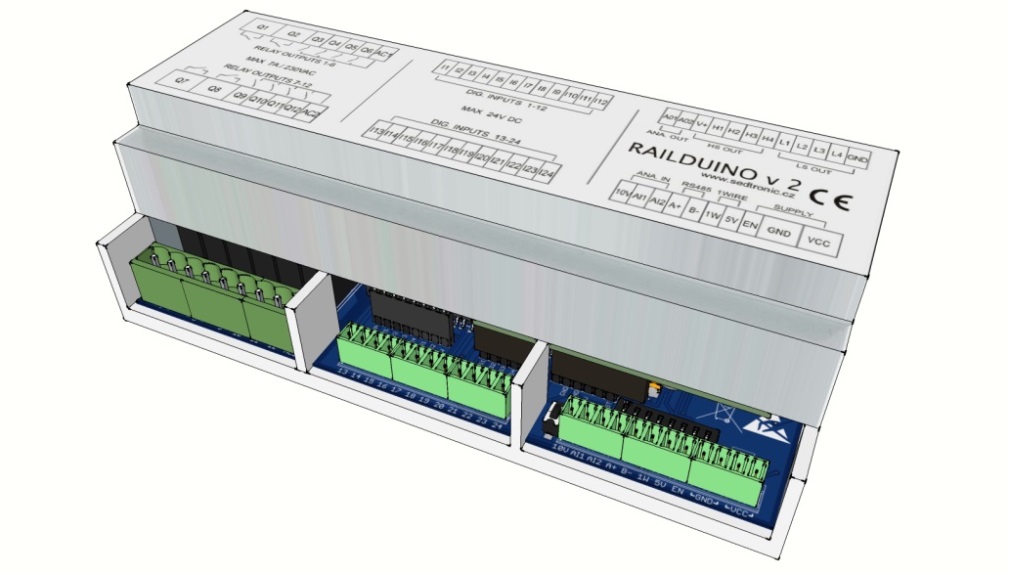
OPERATING INSTRUCTIONS

**RAILDUINO 2.1**

**RS485 LAN WIFI**



**Content**

[1. General instructions and information 2](#_Toc534356948)

[1.1 Symbols used 2](#_Toc534356949)

[1.2 Safety warnings 2](#_Toc534356950)

[1.3 Scope of delivery 2](#_Toc534356951)

[1.4 Delivery and packaging description 2](#_Toc534356952)

[1.5 Storage 2](#_Toc534356953)

[1.6 Installation and commisioning 2](#_Toc534356954)

[1.7 Spare parts 2](#_Toc534356955)

[1.8 Repairs 2](#_Toc534356956)

[1.9 Warranty 2](#_Toc534356957)

[2. Termination of operation and disposal 3](#_Toc534356958)

[2.1 Termination of operation 3](#_Toc534356959)

[2.2 Management and disposal of the packaging 3](#_Toc534356960)

[3. Product description 4](#_Toc534356961)

[3.1. Concept 4](#_Toc534356965)

[3.2. Technical specification 6](#_Toc534356966)

[4. Installation instructions 7](#_Toc534356967)

[4.1 Assembly 7](#_Toc534356968)

[4.2 DIP switch setting 7](#_Toc534356969)

[4.3 Status LED diodes 8](#_Toc534356970)

[4.4 Basic connection between Loxone and Railduino module 8](#_Toc534356971)

[4.5 Communication settings 9](#_Toc534356972)

[4.6 WiFi communication basics – only Railduino Wifi type 9](#_Toc534356973)

[4.7 Description of communication protocol UDP 10](#_Toc534356974)

[4.8 Description of communication protocol MODBUS TCP 11](#_Toc534356975)

[5. Examples of communication between Loxone and Railduino 12](#_Toc534356976)

[5.1 UDP communication settings in Loxone Config 12](#_Toc534356977)

[5.2 Modbus TCP communication settings in Loxone Config 12](#_Toc534356978)

[5.3 Digital inputs – UDP communication 13](#_Toc534356979)

[5.4 Digital inputs – MODBUS TCP communication 14](#_Toc534356980)

[5.5 Relay outputs – UDP communication 15](#_Toc534356981)

[5.6 Relay outputs – MODBUS TCP communication 16](#_Toc534356982)

[4.1 Digital outputs (low side switch) – UDP communication 17](#_Toc534356983)

[4.2 Digital outputs (high side switch) – UDP communication 18](#_Toc534356984)

[4.3 Analog inputs – UDP communication 19](#_Toc534356985)

[4.4 Analog outputs – UDP communication 20](#_Toc534356986)

[4.5 1-wire bus – UDP communication 21](#_Toc534356987)

[5. Other functions and features 23](#_Toc534356988)

[5.1 RS485 Baudrate 23](#_Toc534356989)

[5.2 Ping and Heartbeat functions 23](#_Toc534356990)

[5.3 Function Reset and Railduino module remote power ON/OFF 23](#_Toc534356991)

[7. FAQ 24](#_Toc534356992)

[8. Table of inputs/outputs 25](#_Toc534356993)

[9. Basic parameters and operation conditions 26](#_Toc534356994)

### General instructions and information

### Symbols used

A warning sign, for safe use these instructions must be followed

CE marking, certifies compliance of the product with the legal requirements

The product does not belong to municipal waste and is subject to separate collection

### Safety warnings

##### Warning!

To prevent electrical shock and fire, follow these safety instructions and guidelines. Do not exceed the technical parameters and use the device according to the following description. Read the instructions carefully before putting the device into operation. The device should be installed only by a qualified technician. Use of the device in any other way than the way recommended by the manufacturer may undermine the protection provided by the device. Do not connect the device to the power supply (dangerous voltage) unless it is being installed. Repairs of the module can be carried out only by the manufacturer.

##### Warning!

In an applications with the connection of main voltage of 230V to the output terminals of the device a sufficient distance or insulation from the wires, clamps and enclosing against the surroundings must be provided, due to preservation of protection against electric shock. Behind the front cover of the device there are output terminals, where dangerous voltage can occur.

In the Czech Republic only a qualified person is allowed to install the device (a person min. skilled according to § 5 of Gov.order no. 50/1978 Sb.) after familiarization with these instructions. The device must not be used otherwise than in accordance with these instructions.

To prevent risk of electical injury or fire the maximal operation parameters of the device must not be exceeded, particularly the range of operational temperatures due to heat impact from connected or other technological equipment nearby!

Protect the device from a direct sun light, dust, high temperature, mechanical vibrations and impact, from rain and high moisture. In case of increased ambient temperature above the mentioned limit a ventilation must be ensured.

### Scope of delivery

The product comes with:

- Instructions for the Railduino Module

### Delivery and packaging description

The product is wrapped in a protective electrostatic packaging and placed in a cardboard box. The product must not be exposed to direct rain, vibrations and impacts during the transport.

### Storage

The products are stored in dry non-condensing areas with temperature -40 up to +85 °C.

### Installation and commisioning

The device should be installed only by technicians, who are acquainted with the technical terms, warnings and instructions and who are able to adhere to these instructions. If there are any doubts regarding the right handling with the module, do not hesitate to contact the local distributor or the manufacturer.

Mounting and connection of the device should meet the national legislation governing installations of electric materials, i. e. diameter of conductors, protective fuses and positions. The Railduino Module is intended to be mount on the DIN rail in compliance with the standard EN 60715 or the PR-TS 35 type.

When installing, commissioning, operating and during maintenance pay attention to the instructions mentioned in the chapter 4.

### Spare parts

Each compact part of the product, for which no special procedures or technological operations are necessary when exchanging, can be also ordered as a spare part.

### Repairs

The products are repaired by the manufacturer. For repair the products are sent in a packaging, which ensures shock absorbtion and protects against damage during transportation.

### Warranty

The product is covered by 2 years warranty from the date of delivery specified on the delivery note. The manufacturer warrants the technical and operation parameters of the products to the extent of valid documentation. The warranty period begins from the date of taking over the goods by the buyer or by a carrier. The manufacturer is not liable for defects caused by improper storage, improper external connections, external factors, particularly by quantities of inadmissible sizes, unqualified mounting, incorrect adjustment, improper use or normal wear and tear.

### Termination of operation and disposal

### Termination of operation

During termination of operation, the dismantling and disposal are possible to execute only after the power supply is disconnected.

### Management and disposal of the packaging

If the product Railduino is not further used, or should it be replaced with a new one, it is not disposed with the general household waste. Disposal of this product must be performed in a separate collection. Separate collection allows for recycling and reuse of used products and packaging materials. Reuse of recycled materials helps prevent environmental pollution and reduces the demand for raw materials. When buying new products, stores, local waste disposal or recycling plants will provide information on proper disposal of electronic waste.

To avoid damage of the environment or human health from uncontrolled disposal, we recommend to contact the seller for information on safe disposal of this product.

### Product description

Railduino module is a device working in connection with superior control system (e.g. Loxone) as so called remote inputs/outputs of the control system. With this connection the control system is able to control outputs of the Railduino module (the external equipment e.g. lights, pumps, breakers etc.) or read the values from the inputs of the module (e.g. push buttons, contacts etc.).

Railduino module enables evaluation of the digital sensors / switches through **24x digital inputs,** which are optically separated, the max. input voltage is 24V. Furthermore, it is possible to read status of **2x analogue inputs** through a 8-bit AD converter, the range of analogue values 0-10V. Through **12x relay outputs** it is possible to control devices with max. current of 7A (relays 1,2,7,8) or max 4A (relays 3-6,9-12) at 230V per one output. Other outputs include **4 x digital outputs (High side switch) and 4 x digital outputs (Low side switch),** max. voltage 24V DC, max. switching current 2 A / output. There are another **2x analog ouputs 0-10V.**

Railduino module is equipped with various communication buses, which enable reading of the actual states on the inputs and alternatively controlling of the outputs. It is a serial bus **RS485** (standars protocol Modbus RTU or plain UDP protocol) that makes possible to connect a superior control system, additional Railduino modules or other devices.

It is possible to communicaate via Ethernet, thus the use of **LAN** connectivity (standard Modbus TCP or UDP protocol).

The communication protocols for bus RS485 and LAN are programmed in own plain structure of packets UDP (User defined protocol) or with use of standard protocol Modbus TCP or RTU

Settings of this communication is described in the chapter Settings.

Further the module incorporates a **1-wire** bus (Dallas-Maxim), which enables the connection of favourite temperature sensors DS18B20 or sensors DS2438 (e.g. UNICA 1-wire sensors) to the module.

Railduino module is able to be switched off remotely using the terminal input EN – for more info see the chapter 6.3



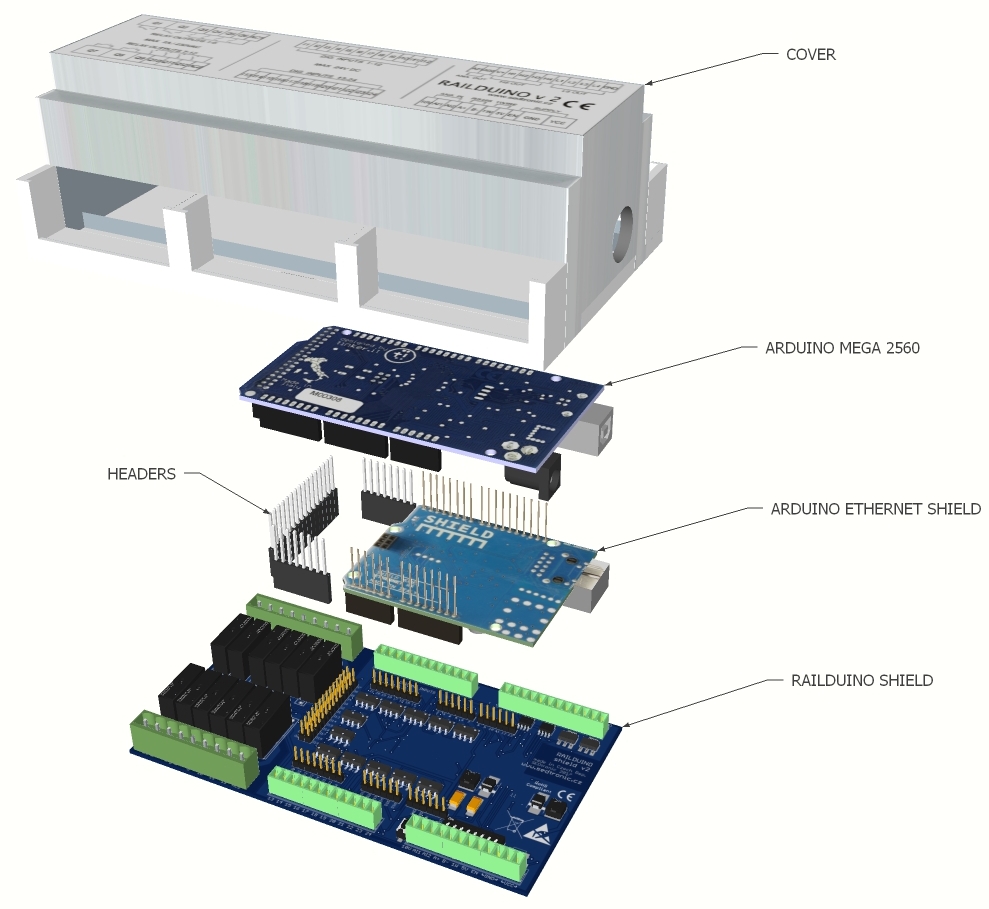
### Concept

The whole module is designed as a system of several parts of printed circuit boards in a DIN type plastic housing (9 modules) that can be mounted on a DIN rail. The Railduino module comprises of the following basic parts:

**Arduino MEGA 2560** - a printed circuit board with a microcontroller in form of an 8-bit processor ATmega 2560 with the frequency of 16 MHz, known as Arduino MEGA 2560 – an open-source project – more can be found on: <http://arduino.cc/>.

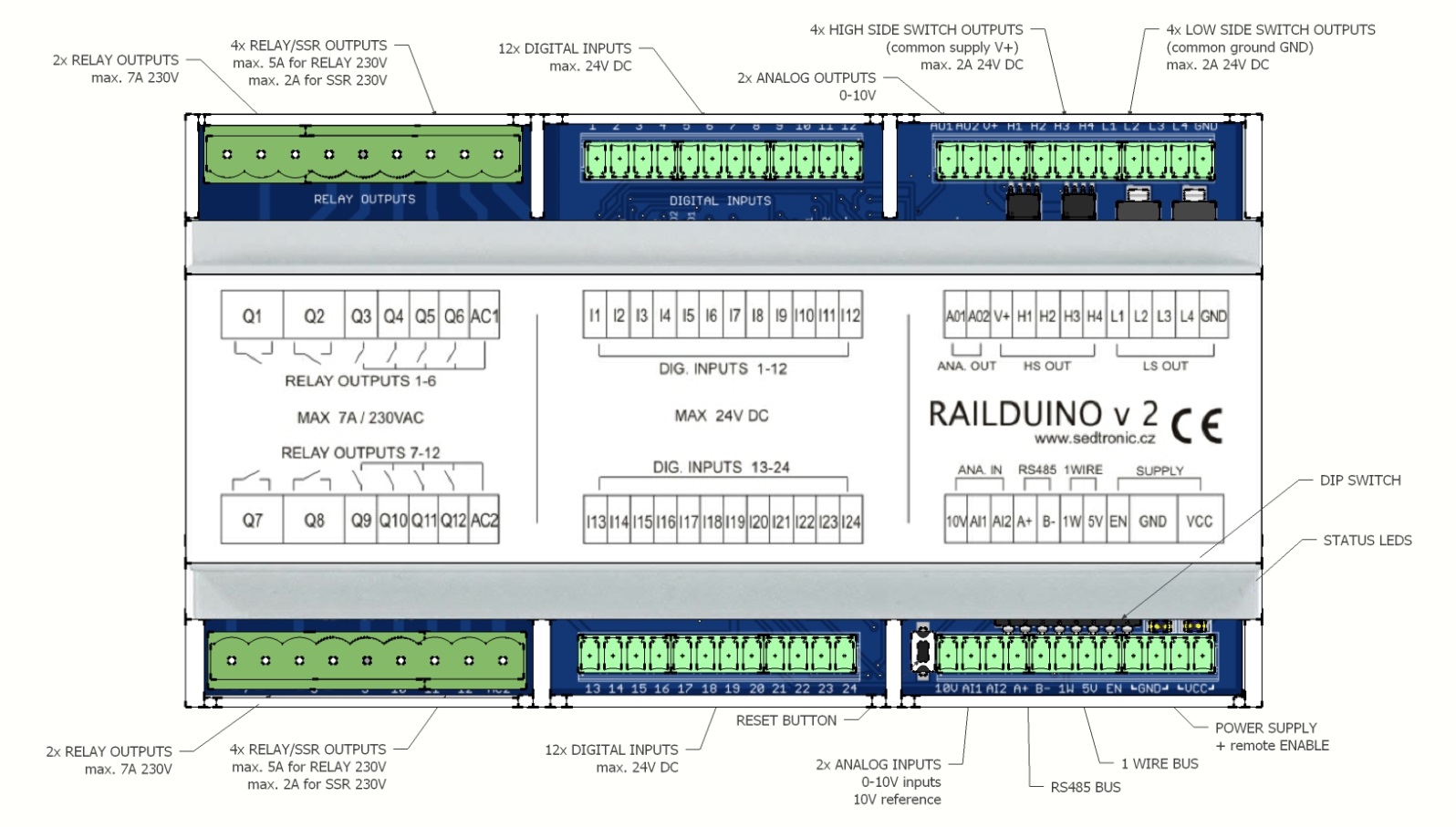
**Railduino shield** is another PCB board enabling connection of all the outside sensors and actuators to the controller, which adjust the signals so that the microcontroller can read or control them. In addition, this board ensures all power supply and communication.

**Ethernet shield** – is an additional (optional) PCB which offers a possibility to connect the whole system to the Ethernet network. When this function is used then the Railduino module is additionally equipped with the headers connectors between shield and Railduino shield.



### Technical specification

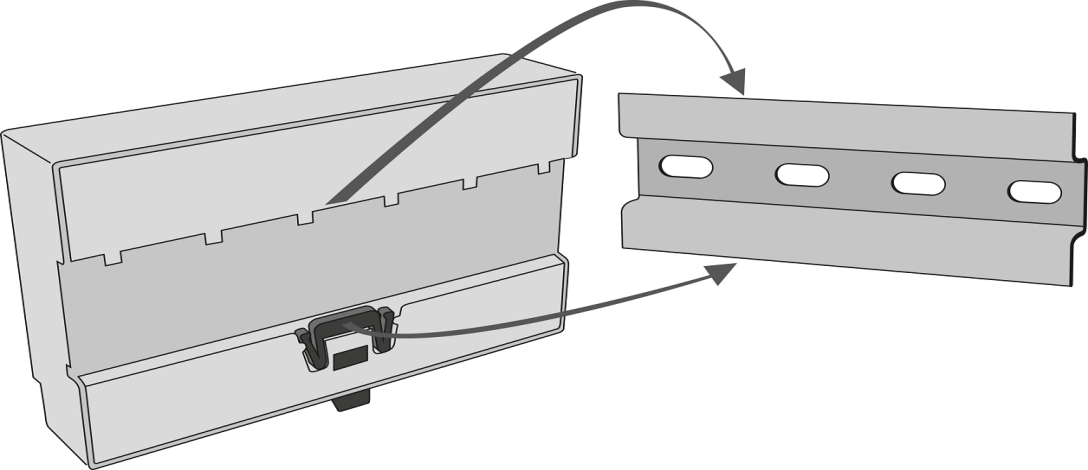
* 24x digital inputs – optically separated, max. input voltage 24V DC
* 12x relay outputs – galvanically separated, max. current 7A / 4A at 230V AC according to desctiption in chap. 5.3
* 2x analog inputs – input voltage 0 - 10V, resolution 8 bits
* 2x analog outputs – output voltage 0 - 10V
* 4x digital outputs – High Side Switch - switching V+ voltage - max. 24V DC, 2A / channel
* 4x digital outputs – Low Side Switch - switching GND - max. 24V DC, 2A / channel
* Communication 1-wire – possibility to connect the sensors Maxim/Dallas DS18B20 or DS2438 (max. 10pcs)
* Communication RS485 – possibility of serial comm. with a master systems (e.g. Loxone RS485 extension)
* Communication USB – possibility to connect to PC for programming purposes – the conector is inside the device
* Communication Ethernet – possibility to connect to LAN e.g. for the Internet (optional)
* Status LED diodes – indication of operation, indication of communication
* Reset button – restart of the module
* Switch DIP – for the address of themodule, additional functions



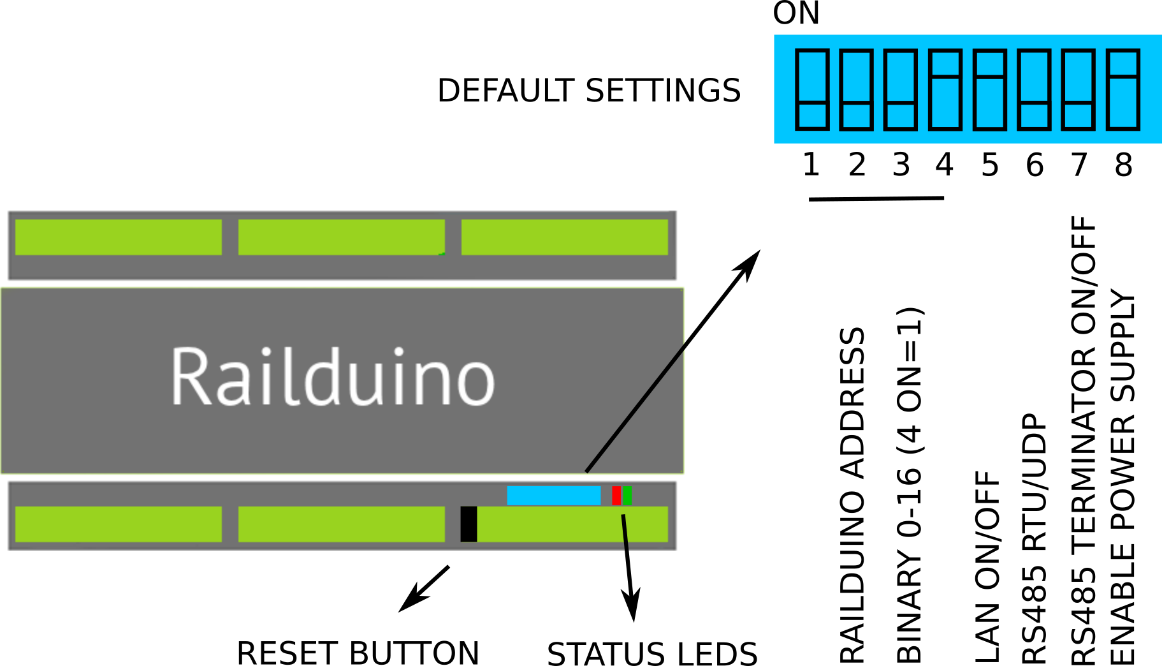
### Installation instructions

### Assembly

Mounting onto DIN rail – click the Railduino Module onto the DIN rail (TS 35) as in the picture.



### DIP switch setting

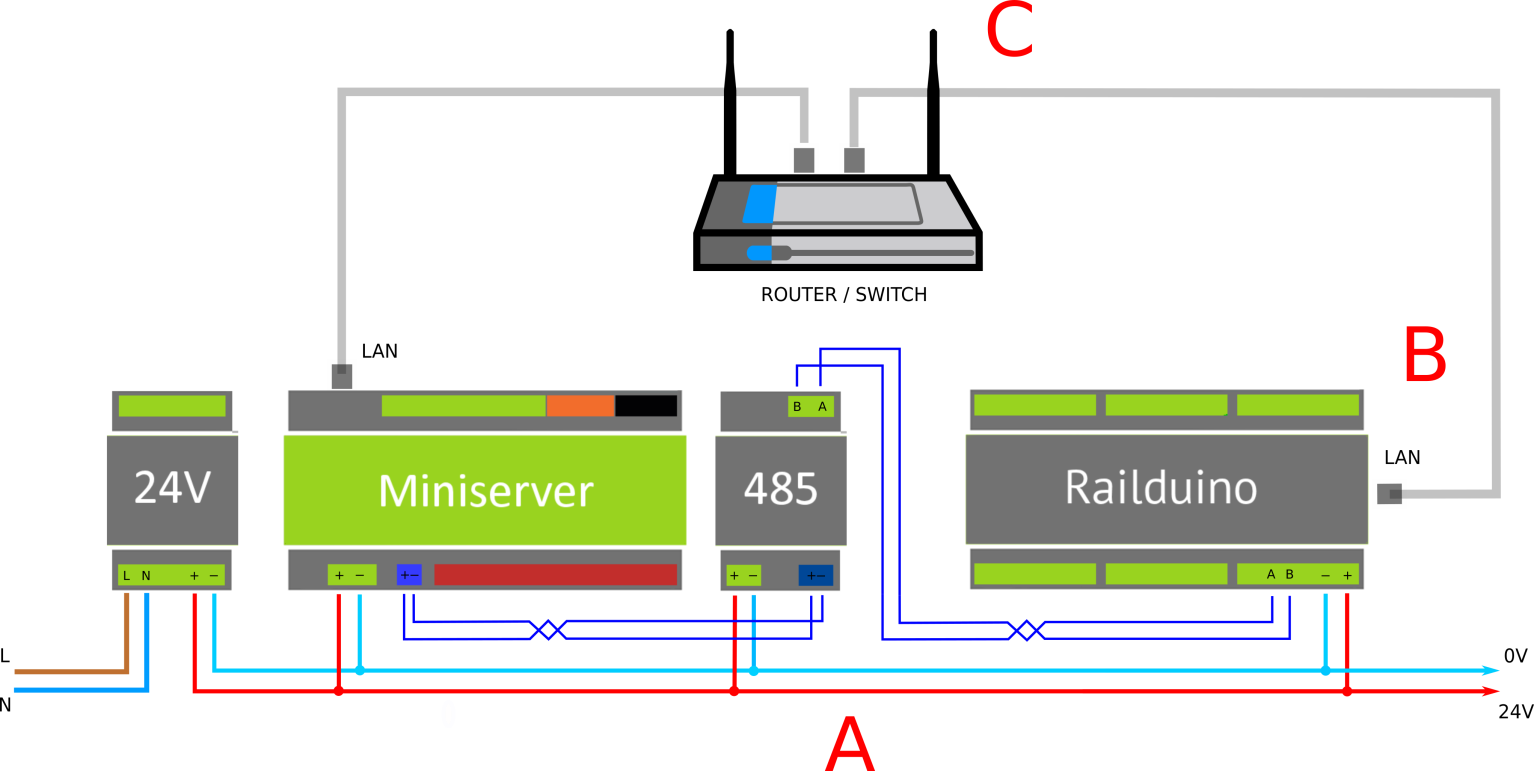


|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PIN | Address | | | | | | | | | | | | | | | |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | x | x | x | x | x | x | x | x | o | o | o | o | o | o | o | o |
| 2 | x | x | x | x | o | o | o | o | x | x | x | x | o | o | o | o |
| 3 | x | x | o | o | x | x | o | o | x | x | o | o | x | x | o | o |
| 4 | x | o | x | o | x | o | x | o | x | o | x | o | x | o | x | o |
| 5 | x | = | LAN OFF | | | | | | o | = | LAN ON (WIFI RESET) | | | | | |
| 6 | x | = | RS485 UDP | | | | | | o | = | RS485 RTU | | | | | |
| 7 | x | = | TERMIN. OFF | | | | | | o | = | TERMIN. ON | | | | | |
| 8 | x | = | ENABLE OFF | | | | | | o | = | ENABLE ON | | | | | |
|  |  |  | x = DIP PIN OFF | | | | | |  |  | o = DIP PIN ON | | | | | |

### Status LED diodes

|  |  |  |  |
| --- | --- | --- | --- |
| LED | Status | Description | Remedy |
| BLUE | do not lit | no power supply or failure | Check the power supply or reset |
| lit | failure | reset |
| flashes 1 Hz | device operating OK | - |
| GREEN | do not lit | no communication in progress | - |
| lit | communication in progress | - |

### Basic connection between Loxone and Railduino module



A – connection to serial bus RS485 – Railduino 2 RS485 type

B – connection to LAN network – Railduino 2 LAN type

C – connection to WiFi network – Railduino 2 WIFI type

There is allways only one option of communication type module – RS485 or LAN or WIFI

In all cases of connections there is used UDP protocol (User defined protocol or Modbus protocol (TCP or RTU) – more in chap. 4.7.

RS485 bus can be terminated with the use of integrated terminator – see DIP switch settings – chap. 4.2

### Communication settings

Default communication bus settings of the Railduino module:

- **RS485** serial bus

* Communication speed 115200 Bd
* Number of data bits 8
* Number of stopbits 1
* Parity none

- **LAN** network

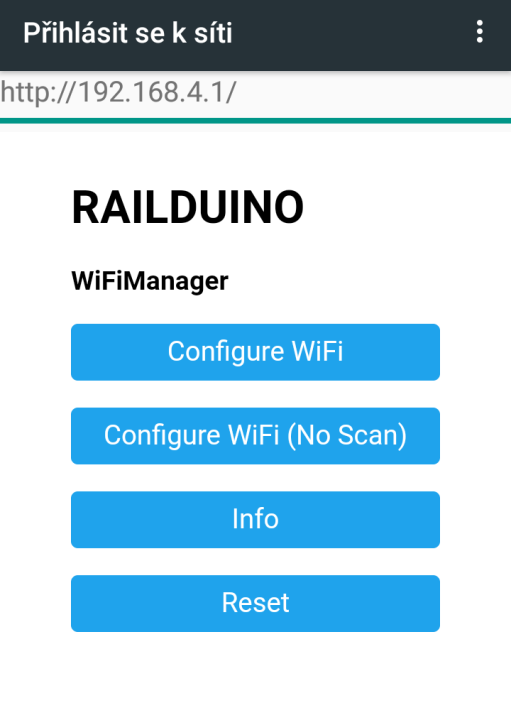
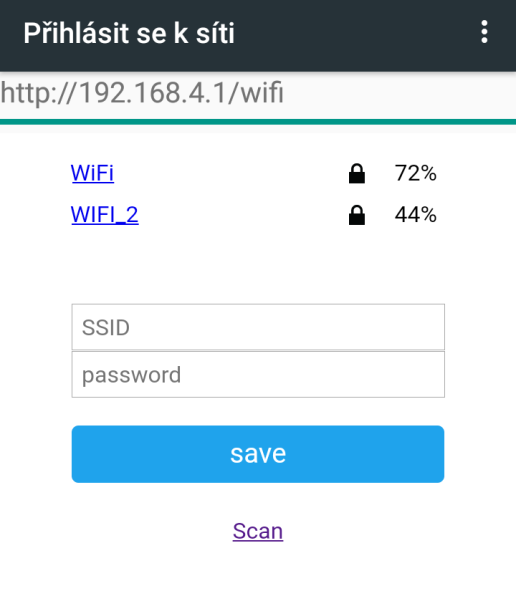
* UDP incoming port 55555
* UDP outgoing port 44444 (in Loxone Config - /dev/udp/255.255.255.255/44444)
* MODBUS TCP port 502
* IP address
  + dynamic – if DHCP server is present
  + static 192.168.150.15x – if DHCP server not present

- **WiFi** network

* UDP incoming port 55555
* UDP outgoing port 44444 (in Loxone Config - /dev/udp/255.255.255.255/44444)
* Default SSID RAILDUINO
  + password railduino
* IP address dynamic – obtained from DHCP

### WiFi communication basics – only Railduino Wifi type

When Railduino module is in default just scan your WiFi network for SSID name „RAILDUINO“ and connect to it with the password „railduino“. You should be redirected to the webpage (192.168.4.1) of the Railduino module where you will configure the WiFi to which Railduino should be connected – see the picture below:



Once the Wifi is successfully connected, the default SSID Railduino is no longer active. To set the default Wifi settings there is need to set the pin no. 5 (LAN ON/OFF) to ON and restart Railduino module.

### Description of communication protocol UDP

The comunnication is based on every IO state change - event based, information about the state is sent when the state of the input changes.

This type of communication is supported at every comm. bus – RS485, LAN, WiFi.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Key word | Railduino address | space | input/output | possible values | space | possible values - °C | space | possible values - V | space | possible vlaues - V |  |  |
|  |  |  |
|  |  |  |
| signal/command | UDP packet example | packet description |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| digital input state | rail | 1-15 |  | di | 1-24 |  | 0,1 |  |  |  |  | rail1 di1 0 | logic zero at dig. input no. 1 at Railduino withs address no. 1 |
| DS18B20 1wire sensor packet | rail | 1-15 |  | 1w | 1w s.n. | | -20-80 |  |  |  |  | rail1 1w 2864fc3008082 25.44 | temp value of 1wire sensor DS18B20 with ser.number 2864fc3008082 |
| DS2438 1wire sensor packet | rail | 1-15 |  | 1w | 1w s.n. | | -20-80 |  | 0-4 |  | 0-0.25 | rail1 1w 2612c3102004f 25.44 1.23 0.12 | values of 1wire sensor DS2438 - temp, Vad, Vcurr, sn. 2612c3102004f |
| analog input state | rail | 1-15 |  | ai | 1-2 |  | 0-255 | |  |  |  | rail1 ai1 127 | value of analog. input no. 1 is 5V (0 = 0V, 255 = 10V) |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| relay on command | rail | 1-15 |  | ro | 1-12 |  | on,off |  |  |  |  | rail1 ro12 on | relay output no. 12 switch on |
| high side switch command | rail | 1-15 |  | ho | 1-4 |  | on,off |  |  |  |  | rail1 ho2 on | dig. output (high side switch) no. 2 switch on |
| low side switch command | rail | 1-15 |  | lo | 1-4 |  | on,off |  |  |  |  | rail1 lo4 off | dig. output (low side switch) no. 4 switch off |
| analog output command | rail | 1-15 |  | ao | 1-2 |  | 0-255 |  |  |  |  | rail1 ao1 255 | analog. output no. 1 value 10V (0 = 0V, 255 = 10V) |
| reset command | rail | 1-15 |  | rst |  |  |  |  |  |  |  | rail1 rst | Reset of railduino no. 1 |

  
  
 In case of use RS485 serial bus communication there must be added symbol **\n** at the end of the **command**

### Description of communication protocol MODBUS

Loxone miniserver is Modbus master and Railduino module acts as a slave module. This protocol can be used only with Railduino LAN or WiFi module option.

At first it is necessary to enter a new communication type ModbusServer to the Loxone configuration and set the communication parameter - IP address for Modbus Server and port.

Further, a new Modbus device is entered into this communication, and the address of this device must by set to 1

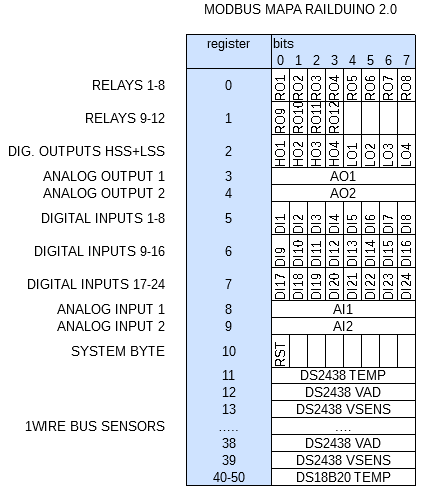
Then, configuration of inputs (sensors) and outputs (actors) can be set:

For reading of **any input** of Railduino Module (digital, analog, 1wire), it is **always necessary** to insert an **analog sensor** in the Loxone configuration (under Modbus device)

For controlling of **any output** of Railduino Module (relay, digital, analog), it is **always necessary** to insert an **analog actor** in the Loxone configuration (under Modbus device) and code its value – see below.

The states of the digital inputs must be read in full bytes from the registers of the Railduino module – please see the table below

Modbus register map (1 register = 1 byte = 8 bits)



More information on the sensors settings in the Loxone configuration is in the example for the Loxone configuration (template file) and in following chapters.

### Following refers only to Railduino 2.1 module:

### Modbus TCP

This type of communication can be used only with **LAN** interface

Supported Modbus functions: **1, 2, 3, 4, 5, 6, 15, 16**

### Modbus RTU

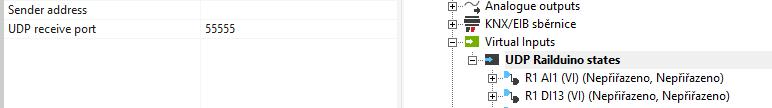
This type of communication can be used only with **RS485** interface

Supported Modbus functions: **3, 6, 16**

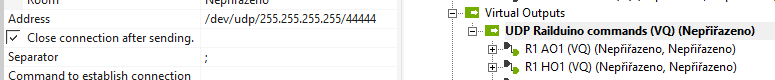
### Examples of communication between Loxone and Railduino

### UDP communication settings in Loxone Config

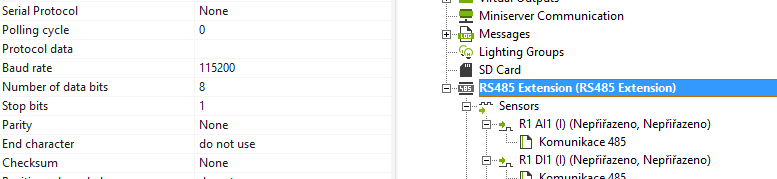
Settings for LAN - UDP communication for sensing inputs in Loxone Config:



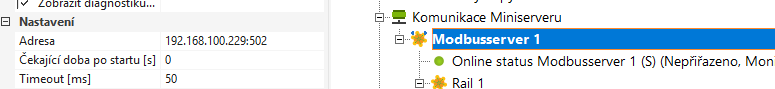
Settings for LAN - UDP communication for controlling outputs in Loxone Config:

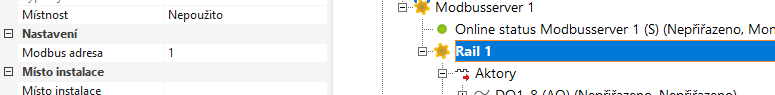


Settings for RS485 - UDP communication in Loxone Config:



### Modbus TCP communication settings in Loxone Config

Settings of Modbus TCP - insert Modbus server and set IP and port in Loxone Config:

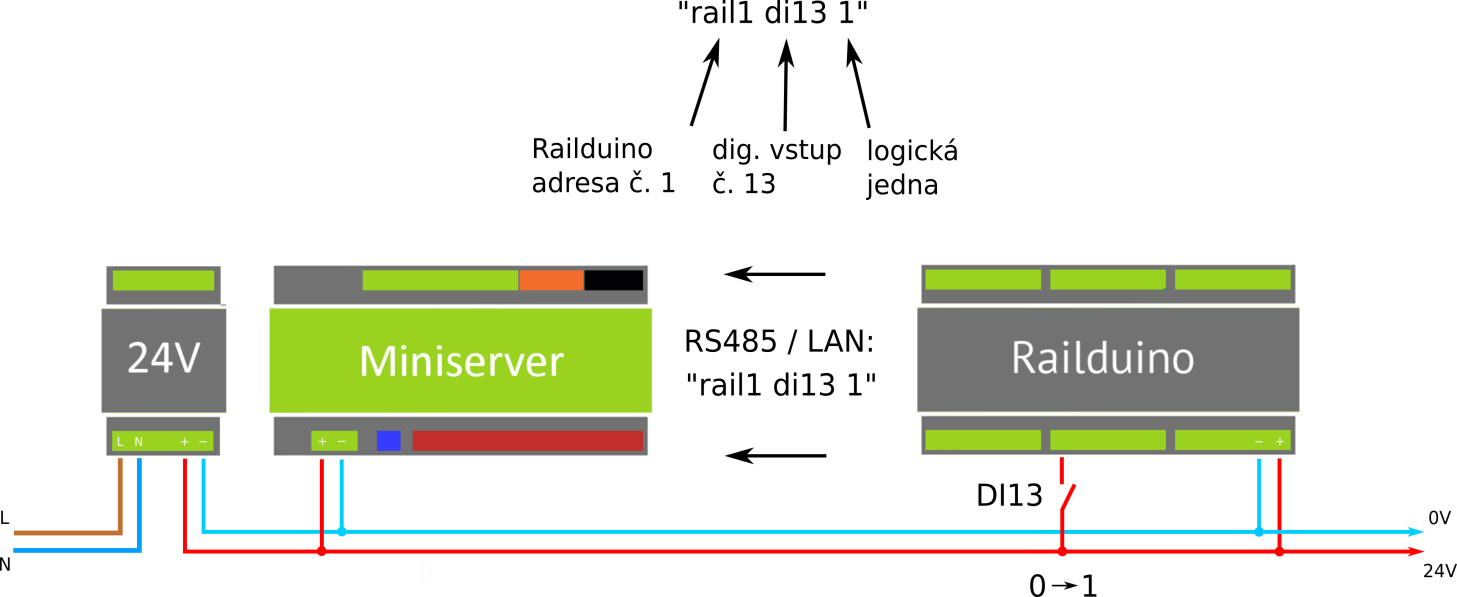
Settings of Modbus TCP - set the address of the Modbus client - Railduino module

### Digital inputs – UDP communication

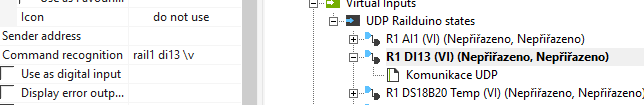
When the change at the digital input is recognized – e.g. push button is pressed, Railduino module sends this information to the control system.



In virtual input command settings it must be **UNchecked** the checkbox Use as digital input !!



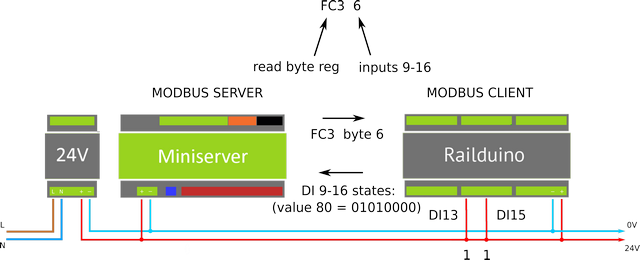
Example of command settings for sensing dig. input no.13 in Loxone Config (LAN connection):



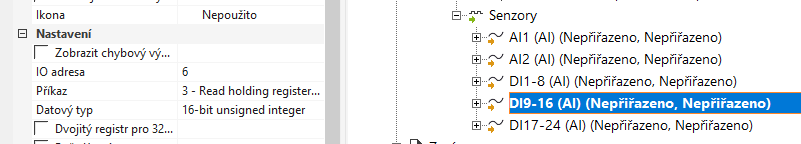
### Digital inputs – MODBUS communication

When the change at the digital input is recognized – e.g. push button is pressed, Railduino module changes the state of the register bytes 5-7. Control system can read the values e.g. with the FC3 command.

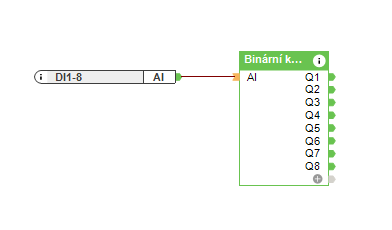
For reading of **any input** of Railduino Module (digital, analog, 1wire), it is **always necessary** to insert an **analog sensor** in the Loxone configuration (under Modbus device) and decode its value – see below.



Example of analog sensor settings for sensing dig. inputs in Loxone Config (LAN/WiFi connection):



Example of using binary decoder block in Loxone Config:



### Relay outputs – UDP communication



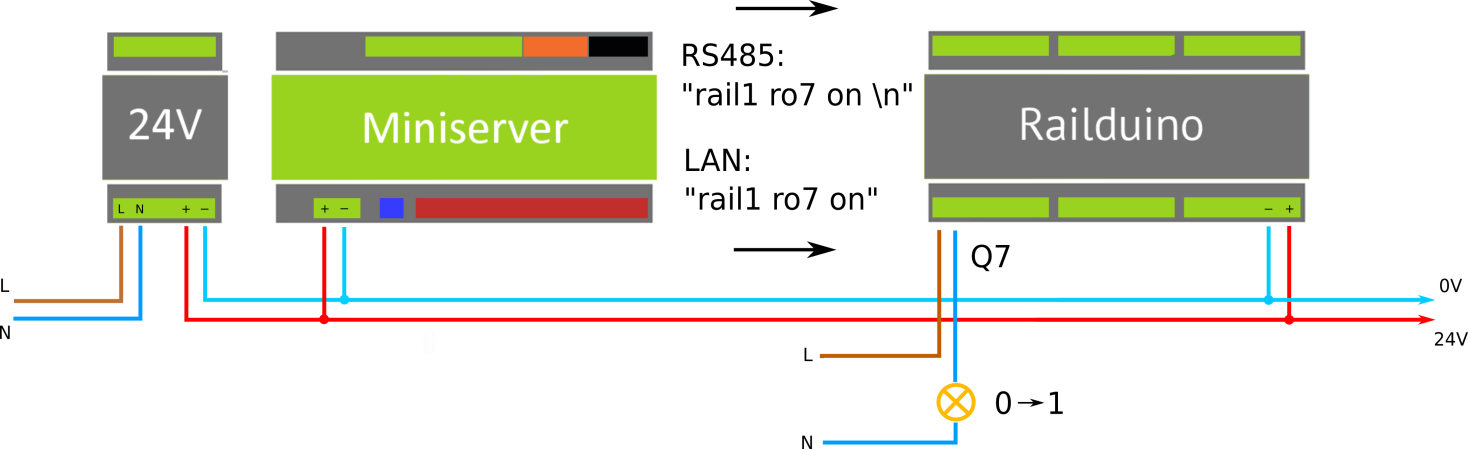
Max. permissible voltage at relay outputs is **230V** AC!

Max. permissible load current at relay outputs no. 1,2,7,8 is **7A** and at outputs 3,4,5,6,9,10,11,12 is **4A** / output!



In case of SSR version of relay outputs the max. permissible load current is limited to **2A** at output and **25 °C** !

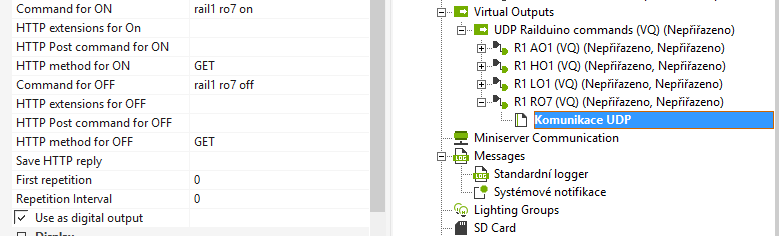
Used SSR relays are equipped with zero cross function – switching at zero voltage level





In virtual output command settings it must be **CHecked** the checkbox Use as digital output !!

Example of command settings for controlling relay output no. **7** in Loxone Config (LAN connection):



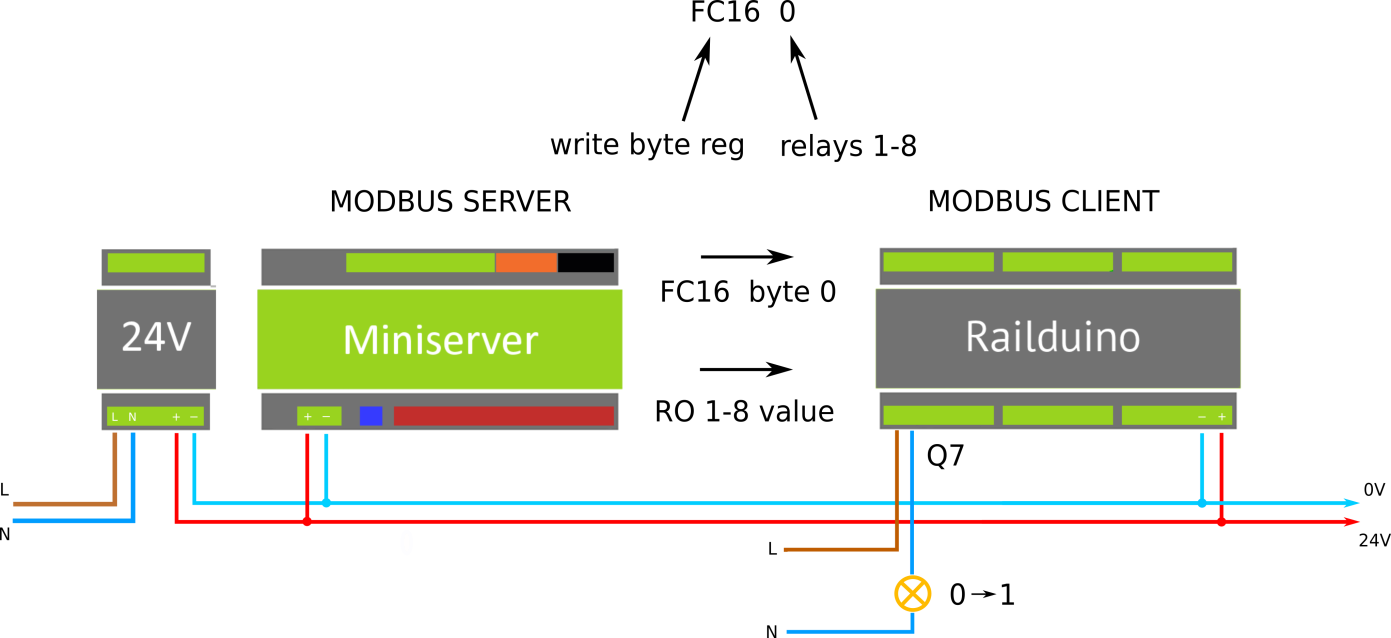
### Relay outputs – MODBUS communication



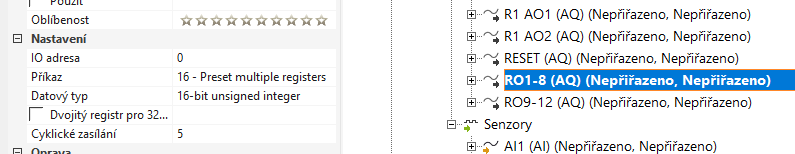
Max. permissible voltage at relay outputs is **230V** AC!

Max. permissible load current at relay outputs no. 1,2,7,8 is **7A** and at outputs 3,4,5,6,9,10,11,12 is **4A** / output!

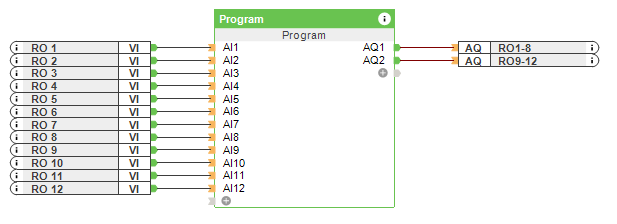
For controlling of **any output** of Railduino Module (relay, digital, analog), it is **always necessary** to insert an **analog actor** in the Loxone configuration (under Modbus device) and code its value – see below.



Example of actor settings for controlling relay outputs in Loxone Config (LAN/WiFi connection):



Example of Loxone program for block Program for controlling relays output using Modbus TCP:



int i, sum, bit;

while(TRUE){

sum = 0;bit = 1;

for (i = 0; i < 8; i++) {

if (getinput(i) == TRUE) {sum += bit;}

bit = bit\*2;

}

setoutput(0,sum);

sum = 0; bit = 1;

for (i = 8; i < 12; i++) {

if (getinput(i) == TRUE) {sum += bit;}

bit = bit\*2;

}

setoutput(1,sum);

}

### Digital outputs (low side switch) – UDP communication

Low side switch is dig. output which connects the ground to the load = common GND is the same as Railduino’s 0V.

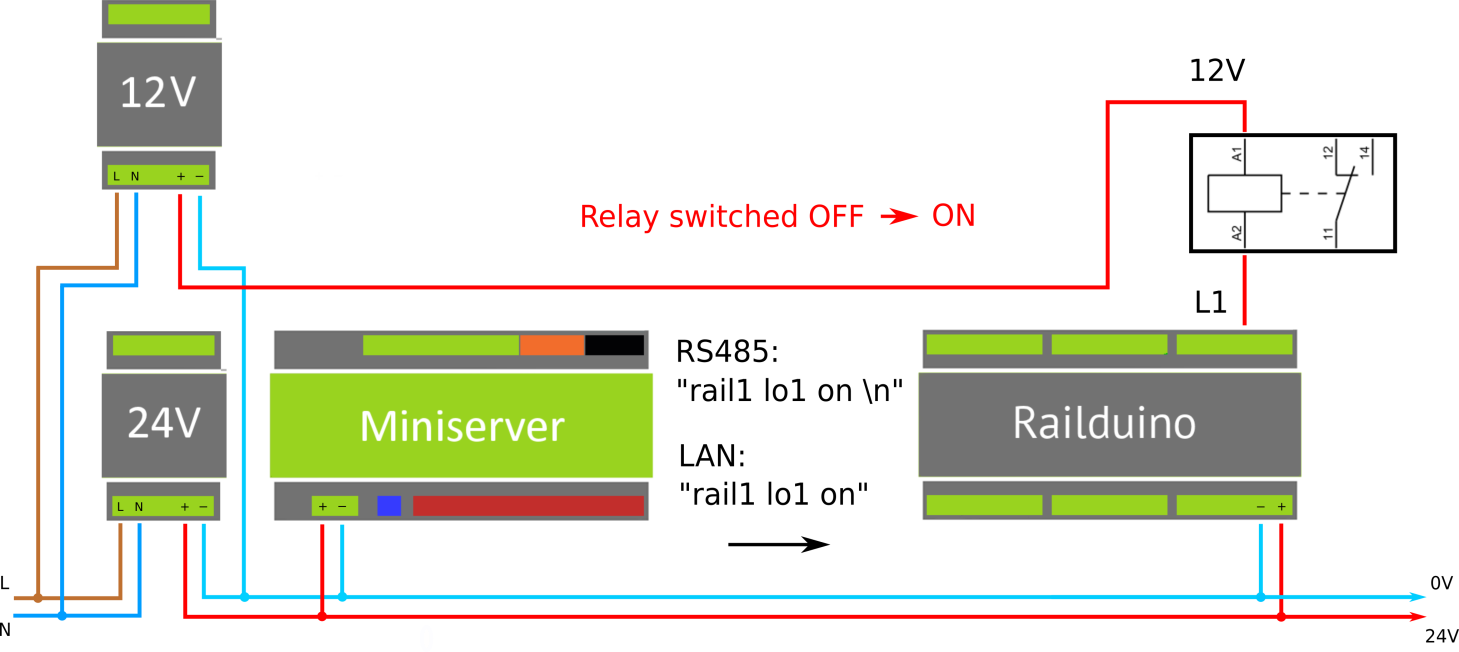
Terminal GND is internally connected to 0V.



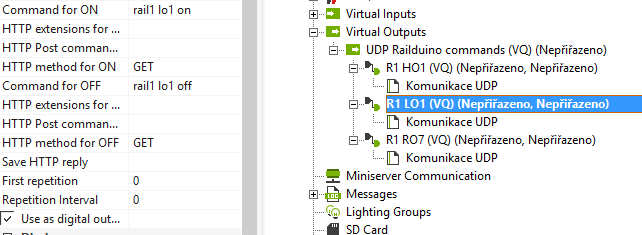
When using two power supplies with different output voltage level the output grounds must be connected!!



Max. permissible voltage at dig. outputs is **24V** DC! Max. permissible load current is **2A** per output!



Example of command settings for controlling digital output (lss) no. **1** in Loxone Config (LAN connection):



### Digital outputs (high side switch) – UDP communication

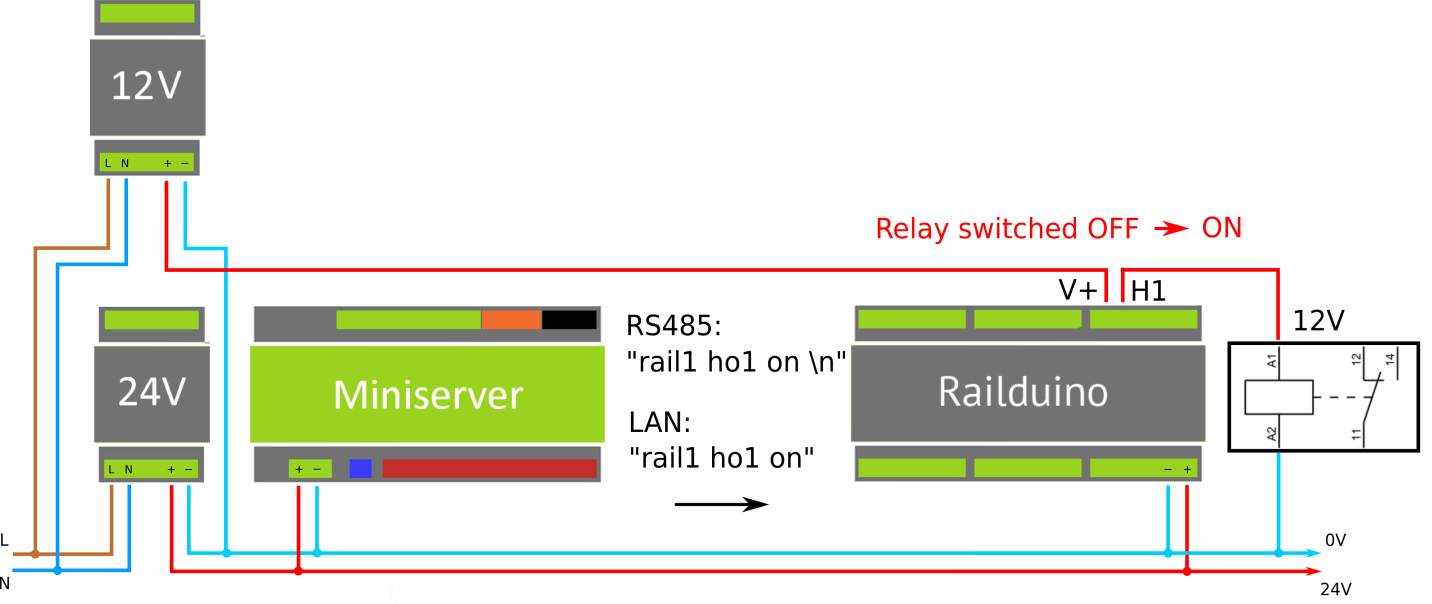
High side switch is dig. output which connects voltage V+ (terminal V+) to the load.



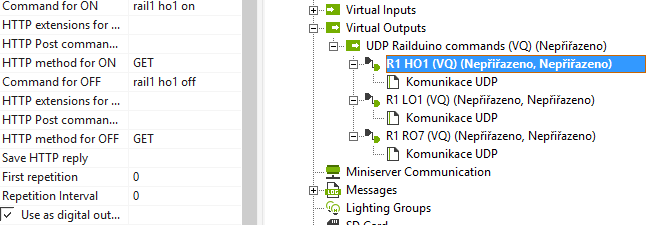
When using two power supplies with different output voltage level the output grounds must be connected!!



Max. permissible voltage at dig. outputs is **24V** DC! Max. permissible load current is **2A** per output!



Example of command settings for controlling digital output (hss) no. **1** in Loxone Config (LAN connection):



### Analog inputs – UDP communication

The readings of analog inputs is made in cycles and when the current value is different to previous value then the module sends this new value to the superior systém.

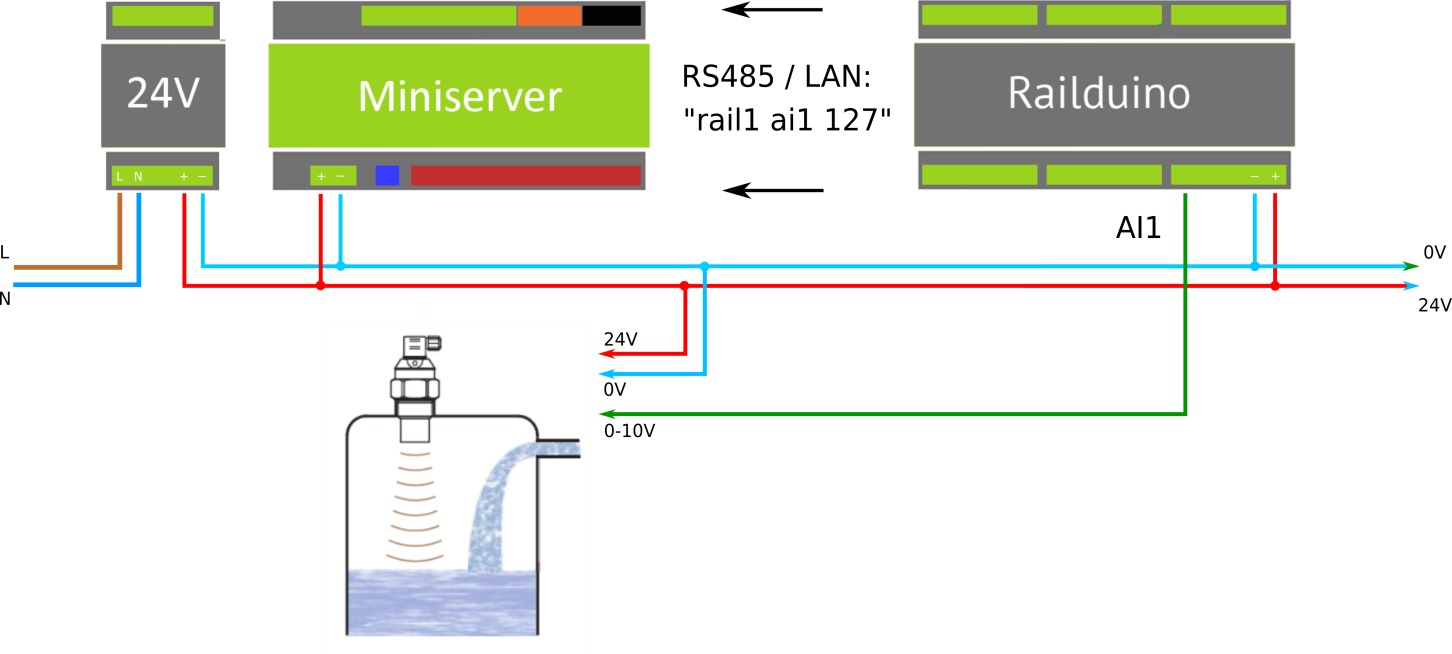
Max. permissible voltage at the analog inputs is **10V** DC! The packets with measured values are sent in format 0-255, e.g. 127 = 5 V



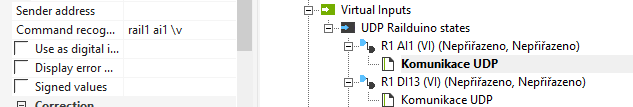
Default cycle for measuring analog inputs: **10** s



In virtual input command settings it must be **UNecked** the checkbox Use as digital input !!



Example of command settings for sensing analog input no.1 in Loxone Config (LAN connection):

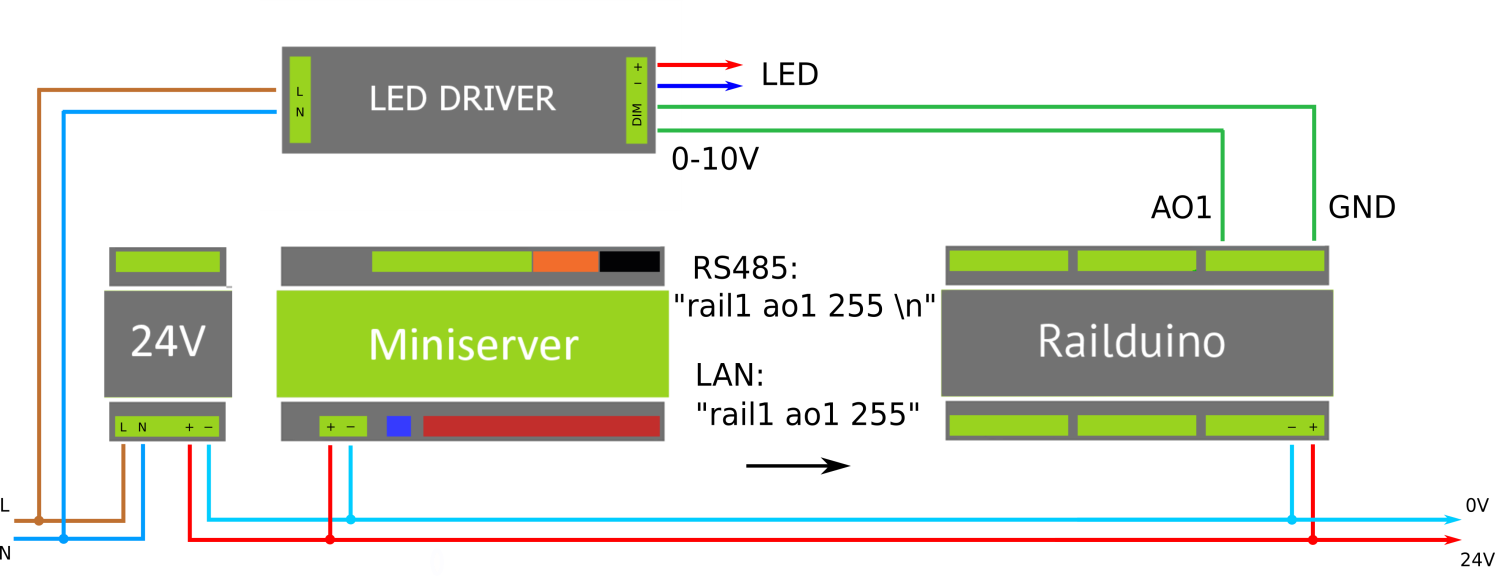


### Analog outputs – UDP communication

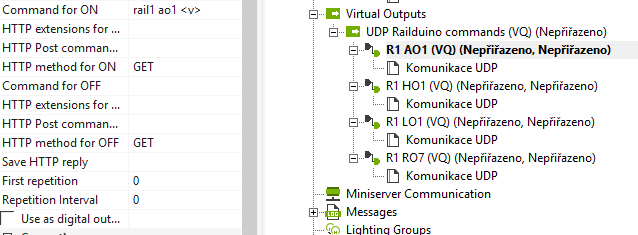
Analog outputs have common ground at the GND terminal, which is connected to 0V of Railduino module.



Values of analog outputs are **0-10V**, where required value of voltage is sent in format 0-255 (= 0-10V).



Example of command settings for controlling of analog output no.1 in Loxone Config (LAN connection):



### 1-wire bus – UDP communication



1wire sensors must be connected to the Railduino module only when module is **switched off**!!

It is possible to read out the serial number of the **max. 10pcs** of 1wire sensor in the **Loxone Config UDP monitor** and then to use it in the command recognition settings. Common ground of the sensors is connected to the 0V power supply of the Railduino module. 1wire sensors readings are made in cycles one by one sensors.

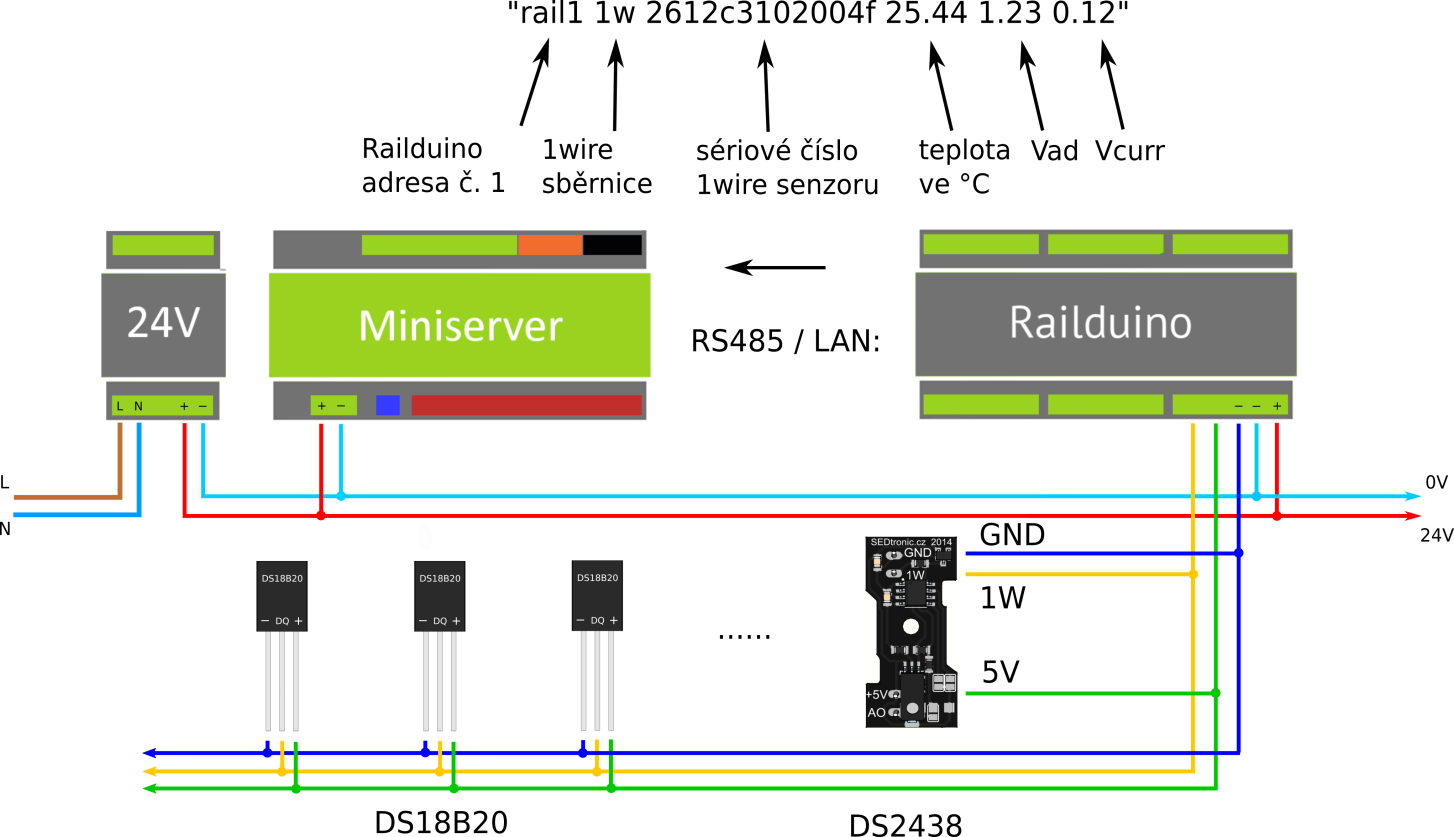
Supported types of the 1wire sensors:

* DS2438 – e.g. Unica 1wire modules
* DS18B20

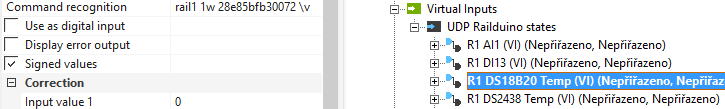


Default measurement cycle:: **30** s

Max. count of 1wire sensors: **10**



Example of command recognition settings for measurement of **temperature** sensor DS18B20 in Loxone Config:

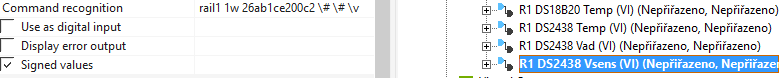


Example of command recognition settings for measurement of **temperature (**sensor DS2438) in Loxone Config:



Example of command recognition settings for measurement of **humidity** (sensor DS2438) in Loxone Config:

 Example of command recognition settings for measurement of **light intensity** (sensor DS2438) in Loxone Config:



### Other functions and features

### RS485 Baudrate

Railduino module has the ability to set two different communication protocols at the serial RS485 bus:

* Modbus RTU – DIP switch pin no. 6 in ON position
* UDP – default – DIP switch pin no. 6 in OFF position

### Ping and Heartbeat functions

The activity of Railduino module can be watched in two ways:

* LAN – using ping command – in Loxone block PING
* RS485 – reading packet Heartbeat – altering values 1 and 0 every minute

### Function Reset and Railduino module remote power ON/OFF

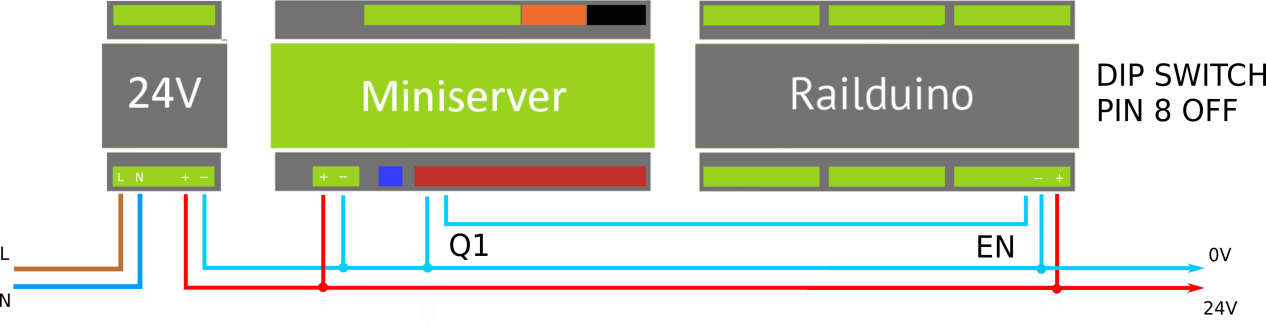
The module can be remotely reset / switched on/off by either software or hardware feature:

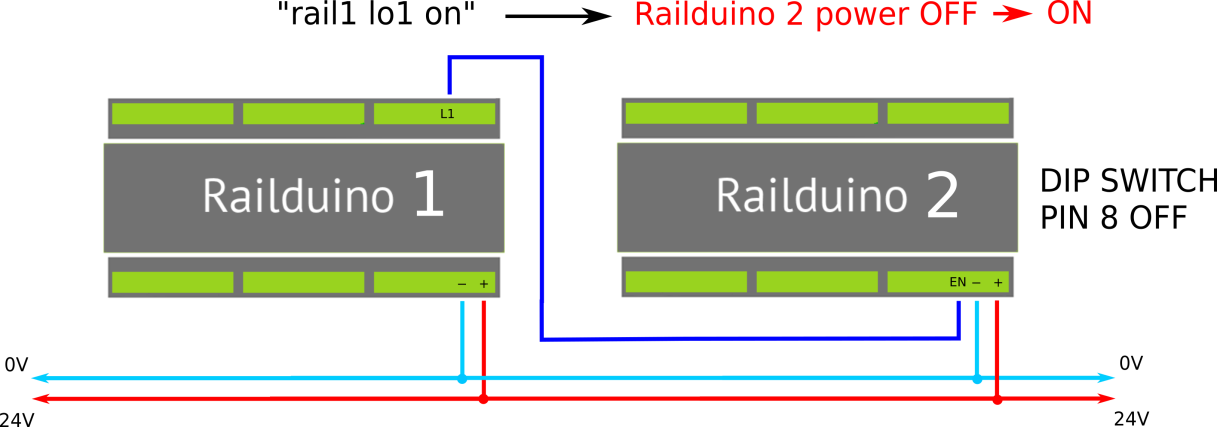
* Software - reset can be made with command “rst” – see. chap. 4.6. – reset command
* Hardware – module can be switched on off with the use of terminal Enable (EN) – see picture below

Below there are examples of the remote OFF/ON wiring with the use of miniserver or another Railduino module.



The DIP switch pin number 8 of the controlled Railduino module must be in postion **OFF**!!





### FAQ

*1. "What kind of connection to use with Loxone system?"*

It depends what are you expecting: LAN connection is fast (delay in terms of microseconds) with immediate responses, but there is need some workaround in LAN safety settings (subnets, VLANs etc..). RS485 is slower (delay in terms of miliseconds), but is simple, safe and reliable.

*2. "How long it will take from pushing the wall switch to actually switching on the lights?"*

Whole procedure of communication from Railduino to Loxone and back can be described as follows:

Request processing by Railduino (pusching the button) and sending the satet to Loxone – in case of LAN max. 20ms, in case of RS485 max. 30ms

Processing in Loxone (reception of information and making of further steps) and sending command to Railduino– in case of LAN max. 25ms, in case of RS485 max. 35ms

Processing in Railduino module (switching the lights on) - in case of LAN max. 20ms, in case of RS485 max. 30ms

So the react time will be max. 95ms (in both cases of LAN or RS485).

Delay is minimum, longer delay will be created by powering up the current power source of LED lights.

*3. "Is it possible to connect more then one Railduino to the same bus?"*

Yes, every modul has its own address which is set using the DIP switch settings. It is possible to connect up to 15 modules operating at the same bus LAN or RS485.

*4. "How is the Railduino programmed with in Loxone Config? Does thy system detect inputs and outputs?"*

It is neccessary to insert the virtual inputs and outputs into the Loxone program and make the correct settings (in case of LAN connection). More details are in the Loxone example file which can be downloaded in the download section.

*5. "What is the power sonsumption of Railduino module?"*

In the worst scenario – Railduino module with LAN connection (24V power supply) has consumption of approx. 120mA (with all relays off) and consumption of approx. 230mA (all relays on). Therefore power consumption is approx. 6W max. Minimal consumption can be lower then 2W (no LAN and all relays off).

*6. "Digital inputs are not working well, I have to push the switch two times, what is the problem?"*

There is bad settings in the virtual input command settings - Use as digital input must be UNchecked!

### Table of inputs/outputs

|  |  |  |  |
| --- | --- | --- | --- |
| Input/output | Tag | Railduino 2.0 SHIELD pin | Arduino MEGA pin |
| dig. input | DI | 1 | 34 |
| dig. input | DI | 2 | 32 |
| dig. input | DI | 3 | 30 |
| dig. input | DI | 4 | 28 |
| dig. input | DI | 5 | 26 |
| dig. input | DI | 6 | 24 |
| dig. input | DI | 7 | 22 |
| dig. input | DI | 8 | 25 |
| dig. input | DI | 9 | 23 |
| dig. input | DI | 10 | 21 |
| dig. input | DI | 11 | 20 |
| dig. input | DI | 12 | 19 |
| dig. input | DI | 13 | 36 |
| dig. input | DI | 14 | 38 |
| dig. input | DI | 15 | 40 |
| dig. input | DI | 16 | 42 |
| dig. input | DI | 17 | 44 |
| dig. input | DI | 18 | 46 |
| dig. input | DI | 19 | 48 |
| dig. input | DI | 20 | 69 |
| dig. input | DI | 21 | 68 |
| dig. input | DI | 22 | 67 |
| dig. input | DI | 23 | 66 |
| dig. input | DI | 24 | 65 |
| relay output | RO | 1 | 37 |
| relay output | RO | 2 | 35 |
| relay output | RO | 3 | 33 |
| relay output | RO | 4 | 31 |
| relay output | RO | 5 | 29 |
| relay output | RO | 6 | 27 |
| relay output | RO | 7 | 39 |
| relay output | RO | 8 | 41 |
| Input/output | Tag | Railduino 2.0 SHIELD pin | Arduino MEGA pin |
| relay output | RO | 9 | 43 |
| relay output | RO | 10 | 45 |
| relay output | RO | 11 | 47 |
| relay output | RO | 12 | 49 |
| analog input | AI | 1 | 64 |
| analog input | AI | 2 | 63 |
| analog output | AO | 1 | 3 |
| analog output | AO | 2 | 2 |
| high switch | HO | 1 | 5 |
| high switch | HO | 2 | 6 |
| high switch | HO | 3 | 7 |
| high switch | HO | 4 | 8 |
| low switch | LO | 1 | 9 |
| low switch | LO | 2 | 11 |
| low switch | LO | 3 | 12 |
| low switch | LO | 4 | 18 |
| RS485 | TX | A+ | 14 |
| RS485 | RX | B- | 15 |
| RS485 | EN | EN | 16 |
| 1 wire | 1W | 1W | 62 |
| DIP switch | DS1 | TERM | - |
| DIP switch | DS2 | EN | - |
| DIP switch | DS3 | DIP1 | 57 |
| DIP switch | DS4 | DIP2 | 56 |
| DIP switch | DS5 | DIP3 | 55 |
| DIP switch | DS6 | DIP4 | 54 |
| DIP switch | DS7 | DIP5 | 58 |
| DIP switch | DS8 | DIP6 | 59 |
| LED blue | LED1 |  | 13 |
| LED green | LED2 |  | 17 |

### Basic parameters and operation conditions

Railduino modules is designed to be mounted in control cabinets onto a DIN rail. The basic parameters are shown in following tables.

**BASIC PARAMETERS**

|  |  |
| --- | --- |
| Protection class | II |
| Device type | Built-in |
| Supply voltage | 12-24 V DC |
| Max. power consumption | max. 6 W |
| IP ratings | IP 20 |
| Dimensions | 160 x 90 x 60 mm |
| Connection |  |
| Relay outputs | Push in terminals, max. 2,5 mm2 |
| Other inputs/outputs | Push in terminals, max. 1,5 mm2 |

**OPERATION CONDITIONS**

|  |  |
| --- | --- |
| Environment | Normal |
| Operational temperature | 0 °C až +55 °C |
| Relative humidity range | 10 % - 95 % |
| Operational position | Vertical |
| Type of operation | Permanent |
| Max. voltage and current at the relay outputs | 250V AC / 7A (no. 1,2,7,8) / 4A (no. 3,4,5,6,9,10,11,12) |
| Max. voltage at the digital inputs | 24V DC |
| Max. voltage and current at the digital outputs | 24V DC / 2A / output |
| Max. voltage at the analog inputs/outputs | 10V DC |

**SEDtronic.eu**

Ing. Pavel Sedláček

Jana Koziny 1628/31

Teplice 415 01

Czech Republic