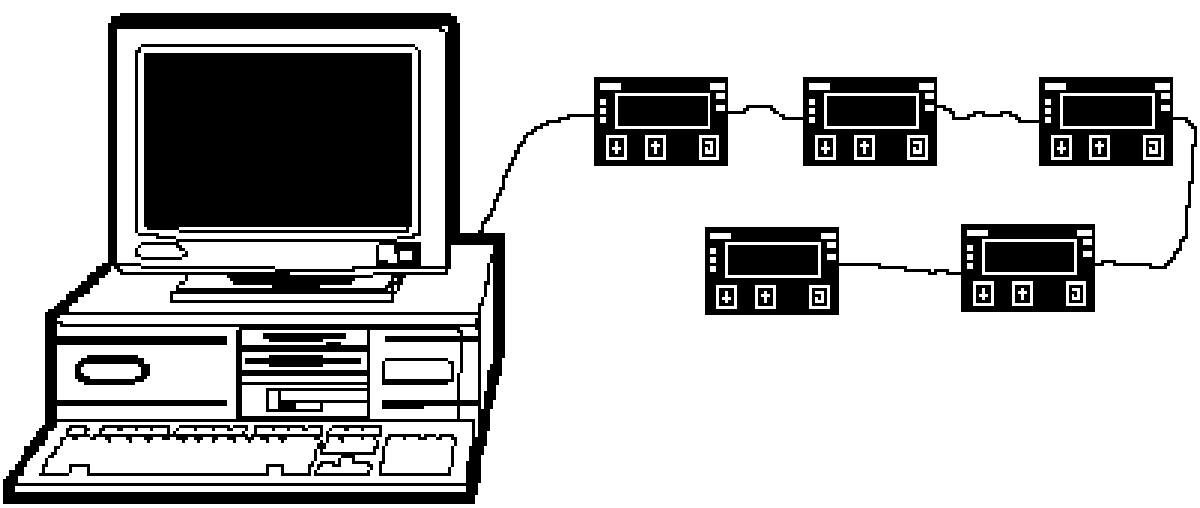
FORM 3769
1st Edition
March,1998

OMEGA DP1610

COMMUNICATION PROTOCOL MANUAL



PREFACE

For high level users who are using a MMI (Man Machine Interface) or SCADA (Supervisory Control And Data Acquisition) program, Section 2 identify all West ASCII communication registers, and in Section 6 identify all MODBUS communication registers which supply all the needed information for interfacing with the instrument's parameters.

For the low level user who will be developing a communication software program, the main portions of this manual will address the message framing and construction, instrument addressing, and all other pertinent details for the applicable protocol. Familiarity with communications and your network is assumed.

Omega Engineering reserves the right to modify our products in line with our policy of continuous product improvement. Information contained in this document is subject to change without notice and does not form any part of any contract between the Omega Engineering and the customer. If you have any suggestions for improvements, or find any errors in this publication, please notify Omega Engineering .

MODBUS is a trademark of Modicon, Inc., Industrial Automation Systems.

Table of Contents

SECTION 1.0 - INTRODUCTION

1.1 RS-485 COMMUNICATION

Communication is at a user selectable rate from the ranges 1200, 2400, 4800 and 9600 baud and the cable used should be suitable for data transfer at this frequency and for the distance required. The transmitters/receivers in the controllers conform to recommendations contained in EIA Standard RS-485.

1.2 AVAILABLE PROTOCOLS

The DP 1610 Digital Indicator has two communication protocols selectable from the Configuration Mode in the instrument:

(1) West ASCII
(2) Modbus RTU

1.3 SELECTION OF COMMUNICATIONS ADDRESS

The communication address for each setpoint programmer is defined in Configuration Mode. The usable address range is 1 to 32 for both protocols.

1.4 CHARACTER TRANSMISSION

Data format for the West ASCII is fixed at seven data bits and one stop bit. The baud rate may be selected to be 1200, 2400, 4800 and 9600 baud. For the West ASCII protocol, the parity is even. For the Modbus RTU protocol, the parity is selectable to be odd, even, or none.

1.5 LINE TURN-ROUND

The communications link is operated as a multi-drop half duplex system. When a device is transmitting, it drives the transmission lines to the appropriate levels; when it is not transmitting, its outputs are set to a high impedance in order that another device can transmit. It is important that a transmitter releases the transmission line before another device starts transmission. This imposes the following restraints on the master device:

(a) The transmitter must release the transmission lines within 6ms of the end of the last character of a message being transmitted. Note that delays due to buffers such as those used in universal asynchronous receivers/transmitters (UARTs) within the master device must be taken into account.

(b) The transmitter must not start transmission until 6mS has elapsed since the reception of the last character of the message.

SECTION 2.0 - WEST ASCII PROTOCOL

The protocol assumes half duplex communications. All communication is initiated by the master device. The master sends a command or query to the addressed slave and the slave replies with an acknowledgment of the command or the reply to the query. All messages, in either direction, comprise:

a) A Start of Message character - R (Programmer Parameters) or L (Controller Parameters)
b) One or two address characters (uniquely defining the slave)
c) A parameter/data string, {DATA}
d) An End of Message character

Messages from the master may be one of five types:

Type 1: {L} {N} ??*

Type 2: {L} {N} {P} {C}*

Type 3: {L} {N} {P} # {DATA}*

Type 4: {L} {N} {P} I *

Where all characters are in ASCII code and:

{L} is the Start of Message character L (Hex 4C).

{N} is the slave DP 1610's address (in the range 1-32); addresses 1-9 may be represented as a single digit (e.g. 7) or in two-digit form, the first digit being zero (e.g. 07).

{P} is a character which identifies the parameter to be interrogated/modified (see Table 2-2)

{C} is the command (see section 2.2 Type 2 Message)

# indicates that (DATA) is to follow (Hex 23)

{DATA} is a string of numeric data in ASCII code (see Table 2-1)

* is the End of Message character (Hex 2A)

No spaces are permitted in messages. Any syntax error in a received message will cause the slave controller to issue no reply and await the Start of Message character.

TABLE 2-1

{DATA} Element	Sign/Decimal Point Position
{DATA} Content	Sign/Decimal Point Position
abcd0	+abcd
abcd1	+abc.d
abcd2	+ab.cd
abcd3	+a.bcd
abcd5	-abcd

TABLE 2-2 COMMAND/PARAMETERS AND IDENTIFIERS

Identifier	Parameter/Command	Type
A	Maximum Process Variable value	Read Only
B	Minimum Process Variable value	Read Only
C	Alarm 1 Value	Read/Write
D	Alarm 1 Hysteresis value	Read/Write
E	Alarm 2 Value[1]	Read/Write
F	Alarm 2 Hysteresis value[1]	Read/Write
G	Scale Range Maximum	Read/Write (linear inputs only)- otherwise Read Only
H	Scale Range Minimum	Read/Write (linear inputs only)- otherwise Read Only
J	Process Variable Offset	Read/Write
L	Instrument Status[2]	Read Only
M	Process Variable Value	Read Only
N	Alarm 3 value[3]	Read/Write
O	Alarm 3 Hysteresis value[3]	Read/Write
Q	Scale Range Decimal Point Position	Read/Write (linear inputs only)- otherwise Read Only
T	Time Elapsed	Read Only
Z	Instrument Commands [4]	Write Only
[	Recorder Output Scale Maximum[5]	Read/Write
\	Recorder Output Scale Minimum[5]	Read/Write
]	Scan Table[6]	Read Only
m	Input Filter Time Constant Value	Read/Write

NOTES ON TABLE 2-2

1. Applicable only if Alarm 2 is configured.
2. See section 2.6.5
3. Applicable only if Alarm 3 is configured
4. See section 2.6.6
5. Applicable only if Output 2 is configured as a Recorder Output.
6. See section 2.7

2.1 TYPE 1 MESSAGE

L {N}??*

This message is used by the master device to determine whether the addressed slave instrument is active. The reply from an active slave instrument is:

L (N)?A*

An inactive instrument will give no reply.

2.2 TYPE 2 MESSAGE

L {N} {P} {C} *

This type of message is used by the master device to interrogate or modify a parameter in the addressed Instrument. {P} is the parameter (as defined in Table 2-2) and {C} represents the command to be executed, which may be one of the following:

+ (Hex 2B)	to increment the value of the parameter defined by {P}
- (Hex 2D)	to decrement the value of the parameter defined by {P}
? (Hex 3F)	to determine the current value of the parameter defined by {P}

The reply from the addressed DP 1610 is of the form:

L {N} {P} {DATA} A *

where {DATA} is five ASCII-coded digits whose format is shown in Table 2-1. The data is the value requested in a query message or the new value of the parameter after modification. If the action requested by the message from the master device would result in an invalid value for that parameter (either because the requested new value would be outside the permitted range for that parameter or because the parameter is not modifiable), the DP 1610 replies with a negative acknowledgment:

L {N} {P} {DATA} N *

The {DATA} string in the negative acknowledgment reply will be indeterminate.

SCAN TABLES

A parameter identifier character] in the message from the master device indicates that a "Scan Table" operation is required. This provides a facility for interrogating the values of a group of parameters and status in a single message from the master device. The reply to such a command would be in the form:

L {N} 25 aaaaa bbbbb ccccc ddddd eeeee A *

For further information, refer to section 2.7

2.3 TYPE 3 MESSAGE

L {N} {P} # {DATA} *

This message type is used by the master to set a parameter to the value specified in {DATA}. The command is not immediately implemented by the slave instrument; the slave will receive this command and will then wait for a Type 4 message (see below). Upon receipt of a Type 3 message, if the {DATA} content and the specified parameter are valid, the slave Instrument replies with:

L {N} {P} {DATA} | *

(where I = Hex 49) indicating that the instrument is ready to implement the command. If the parameter specified is invalid or is not modifiable or if the desired value is outside the permitted range for that parameter, the Instrument replies with a negative acknowledgment:

L {N} {P} {DATA} N *

2.4 TYPE 4 MESSAGE

L {N} {P} | *

This is sent by the master device to the addressed slave Instrument following a successful Type 3 message transaction with the same instrument. Provided that the {DATA} content and the parameter specified in the preceding Type 3 message are still valid, the slave instrument will then set the parameter to the desired value and will reply with:

L {N} {P} {DATA} A *

where {DATA} is the new value of the parameter. If the new value or parameter specified is invalid, the slave Instrument will reply with a negative acknowledgment :

L {N} {P} {DATA} N *

where {DATA} is indeterminate. If the immediately-proceeding message received by the slave instrument was not a Type 3 message, the Type 4 message is ignored.

2.5 INDIVIDUAL PARAMETERS

The individual parameters and how they may be interrogated/modified are described below. Unless otherwise stated, the {DATA} element will follow the standard five-digit format and the decimal position must be correct for the new value to be accepted and for the modification to occur.

NOTE: The communications identifier character {P} for each parameter is shown to the right of each section heading.

2.5.1 PROCESS VARIABLE **{P} = M**

This parameter may be interrogated only, using Type 2 message. If the process variable is out of range, the five-digit {DATA} field in the reply will not contain a number, but will contain **< ?? > 0** (over-range) or **< ?? > 5** (under-range)

2.5.2 PROCESS VARIABLE OFFSET **{P} = J**

This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It modifies the actual process variable value (as measured at the instrument's input terminals) in the following manner:

Modified PV value = Actual PV value + process variable offset value

The modified PV value is limited by the Range Maximum and the Range Minimum and is used for display and alarm purposes and for the recorder outputs.

NOTE: This parameter value should be selected with care. Any adjustment to this parameter is, in effect, an adjustment to the instrument's calibration. Injudicious application of values to this parameter could lead to the displayed PV value having no meaningful relation to the actual PV value.

2.5.3 SCALE RANGE MAXIMUM **{P} = G**

This parameter (which is adjustable on DC linear inputs only) may be interrogated using a Type 2 message or may be modified using a Type 3/4 message sequence. The decimal point position is as for the input range.

2.5.4 SCALE RANGE MINIMUM **{P} = H**

This parameter (which is adjustable on DC linear inputs only) may be interrogated using a Type 2 message or may be modified using a Type 3/4 message sequence. The decimal point position is as for the input range.

2.5.5 SCALE RANGE DECIMAL POINT POSITION **{P} = Q**

This parameter (which is adjustable on DC linear inputs only) may be interrogated using a Type 2 message or may be modified using a Type 3/4 message sequence. The value of this parameter defines the decimal point position, as follows:

Value	**Decimal Point Position**
0	abcd
1	abc.d
2	ab.cd
3	a.bcd

2.5.6 INPUT FILTER TIME CONSTANT **{P} = m**

This parameter (which is adjustable on DC linear inputs only) may be interrogated using a Type 2 message or may be modified using a Type 3/4 message sequence.

2.5.7 RECORDER OUTPUT SCALE MAXIMUM VALUE **{P} = [**

This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It defines the maximum scale value for the instrument's Recorder Output and may be adjusted within the range –1999 to 9999. This value corresponds to the Input Scale Maximum and the decimal point position will always be the same as that for the input.

NOTE: If this parameter is set to a value less than the Recorder Output Minimum Value, the sense of the Recorder output is reversed.

2.5.8 RECORDER OUTPUT SCALE MINIMUM VALUE **{P} = **

This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It defines the minimum scale value for the instrument's Recorder Output and may be adjusted within the range –1999 to 9999. This value corresponds to the Input Scale Minimum and the decimal point position will always be the same as that for the input.

> NOTE: If this parameter is set to a value greater than the Recorder Output Maximum Value, the sense of the Recorder output is reversed.

2.5.9 ALARM 1 VALUE **{P} = C**

This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It defines the level at which Alarm 1 will go active. The decimal point position is as for the input range.

2.6 ALARM 1 HYSTERESIS VALUE **{P} = D**

This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It defines the hysteresis applied to the "safe" side of Alarm 1. The decimal point position is as for the input range.

2.6.1 ALARM 2 VALUE **{P} = E**

This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It defines the level at which Alarm 2 will go active. The decimal point position is as for the input range.

2.6.2 ALARM 2 HYSTERESIS VALUE **{P} = F**

This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It defines the hysteresis applied to the "safe" side of Alarm 2. The decimal point position is as for the input range.

2.6.3 ALARM 3 VALUE **{P} = N**

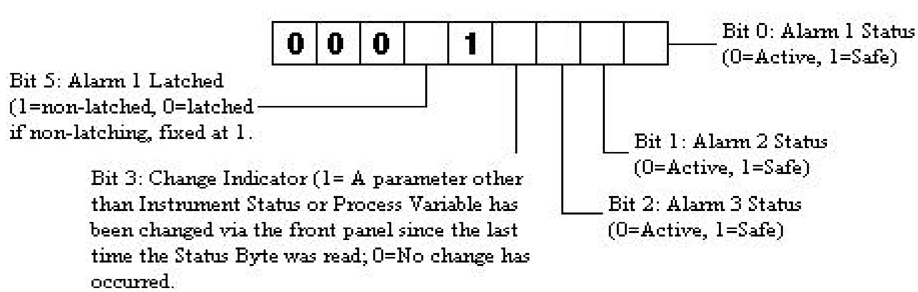
This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It defines the level at which Alarm 3 will go active. The decimal point position is as for the input range.

2.6.4 ALARM 3 HYSTERESIS VALUE **{P} = O**

This parameter may be modified/interrogated, using Type 2 message or a Type 3/4 message sequence. It defines the hysteresis applied to the "safe" side of Alarm 3. The decimal point position is as for the input range.

2.6.5 INSTRUMENT STATUS **{P} = L**

This parameter may be interrogated only, using Type 2 message. The status information is encoded in the four digits as the decimal representation of a binary number. Each bit in the binary number has a particular significance:



2.6.6 INSTRUMENT COMMANDS **{P} = Z**

Only Type 3 or Type 4 messages are allowed with this parameter. In the Type 3 message, the {DATA} field must be one of four five-digit numbers. The reply from the instrument also contains the {DATA} field with the same content. When the master device issues the Type 4 message, the instrument responds with the same {DATA} field content. The commands corresponding to the {DATA} field value are:

00150 = Unlatched Alarm 1 (returns NAK if Alarm 1 is non-latching)
00160 = Reset Process Variable Maximum (to current PV value)
00170 = Reset Process Variable Minimum (to current PV value)
00180 = Reset Time Elapsed (to zero)

2.7 SCAN TABLE **{P} =]**

The Scan Table operation takes the form of a Type 2 interrogation command which accesses a set of information (held in the {DATA} element in the response). The response would be in the form:

L{N}] 25 aaaaa bbbbb ccccc ddddd eeeee A *

These digits are described in Table 2-1 and may comprise:

aaaaa	The current process variable value
bbbbb	The current maximum process variable value
ccccc	The current minimum variable value
ddddd	The current Time Elapsed value
eeeee	The Instrument Status (see section 2.6.5).

2.8 ERROR RESPONSE

The circumstance under which a message received from the master device is ignored are:

Parity error detected
Syntax error detected
Time-out elapsed
Receipt of a Type 4 message without a proceeding Type 3 command message.

Negative acknowledgments will be returned if, in spite of the received message being notationally correct, the DP 1610 can not supply the requested information or perform the requested operation. The {DATA}element of a negative acknowledgment will be indeterminate.

SECTION 3.0 - MODBUS RTU PROTOCOL

The remainder of this document specifies the MODBUS communications protocol as implemented on the applicable Modbus instruments. Source material for this manual is the Modicon MODBUS Protocol Reference Guide PI-MBUS-300, Rev. F.

For the high level user who is using a MMI (Man Machine Interface) program, all you will need to use is the Section 6 to identify all the instruments register assignments and data types.

For the low level user developing a communication software program, this manual will address the message framing and construction, instrument addressing, and all other pertinent information.

This manual does not try to be a complete guide to the MODBUS protocol, but will show how to structure a message that the Modbus instruments will recognize, how to request access to another device, and how errors will be detected and reported.

3.1 MODBUS MESSAGE FRAMING

There are two serial transmission modes for the MODBUS protocol, ASCII or RTU (Remote Transmission Unit) framing. The Modbus instruments use the **RTU** framing method **only**. The MODBUS message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows receiving devices to begin at the start of the message, read the address portion and determine which device is addressed, and know when the message is completed. Partial messages can be detected and errors can be set as a result.

3.2 RTU FRAMING

In RTU framing mode, the message starts with a silent interval of at least 3.5 character times. This is most easily implemented as a multiple of character times at the baud rate that is going to be used in the network. The first field then transmitted is the device address. Binary data is used for transmission for all fields. Network devices monitor the network bus continuously, including the “silent” interval. When the first field is received, each device decodes it to find out if it is being addressed.

Following the last transmitted character, a similar ‘silent’ interval of at least 3.5 character times marks the end of the message. A new message can then begin after this interval. The entire message frame must be transmitted as a continuous stream. If a ‘silent’ interval of more than 1.5 character times occurs before completion of the frame, the receiving device discards the incomplete message and assumes the next byte will be the address field of a new message. If a new message begins earlier than 3.5 character times following a previous message, the receiving device will consider it a continuation of the previous message. This will cause an error, as the value in the final CRC field will not be valid for the combined messages. A typical message frame is shown Figure 1 below.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	n X 8 BITS	16 BITS	T1-T2-T3-T4

Figure 1 : Typical message frame

Note: All bytes are a total of 11 bits long, which include 1 start bit, 8 data bits, 1 parity bit (if used), and 1 or 2 stop bits (depends if a parity bit is used). See Bits per Byte on next page.

3.3 RTU MODE

The main advantage of the RTU mode is that its greater character density allows better data throughput than ASCII for the same baud rate. Each message must be transmitted in a continuous stream. The format for each byte in RTU mode is:

Coding System: 8-bit binary

Bits per Byte: 1 start bit
8 data bits, least significant bit sent first
1 bit for even/odd parity; no bit for no parity
1 stop bit if parity is used; 2 bits if no parity

Baud Rate: 1200, 2400, 4800, and 9600

Error Check Field: Cyclical Redundancy Check (CRC)

3.4 CONTENTS OF THE DATA FIELD

The data field of messages sent from a master to a slave device contains additional information which the slave must use to take the action defined by the function code. This can include items like register addresses, the quantity of items to be handled, and the count of actual data bytes in the field.

For example, if the master requests a slave to read a group of holding registers (function code 03), the data field specifies the starting register and how many registers are to be read. If the master writes to a group of registers in the slave (function code 16), the data field specifies the starting register, how many registers to write, the count of data bytes to follow in the data field, and the data to be written in the registers. If no error occurs, the data field of a response from a slave to a master contains the data requested. If an error occurs, the field contains an exception code that the master application can use to determine the next action to be taken.

3.5 MODBUS FUNCTION SUPPORTED

The following is a list the functions to be supported. The JBUS names are listed first, where such an equivalence exists, as these more closely represent the actual operations. The original Gould MODBUS function names are listed for reference. The MODBUS Function number follows the names. In some cases two function numbers will be supported, as they could be used interchangeably with respect to our unit. Please refer to Figure 2 below for the MODBUS functions that are supported.

	JBUS	MODBUS	FUNCTION NUMBER
A	Read n Bits	Read Coil Status Read Input Status	01 02
B	Read n Words	Read Holding Registers Read Input Registers	03 04
C	Write 1 Bit	Force Single Coil	05
D	Write 1 Word	Preset Single Register	06
E	Loopback Test	Loopback Diagnostic Test	08
F	Write n Words	Preset Multiple Registers	16

Figure 2 : MODBUS Functions that are supported

The instrument will identify itself in reply to a Read Holding Register message which inquires the value of parameter 121 & 122, as specified in the CNOMO documentation, and the Modbus Function 17 (Report Slave ID) will not be supported.

3.6 MESSAGE FORMATS

The first character of each message is an instrument address. The valid range of such an address is 0 to 247. The second character is always the function number. The contents of the remainder of the message depends upon the function number. The maximum length of a message is 256 bytes.

In most cases the instrument is required to reply by echoing the address and function number, together with an echo of all or part of the message received (in the case of a request to write a value or carry out a command) or the information requested (in the case of a read parameter operation).

A message is terminated solely by a delay of at least 3.5 character lengths at the given baud rate, and any character received after such a delay is treated as a potential address at the start of a new message.

Since only the RTU form of the protocol is being supported, each message contains binary data values and is followed by a two-byte CRC16. This is a 16-bit cyclic redundancy checksum. There are two ways to get the CRC values. One is calculated in accordance with a formula which involves recursive division of the data by a polynomial, with the input to each division
being the remainder of the results of the previous one. The other is for you to use a CRC16 look-up table that is referenced in the Modicon Modbus Protocol Reference Guide. More information on how the checksum must be calculated can be found in Modicon MODBUS Protocol Reference Guide PI-MBUS-300, Rev. F.

The following abbreviations or designations are used in specifying the message formats:

IA	Instrument Address
FN	Function Number
PCH	Parameter Code High
PCL	Parameter Code Low
NBH	Number of Bits High
NBL	Number of Bits Low
NWH	Number of Words High
NWL	Number of Words Low
COUNT	Number of bytes of data
CRCH	CRC16 - Cyclic Redundancy Check Value High
CRCL	CRC16 - Cyclic Redundancy Check Value Low

In depicting message formats, parenthesis are used to represent a byte of data. The general format of the message sent to the instrument will consist of 8 or more bytes, consisting of a "standard preamble" (IA and FN bytes), data (dependent on the function), and a CRC16 as follows:

(IA) (FN) (data) (data) (data) (data) . . . . . . (CRCL) (CRCH)

The MODBUS documentation refers to a Starting Address as the pointer to Coil or Input Bits, or Holding or Input Registers. They will be referred to as Parameter Codes in this document.

3.7 READ N BITS - FUNCTION NUMBER 01/02

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by the parameter code of the first data bit to be read, and the two-byte bit count to be read, as follows:

(IA) (1 or 2) (PCH) (PCL) (NBH) (NBL) (CRCL) (CRCH)

The normal reply will echo the first 2 characters of the message received, and will then contain a single-byte data byte count, which will not include itself or the CRC. For this message, there will be one byte of data per eight bits-worth of information requested, with the LSB of the first data byte transmitted depicting the state of the lowest-numbered bit parameter required.

(IA) (FN) (COUNT) (8 bits) (8 bits) (etc.) . . . (last 8 bits) (CRCL) (CRCH)

This function will be used largely to report instrument status information, such as local/remote or auto/manual status, and so a bit set to 1 indicates that the corresponding feature is currently enabled/active/true, and a bit reset to 0 indicates the opposite.

If an exact multiple of eight bits is not requested, the data is padded with trailing zeros to preserve the 8-bit format.

3.8 READ N WORDS - FUNCTION NUMBER 03/04

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by the parameter code of the first parameter to be read, and the two-byte word count to be read, as follows:

(IA) (3 or 4) (PCH) (PCL) (NWH) (NWL) (CRCL) (CRCH)

The reply sent by the instrument echoes the first two characters received and then contains a single-byte data byte count, the value of which does not include either itself or the CRC value to be sent. For this message, the count equals the number of words read times two. Following the byte count, the number of bytes are transmitted, MSB first, followed by the CRC16.

(IA) (FN) (COUNT) (HI) (LO) (HI) (LO) (etc.) (etc.). . . (last HI) (last LO) (CRCL) (CRCH)

3.9 WRITE 1 BIT - FUNCTION NUMBER 05

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by the parameter code of the bit to set and a two-byte word whose MSB contains the desired truth value of the bit, expressed as 0xFF (TRUE) or 0x00 (FALSE), as follows:

(IA) (5) (PCH) (PCL) (State) (0) (CRCL) (CRCH)

Generally, this function will be used to control such features as auto/manual. The normal reply sent by the instrument will be a byte-for-byte echo of the message received.

3.10 WRITE 1 WORD - FUNCTION NUMBER 06

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by the parameter code of the parameter to be written, and the two-byte value to which the parameter will be set, as follows:

(IA) (6) (PCH) (PCL) (Value HI) (Value LO) (CRCL) (CRCH)

The normal response is to echo the message in its entirety.

3.11 WRITE N WORDS - FUNCTION NUMBER 16

The message sent to the instrument will consist of 11 or more bytes consisting of the standard preamble, followed by the parameter code of the first parameter to be written, a two-byte word count, a one-byte byte COUNT, and the series of two-byte words to which the parameters will be set. *** See NOTE at Section 4.3 Multiple Parameter Value Communications.**

(IA) (16) (PCH) (PCL) (NWH) (NWL) (COUNT) (Word HI) (Word LO)
(Word HI) (Word LO) . . . . . (Word HI) (Word LO) (CRCL) (CRCH)

The instrument normally responds with a 8 byte reply, as follows:

(IA) (16) (PCH) (PCL) (NWH) (NWL) (CRCL) (CRCH)

3.12 LOOPBACK DIAGNOSTIC TEST - FUNCTION NUMBER 08

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by a four bytes of zeros, as follows:

(IA) (8) (0) (0) (0) (0) (CRCL) (CRCH)

Full MODBUS support in this area is not appropriate - consequently, the diagnostic code supported is code 0000. In response to the message the instrument must echo the message received exactly.

3.13 ERROR AND EXCEPTION RESPONSES

If the instrument receives a message which contains a corrupted character (parity check fail, framing error. etc.), or if the CRC16 check fails, the instrument ignores the message. If the message is otherwise syntactically flawed (e.g. the byte count or word count is incorrect) the instrument will also not reply.

If the instrument receives a syntactically correct message which contains an illegal value, it will send an exception response, consisting of five bytes as follows:

(IA) (FN) (Exception Number) (CRCL) (CRCH)

The Function Number field consists of the function number contained in the message which caused the error, with its most significant bit set (i.e. function 6 becomes x86), and the Exception Number is one of the codes contained in the following table:

Code	Name	Cause
1	ILLEGAL FUNCTION	Function number out of range
2	ILLEGAL DATA ADDRESS	Parameter ID out of range or not supported
3	ILLEGAL DATA VALUE	Attempt to write invalid data or not enough data words - action not carried out
4	DEVICE FAILURE	N/A
5	ACKNOWLEDGE	N/A
6	BUSY	N/A
7	NEGATIVE ACKNOWLEDGE	N/A

SECTION 4.0 - PARAMETER VALUE TYPES

There are three different types of parameter values, Bit parameter, Single parameter, and Multiple parameter values.

4.1 BIT PARAMETER COMMUNICATIONS

The bit parameters are fairly straight forward. The bits are accessed and transmitted per the protocol while they are stored in the instrument as one-byte entities.

The computer can read single or multiple bits using Function Numbers 01 or 02. The computer can write a single bit using Function Number 05. We do **not** support multiple bit writes via Function Number 15, as this is seldom done in real world applications and MMI or SCADA programs don't always support it either.

4.2 SINGLE PARAMETER VALUE COMMUNICATIONS

The two-byte integer values are straight forward. Values are stored, accessed, transmitted and treated simply as two-byte or word oriented entities.

4.3 MULTIPLE PARAMETER VALUE COMMUNICATIONS

In addition to being able to read or write any one of the data types in a single operation, Function Numbers 03/04 or 16 can be used to read or write a series of consecutive registers that correspond to any mix of data types, provided the number of consecutive registers falls on a parameter boundary. Otherwise, the instrument will not accept the command and will respond with an Exception Code of 3, ILLEGAL DATA VALUE. To simplify coding, the instruments will process data in order of occurrence, and only not accept/process the command for the last parameter if it does not fall on the end of the parameter boundary.

***NOTE:** Support for multi-parameter writes is limited to support of the Multi-word Write Function (Number 16), but will permit writing of one parameter only per message. The multi parameter read function supports a maximum of 10 parameters in one message.

SECTION 5.0 – EXAMPLES

Read Example #1 :

Read the process value from the instrument. Please note: you would use Modbus Function 3 (Read n Words). Abbreviated in the example as **[FN].**

The message sent to "**read from"** the instrument's address 2 **[IA**] would be as follows:

			[address of 1st word]		**[number of words]**		**[CRC16]**	
	(IA)	(FN)	(High byte)	(Low byte)	(NBH)	(NBL)	(CRCL)	(CRCH)
Hex out:	(02)	(03)	(00)	(01)	(00)	(01)	(D5)	(F9)

The response if the process value is "79" :

	(IA)	(FN)	(count)	(1st Value)	(Last Value)	(CRCL)	(CRCH)
Hex in:	(02)	(03)	(02)	(00)	(4F)	(BD)	(BO)

Read Example #2 :

Read the setpoint value **(Word parameter 2)** from the instrument. Please note: you would use Modbus Function 3 (Read n Words).

The message sent to "**read from"** the instrument's address 2 **[IA**] would be as follows:

			[address of 1st word]		**[number of words]**		**[CRC16]**	
	(IA)	(FN)	(High byte)	(Low byte)	(NBH)	(NBL)	(CRCL)	(CRCH)
Hex out:	(02)	(03)	(00)	(02)	(00)	(01)	(25)	(F9)

The response if the setpoint value is "200" :

	(IA)	(FN)	(count)	(1st Value)	(Last Value)	(CRCL)	(CRCH)
Hex in:	(02)	(03)	(02)	(00)	(C8)	(FD)	(D2)

Write Example #1:

The message sent to **write** "450" to the instrument's **setpoint (Word parameter 2)** would be as follows:

			[address of 1st word]		**[value of 450]**		**[CRC16]**	
	(IA)	(FN)	(High byte)	(Low byte)	(Value Hi)	(Value Lo)	(CRCL)	(CRCH)
Hex out:	(02)	(06)	(00)	(02)	(01)	(C2)	(A8)	(38)

The response if the setpoint value is changed to "450" and accepted is:

	(IA)	(FN)	(High byte)	(Low byte)	(NBH)	(NBL)	(CRCL)	(CRCH)
Hex in:	(02)	(06)	(00)	(02)	(01)	(C2)	(A8)	(38)

The instrument should respond by echoing the message. If the instrument did not accept the new setpoint value, say because of a setpoint limit or out range the response would be:

	(IA)	(FN)*	(Exception #)	(CRCL)	(CRCH)
Hex in:	(02)	(86)	(03)	(F2)	(61)

***Note: See Error and Exception Response Section 3.13.**

SECTION 6.0 - MODBUS REGISTERS

SECTION 6.1 - BIT PARAMETERS

Parameter	No.	Notes
Alarm 1 Status	1	R/O 1=Active
Alarm 2 Status	2	R/O 1=Active
Alarm 3 Status	3	R/O 1=Active
Alarm 1 Latched	4	R/O 1=Alarm 1 latched*
PV Under-range Flag	5	R/O 1=Active
PV Over-range Flag	6	R/O 1=Active
Sensor Break Active	7	R/O 1=Active
Reset Latched Alarm	8	Write Only
Reset PV Maximum	9	Write Only
Reset PV Minimum	10	Write Only
Reset Time Elapsed	11	Write Only

*Always returns 0 if Alarm 1 not configured to be latching

SECTION 6.2 - WORD PARAMETERS

Parameter	No.	Notes
Process Variable	1	R/O
PV Maximum	2	R/O
PV Minimum	3	R/O
Time Elapsed	4	R/O
Instrument Status	5	R/O
PV Offset	6	Read/Write
Alarm 1 Value	7	Read/Write
Alarm 2 Value	8	Read/Write Only if Alarm 2 is configured
Alarm 3 Value	9	Read/Write Only if Alarm 3 is configured
Alarm 1 Hysteresis	10	Read/Write
Alarm 2 Hysteresis	11	Read/Write Only if Alarm 2 is configured
Alarm 3 Hysteresis	12	Read/Write Only if Alarm 3 is configured
Filter Time Constant	13	Read/Write
Decimal Point Position	14	Read/Write Read Only for non-linear inputs
Scale Range Minimum	15	Read/Write Read Only for non-linear inputs
Scale Range Maximum	16	Read/Write Read Only for non-linear inputs
Recorder Output Scale Maximum	17	Read/Write Only if Recorder output is configured
Recorder Output Scale Minimum	18	Read/Write Only if Recorder output is configured
Manufacturer ID	121	R/O
Equipment ID	122	R/O Number 1810

***NOTE**: When the PV is over-ranged or under-ranged or when a sensor break condition occurs, the value returned is:

Condition	Hex	Signed	Unsigned
Over-range	F700	-2304	63232
Under-range	F600	-2560	62976
Sensor Break Condition	F800	-2048	63488

The PV Maximum parameter will return the Over-range condition or Sensor Break value (as appropriate) if either condition has occurred since the PV Maximum parameter was last reset.

The PV Minimum parameter will return the Under-range condition or Sensor Break value (as appropriate) if either condition has occurred since the PV Minimum parameter was last reset.

The Time Elapsed parameter will return the over-ranged value if the time exceeds 1000 minutes.

All these parameters return signed values except Time Elapsed (which is unsigned) and Instrument Status (in which Bits 0 – 6 of the status byte return Bit Parameters 1 – 7 respectively – see Bit Parameters Section 6.1)