



***MODBUS - Driver of MODBUS/RTU
Protocol for MASTER Mode
User's Manual***

Doc. No. ENP4035
Version: 29-08-2005

ASKOM[®] and **asix**[®] are registered trademarks of ASKOM Spółka z o.o., Gliwice. Other brand names, trademarks, and registered trademarks are the property of their respective holders.

All rights reserved including the right of reproduction in whole or in part in any form. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without prior written permission from the ASKOM.

ASKOM sp. z o. o. shall not be liable for any damages arising out of the use of information included in the publication content.

Copyright © 2005, ASKOM Sp. z o. o., Gliwice



ASKOM Sp. z o. o., ul. Józefa Sowińskiego 13, 44-121 Gliwice,
tel. +48 (0) 32 3018100, fax +48 (0) 32 3018101,
<http://www.askom.com.pl>, e-mail: office@askom.com.pl

1. MODBUS - Driver of MODBUS/RTU Protocol for MASTER Mode

1.1. Driver Use

The MODBUS driver is used for data exchange with controllers or devices operating according to the MODBUS protocol. The transmission is executed by means of serial interfaces with the use of standard serial ports of an **asix** system computer.

The cooperation of **asix** with the controller by using the MODBUS protocol does not require any controller's program adaptation.

(While implementing the MODBUS protocol the RTU mode, which enables a larger capacity of the interface, was used).

1.2. Declaration of Transmission Channel

The full syntax of declaration of transmission channel working according to the MODBUS protocol is given below:

```
logical_name=MODBUS,id,port,[baud,character,parity,stop, max_i/o,max_register]
```

where:

<i>id</i>	- device identifier (slave id);
<i>port</i>	- serial port name;
<i>baud</i>	- transmission speed in baud; max transmission speed is equal to 115 kBd;
<i>character</i>	- number of bits in a transmitted character;
<i>parity</i>	- parity check type (even,odd,none);
<i>stop</i>	- number of stop bits;
<i>max_i/o</i>	- maximal number of inputs/outputs, the value of which may be transferred by devices within one cycle;
<i>max_register</i>	- maximal number of registers, the state of which may be transferred by the device within one cycle.

Parameters *baud*, *character*, *parity*, *stop*, *max_i/o*, *max_register* and *buffer* are optional. In case of omitting them the following default values are taken:

- transmission speed - 9600 Bd,
- number of bits in a character - 8,
- type of parity check - parity check,
- number of stop bits - 1,
- maximal number of inputs/outputs - 16,
- maximal number of registers - 4.

EXAMPLE

An example declaration of transmission channel working according to the MODBUS protocol is given below:

```
CHAN1=MODBUS,2,COM1,9600,8,even,1,40,16
```

The transmission channel with the logical name CHAN1 has the following parameters defined:

- MODBUS protocol using a serial interface;
- device identifier (slave id) 2;
- port COM1;
- transmission speed of 9600 Bd;
- transmitted character length - 8 bits;
- parity check;
- one stop bit.

1.3. Addressing the Process Variables

The syntax of symbolic address used for variables belonging to the MODBUS driver channel is as follows:

VARIABLE_TYPE *variable_index*

where:

variable_type - string indentifying the variable type in the MODBUS protocol;
variable_index - variable index within a given type.

The following symbols of types of process variables are allowable:

CS	- Coil Status (0X reference);
IS	- Input Status (1X reference);
HR	- Holding Register (4X reference);
IR	- Input Register (3X reference);
HRL	- 2 successive Holding Registers treated as a double word in INTEL format;
HRF	- 2 successive Holding Registers treated as a floating-point number in INTEL format;
HRLM	- 2 successive Holding Registers treated as a double word in MOTOROLA format;
HRFM	- 2 successive Holding Registers treated as a floating-point number in MOTOROLA format;
IRL	- 2 successive Input Registers treated as a double word in INTEL format;
IRF	- 2 successive Input Registers treated as a floating-point number in INTEL format;
IRLM	- 2 successive Input Registers treated as a double word in MOTOROLA format;
IRFM	- 2 successive Input Registers treated as a floating-point number in MOTOROLA format.

EXAMPLES

CS22	- Coil 22
IS197	- Input 197
HR118	- Holding Register 118
IR25	- Input Register 25

The MODBUS driver is loaded as a DLL automatically.

Considering the appearance of controllers using 32-bit registers, the MODBUS driver was extended with the support of 32-bit registers HR and IR:

- HR32L - 32-bit register HR of DWORD type (requires a calculation function based on DWORD, e.g. NOTHING_DW);
- HR32F - 32-bit register HR of FLOAT type (requires a calculation function based on FLOAT, e.g. NOTHING_FP);
- IR32L - 32-bit register of DWORD type (requires a calculation function based on DWORD, e.g. NOTHING_DW);
- IR32F - 32-bit register of FLOAT type (requires a calculation function based on FLOAT, e.g. NOTHING_FP).

EXAMPLE

X1, register HR no 10 as DWORD, HR32L10, CHAN32, 1, 1, NOTHING_DW
 X2, register IR no 20 as FLOAT, IR32F20, CHAN32, 1, 1, NOTHING_FP

1.4. Driver Configuration

The MODBUS protocol driver may be configured by means of the **[MODBUS]** section placed in the application INI file. Individual parameters are transferred in separate items of the section. Each item has the following syntax:

item_name=[number [,number]] [YES/NO]



LOG_FILE=file_name

Meaning - the item allows to define a file where all diagnostic messages of MODBUS driver and information about the contents of telegrams received and sent by the MODBUS driver will be written. If the item does not define the full path, then the log file will be created in the current directory. The log file should be used only while the **asix** start-up.

Default value - by default, the log file is not created.



LOG_OF_TELEGRAMS=YES/NO

Meaning - the item allows to write to the log file (declared by using the item LOG_FILE) the contents of telegrams sent and received by the MODBUS driver within the reading/writing of process variables. Writing the contents of telegrams to the log file should be used only while the **asix** start-up.

Default value - by default, the contents of telegrams are not written to the log file.



NUMBER_OF_REPETITIONS=number

Meaning - the item allows to define maximal number of trials to do the command in case of transmission errors.

Default value - by default, max. 3 repetitions are executed.

Parameter:

number - number of repetitions.



NO_ASCOMM=yes/no

Meaning - determines use of AsComm module to set connection for the MODBUS driver.

Default value - no.



RECV_TIMEOUT=slave_no,time

Meaning - the item allows to specify a maximal waiting time for arriving the first character of an answer from a specified remote device. After passage of this time it is assumed that the device under consideration does not work correctly and the transmission session ends with an error.

Default value - by default, it is assumed that the maximal waiting time for the first character of an answer is equal to 1000 milliseconds.

Parameter:

slave_no - number of slave placed in the declaration of the transmission channel using the MODBUS protocol;
time - number from a range of 100 - 65000 milliseconds.

EXAMPLE

[MODBUS]
 RECV_TIMEOUT=2,400



CHAR_TIMEOUT=slave_no,time

Meaning - the item allows to specify a maximal waiting time for arriving the successive character of the answer from a specified remote device. After passage of this time it is assumed that the device under consideration does not work correctly and the transmission session ends with an error.

Default value - by default, it is assumed that the maximal waiting time for the successive character of the answer is equal to 100 milliseconds.

Parameter:

slave_no - slave no. placed in the declaration of transmission channel using the MODBUS protocol;
time - number from a range of 10 - 2000 milliseconds.

EXAMPLE

[MODBUS]
 CHAR_TIMEOUT=2,400

**BLOCK_READ=NO/YES**

- Meaning - the item enables to set an operation mode, in which values of registers and coils are read individually (the function of block data reading is not used). It is valid for ALL the variables supported by the driver.
- Default value - by default, the mode of block data reading is used.

**TRANSMISSION_DELAY=number**

- Meaning - the item allows to declare a time interval between the end of receiving the answer and sending the successive query to the remote device. The maximal value for this item is equal to 65000 milliseconds.
- Default value - by default, the item assumes a value of transmission time of 3,5 characters.

1.5. Connection by Means of Modem

EXAMPLE

The MODBUS protocol may also exchange data by means of a modem connection.

The MODBUS driver channel is a client of server AsComm named MODBUS:n, where *n* is a number of the serial port taken from the ASMEN channel definition e.g.

if *channel_name*=modbus,4,com3,...

then the client name is MODBUS : 3.

The record given below must be placed in the [MODBUS:n] section for the AsComm program to establish a connection on dial-up links by means of the AsComm program.

Switched_line = Yes

If the modem is connected to an other port than COMn, then you should give the number of this port by means of the parameter *Port* or specify the modem name by means of the parameter *Modem*. You should also give a telephone number and define other parameters required. If MODBUS driver has to communicate with many controllers by means of the same modem, then one should define suitable number of channels assuming the parameter *port* as a virtual transmission channel and place suitable number of sections in the initialization file, by specifying in them an appropriate telephone number.

EXAMPLE

An example of initialization file content.

[ASMEN]

....

Chan1 = MODBUS,1,COM11,9600,8,none,1,16,16

Chan2 = MODBUS,1,COM12,9600,8,none,1,16,16

```
[MODBUS:11]
Switched_line = Yes
Modem = US Robotics
Number = 11111111
```

```
[MODBUS:12]
Switched_line = Yes
Modem = US Robotics
Number = 22222222
```

In the example above `Chan1` will communicate with a controller placed under the telephone number 11111111, and the `Chan2` with a controller placed under the telephone number 22222222. The `US Robotics` modem will be used. The *Modem* parameter may be replaced by the parameter *Port*, which specifies the number of the serial port to which the modem is connected.

You should notice the given above description of using the MODBUS driver on switched links does not include the modem parameterization. The modem configuration depends on types of used modems.

1. MODBUS - DRIVER OF MODBUS/RTU PROTOCOL FOR MASTER MODE	3
1.1. DRIVER USE	3
1.2. DECLARATION OF TRANSMISSION CHANNEL	3
1.3. ADDRESSING THE PROCESS VARIABLES	4
1.4. DRIVER CONFIGURATION	5
1.5. CONNECTION BY MEANS OF MODEM	7