AAEON IPC EC Command Set User Guide

Most Confidential	
Confidential	V
Internal	V
Public	

May 2020
Revision 0.1(draft)

Revision History

Revision	Description	Date	Editor
0.1(draft)	Initial Release	2020/04/08	Ayers_Deng

Contents

Chapter 1 Introduction

1.1 Principle

Text

Chapter 2 AAEON IPC EC Command Set

Host side OS utility/application can use IO port 0x584(DAT port) and 0x585(CMD port) to access/communicate with EC FW.

2.1 I/O Command Set

<table>
<tr><th>CMD</th><th>Function</th><th>Description/Usage</th></tr>
<tr><td>0x85</td><td>Get Panel brightness</td><td>Send Byte0:
Panel index (follow motherboard printing number)

Received Byte0:
Brightness percentage level 0 to 10(100%)</td></tr>
<tr><td>0x86</td><td>Set Panel brightness</td><td>Byte0:
Panel index (follow motherboard printing number)
Byte1:
Brightness percentage level 0 to 10(100%)</td></tr>
<tr><td>0x8E</td><td>Read MISC function</td><td>Send Byte0:
Index number (refer to “CMD 8Eh/9Eh Index Table”)</td></tr>
<tr><td>0x8F</td><td>Read HW ID</td><td>Send Byte0:
0x00 - Board ID
0x01 - Panel ID

Received Byte0:
HW ID value</td></tr>
<tr><td>0x97</td><td>Read FAN RPM</td><td>Send Byte0:
<table><tr><th>High nibble[7:4] - Type</th><th>Low nibble[3:0] - Number</th></tr><tr><td>0 - CPU</td><td>x - refer Note</td></tr><tr><td>1 - System</td><td>x - refer Note</td></tr><tr><td>2 - Chassis</td><td>x - refer Note</td></tr><tr><td>3 - Power Supply</td><td>x - refer Note</td></tr><tr><td colspan="2">0xFF - Reserved for special case</td></tr><tr><td colspan="2">Note:
Number follow motherboard printing number, or zero as default(if exist)</td></tr></table>
Received
Byte0: High Byte
Byte1: Low Byte</td></tr>
<tr><td>0x9E</td><td>Write MISC function</td><td>Byte0:
Index number (refer to “CMD 8Eh/9Eh Index Table”)
Byte1:
Write Data</td></tr>
<tr><td>0xAA</td><td>Read I2C/SMBus</td><td>Todo (ARStmp)</td></tr>
<tr><td>0xAB</td><td>Write I2C/SMBus</td><td>Todo (ARStmp)</td></tr>
</table>

<table>
<tr><td>0xBB</td><td>Read EC FW version</td><td>Received<table>
<tr><td>Case 1 - Standard Platform EC</td><td>Byte0: 0x09 (return total bytes, included Byte0)

Byte1: S - Standard platform EC
Byte2: I/A - Intel/AMD chipset
Byte3/4/5: Platform name
Byte6: F/H/T - EC kernel version type (Formal/Hot Fix/Test)
Byte7/8: version number

Example:
SICMLFxx = Standard Intel CometLake Formal EC kernel FW</td></tr>
<tr><td>Case2 - for Project / Customize dedicated EC</td><td>Byte0: 0x13 (return total bytes, included Byte0)

(part1)
Byte1: P/C - Project/Customize dedicated EC
Byte2: I/A - Intel/AMD chipset
Byte3/4/5: Platform name
Byte6: F/H/T - EC kernel version type (Formal/Hot Fix/Test)
Byte7/8: version number
Byte9: ‘.’ - ASCII code 0x2E

(part2)
Byte 10 to 14: PROJECT_TAG
Byte15: ‘.’ - ASCII code 0x2E

(part3)
Byte16: F/T - FW version type (Formal/Test)
Byte17/18: version number

Example:
PICMLT01.SMH41.T01 =
part1 - Project Intel CML Test EC kernel FW
part2 - for project SMS-H410
part3 - first Test version</td></tr>
</table></td></tr>
</table>

<table>
<tr>
<td>0xC6</td>
<td>Get Watchdog status</td>
<td>
<table>
<tr><th>Type</th><th>Value</th></tr>
<tr><td>WDT status</td><td>Send Byte0:
0x00

Received Byte0:
0 - Inactive/Stop
1 - Activating</td></tr>
<tr><td>Second/Minute mode</td><td>Send Byte0:
0x01

Received Byte0:
0 - Second (default)
1 - Minute</td></tr>
<tr><td>Countdown value or Current Remaining value</td><td>Send
Byte0: 0x02
Byte1:
0 - Countdown value
1 - Current Remaining Value

Received
Byte0: second
Byte1: minute (only Minute mode report the byte)</td></tr>
<tr><td>WDT Expired</td><td>Send Byte0:
0x03

Received Byte0:
0 - not Expired
1 - Expired
Note: EC also set “WDT set(expired) LED” if the project supported the feature)</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td>0xC7</td><td>Set Watchdog</td><td>
<table>
<tr><th>Type</th><th>Value</th></tr>
<tr><td>WDT Stop/Resume</td><td>Byte0:
0x00
Byte1:
0 - Stop WDT (EC will stop WDT counter and record currently timer Remaining Countdown value. Turn off “WDT active LED”, if the project supported WDT LED feature)
1 - Resume WDT (EC will restore previous Remaining Countdown value. Turn on “WDT active LED”, if the project supported WDT LED feature)</td></tr>
<tr><td>Second/Minute mode</td><td>Byte0:
0x01
Byte1:
Definition the same as CMD 0xC6 Second/Minute mode “Received Byte0”
Note:
1. If WDT timer is activating, must Inactive/Stop WDT first through clear “Countdown value” or set “WDT Stop/Resume” value to 0
2. Check status by “WDT status” field</td></tr>
<tr><td>Countdown value</td><td>Byte0:
0x02
Byte1:
Value (0~255 second or minute)
Note:
1. Set value as 0 - clear countdown value and inactive WDT function
2. Set valid value - active WDT function and start countdown
3. Follow “WDT Stop/Resume” LED behavior</td></tr>
<tr><td>Clear WDT Expired</td><td>Byte0:
0x03
Note: EC also clear “WDT set(expired) LED” if the project supported the feature)</td></tr>
<tr><td>Reload/Retrigger WDT countdown value</td><td>Byte0:
0x04
Note:
1. EC according Second/Minute mode to Reload previous Countdown value
2. Start WDT function</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>0xCC</td><td>Read thermal related</td><td>Send Byte0:
Index number (refer to “CMD CCh/CDh Index Table”)</td></tr>
<tr><td>0xCD</td><td>Write thermal related</td><td>Byte0:
Index number (refer to “CMD CCh/CDh Index Table”)
Byte1:
Write Data</td></tr>
<tr><td>0xD0</td><td>Get DIO pin status</td><td>Send Byte0:
DIO pin index (follow motherboard printing number)

Received Byte0:
<table>
<tr><th>High nibble[7:4] - Type</th><th>Low nibble[3:0] - Value</th></tr>
<tr><td>0 - Output</td><td>0 - Low
1 - High</td></tr>
<tr><td>1 - Input</td><td>0 - Low
1 - High</td></tr>
</table></td></tr>
<tr><td>0xD1</td><td>Set DIO pin</td><td>Byte0:
DIO pin index (follow motherboard printing number)
Byte1:
<table>
<tr><th>High nibble[7:4] - Type</th><th>Low nibble[3:0] - Value</th></tr>
<tr><td>0 - Output</td><td>0 - Low
1 - High</td></tr>
<tr><td>1 - Input</td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr>
<td>0xD2</td>
<td>Get LED status</td>
<td>Send Byte0:
<table>
<tr><th>High nibble[7:4] - Type</th><th>Low nibble[3:0] - Number</th></tr>
<tr><td>0 - Power LED</td><td>0 - Power on LED (S0)
1 - AC IN LED
2 - Reserved
3 - Sleep LED (S3)
4 - Hibernate LED (S4)
5 - Soft off LED (S5)</td></tr>
<tr><td>1 - Battery LED</td><td>x - refer Note</td></tr>
<tr><td>2 - Watchdog LED</td><td>0 - WDT active LED
1 - WDT set(expired) LED</td></tr>
<tr><td>3 - Cap Lock LED</td><td>0</td></tr>
<tr><td>4 - Wireless LED</td><td>0</td></tr>
<tr><td>5 - Temperature Alert LED</td><td>x - refer Note</td></tr>
<tr><td>6 - FAN Alert LED</td><td>x - refer Note</td></tr>
<tr><td>7 - HDD Alarm LED (Note: On/Off controlled by SW)</td><td>x - refer Note</td></tr>
<tr><td>8 - Debug LED</td><td>0</td></tr>
<tr><td colspan="2">0xFF - Reserved for special case</td></tr>
<tr><td colspan="2">Note:
Number follow motherboard printing number, or zero as default(if exist)</td></tr>
</table>
Received Byte0:
<table>
<tr><th>LED type</th><th>Value</th></tr>
<tr><td>General purpose LED</td><td>0 - LED not active
1 - LED active
2 - Toggle LED On/Off status (only for CMD 0xD3 Set LED Byte1)</td></tr>
<tr><td>Debug LED status</td><td>HW Error:
0x11 - Power failure
0x12 - System unable to power on
0x13 - Processor not installed
0x14 - New Processor
0x15 - Memory not installed
0x16 - Memory error
0x17 - CPU temperature abnormal
0x18 - FAN speed fault
0x19 - Case open
0x1A - Storage not found
SW Error:
0x40 - POST fail
0x41 - Pre-video memory error
0x42 - Pre-video graphics error
0x43 - Bootable volume not found
0x44 - ROM checksum not valid</td></tr>
</table>
</td>
</tr>
<tr>
<td>0xD3</td>
<td>Set LED</td>
<td>Byte0:
Definition the same as CMD 0xD2 “Send Byte0”
Byte1:
Definition the same as CMD 0xD2 “Received Byte0”</td>
</tr>
</table>

0xD4	Get Voltage	Send Byte0: 0x00 - VCORE 0x01 - VCOREREFIN 0x02 - +12V 0x03 - +5V 0x04 - 5VSB 0x05 - 5VDUAL 0x06 - +3.3V 0x07 - 3VSB 0x08 - +1.8V 0x09 - VMEM 0x0A - RTC 0x0B - VBAT Received (Note: Get Voltage may spend 30ms at most) Byte0: Integer part of Voltage value Byte1: High byte of decimal point Byte2: Low byte of decimal point

2.2 CMD 8Eh/9Eh Index Table

The index table intend for miscellaneous functions which access by CMD 0x8E, 0x9E.

Index	Function	Description/Usage
0x40	Get FAN mode	Send Byte1: Definition the same as CMD 0x97 “Send Byte0” Received Byte0: 0x00 - Auto (default) 0x01 - Silent 0x02 - Performance 0x03 - Full speed 0x10 - Manual (Get only. Through Index 0x43 “Set FAN duty” of the 8Eh/9Eh index table to set value that EC will automatically change FAN mode to Manual mode) 0xFF - Disable
0x41	Set FAN mode	Byte1: Definition the same as CMD 0x97 “Send Byte0” Byte2: Definition the same as Index 0x40 - Get FAN mode “Received Byte0”

0x42	Get FAN duty	Send Byte1: Definition the same as CMD 0x97 “Send Byte0” Received Byte0: Duty cycle value 0 to 255
0x43	Set FAN duty	Byte1: Definition the same as CMD 0x97 “Send Byte0” Byte2: Duty cycle value 0 to 255
0x44	Get FAN Alert RPM	Send Byte1: Definition the same as CMD 0x97 “Send Byte0” Received Byte0: High Byte Byte1: Low Byte
0x45	Set FAN Alert RPM	Byte1: Definition the same as CMD 0x97 “Send Byte0” Byte2: High Byte Byte3: Low Byte Note: 1. Once EC detected FAN RPM lower or equal than the settings value, EC will turn on FAN Alert LED (if the project supported the LED). 2. Default FAN Alert RPM was 0.
0x46	Get FAN Alert Status	Send Byte1: Definition the same as CMD 0x97 “Send Byte0” Received Byte0: 0 - Not Alerted 1 - Alerted
0x47	Clear FAN Alert	Byte1: 1. Definition the same as CMD 0x97 “Send Byte0” 2. 0xFF - Clear all of FAN Alerts

2.3 CMD CCh/CDh Index Table

The index table functions intend for thermal related which access by CMD 0xCC, 0xCD.

Index	Function	Description/Usage

0x00	Read Temperature	Send Byte1: (Temperature Source) 0x00 - PECI 0x10 - Thermal Sensor: CPU 0x11 - Thermal Sensor: VCORE 0x12 - Thermal Sensor: Memory 0x13 - Thermal Sensor: PCIe Graphic 0x14 - Thermal Sensor: PCH 0x15 - Thermal Sensor: Ambient 0x20 - Thermal Sensor: System 0xFF - Reserved for special case Received Byte0: Temperature value (degree C)
0x02	Get Sensor Alert Temperature	Send Byte1: Definition the same as Index 0x00 “Send Byte1” Received Byte0: Temperature value (degree C)
0x03	Set Sensor Alert Temperature	Byte1: Definition the same as Index 0x00 “Send Byte1” Byte2: Temperature value (degree C) Note: 1. Once EC detected Sensor temperature higher than the settings value, EC will turn on Temperature Alert LED (if the project supported the LED). 2. No default Alert Temperature.
0x04	Get Sensor Alert Status	Send Byte1: Definition the same as Index 0x00 “Send Byte1” Received Byte0: 0 - Not Alerted 1 - Alerted
0x05	Clear Sensor Alert	Byte1: 1. Definition the same as Index 0x00 “Send Byte1” 2. 0xFF - Clear all of Thermal Sensor Alerts

Chapter 3 AAEON IPC EC Command Set (Extended)

The chapter commands is extended from previous chapter common part, main concept is used by BIOS team since these Command Set was more bottom layer operation.

3.1 I/O Command Set

CMD	Function	Description/Usage
0x87	Get/Set GPIO pin	Send Byte0: Operation Type Byte1: GPIO Port & Pin Received Byte0: (for Read) pin status, 1-High/0-Low