

1

ARM_STAR 仿真器简介

功能介绍

ARM_Star 仿真器是一款为广大 ARM 工作者设计的高性能仿真器，ARM_Star 仿真器完全实现 ARM RDI 1.5 和 RDI 1.51 协议，与 ARM Multi-ICE™ 兼容，支持所有含有 Embedded-ICE Logic 的 ARM 内核 CPU，在调试软件的控制下，ARM_Star 仿真器可以停止、启动 ARM CPU 的运行，用户通过 ARM_Star 仿真器察看、修改寄存器，存储器，设置断点、单步执行，下载烧写 Flash 程序等。

主要特点：

程序下载速度快

与 PC 机的通信速度最高可达到 12M，下载速度达到 100K---200Kbytes 每秒，单步速度可达 85 步每秒。

支持调试软件多

支持的调试软件有 ADS、SDT、GreenHill Multi2000、IAR EWARM 等，极大地方便了用户。

支持 CPU 种类丰富

支持几乎所有含 ARM 内核 CPU，如 ARM7、ARM9、ARM10、Intel Xscale 等

JTAG 口频率可编程，支持各类不同性能目标板

开放式接口

支持多个内核串联的 CPU 调试，极大的体现了 JTAG 调试器的优势。

无需外加电源

ARM_Star 仿真器功耗低，可以直接从目标板取电，和目标板共用一个电源系统，使用及外出携带方便

免费升级

由于 ARM_Star 的 FirmWare 是每次上电时直接下载到仿真器的，所以，用户可以随时使用最新版本。

2

ARM_STAR 仿真器 硬件使用方法

2.1 ARM_STAR 仿真器对 PC 机的配置要求:

系统环境

Win95, Win98, Windows Me, Win2000, Windows XP
Windows NT 4.0 及以上版本。

硬件环境

Pentium 200MHz 以上 CPU

64M 内存

300M 硬盘空间

计算机并行口

2.2 ARM_STAR 仿真器的功能框

图



DB25 并行接口

用于仿真器跟 PC 机的通信，连接线为普通并口延长线。

20 针 IDC20 JTAG 接口

用于连接仿真器和目标板的 JTAG 口。

电源指示灯

当打开目标板电源，并且仿真器与目标板已正确连接时，电源指示灯就会亮起。

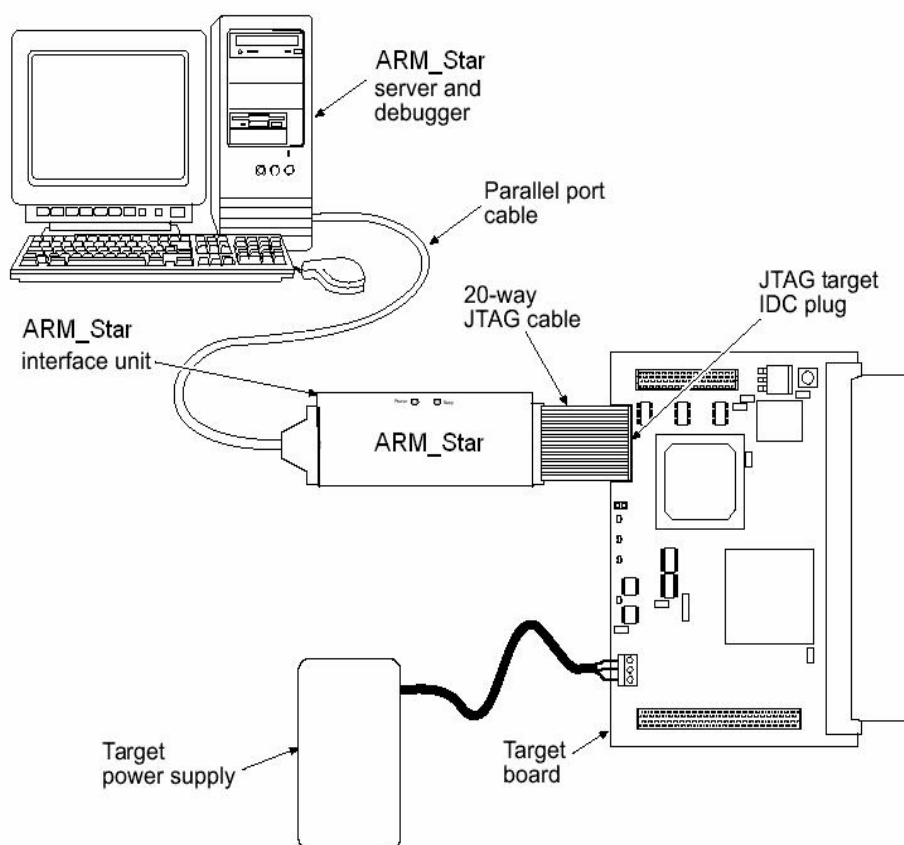
运行指示灯

当仿真器工作时，如：走单步等，运行指示灯闪烁。

JTAG 接口说明

ARM_STAR 默认的 JTAG 接口为 20Pin 标准接口, 如果用户目标板的 JTAG 接口是 14Pin 标准, 请用附件中的 JTAG 转换模块实现 20Pin 到 14Pin 的转换。

ARM_STAR 与目标板连接示意图



用户首次连接出现问题时, 请进入计算机的 **BIOS** 中更改并口模式。

电源需求:

ARM_STAR 仿真器是直接由目标板经过 JTAG 口供电。仿真器 JTAG 口输出信号的电压能自动随目标板电源电压的变化而变化。

3

JTAG 仿真器的介绍

ARM JTAG 接口定义及转换

ARM Jtag 仿真器工作时，只与 CPU 的 JTAG 边界扫描链通信，不占用任何 CPU 资源，不影响 CPU 全速运行速度，ARM CPU 的 JTAG 接口一般有两种：14Pin 和 20Pin，这两种接口的电气特性相同，只是管脚分布不同，可以通过转换模块进行转换，管脚定义如下：

V_{ref}	1	2	V_{supply}
nTRST	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
RTCK	11	12	GND
TDO	13	14	GND
nSRST	15	16	GND
NC	17	18	GND
NC	19	20	GND

20Pin JTAG 口

V_{ref}	1	2	GND
nTRST	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
TDO	11	12	nSRST
VCC	13	14	GND

14Pin JTAG 接口

20Pin JTAG 接口定义表:

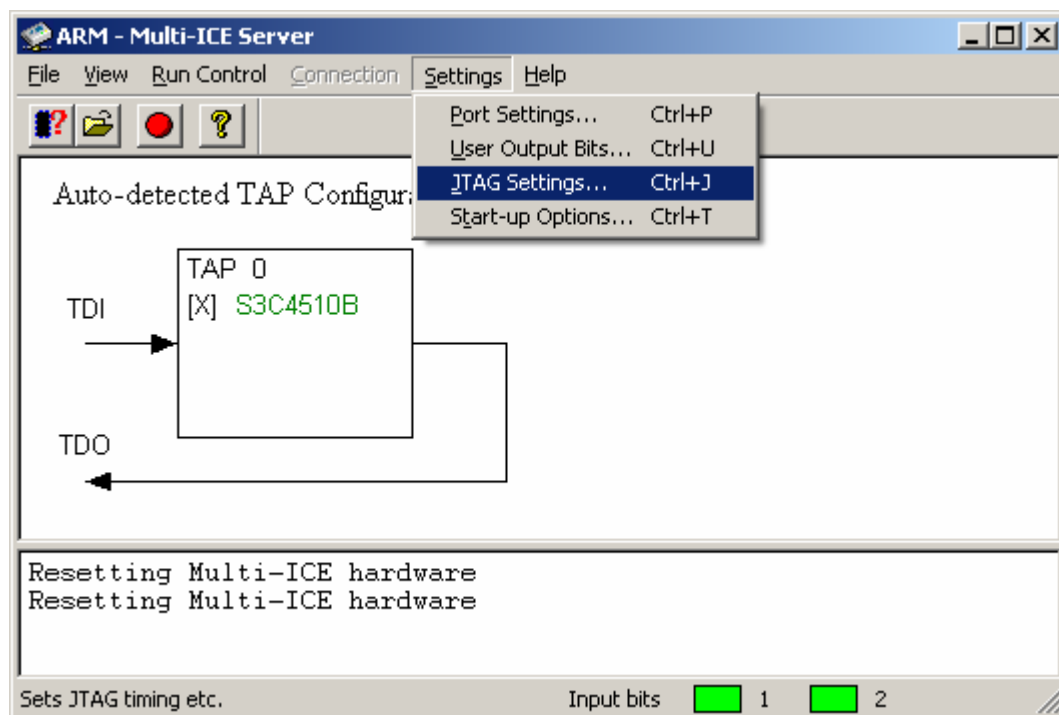
管脚编	管脚定义	信号属性	描述
1	Vref	Input	参考电压, 接V _{supply}
2	V _{supply}	Input	电源
3	nTRST	Output	JTAG TAP 复位, 接上拉
4	GND	Ground	接地
5	TDI	Output	JTAG 数据输入
6	GND	Ground	接地
7	TMS	Output	JTAG 模式选择
8	GND	Ground	接地
9	TCK	Output	JTAG 时钟信号
10	GND	Ground	接地
11	RTCK	Input	目标板反馈时钟信号, 不用时接地
12	GND	Ground	接地
13	TDO	Input	JTAG 数据输出
14	GND	Ground	接地
15	nSRST	Inout	目标板系统复位信号, 上拉
16	GND	Ground	接地
17	NC	Reserved	悬空
18	GND	Ground	接地
19	NC	Reserved	悬空
20	GND	Ground	接地

JTAG 的 TCK 信号频率设置

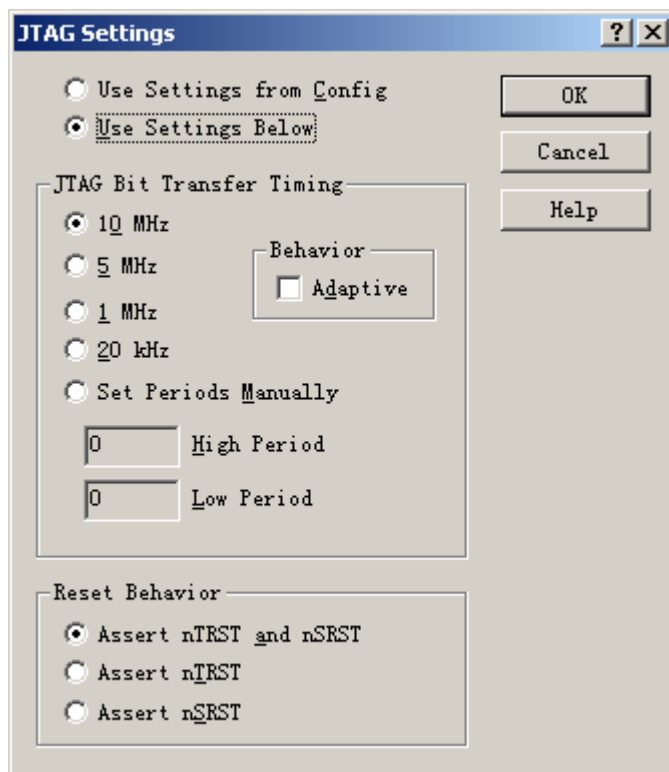
ARM_Star 仿真器的 JTAG 频率可以通过设置参数值来灵活调整, 以便与不同频率特性的目标板相联接, 设置 TCK 频率的方法有两种, 菜单设置和文件设置。

菜单设置

打开 Multi-ICE Server 后请按照以下步骤操作



在下图菜单中直接选中用户所需要的频率，或者在 Set Periods Manually 中直接填写 TCK 的高低电平的时间值，具体不同的 High Period 和 Low Period 的值对应的频率请参考附件中的 Jtag 设置频率表。



配置文件设置：

在配置文件中加入下面文字：

[Timing]

Low = 9 ; TCK 信号的高电平时间

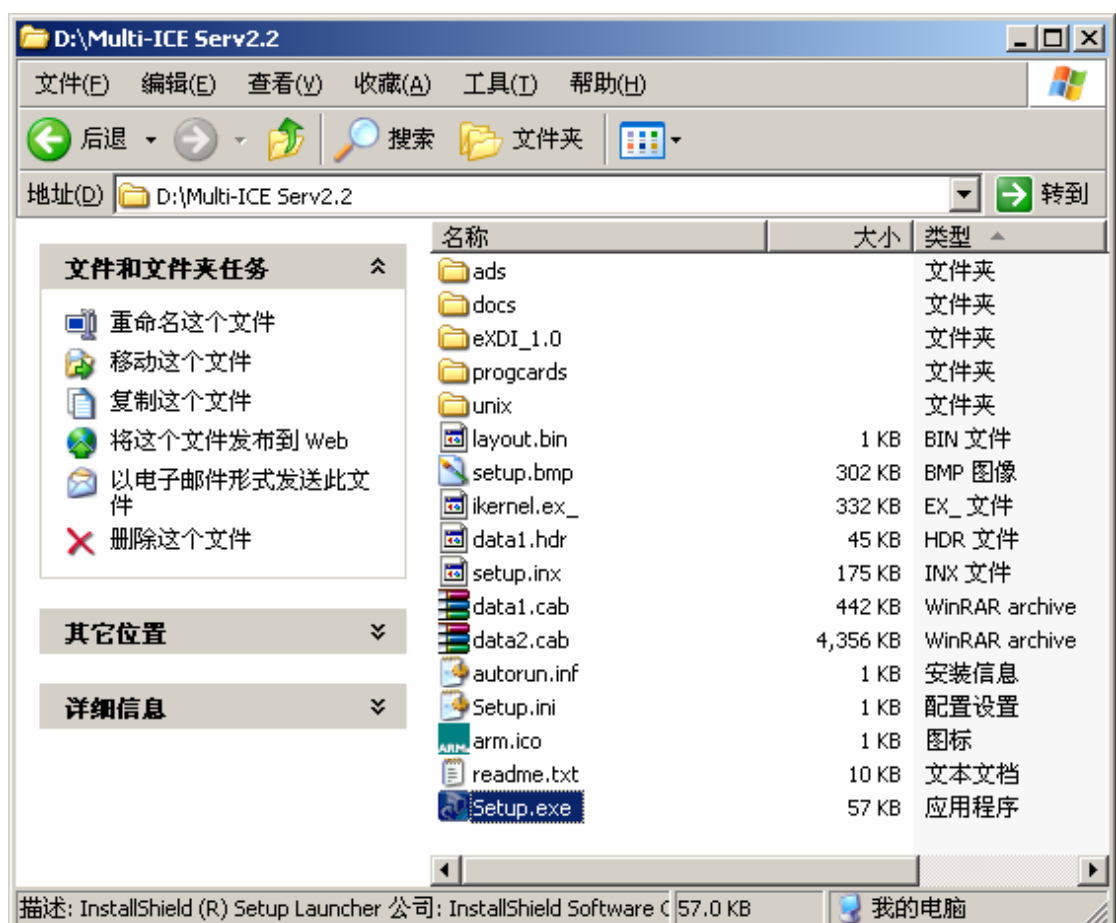
High = 9 ; TCK 信号的低电平时间

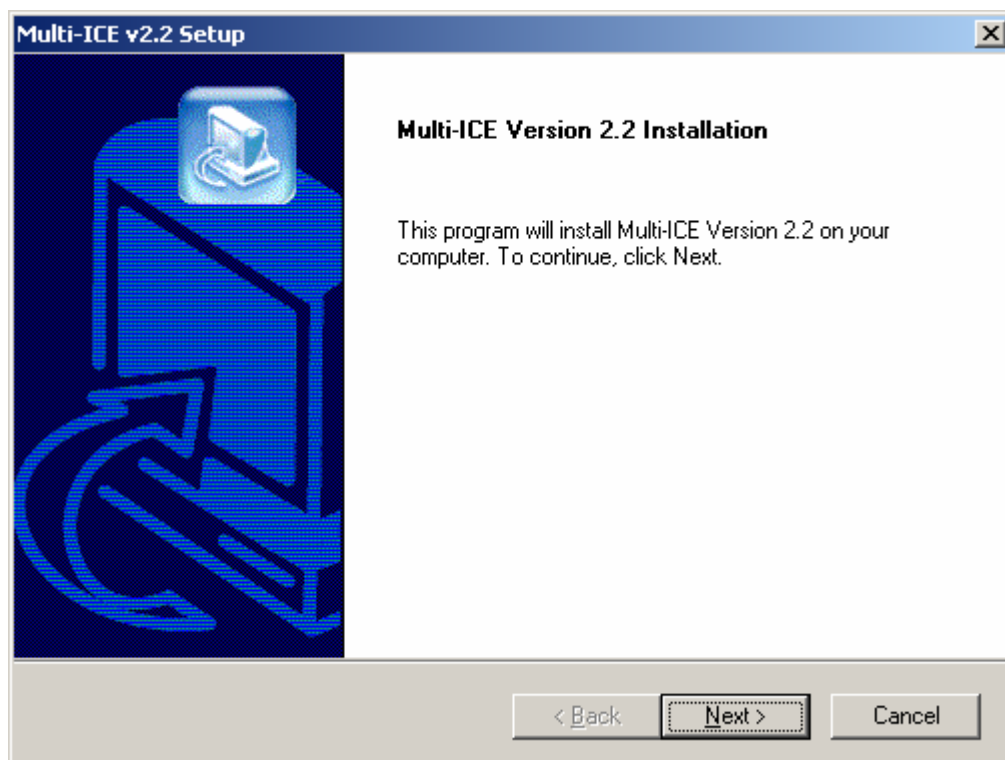
4

仿真器软件 Muti-ICE Server 的安装和使用方法

4. 1 安装 Multi-ICE Server

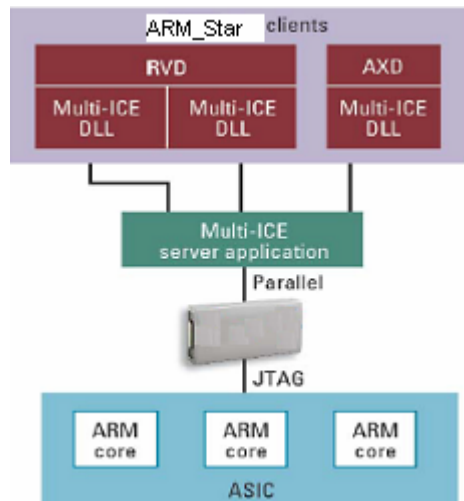
点击 Setup.exe





4. 2 Multi-ICE Server 工作原理介绍

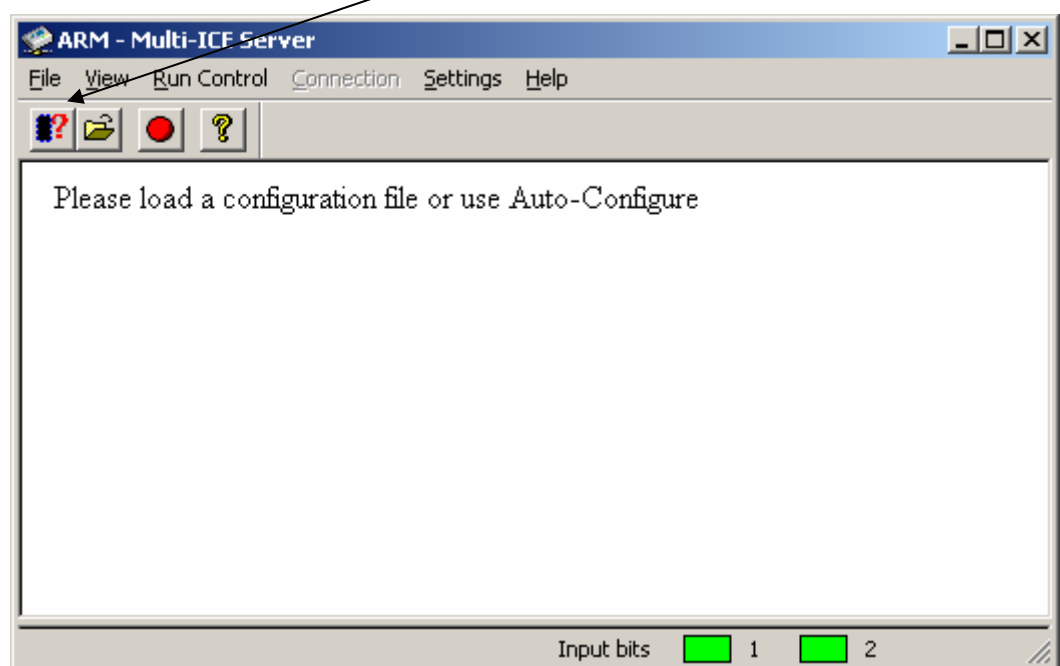
ARM_Star 仿真器是通过 Multi_ICE Server 来实现与其他遵从 RDI 接口标准的调试软件，如 ADS, SDT, GreenHills Multi2000、IAR EWARM 等的连接。



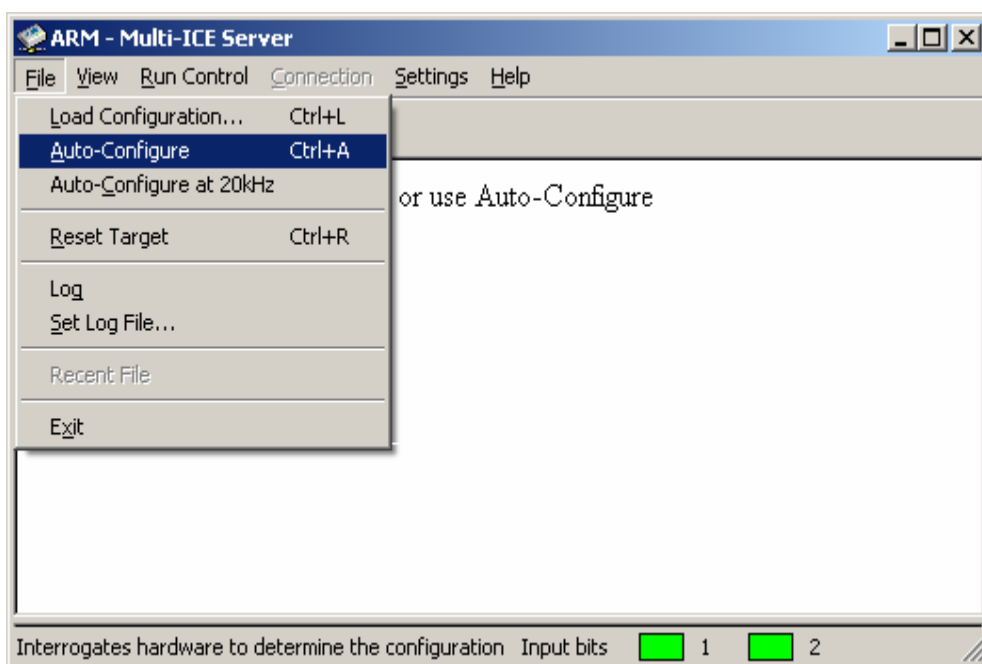
4. 3 Multi-ICE Server 典型应用

确认仿真器连接正确，然后打开电源，启动 MultiICE-Server。

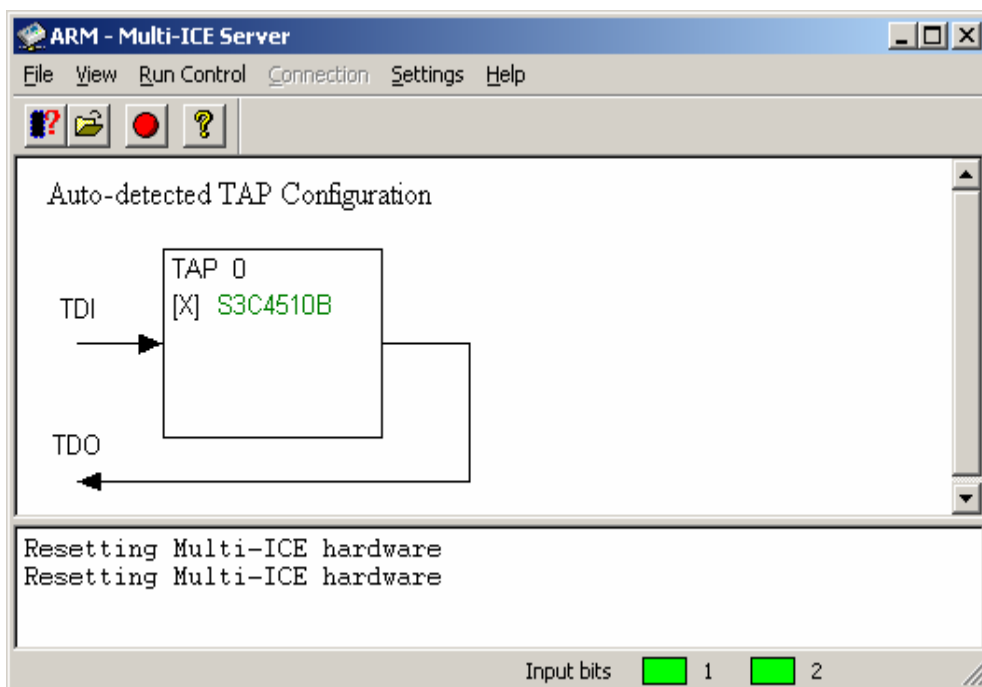
点击 ，在窗口中点击自动配置图标



或者从菜单中选择自动配置
选项



如果目标板工作正常，会出现以下界面：



4. 4 Muliti-ICE Server 配置文件的使用方法

一般情况下，使用 Muilti-ICE 的自动配置功能，都可以自动识别目标 CPU 的种类，在有些特殊情况下，仿真器不能自动识别目标 CPU 时，可以使用手动配置的方法实现连接。

首先，编辑一个 **test.cfg** 文件，典型内容如下：

```
[TITLE]
My Cfg File          ; 您的配置名称，任意写

[TAP0]
ARM7TDMI              ; 目标系统中包含TAP0 控制器
                      ; TAP0 控制器上连着一个ARM7TDMI
                      ; 如果您的CPU内核是ARM920T，那么您在这
                      ; 个位置上就写上ARM920T。

[Timing]
High = 8              ; JTAG 口TCK信号的时序设置
Low = 8               ; TCK 信号的高电平持续时间
Adaptive = OFF        ; TCK 信号的低电平持续时间
                      ; RTCK 功能开或关（ON or OFF）

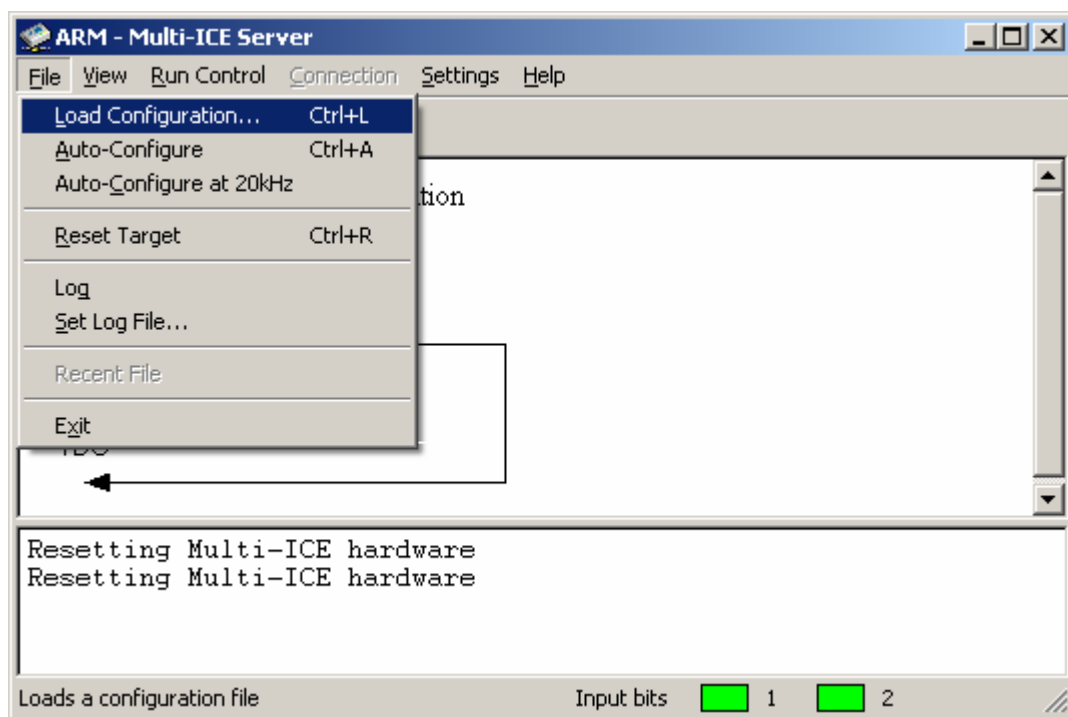
[TAPINFO]
YES                   ; 显示TAP的详细信息

[Reset]
nTRST                ; 复位信号的选择
```

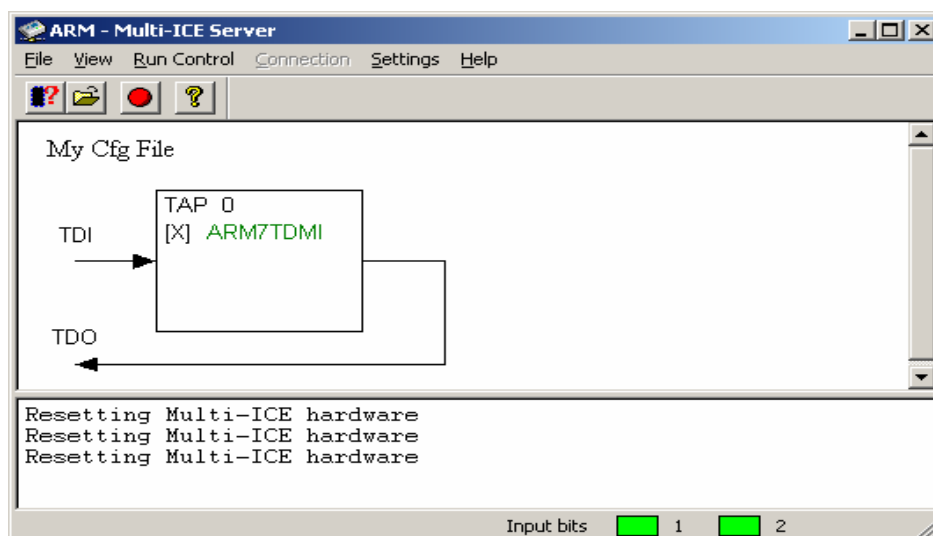
在 Cfg 文件中，[TAP0]表示 CPU 内核的类型，常用的符合语法的内核名称有：

ARM7:	ARM7TDMI	ARM7TDMI-S	ARM720T	ARM710T	ARM7DMI
	ARM740T	ARM7TDI-S	ARM7EJ-S	... 等	
ARM9:	ARM9TDMI	ARM920T	ARM940T	ARM922T	ARM9E-S
	ARM946E-S	ARM966E-S	ARM926EJ-S	... 等	
ARM10:	ARM1020E	ARM1022E	ARM1020T	... 等	
Intel® XScale™ :		XScale-80321	XScale-PXA210	XScale-PXA250	
		XScale-80200	XScale-IXP425	XScale-IXP2400	
		XScale-IXP2800	... 等		

然后，在菜单中 File->Load Configuration



这样，根据用户要求定制的配置就生成了。



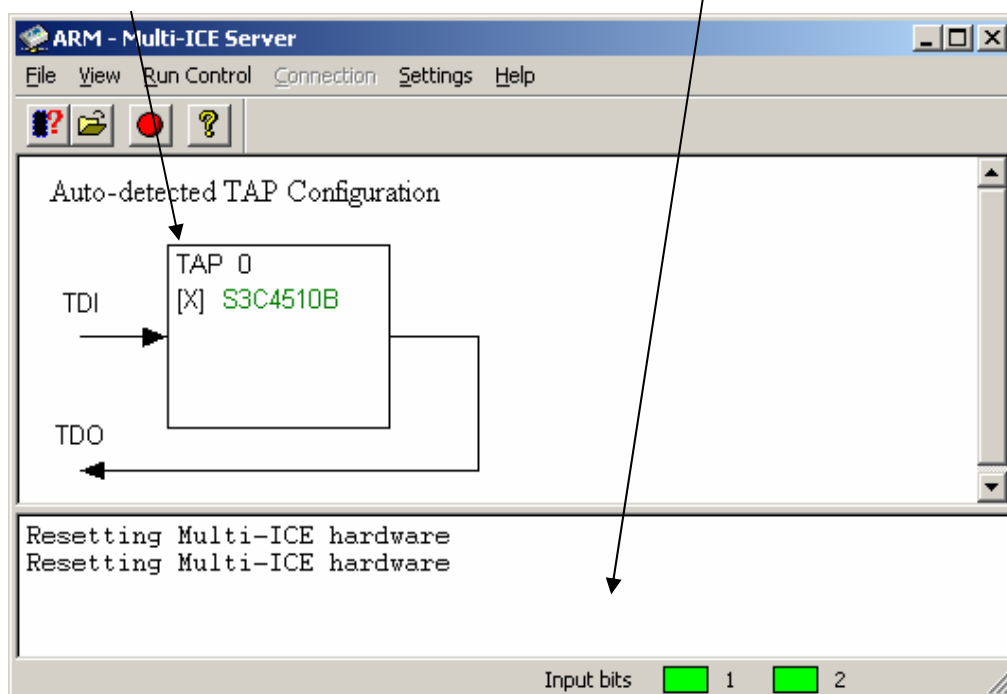
4.5 Multi-ICE Srvr 窗口显示信息

Multi-ICE Srvr 的窗口分两部分：

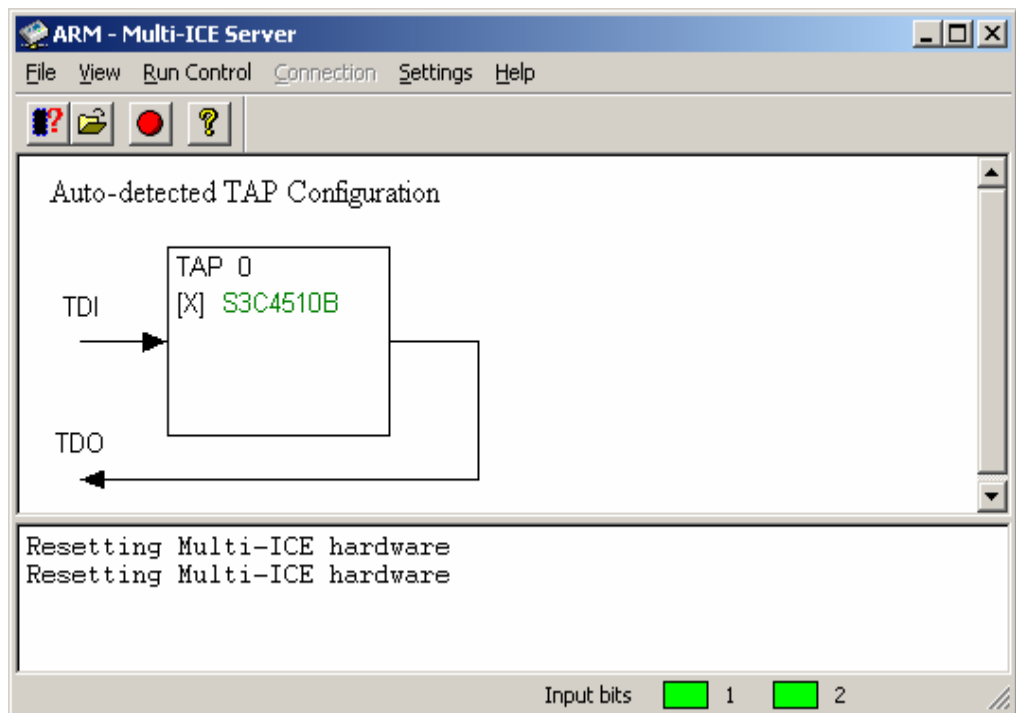
CPU 内核信息窗口

和

调试信息窗口



在 CPU 内核信息窗口中显示的是目标 CPU 的内核种类、数量、连接状态等信息，如下图中：



自动检测显示目标是一个 CPU 是 **Samsung S3C4510B** 的 CPU 的单内核系统，在 **TAP** 配置显示区中用图形的方法直观的把这一结果显示了出来。内核的类型名称 **S3C4510B** 显示成绿色，并且前面加了一个字母 **X**。这表示 **server** 现在还没有连接到任何一个调试程序上去。

处理器类型名称前面的字母叫是状态位，共有四状态位显示：

[S] 处理器处于暂停状态

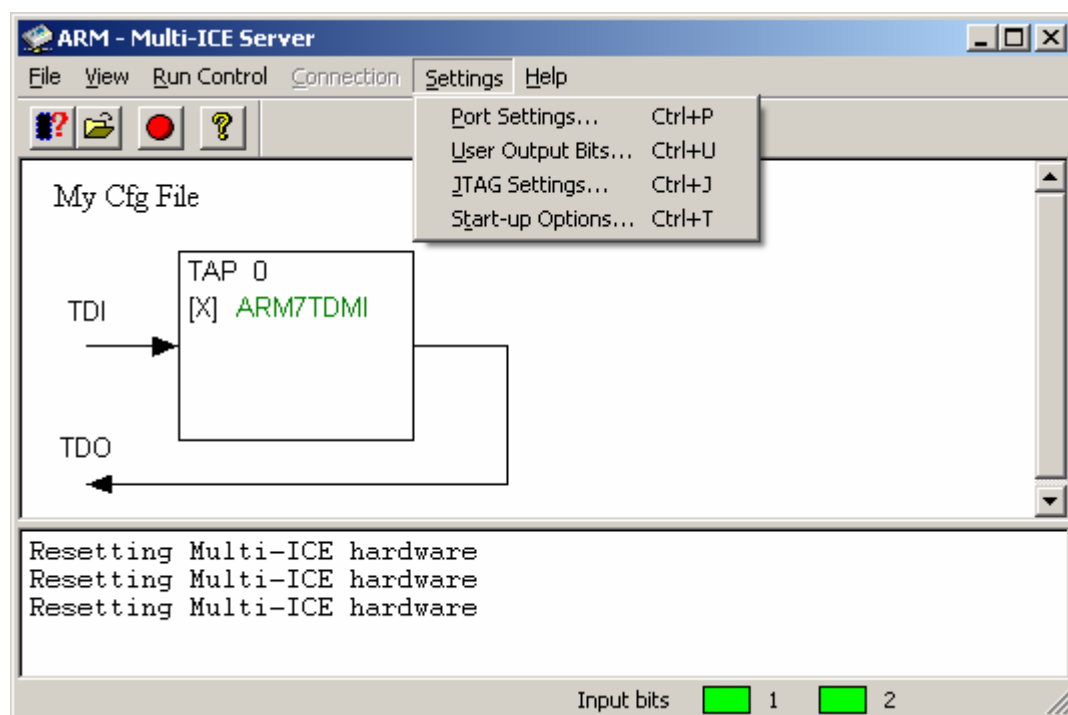
[R] 处理器忙（运行状态）

[D] 处理器处于下载状态

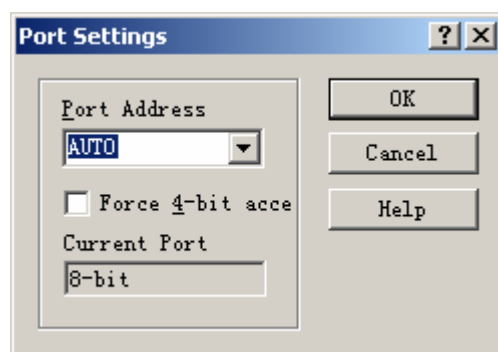
[X] 处理器类型未知或没有被调用

4.6 Multi-ICE Server 其它菜单及设置

在 Multi-ICE Server 的设置中，Setting 菜单比较重要



4.6.1 Setting->Port Setting



Port Address: 设置计算机的并口及其地址选项有:

Auto (自动配置, 默认值)

LPT1 - 0x378 (并口 1)

LPT2 - 0x278 (并口 2)

Force 4-bit access:

4 位并口模式, 默认为不选此项, 如果您的计算机主板型号比较老, 可以选用此项。

AUTO (自动选择, 缺省状态)

LPT1 (选择 LTP1)

LPT2 (选择 LTP2)

Force 4-bit access

使用 4-bit 的数据传输方式, 缺省状态为未选中。当使用某些旧的并口时,

可以尝试使用该选项。

Current port Mode (只读, 不能改写)

显示当前的并口类型, 并口类型请在 PC 的 CMOS 中设置。

通常的并口类型有:

SPP

EPP

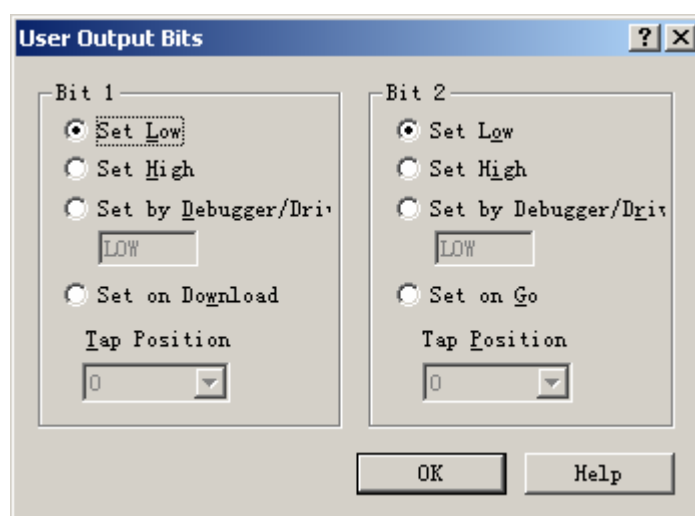
ECP

EPP+ECP

仿真器使用双向的并口数据总线。通常 ECP 或 EPP 类型能够符合要求，但是不同的主板对并口的定义有所区别，使用时，请尝试用不同的方式连接。

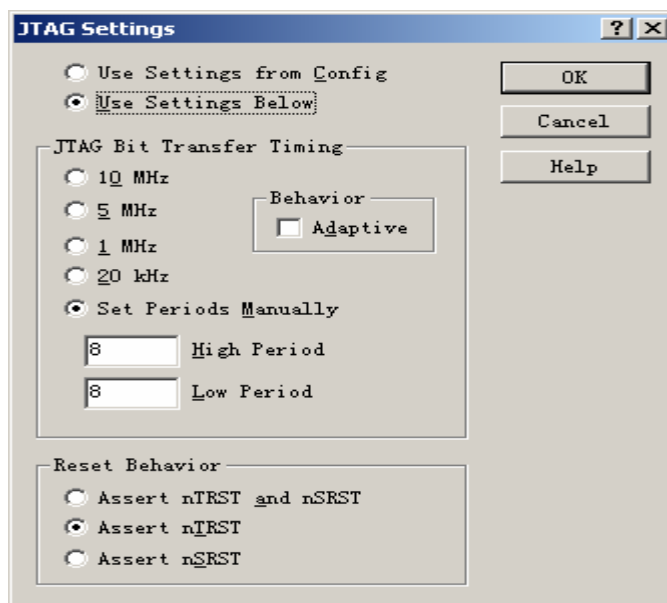
4.6.2 Setting->User Output Bits

设置仿真器接口的附加信号特性，不需要改动



4.6.3 Setting->JTAG Setting

设置 JTAG 口的时序和复位管脚。



JTAG Bit Transfer Timing:

可以设置固定的 TCK 频率，如 10M , 5M, 1M , 20K

也可以设置 TCK 信号的具体时序，Set Period Manually

High Period : TCK 信号的高电平时间

Low Period : TCK 信号的低电平时间

具体数值对应的频率请参考附录。

Behavior:

RTCK 功能开关，默认为关闭，当有些 CPU 的 JTAG 口包含有 RTCK 功能时，可以打开此开关来同步 TCK 和 RTCK 的频率。

Reset Behavior:

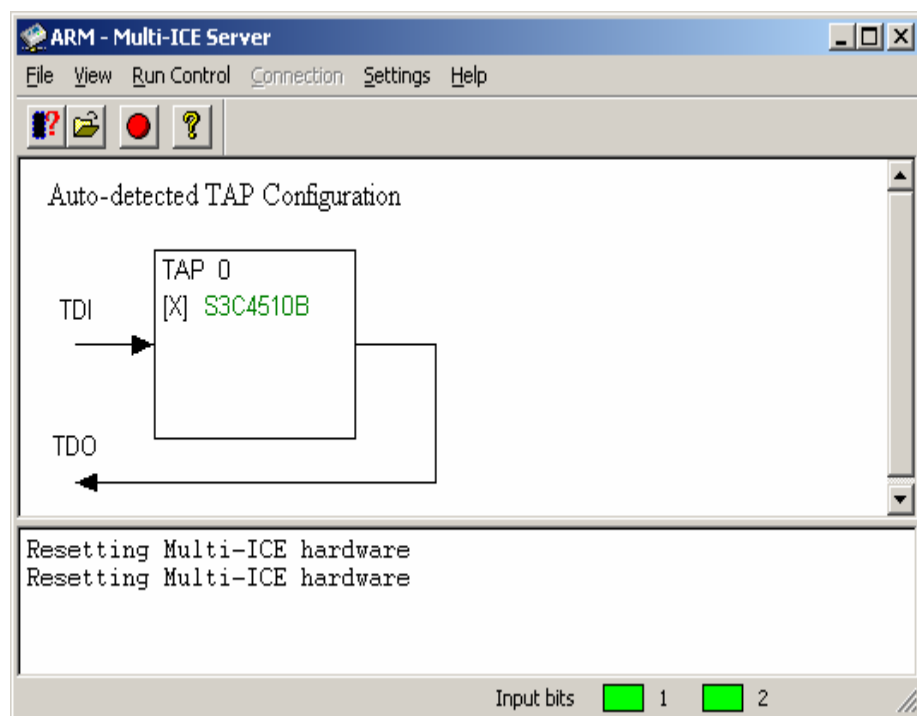
选择仿真器的 JTAG 口输出的复位信号种类。默认是 nTRST.

5

ARM_STAR 仿真器 和其它调试环境的连接

5.1 ARM_STAR 仿真器和 SDT2.51 的连接

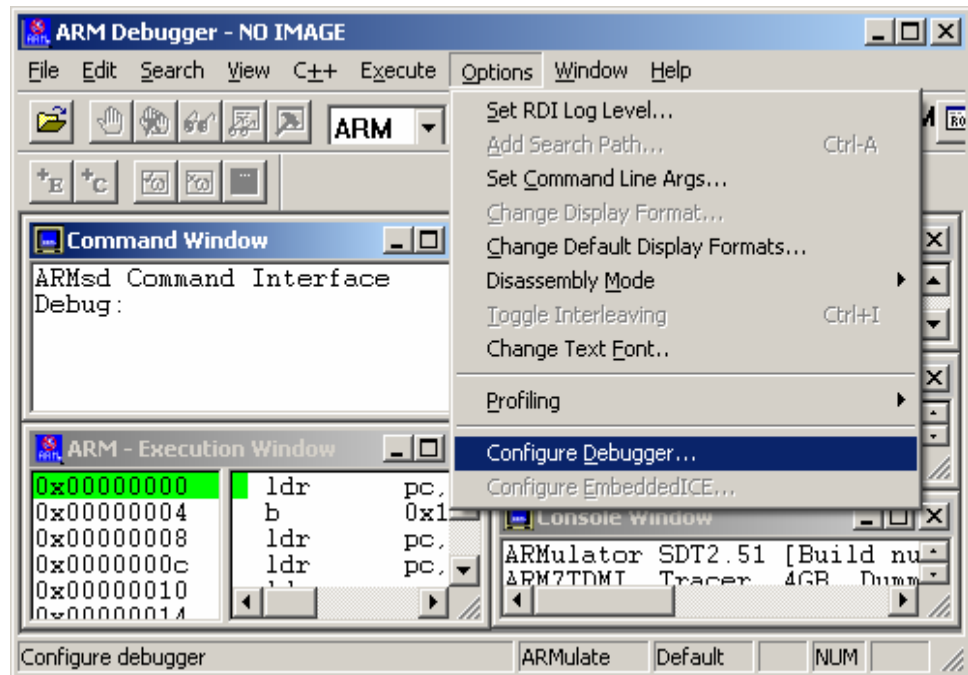
打开 Multi-ICE Server，保证连接正确，出现以下画面，不要关闭。



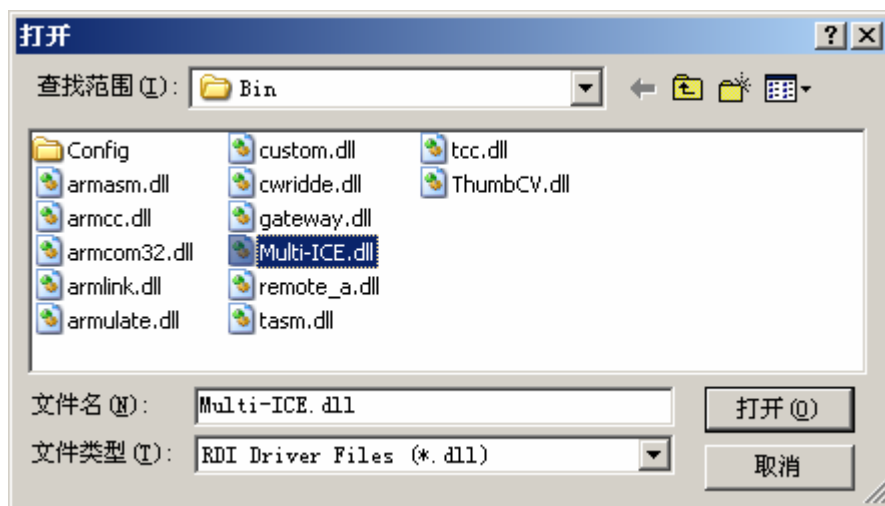
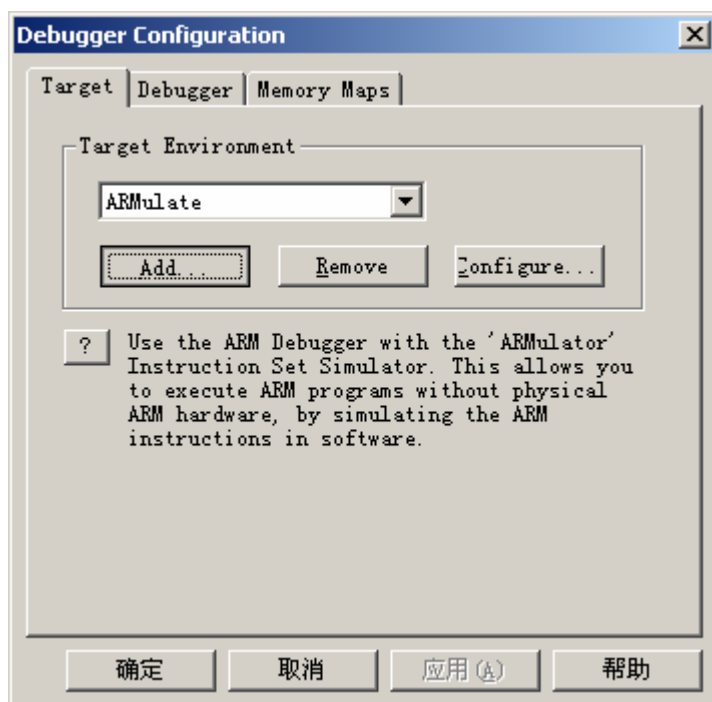
打 开



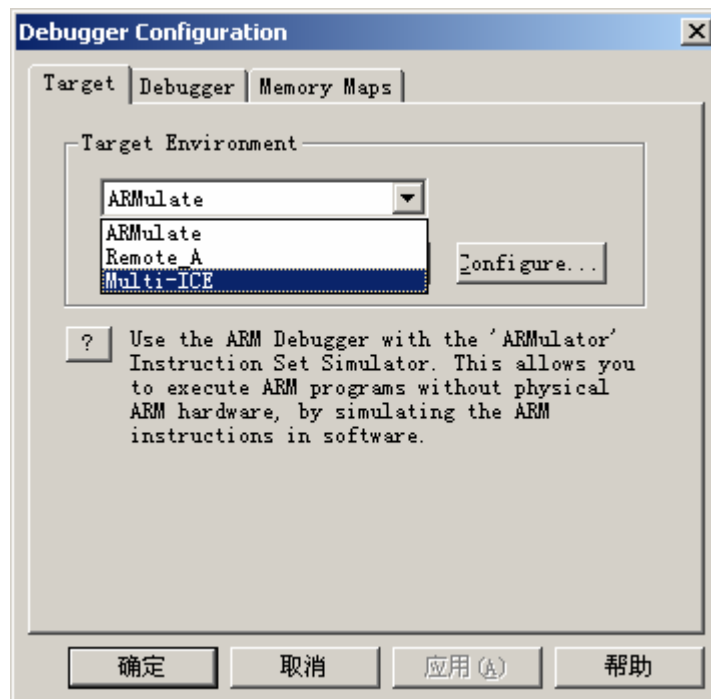
Option->Configure Debugger...



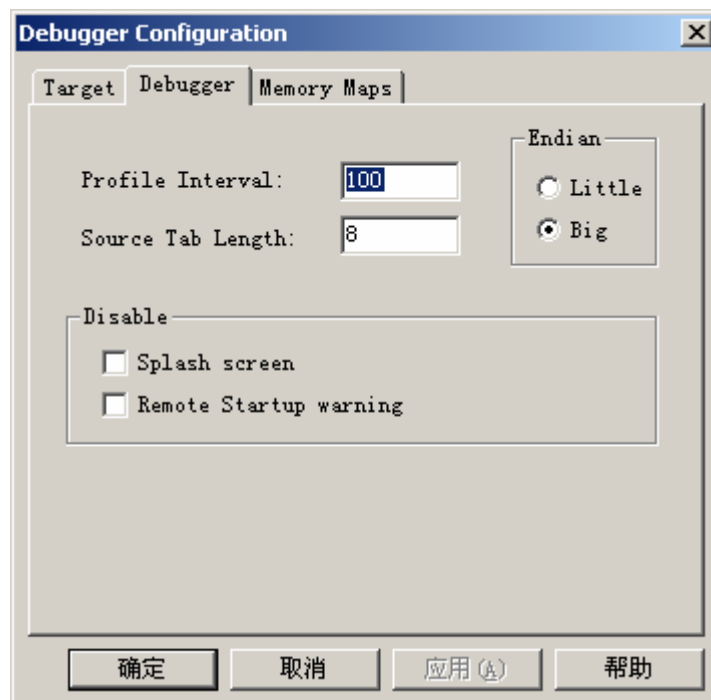
首次使用时，把 Multi-ICE.dll 拷贝到 SDT2.51 安装目录的 bin 目录下，然后在 Debugger Configuration 中添加



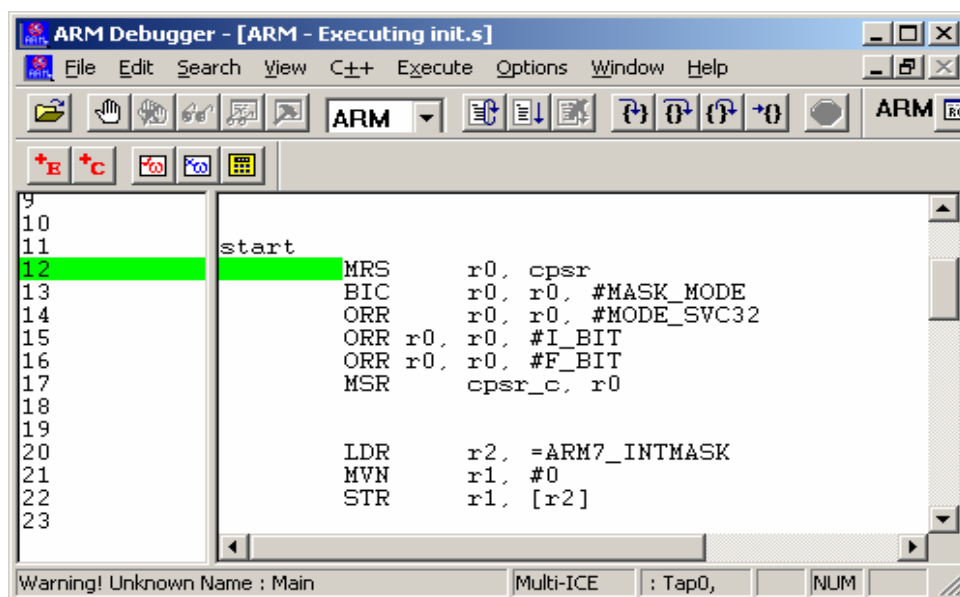
添加完成后，选择 Multi-ICE



选择存储器的大、小端模式

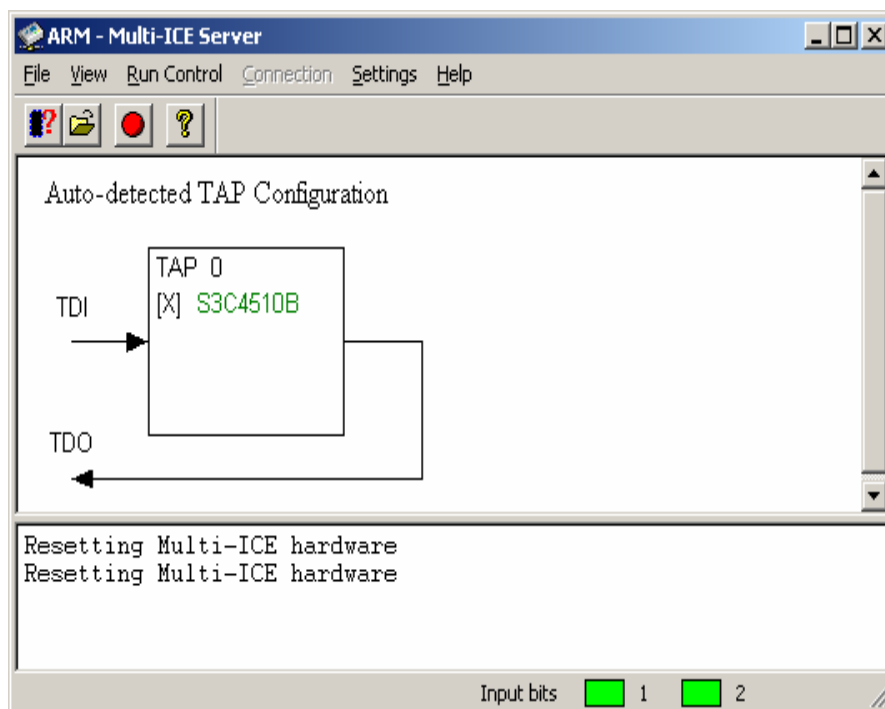


最后，打开 File->Load Image... 装载调试代码进行调试。



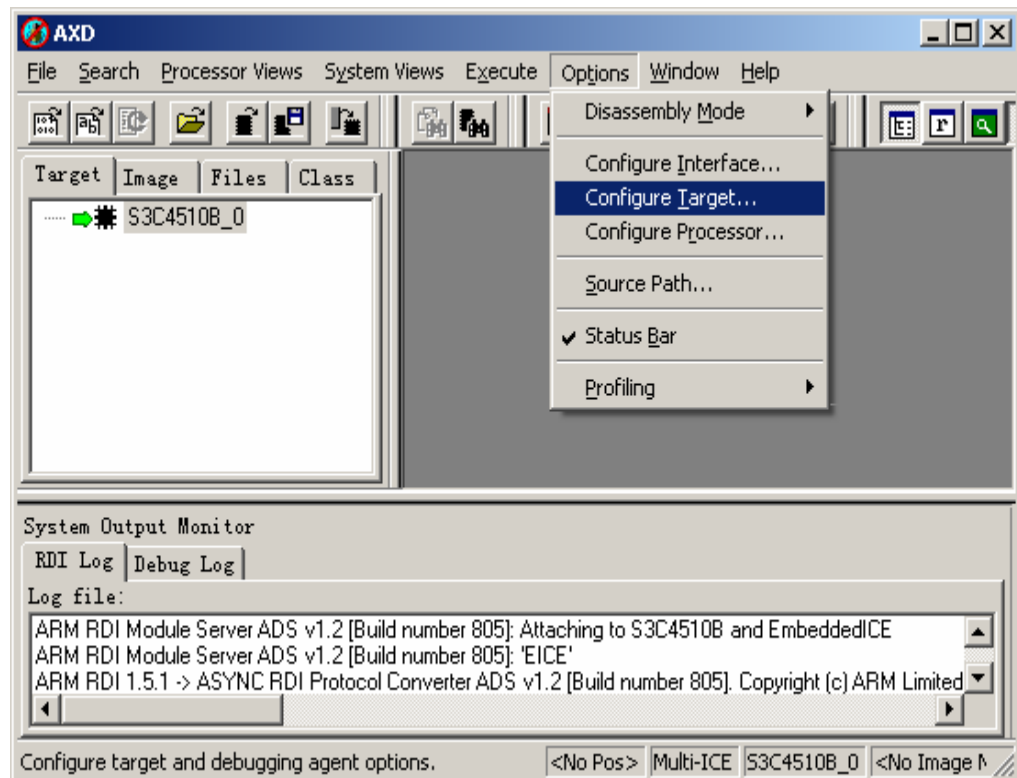
5.2 ARM_STAR 仿真器和 ADS1.2 的连接

打开 Multi-ICE Server，连接正确后，出现以下画面，不要关闭。

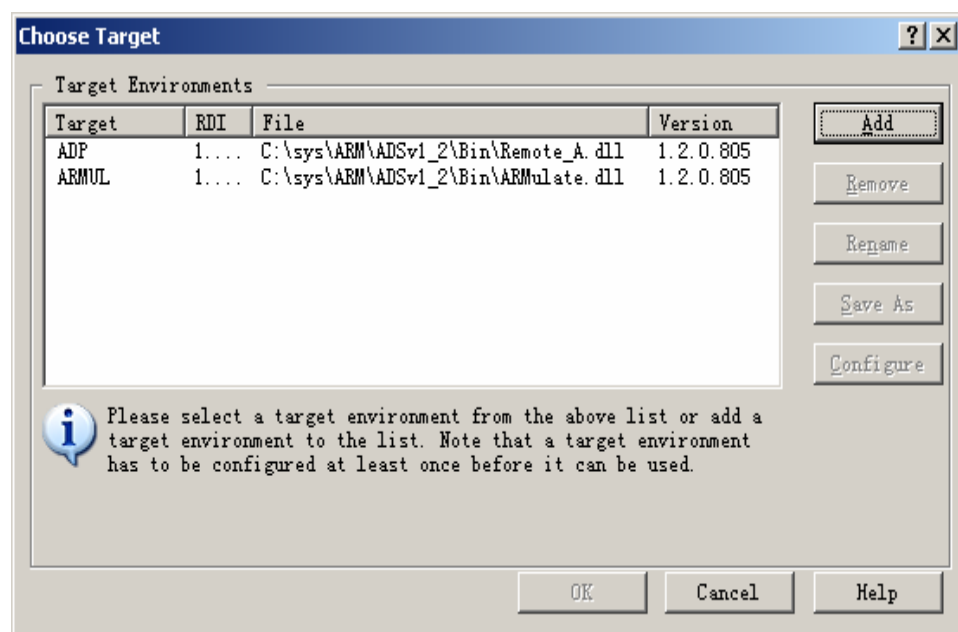


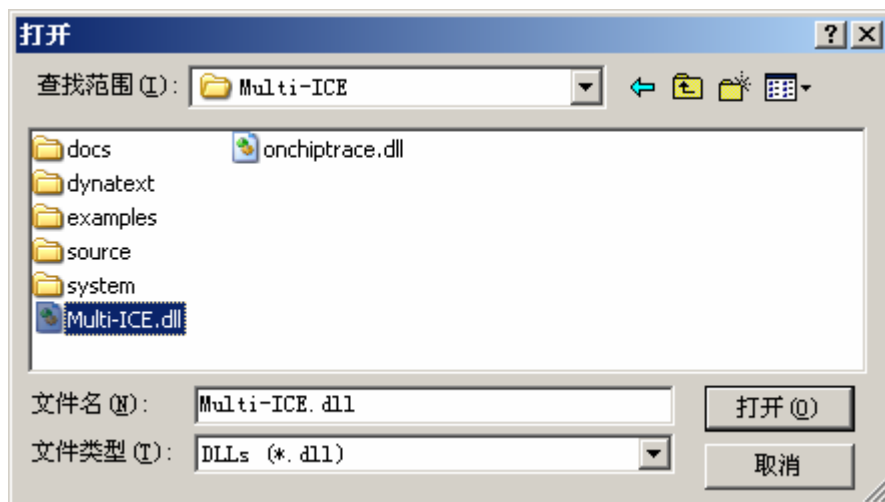


打开 AXD Debugger，在菜单中运行 Option->Config Target

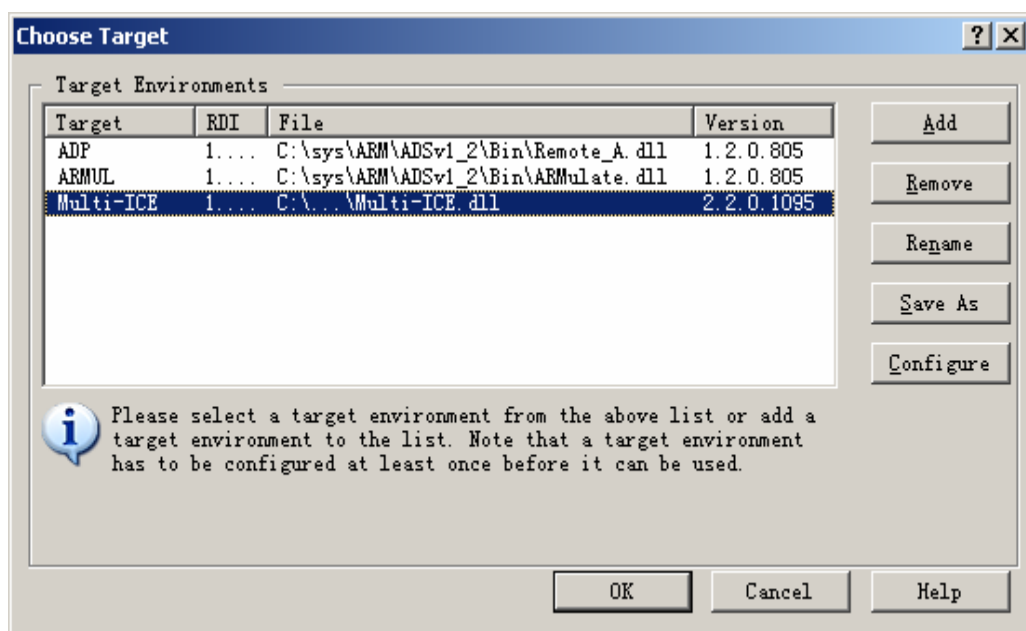


首次使用，需要添加 Multi-ICE.dll 文件

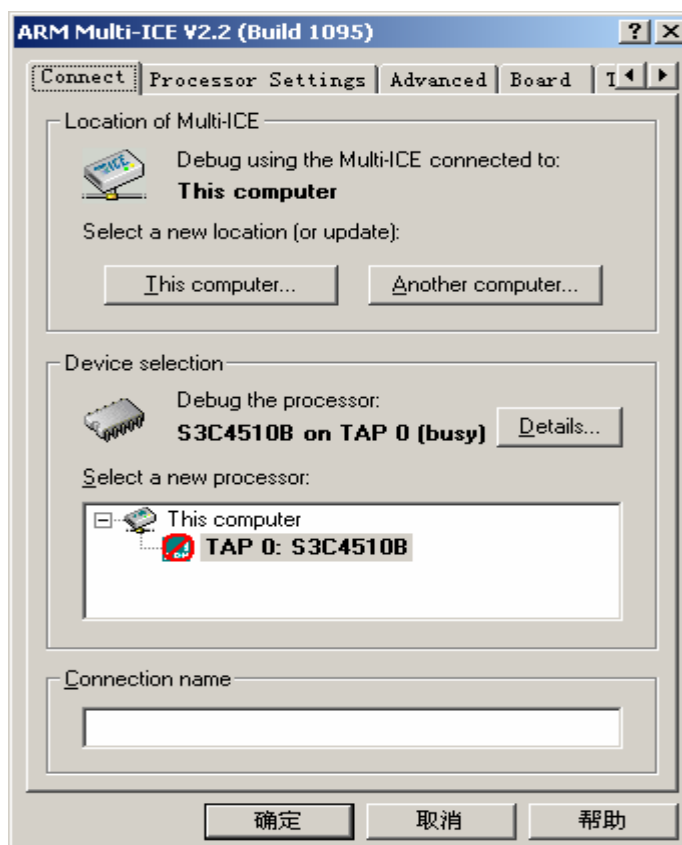
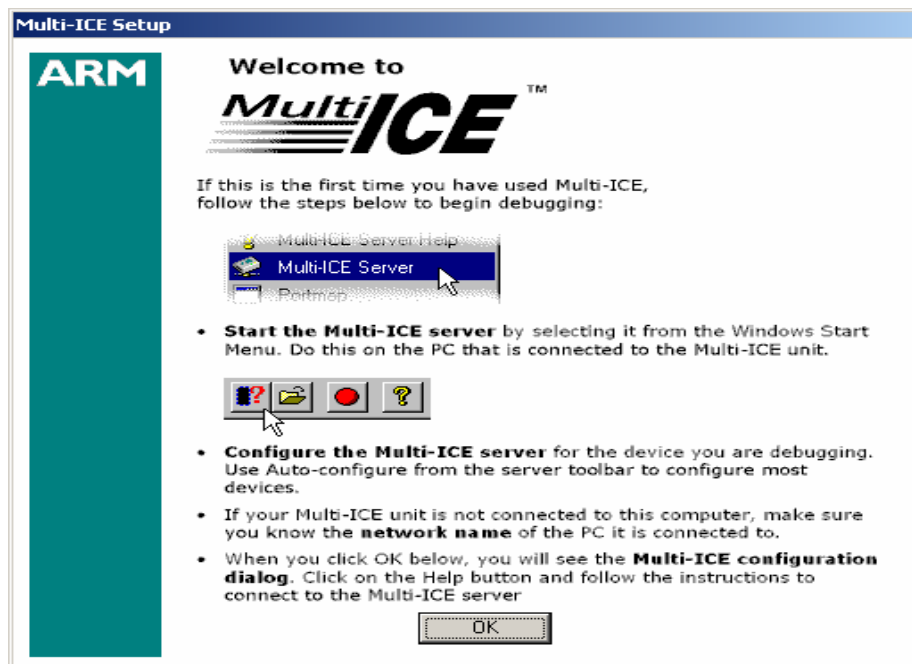




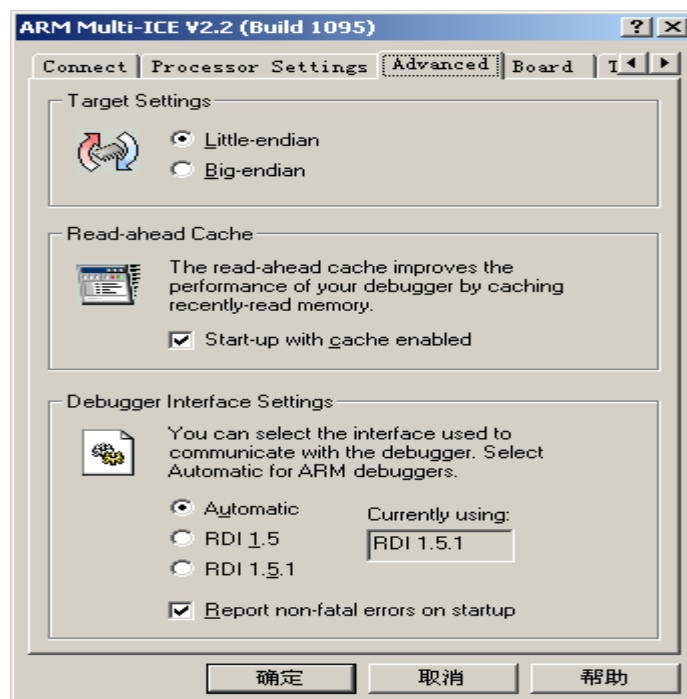
添加后，选择 Configure.



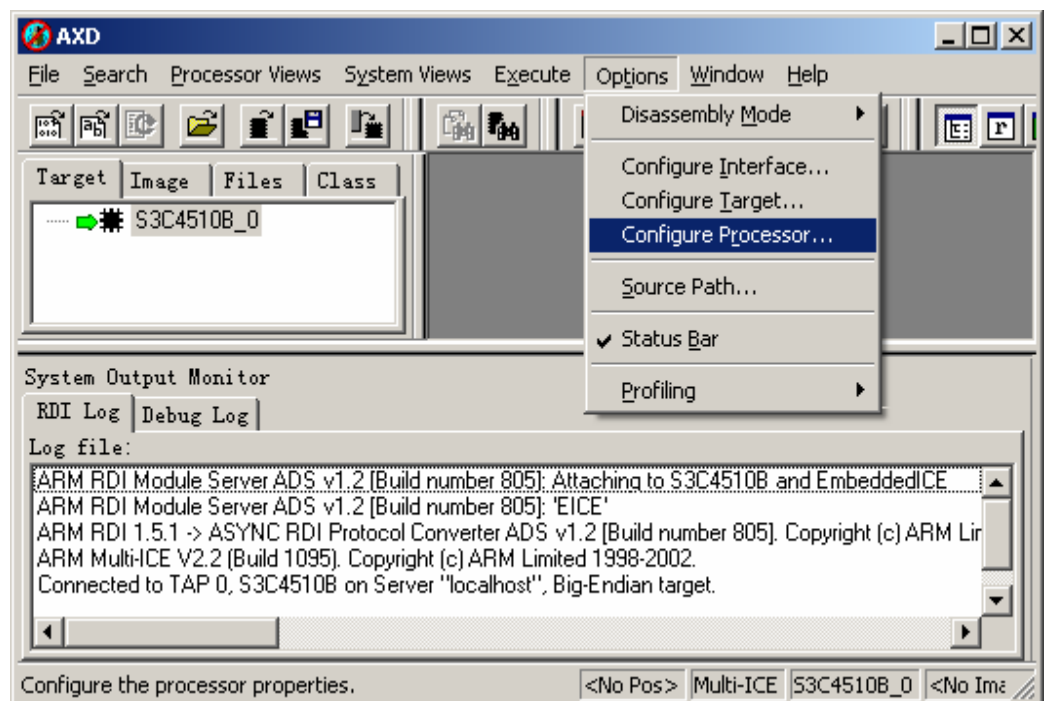
首次 Configure，出现以下画面。



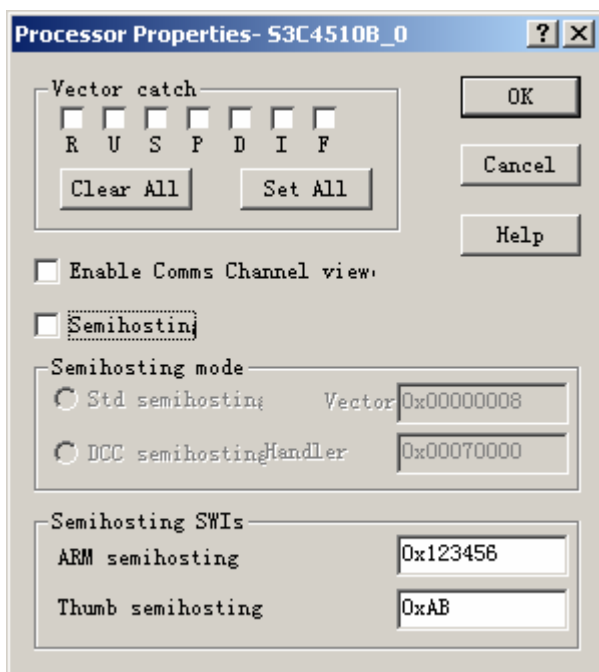
设置存储器的 Big-Endian 或 Little-Endian 模式



在菜单中打开 Option->Configure Processor...



清除所有选项，以免在发生 Exception 异常时，AXD 打断用户调试。



在用仿真器调试时，需要把程序下载到目标板的 RAM 中，但是因为 ARM CPU 的 Memory 地址等参数需要初始化，所以在调入调试程序前需要运行初始化文件。方法是在 Command Line Interface 窗口中运行批处理文件，例如：

```
obey d:\4510.conf
```

4510.conf 是 Samsung4510 评估板的初始化文件，文件的格式如下：

```
fillmem 0x3ff0000 0x3ff0000 0xe7ff0020    32
```

；给 0x3ff0000 地址写值 0xe7ff0020

```
fillmem 0x3ff3010 0x3ff3010 0xf5ffffa    32
```

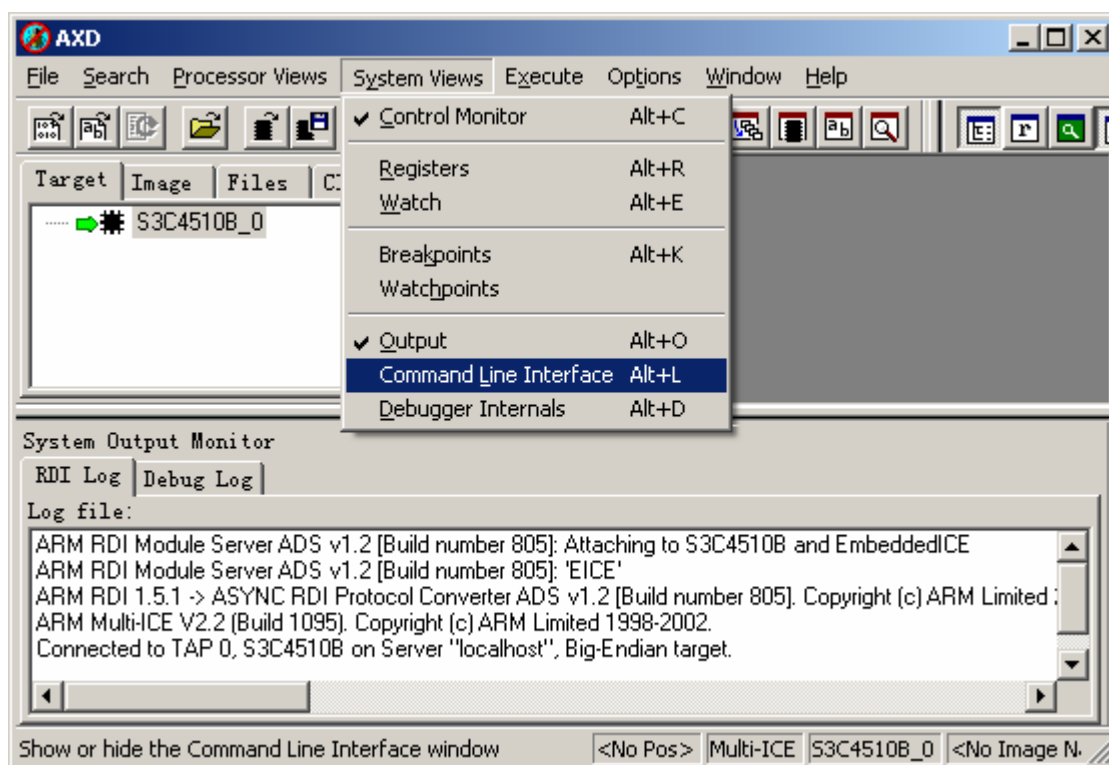
```
fillmem 0x3ff3014 0x3ff3014 0x02000060    32
```

```
fillmem 0x3ff302c 0x3ff302c 0x20040390    32
```

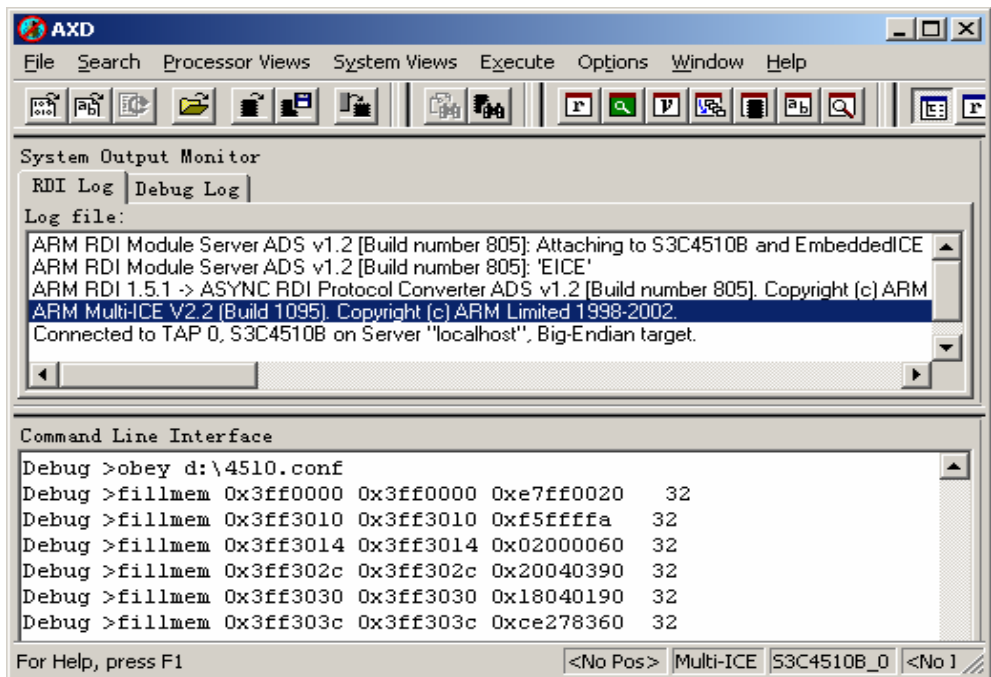
```
fillmem 0x3ff3030 0x3ff3030 0x18040190    32
```

```
fillmem 0x3ff303c 0x3ff303c 0xce278360    32
```

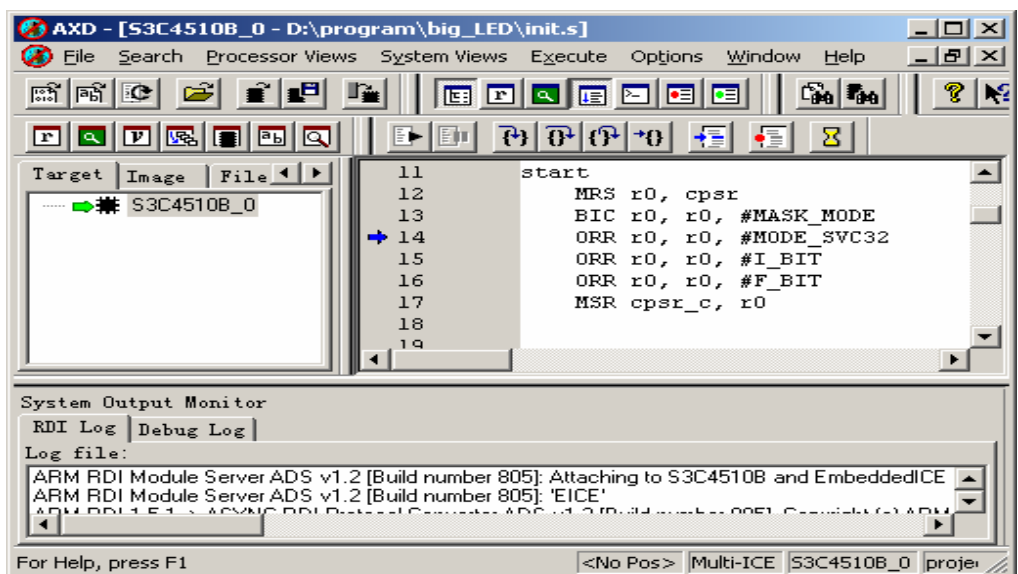
在菜单中运行 Option->Command Line Interface



在弹出的 Command Line Interface 窗口中运行初始化文件。

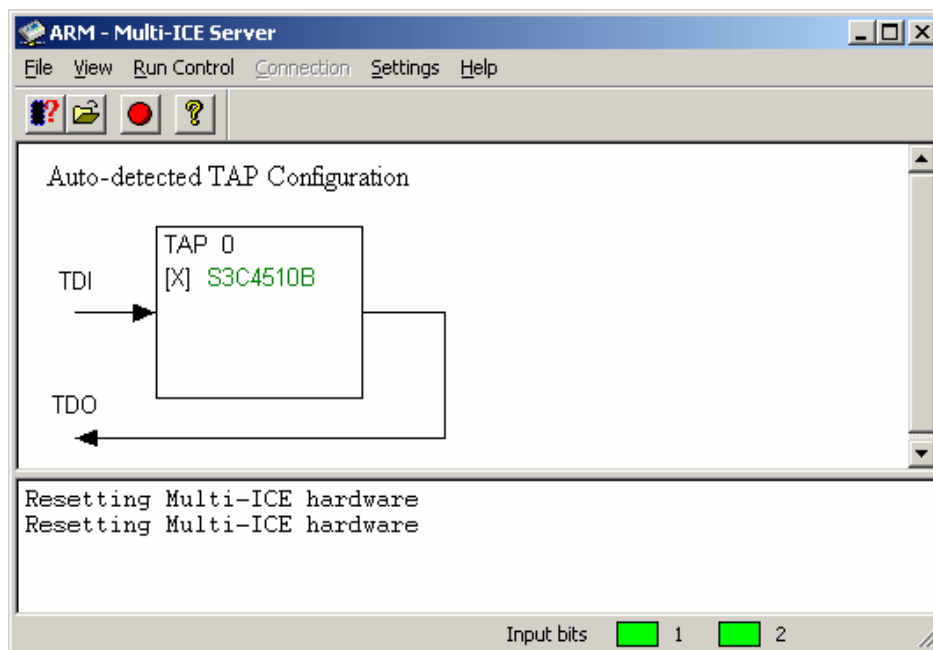


最后，打开 File->Load Image... 装载调试文件，进行调试

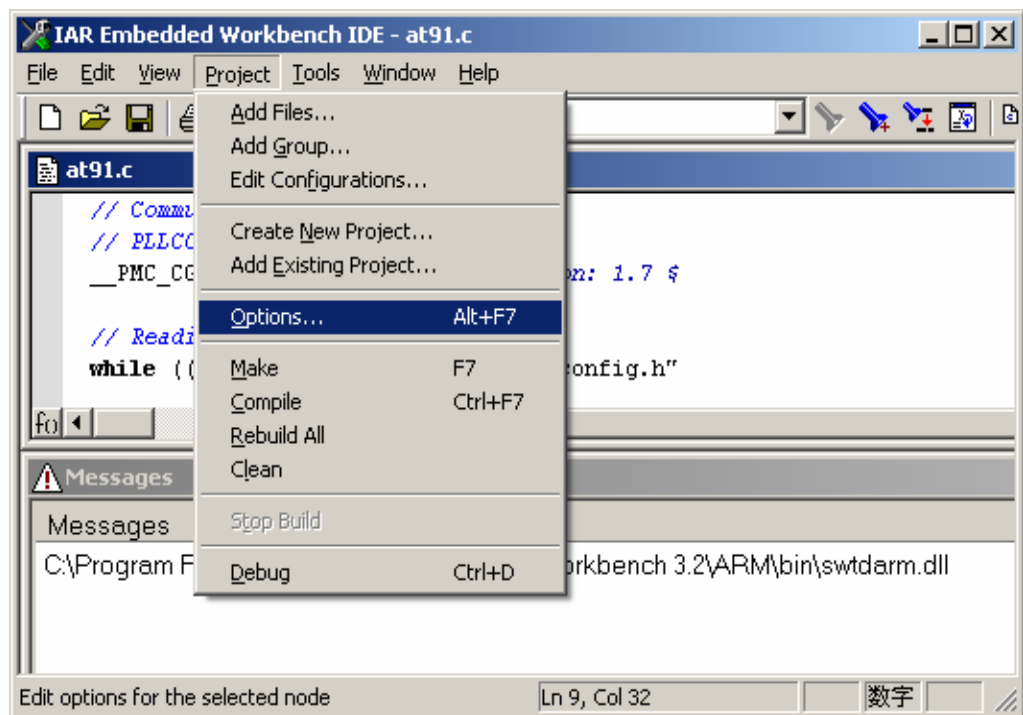


5.3 ARM_STAR 仿真器和 IAR EWARM3.4A 的连接

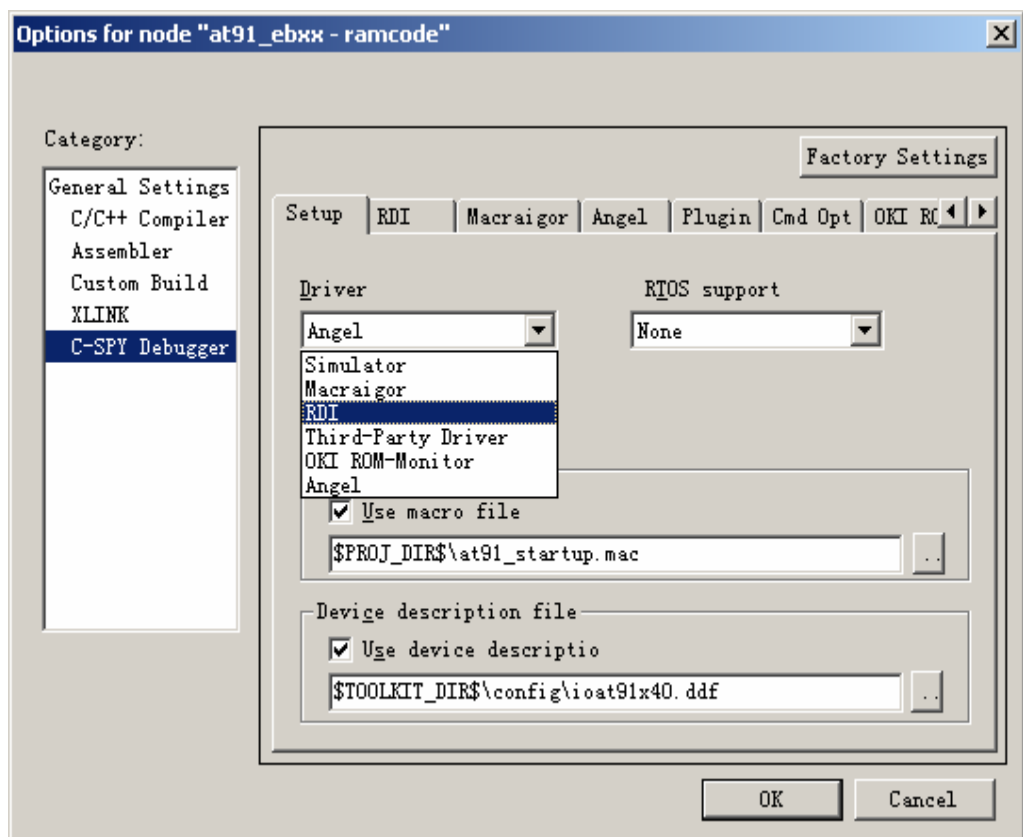
打开 Multi-ICE Server ,连接正确, 出现以下画面, 不要关闭。



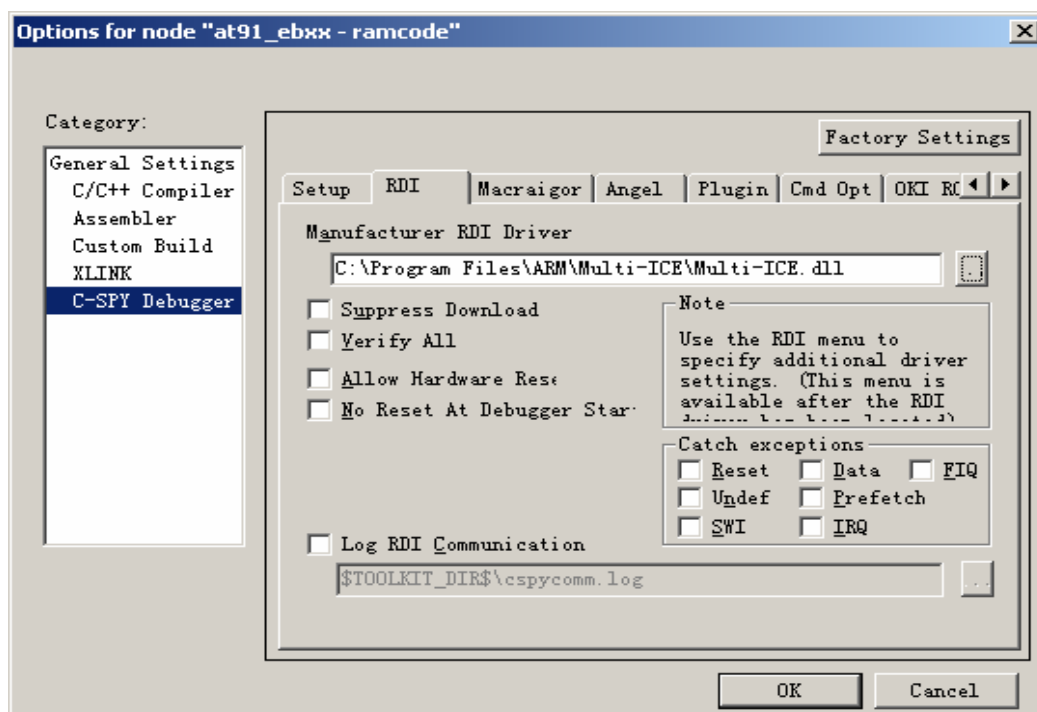
打开 IAR Embedded Workbench, 在调试器中打开一个项目, 运行 Project->Options...



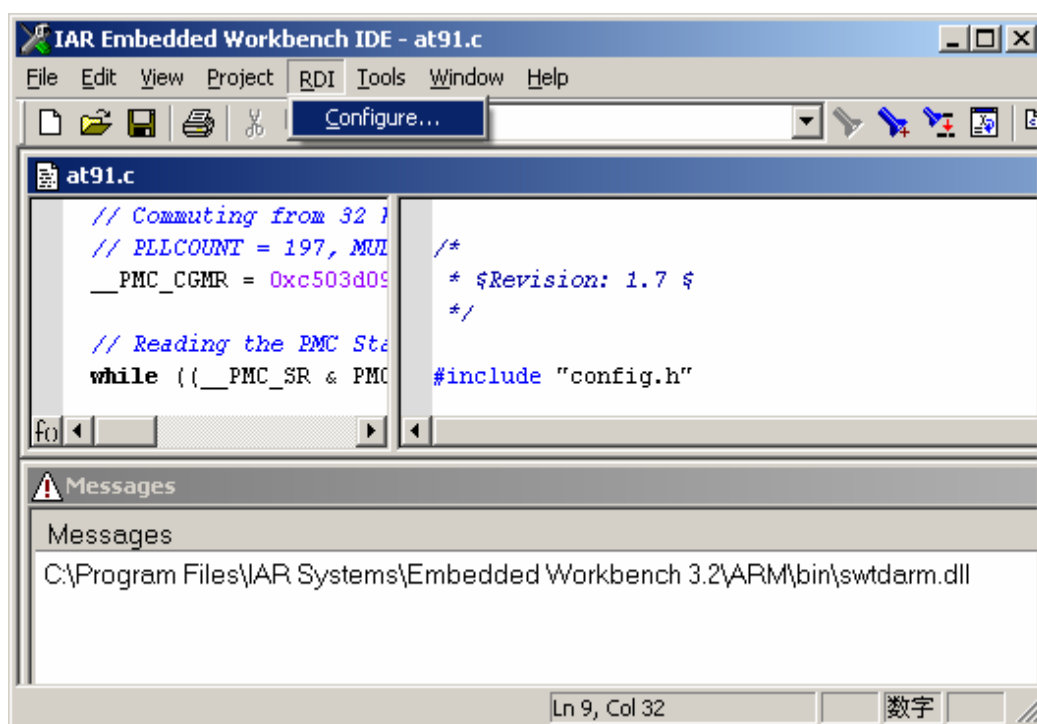
在 Setup 中的 Driver 选项中选择 RDI 协议。



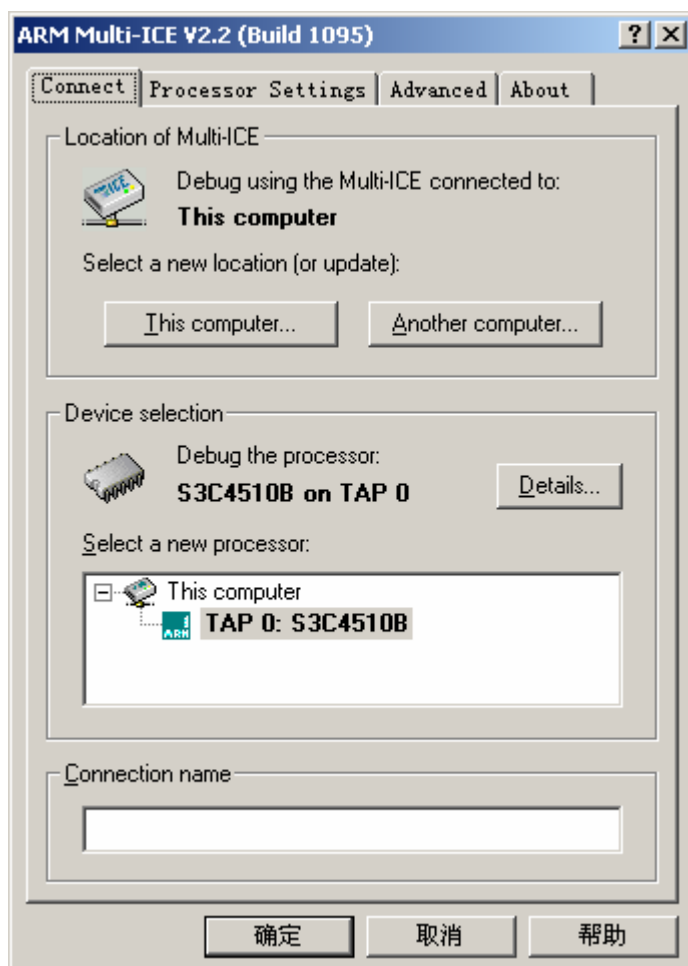
在 RDI 中选择 Multi-ICE.dll



打开 RDI->Configure...



跟先前讲述类似，设置具体目标系统参数



设置完成，点击确定。

6

ARM_STAR 仿真器烧写 Flash

Flash 烧录原理说明

通过仿真器烧写目标板 Flash 的原理很简单, 首先把要烧写到 Flash 中去的 *.bin 文件下载到目标板的 RAM 中去, 然后在 RAM 中运行一段小程序, 把 RAM 中的 *.bin 文件通过运行程序写到 Flash 中, 由于各种类型的 Flash 烧录和擦除算法不尽相同, 故具体运用时需要注意算法的改变, 本例中以 SST 公司的 39VF160 的烧写为模板进行介绍, 以下是 39VF160 的算法表格, 实践证明例子中的算法使用与绝大多数类型的 Flash.

Command Sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
	Addr ¹	Data ²	Addr ¹	Data ²	Addr ¹	Data ²	Addr ¹	Data ²	Addr ¹	Data ²	Addr ¹	Data ²
Word-Program	5555H	AAH	2AAAH	55H	5555H	A0H	WA ³	Data				
Sector-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	SA _x ⁴	30H
Block-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	BA _x ⁴	50H
Chip-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	5555H	10H
Software ID Entry ^{5,6}	5555H	AAH	2AAAH	55H	5555H	90H						
CFI Query Entry ⁵	5555H	AAH	2AAAH	55H	5555H	98H						
Software ID Exit ⁷ / CFI Exit	XXH	F0H										
Software ID Exit ⁷ / CFI Exit	5555H	AAH	2AAAH	55H	5555H	F0H						

例子中的参数:

调试器: ADS1.2

仿真器: ARM_Star

目标板: Samsung S3C4510B Flash:

SST 39VF160

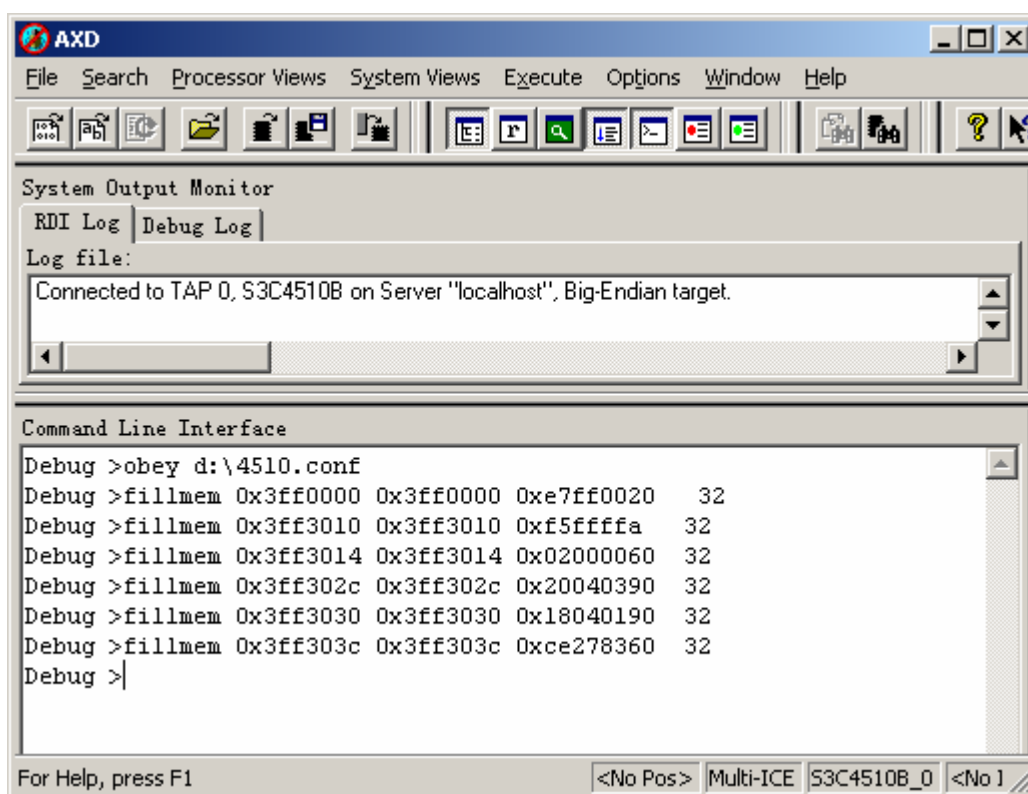
运行初始化文件后, 目标板的 Memory 分布:

16 位 Flash 的起始地址: 0x0

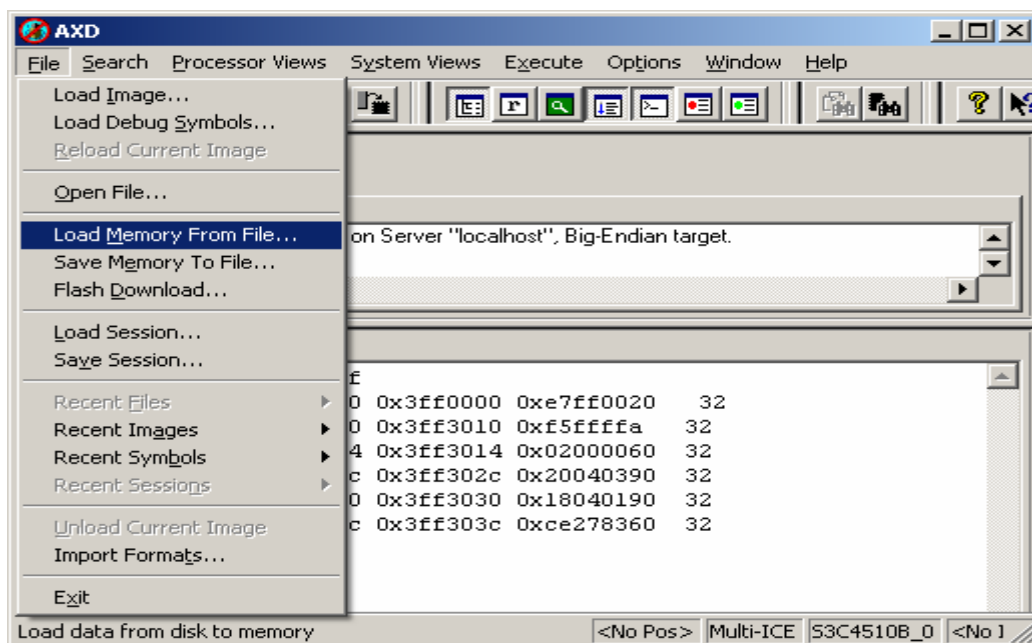
RAM 的起始地址: 0x1000000

6. 1 运行初始化文件

ADS1.2 调试环境中运行初始化文件, 使目标板的存储器位置达到上述要求, 当然, 在您的系统中这个地址可以任意设定。



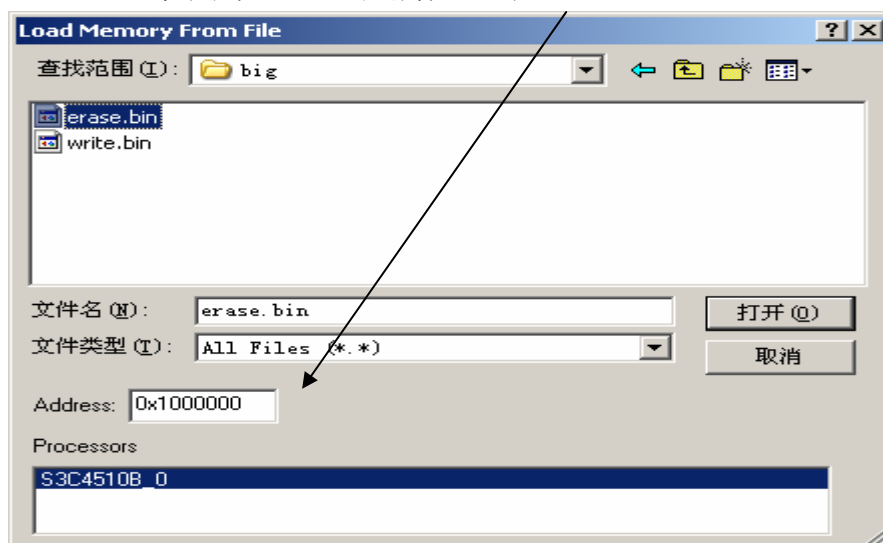
打开 File->Load Memory From File



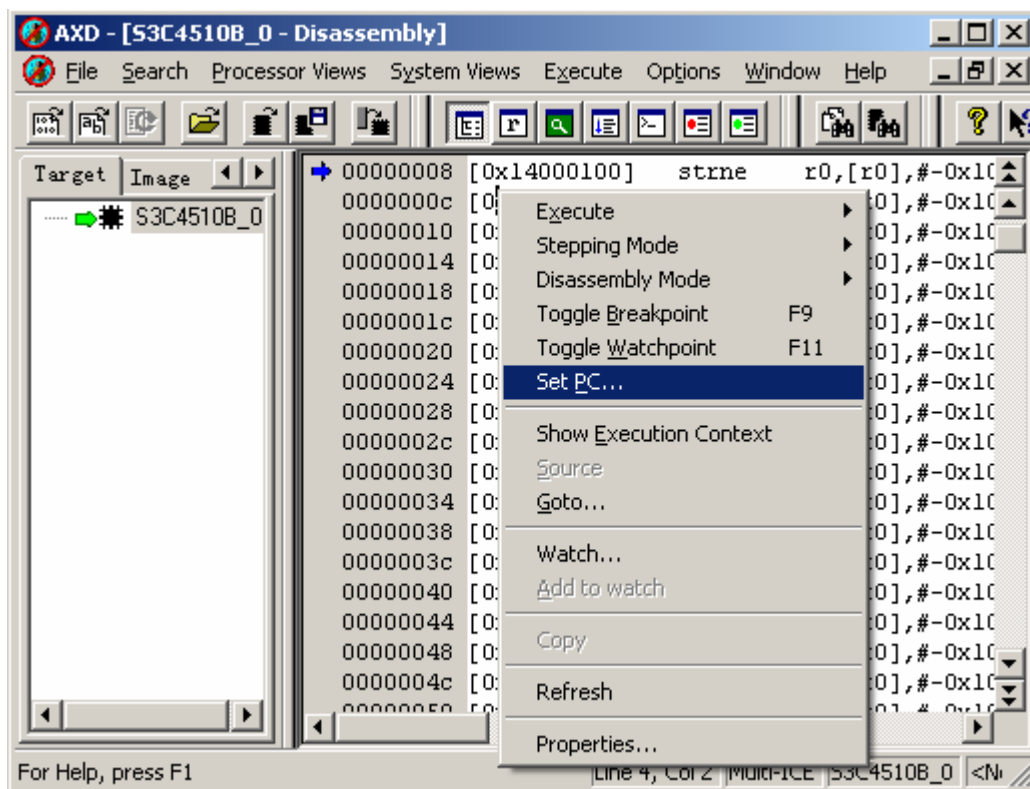
6. 1 运行 erase.bin 文件，擦除 Flash

选择 erase.bin 文件，下载到目标板中。

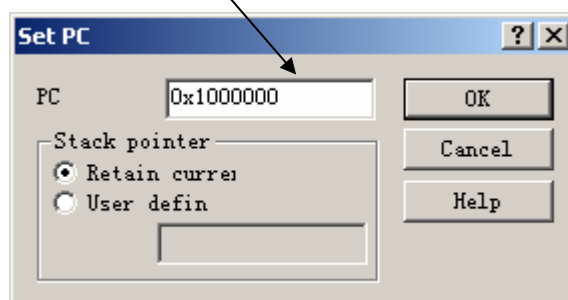
注意：请在 Address: 框中填上您的目标板的 RAM 起始地址。本例中 RAM 起始地址是：0x1000000

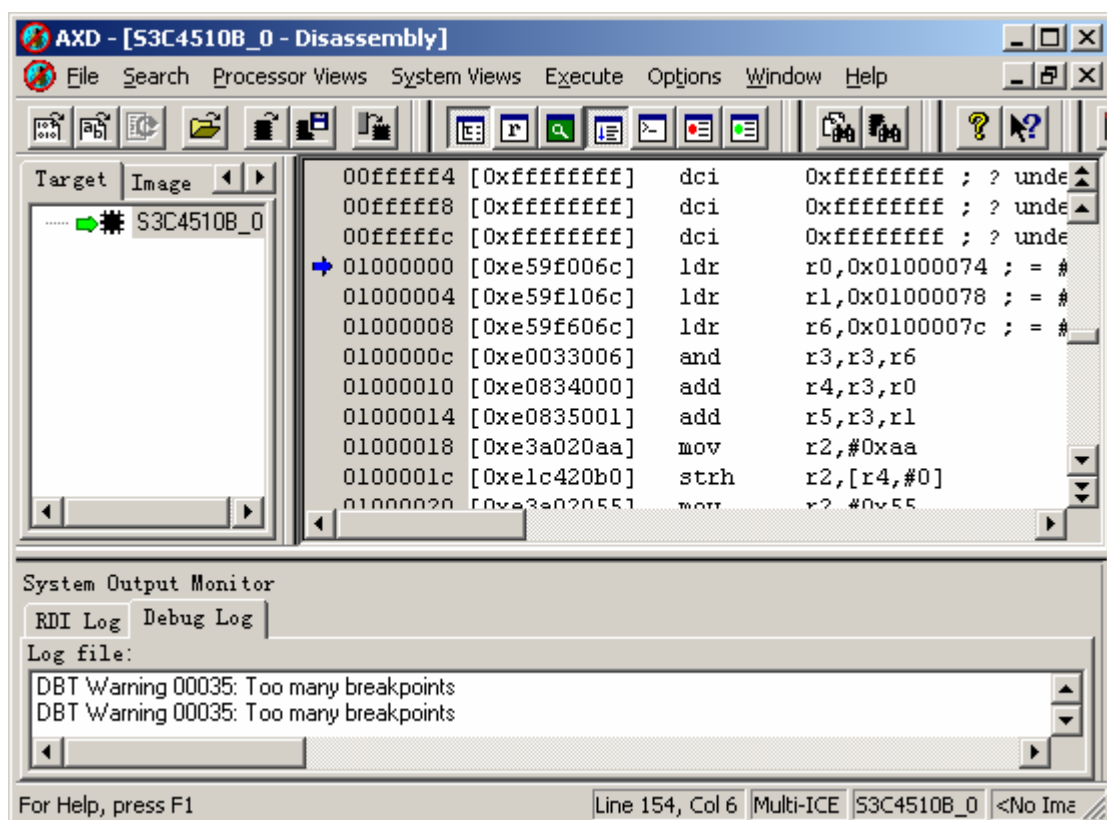


下载完成后，在 Disassembly 窗口中点鼠标右键，选择 Set PC...

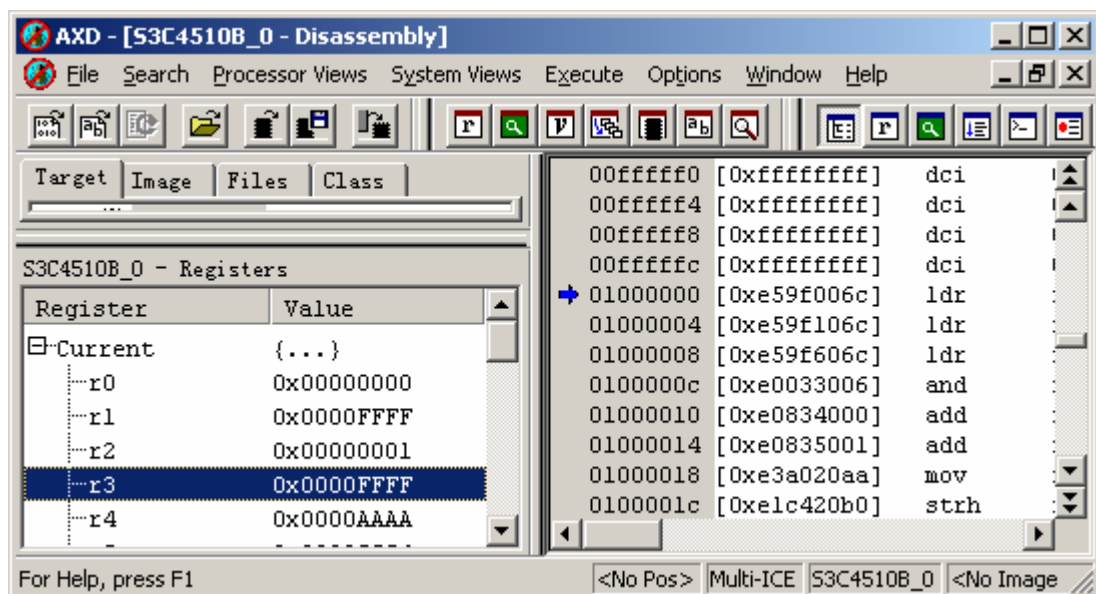


把 PC 值设成目标板的 RAM 起始地址

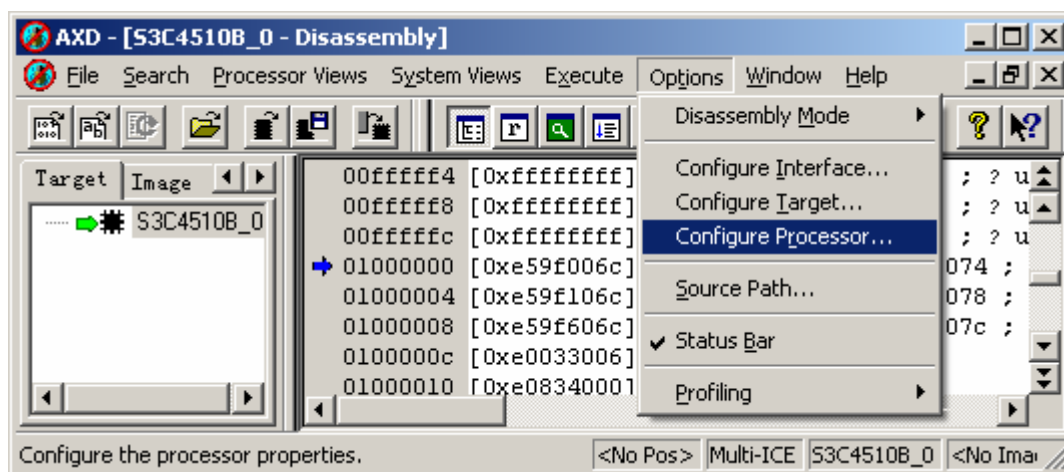




然后，在 Register 窗口中修改 R3 寄存器的值，把用户 Flash 的起始地址填入到 R3 中，本例中 Flash 在 0x0。



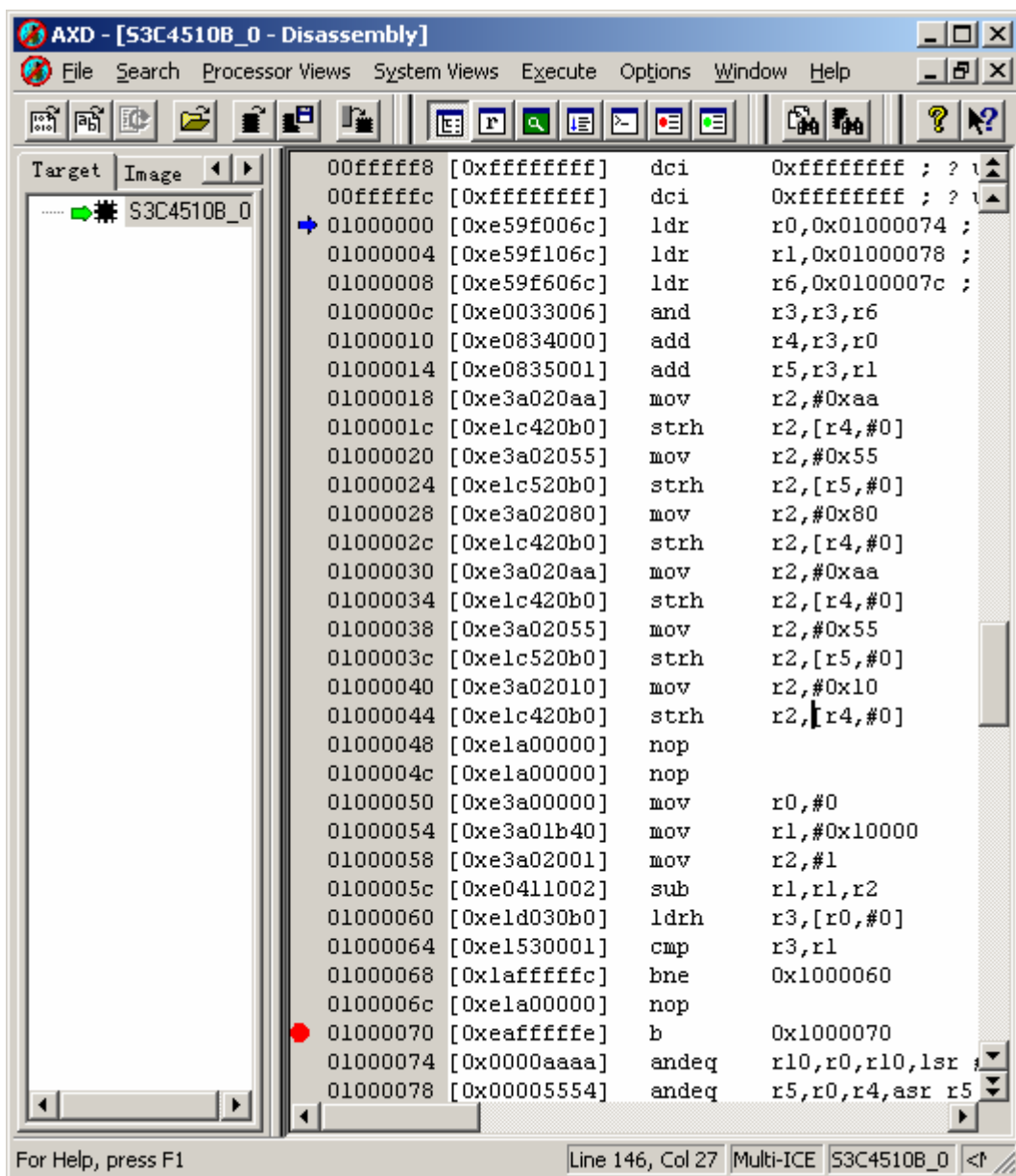
打开 Option->Configure Processor，清除所有 Exception 选项



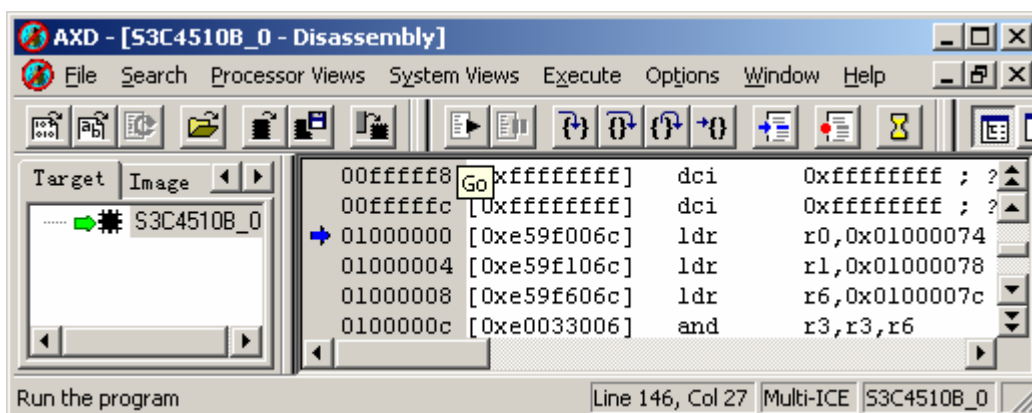
清除所有 Exception 选项



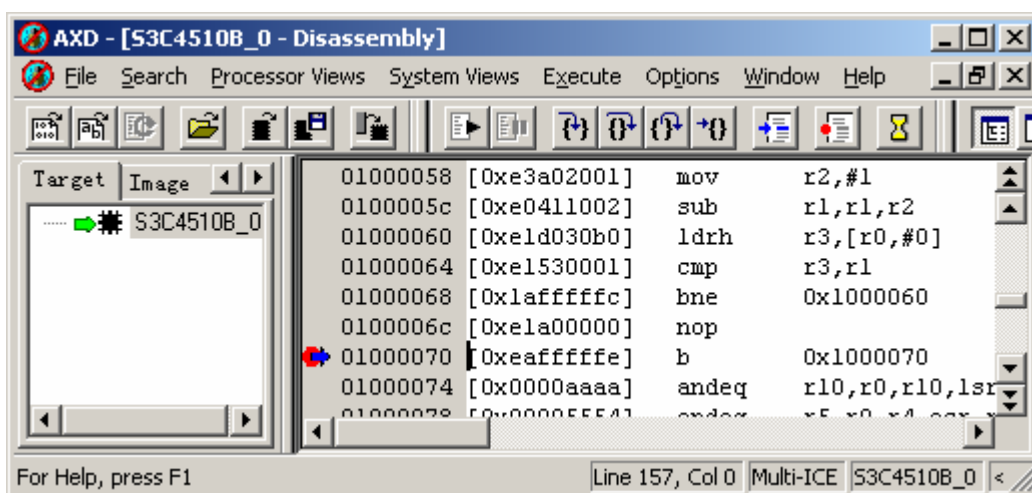
然后，在程序循环的位置设置断点，位置大约在程序起始地址+0x70 的地方，设断点方法：让光标在想设断点的地方闪烁，然后按 F9 实现。



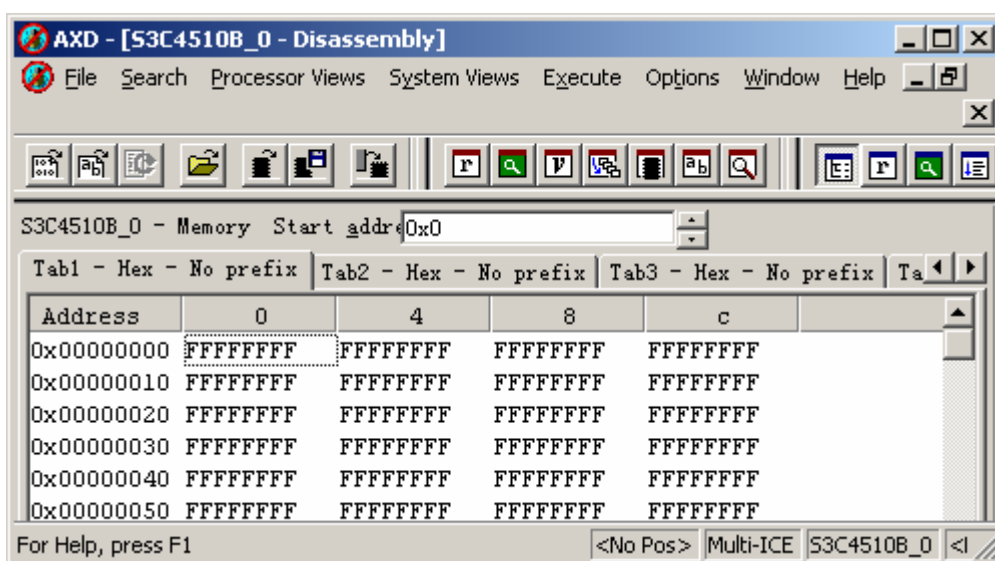
最后，运行程序，点击 GO



程序停下来

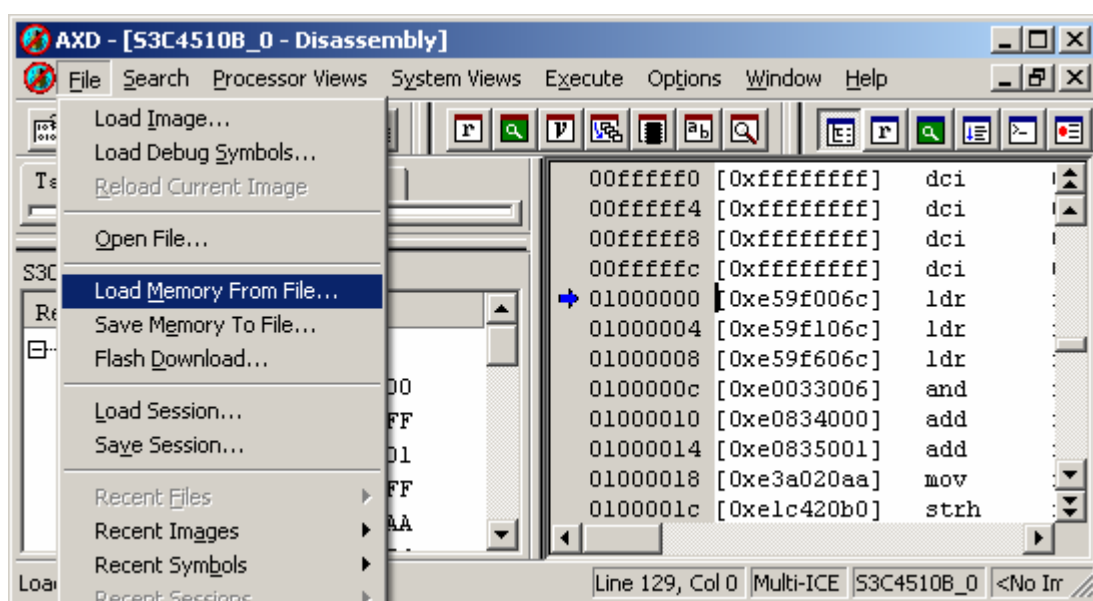


察看 Memory 窗口中 0x0 地址，Flash 中的内容已经被擦除。

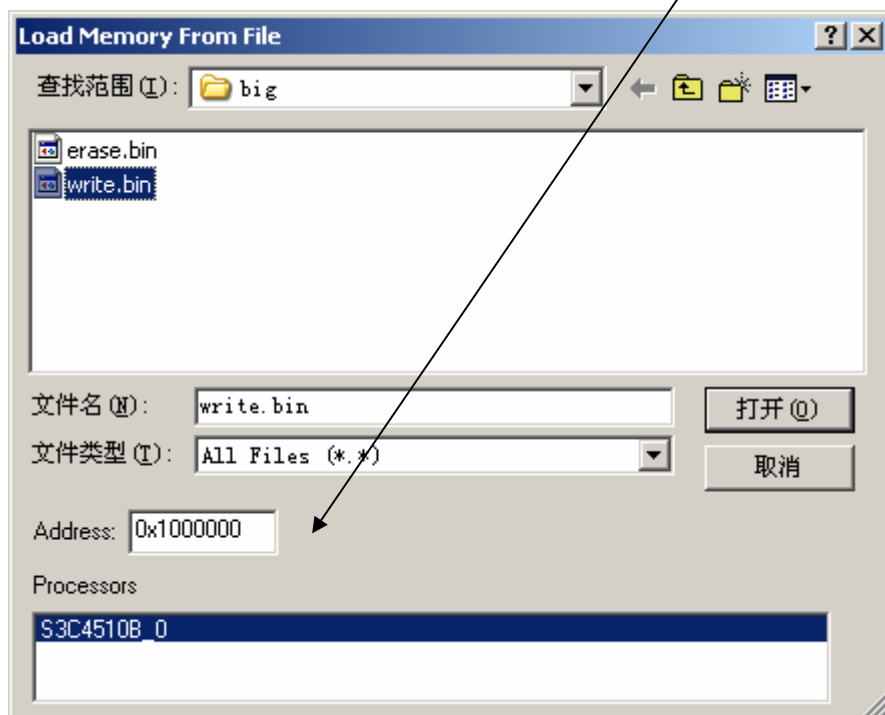


6. 3 运行 erase.bin 文件烧写 Flash

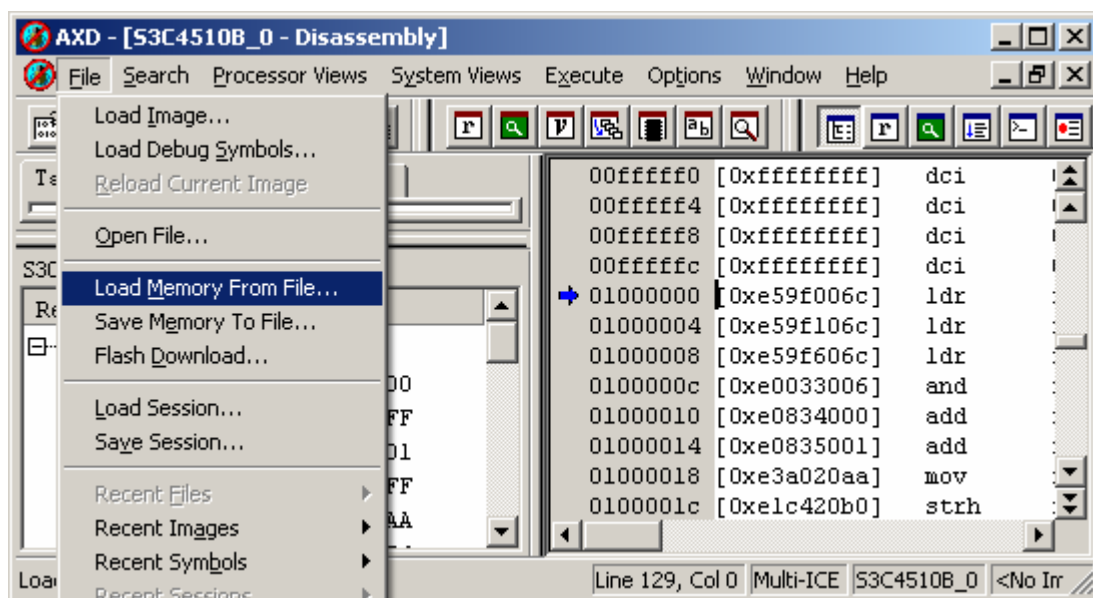
打开 File-〉 Load Memory From File



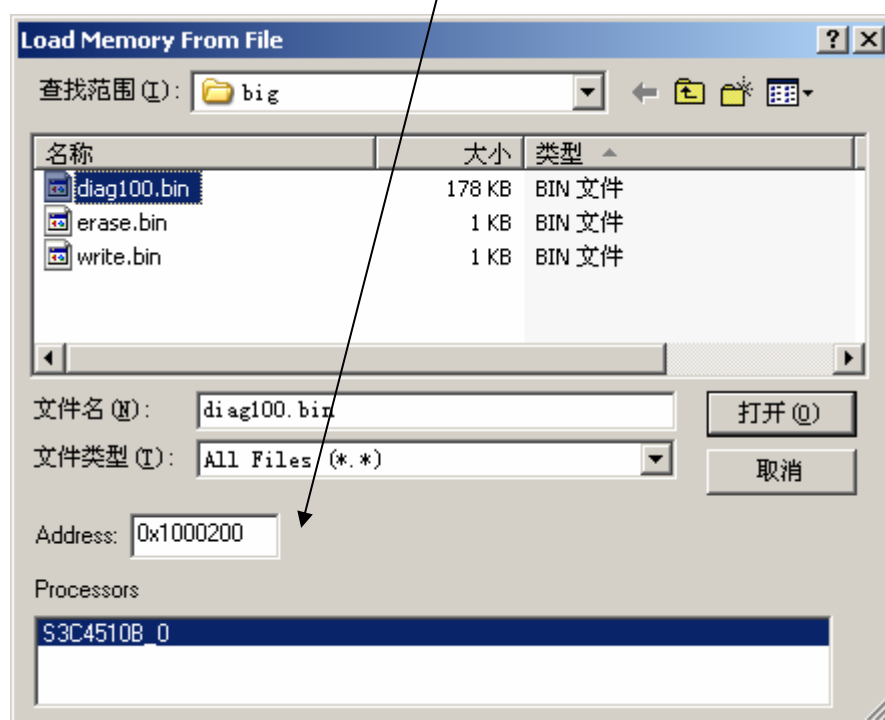
选择 Write.bin 文件，下载地址选 RAM 的起始地址



再打开 File-> Load Memory From File



选择您要烧写到 Flash 中去的文件,地址选择 RAM 起始地址+ 0x200, 本例中是 0x1000200。

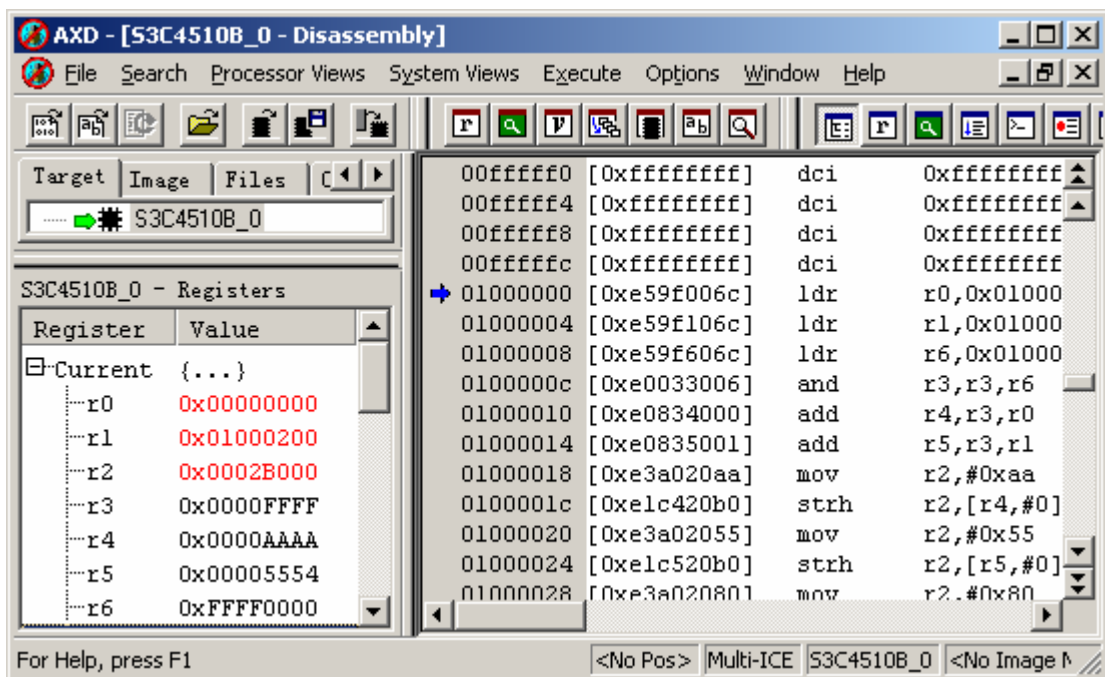


打开 Option->Configure Processor，清除所有 Exception 选项

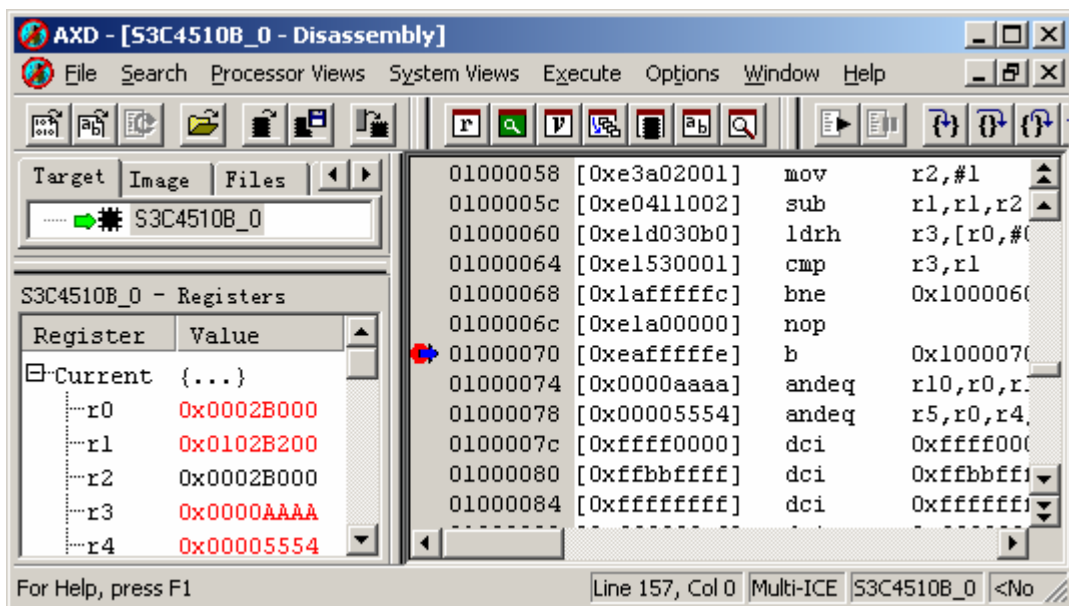


然后，在寄存器窗口中设置几个参数（注意，用十六进制数表示）

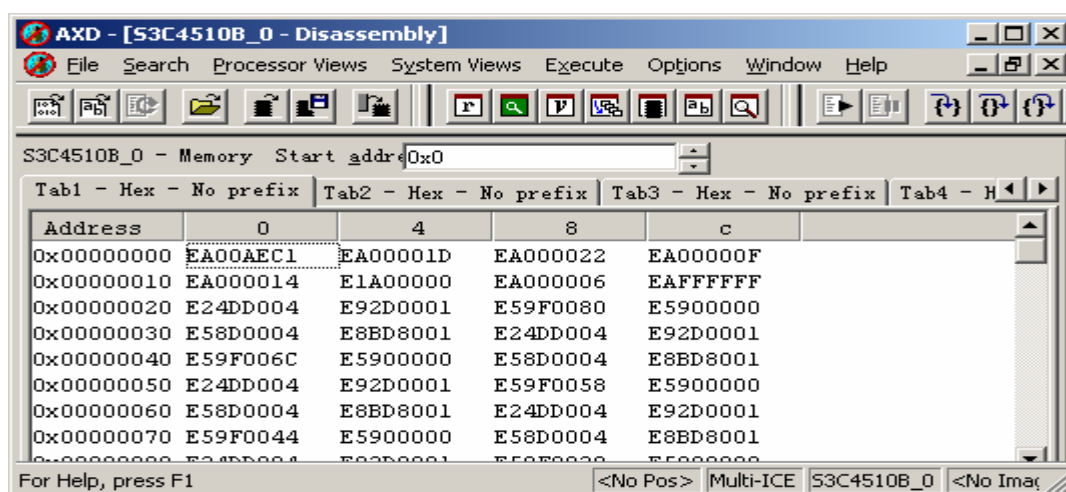
1. Flash 的起始地址，修改 R0
2. 要烧写到 Flash 中去的文件在 RAM 中的起始位置，修改 R1
3. 您要烧写的文件的长度，修改 R2



在程序后面+0x70 的地方设断点，运行



完成后，查看 Memory 窗口中 0x0 地址 Flash 中的值，程序已经烧写成功，重新上电，程序就从 Flash 中运行。



6. 4 于 Flash 烧录的几点注意事项

1. erase.bin 和 write.bin 文件分为大模式和小模式，即 big Endian 和 little Endian，使用时请注意区分
2. 下载任何程序前，请确定目标板 RAM 和 Flash 的地址是否正确，否则，需要运行初始化文件进行定位。
3. ADS 编译后得到*.axf 文件，转换到*.bin 文件请用 Flash 目录中的 elf2hex.bat 文件转换，方法是：
 - 3.1 把*.axf 文件拷贝到 falsh 目录下
 - 3.2 修改 elf2hex.bat 文件，把文件名改写成您的，如：例子中的文件名是 test.axf.

fromelf test.axf -bin -nodebug -nolinkview -output test.bin

- 3.3 执行 elf2hex.bat, bin 文件就会出现在项目目录下。

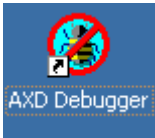
7

在 ADS1.2 下用 ARM_STAR 仿真器 快速开始指南

概述

本章以 ARM 公司的集成调试环境 ADS1.2 为例，介绍如何快速开始程序的编辑，编译和调试。

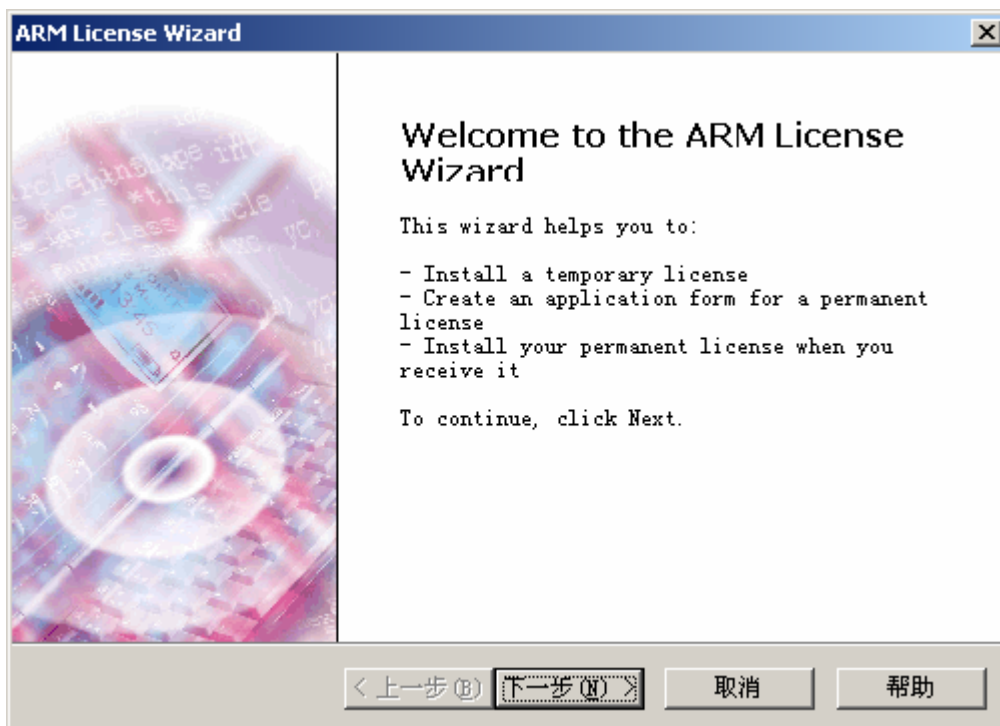
ADS1.2 是英国 ARM 公司开发的一款集源程序编辑，编译，调试于一体的 ARM 开发工具，主要分两部分：

- 1, 编辑，编译器  CodeWarrior For ARM Developer Suite
- 2, 调试器  AXD Debugger

7. 1 ADS1.2 的安装

在光盘中找到 ADS 目录，点击安装软件 Setup.exe，一切按照默认设置安装即可。

当出现如下窗口时，请点击 Cancel 取消。



然后，把向 ARM 公司购买的 license.dat 拷贝到安装目录下的 license 目录中

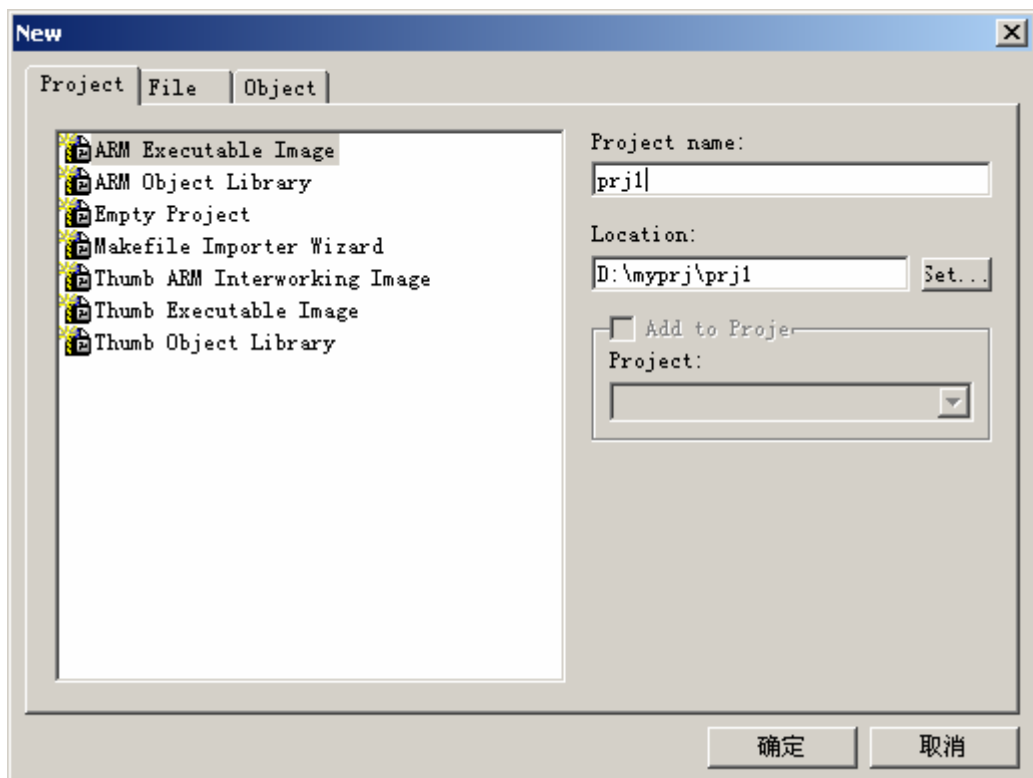
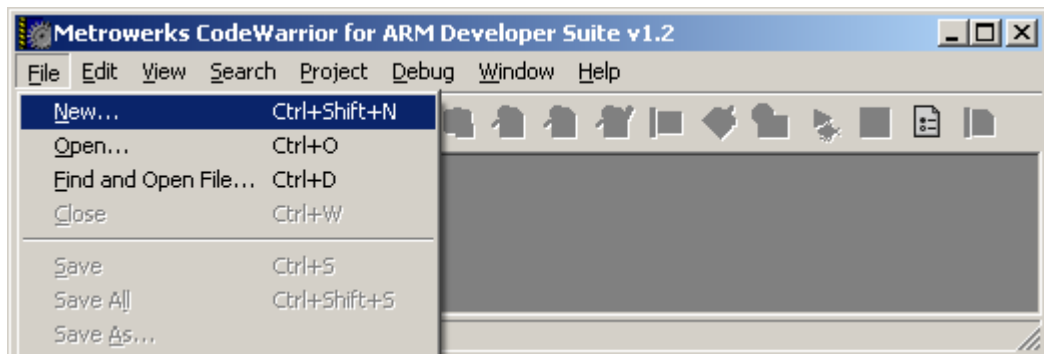


至此，安装完成。

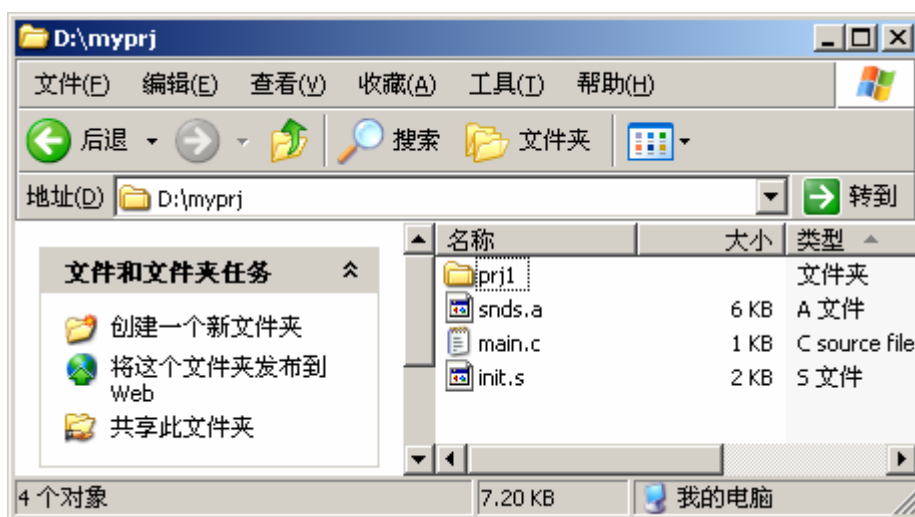
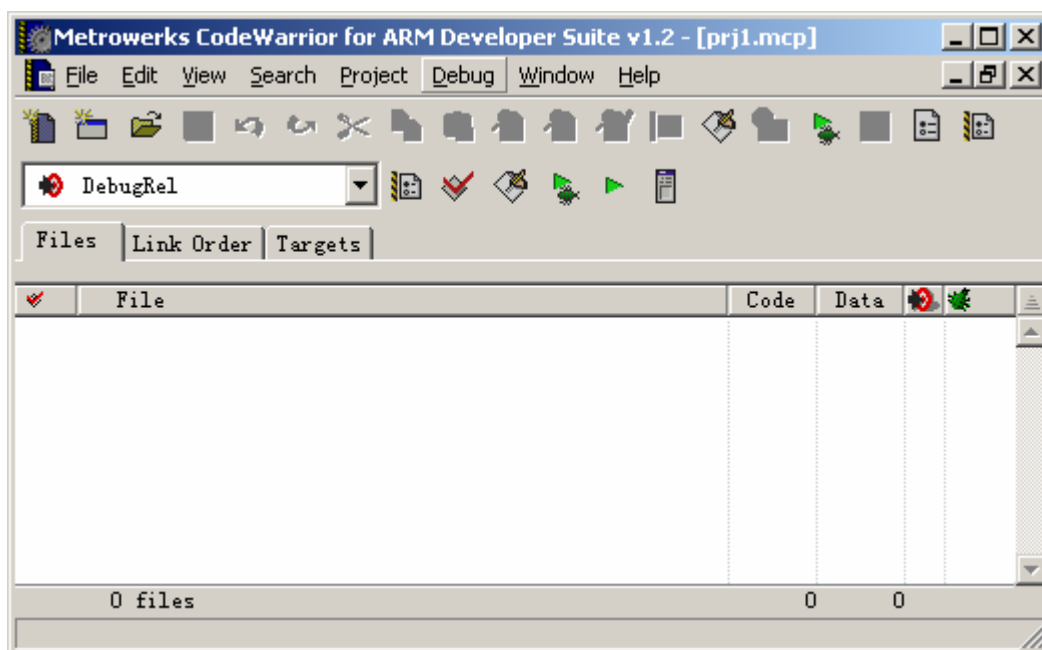
7. 2 CodeWarrior For ARM Developer Suite 编辑，编译项目



打开 CodeWarrior For ARM Developer Suite 调试环境，新建项目



出现如下界面。

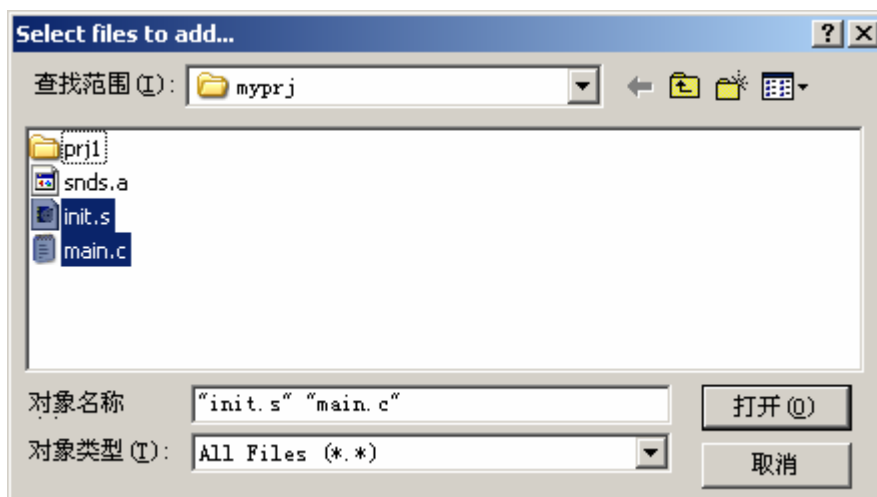
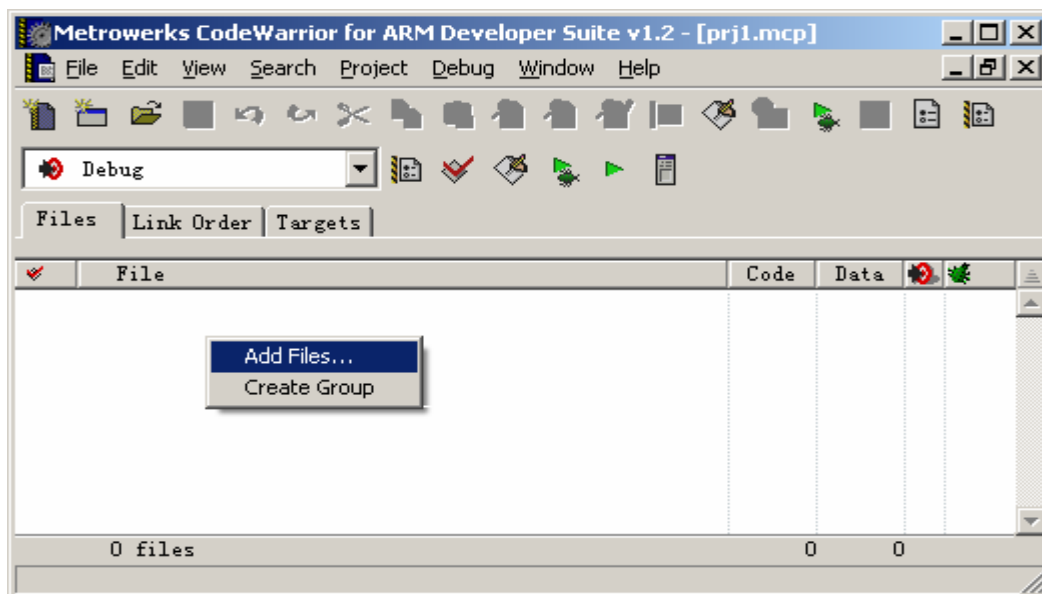


在新建项目的目录下，编辑源程序文件，例子中包含了三个文件：

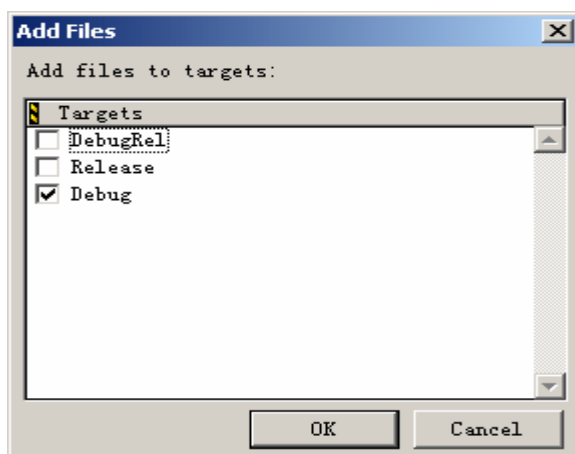
main.c	C 源程序
init.s	汇编语言源程序
snds.a	定义文件

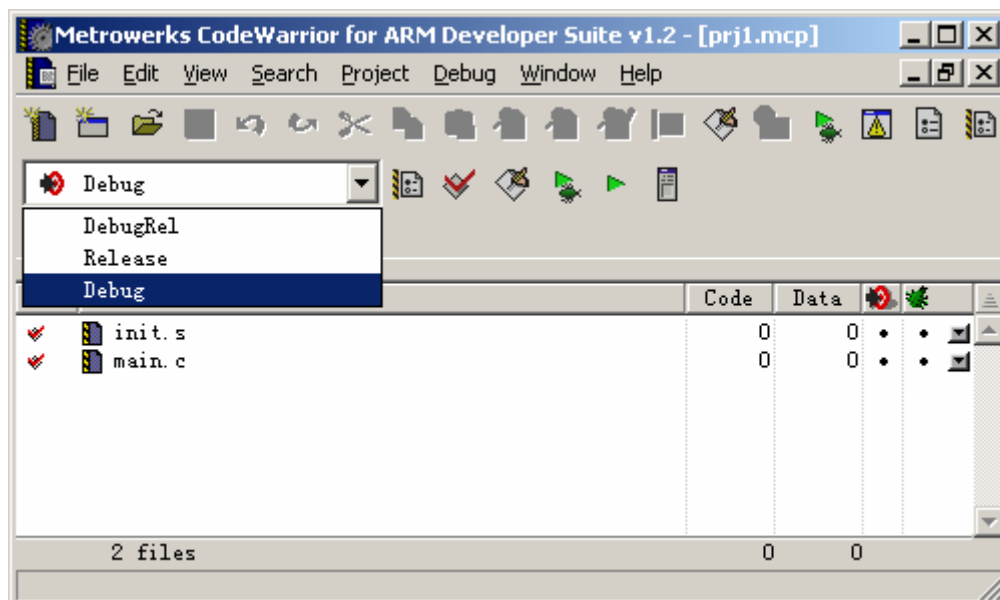
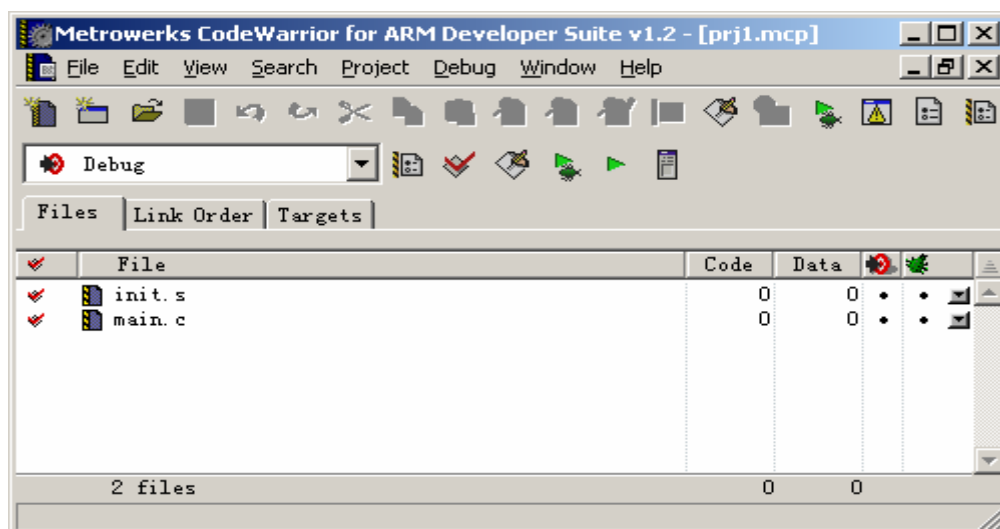
把 init.s 和 main.c 源文件加入到项

目中

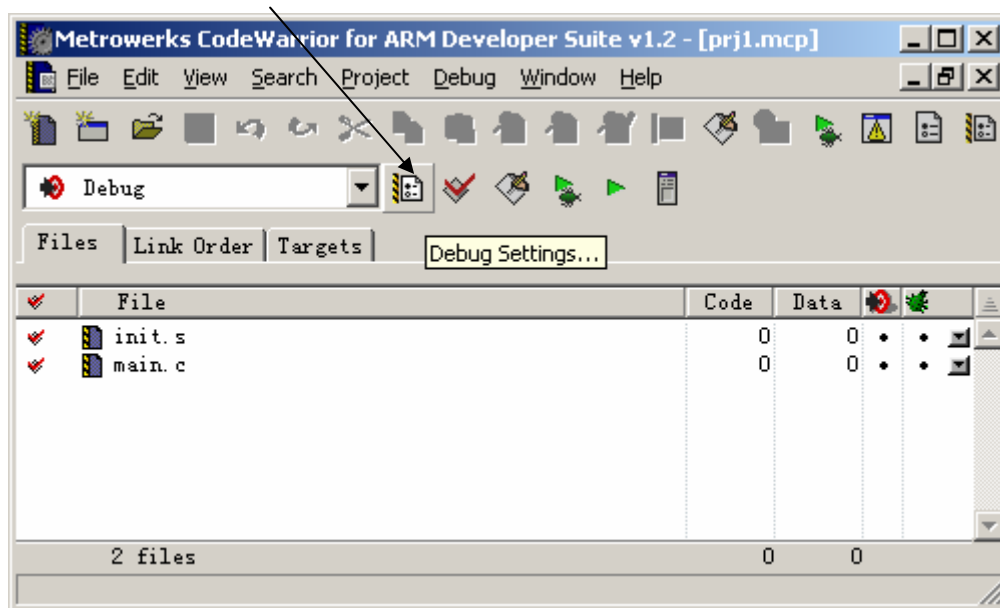


只选择 Debug 调试模式

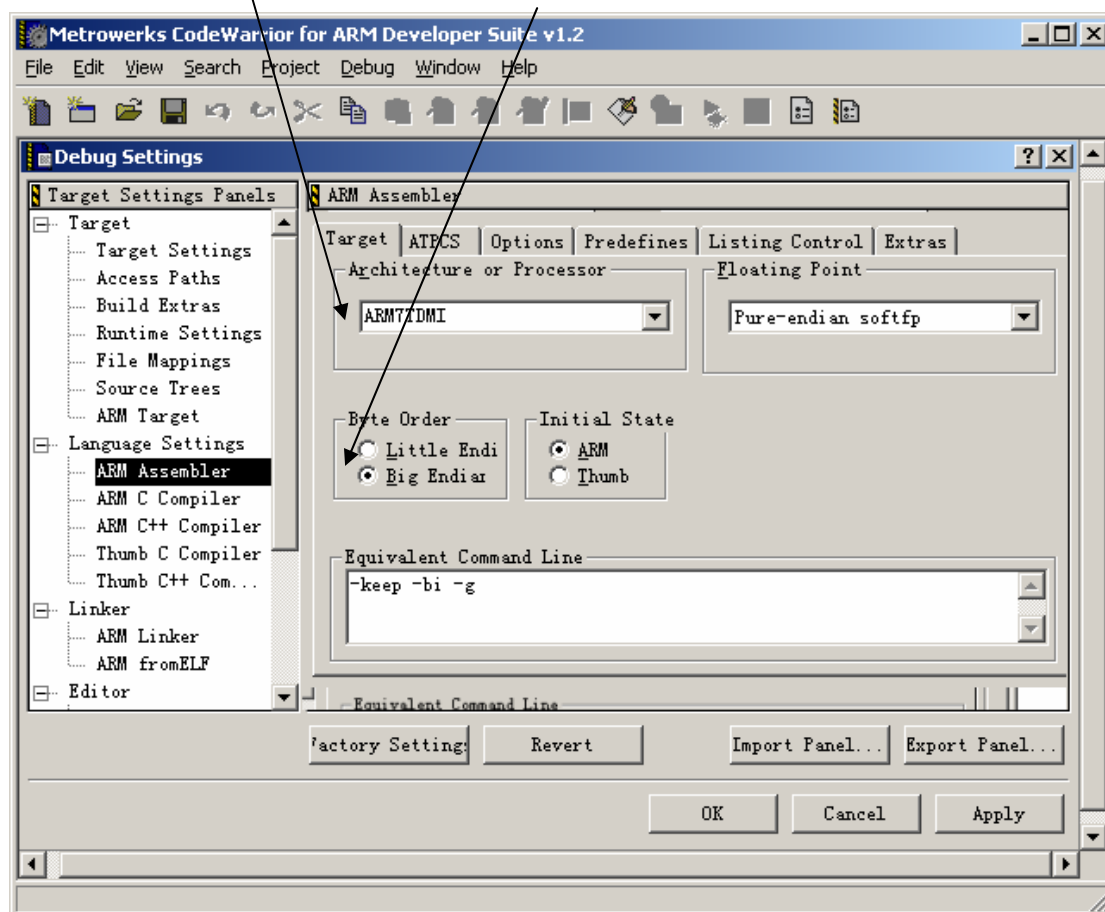




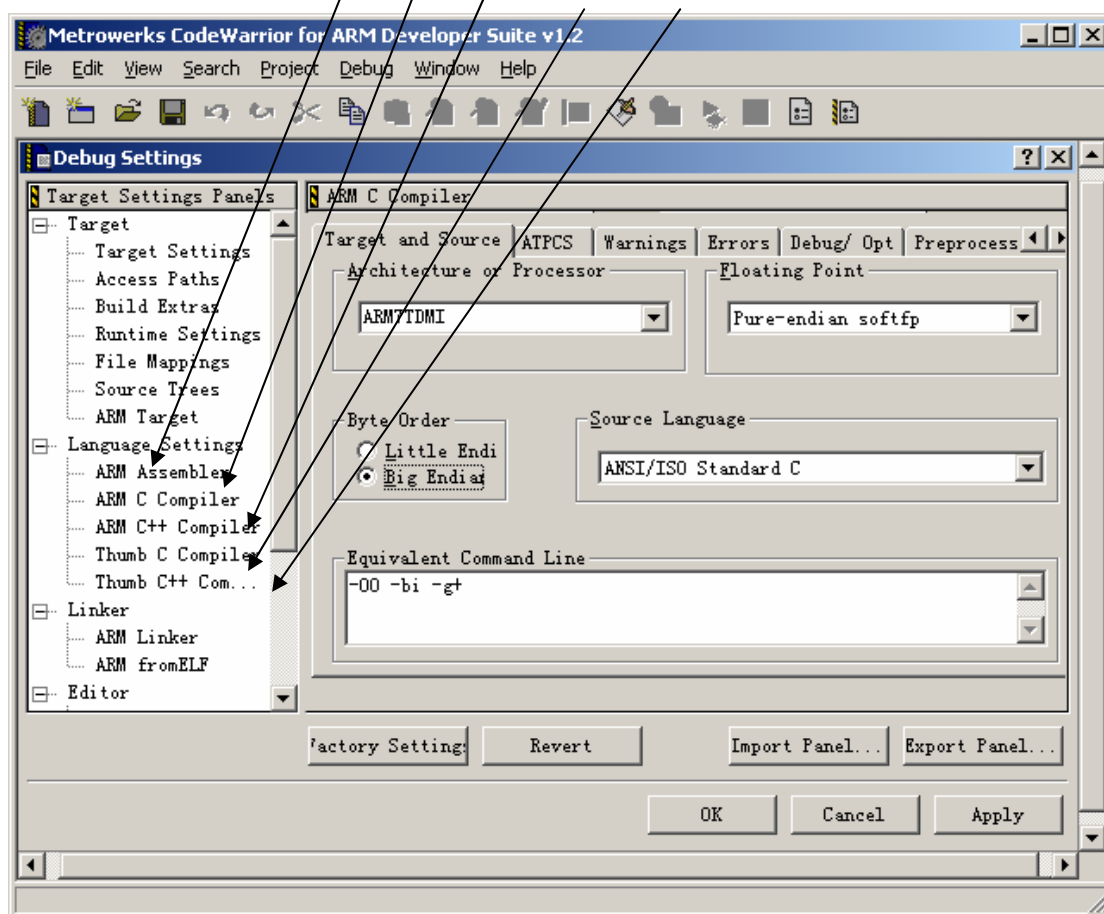
点击编译设置按钮



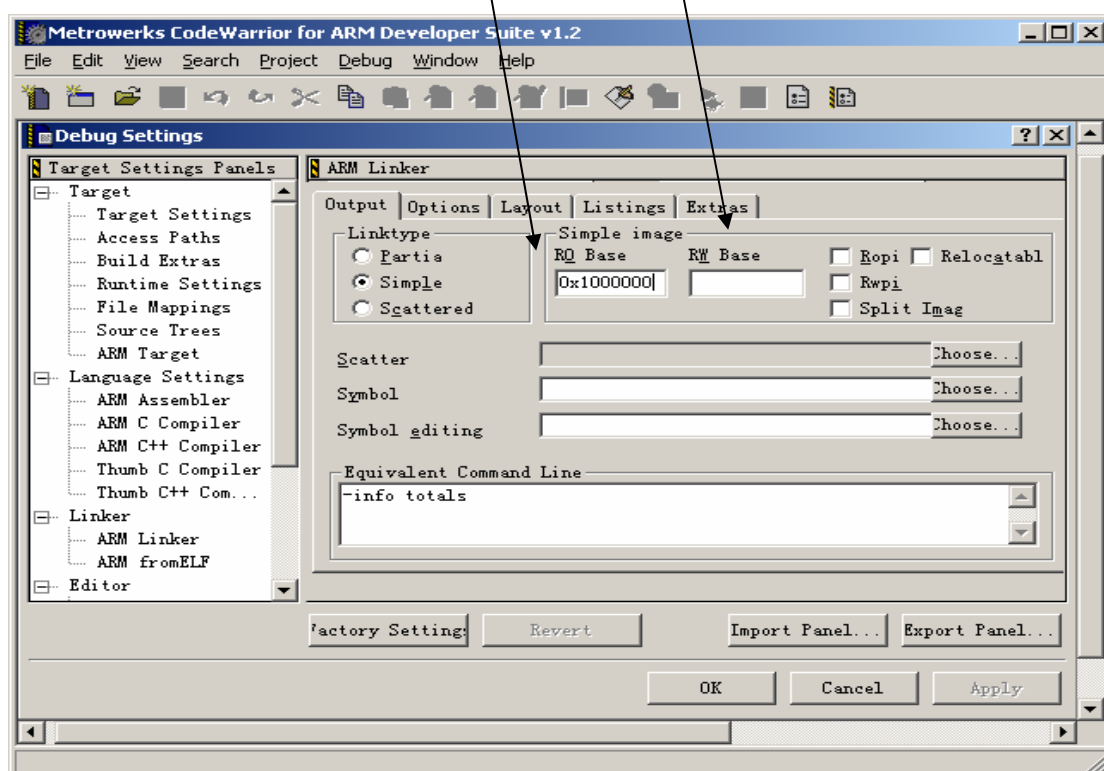
主要设置 CPU 类型和存储器大小端模式



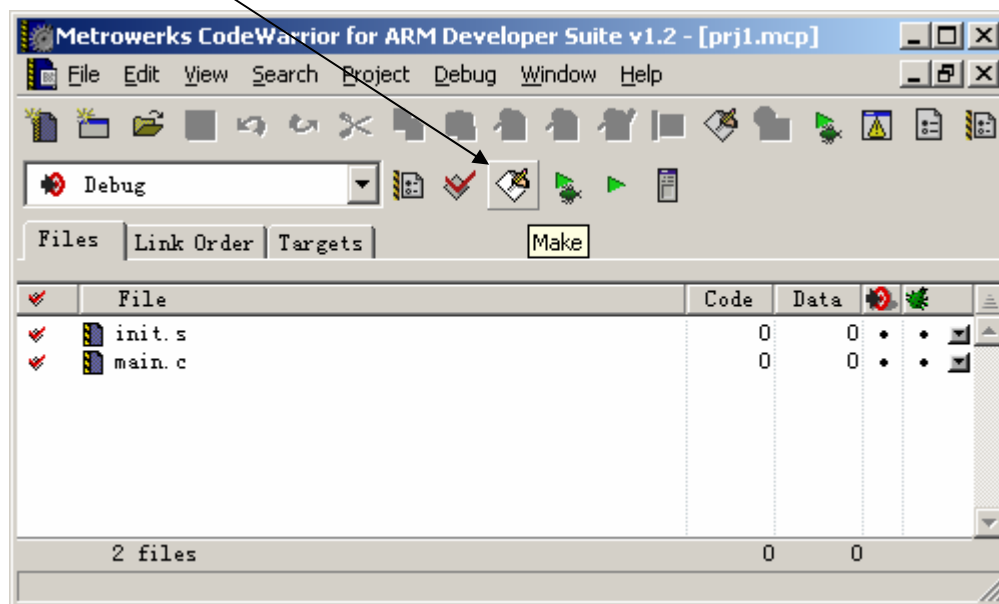
确定不同语言编译器的上述设置相同



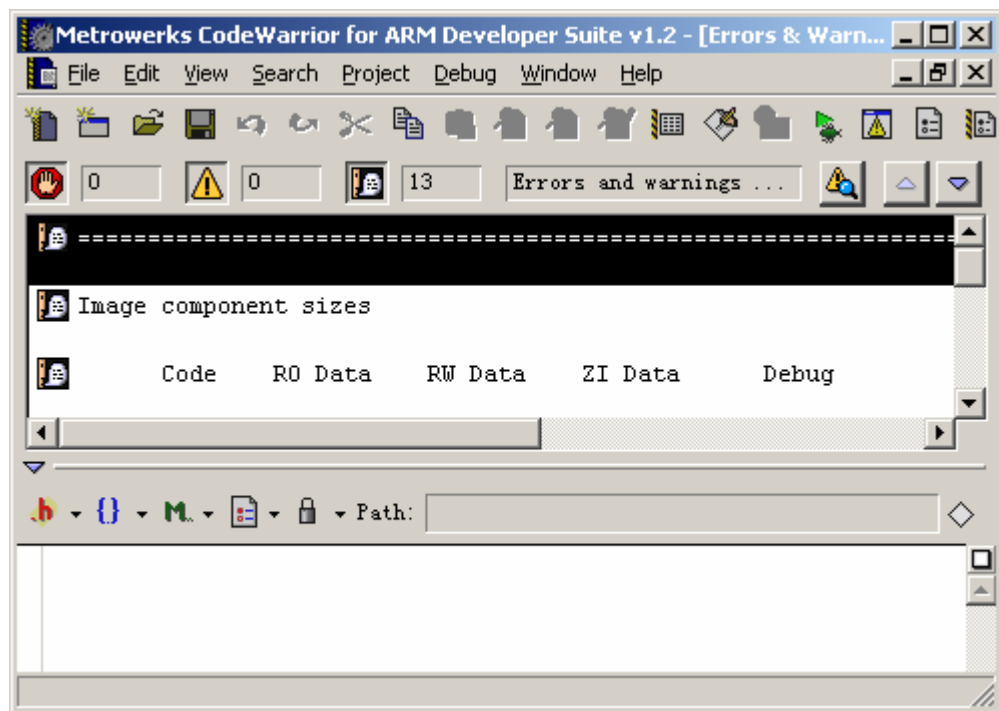
在 ARM Linker 页中选择程序代码和数据代码地址



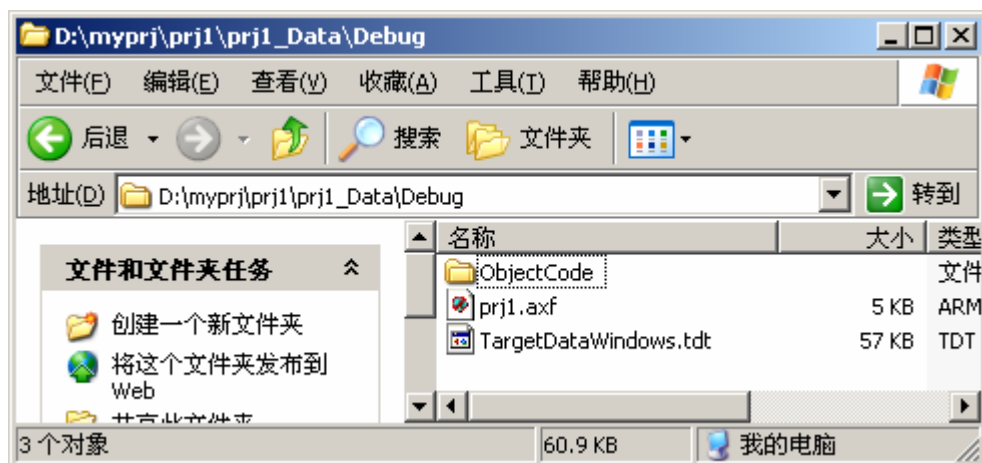
点击编译图标进行编译



编译结果正确。



编译生成 prj1.axf 调试文件

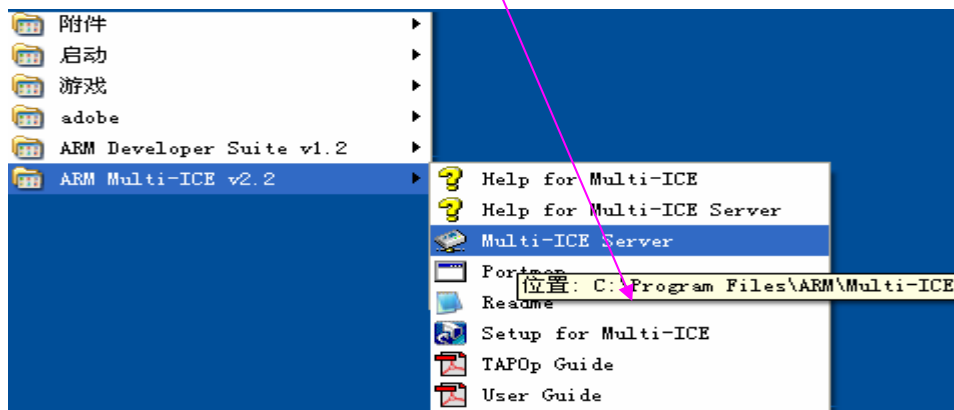


7. 3 源代码调试

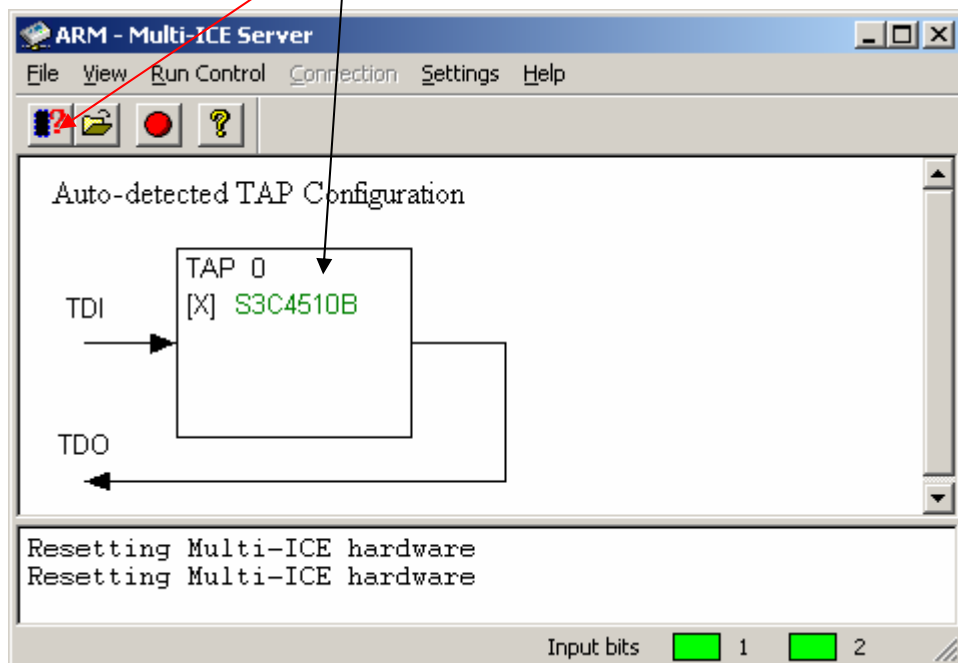
参考第五章中 ARM_STAR 仿真器和 ADS1.2 的连接，正确连接 ADS 和 ARM_Star 仿真器。

7. 3. 1 启动 Multi-ICE Server

在 Windows 开始菜单中运行 Multi-ICE Server



确认目标板电源开关打开，点击自动配置图标，如果正常，就会出现下图，说明 MultiICE 已经连接到目标板



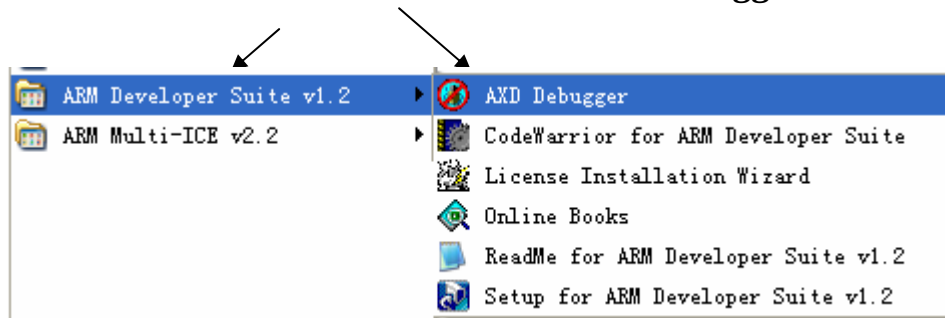
如果，没有成功，

请：

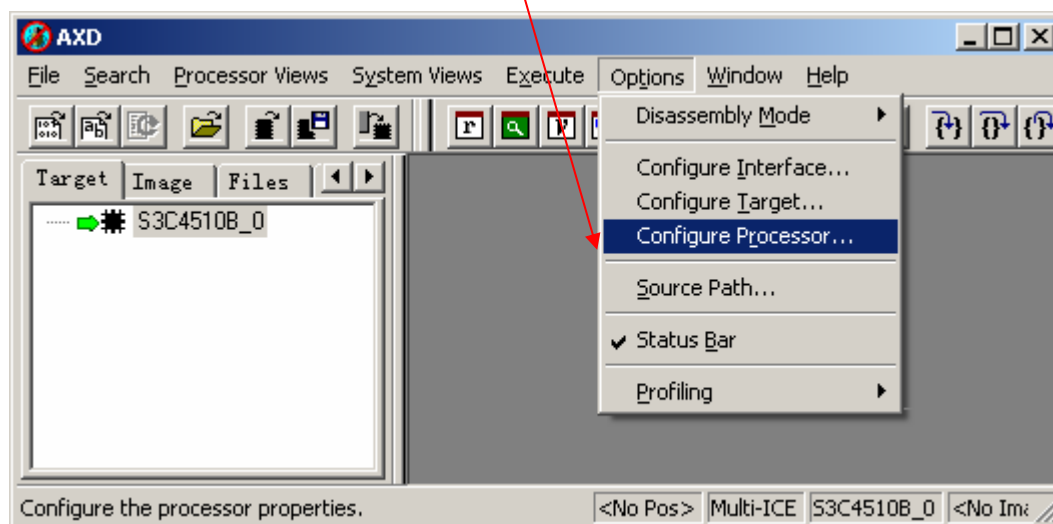
1. 首先检查以上设置是否正确，
2. 然后，检查并口模式
3. 重新启动计算机，在 BIOS 中设置正确的并口模式，
一般正确的是 ECP 或 双向模式，如果碰到问题可以多试几种模式，直到正确为止。
4. 用户的 CPU 需要运行配置文件，才能实现连接。

完成 **MultiICE Server** 启动，打开 **ADS Debugger** 调试环境

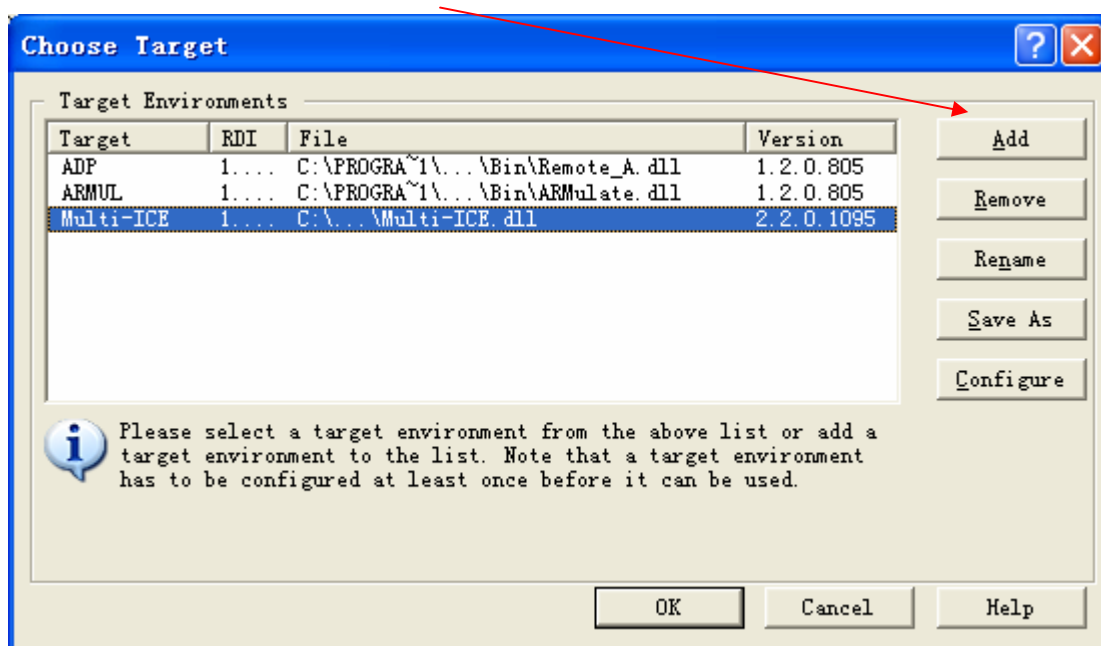
在 Windows 开始菜单中运行 **ADS Debugger**



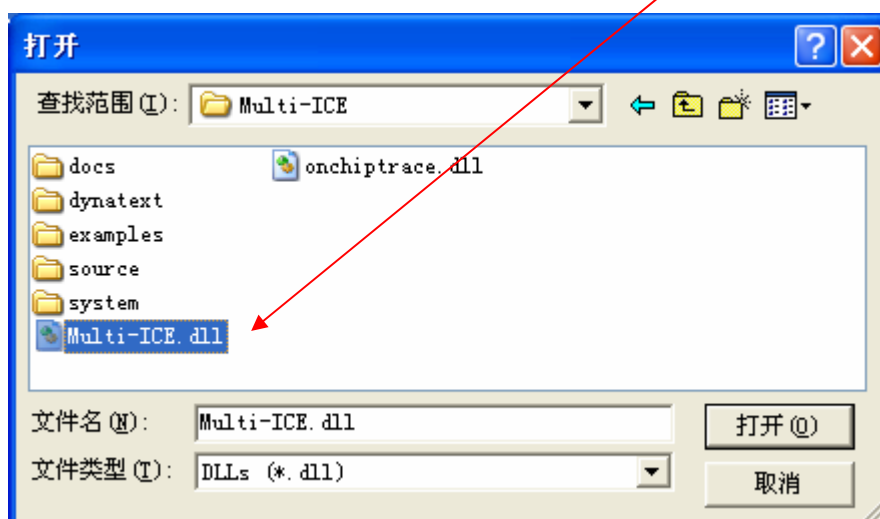
弹出如下窗口，请点击此菜单



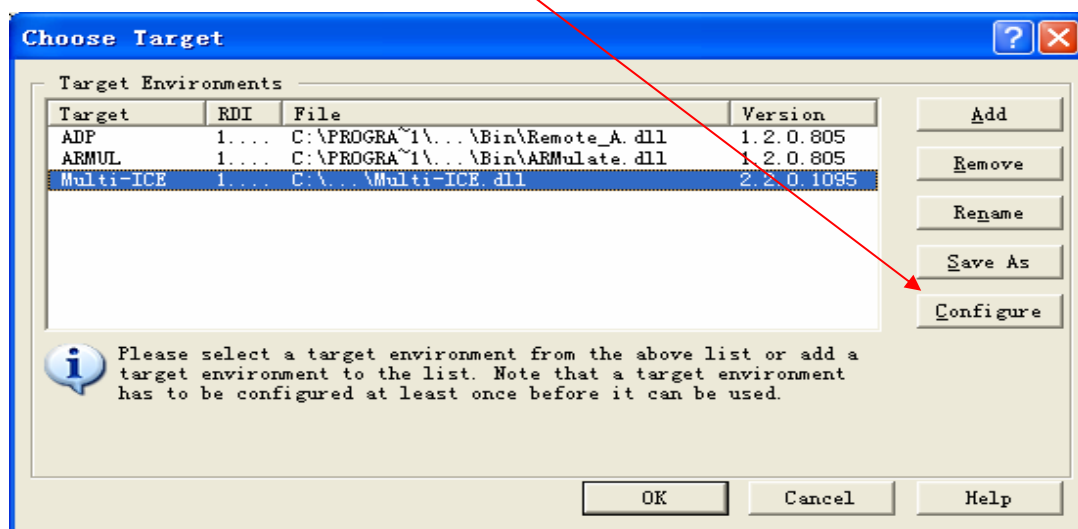
出现如下窗口，点击此按钮



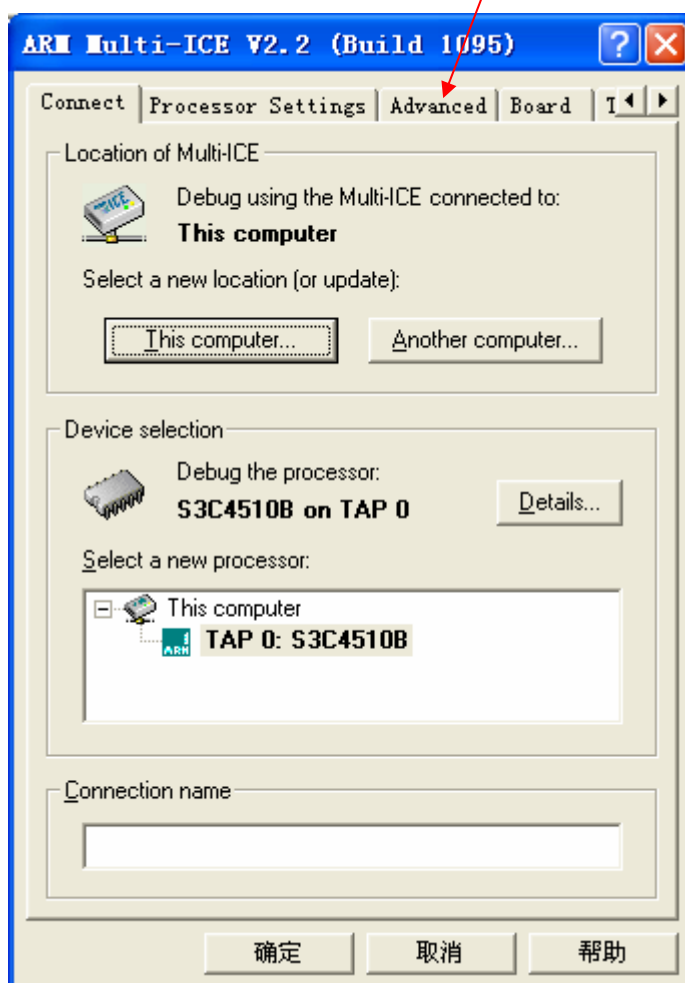
弹出如下窗口，请选择正确的文件，此文件名叫：
Multi-ICE.dll， 它的路径在 Multi-ICE Server
软件的安装目录下，如：C:\Program Files\ARM\Multi-ICE
下面等，找到后按确定钮



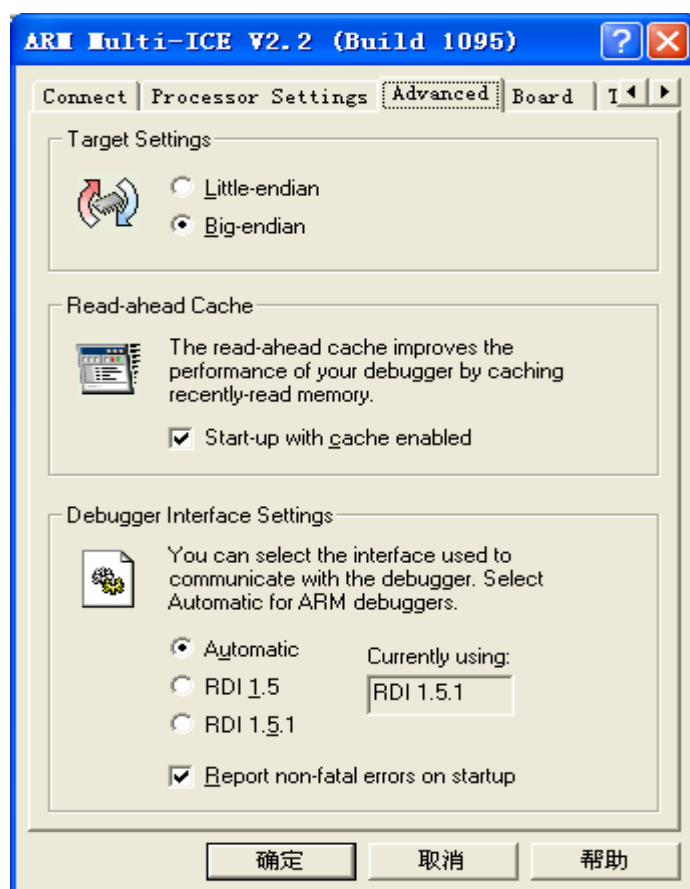
点击此图标进行设置



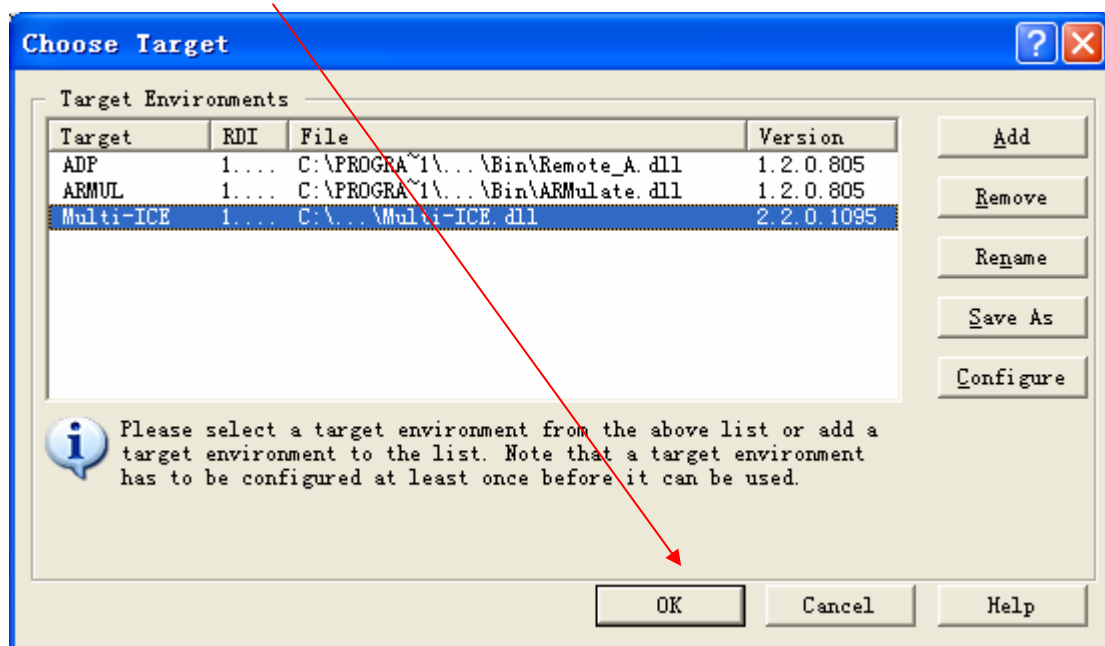
出现如下窗口，请点击此图标进行设置



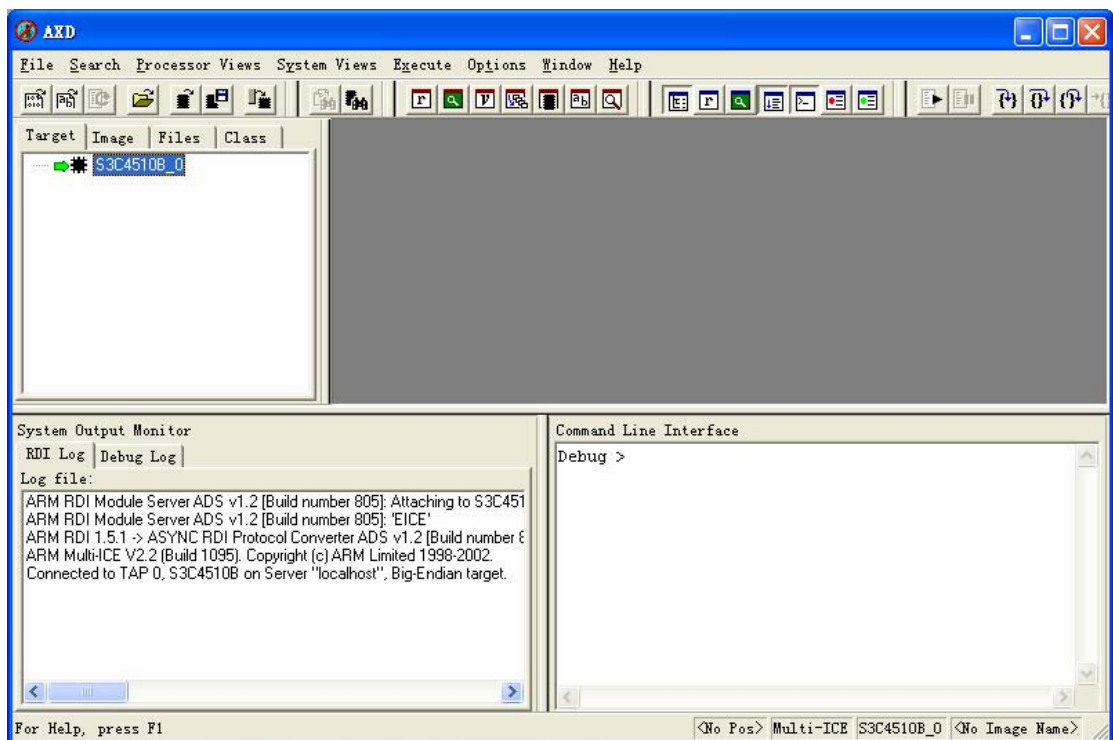
请正确设置，完成后，点击确定



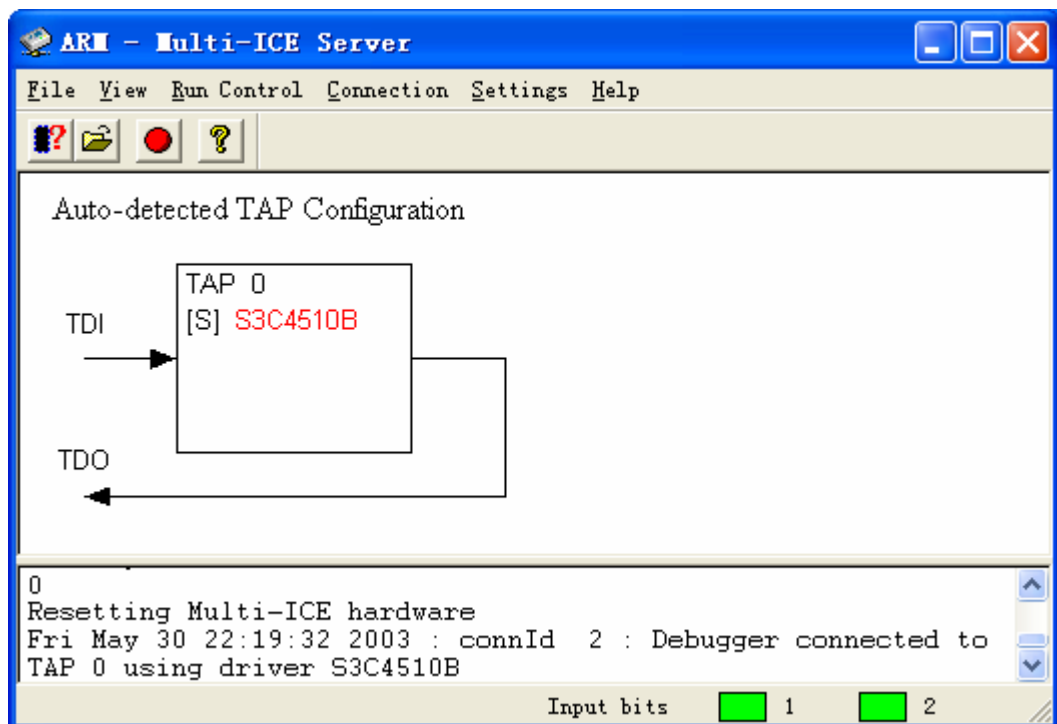
点击 OK



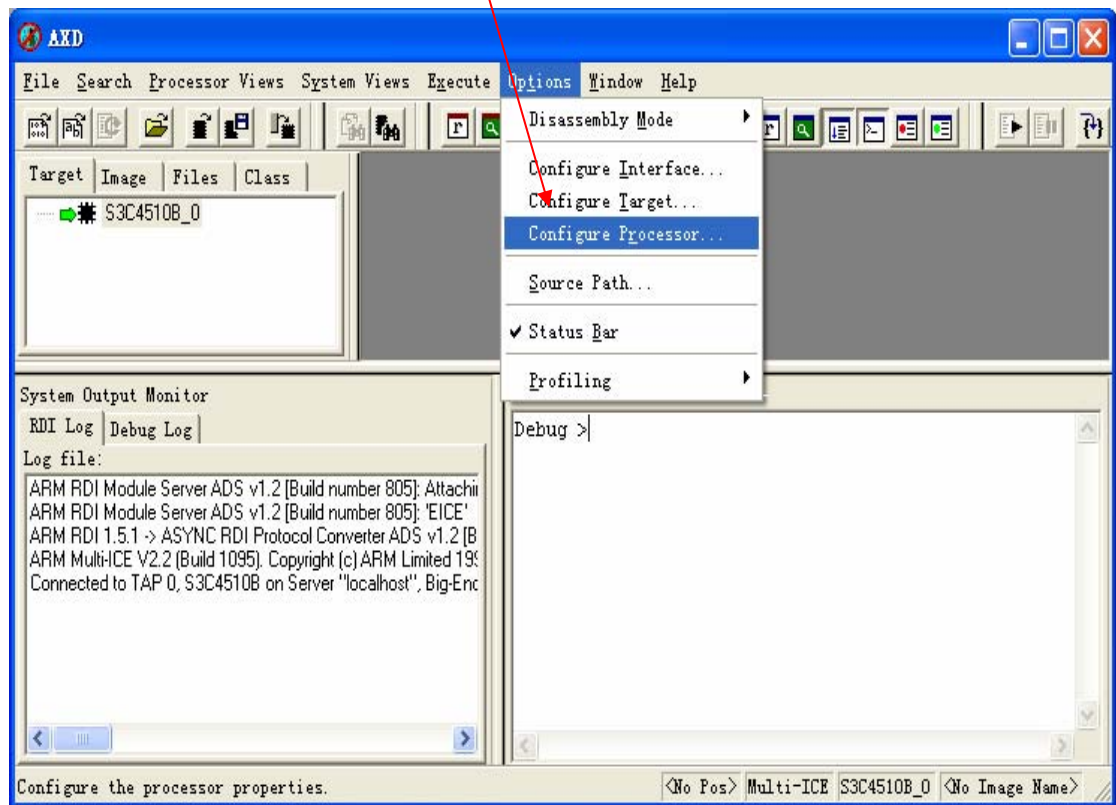
如果正常，出现如下窗口：



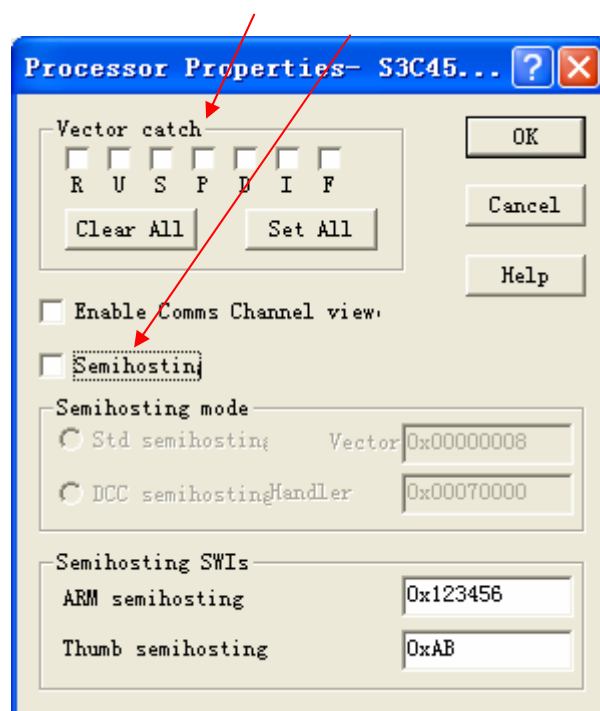
此时，Multi-ICE Server 窗口中部分字符的颜色会改变，表示 Multi-ICEServer 已经与 ADS 调试环境连接起来



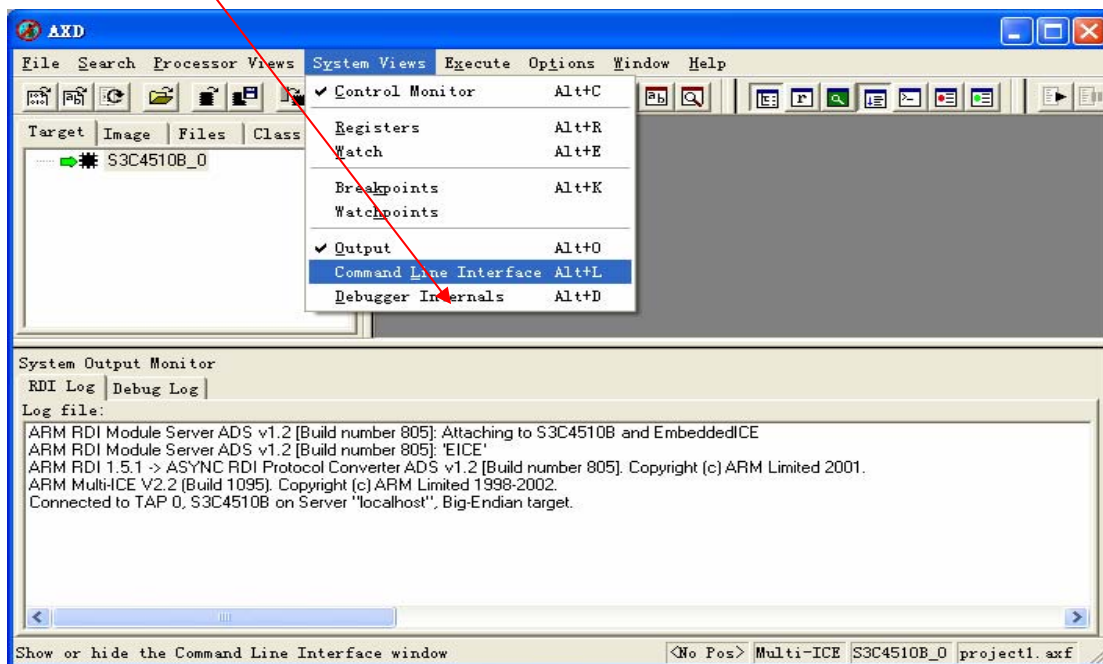
在 ADS 的调试窗口中打开以下菜单



弹出窗口, 把它的设置改变成跟下图一样, 然后点击 OK。



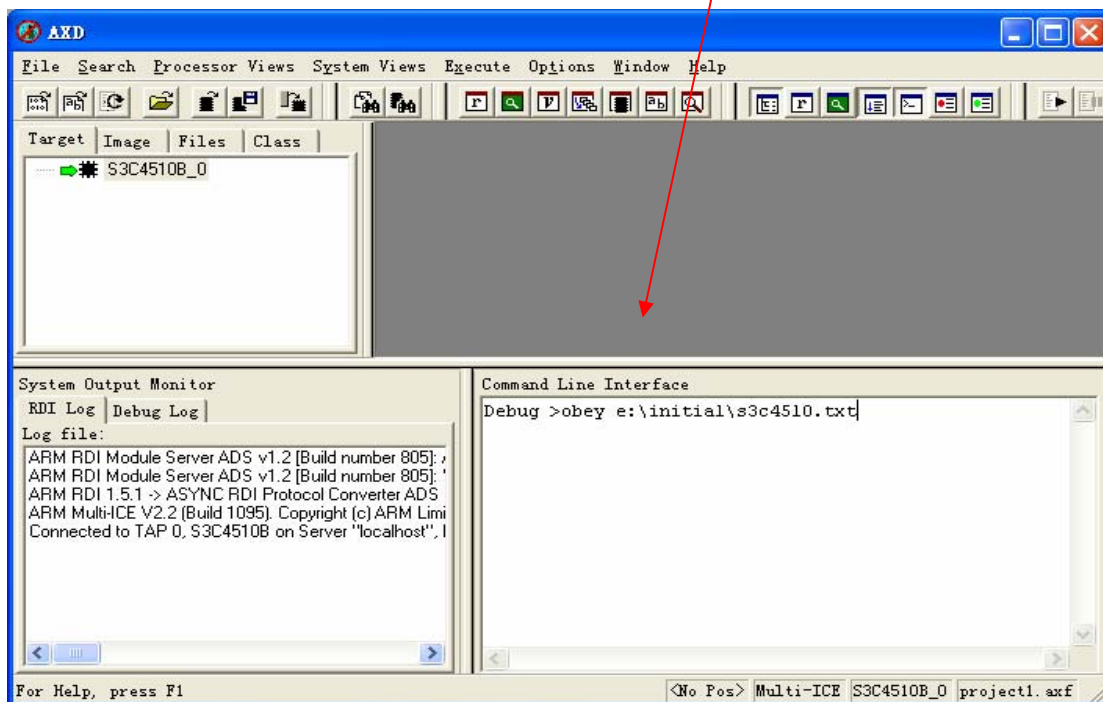
然后，打开以下菜单



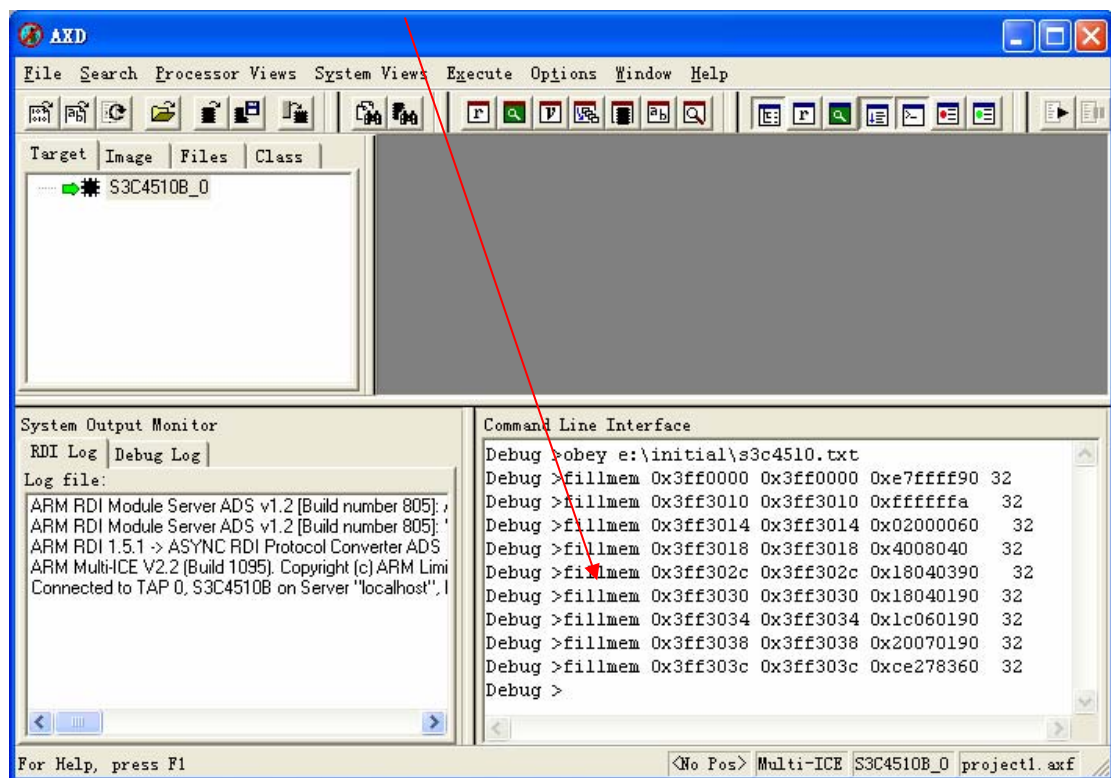
弹出以下窗口，然后用键盘输入以下字符：

`obey e:\initial\s3c4510.txt`

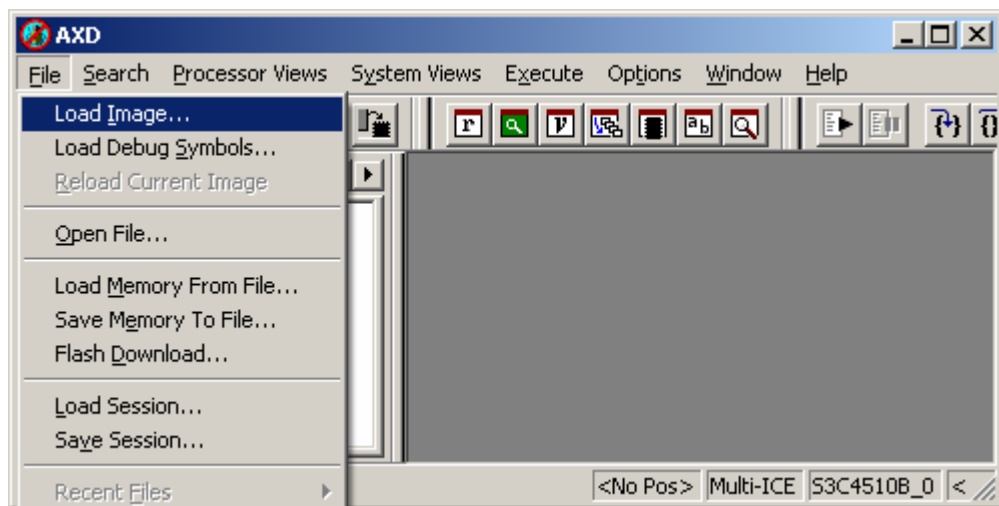
（注：其中 “e:\initial\s3c4510.txt” 表示 s3c4510.txt 文件的路径，此文件在光盘的 \initial 目录下可以找到，<e:\initial\s3c4510.txt>可以是任何您的 s3c4510.txt 文件所在路径，这里只是举例说明。）然后，回车

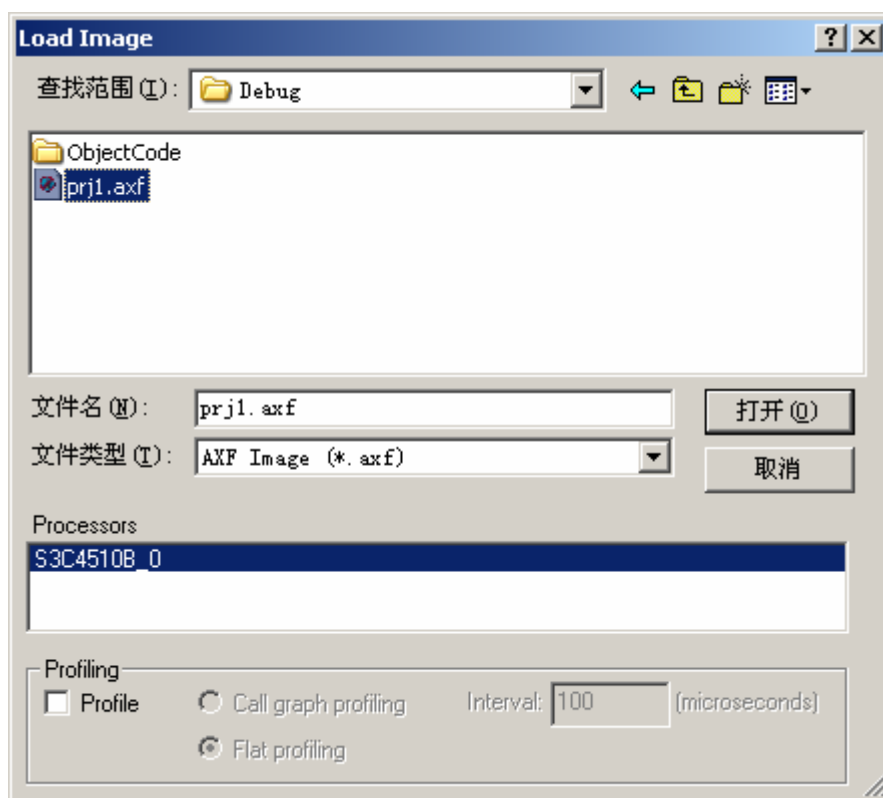


回车后，运行情况如下

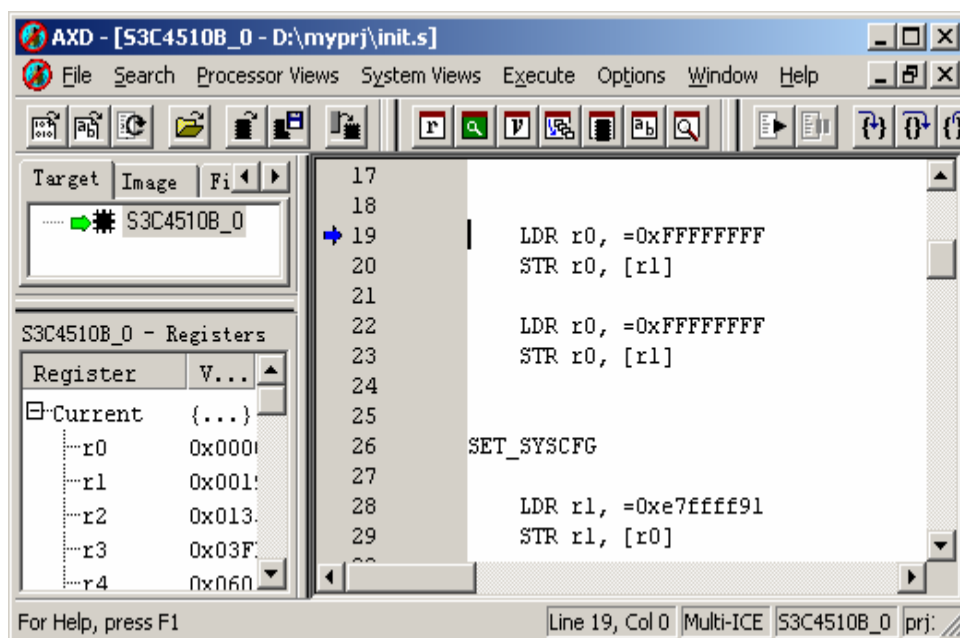


然后，打开以下菜单装载源程序

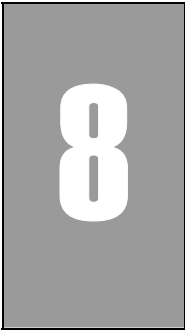




最后，出现源代码窗口，调试即开始



全速运行例子程序，4510 评估板的指示灯将闪烁。



附录

JTAG 接口的 TCK 频率设置参考表

JTAG 的 TCK 频率设置，频率—周期 对照表

Frequency (kHz)	Half- period (ns)	Value	Frequency (kHz)	Half- period (ns)	Value
10000.00	50	0	454.55	1100	21
5000.00	100	1	434.78	1150	22
3333.33	150	2	416.67	1200	23
2500.00	200	3	400.00	1250	24
2000.00	250	4	384.62	1300	25
1666.67	300	5	370.37	1350	26
1428.57	350	6	357.14	1400	27
1250.00	400	7	344.83	1450	28
1111.11	450	8	333.33	1500	29
1000.00	500	9	322.58	1550	30
909.09	550	10	312.50	1600	31
833.33	600	11	294.12	1700	48
769.23	650	12	277.78	1800	49
714.29	700	13	263.16	1900	50
666.67	750	14	250.00	2000	51
625.00	800	15	238.10	2100	52
588.24	850	16	227.27	2200	53
555.56	900	17	217.39	2300	54
526.32	950	18	208.33	2400	55
500.00	1000	19	200.00	2500	56

Frequency (kHz)	Half- period (ns)	Value	Frequency (kHz)	Half- period (ns)	Value
476.19	1050	20	192.31	2600	57
185.19	2700	58	59.52	8400	116
178.57	2800	59	56.82	8800	117
172.41	2900	60	54.35	9200	118
166.67	3000	61	52.08	9600	119
147.06	3400	80	50.00	10000	120
138.89	3600	81	48.08	10400	121
131.58	3800	82	46.30	10800	122
125.00	4000	83	44.64	11200	123
119.05	4200	84	43.10	11600	124
113.64	4400	85	41.67	12000	125
108.70	4600	86	40.32	12400	126
104.17	4800	87	39.06	12800	127
100.00	5000	88	36.76	13600	144
96.15	5200	89	34.72	14400	145
92.59	5400	90	32.89	15200	146
89.29	5600	91	31.25	16000	147
86.21	5800	92	29.76	16800	148
83.33	6000	93	28.41	17600	149
80.65	6200	94	27.17	18400	150
78.13	6400	95	26.04	19200	151
73.53	6800	112	25.00	20000	152
69.44	7200	113	24.04	20800	153
65.79	7600	114	23.15	21600	154

Frequency (kHz)	Half- period (ns)	Value	Frequency (kHz)	Half- period (ns)	Value
62.50	8000	115	22.32	22400	155
21.55	23200	156	7.44	67200	212
20.83	24000	157	7.10	70400	213
20.16	24800	158	6.79	73600	214
19.53	25600	159	6.51	76800	215
18.38	27200	176	6.25	80000	216
17.36	28800	177	6.01	83200	217
16.45	30400	178	5.79	86400	218
15.63	32000	179	5.58	89600	219
14.88	33600	180	5.39	92800	220
14.20	35200	181	5.21	96000	221
13.59	36800	182	5.04	99200	222
13.02	38400	183	4.88	102400	223
12.50	40000	184	4.60	108800	240
12.02	41600	185	4.34	115200	241
11.57	43200	186	4.11	121600	242
11.16	44800	187	3.91	128000	243
10.78	46400	188	3.72	134400	244
10.42	48000	189	3.55	140800	245
10.08	49600	190	3.40	147200	246
9.77	51200	191	3.26	153600	247
9.19	54400	208	3.13	160000	248
8.68	57600	209	3.00	166400	249
8.22	60800	210	2.89	172800	250
7.81	64000	211	2.79	179200	251
2.69	185600	252	2.52	198400	254
2.60	192000	253	2.44	204800	255