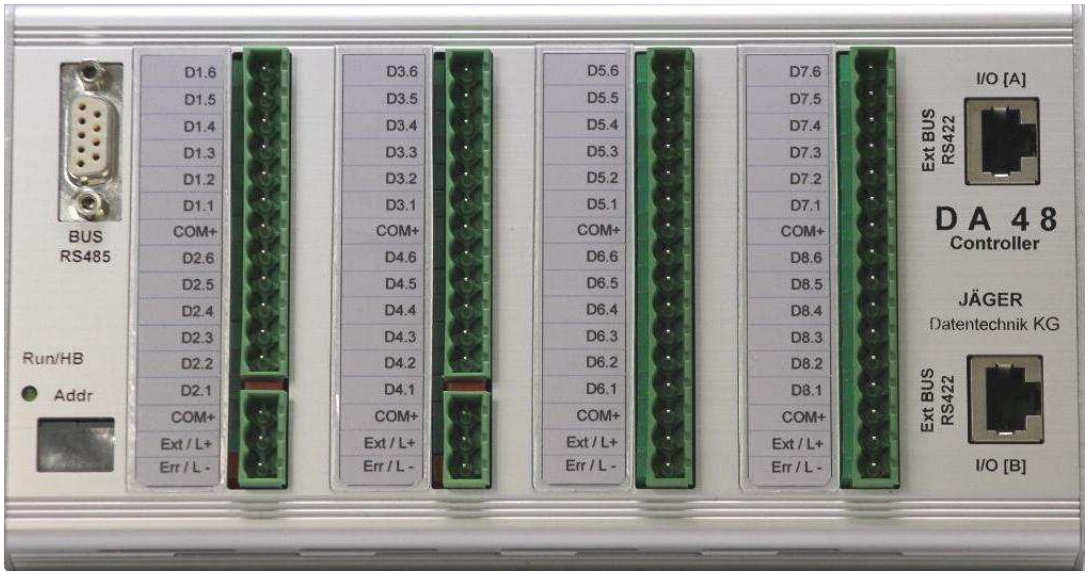
Software Description

MODBUS RTU Slave

DA48-1 SW - Version 1.2



Index

1.0 General Description

The DA48-1 controller is called short ,,DA48-1“ in the following text. It is a modular hardware solution whereby the operation of digital and analog actuators as well as the converting of digital input signals is possible.
The DA48-1 works as a slave group and is controlled by a serial MODBUS RTU protocol.

The protocol implementation is bounded to product-specific specifications of the **Modbus Organization, Inc. 37 Wheeler Rd. North Grafton, MA 01536 USA**.

The single functions are described in detail under the chapter MODBUS functions.
MODBUS TCP/IP connections can be realized by a pre-switched gateway.

DA48-1 has 4 modules which can be individual put together on customer request.
Three modules are available at the moment:

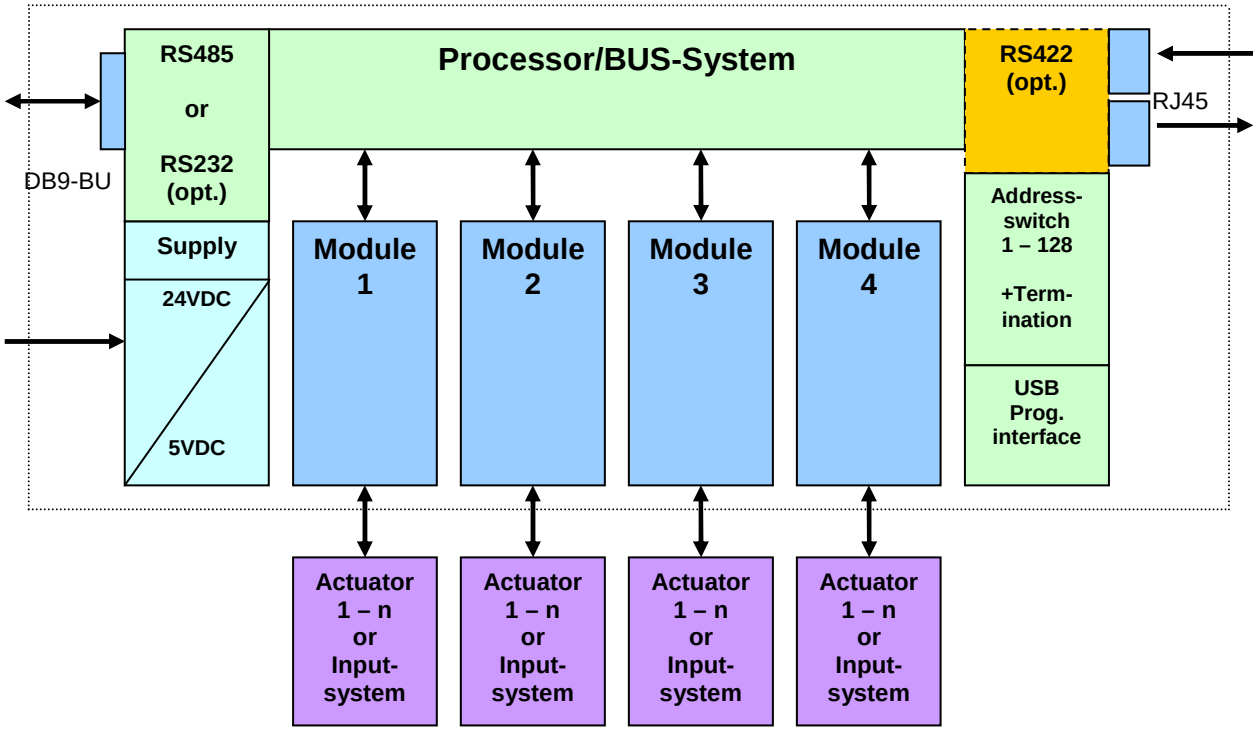
- JDD12-L 12 channel dig. output low-side 24VDC/240mA à output/short-circuit proof
- JDD12-AC 12 channel dig. output galv. isolated 24V/1A AC
- JDA12-H 12 channel analog outputs 0-20mA each
- JDE08-H 8 channel digital input galv. isolated +/-24VDC

More modules are in development.

All output modules have a current measurement, a wire break- and short-circuit monitoring per channel. The data are available by specificated 4xxxx registers.
Besides the known MODBUS order for the controlling of output references additional 4xxxx registers are available. Those are enable special functions like digital outputs with a scale from 0 – 100% in 4- or 6Bit mode.

The DA48-1 is available with RS485-interface in series, but also with RS232-interface for single applications if requested.
By an external accessible address switch can be adjusted up to 128 addresses. The operation of 32 slave assembly groups on one BUS segment is possible by the used RS485 interface modules.
The DA48-1 is supplied by a direct voltage of 12-36V. 24V is standard and the controller is equipped for an installation for DIN rail mounting. Other technical characteristics you can find on the data sheet.
Another RS422 interface for the operation of extension assembly groups (e.g. Relais assembly groups) can be used.
The user gets all the relevant configuration data related to I/O references, registers and module parameter by the command **Report Slave ID** as one data block.
The DA48-1 achieves an extraordinary price/performance ratio by its packing density.

2.0 Schematic structure and function



The installed modules are aknowledged automatically after switching on and the type is saved in an internal FRAM. All programmed user values without dynamic procedures are getting saved. Likewise, internal counters which are reloaded after Restart from the FRAM contents. Output states are not getting saved and are controlled by the BUS Master. Costumer specific adjustments on request anytime.
A green LED of the DA48-1 shows the power supply function and signalizes the regular controller function in seconds range. This LED is also an indicator for the MODBUS command processing. The LED flashes during this procedure. No or static lightening signalizes a malfunction of the controller.
This is only the case during a static disfunction. Is the software not working the DA48-1 is getting restarted by an internal watchdog counter. If the LED flashes after switching on, no address could be adjusted. **In that case the controller is not operation-ready.**

On the right front side the DA48-1 is accessable by a recess of the address switch. Settings on chart 2. On the left front side is the termination switch. Parallel a 120 Ohm resistor is switched.

This switch has to be switched in „Terminate“ position at the first and last assembly group.

Chart 1: LED Display

DA48-1		Modul		Erklärung
LED-HB				permanent OFF = controller error
LED-HB				permanent ON = controller error
LED-HB				After switching on: Fast flashing **= no address adjusted**
LED-HB				controller works: LED flashes in seconds range
LED-HB				controller works: MODBUS command in processing. LED flashes fast
		LED-L+		External voltage source connected
		LED-Error		Module sum error Reason: Short-circuit output or overtemperatur
		per output		Lightens when the output is active
		per output		Flashes in seconds range during wire-break or overcurrent (only digital modules)

Every single output is assigned with a 2x5mm sized rectangle-LED. This LED signalizes the current flow to the actor by a brightness adjustment for an analog output and the panel condition ON/OFF for a digital output.
In case of a wire-break the relevant LED starts to flash, but for an analog output the light stops. Additional a disfunction is signalized by a red ERR-LED.

Display of the controller

E1 Supply voltage below 10VDC

CC **C**ommun**c**ation watchdog has triggered

CE **C**RC Checksum **e**rror

0 1 normal address display **01 – 99** higher addresses with decimal point

0 0. this signalizes address 100

Schematic illustration and LED`s (not to scale)

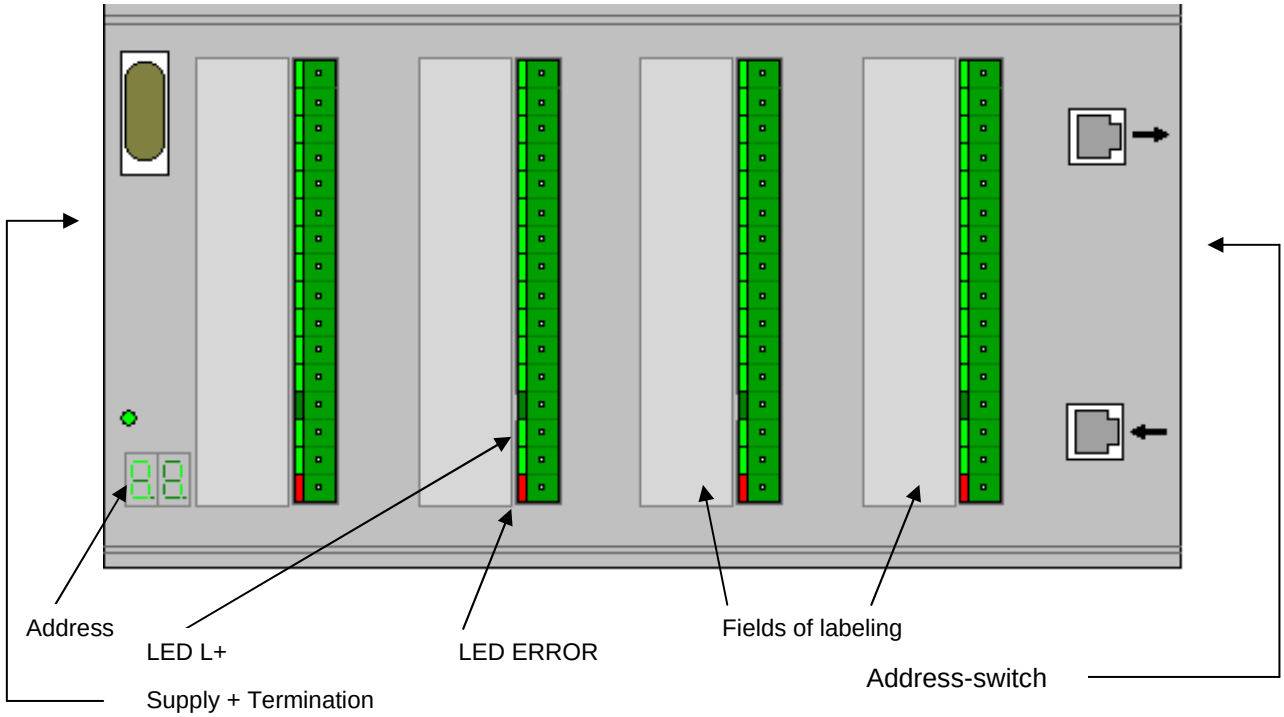
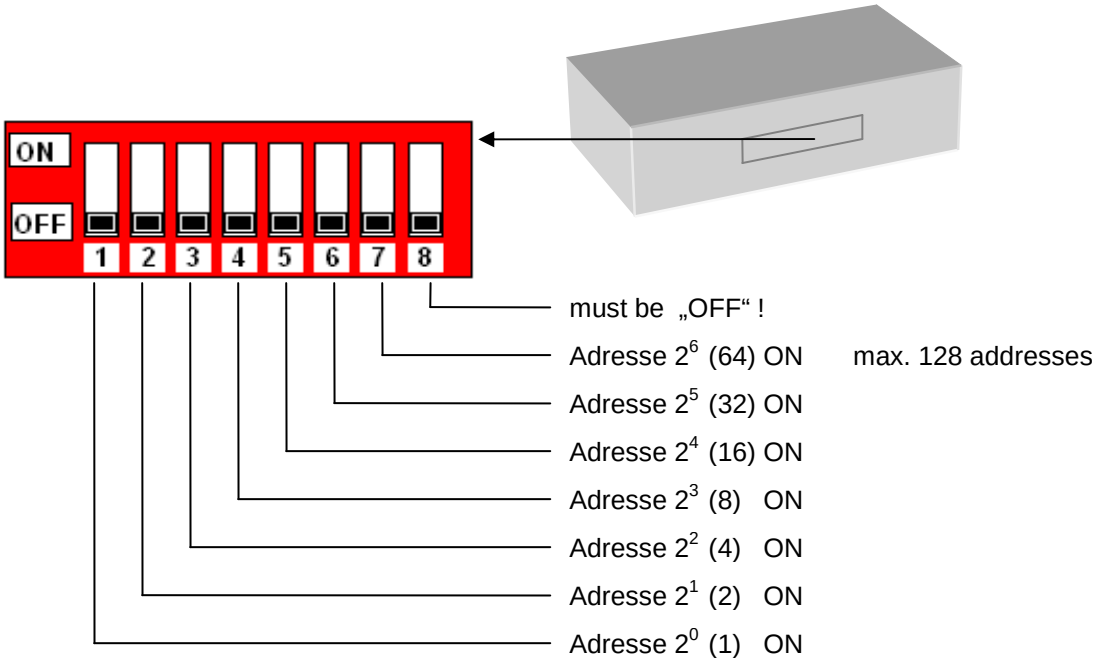


Illustration1: Adress-switch setting



The adjusted address is illustrated down left by a seven segment display.

The below-mentioned chart shows the availability of the DA48-1 with standard adjustment and possible standard references (Number 0XXX and 1XXX cannot being put together):

Address	1	Address-switch
Baudrate	19200	Changeable per diagnostic command
Datenlenge	8 Bit	Changeable per diagnostic command
Stopbit	1 Bit	Changeable per diagnostic command
Parität	Even	Changeable per diagnostic command
I/O Referenzen		
0XXXX	48	Coil outputs
1XXXX	32	discrete inputs
3XXXX	4	Input register
4XXXX	300	Holding register

4 analog output modules come without 0XXX and 1XXX references.

3.0 Calculation of power dissipation

If an assembly group of 48 analog outputs is used, it is important to attend to the connection of the ratio of the connected load per output and the power supply voltage. The assembly group is able to conduct a power dissipation of max. 8Watt without outside cooling. Every further power dissipation in the output stages causes additional assembly group warming and requires extra cooling. This is the case at maximum power of 20mA. To guarantee a simple calculation, the following calculation should be used:

<u>Example</u>

Load/load resistance = 200 Ohm a output = 20mA, feed-in 24V, 48 outputs

Formula: (Pv=power power dissipation per output)

Brackets are calculated from the inside to the outside.

Pv = (supply voltage – (load resistance x 0,02 + 7,2)) x 0,02

For this example is essential:

Pv =(24V – (200Ohm x 0,02A + 7,2V)) = 24V – 11,2V = 12,8V

12,8V x 0,02A = 0,256W (with 48 outputs: 48 x 0,256W = 12,288W) **✝ !!**

Here the feed-in should be reduced to 15V e.g., this way you get:

Pv = (15V – (200Ohm x 0,02A + 7,2V)) x 0,02A = 0,076W (with 48 outputs: 48 x 0,076 = 3,65W) ✓

Or 12V supply:

Pv = (12V – (200Ohm x 0,02A + 7,2V))x0,02A = 0,016W (with 48 outputs: 48 x 0,016 = 0,768W) ✓

Is the result negative, the supply voltage is too low.

4.0 Overview of Holding Register 4XXXX reference (16Bit values)

Reg-No from	Reg-No to	Read/ Write	notation	value
0001	0048	W	Analog output values (0 – 20mA) per output	0 – 10000
0050	0057	W	8 x dig. Output values 4Bit	0 – 10000
0058	0061	-	reserve	
0062	0069	W	8 x dig. Output values 6Bit	0 – 10000
0070	0117	R	Actual current values per analog output	0 – 10000
0118	0165	R	Status dig/analog outputs 1 – 48 max.	1=wire break 2=short
0200		**R/W**	**Module type and grouping module1**	
0201		**R/W**	**Module type and grouping module2**	
0202		**R/W**	**Module type and grouping module3**	
0203		**R/W**	**Module type and grouping module4**	
0204		R/W	res.	
0205		R/W	Analog outputs of 0 or 4mA	0 = 0mA 1 = 4mA
0206		R/W	res.	
0207		R	max. quantity of 0XXX references	12 – 48
0208		R	max. quantity of analog outputs	12 – 48
0209		R	max. quantity discrete inputs 1XXXX	8 – 32
0210		R	Quantity input register 3XXXX	4
0211		R	Quantity holding register 4XXXX	300
0212		R/W	Wire break general control OFF/ON	0/1
0213	0216	R/W	per module 12Bit filter wire break OFF/ON	Bit 0/1 per output
0220		R	Slave address	1 – 128
0221		**R**	**CPU version**	**1.00 – X.XX**
0222		**R**	**value supply voltage of controller**	**12.00 – 24.XX**
0223		**R**	**temperature of the controller**	**20.00 – 75.00**
0224		R	controller status	3 = standard
0225		R	controller serial number	nnnn
0226		R/(W)	RS485 speed	01=4800Bd, 02=9600Bd 03=19200Bd, 04=38400Bd
0227		R/(W)	RS485 data length	01=7Bit, 02=8Bit
0228		R/(W)	RS485 stop bit	01=1 Stopbit, 02=2 Stopbit
0229		R/(W)	RS485 parity	01 = even, 02 = odd 03 = none

Reg-Nr from	Reg-Nr to	Read/ Write	notation	value
0230		R/(W)	RS422 speed (extension BUS)	01=4800Bd, 02=9600Bd 03=19200Bd 04=38400Bd 05=250.000Bd
0231		R/(W)	RS422 data length	01=7Bit, 02=8Bit
0232		R/(W)	RS422 stop bit	01=1 stop bit, 02=2 stop bit
0233		R/(W)	RS422 parity	01 = even, 02 = odd 03 = none
0234		R	Controller mode	1=RTU
0235		R	Reset counter contr. since first startup After each power up counter (nnnn) +1	0 – 65535 – 0
0236		R	CRC-error counter after power up After power up start with 0	0 – 65535 – 0
0237		R	Telegram counter after power up After power up start with 0	0 – 65535 – 0
0238		R	Heart-Beat counter Increment every 500ms After power up start with 0	0 – 65535 – 0
0245		R/W	Low mapp register (INP-Mod.1) 1-4 points to register 253 Bit(n)	00-FF
0246		R/W	High mapp register (INP-Mod.1) 5-8 points to register 253 Bit(n)	00-FF
0247		R/W	Low mapp register (INP-Mod.2) 1-4 points to register 254 Bit(n)	00-FF
0248		R/W	High mapp register (INP-Mod.2) 5-8 points to register 254 Bit(n)	00-FF
0249		R/W	Low mapp register (INP-Mod.3) 1-4 points to register 255 Bit(n)	00-FF
0250		R/W	High mapp register (INP-Mod.3) 5-8 points to register 255 Bit(n)	00-FF
0251		R/W	Low mapp register (INP-Mod.4) 1-4 points to register 256 Bit(n)	00-FF
0252		R/W	High mapp register (INP-Mod.4) 5-8 points to register 256 Bit(n)	00-FF
0253		R/W	Status register input module1	
0254		R/W	Status register input module2	
0255		R/W	Status register input module3	
0256		R/W	Status register input module4	
0257	0260	R/W	Code register input module 1 – 4 (AND)	0/1 input inactive/active
0261		R/W	Activation register for status register 253-256	0/1
0262		R/W	WDT time register (communication control)	1000 – 65000ms
0263		R/W	WDT code and status 0051h – 005F = module1-4 or all If active command is mirrored in the upper 8 bytes	0051-005F

Writing operations in () cannot being executed by the orders Preset Single/Multiple Register.

4.1 Explanation of single register functions

Register	Read/Write		description
40001 - 48		**W**	Every register is assigned to one output channel. By loading a register with a value between 0-10000 the relevant analog output between 0-20mA is actuated.
40050 - 61		**W**	Every register is assigned to one 4Bit digital output. The registers are relevant only when the grouping has been switched to 1. Is the grouping switched to 1, a register value between 0-10000 is setting over to a 4Bit output.
40062 – 69		**W**	Every register is assigned to one 6Bit digital output. The registers are relevant only when the grouping has been switched to 2. Is the grouping switched to 2, a register value between 0-10000 is setting over to a 6Bit output.
40070 – 117	**R**	**-**	Actual current value of an output converted to 0 – 10000. (e.g. 20mA = 10000) Register 40070 is first output at first module.
40118 – 165	**R**	**-**	A status register per output. 1 signalizes wire-break, 2 signalizes over-current at this output.

Register	Read/Write		decription
40200-203	**R**	**W**	**module type and grouping module 1- 4**

Register content:

High-Byte module group								Low-Byte module type							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R	R	R	R	R	W	W	R	R	R	R	R	R	R	R

Module type (self identified by the controller):
01 = 12x digital output module or 2x4/2x6Bit group DC
02 = 12x digital output module High-Side DC
03 = 12x digital output module 3x4Bit group AC/DC
04 – 06 = res.
07 = 12x analog output module Low-Side
08 = 12x analog output module High-Side
09 – 12 = res.
13 = 8x digital input module DC
14 – 18 = res.

Module grouping (only for digital output module bit 8+9)

01 = 3	x 4 Bit digital	data from reg. 4xxxx is mapped to 4Bit
01 = 2	x 4 Bit digital	(only module type 01)
02 = 2	x 6 Bit digital	data from Reg. 4xxxx is mapped to 6Bit

Module sequence (00 = 1. module same type) = Bit 12 + 13
Data is configurated from the controller.
e.g.

00 = 1. module, type 03	(Reg. 40200)
01 = 2. module, type 03	(Reg. 40201)
00 = 1. module, type 13	(Reg. 40202)
01 = 2. module, type 13	(Reg. 40203)

Distribution groups to registers (digital outputs)

Module	3 x 4Bit	2 x 4/6Bit
1	40050 – 40052	40062 – 40063
2	40053 – 40055	40064 – 40065
3	40056 – 40058	40066 - 40067
4	40059 – 40061	40068 - 40069

40212	**R**	**W**	**Wire-break control general OFF=0/ON=1**
			Wire-break control per output
40213	**R**	**W**	12 Bit Map ->module1. Bit set (1) means active.
40214	**R**	**W**	12 Bit Map ->module2. Bit set (1) means active.
40215	**R**	**W**	12 Bit Map ->module3. Bit set (1) means active.
40216	**R**	**W**	12 Bit Map ->module4. Bit set (1) means active.

Register	Read/Write		decription

40221 **R** **-** **CPU-Version**

E.g: Version 1.51

High-Byte								Low-Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**

40222 **R** **-** **Supply value (power supply voltage)**

E.g.: **24,23V**

High-Byte								Low-Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**1**

40223 **R** **-** **DA48-1 Temperature (only whole degree Celsius ℃)**

E.g.: **26,00℃**

High-Byte								Low-Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

It is measured between 20℃ – 75℃
75℃ and higher causes an error situation and the m odule is going to be switched off in case of digital/analog versions.
In this case an ERROR LED is switched on.

4.2 Classification input modules to register

Register	Read/Write	description
40245-246	**R W**	**Register Mapping (High/Low) for 1. input module**

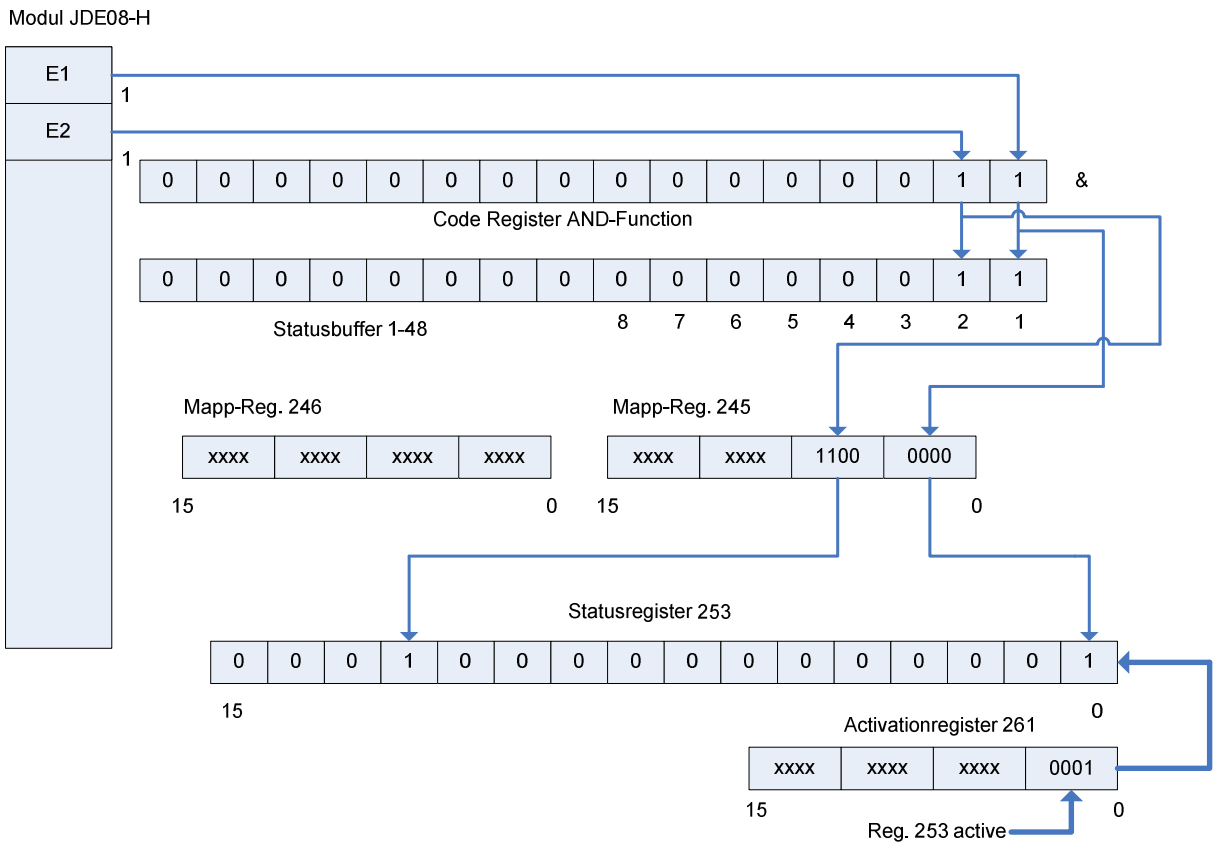
Every half byte shows one position in the associated status register. This way every input 1-8 of a module of a Bit position in the status register can be allocated.

Register 245/246

High-Byte								Low-Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**

Associated status register = **40253**

Example : The first and second input of input module 1 has to be mapped to bitposition 0 and 12 of the status register. For that the reading of a specified status word is possible.
By the MODBUS command READ INPUT status, the read in of the input conditions of 1xxxx references is still possible.
! The data contents may not be redundant when programming the MAPP-registers!
! The allocation is going to be deleted when read in 0-values into the MAPP registers!



(table 3: schematic of register configuration)

Register	Read/Write		description
40245-246	**R**	**W**	**Register Mapping (Low/High) for 1. input module**
40247-248	**R**	**W**	**Register Mapping (Low/High) for 2. input module**
40249-250	**R**	**W**	**Register Mapping (Low/High) for 3. input module**
40251-252	**R**	**W**	**Register Mapping (Low/High) for 4. input module**
40253	**R**	**W**	**Status Register (16Bit) for 1. input module**
40254	**R**	**W**	**Status Register (16Bit) for 2. input module**
40255	**R**	**W**	**Status Register (16Bit) for 3. input module**
40256	**R**	**W**	**Status Register (16Bit) for 4. input module**

Four Code register give the possibility to activate or deactivate every input by setting a “1” or a “0” in the depending bit position.

40257		**W**	**Code register input 1. input module**
40258		**W**	**Code register input 2. input module**
40259		**W**	**Code register input 3. input module**
40260		**W**	**Code register input 4. input module**
40261		**W**	**Activation register for status register (see table 3)**

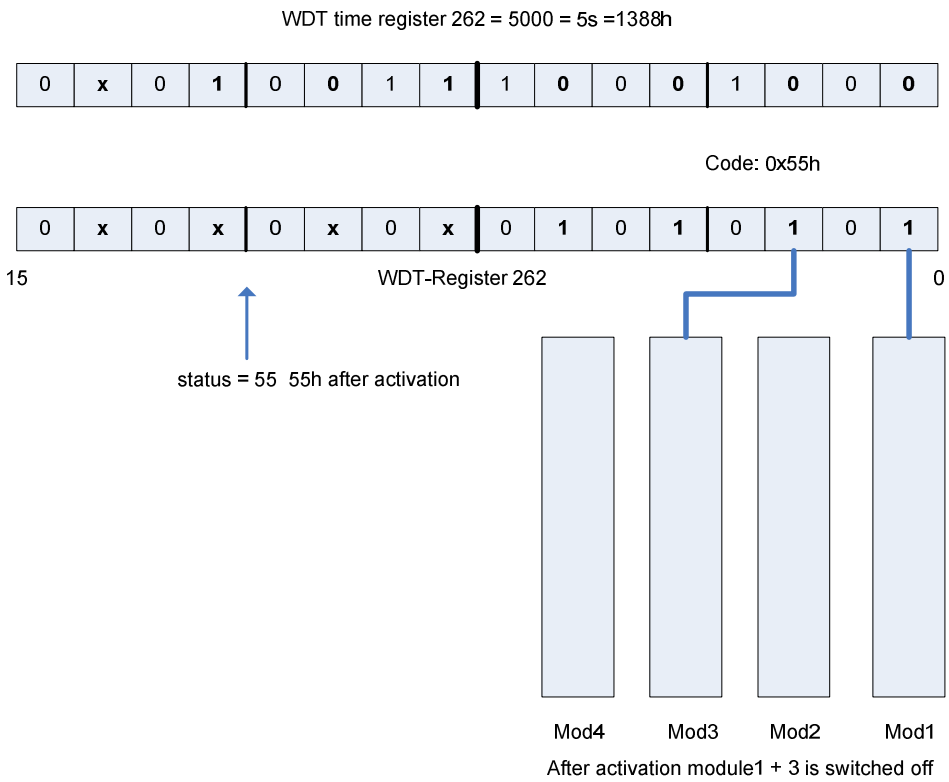
4.3 Communication control

In special applications it is necessary to set up a secure output status if the data connection fails. In that case there are two registers which guarantee that. One register contains the time up to the disconnection, the other one rules which output module is switched of and controlles the activation of the disconnection.

Register	Read/Write		description
40262	**R**	**W**	**Unit time in ms for communication-watchdog** **Value 1000 – 65000ms = 1s – 65s**
40263	**R**	**W**	**The least significant register part contains a code which activates the watchdog.** **(see table 5 for codes)**

By activation the command in the upper byte of the register is readable as a status.
By first incoming telegram the watchdog is released and reloaded to the programmed value and the upper activation register part is set to 0.
The communication watchdog may not be mistaken for the internal watchdog. The internal one is just released by a total software hang-up and resets the entire controller!

Example: Time watchdog = 5s, in case of activation module 1 + 3 switched off



(table 4: schematic of WDT activation)

After programming the time the watchdog function becomes active after sending the function code.
A triggered watchdog is indicated by the errorcode „CE“ on the display.

Code for module shutdown with communcation WDT :

Code	Module1	Module2	Module3	Module4
51_H	●	---	---	---
52_H	---	●	---	---
53_H	●	●	---	---
54_H	---	---	●	---
55_H	●	---	●	---
56_H	---	●	●	---
57_H	●	●	●	---
58_H	---	---	---	●
59_H	●	---	---	●
$5A_H$	---	●	---	●
$5B_H$	●	●	---	●
$5C_H$	---	---	●	●
$5D_H$	●	---	●	●
$5E_H$	---	●	●	●
$5F_H$	●	●	●	●

(table 5: WDT codes)

5.0 Supported MODBUS commands

Code	MODBUS Function	Reg.	Application	Broadcast
01_H	READ COIL Status	**0xxxx**	- Read out digital output condidtions	No
02_H	READ INPUT Status	**1xxxx**	- Read out digital input conditions	No
03_H	READ HOLDING REGISTER	**4xxxx**	- read out measured values, counter values - read out device-configuration	No
04_H	READ INPUT REGISTER	**3xxxx**	- read out binary registers contents H/L	No
05_H	FORCE SINGLE COIL	**0xxxx**	- setting a specific digital output	Yes
06_H	PRESET SINGLE REGISTER	**4xxxx**	- setting a holding register	Yes
08_H	DIAGNOSTIC		- device-connection test (subfunction 0)	No
$0F_H$	FORCE MULTIPLE COILS	**0xxxx**	- setting/simulating digital output conditions	Yes
10_H	PRESET MULTIPLE REGISTERS	**4xxxx**	- setting multiple holding registers	Yes
11_H	REPORT SLAVE ID	**4xxxx**	- complete slave status message (is defined separately)	No
40H	PROG COM	**4xxxx**	- changing the interface parameters	Yes
41H	COM RESET	**---**	- restart communication module	Yes

01H READ COIL STATUS

Request Master -> Slave

Address	Function	Data				CRC-Check
		start address		number of conditions		
Adr	01H	High-Byte	Low-Byte	High-Byte	Low-Byte	Crc16

Answer Slave -> Master

Address	Function	Data			CRC-Check
		number data bytes	condition 8-1	condition 16 - 9	
Adr	01H	8 Bit	8 Bit	8 Bit	Crc16

If the wire-break detection has been activated, the status is set to 0 even though the output has been activated.

02H READ INPUT STATUS

Reading the (digital) input states 0 = OFF, 1 = ON

Request Master -> Slave

Address	Function	Data				CRC-Check
		start address		number of conditions		
Adr	02H	High-Byte	Low-Byte	High-Byte	Low-Byte	Crc16

Answer Slave -> Master

Address	Function	Data			CRC-Check
		number of data bytes	condition 8-1	condition 16 - 9	
Adr	02H	8 Bit	8 Bit	8 Bit	Crc16

03H READ HOLDING REGISTERS

Request Master -> Slave

Address	Function	Data				CRC-Check
		start address		number of register		
Adr	03H	High-Byte	Low-Byte	High-Byte	Low-Byte	Crc16

Answer Slave -> Master

Address	Function	Data		CRC-Check
		number of data bytes	information	
Adr	03H	n (8 Bit)	n/2 Register	Crc16

Max. 125 registers can be read. Register addresses are always relative to the stated reference number. Register 1=0, register 125 = 124.

05H FORCE SINGLE COIL

Request Master -> Slave 1 x coil ON = (FF00) // coil OFF = (00 00)

Address	Function	Data				CRC-Check
		startadr High	startadr Low	data Hi	data Lo	
Adr	05H	00	XX	**FF**	00	Crc16

The slave is sending back the same telegram as an echo.
This command can be also send in the Broadcast Modus. This way outputs can be changed simultanous on all slaves.In this case the slave does not answer.

06H PRESET SINGLE REGISTER

Request Master -> Slave

Address	Function	Data				CRC-Check
		register address		data		
Adr	06H	High-Byte	Low-Byte	High-Byte	Low-Byte	Crc16

Answer Slave -> Master

Address	Function	Data				CRC-Check
		register address		data		
Adr	06H	High-Byte	Low-Byte	High-Byte	Low-Byte	Crc16

08H DIAGNOSTICS

The subfunction 0 is testing if the device is connected with (Adr).
The same telegram is send back 1:1.

Request Master -> Slave

Address	Function	Data				CRC-Check
		subfunction		data		
Adr	08H	0	0	High-Byte	Low-Byte	Crc16

0FH FORCE MULIPLE COILS

Request Master -> Slave

Address	Function	Data						CRC-Check
		start address		no. of conditions		number of Bytes	information	
Adr	0FH	High	Low	High	Low	n	n Bytes	Crc16

Example: Setting digital outputs 1 – 12 of slave 2

Send: `02 0F 00 00 00 0C 02` **`FF 0F`** `crlc_l crc_h`

FF = 1111 1111b: output 1 – 8 ON
0F = 0000 1111b: output 9 – 12 ON, 13 – 16 = OFF

Answer Slave -> Master

Address	Function	Data				CRC-Check
		start address		number of conditions		
addr	0FH	High-Byte	Low-Byte	High-Byte	Low-Byte	Crc16

This command can be also send in the Broadcast Modus. This way outputs can be changed simultanous on all slaves. The slaves do not answer.

10H PRESET MULTIPLE REGISTERS

Request Master -> Slave

Address	Function	Data						CRC-Check
		start address		number of register		number of bytes	information	
Adr	10H	High	Low	High	Low	1-n	2 Bytes/Reg	Crc16

Answer Slave -> Master

Address	Function	Data				CRC-Check
		start address		number of register		
Adr	10H	High-Byte	Low-Byte	High-Byte	Low-Byte	Crc16

Max. 125 registers can be written simultanous. Register addresses are always relative to the stated reference number. Register 1 = 0, register 125 = 124.

Only for reading approved registers cannot being overwritten. The writing option is performed until the next writeable register. This is prevented by an internal lock-table which allows writing accesses on approved registers only.

Writing on an locked register does not lead to an error warning.

This command can be also executed in Broadcast Modus. This way the register contents can be changed simultanous on all slaves. The slaves do not answer.

11H REPORT SLAVE ID

The answer contains a summary of all controller data.

.

Master ->

Address	Function	CRC-Check
Addr	11H	Crc16

Reply:

Address	Function	Data			CRC-Check
		byte count	byte 1	byte (n)	
Addr	11H	n=3Dh	0-ffh	0-ffh	Crc16

Byte	Content
1	Slave ID (0 = DA48-1 Controller)
2	BGR address = controller address
3,4	CPU Version
5,6	serial number
7,8	power supply voltage
9,10	number of discrete outputs
11,12	res.
13,14	number of analog outputs
15,16	res.
17,18	number of discrete intputs
19,20	res.
21,22	number of input registers
23, 24	res.
25, 26	number of holding registers
27,28	module 1, type, grouping, sequence
29, 30	module 2, type, grouping, sequence
31, 32	module 3, type, grouping, sequence
33, 34	module 4, type, grouping, sequence
35,36	res.
37,38	startregister analog outputs
39,40	endregister analog outputs
41,42	startregister discrete outputs 4bit group
43,44	endregister discrete outputs 4bit group
45,46	startregister discrete outputs 6bit group
47,48	endregister discrete outputs 6bit group
49,50	Reset-Counter
51	COM (RS485) Speed
52	COM (RS485) Datalength
53	COM (RS485) Stopbit
54	COM (RS485) Parity
55	COM (RS422) Speed
56	COM (RS422) Datalength
57	COM (RS422) Stopbit
58	COM (RS422) Parity
59	Res.
60	BGR-Mode (1 = default = Modbus RTU)
61	BGR-Status (1=Factory Setup, + 3 = calibrated = normal) …

JÄGER Datentechnik KG
Rhönstr. 24 · 63110 Rodgau
fon: 06106-259 4660
email: info@jaeger-dt.de

40H PROG COM (assembly group specific)

41H COM RESET

By the command **40H** the interface parameters of the assembly group(s) can be changed. By the command **41H** the communication modules of the assembly group(s) with the programmed parameters are restarted subsequent the command **40H**.

Attention: The BUS Master must be also changed subsequent to assure communication.

The latest programmed modification is annulled automatically if the command **COM RESET** is not sent within 10 seconds subsequent the command **PROG COM**.
This commands can be send in Broadcast-Mode.

After sending **COM RESET** the Master may not send another command within 5 seconds.
The successful modification is signalized by a longer HB-LED flashing (~5 sec.).
The modificated parameters are saved in the BGR and are relevant again after the re-start.

Please take the relevant parameters of the overview table "Holding Registers".

1. Request Master -> Slave

Address	Function	Data									CRC-Check
		Count	485 Speed	485 DataLe	485 StopB	485 Parity	422 Speed	422 DataL	422 StopB	422 Parity	
Adr/ (0)	40H	1 Byte 0x08	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	Crc16

1. Answer Slave – Master (not in case address 0)

Address	Function	Data									CRC-Check
		Count	485 Speed	485 DataLe	485 StopB	485 Parity	422 Speed	422 DataL	422 StopB	422 Parity	
Adr	40H	1 Byte 0x08	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	Crc16

2. Request Master -> Slave (within 10 sec.)

Address	Function	Data			CRC-Check
		Check1	Check2	Check3	
Adr/ (0)	41H	0x88	0x55	0x33	Crc16

If this command is sent after 10 seconds or this command is sent without the previous one, the slave answers with an error code.
Currently the RS422 parameters are not relevant. But they must be attached correctly anyway because this way every single parameter is proofed.

6.0 Error Management

Is the CRC of the telegram (calculated by the slave) not similar to the transfered CRC, no receipt is sent and a time out is enforced.

Does the Slave determined that invalid data (e.g. wrong register addresses, contents or function codes have been sent), it sends back an error message to the Master.

Answer Slave – Master

Address	Function	Data	CRC-Check
addr	Code + 80H	Error code	Crc16

Error code	decription
01h	Unsupported function code
02h	Application of unauthorized or not existend register
03h	Unauthorized data values, e.g. wrong number of registers
06h	Device cannot be processed on request currently. Repeat request later.

A CRC-Error is indicated by the error code "CC" on the display.

7.0 CRC-Generation

(C-example)

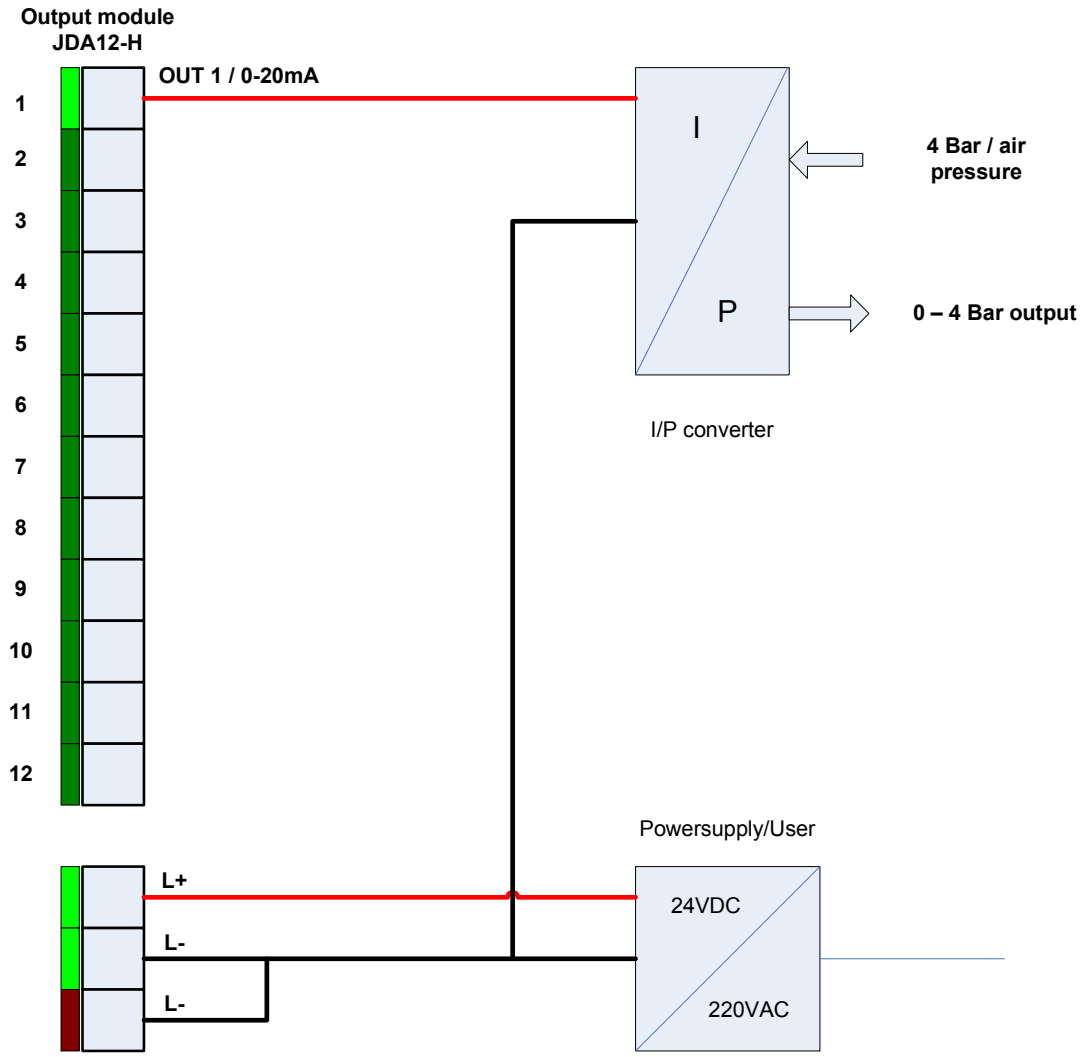
```
WORD Mod_crc_rtu(BYTE * buf, WORD len)
{
        WORD crc ,n;
        BYTE i;
        crc = 0xffff;

        for(n=0;n<len-2;n++)
        {
          crc = crc^((* buf++)&0xff);
          for(i=8;i>0;i--)
          {
              if(crc&1)
              {
                 crc=crc>>1;
                 crc=crc^0xA001;
              }
              else crc=crc>>1;
          }
        }
        return (crc);
}
```

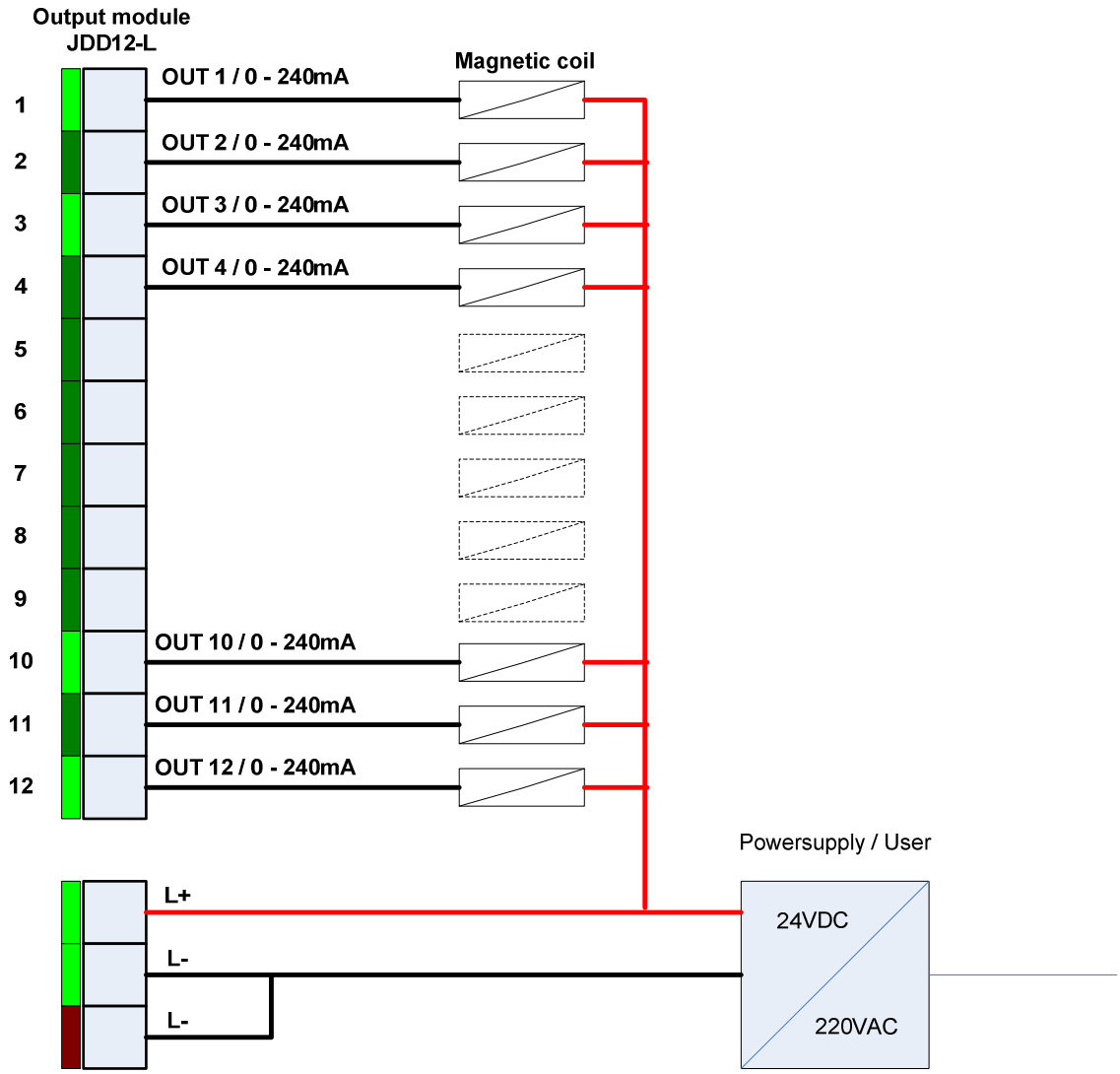
Attention! The CRC – Word is send in the opposite way as data values. Starting with low byte first.

8.0 Connection configuration

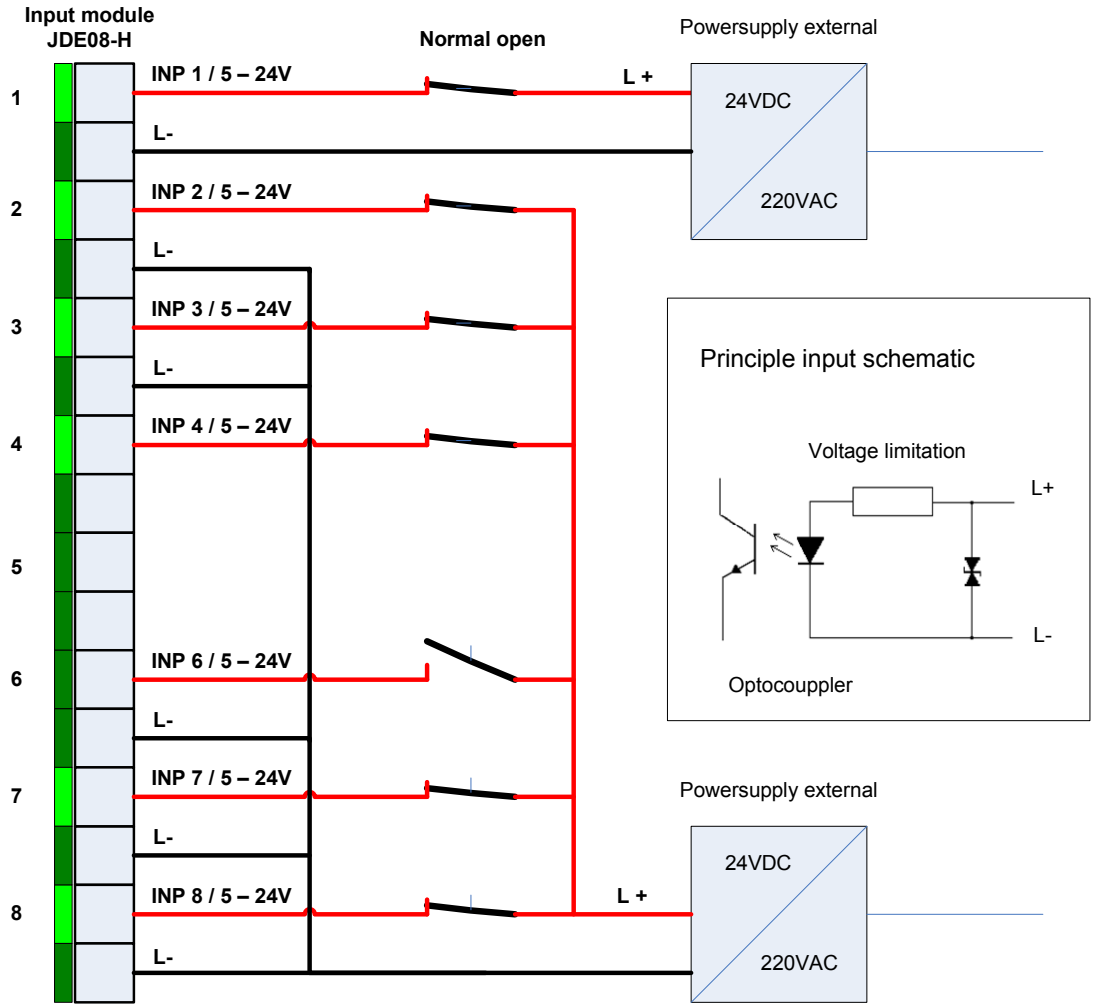
9.0 Connecting to an analog module (JDA12-H)



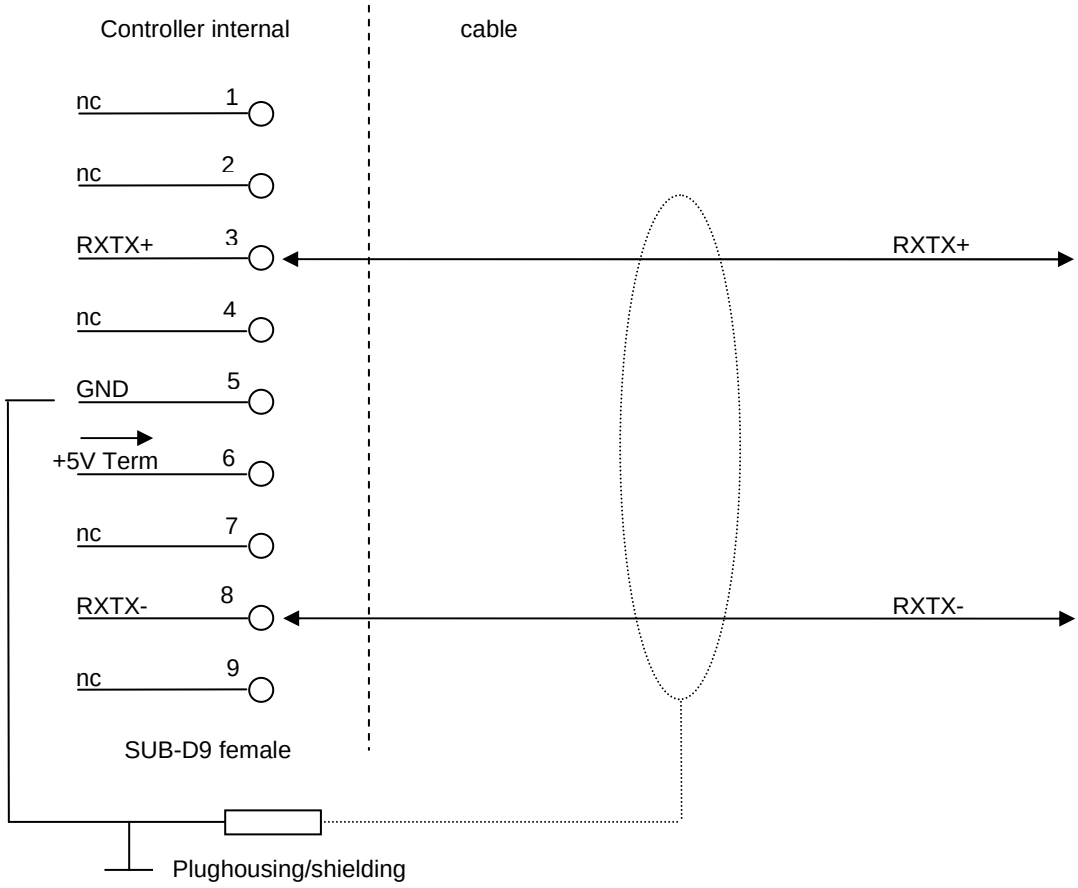
10.0 Connecting to a digital module (JDA12-L)



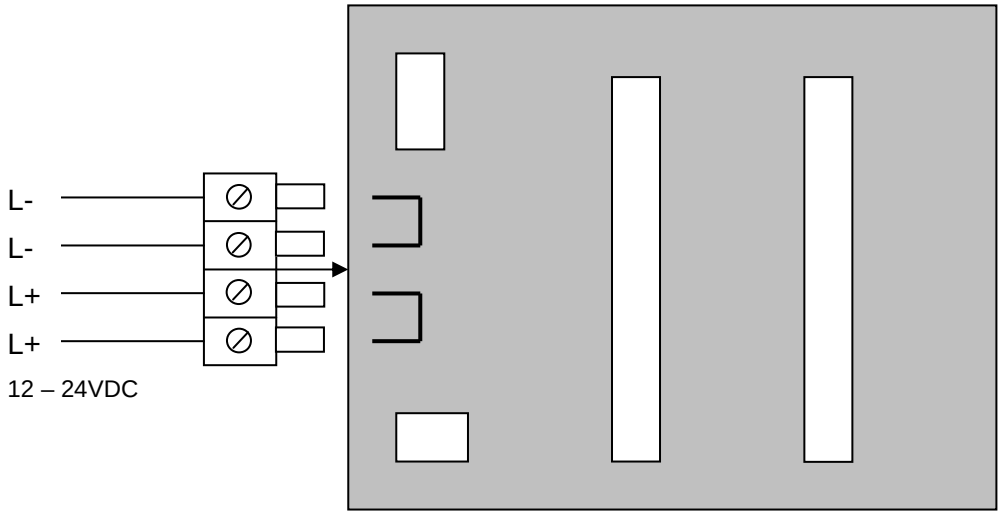
11.0 Connecting to an input module (JDE08-H)



12.0 Connection diagram R S485 SUB-D9 female



13.0 Connection diagram controller supply



14.0 Programming example

Setting a communication watchdog on DA48-1 with address 1

If there is no communication within 10 s module 1 should be deactivated.

1. Set watchdog time (10000ms) in register **40262**
2. Send the data by the command **PRESET SINGLE REGISTER**
3. Values:

Register(-1) = **105_h**
Value = **2710_h**

Send:

```
01 06 01 05 27 10 crlc_l crc_h
```

Response:

```
01 06 01 05 27 10 crlc_l crc_h
```

4. Set WDT-code in register **40263**
5. Value:

Register(-1) = **106_h**
Value = **51_h**

Send:

```
01 06 01 06 00 51 crlc_l crc_h
```

Response:

```
01 06 01 06 00 51 crlc_l crc_h
```

6. Watchdog is active, wait until 10 seconds passed

DA48-1 displays **"CE"**, module 1 is shut-off

(optional)

7. Read watchdog status of register **40263** by sending **READ HOLDING REGISTER**

Send:

```
01 03 01 06 00 01 crlc_l crc_h
```

Response:

```
01 03 02 51 51 crlc_l crc_h
```

Watchdog timer is startet again

DA48-1 displays **"01"** (Address), module 1 is still shut-off

Module 1 needs to be reactivated by sending new data.

Setting output values on DA48-1 with address 1

Module 1 and 2 of DA48-1 are digital output modules type JDD12-AC.
Each module should by switched to 50% output with possible grouping by 1.

1. Set group of module 1 and 2 to 1 in register **40200+40261.**
   This is only necessary one time, because this parameters are stored in th DA48-1 and available at power on again.

2. Send the data by the command **PRESET MULTIPLE REGISTER**
3. Values:

Start register(-1)	=	**$C7_h$**
No of registers	=	**02_h**
No of bytes	=	**04_h**
Values	=	**01_h**

Send:

```
01 10 00 C7 00 02 04 00 01 00 01 crlc_l crc_h
```

Response:

```
01 10 00 C7 00 02 crlc_l crc_h
```

4. Set data in registers **40050 - 40055**
5. Value:

Start Register(-1)	=	**31_h**
No of registers	=	**06_h**
No of bytes	=	**$0C_h$**
Values	=	**1388_h**

Send:

```
01 10 00 31 00 06 00 0C 13 88 13 88 13 88 13 88 13 88 13 88 crlc_l crc_h
```

Response:

```
01 10 00 31 00 06 crlc_l crc_h
```

The output of modules switch to 50% in six 4 Bit groups.

Attention! The program must wait for the response before sending a new command.
This is identical for all commands.
If the command is send with address 0 all connected slaves are activated and no response is send.

--- end of document ---