SPARLING
INSTRUMENTS, LLC.

TigermagEP Modbus Documentation

For use with TigermagEP firmware versions 2.0.16 & 1.10.28

See *Instructions & Operations Manual for TigermagEP series FM626 and FM656* (IDS_TMEP) section 1.4 for instructions on how to determine the firmware version.

If you have another firmware version, please contact Sparling Instruments to receive the appropriate Modbus Documentation.

190910

Summary of Changes from V2.40 (previous version) and Upgraded Versions (V 2.0.16 and Currently Shipping V1.10.28)[1]

Registers No Longer Used: Note that these registers had data that was related to specific hardware used on v2.40 (the previous version)

10008, 31000, 31001, 31002, 31003, 31005, 31009, 31010, 31011, 31012, 31016, 31017, 31018, 40106, 40107

Registers Which Have Changed

Registers	Description	Current Firmwares 1.10.28 & 2.0.16	Previous Firmware 2.40
30003	Total	Returns Forward Totalizer value.	Returns Forward Totalizer value, or if the meter is set to read 'Net' Total on the display, returns Net Totalizer value.
30005	Reverse Total	Returns Reverse Totalizer value.	When Totalizer is set to toggle between forward and reverse total ('F CNT' and 'R CNT') returns Reverse Totalizer value. Otherwise returns "0".
30013	Reading Age	There are no delays between updates, therefore the value is always "0".	Returns the delay between updating values in 10 ms increments.
30017	All Digit Total	Returns all ten digits of Forward Totalizer value.	Does not exist.
30020	All Digit Reverse Total	Returns all ten digits of Reverse Totalizer value.	Does not exist.
40101	Holding Protect	Default is "25600". No matter what is written to it, it always returns this value (25600) even while retaining a new value written to the register. Value returns to default (25600) after power cycle.	Default value is "23294". When written to, it will show the new value.
40102	Modbus I.D.	New value cannot be written, saved, or implemented unless 40101 is unlocked (32273).	New value can be written but cannot be saved or implemented unless 40101 is unlocked (32273).
40103	Baud Rate	New value cannot be written, saved, or implemented unless 40101 is unlocked (32273).	New value can be written but cannot be saved or implemented unless 40101 is unlocked (32273).
40104	Parity	For Parity: 'None', write "0". New value cannot be written, saved, or implemented unless 40101 is unlocked (32273).	For Parity: 'None', write "3". New value can be written but cannot be saved or implemented unless 40101 is unlocked (32273).
40105	Serial Mode	Default is "0". Currently unused.	Default is "1". Unused.

[1] Please See TigermagEP Installation and Operations Manual Section 1.4 *Show Meter Data* for steps to determine firmware version number

Contents

Chapter 1

TigermagEP Modbus Module Reference Manual

The TigermagEP Modbus Enabled Module implements the Modbus protocol over either RS-232 or RS-485, adapting the TigermagEP for use in installations were modbus is the preferred means of reading instrument values. Communications at the J405 terminal block will be in the modbus protocol.

Electrical Configuration

The module can be configured for serial operation over either RS-232 or RS-485. The desired configuration is to be specified at the time the module is ordered.

The serial connection is made at the screw terminals of J405 of the TigermagEP meter.

Protocol	Terminal 1	Terminal 2
RS-485	A+ (In/Out)	B- (In/Out
RS-232	RX (In)	TX (Out)

The meter also provides Terminal 3 connected to its reference Ground, and Terminal 4 connected to +12V via a pull-up resistor for use as an asserted RS-232 DTR handshake signal.

RS-485 Wiring

The default configuration of EPmod is over RS-485. When configured in RS-485 mode, the meter may share the same pair of wires with other devices.

The module acts as a 1/8-Unit Load, which potentially allows 256 meters to share the RS-485 bus. Since modbus devices have addresses ranging from 1 to 247, all addresses could be accommodated by a single bus. In practice, other modbus devices will share the bus with the meters, and so the actual limit may be much lower. A safe limit to assume for planning is 32 devices per RS-485 bus.

Wiring is “two-wire” multidrop[1]. All devices wire their A+ terminals together, their B- terminals together, and must be connected to a common ground. A twisted pair with characteristic impedance of 120 ohms is recommended for the data wires, and a common ground wire must be included in the cable. Shielding is optional. If isolation is required, it must be provided by an external RS-485 isolator device.

Terminal	RS-485 Bus
1 RX	A+
2 TX	B-
3 GND	GND
4 DTR	Not used

[1] See http://www.maxim-ic.com/app-notes/index.mvp/id/763 for a good discussion of the best practices for wiring RS-485 networks.

Notice that although the RS-485 user community and the modbus standards documents both refer to this as a “two-wire” configuration, the cable between devices must have three conductors. Neglecting the ground connection will sometimes work for devices that are physically near each other and independently connected to the same earth ground. Best practice is to always wire the ground along with the positive and negative data wires.

When more than two devices are connected to a single RS-485 bus, they must be wired in a daisy-chain fashion. In other words, each device is connected to at most two others. If the total cable run is longer than a few feet, then termination is required. To terminate, a resistor equal to the cable’s characteristic impedance should be connected from A+ to B- at each end. Do not connect more than two termination resistors to a single bus, and do not place them anywhere other than at the very ends of the bus. A typical value for the termination resistor is 120 ohms.

RS-232 Wiring

When configured in RS-232 mode, the meter may be wired to exactly one other device. The advantage of RS-232 is that interfaces for PCs are commonplace (and used to be a standard port on older PCs). Use a cable with the following wiring to connect to a PC[2]:

Terminal	DE9 Female Pin	Pin DB25
1 RX	3 TX	2 TX
2 TX	2 RX	3 RX
3 GND	5 GND	7 GND
4 DTR	(1 CD, 6 DSR)	(6 DSR, 8 CD)

RS-232 will operate at any baud rate supported by the PC, but operation below 9600 baud will unnecessarily slow the modbus protocol.

Modbus Protocol Operation

The module implements the Modbus application protocol v1.1b, using two-wire serial in RTU mode with configurable slave address, baud rate, and parity, carried over RS-232 or RS-485.

The modbus register map implemented is fully described in its own chapter, below.

Implementation Class

In the nomenclature of the **Modbus over serial line specification and implementation guide (v1.02)** the module is a REGULAR class device, supporting configuration to any baud rate, even, odd or no parity, and responding to any slave id from 1 to 247. It defaults to slave id 1, at 19200 baud, with even parity.

It supports operation over either RS-232 or RS-485 in two-wire mode. Connection is made through three screw terminals on the J405 terminal block.

Supported Modbus Functions

Code	Hex	Data Type	Function
FC1	01	0xxxx	Read Coils
FC2	02	1xxxx	Read Discrete Inputs
FC3	03	4xxxx	Read Holding Register
FC4	04	3xxxx	Read Input Register
FC5	05	0xxxx	Write Single Coil
FC6	06	4xxxx	Write Single Register
FC15	0F	0xxxx	Write Multiple Coils
FC16	10	4xxxx	Write Multiple Registers
FC17	11		Report Slave ID
FC23	17	4xxxx	Read/Write Multiple Registers

Coils (FC1, FC5, and FC15)

Functions 1, 5 and 15 support reading and writing of coils referenced as 0xxxx. Coils supported by the module are referenced beginning with 00001.

Attempting to reference coils outside of the supported range will result in an error status.

Discrete Inputs (FC2)

Function 2 supports reading discrete inputs referenced as 1xxxx. Supported discrete inputs are referenced beginning with 10001. Discrete inputs may be read with function 2.

Holding Registers (FC3, FC6, FC16, and FC23)

Functions 3, 6, 16, and 23 support reading and writing of 16-bit holding registers referenced as 4xxxx. Supported holding registers are referenced beginning with 40101. Holding registers may be read and written using these functions.

Input Registers (FC4)

Function 4 supports reading 16-bit input registers referenced as 3xxxx. Supported inputs consist of current meter readings are referenced beginning with 30001.

Many of the input registers contain portions of values that cannot be completely represented in only 16 bits. These are either 32 bit IEEE floating point values, or 32 bit integer values. Details on their representation are documented below. Note that the register pairs containing such 32 bit values should be read with a single modbus command.

Chapter 2

TigermagEP Module Registers and Configuration

The Modbus protocol allows instruments to be read and updated by a controlling device over a serial or network connection. (If connection to Modbus over TCP/IP is required, then a suitable Modbus TCP/IP to serial bridge device will be required.)

The module can be configured when assembled for either RS-232 or RS-485, and must be wired in a “two-wire” configuration (two data wires plus a ground wire).

It will operate at a user-configurable baud rate, with a choice of parity. Only Modbus RTU mode is supported.

The module is factory configured for Modbus RTU at 19200 baud with even parity.

The modbus protocol assumes that a remote device has values to read such as the measured flow rate or total, settings to make such as the communications baud rate, discrete status indicators such as the flow rate alarms, and contact closure outputs. These are respectively known as Input Registers, Holding Registers, Discrete Inputs, and Coils.

Each Input, Holding, Discrete and Coil is identified by a number. By convention, numbering starts at 1, and items that are normally read or changed together are assigned consecutive numbers.

Data Types

Where possible, values are stored in single 16-bit registers for best compatibility with modbus usual practice. Several values require more range than can be expressed with 16 bits.

The following data types are used for values stored in either Holding or Input registers.

16-bit integers

The natural data type of a single modbus input or holding register is a 16-bit integer, transferred on the wire with the high-order byte first. Individual registers may be either signed (2’s complement, ranging from -32768 to +32767) or unsigned (ranging from 0 to 65535).

32-bit integers

Two registers read together can be combined to make a 32-bit integer. Of the two registers, the one at the lower address will hold the high-order 16 bits. The values may be either signed (ranging from -2,147,483,648 to +2,147,483,647) or unsigned (ranging from 0 to 4,294,967,296).

32-bit floating point

Two registers read together and interpreted as a 32-bit IEEE single precision floating point value. Of the two registers, the one at the lower address will hold the high-order 16 bits.

32-bit ASCII string

Two registers read together and interpreted as four ASCII characters. Of the two registers, the one at the lower address will hold the first two characters.

48-bit binary coded decimal (BCD)

Three registers read together and interpreted as a twelve digit numeral. Of the three registers, the lower the address, the more significant the values. The digits are read in hexadecimal.

Coils

The following Coil is implemented. Values written to implemented coils are not stored persistently across power cycles.

PZR (00001)

Assert this coil to force the flow to zero, clear it to return to normal flow calculation.

Changes to the PZR coil take effect immediately.

Note that clearing this coil might not release the PZR condition if the PZR terminal is also shorted to ground. Use the PZR discrete status indicator (10004) to test whether the PZR condition is in effect or not.

Discrete Status Indicators

The following discrete status indicators are implemented.

Alarm 1 (10001)

Asserted when measured flow exceeds the alarm 1 level.

Alarm 2 (10002)

Asserted when measured flow exceeds the alarm 2 level.

Reverse Flow (10003)

Asserted when the flow rate is negative.

PZR (10004)

Asserted when measured flow is forced to zero by the PZR contact input to the meter or by the modbus PZR coil.

EPD (10005)

Asserted when empty pipe detection is enabled, and an empty pipe is detected.

Fault (10006)

Asserted when the meter detects a fault condition:

“Active on self test failure, empty pipe and during programming, low/no coil drive and failure of external totalizer to keep up with the flow (registration too small).”[1]

Simulator (10007)

Measured flow is simulated. That is, the meter’s internal flow simulator is enabled in place of the meter’s flow sensor. Note that the simulator defaults to 75% of full scale flow, and it will accumulate in the Total register.

The simulator is turned on via the diagnostic menus of the meter which can be accessed either through the magnetic switches on the meter’s face.

Holdoff (10008)

This register is no longer used in the current implementation of Modbus communication.

Reserved (10009 … 10012)

Reserved for future use.

Input Registers

The following input registers are implemented.

For the values stored in more than one input register, use care to read all of the registers in a single modbus command. The data types used to represent the values are described earlier.

Flow (30001,30002)

32-bit floating point flow in configured units.

This is the flow rate such as read from the meter display. It is represented in two registers as a single-precision (32-bit) floating point value.

Total (30003,30004)

32-bit integer total volume in configured units.

This is the total volume of fluid in units of registration and matches the value on the display as COUNT (FWD ONLY) or F CNT (FWD, REV). For totalizer values less than 2^32. For the full 10-digit totalizer value, use registers 30017-30019 (see below).

Reverse Total (30005,30006)

32-bit integer total volume in configured units.

This is the total volume of fluid which has passed through the meter in reverse and matches the value on the display as R CNT (FWD, REV) when configured for separate forward and reverse totals. For totalizer values less than 2^32. For the full 10-digit reverse totalizer value, use registers 30020-30022 (see below).

[1] IDS_TMEP.PDF section 1.7

Flow Units (30007,30008)

Flow units are three ASCII characters. Register 7 contains the first two characters, and register 8 contains the third and an ASCII NUL character padding.

Typical values include the stock units: “GPM”, “LPM”, “CFS”, “LPS”, “M3H”, “MGD”, “F/S”, “M/S”, or “%FS”. In addition, any custom unit name as configured in the meter may appear.

Total Units (30009,30010)

Total units are three ASCII characters. Register 9 contains the first two characters, and register 10 contains the third and an ASCII NUL character padding.

Typical values include the stock units: “GAL”, “CFT”, “LTR”, and “M3”. In addition, any custom unit name as configured in the meter may appear.

Note that the TigermagEP can be configured to use a custom name equal to one of the predefined abbreviations, and if this is done there is no way to know whether the stock or the custom unit is in use.

Total Registration (30011,30012)

32-bit floating point configured total registration.

This value is the number of configured units represented by each count of the forward and reverse totals.

In practice, the registration will usually be a power of 10, but any “reasonable” value can be entered in the meter’s configuration menu.

Reading Age (30013,30014)

32-bit unsigned count representing the amount of time since the meter reading was last successfully captured by the module firmware.

Time is represented as a count of 10ms ticks. The 32-bit count overflows and wraps after about 497 days or a bit more than 16 months.

In the current firmware, this value should always be zero.

Current Time (30015,30016)

32-bit unsigned count representing the current time.

The current time is represented as a count of 10ms ticks since the meter power was applied. The 32-bit count overflows and wraps after about 497 days or a bit more than 16 months.

All Digit Total (30017,30018,30019)

48-bit BCD value showing the complete ten digit totalizer value as shown on the meter display. The first two digits of the twelve digit output will always be zero.

All Digit Reverse Total (30020,30021,30022)

48-bit BCD value showing the complete ten digit reverse totalizer value as shown on the meter display. The first two digits of the twelve digit output will always be zero.

No Longer Used Input Registers (31000-31018)

These registers applied only to the previous implementation of Modbus communications on the TigermagEP.

Holding Registers

The following holding registers are implemented. Holding registers in the module are backed by non-volatile storage. That storage is read when the module firmware starts executing, and some of the register values are used to set up the Modbus protocol.

Writing to the documented holding registers will report success to Modbus as long as only defined registers are written. However, the values written are ignored unless the Holding Protect register contains a specific value. Some values (notably the Modbus configuration) only take effect when power is first applied to the module.

Holding Protect (40101)

At power up, this register holds the value 25600. It may be freely read and written, however only the default value will read back.

Write the value 32273 (0x7e11 in hexadecimal) to this holding register to allow values written to the other holding registers to be persistently stored when they are written.

The default value of this register is restored after power is next applied.

Modbus ID (40102)

Integer ranging from 1 to 247. The factory default is 1. This is the identifier used to access this meter on the modbus, and must be assigned such that no other instrument connected to a single modbus segment has the same modbus ID.

Any value outside the range will be treated as the value 1, which is also the factory default value.

Changes to this register are stored persistently if the Holding Protect register's current value is 0x7e11.

The persistently stored value in this register takes effect after power is next applied.

Modbus baud rate (40103)

Unsigned 16-bit integer specifying the modbus baud rate.

Note that although any baud rate less than 65535 baud may be specified, only certain baud rates are widely understood by modbus compatible equipment. Those rates are 1200, 4800, 9600, **19200**, 38400, and 57600.

The default baud rate is 19200.

All devices on a single modbus serial network must be configured for the same baud rate, parity, and protocol.

Changes to this register are stored persistently if the Holding Protect register's current value is 0x7e11.

The persistently stored value in this register takes effect after power is next applied.

Modbus Parity (40104)

Integer ranging from 0 to 2 selecting the parity configuration from a list.

Value	Parity
0	No parity
1	Odd parity
2	Even parity (default)

All devices on a single modbus serial network must be configured for the same baud rate, parity, and protocol.

The default parity configuration is Even Parity.

Changes to this register are stored persistently if the Holding Protect register's current value is 0x7e11.

The persistently stored value in this register takes effect after power is next applied.

Modbus Serial Mode (40105)

This register is reserved for future use to select the Modbus mode. The current firmware only supports RTU mode, and this register should only be set to 0 to reflect operation in RTU mode.

No Longer Used Holding Registers (40106,40107)

These registers applied only to the previous implementation of Modbus communications on the TigermagEP.

Reserved (40108)

This holding register is reserved for future use.

User (40109, 40110)

These two registers are provided for user use.

Changes to these registers are stored persistently if the Holding Protect register's current value is 0x7e11.

The module will otherwise pay no attention to values stored in these registers.

Quick Reference

These tables show the modbus reference numbers for all coils, discrete status indicators, input registers and holding registers.

The Reference and Offset columns are an attempt to clarify an ambiguity in the Modbus specification. The documents all talk about items in each data source (e.g. coils, discrete inputs, input registers and holding registers) being named starting at 1, but the underlying representation of Modbus commands on the serial wire maps item 1 to address 0. Historically, various controller manufacturers have assumed that no input source would have more than 9999 unique items, and have unified the naming of items so that each data source begins at a large offset. The Reference column follows that convention, and the Offset column shows the zero-based address for that same item as it appears on the wire.

The Size column shows the number of 16-bit registers (for input and holding registers) that contain the value. Read multiple-register items with a single command to guarantee the value read is atomic.

The Data Type column names the data type used to represent values in those registers.

The Description column identifies the meaning of each data source.

Output Coils

Reference	Offset	Data Type	Description
00001	0	bit	Assert PZR Control

Discrete Inputs

Reference	Offset	Data Type	Description
10001	0	bit	Alarm 1 Asserted
10002	1	bit	Alarm 2 Asserted
10003	2	bit	Reverse Flow
10004	3	bit	PZR Input Asserted
10005	4	bit	Empty Pipe Detected
10006	5	bit	Fault Asserted
10007	6	bit	Simulator in Effect
10008	7	bit	No Longer Used
10009	8	bit	Reserved
10010	9	bit	Reserved
10011	10	bit	Reserved
10012	11	bit	Reserved

Input Registers

Reference	Offset	Size	Data Type	Description
30001	0	2	float	Flow rate
30003	2	2	integer	Total volume
30005	4	2	integer	Reverse Total volume
30007	6	2	ASCII	Flow Units
30009	8	2	ASCII	Total Units
30011	10	2	float	Total Registration
30013	12	2	integer	Time since last meter reading
30015	14	2	integer	Current time, 10 ms ticks
30017	16	3	BCD	All Digit Total volume
30020	19	3	BCD	All Digit Reverse Total volume

Holding Registers

Reference	Offset	Size	Data Type	Description
40101	100	1	integer	Unlock with 0x7e11 (32273)
40102	101	1	integer	Modbus ID
40103	102	1	integer	Modbus baud rate
40104	103	1	integer	Modbus parity
40105	104	1	integer	Modbus serial mode
40106	105	1	integer	No Longer Used
40107	106	1	integer	No Longer Used
40108	107	1	n/a	Reserved
40109	108	2	n/a	User defined