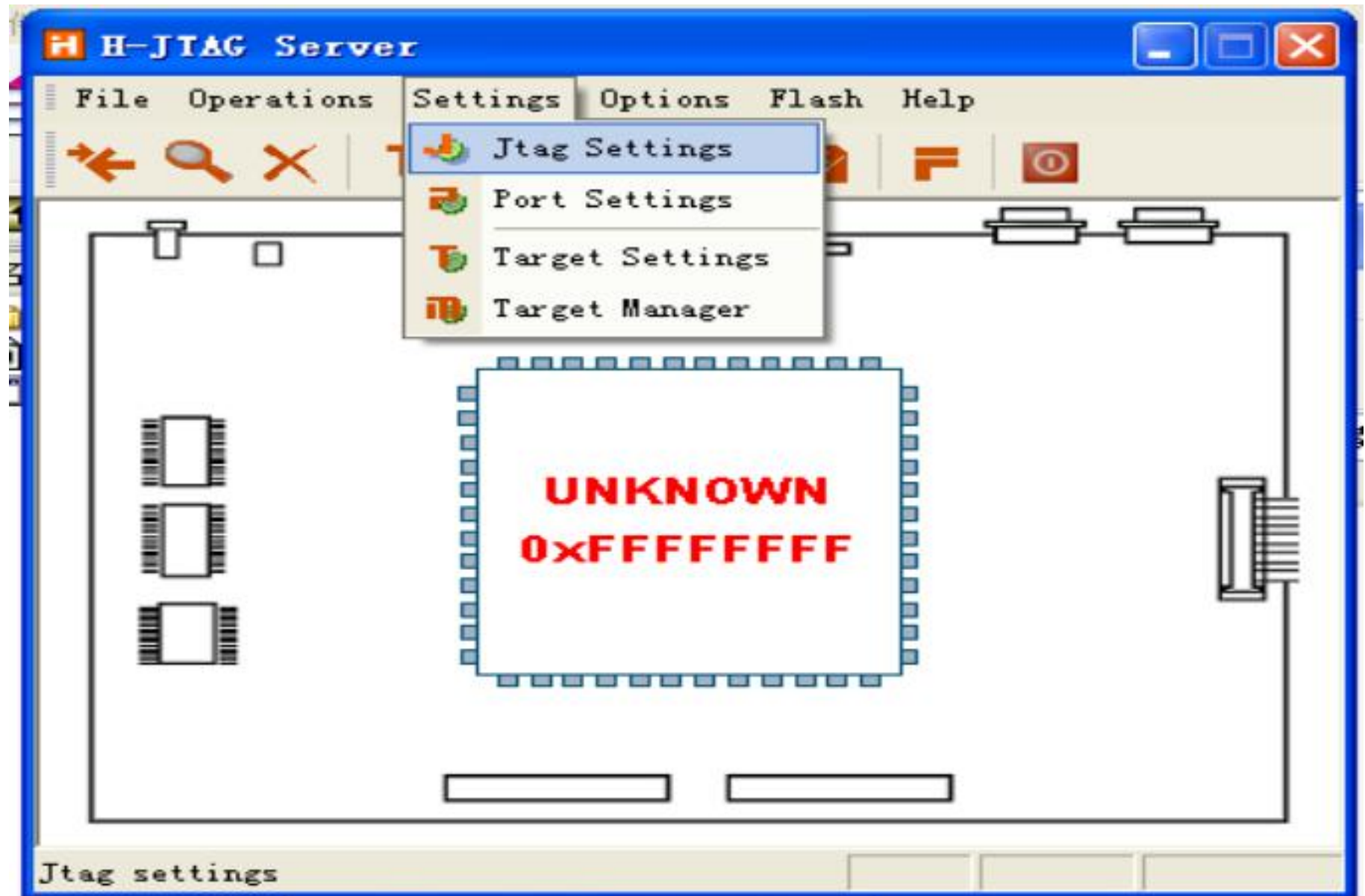
Using ADS

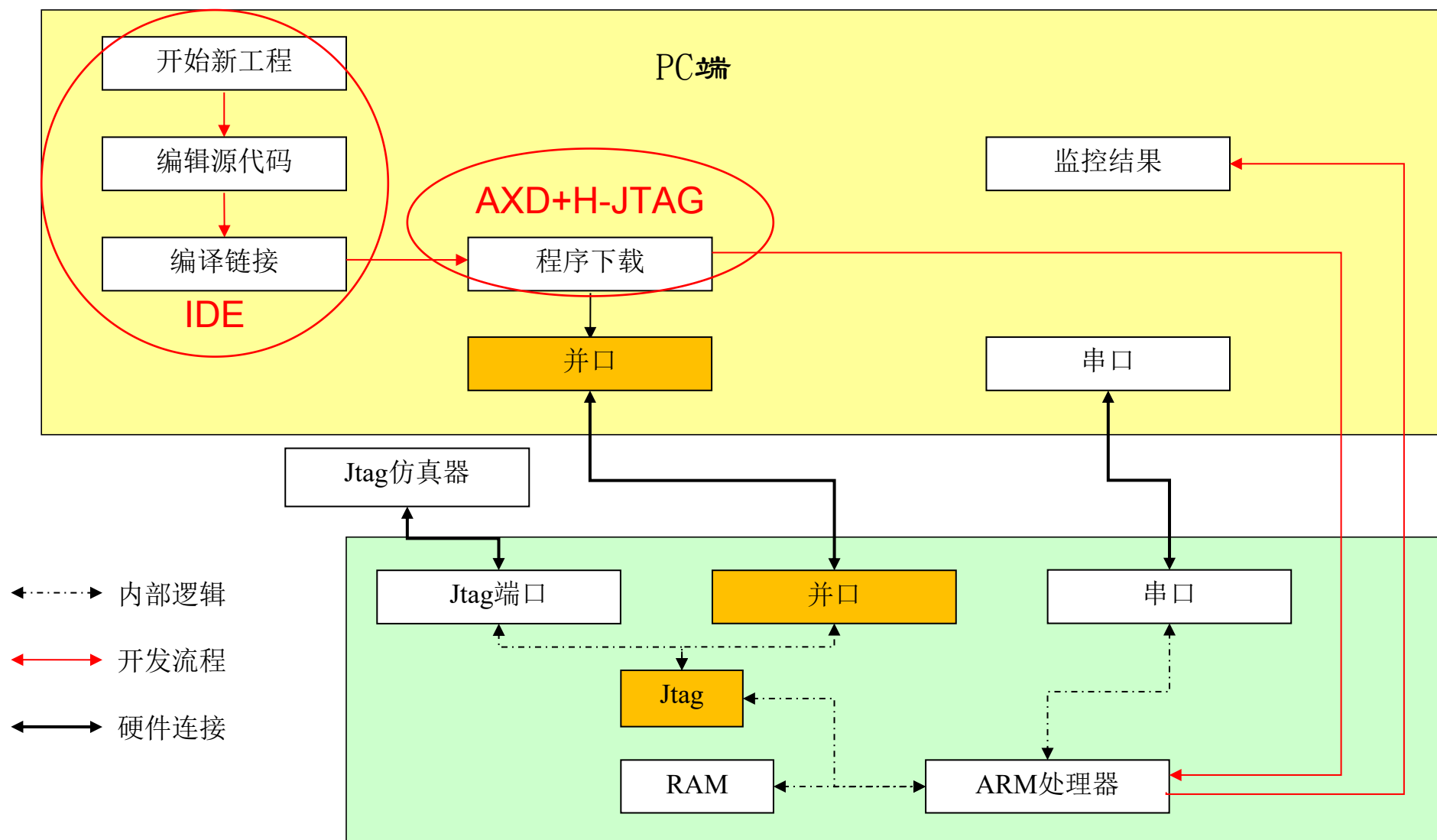
➤ H-JTAG



Using ADS

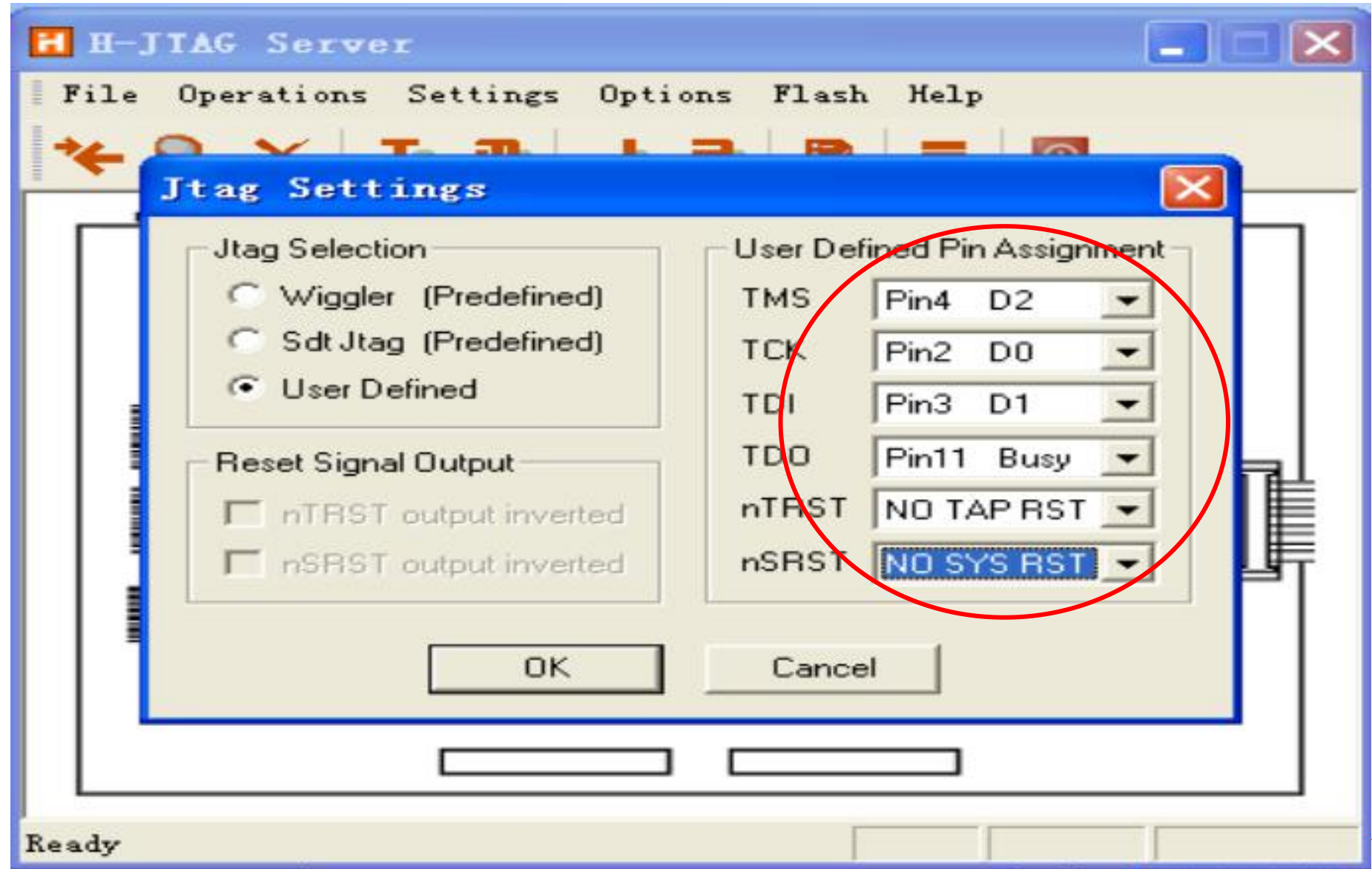
➤ H-JTAG





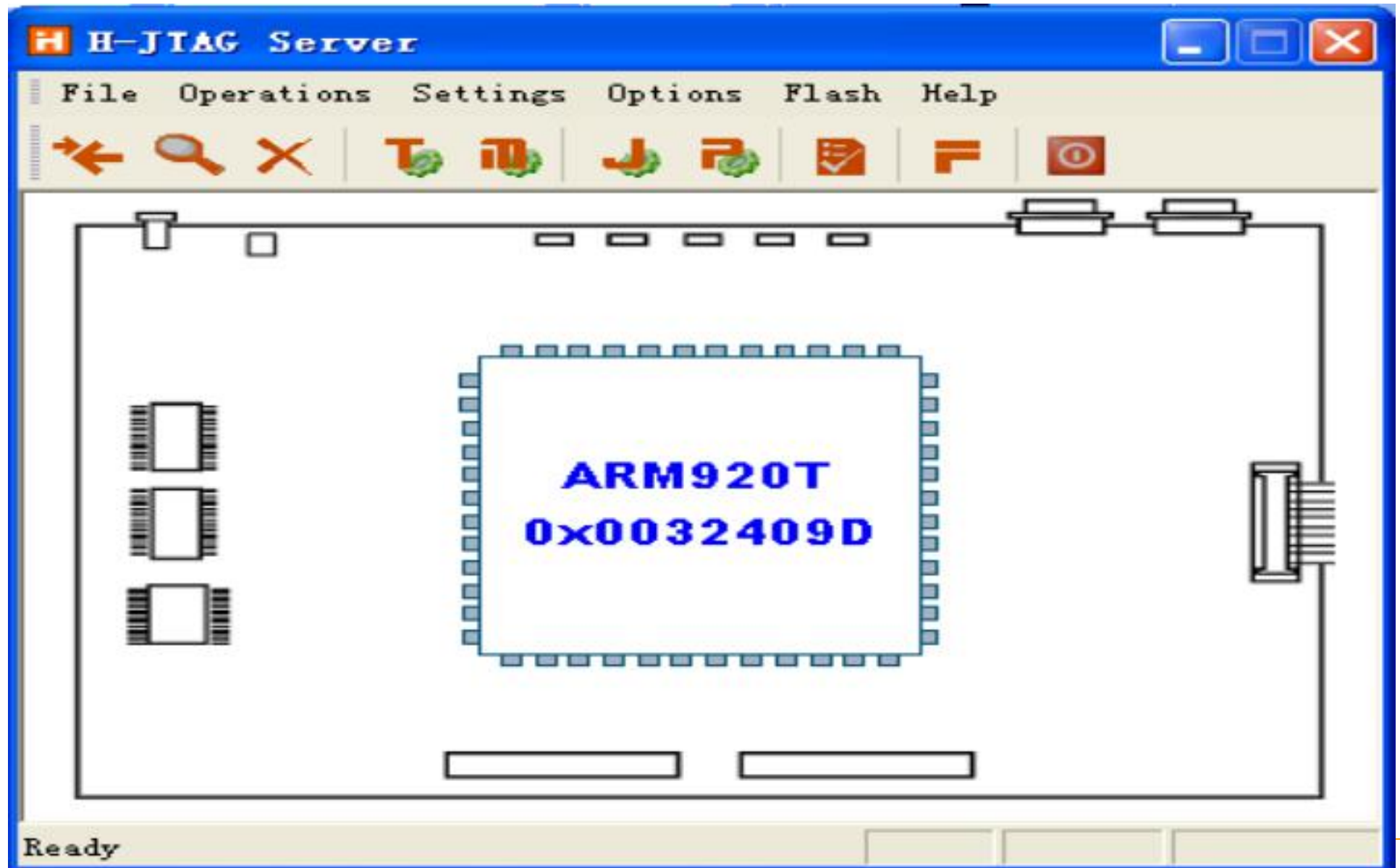
Using ADS

➤ H-JTAG



Using ADS

➤ H-JTAG



➤ Embedded Software Engineering

ENJOY THE COURSE !!!