excelsys

CoolX

CoolX600 PMBus™ Interface Manual

Revolutionary Fanless 600W Modular Power

Contents

List of Figures

List of Tables

List of Equations

1 Introduction

The Power Management Bus (PMBus™) is an open standard which defines a means of communication for power conversion devices. It defines a full set of commands and data structures required by power control and management components. The CoolX Series® PMBus interface facilitates the communication of operating parameters such as output voltage, output current and internal temperature with other PMBus™enabled devices. It also facilitates the remote adjustment of parameters such as output voltage level, current limit and enable status. For more information about PMBus™, please see the System Management Interface Forum website www.PowerSIG.org

1.1 Revision History

Revision	Date	Details
00	23.08.16	First Release
01	30.11.16	Extra details added
02	09.05.17	Corrections
03	12.04.19	Made specific to CX600

2 Connectors

The connectors which are relevant for PMBus operation are located on the Communications Board. The system signal connector J1005 contains the PMBus transmission pins and J1011 is used to set the PMBus address of the system.

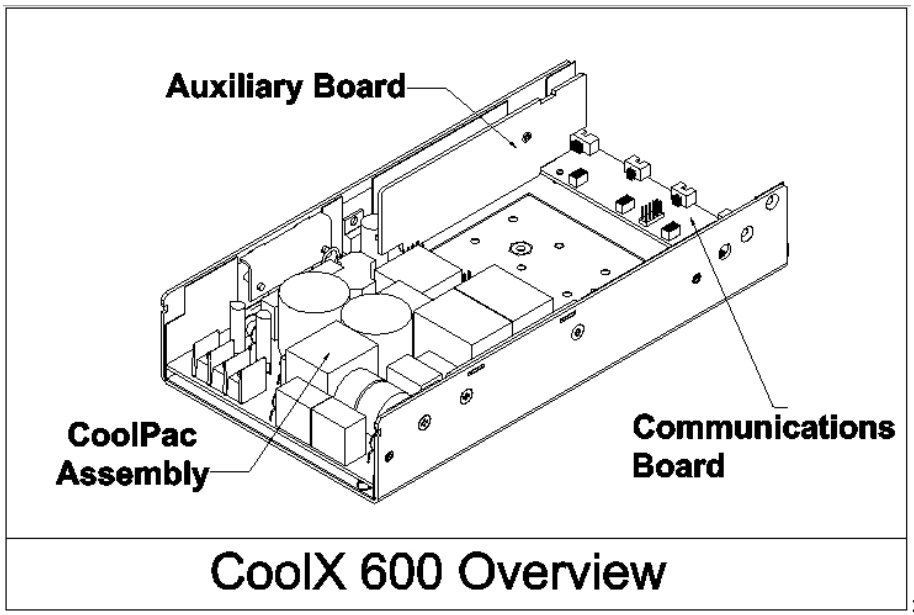


Figure 1: CoolX System Overview

2.1 J1011 - PMBus Addressing

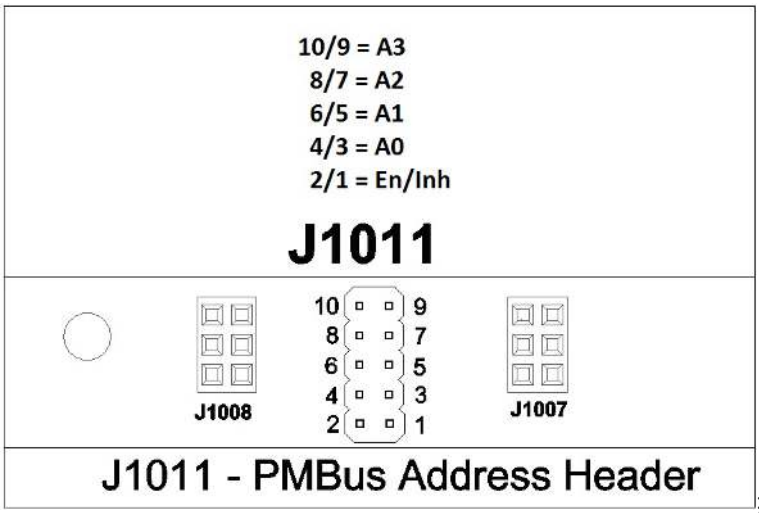
The PMBus standard utilizes 7 bits for addressing. These 7 bits form what is referred to as a slave address. This is the hardware address of the PMBus device and and is independent of the direction of data transfer. During any read or write operation, a direction bit is appended by the master to the 7 bit slave address to indicate the direction of data transfer: 1 = READ operation; 0 = WRITE operation. Sometimes PMBus/I2C vendors will include the direction bit in the address leading to a separate READ ADDRESS and WRITE ADDRESS. In this convention, the WRITE ADDRESS can be calculated as (SLAVE ADDRESS * 2) and the READ ADDRESS can be calculated as ((SLAVE ADDRESS * 2)+1).

The CoolX Series PMBus interface allows the user to modify the lower 4 bits of the slave address, leading to a fixed part of the address and a variable part of the address. The fixed part of the slave address consists of the 3 most significant bits A6, A5, and A4 and *always* equals 101. The variable part of the address consists of the 4 least significant bits A3, A2, A1 and A0 and these bits can be modified by the placement of jumpers on the corresponding 4 pin headers on connector J1011.

A6	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**
1	0	1	A3*	A2*	A1*	A0*

Table 1: Slave Address Structure

* *Determined by the position of the jumper link. Jumper in = logic 1; Jumper out = logic 0.*



;

Figure 2: Connector J1011 Pinout

The address lines for A3, A2, A1 and A0 default to logic 0 (jumper out) (the default slave address ⇒ = 1010000 = 0x50). The placement of a jumper on a header pulls the corresponding address line to a logic 1. The full list of available addresses is therefore as follows:

A6	A5	A4	A3	A2	A1	A0	SLAVE Address	WRITE Address	READ Address
1	0	1	0 / OUT	0 / OUT	0 / OUT	0 / OUT	0x50	0xA0	0xA1
1	0	1	0 / OUT	0 / OUT	0 / OUT	1 / IN	0x51	0xA2	0xA3
1	0	1	0 / OUT	0 / OUT	1 / IN	0 / OUT	0x52	0xA4	0xA5
1	0	1	0 / OUT	0 / OUT	1 / IN	1 / IN	0x53	0xA6	0xA7
1	0	1	0 / OUT	1 / IN	0 / OUT	0 / OUT	0x54	0xA8	0xA9
1	0	1	0 / OUT	1 / IN	0 / OUT	1 / IN	0x55	0xAA	0xAB
1	0	1	0 / OUT	1 / IN	1 / IN	0 / OUT	0x56	0xAC	0xAD
1	0	1	0 / OUT	1 / IN	1 / IN	1 / IN	0x57	0xAE	0xAF
1	0	1	1 / IN	0 / OUT	0 / OUT	0 / OUT	0x58	0xB0	0xB1
1	0	1	1 / IN	0 / OUT	0 / OUT	1 / IN	0x59	0xB2	0xB3
1	0	1	1 / IN	0 / OUT	1 / IN	0 / OUT	0x5A	0xB4	0xB5
1	0	1	1 / IN	0 / OUT	1 / IN	1 / IN	0x5B	0xB6	0xB7
1	0	1	1 / IN	1 / IN	0 / OUT	0 / OUT	0x5C	0xB8	0xB9
1	0	1	1 / IN	1 / IN	0 / OUT	1 / IN	0x5D	0xBA	0xBB
1	0	1	1 / IN	1 / IN	1 / IN	0 / OUT	0x5E	0xBC	0xBD
1	0	1	1 / IN	1 / IN	1 / IN	1 / IN	Reserved	Reserved	Reserved

Table 2: Available Address Space

2.1.1 Parallel Operation

If multiple CoolPacs are to be used in parallel on the same bus then each Communications Board will need to be assigned a unique address through the fitting of one or more jumpers according to Table 2 below. In total, 15 unique addresses are available which limits the maximum possible amount of devices on a single bus to 15 (the highest address is reserved). The buses of all devices must then be paralleled. To do this, simply connect all required PMBus signals (i.e. SDA, SCL, CONTROL (if required) & COMMON) in parallel.

2.2 J1005 - System Connector

This 8 pin connector is used to connect the Communications Board to the bus. The pinout of this connector is as follows:

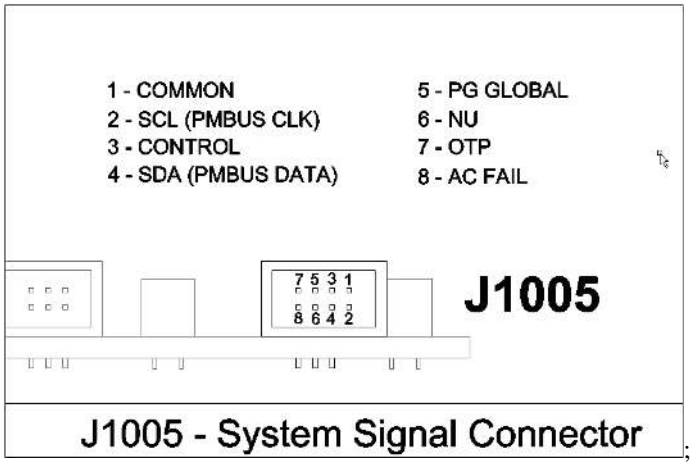


Figure 3: Connector J1005 Pinout

J1005 System Connector	Mating Connector	Crimp terminal
8-way MOLEX 87833-0831	Locking MOLEX 51110-0860	MOLEX 50394
	Non-Locking MOLEX 51110-0850	MOLEX 51110-0856 (includes locking tab & polarization keying)

Table 3: J1005 Connector Details

2.2.1 SDA - Serial Data Line

This is the data line over which all serial communication takes place. It is essential that this pin is connected to the PMBus SDA line. This is an open collector pin which should be pulled up to 5V by the PMBus host device.

2.2.2 SCL - Serial Clock Line

This is the clock line which synchronizes all serial communication over the PMBus. It is essential that this pin is connected to the PMBus SCL line. The CoolX Series PMBus interface is designed to operate with a PMBus clock frequency of 100KHz. This is an open collector pin which should be pulled up to 5V by the PMBus host device.

2.2.3 CONTROL (Global Enable)

This input can be used to enable or disable all CoolMods simultaneously. This pin is pulled up to 5V internally. The default operation (jumper J1011 pins 2/1 not fitted) is that logic high means all modules are enabled. Pulling the pin to

0V will disable all modules. If the jumper is fitted then the logic is reversed i.e. the pin must be pulled to 0V to enable the modules (in this configuration the individual enable input for each module must also be pulled to 0V to enable that module).

2.2.4 COMMON - Signal GND

This should be connected to GND or Signal Return of the PMBus Host device. Note that this is at the same potential as the Auxiliary output ground terminal.

3 PMBus Communication

3.1 PMBus Linear Format

The CoolX Series PMBus interface utilizes the linear data format defined in the PMBus Specification to represent voltage, current and temperature readings. This format presents real world units (Amps, Volts, Degrees) to the host system in a manner which is less computationally difficult for the host system than the alternative direct system.

The data returned consists of the following:

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

These combine to form a two byte word as follows:

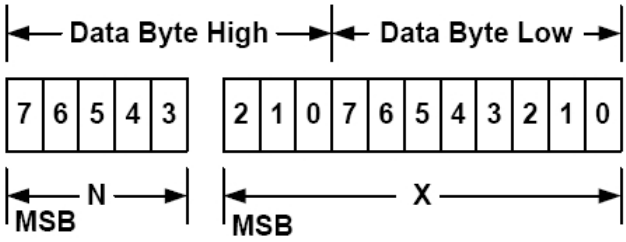


Figure 4: Linear Format Data Structure

3.1.1 Linear Format Decoding

To understand the decoding of the linear format data to obtain the real-world measurement, we will work through an example of an output current measurement:

Sample Data:

- Returned Byte 1 = 0xDB
- Returned Byte 2 = 0x12

The first step is to extract the exponent data:

- 0xDB12 in binary format= **11011**01100010010.
- Exponent bits = 11011
- 11011 converted from two's complement = -5.
- The exponent in this example is therefore -5.

The second step is to extract the mantissa data:

- 0xDB12 converted to binary = 11011**01100010010**
- Mantissa bits = 01100010010
- 01100010010 converted from two's complement = 786.
- The Mantissa in this example is therefore 786.

The final step is simply to calculate the real world value using the two figures obtained above and the equation:

$$Y = X * (2^N) \tag{1}$$

Where:
Y = The real-world value to be calculated e.g. output current (in Amps) in this example.
X = The Mantissa obtained above e.g. 768 in this example.
N = The Exponent obtained above e.g. -5 in this example.

$$I_{out} = 768 * (2^{-5}) = 24.56\,Amps$$

The exact same process is used to calculate temperature readings.

3.2 PMBus Extended Linear Format

The extended linear format works in the same way as the linear format detailed above however, all 16 bits are allocated as mantissa bits which allows for extended precision which is useful when measuring output voltage. For this reason, commands which relate to output voltage will use the extended format. Since the exponent is not available within the returned data, the exponent must be queried (once only) by the system by issuing the VOUT_MODE command which will return the exponent. Please see section 4.1.1 on how to use the VOUT_MODE command to obtain the exponent.

3.3 Packet Error Checking (PEC)

SMBus version 1.1 introduced a Packet Error Checking mechanism to improve communication reliability and robustness. PEC in CoolX is supported as per the SMBus specification but is optional. In order to use PEC, simply request one extra byte for read transactions or write one extra byte for write transactions. The extra byte will be the checksum byte which is a Cyclic Redundancy Check (CRC) checksum. The polynomial used for the CRC calculation in SMBus/PMBus is $X^8 + X^2 + X + 1$ (CRC-8). There are many online resources providing souce code and lookup tables for calculation of the CRC-8 checksum.

4 Supported Commands

The full list of commands currently supported by the CoolX Series PMBus interface is as follows:

Monitoring Commands	Control Commands	Identification Commands
VOUT_MODE	PAGE	MFR_ID
READ_VOUT	OPERATION	MODULE_ID*
READ_IOUT	VOUT_COMMAND	MFR_MODEL
READ_TEMPERATURE_1	ILIMIT_TRIM*	
STATUS_WORD		

Table 4: Command Table

* These commands are manufacturer specific

Commands can return either: a BYTE, a WORD or a BLOCK (multiple bytes including a byte count) as indicated below.

4.1 Monitoring Commands

4.1.1 VOUT_MODE (0x20)]

Protocol: Read Byte
Data Format: Unsigned binary integer
Default Value: N/A

To interpret the extended linear mode readings returned by a module in response to output voltage related commands, the system needs to know the 5 bit exponent which is being used to generate the linear mode data. The VOUT_MODE command will return a byte value which contains the exponent for the selected (paged) module. The byte can be interpreted as follows:

Mode	Bits [7:5]	Bits [4:0]
Linear	000	5 bit mantissa which is returned as part of the output voltage data.

Table 5: VOUT_MODE Command Data Structure

Example: If the exponent is -8, VOUT_MODE will return 0b00011000 = 0x18

4.1.2 READ_VOUT (0x8B)

Protocol: Read Word
Data Format: Extended Linear Format
Default Value: N/A

The READ_VOUT command is used to return the output voltage measurement of the selected (paged) module. The data will be formatted in the Extended Linear format detailed in section 3.2 on page 12 using the exponent which can be found using the VOUT_MODE command (typically -8 for the CoolX series).

Example:
Exponent: -8
Data Returned: Byte 0 = 0x80; Byte 1 = 0x18;

0x1880 = 0d6272

$$V_{out} = \frac{6272}{\frac{1}{2^{-8}}} = 24.50V$$

4.1.3 READ_IOUT (0x8C)

Protocol: Read Word
Data Format: Linear Format
Default Value: N/A

The READ_IOUT command is used to return the output current measurement of the selected (paged) module. The data will be formatted in the Linear format detailed in section 3.1 on page 10.

Example:
Data Returned: Byte 0 = 0x62; Byte 1 = 0xD8;
0xD862 = **0b11011**00001100010
Exponent = 11011 = -5
Mantissa = 00001100010 = 98

$$I_{out} = \frac{98}{\frac{1}{2^{-5}}} = 3.06A$$

4.1.4 READ_TEMPERATURE_1 (0x8D)

Protocol: Read Word
Data Format: Linear Format
Default Value: N/A

The READ_TEMP1 command is used to return the temperature measurement of the selected (paged) module in degrees Celsius. The data will be formatted in the Linear format detailed in section 3.1 on page 10.

Example:
Data Returned: Byte 0 = 0x2D; Byte 1 = 0x00

0x002D = **0b00000**00000101101
Exponent = 00000 = 0
Mantissa = 00000101101

The mantissa Most Significant Bit (MSB) is 0 (positive number) therefore we can just use the decimal value directly = 45. (If the MSB is 1 then this represents a negative temperature and we must convert the mantissa from 2's complement representation).

$$Temperature = \frac{45}{\frac{1}{2^0}} = 45^0C.$$

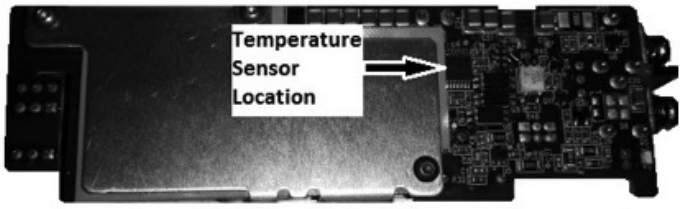


Figure 5: Temperature Sensor Location

4.1.5 STATUS_WORD (0x79)

Protocol: Read Word
Data Format: Unsigned binary integer
Default Value: N/A

To STATUS_WORD command is used to check for the presence of fault conditions such as OTP (Over Temperature Protection) and PG (Power Good) fail. The data returned will be two bytes which contain the flags for both the OTP and PG status. The PG bit of the status word relates to the PG status of the selected (paged) module while the OTP bit represents the entire system (i.e. an OTP event in the CoolPac or in any of the CoolMods will set this bit).

B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	X	**OTP**	X	X

Table 6: Low Byte (Byte 0) Structure

B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	**PG**	X	X	X	X

Table 7: High Byte (Byte 1) Structure

4.2 Control Commands

4.2.1 PAGE (0x00)

Protocol: Read/Write Byte
Data Format: Unsigned binary integer
Default Value: 0x01

As there is only one physical PMBus address within a CoolX system which is shared amongst all of the fitted modules/outputs, the page command is used to select which of the modules subsequent commands are to be applied to. For example, to perform monitoring or control operations on the Module in slot 1, the Page command should be issued with a data byte of 1 prior to the issuing of any further commands. The page shall remain selected until another Page command is received. When read, this command shall return the currently selected page number. The available page numbers are as follows:

Page	Selection
0	CoolPac
1	CoolMod fitted in slot 1
2	CoolMod fitted in slot 2
3	CoolMod fitted in slot 3
4	CoolMod fitted in slot 4

Table 8: Page Numbers

Only commands which can affect individual modules are affected by the page. The list of commands which are affected by the page selection are as follows:

- VOUT_MODE
- VOUT_COMMAND
- READ_VOUT
- READ_IOUT

- READ_TEMP1
- OPERATION
- ILIMIT_TRIM
- MODULE_ID

4.2.2 OPERATION (0x01)

Protocol: Read/Write Byte
Data Format: Unsigned binary integer
Default Value: 0x80

The operation command is used to enable or disable the output of any module. The upper bit (bit 7) is set to 0 to disable the output and is set to 1 to enable the output. The other seven bits are ignored so for example, the following data bytes can be used to enable/disable the output of the module which is currently selected by the page command. Note if the page is 0 then the opeartion will be applied to all modules simultaneously.

Enable output = 0x80
Disable output = 0x00

B7	B6	B5	B4	B3	B2	B1	B0
EN	X	X	X	X	X	X	X

Table 9: Operation Byte Structure

EN: 1 = Output Enabled; 0 = Output Disabled
X: Don't care

4.2.3 VOUT_COMMAND (0x21)

Protocol: Read/Write Word
Data Format: Extended Linear
Default Value: N/A

The VOUT_COMMAND command is used to explicitly set the output voltage of the selected (paged) module to the commanded value. The data should be formatted in the Extended Linear format detailed in section 3.1 on page 10 using the exponent which can be found using the VOUT_MODE command (typically -8 for the CoolX series).

Example:
Exponent = -8; Commanded V_{out} = 36.00V

$$\frac{36}{2^{-8}} = 0d9216 = 0x2400$$

Byte 0 = 0x00; Byte 1 = 0x24

4.2.4 ILIMIT_TRIM [MFR_SPECIFIC_01] (0xD1)

Protocol: Read/Write Word
Data Format: Extended Linear
Default Value: N/A

The ILIMIT_TRIM command is used to explicitly set the current limit of the selected (paged) module to the commanded value. The data should be formatted in the Extended Linear format detailed in section 3.2 on page 12 using the exponent which can be found using the VOUT_MODE command (typically -8 for the CoolX series).

Example:
Exponent = -8
Commanded Ilimit = 3.00A

$$\frac{3.00}{2^{-8}} = 0d768 = 0x0300$$

Byte 0 = 0x00; Byte 1 = 0x03

4.3 Identification Commands

4.3.1 MFR_ID (0x99)

Protocol: Read Block
Data Format: IEC/ISO 8859-1
Default Value: "Excelsys"

The MFR_ID command is used to return a text string which identifies the manufacturer of the system. As per the Read Block protocol, the first byte returned will be an integer representing the amount of characters contained within the string.

4.3.2 MFR_MODEL (0x9A)

Protocol: Read Block
Data Format: IEC/ISO 8859-1
Default Value: N/A

The MFR_MODEL command is used to return a text string which identifies the model number/name of the system. As per the Read Block protocol, the first byte returned will be an integer representing the amount of characters contained within the string.

4.3.3 MODULE_ID [MFR_SPECIFIC_00] (0xD0)

Protocol: Read Byte
Data Format: Unsigned binary integer
Default Value: N/A

The MODULE_ID command is used to return a code representing the model type of the selected (paged) CoolMod. Some of the ID codes in the CoolX family are as follows:

- Cx06 CoolPac = 0x10
- CmA CoolMod = 0x20
- CmB CoolMod = 0x40
- CmC CoolMod = 0x60
- CmD CoolMod = 0x80
- CmE CoolPac = 0x60
- CmF CoolMod = 0x80
- CmM CoolMod = 0x20
- CmN CoolMod = 0x40
- CmP CoolMod = 0x60
- CmQ CoolMod = 0x80

5 Measurement Specifications

5.1 Output Voltage Measurement

Accuracy: +/- 4%
Resolution: (Module Dependent) CmA/CmM=6.6mV; CmB/CmN=16.5mV; CmC/CmP/CmE=44.3mV; CmD/CmQ/CmF=82.4mV

5.2 Output Current Measurement

Accuracy: +/- 4%
Resolution: (Module Dependent) CmA/CmM=40mA; CmB/CmN=29mA; CmC/CmP=16mA; CmD/CmQ=8mA; CmE=44mA; CmF=22mA.

5.3 Temperature Measurement

Accuracy: +/- 10 ^{0}C
Resolution: 1 ^{0}C

5.4 Input Voltage Measurement

Accuracy: +/- 5 Vac typ.*
Resolution: 1 Vac

* This is not a true rms measurement so non-sinusoidal input voltage waveforms will contribute to measurement error.