



**REL-0808 User Guide**

## Table of Contents

|         |   |    |
|---------|---|----|
| 1.      | Operational Instructions.....   | 2  |
| 2.      | Purpose.....  | 2  |
| 3.      | Specifications.....   | 2  |
| 4.      | Resetting Module .....  | 3  |
| 5.      | Module Control Types .....  | 3  |
| 6.      | Module Operational Modes .....  | 3  |
| 7.      | Module Status Indication .....  | 4  |
| 8.      | Module Configuration .....  | 4  |
| 9.      | Module Address Selection.....   | 5  |
| 10.     | Module Startup Procedure.....   | 6  |
| 11.     | Typical Connection Schema.....  | 6  |
| 12.     | ModBus RTU Protocol Functional Characteristics.....                   | 7  |
| 12.1.   | Read channels state (0x01 – Read Coils).....                          | 7  |
| 12.1.1. | Request .....   | 7  |
| 12.1.2. | Response .....  | 7  |
| 12.2.   | Turn single channel ON/OFF (0x05 – Write Single Coil) .....           | 8  |
| 12.2.1. | Request .....   | 8  |
| 12.2.2. | Response .....  | 8  |
| 12.3.   | Turn multiple channels ON/OFF (0x0F – Write Multiple Coils) .....     | 8  |
| 12.3.1. | Request .....   | 8  |
| 12.3.2. | Response .....  | 9  |
| 12.4.   | Reading holding registers (0x03 – Read Holding Registers) .....       | 9  |
| 12.4.1. | Request .....   | 9  |
| 12.4.2. | Response .....  | 9  |
| 12.5.   | Writing single holding register (0x06 – Write Single Registers) ..... | 10 |
| 12.5.1. | Request .....   | 10 |
| 12.5.2. | Response .....  | 10 |
| 12.6.   | Report Slave ID (0x11 – Module Serial Number).....                    | 10 |
| 12.7.   | Read module serial number (0x64 – Module Serial Number).....          | 10 |
| 12.7.1. | Request .....   | 10 |
| 12.7.2. | Response .....  | 10 |
| 12.8.   | Possible error codes .....  | 10 |
| 13.     | Collisions Between Manual Control And Master .....                    | 11 |

## 1. Operational Instructions

Ahatos REL-0808 module should be operated in an environment, not beyond the limit values (see Specifications).

Manufacturer prohibits usage of module in applications that can endanger human life by their action or inaction.

Ahatos REL-0808 Module's manufacturer is not liable for consequences of exploitation of the equipment connected to module.

User must make sure there are no short circuits caused by wires connected to device.

**It is strictly prohibited to plug or unplug wires those are under high voltage! It is strictly prohibited to disassemble module and touch electrical parts and/or violate the integrity of external electrical connections!**

## 2. Purpose

Relay module Ahatos REL-0808 is designed for load control, and consists of eight independent channels.

## 3. Specifications

| Parameter   | Value   |
|---|---|
| Control channels  | 4/8   |
| Maximum channels voltage                                | 250 VAC<br>30 VDC   |
| Maximum current per channel                             | 8 A   |
| Control interface type                                  | RS-485  |
| - Maximum RS-485 network segment length                 | 1200 meters   |
| - Type of devices identification inside network segment | Addressing  |
| - Logical level protocol                                | MODBUS RTU  |
| - Bus settings  | 19.2kbit/s,<br>8 data bits,<br>even parity,<br>1 stop bit |
| Power supply  | +12 VDC   |
| Maximum consumption current                             | 350 mA  |
| Working temperature                                     | +10..50 °C  |
| Allowed humidity  | 5..80 %   |
| Protection class, IEC-952                               | IP20  |
| Weight  | 350 g   |
| Dimensions  | 162 x 90 x 58 mm  |
| Inputs / outputs  |   |
| - Low voltage wires (inputs)                            | Max. 16AWG (1.5 mm <sup>2</sup> )                         |
| - High voltage wires (outputs)                          | Max. 12AWG (4 mm <sup>2</sup> )                           |

## 4. Resetting Module

Multifunction RESET button is used to reset module and to reset module's configuration to factory values.

If RESET button is pressed and being held less than 4 seconds, then module will be rebooted after button is released.

If RESET button is being held more than 4 seconds, then module will reset configuration to factory values and perform boot when button is released. Also, please note that BUSY LED will light up when after 4 seconds when module had reset configuration.

## 5. Module Control Types

User can control channels in two ways:

1. **Low-current push buttons.** Principle of operation of low-voltage push-buttons: manual input channel is a TTL level (with +5 V and a current of several milliamperes), which expects a circuit to a common signal GND. Channel input is considered activated in a moment when button is pressed. At the same time, channel input is deactivated when button is released. So it is required to use pushbuttons without fixation and with normally open contacts
2. **By MODBUS protocol via RS485 bus.** For supported features see "*Functional characteristics point protocol MODBUS RTU*" section of this user guide.

## 6. Module Operational Modes

The module supports two modes of operation (see "*Module Configuration*" section for details):

1. **Asynchronous** - all channels are controlled independently. Two sub-modes are available:
  - Channel state is being switched to opposite by pressing corresponding button;
  - Channel is active while corresponding button is being held pressed
2. **Pairwise** - all channels are combined into a pairs of mutually exclusive relays (for control of blinds and rolling shutters). Channels are combined as following: 1-2, 3-4, 5-6, 7-8. Only one channel of the pair can be enabled at one time. Also, it will automatically turn off after a set period of time. This time period is set as a request parameter. Time can be set between 0..3600 seconds. (see "*Functional Specifications Protocol MODBUS RTU*", recording parameters in registers).

In order to turn off channel from particular pair, second channel should be activated. E.g., if channel #1 was activated, it can be deactivated by timeout or by activating the channel #2 (push corresponding push-button). In order to switch from channel #1 to channel #2, you must double-press button that corresponds to channel #2 (in case MODBUS protocol is used to control channels, then it is possible to switch channels directly from channel #1 to channel #2 without an intermediate step). Also, configuration bit 7 is responsible for controlling user's ability to change direction without necessity to double-press opposite channel's button.

While operating using buttons it has no matter how long button is being pushed and held, the channel will be enabled for a certain period of time only. Also, the channel will not be deactivated when pressing the same button once again. If the button is held, but a request from the master is received to change the state of the channel, then this request will be processed despite the fact that the button pressed and is being held pressed by the user.

If there is no communication with the master via RS485 interface, channels continue to operate using control inputs (push-buttons). When module loses power, all channels will be switched OFF. Also, channels states may be restored to previous states when power supply is attached and/or power is restored. See "*Module configuration*" (switches 2 and 3).

## 7. Module Status Indication

Module input power LED indicates whether appropriate power source is attached to module:

- **OFF** – appropriate power source is not attached
- **ON** – module has appropriate power source attached

Channels’ green LEDs are indicating whether particular channel is active:

- **OFF** – particular channel is not active
- **ON** – particular channel is active (relay is closed)

STATUS green LED can be in the following states:

- **OFF** – there is no power, or failure;
- **Flashes three times per second** - module works, there is no communication via RS-485 or illegitimate query received from master (error CRC16, malformed packet size).  
If this LED flashed along with BUSY LED on module startup, then it indicates module is in error mode because of address is incorrectly configured.
- **ON** – module is OK, communication with master established via RS-485.

BUSY red LED indicates whether module is busy with some task:

- **OFF** – there is no power, or has power and module is idle
- **Flashes three times per second** - module is in busy state, performs the function.  
If this LED flashed along with STATUS LED on module startup, then it indicates module is in error mode because of address is incorrectly configured.
- **ON** – module has reset it’s configuration to factory defaults

If “activate/deactivate channel” request comes from master or inputs, the BUSY indicator flashes three times. If commands are continue to be coming (from inputs to the master), the indicator continues to flash until module will process last command.

In pairwise mode, BUSY LED blinks all the time channel is active (time interval).

## 8. Module Configuration

In order to determine module configuration, switch block should be used. It is located under enclosure’s top plate which needs to be removed (this will open access to configuration jumper-switches).

| Switch #         | Destination                               | Status | Comments   |
|------------------|---|--------|--|
| 1                | Channels mode                             | ON     | Pairwise operational mode (4 channels)   |
|                  |   | OFF    | Asynchronous mode (8 channels)   |
| 2 <sup>I</sup>   | Setting channels state after power outage | ON     | All channels will be turned ON (if switch 3 is ON, all channels will restore previous states)  |
|                  |   | OFF    | All channels will be turned OFF  |
| 3 <sup>I</sup>   | Restore                                   | ON     | If switch 2 is ON, then all channels will restore their previous states<br>If switch 2 is OFF, then all channels will remain OFF in accordance to switch 2 description |
|                  |   | OFF    | All channels will remain in state defined in switch 2 description  |
| 4 <sup>II</sup>  | RS-485 communication presence             | ON     | If communication is not present for longer than 10 seconds, all channels will be turned OFF  |
|                  |   | OFF    | Does not affect channels’ states   |
| 5 <sup>III</sup> | Channels operation mode                   | ON     | When the input is activated, the channel state is reversed   |
|                  |   | OFF    | Channel will be activated while input is active  |
| 6 <sup>IV</sup>  | User priority                             | ON     | User has top priority. It means that if user holds button, then master requests will not be processed and error response will be sent                                  |

| Switch #       | Destination                                   | Status | Comments  |
|----------------|---|--------|---|
|                |   | OFF    | Master/module has priority over user. It means that if user holds button, then master requests will be processed ignoring user presence   |
| 7 <sup>v</sup> | Autoswitch/Change direction in paired channel | ON     | If one of paired channels is turned on and user will press opposite pair member button, then direction will be changed to opposite after specified timeout (can be configured in module register) |
|                |   | OFF    | Does not influence on channels operation  |
| 8              | Option A                                      | N/A    | Reserved for future use   |

<sup>i</sup> - only for asynchronous mode;

<sup>ii</sup> - if channel is being currently controlled by button (input), then channel's state will not change;

<sup>iii</sup> - does not affect the control via RS-485. Used only for asynchronous mode.

<sup>iv</sup> – affects both master and user. It means that if module would like to turn OFF particular channel (e.g. connection to master was lost) but user is holding button, then channel will NOT be turned OFF

<sup>v</sup> – only for pairwise mode. Does not affect the control via RS-485.

After changing the configuration, module needs to be restarted using RESET button.

## 9. Module Address Selection

In order to be able selecting/setting module address, enclosure's top plate needs to be removed (this will open access to address jumper-switches).

| Jumper # | State/Value    | Description  |
|----------|----------------|--|
| 1        | 2 <sup>7</sup> | If jumper is "OFF" – corresponding bit is "0",<br>If jumper is "ON" – corresponding bit is "1" |
| 2        | 2 <sup>6</sup> |  |
| 3        | 2 <sup>5</sup> |  |
| 4        | 2 <sup>4</sup> |  |
| 5        | 2 <sup>3</sup> |  |
| 6        | 2 <sup>2</sup> |  |
| 7        | 2 <sup>1</sup> |  |
| 8        | 2 <sup>0</sup> |  |

If address was changed, then you must reboot module using the RESET button.

Allowed addresses range is from 1 to 247. If address was set to not allowed value, then module will not startup. Busy and Status LEDs will blink constantly indicating module is in error mode because of address value is incorrect. User will need to set correct address value and reset module by turning power OFF and then ON (reset button will not work in this case).

### Addresses Examples:

| Address (decimal) | Address (binary) | Address (dec) | Address (bin) |
|-------------------|------------------|---------------|---------------|
| 16                | 00010000         | 24            | 00011000      |
| 17                | 00010001         | 25            | 00011001      |
| 18                | 00010010         | 26            | 00011010      |
| 19                | 00010011         | 27            | 00011011      |
| 20                | 00010100         | 28            | 00011100      |
| 21                | 00010101         | 29            | 00011101      |
| 22                | 00010110         | 30            | 00011110      |
| 23                | 00010111         | 31            | 00011111      |

## 10. Module Startup Procedure

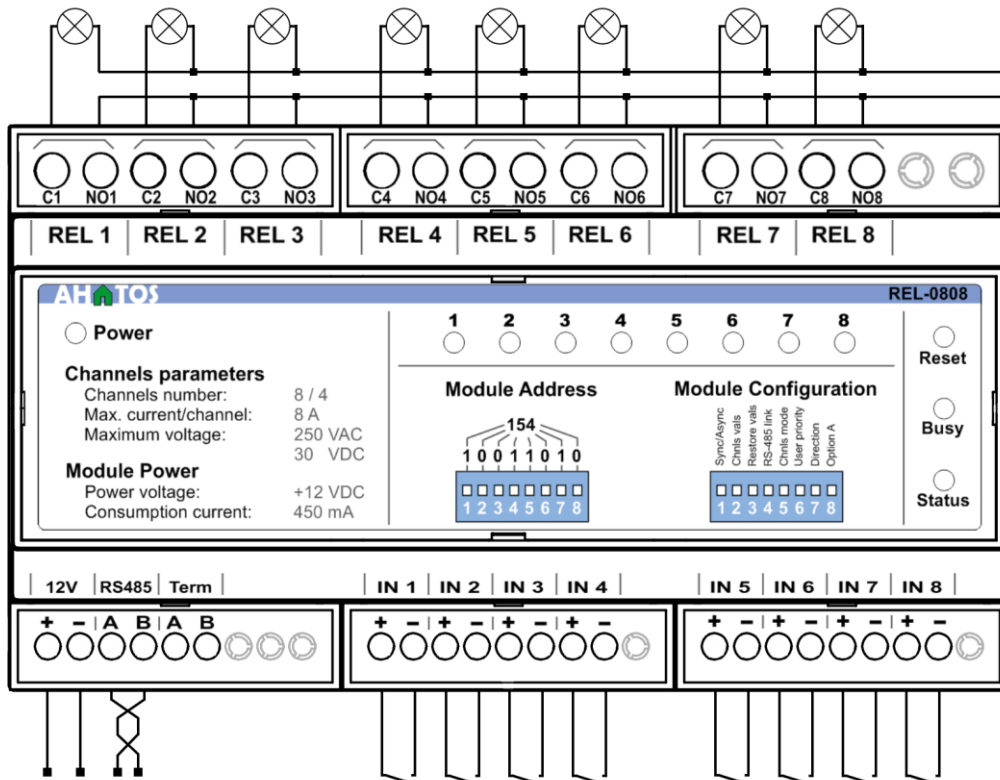
Module should be installed by qualified personnel only. Startup procedure includes:

- Module deployment/installation;
- Address and operational parameters configuration;
- Connecting external electrical circuits.

External electrical circuits/wires should be connected in the following order:

- Turn off high-voltage part;
- Check if there is a short circuit in the load circuit of the high-voltage module according to the scheme drawn up in advance;
- Connect low-voltage switches/buttons;
- Connect RS-485 interface and terminating resistor (if device is the last one on the bus) wires;
- Connect de-energized DC power supply (+12V, 350 mA or more);
- Turn ON power supply;
- Make sure corresponding LEDs are indicating correct module operational status;
- Turn ON high-voltage part.

## 11. Typical Connection Schema



If relay module is connected to the end of RS-485 bus, then you need to connect terminal resistor 120ohm to TERM\_A and TERM\_B.

## 12. ModBus RTU Protocol Functional Characteristics

REL-0808 module implements MODBUS RTU protocol in order to communicate with master device. Please make sure RS-485 bus is configured according to REL-0808 requirements (see “Specifications” section for details).

Module supports the following ModBus RTU functions:

| Function # | Description                                  | Applies to channels operating mode(s) | Allowed via broadcast |
|------------|--|---------------------------------------|-----------------------|
| 0x01       | Read channels state                          | ALL                                   | No                    |
| 0x03       | Read holding registers                       | ALL                                   | No                    |
| 0x05       | Turn ON/OFF single channel                   | ALL                                   | Yes                   |
| 0x06       | Write holding registers                      | ALL                                   | Yes                   |
| 0x0F       | Turn ON/OFF multiple channels simultaneously | Asynchronous mode                     | Yes                   |
| 0x11       | Report Slave ID                              | ALL                                   | No                    |
| 0x64       | Read module serial number                    | ALL                                   | No                    |

General ModBus request and response format is as following:

| Section         | Header         |                        | Data                   | Footer       |
|-----------------|----------------|------------------------|------------------------|--------------|
| Field           | Device Address | ModBus Function Number | Function specific data | CRC16 ModBus |
| Possible values | 0x00..0xF7     | See following sections | Depends on function    | Calculated   |
| Size            | 1 byte         | 1 byte                 |                        | 2 bytes      |

Please note that if *Device Address* is 0 then it means request is broadcast. If request’s ModBus function will support broadcast, then module will perform required actions and will NOT send response to master.

### 12.1. Read channels state (0x01 – Read Coils)

#### 12.1.1. Request

| Section         | Header | Data                    |            |                          |            | Footer |
|-----------------|--------|-------------------------|------------|--------------------------|------------|--------|
| Field           |        | Start Channel # to read |            | Channels numbers to read |            |        |
|                 |        | High byte               | Low byte   | High byte                | Low byte   |        |
| Possible values |        | 0x00                    | 0x00..0x07 | 0x00                     | 0x01..0x08 |        |
| Size            | 1 byte | 1 byte                  | 1 byte     | 1 byte                   |            |        |

Note: “Channels number to read” value should be calculated as 8 minus “Start Channel # to read”

**Example:** 0x01 0x01 0x00 0x00 0x00 0x08 0x3D 0xCC;

#### 12.1.2. Response

| Section         | Header | Data                        |                 | Footer |
|-----------------|--------|-----------------------------|-----------------|--------|
| Field           |        | Number of bytes in response | Channels status |        |
| Possible values |        | 0x01                        | 0x00..0xFF      |        |
| Size            |        | 1 byte                      | 1 byte          |        |

Channels status is placed in corresponding bits of channels status byte. If all eight channels’ status was requested, then all eight bits will be filled with corresponding data like in the following table.



| Bit # | Description               |
|-------|---------------------------|
| 0     | 0 = channel #1 turned OFF |
|       | 1 = channel #1 turned ON  |
| 1     | 0 = channel #2 turned OFF |
|       | 1 = channel #2 turned ON  |
| 2     | 0 = channel #3 turned OFF |
|       | 1 = channel #3 turned ON  |
| 3     | 0 = channel #4 turned OFF |
|       | 1 = channel #4 turned ON  |
| 4     | 0 = channel #5 turned OFF |
|       | 1 = channel #5 turned ON  |
| 5     | 0 = channel #6 turned OFF |
|       | 1 = channel #6 turned ON  |
| 6     | 0 = channel #7 turned OFF |
|       | 1 = channel #7 turned ON  |
| 7     | 0 = channel #8 turned OFF |
|       | 1 = channel #8 turned ON  |

**Example:** 0x01 0x01 0x01 **0x81** 0x91 0xE8  
 (0x81 = 1000 0001. Channels №0 and №7 turned ON)

## 12.2. Turn single channel ON/OFF (0x05 – Write Single Coil)

### 12.2.1. Request

| Section         | Header | Data                     |            |                                       | Footer |
|-----------------|--------|--------------------------|------------|---------------------------------------|--------|
| Field           |        | Start Channel # to write |            | Channel status to be set              |        |
| Possible values |        | High byte                | Low byte   |                                       |        |
| Size            |        | 0x00                     | 0x00..0x07 | 0x0000 – turn OFF<br>0xFF00 – turn ON |        |
| 1 byte          | 1 byte | 2 bytes                  |            |                                       |        |

**Example:** 0x01 0x05 **0x00 0x00 0xFF 0x00** 0x8C 0x3A (Turn ON channel #1)

### 12.2.2. Response

If request was successfully processed – then response will be echo of request.

**Example:** 0x01 0x05 **0x00 0x00 0xFF 0x00** 0x8C 0x3A (Channel #1 turned ON)

## 12.3. Turn multiple channels ON/OFF (0x0F – Write Multiple Coils)

### 12.3.1. Request

| Section         | Header | Data                     |            |                           |            |                      |                  |          | Footer |
|-----------------|--------|--------------------------|------------|---------------------------|------------|----------------------|------------------|----------|--------|
| Field           |        | Start Channel # to write |            | Channels numbers to write |            | Number of data bytes | Channels' values | Reserved |        |
| Possible values |        | High byte                | Low byte   | High byte                 | Low byte   |                      |                  |          |        |
| Size            |        | 0x00                     | 0x00..0x07 | 0x00                      | 0x01..0x08 | 0x01                 | 0x00..0xFF       | 0x00     |        |
| 1 byte          | 1 byte | 1 byte                   | 1 byte     | 1 byte                    | 1 byte     | 1 byte               | 1 byte           |          |        |

Note: “Channels number to write” value should be calculated as 8 minus “Start Channel # to read”

Please refer to “Read channels state (0x01 – Read Coils)’ function’s response description in order to create correct Channels’ value byte (bits from 0 to 7).

**Example:** 0x01 0x0F 0x00 0x00 0x00 0x08 0x01 0xAA 0x00 0x6A 0x20  
(0xAA = 1010 1010. Turn on channels #2, 4, 6, 8 and turn off channels #1, 3, 5, 7)

### 12.3.2. Response

| Section         | Header | Data                    |            |                            | Footer |
|-----------------|--------|-------------------------|------------|----------------------------|--------|
| Field           |        | Start Channel # written |            | Number of channels written |        |
| Possible values |        | High byte               | Low byte   |                            |        |
| Size            |        | 0x00                    | 0x00..0x07 | 0x0000 .. 0x0008           |        |
|                 | 1 byte | 1 byte                  | 2 bytes    |                            |        |

**Example:** 0x01 0x0F 0x00 0x00 0x00 0x08 0x54 0x0D (8 channels’ values were written)

## 12.4. Reading holding registers (0x03 – Read Holding Registers)

| Registers map (one register is 16 bits) |   |
|---|---|
| Address                                 | Description   |
| 0x0000                                  | Module configuration, operational modes                   |
| 0x0001                                  | Pairwise mode timeout, in seconds                         |
| 0x0002                                  | Up time timer, high part                                  |
| 0x0003                                  | Up time timer, low part                                   |
| 0x0004                                  | Pairwise mode autoswitch timeout, value * 25 milliseconds |
| 0x0005                                  | Firmware version, major and minor                         |
| 0x0006                                  | Firmware version, service-pack and build                  |
| 0x0007...0x0010                         | Serial number (10 * 2 bytes)                              |

### 12.4.1. Request

| Section         | Header | Data                           |            |                             | Footer |
|-----------------|--------|--------------------------------|------------|-----------------------------|--------|
| Field           |        | Start register address to read |            | Number of registers to read |        |
| Possible values |        | High byte                      | Low byte   |                             |        |
| Size            |        | 0x00                           | 0x00..0x10 | 0x0001..0x0011              |        |
|                 | 1 byte | 1 byte                         | 2 bytes    |                             |        |

Note: “Number of registers to read” value should be calculated as 4 minus “Start register address to read”

**Example:** 0x01 0x03 0x00 0x00 0x00 0x04 0x44 0x09 (Read 4 registers starting from first one)

### 12.4.2. Response

| Section         | Header | Data                        |                    | Footer |
|-----------------|--------|-----------------------------|--------------------|--------|
| Field           |        | Number of bytes in response | Registers’ values  |        |
| Possible values |        | X = 0x01..2*registers count | 0x0000..0xFFFF * X |        |
| Size            |        | 1 byte                      | X * 2 bytes        |        |

**Example:** 0x01 0x03 0x08 0x00 0x09 0x00 0x3C 0x00 0x00 0x0A 0x51 0x9B 0x8E

- device configuration: **0x0009**
- Pairwise mode timeout: **0x003C** (60 seconds)
- Up-timer: **0x0000A51** (2641 seconds)

## 12.5. Writing single holding register (0x06 – Write Single Registers)

### 12.5.1. Request

| Section         | Header | Data                            |              |                                   | Footer |
|-----------------|--------|---------------------------------|--------------|-----------------------------------|--------|
| Field           |        | Start register address to write |              | Possible values                   |        |
| Possible values |        | High byte                       | Low byte     |                                   |        |
| Size            |        | 0x00                            | 0x01 or 0x04 | 0x0000..0x8CA0 or 0x000A.. 0x0050 |        |
|                 |        | 1 byte                          | 1 byte       | 2 bytes                           |        |

**Example:** 0x01 0x06 **0x00 0x01 0x0E 0x10** 0xDD 0xA6

### 12.5.2. Response

If request was successfully processed – then response will be echo of request.

**Example:** 0x01 0x06 0x00 0x01 **0x0E 0x10** 0xDD 0xA6  
 (pairwise mode timeout is set to 0x0E10 = 3600 seconds)

## 12.6. Report Slave ID (0x11 – Module Serial Number)

This function behaves the same way as 0x64 (Read module serial number).

## 12.7. Read module serial number (0x64 – Module Serial Number)

### 12.7.1. Request

| Section         | Header | Data                   | Footer |
|-----------------|--------|------------------------|--------|
| Field           |        | Any 4 bytes            |        |
| Possible values |        | 0x00000000..0xFFFFFFFF |        |
| Size            |        | 4 bytes                |        |

**Example:** 0x01 0x64 **0x12 0x34 0x56 0x78** 0x0B 0x36

### 12.7.2. Response

| Section         | Header | Data             | Footer |
|-----------------|--------|------------------|--------|
| Field           |        | Unique module ID |        |
| Possible values |        | 0x00..0xFF * 20  |        |
| Size            |        | 4 bytes          |        |

**Example:** 0x01 0x64 **0x10 0xE8 0x0B 0x00** 0xF3 0xC6  
 (There is a module with address 0x01 and it has the following unique ID: 0x10E80B00)

## 12.8. Possible error codes

Error response syntax: device address – function code + 0x80 – error code.

| Error code | Description   |
|------------|---|
| 0x01       | Function is not supported   |
| 0x02       | Incorrect address   |
| 0x03       | Incorrect data  |
| 0x04       | It is impossible to process request (collisions, incorrect parameter value) |

## 13. Collisions Between Manual Control And Master

Pairwise mode – if user is holding one of channels from particular pair and master will send request to set both second channel from the same pair to ON, then module will return error code 0x04. Outputs' state will not be changed.