Protocols

Application Note

Modbus

Table of Contents

1 Introduction

1.1 Purpose

This document describes how to configure MODBUS communications in METEL devices and basic information about protocol.

1.2 Scope

This document describes

- Understanding MODBUS communications
- How to configure METEL s.r.o. switches for MODBUS
- Examples

2 Understanding MODBUS Communications

METEL s.r.o. implemented according to:

www.modbus.org

Standard Modbus ports on Metel controllers use an RS–485 compatible serial interface that defines connector pinouts, cabling, signal levels, transmission baud rates, and parity checking. Controllers can be networked directly via LAN or WAN.

Controllers communicate using a Master–Slave technique, in which only one device (the Master) can initiate transactions (called ‘queries’). The other devices (the Slaves) respond by supplying the requested data to the Master, or by taking the action requested in the query.

The Master can address individual Slaves, or can initiate a broadcast message to all Slaves. Slaves return a message (called a ‘response’) to queries that are addressed to them individually. Responses are not returned to broadcast queries from the master.

The Modbus protocol establishes the format for the master’s query by placing into it the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error–checking field. The Slave’s response message is also constructed using Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error–checking field. If an error occurred in receipt of the message, or if the Slave is unable to perform the requested action, the Slave will construct an error message and send it as its response.

Master - A Modbus Master is typically a host supervisory computer running software that will communicate with one or more Modbus Slave devices. The Master can be a METEL switch or PLC (IPLOG) too.

Slave - A Modbus Slave is typically a device connected via LAN or serial interface waiting for Master to initiates commands.

METEL devices support protocols:

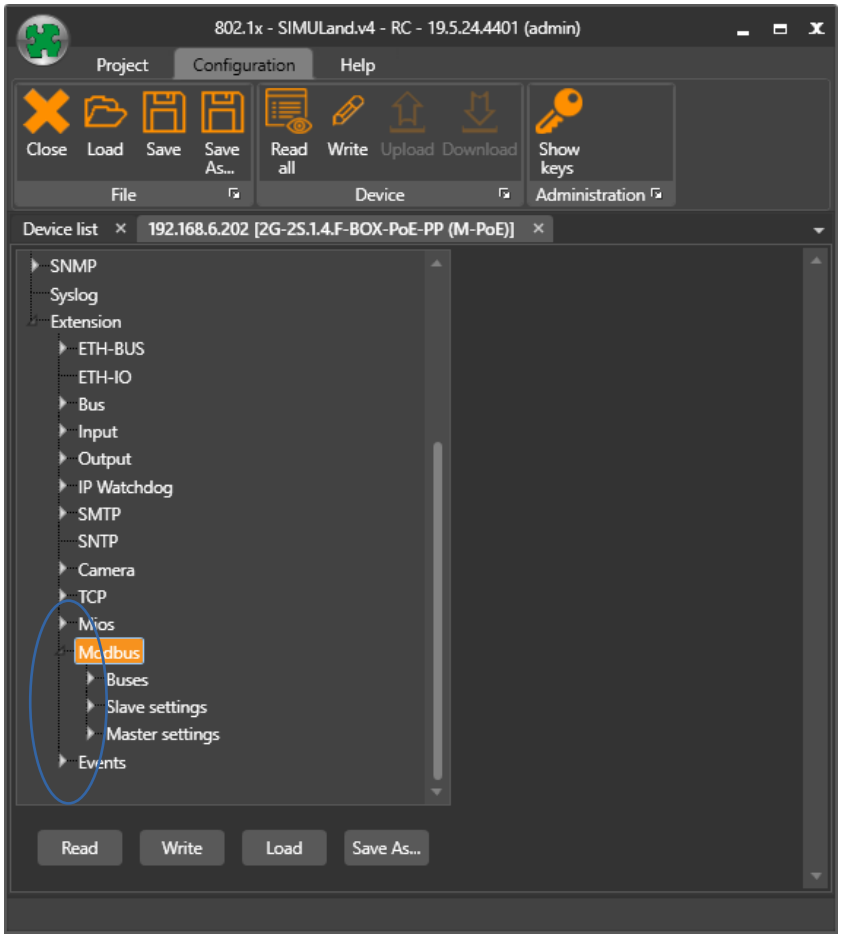
Modbus RTU (RS485 serial interface)

Modbus TCP/IP

3 Configuration

This section describes basic configuration of Modbus.

- **Buses** - Serial interface basic configuration, **None, Slave RTU or Master RTU**.
- **Slave settings** - Configuring the device as a Slave RTU or TCP.
- **Master settings** - Configuring the device as a **Master** and it´s parameters.



3.1 Buses

Item **Buses** enables Modbus RTU mode on serial interfaces BUS1 and BUS2.

Mode - **None** - Modbus is disabled.

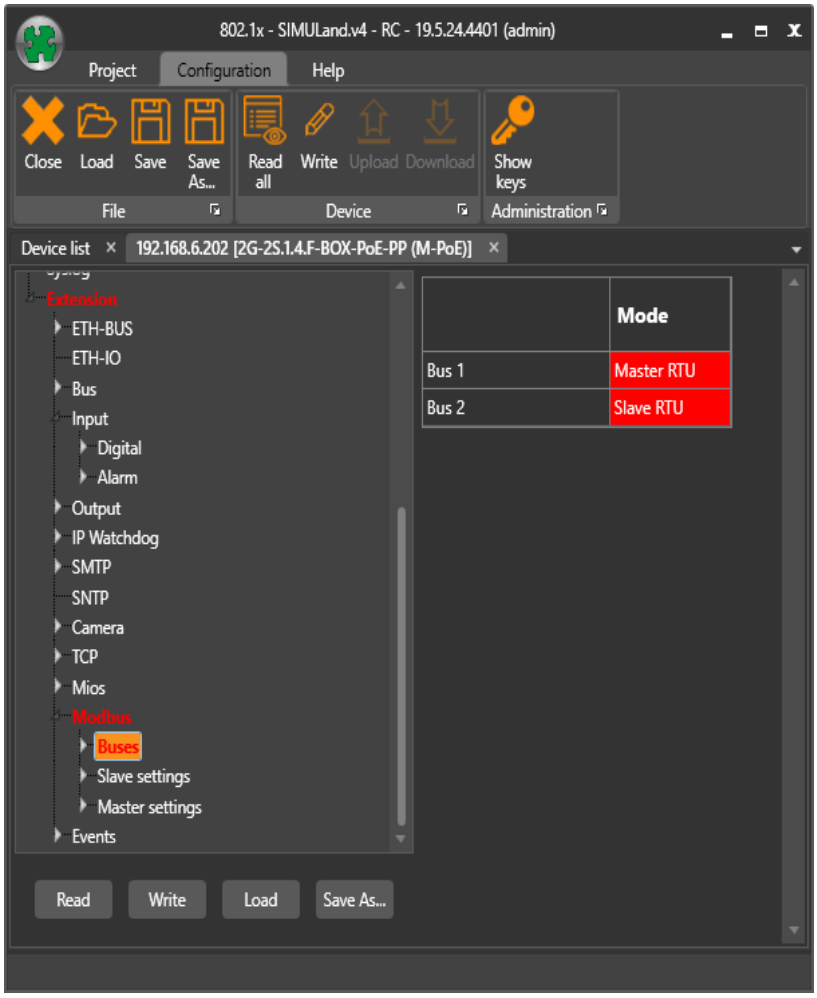
- **Master RTU** - Switch is a Master, sends queries and commands to the Slaves. This mode is usually used if any Modbus Slave device is connected to the serial

interface (Thermometer, Hydrometer, I/O moduls…..)

- **Slave RTU** - Switch is a Slave, connected via serial interface and waits to queries or commands of any other Master Modbus device on the same serial line.

* Switch can be Master on one of the buses and on the second one as a Slave.

* If Modbus is activated no other communication can be used there (MIOS or standard RS485 for I&HAS, FS, PTZ……



3.2 Slave settings

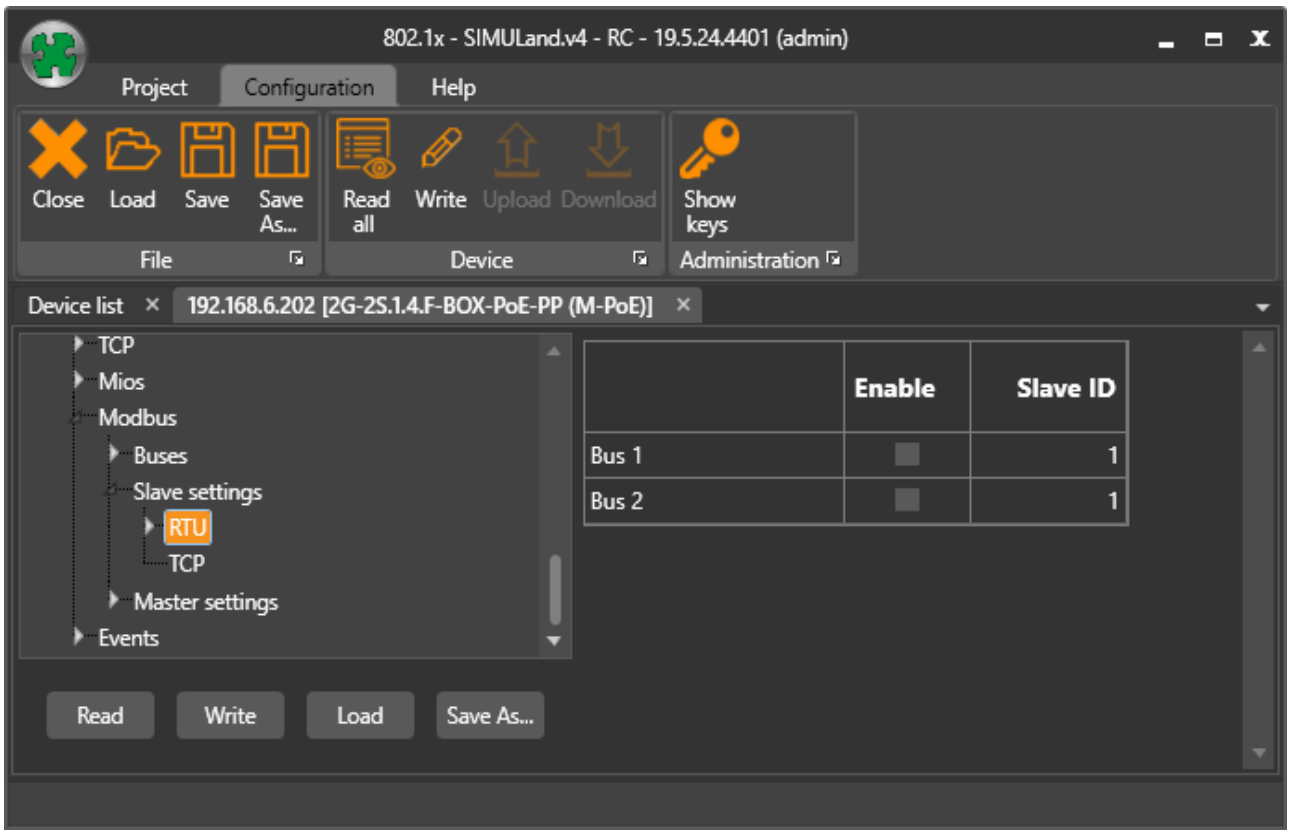
Configuration when the switch behavior is as a Modbus Slave RTU or TCP device.

3.2.1 RTU

Switch is a Slave and connected together via serial interfaces BUS1 or BUS2 with one Modbus Master RTU device and other Modbus Slave devices.

RTU - **Enable** - Activate Slave ID specifying.

- **Slave ID** - Each Slave in a system is assigned a unique unit address from 1 to 247. When the master requests data, the first byte it sends is the Slave address. This way each Slave knows after the first byte whether or not to ignore the message.



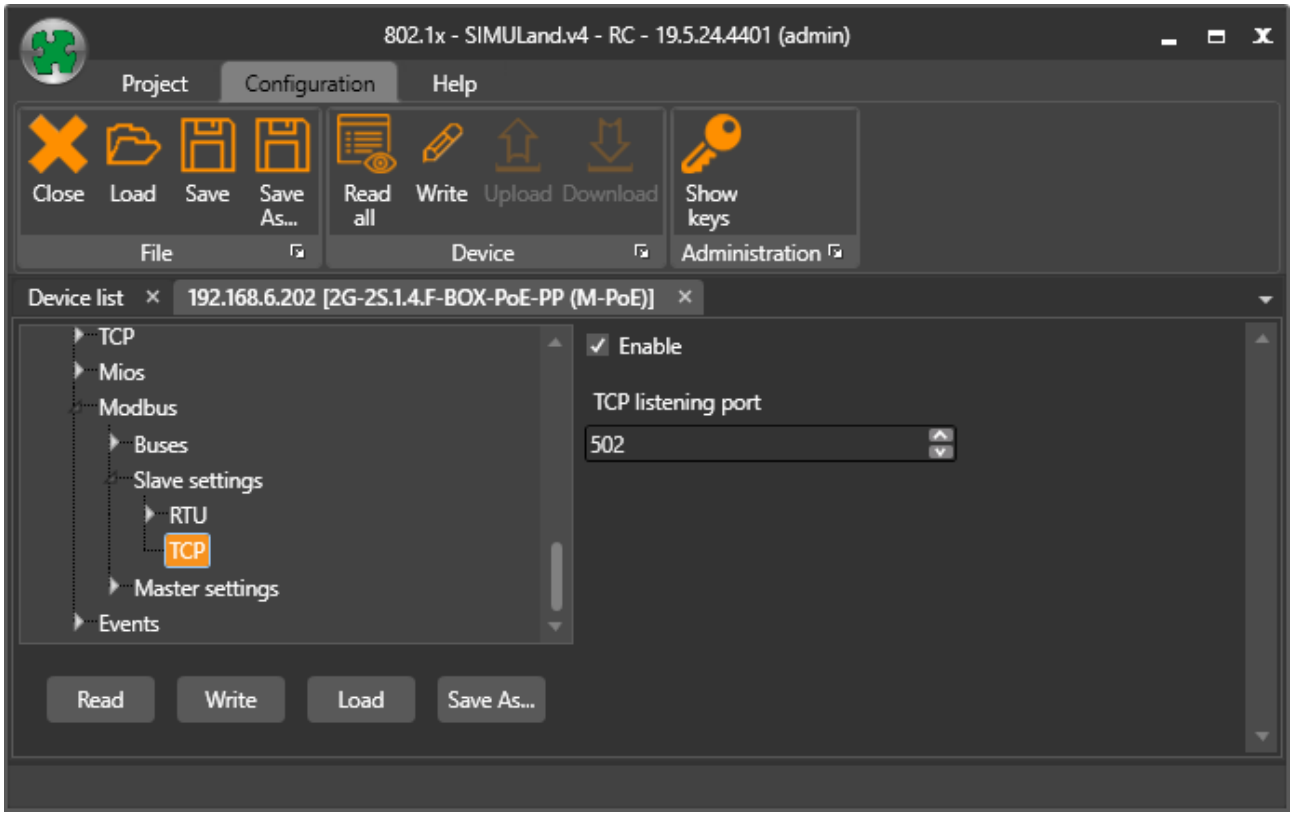
3.3

3.3.1 TCP

Switch is a Slave and connected together via TCP connection with the Modbus TCP Master device.

TCP - **Enable** - Allows TCP connection.

- **TCP listening port** -The listening **TCP port 502 is reserved for Modbus** communications. It is mandatory to listen by default on that port. However, some device or applications might require that another port is dedicated to Modbus over TCP. For that reason, it is possible to change Modbus TCP listening port. It is important to note that even if another TCP server port is configured for Modbus service in certain applications, TCP server port 502 must still be available in v addition to any application specific ports.

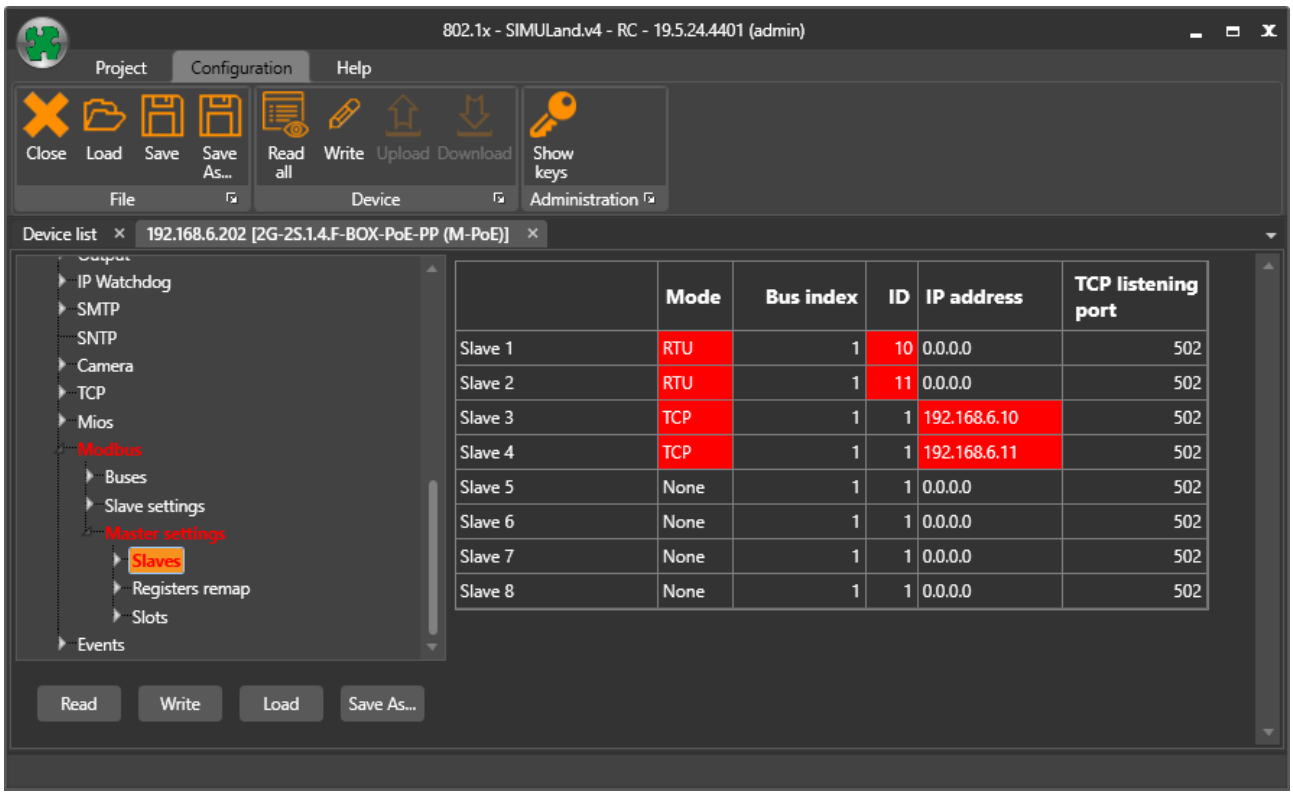


3.4 Master settings

This section describes configuration when the switch behavior is as a Modbus Master RTU or TCP. The configuration of this menu is linked to **Events (**Event management) where reading and writing registers for Modbus Slave device action are created.

3.4.1 Slaves (Modbus Slave connection specifications)

Modbus Master switch supports connection with eight Modbus Slave devices. Modbus Slaves configurations allows use of part of the Slaves with mode RTU or TCP together. Each row represents one Modbus Slave device configuration and creates a link for **Events (**Event management).



	Mode	Bus index	ID	IP address	TCP listening port
Slave 1	RTU	1	10	0.0.0.0	502
Slave 2	RTU	1	11	0.0.0.0	502
Slave 3	TCP	1	1	192.168.6.10	502
Slave 4	TCP	1	1	192.168.6.11	502
Slave 5	None	1	1	0.0.0.0	502
Slave 6	None	1	1	0.0.0.0	502
Slave 7	None	1	1	0.0.0.0	502
Slave 8	None	1	1	0.0.0.0	502

Slave 1...8 - Description, it is saved only in SIMULand.v4 project, not device.

Modbus Slave in RTU mode:

Mode - Modbus Slave mode TCP (LAN, WAN) or RTU (serial interface).

Bus index - Number of serial interface where Slave is connected (BUS1/BUS2).

ID - Modbus Slave device ID connected to serial interface.

Modbus Slave in TCP mode:

IP address - IP address of Slave Modbus device connected to LAN, WAN.

TCP listening port - TCP port on which the Modbus Slave device is listening.

3.4.2 Register remap (Modbus register specifications)

Register remap items specify specific registers and creates a link for **Events (**Event management). Each switch supports configuration for 64 different registers and it´s combination.

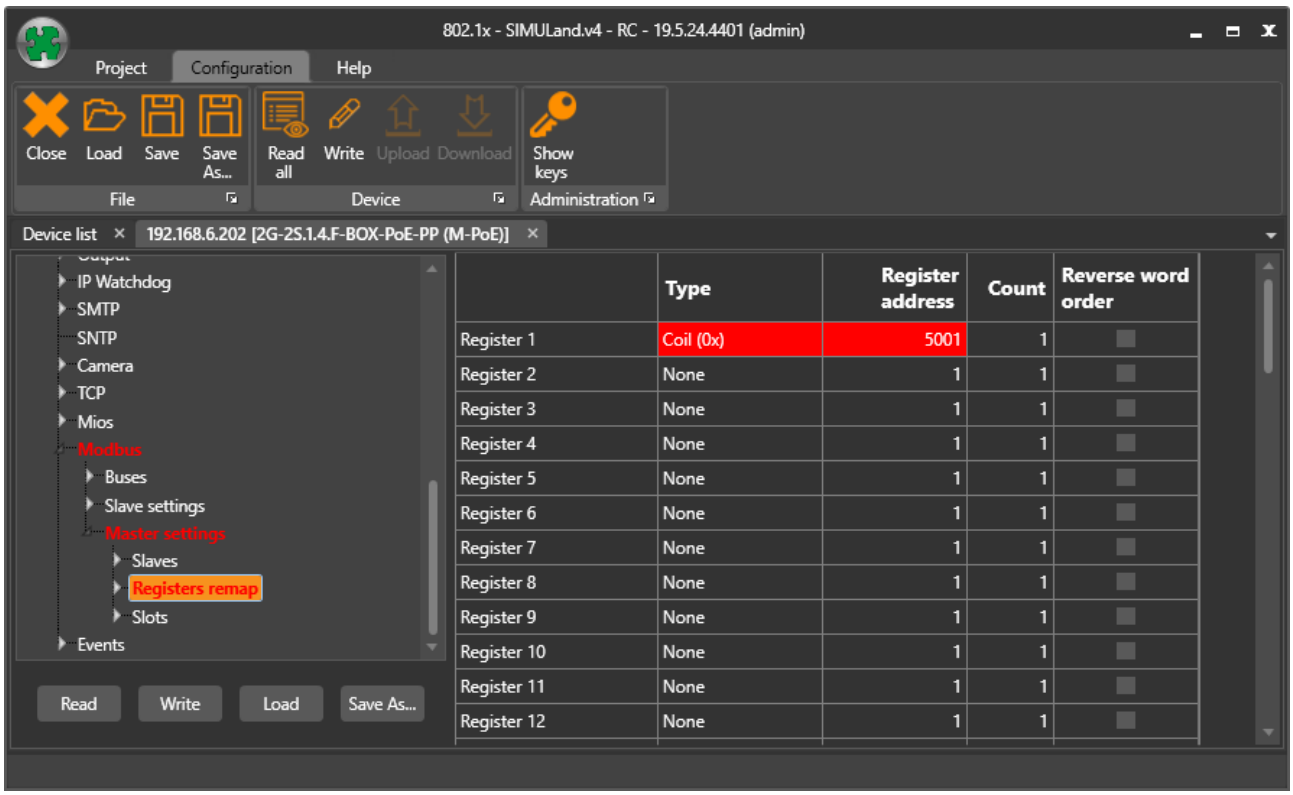
Register 1….64 - Register description, it is saved only in SIMULand.v4 project, not device.

Type - Specifies the type of register:

Register name	Type	Read-Write	Example
Coil	1-bit	Read-Write	Relay Output
Discrete	1-bit	Read-Only	Digital Input state
Holding	16-bit	Read-Write	Analog Outputs
Input	16-bit	Read-Only	Analog Inputs

Count - The quantity of registers to be read, starts at **Register address**. All other fields in the expected response must not exceed the allowed length: 256 bytes.

Revers word order - Proprietary reading of 32-bit messages, in which the reverse order of 16-bit registers is read.



3.4.3 Slots (Slaves – Registers mapping)

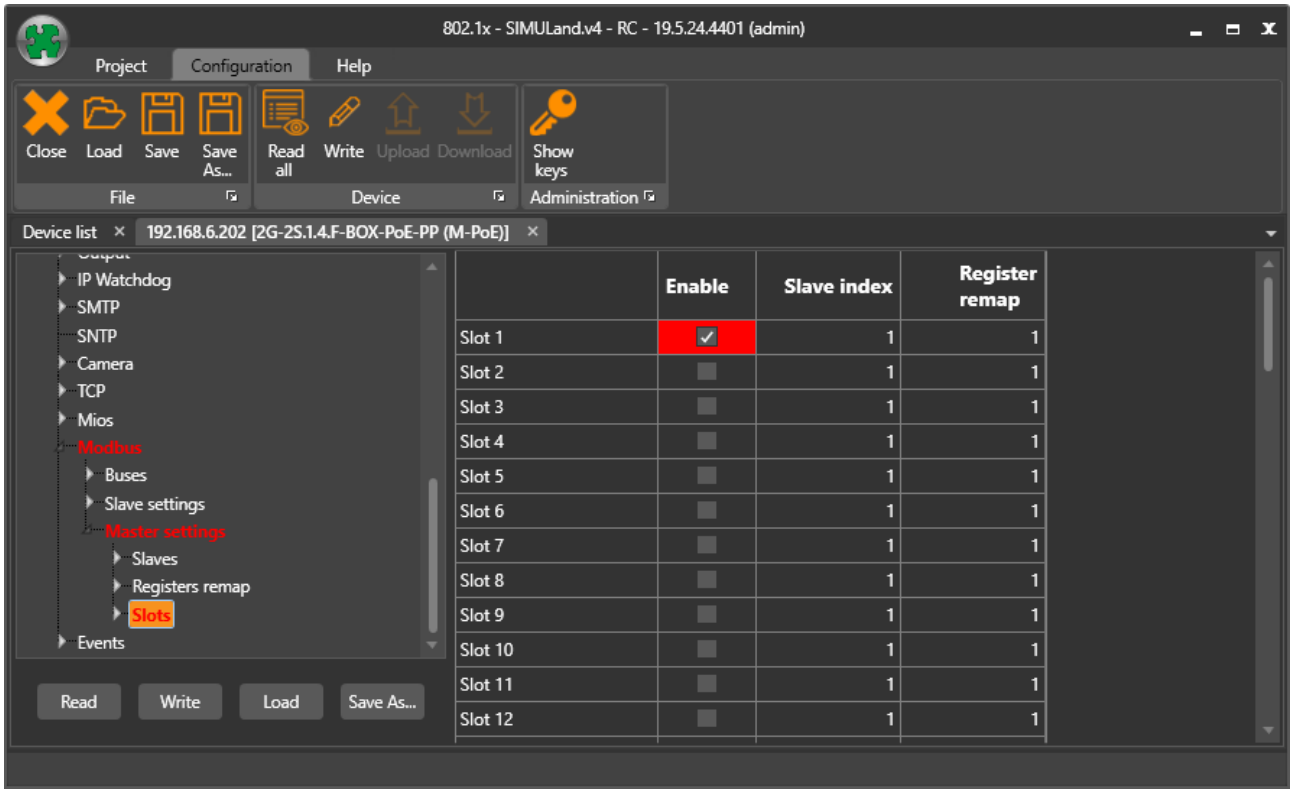
The **Slots** items merge configuration of menus **Slaves** (Modbus Slave connection specifications) and **Register remap** (Modbus register specifications). The combination of these two menus provides for switch all important information for controlling Modbus registers in devices connected at it´s serial interfaces or via TCP.

Slot 1….64 - It is a name or description for connection between tables of menus Slaves and Registers remap, it is saved only in SIMULand.v4 project, not device.

Enable - Activates a specific row.

Slave index - Row number from menu **Slaves**.

Register remap - Row number from menu **Register remap**.



3.5 Events

Menu Events merge all automatic switch actions with Modbus registers.

3.5.1 Modbus TCP reading values of the switch

Example:

Any Modbus Master device (Computer running software, IPLOG, other switch…) will read switch Modbus register merge with Switch Digital Input IN1 state.

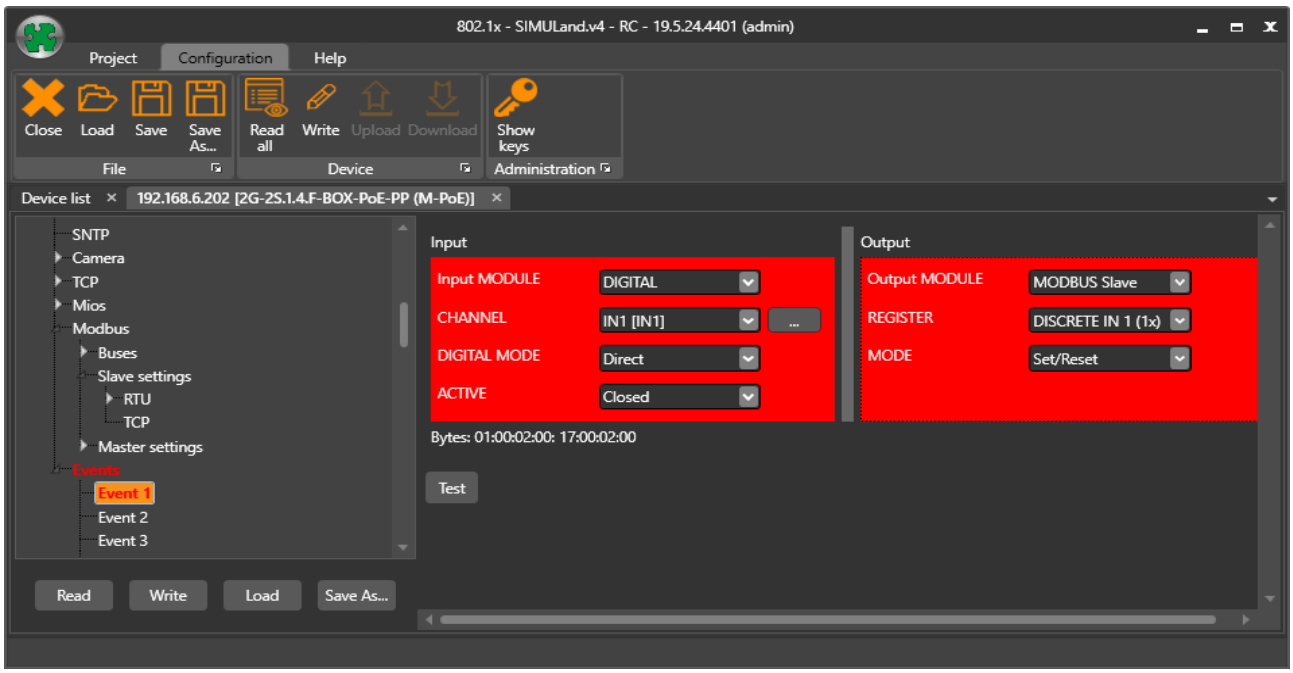
Step by step:

1 Events

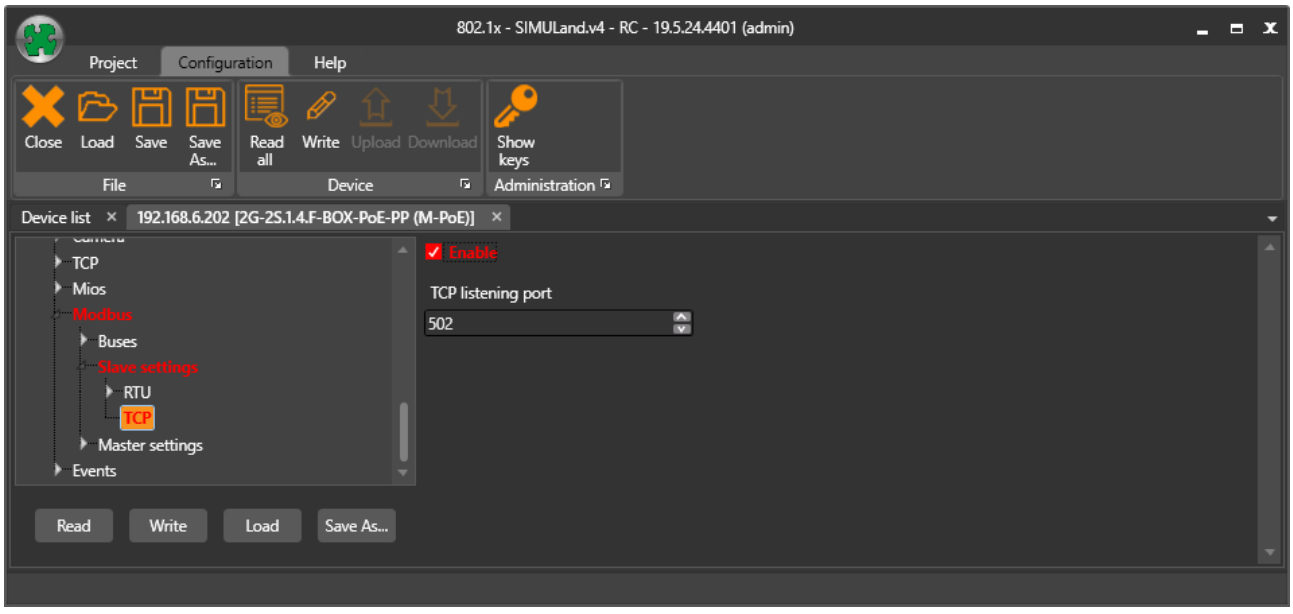
2. Modbus Slave TCP

3. Reading

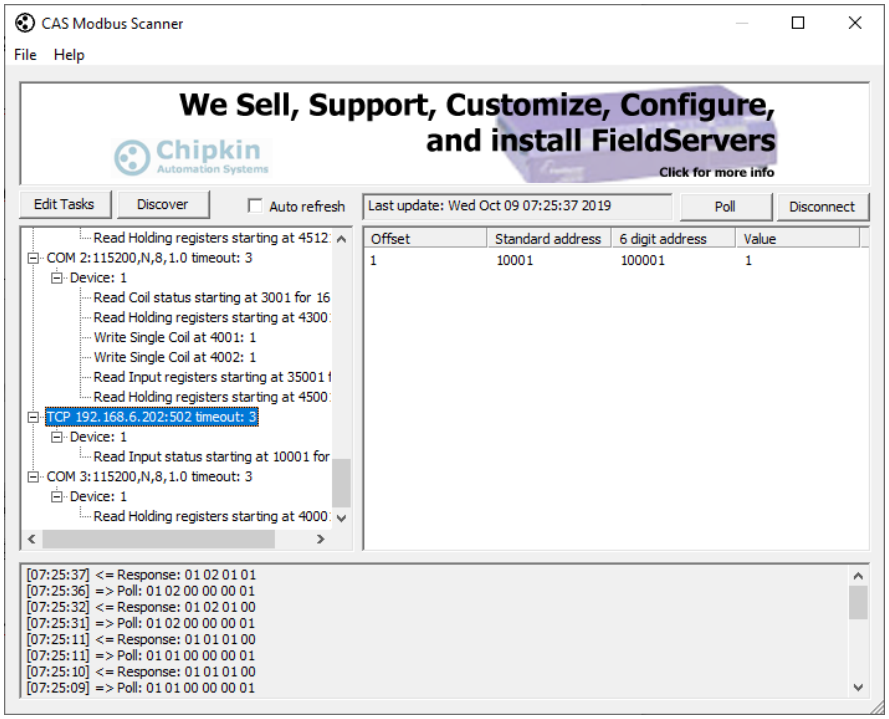
1. Automatic action. Switch copy both states ACTIV / INACTIV of Digital input IN1 to Modbus register Discrete IN 1 (register address 10001).



2. Enable Modbus Slave TCP port in menu **Modbus->Slave settings->TCP**. Port 502 is default TCP port for Modbus communication.



3. Register reading. Software CAS Modbus Scanner reads from IP address 192.168.6.202 port 502 Discrete Modbus register at address 10001.



3.5.2 Reading value of the Modbus RTU Slave device

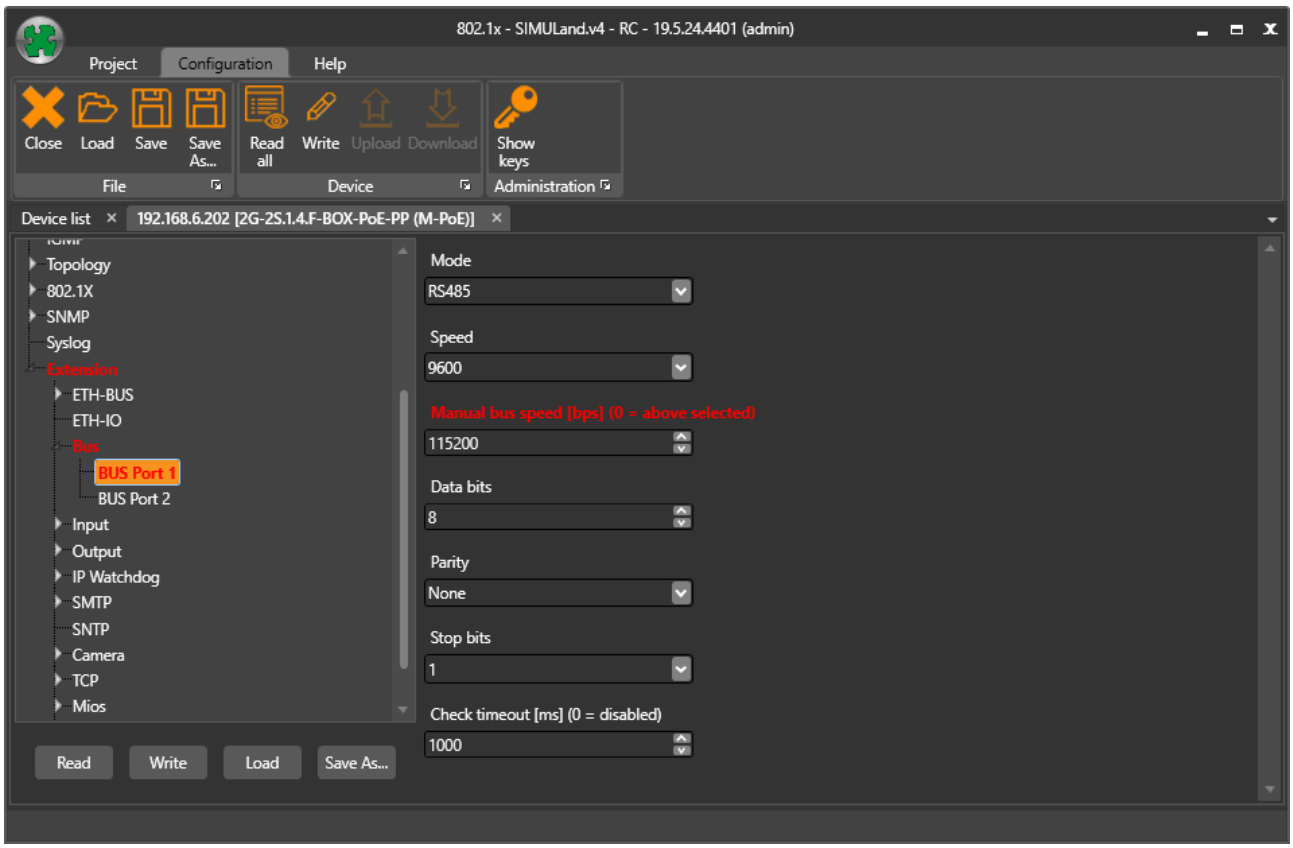
Example:

Digital Input value IN1 (register 3009) of Modbus device (BUS ID 1) connected to serial interface BUS 1 at switch is being read and copy into the Modbus register Discrete 1.

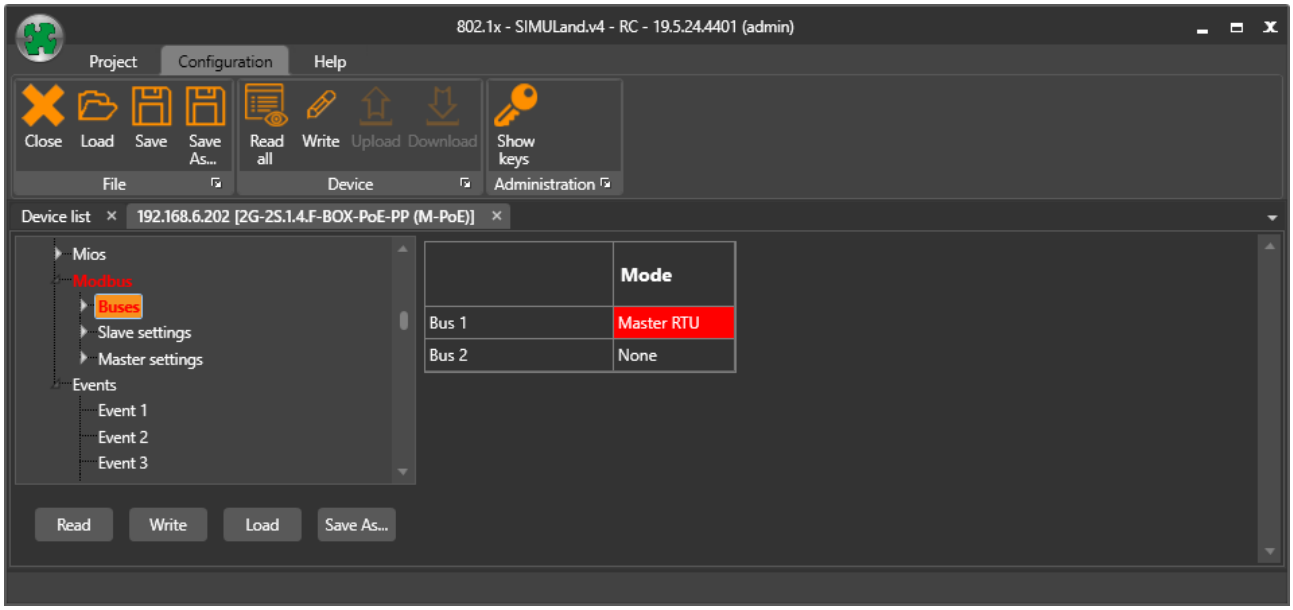
Step by step:

1. **BUS parameters**
2. **Buses**
3. **Slaves**
4. **Registers remap**
5. **Slots**
6. **Events**

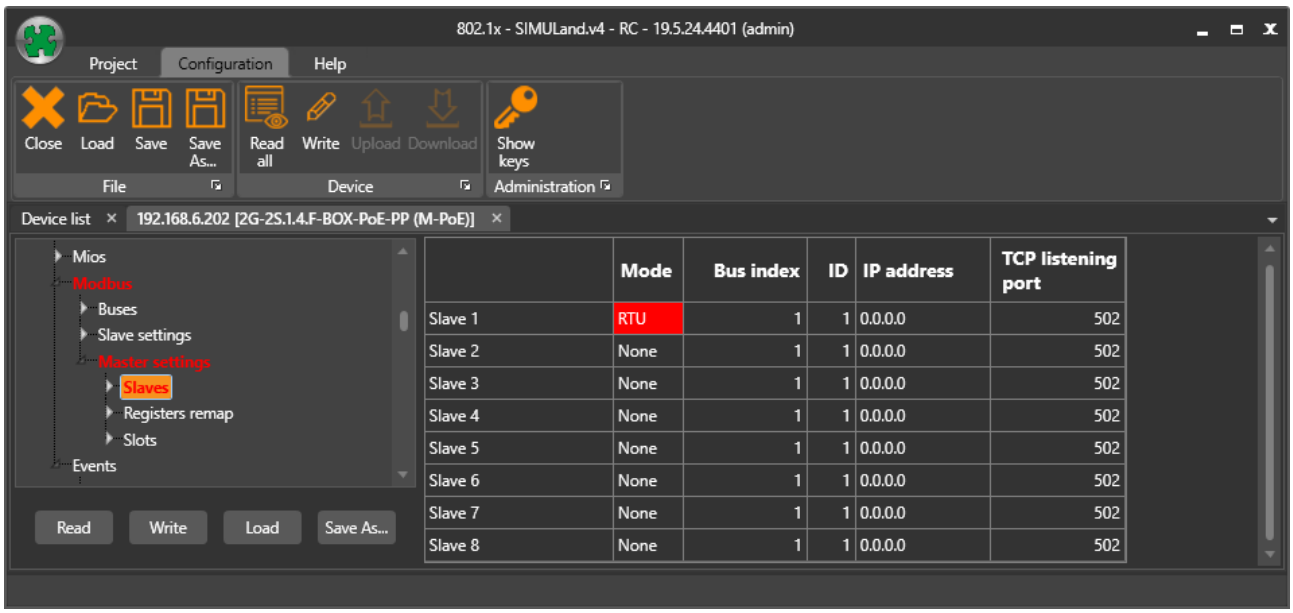
1. The BUS 1 communication parameter settings. Speed, Data bits, Parity, Stop bits.



2. Enable Master RTU mode at BUS 1.

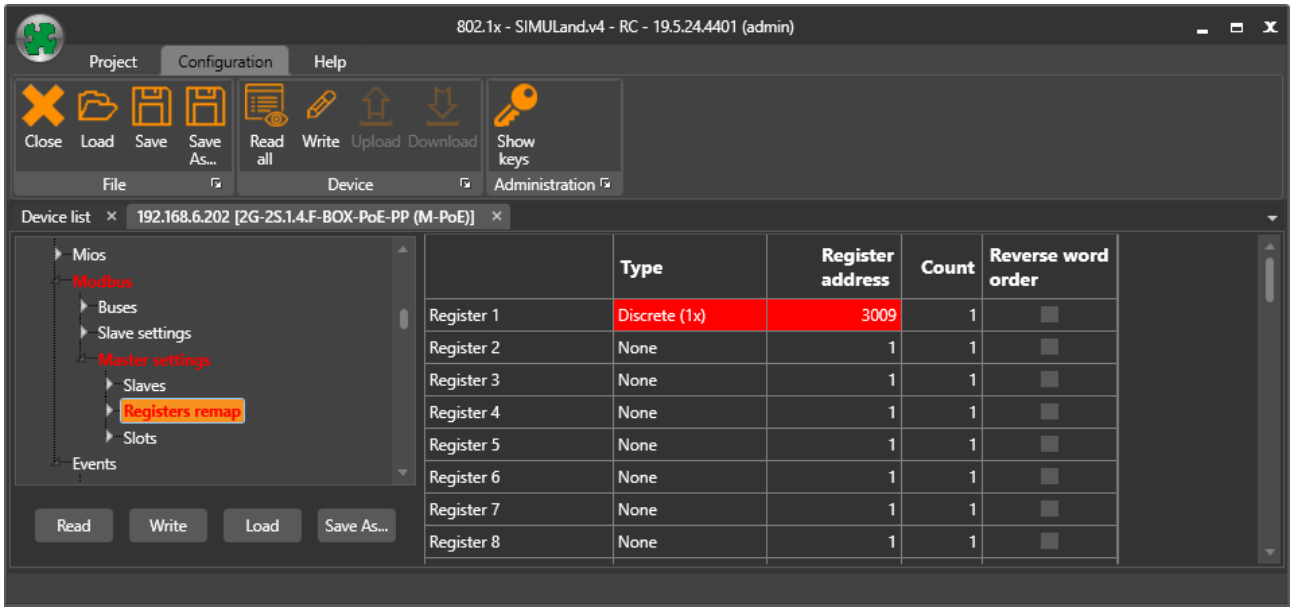


3. Create rules for Modbus Slave device connected at serial interface. Mode RTU, BUS number 1, Modbus Slave bus indentification ID 1.

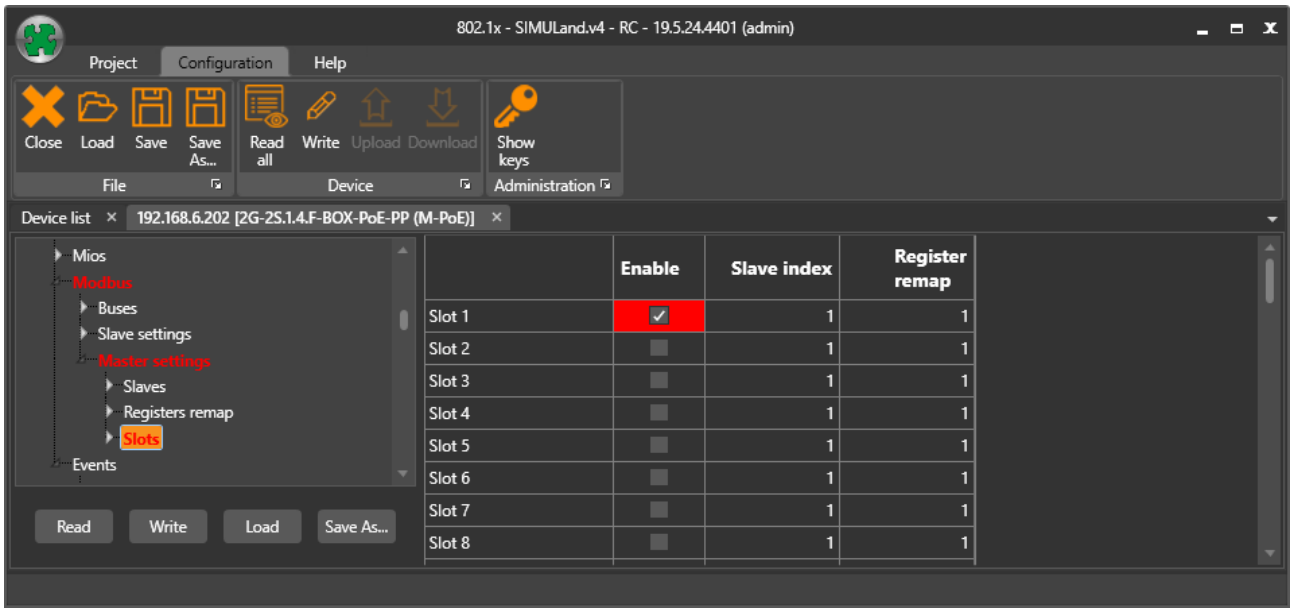


4. Setup register parameters.

Digital Input is 1bit Read-Only register -> type Discreate, register address 3009 . Count 1, only one register will be reading. Each manufacturer provides datasheets with modbus registers informations.



5. Merging Modbus Slaves table with Register remap parameters. **Slaves** table row 1 with **Register remap** slot 1.



6. Automatic action. Copy Digital input 1 state from Modbus Slave device at switch relay Otput 1.

Input part:

Input MODULE MODBUS Master, only Master can read and write registers.

SLOT merge between Modbus device and registers of Modbus->Master settings->Slots.

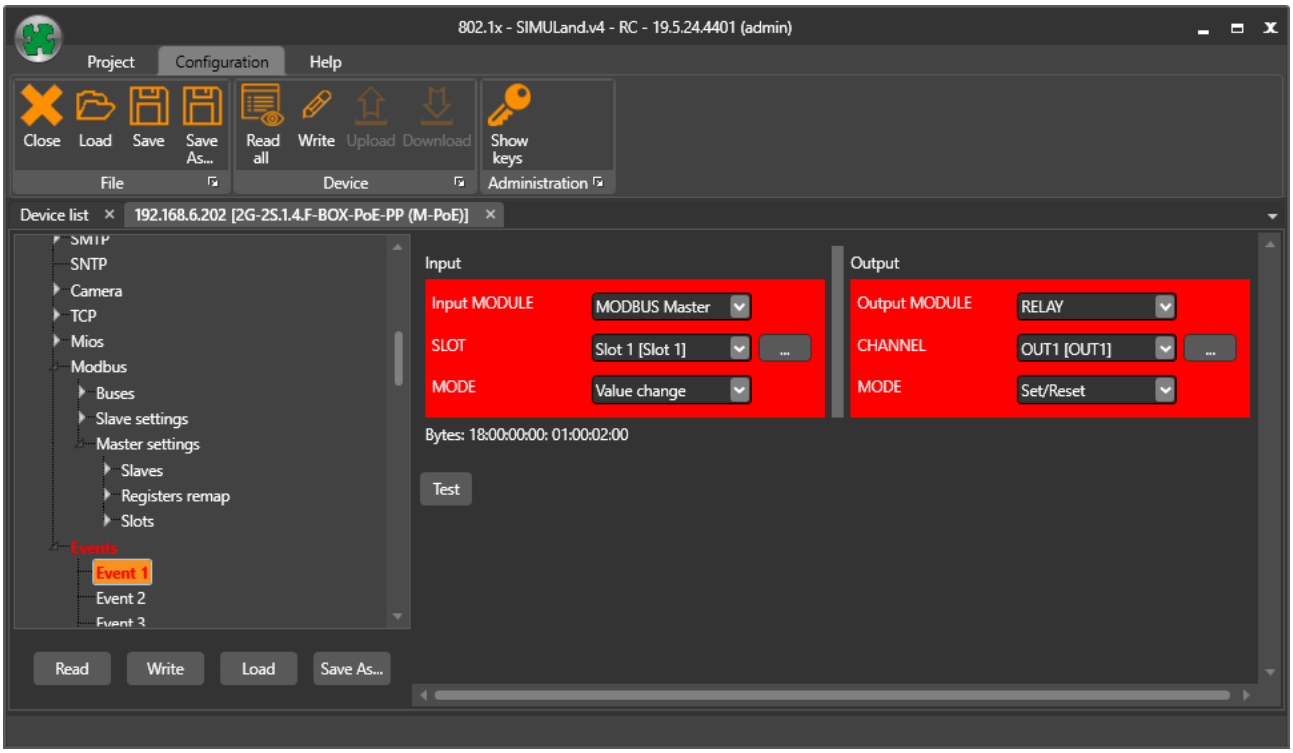
MODE Value change, each input change will be copy to output.

Output part:

Output MODULE RELAY, relay output at switch.

CHANNEL OUT1, output specification (index).

MODE Set/Reset, both digital input states ACTIVE / INACTIVE will corespond to relay states CLOSED / OPEN.



3.6 Extension-BUS

Specifies the physical properties at the serial interface. Used in conjunction with Modbus Master RTU when any Modbus Slave device is connected at serial interface.

Mode - Modbus use mode RS485, other modes are for Control panels.

Speed - List of supported bit rates at the BUS.

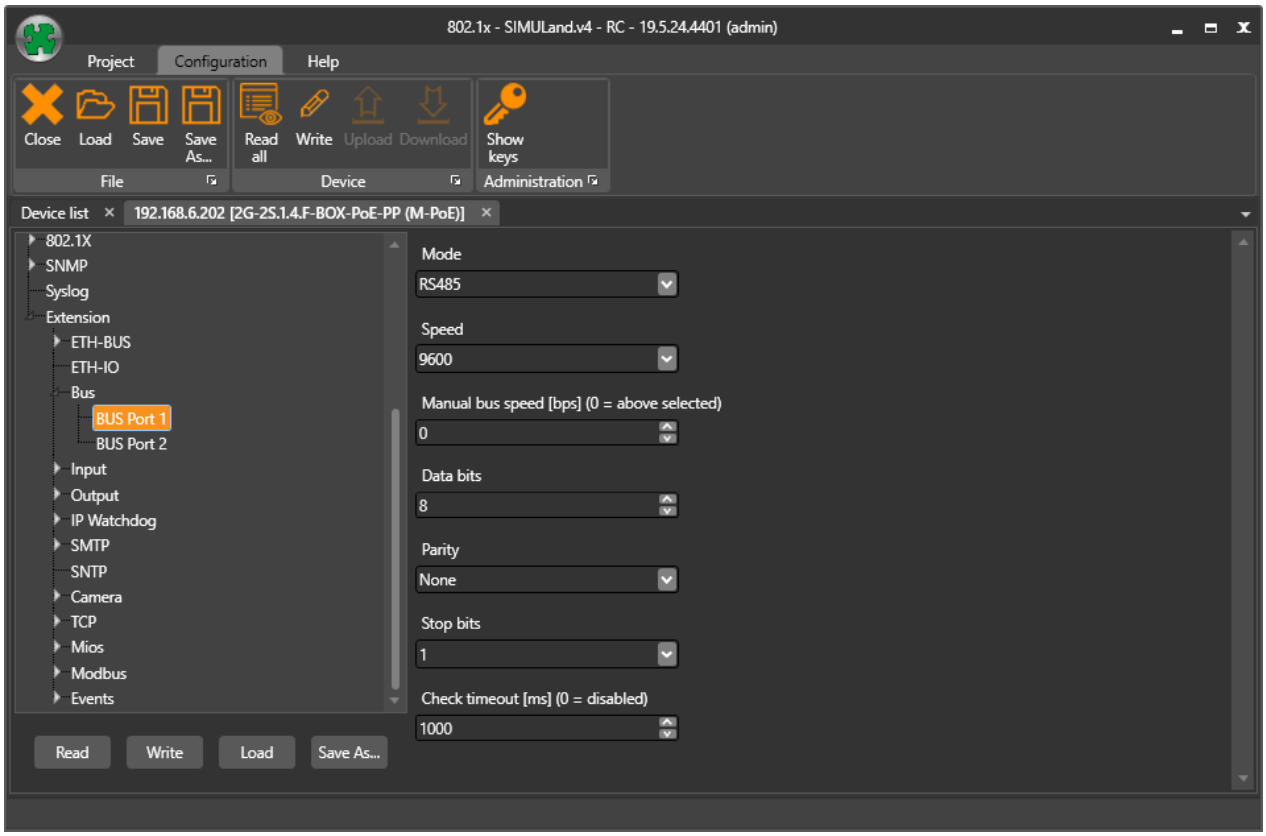
Manual bus speed - Switch supports max bit rate 115200, if Speed List does not contains any of the supported, it is possible to write it manually.

Data bits - The number of Data bits.

Parity - Method of detecting errors in transmission.

Stop bits - The number of Stop bits.

Check timeout - Function that controlls activity on the bus. If at 1000ms (default) there is no activity, the bus is initializing.


802.1x - SIMULand.v4 - RC - 19.5.24.4401 (admin)
Project
Configuration
Help
Close
Load
Save
Save As...
Read all
Write
Upload
Download
Show keys
File
Device
Administration
Device list
192.168.6.202 [2G-2S.1.4.F-BOX-PoE-PP (M-PoE)]
802.1X
SNMP
Syslog
Extension
ETH-BUS
ETH-IO
Bus
BUS Port 1
BUS Port 2
Input
Output
IP Watchdog
SMTP
SNTP
Camera
TCP
Mios
Modbus
Events
Mode
RS485
Speed
9600
Manual bus speed [bps] (0 = above selected)
0
Data bits
8
Parity
None
Stop bits
1
Check timeout [ms] (0 = disabled)
1000
Read
Write
Load
Save As...