

IOMOD 8DI4RO

IOMod 8DI4RO – industrial 8 digital inputs and 4 relay outputs module.

- Firmware version 1
 - IOMOD 8DI4RO User Manual Modbus
 - IOMOD 8DI4RO User Manual IEC 60870-5-103
 - IOMOD 8DI4RO User Manual IEC 60870-5-101
- Firmware version 2
 - IOMOD 8DI4RO User Manual

Firmware version 1

IOMOD 8DI4RO User Manual

Modbus

Introduction

IOMod 8DI4RO is a small-size stand-alone Modbus RTU, IEC 60870-5-103 or IEC-60870-5-101 digital input and relay output controller (protocol depends on firmware). IOMod can be used for industrial applications, where digital signaling is used and robust communication is needed. IOMod is an ideal solution for applications such as data acquisition, control, and process monitoring at remote places. This user manual is written for Modbus protocol firmware version.

Features

- 8 digital inputs;
- Configurable active input signal polarity or input inversion;
- 4 relay outputs;
- Galvanically isolated inputs and outputs;
- Pulsed or latched mode for individual outputs;
- Possible output feedback measurement with inputs;
- Configuration over USB console or MODBUS RTU;
- Values with data and time information;
- Drag and Drop firmware upgrade over USB mass storage;
- Modbus RTU, IEC-60870-5-103, IEC-60870-5-101 communication over RS485;
- Software selectable R_T 120 Ohm termination resistor on RS485 interface;
- LED indication for input/output and data transmission;
- Easy integration with WCC Lite gateway and CloudIndustries.eu platform;

Operational information

IOMod 8DI4RO uses Modbus RTU, IEC 60870-5-103, or IEC 60870-5-101 protocol to communicate with a master device over RS485 interface. Protocol used by the device can be changed by uploading the corresponding firmware.

Status LED

Status LED can be in 2 colors :

Blue - Device connected to USB.


Green - Normal operation.

Rx/Tx LED

The RX/TX LED on the IOMod flashes when data is either being transmitted or received via the RS485 port.

MODBUS operational information

To read output status, send 01 Modbus command (Read Coils) with address of the first register (0) and displacement of 4. Returned value will show all 4 output states (1 - turned On, 0 - turned Off).


Protocol: Modbus Serial RTU **Mode:** Master **STOP** **Port:** COM11 **Baudrate:** 9600 **Parity:** None **Data bits:** 8 **Stop bits:** One

| Slave | Function | Address | Value | FormattedValue | Name | Count |
|-------|----------|---------|-------|----------------|------|-------|
| 1 | 1 | 0 | False | | | 198 |
| 1 | 1 | 1 | True | | | 198 |
| 1 | 1 | 2 | False | | | 198 |
| 1 | 1 | 3 | True | | | 198 |

Job

Name:


Station: Function:

Address: Length:

The fourth byte in 8DI4RO respond shows the status of outputs (0A = 0000 1010, means that RO2 and RO4 outputs are turned on).

| Source | Message | Function | Start | Number | Value | RawData |
|--------|---------|----------|-------|--------|-------|-------------------------|
| VINCI | | 1 | 0 | 4 | | 01 01 00 00 00 04 3D C9 |
| RX | | 1 | | | 0A | 01 01 01 0A D1 8F |

To read input status, send 02 Modbus command (Read Discrete Inputs) with address of first register (0), and displacement of 8. Returned value will show 8 input states.


Protocol: Modbus Serial RTU **Mode:** Master **STOP** **Port:** COM11 **Baudrate:** 9600 **Parity:** None **Data bits:** 8 **Stop bits:** One

| Slave | Function | Address | Value | FormattedValue | Name | Count |
|-------|----------|---------|-------|----------------|------|-------|
| 1 | 2 | 0 | False | | | 30 |
| 1 | 2 | 1 | False | | | 30 |
| 1 | 2 | 2 | False | | | 30 |
| 1 | 2 | 3 | False | | | 30 |
| 1 | 2 | 4 | True | | | 30 |
| 1 | 2 | 5 | False | | | 30 |
| 1 | 2 | 6 | False | | | 30 |
| 1 | 2 | 7 | False | | | 30 |

Job

Name:

Station: Function:


Address: Length:

The fourth byte in 8DI4RO respond shows the status of outputs (10 = 0001 0000, means that fifth input is turned on).

| Source | Message | Function | Start | Number | Value | RawData |
|--------|---------|----------|-------|--------|-------|-------------------------|
| VINCI | | 2 | 0 | 8 | | 01 02 00 00 00 08 79 CC |
| RX | | 2 | | | 10 | 01 02 01 10 A0 44 |

To read input counter values, send 03 Modbus command (Read Holding Registers) or 04 Modbus command (Read Input Registers) with resolution of first register (0) and size of 24. Returned data will show pulse count (first register) and ON time (2nd and 3rd registers) for each input – pulse count of input #2 will be at register 4th, and so on. ON time will be shown as seconds.

Function 3:



Protocol: Modbus Serial RTU
Mode: Master

STOP

Port: COM11 **Baudrate:** 9600 **Parity:** None **Data bits:** 8 **Stop bits:** One

Settings Console **Statistic**

| Slave | Function | Address | Value | FormattedValue | Name | Count |
|-------|----------|---------|-------|----------------|------|-------|
| 1 | 3 | 0 | 2 | | | 50 |
| 1 | 3 | 1 | 10 | | | |
| 1 | 3 | 2 | 0 | | | |
| 1 | 3 | 3 | 1 | | | |
| 1 | 3 | 4 | 6 | | | |
| 1 | 3 | 5 | 0 | | | |
| 1 | 3 | 6 | 1 | | | |
| 1 | 3 | 7 | 6 | | | |
| 1 | 3 | 8 | 0 | | | |
| 1 | 3 | 9 | 11 | | | |
| 1 | 3 | 10 | 33 | | | |
| 1 | 3 | 11 | 0 | | | |
| 1 | 3 | 12 | 7 | | | |
| 1 | 3 | 13 | 741 | | | |
| 1 | 3 | 14 | 0 | | | |
| 1 | 3 | 15 | 1 | | | |
| 1 | 3 | 16 | 7 | | | |
| 1 | 3 | 17 | 0 | | | |
| 1 | 3 | 18 | 1 | | | |
| 1 | 3 | 19 | 7 | | | |
| 1 | 3 | 20 | 0 | | | |
| 1 | 3 | 21 | 3 | | | |
| 1 | 3 | 22 | 17 | | | |
| 1 | 3 | 23 | 0 | | | |


Job

Name: Read Holding Registers

Station: 1 Function: 3
Address: 0 Length: 24

Save Cancel

Function 4:



Protocol: Modbus Serial RTU
Mode: Master

STOP

Port: COM11 **Baudrate:** 9600 **Parity:** None **Data bits:** 8 **Stop bits:** One

Settings Console **Statistic**

| Slave | Function | Address | Value | FormattedValue | Name | Count |
|-------|----------|---------|-------|----------------|------|-------|
| 1 | 4 | 0 | 2 | | | 11 |
| 1 | 4 | 1 | 10 | | | 11 |
| 1 | 4 | 2 | 0 | | | 11 |
| 1 | 4 | 3 | 1 | | | |
| 1 | 4 | 4 | 6 | | | |
| 1 | 4 | 5 | 0 | | | |
| 1 | 4 | 6 | 1 | | | |
| 1 | 4 | 7 | 6 | | | |
| 1 | 4 | 8 | 0 | | | |
| 1 | 4 | 9 | 11 | | | |
| 1 | 4 | 10 | 33 | | | |
| 1 | 4 | 11 | 0 | | | |
| 1 | 4 | 12 | 7 | | | |
| 1 | 4 | 13 | 741 | | | |
| 1 | 4 | 14 | 0 | | | |
| 1 | 4 | 15 | 1 | | | |
| 1 | 4 | 16 | 7 | | | |
| 1 | 4 | 17 | 0 | | | |
| 1 | 4 | 18 | 1 | | | |
| 1 | 4 | 19 | 7 | | | |
| 1 | 4 | 20 | 0 | | | |
| 1 | 4 | 21 | 3 | | | |
| 1 | 4 | 22 | 17 | | | |
| 1 | 4 | 23 | 0 | | | |

Job

Name: Read Input Registers

Station: 1 Function: 4
Address: 0 Length: 24

Save Cancel

Values can be compared with values in Diagnostics Screen in Putty application (this interface will be discussed later):

```

COM4 - PuTTY

----- Diagnostics Screen V1.21.0 -----

Output #1 state: 0
Output #2 state: 1
Output #3 state: 0
Output #4 state: 1

Input #1 state: 0; Pulse count: 2; On time: 10
Input #2 state: 0; Pulse count: 1; On time: 6
Input #3 state: 0; Pulse count: 1; On time: 6
Input #4 state: 0; Pulse count: 11; On time: 33
Input #5 state: 0; Pulse count: 7; On time: 741
Input #6 state: 0; Pulse count: 1; On time: 7
Input #7 state: 0; Pulse count: 1; On time: 7
Input #8 state: 0; Pulse count: 3; On time: 17

```

ON time and pulse count will increase when input pulse is longer than Filter time, which is configured by user in USB terminal menu. Shorter pulses will be ignored in both pulse and ON time registers. From software version 1.10, as capacity of input counter expanded to 32-bits, additional 16 registers depict such wider values in registers 00023-00039. That means that function 3 and 4 commands can be sent with length 40 specified. Then 24-39 registers are going to display only pulse count values:

| | | |
|---|----|---|
| 4 | 24 | 5 |
| 4 | 25 | 1 |
| 4 | 26 | 3 |
| 4 | 27 | 0 |
| 4 | 28 | 3 |
| 4 | 29 | 0 |
| 4 | 30 | 2 |
| 4 | 31 | 0 |
| 4 | 32 | 2 |
| 4 | 33 | 0 |
| 4 | 34 | 2 |
| 4 | 35 | 0 |
| 4 | 36 | 2 |
| 4 | 37 | 0 |
| 4 | 38 | 5 |
| 4 | 39 | 0 |

Job

Name:

Station: Function:

Address: Length:

Two registers are allocated for each pulse count value. If value of pulse count becomes more than 65535 then second register comes into use. The values above can be compared with values from diagnostics screen:

```

COM4 - PuTTY

----- Diagnostics Screen V1.21.0 -----

Output #1 state: 0
Output #2 state: 0
Output #3 state: 0
Output #4 state: 0

Input #1 state: 0; Pulse count: 65541; On time: 10
Input #2 state: 0; Pulse count: 3; On time: 0
Input #3 state: 0; Pulse count: 3; On time: 0
Input #4 state: 0; Pulse count: 2; On time: 0
Input #5 state: 0; Pulse count: 2; On time: 0
Input #6 state: 0; Pulse count: 2; On time: 0
Input #7 state: 0; Pulse count: 2; On time: 0
Input #8 state: 0; Pulse count: 5; On time: 0

```

These input counter values can be changed by using 06 Modbus command.

| | | | |
|---|---|---|-------|
| 1 | 4 | 5 | 10101 |
| 1 | 4 | 6 | 3 |
| 1 | 4 | 7 | 0 |
| 1 | 4 | 8 | 0 |
| 1 | 4 | 9 | 2 |

Function 6

Address: Value:

| Source | Message | Function | Start | Number | Value | RawData |
|--------|---------|----------|-------|--------|-------|-------------------------|
| VINCI | | 6 | 5 | | 27 75 | 01 06 00 05 27 75 43 DC |
| RX | | 6 | 5 | | 27 75 | 01 06 00 05 27 75 43 DC |

To turn single output on or off, send command 05 (Write Single Coil), with output address (0 to 3). To turn output on – send hex value FF00; to turn off – hex value 0000.

| Slave | Function | Address | Value |
|-------|----------|---------|-------|
| 1 | 1 | 0 | False |
| 1 | 1 | 1 | False |
| 1 | 1 | 2 | True |
| 1 | 1 | 3 | False |

Function 5

Address: ☒ Value

| Source | Message | Function | Start | Number | Value | RawData |
|--------|---------|----------|-------|--------|-------|-------------------------|
| VINCI | | 5 | 2 | | FF 00 | 01 05 00 02 FF 00 2D FA |
| RX | | 5 | 2 | | FF 00 | 01 05 00 02 FF 00 2D FA |

To turn multiple outputs on or off, use command 15 (Write Multiple Coils), and send binary coded value for 4 coils at address (0) and length 4.

To invert input states by software configure device over USB terminal. Useable Modbus commands shown in table below.

Supported MODBUS functions

01 (0x01) Read Coil Status

Reads status of relays (Off or On). IOMOD 8DI4RO has 4 relay outputs from address 0 to address 3.

02 (0x02) Read Discrete Inputs

Reads status of digital inputs (Off or On). IOMOD 8DI4RO has 8 digital inputs from address 0 to address 7. These inputs are active-high by default.

03 (0x03) Read Holding Registers

Lets user read counter/timer values dedicated to digital inputs. There are 40 MODBUS registers. Values held in these registers are explained in a table below. There are two types of values - Pulse Counter and On Timer, the latter calculating the time that respective input was held in its active state.

04 (0x04) Read Input Registers

Lets user read counter/timer values dedicated to digital inputs. There are 80 MODBUS registers. Values held in these registers are explained in a table below. There are two types of values - Pulse Counter and On Timer, the latter calculating the time in seconds that respective input was held in its active state. This function is deprecated and mirrors function 0x03 to conform to past versions of IOMOD 16DI.

05 (0x05) Write Single Coil

Sets a single relay output On or Off. Output addresses from 0 to 3 (first output – address 0, last output – address 3).

06 (0x06) Preset Single Register

Sets a single register. Register addresses is identical to “*Read Input Registers*” addresses.

15 (0x0F) Write Multiple Coils

Sets multiple relay output On or Off. Output addresses from 0 to 3 (first output – address 0, last output – address 3).

Modbus register mapping table

| Register | Description | Value range |
|----------------------------------|---------------------------------|-------------|
| Read coil status (01) | | |
| 00000-00007 | Reading digital outputs RO1-RO4 | 0-255 |
| Read discrete inputs (02) | | |
| 00000-00007 | Reading digital inputs DI1-DI8 | 0-255 |

| | Read holding register (03), Read input register (04), Preset Single Register (06) | |
|-------------|--|-----------------|
| 00000 | Pulse count for DI1, Least Significant Word | 0-65535 |
| 00001-00002 | On-time in seconds for DI1, Least Significant Word first* | 0-65535 |
| ... | ... | ... |
| 00021 | Pulse count for DI8, Least Significant Word | 0-65535 |
| 00022-00023 | On-time in seconds for DI8, Least Significant Word first* | 0-65535 |
| 00024-00039 | Pulse count for DI1-DI8, Least Significant Word first* | 0-65535 |
| | Write single coil (05) | |
| 00000-00003 | Writing a single relay output | 0x0000 / 0xFF00 |
| | Write multiple coils (15) | |
| 00000-00003 | Writing multiple relay outputs RO1-RO4 | 0-255 |

*It is advised to set most significant word of counter/timer first

Device configuration

Input inversion and polarity selection

Unlike 8DI8DO IOMod 8DI4RO does not have pull-up resistors. If user desires to turn input status on, when that input signal is low, user then inverts inputs logically. All input indication LED's stay the same (are not inverted).

Input / Output grouping

Sometimes two inputs or two outputs must be captured as one DPI input or output. Inputs and outputs can be grouped into the pairs of two. This allows outputs to be controlled by one DPI command (of address of first output in the group). Only two neighbor pins can be grouped into pair, while first pin in pair must be an odd number pin. When grouped, second pin in the pair is not used anymore – all requests for this pin generate an error. For example – RO1.1-2 and RO2.1-2 can be grouped, after that RO2.1-2 is not used; RO2.1-2 and RO3.1-2 cannot be grouped; RO3.1-2 and RO4.1-2 can be grouped, but RO4.1-2 is not used then, etc.

Fig. 1. picture shows outputs and inputs ungrouped and controlled independently. In this mode, General Interrogation will be composed of 4 output states and 8 input states.

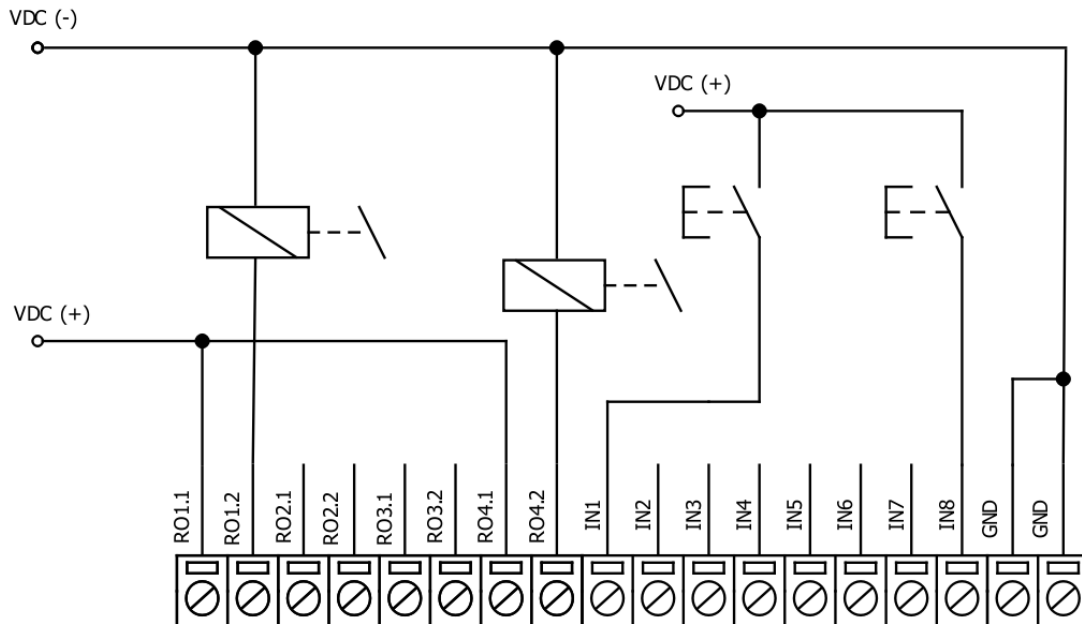


Fig. 1.

Let's consider a situation shown in Fig. 2. In the picture below first two relay outputs are grouped into pair. Also 3rd and 4th inputs are grouped into pair. Now, General interrogation will be composed of 3 output states (with RO2.1-2 missing), and 7 input states (with IN4 missing). Output and input numbers is represented by "Info number" in protocol.

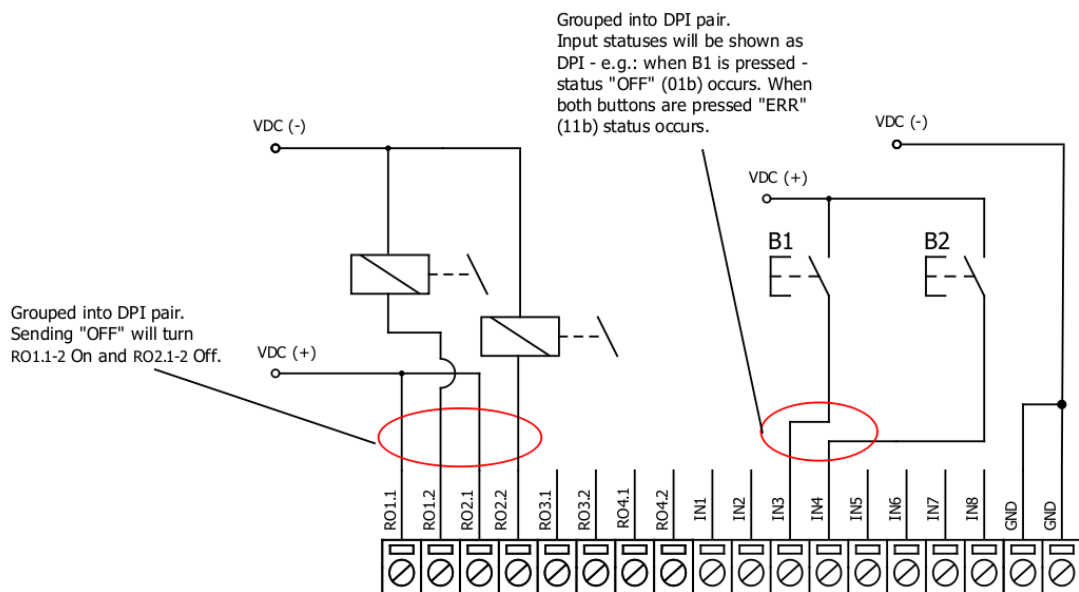


Fig. 2.

Input filter

Input filter is a simple glitch filter with time input. This filter time corresponds to stable time that input must achieve before sending a status change.

Output pulse time

User can configure outputs to be pulse controlled – it means that output will be turned on for configured amount of time. When this time runs out, output is turned off. This is useful when pulse toggle relays are used. Output pulse is independent from output grouping option and can be used on both grouped and ungrouped outputs. When output is grouped, device will allow only one command completion at a time – when output is already turned ON, other “turn ON” requests will be responded with NACK. If user desires latching outputs to be used, output pulse time is set to 0.

In the picture below is depicted an example of pulse output usage. Inputs and outputs are grouped, and output pulse time is set to 1s. When user sends ON command, RO2.1-2 are pulsed for 1s, and relay is set. This will connect NO contact and IN2 will turn on (assuming it is not inverted). When user sends OFF command, RO1.1-2 are pulsed, and relay is reset, turning IN1 on.

Output detection with inputs

Users can detect an output change with inputs.

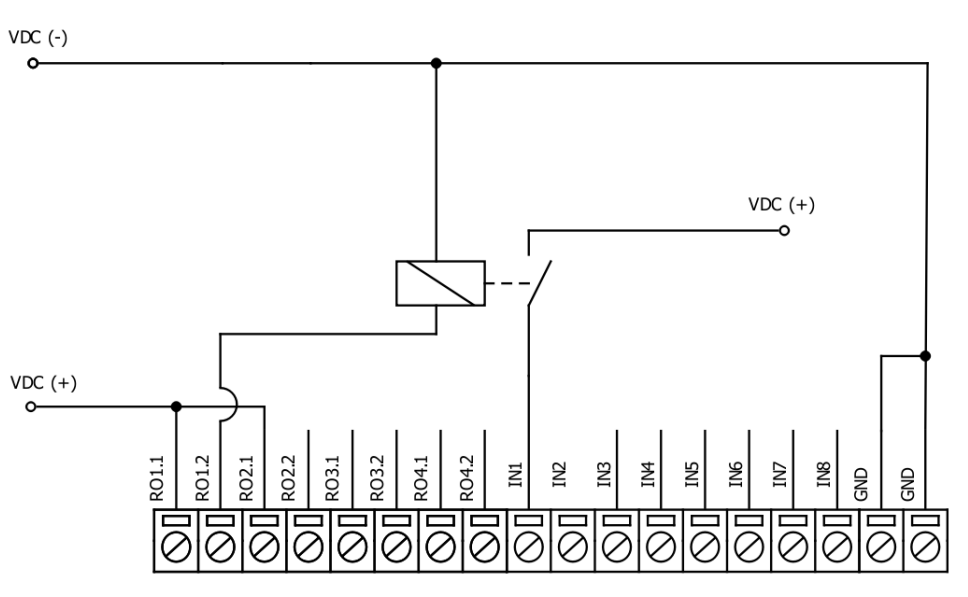


Fig. 3.

To find out if relays are turned on, user can connect relay outputs to IMod inputs (maximum allowed voltage must be taken into account). When relays are turned on, device responds with IEC-60870-5-101 protocol message “Remote Operation”. If inputs are never turned on or off, device will send “Remote Operation” message after time-out period, with current input statuses. Time-out period is configured by user as a Feedback Time.

Technical information

| | System | |
|-----|-----------------------------|--|
| 1. | Dimensions | 101 x 119 x 22.5 mm |
| 2. | Case | IP20, blend PC/ABS self-extinguishing, black |
| 3. | Working environment | Indoors |
| 4. | Operating temperature | from -30°C to +70°C |
| 5. | Humidity | 5-95% RH (non-condensing) |
| 6. | Configuration | USB – serial console, MODBUS RTU |
| 7. | Firmware upgrade | USB – mass storage device |
| | Electrical specifications | |
| | Inputs | |
| 8. | Nominal input voltage range | 12-48VDC |
| 9. | Isolation | 8 X 3kV _(rms) |
| 10. | Maximum input voltage | 60VDC |

| | Output relay contacts ratings | |
|-----|--------------------------------|-----------------------------------|
| 11. | Isolation | 4 X 3kV _(rms) |
| 12. | Rated resistive load | 5 A (at 250 VAC or at 30 VDC) |
| 13. | Rated inductive (L/R=7ms) load | 2 A (at 250 VAC or at 30 VDC) |
| 14. | Maximum switching voltage | 277 VAC 125 VDC |
| 15. | Maximum switching current | 5 A |
| | Power | |
| 16. | Power Supply | 12-24 VDC / 9-33 VDC (full range) |
| 17. | Current consumption | 70 mA |

In the Fig. 5. you can see internal structure of 8DI4RO IOmod. An important fact to metion is that as it can be noticed in the scheme below all inputs are isolated. However ground connection is common for all phototransistors.

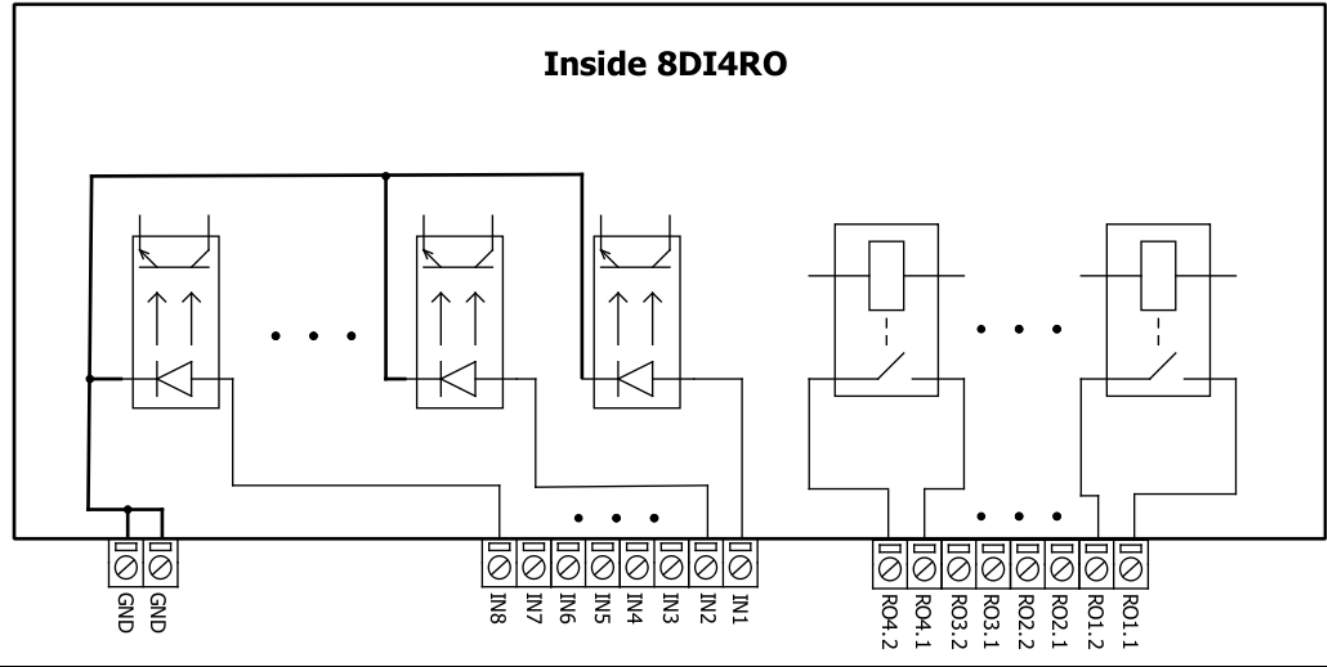


Fig. 4.

Mounting and installation guide

IOmod 8DI4RO RS485 interface

IOmod 8DI4RO has an integrated 120Ω termination resistor which can be enabled or disabled over USB configuration. It is recommended to use termination at each end of the RS485 cable. See typical connection diagram on Fig. 6.

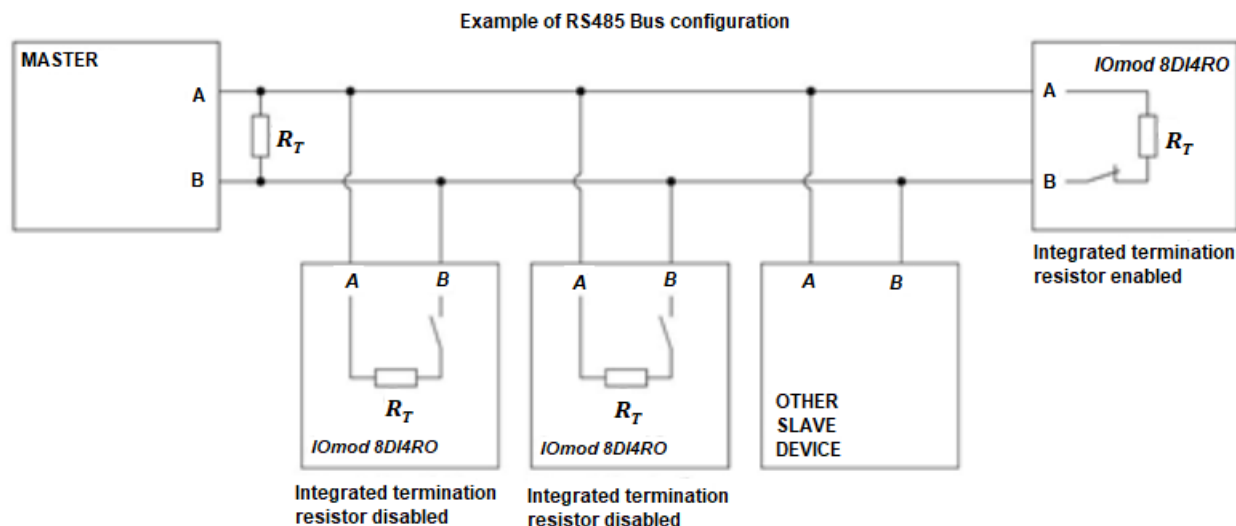


Fig. 5.

IOmod 8DI4RO has 1/8 Unit load receiver which allows to have up to 256 units on line (compared to standard 32 units). To reduce reflections, keep the stubs (cable distance from main RS485 bus line) as short as possible when connecting device.

IOmod 8DI4RO inputs

Typical application of IOmod 8DI4RO inputs is shown on Fig. 2. When default configuration for inputs is applied, user will see inputs connected to +12-24V as “high” or state “1” and input status LED will glow.

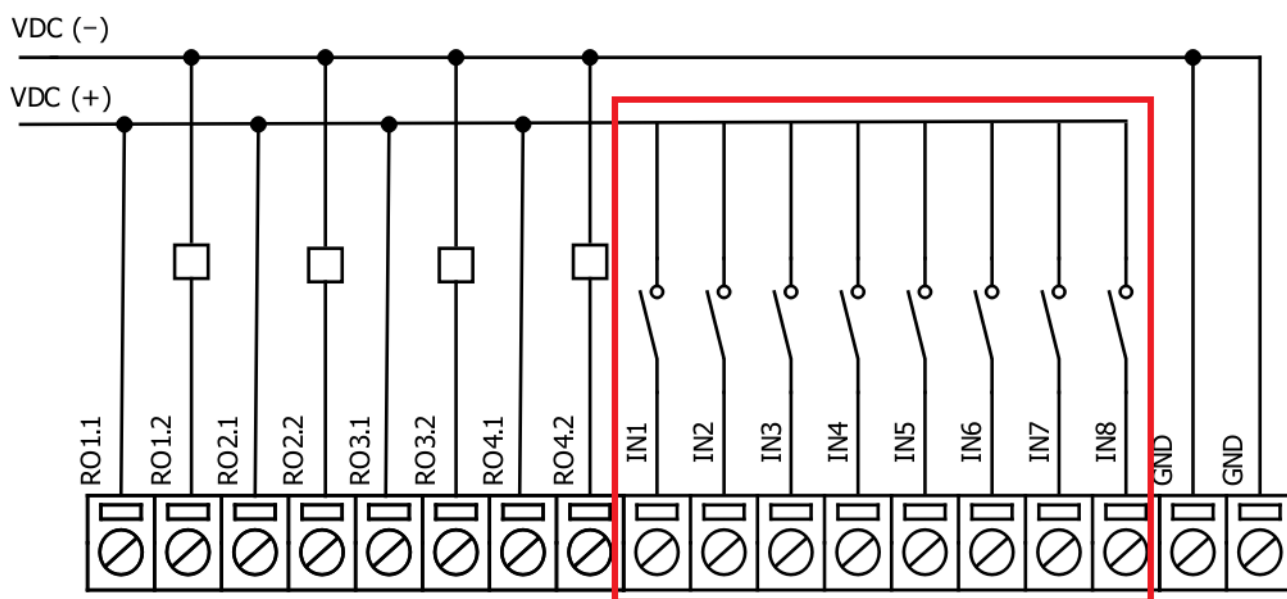


Fig. 6.

User also can configure to enable software input inversion. With this configuration, user will see inputs connected to 0V (see Fig. 3.) as “high” or state “1”, input status LED will NOT glow.

IOmod 8DI4RO outputs

IOmod 8DI4RO has 4 relay outputs. Internal clamp diodes are connected to each output which makes IOmod 8DI4RO ideal for driving inductive loads like relays. Maximum 5A per output is allowed. For higher loads outputs can be connected in parallel. Make sure your power supply can provide enough power. Typical application of outputs is shown on Fig. 4.

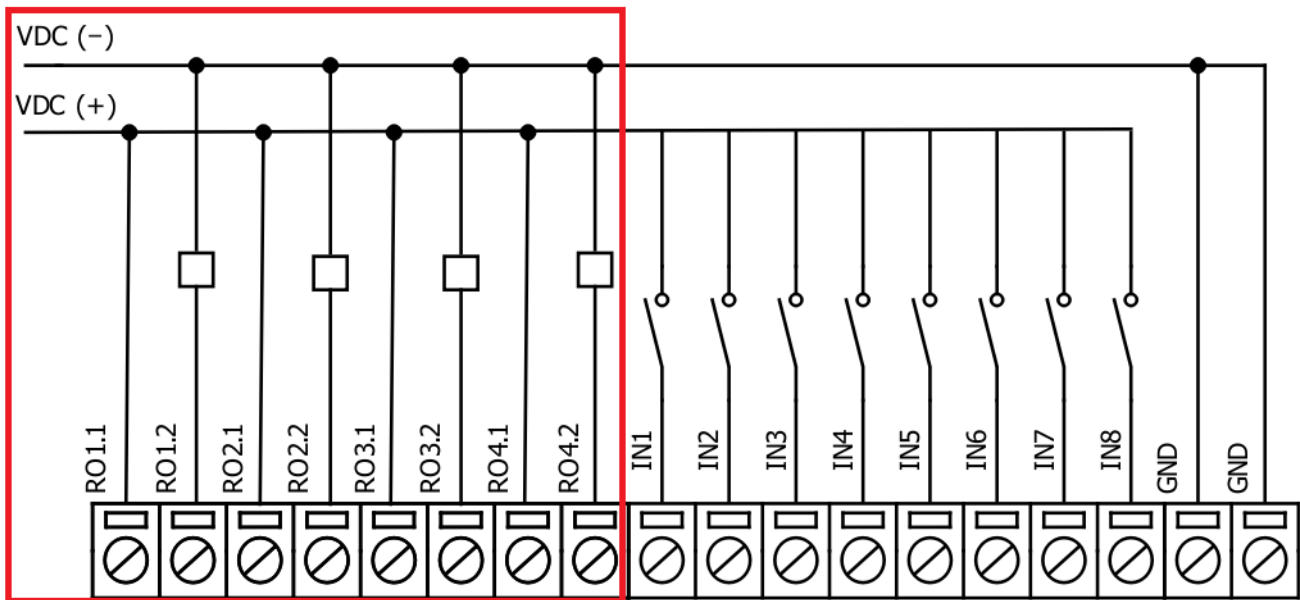


Fig. 7.

Configuration over USB

Driver installation

Device requires USB drivers to work as a Virtual COM port. First-time connection between device and computer could result in "Device driver software was not successfully installed" error (Fig. 5.).

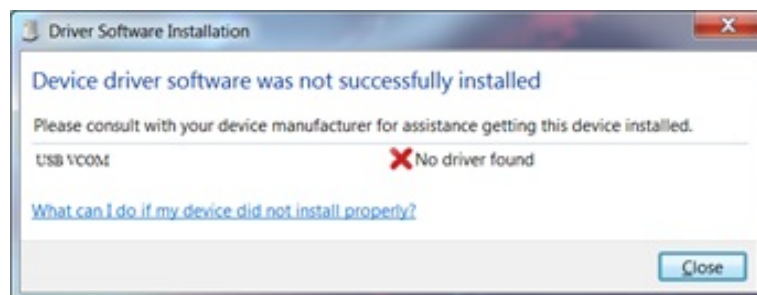


Fig. 8.

Drivers can be installed from Wiki Esleseta -> Downloads -> IOMOD Series Downloads-> Firmware and tools -> IOMOD USB drives for Windows;

User then manually installs drivers by selecting downloaded driver folder:
Go to Control Panel -> Hardware and Sound -> Device Manager
Select failed device;
Press "Update driver software"; screen in Fig. 6. should appear:

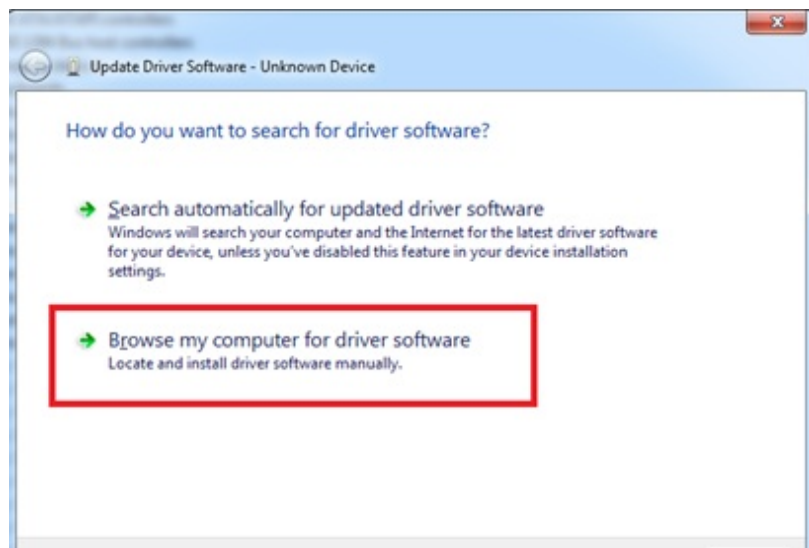


Fig. 9.

Select "x86" driver for 32-bit machine, or x64 for 64-bit machine. If not sure, select root ("vcom") folder (folder in which x64 and x86 lays inside).

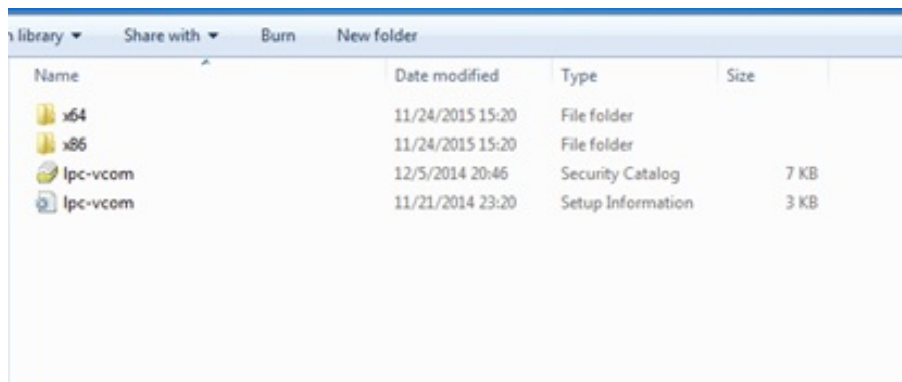


Fig. 10.

IOMod configuration with PuTTY terminal

Configuration of IOMod device is done through CLI (Command Line Interface) on a virtual COM port. Drivers needed for MS Windows to install VCOM will be provided. To open up CLI simply connect to specific V-COM port with terminal software (it is advised to use PuTTY terminal software. If other software is being used, user might need to send <return> symbol after each command). When connected user should immediately see main screen. Accidental close of the terminal window does not stop USB connection, user can connect terminal program again, and press any key on keyboard to show up main menu again.

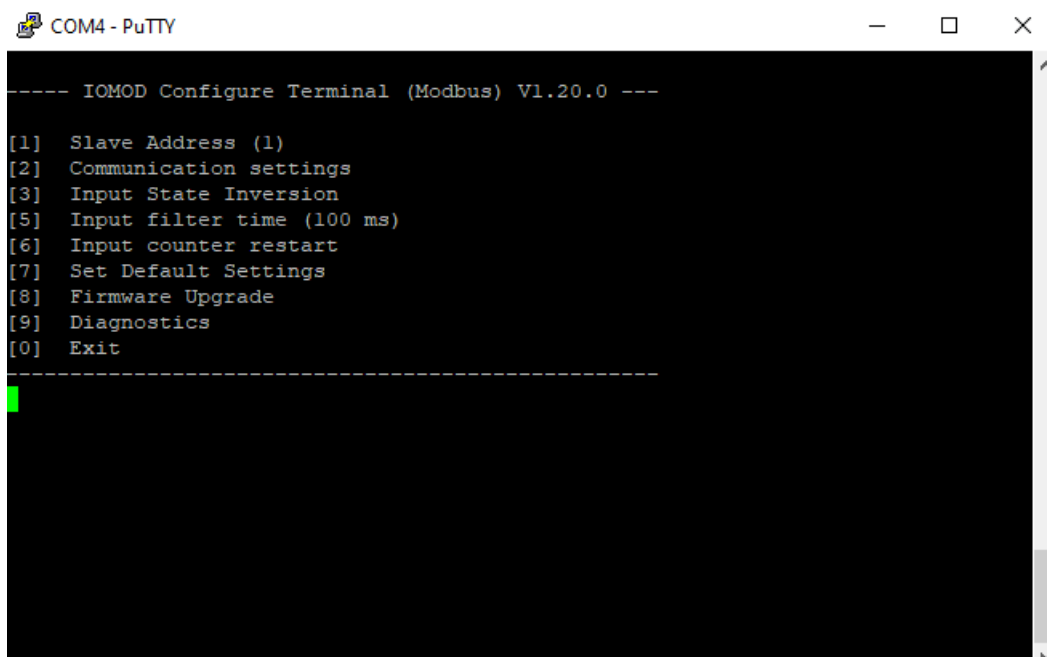


Fig. 11.

Navigation is performed by sending numbers to terminal. User then proceeds by following further on-screen instructions. For example, press [1] to set Slave Device address, press [2] to set Communication settings, press [3] to enable or disable Input State Inversion. When done, press [0] (exit) before disconnecting device physically. Default values is set by pressing [7] on main screen and confirming changes [1]. Communication settings window is shown in Fig. 12.

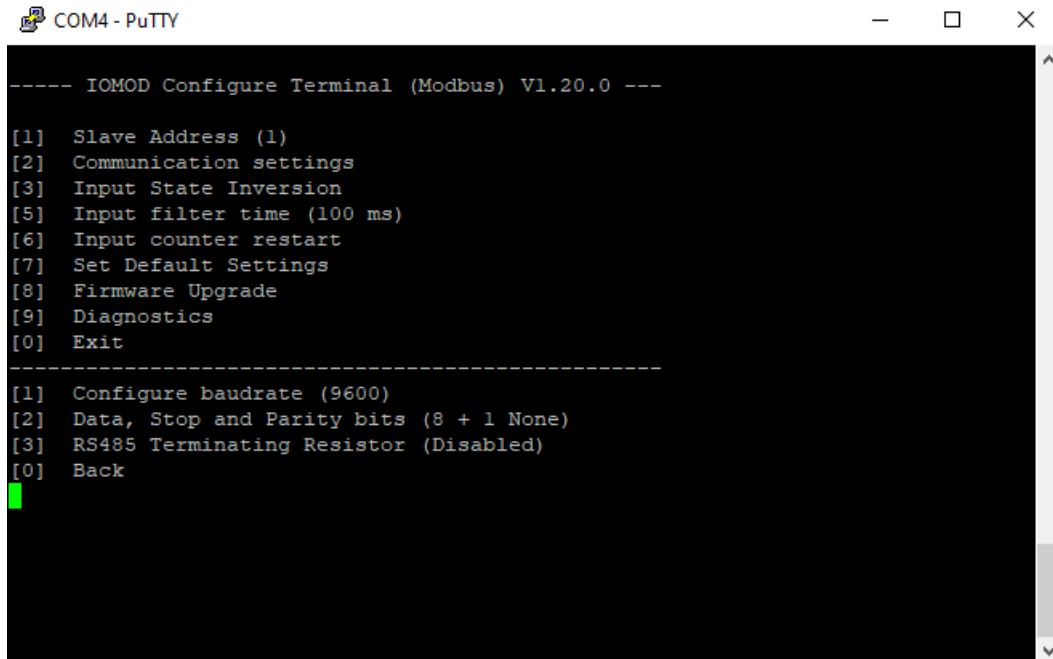


Fig. 12.

Main Menu

| | Menu Name | Function | Values | Default Values |
|-----|----------------------------|--|---|----------------|
| 1. | Slave Address | Modbus Slave address / ID | 1-254 | 1 |
| 2. | Communication settings | Enters configuring screen for communication settings. | - | - |
| 2.1 | Configure baudrate | Configure number of bits transmitted per second | 100 - 256 000 | 9600 |
| 2.2 | Data, Stop and Parity bits | Configure length and content of packets | [1] 8+1 [2] 8+2 Parity: [1] None [2] Odd [3] Even [4] Mark [5] Space | 8+1+N |
| 2.3 | RS485 Terminating Resistor | RS485 120 Ohms Terminating Resistor | [1] Enable [2] Disable | Disabled |
| 3. | Input State Inversion | Input state inversion for individual pins. Inverts states of input signals, so that high input voltage is considered to be logical 0 and vice versa. | [1] Inverted [2] Set Normal | Normal |
| 4. | Input filter time | Filter for short input pulses | 0 - 256 000 (ms) | 100 |
| 5. | Input counter restart | Restarts all input counter registers to 0 | [1] Confirm [0] Cancel | - |
| 6. | Set Default Settings | Sets Default Settings | [1] Confirm [0] Cancel | - |

| | | | | |
|----|------------------|---|---------------------------|---|
| 7. | Firmware Upgrade | Mass Storage Device Firmware Upgrade | [1] Confirm [0] Cancel | - |
| 8. | Diagnostics | Input / Output states [9] USB protocol simulator mode | - | - |
| 0. | Exit | Exit and disconnect | - | - |

Protocol simulator

When entered diagnostics screen, user can turn on USB protocol simulator mode by pressing [9]. When protocol simulator is turned on, device will communicate through USB port rather than RS-485 line. Communication on RS-485 line is closed and all IEC-101 commands will be accepted only from USB. To exit this mode user must restart device.

Firmware upgrade over USB

To update device firmware user must enter main configuration menu and enter Firmware upgrade screen by pressing [8] is shown in Fig. 13.

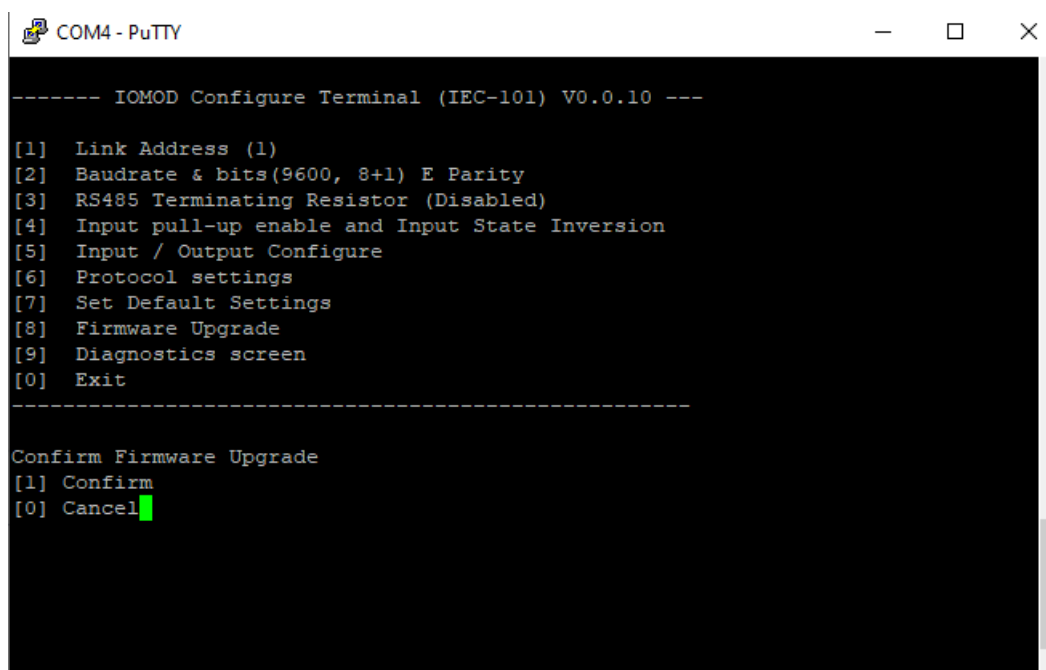


Fig. 17.

Confirm upgrade by pressing [1];

Device should enter a Firmware Upgrade mode. It means that device switches from USB Console mode into Mass storage device and computer recognize it as USB Storage.

Usually after confirming the upgrade a window with an error message appears (Fig. 14)

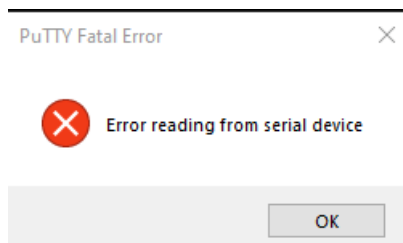


Fig. 18.

User then must **delete** existing file "firmware.bin", and simply drag and drop new firmware file.

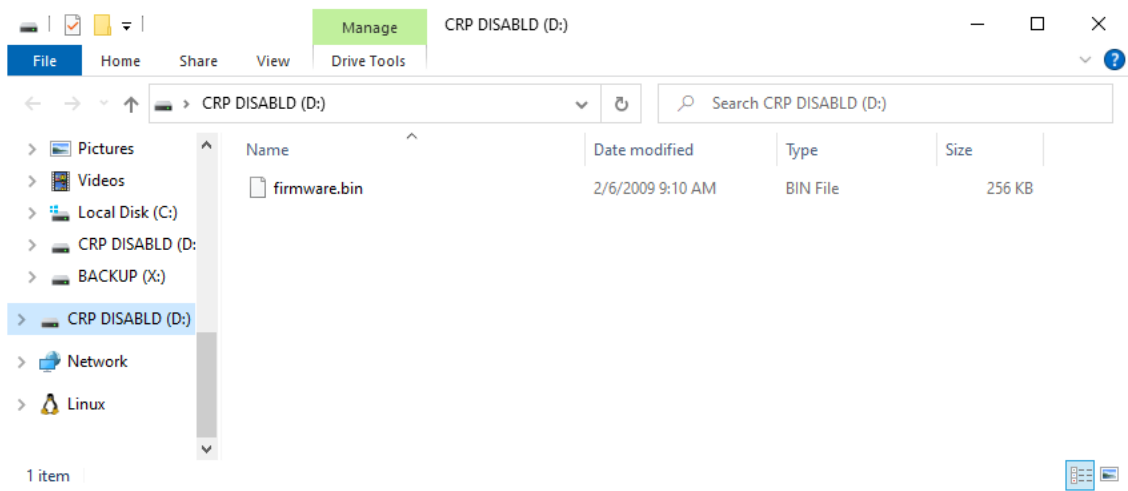


Fig. 19.

Finally, reconnect device and check firmware version. Lastly, you may want to set default settings by pressing [7].

IOMOD 8DI4RO User Manual IEC 60870-5-103

Introduction

IOMod 8DI4RO is a small-size stand-alone Modbus RTU, IEC 60870-5-103 or IEC-60870-5-101 digital input and relay output controller (protocol depends on firmware). IMod can be used for industrial applications, where digital signaling is used and robust communication is needed. IMod is an ideal solution for applications such as data acquisition, control, and process monitoring at remote places. This user manual is written for IEC 60870-5-103 protocol firmware version.

Features

- 8 digital inputs;
- Configurable active input signal polarity or input inversion;
- 4 relay outputs;
- Galvanically isolated inputs and outputs;
- Pulsed or latched mode for individual outputs;
- Possible output feedback measurement with inputs;
- Configuration over USB console or MODBUS RTU;
- Values with data and time information;
- Drag and Drop firmware upgrade over USB mass storage;
- Modbus RTU, IEC-60870-5-103, IEC-60870-5-101 communication over RS485;
- Software selectable R_T 120 Ohm termination resistor on RS485 interface;
- LED indication for input/output and data transmission;
- Easy integration with WCC Lite gateway and CloudIndustries.eu platform;

Operational information

IOMod 8DI4RO uses Modbus RTU, IEC 60870-5-103, or IEC 60870-5-101 protocol to communicate with the master device over the RS485 interface. Protocol used by the device can be changed by uploading the corresponding firmware. Default communication settings are:

| | |
|--------------|------|
| Baud rate | 9600 |
| Data bits | 8 |
| Parity | Even |
| Stop bits | 1 |
| Link address | 1 |

IOMod 8DI4RO configuration can be changed over a USB interface with a terminal console like PuTTY or similar.

Status LED

Status LED can be in 2 colors :

Blue - Device connected to USB.

Green - Normal operation.

Rx/Tx LED

The RX/TX LED on the IOMod flashes when data is either being transmitted or received via the RS485 port.

IEC 60870-5-103 working information

Initialization

IOMod uses a standard IEC-60870-5-103 communication scheme. Initiation, control messages and queries are initiated by the master (controlling station), while IOMod device (controlled station) only answers these requests.

Therefore, the first message should be sent by master to start/restart communication (function code 7) (RESET CU or LINK RESET FCB). This message is answered by IOMod with an acknowledgement (ACK) (0). After that master sends RequestStatusOfLink (9) message. The message is answered by slave with Status (11) message. Finally, master sends ResetOfRemoteLink (0) message which has to be answered with ACK (0) message from slave.

The initialization procedure enables master to proceed with sending other messages defined by IEC- 60870-5-103 protocol. Other messages are ignored until a successful initialization has taken place.

Data polling

When initialization is complete, master may poll IOMod device with both Class 1 and Class 2 requests. Class 2 is used when master polls for a cyclic data. Controlled device answers with a message containing Access Demand flag when spontaneous data exists and master then sends request for Class 1. IOMod would then respond with time-tagged message.

On first Class 1 request IOMod device always asks for the Access Demand to send an identification string. However, if there are spontaneous messages to be sent, they will be sent before the identification string.

Output control

To enable or disable relay outputs master (controlling station) sends commands conforming to the IEC-60870-5-103 protocols. The function type of the output commands has to be 128 (80h). Info number represents the number of output pin (1-4 accordingly). Info elements shows DPI information of output state (0 – intermediate, 1 – off, 2 – on and 3 – not used (defines error)). Successful command is accepted with a positive acknowledge.

Input messages

When input status changes, IOMod device filters input glitches through filters with a user configurable filter time. When the filter is passed device sends “Spontaneous” message with “Information number” fields displaying input pin addresses and default “Function type” of 160 (A0h). Please note that spontaneous messages are answered with a four-byte time structure not containing date info. Controlling station should therefore be able to handle the signals sent before the start of a new day.

Time synchronization

To initiate the time synchronization between devices master must send variable frame, with Function Type = 0 (GLB), Type Identification = 6, Information Number = 0 and Cause of Transmission = 8. Info elements must contain the 7 byte time structure.

As per IEC-60870-5-103 protocol specification time synchronization can be completed for multiple devices using broadcasting messages. It is included in IEC-60870-5-103 firmware since version 1.7.3. To broadcast time synchronization message, link address should be equal to 255.

General interrogation

General Interrogation (GI) is initiated by the master with variable frame, including Function Code = 3 (User data with ACK), Type Identification = 7, Information number = 0 and Cause of Transmission = 9. Slave device then responds with an acknowledgement (ACK). Master gets IOMod data by polling with Class 2 requests. IOMod device responds with a time-tagged message, including DPI states of inputs and outputs (Outputs are sent first). Output and input pin numbers are represented by “Info number” fields in received packets from IOMod. Finally, slave transmits “End of GI” (Cause of Transmission = 10, Type Identification = 8).

Device configuration

Input inversion and polarity selection

Unlike 8DI8DO IOMod 8DI4RO does not have pull-up resistors. If user desires to turn input status on, when that input signal is low, user then inverts inputs logically. All input indication LED's stay the same (are not inverted).

Input / Output grouping

Sometimes two inputs or two outputs must be captured as one DPI input or output. Inputs and outputs can be grouped into the pairs of two. This allows outputs to be controlled by one DPI command (of address of first output in the group). Only two neighbor pins can be grouped into pair, while first pin in pair must be an odd number pin. When grouped, second pin in the pair is not used anymore – all requests for this pin generate an error. For example – RO1.1-2 and RO2.1-2 can be grouped, after that RO2.1-2 is not used; RO2.1-2 and RO3.1-2 cannot be grouped; RO3.1-2 and RO4.1-2 can be grouped, but RO4.1-2 is not used then, etc.

Fig. 1. picture shows outputs and inputs ungrouped and controlled independently. In this mode, General Interrogation will be composed of 4 output states and 8 input states.

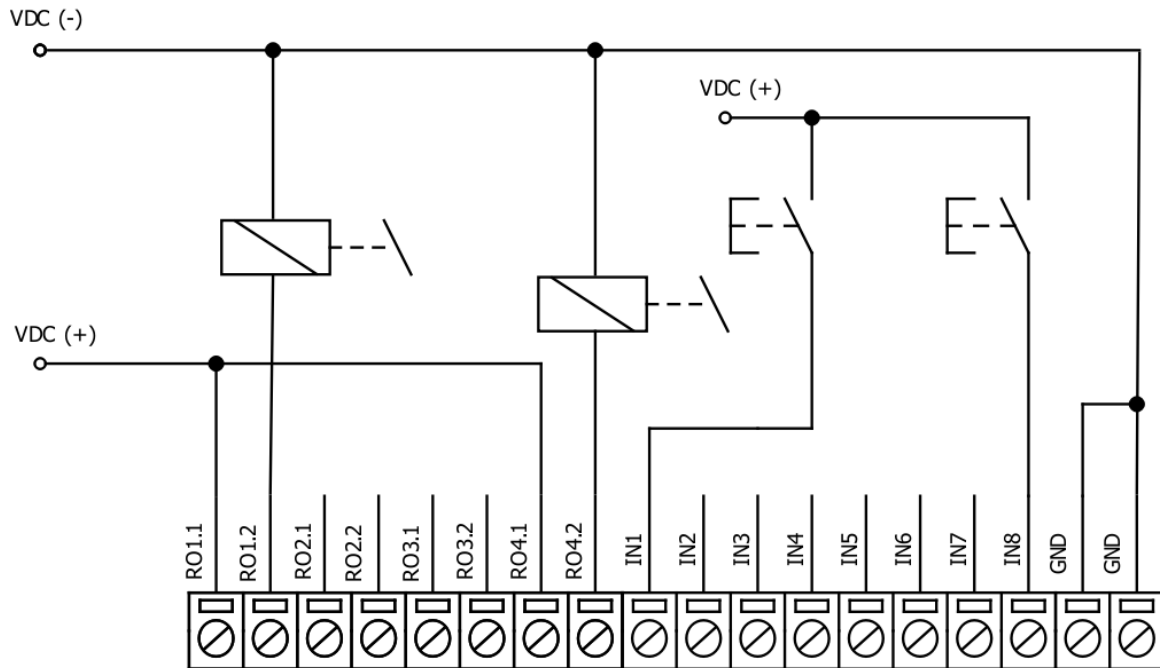


Fig. 1.

Let's consider a situation shown in Fig. 2. In the picture below first two relay outputs are grouped into pair. Also 3rd and 4th inputs are grouped into pair. Now, General interrogation will be composed of 3 output states (with RO2.1-2 missing), and 7 input states (with IN4 missing). Output and input numbers is represented by "Info number" in protocol.

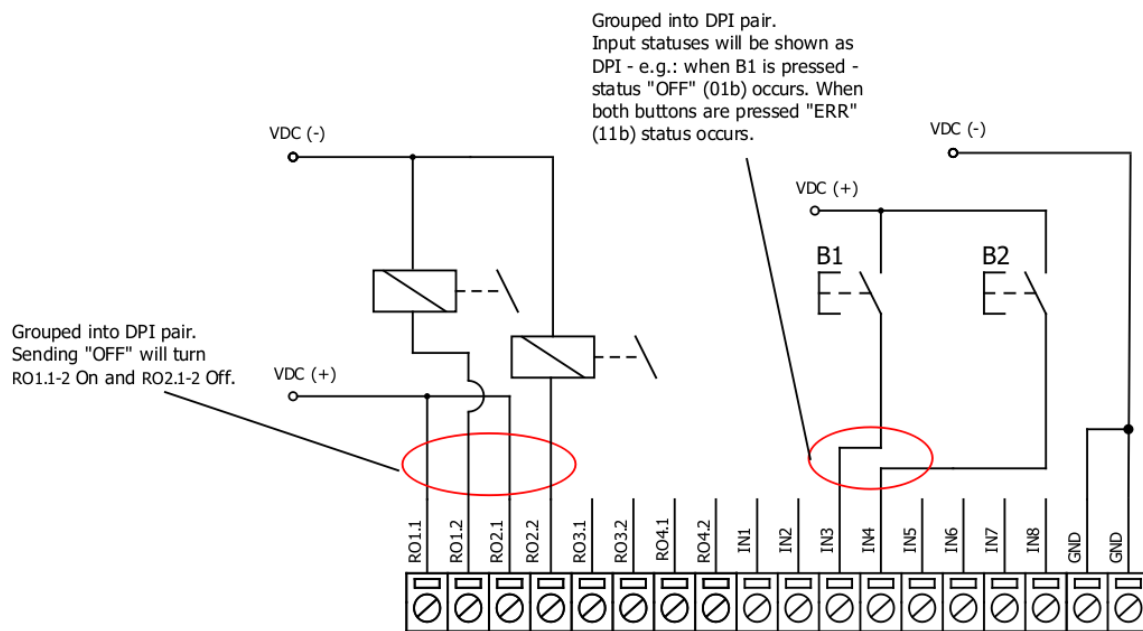


Fig. 2.

Input filter

Input filter is a simple glitch filter with time input. This filter time corresponds to stable time that input must achieve before sending a status change.

Output pulse time

User can configure outputs to be pulse controlled - it means that output will be turned on for configured amount of time. When this time runs out, output is turned off. This is useful when pulse toggle relays are used. Output pulse is independent from output grouping option and can be used on both grouped and ungrouped outputs. When output is grouped, device will allow only one command completion at a time - when output is already turned ON, other "turn ON" requests will be responded with NACK. If user desires latching outputs to be used, output pulse time is set to 0.

In the picture below is depicted an example of pulse output usage. Inputs and outputs are grouped, and output pulse time is set to 1s. When user sends ON command, RO2.1-2 are pulsed for 1s, and relay is set. This will connect NO

contact and IN2 will turn on (assuming it is not inverted). When user sends OFF command, RO1.1-2 are pulsed, and relay is reset, turning IN1 on.

Output detection with inputs

Users can detect an output change with inputs.

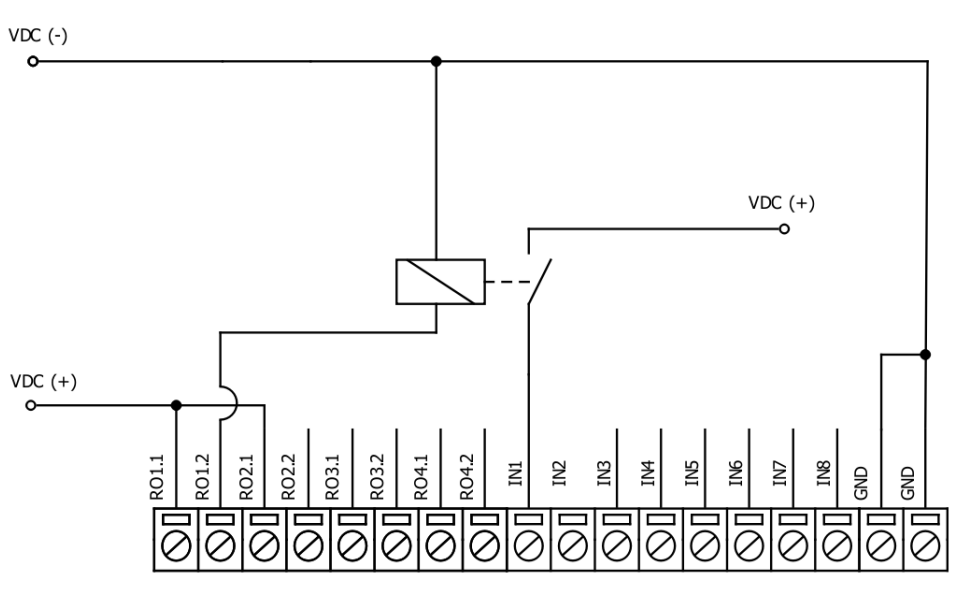


Fig. 4.

To find out if relays are turned on, user can connect relay outputs to IMod inputs (maximum allowed voltage must be taken into account).

Technical information

| | System | |
|----|----------------------------------|--|
| 1. | Dimensions | 101 x 119 x 22.5 mm |
| 2. | Case | IP20, blend PC/ABS self-extinguishing, black |
| 3. | Working environment | Indoors |
| 4. | Operating temperature | from -30°C to +70°C |
| 5. | Humidity | 5-95% RH (non-condensing) |
| 6. | Configuration | USB – serial console, MODBUS RTU |
| 7. | Firmware upgrade | USB – mass storage device |
| | Electrical specifications | |
| | Inputs | |
| 8. | Nominal input voltage range | 12-48VDC |
| 9. | Isolation | 8 X 3kV _(rms) |

| | | |
|-----|--------------------------------------|-----------------------------------|
| 10. | Maximum input voltage | 60VDC |
| | Output relay contacts ratings | |
| 11. | Isolation | 4 X 3kV _(rms) |
| 12. | Rated resistive load | 5 A (at 250 VAC or at 30 VDC) |
| 13. | Rated inductive (L/R=7ms) load | 2 A (at 250 VAC or at 30 VDC) |
| 14. | Maximum switching voltage | 277 VAC 125 VDC |
| 15. | Maximum switching current | 5 A |
| | Power | |
| 16. | Power Supply | 12-24 VDC / 9-33 VDC (full range) |
| 17. | Current consumption | 70 mA |

In the Fig. 5. you can see internal structure of 8DI4RO IOmod. An important fact to metion is that as it can be noticed in the scheme below all inputs are isolated. However ground connection is common for all phototransistors.

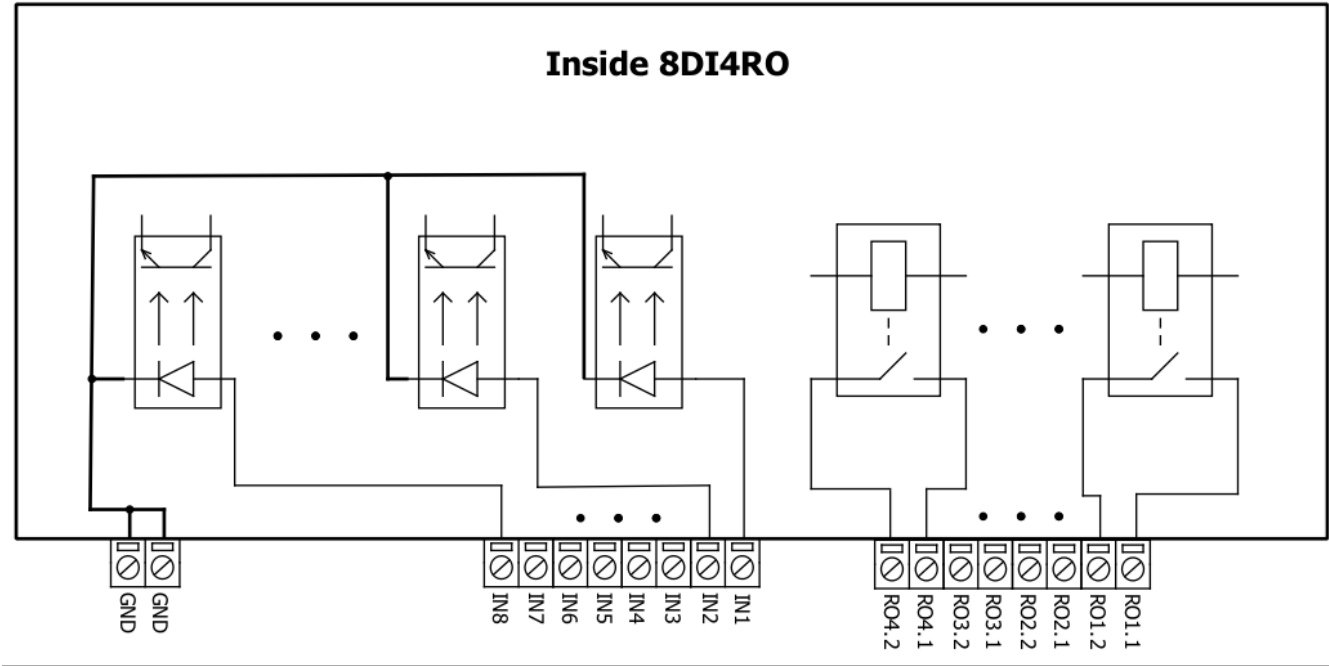


Fig. 5.

Mounting and installation guide

IOmod 8DI4RO RS485 interface

IOmod 8DI4RO has an integrated 120Ω termination resistor which can be enabled or disabled over USB configuration. It is recommended to use termination at each end of the RS485 cable. See typical connection diagram on Fig. 6.

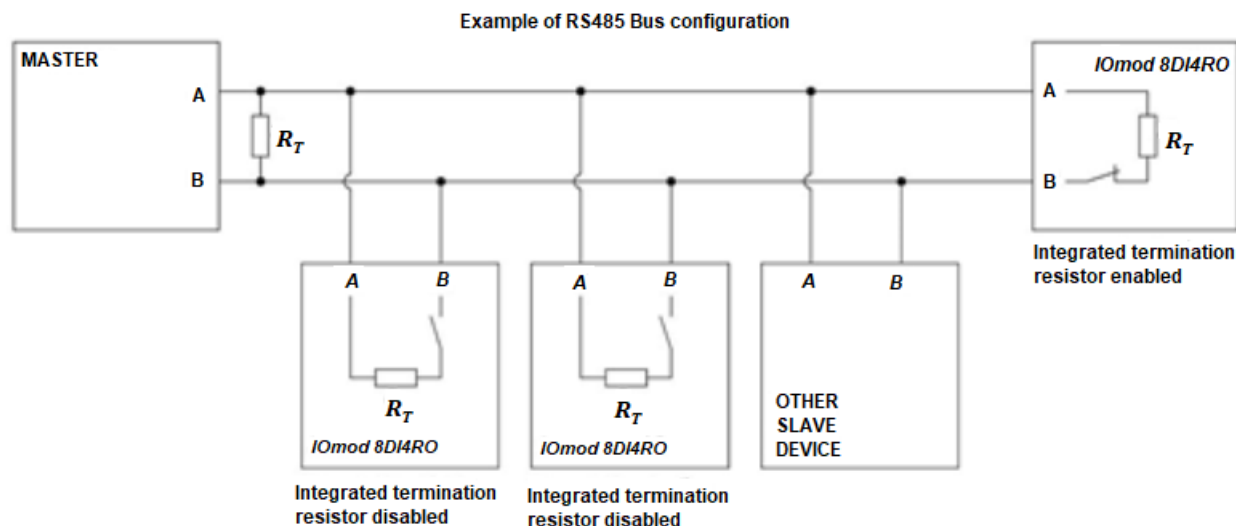


Fig. 6.

IOmod 8DI4RO has 1/8 Unit load receiver which allows to have up to 256 units on line (compared to standard 32 units). To reduce reflections, keep the stubs (cable distance from main RS485 bus line) as short as possible when connecting device.

IOmod 8DI4RO inputs

Typical application of IOmod 8DI4RO inputs is shown on Fig. 2. When default configuration for inputs is applied, user will see inputs connected to +12-24V as "high" or state "1" and input status LED will glow.

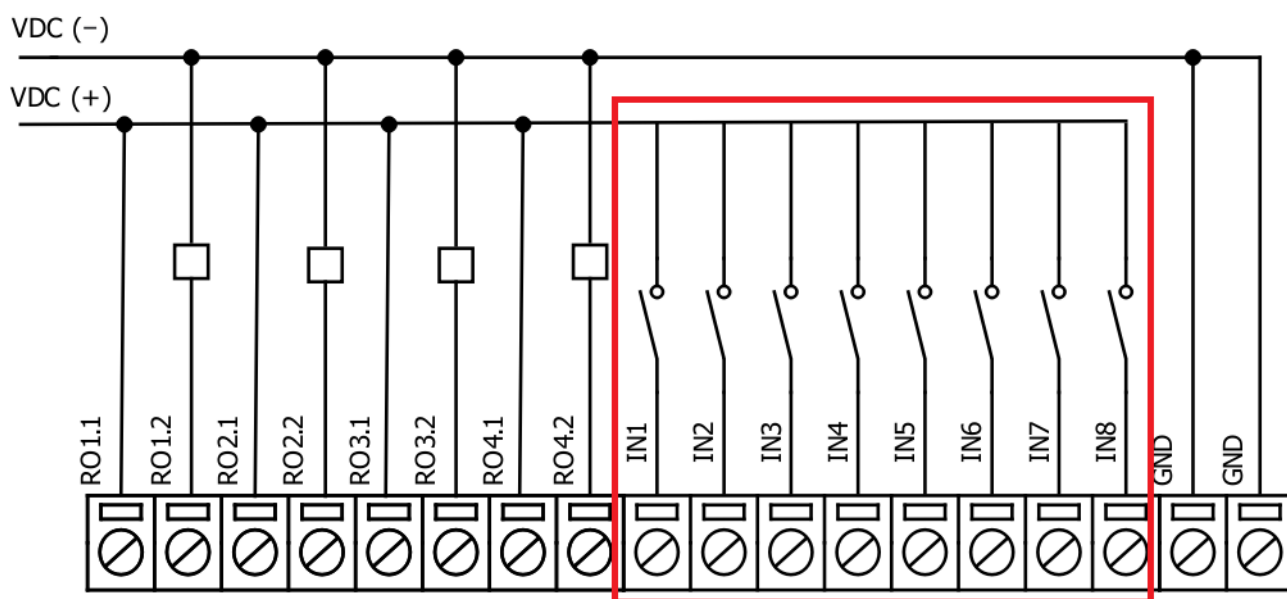


Fig. 7.

User also can configure to enable software input inversion. With this configuration, user will see inputs connected to 0V (see Fig. 3.) as "high" or state "1", input status LED will NOT glow.

IOmod 8DI4RO outputs

IOmod 8DI4RO has 4 relay outputs. Internal clamp diodes are connected to each output which makes IOmod 8DI4RO ideal for driving inductive loads like relays. Maximum 10A per output is allowed. For higher loads outputs can be connected in parallel. Make sure your power supply can provide enough power. Typical application of outputs is shown on Fig. 4.

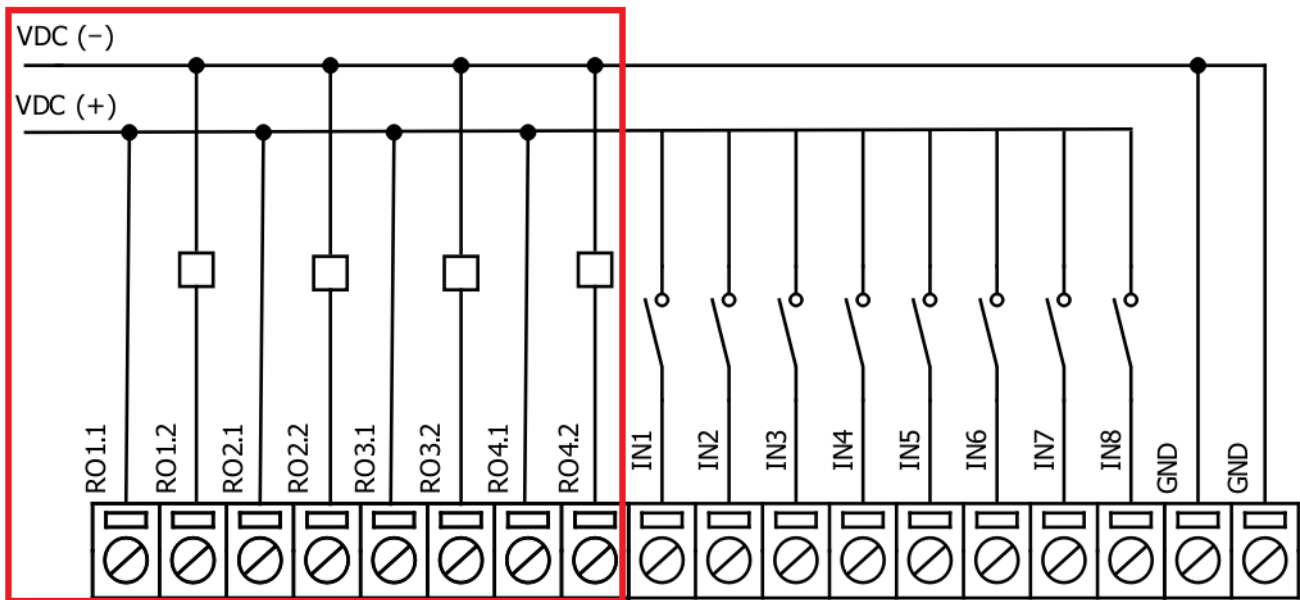


Fig. 8.

Configuration over USB

Driver installation

Device requires USB drivers to work as a Virtual COM port. First-time connection between device and computer could result in "Device driver software was not successfully installed" error (Fig. 5.).

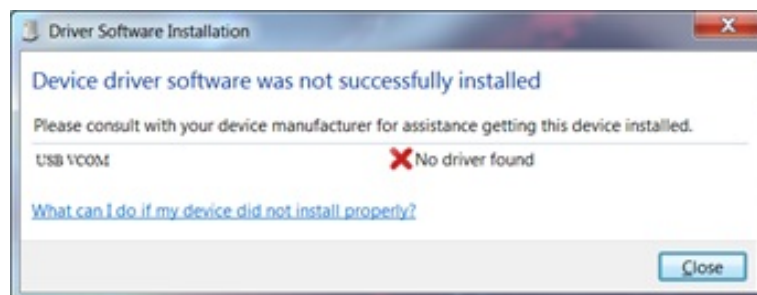


Fig. 10.

Drivers can be installed from Wiki Esleseta -> Downloads -> IOMOD Series Downloads-> Firmware and tools -> IOMOD USB drives for Windows;

User then manually installs drivers by selecting downloaded driver folder:
Go to Control Panel -> Hardware and Sound -> Device Manager
Select failed device;
Press "Update driver software"; screen in Fig. 6. should appear:

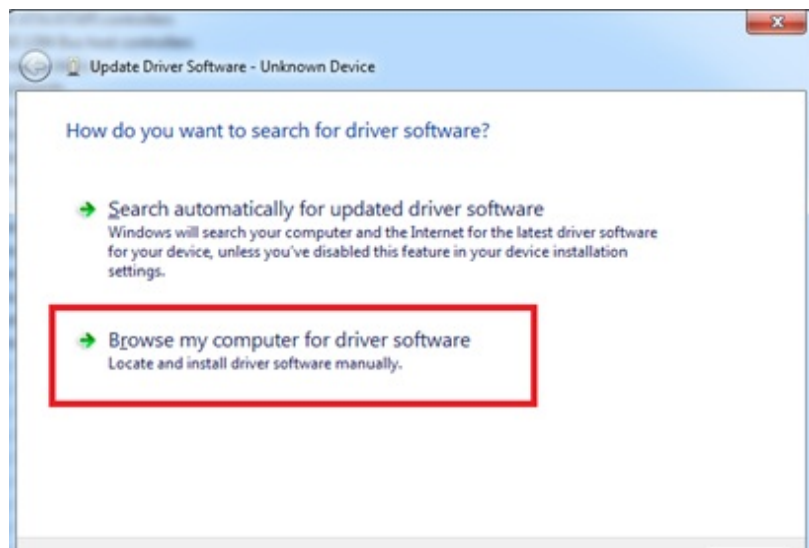


Fig. 11.

Select "x86" driver for 32-bit machine, or x64 for 64-bit machine. If not sure, select root ("vcom") folder (folder in which x64 and x86 lays inside).

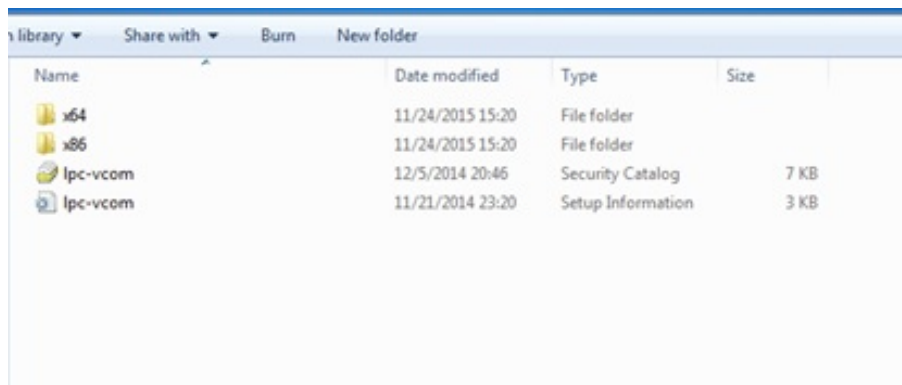


Fig. 12.

IOmod configuration with PuTTY terminal

Configuration of IOmod device is done through CLI (Command Line Interface) on a virtual COM port. Drivers needed for MS Windows to install VCOM will be provided. To open up CLI simply connect to specific V-COM port with terminal software (it is advised to use PuTTY terminal software. If other software is being used, user might need to send <return> symbol after each command). When connected user should immediately see main screen. Accidental close of the terminal window doesn't stop USB connection, user can connect terminal program again, and press any key on keyboard to show up main menu again.

IOMOD 8DI4RO User Manual IEC 60870-5-101

Introduction

IOMod 8DI4RO is a small-size stand-alone Modbus RTU, IEC 60870-5-103 or IEC-60870-5-101 digital input and relay output controller (protocol depends on firmware). IOMod can be used for industrial applications, where digital signaling is used and robust communication is needed. IOMod is an ideal solution for applications such as data acquisition, control, and process monitoring at remote places. This user manual is written for IEC 60870-5-101 protocol firmware version.

Features

- 8 digital inputs;
- Configurable active input signal polarity or input inversion;
- 4 relay outputs;
- Galvanically isolated inputs and outputs;
- Pulsed or latched mode for individual outputs;
- Configuration over USB console or MODBUS RTU;
- Values with data and time information;
- Drag and Drop firmware upgrade over USB mass storage;
- Modbus RTU, IEC-60870-5-103, IEC-60870-5-101 communication over RS485;
- Software selectable R_T 120 Ohm termination resistor on RS485 interface;
- LED indication for input/output and data transmission;
- Easy integration with WCC Lite gateway and CloudIndustries.eu platform;

Operational information

IOMod 8DI4RO uses Modbus RTU, IEC 60870-5-103, or IEC 60870-5-101 protocol to communicate with the master device over the RS485 interface. Protocol used by the device can be changed by uploading the corresponding firmware. Default communication settings are 9600 baud rate, 8E1, Link address - 1.

IOMod 8DI4RO configuration can be changed over a USB interface with a terminal console like PuTTY or similar.

Status LED

Status LED can be in 2 colors :

Blue - Device connected to USB.

Green - Normal operation.

Rx/Tx LED

The RX/TX LED on the IOMod flashes when data is either being transmitted or received via the RS485 port.

IEC 60870-5-101 working information

Initialization

IOMod uses a standard IEC-60870-5-101 communication scheme. Initiation, control messages, and queries are initiated by the master (controlling station), while the IOMod device (controlled station) only answers these requests. Therefore, the first message should be sent by the master to request status of link (function code = 9). This message is answered by IOMod with the status of link (function code = 11) if link is available. Otherwise there is going to be no response. After receiving the status of link the master will send Reset of remote link command (function code = 0) to restart the communication. The IOMod can respond with either acknowledgment (function code = 0, ACK) or Negative Acknowledgment (function code = 1, NACK). If IOMod respond is ACK then the initialization procedure is finished. The described procedure enables the master to proceed with sending other messages defined by the IEC-60870-5-101 protocol.

Data polling

When initialization is complete, the master may request data from the IOMod device with general interrogation. Although according to the protocol specification IOMod will send data on value change. The 8DI4RO IOMod responds with data type 1 (M_SP_NA_1) a single-point information.

Output control

To control device outputs master (controlling station) sends a command conforming to the IEC-60870-5-101 protocols. The IOAs for outputs are 101 (RO1), 102 (RO2), 103 (RO3) and 104 (RO4). By sending the 45-type command (C_SC_NA_1, single point command) with the selected IOA you can either turn ON or OFF outputs. Negative acknowledgement is returned if the sent command is not configured. For example wrong IOA address or Data Type.

Inputs

The IOAs for the inputs are: 9 (DI1), 10 (DI2), 11 (DI3), 12 (DI4), 13 (DI5), 14 (DI6), 15 (DI7) and 16 (DI8).

Time synchronization

To initiate the time synchronization between devices the master must send a Clock Sync command. The command type is C_CS_NA_1 (103) and the Cause of Transmission (COT) has to be 6. The command has to be sent to the correct link address and CASDU, which is the same as the link address by default. If the sent frame is correct the IOMod will respond with a C_CS_NA_1 (103) type command with the COT (cause of transmission) of 0x07 and the p/n bit will be set to 0 (positive confirmation). Also the command will be time-tagged with the **device** time. If the time synchronization feature is disabled or the command is sent to an undefined CASDU the response is the same except the p/n bit will be set to 1 (negative confirmation) and the COT byte will be equal to 0x47.

General interrogation

General Interrogation (GI) is initiated by the master sending the General Interrogation command. The command type is C_IC_NA_1 (100) and the Cause of Transmission (COT) has to be 6. The command has to be sent to the correct link address and CASDU, which is the same as the link address by default. If the sent frame is correct the IOMod will respond with a C_IC_NA_1 (100) type command with the COT (cause of transmission) of 0x07 and the p/n bit will be set to 0 (positive confirmation). Otherwise, it will respond with the same command just that the p/n bit will be set to 1 (negative confirmation) and the COT byte will be equal to 0x47. Then the device will begin to send all of its data. If the GI was successful the IOMOD will also send another 100 type command with the COT (cause of transmission) of 10 (ActTerm) meaning the general interrogation is over.

Device configuration

Input inversion and polarity selection

Unlike 8DI8DO IOMod 8DI4RO does not have pull-up resistors. If user desires to turn input status on, when that input signal is low, user then inverts inputs logically. All input indication LED's stay the same (are not inverted).

Input / Output grouping

Sometimes two inputs or two outputs must be captured as one DPI input or output. Inputs and outputs can be grouped into the pairs of two. This allows outputs to be controlled by one DPI command (of address of first output in the group). Only two neighbor pins can be grouped into pair, while first pin in pair must be an odd number pin. When grouped, second pin in the pair is not used anymore – all requests for this pin generate an error. For example – RO1.1-2 and RO2.1-2 can be grouped, after that RO2.1-2 is not used; RO2.1-2 and RO3.1-2 cannot be grouped; RO3.1-2 and RO4.1-2 can be grouped, but RO4.1-2 is not used then, etc.

Fig. 1. picture shows outputs and inputs ungrouped and controlled independently. In this mode, General Interrogation will be composed of 4 output states and 8 input states.

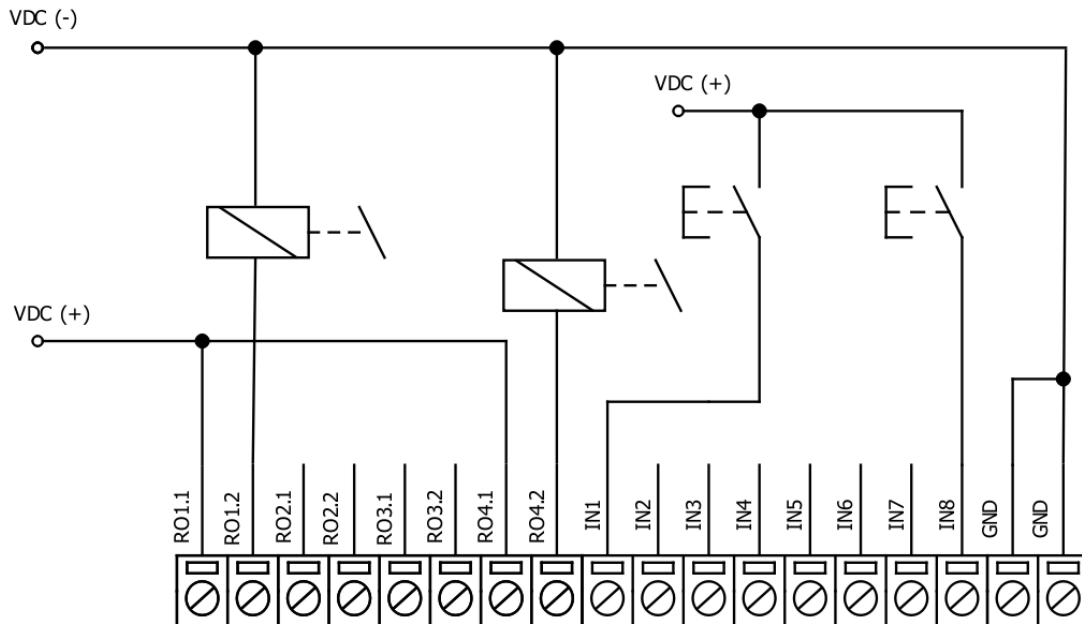


Fig. 1.

Let's consider a situation shown in Fig. 2. In the picture below first two relay outputs are grouped into pair. Also 3rd and 4th inputs are grouped into pair. Now, General interrogation will be composed of 3 output states (with RO2.1-2 missing), and 7 input states (with IN4 missing). Output and input numbers is represented by "Info number" in protocol.

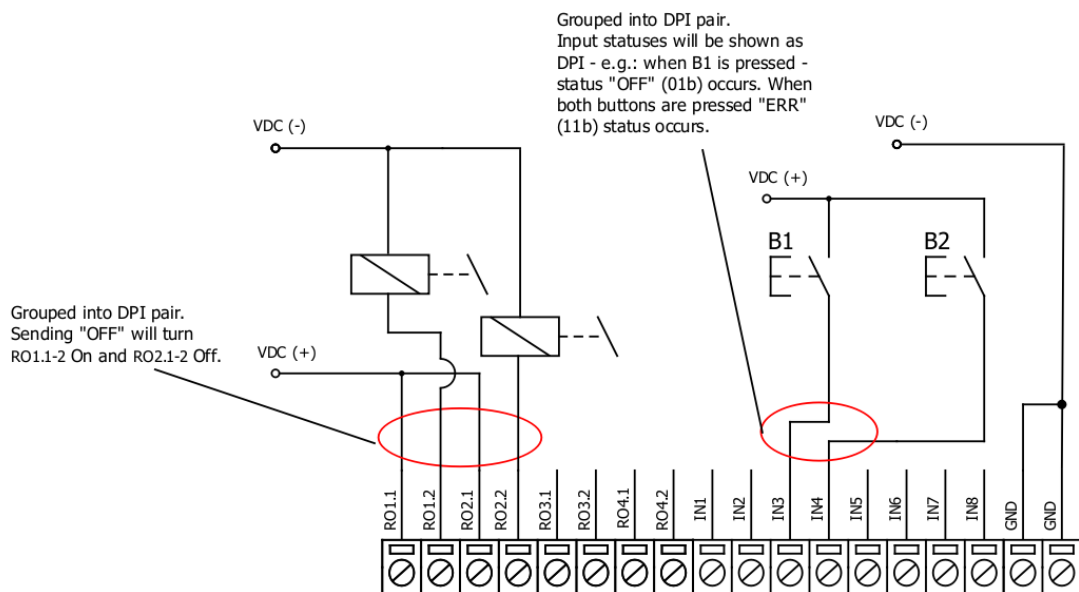


Fig. 2.

Input filter

Input filter is a simple glitch filter with time input. This filter time corresponds to stable time that input must achieve before sending a status change.

Output pulse time

User can configure outputs to be pulse controlled – it means that output will be turned on for configured amount of time. When this time runs out, output is turned off. This is useful when pulse toggle relays are used. Output pulse is independent from output grouping option and can be used on both grouped and ungrouped outputs. When output is grouped, device will allow only one command completion at a time – when output is already turned ON, other “turn ON” requests will be responded with NACK. If user desires latching outputs to be used, output pulse time is set to 0.

In the picture below is depicted an example of pulse output usage. Inputs and outputs are grouped, and output pulse time is set to 1s. When user sends ON command, RO2.1-2 are pulsed for 1s, and relay is set. This will connect NO contact and IN2 will turn on (assuming it is not inverted). When user sends OFF command, RO1.1-2 are pulsed, and relay is reset, turning IN1 on.

Output detection with inputs

Users can detect an output change with inputs.

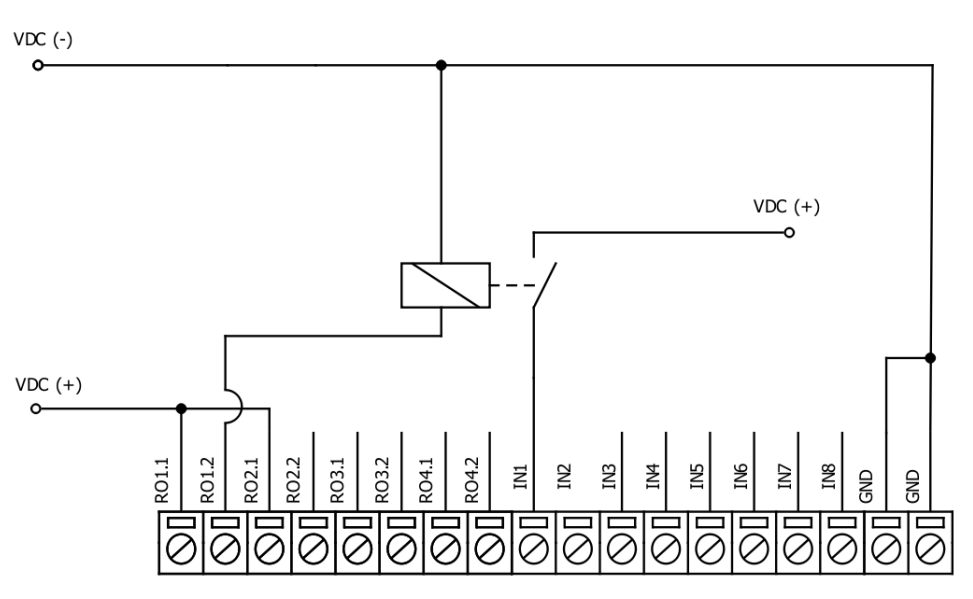


Fig. 3.

To find out if relays are turned on, user can connect relay outputs to IMod inputs (maximum allowed voltage must be taken into account). When relays are turned on, device responds with IEC-60870-5-101 protocol message “Remote Operation”. If inputs are never turned on or off, device will send “Remote Operation” message after time-out period, with current input statuses. Time-out period is configured by user as a Feedback Time.

Technical information

| | System | |
|-----|-----------------------------|--|
| 1. | Dimensions | 101 x 119 x 22.5 mm |
| 2. | Case | IP20, blend PC/ABS self-extinguishing, black |
| 3. | Working environment | Indoors |
| 4. | Operating temperature | from -30°C to +70°C |
| 5. | Humidity | 5-95% RH (non-condensing) |
| 6. | Configuration | USB – serial console, MODBUS RTU |
| 7. | Firmware upgrade | USB – mass storage device |
| | Electrical specifications | |
| | Inputs | |
| 8. | Nominal input voltage range | 12-48VDC |
| 9. | Isolation | 8 X 3kV _(rms) |
| 10. | Maximum input voltage | 60VDC |

| | Output relay contacts ratings | |
|-----|--------------------------------|-----------------------------------|
| 11. | Isolation | 4 X 3kV _(rms) |
| 12. | Rated resistive load | 5 A (at 250 VAC or at 30 VDC) |
| 13. | Rated inductive (L/R=7ms) load | 2 A (at 250 VAC or at 30 VDC) |
| 14. | Maximum switching voltage | 277 VAC 125 VDC |
| 15. | Maximum switching current | 5 A |
| | Power | |
| 16. | Power Supply | 12-24 VDC / 9-33 VDC (full range) |
| 17. | Current consumption | 70 mA |

In the Fig. 5. you can see internal structure of 8DI4RO IOmod. An important fact to metion is that as it can be noticed in the scheme below all inputs are isolated. However ground connection is common for all phototransistors.

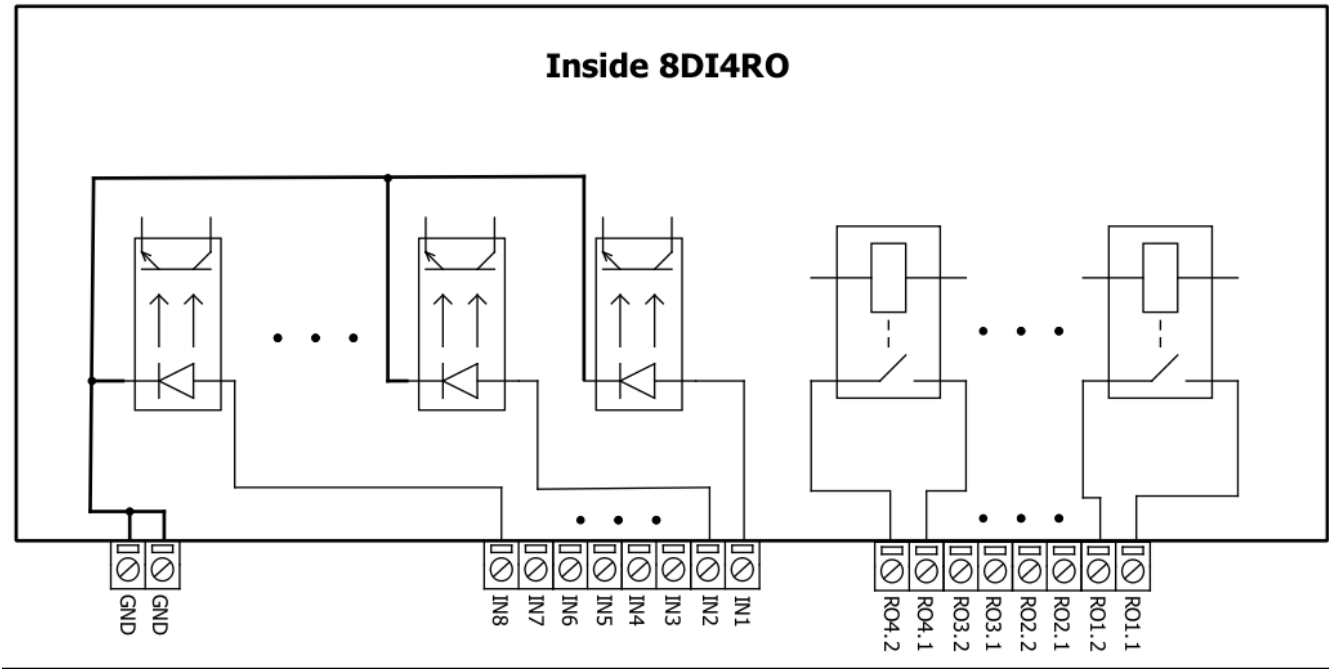


Fig. 4.

Mounting and installation guide

IOmod 8DI4RO RS485 interface

IOmod 8DI4RO has an integrated 120Ω termination resistor which can be enabled or disabled over USB configuration. It is recommended to use termination at each end of the RS485 cable. See typical connection diagram on Fig. 6.

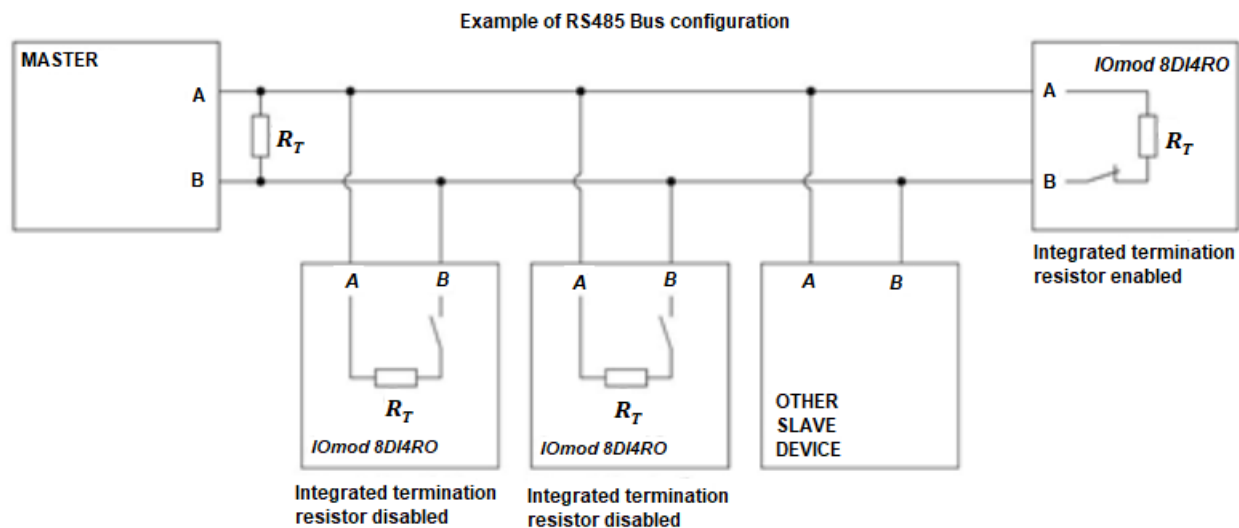


Fig. 5.

IOmod 8DI4RO has 1/8 Unit load receiver which allows to have up to 256 units on line (compared to standard 32 units). To reduce reflections, keep the stubs (cable distance from main RS485 bus line) as short as possible when connecting device.

IOmod 8DI4RO inputs

Typical application of IOmod 8DI4RO inputs is shown on Fig. 2. When default configuration for inputs is applied, user will see inputs connected to +12-24V as "high" or state "1" and input status LED will glow.

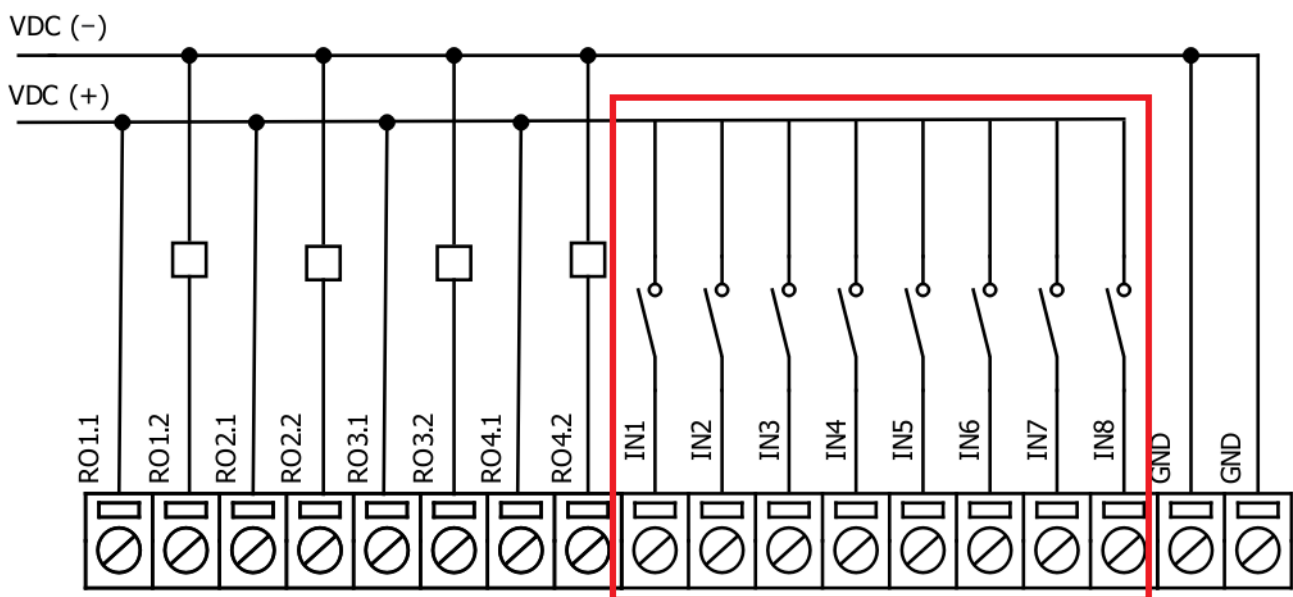


Fig. 6.

User also can configure to enable software input inversion. With this configuration, user will see inputs connected to 0V (see Fig. 3.) as "high" or state "1", input status LED will NOT glow.

IOmod 8DI4RO outputs

IOmod 8DI4RO has 4 relay outputs. Internal clamp diodes are connected to each output which makes IOmod 8DI4RO ideal for driving inductive loads like relays. Maximum 10A per output is allowed. For higher loads outputs can be connected in parallel. Make sure your power supply can provide enough power. Typical application of outputs is shown on Fig. 4.

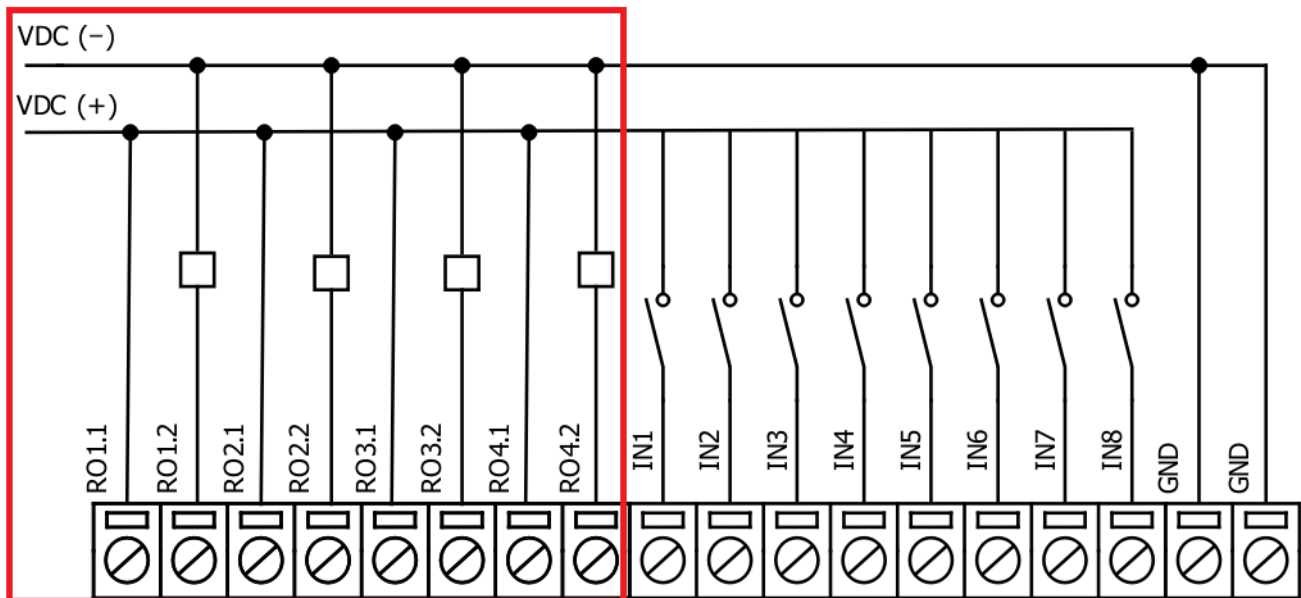


Fig. 7.

Configuration over USB

Driver installation

Device requires USB drivers to work as a Virtual COM port. First-time connection between device and computer could result in "Device driver software was not successfully installed" error (Fig. 5.).

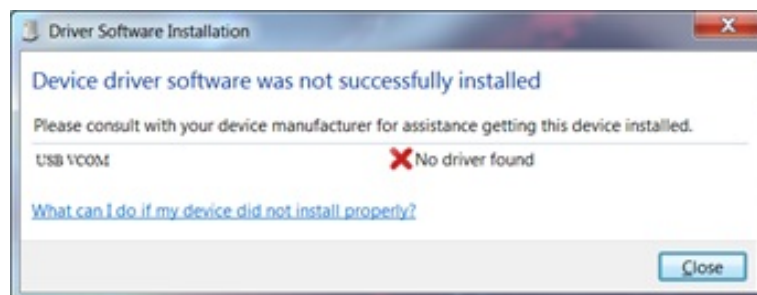


Fig. 8.

Drivers can be installed from Wiki Esleseta -> Downloads -> IOMOD Series Downloads-> Firmware and tools -> IOMOD USB drives for Windows;

User then manually installs drivers by selecting downloaded driver folder:
Go to Control Panel -> Hardware and Sound -> Device Manager
Select failed device;
Press "Update driver software"; screen in Fig. 6. should appear:

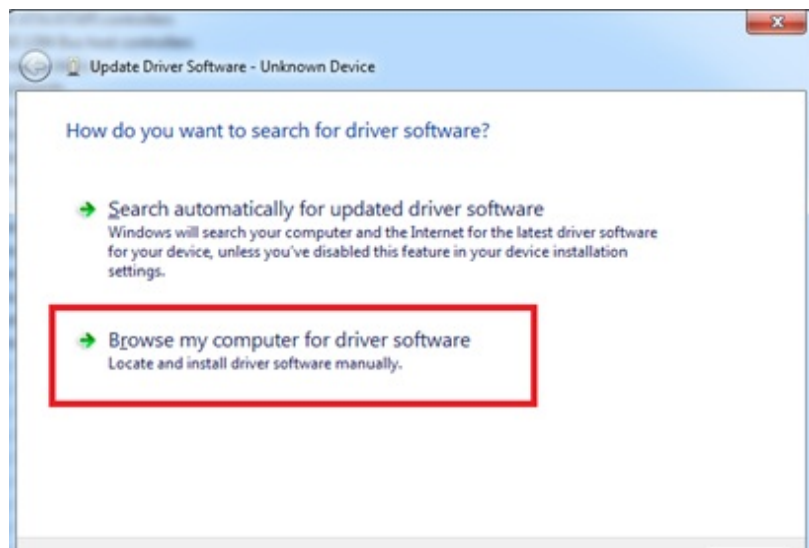


Fig. 9.

Select "x86" driver for 32-bit machine, or x64 for 64-bit machine. If not sure, select root ("vcom") folder (folder in which x64 and x86 lays inside).

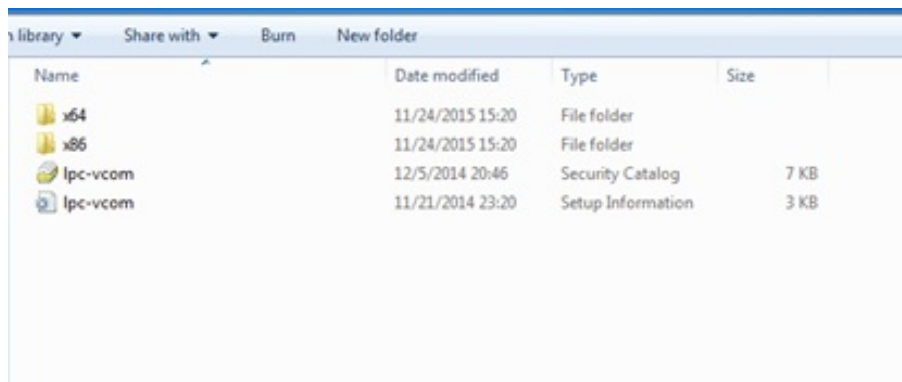


Fig. 10.

IOMod configuration with PuTTY terminal

Configuration of IOMod device is done through CLI (Command Line Interface) on a virtual COM port. Drivers needed for MS Windows to install VCOM will be provided. To open up CLI simply connect to specific V-COM port with terminal software (it is advised to use PuTTY terminal software. If other software is being used, user might need to send <return> symbol after each command). When connected user should immediately see main screen. Accidental close of the terminal window doesn't stop USB connection, user can connect terminal program again, and press any key on keyboard to show up main menu again.

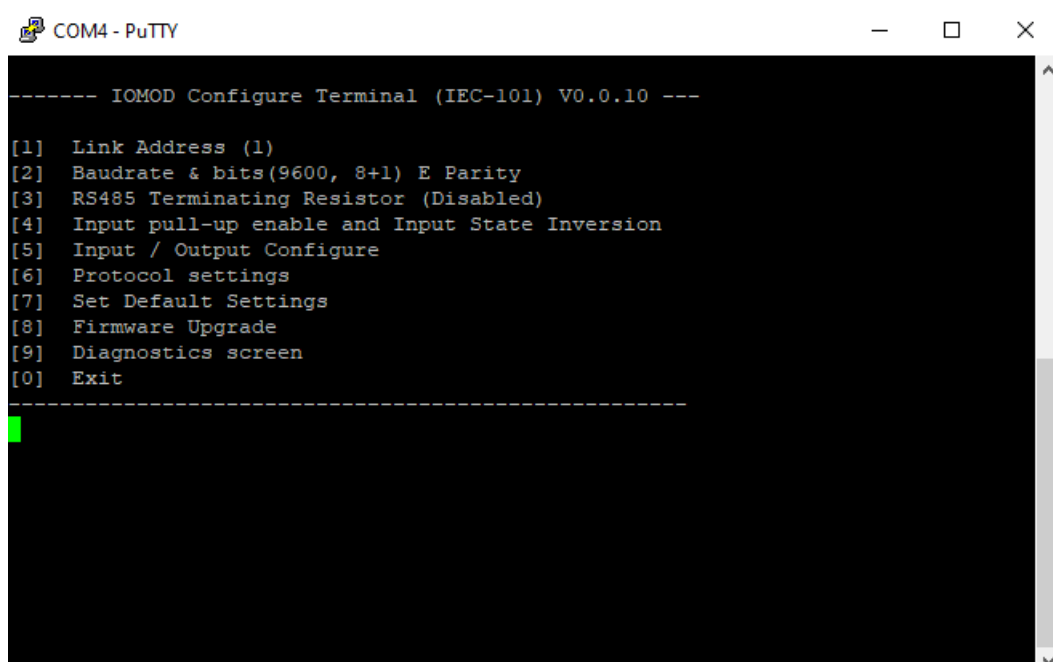
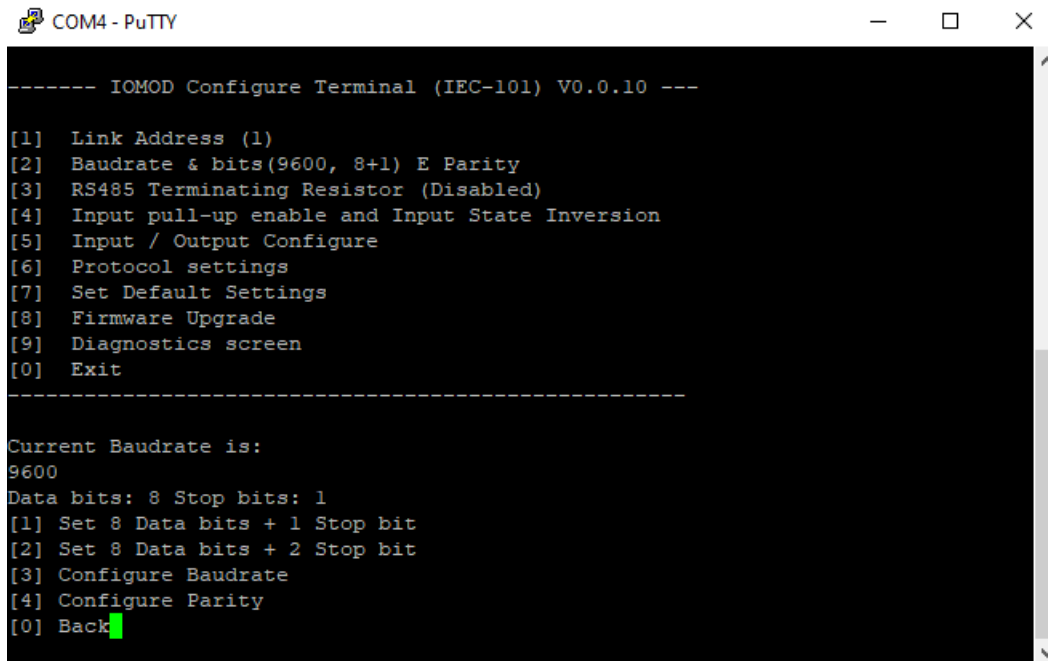


Fig. 11.

Navigation is performed by sending numbers to terminal. User then proceeds by following further on-screen instructions. For example, to set Baud rate, press [2] to enter baud rate screen, press [3] to enable or disable Terminating Resistor. When done, press [0] (exit) before disconnecting device. Default values is set by pressing [7] on main screen and confirming changes [1]. Baud rate window is shown in Fig. 8.



```
COM4 - PuTTY

----- IOMOD Configure Terminal (IEC-101) V0.0.10 -----

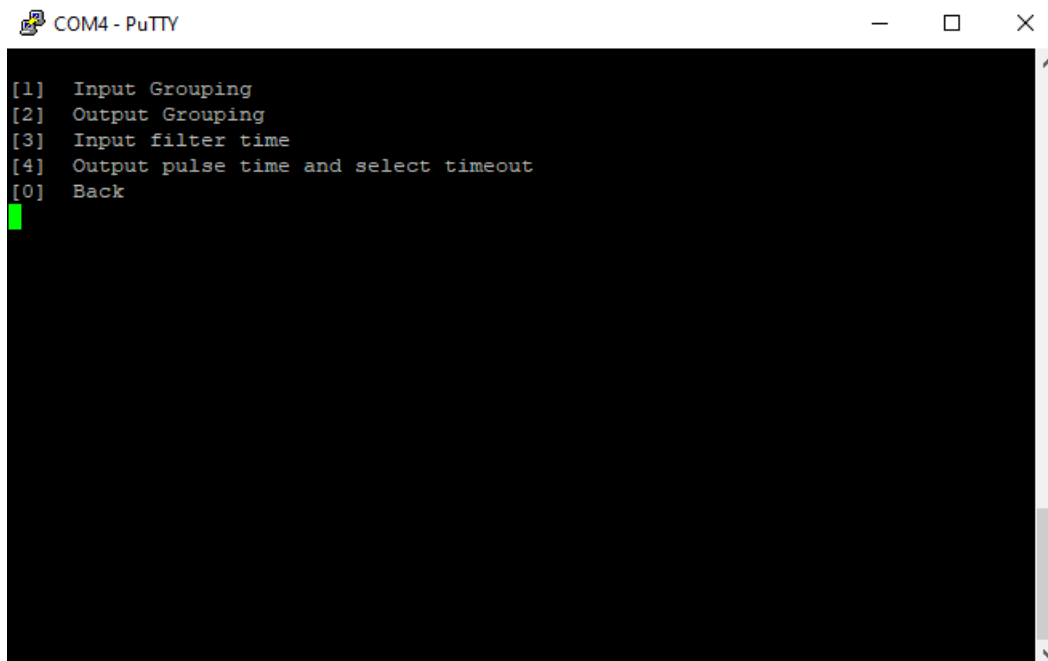
[1] Link Address (1)
[2] Baudrate & bits(9600, 8+1) E Parity
[3] RS485 Terminating Resistor (Disabled)
[4] Input pull-up enable and Input State Inversion
[5] Input / Output Configure
[6] Protocol settings
[7] Set Default Settings
[8] Firmware Upgrade
[9] Diagnostics screen
[0] Exit

-----

Current Baudrate is:
9600
Data bits: 8 Stop bits: 1
[1] Set 8 Data bits + 1 Stop bit
[2] Set 8 Data bits + 2 Stop bit
[3] Configure Baudrate
[4] Configure Parity
[0] Back
```

Fig. 12.

A lot of options can be changed after entering Input / Output configure screen (Fig. 9.). These include filter time, output pulse time, input and output configuration, output detection with inputs screens.



```
COM4 - PuTTY

[1] Input Grouping
[2] Output Grouping
[3] Input filter time
[4] Output pulse time and select timeout
[0] Back
```

Fig. 13.

Input (Fig. 10.) and output grouping screen show the connection between neighbor pins. Vertical bars show that input or outputs are not grouped.

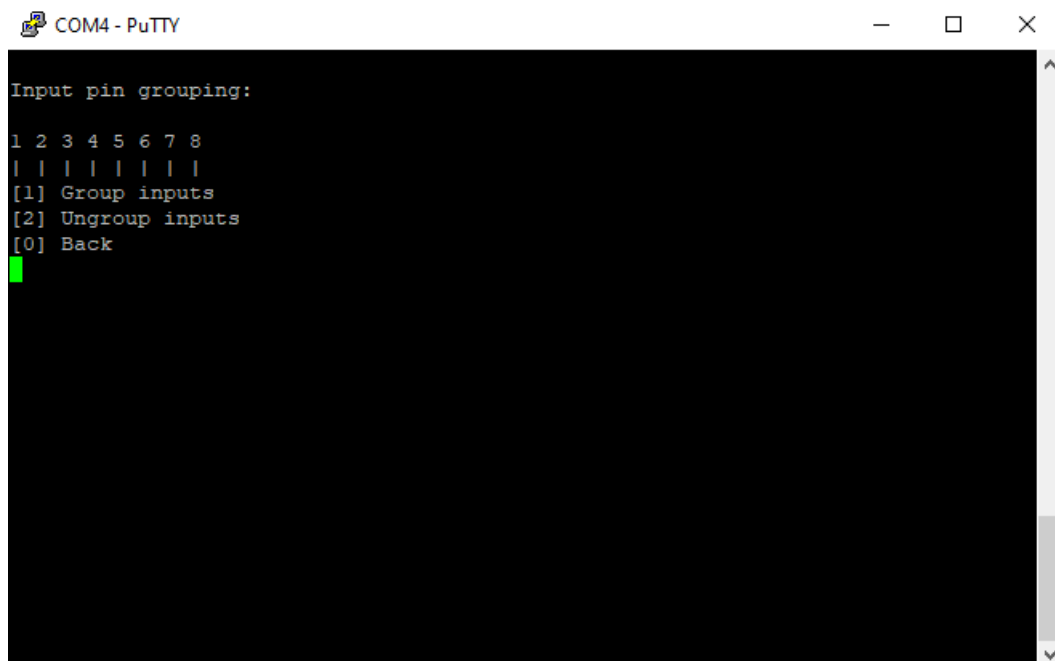


Fig. 14.

A connection between a grouped pair of inputs or outputs is depicted by a slash and a backslash that connects beneath the pair (Fig. 11.).

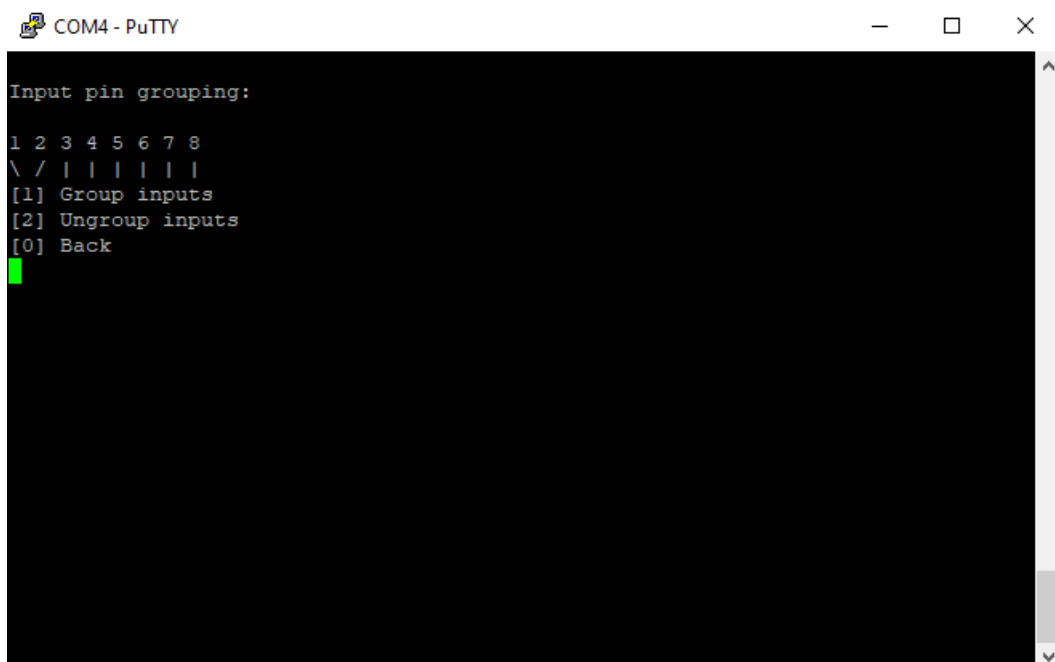


Fig. 15.

Changes in the device and firmware version are shown in a Diagnostics Screen. Such diagnostics screen for IOmod 8DI4RO is shown in Fig. 12.

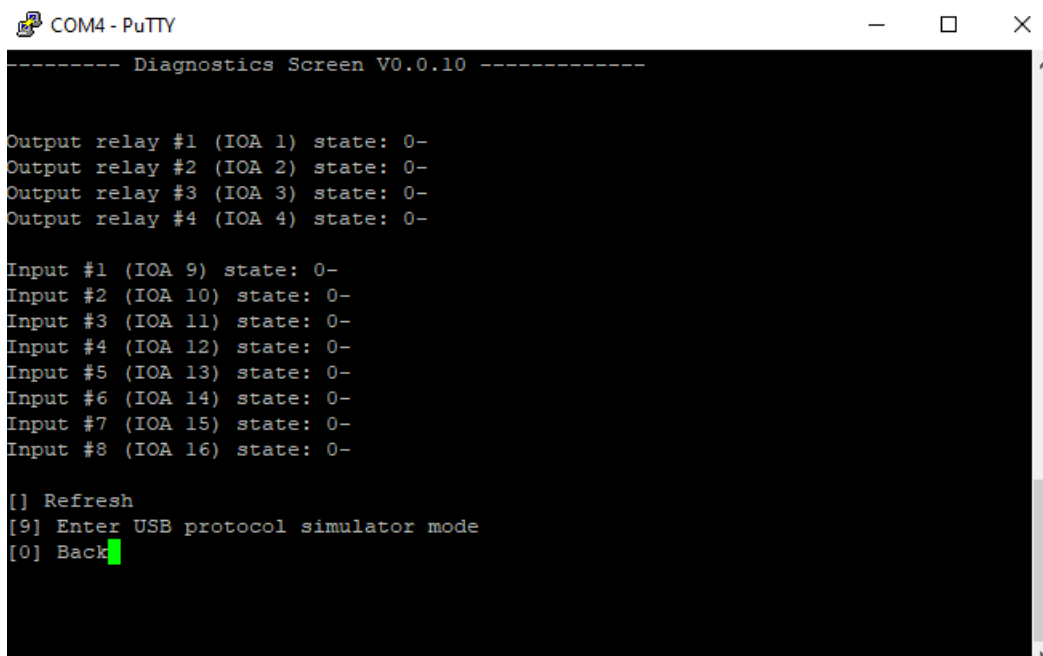


Fig. 16.

Main Menu

| | Menu name | Function | Values | Default values |
|-----|--|--|--|----------------------------------|
| 1. | Link Address | Setts Link address | 1-255 | 1 |
| 2. | Baudrate & bits(9600, 8+1) E Parity | Enters configuring screen for communication settings | 8+1 or 8+2 (Data + Stop), None, Odd, Even, Mark, Space (Parity) | 9600, 8+1, Parity -Even |
| 3. | RS485 Terminating Resistor | RS485 120 Ohms Terminating Resistor | 1 - 2 (Enable/Disable) | Disabled |
| 4. | Input pull-up enable and Input State Inversion | Enables input pull-up resistor. Inputs then activated by low signal; Input inversion (Inverts input states in protocol logic) | 1 - 2 (Enable/Disable) | Both disabled |
| 5. | Input / Output configure | Enters screen for configuring (see 6.1 - 6.5 rows below) | - | - |
| 5.1 | Input grouping; | Groups or ungroups inputs | 8 inputs ungrouped / 4 pairs grouped | All inputs ungrouped by default |
| 5.2 | Output grouping; | Groups or ungroups outputs | 8 outputs ungrouped / 4 pairs grouped | All outputs ungrouped by default |

| | | | | |
|-----|---------------------------------------|---|--|---------|
| 5.3 | Input filter time; | Input glitch filter – minimum stable time to detect input | 1 – 60000 milliseconds | 100 |
| 5.4 | Output pulse time and select timeout; | Sets short and long pulse time and selects timeout for individual outputs. | 0 – 60000 milliseconds (0 if not used) | 0 |
| 6. | Protocol settings; | Sets the time mode for info objects, IOA size and enables or disables the use of command qualifier for output pulses. | - | - |
| 6.1 | Toggle 24/56 bit time | Changes toggle bit time either to 24 or to 56 | 24/56 | 56 |
| 6.2 | Change IOA size | Changes IOA size | 1-3 | 1 |
| 6.3 | Toggle use of command qualifier | Enables or disables toggle use of command qualifier | Disabled/Enabled | Enabled |
| 7. | Set default settings; | Sets Default Settings | (1 to confirm, 0 to cancel) | - |
| 8. | Firmware update; | Mass Storage Device Firmware Upgrade | (1 to confirm, 0 to cancel) | - |
| 9. | Diagnostics screen; | Input / Output states | - | - |
| 0. | Exit; | Exit and disconnect | - | - |

Protocol simulator

When entered diagnostics screen, user can turn on USB protocol simulator mode by pressing [9]. When protocol simulator is turned on, device will communicate through USB port rather than RS-485 line. Communication on RS-485 line is closed and all IEC-101 commands will be accepted only from USB. To exit this mode user must restart device.

Firmware upgrade over USB

To update device firmware user must enter main configuration menu and enter Firmware upgrade screen by pressing [8] is shown in Fig. 13.

```

COM4 - PuTTY

----- IOMOD Configure Terminal (IEC-101) V0.0.10 -----
[1] Link Address (1)
[2] Baudrate & bits(9600, 8+1) E Parity
[3] RS485 Terminating Resistor (Disabled)
[4] Input pull-up enable and Input State Inversion
[5] Input / Output Configure
[6] Protocol settings
[7] Set Default Settings
[8] Firmware Upgrade
[9] Diagnostics screen
[0] Exit
-----

Confirm Firmware Upgrade
[1] Confirm
[0] Cancel

```

Fig. 17.

Confirm upgrade by pressing [1];

Device should enter a Firmware Upgrade mode. It means that device switches from USB Console mode into Mass storage device and computer recognize it as USB Storage.

Usually after confirming the upgrade a window with an error message appears (Fig. 14)

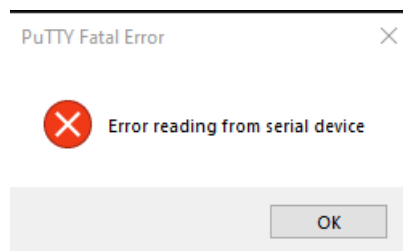


Fig. 18.

User then must **delete** existing file “firmware.bin”, and simply drag and drop new firmware file.

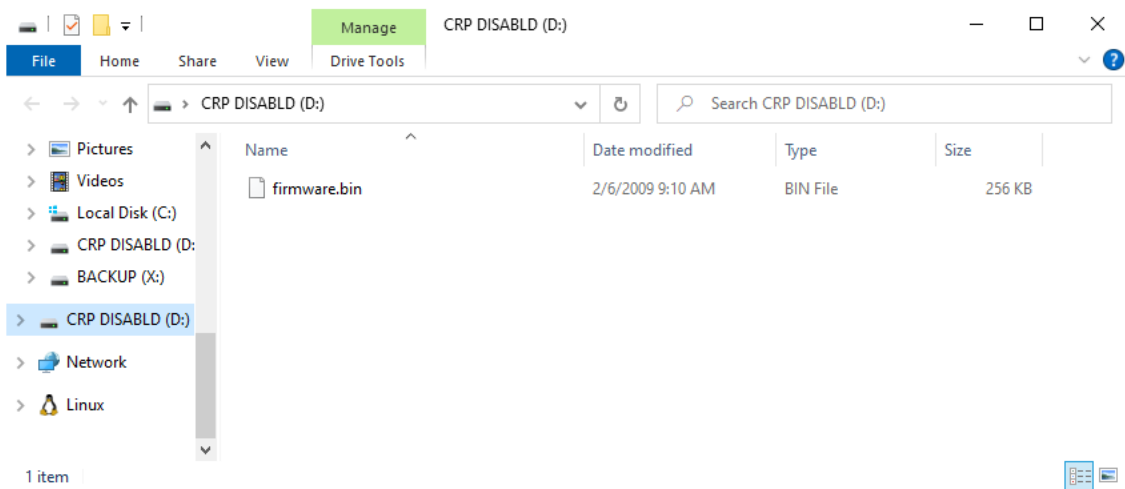


Fig. 19.

Finally, reconnect device and check firmware version. Lastly, you may want to set default settings by pressing [7].

Firmware version 2

IOMOD 8DI4RO User Manual

1. Introduction

IOMOD 8DI4RO is a small-size stand-alone **Modbus RTU, IEC 60870-5-103 or IEC-60870-5-101** digital input and relay output controller (protocol depends on firmware). IOMOD can be used for industrial applications, where digital signalling and robust communication are needed. IOMOD is an ideal solution for applications such as data acquisition, control, and process monitoring at remote places.

1.1 Features

- 8 digital inputs;
- Configurable active input signal polarity or input inversion;
- 4 relay outputs;
- Galvanically isolated inputs and outputs;
- Pulsed or latched mode for individual outputs;
- Possible output feedback measurement with inputs;
- Configuration over USB, RS485 or Ser2Net;
- Values with data and time information;
- Drag and Drop firmware upgrade over USB, RS485 or Ser2Net;
- Modbus RTU, IEC-60870-5-103, IEC-60870-5-101 communication over RS485;
- Software selectable R_T 120 Ohm termination resistor on RS485 interface;
- LED indication for input/output and data transmission;
- Easy integration with WCC Lite gateway and CloudIndustries.eu platform;

1.2 Block diagram

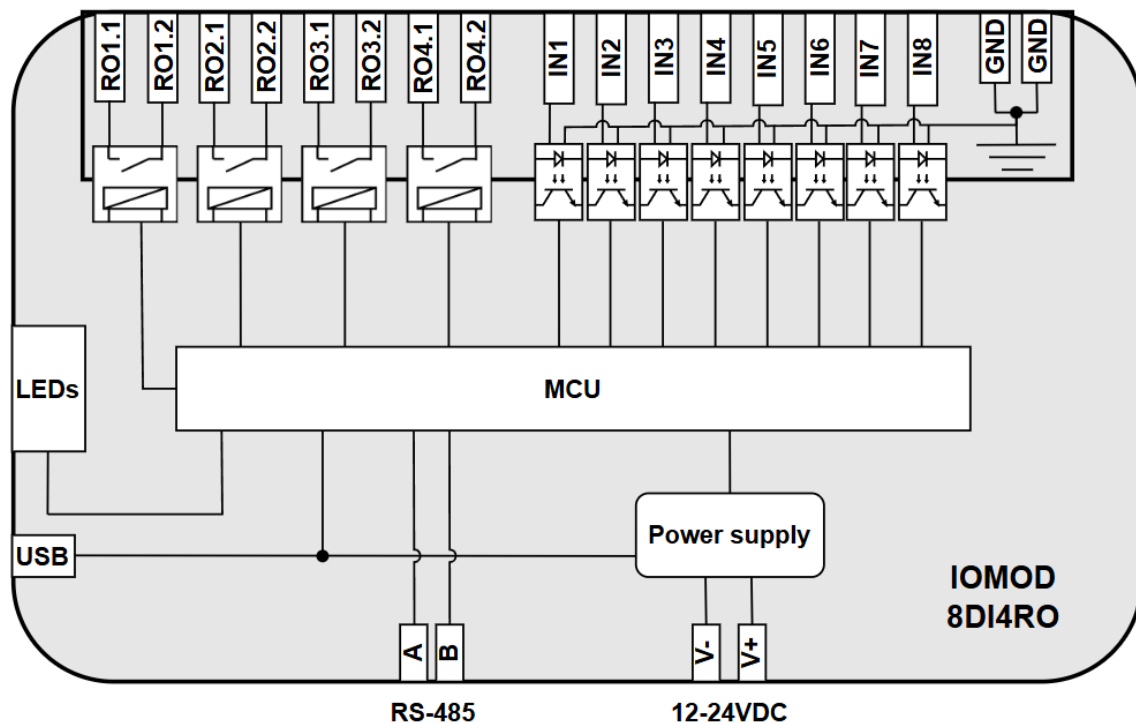


Fig. 1.2.1. IOMOD 8DI4RO internal structure and block diagram

2. Hardware data

2.1 Mechanical drawings

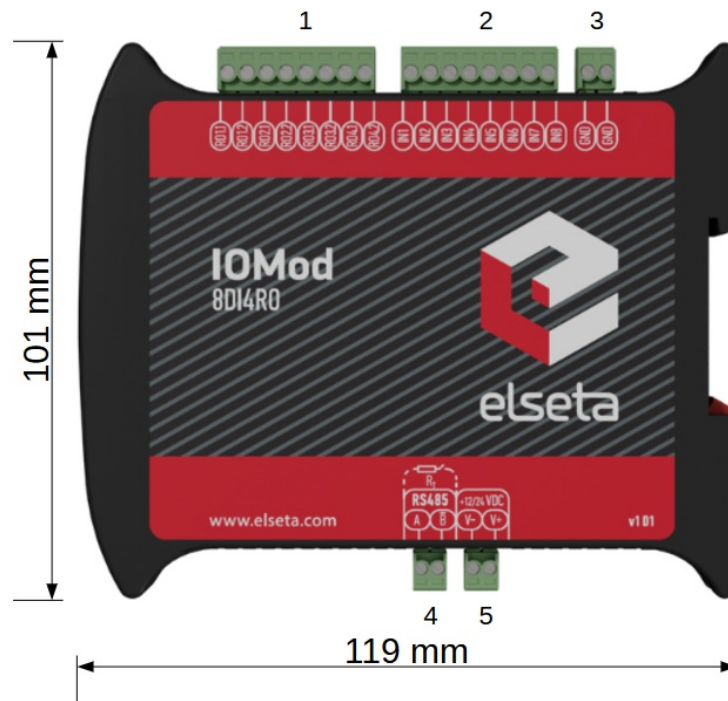


Fig. 2.1.1. IOMod 8DI4RO side view with dimensions and terminals description. 1 - four relay outputs; 2 - eight digital inputs; 3 - ground input; 4 - RS485 interface; 5 - power supply input

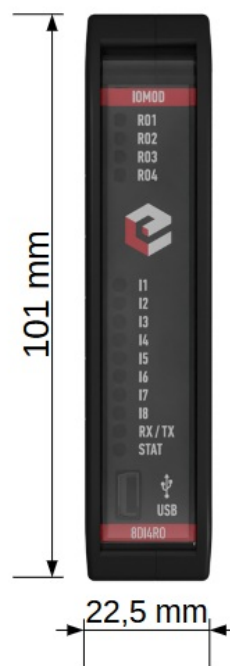


Fig. 2.1.2. IOMod 8DI4RO front view

2.2 Terminal connections

IOMod 8DI4RO has 22 terminals, which are depicted below:



Fig. 2.2.1. IOMod 8DI4RO terminal diagram

The description of each terminal can be found in the table below:

Table 2.2.1. Terminal Specifications

| Terminal number | Terminal name | Description |
|-----------------|---------------|-----------------------|
| 1 | R01.1 | Relay 1 contacts (NO) |
| 2 | R01.2 | |
| 3 | R02.1 | Relay 2 contacts (NO) |
| 4 | R02.2 | |
| 5 | R03.1 | Relay 3 contacts (NO) |
| 6 | R03.2 | |
| 7 | R04.1 | Relay 4 contacts (NO) |
| 8 | R04.2 | |
| 9 | IN1 | |
| 10 | IN2 | |

| | | |
|----|-----------|--------------------------------------|
| 11 | IN3 | Digital inputs |
| 12 | IN4 | |
| 13 | IN5 | |
| 14 | IN6 | |
| 15 | IN7 | |
| 16 | IN8 | |
| 17 | GND | Digital input ground |
| 18 | GND | Digital input ground |
| 19 | A | RS485 non-inverting (positive) input |
| 20 | \bar{B} | RS485 inverting (negative) input |
| 21 | V- | Negative power source input |
| 22 | V+ | Positive power source input |

2.3 Status indication

IOMod 8DI4RO has fourteen LEDs (Fig 2.3.1), which indicate the statuses of the 8 digital inputs, 4 relay outputs, communication, and power.



Fig. 2.3.1. IOMod 8DI4RO LEDs physical location

Table 2.3.1. Description of LEDs.

| Name | LED color | Description |
|---------|------------|--|
| RO1-RO4 | □ (orange) | Indicates digital relay output activation |
| IN1-IN8 | □ (orange) | Indicates digital input activation |
| RX/TX | □ (green) | A blinking green light indicates active communication via the RS485 interface. |
| STAT | □ (green) | The power source is connected to the power supply input. |
| | □ (blue) | IOMod 8DI4RO is connected to an external device via a USB mini cable. |

3. Technical information

Table 4.1. Technical specifications.

| System | | |
|----------------------------------|--|--------------------------|
| Dimension | 101 x 119 x 22.5 mm | |
| Case | IP20, blend PC/ABS self-extinguishing, black | |
| Working environment | Indoors | |
| Operating temperature | From -30°C to +70°C | |
| Recommended operating conditions | 5-95% RH (non-condensing) | |
| Configuration | USB, RS485 | |
| Firmware upgrade | USB, RS485 | |
| Electrical specifications | | |
| Inputs | Nominal input voltage range | 12-48VDC |
| | Isolation | 8 X 3kV _(rms) |
| | | |

| | | |
|---------------------|-----------------------------------|----------------------------|
| Outputs | Maximum input voltage | 60VDC |
| | Isolation | 4 X 3kV _(rms) |
| | Rated resistive load | 5 A (at 250 VAC or 30 VDC) |
| | Rated inductive (L/R=7ms) load | 2 A (at 250 VAC or 30 VDC) |
| | Maximum switching voltage | 277 VAC 125 VDC |
| | Maximum switching current | 5 A |
| Power | | |
| Power Supply | 12-24 VDC / 9-33 VDC (full range) | |
| Current consumption | 70 mA | |

4. Mounting and installation

4.1 Connection Diagrams

In this chapter, the various options for connecting the device are discussed.

4.2 Digital inputs

The typical application of IOMod 8DI4RO inputs is shown in Fig. 4.2.1. When the default input configuration is applied, the user will see inputs connected to +12-24V as “high” or state “1” and the input status LED will glow.

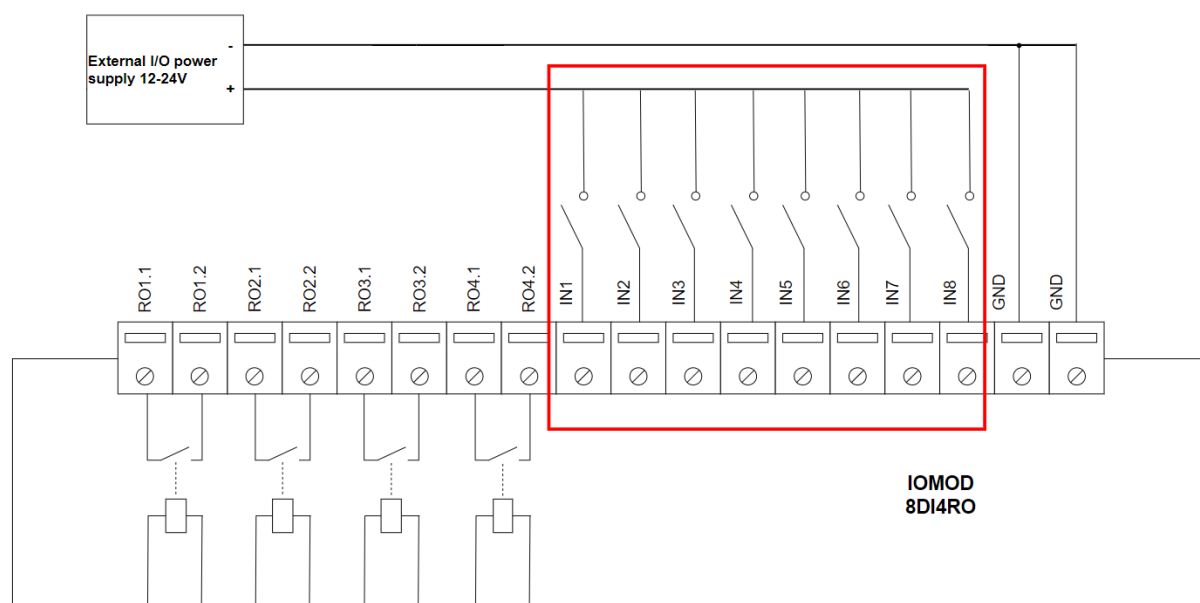


Fig. 4.2.1. IOMod 8DI4RO digital inputs

Users also can configure to enable software input inversion. With this configuration, the user will see inputs connected to 0V (see Fig. 3.) as “high” or state “1”, input status LED will NOT glow.

4.3 Digital relay outputs

IOMod 8DI4RO has 4 relay outputs. Internal clamp diodes are connected to each output, making IOMOD 8DI4RO ideal for driving inductive loads like relays. Maximum 5A per output is allowed. For higher loads outputs can be connected in parallel. Make sure your power supply can provide enough power. The typical application of outputs is shown in Fig. 4.3.1

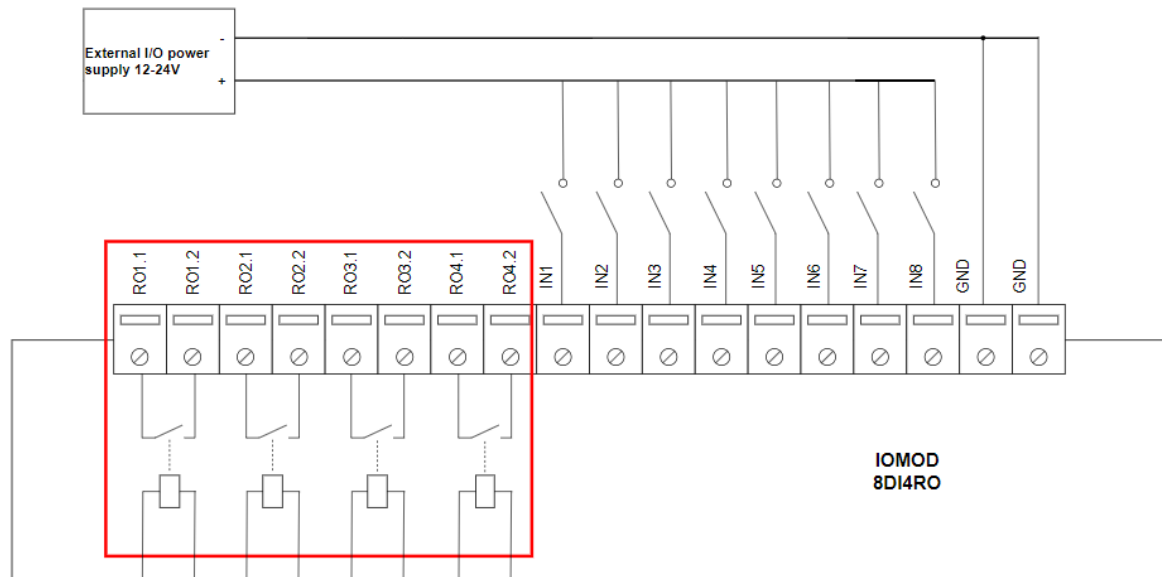


Fig. 4.3.1. IOMod 8DI4RO digital relay outputs

4.4 RS485 Interface

IOMod 8DI4RO has an integrated 120 Ω termination resistor, which can be enabled or disabled via the configuration terminal. It is recommended to use termination at each end of the RS485 cable. See the typical connection diagram in Fig. 4.4.1.

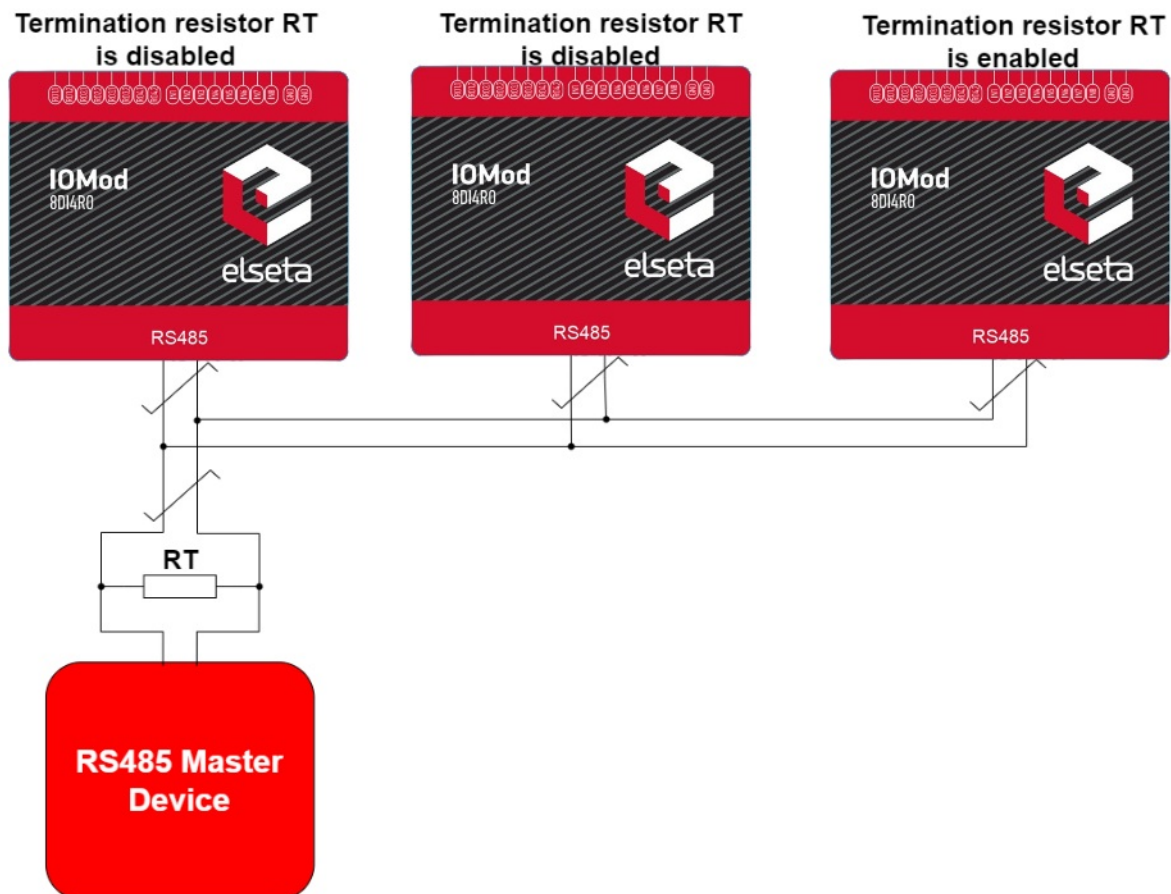


Fig. 4.4.1. Typical IOMod connection diagram

IOMod 8DI4RO has a 1/8 Unit load receiver which allows it to have up to 256 units on the line (compared to standard 32 units). To reduce reflections, keep the stubs (cable distance from the main RS485 bus line) as short as possible when connecting the device.

4.5 Power supply



Fig. 4.5.1. Power supply inputs physical location

4.6 USB connection

The IOMod 8DI4RO device features a USB-mini connection port, which primarily facilitates a physical connection between the IOMod and a PC. By selecting the USB interface and the appropriate communication port in the **IOMod Utility** (Fig. 4.6.1), users can establish a connection to control the device's parameters and monitor its measurements. Additionally, this connection can serve as a power source for the module.

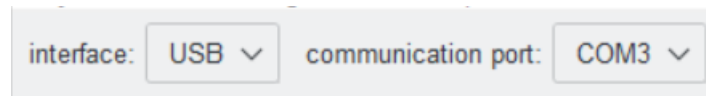


Fig. 4.6.1. IOMod Utility interface and communication port parameters



5. Parametrization

In this section, the IOMOD 8DI4RO settings configuration is described. IOMOD 8DI4RO configuration is performed via IOMod Utility (the manual can be accessed [here](#)). All IOMOD-related settings can be found in the "Iomod settings" tab (Fig. 5.1).

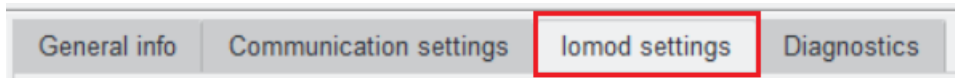


Fig. 5.1. IOMod settings tab

5.1 Iomod Settings

To configure IOMOD 4RTD general settings open the "Iomod settings" tab in IOMOD Utility (Fig. 5.1.1).

| | | | |
|--------------------------|---|--------------------------|---|
| Group inputs | 1+2 <input type="checkbox"/> 3+4 <input type="checkbox"/> 5+6 <input type="checkbox"/> 7+8 <input type="checkbox"/> | Group outputs | 1+2 <input type="checkbox"/> 3+4 <input type="checkbox"/> |
| Swap grouped inputs | 1+2 <input type="checkbox"/> 3+4 <input type="checkbox"/> 5+6 <input type="checkbox"/> 7+8 <input type="checkbox"/> | Swap grouped outputs | 1+2 <input type="checkbox"/> 3+4 <input type="checkbox"/> |
| Invert inputs | 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/> 8 <input type="checkbox"/> | Invert outputs | 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> |
| Input 1 filter, ms | <input type="text" value="50"/> | Input 5 filter, ms | <input type="text" value="50"/> |
| Input 2 filter, ms | <input type="text" value="50"/> | Input 6 filter, ms | <input type="text" value="50"/> |
| Input 3 filter, ms | <input type="text" value="50"/> | Input 7 filter, ms | <input type="text" value="50"/> |
| Input 4 filter, ms | <input type="text" value="50"/> | Input 8 filter, ms | <input type="text" value="50"/> |
| Output 1 short pulse, ms | <input type="text" value="1000"/> | Output 3 short pulse, ms | <input type="text" value="1000"/> |
| Output 2 short pulse, ms | <input type="text" value="1000"/> | Output 4 short pulse, ms | <input type="text" value="1000"/> |
| Output 1 long pulse, ms | <input type="text" value="5000"/> | Output 3 long pulse, ms | <input type="text" value="5000"/> |
| Output 2 long pulse, ms | <input type="text" value="5000"/> | Output 4 long pulse, ms | <input type="text" value="5000"/> |
| Output commands as SBO | | | 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> |
| Output 1 SBO time, ms | <input type="text" value="20000"/> | Output 3 SBO time, ms | <input type="text" value="20000"/> |
| Output 2 SBO time, ms | <input type="text" value="20000"/> | Output 4 SBO time, ms | <input type="text" value="20000"/> |

Fig. 5.1.1 IOMOD Utility with IOMOD 8DI4RO Iomod settings window opened

IOMOD Utility is a tool created to configure IOMODs with firmware version 2. This tool allows users to connect, configure, and diagnose IOMODs such as 8DI8DO, 8DI4RO, 16DI, 8AI, 4RTD, 4CS4VS, METER, and FPI. The Utility's interface allows users to connect to IOMOD via USB port, RS485, and ser2net. More information about this tool and its installation can be found on the documentation page IOMOD Utility.

To configure IOMOD 8DI4RO using IOMOD Utility, first connect to a device or create a template as explained in the IOMOD Utility documentation. Parameters for IOMOD 8DI4RO can be configured on the *Iomod settings* tab. IOMOD with default settings is configured as an IEC101 slave device. Default IOMOD parameters can be seen in the picture above.

Sometimes, two inputs or outputs need to be combined into double point input (DPI) or double command output (DCO). Inputs and outputs can be grouped in pairs of two, where only adjacent pins can be paired, and the first pin in the pair must have an odd number. Once grouped, the second pin in the pair is no longer used, and any requests for it will result in an error. For example, RO1.1-2 and RO2.1-2 can be grouped, but after grouping, RO2.1-2 is disabled. RO2.1-2

and RO3.1-2 cannot be grouped, but RO3.1-2 and RO4.1-2 can be grouped, disabling RO4.1-2.

If desired, input/output groups can be swapped. *Grouped Input/Output Swapping* allows you to exchange the positions of the grouped pins. After swapping, the even-numbered pin can become the first pin in the pair, and the odd-numbered pin follows. For instance, if RO1.1-2 and RO2.1-2 are grouped, you can swap their positions so that RO2.1-2 becomes the first in the pair. Parameters *Group inputs/Group outputs* and *Swap grouped inputs/Swap grouped outputs* are only available for protocols IEC101 and IEC103.

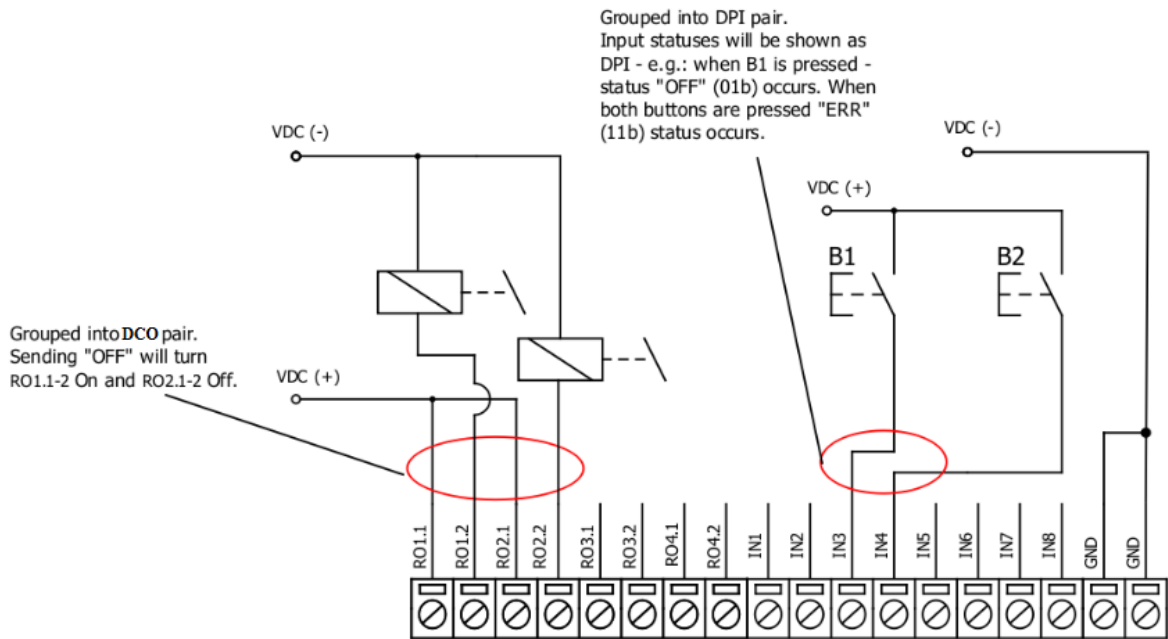


Fig. 5.1.2. Input/Output grouping example

Invert inputs and *Invert outputs* parameters revert the state of the input/output. If the user wants the output status to display as 'ON' when the output signal is in a low state, the outputs can be logically inverted. Unlike 8DI8DO IOMOD 8DI4RO does not have pull-up resistors. If the user desires to turn the input status on, when that input signal is low, the user then inverts inputs logically. All input indication LEDs stay the same (are not inverted).

Input filters define the time after which the input state is considered changed. The input filter is a simple glitch filter with time input. This filter time corresponds to the stable time that input must achieve before sending a status change. For example, if the input state is OFF and the input filter value is 50 ms, the state changes to ON and stays ON for more than 50 ms, then it will be represented as ON, but if the input state was ON for less than 50 ms it will not change and will stay as OFF. This parameter helps to filter values causing less unneeded data.

Users can configure outputs to be pulse controlled – it means that output will be turned on for the configured amount of time. When this time runs out, output is turned off. This is useful when pulse toggle relays are used. The output pulse is independent of the output grouping option and can be used on both grouped and ungrouped outputs. When output is grouped, the device will allow only one command completion at a time – when output is already turned ON, other “turn ON” requests will be responded to with NACK. If the user desires latching outputs to be used, the output pulse time is set to 0.

The IEC101 protocol has settings for long and short pulse commands. Unlike the Modbus RTU protocol, the IEC-60870-5-101 protocol offers options for short and long pulses. A short pulse is typically sent with qualifier 1 (QU/QL = 1), while a long pulse is usually sent with qualifier 2 (QU/QL = 2). The IEC103 and Modbus protocols only have pulse parameters without the ability to configure short or long pulses. The default pulse time for Modbus and IEC103 is the same as the short pulse in IEC101 (1000 ms).

Select Before Operate (SBO) ensures that an output action (such as turning a relay on or off) is carried out only after the output has been explicitly selected. This helps prevent accidental or unauthorized operations, ensuring that the operator's intent is confirmed before acting. For example, the user wants to turn on RO1. First, they issue a select command to RO1. After the selection is acknowledged, the user sends an operation command to turn on RO1. If the command is successful, the IOMod changes the state of the relay, and the output is turned on. If there's an issue, a **negative acknowledgement (NACK)** will be sent, and no change will occur. There is also an output SBO time option that ensures that a relay operation is not immediately executed but is instead delayed to allow for confirmation or review before the operation is carried out.

The Iomod settings' available values and ranges can be seen in the table below (Table 5.1.1).

Table 5.1.1 IOMOD 8DI4RO parameter ranges and default values

| Parameter | Range | Default value |
|----------------------|---------|---------------|
| Input [] filter, ms | 1-65535 | 50 |

| | | |
|-----------------------------|---------|-------|
| Output [] short pulse, ms* | 1-65535 | 1000 |
| Output [] long pulse, ms* | 1-65535 | 5000 |
| Output [] SBO time, ms* | 1-65535 | 20000 |

* The parameters are defined only for the IEC 60870-5-101 communication protocol.

5.2 Diagnostics

The Utility diagnostics windows allow users to connect to IOMOD directly and observe the values in real-time. The 8DI4RO diagnostics window shows input and output states and also allows activation of the outputs. If an input or output is on, the Diagnostics window indicates it with a blue square next to it (unless the input/output is grouped, swapped or inverted, then they would behave in a configured way).

To turn on real-time monitoring of both Diagnostics sections, the "Connect" button to the left of the "Offline" word designation needs to be pressed. After pressing the "Connect" button the word designation of Diagnostics mode changes to "Online", the black circle starts blinking and the button name changes to "Disconnect" (Fig. 5.2.1). When a fault is detected the checkboxes are going to be checked.

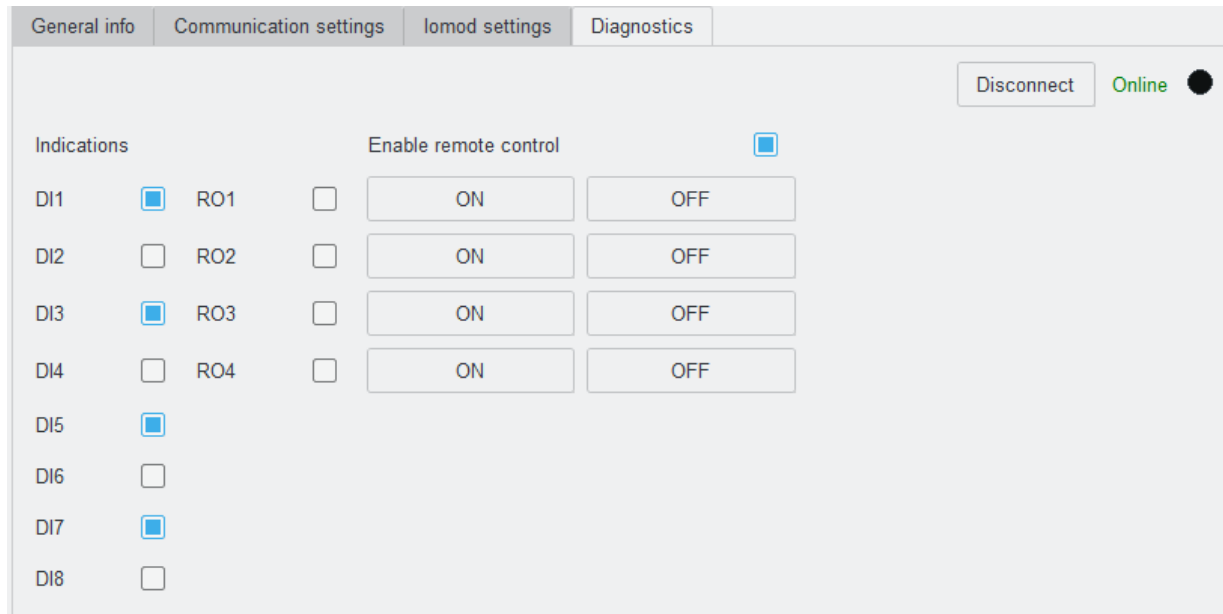


Fig. 5.2.1. IOMOD Utility Diagnostics tab in online mode

6. Communication protocols

IOMOD 8DI4RO uses Modbus (RTU), IEC-60870-103 or IEC-60870-101 protocols over Ser2Net or RS485 connection, which can be used for cable lengths up to 1500 meters and connect up to 30 devices on one line. Default Modbus, IEC-60870-103 and IEC-60870-101 settings are: 19200 bauds/s baudrate, 8E1, Slave (Link) address - 1.

6.1 Modbus RTU Operational information

To read the status of relay outputs (On or Off), Modbus function 01 is used. The IOMOD 8DI4RO has 4 relay outputs, with addresses ranging from 0 to 3.

Modbus function 02 reads the status of digital inputs (On or Off). The IOMOD 8DI4RO has 8 digital inputs, with addresses ranging from 0 to 7. These inputs are active-high by default.

Function 03 allows the user to read counter/timer values related to digital inputs. There are 40 Modbus registers. The values in these registers are described in the table below. Two types of values are stored: Pulse Counter and On Timer. The On Timer calculates the duration for which a respective input remains in its active state.

Function 04 allows the user to read counter/timer values related to digital inputs. There are 80 Modbus registers. The values in these registers are described in the table below. Two types of values are stored: Pulse Counter and On Timer, with the On Timer calculating the time (in seconds) for which a respective input remained in its active state. This function is deprecated and is maintained for backward compatibility with earlier versions of IOMOD 16DI.

To set the state of a single relay output (On or Off), Modbus function 05 is used. The output addresses range from 0 to 3 (the first output is address 0, last output is address 3).

Function 06 sets the value of a single register. The register addresses are identical to those used in the "Read Input Registers" function.

Modbus function 15 sets the state of multiple relay outputs (On or Off) simultaneously. The output addresses range from 0 to 3 (the first output is address 0, last output is address 3).

Table 6.1.1 Modbus register mapping table

| Register | Description | Value Range |
|---|---|-----------------|
| Read coil status (01) | | |
| 00000-00007 | Reading digital outputs RO1-RO4 | 0-255 |
| Read discrete inputs (02) | | |
| 00000-00007 | Reading digital inputs DI1-DI8 | 0-255 |
| Read holding register (03), Read input register (04), Preset Single Register (06) | | |
| 00000 | Pulse count for DI1, Least Significant Word | 0-65535 |
| 00001-00002 | On-time in seconds for DI1, Least Significant Word first* | 0-65535 |
| ... | ... | ... |
| 00021 | Pulse count for DI8, Least Significant Word | 0-65535 |
| 00022-00023 | On-time in seconds for DI8, Least Significant Word first* | 0-65535 |
| 00024-00039 | Pulse count for DI1-DI8, Least Significant Word first* | 0-65535 |
| Write single coil (05) | | |
| 00000-00003 | Writing a single relay output | 0x0000 / 0xFF00 |
| Write multiple coils (15) | | |
| 00000-00003 | Writing multiple relay outputs RO1-RO4 | 0-255 |

*It is advised to set the most significant word of counter/timer first

6.2 IEC 60870-5-101 Operational Information

IEC 60870-5-101 (IEC101) is a communication protocol designed for telecontrol applications in power systems, enabling communication between a master station and slave devices (e.g., Remote Terminal Units or RTUs). It supports two operational modes: **unbalanced mode**, where only the master can initiate communication, and **balanced mode**, where both the master and slaves can initiate data exchanges.

The IOMod 8DI4RO uses the IEC101 protocol to transmit digital input (DI) status and control relay outputs (RO) in a standardized format. Each signal is assigned an Information Object Address (IOA) and a Type Identifier (TI). The protocol efficiently conveys binary status changes (e.g., on/off state of equipment or alarms) and control commands with associated timestamps, enabling reliable communication in automation systems.

IOMod communicates using the standard IEC-60870-5-101 protocol. The master (controlling station) initiates all actions, while the IOMod device (controlled station) only responds to these requests.

- Initiation:**
The master sends the first message to request the link status (function code = 9). If the link is available, IOMod replies with the link status (function code = 11); otherwise, it does not respond.
- Resetting the Link:**

After receiving the link status, the master sends a "Reset Remote Link" command (function code = 0) to restart communication. IOMod will respond with either:

- **ACK (Acknowledgment)**: Function code = 0 (indicating success).
- **NACK (Negative Acknowledgment)**: Function code = 1 (indicating failure).

If IOMod responds with ACK, initialization is complete, and the master can proceed with other IEC-60870-5-101 protocol messages.

Once initialization is complete, the master can request data from the IOMod device using **ageneral interrogation** command. This command retrieves the current status of all inputs and outputs.

In addition, the IOMod automatically sends updates whenever a value changes, as specified by the IEC101 protocol. For example, the 8DI4RO IOMod reports status changes for its digital inputs using **data type 1 (M_SP_NA_1)**, which represents single-point information (on/off states).

This ensures that the master always has up-to-date information without needing to continuously poll the IOMod, making communication efficient and reducing network traffic.

The IOMod 8DI4RO uses the IEC-60870-5-101 protocol for managing inputs and outputs.

The outputs are controlled by the master (controlling station) using a **45-type command (C_SC_NA_1, single-point command)**. Each output is assigned an Information Object Address (IOA):

- **101**: RO1
- **102**: RO2
- **103**: RO3
- **104**: RO4

By specifying the corresponding IOA, the master can turn outputs ON or OFF. If an invalid command is sent (e.g., incorrect IOA or data type), the device responds with a **negative acknowledgement (NACK)**.

The inputs are mapped to the following IOAs:

- **9**: DI1
- **10**: DI2
- **11**: DI3
- **12**: DI4
- **13**: DI5
- **14**: DI6
- **15**: DI7
- **16**: DI8

This setup allows the master to monitor input statuses and control outputs efficiently, ensuring seamless communication between devices.

To synchronize time, the master sends a **Clock Sync command**:

- **Command Type**: C_CS_NA_1 (103)
- **Cause of Transmission (COT)**: 6

The command must target the correct link address and CASDU (defaulted to the link address).

If the frame is valid, the IOMod responds with:

- **Command Type**: C_CS_NA_1 (103)
- **COT**: 0x07 (positive confirmation)
- The response includes the device's timestamp.

If the sync is disabled or the CASDU is incorrect, the response will be:

- **COT**: 0x47 (negative confirmation)
- **p/n bit**: 1

To start the General Interrogation (GI), the master sends the **C_IC_NA_1 (100)** command with the **Cause of Transmission (COT)** set to 6. The command must be sent to the correct link address and CASDU (default is the link address).

- **If the command is valid:**
The IOMod responds with the same **C_IC_NA_1 (100)** command, **COT = 0x07** (positive confirmation), and **p/n bit = 0**. The device then begins to send all its data.
- **If the command is invalid:**
The IOMod responds with the same command, but **p/n bit = 1** (negative confirmation) and **COT = 0x47**.
- **End of GI:**
If the GI is successful, the IOMod will send another **C_IC_NA_1 (100)** command with **COT = 0x10 (ActTerm)**, indicating that the interrogation is complete.

5.3 IEC 60870-5-103 Operational Information

IEC 60870-5-103 is a communication protocol specifically designed for telecontrol and automation systems in the electrical power industry, particularly for substation automation. IEC 60870-5-103 is more specialized for applications related to protection, control, and monitoring in electrical substations. It supports two operational modes: **unbalanced mode**, where only the master can initiate communication, and **balanced mode**, where both the master and slaves can initiate data exchanges.

IOMod communicates using the standard IEC-60870-5-103 protocol. The master (controlling station) initiates all actions, while the IOMod device (controlled station) only responds to these requests.

1. **Reset Command:**
 - The master sends a "Reset CU" or "Link Reset FCB" command (function code = 7).
 - IOMod responds with an acknowledgement (ACK, function code = 0).
2. **Requesting Link Status:**
 - The master sends a "Request Status of Link" command (function code = 9).
 - IOMod replies with the link status (function code = 11).
3. **Resetting the Remote Link:**
 - The master sends a "Reset Remote Link" command (function code = 0).
 - IOMod responds with an acknowledgement (ACK, function code = 0).

Once the initialization process is complete, the master can send other IEC-60870-5-103 protocol messages. Any messages sent before completing the initialization process will be ignored by IOMod.

When initialization is complete, the master may poll the IOMOD device with both Class 1 and Class 2 requests. Class 2 is used when master polls for cyclic data. The controlled device answers with a message containing an Access Demand flag when spontaneous data exists and the master then sends a request for Class 1. IOMod would then respond with a time-tagged message.

On the first Class 1 request IOMod device always asks for the Access Demand to send an identification string. However, if there are spontaneous messages to be sent, they will be sent before the identification string.

To enable or disable relay outputs master (controlling station) sends commands conforming to the IEC-60870-5-103 protocols. The function type of the output commands has to be 128 (80h). Info number represents the number of output pins (1-4 accordingly). Info elements show DPI information of output state:

- **0** – intermediate
- **1** – off
- **2** – on
- **3** – not used (defines error)

The successful command is accepted with a positive acknowledgement.

When an input status changes, the IOMod filters out glitches using a user-configurable filter time. Once the filter is passed, the device sends a "**Spontaneous**" message with the Information Number field indicating the input pin address and a default **Function Type** of 160 (A0h). These messages include a 4-byte timestamp without date information, so the controlling station must handle signals properly, especially at the start of a new day.

To synchronize time between devices, the master sends a variable frame with the following parameters:

- **Function Type:** 0 (GLB)
- **Type Identification:** 6
- **Information Number:** 0
- **Cause of Transmission:** 8
- **Info Elements:** 7-byte time structure

According to the IEC-60870-5-103 protocol specification, time synchronization can be performed for multiple devices using broadcast messages. This feature is available in firmware version 1.7.3 or later. To broadcast a time synchronization message, the link address must be set to **255**.

General Interrogation (GI) starts when the master sends a variable frame with:

- **Function Code:** 3
- **Type Identification:** 7
- **Information Number:** 0
- **Cause of Transmission:** 9

The slave device acknowledges (ACK) the request. The master then retrieves IOMod data through Class 2 polling requests. IOMod responds with time-tagged messages containing DPI states of outputs (sent first) and inputs. PINs are represented by the Info Number fields in the packets. Finally, the slave sends an End of GI message:

- **Cause of Transmission:** 10
- **Type Identification:** 8