

Multi ARM 实时仿真器 安装使用指南

目录

第一章 Multi ARM 介绍	
Multi ARM 产品介绍.....	2
第二章 ARM 开发系统安装	
2.1 软件安装.....	3
2.2 硬件连接.....	5
第三章 Multi ARM 简单使用	
3.1 启动硬件.....	6
3.2 启动软件.....	6
3.2.1 启动 Multi-ICE Server.....	6
3.2.2 启动 ADS.....	8
第四章 Multi-ICE Server 使用	
4.1 Multi-ICE Server 工具栏.....	16
4.2 Multi-ICE Server 菜单选项.....	16
4.2 Multi-ICE Server 状态.....	17
4.3 Multi-ICE Server 中的 JTAG 参数设置.....	19
4.4 Multi-ICE Server 配置.....	22
4.5 调试系统架构体系.....	23
第五章 Multi ARM 硬件介绍	
5.1 Multi ARM 接口定义.....	24
5.2 Multi ARM 接口电平.....	25
5.3 TCK 信号频率设置.....	25
5.5 RTCK 时钟.....	25
5.6 14 脚和 20 脚 JTAG 接口转换.....	25
附录 1 TCK 频率设置.....	27
附录 2 常见问题.....	29

第一章Multi ARM介绍

Multi ARM 产品介绍

Multi ARM 仿真器可以仿真所有 ARM 处理器内核，符合 JTAG IEEE 1149.1 规范，与 ARM 原装 Multi-ICE 100%兼容。可以与 ADS V1.2 集成开发环境相结合，支持全线 ARM 处理器内核。

Multi ARM 仿真器主要特征如下：

- 支持所有内建 embedded-ICE 逻辑单元的 ARM 处理器
- 支持所有符合 RDI 1.50 或 RDI 1.51 规范的调试工具软件
- 兼容不同电压的目标系统
- 支持同时调试多个内核系统
- 不占用目标板上的任何资源
- 支持实时硬件断点
- 修改寄存器、存储器内容
- 支持程序下载及实时调试
- 通信速度快，最高可达 10Mbps, JTAG 速度可设置
- 支持多种目标平台，如 ATMEL、SamSung、Intel、Philips、Sharp、Cirrus Logic、Tricend 等
- 仿真器内含 CPU，有自检功能
- 仿真器内含过压、过流保护电路

Multi ARM 支持的 ARM 内核有：

ARM7TDMI	ARM710T	ARM7TDMI-S
ARM7DMI	ARM720T	ARM740T
ARM7TDI-S	ARM7EJ-S	ARM926EJ-S
ARM9TDMI	ARM9E-S	ARM940T
ARM946E-S	ARM920T	ARM966E-S
ARM1020E	ARM922T	ARM966E-S
Intel Xscale PXA210	PXA250	80200

Multi ARM 仿真器支持以下操作系统

- Windows 95/98/NT/2000/ME/XP
- X86 RedHat Linux 6.2/7.1/7.2

第二章 Multi ARM开发系统安装

Multi ARM 物品清单:

- ① Multi ARM 硬件仿真器
- ② 20 针标准 Multi ARM 接口扁平连接电缆一条
- ③ 并口连接电缆一条
- ④ 直流 5V 开关电源一个
- ⑤ Multi ARM 使用手册光盘

2.1 软件安装

开发 PC 主机系统需求:

Pentium IBM 兼容 PC, 有一个并口

主频 200MHz 以上 CPU

32MB (For Win98) 或 64MB (For Win2000) 以上内存空间

300MB 以上空闲硬盘空间

CD-ROM驱动器

ARM 集成开发环境的安装过程分为二个步骤:

一 安装 ARM 公司的 **Multi-ICE** 驱动程序。

把 ARM 开发系统软件光盘插入 CD-ROM.

用文件管理器打开 ARM 光盘

进入 Multi-ICE 目录

双击 Setup.EXE 图标开始安装 Multi-ICE 驱动软件

二 安装 **ARM Developer Suite(ADS)**集成开发环境.

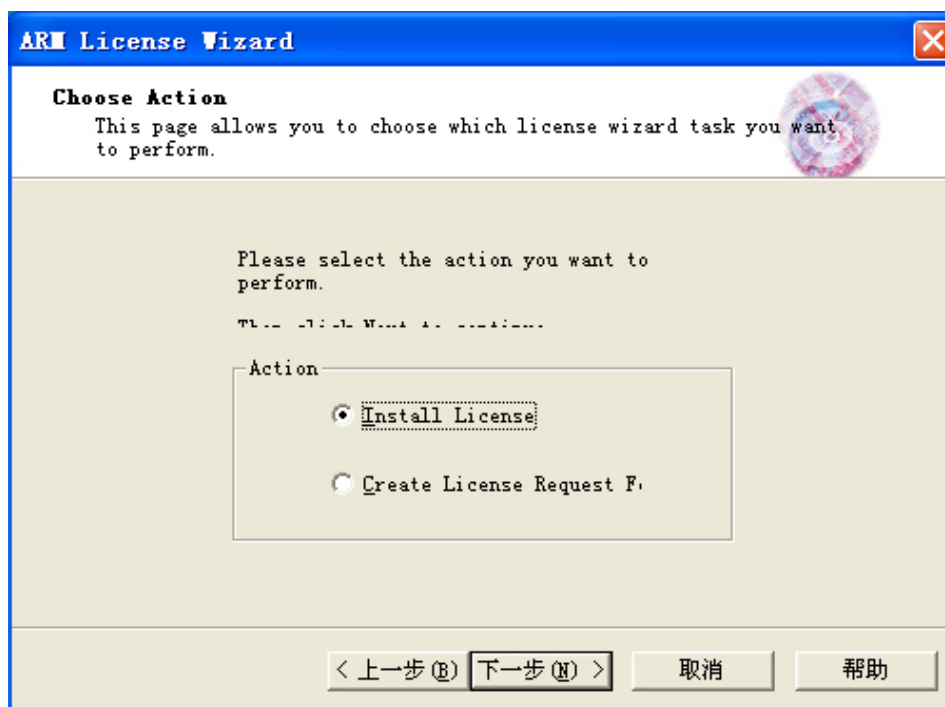
把 ARM 开发系统软件光盘插入 CD-ROM.

用文件管理器打开 ARM 光盘

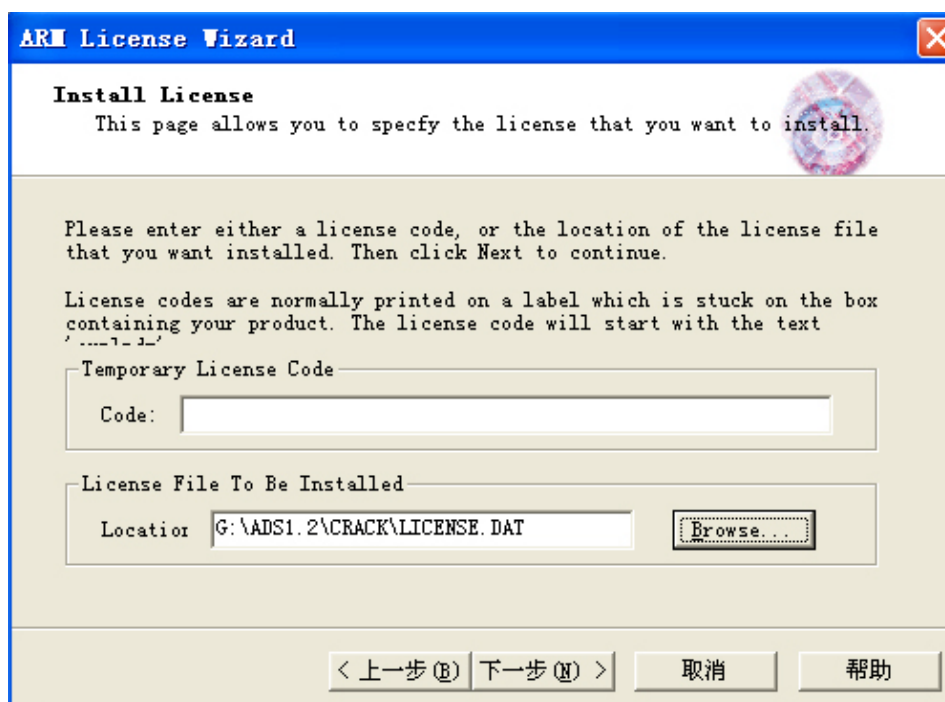
进入 ADS1.2 目录

双击 Setup.EXE 图标开始安装 ADS1.2 集成开发环境

在安装最后会要求指定 license 文件, 画面如下图所示:



选择 Install License, 按<下一步(N)>按钮, 将出现下面的画面:



按<Browse>按钮, 将 license 文件指定为 ARM 光盘上的
G:\ADS1.2\CRACK\LICENSE.DAT (其中 G:为光盘号)

到此, ARM 软件已安装完毕, 请重新启动计算机。

2.2 硬件连接



仿真器实物图

2.2.1 25 芯并口

一端连接 Multi-ARM 仿真器，另一端连接 PC 机的并口

注意：要将打印机的并口工作方式设置为：EPP 或 ECP

2.2.2 电源

Multi-ARM 使用单+5V 电源，仿真器所配电源仅供仿真器本机使用，目标板应使用另外的电源，目标板电压范围为 2.0V 至 5.0V。

2.2.3 20 芯仿真电缆

与目标板连接的 JTAG 仿真电缆，红线为 1 脚。信号定义见第 25 页

2.2.4 10 芯用户 I/O 口

与目标板 I/O 口的连接电缆，一般不使用。信号定义见第 26 页

2.2.5 指示灯

Multi-ARM 有三个指示灯

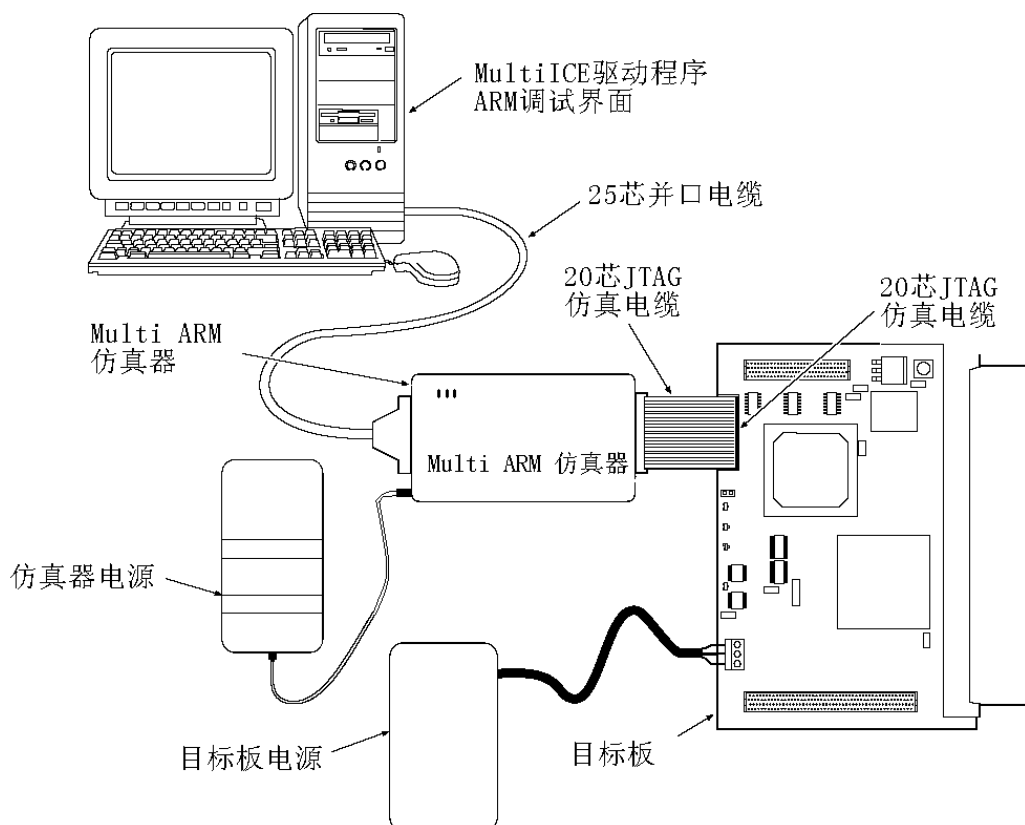
红色：电源指示。当正确接入电源后该指示灯亮。

当电源故障（过压、过流、极性接反）时，仿真器内的保护电路会自动断开电源，该指示灯灭。

绿色：运行指示。当 Multi-ARM 仿真器与 PC 机通信时，该指示灯亮闪烁

黄色：状态指示。在加电时，仿真器首先进行自检，此时该指示灯快速闪烁，当仿真器无法通过自检时，该指示灯常亮，表明仿真器有故障。此后每隔 20 秒仿真器会再次进行自检。当仿真器通过自检后，仿真器会进一步检查目标板是否连接，若目标板连接正确，该指示灯灭，否则该指示灯每秒闪烁一次。

第三章 Multi ARM使用



3.1 启动硬件

按上图所示连接好PC、Multi ARM仿真器、目标板，并将仿真器及目标板电源打开。此时应红色电源指示灯亮，黄色状态指示灯灭。表明仿真器及目标板已连接好。

3.2 启动软件

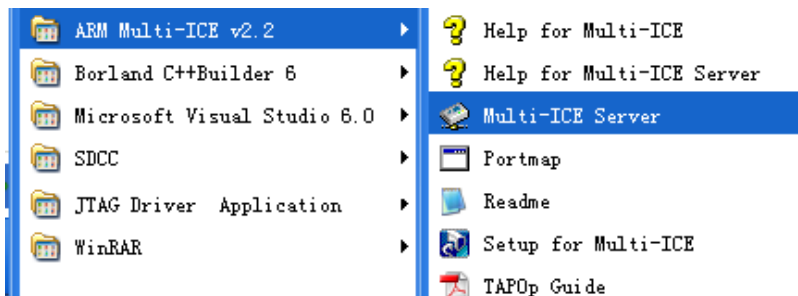
启动软件分二步：

启动 **Multi-ICE Server** 仿真器驱动程序

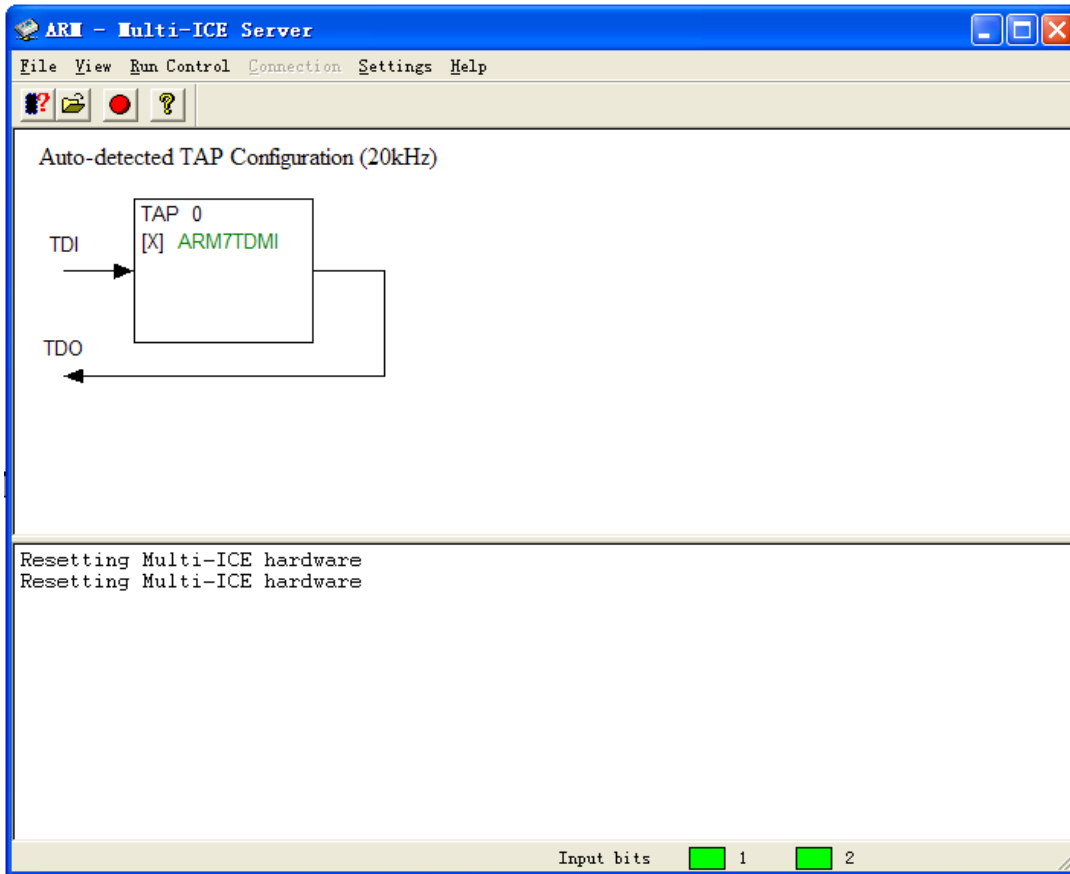
启动 **CodeWarrior for ARM Developer Suite** (ADS 集成开发环境)

3.2.1 启动 Multi-ICE Server （硬件已连接好）

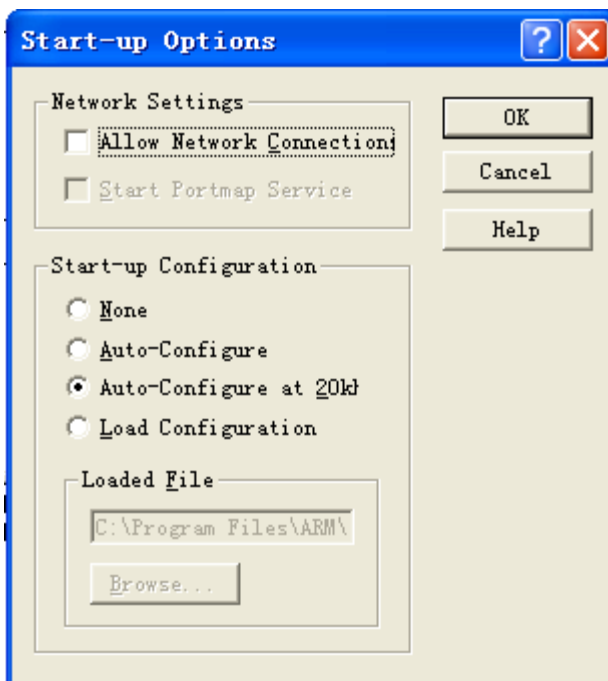
如下图所示，点击 Multi-ICE Server (仿真器驱动程序)



出现以下画面：（第一次运行可能因设置问题，显示一些其它提示框）



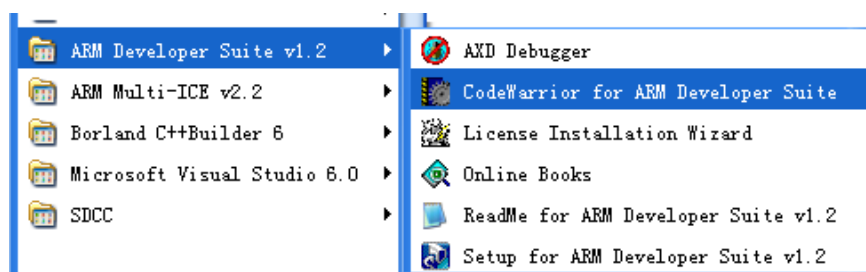
第一次使用时先要作一些设定。点击 Settings\Start-up Options. 如下图所示：



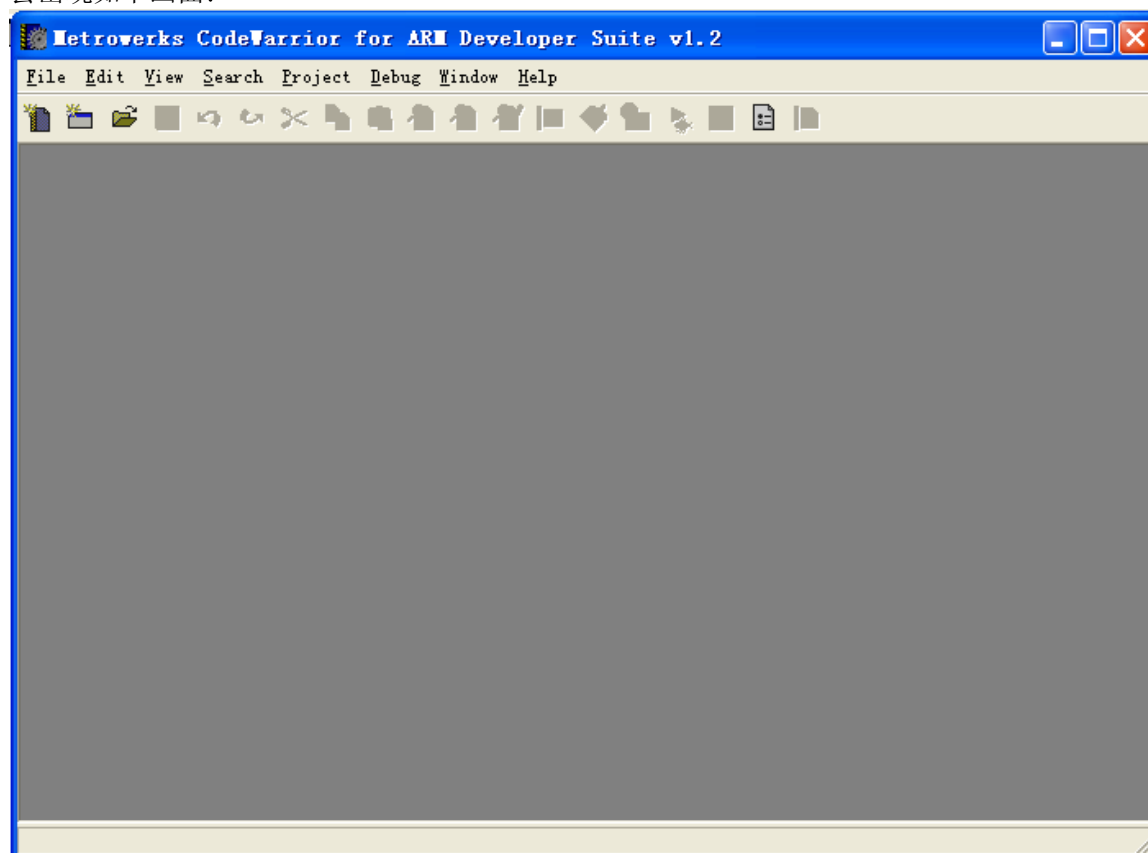
如图所示，去掉 Allow Network Connection 选项（该选项启动网络调试）
选择 Auto-Configure 或 Auto-Configure at 20kHz，具体选用哪一个需要试一试，
Auto-Configure 使用的 JTAG 速度快，对目标板的要求高一些。如果 Auto-Configure
不能联机，再试一试 Auto-Configure at 20kHz。

完成设置，需要退出 Multi-ICE 驱动程序，重新启动 Multi-ICE 驱动程序，因为
新的设置只有在重新启动驱动程序时生效。

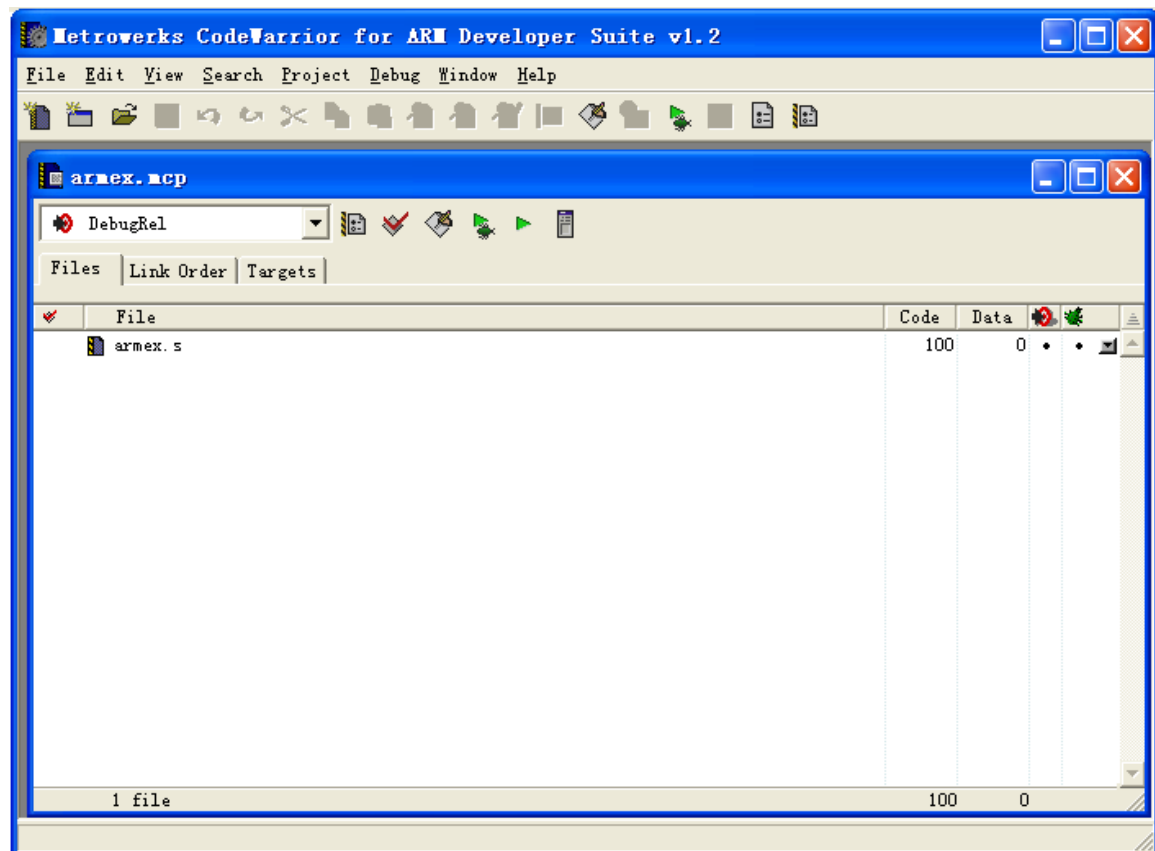
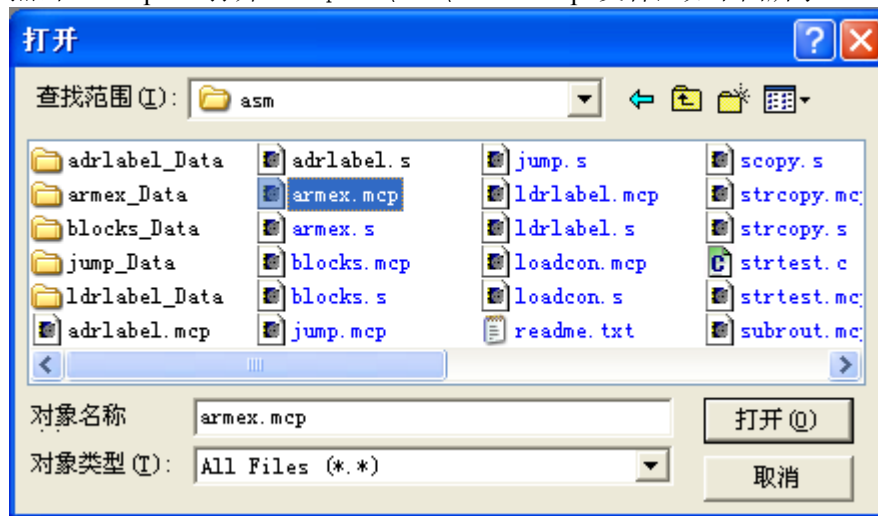
3.2.2 启动 CodeWarrior for ARM Developer Suite （ADS 集成开发环境）



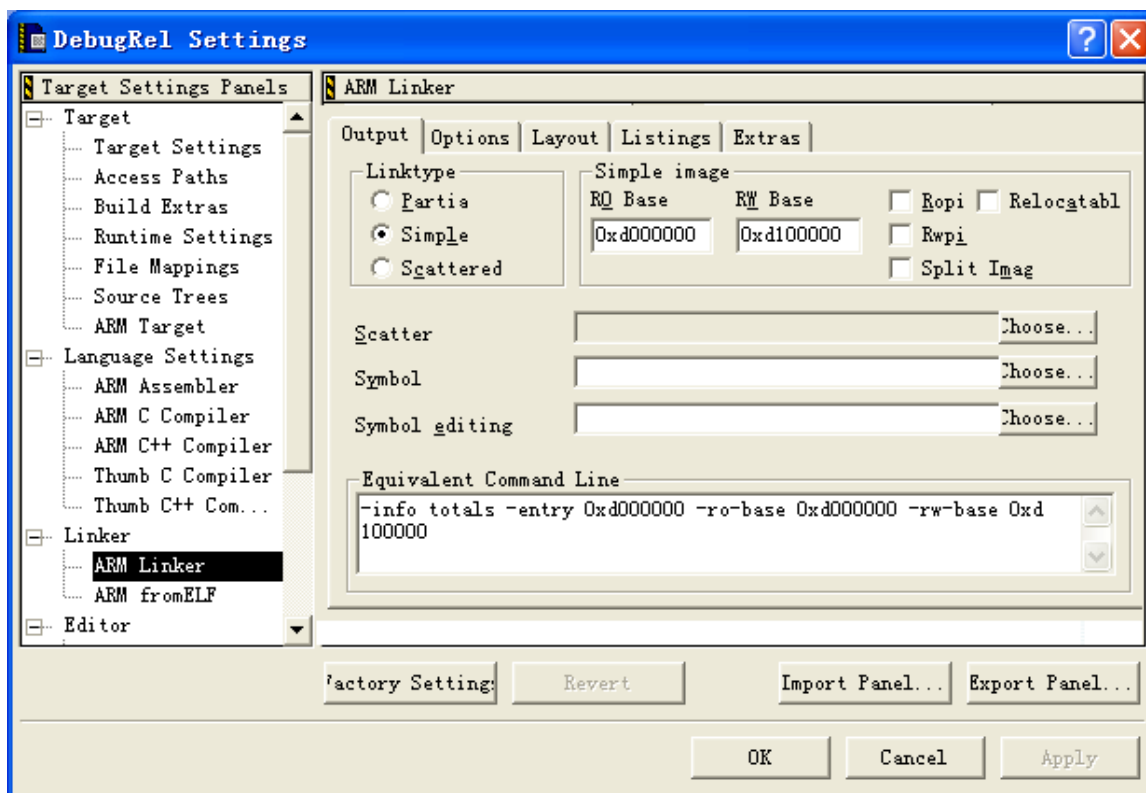
会出现如下画面：



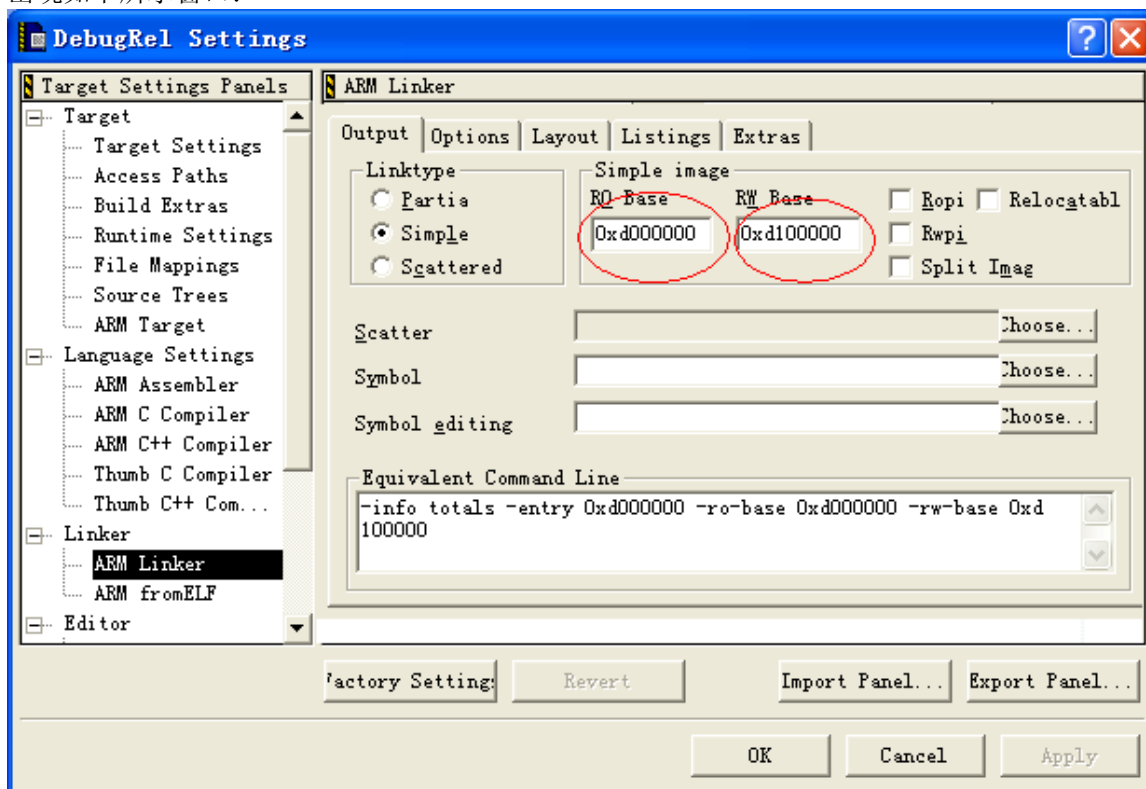
点击 File\Open: 打开 Examples\asm\asmex.mcp 文件，如下图所示：



在调入了项目文件后，还要根据目标板的具体情况进行一些必要的设置工作。
点击如上图所示的 DebugRel Settings...按钮

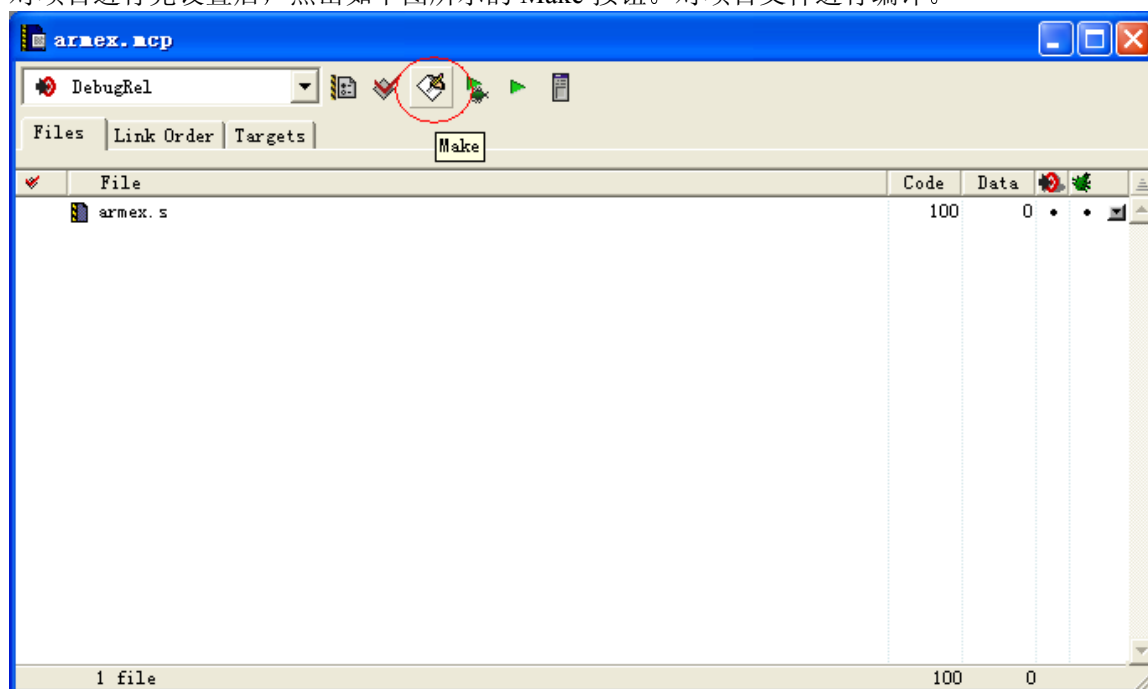


出现如下所示窗口：



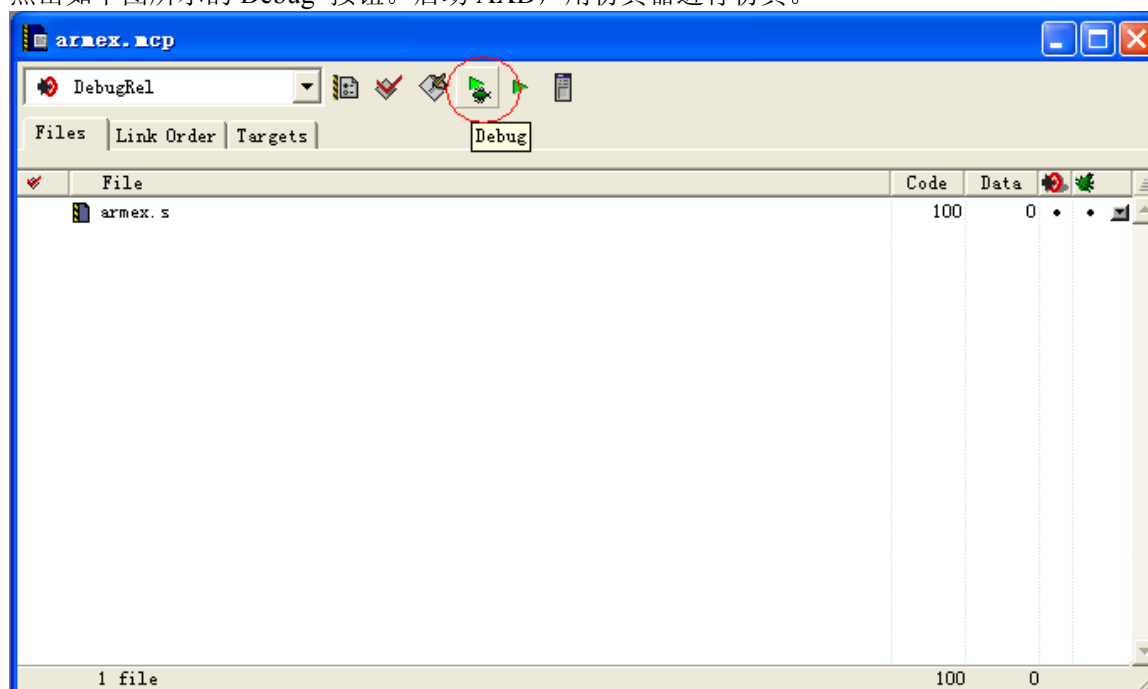
将 RO Base, RW Base 值改为目标板的 ROM, RAM 地址，然后点击 OK 按钮。

对项目进行完设置后，点击如下图所示的 Make 按钮。对项目文件进行编译。

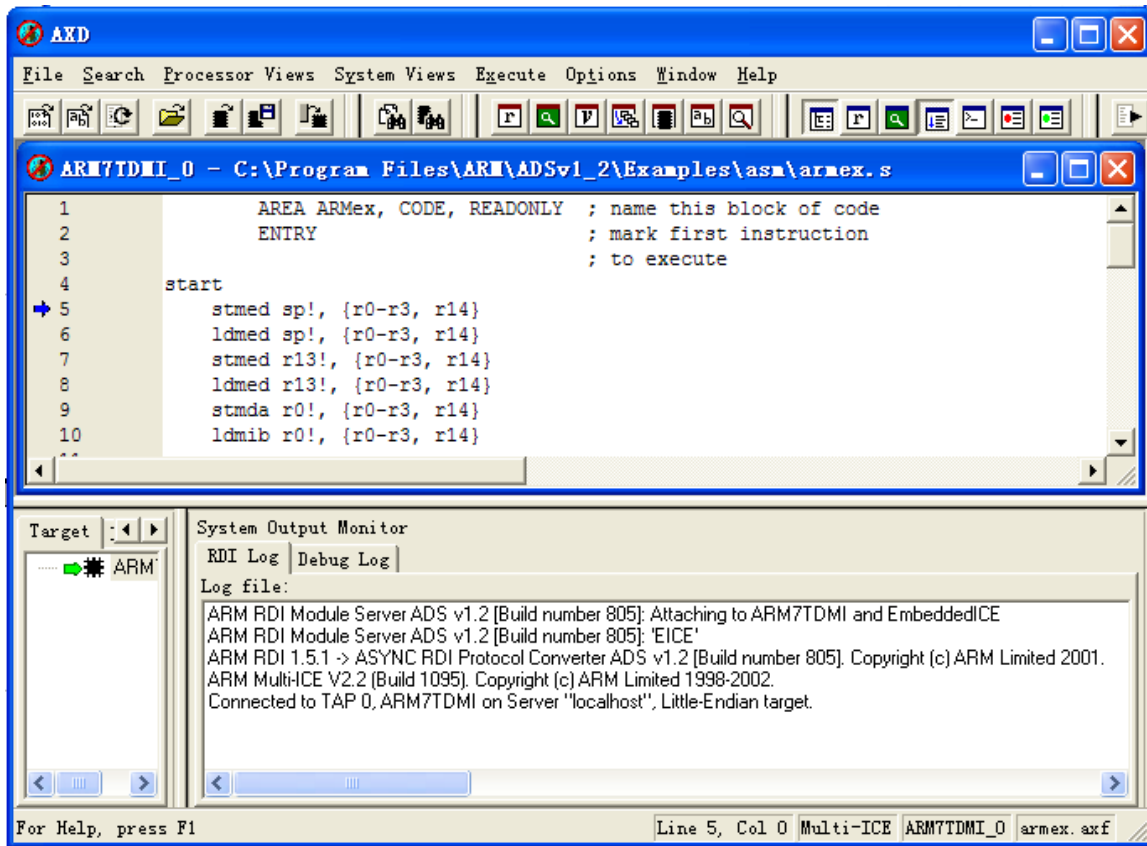


在对项目进行了编译之后如果没有编译错误，就可以进行仿真了。

点击如下图所示的 Debug 按钮。启动 AXD，用仿真器进行仿真。



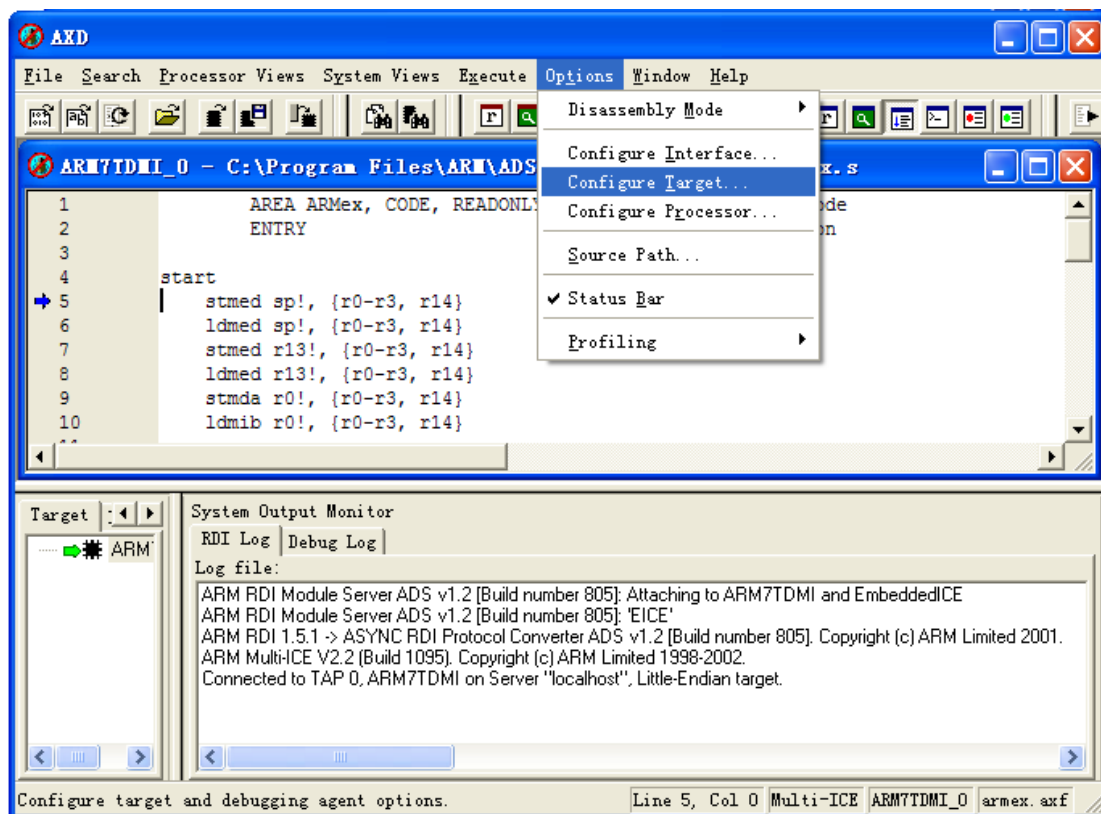
画面如下：



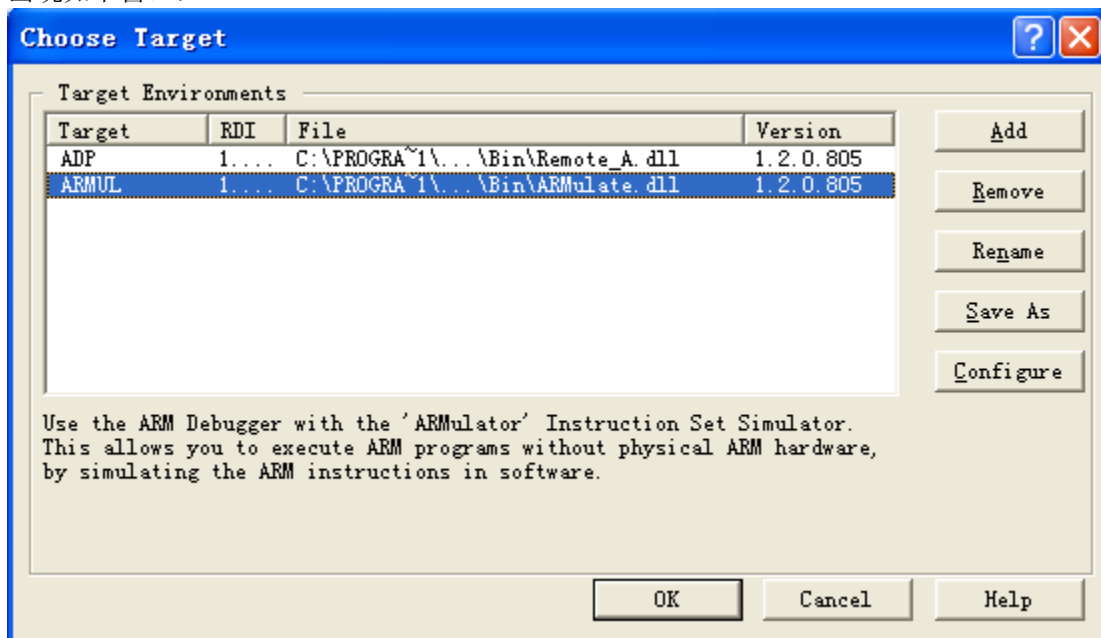
在第一次使用 AXD 时，AXD 的缺省设置是软件模拟，在进行硬件仿真之前要对 AXD 进行一些必要的设置工作。在 AXD 中选择 Multi-ICE Server 作为仿真器。

操作步骤如下：

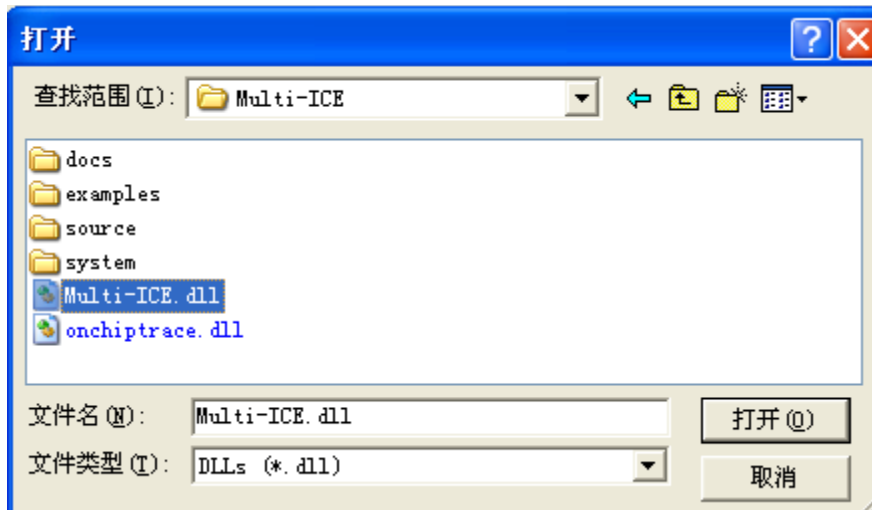
点击如下图所示的 Options\Configure Target



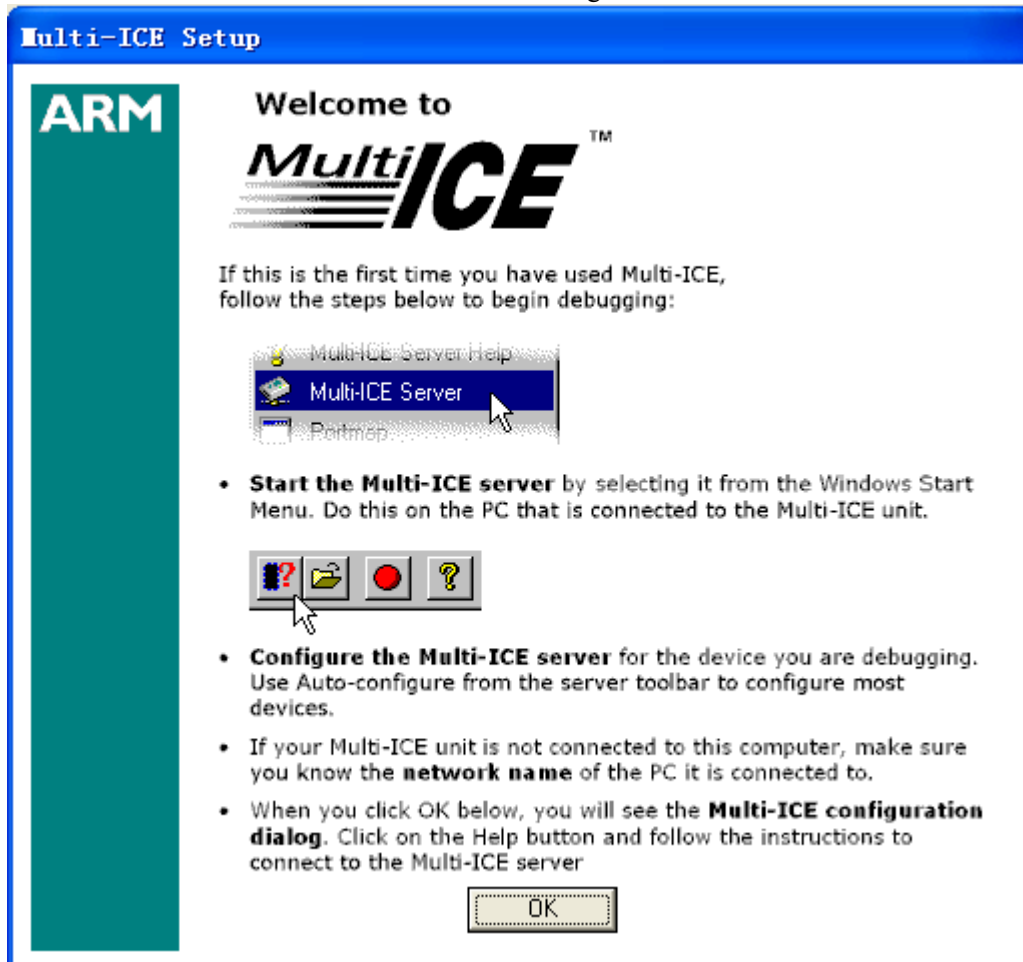
出现如下窗口：



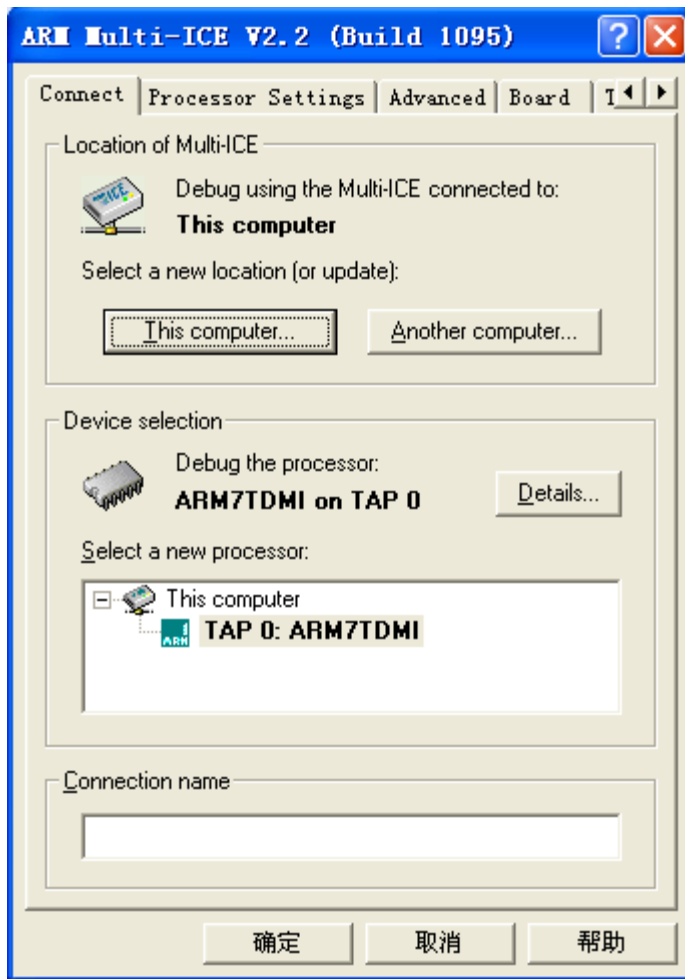
点击<Add>按钮。出现以下画面：



选择 Multi-ICE 的安装路径（如果在安装时没有改路径则是\Program Files\ARM\Multi-ICE）中的 Multi-ICE.dll 然后点击“确定”按钮。再按<Configure>按钮，出现以下画面：



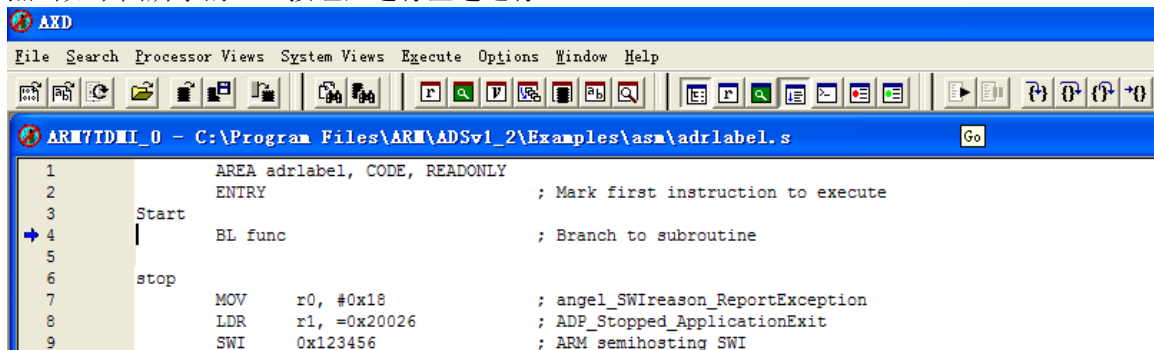
点击“确定”按钮。出现以下画面：



点击<确定>按钮，AXD 就配置好了。

退出 AXD，再次进入 AXD Debug 调试界面就是使用硬件仿真器了。

点击如下图所示的 Go 按钮，进行全速运行。



你还可以单步执行、设置断点、观察变量等操作。具体使用细节请参考 AXD 帮助文件。

第四章 Multi-ICE Server使用

4.1 工具栏

在工具栏里，总共有四个功能按钮。



4.2 Multi-ICE Server 菜单选项

这里介绍一下 Multi-ICE 程序中各个菜单及里面的每个选项。

File 菜单

- Load Configuration** 一读取用户指定的配置文件，对仿真器进行手动配置。
- Auto-Configure** 一自动检测和配置目标系统。
- Auto-Configure at 20KHz** 一自动检测和配置目标系统，
并且设置 TCK 信号的频率为 20KHz。
- Reset Target** 一对目标系统复位，有效的复位信号在
Setting/JTAG settings 菜单调定
- Log** 一输出信息存储到指定的日志文件中。
- Set Log File** 一用于指定日志文件的路径和名字。
- Recent File List** 一显示最近调用过的配置文件路径。
- Exit** 一退出程序。

View 菜单

- Toolbar** 一关闭或打开工具栏。
- Status Bar** 一关闭或打开状态栏。
- RPC Calls** 一允许或屏蔽 RPC 信息在调试信息窗口中显示。
- Clear Debug Window** 一清空调试信息窗口中的显示信息。

Run Control 菜单

- Independent** 一使每个目标系统直接相互独立，不进行交互通信。缺省状态有效。
- All Run** 一启动所有的目标系统。
- All Run/Stop** 一当收到一条启动指令时启动所有的目标系统；
当任何一个目标系统停止时其它的系统也一起停止。
- Custom** 一执行用户自定义的设置。
- Set-up Custom** 一打开用户自定义设置窗口，设置各个目标系统之间的交互方式。
- Load Settings** 一读取以前保存下来的设置文件。
- Save Settings** 一把当前设置保存到文件。

Connection 菜单

这个菜单在调试程序和 Server 建立连接之后才有效。它会给每一个 TAP 控制器分配一个菜单项，为用户提供单独删除某个 TAP 控制器的选项。

Settings 菜单

- Port Settings** 一显示并口设置对话框，用来选择并口的地址和是否使用 4-bit
通信模式。同时显示当前的并口类型设置。在 PC 机的 BIOS 中，
要把并口模式选择为 EPP。

- User Output Bits — 设定用户 OUT2, OUT1。
- JTAG Settings — 显示 JTAG 口的设置对话框, 用来设置 JTAG 口相关的时序信息和复位行为。如果调用了配置文件, 则相应的设置会在窗口中反映出来。
- Start-up Options — 显示设置启动选项的对话框。

Help 菜单

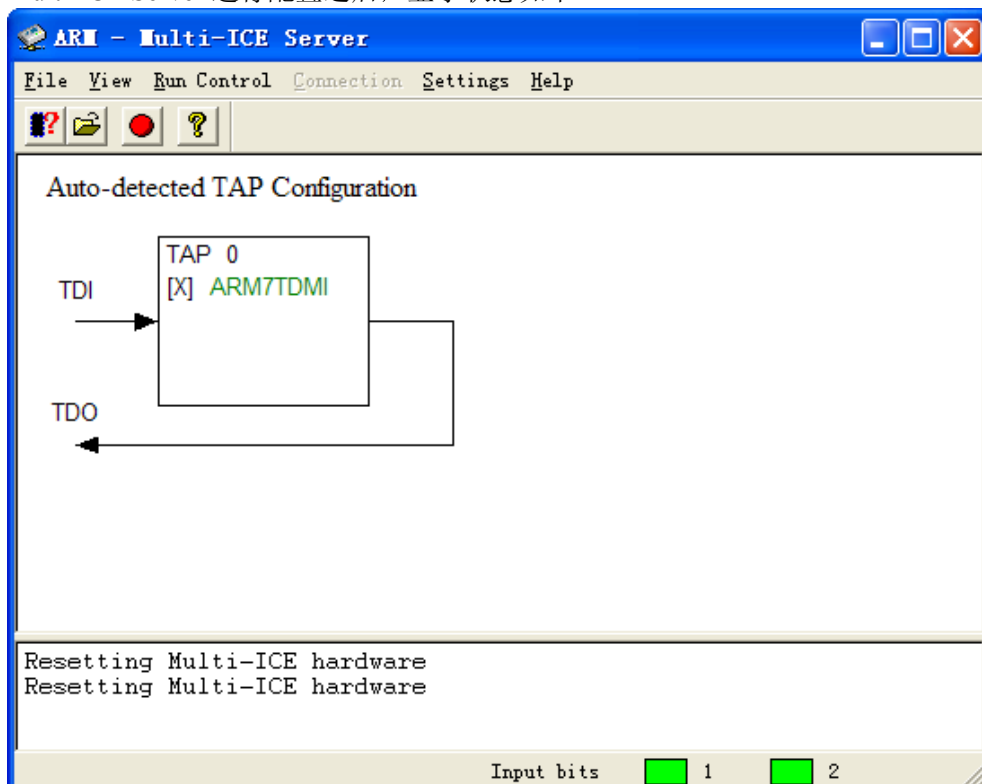
- Help Topics — 启动 Multi-ICE 的帮助系统。
- About Multi-ICE Server—显示软件版本信息。

4.2 Multi-ICE Server 状态

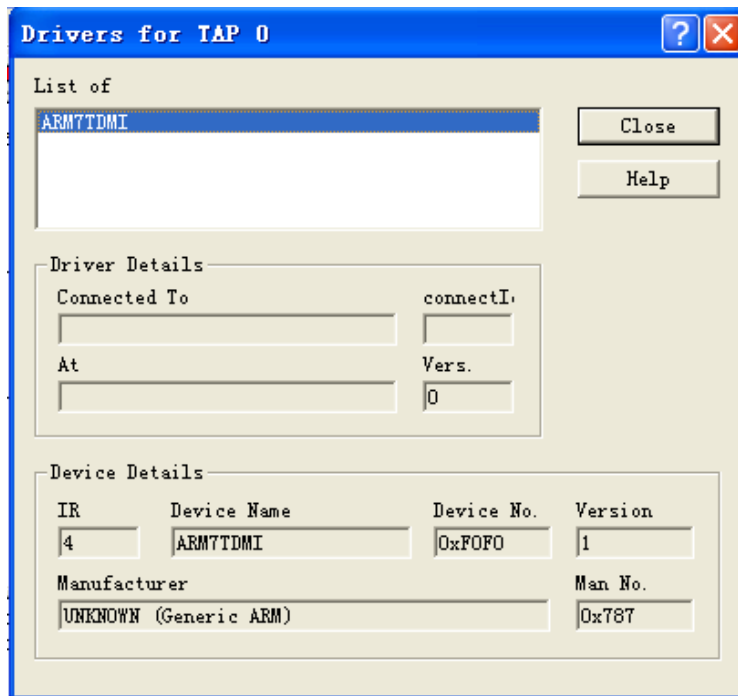
在 Multi-ICE Server 的 TAP 信息显示区, 可以直观的显示当前系统状态。下面通过单内核目标系统为例进行说明。

1. 配置后的状态

Multi-ICE Server 进行配置之后, 显示状态如下:



通过自动检测, 显示目标是一个 ARM7TDMI 的单内核系统, 在 TAP 配置显示区用图形的方式直观的显示出了检测结果。内核的类型名称“ARM7TDMI”显示为绿色, 并且前面有一个表示内核状态的字母, 这里字母“X”表示 Multi-ICE Server 目前还没有连接到任何调试程序中。双击图 4-8 中的处理器类型名字, 会弹出目标处理器附加信息窗口:

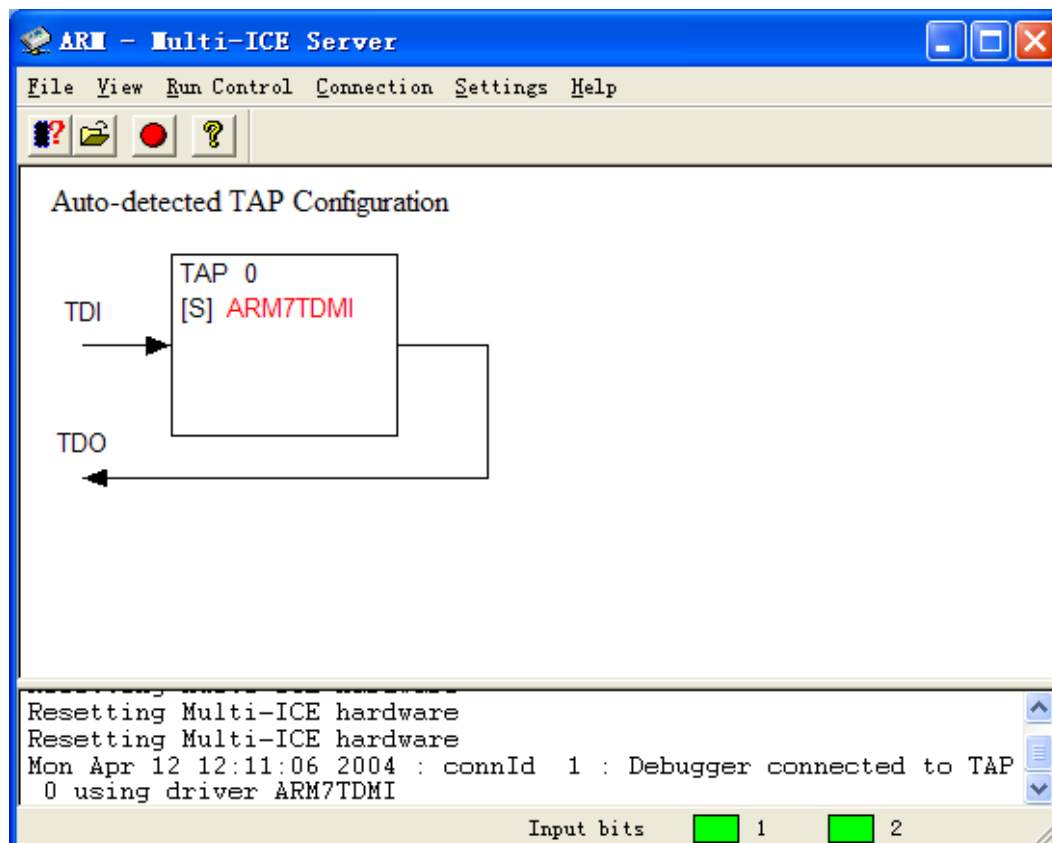


注意：处理器类型名称前面的字母叫做状态位，共有四种状态：

- [S] --处理器处于暂停状态
- [R]--处理器处于运行状态
- [D]--处理器处于下载状态
- [X]--处理器类型未知或没有被调用

2. 连接后的状态

如果有调试程序通过“MultiICE.dll”与 Multi-ICE 建立了连接，则 Multi-ICE 的状态变成如图 4-10 所示。此时处理器前面的状态字母变为“S”，表明已建立连接但正处于空闲状态，处理器名字变为红色，同时信息窗口中的信息增加了相应的内容。

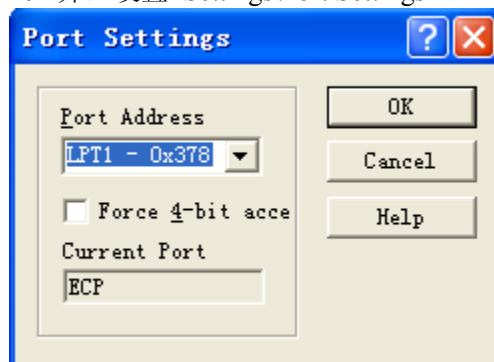


3. 激活后的状态

Multi-ICE 的活动状态有下载和运行两种，对应的字符分别为[D]、[R]

4.3 Multi-ICE Server 中的 JTAG 参数设置

1. 并口设置 Settings\Port Settings



Port Address —— 选择使用的并口地址，有三个可选项：

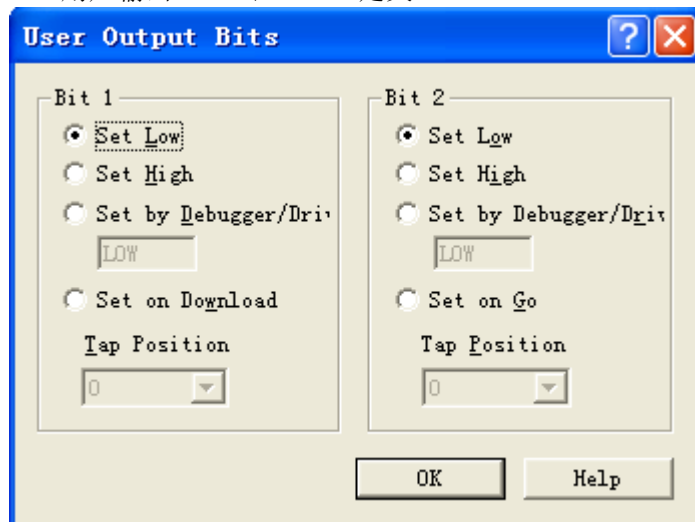
- **AUTO** 自动选择，系统默认选项
- **LPT1** 选择 LPT1
- **LPT2** 选择 LPT2

Force 4-bit access —— 使用 4-bit 的数据传输方式。

Current port Mode —— 显示并口类型，显示 BIOS 中并口设置状态，为只读选项。

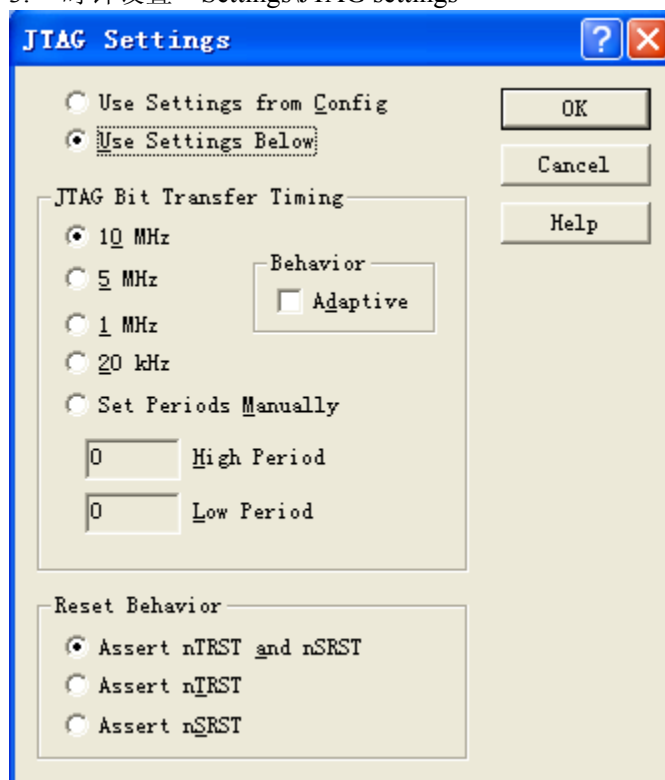
由于 Multi ARM 使用双向的并口数据总线，通常 ECP 或 EPP 类型能够符合要求，不同计算机主板厂商在并口设计上存在一些差异，当第一次使用 Multi ARM 时，需要对并口类型设置多作几次试验。

2. 用户输出 OUT1, OUT2 定义



用户可以通过这个窗口，改变用户输出端 OUT1, OUT2 的电平（高或低）

3. 时钟设置 Settings\JTAG settings

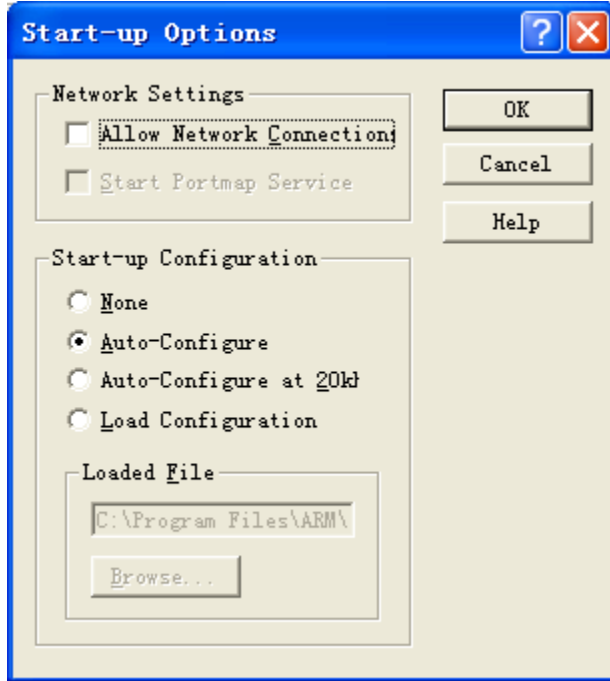


JTAG Bit Transfer Timing —— 设置 TCK 信号频率，如果需要手动设置请参见附录 1。

Behavior —— 选择使用 RTCK 功能。

Reset Behavior —— 选择哪组信号在按下复位按钮后有效。

4. 启动选项设置 Settings\Start-up Options



Network Settings —— 提供两个选项

- Allow Network Connections 使用网络连接功能，启动时自动检测网络设置情况
- Start Portmap Service 仅在上一项选中时有效

Start-up Configuration —— 设置 Multi-ICE Server 程序启动后自动执行的操作

4.4 Multi-ICE Server 配置

Multi-ICE Server 在运行时需要目标处理器的配置信息，比如内核类型和 IR（指令寄存器）长度等等。对 Multi-ICE Server 进行配置，有自动配置和手动配置两种方式。

1. 自动配置

在 Multi-ICE Server 程序中，选择 File\Auto-configure 菜单或者工具栏上的自动配置按钮。如果目标处理器是 ARM 族的内核，Multi ARM 能够自动检测到并进行相应配置，并把结果显示在 Multi-ICE Server 的显示窗口中。

注意：有时候检测 ARM710T/720T/740T/940T 这一系列内核时会返回一个 UNKNOWN 结果，这是 ARM 报告的一个硬件 bug，这时候就只能使用手动配置的方法。

经过自动配置后，JTAG 的通信速率会自动设置为 10MHz，用户可以在 Settings\JTAG Settings 菜单中按自己的要求选择合适的速率。

2. 手动配置

手动配置是通过调用一个用户预先定义好的配置文件来完成对 Multi-ICE Server 的配置。配置文件可以从 File\Load configuration 菜单中调入，配置文件为文本格式文件，文件后缀名为 .cfg。

配置文件主要包括以下几个部分：

- Title
- TAP controller
- Devices attached to each controller
- JTAG timing information
- Other options

下面是配置文件的格式示范：

```
[TITLE]
Double cores configuration demo ; 给配置方案命名
[TAP0]                          ; 目标系统中包含 TAP0 控制器
ARM7TDMI                        ; TAP0 控制上连着一个 ARM7TDMI 核
[TAP1]                          ; 目标系统中包含 TAP1 控制器
ARM7TDMI-S                      ; TAP1 上连着一个 ARM7TDMI-S 核
; 如果有更多的 TAP 控制器和内核，依次往下增加
[Timing]                        ; JTAG 口的时序设置
High = 9                        ; TCK 信号的高电平时间
Low = 9                          ; TCK 信号的低电平时间
Adaptive = ON                   ; RTCK 功能开关（ON 或 OFF）
[TAPINFO]
YES
[Reset]
nTRST
```

通常情况下目标系统只有单内核，仅使用[TAP0] 就可以了，对那些自动配置不能识别的内核，需要在配置文件中指定好正确的内核类型，然后在 Multi-ICE Server 中调入对应的配置文件就可以了。

“TAPINFO”选项主要是为 RISC 开发人员测试芯片时提供的。当该选项打开时，Multi ARM 完成正常的配置工作后，会继续从目标 RISC 中读取内核的其它信息供设计人员分析。这些附加信息可以从双击 Multi-ICE Server 的 TAP 显示窗口中的 TAP 控制器图标而弹出的信息窗口中得到。使用自动配置时，TAPINFO 是一直打开的，当调用用户自定义的配置文件时，TAPINFO 的默认状态时关闭。

“Reset”选项用来定义 Multi ARM 的复位动作。当用户按下 Multi-ICE Server 工具栏上的复位按钮时，Reset 选项内定义的信号就有效。合法的选项包括“nTRST”或“nSRST”，或者两者都选中。

“Timing”部分定义了 JTAG 端口的时序信息。关于 TCK 时钟参数的设置，请参见附录 1。在一个配置文件中，只有 TAP 控制器和内核类型的定义是必需的，其它部分都是可选项。另外，文件中用户的注释语句以分号开头。

4.5 调试系统架构体系

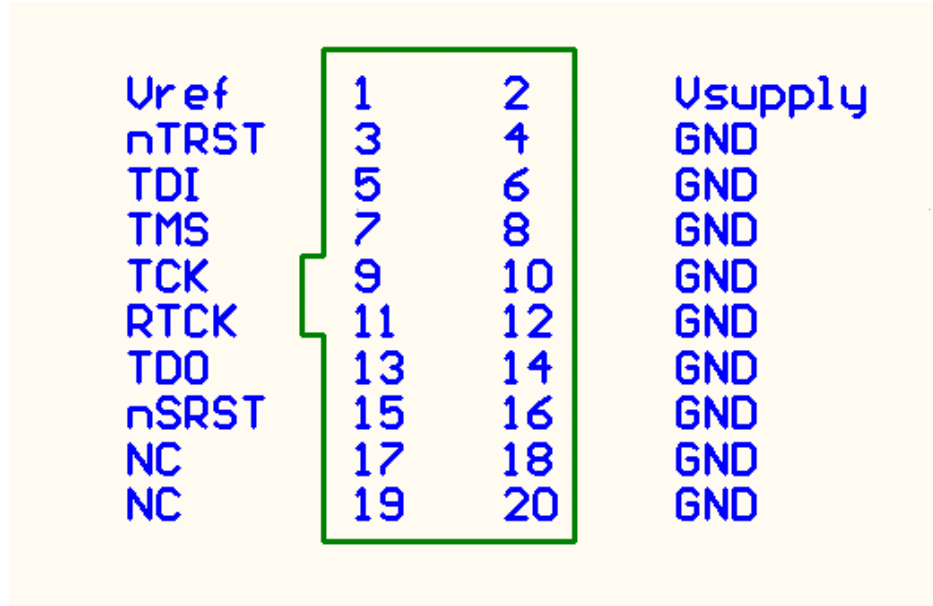
整个调试系统包括集成调试环境和硬件仿真器，它们组成了一个类似于“客户—服务器”模型的系统，服务器端把所有硬件封装起来，给客户端只提供了一个虚拟的处理器接口。

客户端的调试工具软件通过一个动态链接库与服务器相连，所以服务器一侧的硬件和客户端的调试工具软件是相互独立的。在同一个动态链接库的基础上，双方可以各自动行升级或替换，同样也方便了用户选择不同的仿真器和调试软件来组合自己的调试平台。

一个具体的调试程序应该能够支持许多不同的链接库，不同的链接库提供了不同的功能调用。以 ADS 工具包为例，“ARMulate.dll”提供了一个软 ARM 内核，用作程序的逻辑验证。若要把 Multi-ICE Server 程序连接在一起进行实时仿真，则只要调入“MultiICE.dll”就可以了。该动态库文件在 Multi-ICE Server 的安装目录下可以找到。

第五章Multi ARM硬件介绍

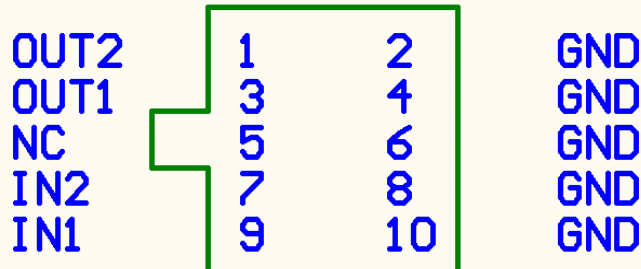
5.1 Multi ARM接口定义



脚号	信号名	方向	说明
1	Vref	输入	接口电平参考电压，通常直接接目标板电源
2	Vsupply	输入	目标板电源。Multi-ARM 不使用目标板电源, 但会检测目标板电源是否正常
3	nTRST	输出	JTAG 复位，目标板上应加上拉电阻
5	TDI	输出	JTAG TDI 数据输出端
7	TMS	输出	JTAG TMS 输出端
9	TCK	输出	JTAG TCK 时钟输出端
11	RTCK	输入	目标板反馈给 Multi-ARM 的时钟信号
13	TDO	输入	JTAG TDO 数据输入端
15	nSRST	输入 / 输出	仿真器可以向目标板发出复位信号，同时检测目标板的复位信号变化。目标板应加上拉电阻
17, 19	NC		保留。不要使用这些脚
4, 6, 8, 10, 12, 14, 16, 18	GND		地

注：早期的目标板可能使用 14 脚的 JTAG 插座, 有关 14 脚插座的信号定义见第 27 页。

用户 I/O 定义



脚号	信号名	方向	说明
1	OUT2	输出	用户可定义的输出端口 2, 可在 Multi-ICE 驱动程序中指定
3	OUT1	输出	用户可定义的输出端口 1, 可在 Multi-ICE 驱动程序中指定
7	IN2	输入	用户可定义的输入端口 2, 可在 Multi-ICE 驱动程序观察它
9	IN1	输入	用户可定义的输入端口 1, 可在 Multi-ICE 驱动程序观察它
5	NC		保留。不要使用此脚
2, 4, 6, 8, 10	GND		地

5.2 Multi ARM接口电平

仿真器推荐的正常使用电压为 2.5 – 5.0V，最大不超过 2.0 – 5.5V。随着电源电压的变化，接口的逻辑高电平和判决门限电平都将随之变化，这使得仿真器能够与不同电压和目标系统在逻辑上兼容。

5.3 TCK信号频率设置

为了与不同速率的目标对象相兼容，JTAG 口的通信速率应调整到一个合适的水平。数据传输速率是由 TCK 信号的频率来决定的，TCK 信号的真实频率跟设置值之间的对应关系参见附录 1。

5.4 目标板JTAG接口设计

目标板使用与仿真器一样的 20 脚针座。RTCK 和 nTRST 这两个信号根据目标 RSIC 是否提供对应的引脚来选用。nSRST 则根据目标系统的设计考虑来选择使用。

在目标系统的 PCB 设计中，最好把 JTAG 接口放置的离目标 RISC 近一些，如果这两者之间的连线过长，会影响 JTAG 口的通信速率。

5.5 RTCK时钟

RTCK 信号用来同步仿真器和目标板之间的通信，而不用关心 TCK 信号的具体频率。在没有收到目标系统的反馈时钟信号之前，仿真器不会触发一个新的 TCK。

如果不使用 RTCK 功能，这个信号脚可以简单的接地处理。如果使用了 RTCK 功能，把这个脚直接连向 ASIC 中对应的管脚。如果激活了 RTCK 功能，但 RISC 并没有提供对应原管脚，可以把 RTCK 和 TCK 这两个信号连在一起，但在连线中最好遵循下面的原则。这一点在边线比较长的时候尤其重要，目的是确保在 TCK 时钟到达目标点的时候才产生反馈信号。

5.6 14脚和20脚JTAG接口转换

有些系统采用一种 14 脚的 JTAG 插座，这两类接口的信号排列如下：

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

这两类接口之间的信号电气特性一样，因此可以把对应的信号直接相连进行转换。

附录1 TCK频率设置

在 Multi-ICE Server 设置中，TCK 不是直接使用频率值来进行设置，而是使用了一组整数，它们之间的对应关系参见附表。用户只需要找到需要的频率，把对应的设置数值填入设置窗口或是配置文件中即可。

TCK 时钟信号的波形不一定要等宽方波，可以高低电平设置分别不同的值，但是我们建议用户使用等宽方波波形，特别是不要设置占空比相差太大的波形。

从附表中可以看出，频率值与对应的周期值不正好是倒数关系，当高低电平的周期都是该设置值时，对应的频率值才是信号频率。若高低电平采用不同设置（通常情况下没有必要），则需要从表附找出各自对应的周期值相加，就是信号的周期，再从周期算得频率。

频率 (kHz)	半周期 (ns)	数值	频率 (kHz)	半周期 (ns)	数值	频率 (kHz)	半周期 (ns)	数值
10000	50	0	138.89	3600	81	18.38	27200	176
5000	100	1	131.58	3800	82	17.36	28800	177
3333.33	150	2	125	4000	83	16.45	30400	178
2500	200	3	119.05	4200	84	15.63	32000	179
2000	250	4	113.64	4400	85	14.88	33600	180
1666.67	300	5	108.7	4600	86	14.2	35200	181
1428.57	350	6	104.17	4800	87	13.59	36800	182
1250	400	7	100	5000	88	13.02	38400	183
1111.11	450	8	96.15	5200	89	12.5	40000	184
1000	500	9	92.59	5400	90	12.02	41600	185
909.9	550	10	89.29	5600	91	11.57	43200	186
833.33	600	11	86.21	5800	92	11.16	44800	187
769.23	650	12	83.33	6000	93	10.78	46400	188
714.29	700	13	80.65	6200	94	10.42	48000	189
666.67	750	14	78.13	6400	95	10.08	49600	190
625	800	15	73.53	6800	112	9.77	51200	191
588.24	850	16	69.44	7200	113	9.19	54400	208
555.56	900	17	65.79	7600	114	8.68	57600	209
526.32	950	18	62.5	8000	115	8.22	60800	210
500	1000	19	59.52	8400	116	7.44	67200	212
476.19	1050	20	56.82	8800	117	7.1	70400	213
454.55	1100	21	54.53	9200	118	6.79	73600	214
434.78	1150	22	52.08	9600	119	6.51	76800	215
416.67	1200	23	50	10000	120	6.25	80000	216
400	1250	24	40.08	10400	121	6.01	83200	217
384.62	1300	25	46.3	10800	122	5.79	86400	218
370.37	1350	26	44.64	11200	123	5.58	89600	219
357.14	1400	27	43.1	11600	124	5.39	92800	220
344.83	1450	28	41.67	12000	125	5.21	96000	221
333.33	1500	29	40.32	12400	126	5.04	99200	222

322.58	1550	30	39.06	12800	127	4.88	102400	223
312.5	1600	31	36.76	13600	144	4.6	108800	240
294.12	1700	48	34.72	14400	145	4.34	115200	241
277.78	1800	49	32.89	15200	146	4.11	121600	242
263.16	1900	50	31.25	16000	147	3.91	128000	243
250	2000	51	29.76	16800	148	3.72	134400	244
238.1	2100	52	28.41	17600	149	3.55	140800	245
227.27	2200	53	27.17	18400	150	3.4	147200	246
217.39	2300	54	26.04	19200	151	3.26	153600	247
208.33	2400	55	25	20000	152	3.13	160000	248
200	2500	56	24.04	20800	153	3	164400	249
192.31	2600	57	23.15	21600	154	2.89	172800	250
185.19	2700	58	22.32	22400	155	2.79	179200	251
178.57	2800	59	21.55	23200	156	2.69	185600	252
172.41	2900	60	20.83	24000	157	2.6	192000	253
166.67	3000	61	20.16	24800	158	2.52	198400	254
147.06	3400	80	19.53	25600	159	2.44	204800	255

附录2 常见问题

随着当前系统设计复杂度的提升，调试过程中碰到问题的几率大大增加。在各种问题中我们需要分析问题的根源和实质，以便去正确的解决。Multi ARM 与目标板的连接仅限于 JTAG 接口，两者之间的耦合度已降到最小，不会对目标板的逻辑和电气性能产生影响。

问题： **Multi-ICE Server** 程序启动失败

原因： BIOS 中的并口类型设置不正确，请修改后重试。一般推荐用户选择 EPP 类型。

问题： **Multi-ICE Server** 程序启动失败，出现 TCP/CP 堆栈出错

原因： 开发 PC 机没有连接网络或没有安装网络服务程序。如果不需要使用网络服务功能，可以在 Multi-ICE Server 的“Settings”——>“Start-up Options”菜单中关闭该功能。如果程序因为该原因自动关闭而造成无法修改选项设置，可以在 Multi-ICE Server 的安装目录下找到一个名为“”的文件并双击运行，则在以后的 Multi-ICE Server 程序运行中就会跳过网络协议堆栈的检测。

问题： **Multi-ICE Server** 程序自动识别目标失败

原因： 1、目标内核不能识别。在本手册前面章节中详细列出了目前版本的 Multi ARM 所支持的内核种类。如果能够检测到正确的 TAP 控制器并建立起通信，但不能返回内核 ID 号，将在屏幕上显示“UNKNOWN”，这时调试程序不能正常工作。用户需要从厂家得知正确的 ARM 芯片内核类型，并新建一个配置文件来进行手动配置。配置方法参见第 4.1.4 小节内容。

2、信号问题。最常遇到的信号问题包括：

Reset 信号没有上拉电阻（Reset 包括 nTRST 和 nSRST），这两个信号应在目标板上用小于 10K 的电阻上拉。

TCK 信号的频率太高。该频率受制于 TAP 控制器，目标板布线和连接电缆等因素，有些 TAP 控制器只支持到 1MHz 速率，这时候需要调整 TCK 频率设置。

问题： 同调试程序连接时提示“Can't stop processor”

原因： 1、在当前环境下 JTAG 的时钟频率过高，请尝试降低 TCK 的频率

2、如果 Multi-ICE Server 程序是手动配置的，则需要检查：

配置错误：处理器类型、连接次序等设置错误

JTAG 问题：进行手动配置时，因先确保在自动时能够检测到“UNKNOWN”，这样说明 JTAG 接口通信正确。

3、ARM 芯片接口的“DBGEN”信号被不正确的拉低，如果该信号有片外接口，请检查。

4、目标处理器的时钟不正确。

问题： 同调试程序连接时出现“Data Abort”提示

原因： 1、当调试程序启动时，处理器将停在当前 PC 值指向的地方，同时读取该 PC 值周围一部分地址的存储器值。如果这些地址正好没有被使用或指向空，就会出现上述提示，此情况属于正常范围。

2、如果使用了存储器（RAM 或 ROM），出现这种问题表明在存储器访问上可能存在数据错误，请检查存储器。

3、当用户下载程序到目标板时，下载的目标地址是在编译器中指定的，如果编译器里设置的目标地址与目标板上可写存储器的物理地址不能对应，也会出现上述提示。因为

调试程序试图往错误的物理地址写入用户程序而出错，这时需要检查并重新设置编译器选项。

4、JTAG 的时钟速率过快。

问题： 在调试过程中经常停止，有时候甚至进不了调试程序

原因： 在排除了调试程序本身的问题后，这一类带有随机性的问题可以考虑是硬件时序或逻辑问题引起的，最主要的可能性是 nTRST 和系统 Reset 信号不可靠。Reset 信号的不可靠包含两方面：正常工作时要求稳定的保持高电平；复位状态时要求有足够时间的低电平。Reset 电路因为简单通常会被忽略，但事实表明很多貌似奇怪的问题最后都归结在这里，无论是从调试的角度还是最后稳定工作的要求，我们都建议用户采用一个好的 Reset 电路（或是 IC）来代替简单的 R—C 电路。

问题： 调试程序出现 “**Hardware interface timeout**” 提示

原因： 1、目标板与 Multi ARM 的连接断开或 Multi ARM 与 PC 之间的连接中断
2、Multi ARM 供电中断或供电不足
3、Multi ARM 与目标板的通信停顿。一般都是因为打开了 RTCK 选项，但是检测不到合适的 RTCK 信号。此时用户可以不使用 RTCK 功能或检查该信号的布线。

问题： 调试程序提示 “**Unable to set breakpoints on exception vecors**”

原因： 地址映射表中的起始地址对应的存储器类型是 ROM，Multi ARM 不能在 ROM 上设置断点。