



# **Modbus Gateway Product manual**

**HDL-ModBus.231**

**Version1.0**

## Table of contents

<b>1. Product Information.....</b>	<b>3</b>
- Internal Database.....	4
- Modbus Interface.....	5
- HDL-Bus Interface .....	6
<b>2. Application Examples.....</b>	<b>7</b>
- Full Monitoring and Control.....	7
- Full Integration.....	8
- Integrate serial Modbus to IP network.....	9
- Third Party Integration.....	10
<b>3. Hardware Information.....</b>	<b>11</b>
- Technical Data.....	11
- On-board I/O.....	11
- I/O Specification & Wiring.....	12
- LED signals.....	12
- Jumpers.....	13
- Dimensions & Weight.....	15
- Enclosure.....	15
- Mounting type.....	16
<b>4. Safety instruction.....</b>	<b>16</b>
<b>5. Software Configuration.....</b>	<b>17</b>
- Features.....	17
- Software Installation.....	18
- System Menu Operation.....	18
- Step by Step Configuration.....	19
Hardware Setup.....	19
Gateway Configuration Menu.....	21
HDL-Bus Events.....	21
Modbus Addresses.....	22
HDL-Bus Actions.....	27
Gateway Parameters.....	29
<b>6. Ordering Information.....</b>	<b>30</b>

## 1. Product Information

Gateway for integration of HDL-Bus equipment into Modbus (RTU, UDP and TCP) control systems enables network communication.



By HDL-Bus to Modbus Gateway you can integrate any HDL-Bus control system into your BMS, SCADA, PLC ... "talking" Modbus.

Gateway acts as one more devices into the HDL-BUS system, maintaining always in memory an updated mirror of the Modbus points declared in its configuration table, ready to be served to Modbus, when requested. *Points in memory can be read and written from a Modbus master device.*

Modbus interface of the Gateway supports Modbus TCP, Modbus UDP and Modbus RTU, software

configurable, and acts as a Modbus slave device. *The Gateway's HDL-Bus interface connects directly to the bus or by UDP socket (default Port 6000).*

During whole the time of operation, the Gateway serves reading, into the memory, of current values for the required points. This is configurable per point, as configuration tables for filling in the process.

HDL-Bus interface supports serial and Ethernet communication, so it is one more device in HDL-Bus network.

The Gateway is multi-function device, because supports two modes - bridge and router (between each two from all the interfaces), according to selected configuration can be defined different functionality. Also, this unit can broadcast time messages to HDL-Bus network, because it has build-in real-time clock (RTC).

Gateway has additional functionality, like integrated 6 open collector (OC) outputs, which can be used to switch on/off any devices (lighting, heating, motor). By I2C bus can be connected more up to 8 relay modules with 16 relays each (8 open collectors + 8 free voltage changeover relays 10A each). All these relays can be retrieved directly from Modbus by using of appropriate coil register operations.

Gateway is configured using **HDL-Bus Gateway Setup**, software tool for Windows, which is supplied along with the purchase of Gateway with no additional cost.

## -Internal Database

The device contains an internal database that consists of areas for application data and configuration information.

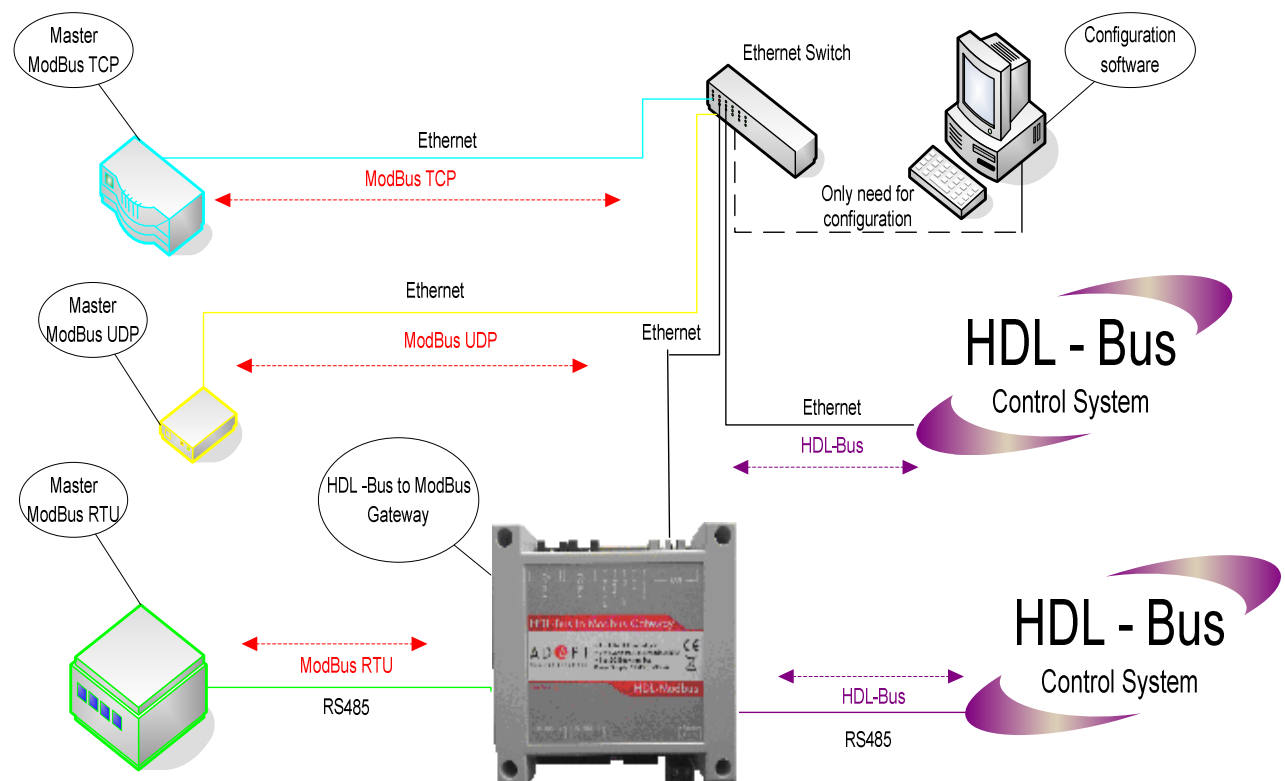
The internal database is shared between all ports on the module and is used, as a conduit to pass information from a device on one network to one or more devices on another network.

### Application Data Area

The data area is used to store and retrieve data by the protocol drivers and for data exchange between protocols. The database is used as a source for write commands to remote devices and holds data collected from the remote devices. Commands defined in the configuration file (stored in the configuration data area) control how the data is to be handled in the database.

### Configuration Data Area

This area contains module configuration information such as port configuration, network information, and command configuration. This configuration file is downloaded to this area from the configuration software.



## - Modbus Interface of Gateway

This interface support 250 events and 250 actions from/to different HDL-Bus devices (relays, sensors, dimmers and others). So, you can communicate between Modbus and HDL-Bus networks.

### Functions supported

- Modbus functions 01 and 02 (*Read Coils* and *Read Discrete Inputs*) can be used to read for bit access.
- Modbus functions 03 and 04 (*Read Holding Registers* and *Read Input Registers*) can be used to read Modbus registers.
- Modbus functions 05 (*Write Single Coil*) can be used to write for bit access.
- Modbus functions 06 and 16 (*Write Single Holding Register* and *Write Multiple Holding Registers*) can be used to write Modbus registers.
- If *poll records* are used to read or write more than one register, it is necessary that the range of addresses requested contains valid address, if not the corresponding Modbus error code will be returned.
- Modbus error codes are supported, they will be sent whenever a non valid Modbus action or address is required.

### Data Format

The Modbus registers are 2 bytes (16 bits), its content is expressed always in MSB..LSB.

### Modbus RTU

- Baud rate can be selected from 1200, 2400, 4800, 9600, 19200 and 38400. (Data Bits: 8, Stop Bits: 1).
- Parity can be selected from none, even and odd.
- Modbus slave number can be configured. Physical connection (RS485-1 or RS485-2) can also be selected, according cable connection.
- Error checking with CRC-16
- Only the line terminals A and B for RS485 connector are used

### Modbus TCP

- The TCP port to use can be configured (by default 502 is used). Up to 5 simultaneously TCP connections can be built.
- The TCP idle timeout can be configured.

### Modbus UDP

- The UDP port to use can be configured (by default 502 is used).

The only one IP address, subnet mask and default router address to use by Gateway can be also configured for both interfaces together – Modbus TCP and UDP.

### Address Map

The Modbus address map is fully configurable and any point in the Gateway can be freely configured with the desired Modbus register address.

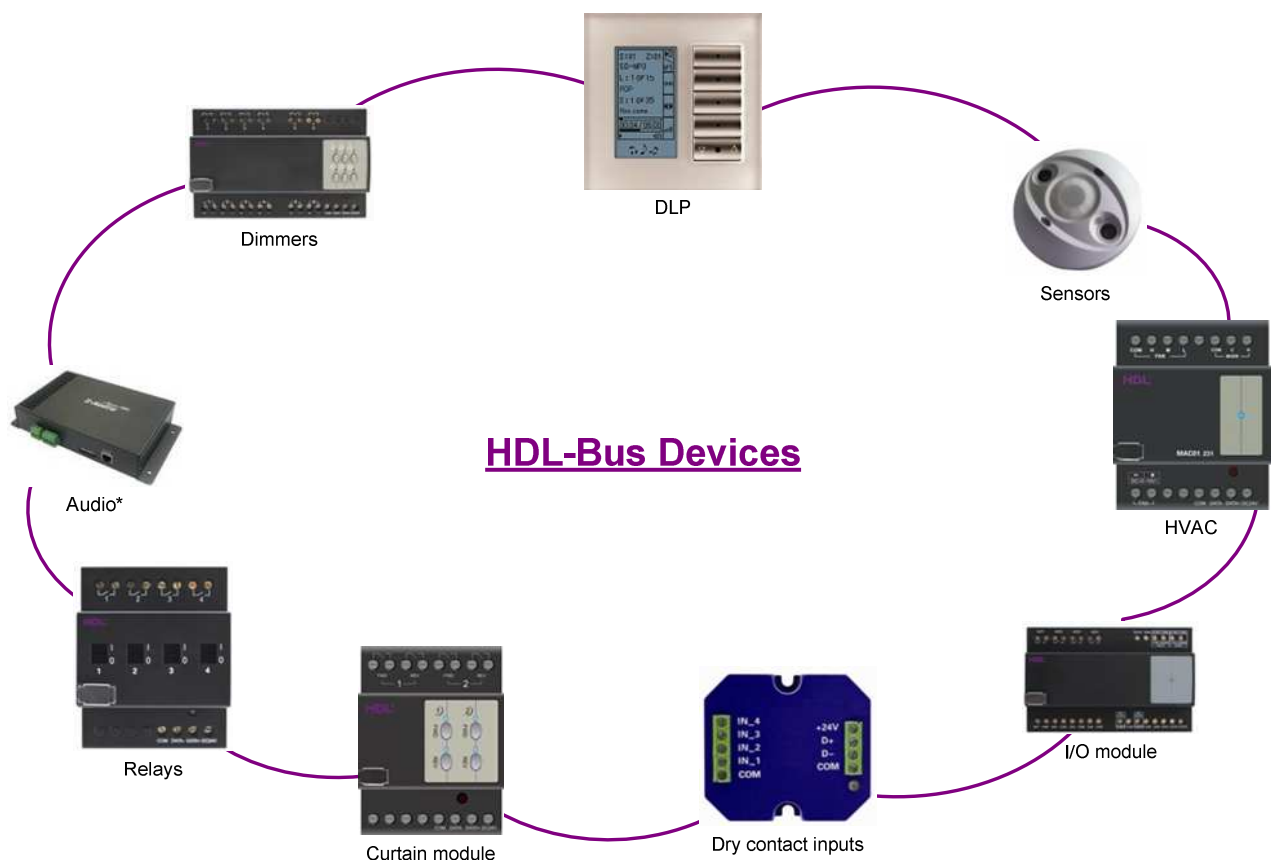
## - HDL-Bus Interface of Gateway

Gateway connects directly to the bus, thanks to its internal RS485 driver, and acts as one more device of the HDL-Bus system with the same behaviour and configuration characteristics.

The HDL-Bus interface supports 254 Universal Switches (US) - internal for the Gateway, which can be readable from Modbus interface by function 02. The setting of these US usually are done by other devices from HDL-Bus network, so they inform Modbus for their own status (for example - movement, contact and so on). It is enough to read their status, which can make any Modbus device, using the Gateway.

Gateway manages all the communication with the HDL-Bus according to its configuration.

- On receive telegram from HDL-Bus, the gateway updates its point's values in memory, ready to be served to Modbus when requested.
- When change in configured Modbus point is received, the corresponding write telegram will be sent to HDL-Bus.
- Gateway periodically (period can be configured) ask to read the status of the HDL-Bus devices and re-transmit the status of all the points.
- This interface support all types of HDL-Bus devices (scene controllers, sensors, HVAC and floor heating controller, inputs, motor controller).

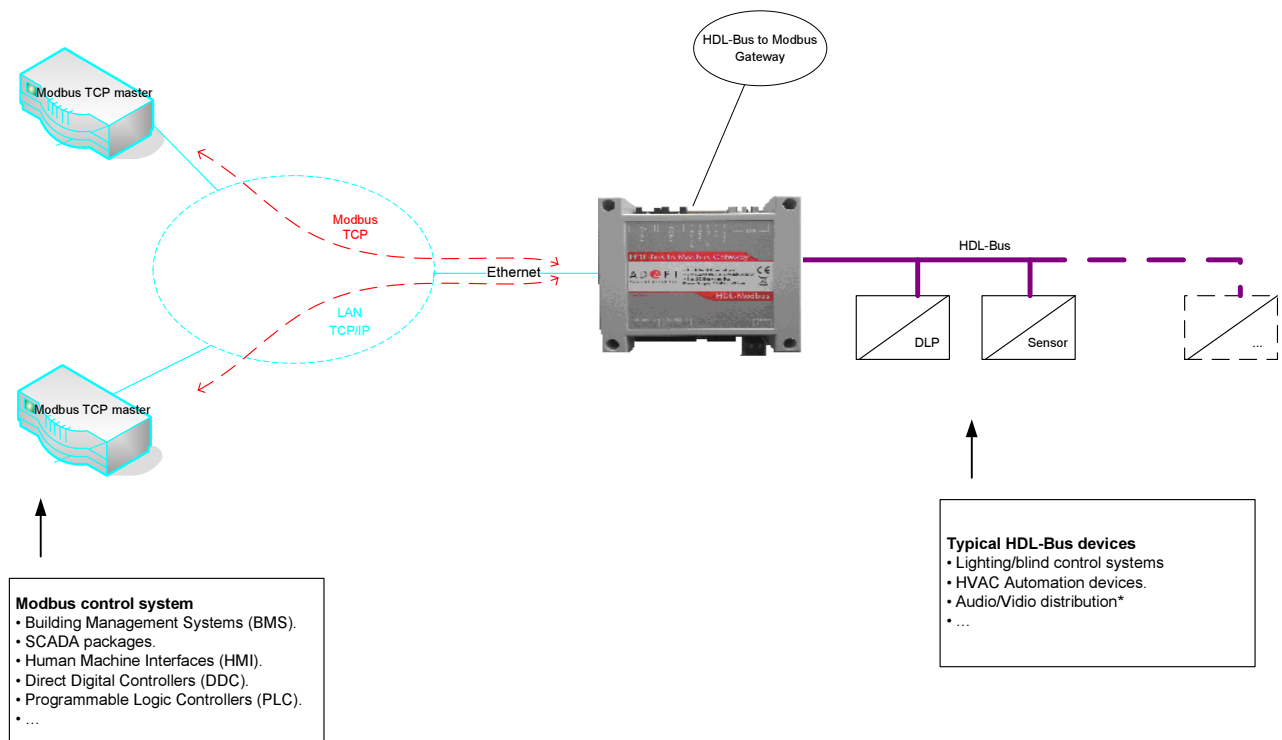


\* audio device's support is under development

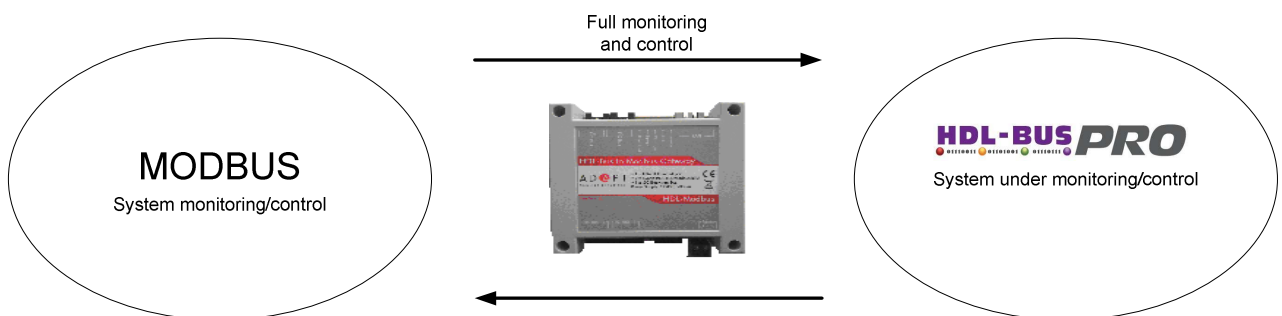
## 2. Application Examples

### - Full Monitoring and Control

(Integration of HDL-Bus and Modbus using HDL-Bus to Modbus Gateway)

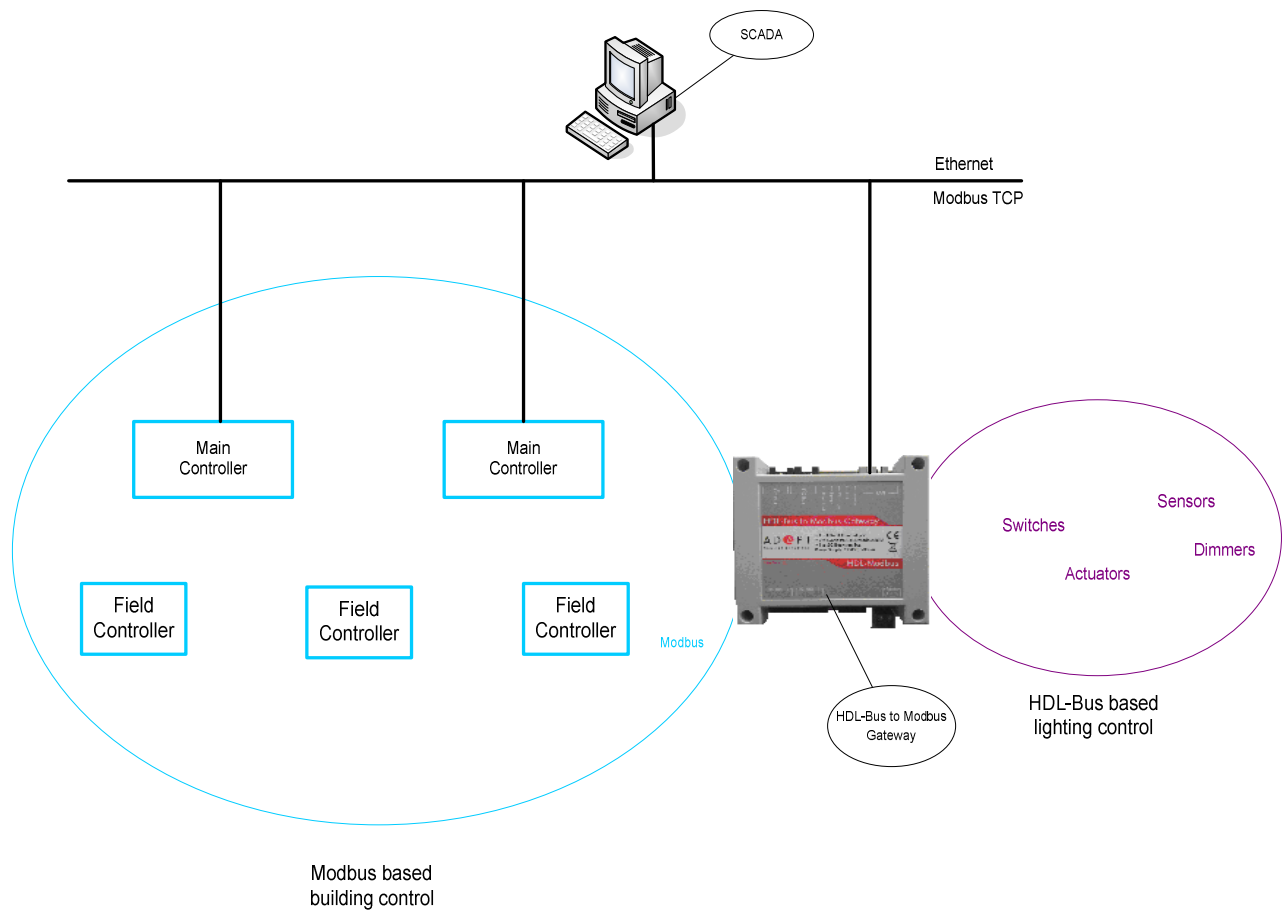


\* audio/video device's support is under development

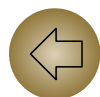


## - Full Integration

(Integrate HDL-Bus based lighting control into Building Management Systems "talking" Modbus)



**MODBUS**  
TCP/RTU



**HDL-Bus to Modbus**  
**Gateway**  
full integration



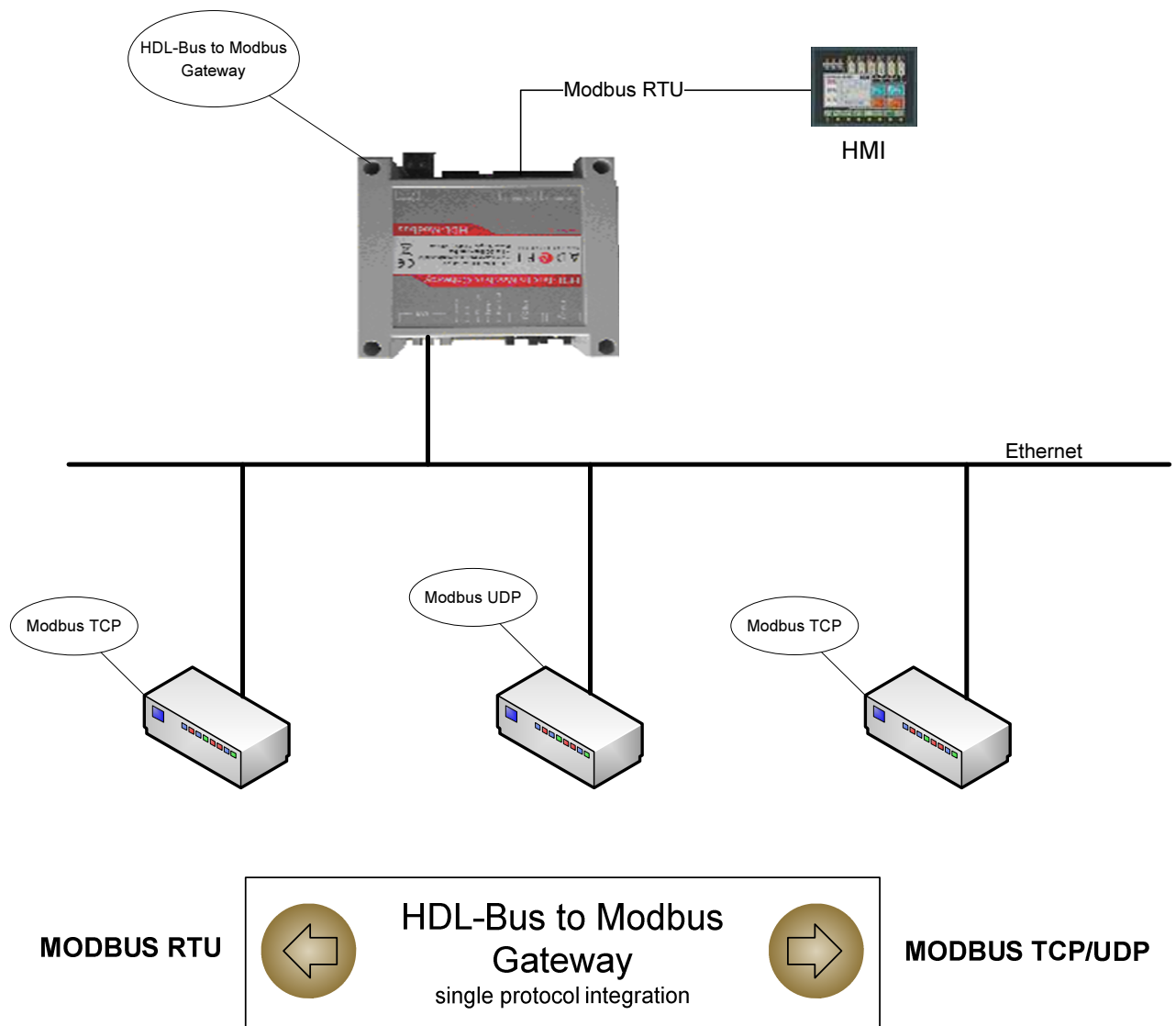
**HDL-BUS PRO**



## - Integrate serial Modbus to IP network

(Link the Modbus RTU devices with Modbus TCP/UDP devices\*)

HMI (Human Machine Interface) and many other systems use a Modbus RTU port and are required access to a discrete DCS (Distributed Control System). These days, however, more and more devices support Ethernet and run under Modbus TCP/UDP mode. In this case, HDL-Bus to Modbus Gateway is the even ideal solution to link the HMI (with serial interface) to Modbus TCP/UDP units, distributed over an Ethernet network.

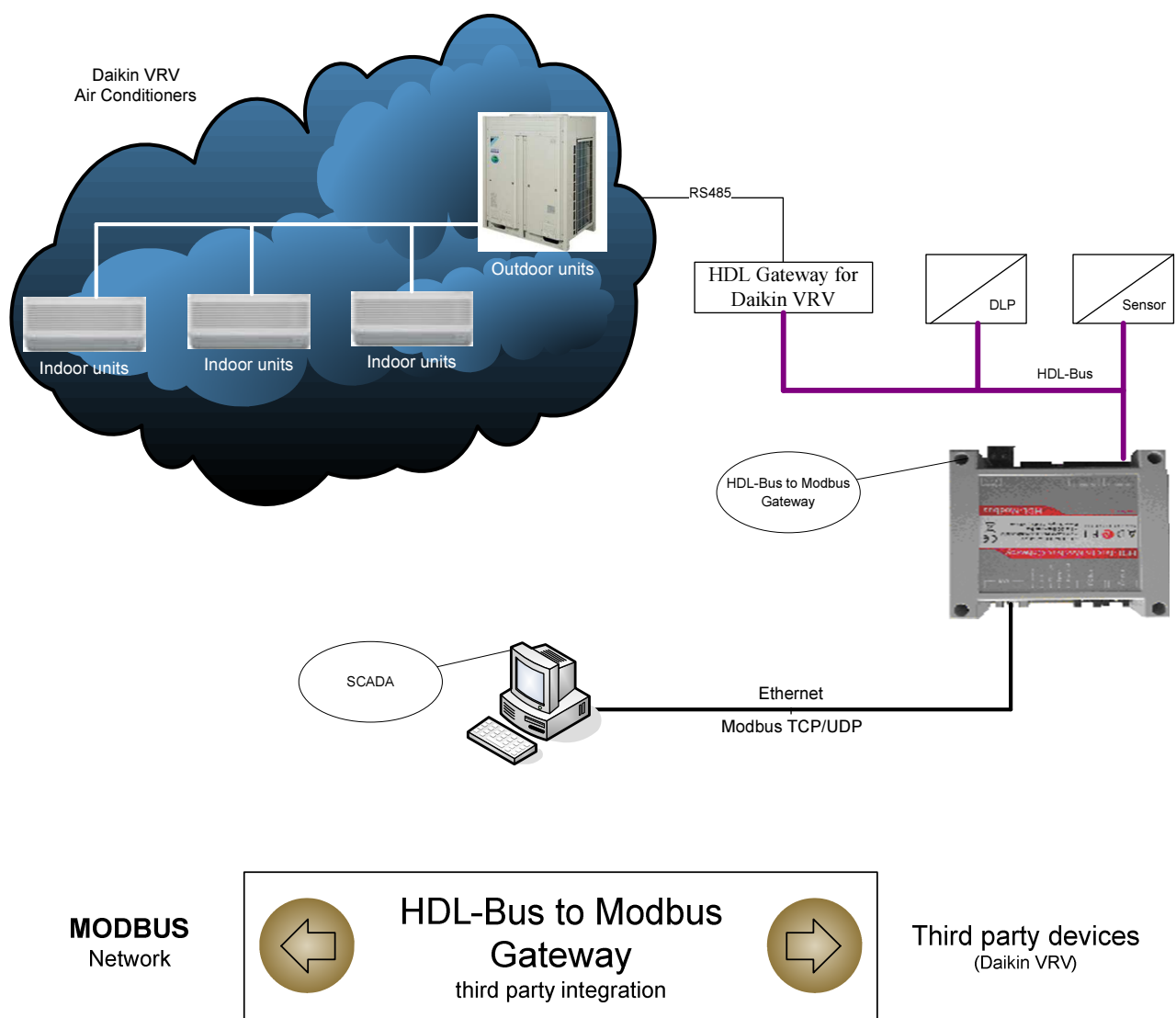


\*under development

### - Third Party Integration

Integrate third party devices (for example Daikin VRV Air Conditioners) into a Modbus enabled control system (BMS, PLC, SCADA, Controller...):

For this application, Daikin VRV Air Conditioning system must be equipped with HDL Gateway for Daikin VRV\* (use RS485 interface of Daikin), this gateway is normally commissioned by HDL partners, contact your nearest HDL distributor for details.



\* Gateway for Daikin VRV under development by HDL

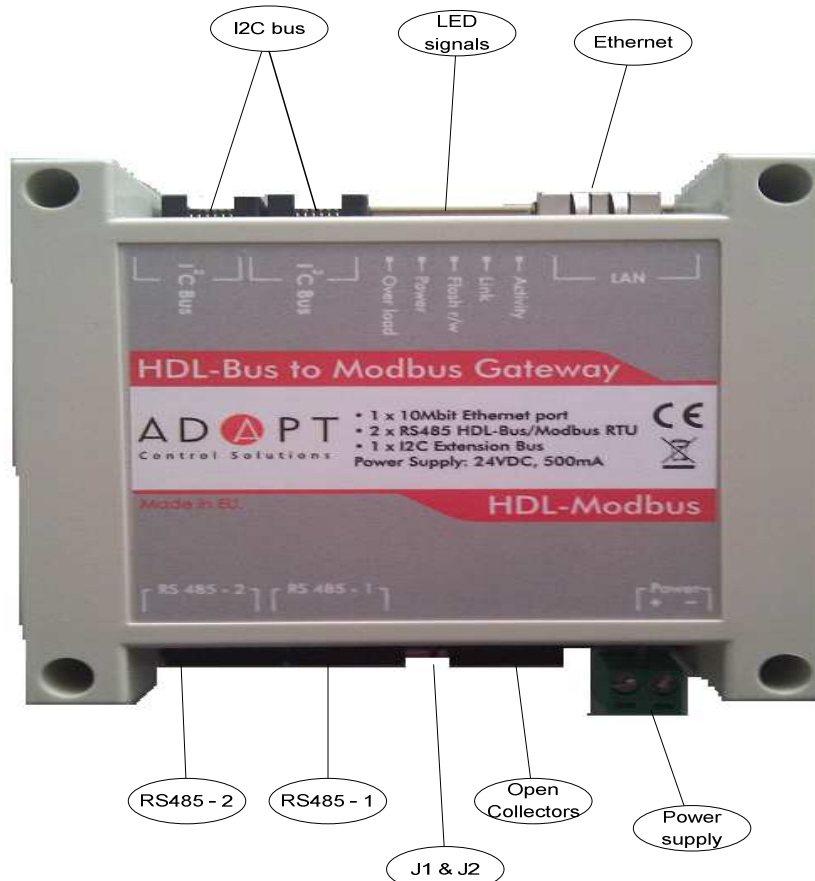
### 3. Hardware Information (signals, modes, diagrams, installation)

#### - Technical Data

- Supply voltage: 24VDC, plug-in terminal block for power connection (2 poles).
- Input current: 1A peak
- Input power: 6 Watt
- Fuse - resettable
- Protection IP20 (IEC60529).
- RoHS conformity: Compliant with RoHS directive (2002/95/CE).
- Certifications: CE
- Environmental characteristics
  - Operating temperature, humidity\*: 0°C ~ +50°C, < 50% RH
  - Storage temperature, humidity\*: -10°C ~ +70°C, < 40% RH
  - (\* non-condensing)

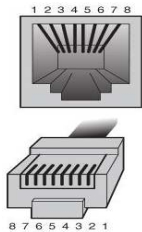
#### - On-board I/O

- 1 Ethernet 10Base: Modbus TCP/UDP connection, IP access to HDL-Bus (cover IP module functionality)
- 2 RS485 ports: HDL-Bus serial interface (+inject power supply for the line, up to 1A continues current per port), Modbus RTU connection
- I2C interface: extension option, by which can be connected up to 8 relay modules (16 relays each). Each relay can be controlled by Modbus Coil Register:
  - 0-127 Register → 1-128 relay.
- 6 relay outputs (OC). Each relay can be controlled by Modbus Coil Register:
  - 128-133 Register → 1-6 relay.
- RS485 line terminated (120Ω) via removable inside jumper for easy installation



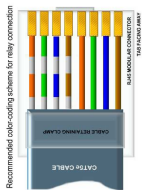
## - I/O Specification & Wiring

- 2 x RJ-45 terminals for all RS232/RS485 inputs and outputs



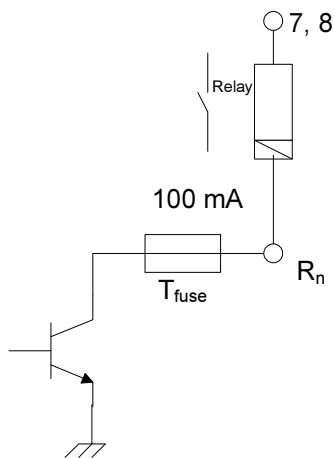
- 1 - GND (COM)
- 2 - RS232 (Rx)
- 3 - RS232 (Tx)
- 4 - HDL-Bus Data+ or Modbus RTU RS485+ (A)
- 5 - HDL-Bus Data- or Modbus RTU RS485- (B)
- 6 - GND (COM)
- 7 - +12/24VDC
- 8 - +12/24VDC

- RJ-45 terminal for Ethernet connectivity
- RJ-45 terminal for 6 OC:



- 1 - Open Collector I ( $R_1$ )
- 2 - Open Collector II ( $R_2$ )
- 3 - Open Collector III ( $R_3$ )
- 4 - Open Collector IV ( $R_4$ )
- 5 - Open Collector V ( $R_5$ )
- 6 - Open Collector VI ( $R_6$ )
- 7 - +12/24VDC
- 8 - +12/24VDC

\* the colors are only recommended



- 2 x RJ-12 connector for I2C link to connect external relay modules



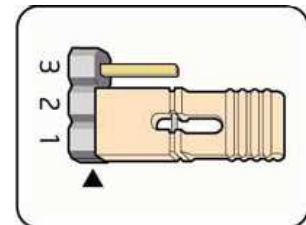
\* the colors are only recommended

## - LED signals

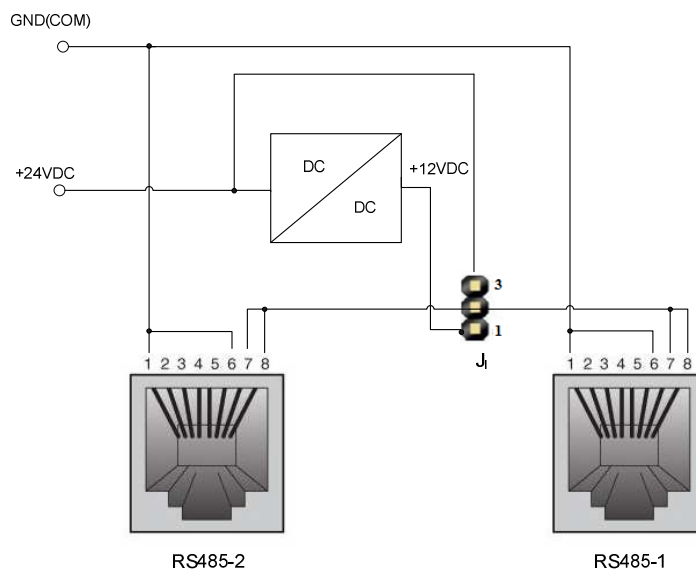
- Activity – Ethernet port activity
- Link – Ethernet port connection

- Flash r/w – configuration read/write
- Power
- Overload – I2C bus protection

### - Jumpers



J<sub>1</sub>J<sub>2</sub>

- J<sub>1</sub> (left - front connector view - see the picture) – choice of power for HDL-Bus/RS485 line:

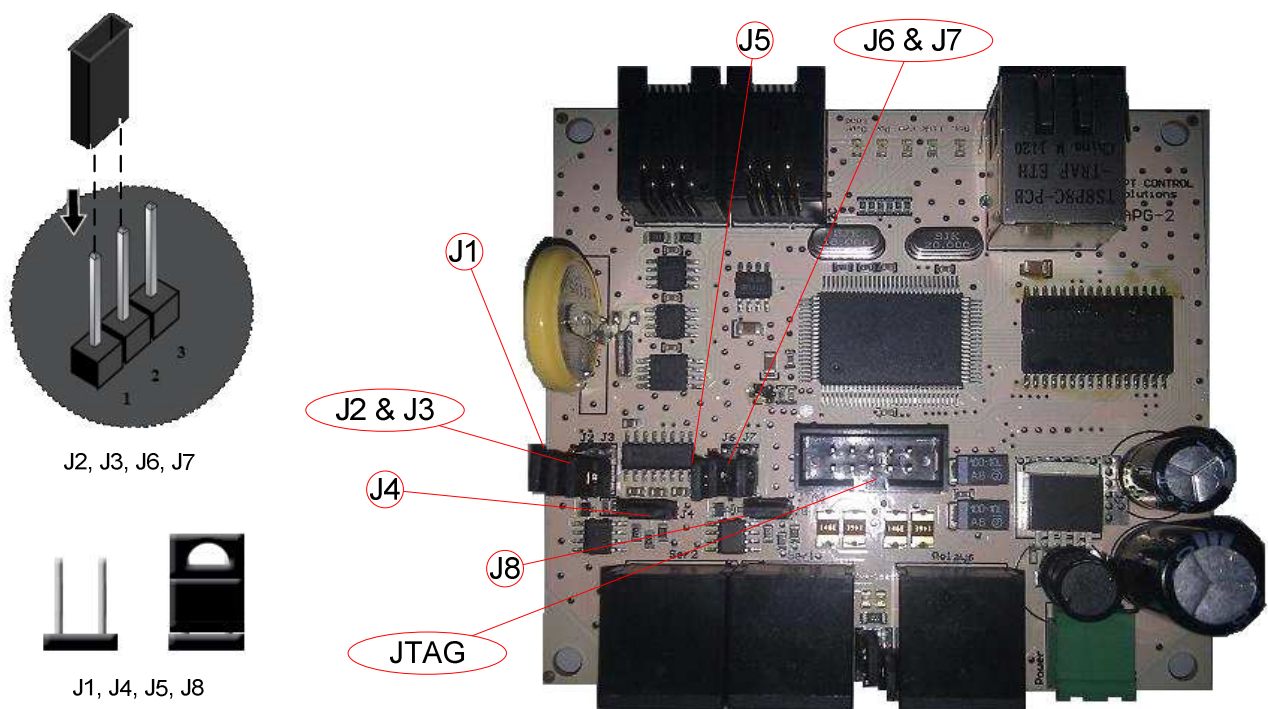


	1-2 = 12V (regulated) – for third party devices
	2-3 = 24V (pass-throw from DC power connector, can be used to supply power from the line contrariwise, also)
	open - does not supply power on the bus

- J<sub>2</sub> (right - front connector view - see the picture) – choice of power for relay modules:

	1-2 = 12V (regulated)
	2-3 = 24V (pass-throw from DC power connector)

Inside on the PCB are situated several jumpers for the serial interfaces configuration (see the PCB picture):



#### Serial interface 1 (RS485-1 labelled)

- J5 - must be closed

- J6 - RS485/RS232

	1-2 = RS485
	2-3 = RS232

- J7 - RS485/RS232


	1-2 = RS485
	2-3 = RS232

- J8 - 120Ω termination for long lines (open – not terminated, close – terminated line)



Serial interface 2(RS485-2 labelled)

- J1 - must be closed

- J2 - RS485/RS232

	1-2 = RS485
	2-3 = RS232

- J3 - RS485/RS232

	1-2 = RS485
	2-3 = RS232

- J4 – 120Ω termination for long lines (open – not terminated, close – terminated line)

Also, there is situated JTAG interface connector on the PCB, which can be used for firmware update, instead of LAN communication.

**- Dimensions & Weight:** 115mm x 89mm x 40mm, 0.100 kg

**- Enclosure:** Plastic, type PC (UL 94 V-0), light grey.



- **Mounting type:** standard DIN rail mount or 4 x screw fixing



#### **4. Safety instructions**

Make sure, you understand the function of each connection before you connect it.

See that all connections are made correctly before turning on the gateway.

Always disconnect the gateway from the power supply when connecting the signal leads, the power supply should be connected last.

Do not apply voltage higher than 30VDC to the gateway. (See also technical data)

In case of an emergency, you can disconnect power from the gateway by pulling out the green connector.

If your application set-up does not allow easy access to the power supply, install a mechanical circuit breaker in the supply-line, going to the gateway.

The gateway has a resettable fuse. In case of short circuit you need to break-up the power supply for a 15sec and turn it on again.

Take care with the environmental limits. Do not exceed them. (See technical data)



## 5. Software Configuration (setup, parameterization)

### - Features

#### Configure Gateway anywhere on the LAN

HDL-Bus to Modbus Gateway is powerful yet very easy to use. An intuitive Windows utility automatically searches for all available HDL-Bus to Modbus Gateway units on the LAN. Once Gateway is connected to an Ethernet network, any Windows host computer running **HDL-Bus Gateway Setup** can automatically search for all installed HDL-Bus to Modbus Gateways, and then proceed with the configuration. DHCP (Automatic IP settings) still is not supported. Only Manual IP is suggested to be used by configuration software.

#### Comprehensive and Powerful Modbus Parameter Settings

In addition to working with most Modbus devices, Gateway provides both basic and advanced communication parameters. For example, in addition to general configurations, Remote Modbus/TCP device ID, Initial Delay Time Adjustment, and Character/Message Timeouts are adjustable right in the Windows utility.

HDL-Bus_Gateway_Setup	<ul style="list-style-type: none"> <li>• Visual engineering tool, easy of use, for gateway's configuration and supervision compatible with Microsoft Windows operating systems, supplied with the gateway free of charge.</li> <li>• Multi-window tool allowing to prepare simultaneously parameters for communication activity with both protocols (systems) and configuration windows to configure all the gateway's parameters and signals.</li> <li>• Signals configuration in XML files for easy and quick configuration (very useful in projects with a lot of points).</li> <li>• Allows configuring the gateway's parameters and signals while in off-line (not connected to the gateway).</li> <li>• Connection to the gateway for download the configuration and supervision by using LAN interface of the PC.</li> <li>• Upgrades for this software tool available free of charge.</li> <li>• Firmware update by LAN interface.</li> <li>• Network settings (IP, mask, gateway IP)</li> <li>• Modbus TCP/UDP port change</li> <li>• HDL-Bus port change (default UDP 6000)</li> <li>• Each interface individual Subnet ID – important in router mode</li> <li>• Gateway mode selection – bridge or router</li> </ul>
-----------------------	---

## - Software Installation

The software installation is very easy. You need only to start Setup.exe (supplied together with your Gateway) and rest of process will be done automatically, as installation procedures will guide you through the process.

## - System Menu Operation

The configuration software has toolbar with several items:

### File

Here you can start configuring new Gateway or load already prepared configuration. After finish of process you can save file and leave the program.

### Tools

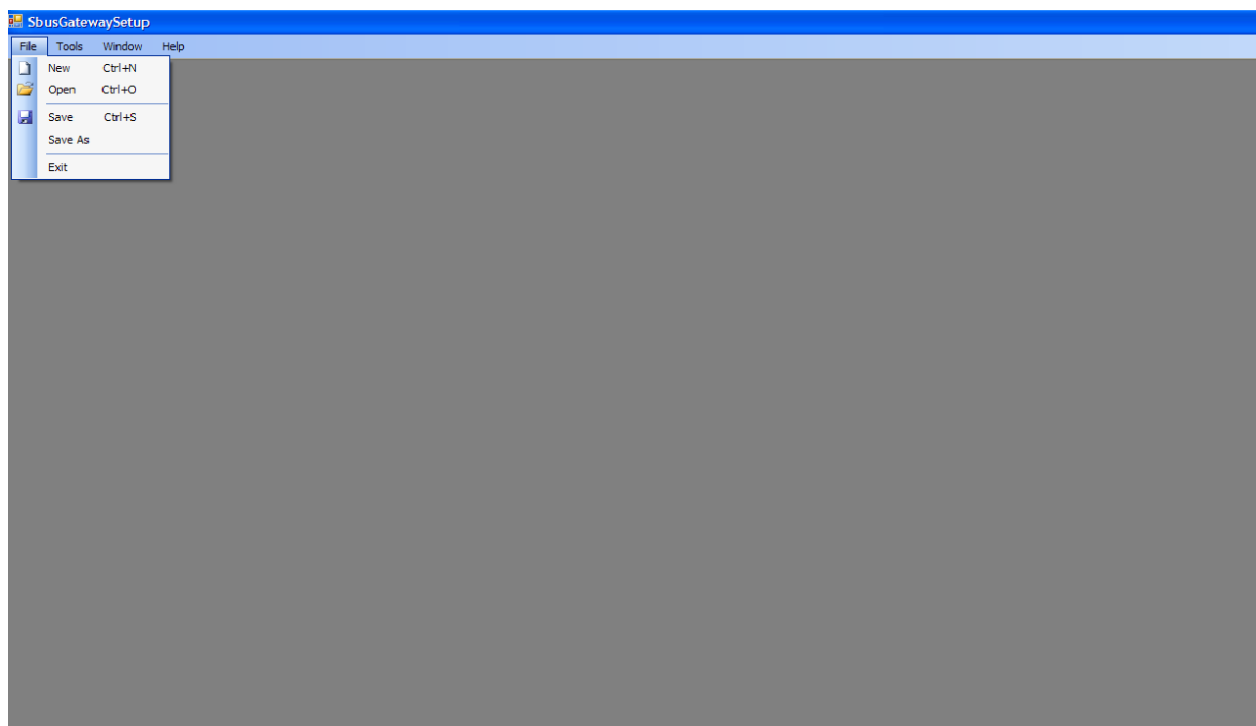
This menu offer Hardware Setup, where you can operate with your Gateway online.

### Windows

By this item you can see and choice opened windows.

### Help

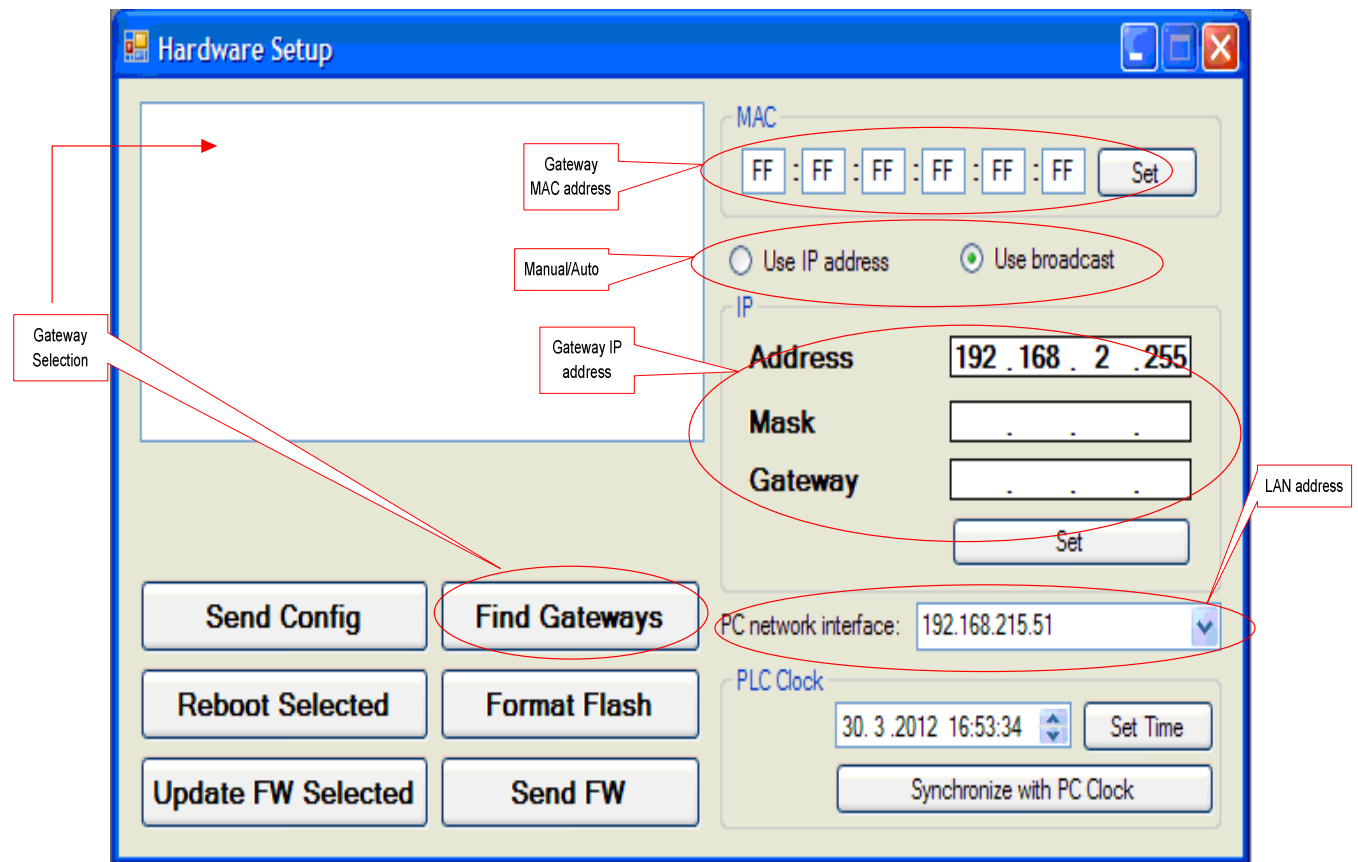
Offers help information.



## - Step by step configuration

### Hardware Setup

First operation, which must be done after you are on-line, it is to setup the interfaces for your computer, as choice from installed IP addresses. So you must go to Tools menu and choice Hardware Setup sub-menu. After it you must select "PC network interface" address (see the picture) from the list:

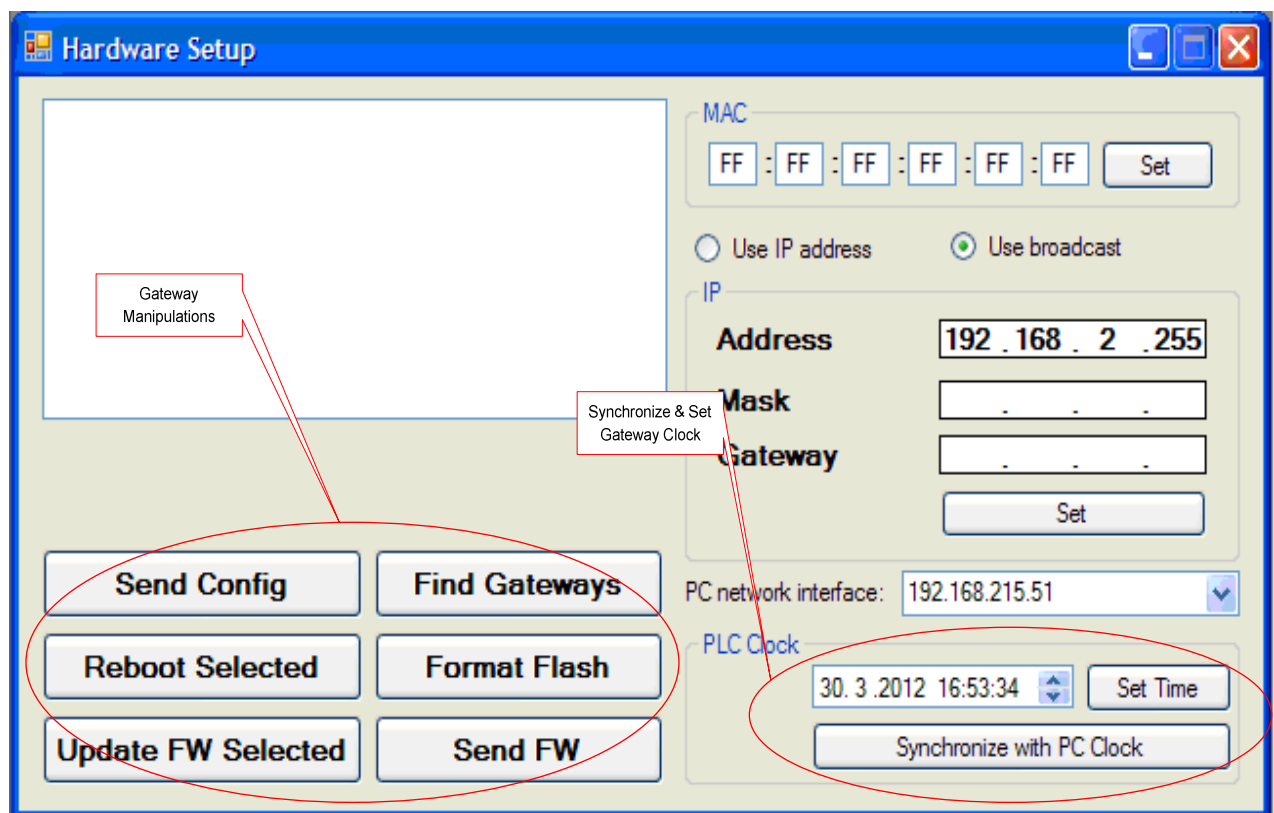


Concerning configuring of your Gateway address, you have two possibilities – automatic and manual, respectively by selecting "Use broadcast" or "Use IP address". You can find and select your device by "Find Gateway" button in both cases. The found modules will be shown in the window above the buttons and after this you can select desired one. The information includes the MAC address of the module and its corresponding IP address, followed by a random number and status of the compiled module configuration, that can be OK or Error followed by error code. Please note, sometimes must wait few minutes before appear Gateway's detail in the window, depending of your LAN traffic and size. For automatic search you must choice "Use broadcast" before hit the button. For second option, you must put Gateway parameters (address, mask and IP gateway) manually before select "Use IP address" and use "Find Gateway" button, by this option you can find IP address or search the specified subnet (for example 192.168.2.255 will define search in area of all the addresses from 192.168.2.00 to 192.168.2.254). Please keep in mind, that if there is a VPN connection between the computer with configuration software and the network, where is installed the Gateway, then will need to be addressed only manually devices,

because the automatic search of the address will not work. If you need to change IP settings for already selected device, then you can use the same parameter's fields (only available after device selection) to put desired network's details. To save the changes, you must hit button "Set", below them. After changing the IP address of the Gateway is better to press the "Find Gateways" and continue the remaining settings with its new IP address (of course, after selection from newly found device list).

**Important!** The Gateway does not support DHCP server using static IP address is required.

Also, you can easy change the MAC address of the selected device – as is shown on the above picture.



By Hardware Setup menu, you can operate with the selected Gateway. You can clear and format Gateway's flash memory. In the flash memory are stored Gateway's configuration files and tables, describing the relationship between ModBus addresses and HDL-Bus devices. Other operation is sending and loading already prepared configuration toward the selected Gateway by button "Send Config". It is recommended, before loading each new configuration to, the flash memory is erased. Delete it, by pressing "Format Flash". You can require reboot of Gateway by button "Reboot Selected", for example after sent new configuration, it is necessary.

You have two possibilities to update firmware of the selected Gateway. First one is remotely and on one step by button "Update FW Selected". The second procedure requires presence of local access to the device. You should connected special dongle in I2C interface (RJ-12 with given short 2<sup>nd</sup> and 3<sup>rd</sup> pin) before sending firmware by "Send FW" button. The last procedure could be

used, if you have error after finishing of remote update due to communication problems, for example.

You can change Real Time Clock setting of your Gateway. For this purpose, select the desired module and verified his clock with that of your computer by pressing “Synchronize with PC Clock”, or by manually entering the date, time and year followed by pressing “Set Time”. Additional functionality of the Gateway is broadcasting of Real Time Clock information trough HDL-Bus network. So, any network device (like DLP) can use this information according his purposes.

### Gateway configuration menu

In Gateway configuration menu you have 3 sections: HDL-Bus events, HDL-Bus actions, Gateway parameters. The first two tables explain relations between HDL-Bus and Modbus details and the last one – parameters, describing the Gateway’s interfaces. To reach this menu you must enter File menu and after it you have two ways: by New (to start new configuration) or by Open sub-menu (to open already prepared configuration):

Event ID	Device Type	Extra Property	HDL-Bus Subnet ID	HDL-Bus Device ID	Modbus register start address	Modbus register end address
1	\$0001 : HDL-SC (Scene Controller)	8 chanel	1	9	1	8
2	\$0002 : HDL-DLP-AC (Panel Aircondition Controller)	AHML CHFA*	1	3	9	13
3	\$0003 : HDL-DLP-FH (Floor Heating Controller)	Don't capture	1	3	14	18
4	\$0006 : HDL-MovSen (Movement Sensor & Dry Contact (1 Reg))		1	33	19	19
5	\$0004 : HDL-THL (THL Sensor)		1	7	20	24
6	\$0005 : HDL-AQ (Air Quality Sensor)		1	6	25	25
7	\$0006 : HDL-MovSen (Movement Sensor & Dry Contact (3 Regs))		1	11	26	28
8	\$0009 : HDL-CLK (Real time clock (6 Regs))		1	12	27	32
9	\$0008 : HDL-LUX (Lux Sensor)		1	18	33	33
10	\$0009 : HDL-CLK (Real time clock (3 Regs))		1	2	34	36
11	\$0007 : HDL-MS04 (4 dry contact sensor (4 Regs))		1	19	37	40
12	\$0000 : HDL-Null (No device)					
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						

### HDL-Bus events

“HDL-Bus events” table can contains up to 250 events. Here you must describe for each HDL device (defined by Subnet ID and Device ID, as HDL-Bus address in the network) corresponding Modbus addresses. So, in arising of event issued from this device, then on relevant Modbus addresses will be written some information, concerning this event. For each device you have choice of next types:

\$0000: HDL-Null (No device)

\$0001: HDL-SC (Scene Controller)

\$0002: HDL-DLP-AC (Panel Air Condition Controller)  
 \$0003: HDL-DLP-FH (Panel Floor Heating Controller)  
 \$0004: HDL-THL (THL Sensor)  
 \$0005: HDL-AQ (Air Quality Sensor)  
 \$0006: HDL-MovSen (Movement Sensor & Dry Contact (1 Reg))  
 \$0006: HDL-MovSen (Movement Sensor & Dry Contact (3 Regs))  
 \$0007: HDL-MS04 (4 dry contact sensor (1 Reg))  
 \$0007: HDL-MS04 (4 dry contact sensor (4 Regs))  
 \$0008: HDL-LUX (LUX Sensor)  
 \$0009: HDL-CLK (Real time clock (3 Regs))  
 \$0009: HDL-CLK (Real time clock (6 Regs))

Some of Device Types have extra specific properties to be selected:

Scene Controller – up to 48 channels (different for relays, dimmers, DMX);

Panel Air Condition Controller – set fan speeds and modes:

Fan speed: Auto, High, Medium, Low

Mode: Cool, Heat, Fan, Auto, Dry

Panel Floor Heating Controller – capture broadcasted air-condition room temperature.

Modbus addresses in the table must be defined only as start address and after that corresponding quantity of registers according selected device type will be reserved. The information, stored in these registers, is according to the different type HDL-Bus devices, which addressed it.

### Modbus addresses

By Modbus's functions you could read/write next data:

#### **- Scene Controller**

(area information and channel status/value)

For this type will be reserved so many the addresses, as are the channels of the controller.

For example, if you have 6-channel dimmer, then will be reserved 6 registers (one for each channel).

#### **Modbus functions 04 (Read Input Registers):**

The high byte of each register describes one area and the low byte describes channel status and value:

Bits:	Area info(high byte)	Channel Info(low byte)
	15, 14, 13, 12, 11, 10, 9, 8	7, 6, 5, 4, 3, 2, 1, 0
Register 0	Area 0	Channel 0
Register 1	Area 1	Channel 1
.....	.....	.....
Register N	Area N	Channel N

The area information is active scene in this area. Special scene numbers are:

- 0 - all channels in this area are OFF;
- 254 - unspecified scene - i.e. after activating a scene, it has been changed by manual change of some channel (individual command for specific channel)

The channel information is channel status and value:

- bits 0...6 – store channel value (0...100) - information in this cell will be refreshed after executing of HDL-Bus commands “channel set” (0x0031), only for specified channel or “get channels status” (0x0033);

- bit 7 – store channel status ON/OFF (1/0) - information in this bit will be refresh after executing of HDL-Bus commands “scene set” (0x0002) or “channel set” (0x0031) and also from broadcast message (0xEFFF).

Please, note due to the nature of "get channel status" (0x0033) command (it is not executing in regular system work, without external initiative), the information in the above cell may be outdated. So, to have refreshed just need to be forced to execute "get channel status" (0x0033).

#### **Modbus functions 03 and 06 (Read Holding Registers and Write Single Holding Register):**

Read/write channel status value (0...100) by these functions in the whole registers:

Register 0	Channel 0 Status Value
Register 1	Channel 1 Status Value
.....	.....
Register N	Channel N Status Value

#### **- Panel Air Condition (AC) Controller (from DLP)**

(AC controller status)

This type covers 5 consecutive Modbus registers (10 bytes) for status information (first 4 for old DLP and all 5 for new DLP):

#### **Modbus functions 04 (Read Input Registers):**

Each byte from the registers has specific description:

Byte 0 (Register 0 - high byte)	AC controller on/off (0 - off, 1 - on)
Byte 1 (Register 0 - low byte)	cool setpoint
Byte 2 (Register 1 - high byte, bits 0...3)	set fan speed 0-Auto, 1-High, 2-Med, 3-Low
Byte 2 (Register 1 - high byte, bits 4...7)	set mode 0-Cool, 1-Heat, 2-Fan, 3-Auto, 4-Dry
Byte 3 (Register 1 - low byte)	lock/unlock panel (0 - lock, 1 - unlock)
Byte 4 (Register 2 - high byte)	room temperature
Byte 5 (Register 2 - low byte)	heat setpoint
Byte 6 (Register 3 - high byte, bits 0...3)	actual fan speed 0-Off, 1-High, 2-Med, 3-Low
Byte 6 (Register 3 - high byte, bits 4...7)	actual mode 0-Cool, 1-Heat, 2-Fan, 4-Dry
Byte 7 (Register 3 - low byte)	auto setpoint
Byte 8 (Register 4 - high byte)	dehumidifier temperature set (only for new DLP)
Byte 9 (Register 4 - low byte)	Unknown

#### **Modbus functions 03 and 06 (Read Holding Registers and Write Single Holding Register):**

Read/write status information by these functions in the whole registers:

Register 0	AC controller on/off (0 - off, 1 - on)
Register 1	room temperature
Register 2	actual setpoint
Register 3	set fan speed 0-Auto, 1-High, 2-Med, 3-Low
Register 4	set mode 0-Cool, 1-Heat, 2-Fan, 3-Auto, 4-Dry

#### **- Panel Floor Heating (FH) Controller (from DLP)**

(FH controller status)

For this type will be reserved 5 consecutive Modbus registers (10 bytes) for status information:

#### **Modbus functions 04 (Read Input Registers):**

Each byte from the registers has specific description:



Byte 0 (Register 0 - high byte)	C/F (0 - Celsius, 1 – Fahrenheit)
Byte 1 (Register 0 - low byte)	FH room temperature
Byte 2 (Register 1 - high byte)	FH controller on/off (0 - off, 1 - on)
Byte 3 (Register 1 - low byte)	set mode: 1-Normal, 2-Day, 3-Night, 4-Away, 5-Timer
Byte 4 (Register 2 - high byte)	Normal setpoint
Byte 5 (Register 2 - low byte)	Day setpoint
Byte 6 (Register 3 - high byte)	Night setpoint
Byte 7 (Register 3 - low byte)	Away setpoint
Byte 8 (Register 4 - high byte)	if set mode == 5(Timer): 0 - Day, 1 - Night, 0xF5 - invalid*
Byte 9 (Register 4 - low byte)	Not used

\* Byte 8 (Return status) supported only in some versions of the firmware.

#### **Modbus functions 03 and 06 (Read Holding Registers and Write Single Holding Register):**

Read/write status information by these functions in the registers:

Register 0	FH controller on/off (0 - off, 1 - on)
Register 1	FH room temperature
Register 2	actual setpoint
Register 3	set mode: 1-Normal, 2-Day, 3-Night, 4-Away, 5-Timer
Register 4	actual mode: 1-Normal, 2-Day, 3-Night, 4-Away

#### **- THL Sensor**

(sensors and contacts status)

This type requires 5 consecutive Modbus registers (10 bytes) for status information:

#### **Modbus functions 04 (Read Input Registers):**

Each byte from the registers has specific description:

Byte 0 (Register 0 - high byte)	Measured temperature + 20
Byte 1 (Register 0 - low byte)	Illumination in Lux (high byte)
Byte 2 (Register 1 - high byte)	Illumination in Lux (low byte)
Byte 3 (Register 1 - low byte)	relative humidity in %
Byte 4 (Register 2 - high byte)	Unknown (0xFF)
Byte 5 (Register 2 - low byte)	Unknown (0xFF)
Byte 6 (Register 3 - high byte)	Dry contact 1, 0 - open, 1 - close
Byte 7 (Register 3 - low byte)	Dry contact 2, 0 - open, 1 - close
Byte 8 (Register 4 - high byte)	Unknown (0x00)
Byte 9 (Register 4 - low byte)	Unknown (0x00)

Please note, very important feature for the THL sensor - broadcasting its temperature in HDL-Bus network, even without request from other network device. The rest of information will be updated, as reply of HDL-Bus command "Query 8in1 movement & CC" (0xDB00), after request from defined actions, which could be activate even cyclic by definition of "Cycle period" in seconds.

#### **Modbus functions 03 (Read Holding Registers):**

Read status information by this function in the registers:

Register 0	temperature
Register 1	Illuminance in Lux
Register 2	relative humidity in %
Register 3	Dry contact 1 state
Register 4	Dry contact 2 state



Another approach for example, is to be uses some internal Universal switch to forward information about motion detection in the supervised area. For this purpose, must be defined some very simple logic in the sensor, which set US after motion detection and in reaching of some level of the LUX, also. If the sensor did not detect some movement for defined period, then the same US (the Gateway has 250 US) will be cleared. So, any Modbus device can read this US (by function 02) and have information about this sensor without any activity in the HDL-Bus network.

#### **- Air Quality Sensor**

(sensor status)

For this type will be reserved only one Modbus register for status information:

#### **Modbus functions 03 and 04 (Read Holding Registers and Read Input Registers):**

Register 0 – low byte    

measured air quality value
----------------------------

The information in the byte will be updated, as reply of HDL-Bus command “Query AQ sensor” (0x1646). The status value could be one of next:

- 0 - good air quality
- 01 - mild concentration
- 02 - medium-polluted
- 03 - heavy pollution.

#### **- Movement Sensor**

(sensor and inputs status)

#### **Modbus functions 04 (Read Input Registers):**

This type requires only one Modbus register for status information:

Register 0 – low byte    

bit mask:
bit 0 = input 1
bit 1 = input 2
bit 2 - unknown
bit 3 - in movement
bit 4..7 - unknown

For refresh of the information just need to be forced to execute "Query Movement sensor" (0xDB00) HDL-Bus command. By setting of “Cycle period” could be updated this information periodically over the defined period. The value of bit 3 could be one of next:

- 0 - no movement
- 1 – movement.

#### **Modbus functions 03 (Read Holding Registers):**

Read status information by this function in 3 Modbus registers:

Register 0	<table border="1"><tr><td>in movement</td></tr></table>	in movement
in movement		
Register 1	<table border="1"><tr><td>Dry contact 1 state</td></tr></table>	Dry contact 1 state
Dry contact 1 state		
Register 2	<table border="1"><tr><td>Dry contact 2 state</td></tr></table>	Dry contact 2 state
Dry contact 2 state		

#### **- 4 Dry Contact Sensor**

(contacts status)

#### **Modbus functions 04 (Read Input Registers):**

For this type will be reserved only one Modbus register for status information:

Register 0 – low byte

bit mask:
bit 0 = input 1
bit 1 = input 2
bit 2 = input 3
bit 3 = input 4
bit 4..7 - unknown

The information in the byte will be updated, as reply of HDL-Bus command “Query Switch sensor” (0x012C). The status value could be one of next:

0 - open

1 - close.

### **Modbus functions 03 (Read Holding Registers):**

Read status information by this function in 4 Modbus registers:

Register 0	Dry contact 1 state
Register 1	Dry contact 2 state
Register 2	Dry contact 3 state
Register 3	Dry contact 4 state

### **- LUX Sensor**

(sensor status)

This type requires only one Modbus register for status information:

### **Modbus functions 03 and 04 (Read Holding Registers and Read Input Registers):**

Register 0 – low byte	measured Illuminance in Lux (low byte)
Register 0 – high byte	measured Illuminance in Lux (high byte)

For refresh of the information just need to be forced to execute "read Lux sensor" (0xD992) HDL-Bus command. The status value could be in range 0 – 5000.

### **- Real time clock**

(Real Time Clock information)

### **Modbus functions 04 (Read Input Registers):**

For this type will be reserved 3 consecutive Modbus registers (6 bytes) for data/time information. Each byte from the registers has specific description:

Byte 0 (Register 0 - high byte)	Years start from 2000
Byte 1 (Register 0 - low byte)	Month 1...12
Byte 2 (Register 1 - high byte)	Day 1..Monthdays
Byte 3 (Register 1 - low byte)	Hour
Byte 4 (Register 2 - high byte)	Minute
Byte 5 (Register 2 - low byte)	Second

### **Modbus functions 03 (Read Holding Registers):**

This type requires 6 consecutive Modbus registers. Read data/time information by this function in the registers:

Register 0	Years start from 2000
Register 1	Month 1...12
Register 2	Day 1..Monthdays
Register 3	Hour
Register 4	Minute
Register 5	Second

### HDL-Bus actions

The second section is “HDL-Bus actions” table, which contains up to 250 pre-defined actions (identified by “Action ID”) initiated from Modbus toward HDL-Bus devices:

Action ID	Next Action	Modbus register start address	HDL-Bus Subnet ID	HDL-Bus Device ID	HDL-Bus Command	Raw Command Length	Parameter 1	Parameter 2	Parameter 3	Cycle period [sec]
1	0	10000	1	9	Single channel lighting control ( \$0031 ) ( P1, M1, P3 )	1			0	
2	0	10001	1	9	Query Scene Controller ( \$0033 ) ( No Parameters )					10
3	0	10002	1	18	Query Lux sensor ( \$D992 ) ( No Parameters )					10
4	0	10003	1	7	Query Switch sensor( \$012C ) ( No Parameters )					10
5	0	10004	1	18	Query 8in1 movement & CC ( \$DB00 ) ( No Parameters )					10
6	0	10005	1	3	Query AC Controller ( \$E0EC ) ( No Parameters )					10
7	0	10006	1	3	Query Floor Heating Controller ( \$1944 ) ( No Parameters )					10
8	0	10007	1	12	Raw Command (Raw)	10				0
9	0	10008	1	19	Invalid switch ( No Parameters )					
10	0	10009	1	16	Invalid switch ( No Parameters )					
11	0	10010	1	17	Invalid switch ( No Parameters )					0
12	0	10011	1	18	Invalid switch ( No Parameters )					
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										

By this table, it is possible to be activated only single action or chain of actions defined by field “Next Action”. Also, any action can be started automatically each period defined in the field “Cycle period” in seconds.

“Modbus register start address” defines the beginning of the memory area, used by concrete action for data transfer.

The corresponding “HDL-Bus Commands” are combination of function and parameters. The actions could be defined by next functions list:

- Invalid switch
- Scene switch ( \$0002 )
- Sequence switch ( \$001A )
- Universal switch ( \$E01C )
- Single channel lighting control ( \$0031 )
- Curtain switch ( \$E3E0 )
- Panel control ( \$E3D8 )
- Query AC Controller ( \$E0EC )
- Query Scene Controller ( \$0033 )
- Query 8in1 movement & CC ( \$DB00 )

Query Switch sensor( \$012C )  
 Query Floor Heating Controller ( \$1944 )  
 Query Lux sensor ( \$D992 )  
 Query air quality sensor ( \$1646 )  
 Raw Command

They should be combination with next parameters, according format of desired HDL-Bus command:

No Parameters  
 P1, P2  
 M1, P2  
 P1, M1  
 M1H, M1L  
 M1, M2  
 P1, P2, P3  
 M1, P2, P3  
 P1, M1, P3  
 P1, P2, M1  
 M1H, M1L, P3  
 P1, M1H, M1L  
 M1H, M1L, M2  
 M1, M2, P3  
 P1, M1, M2  
 M1, M2, M3  
 Raw

Pn – corresponds to each individual parameter from the fields “Parameter 1”, “Parameter 2” and “Parameter 3”, pre-defined to be used for HDL-Bus format;

Mn – describes individual register from the sequences of Modbus registers received for this action;

MnH and MnL – corresponds to high and low part of the register.

With “Raw Command” with parameter “Raw”, could be defined HDL-Bus command, as an action, directly from Modbus devices after write in appropriate Modbus registers the information in HDL-Bus format. The length of this kind of command must be defined in the field “Raw Command Length” in bytes.

The power of using from Modbus for the above commands could be illustrated with one trivial example - control of the lighting scene:

- I<sup>st</sup> possibility - fixed zone and fixed scene: Scene switch ( \$0002 ) (P1, P2);
- II<sup>nd</sup> possibility - fixed zone and Modbus defined scene: Scene switch ( \$0002 ) (P1, M1);
- III<sup>rd</sup> possibility - Modbus defined zone and scene: Scene switch ( \$0002 ) (M1, M2);
- IV<sup>th</sup> possibility -The same like III<sup>rd</sup>, but using only one register instead of two: Scene switch ( \$0002 ) (M1H, M2L).

“Subnet ID” and “Device ID” fields define the HDL-Bus device, to which is directed for implementation desired action.

By fields “Parameter 1”, “Parameter 2” and “Parameter 3” could be pre-defined parameters for HDL-Bus command.

### Gateway Parameters

The last section from Gateway configuration menu is “Gateway Parameters”. Here you must define parameters describing each interface - Modbus TCP, Modbus UDP, HDL-Bus and Modbus RTU ports. Also, in this table you could choose “Gateway mode” – Bridge or Router. The default mode is Bridge. You have possibility to define Real Time “Clock Broadcast Period” in seconds, also.

Modbus TCP port	Modbus TCP idle timeout [s]	Modbus UDP port	Modbus base	HDL-Bus port	Gateway Subnet ID	Gateway Device ID	Clock broadcast period [s]	Gateway Mode
502	502	502	0	6000	1	1	15	Router

PortN	Port Type	BaudRate	Parity	ProtocolID/SlaveID
1	Modbus	38400	Odd	1
2	HDL-Bus	9600	Even	3

Buttons: Gateway, Save, Close

Concerning Modbus TCP, it is two parameters, which may change: “TCP port” and “TCP idle timeout” in seconds.

Concerning Modbus UDP, it is only one parameter to modify – “UDP port”.

“Modbus base” could be only 0 or 1, and setup the begin of the Modbus addresses.

In HDL-Bus, you must define port, Subnet ID and Device ID, as address of the Gateway in this network.

The Gateway supports 2 RS485 ports. Each of them could be “HDL-Bus” or “Modbus”. If port type is “Modbus”, then you can change bound rate and parity according your requirements.

“ProtocolID/SlaveID” has different meaning in Modbus and in HDL-Bus (according RS485 port type). In Modbus port, it is slave identification - SlaveID. In HDL-Bus port, it is usable only in Router mode. Here you define your port Subnet ID in HDL-Bus network. So, Gateway will know where to route HDL-Bus packages.

To make active all the above parameters, you must use “Save” button.

## **6. Ordering Information**

<u>Product Code</u>	<u>Product</u>
HDL-Modbus	HDL-Bus to Modbus Gateway (+ Euro power supply - 230VAC/24VDC)

Options:

<u>Product Code</u>	<u>Product</u>
ARM-I2C-16	Relay Module - 16 relays (8 OC + 8x10A free voltage changeover relays)