HC 485 Operation Manual

POSITION

The basic function of the sensor is to measure position. This is calculated in engineering units using built in calibration values. Position is normally given as the positive or negative displacement from center stroke, or null position.

The calibrated range of the sensor is stored internally, in memory. If the position moves outside this range, the reading will become non-linear, and eventually reach hard limits. Although this can easily be calculated externally, the sensor Status register also provides under and over range indication.

ZERO (TARE)

Normally the LVDT gives a zero reading with the core at the electrical center of the coil, (null position), reading positive as the core is displaced toward the connector end and negative when displaced toward the open end.

The Zero (also referred to as Tare) function allows an alternate zero reference to be used. This is particularly useful in situations where an initial setup of the equipment is required to remove linkage or machine variability. The zero function may also be used when a unipolar output is desired, by performing a zero at end of the stroke.

Activating the zero function establishes an offset zero reference at the current position. A position measurement made immediately after this (assuming position was not changed) will indicate zero (within system resolution).

The Zero function can be removed at any time by writing 0 to the register.

The Zero function can be made semi-permanent by saving the setup to non-volatile memory.

FILTER

The sensor contains advanced digital filtering that provides optimum filtering for position calculations. This is set to provide 200Hz+ bandwidth with a rapid roll off, at higher frequencies.

In certain situations, small, low frequency changes in indicated position or bit dithering may occur. This noise may obscure the intended measurement. To address this problem, additional filtering can be selected by adjusting the **Filter** value.

The use of this filter used does not introduce any distortion or aliasing effects, and does not affect sampling rate or communications speed.

Increasing the filter value increases the filter. A value of one indicates no filtering is employed (full bandwidth). In general terms the filter can be thought of as an additional first order filter whose time constant is (n-1) x Sample rate.

N	Time	Constant Equivalent analog frequency
1	-	200Hz+
2	1.5mS	100Hz
3	3mS	50Hz
100	150mS	1Hz

The filter count can be saved to non-volatile memory.

UNITS

The output of the LVDT is always a single precision floating point value in real position units.

The units may be changed using the supplied setup utility or programmatically to:

meters
centimeters
millimeter
inches
thousands of an inch (mils)
micro-inches

This selection is simply a mathematic scaling of the position measurement. It does not affect calibration, accuracy, resolution or any other performance parameter.

MINIMUM, MAXIMUM AND TOTAL INDICATED RUNOUT (TIR)

Some applications require recording minimum and maximum position readings throughout a measurement cycle, without continuously polling the sensor.

Minimum and maximum position values are computed and updated, at 650 samples per second, and available for reading upon request. The values may be reset to the current position using the **Reset register**.

Total Indicated Runout (TIR) is the arithmetic difference between min and max. It is often used to indicate runout on an eccentric shaft.

RESET

This is a write only register that immediately clears the Min, Max and TIR values.

VELOCITY

Velocity is computed on a sample-by-sample basis and is calculated as the change in position, divided by the change in time ((s/(t). The active position units and filtering are both applied before velocity calculation.

Seeing extraordinary velocity figures when the system is stationary or hardly moving sometimes confuses people. However, the measurement is calculated over a very short period of time. Even a tiny change in position in 1/650 of a second is a high velocity. Filtering of the position signal alleviates this “noise” to a degree but it cannot remove the noise due to quantization error (about 1/40000).

USER ID

Four registers, (64 bits total), are available to the user. There is no restriction on what can be stored. This might be dates, serial numbers or location information to support configuration or traceability. These values may ONLY be changed using the HC-485 configuration software.

STATUS

The Status register gives basic information on sensor operation. This includes the state of the Zero function as well as over and under range information. The format of the Status register is as follows.

Bit	**Clear (0)**	**Set (1)**
15	No communication timeout	Communication timeout
14	Unused	
13	No parity error	Parity error
12	No under range	Under range
11	No over range	Over range
10	Unused	
9	Defined for factory test only	
8	Defined for factory test only	
7	Unused	
6	Unused	
5	Even parity	Odd parity
4	Parity disabled	Parity enabled
3	Communication echo disabled	Communication echo enabled
2	Modbus ASCII mode	Modbus RTU mode
1	Fixed point output	Floating point output
0	Modbus communication mode	I Series communication mode

CHANGING SENSOR OPERATION

Variables below may be set to change sensor operation. Some changes take immediate effect and others, (mostly associated with communications protocol), will not take effect until saved to internal, nonvolatile, memory and power is cycled. These functions are listed below, along with affectivity:

Parameter	**Effect**
Zero	Immediate
Units	Immediate
Filter	Immediate
Address	Save and restart
Baud rate	Save and restart
Parity	Save and restart
Modbus mode	Save and restart
Communication mode	Save and restart
Communication echo	Save and restart
Lead character	Save and restart
Tail character	Save and restart
Fixed/floating point	Save and restart

All of these values can be read back from the sensor. All of these values may be saved in non-volatile memory. Any values that are not changed will be saved in their original state.

It is possible to key protocol information over the standard protocols, however, it is highly recommended that the supplied configuration utility is used. This software has the ability to recover the sensor from some mysterious protocol setup that the user may have inadvertently set.

SAVING SETUP

The Save Setup command is executed by a specific write to one of the registers. This takes current sensor setup data and stores it in non-volatile memory. This will take full effect on the next power on or restart.

The recommended sequence for changing the setup is:

Switch On
Change required parameters
Check everything is set correctly.
Activate "Save Setup"
Verify it was saved correctly
Switch off and back on again.

The supplied setup routine does all of these steps. It is recommended that this be used for sensor setup since it can also cope with any initial protocol setup.

Data integrity is verified by checking each value stored in EEPROM. In addition, an internal checksum is used to verify data restore. Hardware and software are specifically designed to guard against data loss or corruption.

Note: Min, Max and TIR values are strictly volatile and cannot be saved.

COMMUNICATIONS

Serial communications of 7 and 8 bits are supported at baud rates up to 19200. Odd, even, mark, and space parity are supported in ASCII modes only. These parameters may be changed using the supplied setup utility or programmatically. Programmatic changes to communications or protocol settings do not take effect until the data is saved and the unit is restarted. The setup utility program handles this automatically.

DEVICE ADDRESS

For multi-drop communication, every device must be assigned a different device address. The device address for multidrop communication may be changed from the factory default setting of 1, to any value in the range between 1 and 247.

PROTOCOLS

Modbus RTU, Modbus ASCII, Generic ASCII, and I-series ASCII are the standard protocols supported by the sensor. The Generic ASCII format may be used to emulate the Newport Instruments I-series protocol and connect to RD-4 or RD-6 display modules and the I-Server Ethernet module.

MODBUS

Modbus is widely used protocol for RS-485 and other digital networks. A detailed explanation of Modbus is beyond the scope of this document. The specifications are freely available at:

http://www.modicon.com/techpubs/toc7.html

The sensor supports both the RTU and ASCII modes of the Modbus protocol. The RTU mode uses the full 8 bits of a character to send binary data. The ASCII mode only uses 7 bit printable ASCII characters and hexadecimal coding of binary data. Modbus RTU is roughly twice as fast as Modbus ASCII.

The sensor implements the following Modbus functions:
- 4 Read Input Registers
- 6 Preset Single (Holding) Register
- 8 Diagnostic

Function 4 permits multiple register reads. Multiple register writes are **not supported.**

Diagnostic sub-functions include:
- 0 Echo
- 1 Restart
- 2 Status
- 3 Change ASCII input delimiter (Tail)
- 4 Listen only mode

REGISTER DEFINITIONS

Register Definitions

Register	Read Function (4)	Format	Write Function (6)	Range
0	Position	IEEE float	Read only	
1				
2	Minimum	IEEE float	Read only	
3				
4	Maximum	IEEE float	Read only	
5				
6	Velocity	IEEE float	Read only	
7				
8	Runout	IEEE float	Read only	
9				
10	Status		Read only	
11	Unused			
12	User ID1	32 bits	Read only	
13				
14	User ID2	32 bits	Read only	
15				
16 - 31	Unused			
32	Unused		Reset	0
33	Unused		Zero (Tare)	0 or 1
34	Filter count		Filter count	1 - 100
35	Units		Units	0 - 5 (table 1)
36	Address		Address	1 - 247
37	Baud rate		Baud rate	0 - 3 (table 2)
38	Precision		Precision	1 - 8
39	Format		Format	0 - 254 (table 3)
40	Lead character		Lead character	0 - 255
41	Tail character		Tail character	0 - 255
42	Unused		Save setup	0xAA

Units	Value
Meters	0
Centimeters	1
Millimeters	2
Inches	3
Thousands (mil)	4
Micro-inches	5

Table 1

Baud Rate	Value
19200	0
9600	1
4800	2
2400	3

Table 2

Bit	Set (1)	Clear (0)
5	Odd parity	Even parity
4	Parity enable	Parity disable
3	Echo enable	Echo disable
2	Modbus ASCII	Modbus RTU
1	ASCII fixed point	ASCII floating point
0	Modbus	ASCII/I Series

Table 3

ERRORS/EXCEPTIONS

Errors that occur in a message that may indicate a corrupt message will not generate a response from the sensor.

Exception responses will occur for the following reasons:

01	ILLEGAL FUNCTION	The message function is not implemented.
02	ILLEGAL DATAADDRESS	The address (register) does not exist.
03	ILLEGAL DATA VALUE	The value written is not allowable.
04	FAILURE IN ASSOCIATED DEVICE	The device has failed.

MODBUS DATA FORMAT

Modbus was designed based on 16 bit long “word” registers. These registers were called holding registers, input registers and coils reflecting their PLC roots. The sensor uses the definition “holding registers” to relate to the sensor read and write parameters.

Note: Modbus documentation numbers registers from 1, but the messages format numbers them from zero (confusing huh?).

The digital sensor has to work with 8, 16 and 32 bit values so some trickery is required. Firstly, all 16 and 8 bit read writes occur to normal Modbus registers. For 8 bit values the unused part is sent as zero.

For 32 bit values (like floating point values), two adjacent registers have to be read. The order of these words is critical to interpreting the number correctly. The lowest address is the LEAST significant.

Floating-point values are encoded according to IEEE754. (This is a common format used by many computers and software including Microsoft Visual C++ and Visual B)

IEEE 4-byte floating-point format

Most Significant		Least Significant
1 bit	8 bits	21 bits
Sign	Exponent	Mantissa

The value of the number is:
(-1)S * 2(Exponent-127) * 1.Mantissa
Zero is represented by 4 bytes of zeros.
The precision is approximately 7 decimal places

Note: This works for little endian processors such as the Intel X86 architecture. For big endian processors the most significant byte and least significant byte of each register must be swapped.

GENERIC/I-SERIES ASCII PROTOCOL

The generic ASCII and I-series protocols can send sensor data out as plain ASCII text numbers.

For example; a position of 1.054321 comes out as “1.054321”. While this seems logical, it takes a lot more characters to send out this than RTU and can a lot more effort to read and process into a computer/ PLC.

However, ASCII can often be used to provide a simple method to hook simple display and data logging setups where the system merely displays or prints the sensor response. Indeed, the Newport instruments RD4 display can be connected directly to the sensor and display position without requiring any computer or complex programming.

It should be noted that the generic format is not designed to be a replacement for the more robust and higher performance Modbus. However, it is possible to achieve the same functionality.

MESSAGE FORMAT

The message is composed of three parts:

Lead character
Body
Tail character

The lead character has to be recognized as the start of message (rather than some odd fragment that happens to be flying around). It is generally a unique character not used elsewhere in the messages.

The body of the message contains the device address, function and any data.

The first two hexadecimal characters indicate the device address. The next character is the function, followed by any associated data in hexadecimal.

Prrvvvv	Put or write value to (16 bit) register “rr” is two hexadecimal characters indicating register number “vvvv” is four hexadecimal characters indicating value to be written
Grr	Get or read value from (16 bit) register “rr” is two hexadecimal characters indicating register number
Z02	Restart
Xrr	Send floating point value as ASCII string “rr” is two hexadecimal characters indicating register number

The tail character terminates all messages, which mark the end of message.

Examples:

*02P010034	Write 0034h to register 1 in device address 2
*09G10	Get register 16 in device address 9 as hexadecimal string (e.g. 90CD5F34)
*03Z02	Restart device with device address 3
*02X01	Get register 1 in device address 2 as ASCII string (e.g. 12.345)

MESSAGE REPLY

The response from the sensor may optionally echo back the transmitted command. The echo bit of the format register determines this is active. If set to 1, the command message is echoed back from the sensor, followed by the command response:

P	For write operations there is no response.
G	The response is hexadecimal representation of 4 bytes.
X	Sends data in fixed point (e.g. "0.234") or floating point decimal (e.g. 1.234e1).
Z	No response

The tail marks the end of message and is **always** sent.

Example:

Get register 16 in device address 9 as hexadecimal string. Echo enabled and register contains 0x90CD5F34. Head and tail characters are set to "*" and <cr>.

Master Sends: *09G10<cr>

Sensor Sends: *09G1090CD5F34 <cr>

MESSAGE TIMING

The maximum time between characters of a message to the sensor must not be longer than one second.
Changing Protocol characteristics.
Some of the generic protocol message characteristics can be changed to use in other systems.

Lead Character
Change the character that marks start of message. This can be set to any 7-bit character except 0 (null)

Tail Character
Change the character that marks start of message. This can be set to any 7-bit character except 0 (null)

Float/Fixed and Precision
These settings affect the "X" command only. The X command outputs a floating-point number in ASCII. The number of digits given is set by the Precision value. The format may be either fixed (e.g. 20.122) or scientific
(e.g. 2.0122e-1)

Command Echo On/Off
Turn on to echo the command as part of the command response. This should always be set when used with the Newport Instruments I-Server or RD4/6 display.

I-Series

The following settings allow use of the I-Series RD4 display with the sensor.

Head="*"
Tail=<cr> 0x0D (carriage return)
Fixed point
Precision=4
Echo On

The RD4 communications parameters, should be set to match the sensor as follows:

Communications Standard = 485
Baud Rate = Sensor baud rate
Data Format = Sensor Parity
Mode = Host
Device address = sensor device address
Interface device - doesn't matter

The display will then show position.

READING AND WRITING VALUES

Unlike Modbus, read and write operations to the same variable (such as units) are two DIFFERENT addresses.

All write operations (W commands) take a 16 bit hex values. No 32 bit writes are possible. Registers are numbered by word address, starting at 1.

All read operations (R commands) take 32 bit values (8 hex digits). All registers addresses are a 32-bit register address starting at 1.

ERRORS

Errors in the format of the messages will not result in an error message coming back from the sensor. This may seem odd but this is done to prevent a bad message causing in a flood of error messages on the RS485 bus from every device at the same time.

Messages with the correct protocol but some other problem (like trying to read a register that doesn't exist) will result in an error message.

The I-series/Generic ASCII protocol uses two errors that are used by the sensor. These are:

?43	Communications error	Unimplemented function or bad value (writes)
?46	Format Error	Bad hex number or incomplete message,

These will only be reported if the first part of the message (Head and Address) is valid.

ORDERING INFORMATION

NORTH AMERICA

Measurement Specialties
1000 Lucas Way
Hampton, VA 23666
United States
Tel: 1-800-745-8008
Fax: 1-757-766-4297
Sales: pvg.cs.amer@meas-spec.com

EUROPE

Measurement Specialties
(Europe), Ltd.
Hauert 13
44227 Dortmund
Germany
Tel: +49 (0) 231 9740 0
Fax: +49 231 9740 20
Sales: pvg.cs.emea@meas-spec.com

ASIA

Measurement Specialties
(China), Ltd.
No. 26 Langshan Road
Shenzhen High-Tech Park (North)
Nanshan District, Shenzhen 518107
China
Tel: +86 755 3330 5088
Fax: +86 755 3330 5099
Sales: pvg.cs.asia@meas-spec.com

DEALER / REVENDEDOR

Brazil and South America / Brasil e América do Sul

Metrolog Controles de Medição

Address / Endereço:

Rua Sete de Setembro, 2656
13560-181 - São Carlos - SP
Brazil / Brasil

Phone / Telefone:

+55 (16) 3371-0112
+55 (16) 3372-7800

Internet:

www.metrolog.net
metrolog@metrolog.net

www.metrolog.net