OPEN Compute Project

Project Olympus Power Supply Software Interface Specification

Author:
Bryan Kelly, Principal Software Engineering Manager, Microsoft
John Siegler, Senior Hardware Engineer, Microsoft

REVISION HISTORY

Revision/ Date	Notes
V1.0	Ready for Project Olympus github

Your use of this Specification may be subject to other third party rights. THIS SPECIFICATION IS PROVIDED "AS IS." The contributors expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to the Specification. The entire risk as to implementing or otherwise using the Specification is assumed by the Specification implementer and user. IN NO EVENT WILL ANY PARTY BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

CONTRIBUTORS AND LICENSORS OF THIS SPECIFICATION MAY HAVE MENTIONED CERTAIN TECHNOLOGIES THAT ARE MERELY REFERENCED WITHIN THIS SPECIFICATION AND NOT LICENSED UNDER THE OWF CLA OR OWFa. THE FOLLOWING IS A LIST OF MERELY REFERENCED TECHNOLOGY: INTELLIGENT PLATFORM MANAGEMENT INTERFACE (IPMI); I^{2}C IS A TRADEMARK AND TECHNOLOGY OF NXP SEMICONDUCTORS ; EPYC IS A TRADEMARK AND TECHNOLOGY OF ADVANCED MICRO DEVICES INC.; ASPEED AST 2400/2500 FAMILY PROCESSORS IS A TECHNOLOGY OF ASPEED TECHNOLOGY INC.; MOLEX NANOPITCH, NANO PICOBLADE, AND MINI-FIT JR AND ASSOCIATED CONNECTORS ARE TRADEMARKS AND TECHNOLOGIES OF MOLEX LLC; WINBOND IS A TRADEMARK OF WINBOND ELECTRONICS CORPORATION; NVLINK IS A TECHNOLOGY OF NVIDIA; INTEL XEON SCALABLE PROCESSORS, INTEL QUICKASSIST TECHNOLOGY, INTEL HYPER-THREADING TECHNOLOGY, ENHANCED INTEL SPEEDSTEP TECHNOLOGY, INTEL VIRTUALIZATION TECHNOLOGY, INTEL SERVER PLATFORM SERVICES, INTEL MANAGABILITY ENGINE, AND INTEL TRUSTED EXECUTION TECHNOLOGY ARE TRADEMARKS AND TECHNOLOGIES OF INTEL CORPORATION; SITARA ARM CORTEX-A9 PROCESSOR IS A TRADEMARK AND TECHNOLOGY OF TEXAS INSTRUMENTS; GUIDE PINS FROM PENCOM; BATTERIES FROM PANASONIC. IMPLEMENTATION OF THESE TECHNOLOGIES MAY BE SUBJECT TO THEIR OWN LEGAL TERMS.

CONTENTS

1 Project Olympus Power Supply Unit (POPSU) - Overview

This specification provides the PMBUS interface requirements for the POPSUs. The following table provides the power supplies governed by this specification:

Power Supply	**Command Support Required**		
Series Name	Standard	Custom	Battery
Server POPSU without Battery	x	x	
Server POPSU with Battery	x	x	x
JBOD POPSU without Battery	x	x	

Note: JBOD POPSU does not support a battery option.

2 Project Olympus Power Supply Series - PMBUS Commands

2.1 Standard PMBUS Commands

Refer to the PMBUS SPEC Rev 1.2 (http://pmbus.org) for definition of the commands listed in the table below. Some commands have overload their behavior (indicated using the '(Overloaded)' tag in the table), where behavior is described in subsequent sections in this document.

Table 1 - Standard Pmbus commands

Command Description					Applicability	
Command Code	Command Name	PMBUS Transaction Type (Write/Read)	Data Format	#Data Bytes	PSU	PSU w/Batt
0x01	OPERATION	Write Byte/Read Byte	N/A	1	Y	Y
0x03	CLEAR_FAULTS	Send Byte/NA	N/A	0	Y	Y
0x3B	FAN_COMMAND_1	Write Word/Read Word	Linear	2	Y	Y
0x3C	FAN_COMMAND_2	Write Word/Read Word	Linear	2	Y	Y
0x46	IOUT_OC_FAULT_LIMIT	Write Word/Read Word	Linear	2	Y	Y
0x4A	IOUT_OC_WARN_LIMIT	Write Word/Read Word	Linear	2	Y	Y
0x5B	IIN_OC_FAULT_LIMIT	Read Word	Linear	2	Y	Y
0x5D	IIN_OC_WARN_LIMIT	Write Word/Read Word	Linear	2	Y	Y
0x6B	PIN_OP_WARN_LIMIT	Write Word/Read Word	Linear	2	Y	Y
0x79	STATUS_WORD	Write Word/Read Word	N/A	2	Y	Y
0x7A	STATUS_VOUT	Write Byte/Read Byte	N/A	1	Y	Y
0x7B	STATUS_IOUT	Write Byte/Read Byte	N/A	1	Y	Y
0x7C	STATUS_INPUT	Write Byte/Read Byte	N/A	1	Y	Y
0x7D	STATUS_TEMPERATURE	Write Byte/Read Byte	N/A	1	Y	Y
0x7E	STATUS_CML	Write Byte/Read Byte	N/A	1	Y	Y
0x8C	READ_IOUT	NA/Read Word	Linear	2	Y	Y

0x96	READ_POUT	NA/Read Word	Linear	2	Y	Y
0x88	READ_VIN	NA/Read Word	Linear	2	Y	Y
0x89	READ_IIN	NA/Read Word	Linear	2	Y	Y
0x8D	READ_TEMPERATURE_1	NA/Read Word	Linear	2	Y	Y
0x97	READ_PIN	NA/Read Word	Linear	2	Y	Y
0x9A	MFR_MODEL	Block Write/Block Read	ASCII	15	Y	Y
0x9E	MFR_SERIAL	Block Write/Block Read	ASCII	15	Y	Y
0x04	PHASE (Overloaded)	Write Byte/Read Byte	Linear	1	Y	Y
0x19	CAPABILITY (Overloaded)	NA/Read Byte	N/A	1	Y	Y
0x1B	SMBALERT_MASK (Overloaded)	Write Word/Block write-Block read process call	N/A	2	Y	Y
0xADh	IC_DEVICE_ID	Block Read	ASCII	20	Y	Y
0x99h	MFR_ID	Block Read	ASCII	15	Y	Y

Note: MFR_MODEL, MFR_SERIAL, IC_DEVICE_ID and MFR_ID will have the length of the ASCII string as the first byte in the data payload.

2.2 Custom PMBUS Commands

The following table describes the custom PMBUS commands in detail in this section.

Note: JBOD POPSU does not support battery, therefore these commands and functions do not need to be implemented in systems supporting JBOD POPSU specification power supply.

Table 2 - Custom Pmbus commands

Command Description					Applicability		
Cmd Code	Command Name	PMBUS Transaction Type (Write/Read)	Data Format	#Data Bytes	SERVER POPSU w/Batt	SERVER POPSU	JBOD POPSU
0x80	STATUS_MFR_SPECIFIC (BATT_ATTN_INDICATOR)	Write Word/ Read Word	N/A	1	Y	N	N
0xA0	FAULT_HISTORY	Block Read	N/A	24	Y	Y	Y
0xD2	BATT_OP_TIME_100_LOAD	NA/Read Word	Linear	2	Y	N	N
0xD3	PEAK_SHAVE_CONFIG	Write word/ Read Word	N/A	2	Y	N	N

0xD4	PEAK_SHAVE_LIMIT	Write word/ Read Word	N/A	2	Y	N	N
0xD5	BATT_POUT	NA/Read Word	Linear	2	Y	N	N
0xD6	BATT _STATUS	NA/Read Word	N/A	2	Y	N	N
0xD7	BATT_STAT_OF_CHARGE	NA/Read Word	Linear	2	Y	N	N
0xD8	BATT_HEALTH_TEST	Write Word/ Read Word	N/A	2	Y	N	N
0xD0	BATT_ARM	Write Byte/ Read Byte	N/A	1	Y	N	N
0xD1	PSU_FEED	NA/Read Byte	N/A	1	Y	N	N
0xDA	MAX_POUT	Write word/ Read Word	Linear	2	Y	N	N
0xDB	BATT_SOC_LOWER_LIMIT	Write Byte/ Read Byte	Linear	1	Y	N	N
0xDC	BATT_OC_CHARGE_LIMIT	Write Word/ Read Word	Linear	2	Y	N	N

2.3 Firmware Update Commands

The following table describes the PMBUS commands required for PSU firmware update.

Table 3 - Custom Pmbus commands

Command Description					Applicability	
Cmd Code	Command Name	PMBUS Transaction Type (Write/Read)	Data Format	#Data Bytes	PSU	PSU w/Batt
0xEF	READ_FW_INFO	NA/Block write-Block read process call	ASCII	9(read) with 1(write)	Y	Y
0xF0	ENTER PROGRAMMING MODE	Write Word	N/A	1	Y	Y
0xFB	WR_PROGRAM_MEM	Block Write	N/A	8	Y	Y
0xFC	FW_UPDATE_STATUS	Write Byte/Read Byte	N/A	1	Y	Y
0xFD	RD_PROGRAM_MEM	NA/Block write-Block	N/A	8(read) with 2(write)	Y	Y

		read process call				
0xF1	EXIT_PROGRAMMING_MODE	Write Word	N/A	1	Y	Y
0xF2	FW_DELAYS	NA/Block write-Block read process call	N/A	3(read) with 1(write)	Y	Y
0xF8	Send ROM Page	Write Byte/ Read Byte	N/A	1	Y	Y

3 COMMAND DESCRIPTION

This section describes each of the PMBUS commands. We provide detailed description of all the custom commands and provide necessary clarifications to standard PMBUS SPEC commands.

3.1 MAX_POUT (0xDA)

The PSU will record the maximum observed output power (monitoring granularity: 1 second) and report this value via this command in a 2byte linear format. Refer to the standard PMBUS command IIN_OC_FAULT_LIMIT which has a similar format. The monitoring window can be reset by the host by writing the value=0 using this command (write of any other value should return error).

Note that upon reset of the monitoring window, the PSU should overwrite the maximum power with the instantaneous power reading. Also, the maximum power reading should be persistent across PSU MCU resets (following a firmware update).

3.2 PHASE (0x04)

The PHASE command provides the ability to configure, control, and monitor the 3 phases supported by the PSU unit. The power-on default value of this command should be FFh indicating phase agnostic aggregate behavior (explained below). Monitoring of individual phase can be achieved by using the PHASE command with values from 00h to 02h.

After setting appropriate value in the PHASE command, all subsequently executed PMBUS commands should return phase specific values if applicable (refer to details below).

Table 4 – Phase select command.

Value	Feed/Phase
0xFF (Power on Default)	Applies to all 3 phases (aggregate)
0x00	Applies to Phase 1
0x01	Applies to Phase 2

0x02	Applies to Phase 3

The table above (Table 4 – Phase select command.), provides all supported values for the PHASE command. Note that, write of any other value for this PHASE command, other than the ones listed in this table, should return invalid or unsupported data PMBUS response.

The table below (Table 5 – Phase commands.) provides the expected behavior of the PMBUS commands for both aggregate phase (0xFF) and individual phase (values 0x00, 0x01, and 0x02) selection.

Table 5 – Phase commands.

Command Name	Aggregate Behavior	Individual Behavior
CLEAR_FAULTS	Clear faults on *all* phases and battery	Clear faults on *chosen* phase
IIN_OC_FAULT_LIMIT	Aggregate value across *all* phases	Value for the *chosen* phase
IIN_OC_WARN_LIMIT	Aggregate value across *all* phases	Value for the *chosen* phase
PIN_OP_WARN_LIMIT	Aggregate value across *all* phases	Value for the *chosen* phase
STATUS_WORD	Indicate fault if *any* of the phase has fault	Indicate fault if the *chosen* phase has fault
STATUS_INPUT	Indicate fault if *any* of the phase has fault	Indicate fault if the *chosen* phase has fault
READ_VIN	RMS Average value across *all* phases	RMS Value for the *chosen* phase
READ_IIN	RMS Aggregate value across *all* phases	RMS Value for the *chosen* phase
READ_PIN	Aggregate value across *all* phases	Value for the *chosen* phase
SMBALERT_MASK	Aggregate behavior across *all* phases	Behavior for the *chosen* phase
OPERATION	Aggregate behavior across *all* phases. Read will be OR value of all phases.	Behavior for the *chosen* phase

Note: Commands that are not specified in the above table are Not Applicable (N/A) – it will have the same behavior irrespective of the phase selection.

3.3 CAPABILITY (0x19)

This is a standard PMBUS command (refer to the PMBUS SPEC II) and the expected value for this command is follows, Bit 7 : 1 (Packet Error Checking is supported)

Bit 6:5 : 01 (Maximum supported bus speed is 400 Khz)

Bit 4 : 1 (ALERT# signal with expected response supported)

Bit 3 : 1 if Battery present (supported), 0 if Battery NOT present (not supported)

Bit 2:0 : 000 (Reserved)

Note: On JBOD POPSU and SERVER POPSU without battery, bit 5 will read 0.

3.4 SMBALERT_MASK (0x1B)

This command provides the ability to configure events that may trigger SMBALERT signal.

ALERT Trigger Requirements:

Each of the following events should be capable of asserting the SMBALERT. Table provides, events, trigger conditions and the expected response delay,

Table 6 – Alert commands.

Command	Bit Location	Fault Trigger Condition (default values)	Expected trigger delay
STATUS_VOUT (0x7A)	Bit7: Vout_OV_fault	Vo >13.8V	<2ms
	Bit4: Vout_UV_fault	Vo < 9.8V	<2ms
STATUS_IOUT (0X7B)	Bit7: Iout_OC_fault	Iout > max rated current (see details below)	<5-10ms
	Bit5: Iout_OC_warning	Iout > programmable current value (see details below)	<5-10ms
STATUS_INPUT (0x7C)	Bit1: IIN_OC_warning	'Input Power + battery charge power' exceeds warning threshold	<5ms

	Bit2: IIN_OC_fault	'Input Power + battery charge power' exceeds fault threshold	<5ms
	Bit3: Unit off for low output voltage Bit4: Vin_UV_fault	AC_Fail Vac < 180 Vpk (Blackout) Vac< 180 Vrms (Brownout)	<10ms (from either of these events)
STATUS_TEMPERATURE (0x7D)	Bit7: OT_fault	Default threshold value: 20 degrees below max rated temperature.	< 1s
	Bit6: OT_warning	Default threshold value: 20 degrees below max rated temperature.	< 1s
STATUS_CML (0x7E)	Bit0: Other_Logic_Fault Bit1: Other communication Fault Bit2: Reserved Bit3: Processor Fault Detected Bit4: Memory Fault Detected Bit5: Packet Error Check Failed Bit6: Invalid/Unsupported Data Bit7: Invalid/Unsupported Cmd	1.) UART Communication Error 2.) Battery Communication Error	1.) <125ms 2.) <1sec
STATUS_MFR_SPECIFIC – Battery Attention Indicator (0x80)	N/A	N/A	N/A

Note that each of events indicated via corresponding STATUS commands, when triggered, apart from sending ALERT signal, should also register a fault status. This status can be probed via the corresponding STATUS commands. Further, both the fault status bits signal MUST be sticky – it should be cleared only upon receiving an explicit CLEAR_FAULTS command (and not upon removal of the trigger event or power reset of the PSU).

The SMB_ALERT is expected to be non-sticky with a minimum assertion time of 20ms.

Finally, the value of the ALERT thresholds (current, voltage etc.,) configured via separate PMBUS commands (Iout_OC_fault, Iout_OC_warning etc.,) must be respected within +/- 1% accuracy (Note that this deviation accuracy refers to the difference between the measured alert event reading and set alert event threshold (and NOT the deviation from actual event reading). We specify the timing requirement (delay between the ALERT event trigger and assertion of the ALERT signal) in the table. In other words, we expect the ALERT signal to be sent as soon as the ALERT threshold is violated, both in time and in the threshold magnitude.

SMBALERT_MASK command will be used to determine what events raise SMBALERT and what events are masked. By default, only over current should be enabled and all other events should be masked. For all other STATUS_* commands except for over current, the default power-on value of SMBALERT_MASK command should be 0xFF (1111 1111). For STATUS_IOUT, the value should be 0x5F (0101 1111).

The factory-default value for the IOUT_OC_FAULT_LIMIT (0x46) register is specified in the hardware SPEC. This register should be made READ ONLY and any attempt to write should fail.

The factory-default value for the IOUT_OC_WARN_LIMIT (0x4A) is the same as that of IOUT_OC_FAULT_LIMIT. But this register is Read/Write and can later be modified.

There should be no other conditions that cause SMB_ALERT to assert other than overcurrent.

Note: SMB_ALERT is externally tied to a notification mechanism that results in performance degradation and throttling of the server. It is imperative SMB_ALERT does not occur for any event other than OC.

3.5 STATUS_MFR_SPECIFIC OR BATT_ATTN_INDICATOR (0X80)

The table below (Table 7 – Manufacture specific error status.) provides details of the status bits for STATUS_MFR_SPECIFIC PMBUS command.

Table 7 – Manufacture specific error status.

Function	Bit position	Value: 1/0	Conditions
Over temperature (OT)	Bit 7	1	Battery over-temperature is detected *In charging Cell temperature > 53 degree C or Cell temperature < 3 degree C *Not charging Cell temperature > 50 degree C or Cell temperature < 5 degree C Or discharge and charge converter OT is detected

		0	Battery over-temperature is not detected
State of charge critical lower threshold (SOC_LT)	Bit 6	1	State of Charge threshold breach Flag. Active when set. Indicates battery state of charge is below SOC_LT threshold (say <25%)
		0	State of charge is higher than critical (SOC_LT) threshold
Under voltage protection (UVP)	Bit 5	1	Status bit that indicates one or more cells is below the cell minimum operating voltage threshold Or the discharge and charge converter output is < out of limit UVP
		0	Under voltage is not detected
Over voltage charge protection (OV_CHG)	Bit 4	1	Status bit that indicates one or more cells is above the cell voltage threshold that indicates an overcharge condition. Charging is prohibited. Active when set.
		0	No over voltage charge is detected
Over current charge protection (OC_CHG)	Bit 3	1	Indicates Overcurrent in charge condition detected. Active when set. Charge is prohibited until flag is cleared.
		0	Battery charge current is normal
Over current discharge protection (OC_DSCHG)	Bit 2	1	Indicates Overcurrent in discharge condition detected. Active when set. Discharge is prohibited until flag is cleared. Or the discharge and charge converter output is > out of limit OCP
		0	Battery discharge current is normal
Cell Imbalance/Failure (CF)	Bit 1	1	Status bit that indicates Cell imbalance or Weak Cell detected. Active when set.
		0	No cell imbalance or weak cell detected.
Permanent Failure (PF)	Bit 0	1	Indicates Permanent hardware failure detected, such as unrecoverable communication errors internal to the battery, open thermistor, etc., Cell permanent fail condition detected, Momentary cell temperature or cell voltage reached permanent fail limits. Active when set. Charge and discharge is prohibited until flag is cleared.
		0	No permanent failures

3.6 Battery Status (0xD6)

On PSU where battery is supported the following table (Table 8 – Battery Status)provides status bits.

Table 8 – Battery Status

Function	Bit position	Value: 1/0	Conditions

Reserve	Bit 15	1	
		0	
Reserve	Bit 14	1	
		0	
Reserve	Bit 13	1	
		0	
Reserve	Bit 12	1	
		0	
Reserve	Bit 11	1	
		0	
Reserve	Bit 10	1	
		0	
Reserve	Bit 9	1	
		0	
Reserve	Bit 8	1	
		0	
BAL_ACTIVE	Bit 7	1	Status bit indicating that cells are being balanced. Active when set.
		0	
EP_PRESENT	Bit 6	1	Status bit that indicates battery management system is running on external power (+12Vsb). This represents normal operating condition. Active when set.
		0	
PCHG_EN	Bit 5	1	Status bit that indicates that battery cell voltage conditions are below normal charging limits, but above Permanent Fail limits. Active when set.
		0	
DSCHG_EN	Bit 4		Status bit that indicates battery conditions are normal, and things like SOC, cell voltage, balance, etc. are suitable for discharge. Active when set.
CHG_EN	Bit 3	1	Status bit that indicates battery conditions are normal, and battery is capable of accepting charge. Active when set.
		0	
Charging	Bit 2	1	Status bit indicating that charge current is detected. Active when set.
		0	Not Charging
Discharging	Bit 1	1	Status bit indicating that discharge current is detected. Active when set.
		0	Not discharging
Initialized	Bit 0	1	Assembled in battery manufacturer
		0	EEPROM Error is detected

3.7 BATT HEALTH TEST (0xD8)

This command is a Write Word command which can force battery to enter backup mode for 2 seconds.

After test, user can read BATT_HEALTH_STATUS (0xD6) and BATT_FAULT_INDICATOR (0x80) to check if Battery can function properly.

Table 9 – Battery Health Test (0xD8)

#Bytes	1	7	1	1	8	1	8	1	8	1	8	1	1
Content	S	Slave Address	W	A	BATT_HEALTH_TEST (cmd code: 0xD8)	A	Password Low Byte (0xCD)	A	Password High Byte (0xAB)	A	PEC (Optional)	A	P

Below is the suggested test sequence:

1. User send 0xD8 (Battery Health Test) PMBus command with password (0xABCD) to start Battery Health Test.

2. During or after the test, user can monitor the result by reading the 0xD6 (Battery Health Status) and 0x80 (Battery Fault Indicator) through the PMBus Command.

If this command require the battery to be discharged, it should be restricted to less than 2 seconds of discharge.

NOTE: Before battery health test finished, another battery health test command will be ignored. Further, both battery peak shaving (see below) and battery health test can coincide if possible, otherwise, clear failure response must be provided back to the user.

3.8 BATTERY POWER OUTPUT (BATT_POUT, 0xD5)

This command returns the battery power output in Watts. This command follows the same request/response format as the standard PMBUS command, READ_POUT (0x96) specified in the PMBUS SPEC.

3.9 BATTERY STATE OF CHARGE (BATT_STAT_OF_CHARGE, 0xD7)

This command returns the battery charge level and is presented as a percentage between 0 to 100%. This command follows the same request/response format as the standard PMBUS command, READ_POUT (0x96) specified in the PMBUS SPEC.

3.10 BATTERY OPERATION TIME AT 100% LOAD (BATT_OP_TIME_100_LOAD, 0xD2)

This command returns the number of seconds the battery can deliver power at 100% (full load) of its power output capacity, starting from 100% or full capacity.

This command follows the same request/response format as the standard PMBUS command, READ_POUT (0x96) specified in the PMBUS SPEC.

3.11 BATTERY CHARGING OVER CURRENT LIMIT (BATT_OC_CHARGE_LIMIT, 0xDC)

This commands sets the over current limit in Amperes associated with battery charge current. This command follows the same request/response format as the standard PMBUS command, IOUT_OC_FAULT_LIMIT Read/Write Word specified in the PMBUS SPEC. The set limit value will be used for triggering the ALERT# signal on the power device.

3.12 BATTERY STATE OF CHARGE CRITICAL LOWER LIMIT (BATT_SOC_LOWER_LIMIT, 0xDB)

This commands sets the lower limit for the battery state of charge in percentage (between 0 and 100%) The set limit value will be used for triggering the ALERT# signal on the power device.

3.13 PEAK SHAVING CONFIGURATION (PEAK_SHAVE_CONFIG, 0xD3)

This command is a Read/Write Word command which get/set configuration settings associated with peak shaving using battery energy. This command will be used in conjunction with the command to set the power limit (see 0xD4 below).

Configuration parameters for peak shaving:

- Battery lower threshold in percentage (0 to 100) which specifies the threshold beyond which battery will stop performing peak shaving. Default value for this is 50.
- Battery higher threshold in percentage (0 to 100) which specifies the threshold beyond which battery will start performing peak shaving. When this value is 100, it signifies that peak shaving is disabled. Default value is 100 (disabled).

Table 10 – Write Peak Shaving Configuration (0xD3)

1	7	1	1	8	1	8	1	8	1	8	1	1
S	Slave Address	W	A	Cmd code 0xD3	A	Battery Lower Threshold	A	Battery Higher Threshold	A	PEC (Optional)	A	P

Table 11 – Read Peak Shaving Configuration (0xD3)

1	7	1	1	8	1	1	7	1	1	8	1	8	1	8	1	1
S	Slave Address	W	A	Cmd code: 0xD3	A	Sr	Slave Address	R	A	Battery Lower Threshold	A	Battery Higher Threshold	A	PEC (Optional)	N	P

3.14 PEAK SHAVING LIMIT (PEAK_SHAVE_LIMIT, 0xD4)

This command is a Read/Write Word command with "Linear" data format which gets/sets the power limit in watts. The PSU/battery module will try to adhere to this power limit in an opportunistic fashion (refer to the battery peak shaving configuration command 0xD3 above) by using battery energy. This command follows the same request/response format as the standard PMBUS command, IOUT_OC_FAULT_LIMIT (0x46) specified in the PMBUS SPEC.

When the set power limit is about to be breached, the battery will "supplement" power to ensure that the power limit is always enforced. It is important to note that the PSU will still provide power up to the set power limit and the battery will supplement the rest. For instance, if the set power limit is 120W and the actual power draw is 135W, the PSU will provide 120W and the battery will provide the remaining 15W. This peak shaving operation is further determined by the lower/higher configuration battery charge thresholds as specified in the Peak_shave_config (0xD3) command above.

Table 12 – Write Peak Shaving Limit (0xD4)

Write Peak Shaving Limit (0xD4)												
1	7	1	1	8	1	8	1	8	1	8	1	1
S	Slave Address	W	A	Cmd code 0xD4	A	Power Limit Byte 1	A	Power Limit Byte 2	A	PEC (Optional)	A	P

Table 13 – Read Peak Shaving Limit (0xD4)

Read Peak Shaving Limit (0xD4)																
1	7	1	1	8	1	1	7	1	1	8	1	8	1	8	1	1
S	Slave Addres s	W	A	Cmd code: 0xD4	A	S r	Slave Address	R	A	Power Limit Byte 1	A	Power Limit Byte 2	A	PEC (Optional)	N	P

3.15 BATT_ARM (0xD0)

This command is a Read/Write Byte command which can enables/disables PSU to enter battery backup mode when AC is loss. Default setting is battery backup mode enabled. This command's data content will be reset to default value after PSU shutdown.

Table 14 – Write BATT_ARM (0xD0)

1	7	1	1	8	1	8	1	8	1	1
S	Slave Address	W	A	Cmd code 0xD0	A	BATTERY ARM State No Op (0xAA) Enable (0xBB) Disable (0xCC)	A	PEC (Optional)	A	P

Battery ARM state:

No Op (0xAA): Factory-default and power-on-default value and indicates default PSU/Battery operation. In this mode, the PSU will start to source from battery when any of the following events happen,

- If AC is lost,
- When the PSU is removed from the chassis (PS_KILL goes from low to high),

Enable (0xBB): When AC/utility is available and active, this command will forcefully initiate battery sourcing. This command is ignored when battery is already sourcing.

- When battery capacity becomes empty and utility power is available, utility sourcing should resume and the value should revert to 0xAA.
- When battery capacity becomes empty and utility power is not available, the PSU will shut off output. In this event, upon subsequent power on (when utility resumes), the value should go back to 0xAA.

Disable (0xCC):

When Battery sourcing is active, this command will

(i) When AC/utility is available, initiate AC/utility sourcing (and revert value to 0xAA)
(ii) When AC/utility is unavailable, shutdown PSU output and all microcontrollers (and revert value to 0xAA upon subsequent PSU power on).

When utility/AC sourcing is active, this command will not cause any immediate effect but the value is stored and later used when AC is lost.

NOTE: IT IS IMPERATIVE THAT THE BATT_ARM MODE GOES BACK TO 0xAA WHENEVER THE PSU/MICROCONTROLLER IS RESET.

NOTE: PSU must meet ITIC requirement whenever switching between AC and battery.

Table 15 – Read BATT_ARM (0xD0)

1	7	1	1	8	1	1	7	1	1	8	1	8	1	1
S	Slave Address	W	A	Cmd code: 0xD0	A	Sr	Slave Address	R	A	BATTERY ARM State	A	PEC (Optional)	N	P

3.16 PSU_FEED (0xD1)

This command is a Read Byte command which returns the active feed of the PSU.

Table 16 – Read PSU_FEED (0xD1)

S	Slave Address	W	A	Cmd code: 0xD1	A	Sr	Slave Address	R	A	PSU ACTIVE FEED	A	PEC (Optional)	N	P

										0xAA: Primary Feed 0xBB: Backup Feed 0xCC: No Feed (in battery mode)				

3.17 FAN_COMMAND_1/2 (0x3B, 0x3C)

These are standard PMBUS commands used to control the fans in the power supply unit. To set the fans to 100% duty cycle, set the data bytes to 0x64 0x00 (Data Byte Low, Data Byte High). To set the fans to 30% duty cycle, set the data bytes to 0x1E 0x00 (Data Byte Low, Data Byte High).

3.18 FAULT_HISTORY (0xA0)

This read-only command is used to retrieve the failure history log. The failure history event will be stored into non-volatile memory (external flash) if any sub module encounters a fault that causes its output to shut off. The fault history is recorded immediately prior to the 12V output going low. If there is already a log entry present, another fault should not overwrite the existing log entry.

The following conditions should trigger the log history to be recorded:

- Input UVP
- Input OVP
- Output UVP
- Output OVP
- Input OCP
- Output OCP
- Any OTP
- Fan fault
- Watchdog reset
- Both IVS Feed A/B Error

The CLEAR_FAULTS command is used clear the log history. If there are no faults, the hour and minute bytes (bytes 1-3) should return zero. That will indicate to the client that the log is clear.

The following table shows the data returned.

Table 17 – FAULT_HISTORY (0xA0)

Fault History			
Byte number	**Byte order**	**Description**	**Data Format**
0		Phase (single byte)	N/A

		Bit 0: Phase 1 Bit 1: Phase 2 Bit 2: Phase 3 Bit 3: BBU (Battery Backup Unit) The corresponding bit where the fault occurred should be set. If all phases had the fault, the value in this byte should be 0x7.	
1		Minute used when fault occurred (single byte)	N/A
2	Low byte	Hour used when fault occurred	N/A
3	High byte		
4	Low byte	Fault status word (same as 0x79h)	N/A
5	High byte		
6	Low byte	Output voltage when fault occurred	Linear. Note: VOUT_MODE is fixed to 0x1A for converting this field.
7	High byte		
8	Low byte	Output Current when fault occurred	Linear
9	High byte		
10	Low byte	Output Power when fault occurred	Linear
11	High byte		
12	Low byte	Input Voltage when fault occurred	Linear
13	High byte		
14	Low byte	Input Current when fault occurred	Linear
15	High byte		
16	Low byte	Input Power when fault occurred	Linear
17	High byte		
18	Low byte	Ambient Temperature	Linear
19	High byte		

20	Low byte	Fan 1 speed	Linear
21	High byte		
22	Low byte	Fan 2 speed	Linear
23	High byte		
24		Bit map for reset flag and IVS Feed A/B	Linear
25		BATT_ATTN_INDICATOR (0x80)	N/A
26	Low byte	Battery Status (0xD6)	N/A
27	High byte		

Bit map for reset flag and IVS FeedA/B	Description
0	BOR: DSP brown out reset flag
1	WDTO: watchdog time out flag
2	SWR: software reset flag
3	EXTR: external reset flag
4	IOPUWR: illegal opcode reset flag
5	RESERVE
6	Feed A lost
7	Feed B lost

4 Firmware Update Description

This section describes each of the Firmware Update commands.

4.1 Firmware Update Commands and Usage

In this section, we provide the instructions and requirement associated with updating the firmware in the power supply unit.

4.2 Firmware Image Description

The firmware image is logically divided into two regions, bootloader image region and the application image region. The application region is in turn divided in to two regions – region A and region B – which will be used for storing two version of the application image. Note that only one version of the application, either A or B will be active at any time. These three regions, Bootloader, Application Image A and Application Image B of the firmware image along with their recommended size are specified in the table below. Note that the specified size is only a recommendation and we expect provisioning enough storage space (FLASH size) to accommodate future PMBUS command and functional additions.

Each of the three regions will have a corresponding *Intel HEX* formatted image file.

Table 18 – Firmware Regions

Area	Size
Bootloader Image	16Kbytes
Application Image A	32Kbytes
Application Image B	32Kbytes

Table: Firmware regions with recommended size

4.3 Firmware Upgrade Procedure:

1. Firmware update will be performed to one of the 3 regions (Boot loader, Application image A and Application Image B) of the firmware image at any time, using the corresponding region's hex file.
2. Host enters firmware update mode to the chosen region by specific PMBUS Command (enter_programming_mode 0xF0) with specific region. Power device will erase the specified region as part of the enter_programming_mode command.
3. Host reads the *Intel Hex* formatted image file in sequence and perform action based on the record type,
    a. Record type 0x00: Write the firmware image in the specified address location in the HEX file
    b. Record type 0x04: Set the appropriate Page number specified in the HEX file
    c. Record type 0x01: Exit programming.
4. Status code is provided to indicate status/error during firmware update process.
5. When there is any interruption during programming (e.g. power reset), host can restart procedure by exiting then entering the programming mode again.
6. Host verifies integrity of the written image by reading back image data and compare it with HEX file data.
7. After verification, program can exit firmware update mode by specific PMBUS command (0xF1).
8. When firmware update mode is exited, the user can specify the following actions:
    a. Jump to bootloader:

- Power device should jump to new bootloader and run the active application image

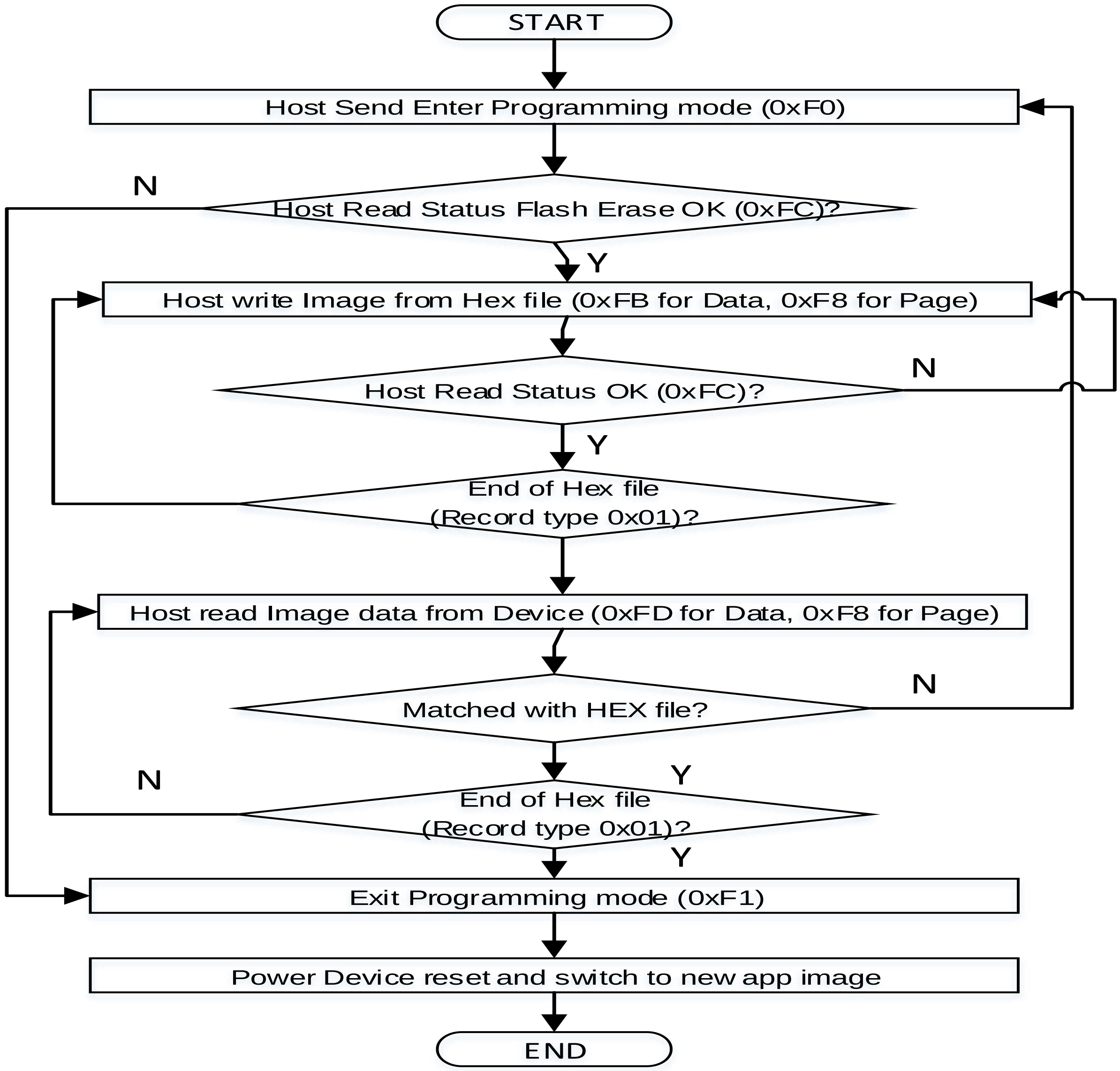
b. Jump to application image:

- Power device should switch to the specified application image
- Once the specified firmware image is loaded correctly, update active application image (refer to READ_FW_INFO command). Any error during application load should revert to the original application image region.

4.3.1 Firmware Upgrade Flowchart

The firmware update follow the workflow outlined in Figure 1 - Firmware Update

Figure 1 - Firmware Update workflow


START
Host Send Enter Programming mode (0xF0)
N
Host Read Status Flash Erase OK (0xFC)?
Y
Host write Image from Hex file (0xFB for Data, 0xF8 for Page)
N
Host Read Status OK (0xFC)?
Y
End of Hex file
(Record type 0x01)?
Host read Image data from Device (0xFD for Data, 0xF8 for Page)
N
Matched with HEX file?
N
Y
End of Hex file
(Record type 0x01)?
Y
Exit Programming mode (0xF1)
Power Device reset and switch to new app image
END

4.4 Firmware Update Command Description

4.4.1.1 *Enter Programming Mode (0xF0)*

This command is used to enter into the firmware upgrade mode from application mode. The programming mode indicates which region should be erased. The power device shall return failure when the host attempts to program (ENTER_PROGRAMMING) the *active* application region.

Table 19 – Enter Programming Mode (0xF0)

1	7	1	1	8	1	8	1	8	1	1
S	Slave Address	W	A	Enter Programming Mode Command Code (0xF0)	A	Mode	A	PEC (Optional)	A	P

Name	Length	Description	Remark
Mode	1 byte	Programming Mode	0 – Bootloader 1 – Application Image A 2 – Application Image B 5 – No erase. Used by host to read program memory
PEC	1 byte	Packet Error Check	CRC8 code is calculated per SMBus Specification.

4.4.1.2 *Send ROM Page (0xF8)*

This command is used to change Flash memory page setting when host is reading/writing program memory. When used in READ mode, the command should return the current page in the flash memory.

Example:

Host need to extract 1 byte highlighted in green from HEX file and send as below format,

:020000040001F9

Table 20 – Send ROM Page (0xF8)

1	7	1	1	8	1	1	7	1	8	1	1
S	Slave Address	W	A	Send ROM Page (0xF8)	A	Sr	Rom Page (0x01)	A	Checksum	A	P

Name	Length	Description	Remark
Rom Page	1 byte	Flash memory page setting	Set target flash memory page in program memory.
Checksum	1 byte	Checksum	Arithmetic sum of all bytes sent, including slave address and its R/W bit. Note that this field is mandatory.

4.4.1.3 Write Program Memory (Write ROM data) (0xFB)

This command is used to transfer data from HEX file to the power device, and write into the program memory. Please refer to Table 21 – HEX File Data Format for more detailed information.

Table 21 – HEX File Data Format

Name	Length	Description	Remark
Data Length	1 byte	Number of bytes in the data field.	16 (0x10) bytes of data are the usual values. Directly get from HEX file
Address offset	2 bytes	16-bit program memory address	Directly get from HEX file
Record Type	1 byte	Defining the type of the data field	Only below record types are supported: 00: Data Record 01: End of File Record 04: Extended Linear Address Record
Data Row	16 bytes	Each data row consists 16 bytes.	
Hex Checksum	1 byte	Checksum of HEX file	This checksum is calculated using all above bytes.

Host shall skip the data length byte, HEX Checksum and record type byte, and then follow below format to write flash

Table 22 – Host Sends Byte 1~8 to power device

1	7	1	1	8	1	8	1	8	1	8	1		8	1	8	1	1

S	Slave Address	W	A	Write program memory Command Code (0xFB)	A	Address mid byte (from HEX file)	A	Address low byte (from HEX file)	A	Data Byte 1 (from HEX file)	A	…	Data byte 8 (from HEX file)	A	Check Sum (Optional)	A	P

Table 23 – Host Sends Byte 9~16 to power device

1	7	1	1	8	1	8	1	8	1	8	1		8	1	8	1	1
S	Slave Address	W	A	Write program memory Command Code (0xFB)	A	Address mid byte (from HEX file)	A	Address low byte| 0x08 (from HEX file)	A	Data Byte 9 (from HEX file)	A	…	Data byte 16 (from HEX file)	A	Check Sum (Optional)	A	P

Below table (Table 24 – Program Memory (write ROM data) Command) provides the Program Memory (write ROM data) Command,

Table 24 – Program Memory (write ROM data) Command

Name	Length	Description	Remark
Address Offset	2 bytes	16-bit program memory address	Directly get from HEX file, should add 0x08 to address low byte if send byte 9~16
Data	8 bytes	8 bytes of the data	
Checksum	1 byte	Calculated by host	Arithmetic sum of all bytes received and sent, including slave address and its R/W bit. Note that this field is mandatory.

4.4.1.4 Read Program Memory (Send Target Address) (0xFD)

It is used set target program memory address (middle byte and low byte) for reading. High byte is set using Send ROM Page (0xF8) command.

To write target program memory address for reading, refer to table: Table 25 – write target program memory address for reading

Table 25 – write target program memory address for reading

1	7	1	1	8	1	8	1	8	1	8	1	8	1

S	Slave Address	W	A	Read Program Memory Command Code (0xFD)	A	Byte count (2)	A	Address Mid Byte	A	Address Low Byte	A

The power device will in turn return the data bytes in the following format. Host need to verify these data bytes with those in HEX file. Refer to Table 26 – Host reads byte 1~8 from power device

Table 26 – Host reads byte 1~8 from power device

1	7	1	1	8	1	8	1		1	8	1	8	1	1
Sr	Slave Address	R	A	Byte count (8)	A	Data Byte 1	A	…	A	Data byte 8	A	Check Sum (Optional)	N	P

Note that bytes 9 to 16 will be queried in a similar fashion with incremented address.

4.4.1.5 *Firmware Update Status (0xFC)*

READ STATUS: This command is used to get the firmware update status. This command will be called by the user to determine the status of the previously executed firmware update command using the 'status byte' (see status table below for all possible values) and

WRITE or CLEAR STATUS: The status code can be cleared by writing a value of '0' to this command. A write value other than '0' should return failure. After clearing status, subsequent firmware status updates should be reflected in the read response.

Table 27 – Firmware Update Status Command

1	7	1	1	8	1	1	7	1	1	8	1	8	1	1
S	Slave Address	W	A	FW update status command code (0xFC)	A	Sr	Slave Address	R	A	Status Byte	A	Checksum	N	P

Table 28 – Firmware Update Status Command Description

Name	Length	Description	Remark
Status	1 byte	Firmware update status	Please refer to FIRMWARE UPGRADE STATUS CODE DEFINITION table (below) for status code definition

Checksum (Optional)	1 byte	Checksum	Arithmetic sum of all bytes received and sent, including slave address and its R/W bit. Note that the host program should check this checksum field to ensure the data integrity.

Table 29 – Status for Firmware Update

Status Code	Name	Description
0x00	No Error	No error on last received
0x01	Checksum error	Received checksum has error
0x02	Block number error	Block number in packet has error
0x03	Length error	Packet length error
0x04	I2C write error	Tried to write a read-only command
0x05	I2C read error	Tried to read a write-only command
0x06	Command error	Received a unsupported command
0x0B	Model ID Error	Received invalid firmware image
0x0C	Flash erase error	Program memory cannot be erased all to 0xFF status
0x0D	Flash erase OK	Program memory has been erased and verified
0x54	Error active region update attempt	Error attempting to update active application region
0x55	Erase In progress	Status when PSU is still erasing its external flash memory
0x56	Flash Write in Progress	Error when host sending 0xFB command too fast
0x57	Write Address Out of Bound	Error attempting to write program memory exceed expected range
0x58	Write Address not in sequence	Write address received by PSU is not in sequence

Table: FIRMWARE UPGRADE STATUS CODE DEFINITION

Note: OEM specific errors can be encoded using status codes from 0xF0 to 0xFF. Details pertaining to these codes will be obtained from OEM data sheet.

4.4.1.6 Exit Programming Mode (0xF1)

This command is used to jump to one of the three regions (typically following ENTER PROGRAMMING and WRITE PROGRAM MEMORY command sequence) which may involve reset to the power device.

Upon executing this command, the power device will attempt to jump to the specified region (load the new firmware image). The power device shall correspondingly update the “Active image” as part of the ‘READ_FW_INFO (0xEF) command if the load was successful.

A jump to the bootloader region shall result in (re) loading the active application image.

Any failure while loading the new application image shall result in the power device automatically reverting to the original application image (the active image before the firmware update).

Note that EXIT_PROGRAMMING_MODE command may also be executed in isolation (without corresponding ENTER PROGRAMMING and WRITE PROGRAM MEMORY command sequence) to jump to different region of firmware.

Table 30 –Exit Program Mode Command

1	7	1	1	8	1	8	1	8	1	1
S	Slave Address	W	A	Exit Program Memory Command Code (0xF1)	A	Mode	A	Checksum	A	P

Table 31 – Exit Program Mode Command Description

Name	Length	Description	Remark
Mode	1 byte	Exit Mode	0 – Jump to Bootloader 1 – Jump to Application Image A 2 – Jump to Application Image B 3 – Exit Programming Mode Only. Do not reset firmware
Checksum	1 byte	Checksum	Arithmetic sum of all bytes sent, including slave address and its R/W bit. Note that this field is mandatory.

4.4.1.7 READ_FW_INFO (0xEF)

This command will be used by the host to query details regarding the firmware image and the host will provide the queried image name (Image A (0xA) or Image B (0xB) or BootLoader (0x0)) as input to this command,

Table 32 – Read_FW_INFO(0xEF)

1	7	1	1	8	1	8	1	8	1
S	Slave Address	W	A	Read Firmware Revision Command Code (0xEF)	A	Byte count (1)	A	Image Queried (0xA or 0xB or 0x0)	A

The power device upon receiving the above packet will respond with the following details

- 1 Active Byte (0x1 indicate active image, 0x0 indicates inactive image)
  - When bootloader image is active, it indicates that the application image is being loaded (and neither A nor B will be active at this time)
- 8 Revision Bytes represent the firmware revision number string in ASCII format

Table 32 – Read_FW_INFO(0xEF)

1	7	1	1	8	1	8	1	8	1	8	1	8	1	1
Sr	Slave Address	R	A	Byte count (9)	A	Active Byte (0x0 or 0x1)	A	…	A	Revision bytes	A	Check Sum (Optional)	N	P

4.4.1.8 FW_ DELAYS (0xF2)

This command obtains the delay or wait time required between subsequent firmware update commands. This command takes 1 byte delay category argument (see below) and return 3 bytes of delay value.

Input (Delay categories):

- BaseDelay (0x0)
- Enter Programming or Erase (0x1)
- DataRead (0x2)
- DataWrite (0x3)
- SetPage (0x4)
- Exit Programming or Reset (0x5)

Table 33 – FW_DELAYS(0xF2)

1	7	1	1	8	1	8	1	8	1
S	Slave Address	W	A	Get firmware delays Command Code (0xF2)	A	Byte count (1)	A	Category Queried (0x0, …, 0x5)	A

For the input argument, BaseDelay (0x0), the power device will return the scaling factor in microseconds which will be used for the rest of the categories. This has to be 100 microseconds as default. For other inputs, the delay will be present with respect to this scaling factor. For instance, a value of 1000 for Erase (0x1) will indicate a delay of 100 milliseconds.

Table 34 – FW_DELAYS(0xF2)

1	7	1	1	8	1	8	1	8	1	8	1	1
Sr	Slave Address	R	A	Byte count (3)	A	Delay High Byte	A	Delay Mid Byte	A	Delay Low Byte	A	P

4.4.2 Additional Firmware Update Details/Requirements

4.4.2.1 FIRMWARE UPDATE CONFORMANCE REQUIREMENTS

- The CRC/checksum of the firmware images should be included as part of the firmware HEX file and the power device shall verify the CRC/checksum after write operation is completed. Mismatched CRC shall result in error – after the write of the last entry in the HEX file, firmware STATUS code will indicate checksum error. Further, subsequent EXIT attempt to the checksum-mismatched image shall fail.
- The power device shall verify if the provided HEX image matches with the model ID of the device. The power device shall return Invalid Model ID error upon receiving a mismatched firmware image.
- PSU firmware update should not cause any interruptions to power output of the PSU
- PSU serial number should not be cleared as part of firmware update
- Firmware update or PSU microcontroller reset should not cause any SMB_ALERT. The SMB_ALERT should only assert on OC event by default.
- During firmware upgrade, the PSU should respond to PMBUS commands and cannot be unresponsive for more than **5 seconds**. In particular, the STATUS commands (STATUS_* commands and BATTERY STATUS) cannot be unresponsive for more than 5 seconds.

4.4.2.2 HEX FILE NAMING CONVENTION

< part#>_ <Model Name>_<Region Name>_CS<Check Sum>_V<Version>.hex

- <Part#> is manufacturer part number for the FW HEX file.
  - Note that this manufacturer part number field of the hex file name will be compared against the PSU’s manufacturer part number (via IC_DEVICE_ID PMBUS command) for sanity checking before attempting to update the firmware.
- <Model Name> should uniquely correspond to the string returned by MFR_MODEL PMBUS command.
- <Region Name> should refer to one of the following image regions
  - RegionA

  - RegionB
  - Bootloader
- <Checksum> is the checksum value of the HEX file.
- <Version> should the value returned from Read_FW_INFO command (post the update)

- For example, a PSU firmware image with name

  "630002797 _PL1600_RegionB_CSD732_V010500.hex"

  corresponds to,

  - Part#: 630002797
  - Model Name: PL1600
  - Region Name: RegionB
  - Checksum: 0xD732
  - Firmware version: 010500

4.4.2.3 *CHECKSUM CALCULATION / VERIFICATION EXAMPLE*

Checksum Calculation example:

Use Write Program Memory Command as an example, assume all program memory data are 0xAA:

Checksum is the arithmetic sum of all bytes sent, including slave address and its R/W bit.

Table 35 – HEX File Checksum Example

1	7	1	1	8	1	8	1	8	1	8	1		8	1	8	1	1
S	Slave Address (0xB0, 8 bit included R/W bit)	W	A	Write program memory Command Code (0xFB)	A	Address mid byte (from HEX file) (Data: 0x18)	A	Address low byte (from HEX file) (Data: 0x00)	A	Data Byte 1 (from HEX file) (Data: 0xAA)	A	…	Data byte 8 (from HEX file) (Data: 0xAA)	A	Check Sum (Mandatory) (0xED)	A	P

Checksum = 2's complement [lowest byte (0xB0 + 0xFB + 0x18 + 0x00 + (0xAA)x8)] = 0xED

Checksum Verification example:

Use Read Program Memory Command as an example, assume all program memory data are 0xBB:

Table 36 – Checksum Example

1	7	1	1	8	1	1	7	1	1	8	1		1	8	1	8	1	1
S	Slave Address (0xB0, 8 bit included R/W bit)	W	A	Read program memory Command Code (0xFD)	A	Sr	Slave Address (0xB1, 8 bit included R/W bit)	R	A	Data Byte 1 (0xBB)	A	…	A	Data byte 8 (0xBB)	A	Check Sum (Mandatory) (0xCA)	N	P

Check if arithmetic sum of all above bytes is zero.

Lowest byte (0xB0+0xFD+0xB1+0xBB x 8+ 0xCA) = 0;

4.4.2.4 OTHER FIRMWARE UPDATE REQUIREMENTS

- The power device shall return failure when the host attempts to program (ENTER_PROGRAMMING) the active application region.
- The power device shall respond to all firmware update commands when running in bootloader mode.
- The provisioning of flash capacity (including any contiguity assumptions) must take in to account future firmware additions. We recommend provisioning flash capacity to 2X the size of the estimated image size.
- User should not start to upgrade program when AC is lost and battery is sourcing power.
- The power device should continue to service other PMBUS command when the firmware upgrade is in progress. Note that PMBUS commands submitted during the EXIT PROGRAMMING or RESET phase of the firmware upgrade should FAIL gracefully without compromising the firmware update procedure.
- A power loss during update should result in an atomic firmware update transaction – the firmware update should either completely succeed with active image changed to the newly written image or should completely fail, reverting back to the previously active image.
- Firmware update should not cause any interruptions to power delivery and other power device functional logic
- Device Serial number should not be cleared as part of firmware update

5 ADDITIONAL REQUIREMENTS:

5.1 I2C

- The I2C address of the PSU shall be 0xB0 (which is also the default address) when address is LOW and shall be 0xB2 when Address is pulled High.
- 400Khz I2C interface speed

5.2 EXPECTED COMMAND ACCURACY AND UPDATE FREQUENCY

Table 37 – Accuracy and Update Frequency

Command	**Measurement Accuracy**	**Update Frequency**
READ_IOUT	+/- 2.5%	100ms
READ_POUT	+/- 2.5%	100ms
READ_PIN	+/- 2.5%	100ms
BATT_POUT	+/- 2.5%	100ms
BATT_STAT_OF_CHARGE	+/- 2.5%	100ms

The above table indicates the accuracy between the measured value (by power device) and actual value.

5.3 BATTERY OUTPUT BEHAVIOR

We expect the PSU to deliver the required voltage (12V or 48V) as long as there is AC input or when the batteries have energy to support the voltage output. We do not expect the PSU to turn OFF battery output or alter output voltage autonomously. The PSU will receive an explicit PMBUS command (refer to BATT_ARM and OPERATION PMBUS command) to shut off power output.

5.4 PSU FAULT STATUS AND SMB_ALERT BEHAVIOR

- Fault status reported by the PSU (via commands such as STATUS_WORD, STATUS_VOUT, STATUS_IOUT, STATUS_INPUT, STATUS_TEMPERATURE) should be sticky or remain un-cleared until an explicit CLEAR_FAULTS command is received.
    - Note that a PSU AC power cycle or PS_ON toggle should NOT clear the fault event.
    - Note that the SMB_ALERT signal should be sticky as long as PSU fault status is in un-cleared state
- SMB_ALERT should be triggered within 1-5ms of over current detection and within 10ms of other alert events such as under voltage, AC loss etc. (When enabled for these events).

- SMB_ALERT should not be triggered for any reason other than **over current**. All other alert triggers should be disabled by default. Note: No condition other than over current should cause SMB_ALERT to assert.
- The SMB_ALERT should be non-sticky, unlike the log bits the SMB_ALERT should only assert when the alerting condition (over current) is currently active. When SMB_ALERT is asserted it should assert for a minimum duration of 20ms, and deassert when the condition subsides thereafter.
- SMB_ALERT should be triggered within +/- 1% deviation from the ALERT threshold. Note that this deviation accuracy refers to the difference between the measured alert event reading and set alert event threshold (and NOT the deviation from actual event reading).
- The IOUT_OC_FAULT_LIMIT and IOUT_OC_WARN_LIMIT register should be factory initialized to the values specified in the SMB_ALERT section.

5.5 Battery Sourcing

When AC power is lost the PSU will source from battery. The battery should run until fully discharged, or the PSU receives the BATT_ARM with 0xCC. See Section: 3.15 BATT_ARM (0xD0). Upon receipt of this command the PSU must attempt to source from AC, losing power if no AC is present.

Note: No artificial timelines or shutdown conditions must be placed upon the battery. The battery is expected to source until depletion.

Information: Upon AC loss the PSU will source from battery, a server motherboard based microprocessor that is external too and fully independent of the PSU will ready the server for shutdown by flushing necessary caches. Once complete the server side microprocessor will send BATT_ARM 0xCC to the PSU followed by BATT_ARM 0xAA.

Note: The system cache flush may take up to 500 seconds, however the server power is typically below 200W during this time.

5.6 Miscellaneous

- All SET PMBUS commands must be persistent across PSU resets (Note that
- Unless otherwise specified all STATIC battery/PSU parameters populated in the firmware as part of the supported PMBUS commands should indicate the value at the end of 4 years of operation at 30C. For instance, BATT_OP_TIME_100_LOAD is one example of a PMBUS command with static value.
- The peak shaving (using battery energy) operation must be disabled when battery_health_test command is in progress.