PMBus™
Power System Management Protocol Specification
Part I – General Requirements, Transport And Electrical Interface

Revision 1.3

18 March 2014

www.powerSIG.org

Filename: PMBus_Specification_Part_I_Rev_1-3_20140318B.docx
Last saved: 19 Mar 2014, 08:51

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, non-infringement, 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
1.1	5 Feb 2007	Second public release.	Robert V. White Astec/Artesyn
1.2	6 Sep 2010	Third public release	Robert V. White Embedded Power Labs
1.3	18 March 2014	Revision 1.3, Updated release candidate	Robert V. White Embedded Power Labs

Table Of Contents

1. Introduction 5
1.1. Specification Scope 5
1.1.1. Specification Structure 5
1.1.2. What Is Included 5
1.1.3. What Is Not Included In The PMBus Specification 5
1.2. Specification Changes Since The Last Revision 5
1.3. Where To Send Feedback And Comments 5
2. Related Documents 5
2.1. Scope 5
2.2. Applicable Documents 5
2.3. Reference Documents 6
3. Reference Information 6
3.1. Signal and Parameter Names 6
3.2. Numerical Formats 6
3.2.1. Decimal Numbers 6
3.2.2. Floating Point Numbers 6
3.2.3. Binary Numbers 6
3.2.4. Hexadecimal Numbers 6
3.2.5. Examples 7
3.3. Byte And Bit Order 7
3.4. Bit And Byte Illustrations 7
3.5. Abbreviations, Acronyms And Definitions 9
4. General Requirements 11
4.1. Compliance 11
4.2. Unassisted Start Up And Operation 12
4.3. High Impedance When Unpowered 12
4.4. PMBus And AVSBus Supply Voltages 12
5. Transport 12
5.1. SMBus, Version 3.0 12
5.2. Bus Speed 12
5.3. Electrical Drive Levels 12
5.4. SMBus Protocols 12
5.5. SMBus Features, Functions, And Protocols That Are Optional In PMBus 13
5.5.1. Packet Error Checking 13
5.5.2. SMBus Address Resolution Protocol (ARP) 13
5.6. Extensions To The SMBus Specification 13
5.6.1. Group Command Protocol 13
5.6.2. Extended Command Protocol 14
5.6.3. Zone Read Protocol 16
5.6.4. Zone Write Protocol 20
5.7. 37hExceptions To The SMBus Specification 20
6. Addressing 20
7. Communication From PMBus Devices To The Host 20

7.1. Communicating Over The Bus 20
7.2. Communicating With An Interrupt Signal 20
8. Hardwired Signals 21
8.1. Electrical Interface 21
8.2. Timing 21
8.3. Control Signal (CONTROL) 21
8.4. Write Protect (WP) 21
8.5. Other Pins 21
9. Accuracy 22
10. Firmware Updates 22
APPENDIX I. Summary Of Changes 23

Table Of Figures

Figure 1. Bit Order Within A Byte 7
Figure 2. Group Command Protocol Without PEC 14
Figure 3. Group Command Protocol With PEC 14
Figure 4. Extended Command Read Byte Protocol 15
Figure 5. Extended Command Read Byte With PEC 15
Figure 6. Extended Command Write Byte Protocol 15
Figure 7. Extended Command Write Byte Protocol With PEC 15
Figure 8. Extended Command Read Word Protocol 16
Figure 9. Extended Command Read Word Protocol With PEC 16
Figure 10. Extended Command Write Word Protocol 16
Figure 11. Extended Command Write Word Protocol With PEC 16
Figure 12. Zone Read Initiation with PMBus Command Code 17
Figure 13. Zone Read Initiation with Status Mask 17
Figure 14. Command Control Code 18

Table Of Tables

Table 1. Bit And Byte Symbols Used In This Specification 7
Table 2: Setting AR and ST Bits 18

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 three parts. Part I, this document, includes the general requirements, defines the transport and electrical interface and timing requirements of hardwired signals.

Part II defines the command language used with the PMBus.

Part III provides the specification for the AVSBus interface.

1.1.2. What Is Included

This specification defines a protocol to manage 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

A summary of the changes between this revision and Revision 1.2 are shown in APPENDIX I.

1.3. Where To Send Feedback And Comments

Please send all comments by email to: techquestions@smiforum.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 device 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 II, Command Language*, System Management Interface Forum, Revision 1.3, March 2014

[A02] *PMBus Power System Management Protocol, Part III, AVSBus*, System Management Interface Forum, Revision 1.3, March 2014

[A03] *System Management Bus (SMBus) Specification*, System Management Interface Forum, Version 3.0, March 2014

[A04] *I²C-bus specification and user manual*, Revision 5, NXP Semiconductors, 9 October 2012

[A05] ISO/IEC 8859-1:1998*, 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1*, and all corrigenda, amendments published through the date of release of this specification.

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.

[R01] PMBus Application Note AN001, *Using The Zone Read And Zone Write Protocols*

3. Reference Information

3.1. Signal and Parameter Names

The names of signals, commands 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. Floating Point Numbers

Numbers explicitly identified as floating point are identified with a suffix of “f”.

3.2.3. Binary Numbers

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

All signed binary numbers are two’s complement.

3.2.4. Hexadecimal Numbers

Numbers in hexadecimal format are indicated by a suffix of “h”.

3.2.5. Examples

255d⇔ FFh ⇔ 11111111b

175d⇔ AFh ⇔ 10101111b

1.2f

3.3. Byte And Bit Order

As specified in the SMBus specification, Version 3.0 [A03]:

- 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.

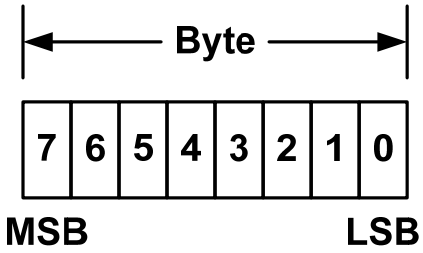
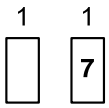
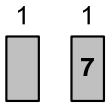
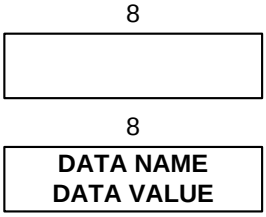
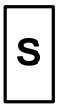


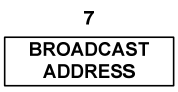
Figure 1. Bit Order Within A Byte

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
	A unshaded vertical rectangle indicates a single bit sent from the host (bus master) to a slave
	A shaded vertical rectangle with a shaded interior indicates a bit sent from a slave device to the bus master.
	An unshaded rectangle with a number over it represents one or more bits, as indicated by the number, sent from the master to the slave. The name of the data or bit field may be included within the rectangle. If the data has a specific value, as might be shown in an example of a command, the value is written below the data or bit field name.

Symbol	Meaning
8  8 	A shaded rectangle with a number over it represents one or more bits, as indicated by the number, sent from the slave to the master. The name of the data or bit field may be included within the rectangle. If the data has a specific value, as might be shown in an example of a command, the value is written below the data or bit field name.
	The START condition sent from a bus master device. The START condition is not a bit and does not have a number 1 over it.
	A REPEATED START condition sent from a bus master device. The REPEATED START condition is not a bit and does not have a number 1 over it.
1 	An Acknowledge (ACK) condition send from the host
1 	A Not Acknowledge (NACK) condition sent from the host
1 	An Acknowledge (ACK) condition sent from a slave device
1 	A Not Acknowledge (NACK) condition sent from a slave device
	A STOP condition sent by a bus master device. The STOP condition is not a bit and does not have a number 1 over it.
7 	The first seven bits of the address byte, generally corresponding to the physical address of the device.
1 	The bit [0] of the address byte with a value of 1, indicating the device is being addressed with a read.

Symbol	Meaning
	The bit [0] of the address byte with a value of 0, indicating the device is being addressed with a write.
	The SMBus broadcast address to which all devices must respond. The value is 0000000b. This is always used only with the bit [0] equal to 0 (write).

3.5. Abbreviations, Acronyms And Definitions

Term	Definition
ACK	ACKnowedge. The response from a receiving unit indicating that it has received a byte. See the SMBus specification [A03] 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.
AVS	Adaptive Voltage Scaling. AVS is used by a device to control its supply voltage, generally to minimize power consumption for a given operating condition.
AVSBus	AVSBus is an interface designed to facilitate and expedite point-to-point communication between an ASIC , FPGA, or other logic, memory, or processor devices and a POL control device on a system for the purpose of adaptive voltage scaling.
Bias, Bias Power	Power to the PMBus device's control circuit or ICs
Clear	When referring to a bit or bits, this means setting the value to zero.
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 the SMBus specification [A03] for more information.

Term	Definition
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 the SMBus specification [A03] 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 the SMBus specification [A03] 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
OP	Overpower
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 the SMBus specification [A03] for more information.
PIN	Input power
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.
Plain Text	Characters stored according to ISO/IEC 8859-1:1998 ([A05])
POL	Point-of-load
Positive Output Current	Current that flows out of the converter's output.
POUT	Output power
Product Literature	Data sheets, product briefs, application notes or any other documentation describing the operation and application of a device.
Set	When referring to a bit or bits, this means setting the value to one.

Term	Definition
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 the SMBus specification [A03] for more information.
SMBus	System Management Bus - See the SMBus specification [A03] 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. General Requirements

4.1. Compliance

The PMBus protocol is intended to cover a wide range of power system architectures and converters.

Not all PMBus devices must support all of the available features, functions and commands.

To be compliant to the PMBus specification, a device must:

- Meet all of the requirements of Part I of the PMBus specification (this document),
- Support at least one of the non-manufacturer specific commands given in Part II of the PMBus specification [A01],
- If a device accepts a PMBus command code, it must execute that function as described in Part II of the PMBus specification [A01],
- If a device does not accept a given PMBus command code, it must respond as described in the Fault Management And Reporting section of Part II of the PMBus specification [A01].

4.2. Unassisted Start Up And Operation

PMBus devices, upon application of power, must start up and begin operation in a controlled manner, as programmed internally or externally, without requiring communication from the serial bus.

4.3. High Impedance When Unpowered

As described in of the SMBus specification, [A03], the electrical signals of a PMBus device must present a high impedance to the bus when the device is unpowered, during startup until fully powered, and during shutdown once the device can no longer assure the proper signal levels.

4.4. PMBus And AVSBus Supply Voltages

The V_{DD} supplies for the PMBus interface and AVSBus interface are independent of each other. It is the user's responsibility to assure that all system and device power sequencing requirements to assure proper operation are met.

5. Transport

5.1. SMBus, Version 3.0

PMBus devices must use the System Management Bus (SMBus), Version 3.0 [A03], for transport with the extensions and exceptions listed below.

5.2. Bus Speed

All PMBus devices must support operation at 100 kHz as described in the SMBus specification [A03]. Support for operation at maximum bus speeds of 400 kHz and 1 MHz, as described in the SMBus specification [A03] is optional for PMBus devices.

5.3. Electrical Drive Levels

A PMBus device shall comply with the corresponding electrical drive levels as given in the SMBus specification [A03] for the maximum bus speed supported by the device.

5.4. SMBus Protocols

PMBus devices must use the SMBus transaction protocol associated with each PMBus command as described in Part II of the PMBus specification [A01].

5.5. SMBus Features, Functions, And Protocols That Are Optional In PMBus

5.5.1. Packet Error Checking

Support for the SMBus Packet Error Checking (PEC) protocol is optional.

5.5.2. SMBus Address Resolution Protocol (ARP)

Support for the SMBus Address Resolution Protocol (ARP) is optional for PMBus devices.

5.6. Extensions To The SMBus Specification

5.6.1. Group Command Protocol

PMBus devices must support the Group Command Protocol. The Group Command Protocol is used to send commands to more than one PMBus device. The commands are sent in one continuous transmission. When the devices detect the STOP condition that ends the sending of commands, they all begin executing the command they received.

It is not a requirement that all devices receive the same command.

No more than one command can be sent to any one device in one Group Command packet.

The Group Command Protocol must not be used with commands that require the receiving device to respond with data, such as the STATUS_BYTE command (PMBus specification, Part II [A01]).

Example: The Group Command Protocol could be used to signal all of the devices on the PMBus to change their margin state at one time. All of the devices on the bus might be instructed to set their output voltage to the VOUT_MARGIN_HIGH with which they have been programmed. Equally valid would be a Group Command Protocol transmission that instructed one device on the bus to set the output voltage to its programmed VOUT_MARGIN_LOW while all of the other devices were instructed to set their output voltage to the programmed VOUT_MARGIN_HIGH values.

As shown below in Figure 2 and Figure 3, the Group Command Protocol uses REPEATED START conditions to separate commands for each device. The Group Command Protocol begins with the START condition, followed by the seven bit address of the first device to receive a command and then by the write bit (0). The slave device ACKs and the master or host sends a command with the associated data byte or bytes.

After the last data byte is sent to the first device, the host or master does NOT send a STOP condition. Instead, it sends a REPEATED START condition, followed by the seven bit address of the second device to receive a command, a write bit and the command code and the associated data bytes.

If, and only if, this is the last device to receive a command, the host or master sends a STOP condition. Otherwise, the host or master sends a REPEATED START condition and starts transmitting the address of the third device to receive a command.

This process continues until all devices have received their command codes, data bytes, and if used and supported, PEC byte. Then when all devices have received their information, the host or master sends a STOP condition.

If PEC is used, then each device's sub-packet has its own PEC byte, computed only for that device's sub-packet, including that device's address.

When the devices who have received a command through this protocol detect the STOP condition, they are to begin execution immediately of the received command.

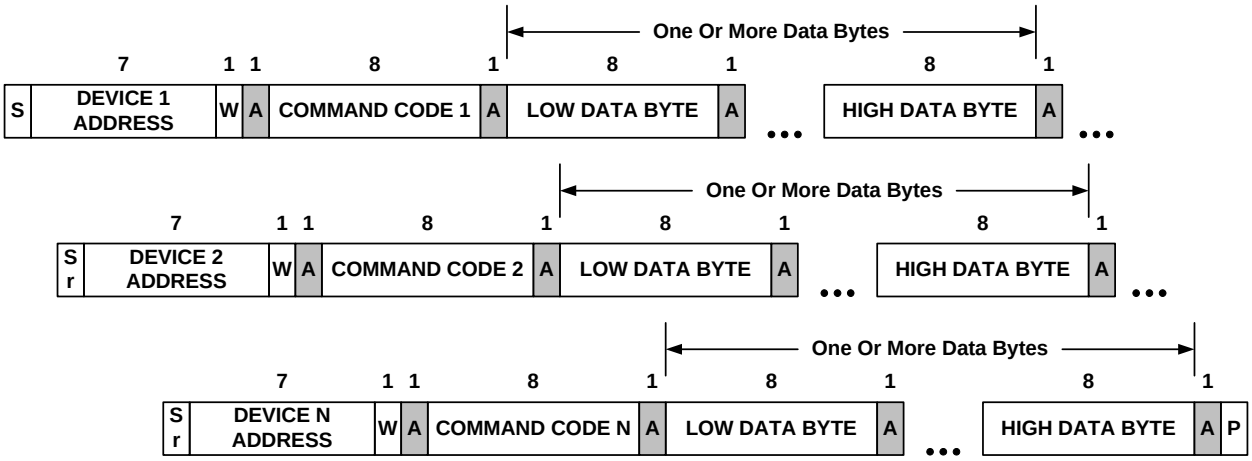


Figure 2. Group Command Protocol Without PEC

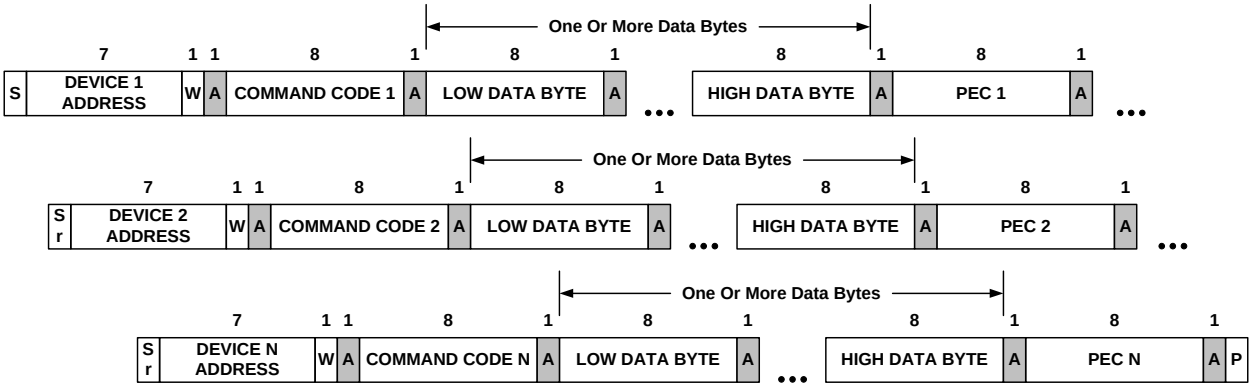


Figure 3. Group Command Protocol With PEC

When using Packet Error Checking with the Group Command Protocol, the PEC byte is calculated using only the address, command and data bytes for each device. For example, in Figure 3, PEC 1 is calculated using Device Address 1 including the Write bit, Command Code 1, and the data associated with Command Code 1. PEC 1 need only be calculated by the device at Device Address 1.

Similarly, PEC Byte 2 is calculated using Device Address 2 including the Write bit, Command Code 2, and the data associated with Command Code 2. Device 1 must not continue calculating PEC 1 after it sees the Repeated Start.

5.6.2. Extended Command Protocol

The Extended Command protocol allows for an extra 256 command codes. This is done by making the command code two bytes. The first byte (the low data byte) is a reserved value indicating that the extended command format is being used. The second byte (the high order byte) is the PMBus command to be executed.

Shown below in Figure 4 through Figure 9 are examples of the Extended Command Protocol with commands that use the READ BYTE, WRITE BYTE, READ WORD, and

WRITE WORD protocols, but the Extended Command Protocol can be used with any PMBus command and its associated protocol, including Manufacturer Specific Commands and any custom protocols they may use.

5.6.2.1. Example: Extended Command Protocol For READ BYTE and WRITE BYTE

The use of the Extended Command Protocol with the READ BYTE and WRITE BYTE protocols are illustrated below.

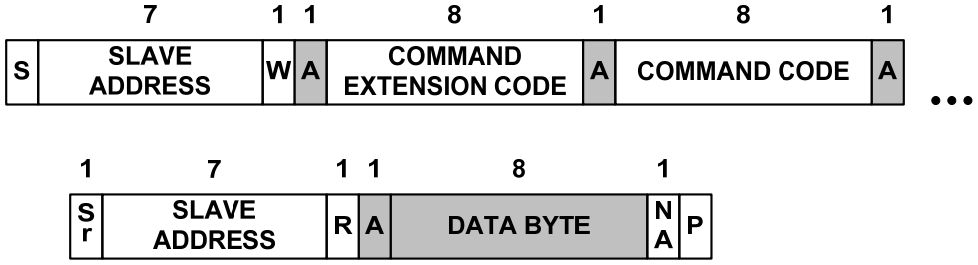


Figure 4. Extended Command Read Byte Protocol

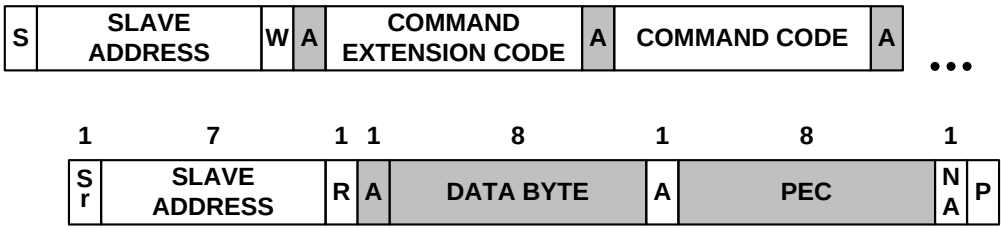


Figure 5. Extended Command Read Byte With PEC

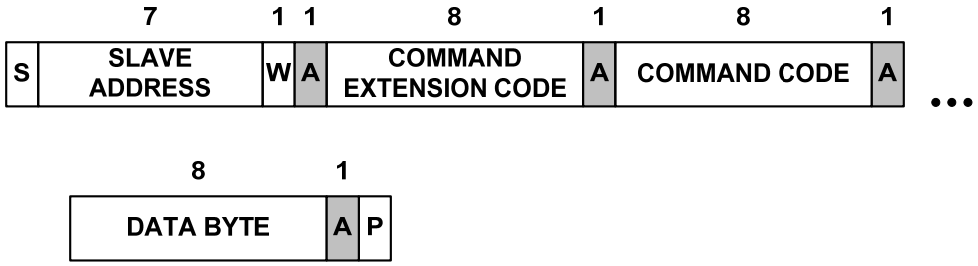


Figure 6. Extended Command Write Byte Protocol

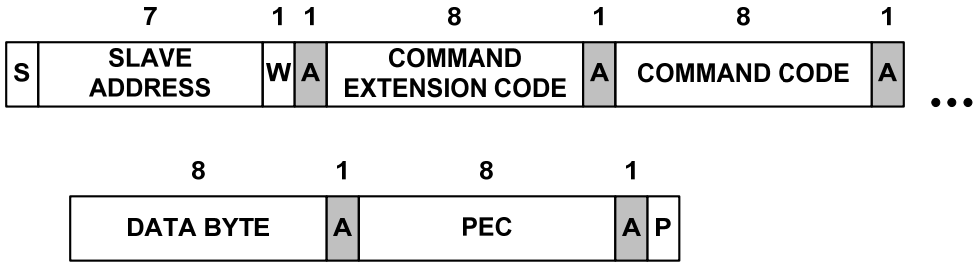


Figure 7. Extended Command Write Byte Protocol With PEC

5.6.2.2. Example: Extended Command Protocol For READ WORD and WRITE WORD

The use of the Extended Command Protocol with the READ WORD and WRITE WORD protocols are illustrated below.

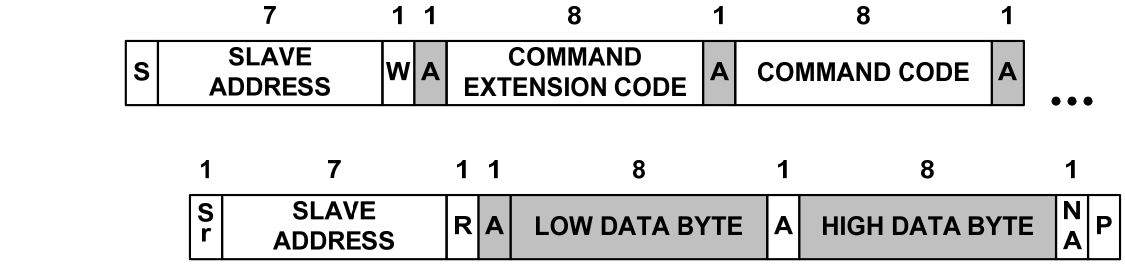


Figure 8. Extended Command Read Word Protocol

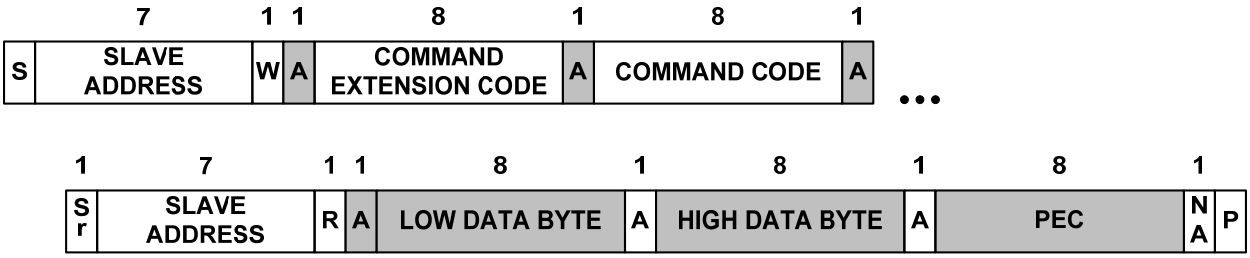


Figure 9. Extended Command Read Word Protocol With PEC

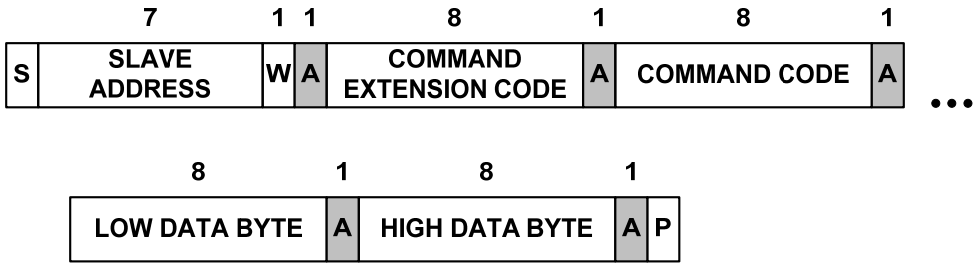


Figure 10. Extended Command Write Word Protocol

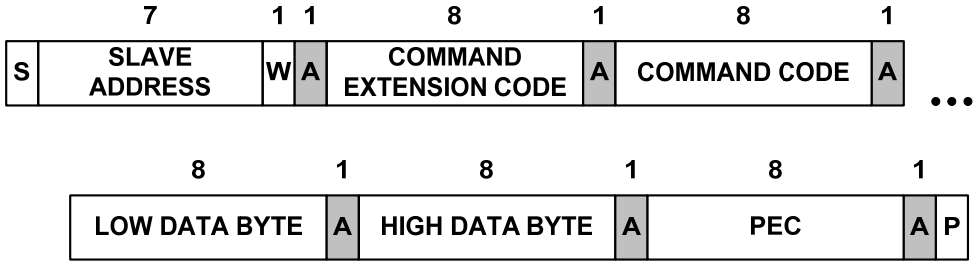


Figure 11. Extended Command Write Word Protocol With PEC

5.6.3. Zone Read Protocol

The Zone Read Protocol is used to query all or some of the devices on a bus, including their pages. A zone is defined as a subset of the PMBus addresses and pages of all devices on a bus. The Zone Read Protocol uses the SMBus address 28h.

Zones are assigned and activated by using the ZONE_CONFIG and ZONE_ACTIVE commands. ZONE_CONFIG sets the device's assigned zone. ZONE_ACTIVE sets the targeted zone for Zone Reads and Zone Writes. Refer to the PMBus Section II specification for description of the ZONE_CONFIG and ZONE_ACTIVE commands.

Zone Read allows a system host to query more than one PMBus device in a single transaction. The system host issues a command to all PMBus devices and each

device attempts to respond with its data and address. Natural SMBus 'bit dominance' arbitration (SMBus Specification, Section 5.3.2) determines which device succeeds in sending its data.

In all responses to Zone Read, the slave device will include its bus address and a page number. The returned slave address allows the system host to determine which device answered the call. The 7 bit device address is always returned in bits [7:1]. If bit [0] is zero, the data returned is from a device that either has no pages or from a device that has pages but the data is a global value from the device. If bit [0] is zero, the returned page data will be FFh. If bit [0] is one, the response is coming from a page within a device with pages.

Systems implementing Zone Read will benefit from using the SMBALERT# connection between the master and all devices. The alert pin is used as a signal to the bus master to begin a query process. Zone Read does not alter the behavior of the Alert Response Address protocol as defined in the SMBus specification.

Support for Zone Read is optional. To support Zone Read, a PMBus device must support clock stretching and repeated START conditions as defined in the SMBus specification.

For examples on how to use the Zone Read protocol, please see the PMBus Application Note AN001 [R01].

5.6.3.1. Zone Read Initiation

Zone Read is initiated by the system host with a write to address 28h. The second byte sent by the system host is the Command Control Code. The Command Control Code, described in Section 5.6.3.2, governs the response of the PMBus devices.

The third byte is either a PMBus command code, as shown in Figure 12, or a status mask, as shown in Figure 13 depending on the contents of the Command Control Code.

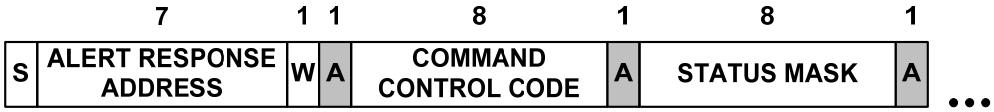


Figure 12. Zone Read Initiation with PMBus Command Code

Figure 13. Zone Read Initiation with Status Mask

The master is allowed to transmit a stop condition at any time during the transaction. This is especially applicable when the master knows when the last device has responded. Otherwise the master can allow the bus to timeout if no more devices are responding. Allowing the bus to timeout will be necessary when a device address discovery is performed.

5.6.3.2. Command Control Code

A Command Control Code byte is used to manage multiple calls. For example, the Command Control Code may specify that the device with successful arbitration ignore the next Zone Read. This enables a system host to read the address 28h multiple times to allow each device an opportunity to respond in an order relative to bit dominance arbitration.

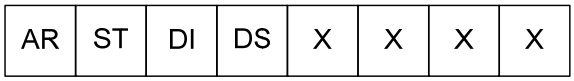


Figure 14. Command Control Code

5.6.3.2.1 Command Control Code Bit [7]: AR (All Respond)

The AR bit defines the All Respond mode.

When AR = 0, all devices respond only once with their data and address, but only one will win the bit dominance arbitration. Thus arbitration success and prioritization are set by the value of the data that is sent by the responding PMBus devices.

When AR = 1, all devices respond with their data and address to every write to the address 28h until they are successful in sending information to the system host. The responses become appended in succession until all devices have successfully arbitrated their response. When a device loses arbitration, it will try again after the last successfully arbitrating device is finished. Each device that is successful in transmitting its response must set an internal Ignore Request flag and ignore subsequent responses until the successful device's Ignore Request flag is reset. When the master receives a NACK or bus timeout, it means no more devices need to arbitrate. The master then resets the Ignore Request flag in all of the PMBus devices by writing to address 28h with AR =1 and ST =1 (command code C0h, indicating that all devices should respond) followed by the status mask FFh (which instructs the PMBus devices to reset their internal Ignore Request flag).

5.6.3.2.2 Command Control Code Bit [6]: ST (STatus response)

The ST bit defines the status response mode.

When ST = 0, all devices respond with the requested PMBus command data and their address.

When ST = 1, all devices respond with the upper byte of their STATUS_WORD and address. In this status response mode, the PMBus Command Code is repurposed as a status mask byte. This mask byte is applied to the upper byte of STATUS_WORD to control the priority of the returned status information. The upper byte of STATUS_WORD is used to ensure communication of all faults and PG status.

Table 2: Setting AR and ST Bits

AR	ST	Description
0	0	Return data / address from single, arbitration winning device

AR	ST	Description
0	1	Return status / address from single, arbitration winning device
1	0	Sequentially return data / address from all devices in arbitration order
1	1	Sequentially return status / address from all devices in arbitration order

The order of arbitration is modified by the DI and DS bits.

5.6.3.2.3 Command Control Code Bit [5]: DI (Data Inversion)

The DI bit defines data bit inversion.

If DI = 0 the data is returned with no inversion of the data bits.

If DI = 1, then each bit in the returned data is inverted. This bit allows the master to alter the arbitration order of responding devices.

5.6.3.2.4 Command Control Code Bit [4]: DS (Data byte Significance)

The DS bit defines data byte order.

IF DS = 0, the byte order is per the SMBus standard (least significant byte first).

If DS = 1, then the data bytes for a read word or a block read transaction are returned starting with the most significant byte and ending with the least significant byte. This bit allows the master to alter the arbitration order of responding devices.

5.6.3.2.5 Command Control Code Bits [3:0]

Bits [3:0] of the Command Control Code are reserved for future use.

5.6.3.3. Device Address Discovery

Slave address discovery can be done by setting AR=1 and ST=1. In this mode each compatible device returns its status and address through arbitration. The PMBus Command Code can be set to any value except FFh for this function.

5.6.3.4. Status Mask Byte

The Status Mask Byte is used to control the status information returned to the system host.

A value of 1 in any bit position of the status mask will cause the correlating status bit to be masked (read as zero). If the Status Mask Byte has any value other than FFh, the returned data byte is the bitwise ANDed value of the upper byte of the STATUS_WORD and the inversion of the bits in the Status Mask Byte. The returned data is calculated as:

Returned Data = STATUS_WORD[8:15] & INV(Status Mask).

If the Status Mask byte is set to FFh, this instructs all devices to reset their Ignore Request flags.

If the Status Mask byte is set to FFh, this instructs all devices to reset their Ignore Request flags.

5.6.4. Zone Write Protocol

The Zone Write Protocol broadcasts data to some or all PMBus devices simultaneously. A zone is defined as a subset of the PMBus addresses and pages of all devices on a bus. Zones are assigned and activated by using the ZONE_CONFIG and ZONE_ACTIVE commands. ZONE_CONFIG sets the device's assigned zone. ZONE_ACTIVE sets the targeted zone for Zone Reads and Zone Writes. Refer to the PMBus Section II specification for description of the ZONE_CONFIG and ZONE_ACTIVE commands.

The data of a command sent to the Zone Write address 37h, will be written into the devices that have matching ZONE_CONFIG and ZONE_ACTIVE values for a write. The Zone Write command is synchronously executed within all zone devices upon assertion of the stop bit at the end of the transaction.

It is up to the device manufacturer to specify how a zone relates to a device that has pages within it.

The Zone Write Protocol uses the SMBus address 37h.

For examples on how to use the Zone Write protocol, please see the PMBus Application Note AN001 [R01].

5.7. Exceptions To The SMBus Specification

None in this revision of the PMBus specification.

6. Addressing

PMBus devices use seven bit addresses as described in the SMBus specification [A03].

Addresses described as reserved in the SMBus specification ([A03]) or the I²C specification [A04] may not be used.

Physical addresses are programmed through pins. PMBus devices are not required to be able to support the entire seven bit address space through pin programming. The addresses available through pin programming are left to the PMBus device manufacturer. How a device's address is set shall be described in the device's product literature.

7. Communication From PMBus Devices To The Host

7.1. Communicating Over The Bus

As an option, PMBus devices may temporarily become bus masters and communicate with the host through the SMBus Host Notify Protocol defined in SMBus specification [A03].

The contents of the two data bytes shall be the same as the contents of the data bytes for the PMBus STATUS_WORD command (PMBus specification, Part II [A01]).

7.2. Communicating With An Interrupt Signal

As an option, PMBus devices may notify the host that they want to communicate with the host by asserting the SMBALERT# signal, as described in Appendix A of the SMBus specification, [A03].

8. Hardwired Signals

8.1. Electrical Interface

The electrical interface for hardwired signals shall comply with the High Power DC Electrical Interface for 100 kHz devices of the SMBus specification [A03].

The only exception to this are the pins used to set the physical address. If the electrical interface to address pins is not in compliance with the High Power DC Electrical Interface for 100 kHz devices of the SMBus specification, [A03], the electrical interface shall be described in the PMBus device's product literature.

8.2. Timing

There are no specific requirements on when a PMBus device must respond to a change in state of a hardwired signal.

8.3. Control Signal (CONTROL)

The CONTROL signal is an input signal on a PMBus device. It is used to turn the device on and off in conjunction with commands received via the serial bus. For more information, see Part II of the PMBus specification [A01].

It can be configured as an active high or active low signal through the ON_OFF_CONFIG command (PMBus specification, Part II [A01]).

The electrical interface for the CONTROL signal shall comply with the High Power DC Electrical Interface for 100 kHz devices of the SMBus specification [A03].

This signal is optional but recommended.

8.4. Write Protect (WP)

The PMBus protocol supports the use of optional Write Protect (WP) signal inputs.

There may be more than one Write Protect input signal with each signal protecting a different type or different portions of memory.

If the WP input is present, then no updates to any internal memory is allowed when the WP input is high or open. Updates are permitted only when the WP signal is low.

The electrical interface for WP signals shall comply with the High Power DC Electrical Interface for 100 kHz devices of the SMBus specification [A03].

8.5. Other Pins

PMBus devices may use pins for programming or configuration. The function and electrical interface of any such pins shall be described in the device's product literature.

Examples of such pins are a RESET pin or pins that are used to set the output voltage to the high or low margin voltages.

Pins that provide a binary input (high or low) shall have an electrical interface that meets requirements of the High Power DC Electrical Interface for 100 kHz devices of the SMBus specification [A03].

For pin functions with an equivalent command (PMBus specification, Part II [A01]), the command received from the bus will override the pin programming. See the description

in Part II of the PMBus specification [A01] on how a PMBus device configures its operating parameters for more information.

9. Accuracy

Each PMBus device will specify in its product literature the accuracy with which the output voltage and other parameters can be set and reported.

10. Firmware Updates

PMBus devices can, as an option, support upgrading its firmware via the SMBus interface. The methods of such updates are left to the discretion of the device manufacturers.

APPENDIX I. Summary Of Changes

DISCLAIMER: The section is provided for reference only and for the convenience of the reader. No suggestion, statement or guarantee is made that the description of the changes listed below is sufficient to design a device compliant with this document.

A summary of the changes made in Part I of the PMBus specification from Revision 1.2 to this revision, 1.3, is given below. This is not an exact list of every change made between the two documents; rather, it is a summary of the changes deemed significant by the editor. For those PMBus Adopters interested in a more detailed accounting of the changes, the PMBus Specification Change And Errata Log is available from the Adopter's section of the PMBus Website.

- In general, the basis of the Part I was changed from Version 2.0 of the SMBus specification to Version 3.0. This includes support for different possible maximum bus speeds (100 kHz, 400 kHz, and 1 MHz), additional protocols (e.g. READ/WRITE 32), and the ability to control the devices in a system based on their assigned zone.
- The Zone Read (Section 5.6.3) and Zone Write (Section 5.6.4) protocols were added
- There are general re-writes and re-arrangement of the sections to improve the flow of the document.