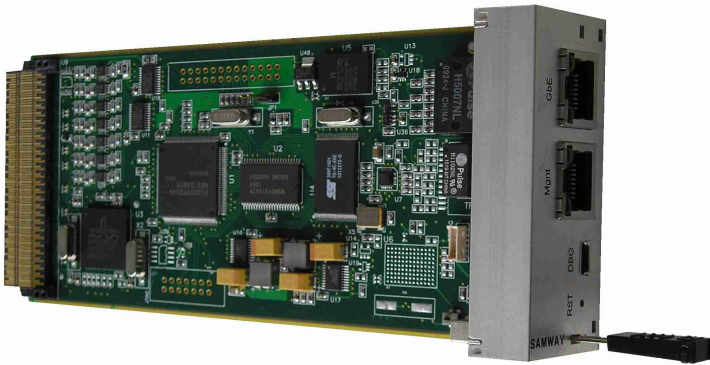


MicroTCA Carrier HUB MCH



Key Features:

- Support for up to 12 AMCs, two CUs and four PMs
- On-board 12 ports unmanaged GbE Switch
- Front panel GbE port connected to the switch
- Front panel 10/100Mbps port for management
- Front panel USB connector for debug
- On-board shelf manager
- Own IPMI software
- Firmware upgrade via IPMI commands (HPM.1) or debug interface

Description:

The [Samway](#)-MCH is a MicroTCA (uTCA/MTCA)Carrier Hub that can provide the advanced management and data switching required in any MicroTCA system.

It's main functions include IPMI controlled power management, Electronic keying, Hot-swap of AMCs and switching functionality for system Fabric A(1GbE) as defined in the AMC.x standard series.

The Samway-MCH uses Samway developed IPMI software. Samway has a vast experience in MicroTCA(uTCA/MTCA) and our portfolio includes all the devices defined in the standard : MCH(MicroTCA Hub Controller), PDM (Power Distribution Module), CU (Cooling Unit), AMC (Advanced Mezzanine Card).

Samway-MCH has been thoroughly tested at the Interoperability Workshops organized by PICMG.

Table of Contents

1 Front Panel and Connectivity.....	3
2 Serial interface	3
2.1 Driver.....	3
2.2 Terminal Program	3
3 Ethernet Management interface.....	4
4 Command Line Interface (CLI)	4
4.1 Log-in	5
4.2 Syntax directory.....	5
4.2.1 List of Commands.....	5
4.2.2 help command	6
4.2.3 fru command	6
4.2.4 channels command	6
4.2.5 sdr command	7
4.2.6 sensor command	8
4.2.7 threshold command	10
4.2.8 frubuffer command	11
4.2.9 parse command	11
4.2.10 bpppc command	12
4.2.11 links command	12
4.2.12 ekey command	13
4.2.13 cu command	13
4.2.14 fanlevel command	13
4.2.15 sendipmb command.....	14
4.2.16 reset command.....	14
4.2.17 carrierno command.....	14
4.2.18 xmodem command	14
4.2.19 scispeed command	14
4.2.20 lanconfig - command	15
4.2.21 saveenv command	15
4.2.22 reboot command	15
4.2.23 logout command	16
4.2.24 passw command	16
4.2.25 voltage command	16
4.2.26 temp command	16
4.2.27 uptime command	16
4.2.28 restore command	16
5 Ordering info.....	17
5.1 Ordering Formula.....	17
5.2 Order Codes.....	17

1 Front Panel and Connectivity



Fig 1: Front Panel

The Front Panel of the Samway-MCH is presented in Fig.1.

The MCH has 3 Front panel connectors:

- GbE – RJ45 connector directly linked with the 16 ports GigaBit Ethernet Switch. The switch provides switching functionality for system Fabric A(1GbE).
- Mgnt – RJ45 connector for 10/100 [Ethernet management interface](#).
- DBG – USB mini B connector for the [Serial interface](#) . It provides access to the on board [CLI \(command line interface\)](#)

The MCH has also a Reset push-Button (RST) which asserts a soft reset for the IPMI management controller without resetting the GbE Switch.

The Samway-MCH is equipped with the MicroTCA Hot-swap Handle and the 3 Leds defined in MicroTCA .

2 Serial interface

The MCH provides a serial interface over which the commands of the Command Line Interface (CLI) can be sent.

The CLI is available via the front panel USB mini B connector.

2.1 Driver

The Samway-MCH uses the [FT232R](#) chip for converting the serial Comand Line Interface to USB signals.

! For successfully connecting a PC to the MCH, a Virtual COM port (VCP) driver is required. Usually the appropriate driver is automatically detected and installed by the operating system.

In case the operating system fails to install the correct driver, you can download the latest driver for FT232R from [here](#), and follow the instructions of the [installation guides](#).

2.2 Terminal Program

Once the driver has been successfully installed you can connect to the MCH using any terminal program.

On Windows systems, we recommend the use of “Tera Term” or “Hyperterminal” as the terminal programs.

Terminal settings:

- 19200 bits per second (default baud rate);this baud rate can be changed using “**scispeed**” CLI command (syntax directory [4.2.17 scispeed](#) - command)
- data bits: 8
- parity: none
- stop bit: 1

In addition, the “**xmodem**” CLI command can be used for file transfer.(syntax directory [4.2.16 xmodem](#) - command)

! When using “**xmodem**” in “Hyperterminal” the transfer of the desired file can take up to 10 seconds to start.

3 Ethernet Management interface

The integrated 10/100Mbps Ethernet interface allows the MCH to be linked to any existing network. The interface supports HTTP and TELNET protocols via TCP/IP.

The user has full access to the commands of the Command Line Interface (CLI) via TELNET.

The use of standard protocols avoids the need for special software or drivers and so achieves platform-independence. The TCP/IP protocol supports 10 simultaneous connections and the maximum packet size is limited to 1k.

On Windows systems, we recommend the use of “Hyperterminal” or “PuTTY” as TELNET client.

! The factory default setting for the MCH IP address is 192.168.16.17 - the IP address must be adjusted via the **lanconfig** command (syntax directory [4.2.18 lanconfig](#) - command (lan configuration)) for use in your network.

Terminal settings:

- Local echo: *off*
- Local line editing: *off*
- Backspace key: *Control-H*

4 Command Line Interface (CLI)

The Command Line Interface (short-form: CLI) is available via the USB interface. The user can read or newly configure and save system parameters via the CLI. Access is divided into 2 profiles and is password-protected.

“user” profile:

System parameters can only be read in this profile – the exception to this write-protect is the **lanconfig** command (syntax directory [4.2.18 lanconfig](#) - command (lan configuration)) for setting the IP, subnet and gateway addresses.

“admin” profile:

Full access to all system parameters. All available CLI commands can be executed. To avoid possible damage or malfunctions, the access data for this profile must only be known to trained personnel with appropriate knowledge and competence relating to the system in which the MCH is used!

The profiles can be changed using **logout** (syntax directory [4.2.21 logout](#) – command)

4.1 Log-in

As soon as you have established a connection, you will be prompted to login.

Default access settings:

login: user
password: USER

login: admin
password: ADMIN



The passwords can be changed using "**passw**" (syntax directory [4.2.22. passw](#) – command)

4.2 Syntax directory

4.2.1 List of Commands

- [help](#): displays a list of all available commands. For a detailed description you should use this user manual.
- [fru](#): displays the active devices in the system (PMs,AMCs and Cus)
- [channels](#): displays information regarding the power status of each Fru in the system: PS1, MP, EN, PP. The command also displays info regarding the PMs in the system: PM's status(redundant,primary,not used), amount of used current, amount of available current.
- [sdr](#): displays the raw sdrs for a requested FRU
- [sensor](#): parses sdr info for all the sensors in the system, or all the sensors of a requested FRU, or a particular sensor. The command displays sensor info in a user friendly manner.
- [threshold](#): changes the desired threshold for the desired sensor to a inputed value.
- [frubuffer](#): displays the raw fru info for all the FRUs in the system
- [parse](#): displays parsed fru info for the desired FRU
- [bpppc](#): **back plane point to point** connectivity. Displays a summary of the backplane AMCs connectivity.
- [links](#): displays a list of all the links defined in the fru info of the desired AMC.
- [ekey](#): displays a summary of the E-Keying process.
- [cu](#): displays information regarding the Cooling units
- [fanlevel](#): the command can either displays the override fan level and local control fan level or set a fanlevel
- [sendipmb](#): sends a raw IPMI command via the IPMB. The command has to be entered in hex without the prefix 0x. The checksum fields can be written as 00 as they are computed by the command.
- [reset](#) : performs a warm or cold reset of the desired FRU.
- [carrierno](#): displays the carrier number.
- [xmodem](#): receives, using the xmodem protocol, the user settings, the FRU file for the carrier, the sdr

or the fru file for the mch

- [scispeed](#): changes the baud rate at which the CLI for the MCH and the bootloader framework operate.
- [lanconfig](#): readout or setting of network parameters.
- [saveenv](#): saves the changes that have been made to the parameters, in the flash memory.
- [reboot](#): restarts the MCH.
- [logout](#): logs out the current user
- [passw](#): changes the password for the current user
- [voltage](#): displays the voltage values for the Management Power and Payload Power rails
- [temp](#): displays the MCH's temperature sensors values
- [uptime](#): displays the amount of time that has passed since the last reboot.
- [restore](#): restores all parameters to the default values

4.2.2 help command

Syntax: help

Functions:

Displays a list of all available commands. For all commands only the syntax and an example are displayed. For a detailed description you should use this user manual.

4.2.3 fru command

Syntax: fru

Functions:

Displays the active devices (PMs,AMCs and CUs) and their operational states. The command also displays the FRU's name(if available).

Example:

```
%>fru
MCMC1:  M4 - 0  MCMC
PM1   :  M4 - 0  PDM
CU1   :  M4 - 0  Cooling  Unit
AMC2  :  M4 - 0  A2:AM4011
AMC3  :  M1 - 9  A3:AM4301
AMC4  :  M4 - 0  AMC-S302-80G
```

4.2.4 channels command

Syntax: channels

Functions:

Displays information regarding the power status of each FRU in the system: PS1, MP, EN, PP. The command also displays information regarding the PMs in the system: PM's status(redundant,primary,not used), amount of used current, amount of available current.

The info is structured using a table. The lines represent the channels, and each of the columns has a different meaning:

```
PS1 : present
MP  : management power
EN  : enable
```

PP : payload power

Legend: "y" - asserted , "-" -not asserted.

Example:

```
%>channels
```

Channel	Primary PM					Redundant PM					Current AMPs
	No.	PS1	MP	EN	PP	No.	PS1	MP	EN	PP	
0	1	Y	Y	Y	Y	2	Y	Y	Y	Y	3.0
1	-	-	-	-	-	-	-	-	-	-	-
2	1	Y	Y	Y	Y	2	Y	Y	Y	Y	2.0
3	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-
5	1	Y	Y	Y	Y	2	Y	Y	Y	Y	3.0
6	1	Y	Y	Y	-	2	Y	Y	Y	-	-
7	1	Y	Y	Y	Y	2	Y	Y	Y	Y	0.4
8	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-

PM no.	Status	Current	Capability	Used	Current
1	Primary	25.0	AMP	8.4	AMP
2	Redundant	25.0	AMP	8.4	AMP
3	Not Present				
4	Not Present				

In this case channel 2 (CU1), is present (PS1 is asserted), receives management power, is enabled and receives payload power(MP1,EN and PP asserted); has a primary PM and a redundant PM assigned(PM1 -primary, PM2 -redundant), and has a current requirement of 2 Amps.

4.2.5 sdr command

Syntax: `sdr amc|pm|cu|mcmc site`

Functions:

Displays the raw sdrs for the requested device. The start of a new sdr if marked with * so it is easier to find a particular record.

If you want a parsing of the sdr data you should use the sensor command.

Example:

```
%>sdr pm 1
PM 1 SDRs Number:6
*00 00 51 12 0E 88 00 00 2A 00 00 00 0A 61 00 C3
 50 44 4D D1 8F 17 CC 3E 00 EF A1 3A 3A 6C 88 F2
*02 00 51 01 33 88 00 00 0A 61 03 42 F2 6F 07 00
 07 00 07 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C8
 48 6F 74 20 53 77 61 70 06 62 25 7A EE BC 50 D1
*06 00 51 01 30 88 00 01 0A 61 03 C5 02 01 04 22
 04 22 12 00 00 04 00 00 C4 00 00 00 00 C0 00 00
 00 00 FF 00 FF BA BA 00 97 97 00 00 00 00 00 C5
 2B 33 2E 33 56 89 8F 74 EA 73 E1 FF 34 9E 83 DD
```

```
*0A 00 51 01 2F 88 00 02 0A 61 03 C5 02 01 04 22
 04 22 12 00 00 04 00 00 41 00 00 00 00 D0 00 00
 00 00 FF 00 FF D8 D8 00 9A 9A 00 00 00 00 00 C4
 2B 31 32 56 72 70 BB 6D F2 E2 4A 3F 18 B7 B9 F4
*0E 00 51 03 16 88 00 03 0A 61 F3 6F 00 00 00 00
 CA 50 44 4D 20 45 76 65 6E 74 73 B2 39 24 4A 7B
*10 00 51 01 34 88 00 04 0A 61 03 C1 F1 6F 0F 00
 0F 00 0F 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C9
 49 50 4D 42 20 4C 69 6E 6B BE F7 3E B3 C2 6D F7
```

4.2.6 sensor command

Syntax: sensor [all] | [amc|pm|cu|mcmc site] | [sensor_no]

Functions:

Parses sdr info for all the sensors in the system, or all the sensors of a requested FRU, or a particular sensor, and displays sensor info in a user friendly manner.

Example1:

```
%>sensor pm 1
-----Sensor List-----

*legend:
Disc -> discrete
Thr  -> threshold
Unsp -> unspecified
l    -> lower
u    -> upper
c    -> critical
nc   -> non-critical
nr   -> non-recoverable

---no--Type--Value--Unit---State-----Name-----
%>
*  0  Disc  Handle Closed          Hot Swap
*  1  Thr   3.29   V           Ok           +3.3V
*  2  Thr  12.22   V           Ok           +12V
*  3  Unsp
*  4  Disc  Raw: 88 C0 08 80        IPMB Link
```

For every sensor certain info is displayed:

- sensor number
- sensor type: discrete, threshold, unspecified
- value if it can be parsed (otherwise a raw value will be displayed)
- measuring unit if available
- current status : **Ok**; limit infringement : lower non-critical (**lnc**), lower critical(**lc**), lower non-recoverable(**lnr**), upper non-critical(**unc**), upper critical(**uc**), upper non-recoverable (**unr**)
- name (if available)

If a more detailed description is desired, the command can be used to display a more in depth parsing of the desired sensor.

Example2:

```
%>sensor 23
-----Sensor Details-----
```



```
* Device: AMC 2
* Sensor no: 23
* Local sensor no: 11
* Name: A2:Temp CPU
* Type: Threshold
* Value: 52.00 deg C
* Status: Ok
* Sensor Capabilities
  Sensor Auto Re-arm Support: Auto
  Sensor Hysteresis Support: hysteresis is readable and settable
  Sensor Threshold Access Support: readable and settable per Reading Mask and
Threshold Mask
  Sensor Event Message Control Support: per thresholds/discrete-state event
* Lower Threshold Reading Mask:
  No Lower threshold comparison is returned
* Upper Threshold Reading Mask:
  upper non-recoverable threshold
  upper critical threshold
  upper non-critical threshold
* Threshold Assertion Event Mask:
  upper non-recoverable going high
  upper critical going high
  upper non-critical going high
* Threshold Deassertion Event Mask:
  upper non-recoverable going high
  upper critical going high
  upper non-critical going high
* Settable Threshold Mask:
  upper non-recoverable threshold
  upper critical threshold
  upper non_critical threshold
* Readable Threshold Mask:
  upper non-recoverable threshold
  upper critical threshold
  upper non_critical threshold
* Current Value: 52.00
* Sensor Units: deg C
* Nominal Reading: 75.00
* Normal Maximum: 85.00
* Normal Minimum: 0.00
* Sensor Maximum Reading: 127.00
* Sensor Minimum Reading: -128.00
```

- * Upper non-recoverable threshold:110.00
- * Upper critical threshold: 100.00
- * Upper non-critical threshold: 90.00
- * Positive-going threshold hysteresis value: 2.00
- * Negative-going threshold hysteresis value: 2.00

In this mode the command displays various info for the desired sensor depending on it's type:

- device: the FRU to which the sensor belongs
- sensor no: the sensor number assigned to it by the mcmc
- local sensor no: the sensor number assign to it by its carrier FRU
- name
- type
- value

For *threshold* sensors:

- status
- sensor capabilities
- threshold masks: Lower Threshold Reading Mask, Upper Threshold Reading Mask Threshold Assertion Event Mask, Threshold Deassertion Event Mask, Settable Threshold Mask, Readable Threshold Mask
- reading info: current value, measurement unit, nominal reading, normal maximum, normal minimum, sensor maximum reading, sensor minimum reading
- threshold values: all available thresholds for the desired sensor
- hysteresis values: positive going and negative going

For *discrete* sensors:

- sensor capabilities
- discrete masks: assertion event mask, deassertion event mask, reading mask

4.2.7 threshold command

Syntax: `threshold sensor_no Inc|lc|Inr|unc|uc|unr threshold_value`

Functions:

Changes the desired threshold for the desired sensor to a inputed value. The sensor number is the one assigned to the sensor by the mcmc and not the local sensor number assigned by the FRU to which the sensor belongs.

- ! Thresholds can be changed only for sensors that advertise that the desired threshold can be set.
- ! Some FRUs don't maintain the changed threshold's value after a reset.

The thresholds abbreviations are:

Inc → lower non-critical

lc → lower critical

lnr → lower non-recoverable
unc → upper non-critical
uc → upper critical
unr → upper non-recoverable

Example1:

```
%>threshold 64 unr 3.2
%>
  Successful!
  New Threshold:3.19
```

In this case the upper non-recoverable threshold of sensor 64 was set to 3.2.

Example2:

```
%>threshold 87 unr 3.1
  Upper non_recoverable threshold can not be set!
%>
  Failed!
```

In this case the user tried to set the upper non-recoverable threshold of sensor 87 to 3.1. Because the sensor didn't allow this type of threshold to be modified the operation failed.

4.2.8 frubuffer command

Syntax: frubuffer

Functions:

Displays the raw fru info for all the FRUs in the system. The FRUs are identified by their IPMB address.

4.2.9 parse command

Syntax: parse [address]

Functions:

Displays parsed FRU information for the device whose no prefix hex ipmb address is entered. If no address is given or the address is FD the command will parse the carrier fru information.

Example: (72 -> amc 1)

```
%>parse 72
  Module Current Requirement
    Amp=3
  AMC P-P Connectivity
ChID:0- Lane0:P0 Lane1:P31 Lane2:P31 Lane3:P31
ChID:1- Lane0:P1 Lane1:P31 Lane2:P31 Lane3:P31
ChID:2- Lane0:P2 Lane1:P31 Lane2:P31 Lane3:P31
ChID:3- Lane0:P3 Lane1:P31 Lane2:P31 Lane3:P31
ChID:4- Lane0:P4 Lane1:P5 Lane2:P6 Lane3:P7
Link:0-ChID:0-Lane0-Eth
Link:1-ChID:1-Lane0-Eth
Link:2-ChID:2-Lane0-Storage
Link:3-ChID:3-Lane0-Storage
Link:4-ChID:4-Lane0 Lane1 Lane2 Lane3-PCIE
Link:5-ChID:4-Lane0 Lane1 Lane2 Lane3-PCIE
Link:6-ChID:4-Lane0-PCIE
Link:7-ChID:4-Lane0-PCIE
```

4.2.10 bpppc command

Syntax: bpppc

Functions:

back plane point to point connectivity: Displays a summary of the backplane AMC's connectivity. The Columns represent all the AMC's in the system and the rows represent all the ports of those FRUs.

Example:

```
%>bpppc
-----Backplane point to point connectivity-----

*Legend :
Pn   : PORT n
Mnx  : MCH n Fabric x
An:x : AMC n port x

AMC   1      2      3      4      5      6      7      8      9      10     11     12
P0 M1A:1 M1A:2 M1A:3 M1A:4 M1A:5 M1A:6 - - - - -
P1 M1A:7 M1A:8 M1A:9 M1A:10 M1A:11 M1A:12 - - - - -
P2 A3:2 A4:2 A1:2 A2:2 - - - - -
P3 A2:3 A1:3 A4:3 A3:3 - - - - -
P4 - - - - -
P5 - - - - -
P6 - - - - -
P7 - - - - -
P8 A4:8 A5:8 A6:8 A1:8 A2:8 A3:8 - - - - -
P9 A4:9 A5:9 A6:9 A1:9 A2:9 A3:9 - - - - -
P10 A4:10 A5:10 A6:10 A1:10 A2:10 A3:10 - - - - -
P11 A4:11 A5:11 A6:11 A1:11 A2:11 A3:11 - - - - -
P12 - - - - -
P13 - - - - -
P14 - - - - -
P15 - - - - -
P16 - - - - -
P17 A4:17 A6:17 A5:17 A1:17 A3:17 A2:17 - - - - -
P18 A4:18 A6:18 A5:18 A1:18 A3:18 A2:18 - - - - -
P19 A4:19 A6:19 A5:19 A1:19 A3:19 A2:19 - - - - -
P20 A4:20 A6:20 A5:20 A1:20 A3:20 A2:20 - - - - -
```

The highlighted **info** translates into : AMC 1 port 8 is linked to AMC 4 port 8.

4.2.11 links command

Syntax: links amc amc_no

Functions:

Displays a list of all the links defined in the fru info of the desired FRU. For every link the command displays:

- the link type
- the ports it uses
- the link's grouping id field
- the link's asymmetric match field
- the links status: enabled,disabled,waiting enable,waiting disable
- partner

Example:

```
%>links amc 2
                                     AMC 2
No--Link Type-----Ports-----Gr.Id--As.Match-Status----Partner-----
0 Ethernet 1000Base-BX 0          0x00 0x00 Enabled MCH1 F-A 2
1 Ethernet 1000Base-BX 1          0x00 0x00 Enabled MCH1 F-A 8
2 Storage SATA        2          0x00 0x02 Enabled AMC4 2
3 Storage SATA        3          0x00 0x02 Disabled
4 PCI Express Gen1-SSC 4 5 6 7    0x00 0x02 Disabled
5 PCI Express Gen1    4 5 6 7    0x00 0x02 Disabled
6 PCI Express Gen1-SSC 4          0x00 0x02 Disabled
7 PCI Express Gen1    4          0x00 0x02 Disabled
```

In this case AMC 2 has 3 enabled links: two connected to the MCH fabric A port 2 and 8 and one connected to AMC 4 port 2.

4.2.12 ekey command

Syntax: ekey

Functions:

Displays a summary of the E-Keying process.

Example:

```
%>ekey
                        Ekeying Summary

AMC  Ports      Device  Ports      Type
-----
AMC2  0           MCH1  F-A 2      Ethernet 1000Base-BX
AMC2  1           MCH1  F-A 8      Ethernet 1000Base-BX
AMC2  2           AMC4   2          Storage SATA
AMC4  2           AMC2   2          Storage SATA
```

4.2.13 cu command

Syntax: cu

Functions:

Displays information regarding the Cooling units. The command displays minimum and maximum fan levels, the normal operating fan level and the current fan level.

Example:

```
%>cu
* CU 1
  Minimum fan level =1
  Maximum fan level =15
  Normal operating level =3
  Current fan level =3
* CU 2 - Not Present
```

4.2.14 fanlevel command

Syntax: fanlevel cu cu_no {fan_level}

Functions:

The command can either displays the override fan level and local control fan level or set a fanlevel if a value is entered after the cooling unit's number.

Example1:

```
%>fanlevel cu 1
  CU 1
%>
  Override fan level = 5
  Local control fan level = 15
  Local control enable state = 0
The fanlevels are displayed.
```

Example2:

```
%>fanlevel cu 1 12
  CU 1
%>
  Override fan level = 12
  Local control fan level = 15
  Local control enable state = 0
The override fanlevel is changed.
```

4.2.15 sendipmb command

Syntax: sendipmb raw_command

Functions:

Sends a raw IPMI command via the IPMB. The command has to be entered in hex without the prefix 0x. The checksum fields **can** be written as 00 as they are computed by the command.

Example:

```
%>sendipmb C2 B0 00 20 08 28 00 28 00
%>20 B4 2C C2 48 28 00 00 CE
```

4.2.16 reset command

Syntax: reset warm|cold amc|cu no.

Functions:

Performs a warm or cold reset of the desired FRU.

Example:

```
%>reset warm amc 3
Done!
```

4.2.17 carrierno command

Syntax: carrierno

Functions:

Displays the carrier number.

Example:

```
%>carrierno
Carrier No =0x0088
```

4.2.18 xmodem command

Syntax: xmodem carrierfru | usersettings | mchfru | mchsdr

Functions:

Receives the user settings, the FRU file for the carrier, the sdr or the fru file for the mch. After the command is entered, the MCH goes into data receive mode and waits for the data to be sent in XMODEM protocol. You can then start the file transfer with your terminal program and select XMODEM as the protocol.



When using "xmodem" in "Hyperterminal" the transfer of the desired file can take up to 10 seconds to start.

Examples:

```
%>xmodem usersettings
Please upload the file...
%>...Done!
```

4.2.19 scispeed command

Syntax: scispeed 9600|19200|38400

Functions:

Changes the baud rate at which the CLI for the MCH and the bootloader framework operate. For the change to become valid the environment has to be saved using CLI command `saveenv` (syntax directory [4.2.19 saveenv](#) - command) and the MCH has to be rebooted, either with the **reboot** command (syntax directory [4.2.20 reboot](#) - command) or by using the reset key.

Example:

```
%>scispeed 9600  
Baud rate changed to 9600.Save Environment and reboot.
```

4.2.20 lanconfig - command

Syntax:

lanconfig [ip | mask| gateway [address]]

Functions:

Readout or setting of network parameters.

- **no parameter** – return of IP, mask and gateway addresses of the LAN interface
- **ip** – IP address of the MCH
- **mask** – network mask
- **gateway** – standard gateway
- **address** – if a value is entered it will be assigned to the chosen parameter



After a new address is set, the change must be saved with **saveenv** (syntax directory [4.2.19 saveenv](#) - command) and the MCH restarted ,either with the **reboot** command (syntax directory [4.2.20 reboot](#) - command) or by using the reset key.

Examples:

- Readout of all network parameters

```
%>lanconfig  
IP=192.168.16.17  
Mask=255.255.255.0  
Gateway=192.168.16.255
```

- Readout of the IP address

```
%>lanconfig ip  
IP=192.168.16.17
```

- Changing the IP address

```
%>lanconfig ip 192.168.18.18  
IP=196.168.18.18
```

4.2.21 saveenv command

Syntax: saveenv**Function:**

Saves the changes that have been made to the parameters,in the flash memory . If the modified parameters aren't saved in flash,they will be lost after reboot.

Examples:

```
%>saveenv  
User Settings saved!
```

4.2.22 reboot command

Syntax: reboot**Function:**

Restarts the MCH.

Examples:

```
%>reboot
```

4.2.23 logout command

Syntax: `logout`

Function:

Loggs out the current user

Examples:

```
%>logout  
login:
```

4.2.24 passw command

Syntax: `passw`

Function:

Changes the password for the current user

Examples:

```
%>passw  
Enter Old password:****  
Enter new password:****  
Use saveenv command before reboot for the change to be successful!
```

4.2.25 voltage command

Syntax: `voltage`

Functions:

Displays the voltage values for the Management Power and Payload Power rails.

Example:

```
%>voltage  
MP=3.36V  
PP=12.11V
```

4.2.26 temp command

Syntax: `temp`

Functions:

Displays the MCH's temperature sensors values.

Example:

```
%>temp  
Temp1=29 DegC  
Temp2=39 DegC
```

4.2.27 uptime command

Syntax: `uptime`

Function:

Displays the amount of time that has passed since the last reboot.

4.2.28 restore command

Syntax: `restore`

* only admin can use restore

Function:

Restores all parameters to the default values. For the restore to be complete a reboot is necessary.

Example:

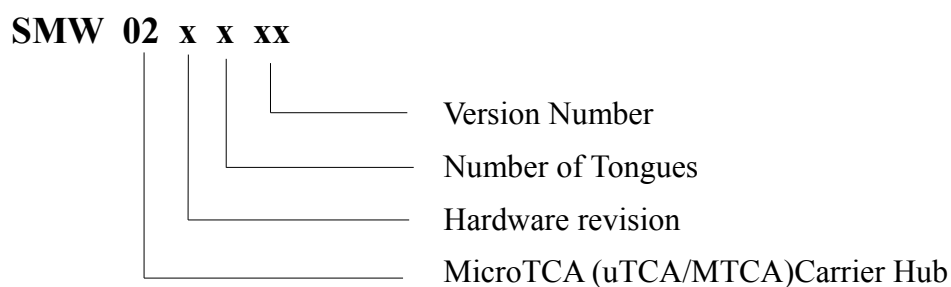
%>restore

Restored to default!

5 Ordering info

5.1 Ordering Formula

For ordering you should use the following ordering formula:



5.2 Order Codes

Part Number	Hardware Rev.	No. of Tongues	Version No.	Description
SMW02B100	B	1	0	MicroTCA (uTCA/MTCA)Carrier Hub System Fabric A switching(1GbE) Support for up to 12 AMCs, 4PMs,2 CUs