Using The PMBus™ Protocol

Robert V. White
Staff Engineer
Artesyn Technologies
Chair, PMBus™ Specification Working Group
12 October 2005
Presentation Overview

- What Is The PMBus™?
- PMBus Basics
- Using The PMBus In The Lab…
- Implementing PMBus
- Command Language Overview
- Data Formats
- Setting The Output Voltage

- On/Off Control
- Sequencing
- Status Reporting
- Fault Management And Reporting
- Monitoring Voltage, Current And Temperature
- Some Other Topics (As Time Allows)
What Is PMBus?
A Standard Way To Communicate With Power Converters Over A Digital Communications Bus
Configure

System

PMBus Device

Configure

PMBus Device
System

Configure

Control

Monitor

PMBus Device

Configure

Control

Monitor

PMBus Device
System Maintenance Processor
– Or –
Spare Gates In An FPGA
– Or –
Laptop Computer
– Or –
Dedicated Controller IC
– Or –
General Purpose Microcontroller
– Or –
Automatic Test Equipment
PMBus Is An Open Standard

• Owned By The System Management Interface Forum (SM-IF)
  – SM-IF Membership Is Open To All
• Royalty Free
• Released Specifications Freely Available
• Works With All Types Of Power Converters
  – AC-DC Power Supplies
  – Isolated DC-DC And Bus Converters
  – Non-Isolated Point-Of-Load Converters
  – Microprocessor Power Converters
PMBus: What It Is Not

• Not A Product Or Product Line
• Not A Standard For A Power Supplies Or DC-DC Converters
  – No Form Factor, Pin Out, Efficiency, Etc.
  – Alliances Like POLA And DOSA Will Define
• No Converter-To-Converter Communication
  – Such As Current Share And Analog Voltage Tracking
  – Left To The IC And Power Supply Manufacturers
  – Including These Would Inhibit Future Innovation
Some Basic PMBus Requirements

- PMBus Devices Must Start Up Safely Without Bus Communication
- PMBus Devices Can Be Used With Or Without A Power System Manager/Controller
- PMBus Devices Support “Set And Forget”
  - Can Be Programmed Once At Time Of Manufacture
  - Then Operate Forevermore Without Bus Communication
- Defaults From Either/Or
  - Non-Volatile Memory
  - Pin Programming
Who Is PMBus?
PMBus Adopters

- Alliance Semiconductor
- Artesyn Technologies
- Emerson/Astec
- International Rectifier
- Intersil Corporation
- Magnetek, Inc.
- Micro Computer Control Corporation (MCC)
- Microchip Technology
- Primarion
- Silicon Laboratories
- Summit Microelectronics, Inc.
- Texas Instruments
- Tyco Electronics Corp.
- Volterra Semiconductor Corporation
- Zilker Labs
System Management Interface Forum, Inc.
System Management Interface Forum, Inc.

SM-IF Membership Open To Any And All

www.powerSIG.org
Use Of PMBus™ Logo And Trademark

• Only Adopters Are Permitted To Use The PMBus™ Trademarks And Logo For Commercial Purposes
  – Commercial Purpose Is Anything Related To The Sale Of Products And Services
  – Helps Assure That PMBus Device Manufacturers Understand The Specification

• The Press May Use The Trademarks And Logo In Articles That Do Not Promote Products Or Services
Specification Structure

• Part I – Physical Layer And Transport
  – Bus & Protocols
  – Discrete Signals
  – Electrical Levels

• Part II – Command Language
  – Commands
  – Data Formats
  – Fault Management “Tutorial”
  – Status Reporting “Tutorial”
  – Information Storage
PMBus™ Connections
PMBus™ Connections

SYSTEM HOST/ BUS MASTER

Optional CONTROL Signal Is For On/Off Control
**PMBus™ Connections**

Optional SMBALERT# Signal Acts As An Interrupt Line And Activates The Alert Protocol
PMBus™ Connections

SYSTEM HOST/ BUS MASTER

Required Hardwired Pins To Set Physical Address

UNIT #1

UNIT #2

UNIT #N

SMBALERT# SIGNAL

CONTROL SIGNAL

SERIAL BUS DATA

SERIAL BUS CLOCK

WRITE PROTECT

PHYSICAL ADDRESS

WRITE PROTECT

PHYSICAL ADDRESS

WRITE PROTECT

PHYSICAL ADDRESS

CONTROL

DATA

CLOCK

WP
PMBus™ Connections

Optional Write Protect Pin To Prevent Unwanted Data Changes
What Is SMBus?

• A Long Existing Standard Bus
• Similar To I²C
  – Synchronous (Clock And Data Lines)
  – Byte Oriented
  – Same Addressing Scheme
  – Same Transmission Control
    • START, STOP, ACK, NACK
• Did Not Require Royalties To Philips
Why SMBus?

- Low Cost Like I²C
- More Robust Than I²C
- More Features Than I²C
  - SMBALERT# Line For Interrupts
  - Packet Error Checking (PEC)
  - Host Notify Protocol
- Generally Electrically Compatible With I²C
- Widely Used In Personal Computers And Small To Medium Servers
SMBus Improvements

• I²C “Noise Sensitivity” – Edge Triggering
  – False START: Timeouts Force Reset
  – False STOP: PMBus Devices Detect Failed Transmissions As Faults

• I²C “Noise Sensitivity” – Corrupt Data
  – Data Rates Permit Digital Filtering
  – Packet Error Checking (PEC)
  – Every Value That Can Be Written Can Be Read
SMBus Improvements

• I²C Slave Device Hangs Bus
  – Timeouts Force Device Reset

• I²C Requires Retrieving Device Information By Polling
  – SMBALERT# Line Acts As An Interrupt
  – Automatic, Lossless Bitwise Arbitration Of Simultaneous Requests

• I²C: 8 Devices Max Of One Type On A Bus
  – No Central Address Control Bureaucracy
  – Over 100 Device Addresses Available
SMBus Limitations

- SMBus and PMBus Specifications Say 100 kHz
  - I²C Says 400 kHz – Which Is Possible If SMBus Setup And Hold Times Are Obeyed
- Capacitance Is A Concern
  - No Explicit Maximum
  - Excessive Capacitance Causes A Violation Of Bus Timing By Slowing Rise Times
  - Minimize Capacitance In Layout
    - Stubs And Branches Not A Concern
  - See SMBus Specification For Details
SMBus Limitations

- SMBus and PMBus - 100 kHz
  - I²C Says 400 kHz – Which Is Possible If SMBus Setup And Hold Times Are Obeyed

- Capacitance Is A Concern
  - No Explicit Maximum
  - Excessive Capacitance Causes A Violation Of Bus Timing By Slowing Rise Times
  - Minimize Capacitance In Layout
    • Stubs And Branches Not A Concern
  - See SMBus Specification For Details

PMBus Going To 400 kHz In Revision 1.1
Addressing

• PMBus Devices Use A 7 Bit Address Per The SMBus Specification
  – Provides More Than 100 Possible Device Addresses After Allowing For Reserved Addresses

• No I²C Style Address Control Assignments Or Limitations

• PMBus Users Can Expect Device Addresses To Be Set By A Mix Of:
  – Hardwired Address Pins
  – High Order Address Bits Set By The PMBus Device Manufacturer
Addressing (cont’d)

• PMBus Device Manufacturers Will Trade Off Cost Of Pins Versus Address Flexibility
• Expect Device Makers To Offer Tri-State Pins Or Resistor Value Programming
• Examples Of The Possibilities
  – 3 Tri-State Pins => 27 Addresses
  – 1 Resistor Programmed Pin => 16–32 Addresses
Basic Packet Structure

**ADDRESS BYTE**  **COMMAND BYTE**  **DATA BYTE 1**

S    7  6  5  4  3  2  1  0  A  7  6  5  4  3  2  1  0  A

**DATA BYTE 2**

7  6  5  4  3  2  1  0  A

**DATA BYTE N**

7  6  5  4  3  2  1  0  A  7  6  5  4  3  2  1  0  A

**OPTIONAL PEC BYTE**

A  P

S  START Signal From Host System  0  READ/WRITE# Bit  A  ACKNOWLEDGE Signal From Converter  P  STOP Signal From Host System
Write Word Packet

- **SLAVE ADDRESS**: 0
- **COMMAND CODE**: A
- **LOW DATA BYTE**: A
- **HIGH DATA BYTE**: A
- **PACKET ERROR CHECKING (PEC)**: A
- **START Signal From Host System**: S
- **READWRITE# Bit**: 0
- **ACKNOWLEDGE Signal From Converter**: A
- **STOP Signal From Host System**: P
Write Word Packet

Packet Error Checking (PEC) is optional in the specification but is expected to be very popular with system OEMs!
Read Word Packet

- **SLAVE ADDRESS**: 0A
- **COMMAND CODE**: A

7 1 8

- **SLAVE ADDRESS**: 1A
- **LOW DATA BYTE**: A
- **HIGH DATA BYTE**: AP

7 1 8 8

- **START Signal From Host System**: S
- **Repeated START Signal From Host System**: Sr
- **READ/WRITE# Bit**: 0A
- **ACKNOWLEDGE Signal From Converter**: A
- **STOP Signal From Host System**: P
- **ACKNOWLEDGE Signal From Host System**: A
Read Word Packet

This Data Is Being Transmitted By The Slave Device To The Host
Using PMBus In The Lab

System Power Bus → Bus Converter → Local/On-Board Power Bus

- POL w/ PMBus → VOUT1
- POL w/ PMBus → VOUT2
- POL w/ PMBus → VOUT3

PMBus/USB Adapter

STD POL → VOUT4
STD POL → VOUT5

PMBus Interface IC

Analog Control Lines (Sense, Enable, Trim, Power Good)
Using PMBus In The Factory
Using PMBus In A System

- Input Power A
- Input Power B
- Front End Power Supply A
- Front End Power Supply B
- POL w/ PMBus
- VOUT1
- POL w/ PMBus
- VOUT2
- POL w/ PMBus
- VOUT3
- STD POL
- VOUT4
- STD POL
- VOUT5
- PMBus Interface IC
- Analog Control Lines (Sense, Enable, Trim, Power Good)
- N/C
- Set And Forget
Using PMBus In A System

- **Input Power A**
  - Front End Power Supply A
  - POL w/ PMBus
    - VOUT1

- **Input Power B**
  - Front End Power Supply B
  - POL w/ PMBus
    - VOUT2

- **System Maintenance Processor**
  - PMBus
  
  - **PMBus Interface IC**
    - STD POL
      - VOUT4
      - Analog Control Lines (Sense, Enable, Trim, Power Good)
    - STD POL
      - VOUT5
PMBus In A System

- **Input Power A**
  - Front End Power Supply A
  - POL w/ PMBus
    - VOUT1

- **Input Power B**
  - Front End Power Supply B
  - POL w/ PMBus
    - VOUT2
  - POL w/ PMBus
    - VOUT3

- **Remote System Maintenance Comm Bus**
  - (IPMI, RS-485, …)
  - PMBus
    - STD POL
      - VOUT4
    - STD POL
      - VOUT5

- **System Maintenance Processor**
  - Analog Control Lines (Sense, Enable, Trim, Power Good)

- **Local/On-Board Power Bus**
  - Remote Host
  - System Maintenance Processor

- **Front End Power Supply A**
  - POL w/ PMBus
  - VOUT1

- **Front End Power Supply B**
  - POL w/ PMBus
  - VOUT2
  - POL w/ PMBus
  - VOUT3

- **Remote Host**
  - System Maintenance Processor
  - Analog Control Lines (Sense, Enable, Trim, Power Good)
PMBus In A System

Local/On-Board Power Bus

POL w/ PMBus

VOUT1

POL w/ PMBus

VOUT2

POL w/ PMBus

VOUT3

STD POL

VOUT4

STD POL

VOUT5

Analog Control Lines (Sense, Enable, Trim, Power Good)

Remote System Maintenance Comm Bus (IPMI, RS-485,...)

System Power Bus

Bus Converter

Remote Host

System Maintenance Processor

PMBus Interface IC
PMBus In A System
PMBus In A System
PMBus In A System

System Power Bus

Bus Converter

System Maintenance Comm Bus (IPMI, RS-485, ...)

PMBus

Board Level Maintenance Processor

PMBus Interface IC

Local/On-Board

POL w/ PMBus

VOUT1

VOUT2

VOUT3

VOUT4

VOUT5

STD POL

STD POL

Analog Control Lines (Sense, Enable, Trim, Power Good)
PMBus Bridge To Other Buses

Bridge Device

System Bus Transmitter & Receiver

Translator System Language To/From PMBus Language

PMBus Transmitter & Receiver

PMBus Device

Extra Gates In An FPGA
– Or –
General Purpose Microcontroller
– Or –
Application Specific IC
Simple Electrical Bi-Directional Isolation
Bus Extensions

- Bus Master
- Multiplexor
- Slave Device
- Repeater
- Slave Device
- Switch
- Slave Device
- Switch
- Slave Device
- Switch
- Slave Device
- Switch
- Slave Device
Redundant Buses

Bus Master A

Bus Master B

PMBus A

PMBus B

Multiplexor

Tx/Rx

Internal Processor

Slave Device

Switch

Tx/Rx

Internal Processor

Slave Device

Switch

Tx/Rx
What’s Needed To Make A PMBus Device?

• Physical/Data Link Layer To Receive & Send Data Over The Bus
  – Plus CONTROL, SMBALERT#, WP, Address Pin Interface

• Memory
  – Received Configuration
  – Device Status And Parametric Information

• The Rest Of The Device
  – Such As Power Control And Conversion Circuits That Use/Supply Stored Information
  – Note That PMBus Does Not Depend On The Type Of Controller: Analog, Digital, Hybrid
PMBus Device Concept
How to Make A PMBus Device

Integrated Solution
- ASSP
- ASIC

Piece Part Solution
- Bus Interface
- Control & Monitor
  - ASIC
  - FPGA
  - GP Microcontroller
Making A PMBus Device

<table>
<thead>
<tr>
<th></th>
<th>Pro</th>
<th>Con</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSP</td>
<td>Minimal Investment</td>
<td>Hard To Differentiate</td>
<td>Low Technical Medium Economic</td>
</tr>
<tr>
<td>ASIC</td>
<td>Have It Your Way</td>
<td>$$$$$ Design Skills</td>
<td>Medium Technical High Economic</td>
</tr>
<tr>
<td>FPGA</td>
<td>Have It Your Way</td>
<td>$$ Design Skills</td>
<td>Medium Technical Medium Economic</td>
</tr>
<tr>
<td>General Purpose Microcontroller</td>
<td>Flexibility &amp; Added Functionality</td>
<td>$ Programming Skills</td>
<td>Medium Technical Medium Economic</td>
</tr>
</tbody>
</table>
Conceptual View Of Memory And Startup

1. Hard Coded Parameters
2. Pin Programmed Values
3. Default Store (Non-Volatile)
4. User Store (Non-Volatile)
5. Bus Communication

Operating Memory (Volatile)

- RESTORE_DEFAULT
- STORE_DEFAULT
- RESTORE_USER
- STORE_USER
Conceptual View Of Memory And Startup

Operating Memory

Conceptual Volatile Memory Store For Device’s Operating Parameters
Conceptual View Of Memory And Startup

At Device Power Up, Values Hard Coded Into The PMBus Device Are Loaded First Into The Operating Memory.
Conceptual View Of Memory And Startup

Next, Pin Programmed Values Are Loaded Into Operating Memory. This Overwrites Any Previously Loaded Values.
Conceptual View Of Memory And Startup

Default Values

Next, Values From The Non-Volatile Default Store (If Provided) Are Loaded. This Overwrites Any Previously Loaded Values.
Conceptual View Of Memory And Startup

Next, Values From The Non-Volatile User Store (If Provided) Are Loaded. This Overwrites Any Previously Loaded Values.
Conceptual View Of Memory And Startup

Bus Communication

Next, Values Sent Via The SMBus Are Loaded. This Overwrites Any Previously Loaded Values.
Used to store a snapshot of the device’s operating state. When power removed and restored, device can resume operation from its last programmed state.

Conceptual View Of Memory And Startup
PMBus–Host Interface

System Device
- Translator Human/System Language To/From PMBus Language
- Bus Transmitter & Receiver

PMBus Device
- Bus Transmitter & Receiver
- Translator PMBus Language To/From Native Device Language/Interface
- Power & Control Circuits
PMBus–Host Interface

System Device

Translator Human/System Language To/From PMBus Language

Bus Transmitter & Receiver

PMBus Device

Bus Transmitter & Receiver

Translator PMBus Language To/From Native Device Language/Interface

Power & Control Circuits

Someone Will Have To Write Code For This
PMBus–Host Interface

Calculation And Conversion For Input To D/A Converters And Output From A/D Converters
PMBus–Host Interface

Highly Platform Dependent!
PMBus/SMBus Interface

• “Bit Banging” With A General Purpose I/O Port On A Microcontroller
  – Can Be Done & Can Be Done Well
  – Pay Attention To The Specification
  – Timing Is Important

• Integrated Into Silicon
  – Many Microcontrollers Have An I²C Port That Can Be Used To Drive SMBus
  – Look For PMBus To Be Built Into I²Cs For Power Conversion And System Monitoring
Command Language

• Commands Consist Of:
  – A Command Code
    • 256 Command Codes (00h To FFh)
  – Zero Or More Data Bytes

• Command Code
  – Not A Register Location!
  – Devices Must Map Command Code To Memory Location Themselves

• Data Byte(s)
  – Defined In The Specification
Data Formats

• More Time Spent On This By Specification Working Group Than Any Other Topic!

• Challenges
  – Wide Range Of Values (Millivolts To Kiloamperes)
  – Wide Range Of Resolution
    • Millivolts For Microprocessors
    • Volts And Amperes For AC Power
  – Positive And Negative Values
  – Limited Computing Power In PMBus Devices
Data Format Choices

- **Resolution**
  - General Purpose: 10 Bits
  - Output Voltage Related: 16 Bits

- **Compute Power**
  - Low: Direct Mode
  - Higher: Literal

- **VRM Mode**
Data Format Choices

- General Purpose: 10 Bits
- Output Voltage Related: 16 Bits
- Low: Direct Mode
- Higher: Literal

Refers to compute power needed in the PMBus device.
Literal Format

- \( X = Y \cdot 2^N \leftrightarrow Y = X \cdot 2^{-N} \)
  - \( X \) = “Real World” Value (Example: 3.3)
  - \( Y \) = Binary Value Sent Over The PMBus
  - \( N \) = Scale Factor

- \( Y \) (Binary Signed Integer)
  - General Purpose Case: 11 Bits
  - Output Voltage Related Data: 16 Bits

- \( N \) (Binary Signed Integer)
  - 5 Bits In Both General Purpose Data And Output Voltage Related Data
Literal Format

\[ X = Y \cdot 2^N \]

\[ Y = X \cdot 2^{-N} \]

- X = “Real World” Value (Example: 3.3)
- Y = Binary Value Sent Over The PMBus
- N = Scale Factor

- Y (Binary Signed Integer)
  - General Purpose Case: 11 Bits
  - Output Voltage Related Data: 16 Bits
- N (Binary Signed Integer)
  - 5 Bits In Both General Purpose Data And Output Voltage Related Data

NOTICE!
This Is The Form That Will Appear In The PMBus Specification Revision 1.1
This Is “Backwards” From What Is In Specification 1.0 Section 7
Literal Format

- Two Ways to Think Of This Format
- “Binary Floating Point”

\[ X = Y \cdot 2^N \]

- \( Y = \text{Mantissa} \)
- \( N = \text{Exponent} \)
Literal Format

- Two Ways to Think Of This Format
- “Binary Floating Point”
  - $Y = \text{Mantissa}$
  - $N = \text{Exponent}$
- Number Of LSBs

$$X = Y \cdot 2^N$$

$Y = \text{Number Of LSBs}$

$2^N \text{ Equals Size Of LSB}$
Literal Format

- Two Ways to Think Of This Format
  - “Binary Floating Point”
    - $Y = \text{Mantissa}$
    - $N = \text{Exponent}$
  - Number Of LSBs

$$X = Y \cdot 2^N$$

Example: $N = -10$

$$2^{-10} = \frac{1}{1024} = 9.766 \times 10^{-4} \Rightarrow 977 \text{ microunits/bit}$$

$Y = \text{Number Of LSBs}$

$2^N \text{ Equals Size Of LSB}$
General Purpose Literal Format

- 10 Bit Resolution
- Wide Range Of Values Possible
  - Maximum Positive: $1023 \times 2^{15} = 33.5217 \times 10^6$
  - Minimum Value: $\pm 1 \times 2^{-16} = \pm 1.526 \times 10^{-5}$
  - Maximum Negative: $-1024 \times 2^{15} = -33.5544 \times 10^6$

General Purpose (10 Bit) Literal Format

<table>
<thead>
<tr>
<th>Data Byte High</th>
<th>Data Byte Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 6 5 4 3</td>
<td>2 1 0 7 6 5 4 3 2 1 0</td>
</tr>
</tbody>
</table>

MSB  N  MSB  Y
Example Conversion With Maximum Resolution

• Given $X = 3.3\, \text{V}$; Calculate $Y$ And $N$

• Maximum Resolution With Largest Possible $Y$
  $- Y_{\text{max}} = 1023$

• Largest Possible $Y \Rightarrow$ Smallest $2^{-N}$
  $- $ Smallest LSB
  $- $ Largest $|N|$
Example Conversion With Maximum Resolution

- Start By Finding $N$
- Can Solve Directly
  - But Complicated
- Or: Start By Dividing $Y_{\text{max}}$ By $X$
- Examine Result And Find Largest $2^{-N}$ That Is Less Than The Result
  - This Gives $N$
- Multiply $X$ By $2^{-N}$ To Get $Y$
  - Convert To 11 Bit Signed Binary Integer

\[
N = \text{int} \left( \log_2 \left( \frac{X}{Y_{\text{MAX}}} \right) \right) = \text{int} \left( \frac{\ln \left( \frac{X}{Y_{\text{MAX}}} \right)}{\ln 2} \right)
\]

\[
\frac{Y_{\text{MAX}}}{X} = \frac{1023}{3.3} = 310.0
\]

\[
\text{max}(2^{-N}) < 310.0 \Rightarrow 256
\]

\[
\Rightarrow N = -8 = 11000b
\]

\[
3.3 \times 2^{-N} = 3.3 \times 2^8
\]

\[
= 3.3 \times 256
\]

\[
= 844.8 \Rightarrow 845
\]

\[
845 \Rightarrow 011010011101b = Y
\]
Literal Mode
Result Sent Over The PMBus

Data Byte High  Data Byte Low

7 6 5 4 3
1 1 0 0 0

2 1 0 7 6 5 4 3 2 1 0
0 1 1 0 1 0 0 1 1 0 1

MSB  N  MSB  Y
Literal Mode As A Non-Integer Binary Value

- Can Think Of N As Telling The Device How Many Binary Places To Move The Binary Point: 11.0100110b
Literal Mode As A Non-Integer Binary Value

- Can Think Of N As Telling The Device How Many Binary Places To Move The Binary Point: 11.0100110b

\[ N = -8 \]

Move Binary Point 8 Places To The Left
Literal Mode As A Non-Integer Binary Value

- Can Think Of $N$ As Telling The Device How Many Binary Places To Move The Binary Point: $11.0100110b$

- Can Also Think Of This As A Sum Of Powers Of 2

\[
\begin{align*}
1 \cdot 2 + 1 \cdot 1 + 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + \ldots \\
0 \cdot \frac{1}{8} + 0 \cdot \frac{1}{16} + 1 \cdot \frac{1}{32} + \ldots \\
1 \cdot \frac{1}{64} + 0 \cdot \frac{1}{128} + 1 \cdot \frac{1}{256}
\end{align*}
\]
Literal Mode As A Non-Integer Binary Value

- Can Think Of N As Telling The Device How Many Binary Places To Move The Binary Point: 11.0100110b

- Can Also Think Of This As A Sum Of Powers Of 2

- Result = 3.0078
Literal Mode As A Non-Integer Binary Value

- Can Think Of N As Telling The Device How Many Binary Places To Move The Binary Point: 11.0100110b

- Can Also Think Of This As A Sum Of Powers Of 2

\[ 1 \cdot 2 + 1 \cdot 1 + 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + \ldots \]

\[ 0 \cdot \frac{1}{8} + 0 \cdot \frac{1}{16} + 1 \cdot \frac{1}{32} + \ldots \]

\[ 1 \cdot \frac{1}{64} + 0 \cdot \frac{1}{128} + 1 \cdot \frac{1}{256} \]

- Result = 3.0078

Error Of 7.8 mV (0.0236%) Is From Rounding 848.8 To 845 And Quantization Errors

Data Byte High Data Byte Low
Literal Mode As
LSB Size & Number Of LSBs

• Also Can Think Of This As
  – The LSB Size \(2^N\)
  – The Number Of LSBs \(Y\)
• \(\text{LSB} = 2^N = 2^{-8}\)
  \[= 0.00390625\]
• \(Y = \text{Number Of LSBs}\)
  \[= 845\]
• \(X = 845 \times 0.00390625\)
  \[= 3.30078125\]
Example Decode

- Received Value:

  11100011 01100111

  High Byte (Received Second)

  Low Byte (Received First)
Example Decode

• Received Value: 11100011 01100111

• Separate Into $N$ And $Y$

\[
N = 11100_{\text{b}} = -4
\]
\[
Y = 01101100111_{\text{b}} = 871
\]
Example Decode

• Received Value: 11100011 01100111
• Separate Into N And Y 11100 01101100111
  \[N = 11100_{\text{b}} = -4\]
  \[Y = 01101100111_{\text{b}} = 871\]
• Calculate X \[X = Y \cdot 2^N = 871 \times 2^{-4}\]
  \[= \frac{871}{16} = 54.438\]
Example Decode

• Received Value: 11100011 01100111

• Separate Into $N$ And $Y$
  
  \[
  N = 11100b = -4 \\
  Y = 01101100111b = 871
  \]

• Calculate $X$

\[
X = Y \cdot 2^N = 871 \times 2^{-4} = \frac{871}{16} = 54.438
\]

• Original Value: 54.46
  – Error: 22 mV => 0.040%
Error: We Got Lucky!

• Suppose Full Scale Was 60 V
• Resolution:  
  \[ 60 \text{ V} \div 1023 = 58.65 \text{ mV/bit} \]

• Some Applications, Such As A Telecomm Rectifier, Need A Much Finer Resolution
  – Typically 10-20 mV/bit

• But Range Of Interest Is Not 0 V to 60 V, More Like 42 V To 58 V
Scaling With Offset

- $X_{\text{MAX}}$
- $X_{\text{MIN}}$
- $Y_{\text{MAX}}$
- $Y_{\text{MIN}}$

Human/Analog/“Real World” Value

PMBus Digital Value

$2^N - 1$
$2^N - 2$
$2^N - 3$
$2^N - 4$

3
2
1
0
Scaling With Offset

Can We Map From: Real World Range Of Values Of Interest

Human/Analog/“Real World” Value

$X_{MAX}$

$X_{MIN}$

$Y_{MIN}$

$Y_{MAX}$

PMBus

Digital Value

Can We Map From: Real World Range Of Values Of Interest

Can We Map From: Real World Range Of Values Of Interest
Scaling With Offset

To:
The Range Of Values Accepted By The PMBus Device?

Human/
Analog/
“Real World” Value

X_{MIN}

\text{To:}

\text{PMBus Digital Value}

Y_{MIN}

Y_{MAX}

2^N-1
2^N-2
2^N-3
2^N-4

PMBus

3
2
1
0

To:
The Range Of Values Accepted By The PMBus Device?
Scaling With Offset

This Is One Of The Motivations For The Direct Mode/Format

PMBus Digital Value

X_{MAX}  \quad  X_{MIN}

Y_{MAX}  \quad  Y_{MIN}

2^N-1  \quad  2^N-2  \quad  2^N-3  \quad  2^N-4

Value

Human/Analog/"Real World" Value
More Direct Mode Motivation

No "Floating Point" Computation! Value From PMBus Is Used Directly
Direct Mode Equation

- The Direct Mode Uses An Equation As Follows:

\[ Y = (mX + b) \cdot 10^R \]

- Where:
  - \( Y \) Is The Value Transmitted To Or Received From The PMBus Device (16 Bits, Signed)
  - \( X \) Is The “Human” Value To Be Encoded
  - \( m \) Is The Scaling Coefficient (16 Bits, Signed)
  - \( b \) Is The Offset Coefficient (16 Bits, Signed)
  - \( R \) Is The Scaling Coefficient (8 Bits, Signed)
Direct Mode Equation

• The Direct Mode Uses An Equation As Follows:

\[ Y = RYm Xb + i \]

• Where:
  - \( Y \) is the value transmitted to or received from the PMBus device
  - \( X \) is the “human” value to be encoded
  - \( m \) is the scaling coefficient (16 bits, signed)
  - \( b \) is the offset coefficient (16 bits, signed)
  - \( R \) is the scaling coefficient (8 bits, signed)

**NOTICE!**

This is the form that will appear in the PMBus specification revision 1.1.

This is “backwards” from what is in specification 1.0 section 7.

- \( b \) is the offset coefficient (16 bits, signed)
- \( R \) is the scaling coefficient (8 bits, signed)
Direct Mode: $m, b$ And $R$

- $m, b$ And $R$ Are Known As The Coefficients
- They Are Supplied By The PMBus Device Manufacturer
- Preferred: Coefficients Stored In The Device And Retrieved By The Host With The COEFFICIENTS Command
- Alternative: Coefficients Are Provided In The Product Literature (Data Sheet, Application Note)
Calculating The Coefficients

• Problem
  – 3 Unknowns \((m, b, R)\)
  – 2 Constraints

• The Two Constraints
  – \(X_{\text{min}} \Rightarrow Y_{\text{min}}\) And \(X_{\text{max}} \Rightarrow Y_{\text{max}}\)

• Solution Procedure
  – Assume \(R\) Is Known And Fixed
  – Solve For \(m\) And \(b\) In Terms Of \(X_{\text{min}}, X_{\text{max}}, Y_{\text{min}}, Y_{\text{max}}\)
  – Use A Tool Like Excel To Solve For \(m\) And \(b\) For Several Values Of \(R\)
  – Choose Largest Possible Values Of \(m\) And \(b\)
Calculating The Coefficients

• The Constraints

\[ X_{\text{min}} \Rightarrow Y_{\text{min}} = 0 \]
\[ X_{\text{max}} \Rightarrow Y_{\text{max}} = 2^n - 1 \]

• Substituting Into The Direct Mode Equation

\[ Y_{\text{min}} = (mX_{\text{min}} + b) \cdot 10^R \]
\[ Y_{\text{max}} = (mX_{\text{max}} + b) \cdot 10^R \]

\[ m = \left( \frac{Y_{\text{max}} - Y_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \right) \cdot 10^{-R} \]
\[ b = \left( Y_{\text{min}} - \frac{Y_{\text{max}} - Y_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} X_{\text{min}} \right) \cdot 10^{-R} \]

• Solving For \( m \) And \( b \)
Calculating The Coefficients

Example

• AC-DC Rectifier For Telecom Applications
  – Wide Range Of Output Voltage To Control Battery Charging
  – Resolution In Range Of 10–20 mV

• Number Of Bits For Input: 10
  – $Y_{min} = 000h$
  – $Y_{max} = 1023d = 3FFh = 1111111111b$

• Output Voltage Range
  – $X_{min} = 44 \text{ Vdc}$
  – $X_{max} = 58 \text{ Vdc}$
  – Resolution: 13.69 mV/bit
Calculating The Coefficients

Example

• Using Microsoft Excel to Solve For \( m \) And \( B \) For Various Values Of \( R \) Yields:

<table>
<thead>
<tr>
<th>R</th>
<th>( m ) (calculated)</th>
<th>( b ) (calculated)</th>
<th>( m ) (rounded)</th>
<th>( b ) (rounded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>730714.2857</td>
<td>-32151428.57</td>
<td>730714</td>
<td>-32151429</td>
</tr>
<tr>
<td>-3</td>
<td>73071.42857</td>
<td>-3215142.857</td>
<td>73071</td>
<td>-3215143</td>
</tr>
<tr>
<td>-2</td>
<td>7307.142857</td>
<td>-321514.2857</td>
<td>7307</td>
<td>-321514</td>
</tr>
<tr>
<td>-1</td>
<td>730.7142857</td>
<td>-32151.42857</td>
<td>731</td>
<td>-32151</td>
</tr>
<tr>
<td>0</td>
<td>73.07142857</td>
<td>-3215.142857</td>
<td>73</td>
<td>-3215</td>
</tr>
<tr>
<td>1</td>
<td>7.307142857</td>
<td>-321.5142857</td>
<td>7</td>
<td>-322</td>
</tr>
<tr>
<td>2</td>
<td>0.730714286</td>
<td>-32.15142857</td>
<td>1</td>
<td>-32</td>
</tr>
<tr>
<td>3</td>
<td>0.073071429</td>
<td>-3.215142857</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>4</td>
<td>0.007307143</td>
<td>-0.321514286</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.000730714</td>
<td>-0.032151429</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Calculating The Coefficients
Example

- Using Microsoft Excel to Solve For \( m \) And \( B \) For Various Values Of \( R \) Yields:

<table>
<thead>
<tr>
<th>( R )</th>
<th>( m ) (calculated)</th>
<th>( b ) (calculated)</th>
<th>( m ) (rounded)</th>
<th>( b ) (rounded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>730714.2857</td>
<td>-32151428.57</td>
<td>730714</td>
<td>-32151429</td>
</tr>
<tr>
<td>-3</td>
<td>73071.42857</td>
<td>-3215142.857</td>
<td>73071</td>
<td>-3215143</td>
</tr>
<tr>
<td>-2</td>
<td>7307.142857</td>
<td>-321514.2857</td>
<td>7307</td>
<td>-321514</td>
</tr>
<tr>
<td>-1</td>
<td>730.7142857</td>
<td>-32151.42857</td>
<td>731</td>
<td>-32151</td>
</tr>
<tr>
<td>0</td>
<td>73.07142857</td>
<td>-3215.142857</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7.307142857</td>
<td>-321.5142857</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.730714286</td>
<td>-32.1514286</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.073071429</td>
<td>-3.2151429</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.007307143</td>
<td>-0.32151</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.000730714</td>
<td>-0.03215</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Values In Red Exceed The Range Of Values Available To A 16 Bit Signed Integer (+32,767 To –32,768)
Calculating The Coefficients

Example

- Using Microsoft Excel to Solve For \( m \) And \( b \) For Various Values Of \( R \) Yields:

<table>
<thead>
<tr>
<th>( R )</th>
<th>( m ) (calculated)</th>
<th>( b ) (calculated)</th>
<th>( m ) (rounded)</th>
<th>( b ) (rounded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>-3215142857</td>
<td>-3215142857</td>
<td>730714</td>
<td>-32151429</td>
</tr>
<tr>
<td>-3</td>
<td>7307142857</td>
<td>-3215142857</td>
<td>730711</td>
<td>-3215143</td>
</tr>
<tr>
<td>-2</td>
<td>7307142857</td>
<td>-3215142857</td>
<td>73071</td>
<td>-321514</td>
</tr>
<tr>
<td>-1</td>
<td>7307142857</td>
<td>-3215142857</td>
<td>731</td>
<td>-32151</td>
</tr>
<tr>
<td>0</td>
<td>73.07142857</td>
<td>-3215142857</td>
<td>73</td>
<td>-3215</td>
</tr>
<tr>
<td>1</td>
<td>7.307142857</td>
<td>-3215142857</td>
<td>7</td>
<td>-322</td>
</tr>
<tr>
<td>2</td>
<td>0.730714286</td>
<td>-32.15142857</td>
<td>1</td>
<td>-32</td>
</tr>
<tr>
<td>3</td>
<td>0.073071429</td>
<td>-3.215142857</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>4</td>
<td>0.007307143</td>
<td>-0.321514286</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.000730714</td>
<td>-0.032151429</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

For Best Resolution, Choose Largest Possible Values Of \( m \) And \( b \):

- \( m = 731 \)
- \( b = -32151 \)
- \( R = -1 \)
Calculating The Coefficients

Example

• Chosen Solution

\[ m: \ 731 \]
\[ b: \ -32151 \]
\[ R: \ -1 \]

• Double Check Calculation

\[
Y_{\text{min}} = (mX_{\text{min}} + b) \cdot 10^R
\]
\[
= (731 \cdot 44 - 32,151) \cdot 10^{-1}
\]
\[
= 1.3 \neq 0
\]

\[
Y_{\text{max}} = (mX_{\text{max}} + b) \cdot 10^R
\]
\[
= (731 \cdot 58 - 32,151) \cdot 10^{-1}
\]
\[
= 1024.7 \neq 1023
\]

More Rounding And Quantization Errors!
Calculating The Coefficients

Example

- Minimum Voltage \( (X_{\text{min}}) \): 44 V
- Maximum Voltage \( (X_{\text{max}}) \): 58 V
- PMBus Device Resolution: 16 Bits

<table>
<thead>
<tr>
<th>R</th>
<th>( m ) (calculated)</th>
<th>( b ) (calculated)</th>
<th>( m ) (rounded)</th>
<th>( b ) (rounded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>468107.1429</td>
<td>-20596714.29</td>
<td>468107</td>
<td>-20596714</td>
</tr>
<tr>
<td>-1</td>
<td>46810.71429</td>
<td>-20596714.29</td>
<td>46811</td>
<td>-2059671</td>
</tr>
<tr>
<td>0</td>
<td>4681.071429</td>
<td>-205967.1429</td>
<td>4681</td>
<td>-205967</td>
</tr>
<tr>
<td>1</td>
<td>468.1071429</td>
<td>-20596.71429</td>
<td>468</td>
<td>-20597</td>
</tr>
<tr>
<td>2</td>
<td>46.81071429</td>
<td>-2059.671429</td>
<td>47</td>
<td>-2060</td>
</tr>
<tr>
<td>3</td>
<td>0.4681071429</td>
<td>-20.59671429</td>
<td>5</td>
<td>-206</td>
</tr>
<tr>
<td>4</td>
<td>0.468107143</td>
<td>-20.59671429</td>
<td>0</td>
<td>-21</td>
</tr>
</tbody>
</table>
Calculating The Coefficients

Example

• Chosen Solution

\[ m : 468 \]
\[ b : -20597 \]
\[ R : 1 \]

• Double Check Calculation

\[ Y_{\text{min}} = (mX_{\text{min}} + b) \cdot 10^R \]
\[ = (468 \cdot 44 - 20,597) \cdot 10^{+1} \]
\[ = -5 \neq 0 \]

\[ Y_{\text{max}} = (mX_{\text{max}} + b) \cdot 10^R \]
\[ = (468 \cdot 58 - 20,597) \cdot 10^{+1} \]
\[ = 65,470 \neq 65,535 \]

Still Have Rounding And Quantization Errors!
What To Do?

• Choices
  – Live With It
  – Adjust The Slope \( (m) \)
  – Adjust the Offset \( (b) \)
  – Adjust Both
  – Adjust \( X_{max} \) And \( X_{min} \)

• Optimization Is Left As An Exercise For The Student
What To Do?

• Choices
  – Live With It
  – Adjust The Slope (m)
  – Adjust Both
  – Adjust $X_{\text{max}}$ And $X_{\text{min}}$

• Optimization Is Left As An Exercise For The Student

Lesson:
You Must Pay Attention To Errors Introduced By Discrete Arithmetic!
Decoding Direct Mode Example

• Example Of Reading The Output Current Of An Isolated DC-DC Bus Converter

• Using COEFFICIENTS Command Returns Values For $m$, $b$ And $R$ As:
  – $m = 850$
  – $b = 0$
  – $R = -2$

• Using READ_IOOUT Command Returns The Value $000000001101001b \Rightarrow 105d$
Decoding Direct Mode
Example

• Use The Inverse Of The Equation Used To Encode

\[ Y = (mX + b) \cdot 10^R \]
\[ X = \frac{1}{m} (Y \cdot 10^{-R} - b) \]
\[ X = \frac{1}{850} \left( 105 \cdot 10^{-(-2)} - 0 \right) \]
\[ = \frac{10500}{850} = 12.35 \]

• Substitute Values And Solve

• Output Current = 12.35 A

Note That These Calculations Are Done In The Host And Not The PMBus Device!
Setting The Output Voltage

Step 1
Which Data Format?
(aka Which Mode)

- Linear
- Direct
- VID

VOUT_MODE Command

Step 2
Set The Output Voltage Using The
VOUT_COMMAND Command
**VOUT_MODE Command**

- **VOUT_MODE** Command Is Sent Separately From Any Other Command, Such As VOUT_COMMAND
- Sent Only When Necessary To Change The Mode
  - Only Once?
- Applies For All Output Voltage Related Commands

![VOUT_MODE Diagram](image)
VOUT_MODE & VOUT_COMMAND

VOUT_MODE Data Byte For VID Mode

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Mode = 001b VID Code Type

VOUT_COMMAND Data Bytes For VID Mode

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

VID (Right Justified)
VOUT_MODE & VOUT_COMMAND

VOUT_MODE Data Byte For Linear Mode

7 6 5 4 3 2 1 0

Mode = 000b Exponent

VOUT_COMMAND Data Bytes For Linear Mode

Data Byte High Data Byte Low

7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0

Y Mantissa

Voltage = Y \cdot 2^N
VOUT_MODE & VOUT_COMMAND

VOUT_MODE
Data Byte For DIRECT Mode

Mode = 010b
Value = 00000b

VOUT_COMMAND Data Bytes For DIRECT Mode

Data Byte High Data Byte Low

7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
Fine Tuning The Output Voltage

VOUT_MARGIN_HIGH

VOUT_COMMAND

VOUT_MARGIN_LOW

3:1 Mux

OPERATION Command

VOUT_MAX

VOUT_SCALE_LOOP

“Reference Voltage Equivalent”

VOUT_TRIM

VOUT_CAL

VOUT_DROOP

+ + -

Limiter

IOUT

X
Margin Testing

Margin Testing Uses Pre-Store Values
Margin Testing

OPERATION Command
Selects Between Stored Setpoint And Upper And Lower Margin Voltages

Margin Testing Uses Pre-Stored Values
Fine Tuning The Output Voltage

VOUT_TRIM
Intended Mostly For End User.
Example: Adjust Voltage On ASIC Terminals In Manufacturing
Fine Tuning The Output Voltage

VOUT_CAL
Intended Mostly For Device Manufacturer
Example: Calibrate Out Reference Voltage Error
Fine Tuning The Output Voltage

VOUT_DROOP
Use For Either
Adaptive Voltage Positioning Or
Passive Current Sharing
Fine Tuning The Output Voltage

VOUT_MAX Helps Prevent “Oops!”
Protect Devices By Limiting The Maximum Voltage Than Can Be Generated
Using And External Divider

VOUT

Error Amp

V_OUT

RESISTOR DIVIDER RATIO

K_R

VOUT_COMMAND

16

K

VOUT_SCALE_LOOP

Error Processing/Control Loop

PMBus Device
Using And External Divider

Simplifies Life For The End User

They Do Not Need To Think About The Voltage Divider

Just Send Command Voltage As They Want It
Example: 1.80 V

PMBus Device

VRE

VOUT

KR

VOUT

SCALE

LOOP

VOUT_COMMAND

16

K

Error Processing/Control Loop

SMI

System Management Interface Forum

124
On/Off Control

- Two Inputs Control Whether A PMBus Device Is Operating Or Not
  - Hardwired CONTROL Pin (Programmable Polarity)
  - OPERATION Command From The Bus
- On/Off Control Totally Programmable
- CONTROL Pin Options
  - Active High Or Active Low
  - Followed Programmed Sequencing Or Shutdown Immediately
## ON_OFF_CONFIG

<table>
<thead>
<tr>
<th>On/Off Control Mode</th>
<th>Device Power</th>
<th>CONTROL Input</th>
<th>Bus Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always ON</td>
<td>If Power, Then ON</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Respond To CONTROL Only</td>
<td>If Power, Respond To CONTROL</td>
<td>Active High</td>
<td>Ignore Bus Commands</td>
</tr>
<tr>
<td>Respond To Bus Only</td>
<td>And Bus Commands As Programmed</td>
<td>Active Low</td>
<td>Ignore CONTROL</td>
</tr>
<tr>
<td>Respond To Both CONTROL</td>
<td></td>
<td>Ignore CONTROL</td>
<td>Respond To Bus Commands</td>
</tr>
<tr>
<td>And Bus</td>
<td></td>
<td>Active High</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Active Low</td>
<td></td>
</tr>
</tbody>
</table>
# OPERATION Command

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>IMMEDIATE OFF (No Sequencing)</td>
<td>N/A</td>
</tr>
<tr>
<td>01</td>
<td>XX</td>
<td>XX</td>
<td>XX</td>
<td>OFF (With Sequencing)</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>00</td>
<td>XX</td>
<td>XX</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>10</td>
<td>01</td>
<td>01</td>
<td>XX</td>
<td>ON</td>
<td>MARGIN LOW (Ignore Fault)</td>
</tr>
<tr>
<td>10</td>
<td>01</td>
<td>10</td>
<td>XX</td>
<td>ON</td>
<td>MARGIN LOW (Act On Fault)</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>01</td>
<td>XX</td>
<td>ON</td>
<td>MARGIN HIGH (Ignore Fault)</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>XX</td>
<td>ON</td>
<td>MARGIN HIGH (Act On Fault)</td>
</tr>
</tbody>
</table>
OPERATION Command

**Ignore Fault**

Prevents sending an alarm or responding to an output undervoltage condition that was deliberately caused by margin testing.

This allows system testing to proceed without special modifications to the power supply/DC-DC converter.
**OPERATION Command**

### Act On Fault

The PMBus Device Will Send An Alarm Or Respond To An Output Undervoltage Condition That Was Deliberately Caused By Margin Testing

This May Be Desired To Protect The System From Extreme Output Voltages

<table>
<thead>
<tr>
<th>Bit</th>
<th>Unit On/Off</th>
<th>Margin State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF (testing)</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>OFF (testing)</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>3</td>
<td>MARGIN LOW (Ignore Fault)</td>
<td>MARGIN LOW (Act On Fault)</td>
</tr>
<tr>
<td>4</td>
<td>MARGIN LOW (Ignore Fault)</td>
<td>MARGIN HIGH (Act On Fault)</td>
</tr>
<tr>
<td>5</td>
<td>MARGIN HIGH (Act On Fault)</td>
<td>MARGIN LOW (Ignore Fault)</td>
</tr>
<tr>
<td>6</td>
<td>MARGIN HIGH (Act On Fault)</td>
<td>MARGIN HIGH (Act On Fault)</td>
</tr>
</tbody>
</table>
Sequencing: Event Driven

- Event Driven Sequencing Is Closed Loop
- Requires Power System Manager To Close The Loop
Sequencing: Time Driven Commands

- Open Loop: Does Not Require Power System Manager
Open Loop Tracking

- To Implement An Open Loop Tracking Turn On, Need To Know:
  - Each Output Voltage
  - Desired Rise Time (TON_RISE) For Just One Output Voltage

- Calculate TON_RISE Of All Other Outputs As Follows:

\[
TON\_RISE(V2) = TON\_RISE(V1) \cdot \frac{V2}{V1}
\]

\[
TON\_RISE(V3) = TON\_RISE(V1) \cdot \frac{V3}{V1}
\]
Status Reporting And Fault Management

• The PMBus Protocol Supports Two Alarm Levels
  – Warnings (Minor Alarms)
  – Faults (Major Alarms)

• Warnings Only Result In Host Being Notified That Attention Is Needed

• Faults Cause The PMBus Device To Respond And Take Action Internally As Programmed

• Parametric Information (e.g. Voltage) Can Also Be Read From PMBus Devices
Notifying The Host Of A Fault

- Host Can Continuously Poll PMBus Devices
- PMBus Device Can Send An Interrupt
  - SMBALERT# Signal Is Optional
  - See The SMBus Specification For Details
- PMBus Device Can Become A Bus Master And Transmit Notice To System Host
  - Optional
  - Requires A More Sophisticated Host And More Sophisticated PMBus Devices
Status Reporting: 3 Levels Of Detail

Level 1:
STATUS_BYTE
Most Critical Info

Level 2:
STATUS_WORD
Adds More Important Info

Level 3:
Status Registers
Detailed Information
## STATUS_BYTE & STATUS_WORD

<table>
<thead>
<tr>
<th>STATUS_WORD</th>
<th>High Byte</th>
<th>Low Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>UNKNOWN_FAULT OR WARNING</td>
<td>MFR_SPECIFIC²</td>
<td>UNIT IS BUSY</td>
</tr>
<tr>
<td>Reserved</td>
<td>VIN_UV_FAULT</td>
<td>UNIT IS OFF</td>
</tr>
<tr>
<td>POWER_GOOD Negated</td>
<td>IOUT_OC_FAULT</td>
<td>VOUT_OV_FAULT</td>
</tr>
<tr>
<td>INPUT_FAULT OR WARNING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IOOUT_FAULT OR WARNING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOUT_FAULTS OR WARNINGS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### STATUS_BYTE

- UNIT IS BUSY
- UNIT IS OFF
- VOUT_OV_FAULT
- IOUT_OC_FAULT
- VIN_UV_FAULT
- TEMPERATURE_FAULT_OR_WARNING
- COMM, LOGIC, MEMORY_EVENT
- OTHER_FAULT_OR_WARNING

### Bit Positions

- 7: UNIT IS BUSY
- 6: UNIT IS OFF
- 5: VOUT_OV_FAULT
- 4: IOUT_OC_FAULT
- 3: VIN_UV_FAULT
- 2: TEMPERATURE_FAULT_OR_WARNING
- 1: COMM, LOGIC, MEMORY_EVENT
- 0: OTHER_FAULT_OR_WARNING
Status Registers

### STATUS_WORD

<table>
<thead>
<tr>
<th>High Byte</th>
<th>Low Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>

#### STATUS_BYTE

- **UNIT IS BUSY**
- **UNIT IS OFF**
- **VOUT_OV FAULT**
- **IOUT_OC FAULT**
- **VIN_UV FAULT**
- **TEMPERATURE FAULT OR WARNING**
- **COMM, LOGIC, MEMORY EVENT**
- **OTHER FAULT OR WARNING**

#### Registers

- **STATUS_VOUT** Register
- **STATUS_IOUT** Register
- **STATUS_INPUT** Register
- **STATUS_MFR** Register
- **STATUS_OTHER** Register
- **STATUS_CML** Register
- **STATUS_TEMPERATURE** Register

1: CML: Communication, Memory, Logic
2: MFR SPECIFIC: Manufacturer Specific
Clearing Status Bits

- 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,
  - 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.
Fault Management: Input

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>OV FAULT</td>
<td>OC FAULT</td>
</tr>
<tr>
<td>OV WARN</td>
<td>OC WARN</td>
</tr>
<tr>
<td>UV WARN</td>
<td></td>
</tr>
<tr>
<td>UV FAULT</td>
<td></td>
</tr>
</tbody>
</table>

Related Commands: VIN_ON, VIN_OFF
Fault Management: Output

Converter

Voltage

- OV FAULT
- OV WARN
- UV WARN
- UV FAULT

Current

- OC FAULT
- OC WARN
- UC FAULT

Related Commands:
POWER_GOOD_ON, POWER_GOOD_OFF
Other Fault Management

Temperature

| OT FAULT | OT WARN | UT WARN | UT FAULT |

Other Faults

| Fan Fault 1 |
| Fan Fault 2 |
| Current Share |
| Power Limiting |
| Communication |
| And more... |
Voltage Or Temperature Fault Response Programming Byte

**RESPONSE**
- 00 - CONTINUE
- 01 - DELAYED OFF
- 10 - SHUTDOWN & RETRY
- 11 - INHIBIT

**DELAY TIME**
- XXX - NUMBER OF DELAY TIME UNITS

**RETRY**
- 000 - LATCH OFF
- 001 - 110: RETRY COUNT
- 111 - CONTINUOUS
Fault Response Examples

- 1 0 0 0 0 0 0 0

  DON’T CARE
  000 = LATCH OFF
  10 = SHUTDOWN AND RETRY

Shut Down And Latch Off
Fault Response Examples

- 10 = SHUTDOWN AND RETRY
- 111 = CONTINUOUS RETRY
- DON’T CARE

Continuous Hiccup Mode
Fault Response Examples

Keep Operating For 3 Time Units. If Fault Still Exists At That Time, Shut Down And Latch Off.
Fault Response Examples

“Time Units” Are Defined In Each Device’s Product Literature

Keep Operating For 3 Time Units. If Fault Still Exists At That Time, Shut Down And Latch Off
Current Fault Options

**RESPONSE**
- 00 - CONTINUE
- 01 - CONTINUE WITH LOW VOLTAGE SHUTDOWN
- 10 - DELAYED OFF
- 11 - SHUTDOWN & RETRY

**DELAY TIME**
- XXX - NUMBER OF DELAY TIME UNITS

**RETRY**
- 000 - LATCH OFF
- 001 - 110: RETRY COUNT
- 111 - CONTINUOUS
Parametric Information

- Input Voltage (READ_VIN)
- Input Current (READ_IIN)
- Output Voltage (READ_VOUT)
- Output Current (READ_IOUT)
- Hold Up Capacitor Voltage (READ_VCAP)
- Temperature (READ_TEMPERATURE_1, _2, _3)
  - Up To 3 Sensors
- Fan Speed (READ_VFAN_1, _2)
  - Up To 2 Fans
- Duty Cycle (READ_DUTY_CYCLE)
- Switching Frequency (READ_FREQUENCY)
Group Commands/Operation

- Used when multiple units need to execute a command simultaneously
- One SMBus transaction used to send commands to multiple addresses
  - Sent in one large packet using repeated STARTs
- Can be same or different commands
  - Example: Command one unit to margin low and all others to margin high
- Commands are executed when SMBus STOP condition received
Interleaving

- **INTERLEAVE Command Sets**
  - Group Number
  - Number Of Units In The Group
  - Switching Order Within The Group

Example Of INTERLEAVE Command Operation

\[
T_{\text{delay}}(\text{Unit } X) = \frac{\text{Interleave Order Of Unit } X}{\text{Number In Group}} \cdot T_s
\]
Multiple Output Units And Paging

• Paging Allows One Physical Address To Be Used To Control Multiple Outputs
  – One Address Per Physical Unit
  – One Page Per Output
  – Pages Contain All The Settings Of Each Output

• Paging Process
  – Set Page For Output Of Interest
  – Send Commands
    • Configure, Control, Read Status
Paging: Multiple Output Units

Multiple Output PMBus Device

PMBus Bus Switch

PAGE 0 Memory

Output 0 Power/Control

PAGE 1 Memory

Output 1 Power/Control

PAGE 2 Memory

Output 2 Power/Control
Paging:
Non-PMBus Device Adapter

Address Pins

PMBus To Non-PMBus Device Adapter

PMBus Bus Switch

PAGE 0 Memory

Device 0 Interface

Non-PMBus Device

PAGE 1 Memory

Device 1 Interface

Non-PMBus Device

PAGE 2 Memory

Device 2 Interface

Non-PMBus Device
Paging:
Non-PMBus Device Adapter

Example Device: Analog Margin/Sequence Controller With PMBus Interface

Example Device: POL Converter With An Analog Interface

PMBus To Non-PMBus Device Adapter

- PAGE 0 Memory
- Device 0 Interface
- PAGE 1 Memory
- Device 1 Interface
- PAGE 2 Memory
- Device 2 Interface

PMBus Switch

PMBus

Non-PMBus Device

Non-PMBus Device

Non-PMBus Device
Data Integrity And Security

• Protecting Against Corrupted Transmissions
  – Packet Error Checking Can Be Used

• Unwanted Or Unintentional Data Changes
  – Write Protect Pin
  – WRITE_PROTECT Command
Manufacturer And User Data

• Manufacturer’s Information
  – Inventory Information (Model Number, Etc.)
  – Ratings Information (Input Voltage Range, Etc.)

• User Data
  – 32 Command Codes For PMBus Device Makers To Support User Inventory And Configuration Data
  – Example: Digital Control Loop Coefficients

• Manufacturer Specific Commands
  – 45 Command Codes Reserved For PMBus Device Makers To Implement Manufacturer Specific Commands
Many Other Configuration Commands

- Maximum Output Voltage
- Maximum Output Power
- Voltage Scale For External Divider Network
- Maximum Duty Cycle
- Switching Frequency
- Turn On/Off Levels For Input Voltage
- Current Scale For Current Sense Resistance
- Current Measurement Calibration
For More Information

www.PMBus.org

info@PMBus.org
Thank You For Your Time And Attention!