



Modbus RTU communication protocol for ELR-4C series

INDEX

Modbus RTU protocol..... 2

- Reading multiple registers [function code 03h]..... 2*
- Write multiple registers [function code 10h]..... 3*
- Diagnostic [function code 08h]..... 4*
- Report slave ID [function code 11h] 4*

Registers..... 6

- Measures – Long format..... 6*
- Measures – Float format..... 6*
- Harmonics Input 1..... 7*
- Harmonics Input 2..... 7*
- Harmonics Input 3..... 7*
- Harmonics Input 4..... 7*
- Setup Input 1..... 8*
- Setup Input 2..... 8*
- Setup Input 3..... 8*
- Setup Input 4..... 8*
- Remote reset [trip/warning state]..... 8*
- Remote test..... 8*

Modbus RTU protocol

Modbus is a master-slave communication protocol able to support up to 247 slaves organized as a bus or as a star network. The physical link layer is RS485. The communication is half-duplex. The network messages can be Query-Response or Broadcast type. The Query-Response command is transmitted from the Master to an established Slave and generally it is followed by an answering message. The Broadcast command is transmitted from the Master to all Slaves and is never followed by an answer.

Generic RTU message structure:

| | | |
|------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------|
| START OF FRAME | = | Starting message marker. (silence on line for time >= 4 characters) |
| ADDRESS FIELD [1 CHAR] | = | Includes device address in which you need to communicate in Query-Response mode. In case the message is a Broadcast type it includes 00. |
| FUNCTION CODE [1 CHAR] | = | The operation code that you need to perform. |
| DATA FIELD [N CHAR] | = | Includes the data field. |
| ERROR CHECK [2 CHARS] | = | Field for the error correction code. |
| END OF FRAME | = | End message marker. (silence on line for time >= 4 characters) |

Wait time for response:

Request of 16 register (64 bytes) → typical 15ms / worst 30 ms

Request of 64 register (128 bytes) → typical 15ms / worst 50 ms

Scan rate max recommended: 250 ms

Reading multiple registers [function code 03h]

Reads the binary contents of holding registers (2X references) in the slave.

Broadcast is not supported. The Query message specified the starting register and quantity of register to be read.

QUERY:

| | 0° Byte | 1° Byte | 2-3° Byte | 4-5° Byte | 6-7° Byte | |
|------------------|---------------|---------------------------------------------|---------------|-----------------------------------------|-----------|--------------|
| Start of Frame | Address Field | Function Code | Start Address | Number of Registers | Check Sum | End of Frame |
| START OF FRAME | = | Starting message marker. | | | | |
| ADDRESS FIELD | = | Device address (0x01... 0xF7) | | (1 byte). | | |
| FUNCTION CODE | = | Operation code (0x03) | | (1 byte). | | |
| START ADDRESS | = | First register address to be read | | (2 byte). | | |
| No. OF REGISTERS | = | Number of registers (max 252 bytes) to read | | (4 bytes [1 long] for 1 measure value). | | |
| CHECK SUM | = | Check sum. | | | | |
| END OF FRAME | = | End message marker. | | | | |

WARNING:

It is possible to read more than one variable at the same time (**max 252 bytes**) only if their addresses are consecutive and the variables on the same line cannot be divided.

The register data in the response message are packet as two bytes per register, with the binary contents right justified within each byte.

For each register, the first byte contains the high order bits and the second contains the low order bits.

RESPONSE:

| | 0° Byte | 1° Byte | 2° Byte | n° Byte | n+1 - n+2° Byte | |
|-------------------|---------------|----------------------------------|-----------------|---------------------------------------------|-----------------|--------------|
| Start of Frame | Address Field | Function Code | Number of Bytes | Data | Check Sum | End of Frame |
| START OF FRAME | = | Starting message marker. | | | | |
| ADDRESS FIELD | = | Device address (0x01... 0xF7) | | (1byte). | | |
| FUNCTION CODE | = | Operation code (0x03) | | (1 Byte). | | |
| No. OF SEND BYTES | = | Number of data bytes (0x00...??) | | (1 byte). 1 register requires 2 data bytes. | | |
| D0, D1, ..., Dn | = | data bytes (0x00...??) | | (Nr. of register x 2 = n. byte). | | |
| CHECK SUM | = | Check sum. | | | | |
| END OF FRAME | = | End message marker. | | | | |

Write multiple registers [function code 10h]

Write values into a sequence of holding registers (2X references).

WARNING: It is possible to write more than one variable at the same time only if their addresses are consecutive and the variables on the same line cannot be divided. (max 64 bytes).

QUERY:

| | 0° Byte | 1° Byte | 2-3° Byte | 4-5° Byte | 6° Byte | n° Byte | n+1 - n+2° Byte | |
|----------------|---------------|---------------|---------------|---------------------|-----------------|---------|-----------------|--------------|
| Start of Frame | Address Field | Function Code | Start Address | Number of Registers | Number of Bytes | Data | Check Sum | End of Frame |

| | | | | | | | | |
|------------------------|---|--------------------------------------------------------|--|--|----------------------------------------------------|--|--|--|
| <i>START OF FRAME</i> | = | <i>Starting message marker.</i> | | | | | | |
| <i>ADDRESS FIELD</i> | = | <i>Device address (0x01... 0xF7)</i> | | | <i>(1 byte).</i> | | | |
| <i>FUNCTION CODE</i> | = | <i>Operation code (0x10)</i> | | | <i>(1 byte).</i> | | | |
| <i>START ADDRESS</i> | = | <i>First register address to be written</i> | | | <i>(2 byte).</i> | | | |
| <i>No. OF REGISTER</i> | = | <i>Number of registers to be written (1 to 4, ...)</i> | | | <i>(2 byte).</i> | | | |
| <i>No. OF BYTES</i> | = | <i>Number of data bytes (HEX)</i> | | | <i>(1 byte): 1 register requires 2 data bytes.</i> | | | |
| <i>D0,D1,...,Dn</i> | = | <i>Data bytes (0x00...?)</i> | | | <i>(1 byte) (Nr. of register x 2 = n. byte).</i> | | | |
| <i>CHECK SUM</i> | = | <i>Check sum.</i> | | | | | | |
| <i>END OF FRAME</i> | = | <i>End message marker.</i> | | | | | | |

The normal response returns the slave address, function code, starting address and quantity of register preset.

RESPONSE:

| | 0° Byte | 1° Byte | 2-3° Byte | 4-5° Byte | 6-7° Byte | |
|----------------|---------------|---------------|---------------|---------------------|-----------|--------------|
| Start of Frame | Address Field | Function Code | Start Address | Number of Registers | Check Sum | End of Frame |

| | | | | | | |
|------------------------|---|---------------------------------------------|--|--|------------------|--|
| <i>START OF FRAME</i> | = | <i>Starting message marker.</i> | | | | |
| <i>ADDRESS FIELD</i> | = | <i>Device address (0x01... 0xF7)</i> | | | <i>(1 byte).</i> | |
| <i>FUNCTION CODE</i> | = | <i>Operation code (0x10)</i> | | | <i>(1 byte).</i> | |
| <i>START ADDRESS</i> | = | <i>First register address to be written</i> | | | <i>(2 byte).</i> | |
| <i>No. OF REGISTER</i> | = | <i>Number of registers to be written</i> | | | <i>(2 byte).</i> | |
| <i>ERROR CHECK</i> | = | <i>Check sum.</i> | | | | |
| <i>END OF FRAME</i> | = | <i>End message marker.</i> | | | | |

BROADCAST COMMAND:

It is possible to send a broadcast command (Address Field equal 0x00) for all write command.

QUERY:

| | 0° Byte | 1° Byte | 2-3° Byte | 4-5° Byte | 6° Byte | n° Byte | n+1 - n+2° Byte | |
|----------------|---------|---------------|---------------|---------------------|-----------------|---------|-----------------|--------------|
| Start of Frame | 0x00 | Function Code | Start Address | Number of Registers | Number of Bytes | Data | Check Sum | End of Frame |

RESPONSE: *No Response.*

Diagnostic [function code 08h]

This function provides a test for checking the communication system. Broadcast is not supported.

The instrument's protocol has only the sub-function 0 of the diagnostics sub-functions set of the standard modbus protocol.

The Query and the Response messages are the following:

QUERY:

| Start of Frame | 0° Byte Address Field | 1° Byte Function Code | 2-3° Byte Sub Function | n° Byte Data | n+1 - n+2° Byte Check Sum | End of Frame |
|----------------|--------------------------|--------------------------|---------------------------|-----------------|------------------------------|--------------|
|----------------|--------------------------|--------------------------|---------------------------|-----------------|------------------------------|--------------|

START OF FRAME = Starting message marker.
ADDRESS FIELD = Device address (0x01... 0xF7) (1 byte).
FUNCTION CODE = Operation code (0x08 HEX) (1 byte).
SUB FUNCTION = Sub-function 0 (0x00 0x00) (2 byte).
DATA = Max 10 data bytes.
CHECK SUM = Check sum.
END OF FRAME = End message marker.

RESPONSE:

The response must be the loopback of the same data.

| Start of Frame | 0° Byte Address Field | 1° Byte Function Code | 2-3° Byte Sub Function | n° Byte Data | n+1 - n+2° Byte Check Sum | End of Frame |
|----------------|--------------------------|--------------------------|---------------------------|-----------------|------------------------------|--------------|
|----------------|--------------------------|--------------------------|---------------------------|-----------------|------------------------------|--------------|

START OF FRAME = Starting message marker.
ADDRESS FIELD = Device address (0x01... 0xF7) (1 byte).
FUNCTION CODE = Operation code (0x08 HEX) (1 byte).
SUB FUNCTION = Sub-function 0 (0x00 0x00) (2 byte).
DATA = Data bytes.
CHECK SUM = Check sum.
END OF FRAME = End message marker.

DIAGNOSTIC EXAMPLE:

| QUERY | | RESPONSE | |
|-------------------|---------------|-------------------|---------------|
| Field Name | Example (Hex) | Field Name | Example (Hex) |
| Slave Address | 0x01 | Slave Address | 0x01 |
| Function Code | 0x08 | Function Code | 0x08 |
| Sub-function Hi | 0x00 | Sub-function Hi | 0x00 |
| Sub-function Lo | 0x00 | Sub-function Lo | 0x00 |
| Data Hi | 0xF1 | Data Hi | 0xF1 |
| Data Lo | 0xA7 | Data Lo | 0xA7 |
| Error Check (CRC) | 0x?? | Error Check (CRC) | 0x?? |
| | 0x?? | | 0x?? |

Report slave ID [function code 11h]

This function returns the type of the instrument and the current status of the slave run indicator. Broadcast is not supported.

The Query and the Response messages are the following:

QUERY:

| Start of Frame | 0° Byte Address Field | 1° Byte Function Code | 2 - 3° Byte Check Sum | End of Frame |
|----------------|--------------------------|--------------------------|--------------------------|--------------|
|----------------|--------------------------|--------------------------|--------------------------|--------------|

START OF FRAME = Starting message marker.
ADDRESS FIELD = Device address (0x01... 0xF7) (1 byte).
FUNCTION CODE = Operation code (0x11) (1 byte).
CHECK SUM = Check sum.
END OF FRAME = End message marker.

RESPONSE:

| Start of Frame | 0° Byte Address Field | 1° Byte Function Code | 2° Byte Byte Count | 3° Byte Slave ID | 4° Byte Run Indicator Status | 5° - 6° Byte Check Sum | End of Frame |
|----------------|--------------------------|--------------------------|-----------------------|---------------------|---------------------------------|---------------------------|--------------|
|----------------|--------------------------|--------------------------|-----------------------|---------------------|---------------------------------|---------------------------|--------------|

START OF FRAME = Starting message marker.
ADDRESS FIELD = Device address (0x01... 0xF7) (1 byte).
FUNCTION CODE = Operation code (0x11) (1 byte).
BYTE COUNT = Number of data bytes (0x02) (1 byte).
SLAVE ID = Slave ID identifier (0x73) (1 byte).
RUN INDICATOR STATUS = Run indicator status (0xFF) (1 byte).
DATA = Data bytes.
CHECK SUM = Check sum.
END OF FRAME = End message marker.

The normal response has the slave ID identifier (0x73) and the run indicator Status (0xFF).

REPORT SLAVE ID EXAMPLE:

| QUERY | | RESPONSE | |
|-------------------|---------------|----------------------|---------------|
| Field Name | Example (Hex) | Field Name | Example (Hex) |
| Slave Address | 0xXX | Slave Address | 0x01 |
| Function Code | 0x11 | Function Code | 0x11 |
| Error Check (CRC) | 0x?? | Byte count | 0x02 |
| | 0x?? | Slave ID | 0x73 |
| | | Run indicator status | 0xFF |
| | | Error Check (CRC) | 0x?? |
| | | | 0x?? |

Error message from slave to master

When a slave device receives a not valid query, it does transmit an error message.

RESPONSE:

| Start of Frame | 0° Byte Address Field | 1° Byte Function Code | 2° Byte Error Code | 3 - 4° Byte Check Sum | End of Frame |
|-----------------------|--------------------------|-------------------------------------------------|-----------------------|--------------------------|--------------|
| <i>START OF FRAME</i> | = | <i>Starting message marker.</i> | | | |
| <i>ADDRESS FIELD</i> | = | <i>Device address (0x01... 0xF7)</i> | <i>(1 byte).</i> | | |
| <i>FUNCTION CODE</i> | = | <i>Operation code with bit 7 high</i> | <i>(1 byte).</i> | | |
| <i>ERROR CODE</i> | = | <i>Message containing communication failure</i> | <i>(1 byte).</i> | | |
| <i>CHECK SUM</i> | = | <i>Check sum.</i> | | | |
| <i>END OF FRAME</i> | = | <i>End message marker.</i> | | | |

ERROR EXAMPLE:

QUERY

| Field Name | Example (Hex) |
|---------------------|---------------|
| Slave Address | 0x01 |
| Function Code | 0x03 |
| Starting Address Hi | 0x00 |
| Starting Address Lo | 0x00 |
| Number Of Word Hi | 0x00 |
| Number Of Word Lo | 0x05 |
| Error Check (CRC) | 0x?? |
| | 0x?? |

RESPONSE

| Field Name | Example (Hex) |
|-------------------|---------------|
| Slave Address | 0x 01 |
| Function Code | 0x83 (1) |
| Error Code | 0x02 (2) |
| Error Check (CRC) | 0x?? |
| | 0x?? |

(1): Function Code transmitted by master with bit 7 high.

(2): Error type:

0x01 = Illegal Function

0x02 = Illegal data address

0x03 = Illegal data value

0x0F = Communication Protection Enabled
(password enabled)

Write PASSWORD parameter before retry.

Registers

The following tables shown all the device registers.

Measures – Long format

| Register ^{HEX} | Register ^{DEC} | Word | Description | R/W | M.U. | Type |
|-------------------------|-------------------------|------|------------------------|-----|-------------------------------------------------------------------------------------------------|----------|
| 100 | 256 | 2 | CURRENT 1 | R | mA | Unsigned |
| 102 | 258 | 2 | CURRENT 2 | R | mA | Unsigned |
| 104 | 260 | 2 | CURRENT 3 | R | mA | Unsigned |
| 106 | 262 | 2 | CURRENT 4 | R | mA | Unsigned |
| 108 | 264 | 2 | CURRENT FILTERED 1 | R | mA | Unsigned |
| 10A | 266 | 2 | CURRENT FILTERED 2 | R | mA | Unsigned |
| 10C | 268 | 2 | CURRENT FILTERED 3 | R | mA | Unsigned |
| 10E | 270 | 2 | CURRENT FILTERED 4 | R | mA | Unsigned |
| 110 | 272 | 2 | MAX CURRENT 1 | R | mA | Unsigned |
| 112 | 274 | 2 | MAX CURRENT 2 | R | mA | Unsigned |
| 114 | 276 | 2 | MAX CURRENT 3 | R | mA | Unsigned |
| 116 | 278 | 2 | MAX CURRENT 4 | R | mA | Unsigned |
| 118 | 280 | 2 | MAX CURRENT FILTERED 1 | R | mA | Unsigned |
| 11A | 282 | 2 | MAX CURRENT FILTERED 2 | R | mA | Unsigned |
| 11C | 284 | 2 | MAX CURRENT FILTERED 3 | R | mA | Unsigned |
| 11E | 286 | 2 | MAX CURRENT FILTERED 4 | R | mA | Unsigned |
| 120 | 288 | 2 | THD 1 | R | % * 100 | Unsigned |
| 122 | 290 | 2 | THD 2 | R | % * 100 | Unsigned |
| 124 | 292 | 2 | THD 3 | R | % * 100 | Unsigned |
| 126 | 294 | 2 | THD 4 | R | % * 100 | Unsigned |
| 128 | 296 | 2 | CREST FACTOR 1 | R | [thousandths] | Unsigned |
| 12A | 298 | 2 | CREST FACTOR 2 | R | [thousandths] | Unsigned |
| 12C | 300 | 2 | CREST FACTOR 3 | R | [thousandths] | Unsigned |
| 12E | 302 | 2 | CREST FACTOR 4 | R | [thousandths] | Unsigned |
| 130 | 304 | 2 | STATUS 1 | R | bit 0 set: ALARM bit 1 set: TRIP bit 2 set: OPEN bit 3 set: DISABLE bit 4 set: OVER | - |
| 132 | 306 | 2 | STATUS 2 | R | See STATUS 1 | - |
| 134 | 308 | 2 | STATUS 3 | R | See STATUS 1 | - |
| 136 | 310 | 2 | STATUS 4 | R | See STATUS 1 | - |

Measures – Float format

| Register ^{HEX} | Register ^{DEC} | Word | Description | R/W | M.U. | Type |
|-------------------------|-------------------------|------|------------------------|-----|---------|----------|
| 200 | 512 | 2 | CURRENT 1 | R | mA | Float |
| 202 | 514 | 2 | CURRENT 2 | R | mA | Float |
| 204 | 516 | 2 | CURRENT 3 | R | mA | Float |
| 206 | 518 | 2 | CURRENT 4 | R | mA | Float |
| 208 | 520 | 2 | CURRENT FILTERED 1 | R | mA | Float |
| 20A | 522 | 2 | CURRENT FILTERED 2 | R | mA | Float |
| 20C | 524 | 2 | CURRENT FILTERED 3 | R | mA | Float |
| 20E | 526 | 2 | CURRENT FILTERED 4 | R | mA | Float |
| 210 | 528 | 2 | MAX CURRENT 1 | R | mA | Unsigned |
| 212 | 530 | 2 | MAX CURRENT 2 | R | mA | Unsigned |
| 214 | 532 | 2 | MAX CURRENT 3 | R | mA | Unsigned |
| 216 | 534 | 2 | MAX CURRENT 4 | R | mA | Unsigned |
| 218 | 536 | 2 | MAX CURRENT FILTERED 1 | R | mA | Unsigned |
| 21A | 538 | 2 | MAX CURRENT FILTERED 2 | R | mA | Unsigned |
| 21C | 540 | 2 | MAX CURRENT FILTERED 3 | R | mA | Unsigned |
| 21E | 542 | 2 | MAX CURRENT FILTERED 4 | R | mA | Unsigned |
| 220 | 544 | 2 | THD 1 | R | % * 100 | Float |
| 222 | 546 | 2 | THD 2 | R | % * 100 | Float |
| 224 | 548 | 2 | THD 3 | R | % * 100 | Float |
| 226 | 550 | 2 | THD 4 | R | % * 100 | Float |
| 228 | 552 | 2 | CREST FACTOR 1 | R | - | Float |
| 22A | 554 | 2 | CREST FACTOR 2 | R | - | Float |
| 22C | 556 | 2 | CREST FACTOR 3 | R | - | Float |
| 22E | 558 | 2 | CREST FACTOR 4 | R | - | Float |

Harmonics Input 1

| Register ^{HEX} | Register ^{DEC} | Word | Description | R/W | Measure Unit | Type |
|-------------------------|-------------------------|------|-------------------------------------------------|-----|--------------|----------|
| 1000 | 4096 | 2 | 1 ST HARMONIC (<i>Fundamental</i>) | R | % * 100 | Unsigned |
| 1002 | 4098 | 2 | 2 ND HARMONIC | R | % * 100 | Unsigned |
| 1004 | 4100 | 2 | 3 RD HARMONIC | R | % * 100 | Unsigned |
| 1006 | 4102 | 2 | 4 TH HARMONIC | R | % * 100 | Unsigned |
| 1008 | 4104 | 2 | 5 TH HARMONIC | R | % * 100 | Unsigned |
| --- | --- | --- | --- | --- | --- | --- |
| 107C | 4220 | 2 | 63 TH HARMONIC | R | % * 100 | Unsigned |

Harmonics Input 2

| Register ^{HEX} | Register ^{DEC} | Word | Description | R/W | Measure Unit | Type |
|-------------------------|-------------------------|------|-------------------------------------------------|-----|--------------|----------|
| 1100 | 4352 | 2 | 1 ST HARMONIC (<i>Fundamental</i>) | R | % * 100 | Unsigned |
| 1102 | 4354 | 2 | 2 ND HARMONIC | R | % * 100 | Unsigned |
| 1104 | 4356 | 2 | 3 RD HARMONIC | R | % * 100 | Unsigned |
| 1106 | 4358 | 2 | 4 TH HARMONIC | R | % * 100 | Unsigned |
| 1108 | 4360 | 2 | 5 TH HARMONIC | R | % * 100 | Unsigned |
| --- | --- | --- | --- | --- | --- | --- |
| 117C | 4476 | 2 | 63 TH HARMONIC | R | % * 100 | Unsigned |

Harmonics Input 3

| Register ^{HEX} | Register ^{DEC} | Word | Description | R/W | Measure Unit | Type |
|-------------------------|-------------------------|------|-------------------------------------------------|-----|--------------|----------|
| 1200 | 4608 | 2 | 1 ST HARMONIC (<i>Fundamental</i>) | R | % * 100 | Unsigned |
| 1202 | 4610 | 2 | 2 ND HARMONIC | R | % * 100 | Unsigned |
| 1204 | 4612 | 2 | 3 RD HARMONIC | R | % * 100 | Unsigned |
| 1206 | 4614 | 2 | 4 TH HARMONIC | R | % * 100 | Unsigned |
| 1208 | 4616 | 2 | 5 TH HARMONIC | R | % * 100 | Unsigned |
| --- | --- | --- | --- | --- | --- | --- |
| 127C | 4732 | 2 | 63 TH HARMONIC | R | % * 100 | Unsigned |

Harmonics Input 4

| Register ^{HEX} | Register ^{DEC} | Word | Description | R/W | Measure Unit | Type |
|-------------------------|-------------------------|------|-------------------------------------------------|-----|--------------|----------|
| 1300 | 4864 | 2 | 1 ST HARMONIC (<i>Fundamental</i>) | R | % * 100 | Unsigned |
| 1302 | 4866 | 2 | 2 ND HARMONIC | R | % * 100 | Unsigned |
| 1304 | 4868 | 2 | 3 RD HARMONIC | R | % * 100 | Unsigned |
| 1306 | 4870 | 2 | 4 TH HARMONIC | R | % * 100 | Unsigned |
| 1308 | 4872 | 2 | 5 TH HARMONIC | R | % * 100 | Unsigned |
| --- | --- | --- | --- | --- | --- | --- |
| 137C | 4988 | 2 | 63 TH HARMONIC | R | % * 100 | Unsigned |

Setup Input 1

Warning: Must be send the entire parameter length (2 words or 1 word – see the long of each parameter) for a correct command setting.

| Register HEX | Register DEC | Word | Description | R/W | Parameters | Default |
|--------------|--------------|------|-----------------------------|-----|-----------------------------------------------------------------------|---------|
| 2000 | 8192 | 2 | ENABLE | R/W | 0: disable 1: enable | 1 |
| 2002 | 8194 | 2 | FILTER | R/W | 0: disable 1: enable | 0 |
| 2004 | 8196 | 2 | TRIP - mA | R/W | 30 ÷ 30'000 | 30 |
| 2006 | 8198 | 2 | TRIP DELAY - ms | R/W | 20 ÷ 10'000 <i>Note: the value must be a multiple of 20 ms.</i> | 20 |
| 2008 | 8200 | 2 | TRIP RECOVERY MODE | R/W | 0: manual 1: automatic | 0 |
| 200A | 8202 | 2 | WARNING DELAY - ms | R/W | 20 ÷ 1'000'000 <i>Note: the value must be a multiple of 20 ms.</i> | 20 |
| 200C | 8204 | 2 | WARNING PERCENT | R/W | 20 ÷ 90 percent of trip | 50 |
| 200E | 8206 | 2 | WARNING RECOVERY MODE | R/W | 0: manual 1: automatic | 1 |
| 2010 | 8208 | 2 | HYSTERESIS [TRIP & WARNING] | R/W | 95 ÷ 50% over the limit set | 90 |
| 2012 | 8210 | 2 | FAIL SAFE | R/W | 0: disable 1: enable | 0 |

Setup Input 2: 2100^{HEX} to 2113^{HEX} / 8448^{DEC} to 8467^{DEC}

Setup Input 3: 2200^{HEX} to 2213^{HEX} / 8704^{DEC} to 8723^{DEC}

Setup Input 4: 2300^{HEX} to 2313^{HEX} / 8960^{DEC} to 8979^{DEC}

Remote reset [trip/warning state]

Warning: Must be send the entire parameter length (2 words or 1 word – see the long of each parameter) for a correct command setting.

| Register HEX | Register DEC | Word | Description | R/W | Parameters |
|--------------|--------------|------|----------------------|-----|-------------------------------------------------------------|
| 2A00 | 10752 | 2 | MANUAL RESET INPUT 1 | W | Write 0x0A0A ^{hex} (2570 ^{dec}) to reset |
| 2A02 | 10754 | 2 | MANUAL RESET INPUT 2 | W | Write 0x0A0A ^{hex} (2570 ^{dec}) to reset |
| 2A04 | 10756 | 2 | MANUAL RESET INPUT 3 | W | Write 0x0A0A ^{hex} (2570 ^{dec}) to reset |
| 2A06 | 10758 | 2 | MANUAL RESET INPUT 4 | W | Write 0x0A0A ^{hex} (2570 ^{dec}) to reset |

Remote test

Warning: Must be send the entire parameter length (2 words or 1 word – see the long of each parameter) for a correct command setting.

| Register HEX | Register DEC | Word | Description | R/W | Parameters |
|--------------|--------------|------|---------------------|-----|-------------------------------------------------------------|
| 2A20 | 10784 | 2 | MANUAL TEST INPUT 1 | W | Write 0x5050 ^{hex} (20560 ^{dec}) to test |
| 2A22 | 10786 | 2 | MANUAL TEST INPUT 2 | W | Write 0x5050 ^{hex} (20560 ^{dec}) to test |
| 2A24 | 10788 | 2 | MANUAL TEST INPUT 3 | W | Write 0x5050 ^{hex} (20560 ^{dec}) to test |
| 2A26 | 10790 | 2 | MANUAL TEST INPUT 4 | W | Write 0x5050 ^{hex} (20560 ^{dec}) to test |



I-26900 Lodi - ITALY - Via S. Fereolo, 9
Tel. +39 0371 30207 / 30761 Fax +39 0371 32819
<http://www.contrel.it> - E-mail: contrel@contrel.it