

## Release Notes

### Schroff Firmware Part Number 63998-09251 Power Entry Module (PEM) for ZR5ATC6 Chassis

**Release Date: September 2, 2009**

#### **Release History:**

Release Date	Schroff Version	PPS version
September 2, 2009	Revision 51, Build 5122	V1.5.1

#### **Introduction:**

The firmware package "63998-09251.zip" contains firmware release Revision 51; build 5122 for the PEM. This IPMC firmware is based on PPS V1.5.1 BMR-AVR-ATCA IPMC firmware and customized for the Schroff PEM.

#### **Important Notice:**



The PEM IPMC firmware is comprised of two parts, the bootloader and the application. There is a bug in the V1.4.5 bootloader that was used in older firmware versions that can prevent the firmware upgrade from completing when using shelf managers that are not based on PPS firmware. The bootloader is not upgraded during the firmware upgrade process. Currently, the only way to upgrade the V1.4.5 bootloader is to use an Atmel JTAG ICE tool.

#### **Contents:**

1	Preparation .....	2
2	Upgrading the PEM to V1.5.1, Revision 51, Build 5122 .....	3
3	Updating the PEM FRU data with a PPS based Shelf Manager.....	5
4	Revision History .....	6
4.1	Schroff Specific Firmware Changes.....	6
4.2	PPS V1.5.1 New Features .....	6
4.3	Firmware Improvements .....	7
4.4	Bug Fixes .....	8

## 1 Preparation

### Firmware files

Unzip the 63998-09251.zip file. The ZIP file contains the following files:

<u>Upgradefw Upgrade file:</u>	
63998-09251.upgrade.img	Upgradefw upgrade image for installing firmware
<u>JTAG Upgrade files:</u>	
63998-09251.install1.hex	JTAG upgrade image for installing generic FRU data
63998-09251.install2.hex	JTAG upgrade image for installing firmware
<u>FRU Data file:</u>	
63998-09251.fru-info.bin	Generic PEM FRU data
<u>Release Notes file:</u>	
63998-09251(V1.5.1).pdf	These release notes
<u>Upgrade Tools:</u>	
upgradefw.exe	Firmware upgrade tool for Windows
cygwin1.dll	Upgradefw DLL file for Windows
upgradefw	Firmware upgrade tool for Linux
upgradefw	Firmware upgrade tool for the Shelf Manager

The table below lists valid values for <IPMC> and <FRU ID> for a 6-slot Schroff chassis.

<IPMC Address >	<FRU ID>	Description
66	0	Right PEM
68	0	Left PEM

## 2 Upgrading the PEM to V1.5.1, Revision 51, Build 5122

The PEM firmware is normally upgraded by using the upgradefw tool. There are three versions of upgradefw; for Windows PC, for Linux PC, and for shelf managers based on PPS firmware.

Example 1 shows how to perform the firmware upgrade from a Windows based PC that has a network connection to the shelf manager in the chassis.

Example 2 shows how to perform the firmware upgrade from a Linux based PC that has a network connection to the shelf manager in the chassis.

Example 3 shows how to perform the firmware upgrade from the shelf manager in the Schroff chassis.

### Example 1:

Zephyr chassis with the Windows upgradefw utility and a network connection to the Shelf Manager:

1. Unzip the 63998-09251.zip file.
2. Move the upgradefw.exe utility, the cygwin1.dll file, and the 63998-09251.upgrade.img firmware file to the same directory on a PC running Windows.
3. Insure that you have a network connection to the Shelf Manager in the Schroff chassis using the "ping" command.
4. Execute the upgradefw utility in a Command window for each PEM.

This text below shows the procedure for programming the right PEM. You need to substitute "192.168.0.2" in the command for the IP address of the Shelf Manager in your chassis. The "66" parameter in the command is the IPMB address for the PEM, either 66 or 68.

```
C:\>upgradefw -n 192.168.0.2,66 63998-09251.upgrade.img
BMR firmware upgrade utility. Pigeon Point Systems (c) 2004-2007.
Upgrade method: LAN/IPMB
Host           : 192.168.0.2
Port           : 623
Target IPMC    : 0x66
```

```
Opened an RMCP session (ID = 00000001).
Opening upgrade session ... OK
Firmware upgrade image: 63998-09251.upgrade.img
Preparing Master MCU for programming .....OK
Programming 49084 bytes to Master MCU at 000000 ... 100%
Preparing Slave MCU for programming ...OK
Programming 6759 bytes to Slave MCU at 000000 ... 100%
Closing upgrade session ... OK
```

### Example 2

Schroff Chassis with the Linux upgradefw utility and a network connection to the Shelf Manager:

1. Unzip the upgrade-fedora.zip file.
2. Install the upgradefw utility on a PC running Fedora Linux.
3. Move the 63998-09251.upgrade.img file to the PC running Fedora Linux.
4. Insure that you have a network connection to the Shelf Manager in the Schroff chassis using the "ping" command.
5. Execute the upgradefw utility once for each PEM.

This text below shows the procedure for programming the right PEM. You need to substitute "192.168.0.2" in the command for the IP address of the Shelf Manager in your chassis. The "66" parameter in the command is the IPMB address for the PEM, either 66 or 68.



```
./util/upgradefw -n 192.168.0.2,66 63998-09251.upgrade.img
BMR firmware upgrade utility. Pigeon Point Systems (c) 2004-2007.
Upgrade method: LAN/IPMB
Host      : 192.168.0.2
Port      : 623
Target IPMC: 0x66

Opened an RMCP session (ID = 00000001).
Firmware upgrade image: 63998-09251.upgrade.img
Preparing Master MCU for programming .....OK
Programming 49084 bytes to Master MCU at 000000 ... 100%
Preparing Slave MCU for programming ...OK
Programming 6759 bytes to Slave MCU at 000000 ... 100%
```

### Example 3:

Schroff Chassis with the upgradefw utility running on the Shelf Manager:

1. Unzip the upgrade-shmm500.zip file.
2. Install the upgradefw utility on one of the shelf managers in the Schroff Chassis.
3. Move the 63998-09251.upgrade.img file to the same shelf manager in the Schroff Chassis.
4. Execute the upgradefw utility once for each PEM.

This text below shows the procedure for programming the right PEM. The “66” parameter in the command is the IPMB address for the PEM, either 66 or 68.

```
# ./upgradefw -I /dev/i2c1,66 upgrade.img
BMR firmware upgrade utility. Pigeon Point Systems (c) 2004-2007.
Upgrade method: IPMB
Device      : /dev/i2c1
Target IPMC: 0x66

Firmware upgrade image: upgrade.img
Preparing Master MCU for programming .....OK
Programming 44755 bytes to Master MCU at 000000 ... 100%
Preparing Slave MCU for programming ...OK
Programming 7969 bytes to Slave MCU at 000000 ... 100%
```

### 3 Updating the PEM FRU data with a PPS based Shelf Manager



It is not necessary to update the FRU data stored in the PEM.  
Only use this procedure if instructed to do so by Schroff technical support.

Do not use the CLI command “clia frudataw” to update the PEM FRU data. Instead, use the script `/var/nvdata/fru_update` to update PEM FRU data. This script saves PEM specific data like serial numbers and manufacturing date. A complete list of saved data is listed in table below

Table 1: Saved variables

Variable Name	Area	Info
Serial number	[Board Info Area]	
Serial number	[Product Info Area]	
Mfg. Date/Time	[Board Info Area]	
Product Version	[Product Info Area]	
Asset Tag	[Product Info Area]	
	Internal Use Area	Complete

#### FRU ID file update:

##### Syntax:

`sh /<path>/fru_update <IPMC> <FRU ID> <path>/<filename>`

**<path>:** Location of the FRU files, default: `/var/nvdata`

**<filename>:** 63998-09251.fru-info.bin

The table below lists valid values for **<IPMC>** and **<FRU ID>** for a 6-slot Schroff chassis.

<IPMC>	<FRU ID>	Description
66	0	PEM

##### Examples:

`sh /var/nvdata/fru_update 66 0 /var/nvdata/63998-09251.fru-info.bin`  
Updates the PEM FRU data preserves the variables listed in Table 1.

```
sh /var/nvdata/fru_update 68 0 63998-09251.fru-info.bin
```

```
Starting update....
Info: Using /var/nvdata/patch_fru_info and /var/nvdata/fru_internal
Saving FRU Data ...
Installing new FRU file '63998-09251.fru-info.bin' ...
Writing Inventory Data Fields:
Product Serial Number: 98765
Board Serial Number: 12345
Manufacturing Date: 2008-10-2:17:10
Product Version: C
Asset Tag: 009200920
Writing Internal FRU Area ...
FRU Data Update completed
```

## 4 Revision History

### 4.1 Schroff Specific Firmware Changes

#### Revision 51 Build 5122

Schroff has made a number of changes to the PPS V1.5.1 base IPMC firmware to make the IPMCs compliant with Polaris Networks ATCA conformance tests and have improved functionality with HPI and Shelf Managers that are not based on PPS firmware.

- Increased chassis FRU data storage space on FRU #1 to 3072 bytes.
- Changed IANA code from Pentair to Schroff.
- All IPMI commands will accept the PPS IANA code in correct and reversed order.
- The Entity Instance and Entity ID for all SDRs (Sensor Data Records) have been changed to associate all sensors with the Management Controller on the FRU. Entity association records have been removed from the SDRs.
- The values for the Assertion Event Mask, Deassertion Event Mask, and Discrete Reading Mask for the hot swap sensor have been changed from 0xFFFF to 0x00FF.
- A one second delay was added to the IPMC boot to improve Hardware Address detection during Hot-Swap.
- The Get Address Info IPMI command response contains the correct Site Type.
- The Get FRU Led State IPMI command response contains the correct Override State.
- The Get Power Level IPMI command response contains the correct Early Power Level and Steady Power Level.
- The Shelf Manager can now read all non-sensor SDR records.

### 4.2 PPS V1.5.1 New Features

Revision 1.5.1 of the BMR-AVR-ATCA Firmware provides the following new features, as compared to revision 1.4.5 of the BMR-AVR-ATCA Firmware:

- Compliance with the ECN 3.0-2.0-002 amendment to PICMG 3.0 R2.0 (referenced as ECN-002 below, refer to 1.3.2 for details)
- In-field firmware upgrade doesn't affect the current state of the payload and LEDs, as long the types and names of the reset-preserved variables are identical in previous and freshly installed versions of the firmware (new file `util/genccks.c`, changes in `boot/boot.c`, `hal/Makefile`, `hal/init.S`, `util/Makefile`, `util/Makefile.vc`). The Boot Loader must be upgraded in order to enable this feature.

NOTE: The Boot Loader can only be updated via JTAG interface using the JTAG ICE tool. The Boot Loader cannot be upgraded by `upgradefw`.

- Support for the IPMI v2.0-defined Version Change sensor (new files `app/master/sensor_version.*`, changes in `app/master/Makefile`, `app/master/app.c`, `app/master/sensor.h`, `include/cfgint.h`, `configs/*/config.h`, `configs/*/sdr-data.inf`)

NOTE: Due to the requirements of the ECN-002 and specifics of the BMRAVR-ATCA firmware, the Version Change sensor is mandatory and therefore must be present in each configuration of the firmware.

### 4.3 Firmware Improvements

Revision 1.5 of the BMR-AVR-ATCA Firmware contains the following improvements as compared to revision 1.4.5 of the BMR-AVR-ATCA Firmware:

- Support for a dummy SIGNAL\_NONE signal has been added, to be used in the CFG\_P2P\_EKEY\_LINKS and CFG\_AMC\_P2P\_EKEY\_LINKS configuration parameters (changes in `app/signal.c`, `app/master/ekeying.c`, `hal/pin.h`, `include/cfgdefs.h`).
- The WDT\_INIT precompiler definition has been added (changes in `hal/wdt.h`, `boot/boot.c`). It replaces the calls to the wdt\_init function to get rid of excessive precompiler conditional directives.
- The Watchdog Timer has been enabled in the Boot Loader (changes in `boot/boot.c`).
- The Set Sensor Event Enable handler has been made more tolerant to malformed requests (changes in `app/master/sensor.c`).
- A queue for unsent hot swap events has been added to avoid event loss (changes in `app/master/sensor_hotswap.c`).
- Support for the auxiliary firmware version has been added (changes in `app/master/ipmi.c`)
- The PICMG extensions version has been changed to 2.2 (changes in `app/master/atca.c`)
- The Get Address Info command handler has been changed to conform to ECN-002 (changes in `app/master/atca.c`)
- The FRU Control Capabilities command handler has been added (changes in `app/master/atca.c`, `include/atcadsfs.h`)
- The Get Power Level command handler has been changed to conform to the PICMG 3.0 ECN 002 specification (changes in `app/master/atca.c`)
- In accordance with ECN-002, handlers for the following ATCA commands have been modified to ignore the following commands if they come from the payload interface:
  - Set IPMB State (PICMG 3.0, section 3.8.4.3)
  - Set Port State (PICMG 3.0, section 3.7.2.4)
  - Compute Power Properties (PICMG 3.0, section 3.9.1.4)
  - Set Power Level (PICMG 3.0, section 3.9.1.4)
  - Set Fan Level (PICMG 3.0, section 3.9.1.5)
  - Bused Resource (PICMG 3.0, section 3.7.3.4)
  - Changes in `app/master/atca.c`.
- The Get Fan Level command handler has been changed to conform to the ECN-002 specification (changes in `app/master/fan.c`).

- Completion codes for those commands that require a FRU ID field have been changed in accordance with ECN-002 (changes in `app/master/atca.c`, `app/master/fan.c`, `app/master/led.c`). If the FRU ID field is greater than the maximum FRU ID, the C9h (Parameter out of range) completion code is returned.
- The Get IPMB Link Info command handler has been added (changes in `app/master/atca.c`, `app/master/ipmb.*`).
- The completion code for the FRU Control command has been changed in accordance with ECN-002 (changes in `app/master/payload.c`). If a response from the payload is not received within the configured timeout, the D5h (Command not supported in present state) completion code is returned.
- The Graceful Reboot command handler has been extended to recognize one additional optional parameter – a completion code (changes in `app/master/payload.c`). The IPMC passes this code on to the Shelf Manager as a result of execution of the FRU Control command.
- The Platform Event Message (PEM) command handler has been added (changes in `app/master/ipmi.c`, `app/master/sensor.*`). The IPMC passes events that are embedded in the PEM commands on to the Shelf Manager. PEM commands are only accepted from the payload interface.
- The `led_set_local` function has been modified to allow using it from timer callbacks. (changes in `app/master/eestore.*`, `app/master/led.c`).
- Notification messages about changes in the E-keying link states have been added (changes in `app/master/ekeying.c`). The firmware outputs these messages to the Serial Debug interface.

## 4.4 Bug Fixes

Revision 1.5 of the BMR-AVR-ATCA Firmware fixes the following bugs of earlier revisions of the BMR-AVR-ATCA Firmware:

- The IPMC might be restarted by the watchdog timer if the IPMB-0 link is not functioning for some reason and there are pending sensor events (changes in `app/master/sensor.c`).
- The firmware failed to compile when the `CFG_SDI_ENABLE` Configuration Parameter is not defined and the Boot Loader is enabled (changes in `boot/boot.c`).
- Incorrect upper limit for the IPMB message size (changes in `app/master/app.c`, `app/slave/app.c`, `boot/boot.c`).
- Incorrect lower limit for the IPMB message header size (changes in `util/upgrade_ipmb.c`).
- The firmware might fail to handle some SIPL commands if they are sent quickly one after another (changes in `app/master/app.c`).
- IPMI messages were put into the main message ring with incorrect sequence numbers (change in `app/master/app.c`).
- Incomplete check for duplicated command requests in the Boot Loader that might cause misbehavior during an upgrade procedure (changes in `boot/boot.c`).
- The Network Function codes for the packets obtained from the promiscuous queue were not checked for correctness (changes in `util/upgrade_ipmb.c`).
- Incorrect signal configuration that might cause signal spikes (changes in `app/signal.c`).
- The Boot Loader might not store the first page of the backup image to Flash (changes in `boot/boot.c`).



- A reset of the payload was not followed by a configured delay in the payload power on/off sequences (`app/master/payload.c`).
- The hot swap sensor's `mask_to_state` function incorrectly handled the M7 state (changes in `app/master/sensor_hotswap.c`).
- The hot swap state change cause was incorrectly reported if the state change was due to triggering the Handle Switch (changes in `app/master/fru_hotswap.c`).
- The responder slave address field in the Get Message message response had an incorrect value (changes in `app/master/ipmi.c`).
- IPMC threshold assert/deassert events were sent in the reversed order (changes in `app/master/sensor_discrete.c`).
- The pull-ups for HIGHZ signals were not disabled (changes in `app/signal.c`).
- The PPS IANA identifier's most and least significant bytes were swapped in requests and responses (changes in `app/master/app.c`, `app/master/app.h`, `app/master/ekeying.c`, `app/master/ipmi.c`, `app/master/sipl.c`, `app/master/sipl.h`, `boot/boot.c`, `configs/*/config.h`, `util/mkupgimg.c`, `util/upgradefw.c`). Commands with the incorrect PPS IANA byte order are still supported for backward compatibility with existing payload software. However, a warning message is printed to the Serial Debug console on receipt of such commands. This support will be removed in a future version of the firmware.