COMPUTING

Basic Blade Services Software on ATCA-7490

Programmer's Reference

P/N: 6806800U15B
October 2016

ARTESYN™
EMBEDDED TECHNOLOGIES

Trademarks

Notice

Limited and Restricted Rights Legend

Contact Address

Artesyn Embedded Technologies
Marketing Communications
2900 S. Diablo Way, Suite 190
Tempe, Arizona 85282

Artesyn Embedded Technologies
Lilienthalstr. 17-19
85579 Neubiberg/Munich
Germany

Contents

List of Tables

List of Figures

Overview of Contents

This manual is divided into following chapters and appendices.

- Chapter 1, *Introduction, on page 19* provides a brief description about the BBS functionality.
- Chapter 2, *Installing the Basic Blade Services Software, on page 21* provides the BBS software packages and its installation procedure.
- Chapter 3, *Operating System, on page 25* provides a brief description about the Operating System and its procedures.
- Chapter 4, *Intel Ethernet Switch Software, on page 35* provides a brief description about Intel Ethernet Switch Software.
- Chapter 5, *Firmware Upgrade Facility, on page 55* provides a brief description about Firmware upgrade facility.
- Chapter 6, *Hardware Platform Management, on page 65* provides a brief description about hardware platform management.
- Chapter 7, *HPI-B Software, on page 111* provides the brief description about HPI-B software.
- Chapter 8, *Board Control Module, on page 113* provides the brief description about board control module.
- Chapter 9, *Kernel and Root File System Config using Linux, on page 119* provides the brief description about kernel and root file system configuration using Linux.
- Chapter 10, *RTM-ATCA-749X, on page 121* provides a brief description about RTM-ATCA-749X interfaces and configuration and also provides a brief description about quick assist technology.
- Appendix A, *Unattended Installation with CentOS 7.1, on page 143* describes about the unattended installation file system.
- Appendix B, *I2C Bus Structure on RTM-7490, on page 145* describes I2C Bus Structure on RTM-7490.
- Appendix C, *Related Documentation, on page 147* provides a listing of related product documentation.

Abbreviations

This document uses the following abbreviations:

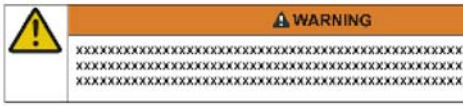
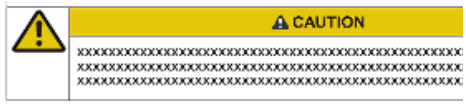
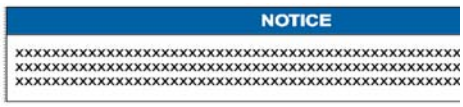
Abbreviation	Definition
ATCA	Advanced Telecommunications Computing Architecture
BBS	Basic Blade Services
BIOS	Basic Input Output System
FCU	FUF Command Line Utility
FPGA	Field Programmable Gate Array
FRI	Firmware Recovery Image
FRU	Field Replaceable Unit
FUF	Firmware Upgrade Facility
GPIO	General Purpose Input/Output
HPI	Hardware Platform Interface
HPM	Hardware Platform Management
IPMB	Intelligent Platform Management Bus
IPMC	Intelligent Platform Management Controller
IPMI	Intelligent Platform Management Interface
IRQ	Interrupt Request
LSP	Linux Support Package
MAC	Media Access Control
OEM	Original Equipment Manufacturer
PCI	Peripheral Component Interconnect
PICMG	PCI Industrial Computers Manufacturers Group
PXE	Preboot Execution Environment
QSFP	Quad Small form-factor Pluggable
RPM	Red Hat Package Manager
RTM	Rear Transition Module
SAS	Serial Attached SCSI

Abbreviation	Definition
SATA	Serial ATA
SDR	Sensor Data Record
SFP	Small form-factor Pluggable
SMI	Serial Management Interface
SNMP	Simple Network Management Protocol
SSD	Solid State Disk
SSH	Secure Shell
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)
bold	Used to emphasize a word
`Screen`	Used for on-screen output and code related elements or commands in body text
`Courier + Bold`	Used to characterize user input and to separate it from system output
Reference	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

Notation	Description
. . .	Omission of information from example/command that is not necessary at the time being
..	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
	Indicates a property damage message
	No danger encountered. Pay attention to important information

Summary of Changes

See the table below for manual revisions and changes.

Part Number	Date	Description
6806800U15B	October 2016	Added Chapter 10, *RTM-ATCA-749X, on page 121* and Appendix B, *I2C Bus Structure on RTM-7490, on page 145*. Updated Chapter 2, *Installing the Basic Blade Services Software, on page 21*, Chapter 4, *Intel Ethernet Switch Software, on page 35*, Appendix A, *Unattended Installation with CentOS 7.1, on page 143*.
6806800U15A	June 2016	Initial version

1.1 Overview

This manual describes the BBS functionality based on CentOS 7.1 Linux. The Basic Blade Services (BBS) software provides a set of services that support the blade on which the software is installed. BBS includes:

- Several custom management functions for the unique hardware provided on the blade.
- A set of management routines for Linux and all hardware interfaces. Management access includes support for Simple Network Management Protocol (SNMP) and a local console interface based on standard Linux command shell.

1.2 Support for other Linux Operating Systems

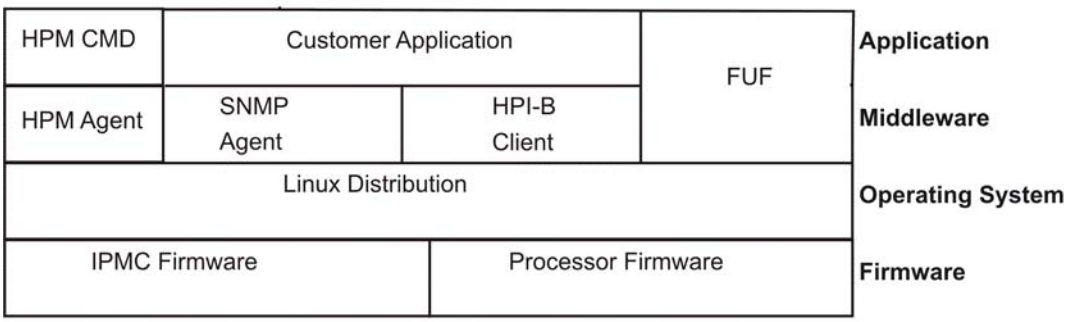
The BBS functionality can be made available for other Linux distributions like Ubuntu on request. Deviations in the functionality will be described in the respective release notes.

1.3 Software Building Blocks

BBS includes a common set of functionality, which is available for all ATCA blades. It also includes a unique set of functionality, which is tailored particularly for this blade.

Figure 1-1 depicts the architecture of the BBS software.

Figure 1-1 BBS Architecture



BBS for the ATCA-7490 consists of following main software and services:

- Firmware Upgrade Facility (FUF): Provides a uniform way to upgrade firmware on Artesyn blades, regardless on which flash locations the firmware is stored. FUF upgrades the BIOS firmware and the IPMC firmware via HPM agent. Currently, FUF consists of Firmware Upgrade Command Line Utility (FCU), flash device drivers, and specially prepared firmware recovery image files.
- Operating System
  CentOS7.1 is the base Operating System for BBS.

NOTICE

The Operating System is not part of the BBS release. You are responsible for installing appropriate Linux OS on the blade.

- Hardware Platform Management
  Hardware Platform Management (HPM) in ATCA systems is based on Intelligent Platform Management Interface (IPMI) specification. IPMI commands can be complex and cumbersome. Using certain set of commands, HPM facilitates the blade and the module-level hardware management.
- HPI-B
  The HPI-B Software packages can be received from your local Artesyn sales representative. For further information, refer to the *System Management Interface Based on HPI-B (Centellis 1440/2000/4440/8440/8840) User's Guide*.

2.1 Overview

The BBS software packages require a pre-installed host Operating System with a kernel 3.10 or later or RHEL 7 like OS.

The best way to perform such host OS installation is using an unattended network installation with a kick start file. For more details, see *Kickstart File for Unattended CentOS7.1 Installation* on page 143.

2.1.1 Package Information

The BBS software is delivered as RPM packages, which can be installed on your target Operating System.

The BBS distribution contains software packages listed in the Table 2-1.

Table 2-1 Software packages

Software	Package Name
Kernel command line	`default.bbs-atca7490`
Checksum of all files and RPMs	`files.sha1sum`
HPM Agent command tool	`bbs-hpmagentcmd-atca7490-<version>-<OS distro>.rpm`
Firmware upgrade utility	`bbs-fcu-atca7490-<version>-<OS distro>.rpm`
Board control utility to get FPGA data	`bbs-boardctrl-atca7490-<version>-<OS distro>.rpm`
BIOS image	`bbs-bios-atca7490-<version>.rpm`
Field-Programmable Gate Array (FPGA) image	`bbs-fpga-atca7490-<version>.rpm`
IPMC Firmware incl Booter (Front board) (should not be used)	`bbs-ipmc-all-atca7490-<version>.rpm`
IPMC Firmware wo Booter (Front board)	`bbs-ipmc-atca7490-<version>.rpm`
IPMC Booter (Front board)	`bbs-ipmc-boot-atca7490-<version>.rpm`
IPMC Firmware (ARTM)	`bbs-artm-atca7360-<version>.rpm`

Table 2-1 Software packages (continued)

Software	Package Name
IPMC Booter Firmware (ARTM)	`bbs-artm-boot-atca7360-<version>.rpm`
IPMC FW for Dual Disk ARTM	`bbs-artm-rtm450n-atca7490-<version>.rpm`
Intel Switch API (IES) with ATCA-7490 specific adaptations	`bbs-ies-atca7490-<version>-.rpm`

The following RPM commands are useful to review package information.

Command	Description
`rpm -qa`	Lists all the installed packages. Use `rpm -qa | grep hpi` to list only HPI packages.
`rpm -ql <package-name>`	Lists the content of a package, where `package-name` is the name of a specific package. For example, `rpm -ql bbs-fcu`.
`rpm -qi <package-name>`	Lists the information about a package, where `package-name` is the name of a specific package. For example, `rpm -qi bbs-fcu`.
`rpm -qf <path to file>`	Finds out to which RPM file belongs to.

2.1.2 Accessing the ATCA-7490 via Serial Console

To invoke the Linux commands or configure BIOS setting, first you need to access the ATCA-7490 via the faceplate serial port. For using a serial console or a terminal emulator, the default serial port settings are:

- 115200 baud
- No parity
- Eight data bits
- One stop bit
- Emulated terminal type: VT100

If you are using the Artesyn provided kickstart file for installation you can access Linux via a Linux shell with the username as root and the password as root. For more information, refer *Login* on page 25.

2.2 BBS Installation

This section describes installing BBS on hosts Operating System, like CentOS7.1. For installing the host OS, refer to the respective release notes provided along with the distribution or by vendor.

Verify that following packages are installed along with the OS.

- pciutils
- OpenIPMI
- ipmitools
- trousers

Installing BBS

1. Copy the BBS package to your target system and run the installation script. This script performs the following:
   - Sets up the udev rules for network device renaming
   - Installs the BBS packages
   - Installs the Linux fm10k driver for the Intel FM10000 (Red Rock Canyon) Ethernet Switch Technology switch device
   - Updates the path variable

NOTICE

The installation script requires root permissions.
During installation of the fm10k driver, the initramfs and the linux modules will be updated if needed.

2.3 Upgrading the BBS Software

BBS software updates are usually delivered as RPM files. You need to follow separate procedures for BBS on hard disk upgrade and diskless client upgrade.

- Upgrading BBS on hard disk
- Upgrading BBS for diskless client

To upgrade the BBS on hard disk

1. Copy the new RPM files to the blade.
2. Stop all BBS related applications. Stop and remove the **boardctrl** kernel driver.
3. Remove the previous files using the `rpm -e <package>` command.
4. Install the newly copied files using the `rpm -Uvh <package-name>` command.
5. Restart the applications and the boardctrl driver again (e.g. using `/etc/init.d/boardctrl restart`)

To upgrade BBS on diskless client

1. Uncompress the ramdisk image
2. Mount the ramdisk image as loop device
3. Update the files in the ramdisk image
4. Unmount the loop device
5. Compress the ramdisk again

3.1 Distribution Description

The Basic Blade services for the ATCA-7490 blade are based on CentOS 7.1 (https://www.centos.org) which is a RHEL7 derivate. The BBS package can be made available for other Operating Systems on request.

3.2 Login

A Linux shell can be accessed via the faceplate serial port.

If you use a serial console or terminal emulator, the serial/RTM port settings are 115200 baud, no parity, 8 data bits, and 1 stop bit.

If you use Secure Shell (SSH) server, it starts in run levels 2–5 and listens on all the Ethernet interfaces. Root login for SSH is not permitted, you need to log in as “admin” user. If you use SSH, refer *Network Services Configuration* on page 27 for default IP address assignments.

If you want to login as `root` via SSH, you need to first configure SSH using the console serial port. Set `PermitRootLogin` to set in the `/etc/ssh/sshd_config` file. To effect the changes, you must either reboot the blade or run the `/etc/init.d/ssh restart` command.

The following table lists available default login accounts.

Login Name	Password	Description
admin	bbsadmin	Non-privileged user account
root	root	Privileged user account

⚠ WARNING

It is recommended to change the passwords after the installation and to disable the root login.

3.3 Linux Services Initialization

Basic blade specific services like IES Switch service are started by the systemd during board startup. Further information on systemd is available at RHEL 7 - System Admin Guide.

The following table gives an overview of which services are installed and started by the BBS. These scripts can be found at `/usr/lib/system/system/` and a symbolic link of it at `/etc/system/system/multi-user.target.wants`.

Table 3-1 Generic Linux Run Levels

Service	Dependencies	Description
`boardctrl.service`	Multi-user.target, after network	Inserts and starts the boardctrl driver.
`osBootSensor.service`	Multi-user.target, after network	Sends IPMI message boot completed to the IPMC a potential running OS boot supervision wathchdog is stopped. This is usually one of the last actions before the linux prompt is reached.
`switchSetup.service`	Multi-user.target, after network	Initialization of the Intel FM10000.
`bbsvlan.service`	Multi-user.target, after network, switchSetup	Setup static IP addresses for the blade dependent on the location in the shelf.
`ethdevordering.service`	Multi-user.target, after bbsvlan	Generates `ifcfg-<interface>` scripts and can also be used to rename the interfaces again.
`hpmagentd.service`	Multi-user.target, after network	This service starts the hpmagent daemon which handles. For example, ipmi callbacks.

3.4 Network Services Configuration

The following sections describe the default configuration for network services.

3.4.1 ATCA-7490 Ethernet Interfaces

The Ethernet devices, such as enp1s0 setup by CentOS7.1 are renamed to more meaningful names in ATCA-7490. This renaming will be performed for the front panel Ethernet devices, the base interface devices and the RTM network devices. For the renaming udev rules (`/etc/udev/rules.d/70-network.rules`) are used. For ARTM-7490, the "eth" network device is renamed using the netdev-renamer tool. The Fabric interfaces will not be renamed.

The following table specifies the Ethernet devices supported by ATCA-7490.

Device Name	Description	Speed	Location	IP address	Driver Name
base1, base2	Base Interface	1 GbE	Front Board -> Backplane	Static IP address. It is computed as:`192.168.<baseIf>.<slot number*10>`baseIf can have value of; '21, 22" slotnumber specifies the logical slot number converted to decimal. The setup of the IP Addresses for Base IF is done in the`/etc/init.d/bbsvlan.sh` file.	Intel - igb
enp5s0, enp7s0, enp129s0, enp131s0	Fabric Interface	KR4 - 40 GbE	Front Board -> Backplane. 4 fabric interfaces are only available in a dual-dual star backplane	Static IP address. It is computed as:`192.168.<fabricIf>.<slot number*10>`fabricIf can have value of; '11, 12, 13, 14" slotnumber specifies the logical slot number converted to decimal. The setup of the IP Addresses for Fabric IF is done in the`/etc/init.d/bbsvlan.sh` file. In the default configuration coming with the BBS enp5s0 is routed to fabric1_1, enp7s0 to fabric1_2, enp129s0 to fabric2_1 and enp131s0 to fabric2_2.	Intel fm10k and IES Switch API
front1, front2	Front Panel Interface	1GbE	Front Board	There will be no IP address assigned by default.	Intel - igb
rtm1..n (optional)	RTM Panel Interface	10GbE, 100Gb	RTM (optional)	There will be no IP address assigned by default.	Intel - fm10k and IES Switch API

Front Board Network devices Setup

Table 3-2 Front Board Network Devices

PCIDev	NIC Device	PCI speed/width	IF Name	Driver
0000:01:00.0	Intel I350	5 GT/s x 4	Base1	igb
0000:01:00.1	Intel I350	5 GT/s x 4	Base2	igb
0000:01:00.2	Intel I350	5 GT/s x 4	Front1	igb
0000:01:00.3	Intel I350	5 GT/s x 4	Front2	igb
0000:03:00.0	Intel FM10000	5 GT/s x 1	enp3s0 - Admin Port of Intel FM10000	fm10k
0000:05:00.0	Intel RRC	8GT/s x 8	enp5s0 - Fabric IF 1 on CPU0	fm10k
0000:05:00.0	Intel RRC	8GT/s x 8	enp7s0 - Fabric IF 2 on CPU0	fm10k
0000:81:00.0	Intel RRC	8GT/s x 8	enp129s0 - Fabric IF 1 on CPU1	fm10k
0000:83:00.0	Intel RRC	8GT/s x 8	enp131s0 - Fabric IF 2 on CPU1	fm10k

The fabric interfaces and the RTM-7490 Interfaces are handled by the Intel FM10000 combined NIC/Switch device. The Ethernet type and speed can be configured.

3.4.2 ATCA-7490 Fabric Interface

The ATCA-7490 blade is equipped with an Intel FM10000and FM10840 combined NIC/switch device. Two fabric interfaces are connected to the first CPU and Two fabric interfaces to the second CPU of the board. This allows 4 x 40Gbs network configurations in an appropriate Quadstar shelf. For example, Artesyn Centellis 8000 series.

3.4.2.1 Network Driver - IRQ Assignment

In order to gain maximum network performance, it is inevitable to assign the interrupts for the network devices to the CPU cores, the devices are physically connected too. The IRQs must also not be routed to the first CPU core.

This IRQ assignment for the network devices is performed by the script:
`/opt/bladeservices/tools/set_irq_affintiy`

3.5 Tools

This section describes `CPUSpeed` and `IPMIBPAR` tools, which can be used to change the processor performance governors and IPMI Boot Parameter list.

3.5.1 Performance Tool - cpuspeed

The performance tool - cpuspeed allows to change the processor performance governors and the core frequency (for userspace governor) on a per core base. It utilizes data stored in the `/sys/device/system/cpu` directory.

The following table describes various governors:

Governor	Description
Performance	Core is running with maximum frequency.
Ondemand	Cores in idle state are running at lowest frequency. When a certain CPU core is used, the frequency of the core is increased up to maximum.
Powersave	Core is running with minimum frequency.
Userspace	Core frequency can be adjusted by the user (in steps).

If the P-States are limited by BIOS, the required driver is not loaded and therefore the CPUSpeed tool cannot work.

CPUSpeed supports the following options.

Option	Description
-d	Dumps CPU Frequency/Governor Info.
-h	Provides help.
-p	Print supported governors.
-s	Sets governor/frequency. It supports the following options: ● -c: Specifies the core. Valid values are 0... <mac CPU cores>. Omitting this option means, all cores. ● -g: Specifies governors, such as performance, powersave, ondemand, and userspace.

Example:

```
root@atca7490:/proc/boardinfo# cpuspeed -s -g performance
```

This command sets the performance governor for all CPU cores.

NOTICE

The CPUspeed tool can be used only when the intel_pstate driver is deactivated in the Linux kernel.

3.5.2 IPMIBPAR

The BIOS stores many of its settings in the IPMI Boot Parameter List. With the `IPMIBPAR` tool, you can change these BIOS settings from the running Operation System. The changes will become active during the next reboot. It supports the following options:

Option	Description
`-d`	Enables debug output.
`-a xx`	IPMB Address, if not present, local IPMC is used.

Option	Description
`-i`	Gets device ID.
`-g`	Gets IPMI Boot Parameter USER area.
`-s file`	Stores IPMI Boot Parameter (USER area), read from file.
`-h`	Provides Help.

The following example describes the steps required to change the BootOrder from Onboard SATA disk to FrontNetwork2.

1. Read the IPMI boot parameter USER area from IPMC.

```
[root@ATCA-7490-6 ~]# ipmibpar -g
ipmibpar - Version 1.02 - IPMI Boot Parameter Demo
Copyright 2015 Artesyn Embedded Technologies
Read System Boot Options from USER area (local IPMC)
Hexdump IPMI Boot Parameter:
Size = 1165 (0x48d)
0000  89 04 72 74 6d 5f 61 75 74 6f 3d 6f 6e 00 72 74
<..rtm_auto=on.rt>
0010  6d 5f 63 70 75 30 5f 62 69 66 3d 78 38 78 38 00
<m_cpu0_bif=x8x8.>
0020  72 74 6d 5f 63 70 75 31 5f 62 69 66 3d 78 38 78
<rtm_cpu1_bif=x8x>
0030  38 00 72 74 6d 5f 63 70 75 30 5f 33 61 3d 61 75
<8.rtm_cpu0_3a=au>
0040  74 6f 00 72 74 6d 5f 63 70 75 30 5f 33 62 3d 61
<to.rtm_cpu0_3b=a>
0050  75 74 6f 00 72 74 6d 5f 63 70 75 30 5f 33 63 3d
<uto.rtm_cpu0_3c=>
IPMI Boot Parameter:
rtm_auto=on
rtm_cpu0_bif=x8x8
rtm_cpu1_bif=x8x8
rtm_cpu0_3a=auto
rtm_cpu0_3b=auto
rtm_cpu0_3c=auto
rtm_cpu0_3d=auto
rtm_cpu1_3a=auto
```

```
rtm_cpu1_3b=auto
rtm_cpu1_3c=auto
….
2. Save the ...
rtm_auto=on
rtm_cpu0_bif=x8x8
rtm_cpu1_bif=x8x8
rtm_cpu0_3a=auto
rtm_cpu0_3b=auto
rtm_cpu0_3c=auto
rtm_cpu0_3d=auto
rtm_cpu1_3a=auto
rtm_cpu1_3b=auto
rtm_cpu1_3c=auto
…..
```

2. Save the received IPMI Boot Parameter list into a file (For example, bootparam.log) and change the boot order as follows.

```
artm_net_boot=off
artm_sas_boot=on
basenet_boot=on
baudrate=115200
boot_order=frontnet2,sataonboard,basenet1,efishell,usbhdd
com_term_type=vt100+
frontnet_boot=on
ipmi_irq=off
os_boot_watchdog=off,5,reset
usb_boot=on
```

3. Write the IPMI parameter list file (For example, bootparam.log).

```
ipmibpar -s <filename>
```

3.5.3 CoreTemp

At `/opt/bladeservices/tools` you can find a script named, `coreTempDump`, which reads and displays the CPU temperature on a per core base. It also shows whether the temperature exceeds the maximum allowed core temperature (indication for CPU throttling).

Syntax: `coreTempDump.sh`

For example,

```
root@ATCA7490:~# coreTempDump.sh
=======================================================
===                                                   ===
===               CoreTemperature                     ===
===                                                   ===
=======================================================
  Nr of Cores:          72
  Nr of Physical Cores: 2
--- Core Temperature Levels ---------------------
      Max Temp.      :  91 (degree)
      Critical Temp. :  101 (degree)
 core  | temp  | status
------------------------
    0    |     43    |   OK
    1    |     38    |   OK
    2    |     38    |   OK
    3    |     37    |   OK
    4    |     37    |   OK
......
```

The Intel FM10000 (Red Rock Canyon) Ethernet Switch Technology device is a combined NIC/Switch device. The switching functionality can be managed with the Intel Ethernet Switch API.

4.1 ATCA-7490 Specific Intel FM 10000 Device Configuration

The board infrastructure is reflected in the configuration file in `/etc/ies/config/ltrr/fm_platform_attributes.cfg`. It defines:

- Port Mapping
- The Interface type settings
- The Ethernet modes of the ports
- Port capabilities, Serdes specific configurations and
- The RTM I2c Multiplexer, I2c IO Expander and Retimer settings

The default configuration sets up 4 x 40Gbs interfaces for the four fabric interfaces of the board. Additionally, the update channel interface is pre-configured as 40Gbs, the 2 x 100 Gbs and 6 x 10 Gbs interfaces on the ARTM-7490 are defined.

The Intel FM10000 (Red Rock Canyon) Ethernet Switch Technology configuration for ATCA-7490 has total 48 ports, In that, 36 of them are network connect to host driver using uio0 (derived from netdev enp3s0 sw 0) .

```
00/PEP8      :   UP   8G   1 FM_PORT_MODE_UP|FM_PORT_STATE_UP
01/Fab1_1    :   UP  40G  11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN '
VLAN 11
02/Fab1_1    : DOWN   0G  11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
03/Fab1_1    : DOWN   0G  11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
04/Fab1_1    : DOWN   0G  11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
05/Fab2_1    :   UP  40G  12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN '
VLAN 12
06/Fab2_1    : DOWN   0G  12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
07/Fab2_1    : DOWN   0G  12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
08/Fab2_1    : DOWN   0G  12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
```

```
09/Fab1_2     :   UP   40G   13 FM_PORT_MODE_UP|FM_PORT_STATE_UP    '
VLAN 13
10/Fab1_2     : DOWN    0G   13 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
11/Fab1_2     : DOWN    0G   13 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
12/Fab1_2     : DOWN    0G   13 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
13/Fab2_2     :   UP   40G   14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN '
VLAN 14
14/Fab2_2     : DOWN    0G   14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
15/Fab2_2     : DOWN    0G   14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
16/Fab2_2     : DOWN    0G   14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
17/Update_1   : DOWN    0G    1
FM_PORT_MODE_ADMIN_PWRDOWN|FM_PORT_STATE_DOWN
..............
40/PEP16      :   UP   64G   11
41/PEP16      : DOWN    0G   11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
42/PEP16      :   UP   64G   13
43/PEP16      : DOWN    0G   13 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
44/PEP16      :   UP   64G   12
45/PEP16      : DOWN    0G   12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
46/PEP16      :   UP   64G   14
47/PEP16      : DOWN    0G   14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
Result: Int32: 0 = 0 = 0x00000000 ports.
```

4.1.1 Intel FM10000 Initialization

The Intel FM 10000 device will be started as a service during OS startup either as `/etc/init.d/switchSetup.sh start` or `systemctl restart switchSetup.service`. During the initialization phase of the Intel FM10000 (Red Rock Canyon) Ethernet Switch Technology device the base configuration will be loaded. The configuration file is located at `/opt/bladeservices/etc/fmShell_master.scr`. It contains the basic port and vlan setup.

```
root@ATCA7490:~# /etc/init.d/switchSetup.sh start
/etc/ies/config/ltrr/fm_platform_attributes.cfg
Found netdev enp3s0(derived from mgmt pep 8 sw 0)
Connect to host driver using[ 1111.279770] device enp3s0 entered
promiscuous moeuio0 (derived from netdev enp3s0 sw 0)

...
```

```
fm10k 0000:05:00.0 enp5s0: NIC Link is up
IPv6: ADDRCONF(NETDEV_CHANGE): enp7s0.13: link becomes ready
IPv6: ADDRCONF(NETDEV_CHANGE): enp3s0: link becomes ready
IPv6: ADDRCONF(NETDEV_CHANGE): enp129s0: link becomes ready
fm10k 0000:83:00.0 enp131s0: NIC Link is up
IPv6: ADDRCONF(NETDEV_CHANGE): enp129s0.12: link becomes ready
IPv6: ADDRCONF(NETDEV_CHANGE): enp131s0: link becomes ready
fm10k 0000:07:00.0 enp7s0: Reset interface
fm10k 0000:07:00.0 enp7s0: NIC Link is up
IPv6: ADDRCONF(NETDEV_CHANGE): enp131s0.14: link becomes ready
fm10k 0000:03:00.0 enp3s0: Reset interface
Setting up default VLAN 11, enabling ports (1..4) and (40,41)
Setting up default VLAN 12, enabling ports (5..8) and (44,45)
Setting up default VLAN 13, enabling ports (9..12) and (42,43)
Setting up default VLAN 14, enabling ports (13..16) and (46,47)
….
fm10k 0000:05:00.0 enp5s0: NIC Link is up
fm10k 0000:07:00.0 enp7s0: NIC Link is up
fm10k 0000:81:00.0 enp129s0: Reset interface
fm10k 0000:81:00.0 enp129s0: NIC Link is up
fm10k 0000:03:00.0 enp3s0: NIC Link is up
fm10k 0000:83:00.0 enp131s0: NIC Link is down
fm10k 0000:83:00.0 enp131s0: NIC Link is up
```

The Intel FM10000 (Red Rock Canyon) Ethernet Switch Technology switch status can be checked using `/etc/init.d/switchSetup.sh status`

4.1.2 Default configuration

The Network interfaces of the Red Rock Canyon device are as follows:

Table 4-1 Default Configuration

Network Interface	Equivalent BackPlane IF
enp5s0.11	First fabric interface connected to CPU0
enp7s0.13	Second fabric interface connected to CPU0
enp129s0.12	First fabric interface connected to CPU1

Table 4-1 Default Configuration (continued)

Network Interface	Equivalent BackPlane IF
enp131s0.14	Second fabric interface connected to CPU1
enp7s0.19	Update channel to partner board

Note: IP addresses are set up by a BBS startup script.

For example, the configuration (`/opt/bladeservices/etc/fmShell_master.scr`) is loaded from the IES Switch Software during Switch Startup. The commands are as follows:

```
/printf ("Setting up default VLAN 11, enabling ports (1..4
(fabric1_1)) and (40,41)\n")
/fmCreateVlan (0,11)
/fmAddVlanPort(0,11,(40,41,1,2,3,4),0)
/fmSetVlanPortState (0,11,(40,41,1,2,3,4),FM_STP_STATE_FORWARDING)
/defaultVlan=11
/fmSetPortAttribute(0,(40,41,1,2,3,4),FM_PORT_DEF_VLAN,&defaultVla
n)
/fmSetPortState (0,(40,41,1,2,3,4),FM_PORT_STATE_UP,0)
/printf ("Setting up default VLAN 12, enabling ports (5..8
(fabic2_1)) and (44,45)\n")
/fmCreateVlan (0,12)
/fmAddVlanPort(0,12,(44,45,5,6,7,8),0)
/fmSetVlanPortState (0,12,(44,45,5,6,7,8),FM_STP_STATE_FORWARDING)
/defaultVlan=12
/fmSetPortAttribute
(0,(44,45,5,6,7,8),FM_PORT_DEF_VLAN,&defaultVlan)
/fmSetPortState (0,(44,45,5,6,7,8),FM_PORT_STATE_UP,0)
/printf ("Setting up default VLAN 13, enabling ports (9..12
(fabric1_2)) and (42,43)\n")
/fmCreateVlan (0,13)
/fmAddVlanPort(0,13,(42,43,9,10,11,12),0)
/fmSetVlanPortState
(0,13,(42,43,9,10,11,12),FM_STP_STATE_FORWARDING)
/defaultVlan=13
/fmSetPortAttribute(0,(42,43,9,10,11,12),FM_PORT_DEF_VLAN,&default
Vlan)
/fmSetPortState (0,(42,43,9,10,11,12),FM_PORT_STATE_UP,0)
```

```
/printf ("Setting up default VLAN 14, enabling ports (13..16
(fabric2_2)) and (33,46,47)\n")
/fmCreateVlan (0,14)
/fmAddVlanPort(0,14,(46,47,33,13,14,15,16),0)
/fmSetVlanPortState
(0,14,(46,47,33,13,14,15,16),FM_STP_STATE_FORWARDING)
/defaultVlan=14
/fmSetPortAttribute(0,(46,47,33,13,14,15,16),FM_PORT_DEF_VLAN,&def
aultVlan)
/fmSetPortState (0,(46,47,33,13,14,15,16),FM_PORT_STATE_UP,0)
/printf ("Setting up VLAN 19 for Update Channel, enabling ports
(17,18,19,20 on (fabric1_1)) \n")
/fmCreateVlan (0,19)
/fmAddVlanPort(0,19,(40,41,17,18,19,20),0)
/fmSetVlanPortState
(0,19,(40,41,17,18,19,20),FM_STP_STATE_FORWARDING)
/defaultVlan=19
/fmSetPortAttribute
(0,(17,18,19,20),FM_PORT_DEF_VLAN,&defaultVlan)
/fmSetPortState (0,(40,41,17,18,19,20),FM_PORT_STATE_UP,0)
A flat setup (e.g. without using VLANs) could look like the
following:
/printf ("Setting up default VLAN 1, enabling all ports\n");
/fmCreateVlan (0,1)
/fmAddVlanPort (0,1,(0..48),0)
/fmSetVlanPortState (0,1,(0..48),FM_STP_STATE_FORWARDING)
/defaultVlan=1
/fmSetPortAttribute (0,(0..48),FM_PORT_DEF_VLAN,&defaultVlan)
/fmSetPortState (0,(0..48),FM_PORT_STATE_UP,1)
/ena=1
/fmSetPortAttribute (0,(0..48),FM_PORT_LEARNING,&ena)
```

4.1.3 Red Rock Canyon - Software

4.1.3.1 Firmware

The Red Rock Canyon loads its base configuration (e.g. PCIe setup) from a SPI flash during powerup. Usually, this device is no subject to be changed. However, the SPI flash can be updated if needed by Intel specific tools. In such cases, consult your local Artesyn FAE representative.

The version of the NVM image can be retrieved by the following `fmShell` command.

```
[root@ATCA-7490-8]# fmShell -Ls -- -e "fmDbgDumpBsmScratch 0,10"
Connect to host driver using /dev/uio0
Versions
==================
EEPROM VERSION          [401]      : 0x00020221 (02.21)
CHIP_VERSION                       : B0
```

4.1.3.2 Driver

The used driver version can be retrieved by the following command:
`ethtool -i <RRC Admin Port>`.

Example,

```
root@ATCA-7490-8]# ethtool -i enp3s0
driver: fm10k
version: 0.19.6
firmware-version:
bus-info: 0000:03:00.0
```

The latest Intel driver for the Red Rock Canyon device can be downloaded from: https://sourceforge.net/projects/e1000/files/fm10k%20stable/. It can be downloaded and must be compiled and installed on the target system. After installation the driver should be available at `/lib/modules/`uname -r`/kernel/drivers/net/Ethernet/intel/fm10k/fm10k.ko`.

4.1.3.3 Intel Ethernet Switch Software

The IES Switch Software as provided in the BBS is currently based on the Intel IES-Release 4.1.7. The IES Software released in the BBS of the ATCA-7490 blade comes with blade specific configurations, initialization scripts and additional adaptations to support the ATCA-7490 and RTM-7490 design.

4.1.4 Red Rock Canyon Administration

Configure the Red Rock Canyon switch device or you can either use the Intel TestPoint Software or the fmShell.

4.1.4.1 fmShell

`fmShell` is a C-Shell which can run Red Rock Canyon switch commands either

- in command line mode
- in interactive shell mode

In order to use `fmShell` you need to start the switch API software. For example, using `systemctl start switchSetup.service` or `/etc/init.d/switchSetup.sh`

On ATCA-7490, there is only one switch device supported therefore, replace the string `<switch>` with "0". For more details, see examples below.

4.1.4.1.1 Interactive Mode

In order to run the switch commands in the interactive shell mode, you need to start `fmClient`. You are now in the `fmShell`, where you can perform your tasks. With "`Strg-C`" you can exit the shell again.

```
[root@ATCA-7490-8 ~]# fmClient
fmShell>
fmShell> help "tool"
-Gives an overview of supported commands
```

Initialization

Command	Description
`switchInit`	Initialize IES or switch access
`fmSetSwitchState <switch>,<enable>`	fmSetSwitchState 0,1 Enables switch #0

Port Status

Command	Description
`ps -1`	Show states of all ports
`ps <port>`	Show states of selected port
`fmDbgDumpPort <switch>,<port>`	Show state of specific port
`fmSetPortState <switch>,<port>,FM_PORT_STATE_UP,0`	Enable specified port
`fmSetPortState <switch>,<port>,FM_PORT_STATE_DOWN,0`	Disable specified port
`fmSetPortState <switch>,<port>,FM_PORT_STATE_ADMIN_PWR DOWN,0`	Power down port (disable transmitter)
`fmDbgDumpPortAttributes <switch>,<port>`	Show all port attributes
`fmSetPortAttribute <switch>,<port>,FM_PORT_xxx,&attribute`	Set port attribute (refer to IES doc)
`fmDbgDumpPortStateTransitions <switch>,<port>`	Show history of port's state stransitions

Example,

```
fmShell> ps -1
Port   Name   State Speed VLAN PORT_MODE, PORT_STATE
0/    PEP8:   UP   8G   1 FM_PORT_MODE_UP|FM_PORT_STATE_UP
01/    Fab1_1:   UP  40G  11 FM_PORT_MODE_UP|FM_PORT_STATE_UP
02/    Fab1_1: DOWN   0G  11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
03/    Fab1_1: DOWN   0G  11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
04/    Fab1_1: DOWN   0G  11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
```

```
05/    Fab2_1:   UP  40G  12 FM_PORT_MODE_UP|FM_PORT_STATE_UP
06/    Fab2_1: DOWN   0G  12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
07/    Fab2_1: DOWN   0G  12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
08/    Fab2_1: DOWN   0G  12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
09/    Fab1_2:   UP  40G  13 FM_PORT_MODE_UP|FM_PORT_STATE_UP
10/    Fab1_2: DOWN   0G  13 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
11/    Fab1_2: DOWN   0G  13 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
12/    Fab1_2: DOWN   0G  13 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
13/    Fab2_2:   UP  40G  14 FM_PORT_MODE_UP|FM_PORT_STATE_UP
14/    Fab2_2: DOWN   0G  14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
15/    Fab2_2: DOWN   0G  14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
16/    Fab2_2: DOWN   0G  14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
17/    Update_1: DOWN   0G  19 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
18/    Update_2: DOWN   0G  19 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
19/    Update_3: DOWN   0G  19 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
20/    Update_4: DOWN   0G  19 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
21/      SFP6:   UP  10G  78 FM_PORT_MODE_UP|FM_PORT_STATE_UP
22/      SFP5:   UP  10G  77 FM_PORT_MODE_UP|FM_PORT_STATE_UP
23/      SFP4:   UP  10G  76 FM_PORT_MODE_UP|FM_PORT_STATE_UP
24/      SFP3: DOWN  10G  75 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
25/      SFP2: DOWN  10G  74 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
26/      SFP1: DOWN  10G  73 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
27/      NC: DOWN   0G   1
FM_PORT_MODE_ADMIN_PWRDOWN|FM_PORT_STATE_DOWN
28/      NC: DOWN   0G   1
FM_PORT_MODE_ADMIN_PWRDOWN|FM_PORT_STATE_DOWN
29/  QSFP28_2: DOWN 100G  32 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
30/  QSFP28_2: DOWN   0G  32 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
31/  QSFP28_2: DOWN   0G  32 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
32/  QSFP28_2: DOWN   0G  32 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
33/  QSFP28_1: DOWN 100G  31 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
34/  QSFP28_1: DOWN   0G  31 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
35/  QSFP28_1: DOWN   0G  31 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
36/  QSFP28_1: DOWN   0G  31 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
37/     TE1:   UP   0G   1 ?|FM_PORT_STATE_UP
38/     TE2:   UP   0G   1 ?|FM_PORT_STATE_UP
39/     FIBM: DOWN   0G   1 ?|FM_PORT_STATE_UP
40/     PEP16:   UP  64G  11
```

```
41/     PEP16: DOWN   0G  11 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
42/     PEP16:   UP  64G  13
43/     PEP16: DOWN   0G  13 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
44/     PEP16:   UP  64G  12
45/     PEP16: DOWN   0G  12 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
46/     PEP16:   UP  64G  14
47/     PEP16: DOWN   0G  14 FM_PORT_MODE_UP|FM_PORT_STATE_DOWN
```

Example,

```
fmShell> ps 1
Port   Name   State Speed VLAN PORT_MODE, PORT_STATE
01/    Fab1_1:   UP  40G  11 FM_PORT_MODE_UP|FM_PORT_STATE_UP
```

Example,

```
fmShell> fmDbgDumpPort 0,1
State for port 1:
mode                   : 0
submode                : 0
speed                  : 40000
link                   : 1
lastSourceMask         : 0000ffff ffffffff
dropTagged             : FALSE
defaultVID             : 11
linkStateChangePending : FALSE
pendingLinkStateValue  : 00000000
security               : 0
learning               : TRUE
securityTrap           : 0
portSecurityEnabled    : 0
fmShell> fmDbgDumpPortAttributes 0,1
Attributes                  : Original value  Current value
LAG value
-------------------------------------------------------------------
--------------------
```

```
FM_PORT_AUTONEG                : N/A          3                    Not
in lag
FM_PORT_AUTONEG_BASEPAGE        : N/A          0x1004001
Not in lag

FM_PORT_AUTONEG_PARTNER_BASEPAGE  : N/A         0x15ec041
Not in lag
FM_PORT_AUTONEG_NEXTPAGES        : N/A          0
Not in lag
FM_PORT_AUTONEG_PARTNER_NEXTPAGES : N/A          2
Not in lag
FM_PORT_AUTONEG_25G_NEXTPAGE_OUI  : N/A         6976381
Not in lag
FM_PORT_BCAST_FLOODING          : N/A          FALSE
Not in lag
FM_PORT_BCAST_PRUNING           : N/A          FALSE
Not in lag
...
```

Counters

Command	Description
`bdPortCountShow port[,all[,filter]]`	Show (changed or all) counters of port (-1 for all), optionally filter counter mode using substring match (For example, filter= "E" for all error counters).

Example,

```
fmShell> bdPortCountShow 1

ps->maxPorts: 48 FM10000_NUM_PORTS: 48

1/Fab1_1: RxUcstPkts            :  < +348                    348
1/Fab1_1: RxUcstPktsNonIP        :  < +348                    348
1/Fab1_1: Rx64Pkts              :  < +10                      10
1/Fab1_1: Rx65to127Pkts          :  < +338                    338
1/Fab1_1: Rx64octets            :  < +640                    640
1/Fab1_1: Rx65to127octets        :  < +34,476              34,476
1/Fab1_1: RxOctetsNonIp          :  < +35,116              35,116
```

```
1/Fab1_1: RxUcstOctetsNonIP     :  < +35,116                  35,116
1/Fab1_1: RxGoodOctets          :  < +35,116                  35,116
1/Fab1_1: RxPriorityPkts[0]     :  < +348                        348
1/Fab1_1: RxPriorityOctets[0]   :  < +35,116
35,116

1/Fab1_1: FIDForwardedPkts      :  > +348                        348
1/Fab1_1: FIDForwardedOctets    :  = +35,116                  35,116
1/Fab1_1: TxUcstPkts            :  > +347                        347
```

VLANS

Command	Description
`fmDbgDumpVid <switch>`	Dump VLANs.
`fmCreateVlan <switch>,<vlan>`	Create new VLAN.
`fmAddVlanPort <switch>,<vlan>,<port>,<tagged>`	Add port to vlan (tagged=1 or 0).
`fmSetVlanPortState <switch>,<vlan>,<port>, FM_STP_STATE_FORWARDING`	Enable packet forwarding for vlan/port.
`fmSetPortAttribute <switch>,<port>,FM_PORT_DEF_VLAN,&vlan`	Assign default vlan to port (vlan variable needs to be declared and assigned).
`fmSetVlanAttribute <switch>,<port>,FM_VLAN_REFLECT,&val`	SetVLANattribute(here:enablereflection if val==1).

Example,

```
fmShell> fmDbgDumpVid 0

Vid#            Entry Reflect   Vcnt   FID1 FID2_IVL   FID2   IGMP
ARP Membership/Tagging
11 000000003600000001e       1      0      0      255     -1 notrap
notrap   1   2   3   4  37  38  40  41
12 00000003060000001e0       1      0      0      255     -1 notrap
notrap   5   6   7   8  37  38  44  45
13 00000000c6000001e00       1      0      0      255     -1 notrap
notrap   9  10  11  12  37  38  42  43
14 0000000c0620001e000       1      0      0      255     -1 notrap
```

```
notrap  13  14  15  16  33  37  38  46  47
19 00000000360001e0000      1       0       0       255     -1 notrap
notrap  17  18  19  20  37  38  40  41
31 0000000007e00000000      1       0       0       255     -1 notrap
notrap  33  34  35  36  37  38
32 00000000061e0000000      1       0       0       255     -1 notrap
notrap  29  30  31  32  37  38
71 0000000037e00000000      1       0       0       255     -1 notrap
notrap  33  34  35  36  37  38  40  41
72 00000003061e0000000      1       0       0       255     -1 notrap
notrap  29  30  31  32  37  38  44  45
73 00000000c6004000000      1       0       0       255     -1 notrap
notrap  26  37  38  42  43
74 0000000c06002000000      1       0       0       255     -1 notrap
notrap  25  37  38  46  47
75 0000000036001000000      1       0       0       255     -1 notrap
notrap  24  37  38  40  41
76 0000000306000800000      1       0       0       255     -1 notrap
notrap  23  37  38  44  45
77 00000000c6000400000      1       0       0       255     -1 notrap
notrap  22  37  38  42  43
78 0000000c06000200000      1       0       0       255     -1 notrap
notrap  21  37  38  46  47
Result: Int32: 0 = 0 = 0x00000000
```

MAC Table

Command	Description
`fmDbgDumpMACTable`	Dump MAC Address Table.
`fmDbgDumpMACTableEntry <switch>,<vlan>,"<MAC>"`	Dump MAC Address table entry in a vlan.

Example,

```
fmShell> fmDbgDumpMACTable
```

```
MA_TABLE[12530]:
CACHE ENTRY         DMAC ENTRY          SMAC ENTRY
State           : Locked              Valid               Valid
MAC Address     : 3333fffe8006        3333fffe8006        3333fffe8006
FID             : 14                  14                  14
Address Type    : STATIC              --                  --
Glort           : --                  0x300f              0x300f
Port            : 63                  63                  63
Trigger         : 1                   1                   1
Secure          : false               false               false
MA_TABLE[12599]:
….
10 entries listed
```

SFP+/QSFP28

Command	Description
`fmPlatformDumpXcvrEeprom <switch>,<port>`	Dump data for SFP+/QSFP28 devices connected to the specified switch port.
`fmPlatformDumpXcvrState <switch>,<port>`	Show status/control lines for SFP+/QSFP28 devices connected to the specified switch port.

Example,

```
fmShell> fmPlatformDumpXcvrEeprom 0,26
Switch: 0 Port: 26 EEPROM
Page 0
00: 03 04 21 01 00 00 04 41 84 80 d5 00 78 00 00 00   ..!....A....x...
10: 00 00 01 00 4d 6f 6c 65 78 20 49 6e 63 2e 20 20    ....Molex Inc.
20: 20 20 20 20 00 00 09 3a 37 34 37 35 32 2d 31 31       ...:74752-11
30: 30 31 20 20 20 20 20 20 20 20 20 20 00 00 00 69    01          ...i
40: 00 00 00 00 32 31 36 31 33 30 30 33 30 20 20 20    ....216130030
50: 20 20 20 20 31 32 30 36 30 39 20 20 00 00 00 12       120609  ....
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

```
90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................

Identifier[00]: SFP(0x3)

Connector Type[02]: Copper Pigtail(0x21)
Spec Comp[03]: 1X CopperPass 1000BASE-CX SD EL AL PassiveCable
TwinAxial(TW)
Length[12]: 1m
VendorName[14]: Molex Inc.
VendorOUI[25]: 0x00093a :
VendorPN[28]: 74752-1101
VendorRev[38]: CC_Base[3f]: 69 Calculated: 69
VendorSN[44]: 216130030
DateCode[54]: 120609
CC_EXT[5f]: 12 Calculated: 12
Result: Int32: 0 = 0 = 0x00000000
fmShell> fmPlatformDumpXcvrState 0,26
Switch: 0 Port: 26
ENABLE: no
PRESENT: yes
RXLOS: no
TXFAULT: no
Result: Int32: 0 = 0 = 0x00000000
```

Packet IO

Command	Description
`bdPktSend <dest>, <port>, <vlan>, <size>`	Send packet to 64-bit dest, port (-1=switched), vlan and packet size.

Miscellaneous

Command	Description
`fmDbgDumpDriverCounts`	Send packet to 64-bit dest, port (-1=switched), vlan and packet size.
`fmDbgDumpMemoryUsage`	Dumps switch internal memory usage.
`fmDbgDumpScheduler`	Dumps switch internal scheduler information.
`fmDbgDumpStatChanges`	Dumps statistic changes for all ports.

SerDes Debugging/Tuning

Command	Description
`fmDbgDumpPortMapV2 <switch>,1,2`	This command provides the link between switch ports and Serdes Lanes.
`fmDbgSetSerDesTxPattern <switch>,<serdes>,"prbs31"`	Set PRBS Pattern 31 in Serdes Transmit path.
`fmDbgSetSerDesRxPattern <switch>,<serdes>,"prbs31"`	Set PRBS Pattern 31 in Serdes Receive path.
`fmDbgDumpSerDes 0,<serdes>,"status"`	Dumps SerDes Status information.

Example,

```
fmShell> fmDbgDumpPortMapV2 0,1,2
LogPort PhysPort FabricPort EPL/PEP Lane SERDES SBUS      Type
polarity

1.0       1       0       0      0    0    02    40GBASE_KR4     NONE
1.1      --      --      --      1    1    03     ----------     NONE

1.2      --      --      --      2    2    04     ----------     NONE

1.3      --      --      --      3    3    05     ----------     NONE

2         2       1       0      1    1    03       DISABLED     NONE
```

```
3          3        2        0        2     2     04        DISABLED     NONE
4          4        3        0        3     3     05        DISABLED     NONE
5.0        5        4        1        0     4     06     40GBASE_KR4     NONE
5.1       --       --       --        1     5     07      ----------     NONE
5.2       --       --       --        2     6     08      ----------     NONE
5.3       --       --       --        3     7     09      ----------     NONE
6          6        5        1        1     5     07        DISABLED     NONE
7          7        6        1        2     6     08        DISABLED     NONE
8          8        7        1        3     7     09        DISABLED     NONE
9.0        9        8        2        0     8     0a     40GBASE_KR4     NONE
9.1       --       --       --        1     9     0b      ----------     NONE
9.2       --       --       --        2    10     0c      ----------     NONE
9.3       --       --       --        3    11     0d      ----------     NONE
10        10        9        2        1     9     0b        DISABLED     NONE
11        11       10        2        2    10     0c        DISABLED     NONE
12        12       11        2        3    11     0d        DISABLED     NONE
13.0      13       12        3        0    12     0e     40GBASE_KR4     NONE
13.1      --       --       --        1    13     0f      ----------     NONE
13.2      --       --       --        2    14     10      ----------     NONE
13.3      --       --       --        3    15     11      ----------     NONE
....
```

```
fmShell> fmDbgDumpSerDes 0,0,"status"
---------------------------------
SERDES 0: General Status
---------------------------------

KR support          : YES
TxReady/RxReady     : Y/Y
Cfg Speed           : (1)- 10 Gbps
Loopback            : SERDES_LB_OFF
Polarity            : SERDES_POLARITY_NONE
TX Data Sel         : SERDES_TX_DATA_SEL_CORE
RX Data Sel         : SERDES_RX_CMP_DATA_PRBS7
Error Counter       : 63
signalOk            : 1
PreCursor           : 4
Atten               : 11
PostCursor          : 0
RX Termination      : SERDES_RX_TERM_AVDD
Initial Eye Height  : 32 (500 mV)
Forced Reset Count  : 0
Result: Int32: 0 = 0 = 0x00000000
```

Command Line Mode

You can run the switch commands also in a command line mode, if you want to run the commands from a script environment you can use the same commands as in the interactive shell mode. You need to specify the switch commands as follows:

```
fmShell -Ls -- -e <switch command>
```

Example,

```
fmShell -Ls -- -e "ps -1"
```

4.1.4.1.2 TestPoint Software

TestPoint is Intel's proprietary configuration scripting tool designed to offer a simple method to exercise the Intel ® Ethernet Switch Family Software API on a target platform. The software is designed to be easy to use, easy to modify and is intended for software development, testing, demonstration and manufacturing.

The software is not designed to replace a full scale Ethernet switch management software implementation.TestPoint source code is delivered bundled with each release of the Intel ® Ethernet Switch Family Software API. TestPoint may be adapted to run on any custom platform running Linux with Perl installed. Use of TestPoint is optional and is not required to run the software API or switch devices.

TestPoint is installed at `/opt/bladeservices/lib/TestPoint/perl` on the ATCA-7490 blade. Run the IES initialization (as described in Chapter 4, *Intel FM10000 Initialization*) first (if not already done), then change into the directory `/opt/bladeservices/lib/TestPoint/perl` and run `TestPointShared`.

```
[root@ATCA-7490-8 perl]# ./TestPointShared
kernel.shmmax = 536870912
Setting up multi-process key=501
>> Loading TestPoint Module (-)
>> Module loaded in 3 seconds
TestPoint - interactive configuration environment [4.1]
Copyright (C) 2007-2016 Intel Corporation.  All rights reserved.
Unauthorized disclosure is prohibited.
Instantiating platform libertyTrail...
Connect to host driver using /dev/uio0
Basic Help:

<tab>     - tab complete commands
?         - context sensitive help
!0,2..4   - override selected switches
load      - load a saved script
save      - save the command history as a script
quit      - quit application
Running as non-master process, skipping startup RC...
TestPoint loaded in 6 seconds
<0>%
<0>% show switch info
Switch Information:
------------------------------------------------------------
Switch Family        FM10000
Switch Model         FM10840
```

```
Switch Revision      B0
<0>% quit
Exiting TestPoint …
Terminate the API.
```

5.1 Overview

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

5.2 Firmware Recovery Image Files

To FCU supports firmware images in HPM.1 format. HPM.1 is a PICMG standard to upgrade IPMCs.

By default, the image files for the current hardware configurations are loaded as part of the BBS software in `/opt/bladeservices/rom` directory, when the blade-specific firmware support packages are installed.

The following image files are currently supported.

Filename	Description
`atca-7490-cpu-<version>.hpm`	BIOS image
`atca-7490-ipmc-boot-<version>.hpm`	Firmware for the IPMC booter
`atca-7490-ipmc-<version>.hpm`	Firmware for the IPMC
`atca-7490-ipmc-all-<version>.hpm`	Combined image for IPMC and IPMC booter (only delivered when the update of the booter is required)
`wiht atca-7490_glue_fpga_<version>.bin.hpm`	Glue FPGA Bitstream

NOTICE

The **`ipmc-boot`** and the **`ipmc-all`** firmware will not be installed on the board any more. If there is a need to upgrade the boot-loader, this will be explicitly stated in the release notes. The firmware images will only be released in hpm.1 file format.

5.3 fcu—Firmware Upgrade Command-Line Utility

fcu displays upgrade capabilities of firmware devices on a board. If the board is managed by an IPMC, then fcu requests IPMC to determine the board type. A board may have multiple devices, which are low-level hardware components, like BIOS, FPGA, and IPMC and so on. fcu abstracts firmware upgrade operations to a common set of operations for all devices. The fcu user does not need to know internals of a device. fcu identifies a device by its name. The name is a combination of the board name and the device properties.

For example, the BIOS for the ATCA-7490 has the name "atca-7490-cpu". fcu provides the following upgrade operations:

- Query the device to return the firmware version and other information
- Query the firmware image to return the firmware version and other information
- Validate the firmware image
- Verify whether the image is applicable on the target device
- Upgrade the device with the given firmware image
- Compare the firmware image with an installed firmware image on the device
- Mark a bank of the device to become active after the next reset or power cycle
- Activate a bank of the device immediately

5.3.1 Query Operation

Using the `Query` operation, fcu returns 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. Query operation shows all the banks of a device. One of these banks is the active version, which means that the device was booted with the firmware installed in that bank. The device might have a second bank that contains the rollback version. You can switch to the rollback version with the help of activate operation or with mark operation in combination with a reboot or power-cycle.

If the device supports the mark operation, the query operation shows which bank is marked for next use. Furthermore, fcu shows the capabilities of a device. Device capabilities are set of fcu operations, such as manual, automatic rollback, or self test implemented. The following example shows the BIOS of the ATCA-7490 board.

```
root@ATCA7490:~# fcu --query -d atca-7490-cpu
********************[[[[[REPORT BEGIN]]]]]********************

Operation: Query

#10 Device   : atca-7490-cpu
Bank #1 -      Active Version: 0.5.00000000
Bank #0 -    Rollback Version: 0.5.00000000
Bank marked for next use: #1
********************[[[[[ REPORT END ]]]]]********************
```

In the first line, the number of the device and its name are displayed. In Bank1, the active version is stored and this bank is also marked for next use. Bank0 contains the rollback version. The device supports the upgrade, mark, and compare operation. The activate operation is not supported. The version is always a combination of decimal numbers.

5.3.2 Show Operation

Show operation does not access any device. It only operates with the firmware image and it shows the metadata, which is part of the image. Furthermore, it validates the firmware image to compare the checksum part of the metadata against the checksum of the raw image. The output of the show operation is similar to the output of the query operation.

```
root@ATCA7490:~# fcu --show --file /opt/bladeservices/rom/atca-7490-cpu-
0.5.0.hpm
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Show
Manufacturer : ARTESYN
Board        : atca-7490
#00 Device   : ATCA7490 BIOS Image
Bank #0 -             Version: 0.05.00000000
********************[[[[[ REPORT END ]]]]]********************
```

Additionally, this operation shows the name of the manufacturer and the name of the board to which the firmware of this image is compatible to. A firmware image does not have multiple banks. So you can see only one bank.

5.3.3 Mark Operation

`Mark` operation selects the bank that is used after the next reset or reboot. With this operation, you have to specify the name of the device and the bank.

NOTICE

Not all devices support this operation.

5.3.4 Activate Operation

`Activate` operation is similar to that of `mark` operation. Because, this operation also sets a bank to active state; but the bank is activated immediately, which is not in the case of `mark` operation. Not every device supports this operation. IPMCs that have the HPM.1 must support the `Activate` operation on the rollback bank and the deferred bank. If the rollback bank is activated by this operation, a manual rollback is performed.

5.3.5 Compare Operation

With `Compare` operation, you can compare firmware images on the target with images in the upgrade file. You have to specify the firmware image file and the bank which you want to compare.

5.3.6 Upgrade Operation

`Upgrade` operation uploads the firmware image to the device. To have a valid firmware image in one bank, fcu tries to protect the active bank being overwritten by a new firmware image. For devices like IPMC, this protection is done by the IPMC firmware itself, which communicates with fcu during the image upload. The IPMC firmware selects the bank to write the new image too. Fcu determines this bank by reading an IPMI sensor, which knows the active bank. For that reason, you have to specify the firmware image with the `Upgrade` operation.

5.3.7 Verify Operation

`Verify` operation checks if the firmware image is applicable on a device of the blade. It prevents you from writing a firmware image into a non-compatible device. The `Verify` operation is always done before an Upgrade operation. So, you do not have to specify it explicitly when you perform an upgrade.

5.3.8 Command-Line Options

```
fcu --help
```

Usage

```
fcu [operations] [operands]
```

Operations

`--query`

`-q` Set to perform a query operation

`--upgrade`

`-u` Upgrade the unused version of firmware

This operation requires the --file flag

`--help`

`-h` Display this help message

`--show`

`-s` Display information about the target which is included in the given upgrade file

This operation requires the --file flag

`--verify`

`-v` Set to perform a verification of an upgrade file

This operation does not install the upgrade image

This operation requires the --file flag to be set

`--mark`

`-m` Mark the specified bank as next to boot

This operation depends on the --bank and the --device flag

`--activate`

`-r` Activate the specified bank

This operation depends on the --bank and the --device flag

`--compare`

`-c` Compare the operation firmware with the image specified by the --file flag

`--version`

Display the version of this utility

Operands

`--device=<device name>`

`-d` Device to perform operation on

`--file=<filename>`

`-f` Filename of the firmware file

`--bank=<bankletter>`

`-b` Bank-letter for mark/compare command

`--level=[0-7]`

severity level of logging, 7 logs everything, default is 5

`--log=ARG`

file name for logging

5.4 Upgrading Firmware Image

This section describes recommended procedures for upgrading firmware devices.

NOTICE

The shown file names and paths are only for example and should be replaced with file names and paths applicable to your configuration.

5.4.1 BIOS Firmware Upgrade of the ATCA-7490 Blade

Follow these steps for BIOS firmware upgrade on the ATCA-7490 blade:

1. Query the current firmware versions.

```
root@ATCA-7490:/root# fcu --query -d atca-7490-cpu
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Query
 #08 Device   : atca-7490-cpu
  Bank #1 -      Active Version: 0.5.00000000
  Bank #0 -    Rollback Version: 0.5.00000002
 Bank marked for next use: #1
  Capabilities        :
       Upgrade operation   : (x)
          Mark operation   : (x)
      Activate operation   : ( )
      Compare operation    : (x)
       Service affected    : ( )
       Manual Rollback     : (x)
       Automatic Rollback  : ( )
       Self Test           : ( )

********************[[[[[ REPORT END ]]]]]********************
```

2. Upgrade bank 0 to version 0.9.4

```
root@ATCA-7490:/mnt# fcu -uf /opt/bladeservices/rom/atca-7490-
cpu-0.9.4.hpm
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Upgrade
Hardware Write Protection off!
  erasing ...100 %
  writing ...100 %
Result   : Success
********************[[[[[ REPORT END ]]]]]********************
```

Note: The Firmware will be released in hpm.1 file format, only. FRI file format is no longer supported.

3. Mark Bank0 for next use.

```
root@ATCA-7490:/mnt# fcu -m -b 0 -d atca-7490-cpu
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Mark
Result   : Success
********************[[[[[ REPORT END ]]]]]********************
```

4. Check if the new firmware version (here 0.9.4) is in Bank0 and whether the Bank0 is marked.

```
root@ATCA-7490:/mnt# fcu -q -d atca-7490-cpu
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Query
 #08 Device   : atca-7490-cpu
  Bank #1 -      Active Version: 0.9.00000003
  Bank #0 -    Rollback Version: 0.9.00000004
 Bank marked for next use: #0
  Capabilities        :
       Upgrade operation   : (x)
          Mark operation   : (x)
      Activate operation   : ( )
      Compare operation    : (x)
       Service affected    : ( )
       Manual Rollback     : (x)
       Automatic Rollback  : ( )
```

```
       Self Test           : ( )
********************[[[[[ REPORT END ]]]]]********************
```

Note: BIOS will be activated after a powercycle.

5.4.2 FPGA Upgrade

The FPGA on the ATCA-7490 board can also be updated with the FCU tool.

To upgrade FPGA on the ATCA-7490 blade:

1. Run the following command:
   `fcu -uf /opt/bladeservices/rom/atca-7490_glue_fpga-02.bin.hpm`
   ```
   ********************[[[[[REPORT BEGIN]]]]]********************
   Operation: Upgrade
   preparation stage started
   capabilities of target and image header successfully compared
   properties of component FPGA retrieved
   properties of component FPGA retrieved
   preparation stage successfully finished
   preparing upload
   upload stage of ATCA7490 FPGA Image started
   initializing upload
   uploading ...100 %
   finishing upload
   upload stage finished
   activation stage of new firmware started
   Result   : Success
   ********************[[[[[ REPORT END ]]]]]********************
   ```
   Example: After a successful upgrade, above output is displayed.
2. To switch the FPGA bank.
   To read the current FPGA bank, run the following command:
   `fcu -m -b BANK<0/1> -d "`**`PYLD FPGA`**`"`
3. Power cycle the board to run the new FPGA image.

6.1 Overview

Hardware management in ATCA systems is based on the Intelligent Platform Management Interface (IPMI) specification. IPMI commands can be complex and cumbersome. To facilitate blade-level management, Artesyn provides the Hardware Platform Management (HPM) package, which provides a set of commands that are based on IPMI commands. These commands are easy to use than the IPMI command itself. A HPM command can encapsulate a sequence of IPMI commands. For example, reading the FRU Inventory data. A HPM command can be the unifier for OEM IPMI commands that are different on different blade types. For example, reading the CPU boot bank. For a catalogue of supported IPMI commands of the blade, refer to the respective IPMI manual.

The HPM package consists of:

- HPM daemon, hpmagentd
- Command line utility, hpmcmd
- Script framework for managing shutdown, reboot, and local ekeying events

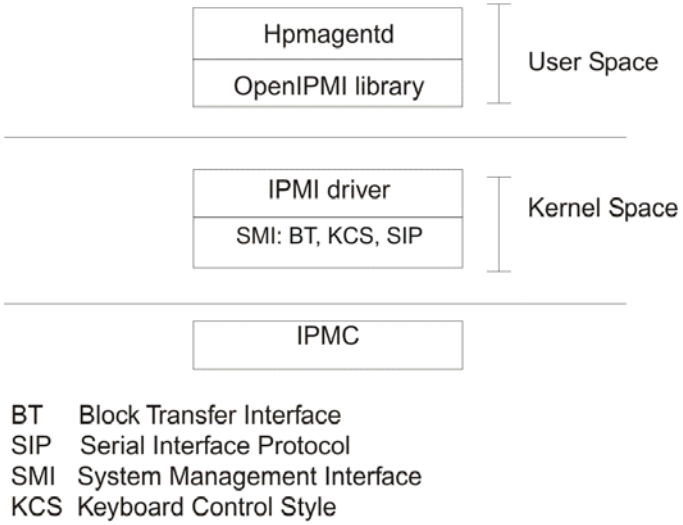
The HPM daemon is responsible to wait for events from the IPMC to perform a graceful shutdown/reboot of the Operating System and to react when the link state of a channel's port is changed.

The utility hpmcmd displays the response of commands on the console in a human-readable format. HPM commands include:

- Retrieving and modifying FRU data
- Reading and controlling status of IPMI-controlled LEDs
- Communicating local slot location information
- Retrieving the event messages from the SEL of the IPMC

The hpmagentd and hpmcmd make use of OpenIPMI driver to talk to the local IPMC. The following figure shows the software levels that are involved in the HPM architecture:

Figure 6-1 Software Levels of the HPM Architecture



The System Management Interface (SMI) driver provides the low level interface for talking to the IPMC and could be a KCS driver or Block Transfer (BT) driver or other. If you need more information about the software aspects of the blade IPM controller, refer to the respective IPMI manual.

6.2 hpmagentd—HPM Agent Daemon

6.2.1 Description

The `hpmagentd` is the service to process events from the local IPMC. For any incoming event, it calls the respective script which is part of the hpmagentcmd package. Event data is passed to a script by command line arguments. You can modify a script to fulfill your requirements. The following events are handled by the daemon:

- **Graceful Shutdown** - When the IPMC receives a FRU activation request to deactivate a FRU, then it redirects the command to the hpmagentd through the IPMI driver. The hpmagentd invokes the shutdown script which is located in `/opt/bladeservices/bin/hpmshutdown`. By default, within the script `shutdown -h now` is called, which initiates a shutdown of the Linux immediately.

NOTICE

The IPMC powers down the processor in any case after a certain time following this command. You may adjust this time with the Graceful Shutdown Timeout parameter of the IPMC, which can be set with a SetSystemBootOption IPMI command.

- **Graceful Reboot** - On receiving a FRU control request to gracefully reboot the payload, the IPMC sends the command to the hpmagentd. The daemon invokes the reboot script which is available at `/opt/bladeservices/bin/hpmreboot`.
- **Ekeying Events** - When the IPMC modifies the link state of a port, then it notifies the hpmagentd about that change. If the link goes down, then the script hpmekeydown is invoked, otherwise, the daemon calls hpmekeyup. Both scripts are placed in `/opt/bladeservices/bin`. By default, these scripts are empty. To identify which port was changed, the hpmagentd passes an argument to a hpmekey script in the format: First the interface is specified: BC for Base Channel, FC for Fabric Channel, UC for Update Channel, and AMC for AMC Channel. The channel number and the port numbers of the channel are specified in the below example.
  For example, the ports 1,2,3,4 of the channel 1 of a base interface is changed, then the argument looks like: "BC1.1,2,3,4".

6.2.2 Deployment

By default, the HPM daemon is installed in the `/opt/bladeservices/bin` directory. With the `hpmagentd` binary, the scripts `hpmreboot`, `hpmshutdown`, `hpmekeyup`, and `hpmekeydown` are stored in that directory. Additionally, there is an `init` script hpm to start and stop the daemon and the `hpmvar` script, which exports some important variables to `/etc/default/hpmvars` to describe the board.

Synopsis

```
hpmagentd [options]
```

Parameters

```
 --log =<file name>
```

You may specify a log file for the daemon with this option. If you do not use it, then the `hpmagentd` logs to the syslog.

```
-l  --level
```

Specifies the level of message logging, where 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

```
-v --version
```

Displays the version of the daemon

```
-L --disable-led
```

Disable the LED management

`-r --reboot-script=<script>`

Use the specified Blade Reboot script. The default script is `/opt/bladeservices/bin/hpmreboot`

`-s --shutdown-script=<script>`

Use the specified Blade Shutdown script. The default script is `/opt/bladeservices/bin/hpmshutdown`

`-u --ekey-up=<script>`

Called when a port is enabled. Default is `/opt/bladeservices/bin/hpmekeyup`

`-d --ekey-up=<script>`

Called when a port is disabled. Default is `/opt/bladeservices/bin/hpmekeydown`

`-h --help`

Displays the help message

`-i --dont-daemonize`

Run interactively

6.2.3 hpm - Init.d Script

The `hpm init.d` script allows to start, stop, and restart the `hpmagentd`. It can be linked to a run-level to automatically start the daemon at Linux boot time and to stop it when Linux shuts down.

Synopsis

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

Parameters

`Start` - Starts the hpmagentd

`Stop` - Stops the hpmagentd

`Restart` - Stops and starts the hpmagentd again

`Status` - Reports if the hpmagentd is currently running or not

6.3 hpmcmd—HPM Command Utility

6.3.1 Overview

The HPM command utility communicates directly to the IPMC through the IPMI driver, which is part of the Linux kernel. It takes care of translating the user-friendly commands into elaborated IPMI commands that the IPMC is able to understand. Those IPMI commands are transferred to the local IPMC. The HPM command utility can be started in interactive mode, where a prompt is displayed and the user enters commands; or it can process a single command.

By default, the hpmcmd binary is installed in `/opt/bladeservices/bin` directory.

Synopsis

```
hpmcmd [options]
```

Parameters

`-c` Processes a single command

`--help -h` Displays this help message

`-v` Verbose mode for some commands

Some commands like `fruinfoget`, print more details if these options is given. Commands, which do not support, the verbose option ignores it.

`-t` Sends the command to a remote target.

For more information, refer *Target Addressing with hpmcmd* on page 72. If this option is not given, then the command goes to the local IPMC.

`-p` Changes the prompt when `hpmcmd` runs in interactive mode.

`-o` Prints results to a file

6.3.2 Target Addressing with hpmcmd

Using the `-t` option, you can send commands to other IPMCs or MMCs, which participate on an IPMB.

Syntax: -t <IPMB address>[: MMC address]. The addresses must be set in hexadecimal format.

To send the command to an AMC attached on this blade use:

```
-t0:72 or -tlc:c0
```

To send the command to another IPMC type:

```
-t 92
```

To send the command to a MMC, which is attached on another blade in the shelf specify:

```
-t 82:72
```

6.3.3 Command Overview

The following table lists all commands from the hpmcmd program available on ATCA-7490. You can display this list and a short description about command using the help command (see, section *help* on page 86). A detailed description of the commands is provided in section *Supported Commands* on page 74.

Table 6-1 Command Overview

Command	Description
bootbankget	Gets the bootbank to boot from
bootbankset	Sets the bootbank to boot from
bootparamerase	Erases boot parameter value
bootparamget	Gets the boot parameter value
bootparamset	Sets a boot parameter value
chinfo	Retrieves the channel information
cmd	Executes IPMI commands
deviceid	Gets the Device ID

Table 6-1 Command Overview (continued)

Command	Description
frudata	Allows to get FRU info in hexadecimal numbers
fruinfoget	Gets the string fields from the FRU
fruinv	Gets the FRU size and addressable units
fruread	Reads 'x' number of bytes from the FRU
fruwrite	Writes 'x' number of bytes to the FRU
fwprogevent	Sends a Firmware Progress Sensor Event
help	Gets the list of commands
ipmbaddress	Gets the IPMB address
ipmcstatus	Gets the IPMC status
Lancfgget	Gets the LAN configuration parameter
Lancfgset	Sets the LAN configuration parameter
ledget	Gets the state of a specific FRU LED
ledprop	Get the LED properties for this FRU
ledset	Controls the state of a specific FRU LED
loglevelget	Gets the hpmagentd log level
macaddress	Lists the MAC addresses
partnumber	Gets the board part number
physlotnumber	Gets the board physical slot number
portget	Shows the current state of network interfaces governed by E_Keying events
portset	Enables/Disables the ports in a channel
posttypeget	Gets the posttype to run at boot
posttypeset	Sets the posttype to run at boot
sdr	Shows SDR records
sdr_dump	Shows the SDR records in raw format
sdrinfo	Shows SDR information
sel	Shows SEL records

Table 6-1 Command Overview (continued)

Command	Description
selclear	Erases all contents from the SEL
selinfo	Shows SEL information
sendamc	Sends an IPMI request to a MMC behind a remote IPMC
sendcmd	Sends an IPMI request to an IPMB address IPMC
serialoutputget	Determines which serial output source goes to a particular serial port connector
serialoutputset	Selects the serial output source of the serial port connector
shelfaddress	Gets the Shelf Address String
shelfslots	Prints the number of slots in the shelf
shelftype	Gets the Shelf Type from the Shelf FRU (Board Product Name)
slotmap	Prints the slotmap of the shelf
slotnumber	Shows the board slot number
solcfgget	Gets SOL configuration parameter
solcfgset	Sets SOL configuration parameter
version	Shows the hpmcmd version
watchdog	Controls Payload WDT functionality

6.3.4 Supported Commands

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

6.3.4.1 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 on Artesyn ATCA blades is stored in redundant persistent memory devices. This allows the firmware image in one bank to serve as a backup for other bank. During normal operation, the CPU on a blade determines which bank to boot from — based on a GPIO signal controlled by the IPMC. This bank is considered as 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>
```

Parameters

`payload_cpu_selector`

Is an integer between 0 and the number of CPU devices supported on the blade. On the ATCA-7490 the `payload_cpu_selector` is 0.

Example

```
hpmcmd -c bootbankget 0
```

BANK1

```
hpmcmd -c bootbankget 1
```

BANK0

6.3.4.2 bootbankset

Description

This command sets the boot bank for a particular CPU from which the blade 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 blade. On the ATCA-7490 the `payload_cpu_selector` is 0.

`newBootBank`

Can be set to `BANK1`, `BANK2`...

Example

```
hpmcmd -c bootbankset 1 BANK 1
```

6.3.4.3 bootparamerase

Description

This command erases the boot parameter.

Synopsis

```
bootparamerase section [name] [-t ipmbAddr[:mmcAddr]]
```

Parameters

`section`

Can have value as `USER`, `DEFAULT`, `TEST`, or `OS_PARAM`.

`name`

Specifies name of the parameter.

`t`

Sends the command to `ipmbAddr:mmcAddr`.

6.3.4.4 bootparamget

Description

This command gets the boot parameter value.

Synopsis

```
bootparamget section [name] [-t ipmbAddr[:mmcAddr]]
```

Parameters

`section`

Can have value as `USER`, `DEFAULT`, `TEST`, or `OS_PARAM`.

`name`

Specifies name of the parameter.

`t`

Sends the command to `ipmbAddr:mmcAddr`.

6.3.4.5 bootparamset

Description

This command sets the boot parameter value.

Synopsis

```
bootparamset section name=value [-t ipmbAddr[:mmcAddr]]
```

Parameters

`section`

Can have value as `USER`, `TEST`, or `OS_PARAM`.

`name`

Specifies name of the parameter.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

6.3.4.6 cmd

Description

This command allows you to enter the 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 address> <netfn cmd> <cmd data>
```

Parameters

`ipmi command`

The `ipmi` command specifies the sequence of hexadecimal bytes. The `ipmi` command is entered using the ipmicmd tool from the OpenIPMI library. The `ipmi` command can have value, such as:

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

In this example:

XX specifies netfn in hexadecimal.

ZZ specifies the command number, as stated in the IPMI/PICMG specification.

`W1` to `Wn` specifies the data bytes which the command supports.

`ipmi address`

The IPMI address specifies the IPMC that receives the command, it can be the local IPMC or another IPMC on the IPMB. The IPMI address for the local IPMC consists of `<f LUN>`, where `f` is the BMC channel number. The IPMI address for a remote IPMC consists of `<0 SA LUN>`, where SA is the slave address.

`netfn cmd`

Identifies the command type.

`cmd data`

Specifies the message data associated with the command.

Example

GetDeviceId command to the local IMPC:
`hpmcmd -c cmd f 0 6 1`

GetDeviceId command to the remote IPMC on address 9a:
`hpmcmd -c cmd 0 9a 0 6 1`

GetDeviceId command to the AMC attached on this blade. MMC address of the AMC is 7a:
`hpmcmd -c cmd 7 7a 0 6 1`

6.3.4.7 deviceid

Description

This command retrieves the raw IPMI `GetDeviceID` command response and decodes the IPMI message.

Synopsis

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

Parameters

`-t`

Sends the command to `ipmbAddre:mmcAddr`.

Example

```
root@ATCA-7490:~# hpmcmd -c deviceid

DEVICEID INFORMATION
--------------------
Device Id           =  0x00
Device Revision     =  0x00
Device Mode         =  Normal Operation; Supports Device SDR
Firmware Revision   =  1.20
IPMI Version        =  2.0
```

```
Device Support      =  IPMB Evnt Gen; FRU; SEL; Sensor;
Manufacturer ID     =  0x000065CD
Product ID          =  0x201A
Auxiliary Revision  =  0x00000024 4
```

6.3.4.8 chinfo

Description

Retrieves information about an IPMI channel.

Synopsis

`Chinfo <channel>`

Parameters

`Channel`-Channel number

Example

```
root@ATCA7490:~# hpmcmd -c chinfo 0
Channel Medium Type   : IPMB (I2C)
Channel Protocol Type : IPMB-1.0
Session Support       : session-less
Active Session Count  : 0
Protocol Vendor ID    : 001BF2
```

6.3.4.9 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 ATCA-7490 front board and 1 for the ARTM.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
root@ATCA7490:~# hpmcmd -c frudata 0
01 00 00 01 0b 14 00 df 01 0a 19 20 40 90 c7 41
52 54 45 53 59 4e e3 50 43 41 2c 41 54 43 41 2d
37 34 38 30 2f 31 30 35 57 2f 30 47 42 2f 52 31
2e 30 2f 43 46 47 30 30 30 30 c7 45 31 35 32 44
38 43 ca 30 31 30 36 38 34 38 4e 30 31 c2 32 20
c1 00 00 00 00 00 00 14 01 09 19 c7 41 52 54 45
53 59 4e c9 41 54 43 41 2d 37 34 38 30 da 41 54
43 41 2d 37 34 38 30 2f 30 47 42 2f 52 31 2e 30
2f 43 46 47 30 30 30 30 c5 52 2e 31 2e 30 c7 45
```

6.3.4.10 fruinfoget

Description

This command retrieves information from the specified FRU.

Synopsis

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

Parameters

`fruid`

Is 0 for the ATCA-7490 front board and 1 for the ARTM.

`field`

Is one of the following data fields. If no field is specified, it retrieves the entire FRU information.

Field	Description
bmanufacturer	Board area manufacturer
bproductname	Board area product name
bserialnumber	Board area serial number
bpartnumber	Board area part number
pmanufacturer	Product area manufacturer
pproductname	Product area product name
ppartnumber	Product area part number
pversion	Product area version
pserialnumber	Product area serial number
passettag	Product area asset tag

`-v`

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

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
root@ATCA7490:~# hpmcmd -c fruinfoget 0
Common Header:
    Format Version = 1
Board Info Area:
    Version   = 1
    Language Code           = 25
    Mfg Date/Time          = Dec 22 00:00:00 2013 (9453600 minutes
since 1996)
    Board Manufacturer      = ARTESYN
    Board Product Name      = PCA,ATCA-7490/105W/0GB/R1.0/CFG0000
    Board Serial Number     = E152D8C
    Board Part Number       = 0106848N01
```

```
    FRU Programmer File ID   = 2
Product Info Area:
    Version   = 1
    Language Code            = 25
    Manufacturer Name        = ARTESYN
    Product Name             = ATCA-7490
    Product Part / Model#    = ATCA-7490/0GB/R1.0/CFG0000
    Product Version          = R.1.0
    Product Serial Number    = E152D8C
    Asset Tag                =
    FRU Programmer File ID   =
Multi Record Area:
    PICMG LED Descriptor Record (ID=0x2f)
        Version = 0
    OEM MAC Addresses Record (ID=0x01)
        Version = 1
    Artesyn Unknown Record (ID=0x10)
    PICMG Board Point-to-Point Connectivity Record (ID=0x14)
        Version = 1
    AMC Carrier Information Table Record (ID=0x1a)
        Version = 0
    AMC Carrier Activation and Current Management Record (ID=0x17)
        Version = 0
    AMC Carrier Point-to-Point Connectivity Record (ID=0x18)
        Version = 0
    AMC Point-to-Point Connectivity Record (ID=0x19)
        Version = 0
```

6.3.4.11 fruinv

Description

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

Synopsis

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

Parameters

`fruid`

Is 0 for the ATCA-7490 Front board and 1 for the ARTM.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

Example

FrontBoard:

```
root@ATCA-7490:~# hpmcmd -c fruinv 0
FruSize = 2048
Accessed Units = Bytes
```

ARTM:

```
root@ATCA-7490:~# hpmcmd -c fruinv 1
FruSize = 512
Accessed Units = Bytes
```

6.3.4.12 fruread

Description

This command gets nBytes of fruid from the startAddress of the specified FRU.

Synopsis

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

Parameters

`fruid`

Is 0 for the ATCA-7490 Front board and 1 for the ARTM.

`startAddress`

Is the starting address for reading the `fruid`.

`nBytes`

Number of bytes to read in decimal.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
root@ATCA7490:~# hpmcmd -c fruread 0 0 100
01 00 00 01 0b 14 00 df 01 0a 19 20 40 90 c7 41
52 54 45 53 59 4e e3 50 43 41 2c 41 54 43 41 2d
37 34 38 30 2f 31 30 35 57 2f 30 47 42 2f 52 31
2e 30 2f 43 46 47 30 30 30 30 c7 45 31 35 32 44
38 43 ca 30 31 30 36 38 34 38 4e 30 31 c2 32 20
c1 00 00 00 00 00 00 14 01 09 19 c7 41 52 54 45
53 59 4e c9
```

6.3.4.13 fruwrite

Description

This command allows to write hexadecimal byte values to `fruid` starting at `startAddr`.

Synopsis

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

Parameters

`fruid`

Is 0 for the ATCA-7490 Front board and 1 for the ARTM.

`startAddress`

Is the starting address for writing.

`hexval1 .. hexvalN`

Is the hexadecimal value to write.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

6.3.4.14 fwprogevent

Description

This command sends a Firmware Progress Sensor Event to the Shelf Manager SEL. Refer *IPMI specifications* for details on values.

Synopsis

```
fwprogevent <data1> <data2> <data3>
```

Parameters

`data1`

Stores hexadecimal value as "00" for Error, "01" for Hang, and "02" for Progress.

`data2`

Stores hexadecimal value as "00-0D" for Error, "00-19" for Hang or Progress.

`data3`

Stores hexadecimal value as "FF", unless an OEM data2 is specified.

6.3.4.15 help

Description

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

Synopsis

```
help
```

6.3.4.16 ipmbaddress

Description

This command retrieves the blade IPMB address.

Synopsis

```
ipmbaddress
```

Example:

```
hpmcmd -c ipmbaddress
ipmbaddress is 0x88
```

6.3.4.17 ipmcstatus

Description

This command retrieves the status of given IPMC.

Synopsis

```
ipmcstatus [-t ipmbAddr]
```

Parameters

`-t`

Specifies the target with `ipmbAddr`.

Example

```
hpmcmd -c ipmcstatus
IPMC Mode                =  NORMAL
Payload Control          =  Enabled
IPMC Outstanding Events  =  None
```

6.3.4.18 ledget

Description

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

Synopsis

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

Parameters

`fruid`

Is 0 for the ATCA-7490 Front board and 1 for the ARTM.

`led`

Is BLUE for the hot swap LED or LEDN for FRU LED<n>. <n> is a number between 1 and the maximum FRU LEDs supported by the blade.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
hpmcmd -c ledget 0 led1

Current State = OVERRIDE

State Function/(ms) Duration (ms)      Color

-------------------------------------------

Override         Off       Always        Red
```

6.3.4.19 ledprop

Description

This command displays the FRU LED properties under IPMC control.

Synopsis

```
ledprop <fruid>
```

Parameters

`fruid`

Is 0 for the ATCA-7490 Front board and 1 for the ARTM.

Example

```
hpmcmd -c ledprop 0

FRU LEDs under IPMC control:    LED0 = Blue, Default: Blue
LED1 = Amber,Red, Default: Red
LED2 = Amber,Green,Red Default: Green
LED3 = Amber, Default: Amber
```

6.3.4.20 ledset

Description

This command controls the override state of a specific FRU LED.

Synopsis

```
ledset <fruid> <led> <operation> [offms] [onms] [color] [-t
ipmbAddr[:mmcAddr]]
```

Parameters

`fruid`

Is 0 for the ATCA-7490 Front board and 1 for the ARTM.

`led`

Is BLUE for the hot swap LED or LEDN for FRU LED<n>. <n> is a number between 0 and the maximum FRU LEDs supported by the blade.

`operation`

ON = enable override state and turn LED on.

OFF = enable override state and turn LED off.

BLINK = enable override state and blink LED; off_duration and on_duration specify the blink duration; the default on and off duration is 300 ms.

LOCAL = cancel override state and restore LED control to the IPMC, that is, local state.

TEST = run lamp test for specified on_duration, then restore prior state. The default duration is 5000ms.

`offms`

Specifies OFF duration in milliseconds. It can have value from 10ms to 2500ms in the 10ms increments. It is valid only if the operation is BLINK.

`onms`

Specifies ON duration in milliseconds. It can have value from 10ms to 2500ms in the 10ms increments. It is valid only if the operation is BLINK.

`color`

LED0 = BLUE
LED1 = RED or AMBER
LED2 = GREEN (if supported by IPMC)
LED3 = AMBER (if supported by IPMC)

`-t ipmbAddr`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
hpmcmd -c ledset 0 led1 on
```

6.3.4.21 loglevelget

Description

This command retrieves the current `hpmagentd` log level. The log level of hpmcmd can be set with the environment variable, `HPMCMD_LOG_LEVEL`. For example, export HPMCMD_LOG_LEVEL=7 to set debug level. All log messages are sent to the syslog.

Log levels:
0 = Emergency
1 = Alert

2 = Critical
3 = Error
4 = Warning
5 = Notice
6 = Information
7 = Debug

Synopsis

```
loglevelget
```

Example

```
hpmcmd -c loglevelget

5
```

6.3.4.22 shelftype

Description

This command retrieves the shelf FRU (IPMB 20) Board Area Product Name (FRU 254).

Synopsis

shelftype

Example

```
hpmcmd -c shelftype

AXP-1440
```

6.3.4.23 macaddress

Description

This command retrieves the list of available MAC addresses.

Synopsis

```
macaddress [fruid]
```

Parameters

`Fruid`

Is 0 for the ATCA-7490 Front board and 1 for the ARTM.

Example

```
FrontBoard:
root@ATCA7490:~# hpmcmd -c macaddress 0
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:7a:f8
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:7a:f9
Base Interface             MAC Addr :  ec:9e:cd:11:7a:f6
Base Interface             MAC Addr :  ec:9e:cd:11:7a:f7
Fabric Interface           MAC Addr :  ec:9e:cd:11:7a:fa
Fabric Interface           MAC Addr :  ec:9e:cd:11:7a:fb
Fabric Interface           MAC Addr :  ec:9e:cd:11:7a:fc
Fabric Interface           MAC Addr :  ec:9e:cd:11:7a:fd
Serial over Lan Interface  MAC Addr :  ec:9e:cd:11:7a:fe
Serial over Lan Interface  MAC Addr :  ec:9e:cd:11:7a:ff
ARTM:
root@ATCA7490:~# hpmcmd -c macaddress 1
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f2
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f3
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f4
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f5
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f6
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f7
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f8
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f9…
```

6.3.4.24 Lancfgget

Description

Gets LAN configuration parameter

Synopsis

```
lancfgget <channel> [param]
```

Parameters

`channel`

channel number

`param`

auth-type-support

auth-type-enables

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-gateway-mac-addr

backup-gateway-addr

backup-gateway-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

Example

```
root@ATCA-7490:~# hpmcmd -c lancfgget 1
IP Address         : 192.168.25.10
Subnet Mask        : 255.255.0.0
Default Gateway    : 172.16.0.1
```

6.3.4.25 Lancfgset

Description

Sets LAN configuration parameter

Synopsis

```
lancfgset <channel> <param> <value>
```

Parameters

`channel`

channel number

Example

```
root@ATCA-7490:~# hpmcmd -c lancfgset 1 ip-addr 192.168.25.10
Successful lancfgset Operation
```

6.3.4.26 partnumber

Description

This command retrieves the part number (FRU 0) of the main blade.

Synopsis

```
partnumber
```

Example

```
hpmcmd -c partnumber
```

6.3.4.27 physlotnumber

Description

This command retrieves the physical slot number in which the blade is plugged in.

Synopsis

```
hpmcmd -c physlotnumber
```

6.3.4.28 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] [devid]
```

Parameters

`interface`

Valid values are:

```
BASE | FABRIC | UPDATE | AMC
```

`channel`

A number from 1 to the maximum number of channels supported for the interface. Node blades usually support 2 Base and 2 to 4 Fabric channels, and switch blades support 16 Base, 15 Fabric, and 1 Update channels.

`devid`

For AMC only: It is an on-Carrier device ID that identifies the on-Carrier device to which the desired channel is connected.

Example:

```
root@ATCA-7490:~# hpmcmd -c portget

  STATE   INTERFACE   CHANNEL   LINKTYPE   LINKEXT   GROUP   PORTS
-------------------------------------------------------------------
 ENABLED      BASE       1         BASE       0        0     0
 ENABLED      BASE       2         BASE       0        0     0
 ENABLED    FABRIC       1        ETHER       4        0     0,1,2,3
DISABLED    FABRIC       1     RESERVED       1        0     0,1,2,3
DISABLED    FABRIC       1        ETHER       3        0     0
 ENABLED    FABRIC       2        ETHER       4        0     0,1,2,3
DISABLED    FABRIC       2     RESERVED       1        0     0,1,2,3
DISABLED    FABRIC       2        ETHER       3        0     0
DISABLED    FABRIC       3     RESERVED       1        0     0,1,2,3
DISABLED    FABRIC       3        ETHER       4        0     0,1,2,3
DISABLED    FABRIC       3        ETHER       3        0     0
DISABLED    FABRIC       4     RESERVED       1        0     0,1,2,3
DISABLED    FABRIC       4        ETHER       4        0     0,1,2,3
DISABLED    FABRIC       4        ETHER       3        0     0
DISABLED    UPDATE       3          OEM       0        0     0
DISABLED    UPDATE       4          OEM       0        0     0
DISABLED    UPDATE       5          OEM       0        0     0
```

6.3.4.29 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 values are:

```
BASE | FABRIC | UPDATE | AMC
```

`chan`

It is a number from 1 to the maximum number of channels supported for the interface. Node blades usually support 2 Base and 2 Fabric channels, and switch blades support 16 Base, 15 Fabric, and 1 Update channels.

`grpid`

Specifies the group ID. Always 0 according to the current shelf FRU information.

`type`

Valid values are:

```
BASE | ETHER | EXPRESS | INFINI | STAR | OEM
```

`typeX`

Valid values are:
`0` (for 1000Base-BX)
`1` (for 10GBase-BX4)
`2` (for FC-PI)

`ports`

A sequence of ports to act on.
For base and update channels, port is always 0.
For fabric channels, port can specify up to 4 ports as specified in PICMG 3.1:
Option 1: 0 (for port 0)
Option 2: 01 (for ports 0,1)
Option 3: 0123 (for ports 0,1,2,3)
Option 7: 3 (for port 3)

`oper`

Valid values are `DISABLE` or `ENABLE`.

`devid`

For AMC only: It is an on-Carrier device ID that identifies the on-Carrier device to which the desired channel is connected.

`-t ipmbAddr`

Sends the command to `ipmbAddr:mmcAddr`.

Example

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

6.3.4.30 posttypeget

Description

This command retrieves the postType to which the board is currently set to run at boot time, for the specified CPU.

Synopsis

```
posttypeget <payload_cpu_selector>
```

Parameters

`payload_cpu_selector`

The specified CPU is set to postType to run. Is 0 for the GPP, 1 for the SPP.

Example

```
hpmcmd -c posttypeget 1
```

```
LONG
```

6.3.4.31 posttypeset

Description

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

Synopsis

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

Parameters

`payload_cpu_selector`

It is an integer between 0 and number of CPU devices supported per board.

`newPostType`

Valid values are: `SHORT | LONG`.

6.3.4.32 sdr

Description

This command shows the SDR records.

Synopsis

```
sdr
```

Example

```
root@ATCA7490:~# hpmcmd -c sdr

recID 1: management controller device locator record
     I2C slave addr:  44
    Channel number:  00
    Power state:     06
    Global init:     0C
    Capabilities:    2D
    Entity Id:       PICMG front board
    Entity instance: 60
    OEM:             00
    Id string:       ATCA-7490

recID 2: full sensor record
    owner is IPMB 88 sensor num 00 on lun 00 channel 00
    logical entity: PICMG front board - instance 60
    Hot Swap Carrier : FRU hot swap : sensor-specific discrete

recID 3: full sensor record
    owner is IPMB 88 sensor num 01 on lun 00 channel 00
    logical entity: RTM - instance 60
    Hot Swap RTM : FRU hot swap : sensor-specific discrete

recID 4: full sensor record
    owner is IPMB 88 sensor num 02 on lun 00 channel 00
    logical entity: PICMG front board - instance 60
    IPMB Physical : IPMC physical link : sensor-specific discrete

recID 5: full sensor record
    owner is IPMB 9C sensor num 06 on lun 00 channel 00
    logical entity: PICMG front board - instance 60
    Version change : 2B : sensor-specific discrete
```

6.3.4.33 sdr_dump

Description

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

Synopsis

```
sdr_dump
```

Example

```
root@ATCA7490:~# hpmcmd -c sdr_dump
SDR Records:
01 00 51 00 12 88 00 cc 2d 00 00 00 30 00 00 00  "..Q....Ì-...0..."
00 00 00 00 7c 00 00 00 00 00 00 00 00 00 00 00  "....|..........."
e0 06 02 00 00 00 00 00 02 00 00 00 00 00 00 00  "à..............."
```

6.3.4.34 sdrinfo

Description

This command shows the SDR information.

Synopsis

```
sdrinfo
```

Example

```
root@ATCA-7490:~# hpmcmd -c sdrinfo
SDR Information:
LUN 0 has 068 sensors; dynamic sensor population
LUN 1 has 000 sensors
LUN 2 has 000 sensors
LUN 3 has 000 sensors
```

6.3.4.35 sendamc

Description

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

Synopsis

```
sendamc <IPMBaddress> <MMCaddress> <netfn> <cmd> <data0> ...
<datan>
```

Parameters

`IPMBaddress`

Destination IPMB address in hex digits.

`MMCaddress`

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.

6.3.4.36 sendcmd

Description

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

Synopsis

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

Parameters

`IPMBaddress`

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 90 06 59
07 59 C1
```

6.3.4.37 shelfaddress

Description

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

Synopsis

```
shelfaddress
```

Example

```
hpmcmd -c shelfaddress
01
```

6.3.4.38 shelfslots

Description

This command retrieves the total number of blade slots in the shelf.

Synopsis

```
shelfslots
```

Example

```
hpmcmd -c shelfslots

14 slots
```

6.3.4.39 slotmap

Description

This command displays a slotmap table for the shelf where the blade is installed.

Synopsis

```
slotmap
```

Example

```
hpmcmd -c slotmap

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

6.3.4.40 slotnumber

Description

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

Synopsis

```
slotnumber Parameters
```

Example

```
hpmcmd -c slotnumber
4
```

6.3.4.41 sel

Description

Shows the SEL records.

Synopsis

`Sel`

Example

```
root@ATCA7490:~# hpmcmd -c sel
0x01A2: Event: at: Dec 31 19:00:00 1969; from:(0xee,0,0);
sensor:(0xd7,30); event:(0x6f,asserted): a0 65 00

0x01A3: Event: at: Dec 31 19:00:00 1969; from:(0xee,0,0);
sensor:(0xda,34); event:(0x6f,asserted): a0 01 00

0x01A4: Event: at: Dec 31 19:00:00 1969; from:(0xee,0,0);
sensor:(0x08,31); event:(0x6f,asserted): 00 ff ff

0x01A5: Event: at: Dec 31 19:00:00 1969; from:(0xee,0,0);
sensor:(0x08,32); event:(0x6f,asserted): 00 ff ff
```

6.3.4.42 selinfo

Description

Shows the SEL information.

Synopsis

`Sel`

Example

```
hpmcmd -c selinfo
```

```
root@ATCA7490:~# hpmcmd -c selinfo
SEL version: 1.5
Number of log entries: 1023
Free space: 0 bytes
Events have been dropped due to lack of space in the SEL
```

```
Last addition timestamp: Sep 22 12:06:55 2014
Last erase timestamp: Dec 31 19:00:00 1969
Supported operations:
 - Reserve command supported
```

6.3.4.43 selclear

Description

Erases all contents of the SEL.

Synopsis

`Selclear`

Example

`hpmcmd -c selclear`

6.3.4.44 serialoutputget

Description

Determines which serial output source goes to a particular serial port connector.

Synopsis

`serialoutputget <Serial Connector Type> <Instance Number>`

Parameters

`Serial Connector Type` -

0 Faceplate Connector

1 Backplane Connector

2 Onboard Connector

3 Onboard Device (route to another chipset)

`Instance Number` -

zero-based instance number

Example

```
hpmcmd -c serialoutputget 0 1

Serial Output Selector: 1
```

6.3.4.45 serialoutputset

Description

Select the serial port output source for a serial port connector.

Synopsis

```
serialoutputset <Serial Connector Type> <Instance Number> <Serial
Output Selector>
```

Parameters

`Serial Connector Type` -

0 Faceplate Connector

1 Backplane Connector

2 Onboard Connector

3 Onboard Device (route to another chipset)

`Instance Number` -

zero-based instance number

`Serial Output Selector` -

an integer value >= 0

6.3.4.46 solcfgget

Description

Gets SOL configuration parameter

Synopsis

```
solcfgget <channel> [param]
```

Parameters

`channel` - channel number

`param` -

enable

authentication

char-settings

retry

nonvolatile-bit-rate

volatile-bit-rate

payload-channel

payload-port

Example

```
root@ATCA7490:~# hpmcmd -c solcfgget 1
Enabled                              : true
Force Encryption                     : false
Force Authentication                 : false
Privilege Level                      : Admin
Character Accumulate Interval (ms): 5
Character Send Threshold             : 1
Retry Count                          : 1
Retry Interval                (ms): 50
Non-Volatile Bit Rate        (kps): 115.2
Volatile Bit Rate            (kps): 115.2
Payload Port                         : 623
```

6.3.4.47 solcfgset

Description

Sets SOL configuration parameter

Synopsis

```
solcfgset <channel> <param> <value>
```

where:

`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

6.3.4.48 version

Description

This command displays the version of the hpmcmd software.

Synopsis

```
version
```

Example

```
hpmcmd -c version
3.11.3
```

6.3.4.49 watchdog

Description

This command is used to 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`

Possible values are

Value	Description
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

7.1 Overview

To ease the implementation of highly available systems with off-the-shelf building blocks, the Service Availability Forum (SA Forum) and HPI-B 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.

You can get the HPI-B Software packages from local Artesyn sales representative. For more information on Artesyn Embedded Technologies - Embedded Computing's HPI-B implementation, refer to the *System Management Interface Based on HPI-B User's Guide.*

8.1 Overview

Board control is a Linux kernel module, which provides access to the Glue logic FPGA of the ATCA-7490 board. The `boardctrl` driver is installed during boot, which provides access to the BIOS version registers. After successful installation, an `init` script for loading/unloading the driver can be found at `/etc/init.d/boardctrl`.

The boardctrl driver is a prerequisite for the firmware upgrade tool FCU.

After loading the driver, you will find data provided at `/proc/boardinfo`, shown in the Table 8-1.

8.2 Board Control Tool

The board control module provides an IOCTL (input/output control) interface, which can be used by the userland applications. As an example, `fpgatool` can be used.

8.2.1 FPGATOOL

Description

This command allows you read/write to the FPGA register set.

Note: Some of the registers are write-protected and cannot be overwritten by userland applications.

FPGATOOL can be found at `/opt/bladeservices/bin/fpga_tool`.

Synopsis

```
root@ATCA7490:~# fpgatool

usage: fpgatool <command> <parameter> Available commands: read
write dump info
```

```
Example: fpgatool dump
Offset: 0x00    Value: 0x90 - 10010000 [Module Identification Register]
Offset: 0x01  Value: 0x74 - 01110100 [Module Identification Register]
Offset: 0x02  Value: 0x02 - 00000010 [FPGA Version Register]
Offset: 0x03  Value: 0x01 - 00000001 [Serial Redirection Control Register ]
Offset: 0x04  Value: 0x00 - 00000000 [Serial over LAN Control Register ]
Offset: 0x05  Value: 0x00 - 00000000 [Serial Line Routing Register]
Offset: 0x06  Value: 0x00 - 00000000 [IPMC Power Level Register]
Offset: 0x07  Value: 0x00 - 00000000 [IPMC Power Level Multiplier Register]
Offset: 0x08  Value: 0x00 - 00000000 [ME Power Failure State Register]
Offset: 0x09  Value: 0x00 - 00000000 [ME Power Failure Cause Register]
Offset: 0x0a  Value: 0x00 - 00000000 [Payload Power Failure State Register]
....
```

Table 8-1 FPGA Tool

File	Description	Sample output

`bios`	Shows the version and release date of the currently installed BIOS	`BIOS Vendor : ARTESYN`
		`BIOS Version : 0.4.0`
		`BIOS Release date: 03/07/2016`

`board`	Shows the board name/version as provided by the BIOS.	`Board Vendor: ARTESYN`
		`Board Name: ATCA-7490`
		`Board Version: R.1.0`
		`Board Serial Number: E168A86`

Table 8-1 FPGA Tool

File	Description	Sample output

`temp-Info`	Reads some temperature sensors (For example, CPU package temperature, maximum temperature of the DIMM channels, PCH package temperature.	`0xff:0x1e.0x00-0x00c8: Package_Temperature: 52 C`
		`0xff:0x1e.0x00-0x00cc: PPO_Temperature: 47 C`
		`0xff:0x1e.0x00-0x0060: MEM_TRML_Temperature:`
		`Channel0_Max_Temperature: 44 C`
		`Channel1_Max_Temperature: 0 C`
		`Channel2_Max_Temperature: 0 C`
		`Channel3_Max_Temperature: 0 C`
		`0x7f:0x1e.0x00-0x00c8: Package_Temperature: 49 C`
		`0x7f:0x1e.0x00-0x00cc: PPO_Temperature: 44 C`
		`0x7f:0x1e.0x00-0x0060: MEM_TRML_Temperature:`
		`Channel0_Max_Temperature: 0 C`
		`Channel1_Max_Temperature: 0 C`
		`Channel2_Max_Temperature: 0 C`
		`Channel3_Max_Temperature: 0 C`
		`0x00:0x1f.0x06-0x0010: Wellsburg Temperature: 42 C`

Table 8-1 FPGA Tool

File	Description	Sample output

`eccInfo`	Reads the Error counters of the 2-memory controller of the board	`Corrected Error Count per Rank:`
		`1 2`
		`3 4 5 6 7 8`
		`CPU0 - Channel0: DIMM01 0000 0000 0000 0000 - DIMM02 0000 0000 0000 0000`
		`CPU0 - Channel1: DIMM03 0000 0000 0000 0000 - DIMM04 0000 0000 0000 0000`
		`CPU0 - Channel2: DIMM05 0000 0000 0000 0000 - DIMM06 0000 0000 0000 0000`
		`CPU0 - Channel3: DIMM07 0000 0000 0000 0000 - DIMM08 0000 0000 0000 0000`
		`CPU1 - Channel0: DIMM09 0000 0000 0000 0000 - DIMM10 0000 0000 0000 0000`
		`CPU1 - Channel1: DIMM11 0000 0000 0000 0000 - DIMM12 0000 0000 0000 0000`
		`CPU1 - Channel2: DIMM13 0000 0000 0000 0000 - DIMM14 0000 0000 0000 0000`
		`CPU1 - Channel3: DIMM15 0000 0000 0000 0000 - DIMM16 0000 0000 0000 0000`

`fpga`	Shows some additional FPGA
	Information (CPU presence, CPU Type, and so on)
		`Board Module Version: 0x7490`
		`FPGA version: 0x02`
		`BIOS Reset Source: 0x00`
		`OS Reset Source: 0x41`
		`IPMC Reset Source: 0x00`
		`LED Control Register: 0x00`

Table 8-1 FPGA Tool

File	Description	Sample output
summary	Shows the summary of the board state (FPGA registers) and BIOS provided information	root@ATCA7490:~# cat /proc/boardinfo/fpga summary Board Vendor: ARTESYN Board Name: ATCA-7490 Board Version: R.1.0 Board Serial Number: E168A86 BIOS Vendor: ARTESYN BIOS Version: 0.4.0 BIOS Release Date: 03/07/2016 Memory Module: Device/Bank: J15/CPU 0 Size: 8192 Mbyte Data Width: 64 Bit Manufacturer: Micron Memory Module: Device/Bank: J21/CPU1 Size: 8192 Mbyte Data Width: 64 Bit Manufacturer: Micron IPMI Interface Type: 1 KCS (Keyboard Control Style) IPMI Spec Rev: 2.0 I2C Slave Addr: 0x90 NV Stor.Dev.Addr: Not Present Base Addr: 0x00000CA3 IRQ: 0x0

9.1 Kernel configuration

The ATCA-7490 uses the 64-bit kernel configuration that comes along with OS distribution. The kernel configuration file is usually stored in the boot directory of the host system.

9.1.1 Additional Kernel Patches

Table 9-1 provides the list of additional kernel patches, which were applied during the kernel build.

Table 9-1 Kernel Patches

Name	Host OS	Description
NA	CentOS7.1	No patches available.

9.1.2 Additional Kernel Drivers

The Intel fm10000 Switch devices requires a new network driver (fm10k), which is not included in the current OS release. As long as this driver is not available, you need to download, compile and install the driver by yourself, if you are not using the Artesyn provided BBS tools.

You can download the latest driver and function driver (required for virtualization) from: http://sourceforge.net/projects/e1000/files/fm10k%20stable/

Installing fm10k device driver

1. After downloading the driver, copy it to the driver package in the target system.
2. Extract the TAR file using `xzvf fm10k-<version>.tgz` command.
   In the above command, replace `<version>` with the version of the device driver.
3. Go to `fm10k-<version>/src/` directory and run `make install` command to install the driver.

10.1 RTM-ATCA-749X Overview

The RTM-ATCA-749X blade is available in two variants.

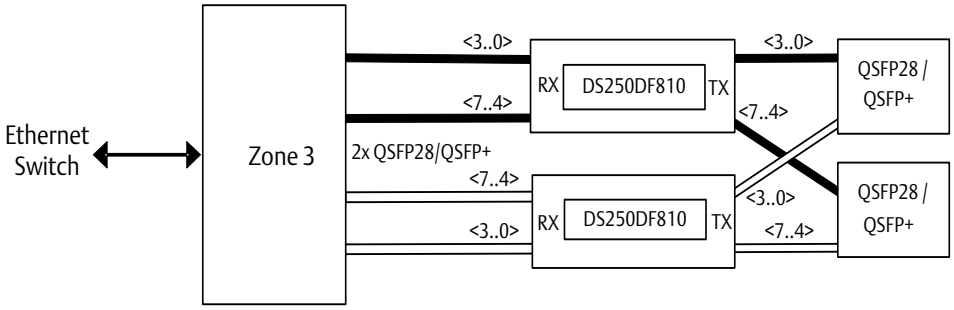
- RTM-ATCA-749X-100G
  - Two 100Gb Ethernet interfaces via QSFP28 pluggable transceivers
  - Six 25Gb Ethernet interfaces via SFP28 pluggable transceivers
- RTM-ATCA-749X-100GHA
  - Two 100Gb Ethernet interfaces via QSFP28 pluggable transceivers
  - Six 25Gb Ethernet interfaces via SFP28 pluggable transceivers
  - Two Intel Communications Chipset 8900 Series PCH with Integrated Accelerator

For more information about RTM-ATCA-749X, refer to *RTM-ATCA-749X Installation and Use* manual.

10.2 RTM-ATCA-749X-100G / RTM-ATCA-749X-100GHA Interfaces

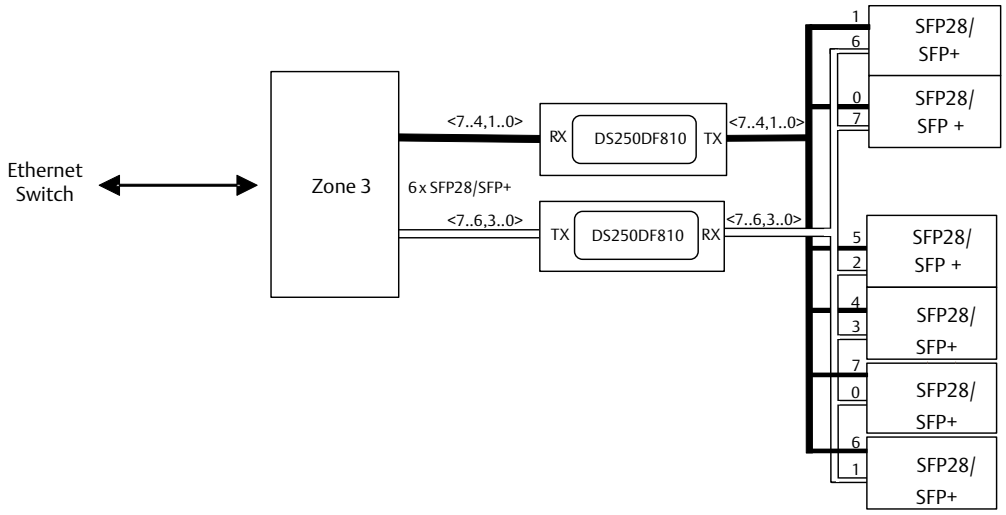
The RTM-ATCA-749X-100Gxx provides two QSFP28 cages to support two Ethernet 40G/100G interfaces. The Ethernet 40GbE/100GbE interfaces are connected via two Texas Instruments DS250DF810 25Gbps Multi-Rate 8-Channel Retimer to the FM10K on the ATCA-7490 and support 100GBase-CR4 according IEEE Clause 92, 100GBase-SR4 according IEEE Clause 7283E, 40GBase-CR4 according IEEE Clause 85 and 40GBase-SR4 according IEEE Clause 86A. The RTM-ATCA-749X-100Gxx does not support polarization-mode dispersion (PMD) control function as defined in IEEE 802.3 Chapter 72.6.10. The training frame structure used by the 100GBASE-CR4 and 10GBASE-CR4 PMD control function is not supported from the Retimer.

Figure 10-1 QSFP28/QSFP+ Interfaces



The RTM-ATCA-749X-100Gxx provides a dual and a quad SFP28/SFP+ cage to support six Ethernet 25GbE/10GbE interfaces. The Ethernet interfaces are connected via two Texas Instruments DS250DF810 25Gbps Multi-Rate 8-Channel Retimer to the FM10K and support 10GBase-CR according SFF-8431 and 10GBase-SR according SFF-843. The 25GbE is a non-standard and electrically defined as a single lane of corresponding 100GbE specification. The RTM-ATCA-749X-100Gxx does not support PMD control function as defined in IEEE 802.3 Chapter 72.6.10. The training frame structure used by the 10GBASE-CR4 PMD control function is not supported from the Retimer.

Figure 10-2 SFP28/SFP+ Interfaces

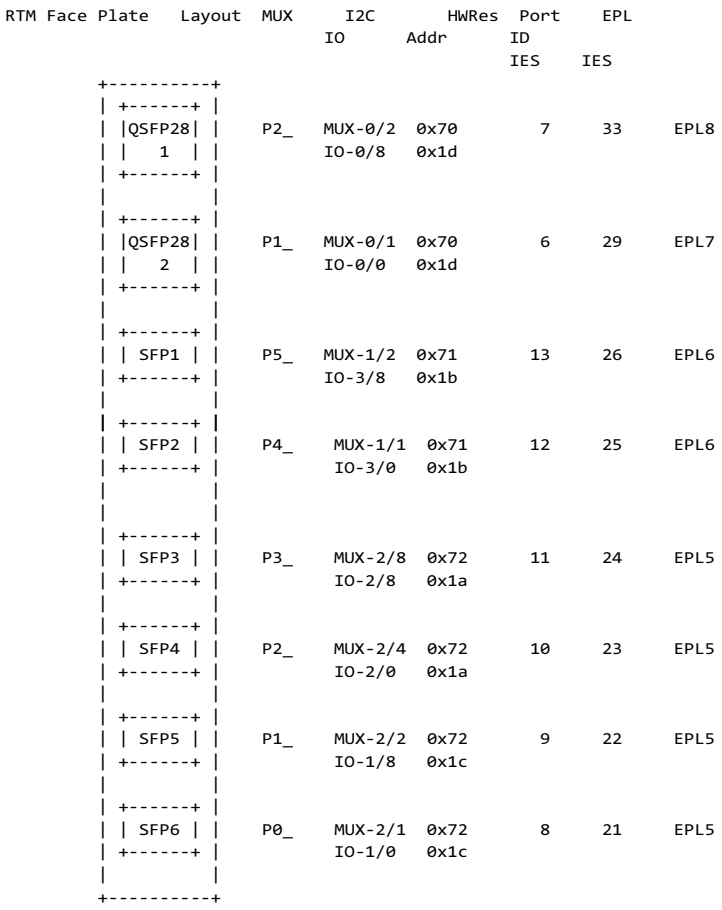


For more information about RTM-ATCA-749X interfaces, refer to *RTM-ATCA-749X Installation and Use* manual.

10.2.1 QSFP28/QSFP+/SFP28/SFP+ Interfaces Configuration

The following figure shows the faceplate of the RTM-ATCA-7490 with the SFP+/SFP28/QSFP28 ports. MUX and I2C Address are used to access the SFP devices via the I2C bus. The I2C bus master is the FM10k device. The HWRes Id and the port number are used by the IES Switch software to access/identify the ports. This switch port configuration, Retimer configuration is available in the IES configuration file at */etc/ies/config/ltrr/fm_platform_attributes_rtm.cfg*. The I2C bus structure of the RTM is shown in the Appendix. Switch port and VLAN configuration can be available at */opt/bladeservices/etc/fmShell_master.scr*.

Figure 10-3 RTM-ATCA-749X Faceplate



The RTM interfaces are assigned as shown in the following table. A QSFP28/QSFP+ requires four switch ports when they are configured in combined mode. For example,100GBase-SR4. One could also configure 4 x 10GBase-SR/25GBase-SR ports, for example, to support a 40G or 100G SFP+/SFP28 Parallel Fan-Out Active Optical Cable.

Table 10-1 QSFP28/QSFP+/SFP28/SFP+ Configuration

RTM Faceplate Number	Device	HWResId	PortIdx	Port Number	Description	VLAN assignment
QSFP28 – 1	QSFP28/QSFP+	7	33	33.0	100Gb/s	enp131s0.32
			34	33.1		
			35	33.2		
			36	33.4		
QSFP28 – 2	QSFP28/QSFP+	6	29	29.0	100Gb/s	enp7s0.31
			30	29.1		
			31	29.2		
			32	29.3		
SFP1	SFP28/SFP+	13	26	26	25Gb/s	enp7s0.71
SFP2	SFP28/SFP+	12	25	25	25Gb/s	enp131s0.72
SFP3	SFP28/SFP+	11	24	24	10Gb/s	enp5s0.73
SFP4	SFP28/SFP+	10	23	23	10Gb/s	enp129s0.74
SFP5	SFP28/SFP+	9	23	23	10Gb/s	enp7s0.75
SFP6	SFP28/SFP+	8	21	21	10Gb/s	enp131s0.76

10.2.2 DS250DF810 Retimer Configuration on RTM-7490

Four Texas Instruments DS250DF810 Retimer devices of are used on RTM-7490. The retimer devices are assigned as shown in the following table. The configuration of the retimer devices is done in the IES configuration file of the FM10k device (See */etc/ies/config/ltrr/fm_platform_attributes_rtm.cfg*). Within the IES configuration file the network ports are usually configured as optical or as copper devices dependent on the SFP+/SFP28/QSFP which are being used..

On RTM-7490, all SFP+/SFP28/QSFP+/QSFP28-ports must be configured as optical interfaces, because the retimer locks the signal and the FM10K must not change its settings, as it would normally do if copper interfaces are configured.

Table 10-2 DS250DF810 Retimer Configuration on RTM-7490

Speed	Direction	Retimer Number
2 x 4 lanes a 10Gbit/s or 25GBit/s	Egress Mode (FM10K → QSFP28)	0
2 x 4 lanes a 10Gbit/s or 25GBit/s	Ingress Mode (QSFP28 → FM10K)	1
8 independent lanes a 10Gbit/s or 25Gbit/s	Egress Mode (FM10K → SFP/SFP28)	2
8 independent lanes a 10Gbit/s or 25Gbit/s	Ingress Mode (SFP/SFP28 → FM10K)	3

10.2.3 DS250DF810 Retimer Channel Assignments

Each retimer supports 8 independent channels. This channel assignment is covered in the IES software and it is usually necessary only for debugging purpose.

Table 10-3 DS250DF810 Retimer Channel Assignments

PortIdx	Port	Retimer Egress		Retimer Ingress		
		Retimer Number	Channel	Retimer Number	Channel	Name on Faceplate
21		2	6	3	1	SFP6
22		2	7	3	0	SFP5
23		2	4	3	3	SFP4
24		2	5	3	2	SFP3
25		2	0	3	7	SFP2
26		2	1	3	6	SFP1
	NC	2	2	3	4	Not connected
	NC	2	3	3	5	Not connected
29	29.0	0	6	1	4	QSFP28_2 RX/TX0

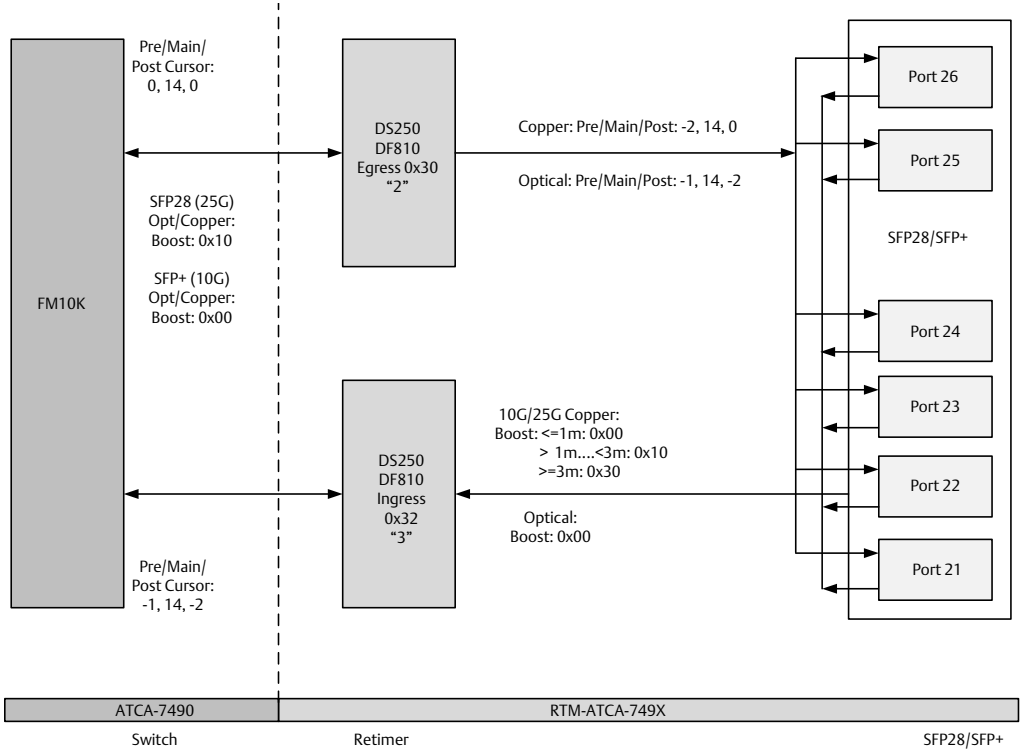
Table 10-3 DS250DF810 Retimer Channel Assignments

PortIdx	Port	Retimer Egress		Retimer Ingress		
		Retimer Number	Channel	Retimer Number	Channel	Name on Faceplate
30	29.1	0	7	1	5	QSFP28_2 RX/TX1
31	29.2	0	4	1	3	QSFP28_2 RX/TX2
32	29.3	0	5	1	2	QSFP28_2 RX/TX3
33	33.0	0	2	1	5	QSFP28_1 RX/TX0
34	33.1	0	3	1	4	QSFP28_1 RX/TX1
35	33.2	0	0	1	7	QSFP28_1 RX/TX2
36	33.3	0	1	1	6	QSFP28_1 RX/TX3

10.2.4 SFP28/SFP+ Retimer Configuration

The following figure gives an overview on how the SFP+/SFP28 interface ports are connected via the retimer to the FM10K devices. Additionally, the basic retimer settings are included, which are automatically chosen dependent on the configuration in the FM10K device and the device/cable type on the connectors of the SFP+/SFP28 ports.

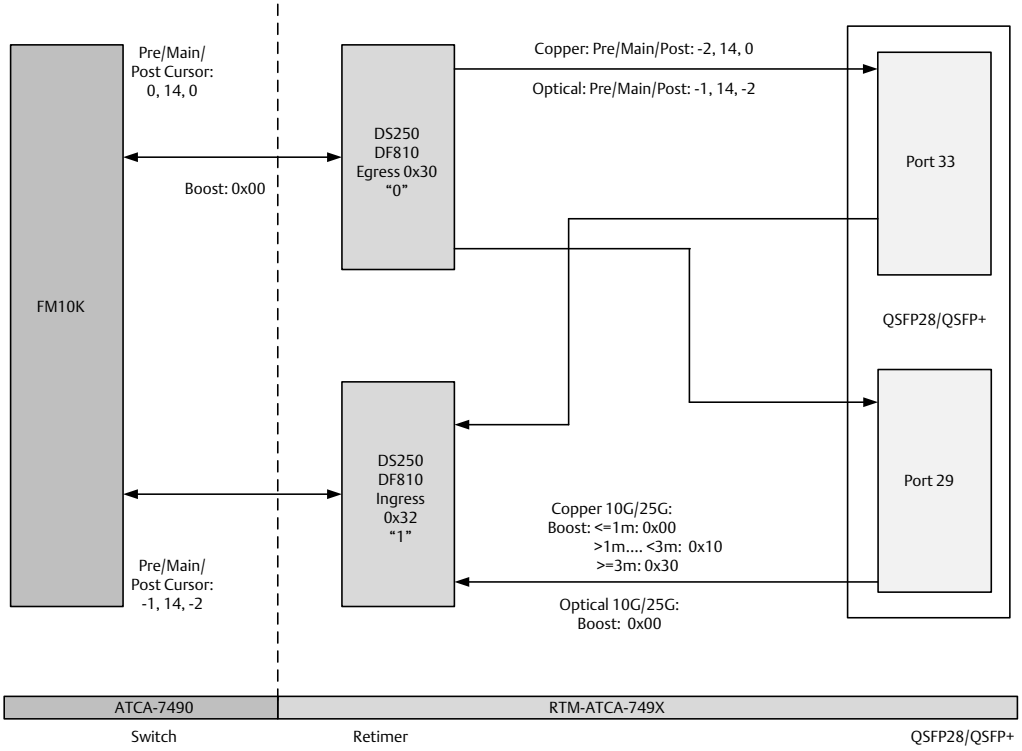
Figure 10-4 SFP28/SFP+ Retimer Configuration



10.2.5 QSFP+/QSFP28 Retimer Configuration

The following figure provides an overview on how the QSFP+/QSFP28 interface ports are connected via the retimer to the FM10K devices. Additionally, the basic retimer settings are included, which are automatically chosen dependent on the configuration in the FM10K device and the device/cable type on the connectors of the QSFP+/QSFP28 ports.

Figure 10-5 QSFP+/QSFP28 Retimer Configuration



When the SFP+/SFP28/QSFP+/QSFP28 device is plugged into the cage or the switch software is restarted the IES software (**fmPlatformRetimerConfigSet**) configures the retimer devices according to the device information and the FM10K configuration. The following settings of the retimer are set:

- Speed
- Pre-/Main-/Post-Cursor dependent on the attached devices (copper/optical /speed)
- Boost value

10.2.6 DS250DF810 Retimer Channel configuration

10.2.6.1 DS250DF810 Retimer Set Configuration

This command sets the retimer configuration according to the RRC configuration file and the detected SFP+/SFP28/QSFP+/QSFP28 device information. This command will also be called when the FM10K switch software is initialized or when SFP+/SFP28/QSFP+/QSFP28 devices are plugged into the system.

```
fmShell -Ls -- -e "fmPlatformRetimerConfigSet 0,<port>"
```

For example, if the port number is 29:

```
root@ATCA7490:~# fmShell -Ls -- -e "fmPlatformRetimerConfigSet
0,29"

Connect to host driver using /dev/uio0
Result: Int32: 0 = 0 = 0x00000000
```

10.2.6.2 DS250DF810 Retimer Configuration check

With this command, you can check the configuration data and the transceiver information used for the retimer configuration.

```
fmShell -Ls -- -e "fmPlatformRetimerConfigCheck 0,<port>"
```

In the example below, an optical 40G QSFP is located in port 29. If the FM10K configuration does not fit to the transceiver found in the specified port, a warning is issued.

```
root@ATCA7490:~# fmShell -Ls -- -e "fmPlatformRetimerConfigCheck
0,29"

Connect to host driver using /dev/uio0
RRC config:       [0.29] hwResId: 6 intfType: QSFP_LANE0, speed:
40000
Retimer config:   phyNum: 0, phyPort: 6 lanes: (6,1)
Port Status:      link:   UP speed:  40G
XCVR              present, type: XCVR_TYPE_40G_QSFP_OPT, length: 0
m,  optical yes
Result: Int32: 0 = 0 = 0x00000000
```

10.2.6.3 Signal Detection on Retimer

This command can be used to check if a signal is detected and if the retimer is locked for the specified retimer channels.

```
fmShell -Ls -- -e "fmPlatformRetimerDumpInfo
0,<retimerId>,\"status\""
```

Or

```
fmShell -Ls -- -e "fmPlatformRetimerGetSignalStatus 0,<retimerId>,-
1"
```

For example,

```
fmShell -Ls -- -e "fmPlatformRetimerDumpInfo 0,0,\"status\""

Connect to host driver using /dev/uio0
sw: 0, phyIdx: 0, channel: 0 SignalDetect: 1 CDRLock: 1 speed:
25.78125 Gbps
sw: 0, phyIdx: 0, channel: 1 SignalDetect: 1 CDRLock: 1 speed:
25.78125 Gbps
sw: 0, phyIdx: 0, channel: 2 SignalDetect: 1 CDRLock: 1 speed:
25.78125 Gbps
sw: 0, phyIdx: 0, channel: 3 SignalDetect: 1 CDRLock: 1 speed:
25.78125 Gbps
sw: 0, phyIdx: 0, channel: 4 SignalDetect: 1 CDRLock: 1 speed:
10.3125 Gbps
sw: 0, phyIdx: 0, channel: 5 SignalDetect: 1 CDRLock: 1 speed:
10.3125 Gbps
```

```
sw: 0, phyIdx: 0, channel: 6 SignalDetect: 1 CDRLock: 1 speed:
10.3125 Gbps
sw: 0, phyIdx: 0, channel: 7 SignalDetect: 1 CDRLock: 1 speed:
10.3125 Gbps
Result: Int32: 0 = 0 = 0x00000000
```

10.2.7 Manual Adaptation of the Retimer Channel settings

There are different sets of default settings for the retimer channels included in the ATCA-7490 specific part of the IES API Switch software. If it turns out that these default settings are not working with the used SFP28/QSFP+/QSFP28 devices/cables one can override these settings in the FM10K configuration file. During switch start these parameters are read and will be used instead of the default ones.

For Ingress Retimer:

```
api.platform.config.switch.0.portIndex.<port>.retimerIngressAdaptM
ode  int   1
api.platform.config.switch.0.portIndex.<port>.retimerIngressPreCur
sor  int   -1
api.platform.config.switch.0.portIndex.<port>.retimerIngressMainCu
rsor int   14
api.platform.config.switch.0.portIndex.<port>.retimerIngressPostCu
rsor  int   -2
#for QSFP28_DAC > 1 m and  < 3 m length use boost value of 0x10
#for QSFP28_DAC >= 3 m  length use boost value of 0xc0
api.platform.config.switch.0.portIndex.<port>.retimerIngressBoostV
alueint 0x0
api.platform.config.switch.0.portIndex.<port>.retimerIngressBoostO
verride bool TRUE
api.platform.config.switch.0.portIndex.<port>.retimerIngressEqHiGa
in      bool TRUE
```

For Egress Retimer:

```
api.platform.config.switch.0.portIndex.<port>.retimerEgressAdaptMo
de  int   1
api.platform.config.switch.0.portIndex.<port>.retimerEgressPreCurs
or  int   -2
```

```
api.platform.config.switch.0.portIndex.<port>.retimerEgressMainCur
sor int   14
api.platform.config.switch.0.portIndex.<port>.retimerEgressPostCur
sor  int   0
api.platform.config.switch.0.portIndex.<port>.retimerEgressBoostVa
lueint 0x0
api.platform.config.switch.0.portIndex.<port>.retimerEgressBoostOv
erride bool TRUE
api.platform.config.switch.0.portIndex.<port>.retimerEgressEqHiGai
n      bool TRUE
```

All parameters must be defined for one direction. Otherwise, you will end up with a mixture of defined and default parameter settings. The parameter settings can be checked using the `fmPlatformRetimerShowConfig` command. Additionally the eye opening on per channel base is an indication for the quality of the signal. The higher and wider the eye opening the better the signal.

10.2.8 Retimer Debugging Commands

10.2.8.1 DS250DF810 Retimer Eye Opening per Channel

```
fmShell -Ls -- -e "fmPlatformRetimerDbgShowEyeOpeing 0,<retimerId>"
```

For example, for retimer id 0,

```
root@ATCA7490:~# fmShell -Ls -- -e
"fmPlatformRetimerDbgShowEyeOpening 0,0"

Connect to host driver using /dev/uio0

In the following the first 4 channels are configured in 25Gbit/s
mode the next 4 channels are configured as 10Gbit/s mode.

--- PHY: 0 hwResourceId: 8 phyCfg->addr: 0x18 ---
Channel: 0 0.500 243.8
Channel: 1 0.562 256.2
Channel: 2 0.531 237.5
Channel: 3 0.625 287.5
```

```
Channel: 4 0.844 612.5
Channel: 5 0.812 612.5
Channel: 6 0.812 625.0
Channel: 7 0.844 637.5
Result: Int32: 0 = 0 = 0x00000000
```

The result shows the eye width and eye height. For 25Gbit/s lanes the eye width should be between >= 0.46 and >=200 in order to have a good signal quality and low error rates.

10.2.8.2 DS250DF810 Retimer Eye Diagram per Port

The following command prints the eye diagram of the given DS250DF810 Retimer port.

For example,

fmShell> fmPlatformRetimerDumpInfo 0,3,"eyeDump","0"

```
phyIdx: 3 hwResourceId: 11, phyCfg->addr: 0x19 channel: 0
vertical eye range: using +/-400mV...
```

10.2.8.3 DS250DF810 Retimer Lane Configuration

```
fmPlatformRetimerDumpInfo 0,<retimerId>,"config"
Or
fmShell -Ls -- -e "fmPlatformRetimerShowConfig 0,$phy"

fmShell -Ls -- -e "fmPlatformRetimerDumpInfo 0,0,\"config\""
```

root@ATCA7490:~# **fmShell -Ls -- -e "fmPlatformRetimerDumpInfo 0,0,\"config\""**

```
Connect to host driver using /dev/uio0
----- RETIMER: 0 -----
phyIdx: 0 hwResourceId: 8, phyCfg->addr: 0x18
channel: 0
   adaptMode: 1 CTLE Boost Override: Enabled CTLE Boost: 0x00
   DFE: disabled DFE TAPs: disabled
      DFE Taps: -03 +00 +00 +00 +00
   Pre-/Post Cursor FIR: Enabled
```

```
        PreCursor: -1 MainCursor: +14 PostCursor: -2
    SignalDetect: 1 CDRLock: 1 speed: 25.78125 Gbps
    EQGain: 1 VGA_SEL_Gain: 0
    0x12: 0x83 0x1e: 0xe9 0x23: 0x40 0x71: 0x23

channel: 1

    adaptMode: 1 CTLE Boost Override: Enabled CTLE Boost: 0x00
    DFE: disabled DFE TAPs: disabled
       DFE Taps: -03 +00 +00 +00 +00
    Pre-/Post Cursor FIR: Enabled
        PreCursor: -1 MainCursor: +14 PostCursor: -2
    SignalDetect: 1 CDRLock: 1 speed: 25.78125 Gbps
    EQGain: 1 VGA_SEL_Gain: 0
    0x12: 0x83 0x1e: 0xe9 0x23: 0x40 0x71: 0x23
```

10.3 RTM-ATCA-749X-100GHA-DH895x – Acceleration devices

The RTM-ATCA-7490-100GHA is equipped with two optional Intel DH895x (ColetoCreek) Acceleration devices. These devices can be used for Data compression/decompression and cryptographic tasks.

10.4 Software Overview

Intel provides software for DH895x Acceleration devices, which includes drivers and acceleration code that runs on the Intel® architecture (IA) CPUs and on the accelerator devices. The software development package can be downloaded from the Intel website. For other Linux distributions, the QAT package must be compiled and installed manually.

10.5 The Intel Acceleration Software Subsystem

A subsystem (provided by Intel), which includes the software components that provide acceleration to applications running on the PCH. It contains:

- Services (Cryptographic, Data Compression)
  Includes the firmware that drives the various workload slices in the accelerator(s), and the associated Intel® architecture Service libraries that expose these workloads via Application Program Interfaces (API). These service libraries use the Acceleration Driver Framework (ADF) to plug into the OS and gain access to the hardware to communicate with the firmware.
- Intel® QuickAssist Technology APIs
  The Intel® QuickAssist Technology APIs provide service level interfaces for customer applications or Ecosystem Middleware to access the accelerator(s) in the PCH. More detail on the APIs and associated architecture are found in the Intel QuickAssist Technology API Programmer's Guide (https://01.org/sites/default/files/page/330684-001us_api_pg.pdf).
- Acceleration Driver Framework (ADF)
  The Acceleration Driver Framework (ADF) includes infrastructure libraries that provide various services to the different software components of the acceleration drivers. The software framework is used to provide the acceleration services API to the application. A configuration file enables customization of system operation.
  Note: For more information see, https://01.org/packet-processing/intel%C2%AE-quickassist-technology-drivers-and-patches.

The detailed information on the Intel DH8955 APIs and Software can be found at www.intel.com.

10.6 QuickAssist Software on the ATCA-7490 board

The BBS for ATCA-7490 comes with precompiled sample code from Intel for demonstration purpose.

Before installation of the RPM packages in the Root File system (For example, during network boot or first boot from disk) the version of the DH895x devices is read. Depending on the version information, the suitable RPM is installed.

When the ATCA-7490 blade is up and running, you can get version information as follows:

```
root@ATCA7490:~# /etc/init.d/accDevInfo.sh
##############################################
#    Intel DH8955 Accelerator Device Info  #
##############################################
   DeviceSteppingSKU
----------------------------------------
   09:00.0 A0 SKU3 or 4
   85:00.0 A0 SKU3 or 4
```

10.7 Running the QuickAssist Demo Software

10.7.1 Grub/netboot configuration

The `grub- or netboot` configuration file must be enhanced with the following parameters:

```
acpi_enforce_resources=lax intel_iommu=off
```

10.7.2 Installation of the QuickAssist Demo Software

To run the QuickAssist Demo software, install the `bbs-qat-atca7490-_<Version>-<OS distro> rpm-`package. This is usually done automatically during board startup. The package installs the required kernel drivers, demo applications, firmware, and test suites.

10.7.3 Start of the QuickAssist Demo Software

After installation of the RPM package, the QuickAssist service can be started using the **`qat_service start`** command.

```
root@ATCA7490:~# /etc/init.d/qat_service start
alg: No test for __gcm-aes-aesni (__driver-gcm-aes-aesni)
qat_1_6_adf 0000:09:00.0: PCI INT A -> GSI 42 (level, low) -> IRQ 42
qat_1_6_adf 0000:09:00.0: setting latency timer to 64
qat_1_6_adf 0000:85:00.0: PCI INT A -> GSI 64 (level, low) -> IRQ 64
qat_1_6_adf 0000:85:00.0: setting latency timer to 64
Loading SAL Module ...
qat_mem_module_init: get dynamic major 238
Loading QAE MEM Module ...
Processing file: /etc/dh895xcc_qReading config file.
a_dev0.conf
Starting acceleration device icp_dev0.
Resetting device icp_dev0
qat_1_6_adf 0000:09:00.0: irq 623 for MSI/MSI-X
qat_1_6_adf 0000:09:00.0: firmware: requesting
dh895xcc/mof_firmware.bin
qat_1_6_adf 0000:09:00.0: firmware: requesting
```

```
dh895xcc/mmp_firmware.bin
Started AE 0
....
Started AE 11
Processing file:Reading config file.
 /etc/dh895xcc_qStarting acceleration device icp_dev1.
Resetting device icp_dev1
a_dev1.conf
qat_1_6_adf 0000:85:00.0: irq 656 for MSI/MSI-X
qat_1_6_adf 0000:85:00.0: firmware: requesting
dh895xcc/mof_firmware.bin
qat_1_6_adf 0000:85:00.0: firmware: requesting
dh895xcc/mmp_firmware.bin
Started AE 0
...
Started AE 11
There is 2 acceleration device(s) in the system:
 icp_dev0 - type=dh895xcc, inst_id=0, node_id=0,  bdf=09:00:0,
#accel=6, #enginp
 icp_dev1 - type=dh895xcc, inst_id=1, node_id=1,  bdf=85:00:0,
#accel=6, #enginp
root@ATCA7490:~#

root@ATCA7490:~# cpa_sample_code signOfLife=1
icp_sal_userStartMultiProcess("SSL") started
qaeMemInit started
*** QA version information ***
device ID= 0
firmware = 2.1.0
mmp      = 1.0.0
software = 2.1.0
hardware = A0
*** END QA version information ***
*** QA version information ***
device ID= 1
firmware = 2.1.0
mmp      = 1.0.0
software = 2.1.0
```

```
hardware = A0
*** END QA version information ***

....
Inst 0, Affin: 0, Dev: 0, Accel 0, EE 0, BDF 09:00:00
Inst 1, Affin: 2, Dev: 0, Accel 0, EE 0, BDF 09:00:00
Inst 2, Affin: 1, Dev: 1, Accel 0, EE 0, BDF 85:00:00
Inst 3, Affin: 3, Dev: 1, Accel 0, EE 0, BDF 85:00:00
--------------------------------------
API                     Data_Plane
Session State           STATELESS
Algorithm               DEFLATE
Huffman Type            STATIC
Mode                    ASYNCHRONOUS
Direction               DECOMPRESS
Packet Size             8192
Compression Level       3
Corpus                  CALGARY_CORPUS
Number of threads       4
Total Responses         1584
Total Retries           3976
Clock Cycles Start      3572122098612
Clock Cycles End        3572126921442
Total Cycles            4822830
CPU Frequency(kHz)      1796022
Throughput(Mbps)        51904
Compression Ratio       0.51
--------------------------------------
```

Now, several tests are executed on both the DH895x devices on the Accelerator module.

The detailed information on Intel QuickAssist Technology can be found in:

- *Intel Communications Chipset 89xx Series Software Release Notes*
- *Intel Communications Chipset 89xx Series Software Programmer's Guide*
- *Intel QuickAssist Technology API Programmer's Guide*

- *Intel QuickAssist Technology Cryptographic API Reference Manual*
- *Intel QuickAssist Technology Data Compression API Reference Manual*

A Unattended Installation with CentOS 7.1

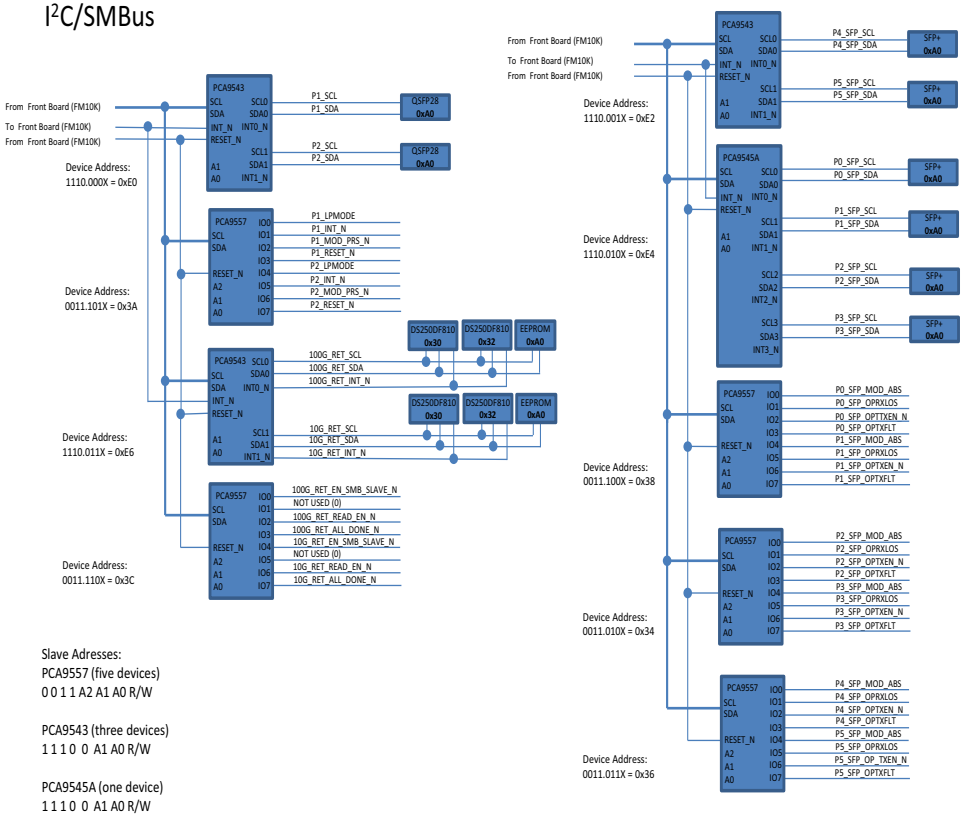
A.1 Kickstart File for Unattended CentOS7.1 Installation

A kickstart file for an unattended installation of CentOS7.1 is included in the BBS Software release (see for *scripts/anaconda-ks-centos71.115200.sda.bootif* (SCML41068)).

B.1 I2C Bus Structure on RTM-7490

This figure shows the I2C bus structure of the RTM-7490. The I2C Bus master is the FM10K device. This I2C Bus structure is reflected in the FM10K configuration for the RTM-749X.

Figure B-1 I2C Bus Structure on RTM-7490



C.1 Artesyn Embedded Technologies - Embedded Computing Documentation

The publications listed below are referenced in this manual. You can obtain electronic copies of Artesyn Embedded Technologies - Embedded Computing publications by contacting your local Artesyn sales office. For released products, you can also visit our Web site for the latest copies of our product documentation.

1. Go to www.artesyn.com/computing/support/product/technical-documentation.php.
2. Under FILTER OPTIONS, click the Document types drop-down list box to select the type of document you are looking for.
3. In the **Search** text box, type the product name and click GO.

Table C-1 Artesyn Embedded Technologies - Embedded Computing Publications

Document Title	Publication Number
ATCA-7490 Installation and Use Guide	6806800U11
ATCA-7490 Quick Start Guide	6806800U13
RTM-ATCA-749X Installation and Use Guide	6806800U12

ARTESYN™
EMBEDDED TECHNOLOGIES