神威平台
维护使用说明

2015 年 3 月

成都申威科技有限责任公司

成都市华府大道四段电子科大科技园 D22 栋
Building D22, National University Science and technology park,
Section 4, Huafu Avenue, Chengdu

免责声明

本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

成都申威科技有限责任公司

Chengdu Sunway Technology Corporation Limited

地址：成都市华府大道四段电子科大科技园 D22 栋

Building D22, National University Science and technology park,

Section 4, Huafu Avenue, Chengdu

Mail：sales@swcpu.cn

Tel ：028-68769016

Fax：028-68769019

阅读指南

《神威平台维护使用说明》主要描述了神威维护平台的串口命令说明、调试工具介绍、平台配置 CPU 频率说明等内容。

文档修订

文档更新记录	文档名	神威平台维护使用说明
	版本号	V2.2
	创建人	研发部
	创建日期	2013-10-11

版本更新

版本号	更新内容	更新日期
V1.0	形成说明手册 2013 版	2013-10-11
V1.1	增加响应包错误处理功能（错误码：见附录 A）	2013-12-24
V1.2	增加操作命令 2.3.12~2.3.37	2013-12-27
V1.3	增加支持 SW4A 寄存器读写的 rio 和 wio，维护接口复位命令 mtrst4a	2014-02-22
V2.0	增加状态检测和扫描、写扫描链和 readme 命令，生成版本 V2.0	2014-04-01
V2.1	增加 boardid 和 readme 命令说明	201-07-21
V2.2	增加 cpufreq4a 命令说明	2015-02-05
V2.2	增加 rmpc 命令说明	2015-03-02

技术支持

可通过邮箱或问题反馈网站向我司提交产品使用的问题，并获取技术支持。

售后服务邮箱：sales@swcpu.cn

问题反馈网址：http://www.swcpu.cn/

目　　录

1 串口命令手册

神威平台（以 SW1610/SW410/SW1600 处理器构成的计算平台）的主板上有 2 片 SPI 接口的 4MB 大小的 flash，第一片 flash 只能通过串口对其进行编程，第二片 flash 一般通过维护网口对其进行编程。

串口命令主要用于烧写和修改 firmware，firmware 共有 3 个文件组成:bmc_file、srom_file、config_file,在第一片 FLASH 中分区放置。目前 FLASH 的容量为 4MB。具体放置区域如下图所示：

文件名称	地址（16 进制）	空间大小	实际文件大小
bmc_file	0~300000	3MB	1.5MB
srom_file	300000~308000	32KB	11KB
config_file	308000~309000	4KB	16B
Reserved	309000~400000	247sector	

其中 config_file 主要用于给维护网络一个初始的 IP 地址以及记录 srom 的长度，具体定义如下：

地址	内容	备注
0x0~0x3	初始 IP 地址	
0x4~0x8	初始 MAC 地址低 4 字节	
0x9~0xc	SROM 的字节数	

原来放在第一片 flash 中的 bios、hmcode、以及配置文件的一部分现在放到了第二片 flash 中，可以通过维护网络远程读取、烧写、修改第二片 flash 中的内容。

第 2 片 Flash 中的空间分配如下图所示：

基地址：0x1000000

起始地址	内容	大小
0	硬件参数信息区	0x80000(512KB)
0	主存信息	0x1000(4KB)
0x1000	CPU 信息	0x1000(4KB)
0x2000	主板型号信息	0x1000(4KB)
0x3000	显存信息	0x1000(4KB)
0x4000	保留	0x1000(4KB)
0x5000	套片信息	0x1000(4KB)
0x6000	套片网卡 MAC 地址	0x1000(4KB)
0x7000	BMC 硬件设计版本信息	0x1000(4KB)
0x8000	BMC 文件信息	0x1000(4KB)
0x9000	SROM 文件信息	0x1000(4KB)
0xa000	HMCODE 文件信息	0x1000(4KB)
0xb000	BIOS 文件信息	0x1000(4KB)
0xc000	套片配置文件	0x1000(4KB)
0xd000	保留	0x2000(4KB)

0xf000	config 信息	0x1000(4KB)
0x10000	BIOS 设置信息	0x1000(4KB)
0x11000	保留	0x1000(4KB)
0x12000	幻数及硬件参数版本信息	0x1000(4KB)
0x13000	保留	0xd000(52KB)
0x20000	Srom	0x8000(32KB)
0x28000	保留	0x58000(352KB)
0x80000	HMCODE 固件	0x80000（512KB）
0x100000	BIOS 固件	0x300000（3MB）

CFG info 定义：

0ffset	**内容**	
0x0~0x3	BIOS 字节数	
0x4~0x7	IP 地址	
0x8~0xb	MAC 地址低 4 字节	
0xc	自引导选择参数	aa-开机自引导；其他值都不自引导
0xd	IP 选择	55-使用第二片 flash 中的 ip
0xe	Reserved	
0xf	SROM 选择	aa-使用第二片 flash 中的 srom
0x10~0x11	初始化核心数	F-4 核；FFFF-16 核
0x12	CPU 核心频率	2F 核心频率配置
0x13	存控频率	2F 存储控制器时钟配置
0x14	核心互联频率	
0x15	CPU 的 TESTOUTSEL	默认为 1
0x16	选择温度配置方式	55-从第二片 flash 读取，否则使 用固件中配置
0x17	高温	
0x18	低温	

注：2 片 flash 中都有 IP 地址和 MAC 地址低 4 字节的定义，主要是由于第二片 flash 只能通过维护网络进行编程，在此之前要保证维护网络可用，所以在第一片 flash 中给其分配一个初始的 IP 地址，当 config 文件已经烧入第二片 flash 中时，维护网络的 IP 地址就以第二片中的值为准。要使更改的 IP 地址生效，必须按下板上的 reset 按钮，对维护进行复位；或者拔插一次电源。

命令使用要求：设置环境变量，指向 command 目录。

1.1 向 FLASH 烧入 BMC 文件

命令格式：

wflash-all-2f_new com_number bmc_file srom_file config_file ip_addr

参数说明：

- com_number：串口号，标识烧固件时所使用的串口。
- bmc_file：BMC 二进制文件。
- srom_file：srom 二进制文件。
- config_file：配置二进制文件，具体配置见附录。
- ip_addr：初始 IP 地址。

例：
wflash-all-2f_new com1 atx0526_2.bin srom 2f_cfg.bin 1292.168.1.16
说明：上例将所有 BMC 相关文件烧入 FLASH 中，并将初始 IP 值设为 192.168.1.16.

1.2 更改 FLASH 中的 BMC 二进制文件

命令格式：

Cbmcfile_2f com_number bmc_file

参数说明：

- com_number：串口号，标识烧固件时所使用的串口。
- bmc_file：BMC 二进制文件。

1.3 更改 FLASH 中的 srom 二进制文件

命令格式：

Csrom_2f com_number srom_file

参数说明：

- com_number：串口号，标识烧固件时所使用的串口。
- srom_file：srom 二进制文件。

1.4 更改 FLASH 中的 config 二进制文件

命令格式：

Ccfgfile_2f com_number config_file

参数说明：

- com_number：串口号，标识烧固件时所使用的串口。
- config_file：config 二进制文件。

1.5 更改初始 IP 地址信息

命令格式：

Cipaddr_2f com_number ip_addr

参数说明：

- com_number：串口号，标识烧固件时所使用的串口。
- ip_addr：IP 地址。

2 调试工具

2.1 vi 维护环境安装

2.1.1 完全新环境

在执行维护命令时，必须在客户端进行如下操作：

1) 安装维护命令包。在 Linux 环境下，需要设置 PATH 和 CLASSPASS 环境变量。PATH 指向维护系统的 bin 目录；CLASSPATH 指向维护系统的 classes 目录）
2) 安装 JAVA 虚拟机。下载 Linux 系统下对应的 JDK，要求版本在 1.6 以上，下载完成后，进行安装。
   - ◆ 拷贝1.6的安装包：
   - ◆ 安装：./jdk-6u26-linux-i586.bin
   - ◆ 设置环境变量：$PATH查看当前PATH的设置，vi /root/.bash_profile,export PATH=/usr/java/jdk1.6.0_11/bin:$PATH
   - ◆ 退出，重新登录，查看java版本java -version，看是否已经变为1.6，如果还是没有，依次查看一下文件：.bash_profile .bashrc /etc/profile

■ 说明：登录 linux 时，一般先启动 etc/profile,再启动 bash_profile,bashrc，etc/profile 为系统每个用户配置信息，bash_profile 每个用户自己的配置文件

3) 修改 hosts 文件。将安装包中的 hosts 文件，拷贝到/etc 目录下。
   注：目前支持两种模式，如访问目标机 192.168.1.40，host 设置
   192.168.1.40　　bmc40
   192.168.1.40　　bmc00040
   -o 40:0:0:0 和-o 0:0:40:0 都可以访问到目标机(说明：若访问 -o 0:0:0:0 则只能访问 bmc0)
4) 安装 CPU 监测工具。将安装包中的 SW2F_Monitor.jar 文件，拷贝到/root 目录下。

2.2 维护命令格式说明

维护命令的基本格式为：

命令名 <参数…> [参数…] [-help]

说明：<>内的参数表示该命令的必需参数；[]内的参数表示该命令的可选参数。[-help]参数将打印出该命令简单的命令格式说明。

2.3 维护命令说明

2.3.1读存储器

rmem [-o bmcid:midpid:cardid:cupid[:coreid]] [-cache] [-bwdqA] <-a address> [-l length] [-f filename]

读 CPU 核心主存。

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-cache：可 cache 方式读，缺省为不可 cache 方式读

-bwdqA：以字节/字/双字/四倍字/ASCII 方式显示读出结果，缺省为双字方式。

-a address：指定主存起始地址

-l length：指定读主存的长度，缺省为 128 字节。

-f filename：将读出结果按照二进制数据格式保存到指定文件，不打印输出。

正确返回：

read mem result 0xXXX(地址)：

XX_XX_XX_XX_XX_XX_XX_XX_XX_XX_XX_XX_XX_XX_XX_XX(数据)

错误返回：

见附录 A

2.3.2写存储器

wmem [-o bmcid:midpid:cardid:cupid[:coreid]] <-a address> [-v value[,value[,value...]]] [-cache] [-bwdqA]

写数据到 CPU 核心主存。

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-a address：指定主存起始地址

-cache：可 cache 方式写，缺省为不可 cache 方式写。

-bwdqA：指定输入数据以字节/字/双字/四倍字/ASCII 方式为单位，缺省为双字方式。

-v value[,value[,value...]]：指定将写入的数据，缺省时接受命令行输入。

正确返回：

write success

错误返回：

见附录 A

2.3.3加载文件到存储器

loadfile [-o bmcid:midpid:cardid:cupid[:coreid]] <-a address> <-f filename> [-cache] [-c] [-w]

加载文件中数据到 CPU 核心主存。

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-a address：指定主存起始地址。

-f filename：包含加载数据的文件。

-cache：可 cache 方式写，缺省为不可 cache 方式写。

-c：将写入数据读出与源数据进行比较。

-w：指定数据比较不等时继续操作，缺省时比较不等时中断操作。

正确返回：

write file to memory XX bytes succeed!

错误返回：

见附录 A

2.3.4读 SW2F 的 IO 寄存器

rio2f [-o bmcid:midpid:cardid:cpuid[:coreid]] [-t regType] <-a regAddress|-n regName> [-m mask] [-s serialNo] [-i index] [-p]

读 CPU 的 IO 寄存器。

参数说明：

-o bmcid:midpid:cardid:cpuid[:coreid] ：指定 BMC 号：中板号：计算刀片号：CPU 号[：核心号]。对于读 CG 相关 IO 寄存器，可不指定寄存器类型，直接通过目标号中的核心号来指定。

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改.

-t regType:

cg0: registers of CG0
cg1: registers of CG1
cg2: registers of CG2
cg3: registers of CG3
cpm: registers of CPM
pub0: registers of PUB0
pub1: registers of PUB1
pcie0: registers of PCI-E0
pcie1: registers of PCI-E1
mi: registers of CPU's maintenance interface
si: registers of CPU's system interface

-a regaddress：指定寄存器地址。

-n regname：指定寄存器名称。当通过寄存器名称来读 CG 相关寄存器时，可通过在目标中指定核心号来得到 CG 号。

-m mask：寄存器数据有效指示位，0xf 表示寄存器低 32 位有效，0xf0 表示寄存器高 32 位有效，0xff 表示寄存器 64 位数据有效，缺省为 0xff。本参数仅对 PCI-E 相关 IO 寄存器有效。

-p ： 打印所有的 IO 寄存器名称。

正确返回：

0xXXXX,XXXX,XXXX,XXXX （注：64 位数据）

错误返回：

见附录 A

2.3.5写 SW2F 的 IO 寄存器

wio2f [-o bmcid:midpid:cardid:cpuid[:coreid]] [-t regType] <-a regAddress|-n regName> <-v value> [-m mask] [-s serialNo] [-i index] [-or|-xor|-and] [-p]

写数据到 CPU 的 IO 寄存器。

参数说明：

-o bmcid:midpid:cardid:cpuid[:coreid] ：指定 BMC 号：中板号：计算刀片号：CPU 号[：核心号]。对于写 CG 相关 IO 寄存器，可不指定寄存器类型，直接通过目标号中的核心号来指定。

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改.

-t regType：寄存器分类

cg0: registers of CG0
cg1: registers of CG1
cg2: registers of CG2
cg3: registers of CG3
cpm: registers of CPM
pub0: registers of PUB0
pub1: registers of PUB1
pcie0: registers of PCI-E0
pcie1: registers of PCI-E1
mi: registers of CPU's maintenance interface
si: registers of CPU's system interface

-a regaddress：指定寄存器地址。

-n regname：指定寄存器名称。当通过寄存器名称来读 CG 相关寄存器时，可通过在目标中指定核心号来得到 CG 号。

-v value：将写入的数据，64 位或低 32 位有效，仅 PCIE 寄存器可能为 32 位。

-m mask：寄存器数据有效指示位，0xf 表示寄存器低 32 位有效，0xf0 表示寄存器高 32 位有效，0xff 表示寄存器 64 位数据有效，缺省为 0xff。本参数仅对 PCI-E 相关 IO 寄存器有效。

-or|-xor|-and：分别表示或写、异或写和与写，缺省为覆盖写。

-p ： 打印所有的 IO 寄存器名称。

正确返回：

write success

错误返回：

见附录 A

2.3.6读 SW4A 的 IO 寄存器

rio [-o bmcid:midpid:cardid:cpuid[:coreid]] [-t regType] <-a regAddress|-n regName> [-m mask] [-s serialNo] [-i index] [-p]

读 CPU 的 IO 寄存器。

参数说明：

-o bmcid:midpid:cardid:cpuid[:coreid] ：指定 BMC 号：中板号：计算刀片号：CPU 号[：核心号]。对于读 CG 相关 IO 寄存器，可不指定寄存器类型，直接通过目标号中的核心号来指定。

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改.

-t regType:

cg0: registers of CG0
cpm: registers of CPM
mcbox0: registers of MCBOX0
mcbox1: registers of MCBOX1
pub0: registers of PUB0
pub1: registers of PUB1
pcie0: registers of PCI-E0
pcie1: registers of PCI-E1
mi: registers of CPU's maintenance interface
si: registers of CPU's system interface

-a regaddress：指定寄存器地址。

-n regname：指定寄存器名称。当通过寄存器名称来读 CG 相关寄存器时，可通过在目标中指定核心号来得到 CG 号。

-m mask：寄存器数据有效指示位，0xf 表示寄存器低 32 位有效，0xf0 表示寄存器高 32 位有效，0xff 表示寄存器 64 位数据有效，缺省为 0xff。本参数仅对 PCI-E 相关 IO 寄存器有效。

-p ：打印所有的 IO 寄存器名称。

正确返回：

0xXXXX,XXXX,XXXX,XXXX （注：64 位数据）

错误返回：

见附录 A

例如：查看 CPUID

rio –o bmcnum:0:0:0 –t mi –n cpuid

2.3.7写 SW4A 的 IO 寄存器

wio [-o bmcid:midpid:cardid:cpuid[:coreid]] [-t regType] <-a regAddress|-n regName> <-v value> [-m mask] [-s serialNo] [-i index] [-or|-xor|-and] [-p]

写数据到 CPU 的 IO 寄存器。

参数说明：

-o bmcid:midpid:cardid:cpuid[:coreid] ：指定 BMC 号：中板号：计算刀片号：CPU 号[：核心号]。对于写 CG 相关 IO 寄存器，可不指定寄存器类型，直接通过目

标号中的核心号来指定。

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改.

-t regType：寄存器分类

cg0: registers of CG0
cpm: registers of CPM
mcbox0: registers of MCBOX0
mcbox1: registers of MCBOX1
pub0: registers of PUB0
pub1: registers of PUB1
pcie0: registers of PCI-E0
pcie1: registers of PCI-E1
mi: registers of CPU's maintenance interface
si: registers of CPU's system interface

-a regaddress：指定寄存器地址。

-n regname：指定寄存器名称。当通过寄存器名称来读 CG 相关寄存器时，可通过在目标中指定核心号来得到 CG 号。

-v value：将写入的数据，64 位或低 32 位有效，仅 PCIE 寄存器可能为 32 位。

-m mask：寄存器数据有效指示位，0xf 表示寄存器低 32 位有效，0xf0 表示寄存器高 32 位有效，0xff 表示寄存器 64 位数据有效，缺省为 0xff。本参数仅对 PCI-E 相关 IO 寄存器有效。

-or|-xor|-and：分别表示或写、异或写和与写，缺省为覆盖写。

-p ：打印所有的 IO 寄存器名称。

正确返回：

write success

错误返回：

见附录 A

2.3.8CPU 硬复位

rstcpu <-o bmcid:midpid:cardid:cpuid> [-core corestart]

复位开发板上的 CPU。

参数说明：

-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

-core corestart：核心断连参数。一个需要复位的 CPU 对应一个参数，0~15 位对应核心 0~15，‘1’表示连接。

正确返回：

RstCpu Success

2.3.9CPU 扫错（SW2F）

scaniferr <-o bmcid:midpid:cardid:cpuid>

查看 CPU 的 cgx_fault_state，看是否有报错

参数说明：

-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

正确返回：

BMC (11:0:0:0 , 0x000f) status : OK![Freq : 11: core/memory/xbx frequency : 1200MHz / 266.67MHz / 1000MHz]

说明：扫错需要结合监测界面一起看，ROB 超时等信息是在监测界面上体现

2.3.10 读 PC 值（SW2F）

rpc <-o bmcid:midpid:cardid:cpuid>
读各个核的 PC 值。

2.3.11 维护复位(SW2F)

mtrst <-o bmcid:midpid:cardid:cpuid>
维护复位。
正确返回：
`Maintenance Reset succeed!`
错误返回：
见附录 A

2.3.12 维护复位(SW4A)

mtrst4a <-o bmcid:midpid:cardid:cpuid>
维护复位。
正确返回：
`Maintenance Reset succeed!`
错误返回：
见附录 A

2.3.13 加载 SROM 到 icache 中

loadicache <-o bmcid:midpid:cardid:cpuid> <-f filename>
参数说明：
-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号
-f filename:需要加载到 flash 中的文件名

2.3.14 状态监测和扫描

scan_out <-o cabid:midpid[:cardid[:cpuid]> <-a startAddr> <-l length> <-s startBit> [-bw bitWidth] [-ra rightAlignBitWidth] [-bwdq] [-t]
-a startAddr：起始地址，缺省值为 0
-l length：读出的字节长度，缺省值为 128
-s startBit： 读出字节的有效起始位: [0~7]
-bw bitWidth:：读出字节的位宽: [1~8]. 缺省为 1

-ra： 右对齐位宽。.
-bwdq：打印模式:：字节/字/双字/四字 . 缺省为字.
-t：用于测试平台。

2.3.15 写扫描链

scanin [-o bmcid:midpid:cardid:cupid[:coreid]] <-a startAddr> <-f filename>|[-v data0[,data1[,data2...]]]
将数据扫描到 CPU 的扫描链中。
参数说明：
-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号
midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；
coreid：缺省表示访问 CG0
-a startAddr：指定扫描链起始地址
-f filename：二进制写入文件。
-v data0[,data1[,data2...]]：指定将写入的数据，缺省时接受命令行输入。
-bwdq：输入数据长度模式：字节/字/双字/四字 .
错误返回：
见附录 A

2.3.16 加载文件到第二片 flash

loadflash <-o bmcid:midpid:cardid:cpuid> <-a address> <-f filename>[-c]
参数说明：
-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号
-a address：flash 中的偏移地址
-f filename:需要加载到 flash 中的文件名
说明：该命令为硬件人员调试使用，如果要加载特定文件，请使用下列命令

2.3.17 读第二片 flash 中的内容

rflash <-o bmcid:midpid:cardid:cpuid> [-bwdqA] <-a address> [-l length] <-f filename>
参数说明：
-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号
-a address：flash 中的偏移地址，比如，bios 存放的起始地址为 0x100000；hmcode 存放的起始地址为 0x10000；配置信息存放的起始地址为 0xf000
-l length：读出数据的字节数
-f filename：读出的数据保存到文件中（可选）
- bwdqA：结果显示格式，b--字节 w--字（2 字节）d—双字（4 字节） A—ASCII 码（必须是大写）
例：

rflash -o 11:0:0:0 -a 0xf000 -l 0x20 –b
返回：
0x000000f000:　00 00 10 00 c0 a8 01 0b 0b 01 a8 c0 aa 00 06 00
0x000000f010:　00 0f 0d 06 09 01 00 00 00 00 00 00 00 00 00 00

2.3.18 加载 BIOS 到第二片 flash

loadbios　<-o bmcid:midpid:cardid:cpuid>　<-f filename>[-c]
参数说明：
-o bmcid:midpid:cardid:cpuid ： 指定 BMC 号：中板号：计算刀片号：CPU 号
-f filename:需要加载到 flash 中的文件名
-c 与源文件进行比较。
注：该命令自动将 BIOS 加载到第二片 flash 中定义的地址，并自动将 BIOS 的文件长度写入配置文件所对应的位置，并烧录到 flash 中。

2.3.19 加载 hmcode 到第二片 flash

loadhmcode　<-o bmcid:midpid:cardid:cpuid>　<-f filename>[-c]
参数说明：
-o bmcid:midpid:cardid:cpuid ： 指定 BMC 号：中板号：计算刀片号：CPU 号
-f filename:需要加载到 flash 中的文件名
-c 与源文件进行比较。

2.3.20 加载 SROM 到第二片 flash

loadsrom　<-o bmcid:midpid:cardid:cpuid>　<-f filename>[-c]
参数说明：
-o bmcid:midpid:cardid:cpuid ： 指定 BMC 号：中板号：计算刀片号：CPU 号
-f filename:需要加载到 flash 中的文件名
-c 与源文件进行比较。
注：该命令自动将 SROM 加载到第二片 flash 中定义的地址，并自动将 BIOS 的文件长度写入配置文件所对应的位置，并烧录到 flash 中。

2.3.21 读版本信息

rversion　<-o bmcid:midpid:cardid:cpuid>
参数说明：
-o bmcid:midpid:cardid:cpuid ： 指定 BMC 号：中板号：计算刀片号：CPU 号
正确返回：
BMC firmware Version ： XX.XX.XX.XX
Srom Information ： XX.XX.XX.XX
Hmcode Information：XX.XX.XX.XX
BIOS Information：XX.XX.XX.XX

各文件信息

错误返回：

见附录 A

2.3.22 读取/设置 CPUID

cpuid <-o bmcid:midpid:cardid:cpuid>

读取 CPU ID 的值。

cpuid <-o bmcid:midpid:cardid:cpuid> [-v cpuid]

参数说明：

-v：设置 CPU id 的值。

2.3.23 读取/设置 BOARDID

boardid <-o bmcid:midpid:cardid:cpuid>

读取主板 ID 的值。

boardid <-o bmcid:midpid:cardid:cpuid> [-v cpuid]

参数说明：

-v：设置 board id 的值。

2.3.24 读取/修改 CPU 的频率配置

cpufreq <-o bmcid:midpid:cardid:cpuid >

读取 CPU 的核心频率、存控频率以及核心互联频率

参数说明：

-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

正确返回：

例：BMC 15: core/memory/xbx frequency(set) : 1200MHz / 266.67MHz / 1000MHz

BMC 15: core/memory/xbx frequency(effect) : 1200MHz / 266.67MHz / 1000MHz

cpufreq <-o bmcid:midpid:cardid:cpuid> [-c core_freq] [-m mm_freq] [-x xbx_freq] [-p]

修改 CPU 的核心频率、存控频率以及核心互联频率

参数说明：

-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

-c core_freq：需要配的核心频率值

-m mm_freq：需要配的存控频率值

-x xbx_freq：需要配的核心互联频率值

-p 打印出频率和值的对应关系

注：每个值都可单独设置

2.3.25 读取/修改 CPU 的温度阈值

cputemper <-o bmcid:midpid:cardid:cpuid >

读取 CPU 的核心频率、存控频率以及核心互联频率

参数说明：

-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

正确返回：

例：cpu temperature: XX.XX

cpu lower temperature alarm trip:- XX

cpu upper temperature alarm trip: XX

cputemper <-o bmcid:midpid:cardid:cpuid> [-h high_temp] [-l low_temp]

修改 CPU 的核心频率、存控频率以及核心互联频率

参数说明：

-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

-h high_temp：需要配的高温值

-l low_temp：需要配的低温值

注：每个值都可单独设置

2.3.26 修改维护端口 IP 地址

cmtip <-o bmcid:midpid:cardid:cpuid> [-v ip]

参数说明：

-v: 维护网口的 ip 地址

例：cmtip -o 10:0:0:0 -v 192.168.1.11

说明：修改完后需要维护复位或断电重启后才能生效

2.3.27 修改启动的核 core_start

winitcore <-o bmcid:midpid:cardid:cpuid> [-v core_start]

参数说明：

-v core_start：每 bit 代表一个核，比如起 0 核组 4 个核，core_start=0xf

默认为 0xf

2.3.28 读 IIC 设备

riic <-o bmcid:midpid:cardid:cpuid> <-m master> <-b busid> <-a address> <-rega regAddr> <-l readlen> [-f filename]|[-ps]

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-m master：IIC 主控制器选择。

-b busid：IIC 通道号选择。

-a address：IIC 设备的 SLAVE 地址

-rega regaAddr：IIC 设备的寄存器地址。
-l readlen：需要读取的字节长度。
-f filename：将读出结果按照二进制数据格式保存到指定文件，不打印输出。
-ps：采用 GBK 码显示结果。
正确返回：
0x address ： XX　XX　XX(显示字节数=readlen)
或 Write data to file filename Succeed!
错误返回：
见附录 A

2.3.29 写 IIC 设备

wiic　<-o bmcid:midpid:cardid:cpuid> <-m master> <-b busid> <-a address> <-rega regAddr> <-s string>| <-v value1[,value2…]>|<-f filename> [-c]
参数说明：
-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号
midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；
coreid：缺省表示访问 CG0
-m master：IIC 主控制器选择。
-b busid：IIC 通道号选择。
-a address：IIC 设备的 SLAVE 地址
-rega regaAddr：IIC 设备的寄存器地址。
-f filename：二进制写入文件。
-c 与写入的文件做比较。
正确返回：
`Write data to IIC device Succeed!`
错误返回：
见附录 A

2.3.30 读 8111 系列网卡的 EEPROM 信息

r8111x　<-o bmcid:midpid:cardid:cpuid> <-a address> <-sta startAddress> <-l length> [-f filename]|[-ps]
参数说明：
-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号
midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；
coreid：缺省表示访问 CG0
-a address：8111 系列网卡访问地址（偏移为 50h）.
-sta startAddress：EEPROM 起始地址。
-l length：需要读取的字节长度。
-f filename：将读出结果按照二进制数据格式保存到指定文件，不打印输出。
-ps：采用 String 形式显示结果。

正确返回：

0x address ： XX　XX　XX(显示字节数=readlen)

或 Write data to file filename Succeed!

错误返回：

见附录 A

2.3.31 写 8111 系列网卡的 EEPROM 信息

w8111x <-o bmcid:midpid:cardid:cpuid> <-a address> <-sta startAddress> <-f filename>|<-s string>|<-v byte0[,byte1...]> [-c]

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-a address：8111 系列网卡访问地址（偏移为 50h）.

-sta startAddress：EEPROM 起始地址。

-f filename：二进制写入文件。

-s string：写入字符串。

-v byte0,byte1：写入字节序列。

-c 与写入的文件做比较。

正确返回：

```
Write data to EEPROM succeed .
```

错误返回：

见附录 A

2.3.32 自动写 8111 系列网卡的 EEPROM 信息

w8111x-auto <-o bmcid:midpid:cardid:cpuid> <-a address> <-f filename> [-macf mac_filename] [-c]

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-a address：8111 系列网卡访问地址（偏移为 50h）.

-f filename：二进制写入文件。

-macf mac_filename：存放 MAC 地址的二进制文件（低 4 字节）。

-c 与写入的文件做比较。

正确返回：

```
Write data to EEPROM succeed .
```

错误返回：

见附录 A

说明：

mac_filename 为可选项，如果有，则自动增加 mac_filename 中的 MAC 地址；
如果没有，则自动增加 filename 中的 MAC 地址。

2.3.33 读 Intel 系列网卡的 EEPROM 信息

rintel <-o bmcid:midpid:cardid:cpuid> <-a address> <-sta startAddress> <-l length> [-f filename]|[-ps]

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-a address：intel 系列网卡访问地址（偏移为 10h）.

-sta startAddress：EEPROM 起始地址。

-l length：需要读取的字节长度。

-f filename：将读出结果按照二进制数据格式保存到指定文件，不打印输出。

-ps：采用 String 形式显示结果。

正确返回：

0x startAddress ： XX XX XX(显示字节数=length)

或 Write data to file filename Succeed!

错误返回：

见附录 A

2.3.34 写 Intel 系列网卡的 EEPROM 信息

wintel <-o bmcid:midpid:cardid:cpuid> <-a address> <-sta startAddress> <-f filename>|<-s string>|<-v byte0[,byte1...]> [-c]

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-a address：Intel 系列网卡访问地址（偏移为 10h）.

-sta startAddress：EEPROM 起始地址。

-f filename：二进制写入文件。

-s string：写入字符串。

-v byte0,byte1：写入字节序列。

-c 与写入的文件做比较。

正确返回：

```
Write data to EEPROM succeed .
```

错误返回：

见附录 A

2.3.35 自动写 8257x 系列网卡的 EEPROM 信息

w8257x-auto <-o bmcid:midpid:cardid:cpuid> <-a address> <-f filename> [-macf mac_filename] [-c]

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-a address：8257x 系列网卡访问地址（偏移为 10h）.

-f filename：二进制写入文件。

-macf mac_filename：存放 MAC 地址的二进制文件（低 4 字节）。

-c 与写入的文件做比较。

正确返回：

```
Write data to EEPROM succeed .
```

错误返回：

见附录 A

说明：

mac_filename 为可选项，如果有，则自动增加 mac_filename 中的 MAC 地址；如果没有，则自动增加 filename 中的 MAC 地址。

2.3.36 自动写 i350 网卡的 EEPROM 信息

wi350-auto <-o bmcid:midpid:cardid:cpuid> <-a address> <-f filename> [-macf mac_filename] [-c]

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-a address：i350 网卡访问地址（偏移为 10h）.

-f filename：二进制写入文件。

-macf mac_filename：存放 MAC 地址的二进制文件（低 4 字节）。

-c 与写入的文件做比较。

正确返回：

```
Write data to EEPROM succeed .
```

错误返回：

见附录 A

说明：

mac_filename 为可选项，如果有，则自动增加 mac_filename 中的 MAC 地址；如果没有，则自动增加 filename 中的 MAC 地址。

2.3.37 读 BMC 内部寄存器

rport <-o bmcid:midpid:cardid:cpuid> <-t type> <-a address>

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-t type：访问类型选择：1，2 or 4 字节寄存器。

-a address：IO 端口地址。

正确返回：

0xXX_XX_XX_XX

错误返回：

见附录 A

2.3.38 写 BMC 内部寄存器

wport <-o bmcid:midpid:cardid:cpuid> <-t type> <-a address> <-v value>

参数说明：

-o bmcid:midpid:cardid:cpuid:coreid ：指定 BMC 号：中板号：计算刀片号：CPU 号：核心号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

coreid：缺省表示访问 CG0

-t type：访问类型选择：1，2 or 4 字节寄存器。

-a address：IO 端口地址。

-v value：写入寄存器的数据。

正确返回：

```
Write data to IO port success!
```

错误返回：

见附录 A

2.3.39 写 1 字节 BMC 内部寄存器

命令格式：

writeioport1 bmcnum addr data

参数说明：

bmcnum：BMC 号。

addr：寄存器的地址。

data：向寄存器写入的数据。

2.3.40 读 1 字节 BMC 内部寄存器

命令格式：

readioport1 bmcnum addr

参数说明：

bmcnum：BMC 号。

addr：寄存器的地址

2.3.41 写 2 字节 BMC 内部寄存器

命令格式：

writeioport2 bmcnum addr data

参数说明：

bmcnum：BMC 号。

addr：寄存器的地址。

data：向寄存器写入的数据。

2.3.42 读 2 字节 BMC 内部寄存器

命令格式：

readioport2 bmcnum addr

参数说明：

bmcnum：BMC 号。

addr：寄存器的地址。

2.3.43 写 4 字节 BMC 内部寄存器

命令格式：

writeioport4(writeioport) bmcnum addr data

参数说明：

bmcnum：BMC 号。

addr：寄存器的地址。

data：向寄存器写入的数据。

2.3.44 读 4 字节 BMC 内部寄存器

命令格式：

readioport4(readioport) bmcnum addr

参数说明：

bmcnum：BMC 号。

addr：寄存器的地址。

2.3.45 显示客户端版本信息

命令格式：

readme [-i]

参数说明：

i：版本更改信息记录。

2.3.46 读取/修改 CPU 的频率配置

cpufreq4a <-o bmcid:midpid:cardid:cpuid >

读取 CPU 的核心频率、存控频率以及核心互联频率

参数说明：

-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

正确返回：

例：BMC 15: core/memory/xbx frequency(set) : 1200MHz / 266.67MHz / 1000MHz

BMC 15: core/memory/xbx frequency(effect) : 1200MHz / 266.67MHz / 1000MHz

cpufreq4a <-o bmcid:midpid:cardid:cpuid> [-c core_freq] [-m mm_freq] [-x xbx_freq] [-p]

修改 CPU 的核心频率、存控频率以及核心互联频率

参数说明：

-o bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

-c core_freq：需要配的核心频率值

-m mm_freq：需要配的存控频率值

-x xbx_freq：需要配的核心互联频率值

-p 打印出频率和值的对应关系

注：每个值都可单独设置

2.4 维护脚本和工具使用说明

2.4.1CPU 复位

srvrst.pl bmcid:midpid:cardid:cupid core_start

复位开发板上的 CPU，并轮询等待复位结束。

参数说明：

bmcid:midpid:cardid:cpuid ：指定 BMC 号：中板号：计算刀片号：CPU 号

midpid，cardid，cpuid 都为 0，为了与 1P 命令兼容，减少软件脚本修改；

corestart：核心断连参数。一个需要复位的 CPU 对应一个参数，0~15 位对应核心 0~15，‘1’表示连接。例如，要起 0 核组，则为 0xf；

正确返回：

CPU 0 reset OK!!!! :<) !!!

SROM initialize passed cores : 0xXX

错误返回：
CPU 0 reset failed twice :<(

2.4.2查看各核打印信息

cgrrk bmcnum core_id
查看内核打印信息。
参数说明：
bmcnum：指定 BMC 号。
core_id：核号。

2.4.3读取 SW410B 的 PC 值

rmpc bmcnum
读取 SW410B 的 PC 值。

2.5 CPU 监测工具的使用

1) 确认 JDK 版本是否为 1.6 以上，否则，升级至 1.6 以上版本。
2) 在根目录下运行 java –jar SW2F_Monitor.jar，会出现如下图型界面。

3 神威平台频率配置（SW2F）

3.1核心时钟配置（返回）

核心时钟配置：根据参考时钟输入引脚（RCLK）的时钟频率和核心时钟配置引脚 CFG_CORE [4:0]_H，通过 PLL 产生核心时钟，在 RCLK 输入时钟频率为 200MHz 的条件下，可配置核心时钟频率为 300～1700MHz。在 PLL 旁路模式下，核心时钟频率即为输入参考时钟 RCLK 的频率。核心时钟配置引脚的具体定义如表 3-1 所示。

表 3-1：核心时钟配置表

CFG_CORE[4:0]_H	X 分频器	Y 分频器	Z 分频器	BW	VCO 频率	输出频率
0	2	16	8	8	1600	200（PLL 旁路）
1	2	24	8	12	2400	300
2	2	32	8	16	3200	400
3	2	20	4	10	2000	500
4	2	24	4	12	2400	600
5	2	28	4	14	2800	700
6	2	32	4	16	3200	800
7	2	18	2	9	1800	900
8	2	19	2	10	1900	950
9	2	20	2	10	2000	1000
10	2	21	2	11	2100	1050
11	2	22	2	11	2200	1100
12	2	23	2	12	2300	1150
13	2	24	2	12	2400	1200
14	2	25	2	13	2500	1250
15	2	26	2	13	2600	1300
16	4	53	2	27	2650	1325
17	2	27	2	14	2700	1350
18	4	55	2	28	2750	1375
19	2	28	2	14	2800	1400
20	4	57	2	29	2850	1425
21	2	29	2	15	2900	1450
22	4	59	2	30	2950	1475
23	2	30	2	15	3000	1500

24	4	61	2	31	3050	1525
25	2	31	2	16	3100	1550
26	4	63	2	32	3150	1575
27	2	32	2	16	3200	1600
28	4	65	2	33	3250	1625
29	2	33	2	17	3300	1650
30	4	67	2	34	3350	1675
31	2	34	2	17	3400	1700

3.2存储控制器时钟配置（返回）

存储控制器时钟配置：根据参考时钟输入引脚（RCLK）和存储器接口配置引脚 CFG_MM [3:0]_H，通过 PLL 输出频率 100MHz～933.33MHz，存储控制器工作时钟 SCLK 为 PLL 输出频率的 2 分频，故 SCLK 的工作频率范围为 50MHz～466.67MHz。PLL 旁路时 SCLK 的工作频率是 50MHz。存储控制器时钟配置引脚的具体定义如表 3-2 所示。

PLL 产生两倍频的时钟用来生成四个 CG 的存控时钟，使得四个 CG 的存控时钟相互错开 90°。具体如下：

存控时钟 0：2 倍频时钟正沿二分频产生；

存控时钟 1：存控时钟 0 的反向生成；

存控时钟 2：2 倍频时钟负沿二分频产生；

存控时钟 3：存控时钟 2 的反向生成。

表 3-2：存储控制器时钟配置表

CFG_MM[3:0]_H	X 分频器	Y 分频器	Z 分频器	BW	VCO 频率	输出频率	SCLK 工作频率
0	6	24	8	12	800	100	50（PLL 旁路）
1	6	64	8	32	2133.333	266.667	133.333
2	6	80	8	40	2666.667	333.333	166.667
3	6	96	8	48	3200	400	200
4	6	56	4	28	1866.665	466.667	233.333
5	6	60	4	30	2000	500	250
6	6	64	4	32	2133.333	533.333	266.667
7	6	68	4	34	2266.666	566.667	283.333
8	6	72	4	36	2400	600	300
9	6	76	4	38	2533.333	633.333	316.667
10	6	80	4	40	2666.666	666.667	333.333
11	6	84	4	42	2800	700	350
12	6	88	4	44	2933.333	733.333	366.667
13	6	92	4	46	3066.667	766.667	383.333

14	6	96	4	48	3200	800	400
15	6	56	2	28	1866.667	933.333	466.667

3.3互连时钟配置（返回）

互连时钟配置：根据参考时钟输入引脚（RCLK）的时钟频率和核心时钟配置引脚 CFG_XBX [3:0]_H，通过 PLL 产生互连时钟，在 RCLK 输入时钟频率为 200MHz 的条件下，可配置核心时钟频率为 300～1300MHz。在 PLL 旁路模式下，互连时钟频率即为输入参考时钟 RCLK 的二分频。互连时钟配置引脚的具体定义如表 3-3 所示。

表 3-3：互连时钟配置表

CFG_X[3:0]_H	X 分频器	Y 分频器	Z 分频器	BW	VCO 频率	输出频率
0	2	16	8	8	1600	100（PLL 旁路）
1	2	24	8	12	2400	300
2	2	32	8	16	3200	400
3	2	20	4	10	2000	500
4	2	24	4	12	2400	600
5	2	28	4	14	2800	700
6	2	32	4	16	3200	800
7	2	18	2	9	1800	900
8	2	19	2	10	1900	950
9	2	20	2	10	2000	1000
10	2	21	2	11	2100	1050
11	2	22	2	11	2200	1100
12	2	23	2	12	2300	1150
13	2	24	2	12	2400	1200
14	2	25	2	13	2500	1250
15	2	26	2	13	2600	1300

4）PCI-E 接口时钟配置：根据 PCI-E 参考时钟输入引脚（PCI_CLK），通过 PLL 产生 PCI-E 接口时钟，在 PCI_CLK 输入时钟频率为 100MHz 的条件下，PCI-E 接口时钟频率为 250MHz。可根据 MIU IOR：PCIEx_CLK_SELx 选择 PCI-E 接口时钟频率与维护时钟 MT_CLK_H 相同，以降低功耗；

4 神威平台频率配置（SW410B）

表 4-1：核心时钟配置表（寄存器）

IOR_CFG_CORE[4:0]	核心工作频率（**MHz**）
0	旁路（200MHz）
1	300
2	400
3	500
4	600
5	700
6	800
7	900
8	1000
9	1050
10	1100
11	1150
12	1200
13	1250
14	1300
15	1325
16	1350
17	1375
18	1400
19	1425
20	1450
21	1475
22	1500
23	1525
24	1550

25	1575
26	1600
27	1625
28	1650
29	1675
30	1700
31	1725

表 4-2：存储控制器时钟配置表（寄存器）

IOR_CFG_MM[3:0]	MC 时钟频率（MHz）
0	旁路
1	233
2	266
3	283
4	300
5	316
6	333
7	350
8	366
9	383
10	400
11	433
12	466
13	500
14	533
15	583

表 4-3：互连时钟配置表（寄存器）

IOR_CFG_XBX[3:0]	CPM 及 IPU 工作频率（MHz）
0	旁路（200MHz）
1	400
2	600
3	800
4	900

5	1000
6	1050
7	1100
8	1150
9	1200
10	1250
11	1275
12	1300
13	1312.5
14	1325
15	1350

附录 A

错误码	错误信息	说明
0x50	SW2 report read back failed,control error	错误读响应（带数据，含控制错）
0x51	SW2 report read back failed,illegal address	非法地址读响应（不带数据）
0x52	SW2 report write failed,illegal address	非法地址写结束
0x53	SW2 report req package parity Error	维护串口校验错响应（不带数据）
0x54	SW2 report illegal cmd ack respond	维护串口非法命令响应（不带数据）
0x55	SW2 report write failed,control error	带控制错写结束
0x56	BMC report wait ack timeout	BMC 维护超时
0x57	BMC report console req package parity Error	BMC 命令校验错响应
0x58	BMC report read or write error	BMC 读写错误
0x59	Console report ack cmd not defined	未定义的响应包