Eastron SDM120CT-MSmart Meter Modbus Protocol Implementation V1.3

1 MODBUS Protocol Message Format

The MODBUS Protocol defines the format for the master's query and the slave's response.
The query contains the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error-checking field.
The response contains fields confirming the action taken, any data to be returned, and an error-checking field. If an error occurred in receipt of the message then the message is ignored, if the slave is unable to perform the requested action, then it will construct an error message and send it as its response. The MODBUS Protocol functions used by the Eastron Digital meters copy 16 bit register values between master and slaves. However, the data used by the Eastron Digital meter is in 32 bit IEEE 754 floating point format. Thus each instrument parameter is conceptually held in two adjacent MODBUS Protocol registers. Query
The following example illustrates a request for a single floating point parameter i.e. two 16-bit Modbus Protocol Registers.

First Byte Last Byte

Slave Address	Function Code	Start Address (Hi)	Start Address (Lo)	Number of Points (Hi)	Number of Points (Lo)	Number of Points (Lo)	Error Check (Lo)	Error Check (Hi)

Slave Address: 8-bit value representing the slave being addressed (1 to 247), 0 is reserved for the broadcast address. The Eastron Digital meters do not support the broadcast address.
Function Code: 8-bit value telling the addressed slave what action is to be performed. (3, 4, 8 or 16 are valid for Eastron Digital meter)
Start Address (Hi): The top (most significant) eight bits of a 16-bit number specifying the start address of the data being requested.
Start Address (Lo): The bottom (least significant) eight bits of a 16-bit number specifying the start address of the data being requested. As registers are used in pairs and start at zero, then this must be an even number.
Number of Points (Hi): The top (most significant) eight bits of a 16-bit number specifying the number of registers being requested.
Number of Points (Lo): The bottom (least significant) eight bits of a 16-bit number specifying the number of registers being requested. As registers are used in pairs, then this must be an even number.
Error Check (Lo): The bottom (least significant) eight bits of a 16-bit number representing the error check value.
Error Check (Hi): The top (most significant) eight bits of a 16-bit number representing the error check value.

Response

The example illustrates the normal response to a request for a single floating point parameter i.e. two 16-bit Modbus Protocol Registers.

First Byte Last Byte

Slave Address	Function Code	Byte Count	First Register (Hi)	First Register (Lo)	Second Register (Hi)	Second Register (Lo)	Error Check (Lo)	Error Check (Hi)

Slave Address: 8-bit value representing the address of slave that is responding.
Function Code: 8-bit value which, when a copy of the function code in the query, indicates that the slave recognised the query and has responded. (See also Exception Response).
Byte Count: 8-bit value indicating the number of data bytes contained within this response
First Register (Hi)*: The top (most significant) eight bits of a 16-bit number representing the first register requested in the query.
First Register (Lo)*: The bottom (least significant) eight bits of a 16-bit number representing the first register requested in the query.
Second Register (Hi)*: The top (most significant) eight bits of a 16-bit number representing the second register requested in the query.
Second Register (Lo)*: The bottom (least significant) eight bits of a 16-bit number representing the second register requested in the query.
Error Check (Lo): The bottom (least significant) eight bits of a 16-bit number representing the error check value.
Error Check (Hi): The top (most significant) eight bits of a 16-bit number representing the error check value.

Exception Response

If an error is detected in the content of the query (excluding parity errors and Error Check mismatch), then an error response (called an exception response), will be sent to the master. The exception response is identified by the function code being a copy of the query function code but with the most-significant bit set. The data contained in an exception response is a single byte error code.

First Byte Last Byte

Slave Address	Function Code	Error Code	Error Check (Lo)	Error Check (Hi)

Slave Address: 8-bit value representing the address of slave that is responding.
Function Code: 8 bit value which is the function code in the query OR'ed with 80 hex, indicating that the slave either does not recognize the query or could not carry out the action requested.
Error Code: 8-bit value indicating the nature of the exception detected.
(See “Table Of Exception Codes“ later).

Error Check (Lo): The bottom (least significant) eight bits of a 16-bit number representing the error check value.
Error Check (Hi): The top (most significant) eight bits of a 16-bit number representing the error check value.

2 Read Input Registers

2.1 MODBUS Protocol code 04 reads the contents of the 3X registers.

Example

The following query will request 'Volts 1' from an instrument with node address 1:

Field Name	Example(Hex)
Slave Address	01
Function	04
Starting Address High	00
Starting Address Low	00
Number of Points High	00
Number of Points Low	02
Error Check Low	71
Error Check High	CB

Note: Data must be requested in register pairs i.e. the "Starting Address" and the "Number of Points" must be even numbers to request a floating point variable. If the "Starting Address" or the "Number of points" is odd then the query will fall in the middle of a floating point variable the product will return an error message.

The following response returns the contents of Volts 1 as 230.2. But see also "Exception Response" later.

Field Name	Example (Hex)
Slave Address	01
Function	04
Byte Count	04
Data, High Reg, High Byte	43
Data, High Reg, Low Byte	66
Data, Low Reg, High Byte	33
Data, Low Reg, Low Byte	34
Error Check Low	1B
Error Check High	38

2.2 Read Holding Registers

MODBUS Protocol code 03 reads the contents of the 4X registers.

Example

The following query will request the prevailing 'Network Node':

Field Name	Example (Hex)
Slave Address	01
Function	03
Starting Address High	00
Starting Address Low	00
Number of Points High	00
Number of Points Low	14
Error Check Low	C4
Error Check High	0B

Note: Data must be requested in register pairs i.e. the "Starting Address" and the "Number of Points" must be even numbers to request a floating point variable. If the "Starting Address" or the "Number of points" is odd then the query will fall in the middle of a floating point variable the product will return an error message.

The following response returns the contents of Demand Time as 1, but see also "Exception Response" later.

Field Name	Example (Hex)
Slave Address	01
Function	03
Byte Count	04
Data, High Reg, High Byte	3F
Data, High Reg, Low Byte	80
Data, Low Reg, High Byte	00
Data, Low Reg, Low Byte	00
Error Check Low	F7
Error Check High	CF

2.3 Write Holding Registers

MODBUS Protocol code 10 (16 decimal) writes the contents of the 4X registers.

Example

The following query will set the Network Node to 60:

Field Name	Example (Hex)
Slave Address	01
Function	10
Starting Address High	00
Starting Address Low	14
Number of Registers High	00
Number of Registers Low	02
Byte Count	04

Data, High Reg, High Byte	42
Data, High Reg, Low Byte	70
Data, Low Reg, High Byte	00
Data, Low Reg, Low Byte	00
Error Check Low	67
Error Check High	D5

Note: Data must be written in register pairs i.e. the “Starting Address“ and the “Number of Points” must be even numbers to write a floating point variable. If the “Starting Address” or the “Number of points” is odd then the query will fall in the middle of a floating point variable the product will return an error message. In general only one floating point value can be written per query

The following response indicates that the write has been successful. But see also “Exception Response “later.

Field Name	**Example (Hex)**
Slave Address	01
Function	10
Starting Address High	00
Starting Address Low	14
Number of Registers High	00
Number of Registers Low	02
Error Check Low	E0
Error Check High	08

Register Map:

Function code 04 to read input parameters:

Address Register	**Input Register Parameter**				**Modbus Protocol Start Address Hex**	
	Parameter	**Length (bytes)**	**Data Format**	**Units**	**Hi byte**	**Lo byte**
30001	Voltage	4	Float	Volts	00	00
30007	Current	4	Float	Amps	00	06
30013	Active power	4	Float	Watts	00	0C
30019	Apparent power	4	Float	VA	00	12
30025	Reactive power	4	Float	VAr	00	18
30031	Power factor	4	Float	None	00	1E
30071	Frequency	4	Float	Hz	00	46
30073	Import active energy	4	Float	kWh	00	48
30075	Export active energy	4	Float	kWh	00	4A
30077	Import reactive energy	4	Float	kvarh	00	4C
30079	Export reactive energy	4	Float	kvarh	00	4E
30085	Total system power	4	Float	W	00	54

	demand					
30087	Maximum total system power demand	4	Float	W	00	56
30089	Import system power demand	4	Float	W	00	58
30091	Maximum Import system power demand	4	Float	W	00	5A
30093	Export system power demand	4	Float	W	00	5C
30095	Maximum Export system power demand	4	Float	W	00	5E
30259	current demand.	4	Float	Amps	01	02
30265	Maximum current demand.	4	Float	Amps	01	08
30343	Total active energy	4	Float	kWh	01	56
30345	Total reactive energy	4	Float	Kvarh	01	58

Function code 10 to set holding parameter ， function code 03 to read holding parameter

Address Register	Holding Register Parameter		Modbus Protocol Start Address Hex		Description	Mode
	Parameters	Format	Hi byte	Lo byte		
40001	Demand Time		00	00	Read minutes into first demand calculation. When the Demand Time reaches the Demand Period then the demand values are valid. **Length : 4 byte** **Data Format : Float**	ro
40003	Demand Period		00	02	Write demand period: 0~60 minutes, Default 60. Range: 0~60, 0 means function closed **Length : 4 byte** **Data Format : Float**	r/w
40013	Relay Pulse Width		00	0C	Write relay on period in milliseconds: 60, 100 or 200, default 100ms. **Length : 4 byte** **Data Format : Float**	r/w
40019	Network Parity Stop		00	12	Write the network port parity/stop bits for MODBUS Protocol, where: 0 = One stop bit and no parity, default. 1 = One stop bit and even parity. 2 = One stop bit and odd parity.3 = Two stop bits and no parity.Requires a restart to become effective. **Length : 4 byte** **Data Format : Float**	r/w
40021	Meter ID		00	14	Ranges from 1 to 247, .Default ID is 1. **Length : 4 byte** **Data Format : Float**	r/w
40029	Baud rate		00	1C	Write the network port baud rate for MODBUS Protocol, where: 0 = 2400 baud(default) 1 = 4800 baud. 2 = 9600 baud 5=1200 baud . **Length : 4 byte** **Data Format : Float**	r/w
40051	CT1		00	32	CT Primary current	r/w

				Ranges from 5 to 9999，Default ID is 5 **Length : 4 byte** **Data Format : Float**	
40053	CT2	00	34	CT2 Range: 1A or 5A , Default 5A Length : 4 byte Data Format : Float	r/w
40087	Pulse 1 output mode	00	56	Write MODBUS Protocol input parameter for pulse out 1: 0001: Import active energy, 0002: Import + export active energy, 0004: Export active energy, (default). 0005: Import reactive energy, 0006: Import + export reactive energy, 0008: Export reactive energy, **Length : 4 byte** **Data Format : Float**	r/w
461457	Reset historical data	F0	10	**00 00 = reset demand info** **Length : 2 byte** **Data Format:Hex**	wo
463745	Time of scroll display	F9	00	Range：0-30s Default 0:does not display in turns **Length : 2 byte** **Data Format : BCD**	r/w
463761	Pulse 1 output	F9	10	0000:0.001kWh/imp 0001:0.01kWh/imp(default) 0002:0.1kWh/imp 0003:1kWh/imp 0004:10kWh/imp 0005:100kWh/imp 0006:1000kWh/imp **Length : 2 byte** **Data Format : Hex**	r/w
463777	Measurement mode	F9	20	0001:mode 1(total = import) 0002:mode 2 (total = import + export）(default) 0003:mode 3 (total = import - export) **Length : 2 byte** **Data Format : Hex**	r/w
464513	Serial number	FC	00	Serial number **Length : 4 byte** **Data Format : unsigned int32** **Note: Only read**	ro