



# **C-Sky Debugger Server**

## **User Guide**

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



平头哥半导体有限公司

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

## 目录:

1. 引言.....	9
1.1. 定义 .....	9
1.2. 功能简述 .....	9
2. Windows 篇 .....	10
2.1. ckserver 及 ICE 驱动安装 .....	10
2.1.1. 安装包获取 .....	10
2.1.2. ckserver 安装步骤.....	10
2.1.3. ICE 驱动安装步骤 .....	17
2.2. 运行环境 .....	20
2.3. console 版 ckserver 使用说明 .....	20
2.3.1. 运行参数 .....	20
2.3.2. ckserver 脚本配置功能说明.....	24
2.3.3. 运行说明 .....	28
2.4. GUI 版使用说明 .....	30
2.4.1. 主界面介绍 .....	30
2.4.2. 菜单栏工具栏介绍 .....	30
2.4.3. 启动配置文件说明 .....	34
2.4.4. 常用功能介绍 .....	35
2.4.5. 运行说明 .....	39
3. Linux 篇 .....	41

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

3.1.	ckserver 及 ICE 驱动安装 .....	41
3.1.1.	安装包获取 .....	41
3.1.2.	ckserver 安装步骤.....	41
3.2.	运行环境 .....	42
3.3.	运行参数 .....	42
3.4.	Jtag 脚本配置功能说明 .....	42
3.5.	运行说明 .....	42
4.	半主机功能说明.....	43
5.	调试输出功能说明.....	44
6.	命令行功能说明.....	45
7.	XML 使用说明 .....	47
7.1.	简介 .....	47
7.2.	XMI 文件格式.....	47
7.2.1.	编写规则 .....	47
7.2.2.	举例描述 .....	48
7.2.3.	寄存器编号 .....	50
7.2.4.	扩展的 TEE 寄存器描述 .....	54
7.3.	版本及使用说明 .....	57
7.3.1.	UI 版.....	57
7.3.2.	Console 版 .....	58
8.	多核调试操作使用说明.....	61

8.1.	简介 .....	61
8.2.	调试环境需求 .....	62
8.3.	多核单端口模式 .....	63
8.3.1.	操作步骤 .....	63
8.3.2.	线程操作 .....	69
8.4.	多核多端口模式 .....	73
8.4.1.	操作步骤 .....	73
9.	Vendor ICE 支持 .....	78
10.	Example 工程 .....	80
11.	常见问题及解决方法 .....	80

## 图表目录

表格 1-1 简写定义列表 .....	9
图 1-1 CSkyDebugServer-Server 通信方式 .....	9
图 2-1 ckserver 安装步骤-1 .....	11
图 2-2 ckserver 安装步骤-2 .....	12
图 2-3 ckserver 安装步骤-3 .....	12
图 2-4 ckserver 安装步骤-4 .....	13
图 2-5 ckserver 安装步骤-5 .....	14
图 2-6 ckserver 安装步骤-6 .....	15
图 2-7 ckserver 安装步骤-7 .....	16

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

图 2-8 驱动更新选择对话框 .....	16
图 2-9 ckserver 安装步骤-8 .....	17
图 2-10 windows xp 系统 ICE 驱动安装-1.....	18
图 2-11 windows xp 系统 ICE 驱动安装-2.....	19
图 2-12 windows xp 系统 ICE 驱动安装-3.....	20
表格 2-1 ckserver 运行参数列表.....	20
图 2-13 console 版 ckserver 运行界面 .....	29
图 2-14 ckserver GUI 主界面.....	30
表格 2-2 File 菜单栏.....	30
表格 2-3 View 菜单栏 .....	31
表格 2-4 Control 菜单栏 .....	31
表格 2-5 Setting 菜单栏 .....	32
表格 2-6 Tools 菜单栏 .....	33
表格 2-7 Help 菜单栏.....	34
图 2-15 Target Setting 对话框 .....	36
图 2-16 Socket Setting 对话框 .....	37
图 2-17 固件升级对话框 .....	38
图 2-18 HAD 寄存器操作对话框 .....	39
图 2-19 ckserver GUI 运行界面.....	40
图 3-1 Linux ckserver 运行界面 .....	43
表 7-1 abiv1 寄存器编号.....	51

表 7-2 abiv2 寄存器编号 .....	52
图 7-1 TDFile Setting 菜单选项 .....	58
图 7-2 TDFile Setting 快捷按钮 .....	58
图 7-3 Target Description File 对话框 .....	58
图 7-4 DebugServer 快捷方式属性 .....	59
图 7-5 为快捷方式添加启动参数 .....	60
图 7-6 修改启动参数后点击确认按钮 .....	61
图 8-1 多核调试整体框架 .....	62
图 8-2 多核单端口模型图 .....	63
图 8-3 UI 版本 DebugServer 首次连接刚上电的 CK860MP .....	64
图 8-4 Console 版本 DebugServer 首次连接刚上电的 CK860MP (linux 同) .....	64
图 8-5 启动 GDB 并唤醒 CPU 1 .....	65
图 8-6 断开 UI 版本 DebugServer .....	66
图 8-7 U 版本 DebugServer 已多核单端口连接 CK860MP 显示 .....	67
图 8-8 Console 版本 DebugServer 已多核单端口连接 CK860MP 显示 (linux 同) .....	68
图 8-9 GDB info thread 查看线程 .....	69
图 8-10 GDB 端切换线程查看 CPU 1 的寄存器信息 .....	70
图 8-11 GDB 端切换线程查看 CPU 0 的寄存器信息 .....	71
图 8-12 GDB 端查看当前线程 .....	72
图 8-13 GDB 端切换线程并设置 CPU 0 的 PC 至 0x10000 .....	72

图 8-14 多核多端口模型图 .....	73
图 8-15 UI 界面设置 -no-multicore-threads 模式 .....	74
图 8-16 UI 通过 default.ini 设置 -no-multicore-threads 模式 .....	75
图 8-17 UI 版本 DebugServer 已多核多单端口连接 CK860MP 显示 .....	76
图 8-18 Console 版本 DebugServer 已多核多单端口连接 CK860MP 显示 (linux 同) .....	77
表格 9-1 Link Porting 接口列表 .....	79
表格 11 常见问题及解决方法 .....	80

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



# 1. 引言

本用户手册主要针对 C-Sky Debugger Server 进行功能使用介绍，本软件支持 windows 和 linux 平台，用户手册则分 Windows 篇和 Linux 篇分别做介绍。

## 1.1. 定义

表格 1-1 简写定义列表

名词	含义
ddc	下载直通通道，可大幅提高数据下载速度
ckserver	C-Sky Debugger Server 简写，调试代理服务程序
ICE	在线仿真器（In Circuit Emulator），在这里指的是 CKLINK

## 1.2. 功能简述

ckserver 接收 C-SKY GDB 发送的调试原语操作命令，按照 JTAG 协议发送相应命令到调试硬件接口（HAD），并控制调试指令执行，获取调试数据返回给 C-SKY GDB。C-SKY GDB 与 ckserver 采用 socket 方式进行通信，调试器 GDB 和调试代理服务器可以运行在不同的主机上。ckserver 和目标机之间通过 ICE 按照 JTAG 协议进行通信。

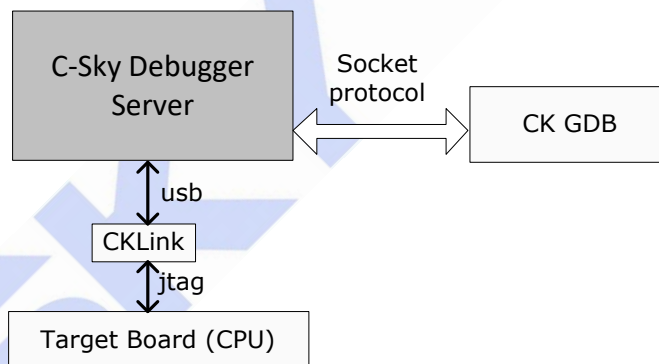


图 1-1 CSkyDebugServer-Server 通信方式

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

## 2. Windows 篇

ckserver 在 Windows 主机系统下支持图形用户界面（GUI 版本）和命令行（console 版本）两种运行方式。

### 2.1. ckserver 及 ICE 驱动安装

ICE 的驱动打包在 ckserver 的安装包中，安装 ckserver 时，勾选 ICE Driver 选项，安装时会将驱动文件拷贝到系统目录下。插上 ICE 设备后，Windows 会自动为 ICE 安装驱动。

#### 2.1.1. 安装包获取

从 C-SKY 公司的 OCC 平台 [https://occ.t-head.cn/community/download\\_detail?id=616215132330000384](https://occ.t-head.cn/community/download_detail?id=616215132330000384) 或技术支持处获取安装包，安装包中包含：Windows 平台的 CSKY-DebugServer-windows\*.zip 压缩文件和 Linux 平台的两个 CSKY-DebugSever-linux-\*.sh 安装文件（分别对应 32 位和 64 位系统）。

#### 2.1.2. ckserver 安装步骤

1. 解压 CSKY-DebugServer-windows\*.zip 文件，双击运行解压出的 Setup.exe 将弹出 ckserver 安装界面，点击 next。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

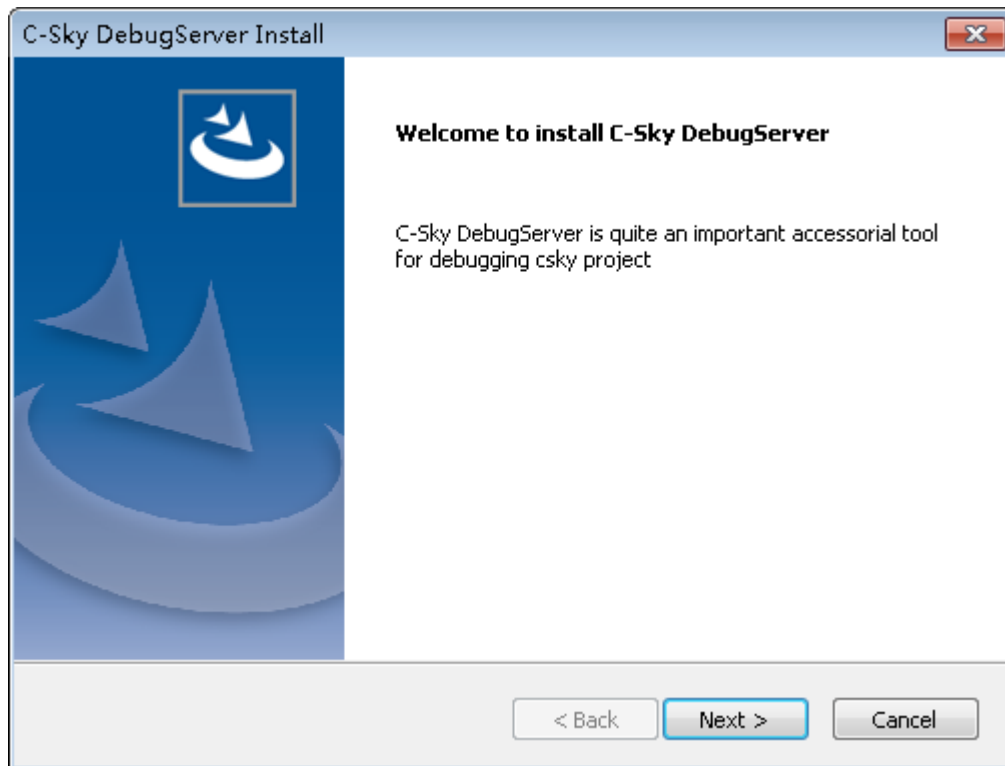
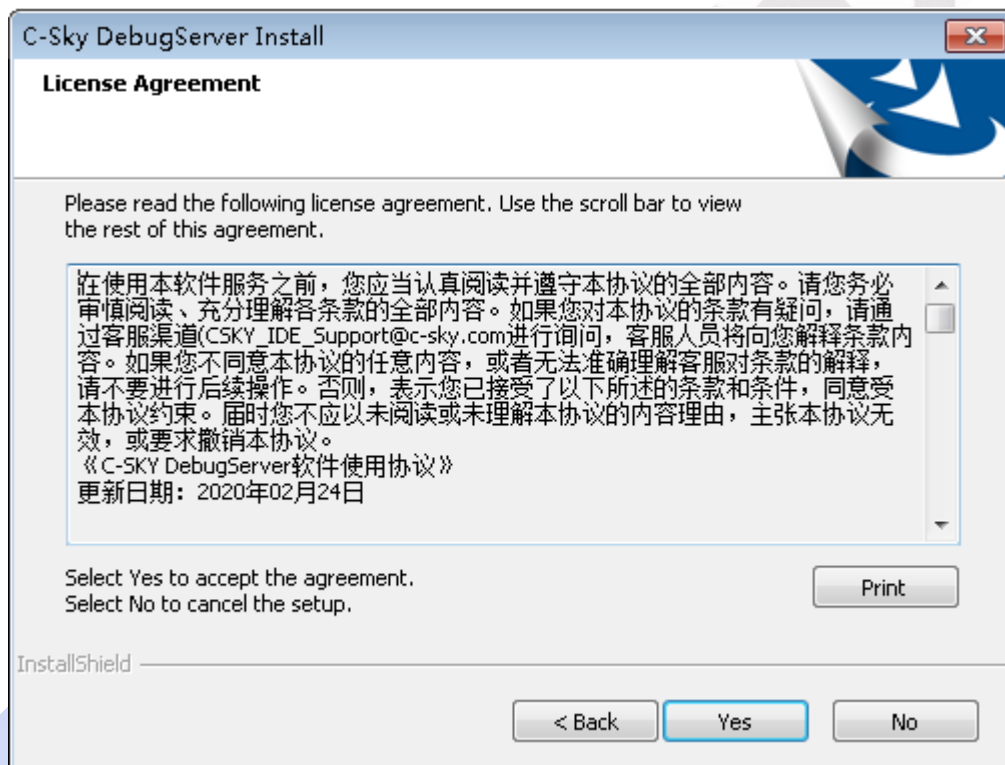


图 2-1 ckserver 安装步骤-1

2.请认真查看 ckserver 的软件使用协议，确认后点击 yes 继续。



## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

图 2-2 ckserver 安装步骤-2

3.输入用户名和企业名称，选择用户，点击 next。

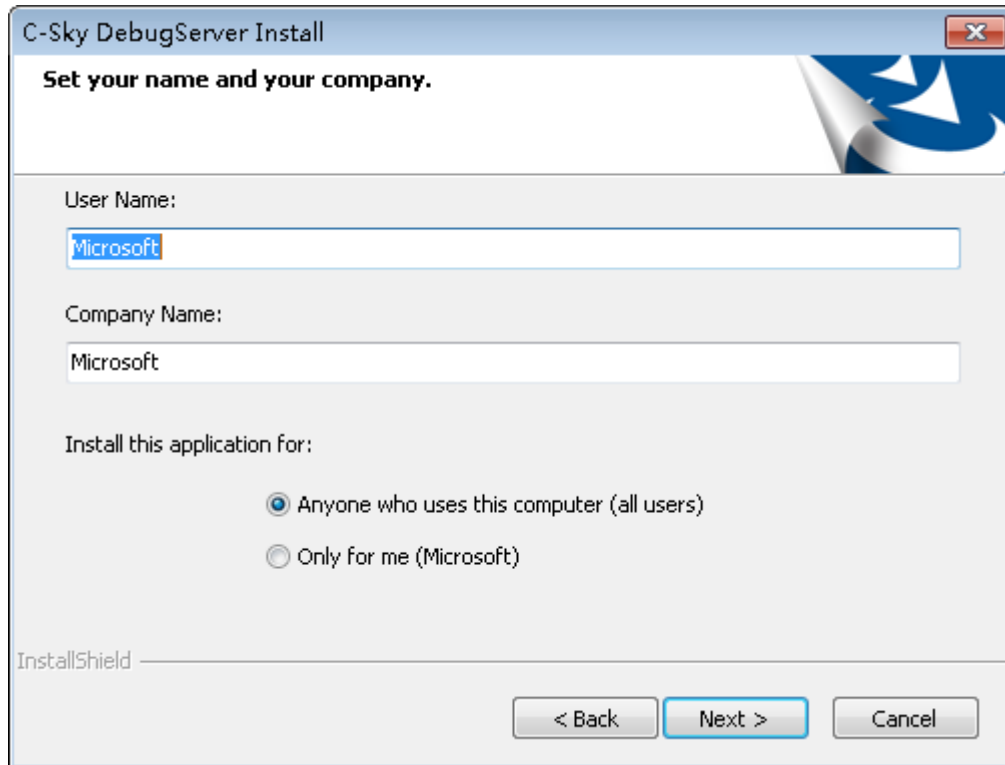


图 2-3 ckserver 安装步骤-3

4.选择安装目录。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

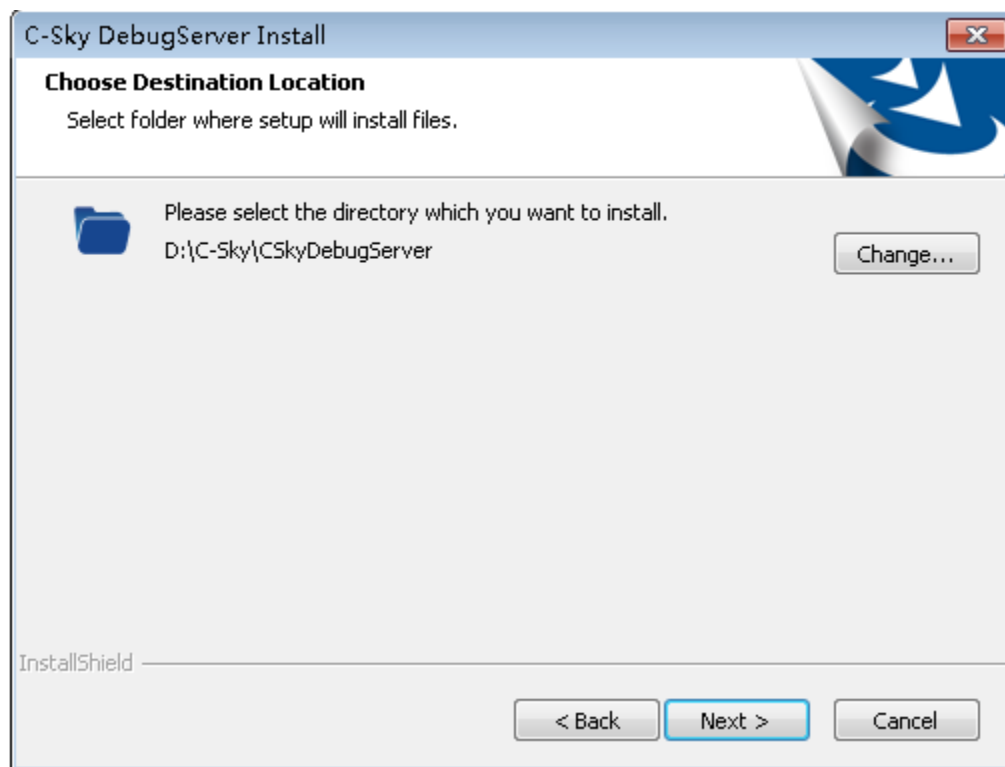


图 2-4 ckserver 安装步骤-4

5. 选择安装内容，其中： C-SKY Debug Proxy Server 是 C-SKY 调试代理程序；ICE Driver 是 ICE 设备驱动安装；Tutorial 是用户手册。建议全选， 点击 next。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

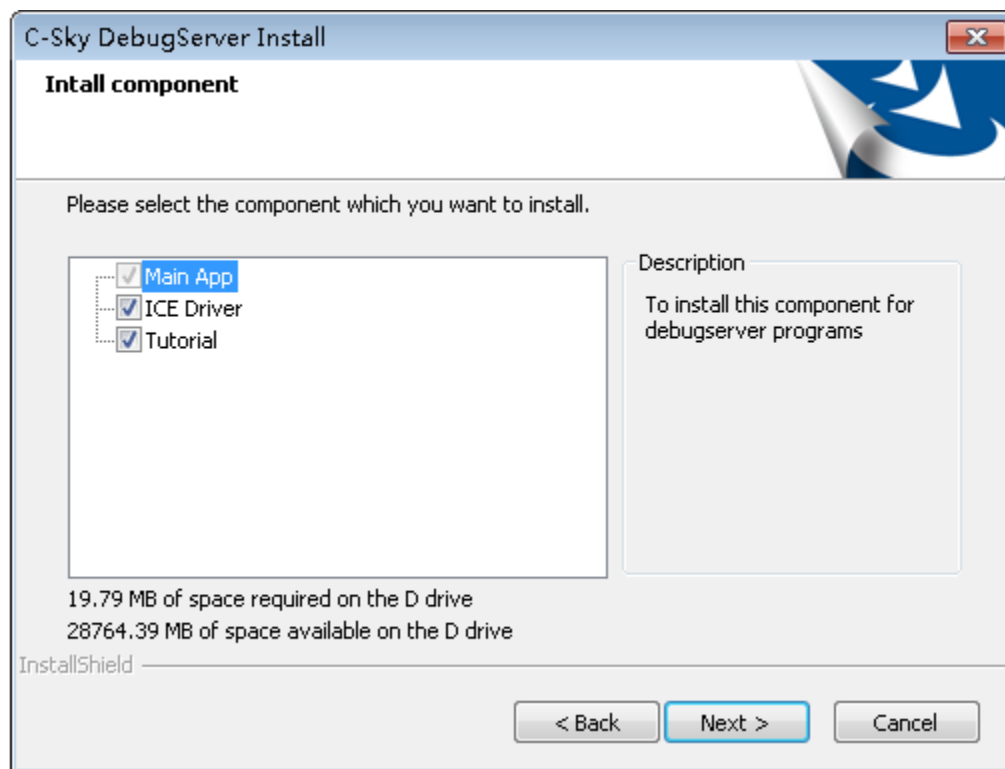


图 2-5 ckserver 安装步骤-5

6.该页面将显示用户信息及安装目录，确认无误后点击 **Next** 开始安装。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

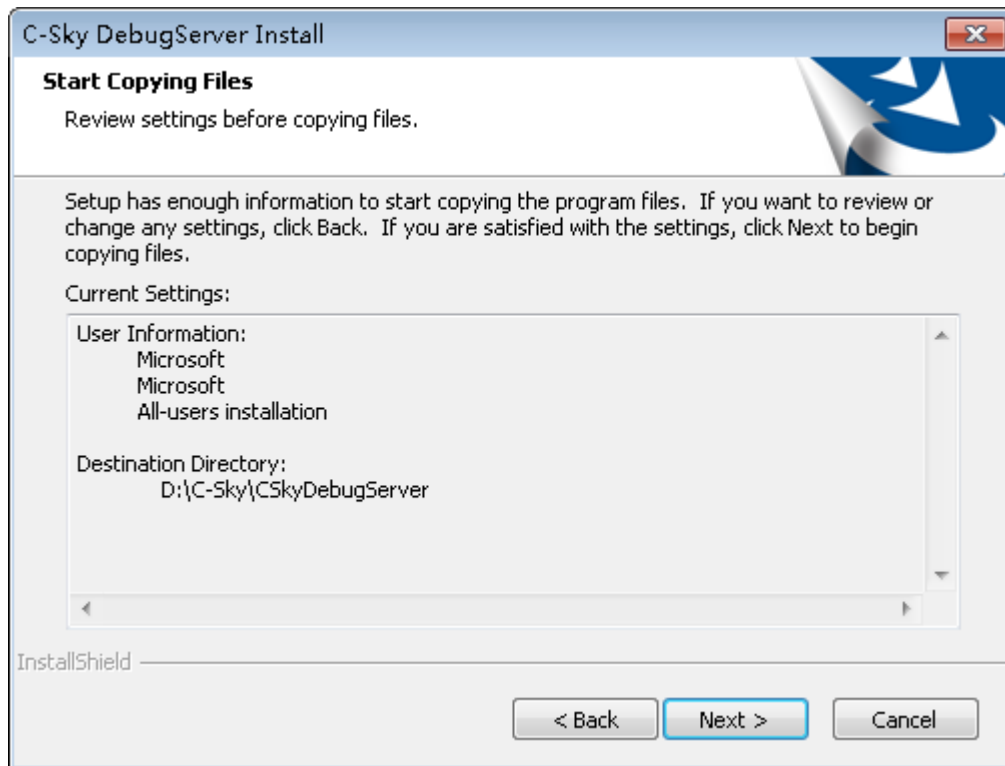


图 2-6 ckserver 安装步骤-6

7.开始安装。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

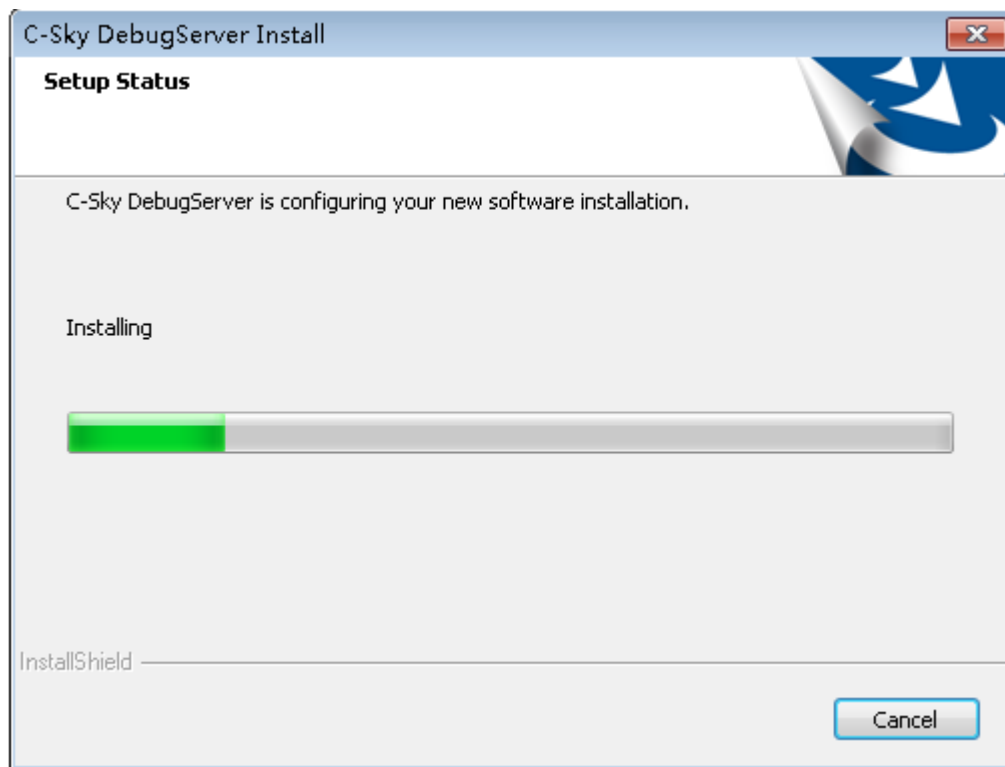


图 2-7 ckserver 安装步骤-7

8.若您的 PC 上已经安装过 ICE 驱动，会提示是否进行驱动更新，建议选择“是”，将您的 ICE 更新最新的驱动。

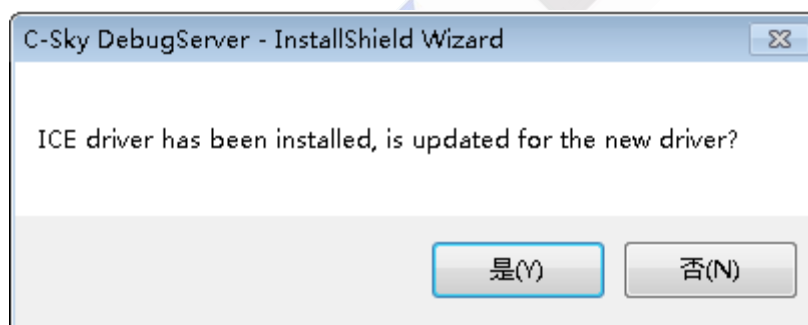


图 2-8 驱动更新选择对话框

9.ckserver 安装完成，点击 finish。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



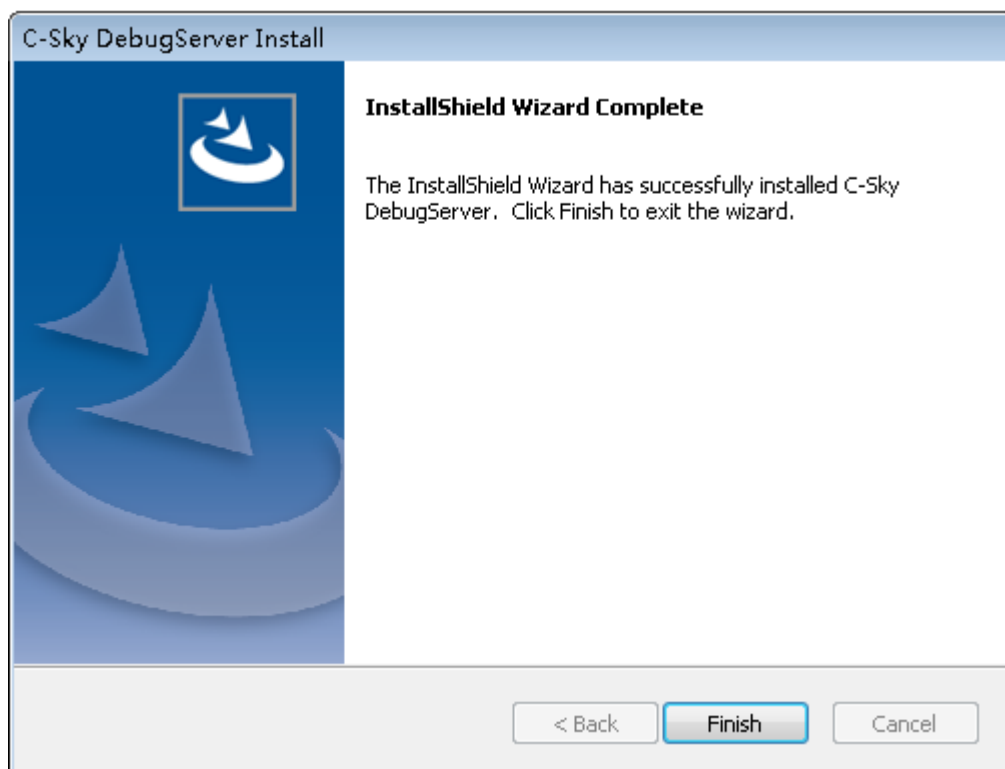


图 2-9 ckserver 安装步骤-8

### 2.1.3. ICE 驱动安装步骤

1. 在“2.1.2 节 ckserver 安装步骤”中，如果勾选了 ICE Driver 选项。那么 ckserver 安装过程中会将驱动相关文件拷贝到系统的目录下。
2. 将您的 ICE 设备重新连接至 PC，
  - 如果您使用的是 windows 7 系统，那么设备插上后会自动安装好驱动；
  - 如果您使用的是 windows xp 及以下系统，ICE 设备重新连接后，系统会提示设备安装驱动，以 CKLink-Lite V1 为例,步骤如下：
    - (1) 系统弹出硬件安装向导，选择“自动安装软件”，点击下一步

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

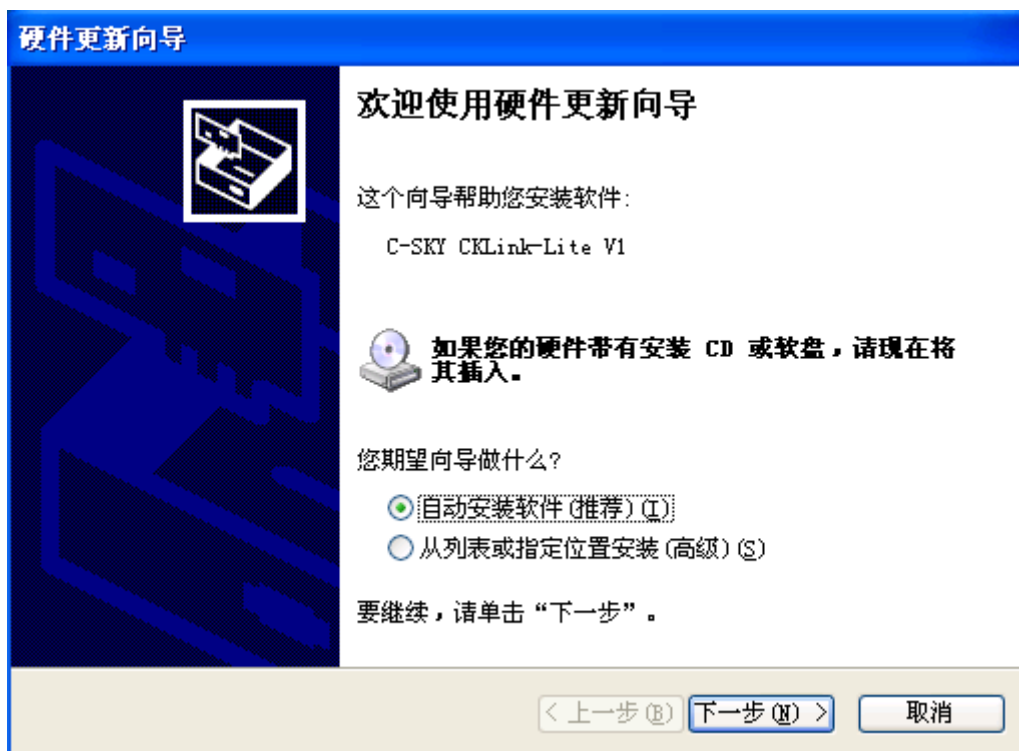


图 2-10 windows xp 系统 ICE 驱动安装-1

## (2) 驱动安装

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



图 2-11 windows xp 系统 ICE 驱动安装-2

(3) 驱动安装完成

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



图 2-12 windows xp 系统 ICE 驱动安装-3

## 2.2. 运行环境

ckserver 运行要求 windows2000 及以上的版本，输入输出设备要求具有 USB 接口。  
ckserver 可配合 C-SKY GDB 使用，支持所有版本。

## 2.3. console 版 ckserver 使用说明

### 2.3.1. 运行参数

1. console 版 ckserver 通过执行 ckserver 的运行参数来实现 ckserver 的设置。常用输入参数见下表。

表格 2-1 ckserver 运行参数列表

用户接口	定义	默认值
【-setclk xxx】	设定 ICE Jtag 时钟频率，默认单位 Mhz，支持 Khz 为单位的格式，如设置-setclk 12000k 表示频率设置为 12M。	默认值为 12，即 ICE 频率为 12M

C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

	<p>每种类型 ICE 有固定的上限频率，具体如下：</p> <p>CKLINK-PRO      24Mhz</p> <p>CKLINK-V1        24Mhz</p> <p>CKLINK-Lite      2500khz</p>	
【-port XXX】	设定 socket 通信端口	1025
【-prereset】	指定在获取 ICE 控制后发起 reset 操作	默认不执行
【-noddc】	不使用数据下载直通通道	默认使用
【-nocacheflush】	ckserver 执行单步和退出调试模式时不刷 cache	默认刷 cache
【-setcdi 2/5】	之后跟的参数为 2/5，即连接 TARGET 之前设置 ICE 的工作方式，2 线或 5 线	默认不执行
【-mtcrdelay/delay】	设置 ICE 写完 CPU 控制寄存器后等待的时间	默认为 1ms
【-targetinit filepath】	在获取 ICE 后执行的目标初始化脚本，脚本类型为 GPIO 或 JTAG 脚本，执行脚本后继续 Server 启动流程	无
【-scr filename】	指定执行 cskserver 操作脚本(gpio 或 jtag 脚本)，这个参数指定后，server 功能不再开启，运行完脚本后程序退出	无
【-configpath/-configfilepath filepath】	指定配置 ICE 时，获取 ICE 固件的路径	默认为 ckserver 可执行程序所在目录
【-tdescfile filepath】	为 gdb 指定描述目标板寄存器的 xml 文件	默认根据 CPUID 来指定一个默认的 tdesc-xml 文件
【-{no-}trst】	在执行 reset 命令时是否执行 treset	默认执行
【-nrstdelay】	设置 Nreset 延时，确保 ICE 可以生成稳定的硬件复位 (nreset) 信号，单位 10us	默认为 1ms

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

【-trstdelay】	设置 <b>Treset</b> 延时，确保 ICE 可以产生稳定的复位信号，用于复位 HAD 状态机，单位 10us	默认为 1.1ms
【-rstwait】	设置延时，确保在目标板收到复位信号后，目标板复位流程执行结束	默认为 50ms
【-no-multicore-threads】	设置连接 CK860 多核调试时将为每个核开启一个调试端口，GDB 连接一个端口为调试该端口对应的核。默认为假，DebugServer 只会开启一个调试端口，同时封装多核为多个线程信息发送给 GDB。GDB 调试线程，则在调试线程对应的核。	默认为假
【-local-semi/-ls】	程序发生的 <b>Semihost</b> 请求由 ckserver 完成	默认情况下， <b>Semihosting</b> 的执行在 GDB 中完成，可通过[-local-semi/-ls]选项重设为在 ckserver 中执行。  注意：在 Windows 中加-local-semi/-ls 时，不支持 isatty 和 system 操作。
【-dcomm=ldcc】	启动 jtag 输入输出通道（需要硬件支持）	默认不开启
【-disable-cmdline】	不开启命令行功能	默认开启
【-set-isa_version v1/v2/v3/v4/v5】	设置 ICE 的 HAD Version 版本	默认由 ckserver 自动设置

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

【-set-hacr_width 8/16】	指定设置连接目标板时 HACR 的宽度	默认为 ckserver 自动探测
【-no-cpuid-check】	连接目标板过程中不读取 CPUID 信息	默认读取并检查
【-cacheflush-delay xxx】	设置 ckserver 中刷 cache 行为的延时，确保刷 cache 行为正常结束	默认 100ms
【--debug usb/connect/target/remote/djp/sys/all】	usb:记录 ckserver 与 ICE 之前交互的协议包 connect:描述连接开发板的细致过程 target:记录 target 抽象层函数调用信息 remote:记录 remote 协议交互信息 djp:记录 djp 协议交互信息 sys:记录 ckserver 程序主循环信息 all:打印以上所有 Log 信息	默认没有 log 信息输出
【-arch csky/ricsv】	选择连接的调试架构，csky 指代 CSKY HAD 调试架构，ricsv 指代 RISCV DM 调试架构	默认为 csky
【-cmd-script file】	执行 ckserver 命令行脚本，即可以将 cmdline 处的命令写成文件，在 DebugServer 连接上目标板之后执行	无
【-list-ice】	罗列当前连接在 PC 上的 ICE	无
【-select-ice xxx】	根据-list-ice 罗列的 ICE 列表，指定 ICE 进行连接	无
【-list-vendor】	显示支持的 ICE 设备产商，如 CKLink	
【-select-vendor】	选择 ICE 的 Vendor	默认选择 CK-LINK
【-v/-version】	查看程序版本号信息	
【-h/--h/--help】	查看帮助信息	

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

## 2. 参数配置示例

这里将列举出常用的输入举例，ckserver 是我们应用程序。如下所示：

### ●ckserver

该命令相当于 `ckserver -setclk 12M -port 1025`。使用默认选项。

### ●ckserver -stclk 13000k -port 1111

设置 socket 通信端口为 1111，硬件 ICE 的频率设置为 13M。

### ●ckserver -scr E:\jtag.ini

执行指定的 ckserver 脚本。

### ●连接调试架构依据 RISCv 调试架构 0.11，0.13 版本的开发板时一定要指定 -arch riscv 参数

## 2.3.2. ckserver 脚本配置功能说明

Console 版 ckserver 支持脚本运行功能，脚本分为 JTAG 脚本和 GPIO 脚本。JTAG 脚本可直接操作 Jtag 扫描链寄存器，通过自定义组合，即可访问相应的 HAD 寄存器，亦可以用于操作其他硬件；GPIO 脚本可控制 ICE JTAG 引脚电平的输出，也可以获取 ICE JTAG 引脚的电平，用户通过该脚本可产生自定义波形。可产生对应引脚上需要的波形，也可以读取 JTAG 引脚上的电平状态。

以下两个小节对两种类型脚本的使用方法进行详细介绍：

### 2.3.2.1. JTAG 脚本

ckserver 支持通过执行 Jtag 脚本来读写 HAD 寄存器。

JTAG 脚本文件介绍说明如下：

1. 文件后缀没有要求，可以是相对路径
2. 语法规则

(1) 脚本中可以描述多次的 jtag 操作，每个操作单元必须以[JTAGx]作为关键字，其中 x 表示数字。[JTAGx]要求必须大写，并数字连续，一旦出现不连续，后续不执行。

(2) IR/DR 必须大写，格式如下：

IR=[ir 长度], byte0, byte1 ... ——IR 写

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



DR=W,[dr 长度] byte0, byte1 ... —— DR 写

DR=R, [dr 长度] ——DR 读

(3) IR/DR 长度单位是 bit, 但必须为的 8 倍数 (以后可能会支持非 8 对齐的 bit 数), 这里对 ir/dr 长度支持 byte 倍数。

(4) byte 内容必须为十六进制, 可以有前缀 0x 也可以没有。

(5) 移进 jtag dr 的字节顺序按照先后顺序, 同样读 DR 时, 输出顺序也是按照字节先后顺序。

### 3. 脚本实例

[JTAG0]

IR=x, 0x8

DR=R, 32

[JTAG1]

IR=x, 0x8E

DR=R, 16

### 2.3.2.2. GPIO 脚本

Ckserver 支持通过 GPIO 脚本控制 JTAG 引脚电平的输出, 可产生对应引脚上需要的波形; 也可以读取 JTAG 引脚上的电平状态。

GPIO 脚本文件介绍说明如下:

1. C-SKY GPIO 脚本以[CSKY\_ICE\_GPIO]字段表征, 如果不是以该字段开头的脚本文件, 即认为不是 C-SKY GPIO 脚本。
2. 文件后缀没有要求, 可以是相对路径
3. 语法规则

- (1) 脚本以行为解析执行单位, 每一行进行一次解析执行。同一行中可以有多个语句, 语句之间逗号隔开。最后一条语句后面不得加逗号, 否则为语法错误。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- (2) 同一行的多条语句，允许对同一个寄存器的同一个位进行操作。采取后面的覆盖前面的原则。如 `TDI=1, TMS=1, NRST=0, TDI =0`，那么最终解析执行的结果控制 TDI 的电平输出为 0；
- (3) 对单个引脚赋值语句中如 `TMS=1`，赋值必须是 0 或者 1，否则语法错误。以 0x 开头的值视为 16 进制，否则 10 进制处理。
- (4) 在执行脚本过程中，各个 JTAG 引脚电平的初始化值为 0，本次操作中没有对该引脚进行赋值输出操作，那么该引脚的电平保持上次操作时的电平；
- (5) `GPIO_OE` 寄存器默认的值位 0x3f，即 JTAG 各个引脚对应的均为输出模式。对 `GPIO_OE` 寄存器进行赋值操作更改引脚的输入输出模式，`GPIO_OE` 寄存器一经赋值后一直有效，直到下一次对 `GPIO_OE` 进行赋值；
- (6) 脚本支持 repeat 循环语句，语法为：`REPT=x..... ENDR`。支持嵌套循环，但是 `REPT` 和 `ENDR` 必须得匹配，否则语法错误；`REPT` 后面的等号不能省略，循环次数 `x` 不能小于 0，否则语法错误；`REPT` 语句和 `ENDR` 语句必须单独一行，否则语法错误；
- (7) `PRINT` 语句用来读取 `GPIO_IN` 寄存器的值，语法为 `PRINT=GPIO_IN`，否则语法错误；
- (8) 脚本内容以 `START` 开始，以 `END` 结尾，否则提示语法错误；
- (9) 脚本不区分大小写；
- (10) #后的语句为注释部分。

#### 4. 支持的语句类型

CSKY\_ICE\_GPIO 脚本支持的几类语句：

- 对引脚输入/输出方向控制语句，对各引脚的输入输出方向进行控制,如 `GPIO_OE = 0x1f`;
- 对单个引脚进行输出的赋值语句，控制单个引脚的输出。如 `TMS = 1`，将 `GPIO_OUT` 寄存器中 `TMS` 引脚对应的位赋值为 1；
- 对所有引脚一次性赋值语句，如 `GPIO_OUT=0x1d`，将值赋值给 `GPIO_OUT` 寄存器；
- `PRINT=GPIO_IN` 语句，读取 `GPIO_IN` 寄存器的值，并且打印。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- REPT, ENDR 语句, 即 repeat 循环语句, REPT=x, 其中 x 为循环次数, 即将 REPT 和 ENDR 之间的语句循环 x 次。

**注意:** 脚本中的输出赋值语句如 TMS=1 和输入打印 PRINT=GPIO\_IN, 仅仅是对 ICE 中的 GPIO\_OUT 和 GPIO\_IN 寄存器进行读写, 只有在 GPIO\_OE 寄存器的操作中, JTAG 引脚才对对应的位进行正确的设置。比如, TMS 对应的 GPIO\_OE[3]=1 时, 即输出模式, GPIO\_OUT[3] 的值才会反应在 TMS 引脚上; 同样的, 只有当 GPIO\_OE[3]=0, 即输入模式时, PRINT=GPIO\_IN 语句读回来的 GPIO\_IN[3] 才是真正 TMS 引脚的电平值, 否则该值无效。

## 5. 脚本实例

```
[CSKY_ICE_GPIO]

START

GPIO_OE = 0x1f

TMS=1, TDO=0, TDI=1

TDI=1,TMS=1,NRST=0, TDI =0

PRINT= GPIO_IN    #print the value of GPIO_DATA

GPIO_OE = 0x13

REPT = 10

TMS =0

TMS = 1

ENDR


TRST=1

#GPIO_CTRL=0x1d

END
```

### 2.3.2.3. 脚本执行

Console 版 ckserver 执行脚本为: 在 ckserver 运行参数 (2.3.1) 中 **【-scr filename】**。

GUI 版 ckserver 执行脚本在菜单栏工具栏 (2.4.2) 中设有对应按钮 。

注: 由于脚本执行需要连接 ICE 及开发板, 故 GUI 版 ckserver 执行脚本前先运行 ckserver 确认硬件已连接正确, 再停止 ckserver, 点击脚本按钮选择脚本执行。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

### 2.3.3. 运行说明

#### 1. 步骤

- 点击“开始->所有程序->C-Sky -> C-Sky DebugServer->bin->DebugServerConsole.exe 以默认运行参数打开，运行界面如图 2-12 所示。

- 如果用户需要更改运行参数，可以“CTRL+c”断开连接；此时当前目录为安装目录，只需输入待运行参数的运行命令，如“DebugServerConsole.exe -port 1028”即可。

- 运行 C-SKY GDB 应用程序进行调试：

1. 使用 CSKY 工具链生成 CSKY elf 程序 a.out

2. 启动 GDB，比如 csky-\*-gdb a.out

3. 根据 ckserver 界面上的提示，在 GDB 的命令行输入连接命令，比如 target remote 172.16.28.158:1025

4. 当 GDB 连接上 ckserver 之后，可以进行 GDB 常用的操作，比如：

- ① load // 下载程序至开发板
- ② break main // 在 main 函数处设置断点
- ③ continue // 运行程序
- ④ info registers r0 // 查看寄存器 r0
- ⑤ print var\_a // 查看程序变量 var\_a

5. GDB 的使用方法 GNU GDB 保持一致

- 如果使用 CDK 或 CDS，请参考开发环境所带的用户文档

#### 2. 运行界面

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

```

+-----+
! C-Sky Debugger Server (Build: Mar  2 2020)
! User Layer Version : 5.9.03
! Target Layer version : 2.0
! Copyright (C) 2019 Hangzhou C-SKY Microsystems co.,ltd
!
+-----+
C-SKY: CKLink_Pro_U1, App_ver 0.1, Bit_ver 1.10, Clock 12000.000KHz,
      2-wire, With DDC, Cache Flush On.
+--- CPU 0 ---+
C-SKY CPU ID:
      0x04880063
Target Chip Info:
      CPU Type is CK802, in LITTLE Endian.

GDB connect command for CPU 0:
      target remote 172.16.28.158:1025
      target remote 192.168.204.1:1025
      target remote 192.168.145.1:1025

***** DebuggerServer Commands List *****

singlestep/si
      execute single-step in the target
reset
      reset the target
pctrace
      show the PCFIFO(8 <= length <= 4096, default 8)
print/p
      print /x[d/f/o] *memory[$registers], eg p /x *0x20000000
      print target
set
      Set *memory[$registers]=value, eg. set $r0=0x1234
quit/q
      quit Debugger Server
help/h
      show help informations
CTRL+B ENTER
      switch input channel
*****
DebuggerServer$

```

图 2-13 console 版 ckserver 运行界面

### 3. 注意事项

- (1) 如果运行 ckserver 时提示 ICE Upgrade, 选择“是”, 进行固件升级, 升级完成后需要重新插拔 ICE, 使用 ckserver 重新连接。
- (2) 当目标为多核的情况下, ckserver 针对 CPU 在结构中的编号, 从用户指定的端口 (默认从 1025) 开启多个端口供 GDB 连接。
- (3) 多核调试时, 如果使用 -tdescfile 为 GDB 指定描述寄存器的信息, 假设指定的文件为 file.xml, 则第二个 CPU 的 xml 文件需存放于与 file.xml 同级目录下, 名称为 file2.xml, 第三个 CPU 的 xml 文件为 file3.xml。即第 n 个 CPU 的 xml 为 filen.xml。

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

## 2.4. GUI 版使用说明

### 2.4.1. 主界面介绍

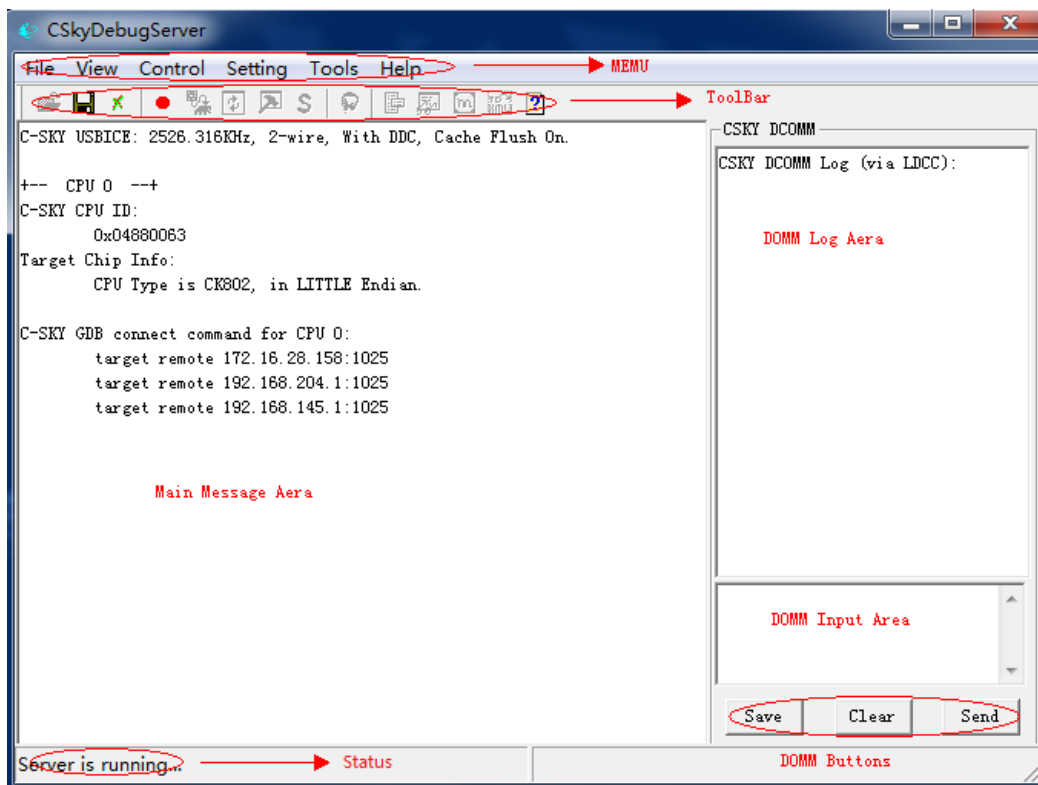


图 2-14 ckserver GUI 主界面

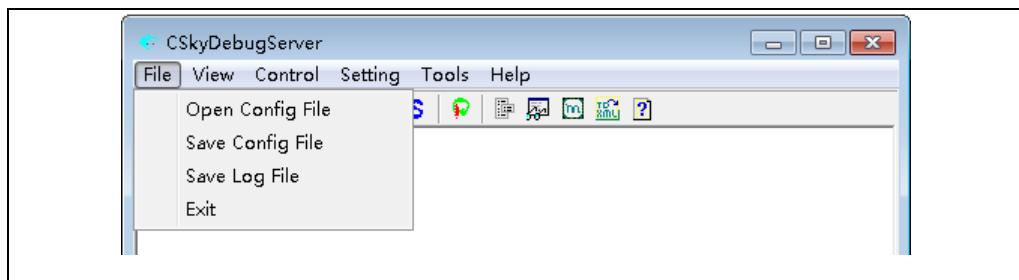
### 2.4.2. 菜单栏工具栏介绍

表格 2-2 File 菜单栏


菜单名称	功能介绍	工具栏按钮
Open Config File	打开 Target 和 Socket 信息的配置文件	
Save Config File	保存 Target 和 Socket 配置信息文件	
Save Log file	保存 log 信息到 log 文件	/
Exit	关闭程序	/

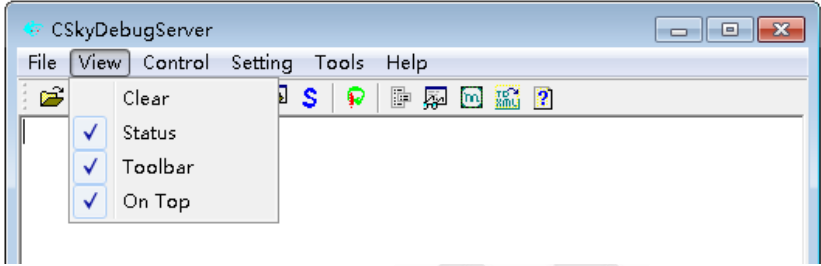
**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).






表格 2-3 View 菜单栏

菜单名称	功能介绍	工具栏按钮
Clear	清空 message area	
Status	显示/隐藏状态栏	/
Toolbar	显示/隐藏工具栏	/
On Top	显示总在最前	/

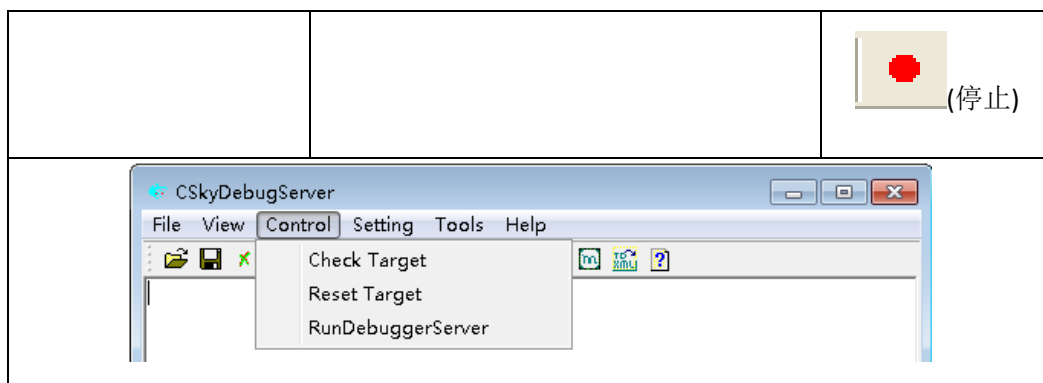
表格 2-4 Control 菜单栏

菜单名称	功能介绍	工具栏按钮
Check Target	检查连接的 Target 目标	
Reset Target	对目标板进行复位，复位功能依赖于目标板 jtag nrst 信号连接方式	
RunDebuggerServer	运行/停止 Debugger Server	 (运行)

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).





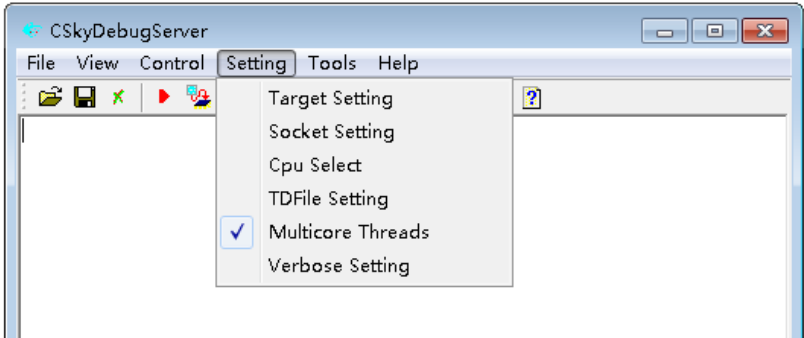
表格 2-5 Setting 菜单栏

菜单名称	功能介绍	工具栏按钮
Target Setting	设置 ICE 配置参数，包括 ICE 工作频率、是否使用 ddc 等，详见图 2-14 及其说明	
Socket Setting	设置通信端口（默认 1025），详见图 2-15 及其说明	
Cpu Select	在调试多核开发板时，在 ckserver 界面当查看 HAD/CPU 寄存器时需先选择对应的 CPU	
TDFile Setting	为 GDB 指定描述目标板寄存器信息的 xml 文件	
Multicore Threads	选中表明 ckserver 在 CK860 多核调试时只会开启一个调试端口，同时封装多核为多个线程信息发送给 GDB。GDB 调试线程，则在调试线程对应的核。  未选中表明连接 CK860 多核调试时将为每个核开启一个调试端口，GDB 连接一个端口为调试该端口对应的核。	无快捷键
Verbose Setting	设置 ckserver 的 log 信息输出：  usb: 记录 ckserver 与 ICE 之前交互的协议包  connect: 描述连接开发板的细致过程	无快捷键

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



	<p><b>target:</b>记录 target 抽象层函数调用信息</p> <p><b>remote:</b>记录 remote 协议交互信息</p> <p><b>djp:</b>记录 djp 协议交互信息</p> <p><b>sys:</b>记录 ckserver 程序主循环信息</p> <p><b>all:</b>打印以上所有 Log 信息</p> <p>选择需要的 log 选项后, 点击 <b>select</b>, 重新使用 ckserver 连接开发板即可。</p>	
		

表格 2-6 Tools 菜单栏

菜单名称	功能介绍	工具栏按钮
Upgrade Firmware	<p>升级 ICE 设备固件（注意：升级文件与 ICE 盒子类型要保持一致,具体对应如下：</p> <p>CKLINK_LITE_V2: cklink_lite.hex</p> <p>CKLINK_PRO_V1: cklink_v1.iic</p> <p>CKLINK_PRO_V2: cklink_pro.iic</p> <p>）</p>	
HAD Register	操作 HAD 模块的寄存器	
Execute Script	执行 ckserver 脚本	

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



DM Register	查看 RISC-V 调试架构下的 Debug Module 寄存器	暂无

表格 2-7 Help 菜单栏

菜单名称	功能介绍	工具栏按钮
About	显示 Debugger Server 版本信息	
ICE Info	显示当前已连接的 ICE 盒子的类型及固件信息	/

2.4.3. 启动配置文件说明

启动文件默认为与 ckserver 所在目录下的 default.ini 文件，文件内容说明：

【TARGET】标签：

- JTAGTYPE=USBICE; 配置在线调试器类型，当前仅支持 USBICE
- ICECLK=1200; 配置在线调试器工作频率为 12Mhz
- DDC=TRUE; 开启 CSKY 硬件快速下载通道

C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- CACHEFLAG=TRUE; 单步, 退出调试模式时是否刷 cache
- MTCRDELAY=10; 执行 mtr 指令的延时
- TARGETINIFILE=; 在获取 ICE 后执行的目标初始化脚本, 脚本类型为 GPIO 或 JTAG 脚本, 执行脚本后继续 Server 启动流程
- PRESETCDI=2/5; 指定获取 ICE 后操作目标板的模式, 2 线或 5 线
- PRERESET=FALSE; 开启在获取 ICE 后向目标板发起 NReset 信号
- TDESCXMLFILE=; 指定描述目标板寄存器信息的 Xml 文件
- NRESETDELAY=100; 设置 ckserver 发出 NReset 信号的时间长度
- TRESETDELAY=110; 设置 ckserver 发出 TReset 信号的时间长度
- RESETWAIT=50; 设置等待 Reset 操作完成的延时
- MULTICORETHREADS=TRUE; 设置调试 CK860MP 的模式
- DCOMMTYPE=LDCC; 开启检查 DCOMM 输出信息, 检查方式为 LDCC
- LOCALSEMIHOST=FALSE; 设置处理程序半主机请求由 ckserver 完成
- DEBUGARCH=CSKY; 设置当前连接的调试模块, CSKY/RISCV, 分别指代 CSKY HAD 和 RISCV DM
- DMSPEEDUP=TRUE; 读写 GPR 寄存器, 读写内存操作由 ICE 封装, 为 FALSE 时则有 ckserver 上层操作 JTAG 完成这些操作

#### 【SOCKETSERVER】

- ◆ SOCKETPORT=1025; 设置 ckserver 开启 Socket 服务的端口

## 2.4.4. 常用功能介绍

### 1. 打开/保存配置文件

Ckserver 支持使用已有的配置文件来完成 ckserver 配置[File->Open Config File]以及将当前配置保存为配置文件[File->Save Config File], 配置信息包括 CPU 频率、是否使用 ddc 等

### 2. target 配置窗口

除了上述使用已有配置文件, 还可以使用手动配置, [Setting->Target Setting]打开配置信息对话框。如图所示,

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

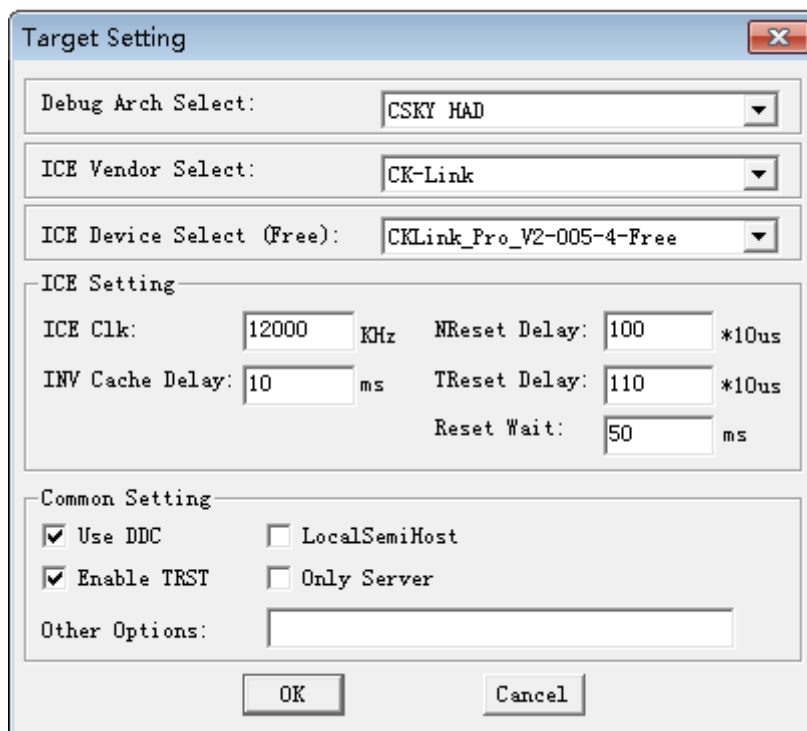


图 2-15 Target Setting 对话框

其中:

- Debug Arch Select: 选择调试架构, 可选择 CSKY HAD 和 RISC-V DM
- ICE Vendor Select: 选择不同厂商的 ICE 设备, 默认为 CK-Link
- ICE Device Select(Free): 指定 ICE 连接
- ICE Setting: 设置为使用 ICE 作为连接目标时的信息, 包括: ICE 频率, mtrr 延时;
- NReset Delay: 设置 NReset 延时, 确保 ICE 可以生成稳定的硬件复位 (nreset) 信号, 单位 10us, 默认为 1ms。
- TReset Delay: 设置 TReset 延时, 确保 ICE 可以产生稳定的复位信号, 用于复位 HAD 状态机, 单位 10us, 默认为 1.1ms。
- Reset Wait: 设置延时, 确保在目标板收到复位信号后, 目标板复位流程执行结束, 默认 50ms。
- Use DDC: 选择下载时是否使用 DDC 直通通道下载。
- Enable TRST: 执行 Reset Target 时是否执行 TRST。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- LocalSemiHost: 指当程序发生 semihosting 请求时是否由 ckserver 完成, 默认由 GDB 完成。

### 3. Socket 端口设置

[Target Setting->Socket Setting]打开 Socket 端口设置对话框, 如图所示,

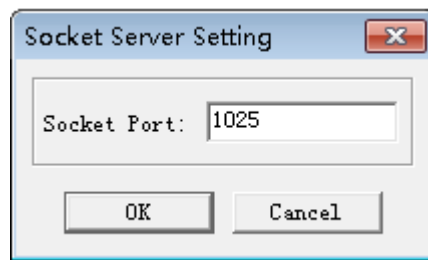


图 2-16 Socket Setting 对话框

手动输入端口号即可, 默认值为 1025 。

### 4. 固件升级对话框

[Tools->UpgradeFirmware]打开固件升级对话框, 如图所示,

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

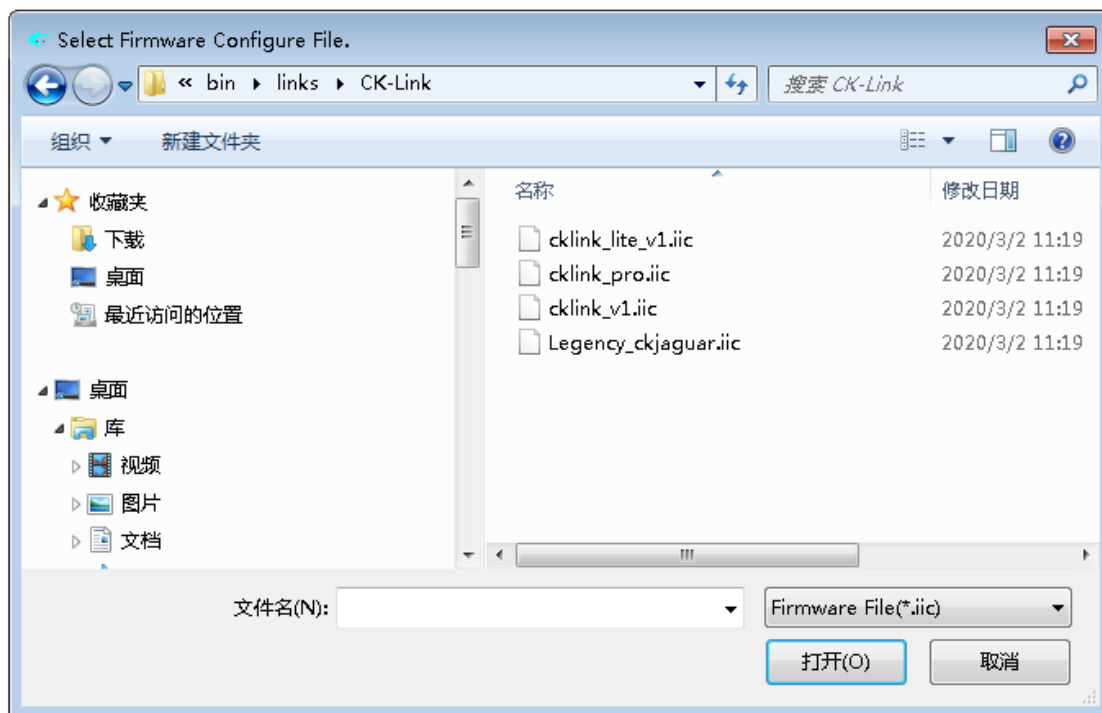


图 2-17 固件升级对话框

选择与你的 ICE 设备对应的的升级文件即可，升级文件与 ICE 盒子类型要保持一致,具体对应如下：

CKLINK\_LITE\_V2: cklink\_lite.hex

CKLINK\_PRO\_V1: cklink\_v1.iic

CKLINK\_PRO\_V2: cklink\_pro.iic

## 5. Had 寄存器操作窗口

[Tools->HAD Register]打开 HAD 寄存器的操作对话框，可以进行 HAD 寄存器的读写操作如图所示：

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

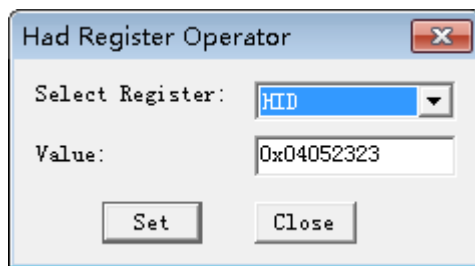


图 2-18 HAD 寄存器操作对话框

读操作:

Select Register 选择要读取的 HAD 寄存器; Value 会自动显示为该寄存器的值。

写操作:

Select Register 选择要写入的 HAD 寄存器, 在 Value 中写入想要写入的值, 点击 Set, 即可。

## 2.4.5. 运行说明

### 1. 步骤

- 点击“开始->所有程序->C-Sky ToolKit-> C-Sky Debugger-> CSkyDebugger, 此时会以默认配置打开 ckserver, 界面如图所示。
- 如果需要更改 ckserver 配置, 点击 RunDebuggerServer (运行/停止调试 ckserver) 断开连接, 然后按照“2.4.2 菜单栏工具栏介绍”进行配置。然后点击 RunDebuggerServer (运行/停止调试 ckserver) 重新连接。
- 运行 C-SKY GDB 应用程序进行调试:
  1. 使用 CSKY 工具链生成 CSKY elf 程序 a.out
  2. 启动 GDB, 比如 csky-\*-gdb a.out
  3. 根据 ckserver 界面上的提示, 在 GDB 的命令行输入连接命令, 比如 target remote 172.16.28.158:1025
  4. 当 GDB 连接上 ckserver 之后, 可以进行 GDB 常用的操作, 比如:

① load // 下载程序至开发板

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- ② `break main`           // 在 main 函数处设置断点
- ③ `continue`            // 运行程序
- ④ `info registers r0`    // 查看寄存器 r0
- ⑤ `print var_a`         // 查看程序变量 var\_a

## 5. GDB 的使用方法 GNU GDB 保持一致

- 如果使用 CDK 或 CDS，请参考开发环境所带的用户文档

## 2. 运行界面

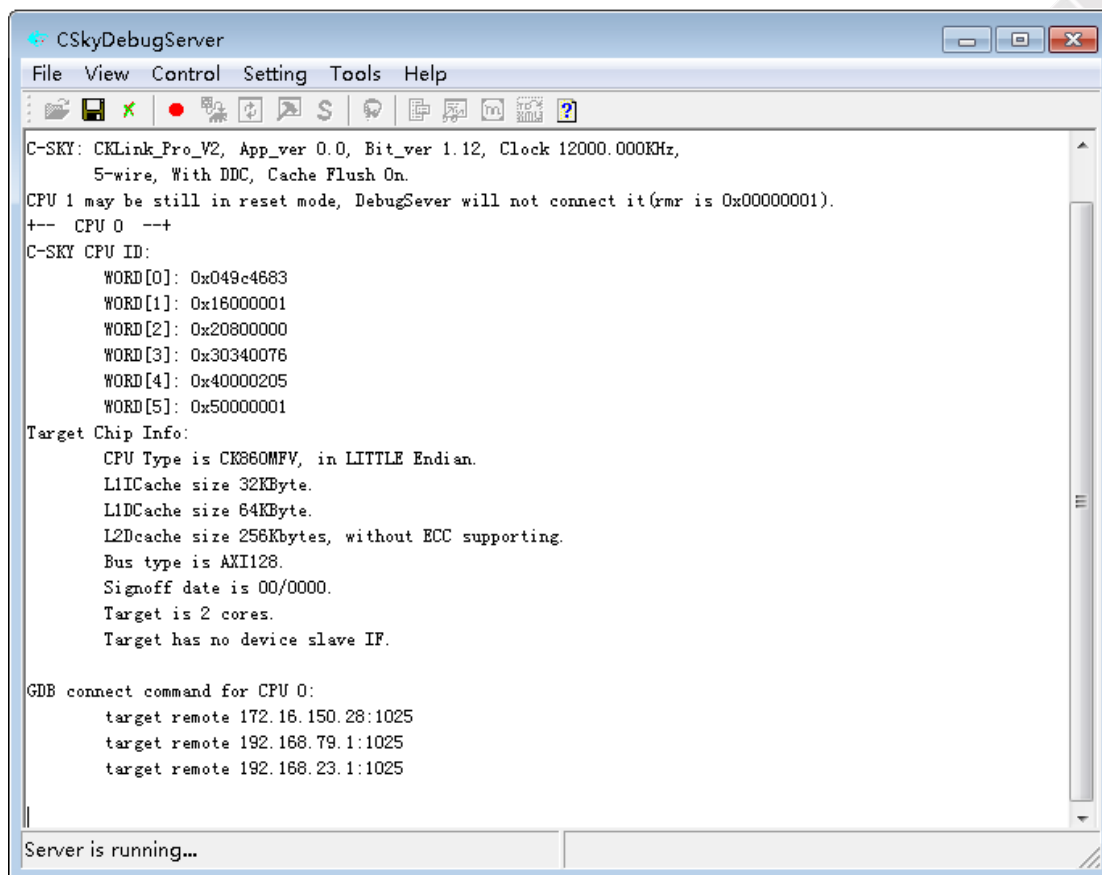


图 2-19 ckserver GUI 运行界面

## 3. 注意事项

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



(1) 如果运行 ckserver 时提示 ICE Upgrade, 选择“是”, 进行固件升级, 升级完成后需要重新插拔 ICE, 使用 ckserver 重新连接。

## 3. Linux 篇

Linux 版本的 ckserver 没有 GUI 版本, 只支持命令行输入, 所以 Linux 版本的操作同 windows 下的 console 版本的操作

### 3.1. ckserver 及 ICE 驱动安装

#### 3.1.1. 安装包获取

从 C-SKY 公司的 OCC 平台 [https://occ.t-head.cn/community/download\\_detail?id=616215132330000384](https://occ.t-head.cn/community/download_detail?id=616215132330000384) 或技术支持处 获取安装包, 安装包中包含 Windows 下的 CSKY-DebugServer-windows\*.exe 安装文件和 Linux 平台的两个 CSKY-DebugServer-linux-\*.sh 安装文件 (分别对应 32 位和 64 位)。

CSKY-DebugServer-linux-i386-\*.sh 是 32 位机的安装包, CSKY-DebugServer-linux-x86\_64-\*.sh 是 64 位机的安装包。请对应主机系统, 选择对应的安装包。

#### 3.1.2. ckserver 安装步骤

1. 在进行安装过程中, 用户需要获得 sudo 权限。
2. 通过命令 `chmod+x` 增加安装包的执行权限
3. 执行 `sudo ./CSKY-DebugServer-linux-*.sh -i` 开始安装
4. 给出提示 “Do you agree to install the DebugServer[yes/no]”, 输入 yes。
5. 系统提示设置安装路径 “Set full installing path:”

(1) 安装到用户指定目录: 用户输入安装路径 (绝对路径), 安装会给出提示 “This software will be installed to the path:(用户输入的路径)? [yes/no/cancel]:”, 确认路径无误, 输入 “yes” 并按下回车键。此时安装包将进行安装, 安装完成后将提示:

“Done!

You can use command “DebugServerConsole” to start DebugServerConsole!

(NOTE: The full path of ‘DebugServerConsole.elf’ is 用户输入的路径/C-Sky\_DebugServer) ”

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

(2) 默认路径安装: 直接按回车, 那么会给出提示 “This software will be installed to the default path: (/usr/bin/)?[yes/no/cancel]:”, 用户输入 yes, 软件会安装到默认路径 “/usr/bin/” 目录下。安装完成后提示

“Done !

You can use command “DebugServerConsole” to start DebugServerConsole !

(NOTE: The full path of ‘DebugServerConsole.elf’ is /usr/bin/C-Sky\_DebugServer) ”

## 3.2. 运行环境

ckserver 运行支持 Ubuntu 等大多数 Linux 平台, 输入输出设备要求具有 USB 接口。  
ckserver 配合 C-SKY GDB 使用, 支持所有版本。

## 3.3. 运行参数

同 “2.3.1 运行参数”。

## 3.4. Jtag 脚本配置功能说明

同 “2.3.2 Jtag 脚本配置功能说明”。

## 3.5. 运行说明

### 1. 步骤

- 安装完成后, 在任意目录下通过命令 “DebugServerConsole + 运行参数”, 来打开 ckserver, 如果不加运行参数则使用默认运行参数运行 ckserver。
- 运行 C-SKY GDB 应用程序进行调试, 运行 C-SKY GDB 应用程序进行调试:
  1. 使用 CSKY 工具链生成 CSKY elf 程序 a.out
  2. 启动 GDB, 比如 csky-\*-gdb a.out
  3. 根据 ckserver 界面提示, 在 GDB 的命令行输入连接命令, 比如 target remote 172.16.28.168:1025
  4. 当 GDB 连接上 ckserver 之后, 可以进行 GDB 常用的操作, 比如:

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- ⑥ `load` // 下载程序至开发板
- ⑦ `break main` // 在 `main` 函数处设置断点
- ⑧ `continue` // 运行程序
- ⑨ `info registers r0` // 查看寄存器 `r0`
- ⑩ `print var_a` // 查看程序变量 `var_a`

(1) GDB 的使用方法 GNU GDB 保持一致

- 如果使用 CDK 或 CDS，请参考开发环境所带的用户文档

2.运行界面与【2.3.3 运行说明】windows console 版本同

图 3-1 Linux ckserver 运行界面

### 3.注意事项

- (1) 在运行 `ckserver` 之前，查看 `/etc/hosts` 文件，查看是否已经存在主机 IP 和主机名，若不存在，则用户在首行添加。
- (2) 如果运行 `ckserver` 时提示 `ICE Upgrade`，选择“是”，进行固件升级，升级完成后需要重新插拔 ICE，使用 `ckserver` 重新连接。

## 4. 半主机功能说明

Semihosting(半主机)是指在调试过程中，使用 `host` 的输入输出替换设备端的输入输出的一种调试方式。例如 `host` 机器的键盘、磁盘读写、屏幕输出等输入输出设备都可以作为目标板的输入输出来使用。

通过 Semihosting 可以减少在开发过程中很多硬件设备的依赖，仅需要通过一种调试方式接入目标板即可。

semihosting 支持的常见的操作有

#### 1.open

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- 2.close
- 3.read
- 4.write
- 5.lseek
- 6.rename
- 7.unlink
- 8.stat
- 9.fstat
- 10.gettimeofday
- 11.isatty
- 12.system

CSKY 半主机使用步骤:

1. 编写 CSKY 裸程序，其中包含有文件操作，或输入输出
2. 编译工程时，编译器添加编译选项“-lsemi”，工具链版本要求大于等于 V3.8.x
3. 使用 CSKY 调试器下载程序并调试，ckserver 要求大于等于 V5.2.0
4. 运行程序，程序中涉及 Semihosting 操作将会有 GDB 完成。比如 printf 输出将在 gdb 界面输出，fopen 可以打开 GDB 所在主机的文件等。

Semihosting 操作默认由 GDB 完成。如果由 ckserver 完成，方法为在 ckserver 启动时添加【-local-semi/-ls】参数（UI 版本启动方式请查阅 2.4 章节），则程序中的 Semihosting 请求将由 ckserver 完成。比如 printf 输出将在 ckserver 的界面中输出，fopen 可以打开 ckserver 所在主机的文件等[默认情况下，Semihosting 的执行在 GDB 中完成，可通过[-local-semi/-ls]选项重设为在 ckserver 中执行。

注意：在 Windows 中加-local-semi/-ls 时，不支持 isatty 和 system 操作。

## 5. 调试输出功能说明

CSKY DCOM(Debug Communication)是基于 CSKY JTAG 通路支持调试过程中输出信息的通道，

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

目前仅支持含有 LDCC（CSKY Light Debug Communication Channel）功能的 CPU。

使用步骤，以包含 LDCC 的 CK802 为例：

1. 编写 CSKY 裸程序代码，实现 fputc 如下：

```
#define LDCC_DATA_P      0xe001105c      /* LDCC Register. */

#define LDCC_BIT_STATUS  0x80000000      /* LDCC Status bit. */

int fputc (int ch, FILE *f)

{

    volatile unsigned int *pdata = LDCC_DATA_P;

    /* Waiting for data read. */

    while (*pdata & LDCC_BIT_STATUS);

    *pdata = ch;

    return 0;

}
```

2. 使用 CSKY ELF 工具链编译代码，生成 Elf 文件
3. 启动 ckserver，执行” DebugServerConsole.exe -dcomm=ldcc”，UI 版本参考 2.4.3 章节说明
4. 使用 CSKY GDB 连接 ckserver 并全速运行程序
5. 则程序中使用 printf 输出的信息将通过 LDCC 传输至 ckserver 窗口显示

## 6. 命令行功能说明

CSKY 提供了命令行进行 target 调试手段，用户可以在 ckserver 启动后，通过相关命令实现读写 HAD 寄存器，读写 CPU 寄存器，读写内存等操作。支持的操作如下：

命令	描述	例子
----	----	----

### C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

p/print	1. 打印寄存器(\$)/内存(*)值, 内存打印仅支持 Word 大小。已知位域含义的寄存器将会按照位域显示该寄存器的值。 2. target, 打印 target 信息	p \$psr p *0x10000000 p target
set	设置寄存器(\$)/内存(*)值	set \$r0=0x10000000 set *0x10000000=0x1
reset	执行 Nreset 复位操作	reset
q/quit	退出 ckserver	q
help	打印 help 信息	help

该功能仅在 DebugServerConsole 版本提供, 用户启动 DebugServerConsole 后即可使用上述命令进行需要的操作, UI 版本 DebugServer 已经在界面上实现了上述操作。

附寄存器列表:

CSKY HAD 调试架构:

HAD 寄存器: hid, htcr, mbca, mbcb, pcfifo, baba, babb, bama, bamb, cpuscr, bypass, hcr, hsr, ehcr, wbbcr, psr, pc, ir, csr, dcccdata, ldcccdata, ddcaddr, ddcdata, bsel, hcdi, cpusel, cpust, hacr

RISCV DM 调试架构:

Debug Module 0.13 寄存器: data0, data1, data2, data3, data4, data5, data6, data7, data8, data9, data10, data11, dmcontrol, dmstatus, hartinfo, haltsum1, hawindowssel, hawindow, abstractcs, command, abstractauto, confsttptr0, confsttptr1, confsttptr2, confsttptr3, nextdm, progbuf0, progbuf1, progbuf2, progbuf3, progbuf4, progbuf5, progbuf6, progbuf7, progbuf8, progbuf9, progbuf10, progbuf11, progbuf12, progbuf13, progbuf14, progbuf15, authdata, haltsum2, haltsum3, sbaddress3, sbcs, sbaddress0, sbaddress1, sbaddress2, sbdata0, sbdata1, sbdata2, sbdata3, haltsum0.

Debug Module 0.11 寄存器: dmcontrol, dminfo, authdata0, authdata1, serdata, serstatus, sbaddress0, sbaddress1, sbdata0, sbdata1, haltsum, sbaddress2, sbdata2, sbdata3, cleardebint, serhaltnot, serinfo, sersend0, serrecv0, serstat0.

CPU 寄存器列表: CPU 寄存器列表请查看 DebugServer 安装目录 tdescriptions 文件夹中对应的寄存器描述。

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

## 7. XML 使用说明

### 7.1. 简介

基于 GDB 本身对 XML 文件描述寄存器的支持，CSKY-GDB 实现了一套 XML 文件描述调试目标寄存器的扩展机制。C-SKY-GDB 根据 XML 描的寄存器名称，个数，寄存器组信息等来组建内部寄存器运行需求。在此机制下，用户可以根据接口，灵活方便的使用 XML 文件描述目标的寄存器信息。

### 7.2. XML 文件格式

#### 7.2.1. 编写规则

GDB 内部对描述文件规则有：

- ◆ 描述文件遵守 XML 文件规则
- ◆ XML 文件中 **target** 为根元素，具有 **version** 属性，默认为 1.0
- ◆ Target 下同级的元素有 **architecture**，**feature**
- ◆ Architecture 的值表明该文件描述的体系结构，比如 **arm**，**mips**，**csky**
- ◆ Feature 有 **name** 属性，该属性表明了对寄存器的划分
- ◆ Feature 下主要元素为 **reg**，**reg** 的属性有 **name**（寄存器名称），**bitsize**（位宽），**regnum**（GDB 内部编号），**type**（数据类型），**group**（该寄存器所属组名），**save-restore**（GDB 内部使用的属性）；这些属性中 **name** 和 **bitesize** 为必要属性，其他的为可选属性
- ◆ Feature 下还有 **vector**，**struct**，**union**，**flags** 元素，这些元素是作为对目标寄存器类型的描述的扩展，在附录中描述

CSKY-GDB 在上述规则的基础上补充如下规则：

- Target 具有 **version** 属性，默认为“1.0”，而当前 CSKY-GDB 只支持“1.0”
- Architecture 中必须为 **csky**
- Feature 名称为 **org.gnu.csky.abiv1.xxx** 或 **org.gnu.csky.abiv2.xxx**（xxx 为可变内容，只是对寄存器的划分）

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



- 每个 Feature 中的第一个 reg 需要有 regnum 属性，之后 regnum 连续的 reg 可没有该属性，自动根据上一个 reg 的 regnum 加 1；如过出现不连续的，则需要有 regnum 属性
- 多个寄存器名称使用一个 regnum，此时，除过第一个 regnum 之后的 regnum 对应的 reg 将作为该寄存器别名支持，此时该寄存器除了 name 和 regnum 属性被使用外，其他属性将被忽略
- 如果 reg 需要在 group 中显示，则需有 group 属性
- Reg 的属性值不应出现特殊字符
- 对 reg 添加 regs 属性，描述寄存器组合的 pseudo 寄存器
- 对 reg 添加 addr 属性，描述内存作为 pseudo 寄存器，由 type 属性决定内存长度
- Pseudo 寄存器的 reg 描述编写在特定名称为“org.gnu.csky.pseudo”的 feature 下
- Name, group 属性的合法字符范围为“a/A-z/Z, 0-9, \_”，且 name 的长度不应大于 15 个字符

### 7.2.2. 举例描述

CSKY 描述目标寄存器信息的 xml 文件样本如下：

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



```

<?xml version="1.0"?>

<!DOCTYPE target SYSTEM "gdb-target.dtd">

<target version="1.0">

<architecture>csky</architecture>

  <feature name="org.gnu.csky.abiv1.gpr">

    <reg name="r0" bitesize="32" regnum="0" type="uint32"
group="gpr" save-restore="yes"/>

    <reg name="r1" bitesize="32" regnum="1" group="gpr"/>

    <reg name="r2" bitesize="32" regnum="2" group="gpr"/>

    ...

  </feature>

  <feature name="org.gnu.csky.pseudo">

    <reg name="r01" bitesize="64" type="uint64" group="pseudo" regs
="0,1"/>

    <reg name="memr" bitesize="32" type="uint32" addr="0x1234"
group="pseudo"/>

    ...

  </feature>

</target>

```

注释:

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- XML 文件中 target 为根元素，version 为 1.0
- architecture 中表明该文件描述的是 csky 体系结构
- feature 的 name 为 org.gnu.csky.abiv1.gpr 表明该 feature 描述的是 CSKY 体系结构下 abiv1 的通用寄存器信息
- 描述的第一个寄存器的名称为 r0，位宽为 32，在 GDB 内部编号为 0，类型为 uint32，属于寄存器组 gpr
- 描述的第二个寄存器的名称为 r1，位宽为 32，在 GDB 内部编号为 1
- 描述的第三个寄存器的名称为 r2，位宽为 32，在 GDB 内部编号为 2
- 第二个 feature 的 name 为“org.gnu.csky.pseudo”表明该 feature 是作为描述 pseudo 寄存器的特定 feature
- 第二个 feature 的第一个寄存器名称为 r01，位宽 64 位，类型为 uint64，是由真实的物理寄存器 r0，r1 组合而成，属于寄存器组 pseudo
- 第二个 feature 的第二个寄存器名称为 memr，位宽为 32，其值由内存地址 0x1234 开始后的 32bits 构成，类型为 unsigned int，属于寄存器组 pseudo

### 7.2.3. 寄存器编号

在 XML 文件中使用到的寄存器编号，即 **regnum** 的内容指表格【错误!未找到引用源。】和【错误!未找到引用源。】中的 **remote 编号项** 内容。下列表格中对目前支持的和可预期支持的寄存器做了编号，对寄存器名称并未一一列出。在 CSKY-GDB 支持 XML 后，寄存器的名称可以根据需求修改。

#### 7.2.3.1. Abiv1 寄存器

Abiv1 目前的寄存器有：

- ◆ General registers r0~r15
- ◆ Optional registers r0~r15
- ◆ Cr0~cr31 控制寄存器
- ◆ Hi,Lo
- ◆ FPU
- ◆ PC
- ◆ FPC-Control

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

◆ MMU

寄存器编号为:

表 7-1 abiv1 寄存器编号

寄存器名称	GDB 编号	备注
R0~r15	0~15	列表中的 remote 编号均依照当前 GDB 对寄存的编号处理编写
Hi,Lo	20, 21	
FPU	24~55	
PC	72	
Optional registers r0~r15	73~88	
Cr	89~119	没有 cr31
FPC-Control (命名为 cp1cr0~cp1cr6)	121~127	
MMU (命名为 cp15cr0~cp15cr16 和 cp15cr29~cp15cr31)	128~147	总共 20 个 MMU 寄存器, remote 编号连续

### 7.2.3.2. Abiv2 寄存器编号

Abiv1 目前的寄存器有:

- ◆ General registers r0~r31
- ◆ Optional registers r0~r15
- ◆ Hi,lo

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- ◆ FPU/VPU
- ◆ Profiling
- ◆ PC
- ◆ Cr\_Bank0
- ◆ Cr\_Bank1
- ◆ Cr\_Bank3
- ◆ Cr\_Bank15

寄存器编号为:

表 7-2 abiv2 寄存器编号

寄存器名称	GDB 编号	备注
R0~r15	0~15	
R16~r31	16~31	
Hi, lo	36, 37	
FPU/VPU	40~71	
PC	72	
Ar0~ar15	73~88	
Cr0~cr31	89~120	
FPU/VPU_CR	121~123	
Usp	127	
Mmu (bank15)	128~136	
Prof-soft-general	140~143	
Prof-cr	144~157	
Prof-arch	160~174	
Prof-exten	176~188	
Bank1	189~220	

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

Bank3	221~252	
Bank15 剩余	253~275	
Bank2	276~307	
Bank4	308~339	
Bank5	340~371	
Bank6	372~403	
Bank7	404~435	
Bank8	436~467	
Bank9	468~499	
Bank10	500~531	
Bank11	532~563	
Bank12	564~595	
Bank13	596~627	
Bank14	628~659	
Bank16	660~691	
Bank17	692~723	
Bank18	724~755	
Bank19	756~787	
Bank20	788~819	
Bank21	820~851	
Bank22	852~883	
Bank23	884~915	
Bank24	916~947	
Bank25	948~979	
Bank26	980~1011	

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

Bank27	1012~1043	
Bank28	1044~1075	
Bank29	1076~1107	
Bank30	1108~1139	
Bank31	1140~1171	
Cr16~cr31	1172~1187	

## 7.2.4. 扩展的 TEE 寄存器描述

### 7.2.4.1. 扩展简述

在 CSKY TEE（安全）编程模型中，存在同一个寄存器在 TEE 和 REE 两个世界中各有一份。这些寄存器均需要切换世界或通过寄存器映射来读写，具体可查看 CSKY TEE 的相关手册。GDB 配合 DebugServer 的 xml 文件可辅助用户在任一世界时，查看当前以及另一个世界的寄存器信息（版本需求 CSKY GDB >= V3.10.0, DebugServer >= V4.5.0），由 DebugServer 辅助切换世界并读取对应世界的寄存器。

### 7.2.4.2. 编写规则

1. TEE 寄存器描述功能添加在 `pseudo` 寄存器内部中，故其遵循【7.2.1 错误!未找到引用源。】中 `pseudo` 寄存器相关的所有规则
2. TEE 寄存器使用“`env`”属性表明该寄存器属于 TEE 还是 REE 的寄存器，属性值只有“`ree`”，“`tee`”可选
3. 使用“`regs`”属性来表明对应的物理寄存器编号
4. `bytesize`, `type` 根据实际属性填写

### 7.2.4.3. 举例说明

下面是以 CK810T 为例的寄存器描述文件示例(用户可在 DebugServer V5.4.0 及之后的版本安装目录中，查看 `tdescriptions` 文件夹中的文件做参考)：

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

```

<?xml version="1.0"?>

<target>

  <architecture>csky</architecture>

  <feature name="org.gnu.csky.abiv2.gpr">

    <reg name="r0"  bitsize="32" regnum="0"  group="gpr"/>
    <reg name="r1"  bitsize="32" regnum="1"  group="gpr"/>
    <reg name="r2"  bitsize="32" regnum="2"  group="gpr"/>
    <reg name="r3"  bitsize="32" regnum="3"  group="gpr"/>
    <reg name="r4"  bitsize="32" regnum="4"  group="gpr"/>
    <reg name="r5"  bitsize="32" regnum="5"  group="gpr"/>
    <reg name="r6"  bitsize="32" regnum="6"  group="gpr"/>
    <reg name="r7"  bitsize="32" regnum="7"  group="gpr"/>
    <reg name="r8"  bitsize="32" regnum="8"  group="gpr"/>
    <reg name="r13" bitsize="32" regnum="13" group="gpr"/>
    <reg name="r14" bitsize="32" regnum="14" group="gpr"/>
    <reg name="r15" bitsize="32" regnum="15" group="gpr"/>
    <reg name="pc"  bitsize="32" regnum="72"/>

  </feature>

  <feature name="org.gnu.csky.abiv2.cr">

    <reg name="psr"  bitsize="32" regnum="89"  group="cr"/>
    <reg name="vbr"  bitsize="32" regnum="90"  group="cr"/>
    <reg name="epsr" bitsize="32" regnum="91"  group="cr"/>
    <reg name="epc"  bitsize="32" regnum="93"  group="cr"/>
    <reg name="ss4"  bitsize="32" regnum="99"  group="cr"/>
    <reg name="gcr"  bitsize="32" regnum="100" group="cr"/>
    <reg name="gsr"  bitsize="32" regnum="101" group="cr"/>
  
```

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

```

<reg name="cpuid"  bitsize="32" regnum="102" group="cr"/>
<reg name="ccr"   bitsize="32" regnum="107" group="cr"/>
<reg name="capr"  bitsize="32" regnum="108" group="cr"/>
<reg name="pacr"  bitsize="32" regnum="109" group="cr"/>
<reg name="prsr"  bitsize="32" regnum="110" group="cr"/>
<reg name="chr"   bitsize="32" regnum="120" group="cr"/>

</feature>

<feature name="org.gnu.csky.abiv2.tee">

  <reg name="nt_usp"  bitsize="32" regnum="127" group="ree"/>

  <reg name="ebr"    bitsize="32" regnum="190" group="cr"/>

  <reg name="dcr"    bitsize="32" regnum="229" group="cr"/>

  <reg name="t_usp"  bitsize="32" regnum="228" group="tee"/>

  <reg name="t_pcr"  bitsize="32" regnum="230" group="tee"/>

</feature>

<feature name="org.gnu.csky.pseudo">

  <reg name="t_psr"   bitsize="32" regs="89"   group="tee" type="int32" env="tee"/>
  <reg name="t_vbr"   bitsize="32" regs="90"   group="tee" type="int32" env="tee"/>
  <reg name="t_epsr"  bitsize="32" regs="91"   group="tee" type="int32" env="tee"/>
  <reg name="t_epc"   bitsize="32" regs="93"   group="tee" type="int32" env="tee"/>
  <reg name="t_ebr"   bitsize="32" regs="190"  group="tee" type="int32" env="tee"/>
  <reg name="nt_psr"  bitsize="32" regs="89"   group="ree" type="int32" env="ree"/>
  <reg name="nt_vbr"  bitsize="32" regs="90"   group="ree" type="int32" env="ree"/>
  <reg name="nt_epsr" bitsize="32" regs="91"   group="ree" type="int32" env="ree"/>
  <reg name="nt_epc"  bitsize="32" regs="93"   group="ree" type="int32" env="ree"/>
  <reg name="nt_ebr"  bitsize="32" regs="190"  group="ree" type="int32" env="ree"/>

</feature>

```

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



</target>

根据上面的 xml 文件描述，用户在 CSKY GDB 层面查看 psr 即为当前世界的 psr，查看 t\_psr 则为安全世界的 psr，查看 nt\_psr 则为非安全世界的 psr。也就是说 psr 可能等于 t\_psr，也可能等于 nt\_psr，依据当前所处的世界而定。

## 7.3. 版本及使用说明

Ckserver 的 Proxylayer version 高于 1.8 是，支持与 C-SKY GDB 的 XML 文件交互。

ckserver 在 Proxylayer version 大于等于 1.8 时支持了与 GDB 进行 xml 文件交互的命令协议，即当新版本 gdb 在 target remote ip:port 命令连接 ckserver 成功后并获取 ckserver 的相关版本信息。Gdb 发现 ckserver 的 proxylayer 版本大于等于 1.8，则向 ckserver 发起获取 xml 文件的请求。此时 ckserver 则根据用户指定的 xml 文件路径，打开并发送文件内容给 GDB。

ckserver 分别在 UI 版和 Console 版给出用户指定 xml 文件的接口，下面将对指定文件接口做详细描述。

### 7.3.1. UI 版

UI 版 ckserver 提供了在启动配置文件（default.ini）中指定 XML 文件和界面窗口指定 XML 文件的方式。

在 default.ini 中：

Default.ini 作为 UI 版 ckserver 的默认配置启动文件，在该文件中提供了用户设置 xml 文件路径的变量“TDESCXMLFILE”。用户可将编写好的 xml 文件路径赋值于该变量，当 GDB 发起获取描述目标描述信息的 xml 文件时，ckserver 将打开用户指定的 xml 文件，并与 GDB 进行交互。

当用户使用 UI 窗口修改过 xml 文件路径后，ckserver 关闭时，将提示用户是否修改默认的配置。用户选择是，则按照当前 ckserver 的相关配置保存成配置文件。

在 UI 窗口中：

UI 版设置窗口指在 UI 版 ckserver 的窗口界面上提供用户选择 xml 文件的接口。目前，在 setting 菜单栏下添加 TDFile setting 选项，如下图：

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

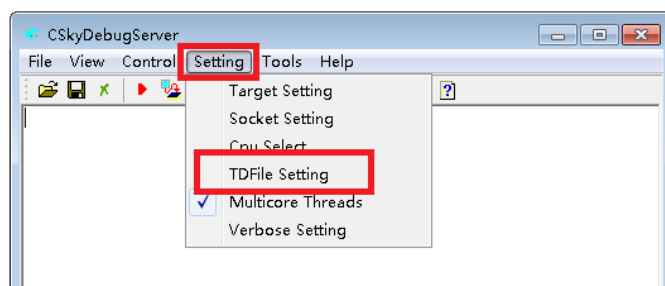


图 7-1 TDFile Setting 菜单选项

在工具条中添加快捷启动选择 xml 文件选择按钮，如下图：

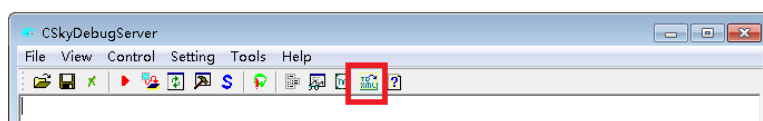


图 7-2 TDFile Setting 快捷按钮

以上两种方式点击后均可弹出 xml 选则对话框，如下图：



图 7-3 Target Description File 对话框


之后，点击 Browse 按钮选择本地描述目标的 xml 文件，确认后点击 Ok 按钮。

### 7.3.2. Console 版

Ckserver Console 版添加启动参数“-tdescfile +filepath”选项来指定描述目标寄存器的 xml 文件。

**Windows 环境中：**

#### 1) 建立 Console 版 ckserver 的快捷方式

	DebugServerConsole.exe	2018/6/15 17:35	应用程序	316 KB
	DebugServerConsole.exe - 快捷方式	2018/6/19 10:14	快捷方式	2 KB

#### 2) 右击打开快捷方式并点击属性

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

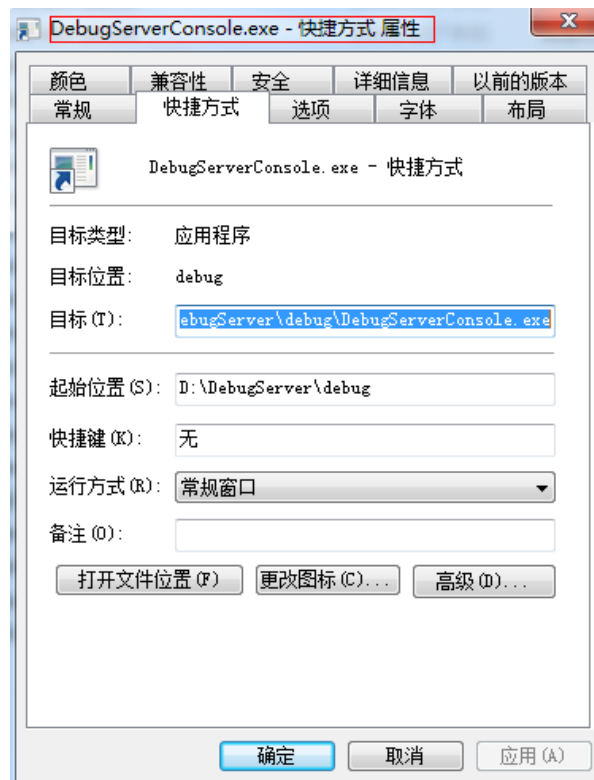


图 7-4 DebugServer 快捷方式属性

- 3) 在快捷方式页面的目标栏程序后面添加 **-tdescfile xmlpath**, xmlpath 为用户指定的 xml 文件路径

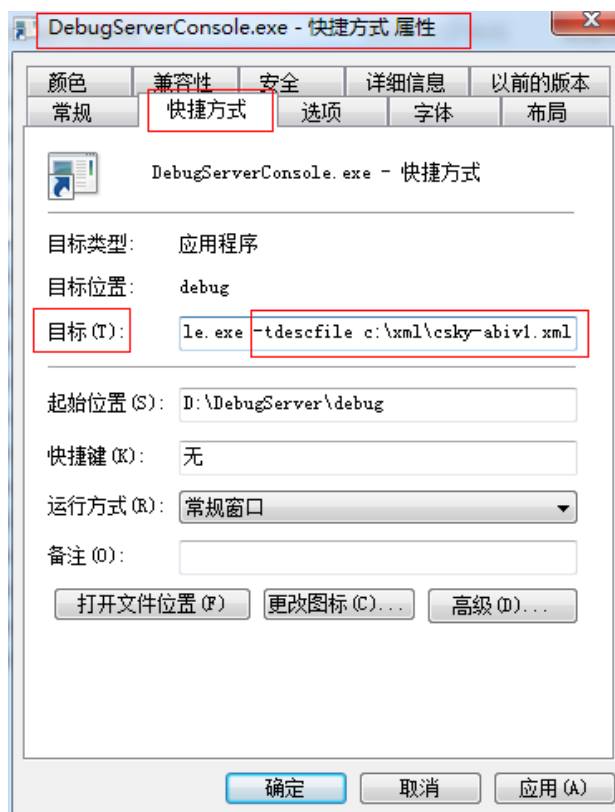


图 7-5 为快捷方式添加启动参数

- 4) 确认后点击确定按钮，启动该快捷方式，使用 **csky-gdb** 连接后进行正常调试。

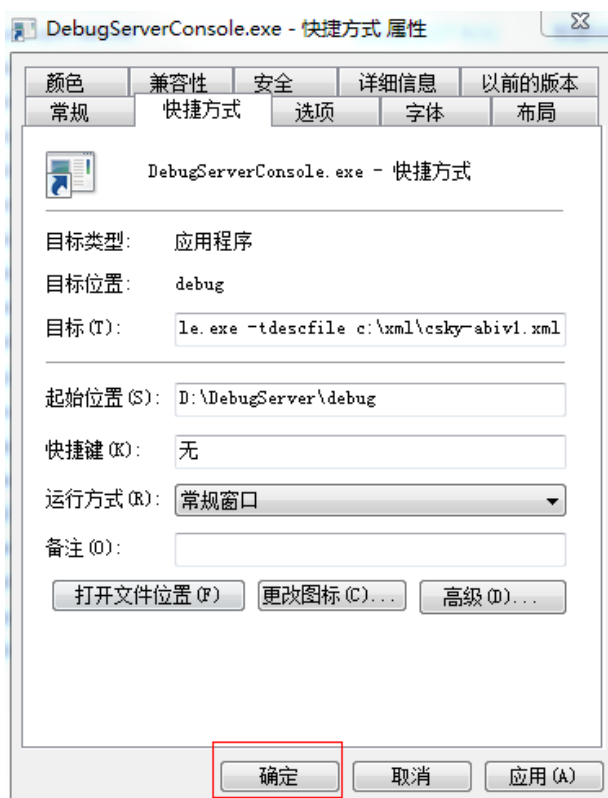


图 7-6 修改启动参数后点击确认按钮

Linux 环境中：

使用 `sudo ./Debugserver.elf -tdescfile /home/xxx/gdbxml/csky-abiv1.xml` 启动 linux 版 ckserver，之后使用 C-SKY GDB 连接调试。

## 8. 多核调试操作使用说明

### 8.1. 简介

C-SKY CK860MP 是面向嵌入式系统和 SoC 应用领域的 32 位超高性能嵌入式多核心处理器，调试采用多个核单个 JTAG 接口的调试框架，通过一个共享的 JTAG 接口访问各个 CORE 的辅助调试单元（HAD），触发 CORE 进出调试模式和访问处理器资源。

同时 CK860MP 设置了一个集中的事件传输模块（ETM）用于支持多核之间调试事件（进入调试和退出调试）的传输。当 CK860 核心在接收到 ICE 发送的调试命令产生调试进入或者调试退出事件时，同步将该事件发送到 ETM，由 ETM 决定是否将该事件转发给其他的 CORE，

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

实现多个核之间进行进入和退出调试信号交互，响应的目的。

即当一个核进入或退出调试模式时，可以选择将该信号发送给其他核。其他核可以选择是否响应该信号，同步进入或退出调试模式。硬件模型如下图：

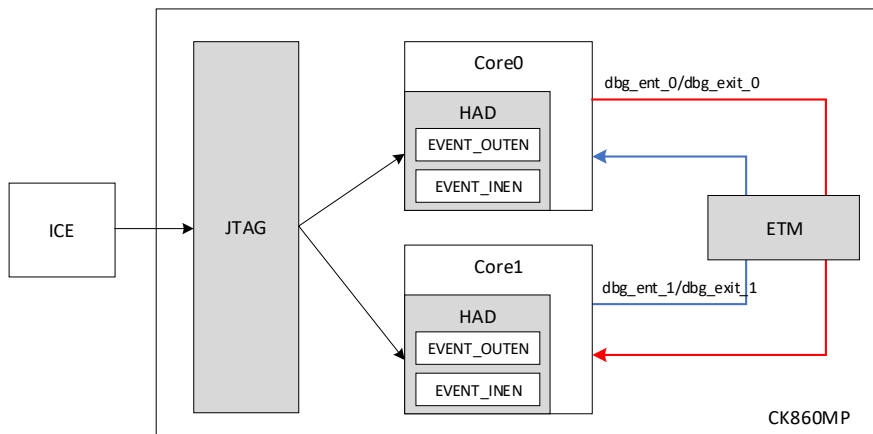


图 8-1 多核调试整体框架

同时 CSKY 实现 RISC-V 多核 C910MP 的 ETM 模块与 C860MP 的实现一致，调试方式也一致。

针对多核调试框架，CSKY 调试软件同步支持了两种调试模式：多核单个端口，多核多个端口，下面就两种调试模式做介绍。

## 8.2. 调试环境需求

### 硬件需求：

1. Ck860MP 开发板
2. CSKY ICE CKLINK\_PRO\_V2 盒子一个，配套排线，USB 线
3. PC 机，windows 系统安装有 mingw 或系统为 linux 系统

### 软件需求：

1. Csky-\*abiv2\*-gdb V3.6.x 工具链
2. DebugServer V4.4.0 软件（根据环境获取 windows 版本或 linux 版本）

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

### 8.3. 多核单端口模式

在该模式下，DebugServer 将只开启一个服务端口供 CSKY-GDB 连接。CSKY-GDB 通过“target remote ip:port”方式连接上 DebugServer 后，DebugServer 将多个核的信息封装为线程信息反馈给 CSKY-GDB。用户在 CSKY-GDB 端可以以 Thread 的方式查看，调试多个 CPU。

该模式下，多个 CPU 将互相发送并响应其他核的调试信号，意为其中一个 CPU 进入或退出调试模式时，其他 CPU 同步进入或退出调试模式。

模型如下：

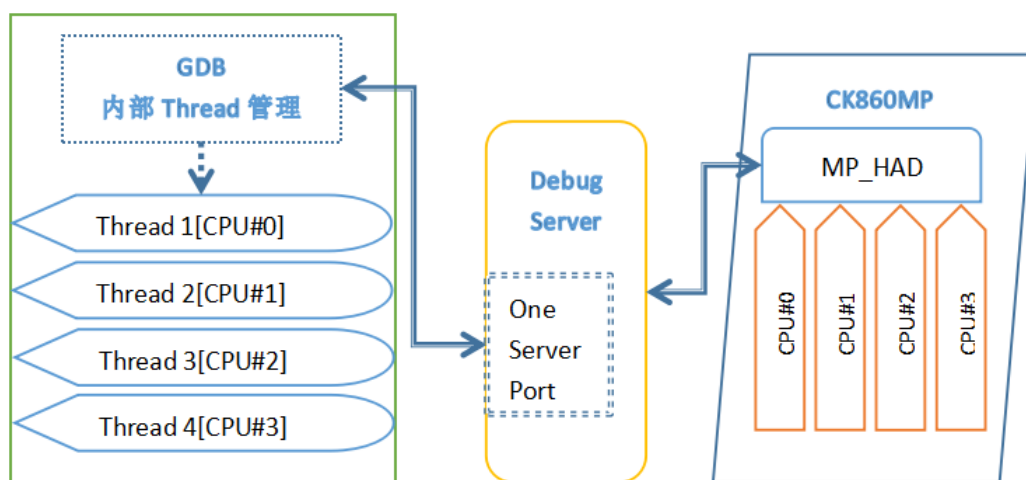


图 8-2 多核单端口模型图

#### 8.3.1. 操作步骤

此处以 2 个核的 CK860MP 为例：

1. 给开发板上电，连接 ICE，连接 ICE 的 USB 线连接至 PC 机
2. 运行 DebugServer，显示界面如下：
  - (1) UI 版本界面

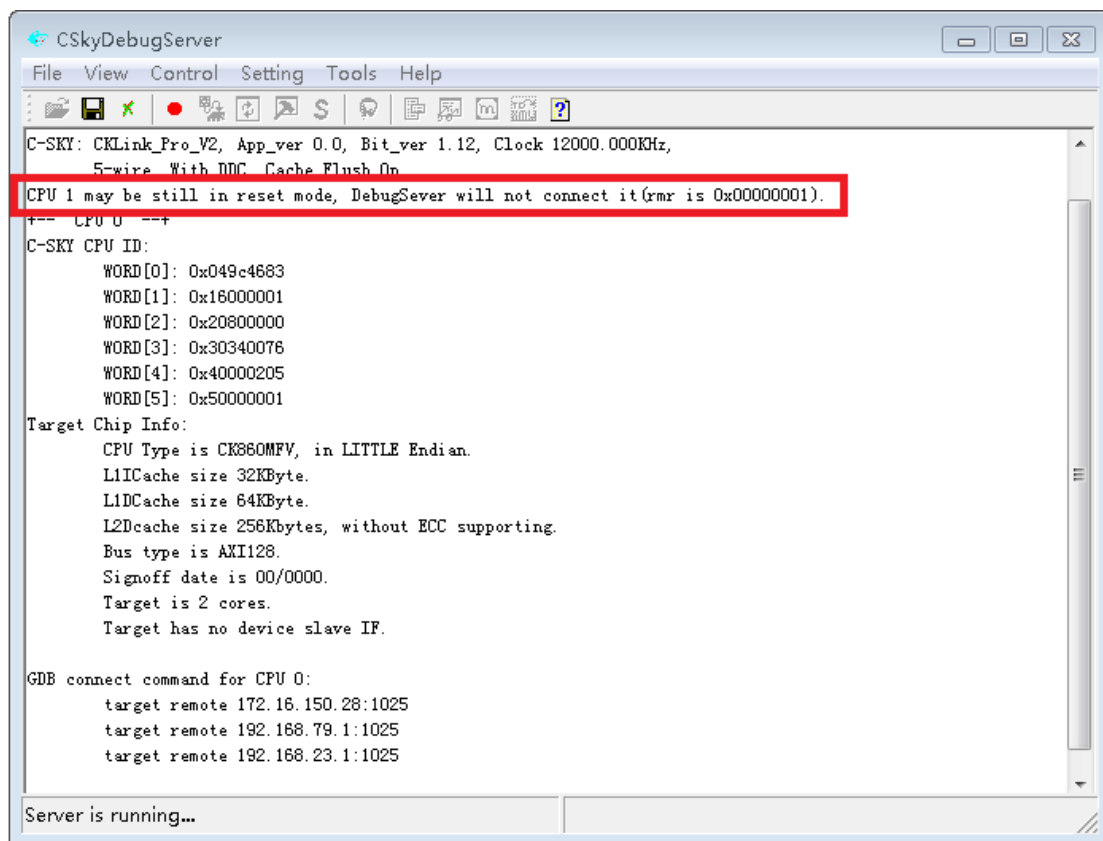


图 8-3 UI 版本 DebugServer 首次连接刚上电的 CK860MP

(2) Console 版本界面（linux 版本同）与 Windows UI 版本一致。

图 8-4 Console 版本 DebugServer 首次连接刚上电的 CK860MP（linux 同）

此时 DebugServer 正常打印 CPU 0 的 CPUID 信息，并打印 CPU 1 无法正常操作，该提示信息正常。原因是第一次上电的状态下，除过 CPU 0 之外的其他核均处于复位状态，故 JTAG 无法获取到 CPU 1 的信息，打印无法操作。

### 3. 启动 GDB 唤醒 CPU 1:

- (1) 启动 csky-\*abiv2\*-gdb
- (2) 在 gdb 的命令行输入 "target remote ip:port" (DebugServer 界面显示的连接命令)
- (3) 连接成功后，命令行输入：set \$cr29 = 3，唤醒 CPU 1（cr29 的描述还请查阅 CK860MP 用户手册）

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



```

GNU gdb (C-SKY Tools V3.7.4-ck805 Minilibc abiv2) 7.12
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=csky-elfabiv2".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(cskygdb) target remote 172.16.150.77:1025
Remote debugging using 172.16.150.77:1025
warning: No executable has been specified and target does not support
determining executable automatically. Try using the "file" command.
0x1fbd819c in ?? ()
(cskygdb) set $cr29=0x3
(cskygdb) quit
A debugging session is active.

        Inferior 1 [Remote target] will be detached.

Quit anyway? (y or n) y
Detaching from program: , Remote target
Ending remote debugging.
[toolsbuild@yulinglong ~]$

```

图 8-5 启动 GDB 并唤醒 CPU 1

- (4) 命令行输入: quit, 退出 gdb
4. 重新启动 DebugServer, 识别多核
- (1) UI 版 DebugServer 操作:
  - ① 点击暂停按钮

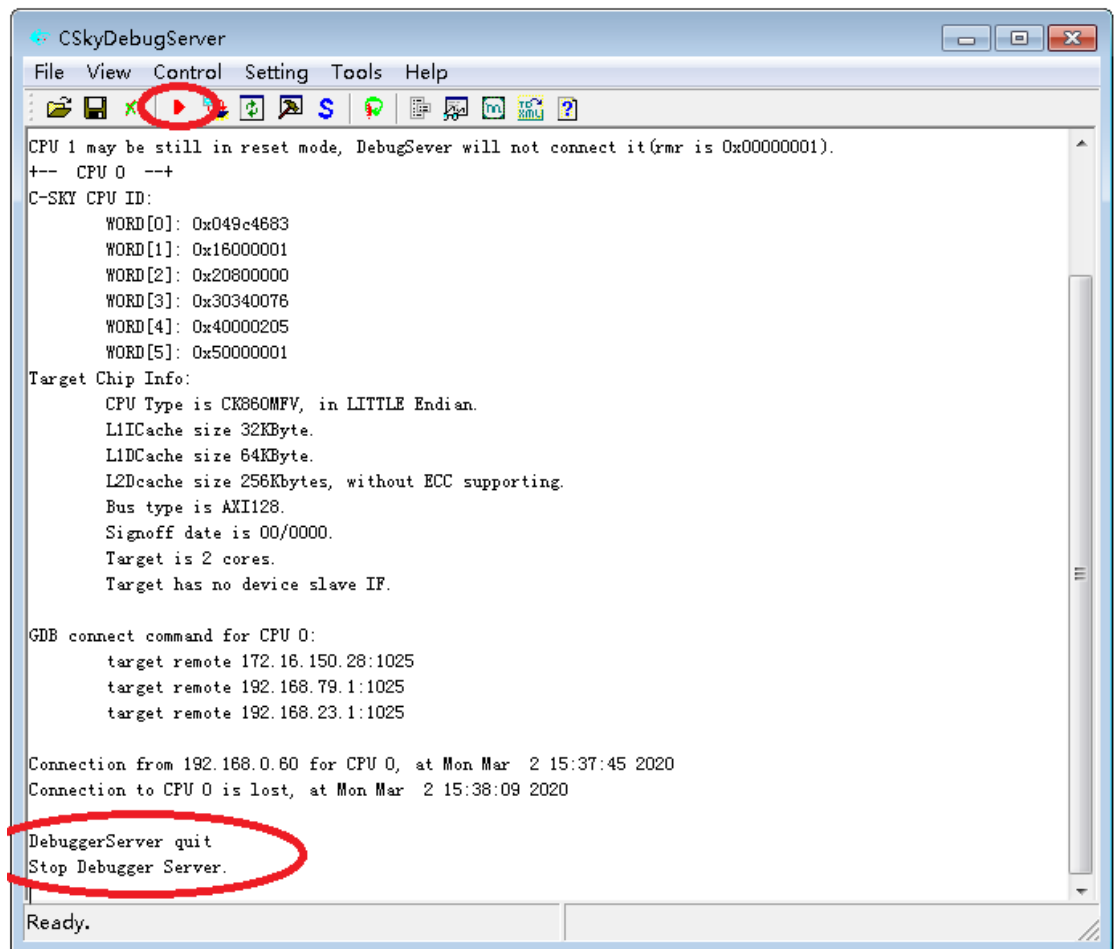


图 8-6 断开 UI 版本 DebugServer

② 重新连接，界面显示如下

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

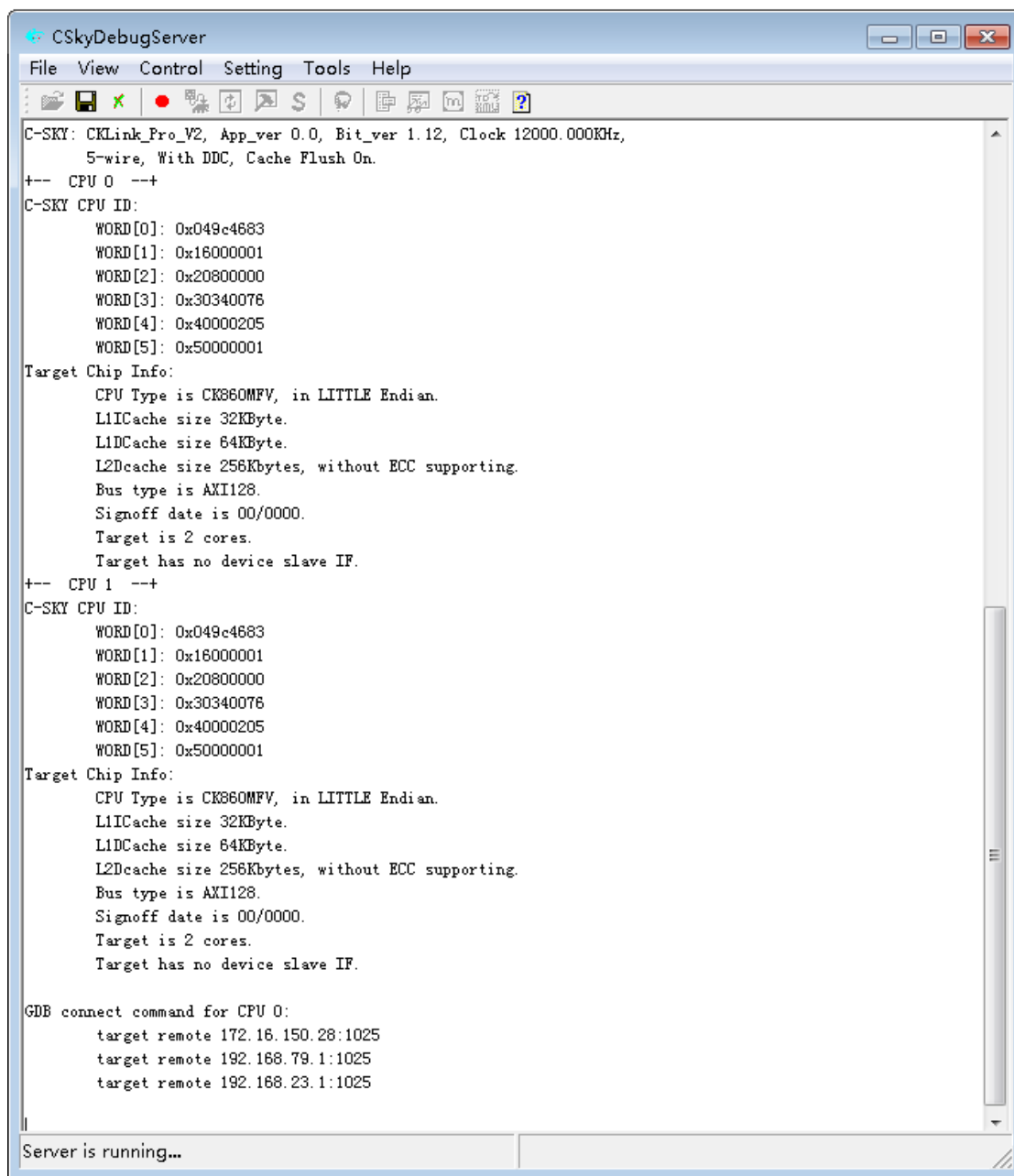


图 8-7 U 版本 DebugServer 已多核单端口连接 CK860MP 显示

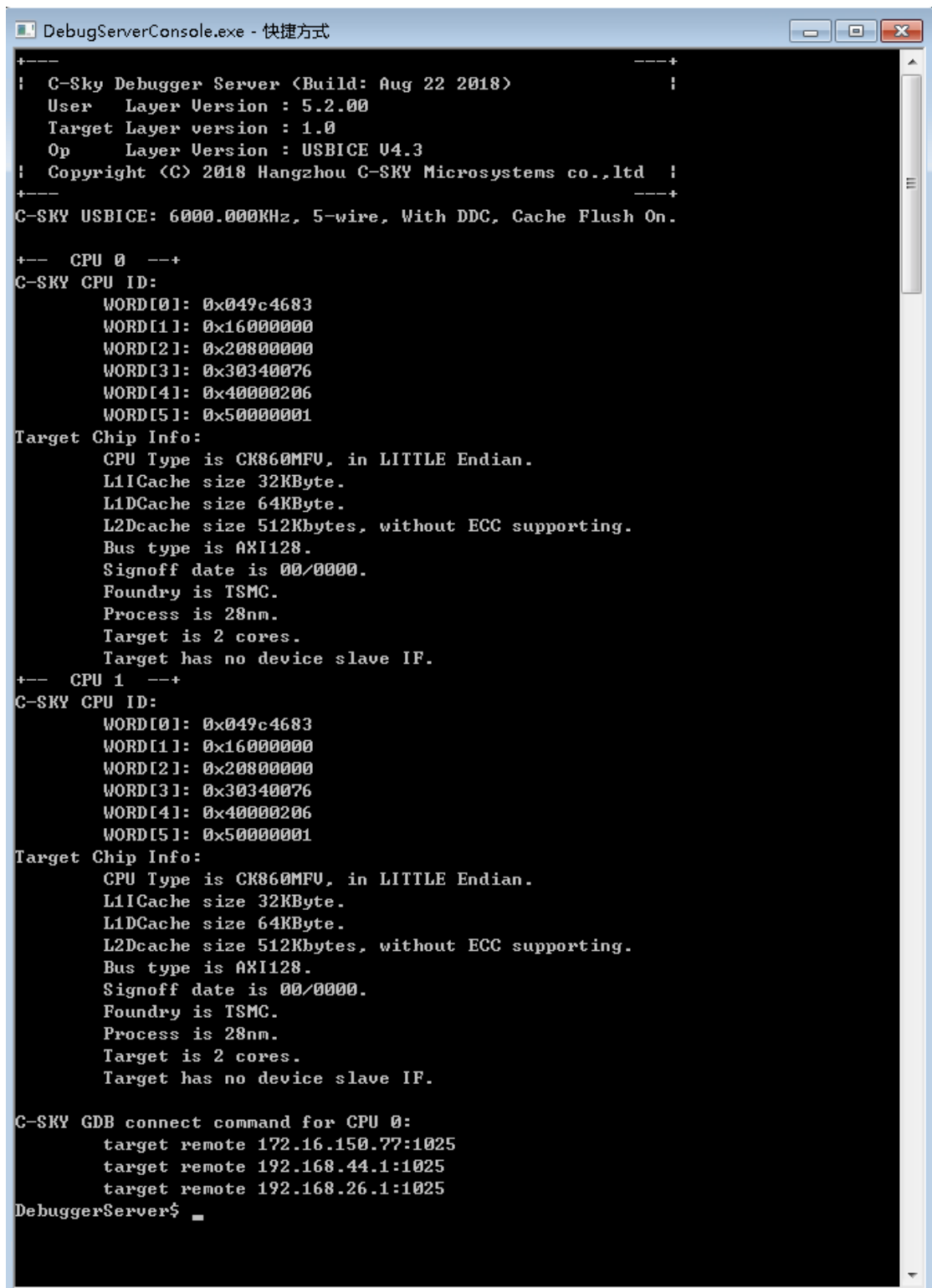
此时 DebugServer 将正常打印 CPU 0 和 CPU 1 的 CPUID 信息。

(2) Console 版 DebugServer 操作（linux 版本同）：

- ① 使用 Ctrl+c 结束 DebugServerConsole
- ② 重新运行 DebugServerConsole

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



```

DebugServerConsole.exe - 快捷方式
+----+
! C-Sky Debugger Server <Build: Aug 22 2018>
! User Layer Version : 5.2.00
! Target Layer version : 1.0
! Op Layer Version : USBICE V4.3
! Copyright <C> 2018 Hangzhou C-SKY Microsystems co.,ltd !
+----+
C-SKY USBICE: 6000.000KHz, 5-wire, With DDC, Cache Flush On.

+--- CPU 0 ---+
C-SKY CPU ID:
WORD[0]: 0x049c4683
WORD[1]: 0x16000000
WORD[2]: 0x20800000
WORD[3]: 0x30340076
WORD[4]: 0x40000206
WORD[5]: 0x50000001
Target Chip Info:
CPU Type is CK860MF0, in LITTLE Endian.
L1ICache size 32KByte.
L1DCache size 64KByte.
L2Dcache size 512Kbytes, without ECC supporting.
Bus type is AXI128.
Signoff date is 00/0000.
Foundry is TSMC.
Process is 28nm.
Target is 2 cores.
Target has no device slave IF.

+--- CPU 1 ---+
C-SKY CPU ID:
WORD[0]: 0x049c4683
WORD[1]: 0x16000000
WORD[2]: 0x20800000
WORD[3]: 0x30340076
WORD[4]: 0x40000206
WORD[5]: 0x50000001
Target Chip Info:
CPU Type is CK860MF0, in LITTLE Endian.
L1ICache size 32KByte.
L1DCache size 64KByte.
L2Dcache size 512Kbytes, without ECC supporting.
Bus type is AXI128.
Signoff date is 00/0000.
Foundry is TSMC.
Process is 28nm.
Target is 2 cores.
Target has no device slave IF.

C-SKY GDB connect command for CPU 0:
target remote 172.16.150.77:1025
target remote 192.168.44.1:1025
target remote 192.168.26.1:1025
DebuggerServer$

```

图 8-8 Console 版本 DebugServer 已多核单端口连接 CK860MP 显示（linux 同）

此时 DebugServer 将正常打印 CPU 0 和 CPU 1 的 CPUID 信息。

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

## 5. 启动 GDB，调试多核：

- (1) 启动 csky-\*abiv2\*-gdb
- (2) 在 gdb 的命令行输入“target remote ip:port”(DebugServer 界面显示的连接命令)
- (3) 在 gdb 的命令行输入：info threads，显示

```
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=csky-elfabiv2".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(cskygdb) target remote 172.16.150.77:1025
Remote debugging using 172.16.150.77:1025
warning: No executable has been specified and target does not support
determining executable automatically. Try using the "file" command.
0x1fbd819c in ?? ()
(cskygdb) info threads
  Id   Target Id         Frame
* 1    Thread 1 (CPU#0)  0x1fbd819c in ?? ()
  2    Thread 2 (CPU#1)  0x00000000 in ?? ()
(cskygdb) █
```

图 8-9 GDB info thread 查看线程

如上图所示，GDB 内部显示了两条线程 Thread 1 和 Thread 2，分别对应 CPU 0 和 CPU 1。

## 8.3.2. 线程操作

在【8.3.1】操作的基础上，GDB 端 Thread 将与 CPU 一一对应，那么如果用户需要：

### 1. 查看 CPU 1 的寄存器信息（以下为 gdb 命令行输入信息）

- (1) thread 2 (将 gdb 内部的线程切换至 Thread 2)
- (2) info registers (此时 gdb 将显示 CPU 1 的 GPR，PC，PSR 等寄存器信息)

```

Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=csky-elfabi2".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(cskygdb) target remote 172.16.150.77:1025
Remote debugging using 172.16.150.77:1025
warning: No executable has been specified and target does not support
determining executable automatically. Try using the "file" command.
0x1fbd819c in ?? ()
(cskygdb) info threads
   Id   Target Id         Frame
* 1     Thread 1 (CPU#0)  0x1fbd819c in ?? ()
   2     Thread 2 (CPU#1)  0x00000000 in ?? ()
(cskygdb) thread 2
[Switching to thread 2 (Thread 2)]
#0  0x00000000 in ?? ()
(cskygdb) i r
r0          0xb60cd4 11930836
r1          0x0      0
r2          0x80c9f8f6 -2134247178
r3          0x0      0
r4          0x1      1
r5          0x1fff 8191
r6          0x965ba8 9853864
r7          0x2      2
r8          0x0      0
r9          0x9f060000 -1626996736
r10         0x9f060000 -1626996736
r11         0x0      0
r12         0x9f060000 -1626996736
r13         0xfffffe000 -8192
r14         0x1dcd6500 0x1dcd6500
r15         0x80ec3d30 -2132001488
r16         0xb60cd4 11930836
r17         0x0      0
r18         0x80c9f8f6 -2134247178
r19         0x0      0
r20         0x1      1
r21         0x1fff 8191
r22         0x965ba8 9853864
r23         0x2      2
r24         0x0      0
r25         0x9f060000 -1626996736
r26         0x9f060000 -1626996736
r27         0x0      0
r28         0x9f060000 -1626996736
r29         0xfffffe000 -8192
r30         0x1dcd6500 500000000
r31         0x80ec3d30 -2132001488
pc          0x0      0x0
epc         0x0      0x0
psr         0x80000000 -2147483648
epsr        0x0      0
(cskygdb) █

```

图 8-10 GDB 端切换线程查看 CPU 1 的寄存器信息

## 2. 查看 CPU 0 的寄存器信息

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

- (1) thread 1 (将 gdb 内部的线程切换至 Thread 1)
- (2) info registers (此时 gdb 将显示 CPU 1 的 GPR, PC, PSR 等寄存器信息)

```

(cskygdb) thread 1
[Switching to thread 1 (Thread 1)]
#0  0x1fbd819c in ?? ()
(cskygdb) i r
r0      0x33      51
r1      0xa       10
r2      0x20      32
r3      0x20      32
r4      0x16      22
r5      0x80e94000 -2132197376
r6      0x0        0
r7      0x0        0
r8      0x0        0
r9      0x0        0
r10     0x80e95f94 -2132189292
r11     0x0        0
r12     0x3        3
r13     0x1        1
r14     0x1fc1ffe4 0x1fc1ffe4
r15     0x1fbd819c 532513180
r16     0x33      51
r17     0xa       10
r18     0x20      32
r19     0x20      32
r20     0x16      22
r21     0x80e94000 -2132197376
r22     0x0        0
r23     0x0        0
r24     0x0        0
r25     0x0        0
r26     0x80e95f94 -2132189292
r27     0x0        0
r28     0x3        3
r29     0x1        1
r30     0x1fc1ffe4 532807652
r31     0x1fbd819c 532513180
pc      0x1fbd819c 0x1fbd819c
epc     0x0        0x0
psr     0x80000101 -2147483391
epsr    0x0        0
(cskygdb) █

```

图 8-11 GDB 端切换线程查看 CPU 0 的寄存器信息

### 3. 设置 CPU 0 的 PC 到 0x10000

- (1) 如果已经在 Thread 1, 则用再输入 Thread 1 来切换 (线程编号为 CPU 编号+1)。  
可以在 gdb 的命令行输入“thread”, 或“info threads” gdb 将显示当前线程:



```
(cskygdb) thread
[Current thread is 1 (process <main>)]
(cskygdb) info threads
  Id   Target Id         Frame
* 1    process <main> (CPU#0 [running]) 0x1fbd819c in ?? ()
  2    process <main> (CPU#1 [running]) 0x00000000 in ?? ()
(cskygdb) █
```

图 8-12 GDB 端查看当前线程

打印信息中，最前面带“\*”的即为当前线程

- (2) set \$pc = 0x10000 (设置 CPU 0 的 PC 为 0x10000)

```
(cskygdb) thread
[Current thread is 1 (process <main>)]
(cskygdb) info threads
  Id   Target Id         Frame
* 1    process <main> (CPU#0 [running]) 0x1fbd819c in ?? ()
  2    process <main> (CPU#1 [running]) 0x00000000 in ?? ()
(cskygdb) thread 1
[Switching to thread 1 (process <main>)]
#0 0x1fbd819c in ?? ()
(cskygdb) set $pc=0x10000 █
```

图 8-13 GDB 端切换线程并设置 CPU 0 的 PC 至 0x10000

4. 所有的查看和修改操作需要对指定 CPU 进行查看寄存器，设置寄存器，查看 backtrace，内存等均需要在查看前确认是否为对应线程。

5. 运行程序：由于该模式下调试信号相互响应，即用户在 gdb 的命令行输入 stepi, step, next, continue 等让程序运行的时候，该命令虽然是发给当前 CPU，但该退出调试模式的信号会被其他核响应。

- (1) Si 时，所有 CPU 均会执行 si

- (2) Step, continue 等命令时，当前线程对应的 CPU 退出调试模式，其他 CPU 均跟随退出调试模式。直到所有 CPU 中有一个 CPU 因为断点或其他原因进入调试模式后，其他的 CPU 将被第一个进入调试模式的 CPU 拉入调试模式。GDB 获取到 CPU 进入调试模式后，打印相关提示信息，并将当前线程切换至第一个进入调试模式的 CPU 对应的线程。

## 6. 设置断点

- (1) 软件断点，该断点对所有 CPU 有效
- (2) 硬件断点，该断点对所有 CPU 有效
- (3) 如果只想某个 CPU 停止遇到断点停止时，可在设置断点时加入 Thread 信息，比如：

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



“break \*0x10000 thread 2”

即当 Thread 2（CPU 1）遇到该断点时，GDB 停止。

## 8.4. 多核多端口模式

在该模式下，DebugServer 将为每个 CPU 开启一个服务端口供 CSKY-GDB 连接，以 port 作为区分。CSKY-GDB 通过“target remote ip:port”方式连接上 DebugServer 后，命令中 port 指定后即指定连接该 port 对应的 CPU。对应关系在 DebugServer 的界面上可以看到。

该模式下，多个 CPU 将不互相发送并不响应其他核的调试信号，意为被调试的核独立，其中一个 CPU 进入或退出调试模式，都不会影响其他核的运行。可以认为用户在调试多个开发板，只是多个开发板的内存是一份。

模型如下：

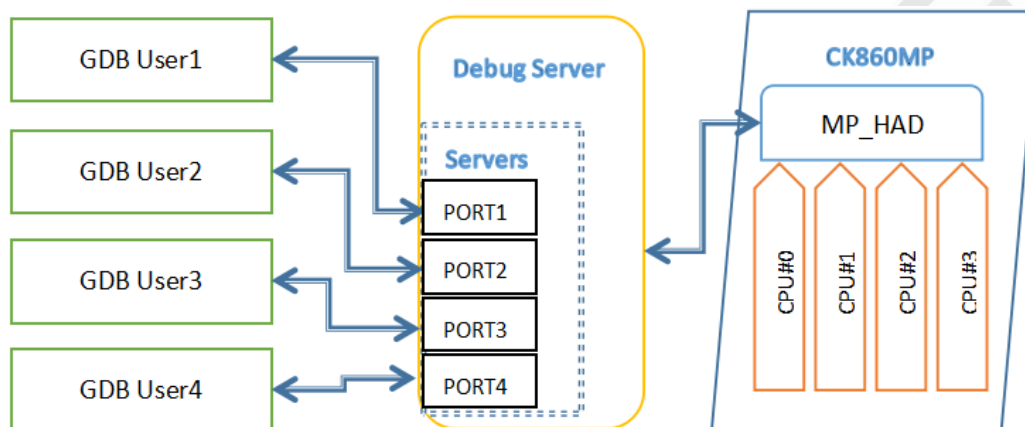


图 8-14 多核多端口模型图

### 8.4.1. 操作步骤

此处以 2 个核的 CK860MP 为例：

1. 与【操作步骤】的 1 操作一致
2. 与【操作步骤】的 2 操作一致
3. 与【操作步骤】的 3 操作一致

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

#### 4. 重新启动 DebugServer，识别多核：

##### (1) UI 版本 DebugServer

- ① 可选择先启动 DebugServer，运行起来后暂停 server。点击菜单中的“Setting->Multicore Threads”，当该选择如下图时，再次启动 Server：

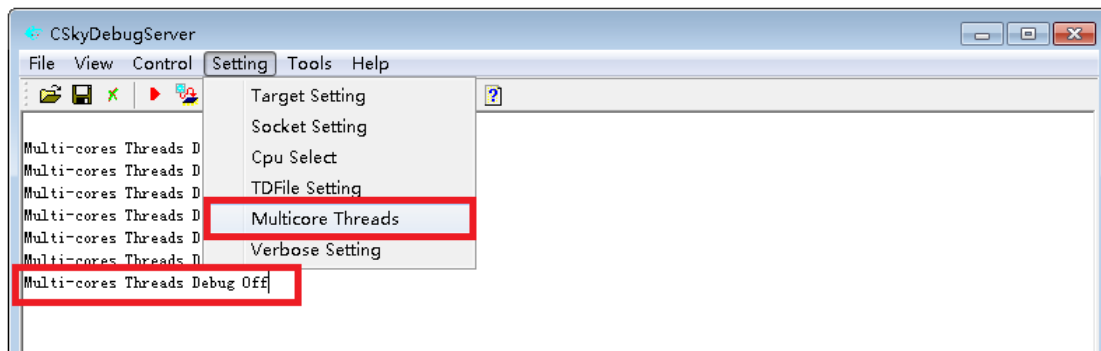


图 8-15 UI 界面设置 -no-multicore-threads 模式

- ② 也可选择修改 DebugServer 程序所在目录的 default.ini 中的“MULTICORETHREADS=TRUE”为“MULTICORETHREADS=FALSE”，然后启动 DebugServer

名称	修改日期	类型	大小
links	2020/3/2 14:44	文件夹	
t descriptions	2020/3/2 14:44	文件夹	
CSkyDebugServer.exe	2020/3/2 11:19	应用程序	3,606 KB
DebugServerConsole.exe	2020/3/2 11:19	应用程序	296 KB
default.ini	2020/3/2 11:19	配置设置	2 KB
libusb-1.0.dll	2020/3/2 11:19	应用程序扩展	101 KB
libusb-1.0.pdb	2020/3/2 11:19	Intermediate file	931 KB
Modules.dll	2020/3/2 11:19	应用程序扩展	381 KB
Target.dll	2020/3/2 11:19	应用程序扩展	3,588 KB
Utils.dll	2020/3/2 11:19	应用程序扩展	289 KB
XmlParser.dll	2020/3/2 11:19	应用程序扩展	663 KB

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

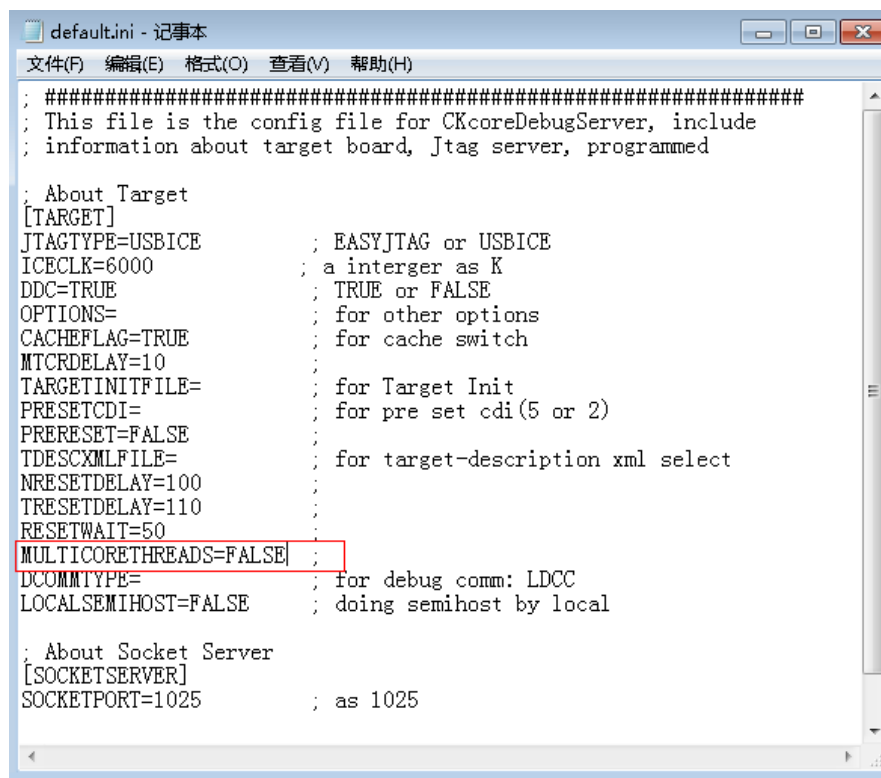


图 8-16 UI 通过 default.ini 设置 -no-multicore-threads 模式

③ 启动 DebugServer 后如下图:

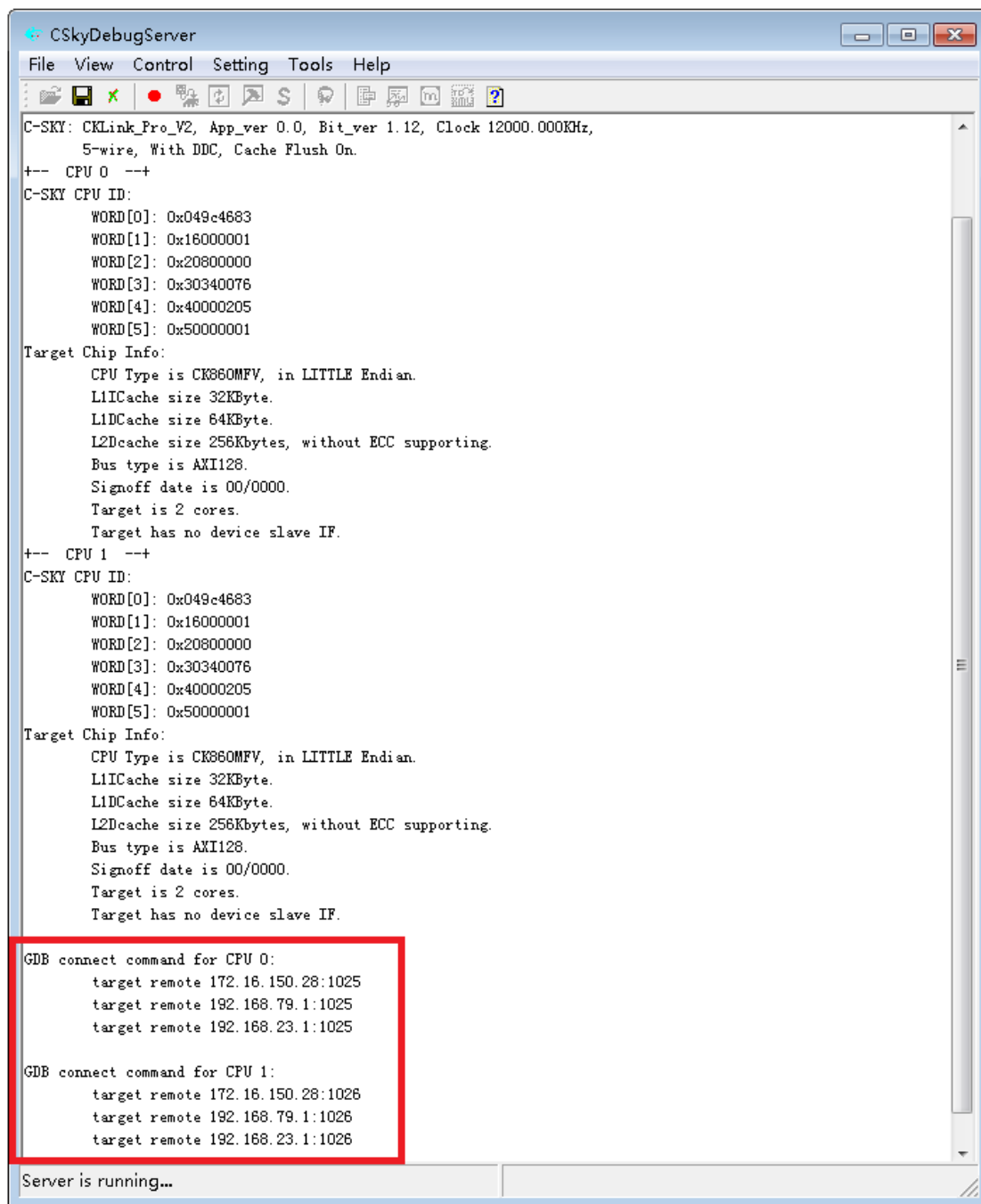


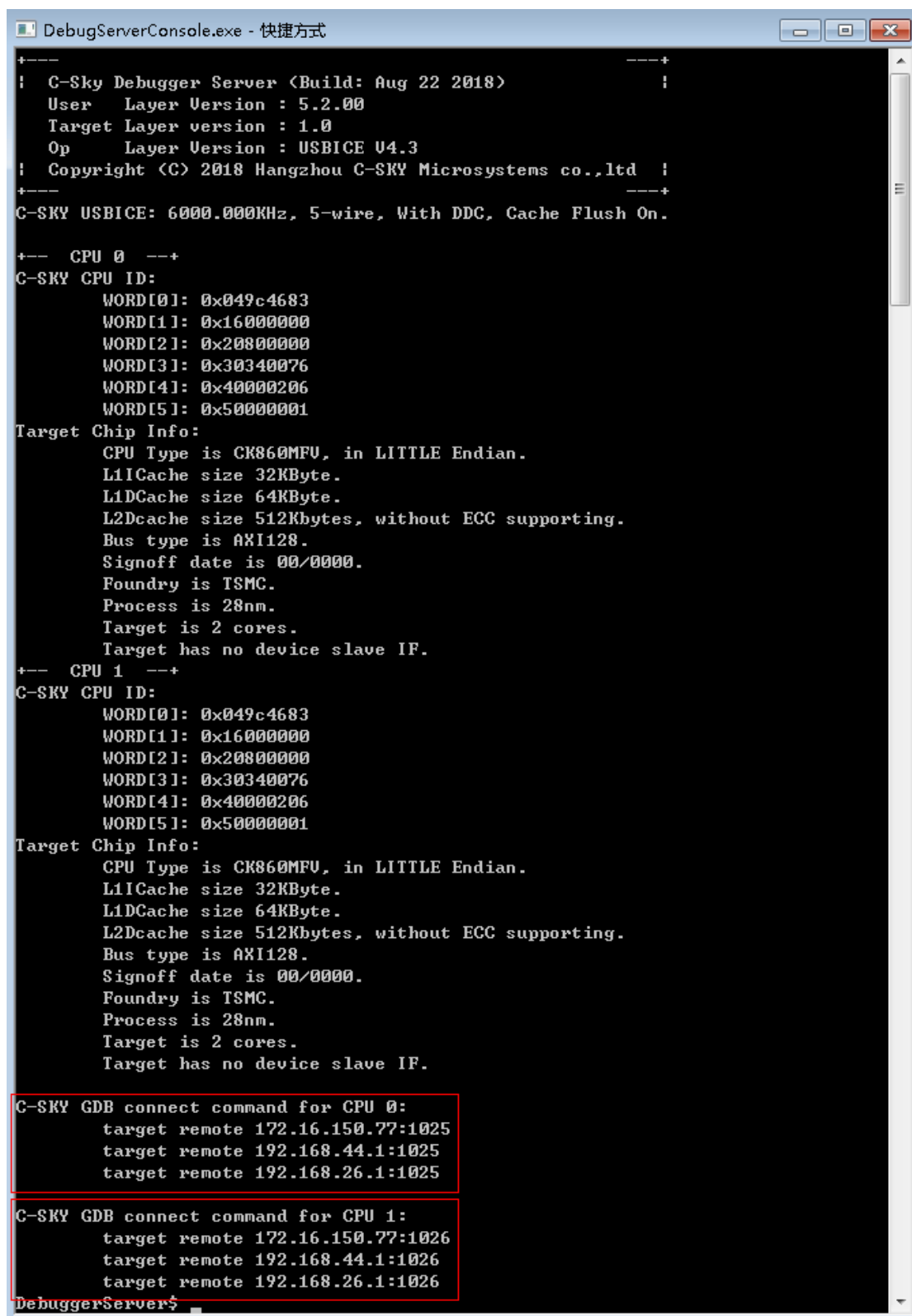
图 8-17 UI 版本 DebugServer 已多核多端口连接 CK860MP 显示

(2) Console 版本 DebugServer (linux 版本同) :

启动 DebugServerConsole 加启动参数: `-no-multicore-threads`

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



```

DebugServerConsole.exe - 快捷方式
+---+
! C-Sky Debugger Server <Build: Aug 22 2018> !
User Layer Version : 5.2.00
Target Layer version : 1.0
Op Layer Version : USBICE V4.3
! Copyright (C) 2018 Hangzhou C-SKY Microsystems co.,ltd !
+---+
C-SKY USBICE: 6000.000KHz, 5-wire, With DDC, Cache Flush On.

+-- CPU 0 --+
C-SKY CPU ID:
WORD[0]: 0x049c4683
WORD[1]: 0x16000000
WORD[2]: 0x20800000
WORD[3]: 0x30340076
WORD[4]: 0x40000206
WORD[5]: 0x50000001
Target Chip Info:
CPU Type is CK860MF0, in LITTLE Endian.
L1ICache size 32KByte.
L1DCache size 64KByte.
L2Dcache size 512Kbytes, without ECC supporting.
Bus type is AXI128.
Signoff date is 00/0000.
Foundry is TSMC.
Process is 28nm.
Target is 2 cores.
Target has no device slave IF.

+-- CPU 1 --+
C-SKY CPU ID:
WORD[0]: 0x049c4683
WORD[1]: 0x16000000
WORD[2]: 0x20800000
WORD[3]: 0x30340076
WORD[4]: 0x40000206
WORD[5]: 0x50000001
Target Chip Info:
CPU Type is CK860MF0, in LITTLE Endian.
L1ICache size 32KByte.
L1DCache size 64KByte.
L2Dcache size 512Kbytes, without ECC supporting.
Bus type is AXI128.
Signoff date is 00/0000.
Foundry is TSMC.
Process is 28nm.
Target is 2 cores.
Target has no device slave IF.

C-SKY GDB connect command for CPU 0:
target remote 172.16.150.77:1025
target remote 192.168.44.1:1025
target remote 192.168.26.1:1025

C-SKY GDB connect command for CPU 1:
target remote 172.16.150.77:1026
target remote 192.168.44.1:1026
target remote 192.168.26.1:1026
DebuggerServer$

```

图 8-18 Console 版本 DebugServer 已多核多单端口连接 CK860MP 显示 (linux 同)

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

## 5. GDB 连接，调试

- (1) 此时，DebugServer 界面上已经显示了连接 CPU 0 和 CPU 1 的连接命令，IP 相同，Port 做区分。
- (2) 此时 GDB 调试可认为在调试单核，与单核调试无异（调试者需要知道此时各个核是共享内存的即可）。

# 9. Vendor ICE 支持

在业务的发展过程中，我们的客户具备了开发设计 Link 的能力。在 DebugServer 应对客户自己设计开发的 Link。因此 DebugServer 增加了客户设计开发的 Link 功能。

该功能的实现方式是：

安装目录下的 links 目录中存放着多个 Vendor 目录，每个目录中保存的客户实现的 Link 库文件。库文件需要实现一些接口用于 porting DebugServer 完成调试功能。

porting 接口见表格 9-1 Link Porting 接口列表，详细也可见示例程序的 Include/link.h

接口	接口原型	描述
link_init	Int link_init (dbg_server_cfg_t *cfg);	初始化 link option
link_open	void*link_open(dbg_server_cfg_t *cfg, void *unique);	打开 link 设备
link_close	void link_close (void *handle);	关闭 link 设备
link_config	int link_config (void *handle, enum LINK_CONFIG_KEY key, unsigned int value);	配置 link, 包括频率、复位时间等 具体配置项可 option
link_upgrade	int link_upgrade (void *handle, const char *path);	固件升级 option
link_memory_read	int link_memory_read (void *handle, uint64_t addr, int xlen, uint8_t *buff, int length, int	读 target 内存

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

	mode);	
link_memory_write	int link_memory_write (void *handle, uint64_t addr, int xlen, uint8_t *buff, int length, int mode);	写 target 内存
link_register_read	int link_register_read (void *handle, int regno, uint8_t *buff, int nbyte);	读 target 通用寄存器
link_register_write	int link_register_write (void *, int regno, uint8_t *buff, int nbyte);	写 target 通用寄存器
link_jtag_operator	int link_jtag_operator (void *handle, int ir_len, unsigned char *ir, int dr_len, unsigned char *dr_r, unsigned char *dr_w, int read);	执行一次 JTAG 操作 option
link_gpio_operator	int link_gpio_operator (void *, int gpio_out, int *gpio_in, int gpio_eo, int gpio_mode);	执行一次 GPIO 操作 option
link_show_info	int link_show_info (void *, dbg_server_cfg_t *cfg, void (*func)(const char *, ...));	显示 link 信息。 option
link_reset	int link_reset (void *handle, int hard);	复位 link
link_get_device_list	int link_get_device_list (struct link_dev *dev, int *count);	获取 link 列表
THE_NAME_OF_LINK	const char * THE_NAME_OF_LINK (void);	LINK 名字, 该接口作为 DLL 识别接口, 必须实现。

表格 9-1 Link Porting 接口列表

## C-Sky Confidential

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).

## 10. Example 工程

针对不同的应用场景，客户某些时候需要自己实现一些操作去控制或者读写芯片信息。因此我们提供一个示例工程，告知客户如何通过 **target** 接口去操作芯片。

在 C-SKY DebugServer 工具的安装目录下存在一个 **Example** 目录，在 **linux** 和 **windows** 下分别有不同的使用方式。

**linux** 下，我们提供提供 **Makefile** 编译工程。依赖工具：**make, gcc, g++**。

**windows** 下，我们提供 **VS** 工程。依赖工具：**Virtual Studio**。

打开工程，源文件中我们可以看到一些操作。包含：

1. 初始化过程及连接目标板
2. 寄存器读写操作
3. 内存读写操作
4. 运行/停止等
5. 断开连接

详细信息请打开工程查看。

## 11. 常见问题及解决方法

表格 2 常见问题及解决方法

问题类型	错误提示	解决方法
ICE 连接失败	<pre>ERROR: No C-SKY ICE connected to Your PC or your C-SKY ICE driver not installed correctly! Input enter to exit...</pre>	确保 ICE 重新连接并且 ICE 驱动成功安装
Target 连接失败	<pre>ERROR: Fail to enter debug mode! Error: Can't enter debug mode, please check the target board physical link. Input enter to exit...</pre>	ICE 连接好目标板，并且保证目标板上电，重新连接
端口绑定失败	<pre>ERROR: Fail to bind socke port 1025, please change another one. ERROR: Fail to create socket server. Input enter to exit...</pre>	打开端口设置菜单选项卡，重新选择端口号，再次连接

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).



版本升级问题	<pre> WARNING: ICE Upgrading ignored, it may cause function lose or error!!! ERROR: DebugServer can't implement ICE config! Input enter to exit... </pre>	重新打开 ckserver 连接 ICE，出现 ICE 升级提示时，选择 'y'，升级成功后，重新插拔 ICE，再次使用 ckserver 进行连接即可。
ICE 与 ckserver 版本兼容问题	<pre> ERROR: You ICE's version is newer than Your DebugServer, Please update Your DebugServer! ERROR: DebugServer can't implement ICE config! Input enter to exit... </pre>	此问题针对 cklink_lite 型号 ICE，ckserver 版本过老会出现此问题，应该从 C-SKY 官网下载更新最新版本的 ckserver
调试架构连接问题	连接 CSKY 发布的 CPU 连接时可直接连接，如果是连接的开发板调试架构是遵循 RISC-V DM 文档，则需要指定 -arch	console 版本添加启动参数 -arch riscv；UI 版本在 Setting->Target Setting 中 Debug Arch Select 中选择 RISC-V DM

**C-Sky Confidential**

The information contained herein is confidential and proprietary and is not to be disclosed outside of Hangzhou C-Sky Microsystems except under a Non-Disclosure Agreement (NDA).