Seametrics

Modbus® Interface Description

ISO 9001 Certified Company

Introduction

Addresses and Values

Appendix: Reading and Writing

What is Modbus®?

Modbus® is an industry standard serial protocol for communication between industrial electronic devices and various control and display equipment, such as PLCs, SCADA systems, and panel meters. Modbus® was developed by Modicon®and released for use in 1979. It is an open protocol, available for use without royalties. It is typically used to set parameters or gather data from various instrumentation.

The official Modbus® specification can be found at: www.modbus.org/specs.php

A good source for an introduction to Modbus® is: www.simplymodbus.ca/FAQ.htm#Modbus

Modbus® on Seametrics iMAG 4700 and AG3000 Magmeters

Several iMAG 4700 and AG3000 models can be factory configured for Modbus® RTU communication, providing a half-duplex, isolated, RS485 serial communications port using the Modbus® messaging protocol. The interface contains an RS485 transceiver which implements a fail-safe receiver to assure that a mark condition will be sensed even when the cable is disconnected or undriven. Under this condition, a bias network is unnecessary for the iMAG 4700/AG3000 interface to function properly. It is assumed, however, that the master contains either a Fail Safe receiver or a bias network for proper operation of the RS485 network as whole. A useful reference on this subject is an Applications Note from Texas Instruments: www.ti.com/lit/an/slyt514/slyt514.pdf

For attachment to the RS485 network, the iMAG 4700/AG3000 comes equipped with a 6 or 8 conductor cable. For Modbus®, the following connections are used:

Isolated Ground	Grey or White Wire (White wire only if there are two white wires available)
A/TX[+]	Orange Wire
B/RX[-]	Blue Wire

The iMAG 4700/AG3000 Modbus® interface can work with network cable lengths up to 50 feet without termination. If a longer cable length is used, then DC or AC termination may be needed if the meter is at the end of the network. The 103848 Modbus option board has a 120-ohm termination resistor built into the board. The termination resistor is shipped in an inactive state and does not need to be used. To activate the resistor, move the jumper at JP1 from the storage pins 3 and 4, to the active pins 1 and 2. Only use one termination resistor per system, and only at the last device on the string.

Modbus Option Board Variations

As of spring 2021, a new Modbus option board has been released. To help identify which board you have, this new board has one jumper that will be located in the center of the exposed board. This is the 103848 low power Modbus option board, and will also have two small labels, one of which will note the 103848 part number.

The previous Modbus board was a part number 102151 and can be identified by the 20 pin (2 x 10) jumper array in one corner of the board.

Battery Power

Meters having the old 102151 Modbus option board must be configured for, and connected to, external power due to the power consumption of the Modbus board.

Meters using the 103848 Modbus option board can be configured in the factory to run off external power, or primarily off the internal battery.

BX, battery configured meters can run Modbus on the internal battery alone when equipped with the 103848 Modbus option board.

Modbus communications use a considerable amount of power, and battery life will be affected by the addresses polled and the polling frequency. To maximize battery life, keep the information polled and polling frequency to a minimum.

If running a BX, battery configured meter while using the 103848 Modbus option board, the sample rate (SAMP) should be set to at least 5 seconds. In this case it makes no sense to poll the Modbus any more frequently than every 5 seconds as the values will not have been updated. Consider how often you actually need the polled data and what data you need.

Polling a BX configured meter for 1 or 2 addresses at 5 seconds or longer should result in a battery life of about 1 year.

If you need to poll multiple addresses and/or poll frequently it is recommended to run the meter off external power so as not to deplete the battery quickly.

Any meter can be run off external power.

Wiring and Board Placement

Inside the housing, your Modbus configured meter will have a wiring harness with a 5-pin plug which will have 3 or 5 of the positions wired. This is your Modbus board connector and will plug into the Modbus board. Remote meters will have a 5-pin plug with 4 positions wired, this is not the Modbus plug. Your meter will be wired in the factory so you should never need to move this plug. If needed or desired in the field, there is also a 3-position terminal block on the 103848 Modbus board that can be used to wire in your Modbus communications. Use either the 5-pin plug, or the 3-position terminal block, not both. The 3-position terminal block is wired:

A/TX [+]	Position 3
B/RX [-]	Position 2
Isolated ground	Position 1

If you have multiple option boards in your meter, the Modbus board must be located on the top of the stack as you look at it from the rear of the display due to the size of components on the board.

Introduction to Function Codes

The Modbus® function codes implemented for the iMAG 4700/AG3000 consist of the following:

Function code *Hex*	*Decimal*	Description
0x03	03	Read Holding Register
0x04	04	Read Input Register
0x06	06	Write Single Register
0x10	16	Write Multiple Registers

For more details, see the appendix.

When quantities are written to the meter, they are also stored in an internal EEPROM device. As such, there is a limit as to how many times the device may be written. With this in mind, the write operation should be used cautiously and sparingly.

Introduction to Data Types

Data and parameters are stored in 16-bit registers (1 word), each register being 2 bytes.

The following data types are used in the iMAG 4700/AG3000 Modbus® interface:

Integer	16 bit unsigned value corresponding to 1 register
Float	32 bit IEEE floating point number corresponding to 2 sequential registers
String	16 ASCII characters correspond to 16 sequential registers

For more details, see the appendix.

Register Addressing

The physical register addresses for the iMAG 4700/AG3000 meters start numbering from zero (0-based addressing)—the first address is 0, the second is 1, etc. On the other hand, Modbus® protocol considers the first logical address to be 1, the second logical address to be 2, etc. (1-based addressing). For example, to view or set TUNITS, you have to read the physical address 1000 (0x03E8). Some programs and equipment when asked to read address 1000 (0x03E8) will read that physical address. Others however will read the logical address, which is actually the physical address 999 (0x03E7). With these programs and equipment you must add a one to the address—thus in this example you would request TUNITs at address 1001 (0x03E9).

Still other programs and equipment require the addition of such numbers as 40,000 or 400,000 to the address to indicate reading/writing to holding registers. These may or may not require the addition of one to the physical address, as well. Check with your system documentation to determine what style of register addressing is required.

All addressing in this document is based on the physical address (0-based addressing).

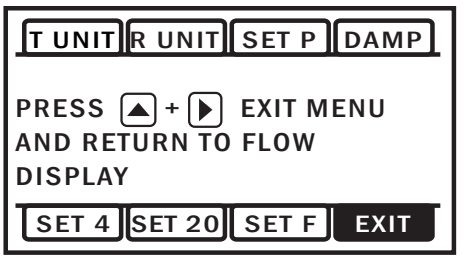
If you have trouble establishing communication, pick a single address such as TOTAL-FWD and poll the 0-based address. Then, poll that address plus 1 or that address -1.

Comm Settings and Modbus® Address

COMM Settings

When communicating between a controller and the meter, both the controller and the meter must be set to the same baud rate and parity. These values must be set using the on-meter menu system.

To access the COMM settings: From the main menu navigate to the EXIT tab and tap ▲ five times. This will bring you to a submenu. Navigate to the COMM tab and select your desired communication setting. **The default is 19,200 baud with no parity (19200 NONE)**. See the COMM entry in the Address Map table on the following pages for other options.

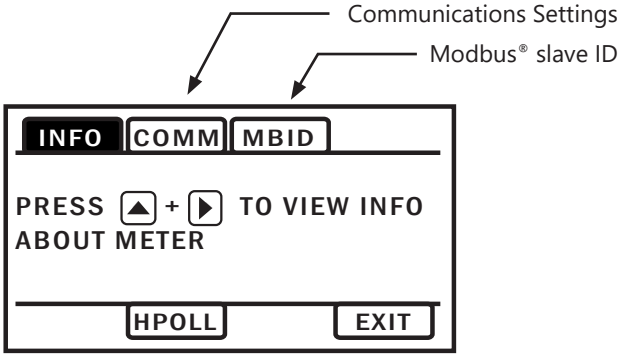


Main Menu

Modbus® Slave ID

Each slave device must have a unique Modbus® slave ID, which the controller will use to communicate with that particular device. **The iMAG 4700/AG3000 meters default to address 1**. If you need to change this, you can do so using the on-meter menu system.

To access the Modbus® slave ID setting (MBID): From the main menu navigate to the EXIT tab and tap ▲ five times. This will bring you to a submenu. Navigate to the MBID tab and set your desired address **(1–247)**.



Sub-Menu

Address Map

Item	Description	On Menu or Display	Via Modbus	R/W	Data Values	Address (0-Based)		Data Type
						Decimal	Hex	
TUNIT	View or select units for displaying the contents of the totalizer *Note: Values 7-21 may be minus 1 due to firmware change.*	Y	Y	R/W	0 Gallons 1 Gallons x 10 2 Gallons x 100 3 Gallons x 1000 4 Million Gallons 5 Cubic Feet 6 Cubic Feet X 100 7 Cubic Feet x 1000 8 Second Foot Day 9 Million Cubic Feet 10 Imperial Gallons 11 Imp. Gal. x 1000 12 Million Imp. Gal. 13 Liters 14 Kilo Liters 15 Mega Liters 16 Cubic Meters 17 Cubic Meters x 1000 18 Barrels 19 Acre Inches 20 Acre Feet 21 Fluid Ounces	1000	0x03E8	Integer
RUNIT	View or select units for displaying the current computed rate. (Need also to set a time base, RUNIT TIME, below.)	Y	Y	R/W	0 Gallons 1 Million Gallons 2 Cubic Feet 3 Imperial Gallons 4 Million Imp. Gal. 5 Liters 6 Mega Liters 7 Cubic Meters 8 Barrels 9 Fluid Ounces	1001	0x03E9	Integer
RUNIT TIME	View or set time base for RUNITs	Y	Y	R/W	0 Second 1 Minute 2 Hour 3 Day	1002	0x03EA	Integer
DAMP	View or select the amount of damping so as to stabilize the display of the rate. The selection corresponds to a window, in cycles, over which averaging takes place. These windows, in turn, are overlapped in time such that a new average value is computed every cycle.	Y	Y	R/W	0-99 cycles	1003	0x03EB	Integer
SETF	View or set the maximum frequency available at the high frequency output. This frequency corresponds to Q4, the maximum flow rate for a given pipe diameter. The frequency output is scaled between zero flow and Q4.	Y	Y	R/W	0 500 Hz 1 1 kHz 2 2 kHz 3 5 kHz 4 10 kHz	1004	0x03EC	Integer
SETP UNITS	View or select units for pulse output	Y	Y	R/W	0 Gallons 1 Cubic Feet 2 Imperial Gallons 3 Liters 4 Cubic Meters 5 Barrels 6 Fluid Ounces	1005	0x03ED	Integer
CLEAR BATCH TOTALS	Reset batch totals to zero	N	Y	W	01 Batch Forward 02 Batch Reverse 03 Both Fwd and Rev	1006	03EE	Integer
STATUS	Reports communication failure and empty pipe	Y	Y	R	00 00 EMPTY PIPE 00 01 COMM FAIL 00 02 FULL PIPE	5000	0x1388	Integer
SET4	View or set the rate corresponding to the 4 mA setting for the 4-20 mA loop. It specifies the lower limit of the rate that will result in controlling 4 mA in the loop.	Y	Y	R/W	00000.0 to 99999.9 (units specified by RUNIT)	7000	0x1B58	Float

Continued on next page

Address Map *(continued)*

Item	Description	On Menu or Display	Via Modbus	R/W	Data Values	Address		Data Type
						Decimal	Hex	
SET20	View or set the rate corresponding to the 20 mA setting for the 4-20 mA loop. It specifies the upper limit of the rate that will result in controlling 20 mA in the loop. The result of defining the SET4 and SET20 limits is to scale the rate between these limits to the span between 4 mA and 20 mA	Y	Y	R/W	00000.0 to 99999.9 (units specified by RUNIT)	7002	0x1B5A	Float
SETP	View or set the amount of fluid measured, in the selected units, for emitting a pulse output. As an example, if the rate is 1 gallon/sec., and SETP is 10 gallons, then one pulse is emitted for every 10 gallons which are metered. This occurs once every 10 seconds. Note, also, that the units for SETP are independently set.	Y	Y	R/W	00000.0 to 99999.9	7004	0x1B5C	Float
FLOW RATE	View flow rate in selected units	Y	Y	R	Computed rate in units specified by RUNIT	7006	0x1B5E	Float
TOTAL - FWD	View forward flow total in selected units	Y	Y	R	Current total volume in forward direction in unit specified by TUNIT	7008	0x1B60	Float
TOTAL - REV	View reverse flow total in selected units	Y	Y	R	Current total volume in reverse direction in unit specified by TUNIT	7010	0x1B62	Float
ZRADC	One of two calibration constants	N	Y	R	Calibrated constant derived from calibration procedure	7012	0x1B64	Float
FSADC	One of two calibration constants	N	Y	R	Calibrated constant derived from calibration procedure	7014	0x1B66	Float
CUTOFF	Corresponds to the lowest measurable flow rate	N	Y	R	Lowest measurable flow rate in gpm	7016	0x1B68	Float
Q4	Corresponds to the maximum flow rate	N	Y	R	Maximum measurable flow rate in gpm based on flow rate of 10 meters/sec and a given pipe diameter	7018	0x1B6A	Float
VOLTAGE	Reports external voltage used to power meter	N	Y	R	Measured external voltage	7020	0x1B6C	Float
TOTAL – BATCH FORWARD	View BATCH forward flow total in selected units	Y	Y	R	Current total BATCH volume in forward direction in unit specified by TUNIT	7022	1B6E	Float
TOTAL - BATCH REVERSE	View BATCH reverse flow total in selected units	Y	Y	R	Current total BATCH volume in reverse direction in unit specified by TUNIT	7024	1B70	Float
SERIAL #	View meter serial number	Y	Y	R	ASCII string containing serial number	6000	0x1770	String
MODEL #	View meter model number	Y	Y	R	ASCII string containing model number	6008	0x1778	String
LOWER FIRMWARE	View body firmware version number	Y	Y	R	ASCII string containing firmware version	6016	0x1780	String
UPPER FIRMWARE	View head firmware version number	Y	Y	R	ASCII string containing firmware version	6024	0x1788	String
MODBUS FIRMWARE	View Modbus firmware version number (Must be polled as seperate transaction)	N	Y	R	ASCII string containing firmware version number	6032-6039	0X1790	String
MBID	View or set the Modbus slave ID	Y	N	R/W	001 to 247 **Default is 001**	n/a	n/a	3-digit BCD
COMM	View or set communication parameters	Y	N	R/W	Baud Rate Parity 38400 None 38400 Even 38400 Odd 19200 None 19200 Even 19200 Odd 9600 None 9600 Even 9600 Odd **Default is 19200/None**	n/a	n/a	String

Note: Due to values added in newer models, your address map may be slightly different.

Data Types

Data and parameters are stored in 16-bit registers (1 word), each register being 2 bytes. All data is in "big-endian" format, or high word/high byte first. This is also sometimes called "float-inverse" or "Float AB CD". The symptom of using the wrong format is a response value that is readable but is entirely wrong.

The following data types are used in the iMAG 4700/AG3000 Modbus® interface:

Integer	16 bit unsigned value corresponding to 1 register
Float	32 bit IEEE floating point number corresponding to 2 sequential registers
String	16 ASCII characters correspond to 16 sequential registers

Function Codes

The Modbus® function codes implemented for the iMAG 4700/AG3000 consist of the following:

Function code *Hex*	*Decimal*	Description
0x03	03	Read Holding Register *Can be used to read one or more integer, float, or string values.* *Requires starting register address and number of registers to read.*
0x04	04	Read Input Register *Can be used to read one or more integer, float, or string values.* *Requires starting register address and number of registers to read.*
0x06	06	Write Single Register *Can be used to write one integer value.* *Requires register address and value.*
0x10	16	Write Multiple Registers *Can be used to write one or more integer, float, or string values.* *Requires starting register address, number of registers to write, and the values.*

Remember! Values may take more than one register. For example, if reading or writing two float values, you would specify four registers, as each floating point number corresponds to two sequential registers.

When quantities are written to the meter, they are also stored in an internal EEPROM device. As such, there is a limit as to how many times the device may be written. With this in mind, the write operation should be used cautiously and sparingly.

Inhanced Accuracy of Very Large Totals

The values for FLOWRATE, FORWARD TOTAL, and REVERSE TOTAL are stored internally in the meter as 64-bit, double precision IEEE numbers. Because the Modbus float values stated previously are represented by 32-bit IEEE single precision floating point numbers, when a total exceeding 16,777,216 units is transmitted by Modbus, the 32-bit float value will include a certain amount of rounding which is relatively insignificant compared to the value itself.

If absolute precision is required in the TOTAL values communicated by Modbus, the FLOWRATE, FORWARD TOTAL and REVERSE TOTAL can be expressed as 64-bit IEEE (double) floating point numbers each corresponding to 4 sequential registers (8 bytes.)

These additional 64-bit addresses are:

Item	Description	On Menu or Display	Via Modbus	R/W	Data Values	Address		Data Type
						Decimal	Hex	
FLOW RATE	View flow rate in selected units	Y	Y	R	Computed rate in units specified by RUNIT	3000	0XBB8	Double Float
TOTAL – FWD	View forward flow total in selected units	Y	Y	R	Current total volume in forward direction in unit specified by TUNIT	3004	0XBBC	Double Float
TOTAL – REV	View reverse flow total in selected units	Y	Y	R	Current total volume in reverse direction in unit specified by TUNIT	3008	0XBCO	Double Float

Each quantity occupies 4 registers

Creating a Modbus® Command

When using your controller (PLC, SCADA system, etc.) to access the meter, you will typically be asked for the following:

For Reading: (function code 0x03 or 0x04/03 or 04 decimal)

- Modbus slave ID
- Function code[1]
- Starting register address[1]
- Number of registers to read[2]

For Writing—single integer: (function code 0x06/06 decimal)

- Modbus slave ID
- Function code[1]
- Register address[1]
- Value to write[3]

For Writing—multiple values: (function code 0x10/16 decimal)

- Modbus slave ID
- Function code[1]
- Starting register address[1]
- Quantity of registers[2]
- Byte count[2]
- Values to write[3]

From the information you enter, the controller will construct a command string to send to the meter. The controller will automatically append a CRC code at the end. This is a two-byte computed value unique to that string. When the meter receives the command string, it will compute its own CRC and verify that it matches the one sent by the controller, thus verifying that all bytes were received correctly. If the CRC codes do not match, an error will be returned to the controller.

1 Some systems may combine the function code and register address. For example, 41001 may mean read holding register 1001, same as function 03 with a register address of 1001 (0x03E9). Consult your controller documentation for details specific to your equipment.

2 Some systems will ask for the number of values and type of value rather than quantity of registers. It will automatically compute the number of registers and byte count. For example, the user would specify 2 values of type float and the system would translate that to 4 registers (or 8 bytes).

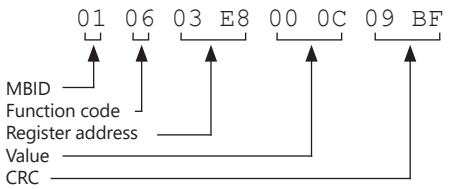
3 Some systems may ask for values in decimal, others may ask for values in hex.

Programming Examples

In this section, some typical programming examples are given for programming the iMAG 4700/AG3000 meter through the Modbus® interface. These examples display the final strings that would be sent to the meter. Remember that in most cases, you will enter requested values into the particular interface of your controller, which will then create the actual programming strings that are sent to the meter. (Note that a space has been added between each byte for legibility. These spaces would not be in an actual command string.)

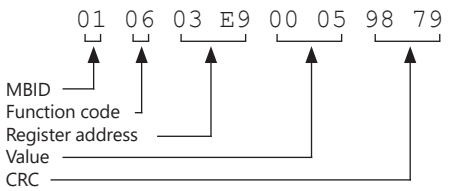
To set TUNIT (Total Units)

To program the total units for liters, we need to write the TUNIT register at address 1000 (0x03E8) with a value of 12 (0x0C). To accomplish this, we can use the Write Single Register command (0x06) with the following string.



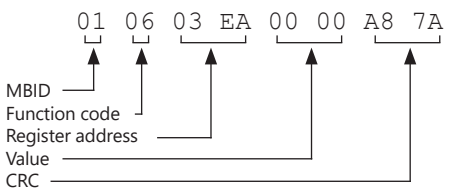
To set RUNIT (Rate Units)

To program the rate units for liters, we need to write the RUNIT register at address 1001 (0x03E9) with a value of 5. To accomplish this, we can use the Write Single Register command (0x06) with the following string.



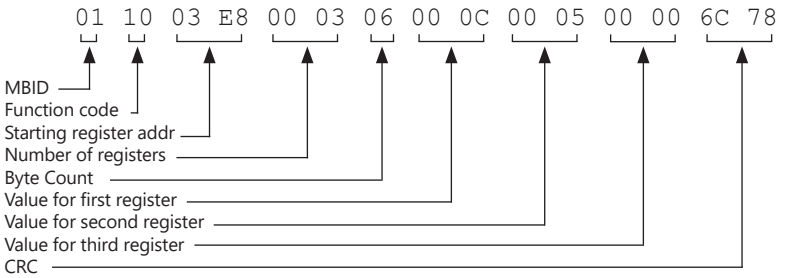
To set RUNIT Time (Rate Unit Time Base)

To program the rate units time base for hours, we need to write the RUNIT TIME register at address 1002 (0x03EA) with a value of 0. To accomplish this, we can use the Write Single Register command (0x06) with the following string.



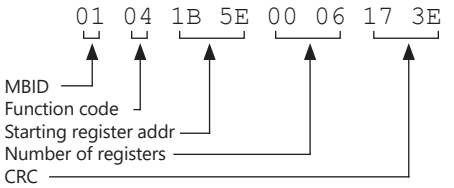
To set TUNIT, RUNIT, and RUNIT Time with a single command

As an alternative to writing each register individually with the Write Single Register command, the Write Multiple Registers command (0x10) can be used to write all 3 registers in one command operation. The following string can be used to write TUNIT, RUNIT, and RUNIT TIME registers. In this example, the same three values are written as were written in the previous examples, i.e., total units in liters, rate units in liters, and time base in seconds.



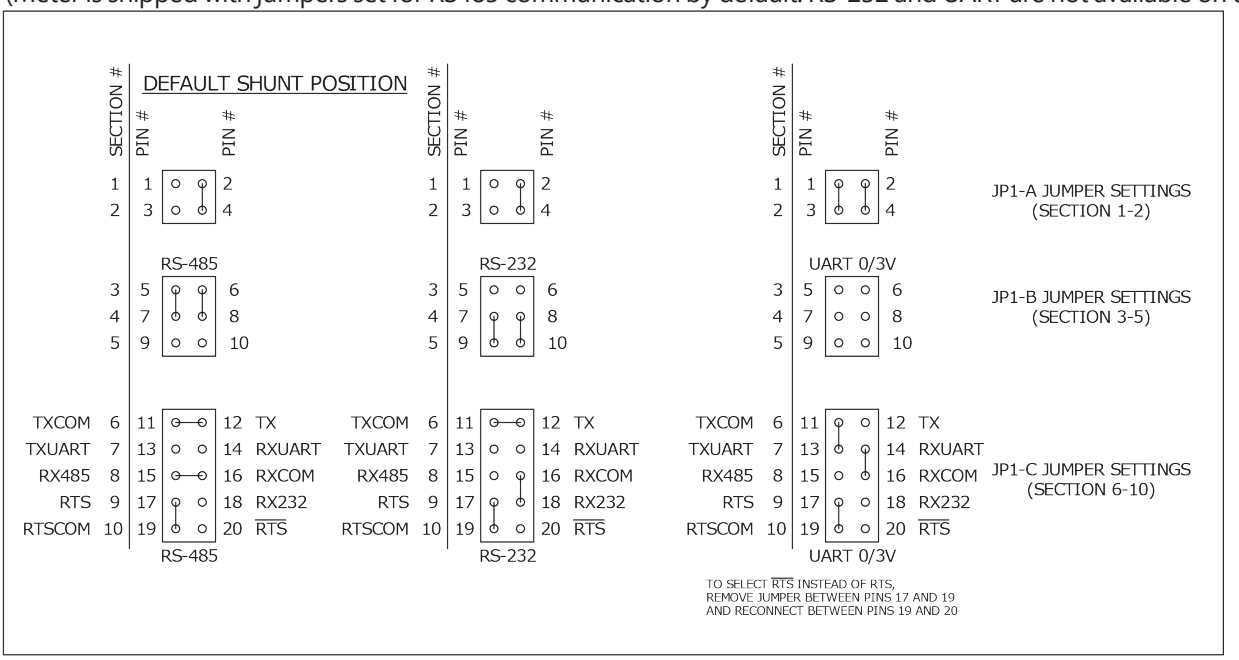
To Monitor Flow Rate and Total

Modbus® can be used to monitor the flow rate and total from the iMAG 4700/AG3000 meter. This can be accomplished with either the Read Holding Registers (0x03) or Read Input Registers (0x04) command. To read flow rate, flow total-forward, and flow-total-reverse in one operation we would use a starting address of 7006 (0x1B5E) and the quantity of registers of 6. Six registers are required since these are floating point numbers requiring 2 registers for each value. Either function code 0x03 or 0x04 can be used.



Modbus Board 102151 Only Jumper Configuration

(Meter is shipped with jumpers set for RS485 communication by default. RS-232 and UART are not available on the 103848 board.)



APPENDIX: READING AND WRITING

Modbus® Responses

Following transmission of a Modbus® command, the meter will either return a successful response or an error response.

Successful Responses

Successful responses will return the following:

Read: (0x03 or 0x04)	Function code, byte count, requested value(s)
Write Single Reg: (0x06)	Function code, register address, value written
Write Multiple Reg: (0x10)	Function code, starting register, quantity of registers written

Failed Responses

Failed responses will return the following:

Read: (0x03 or 0x04)	Error code 0x83 or 0x84, exception code
Write Single Reg: (0x06)	Error code 0x86, exception code
Write Multiple Reg: (0x10)	Error code 0x90, exception code

Exception code 4

An exception code 4 will be returned in the following circumstances:

- **Invalid Address:** An attempt has been made to write to a non-existent register.
- **Wrong number of registers:** The data type implies that a certain number of registers need to designated either for read or write operations. For example, if we wish to write a floating point value then the number of designated registers must be modulus 2 since 2 registers are needed to hold a floating point value. As an example, if a single floating point number is read, and only 1 register is specified for the quantity of registers, then a an exception code is sent.
- **Addressed registers contain mixed data types:** When multiple registers are either written or read, all the implied data types must be must be the same, e.g, all floating point or all integers. If there is a mixture, then an exception code is sent.

Timeout

The conditions under which a timeout results is as follows:

- **Wrong slave ID:** An incorrect slave ID is designated.
- **Wrong COMM parameters:** The baud rate and parity are incorrectly selected.
- **Wrong register address**