

**Irene Hahner**  
Product Management  
12/12/2012

- » Open-standard hardware manageability interface specification
- » Common interface for accessing system components
- » Asynchronous, message based interface
- » Standard IPMI utilities to manage different vendor hardware

## » Device monitoring

- Sensors, events



## » Device management

- Power on, power off, reset

## » Automated system configuration

- Interface description, E-keying
- Power budgeting

## » Inventory Data

- Serial number, index, build date, vendor name...



## » ipmitool

- one executable, open source Linux CLI tool
- the swiss army knife for ipmi related tasks
- Can be used on local or remote system
- To read sensors, SDR, SEL, power states
- To set power states (on, off, cycle, reset)
- To send events to SEL
- To set boot order



## » Openipmi, ipmicmd

- driver,libs and tools, open source linux tool
- driver included since years in all linux distributions

## » Command line interface of MCH (clicm)

- Easy to use, low level commands not always needed

## » **Sensors**

- Report system status
- Fans speed, CPU Temperature, voltage...

## » **SDR - Sensor data repository**

- Collection of sensor data – definition of threshold values

## » **Events**

- temperature sensor threshold exceeds limit,  
voltage too low, hot swap state changed, board reset

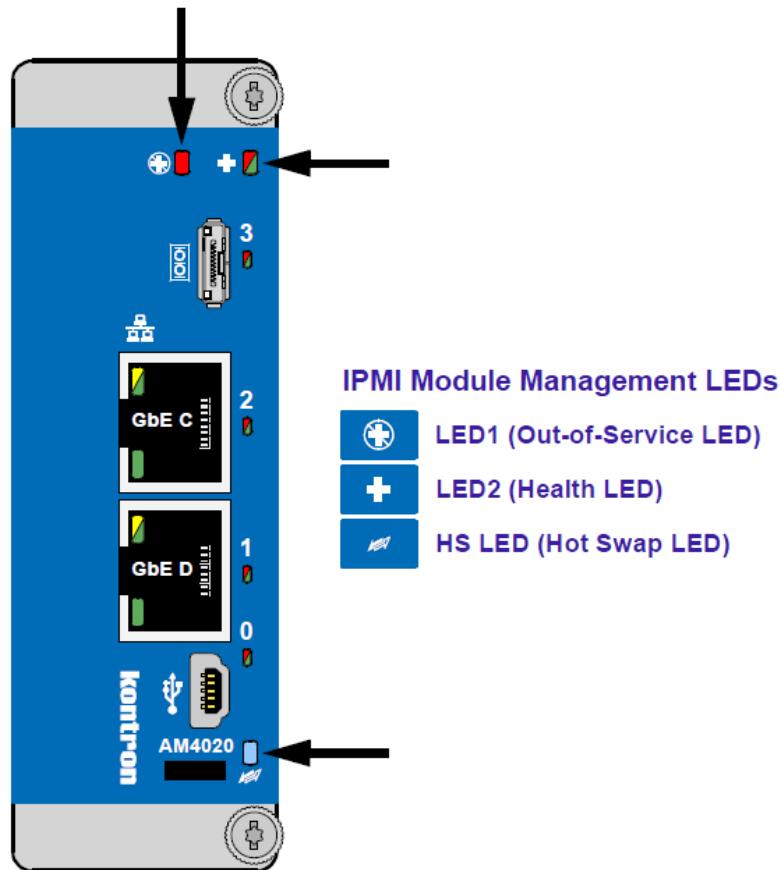
## » **SEL - System Event Log**

- a log of events
- SEL is stored inside MCH, MCH (carrier manager) adds timestamp

## » **FRU data - Field Replaceable Unit data**

- vendor ID, product ID, firmware revision, power requirements....

## » Controlled by MMC



Sensor Number / Name	Sensor Type (Code) / Event/Reading Type (Code)	Ass. Mask / Deass. Mask / Reading Mask	Description	Health LED Red on Error
0Eh / A1:Temp CPU	Temperature (01h) / Threshold (01h)	1A81h / 7A81h / 3939h	CPU die temperature	Y
0Fh / A1:Temp PCH	Temperature (01h) / Threshold (01h)	0A80h / 7A80h / 3838h	PCH temperature	Y
10h / A1:Temp MCH	Temperature (01h) / Threshold (01h)	1A81h / 7A81h / 3939h	MCH temperature	Y
16h / A1:Temp Air	Temperature (01h) / Threshold (01h)	7A95h / 7A95h / 3F3Fh	Air temperature near AMC edge-connector	Y
17h / A1:Board 3.3vIPM	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	AMC Management Power (MP) 3.3V	Y
18h / A1:Board 12.0v	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	AMC Payload Power (PWR) 12V	Y
19h / A1:Board 5.0V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 5V supply	Y
1Ah / A1:Board 3.3V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 3.3V supply	Y
1Bh / A1:Pwr Good	Power supply (08h) / OEM (77h)	0000h / 0000h / 0887h	States of all power lines	N

Sensor Number / ID String	0Eh / A1:Temp CPU	0Fh / A1:Temp PCH	10h / A1:Temp MCH	16h / A1:Temp Air
<b>Upper non-recoverable</b>	115 °C	116 °C	110 °C	95 °C
<b>Upper critical</b>	105 °C	111 °C	100 °C	90 °C
<b>Upper non-critical</b>	95 °C	101 °C	90 °C	80 °C
<b>Normal max.</b>	90 °C	96 °C	85 °C	75 °C

- » Read out one sensor with “Get Sensor Reading cmd”
  - Build command according to spec
  - Need Network/Function code, Cmd and parameters
- » From IPMI Spec , Table 44-11 Command Number Assignments
- » Get Sensor Reading: Net/Fun = S/E; Cmd = 0x2D

Sensor Device Commands							
Get Device SDR Info	35.2	S/E	20h	I	I	I	I
Get Device SDR	35.3	S/E	21h	I	I	I	I
Reserve Device SDR Repository	35.4	S/E	22h	I	I	I	I
Get Sensor Reading Factors	35.5	S/E	23h		X		
Set Sensor Hysteresis	35.6	S/E	24h			X	
Get Sensor Hysteresis	35.7	S/E	25h		X		
Set Sensor Threshold	35.8	S/E	26h			X	
Get Sensor Threshold	35.9	S/E	27h		X		
Set Sensor Event Enable	35.10	S/E	28h			X	
Get Sensor Event Enable	35.11	S/E	29h		X		
Re-arm Sensor Events	35.12	S/E	2Ah			X	
Get Sensor Event Status	35.13	S/E	2Bh		X		
Get Sensor Reading	35.14	S/E	2Dh		X		
Set Sensor Type	35.15	S/E	2Eh			X	

- » From IPMI Spec , Table 5-1 Network Function Codes
- » Net/Fun: S/E = 0x04

*Table 5-1, Network Function Codes*

Value(s)	Name	Meaning	Description
00, 01	Chassis	Chassis Device Requests and Responses	00h identifies the message as a command/request and 01h as a response, relating to the common chassis control and status functions.
02*, 03*	Bridge	Bridge Requests and Responses	02h (request) or 03h (response) identifies the message as containing data for bridging to the next bus. This data is typically another message, which may also be a bridging message. This function is present only on bridge nodes.
04, 05	Sensor /Event	Sensor and Event Requests and Responses	This functionality can be present on any node. 04h identifies the message as a command/request and 05h as a response, relating to the configuration and transmission of Event Messages and system Sensors.
06, 07	App	Application Requests and Responses	06h identifies the message as an application command/request and 07h a response. The exact format of application messages is implementation-specific for a particular device, with the exception of App messages that are defined by the IPMI specifications.  Note that it is possible that new versions of this specification will identify new App commands. To avoid possible conflicts with future versions of this specification, it is highly recommended that the OEM/Group network functions be used for providing 'value added' functions rather than the App network function code.
08, 09	Firmware	Firmware Transfer Requests and Responses	The format of firmware transfer requests and responses matches the format of Application messages. The type and content of firmware transfer messages is defined by the particular device.
0A, 0B	Storage	Non-volatile storage Requests and Responses	This functionality can be present on any node that provides non-volatile storage and retrieval services.

- » From IPMI Spec Command Description
- » just one parameter: sensor number

*Table 35-15, Get Sensor Reading Command*

Request Data	1	sensor number (FFh = reserved)
Response Data	1	Completion Code.
	2	Sensor reading <u>Byte 1:</u> byte of reading. Ignore on read if sensor does not return an numeric (analog) reading.
	3	[7] - 0b = All Event Messages disabled from this sensor [6] - 0b = sensor scanning disabled [5] - 1b = reading/state unavailable (formerly "initial update in progress"). This bit is set to indicate that a 're-arm' or 'Set Event Receiver' command has been used to request an update of the sensor status, and that update has not occurred yet. Software should use this bit to avoid getting an incorrect status while the first sensor update is in progress. This bit is only required if it is possible for the controller to receive and process a 'Get Sensor Reading' or 'Get Sensor Event Status' command for the sensor before the update has completed. This is most likely to be the case for sensors, such as fan RPM sensors, that may require seconds to accumulate the first reading after a re-arm. The bit is also used to indicate when a reading/state is unavailable because the management controller cannot obtain a valid reading or state for the monitored entity, typically because the entity is not present. See <i>Section 16.4, Event Status, Event Conditions, and Present State</i> and <i>Section 16.6, Re-arming</i> for more information. [4:0] - reserved. Ignore on read.

- » Sensor number from Vendor Manual
- » Sensor number 0x1B:

**Table 10: Sensor List**

Sensor Number / Name	Sensor Type (Code) / Event/Reading Type (Code)	Ass. Mask / Deass. Mask / Reading Mask	Description	Health LED Red on Error
17h / A1:Board 3.3vIPM	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	AMC Management Power (MP) 3.3V	Y
18h / A1:Board 12.0v	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	AMC Payload Power (PWR) 12V	Y
19h / A1:Board 5.0V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 5V supply	Y
1Ah / A1:Board 3.3V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 3.3V supply	Y
1Bh / A1:Pwr Good	Power supply (08h) / OEM (77h)	0000h / 0000h / 0887h	States of all power lines	N
1Ch / A1:Pwr Good Evt	Power supply (08h) / OEM (77h)	0000h / 0887h / 0887h	Power fail events for all power lines	Y

## » Read out one sensor with “Get Sensor Reading cmd“

- Build command according to spec:
  - Network/Function code: 0x04
  - Cmd: 0x2D
  - Sensor#: 0x1B

```
ipmitool raw 0x04 0x2D 0x1B
```

00 40 87 88

Byte 1 completion code (not shown by ipmitool)

Byte 2 0x00

Byte 3 0x40

Byte 4 0x87

Byte 5 0x88

- » Byte 2 0x00
  - ignore
  
- » Byte 3 0x40
  - Events disabled
  - Sensor is running

*Table 35-15, Get Sensor Reading Command*

Request Data	
Response Data	
1	sensor number (FFh = reserved)
1	Completion Code.
2	Sensor reading Byte 1: byte of reading. Ignore on read if sensor does not return an numeric (analog) reading.
3	<p>[7] - 0b = All Event Messages disabled from this sensor</p> <p>[6] - 0b = sensor scanning disabled</p> <p>[5] - 1b = reading/state unavailable (formerly "initial update in progress"). This bit is set to indicate that a 're-arm' or 'Set Event Receiver' command has been used to request an update of the sensor status, and that update has not occurred yet. Software should use this bit to avoid getting an incorrect status while the first sensor update is in progress. This bit is only required if it is possible for the controller to receive and process a 'Get Sensor Reading' or 'Get Sensor Event Status' command for the sensor before the update has completed. This is most likely to be the case for sensors, such as fan RPM sensors, that may require seconds to accumulate the first reading after a re-arm. The bit is also used to indicate when a reading/state is unavailable because the management controller cannot obtain a valid reading or state for the monitored entity, typically because the entity is not present. See <i>Section 16.4, Event Status, Event Conditions, and Present State</i> and <i>Section 16.6, Re-arming</i> for more information.</p> <p>[4:0] - reserved. Ignore on read.</p>

» Byte 4 0x87

- State 7 asserted
- State 2 asserted
- State 1 asserted
- State 0 asserted

(4)	<u>For threshold-based sensors</u> Present threshold comparison status [7:6] - reserved. Returned as 1b. Ignore on read. [5] - 1b = at or above ( $\geq$ ) upper non-recoverable threshold [4] - 1b = at or above ( $\geq$ ) upper critical threshold [3] - 1b = at or above ( $\geq$ ) upper non-critical threshold [2] - 1b = at or below ( $\leq$ ) lower non-recoverable threshold [1] - 1b = at or below ( $\leq$ ) lower critical threshold [0] - 1b = at or below ( $\leq$ ) lower non-critical threshold
	<u>For discrete reading sensors</u> [7] - 1b = state 7 asserted [6] - 1b = state 6 asserted [5] - 1b = state 5 asserted [4] - 1b = state 4 asserted [3] - 1b = state 3 asserted [2] - 1b = state 2 asserted [1] - 1b = state 1 asserted [0] - 1b = state 0 asserted

» Byte 5 0x88

- State 11 asserted

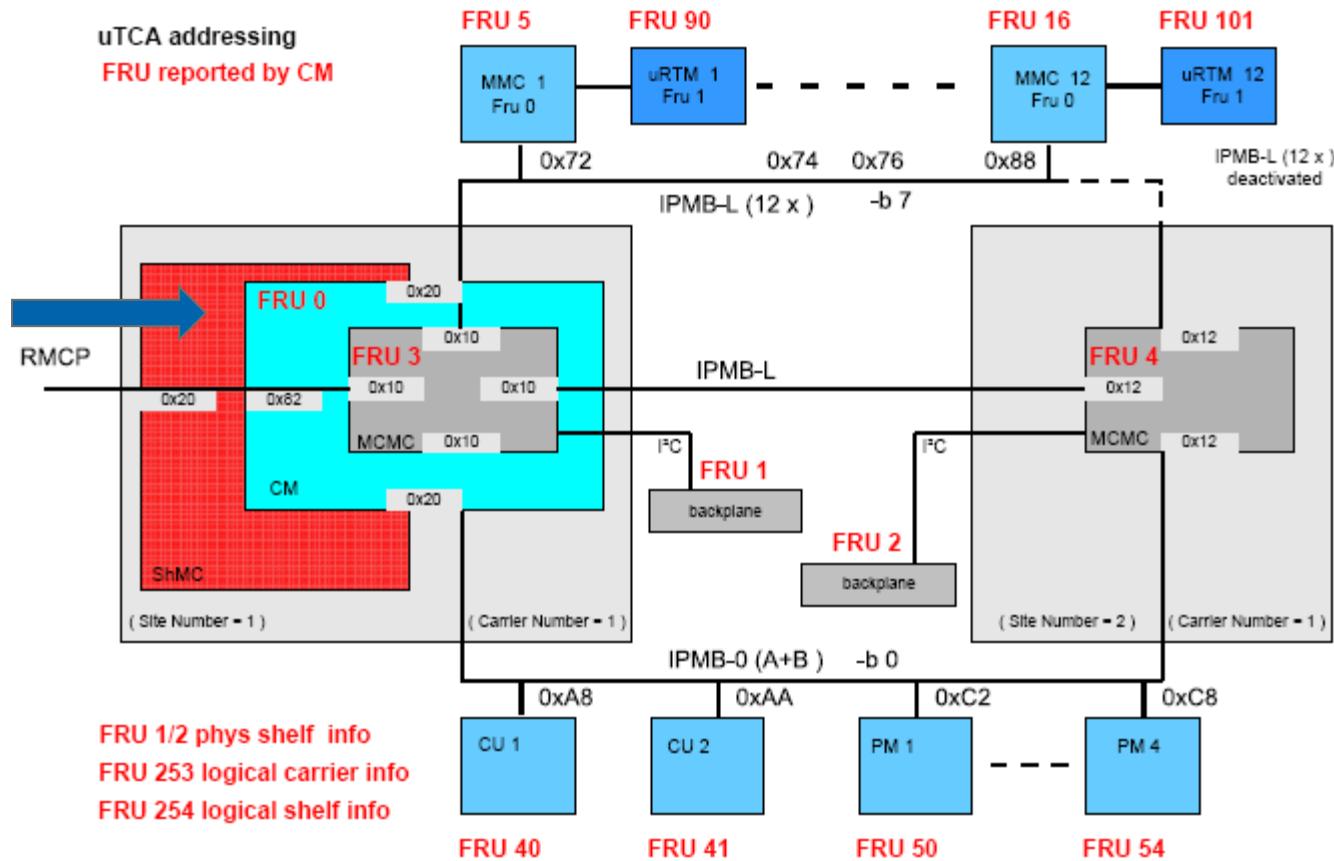
(5)	<u>For discrete reading sensors only. (Optional)</u> (00h Otherwise) [7] - reserved. Returned as 1b. Ignore on read. [6] - 1b = state 14 asserted [5] - 1b = state 13 asserted [4] - 1b = state 12 asserted [3] - 1b = state 11 asserted [2] - 1b = state 10 asserted [1] - 1b = state 9 asserted [0] - 1b = state 8 asserted
-----	--

## » Sensor definition from Vendor Manual

**Table 14: OEM Event/Reading Types (Continued)**

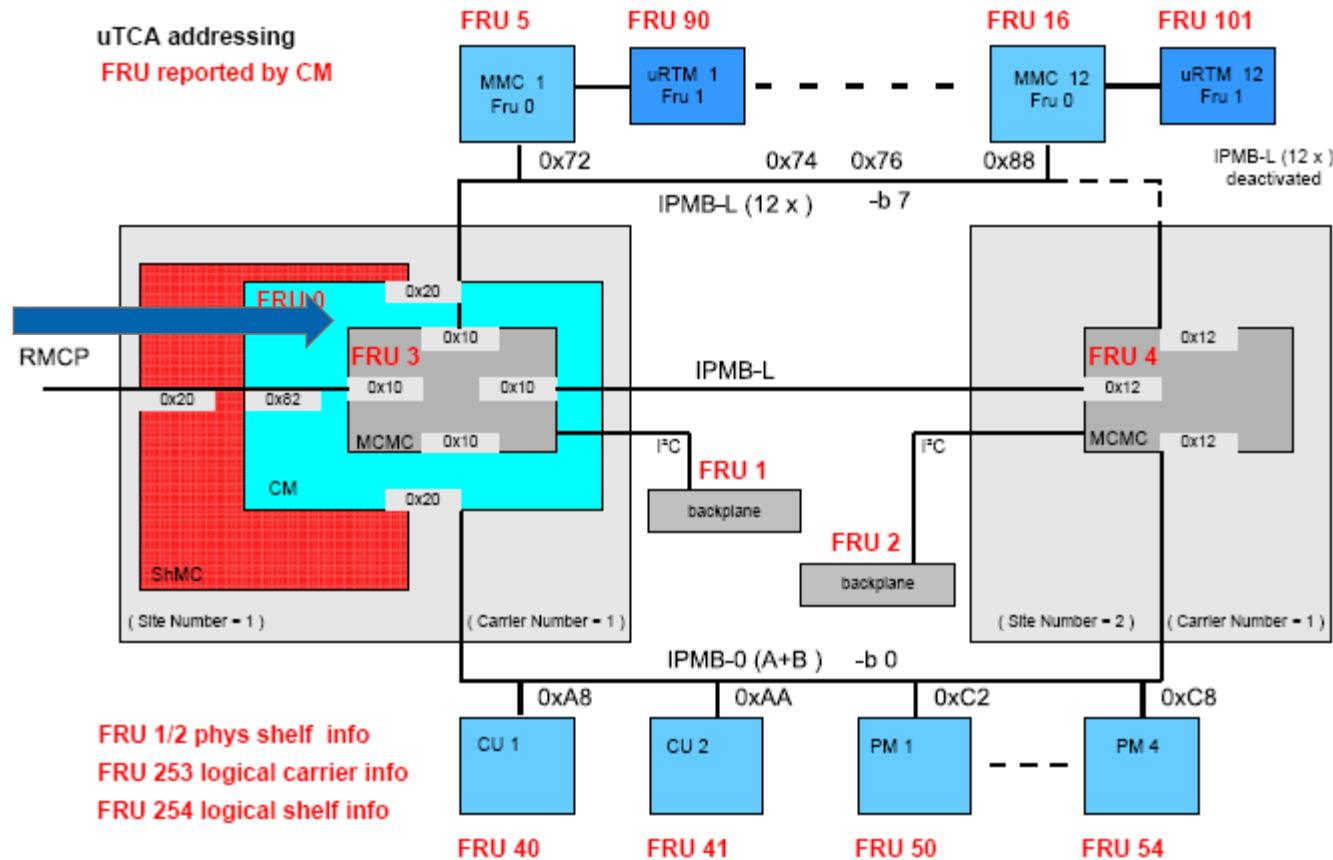
OEM SENSOR TYPE (CODE)	OEM EVENT/READING TYPE (CODE)	DESCRIPTION	
Power Supply (08h) i.e. for Power Good /	77h (OEM)	Sensor-specific Offset	Event
State 0 asserted	Power Good Event	0h	12V good (PWR)
State 1 asserted		1h	5V good
State 2 asserted		2h	3V3 good
		3h	Reserved
		4h	Reserved
		5h	Reserved
		6h	Reserved
State 7 asserted		7h	vccCore good
		8h	Reserved
		9h	Reserved
		Ah	Reserved
State 11 asserted		Bh	3V3IPMI good (MP)
		Ch	Reserved

- » Different options to access Management Controls on MCHs, AMCs, CUs, PMs
- » Via network interface to ShMC
- » Via network interface to ShMC, bridged to CM
- » Via network interface to ShMC, bridged to CM, bridged to AMC
- » Via KCS interface to local MMC



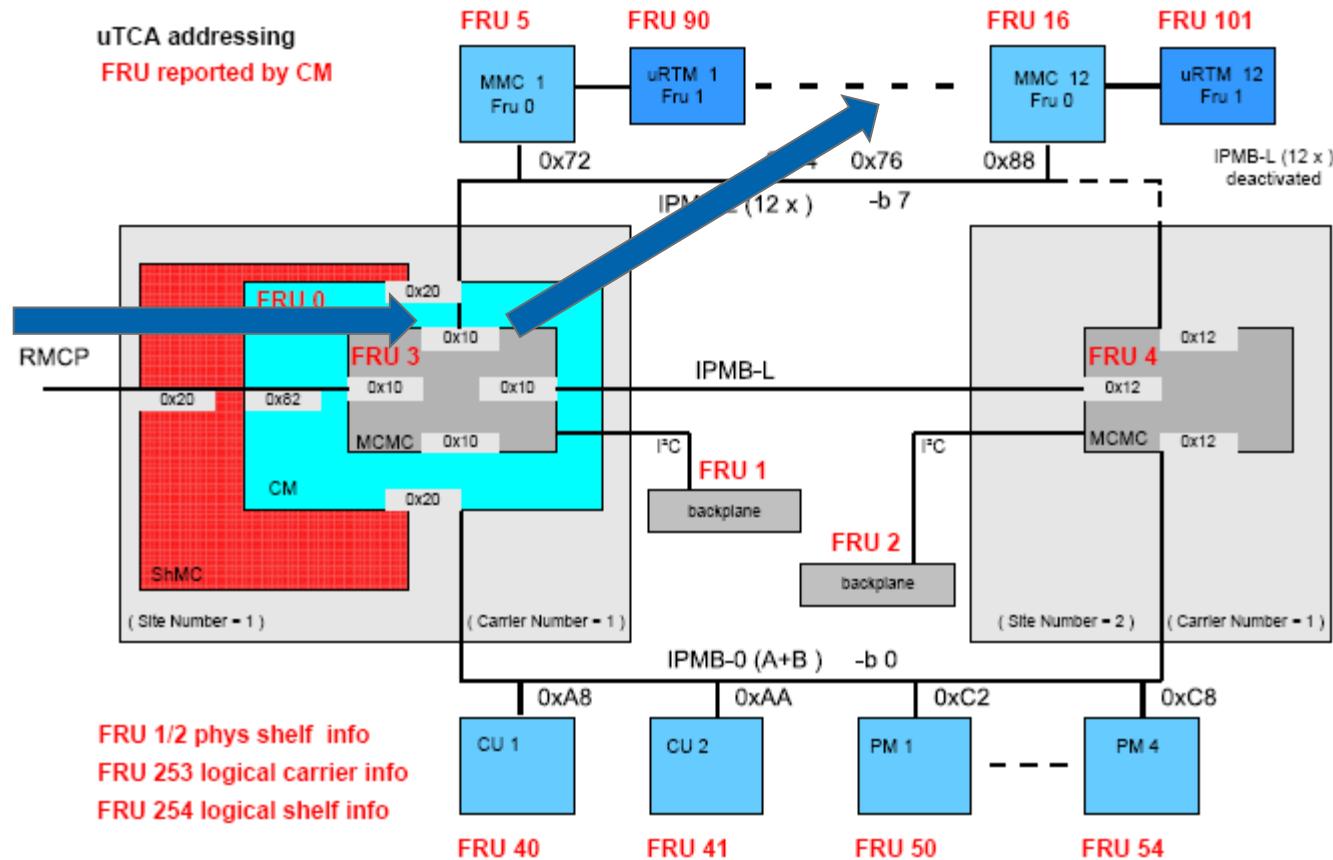
```
ipmitool -I lan -H <ip> -U admin -P admin -A PASSWORD mc info
```

Get Device ID of **ShMC** via network access



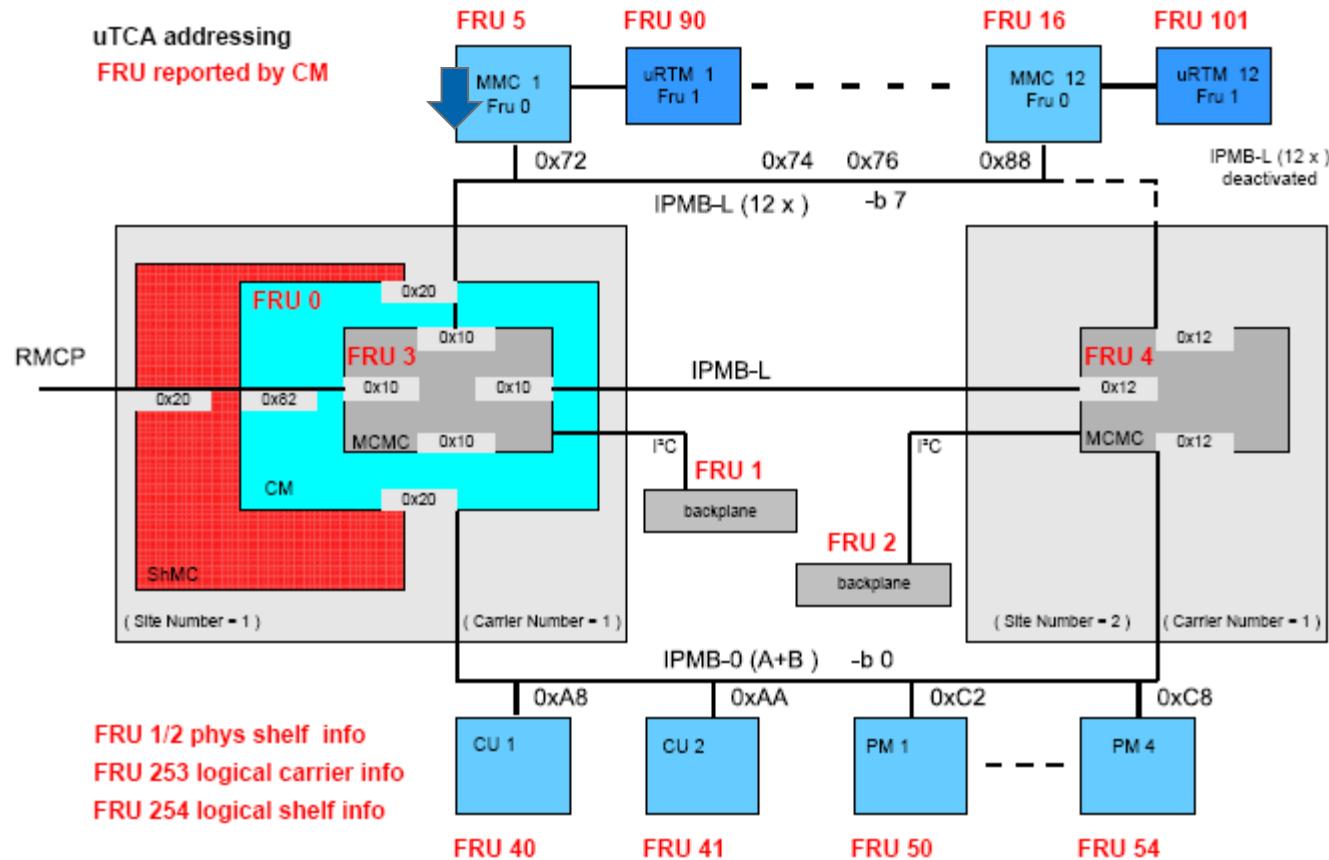
ipmitool -I lan -H <ip> -U admin -P admin -A PASSWORD **-t 0x82** mc info

Get Device ID of **CM** via network access



```
ipmitool -I lan -H <ip> -U admin -P admin -A PASSWORD -T 0x82 -t 0x76 -b 7 mc info
```

Get Device ID of AMC via network access



ipmitool –m 0x72 –t 0x72 mc info

Get Device ID via local KCS interface (payload towards MMC)

OMVIU = OM (Open Modular Platform)  
VIualisation.

- » OMVIU is a graphical tool for remote monitoring and control of MicroTCA systems



## Benefits

- » Reducing learning curve getting familiar with MicroTCA
- » Reducing development time -> saving development costs

OMVIU MicroTCA Configuration Management Software

GUI Connections Config Info Help

**Connections**

- Kontron AM4901
- Kontron AM4904
- Kontron AM4910**
  - Carrier1**
    - MCH1
    - FRU Data
    - Sensors
    - E-Keying
    - HPM.1 Update
    - FRU Control
  - AMC1**
    - FRU Data
    - Sensors
    - E-Keying
    - HPM.1 Update
    - FRU Control
  - AMC3**
    - FRU Data
    - Sensors
    - E-Keying
    - HPM.1 Update
    - FRU Control
  - AMC5**
  - AMC6**
  - PM1**
    - FRU Data
    - Sensors
  - Carrier P2P
  - SEL

**Sensor count**: 30      **LUN 0 sensors**:       **LUN 2 sensors**:   
**dynamic sensor table**:       **LUN 1 sensors**:       **LUN 3 sensors**:   
**dynamic time stamp**: 0

RecID	Nr.	Name	Type	Value/State	Threshold
9	8	A1:MMC Stor Err	Management Subsystem Health	0 1 2 3 4 5 6 7 8 9 0 1 2 3	---
11	10	A1:MMC FwUp	FWUM status	0 1 2 3 4 5 6 7 8 9 0 1 2 3	---
13	12	A1:Config Error	Type: 0xce	0 1 2 3 4 5 6 7 8 9 0 1 2 3	---
14	13	A1:Board Reset	Reset Sensor	0 1 2 3 4 5 6 7 8 9 0 1 2 3	---
15	14	<b>A1:Temp CPU</b>	Temperature	41.0	OK
16	15	A1:Temp PCH	Temperature	59.0	OK
17	16	A1:Temp MCH	Temperature	36.0	OK
23	22	A1:Temp Air	Temperature	28.0	OK
24	23	A1:Board 3.3vPPM	Voltage	3.32	OK
25	24	A1:Board 12.0v	Voltage	11.97	OK
26	25	A1:Board 5.0V	Voltage	5.09	OK
27	26	A1:Board 2.2v	Voltage	2.20	OK

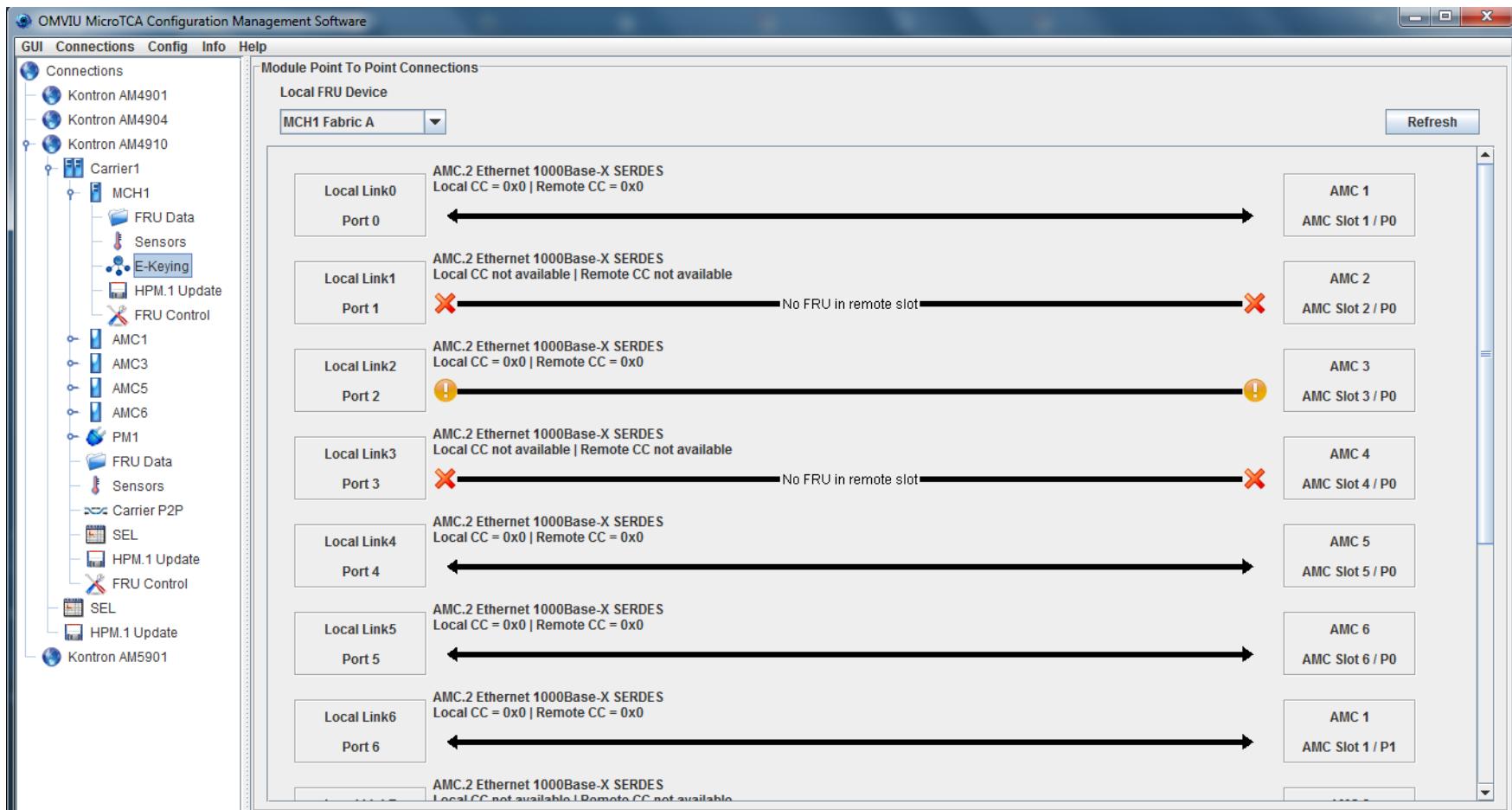
**Analog Sensor details**

**Sensor ID**: A1:Temp CPU  
**Actual Value**: 41.0 degrees C

Value	Thresholds	Hysteresis	Info	Factors	Events	HEX Dump
Lower Non Critical	not readable	degrees C				
Lower Critical	not readable	degrees C				
Lower Non Recoverable	not readable	degrees C				
Upper Non Critical	95.0	degrees C				
Upper Critical	105.0	degrees C				
Upper Non Recoverable	115.0	degrees C				

**Set Thresholds**

Reload all      Refresh all



- » Kontron MCH provides feature to examine IPMI traffic
  - IPMB-0 and IPMB-L traffic forwarded via network
  - Wireshark running on external system decodes IPMI data
- Wireshark:
  - Listen on network interface to all traffic
  - Set filter ipmi.session.id and ipmi.header.command != 0x28
- MCH, clicm:
  - clicm monitor ip <ip-wireshark-system>
  - clicm monitor ip enable
  - clicm monitor ip disable

4022\_clkekpbpl\_4022\_slot3.bin - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	I2C-1	0x3c	IPMI/ATC	Req, [ATCA] Get PICMG Properties, seq 0x0f
2	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Get PICMG Properties, seq 0x0f
3	0.000000	I2C-1	0x3b	IPMI/ATC	Req, [ATCA] Get PICMG Properties, seq 0x10
4	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Get PICMG Properties, seq 0x10
5	0.000000	I2C-1	0x3d	IPMI/ATC	Req, [ATCA] Get PICMG Properties, seq 0x11
6	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Get PICMG Properties, seq 0x11
7	0.000000	I2C-1	0x3e	IPMI/ATC	Req, [ATCA] Get PICMG Properties, seq 0x12
8	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Get PICMG Properties, seq 0x12
9	0.000000	I2C-1	0x3c	IPMI/ATC	Req, [ATCA] Get PICMG Properties, seq 0x13
10	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Get PICMG Properties, seq 0x13
11	0.000000	I2C-1	0x3b	IPMI/ATC	Req, [ATCA] Get PICMG Properties, seq 0x14
12	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Get PICMG Properties, seq 0x14
13	0.000000	I2C-1	0x10	IPMI/ATC	Req, Platform Event, seq 0x31
14	0.000000	I2C-1	0x3b	IPMI/ATC	Rsp, Platform Event, seq 0x31
15	0.000000	I2C-1	0x3b	IPMI/ATC	Req, [ATCA] Set FRU LED State, seq 0x15
16	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Set FRU LED State, seq 0x15
17	0.000000	I2C-1	0x3d	IPMI/ATC	Req, [ATCA] Get PICMG Properties, seq 0x16
18	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Get PICMG Properties, seq 0x16
19	0.000000	I2C-1	0x3e	IPMI/ATC	Req, [ATCA] Get PICMG Properties, seq 0x17
20	0.000000	I2C-1	0x10	IPMI/ATC	Rsp, [ATCA] Get PICMG Properties, seq 0x17
21	0.000000	I2C-1	0x3b	IPMI/ATC	Req, [ATCA] Set FRU LED State, seq 0x18

# Frame 15 (13 bytes on wire, 13 bytes captured)  
 # Inter-Integrated Circuit (Data)  
 # Intelligent Platform Management Interface  
 # Response in: 161  
 # Header: [ATCA] Set FRU LED State (Request) from 0x20 to 0x76  
 # Data  
 FRU ID: 0  
 LED ID: 0  
 Function: LED Blinking override, off-duration 900ms (0x5a)  
 On-duration: 100ms  
 # Color: Do not change  
 Data checksum: 0x13 (correct)

```
0000  76 b0 da 20 54 07 00 00 00 5a 0a 0e 13      V.. T... .Z...
```

Frame (frame), 13 bytes | Packets: 666 Displayed: 666 Marked: 0 | Profile: Default



[www.kontron.com](http://www.kontron.com)

Munich/Eching Kaufbeuren Deggendorf Toulon Copenhagen Chichester Linz/Hagenberg Solothurn Brussels Liberec Pilsen Warsaw  
San Diego/Poway Columbia Fremont Montreal Beijing Hong Kong Taipei Penang Bangalore Mumbai Moscow Kiev Sydney