» User Guide «

IPMI Firmware User Guide

for the

AM5030

CPU Module

Doc.ID: 1042-7364, Rev. 1.0
October 22, 2010

Revision History

Publication Title:	IPMI Firmware User Guide for the AM5030 CPU Module	
Doc. ID:	1042-7364	
Rev.	**Brief Description of Changes**	**Date of Issue**
1.0	Initial issue	22-Oct-2010

Imprint

Kontron Modular Computers GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Modular Computers GmbH

Sudetenstraße 7

D-87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

+49(0)800-SALESKONTRON

sales@kontron.com

For information about other Kontron products, please visit our Internet website: **www.kontron.com**

Disclaimer

Table of Contents

This page has been intentionally left blank.

List of Tables

This page has been intentionally left blank.

1. Introduction

1.1 Terminology and Acronym Definitions

The following table provides descriptions for terms and acronyms used in this guide. The descriptions are derived primarily from the IPMI specifications.

Table 1: Terminology and Acronym Definitions

TERM or ACRONYM	DESCRIPTION
AMC	Advanced Mezzanine Card
BSP	Board Support Package
DMI	Desktop Management Interface
FRU	Field Replaceable Unit
FWH	Firmware Hub
I^2C	Inter-Integrated Circuit
IPMB	Intelligent Platform Management Bus
IPMB-0	AdvancedTCA shelf-level IPMB
IPMB-L	Local, on-carrier IPMB that links the carrier IPMC with the MMCs of installed modules
IPMC	Intelligent Platform Management Controller located on AMC carrier
IPMI	Intelligent Platform Management Interface
IOL	IPMI over LAN. An MMC is accessed via LAN, not IPMB
KCS	Keyboard Controller Style
MMC	Module Management Controller – an IPMI controller located on the AMC module
MP	Management Power
PICMG	PCI Industrial Computer Manufacturer Group
PWR	Payload Power
SDR	Sensor Data Record
SDRR	Sensor Data Record Repository
SEL	System Event Log
SMBIOS	System Management BIOS
SMS	System Management Software (designed to run under the OS)
SOL	Serial over LAN. A serial interface is redirected by LAN using the RMCP+ protocol.

1.2 Related Publications

The following publications contain information relating to this product.

Table 2: Related Publications

PRODUCT	PUBLICATION
IPMI	IPMI Specification V2.0
IPMI	IPMI- Platform Management FRU Information Storage Definition v1.0, Document Revision 1.1
MicroTCA	PICMG® MTCA.0 Micro Telecommunications Computing Architecture R1.0
AMC	PICMG® AMC.0, Advanced Mezzanine Card Specification R2.0 PICMG® AMC.1, PCI Express R2.0 PICMG® AMC.2, Gigabit Ethernet R1.0 PICMG® AMC.3, Storage Interfaces R1.0
AM5030	AM5030 User Guide, ID: 1036-3302, Rev. 1.0 AM5030 uEFI BIOS User Guide, ID: 1037-1209, Rev. 1.0
AM5030	AM5030 Linux Board Support Package
IPMI Tools	“ipmitool” documentation: http://ipmitool.sourceforge.net
IPMI Tools	OpenIPMI documentation: http://www.openipmi.sourceforge.net

As a hot-swappable field replaceable unit (FRU), the AM5030 follows the stringent carrier grade RASM feature set, namely - Reliability, Availability, Serviceability, Maintainability.

Built in accordance to the AMC.0 specification, the AM5030 is also AMC.1, AMC 2, and AMC.3 compliant and is easily managed via IPMI v1.5/v2.0.

As with every Advanced Mezzanine Card (AMC) the AM5030 is equipped with a Module Management Controller (MMC).

1.3 IPMI in AdvancedMC / AdvancedTCA Environment

The Module Management Controller is a crucial component of any AMC module. Besides acting as a regular IPMI management controller (sensor monitoring, event logging, etc.), it also provides an interface to all necessary data related to module power requirements and implemented interfaces (E-Keying). Further, it plays an active role in the module hot swap state management. The carrier IPMI Controller (IPMC) communicates with the MMC using the local IPMB (IPMB-L) bus. In an ATCA/AMC environment, it is the IPMC that actually turns on/off module (payload) power. However, before the IPMC enables the module payload power, various criteria must be satisfied by both the carrier and the module, including power requirements and capabilities, matching interfaces, current module hot swap state, and any other special conditions as specified by the Shelf Manager policy.

1.4 Module Management Controller Hardware

On the AM5030 processor AMC module, the MMC is implemented using the NXP LPC2368 microcontroller with 512 kB of internal flash and 56 kB of RAM.

An external 64 kB serial EEPROM chip is used for firmware private data and for FRU Inventory storage. An additional external 2 MB serial SPI-Flash is used for redundant firmware image storage.

The Module Management Controller implements one local Keyboard Style Interface (KCS) with interrupt support for communication with system side management software and the uEFI BIOS. The IPMB-L bus is used for interconnection with the IPMC.

IPMI over LAN (IOL) as well as Serial Over LAN (SOL) is supported on all four Ethernet channels of the module. SOL is only available on one Ethernet channel at a time.

The Module Management Controller provides access to various board sensors which permit the monitoring of:

- System power voltages: +12V (PWR), +5V, +3.3V, +3.3V (MP)
- Temperatures: inlet and outlet near AMC edge-connectors
- Power Good, LAN links, IPMB link, board reset, post code, boot error, CPU States (processor hot, thermal trip, …), IPMB-L state, Health error, IPMI watchdog etc.

2. MMC Firmware

2.1 Key Features

The following are key features of the AM5030 MMC Firmware:

- Compliant with the related IPMI and PICMG® specifications
- Firmware designed and specially made for AdvancedMC environments (ATCA, µTCA)
- Supports one KCS interface with interrupt support
- Supports the local IPMB (IPMB-L) interface
- Out-of-Band management and monitoring using IPMB-L interface permits access to sensors regardless of module CPU state
- Sensor thresholds fully configurable
- Sensor names prefixed with AMC module Bay ID (A1…4, B1…4)
- Usable in µTCA slots 1…12. Sensor names for slots 9…12 are prefixed with C1…C4
- Complete IPMI watchdog functionality
- Complete FRU functionality
- Firmware can be updated in the field
- Two firmware banks implemented, firmware bank management is done by the open tool “ipmitool” (functions “hpm” or “fwum”)
- Down loading new firmware image does not break currently running firmware activities
- Manual and automatic firmware image roll-back in case of upgrade failure

- Interoperable with other AMC, ATCA, or IPMI solutions
- uEFI BIOS fail-over control, for automatic EFI firmware bank switching after having detected a not working EFI
- OEM commands for uEFI BIOS firmware bank selection and uEFI BIOS boot order override
- IPMI over LAN (IOL) support
- Serial over LAN (SOL) support
- Graceful shutdown support
- The “Health” LED shows MMC's heartbeat and pulses on KCS interface traffic

2.2 Supported IPMI and ATCA Commands

2.2.1 Standard IPMI Commands

Part of the command list in IPMI specification 2.0

M = mandatory, O = optional

Table 3: Standard IPMI Commands

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
IPM DEVICE “GLOBAL” COMMANDS				M
Get Device ID	20.1	App	01h	M / Yes
Cold Reset	20.2	App	02h	O / Yes
Warm Reset	20.3	App	03h	O / No
Get Self Test Results	20.4	App	04h	O / Yes
Manufacturing Test On	20.5	App	05h	O / No
Set ACPI Power State	20.6	App	06h	O / No
Get ACPI Power State	20.7	App	07h	O / No
Get Device GUID	20.8	App	08h	O / No
Broadcast “Get Device ID”	20.9	App	01h	M / Yes
BMC WATCHDOG TIMER COMMANDS				O
Reset Watchdog Timer	27.5	App	22h	O / Yes
Set Watchdog Timer	27.6	App	24h	O / Yes
Get Watchdog Timer	27.7	App	25h	O / Yes

Table 3: Standard IPMI Commands

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
BMC DEVICE AND MESSAGING COMMANDS				O
Set BMC Global Enables	22.1	App	2Eh	O / Yes
Get BMC Global Enables	22.2	App	2Fh	O / Yes
Clear Message Flags	22.3	App	30h	O / Yes
Get Message Flags	22.4	App	31h	O / Yes
Enable Message Channel Receive	22.5	App	32h	O / Yes
Get Message	22.6	App	33h	O / Yes
Send Message	22.7	App	34h	O / Yes
Read Event Message Buffer	22.8	App	35h	O / Yes
Get BT Interface Capabilities	22.9	App	36h	O / No
Get System GUID	22.14	App	37h	O / No
Get Channel Authentication Capabilities	22.13	App	38h	O / Yes
Get Session Challenge	22.15	App	39h	O / Yes
Activate Session	22.17	App	3Ah	O / Yes
Set Session Privilege Level	22.18	App	3Bh	O / Yes
Close Session	22.19	App	3Ch	O / Yes
Get Session Info	22.20	App	3Dh	O / Yes
Get AuthCode	22.21	App	3Fh	O / No
Set Channel Access	22.22	App	40h	O / Yes
Get Channel Access	22.23	App	41h	O / Yes
Get Channel Info	22.24	App	42h	O / Yes
Set User Access	22.26	App	43h	O / Yes
Get User Access	22.27	App	44h	O / Yes
Set User Name	22.28	App	45h	O / Yes
Get User Name	22.29	App	46h	O / Yes
Set User Password	22.30	App	47h	O / Yes

Table 3: Standard IPMI Commands

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
Activate Payload	24.1	App	48h	O / Yes
Deactivate Payload	24.2	App	49h	O / Yes
Get Payload Activation Status	24.4	App	4Ah	O / Yes
Get Payload Instance Info	24.5	App	4Bh	O / Yes
Set User Payload Access	24.6	App	4Ch	O / Yes
Get User Payload Access	24.7	App	4Dh	O / Yes
Get Channel Payload Support	24.8	App	4Eh	O / Yes
Get Channel Payload Version	24.9	App	4Fh	O / Yes
Get Channel OEM Payload Info	24.10	App	50h	O / No
Master Write-Read	22.11	App	52h	O / No
Get Channel Cipher Suits	22.15	App	54h	O / No
Suspend/Resume Payload Encryption	24.3	App	55h	O / Yes
Set Channel Security Keys	22.25	App	56h	O / No
Get System Interface Capabilities	22.9	App	57h	O / No
CHASSIS DEVICE COMMANDS				O
Get Chassis Capabilities	28.1	Chassis	00h	O / Yes
Get Chassis Status	28.2	Chassis	01h	O / Yes
Chassis Control	28.3	Chassis	02h	O / Yes
Chassis Reset	28.4	Chassis	03h	O / No
Chassis Identify	28.5	Chassis	04h	O / No
Set Chassis Capabilities	28.7	Chassis	05h	O / No
Set Power Restore Policy	28.8	Chassis	06h	O / No
Get System Restart Cause	28.11	Chassis	07h	O / No
Set System Boot Options	28.12	Chassis	08h	O / No
Get System Boot Options	28.13	Chassis	09h	O / No
Get POH Counter	28.14	Chassis	0Fh	O / Yes

Table 3: Standard IPMI Commands

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
EVENT COMMANDS				M
Set Event Receiver	29.1	S/E	01h	M / Yes
Get Event Receiver	29.2	S/E	02h	M / Yes
Platform Event (a.k.a. “Event Message”)	29.3	S/E	03h	M / Yes
PEF AND ALERTING COMMANDS				O
Get PEF Capabilities	30.1	S/E	10h	O / No
Arm PEF Postpone Timer	30.2	S/E	11h	O / No
Set PEF Configuration Parameters	30.3	S/E	12h	O / No
Get PEF Configuration Parameters	30.4	S/E	13h	O / No
Set Last Processed Event ID	30.5	S/E	14h	O / No
Get Last Processed Event ID	30.6	S/E	15h	O / No
Alert Immediate	30.7	S/E	16h	O / No
PET Acknowledge	30.8	S/E	17h	O / No
SENSOR DEVICE COMMANDS				M
Get Device SDR Info	35.2	S/E	20h	M / Yes
Get Device SDR	35.3	S/E	21h	M / Yes
Reserve Device SDR Repository	35.4	S/E	22h	M / Yes
Get Sensor Reading Factors	35.5	S/E	23h	O / No
Set Sensor Hysteresis	35.6	S/E	24h	O / Yes
Get Sensor Hysteresis	35.7	S/E	25h	O / Yes
Set Sensor Threshold	35.8	S/E	26h	O / Yes
Get Sensor Threshold	35.9	S/E	27h	O / Yes
Set Sensor Event Enable	35.10	S/E	28h	O / Yes
Get Sensor Event Enable	35.11	S/E	29h	O / Yes
Re-arm Sensor Events	35.12	S/E	2Ah	O / No

Table 3: Standard IPMI Commands

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
Get Sensor Event Status	35.13	S/E	2Bh	O / No
Get Sensor Reading	35.14	S/E	2Dh	M / Yes
Set Sensor Type	35.15	S/E	2Eh	O / No
Get Sensor Type	35.16	S/E	2Fh	O / No
FRU DEVICE COMMANDS				M
Get FRU Inventory Area Info	34.1	Storage	10h	M / Yes
Read FRU Data	34.2	Storage	11h	M / Yes
Write FRU Data	34.3	Storage	12h	M / Yes
SDR DEVICE COMMANDS				O
Get SDR Repository Info	33.9	Storage	20h	O / No
Get SDR Repository Allocation Info	33.10	Storage	21h	O / No
Reserve SDR Repository	33.11	Storage	22h	O / No
Get SDR	33.12	Storage	23h	O / No
Add SDR	33.13	Storage	24h	O / No
Partial Add SDR	33.14	Storage	25h	O / No
Delete SDR	33.15	Storage	26h	O / No
Clear SDR Repository	33.16	Storage	27h	O / No
Get SDR Repository Time	33.17	Storage	28h	O / No
Set SDR Repository Time	33.18	Storage	29h	O / No
Enter SDR Repository Update Mode	33.19	Storage	2Ah	O / No
Exit SDR Repository Update Mode	33.20	Storage	2Bh	O / No
Run Initialization Agent	33.21	Storage	2Ch	O / No
SEL DEVICE COMMANDS				O
Get SEL Info	40.2	Storage	40h	O / No
Get SEL Allocation Info	40.3	Storage	41h	O / No

Table 3: Standard IPMI Commands

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
Reserve SEL	40.4	Storage	42h	O / No
Get SEL Entry	40.5	Storage	43h	O / No
Add SEL Entry	40.6	Storage	44h	O / No
Partial Add SEL Entry	40.7	Storage	45h	O / No
Delete SEL Entry	40.8	Storage	46h	O / No
Clear SEL	40.9	Storage	47h	O / No
Get SEL Time	40.10	Storage	48h	O / No
Set SEL Time	40.11	Storage	49h	O / No
Get Auxiliary Log Status	40.12	Storage	5Ah	O / No
Set Auxiliary Log Status	40.13	Storage	5Bh	O / No
LAN DEVICE COMMANDS				O
Set LAN Configuration Parameters	23.1	Transport	01h	O / Yes
Get LAN Configuration Parameters	23.2	Transport	02h	O / Yes
Suspend BMC ARPs	23.3	Transport	03h	O / Yes
Get IP/UDP/RMCP Statistics	23.4	Transport	04h	O / Yes
SERIAL/MODEM DEVICE COMMANDS				O
Set Serial/Modem Configuration	25.1	Transport	10h	O / No
Get Serial/Modem Configuration	25.2	Transport	11h	O / No
Set Serial/Modem Mux	25.3	Transport	12h	O / No
Get TAP Response Codes	25.4	Transport	13h	O / No
Set PPP UDP Proxy Transmit Data	25.5	Transport	14h	O / No
Get PPP UDP Proxy Transmit Data	25.6	Transport	15h	O / No
Send PPP UDP Proxy Packet	25.7	Transport	16h	O / No
Get PPP UDP Proxy Receive Data	25.8	Transport	17h	O / No
Serial/Modem Connection Active	25.9	Transport	18h	O / No

Table 3: Standard IPMI Commands

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
Callback	25.10	Transport	19h	O / No
Set User Callback Options	25.11	Transport	1Ah	O / No
Get User Callback Options	25.12	Transport	1Bh	O / No
SOL Activating	26.1	Transport	20h	O / Yes
Get SOL Configuration Parameters	26.2	Transport	21h	O / Yes
Set SOL Configuration Parameters	26.3	Transport	22h	O / Yes
BRIDGE MANAGEMENT COMMANDS (ICMB)				O
Get Bridge State	[ICMB]	Bridge	00h	O / No
Set Bridge State	[ICMB]	Bridge	01h	O / No
Get ICMB Address	[ICMB]	Bridge	02h	O / No
Set ICMB Address	[ICMB]	Bridge	03h	O / No
Set Bridge Proxy Address	[ICMB]	Bridge	04h	O / No
Get Bridge Statistics	[ICMB]	Bridge	05h	O / No
Get ICMB Capabilities	[ICMB]	Bridge	06h	O / No
Clear Bridge Statistics	[ICMB]	Bridge	08h	O / No
Get Bridge Proxy Address	[ICMB]	Bridge	09h	O / No
Get ICMB Connector Info	[ICMB]	Bridge	0Ah	O / No
Get ICMB Connection ID	[ICMB]	Bridge	0Bh	O / No
Send ICMB Connection ID	[ICMB]	Bridge	0Ch	O / No
DISCOVERY COMMANDS (ICMB)				O
Prepare For Discovery	[ICMB]	Bridge	10h	O / No
Get Addresses	[ICMB]	Bridge	11h	O / No
Set Discovered	[ICMB]	Bridge	12h	O / No
Get Chassis Device ID	[ICMB]	Bridge	13h	O / No
Set Chassis Device ID	[ICMB]	Bridge	14h	O / No

Table 3: Standard IPMI Commands

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
BRIDGING COMMANDS (ICMB)				O
Bridge Request	[ICMB]	Bridge	20h	O / No
Bridge Message	[ICMB]	Bridge	21h	O / No
EVENT COMMANDS (ICMB)				O
Get Event Count	[ICMB]	Bridge	30h	O / No
Set Event Destination	[ICMB]	Bridge	31h	O / No
Set Event Reception State	[ICMB]	Bridge	32h	O / No
Send ICMB Event Message	[ICMB]	Bridge	33h	O / No
Get Event Destination	[ICMB]	Bridge	34h	O / No
Get Event Reception State	[ICMB]	Bridge	35h	O / No
OEM COMMANDS FOR BRIDGE NETFN				O
OEM Commands	[ICMB]	Bridge	C0h-FEh	O / No
OTHER BRIDGE COMMANDS				O
Error Report	[ICMB]	Bridge	FFh	O / No

2.2.2 AdvancedTCA and AMC Commands

Part of the command list in PICMG 3.0 R 2.0 AdvancedTCA Base Specification and the PICMG AMC.0 Advanced Mezzanine Card Specification, R 1.0,

M = mandatory, O = optional

Table 4: AdvancedTCA and AMC Commands

COMMAND	PICMG 3.0 SPEC. TABLE	NETFN	CMD	KONTRON SUPPORT ON MMC
AdvancedTCA				M
Get PICMG Properties	3-9	PICMG	00h	M / Yes
Get Address Info	3-8	PICMG	01h	N/A
Get Shelf Address Info	3-13	PICMG	02h	N/A
Set Shelf Address Info	3-14	PICMG	03h	N/A
FRU Control	3-22	PICMG	04h	M / Yes [1]
Get FRU LED Properties	3-24	PICMG	05h	M / Yes
Get LED Color Capabilities	3-25	PICMG	06h	M / Yes
Set FRU LED State	3-26	PICMG	07h	M / Yes
Get FRU LED State	3-27	PICMG	08h	M / Yes
Set IPMB State	3-51	PICMG	09h	N/A
Set FRU Activation Policy	3-17	PICMG	0Ah	N/A
Get FRU Activation Policy	3-18	PICMG	0Bh	N/A
Set FRU Activation	3-16	PICMG	0Ch	N/A
Get Device Locator Record ID	3-29	PICMG	0Dh	M / Yes
Set Port State	3-41	PICMG	0Eh	N/A
Get Port State	3-42	PICMG	0Fh	N/A
Compute Power Properties	3-60	PICMG	10h	N/A
Set Power Level	3-62	PICMG	11h	N/A
Get Power Level	3-61	PICMG	12h	N/A
Renegotiate Power	3-66	PICMG	13h	N/A
Get Fan Speed Properties	3-63	PICMG	14h	N/A
Set Fan Level	3-65	PICMG	15h	N/A

Table 4: AdvancedTCA and AMC Commands

COMMAND	PICMG 3.0 SPEC. TABLE	NETFN	CMD	KONTRON SUPPORT ON MMC
Get Fan Level	3-64	PICMG	16h	N/A
Bused Resource	3-44	PICMG	17h	N/A
Get IPMB Link Info	3-49	PICMG	18h	N/A
AMC	**AMC.0 TABLE**			
Set AMC Port State	3-27	PICMG	19h	O / Yes
Get AMC Port State	3-28	PICMG	20h	O / Yes
Set Clock State	3-44	PICMG	2Ch	O / Yes
Get Clock State	3-45	PICMG	2Dh	O / Yes

[1] Only “FRU Control - Cold Reset” and “FRU Control - Quiesce” are supported.

3. OEM Commands and Command Extensions

3.1 Get Device ID Command with OEM Extensions

The IPMI specification defines four optional bytes in the response to 'Get Device ID'. The response bytes [13:16] hold the 'Auxiliary Firmware Revision Information'.

Table 5: Get Device ID Command with OEM Extensions

COMMAND	LUN	NetFn	CMD
Get Device ID command with OEM extensions	00h	App = 06h	01h
REQUEST DATA			
Byte	Data Field		
-	-		
RESPONSE DATA			
Byte	Data Field		
1	Completion Code		
2:12	Regular Get Device ID Command response fields		
13	Release number of the management controller firmware: 10h for R10, 11h for R11, ...		
14	Module Geographical Address (site number): 1 ... 8 = Module in AMC bay A1, A2, A3, A4, B1, B2, B3, B4 or in µTCA slot 1 ... 8 with bus addresses 72h, 74h, 76h, 78h, 7ah, 7ch, 7eh, 80h 9 ...12 = Module in µTCA slot 9 ... 12 = Bay C1, C2, C3, C4 with bus addresses 82h, 84h, 86h,88h 0, > 12 = Module position is not in range. The IPMB-L bus is switched off.		
15	Reserved		
16	Reserved		

3.2 Set Control State (Firmware Hub, Boot Order)

Table 6: Set Control State

COMMAND	LUN	NetFn	CMD
Set Control State (Firmware Hub, Boot Order)	00h	OEM = 3Eh	20h
REQUEST DATA			
Byte	Data Field		
1	Control ID: 00h = Firmware Hub (BIOS Flash) selection 9Dh = BIOS Boot Order Configuration		
2	Control State for BIOS Flash selection: (These settings are stored in EEPROM and applied (to logic) each time the IPMI controller detects power-on) 00h = BIOS Flash selection is not inverted 01h = BIOS Flash selection is logically inverted Please note that this selection will be automatically toggled by the IPMI controller during a failing boot process. Other payload sided settings may additionally modify this selection. Control State for BIOS Boot Order Configuration: (These settings are stored in EEPROM and applied (to logic) each time the IPMI controller detects power-on) 00h = No override, boot as usual 01h = Next boot device is: Floppy 02h = Next boot device is: HDD 03h = Next boot device is: CD 04h = Next boot device is: Network 05h = Next boot device is: USB Floppy 06h = Next boot device is: USB HDD 07h = Next boot device is: USB CDROM		
RESPONSE DATA			
Byte	Data Field		
1	Completion Code		

3.3 Get Control State (Firmware Hub, Boot Order)

Table 7: Get Control State

COMMAND	LUN	NetFn	CMD
Get Control State (Firmware Hub, Boot Order)	00h	OEM = 3Eh	21h
REQUEST DATA			
Byte	Data Field		
1	Control ID: 00h = Firmware Hub (BIOS Flash) selection 9Dh = BIOS Boot Order Configuration		
RESPONSE DATA			
Byte	Data Field		
1	Completion Code		
4	Control State (refer to 3.2) 00h .. 01h for control ID = Firmware Hub Flash Select 00h .. FFh for control ID = Boot Order Configuration		

3.4 OEM Module Quiescence Feedback

3.4.1 Overview

If the OS doesn't fully support ACPI this command provides support to control a graceful shut down of the AM5030. There is a Graceful Reboot and Shutdown Daemon (grnsd) for Linux included in the newest Linux BSPs being offered by Kontron. It works as described in the following “Usage for a self written shut down daemon” below.

If ACPI is fully supported this command can be used to set a timeout time for the case that the ACPI means (ACPI daemon etc.) are unable to shut down the system in time. As a default value at system start this time is set to 0 (endless wait).

For further information, refer to 9, Hot Swap.

3.4.2 Usage for a Self Written Shut Down Daemon

This command normally is used by a shut down daemon in a non-ACPI supporting software environment. If a timeout time has to be set to avoid an endless waiting for the sleep state the daemon calls this command after system start with the “set quiesce wait timeout” bit set and the “Quiesce wait timeout” time <> 0. Afterwards the daemon calls this command cyclically with the “OS daemon present” bit set. When the MMC gets a FRU Control (Quiesce) request from the carrier (e.g. during a Hot Swap sequence) it sets the “quiesce request (FRU Control)” bit in its command response. After the daemon sees this bit set in the response it should shut down the system. After having set the “quiesce request (FRU Control)” bit the MMC starts the timeout timer (if a timeout time was defined) and monitors the sleep signal line to recognize the sleep state which should be caused by the shut down. When the MMC detects the sleep state (signal) or it receives a command with the “quiescence acknowledge” bit set or the timeout timer has expired, the MMC sends a “Module Hot Swap event” message to the carrier, and in the following the payload power will be switched off.

3.4.3 If no Daemon is Announced as Present

If no command call announces that a daemon is present, the MMC automatically uses the default timeout time 0 (endless wait) during the Hot Swap process. But if the timeout time was set to a value 1…255 this time will be used in any case while waiting for the sleep state (signal).

COMMAND	LUN	NetFn	CMD
OEM Module Quiescence Feedback	00h	OEM = 3Eh	40h
REQUEST DATA			
Byte	Data Field		
1	Control bits: [7] - 1b = set quiesce wait timeout [6] - 1b = quiescence acknowledge (OS ready) [5] - 1b = OS daemon present [4:0] Reserved		
2	Quiesce wait timeout [sec] a) An OS daemon is present (refer to bits above): This is the maximum time from the moment on that the MMC receives FRU Control (Quiesce) request until it sends back the appropriate Module Hot Swap event message. b) No OS daemon is present (refer to bits above): This is the maximum time from the moment on that the MMC receives FRU Control (Quiesce) request until it sends back the appropriate Module Hot Swap event message. If sleep state is recognized before timeout then the Module Hot Swap event message will be sent immediately. If the time is set to 0 (default after reset) then the Module Hot Swap event message will only be sent after recognition of sleep state (signal).		
RESPONSE DATA			
Byte	Data Field		
1	Completion Code		
2	Control bits: [7] - Reserved [6] - 1b = quiescence acknowledge (OS ready) [5] - 1b = OS daemon present [4] - 1b = quiesce request (FRU Control) [3] - Reserved [2] - 1b = graceful reboot request (FRU Control) [1] - 1b = quiescence reached (MMC acknowledge) [0] - 1b = module hot swap switch opened		
4	Quiesce wait timeout (valid only if OS daemon present = 1)		

Settings changed with this command are volatile (in particular quiesce timeout and OS daemon present). Bits [6:5] are always settable, but once the quiesce request comes they cannot be cleared until quiescence state is entered and exited.

4. Sensors Implemented on the AM5030

The MMC includes many sensors for voltage or temperature monitoring and various others for pass/fail type signal monitoring.

Every sensor is associated with a Sensor Data Record (SDR). Sensor Data Records contain information about the sensors identification such as sensor type, sensor name, sensor unit. SDRs also contain the configuration of a specific sensor such as threshold, hysteresis or event generation capabilities that specify sensor's behavior. Some fields of the sensor SDR are configurable using IPMI commands others are always set to built-in default values.

Finally, one field, which is the sensor owner, must reflect the module addresses that allow the AMC Carrier to identify the owner of the sensor when it is scanned and merged into the AMC Carrier's SDR repository.

From the IPMI perspective, the MMC is set up as a satellite management controller (SMC). The MMC supports sensor devices IPMI commands and uses the static sensor population feature of IPMI. All Sensor Data Records can be queried using Device SDR commands.

Each sensor has a name field in its SDR. The sensor name has a prefix, which is automatically adapted, dependent on the physical position of the module in a carrier or in a µTCA chassis.

The following prefixes are used for all sensors of an AMC module:

Table 8: Sensor Name Prefix

AMC Bay	1	2	3	4	5	6	7	8	-	-	-	-
µTCA slot	1	2	3	4	5	6	7	8	9	10	11	12
Sensor Name Prefix	A1:	A2:	A3:	A4:	B1:	B2:	B3:	B4:	C1:	C2:	C3:	C4:

Module sensors that have been implemented are listed in the sensor list below.

4.1 Sensor List

For OEM (Kontron) specific sensor types and codes in the following table please refer to chapter 4.3.

Table 9: Sensor List

Sensor Number Name	Sensor Type (Code) / Event/Reading Type (Code)	Ass. Mask / Deass. Mask / Reading Mask	Description	Health LED Red on Error
00h / A1:IPMI Info-1	OEM Firmware Info 1 (C0h) / OEM (70h)	0003h / 0000h / 7FFFh	For internal use only	N
01h / A1:IPMI Info-2	OEM Firmware Info 2 (C0h) / OEM (71h)	0003h / 0000h / 7FFFh	For internal use only	N
02h / A1:IPMI Watchdog	Watchdog (23h) / Sensor-specific (6Fh)	010Fh / 0000h / 010Fh	Watchdog 2	Y
03h / A1:FRU Agent	OEM (C5h) / Discrete (0Ah)	0140h / 0000h / 0147h	FRU agent	N
04h / A1:Health Error	Platform Alert (24h) / Digital discrete (03h)	0000h / 0000h / 0003h	Aggregate states (power, temperature etc.). Visualization by the Health LED.	Y
05h / A1:MMC Reboot	Platform Alert (24h) / Digital discrete (03h)	0002h / 0000h / 0003h	MMC reboot active state. Is asserted during boot time.	N
06h / A1:ModuleHotSwap	OEM (F2h) / Sensor-specific (6Fh)	001Fh / 0000h / 001Fh	Hot swap sensor	N
07h / A1:IPMBL State	OEM (C3h) / Sensor-specific (6Fh)	0007h / 0000h / 000Fh	State of IPMB-L bus	N
08h / A1:Storage Err	Mgmt. Subsyst. Health (28h) / Sensor-specific	0002h / 0000h / 0003h	Storage error	N
0Ah / A1: MMC FwUp	Firmware Upgrade Manager (C7h) / Sensor specific (6Fh)	010Fh / 0000h / 010Fh	Status of Firmware Upgrade Manager	N
0Dh / A1:Board Reset	OEM (C4h) / Sensor-specific (6Fh)	04DEh / 0000h / 04DEh	Board reset event	Y
15h / A1:Temp AMC In	Temperature (01h) / Threshold (01h)	7A95h / 7A95h / 3F3Fh	Inlet temperature near AMC edge-connector	Y
16h / A1:Temp Air	Temperature (01h) / Threshold (01h)	7A95h / 7A95h / 3F3Fh	Outlet 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

Table 9: Sensor List

Sensor Number Name	Sensor Type (Code) / Event/Reading Type (Code)	Ass. Mask / Deass. Mask / Reading Mask	Description	Health LED Red on Error
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
1Dh / A1:CPU status	Processor (07h) / Sensor-specific (6Fh)	0463h / 0400h / 04E3h	CPU thermal alarm sensor	N
1Eh / A1:FWH0 Boot Err	Boot Error (1Eh) / Sensor-specific (6Fh)	0008h / 0008h / 0008h	Firmware Hub 0 boot error	Y
1Fh / A1:FWH1 Boot Err	Boot Error (1Eh) / Sensor-specific (6Fh)	0008h / 0008h / 0008h	Firmware Hub 1 boot error	Y
20h / A1:POST Value	OEM Post Value (C6h) / OEM (78h)	0000h / 0000h / 00FFh	POST Value (from host I/O port 80h)	N
21h / A1:Lan AMC0 Link	LAN (27h) / Sensor-specific (6Fh)	0000h / 0000h / 0003h	LAN link status – AMC port 0	N
22h / A1:Lan AMC1 Link	LAN (27h) / Sensor-specific (6Fh)	0000h / 0000h / 0003h	LAN link status – AMC port 1	N
24h / A1:Lan Front0 Lk	LAN (27h) / Sensor-specific (6Fh)	0000h / 0000h / 0003h	LAN link status – Front port 0	N
25h / A1:Lan Front1 Lk	LAN (27h) / Sensor-specific (6Fh)	0000h / 0000h / 0003h	LAN link status – Front port 1	N
26h / A1:Temp VRegHot	Temperature (01h) / Threshold (01h)	0280h / 3280h / 1818h	See description below:	Y
	Virtual temperature sensor based on VCORE_HOT and VCORE_FAN, processor core voltage regulator's over-temperature signals. Sensor readings: 0°C: no over-temperature 110°C: VCORE_HOT active 100°C: VCORE_FAN active			
27h / A1:Temp ProcHot	Temperature (01h) / Threshold (01h)	0280h / 3280h / 1818h	See description below:	Y
	Virtual temperature sensor based on PROCHOT and THERMTRIP, CPU's over-temperature signals. Sensor readings: 0°C: no over-temperature 125°C: THERMTRIP active 105°C: PROCHOT active			

4.2 Sensor Thresholds

The AM5030 CPU module is available with only one operating temperature range. The standard temperature range threshold is defined by Table 10. Table 11 provides voltage sensor thresholds.

Table 10: Thresholds - Standard Temperature Range

Sensor Number / ID string	15h / A1:Temp AMC In	16h / A1:Temp Air
Upper non-recoverable	85 °C	95 °C
Upper critical	80 °C	90 °C
Upper non critical	70 °C	80 °C
Normal max	65 °C	75 °C
Nominal	55 °C	65 °C
Normal min	0 °C	0 °C
Lower non-critical	-5 °C	-5 °C
Lower critical	-7 °C	-7 °C
Lower non-recoverable	-10 °C	-10 °C

Table 11: Voltage Sensor Thresholds

Sensor Number / ID string	17h / A1:Board 3.3vIPM	18h / A1:Board 12.0v	19h / A1:Board 5.0V	1Ah / A1:Board 3.3V
Upper non-recoverable	n.a.	n.a.	n.a.	n.a.
Upper critical	3.50 V	13.4 V	5.36 V	3.50 V
Upper non critical	n.a.	n.a.	n.a.	n.a.
Normal max	3.46 V	13.2 V	5.31 V	3.46 V
Nominal	3.30 V	12.0 V	5.00 V	3.30 V
Normal min	3.13 V	10.8 V	4.70 V	3.13 V
Lower non-critical	n.a.	n.a.	n.a.	n.a.
Lower critical	3.11 V	10.7 V	4.67 V	3.11 V
Lower non-recoverable	n.a.	n.a.	n.a.	n.a.

4.3 OEM Event/Reading Types

OEM (Kontron) specific sensor types and codes are presented in the following table.

Table 12: OEM Event/Reading Types

OEM SENSOR TYPE (CODE)	OEM EVENT/READING TYPE (CODE)	DESCRIPTION	
Firmware Info 1 (C0h)	70h	Internal Diagnostic Data	
Firmware Info 2 (C0h)	71h	Internal Diagnostic Data	
Board Reset (C4h)	6Fh (sensor type specific)	Sensor-specific Offset	Event
		00h	Reserved
		01h	HwPowerReset
		02h	PCIReset
		03h	HwWatchDogReset
		04h	SoftReset
		05h	Reserved
		06h	ColdReset
		07h	IPMICommand
		08h	Reserved
		09h	Reserved
		0Ah	BMCWatchdog
IPMBL State (C3h)	6Fh (sensor type specific)	Sensor discrete State	Meaning
		08h	IPMB-L running
		others	IPMB-L not running
Post Value (C6h)	6Fh (sensor type specific)	Sensor discrete State	Meaning
		Bits [7:0]	Post Value (read from host I/O port 80h)
		Bits [15:8]	Reserved

Table 12: OEM Event/Reading Types

OEM SENSOR TYPE (CODE)	OEM EVENT/READING TYPE (CODE)	DESCRIPTION	
Firmware Upgrade Manager (C7h)	6Fh (sensor type specific)	Sensor-specific Offset	Event
		0h	First Boot after upgrade
		1h	First Boot after rollback (error)
		2h	First Boot after errors (watchdog)
		3h	First Boot after manual rollback
		4h	Reserved
		5h	Reserved
		6h	Reserved
		7h	Reserved
		8h	Firmware Watchdog Bite, reset occurred
Power supply (08h) i.e. for Power Good / Power Good Event	77h (OEM)	Sensor-specific Offset	Event
		0h	12V good (PWR)
		1h	5V good
		2h	3V3 good
		3h	Reserved
		4h	Reserved
		5h	Reserved
		6h	Reserved
		7h	vccCore good
		8h	Reserved
		9h	Reserved
		Ah	Reserved
		Bh	3V3IPMI good (MP)
		Ch	Reserved
Hot swap sensor (F2h)	6Fh (sensor type specific)	Sensor-specific Offset	Event
		00h	Handle close
		01h	Handle open
		02h	Quiesced
		03h	Backend Power Failure
		04h	Backend Power Shutdown

5. Firmware Code

5.1 Structure and Functionality

MMC firmware code is organized into boot code and operational code (IPMI Firmware). Both are stored in the internal flash of the micro-controller.

An additional external serial flash device is used for holding redundant copies of the operational code. This additional flash device is organized to have two banks. One of them will always hold a copy of the active operational code. The other firmware bank holds either a newly downloaded firmware or the 'previous good' operational code for rollback.

Upon a MMC start or reset, the controller first executes the boot code. The boot code will check the status of the firmware banks and calculate a checksum of the operational code. Upon successful verification of the operational code checksum, the firmware will execute the operational code. The operational code is upgradable in the field.

5.2 Firmware Upgrade

The standard way to upgrade the MMC's operational code is to use the open tool “ipmitool” (see Table 2, Related Publications). This tool allows download and activation of new operational code and also rollback to the 'last known good' operational code. Additionally, the status and the firmware version of the redundant firmware copies can be checked.

For local or remote firmware upgrade the following IPMI interfaces are available:

- KCS interface (locally, requires active payload, but fast)
- IPMB (remote, independent of the payload state)
- LAN (remote, via IOL, requires also active payload)

During the download process the currently running operational code operates as usual until the activation command is issued. During the activation process the MMC is off line for about 45 seconds while the boot code is re-organizing the firmware storage. Afterwards, it starts the new operational code. If this doesn't succeed, after a time out the boot code performs an automatic rollback to the 'last known good' operational code.

5.2.1 Firmware File Formats

Firmware images for upgrade are provided in two formats:

- Firmware in binary format, e.g. **FW_IPMI_**<*BOARD*>**_**<*REL*>**_FWUM.bin**, for usage with ‘`ipmitool fwum ..`’ commands.
- Firmware images in the PICMG defined HPM.1 file format, e.g. **FW_IPMI_**<*BOARD*>**_**<*REL*>**_FWUM.hpm**, for usage with ‘`ipmitool hpm ..`’ commands.

  Where:

  <*BOARD*> identifies to board family of the MMC’s firmware

  <*REL*> identifies to release (version) of MMC’s firmware.

5.2.2 Firmware Upgrade - “ipmitool hpm”

Firmware upgrade using a HPM.1 file requires at least “ipmitool” version 1.8.10.

The firmware upgrade procedure starts with downloading the HPM.1 file using:

‘**ipmitool hpm upgrade** *<HPM.1_FWFile>*.***hpm*** **all**’

for example.

The next step is the activation of the newly downloaded MMC firmware, this is done using:

‘**ipmitool hpm activate**’

To get detailed information about the firmware versions which are currently active or of the redundant copies, the following commands can be used:

- ‘**ipmitool hpm compprop 1 1**’

  returns detailed version information of active MMC firmware.

- ‘**ipmitool hpm compprop 1 3**’

  returns version of rollback copy (only valid if a newly downloaded firmware is already activated).

- ‘**ipmitool hpm compprop 1 4**’

  returns version of the newly downloaded MMC firmware (only valid after download and before activation).

Detailed information about the MMC’s HPM.1 upgrade capabilities can be determined using:

‘**ipmitool hpm targetcap**’

5.2.3 Firmware Upgrade - “ipmitool fwum”

The firmware upgrade procedure starts with the downloading of the binary firmware file using:

‘**ipmitool fwum download** *<Binary_FWFile>*.***bin***’

for example.

The next step is the activation of the newly downloaded MMC firmware. This is done using

‘**ipmitool fwum upgrade**’

To get detailed information about the firmware versions which are currently active or of the redundant copies the following command can be used:

‘**ipmitool fwum status**’

returns detailed version information of redundant firmware copies.

Some information about the MMC’s upgrade capabilities can be determined using the command:

‘**ipmitool fwum info**’

5.3 MMC Firmware Configuration

For initial setup and to get some basic information of the AM5030 Module Management Controller the AM5030 EFI shell is used. Refer to chapter ‘uEFI Shell’ of ‘AM5030 uEFI BIOS – User Guide’ for more information.

Besides the built-in uEFI Shell commands, the Kontron uEFI implementation provides a number of additional commands, related to the specific hardware features of the system.

The Kontron uEFI Shell command for configuration of the System Management is the ‘**kipmi**’ command. The ‘**kipmi**’ command provides a set of parameters to support various IPMI Management controllers. Note that not all parameters have an impact on the AM5030 MMC.

On the AM5030 the ‘**kipmi**’ command may be used with following parameters:

- `kipmi` without any parameter, displays a list of available parameters
- `kipmi irq` provides information about currently selected IRQ used for KCS System Interface. An additional parameter 10, 11 or 0, sets KCS IRQ configuration to *IRQ10*, *IRQ11* or to *no IRQ* at all
- `kipmi fru` for reading the FRU inventory data
- `kipmi net` for basic IOL/SOL configuration
- `kipmi raw` for execution of raw IPMI commands (with additional parameters)

Using the ‘**kipmi**’ command with parameters other than those listed here, doesn’t have any impact on the AM5030 Module Management Controller.

5.4 KCS Interface Interrupt

The default factory setting of a AM5030 for its KCS interface is ‘*no IRQ’.* When changing the configuration, the uEFI creates/updates an entry in the SMBIOS table. This record contains the following information (among others):

- type of the supported interface (KCS style)
- selected interrupt (10, 11 or none)

This information is needed by the Operating System’s KCS interface kernel driver when it is loaded. Changing the KCS interrupt number from uEFI shell requires a restart of the uEFI BIOS for a correct set up of the SMBIOS table. So issue a ‘**reset**’ command to leave the uEFI shell after changing the KCS Interrupt selection.

5.5 Firmware / Module Identification

There are two ways, by means of IPMI, to identify the AM5030 Module Management Controller Firmware:

- Issuing a IPMI Command *Get Device ID*
- Read the Device Locator Record (SDR Type 12h)

A full description of the IPMI command **‘Get Device ID’** and the Device Locator Record (SDR Type 12h) can be found in the IPMI specification, refer to Table 2, Related Publications.

5.5.1 IPMI Command: ‘Get Device ID’

The response on the IPMI command **‘Get Device ID’** offers the following information (among others):

- Manufacturer ID = 3A98h / 15000d (Kontron IANA ID)
- Device ID = 20h (NXP LPC2368)
- Product ID = identifies the Firmware (its board family firmware))
- Firmware Revision in bytes 4:5 - depends on the core version of the running firmware.
- The SDR revision in byte 13 (OEM part of the response) is the firmware revision.

For a description of the OEM extensions refer to chapter 3.1, “Get Device ID Command with OEM Extensions”

5.5.2 Device Locator Record

The Device Locator Record (SDR Type 12h) contains a Device ID String which identifies the MMC as AM5030 MMC. Additionally some run-time information like AMC slot and the slot dependent IPMB address is available in this record.

For example, when using the Linux “ipmitool” on a AM5030 placed in the first AMC slot of a µTCA system, by calling:

‘ipmitool sdr list mcloc’

the following information is displayed:

A1:AM5030 | … @72h | ok

6. FRU Information

The MMC provides 4 kB of non-volatile storage space for FRU information. Some of the data stored there, like the Module Current Requirements record or E-Keying information (refer to AMC.0 specification for details), are mandatory for module functionality in the ATCA/AMC environment.

Please note that missing FRU information possibly will prevent the AMC module from being accepted by the carrier controller during the Hot Swap process, and the module will possibly not receive payload power.

Full low-level access to read or write a module's FRU Information is provided by regular IPMI FRU Device commands. Please be careful when writing FRU information directly using standard IPMI commands. Damaging the FRU Information may lead to a non-working payload.

7. E-Keying

E-Keying has been defined in the AMC.0 R2.0 Specification to prevent module damage or improper operation, and to verify bay connection compatibility. Therefore the FRU Data of an AMC module contains PICMG defined records which describe the module's AMC interoperability:

- Module Current Requirements Record
- Clock Configuration Record, for the PCI-Express reference clock
- AMC Point-to-point record, describing module's AMC port capabilities

The IPMI commands **'Set AMC Port State'** and **'Get AMC Port State'** defined by the AMC.0 specification are used by the carrier or MCH for either granting or rejecting the E-keys (i.e. enabling or disabling of AMC Ports during E-Keying).

Which AMC port connections are activated will be decided during E-keying. The information which AMC port is enabled or not, can be directly read from board's E-Keying Configuration registers (IAKEY0 and IAKEY1) at addresses 298h / 299h.

The 'DIP Switch SW2' can be used to forcibly disable some AMC Ports if required. Please refer to the "AM5030 User Guide" for details.

7.1 PCI Express Lane Width – x4 or x1

The AM5030 supports either one PCI-E x4 connection (default) or one PCI-E x1 connection alternatively. Both PCI-E Gen1 and PCI-E Gen2 frequencies are supported.

7.2 PCI Express Reference Clock

Both sides (Root Complex and Endpoint) of a PCI-Express connection should be driven by a common reference clock. The PCI-E reference clock may be generated locally by the module or acquired from the AMC connector.

The AM5030 (PCI-Express Root Complex) may act either as clock receiver or as clock source. This is described by the Clock Configuration Record (for the PCI-Express reference clock) and defined by the "AMC.1 R2.0, PCI Express on AMC" specification.

7.2.1 Clock Receiver

The PCI-E reference clock provided by the carrier may be slightly modulated (SSC - Spread Spectrum Clock). The FRU E-Keying data for AM5030 contains several AMC Link Descriptors for the PCI-Express channel, describing either SSC or non-SSC and the PCI-E Gen2 or Gen1 clock capabilities.

The carrier's IPMC or the MCH selects the 'matching' Link descriptor (SSC / non-SSC and Gen2/Gen1) during E-keying using the 'Set AMC Port State' command.

7.2.2 Clock Source

When the AM5030 acts as clock source for the PCI-Express reference clock, the clock signal must be routed also to the PCI-Express Endpoint. The backplane, the carrier's IPMC or the MCH must be capable of doing this (Clock E-Keying according to AMC.1 R2.0).

The information which one (AMC clock or local clock) is used as PCI-Express reference clock, can be directly read from board's E-Keying Clock Configuration register (ICKEY0) at address 297h.

The 'DIP Switch SW2' can be used to forcibly configure the PCI-Express reference clock. Please refer to the "AM5030 User Guide" for details.

8. uEFI BIOS Failover Control - Automatic Flash Selection

After each payload CPU reset the Management Controller selects the uEFI BIOS Flash by applying the related non-volatile parameter. Then it waits for a message from the uEFI BIOS. This message contains a checksum report, i.e. it reports whether the Boot Flash's checksum is right or wrong.

If either the checksum is wrong or the message is not received within a given time, then the currently used uEFI BIOS is assumed to contain an invalid or a corrupted image. In this case the Management Controller toggles the related non-volatile parameter and generates a "Boot Error - Invalid boot sector" event. The sensor event is generated either by sensor "FWH0 Boot Err" or "FWH1 Boot Err", depending on which uEFI BIOS bank failed.

After selecting the alternate uEFI BIOS bank, the payload CPU is reset and the Management Controller waits for the checksum report message from EFI again.

The number of retries depends on the error condition (no message from EFI at all or checksum error).

The number within the names of the two related sensors "FWH0/1 Boot Err" corresponds to the value of the non-volatile parameter, not to the absolute number of the uEFI BIOS firmware bank (which is not known by the MMC).

9. Hot Swap

As a hot-swappable field replaceable unit (FRU), the AM5030 also follows the same stringent carrier grade RASM feature set, namely - Reliability, Availability, Serviceability, Maintainability. When offered in combination with AdvancedTCA platforms, TEM (Telecom Equipment Manufacturers) clients literally conserve valuable system AdvancedTCA system slots. The AM5030 supports Full Hot Swap capability as per PICMG 3.0. It can be removed from or installed in the system while it is on (without powering-down the system). Please refer to the PICMG 3.0 specification for additional details.

During Hot Swap of a working module the payload side has to be shut down automatically on command of the MMC and the end of shut down has to be signalled back to the MMC. Because the AM5030 supports ACPI, an OS on the payload side which supports this too makes shut down very easy. If the OS doesn't support ACPI there is a special method to be used.

9.1 Method 1: The Payload OS Supports ACPI

Requirements:

- ACPI support must be enabled in the BIOS menu.
- The ACPI daemon must be active.
- An ACPI power button event must result in a sleep state.

Part of the Hot Swap Operation sequence to be processed by MMC and OS:

- On command of the carrier controller, the MMC simulates the pressing and release of the “power button” to force an ACPI event.
- The ACPI daemon detects this ACPI event and initiates the shut down of the payload software system.
- At the end of shut down the payload hardware system reports the sleep state to the MMC by setting the appropriate signal line.
- The MMC detects the sleep state and reports this to the carrier controller (“quiesced”) so that the Hot Swap processing can be continued and finished.

By default the MMC waits endlessly for the sleep state. Please note: Some Shelf Managers or MCHs use a time out to simply switch off of a module which needs too much time to reach sleep state. As this might be an undesirable situation, refer to the appropriate manual for further assistance. In any event, if an endless wait is to be avoided, it is possible to set a timeout time for the module's MMC after which the system will be switched off unconditionally. For the setting of the timeout refer to 3.4, OEM Module Quiescence Feedback.

9.2 Method 2: The Payload OS Does Not Support ACPI

Requirements:

- At system start on the payload side the Kontron shutdown daemon 'grnsd' must be started. It is included in the Linux board support packages for the AM5030. This daemon communicates cyclically with the MMC for the exchange of states, commands and acknowledgments. For this, it uses the “OEM Module Quiescence Feedback” command. Refer to chapter 3.4. In principle it plays the same role as the ACPI daemon of Method 1 above.

Part of the Hot Swap Operation sequence to be processed by MMC and OS:

- On command of the carrier controller the MMC sets a “shut down request” flag.
- The 'grnsd' daemon recognizes this request in the response to its cyclical “OEM Module Quiescence Feedback” command and initiates the shut down of the payload software system.
- At the end of the shut down process, the 'grnsd' daemon informs the MMC by setting the appropriate flag when calling the “OEM Module Quiescence Feedback” command.
- The MMC reports this to the carrier controller so that the Hot Swap processing can be continued and finished.

By default the MMC waits endlessly for this information. If an endless wait is to be avoided, it is possible to set a timeout time after which the system will be switched off unconditionally. For the setting of the timeout refer to 3.4, OEM Module Quiescence Feedback.

10. LAN Functions

10.1 Overview

The two Ethernet channels on the AMC Fabric Interface and also the two Ethernet channels on the front panel can - in parallel to their 'normal' use - be used for the following special purposes:

- IPMI over LAN (IOL)
- Serial over LAN (SOL)

Common for both kinds of communication is the use of the RMCP/RMCP+ for the packing of the data to be transferred. On Ethernet the port 623 is used for transfers with this protocol.

While IOL serves to transport IPMI commands and their responses, the SOL serves to transport any serial data. In each case the MMC serves as a protocol encoder and decoder. Please note that IOL is able to use both RMCP and RMCP+ protocols. SOL works only with the RMCP+ protocol. In addition, both IOL and SOL require that the payload Ethernet device be powered. Therefore the module (payload) must be fully powered.

The following table shows the assignment of the four IOL / SOL channels to the Gigabit Ethernet interfaces.

Table 13: IOL/SOL Channel Assignment

IOL/SOL Channel	Gigabit Ethernet Interface	Connector / Location
1	GbE-A	AMC Port 0 / AMC Card-edge connector
2	GbE-B	AMC Port 1 / AMC Card-edge connector
3	GbE-C	Lower Front panel connector J2A
4	GbE-D	Upper Front panel connector J2B

10.2 Setting up the Ethernet Channel

There are two methods to prepare the MMC's SOL and IOL LAN parameters for the four possible Ethernet channels:

- In the uEFI BIOS from the EFI-Shell using **‘`kipmi net`’**
- By use of the open tool “ipmitool” or raw IPMI commands (via KCS or IPMB-L)

Both of the methods are compatible with one another, meaning that they both set or show the exact same parameters as the other.

The setup is separate for all four channels. When the MAC addresses are set the ones which are programmed into the hardware must be re-used. This is a restriction. The IP addresses of a channel being used by 'normal' payload traffic and IOL/SOL traffic may differ but need not differ as long as the RMCP port 623 is not used in parallel by payload and IOL/SOL.

10.3 Basic Setup from EFI-Shell

With the **'kipmi net'** command from EFI-Shell some basic settings like IP address, sub-net mask and gateway address can be setup for all of the four Ethernet channels.

10.4 Setup by "ipmitool" or Raw IPMI Commands

The open tool "ipmitool" offers commands for the setup of the four Ethernet channels. All possible options are shown by issuing:

```
ipmitool lan set
```

If "ipmitool" is not usable, the LAN parameters can be set by using the raw IPMI commands which are defined for this.

To show the current LAN parameters for a channel, "ipmitool" offers the command:

```
ipmitool lan print <channel = 1, 2, 3, 4>
```

10.5 Setup of User and Password

The open tool "ipmitool" offers commands for the listing and manipulation of user accounts for channels 1 through 4. An overview can be obtained by issuing:

```
ipmitool user
```

The predefined users for a channel can be listed by the command:

```
ipmitool user list <channel = 1, 2, 3, 4>
```

The AM5030 has for every channel these predefinitions in non-volatile memory:

```
ID  Name             Callin  Link Auth  IPMI Msg   Channel Priv Limit
1                    false   true       true       USER
2   admin            false   true       true       ADMINISTRATOR
```

Please note that admin's password is preset with 'admin'.

Changed users and passwords stay valid after payload power off.

The user (ID number, from above) must be activated by:

```
ipmitool user enable <user number>
```

10.6 IPMI Over LAN (IOL)

IPMI over LAN is used to allow the IPMI controller to communicate with the MMC via LAN using the RMCP or RMCP+ protocol. The data which is transferred are IPMI commands and the responses to them.

To enable the LAN support after parameter setup this command has to be issued:

```
ipmitool lan set <channel = 1, 2, 3, 4> access on
```

Please note that the following commands must use the IP address which belongs to the enabled channel.

The open tool “ipmitool” can serve as a control program and user interface for this. “ipmitool” allows the issuing of raw IPMI commands such as:

```
ipmitool -I lan -H 192.168.3.189 -U admin -P admin -A PASSWORD raw 6 1
```

or to call complex functions like 'mc info':

```
ipmitool -I lan -H 192.168.3.189 -U admin -P admin -A PASSWORD mc info
```

This uses many raw IPMI commands to get all needed information.

10.7 Serial Over LAN (SOL)

Serial over LAN connects the COM1 or /dev/ttyS0 respectively of the AM5030's payload side to an Ethernet channel. The MMC resides between this serial interface and one of the Ethernet channels. It serves as an encoder and a decoder for the used RMCP+ protocol and controls the data stream. Outside the AM5030 for example, the open tool “ipmitool” can be used to drive the SOL session i.e. it offers a console function to communicate via Ethernet with the AM5030's serial interface.

The serial interface can be used as a connection, for example:

- To a user program on the AM5030 payload
- To the BIOS console redirection function. Refer to the BIOS setup menu “Main>Serial Port Console Redirection”. There the serial parameters for this purpose can be set. Please note that after BIOS start, the OS gets active in most cases (except e.g. DOS) and the console redirection stops working because the OS doesn't use BIOS functions to drive the console.
- To a Linux login console. This can be activated after payload start, for example. by the command:

  ```
  getty -h 115200 /dev/ttyS0
  ```

SOL supports and requires serial hardware handshake. This should be activated for the serial port. Otherwise transmitted data might get lost. In any case the same serial parameters for the used payload side serial interface and the MMC's serial interface must be used. The parameters for the MMC's serial interface can be set by the ‘ipmitool sol set …’ command. Calling ‘ipmitool sol set’ shows all options that can be set.

To start an SOL session with “ipmitool”, additional parameters are required:

```
ipmitool -I lanplus -H 192.168.3.189 -U admin -P admin \
         -L USER -C 0 sol activate
```

Description of the additional parameters:

- `-I lanplus` SOL session uses RMCP+ protocol only
- `-L USER` Default privilege level for SOL session is USER.
- `-C 0` Firmware supports ‘straight password authentication’ only. For SOL sessions, Cipher Suite ID **0** must be used.

Other commands which are possible are shown when issuing 'ipmitool sol help'.

11. OS Support / Tools

11.1 Linux Tools

11.1.1 OpenIPMI - KCS driver

Normally all drivers and kernel modules needed for communication between the payload sided software and the Management Controller firmware via the KCS interface come with the distribution. Newest sources can be downloaded from: ‘http://openipmi.sourceforge.net’. There may be downloaded the OpenIPMI project as well. The OpenIPMI library package includes some applications and the needed libraries.

11.1.2 IPMI Tool

Another very useful all-in-one tool is “ipmitool” (http://ipmitool.sourceforge.net). It provides a user friendly interface to many IPMI features and extensions, for example, to get sensor readings, change sensor thresholds or to access other Management Controllers via IPMB. Before “ipmitool” can be used the OpenIPMI driver, mentioned above, must be loaded too.

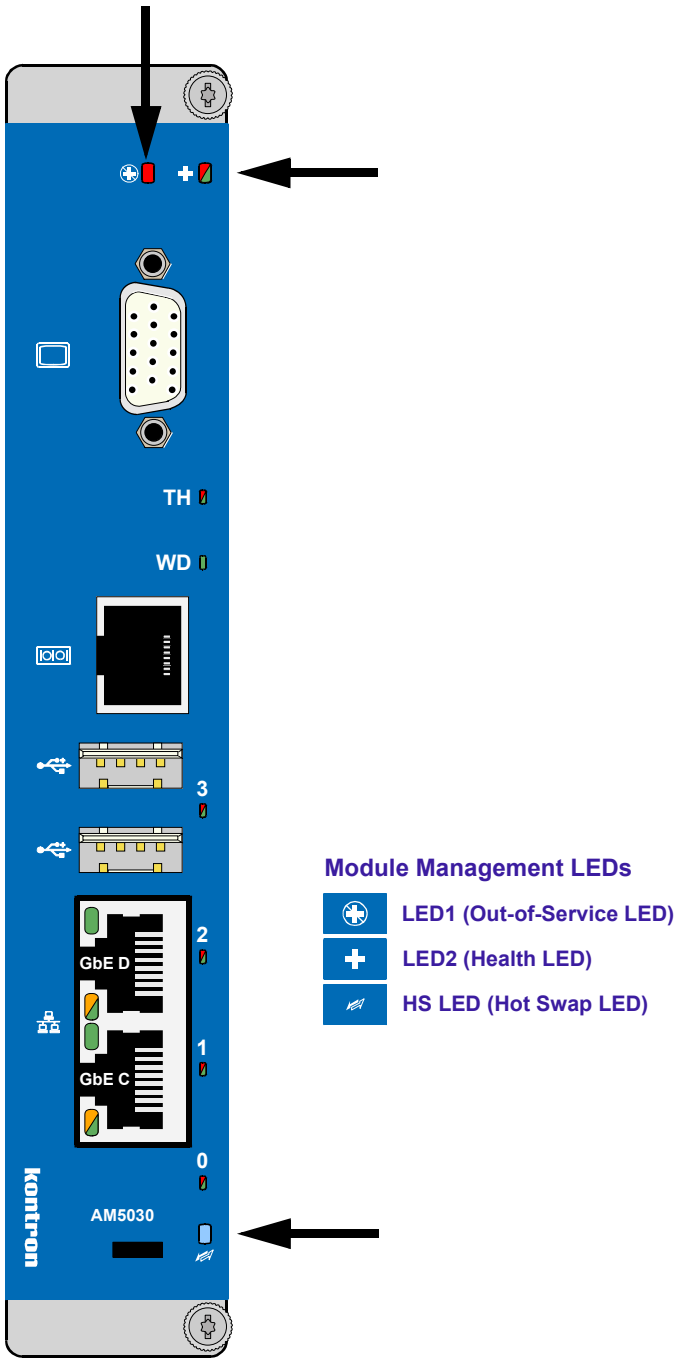
11.2 OS Support - Board Support Packages

To see which Operating Systems are supported refer to the board's data sheet, please visit ‘http://www.kontron.com’ to download the AM5030 data sheet. Also have a look in the download section for latest versions of Board Support Packages or Firmware Updates.

For further information concerning IPMI, refer to the BSP documentation for the respective OS.

12. IPMI Module Management LEDs

There are three IPMI Module Management LEDs on the front panel of the AM5030. The following figure illustrates the AM5030 module and the location of the LEDs



The following table describes the functioning of the Module Management LEDs.

Table 14: Module Management LEDs Function

LED	COLOR	STATE	NORMAL MODE	OVERRIDE MODE selectable by user or carrier, depending on PICMG LED command
LED1 (Out-of-Service LED)	red	off	Default	By user: • Only lamp test
		on	MMC out of service or in reset state	
		blinking	MMC Firmware upgrade	
LED2 (Health LED)	green/ red+amber	green: blinking	MMC running showing its heartbeat	By user: • Only lamp test
		green: blinking and pulsing	MMC heart beat and KCS traffic	
		red: on + amber: blinking	Health error detected + MMC heart beat	
		red: on + amber: blinking and pulsing	Health error detected + MMC heart beat and KCS traffic	
HS LED (Hot Swap LED)	blue	on	a) Module ready for hot swap extraction, or b) Module has just been inserted in a powered system	By carrier: • On • Off • Slow/Fast Blinking
		blinking	Module hot swap in progress; module not ready for extraction	By user: • Only lamp test
		off	Module is in normal operation	