



Hi3518A/Hi3518C/Hi3518E/Hi3516C U-boot 移植应用 开发指南

文档版本 01

发布日期 2014-02-26

版权所有 © 深圳市海思半导体有限公司 2012-2014。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址： 深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址： <http://www.hisilicon.com>

客户服务电话： +86-755-28788858

客户服务传真： +86-755-28357515

客户服务邮箱： support@hisilicon.com



前 言

概述

本文档主要介绍本在 Hi3518A/Hi3518C/Hi3518E/Hi3516C 单板上如何移植和烧写 U-boot（Hi3518A/Hi3518C/ Hi3518E/Hi3516C 单板的 Bootloader）的相关操作及如何使用 ARM 调试工具。

产品版本

与本文档相对应的产品版本如下。

| 产品名称 | 产品版本 |
|------------|------|
| Hi3518A 芯片 | V100 |
| Hi3518C 芯片 | V100 |
| Hi3518E 芯片 | V100 |
| Hi3516C 芯片 | V100 |

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。



| 修订日期 | 版本 | 修订说明 |
|------------|-------|-------------------------------|
| 2014-02-26 | 01 | 补充 Hi3518E 的信息。 |
| 2012-12-25 | 00B30 | 补充 Hi3516C 的信息。 |
| 2012-11-25 | 00B20 | 3.2 初始化脚本及 U-boot 中修改初始化脚本名称。 |
| 2012-08-30 | 00B10 | 第 2 次版本发布。 |
| 2012-08-15 | 00B01 | 第 1 次临时版本发布。 |



目 录

| | |
|---|----|
| 前 言..... | i |
| 1 概述..... | 1 |
| 1.1 概述..... | 1 |
| 1.2 U-boot 目录结构 | 1 |
| 2 移植 U-boot..... | 3 |
| 2.1 概述..... | 3 |
| 2.2 修改 Flash 驱动 | 3 |
| 2.3 编译 U-boot | 3 |
| 2.4 配置 DDR 存储器 | 4 |
| 2.5 配置管脚复用..... | 4 |
| 2.6 生成最终使用的 U-boot 镜像..... | 4 |
| 3 烧写 U-boot..... | 5 |
| 3.1 概述..... | 5 |
| 3.2 初始化脚本及 U-boot..... | 5 |
| 3.3 通过 RealView Debugger4.0 烧写 U-boot | 5 |
| 3.3.1 启动 RealView Debugger4.0 | 6 |
| 3.3.2 加载初始化脚本..... | 6 |
| 3.3.3 打开映像文件..... | 7 |
| 3.4 两种 Flash 的 U-boot 烧写方法 | 8 |
| 3.4.1 SPI Flash 烧写方法 | 8 |
| 3.4.2 NAND Flash 烧写方法 | 8 |
| 4 如何使用 ARM 调试工具 | 9 |
| 4.1 概述..... | 9 |
| 4.2 ARM 调试工具简介 | 9 |
| 4.2.1 RealView -ICE 仿真器 | 9 |
| 4.2.2 RealView Debugger4.0 | 9 |
| 4.3 使用 ARM 调试工具..... | 10 |



插图目录

| | |
|---------------------------------------|----|
| 图 3-1 RealView Debugger 窗口 | 6 |
| 图 3-2 载入 U-boot 初始化文件后串口终端命令窗口提示..... | 7 |
| 图 4-1 RealView Debugger 窗口 | 11 |
| 图 4-2 Connetct to Target 窗口 | 11 |
| 图 4-3 Choose RealView-ICE | 12 |
| 图 4-4 RealView Debugger 信息提示框 | 12 |
| 图 4-5 RealView Debugger 信息提示框 2 | 13 |
| 图 4-6 Connect to ARM9 | 13 |



表格目录

| | |
|----------------------------|---|
| 表 1-1 U-boot 的主要目录结构 | 1 |
|----------------------------|---|



1 概述



说明

Hi3518A、Hi3518C、Hi3518E 及 Hi3516C 的原理和操作基本一致。与芯片相关的操作本文后面均以 Hi3518A 为例进行说明。

1.1 概述

Hi3518A 单板的 Bootloader 采用 U-boot。当选用的外围芯片的型号与单板上外围芯片的型号不同时，需要修改 U-boot 配置文件，主要包括存储器配置、管脚复用。

1.2 U-boot 目录结构

U-boot 的主要目录结构如表 1-1 所示，详细目录说明请阅读 U-boot 目录下的 README 文档。

表1-1 U-boot 的主要目录结构

| 目录名 | 描述 |
|-----------------|-------------------------------|
| arch | 各种芯片架构的相关代码、U-boot 入口代码。 |
| board | 各种单板的相关代码，主要包括存储器驱动等。 |
| board/hi3518/ | 海思单板的相关代码。 |
| lib_XXX | 各种体系结构的相关代码，如 ARM、MIPS 的通用代码。 |
| include | 头文件。 |
| include/configs | 各种单板的配置文件。 |
| common | 各种功能（命令）实现文件。 |
| drivers | 网口、Flash、串口等的驱动代码。 |
| net | 网络协议实现文件。 |



| 目录名 | 描述 |
|-----|-----------|
| fs | 文件系统实现文件。 |



2 移植 U-boot

2.1 概述

在 Hi3518A 单板上所选用的外围芯片型号如下：

- DDR SDRAM: K4B1G1646E-HCH9
- NAND Flash: TC58NVG1S3ETA00
- SPI Flash: MX25L12835E

如果选用的外围芯片不是以上型号时，需要适当修改 SDK 中的“osdrv/tools/pc_tools/uboot_tools”目录下的配置表格，对应的单板才能正常运行。

2.2 修改 Flash 驱动

对于 00.hardware/board/documents_cn 路径下的 Flash 兼容性列表中的器件，U-boot 都可以支持，不需要做特别移植。用户使用列表外的器件时，移植工作需要特殊支持。

2.3 编译 U-boot

当所有以上移植步骤完成后，就可以编译 U-boot，操作如下：

```
make ARCH=arm CROSS_COMPILE=arm-hisiv100nptl-linux- hi3518a_config //然后  
选择对应的单板  
make ARCH=arm CROSS_COMPILE=arm-hisiv100nptl-linux-
```

编译成功后，将在 U-boot 目录下生成 u-boot.bin。



说明

- 编译 Hi3518C 的 U-boot 请使用配置文件 hi3518c_config；编译 Hi3518E 的 U-boot 请使用配置文件 hi3518e_config；编译 Hi3516C 的 U-boot 请使用配置文件 hi3516c_config。
- CROSS_COMPILE=arm-hisiv100nptl-linux- 指定了编译工具链为 arm-hisiv100nptl-linux-，如果需要使用 arm-hisiv200-linux-工具链，请指定：CROSS_COMPILE= arm-hisiv200-linux-
- 这一步生成的 u-boot.bin 只是一个中间件，并不是最终在单板上执行的 U-boot 镜像。

2.4 配置 DDR 存储器

在 Windows 下打开 SDK 中的“osdrv/tools/pc_tools/uboot_tools”目录下的配置表格。当选用不同的 DDR SDRAM 时，需要针对不同器件的特性，对配置工作表中的相关标签页进行修改。

2.5 配置管脚复用

如果管脚复用有变化，还需要对配置表格中的相关标签页进行修改。

2.6 生成最终使用的 U-boot 镜像

完成配置表格的修改后，保存表格。单击表格第一个标签页上的按钮“Generage reg bin file”，生成临时文件 reg_info.bin。

将临时文件 reg_info.bin 和编译 u-boot 得到的 u-boot.bin 都拷贝到 SDK 中的“osdrv/tools/pc_tools/uboot_tools 目录下，执行命令：

```
mkboot.sh reg_info.bin u-boot-200MHZ.bin
```

其中 u-boot-200MHZ.bin 就是能够在单板上运行的 U-boot 镜像。



3 烧写 U-boot

3.1 概述

如果待移植单板中已有 U-boot 运行，则可以通过串口或网口与服务器连接，直接更新 U-boot。

如果是第一次烧写，则需要通过网口使用 RVDS 工具进行烧写。由于芯片特性，必须有通过 RVDS 下载到单板的脚本，该脚本能够对存储器和芯片进行初始化。在 Hi3518A SDK 中提供了相应的初始化脚本，当选用了不同的外围芯片，则需要重新配置初始化脚本才能使用。

3.2 初始化脚本及 U-boot

初始化脚本是实现 U-boot 相同的初始化内存控制器的功能。在生成的 SDK 目录的“package/images”目录下，有 u-boot-200MHZ.bin 文件，在“osdrv/tools/pc_tools/uboot_tools”目录包含以下初始化程序文件：

- V1.0_Hi3518AV100_MDDRC_ASIC_1to2_440MHz_200MHz_400MHz_SigI CH_Ecc Off_ExOn_init_script.log
- V1.0_Hi3518CV100_MDDRC_ASIC_1to2_440MHz_200MHz_400MHz_SigI CH_Ecc Off_ExOn_init_script.log
- V1.0_Hi3518EV100_MDDRC_ASIC_1to2_440MHz_200MHz_400MHz_SigI CH_Ecc Off_ExOn_init_script.log
- V1.0_Hi3516CV100_MDDRC_ASIC_1to2_440MHz_220MHz_440MHz_SigI CH_Ecc Off_ExOn_init_script.log



说明

osdrv/tools/pc_tools/uboot_tools”目录的四个初始化程序文件分别适用于 Hi3518A、Hi3518C、Hi3518E 和 Hi3516C。

3.3 通过 RealView Debugger4.0 烧写 U-boot

通过 RealView Debugger4.0 烧写 U-boot 时，要按照以下顺序进行：

- 启动 RealView Debugger4.0



- 加载初始化脚本
- 打开映像文件

3.3.1 启动 RealView Debugger4.0



说明

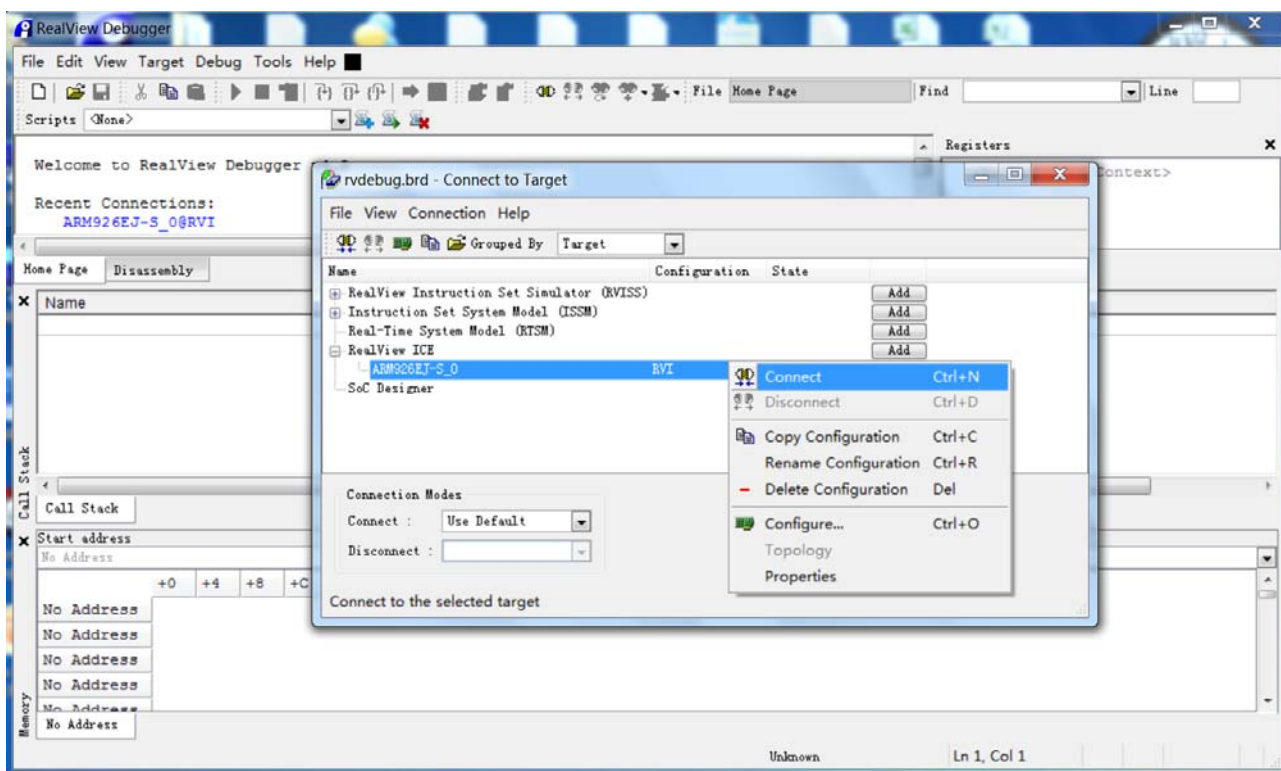
关于使用 RealView Debugger4.0 工具请参见“4 如何使用 ARM 调试工具”。

启动 RealView Debugger4.0 的步骤如下：

1. 运行 RealView Debugger4.0 应用程序。
2. 查找 RealView ICE 仿真器。
3. 连接目标板的 ARM9 芯片。如图 3-1 所示。
4. 双击“ARM926EJ-S_0”连接单板。

----结束

图3-1 RealView Debugger 窗口



3.3.2 加载初始化脚本

打开初始化脚本并运行的步骤如下：

1. 选择“Tools > Include Commands from File ...”



2. 打开“Include Commands from File”对话框，找到名字为 V1.0_Hi3518AV100_MDDRC_ASIC_1to2_440MHz_200MHz_400MHz_Sig1CH_EccOff_ExOn_init_script.log”的映像文件，单击“打开”，把初始化脚本加载到存储器中。
3. 单击图 3-1 中的“▶”（RUN/F5）按钮，运行。

----结束

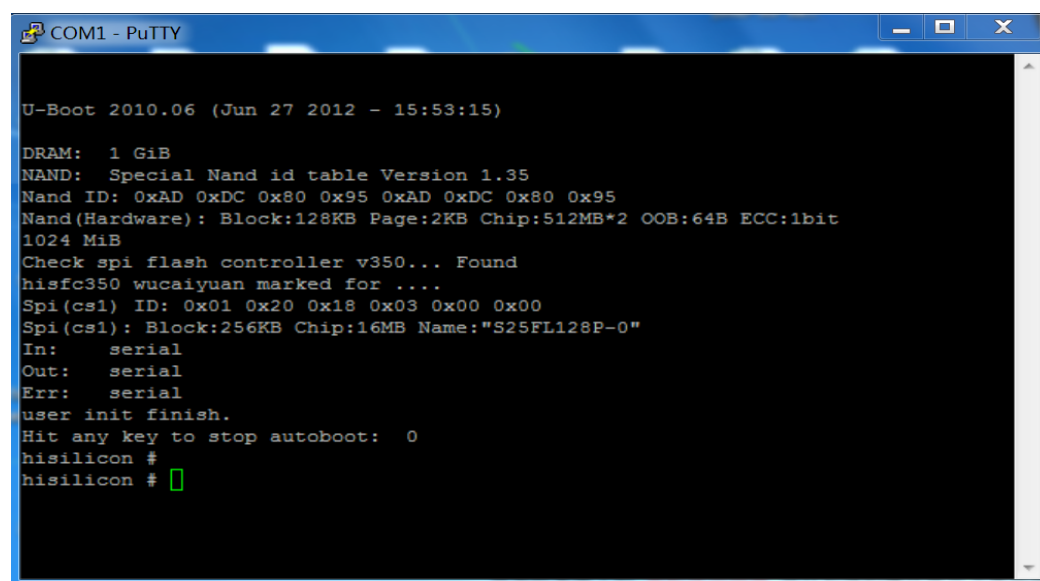
3.3.3 打开映像文件

再次打开映像文件的步骤如下：

1. 选择图 3-1 中的“Target > Load image...”。
2. 打开“Load Image”对话框，找到名字为“u-boot”的映像文件，单击“打开”，把映像文件加载到存储器中。
3. 再次单击“▶”（RUN/F5），此时在与单板相连的串口终端显示如图 3-2 所示。
4. 此时系统已经进入 U-boot 环境在 DDR 中执行，根据启动的 Flash 类型可通过“3.4 两种 Flash 的 U-boot 烧写方法”中的方法将 U-boot 下载到 Flash 中。

----结束

图3-2 载入 U-boot 初始化文件后串口终端命令窗口提示



```
COM1 - PuTTY
U-Boot 2010.06 (Jun 27 2012 - 15:53:15)

DRAM:  1 GiB
NAND:  Special Nand id table Version 1.35
Nand ID: 0xAD 0xDC 0x80 0x95 0xAD 0xDC 0x80 0x95
Nand(Hardware): Block:128KB Page:2KB Chip:512MB*2 OOB:64B ECC:1bit
1024 MiB
Check spi flash controller v350... Found
hisfc350 wucaiyuan marked for ....
Spi(cs1) ID: 0x01 0x20 0x18 0x03 0x00 0x00
Spi(cs1): Block:256KB Chip:16MB Name:"S25FL128P-0"
In:     serial
Out:    serial
Err:    serial
user init finish.
Hit any key to stop autoboot:  0
hisilicon #
hisilicon #
```

烧写 U-boot 完成后，程序自动停止运行，然后关闭 RealView Debugger 软件。这时复位或者重新上电，启动单板，U-boot 就能自行启动。



3.4 两种 Flash 的 U-boot 烧写方法

3.4.1 SPI Flash 烧写方法

SPI Flash 烧写方法如下：

1. 在内存中运行起来之后在串口终端中输入：

```
hisilicon# mw.b 0x82000000 ff 0x100000 /* 对内存初始化*/  
hisilicon# tftp 0x82000000 u-boot-200MHZ.bin /*U-boot下载到内存*/  
hisilicon# sf probe 0 /*探测并初始化SPI flash*/  
hisilicon# sf erase 0x0 0x100000 /*擦除 1M大小*/  
hisilicon# sf write 0x82000000 0x0 0x100000 /*从内存写入SPI Flash*/
```

2. 上述步骤操作完成后，重启系统可以看到 U-boot 烧写成功。

----结束

3.4.2 NAND Flash 烧写方法

NAND Flash 烧写方法如下：

1. 在内存中运行起来之后在串口终端中输入：

```
hisilicon# nand erase 0 100000 /*擦除 1M大小*/  
hisilicon# mw.b 0x82000000 ff 100000 /* 对内存初始化*/  
hisilicon# tftp 0x82000000 u-boot-200MHZ.bin /*U-boot下载到内存*/  
hisilicon# nand write 0x82000000 0 100000 /*从内存写入NAND Flash*/
```

2. 重启系统可以看到 U-boot 烧写成功。

----结束



4 如何使用 ARM 调试工具

4.1 概述

本章介绍了关于 ARM 处理器调试用到的调试工具的使用方法，调试工具包括：

- [RealView -ICE 仿真器](#)
- [RealView Debugger4.0](#)

4.2 ARM 调试工具简介

4.2.1 RealView -ICE 仿真器

RealView -ICE 仿真器是一款高性能实时仿真器，该款仿真器支持 ARM 内核结构（如 ARM7、ARM9 Xscale、ARM11、ARM_CORTEXA9）的处理器，是进行基于 ARM 开发调试的必备工具。

RealView -ICE 仿真器的特点如下：

- 支持内含 Embedded ICETM logic 的 ARM 内核芯片，同时也支持尚在开发中的 ARM 内核芯片，包括 ARM7、ARM710、ARM720、ARM740、ARM9、ARM920、ARM10、ARM11 等 ARM 内核调试。
- 支持目标系统的电源从 1.8V~5V 自适应。
- 通过网口连接仿真器，用户可以轻松修改寄存器、存储器，设置硬件断点，增加观察窗口等。可以在二进制文件的开始位置或者结束位置保存自定义的数据和调色板。

4.2.2 RealView Debugger4.0

RealView Debugger4.0 是由 ARM 公司提供的软件开发调试软件，为软件开发人员开发高质量的 ARM 代码。RealView Debugger 提供了如下便利：

- 支持简单和复杂的断点设置。
- 提供观察窗口，可查看变量的变化情况。
- 支持所有新的和已经存在的 ARM 处理器。



- 增强的 Windows 管理模式。
- 增强的数据显示、修改和编辑。
- 提供完备的命令行接口。

4.3 使用 ARM 调试工具

要使用 Multi-ICE 进行程序调试或者向开发板烧写 U-boot 程序，首先必须启用 Multi-ICE server 程序查找到当前硬件连接的 ARM 芯片，然后才能通过 RealView Debugger 调试软件进行程序调试或者向开发板烧写 U-boot 程序。

关于使用 ARM 调试工具的更详细描述请参见 ARM 公司提供的文档。下面介绍如何使用 RealView Debugger。

1. 安装 ARM developer Suite v4.0。ARM developer Suite v4.0 是由 ARM 公司提供的 RealView Debugger 安装程序。安装前，请先阅读 ARM 的相关文档。
2. 运行 RealView Debugger。运行后，RealView Debugger 启动如图 4-1 所示。
3. 连接 RealView-ICE。单击“Selects targets to connect to”，弹出如图 4-2 所示的窗口。单击 RealView-ICE 后的 add，查找并连接仿真器，如图 4-3 所示。
4. 连接 ARM9。单击图 4-4 中的“Auto Configure”按钮，可扫描到 ARM9，出现如图 4-5 所示的窗口。然后保存并关闭，在 RealView ICE 下面就能看到目标 ARM926EJ-S_0，再双击连接就可以进行调试，如图 4-6 所示。

----结束



图4-1 RealView Debugger 窗口

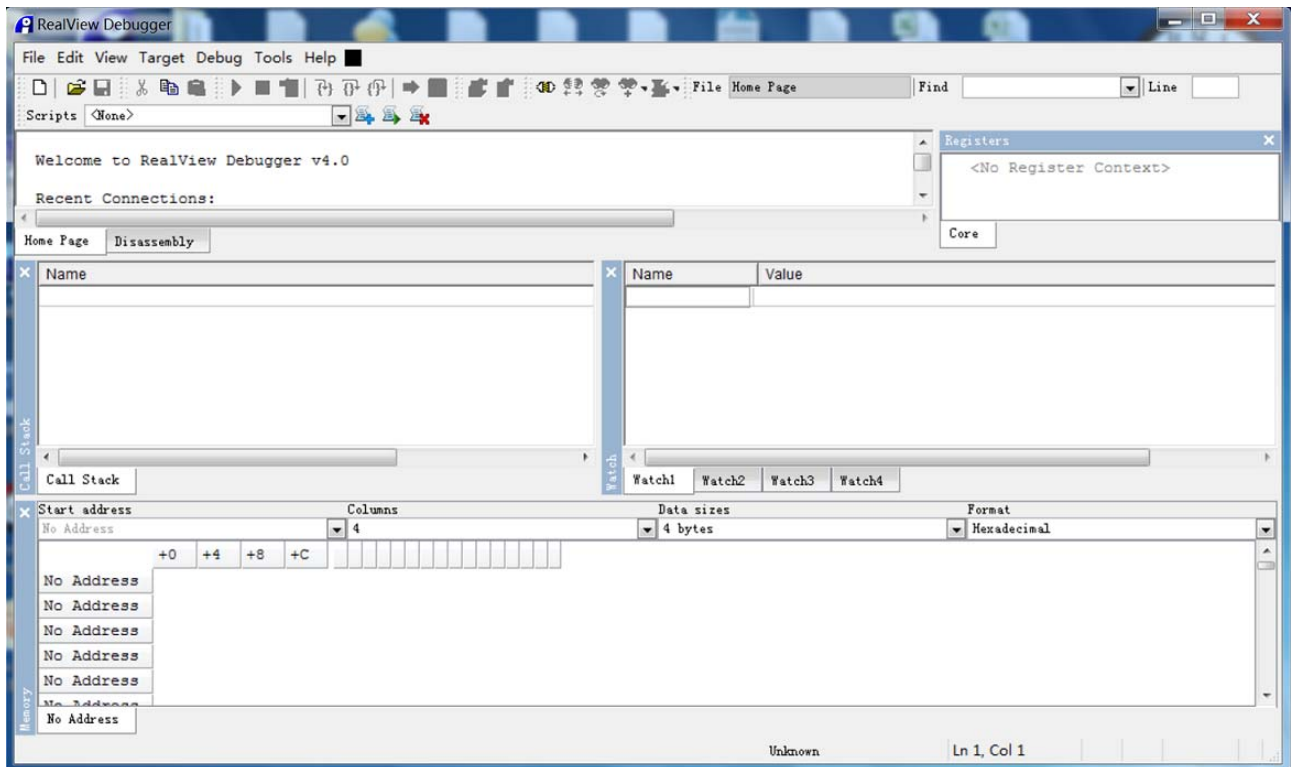


图4-2 Connect to Target 窗口

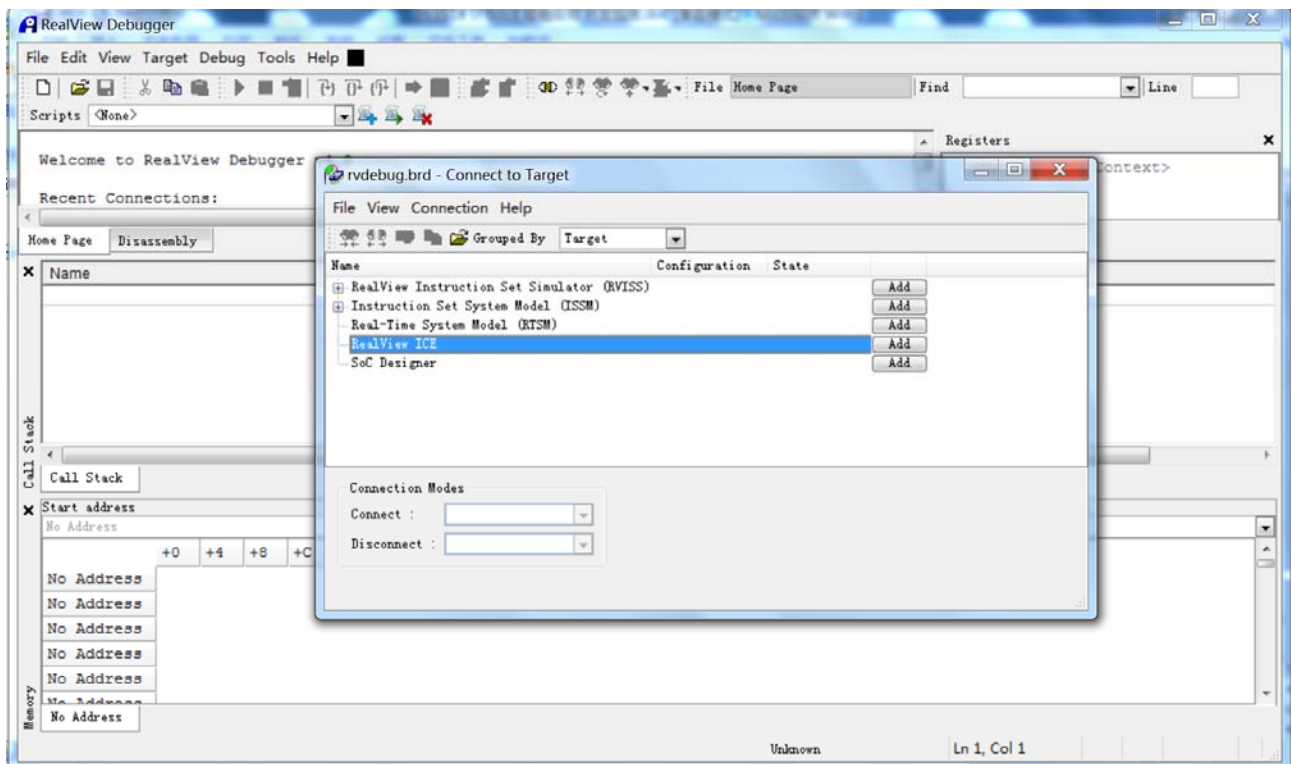




图4-3 Choose RealView-ICE

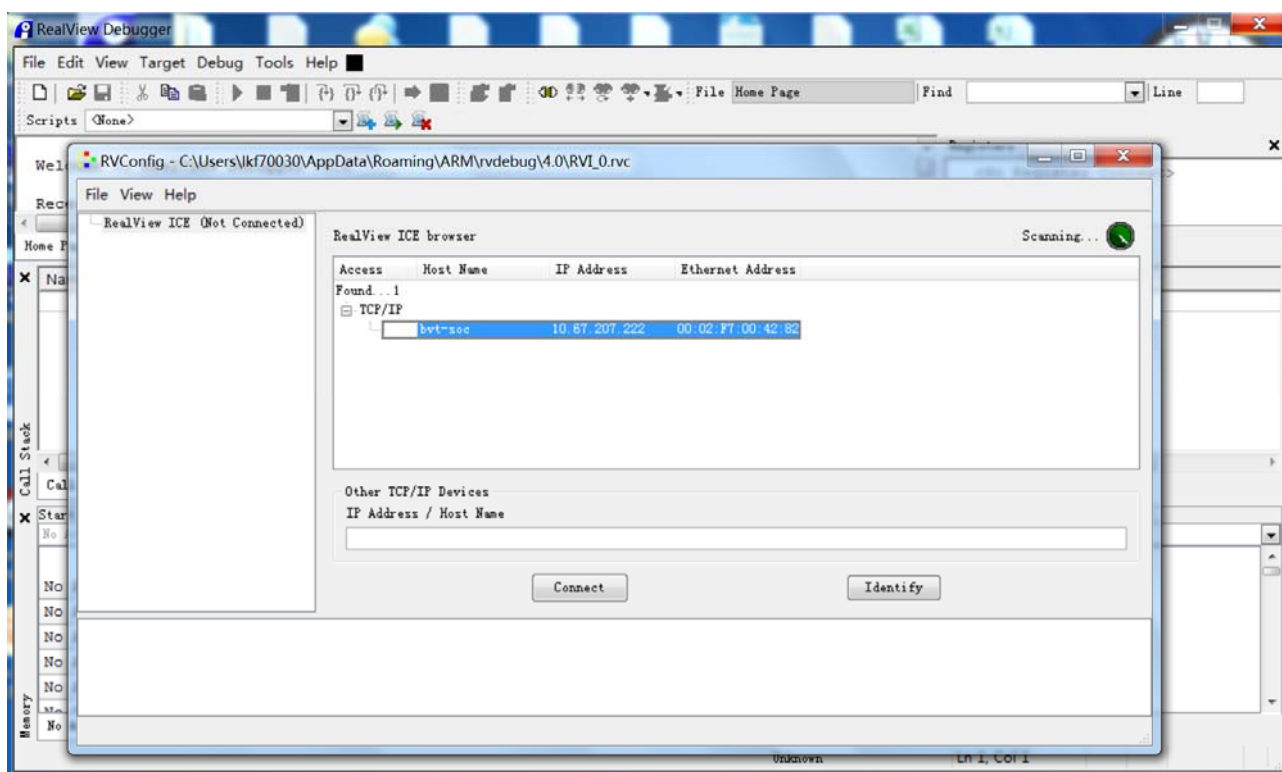


图4-4 RealView Debugger 信息提示框

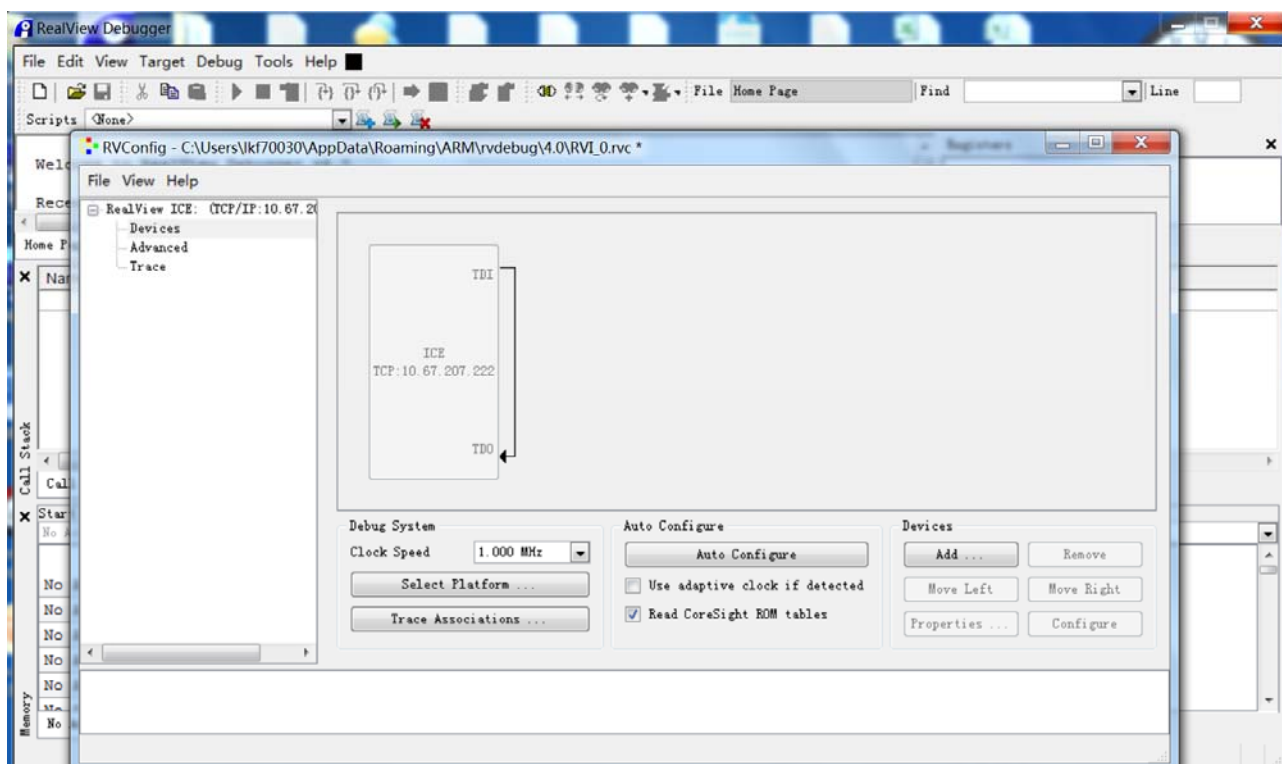




图4-5 RealView Debugger 信息提示框 2

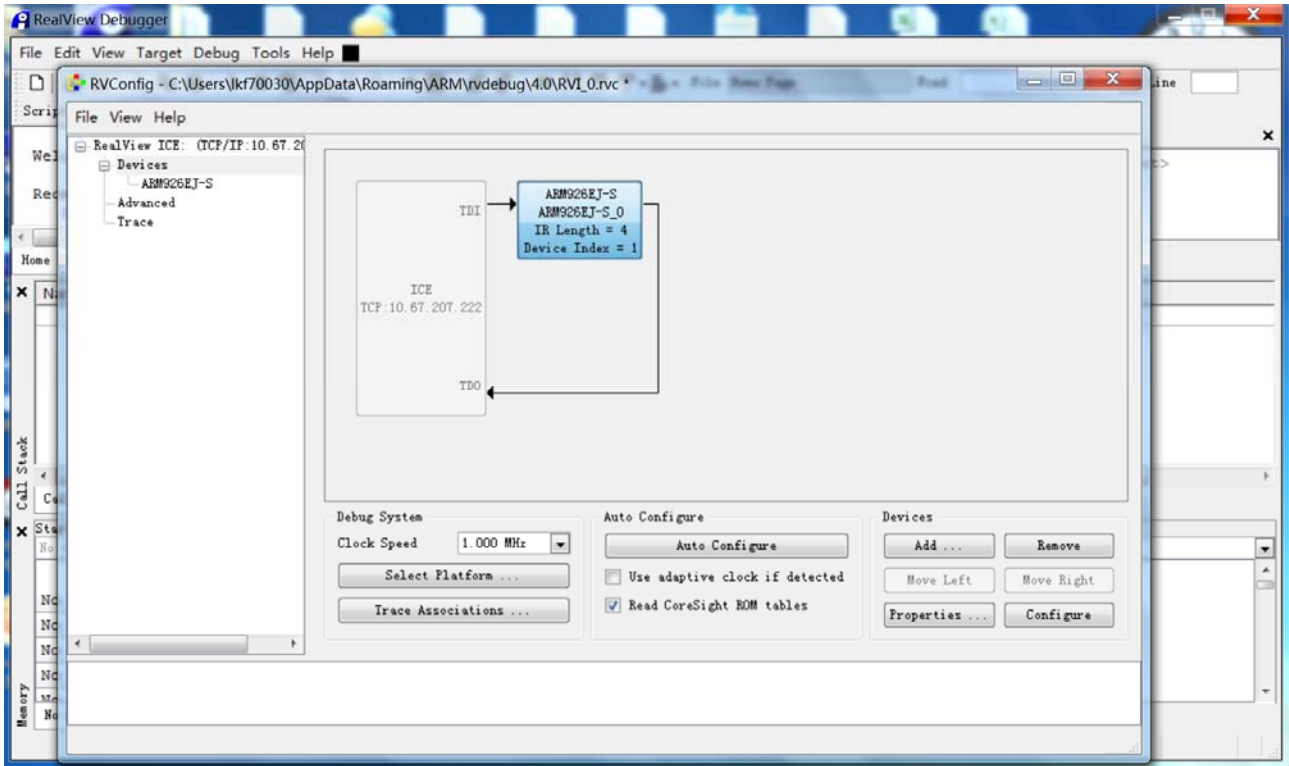


图4-6 Connect to ARM9

