

---

# PrAMC-7311 CentOS 7.1 BBS

User Guide

P/N: 6806800T72B

January 2020

---



© 2020 SMART Embedded Computing™, Inc.

All Rights Reserved.

## Trademarks

The stylized "S" and "SMART" is a registered trademark of SMART Modular Technologies, Inc. and "SMART Embedded Computing" and the SMART Embedded Computing logo are trademarks of SMART Modular Technologies, Inc. All other names and logos referred to are trade names, trademarks, or registered trademarks of their respective owners. These materials are provided by SMART Embedded Computing as a service to its customers and may be used for informational purposes only.

## Disclaimer\*

SMART Embedded Computing (SMART EC) assumes no responsibility for errors or omissions in these materials. **These materials are provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.** SMART EC further does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SMART EC shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. SMART EC may make changes to these materials, or to the products described therein, at any time without notice. SMART EC makes no commitment to update the information contained within these materials.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to a SMART EC website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of SMART EC.

It is possible that this publication may contain reference to or information about SMART EC products, programming, or services that are not available in your country. Such references or information must not be construed to mean that SMART EC intends to announce such SMART EC products, programming, or services in your country.

## Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by SMART Embedded Computing.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

### **SMART Embedded Computing, Inc.**

2900 S. Diablo Way, Suite 190

Tempe, Arizona 85282

USA

\*For full legal terms and conditions, visit [www.smartembedded.com/ec/legal](http://www.smartembedded.com/ec/legal)

# Table of Contents

---

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Software Building Blocks	15
<b>2</b>	<b>Installing the Basic Blade Services Software</b>	<b>17</b>
2.1	Baseline CentOS 7.1 Installation	17
2.2	Package Information	24
2.3	Available BBS Packages	24
2.4	BBS Software Installation	25
2.4.1	HPM and FUF Installation	25
2.4.2	Installation of Patched Drivers	25
<b>3</b>	<b>Firmware Upgrade Facility</b>	<b>29</b>
3.1	Firmware Recovery Image Files	29
3.2	Backup Concept	29
3.3	fcu—Firmware Upgrade Command-Line Utility	30
3.4	Upgrading a Firmware Image	33
3.4.1	BIOS Upgrade	33
3.4.2	IPMC Upgrade	34
<b>4</b>	<b>Hardware Platform Management</b>	<b>37</b>
4.1	hpmagentd—HPM Agent Daemon	38
4.2	hpm—Start-Up Script	40
4.3	hpm—Shutdown and Reboot Scripts	41
4.4	hpmcmd—HPM Command Utility	41
4.4.1	Command Overview	43
4.4.2	Supported Commands	45
4.4.2.1	bootbankget	45
4.4.2.2	bootbankset	46
4.4.2.3	chinfo	46
4.4.2.4	cmd	47
4.4.2.5	deviceid	47
4.4.2.6	frudata	48
4.4.2.7	fruinfoget	48
4.4.2.8	fruinfoget	49

## Table of Contents

---

4.4.2.9	fruinv	51
4.4.2.10	fruread	51
4.4.2.11	fruwrite	52
4.4.2.12	help	52
4.4.2.13	ipmbaddress	53
4.4.2.14	ipmcstatus	53
4.4.2.15	lancfgget	53
4.4.2.16	lancfgset	55
4.4.2.17	ledget	55
4.4.2.18	ledprop	56
4.4.2.19	ledset	56
4.4.2.20	loglevelget	58
4.4.2.21	macaddress	58
4.4.2.22	partnumber	58
4.4.2.23	physlotnumber	59
4.4.2.24	portget	59
4.4.2.25	portset	60
4.4.2.26	postypeget	61
4.4.2.27	postypeset	61
4.4.2.28	sdr	62
4.4.2.29	sdr_dump	64
4.4.2.30	sdrinfo	65
4.4.2.31	sendamc	65
4.4.2.32	sendcmd	66
4.4.2.33	shelfaddress	67
4.4.2.34	shelfslots	67
4.4.2.35	shelftype	67
4.4.2.36	slotmap	68
4.4.2.37	slotnumber	68
4.4.2.38	solcfgget	69
4.4.2.39	solcfgset	70
4.4.2.40	version	70
4.4.2.41	watchdog	71
<b>5</b>	<b>HPI-B Software</b>	<b>73</b>
5.1	Package Information	73
5.2	Configuring the HPI-B client	73

<b>A</b>	<b>Related Documentation .....</b>	<b>75</b>
A.1	SMART Embedded Computing Documentation .....	75
A.2	Related Specifications .....	75
A.3	Additional Resources .....	76

## Table of Contents

---

# List of Figures

---

Figure 1-1	BBS Architecture .....	15
Figure 4-1	Software Levels of the HPM Architecture .....	38

## List of Figures

---



# List of Tables

---

Table 2-1	BBS Packages .....	24
Table 4-1	Command Overview .....	43
Table 5-1	List of HPI-B packages .....	73
Table A-1	SMART EC Publications .....	75
Table A-2	Related Specifications .....	75
Table A-3	Additional Resources .....	76

## List of Tables

---

# About this Manual

---

## Overview of Contents

This manual is divided into the following chapters and appendices.

*Chapter 1, Introduction on page 15*

*Chapter 2, Installing the Basic Blade Services Software on page 17*

*Chapter 3, Firmware Upgrade Facility on page 29*

*Chapter 4, Hardware Platform Management on page 37*

*Chapter 5, HPI-B Software on page 73*

*Appendix A, Related Documentation on page 75*

## Abbreviations

This document uses the following abbreviations:

<b>Abbreviation</b>	<b>Definition</b>
AMC	Advanced Mezzanine Card
AdvancedTCA	Advanced Telecommunications Computing Architecture
ATCA	Advanced Telecommunications Computing Architecture
BBS	Basic Blade Services
BIOS	Basic Input Output System
ECC	Embedded Communications Computing
EVQ	Event Queue
FCU	FUF Command Line Utility
FM	Fault Management
FPGA	Field Programmable Gate Array
FRI	Firmware Recovery Image
FRU	Field Replaceable Unit
FUF	Firmware Upgrade Facility
GPIO	General Purpose Input/Output
HPI	Hardware Platform Interface



---






<b>Abbreviation</b>	<b>Definition</b>
HPM	Hardware Platform Management
IPMB	Intelligent Platform Management Bus
IPMC	Intelligent Platform Management Controller
IPMI	Intelligent Platform Management Interface
LUN	Logic Unit Number
MAC	Media Access Control
MMC	Module Management Controller
OEM	Original Equipment Manufacturer
PCI	Peripheral Component Interconnect
PICMG	PCI Industrial Computers Manufacturers Group
RMCP	Remote Monitoring and Control Protocol
RPM	RedHat Package Manager
RTM	Rear Transition Module
SAF	Service Availability Forum
SATA	Serial ATA
SDR	Sensor Data Record
SMI	Serial Management Interface
SNMP	Simple Network Management Protocol
SOL	Serial Over Lan
SSU	Synchronization Supply Unit
TAR	Tape Archiver
TFTP	Trivial File Transfer Protocol

---

## Conventions

The following table describes the conventions used throughout this manual.

Notation	Description
0x00000000	Typical notation for hexadecimal numbers (digits are 0 through F), for example used for addresses and offsets
0b0000	Same for binary numbers (digits are 0 and 1)
<b>bold</b>	Used to emphasize a word
Screen	Used for on-screen output and code related elements or commands. Sample of Programming used in a table (9pt)
<b>Courier + Bold</b>	Used to characterize user input and to separate it from system output
<i>Reference</i>	Used for references and for table and figure descriptions
File > Exit	Notation for selecting a submenu
<text>	Notation for variables and keys
[text]	Notation for software buttons to click on the screen and parameter description
...	Repeated item for example node 1, node 2, ..., node 12
.	Omission of information from example/command that is not necessary at the time
..	Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)
	Logical OR
	Indicates a hazardous situation which, if not avoided, could result in death or serious injury
	Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury

Notation	Description
	Indicates a property damage message
	Indicates a hot surface that could result in moderate or serious injury
	Indicates an electrical situation that could result in moderate injury or death
<p data-bbox="272 725 386 777"><b>Use ESD protection</b></p> 	Indicates that when working in an ESD environment care should be taken to use proper ESD practices
	No danger encountered, pay attention to important information

## Summary of Changes

Part Number	Date	Description
6806800T72B	January 2020	Rebranded to SMART Embedded Computing template.
6806800T72A	September, 2015	Initial version.

# Introduction

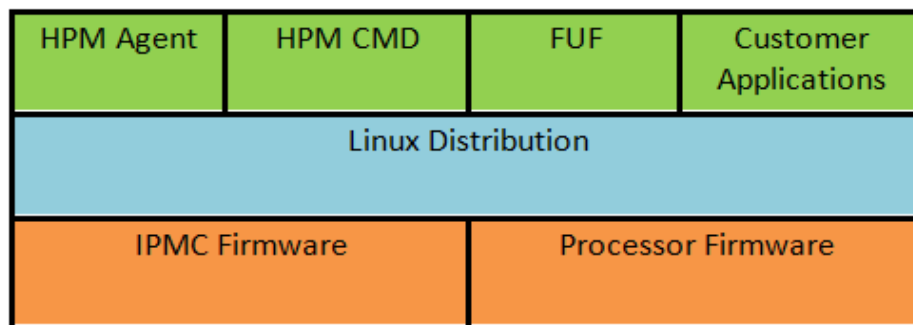
## 1.1 Software Building Blocks

Basic Blades Services (BBS) software provides useful packages and board specific driver patches that are not available from the open-source community.

BBS software provides additional functionality over a baseline Linux installation for AdvancedTCA blades and AMC modules. The functionality is tailored to a particular blade or module.

The figure below depicts the architecture of BBS software.

*Figure 1-1 BBS Architecture*



BBS software provides the following functionality:

### Firmware Upgrade Facility

The Firmware Upgrade Facility (FUF) provides a uniform way to upgrade firmware on SMART EC blades and AMC modules regardless of which flash locations store the firmware. The FUF can upgrade the BIOS firmware as well as the IPMC firmware. The FUF currently consists of a Firmware Upgrade Command Line Utility (FCU) and flash device drivers. It supports specially prepared firmware recovery image (FRI) files as well as HPM.1 compatible firmware files. The FUF can be used on switch and node blades and on AMC modules.

### Hardware Platform Management

HPM in AdvancedTCA systems is based on the Intelligent Platform Management Interface specification (IPMI). IPMI commands can be complex and cumbersome. HPM provides a more user-friendly method to do blade or module-level hardware management.

## Software Building Blocks

---

### HPM Agent

The HPM agent daemon handles local communication to the intelligent platform management controller (IPMC) on a module using the SMI. It invokes shutdown or reboot scripts based on received IPMI requests.



# Installing the Basic Blade Services

---

## Software

The CentOS 7.1 BBS software is expected to be installed natively on a PrAMC-7311 that has CentOS 7.1 installed on the SATA HDD mounted on the ATCA-F140. The HDD can be routed to the ATCA-F140 or to the PrAMC-7311. The HDD is routed to the PrAMC-7311 by running a `hpm muxctrl` command on the ATCA-F140.

```
hpmcmd -c muxctrl mux15 B2
```

The serial port settings for the PrAMC-7311 face plate serial port are: 115200 baud, no parity, eight data bits, and one stop bit.

## 2.1 Baseline CentOS 7.1 Installation

You can install baseline CentOS 7.1 by using CentOS provided images and install guides, or by using a method provided in the CentOS 7.1 software bundle.

The method provided in the software bundle to install a baseline CentOS is to PXE boot the provided `kernel` and `initramfs` and run an interactive tool after booting to install to a disk attached to the PrAMC-7311. This method does not have the flexibility of the default CentOS installer. For example, the installation will use EXT4 file-systems and will not use LVM. If the method provided does not meet your needs, install baseline CentOS using the standard CentOS provided media/images and installation guides.

The following files are provided in the software package:

- `vmlinuz-3.10.0-229.11.1.el7.x86_64`
- `initramfs-3.10.centos.cpio.gz`
- `pxelinux.default`

Copy `vmlinuz-3.10.0-229.11.1.el7.x86_64` and `initramfs-3.10.centos.cpio.gz` files to your PXE server. Then use a PXE configuration file similar to `pxelinux.default` to boot the kernel and `initramfs`.

After the kernel boots to login prompt, login as `root/root`.

You can then run the `itd_legacy_centos` tool to install CentOS 7.1 to a connected disk. Before you run this tool, copy the `initramfs-3.10.centos.cpio.gz` file to a directory off the root `tftpboot` directory on your TFTP server. Then create a `files.sha1sum` file in the same directory which contains the output of `sha1sum initramfs-3.10.centos.cpio.gz`. The `initramfs-3.10.centos.cpio.gz` and `files.sha1sum` files should be the only files in this directory.

## Baseline CentOS 7.1 Installation

---

For example,

```
# cd /tftpboot
# mkdir centos
# cp <.../initramfs-3.10.centos.cpio.gz>.
# shalsum initramfs-3.10.centos.cpio.gz > files.shalsum
You can now run "itd_legacy_centos" to install and the log file
will look something like this:
# itd_legacy_centos
*****
This script will install a supplied kernel and RFS to a local disk
*****
*****
Checking for necessary commands...
    awk..... [exists]
    chroot..... [exists]
    mount.....[exists]
    umount.....[exists]
    tar.....[exists]
    gzip.....[exists]
    mkdir.....[exists]
    rmdir.....[exists]
    rm.....[exists]
    cp.....[exists]
    mv.....[exists]
    date.....[exists]
    chmod.....[exists]
    chown.....[exists]
    grep.....[exists]
    dd.....[exists]
    stty.....[exists]
    sed.....[exists]
    fsck.ext3.....[exists]
    ifconfig.....[exists]
    mke2fs.....[exists]
    mkswap.....[exists]
    reboot.....[exists]
```

```
fdisk.....[exists]
parted.....[exists]
hwclock.....[exists]
blkid.....[exists]
cut.....[exists]
sftp.....[exists]
seq.....[exists]
shalsum.....[exists]
tr.....[exists]
ntupdate.....[exists]
```

Necessary commands found, safe to continue...

-----

The following disks are available:

/dev/sda: 500GB

Select the disk/flash device where you want to have the filesystem installed

(e.g. sda) []: sda

Grub Disk Name: hd0

-----

Do you wish to begin the installation? [y/n] y

Verifying hard drive size of /dev/sda...HDD size is 500000 MB

root size is 483171 MB

GNU Parted 3.1

Using /dev/sda

Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) unit MB

(parted) mkpart primary 1 512

(parted) mkpart primary 512 483683

(parted) mkpart primary linux-swap 483683 -0

(parted) set 1 boot on

(parted) q

Information: You may need to update /etc/fstab.

Formatting...1...2...Done

Creating Swap...Done

Boot UUID: ebb81ce1-0312-4090-aaee-49cc40316514

Root UUID: 0046bb6f-3a10-4ac2-aaaf-93520c6d9fa9

## Baseline CentOS 7.1 Installation

---

```
Swap UUID: 04695dfe-e828-45e0-a0d7-6f44a4fa8eac
Which TFTP server do you wish to use? [xxx.xxx.xxx.xxx]
10.206.154.6
TFTP Server active.
What is the installation files directory? [] /tftpboot/centos
What is the tftp user name? [] aprem
Downloading files.shalsum from /tftpboot/centos
The authenticity of host '10.206.154.6 (10.206.154.6)' can't be
established.
ECDSA key fingerprint is
62:f9:f1:52:30:f2:56:22:31:ad:46:1c:f5:77:aa:e5.
Are you sure you want to continue connecting (yes/no)? yes
aprem@10.206.154.6's password:
....Done
Downloading initramfs-3.10.centos.cpio.gz from /tftpboot/centos...
aprem@10.206.154.6's password:
Done.
Do you wish to set the TimeZone? [Y/n]
Please identify a location so that time zone rules can be set
correctly.
Please select a continent or ocean.
1) Africa
2) Americas
3) Antarctica
4) Arctic Ocean
5) Asia
6) Atlantic Ocean
7) Australia
8) Europe
9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the Posix TZ format.
#? 2
```

- |                          |                             |
|--------------------------|-----------------------------|
| 1) Anguilla              | 28) Haiti                   |
| 2) Antigua & Barbuda     | 29) Honduras                |
| 3) Argentina             | 30) Jamaica                 |
| 4) Aruba                 | 31) Martinique              |
| 5) Bahamas               | 32) Mexico                  |
| 6) Barbados              | 33) Montserrat              |
| 7) Belize                | 34) Nicaragua               |
| 8) Bolivia               | 35) Panama                  |
| 9) Brazil                | 36) Paraguay                |
| 10) Canada               | 37) Peru                    |
| 11) CaribbeanNetherlands | 38) Puerto Rico             |
| 12) Cayman Islands       | 39) St Barthelemy           |
| 13) Chile                | 40) St Kitts & Nevis        |
| 14) Colombia             | 41) St Lucia                |
| 15) Costa Rica           | 42) St Maarten (Dutch part) |
| 16) Cuba                 | 43) St Martin (French part) |
| 17) Curacao              | 44) St Pierre & Miquelon    |
| 18) Dominica             | 45) St Vincent              |
| 19) Dominican Republic   | 46) Suriname                |
| 20) Ecuador              | 47) Trinidad & Tobago       |
| 21) El Salvador          | 48) Turks & Caicos Is       |
| 22) French Guiana        | 49) United States           |
| 23) Greenland            | 50) Uruguay                 |
| 24) Grenada              | 51) Venezuela               |
| 25) Guadeloupe           | 52) Virgin Islands (UK)     |
| 26) Guatemala            | 53) Virgin Islands (US)     |
| 27) Guyana               |                             |

#? 49

## Baseline CentOS 7.1 Installation

---

Please select one of the following time zone regions.

- 1) Eastern Time
- 2) Eastern Time - Michigan - most locations
- 3) Eastern Time - Kentucky - Louisville area
- 4) Eastern Time - Kentucky - Wayne County
- 5) Eastern Time - Indiana - most locations
- 6) Eastern Time - Indiana - Daviess, Dubois, Knox & Martin Counties
- 7) Eastern Time - Indiana - Pulaski County
- 8) Eastern Time - Indiana - Crawford County
- 9) Eastern Time - Indiana - Pike County
- 10) Eastern Time - Indiana - Switzerland County
- 11) Central Time
- 12) Central Time - Indiana - Perry County
- 13) Central Time - Indiana - Starke County
- 14) Central Time - Michigan - Dickinson, Gogebic, Iron & Menominee Counties
- 15) Central Time - North Dakota - Oliver County
- 16) Central Time - North Dakota - Morton County (except Mandan area)
- 17) Central Time - North Dakota - Mercer County
- 18) Mountain Time
- 19) Mountain Time - south Idaho & east Oregon
- 20) Mountain Standard Time - Arizona (except Navajo)
- 21) Pacific Time
- 22) Pacific Standard Time - Annette Island, Alaska
- 23) Alaska Time
- 24) Alaska Time - Alaska panhandle
- 25) Alaska Time - southeast Alaska panhandle
- 26) Alaska Time - Alaska panhandle neck
- 27) Alaska Time - west Alaska
- 28) Aleutian Islands
- 29) Hawaii
- #? 20

The following information has been given:

United States

Mountain Standard Time - Arizona (except Navajo)

Therefore, TZ= 'America/Phoenix' will be used.

```
Local time is now: Fri Sep 11 15:19:11 MST 2015.
Universal Time is now: Fri Sep 11 22:19:11 UTC 2015.
Is the above information OK?
1) Yes
2) No
#? 1
```

You can make this change permanent by appending the line `TZ= America/Phoenix; export TZ` to the file `.profile` in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you can use the `/usr/bin/tzselect` command in shell scripts:

```
America/Phoenix
Do you wish to use NTP to set the current time? [Y/N]
Please enter the NTP server address [xxx.xxx.xxx.xxx] 127.0.0.1
Checking SHA1 Checksum for initramfs-3.10.centos.cpio.gz...Good
Done
echo "Timezone set to: America/Phoenix" in /etc/profile
-----
Beginning final configuration
-----
Choose a hostname for this machine [] pramc7311
Set Root password
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: All authentication tokens updated successfully.
Root password set.
Done.
Installing for i386-pc platform.
Installation finished. No error reported.
Generating grub configuration file...
Found linux image: /boot/vmlinuz-3.10.0-229.11.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-229.11.1.el7.x86_64.img
done
Unmounting local filesystems...done.
```

## Package Information

---

-----  
-----  
After the tool completes reboot, break into the BIOS to select the disk you installed as the first choice, and boot off this disk.



The CentOS installed in this procedure already uses the updated drivers, and has all the BBS packages installed. However, you still need to configure HPI-B, as described in the README.HPI-B document included in the software bundle.

## 2.2 Package Information

BBS software packages are packaged as RPM packages. Driver patches are against the baseline drivers in the CentOS 7.1 kernel.

The following RPM commands are useful to list /install/remove packages.

Command	Description
<code>rpm -qa</code>	List all installed packages. Use <code>rpm -qa   grep hpi</code> to list only HPI packages.
<code>rpm -i package-name</code>	Install a package, where package-name is the name of a specific package, for example, <code>rpm -i openhpi</code> .
<code>rpm -e package-name</code>	Uninstall a package, where package-name is the name of a specific package, for example, <code>rpm -e openhpi</code> .

For more information about the rpm command, see its `man` page.

## 2.3 Available BBS Packages

The BBS distribution contains the following packages.

*Table 2-1 BBS Packages*

File	Description
<code>bbs-fcu-pramc7311-3.9-13-centos7.rpm</code>	CentOS X86_64 FUF software package
<code>bbs-hpmcmd-pramc7311-3.10-3-4-4-centos7.rpm</code>	CentOS X86_64 HPM software package
<code>bbs-hpmagent-pramc7311-3.4-0-centos7.rpm</code>	CentOS X86_64 HPMAGENT software



## 2.4 BBS Software Installation

This section describes the installation of the HPM and FUF packages and also the installation of the patched drivers.

### 2.4.1 HPM and FUF Installation

The HPM and FUF software is delivered in a tar ball, named `PrAMC7311-CentOS-BBS-<version>.tar.gz`. This tar file includes the software packages for HPM, HPMAgent, FUF, and HPI-B, and some patched driver modules. Copy this tar file to the target PrAMC-7311 to any suitable location for installation.

Before installing these modules, install the `OpenIPMI` package using the following yum command:

```
# yum install OpenIPMI
```

To install HPM and FUF execute the following steps:

1. Run the shell command `tar` to unzip the package tarball, for example:

```
# tar -zxvf PrAMC7311-CentOS-BBS-<version>.tar.gz
```

```
# cd PrAMC7311-CentOS-BBS-<version>
```

2. Run the shell script `install.sh` to install the BBS utility packages:

```
# ./install.sh
```

3. After completing the installation, add `/opt/bladervices/bin` to the `PATH` environment variable or logout and login again:

```
# export PATH=$PATH:/opt/bladervices/bin
```

To uninstall HPM and FUF, execute the following shell command:

```
# uninstall
```

### 2.4.2 Installation of Patched Drivers

Patched `e1000e` and `ipmi-poweroff` drivers are provided to address functionality/issues specific to the PrAMC-7311.

The patch to the `ipmi-poweroff` driver provides an AMC power-off function.

The patch to the `e1000e` driver addresses slow link loss detection on the Intel dual-port 82571EB ports.

The patched `e1000e` driver is provided in the `e1000e.tar.gz` file, and the patched `ipmi` driver is provided in the `ipmi.tar.gz` file. The baseline drivers are from the CentOS 7.1 kernel (kernel 3.10.0-229.11.1).

## Installation of Patched Drivers

---

The kernel-devel package has to be installed before installing the patched drivers. If the package is not already installed, it can be installed using the command: `yum install kernel-devel`.

The patched e1000e driver can be installed using the following commands:

```
# cd <BSP Install Dir>/LSP
# tar zxvf e1000e.tar.gz
# cd e1000e
# make -C /lib/modules/`uname -r`/build M=`pwd` modules
# cp e1000e.ko /lib/modules/`uname -r`/build M=`pwd` modules
# depmod -a
```

To use the updated driver without having to reboot, take down the interfaces that use the e1000e driver (for example: “`ifconfig enp2s0f0 down`” and “`ifconfig enp2s0f1 down`”), unload the driver using “`rmmmod e1000e`”, load the updated driver using “`modprobe e1000e`”, and configure the interfaces that the e1000e driver controls again.



**The driver source provided is from the CentOS 7.1 kernel (3.10.0-229.11.1). If you use a different kernel, the provided driver may not build when using the instructions above. If you use a different kernel, then you should apply the patch provided (e1000e.patch) to the e1000e driver source from the kernel you use and build an updated driver.**

For example:

```
$ cd <kernel_src>/drivers/net/ethernet/intel/e1000e
$ cp <provided e1000e.patch> here
$ patch -Np1 < e1000e.patch (if the patch applies without errors,
you can build)
$ make -C /lib/modules/`uname -r`/build M=`pwd` modules
# cp e1000e.ko /lib/modules/`uname-r`/kernel/drivers/net/
ethernet/intel/e1000e
# depmod -a
```

The patched ipmi-poweroff driver can be installed using the following commands:

```
# cd <BSP Install Dir>/LSP
# tar zxvf ipmi.tar.gz
# cd ipmi
# make -C /lib/modules/`uname -r`/build M=`pwd` modules
# cp ipmi_poweroff.ko /lib/modules/`uname- r`/kernal/driver/
char/ipmi
# depmod -a
```

To use the updated driver, unload the ipmi-poweroff driver if it is already loaded using `rmmod ipmi_poweroff`, and load the updated driver using `modprobe ipmi_poweroff`.



**The driver source provided is from the CentOS 7.1 kernel (3.10.0-229.11.1). If you use a different kernel the provided driver may not build when using the instructions above. If you use a different kernel then you should apply the patch provided to the IPMI driver source from the kernel you use and build an updated driver.**

For example:

```
# cd <BSP Install Dir>/LSP
# tar zxvf ipmi.tar.gz
# cd ipmi
# make -C /lib/modules/`uname -r`/build M=`pwd` modules
# cp ipmi_poweroff.ko /lib/modules/`uname -r`/kernel/driver/
char/ipmi
# depmod -a
```

## Installation of Patched Drivers

---

# Firmware Upgrade Facility

---

The Firmware Upgrade Facility (FUF) provides a uniform way to upgrade firmware on SMART EC hub blades, node blades, and AMC modules. It consists of a Firmware Upgrade Command-line Utility (FCU), flash device drivers, and specially prepared firmware recovery image files.

## 3.1 Firmware Recovery Image Files

The PrAMC-7311 supports the HPM.1 IPMI standard. FCU works with specially prepared firmware recovery image (FRI) files or with HPM.1 compatible files. The following firmware packages are currently available:

Filename	Description
pramc-7311-cpu.fri	BIOS firmware image for PrAMC-7311
pramc-7311-ipmc-boot-<version>.hpm	IPMC bootloader
pramc-7311-ipmc-<version>.hpm	IPMC firmware

## 3.2 Backup Concept

The BIOS firmware for the PrAMC-7311 is stored in redundant, persistent memory devices. This allows the firmware image in one bank to serve as a backup for the other bank. This is particularly useful for firmware upgrades.

During normal operation, the CPU determines which bank to boot from based on a chip select signal controlled by the IPMC. This bank is considered the active boot device. FCU only allows you to upgrade an inactive device. It determines whether a device is active or inactive by querying the IPMC to learn which device is marked to be used at boot. Because you can change the active device with the FCU mark operation, active status does not necessarily indicate which device was used on the last boot. It simply represents which device is set for use on the next boot.

The IPMC firmware consists of a boot loader as well as an active and a stand-by IPMI firmware. The boot loader maintains both the active and stand-by firmware in the flash memory of the PrAMC-7311. Both the boot loader as well as the IPMI firmware images can be upgraded.

## fcu—Firmware Upgrade Command-Line Utility

---

Each time the IPMC firmware is upgraded, the most recent firmware version is kept in flash memory and the older firmware version is overwritten by the new one. Once the new IPMI firmware is programmed, the IPMC resets itself to boot from the new image. The boot loader validates the new IPMC firmware. Provided the IPMC can power up successfully the current image is made active and the previously active image is made backup. In case of power-up failures, the boot loader automatically recovers from crisis and boots from the previous image.

### 3.3 fcu—Firmware Upgrade Command-Line Utility

#### Description

The Firmware Upgrade Command-line Utility (FCU) allows you to:

- Query the current versions of firmware installed on a module and determine which firmware devices are active
- Verify that a specified upgrade image is sound and compatible with the current hardware
- Upgrade a firmware image
- Mark a device to be used as the boot source on the next reset
- Display detailed information about a firmware upgrade image file

By default, the FCU binary executable is installed in `/opt/bladervices/bin`. This directory has been added to the `PATH` environment variable.

FCU works in conjunction with device drivers created specifically for the flash devices on SMART EC modules.

The FCU verify and upgrade operations require specially prepared FRI files or HPM.1 compatible image files. For more information, see [Firmware Recovery Image Files](#).

#### Synopsis

```
fcu --help [-t<slave address>]
fcu --version
fcu -q [-d <device-id>]
fcu -v -f <filename>
fcu -u -f <filename>
fcu -a -f <filename>
fcu -s <filename>
fcu -m -b <bank-letter> -d <device-id>
```

### Parameters

`-b <bank-letter>`  
`--bank=<bank-letter>`

Specifies the flash bank (For example: A/B or 0/1), to mark for next boot, where `<bank-letter>` is the letter designating a specific bank. This option is used with the mark operation. Use the query option `-q` to list available banks.

`-d <device-id>`  
`--device=<device-id>`

Specifies a target firmware device, where `<device-id>` is the name of the device. This option is used with the mark or query operations. Device ID values vary by hardware. You can display supported devices on a given module by using `fcu -help`. Currently supported values are listed in the following table.

Device ID	Description
pramc-7311-cpu.fri	CPU firmware device for PrAMC-7311
H8S-AMCm F/W	IPMC firmware device for PrAMC-7311
H8S-AMCm B/L	IPMC boot loader device for PrAMC-7311

`-f <filename>`  
`--file=<filename>`

Specifies the FRI or HPM.1 image file, where `<filename>` is the complete path and filename of the file. This option is used with the verify and upgrade operations.

`--help`

Displays a brief message describing command usage. It also displays a list of the devices supported on the module. This option is exclusive and should not be used with other options.

`-m`  
`--mark`

Tells FCU to set the boot select so that on the next boot the specified firmware bank will be active.

Currently, the mark operation only supports CPU firmware devices.

`-q`  
`--query`

## fcu—Firmware Upgrade Command-Line Utility

---

Tells FCU to return firmware information for a specific device (if used with -d) or information about all firmware devices. The query operation is exclusive and is not intended to be combined with other operations.

```
-s  
--show
```

Display information about the target which the selected upgrade file corresponds to. Use this command to display the current firmware version status of the target device.

```
-u  
--upgrade
```

Tells FCU to upgrade the currently inactive bank of the device specified by the target FRI file. The file option -f is required. The upgrade operation may be combined with the verify and mark operations.

```
-v  
--verify
```

Tells FCU to verify the image file specified by the required -f option. This operation verifies that the specified file is sound and compatible with the current hardware. The verify operation may be combined with the upgrade and mark operations.

```
--version
```

Displays version information for the utility. This option is exclusive and should not be used with other options.

### Usage

Some FCU options can be combined. Some options are exclusive. The following list describes the valid option combinations:

```
--help  
--mark --bank=<bank-letter> --device=<device-id>  
--query  
--query --device=<device-id>  
--show --file=<filename>  
--upgrade --file=<filename>  
--verify --file=<filename>  
--verify --upgrade --file=<filename>  
--version
```



Multi character options may be abbreviated so long as they are unique. For example, `--upg` is equivalent to `--upgrade`. Typing `--ver`, however, will not work since it matches both `--verify` and `--version`.

Single-character options may be combined without repeating the hyphen, as in these examples:

```
fcu -vf <filename>
fcu -q -d <device-id>
```

Options are not case-sensitive. For example, `--help` is equivalent to `--HeLp`. However, option arguments, such as filename and device ID, are case-sensitive.

When upgrading firmware, it is strongly recommended that you upgrade only one device at a time. While FCU performs many checks during upgrade to ensure success, if something goes wrong and both firmware banks become corrupted, the module will be inoperable.

## 3.4 Upgrading a Firmware Image

This section describes recommended procedures for upgrading firmware devices. The procedures for upgrading the BIOS and IPMC differ slightly.

### 3.4.1 BIOS Upgrade

The BIOS can only be upgraded from the module on which the BIOS is running. You have to upgrade the BIOS by using `fcu`. The `hpmcmd` command cannot be used to upgrade the BIOS.

#### Upgrading the BIOS for PrAMC-7311

Follow these steps to upgrade the BIOS.

1. Query the current BIOS firmware images on the module.

```
fcu -qd <device-id>
```

where `<device-id>` is the name of the device to be upgraded.

2. Verify the integrity of the upgrade file,  

```
fcu --verify -f <path+filename>
```

where `<path+filename>` is the complete path to the upgrade file, for example:

```
fcu --verify -f /opt/bladeservices/rom/pramc-7311-cpu.fri
```

Be sure the upgrade image is actually newer than the current firmware image.

## IPMC Upgrade

---

3. Upgrade the firmware image.

```
fcu --upgrade -f <path+filename>
```

where <path+filename> is the complete path to the upgrade file, for example:

```
fcu --upgrade -f /opt/bladeservices/rom/pramc-7311-cpu.fri
```

FCU writes the new image and then reads back the image and performs a binary compare to ensure that the write was successful. If the upgrade was not successful, you will see an error message. Try the upgrade again. If it is still not successful, contact your SMART EC representative.

4. Query the new image to ensure that the version information is correct,

```
fcu -qd <device-id>
```

where <device-id> is the name of the upgraded device, for example:

```
fcu -qd pramc-7311-cpu
```

5. Mark the new image as active so that it will be used for the next boot, for example:

```
fcu --mark -b <bank-letter> -d <device-id>
```

where <bank-letter> is the letter or number of the upgraded bank and <device-id> is the name of the upgraded device, for example:

```
fcu --mark -b a -d pramc-7311-cpu
```

Or

```
fcu --mark -b 0 -d pramc-7311-cpu
```

### 3.4.2 IPMC Upgrade

#### Upgrading IPMC for PrAMC-7311

Follow these steps to upgrade an IPMC:

1. Query the current IPMC firmware images on the module.

```
fcu -qd <device-id>
```

where <device-id> is the name of the device to be upgraded.

2. Verify the integrity of the upgrade file,

```
fcu --verify -f <path+filename>
```

where <path+filename> is the complete path to the upgrade file, for example:

```
fcu --verify -f /opt/bladeservices/rom/pramc-7311-ipmc.fri
```

Be sure the upgrade image is actually newer than the current firmware image.

3. Upgrade the firmware image,

```
fcu --upgrade -f <path+filename>
```

where <path+filename> is the complete path to the upgrade file, for example:

```
fcu --upgrade -f /opt/bladeservices/rom/pramc-7311-ipmc.fri
```

FCU writes the new image and then reads back the image and performs a binary compare to ensure that the write was successful. If the upgrade was not successful, you will see an error message. Try the upgrade again. If it is still not successful, contact your SMART EC representative.

4. Query the new image to ensure that the version information is correct,

```
fcu -qd <device-id>
```

where <device-id> is the name of the upgraded device, for example:

```
fcu -qd pramc-7311-ipmc
```

If the version you just installed is now the active image, the upgrade was successful.



# Hardware Platform Management

---

Hardware management is based on the Intelligent Platform Management Interface (IPMI) specification. IPMI commands can be complex and cumbersome. To facilitate module-level management, SMART EC provides the Hardware Platform Management (HPM) package that provides a set of commands that are based on IPMI commands but which are easier to use than the IPMI commands. An HPM command can encapsulate a sequence of IPMI commands. An HPM command can be the unifier for OEM IPMI commands that are different on different module types, for example reading the BIOS boot bank. For a catalogue of supported IPMI commands of the module refer to the respective IPMI manual.



**Since HPM provides full access to the underlying IPMI subsystem, HPM commands should be used with special care. Improper usage of HPM commands may cause system corruption or system failure. Examples of improper usage are modifying FRU data, change blue led state, change e-keying, etc.**

The HPM package consists of

- HPM daemon called `hpmagentd`
- Command line client called `hpmcmd`
- Script framework for managing shutdown and reboot events

The `hpmcmd` executes the incoming HPM commands and returns the result

HPM commands include:

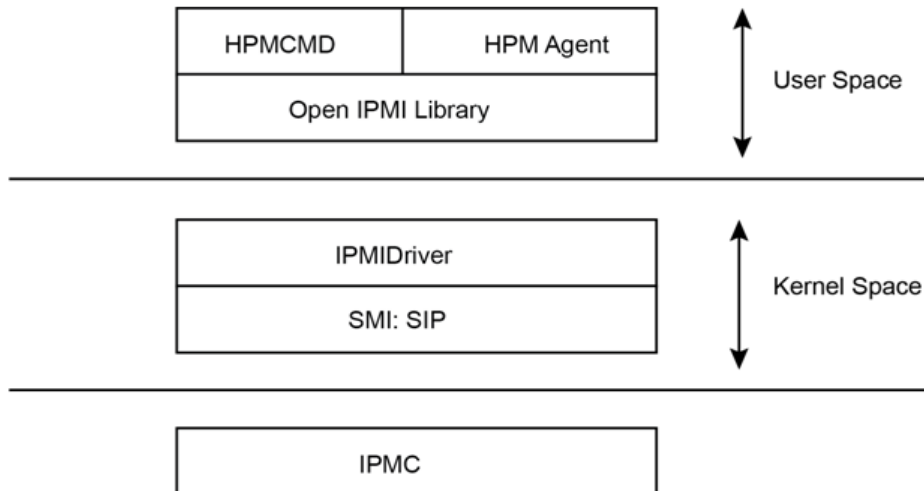
- Retrieving FRU data
- Reading and controlling status of IPMI-controlled LEDs
- Communicating local slot location information

HPM Agent executes shutdown and reboot scripts in response to shutdown or graceful reboot events received.

## hpmagentd—HPM Agent Daemon

The `hpmcmd` and `hpmagentd` makes use of OpenIPMI to talk to the local IPMC using the SMI. This SMI gets set up by the OpenIPMI driver. OpenIPMI consists of two main parts: A device driver that goes into the Linux kernel, and a user-level library. The following picture shows the software levels that are involved in the HPM architecture:

*Figure 4-1 Software Levels of the HPM Architecture*



SMI: System Management Interface  
SIP: Serial Interface Protocol

The SMI (System Management Interface) driver provides the low level interface for talking to the IPMC. The communication is based on a serial interface protocol.

If you need more information about the software aspects of the blade and module IPM controller, refer to the respective IPMI manual.

## 4.1 hpmagentd—HPM Agent Daemon

### Description

The HPM agent daemon handles local communication to the intelligent platform management controller (IPMC) on a module using the SMI. This SMI gets set up by the OpenIPMI driver.

By default, the `hpmagentd` binary executable is installed in `/opt/bladeservices/bin/`. This directory has been added to the `PATH` environment variable.

This daemon has an init script called `hpm` that will start the daemon in run level 3 with the default settings.

When `hpmagentd` receives a graceful reboot or shutdown alert from the IPMC, it will call the respective script to run the reboot or shutdown sequence.

### Synopsis

```
hpmagentd [-l log-level] [-r reboot-script] [-s shutdown-script]
hpmagentd {-i | -h | -v}
```

### Parameters

`-l log-level`

Specifies the level of message logging, where `log-level` is one of the standard syslog levels:

Log Level	Description
0	Emergency
1	Alert
2	Critical
3	Error
4	Warning
5	Notice (default)
6	Information
7	Debug

`-r reboot-script`

Specifies a graceful reboot script that will be called when a module graceful reboot request is received by the MMC, where `reboot-script` is the complete path and filename of the target script. The default is `/opt/bladervices/bin/hpmreboot` (see [hpm—Shutdown and Reboot Scripts on page 41](#)).

`-s shutdown-script`

Specifies a shutdown script that will be called when a module shutdown request is received by the MMC, where `shutdown-script` is the complete path and filename of the target script. The default is `/opt/bladervices/bin/hpmshutdown` (see [hpm—Shutdown and Reboot Scripts on page 41](#)).

## hpm—Start-Up Script

---

-i

hpmagentd runs interactively, that is it will not run as daemon.

-h

Displays a brief message about command usage.

-v

Displays the version of hpmagentd

## 4.2 hpm—Start-Up Script

### Description

An HPM agent init script, hpm, allows you to start, stop, and restart the HPM agent daemon using the agent's default option settings. By default, this script is installed in the `/opt/bladesevices/etc/init.d` directory during installation of the BBS software. The install script installs a system hpm service to automatically start the HPM agent when the system boots. The hpm agent can be started/restarted/stopped etc. using `systemctl` ("systemctl status hpm").

### Synopsis

```
hpm {start | stop | restart | reload | status}
```

### Parameters

start

Starts the hpm agent daemon.

stop

Terminates the hpm agent daemon.

restart

Terminates and then starts the hpm agent daemon.

reload

Terminates and then starts the hpm agent daemon.



## 4.3 hpm—Shutdown and Reboot Scripts

### Description

At any time during normal operation, a shelf manager may issue a shutdown (FRU Activation Deactivate) or graceful reboot (FRU Control Reboot) request to the MMC on a given module. The MMC then forwards this information to the HPM agent. The HPM agent listens for such requests from the MMC. When it receives a request, it calls the respective script to run the reboot or shutdown sequence. In case of a shutdown indication, all running processes should be notified about the shutdown. In case of a reboot notification, the payload is responsible for invoking the reboot procedure. The MMC is not involved in this process. This allows processes currently running on the module to prepare for shutdown. After the notification, it takes roughly 30 seconds before the payload is powered off.

Two default scripts, `hpmshutdown` and `hpmreboot`, are installed by default in the `/opt/bladervices/bin` directory. Currently, these scripts simply print a banner indicating they have run and then issue `shutdown -h now` (`hpmshutdown` script) or `reboot` (`hpmreboot` script).

You may modify the default scripts to suit the needs of your system application or create new scripts. If you create new scripts, use the `-s` and `-r` options when starting `hpmagentd` to specify the new locations and names of the scripts. You may also need to update the `hpm` start up script in `/opt/bladervices/etc/init.d/hpm`.

### Synopsis

```
hpmshutdown
```

```
hpmreboot
```

## 4.4 hpmcmd—HPM Command Utility

### Description

The HPM command utility accepts commands from the user and executes them. Once a command is sent, the `hpmcmd` program waits until the answer from the IPMC is received or until a time-out occurs.

The HPM command utility can be started in interactive mode, where a prompt is displayed and the user enters commands; it can read in a file of commands; or it can process a single command.

By default, the `hpmcmd` binary executable is installed in `/opt/bladervices/bin`. During installation of the BBS software, this directory is added to the `PATH` environment variable.

## hpmcmd—HPM Command Utility

---

If you do not provide any options you will see the following prompt once the program starts running:

```
hpmcmd>
```

From there you can start executing commands.

### Synopsis

```
hpmcmd [options]
```

```
-- help
```

```
-c process a single command
```

```
-h displays this help message
```

```
-v verbose mode for some commands
```

```
-t send the command to a remote target
```

```
-f file option used by some hpcmds
```

```
-p change the prompt
```

### Parameters

```
-p new-prompt
```

Specifies the prompt you would like to have for the hpmcmd interactive mode, where new-prompt is any string. The default prompt is hpmcmd>. This option should not be combined with the -r or -c options.

```
-c command
```

This option executes a single command and terminates, where command is one of the supported commands. This allows you to use the arrow history functions supported in the base shell; a history is not available inside the hpmcmd program. This option should not be combined with the -i option.

If this option is combined with -o, -c should be last option entered, since all arguments that follow -c on the command line will be considered part of the command.

## 4.4.1 Command Overview

The following table lists all commands from the `hpmcmd` program and shows for which blades/modules they are available. You can display this list and a short command description using the help command (see section [help on page 58](#)). A detailed description of the commands is given in section [Supported Commands on page 49](#).

Table 4-1 Command Overview

Command	Description
<i>bootbankget</i>	Gets the bootbank to boot from
<i>bootbankset</i>	Sets the bootbank to boot from
<i>chinfo</i>	Retrieve channel information
<i>cmd</i>	Execute any IPMI command
<i>deviceid</i>	Gets the Device Id.
<i>frudata</i>	Allows to get FRU info in hex numbers
<i>fruinfoget</i>	Gets string fields from the FRU
<i>fruinfoset</i>	Sets string fields of the FRU
<i>fruinv</i>	Allows to get the FRU size and addressable units
<i>fruread</i>	Allows to read x number of bytes from the FRU
<i>fruwrite</i>	Allows to write x number of bytes from the FRU
<i>help</i>	List of hpmcmd commands
<i>ipmbaddress</i>	Get the IPMB address
<i>ipmcstatus</i>	Get the IPMC status
<i>lancfgget</i>	Get LAN configuration parameter
<i>lancfgset</i>	Set LAN configuration parameter
<i>ledget</i>	Gets the state of a specific FRU LED
<i>ledprop</i>	Get the LED properties for this FRU
<i>ledset</i>	Controls the state of a specific FRU LED
<i>loglevelget</i>	Gets the hpmagentd log level

## Command Overview

---

Table 4-1 Command Overview (continued)

Command	Description
<i>macaddress</i>	Lists the MAC addresses
<i>partnumber</i>	Gets the board part number
<i>physlotnumber</i>	Lists the physical slot location
<i>portget</i>	Shows the current state E-Key governed intfs
<i>portset</i>	Enables/Disables ports in a channel
<i>postypeget</i>	Gets the POST type to run at boot
<i>postypeset</i>	Sets the POST type to run at boot
<i>sdr</i>	Shows the SDR records
<i>sdr_dump</i>	shows the SDR records in binary and hex format
<i>sdrinfo</i>	shows the SDR information
<i>sendamc</i>	Sends any IPMI command to a remote AMC or MMC of a remote IPMC IPMB-L
<i>sendcmd</i>	Sends an IPMI request to the IPMC
<i>shelfaddress</i>	Gets the shelf address string
<i>shelfslots</i>	Prints the number of slots in the shelf
<i>shelftype</i>	Gets the Shelf Type from the Shelf FRU (Board Product Name)
<i>slotmap</i>	Prints the slotmap of the shelf
<i>slotnumber</i>	Shows the board logical slot number
<i>solcfgget</i>	Get SOL configuration parameter
<i>solcfgset</i>	Set SOL configuration parameter
<i>version</i>	Shows the hpmCmd version and the hpmagentd version
<i>watchdog</i>	Control Payload WDT functionality

### 4.4.2 Supported Commands

This section lists the supported commands. All commands are case insensitive. The examples illustrate the use of `hpmcmd` in single-command mode (`-c`). If you start `hpmcmd` without the `-c` or `-i` options (that is, interactive mode), you simply enter these commands at the HPM command prompt.

Some of the `hpm` commands can be sent to a remote IPMC by specifying the `-t` option. This option is not mandatory. If it is not specified, the command is sent to the local IPMC.

The following commands support the `-t` option:

- deviceid
- frudata
- fruinfoget
- fruinv
- fruread
- macaddress
- partnumber

The following sections describe the available commands.

#### 4.4.2.1 bootbankget

##### Description

This command retrieves the boot bank which is currently marked as active for the CPU specified by `payload_cpu_selector`.

Firmware for the CPU is stored in redundant, persistent memory devices. This allows the firmware image in one bank to serve as a backup for the other bank. During normal operation, the CPU on a module determines which bank to boot from based on a GPIO signal controlled by the IPMC. This bank is considered the active boot device.

Because you can change the “active” device with the `hpmcmd bootbankset` command, active status does not necessarily indicate which device was used on the last boot. It simply represents which device is set to be used on the next boot.

##### Synopsis

```
bootbankget <payload_cpu_selector>
```

## Supported Commands

---

### Parameters

payload\_cpu\_selector

Is an integer between 0 and the number of CPU devices supported on the module.

On the PrAMC-7311 the only valid value for payload\_cpu\_selector is 0.

### Example

```
hpmcmd -c bootbankget 0
```

### 4.4.2.2 bootbankset

#### Description

This command sets the boot bank for a particular CPU from which the module is supposed to boot.

#### Synopsis

```
bootbankset <payload_cpu_selector> <newBootBank>
```

#### Parameters

payload\_cpu\_selector

Is an integer between 0 and the number of CPU devices supported on the module.

newBootBank

Can be set to BANK0 or BANK1

#### Example

```
hpmcmd -c bootbankset 0 BANK1
```

### 4.4.2.3 chinfo

#### Description

This command allows you to retrieve the channel information.

#### Synopsis

```
chinfo <channel> [-t ipmb addr [:mmc addr]]
```

#### Parameters

channel - channel number

-t - sends the command to ipmb addr:mmc addr

### Example

```
hpmcmd -c chinfe 0
```

### 4.4.2.4 cmd

#### Description

This command allows you to enter commands understood by the IPMC. Commands are entered as a sequence of hexadecimal numbers as defined in the *IPMI 1.5 Specification*.

#### Synopsis

```
cmd <IPMI command>
```

#### Parameters

IPMI command - Sequence of hex bytes as entered using the ipmicmd

tool from the OpenIPMI library.

```
cmd 0f 00 XX ZZ W1 W2 ... Wn)
```

where:

XX = netfunc as it is in hex

ZZ = cmd number as stated in the IPMI/PICMG spec

W1-Wn = data bytes according to what the command supports

### Example

GetDeviceId command to the local IMPC:

```
hpmcmd -c cmd f 0 6 1
```

GetDeviceId command to the remote IPMC on address 7a:

```
hpmcmd -c cmd 0 7a 0 6 1
```

### 4.4.2.5 deviceid

#### Description

This command retrieves the raw IPMI Get Device ID response and decodes the IPMI message.

#### Synopsis

```
deviceid -t [ipmbAddr[:mmcAddr]]
```

## Supported Commands

---

### Parameters

-t

Sends the command to ipmbAddr:mmcAddr. ipmbAddr is the string lc if it is a local mmcAddr.

### Example

```
hpmcmd -c deviceid
```

### 4.4.2.6 frudata

#### Description

This command dumps the content of the FRU data in hexadecimal format.

#### Synopsis

```
frudata <fruid> [-t ipmbAddr[:mmcAddr]]
```

#### Parameters

fruid

Is 0 for the main module.

-t

Sends the command to ipmbAddr:mmcAddr. ipmbAddr is the string lc if it is a local mmcAddr.

#### Example

```
hpmcmd -c frudata 0
```

```
hpmcmd -c frudata 0 -t 20
```

### 4.4.2.7 fruinfoget

#### Description

This command retrieves information from the specified FRU.

#### Synopsis

```
fruinfoget <fruid> [field] [-v] [-t ipmbAddr[:mmcAddr]]
```

#### Parameters

fruid



0 for the Main Board FRU and other for the RTM or AMC FRU.

field

Is one of the following data fields. If no field is specified, it retrieves the whole fruinfo for that FRU.

Field	Description
bmanufacturer	Board manufacturer
bproductname	Board product name
bserialnumber	Board serial number
bpartnumber	Board part number
pmanufacturer	Product manufacturer
pproductname	Product product name
ppartnumber	Product part number
pversion	Product version number
pserialnumber	Product serial number
passetag	Product inventory asset identifier

-v

Verbose mode to get point-to-point connectivity information where no specific field is requested.

-t

Sends the command to ipmbAddr:mmcAddr . ipmbAddr is the string lc if it is a local mmcAddr.

### Example

```
hpmcmd -c fruinfoget 1 bmanufacturer
```

The following example for fruinfoget is without fields and -v option.

```
hpmcmd -c fruinfoget 0
```

### 4.4.2.8 fruinfoset

#### Description

This command sets some individual field in a FRU, or replaces the whole content of the FRU from a file.

## Supported Commands

---

### Synopsis

```
fruinfoset <fruid> -f <frufilepath> [-t ipmbAddr[:mmcAddr]]
fruinfoset <fruid> [field] <newvalue> [-t ipmbAddr[:mmcAddr]]
```

### Parameters

`fruid`

0 for the Main Board FRU and other for the RTM or AMC FRU.

`newvalue`

It is the new value, less than 16 bytes long, to be set.

`Frufilepath`

It is the full FRU info binary file path, when using a file.

`field`

Is one of the following data fields. If no field is specified, it retrieves the whole fruinfo for that FRU.

Field	Description
bmanufacturer	Board manufacturer
bproductname	Board product name
bserialnumber	Board serial number
bpartnumber	Board part number
pmanufacturer	Product manufacturer
pproductname	Product product name
ppartnumber	Product part number
pversion	Product version number
pserialnumber	Product serial number
passetag	Product inventory asset identifier

`-t`

Sends the command to ipmbAddr:mmcAddr. ipmbAddr is the string lc if it is a local mmcAddr.

### Example

```
hpmcmd -c fruinfoset 1 bmanufacturer Artesyn
```

### 4.4.2.9 fruinv

#### Description

This command retrieves the FRU size and the addressable unit for the specified FRU.

#### Synopsis

```
fruinv <fruid> [-t ipmbAddr[:mmcAddr]]
```

#### Parameters

fruid

0 for the Main Board FRU and other for the RTM or AMC FRU.

-t

Sends the command to ipmbAddr:mmcAddr. ipmbAddr is the string lc if it is a local mmcAddr.

#### Example

```
hpmcmd -c fruinv 0
```

### 4.4.2.10 fruread

#### Description

This command gets nBytes of FRU fruid from the startAddress in hex notation.

#### Synopsis

```
fruread <fruid> <startAddress> <nBytes> [-t ipmbAddr [:mmcAddr]]
```

#### Parameters

fruid

0 for the Main Board FRU and other for the RTM or AMC FRU.

startAddress

The starting point from where to read.

nbytes

Number of bytes to read in decimal; cannot exceed 16 because of IPMI message size limitations.

-t

Sends the command to ipmbAddr:mmcAddr.

## Supported Commands

---

### Example

```
hpmcmd -c fruread 0 0 10
hpmcmd -c fruread 0 0 10 -t 20
```

### 4.4.2.11 fruwrite

#### Description

This command writes hex byte values to FRU fruid starting at the startAddr in hex notation.

#### Synopsis

```
fruwrite <fruid> <startAddress> <hexval1> [hexval2] [...]
[hexval16] [-t ipmbAddr [:mmcAddr]]
```

#### Parameters

fruid

0 for the main board FRU and other for the RTM or AMC FRU.

startAddress

The starting point from where to write.

hexvalN

Hex byte value to write.

-t

Sends the command to ipmbAddr:mmcAddr.

### 4.4.2.12 help

#### Description

This command lists the available commands from the hpmcmd program with a brief explanation about the command.

#### Synopsis

```
help
```

### 4.4.2.13 ipmbaddress

#### Description

This command retrieves the module IPMB address.

#### Synopsis

```
ipmbaddress
```

#### Example

```
hpmcmd -c ipmbaddress
```

### 4.4.2.14 ipmcstatus

#### Description

This command retrieves the module ipmc status.

#### Synopsis

```
ipmcstatus
```

#### Example

```
hpmcmd -c ipmcstatus
```

### 4.4.2.15 lancfgget

#### Description

This command gets LAN configuration parameter.

#### Synopsis

```
lancfgget <channel> [param] [-t ipmbAddr[:mmcAddr]]
```

#### Parameters

channel

channel number

param

auth-type-support

auth-type-enables

## Supported Commands

---

ip-addr  
ip-addr-src  
mac-addr  
subnet-mask  
ipv4-header-params  
primary-rmcp-port  
secondary-rmcp-port  
bmc-generated-arp-control  
gratuidous-arp-interval  
default-gateway-addr  
default-geteway-mac-addr  
backup-gateway-addr  
backup-geteway-mac-addr  
community-string  
num-destinations  
destination-type  
destination-addr  
vlan-id  
vlan-prio  
rmcp-cipher-support  
rmcp-ciphers  
rmcp-priv-levels  
dst-addr-vlan-tags

-t  
sends the command to ipmbAddr:mmcAddr.

### Example

```
hpmcmd -c lanCfgget 1
```

### 4.4.2.16 `lancfgset`

#### Description

This command sets LAN configuration parameter.

#### Synopsis

```
lancfgset <channel> <param> <value> [-t ipmb_addr[:mmc_addr]]
```

#### Parameters

channel

channel number

param

ip-addr

subnet-mask

default-gateway-addr

value

IP address in string format.

#### Example

```
hpmcmd -c lancfgset 1 ip-addr 172.16.0.222
```

### 4.4.2.17 `ledget`

#### Description

This command gets information about a specified LED controlled by the IPMC.

#### Synopsis

```
ledget <fruid> <led> [-t ipmbAddr[:mmcAddr]]
```

#### Parameters

fruid

FRUID.

led

BLUE (for the hotswap LED)

LEDN (for FRU led N, n is a number between 1 and the maximum number of FRULEDs supported by the board)

## Supported Commands

---

-t

Sends the command to ipmbAddr:mmcAddr. ipmbAddr is the string lc if it is a local mmcAddr.

### Example

```
hpmcmd -c ledget 0 BLUE
hpmcmd -c ledget 0 led1
```

### 4.4.2.18 ledprop

#### Description

This command lists the LEDs in this FRU controlled by the IPMC.

#### Synopsis

```
ledprop <fruid>
```

#### Parameters

fruid

0 for the Main Board FRU and 1 for the RTM FRU.

#### Example

```
hpmcmd -c ledprop 0
```

### 4.4.2.19 ledset

#### Description

This command controls the override state of a specific FRU LED. The RTM FRU LEDs reflect the state of the main blade (FRU 0) LEDs. Therefore, overriding the state to something different than the main FRU LED state will not have any effect.

The blue LED is the only one that can be controlled separately.

#### Synopsis

```
lledset <fruid> <led> OFF [-t ipmb_addr[:mmc_addr]]
ledset <fruid> <led> LOCAL [-t ipmb_addr[:mmc_addr]]
ledset <fruid> <led> ON [color] [-t ipmb_addr[:mmc_addr]]
ledset <fruid> <led> TEST [duration] [-t ipmb_addr[:mmc_addr]]
```



```
ledset <fruid> <led> BLINK [offms] [onms] [color] [-t  
ipmb_addr[:mmc_addr]]
```

### Parameters

fruid

0 for the front board. Other for RTMs or AMCs.

led

BLUE (for the hotswap LED)

LEDN (for FRU led N, n is a number between 0 and the maximum number of FRU LEDs supported by the board)

operation

ON (for turning it on)

OFF (for turning it off)

BLINK (for blinking (dft 300ms/300ms))

LOCAL (for return to local control)

TEST (for running lamptest (dft 5000ms))

offms

Off duration when blinking; Values from 10ms to 2500ms in 10ms increments.

onms

On duration when blinking; Values from 10ms to 2500ms in 10ms increments.

duration

lamptest duration. values from 100ms to 12800ms in increments of 100ms should be used.

color

LED0 = BLUE

LED1 = RED or AMBER

LED2 = GREEN (if supported by IPMC)

LED3 = AMBER (if supported by IPMC)

-t

sends the command to ipmb\_addr:mmc\_addr

### Example

```
hpncmd -c ledset 0 led1 on
```

## Supported Commands

---

### 4.4.2.20 loglevelget

#### Description

This command retrieves the current `hpm` log level.

#### Synopsis

```
loglevelget
```

#### Example

```
hpmcmd -c loglevelget
```

### 4.4.2.21 macaddress

#### Description

This command retrieves a list of available MAC addresses.

#### Synopsis

```
macaddress [-t ipmbAddr]
```

#### Parameters

```
-t ipmbAddr
```

Sends the command to `ipmbAddr`.

#### Example

```
hpmcmd -c macaddress  
hpmcmd -c macaddress -t 20
```

### 4.4.2.22 partnumber

#### Description

This command retrieves the part number of the main module.

#### Synopsis

```
partnumber [-t ipmbAddr]
```

### Parameters

`-t ipmbAddr`

Sends the command to ipmbAddr.

### Example

```
hpmcmd -c partnumber
hpmcmd -c partnumber - t 20
```

### 4.4.2.23 physlotnumber

#### Description

This command retrieves the physical slot number in which the blade is plugged in. It is only applicable to AdvancedTCA system environments.

#### Synopsis

`physlotnumber`

#### Example

```
hpmcmd -c physlotnumber
```

### 4.4.2.24 portget

#### Description

This command shows the current state of interfaces governed by e-keying. If no channel is specified, portget returns data for all channels in the specified interface. If neither interface nor channel are specified, portget will return data for all interfaces.

#### Synopsis

```
portget [interface] [channel]
```

#### Parameters

`interface`

The only valid value for PrAMC-7311 is AMC.

`channel`

Is an integer in the following range:

0 and 1 for Ethernet links for Base

2 and 3 for SATA storage links

4 and 5 for PCI-Express links

## Supported Commands

---

### Example

```
hpmcmd -c portget
hpmcmd -c portget AMC 0
```

### 4.4.2.25 portset

#### Description

This command enables and disables ports in a channel. The following table lists the valid values for each parameter.

#### Synopsis

```
portset <intf> <chan> <grpId> <type> <typeX> <ports> <oper> [devid]
[-t ipmbAddr[:mmcAddr]]
```

#### Parameters

intf

Valid value for the PrAMC-7311 is: AMC

chan

Is an integer in the following range:

0 and 1 for Ethernet links for Base

2 and 3 for SATA storage links

4 and 5 for PCI-Express links

grpId

0 for Ethernet links and SATA storage links

0 or 2 for PCI-Express links

type

Valid values are:

Valid Value	Description
ETHER	for fabric interface
STORAGE	for storage links

ports

0 for Ethernet links

0123 for PCI-Express links

oper

Valid values are DISABLE or ENABLE.

### Example

```
hpmcmd -c portset base 1 0 base 0 0 enable
```

### 4.4.2.26 posttypeget

#### Description

This command retrieves the post\_type to which the board is currently set to run at boot time for the particular CPU specified.

#### Synopsis

```
posttypeget <payload_cpu_selector>
```

#### Parameters

payload\_cpu\_selector

The particular CPU specified is set to posttype to run.

#### Example

```
hpmcmd -c posttypeget 0
```

### 4.4.2.27 posttypeset

#### Description

This command sets the posttype to which the board is currently set to run at boot time for the particular CPU specified.

#### Synopsis

```
posttypeset <payload_cpu_selector> <newPostType>
```

#### Parameters

payload\_cpu\_selector

Integer between 0 and number of CPU devices supported per board.

newPostType

POST type. Supported values are: SHORT and LONG

#### Example

```
hpmcmd -c posttypeset 0 LONG
```

## Supported Commands

---

### 4.4.2.28 sdr

#### Description

This command shows the SDR records.

#### Synopsis

```
sdr
```

#### Example

```
hpmcmd -c sdr
recID 1: management controller device locator record
    I2C slave addr: 3D
    Channel number: 00
    Power state: 06
    Global init: 0C
    Capabilities: 29
    Entity Id: AMC
    Entity instance: 65
    OEM: 00
    Id string: PrAMC-731x

recID 2: full sensor record
    owner is IPMB 7A sensor num 00 on lun 00 channel 00
    logical entity: AMC - instance 65
    Module Hot Swap: module hot swap: sensor-specific discrete

recID 3: full sensor record
    owner is IPMB 84 sensor num 02 on lun 00 channel 00
    logical entity: AMC - instance 65
    MP +3.3V: voltage; threshold

recID 4: full sensor record
    owner is IPMB 7A sensor num 02 on lun 00 channel 00
    logical entity: AMC - instance 65
    +12V: voltage: threshold

recID 5: full sensor record
    owner is IPMB 7A sensor num 03 on lun 00 channel 00
```

```
logical entity: AMC - instance 65
UCD3V3: voltage: threshold
recID 6: full sensor record
owner is IPMB 7A sensor num 04 on lun 00 channel 00
logical entity: AMC - instance 65
VCCCORE: voltage: threshold

recID 7: full sensor record
owner is IPMB 7A sensor num 05 on lun 00 channel 00
logical entity: AMC - instance 65
MMC Health : management subsystem health : sensor-specific
discrete
recID 8: full sensor record
owner is IPMB 7A sensor num 06 on lun 00 channel 00
logical entity: AMC - instance 65
Version Change: 2B: sensor-specific discrete
recID 9: full sensor record
owner is IPMB 7A sensor num 07 on lun 00 channel 00
logical entity: AMC - instance 65
BMC Watchdog : watchdog 2 : sensor-specific discrete
recID 10: full sensor record
owner is IPMB 7A sensor num 08 on lun 00 channel 00
logical entity: AMC - instance 65
Processor : processor : sensor-specific discrete
recID 11: full sensor record
owner is IPMB 7A sensor num 09 on lun 00 channel 00
logical entity: AMC - instance 65
ALLPGOOD : power supply : generic
recID 12: full sensor record
owner is IPMB 7A sensor num 0A on lun 00 channel 00
logical entity: AMC - instance 65
F/W Progress : system firmware progress : sensor-specific
discrete
recID 13: full sensor record
owner is IPMB 7A sensor num 0B on lun 00 channel 00
logical entity: AMC - instance 65
```

## Supported Commands

---

```
PCH Temp : temperature : threshold
recID 14: full sensor record
  owner is IPMB 7A sensor num 0C on lun 00 channel 00
  logical entity: AMC - instance 65
  Inlet Temp : temperature : threshold
recID 15: full sensor record
  owner is IPMB 7A sensor num 0D on lun 00 channel 00
  logical entity: AMC - instance 65
  Power Temp : temperature : threshold
recID 16: full sensor record
  owner is IPMB 7A sensor num 0E on lun 00 channel 00
  logical entity: AMC - instance 65
  Ethernet Temp : temperature : threshold
recID 17: full sensor record
  owner is IPMB 7A sensor num 0F on lun 00 channel 00
  logical entity: AMC - instance 65
  CPU Temp : temperature : threshold
recID 18: full sensor record
  owner is IPMB 7A sensor num 10 on lun 00 channel 00
  logical entity: AMC - instance 65
  Boot Error : boot error : sensor-specific discrete
recID 19: full sensor record
  owner is IPMB 7A sensor num 11 on lun 00 channel 00
  logical entity: AMC - instance 65
  OS Boot : os boot : sensor-specific discrete
recID 20: full sensor record
  owner is IPMB 7A sensor num 12 on lun 00 channel 00
  logical entity: AMC - instance 65
  Boot Bank : D2 : sensor-specific discrete
```

### 4.4.2.29 sdr\_dump

#### Description

This command shows the SDR records in binary and hex format.



### Synopsis

```
sdr_dump
```

### Example

```
hpmcmd -c sdr_dump
SDR Records:
01 00 51 01 39 20 00 10 14 61 7f 69 02 01 04 22 "..Q.9 ...a.i..."
04 22 12 12 00 04 00 00 33 00 00 00 00 c0 07 cd ".".....3.....î"
d0 ca ff 00 00 d8 00 00 c2 00 01 01 00 00 00 ce ".....î"
53 42 43 20 2b 31 2e 30 35 56 20 56 74 74 "SBC +1.05V Vtt"
.
.
.
61 67 65 20 45 "age E"
```

### 4.4.2.30 sdrinfo

#### Description

This command shows the SDR information.

#### Synopsis

```
sdrinfo
```

#### Example

```
hpmcmd -c sdrinfo
SDR Information:
LUN 0 has 066 sensors; static sensor population
LUN 1 has 000 sensors; static sensor population
LUN 2 has 000 sensors; static sensor population
LUN 3 has 000 sensors; static sensor population
```

### 4.4.2.31 sendamc

#### Description

This command allows sending any of the commands supported in the IPMI spec to a remote AMC or MMC of a remote IPMC IPMB-L.

## Supported Commands

---

### Synopsis

```
sendamc <IPMAddress> <MMCAaddress> <netfn> <cmd> <data0> ...  
<datan>
```

### Parameters

IPMAddress

Destination IPMB address in hex digits.

MMCAaddress

Destination MMC address in hex digits.

netfn

IPMI request net function in hex digits.

cmd

IPMI request command in hex digits.

data0-datan

IPMI request data bytes. if any, in hex digits.

### 4.4.2.32 sendcmd

#### Description

This command allows a user to send any of the commands supported in the IPMI spec to a remote IPMC.

#### Synopsis

```
sendcmd <IPMAddress> <netfn> <cmd> <data0> ... <dataN>
```

#### Parameters

IPMAddress

Destination IPMB address in hex digits.

netfn

IPMI request net function in hex digits.

cmd

IPMI request command in hex digits

data0 ... dataN

IPMI request data bytes. if any, in hex digits.

### Example

```
hpmcmd -c sendcmd 7a 0a 11 00 00 00 10
Completion code: 0x00 (0) Success
Response data : 10 01 00 00 01 0A 13 00 E1 01 09 19 E1 07 88 C7 45
hpmcmd -c sendcmd 20 0a 11 00 00 00 10
Completion code: 0x00 (0) Success
Response data : 10 01 00 00 01 09 11 00 E4 01 08 19 B9 8F 87 C7 45
```

### 4.4.2.33 shelfaddress

#### Description

This command retrieves the shelf address string from the shelf FRU.

#### Synopsis

```
shelfaddress
```

#### Example

```
hpmcmd -c shelfaddress
3
```

### 4.4.2.34 shelfslots

#### Description

This command retrieves the number of slots in the shelf.

#### Synopsis

```
shelfslots
```

#### Example

```
hpmcmd -c shelfslots
14 slots
```

### 4.4.2.35 shelftype

#### Description

This command retrieves the shelf type from the shelf FRU (Board Product Name)

## Supported Commands

---

### Synopsis

shelftype

### Example

```
hpmcmd -c shelftype
AXP-1440
```

### 4.4.2.36 slotmap

#### Description

This command retrieves the slotmap of the shelf.

#### Synopsis

slotmap

#### Example

```
hpmcmd -c slotmap
```

```
-----
-
Physical Slot : 01 02 03 04 . 05 06 07 08 . 09 10 11 12 . 13 14
Logical Slot  : 13 11 09 07 . 05 01 03 04 . 02 06 08 10 . 12 14
IPMB Address  : 9A 96 92 8E . 8A 82 86 88 . 84 8C 90 94 . 98 9C
-----
-
```

### 4.4.2.37 slotnumber

#### Description

This command retrieves the logical slot number of the slot where the module is plugged in.

#### Synopsis

slotnumber

#### Example

```
hpmcmd -c slotnumber
2
```

### 4.4.2.38 solcfgget

#### Description

This command retrieves the SOL configuration parameter.

#### Synopsis

```
solcfgget <channel> [param] [-t ipmbAddr[:mmcAddr]]
```

#### Parameters

channel

channel number

param

enable

authentication

char-settings

retry

nonvolatile-bit-rate

volatile-bit-rate

payload-channel

payload-port

-t

sends the command to ipmbAddr:mmcAddr

#### Example

```
hpmcmd -c solcfgget 1 retry
```

```
Retry Count           : 1
Retry Interval        (ms): 50
```

```
hpmcmd -c solcfgget 0
```

```
Enabled               : true
Force Encryption      : false
Force Authentication   : false
Privilege Level        : User
Character Accumulate Interval (ms): 5
Character Send Threshold : 1
Retry Count           : 1
```

## Supported Commands

---

Retry Interval	(ms): 50
Non-Volatile Bit Rate	(kps): 115.2
Volatile Bit Rate	(kps): 115.2
Payload Port	: 623

### 4.4.2.39 solcfgset

#### Description

This command retrieves the SOL configuration parameter.

#### Synopsis

```
solcfgset <channel> <param> <value> [-t ipmbAddr[:mmcAddr]]
```

#### Parameters

channel

channel number

param

force-encryption true|false

force-authentication true|false

privilege-level user|operator|administrator|oem

char-accumulate-interval 1-1275 (ms)

char-send-threshold 0-255

retry-count 0-255

retry-interval 0-2550 (ms)

non-volatile-bitrate 9.6|19.2|38.4|57.6|115.2

volatile-bitrate 9.6|19.2|38.4|57.6|115.2

port 0-255

-t

sends the command to ipmbAddr:mmcAddr

### 4.4.2.40 version

#### Description

This command retrieves the version of the hpmcmd software and sends a request to get the version of the hpmagent daemon that is running. Once the information is gathered, it is printed.

**Synopsis**

```
version
```

**Example**

```
hpmcmd -c version
3.10.2
```

**4.4.2.41 watchdog****Description**

This command is used handle the payload BMC watchdog.

**Synopsis**

```
watchdog set <tmr_use> <tmr_action> <pre_timeout> <flags>
<lsb_val> <msb_val>
watchdog set default
watchdog get
watchdog start
watchdog stop
watchdog reset
```

**Parameters**

```
set
```

Name	Possible values
tmr_use	dont_stop stop
tmr_action	no_action hard_reset power_cycle power_down
pre_timeout	0-255
flags	clear dont_clear
lsb_val	0-255
msb_val	0-255

## Supported Commands

---

Name	Possible values
user	

### Example

```
hpmcmd -c watchdog get
hpmcmd -c watchdog set stop no_action 5 clear 100 1
hpmcmd -c watchdog stop
hpmcmd -c watchdog start
```



# HPI-B Software

To help ease the implementation of highly available systems with off-the-shelf building blocks, the Service Availability Forum (SA Forum) Hardware Platform Interface (HPI) specification HPIB defines a set of platform-independent programming interfaces to monitor and control systems such as AdvancedTCA systems, designed to provide high availability. HPI provides applications and middleware a consistent, standardized interface for managing hardware components.

## 5.1 Package Information

This BBS release contains the following HPI-B packages.

*Table 5-1 List of HPI-B packages*

File	Description
bbs-hpib_2.7.5-1.x86_64-centos7.1-linux.rpm	This package contains the shared libraries to be used by HPI-B clients and some example HPI-B client applications.
bbs-hpib-devel_2.7.5-1.x86_64-centos7.1-linux.rpm	This package contains the HPI-B header files and static libraries used for the development of HPI-B clients. This is part of host development toolkit.
bbs-hpib-clientsrc_2.7.5-1.x86_64-centos7.1-linux.rpm	This package contains client sources as part of devkit for the application. This is part of host development toolkit.

## 5.2 Configuring the HPI-B client

HPI-B client software comprises of the packages listed in [List of HPI-B packages on page 73](#).

It consists of files required to develop your own client applications and also precompiled example applications. The counterpart of the HPI-B clients are the HPI-B daemons which run on the system manager blades (in our case, F125/F140). To configure HPI-B client, refer to *System Management Interface Based on HPI-B 2.0 User's Guide*.



# Related Documentation

## A.1 SMART Embedded Computing Documentation

The documentation listed is referenced in this manual. Technical documentation can be found by using the Documentation Search at <https://www.smartembedded.com/ec/support/> or you can obtain electronic copies of SMART EC documentation by contacting your local sales representative.

*Table A-1 SMART EC Publications*

Document Title	Publication Number
System Management Interface Based on BBS HPI-B 2.0 (Centellis 4620/4440)	6806800P21
PrAMC-7311 Installation and Use	6806800P34
ATCA-F125 Installation and Use	6806800J94
ATCA-F120 Installation and Use	6806800D06
ATCA-F140 Installation and Use	6806800M67

## A.2 Related Specifications

For additional information, refer to the following table for related specifications. As an additional help, a source for the listed document is provided. Please note that, while these sources have been verified, the information is subject to change without notice.

*Table A-2 Related Specifications*

Document Title	Source
IPMI Specifications <a href="http://www.intel.com/design/servers/ipmi">http://www.intel.com/design/servers/ipmi</a>	
IPMI Spec v1.5, Document Revision 1.1, February 20, 2002	Intel Corporation, Hewlett-Packard, DEC, NEC
IPMI v1.5 Addenda, Errata, and Clarifications, Addendum Document Revision 5, January 29, 2004	Intel Corporation, Hewlett-Packard, DEC, NEC
Intelligent Platform Management Interface Specification v1.0, Document Revision 1.1, November 15 1999	Intel Corporation, Hewlett-Packard, NEC, Dell
IPMI Implementer's Guide, Draft Version 0.7, September 16, 1998	Intel Corporation

## Additional Resources

---

*Table A-2 Related Specifications (continued)*

<b>Document Title</b>	<b>Source</b>
IPMI Platform Management FRU Information Storage Definition V1.0, September 27, 1999	Intel Corporation
PCI Industrial Computer Manufacturers Group (PICMG) Specifications <a href="http://www.picmg.org">http://www.picmg.org</a>	
PICMG 3.0 Revision 1.0 Advanced Telecommunications Computing Architecture (AdvancedTCA) Base Specification, December 2002	PICMG
PICMG 3.1 Revision 1.0 Specification Ethernet/Fibre Channel for AdvancedTCA Systems, January 2003	PICMG
Service Availability Forum Specifications <a href="http://www.saforum.org">http://www.saforum.org</a>	
SAI-HPI-B.01.01 Hardware Platform Interface Specification	SA Forum
SAI-AIS-A.01.01 Application Interface Specification	SA Forum
SAI-HPI-SNMP-B.01.01	SA Forum
SAIM-HPI-B.01.01-ATCA SAF HPI-to-AdvancedTCA Mapping Specification	SA Forum

## A.3 Additional Resources

The following table lists additional resources which may be useful in working with SMART EC's AdvancedTCA systems.

*Table A-3 Additional Resources*

<b>Resource</b>	<b>Source</b>
OpenHPI open source software project <a href="http://openhpi.org">http://openhpi.org</a>	
OpenHPI 1.0 Manual	OpenHPI
OpenHPI NetSNMP Subagent Development Manual	OpenHPI
Net-SNMP <a href="http://net-snmp.sourceforge.net/">http://net-snmp.sourceforge.net/</a>	
Pigeon Point Systems <a href="http://www.pigeonpoint.com">http://www.pigeonpoint.com</a>	

*Table A-3 Additional Resources (continued)*

<b>Resource</b>	<b>Source</b>
IPM Sentry Shelf-External Interface Reference	Pigeon Point Systems
IPM Sentry Shelf Manager User Guide	Pigeon Point Systems
OpenIPMI <a href="http://openipmi.sourceforge.net/">http://openipmi.sourceforge.net/</a>	
Initramfs <a href="http://en.gentoo-wiki.com/wiki/Initramfs">http://en.gentoo-wiki.com/wiki/Initramfs</a>	

## Additional Resources

---



