Valere Power
I²C Design Guidelines for Rectifiers

Version 2.6
December, 2006

Summary

This document is intended to provide a high level overview of the interface between Valere rectifiers and a host system using I2C communications.

The Valere I2C Communications Protocol provides a means for many devices in a Valere Power System to communicate with a host controller over an I2C serial interface. The host controller initiates all commands and requests information from other devices on the serial bus. The protocol manages data transfer between a master (the host controller) and multiple slaves (rectifiers, power entry modules, DC converters, etc.), allows detection of devices by polling, and stable system operation without requiring operator intervention.

The protocol can be viewed as a simple 3-layer hierarchy with each layer providing services to the layer above it. The layers in this protocol are physical, data, and application. The physical layer provides the movement of data bytes between devices. The data layer packages the data bytes into discrete packets with error detection. The application layer allows the controller to gather status and take action on the devices contained within the power system.

This protocol allows the user to gather status and alarming information from rectifier modules, change system voltage and current limit setpoints, enable and disable the primary DC output, and perform tests like lamp test for the front panel LEDs.

Please contact your Valere representative for more detailed interface information. In addition, Valere controllers and system interfaces offer options for RS232, 10/100 LAN, SNMP, and Modem interfaces along with GUI interfaces to simplify your design process.

WARNING: Writing to an unspecified memory address via the I^2C bus may render a module and/or system inoperable.

Physical Layer

Hardware Interconnect

The Valere Power System contains a separate I2C bus on each shelf using a bus hub to isolate the I2C bus on each shelf. The bus operates at 20K baud using a standard 7-bit I2C addressing scheme in a single master, multiple slave configuration. The I2C General Call is supported by this protocol to allow the host controller to broadcast information to all slave devices on the network. An extension of the general call is used to address groups of devices. A discussion of the I2C bus specification is outside the scope of this design guide. Refer to the sources indicated in the reference section of this document for information on the I2C bus specification.

Noise, Isolation, and Grounding

The I2C bus is a 2-wire, open-drain configuration with 10K pull-up resistors from SCL and SDA to VDD on the control boards in each of the devices. In the Valere Power System application, I2C bus hubs are used to limit the maximum number of devices can exist on a physical bus at one time to 8. This minimizes the bus loading and provides a means of fault isolation in the system. Resistors in series with the outputs of the I2C lines on each device protect against high voltage line transients as recommended in Section 16 of the Philips I2C Specification.

It is recommended that the host controller incorporate an I2C isolation circuit to minimize common mode voltage variations. The slave devices have protection against voltage transients but are not designed for significant over voltage transients and may be damaged if a non-isolated interface is used. The I2C bus of the slave devices is referenced to the V- bus of the plant and is non-isolated. All implementations should reference the I2C interface to V-.

Bus Termination

As line speeds and lengths increase, the problem of signal reflections becomes important. Lines should be properly terminated by a pull-up resistor at each device. Termination is discussed in Section 17 of the Philips I2C Specification.

Maximum Loading

The maximum number of devices on an I2C bus is limited by the capacitance in the SCL and SDA lines and the value of the pull-up resistors. The I2C bus is intended to support a maximum of 3mA current sink and 400pF capacitance.

Data Layer

In this system, a single master device on the bus will periodically initiate communications with slave devices. The protocol allows the master device to read or write data to a particular slave device, and write data to all slave devices on the I2C bus using the General Call Address. The General call is extended to provide a way of addressing up to 7 different groups of devices for a group broadcast write command. The length of data read from or written to a slave device is variable from 1 to 16 bytes in the master. Due to RAM limitations in the slave devices, the maximum write data length in the slaves is 12 bytes. The protocol also allows a data address to be specified. Using data-address and data-length parameters, the master node can request any or all of the data available from the slave node.

Using checksums, the master and slave devices in the system verify that the data received was valid. If the data is not valid it is re-transmitted up to 3 times. The protocol also handles errors such as hot-swapping and no response conditions gracefully.

Master Device Message Format

The master device initiates all communication on the I2C bus. This section is a description of the protocol implemented by the master. The master may initiate one of two message types: a data write message, or a data request message.

Data Write Message Format

The format for a data write message is shown in Figure 1 (individual call) & Figure 2 (general call). The data write message begins with the master initiating a START condition. When the START condition completes, the master device sends the I2C address of the slave node with the R/W bit cleared to indicate data will be written to the slave device.

The next byte sent (DATA_LEN) provides the byte count and group information. The DATA_LEN byte serves three purposes. First, bit3-bit0 indicate the number of data bytes to be written to the slave device. Second, bit6-bit4 contain the group address for general call messages. Third, bit7 indicates whether data will be written to, or read from the slave. In this case, the MSB is cleared to indicate that a write is will be performed. The MSB of the DATA_LEN byte performs a similar function to the R/W bit in the I2C address byte.

The next byte sent by the master indicates the starting address in the slave node data buffer that will be written to or read from. This is the DATA_OFF byte. Each slave device maintains a range of data memory for protocol variables.

In a data write message, the number of data bytes specified by bit3-bit0 of DATA_LEN will follow the DATA_OFF byte. When the last byte of data has been sent, the master sends an 8-bit, 2's complement checksum of all data previously sent, including the I2C slave node address byte. If the command is a general call write command, a STOP condition is sent next to end the write command. If the command is a write command to a specific address, instead of a STOP condition, the master then sends a RESTART condition. When the RESTART condition completes, the master device sends the I2C address of the slave node with the R/W bit set to indicate data will be read from the slave device. The master then reads back the slave COMM_STAT byte to determine if the data was transmitted successfully. Finally, the master device terminates the data write message by initiating a STOP condition.

Figure 1 & Figure 2 indicate the data write message format for non-global and global operations.

S	ADDR 0	0 000 LEN	OFFSET	DATA	DATA	CHECKSUM	RS	ADDR	1	COMM STAT	P
start	I2C slave address	data length byte	addr offset byte	Number of data bytes specified by LEN		8-bit check sum	restart	I2C slave address		slave comm status byte	stop

Figure 1 - Data Write Message Format (for non-global)

S	0	0	0	Group	LEN	OFFSET	DATA	DATA	CHECKSUM	P
start	I2C slave address		data length byte			addr offset byte	Number of data bytes specified by LEN		8-bit check sum	restart

Figure 2 - Data Write Message Format (for global)

Data Request Message Format

The format for a data request message is shown in Figure 3. Following the START condition, the master device sends the address of the slave node with the R/W bit cleared to indicate a data write to the I2C slave device. Next, the LEN byte is sent. The 4 LSB's of this byte indicate the number of data bytes to be read from the slave. Because a data read from the slave should be performed, the MSB is set and bit6-bit4 are cleared. The OFFSET byte follows the LEN byte and indicates the starting address in the slave data memory from which data will be read. Next, the master device sends an 8- bit 2's complement checksum of the slave address, data length byte, and data offset byte that were sent in the data request message.

S	ADDR	0	1000	LEN	OFFSET	CHECKSUM	RS	ADDR	1	COMM STAT	DATA	DATA	CHECKSUM	NACK
start	I2C slave address		data length byte		address offset byte	8-bit check sum	restart	I2C slave address		slave comm status byte	Number of data bytes specified by LEN		16-bit check sum	master NACK

Figure 3 - Data Request Message Format

Slave Node Message Processing

In general, the master device may read data from the slave after a data write or data request message by initiating a RESTART condition on the I2C bus and sending the slave address with the R/W bit set. The type of message that was previously sent by the master and its validity determines what data will be returned by the slave.

Each slave node maintains several status bits to indicate the validity of messages sent by the master device. These status bits are stored in the communications status byte (COMM_STAT). Table 1 shows the significance of each bit.

The COMM_STAT byte is always the first data byte to be returned in any data transfer from the slave node to the master node. This allows the master to verify that the previously sent message was processed correctly by the slave node. If, for example, the master sent a data write message, the value of COMM_STAT would be 00h if the slave successfully received the data. If the master previously sent a data request message, the value of COMM_STAT would be 80h. If the master receives any other values for the COMM_STAT byte, some type of error occurred and the master will resend the message.

If a data write message was previously sent to the slave node, the master does not need to receive any more bytes from the slave node after the COMM_STAT byte is read. General call messages do not perform the COMM_STAT read due to their multicast nature. For a data request message, the master will read the COMM_STAT byte plus the number of data bytes specified by DATA_LEN.

A 2's complement, 16-bit checksum is calculated for the data returned to the master. The checksum value includes the COM_STAT byte, plus all data bytes that were returned. The master

device receives two checksum bytes after the data bytes. The 16 bit checksum is in “little Indian“ format – first byte received is LSB second byte received is MSB. If the master determines that a checksum error occurred, it tries to read the data from the slave again.

Bit	Bit Name	Description
0	chkfail	Indicates a checksum failure occurred for the last message sent
1	rxerror	Indicates the slave node did not interpret the last master message correctly
2	ovflow	Indicates the master device has requested to read/write one or more bytes of data outside the valid range of addresses for that particular slave
3	sspov	Indicates an overflow occurred in the SSP module for a given slave address because the slave was not able to process the incoming I2C data fast enough.
4		Unused
5		Unused
6		Unused
7	r_w	Indicates whether the last message from the master was a data request (R/W=1) or a data write message (R/W=0)

Table 1 - COMM STAT Bit Definitions

Application Layer

The host controller interacts with devices using the READ and WRITE packets. Each packet carries a unique body that details the variables and values of interest in the device. A READ packet transmits the variable offset address and length to the device, which then returns the number of bytes requested to the host controller. The WRITE packet transmits a variable offset address, length and new data to a device, which records the new information. The WRITE packet is also used to cause specific actions to occur within the device.

Device Groupings (For Global Call Only)

Slave devices are assigned to group IDs for group level control. Groups enable general calls to specific system devices.

Group 0 calls are global calls to all system devices.

Group 1 calls are specific calls to rectifier devices.

Other groups are reserved for future devices.

Group Types	Grp Num
All Devices	0
Rectifier	1
MPS Low Voltage Disconnect	2
DC/DC Converters	3
Ringer	4
Distribution	5
TRIO	6
Reserved	7

Table 2 - Group Types – bit6, bit5, bit4 of DATA_LEN

Data Type Definitions

Data Types	Length (Bytes)	Data Type Definition
null	0	no value
uchar	1	8-bit unsigned integer
char	1	8-bit signed integer
ushort	2	16-bit unsigned integer
short	2	16-bit signed integer
ulong	4	32-bit unsigned integer
long	4	32-bit signed integer

Table 3 - Basic Data Types

Note: All multi-byte integer data types are LITTLE ENDIAN format (LSB, MSB)

Device Addressing

Each device on the I2C bus must have a unique physical address. Rectifiers are designed to automatically detect their location in a system and assign their individual address. Each system has unique addressing characteristics; the following examples highlight the most common versions:

V Rectifier Shelf Addressing – Compact M-series and N-series shelf Family

SLOT LOCATION	ADDRESS
1	10h
2	12h
3	14h
4	16h
5	18h

Note: Slot 1 is the far left shelf as seen from the front of the system; Slot 5 is the far right slot

X Rectifier Shelf Addressing – 19" SX-series shelf Family

SLOT LOCATION	ADDRESS (ADR1 = LOW and using J1 connector)	ADDRESS (ADR1 = FLOAT and using J1 or J3 connector)
0	10h	18h
1	12h	1Ah
2	14h	1Ch

Note: Slot 0 is the far left shelf as seen from the front of the system; Slot 2 is the far right slot

H and J Rectifier Shelf Addressing – HM-series and JM-series shelf Family

SLOT LOCATION	ADDRESS (ADR1 = LOW and using J1 connector)	ADDRESS (ADR1 = FLOAT and using J1 or J3 connector)
0	10h	18h
1	12h	1Ah
2	14h	1Ch
3	16h	1Eh

Note: Slot 0 is the far left shelf as seen from the front of the system; Slot 3 is the far right slot

Example Packet Transmissions from Master to Slave

Read Status

Slave address =	1Eh	Protocol Offset	=	00h
Slave status =	0107h	Protocol Length	=	02h

I2C command sequence:

Protocol Variable Name	Data on I2C Bus	I2C State Name
	S	Start
Slave Address	1Eh	Address Write
Data Length	82h	Data Write
Data Offset	00h	Data Write
Checksum8	60h	Data Write
	RS	Restart
Slave Address	1Fh	Address Read
ComStat	80h	Data Read
Status LSB	07h	Data Read
Status MSB	01h	Data Read
Checksum16 LSB	78h	Data Read
Checksum16 MSB	FFh	Data Read
	NAK	Not Acknowledge

Write Lamp Test command

Slave address = 1Eh

Protocol Offset = 02h
Protocol Length = 01h
Protocol Data = 01h

I2C command sequence:

Protocol Variable Name	Data on I2C Bus	I2C State Name
	S	Start
Slave Address	1Eh	Address Write
Data Length	01h	Data Write
Data Offset	02h	Data Write
Command Data	01h	Data Read
Checksum8	DEh	Data Write
	RS	Restart
Slave Address	8Fh	Address Read
ComStat	00h	Data Read
	P	Stop

References

Document #	Description
I2C Bus Specification	Philips Semiconductor, Version 2.1 - http://www.semiconductors.philips.com/acrobat_download/literature/9398/39340011.pdf

Rectifier Variable Definitions

Table A.1.1 - Rectifier RAM Variables for IWN Rectifiers

Variable Name	Offset	Data Type	Description	Read/Write
STATUS • STAT_DC_ON • STAT_BOOST_OK • STAT_AC_OK • STAT_HVSD	00h	ushort	Rectifier status 0001h 1 = DC/DC enabled 0002h 1 = Boost Voltage OK 0004h 1 = AC input voltage OK 0008h 1 = Output shut down due to hi-voltage	Read Only
• STAT_FAN_FAIL • STAT_INT_TEMP • STAT_ILIM			0010h 1 = Fan failure 0040h 1 = Heatsink over temperature alarm 0080h 1 = Current limit	
• STAT_UV_ALARM • STAT_UVSD • -------- • STAT_DC_ENABLE			0100h 1 = Output Voltage below UV alarm threshold 0200h 1 = Output shut down due to lo-voltage 0400h Unused 0800h 1 = DC enable asserted	
• STAT_REMOTE_OFF • STAT_MOD_DISABLE • STAT_SHORT_PIN • -------			1000h 1 = Shutdown due to I2C remote off command 2000h 1 = Shutdown due to MOD_DISABLE input 4000h 1 = Shutdown due to short pin transition 8000h Unused	
COMMAND • ------- • CMD_ON • CMD_OFF • CMD_ALARM_ON	02h	uchar	Rectifier command 01h Unused 02h 1 = Turn on Rectifier 04h 1 = Turn off Rectifier 08h 1 = Enable Latent DC Failure Alarm	Write
• CMD_ALARM_OFF • --------			10h 1 = Disable Latent DC Failure Alarm 80h Unused	
LOCATION	03h	uchar	Number that identifies device location in a shelf. (Hi-Nibble = Group type Lo-Nibble = Lo-Nibble of I2C Address)	Read Only
TEMP_SEC_OR	04h	char	Output OR-ing FET temperature in degrees C Range 0C to +116C, 255=short, +127=open	Read Only
TEMP_SEC_DIODES	05h	char	DC/DC output diode temperature in degrees C Range 0C to +116C, 255=short, 127=open	Read Only
MEASURE_VOUT	06h	ushort	Output Voltage in Volts x 100	Read Only
MEASURE_IOUT	08h	ushort	Output Current in Amps x 100	Read Only
SETPOINT_ILIMIT	0Ah	ushort	Current Limit Set Point in Amps x 100	Read/Write
SETPOINT_VOUT	0Ch	ushort	Output Voltage Set Point in Volts x 100	Read/Write
SETPOINT_HVSD	0Eh	ushort	Unit level High Voltage Shut Down Set Point in Volts x 100	Read/Write
SETPOINT_FAN	10h	uchar	Fan period, 38 - Low, 27 – Med, 1B – High, read only	Read Only
RESERVED	11h-3Fh	Null		Read Only

Table A.1.2 - Rectifier EEPROM Variables for IWN Rectifiers

Variable Name	Offset	Data Type	Description	Read/Write
FIRMWARE_VERSION	40h	ushort	Version of firmware in device. 40h - LSB = MM – version minor 41h - MSB = VV – version major	Read Only
SERIAL_NUMBER	42h	uchar[12]	Device serial number as an array of ASCII characters	Read Only
MODEL_NUMBER	4Eh	uchar[8]	Device model number as an array of ASCII characters	Read Only
TEST_DATE	56h	uchar[6]	Date device passed manufacturing test as an array of bytes: 56h - MM – month 1-12 (LSB of array) 57h - DD – day 1-31 58h - YY – year 00-99 59h - hh – hour 1-24 5Ah - mm – minute 0-59 5Bh - ss – second 0-59 (MSB of array)	Read Only
CAPACITY	5Ch	ushort	Maximum current capacity of Rectifier in Amps x 100.	Read Only
CAL_AD_VOUT	5Eh	uchar[5]	Calibration data for VOUT A/D channel 5E - [0] – Scale Factor Numerator 5F - [1] – Scale Factor Denominator LSB 60 - [2] – Scale Factor Denominator MSB 61 - [3] – Offset LSB 62 - [4] – Offset MSB	Read Only
CAL_AD_IOUT	63h	uchar[5]	Calibration data for IOUT A/D channel 63 - [0] – Scale Factor Numerator 64 - [1] – Scale Factor Denominator LSB 65 - [2] – Scale Factor Denominator MSB 66 - [3] – Offset LSB 67 - [4] – Offset MSB	Read
CAL_PWM_VOUT	68h	uchar[5]	Calibration data for VOUT PWM channel 68 - [0] – Scale Factor Numerator 69 - [1] – Scale Factor Denominator LSB 6A - [2] – Scale Factor Denominator MSB 6B - [3] – Offset LSB 6C - [4] – Offset MSB	Read
CAL_PWM_ILIM	6Dh	uchar[5]	Calibration data for ILIM PWM channel 6D - [0] – Scale Factor Numerator 6E - [1] – Scale Factor Denominator LSB 6F - [2] – Scale Factor Denominator MSB 70 - [3] – Offset LSB 71 - [4] – Offset MSB	Read
VSET_DEFAULT	72h	ushort	Power-up value for voltage setpoint	Read/Write
ISET_DEFAULT	74h	ushort	Power-up default for current limit setpoint	Read/Write
HVSD_DEFAULT	76h	ushort	Power-up default for HVSD setpoint	Read/Write
OR_FET_TEMP_THRESHOLD	78h	uchar	Output OR-ing FET temperature in degrees C Range 0C to +116C, 255=short, +127=open	Read Only
DC_RECT_THRESHOLD	79h	uchar	DC/DC output diode temperature in degrees C Range 0C to +116C, 255=short, 127=open	Read Only
RUN_TIME	7Ah	ushort	Rectifier operating time in Hours x 2	Read Only
UV_ALARM_THRESHOLD	7Ch	ushort	Threshold for undervoltage alarm	
Reserved	7Eh	null	Reserved	
CONFIG_DEFAULT	7Fh	uchar	Contains default configuration information:	Read/Write
• HV_RESTART • ALRM_POLARITY • I2C_COMM_TIMEOUT			01h 1=3X restart, 0=no restart 02h 1=contact alarms NC, 0=contact alarms NO 08h 1=Enabled, 0=Disabled	
-------- -------- -------- --------			01h Unused 02h Unused 04h Unused 08h Unused	

Note: Addresses 00h-3Fh access RAM variables. Addresses 40h-FEh access EEPROM variables.