kontron

» User Guide «

AM4022
IPMI Firmware

Doc. ID: 1052-3624, Rev. 1.0
August 8, 2012

If it’s embedded, it’s Kontron.

Revision History

Publication Title:	AM4022 IPMI Firmware User Guide	
Doc. ID:	1052-3624	
Rev.	Brief Description of Changes	Date of Issue
1.0	Initial issue	8-Aug-2012

Imprint

Kontron Europe GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Europe GmbH
Sudetenstraße 7
D - 87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

+49 (0) 800-SALESKONTRON
sales@kontron.com

For further information about other Kontron products, please visit our Internet web site: www.kontron.com.

Disclaimer

Copyright © 2012 Kontron AG. All rights reserved. All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

Table of Contents

Revision History 2
Imprint 2
Disclaimer 2
Table of Contents 3
List of Tables 7

1. Introduction 9

1.1 Terminology and Acronym Definitions 9
1.2 Related Publications 10
1.3 IPMI in AdvancedMC / AdvancedTCA Environment 10
1.4 Module Management Controller Hardware 11

2. MMC Firmware 11

2.1 Key Features 11
2.2 Supported IPMI and ATCA Commands 12
2.2.1 Standard IPMI Commands 12
2.2.2 AdvancedTCA and AMC Commands 20

3. OEM Commands and Command Extensions 22

3.1 Get Device ID Command with OEM Extensions 22
3.2 Set Control State (Firmware Hub, Boot Order) 23
3.3 Get Control State (Firmware Hub, Boot Order) 24
3.4 OEM Module Quiescence Feedback 25
3.4.1 Usage if a Shutdown Daemon is Announced as Present 26
3.4.2 Usage if no Shutdown Daemon is Announced as Present 26

4. Sensors Implemented on the AM4022 27

4.1 Sensor List 28
4.2 Sensor Thresholds 30
4.3 OEM Event/Reading Types 32

5. Firmware Code 35

5.1 Structure and Functionality 35

5.2 MMC Firmware Configuration 35

5.3 KCS Interface Interrupt 35

5.4 Firmware / Module Identification 36

5.5 Firmware Upgrade 37

5.5.1 Firmware File Formats 37

5.5.2 Firmware Upgrade - “ipmitool hpm” 38

5.5.3 Firmware Upgrade - “ipmitool fwum” 38

6. FRU Information 39

6.1 FRU Version Identification 39

6.2 FRU Data Update 39

7. E-Keying 40

8. uEFI BIOS Failover Control - Automatic SPI Boot Flash Selection 40

9. OS Boot Order Selection by OEM IPMI 41

10. Hot Swap 41

10.1 Method 1: The Payload OS Supports ACPI 41

10.2 Method 2: The Payload OS Does Not Support ACPI 42

11. LAN Functions 42

11.1 Overview 42

11.2 Setting Up the Ethernet Channel 42

11.3 Basic Setup from uEFI Shell 43

11.4 Setup by “ipmitool” or IPMI Commands 43

11.5 Setup of User Accounts and Password 43
11.6 IPMI Over LAN (IOL) 44
11.7 Serial Over LAN (SOL) 44

12. OS Support / Tools 45
12.1 Linux Tools 45
12.2 OS Support - Board Support Packages 46

13. IPMI Module Management LEDs 46

This page has been intentionally left blank.

List of Tables

1 Terminology and Acronym Definitions 9
2 Related Publications 10
3 Standard IPMI Commands 12
4 AdvancedTCA and AMC Commands 20
5 Get Device ID Command with OEM Extensions 22
6 Set Control State 23
7 Get Control State 24
8 OEM Module Quiescence Feedback 25
9 Sensor Name Prefix 27
10 Sensor List 28
11 Thresholds - Standard Temperature Range 30
12 Thresholds - Extended Temperature Range 30
13 Voltage Sensor Thresholds 31
14 OEM Event/Reading Types 32
15 Module Management LED Functions 47

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 / 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 the 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
AM4022	AM4022 User Guide, ID 1052-0183, Rev. 1.0 AM4022 uEFI BIOS User Guide, ID 1053-1333, Rev. 1.0
AM4022 BSP	AM4022 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 AM4022 follows the stringent carrier grade RASM feature set, namely - Reliability, Availability, Serviceability, Maintainability.

Built in accordance with the AMC.0 specification, the AM4022 is also compliant with the AMC.1, AMC.2, and AMC.3 specifications and is easily managed via its IPMI v2.0-compliant management features.

As with every Advanced Mezzanine Card (AMC), the AM4022 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 AM4022 processor AMC module, the MMC is implemented using an NXP® ARM7 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 4 MB serial SPI flash is used for redundant firmware image storage.

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

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

The MMC provides access to various sensors which permit the monitoring of:

- System power voltages: +12V (PWR), +5V, +3.3V, +3.3V (MP)
- Temperatures: CPU and PCH die as well as airflow near AMC edge-connector
- Power Good, LAN links, IPMB link, board reset, POST code, boot error, CPU States (processor hot, THERMTRIP, …), IPMB-L state, Health error, IPMI watchdog, etc.

2. MMC Firmware

2.1 Key Features

The following are key features of the AM4022 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 the module’s 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
- Firmware image management may done by the open tool “ipmitool” (functions “hpm” or “fwum”)
- Downloading new firmware image does not break currently running firmware activities
- Manual firmware image roll-back in case of upgrade failure
- Interoperable with other AMC, ATCA, or IPMI solutions

- Fail-over control to a recovery BIOS in the event a non-working uEFI BIOS is detected
- 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 indicates whether the module is healthy (normal operation) and all sensors are within the specified range (green) or at least one sensor is out of range (amber).

2.2 Supported IPMI and ATCA Commands

2.2.1 Standard IPMI Commands

The following table shows an excerpt from the command list specified in the IPMI specification 2.0. The shaded table cells indicate commands supported by the AM4022 MMC.

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 (Continued)

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 (Continued)

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 (Continued)

COMMAND	IPMI 2.0 SPEC. SECTION	NETFN	CMD	KONTRON SUPPORT ON MMC
EVENT COMMANDS				M
Set Event Receiver	29.1	S/E	00h	M / Yes
Get Event Receiver	29.2	S/E	01h	M / Yes
Platform Event (a.k.a. “Event Message”)	29.3	S/E	02h	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 (Continued)

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 (Continued)

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 (Continued)

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 (Continued)

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

The following table shows an excerpt from the command lists specified in the PICMG 3.0 R 2.0 AdvancedTCA Base Specification and the PICMG AMC.0 Advanced Mezzanine Card Specification, R 1.0. The shaded table cells indicate commands supported by the AM4022 MMC.

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

Table 4: AdvancedTCA and AMC Commands (Continued)

COMMAND	PICMG 3.0 SPEC. TABLE	NETFN	CMD	KONTRON SUPPORT ON MMC
Set Fan Level	3-65	PICMG	15h	N/A
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	1Ah	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

<table>
<tr><th colspan="2">COMMAND</th><th>LUN</th><th>NetFn</th><th>CMD</th></tr>
<tr><td colspan="2">Get Device ID command with OEM extensions</td><td>00h</td><td>App = 06h</td><td>01h</td></tr>
<tr><th colspan="5">REQUEST DATA</th></tr>
<tr><th>Byte</th><th colspan="4">Data Field</th></tr>
<tr><td>-</td><td colspan="4">-</td></tr>
<tr><th colspan="5">RESPONSE DATA</th></tr>
<tr><th>Byte</th><th colspan="4">Data Field</th></tr>
<tr><td>1</td><td colspan="4">Completion Code</td></tr>
<tr><td>2 - 12</td><td colspan="4">Regular Get Device ID command response fields</td></tr>
<tr><td>13</td><td colspan="4">Release number of the MMC firmware:
10h for R10,
11h for R11,
...</td></tr>
<tr><td>14</td><td colspan="4">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</td></tr>
<tr><td>15 - 16</td><td colspan="4">Reserved</td></tr>
</table>

3.2 Set Control State (Firmware Hub, Boot Order)

Table 6: Set Control State

<table>
<tr><th colspan="2">COMMAND</th><th>LUN</th><th>NetFn</th><th>CMD</th></tr>
<tr><td colspan="2">Set Control State (Firmware Hub, Boot Order)</td><td>00h</td><td>OEM = 3Eh</td><td>20h</td></tr>
<tr><th colspan="5">REQUEST DATA</th></tr>
<tr><th>Byte</th><th colspan="4">Data Field</th></tr>
<tr><td>1</td><td colspan="4">Control ID:
00h = Firmware hub (SPI boot flash) Selection
9Dh = Boot Order Configuration</td></tr>
<tr><td>2</td><td colspan="4">Control State for SPI boot flash selection:
(These settings are stored in EEPROM and applied (to logic) each time the IPMI controller detects power-on)
00h = Standard SPI boot flash is selected (default)
01h = Recovery SPI boot flash is selected
Please note that this selection may be forcibly overridden either by the DIP Switch SW3, switch 2 (refer to the AM4022 User Guide, Table 4-3).
To be able to change the SPI boot flash selection via the Set Control State command, the recovery SPI boot flash must not be previously selected.
In case of a failed boot process from the standard SPI boot flash, the IPMI controller will select the recovery SPI boot flash and boot the board again.
In case of a boot failure from the recovery SPI boot flash, the board locks up.
Refer to Chapter 8, uEFI BIOS Failover Control - Automatic SPI Boot Flash Selection, for further information.
Control State for 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</td></tr>
<tr><th colspan="5">RESPONSE DATA</th></tr>
<tr><th>Byte</th><th colspan="4">Data Field</th></tr>
<tr><td>1</td><td colspan="4">Completion Code</td></tr>
</table>

3.3 Get Control State (Firmware Hub, Boot Order)

Table 7: Get Control State

<table>
<tr><th colspan="2">COMMAND</th><th>LUN</th><th>NetFn</th><th>CMD</th></tr>
<tr><td colspan="2">Get Control State (Firmware Hub, Boot Order)</td><td>00h</td><td>OEM = 3Eh</td><td>21h</td></tr>
<tr><th colspan="5">REQUEST DATA</th></tr>
<tr><th>Byte</th><th colspan="4">Data Field</th></tr>
<tr><td>1</td><td colspan="4">Control ID:
00h = Firmware hub (SPI boot flash) selection
9Dh = Boot Order Configuration</td></tr>
<tr><th colspan="5">RESPONSE DATA</th></tr>
<tr><th>Byte</th><th colspan="4">Data Field</th></tr>
<tr><td>1</td><td colspan="4">Completion Code</td></tr>
<tr><td>4</td><td colspan="4">Control State (refer to Chapter 3.2, Set Control State)
00h .. 01h for control ID = Firmware hub (SPI boot flash) selection
00h .. FFh for control ID = Boot Order Configuration</td></tr>
</table>

3.4 OEM Module Quiescence Feedback

This command is used to control a graceful shutdown of the AM4022 and is a prerequisite for the hot swap feature. For further information on hot swap, refer to Chapter 10, Hot Swap.

If the software environment does not support ACPI, a self-written shutdown daemon, can be used to shut down the system in an orderly manner. For this purpose, Kontron’s latest BSP includes a Graceful Reboot and Shutdown Daemon, “grnsd”.

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).

Table 8: OEM Module Quiescence Feedback

<table>
<tr><th colspan="2">COMMAND</th><th>LUN</th><th>NetFn</th><th>CMD</th></tr>
<tr><td colspan="2">OEM Module Quiescence Feedback</td><td>00h</td><td>OEM = 3Eh</td><td>40h</td></tr>
<tr><th colspan="5">REQUEST DATA</th></tr>
<tr><th>Byte</th><th colspan="4">Data Field</th></tr>
<tr><td>1</td><td colspan="4">Control bits:
[7] - 1b = set quiesce wait timeout
[6] - 1b = quiescence acknowledge (OS ready)
[5] - 1b = OS daemon present
[4:0] Reserved</td></tr>
<tr><td>2</td><td colspan="4">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, the Module Hot Swap event message will be sent immediately. If the time is set to 0 (default after reset,) the Module Hot Swap event message will only be sent after recognition of sleep state (signal).</td></tr>
<tr><th colspan="5">RESPONSE DATA</th></tr>
<tr><th>Byte</th><th colspan="4">Data Field</th></tr>
<tr><td>1</td><td colspan="4">Completion code</td></tr>
<tr><td>2</td><td colspan="4">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</td></tr>
<tr><td>4</td><td colspan="4">Quiesce wait timeout (valid only if OS daemon present = 1)</td></tr>
</table>

3.4.1 Usage if a Shutdown Daemon is Announced as Present

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 shutdown. 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.

3.4.2 Usage if no Shutdown 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).

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 AM4022

The MMC includes several 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 9: 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

The following table indicates all sensors available on the AM4022. For further information on Kontron's OEM specific sensor types and sensor event type codes presented in the following table, please refer to Chapter 4.3, OEM Event/Reading Types.

Table 10: Sensor List

Sensor Number / Name	Sensor Type (Code) / Event/Reading Type (Code)	Ass. Mask / Deass. Mask / Reading Mask	Description	Health LED Shows 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:Stor Err	Mgmt. Subsyst. Health (28h) / Sensor-specific (6Fh)	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
0Ch / A1: Config Error	OEM (CEh) / Sensor-specific (6Fh)	017Ch / 0000h / 077Dh	Configuration Error	Y
0Dh / A1:Board Reset	OEM (C4h) / Sensor-specific (6Fh)	04DEh / 0000h / 04DEh	Board reset event	Y
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
11h / A1:Temp Air	Temperature (01h) / Threshold (01h)	7A95h / 7A95h / 3F3Fh	Air temperature near AMC edge-connector	Y
12h / A1:Board 3.3vIPM	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	AMC Management Power (MP) 3.3V	Y

Table 10: Sensor List

Sensor Number / Name	Sensor Type (Code) / Event/Reading Type (Code)	Ass. Mask / Deass. Mask / Reading Mask	Description	Health LED Shows Error
13h / A1:Board 12.0v	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	AMC Payload Power (PWR) 12V	Y
14h / A1:Board 5.0V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 5V supply	Y
15h / A1:Board 3.3V	Voltage (02h) / Threshold (01h)	2204h / 2204h / 1212h	Board 3.3V supply	Y
16h / A1:Pwr Good	Power supply (08h) / OEM (77h)	0000h / 0000h / 0887h	States of all power lines	N
17h / A1:Pwr Good Evt	Power supply (08h) / OEM (77h)	0000h / 0887h / 0887h	Power fail events for all power lines	Y
18h / A1:CPU status	Processor (07h) / Sensor-specific (6Fh)	0463h / 0400h / 04E3h	CPU aggregate status	Y
19h / A1:FWH0 Boot Err	Boot Error (1Eh) / Sensor-specific (6Fh)	0008h / 0008h / 0008h	Firmware Hub 0 boot error	Y
1Ah / A1:FWH1 Boot Err	Boot Error (1Eh) / Sensor-specific (6Fh)	0008h / 0008h / 0008h	Firmware Hub 1 boot error	Y
1Bh / A1:POST Value	OEM Post Value (C6h) / OEM (78h)	0000h / 0000h / 00FFh	POST Value (from host I/O port 80h)	N
1Ch / A1:Lan AMC0 Link	LAN (27h) / Sensor-specific (6Fh)	0000h / 0000h / 0003h	LAN link status – AMC port 0	N
1Dh / A1:Lan AMC1 Link	LAN (27h) / Sensor-specific (6Fh)	0000h / 0000h / 0003h	LAN link status – AMC port 1	N
1Eh / A1:Lan Front0 Lk	LAN (27h) / Sensor-specific (6Fh)	0000h / 0000h / 0003h	LAN link status – Front port 0 (upper)	N
1Fh / A1:Lan Front1 Lk	LAN (27h) / Sensor-specific (6Fh)	0000h / 0000h / 0003h	LAN link status – Front port 1 (lower)	N

4.2 Sensor Thresholds

The AM4022 CPU module is available for two different operating temperature ranges. For each operating temperature range, a set of temperature thresholds for the sensors is defined. The standard temperature range uses thresholds defined by Table 11, and the extended temperature range uses thresholds defined by Table 12. Table 13 provides voltage sensor thresholds.

Table 11: Thresholds - Standard Temperature Range

Sensor Number / ID String	0Eh / A1:Temp CPU	0Fh / A1:Temp PCH	11h / A1:Temp Air
Upper non-recoverable	115 °C	118 °C	100 °C
Upper critical	105 °C	108 °C	90 °C
Upper non-critical	95 °C	98 °C	80 °C
Normal max.	90 °C	93 °C	75 °C
Nominal	80 °C	83 °C	65 °C
Normal min.	3 °C	3 °C	0 °C
Lower non-critical	1 °C	n.a.	-5 °C
Lower critical	n.a.	n.a.	-7 °C
Lower non-recoverable	n.a.	n.a.	-10 °C

Table 12: Thresholds - Extended Temperature Range

Sensor Number / ID string	0Eh / A1:Temp CPU	0Fh / A1:Temp PCH	11h / A1:Temp Air
Upper non-recoverable	115 °C	118 °C	100 °C
Upper critical	105 °C	108 °C	90 °C
Upper non-critical	95 °C	98 °C	80 °C
Normal max.	90 °C	93 °C	75 °C
Nominal	80 °C	83 °C	65 °C
Normal min.	3 °C	3 °C	0 °C
Lower non-critical	1 °C	n.a.	-40 °C
Lower critical	n.a.	n.a.	-42 °C
Lower non-recoverable	n.a.	n.a.	-45 °C

Table 13: 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

Kontron’s OEM specific sensor types and sensor event type codes are presented in the following table.

Table 14: 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	
Config Error (CEh)	6Fh (sensor type specific)	Sensor-specific Offset	Event
		00h	Configuration OK, no error.
		01h	Reserved
		02h	Unsupported Board
		03h	Unsupported Hardware
		04h	Reserved
		05h	Reserved
		06h	Reserved
		07h	Reserved
		08h	PCIe Configuration Conflict with (active) uEFI setting
		09h	PCIe (uEFI FW bank 0) – conflict
		0Ah	PCIe (uEFI FW bank 1) – conflict
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

Table 14: OEM Event/Reading Types (Continued)

OEM SENSOR TYPE (CODE)	OEM EVENT/READING TYPE (CODE)	DESCRIPTION	
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
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

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 / 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 SPI 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 an 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 MMC Firmware Configuration

For initial setup and to get some basic information on the AM4022 MMC, use the AM4022 uEFI Shell or external IPMI access. For further information refer to the AM4022 uEFI BIOS User Guide, Chapter 6, uEFI Shell.

Beside 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 AM4022 MMC.

For a complete list of the **kipmi** parameters, refer to the AM4022 uEFI BIOS User Guide, Chapter 6, uEFI Shell.

5.3 KCS Interface Interrupt

The default factory setting of a AM4022 for its KCS interface is IRQ11. When changing the configuration, the uEFI BIOS 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 (11 or none)

This interrupt configuration is needed by the operating system’s KCS interface kernel driver when it is loaded.

Changing the KCS interrupt number using the **kipmi irq** uEFI Shell command requires a restart of the uEFI BIOS. To restart the AM4022, issue the **reset** uEFI Shell command.

5.4 Firmware / Module Identification

IPMI provides two methods to identify the AM4022 MMC firmware:

- Issuing the IPMI Command `Get Device ID`
- Reading 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. For further information refer to Table 2, Related Publications.

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® ARM7 microcontroller)
- Product ID = identifies the firmware (its board family firmware)
- Firmware revision (byte 4:5) reflects the version of the running firmware, which will change after firmware update.
- SDR revision (byte 13, OEM extension) will be incremented with each firmware update

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

Device Locator Record

The Device Locator Record (SDR Type 12h) contains a Device ID String which identifies the MMC as AM4022 MMC. Additionally, some run-time information such as AMC slot and slot-dependent IPMB address is available in this record.

For example, when using the Linux “ipmitool” on a AM4022 placed in the first AMC slot of a µTCA system, by calling:

```
ipmitool sdr list mcloc
```

the following information is displayed:

```
A1:AM4022 | … @72h | ok
```

5.5 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 timeout the boot code performs an automatic rollback to the “last known good” operational code.

5.5.1 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.

5.5.2 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, for example, the following command:

```
ipmitool hpm upgrade <HPM.1_FWFile>.hpm all
```

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

```
ipmitool hpm activate
```

Detailed information about the currently active firmware versions or the redundant copies can be obtained using the commands mentioned below.

To obtain detailed version information of the active MMC firmware, use the following command:

```
ipmitool hpm compprop 1 1
```

To obtain the version of the rollback copy (only valid if a newly downloaded firmware is already activated), use the following command:

```
ipmitool hpm compprop 1 3
```

To obtain the version of the newly downloaded MMC firmware (only valid after download and before activation), use the following command:

```
ipmitool hpm compprop 1 4
```

To obtain detailed information about the MMC’s HPM.1 upgrade capabilities, use the following command:

```
ipmitool hpm targetcap
```

To perform a manual rollback to the previously good firmware image, use the following command:

```
ipmitool hpm rollback
```

5.5.3 Firmware Upgrade - “ipmitool fwum”

“ipmitool” version 1.8.9 doesn’t support HPM.1 correctly. Tool versions prior to this do not support HPM.1 at all.

The firmware upgrade procedure starts with downloading the binary firmware file using, for example, the following command:

```
ipmitool fwum download <Binary_FWFile>.bin
```

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

```
ipmitool fwum upgrade
```

Detailed information about the currently active firmware versions and the redundant copies can be obtained using the following command:

```
ipmitool fwum status
```

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

```
ipmitool fwum info
```

To perform a manual rollback to the previously good firmware image, use the following command:

```
ipmitool fwum rollback
```

6. FRU Information

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

6.1 FRU Version Identification

The FRU data fields, as defined in the IPMI specification 2.0, are used to record the version of the FRU installed. The revision number is incremented for each new release of FRU data.

Example of board FRU ID: “STD_R01”

Example of product FRU ID: “STD_R01”

6.2 FRU Data Update

Update of the FRU data can be done via regular IPMI FRU device commands. The correct FRU data must be prepared at the factory.

7. E-Keying

E-Keying has been defined in the AMC.0 R2.0 Specification to prevent module damage, prevent malfunction, and 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).

The DIP Switch SW2 can be used to forcibly disable some AMC ports and to configure the PCI Express reference clock if required.

The AM4022 does not have provisions for an external PCIe clock input (FCLKA). For this reason it does not support Spread Spectrum clocking (SSC) which is also indicated in the module’s FRU data.

Please refer to the AM4022 User Guide for further information.

8. uEFI BIOS Failover Control - Automatic SPI Boot Flash Selection

The uEFI BIOS code is stored in two different SPI Boot flash devices designated as the standard SPI boot flash and the recovery SPI boot flash.

By default, the uEFI BIOS code stored in the standard SPI boot flash is executed first. If this fails, the uEFI BIOS code in the recovery SPI boot Flash is then executed.

During boot-up, the uEFI BIOS reports its operational status to the MMC within a given time. If the status is "failed" or not reported within the given time, the MMC selects the recovery SPI boot flash, resets the board's processor, and waits for the status report from the uEFI BIOS again.

In the event the recovery boot operation fails, the MMC reports it, but takes no further action of its own.

When a boot operation fails, a "Boot Error - Invalid boot sector" event is asserted for the related sensor:

- "FWH0 Boot Err" sensor indicates the standard SPI boot flash has failed
- "FWH1 Boot Err" sensor indicates the recovery SPI boot flash has failed

For further information regarding SPI boot flash selection, refer to Chapter 3.2, Table 6.

9. OS Boot Order Selection by OEM IPMI

Normally the uEFI BIOS will apply the OS boot order which was selected in the uEFI BIOS menu “uEFI Boot/Boot Option Priorities”. But there is another alternative boot order which is stored in the IPMI controller's non-volatile memory. This boot order can be set and read by IPMI OEM commands. At payload start the IPMI controller writes this boot order into a register where the uEFI BIOS can read it. If this IPMI controller's boot order has a non-zero value, the uEFI BIOS will use it instead of its own boot order.

10. Hot Swap

As a hot-swappable field replaceable unit (FRU), the AM4022 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 AM4022 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 shutdown has to be signalled back to the MMC. Because the AM4022 supports ACPI, an OS on the payload side which supports this too makes shutdown very easy. If the OS doesn't support ACPI, there is a special method to be used.

10.1 Method 1: The Payload OS Supports ACPI

Requirements:

- The ACPI daemon must be active.
- An ACPI power button event must result in a sleep state.

Hot swap operation sequence 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 shutdown of the payload software system.
- At the end of shutdown, 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 that some shelf managers or MCHs use a timeout to simply switch off 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 Chapter 3.4, OEM Module Quiescence Feedback.

10.2 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 AM4022. This daemon communicates cyclically with the MMC for the exchange of states, commands and acknowledges. 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.

Hot swap operation sequence 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 shutdown of the payload software system.
- At the end of the shutdown 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 Chapter 3.4, OEM Module Quiescence Feedback.

11. LAN Functions

11.1 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+ protocol for the packing of the data to be transferred. The RMCP/RMCP+ protocol uses the TCP port 623 by default.

While IOL serves to transport IPMI commands and their responses, SOL serves to transport any serial data. In each case, the MMC serves as a protocol encoder and decoder. IOL is able to use both RMCP and RMCP+ protocols. SOL works only with the RMCP+ protocol.

Please note that IOL and SOL need the Ethernet device to be powered. Therefore, the module (payload) must be fully powered.

11.2 Setting Up the Ethernet Channel

There are two methods to configure the LAN settings (IOL/SOL) for the four Ethernet channels:

- By use of the **kipmi net** uEFI Shell command in the uEFI BIOS

- By use of the open tool “ipmitool” or IPMI commands

The setup methods are compatible, i.e. both methods show the parameters which are set by the other one.

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 port 623 is not used in parallel by payload and IOL/SOL.

The four Ethernet ports provided by the AM4022 are assigned to the following channels:

Channel 1: AMC port 0

Channel 2: AMC port 1

Channel 3: Front port 0, GbE C, upper connector (J4)

Channel 4: Front port 1, GbE D, lower connector (J3)

11.3 Basic Setup from uEFI Shell

With the **'kipmi net'** command from uEFI Shell some basic settings such as IP address, subnet mask and gateway address can be setup for all of the four Ethernet channels.

11.4 Setup by “ipmitool” or 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 standard IPMI commands as defined in the IPMI specification.

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

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

11.5 Setup of User Accounts 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 user accounts for a channel can be listed using the following command:

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

For every channel, the AM4022 has 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 the **`ADMINISTRATOR`** password is preset with **`admin`**.

Changed accounts and passwords stay valid after payload power-off.

The accounts must be activated using the following command:

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

11.6 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 the RMCP+ protocol. The data transferred are IPMI commands and the responses to them.

To enable LAN support after parameter setup this command must 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 generic 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.info2:

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

This uses many generic IPMI commands to get the information needed.

11.7 Serial Over LAN (SOL)

Serial over LAN connects the COM1 or /dev/ttyS0 respectively of the AM4022'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 AM4022 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 AM4022's serial interface.

The MMC firmware supports only “straight password authentication” SOL sessions with default privilege level USER.

Opening an SOL session requires special parameters as shown below:

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

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

- To a user program on the AM4022 payload
- To the uEFI BIOS. Refer to the Main Setup menu, Serial Port Console Redirection function in the AM4022 uEFI BIOS User Guide. The serial parameters can be set via this function.
- 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 using the following command:

```
ipmitool sol set
```

This command shows all options that can be set.

Further options are listed after issuing the following command:

```
ipmitool sol help
```

12. OS Support / Tools

12.1 Linux Tools

OpenIPMI - KCS driver

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

“ipmitool”

Another very useful all-in-one tool is the “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 access other Management Controllers via IPMB. Before “ipmitool” can be used, the OpenIPMI driver mentioned above must be loaded too.

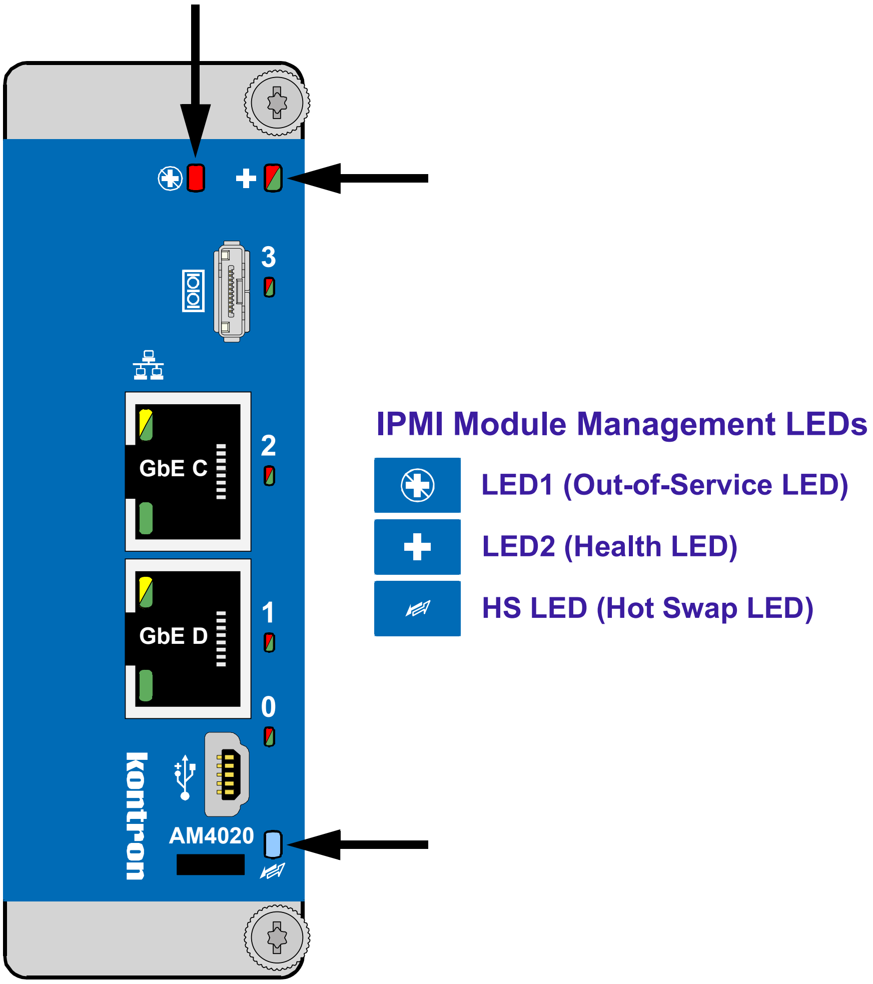
12.2 OS Support - Board Support Packages

To see which Operating Systems are supported refer to the board's data sheet. Please visit http://www.kontron.com to get the data sheet. Please 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.

13. IPMI Module Management LEDs

There are three IPMI Module Management LEDs on the front panel of the AM4022. The following figure illustrates a Full-Size AM4022 module and the location of the LEDs.



The following table describes the functions of the IPMI Module Management LEDs.

Table 15: Module Management LED Functions

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/amber/red	off	Payload is off; module is not powered	By user: • Only lamp test
		green	Module is healthy (normal operation) and all related sensors are within the specified range	
		amber	Payload is on and at least one sensor is out of range	
		red	Reserved	
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 By user: • Only lamp test
		blinking	Module hot swap in progress; module not ready for extraction	
		off	Module is in normal operation	

This page has been intentionally left blank.