

# DATA STREAM RS485 DIGITAL TRANSDUCER

## Programming

Programming of the CR Magnetics DATA STREAM Series is straight forward and easily implemented via user programs or The DATA STREAM Software. The programming language is ASCII based and consists of Commands and Return data.

List of Common Commands				
Command	Preamble	Addr	Dir	Data
Read Transducer Name	\$	01 to FF	M	N/A
Read All Data	#	01 to FF	A	N/A
Read Energy Totalizer	#	01 to FF	W	N/A
Read Transducer Configuration	\$	01 to FF	2	N/A
Set Transducer Configuration	%	01 to FF	N/A	New Addr, Input Range, Baud
Clear Energy Totalizer	&	01 to FF	N/A	Totalizer Period Number
Variable Delay	<	01 to FF	N/A	Adjustable Delay

### Command Structure

Each command sent to the DATA STREAM has the following structure:

Preamble Addr [Dir] [Data] <cr>

- Preamble** A single ASCII character which designates type of command.
- Addr** 2 ASCII characters which selects the address of the transducer.
- [Dir]** A single ASCII character directive which qualifies the command, not used on all commands.
- [Data]** A set of ASCII characters required by certain command functions.
- <cr>** A carriage return must be placed as an end of line character.

### Read Transducer Name Command

Preamble \$  
 Addr 01 to FF  
 Dir M  
 Data Not Used

**Transducer Correct Response:** !  
 Addr (01 to FF)  
 <cr>

**Transducer Wrong Response:** ?  
 Addr (01 to FF)  
 <cr>

**Example:** Send \$01M<cr>  
 Receive !01CRD5110-150-5<cr>

In this case, a CRD5110 Single Phase Multifunction Transducer with a range of 150 Volts and 5 amps is programmed with the address 01, and returns the part number with address

### Read All Data Command

Preamble #  
 Addr 01 to FF  
 Dir A  
 Data not used

**Transducer Correct Response:** Voltage % (+/-X.XXXX)  
 Current % (+/-X.XXXX)  
 Power % (+/-X.XXXX)  
 VARS % (+/-X.XXXX)  
 Power Factor % (+/-X.XXXX)  
 Frequency (XX.XXX)  
 <cr>

**Transducer Wrong Response:** ?  
 Addr (01 to FF)  
 <cr>

**Example:** Given a CRD5110, programmed with a Voltage range of 500VAC RMS, and a Current range of 5 AAC RMS, Address 1B.  
 Input Voltage = 300 VAC RMS,  
 Input current = 4 AAC RMS,  
 Power Factor = 1.000 (Pure resistive)  
 Frequency of 50 Hz.

**Send** #1BA<cr>  
**Receive** >+0.6000+0.8000+0.4800+0.0000+1.000050.000<cr>

All signed data is returned as a percent of Full Scale.  
 Thus the formula:

$$\text{Value} = \text{Data}(\%) \times \text{Full Scale}$$

Will give the actual value of the parameter.

For this example:  
 Voltage = +0.6000% X 500 = 300 VAC RMS  
 Current = +0.4000% X 5 = 4 AAC RMS  
 Real Power = +0.4800% X 500 X 5 = 1200 Watts  
 VARs = +0.0000 X 500 X 5 = 0.0000 Vars  
 Power Factor = +1.0000 X 100% = 100% PF (purely resistive)

The Frequency Value is the Actual Data:  
 Frequency = 50.000 Hertz

**NOTE:** Multifunction units will return Voltage, Current, Power, Vars, Power Factor, and Frequency. Single function units will return only the single function. For Multiphase units, the Voltage and Current Data will be returned in the following order:

#### 3 Phase 3 Wire

Voltage <sub>1,2</sub>, Current <sub>1</sub>, Voltage <sub>3,2</sub>, Current <sub>3</sub>,  
 Power, Vars, Power Factor, Frequency

#### 3 Phase 4 wire

Voltage <sub>1</sub>, Current <sub>1</sub>, Voltage <sub>2</sub>, Current <sub>2</sub>, Voltage <sub>3</sub>,  
 Current <sub>3</sub>, Power, Vars, Power Factor, Frequency.



3500 Scarlet Oak Blvd. St. Louis MO USA 63122 V: 636-343-8518 F: 636-343-5119

Web: <http://www.crmagnetics.com>

E-mail: [sales@crmagnetics.com](mailto:sales@crmagnetics.com)

## Programming (continued)

### Read Energy Totalizer Command

**Preamble** #  
**Addr** 01 to FF  
**Dir** W  
**Data** not used

Transducer Correct Response:  
 Period Counter (01 to FF)  
 KWHr Data (+/-XXXXXX)  
 KVHr Data (+/-XXXXXX)  
 Checksum (XX)  
 <cr>

Transducer Wrong Response: ?  
 Addr (01 to FF)  
 <cr>

#### Description of Totalizer Function

The energy totalizer function is a method by which the total amount of energy in Kwhrs and KVARHrs used over a period of time can be measured. A data location within the program keeps a running total of the amount of energy used. This total starts from zero and begins totalizing the instant power is turned on to the transducer.

Totalizer data is outputted when the Read Energy Totalizer Command is sent to the transducer. At this point, the transducer outputs the amount of energy totaled since the last clear or restart. A counter called the time period counter, keeps track of all instances of requested totalizer data. Thus, at turn on, the period counter is set to zero. After the first read of totalizer data, the counter is incremented by 1 and reads 1. Future requests for data increase the period counter by 1. This counter turns back to zero after 255 counts (FF).

It is important to know that the output data represents the amount of energy used since the last clear or restart. This method makes it very useful for total energy used calculations.

A Clear Totalizer Command function is also provided to reset the totalizer as needed.

The largest period of totalizer data is 1,553.5 Hours with the full scale voltage and current being input to the device. Longer periods are possible with lower input voltages and currents.

The data output has gain and function factors that must be included when calculating actual Kwhr/KVARHr. The formula is:

$$\text{Energy} = \text{Data Read} \times \text{Volt Range} \times \text{Current Range} / 3600000$$

Units are Kwhrs and KVARHrs depending on data read.

Please note: Unlike other data read from the transducer, the energy data is transmitted as the ASCII representation of the HEX value. Hence, when the data would return as ASCII "FF", this is equivalent to the decimal value 255.

Checksum data is provided as a verifier of correct data communication. The Checksum data is generated by totaling the sum of all byte by byte data, in HEX, sent from the transducer after the Read Energy Totalizer Command is accepted, then taking this HEX value and performing a logical AND with the value FFH, and outputting it. The data is an ASCII representation of the HEX value.

**Example:** Given a CRD5110, programmed with a Voltage range of 500VAC RMS, and a Current range of 5 AAC RMS, and Address 1B. Input Voltage =300 VAC RMS, Input current = 4 AAC RMS, Power Factor = 1.000 (Pure resistive) and a frequency of 50 Hz. Energy data has not been requested since turn on of the part. The command is sent after the part has been running one hour.

**Send** #1BW<cr>

**Receive** >01+0006C0+0000004E<cr>

Time Period = 01H = 1 Decimal  
 KwHr Data = +0006C0H = 1,728 Decimal  
 KwHr Actual = 1,728 X 500 X 5 / 3600000 = 1.2 Kwhrs  
 KVARHr Data = +000000H = 0 Decimal  
 KVARHrs Actual = 0 KVARHrs  
 Checksum = 1EH

Doing the Check on the Checksum:

3EH+30H+31H+2BH+30H+30H+30H+36H+43H+30H+2BH+30H+30H+30H+30H+30H+30H = 34EH AND FFH = 4EH

### Read Transducer Configuration Command

**Preamble** \$  
**Addr** 01 to FF  
**Dir** 2  
**Data** not used

**Transducer Correct Response:** !  
 Addr (01 to FF)  
 Settable Delay (01 ~ FF)  
 Baud Rate Code (See Below)  
 Data Format (Always 01)  
 <cr>

**Transducer Wrong Response:** ?  
 Addr (01 to FF)  
 <cr>

**Example: Send** \$0A2<cr>  
**Receive** !0A000601<cr>

In this case, a CRD5110 Multifunction transducer is programmed with the address 0A. Reading the configuration returns 0A for the address, 00 for the Input Range (fixed), 06 for 9600 bps Baud rate, and 01 for no check sum on standard data (Data Format is fixed).

Baud Rate Codes are as follows:

03 = 1200 bps  
 04 = 2400 bps  
 05 = 4800 bps  
 06 = 9600 bps (Default)  
 07 = 19200 bps

08 = 38400 bps

09 = 57600 bps

0A = 115200 bps

## Programming (continued)

### Set Transducer Configuration Command

**Preamble**            %  
**Addr**                01 to FF  
**Dir**                 not used  
**Data**                New Addr - 01 to FF  
                       Input Range - Always Set To 00  
                       Baud Rate - 03 to 0A  
                       Data Format - Always Set to 01

**Transducer Correct Response:** !  
                                   New Addr (01 to FF)  
                                   <cr>

**Transducer Wrong Response:** ?  
                                   Addr (01 to FF)  
                                   <cr>

**Example: Send**        %0A0B000701<cr>  
**Receive**            !0B<cr>

In this case, a CRD5110 Multifunction transducer is programmed with the address 0A. The command changes the address to 0B, and sets the baud rate to 19,200 bps.

### Clear Energy Totalizer Command

**Preamble**            &  
**Addr**                01 to FF  
**Dir**                 not used  
**Data**                Time Period Count

**Transducer Correct Response:** !  
                                   Addr (01 to FF)  
                                   <cr>

**Transducer Wrong Response:** ?  
                                   Addr (01 to FF)  
                                   <cr>

#### Description of Clear Totalizer Function

The Clear Energy Totalizer Command is a method by which the internal energy totalizer can be reset to zero. The programmer simply addresses the transducer with the appropriate address, and also sends the correct time period number which will then clear the totalizer to zero.

The correct time period number is the value that was received with the last Read Totalizer Data Command. An incorrect period number will result in a command error, and the totalizer will not be cleared. Clearing the totalizer properly does not reset the Time Period Counter. Upon proper totalizer clear, the Time Period Counter will be increased by 1.

Should an invalid Clear Totalizer Command be refused (because of an incorrect address or an incorrect Time Period Number), the totalizer will not be cleared, energy data will continue to be totalized to the sum, and the Time Period Counter will not be incremented.

**Example:** A CRD5110 with Address 0A needs to have its totalizer cleared. The totalizer data has been read 3 previous times.

**Send:**            &0A04<cr>  
**Receive:**        ?0A<cr>

The CRD5110 received an order to clear the totalizer, but an improper time period number was received.

**Send:**            &0A03<cr>  
**Receive:**        !0A<cr>

The command was sent properly, as the correct time period number was sent. Please note that the Time Period Counter has been increased by 1, so the next Read Totalizer Command will give a time period number of 5.

### Set Variable Delay Command

**Preamble**            <  
**Addr**                01 to FF  
**Dir**                 Not Used  
**Data**                Delay - 01 to FF

**Transducer Correct Response:** !  
                                   Addr (01 to FF)  
                                   <cr>

**Transducer Wrong Response:** ?  
                                   Addr ( 01 to FF)  
                                   <cr>

**Example: Send**        < 01A0<cr>  
**Receive**            !01<cr>

In this case, a CRD5110 Single Phase Multifunction Transducer with a range of 120 Volts and 5 amps is programmed with the address 01, and returns its last 4 digits to the system. The Delay command changes the variable delay to 0xA0 (160 Decimal, adding 160ms response delay. Default value of 0x01 (1 Decimal), changeable to 0xFF (255 Decimal) giving an approximate response delay between 1-2.55ms.

### Reset to Factory Defaults Command

As an aid to system startup, a Reset to Factory Defaults Command is provided. Since this command resets the transducer to Address 01 and Baud rate 9600 bps, no address is needed to run the command.

**NOTE: THIS COMMAND CANNOT BE USED ON NETWORKED TRANSDUCERS. ALL TRANSDUCERS ON THE NETWORK WILL BE RESET, AND THE RESULTING RETURN INFO WILL CAUSE NUMEROUS BUS CONFLICTS.**

#### THIS IS FOR ONE TO ONE CONFIGURATIONS ONLY !!!

**Example:** Need to reset a transducer to Factory settings via a RS232 comm port.

**Receive:** !01<cr>

**Send:**            @CEAFW<cr>

### Read Software Revision > or = 2.13Version

**Send**                \$XXV <cr>  
                           XX - Address

**Response**            !XXY.YY <cr>  
                           Y.YY = Software Revision