



Hi3520 U-boot 移植应用

Application Notes

文档版本 00B02

发布日期 2009-09-30

版权所有 © 深圳市海思半导体有限公司 2009。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



目 录

前 言.....	1
1 概述.....	1-1
1.1 概述.....	1-1
1.2 U-boot 目录结构.....	1-1
2 移植 U-boot.....	2-1
2.1 概述.....	2-1
2.2 创建目录和文件.....	2-1
2.2.1 拷贝修改参考代码.....	2-1
2.2.2 增加新的配置文件.....	2-2
2.2.3 修改 Makefile.....	2-2
2.3 配置 DDR 存储器.....	2-2
2.4 修改 Flash 驱动.....	2-3
2.5 编译 U-boot.....	2-3
3 烧写 U-boot.....	3-1
3.1 概述.....	3-1
3.2 初始化程序及 U-boot.....	3-1
3.3 通过 RealView Debugger 烧写 U-boot.....	3-1
3.3.1 启动 RealView Debugger.....	3-1
3.3.2 打开映象文件并运行.....	3-2
3.3.3 再次打开映象文件.....	3-3
3.4 两种 Flash 的 U-boot 烧写方法.....	3-4
3.4.1 NOR Flash 烧写方法.....	3-4
3.4.2 NAND Flash 烧写方法.....	3-4
4 如何使用 ARM 调试工具.....	4-1
4.1 概述.....	4-1
4.2 ARM 调试工具简介.....	4-1
4.2.1 RealView -ICE 仿真器.....	4-1
4.2.2 RealView Debugger.....	4-1
4.3 使用 ARM 调试工具.....	4-2



插图目录

图 3-1 RealView Debugger 窗口	3-2
图 3-2 设置 ITCM Regions.....	3-3
图 3-3 载入 U-boot 初始化文件后超级终端命令窗口提示.....	3-4
图 4-1 RealView ICE Config IP 窗口	4-3
图 4-2 RealView Debugger 窗口	4-4
图 4-3 Connetct to Target 窗口	4-5
图 4-4 Choose RealView-ICE	4-6
图 4-5 RealView Debugger 信息提示框	4-7



表格目录

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



前言

概述

本文档主要介绍本在 Hi3520 单板上如何移植和烧写 U-boot (Hi3520 单板的 Bootloader) 的相关操作及工具使用。

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
Hi3520 H.264 编解码处理器	V100

读者对象



本文档主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

约定

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 注意	表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 说明	表示是正文的附加信息，是对正文的强调和补充。



通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用 黑体 。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
“Terminal Display”格式	“Terminal Display”格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。
“”	用双引号表示文件路径。如“C:\Program Files\Huawei”。

命令行格式约定

格式	意义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用 加粗 字体表示。
<i>斜体</i>	命令行参数（命令中必须由实际值进行替代的部分）采用 <i>斜体</i> 表示。
[]	表示用“[]”括起来的部分在命令配置时是可选的。
{ x y ... }	表示从两个或多个选项中选取一个。
[x y ...]	表示从两个或多个选项中选取一个或者不选。
{ x y ... } *	表示从两个或多个选项中选取多个，最少选取一个，最多选取所有选项。
[x y ...] *	表示从两个或多个选项中选取多个或者不选。

表格内容约定

内容	说明
-	表格中的无内容单元。
*	表格中的内容用户可根据需要进行配置。



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
2009-09-30	00B02	优化部分描述。
2009-08-31	00B01	第一次发布。



1 概述

1.1 概述

Hi3520 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，需要修改系统中相关硬件的驱动代码，主要包括存储器配置、Flash 驱动。

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
Cpu	各种 CPU 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、Flash 驱动等。
Board/hi3520v100/	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、串口等的驱动代码。
drivers/hisilicon	海思芯片部分模块的驱动代码。
Net	网络协议实现文件。
Fs	文件系统实现文件。



2 移植 U-boot

2.1 概述

在 Hi3520 单板上所选用的外围芯片型号如下：

- DDR SDRAM: MT47H64M16-25E
- NAND Flash: HY27UG088G5M_TPCB

如果选用的外围芯片不是以上型号时，需要适当修改 U-boot 代码，对应的单板才能正常运行。

Hi3520 含一个主 ARM 核（ARM1176）和一个从 ARM 核（ARM926），本章描述针对主 ARM 核，从 ARM 核的 U-boot 移植与主 ARM 核类似。从 ARM 的 U-boot 是不需要移植的，可以直接使用 SDK 中发布的二进制文件。

2.2 创建目录和文件

创建待移植单板相关代码的目录和文件时，只需拷贝、修改相应的参考代码，不用手动编写所有代码。在本节的示例中，待移植单板的参考代码存放在 U-boot 的目录 board/和 include/asm-arm/中。

2.2.1 拷贝修改参考代码

在 Hi3520 SDK 中，U-boot 代码位于“source/os/u-boot-2008.10”目录中，需要先将“board/hi3520v100”目录下的代码拷贝到目录“board/hi3520v100-myboard”中，操作如下：

```
#cd sdk/source/os/u-boot-2008.10/  
#cp -a board/hi3520v100 board/hi3520v100-myboard //拷贝Hi3520代码到目录  
hi3520v100 -myboard
```

再对“board/hi3520v100-myboard”目录下的 u-boot.lds 文件内容进行如下修改：

```
board/hi3520v100-myboard/libhi3520v100.a (.text)
```



2.2.2 增加新的配置文件

在“include/configs/”目录下增加新的单板配置文件 hi35X0v100-myboard.h，操作如下：

```
#cp include/configs/hi3520v100_XXM.h include/configs/hi3520v100-  
myboard.h (XXM表示总线频率)
```

2.2.3 修改 Makefile

修改 Makefile 增加单板配置的编译选项。

在 U-boot 代码根目录下增加 hisilicon_board-myboard.mk 文件，在该文件中增加的配置操作如下：

```
hi3520v100-myboard _config:    unconfig  
@./mkconfig hi35X0v100-myboard arm arm1176 hi3520v100-myboard NULL  
hi3520v100
```

完成以上所有操作后，U-boot 代码已添加移植所需的框架，然后需要对驱动代码做相应的修改和配置。

2.3 配置 DDR 存储器

存储器配置文件为 hi35X0-myboard 目录下的 lowlevel_init.c，其主要功能是完成系统控制、DDR SDRAM 和 DDRCA、DDRCB、SMI 控制器的配置。当选用不同的 DDR SDRAM 时，其对应的电气特性、模式寄存器等需要相应的修改。在 U-boot 中将需要配置的参数以宏定义的形式在单板配置文件中定义，即在 hi3520v100-myboard.h 文件中有如下宏定义：

```
/* DDRC Config */  
#define CFG_DDRCA_BUSWITH DDRC_BUSWITH_32BITS  
#define CFG_DDRCA_BANK DDRC_CHIP_8BANK  
#define CFG_DDRCA_ROW CFG_DDRC_ROW_13  
#define CFG_DDRCA_COL CFG_DDRC_COL_10  
#define CFG_DDRCA_WR_DELAY DDR_WR_DELAY_5  
#define CFG_DDRCA_CAS_DELAY DDR_CAS_DELAY_6
```

针对不同的器件，进行以上修改后，DDR 初始化即完成。



注意

DDRCA 和 DDRCB 仅支持 DDR2-800 标准器件。



2.4 修改 Flash 驱动

U-boot 里已经包含支持 CFI（Common Flash Interface）接口标准的 Flash 驱动，不需要做特别移植。特殊用户使用非 CFI 接口 Flash 芯片，移植工作需要特殊支持。

2.5 编译 U-boot

当所有以上移植步骤完成后，就可以编译 U-boot，操作步骤如下：

```
make mrproper  
make hiconfig //然后选择对应的单板  
make
```

编译成功后，将在 U-boot 目录下生成 u-boot.bin。



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，具体操作请参见《Hi3520 Linux 开发环境 用户指南》。

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 下载到单板的应用程序，该程序能够对存储器和芯片进行初始化。在 Hi3520 SDK 中提供了相应的初始化程序，当选用了不同的外围芯片，则需要重新编译初始化程序才能使用。

3.2 初始化程序及 U-boot

初始化程序是实现 U-boot 相同的初始化内存控制器的功能。在生成的 SDK 目录的“pub/images/”目录下有 u-boot-hi3520v100_200M.bin 文件，“pub/images/u-boot-2008.10_200M”目录包含 hisiex-dramer 和 u-boot 两个初始化程序文件。

3.3 通过 RealView Debugger 烧写 U-boot

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

- 启动 RealView Debugger
- 打开映象文件并运行
- 再次打开映象文件

3.3.1 启动 RealView Debugger



说明

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

启动 RealView Debugger 的步骤如下：

步骤 1 运行 RealView Debugger 应用程序。



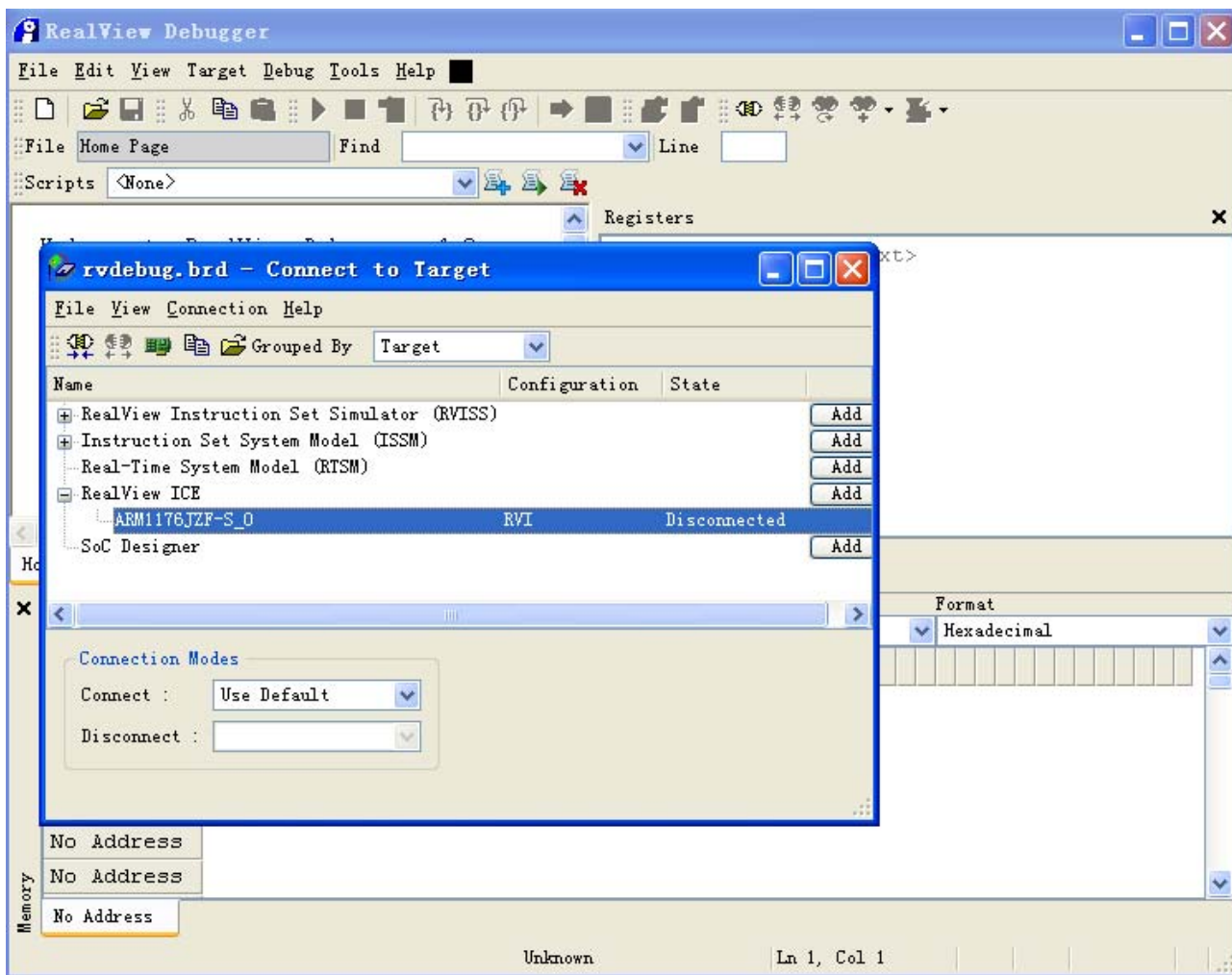
步骤 2 查找 RealView ICE 仿真器。

步骤 3 连接目标板的 ARM1176 芯片。如图 3-1 所示。

步骤 4 双击“ARM1176JZF...”连接单板。

----结束

图3-1 RealView Debugger 窗口



3.3.2 打开映像文件并运行

打开映像文件并运行的步骤如下：

步骤 1 修改 ITCM Regions 值为 0x0000000D，如图 3-2 所示。

步骤 2 选择“File > Load image...”。

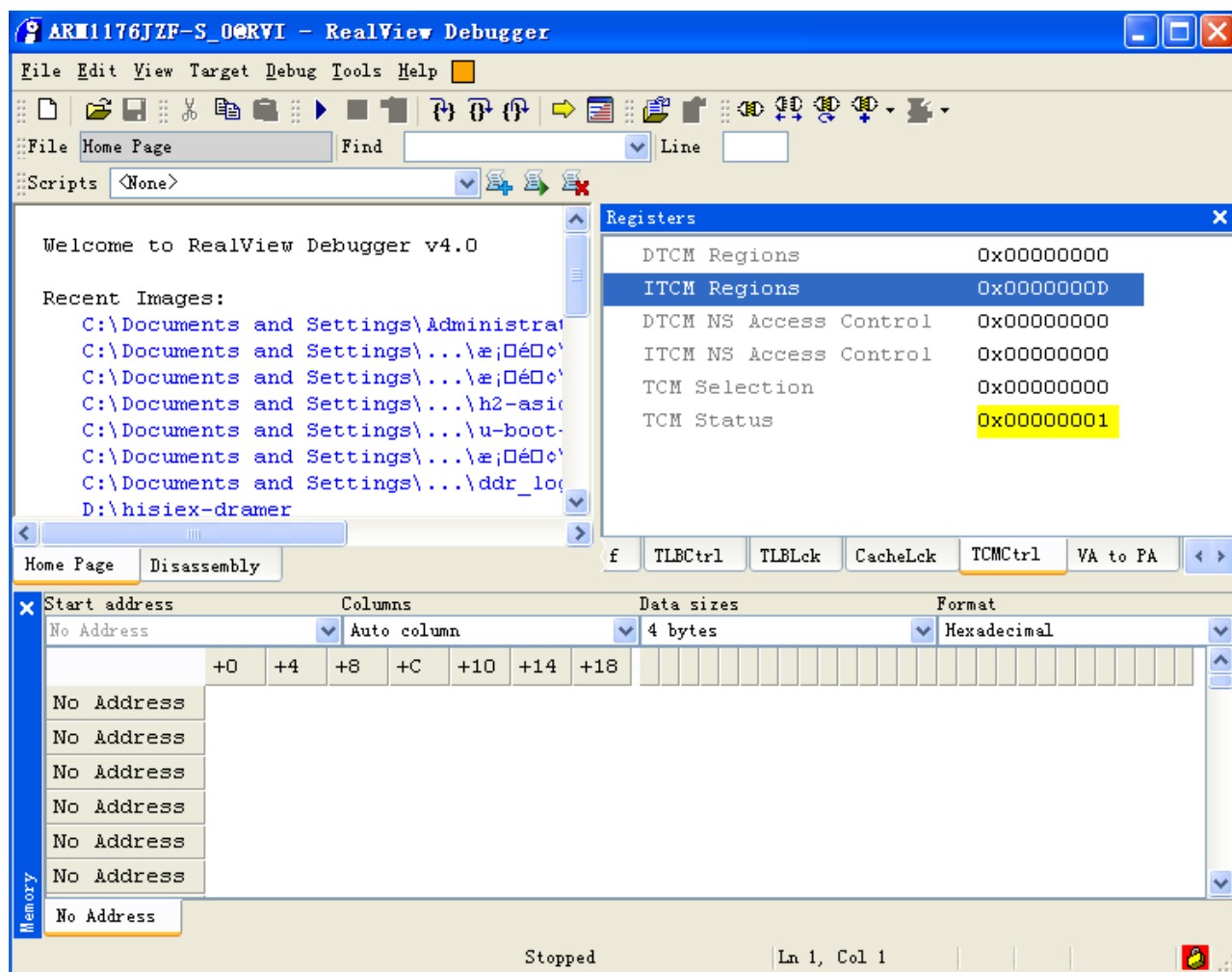
步骤 3 打开“Load Image”对话框，找到名字为“hisiex-dramer”的映像文件，单击“打开”，把映像文件加载到存储器中。



步骤 4 单击 “▶”（GO/F5）。

----结束

图3-2 设置 ITCM Regions



3.3.3 再次打开映象文件

再次打开映象文件的步骤如下：

步骤 1 选择 “File > Load image...”。

步骤 2 打开 “Load Image” 对话框，找到名字为 “u-boot” 的映象文件，单击 “打开”，把映象文件加载到存储器中。

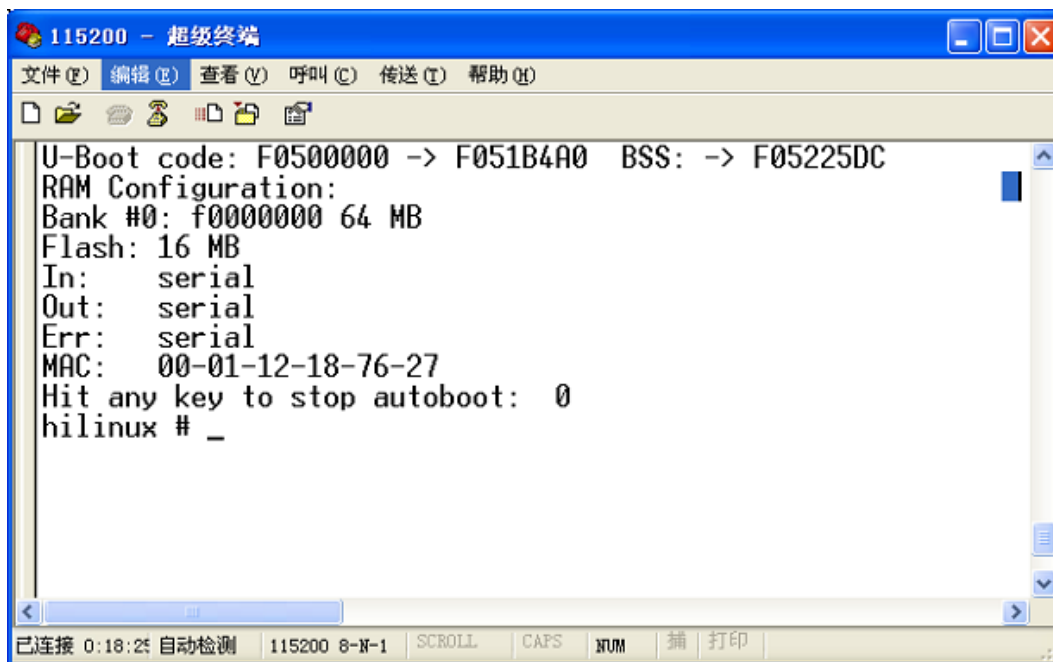
步骤 3 再次单击 “▶”（GO/F5），此时在与单板相连的超级终端显示如图 3-3 所示。

步骤 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3-3 载入 U-boot 初始化文件后超级终端窗口提示



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 NOR Flash 烧写方法

NOR Flash 烧写方法如下：

步骤 1 在内存中运行起来之后在超级终端中输入：

```
hisilicon# protect off 0x80000000 0x800fffff /*去除写保护*/  
hisilicon#erase 0x80000000 +0x100000 /*擦除Flash*/  
hisilicon#tftp 0x80000000 u-boot.bin /*拷贝内核到0x80100000位置*/
```

步骤 2 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：

步骤 1 在内存中运行起来之后在超级终端中输入：

```
hisilicon# nand erase 0 100000 /*擦除 1M大小*/
```




```
hisilicon# mw.b e1500000 ff 100000 /* 对内存初始化*/  
hisilicon# tftp e1500000 u-boot.bin /*U-boot下载到内存*/  
hisilicon# nand write e1500000 0 100000 /*从内存写入NAND Flash*/
```

步骤 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 Debugger](#)

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）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 Debugger

RealView Debugger 是由 ARM 公司提供的软件开发调试软件，为软件开发者开发高质量的 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 进行程序调试或者往开发板烧写 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往开发板烧写 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。

- 步骤 1 安装 ARM developer Suite v4.0。ARM developer Suite v2.0 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。
- 步骤 2 配置 RealView-ICE 网络参数。运行 RealView ICE Config IP 配置网络参数，如[图 4-1](#)所示。
- 步骤 3 运行 RealView Debugger。运行后，RealView Debugger 启动如[图 4-2](#)所示。
- 步骤 4 连接 RealView-ICE。单击“Selects targets to connect to”，弹出如[图 4-3](#)所示的窗口。单击 RealView-ICE 后的 add，查找并连接仿真器，如[图 4-4](#)所示。
- 步骤 5 连接 ARM1176。单击[图 4-4](#)中的“Connect”按钮，可扫描到 ARM1176，出现如[图 4-5](#)所示的窗口。然后关闭，单击提示对话框的“Yes”按钮，在 RealView ICE 下面就能看到目标 ARM1176，再双击连接就可以进行调试，如[图 3-1](#)所示。



图4-1 RealView ICE Config IP 窗口

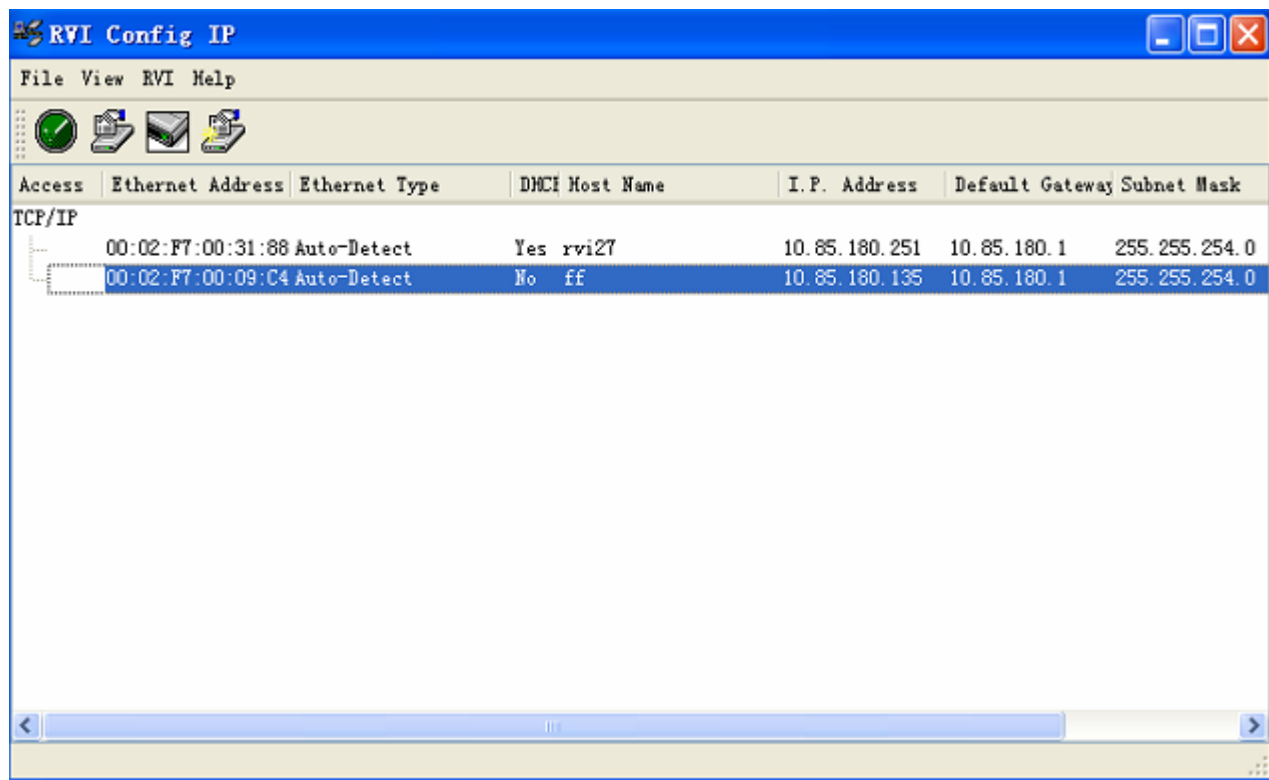




图4-2 RealView Debugger 窗口

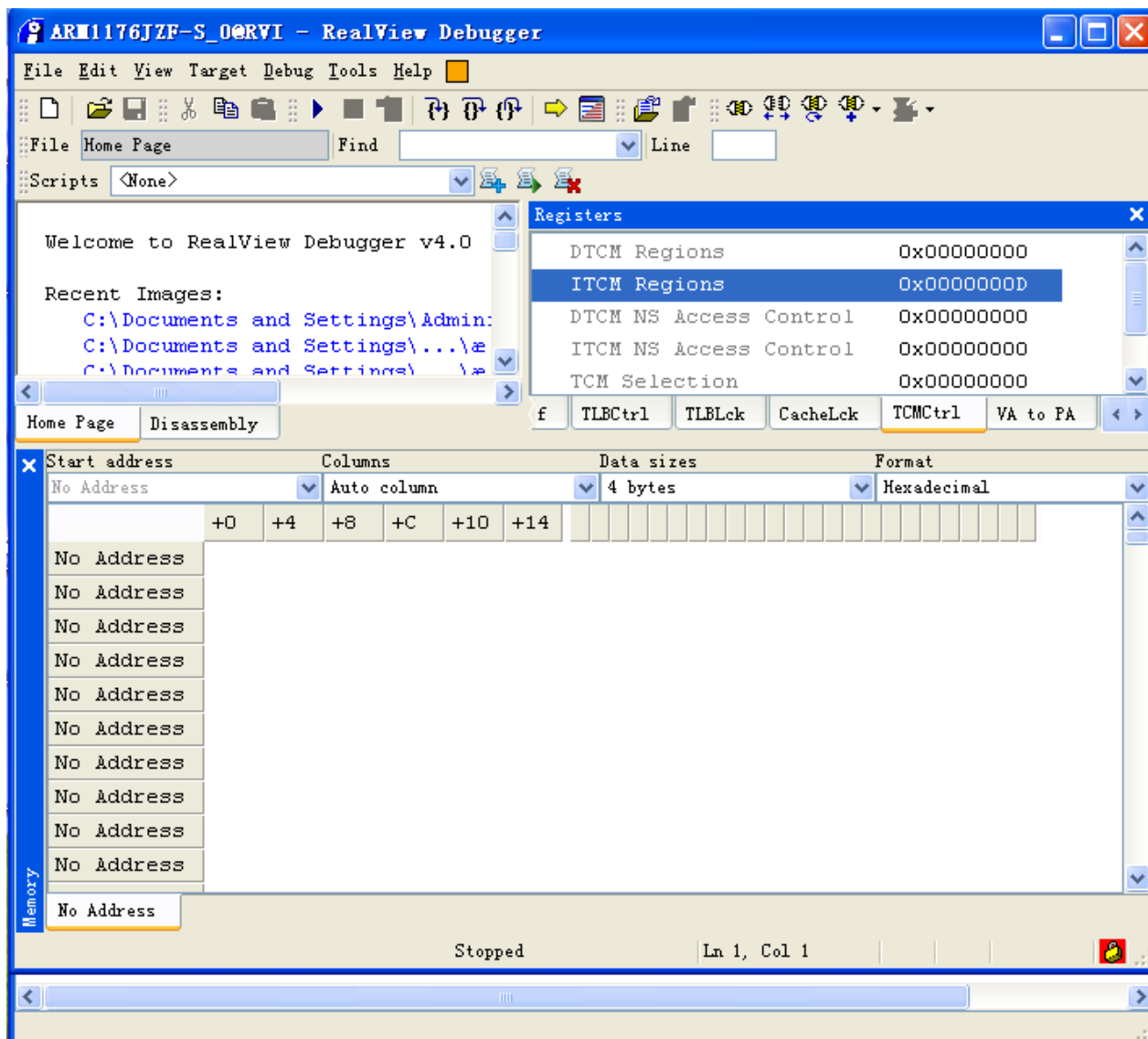




图4-3 Connetct to Target 窗口

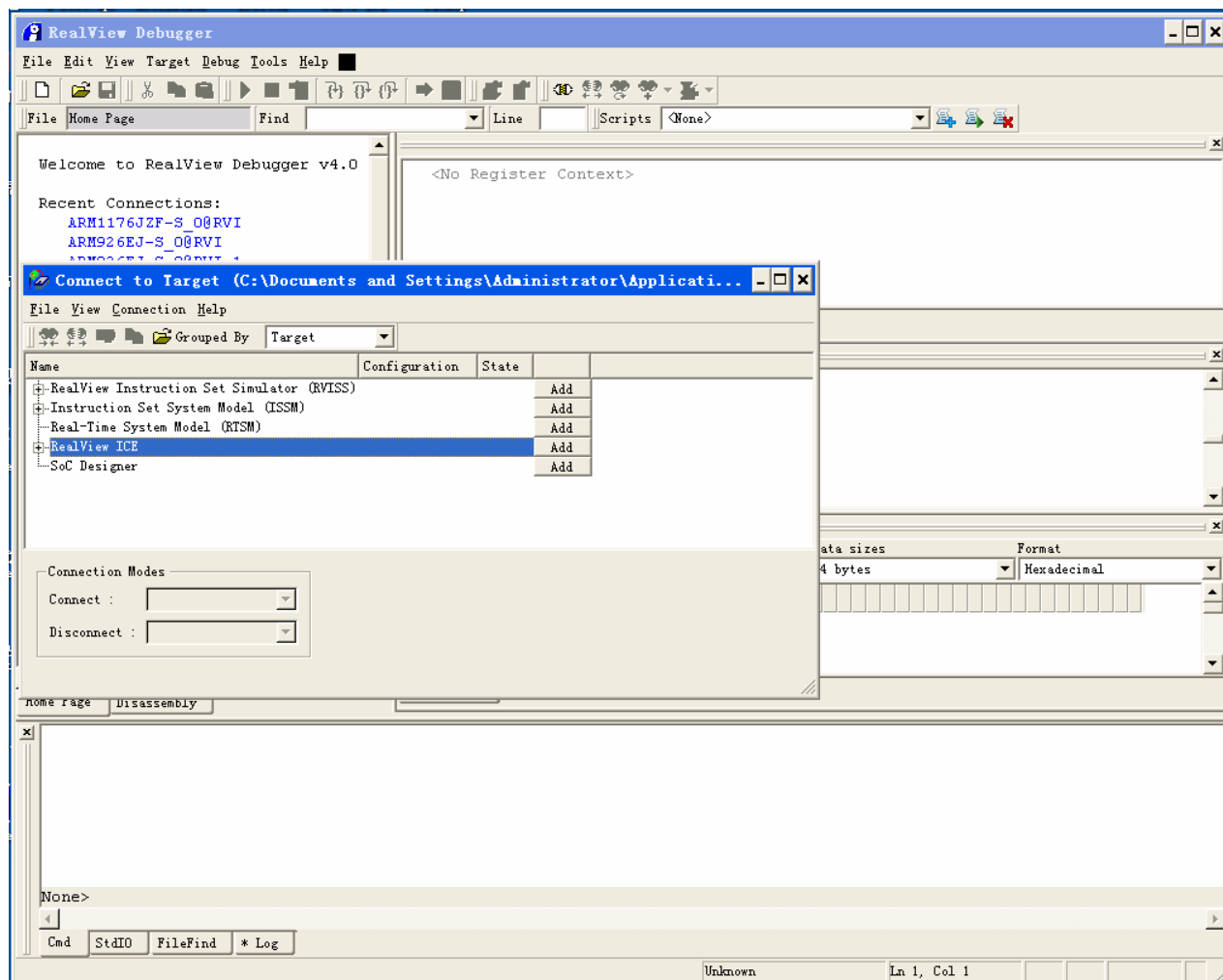




图4-4 Choose RealView-ICE

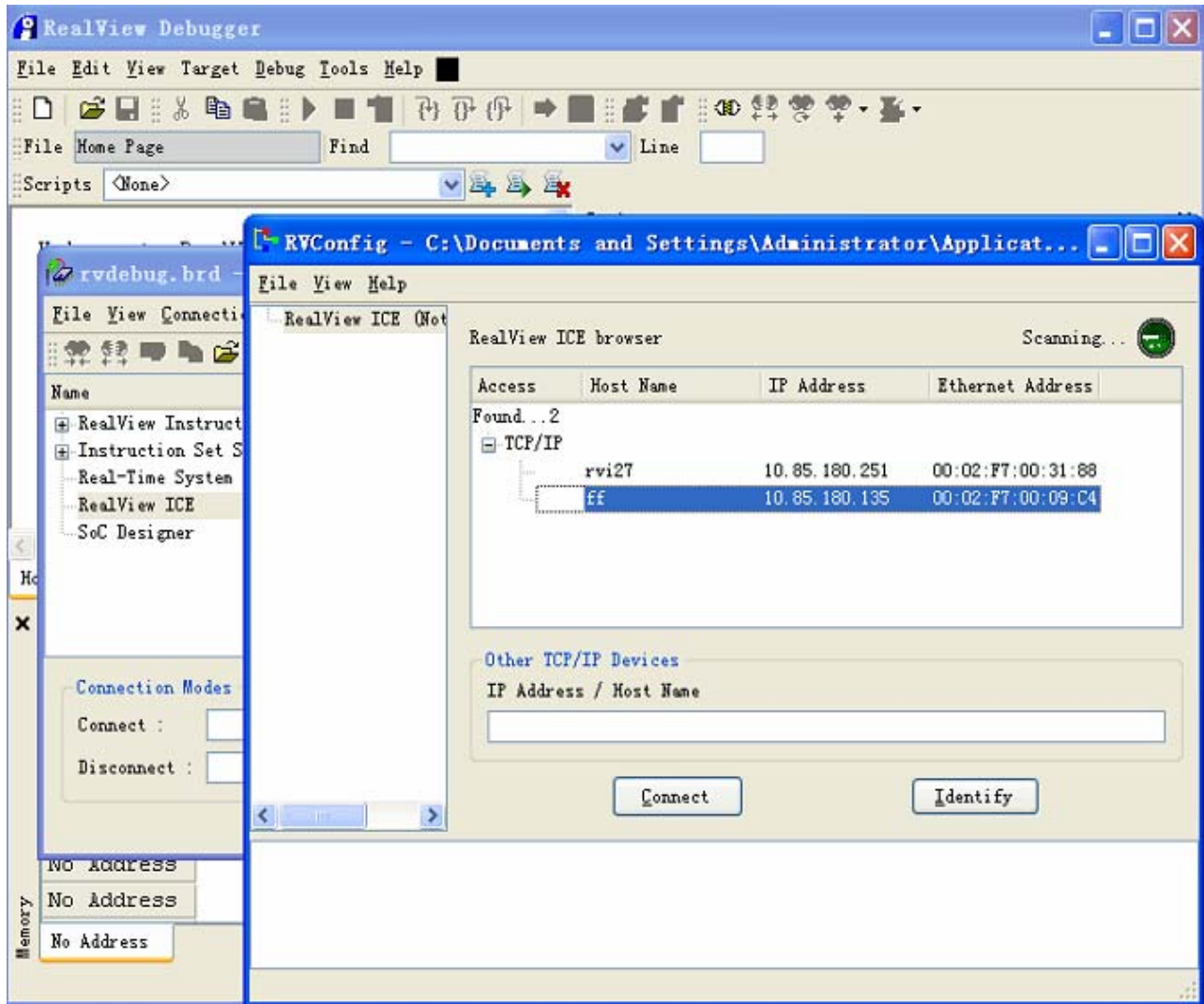




图4-5 RealView Debugger 信息提示框

