Modbus Module Manual

Version 1.0

REV00-20160512

GENERAL INFORMATION

DIVUS GmbH
Pillhof 51
I-39057 Eppan (BZ) - Italy

Operating instructions, manuals and software are protected by copyright. All rights are reserved. Copying, multiplication, translation and conversion, either partially or as a whole, is not permitted. You are allowed to make a single copy for backup purposes.
We reserve the right to make changes to the manual without prior notification.
We assume no responsibility for any errors or omissions that may appear in this document.

We do not assume liability for the flawlessness and correctness of the programs and data contained on the delivered discs.

You are always welcome to inform us of errors or make suggestions for improving the program.

The present agreement also applies to special appendices to the manual.

This manual can contain terms and descriptions, which improper use by third can harm the copyrights of the author.

Please read the manual before beginning and keep the manual for later use.

The manual has been conceived and written for users who are experienced in the use of PCs and automation technology.

CONVENTIONS

[KEYS]	Keys that are to be pressed by the user are given in square brackets, e.g. [CTRL] or [DEL]
COURIER	On-screen messages are given in the Courier font, e.g. C:\>
COURIER BOLD	Keyboard input to be made by the user are given in Courier bold, e.g. C:\>**DIR**
„…“	Names of buttons to be pressed, menus or other onscreen elements and product names are given within double quotes. (e.g. “Configuration”).
PICTOGRAMS	In this manual the following symbolic are used to indicate particular text blocs.
	Caution! A dangerous situation may arise that may cause damage to material.
	Hint Hints and additional notes
	New New features

INDEX:

1 GENERAL OVERVIEW 4

1.1 WHAT IS KNXCONTROL AND WHAT IS DIVUS OPTIMA? 4

1.2 WHAT IS THE GOAL OF THIS MANUAL? 4

1.3 REQUIREMENTS 4

1.4 SUPPORTED FUNCTIONALITY 5

1.5 REFERENCES 5

2 INSTALLATION 6

2.1 CONNECTION 6

2.1.1 RTU MODBUS 6

2.1.2 TCP/IP MODBUS 6

3 MODBUS LINES 7

3.1 INTRODUCTION 7

3.2 NEW MODBUS LINE 7

3.3 INTRODUCTION TO THE MODBUS DEVICES 9

3.4 CREATING NEW REGISTERS 10

3.5 MULTIPLE READING AND WRITING 12

3.6 REGISTER DETAILS 13

3.7 REGISTER EVENTS 14

4.1 NOTES 15

1 General overview

1.1 WHAT IS KNXCONTROL AND WHAT IS DIVUS OPTIMA?

KNXCONTROL defines a family of products for supervising and managing Home & Building Automation plants which were realised basing on the world-wide KNX standard. The KNXCONTROL products allow to manage any device of a system through access to DIVUS OPTIMA over a web browser using any type of computer, touchpanel, tablet or smartphone from both inside and outside of the local network. The KNXCONTROL family is composed of DIVUS KNX-SERVER, KNX-SUPERIO and the PDK.

DIVUS OPTIMA is the web interface to manage and visualise a KNX system. OPTIMA offers a complete set of base functionality which may be further expanded through its additional modules which cover special needs and functionalities.

Our website www.divus.eu offers data sheets, the PDK as well as all the technical manuals and documentations needed to get started and use KNXCONTROL devices productively. Get them for free in the download section.

1.2 WHAT IS THE GOAL OF THIS MANUAL?

This manual contains all the information to install, configure and use the functions specific to the MODBUS technology inside OPTIMA. It is mainly aimed to installers but may also help users who want to customize their home automation system.

Prerequisites are: a good knowledge of the MODBUS protocol, of the devices to manage on the plant and of OPTIMA; many sections of this manual refer to general concepts which are freely available for download on www.divus.eu and explained in the:

- *OPTIMA Administrator* manual and
- *OPTIMA User manual*

1.3 REQUIREMENTS

To be able to integrate MODBUS devices in Optima you will need:

- A KNXCONTROL device with Optima version 2.0.1 or newer or the PDK version 2.0.1 or newer. The latter is available as a free download on our website www.divus.eu.
- The MODBUS module activation license
- An RS485 cable.

To insert the MODBUS module activation license, proceed in this way:

- Access the administration area of OPTIMA as administrator
- Under SETUP in the navigation bar, choose LICENSE AND MODULES
- Find the MODBUS row
- Insert the license key
- Push the SAVE button and wait for the page to reload
- Load the same page again to check whether the license is recognized as valid
- Go to Technologies – Modbus to manage the new module after a browser refresh

For further details, please look up the OPTIMA Administrator manual.

1.4 SUPPORTED FUNCTIONALITY

Integrating the *MODBUS* protocol in OPTIMA allows to manage a multitude of functionalities within the automation system. This is due to the wide range of devices available on the market – be it devices which can be interfaced or proper MODBUS devices.

A MODBUS system may offer to manage (among others):

- Scenarios/scenes
- Illumination
- Shading
- Climate
- Security

Moreover it is possible to interface the functionalities already present in OPTIMA, thus further expanding the system's capabilities and possibilities.

1.5 REFERENCES

For further information regarding the MODBUS protocol, we recommend to consult the documentation on:

www.modbus.org/tech.php

We also advice to check out the OPTIMA Administrator manual available for download from our website www.divus.eu.

2 Installation

2.1 CONNECTION

2.1.1 RTU MODBUS

If the MODBUS device uses the serial port you'll need an RS485 cable[1].

2.1.2 TCP/IP MODBUS

The devices using TCP/IP MODBUS connect to the KNXCONTROL device over the LAN. They will have their own IP address and a port for communication, both changeable in the device's settings, and will connect through common Ethernet cables or over Wi-Fi.

[1] In reality there is no such thing as an RS485 cable. There is a socket on the KNX Server for these connections. So depending on the device's interface, you may choose a socket-to-socket cable or one which has the device's interface on one end and the single cables to connect to the socket on the other end.

3 MODBUS lines

3.1 INTRODUCTION

OPTIMA may use more than one MODBUS interface creating different MODBUS lines for each communication channel; these lines may operate at the same time as long as they are set to operate on different communication channels. The ones currently available are:

- RS485
- Network (RJ-45, LAN)

For each line you may create a certain number of objects identified as MODBUS registers, belonging to the single MODBUS devices, which can later on be inserted into the visualisation pages. It is also possible to manage more than one slave device if it is "cascaded" on the same RS485 line; in this case you would create a single line in OPTIMA and distinguish the devices by their slave address, as explained later on.

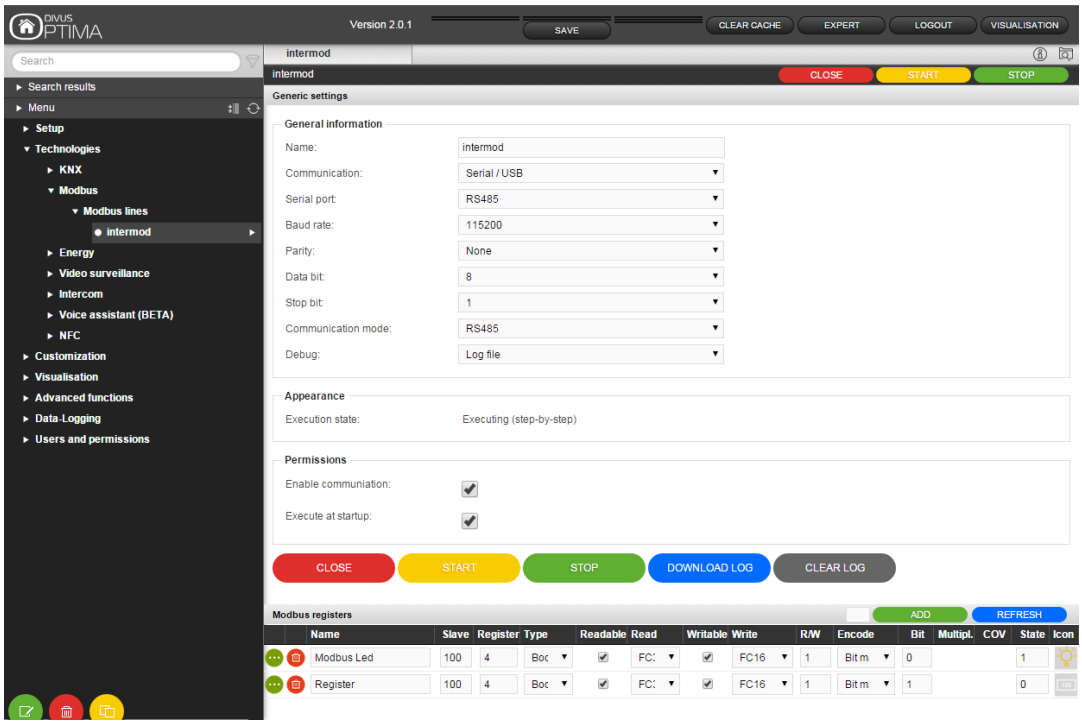
NOTE: It is important to create one single line for each communication port in OPTIMA. Otherwise there will be communication issues (conflicts).

3.2 NEW MODBUS LINE

To create a new MODBUS line:

- Access OPTIMA's administration
- Open the section Technologies from the navigation bar.
- Select MODBUS and then select "Modbus lines"
- Push the + ADD button in the bottom left toolbar

A new bus line is created and added to the list. Now you can reach the line's detail page by selecting it and then clicking the ellipsis symbol on its right or the EDIT button in the bottom left toolbar:



The following general settings are available on a MODBUS line's detail page (some of them depend on whether you chose *Serial/USB* or *Network* as *Communication*:

NAME	For an identifying label of the MODBUS line
COMMUNICATION	Allows to choose between *Serial/USB* (for RS485) or *Network*
SERIAL PORT	Allows to choose between different ports of the KNX Server. Choose RS485!
BAUD RATE	Allows to set the default communication speed
PARITY	Default is none. Devices may use the parity bit. If so, set the "how" and "where" here. Leave on "none"
DATA BIT	Default is 8 bits.
STOP BIT	Default is 1 bit.
COMMUNICATION MODE	Allows to choose between *RS232* and *RS485*. Choose *RS485*!
DEBUG	Choices are *Log file* (activates a debug logging) or *None* (no debugging). When enabled, the activity on the MODBUS line is logged to a file which may be downloaded using the blue *DOWNLOAD LOG* button. The debugging might help to make the system's commissioning easier/faster, but should only be used when really needed. The grey *DELETE LOG* button may be used to clear the log file's content. Both buttons are visible only after the *DEBUG* function was activated.
EXECUTION STATE	Shows whether the line is active or not
ENABLE COMMUNICATION	Allows to enable/disable the communication

EXECUTE AT STARTUP	Allows to start the MODBUS communication automatically on boot/reboot
QUERY TIME [MS]	(only visible in *expert mode*) Allows to set the time between the query of one register and the one to the next register

Using the yellow and green START and STOP buttons you can enable/disable the MODBUS line management.

NOTE: Every time a change to the configuration is done, the MODBUS line must be stopped and restarted using the *STOP* and *START* buttons

3.3 INTRODUCTION TO THE MODBUS DEVICES

Once the communication is configured, it is possible to start adding registers of the slave devices connected on the MODBUS line. The table of all the read and/or write registers of each device is usually available in the device's technical manual.

The technical information needed in order to make OPTIMA communicate with the device is the following:

COMMUNICATION PARAMETERS	For RTU: • Baud rate • Parity • Data bits • Stop bit • RS232 or RS485 For TCP: • IP address • Com. port
SLAVE ADDRESS	A MODBUS device always has a unique address which may usually be changed. (Please refer to your device's documentation for details.) Valid addresses go from 1 to 247. The KNXSERVER playing the master will not need such an address. The address 0 is reserved for broadcast messages.
FUNCTIONS	In the MODBUS world, the function refers to the second byte of a message. It is sent by the master and indicates in which slave's register table it has to go to access the data and whether it's a reading or writing operation. Reading: FC 01: Read coil status FC 02: Read input status FC 03: Read holding register FC 04: Read input registers Writing: FC 05: Force single coil FC 06: Preset single register FC 15: Force multiple coils FC 16: Preset multiple registers

REGISTERS	A register contains the value to read or write from OPTIMA. A device has a map identifying the registers and declaring their content's data type (e.g. temperature, on/off or alarm signalling type). As will be shown later, the registers are the objects available to OPTIMA's visualisation to interact with the devices over MODBUS. Once they are inside OPTIMA, it's easy to use them like any other object (in rooms, in scenarios, etc.). Consult the OPTIMA Administrator manual available on www.divus.eu for more details.
DATA TYPE	Tells the range of values each register may assume and the arithmetic operations which may be executed on those values.

Once you have all the information detailed above, it's recommended to do some read and write tests from a computer with a MODBUS software. This step helps to check the basic functionality (cable connection, configuration etc.) before connecting and testing from OPTIMA.

Here are some links offering software for MODBUS communication from PC:

http://www.modbustools.com/modbus_poll.html
https://oceancontrols.com.au/OCS-011.html
http://www.qmodbus.sourceforge.net/

3.4 CREATING NEW REGISTERS

To create one or more registers for each slave connected to the MODBUS line on OPTIMA, do the following:

- Open the MODBUS line in OPTIMA's administration
- Insert the amount of registers to be created next to the ADD button in the "Modbus registers" bar.
- Push the ADD button. The new registers are added to the list

Once created, you may rename the single registers, change their options or open their detail view.

NAME	Label used inside OPTIMA to identify the register of the MODBUS slave
SLAVE	Address of the slave device to which the register belongs
REGISTER	Address of the register

TYPE	Defines the data type of the register: • Boolean (ON/OFF) • Numeric (integer, float, etc)
READABLE	Flag to enable if the register is readable
READ	If readable, defines how to query the value: • FC1 (Read Coil Status) • FC2 (Read Input Register) • FC3 (Read Holding Registers) • FC4 (Read Input Registers)
WRITABLE	Flag to enable if the register is writable
WRITE	If the register is writable, defines the way to correctly set it: • FC05 (Force Single Coil) • FC06 (Preset Single Register) • FC15 (Force Multiple Coils) • FC16 (Preset Multiple Registers)
R/W	(Function currently not implemented)
ENCODE	Defines how the value should be interpreted:

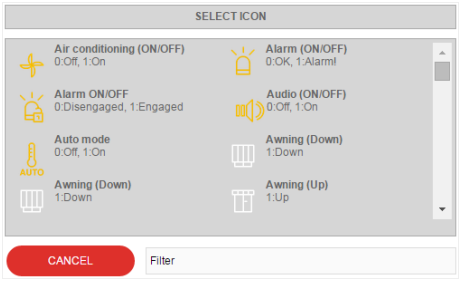
Unsigned integer	1 register (2 Bytes)
Signed integer	1 register (2 Bytes)
Unsigned Long	2 registers (2 Bytes) *
Signed Long	2 registers (4 Bytes) *
Signed long inverted **Hint**: The sequence of the 2 registers is inverted compared to the *Long* encoding	2 registers (4 Bytes) *
Floating point **	2 registers (4 Bytes) *
Floating point inverted ** **Hint**: The sequence of the 2 registers is inverted compared to the *Floating point* encoding	2 registers (4 Bytes) *
Bit mask	1 registers (2 bytes) (see BIT below)

* The 4 Bytes encoding needs the R/W field to be set to 2 to become selectable, because 2 registers need to be read/written.

** The floating point encoding only supports the reading from a slave device, not the writing.

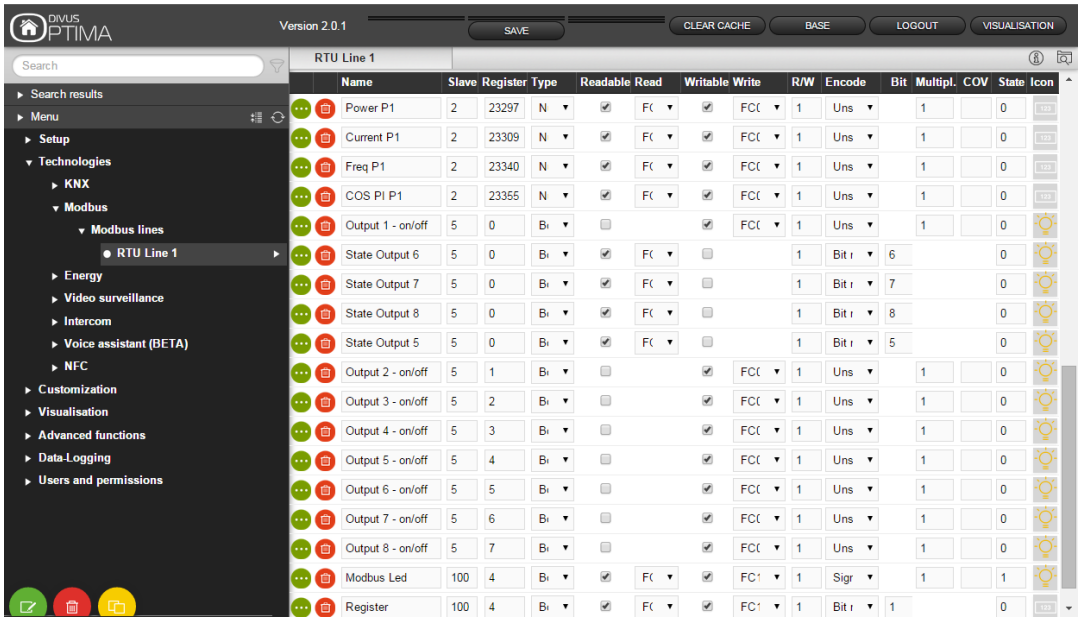
BIT	Available only if the ENCODE field is set to bit mask, this is used if e.g. a 1 byte register is composed of 8 bits which all have a distinct meaning (8 switches) So here you may set what bit should be read/written through the bit mask
MULTIPL.	The value of the register is multiplied by this value
COV	When handling numerical values, this option allows to set a lower threshold value, below which value changes will not be treated as such. Use the "." as decimal separator.
STATE	Current value of the register

ICON



Here you can set the icon and graphical element which will be shown in the visualisation. The available items depend on the data type of the object.

Example of configuration with multiple registers belonging to 2 slave devices on the same MODBUS line:



Hint: Press the REFRESH button after you added and/or changed registers to make them active system-wide.

3.5 MULTIPLE READING AND WRITING

It's possible to read/write multiple registers at once. This option is necessary for registers using the 4 Bytes encoding (when set to 2) and may generally be used when using contiguous registers to make reading/writing operations faster (if the device allows it).

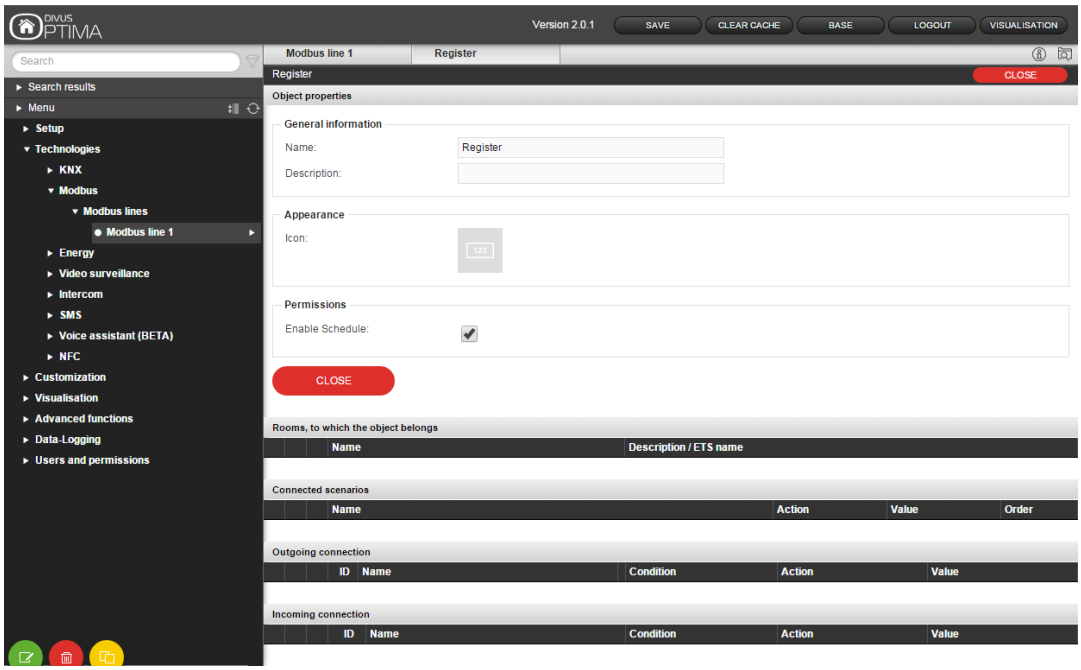
To activate this function, the R/W setting is set to a value higher than one. This automatically will add the number of sub-objects to the current (main) object, using the registers contiguous to the main.

		Name	Slave	Register	Typ	Lesbar	Lesen	Schreibbar	Schreiben	S/L	Verschlüsselung	Bit	Multipl.	COV	Status	Symbol
⋯	🗑	Register	1	1		☑	F	☑	FC0E	1	Unsigned		1		0	123
⋯	🗑	Register	1	2		☑	F	☑	FC0E	1	Unsigned		1		0	123
⋯	🗑	Register	1	3		☑	F	☑	FC0E	1	Unsigned		1		0	123
⋯	🗑	Register	1	4		☑	F	☑	FC0E	1	Unsigned		1		0	123
⋯	🗑	Register	1	10		☑	F	☑	FC1E	4	Unsigned		1		0	123
		Register	1	11		☑	F	☑	FC1E		Unsigned		1		0	123
		Register	1	12		☑	F	☑	FC1E		Unsigned		1		0	123
		Register	1	13		☑	F	☑	FC1E		Unsigned		1		0	123

The sub-objects (see last three in screenshot above) are bound to the main one: they'll share the same R/W setting and encoding. At the same time, their multiplier, COV and icon may be set autonomously.

3.6 REGISTER DETAILS

Clicking on the EDIT button of a register you reach the register's detail page which basically shows the same options already explained in the previous chapter. Switching to the EXPERT view on the top menu bar, a new description field becomes visible. You may use it to store additional information about the object. Like for most other objects in OPTIMA, the description field will also be searched for matches by OPTIMA's search function.



3.7 REGISTER EVENTS

Within the detail view of a register, you can associate actions to the change of state of a register. This change may be caused by the user, by events on the KNX bus or by advanced functions like scenarios, logics etc. In OPTIMA 2 these events are called *OUTGOING CONNECTIONS*.

To create a new *outgoing connection*, you need to:

- Search/go to the object you want to command
- Drag it to the *OUTGOING CONNECTIONS* list and drop it there
- Select the *CONDITION* of the register which shall trigger the action on the dropped object
- Select the ACTION to be executed on the dropped object – the options will change depending on the object type.
- If available, select the value which should be used for the action

In this way you could e.g. play a scenario using a push button connected to a MODBUS input. The procedure above can be repeated to have any number of actions triggered by a single register's status changes.

In the opposite direction, it is also possible to have a MODBUS register switched/commanded by other objects in OPTIMA. The triggering object will appear in the *INCOMING CONNECTIONS* section.

See the *OPTIMA Administrator Manual* for further details.

4 Appendix

4.1 NOTES