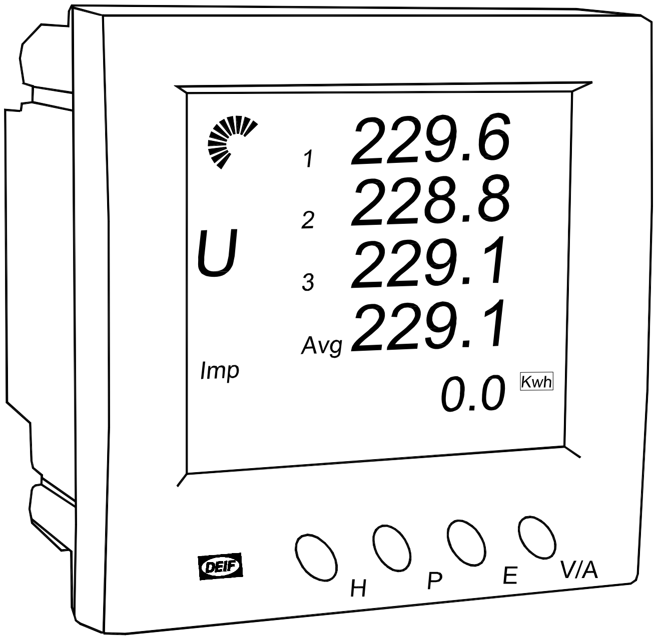
Modbus Communication Manual

-power in control

Multi-instrument MIC

4189320010B



- *Measurement and functions of the MIC*
- *Modbus protocol*
- *Modbus addresses*

DEIF A/S

DEIF A/S, Frisenborgvej 33
DK-7800 Skive, Denmark

Tel.: +45 9614 9614, Fax: +45 9614 9615
E-mail: deif@deif.com, URL: www.deif.com

Table of contents

1. WARNINGS AND LEGAL INFORMATION 3

LEGAL INFORMATION AND RESPONSIBILITY 3
ELECTROSTATIC DISCHARGE AWARENESS 3
SAFETY ISSUES 3
CE-MARKING 3
DEFINITIONS 3

2. MEASUREMENT AND FUNCTIONS OF THE MIC 4

MEASURING PRINCIPLE 4

3. MODBUS PROTOCOL 11

INTRODUCTION OF THE MODBUS PROTOCOL 11
FORMAT OF COMMUNICATION 13

4. MODBUS ADDRESSES 17

MODBUS 17
PARAMETER SETTINGS 17
BASIC ANALOGUE MEASUREMENTS 19
ENERGY MEASUREMENTS 20
PHASE ANGLE MEASUREMENTS 21
MAX./MIN. STATISTICS VALUE WITH TIME STAMPS 22
POWER QUALITY MEASUREMENTS 26
ALARM PARAMETER REGISTER SETTING 27
ALARM SETTING 28
ALARM RECORDING 30
STATUS DIGITAL INPUT 33

1. Warnings and legal information

This chapter includes important information about general legal issues relevant in the handling of DEIF products. Furthermore, some overall safety precautions will be introduced and recommended. Finally, the highlighted notes and warnings, which will be used throughout the document, are presented.

Legal information and responsibility

DEIF takes no responsibility for installation or operation of the engine set. If there is any doubt about how to install or operate the engine controlled by the unit, the company responsible for the installation or the operation of the set must be contacted.

The units are not to be opened by unauthorised personnel. If opened anyway, the warranty will be lost.

Electrostatic discharge awareness

Sufficient care must be taken to protect the terminals against static discharges during the installation. Once the unit is installed and connected, these precautions are no longer necessary.

Safety issues

Installing the unit implies work with dangerous currents and voltages. Therefore, the installation should only be carried out by authorised personnel who understand the risks involved in working with live electrical equipment.

Be aware of the hazardous live currents and voltages. Do not touch any AC measurement inputs as this could lead to injury or death.

CE-marking

The MIC is CE-marked according to the EMC-directive for industrial environments, which normally covers the most common use of the product.

Definitions

Throughout this document a number of notes and warnings will be presented. To ensure that these are noticed, they will be highlighted in order to separate them from the general text.

Notes

The notes provide general information which will be helpful for the reader to bear in mind.

Warnings

The warnings indicate a potentially dangerous situation which could result in death, personal injury or damaged equipment, if certain guidelines are not followed.

2. Measurement and functions of the MIC

Measuring principle

The multi-instrument MIC is based on a modern digital signal processing platform, where all current and voltage input signals are digitally sampled and all values are calculated from these sampled signals. This ensures a very accurate measuring system, which can also measure total harmonic distortion of both voltage and current.

Voltage (U)

The MIC calculates the true RMS value of the three phase-neutral voltages (V1, V2, V3) and the three phase-phase voltages (V12, V23, V31) by the following formula:

$$U_{RMS} = \sqrt{\frac{\sum_{n=1}^{N} u_n^2}{N}}$$

N = Number of sampled values/period
u_n = Sampled value
The MIC uses N = 64

The average phase-neutral voltage (Vlnavg) and the average phase-phase voltage (Vllavg) are calculated from these voltages.

All voltages can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the voltage values:

$$U = Rx \times \frac{PT1}{PT2 \times 10}$$

Rx = Register value
$PT1$ and $PT2$ = Voltage transformer data

Current (I)

The MIC calculates the true RMS value of the three phase currents by the following formula:

$$I_{RMS} = \sqrt{\frac{\sum_{n=1}^{N} i_n^2}{N}}$$

N = Number of sampled values/period
i_n = Sampled value
The MIC uses N = 64

The average current and the neutral current are calculated from these three phase currents.

All currents can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the current values:

$$I = Rx \times \frac{CT1}{5 \times 1000}$$

Rx = Register value
$CT1$ = Current transformer data

Power (P)

The three-phase power and system total power are calculated and displayed in the MIC.

All power values can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the power values:

$$P = Rx \times \frac{PT1}{PT2} \times \frac{CT1}{5}$$

Rx = Register value
$PT1$ and $PT2$ = Voltage transformer data
$CT1$ = Current transformer data

Reactive power (Q)

The three-phase reactive power and system total reactive power are calculated and displayed in the MIC.

All reactive power values can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the reactive power values:

$$Q = Rx \times \frac{PT1}{PT2} \times \frac{CT1}{5}$$

Rx = Register value
$PT1$ and $PT2$ = Voltage transformer data
$CT1$ = Current transformer data

Apparent power (S)

The three-phase apparent power and system total apparent power are calculated and displayed in the MIC.

All apparent power values can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the apparent power values:

$$S = Rx \times \frac{PT1}{PT2} \times \frac{CT1}{5}$$

Rx = Register value
$PT1$ and $PT2$ = Voltage transformer data
$CT1$ = Current transformer data

Frequency (F)

The frequency of phase 1 (voltage input) is measured as system frequency.

The frequency can be viewed on the display. It can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the frequency value:

$$F = \frac{Rx}{100}$$

Rx = Register value

Energy (Ep)

The energy is time integral of power. The unit is kWh. As power has direction, positive means consumption and negative means generating. The MIC has 4 energy counters:

- Import (imp): Consumption of energy
- Export (exp): Generating energy
- Total: Absolute sum of import and export energy
- Net: Algebraic sum of import and export energy

Each of the four energies is measured and stored independently.

The energy registers in the MIC will be cleared to zero and start again when energy data accumulate to $1x10^9$ kWh.

The energy counters can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the energy value:

$$Ep = \frac{Rx}{10} \quad \text{[kWh]}$$

Rx = Register value

Reactive energy (Eq)

The reactive energy is time integral of reactive power. The unit is kVArh. As reactive power has direction, positive means inductive and negative means capacitive.

- Import (imp): Inductive reactive energy
- Export (exp): Capacitive reactive energy
- Total: Absolute sum of import and export reactive energy
- Net: Algebraic sum of import and export reactive energy

Each of the four reactive energies is measured and stored independently.

The reactive energy registers in the MIC will be cleared to zero and start again when energy data accumulate to $1x10^9$ kVArh.

The reactive energy counters can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the energy value:

$$Eq = \frac{Rx}{10} \quad \text{[kVArh]}$$

Rx = Register value

Energy pulse output

The MIC4224 has two digital outputs (DO) which can be selected as energy pulse output. Any two of the 8 energy and reactive energy counters can be assigned to be the pulse output. The pulse width and pulse ratio can be set; pulse width means the duration of the pulse and pulse ratio means how much energy that one pulse represents. When the energy accumulates to the setting limit, there will be a pulse output from the assigned DO port.

Pulse output assignment register:
Any integer from 0 to 8. The digit 0 means no assignment, while 1 to 8 correspond to Ep_imp, Ep_exp, Eq_imp, Eq_exp, Ep_total, Ep_net, Eq_total and Eq_net respectively.

Pulse ratio setting register:
Any integer from 1 to 6000. One digit represents 0.1 kWh or kVArh. Consequently the minimum energy pulse ratio is: 0.1kWh/pulse (0.1 kVArh/pulse) and the maximum energy pulse ratio: 600 kWh/pulse (600 kVArh/pulse).

Pulse width setting register:
Any integer from 1 to 50. One digit represents 20ms (e.g. min. 20ms, max. 1000ms).

The minimum time interval between two adjoining output pulses is 20ms. If the pulse width is 20ms, then the maximum number of output pulses is: 25/s. If the pulse width is 80ms, then the maximum number of output pulse is: 10/s.

In practice the pulse width and the pulse ratio should be selected according to the maximum system power Pmax. The relation of the two parameters should satisfy the following formula:

$$Pulse\_ratio > \frac{(pulse\_width + 1) * P\max}{18000}$$

In the formula the unit of Pmax is kW or kVAr. Recommended pulse ratio is 3 to 5 times the right side value of the above formula.

Maximum demand

The MIC can provide demand values of power, reactive power and apparent power. The MIC uses the sliding window statistics method. The width of the sliding window time can be set between 1 and 30 minutes. The window slides one minute each time. The average demand will always be calculated every minute. If the sliding window time is 3 minutes, it means that the MIC uses the last 3 minutes for calculation of the average demand.

The demand values can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formulas to calculate the power values:

$$Dmd\_P = Rx \times \frac{PT1}{PT2} \times \frac{CT1}{5}$$

Rx = Register value
$PT1$ and $PT2$ = Voltage transformer data
$CT1$ = Current transformer data

$$Dmd\_Q = Rx \times \frac{PT1}{PT2} \times \frac{CT1}{5}$$

Rx = Register value
$PT1$ and $PT2$ = Voltage transformer data
$CT1$ = Current transformer data

$$Dmd\_S = Rx \times \frac{PT1}{PT2} \times \frac{CT1}{5}$$

Rx = Register value
$PT1$ and $PT2$ = Voltage transformer data
$CT1$ = Current transformer data

THD (total harmonic distortion)

The MIC calculates the THD for the three-phase current, for the average current, for the three phase-neutral voltages and for the average phase-neutral voltage. (If the voltage input is 2LL, the MIC will calculate the THD for the three phase-phase voltages). The THD is expressed as a percentage of harmonics due to the fundamental frequency. The MIC uses a true RMS measurement technique, which provides accurate measurement with harmonics present up to the 31st harmonic.

The THD values can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the THD values:

$$THD = \frac{Rx}{100} \quad [\%]$$

Rx = Register value

Three-phase unbalance factor

The MIC can measure the three-phase voltage unbalance factor and three-phase current unbalance factor. The unbalance factor is expressed in percentage.

$$Voltage\ unbalance\ factor = \frac{Max\ difference\ value\ of\ the\ three\ voltages}{Average\ value\ of\ the\ three\ voltages}$$

$$Current\ unbalance\ factor = \frac{Max\ difference\ value\ of\ the\ three\ currents}{Average\ value\ of\ the\ three\ currents}$$

The unbalance factors can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formulas to calculate the unbalance factors:

$$I\_unbl = \frac{Rx}{100} \quad [\%]$$

Rx = Register value

$$U\_unbl = \frac{Rx}{100} \quad [\%]$$

Rx = Register value

Max./min. statistics

The MIC stores the maximum and minimum value of the following values: The three phase-neutral voltages, the three phase-phase voltages, the three phase currents, the power, the reactive power, the apparent power, the power factor, the frequency, the system total power demand, the system total reactive power demand and the system total apparent power demand.

The statistics values can be viewed on the display. The values can, together with the time stamp of each value, be read remotely via the RS485 communication and the utility software MicLink.

The MIC stores these statistics data in a nonvolatile RAM. The statistics data can be reset from the front panel and from the RS485 communication.

If you collect the data from the instrument via the Modbus, please use the different formulas to calculate the real measuring values.

Real time clock

The MIC has a built-in real time clock. The date, month, year, hour, minute and second can be viewed on the display. They can also be read remotely via the RS485 communication and the utility software MicLink.

The clock can be set from the front keys or from the RS485 communication.

Switching from winter time to summer time must be done manually.

Running hour counter

The MIC also has a running hour counter. It will simply show the number of hours the instrument has been connected to the aux. supply.

The running hour counter can be viewed on the display. It can also be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the running hour value:

$$RH = \frac{Rx}{100} \text{ [h]}$$

Rx = Register value

Phase angle difference

The MIC also calculates the phase angle relationship between the different voltages and currents.

The values the MIC calculates depend on the voltage input setting.

With the 2LL setting the MIC will calculate the phase difference U23, I1, I2 and I3 relative to U12.

With the 2LN and 3LN setting the MIC will calculate the phase difference U2, U3, I1, I2, I3 relative to U1.

The phase angle difference can only be read remotely via the RS485 communication and the utility software MicLink.

If you collect the data from the instrument via the Modbus, please use the following formula to calculate the phase angle difference:

$$Phase\ angle = \frac{Rx}{10} \text{ [°]}$$

Rx = Register value

Over/under limit alarming

In the multi-instrument MIC4224 the two digital outputs can also be used as limit alarms. They can trigger a light or sound alarm when one of the measured values is over/under a preset limit. If the alarm is triggered, the limit value and time stamp will be recorded. The MIC will store the last 9 alarm events. Resetting of an alarm is also recorded as an alarm event.

The stored alarm events will be cleared, if the MIC4224 is disconnected from the aux. supply.

A maximum of 9 limit conditions can be set up. Each limit condition will trigger the assigned digital output. You can choose the majority of the measured parameters (35 parameters).

The setup of the limit alarm function can only be done through the RS485 communication. We recommend the use of our utility software MicLink.

Relay output

The MIC4224 also has two relay outputs. They can be used to remote-control an electric switch or other equipment in the switchboard. The relay can only be activated through the RS485 communication. There are two output modes of the relay, latching or momentary. Momentary mode is often used to control the electric switch. The closing time interval can be selected between 50ms and 3000ms.

3. Modbus protocol

Introduction of the Modbus protocol

The Modbus RTU protocol is used for communication in the MIC. The data format and error check method are defined in the Modbus protocol. The half duplex query and response mode is adopted in the Modbus protocol. There is only one master device in the communication net. The others are slave devices, waiting for the query of the master.

Transmission mode

The mode of transmission defines the data structure within a frame and the rules used to transmit data. The mode is defined in the following which is compatible with Modbus RTU mode:

Coding system	8-bit binary
Start bit	1
Data bits	8
Parity	No parity
Stop bit	1
Error checking	CRC check

Framing

Address	Function	Data	Check
8-bits	8-bits	N x 8-bits	16-bits

Table 3.1 Data frame format

Address field

The address field of a message frame contains eight bits. Valid slave device addresses are in the range of 0~247 decimals. A master addresses a slave by placing the slave address in the address field of the message. When the slave sends its response, it places its own address in this address field of the response to let the master know which slave is responding.

Function field

The function code field of a message frame contains eight bits. Valid codes are in the range of 1~255 decimals. When a message is sent from a master to a slave device, the function code field tells the slave what kind of action to perform.

Code	Meaning	Action
01	Read relay output status	Obtain current status of relay output
02	Read digital input (DI) status	Obtain current status of digital input
03	Read data	Obtain current binary value in one or more registers
05	Control relay output	Force relay to a state of ON or OFF
16	Preset multiple-registers	Place specific binary values into a series of consecutive multiple-registers

Table 3.2 Function code

Data field

The data field is constructed using sets of two hexadecimal digits in the range of 00 to FF hexadecimal. The data field of messages sent from a master to slave devices contains additional information which the slave must use to take the action defined by the function code. This can include items like discrete and register addresses, the quantity of items to be handled and the count of actual data bytes in the field. For example, if the master requests a slave to read a group of holding registers (function code 03), the data field specifies the starting register and how many registers are to be read. If the master writes to a group of registers in the slave (function code 10 hexadecimal), the data field specifies the starting register, how many registers to write, the count of data bytes to follow in the data field and the data to be written into the registers.

If no error occurs, the data field of a response from a slave to a master contains the data requested. If an error occurs, the field contains an exception code that the master application can use to determine the next action to be taken. The data field can be nonexistent (of zero length) in certain kinds of messages.

Error check field

Messages include an error checking field that is based on a Cyclical Redundancy Check (CRC) method. The CRC field checks the contents of the entire message. It is applied regardless of any parity check method used for the individual characters of the message. The CRC field is two bytes, containing a 16 bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message.

The receiving device recalculates a CRC during receipt of the message and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results. The CRC is started by first preloading a 16-bit register to all 1s. Then a process begins of applying successive 8-bit bytes of the message to the current contents of the register. Only the eight bits of data in each character are used for generating the CRC. Start and stop bits and the parity bit do not apply to the CRC. During generation of the CRC each 8-bit character is exclusive ORed with the register contents. Then the result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is then exclusive ORed with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place. This process is repeated until eight shifts have been performed. After the last (eighth) shift, the next 8-bit byte is exclusive ORed with the register current value, and the process is repeated for eight more shifts as described above. The final contents of the register, after all the bytes of the message have been applied, is the CRC value. When the CRC is appended to the message, the low-order byte is appended first, followed by the high-order byte.

Format of communication

Explanation of frame.

Addr.	Funct.	Data start reg hi	Data start reg lo	Data # of regs hi	Data # of regs lo	CRC16 hi	CRC16 lo
06H	03H	00H	00H	00H	21H	84H	65H

Table 3.3 Explanation of frame

The meaning of each abbreviation in the headline of table 3.3 is described below:

Addr.:	Address of slave device
Funct.:	Function code
Data start reg hi:	Start register address high byte
Data start reg lo:	Start register address low byte
Data # of regs hi:	Number of register high byte
Data # of regs lo:	Number of register low byte
CRC16 hi:	CRC high byte
CRC16 lo:	CRC low byte

Read status of relay

Function code 01.
This function code is used to read status in the MIC.
1 = ON, 0 = OFF.

There are 2 relays in the MIC. The address of each relay is:
Relay 1 = 0000H, relay 2 = 0001H.

The following query is to read the relay status of MIC number 17.

Query

Addr.	Funct.	Relay start reg hi	Relay start reg lo	Relay # of regs hi	Relay # of regs lo	CRC16 hi	CRC16 lo
11H	01H	00H	00H	00H	02H	BFH	5BH

Table 3.4 Read the status of relay 1 and relay 2 query message

Response

The MIC response includes the MIC address, function code, quantity of data byte, the data and error checking. An example of a response to read the status of relay 1 and relay 2 is shown in table 3.5. The status of relay 1 and relay 2 is responding to the last 2 bits of the data.

Relay 1: Bit0, relay 2: Bit1.

Address	Function code	Byte count	Data	CRC hi	CRC lo
11H	01H	01H	02H	D4H	89H

Table 3.5 Relay status response

The content of the data is:

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0
MSB							LSB

Relay 1 = OFF (LSB), relay 2 = ON (left to LSB).

Read status of DI

Function code 02.
1 = ON, 0 = OFF.

There are 4 DIs in the MIC. The address of each DI is:
DI1 = 0000H, DI2 = 0001H, DI3 = 0002H, DI4=0003H.

The following query is to read the 4 DI status of MIC number 17.

Query

Addr.	Funct.	DI start addr. hi	DI start addr. lo	DI num hi	DI num lo	CRC16 hi	CRC16 lo
11H	02H	00H	00H	00H	04H	7BH	59H

Table 3.6 Read 4 DIs query message

Response

The MIC response includes the MIC address, function code, quantity of data characters, the data characters and error checking. An example of a response to read the status of 4 DIs is shown in table 3.7. The status of each DI is responding to the last 4 bits of the data.

DI1: Bit0, DI2: Bit1, DI3: Bit2, DI4: Bit3.

Address	Function code	Byte count	Data	CRC16 hi	CRC16 lo
11H	02H	01H	03H	E5H	49H

Table 3.7 Read status of DI

The content of the data is:

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0
MSB							LSB

DI1 = ON, DI2 = ON, DI3 = OFF, DI4 = OFF.

Read data (function code 03)

Query

This function allows the master to obtain the measurement results of the MIC. Table 3.8 is an example to read the 3 measured data (F, V1 and V2) from slave device number 17, the data address of F is 0130H, V1 is 0131H and V2 is 0132H.

Addr.	Funct.	Data start addr. hi	Data start addr. lo	Data # of regs hi	Data # of regs lo	CRC16 regs hi	CRC16 regs lo
11H	03H	01H	30H	00H	03H	06H	A8H

Table 3.8 Read F, V1, V2 query message

Response

The MIC response includes the MIC address, function code, quantity of data byte, data and error checking. An example of a response to read F, V1 and V2 (F = 1388H (50.00Hz), V1 = 03E7H (99.9V), V2 = 03E9H (100.1V) is shown in table 3.9.

Addr.	Funct.	Byte count	Data 1 hi	Data 1 lo	Data 2 hi	Data 2 lo	Data 3 hi	Data 3 lo	CRC16 hi	CRC16 lo
11H	03H	06H	13H	88H	03H	E7H	03H	E9H	7FH	04H

Table 3.9 Read F, V1 and V2 message

Control relay (function code 05)

Query

This message forces a single relay either ON or OFF. Any relay that exists within the MIC can be forced to be either status (ON or OFF). The address of relays starts at 0000H (relay 1 = 0000H, relay 2 = 0001H). The data value FF00H will set the relay ON, and the value 0000H will turn it OFF. All other values are illegal and will not affect that relay.

The example below is a request to the MIC number 17 to turn on relay 1.

Addr.	Funct.	DO addr. hi	DO addr. lo	Value hi	Value lo	CRC16 hi	CRC16 lo
11H	05H	00H	00H	FFH	00H	8EH	AAH

Table 3.10 Control relay query message

Response

The normal response to the command request is to retransmit the message as received after the relay status has been altered.

Addr.	Funct.	Relay addr. hi	Relay addr. lo	Value hi	Value lo	CRC hi	CRC lo
11H	05H	00H	00H	FFH	00H	8EH	AAH

Table 3.11 Control relay response message

Preset/reset multi-register (function code 16)

Query

Function 16 allows the user to modify the contents of a multi-register. Any register that exists within the MIC can have its contents changed by this message.

The example below is a request to a MIC number 17 to preset Ep_imp (17807783.3kWh), while its Hex value 0A9D4089H. Ep_imp data address is 0156H and 0157H.

Address	11H
Function	10H
Data start register high	01H
Data start register low	56H
Data register high	00H
Data register low	02H
Byte count	04H
Value high	0AH
Value low	9DH
Value high	40H
Value low	89H
CRC high	4DH
CRC low	B9H

Table 3.12 Preset kWh query message

Response

The normal response to a preset multi-register request includes the MIC address, function code, data start register, the number of registers and error checking.

Address	11H
Function code	10H
Data start address high	01H
Data start address low	56H
Data register high	00H
Data register low	02H
CRC high	A2H
CRC low	B4H

Table 3.13 Preset multi-registers response message

4. Modbus addresses

Modbus

BIT	binary bit	
word	unsign integer of 16 bit	Unsigned integer (16 bit)
Integer	sign integer of 16 bit	Signed integer (16 bit)
dword	unsign integer of 32 bit	Unsigned integer (32 bit)

Parameter settings

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
0100H	Access Code	0~9999	Unsigned integer (16 bit)	03 Read data/16 Presetting	AC=Rx
0101H	Communication Address	1~247	Unsigned integer (16 bit)	03 Read data/16 Presetting	CA=Rx
0102H	Baud Rate	600~38400	Unsigned integer (16 bit)	03 Read data/16 Presetting	BR=Rx
0103H	Voltage Input Wiring Type	0:3LN 1:2LN 2:2LL	Unsigned integer (16 bit)	03 Read data/16 Presetting	VIWT=Rx
0104H	Current Input Wiring Type	0:3CT 1:1CT 2:2CT	Unsigned integer (16 bit)	03 Read data/16 Presetting	CIWT=Rx
0105H 0106H	PT1(high) PT1 (low)	100~500000	Unsigned integer (32 bit)	03 Read data/16 Presetting	PT1=Rx
0107H	PT2	100~400	Unsigned integer (16 bit)	03 Read data/16 Presetting	PT2=Rx
0108H	CT1	5~10000	Unsigned integer (16 bit)	03 Read data/16 Presetting	CT1=Rx
0109H	DO type	0: Pulse 1:Alarm	Unsigned integer (16 bit)	03 Read data/16 Presetting	DOT=Rx
010AH	Energy Variable Number (DO1)	0~8	Unsigned integer (16 bit)	03 Read data/16 Presetting	EVN(DO1)=Rx
010BH	Energy Variable Number (DO2)	0~8	Unsigned integer (16 bit)	03 Read data/16 Presetting	EVN(DO2)=Rx
010CH	Pulse Width	1~50	Unsigned integer (16 bit)	03 Read data/16 Presetting	PW=Rx
010DH	Pulse Rate	1~6000	Unsigned integer (16 bit)	03 Read data/16 Presetting	PR=Rx
010EH	Relay 1 working mode	0: Latch 1: Momentary	Unsigned integer (16 bit)	03 Read data/16 Presetting	
010FH	Relay 1 Pulse Width	50~3000	Unsigned integer (16 bit)	03 Read data/16 Presetting	
0110H	Relay 2 working mode	0: Latch 1: Momentary	Unsigned integer (16 bit)	03 Read data/16 Presetting	
0111H	Relay 2 Pulse Width	50~3000	Unsigned integer (16 bit)	03 Read data/16 Presetting	
0112H	Display backlight Time	0~120	Unsigned integer (16 bit)	03 Read data/16 Presetting	
0113H	Demand sliding window time	1~30	Unsigned integer (16 bit)	03 Read data/16 Presetting	
0114H	Clear Max/Min	Write 000AH	Unsigned integer (16 bit)	16 Presetting	

Real time clock

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
032AH	year	2000~2099	Unsigned integer (16 bit)	03 Read data/16 Presetting	year=Rx
032BH	mon	1~12	Unsigned integer (16 bit)	03 Read data/16 Presetting	mon=Rx
032CH	day	1~31	Unsigned integer (16 bit)	03 Read data/16 Presetting	day=Rx
032DH	hour	0~23	Unsigned integer (16 bit)	03 Read data/16 Presetting	hour=Rx
032EH	min	0~59	Unsigned integer (16 bit)	03 Read data/16 Presetting	min=Rx
032FH	sec	0~59	Unsigned integer (16 bit)	03 Read data/16 Presetting	sec=Rx

Running hour meter

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
03A5H 03A6H	RHh (high) RHl (low)	0~42949018	Unsigned integer (32 bit)	03 Read data	RH=Rx/100 [h]
0115H	Clear running hour counter	Write 000AH	Unsigned integer (16 bit)	16 Presetting	

Basic analogue measurements

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
0130H	Frequency F	0~7000	Unsigned integer (16 bit)	03 Read data	F=Rx/100 [Hz]
0131H	Phase Voltage V1	0~65535	Unsigned integer (16 bit)	03 Read data	V1=Rx*(PT1/PT2)/10 [V]
0132H	Phase Voltage V2	0~65535	Unsigned integer (16 bit)	03 Read data	V2=Rx*(PT1/PT2)/10 [V]
0133H	Phase Voltage V3	0~65535	Unsigned integer (16 bit)	03 Read data	V3=Rx*(PT1/PT2)/10 [V]
0134H	Average Phase Voltage Vlnavg	0~65535	Unsigned integer (16 bit)	03 Read data	Vlnavg=Rx*(PT1/PT2)/10 [V]
0135H	Line Voltage V12	0~65535	Unsigned integer (16 bit)	03 Read data	V12=Rx*(PT1/PT2)/10 [V]
0136H	Line Voltage V23	0~65535	Unsigned integer (16 bit)	03 Read data	V23=Rx*(PT1/PT2)/10 [V]
0137H	Line Voltage V31	0~65535	Unsigned integer (16 bit)	03 Read data	V31=Rx*(PT1/PT2)/10 [V]
0138H	Average Line Voltage Vllavg	0~65535	Unsigned integer (16 bit)	03 Read data	Vllavg=Rx*(PT1/PT2)/10 [V]
0139H	Current I1	0~65535	Unsigned integer (16 bit)	03 Read data	I1=Rx*(CT1/5)/1000 [A]
013AH	Current I2	0~65535	Unsigned integer (16 bit)	03 Read data	I2=Rx*(CT1/5)/1000 [A]
013BH	Current I3	0~65535	Unsigned integer (16 bit)	03 Read data	I3=Rx*(CT1/5)/1000 [A]
013CH	Average Current Iavg	0~65535	Unsigned integer (16 bit)	03 Read data	Iavg=Rx*(CT1/5)/1000 [A]
013DH	Neutral Line Current In	0~65535	Unsigned integer (16 bit)	03 Read data	In=Rx*(CT1/5)/1000 [A]
013EH	Phase Power P1	-32768~32767	Signed integer (16 bit)	03 Read data	P1=Rx*(PT1/PT2)*(CT1/5) [W]
013FH	Phase Power P2	-32768~32767	Signed integer (16 bit)	03 Read data	P2=Rx*(PT1/PT2)*(CT1/5) [W]
0140H	Phase Power P3	-32768~32767	Signed integer (16 bit)	03 Read data	P3=Rx*(PT1/PT2)*(CT1/5) [W]
0141H	System Power Psum	-32768~32767	Signed integer (16 bit)	03 Read data	Psum=Rx*(PT1/PT2)*(CT1/5) [W]
0142H	Phase Reactive Power Q1	-32768~32767	Signed integer (16 bit)	03 Read data	Q1=Rx*(PT1/PT2)*(CT1/5) [var]
0143H	Phase Reactive Power Q2	-32768~32767	Signed integer (16 bit)	03 Read data	Q2=Rx*(PT1/PT2)*(CT1/5) [var]
0144H	Phase Reactive Power Q3	-32768~32767	Signed integer (16 bit)	03 Read data	Q3=Rx*(PT1/PT2)*(CT1/5) [var]
0145H	System Reactive Power Qsum	-32768~32767	Signed integer (16 bit)	03 Read data	Qsum=Rx*(PT1/PT2)*(CT1/5) [var]
0146H	Phase Apparent Power S1	0~65535	Unsigned integer (16 bit)	03 Read data	S1=Rx*(PT1/PT2)*(CT1/5) [VA]
0147H	Phase Apparent Power S2	0~65535	Unsigned integer (16 bit)	03 Read data	S2=Rx*(PT1/PT2)*(CT1/5) [VA]
0148H	Phase Apparent Power S3	0~65535	Unsigned integer (16 bit)	03 Read data	S3=Rx*(PT1/PT2)*(CT1/5) [VA]
0149H	System Apparent Power Ssum	0~65535	Unsigned integer (16 bit)	03 Read data	Ssum=Rx*(PT1/PT2)*(CT1/5) [VA]
014AH	Phase Power Factor PF1	-1000~1000	Signed integer (16 bit)	03 Read data	PF1=Rx/1000 [ ]
014BH	Phase Power Factor PF2	-1000~1000	Signed integer (16 bit)	03 Read data	PF2=Rx/1000 [ ]
014CH	Phase Power Factor PF3	-1000~1000	Signed integer (16 bit)	03 Read data	PF3=Rx/1000 [ ]
014DH	System Power Factor PF	-1000~1000	Signed integer (16 bit)	03 Read data	PF=Rx/1000 [ ]

014EH	Voltage Unbalance Factor U_unbl	0~3000	Unsigned integer (16 bit)	03 Read data	U_unbl=Rx/100 [%]
014FH	Current Unbalance Factor I_unbl	0~3000	Unsigned integer (16 bit)	03 Read data	I_unbl=Rx/100 [%]
0150H	Load Type RT (L/C/R)	76/67/82	Unsigned integer (16 bit)	03 Read data	ASCII of L,C,R Low 8bit of register
0151H	Power Demand Dmd_P	-32768~32767	Signed integer (16 bit)	03 Read data	Dmd_P=Rx*(PT1/PT2)*(CT1/5) [W]
0152H	Reactive power Demand Dmd_Q	-32768~32767	Signed integer (16 bit)	03 Read data	Dmd_Q=Rx*(PT1/PT2)*(CT1/5) [var]
0153H	Apparent Power Demand Dmd_S	0~65535	Unsigned integer (16 bit)	03 Read data	Dmd_S=Rx*(PT1/PT2)*(CT1/5) [VA]

Energy measurements

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
0156H 0157H	Import Energy Ep_imp	0~99999999.9	Unsigned integer (32 bit)	03 Read data 16 Presetting	Ep=Rx/10 [kWh]
0158H 0159H	Export Energy Ep_exp	0~99999999.9	Unsigned integer (32 bit)	03 Read data 16 Presetting	Ep=Rx/10 [kWh]
015AH 015BH	Import Reactive Energy Eq_imp	0~99999999.9	Unsigned integer (32 bit)	03 Read data 16 Presetting	Eq=Rx/10 [kVArh]
015CH 015DH	Export Reactive Energy Eq_exp	0~99999999.9	Unsigned integer (32 bit)	03 Read data 16 Presetting	Eq=Rx/10 [kVArh]
015EH 015FH	Absolute Sum of Energy Ep_total	0~99999999.9	Unsigned integer (32 bit)	03 Read data 16 Presetting	Ep=Rx/10 [kWh]
0160H 0161H	Algebra Sum of Energy Ep_net	0~99999999.9	Unsigned integer (32 bit)	03 Read data 16 Presetting	Ep=Rx/10 [kWh]
0162H 0163H	Absolute Sum of Reactive Energy Eq_total	0~99999999.9	Unsigned integer (32 bit)	03 Read data 16 Presetting	Eq=Rx/10 [kVArh]
0164H 0165H	Algebra Sum of Reactive Energy Eq_net	0~99999999.9	Unsigned integer (32 bit)	03 Read data 16 Presetting	Eq=Rx/10 [kVArh]

Phase angle measurements

Only coupling: 2LN and 3LN

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
039DH	Phase difference V1/V2	0~3600	Signed integer (16 bit)	03 Read data	Phase angleV1/V2=Rx/10 [Degree]
039EH	Phase difference V1/V3	0~3600	Signed integer (16 bit)	03 Read data	Phase angleV1/V3=Rx/10 [Degree]
039FH	Phase difference V1/I1	0~3600	Signed integer (16 bit)	03 Read data	Phase angleV1/I1=Rx/10 [Degree]
03A0H	Phase difference V1/I2	0~3600	Signed integer (16 bit)	03 Read data	Phase angleV1/I2=Rx/10 [Degree]
03A1H	Phase difference V1/I3	0~3600	Signed integer (16 bit)	03 Read data	Phase angleV1/I3=Rx/10 [Degree]

Only coupling: 2LL ad 3LL

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
03A2H	Phase difference V12/V23	0~3600	Signed integer (16 bit)	03 Read data	Phase angleV12/V23=Rx/10 [Degree]
03A3H	Phase difference V12/I1	0~3600	Signed integer (16 bit)	03 Read data	Phase angleV12/I1=Rx/10 [Degree]
03A4H	Phase difference V12/I3	0~3600	Signed integer (16 bit)	03 Read data	Phase angleV12/I3=Rx/10 [Degree]

Max./min. statistics value with time stamps

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
0239H	V1_max	0~65535	Unsigned integer (16 bit)	03 Read data	V1_max=Rx*(PT1/PT2)/10 [V]
023AH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
023BH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
023CH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
023DH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
023EH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
023FH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0240H	V2_max	0~65535	Unsigned integer (16 bit)	03 Read data	V2_max=Rx*(PT1/PT2)/10 [V]
0241H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0242H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0243H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0244H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0245H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0246H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0247H	V3_max	0~65535	Unsigned integer (16 bit)	03 Read data	V3_max=Rx*(PT1/PT2)/10 [V]
0248H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0249H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
024AH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
024BH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
024CH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
024DH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
024EH	V12_max	0~65535	Unsigned integer (16 bit)	03 Read data	V12_max=Rx*(PT1/PT2)/10 [V]
024FH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0250H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0251H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0252H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0253H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0254H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0255H	V23_max	0~65535	Unsigned integer (16 bit)	03 Read data	V23_max=Rx*(PT1/PT2)/10 [V]
0256H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0257H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0258H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0259H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
025AH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
025BH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
025CH	V31_max	0~65535	Unsigned integer (16 bit)	03 Read data	V31_max=Rx*(PT1/PT2)/10 [V]
025DH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
025EH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
025FH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0260H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0261H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0262H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0263H	I1_max	0~65535	Unsigned integer (16 bit)	03 Read data	I1_max=Rx*(CT1/5)/1000 [A]
0264H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0265H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0266H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0267H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0268H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0269H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
026AH	I2_max	0~65535	Unsigned integer (16 bit)	03 Read data	I2_max=Rx*(CT1/5)/1000 [A]
026BH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
026CH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
026DH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
026EH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
026FH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0270H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx

0271H	I3_max	0~65535	Unsigned integer (16 bit)	03 Read data	I3_max=Rx*(CT1/5)/1000 [A]
0272H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0273H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0274H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0275H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0276H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0277H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0278H	P_max	-32768~32767	Signed integer (16 bit)	03 Read data	P_max=Rx*(PT1/PT2)*(CT1/5) [W]
0279H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
027AH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
027BH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
027CH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
027DH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
027EH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
027FH	Q_max	-32768~32767	Signed integer (16 bit)	03 Read data	Q_max=Rx*(PT1/PT2)*(CT1/5)
0280H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	[VAr]
0281H	mon	1~12	Unsigned integer (16 bit)	03 Read data	year=Rx
0282H	day	1~31	Unsigned integer (16 bit)	03 Read data	mon=Rx
0283H	hour	0~23	Unsigned integer (16 bit)	03 Read data	day=Rx
0284H	min	0~59	Unsigned integer (16 bit)	03 Read data	hour=Rx
0285H	sec	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
					sec=Rx
0286H	S_max	0~65535	Unsigned integer (16 bit)	03 Read data	S_max=Rx*(PT1/PT2)*(CT1/5) [VA]
0287H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0288H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0289H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
028AH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
028BH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
028CH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
028DH	PF_max	-1000~1000	Signed integer (16 bit)	03 Read data	PF_max=Rx/1000 [ ]
028EH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
028FH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0290H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0291H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0292H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0293H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0294H	F_max	0~7000	Unsigned integer (16 bit)	03 Read data	F_max=Rx/100 [Hz]
0295H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0296H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0297H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0298H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0299H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
029AH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
029BH	Dmd_P_max	-32768~32767	Signed integer (16 bit)	03 Read data	Dmd_P_max=Rx*(PT1/PT2)*(CT1/5) [W]
029CH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
029DH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
029EH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
029FH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02A0H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02A1H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02A2H	Dmd_Q_max	-32768~32767	Signed integer (16 bit)	03 Read data	Dmd_Q_max=Rx*(PT1/PT2)*(CT1/5) [VAr]
02A3H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02A4H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02A5H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02A6H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02A7H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02A8H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02A9H	Dmd_S_max	0~65535	Unsigned integer (16 bit)	03 Read data	Dmd_S_max=Rx*(PT1/PT2)*(CT1/5) [VA]
02AAH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02ABH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02ACH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02ADH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02AEH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02AFH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx

Min. Record

02B0H	V1_min	0~65535	Unsigned integer (16 bit)	03 Read data	V1_min=Rx*(PT1/PT2)/10 [V]
02B1H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02B2H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02B3H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02B4H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02B5H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02B6H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02B7H	V2_min	0~65535	Unsigned integer (16 bit)	03 Read data	V2_min=Rx*(PT1/PT2)/10 [V]
02B8H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02B9H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02BAH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02BBH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02BCH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02BDH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02BEH	V3_min	0~65535	Unsigned integer (16 bit)	03 Read data	V3_min=Rx*(PT1/PT2)/10 [V]
02BFH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02C0H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02C1H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02C2H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02C3H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02C4H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02C5H	V12_min	0~65535	Unsigned integer (16 bit)	03 Read data	V12_min=Rx*(PT1/PT2)/10 [V]
02C6H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02C7H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02C8H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02C9H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02CAH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02CBH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02CCH	V23_min	0~65535	Unsigned integer (16 bit)	03 Read data	V23_min=Rx*(PT1/PT2)/10 [V]
02CDH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02CEH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02CFH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02D0H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02D1H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02D2H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02D3H	V31_min	0~65535	Unsigned integer (16 bit)	03 Read data	V31_min=Rx*(PT1/PT2)/10 [V]
02D4H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02D5H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02D6H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02D7H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02D8H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02D9H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02DAH	I1_min	0~65535	Unsigned integer (16 bit)	03 Read data	I1_min=Rx*(CT1/5)/1000 [A]
02DBH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02DCH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02DDH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02DEH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02DFH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02E0H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02E1H	I2_min	0~65535	Unsigned integer (16 bit)	03 Read data	I2_min=Rx*(CT1/5)/1000 [A]
02E2H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02E3H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02E4H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02E5H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02E6H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02E7H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02E8H	I3_min	0~65535	Unsigned integer (16 bit)	03 Read data	I3_min=Rx*(CT1/5)/1000 [A]
02E9H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02EAH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02EBH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02ECH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02EDH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02EEH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx

02EFH	P_min	-32768~32767	Signed integer (16 bit)	03 Read data	P_min=Rx*(PT1/PT2)*(CT1/5) [W]
02F0H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02F1H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02F2H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02F3H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02F4H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02F5H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02F6H	Q_min	-32768~32767	Signed integer (16 bit)	03 Read data	Q_min=Rx*(PT1/PT2)*(CT1/5) [VAr]
02F7H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02F8H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
02F9H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
02FAH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
02FBH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
02FCH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
02FDH	S_min	0~65535	Unsigned integer (16 bit)	03 Read data	S_min=Rx*(PT1/PT2)*(CT1/5) [VA]
02FEH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
02FFH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0300H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0301H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0302H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0303H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0304H	PF_min	-1000~1000	Signed integer (16 bit)	03 Read data	PF_min=Rx/1000 [ ]
0305H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0306H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0307H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0308H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0309H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
030AH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
030BH	F_min	0~7000	Unsigned integer (16 bit)	03 Read data	F_min=Rx/100 [Hz]
030CH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
030DH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
030EH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
030FH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0310H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0311H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0312H	Dmd_P_min	-32768~32767	Signed integer (16 bit)	03 Read data	Dmd_P_min=Rx*(PT1/PT2)*(CT1/5) [W]
0313H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0314H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0315H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0316H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0317H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0318H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0319H	Dmd_Q_min	-32768~32767	Signed integer (16 bit)	03 Read data	Dmd_Q_min=Rx*(PT1/PT2)*(CT1/5) [VAr]
031AH	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
031BH	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
031CH	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
031DH	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
031EH	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
031FH	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0320H	Dmd_S_min	0~65535	Unsigned integer (16 bit)	03 Read data	Dmd_S_min=Rx*(PT1/PT2)*(CT1/5) [VA]
0321H	year	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0322H	mon	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0323H	day	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0324H	hour	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0325H	min	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0326H	sec	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx

Power quality measurements

Function code: 03 read data.

Addr.	**Parameter**	**Range**	**Object Type**	**Function code**	**Relation between register value Rx and real value**
0168H	Total Harmonic Distortion of V1 or V12 THD_V1	0~10000	Unsigned integer (16 bit)	03 Read data	THD_V1=Rx/100 [%]
0169H	Total Harmonic Distortion of V2 or V23 THD_V2	0~10000	Unsigned integer (16 bit)	03 Read data	THD_V2=Rx/100 [%]
016AH	Total Harmonic Distortion of V3 or V31THD_V3	0~10000	Unsigned integer (16 bit)	03 Read data	THD_V3=Rx/100 [%]
016BH	Average Total Harmonic Distortion of Voltage THD_V	0~10000	Unsigned integer (16 bit)	03 Read data	THD_V=Rx/100 [%]
016CH	Total Harmonic Distortion of I1 THD_I1	0~10000	Unsigned integer (16 bit)	03 Read data	THD_I1=Rx/100 [%]
016DH	Total Harmonic Distortion of I2 THD_I2	0~10000	Unsigned integer (16 bit)	03 Read data	THD_I2=Rx/100 [%]
016EH	Total Harmonic Distortion of I3 THD_I3	0~10000	Unsigned integer (16 bit)	03 Read data	THD_I3=Rx/100 [%]
016FH	Average Total Harmonic Distortion of Current THD_I	0~10000	Unsigned integer (16 bit)	03 Read data	THD_I=Rx/100 [%]

Alarm parameter register setting

Over/under the limit alarming can be set for the following 35 parameters:

VAr #	Parameter
0	Frequency F
1	Phase Voltage V1
2	Phase Voltage V2
3	Phase Voltage V3
4	Average Phase Voltage Vlnavg
5	Line Voltage V12
6	Line Voltage V23
7	Line Voltage V31
8	Average Line Voltage Vllavg
9	Current I1
10	Current I2
11	Current I3
12	Average Current Iavg
13	Neutral Line Current In
14	Phase Power P1
15	Phase Power P2
16	Phase Power P3
17	System Power Psum
18	Phase Reactive Power Q1
19	Phase Reactive Power Q2
20	Phase Reactive Power Q3
21	System Reactive Power Qsum
22	Phase Apparent Power S1
23	Phase Apparent Power S2
24	Phase Apparent Power S3
25	System Apparent Power Ssum
26	Phase Power Factor PF1
27	Phase Power Factor PF2
28	Phase Power Factor PF3
29	System Power Factor PF
30	Voltage Unbalance Factor U_unbl
31	Current Unbalance Factor I_unbl
32	Power Demand Dmd_P
33	Reactive power Demand Dmd_Q
34	Apparent Power Demand Dmd_S

Alarm setting

Function code: 03 for reading, 16 for presetting.

Addr.	Parameter	Range	Object Type	Function code
0330H	9 condition inequalities enable registers	Bit 0~8 corresponding to 1st~9th inequality	Integer	03 Read data/ 16 Presetting
0331H	Time limit	0~255 (0-76500ms)	Integer	03 Read data/ 16 Presetting
0332H	Register associated DO1 with inequalities, AssociatedDO1	Bit 0~8 corresponding to 1st~9th inequality 1:Yes 0:No	Integer	03 Read data/ 16 Presetting
0333H	Register associated DO2 with inequalities, AssociatedDO2	Bit0~8 corresponding to 1st~9th inequality 1:Yes 0:No	Integer	03 Read data/ 16 Presetting
0334H	Register associated 1st inequality with one of the 34 variables, var1	0~34	Integer	03 Read data/ 16 Presetting
0335H	Relation symbol selecting register, INEQU_sign1	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
0336H	Limit value for 1st inequality, Ref1	Related with variable	Word or Integer	03 Read data/ 16 Presetting
0337H	Register associated 2nd inequality with one of the 34 variables, var2	0~34	Integer	03 Read data/ 16 Presetting
0338H	Relation symbol selecting register, INEQU_sign2	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
0339H	Limit value for 2nd inequality, Ref2	Related with variable	Word or Integer	03 Read data/ 16 Presetting
033AH	Register associated 3rd inequality with one of the 34 variables, var3	0~34	Integer	03 Read data/ 16 Presetting
033BH	Relation symbol selecting register, INEQU_sign3	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
033CH	Limit value for 3rd inequality, Ref3	Related with variable	Word or Integer	03 Read data/ 16 Presetting
033DH	Register associated 4th inequality with one of the 34 variables, var4	0~34	Integer	03 Read data/ 16 Presetting
033EH	Relation symbol selecting register: INEQU_sign4	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
033FH	Limit value for 4th inequality, Ref4	Related with variable	Word or Integer	03 Read data/ 16 Presetting
0340H	Register associated 5th inequality with one of the 34 variables, var5	0~34	Integer	03 Read data/ 16 Presetting
0341H	Relation symbol selecting register, INEQU_sign5	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
0342H	Limit value for 5th inequality, Ref5	Related with variable	Word or Integer	03 Read data/ 16 Presetting
0343H	Register associated 6th inequality with one of the 34 variables, var6	0~34	Integer	03 Read data/ 16 Presetting
0344H	Relation symbol selecting register, INEQU_sign6	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
0345H	Limit value for 6th inequality, Ref6	Related with variable	Word or Integer	03 Read data/ 16 Presetting
0346H	Register associated 7th inequality with one of the 34 variables, var7	0~34	Integer	03 Read data/ 16 Presetting
0347H	Relation symbol selecting register, INEQU_sign7	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
0348H	Limit value for 7th inequality, Ref7	Related with variable	Word or Integer	03 Read data/ 16 Presetting
0349H	Register associated 8th inequality with one of the 34 variables, var8	0~34	Integer	03 Read data/ 16 Presetting

034AH	Relation symbol selecting register, INEQU_sign8	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
034BH	Limit value for 8th inequality, Ref8	Related with variable	Word or Integer	03 Read data/ 16 Presetting
034CH	Register associated 9th inequality with one of the 34 variables, var9	0~34	Integer	03 Read data/ 16 Presetting
034DH	Relation symbol selecting register, INEQU_sign9	0:< low limit 1:> up limit	Integer	03 Read data/ 16 Presetting
034EH	Limit value for 9th inequality, Ref9	Related with variable	Word or Integer	03 Read data/ 16 Presetting

Alarm recording

Function code: 03 for reading.

Addr.	Parameter	Range	Object Type	Function code	Relation between register value Rx and real value
0354H	Limit Alarm Status of the 9 inequalities	Bit0~8 corresponding to 1st~9th inequality 0: No 1: Yes	Integer	03 Read data	
0355H	Variable Number of the 1st Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
0356H	Variable Value of the 1st Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
0357H	Year of 1st Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0358H	Month of 1st Alarm Record 1~12 word R	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0359H	Date of 1st Alarm Record word R	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
035AH	Hour of 1st Alarm Record word R	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
035BH	Minute of 1st Alarm Record word R	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
035CH	Second of 1st Alarm Record 0~59 word R	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
035DH	Variable Number of the 2nd Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
035EH	Variable Value of the 2nd Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
035FH	Year of 2nd Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0360H	Month of 2nd Alarm Record	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0361H	Date of 2nd Alarm Record	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0362H	Hour of 2nd Alarm Record	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0363H	Minute of 2nd Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0364H	Second of 2nd Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0365H	Variable Number of the 3rd Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
0366H	Variable Value of the 3rd Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
0367H	Year of 3rd Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0368H	Month of 3rd Alarm Record	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0369H	Date of 3rd Alarm Record	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
036AH	Hour of 3rd Alarm Record	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
036BH	Minute of 3rd Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
036CH	Second of 3rd Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx

036DH	Variable Number of the 4th Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
036EH	Variable Value of the 4th Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
036FH	Year of 4th Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0370H	Month of 4th Alarm Record	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0371H	Date of 4th Alarm Record	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0372H	Hour of 4th Alarm Record	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0373H	Minute of 4th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0374H	Second of 4th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0375H	Variable Number of the 5th Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
0376H	Variable Value of the 5th Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
0377H	Year of 5th Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0378H	Month of 5th Alarm Record	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0379H	Date of 5th Alarm Record	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
037AH	Hour of 5th Alarm Record	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
037BH	Minute of 5th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
037CH	Second of 5th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
037DH	Variable Number of the 6th Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
037EH	Variable Value of the 6th Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
037FH	Year of 6th Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0380H	Month of 6th Alarm Record	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0381H	Date of 6th Alarm Record	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0382H	Hour of 6th Alarm Record	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0383H	Minute of 6th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0384H	Second of 6th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0385H	Variable Number of the 7th Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
0386H	Variable Value of the 7th Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
0387H	Year of 7th Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0388H	Month of 7th Alarm Record	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0389H	Date of 7th Alarm Record	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
038AH	Hour of 7th Alarm Record	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
038BH	Minute of 7th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
038CH	Second of 7th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx

038DH	Variable Number of the 8th Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
038EH	Variable Value of the 8th Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
038FH	Year of 8th Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0390H	Month of 8th Alarm Record	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0391H	Date of 8th Alarm Record	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
0392H	Hour of 8th Alarm Record	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
0393H	Minute of 8th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
0394H	Second of 8th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx
0395H	Variable Number of the 9th Alarm record	0~34	Unsigned integer (16 bit)	03 Read data	
0396H	Variable Value of the 9th Alarm Record	-32768~32767 or 0~65535	Integer or Word	03 Read data	
0397H	Year of 9th Alarm Record	2000~2099	Unsigned integer (16 bit)	03 Read data	year=Rx
0398H	Month of 9th Alarm Record	1~12	Unsigned integer (16 bit)	03 Read data	mon=Rx
0399H	Date of 9th Alarm Record	1~31	Unsigned integer (16 bit)	03 Read data	day=Rx
039AH	Hour of 9th Alarm Record	0~23	Unsigned integer (16 bit)	03 Read data	hour=Rx
039BH	Minute of 9th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	min=Rx
039CH	Second of 9th Alarm Record	0~59	Unsigned integer (16 bit)	03 Read data	sec=Rx

Status digital input

Function code: 02 read digital input status.

Address	Parameter	Range	Object Type	Type of access
0000H	DI1	0:off 1:on	bit	R
0001H	DI2	0:off 1:on	bit	R
0002H	DI3	0:off 1:on	bit	R
0003H	DI4	0:off 1:on	bit	R

Status relay and control

Function code: 01 read relay output status, or 05 control relay output.

Address	Parameter	Range	Object Type	Type of access
0000H	Relay1	0:off 1:on	bit	R/W
0001H	Relay2	0:off 1:on	bit	R/W

DEIF A/S reserves the right to change any of the above.