Smith Meter® AccuLoad® IV

Modbus Communications Manual

Bulletin MN06202L Issue/Rev. 0.0 (9/23)

Important

All information and technical specifications in this document have been carefully checked and compiled by the author; however, we cannot completely exclude the possibility of errors. TechnipFMC is always grateful to be informed of any errors; contact us at TechnipFMC.com.

Caution

The default or operating values used in this document and in the configuration parameters of the AccuLoad IV are for factory testing only and should not be construed as default or operating values for your metering system. Each metering system is unique and each configuration parameter must be reviewed and programmed for that specific metering system application.

Disclaimer

TechnipFMC hereby disclaims all responsibility for damages, including but not included to consequential damages arising out of or related to the inputting of incorrect or improper program or default values entered in connection with the AccuLoad IV.

Technical Support

Field Service Response Center
24/7 Technical Support/Schedule a Technician: +1 844.798.3819

System installation supervision, startup, and commissioning services are available.

Customer Support

Customer Service
TechnipFMC
Measurement Solutions
1602 Wagner Avenue
Erie, PA 16510 USA
+1 814.898.5000
MS.ResponseCenter@TechnipFMC.com
TechnipFMC.com

Literature Library
info.smithmeter.com/literature

Contents

1 Introduction and Overview

The Modbus protocol was developed by Modicon, Inc. to be a concise method of transferring data to and from programmable logic controllers (PLCs). It has become a default standard in many areas of industrial automation where supervisory control or remote data collection is required. In a Modbus system, a host (client) communicates with one or multiple field devices (servers). The AccuLoad IV acts as a server device only; an external host must act as the client to query or control the AccuLoad IV. Each configured arm in the AccuLoad IV must have a unique communication address in the range of 1 to 99. Host messages to address 0 (the Modbus broadcast address) are not currently supported (are ignored) by the AccuLoad IV. For more information regarding Modbus communications specifics, refer to the section Appendix A: Modbus Communications Primer on page 1.

1.1 Number Conventions in this Manual

Throughout this document, numeric values, such as addresses, register values, or data arguments intended to be interpreted as hexadecimal are preceded by "0x". If "0x" is not present, the value should be interpreted as a decimal number.

1.2 Modbus Register Range

The AccuLoad IV uses the full register range allowed by the Modbus specification, 0 through 65535. Some supervisory computer Modbus driver packages artificially limit the register range; these host drivers are not recommended for use with the AccuLoad.

Modbus register numbering is sometimes a source of confusion, as the register identifications (IDs) used by some clients start at 1, while the actual Modbus register addresses in a Modbus server start at 0, so the register values sent on the wire start at 0. Therefore, some client software subtracts 1 from the register field prior to transmitting the Modbus command. This manual assumes a zero-based addressing scheme, so for those Modbus clients that pre-decrement the specified register, you must add 1 to the register number given in the AccuLoad Operator Reference Manual (MN06200) to get the desired value.

1.3 Floating Point Endian Control

Floating-point numbers are not defined in the Modbus specification; there are nearly as many variations of how it is supported as there are vendors. Most often, Modbus registers are combined sequentially to make up an IEEE single-precision or double-precision floating point number; this is the case in the AccuLoad IV. Two registers are needed for single-precision, and four registers are needed for double-precision numbers. However, there are several ways to map floating point values to Modbus registers. To ensure compatibility with off-the-shelf drivers, three popular variations of byte ordering for floating point numbers are supported. See system program code 732.

The AccuLoad IV returns a single-precision representation for pi (3.14159) using function 3 registers 2106 and 2107. Registers 2108 through 2111 return a double-precision representation of pi. These registers are useful for configuring compatible byte ordering for Modbus host drivers. Program the host to display these registers as appropriate floats, and change program code 732 until pi is properly interpreted by the host.

1.4 Changing Program Mode Parameters

The AccuLoad IV limits access to program mode parameters to ensure a valid configuration is present before loading is allowed. Only one source (COM port or keypad) is granted access at any one time. Also, additional requirements must be met to modify program mode parameters. Using Modbus, the following procedure must be used:

1. Clear the Program Mode Result field (function 6, register 2050, data 0).
2. Make any program code changes by writing the new value to the appropriate location using function 6 or 16. If successful, the first write will set the Program Mode state for this port to 1. Some data, for example text strings, encompass multiple registers. These registers should be written together or in ascending order.

   > All registers for the data must be updated to effect a change.

3. Issue the Program Mode Logout - Save Changes command (function 6, register 2048, data 1). Alternatively, discard the changes made since the last logout by writing data = 2.

4. Repeatedly read the Program Mode state (function 3, register 2049) until the return value is 0. Possible values are:
   - 0—Not in Program Mode using this port
   - 1—In Program Mode using this port
   - 2—Checking for configuration errors. When complete (state not 2), the AccuLoad IV will automatically reset the state to 0. This register reflects the Program Mode state on this port only.
5. Read the Program Mode result (function 3, register 2050) to determine success or failure.
   - 0—Program Mode exited normally
   - 1—Preempted by the keypad (any pending changes made using COM port are lost)
   - 2—Critical configuration errors exist that must be corrected before loading can commence (changes are saved)
   - 3—This port was reset externally (any pending changes are lost)

1.5 AccuLoad IV Modbus Map—Overview

Table 1: AccuLoad IV Modbus Map

Function 1 Read Coil **Function 5** Force Single Coil **Function 15** Force Multiple Coils	**Function 2** Read Input Status	**Function 3** Read Holding Register **Function 6** Preset Holding Register **Function 16** Preset Multiple Regs	**Function 4** Read Input Registers
Coils 43-120 Set General-Purpose Digital Output States	Coils 0-120 Read Digital I/O States	Register 2112 Set User Alarms	Registers 2048-22351 Registers 29696-29890 Read Run Data Values
Coils 128-2703 Reset Alarm Indicators System, Arm, Meter, Product, Injector	Coils 128-2703 Read Alarm Indicators		Registers 3008-3023 Read User Timers
	Coils 4096-4230 Read Status Flags	Registers 2496-2511 Write User Timers	Registers 22528-28883 Read Recipe Totalizers and Recipe Calculated Program Mode Values
		Registers 2560-2915 Read/Write User Variables	
		Registers 2944-22721 Registers 22912-23103 Registers 23424-25593 Read/Write Program Mode Parameter Values	
		Registers 22784-22795 Write to General-Purpose Analog Output Channels	
		Registers 23296-23423 Set User Text	
Coil 4096 **Issue Initiate Extended Service Command**		**Registers 0-1023** **Write to Extended Services Input Buffer Area**	**Registers 0-2047** **Read Extended Services Output Buffer Area**

> For Registers 2944-22656, refer to section 1.4: Changing Program Mode Parameters on page 2 for details.

> For Coil 4096, Registers 0-1023, and Registers 0-2047, refer to Extended Services on page 10 for an in-depth explanation on accessing advanced features, such as prompting and transaction control using these areas.

2 AccuLoad IV Modbus Examples

2.1 Examples of AccuLoad IV Modbus Commands

2.1.1 Example 1—Clear User Alarm 2 Using Function 5—Force Single Coil

The alarm clearing action is mapped to Function 1, 5, and 15. A write is required to clear an alarm, therefore Function 5 (force coil) is used. The table below shows that this alarm's reset point is located at Coil 144 (0x0090).

> To force a single coil on using Function 5, Modbus requires the specific data value of 65280 (0xFF00) be written.

Table 2: Example 1—Force Coil Message

Force Coil Message							
Address 0x01	Function 0x05	Coil (MSB) 0x00	Coil (LSB) 0x90	Data (MSB) 0xFF	Data (LSB) 0x00	CRC (LSB) 8C	CRC (MSB) 17

Table 3: Example 1—Response to Force Coil Message

Response to Force Coil Message							
Address 0x01	Function 0x05	Coil (MSB) 0x00	Coil (LSB) 0x90	Data (MSB) 0xFF	Data (LSB) 0x00	CRC (LSB) 8C	CRC (MSB) 17

2.1.2 Example 2—Set Digital Outputs 1, 6, and 9 using Function 15—Force Multiple Coils

Since Modbus Function 15 allows forcing of multiple coils, it can be used to set multiple general purpose digital outputs. Modbus treats the data sent with a Force Multiple Coils Command as a series of bit-mapped coil states. This demonstrates one of the strengths of Modbus: A considerable amount of functionality can be packed into a very concise command. Each byte in the data area of the command represents the state of eight coils. The first byte's low-order bit represents Digital Output 1, and the first byte’s high order bit represents Digital Output 8.

Starting Coil Number: 43
Number of Coils: 16
Data: 0x2101 (bit-packed data, binary 00100001 and 00000001, corresponding to digital inputs 1, 6, and 9)

Table 4: Example 2—Force Multiple Coil Message

Force Multiple Coil Message										
Address 0x01	Function 0x0F	Starting Coil (MSB) 0x00	Starting Coil (LSB) 0x2B	# Coils Written (MSB) 00	# Coils Written (LSB) 10	# of Data Bytes 02	Data Byte 1 20	Data Byte 2 21	CRC (LSB) 85	CRC (MSB) 98

Table 5: Example 2—Response to Force Multiple Coils Message

Response to Force Multiple Coils Message							
Address 0x01	Function 0x0F	Starting Coil (MSB) 0x00	Starting Coil (LSB) 0x2B	# Coils Written (MSB)	# Coils Written (LSB)	CRC (LSB) 0x24	CRC (MSB) 0x0F

2.1.3 Example 3

Interrogation Message: Read the meter K factor (program code "arm 1, meter 1, parameter 301")
AccuLoad Address: 01
Function Code: 03 (Read Holding Registers)
Beginning Register Number: 5698
Number of Registers: 2

Table 6: Example 3—Read Holding Registers, Interrogation Message

Read Holding Registers							
Interrogation Message							
Address 0x01	Function Code 0x03	Beginning Register (MSB) 0x16	Beginning Register (LSB) 0x42	Number of Req. Regs (MSB) 0x00	Number of Req. Regs (LSB) 0x02	CRC16 (LSB) 0x60	CRC16 (MSB) 0x57

Response Message: K factor = 100.0

> Note: AccuLoad host communications set to "Little 16" endian to match Intel PC.

AccuLoad Address: 01
Function Code: 03 (Read Holding Registers)
Byte Count: 04

Table 7: Example 3—Read Holding Registers, Response Message

Read Holding Registers								
Response Message								
Address 0x01	Function Code 0x03	Byte Count 0x04	MSB of the First Data Reg. 0x00	LSB of the First Data Reg. 0x00	MSB of the Second Data Reg. 0x42	LSB of the Second Data Reg. 0xC8	CRC16 (LSB) 0xCB	CRC16 (MSB) 0x05

2.1.4 Example 4

Interrogation Message: Write the value 1 to Boolean/Algebraic Boolean User Variable #1
AccuLoad Address: 01
Function Code: 06 (Write a Holding Register)
Beginning Register Number: 2816
Number of Registers: 1

Table 8: Example 4—Write Holding Register, Interrogation Message

Write Holding Register							
Interrogation Message							
Address 0x01	Function Code 0x06	Register (MSB) 0x0B	Register (LSB) 0x00	Data (MSB) 0x00	Data (LSB) 0x01	CRC16 (LSB) 0x4A	CRC16 (MSB) 0x2E

Response Message:
AccuLoad Address: 01
Function Code: 06 (Write a Holding Register)
Beginning Register Number: 2816
Number of Registers: 1

Table 9: Example 4—Write Holding Register, Response Message

Write Holding Register							
Response Message							
Address 0x01	Function Code 0x06	Beginning Register (MSB) 0x0B	Beginning Register (LSB) 0x00	Data Req. Regs (MSB) 0x00	Data Req. Regs (LSB) 0x01	CRC16 (LSB) 0x4A	CRC16 (MSB) 0x2E

2.1.5 Example 5

Interrogation Message: Write the value 10.0 to Boolean/Algebraic User Float Variable #1 (“Little 16” endian)
AccuLoad Address: 01
Function Code: 16 (Write Multiple Registers)
Beginning Register Number: 2560
Number of Registers: 2

Table 10: Example 5—Write Holding Register (Multiple), Interrogation Message

Write Holding Register (Multiple)												
Interrogation Message												
Address 0x01	Function Code 0x10	Beginning Reg MSB 0x0A	Beginning Reg LSB 0x00	No. Regs MSB 0x00	No. Regs LSB 0x02	Byte Count 0x04	Data Reg 1 MSB 0x00	Data Reg 1 LSB 0x00	Data Reg 2 MSB 0x41	Data Reg 2 LSB 0x20	CRC 16 LSB 0x6C	CRC 16 MSB 0x87

Response Message:

AccuLoad Address: 01
Function Code: 16

Write Holding Register (Multiple)							
Response Message							
Address 0x01	Function Code 0x10	Beginning Reg MSB 0x0A	Beginning Reg LSB 0x00	No. Regs MSB 0x00	No. Regs LSB 0x02	CRC16 LSB 0x42	CRC16 MSB 0x10

2.2 Function 08—Diagnostics (Loopback Diagnostics)

Diagnostic Subfunction	Purpose
00	Loops the received query back out of the port

Function	Address Range	Note
6, 16	All long integers	All elements must be written in order (lo to hi Modbus address); the target value will be changed upon writing the last element.
6, 16	All floating point (single)	All elements must be written in order (lo to hi Modbus address); the target value will be changed upon writing the last element.
6, 16	All floating point (double)	All elements must be written in order (lo to hi Modbus address); the target value will be changed upon writing the last element.
6, 16	All text strings (ASCII characters)	All elements must be written in order (lo to hi Modbus address); the target value will be changed upon writing the last element.
All	All	Any "offset specification" (product id, etc.) are "zero" based (0=1, 1=2, 2=3, etc.) except where noted.

3 AccuLoad IV Modbus Register Map

Due to the large amount of information in this section, it is unwieldy when printed; therefore, the complete register map is no longer included in this document. Instead, a spreadsheet containing the specific register mapping for various AccuLoad IV data points can be obtained by request through our sales and support organization. Contact our Response Center or your account sales team to obtain a copy.

4 Extended Services

4.1 Accessing Transaction Control and Other Features Using Modbus

The Extended Services feature of the AccuLoad IV allows access to functions such as transaction control, display control, and stored event/transaction data retrieval. The AccuLoad IV Modbus protocol supports the Extended Services features completely. The requirements to interact with the services are described below.

To use an extended service using Modbus, the following steps are required:

1. Using Function 3, write the desired command values and any additional data required as specified for that command (the Extended Services packet) into the Extended Services command buffer that starts at holding register 1 (Functions 3, 6, and 16, Register 1).
2. Write the total length of the above packet in bytes into holding Register 0.
3. To invoke the service, write a Force Coil 4096 ON Command (Function 5 or 15, Address 4096, data "on").
4. Retrieve the result of the service by reading the result packet in the extended service outbound buffer at the start of the input register area (Function 4, Register 1). The number of bytes in the response packet is located at Register 0.

The following tables show the mapping of the Extended Services packets into Modbus register space, and controlling packet submission for processing:

Table 11: Modbus Register Map 1—Functions 3, 6, and 16—Host Command Write Packet Format

Modbus Holding Register	0	1	2	3	…	513
Data Type	Unsigned integer	Varied—Depends on packet				
Content	Number of valid bytes in packet	<packet> byte 0, byte 1	<packet> byte 2, byte 3	<packet> byte 4, byte 5	<packet>	<packet> byte 1022, byte 1023

Table 12: Modbus Register Map 2—Function 4—Host Command Write Packet Format

Modbus Input Register	0	1	2	3	…	513
Data Type	Unsigned Integer	Varied—Depends on packet				
Content	Number of valid bytes in packet	<packet> byte 0, byte 1	<packet> byte 2, byte 3	<packet> byte 4, byte 5	<packet>	<packet> byte 1022, byte 1023

Table 13: Modbus Register Map 3—Function 1, 5, and 15—Host Command Write Packet Format

Modbus Coil	4096
Content	Writing an "on" to this coil causes the AccuLoad to submit the packet located in the holding register area for processing. The Modbus input register area (Function 4) then holds the response packet.

Each packet has the following structure:

Table 14: Packet Data

Packet Data	
Router Info—16 -bit integer	Service Specific Data (any length up to 1022 bytes)

The “Services Router” (OSI network layer) examines the received packet and routes it to the specific service handler specified. "Router Info" is a 16-bit control word which primarily indicates what application (service) receives the data. The "service specific data" is passed to the applications routine, and its form may vary from service to service. The router info word is broken down as follows:

Table 15: Router Info Word (16 bit)

Router Info Word (16 bit)															
First Byte								Second Byte							
bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
C	U	R	R	S	S	S	S	S	S	S	S	S	S	S	S

C — Command response flag. 0=Command (host to AccuLoad), 1=Response (AccuLoad to host).

U — Unused; reserved for future use. Always set this to zero.

R — Router status. Command packets should always set this to binary 00 (normal packet). Responses may have this set to binary 00 (normal response), binary 01 (service specified doesn't exist) or binary 10 (router error). Any response other than binary 00 means that no service specific data follows. Binary 11 is reserved for future use.

S — 12-bit number referring to the specific service (application layer) routine (see the table below).

The 12-bit service numbers are typically unique among different models that support Extended Services. A section of service codes (0x000-0x0FF) is reserved for services that must be identical among all instruments supporting extended services. Service codes 0x100 and above must be unique, even among different model instruments. The following table lists the service codes for the AccuLoad IV:

Table 16: AccuLoad IV Service Codes

Service Code	Model	Function
0x000	All	Unit information: model number, options, serial number, etc.
0x001	All	Read clock
0x002	All	Set clock
0x0400	AccuLoad IV	Transaction control
0x0401	AccuLoad IV	Display control (prompting, etc)
0x0402	AccuLoad IV	Read from event log
0x0403	AccuLoad IV	Search event log
0x0404	AccuLoad IV	Read from transaction log
0x0405	AccuLoad IV	Search transaction log
0x0406	AccuLoad IV	Read audit trail
0x0407	AccuLoad IV	Search audit trail

All of the AccuLoad IV responses will have at least one 16-bit unsigned integer response code, possibly followed by more data (herein referred to as the "standard response code"). The response codes are broken down into four classes as follows:

Table 17: Response Code Range and Class

Response Code Range	Class
0x0000 – 0x3fff	Normal response: Command was executed.
0x4000 – 0x7fff	Warning response: Command was executed; instrument communications status has not changed; a notable event occurred when the command was executed.
0x8000 – 0xbfff	Critical response: Command was not executed; instrument communications status has changed.
0xc000 – 0xffff	Fatal response: Command was not executed; instrument communications status has changed and recovery is necessary to continue.

Table 18: Standard Response Codes

Error Message	Error Code (hexadecimal)
No error	0x0000
Download successful	0x0001
Empty config	0x8000
Block not found	0x8001
Bad message format	0x8002
Block out of sequence	0x8003
Invalid flash data	0x8004
Target buffer too small	0x8005
Bad CRC	0x8006
No config avail	0x8007
Download already in progress	0x8008
Bad database spec	0x8009
In program mode	0x800a
Released	0x800b
Bad value	0x800c
Flow active	0x800d
No transactions ever done	0x800e

Error Message	Error Code (hexadecimal)
Operation not allowed	0x800f
Wrong control mode	0x8010
Transaction in progress	0x8011
Alarm active	0x8012
Storage full	0x8013
Operation out of sequence	0x8014
Power fail during transaction	0x8015
Authorized	0x8016
Program code not used	0x8017
Disp key in remote control	0x8018
Ticket not in printer	0x8019
No key data pending	0x801a
No transaction in progress	0x801b
Option not installed	0x801c
Start after stop delay	0x801d
Permissive delay active	0x801e
Print request pending	0x801f
No meter enabled	0x8020
Must be in program mode	0x8021
Ticket alarm during transaction	0x8022
Volume type not selected	0x8023
Exactly one rec must be enabled	0x8024
Batch limit reached	0x8025
Checking entries	0x8026
Prod rec add not assigned	0x8027
Must use mini protocol	0x8028
Buffer error	0x8029
Keypad locked	0x802a
Data recall error	0x802b
Internal error	0x802c
Transmit reply	0x802d
Unknown error	0x802e
Unused batch	0x802f
Packing memory (wait, and try again)	0x8030
Data not available	0x8031
Card In Required	0x8032
Too many shared additives	0x8033
Max active arms in use	8x8034
Transaction not standby	0x8035
Swing arm out of position	0x8036
No current batch on arm	0x8037
Internal flash error	0xc000
Flash has overrun	0xc001
Internal buffer error	0xc002

Error Message	Error Code (hexadecimal)
Buffer allocation error	0xc003
Buffer overrun	0xc004
Flash erase error	0xc005
Flash write error	0xc006

4.2 0x0000 Read Unit Information

This command returns standard unit information such as manufacturer code, model number, and serial number if available. This command is included so the host, or other units in a shared protocol, can determine what type of unit is occupying a certain address on the COM port.

4.2.1 Command Data for Read Unit Information

Table 19: 0x0000 Read Unit Information Command Data

Modbus Address	Example Data	Data Type	Description
0	2	unsigned	Size of Extended Services Packet
1	0x0000	unsigned	Router Word, "Read unit information" service

4.2.2 Response Data for Read Unit Information

Table 20: 0x0000 Read Unit Information Response Data

Modbus Address	Example Data	Data Type	Description
0	30	unsigned	Size of extended services packet
1	0x8000	unsigned	Router word, "read unit information" service
2	0x0000	unsigned	Standard response code
3	0x0001	unsigned	Unit manufacturer code (Smith=0x0001)
4	0x0011	unsigned	Unit model code (ALIII-D=0x0011) (ALIII-X=0x0013, ALIV=0x0014)
5 - 12	""	text	Unit serial number (not used in AccuLoad)
13	1002 = 10.02	unsigned	Firmware revision
14 - 15	0xA4E3025A	unsigned long	ROM CRC-32
16 -			Reserved for future additions

4.3 0x0001 Read Clock

This command returns the time and date indicated by the internal clock.

4.3.1 Command Data for Read Clock

Table 21: 0x0001 Read Clock Command Data

Modbus Address	Example Data	Data Type	Description
0	2	unsigned	Size of extended services packet
1	0x0001	unsigned	Router word, "read clock" service

4.3.2 Response Data for Read Clock

Table 22: 0x0001 Read Clock Response Data

Modbus Address	Example Data	Data Type	Description
0	18	unsigned	Size of extended services packet
1	0x8001	unsigned	Router word, "read clock" service
2	0x0000	unsigned	Standard response code
3	2000	unsigned	Year
4	6	unsigned	Month
5	4	unsigned	Day
6	0	unsigned	(reserved)
7	59	unsigned	Seconds
8	31	unsigned	Minutes
8	21	unsigned	Hours
9	0	unsigned	(reserved)

4.4 0x0002 Set Clock

This command sets the time and date in the internal clock.

4.4.1 Command Data for Set Clock

Table 23: 0x0002 Set Clock Command Data

Modbus Address	Example Data	Data Type	Description
0	18	unsigned	Size of extended services packet
1	0x0002	unsigned	Router word, "set clock" service
2	2009	unsigned	Year
3	6	unsigned	Month
4	4	unsigned	Day
5	0	unsigned	(reserved)
6	59	unsigned	Seconds
7	31	unsigned	Minutes
8	21	unsigned	Hours
9	0	unsigned	(reserved)

4.4.2 Response Data for Set Clock

Table 24: 0x0002 Set Clock Response Data

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x8001	unsigned	Router word, "set clock" service
2	0x0000	unsigned	Standard response code

4.5 0x0400 Transaction Control

This service allows transactions to be controlled using communications. There are several different subcommands variations.

4.5.1 Variation #1 Command Data

Authorize transaction, equivalent to Smith Protocol Commands AP and AU:

The AP Command authorizes the transaction to the Preset Prompt, and the AU Command authorizes the transaction and leaves the AccuLoad at the ready prompt. Both authorizations can be used with or without authorizing additives.

Modbus Address	Example Data	Data Type	Description
0	10	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	0	unsigned	Sub-command (0=variation 1)
3	1	unsigned	Prompting option; 0=wait for set key, 1=show preset screen now
4	-1	long integer	Additive selection; bits 0-23 indicate injector selections 1-24, 0=off, 1=on. Value of -1 means "use all injectors programmed"

4.5.2 Variation #2 Command Data

Set transaction, equivalent to Smith Protocol Command TA:

The TA command sets the maximum transaction volume. Units must correspond to what is programmed in the AccuLoad.

Modbus Address	Example Data	Data Type	Description
0	8	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	1	unsigned	Sub-command (1=variation 2)
3 - 4	100.0	float	preset volume

4.5.3 Variation #3 Command Data

Allocate recipes, equivalent to Smith Protocol Command AB):

The AB Command allocates the recipes that will be allowed for that transaction.

Modbus Address	Example Data	Data Type	Description
0	12	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	2	unsigned	Sub-command (2=variation 3)
3 - 4	0x00000021	unsigned long	Recipe selections; bits 0-31 enable recipes 1-32 (0=disabled, 1=enabled)
5 - 6	0x00000000	unsigned long	Recipe selections; bits 32-49 enable recipes 33-50 (0=disabled, 1=enabled). Bits 27-31 are reserved for future use.

4.5.4 Variation #4 Command Data

Set batch, equivalent to Smith Protocol Command SB:

The SB Command authorizes the batch and presets the volume for that batch. Additives can also be selected with the SB Command. The preset value must not exceed the programmed maximum batch size and must not be below the programmed minimum batch size. A batch size of 0 allows the driver to enter the batch size.

Modbus Address	Example Data	Data Type	Description
0	12	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	3	unsigned	Sub-command (3=variation 4)
3 - 4	250.0	float	Preset volume
5 - 6	0X 0000 0005	unsigned long	Additive selection; bits 0-23 enable injectors 1-24 (0=disabled, 1=enabled). Bits 24-31 are reserved for future use.

4.5.5 Variation #5 Command Data

End batch, equivalent to Smith Protocol Command EB:

The EB Command cancels the remaining batch volume and ends the batch, closing the valve if it has not already been commanded to do so.

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	4	unsigned	Sub-command (4=variation 5)

4.5.6 Variation #6 Command Data

End transaction, equivalent to Smith Protocol Command ET:

The ET Command ends the transaction, removes authorization, and flags the transaction as complete.

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	5	unsigned	Sub-command (5=variation 6)

4.5.7 Variation #7 Command Data

Remote start, equivalent to Smith Protocol Command SA:

The SA command remotely starts the AccuLoad IV. It is the same as pressing the START key on the AccuLoad.

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	6	unsigned	Sub-command (6=variation 7)

4.5.8 Variation #8 Command Data

Remote stop, equivalent to Smith Protocol Command SP:

The SP Command instructs the AccuLoad IV to stop, halting product delivery. The valve and pump are shut down whether flow is present or not. If a batch is in progress, the START key or the Remote Start Command must be used to continue the batch.

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	7	unsigned	Sub-command (7=variation 8)

4.5.9 Variations 1-8 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	4	unsigned	Sub-command (4=variation 5)

4.5.10 Variation #9 Command Data

Read status flags:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	8	unsigned	Sub-command (8=variation 9)

4.5.11 Variation #9 Response Data

Flags: 0=False, non-zero=True

Modbus Address	Example Data	Data Type	Description
0	38	unsigned	Size of extended services packet
1	0x8300	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	8	unsigned	Sub-command (8=variation 9)
4	1	unsigned	Authorized flag
5	1	unsigned	Released flag (valve commanded to open)
6	1	unsigned	Transaction in progress flag
7	0	unsigned	Batch done flag
8	0	unsigned	Transaction done flag
9	0	unsigned	Start/stop delay active flag
10	0	unsigned	Valve open delay active flag
11	1	unsigned	Product flowing flag
12	0	unsigned	Injectors authorized using communications flag
13	0	unsigned	Proving in progress flag
14	0	unsigned	Alarm active flag
15	0	unsigned	In program mode flag
16	0	unsigned	Checking program mode parameters flag
17	1	unsigned	Program value changed flag
18	1	unsigned	Power fail occurred flag
19	0	unsigned	Transaction report queued for printing flag
20	A	unsigned	Swing arm position
21	0	unsigned	In Standby mode
22	0	unsigned	Storage full
23	0	unsigned	Transaction lock in effect

4.5.12 Variation #10 Clear Transaction Done Flag and Batch Done Flag

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	9	unsigned	Sub-command (9=variation 10)

4.5.13 Variation #10 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	9	unsigned	Sub-command (9=variation 10)

4.5.14 Variation #11 Clear Batch Done Flag

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	10	unsigned	Sub-command (10=variation 11)

4.5.15 Variation #11 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	10	unsigned	Sub-command (10=variation 11)

4.5.16 Variation #12 Clear Power Fail Flag

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	11	unsigned	Sub-command (11=variation 12)

4.5.17 Variation #12 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	11	unsigned	Sub-command (11=variation 12)

4.5.18 Variation #13 Clear Program Parameter Changed Flag

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	12	unsigned	Sub-command (12=variation 13)

4.5.19 Variation #13 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	12	unsigned	Sub-command (12=variation 13)

4.5.20 Variation #14 Command Data

Remote stop on an arm, equivalent to Smith protocol command ST:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	13	unsigned	Sub-command (13=variation 14)

4.5.21 Variation #14 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	13	unsigned	Sub-command (13=variation 14)

4.5.22 Variation #15 Command Data

Set batch fixed, equivalent to Smith protocol command SF:

Modbus Address	Example Data	Data Type	Description
0	12	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	14	unsigned	Sub-command (14=variation 15)
3 - 4	2000	float	Preset volume
5 - 6	0X 0000 0005	Long integer	Additive selection; Bits 0-23 enable injectors 1-24 (0=disabled, 1=enabled). Bits 24-31 are reserved for future use.

4.5.23 Variation #15 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	14	unsigned	Sub-command (14=variation 15)

4.5.24 Variation #16 Command Data

Clear Standby Mode Status and Resume Communications is equivalent to Smith protocol command RE SA:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	15	unsigned	Sub-command (15=variation 16)

4.5.25 Variation #16 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	15	unsigned	Sub-command (15=variation 16)

4.5.26 Variation #17 Command Data

Clear Standby Transaction Lock, equivalent to Smith protocol command CT:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	16	unsigned	Sub-command (16=variation 17)

4.5.27 Variation #17 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	16	unsigned	Sub-command

4.5.28 Variation #18 Command Data

New Recipe, equivalent to Smith protocol command NR:

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	17	unsigned	Sub-command (17=variation 18)
3	2	unsigned	Recipe number

4.5.29 Variation #19 Command Data

Print report to printer, equivalent to Smith protocol command PP:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0400	unsigned	Router word, "transaction control" service
2	18	unsigned	Sub-command (18=variation 19)

4.5.30 Variations #18 and #19 Response Data

Modbus Address	Example Data	Data Type	Description
0	6 or 12	unsigned	Size of extended services packet
1	0x8400	unsigned	Router word, "transaction control" service
2	0x0000	unsigned	Standard return code
3	17 or 18	unsigned	Sub-command

4.6 0x0401 Display Control

This service allows the display to be controlled using communications. There are several different subcommands/variations.

4.6.1 Variation #1 Command Data

Write to display, equivalent to Smith protocol commands WA, WB, WC, WD, WE, WP, WQ, and WX:

Modbus Address	Example Data	Data Type	Description
0	32	unsigned	Size of extended services packet
1	0x0401	unsigned	Router word, "display control" service
2	0	unsigned	Sub-command (0=variation 1)
3	1	unsigned	Display line number (1 to 5) – A command must be sent to display line 1 before any additional commands may be sent to display lines 2-5.
4	60	unsigned	Prompt timeout (in seconds, 0=no timeout) – allows the prompt to "expire" after a certain amount of time if the operator ignores the prompt.
5	0	unsigned	Wait for Set Key Pressed Before Displaying prompt; 0=no, 1=yes. Selecting "yes" here prevents the prompt from showing immediately; the prompt is held up until the operator presses the SET key.
6	5	unsigned	Expected response length; 0=no response (display text only). 1-20 = number of characters that must be entered. Add 40 to the character count if entering fewer characters is permissible.
7	93	unsigned	Operator entry procedure: 38 = operator enters number and then any function key but CLEAR or STOP to terminate entry. 91 = ENTER must be pressed before entering a number, any function key but CLEAR or STOP to terminate entry. 93 = ENTER must be pressed before entering a number, ENTER must be pressed to terminate entry.
8	1	unsigned	Security echo: 0=normal character echo displayed, 1=security character echo "X" displayed
9 - 23	"Enter Truck ID"	text [30 chars max]	Text message to display. Messages that are too long will be truncated on the display.

4.6.2 Variation #1 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8401	unsigned	Router word, "display control" service
2	0x0000	unsigned	Standard return code
3	0	unsigned	Sub-command (0=variation 1)

4.6.3 Variation #2 Command Data

Release the keypad and display, equivalent to Smith protocol command DA:

The DA command returns the control of the keypad and display to the AccuLoad IV.

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0401	unsigned	Router word, "display control" service
2	1	unsigned	Sub-command (1=variation 2)

4.6.4 Variation #2 Response Data

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended Services Packet
1	0x8401	unsigned	Router word, "display control" service
2	0x0000	unsigned	Standard return code
3	1	unsigned	Sub-command (1=variation 2)

4.6.5 Variation #3 Command Data

Get last key that was pressed, equivalent to Smith protocol command GK:

The GK command retrieves the last key that was pressed at the AccuLoad keypad. If no key is pressed, the AccuLoad IV returns a "no" response.

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0401	unsigned	Router word, "display control" service
2	2	unsigned	Sub-command (2=variation 3)

4.6.6 Variation #3 Response Data

Modbus Address	Example Data	Data Type	Description
0	8	unsigned	Size of extended services packet
1	0x8401	unsigned	Router word, "display control" service
2	0x0000	unsigned	Standard return code
3	2	unsigned	Sub-command (2=variation 3)
4	"A1"	text	ASCII characters, string may be null terminated (\0). Leading asterisk if arm in focus; leading space otherwise. "0"-"9"=keys 0-9, "E1"=ENTER, "P1"=PRINT, "A1"=START, "B1"=SET, "C1"=CLEAR, "S1"=STOP

4.6.7 Variation #4 Command Data

Read keypad data, equivalent to Smith protocol command RK:

The RK command instructs the AccuLoad to transmit any pending data entered at the keypad to the requesting device.

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0401	unsigned	Router word, "display control" service
2	3	unsigned	Sub-command (3=variation 4)

4.6.8 Variation #4 Response Data

Modbus Address	Example Data	Data Type	Description
0	14	unsigned	Size of extended services packet
1	0x8401	unsigned	Router word, "display control" service
2	0x0000	unsigned	Standard return code
3	3	unsigned	Sub-command (3=variation 4)
4 - 7	"14937E1"	text	ASCII characters, string may be null terminated (\0). "0"-"9"=keys 0-9, Last two characters indicate the key that was pressed to terminate prompt entry: "E1"=ENTER, "P1"=PRINT, "A1"=START, "B1"=SET, "C1"=CLEAR, "S1"=STOP

4.6.9 Variation #5 Command Data

Read status flags:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0401	unsigned	Router word, "display control" service
2	4	unsigned	Sub-command (4=variation 5)

4.6.10 Variation #5 Response Data

Flags: 0=False, non-zero=True

Modbus Address	Example Data	Data Type	Description
0	26	unsigned	Size of extended services packet
1	0x8401	unsigned	Router word, "display control" service
2	0x0000	unsigned	Standard return code
3	4	unsigned	Sub-command (4=variation 5)
4	0	unsigned	Authorized flag
5	1	unsigned	Keypad data pending flag
6	0	unsigned	Delayed prompt in effect flag
7	0	unsigned	Display message timeout flag
8	0	unsigned	Alarm active flag
9	0	unsigned	In program mode flag
10	0	unsigned	Checking program mode parameters flag
11	1	unsigned	Program value changed flag
12	0	unsigned	Power fail occurred flag
13	0	unsigned	Transaction report queued for printing flag

4.6.11 Variation #6 Command Data

Switch to full screen mode (sets focus when using System Status Display):

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0401	unsigned	Router word, "display control" service
2	5	unsigned	Sub-command (5=variation 6)

4.6.12 Variation #6 Response Data

Flags: 0=False, non-zero=True

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8401	unsigned	Router word, "display control" service
2	0x0000	unsigned	Standard return code
3	5	unsigned	Sub-command (5=variation 6)

4.6.13 Variation #7 Command Data

Switch to Split Screen Mode (ignored in AccuLoad IV):

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0401	unsigned	Router word, "display control" service
2	6	unsigned	Sub-command (6=variation 7)

4.6.14 Variation #7 Response Data

Flags: 0=False, non-zero=True

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8401	unsigned	Router word, "display control" service
2	0x0000	unsigned	Standard return code
3	6	unsigned	Sub-command (6=variation 7)

4.6.15 Variation #8 Command Data

Force swing arm to opposite MMI:

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x0401	unsigned	Router word, "display control" service
2	7	unsigned	Sub-command (7=variation 8)
3	65	unsigned	Side to position arm; either 65 (0x41) for MMI “A” or 66 (0x42) for MMI “B”

4.6.16 Variation #8 Response Data

Flags: 0=False, non-zero=True

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x8401	unsigned	Router word, "display control" service
2	0x0000	unsigned	Standard return code
3	7	unsigned	Sub-command (7=variation 8)

4.7 0x0402 Read Event Log

This service reads a record from the specified event log stored in non-volatile storage.

4.7.1 Command Data for Read Event Log

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x0402	unsigned	Router word, "read event log" service
2 - 3	87921	unsigned long	Sequence number of event

4.7.2 Response Data for Read Event Log

Modbus Address	Example Data	Data Type	Description
0	114	unsigned	Size of extended services packet
1	0x8402	unsigned	Router word, "read event log" service
2	0x0000	unsigned	Standard response code
3 - 47	"Transaction Ended Arm 2 - Gasoline 89 Transaction #6"	text	Message text; may be null terminated (\0)
48	0	unsigned	(reserved)
49	2000	unsigned	Log year
50	1	unsigned	Log month
51	25	unsigned	Log day
52	0	unsigned	(reserved)
53	47	unsigned	Log seconds
54	21	unsigned	Log minutes
55	16	unsigned	Log hours
56	0	unsigned	(reserved)
57	0	unsigned	(reserved)

4.8 0x0403 Search Event Log

This service searches the event log for the latest entry, the oldest entry still available in memory, or the most recent entry that falls before a given date and time. The service returns the sequence number of the entry.

4.8.1 Variation #1 Command Data

Returns most recent record number:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0403	unsigned	Router word, "search event log" service
2	1	unsigned	Sub-command (1=variation 1)

4.8.2 Variation #2 Command Data

Returns oldest available record number:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0403	unsigned	Router word, "search event log" service
2	2	unsigned	Sub-command (2=variation 2)

4.8.3 Variation #3 Command Data

Searches on specified date and time:

Modbus Address	Example Data	Data Type	Description
0	20	unsigned	Size of extended services packet
1	0x0403	unsigned	Router word, "search event log" service
2	3	unsigned	Sub-command (3=variation 3)
3	2000	unsigned	Log year
4	4	unsigned	Log month
5	12	unsigned	Log day
6	0	unsigned	(reserved)
7	0	unsigned	Log seconds
8	0	unsigned	Log minutes
9	14	unsigned	Log hours
10	0	unsigned	(reserved)

4.8.4 Variation #3 Response Data

Modbus Address	Example Data	Data Type	Description
0	8	unsigned	Size of extended services packet
1	0x8403	unsigned	Router word, "search event log" service
2	0x0000	unsigned	Standard response code
3 - 4	93543	unsigned long	Sequence number of log entry

4.9 0x0404 Read Transaction Log

This command retrieves the transaction specified by the sequence number from nonvolatile storage and places the data in the extended service response buffer in the following order. The data can then be read from the buffer by the host.

4.9.1 Command Data for Command Code 0

Modbus Address	Example Data	Data Type	Description
0	8	unsigned	Size of extended services packet
1	0x0404	unsigned	Router word, "read transaction log" service
2 - 3	84118	unsigned long	Sequence number of transaction log entry
4	0	unsigned	Command code; 0=transaction data, 1-10=batch data, batch 1-10, 11=bay non-resettable totals, 12=user registers

4.9.2 Response Data for for Command Code 0

Transaction data:

Modbus Address	Description	Data Type
5	Transaction number	unsigned integer
6	Number of batches delivered	unsigned integer
7	Transaction end time (year)	unsigned integer
8	Transaction end time (month)	unsigned integer
9	Transaction end time (day)	unsigned integer
10	Transaction end time (day of week)	unsigned integer
11	Transaction end time (seconds)	unsigned integer
12	Transaction end time (minutes)	unsigned integer
13	Transaction end time (hours)	unsigned integer
14	Transaction end time (reserved)	unsigned integer
15	Prompt #1 response	unsigned long integer
17	Prompt #2 response	unsigned long integer
19	Prompt #3 response	unsigned long integer
21	Prompt #4 response	unsigned long integer
23	Prompt #5 response	unsigned long integer

Modbus Address	Description	Data Type
25	Number of transaction alarms	unsigned integer
26	Transaction alarm 1 log	char[10]
31	Transaction alarm 2 log	char[10]
36	Transaction alarm 3 log	char[10]
41	Transaction alarm 4 log	char[10]
46	Transaction alarm 5 log	char[10]
51	Transaction alarm 6 log	char[10]
56	Transaction alarm 7 log	char[10]
61	Transaction alarm 8 log	char[10]
66	Transaction alarm 9 log	char[10]
71	Transaction alarm 10 log	char[10]
76	Transaction alarm 11 log	char[10]
81	Transaction alarm 12 log	char[10]
86	Transaction alarm 13 log	char[10]
91	Transaction alarm 14 log	char[10]
96	Transaction alarm 15 log	char[10]
101	Transaction alarm 16 log	char[10]
106	Transaction alarm 17 log	char[10]
111	Transaction alarm 18 log	char[10]
116	Transaction alarm 19 log	char[10]
121	Transaction alarm 20 log	char[10]
126	Transaction average meter factor	float
128	Transaction average temperature	float
130	Transaction average density	float
132	Transaction average pressure	float
134	Transaction average CTL	float
136	Transaction average CPL	float
138	Transaction additive 1 volume	double
142	Transaction additive 2 volume	double
146	Transaction additive 3 volume	double
150	Transaction additive 4 volume	double
154	Transaction additive 5 volume	double
158	Transaction additive 6 volume	double
162	Transaction additive 7 volume	double
166	Transaction additive 8 volume	double
170	Transaction additive 9 volume	double
174	Transaction additive 10 volume	double
178	Transaction additive 11 volume	double
182	Transaction additive 12 volume	double
186	Transaction additive 13 volume	double
190	Transaction additive 14 volume	double
194	Transaction additive 15 volume	double
198	Transaction additive 16 volume	double
202	Transaction additive 17 volume	double

Modbus Address	Description	Data Type
206	Transaction additive 18 volume	double
210	Transaction additive 19 volume	double
214	Transaction additive 20 volume	double
218	Transaction additive 21 volume	double
222	Transaction additive 22 volume	double
226	Transaction additive 23 volume	double
230	Transaction additive 24 volume	double
234	Transaction raw volume	double
238	Transaction gross volume	double
242	Transaction GST volume	double
246	Transaction GSV volume	double
250	Transaction mass	double
254	Transaction product 1 ending non-resettable raw totalizer	double
258	Transaction product 1 ending non-resettable gross totalizer	double
262	Transaction product 1 ending non-resettable GST totalizer	double
266	Transaction product 1 ending non-resettable GSV totalizer	double
270	Transaction product 1 ending non-resettable mass totalizer	double
274	Transaction product 2 ending non-resettable raw totalizer	double
278	Transaction product 2 ending non-resettable gross totalizer	double
282	Transaction product 2 ending non-resettable GST totalizer	double
286	Transaction product 2 ending non-resettable GSV totalizer	double
290	Transaction product 2 ending non-resettable mass totalizer	double
294	Transaction product 3 ending non-resettable raw totalizer	double
298	Transaction product 3 ending non-resettable gross totalizer	double
302	Transaction product 3 ending non-resettable GST totalizer	double
306	Transaction product 3 ending non-resettable GSV totalizer	double
310	Transaction product 3 ending non-resettable mass totalizer	double
314	Transaction product 4 ending non-resettable raw totalizer	double
318	Transaction product 4 ending non-resettable gross totalizer	double
322	Transaction product 4 ending non-resettable GST totalizer	double
326	Transaction product 4 ending non-resettable GSV totalizer	double
330	Transaction product 4 ending non-resettable mass totalizer	double
334	Transaction product 5 ending non-resettable raw totalizer	double
338	Transaction product 5 ending non-resettable gross totalizer	double
342	Transaction product 5 ending non-resettable GST totalizer	double
346	Transaction product 5 ending non-resettable GSV totalizer	double
350	Transaction product 5 ending non-resettable mass totalizer	double
354	Transaction product 6 ending non-resettable raw totalizer	double
358	Transaction product 6 ending non-resettable gross totalizer	double
362	Transaction product 6 ending non-resettable GST totalizer	double
366	Transaction product 6 ending non-resettable GSV totalizer	double
370	Transaction product 6 ending non-resettable mass totalizer	double
374	Transaction end time (text)	char[22]
385	Transaction start time (text)	char[22]

Modbus Address	Description	Data Type
396	Alphanumeric prompt response #1	char[20]
406	Alphanumeric prompt response #2	char[20]
416	Alphanumeric prompt response #3	char[20]
426	Alphanumeric prompt response #4	char[20]
436	Alphanumeric prompt response #5	char[20]
446	Driver card raw card data	char[48]
470	Driver card field 1	char[32]
486	Driver card field 2	char[32]
502	Driver card field 3	char[32]
518	Driver card HID factory code	char[32]
534	Driver card HID number	char[32]
550	Transaction Vapor Recovery	double

4.9.3 Response Data for Command Code 1-10

Batch data, 1-10 is batch number:

Modbus Address	Description	Data Type
5	Product delivered	unsigned integer
6	Recipe delivered	unsigned integer
7	HM class product	unsigned integer
8	Additives delivered (bit map)	unsigned long integer
10	Batch average flow rate	float
12	Batch average meter factor	float
14	Batch average temperature	float
16	Batch average density	float
18	Batch average pressure	float
20	Batch average CTL	float
22	Batch average CPL	float
24	Meter pulses	double
28	Batch raw volume	double
32	Batch gross volume	double
36	Batch GST volume	double
40	Batch GSV volume	double
44	Batch mass	double
48	Batch additive 1 volume	double
52	Batch additive 2 volume	double
56	Batch additive 3 volume	double
60	Batch additive 4 volume	double
64	Batch additive 5 volume	double
68	Batch additive 6 volume	double
72	Batch additive 7 volume	double
76	Batch additive 8 volume	double
80	Batch additive 9 volume	double
84	Batch additive 10 volume	double

Modbus Address	Description	Data Type
88	Batch additive 11 volume	double
92	Batch additive 12 volume	double
96	Batch additive 13 volume	double
100	Batch additive 14volume	double
104	Batch additive 15 volume	double
108	Batch additive 16 volume	double
112	Batch additive 17 volume	double
116	Batch additive 18 volume	double
120	Batch additive 19 volume	double
124	Batch additive 20 volume	double
128	Batch additive 21 volume	Double
132	Batch additive 22 volume	Double
136	Batch additive 23 volume	Double
140	Batch additive 24 volume	Double
144	Number of batch alarms	unsigned integer
145	Batch alarm 1 log	char[8]
149	Batch alarm 2 log	char[8]
153	Batch alarm 3 log	char[8]
157	Batch alarm 4 log	char[8]
161	Batch alarm 5 log	char[8]
165	Batch alarm 6 log	char[8]
169	Batch alarm 7 log	char[8]
173	Batch alarm 8 log	char[8]
177	Batch alarm 9 log	char[8]
181	Batch alarm 10 log	char[8]
185	Batch product 1 average flow rate	float
187	Batch product 1 average meter factor	float
189	Batch product 1 average temperature	float
191	Batch product 1 average density	float
193	Batch product 1 average pressure	float
195	Batch product 1 average vapor pressure	float
197	Batch product 1 average CTL	float
199	Batch product 1 average CPL	float
201	Batch product 1 average CCF	float
203	Batch product 1 reference density	float
205	Batch product 1 relative density	float
207	Batch product 1 API density	float
209	Batch product 1 meter pulses	double
213	Batch product 1 raw volume	double
217	Batch product 1 gross volume	double
221	Batch product 1 GST volume	double
225	Batch product 1 GSV volume	double
229	Batch product 1 mass	double
233	Batch product 2 average flow rate	float

Modbus Address	Description	Data Type
235	Batch product 2 average meter factor	float
237	Batch product 2 average temperature	float
239	Batch product 2 average density	float
241	Batch product 2 average pressure	foat
243	Batch product 2 average vapor pressure	float
245	Batch product 2 average CTL	float
247	Batch product 2 average CPL	float
249	Batch product 2 average CCF	float
251	Batch product 2 reference density	float
253	Batch product 2 relative density	float
255	Batch product 2 API density	float
257	Batch product 2 meter pulses	double
261	Batch product 2 raw volume	double
265	Batch product 2 gross volume	double
269	Batch product 2 GST volume	double
273	Batch product 2 GSV volume	double
277	Batch product 2 mass	double
281	Batch product 3 average flow rate	float
283	Batch product 3 average meter factor	float
285	Batch product 3 average temperature	float
287	Batch product 3 average density	float
289	Batch product 3 average pressure	float
291	Batch product 3 average vapor pressure	float
293	Batch product 3 average CTL	float
295	Batch product 3 average CPL	float
297	Batch product 3 average CCF	float
299	Batch product 3 reference density	float
301	Batch product 3 relative density	float
303	Batch product 3 API density	float
305	Batch product 3 meter pulses	double
309	Batch product 3 raw volume	double
313	Batch product 3 gross volume	double
317	Batch product 3 GST volume	double
321	Batch product 3 GSV volume	double
325	Batch product 3 mass	double
329	Batch product 4 average flow rate	float
331	Batch product 4 average meter factor	float
333	Batch product 4 average temperature	float
335	Batch product 4 average density	float
337	Batch product 4 average pressure	float
339	Batch product 4 average vapor pressure	float
341	Batch product 4 average CTL	float
343	Batch product 4 average CPL	float
345	Batch product 4 average CCF	float

Modbus Address	Description	Data Type
347	Batch product 4 reference density	float
349	Batch product 4 relative density	float
351	Batch product 4 API density	float
353	Batch product 4 meter pulses	double
357	Batch product 4 raw volume	double
361	Batch product 4 gross volume	double
365	Batch product 4 GST volume	double
369	Batch product 4 GSV volume	double
373	Batch product 4 mass	double
377	Batch product 5 average flow rate	float
379	Batch product 5 average meter factor	float
381	Batch product 5 average temperature	float
383	Batch product 5 average density	float
385	Batch product 5 average pressure	float
387	Batch product 5 average vapor pressure	float
389	Batch product 5 average CTL	float
391	Batch product 5 average CPL	float
393	Batch product 5 average CCF	float
395	Batch product 5 reference density	float
397	Batch product 5 relative density	float
399	Batch product 5 API density	float
401	Batch product 5 meter pulses	double
405	Batch product 5 raw volume	double
409	Batch product 5 gross volume	double
413	Batch product 5 GST volume	double
417	Batch product 5 GSV volume	double
421	Batch product 5 mass	double
425	Batch product 6 average flow rate	float
427	Batch product 6 average meter factor	float
429	Batch product 6 average temperature	float
431	Batch product 6 average density	float
433	Batch product 6 average pressure	float
435	Batch product 6 average vapor pressure	float
437	Batch product 6 average CTL	float
439	Batch product 6 average CPL	float
441	Batch product 6 average CCF	float
443	Batch product 6 reference density	float
445	Batch product 6 relative density	float
447	Batch product 6 API density	float
449	Batch product 6 meter pulses	double
453	Batch product 6 raw volume	double
457	Batch product 6 gross volume	double
461	Batch product 6 GST volume	double
465	Batch product 6 GSV volume	double

Modbus Address	Description	Data Type
469	Batch product 6 mass	double
473	Batch last density sample	float
475	Batch contaminant percentage	float
477	Batch load arm (for bay-based transactions)	unsigned integer
478	Additive 1 flow control Inj GV volume	double
482	Additive 1 flow control Inj GST volume	double
486	Additive 1 flow control Inj mass	double
490	Additive 2 flow control Inj GV volume	double
494	Additive 2 flow control Inj GST volume	double
498	Additive 2 flow control Inj Mass	double
502	Additive 3 flow control Inj GV volume	double
506	Additive 3 flow control Inj GST volume	double
510	Additive 3 flow control Inj mass	double
514	Additive 4 flow control Inj GV volume	double
518	Additive 4 flow control Inj GST volume	double
522	Additive 4 flow control Inj mass	double
526	Batch Vapor Recovery	double

4.9.4 Response Data for Command Code 11

Bay non-resettable totals:

Modbus Address	Description	Data Type
5	Arm 1 product 1 IV non-resettable totalizer	double
9	Arm 1 product 1 GV non-resettable totalizer	double
13	Arm 1 product 1 GST non-resettable totalizer	double
17	Arm 1 product 1 GSV non-resettable totalizer	double
21	Arm 1 product 1 mass non-resettable totalizer	double
25	Arm 1 product 2 IV non-resettable totalizer	double
29	Arm 1 product 2 GV non-resettable totalizer	double
33	Arm 1 product 2 GST non-resettable totalizer	double
37	Arm 1 product 2 GSV non-resettable totalizer	double
41	Arm 1 product 2 mass non-resettable totalizer	double
45	Arm 1 product 3 IV non-resettable totalizer	double
49	Arm 1 product 3 GV non-resettable totalizer	double
53	Arm 1 product 3 GST non-resettable totalizer	double
57	Arm 1 product 3 GSV non-resettable totalizer	double
61	Arm 1 product 3 mass non-resettable totalizer	double
65	Arm 1 product 4 IV non-resettable totalizer	double
69	Arm 1 product 4 GV non-resettable totalizer	double
73	Arm 1 product 4 GST non-resettable totalizer	double
77	Arm 1 product 4 GSV non-resettable totalizer	double
81	Arm 1 product 4 mass non-resettable totalizer	double
85	Arm 1 product 5 IV non-resettable totalizer	double
89	Arm 1 product 5 GV non-resettable totalizer	double

Modbus Address	Description	Data Type
93	Arm 1 product 5 GST non-resettable totalizer	double
97	Arm 1 product 5 GSV non-resettable totalizer	double
101	Arm 1 product 5 mass non-resettable totalizer	double
105	Arm 1 product 6 IV non-resettable totalizer	double
109	Arm 1 product 6 GV non-resettable totalizer	double
113	Arm 1 product 6 GST non-resettable totalizer	double
117	Arm 1 product 6 GSV non-resettable totalizer	double
121	Arm 1 product 6 mass non-resettable totalizer	double
125	Arm 2 product 1 IV non-resettable totalizer	double
129	Arm 2 product 1 GV non-resettable totalizer	double
133	Arm 2 product 1 GST non-resettable totalizer	double
137	Arm 2 product 1 GSV non-resettable totalizer	double
141	Arm 2 product 1 mass non-resettable totalizer	double
145	Arm 2 product 2 IV non-resettable totalizer	double
149	Arm 2 product 2 GC non-resettable totalizer	double
153	Arm 2 product 2 GST non-resettable totalizer	double
157	Arm 2 product 2 GSV non-resettable totalizer	double
161	Arm 2 product 2 mass non-resettable totalizer	double
165	Arm 2 product 3 IV non-resettable totalizer	double
169	Arm 2 product 3 GV non-resettable totalizer	double
173	Arm 2 product 3 GST non-resettable totalizer	double
177	Arm 2 product 3 GSV non-resettable totalizer	double
181	Arm 2 product 3 Mass non-resettable totalizer	double
185	Arm 2 product 4 IV non-resettable totalizer	double
189	Arm 2 product 4 GV non-resettable totalizer	Double
193	Arm 2 product 4 GST non-resettable totalizer	double
197	Arm 2 product 4 GSV non-resettable totalizer	double
201	Arm 2 product 4 mass non-resettable totalizer	double
205	Arm 2 product 5 IV non-resettable totalizer	double
209	Arm 2 product 5 GV non-resettable totalizer	double
213	Arm 2 product 5 GST non-resettable totalizer	double
217	Arm 2 product 5 GSV non-resettable totalizer	double
221	Arm 2 product 5 mass non-resettable totalizer	double
225	Arm 2 product 6 IV non-resettable totalizer	double
229	Arm 2 product 6 GV non-resettable totalizer	double
233	Arm 2 product 6 GST non-resettable totalizer	double
237	Arm 2 product 6 GSV non-resettable totalizer	double
241	Arm 2 product 6 Mass non-resettable totalizer	double
245	Arm 3 product 1 IV non-resettable totalizer	double
249	Arm 3 product 1 GV non-resettable totalizer	double
253	Arm 3 product 1 GST non-resettable totalizer	double
257	Arm 3 product 1 GSV non-resettable totalizer	double
261	Arm 3 product 1 Mass non-resettable totalizer	double
265	Arm 3 product 2 IV non-resettable totalizer	double

Modbus Address	Description	Data Type
269	Arm 3 product 2 GV non-resettable totalizer	double
273	Arm 3 product 2 GST non-resettable totalizer	double
277	Arm 3 product 2 GSV non-resettable totalizer	double
281	Arm 2 product 2 mass non-resettable totalizer	double
285	Arm 3 product 3 IV non-resettable totalizer	double
289	Arm 3 product 3 GV non-resettable totalizer	double
293	Arm 3 product 3 GST non-resettable totalizer	double
297	Arm 3 product 3 GSV non-resettable totalizer	double
301	Arm 3 product 3 mass non-resettable totalizer	double
305	Arm 3 product 4 IV non-resettable totalizer	double
309	Arm 3 product 4 GV non-resettable totalizer	double
313	Arm 3 product 4 GST non-resettable totalizer	double
317	Arm 3 product 4 GSV non-resettable totalizer	double
321	Arm 3 product 4 mass non-resettable totalizer	double
325	Arm 3 product 5 IV non-resettable totalizer	double
329	Arm 3 product 5 GV non-resettable totalizer	double
333	Arm 3 product 5 GST non-resettable totalizer	double
337	Arm 3 product 5 GSV non-resettable totalizer	double
341	Arm 3 product 5 mass non-resettable totalizer	double
345	Arm 3 product 6 IV non-resettable totalizer	double
349	Arm 3 product 6 GV non-resettable totalizer	double
353	Arm 3 product 6 GST non-resettable totalizer	double
357	Arm 3 product 6 GSV non-resettable totalizer	double
361	Arm 3 product 6 mass non-resettable totalizer	double
365	Arm 4 product 1 IV non-resettable totalizer	double
369	Arm 4 product 1 GV non-resettable totalizer	double
373	Arm 4 product 1 GST non-resettable totalizer	double
377	Arm 4 product 1 GSV non-resettable totalizer	double
381	Arm 4 product 1 mass non-resettable totalizer	double
385	Arm 4 product 2 IV non-resettable totalizer	double
389	Arm 4 product 2 GV non-resettable totalizer	double
393	Arm 4 product 2 GST non-resettable totalizer	double
397	Arm 4 product 2 GSV non-resettable totalizer	double
401	Arm 4 product 2 mass non-resettable totalizer	double
405	Arm 4 product 3 IV non-resettable totalizer	double
409	Arm 4 product 3 GV non-resettable totalizer	double
413	Arm 4 product 3 GST non-resettable totalizer	double
417	Arm 4 product 3 GSV non-resettable totalizer	double
421	Arm 4 product 3 Mass non-resettable totalizer	double
425	Arm 4 product 4 IV non-resettable totalizer	double
429	Arm 4 product 4 GV non-resettable totalizer	double
433	Arm 4 product 4 GST non-resettable totalizer	double
437	Arm 4 product 4 GSV non-resettable totalizer	double
441	Arm 4 product 4 mass non-resettable totalizer	double

Modbus Address	Description	Data Type
445	Arm 4 product 5 IV non-resettable totalizer	double
449	Arm 4 product 5 GV non-resettable totalizer	double
453	Arm 4 product 5 GST non-resettable totalizer	double
457	Arm 4 product 5 GSV non-resettable totalizer	double
461	Arm 4 product 5 mass non-resettable totalizer	double
465	Arm 4 product 6 IV non-resettable totalizer	double
469	Arm 4 product 6 GV non-resettable totalizer	double
473	Arm 4 product 6 GST non-resettable totalizer	double
477	Arm 4 product 6 GSV non-resettable totalizer	double
481	Arm 4 product 6 mass non-resettable totalizer	double
485	Arm 5 product 1 IV non-resettable totalizer	double
489	Arm 5 product 1 GV non-resettable totalizer	double
493	Arm 5 product 1 GST non-resettable totalizer	double
497	Arm 5 product 1 GSV non-resettable totalizer	double
501	Arm 5 product 1 mass non-resettable totalizer	double
505	Arm 5 product 2 IV non-resettable totalizer	double
509	Arm 5 product 2 GV non-resettable totalizer	double
513	Arm 5 product 2 GST non-resettable totalizer	double
517	Arm 5 product 2 GSV non-resettable totalizer	double
521	Arm 5 product 2 mass non-resettable totalizer	double
525	Arm 5 product 3 IV non-resettable totalizer	double
529	Arm 5 product 3 non-resettable totalizer	double
533	Arm 5 product 3 GST non-resettable totalizer	double
537	Arm 5 product 3 GSV non-resettable totalizer	double
541	Arm 5 product 3 mass non-resettable totalizer	double
545	Arm 5 product 4 IV non-resettable totalizer	double
549	Arm 5 product 4 GV non-resettable totalizer	double
553	Arm 5 product 4 GST non-resettable totalizer	double
557	Arm 5 product 4 GSV non-resettable totalizer	double
561	Arm 5 product 4 mass non-resettable totalizer	double
565	Arm 5 product 5 IV non-resettable totalizer	double
569	Arm 5 product 5 GV non-resettable totalizer	double
573	Arm 5 product 5 GST non-resettable totalizer	double
577	Arm 5 product 5 GSV non-resettable totalizer	double
581	Arm 5 product 5 mass non-resettable totalizer	double
585	Arm 5 product 6 IV non-resettable totalizer	double
589	Arm 5 product 6 GV non-resettable totalizer	double
593	Arm 5 product 6 GST non-resettable totalizer	double
597	Arm 5 product 6 GSV non-resettable totalizer	double
601	Arm 5 product 6 mass non-resettable totalizer	double
605	Arm 6 product 1 IV non-resettable totalizer	double
609	Arm 6 product 1 GV non-resettable totalizer	double
613	Arm 6 product 1 GST non-resettable totalizer	double
617	Arm 6 product 1 GSV non-resettable totalizer	double

Modbus Address	Description	Data Type
621	Arm 6 product 1 mass non-resettable totalizer	double
625	Arm 6 product 2 IV non-resettable totalizer	double
629	Arm 6 product 2 GV non-resettable totalizer	double
633	Arm 6 product 2 GST non-resettable totalizer	double
637	Arm 6 product 2 GSV non-resettable totalizer	double
641	Arm 6 product 2 mass non-resettable totalizer	double
645	Arm 6 product 3 IV non-resettable totalizer	double
649	Arm 6 product 3 GV non-resettable totalizer	double
653	Arm 6 product 3 GST non-resettable totalizer	double
657	Arm 6 product 3 GSV non-resettable totalizer	double
661	Arm 6 product 3 mass non-resettable totalizer	double
665	Arm 6 product 4 IV non-resettable totalizer	double
669	Arm 6 product 4 GV non-resettable totalizer	double
673	Arm 6 product 4 GST non-resettable totalizer	double
677	Arm 6 product 4 GSV non-resettable totalizer	double
681	Arm 6 product 4 mass non-resettable totalizer	double
685	Arm 6 product 5 IV non-resettable totalizer	double
689	Arm 6 product 5 GV non-resettable totalizer	double
693	Arm 6 product 5 GST non-resettable totalizer	double
697	Arm 6 product 5 GSV non-resettable totalizer	double
701	Arm 6 product 5 mass non-resettable totalizer	double
705	Arm 6 product 6 IV non-resettable totalizer	double
709	Arm 6 product 6 GV non-resettable totalizer	double
713	Arm 6 product 6 GST non-resettable totalizer	double
717	Arm 6 product 6 GSV non-resettable totalizer	double
721	Arm 6 product 6 mass non-resettable totalizer	double

4.9.5 Response Data for Command Code 12

User registers:

Modbus Address	Description	Data Type
5	Boolean variable 1	unsigned char
6	Boolean variable 2	unsigned char
7	Boolean variable 3	unsigned char
8	Boolean variable 4	unsigned char
9	Boolean variable 5	unsigned char
10	Floating point 1	float
12	Floating point 2	float
14	Floating point 3	float
16	Floating point 4	float
18	Floating point 5	float
20	Text 1	text
36	Text 2	text
52	Text 3	text
68	Text 4	text
84	Text 5	text
100	Text 6	text
116	Text 7	text
132	Text 8	text

4.10 0x0405 Search Transaction Log

This service searches the transaction log for the latest entry, the oldest entry still available in memory, or the most recent entry that falls before a given date and time. The service returns the sequence number of the entry.

4.10.1 Variation #1 Command Data

Returns most recent record number:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0405	unsigned	Router word, "search transaction log" service
2	1	unsigned	Sub-command (1=variation 1)

4.10.2 Variation #2 Command Data

Returns oldest available record number:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0405	unsigned	Router word, "search transaction log" service
2	2	unsigned	Sub-command (2=variation 2)

4.10.3 Variation #3 Command Data

Searches on specified date and time:

Modbus Address	Example Data	Data Type	Description
0	20	unsigned	Size of extended services packet
1	0x0405	unsigned	Router word, "search transaction log" service
2	3	unsigned	Sub-command (3=variation 3)
3	2000	unsigned	Log year
4	4	unsigned	Log month
5	12	unsigned	Log day
6	0	unsigned	(reserved)
7	0	unsigned	Log seconds
8	0	unsigned	Log minutes
9	14	unsigned	Log hours
10	0	unsigned	(reserved)

4.10.4 Variations #1-3 Response Data

Modbus Address	Example Data	Data Type	Description
0	8	unsigned	Size of extended services packet
1	0x8405	unsigned	Router word, "search transaction log" service
2	0x0000	unsigned	Standard response code
3 - 4	93543	unsigned long	Sequence number of log entry

4.11 0x0406 Read Audit Trail Entry

This service reads a record from the audit trail log stored in nonvolatile storage.

4.11.1 Command Data for Read Audit Trail Entry

Modbus Address	Example Data	Data Type	Description
0	6	unsigned	Size of extended services packet
1	0x0406	unsigned	Router word, "read audit log" service
2 - 3	87921	unsigned long	Sequence number of event

4.11.2 Response Data for Read Audit Trail Entry

Modbus Address	Example Data	Data Type	Description
0	114	unsigned	Size of extended services packet
1	0x8406	unsigned	Router word, "read audit log" service
2	0x0000	unsigned	Standard response code
3 - 38	"SY:708 3 Remote Control 5 Poll & Program"	text	Message text; may be null terminated (\0)
39	0	unsigned	(reserved)
40	2000	unsigned	Log year
41	1	unsigned	Log month
42	25	unsigned	Log day
43	0	unsigned	(reserved)
44	47	unsigned	Log seconds
45	21	unsigned	Log minutes
46	16	unsigned	Log hours
47	0	unsigned	(reserved)
48	0	unsigned	(reserved)

4.12 0x0407 Search Audit Trail Log

This service searches the audit trail log for the latest entry, the oldest entry still available in memory, or the most recent entry that falls before a given date and time. The service returns the sequence number of the entry.

4.12.1 Variation #1 Command Data

Returns most recent record number:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0407	unsigned	Router word, "search audit log" service
2	1	unsigned	Sub-command (1=variation 1)

4.12.2 Variation #2 Command Data

Returns oldest available record number:

Modbus Address	Example Data	Data Type	Description
0	4	unsigned	Size of extended services packet
1	0x0407	unsigned	Router word, "search audit log" service
2	2	unsigned	Sub-command (2=variation 2)

4.12.3 Variation #3 Command Data

Searches on specified date and time:

Modbus Address	Example Data	Data Type	Description
0	20	unsigned	Size of extended services packet
1	0x0407	unsigned	Router word, "search audit log" service
2	3	unsigned	Sub-command (3=variation 3)
3	2000	unsigned	Log year
4	4	unsigned	Log month
5	12	unsigned	Log day
6	0	unsigned	(reserved)
7	0	unsigned	Log seconds
8	0	unsigned	Log minutes
9	14	unsigned	Log hours
10	0	unsigned	(reserved)

4.12.4 Variations #1-3 Response Data

Modbus Address	Example Data	Data Type	Description
0	8	unsigned	Size of extended services packet
1	0x8407	unsigned	Router word, "search audit log" service
2	0x0000	unsigned	Standard response code
3 - 4	93543	unsigned long	Sequence number of log entry

5 Related Publications

The following literature can be obtained from TechnipFMC Measurement Solutions Literature Fulfillment at Measurement.Fulfillment@TechnipFMC.com or online at info.smithmeter.com/literature/online_index.html.

When requesting literature, please reference the appropriate title and document number, as follows:

- AccuLoad IV Installation and Maintenance Manual (MN06201)
- AccuLoad IV Operator Reference Manual (MN06200)
- AccuLoad IV Smith Communications Manual (MN06204L)
- AccuLoad IV Parts List (PO06200)
- AccuLoad IV Specifications (SS06200)
- AccuLoad Calculations Technical Paper (TP06004)

Appendix A: Modbus Communications Primer

The AccuLoad IV Modbus interface is designed to conform to a subset of the "Modicon Modbus Protocol Reference Guide" PI-MBUS-300 Rev. D (Modicon, Inc., Industrial Automation Systems). Modbus can be implemented on various transmission mediums (such as RS-232 or RS-485 communication ports). Transmission of data is serial and asynchronous.

The Host Message

The host transmits a message on the communications line that represents a specific query or command. The address specifies which server device is to act on the message. The function in the query tells the addressed server device what kind of action to perform. The register word specifies what particular internal state/value of the server is of interest to the host. The data bytes contain any additional information that the server will need to perform the function. For example, function code 03 will query the server to read holding registers and respond with their contents. The register field must contain information telling the server which registers to read and the data field specifies how many registers to read. The error check or CRC (cyclical redundancy check) field enables the server to validate the integrity of the message contents.

The Response

If the server makes a normal response, the function byte in the response is an echo of the function in the query. The data bytes contain the data collected by the server, such as register values or status. If an error occurs, the function code is modified to indicate that the response is an error response, and the data bytes contain a code that describes the error. The error check field allows the client to confirm that the message contents are valid.

A.1 RTU Framing

Every Modbus message begins with a silent interval of at least 3.5 character times. Multiply the character times by the current network baud rate to determine the length of the silent interval (see T1-T2-T3-T4 in the figure below). Next, the AccuLoad address field is transmitted.

Characters for all fields are transmitted as binary bytes. In this manual, characters are represented by hexadecimal 0-9, A-F. All networked devices constantly monitor the network bus. This monitoring occurs even during silent intervals. As each AccuLoad receives the first field which is the address field, it decodes it to determine if it is the AccuLoad being addressed.

A second silent interval of at least 3.5 character times follows the last transmitted character of each message, after which a new message can begin. The new message must be transmitted as a continuous stream, with no silent interval in excess of 3.5 character times. If an excessively long silent interval occurs before completion of the frame, the receiving AccuLoad will disregard the entire incomplete message and wait for the address field of the next new message.

If a silent interval is less than 3.5 character times, the receiving AccuLoad will be unable to recognize it as the start of a new message and will attempt to read it as a part of the prior message. These combined messages will result in an invalid value in the final CRC field, and an error will result. A typical message frame is shown below.

3.5 Character Time Delay	Address	Function	Register	Data	CRC	3.5 Character Time Delay
	1 byte	1 byte	2 bytes	n bytes	2 bytes	

The starting 3.5 character-time ending delay for one message may be the same actual delay as the starting 3.5 character time for the next message (it is not necessary for the client to delay twice between messages as long as the duration exceeds the specified delay).

A.1.1 How Characters are Transmitted Serially

When messages are transmitted on standard Modbus serial networks, each character or byte is sent in this order from left to right:

With Parity Checking (8-bit word, 1 stop)										
Start	1	2	3	4	5	6	7	8	Par	Stop

With Parity Checking (8-bit word, 2 stop)										
Start	1	2	3	4	5	6	7	8	Stop	Stop

A.1.2 Data Addresses in Modbus Messages

All data addresses in Modbus messages are referenced to zero; the first occurrence of a data item is addressed as item number zero.

A.2 Modbus Functions

The following Modbus functions have been implemented in the AccuLoad IV:

Table 25: Modbus Functions

Code	Function	Description
01	Read Relay Status	Reads the binary data from the (read/write) set of variables
02	Read Input Status	Reads the binary data from the "inputs" (read only) set of variables
03	Read Integer Registers (Read/Write Register Set)	Retrieves the current data from the requested registers
04	Read Integer Registers (Read Only Register Set)	Retrieves the current data from the requested registers
05	Force Single Relay	Changes the state of a binary (read/write)
06	Write (Preset) Single Register	Places a specific value into a (read/write) register
08	Loop Back Diagnostic Text	Diagnostic test message sent to the AccuLoad to evaluate communications processing Note: Only the return query data diagnostic code is supported.
15	Force Multiple Relays	Changes the state of multiple binary (read/write)
16	Write (Preset) Multiple Registers	Places specific values into a series of consecutive (read/write) registers

A.3 Client/Server Communications

The client communicates with the AccuLoad by sending messages containing function codes. Function codes indicate the actions the AccuLoad is to perform.

The AccuLoad's response to the client uses the function code field to report on the status of the task it was assigned. The two possible reports are:

1. A normal, error-free response
2. An exception response, indicating an error

A normal response repeats the original function code. An exception response returns a code that corresponds to the original function code, with its most-significant bit set to a logic 1.

For example, a client directs an AccuLoad to read a group of holding registers by sending the following function code:

0000 0011 (Hexadecimal 03)

If the AccuLoad completes the action without error, its response echoes the original command. If an error occurs, the AccuLoad returns the following message:

1000 0011 (Hexadecimal 83)

The AccuLoad augments its exception response by adding a code in the data field that indicates what type of error occurred. The exception response is handled according to the parameters of the application program controlling the client device.

For example, if the relay address is absent in the AccuLoad device, the AccuLoad will return the exception response with the exception code shown (02). This response indicates an invalid data address for the AccuLoad.

A listing of the exception codes appears below:

Code	Name	Meaning
01	Illegal Function	The function code received in the query is not an allowable action for the server; if a Poll Program Complete command was issued, this code indicates that no program function preceded it
02	Illegal Data Address	The data address received in the query is not an allowable address for the AccuLoad
03	Illegal Data Value	A value contained in the query data field is not an allowable value for the AccuLoad
04	Command Error	An unrecoverable error occurred while the AccuLoad was attempting to perform the requested action

A.4 Contents of the Data Field

The data field consists of sets of two hexadecimal digits, in the range of 00 to FF hexadecimal.

The AccuLoad reads the data field sent by the client to perform the actions indicated by the function code. The data field contains information such as discrete and register addresses, the number of items to be handled, and the count of actual data bytes in the field.

If, for example, the client directs an AccuLoad to read a group of holding registers (function code 03), the data field sent by the client must also indicate the starting register and the number of registers to be read. If the client writes to a group of registers in the server (function code 10 hexadecimal), the data field sent by the client must also indicate the starting register, the number of registers to be written, the count of data bytes to follow in the data field, and the data to be written into the registers.

Assuming that no error in communication interferes, the data field of a response from a server to a client contains the requested data. If an error does occur, the field contains an exception code that the application controlling the client can use to determine the next action to be taken.

A.4.1 Beginning Register

This register identifies the beginning register from which the client is requesting information. This two-byte field lists the most significant digit first and the least significant digit last.

A.4.2 Number of Requested Registers

This field identifies the number of consecutive registers from which the client is requesting information. This two-byte field lists the most significant digit first and the least significant digit last. The response is limited to 250 bytes of information.

A.4.3 Error Check (CRC16)

This field allows the AccuLoad IV and the supervisory system to check for errors in the transmission of commands and responses. Electrical noise or other interference may cause changes in transmitted data. The capacity to check for errors prevents the receiving device from responding to a message that has changed.

Error-checking in RTU mode is built on the Cyclical Redundancy Check (CRC) method. The entire message is subject to scrutiny by the CRC field, and the CRC is applied regardless of any other parity check method that might be in effect.

The CRC consists of a two-byte field containing a 16-bit binary value. The transmitting device calculates the CRC value and adds the CRC to the message. The receiving device then recalculates the CRC when the message is received, and compares the first value with the second. An error results when the two message values are unequal.

The CRC is initiated by pre-loading a 16-bit register to all 1s. Successive 8-bit bytes of the message are then applied to the current contents of the register. The CRC is generated only by the eight bits of data in each character. Start and stop bits, and the parity bit if one is used, are not taken into account.

When the CRC is generated, each 8-bit character is exclusive ORed with the register contents. The result is then shifted toward the least significant bit (LSB), and a zero added to the most significant bit (MSB) position. The LSB is extracted and examined. Assuming the LSB was a 1, the register is then exclusive ORed with a preset, fixed value. If the LSB was a 0, there will be no exclusive OR.

The process consists of eight shifts. After the eighth and final shift, the next 8-bit byte is exclusive ORed with the register's current value. The process is then repeated for an additional eight shifts. The final content of the register, after all the bytes of the message have been applied, is the CRC value.

A.5 Placing the CRC into the Message

When the 16-bit CRC (2 8-bit bytes) is transmitted in the message, the low-order byte will be transmitted first, followed by the high-order byte. For example, if the CRC value is 1241 hex (0001 0010 0100 0001):

Addr	Func	Data Count	Data	Data	Data	Data	CRC Lo	CRC Hi
							41	12

A.5.1 Field Contents in Modbus Messages

Examples of a Modbus query message and normal response are shown in the tables on the following page. The field contents in both examples are displayed in hexadecimal.

In this example, the client sends a Read Holding Registers request to AccuLoad address 06. The AccuLoad is specifically directed to return data from three holding registers, starting with address 0107 (006B hex).

As is the case in any normal response, the AccuLoad first echoes the function code sent by the client. The AccuLoad then transmits the byte count field, indicating the number of 8-bit data items being returned. Finally, the AccuLoad returns the 8-bit bytes containing the requested data.

A.5.1.1 How to Use the Byte Count Field

When constructing responses in buffers, use a byte count value that equals the count of 8-bit bytes in the message data. The value is exclusive of all other field contents, including the byte count field. The AccuLoad response example illustrates a typical byte count field in a normal response.

Client Query		
Field Name	**Example (Hex)**	**RTU 8-Bit Field**
Header		None
AccuLoad Address	06	0000 0110
Function	03	0000 0011
Starting Address Hi	00	0000 0000
Starting Address Lo	6B	0110 1011
Number of Registers Hi	00	0000 0000
Number of Registers Lo	03	0000 0011
Error Check		CRC (16 bits)
	Total Bytes:	8

AccuLoad Response		
Field Name	**Example (Hex)**	**RTU 8-Bit Field**
Header		None
AccuLoad Address	06	0000 0110
Function	03	0000 0011
Byte Count	06	0000 0110
Data Hi	02	0000 0010
Data Lo	2B	0010 1011
Data Hi	00	0000 0000
Data Lo	00	0000 0000
Data Hi	00	0000 0000
Data Lo	63	0110 0011
Error Check		CRC (16 bits)
	Total Bytes:	11

The AccuLoad monitors the amount of time between the receipt of characters. If three and one-half character times elapse without the AccuLoad seeing a new character or the end of a frame, the message is flushed and the next characters received will be viewed as an address. If the address is for that AccuLoad, it will respond. If the address is not for that AccuLoad, the message will be flushed and it will look for the next message.

A.5.2 Address

The address is the first field in the frame and consists of one byte (eight bits) of information. The address is the unique identification of the AccuLoad (server) that is to receive the message that is sent by the supervisory system (client). Each AccuLoad address must be unique so that only the addressed server will respond to a query. The address is also part of the response message sent back to the client from the AccuLoad when data is requested. By returning the address as part of the response, the client can tell which AccuLoad the data is coming from.

A.6 Query Responses

The first two fields of the response to the read only message are identical to the command. The AccuLoad returns the address and the function code that was transmitted to the unit. The next field is the byte count.

A.6.1 Byte Count

The byte count is sent to the client (supervisory system) indicating how much data is being sent from the AccuLoad. In the example shown, the command requested data from these registers and each register contains two bytes of data.

A.6.2 Data Register

Each of the data registers of unsigned characters contains two bytes of data. The response message returns the data with the most significant byte of data first and the least significant byte second. Data can be requested and returned from a number of registers with a single interrogation message. The limit on the amount of data returned from the AccuLoad to the client is 256 bytes. The data lengths for the data types currently used by the AccuLoad are as follows:

Data Length	
Type	Binary
Double	8 bytes
Integer	2 bytes
Long Integer	4 bytes
Text String	Variable length
Character	2 bytes (high order byte set to zero)
CRC-16	2 bytes
Float	4 bytes
Unsigned Integer	2 bytes
Unsigned long	4 bytes
Unsigned character	2 bytes (high order byte set to zero)

The error-checking sequence is the same as described in the paragraph under Read Only Message.

A.7 Function 01—Read Relay Status

A.7.1 Description of Read Relay Status

Reads the ON/OFF status of discrete variables in the AccuLoad. The maximum number of "coils" per response is 256 in the AccuLoad IV.

A.7.2 Query of Read Relay Status

The query message specifies the starting register and quantity of registers to be read.

There are now no variables to read from this group. If there were, this is an example of a request to read variables 20 through 56 from AccuLoad device 17:

Query	
Field Name	**Example (Hex)**
AccuLoad Address	0x11
Function	0x01
Starting Address Hi	0x00
Starting Address Lo	0x13
No. of Points Hi	0x00
No. of Points Lo	0x25
Error Check (CRC)	(calculated)

A.7.3 Response of Read Relay Status

A response message consists of a relay status packed as one relay per bit of the data field. Status is indicated by means of the following code: 0 = OFF; 1 = ON. The first data byte is contained in the LSB, and specifies the relay addressed in the query. All other relays follow from "low order to high order" in subsequent bytes.

The returned relay quantity must be a multiple of eight; otherwise, it will be padded with zeros toward the high order end of the byte. The assembled bytes of data are specified in the byte count field.

An example of a response to the preceding query appears below:

Response	
Field Name	**Example (Hex)**
AccuLoad Address	0x11
Function	0x01
Byte Count	0x05
Data (Relays 27-20)	0xCD
Data (Relays 35-28)	0x6B
Data (Relays 43-36)	0xB2
Data (Relays 51-44)	0x0E
Data (Relays 56-52)	0x1B
Error Check (CRC)	(calculated)

The status of relays 27 through 20 is shown as the byte value CD hex, or binary 1100 1101. Relay 27 is the MSB of the byte, and relay 20 is the LSB. The status of relays 27 through 20 is expressed from left to right as ON-ON-OFF-OFF-ON-ON-OFF-ON.

Bits within a byte are shown with the MSB to the left and the LSB to the right; therefore, the relays in the first byte are “27 through 20,” from left to right. Relays “35 through 28" are contained in the next byte, again from left to right. As the bits are transmitted serially, they flow from LSB to MSB (for example, 20 through 27, 28 through 35, and so on).

In the last data byte, the status of relays 56 through 52 is shown as the byte value 1B hex, or binary 0001 1011. Relay 56 is in the fourth bit position from the left, and relay 52 is the LSB of this byte. The status of relays 56 through 52 is expressed as ON-ON-OFF-ON-ON. The three remaining bits toward the high order end are padded with zeros.

A.8 Function 02—Read Input Status

A.8.1 Description of Read Input Status

Reads the ON/OFF status of discrete "inputs" (read only binary references) in the AccuLoad. The maximum number of parameters supported by AccuLoad IV is limited to 256 per query.

A.8.2 Query of Read Input Status

The query message specifies the starting "input" and quantity of "inputs" to be read. "Inputs" are addressed starting at zero: inputs 1 through 16 are addressed as 0 through 15.

An example of a request to read the states of inputs 1024 to 1033 from AccuLoad 17 is shown below:

Query	
Field Name	**Example (Hex)**
AccuLoad Address	0x11
Function	0x02
Starting Address Hi	0x00
Starting Address Lo	0xC4
No. of Points Hi	0x00
No. of Points Lo	0xOA
Error Check (CRC)	(calculated)

A.8.3 Response of Read Input Status

The input status is packed in the response message as one input per bit of the data field. Status is indicated as 0 = OFF; 1 = ON. The input addressed in the query appears in the LSB of the first data byte. The other inputs follow toward the high order end of this byte, and from low order to high order in all subsequent bytes.

The returned input quantity must be a multiple of eight; otherwise, the remaining bits in the final data byte will be padded with zeros toward the high order end of the byte. The quantity of complete bytes of data is indicated in the byte count field.

An example of a response to the preceding query appears below:

Response	
Field Name	**Example (Hex)**
AccuLoad Address	0x11
Function	0x02
Byte Count	0x02
Data (Inputs 1031-1024)	0xAC
Data (Inputs 1033-1032)	0x01
Error Check (CRC)	(calculated)

The status of inputs 1031 through 1024 is shown as the byte value AC hex, or binary 1010 1100. Input 1031 is the MSB of this byte and input 1024 is the LSB. The status of inputs 1031 through 1024 is expressed as ON-OFF-ON-OFF-ON-ON-OFF-OFF, from left to right.

The status of inputs 1033 through 1032 are shown as the byte value 01 hex, or binary 0000 0001. Input 1033 is in the seventh bit position from the left and input 1032 is the LSB. The status of inputs 1033 through 1032 is OFF-ON. The six remaining bits toward the high order end are padded with zeros, since the returned input quantity must be a multiple of eight.

A.9 Function 03—Read Holding Registers

A.9.1 Description of Read Holding Registers

Reads the binary contents of holding registers (read/ write registers).

A.9.2 Query Read Holding Registers

The query message specifies the starting register and quantity of registers to be read. Registers are addressed starting at zero.

An example of a request to read registers 107 through 109 from AccuLoad 17 is shown below:

Query	
Field Name	**Example (Hex)**
AccuLoad Address	0x11
Function	0x03
Starting Address Hi	0x00
Starting Address Lo	0x6B
No. of Points Hi	0x00
No. of Points Lo	0x03
Error Check (CRC)	(calculated)

A.9.3 Response Read Holding Registers

Each register data in the response message contains two bytes. The binary contents are right justified within each byte. Within each register, the first byte contains the high order bits and the second byte contains the low order bits.

An example of a response to the preceding query is shown below:

Response	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	03
Byte Count	06
Data Hi (Register 107)	02
Data Lo (Register 107)	2B
Data Hi (Register 108)	00
Data Lo (Register 108)	00
Data Hi (Register 109)	00
Data Lo (Register 109)	64
Error Check (CRC)	--

A.10 Function 04—Read Input Registers

A.10.1 Description of Read Input Registers

This function reads the binary contents of "input registers" in the AccuLoad. These are "read-only" values; they cannot be written.

A.10.2 Query of Read Input Registers

The query message specifies the starting register and quantity of registers to be read. Registers are addressed starting at zero.

An example of a request to read register 8 from AccuLoad 17 appears below:

Query	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	04
Starting Address Hi	00
Starting Address Lo	08
No. of Points Hi	00
No. of Points Lo	01
Error Check (CRC)	--

A.10.3 Response of Read Input Registers

Each register data in the response message contains two bytes. The binary contents are right justified within each byte. Within each register, the first byte contains the high order bits and the Second byte contains the low order bits.

An example of a response to the preceding query appears below:

Response	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	04
Byte Count	02
Data Hi (Register 30009)	00
Data Lo (Register 30009)	0A
Error Check (CRC)	--

A.11 Function 05—Force Single Relay

A.11.1 Description of Force Single Relay

Forces a single relay either ON or OFF.

A.11.2 Query of Force Single Relay

The query message specifies the relay reference to be forced. Relays are addressed starting at zero.

A constant in the query data field indicates the required ON/OFF state. A value of FF 00 hex directs the relay to be ON. A value of 00 00 directs the relay to be OFF. No other value is valid, nor will it affect the relay.

An example of a request to force relay 150 ON in AccuLoad 17 appears below: (Reset User Alarm #9).

Query	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	05
Relay Address Hi	00
Relay Address Lo	96
Force Data Hi	FF
Force Data Lo	00
Error Check (CRC)	--

A.11.3 Response of Force Single Relay

An echo of the query, returned after the relay status has been forced, indicates a normal response.

An example of a response to the preceding query appears below:

Response	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	05
Relay Address Hi	00
Relay Address Lo	96
Force Data Hi	FF
Force Data Lo	00
Error Check (CRC)	--

A.12 Function 06—Preset Single Register

A.12.1 Description of Preset Single Register

Presets a value into a single holding register.

A.12.2 Query of Preset Single Register

The query message specifies the register reference to be preset. Registers are addressed starting at zero. The requested preset value is specified in the query data field.

An example of a request to preset register 1 to 0x0003 (hex) in AccuLoad 17 appears below:

Query	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (CRC)	--

A.12.3 Response of Preset Single Register

An echo of the query, returned after the register contents have been preset, is a normal response.

An example of a response to the preceding query appears below:

Response	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (CRC)	--

A.13 Function 08—Diagnostics

A.13.1 Description of Diagnostics

Modbus function 08 is a diagnostic test that checks the client/AccuLoad communication system. A two- byte sub-function code field in the query defines the test to be performed. In a normal response, the AccuLoad echoes both the function code and sub-function code.

A two-byte data field is used in most of the tests. The data field contains control information or diagnostic data that is sent to the AccuLoad. In some tests, the AccuLoad returns diagnostic data in the data field of a normal response.

An example of a diagnostics query and response appears below. The query indicates the location of the function code, sub-function code, and the data field within the messages.

A list of sub-function codes supported by the controllers is shown on the following page. Each sub-function code is listed, along with an example of the data field content that applies to that diagnostic.

A.13.2 Query of Diagnostics

An example of a Return Query Data request to server device 17 appears below. This request involves a sub-function code of zero 0x0000 (hex) in the two- byte field. The data to be returned is sent in the two-byte data field 0xA537 (hex).

Query	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	08
Sub-function Hi	00
Sub-function Lo	00
Data Hi	A5
Data Lo	37
Error Check (CRC)	--

A.13.3 Response of Diagnostics

A loop-back of data is the normal response to a Return Query Data request. The function and sub-function codes are also echoed.

Response	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	08
Sub-function Hi	00
Sub-function Lo	00
Data Hi	A5
Data Lo	37
Error Check (CRC)	--

A.13.4 Subfunction "00", Return Query Data

A normal response to the data passed in the query data field is an echo of the original message.

Sub-function	Data Field (Query)	Data Field (Response)
0x0000	Any	Echo Query Data

A.14 Function 15 (0F Hex) Force Multiple Relays

A.14.1 Description

Forces each relay in a sequence of relays to either ON or OFF. The maximum number of parameters by AccuLoad IV is limited to 256 per query.

A.14.2 Query

The query message specifies the relay references to be forced. Relays are addressed starting at zero; thus, relay 1 is addressed as 0.

The contents of the query data field specify whether a state is ON or OFF. A logical “1” in a bit position of the field requests the corresponding relay to be ON. A logical “0” requests that the relay be OFF.

An example of a request to force a series of ten relays starting at address 15, or OF hex in AccuLoad 17, appears below.

The query data content consists of two bytes: CD 01 hex (1100 1101 0000 0001 binary). The binary bits correspond to the relays as shown below:

Bit:	1	1	0	0	1	1	0	1		0	0	0	0	0	0	0	1
Relay:	22	21	20	19	18	17	16	15		-	-	-	-	-	-	24	23

The first byte transmitted (CD hex) addresses relays 22 through 15, with the least significant bit corresponding to the lowest relay (15) in this set.

The next byte transmitted (01 hex) addresses relays 24 to 23, with the least significant bit corresponding to the lowest relay (23) in this set. Unused bits in the last data byte are padded with zeros.

Query	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	0F
Relay Address Hi	00
Relay Address Lo	OF
Quantity of Relays Hi	00
Quantity of Relays Lo	0A
Byte Count	02
Force Data Hi (Relays 27-20)	CD
Force Data Lo (Relays 29-28)	01
Error Check (CRC)	--

A.14.3 Response

The normal response consists of the server address, function code, starting address, and number of relays forced.

An example of a response to the preceding query appears below:

Response	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	0F
Relay Address Hi	00
Relay Address Lo	OF
Quantity of Relays Hi	00
Quantity of Relays Lo	0A
Error Check (CRC)	--

A.15 Function 16 (10 Hex) Preset Multiple Registers

A.15.1 Description

Presets values into a sequence of holding registers.

A.15.2 Query

The query message specifies the register references to be preset. Registers are addressed beginning with zero.

An example of a request to preset two registers starting at 1 to 0x000A and 0x0102 (hex), in AccuLoad 17, appears below:

Query	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	10
Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Byte Count	04
Data Hi	00
Data Lo	0A
Data Hi	01
Data Lo	02
Error Check (CRC)	--

A.15.3 Response

A normal response consists of the server address, function code, starting address, and quantity of registers preset.

An example of a response to the preceding query appears below:

Response	
Field Name	**Example (Hex)**
AccuLoad Address	11
Function	10
Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Error Check (CRC)	--

A.16 Exception Responses

When a client device sends a query to an AccuLoad, there are three possible outcomes:

1. The AccuLoad receives the query with no communication errors, handles the query normally, and returns a normal response.
2. A communication error bars the AccuLoad from receiving the query, so no response is returned. The client program eventually processes a timeout condition for the query.
3. The AccuLoad receives the query without error, but returns no response. The client program eventually processes a timeout condition for the query.

Two fields in the exception response message differentiate it from a normal response:

Function Code Field

An AccuLoad normally echoes the function code of the original query in the function code field of the response. Because the values of all function codes are below 80 hexadecimal, all function codes have a most-significant bit (MSB) of 0. In an exception response, however, the server sets the MSB of the function code to 1. The value of the function code in an exception response is therefore 0x80 (hex) higher than the value for a normal response.

Accordingly, the application program controlling the client can quickly recognize the exception response and derive the exception code from the data field.

Data Field

A normal response consists of any data or statistics in the data field requested by the query. An exception response consists of an exception code in the data field. The code indicates the AccuLoad condition that caused the exception.

An example of a client query and AccuLoad exception response is shown in the table below. The field examples are given in hexadecimal.

Query		
Byte	**Contents**	**Example**
1	AccuLoad Address	0A
2	Function	01
3	Starting Address Hi	28
4	Starting Address Lo	0A
5	No. of Relays Hi	00
6	No. of Relays Lo	01
7	CRC	--

Exception Response		
Byte	**Contents**	**Example**
1	AccuLoad Address	0A
2	Function	81
3	Exception Code	02
4	CRC	--

Here, the client addresses a query to AccuLoad 10. The function code (01) is for a Read Relay Status operation that requests the status of the relay at address 10250 (0x280A hex). The number of relays field (0001) specifies that only one relay is to be read.

This document takes precedence over and supersedes in their entirety all previous versions or revisions.

Bulletin MN06202L Issue/Rev. 0.0 (9/23)

TechnipFMC.com

TechnipFMC Corporate Headquarters
13460 Lockwood Road
Building S01
Houston, TX 77044 USA
+1 281.591.4000

USA Operations
1602 Wagner Avenue
Erie, PA 16510 USA
+1 814.898.5000

Germany Operations
Smith Meter GmbH
Regentstrasse 1
25474 Ellerbek, Germany
+49 4101 304.0