

# 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). 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 60870-5-103 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.

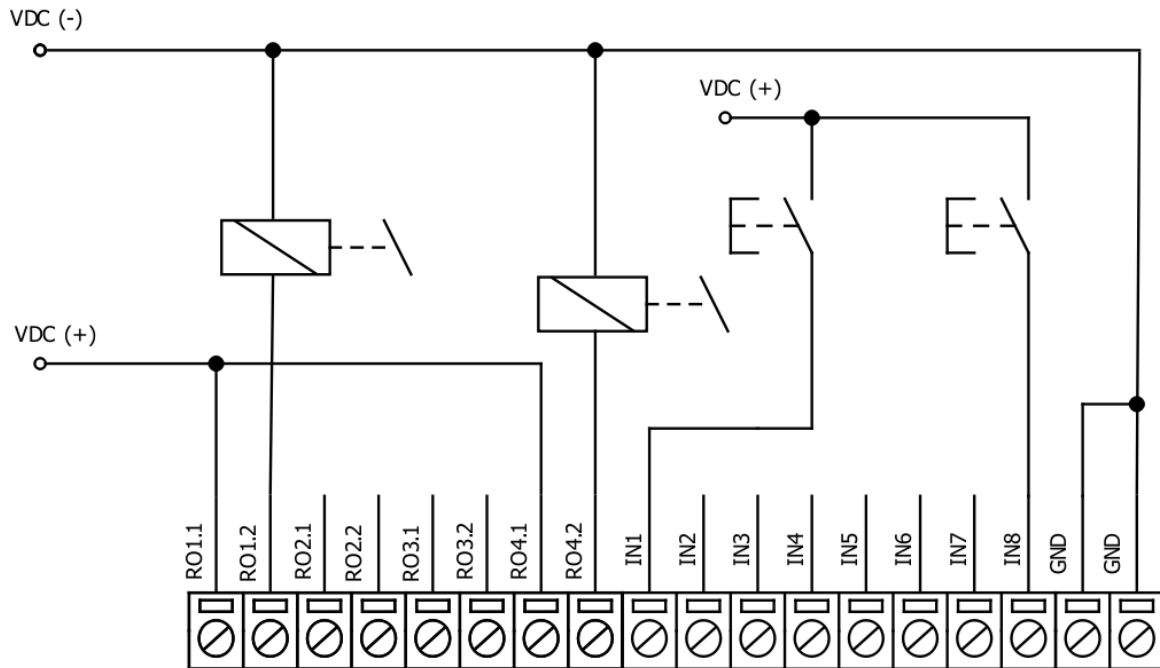


Fig. 1.

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.

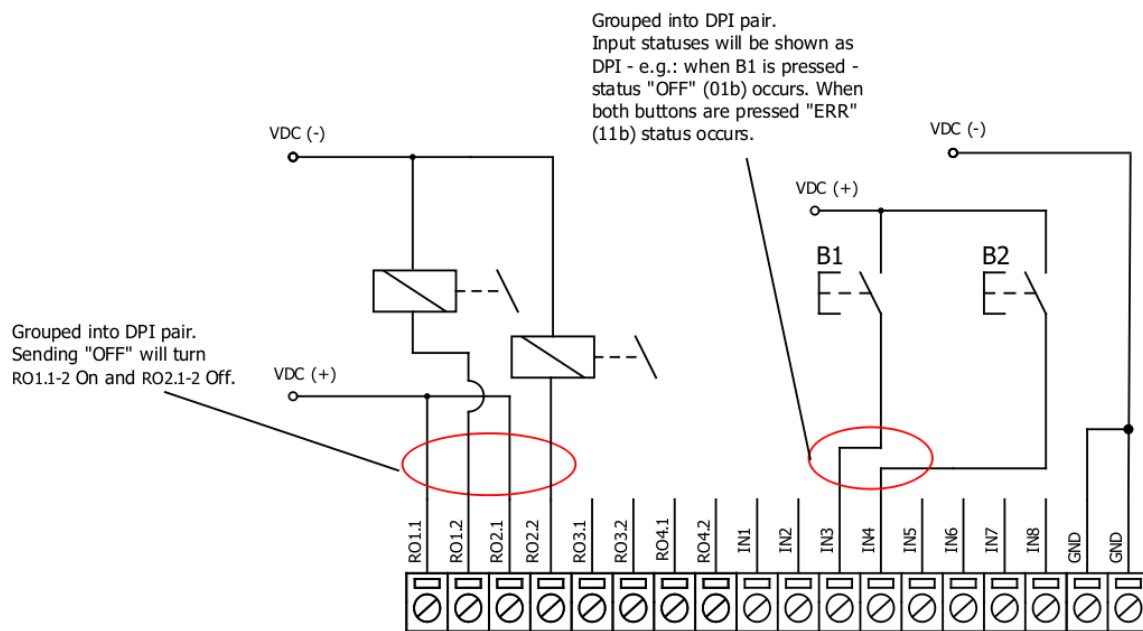


Fig. 2.

## 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.

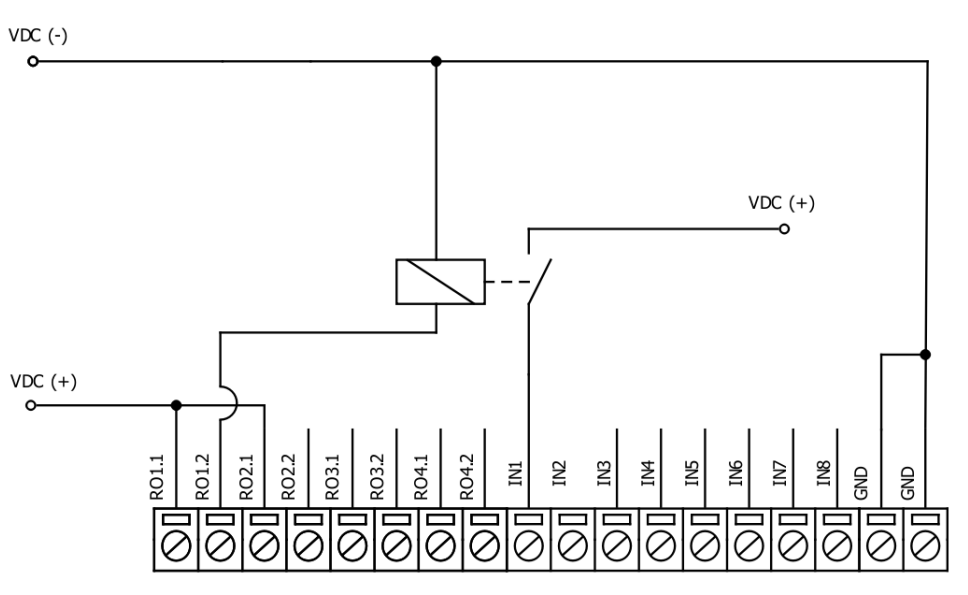


Fig. 4.

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
	<b>Electrical specifications</b>	
	<b>Inputs</b>	
8.	Nominal input voltage range	12-48VDC
9.	Isolation	8 X 3kV <sub>(rms)</sub>

10.	Maximum input voltage	60VDC
	<b>Output relay contacts ratings</b>	
11.	Isolation	4 X 3kV <sub>(rms)</sub>
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
	<b>Power</b>	
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 mention is that as it can be noticed in the scheme below all inputs are isolated. However ground connection is common for all phototransistors.

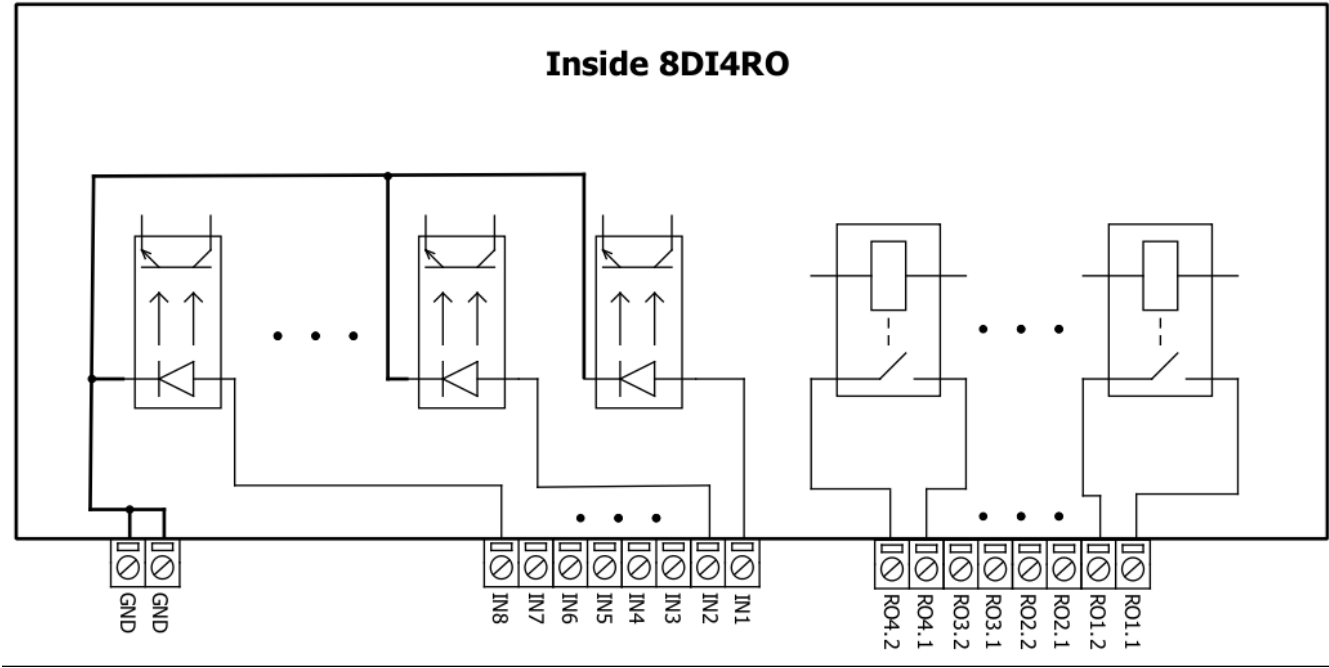


Fig. 5.

# 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.

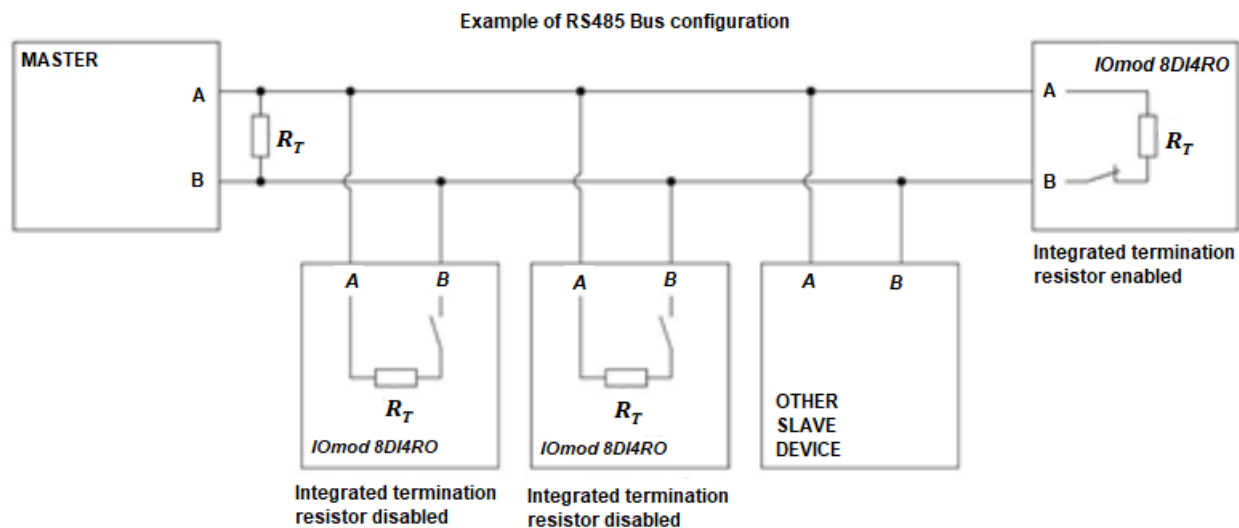


Fig. 6.

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.

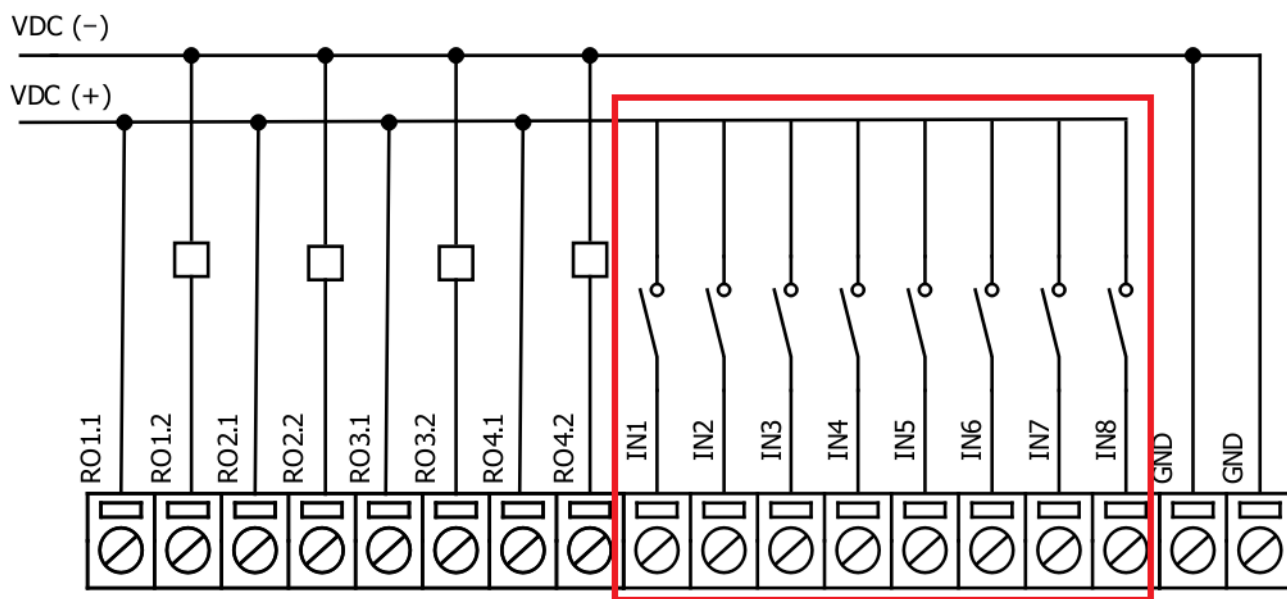


Fig. 7.

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.

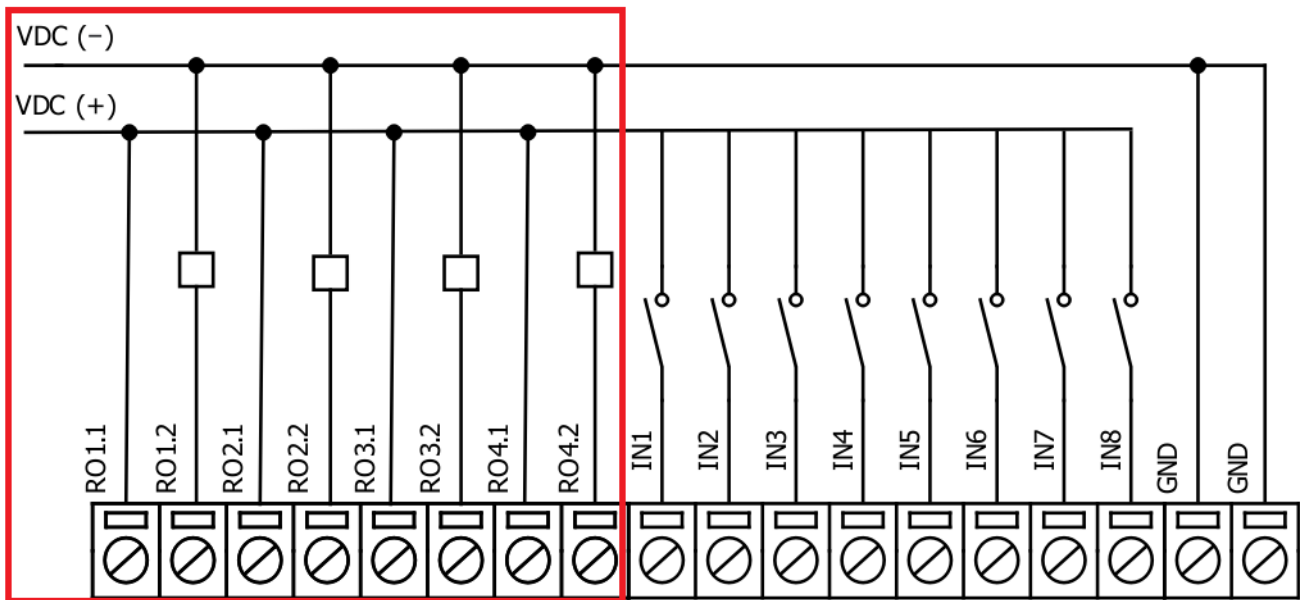


Fig. 8.

# 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.).

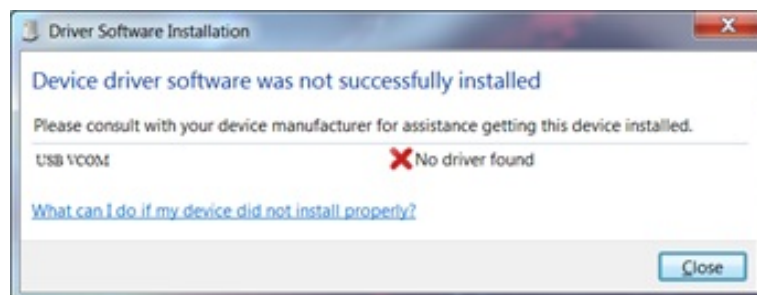


Fig. 10.

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:

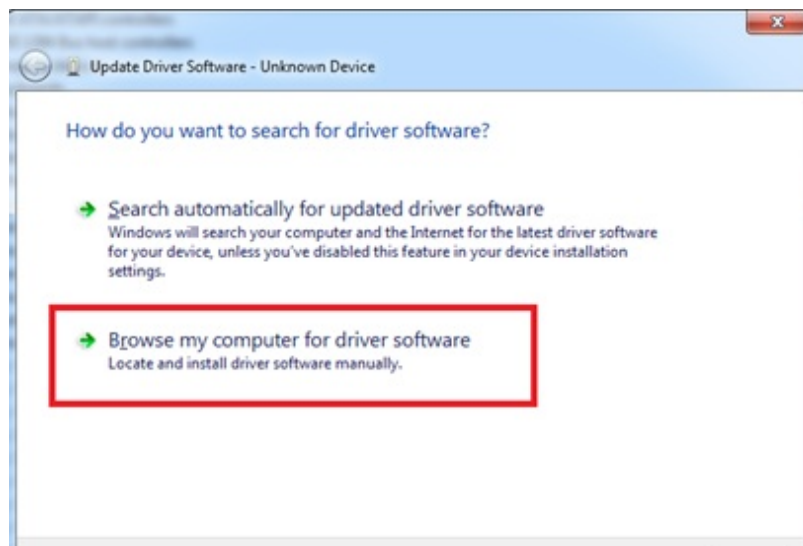


Fig. 11.

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).

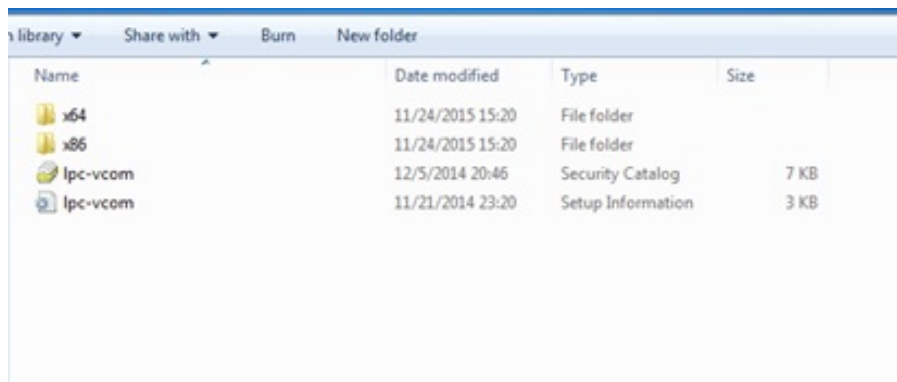


Fig. 12.

## 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.

🕒Revision #26

★Created 3 May 2023 06:22:13 by Andrej

✍Updated 24 January 2025 08:15:04 by Andrej