

AIC[®]

BMC

**Special IPMI Commands
and Sensors**

Table of Contents

Chapter 1. Special IPMI Commands	3
1.1 IPMI Standard Command	3
1.1.1 Get Device ID (NetFn: 0x06, Command: 0x01)	3
1.2 IPMI AIC OEM Command	4
1.2.1 Get BMC Firmware Name (NetFn: 0x3A, Command: 0x33)	4
1.2.2 Set all LED pattern (NetFn: 0x3A, Command: 0x02)	4
1.2.3 Get NTP synchronization status (NetFn: 0x3A, Command: 0x72)	4
1.2.4 Get NTP configuration (NetFn: 0x36, Command: 0xB7)	4
1.2.5 Get PWM group output (NetFn: 0x36, Command: 0xFB)	5
1.2.6 Set PWM group manual control (NetFn: 0x36, Command: 0xFA)	5
1.2.7 Set PWM group optimal control (NetFn: 0x36, Command: 0xF9)	5
1.2.8 Get power on minutes (NetFn: 0x36, Command: 0xD6)	5
1.2.9 Get multiple sensor reading (NetFn: 0x36, Command: 0xBD)	5
1.2.10 Get CPU Info (NetFn: 0x3A, Command: 0x2B)	6
1.2.11 Master write read (NetFn: 0x3A, Command: 0x6A)	6
1.2.12 Set FAN PWM setting for AC power on (NetFn: 0x36, Command: 0xFC)	6
1.2.13 Get FAN PWM setting for AC power on (NetFn: 0x36, Command: 0xFD)	7
1.2.14 Set_FRU_Mfg_Date (NetFn: 0x3A, Command: 0xF0)	7
1.2.15 Secure_Write_FRU (NetFn: 0x3A, Command: 0xF1)	8
1.2.16 Buzzer_Control (NetFn: 0x3A, Command: 0xE3)	9
1.2.17 Set GPU Temperature Compensation (NetFn: 0x3C, Command: 0x5A)	9
Chapter 2. Specific Sensors in SDR	11
2.1 PSUX_Status	11

Document Release History

Release Date	Version	Author	Update Content
Nov. 7, 2025	1.5	Jethro Yeh	Add the following two OEM commands. - 1.2.14 Set_FRU_Mfg_Date - 1.2.15 Secure_Write_FRU
Nov. 17, 2025	1.6	Jethro Yeh	Modify the contents of 1.2.15 Secure_Write_FRU. - FRU field data must be 2 to 24 bytes.
Apr. 28, 2026	1.7	Alan Huang	Add the Buzzer_Control OEM commands
May. 21, 2026	1.8	Donna Hu	Add GPU temperature compensation OEM commands

Chapter 1. Special IPMI Commands

1.1 IPMI Standard Command

1.1.1 Get Device ID (NetFn: 0x06, Command: 0x01)

Only the fields highlighted in green are AIC specific.

The firmware version is <Firmware Revision 1>.<Firmware Revision 2>.<Byte 1 of Auxiliary Firmware Revision Information>.

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2	Device ID
	3	Device Revision
	4	Firmware Revision 1 (Major Firmware Revision)
	5	Firmware Revision 2 (Minor Firmware Revision)
	6	IPMI Version
	7	Additional Device Support
	8:10	Manufacturer ID, LS Byte first.
	11:12	Product ID, LS Byte first.
	13	Byte 1 of Auxiliary Firmware Revision Information
	14	Byte 2 of Auxiliary Firmware Revision Information
	15	Byte 3 of Auxiliary Firmware Revision Information
	16	Byte 4 of Auxiliary Firmware Revision Information

1.2 IPMI AIC OEM Command

Only the IPMI standard commands with AIC specific implementation are listed below.

1.2.1 Get BMC Firmware Name (NetFn: 0x3A, Command: 0x33)

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2:14	BMC Platform Name

1.2.2 Set all LED pattern (NetFn: 0x3A, Command: 0x02)

	Byte	Data Field
Request Data	1	Control ID LED and health LED 00h : Lock LED status and turn off all LED 01h : Lock LED status and turn on all LED FFh : Unlock LED
Response Data	1	Completion Code

1.2.3 Get NTP synchronization status (NetFn: 0x3A, Command: 0x72)

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2	NTP synchronization status 00h : Automatic sync mode, sync failed. 01h : Automatic sync mode, sync is successful. 02h : Automatic sync mode, but sync has not yet succeeded. 03h : Manual mode.

1.2.4 Get NTP configuration (NetFn: 0x36, Command: 0xB7)

	Byte	Data Field
Request Data	1	Parameter : 01h : Primary server name 02h : Second server name 03h : NTP configuration status
Response Data	1	Completion Code
	2:N	For Parameter 01h: Primary NTP server name For Parameter 02h: Second NTP server name For Parameter 03h: 00h : Disable status of NTP. 01h : Enable status of NTP. 02h : Failure status of NTP.

1.2.5 Get PWM group output (NetFn: 0x36, Command: 0xFB)

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2	Fan mode: 00h : Manual control mode. 01h : Optimal control mode.
	3	Protect status: 00h : Protection mode disabled. 01h : Protection mode enabled.
	4:19	Display currently all duty for 16 PWM
	20:35	Display manually all duty for 16 PWM

1.2.6 Set PWM group manual control (NetFn: 0x36, Command: 0xFA)

	Byte	Data Field
Request Data	1	Protect setting 00h : Disable protection mode. 01h : Enable protection mode.
	2:17	Set duty for 16 PWM
Response Data	1	Completion Code

1.2.7 Set PWM group optimal control (NetFn: 0x36, Command: 0xF9)

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code

1.2.8 Get power on minutes (NetFn: 0x36, Command: 0xD6)

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2:5	Power on minutes

1.2.9 Get multiple sensor reading (NetFn: 0x36, Command: 0xBD)

	Byte	Data Field
Request Data	1	No.1 Sensor number
	2	No.1 Sensor LUN
	3:4	No.2 Sensor number No.2 Sensor LUN
	5:N	Sensor number Sensor LUN
Response Data	1	Completion Code
	2	No.1 Sensor reading

3	No.1 Sensor LUN
4:5	No.2 Sensor reading
	No.2 Sensor LUN
6:N	Sensor number Sensor LUN

1.2.10 Get CPU Info (NetFn: 0x3A, Command: 0x2B)

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2:33	CPU information. If the platform supports dual CPUs, the information for both CPUs is identical.

1.2.11 Master write read (NetFn: 0x3A, Command: 0x6A)

	Byte	Data Field
Request Data	1	Channel number It's ignored when Bus Type = 0x1.
	2	Bus ID It's 0x0 when Bus Type = 0x0.
	3	Bus Type 0x0: Public IPMB bus. The Channel Number is used to select the target bus. 0x1: Private bus. The Bus ID is used to select the target bus. Others: Invalid
	4	8-bit Slave Address
	5	Read Count The maximum is 50 bytes.
Response Data	6:N	Data to write. The maximum is 50 bytes.
	1	Completion Code
	2:N	Bytes read from the specified slave address. The maximum is 50 bytes.

1.2.12 Set FAN PWM setting for AC power on (NetFn: 0x36, Command: 0xFC)

	Byte	Data Field
Request Data	1	AC PWR On FAN PWM Duty Cycle Mode: 0x0: Use Default value. (Default 50) 0x1: Use manual settings. Use "AC PWR On FAN PWM Duty Cycle" field as setting.
	2	AC PWR On FAN PWM Duty Cycle: Range: 30~100 Note: Ignored when "AC PWR On FAN PWM Duty Cycle Mode" is 0x0; any value is acceptable.
Response Data	1	Completion Code

1.2.13 Get FAN PWM setting for AC power on (NetFn: 0x36, Command: 0xFD)

	Byte	Data Field
Request Data	-	
Response Data	1	Completion Code
	2	AC PWR On FAN PWM Duty Cycle Mode. 0x0: Use Default value. (Default 50) 0x1: Use manual settings. Use "AC PWR On FAN PWM Duty Cycle" field as setting.
	3	AC PWR On FAN PWM Duty Cycle Note: This field is disregarded when the "AC PWR On FAN PWM Duty Cycle Mode" is 0x0.
	4:7	PWM Number Mask (For AIC internal use only. Please ignore)

1.2.14 Set_FRU_Mfg_Date (NetFn: 0x3A, Command: 0xF0)

This command updates the Manufacturing Date field in the FRU.bin file (Board Info Area).

The command **does not write to EEPROM** – the update only applies to the temporary **FRU.bin** file.

	Byte	Data Field
Request Data	1	Target FRU Device ID 0x00 = Mainboard FRU
	2~4	Board Manufacturer Date 3-byte "minutes since 1/1/96 UTC" format, per IPMI FRU spec. Low byte first.
Response Data	1	Completion Code 0x00: Success 0xC7: Request data length invalid 0xFF: Checksum invalid or file I/O error

Notes:

1. The 3-byte "Board Manufacturer Date" date format follows the IPMI FRU specification.
2. Checksum verification is performed before updating. If the checksum is invalid, the command returns 0xFF and aborts.
3. Refer to the **Update FRU Guide** document for details about the relationship between FRU, **FRU.bin**, and **EEPROM**.

Special Note on Board Manufacturer Date:

1. Format is 3 bytes, little-endian (LSB first).
2. Unit: Minutes since base date: 1996-01-01 00:00:00 UTC.
3. Maximum representable timestamp: 2027-11-24 20:15 UTC
4. Handling of Out-of-Range Dates:

- The sender **MUST** check whether the provided timestamp exceeds the maximum supported time (2027-11-24 20:15 UTC).
- If the time is after this maximum, the field value **MUST** be written as 0x00 0x00 0x00, which is interpreted as "unspecified" per IPMI FRU specification.

1.2.15 Secure_Write_FRU (NetFn: 0x3A, Command: 0xF1)

This command securely writes all FRU data from **FRU.bin** back into **EEPROM**.

A password must be provided to authorize the write operation.

After 5 consecutive invalid password attempts, the command is disabled until the next BMC restart.

Notes:

1. All FRU area checksums are verified before writing.
2. The command will abort if any FRU field data exceeds **2~24 bytes**.
3. This command should be executed after using "ipmitool fru edit" to make FRU modifications persistent after a BMC reset.
4. Refer to the **Update FRU Guide** document for details about the relationship between FRU, **FRU.bin**, and **EEPROM**.

	Byte	Data Field
Request Data	1	Target FRU Device ID 0x00 = Mainboard FRU
	2~N	Password ASCII password string for security validation.
Response Data	1	Completion Code 0x00: Success 0xC4: Out of space (FRU field data is not between 2 and 24 bytes) 0xC9: Parameter out of range (invalid length) 0xCC: Invalid data field (wrong password) 0xD5: Operation not supported (too many password failures) 0xFF: Unspecified error

Two password levels are supported:

Permission	Description
AIC_Factory	Writes all FRU fields to EEPROM.
AIC_Customer	Writes all fields except <i>ChassisSerial</i> and <i>BoardSerial</i> to EEPROM.

1.2.16 Buzzer_Control (NetFn: 0x3A, Command: 0xE3)

This command controls the buzzer beep patterns with the given parameter.

The system buzzer must be connected to the motherboard's buzzer header for the buzzer function to work.

Notes:

1. The long beep will be executed first, followed by the short beep.
2. The **Number of long beeps** and **Number of short beeps** are accepted only when **Byte 1** is set to **0x1(On)**.
3. Single terms of long beeps is **ON** for **600ms** and **OFF** for **300ms**.
4. Single terms of short beeps is **ON** for **200ms** and **OFF** for **300ms**

	Byte	Data Field
Request Data	1	Buzzer state 0x0 = Off 0x1 = On
	2	Number of long beeps
	3	Number of short beeps
Response Data	1	Completion Code 0x00: Success 0xC7: Request data length invalid 0xCC: Invalid data field 0xFF: Unspecified error

1.2.17 Set GPU Temperature Compensation (NetFn: 0x3C, Command: 0x5A)

This command sets the GPU temperature compensation offset value stored in EEPROM. The offset is applied to the GPU temperature sensor reading before the final temperature is reported.

A positive offset increases the reported GPU temperature, while a negative offset decreases the reported GPU temperature

	Byte	Data Field
Request Data	1	Compensation offset value (0x00~0xFF)
Response Data	1	Completion Code 0x00 : Success 0xD6 : EEPROM write failure or internal error

Notes:

1. The compensation offset is stored in EEPROM.

2. The offset value is represented as an 8-bit signed integer using two's complement format.
3. The valid offset range is from -128 to +127.
4. The configured offset will take effect on subsequent GPU temperature readings.
5. Example:
 - i. Set compensation offset to -2 (ipmitool raw 0x3C 0x5A 0xFE)
 - ii. Set compensation offset to 2 (ipmitool raw 0x3C 0x5A 0x02)

Chapter 2. Specific Sensors in SDR

2.1 PSU_x_Status

PMBus STATUS_WORD (79h) message contents			SDR Sensor Bit Number
Byte	Bit Number	Status Bit Name	
Low	7	BUSY	X
	6	OFF	Bit6
	5	VOUT_OV_FAULT	Bit5
	4	IOUT_OC_FAULT	Bit4
	3	VIN_UV_FAULT	Bit3
	2	TEMPERATURE	Bit2
	1	CML	Bit1
	0	NONE OF THE ABOVE	Bit0
High	7	VOUT	Bit14
	6	IOUT/POUT	Bit13
	5	INPUT	Bit12
	4	MFR_SPECIFIC	Bit11
	3	POWER_GOOD#	Bit10
	2	FANS	Bit9
	1	OTHER	Bit8
	0	UNKNOWN	Bit7

p.s: **x** represents the number of PSUs in the system.

Example: There are 2 PSUs in the system. It represents PSU1_Status and PSU2_Status.