PMBus™
Power System Management Protocol Specification
Part II – Command Language

Revision 1.0

28 March 2005

www.powerSIG.org

Filename: PMBus_Specification_Part_II_Rev_1_0_20050328.doc
Last saved: 28 Mar 2005, 15:40

DISCLAIMER

This specification is provided "as is" with no warranties whatsoever, whether express, implied or statutory, including but not limited to any warranty of merchantability, noninfringement or fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification or sample.

In no event will any specification co-owner be liable to any other party for any loss of profits, loss of use, incidental, consequential, indirect or special damages arising out of this specification, whether or not such party had advance notice of the possibility of such damages. Further, no warranty or representation is made or implied relative to freedom from infringement of any third party patents when practicing the specification.

Other product and corporate names may be trademarks of other companies and are used only for explanation and to the owner's benefit, without intent to infringe.

REVISION HISTORY

REV	DATE	DESCRIPTION	EDITED BY
1.0	28 Mar 2005	First public release.	Robert V. White Artesyn Technologies

Table Of Contents

Table Of Figures

Table Of Tables

1. Introduction

The Power Management Bus ("PMBus™") is an open standard protocol that defines a means of communicating with power conversion and other devices.

For more information, please see the System Management Interface Forum Web site: www.powerSIG.org.

1.1. Specification Scope

1.1.1. Specification Structure

The PMBus specification is in two parts. Part I includes the general requirements, defines the transport, and defines the electrical interface and timing requirements of hardwired signals.

Part II, this document, describes the operation of commands, data formats, fault management and defines the command language used with the PMBus.

1.1.2. What Is Included

This specification defines a protocol to manage a power converters and a power system via communication over a digital communication bus.

1.1.3. What Is Not Included In the PMBus Specification

The PMBus specification is not a definition or specification of:

- A particular power conversion device or family of power conversion devices
- A specification of any individual or family of integrated circuits.

This specification does not address direct unit to unit communication such as analog current sharing, real-time analog or digital voltage tracking, and switching frequency clock signals.

1.2. Specification Changes Since The Last Revision

This is the first public release.

1.3. Where To Send Feedback And Comments

Please send all comments by email to: questions@powerSIG.org.

2. Related Documents

2.1. Scope

If the requirements of this specification and any of the reference documents are in conflict, this specification shall have precedence unless otherwise stated.

Referenced documents apply only to the extent that they are referenced.

The latest version and all amendments of the referenced documents at the time the power system is released to manufacturing apply.

2.2. Applicable Documents

Applicable documents include information that is, by extension, part of this specification.

[A01] PMBus Power System Management Protocol, Part I, General Requirements, Transport And Electrical Interface

[A02] SBS Implementers Forum, *System Management Bus (SMBus) Specification*, Version 1.1, 11 November 1998

[A03] SBS Implementers Forum, *System Management Bus (SMBus) Specification*, Version 2.0, 3 August 2000

[A04] *The I²C-Bus Specification*, Version 2.1, Philips Semiconductors, January 2000

2.3. Reference Documents

Reference documents have background or supplementary information to this specification. They do not include requirements or specifications that are considered part of this document.

None in this revision.

3. Reference Information

3.1. Signal and Parameter Names

The names of signals and parameters are given in capital letters. Underscores are used to separate words rather than embedded spaces (example: SIGNAL_NAME).

The names of signals that are active low and parameters that are true when the value is 0 are indicated with an octothorpe (#) suffix (example: WRITE# means that the device can be written when the signal is low).

3.2. Numerical Formats

All numbers are decimal unless explicitly designated otherwise.

3.2.1. Decimal Numbers

Numbers explicitly identified as decimal are identified with a suffix of “d”.

3.2.2. Binary Numbers

Numbers in binary format are indicated by a suffix of ‘b’. Unless otherwise indicated, all binary numbers are unsigned.

3.2.3. Hexadecimal Numbers

Numbers in hexadecimal format are indicated by a suffix of ‘h’.

3.2.4. Examples

255d⇔ FFh ⇔ 11111111b

175d⇔ AFh ⇔ 10101111b

3.3. Bit And Byte Order

As specified in [A02]:

- When data is transmitted, the lowest order byte is sent first and the highest order byte is sent last.
- Within any byte, the most significant bit (MSB) is sent first and the least significant bit (LSB) is sent last.

3.4. Bit And Byte Illustrations

The transmission of bits, bytes and packets is illustrated in this section.

In all cases, the least significant bit is indicated as Bit 0. The most significant bit of a byte is always Bit 7, as shown below in Figure 1

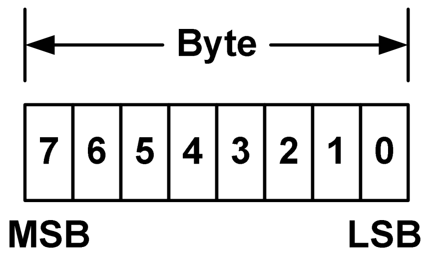

Figure 1. Bit Order Within A Byte

Within this specification, transactions over the PMBus are described. The symbols used to describe the details of those transactions and protocols are shown in Table 1.

Table 1. Bit And Byte Symbols Used In This Specification

Symbol	Meaning
7	A vertical rectangle indicates a single bit sent from the host (bus master) to a slave
7	A vertical rectangle with a shaded interior indicates a bit sent from a slave device to the bus master.
7	A rectangle with a number over it represents one or more bits, as indicated by the number
S	The START condition sent from a bus master device
Sr	A REPEATED START condition sent from a bus master device
A	An Acknowledge (ACK) condition send from the host
NA	A Not Acknowledge (NACK) condition sent from the host
A	An ACKnowledge condition sent from a slave device

Symbol	Meaning
NA	A NOT ACKnowledge condition sent from a slave device
P	A STOP condition sent by a bus master device
7 SLAVE ADDRESS	The first seven bits of the address byte, generally corresponding to the physical address of the device.
R	The eighth bit of the address byte with a value of 1, indicating the device is being addressed with a read.
W	The eighth bit of the address byte with a value of 0, indicating the device is being addressed with a write.
7 BROADCAST ADDRESS	The SMBus broadcast address to which all devices must respond. The value is 0000000b. This always used only with the eighth bit equal to 0 (write).
8 COMMAND CODE	A one byte value that indicates a command the slave device is to execute
8 LOW DATA BYTE	In a two byte value, the lower order byte (bits [7:0]).
8 HIGH DATA BYTE	In a two byte value, the higher order byte (bits [15:8]).
8 PEC	A byte with the Packet Error Check (PEC) value, if used.
• • •	The bit/byte/packet diagram is continued on the next line.

3.5. Abbreviations, Acronyms And Definitions

Term	Definition
ACK	ACKnowedge. The response from a receiving unit indicating that it has received a byte. See [A02] for more information.
Assert, Asserted	A signal is asserted when the signal is true. For example, a signal called FAULT is asserted when a fault has been detected. See Negate.
Bias, Bias Power	Power to the PMBus device's control circuit or ICs
Default Store	A non-volatile memory store most typically used by the PMBus device manufacturer to store default values
Disable, Disable Output	To instruct the PMBus device to stop the power conversion process and to stop delivering energy to the output. The device's control circuitry remains active and the device can communicate via the SMBus.
Enable, Enable Output	To instruct the PMBus device to start the power conversion process and to start delivering energy to the output.
Host	A host is a specialized master that provides the main interface to the system's CPU. A host must be a master-slave and must support the SMBus host notify protocol. There may be at most one host in a system. See [A02] for more information.
IIN	Input current
Inhibit	To stop the transfer of energy to the output while a give condition, such as excessive internal temperature, is present.
IOUT	Output current
LSB	Least significant bit
Master	A master is a device that issues commands, generates the clocks, and terminates the transfer. See [A02] for more information.
MFR	Manufacturer
MSB	Most significant bit
NACK	Not ACKnowledge. The response from a receiving unit that it has received invalid data. See [A02] for more information.
Negate, Negated	A signal is negated when the signal is false. For example, a signal called FAULT is negated when no fault has been detected. See Assert.
Negative Output Current	Current that flows into the converter's output.
OC	Overcurrent
Operating Memory	The conceptual location where a PMBus maintains the data and parameters it uses operate.
OT	Overtemperature
OV	Overvoltage
PEC	Packet Error Checking. See [A02] for more information.

Term	Definition
Pin Programmed Values	Values entered into the PMBus device through physical pins. Values can be set, for example, by connecting a pin to ground, connecting a pin to bias power, leaving the pin unconnected or connecting the pin to ground or bias through a resistor.
POL	Point-of-load
Positive Output Current	Current that flows out of the converter's output.
Product Literature	Data sheets, product briefs, application notes or any other documentation describing the operation and application of a device.
Shut Down	Disable or turn off the output. This generally implies that the output remains off until the device is instructed to turn it back on. The device's control circuit remains active and the device can respond to commands received from the SMBus port.
Sink (Current)	A power converter sinks current when current is flowing from the load into the converter's output. The current in this condition is declared to be negative.
Slave	A slave is a device that is receiving or responding to a command. See [A02] for more information.
SMBus	System Management Bus - See [A02] for more information.
Source (Current)	A power converter sources current when current is flowing from the converter's output to the load. The current in this condition is declared to be positive.
Turn Off	Turn Off means to "turn off the output", that is, stop the delivery of energy to the device's output. The device's control circuit remains active and the device can respond to commands received from the SMBus port. The same as Disable. See Turn On.
Turn On	Turn On means to "turn on the output", that is, start the delivery of energy to the device's output. The same as Enable. See Turn Off.
UC	Undercurrent (Excessive sink current by a synchronous rectifier)
User Store	A non-volatile memory store most often used by the PMBus device user to store an image, or snapshot, of the Operating Memory.
UT	Undertemperature
UV	Undervoltage
VIN	Input voltage
VOUT	Output voltage
X	When used to define a binary value X means that the value of that bit is "don't care".

4. Addressing And Grouping

4.1. Device Addresses

Individual PMBus devices are assigned a 7 bit address through a combination of manufacturer fixed bits and user assigned bits. This is described in [A01].

4.2. General Call Address (Global Broadcast)

PMBus devices may respond to the General Call address (00h) as well as their own physical address.

4.3. Sending Commands To A Group

Commands may be sent to more than one PMBus device for simultaneous execution using the Group Command Protocol, described in [A01]

5. Commands

5.1. Commands And Command Codes

PMBus commands are one byte command codes. A listing of PMBus commands and their hexadecimal command codes are listed in APPENDIX I in Table 19.

Command codes are not register addresses in PMBus devices. The mapping of PMBus command codes to memory locations in a PMBus device is left to PMBus device manufacturer.

5.2. Command Extensions

To provide more than the 256 commands possible with a one byte command code, the PMBus provides for two “command extensions” (Section 25). One of these extensions is made available to PMBus device manufacturers for manufacturer specific commands. The other is reserved for the future inclusion in the PMBus specifications.

The Command code extensions essentially make the command code two bytes long. The first byte transmitted is the Command Code Extension command code. The second byte of the extended command code identifies the action the PMBus device is to take.

These two command extensions use the Command Extension Protocols described in [A01].

5.3. Command Execution

PMBus devices are to process and execute commands as soon as possible after the STOP condition is recognized. PMBus devices do not wait for a separate “Execute” command that launch the previously received command.

5.4. Writing And Reading PMBus Devices

5.4.1. Read Or Write Set Through Address Byte

Separate command codes are not provided for writing and reading PMBus devices. Whether a command is interpreted as a write or a read is determined by the eighth bit of the address byte ([A02]).

5.4.2. Every Parameter That Can Be Written Must Be Readable

Any command that is used to write a parameter into a PMBus device must also support the host reading that parameter from the PMBus device. For example, the VOUT_OV_FAULT_LIMIT command is used to set the output overvoltage threshold value. To set output overvoltage fault threshold, the host uses the SMBus Write Word Protocol ([A02]) to send the command code for the VOUT_OV_FAULT_LIMIT command plus two data bytes to the PMBus device.

If the host wants to know what the output voltage at which a PMBus device will declare an output overvoltage fault, it uses the SMBus Read Word protocol. It sends the PMBus device's seven bit address plus a 1 in the eighth bit, the command code for VOUT_OV_FAULT_LIMIT, and then the PMBus device sends to the host copies of the two bytes it has in memory for the output overvoltage fault limit.

5.4.3. Commands May Be Read Only

Not all commands must support writing parameters into a PMBus device. Some commands, such as those that read back status bits or parameters like output voltage, are inherently read only. PMBus device manufacturers may also make some commands available for reading, but not for writing. Examples might be the VOUT_MODE command (which sets the format of output voltage commands) and commands related to inventory information, such as MFR_MODEL (which can be used to retrieve the manufacturer's model number).

6. Memory Model, Startup Behavior And Defaults

At the conceptual level, PMBus devices operate from values, such as the commanded output voltage, stored in volatile memory. This volatile memory, for purposes of describing the conceptual operation of a PMBus device, is called the Operating Memory. When bias power is applied and the PMBus device control circuitry starts operating, the Operating Memory is loaded from one or more of the following places:

- Values hard coded into an IC design (if any),
- Values programmed from hardware pins (if any),
- A non-volatile memory called the Default Store (if supported in the device),
- A non-volatile memory called the User Store (if supported in the device), or
- Communications from the SMBus.

The relationships between the conceptual Operating Memory and each of the possible sources for loading the Operating Memory, are illustrated in Figure 2 below.

6.1. Order Of Memory Loading And Precedence

To illustrate the precedence of loading parameters into the conceptual Operating Memory, this section uses the conceptual model shown in Figure 2 and Figure 3. This model, and the discussion in this section, are only to illustrate the precedence of how parameters are set within the PMBus device. Any implementation is acceptable so long as it preserves the precedence described in this section.

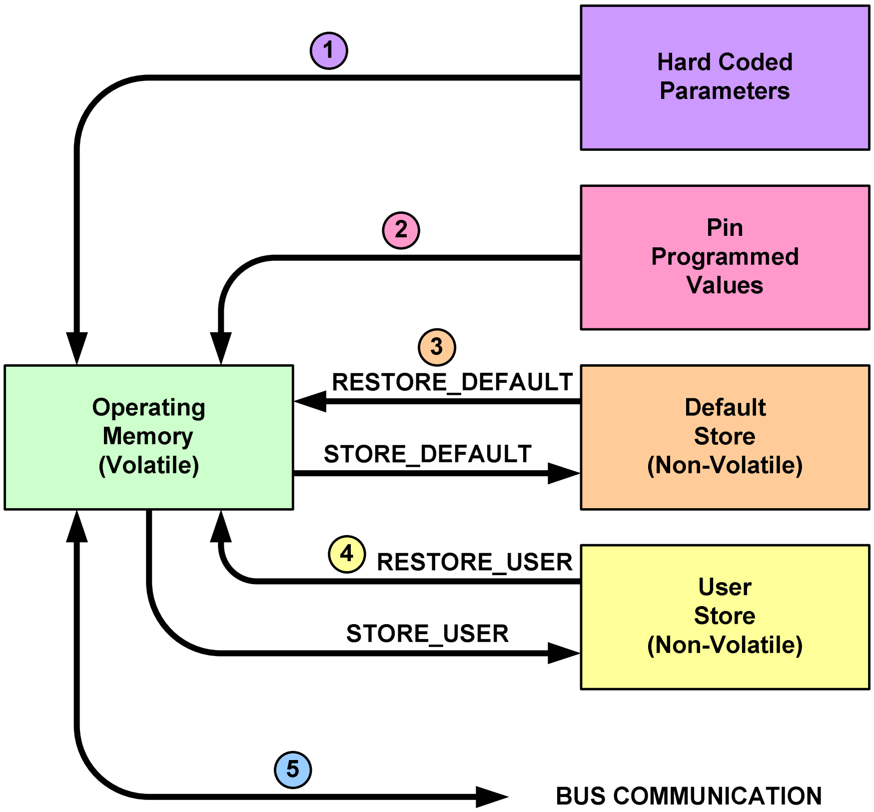

Figure 2. Conceptual View Of Possible PMBus Device Memory And Communication

The first parameters loaded into the Operating Memory are any hard coded parameters.

The second parameters loaded into the Operating Memory come from the pin programming. If any of the parameters programmed by the pins are the same as a parameter that was hard coded, the pin programmed value overwrites the previously loaded hard coded value.

This is the general rule: When parameters are loaded, they will overwrite the same parameter that is already in the Operating Memory.

The third set of parameters loaded comes from the optional non-volatile Default Store, if it exists. The values in the Default Store are usually programmed by the PMBus device manufacturer. The device manufacturer may or may not allow the user to overwrite the manufacturer provided values in the Default Store.

The fourth set of parameters loaded comes from the optional non-volatile User Store, if it exists. The User Store is most often used to store a “snapshot” of the Operating Memory once a device has been programmed and adjusted for operation. By storing a copy of the Operating Memory in the User Store, a device will resume operation with the last set of values stored by the User.

And finally, once the previous steps have finished, the PMBus device will start accepting commands from the SMBus. Note that this means that values written from the bus will overwrite all previous values, including those that were hard coded, pin programmed or copied from the Default and User Stores.

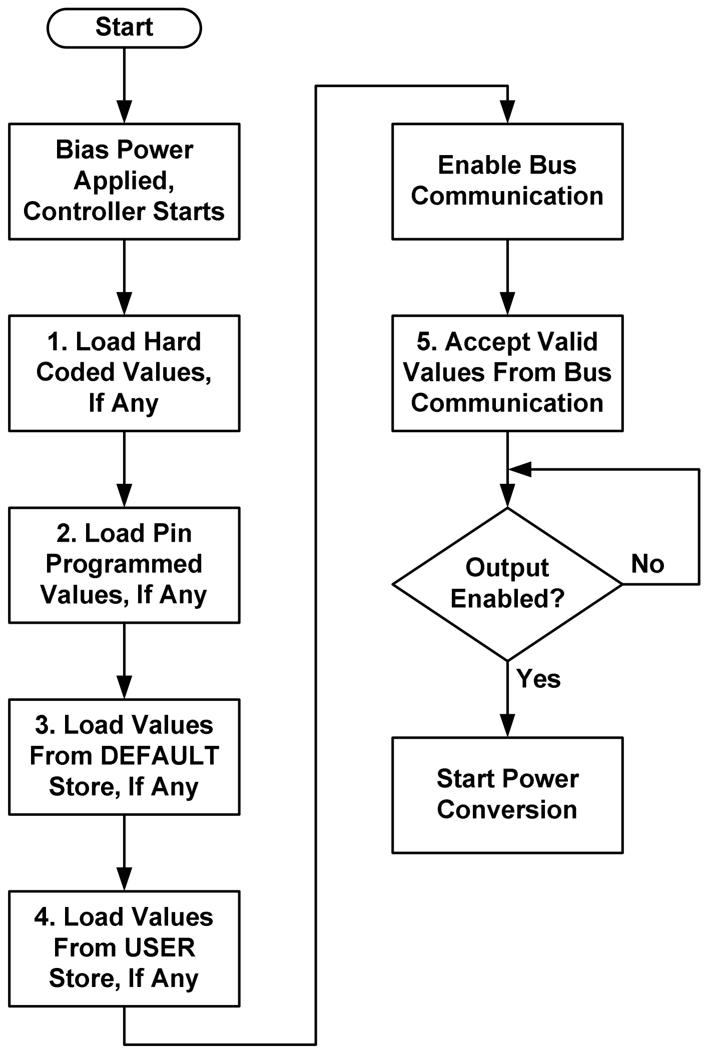

Figure 3. Flowchart Of Conceptual Loading Operating Memory At Startup

6.2. The Default And User Stores

The Default Store and User Store are optional.

Four commands are provided to manipulate the contents of these two non-volatile memory stores.

To copy the entire contents of Operating Memory into the Default Store, the STORE_DEFAULT_ALL command (Section 11.2) is used. To store just one parameter in the Default Store, the STORE_DEFAULT_CODE command (Section 11.4) is used. PMBus device manufacturers may not permit these operations. If STORE_DEFAULT_ALL or STORE_DEFAULT_CODE are permitted, they may generally be commanded when the PMBus device is operating and supplying power to the output. However, this may result in unpredictable and even catastrophic results. It is recommended that the output be disabled before issuing a STORE_DEFAULT_ALL or STORE_DEFAULT_CODE command.

To copy the entire contents of the Default Store into Operating Memory, the RESTORE_DEFAULT_ALL command (Section 11.3) is used. To copy just one parameter from the Default Store to Operating Memory, the RESTORE_DEFAULT_CODE command (Section 11.5) is used. These commands may generally be executed while the device is operating, but can result in unpredictable and even catastrophic results. It is recommended that the output be disabled before issuing a RESTORE_DEFAULT_ALL or RESTORE_DEFAULT_CODE command.

To copy the entire contents of Operating Memory into the User Store, the STORE_USER_ALL command (Section 11.6) is used. To store just one parameter in the User Store, the STORE_USER_CODE command (Section 11.8) is used. The STORE_USER_ALL or STORE_USER_CODE commands may generally be issued when the PMBus device is operating and supplying power to the output. However, this may result in unpredictable and even catastrophic results. It is recommended that the output be disabled before issuing a STORE_USER_ALL or STORE_USER_CODE command.

To copy the entire contents of the User Store into Operating Memory, the RESTORE_USER_ALL command (Section 11.7) is used. To copy just one parameter from the User Store to Operating Memory, the RESTORE_USER_CODE command (Section 11.9) is used. These commands may be generally be executed while the device is operating and supplying power to the output, but this can result in unpredictable and even catastrophic results. It is recommended that the output be disabled before issuing a RESTORE_USER_ALL or RESTORE_USER_CODE command.

7. Data Formats

Except for the output voltage (see Section 8), PMBus devices generally receive and report data in two possible formats. Any given device need support only one of the two formats.

PMBus devices using the first format, LITERAL, receive and transmit values as volts, amperes, milliseconds or degrees Celsius. This format provides the least burden on the host and minimizes bus traffic at the expense of data manipulation in the PMBus device.

PMBus devices using the second format, DIRECT receive and transmit data as a two byte two's complement binary integer. To command a value or interpret a value received from a PMBus device, the host must perform calculations using coefficients retrieved from the PMBus device. This format provides the least burden on the PMBus device at the expense of more complex calculations in the host.

Any parameters that do not use either of these formats have their data format described explicitly in the section describing the command that receives or transmits that parameter.

The product literature for each PMBus device shall describe which data format is used for each PMBus command the device supports.

7.1. LITERAL Data Format

The Literal Data Format is typically used for commanding and reporting the parameters such as the following:

- Output Current,
- Input Voltage,
- Input Current,
- Operating Temperatures,

- Time (durations) and
- Energy Storage Capacitor Voltage.

The Literal Data Format is a two byte value with:

- An 11 bit, two's complement mantissa and
- A 5 bit, two's complement exponent (scaling factor).

The format of the two data bytes is illustrated in Figure 4.

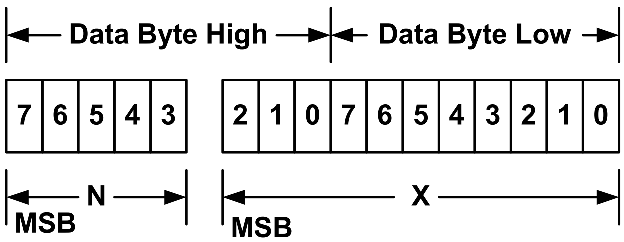

Figure 4. Literal Data Format Data Bytes

The relation between X, N and the communicated value is:

$Y = X \cdot 2^N$

Where, as described above:

Y is the value being communicated;

X is an 11 bit, two's complement integer; and

N is a 5 bit, two's complement integer.

7.2. DIRECT Data Format

DIRECT format data is a two byte, two's complement binary integer. DIRECT format data may be used with any command that sends or reads a parametric value.

If a PMBus device uses DIRECT form data, this shall be clearly described in the product literature.

7.2.1. Interpreting Received Values

The host system uses the following equation to convert the value received from the PMBus device into a reading of volts, amperes, degrees Celsius or other units as appropriate:

$$Y = (mX + b) \bullet 10^R$$

Where:

Y, is the calculated value in the appropriate units (A, V, °C, etc.);

m, the slope coefficient, is a two byte, two's complement integer;

X, a two byte two's complement integer received from the PMBus device;

b, the offset, is a two byte, two's complement integer; and

R, the exponent, is a two byte, two's complement integer.

7.2.2. Sending A Value

To send a value, the host must use the equation in Section 7.2.1 solved for X:

$$X = \frac{1}{m}(Y \bullet 10^{-R} - b)$$

Where:

X is the two byte two's complement integer to be sent to the unit that is most closely equivalent to the decimal value calculated from m, Y, R and b;

m, the slope coefficient, is the decimal value equivalent to the two byte, two's complement integer retrieved from the PMBus device;

Y, a decimal value, in units such as amperes or volts, to be converted for transmission;

b, the offset, is the decimal value equivalent to the two byte, two's complement integer retrieved from the PMBus device; and

R, the exponent, is the decimal value equivalent to the one byte, two's complement integer retrieved from the PMBus device.

7.2.3. Retrieving The Coefficients

The m, b and R coefficients used in the DIRECT mode are programmed by the device manufacturer and are typically read only.

Before a host either sends information to or retrieves information from a PMBus device using DIRECT mode, it must retrieve the appropriate coefficients for the parameter of interest using the COEFFICIENTS command (Sections 7.2 and 14.1). Note that for a given parameter, such as output voltage, the coefficients used to set the value and to read the value are generally not the same.

The values used by the COEFFICIENTS command to retrieve m, b and R for each supported parameter are listed in APPENDIX II.

7.3. Accuracy

The accuracy of commanded and reported data shall be given in the PMBus device's product literature.

7.4. Resolution

PMBus devices may have an internal data resolution less than the transmitted value. For example, VOUT_COMMAND sends 16 bits in its data bytes. Yet a PMBus device might use only 10 of the 16 in commanding an output voltage. This is permitted and considered compliant.

When reading back information from a PMBus that uses a native resolution less than the number of bits used in the write version of the command, it is permissible for the PMBus device to return zero values for the lower order bits it does not support. In the example about, with the 10 bit resolution for output voltage, using the SMBus Read Word protocol with the VOUT_COMMAND command code would return the 10 highest order bits that were sent to the device. The six lowest order bits would be all zeros regardless of what was sent to the device with the original SMBus Write Word command with the VOUT_COMMAND command code. This behavior is considered compliant.

8. Data Formats For The Output Voltage And Output Voltage Related Parameters

Voltage data for commanding or reading the output voltage or related parameters (such as the overvoltage threshold) can be in one of three different formats depending on the type of device. PMBus device product literature shall clearly identify which of the formats the device is capable of supporting.

The three formats of commanding and reporting voltage are:

- A LINEAR scale that is commanded and reported using a two byte unsigned binary integer with a scaling factor (similar in concept to a mantissa and exponent),
- A format that supports transmitting the VID codes of popular microprocessors via the PMBus, and
- The DIRECT format (7.2) that uses an equation and device supplied coefficients.

Power supplies and power converters generally have no way of knowing how there outputs are connected to ground. Within the power supply, all output voltages most commonly are treated as “positive”. Accordingly, all output voltages and output voltage related parameters of PMBus devices are commanded and reported as positive values. It is up to the system to know that a particular output is negative, it that is of interest to the system.

8.1. Two Step Process

Commanding or reading a voltage requires two steps.

The first step is to set or read which of the three formats (LINEAR, VID, DIRECT) the device uses for output voltage related data. This is done with the VOUT_MODE command (Section 8.2).

The VOUT_MODE command is only issued when the format of the output voltage data changes. For some devices, this may be written only once in the device’s life.

After the VOUT_MODE command is used to set or read the format of the output voltage data, other commands are used to set, adjust or read back output voltage related information. For example, the VOUT_COMMAND is used to set the voltage to which the device should set the output. The VOUT_OV_FAULT_LIMIT command is used to set the output overvoltage fault threshold.

8.2. VOUT_MODE Command

The data byte for the VOUT_MODE command is one byte that consists of a three bit Mode and a five bit Parameter as shown in Figure 5. The three bit Mode sets whether the device us using the Linear, VID or Direct modes for output voltage related commands. The five bit Parameter provides more information about the selected mode, such as which manufacturer’s VID codes are being used.

Sending the VOUT_MODE command with the address set for writing sets the Mode and Parameter into the PMBus device, if it accepts changes to these values.

PMBus devices may have the Mode and Parameter set at the time of manufacture and may not permit the user to change these values. In this case, the device shall NACK the VOUT_MODE command’s data byte.

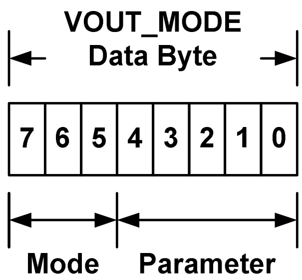

Figure 5. VOUT_MODE Command Data Byte Structure

If a device accepts the VOUT_MODE command, the Mode and Parameter are retained until changed with another VOUT_MODE command or until the bias power is removed.

Sending the VOUT_MODE command using the SMBus Read Byte protocol returns one byte with the Mode and Parameter as shown in Figure 5.

The details of the Mode and Parameter bits for the three data formats are given below.

8.3. Data Bytes For Output Voltage Related Commands

There are several commands that either set or adjust the output voltage, or a related parameter, of a converter that supports the PMBus protocol. Some examples are:

- VOUT_COMMAND which causes the converter to set its output voltage to the commanded value;
- VOUT_TRIM, which is available to the device user to trim the output voltage; and
- VOUT_OV_FAULT_LIMIT, which sets the output voltage above which an output overvoltage fault is declared.

All output voltage related commands use two data bytes. The contents of those data bytes depend on the voltage data format in use (set by the VOUT_MODE command) and are described below.

8.3.1. Linear Mode

The data bytes for the VOUT_MODE and VOUT_COMMAND when using the Linear voltage data format are shown in Figure 6.

Note that the VOUT_MODE command is sent separately from output voltage related commands and only when the output voltage format changes. VOUT_MODE is not sent every time an output voltage command is sent.

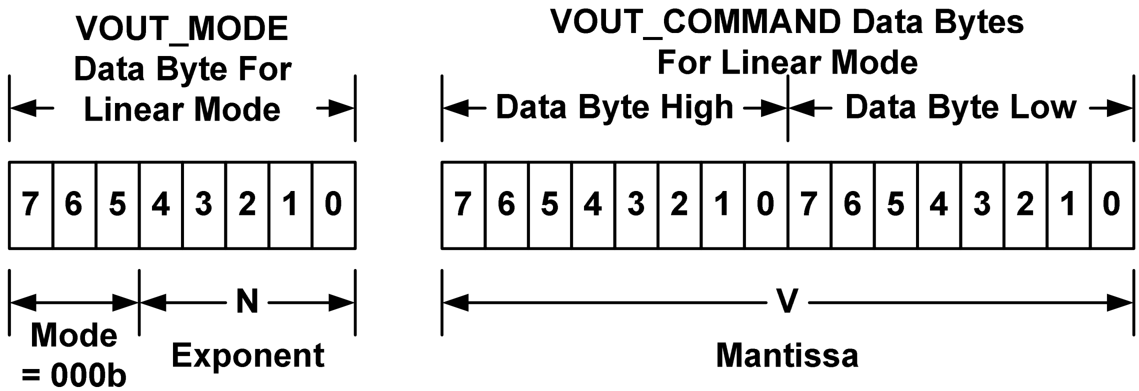

Figure 6. Linear Format Data Bytes

The Mode bits are set to 000b.

The Voltage, in volts, is calculated from the equation:

$$Voltage = V \bullet 2^N$$

Where:

Voltage is the parameter of interest in volts;

V is a 16 bit unsigned binary integer; and

N is a 5 bit two's complement binary integer.

8.3.2. VID Format

The data bytes for the VOUT_MODE and VOUT_COMMAND when using the VID voltage data format are shown in Figure 7. Note that the VOUT_MODE command is sent separately from output voltage related commands and only when the output voltage format changes. VOUT_MODE is not sent every time an output voltage command is sent.

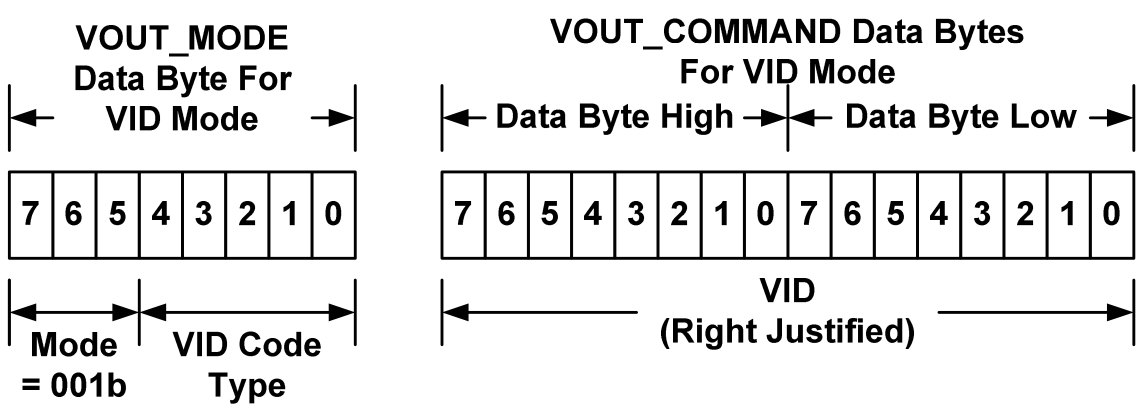

Figure 7. VID Format Data Bytes

The Mode bits are set to 001b. The VID Code Type is an unsigned binary integer. The defined values of VID Code Type are given below in Table 2. Any VID Code Types not listed in Table 2 are reserved for future use and shall not be used until listed in a future revision of this specification.

Table 2. VID Types Supported By PMBus

VID Code Type	Microprocessor Family
00h	Not Used
01h	Reserved For A Future Generation Intel Microprocessor
02h	Reserved For A Future Generation Intel Microprocessor
03h	Reserved For A Future Generation Intel Microprocessor
04h	Reserved For A Future Generation Intel Microprocessor
10h	Reserved For A Future Generation AMD Microprocessor
11h	Reserved For A Future Generation AMD Microprocessor
1Ch	Reserved For Future Use
1Dh	Reserved For Future Use
1Eh	PMBus Device Manufacturer Specific
1Fh	PMBus Device Manufacturer Specific

VID Code Types 1Eh and 1Fh are provided so that PMBus device makers can provide customized or manufacturer specific VID codes. The details of the relationship between the VID codes and output voltage shall be provided in the PMBus device product literature.

Within the output voltage related command data bytes, the VID code shall be right justified with VID0 in bit 0 of the lower data byte, VID1 in bit 1 of the lower byte and so forth until all applicable VID bits are used. Any unused bits in the data bytes shall be filled with zeroes.

8.3.3. DIRECT Format

The DIRECT data format can also be used to command or read output voltage related values. See Section 7.2 for the details on this data format is used.

When the DIRECT format is used to set the output voltage, the coefficients m, b and R are generally chosen by the PMBus device manufacturer so that the minimum voltage to be commanded results in a value of 0 for X. The result of the equation for the maximum value to be commanded generally results in a value of 2^{16}-1. The result of the calculation is converted to a 16 bit unsigned binary integer and transmitted as the data bytes of a VOUT_COMMAND command.

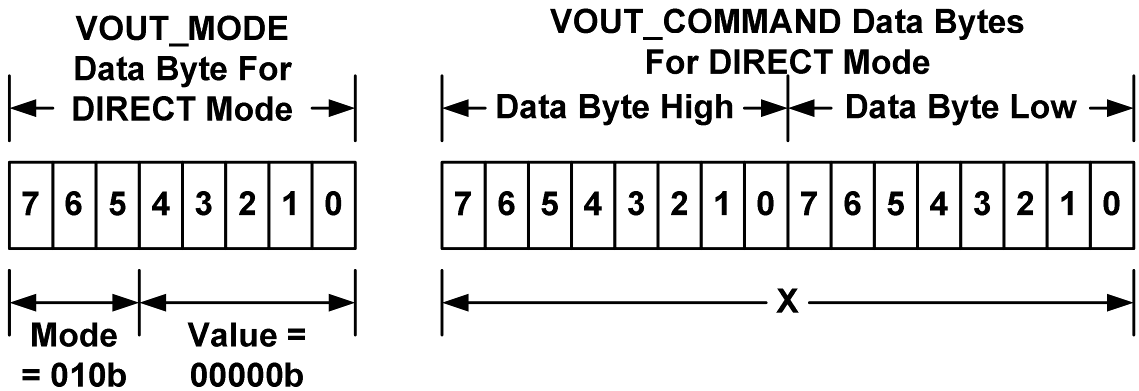

Figure 8. DIRECT Format Mode Data Bytes

The X shown in the VOUT_COMMAND data byte in Figure 8 is the value used in conjunction with the coefficients m, b and R to calculate the desired value. See Section 7.2 for the details.

9. Setting And Monitoring The Output Voltage

There are several commands that affect how a PMBus device responds to output voltage related commands. This section provides a conceptual description of how those commands work. The actual implementation is left to the PMBus device manufacturers.

9.1. VOUT_SCALE_LOOP And VOUT_SCALE_MONITOR

In typical power converters the output voltage of a converter is sensed through a resistive voltage divider, as illustrated in Figure 9. The resistive divider reduces, or scales, the output voltage so that when the output voltage is corrected, the value sensed by the control circuit is equal to the reference voltage.

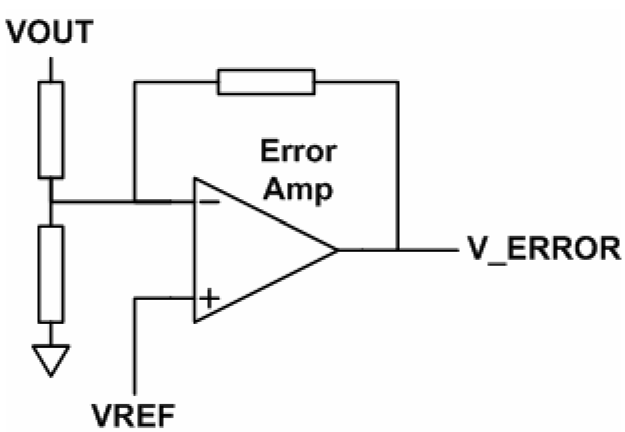

Figure 9. Output Voltage Sensing In A Typical Power Converter

Many converters supporting the PMBus protocol will have a resistive voltage divider between the output and the input to the converter's control circuit or IC. However, commands sent over the PMBus command the output voltage. To allow PMBus devices to map between the commanded voltage (such as 3.3 V), and the voltage at the control circuit input (perhaps 3.3 V divided down to match a reference voltage of 1.2 V), the VOUT_SCALE_LOOP (Section 13.10) command is used.

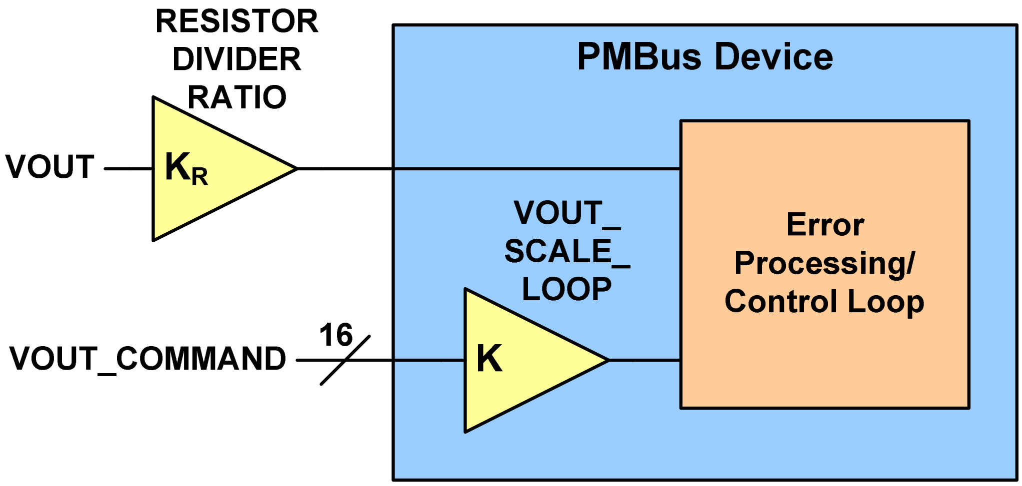

Figure 10. Conceptual View Of The Application Of The VOUT_SCALE_LOOP Command

Figure 10 shows a conceptual view of how the VOUT_SCALE_LOOP command works. The output voltage, VOUT, is processed through a resistive divider with a ratio of output to input equal to K_R. Suppose, for example, the output voltage was 3.3 V and that the desired input to the PMBus device is 1.2 V. Then K_R is calculated as follows:

$$K_R = \frac{1.2\text{V}}{3.3\text{V}} = 0.3636...$$

The PMBus device needs to take account of the external resistive divider when processing output voltage related commands. The simplest concept is simply to think of the voltage command being scaled by the same amount and the actual output voltage. This shown by the 16 bit VOUT_COMMAND being applied to a gain block labeled as VOUT_SCALE_LOOP. If the gain of that block, K, is the same as the resistive divider ration, K_R, then in concept, the values applied to the control circuitry from the output voltage sensing network and the voltage command input, will be the same when the output is at the desired value.

This discussion illustrates the concept and use of the VOUT_SCALE_COMMAND. PMBus device users are instructed to consult the PMBus device manufacturer's product literature for information on how this command is implemented in any devices of interest.

In devices that provide an independent path for sensing the output voltage, such as for the output overvoltage protection circuit or the circuit that processes the sensed output voltage for the READ_VOUT command, a second scale factor, VOUT_SCALE_MONITOR (Section 13.11), is provided. This scale factor, in concept, works the same as the VOUT_SCALE_LOOP command. PMBus device users are directed to the manufacturer's literature for information on the VOUT_SCALE_COMMAND is used in any devices of interest.

9.2. Setting The Output Voltage

There are several commands that are used in commanding the output voltage of a power converter with a PMBus interface. These include:

- VOUT_MODE (Section 8.2),
- VOUT_COMMAND (Section 8),

- VOUT_TRIM (Section 13.3),
- VOUT_CAL (Section 13.4),
- VOUT_MAX (Section 13.5),
- VOUT_MARGIN_HIGH (Section 13.6),
- VOUT_MARGIN_LOW (Section 13.7),
- VOUT_DROOP (as a function of IOUT) (Section 13.9), and
- VOUT_SCALE_LOOP (Sections 13.10 and 9.1).

Figure 11 shows a conceptual view of how these commands are used to control the output voltage. The actual implementation is left to the PMBus device makers so long as the overall behavior is the same as shown in Figure 11.

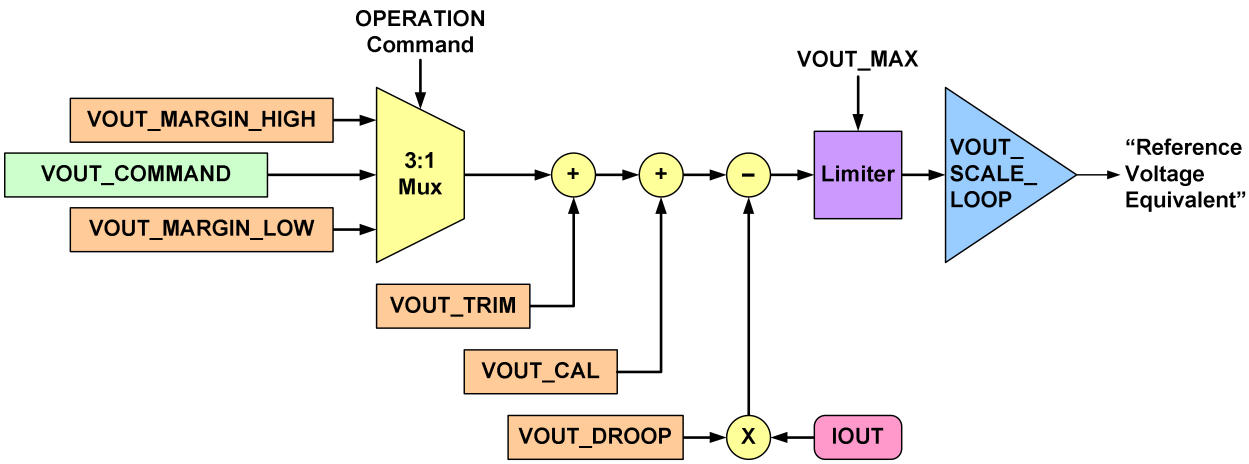

Figure 11. Conceptual View Of How Output Voltage Related Commands Are Applied

In Figure 11, the values of the various parameters may come from:

- Hard coded values embedded in the PMBus device,
- Pin programming,
- The conceptual non-volatile Default Store,
- The conceptual non-volatile User Store, or
- Commands received from the SMBus port.

This process of loading parameters was described in Section 6.

The process of setting the output voltage starts with three basic commands for output voltage: VOUT_COMMAND, VOUT_MARGIN_HIGH and VOUT_MARGIN_LOW. One of these three values is selected by the OPERATION command (Section 12.1) and passed on to the rest of the output voltage command processing.

The next step is to add the value in the VOUT_TRIM register to the output of the conceptual multiplexer. The value in the VOUT_TRIM register is a two's complement number that can either add to or subtract from the value from the conceptual multiplexer. The VOUT_TRIM register will typically be used by the end user to adjust the output voltage once the PMBus device is assembled into the end user's system. This might be done, for example, to adjust the voltage at the pins of a critical IC to optimize its performance.

Next, the value from the VOUT_CAL register is added. This is also a two's complement number and can add to or subtract from the voltage command value. The VOUT_CAL register will typically be used by the PMBus device manufacturer to adjust the output voltage in their factory.

Next, if the PMBus device has an output voltage droop characteristic, it is applied. The VOUT_DROOP coefficients are always greater than or equal to zero and droop is only applied if the output current is greater than zero. The value of the VOUT_DROOP coefficient and the value of output current are multiplied and the result is always subtracted from the voltage command. VOUT_DROOP can only act to reduce the output voltage. It can never act to increase the output voltage.

The next step is compare the commanded voltage developed so far with the maximum permissible output voltage set by the VOUT_MAX command. If the calculated voltage command would create an output voltage greater than the VOUT_MAX value, the PMBus device limits the command voltage passed to the controller to the VOUT_MAX value. It also sets an alarm as described in Section 13.5.

The next step is to apply the same scaling factor to the calculated voltage command as is applied to the external output voltage by a resistive divider. This is done by multiplying the calculated voltage command by VOUT_SCALE_LOOP.

At this point, the converter now has a calculated value that is used as the equivalent to the reference voltage in standard analog controller. This is the value to which the sensed output voltage is compared when making decisions about adjusting the converter's duty cycle.

10. Fault Management And Reporting

The PMBus protocol provides a comprehensive set of tools for monitoring the operation of and managing the faults in a PMBus device. Provisions are made for a host or power system manager to read a wide range of parametric values, such as the output voltage or output current. The PMBus protocol also includes the ability to program fault or warning levels for every important aspect of a power conversion device.

10.1. Monitoring Operation

The host or power system manager can use READ commands to ask a PMBus device about its current state. To simplify the PMBus devices, there is one READ command for each parameter, such as output voltage or device temperature. The details of the READ commands are given in Section 18.

10.2. General Description Of PMBus Device Fault Management

The PMBus protocol supports setting warning (minor alarm) and fault (major alarm) thresholds for nearly every possible event.

If the PMBus device detects that one of these thresholds has been exceeded, a bit corresponding to the condition is latched.

10.2.1. Warning Conditions

Warning conditions are an indication that the device has a problem but can continue operating.

When the PMBus device detects a warning condition, the device sets the corresponding bit in the status registers. This bit remains set until cleared as described in Section 10.2.3.

Depending on what the PMBus device supports, it will:

- Simply set the warning condition bit and wait for the host or power system manager to poll it or
- The PMBus device may notify the host that a warning condition has occurred (Section 10.6).

10.2.2. Fault Conditions

Fault conditions are more serious than a warning condition and may require the PMBus device to disable the output and stop the transfer of energy to the output.

For fault conditions, the PMBus device can be programmed with a wide range of responses such as shut down immediately and latch off, shut down and retry or continue to operate for a specified delay time before shutting down. The possible fault responses are described in Section 10.5.

In addition, the PMBus device will set the corresponding fault bit in the status registers. This bit remains set until cleared as described in Section 10.2.3.

Depending on what the PMBus device supports, it will:

- Simply set the fault condition bit and wait for the host or power system manager to poll it or
- The PMBus device may notify the host that a fault condition has occurred (Section 10.6).

10.2.3. Clearing The Warning Or Fault Bit

Any warning or fault bits set in the status registers remain set, even if the fault or warning condition is removed or corrected, until:

- The device receives a CLEAR_FAULTS command (Section 15.1),
- A RESET signal (if one exists) is asserted,
- The output is commanded through the CONTROL pin, the OPERATION command, or the combined action of the CONTROL pin and OPERATION command, to turn off and then to turn back on
- Bias power is removed from the PMBus device.

If the warning or fault condition is present when the bit is cleared, the bit is immediately set again. The device shall respond as described in Section 10.2.1 or Section 10.2.2 as appropriate.

10.3. Status Registers

The PMBus protocol provides three levels of status registers. This allows host or power system managers to retrieve the most important information in a fast, one byte transaction. Based on this information the host can act or request more detailed information. Figure 12 shows the relationship between the STATUS_BYTE register, the STATUS_WORD register and the more detailed status registers.

As shown in Figure 12, the STATUS_BYTE register contains the most important fault and warnings. This allows the most basic PMBus devices to provide the most critical information at the lowest cost. The STATUS_WORD includes the STATUS_BYTE as its lower byte. In the higher byte of the STATUS_WORD, there are additional bits providing more information about the status of the PMBus device.

In more advanced PMBus devices, there are seven registers with even more detailed information about the status of the unit. The host or power system manager knows which of these to read based on which bits are set in the STATUS_BYTE or STATUS_WORD.

The bits in each register are set individually. That is, if one bit in a status register is set, and another fault or warning requires another bit in that register to be set, that additional bit is set.

When the status registers are cleared, all bits in all registers are cleared simultaneously.

The details of the STATUS_BYTE, STATUS_WORD and other status registers are given in Section 17.

10.4. Setting Fault And Warning Thresholds

Section 15 includes a comprehensive list of commands to set fault and warning thresholds that PMBus devices may support.

Not all PMBus devices will support all of the fault detection, reporting and management functions and features. The PMBus device product literature shall indicate which features and function it supports.

10.5. Setting The Response To A Detected Fault Condition

Commands are provided to set the response to each fault condition. These commands have one data byte that describes how the device should respond to the fault. Each of the fault response commands requires that the user make three choices about how the device will respond to the fault condition.

The first option is called the Response. The choice to be made is whether or not the device is to continue operating, shutdown or disable the output while the fault condition is present (Inhibit).

The second option is the Retry Setting. The choice to be made for the Retry Setting is whether or not to attempt to restart operation if the device shut down in response to a fault.

The third option is Delay Time. The choice to be made here depends on the choices for the Response and Retry Settings. The device user must choose either:

- The period of time the unit continues to operate without shutting down after a fault is detected, or
- The time between retry attempts.

The details are given in the following sections.

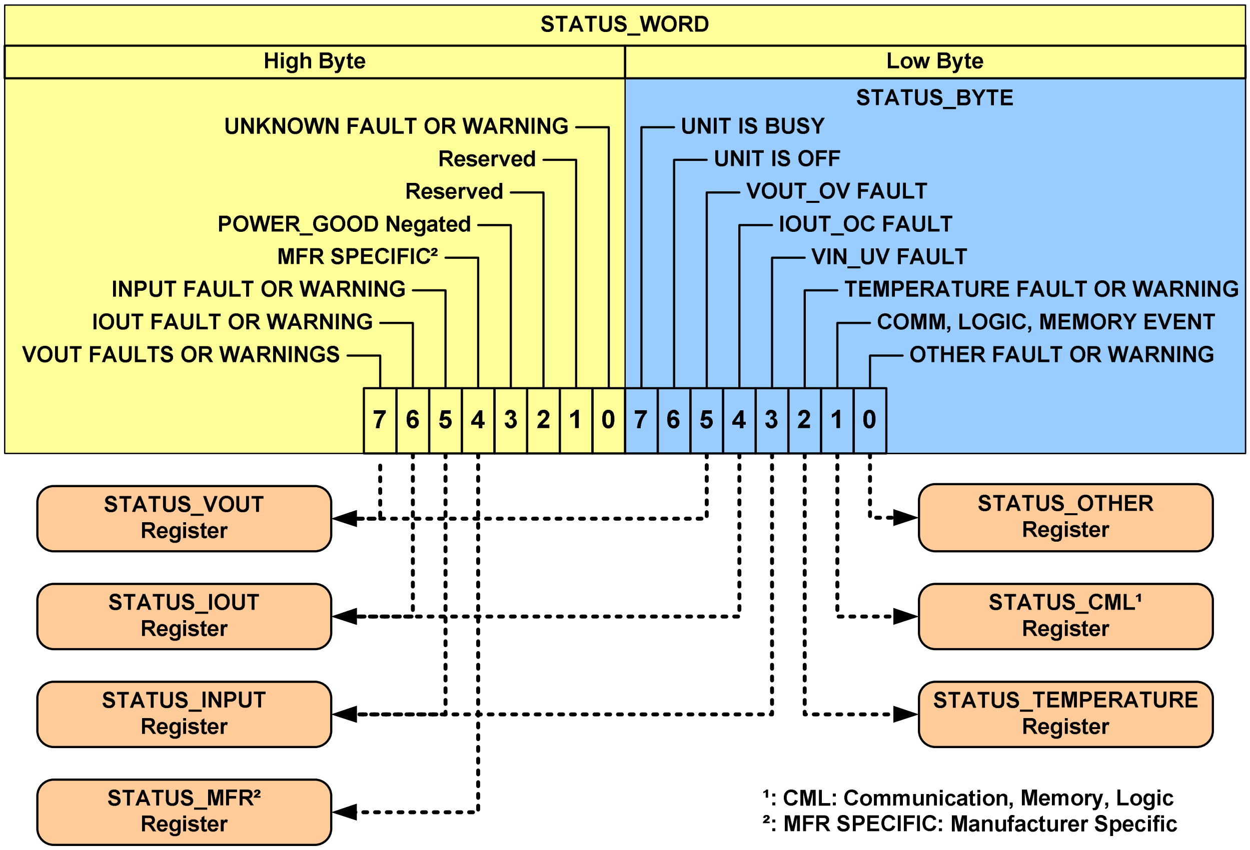

Figure 12. Status Register Map

10.5.1. Response To Voltage And Temperature Faults And Warnings

The data byte specifying the response to a voltage or temperature fault is detailed in Table 3.

Table 3. Voltage And Temperature Faults Response Data Byte Details

Bits	Description	Value	Meaning
7:6	Response For all values of bits [7:6], the device: • Sets the corresponding fault bit in the status registers and • If the device supports notifying the host (Section 10.6), it does so. The fault bit, once set, is	00	The PMBus device continues operation without interruption.
		01	The PMBus device continues operation for the delay time specified by bits [2:0] and the delay time unit specified for that particular fault. If the fault condition is still present at the end of the delay time, the unit shuts down and latches off until the unit is permitted to restart as described in Section 10.7, even if the fault condition is removed or corrects itself after the unit has shutdown.

Bits	Description	Value	Meaning
	cleared only in accordance with Section 10.2.3 and not when the fault condition is removed or is corrected.	10	The device shuts down (diables the output) and responds according to the retry setting in bits [5:3].
		11	The device's output is disabled while the fault is present. Operation resumes and the output is enabled when the fault condition no longer exists.
5:3	Retry Setting	000	A zero value for the Retry Setting means that the unit does not attempt to restart. The output remains disabled until the fault is cleared (Section 10.7).
		001-110	The PMBus device attempts to restart the number of times set by these bits. The minimum number is 1 and the maximum number is 6. If the device fails to restart (the fault condition is no longer present and the device is delivering power to the output and operating as programmed) in the allowed number of retries, it disables the output and remains off until the fault is cleared as described in Section 10.7. The time between the start of each attempt to restart is set by the value in bits [2:0] along with the delay time unit specified for that particular fault.
		111	The PMBus device attempts to restart continuously, without limitation, until it is commanded OFF (by the CONTROL pin or OPERATION command or both), bias power is removed, or another fault condition causes the unit to shut down.
2:0	Delay Time	XXX	The number of delay time units, which vary depending on the type of fault. This delay time is used for either the amount of time a unit is to continue operating after a fault is detected or for the amount of time between attempts to restart.

10.5.2. Response To Current Faults

The data byte specifying the response to a current fault is detailed in Table 4.

Table 4. Current Fault Response Data Byte Details

Bits	Description	Value	Meaning
7:6	Response For all values of bits [7:6], the device: • Sets the corresponding fault bit in the status registers and • If the device supports notifying the host (Section 10.6), it does so. The fault bit, once set, is cleared only in accordance with Section 10.2.3 and not when the fault condition is removed or is corrected.	00	The PMBus device continues to operate indefinitely while maintaining the output current at the value set by IOUT_OC_FAULT_LIMIT (Section 15.8) without regard to the output voltage (known as constant-current or brickwall limiting).
		01	The PMBus device continues to operate indefinitely while maintaining the output current at the value set by IOUT_OC_FAULT_LIMIT (Section 15.8) as long as the output voltage remains above the minimum value specified by IOUT_OC_UV_FAULT_LIMIT (Section 15.10). If the output voltage is pulled down to less than that value, then the PMBus device shuts down and responds according to the Retry setting in bits [5:3].
		10	The PMBus device continues to operate, maintaining the output current at the value set by IOUT_OC_FAULT_LIMIT (Section 15.8) without regard to the output voltage, for the delay time set by bits [2:0] and the delay time units for specified in the IOUT_OC_FAULT_RESPONSE (Section 15.9).
		11	The PMBus device shuts down and responds as programmed by the Retry Setting in bits [5:3].
5:3	Retry Setting	000	A zero value for the Retry Setting means that the unit does not attempt to restart. The output remains disabled until the fault is cleared (Section 10.7).

Bits	Description	Value	Meaning
		001-110	The PMBus device attempts to restart the number of times set by these bits. The minimum number is 1 and the maximum number is 6. If the device fails to restart (the fault condition is no longer present and the device is delivering power to the output and operating as programmed) in the allowed number of retries, it disables the output and remains off until the fault is cleared as described in Section 10.7. The time between the start of each attempt to restart is set by the value in bits [2:0] along with the delay time unit specified for that particular fault.
		111	The PMBus device attempts to restart continuously, without limitation, until it is commanded OFF (by the CONTROL pin or OPERATION command or both), bias power is removed, or another fault condition causes the unit to shut down.
2:0	Delay Time	XXX	The number of delay time units, which vary depending on the type of fault. This delay time is used for either the amount of time a unit is to continue operating after a fault is detected or for the amount of time between attempts to restart.

10.6. Reporting Faults And Warnings To The Host

PMBus devices may support notifying the host if a fault or warning is detected.

There are two means available for a PMBus device to notify the host of a fault or error: the SMBALERT# signal and direct communication from the PMBus device to the host. PMBus devices shall support at most one of the two methods.

10.6.1. SMBALERT# Signal And Process

The SMBALERT# process is described in [A02].

10.6.2. Direct PMBus Device To Host Communication

PMBus devices may temporarily become bus masters, as permitted in [A02], in order to send notice to the host that a fault or an error has occurred. The format of the packet is shown in Figure 13.

1	7	1	1	7	1	1	8	1	8	1	1
S	HOST ADDRESS 0001 000	W	A	DEVICE'S ADDRESS	X	A	LOW DATA BYTE	A	HIGH DATA BYTE	A	P

Note That It Is The PMBus Device That Wants To Notify The Host That Sends The START Condition, Host Address, etc. It Is The Host Sending The ACK Conditions.

Figure 13. Packet Structure For PMBus Device To Notify Host

The data bytes are the same as the STATUS_WORD command (Section 17.2).

10.7. Clearing A Shutdown Due To A Fault

Any device that has shut down due to a fault condition remains off until:

- A RESET signal (if one exists) is asserted,
- The output is commanded through the CONTROL pin, the OPERATION command, or the combined action of the CONTROL pin and OPERATION command, to turn off and then to turn back on, or
- Bias power is removed from the PMBus device.

11. Address And Memory Related Commands

11.1. WRITE_PROTECT

The WRITE_PROTECT command is used to control writing to the PMBus device. The intent of this command is to provide protection against accidental changes. This command is not intended to provide protection against deliberate or malicious changes to a device's configuration or operation.

All supported commands may have their parameters read, regardless of the WRITE_PROTECT settings.

This command has one data byte, described in Table 5.

Table 5. WRITE_PROTECT Command Data Byte

Bits	Value	Meaning
7	1	Disable all writes except for the WRITE_PROTECT command
	0	Enable writes as permitted in bits [6:n]
6	1	Disable all writes except for the WRITE_PROTECT, OPERATION and PAGE commands
	0	Enable all writes as permitted in bits [5:n]
5	1	Disable all writes except for those permitted in bits [7:6] and the ON_OFF_CONFIG and VOUT_COMMAND commands.
	0	Enable all writes as permitted in bits [4:n]
4	X	Reserved
3	X	Reserved
2	X	Reserved

Bits	Value	Meaning
1	X	Reserved
0	1	Disable all writes except those permitted in bits [7:1].
	0	Enable writes to all supported commands.

11.2. STORE_DEFAULT_ALL

The STORE_DEFAULT_ALL command instructs the PMBus device to copy the entire contents of the Operating Memory to the matching locations in the non-volatile Default Store memory. Any items in Operating Memory that do not have matching locations in the Default Store are ignored.

It is permitted to use the STORE_DEFAULT_ALL command while the device is operating. However, the device may be unresponsive during the copy operation with unpredictable, undesirable or even catastrophic results. PMBus device users are urged to contact the PMBus device manufacturer about the consequences of using the STORE_DEFAULT command while the device is operating and providing output power.

This command has no data bytes.

This command is write only.

11.3. RESTORE_DEFAULT_ALL

The RESTORE_DEFAULT_ALL command instructs the PMBus device to copy the entire contents of the non-volatile Default Store memory to the matching locations in the Operating Memory. Any items in Default Store that do not have matching locations in the Operating Memory are ignored.

It is permitted to use the RESTORE_DEFAULT_ALL command while the device is operating. However, the device may be unresponsive during the copy operation with unpredictable, undesirable or even catastrophic results. PMBus device users are urged to contact the PMBus device manufacturer about the consequences of using the RESTORE_DEFAULT_ALL command while the device is operating and providing output power.

This command has no data bytes.

This command is write only.

11.4. STORE_DEFAULT_CODE

The STORE_DEFAULT_CODE command instructs the PMBus device to copy the parameter whose Command Code matches value in the data byte, from the Operating Memory to the matching location in the non-volatile Default Store memory.

If the device does not permit saving this parameter in the Default Store, or if the device does not support the Command Code specified in the data byte, then the device must notify the host that the command failed, as described in [A01].

It is permitted to use the STORE_DEFAULT_CODE command while the device is operating. However, the device may be unresponsive during the copy operation with unpredictable, undesirable or even catastrophic results. PMBus device users are urged to contact the PMBus device manufacturer about the consequences of using the STORE_DEFAULT_CODE command while the device is operating and providing output power.

This command has one data byte, formatted as an unsigned binary integer.

This command is write only.

11.5. RESTORE_DEFAULT_CODE

The RESTORE_DEFAULT_CODE command instructs the parameter whose Command Code matches value in the data byte, from the non-volatile Default Store memory to the matching location in the Operating Memory.

If the device does save this parameter in the Default Store, or if the device does not support the Command Code specified in the data byte, then the device must notify the host that the command failed, as described in [A01].

It is permitted to use the RESTORE_DEFAULT_CODE command while the device is operating. However, the device may be unresponsive during the copy operation with unpredictable, undesirable or even catastrophic results. PMBus device users are urged to contact the PMBus device manufacturer about the consequences of using the RESTORE_DEFAULT_ALL command while the device is operating and providing output power.

This command has one data byte, formatted as an unsigned binary integer.

This command is write only.

11.6. STORE_USER_ALL

The STORE_USER_ALL command instructs the PMBus device to copy the entire contents of the Operating Memory to the matching locations in the non-volatile User Store memory. Any items in Operating Memory that do not have matching locations in the User Store are ignored.

It is permitted to use the STORE_USER_ALL command while the device is operating. However, the device may be unresponsive during the copy operation with unpredictable, undesirable or even catastrophic results. PMBus device users are urged to contact the PMBus device manufacturer about the consequences of using the STORE_USER_ALL command while the device is operating and providing output power.

This command has no data bytes.

This command is write only.

11.7. RESTORE_USER_ALL

The RESTORE_USER_ALL command instructs the PMBus device to copy the entire contents of the non-volatile User Store memory to the matching locations in the Operating Memory. Any items in User Store that do not have matching locations in the Operating Memory are ignored.

It is permitted to use the RESTORE_USER_ALL command while the device is operating. However, the device may be unresponsive during the copy operation with unpredictable, undesirable or even catastrophic results. PMBus device users are urged to contact the PMBus device manufacturer about the consequences of using the RESTORE_USER_ALL command while the device is operating and providing output power.

This command has no data bytes.

This command is write only.

11.8. STORE_USER_CODE

The STORE_USER_CODE command instructs the PMBus device to copy the parameter whose Command Code matches value in the data byte, from the Operating Memory to the matching location in the non-volatile User Store memory.

If the device does not permit saving this parameter in the User Store, or if the device does not support the Command Code specified in the data byte, then the device must notify the host that the command failed, as described in [A01].

It is permitted to use the STORE_USER_CODE command while the device is operating. However, the device may be unresponsive during the copy operation with unpredictable, undesirable or even catastrophic results. PMBus device users are urged to contact the PMBus device manufacturer about the consequences of using the STORE_USER_CODE command while the device is operating and providing output power.

This command has one data byte, formatted as an unsigned binary integer.

This command is write only.

11.9. RESTORE_USER_CODE

The RESTORE_USER_CODE command instructs the parameter whose Command Code matches value in the data byte, from the non-volatile User Store memory to the matching location in the Operating Memory.

If the device does save this parameter in the User Store, or if the device does not support the Command Code specified in the data byte, then the device must notify the host that the command failed, as described in [A01].

It is permitted to use the RESTORE_USER_CODE command while the device is operating. However, the device may be unresponsive during the copy operation with unpredictable, undesirable or even catastrophic results. PMBus device users are urged to contact the PMBus device manufacturer about the consequences of using the RESTORE_USER_ALL command while the device is operating and providing output power.

This command has one data byte, formatted as an unsigned binary integer.

This command is write only.

11.10. PAGE

The page command provides the ability to configure, control and monitor through only one physical address either:

- Multiple outputs on one unit or
- Multiple non-PMBus devices through a PMBus device to non-PMBus device adapter or bridge.

Figure 14 and Figure 15 illustrate these concepts.

Each PAGE contains the Operating Memory (and at the option of the device manufacturer, User Store and Default Store) for each output. Each page may offer the full range of PMBus commands available for each output or non-PMBus device.

PMBus device manufacturers may also use multiple pages within a single PMBus device to offer additional commands or memory space for one or more outputs.

The data byte for the PAGE command is an unsigned binary integer.

Pages 00h through 1Fh are reserved specifically for multiple outputs on a device with a single physical address.

Setting the page to FFh means that all following commands are to applied to all outputs.

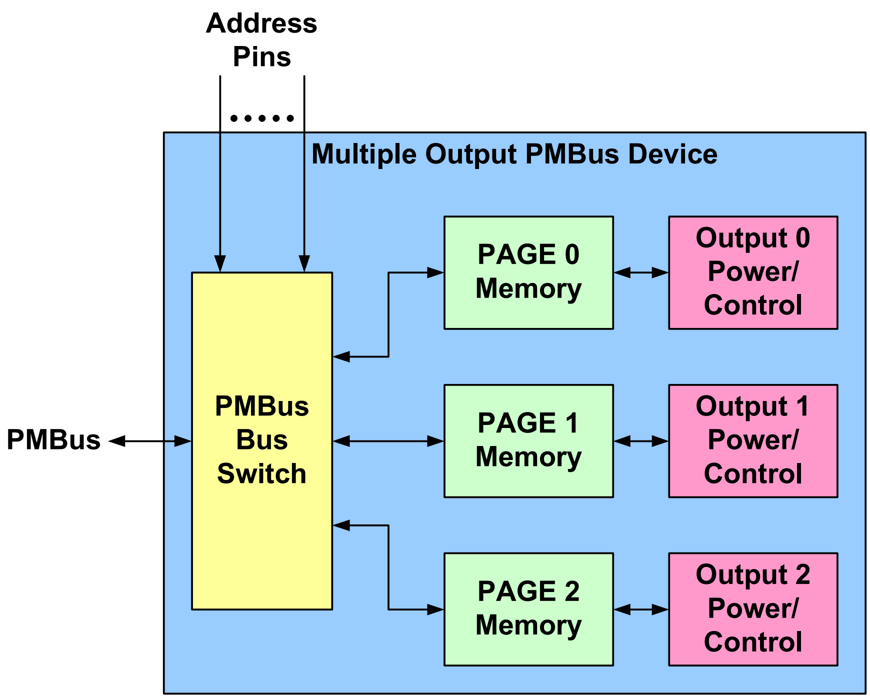

Figure 14. Conceptual View Of Paging Used For A Multiple Output PMBus Device

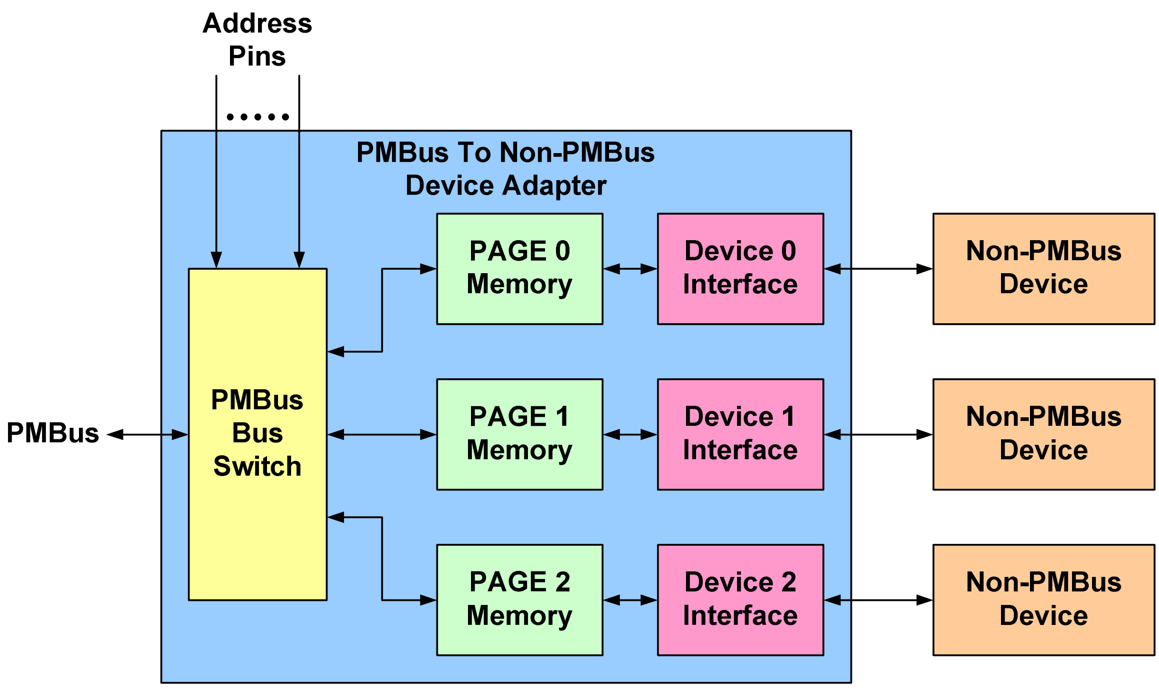

Figure 15. Conceptual View Of Using Paging With A PMBus To Non-PMBus Device Adapter

12. On, Off And Margin Testing Related Commands

12.1. OPERATION

The OPERATION command is used to turn the unit on and off in conjunction with the input from the CONTROL pin. It is also used to cause the unit to set the output voltage to the upper or lower MARGIN VOLTAGEs. The unit stays in the commanded operating mode until a subsequent OPERATION command instructs the device to change to another mode.

The contents of the data byte are shown below in Table 6.

Bits [7:6] determine how the device responds when commanded to turn the output off via the OPERATION command. If bits [7:6] are 00b, then the device turns off immediately and ignores any programmed turn-off delay and fall time. If bits [7:6] are 01b, the device powers down according to the programmed turn-off delay and fall time.

Any value not shown in the table is an invalid command. Bits [1:0] are not used at this time.

When using the SMBus Read Word command to retrieve the current contents of the OPERATION command data bytes, it should be noted that the values retrieved may not match the last programmed values. This might happen, for example, if pin programming is used for margin testing

Table 6. OPERATION Data Byte Contents

Bits [7:6]	Bits [5:6]	Bits [3:2]	Bits [1:0]	Unit On Or Off	Margin State
00	XX	XX	XX	Immediate Off (No Sequencing)	N/A
01	XX	XX	XX	Soft Off (With Sequencing)	N/A
10	00	XX	XX	On	Off
10	01	01	XX	On	Margin Low (Ignore Fault)
10	01	10	XX	On	Margin Low (Act On Fault)
10	10	01	XX	On	Margin High (Ignore Fault)
10	10	10	XX	On	Margin High (Act On Fault)

12.2. ON_OFF_CONFIG

The ON_OFF_CONFIG command configures the combination of CONTROL pin input and serial bus commands needed to turn the unit on and off. This includes how the unit responds when power is applied.

The default response for any PMBus device is specified by the device manufacturer.

The details of the ON_OFF_CONFIG data byte are shown in Table 7

Table 7. ON_OFF_CONFIG Data Byte

Bit Number	Purpose	Bit Value	Meaning
[7:5]		N/A	Reserved For Future Use
4	Sets the default to either operate any time power is present or for the on/off to be controlled by CONTROL pin and serial bus commands	0	Unit powers up any time power is present regardless of state of the CONTROL pin
		1	Unit does not power up until commanded by the CONTROL pin and OPERATION command (as programmed in bits [2:0]).
3	Controls how the unit responds to commands received via the serial bus	0	Unit ignores the on/off portion of the OPERATION command from serial bus
		1	Unit responds to the on/off portion of the OPERATION command. Depending on bit [1], the CONTROL pin may also be required to instruct the device to start before the output is energized.
2	Controls how the unit responds to the CONTROL pin	0	Unit ignores the CONTROL pin (on/off controlled only the OPERATION command)
		1	Unit requires the CONTROL pin to be asserted to start the unit. Depending on bit [2], the OPERATION command may also be required to instruct the device to start before the output is energized.
1	Polarity of the CONTROL pin	0	Active low (Pull pin low to start the unit)
		1	Active high (Pull high to start the unit)
0	CONTROL pin action when commanding the unit to turn off	0	Use the programmed turn off delay (Section 16.5) and fall time (Section 16.6)
		1	Turn off the output and stop transferring energy to the output as fast as possible. The device's product literature shall specify whether or not the device sinks current to decrease the output voltage fall time.

13. Output Voltage Related Commands

13.1. VOUT_MODE

The operation of the VOUT_MODE command is described in Section 8.

13.2. VOUT_COMMAND

The operation of the VOUT_COMMAND command is described in Section 8.

13.3. VOUT_TRIM

The VOUT_TRIM command is used to apply a fixed offset voltage to the output voltage command value. It is most typically used by the end user to trim the output voltage at the time the PMBus device is assembled into the end user's system.

The VOUT_TRIM has two data bytes formatted per the VOUT_MODE command that is in effect. The effect of this command depends on the settings of the VOUT_MODE command (Section 8).

This command may not be used if the unit is working with the VID format for output voltage. Attempts to apply this command when the unit is operating in VID format must be NACK'ed.

The default value is 0000h.

13.4. VOUT_CAL

The VOUT_CAL command is used to apply a fixed offset voltage to the output voltage command value. It is most typically used by the PMBus device manufacturer to calibrate a device in the factory.

The VOUT_CAL has two data bytes formatted to match the VOUT_MODE command. The effect of this command depends on the settings of the VOUT_MODE command (Section 8).

This command may not be used if the unit is working with the VID format for output voltage. Attempts to apply this command when the unit is operating in VID format must be NACK'ed.

The default value is 0000h.

13.5. VOUT_MAX

The VOUT_ MAX command sets an upper limit on the output voltage the unit can command regardless of any other commands or combinations.

If a PMBus device supports this command, it must be able to detect that an attempt has been made to program the output to a voltage in excessive of the value set by the VOUT_MAX command. This will be treated as a warning condition and not a fault condition. If an attempt is made to program the output voltage higher than the limit set by this command, the device shall respond as follows:

- The commanded output voltage shall be set to VOUT_MAX,
- The OTHER bit shall be set in the STATUS_BYTE,
- The VOUT bit shall be set in the STATUS_WORD,
- The VOUT_MAX Warning bit shall be set in the STATUS_VOUT register (Section 17.3), and
- The device notifies the host as described in Section 10.2.1.

The data bytes are two bytes formatted according the setting of the VOUT_MODE command (Section 8).

13.6. VOUT_MARGIN_HIGH

This VOUT_MARGIN_HIGH command loads the unit with the voltage to which the output is to be changed when the OPERATION command is set to "Margin High"

The data bytes are two bytes formatted according the setting of the VOUT_MODE command (Section 8).

13.7. VOUT_MARGIN_LOW

This VOUT_MARGIN_LOW command loads the unit with the voltage to which the output is to be changed when the OPERATION command is set to “Margin Low”

The data bytes are two bytes formatted according the setting of the VOUT_MODE command (Section 8).

13.8. VOUT_TRANSITION_RATE

When a PMBus device receives either a VOUT_COMMAND or OPERATION (Margin High, Margin Low, Margin Off) that causes the output voltage to change, this command sets the rate in mV/µs at which the output should change voltage. This commanded rate of change does not apply then the unit is commanded to turn on or to turn off.

The VOUT_TRANSITION_RATE command has two data bytes formatted either in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The value of FFh FFh indicates that the device should make the transition as quickly as possible.

13.9. VOUT_DROOP

The VOUT_DROOP sets the rate, in mV/A (mΩ) at which the output voltage decreases with increasing current for use with Adaptive Voltage Positioning requirements and passive current sharing schemes.

Each device implements the droop calculation based on its own current with the value with which it has been programmed regardless of whether or not any other units are operating with their outputs in parallel.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 0 mΩ.

13.10. VOLTAGE_SCALE_LOOP

The operation of this command is discussed in Section 9.1.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 1.

13.11. VOLTAGE_SCALE_MONITOR

The operation of this command is discussed in Section 9.1.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 1.

14. Other Commands

14.1. COEFFICIENTS

The COEFFICIENTS command is used to retrieve the m, b and R coefficients needed by data in the DIRECT format.

This command uses the Block Read/Write Process Call as described in [A01]. This command has two data bytes, each of which is an unsigned binary integer. More information on the function and application of this command are given in Section 7.2.

14.2. POUT_MAX

The POUT_MAX commands set the output power, in watts, at which the unit starts regulating in constant power mode instead of constant voltage. This command is typically used in systems that charge batteries.

The POUT_MAX command has two data bytes formatted either in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the manufacturer.

14.3. MAX_DUTY

The MAX_DUTY command sets the maximum duty cycle of the unit's power conversion stage.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

14.4. FREQUENCY_SWITCH

The FREQUENCY_SWITCH command sets the switching frequency, in kHz, of a PMBus device.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

14.5. VIN_ON

The VIN_ON command sets the value of the input voltage, in volts, at which the unit should start power conversion.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

14.6. VIN_OFF

The VIN_OFF command sets the value of the input voltage, in volts, at which the unit, once operation has started, should stop power conversion.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

14.7. INTERLEAVE

The INTERLEAVE command is used to arrange multiple units so that their switching periods can be distributed in time. This may be used to facilitate paralleling of multiple units or to reduce ac currents injected into the power bus.

To get best advantage from setting the interleave, the units should have their switching frequency clocks well synchronized.

The INTERLEAVE command data bytes include three pieces of information:

- A group identification number (4 bits),
- The number of units in the group (4 bits) and
- The interleave order for this particular unit (4 bits). This number ranges in value from zero to one less than the number of units in the group.

The group identification number allows for up to fifteen groups. Group Identification Number 0 is reserved to mean not a member of an interleaved group. If the group identification number is 0, then the number of units in the group and the interleave order shall also be 0.

The format of the data bytes is shown below in Table 8.

Table 8. INTERLEAVE Data Bytes Format

Byte	High Byte								Low Byte							
Bit Number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Contents	Not Used				Group ID Number				Number In Group				Interleave Order			
Default Value	00				00				00				00			

An example of the function of the INTERLEAVE command is shown in Figure 16. In this example, there are four converters in Group Number 9. The first converter, UNIT 1, is assigned Interleave Order 0; Unit 2 is assigned Interleave Order 1 and so forth. Unit 1, with interleave order, starts its switching cycle at when the Master Clock (not defined by the PMBus protocol), starts a new switching cycle. Unit 2, second in the interleave order, starts its on time after a delay of one quarter of the Master Clock period. The one quarter cycle delay for Unit 2 is calculated as:

$$Tdelay(Unit\ 2) = \frac{Interleave\ Order\ Of\ Unit\ 2}{Number\ In\ Group} \bullet T_S = \frac{0001b}{0100b} \bullet T_S = \frac{1}{4} \bullet T_S$$

In general, for Unit N, the delay time from the triggering edge of the Master Clock to the start of Unit N's one time is:

$$Tdelay(Unit\ N) = \frac{Interleave\ Order\ Of\ Unit\ N}{Number\ In\ Group} \bullet T_S$$

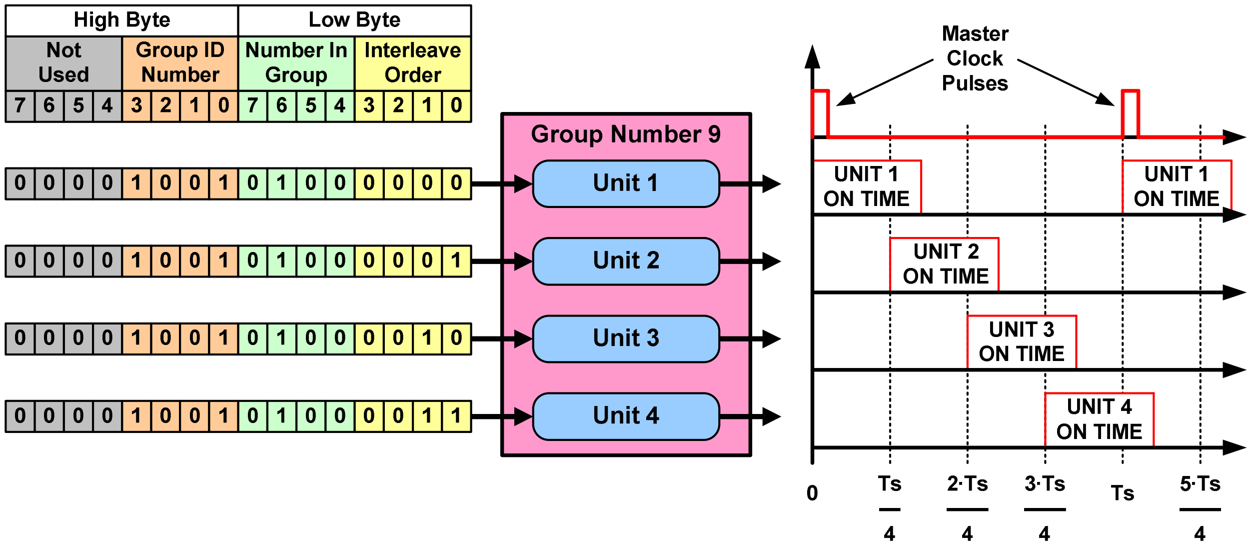

Figure 16. Illustration Of The INTERLEAVE Command Function

14.8. IOUT_SCALE

The IOUT_SCALE command is used to set the ratio of the voltage at the current sense pins to the sensed current. For devices using a fixed current sense resistor, it is the same value as the resistance of the resistor.

This command may also be used to calibrate the device's current sensing circuit. For example, automatic test equipment could load a device to a precisely know output current. It would the use command READ_IOUT command to determine what current the device is reporting. A second measurement at a different load current would also typically be taken. Using the known currents drawn by the test equipment and the two currents resorted by the device, the test equipment can then calculate the best values of IOUT_SCALE and IOUT_CAL_OFFSET (below) to minimize the error in the current sensing circuit.

The units of the IOUT SCALE factor are mΩ.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 0 mΩ.

14.9. IOUT_CAL_OFFSET

The IOUT_CAL_OFFSET is used to null out any offsets in the output current sensing circuit. This command is most often used in conjunction with the IOUT_SCALE command (above) to minimize the error of the current sensing circuit.

The units of the IOUT_CAL_OFFSET are amperes.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 0 amperes.

14.10. VFAN_1

The VFAN_1 command is used to set the voltage, in volts, on a fan in the PMBus unit.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the device manufacturer in the product literature.

14.11. VFAN_2

The VFAN_2 command is used to set the voltage, in volts, on a fan in the PMBus unit.

This command has two data bytes formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the device manufacturer in the product literature.

15. Fault Related Commands

15.1. CLEAR_FAULTS

The CLEAR_FAULTS command is used to clear any fault bits that have been set. This command clears all bits in all status registers simultaneously.

The CLEAR_FAULTS does not cause a unit that has latched off for a fault condition to restart. Units that have shut down for a fault condition are restarted as described in Section 10.7.

If the fault is still present when the bit is cleared, the fault bit shall immediately be set again and the host notified by the usual means.

This command is write only.

15.2. VOUT_OV_FAULT_LIMIT

The VOUT_OV_FAULT_LIMIT command sets the value of the output voltage measured at the sense or output pins that causes an output overvoltage fault.

The data bytes are two bytes formatted according to the setting of the VOUT_MODE command (Section 8).

The default value is specified by the device manufacturer in the product literature.

15.3. VOUT_OV_FAULT_RESPONSE

The VOUT_OV_FAULT_RESPONSE command instructs the device on what action to take in response to an output overvoltage fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the VOUT_OV bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD,
- Sets the VOUT Overvoltage Fault bit in the STATUS_VOUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

The default value is specified by the device manufacturer in the product literature.

15.4. VOUT_OV_WARN_LIMIT

The VOUT_OV_WARN_LIMIT command sets the value of the output voltage at the sense or output pins that causes an output voltage high warning. This value is typically less than the output overvoltage threshold.

The data bytes are two bytes formatted according the setting of the VOUT_MODE command (Section 8).

In response to the VOUT_OV_WARN_LIMIT being exceeded, the device:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD,
- Sets the VOUT Overvoltage Warning bit in the STATUS_VOUT register, and
- Notifies the host as described in Section 10.2.1.

The default value is specified by the device manufacturer in the product literature.

15.5. VOUT_UV_WARN_LIMIT

The VOUT_UV_WARN_LIMIT command sets the value of the output voltage at the sense or output pins that causes an output voltage low warning. This value is typically greater than the output undervoltage fault threshold.

This fault is masked until the unit reaches the programmed output voltage. This fault is also masked when the unit is disabled.

The data bytes are two bytes formatted according to the setting of the VOUT_MODE command (Section 8).

In response to the VOUT_UV_WARN_LIMIT being exceeded, the device:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD,
- Sets the VOUT Undervoltage Warning bit in the STATUS_VOUT register, and
- Notifies the host as described in Section 10.2.1.

The default value is specified by the device manufacturer in the product literature.

15.6. VOUT_UV_FAULT_LIMIT

The VOUT_UV_FAULT_LIMIT command sets the value of the output voltage at the sense or output pins that causes an output undervoltage fault.

This fault is masked until the unit reaches the programmed output voltage. This fault is also masked when the unit is disabled.

The data bytes are two bytes formatted according the setting of the VOUT_MODE command (Section 8).

The default value is 00h 00h.

15.7. VOUT_UV_FAULT_RESPONSE

The VOUT_UV_FAULT_RESPONSE command instructs the device on what action to take in response to an output undervoltage fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD,
- Sets the VOUT Undervoltage Fault bit in the STATUS_VOUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

The default value is specified by the device manufacturer in the product literature.

15.8. IOUT_OC_FAULT_LIMIT

The IOUT_OC_FAULT_LIMIT command sets the value of the output current, in amperes, that causes the overcurrent detector to indicate an overcurrent fault condition.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the device manufacturer in the product literature.

15.9. IOUT_OC_FAULT_RESPONSE

The IOUT_OC_FAULT_RESPONSE command instructs the device on what action to take in response to an output overcurrent fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the IOUT_OC bit in the STATUS_BYTE,
- Sets the IOUT bit in the STATUS_WORD,
- Sets the IOUT Overcurrent Fault bit in the STATUS_IOUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

The default value is specified by the device manufacturer in the product literature.

15.10. IOUT_OC_LV_FAULT_LIMIT

In the case where the response to an overcurrent condition is to operate in a constant current mode unless the output voltage is pulled below the specified value, the IOUT_OC_LV_FAULT_LIMIT specifies that voltage threshold.

The data bytes are two bytes formatted according the setting of the VOUT_MODE command (Section 8).

The default value is 00h 00h.

15.11. IOUT_OC_LV_FAULT_RESPONSE

The IOUT_OC_LV_FAULT_RESPONSE command instructs the device on what action to take in response to an output overcurrent fault when the output voltage has been pulled below the specified threshold. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the IOUT_OC bit in the STATUS_BYTE,
- Sets the IOUT bit in the STATUS_WORD,
- Sets the IOUT Overcurrent And Low Voltage Fault bit in the STATUS_IOUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

The default value is specified by the device manufacturer in the product literature.

15.12. IOUT_OC_WARN_LIMIT

The IOUT_OV_WARN_LIMIT command sets the value of the output current that causes an output overcurrent warning.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

In response to the IOUT_OC_WARN_LIMIT being exceeded, the device:

- Sets the OTHER bit in the STATUS_BYTE
- Sets the IOUT bit in the STATUS_WORD,
- Sets the IOUT Overcurrent Warning bit in the STATUS_IOUT register, and
- Notifies the host as described in Section 10.2.1.

The default value is specified by the device manufacturer in the product literature.

15.13. IOUT_UC_FAULT_LIMIT

For units with a synchronous rectifier in the output, current can flow from the unit to the load or from the load into the output. When current is flowing from the unit to the load the unit is said to be sourcing current and the output current declared to be positive. When current is flowing into the unit from the load, the units is said to be sinking current and the current is declared to be negative.

This command sets the maximum amount of sink current, in amperes, that is allowed before action is taken.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 0 A.

15.14. IOUT_UC_FAULT_RESPONSE

The IOUT_OC_FAULT_RESPONSE command instructs the device on what action to take in response to an output overcurrent fault. The data byte is in the format given in Section 10.5.1.

For this fault condition, the Inhibit Operation option refers only to stopping the synchronous rectification (allowing the output current to freewheel through the freewheel device) and not to turning off the output (stopping the transfer of energy to the output).

The device also:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the IOUT bit in the STATUS_WORD,
- Sets the IOUT Undercurrent Fault bit in the STATUS_IOUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device’s product literature.

The default value is specified by the device manufacturer in the product literature.

15.15. POUT_FAULT_LIMIT

This fault limit will be described in a future release of this specification.

15.16. POUT_FAULT_RESPONSE

This fault response will be described in a future release of this specification.

15.17. OT_FAULT_LIMIT

The OT_FAULT_LIMIT command set the temperature, in degrees Celsius, of the unit at which it should indicate an Overtemperature Fault.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the device manufacturer in the product literature.

15.18. OT_FAULT_RESPONSE

The OT_FAULT_RESPONSE command instructs the device on what action to take in response to an overtemperature fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the TEMPERATURE bit in the STATUS_BYTE,
- Sets the Overtemperature Fault bit in the STATUS_TEMPERATURE register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device’s product literature.

The default value is specified by the device manufacturer in the product literature.

15.19. OT_WARN_LIMIT

The OT_WARN_LIMIT command set the temperature, in degrees Celsius, of the unit at which it should indicate an Overtemperature Warning alarm.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

In response to the OT_WARN_LIMIT being exceeded, the device:

- Sets the TEMPERATURE bit in the STATUS_BYTE,
- Sets the Overtemperature Warning bit in the STATUS_TEMPERATURE register, and
- Notifies the host as described in Section 10.2.1.

The default value is specified by the device manufacturer in the product literature.

15.20. UT_WARN_LIMIT

The UT_WARN_LIMIT command set the temperature, in degrees Celsius, of the unit at which it should indicate an Undertemperature Warning alarm.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

In response to the UT_WARN_LIMIT being exceeded, the device:

- Sets the TEMPERATURE bit in the STATUS_BYTE,
- Sets the Undertemperature Warning bit in the STATUS_TEMPERATURE register, and
- Notifies the host as described in Section 10.2.1.

The default value is specified by the device manufacturer in the product literature.

15.21. UT_FAULT_LIMIT

The UT_FAULT_LIMIT command sets the temperature, in degrees Celsius, of the unit at which it should indicate an Undertemperature Fault.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the device manufacturer in the product literature.

15.22. UT_FAULT_RESPONSE

The UT_FAULT_RESPONSE command instructs the device on what action to take in response to an undertemperature fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the TEMPERATURE bit in the STATUS_BYTE,
- Sets the Undertemperature Fault bit in the STATUS_TEMPERATURE register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

The default value is specified by the device manufacturer in the product literature.

15.23. VIN_OV_FAULT_LIMIT

The VIN_OV_FAULT_LIMIT command sets the value of the input voltage that causes an input overvoltage fault.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the device manufacturer in the product literature.

15.24. VIN_OV_FAULT_RESPONSE

The VIN_OV_FAULT_RESPONSE command instructs the device on what action to take in response to an input overvoltage fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the OTHER bit in the STATUS_BYTE,
- Set the INPUT bit in the upper byte of the STATUS_WORD,
- Sets the VIN Overvoltage Fault bit in the STATUS_INPUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

The default value is specified by the device manufacturer in the product literature.

15.25. VIN_OV_WARN_LIMIT

The VIN_OV_WARN_LIMIT command sets the value of the input voltage that causes an input voltage high warning. This value is typically less than the input overvoltage fault threshold.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

In response to the VIN_OV_WARN_LIMIT being exceeded, the device:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the INPUT bit is the upper byte of the STATUS_WORD,
- Sets the VIN Overvoltage Warning bit in the STATUS_INPUT register, and
- Notifies the host as described in Section 10.2.1.

The default value is specified by the device manufacturer in the product literature.

15.26. VIN_UV_WARN_LIMIT

The VIN_UV_WARN_LIMIT command sets the value of the input voltage that causes an input voltage low warning. This value is typically greater than the input undervoltage fault threshold.

This alarm is masked until the input exceeds the input startup threshold and the unit has been enabled.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

In response to the VIN_UV_WARN_LIMIT being exceeded, the device:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the INPUT bit is the upper byte of the STATUS_WORD,
- Sets the VIN Undervoltage Warning bit in the STATUS_INPUT register, and
- Notifies the host as described in Section 10.2.1.

The default value is specified by the device manufacturer in the product literature.

15.27. VIN_UV_FAULT_LIMIT

The VIN_UV_FAULT_LIMIT command sets the value of the input voltage that causes an input undervoltage fault.

This alarm is masked until the input exceeds the input startup threshold and the unit has been enabled.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the device manufacturer in the product literature.

15.28. VIN_UV_FAULT_RESPONSE

The VIN_UV_FAULT_RESPONSE command instructs the device on what action to take in response to an input undervoltage fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the VIN_UV bit in the STATUS_BYTE,
- Sets the INPUT bit is the upper byte of the STATUS_WORD,
- Sets the VIN Undervoltage Fault bit in the STATUS_INPUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

15.29. IIN_OC_FAULT_LIMIT

The IIN_OC_FAULT_LIMIT command sets the value of the input current that causes an input overcurrent fault.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is specified by the device manufacturer in the product literature.

15.30. IIN_OC_FAULT_RESPONSE

The IIN_OC_FAULT_RESPONSE command instructs the device on what action to take in response to an input overcurrent fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the INPUT bit is the upper byte of the STATUS_WORD,
- Sets the IIN Overvoltage Fault bit in the STATUS_INPUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

The default value is specified by the device manufacturer in the product literature.

15.31. IIN_OC_WARN_LIMIT

The IIN_OC_WARN_LIMIT command sets the value of the input current that causes an input current high warning.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

In response to the IIN_OC_WARN_LIMIT being exceeded, the device:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the INPUT bit is the upper byte of the STATUS_WORD,
- Sets the IIN Overcurrent Warning bit in the STATUS_INPUT register, and
- Notifies the host as described in Section 10.2.1.

The default value is specified by the device manufacturer in the product literature.

15.32. POWER_GOOD Signal Limits

For PMBus devices that offer a POWER_GOOD signal, these commands are used for setting the output voltage at which a power good signal should be asserted and negated.

Power Good signals will be device and manufacturer specific. Many factors other than output voltage may be used to determine whether or not the POWER_GOOD is to be asserted. PMBus device users are instructed to consult the device manufacturer's product literature for the specifics of the device they are using.

15.32.1. POWER_GOOD_ON

The POWER_GOOD_ON command sets the output voltage at which an optional POWER_GOOD signal should be asserted.

The data bytes are two bytes formatted according the setting of the VOUT_MODE command (Section 8).

The default value is specified by the device manufacturer in the product literature.

15.32.2. POWER_GOOD_OFF

The POWER_GOOD_OFF command sets the output voltage at which an optional POWER_GOOD signal should be negated.

The data bytes are two bytes formatted according the setting of the VOUT_MODE command (Section 8).

The default value is specified by the device manufacturer in the product literature.

16. Output Voltage Sequencing Commands

16.1. TON_DELAY

The TON_DELAY sets the time, in ms, from when a start condition is received (CONTROL signal asserted and a valid OPERATION command received) until the output voltage starts to rise.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 0 ms.

16.2. TON_RISE

The TON_RISE sets the time, in ms, from when the output starts to rise until the voltage has entered the regulation band.

A value of 0 ms instructs the unit to bring its output voltage to the programmed regulation value as quickly as possible.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 0 ms.

16.3. TON_MAX_FAULT_LIMIT

The TON_MAX_FAULT_LIMIT command sets an upper limit, in ms, on how long the unit can attempt to power up the output without reaching the output undervoltage fault limit (Section 0).

A value of 0 ms means that there is no limit and that the unit can attempt to bring up the output voltage indefinitely.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value of the data bytes is 0 ms.

16.4. TON_MAX_FAULT_RESPONSE

The TON_MAX_FAULT_RESPONSE command instructs the device on what action to take in response to an TON_MAX fault. The data byte is in the format given in Section 10.5.1.

The device also:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD,
- Sets the TON_MAX_FAULT… bit in the STATUS_VOUT register, and
- Notifies the host as described in Section 10.2.2.

The delay time unit is specified by the device manufacturer in the device's product literature.

The default value is specified by the device manufacturer in the product literature.

16.5. TOFF_DELAY

The TOFF_DELAY sets the time, in ms, from a stop condition is received (CONTROL signal negated or a valid OPERATION command received) until the unit stops transferring energy to the output.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 0 ms.

16.6. TOFF_FALL

The TOFF_FALL sets the time, in ms, from the end of the turn-off delay time (Section 16.5) until the voltage is commanded to zero. Note that this command can only be used with a device whose output can sink enough current to cause the output voltage to decrease at a controlled rate.

A value of 00h 00h means that the device should ramp the output voltage down as fast as it can without exceeding the IOUT_UC_FAULT_LIMIT current (Section 15.13).

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value is 00h 00h.

16.7. TOFF_MAX_FAULT_LIMIT

The TON_MAX_FAULT_LIMIT command sets an upper limit on how long the unit can attempt to power down the output without reaching 12.5% of the output voltage programmed at the time the converter is turned off. (Section 16.6).

A value of 7Fh FFh means that there is no limit and that the unit waits indefinitely for the output voltage to decay.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

The default value of the data bytes is FFh FFh.

16.8. TOFF_MAX_FAULT_RESPONSE

In response to the TOFF_MAX_FAULT_LIMIT being exceeded, the device:

- Sets the OTHER bit in the STATUS_BYTE,
- Sets the VOUT bit is the upper byte of the STATUS_WORD,
- Sets the TOFF_MAX_FAULT… bit in the STATUS_VOUT register, and
- Notifies the host as described in Section 10.2.1.

17. Unit Status Commands

This section describes commands to retrieve status information from PMBus units. Status information is binary.

A value of 1 indicates a fault or warning event has occurred and a value of 0 indicates that a fault or warning event has not occurred.

All of the commands are read only.

Bits for unsupported features shall be reported as zero.

17.1. STATUS_BYTE

The STATUS_BYTE command returns one byte of information with a summary of the most critical faults.

The STATUS_BYTE message content is described in Table 9

Table 9. STATUS_BYTE Message Contents

Bit Number	Status Bit Name	Meaning
7	BUSY	The device is busy.
6	OFF	This bit is asserted if the unit is not providing power to the output, regardless of the reason, including simply not being enabled.
5	VOUT_OV	An output overvoltage fault has occurred
4	IOUT_OC	An output overcurrent fault has occurred
3	VIN_UV	An input undervoltage fault has occurred
2	TEMPERATURE	A temperature fault or warning has occurred
1	CML	A communications, memory or logic fault has occurred
0	OTHER	A fault or warning not listed in bits [7:1] has occurred

17.2. STATUS_WORD

The STATUS_WORD command returns two bytes of information with a summary of the unit's fault condition. Based on the information in these bytes, the host can get more information by reading the appropriate status registers.

The low byte of the STATUS_WORD is the same register as the STATUS_BYTE command.

The STATUS_WORD message content is described in Table 10.

Table 10. STATUS_WORD Message Contents

Byte	Bit Number	Status Bit Name	Meaning
Low	7	BUSY	The device is busy.
	6	OFF	This bit is asserted if the unit is not providing power to the output, regardless of the reason, including simply not being enabled.
	5	VOUT_OV	An output overvoltage fault has occurred
	4	IOUT_OC	An output overcurrent fault has occurred
	3	VIN_UV	An input undervoltage fault has occurred
	2	TEMPER-ATURE	A temperature fault or warning has occurred
	1	CML	A communications, memory or logic fault has occurred
	0	OTHER	A fault or warning not listed in bits [7:1] has occurred
High	7	VOUT	An output voltage fault or warning has occurred
	6	IOUT	An output current fault or warning has occurred
	5	INPUT	An input voltage or input current fault or warning has occurred
	4	MFR	A manufacturer specific fault or warning has occurred
	3	POWER_ GOOD#	The POWER_GOOD signal, if present, is negated
	2	Reserved	Reserved for future user
	1	Reserved	Reserved for future user
	0	UNKNOWN	A fault type not given in other bits has been detected

17.3. STATUS_VOUT

The STATUS_VOUT command returns one data byte with contents as follows:

Table 11. STATUS_VOUT Data Byte

Bit	Meaning
7	VOUT Overvoltage Fault
6	VOUT Overvoltage Warning
5	VOUT Undervoltage Warning
4	VOUT Undervoltage Fault
3	VOUT_MAX Warning (An attempt has been made to set the output voltage to value higher than allowed by the VOUT_MAX command (Section 13.5).

Bit	Meaning
2	TON_MAX_FAULT or VOUT Tracking Error On Power-Up (Manufacturer Defined)
1	TOFF_MAX_FAULT or VOUT Tracking Error On Power-Down (Manufacturer Defined)
0	Reserved

17.4. STATUS_IOUT

The STATUS_IOUT command returns one data byte with contents as follows:

Table 12. STATUS_IOUT Data Byte

Bit	Meaning
7	IOUT Overcurrent Fault
6	IOUT Overcurrent And Low Voltage Shutdown Fault
5	IOUT Overcurrent Warning
4	IOUT Undercurrent Fault
3	Current Share Fault (Manufacturer Defined)
2	Power Limiting – Unit is operating with the output in constant power mode at the power set by the POUT_MAX command (Section 14.2).
1	Reserved
0	Reserved

17.5. STATUS_INPUT

The STATUS_INPUT command returns one data byte with contents as follows:

Table 13. STATUS_INPUT Data Byte

Bit	Meaning
7	VIN Overvoltage Fault
6	VIN Overvoltage Warning
5	VIN Undervoltage Warning
4	VIN Undervoltage Fault
3	Unit Is Off For Insufficient Input Voltage [1]
2	IIN Overcurrent Fault
1	IIN Overcurrent Warning
0	Reserved

[1] Either the input voltage has never exceeded the input turn-on threshold (Section 0)or if the unit did start, the input voltage decreased below the turn-off threshold (Section 0).

17.6. STATUS_TEMPERATURE

The STATUS_TEMPERATURE command returns one data byte with contents as follows:

Table 14. TEMPERATURE_FAULT_STATUS Data Byte

Bit	Meaning
7	Overtemperature Fault
6	Overtemperature Warning
5	Undertemperature Warning
4	Undertemperature Fault
3	Reserved
2	Reserved
1	Reserved
0	Reserved

17.7. STATUS_CML (Communications, Logic, And Memory)

The STATUS_CML command returns one data byte with contents as follows:

Table 15. CML_FAULT_STATUS Data Byte

Bit	Meaning
7	Invalid Or Unsupported Command Received
6	Invalid Or Unsupported Data Received
5	Packet Error Check Failed
4	Memory Fault Detected
3	Processor Fault Detected
2	Reserved
1	A communication fault other than the ones listed in this table has occurred
0	A fault whose type cannot be determined has occurred.

17.8. STATUS_OTHER

The STATUS_OTHER command returns one data byte with contents as follows:

Table 16. STATUS_OTHER Data Byte

Bit	Meaning
7	Fan Fault 1
6	Fan Fault 2
5	Input Fuse Or Circuit Breaker Fault A
4	Input Fuse Or Circuit Breaker Fault B
3	Input OR-ing Device A Fault

Bit	Meaning
2	Input OR-ing Device B Fault
1	Output OR-ing Device Fault
0	Unspecified Or Unknown Fault

17.9. STATUS_MFR_SPECIFIC

The STATUS_MFR_SPECIFIC command returns one data byte with contents as follows:

Table 17. STATUS_MFR_SPECIFIC Data Byte

Bit	Meaning
7	Manufacturer Defined
6	Manufacturer Defined
5	Manufacturer Defined
4	Manufacturer Defined
3	Manufacturer Defined
2	Manufacturer Defined
1	Manufacturer Defined
0	Manufacturer Defined

18. Reading Parametric Information

The READ commands allow the host to read various parameters of the PMBus device. These commands are read only.

18.1. READ_VIN

The READ_VIN command returns the input voltage in volts.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

18.2. READ_IIN

The READ_IIN command returns the input current in amperes.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

18.3. READ_VCAP

The READ_VCAP command returns voltage on the energy storage (hold-up or ride-through) capacitor in volts.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

18.4. READ_VOUT

The READ_VOUT command returns the actual, measured (not commanded) output voltage in the same format as set by the VOUT_MODE command.

If the VOUT_MODE is set for Linear or Direct format, the returned value is in volts. If the VOUT_MODE is set to VID format, then the returned value is the VID code corresponding to the voltage closest to the measured voltage.

18.5. READ_IOUT

The READ_IOUT command returns the measured output current in amperes.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

18.6. READ_TEMPERATURE

Up to three temperature readings can be returned for each device. The device's product literature shall describe the temperature being measured. For example, an ac-dc power supply might return the temperature of a critical heatsink and the temperature of the inlet cooling air.

The three commands for reading temperature are:

- READ_TEMPERATURE_1,
- READ_TEMPERATURE_2, and
- READ_TEMPERATURE_3.

Each returns the temperature in degree Celsius. The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

18.7. READ_FAN_SPEED

Up to two fan speed readings can be returned for each device. The two commands for reading fan speed are:

- READ_FAN_SPEED_1 and
- READ_FAN_SPEED_2.

The value returned is in RPM.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

18.8. READ_VFAN

Up to two fan voltages can be returned for each device. The two commands for reading fan voltage are:

- READ_VFAN_1 and
- READ_VFAN_2.

The value returned is in volts.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

18.9. READ_DUTY_CYCLE

The READ_DUTY_CYCLE command returns the duty of the PMBus device's main power converter in percent.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

18.10. READ_FREQUENCY

The READ_FREQUENCY command returns the switching frequency of the PMBus device's main power converter in kilohertz. This command returns the actual switching frequency and not the commanded switching frequency.

The two data bytes are formatted in the Literal Data format (Section 7.1) or in the DIRECT format (Section 7.2). The PMBus device product literature shall clearly state which format the device uses.

19. Reserved

This section number is reserved for future use.

20. Reserved

This section number is reserved for future use.

21. Reserved

This section number is reserved for future use.

22. Manufacturer's Information

22.1. PMBUS_REVISION

PMBUS_REVISION command stores or reads the revision of the PMBus to which the device is compliant.

The command has one data byte. Bits [7:5] indicate the revision of PMBus specification Part I to which the device is compliant. Bits [4:0] indicate the revision of PMBus specification Part II to which the device is compliant. The permissible values are shown in Table 18.

Devices may support this as a read only command.

Table 18. PMBus Revision Data Byte Contents

Bits [7:5]	Part I Revision		Bits [4:0]	Part II Revision
000	1.0		0000	1.0

22.2. Inventory Information

The PMBus protocol provides commands for the storage and retrieval of the device manufacturer's inventory information. This is more typically the manufacturer of an assembled power supply or dc-dc converter than an IC manufacturer.

The length of data for type of inventory information varies from manufacturer to manufacturer so the length of the data for each type is not specified. Instead, if a PMBus device supports manufacturer's inventory information, the device's product literature will state the total space available, in bytes, for all inventory information.

SMBus block write and block read commands ([A02], Section 7.5.7) are used to write and retrieve inventory information. The block write and read commands require that the first data byte be the number of bytes to follow (Byte Count). The bytes used for the byte count take up space in the available memory. For example, supposed the MFR_ID is six bytes. The manufacturer sends the number 6 (the Byte Count) plus six bytes of data, for a total of seven bytes. If the available memory was 128 bytes before the MFR_ID is loaded, then 121 bytes are available after.

Manufacturer's inventory information is always loaded using one byte ASCII characters.

22.2.1. MFR_ID

The MFR_ID commands loads the unit with ASCII characters that contain the manufacturer's ID (name, abbreviation or symbol that identifies the unit's manufacturer). This is typically only done once at the time of manufacture.

22.2.2. MFR_MODEL

The MFR_MODEL command loads the unit with ASCII characters that contain the manufacturer's model number. This is typically done once at the time of manufacture.

22.2.3. MFR_REVISION

The MFR_REVISION command loads the unit with ASCII characters that contain the manufacturer's revision number. This is typically done once at the time of manufacture.

22.2.4. MFR_LOCATION

The MFR_LOCATION command loads the unit with ASCII characters that identify the facility that manufactured the unit. This is typically done once at the time of manufacture.

22.2.5. MFR_DATE

The MFR_DATE command loads the unit with ASCII characters that identify the unit's date of manufacture. This is typically done once at the time of manufacture.

The recommended MFR_DATE format is YYMMDD where Y, M and D are integer values from 0 to 9, inclusive.

22.2.6. MFR_SERIAL

The MFR_SERIAL command loads the unit with ASCII characters to uniquely identify the unit (serial number). This is typically done at the time of manufacture.

22.3. Manufacturer Ratings

The following commands provide the ability for manufacturers to provide summary information about the unit's ratings. This information serves as an electronic nameplate for the user's convenience. The PMBus device does not enforce or report as a warning or a fault any violation of these ratings.

Each of the Manufacturer's Ratings commands has two data bytes in either the Literal format (Section 7.1) or the DIRECT format (Section 7.2). The PMBus device's product literature shall clearly state which format the device supports.

22.3.1. MFR_VIN_MIN

The MFR_VIN_MIN command sets or retrieves the minimum rated value, in volts, of the input voltage.

22.3.2. MFR_VIN_MAX

The MFR_VIN_MAX command sets or retrieves the maximum rated value, in volts, of the input voltage.

22.3.3. MFR_IIN_MAX

The MFR_IIN_MIN command sets or retrieves the maximum rated value, in amperes, of the input current.

22.3.4. MFR_PIN_MAX

The MFR_PIN_MIN command sets or retrieves the maximum rated value, in watts, of the input power.

22.3.5. MFR_VOUT_MIN

The MFR_VOUT_MIN command sets or retrieves the minimum rated value, in volts, to which the output voltage may be set.

22.3.6. MFR_VOUT_MAX

The MFR_VOUT_MAX command sets or retrieves the maximum rated value, in volts, to which the output voltage may be set.

22.3.7. MFR_IOUT_MAX

The MFR_IOUT_MAX command sets or retrieves the maximum rated value, in amperes, to which the output may be loaded.

22.3.8. MFR_POUT_MAX

The MFR_POUT_MAX command sets or retrieves the maximum rated output power, in watts, that the unit is rated to supply.

22.3.9. MFR_TAMBIENT_MAX

The MFR_TAMBIENT_MAX command sets or retrieves the maximum rated ambient temperature, in degrees Celsius, in which the unit may be operated.

22.3.10. MFR_TAMBIENT_MIN

The MFR_TAMBIENT MIN command sets or retrieves the minimum rated ambient temperature, in degrees Celsius, in which the unit may be operated.

23. User Data And Configuration

The PMBus protocol reserves 16 commands for PMBus device manufacturers to provide memory for their customers to store information. These commands, for example, could be used to store end user specific inventory information or configuration information such as digital control loop coefficients.

The names of the commands are USER_DATA_00 through USER_DATA_15.

Each of these commands may use the block write and block read to store and retrieve up to 255 bytes of data for each command for a maximum possible User Data storage of 4,080 byes.

The PMBus device's product literature shall describe the manufacturer's implementation of these commands.

24. Manufacturer Specific Commands

The PMBus protocol reserves 46 command codes for manufacturer specific commands. These commands will be unique to a particular device or manufacturer and allow for unique or proprietary extensions to the PMBus protocol.

The names of the commands are MFR_SPECIFIC_00 through MFR_SPECIFIC_45.

The PMBus device's product literature shall describe the manufacturer's implementation of these commands.

25. Command Extensions

25.1. MFR_SPECIFIC_COMMAND_EXT

The MFR_SPECIFIC_COMMAND_EXT is used to allow PMBus device manufacturers to extend the command set beyond the available 256 command codes.

This command uses the Extended Command: Read/Write Byte or Extended Command: Read/Write Word protocols described in [A01].

25.2. PMBUS_COMMAND_EXT

The PMBUS_COMMAND_EXT is reserved for future use to extend the PMBus command set beyond the available 256 command codes.

This command uses the Extended Command: Read/Write Byte or Extended Command: Read/Write Word protocols described in [A01].

APPENDIX I. Command Summary

Any command codes not used in Table 19 are reserved for future use.

Table 19. Command Summary

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
00h	PAGE	R/W Byte	1		
01h	OPERATION	R/W Byte	1		
02h	ON_OFF_CONFIG	R/W Byte	1		
03h	CLEAR_FAULTS	Send Byte	0		
04h	Reserved				
05h	Reserved				
06h	Reserved				
07h	Reserved				
08h	Reserved				
09h	Reserved				
0Ah	Reserved				
0Bh	Reserved				
0Ch	Reserved				
0Dh	Reserved				
0Eh	Reserved				
0Fh	Reserved				
10h	WRITE_PROTECT	R/W Byte	1		
11h	STORE_DEFAULT_ALL	Send Byte	0		
12h	RESTORE_DEFAULT_ALL	Write Byte	1		
13h	STORE_DEFAULT_CODE	Send Byte	0		
14h	RESTORE_DEFAULT_CODE	Write Byte	1		
15h	STORE_USER_ALL	Send Byte	0		
16h	RESTORE_USER_ALL	Write Byte	1		
17h	STORE_USER_CODE	Send Byte	0		
18h	RESTORE_USER_CODE	Write Byte	1		
19h	Reserved				
1Ah	Reserved				
1Bh	Reserved				

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
1Ch	Reserved				
1Dh	Reserved				
1Eh	Reserved				
1Fh	Reserved				
20h	VOUT_MODE	R/W Byte	1		
21h	VOUT_COMMAND	R/W Word	2		
22h	VOUT_TRIM	R/W Word	2		
23h	VOUT_CAL	R/W Word	2		
24h	VOUT_MAX	R/W Word	2		
25h	VOUT_MARGIN_HIGH	R/W Word	2		
26h	VOUT_MARGIN_LOW	R/W Word	2		
27h	VOUT_TRANSITION_RATE	R/W Word	2		
28h	VOUT_DROOP	R/W Word	2		
29h	VOLTAGE_SCALE_LOOP	R/W Word	2		
2Ah	VOLTAGE_SCALE_MONITOR	R/W Word	2		
2Bh	Reserved				
2Ch	Reserved				
2Dh	Reserved				
2Eh	Reserved				
2Fh	Reserved				
30h	COEFFICIENTS	Block R/W Process Call	6 (Includes Byte Count)		
31h	POUT_MAX	R/W Word	2		
32h	MAX_DUTY	R/W Word	2		
33h	FREQUENCY_SWITCH	R/W Word	2		
34h	Reserved				
35h	VIN_ON	R/W Word	2		
36h	VIN_OFF	R/W Word	2		
37h	INTERLEAVE	R/W Word	2		
38h	IOUT_SCALE	R/W Word	2		
39h	IOUT_CAL_OFFSET	R/W Word	2		
3Ah	VFAN_1	R/W Word	2		

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
3Bh	VFAN_2	R/W Word	2		
3Ch	Reserved				
3Dh	Reserved				
3Eh	Reserved				
3Fh	Reserved				
40h	VOUT_OV_FAULT_LIMIT	R/W Word	2		
41h	VOUT_OV_FAULT_RESPONSE	R/W Byte	1		
42h	VOUT_OV_WARN_LIMIT	R/W Word	2		
43h	VOUT_UV_WARN_LIMIT	R/W Word	2		
44h	VOUT_UV_FAULT_LIMIT	R/W Word	2		
45h	VOUT_UV_FAULT_RESPONSE	R/W Byte	1		
46h	IOUT_OC_FAULT_LIMIT	R/W Word	2		
47h	IOUT_OC_FAULT_RESPONSE	R/W Byte	1		
48h	IOUT_OC_LV_FAULT_LIMIT	R/W Word	2		
49h	IOUT_OC_LV_FAULT_RESPONSE	R/W Byte	1		
4Ah	IOUT_OC_WARN_LIMIT	R/W Word	2		
4Bh	IOUT_UC_FAULT_LIMIT	R/W Word	2		
4Ch	IOUT_UC_FAULT_RESPONSE	R/W Byte	1		
4Dh	Reserved For POUT_FAULT_LIMIT	R/W Word	2		
4Eh	Reserved For: POUT_MAX_FAULT_RESPONSE	R/W Byte	1		
4Fh	OT_FAULT_LIMIT	R/W Word	2		
50h	OT_FAULT_RESPONSE	R/W Byte	1		
51h	OT_WARN_LIMIT	R/W Word	2		
52h	UT_WARN_LIMIT	R/W Word	2		
53h	UT_FAULT_LIMIT	R/W Word	2		
54h	UT_FAULT_RESPONSE	R/W Byte	1		
55h	VIN_OV_FAULT_LIMIT	R/W Word	2		
56h	VIN_OV_FAULT_RESPONSE	R/W Byte	1		
57h	VIN_OV_WARN_LIMIT	R/W Word	2		
58h	VIN_UV_WARN_LIMIT	R/W Word	2		
59h	VIN_UV_FAULT_LIMIT	R/W Word	2		
5Ah	VIN_UV_FAULT_RESPONSE	R/W Byte	1		

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
5Bh	IIN_OC_FAULT_LIMIT	R/W Word	2		
5Ch	IIN_OC_FAULT_RESPONSE	R/W Byte	1		
5Dh	IIN_OC_WARN_LIMIT	R/W Word	2		
5Eh	POWER_GOOD_ON	R/W Word	2		
5Fh	POWER_GOOD_OFF	R/W Word	2		
60h	TON_DELAY	R/W Word	2		
61h	TON_RISE	R/W Word	2		
62h	TON_MAX_FAULT_LIMIT	R/W Word	2		
63h	TON_MAX_FAULT_RESPONSE	R/W Byte	1		
64h	TOFF_DELAY	R/W Word	2		
65h	TOFF_FALL	R/W Word	2		
66h	TOFF_MAX_FAULT_LIMIT	R/W Word	2		
67h	TOFF_MAX_FAULT_RESPONSE	R/W Byte	1		
68h	Reserved				
69h	Reserved				
6Ah	Reserved				
6Bh	Reserved				
6Ch	Reserved				
6Dh	Reserved				
6Eh	Reserved				
6Fh	Reserved				
70h	Reserved (Test Input Fuse A)				
71h	Reserved (Test Input Fuse B)				
72h	Reserved (Test Input OR-ing A)				
73h	Reserved (Test Input OR-ing B)				
74h	Reserved (Test Output OR-ing)				
75h	Reserved				
76h	Reserved				
77h	Reserved				
78h	STATUS_BYTE	Read Byte	1		
79h	STATUS_WORD	Read Word	2		
7Ah	STATUS_VOUT	Read Byte	1		
7Bh	STATUS_IOUT	Read Byte	1		

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
7Ch	STATUS_INPUT	Read Byte	1		
7Dh	STATUS_TEMPERATURE	Read Byte	1		
7Eh	STATUS_CML	Read Byte	1		
7Fh	STATUS_OTHER	Read Byte	1		
80h	STATUS_MFR_SPECIFIC	Read Byte	1		
81h	Reserved				
82h	Reserved				
83h	Reserved				
84h	Reserved				
85h	Reserved				
86h	Reserved				
87h	Reserved				
88h	READ_VIN	Read Word	2		
89h	READ_IIN	Read Word	2		
8Ah	READ_VCAP	Read Word	2		
8Bh	READ_VOUT	Read Word	2		
8Ch	READ_IOUT	Read Word	2		
8Dh	READ_TEMPERATURE_1	Read Word	2		
8Eh	READ_TEMPERATURE_2	Read Word	2		
8Fh	READ_TEMPERATURE_3	Read Word	2		
90h	READ_FAN SPEED_1	Read Word	2		
91h	READ_FAN SPEED_2	Read Word	2		
92h	READ_VFAN_1	Read Word	2		
93h	READ_VFAN_2	Read Word	2		
94h	READ_DUTY_CYCLE	Read Word	2		
95h	READ_FREQUENCY	Read Word	2		
96h	Reserved				
97h	Reserved				
98h	PMBUS_REVISION	Read Byte	1		
99h	MFR_ID	R/W Block	Variable		
9Ah	MFR_MODEL	R/W Block	Variable		
9Bh	MFR_REVISION	R/W Block	Variable		
9Ch	MFR_LOCATION	R/W Block	Variable		

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
9Dh	MFR_DATE	R/W Block	Variable		
9Eh	MFR_SERIAL	R/W Block	Variable		
9Fh	Reserved				
A0h	MFR_VIN_MIN	Read Word	2		
A1h	MFR_VIN_MAX	Read Word	2		
A2h	MFR_IIN_MAX	Read Word	2		
A3h	MFR_PIN_MAX	Read Word	2		
A4h	MFR_VOUT_MIN	Read Word	2		
A5h	MFR_VOUT_MAX	Read Word	2		
A6h	MFR_IOUT_MAX	Read Word	2		
A7h	MFR_POUT_MAX	Read Word	2		
A8h	MFR_TAMBIENT_MAX	Read Word	2		
A9h	MFR_TAMBIENT_MIN	Read Word	2		
AAh	Reserved				
ABh	Reserved				
ACh	Reserved				
ADh	Reserved				
AEh	Reserved				
AFh	Reserved				
B0h	USER_DATA_00	Block R/W	Variable		
B1h	USER_DATA_01	Block R/W	Variable		
B2h	USER_DATA_02	Block R/W	Variable		
B3h	USER_DATA_03	Block R/W	Variable		
B4h	USER_DATA_04	Block R/W	Variable		
B5h	USER_DATA_05	Block R/W	Variable		
B6h	USER_DATA_06	Block R/W	Variable		
B7h	USER_DATA_07	Block R/W	Variable		
B8h	USER_DATA_08	Block R/W	Variable		
B9h	USER_DATA_09	Block R/W	Variable		
BAh	USER_DATA_10	Block R/W	Variable		
BBh	USER_DATA_11	Block R/W	Variable		
BCh	USER_DATA_12	Block R/W	Variable		
BDh	USER_DATA_13	Block R/W	Variable		

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
BEh	USER_DATA_14	Block R/W	Variable		
BFh	USER_DATA_15	Block R/W	Variable		
C0h	Reserved				
C1h	Reserved				
C2h	Reserved				
C3h	Reserved				
C4h	Reserved				
C5h	Reserved				
C6h	Reserved				
C7h	Reserved				
C8h	Reserved				
C9h	Reserved				
CAh	Reserved				
CBh	Reserved				
CCh	Reserved				
CDh	Reserved				
CEh	Reserved				
CFh	Reserved				
D0h	MFR_SPECIFIC_00	Mfr. Defined	Mfr. Defined		
D1h	MFR_SPECIFIC_01	Mfr. Defined	Mfr. Defined		
D2h	MFR_SPECIFIC_02	Mfr. Defined	Mfr. Defined		
D3h	MFR_SPECIFIC_03	Mfr. Defined	Mfr. Defined		
D4h	MFR_SPECIFIC_04	Mfr. Defined	Mfr. Defined		
D5h	MFR_SPECIFIC_05	Mfr. Defined	Mfr. Defined		
D6h	MFR_SPECIFIC_06	Mfr. Defined	Mfr. Defined		
D7h	MFR_SPECIFIC_07	Mfr. Defined	Mfr. Defined		
D8h	MFR_SPECIFIC_08	Mfr. Defined	Mfr. Defined		
D9h	MFR_SPECIFIC_09	Mfr. Defined	Mfr. Defined		
DAh	MFR_SPECIFIC_10	Mfr. Defined	Mfr. Defined		
DBh	MFR_SPECIFIC_11	Mfr. Defined	Mfr. Defined		
DCh	MFR_SPECIFIC_12	Mfr. Defined	Mfr. Defined		
DDh	MFR_SPECIFIC_13	Mfr. Defined	Mfr. Defined		
DEh	MFR_SPECIFIC_14	Mfr. Defined	Mfr. Defined		

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
DFh	MFR_SPECIFIC_15	Mfr. Defined	Mfr. Defined		
E0h	MFR_SPECIFIC_16	Mfr. Defined	Mfr. Defined		
E1h	MFR_SPECIFIC_17	Mfr. Defined	Mfr. Defined		
E2h	MFR_SPECIFIC_18	Mfr. Defined	Mfr. Defined		
E3h	MFR_SPECIFIC_19	Mfr. Defined	Mfr. Defined		
E4h	MFR_SPECIFIC_20	Mfr. Defined	Mfr. Defined		
E5h	MFR_SPECIFIC_21	Mfr. Defined	Mfr. Defined		
E6h	MFR_SPECIFIC_22	Mfr. Defined	Mfr. Defined		
E7h	MFR_SPECIFIC_23	Mfr. Defined	Mfr. Defined		
E8h	MFR_SPECIFIC_24	Mfr. Defined	Mfr. Defined		
E9h	MFR_SPECIFIC_25	Mfr. Defined	Mfr. Defined		
EAh	MFR_SPECIFIC_26	Mfr. Defined	Mfr. Defined		
EBh	MFR_SPECIFIC_27	Mfr. Defined	Mfr. Defined		
ECh	MFR_SPECIFIC_28	Mfr. Defined	Mfr. Defined		
EDh	MFR_SPECIFIC_29	Mfr. Defined	Mfr. Defined		
EEh	MFR_SPECIFIC_30	Mfr. Defined	Mfr. Defined		
EFh	MFR_SPECIFIC_31	Mfr. Defined	Mfr. Defined		
F0h	MFR_SPECIFIC_32	Mfr. Defined	Mfr. Defined		
F1h	MFR_SPECIFIC_33	Mfr. Defined	Mfr. Defined		
F2h	MFR_SPECIFIC_34	Mfr. Defined	Mfr. Defined		
F3h	MFR_SPECIFIC_35	Mfr. Defined	Mfr. Defined		
F4h	MFR_SPECIFIC_36	Mfr. Defined	Mfr. Defined		
F5h	MFR_SPECIFIC_37	Mfr. Defined	Mfr. Defined		
F6h	MFR_SPECIFIC_38	Mfr. Defined	Mfr. Defined		
F7h	MFR_SPECIFIC_39	Mfr. Defined	Mfr. Defined		
F8h	MFR_SPECIFIC_40	Mfr. Defined	Mfr. Defined		
F9h	MFR_SPECIFIC_41	Mfr. Defined	Mfr. Defined		
FAh	MFR_SPECIFIC_42	Mfr. Defined	Mfr. Defined		
FBh	MFR_SPECIFIC_43	Mfr. Defined	Mfr. Defined		
FCh	MFR_SPECIFIC_44	Mfr. Defined	Mfr. Defined		
FDh	MFR_SPECIFIC_45	Mfr. Defined	Mfr. Defined		
FEh	MFR_SPECIFIC_COMMAND EXT	Extended Command	1 Or 2		

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes	Reserved For Future Use	Reserved For Future Use
FFh	PMBUS_COMMAND_EXT	Extended Command	1 Or 2		

APPENDIX II. Paramenter Codes For Retrieving DIRECT Format Coefficients

Table 20. COEFFICIENTS Command Parameter Codes

Parameter	Parameter Codes For Writing Or Reading A DIRECT Mode Value	
	Writing	Reading
VOUT_COMMAND	00h 00h	00h 01h
VOUT_TRIM	00h 02h	00h 03h
VOUT_CAL	00h 04h	00h 05h
VOUT_MAX	00h 06h	00h 07h
VOUT_MARGIN_HIGH	00h 08h	00h 09h
VOUT_MARGIN_LOW	00h 0Ah	00h 0Bh
VOUT_TRANSITION_RATE	00h 0Ch	00h 0Dh
VOUT_DROOP	00h 0Eh	00h 0Fh
VOLTAGE_SCALE_LOOP	00h 10h	00h 11h
VOLTAGE_SCALE_MONITOR	00h 12h	00h 13h
POUT_MAX	00h 20h	00h 21h
MAX_DUTY	00h 22h	00h 23h
FREQUENCY_SWITCH	00h 24h	00h 25h
VIN_ON	00h 26h	00h 27h
VIN_OFF	00h 28h	00h 29h
INTERLEAVE	00h 2Ah	00h 2Bh
IOUT_SCALE	00h 2Ch	00h 2Dh
IOUT_CAL_OFFSET	00h 2Eh	00h 2Fh
VFAN_1	00h 30h	00h 31h
VFAN_2	00h 32h	00h 33h
VOUT_OV_FAULT_LIMIT	00h 40h	00h 41h
VOUT_OV_WARN_LIMIT	00h 42h	00h 43h
VOUT_UV_WARN_LIMIT	00h 44h	00h 45h
VOUT_UV_FAULT_LIMIT	00h 46h	00h 47h
IOUT_OC_FAULT_LIMIT	00h 48h	00h 49h
IOUT_OC_LV_FAULT_LIMIT	00h 4Ah	00h 4Bh
IOUT_OC_WARN_LIMIT	00h 4Ch	00h 4Dh

Parameter	Parameter Codes For Writing Or Reading A DIRECT Mode Value	
	Writing	Reading
IOUT_UC_FAULT_LIMIT	00h 4Eh	00h 4Fh
Reserved For POUT_FAULT_LIMIT	00h 50h	00h 51h
OT_FAULT_LIMIT	00h 52h	00h 53h
OT_WARN_LIMIT	00h 54h	00h 55h
UT_WARN_LIMIT	00h 56h	00h 57h
UT_FAULT_LIMIT	00h 58h	00h 59h
VIN_OV_FAULT_LIMIT	00h 5Ah	00h 5Bh
VIN_OV_WARN_LIMIT	00h 5Ch	00h 5Dh
VIN_UV_WARN_LIMIT	00h 5Eh	00h 5Fh
VIN_UV_FAULT_LIMIT	00h 60	00h 61h
IIN_OC_FAULT_LIMIT	00h 62h	00h 63h
IIN_OC_WARN_LIMIT	00h 64h	00h 65h
POWER_GOOD_ON	00h 66h	00h 67h
POWER_GOOD_OFF	00h 68h	00h 69h
TON_DELAY	00h 6Ah	00h 6Bh
TON_RISE	00h 6Ch	00h 6Dh
TON_MAX_FAULT_LIMIT	00h 6Eh	00h 6Fh
TOFF_DELAY	00h 70h	00h 71h
TOFF_FALL	00h 72h	00h 73h
TOFF_MAX_FAULT_LIMIT	00h 74h	00h 76h
READ_VIN	N/A	00h 90h
READ_IIN	N/A	00h 91h
READ_VCAP	N/A	00h 92h
READ_VOUT	N/A	00h 93h
READ_IOUT	N/A	00h 94h
READ_TEMPERATURE_1	N/A	00h 95h
READ_TEMPERATURE_2	N/A	00h 96h
READ_TEMPERATURE_3	N/A	00h 97h
READ_FAN SPEED_1	N/A	00h 98h
READ_FAN SPEED_2	N/A	00h 99h
READ_VFAN_1	N/A	00h 9Ah
READ_VFAN_2	N/A	00h 9Bh

Parameter	Parameter Codes For Writing Or Reading A DIRECT Mode Value	
	Writing	Reading
READ_DUTY_CYCLE	N/A	00h 9Ch
READ_FREQUENCY	N/A	00h 9Dh
MFR_VIN_MIN	00h B0h	00h B1h
MFR_VIN_MAX	00h B2h	00h B3h
MFR_IIN_MAX	00h B4h	00h B5h
MFR_PIN_MAX	00h B6h	00h B7h
MFR_VOUT_MIN	00h B8h	00h B9h
MFR_VOUT_MAX	00h BAh	00h BBh
MFR_IOUT_MAX	00h BCh	00h BDh
MFR_POUT_MAX	00h BEh	00h BFh
MFR_TAMBIENT_MAX	00h C0h	00h C1h
MFR_TAMBIENT_MIN	00h C2h	00h C3h
USER_DATA_00	01h 00h	01h 01h
USER_DATA_01	01h 02h	01h 03h
USER_DATA_02	01h 04h	01h 05h
USER_DATA_03	01h 06h	01h 07h
USER_DATA_04	01h 08h	01h 09h
USER_DATA_05	01h 0Ah	01h 0Bh
USER_DATA_06	01h 0Ch	01h 0Dh
USER_DATA_07	01h 0Eh	01h 0Eh
USER_DATA_08	01h 10h	01h 11h
USER_DATA_09	01h 12h	01h 13h
USER_DATA_10	01h 14h	01h 15h
USER_DATA_11	01h 16h	01h 17h
USER_DATA_12	01h 18h	01h 19h
USER_DATA_13	01h 1Ah	01h 1Bh
USER_DATA_14	01h 1Ch	01h 1Dh
USER_DATA_15	01h 1Eh	01h 1Fh
MFR_SPECIFIC_00	02h 00h	02h 01h
MFR_SPECIFIC_01	02h 02h	02h 03h

Parameter	Parameter Codes For Writing Or Reading A DIRECT Mode Value	
	Writing	Reading
MFR_SPECIFIC_02	02h 04h	02h 05h
MFR_SPECIFIC_03	02h 06h	02h 07h
MFR_SPECIFIC_04	02h 08h	02h 09h
MFR_SPECIFIC_05	02h 0Ah	02h 0Bh
MFR_SPECIFIC_06	02h 0Ch	02h 0Dh
MFR_SPECIFIC_07	02h 0Eh	02h 0Fh
MFR_SPECIFIC_08	02h 10h	02h 11h
MFR_SPECIFIC_09	02h 12h	02h 13h
MFR_SPECIFIC_10	02h 14h	02h 15h
MFR_SPECIFIC_11	02h 16h	02h 17h
MFR_SPECIFIC_12	02h 18h	02h 19h
MFR_SPECIFIC_13	02h 1Ah	02h 1Bh
MFR_SPECIFIC_14	02h 1Ch	02h 1Dh
MFR_SPECIFIC_15	02h 1Eh	02h 1Fh
MFR_SPECIFIC_16	02h 20h	02h 21h
MFR_SPECIFIC_17	02h 22h	02h 23h
MFR_SPECIFIC_18	02h 24h	02h 25h
MFR_SPECIFIC_19	02h 26h	02h 27h
MFR_SPECIFIC_20	02h 28h	02h 29h
MFR_SPECIFIC_21	02h 2Ah	02h 2Bh
MFR_SPECIFIC_22	02h 2Ch	02h 2Dh
MFR_SPECIFIC_23	02h 2Eh	02h 2Fh
MFR_SPECIFIC_24	02h 30h	02h 31h
MFR_SPECIFIC_25	02h 32h	02h 33h
MFR_SPECIFIC_26	02h 34h	02h 35h
MFR_SPECIFIC_27	02h 36h	02h 37h
MFR_SPECIFIC_28	02h 38h	02h 39h
MFR_SPECIFIC_29	02h 3Ah	02h 3Bh
MFR_SPECIFIC_30	02h 3Ch	02h 3Dh
MFR_SPECIFIC_31	02h 3Eh	02h 3Fh
MFR_SPECIFIC_32	02h 40h	02h 41h
MFR_SPECIFIC_33	02h 42h	02h 43h
MFR_SPECIFIC_34	02h 44h	02h 45h

Parameter	Parameter Codes For Writing Or Reading A DIRECT Mode Value	
	Writing	Reading
MFR_SPECIFIC_35	02h 46h	02h 47h
MFR_SPECIFIC_36	02h 48h	02h 49h
MFR_SPECIFIC_37	02h 4Ah	02h 4Bh
MFR_SPECIFIC_38	02h 4Ch	02h 4Dh
MFR_SPECIFIC_39	02h 4Eh	02h 4Fh
MFR_SPECIFIC_40	02h 50h	02h 51h
MFR_SPECIFIC_41	02h 52h	02h 53h
MFR_SPECIFIC_42	02h 54h	02h 55h
MFR_SPECIFIC_43	02h 56h	02h 57h
MFR_SPECIFIC_44	02h 58h	02h 59h
MFR_SPECIFIC_45	02h 5Ah	02h 5Bh