Omega RS485 Transducer Command Reference

This document provides information necessary for communicating with Omega transducers that support RS485 communication, such as the PX409 series. Since typically computers do not natively support the RS485 physical layer, some type of RS485 to computer converter must be used, sold separately. The remainder of this document assumes that the converter supplied presents itself as a COM port to the computer, similar to how an Omega USB device would. Using such a converter allows flexibility for the user to use a terminal program to communicate with the transducer, or to develop a custom API to communicate with the transducer.

Commands are case sensitive. Brackets **[ ]** indicate optional part of command. The following represent ASCII characters: "**$^{S}_{P}$"** = Space (0x20 ASCII); "**$^{C}_{R}$**" = Carriage return (0x0D ASCII); "**$^{L}_{F}$**" = Line Feed (0x0A ASCII); "**>**" = 0x3E ASCII; "**#**" = 0x23 ASCII; "**@**" = 0x40 ASCII. Commands from the master (computer) to the slave (transducer) must begin with a "#" and end with a "**$^{C}_{R}$**". The transducer responds with a leading "**@**".

Sending an invalid command, sending a command before the transducer has responded to a previous command, sending individual ASCII numbers faster than the baud rate of the transducer, or sending a number which is out of range (plus pressing the ENTER key) will return the following response from the transducer:

@[ADDR] @command $^{S}_{P}$ unsupported $^{C}_{R}$ $^{L}_{F}$>

To communicate with the 485 transducer, the PC must be running a terminal program, or similar program that can interface with the chosen RS485 to computer converter. The serial port connection settings are:

Half Duplex

Bits Per Second:	115200
Data Bits:	8
Parity:	None
Stop Bits:	1
Flow Control:	None

Note: When a 485 transducer is connected to the computer, there will be a 5-7 second delay before communication can begin with the transducer. This is due to the time it takes for the transducer's firmware to boot and initialize.

Communication Modes

The PX409-485 transducer has two communication modes:

1) Multidrop Addressed mode – one to 126 transducers can exist together on an independent 485 bus with other transducers in this mode. Each command sent by the master (computer) is preceded by either a) a particular transducer's address, or b) the broadcast address. The transducer will respond with its address and the defined response. The notation **[ADDR]** for each command applies to this mode. [ADDR]= a three digit ASCII string from 000 through 127. The transducer is in addressed mode when RSMODE = 1, which is the default.
2) Stand-alone mode – only one transducer in this mode can on the RS485 bus besides the master (computer). The transducer's address should not be sent to the transducer, and the transducer will not respond with an address. Do not use the **[ADDR]** notation for each command. The transducer is in stand-alone mode when RSMODE = 0.

In both modes, the transducer will respond using the protocol defined in this document. The **[ADDR]** (= Transducer's address or broadcast address) notation is only used in Addressed mode, i.e. only when commands are sent/received in RSMODE =1. All commands are available in either mode, with the exception of the PC and PS commands, which are only available in stand-alone mode.

Transducer Specific Messages

Note: **[ADDR]** is the Transducer's address, is a three byte ASCII number, and is only used when commands are sent/received in RSMODE = 1. Default Transducer Address = 123.

1) RSMODE – Read/write. Returns/sets the transducer's communication mode.

   Syntax: **# [ADDR] RSMODE [$^{S}_{P}$ n] $^{C}_{R}$ [$^{L}_{F}$]** where **n** is a 1 (addressed mode) or a 0 (stand-alone mode)

   Returns: **@ [ADDR] RSMODE $^{S}_{P}$ = $^{S}_{P}$ x $^{C}_{R}$ $^{L}_{F}$>** where x is a 1 (addressed mode) or a 0 (stand-alone mode)

2) ENQ – Read only. Returns the transducer's Unit ID, firmware version, range, and engineering units, all in ASCII. The Unit ID for PX409-485 transducers is "485PX1".

   Syntax: **# [ADDR] ENQ$^{C}_{R}$ [$^{L}_{F}$]**

   Returns**: **@ [ADDR] aaaaabb $^{C}_{R}$ $^{L}_{F}$** (transducer's Unit ID) where "a" is an Upper Case character or positive integer (returned as an ASCII character), and b is a positive integer, returned as an ASCII character. – This field indicates the type of transducer.

   c.c.cc.cccc $^{C}_{R}$ $^{L}_{F}$ where c is a positive integer returned as an ASCII character.

   Snnnnnnn.mmm $^{S}_{P}$ to $^{S}_{P}$ Snnnnnnn.mmm $^{S}_{P}$ [xxxxxxxx $^{S}_{P}$ Z] $^{C}_{R}$ $^{L}_{F}$> (transducer range and engineering units) where both **n** and **m** are ASCII characters representing integers between 0 and 9. **n** has a length between 1 and 7 digits, and **m** has a length between 0 and 3 digits. **x** represents the engineering units of the transducer. **x** can be any ASCII character, positive integer, or symbol, and is 0-8 characters in length. **x** can also be omitted from the response (in this case, **Z** will also be omitted). **Z** can be "A", "G", "D", "V", or blank. **S** is a sign character, and is either the minus ASCII character (0x2D), or nothing. **n**, **m**, **x**, the decimal place, **S**, and **Z** are returned as ASCII characters.

3) IFILTER - Read/Write. Reads or sets the IIR filter period (time constant). Default = 0.

Syntax: **# [ADDR] IFILTER [$^{S}_{P}$ nnn] $^{C}_{R}$ [$^{L}_{F}$]** where optional **nnn** is 0 or 1 (disabled), or between 2 to 255.

Returns: **@ [ADDR] I $^{S}_{P}$ = $^{S}_{P}$ xxx $^{C}_{R}$ $^{L}_{F}$>** where **xxx** is an ASCII representation of the IFILTER setting, and can be 1, 2, or 3 bytes long.

4) MFILTER – Read/Write. Reads or sets the Moving Average filter order. Default = 4.

Syntax: **# [ADDR] MFILTER [$^{S}_{P}$ nn] $^{C}_{R}$ [$^{L}_{F}$]** where optional **nn** is 0 or 1 (disabled), or between 2 to 63.

Returns: **@ [ADDR] M $^{S}_{P}$ = $^{S}_{P}$ xx$^{C}_{R}$ $^{L}_{F}$>** where **xx** is an ASCII representation of the MFILTER setting, and can be 1 or 2 bytes long.

5) AVG – Read/Write. Reads or sets the number of data points to be averaged for the boxcar average filter. Valid values are 0, 2, 4, 8 and 16. Note: the output rate is determined by the RATE command setting divided by this value (excluding 0). AVG x sets the averaged number. Note: the boxcar changes the rate of the readings returned by the PC command. This is because the boxcar averages the specified number of readings given by **nn**, and outputs one reading for the group. Default = 0.

Syntax: **# [ADDR] AVG [$^{S}_{P}$ nn] $^{C}_{R}$ [$^{L}_{F}$]** Where **nn** is an optional number.

Returns: **@ [ADDR] AVG $^{S}_{P}$ = $^{S}_{P}$ xx $^{C}_{R}$ $^{L}_{F}$>** Where **x** is 0 or 1 (disabled), or 2, 4, 8 or 16.

6) RATE – Read/write. Reads or sets the transducer update rate. Valid Values are 0=5sps, 1=10sps, 2=20sps, 3=40sps, 4=80sps, 5=160sps, 6=320sps, 7=640sps. Default = 320SPS

Syntax: **# [ADDR] RATE [$^{S}_{P}$ nn] $^{C}_{R}$ [$^{L}_{F}$]** where **nn** = a 1-2 digit positive integer representing samples per second as defined above (two bytes for future flexibility)

Returns: **@ [ADDR] RATE $^{S}_{P}$ =xx $^{C}_{R}$ $^{L}_{F}$>** where **xx** is an ASCII representation of the RATE setting (as defined above), and can be 1-2 bytes long.

7) P – Read. Sends single ASCII reading (decimal point also sent as ASCII). Data is post filter, and scaled to the native engineering units and type of transducer.

Syntax: **# [ADDR] P $^{C}_{R}$ [$^{L}_{F}$]**

Returns: **@ [ADDR]** (variable length ASCII packet) **$^{C}_{R}$ $^{L}_{F}$>**

Example: "**-0.016 $^{S}_{P}$ PSI $^{S}_{P}$ G$^{C}_{R}$ $^{L}_{F}$>**"

will be "**2D 30 2E 30 31 36 20 50 53 49 20 47 0D 0A 3E**" hex.

8) PC – Starts continuous stream of readings from the transducer, at an update rate specified by the RATE command. Data is in 4 byte IEEE 754 format (1 bit sign, 8 bits exponent, 23 bits significand (mantissa)), plus sync byte(s), plus packet type. Data is post filter, and is a scaled floating point representation of the transducer's native engineering units. Data is sent Little Endian to be compatible with the PC. The "PC" command is only available in standalone mode.

Syntax: **# PC $^{C}_{R}$ [$^{L}_{F}$]**

Returns: Streaming 7 to 11 byte packet at the rate of the current RATE setting. The packet is variable length because if one or more data bytes are 0xAA, a 0xAA bit stuff byte is added after each 0xAA data byte occurrence, whereby the sync byte is the only occurrence of a single 0xAA byte. This allows the user PC software to parse each message accurately and easily.

Byte #	1	2	3	4	4a	5	5a	6	6a	7	7a
Data	@	AA	3B	XX	AA	XX	AA	XX	AA	XX	AA
Description	Start of Frame Character	sync	packet type	LSB float	bit stuff**	data	bit stuff**	data	bit stuff**	MSB float	bit stuff**

*XX = 0x00 to 0xFF

** optional- only included if previous data byte is 0xAA, otherwise not present

Note: Stream must be stopped ("PS" command) before any other command can be executed.

9) PS - Stops continuous stream of readings. This command is only available in stand-alone mode.

Syntax: **# PS $^{C}_{R}$ [$^{L}_{F}$]**

Returns: Null

10) B – Read. Sends single Binary reading.

Syntax: **# [ADDR] B $^{C}_{R}$ [$^{L}_{F}$]**

Returns: **@ [ADDR][ieee 754 #] $^{C}_{R}$ $^{L}_{F}$>**

iEEE 754 # = 32 bit floating point number in IEEE 754 format (4 bytes), Little Endian.

11) UADR – Read/write. Returns/sets the transducer's address.

Syntax: **# [ADDR] UADR [$^{S}_{P}$ nnn] $^{C}_{R}$ [$^{L}_{F}$]** where nnn = a 3 digit positive integer between 001 and 127, representing the transducer's address. Omit **[$^{S}_{P}$ nnn]** to query the transducer's address. Default is Address 123.

Returns: **@ [ADDR] UADR $^{S}_{P}$ =xxx $^{C}_{R}$ $^{L}_{F}$>** where **xxx** the transducer's address.

12) SNR – Read. Reads the transducer's serial number, which is a factory assigned value.

Syntax: **# [ADDR] SNR $^{C}_{R}$ [$^{L}_{F}$]**

Returns: **@ [ADDR] SNR $^{S}_{P}$ =xxxxxx $^{C}_{R}$ $^{L}_{F}$>** where **xxxxxx** is the serial number of the transducer.

13) TERM – Read/Write. Reads/Sets the internal 120 Ohm termination resistor across A and B. Default = 0 (termination resistor off).

Syntax: **# [ADDR] TERM [$^{S}_{P}$ n] $^{C}_{R}$ [$^{L}_{F}$]** where **n** is an ASCII 1 (add termination resistor) or 0 (remove termination resistor)

Returns: **@ [ADDR] TERM $^{S}_{P}$ = x $^{C}_{R}$ $^{L}_{F}$>** where **x** is an ASCII 1 (termination resistor is present) or 0 (termination resistor is not present)

14) ANAEN – Read/Write. Reads/Sets the Analog output state (on/off). Default = 1 (output on).

Syntax: **# [ADDR] ANAEN [$^{S}_{P}$ n] $^{C}_{R}$ [$^{L}_{F}$]** where **n** is an ASCII 1 (analog output is on) or 0 (analog output is off)

Returns: **@ [ADDR] ANAEN $^{S}_{P}$ = x $^{C}_{R}$ $^{L}_{F}$>** where **x** is an ASCII 1 (analog output is on) or 0 (analog output is off)