FOR CONTROLLERS FIRMWARE V 649 AND FOLLOWING / STANDARD VERSION

MODBUS MANUAL FOR LD CONTROLLERS SERIES

ATTENZIONE - WARNING - ATTENTION
Impostare ID RS485 dello strumento a 1
Set RS485 controller's ID to 1
Définissez l'ID du contrôleur RS485 sur 1

WARNING: ENABLING MODBUS OPTION WILL DISABLE ERMES FUNCTIONALITIES

R6-02-24

NORME CE
EC RULES(STANDARD EC)
NORMAS DE LA CE

Direttiva Bassa Tensione
Low Voltage Directive
Directiva de baja tensión
} **2014/35/UE**

Direttiva EMC Compatibilità Elettromagnetica
EMC electromagnetic compatibility directive
EMC directiva de compatibilidad electromagnética
} **2014/30/UE**

GENERAL SAFETY INFORMATION

Danger! During an emergency of any nature within the environment where the pump group is installed, it is necessary to immediately turn off the power to the system and disconnect the instrument from the socket!

If particularly aggressive chemical materials are used, it is necessary to scrupulously follow the regulations regarding the use and storage of these substances!

If you install the instrument outside the European Community, comply with local safety regulations!

The manufacturer cannot be held responsible for damage to people or property caused by poor installation or incorrect use!

Attention! Install the instrument so that it is easily accessible whenever maintenance is required!
Never obstruct the place where the instrument is located!

The instrument must be slaved to an external control system. In case of lack of water, the dosing must be stopped.

The assistance and maintenance of the instrument and all its accessories must always be carried out by qualified personnel!

Always empty and carefully wash pipes that have been used with particularly aggressive chemical materials! Wear the most suitable safety devices for the maintenance procedure!

Always carefully read the chemical characteristics of the product to be dosed!

All operations must be carried out when the instrument is not connected to the power supply!

The MODBUS protocol

MODBUS is a serial communication protocol created in 1979 by MODICON (a company now part of the Schneider Electric group) to connect its programmable logic controllers (PLCs). AND
has become a *de facto* standard in industrial communication and is currently one of the most widespread connection protocols in the world among industrial electronic devices. The main reason for such a high use of MODBUS compared to other communication protocols is that this is an open and royalty-free protocol.

With the MODBUS protocol we define the format and mode of communication between a "master" that manages the system and one or more "slaves" that respond to queries from the master. Our device is a "slave".

The device address (ID), data format, and communication baud rate can be set directly from the MODBUS Communication menu from device menu.
MODBUS allows the connection of a master (e.g. a PC) and various “slaves” (e.g. measurement and control systems). Two versions are available: one for serial interface (RS-232 and RS-485) and one for ETHERNET.

The following operating modes can be distinguished for data transmission:

- MODBUS TCP: ETHERNET TCP/IP communication based on the client/server model
- MODBUS RTU: asynchronous serial transmission via RS-232 or RS-485
- MODBUS ASCII: similar to the RTU protocol except for a different data format used
  relatively rarely

In our case the **operating mode is RTU (asynchronous serial transmission via RS-485).**

MODBUS RTU

MODBUS RTU realizes a “master / slave” serial communication via RS-232 or RS-485. In order to address MODBUS RTU, the serial communication parameters must first be known and/or defined. These parameters include baud rate, parity, and stop bits. The “slave” addresses that must be also come into play here
directed by the "master".

Message format

The message format between the “master” and the “slave” includes:

- The address of the device with which the master established the transaction (address 0 corresponds to a broadcast message sent to all "slave" devices).
- The code of the function that is to be, or has been, executed.
- The data that needs to be exchanged.
- The error control composed according to the CRC16 algorithm.

If a device detects an error in the received message (format, parity or CRC16) or the address does not correspond to an online device, the message is considered invalid and discarded.
A “slave” that detects an error in the message will therefore not perform the action and will not respond to the request.

Data Format

Devices with MODBUS protocol use the following data formats for communication

8N1 format (default): 8 data bits, without any parity check (“No parity”) and with 1 stop bit. 8O1 format: 8 data bits, parity control on even bits (“Odd parity”) and with 1 stop bit.

8E1 format: 8 data bits, parity control on odd bits (“Even parity”) and with 1 stop bit. 8N2 format: 8 data bits, no parity check (“No parity”) and with 2 stop bits.

The polling speed must be equal to or greater than 500ms (milliseconds).

The address

MODBUS transactions always involve the master, which manages the line, and one "slave" at a time (except in the case of broadcast messages).

To identify the recipient of the message, a byte containing the numerical address of the selected device is transmitted as the first character.

Each of the "slaves" will therefore have been assigned a different numerical address that uniquely identifies it.

The eligible addresses are those from 1 to 255.

The address 0, which cannot be assigned to a "slave", placed at the head of the message transmitted by the master indicates that this is "broadcast", i.e. directed to all the "slaves" at the same time. Can only messages that do not require a response to carry out their function, therefore only assignments, can be transmitted as broadcasts.

The function code

The second character of the message transmitted by the master identifies the function that must be performed, to which the "slave" in turn responds with the same code to indicate that the function has been performed.

In our case, the only MODBUS functions that can be used are those shown below:

FUNCTION	DESCRIPTION
03	Reading registers
06	Single register setting
10	Setting up multiple registers

The last two characters of the message contain the cyclic redundancy code (Cyclic Redundancy Check) calculated according to the CRC16 algorithm.

MODBUS Data Addresses

Data Address	Offset	Associated number	GUY
0000- 270E Hex	40001	40001- 49999	R/W

THE MODBUS FUNCTIONS

Below is a detailed description of the MODBUS functions used.

Reading registers (03)

With this function, contiguous blocks of 16-bit internal registers are read from the "slave" device.

This function allows you to request the value of 16-bit registers (words) containing variables
numeric. Broadcast mode is not allowed.

Request

In addition to the address of the "slave" and the function code (03), the message contains the starting address ("Starting Address") expressed on two bytes and the "number of words" to be read also on two bytes. The maximum number of words that can be read is 125.

Example: Request to read the register with address 40001 (the first) from the "slave" with ID 01.

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	00	00	00	01	84	0A

Answer

In addition to the address of the "slave" and the function code (03), the response message includes the number of bytes read and the data contained in the read register.
Registers are made up of two bytes each, the first of which contains the most significant part.

Example: Response to the request above.

ID	FUNCTION	Number bytes read	DATE Address 0000 (HIGH)	DATE Address 0000 (LOW)	CRC (HIGH)	CRC (LOW)
01	03	02	00	00	B8	44

Request

Example: Read request from the "slave" with ID 1 of registers from 40001 to 40003.

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	00	00	00	03	05	CB

Answer

In addition to the address of the "slave" and the function code (03), the response message includes the number of bytes read and the data contained in the read registers.

Registers are made up of two bytes each, the first of which contains the most significant part.

Example: Response to the request above.

ID	FUNCTION	Number bytes read	DATE Address 0000 (HIGH)	DATE Address 0000 (LOW)	DATE Address 0001 (HIGH)	DATE Address 0001 (LOW)	DATE Address 0002 (HIGH)	DATE Address 0002 (LOW)	CRC (HIGH)	CRC (LOW)
01	03	06	00	00	00	00	00	00	21	75

Function Code (03) – Read pending registers

Request	Function Code	1 byte	0x03
	Starting address	2 bytes	From 0x0000 to 0xFFFF
	Number of registers	2 bytes	1 to 125 (0x01 to 0x7D)

Answer	Function Code	1 byte	0x03
	Number of bytes read	1 byte	2xN
	Register value	2N bytes	“N” is the number of registers

Single register setting (06)

This function allows you to set the value of a single 16-bit register. In addition to the address of the "slave" and the function code (06), the message contains the address of the variable expressed in two bytes and
the value that must be assigned. Broadcast mode is allowed. Example of Request (LEVEL
ALARM ENABLED and NC CONTACT): set the value 03 on the "slave" with ID 01 of register 40104.

ID	FUNCTION	Address (HIGH)	Address (LOW)	DATE WORD (HIGH)	DATE WORD (LOW)	CRC (HIGH)	CRC (LOW)
01	06	00	67	00	03	78	14

Answer

In addition to the address of the "slave" and the function code (06), the response message contains the address of the variable expressed in two bytes and the value assigned to it.

ID	FUNCTION	Address (HIGH)	Address (LOW)	DATE WORD (HIGH)	DATE WORD (LOW)	CRC (HIGH)	CRC (LOW)
01	06	00	67	00	03	78	14

Setting more than one register (10)

This function allows you to set the value of a consecutive block of 16-bit registers. Broadcast mode is allowed. In addition to the address of the "slave" and code 10, the message contains the starting address, the number of words to write, how many bytes the words are made up of and the value of the registers. In our case it is allowed to write only one word at a time and only words of 2 or 4 bytes.

Since we use the function with code 06 to write 2 bytes, we use this function to write words made up of four bytes.

Example: Set the pump having ID1 in CONSTANT mode (location 40140) at 80,000 L/h

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	Bytes for Words	DATE Word Address 008C (HIGH)	DATE Word Address 008C (LOW)	DATE Word Address 008B (HIGH)	DATE Word Address 008B (LOW)	CRC (HIGH)	CRC (LOW)
01	10	00	8B	00	01	04	00	01	38	80	F9	EF

Answer

In addition to the address of the "slave" and the function code (10), the message includes the starting address and the number of words written.

Example: Response to the request above.

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	10	00	8B	00	01	71	E3

ERROR MANAGEMENT

Two types of errors can occur during transmission, handled differently: errors
transmission and operational errors. Transmission errors are errors that occur if the message sent is compromised during sending and is therefore poorly received. In this case the error is detected by a possible bit parity check, if active in the serial transmission, or by a CRC check. The “slave”
that detects errors of this type in the message considers it invalid, discards the message without considering it and does not respond. However, if the message is correct in its form, without transmission errors, an error could occur in the content of the message itself, such as
a requested function, for any reason, is not executable, or the wrong content is addressed, an operational error occurs. The "slave" device responds to this error with an exception message.
This message consists of the address, the requested function delta code, an error code and the CRC. To indicate that the response is an error notification, the function code is returned with the most significant bit at "1".

The structure of the answer is as follows:

“SLAVE” ADDRESS	FUNCTION	ERROR CODE	CRC (HIGH)	CRC (LOW)

Request to a “slave” with wrong ID
Request

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
04	03	00	00	00	03	05	93

Answer

The message is considered invalid and there is no response.

Request

Request with wrong CRC

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	00	00	00	03	80	BB

Answer

The message is considered invalid and there is no response.

Request

Requests for content that does not exist in the “slave” Address 40566)

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	02	35	00	01	95	BC

Response (ILLEGAL DATA ADDRESS)

ID	FUNZ	Exception code	CRC (HI)	CRC (LO)
01	83	02	C0	F1

Request

Requests for content that does not exist in the "slave" (address 40014)

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	00	0D	00	03	94	08

Response (ILLEGAL DATA ADDRESS)

ID	Function	Exception queues	CRC (HIGH)	CRC (LOW)
01	83	02	C0	F1

Request (ILLEGAL DATA VALUE).

Attempt to write to a register (address 40100) a value that is not permitted for this address.

ID	FUNCTION	Address (HIGH)	Address (LOW)	DATE WORD (HIGH)	DATE WORD (LOW)	CRC (HIGH)	CRC (LOW)
01	06	00	63	00	04	78	17

Answer

ID	FUNCTION	Exception code	CRC (HIGH)	CRC (LOW)
01	86	03	02	61

Request

Function does not exist

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	08	00	0D	00	01	21	CB

Answer

(ILLEGAL FUNCTION VALUE)

ID	FUNCTION	Exception	CRC (HIGH)	CRC (LOW)
01	80	01	80	00

Exception codes

CODE		DESCRIPTION
01	ILLEGAL FUNCTION VALUE	Non-existent function
02	ILLEGAL DATA ADDRESS	The address referenced by the data field is not a permitted address on the addressed “slave”. Attempt to write to a read-only register.
03	ILLEGAL DATA VALUE	The value to be assigned for the data field is not allowed for this address.
05	BUSY WRITING	n/a

Connect the RS485 wires on the mainboard to use the MODBUS protocol.
The native version does not require additional hardware to function.

REGISTER	ADDRESS	VALUES	PROPERTY'
VALUE 1 SETPOINT OUT1 IS CH1	40001		R/W
VALUE 2 SETPOINT OUT1 IS CH1	40002		R/W
VALUE 1 P/M OUT1 IS CH1 IN PROPORTIONAL MODE	40003		R/W
VALUE 2 P/M OUT1 IS CH1 IN PROPORTIONAL MODE	40004		R/W
MODE SETPOINT OUT1 IS CH1	40005	0 ON/OFF 1 PROP2 Dis.	R/W
WAIT IN ON/OFF MODE OUT1 IS CH1	40007		R/W
VALUE 1 SETPOINT OUT IS CH2	40010		R/W
VALUE 2 SETPOINT OUT IS CH2	40011		R/W
VALUE 1 P/M OUT IS CH2 IN PROPORTIONAL MODE	40012		R/W
VALUE 2 P/M OUT IS CH2 IN PROPORTIONAL MODE	40013		R/W
MODE SETPOINT OUT IS CH2	40014	0 ON/OFF 1 PROP2 Dis.	R/W
			R
WAITING IN CH2 PULSE ON/OFF MODE	40016		R/W
VALUE 1 SETPOINT OUT2 IS CH1	40019		R/W
VALUE 2 SETPOINT OUT2 IS CH1	40020		R/W
VALUE 1 P/M OUT2 IS CH1 IN PROPORTIONAL MODE	40021		R/W
VALUE 2 P/M OUT2 IS CH1 IN PROPORTIONAL MODE	40022		R/W
MODE SETPOINT OUT2 IS CH1	40023	0 ON/OFF 1 PROP2 Dis.	R/W
			R/W
WAIT IN ON/OFF MODE OUT2 IS CH1	40025		R/W
			R/W
			R/W
VALUE 1 SETPOINT OUT RELAY CH1	40028		R/W
VALUE 2 SETPOINT OUT RELAY CH1	40029		R/W
VALUE 1 PERC/SEC OUT RELAY [PERC SEC-> PROP PWM MODE FIXED PWM] CH1	40030		R/W
VALUE 2 PERC/SEC OUT RELAY [PERC SEC-> PROP PWM MODE FIXED PWM] CH1	40031		R/W
MODE SETPOINT OUT RELAY CH1	40032	0 PROP PWM 1 ON/OFF2 Fixed PWM3 Dis	R/W
			R/W
			R/W
			R/W
			R/W
VALUE 1 SETPOINT OUT RELAY CH2	40037		R/W
VALUE 2 SETPOINT OUT RELAY CH2	40038		R/W
VALUE 1 PERC/SEC OUT RELAY [PERC SEC-> PROP PWM MODE FIXED PWM] CH2	40039		R/W
VALUE 2 PERC/SEC OUT RELAY [PERC SEC-> PROP PWM MODE FIXED PWM] CH2	40040		R/W
MODE SETPOINT OUT RELAY CH2	40041	0 PROP PWM 1 ON/OFF2 Fixed PWM3 Dis	R/W
PARAMETER delay	40046	Time in minutes	R/W
PARAMETER MODE	40047	0: PH PRIORITY 1: NO PRIORITY	R/W
PARAMETERS Tau	40048		R/W
PARAMETERS Pcode1	40049		R/W
PARAMETERS Pcode2	40050		R/W
PARAMETERS Pcode3	40051		R/W
PARAMETERS Pcode4	40052		R/W
PARAMETER TEMP_VIEW	40053	0 NO 1 YES	R/W
			R/W
probe_failure CH1 mode	40055	1 STOP 0 DOSE	R/W
probe_failure CH1 time	40056	Time in minutes 0: OFF editable value from 100 to 250 min	R/W
probe_failure CH2 mode	40057	1 STOP 0 DOSE	R/W
probe_failure CH2 time	40058	Time in minutes 0: OFF editable value from 100 to 250 min	R/W
			R/W
dosing_alarm CH1 mode	40060	1 STOP 0 DOSE	R/W
dosing_alarm CH1 time	40061	Time in minutes 0: OFF editable value from 1 to 100 min	R/W

dosing_alarm CH2 mode	40062	1 STOP 0 DOSE	R/W
dosing_alarm CH2 time	40063	Time in minutes 0: OFF editable value from 1 to 100 min	R/W
			R/W
flow_setting mode	40065	0:Disable 1:Reverse 2:Direct	R/W
flow_setting time	40066	Time in minutes ALLOWED VALUES 0-99 min	R/W
			R/W
THRESHOLD ALARM CH1 MODE High	40068	0:disable 1 :Enable	R/W
THRESHOLD ALARM CH1 MODE Low	40069	0:disable 1 :Enable	R/W
CH1 THRESHOLD ALARM VALUE High	40070		R/W
CH1 THRESHOLD ALARM VALUE Low	40071		R/W
THRESHOLD ALARM CH1 VALUE Time	40072	Time in minutes editable value from 0 to 99 min	R/W
THRESHOLD ALARM CH1 VALUE Mode	40073	1 STOP 0 DOSE	R/W
CH2 MODE High THRESHOLD ALARM	40075	0:disable 1 :Enable	R/W
CH2 MODE Low THRESHOLD ALARM	40076	0:disable 1 :Enable	R/W
CH2 THRESHOLD ALARM VALUE High	40077		R/W
CH2 THRESHOLD ALARM VALUE Low	40078		R/W
CH2 THRESHOLD ALARM VALUE Time	40079	Time in minutes editable value from 0 to 99 min	R/W
CH2 THRESHOLD ALARM VALUE Mode	40080	1 STOP 0 DOSE	R/W
			R/W
set_clock.format //European time format (value 0) or American (value 1)	40082	European time format (value 0) or American time format (value 1)	W
set_clock.am_pm //AM -> 0, PM -> 1	40083	//AM -> 0, PM -> 1	W
set_clock.day	40084		W
set_clock.month	40085		W
set_clock.year	40086		W
set_clock.hour	40087		W
set_clock.minute	40088		W
MA_OUTPUT CH1 MAX value	40090		R/W
MA_OUTPUT CH1 MIN value	40091		R/W
MA_OUTPUT CH1 Mode	40092	0: 0/20mA 1:4/20mA	R/W
MA_OUTPUT CH1 En_DIS_on_alarm	40093	0: Disable on Alarm 1: Enable on Alarm	R/W
MA_OUTPUT CH2 MAX value	40095		R/W
MA_OUTPUT CH2 MIN value	40096		R/W
MA_OUTPUT CH2 Mode	40097	0: 0/20mA 1:4/20mA	R/W
MA_OUTPUT CH2 En_DIS_on_alarm	40098	0: Disable on Alarm 1: Enable on Alarm	R/W
MA_OUTPUT_Temp.MAX	40100	°C value with 1 decimal point / °F no decimal points	R/W
MA_OUTPUT_Temp.MIN	40101	°C value with 1 decimal point / °F no decimal points	R/W
MA_OUTPUT_Temp.Mode	40102	0: 0/20mA 1:4/20mA	R/W
MA_OUTPUT_Temp.En_DIS_on_alarm	40103	0: Disable on Alarm 1: Enable on Alarm	R/W
LOG Enable	40105	1: Enable 0: Disable	R/W
LOG Time Hour	40106		R/W
LOG Time Minute	40107		R/W
LOG.Time.Am	40108	1 YES 0 NO	R/W
LOG.Time.Pm	40109	1 YES 0 NO	R/W
LOG.Every.Hour	40110		R/W
LOG.Every.Minute	40111		R/W
LOG.Every.Am	40112	1 YES 0 NO	R/W
LOG.Every.Pm	40113	1 YES 0 NO	R/W
LOG.OUT_E	40114	1: LOG OUT Enable 0: LOG OUT disable	R/W
Alarm	40861	General alarm 0: No alarm 1: Alarm	R
Level alarm LEV1 CH1	40862	Alarm level 0: No alarm 1: Alarm	R

LEV2 CH1 level alarm	40863	Alarm level 0: No alarm 1: Alarm	R
LEV CH2 level alarm	40864	Alarm level 0: No alarm 1: Alarm	R
STBY	40866	STBY Alarm 0: No alarm 1: Alarm	R
SEPR	40867	SEPR Alarm 0: No alarm 1: Alarm	R
OUT IS1 CH1 dosage alarm	40868	0 No alarm 1 Dose alarm 2 Stop alarm	R
OUT IS CH2 Dosing Alarm	40869	0 No alarm 1 Dose alarm 2 Stop alarm	R
OUT IS2 CH1 dosage alarm	40870	0 No alarm 1 Dose alarm 2 Stop alarm	R
Dosing alarm OUT RELAY CH1	40871	0 No alarm 1 Dose alarm 2 Stop alarm	R
Dosing alarm OUT RELAY CH2	40872	0 No alarm 1 Dose alarm 2 Stop alarm	R
CH1 minimum maximum value alarm	40873	0 No alarm 1 Dose alarm 2 Stop alarm	R
CH2 minimum maximum value alarm	40874	0 No alarm 1 Dose alarm 2 Stop alarm	R
COMMA position reading channel (ACCORDING TO THE PROBE) CH1	40875	Current measurement division factor. Possible values 4: div 1, 3: div 10,2: div 100, 2: div 1000	R
COMMA position reading channel (ACCORDING TO THE PROBE) CH2	40876	Current measurement division factor. Possible values 4: div 1, 3: div 10,2: div 100, 2: div 1000	R
COMMA position temperature channel	40877	Current measurement division factor. Possible values 4: div 1, 3: div 10,2: div 100, 2: div 1000	R
OUTPUT STATUS IS1 CH1	40878	0 OFF 1 ON	R
IS2 CH1 OUTPUT STATUS	40879	0 OFF 1 ON	R
CH1 RELAY OUTPUT STATUS	40880	0 OFF 1 ON	R
EXIT STATUS IS CH2	40881	0 OFF 1 ON	R
CH2 RELAY OUTPUT STATUS	40882	0 OFF 1 ON	R
			R
READING CHANNEL CH1	40474		R
READING CHANNEL CH2	40475		R
TEMPERATURE READING	40476		R
			R
SERIAL PROBE FAILURE	40500	0 No alarm 1 Probe disconnected alarm	R
SERVICE READING IN mV CH1	40501	for SCL probes it is a negative value	R
SERVICE READING IN mV CH2	40502	for SCL probes it is a negative value	R
SERVICE TEMPERATURE IN mV	40503		R
DATE: DAY	40504		R
DATE: MONTH	40505		R
DATE: YEAR	40506		R
DATE: TIME	40507		R
DATE: MINUTES	40508		R
DATE: SECONDS	40509		R
&runtime_S.AM_PM	40510	2: EUROPEAN FORMAT 1: PM 0: AM	R

Disposal of end-of-life equipment by users

This symbol warns you not to dispose of the product with normal waste. Respect human health and the environment by taking discarded equipment to a designated collection center for the recycling of electronic and electrical equipment. For further information visit the online site.

All the materials used for the construction of the dosing pump and for this manual can be recycled and thus help maintain the incalculable environmental resources of our planet. Do not disperse harmful materials into the environment! Find out from the competent authority about the recycling programs for your area!

MODBUS MANUAL FOR LD CONTROLLERS SERIES

ATTENZIONE - WARNING - ATTENTION
Impostare ID RS485 dello strumento a 1
Set RS485 controller's ID to 1
Définissez l'ID du contrôleur RS485 sur 1

CE

R6-02-24

NORME CE
EC RULES(STANDARD EC)
NORMAS DE LA CE

Direttiva Bassa Tensione
Low Voltage Directive
Directiva de baja tensión
} **2014/35/UE**

Direttiva EMC Compatibilità Elettromagnetica
EMC electromagnetic compatibility directive
EMC directiva de compatibilidad electromagnética
} **2014/30/UE**

GENERAL SAFETY INFORMATION

Danger! During an emergency of any nature within the environment where the pump group is installed, it is necessary to immediately turn off the power to the system and disconnect the instrument from the socket!

If particularly aggressive chemical materials are used, it is necessary to scrupulously follow the regulations regarding the use and storage of these substances!

If you install the instrument outside the European Community, comply with local safety regulations!

The manufacturer cannot be held responsible for damage to people or property caused by poor installation or incorrect use!

Attention! Install the instrument so that it is easily accessible whenever maintenance is required!
Never obstruct the place where the instrument is located!

The instrument must be slaved to an external control system. In case of lack of water, the dosing must be stopped.

The assistance and maintenance of the instrument and all its accessories must always be carried out by qualified personnel!

Always empty and carefully wash pipes that have been used with particularly aggressive chemical materials! Wear the most suitable safety devices for the maintenance procedure!

Always carefully read the chemical characteristics of the product to be dosed!

All operations must be carried out when the instrument is not connected to the power supply!

The MODBUS protocol

MODBUS is a serial communication protocol created in 1979 by MODICON (a company now part of the Schneider Electric group) to connect its programmable logic controllers (PLCs). AND
has become a *de facto* standard in industrial communication and is currently one of the most widespread connection protocols in the world among industrial electronic devices. The main reason for such a high use of MODBUS compared to other communication protocols is that this is an open and royalty-free protocol.

With the MODBUS protocol we define the format and mode of communication between a "master" that manages the system and one or more "slaves" that respond to queries from the master. Our device is a "slave".

The device address (ID), data format, and communication baud rate can be set directly from the MODBUS Communication menu from device menu.
MODBUS allows the connection of a master (e.g. a PC) and various “slaves” (e.g. measurement and control systems). Two versions are available: one for serial interface (RS-232 and RS-485) and one for ETHERNET.

The following operating modes can be distinguished for data transmission:

- MODBUS TCP: ETHERNET TCP/IP communication based on the client/server model
- MODBUS RTU: asynchronous serial transmission via RS-232 or RS-485
- MODBUS ASCII: similar to the RTU protocol except for a different data format used
  relatively rarely

In our case the **operating mode is RTU (asynchronous serial transmission via RS-485).**

MODBUS RTU

MODBUS RTU realizes a “master / slave” serial communication via RS-232 or RS-485. In order to address MODBUS RTU, the serial communication parameters must first be known and/or defined. These parameters include baud rate, parity, and stop bits. The “slave” addresses that must be also come into play here
directed by the "master".

Message format

The message format between the “master” and the “slave” includes:

- The address of the device with which the master established the transaction (address 0 corresponds to a broadcast message sent to all "slave" devices).
- The code of the function that is to be, or has been, executed.
- The data that needs to be exchanged.
- The error control composed according to the CRC16 algorithm.

If a device detects an error in the received message (format, parity or CRC16) or the address does not correspond to an online device, the message is considered invalid and discarded.
A “slave” that detects an error in the message will therefore not perform the action and will not respond to the request.

Data Format

Devices with MODBUS protocol use the following data formats for communication

8N1 format (default): 8 data bits, without any parity check (“No parity”) and with 1 stop bit. 8O1 format: 8 data bits, parity control on even bits (“Odd parity”) and with 1 stop bit.

8E1 format: 8 data bits, parity control on odd bits (“Even parity”) and with 1 stop bit. 8N2 format: 8 data bits, no parity check (“No parity”) and with 2 stop bits.

The polling speed must be equal to or greater than 500ms (milliseconds).

The address

MODBUS transactions always involve the master, which manages the line, and one "slave" at a time (except in the case of broadcast messages).

To identify the recipient of the message, a byte containing the numerical address of the selected device is transmitted as the first character.

Each of the "slaves" will therefore have been assigned a different numerical address that uniquely identifies it.

The eligible addresses are those from 1 to 255.

The address 0, which cannot be assigned to a "slave", placed at the head of the message transmitted by the master indicates that this is "broadcast", i.e. directed to all the "slaves" at the same time. Can only messages that do not require a response to carry out their function, therefore only assignments, can be transmitted as broadcasts.

The function code

The second character of the message transmitted by the master identifies the function that must be performed, to which the "slave" in turn responds with the same code to indicate that the function has been performed.

In our case, the only MODBUS functions that can be used are those shown below:

FUNCTION	DESCRIPTION
03	Reading registers
06	Single register setting
10	Setting up multiple registers

The last two characters of the message contain the cyclic redundancy code (Cyclic Redundancy Check) calculated according to the CRC16 algorithm.

MODBUS Data Addresses

Data Address	Offset	Associated number	GUY
0000- 270E Hex	40001	40001- 49999	R/W

THE MODBUS FUNCTIONS

Below is a detailed description of the MODBUS functions used.

Reading registers (03)

With this function, contiguous blocks of 16-bit internal registers are read from the "slave" device.

This function allows you to request the value of 16-bit registers (words) containing variables
numeric. Broadcast mode is not allowed.

Request

In addition to the address of the "slave" and the function code (03), the message contains the starting address ("Starting Address") expressed on two bytes and the "number of words" to be read also on two bytes. The maximum number of words that can be read is 125.

Example: Request to read the register with address 40001 (the first) from the "slave" with ID 01.

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	00	00	00	01	84	0A

Answer

In addition to the address of the "slave" and the function code (03), the response message includes the number of bytes read and the data contained in the read register.
Registers are made up of two bytes each, the first of which contains the most significant part.

Example: Response to the request above.

ID	FUNCTION	Number bytes read	DATE Address 0000 (HIGH)	DATE Address 0000 (LOW)	CRC (HIGH)	CRC (LOW)
01	03	02	00	00	B8	44

Request

Example: Read request from the "slave" with ID 1 of registers from 40001 to 40003.

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	00	00	00	03	05	CB

Answer

In addition to the address of the "slave" and the function code (03), the response message includes the number of bytes read and the data contained in the read registers.

Registers are made up of two bytes each, the first of which contains the most significant part.

Example: Response to the request above.

ID	FUNCTION	Number bytes read	DATE Address 0000 (HIGH)	DATE Address 0000 (LOW)	DATE Address 0001 (HIGH)	DATE Address 0001 (LOW)	DATE Address 0002 (HIGH)	DATE Address 0002 (LOW)	CRC (HIGH)	CRC (LOW)
01	03	06	00	00	00	00	00	00	21	75

Function Code (03) – Read pending registers

Request	Function Code	1 byte	0x03
	Starting address	2 bytes	From 0x0000 to 0xFFFF
	Number of registers	2 bytes	1 to 125 (0x01 to 0x7D)

Answer	Function Code	1 byte	0x03
	Number of bytes read	1 byte	2xN
	Register value	2N bytes	“N” is the number of registers

Single register setting (06)

This function allows you to set the value of a single 16-bit register. In addition to the address of the "slave" and the function code (06), the message contains the address of the variable expressed in two bytes and
the value that must be assigned. Broadcast mode is allowed. Example of Request (LEVEL
ALARM ENABLED and NC CONTACT): set the value 03 on the "slave" with ID 01 of register 40104.

ID	FUNCTION	Address (HIGH)	Address (LOW)	DATE WORD (HIGH)	DATE WORD (LOW)	CRC (HIGH)	CRC (LOW)
01	06	00	67	00	03	78	14

Answer

In addition to the address of the "slave" and the function code (06), the response message contains the address of the variable expressed in two bytes and the value assigned to it.

ID	FUNCTION	Address (HIGH)	Address (LOW)	DATE WORD (HIGH)	DATE WORD (LOW)	CRC (HIGH)	CRC (LOW)
01	06	00	67	00	03	78	14

Setting more than one register (10)

This function allows you to set the value of a consecutive block of 16-bit registers. Broadcast mode is allowed. In addition to the address of the "slave" and code 10, the message contains the starting address, the number of words to write, how many bytes the words are made up of and the value of the registers. In our case it is allowed to write only one word at a time and only words of 2 or 4 bytes.

Since we use the function with code 06 to write 2 bytes, we use this function to write words made up of four bytes.

Example: Set the pump having ID1 in CONSTANT mode (location 40140) at 80,000 L/h

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	Bytes for Words	DATE Word Address 008C (HIGH)	DATE Word Address 008C (LOW)	DATE Word Address 008B (HIGH)	DATE Word Address 008B (LOW)	CRC (HIGH)	CRC (LOW)
01	10	00	8B	00	01	04	00	01	38	80	F9	EF

Answer

In addition to the address of the "slave" and the function code (10), the message includes the starting address and the number of words written.

Example: Response to the request above.

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	10	00	8B	00	01	71	E3

ERROR MANAGEMENT

Two types of errors can occur during transmission, handled differently: errors
transmission and operational errors. Transmission errors are errors that occur if the message sent is compromised during sending and is therefore poorly received. In this case the error is detected by a possible bit parity check, if active in the serial transmission, or by a CRC check. The “slave”
that detects errors of this type in the message considers it invalid, discards the message without considering it and does not respond. However, if the message is correct in its form, without transmission errors, an error could occur in the content of the message itself, such as
a requested function, for any reason, is not executable, or the wrong content is addressed, an operational error occurs. The "slave" device responds to this error with an exception message.
This message consists of the address, the requested function delta code, an error code and the CRC. To indicate that the response is an error notification, the function code is returned with the most significant bit at "1".

The structure of the answer is as follows:

“SLAVE” ADDRESS	FUNCTION	ERROR CODE	CRC (HIGH)	CRC (LOW)

Request to a “slave” with wrong ID
Request

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
04	03	00	00	00	03	05	93

Answer

The message is considered invalid and there is no response.

Request

Request with wrong CRC

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	00	00	00	03	80	BB

Answer

The message is considered invalid and there is no response.

Request

Requests for content that does not exist in the “slave” Address 40566)

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	02	35	00	01	95	BC

Response (ILLEGAL DATA ADDRESS)

ID	FUNZ	Exception code	CRC (HI)	CRC (LO)
01	83	02	C0	F1

Request

Requests for content that does not exist in the "slave" (address 40014)

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	03	00	0D	00	03	94	08

Response (ILLEGAL DATA ADDRESS)

ID	Function	Exception queues	CRC (HIGH)	CRC (LOW)
01	83	02	C0	F1

Request (ILLEGAL DATA VALUE).

Attempt to write to a register (address 40100) a value that is not permitted for this address.

ID	FUNCTION	Address (HIGH)	Address (LOW)	DATE WORD (HIGH)	DATE WORD (LOW)	CRC (HIGH)	CRC (LOW)
01	06	00	63	00	04	78	17

Answer

ID	FUNCTION	Exception code	CRC (HIGH)	CRC (LOW)
01	86	03	02	61

Request

Function does not exist

ID	FUNCTION	Starting Address (HIGH)	Starting Address (LOW)	Number of Words (HIGH)	Number of Words (LOW)	CRC (HIGH)	CRC (LOW)
01	08	00	0D	00	01	21	CB

Answer

(ILLEGAL FUNCTION VALUE)

ID	FUNCTION	Exception	CRC (HIGH)	CRC (LOW)
01	80	01	80	00

Exception codes

CODE		DESCRIPTION
01	ILLEGAL FUNCTION VALUE	Non-existent function
02	ILLEGAL DATA ADDRESS	The address referenced by the data field is not a permitted address on the addressed “slave”. Attempt to write to a read-only register.
03	ILLEGAL DATA VALUE	The value to be assigned for the data field is not allowed for this address.
05	BUSY WRITING	n/a

Disposal of end-of-life equipment by users

This symbol warns you not to dispose of the product with normal waste. Respect human health and the environment by taking discarded equipment to a designated collection center for the recycling of electronic and electrical equipment. For further information visit the online site.

All the materials used for the construction of the dosing pump and for this manual can be recycled and thus help maintain the incalculable environmental resources of our planet. Do not disperse harmful materials into the environment! Find out from the competent authority about the recycling programs for your area!