

## 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

Revision History	2
Contents	3
Chapter 1 Introduction	4
1.1 Principle	4
Chapter 2 AAEON IPC EC Command Set	5
2.1 I/O Command Set	5
2.2 CMD 8Eh/9Eh Index Table	11
2.3 CMD CCh/CDh Index Table	12
Chapter 3 AAEON IPC EC Command Set (Extended)	14
3.1 I/O Command Set	14

# 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

CMD	Function	Description/Usage										
0x85	Get Panel brightness	Send Byte0: Panel index (follow motherboard printing number)  Received Byte0: Brightness percentage level 0 to 10(100%)										
0x86	Set Panel brightness	Byte0: Panel index (follow motherboard printing number) Byte1: Brightness percentage level 0 to 10(100%)										
0x8E	Read MISC function	Send Byte0: Index number (refer to “ <a href="#">CMD 8Eh/9Eh Index Table</a> ”)										
0x8F	Read HW ID	Send Byte0: 0x00 - Board ID 0x01 - Panel ID  Received Byte0: HW ID value										
0x97	Read FAN RPM	Send Byte0: <table border="1"><thead><tr><th>High nibble[7:4] - Type</th><th>Low nibble[3:0] - Number</th></tr></thead><tbody><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></tbody></table> Note: Number follow motherboard printing number, or zero as default(if exist)  Received Byte0: High Byte Byte1: Low Byte	High nibble[7:4] - Type	Low nibble[3:0] - Number	0 - CPU	x - refer Note	1 - System	x - refer Note	2 - Chassis	x - refer Note	3 - Power Supply	x - refer Note
High nibble[7:4] - Type	Low nibble[3:0] - Number											
0 - CPU	x - refer Note											
1 - System	x - refer Note											
2 - Chassis	x - refer Note											
3 - Power Supply	x - refer Note											
0x9E	Write MISC function	Byte0: Index number (refer to “ <a href="#">CMD 8Eh/9Eh Index Table</a> ”) Byte1: Write Data										
0xAA	Read I2C/SMBus	Todo (ARStmp)										
0xAB	Write I2C/SMBus	Todo (ARStmp)										

0xBB	Read EC FW version	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>	Case 1 - Standard Platform EC	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	Case2 - for Project / Customize dedicated EC	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
Case 1 - Standard Platform EC	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					
Case2 - for Project / Customize dedicated EC	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					

0xC6	Get Watchdog status	Type	Value
		WDT status	Send Byte0: 0x00  Received Byte0: 0 - Inactive/Stop 1 - Activating
		Second/Minute mode	Send Byte0: 0x01  Received Byte0: 0 - Second (default) 1 - Minute
		Countdown value or Current Remaining value	Send Byte0: 0x02 Byte1: 0 - <a href="#">Countdown value</a> 1 - Current Remaining Value  Received Byte0: second Byte1: minute (only <a href="#">Minute mode</a> report the byte)
		WDT Expired	Send Byte0: 0x03  Received Byte0: 0 - not Expired 1 - Expired Note: EC also set " <a href="#">WDT set(expired) LED</a> " if the project supported the feature)

0xC7	Set Watchdog	Type	Value
		WDT Stop/Resume	<p>Byte0: 0x00</p> <p>Byte1: 0 - Stop WDT (EC will stop WDT counter and record currently timer <a href="#">Remaining Countdown value</a>. Turn off "<a href="#">WDT active LED</a>", if the project supported WDT LED feature)</p> <p>1 - Resume WDT (EC will restore previous <a href="#">Remaining Countdown value</a>. Turn on "<a href="#">WDT active LED</a>", if the project supported WDT LED feature)</p>
		Second/Minute mode	<p>Byte0: 0x01</p> <p>Byte1: Definition the same as <a href="#">CMD 0xC6 Second/Minute mode "Received Byte0"</a></p> <p>Note:</p> <ol style="list-style-type: none"> <li>1. If WDT timer is activating, must Inactive/Stop WDT first through clear "<a href="#">Countdown value</a>" or set "<a href="#">WDT Stop/Resume</a>" value to 0</li> <li>2. Check status by "<a href="#">WDT status</a>" field</li> </ol>
		Countdown value	<p>Byte0: 0x02</p> <p>Byte1: Value (0~255 <a href="#">second or minute</a>)</p> <p>Note:</p> <ol style="list-style-type: none"> <li>1. Set value as 0 - clear countdown value and inactive WDT function</li> <li>2. Set valid value - active WDT function and start countdown</li> <li>3. Follow "<a href="#">WDT Stop/Resume</a>" LED behavior</li> </ol>
		Clear WDT Expired	<p>Byte0: 0x03</p> <p>Note: EC also clear "<a href="#">WDT set(expired) LED</a>" if the project supported the feature)</p>
		Reload/Retrigger WDT countdown value	<p>Byte0: 0x04</p> <p>Note:</p> <ol style="list-style-type: none"> <li>1. EC according <a href="#">Second/Minute mode</a> to Reload previous <a href="#">Countdown value</a></li> <li>2. Start WDT function</li> </ol>



0xCC	Read thermal related	Send Byte0: Index number (refer to “ <a href="#">CMD CCh/CDh Index Table</a> ”)						
0xCD	Write thermal related	Byte0: Index number (refer to “ <a href="#">CMD CCh/CDh Index Table</a> ”) Byte1: Write Data						
0xD0	Get DIO pin status	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>	High nibble[7:4] - Type	Low nibble[3:0] - Value	0 - Output	0 - Low 1 - High	1 - Input	0 - Low 1 - High
High nibble[7:4] - Type	Low nibble[3:0] - Value							
0 - Output	0 - Low 1 - High							
1 - Input	0 - Low 1 - High							
0xD1	Set DIO pin	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>	High nibble[7:4] - Type	Low nibble[3:0] - Value	0 - Output	0 - Low 1 - High	1 - Input	0
High nibble[7:4] - Type	Low nibble[3:0] - Value							
0 - Output	0 - Low 1 - High							
1 - Input	0							

0xD2	Get LED status	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">Note: Number follow motherboard printing number, or zero as default(if exist)</td></tr></table>	High nibble[7:4] - Type	Low nibble[3:0] - Number	0 - Power LED	0 - Power on LED (S0) 1 - AC IN LED 2 - Reserved 3 - Sleep LED (S3) 4 - Hibernate LED (S4) 5 - Soft off LED (S5)	1 - Battery LED	x - refer Note	2 - Watchdog LED	0 - WDT active LED 1 - WDT set(expired) LED	3 - Cap Lock LED	0	4 - Wireless LED	0	5 - Temperature Alert LED	x - refer Note	6 - FAN Alert LED	x - refer Note	7 - HDD Alarm LED (Note: On/Off controlled by SW)	x - refer Note	8 - Debug LED	0	Note: Number follow motherboard printing number, or zero as default(if exist)	
		High nibble[7:4] - Type	Low nibble[3:0] - Number																					
		0 - Power LED	0 - Power on LED (S0) 1 - AC IN LED 2 - Reserved 3 - Sleep LED (S3) 4 - Hibernate LED (S4) 5 - Soft off LED (S5)																					
		1 - Battery LED	x - refer Note																					
		2 - Watchdog LED	0 - WDT active LED 1 - WDT set(expired) LED																					
		3 - Cap Lock LED	0																					
		4 - Wireless LED	0																					
		5 - Temperature Alert LED	x - refer Note																					
		6 - FAN Alert LED	x - refer Note																					
		7 - HDD Alarm LED (Note: On/Off controlled by SW)	x - refer Note																					
		8 - Debug LED	0																					
		Note: Number follow motherboard printing number, or zero as default(if exist)																						
		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 <a href="#">CMD 0xD3 Set LED Byte1</a>)</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>	LED type	Value	General purpose LED	0 - LED not active 1 - LED active 2 - Toggle LED On/Off status (only for <a href="#">CMD 0xD3 Set LED Byte1</a> )	Debug LED status	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																
LED type	Value																							
General purpose LED	0 - LED not active 1 - LED active 2 - Toggle LED On/Off status (only for <a href="#">CMD 0xD3 Set LED Byte1</a> )																							
Debug LED status	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																							
0xD3	Set LED	Byte0: Definition the same as <a href="#">CMD 0xD2 "Send Byte0"</a> Byte1: Definition the same as <a href="#">CMD 0xD2 "Received Byte0"</a>																						

0xD4	Get Voltage	<p>Send Byte0:</p> <p>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</p> <p>Received</p> <p>(Note: Get Voltage may spend 30ms at most)</p> <p>Byte0: Integer part of Voltage value Byte1: High byte of decimal point Byte2: Low byte of decimal point</p>
------	-------------	---

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

0x42	Get FAN duty	Send Byte1: Definition the same as <a href="#">CMD 0x97 "Send Byte0"</a>  Received Byte0: Duty cycle value 0 to 255
0x43	Set FAN duty	Byte1: Definition the same as <a href="#">CMD 0x97 "Send Byte0"</a> Byte2: Duty cycle value 0 to 255
0x44	Get FAN Alert RPM	Send Byte1: Definition the same as <a href="#">CMD 0x97 "Send Byte0"</a>  Received Byte0: High Byte Byte1: Low Byte
0x45	Set FAN Alert RPM	Byte1: Definition the same as <a href="#">CMD 0x97 "Send Byte0"</a> Byte2: High Byte Byte3: Low Byte  Note: <ol style="list-style-type: none"> <li>Once EC detected FAN RPM lower or equal than the settings value, EC will turn on <a href="#">FAN Alert LED</a> (if the project supported the LED).</li> <li>Default FAN Alert RPM was 0.</li> </ol>

## 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  Received Byte0: Temperature value (degree C)

0x02	Get Sensor Alert Temperature	<p>Send Byte1: Definition the same as <a href="#">Index 0x00 "Send Byte1"</a></p> <p>Received Byte0: Temperature value (degree C)</p>
0x03	Set Sensor Alert Temperature	<p>Byte1: Definition the same as <a href="#">Index 0x00 "Send Byte1"</a></p> <p>Byte2: Temperature value (degree C)</p> <p>Note:</p> <ol style="list-style-type: none"> <li>1. Once EC detected Sensor temperature higher than the settings value, EC will turn on <a href="#">Temperature Alert LED</a> (if the project supported the LED).</li> <li>2. No default Alert Temperature.</li> </ol>

## 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