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

Revision 1.0

28 March 2005

www.powerSIG.org

Filename: PMBus_Specification_Part_I_Rev_1_ 0_20050324.doc
Last saved: 28 Mar 2005, 15:30

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, 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.

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, voltage tracking, and clock signals for interleaving and 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 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

[A02] SBS Implementers Forum, *System Management Bus (SMBus) Specification*, Version 1.1, 11 December 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, 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 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. Byte And Bit 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.

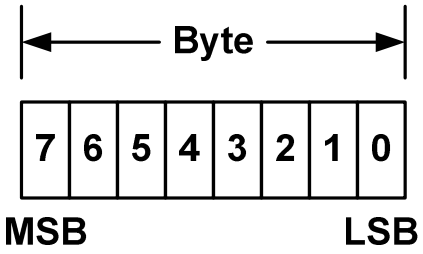
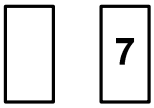
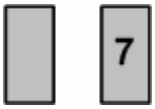
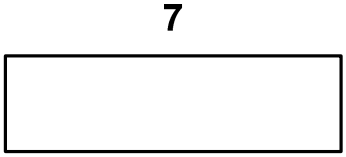
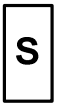
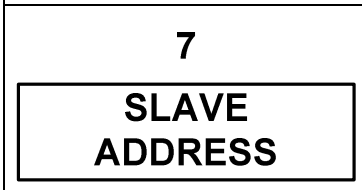
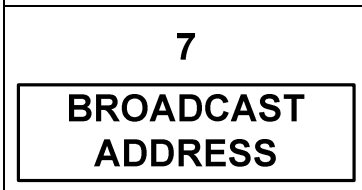


Figure 1. Bit Order Within A Byte

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

Symbol	Meaning
	An Acknowledge (ACK) condition sent from a slave device
	A Not Acknowledge (NACK) condition sent from a slave device
	A STOP condition sent by a bus master device
	The first seven bits of the address byte, generally corresponding to the physical address of the device.
	The eighth bit of the address byte with a value of 1, indicating the device is being addressed with a read.
	The eighth bit 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 eighth bit 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 [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

Term	Definition
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.
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.

Term	Definition
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. 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 [A01],
- A device must either accept, acknowledge and execute a PMBus command as specified in [A01] or reject the command by either:
  - NACKing the command or one of the data bytes as they are being received, or
  - If a device cannot support NACKing the command code or data bytes as they are being received,, then the device shall:
    - ACK the command code byte and all data bytes,
    - Set the CML bit in the STATUS_BYTE register (See [A01]),
    - Support reading the STATUS_BYTE,
    - Set the appropriate bit in the STATUS_CML register (See [A01]) (if that register is supported),
    - Support reading the STATUS_CML register (See [A01]) (if the STATUS_CML register is supported), and
    - Notify the host through the SMBALERT# signal (if supported) or the HOST NOTIFY process (if supported).

4.2. Unassisted Start Up And Operation

PMBus devices, upon application of power to the control circuit, must safely start and operate without communication from the serial bus.

5. Transport

5.1. SMBus, Version 1.1

PMBus devices must use the System Management Bus (SMBus), Version 1.1 [A02], for transport with the extensions and exceptions listed below.

5.2. Extensions To The SMBus, Version 1.1 Specification

5.2.1 Block Write-Block Read Process Call

To support certain commands of the PMBus Command Language ([A01]), PMBus devices must support the “Block write-block-read process call” described in Section 5.5.8 of Version 2.0 of the SMBus Specification ([A03]).

5.2.2 Host Notify Protocol

PMBus devices may support the Host Notify Protocol described in Section 5.5.9 of Version 2.0 of the SMBus Specification ([A03]).

If a PMBus device supports the Host Notify Protocol, the two data bytes sent to the host are the same as the data bytes returned by the STATUS_WORD command ([A01]),

5.2.3 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 received in one transmission. When the devices detect the STOP condition that ends the command, they all begin executing the command they received.

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

The Group Command Protocol must not be used with commands that require the receiving device to respond with data, such as the READ_STATUS_BYTE command ([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 MARGIN_HIGH_VOLTAGE 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 MARGIN_LOW_VOLTAGE while all of the other devices were instructed to set their output voltage to the programmed MARGIN_HIGH_VOLTAGE 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.

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.

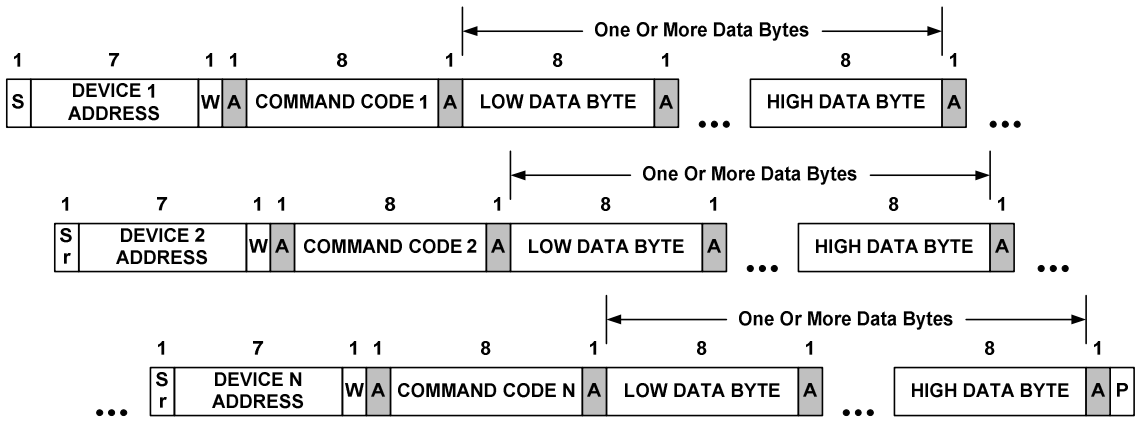


Figure 2. Group Command Protocol Without PEC

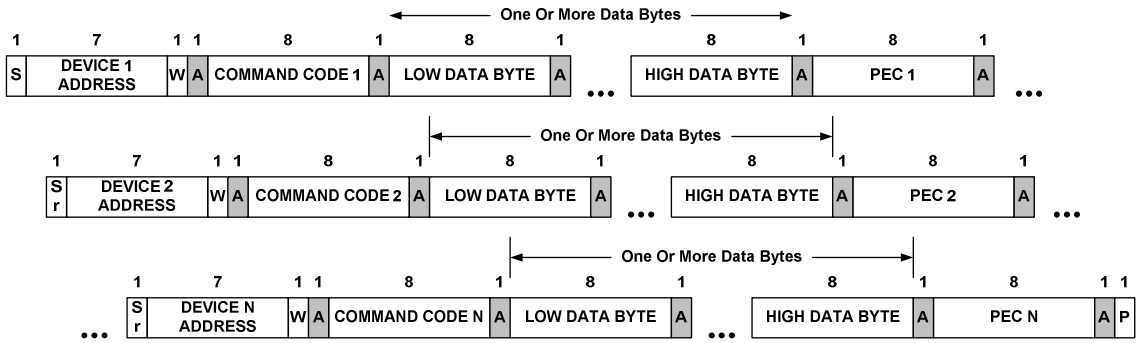


Figure 3. Group Command Protocol With PEC

5.2.4 Extended Command: Read/Write Byte

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 command to be executed. The details of the protocol for commands that read or write one byte are illustrated below.

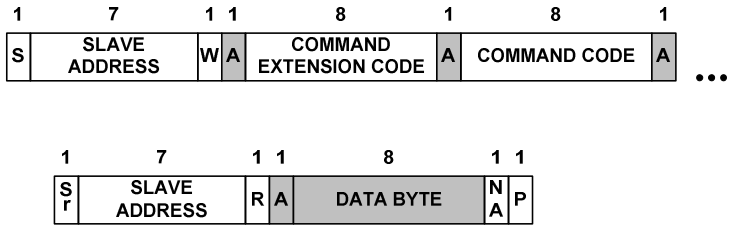


Figure 4. Extended Command Read Byte Protocol

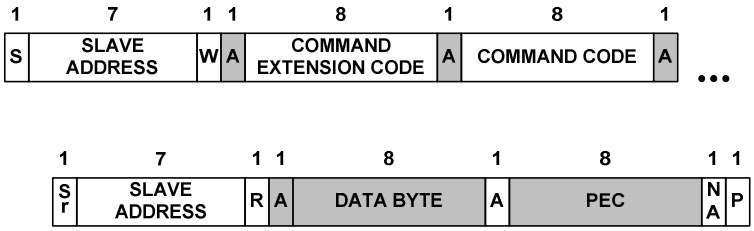


Figure 5. Extended Command Read Byte With PEC

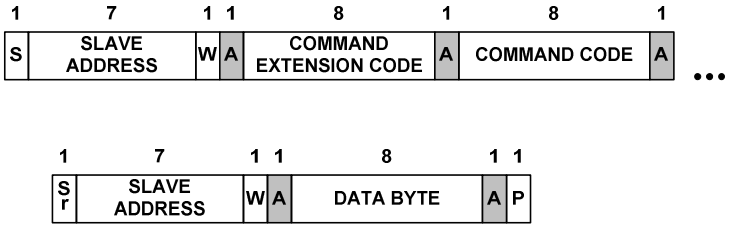


Figure 6. Extended Command Write Byte Protocol

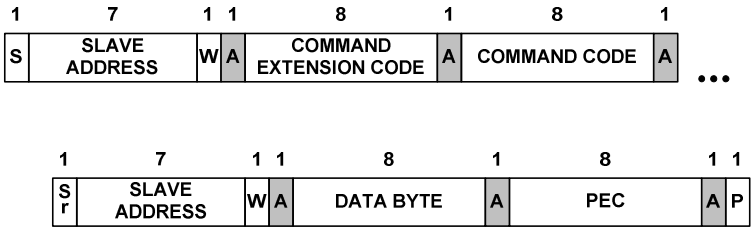


Figure 7. Extended Command Write Byte Protocol With PEC

5.2.5 Extended Command: Read/Write Word

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 command to be executed. The details of the protocol for commands that read or write one byte are illustrated below.

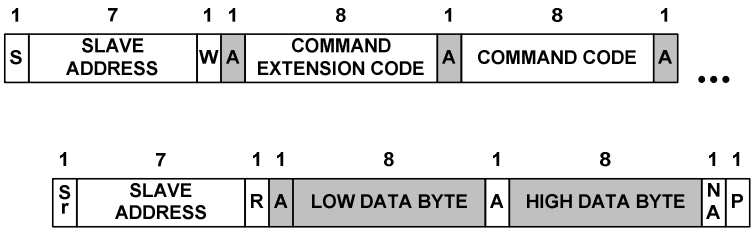


Figure 8. Extended Command Read Word Protocol

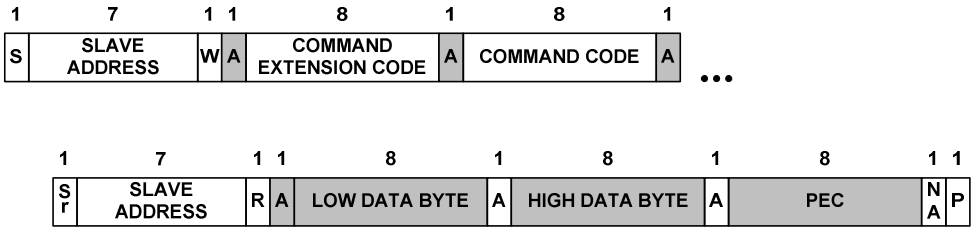


Figure 9. Extended Command Read Word Protocol With PEC

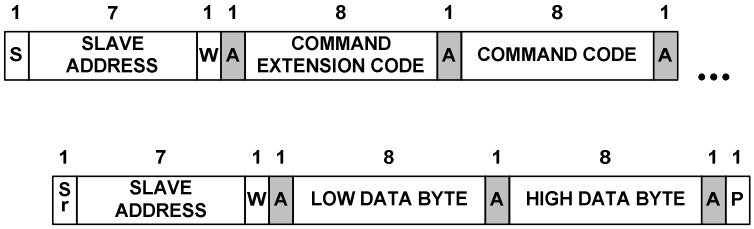


Figure 10. Extended Command Write Word Protocol

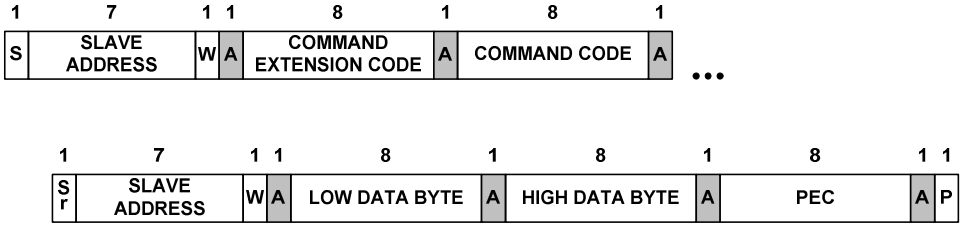


Figure 11. Extended Command Write Word Protocol With PEC

5.3. Exceptions To The SMBus, Version 1.1 Specification

Block writes and reads ([A02], Section 7.5.7) are permitted to have up to 255 data bytes.

6. Addressing

PMBus devices use seven bit addresses as described in [A02].

Addresses described as reserved in [A02] or [A04] may not be used.

Physical addresses are programmed through pins. PMBus devices are not required to be able to address 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. Packet Error Checking

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

8. Communication From PMBus Devices To The Host

8.1. Communicating Over The Bus

As an option, PMBus devices may temporarily become bus masters and communicate with the host through the protocol defined in Section 7.6 of [A02].

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

8.2. 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 [A02].

9. Hardwired Signals

9.1. Electrical Interface

The electrical interface for hardwired signals shall be as described in Section 8 of [A02].

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 to Section 8 of [A02], the electrical interface shall be described in the PMBus device's product literature.

9.2. Timing

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

9.3. Control Signal (CONTROL)

The CONTROL signal is an input signal on a power converter. It is used to turn the unit on and off in conjunction with commands received via the serial bus. For more information, see [A01].

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

The electrical interface for the CONTROL signal shall be as described in Section 8 of [A02].

This signal is optional but recommended.

9.4. 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 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 be as described in Section 8 of [A02].

9.5. 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 the meets the requirements of Section 8 of [A02].

For pin functions with an equivalent command ([A01]), the command received from the bus will override the pin programming. See the description in [A01] on how a PMBus device configures its operating parameters for more information.

10. 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.

11. 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.