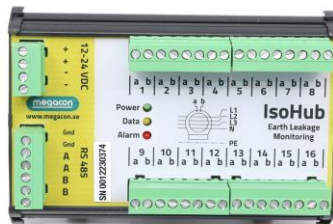
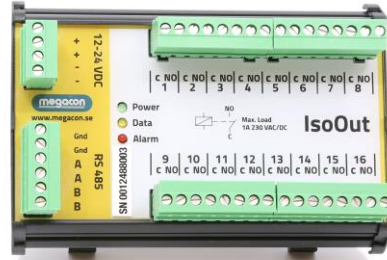
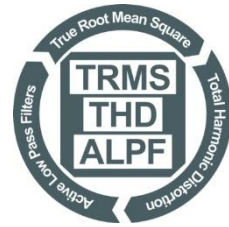




Due to our policy of continual improvement, specifications may change without prior notice

IsoBase, IsoBox, IsoHub, IsoOut Earth Leakage Monitoring System ModBus Implementation User Manual





Due to our policy of continual improvement, specifications may change without prior notice

Contents

| | | |
|------|---|---|
| 1 | ModBus Implementation for IsoBase | 3 |
| 1.1 | Available data | 3 |
| 2 | Remarks..... | 4 |
| 2.1 | Date & Time | 4 |
| 2.2 | IsoHub state and SN..... | 4 |
| 2.3 | IsoBase battery info | 4 |
| 2.4 | Input properties | 4 |
| 2.5 | Input maximum and minimum values | 4 |
| 2.6 | Input states | 4 |
| 2.7 | Input memory | 4 |
| 2.8 | Input event counter | 5 |
| 2.9 | Input labels | 5 |
| 2.10 | Event memory..... | 5 |
| 2.11 | Power Memory | 6 |
| 2.12 | Diagnostics | 6 |
| 2.13 | Read device identification..... | 6 |
| 2.14 | Input Filter Active..... | 6 |



Due to our policy of continual improvement, specifications may change without prior notice

1 ModBus Implementation for IsoBase

1.1 Available data

| Data content | Address (PDU) | Address decimal | Number of words | Read/Write | Function codes | Remarks |
|-------------------------------|---------------|-----------------|-----------------|------------|----------------|---------|
| Date & Time | 0000h..0006h | 0..6 | 7 | rw | 3,4,16 | 1) |
| IsoHub state and SN | 0100h..010Fh | 256..271 | 8*2 | r | 3,4 | 2) |
| IsoBase battery info | 0200h | 32 | 1 | r | 3,4 | 3) |
| Input warning level | 0300h..037Fh | 768..895 | 128*1 | rw | 3,4,6,16 | 4) |
| Input alarm level | 0380h..03FFh | 896..1023 | 128*1 | rw | 3,4,6,16 | 4) |
| Input delay | 0400h..047Fh | 1024..1151 | 128*1 | rw | 3,4,6,16 | 4) |
| Input current | 0480h..04FFh | 1152..1279 | 128*1 | r | 3,4 | 4) |
| Input maximum | 0500h..057Fh | 1280..1407 | 128*1 | rw | 3,4,6,16 | 4), 5) |
| Input minimum | 0580h..05FFh | 1408..1535 | 128*1 | rw | 3,4,6,16 | 4), 5) |
| Input state | 0600h..067Fh | 1536..1663 | 128*1 | r | 3,4 | 4), 6) |
| Input memory | 0680h..06FFh | 1664..1791 | 128*1 | rw | 3,4,6,16 | 4), 7) |
| Input total event count | 0700h..077Fh | 1792..1919 | 128*1 | r | 3,4 | 4) |
| Input new event count | 0780h..07FFh | 1920..2047 | 128*1 | rw | 3,4,6,16 | 4), 8) |
| Input labels | 0800h..0BFFh | 2048..3071 | 128*8 | rw | 3,4,6,16 | 9) |
| Input TRMS w/o filter (V4.0) | 0C00h..0C7Fh | 3072..3199 | 128*1 | r | 3,4 | 4) |
| Input TRMS with filter (V4.0) | 0C80h..0CFFh | 3200..3327 | 128*1 | r | 3,4 | 4) |
| Input THD (V4.0) | 0D00h..0D7Fh | 3328..3455 | 128*1 | r | 3,4 | 4) |
| Input filter active (V4.0) | 0D80h..0DFFh | 3456..3583 | 128*1 | rw | 3,4,6,16 | 4), 14) |
| | | | | | | |
| Event memory | | | | r | 100 | 10) |
| Power memory | | | | r | 105 | 11) |
| | | | | | | |
| Diagnostics | | | | r | 8 | 12) |
| Device Identification | | | | r | 43 14 | 13) |

Generally it is not possible to access 2 different logical data blocks with only one message. There is no automatic incrementation of the register address into the next data block.

Example: A trial to read the last 3 registers of the input delay block and the first register of the input current block will lead to an exception response "illegal data address".

Provided communication parameters are:

- 9600, 19200bps
- Modes 8E1, 8O1, 8N2, 8N1(V3.2)



Due to our policy of continual improvement, specifications may change without prior notice

2 Remarks

2.1 Date & Time

This data block is only accessible as a complete data record. For example, it is not possible to write just the year of the time stamp. This will lead to an exception response "illegal data address".

| Address | Content |
|---------|-----------------------------------|
| 0000h | year 0..99d |
| 0001h | month 1..12d |
| 0002h | day 1..31d |
| 0003h | hour 0..23d |
| 0004h | minute 0..59d |
| 0005h | second 0..59d |
| 0006h | daylight, 1: automatic, 0: manual |

2.2 IsoHub state and SN

This data block is only accessible as a complete data record.

| Address | Content |
|---------|--|
| 0100h | SN IsoHub no.1 |
| 0101h | State IsoHub no.1, 0: not installed, 1: ok, 2: error |
| 0102h | SN IsoHub no.2 |
| 0103h | State IsoHub no.2... |

2.3 IsoBase battery info

| Address | Content |
|---------|------------------------------|
| 0200h | battery state, 0: ok, 1: low |

2.4 Input properties

Each word register can be accessed randomly. A maximum number of 125 word registers can be transferred in one message when reading. When writing the maximum number of registers are 123. Calculation of the start address of a specific input: $\text{address} = \text{start address} + (\text{Inputno} - 1)$

2.5 Input maximum and minimum values

The maximum and minimum values can be written to in order to be reset. The value 0d will reset the maximum values, the value 10000d will reset the minimum values.

2.6 Input states

The input state is represented as value 0 for OK, 1 for warning and 2 for alarm.

2.7 Input memory

The input memory is represented as value 0 for OK and 1 for alarm memory. The alarm memory can be cleared by writing the value 0d.



Due to our policy of continual improvement, specifications may change without prior notice

2.8 Input event counter

The input new event counter can be reset by writing the value 0d.

2.9 Input labels

Each input 1..128 are randomly accessible. The label consists of 8 ASCIIs stored in 8 word registers. Allowed characters are 0..9, A..Z, a..z and <Space>. Calculation of the start address of a specific input:
 $address = 800h + (Inputno. - 1) * 8$

2.10 Event memory

The event memory is too big to fit into the Modbus memory model. Therefore a user-defined function code is used.

Furthermore the memory content is too large to fit into one message, so it is divided into 3 parts.

Each part contains max. 21 events. Oldest events are sent first.

Master request

| | | |
|-----------|---------|--------|
| Address | 1..247d | 1 Byte |
| Function | 100d | 1 Byte |
| Input no. | 1..128d | 1 Byte |
| Part no. | 1..3d | 1 Byte |
| CRC16 | x | 2 Byte |

Isobase response

| | | |
|---------------|---------|--------|
| Address | 1..247d | 1 Byte |
| Function | 100d | 1 Byte |
| Input no. | 1..128d | 1 Byte |
| Part no. | 1..3d | 1 Byte |
| No. of events | 0..21d | 1 Byte |

For each event:

| | | | |
|-----------|----------|--------|-----------------------------|
| New state | 0..2d | 1 Byte | 0: OK, 1: Warning, 2: Alarm |
| Year | 0..99d | 1 Byte | |
| Month | 1..12d | 1 Byte | |
| Day | 1..31d | 1 Byte | |
| Hour | 0..23d | 1 Byte | |
| Minute | 0..59d | 1 Byte | |
| Current | 0..9999d | 2 Byte | |
| CRC16 | x | 2 Byte | |



Due to our policy of continual improvement, specifications may change without prior notice

2.11 Power Memory

The power memory is too big to fit into the Modbus memory model. Therefore a user-defined function code is used.

Furthermore the memory content is too large to fit into one message, so it is divided into 2 parts.

Each part contains max. 32 events. Oldest events are sent first.

Master request

| | | |
|----------|---------|--------|
| Address | 1..247d | 1 Byte |
| Function | 105d | 1 Byte |
| Part no. | 1..2d | 1 Byte |
| CRC16 | x | 2 Byte |

Isobase response

| | | | |
|-----------------|---------|--------|---------------------------|
| Address | 1..247d | 1 Byte | |
| Function | 100d | 1 Byte | |
| Part no. | 1..2d | 1 Byte | |
| No. of events | 0..32d | 1 Byte | |
| For each event: | | | |
| New state | 0..1d | 1 Byte | 0: Power on, 1: Power off |
| Year | 0..99d | 1 Byte | |
| Month | 1..12d | 1 Byte | |
| Day | 1..31d | 1 Byte | |
| Hour | 0..23d | 1 Byte | |
| Minute | 0..59d | 1 Byte | |
| CRC16 | x | 2 Byte | |

2.12 Diagnostics

Only the subfunction echo is supported. There are no message counters implemented in the IsoBase.

2.13 Read device identification

The device ID consists of vendor name, product code and revision number (Basic device identification). The conformity level is 01h.

2.14 Input Filter Active

The input filter state is represented as value 0 for off and 1 for active. The filter can only be activated when the particular input is measured by a IsoHub with TRMS function.

