EXDUL-516E

EDP No.: A-374340

EXDUL-516S

EDP No.: A-374320

10 optocoupler isolated digital inputs
8 optocoupler isolated digital outputs
1 16-bit-counter
LCD Display (EXDUL-516E only)

Common ground connection at outputs

wasco®

User‘s Guide

wasco®

Registered Trademarks

Windows®, Visual Basic®, Visual C++®, Visual C#® are registered trademarks of Microsoft.
wasco® is registered trademark.
EXDUL® is registered trademark.

Disclaimer

The information in this manual is intended to be accurate and reliable. The company Messcomp Datentechnik GmbH does not assume any liability for the use of the multi functional module EXDUL-516 and this documentation, neither for direct nor indirect damages.

Important Information:

This manual was made up for modules EXDUL-516E and EXDUL-516S. EXDUL-516E additionally provides an LCD display, all other functions are identical. All commands and functions concerning the LCD display are not applicable for EXDUL-516S.

wasco®

Table of Contents

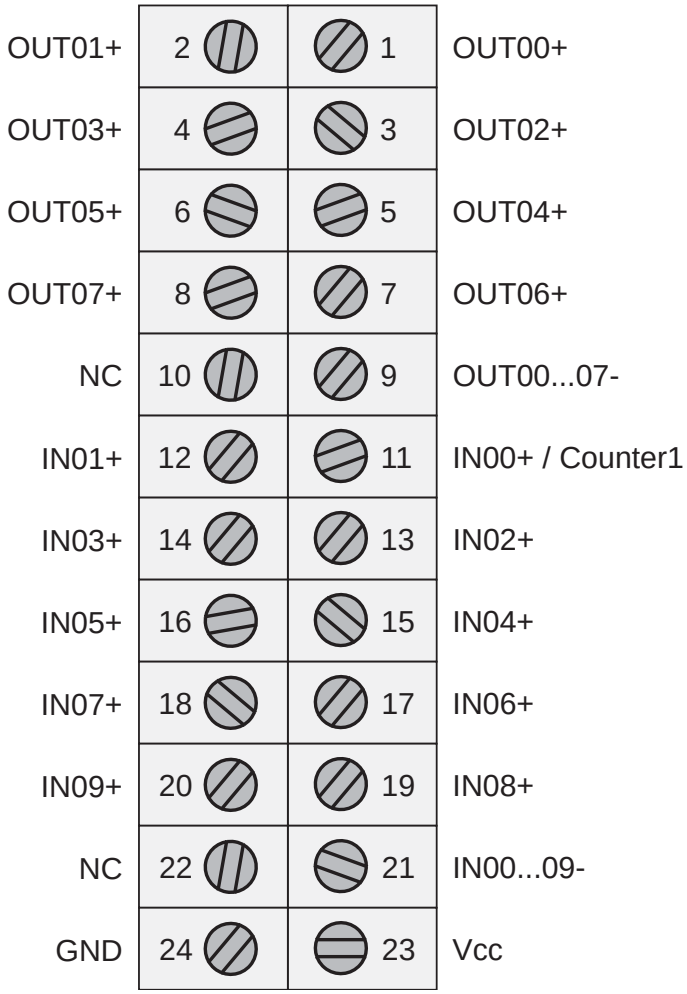
Table of Contents

1. Introduction

EXDUL-516E and EXDUL-516S are network-compatible digital I/O modules with Ethernet interface. Each module provides 10 digital inputs and 8 digital outputs galvanically isolated by high-quality optocouplers and is equipped with additional protection diodes. All input optocouplers have integrated schmitt trigger function. Special high power output optocouplers manage a maximum switching current of up to 150 mA. One of the ten input optocouplers is programmable and usable as a digital counter if required. EXDUL-516E provides additionally an LCD display to show I/O status information or user specific data. The module can be connected easily and conveniently to a network or PC in a plug and play manner via an Ethernet interface. The connections for the necessary external power supply as well as the connections for the input and output optocouplers are led to a 24-pin screw terminal block. The compact chassis enables the module to be used as a portable device with a notebook. For mechanical engineering control applications it also allows easy wall mounting or uncomplicated clipping on DIN-EN mounting rails.

2. Connection Terminals

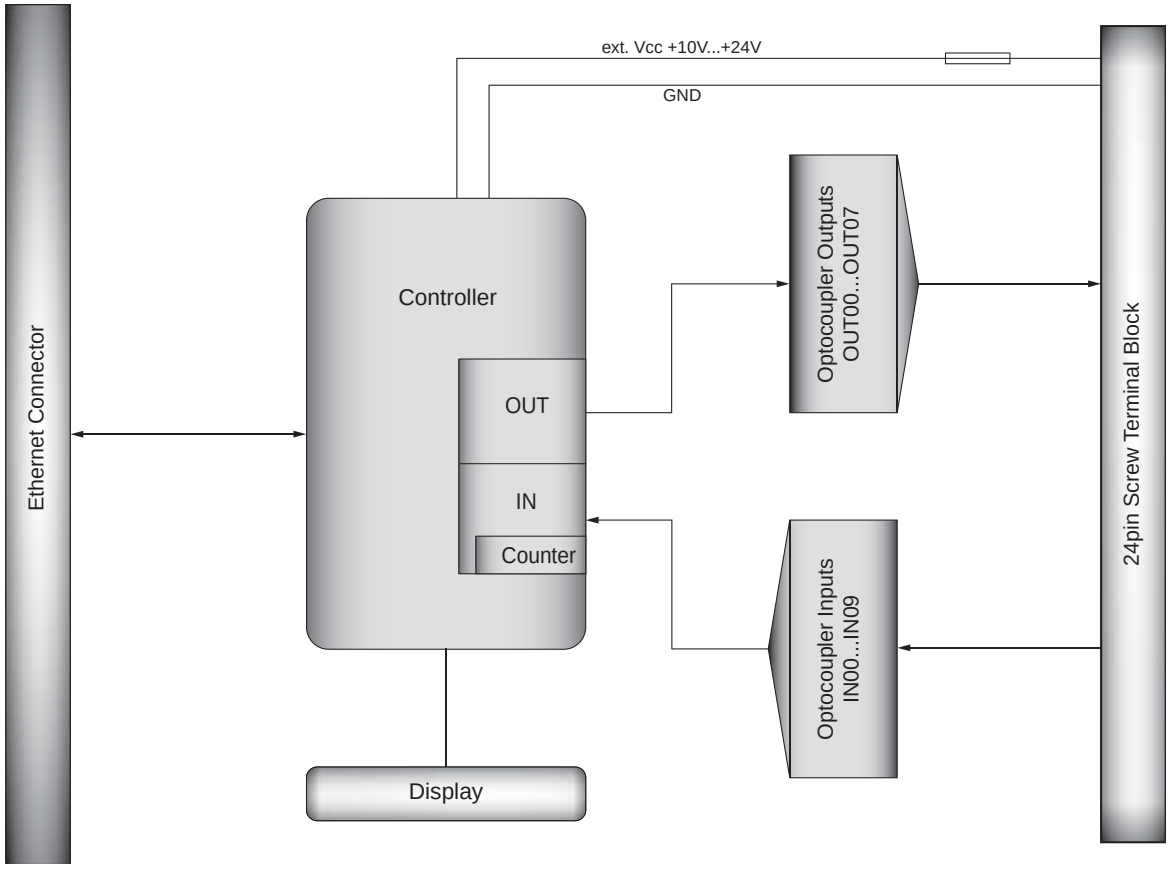
2.1 Terminal Layout of CN1



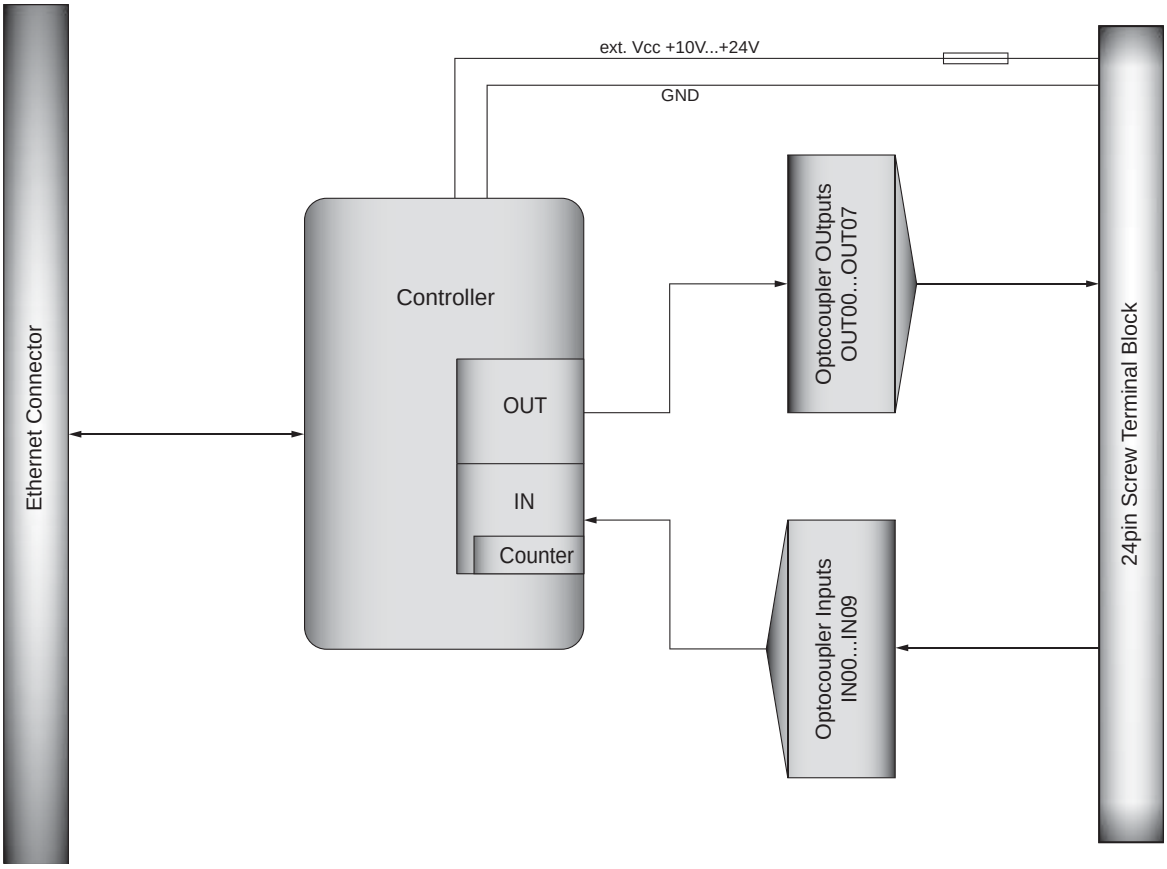
Vcc:
Connector for external supply voltage
GND:
Ground connection when using external supply voltage
NC:
not connected

3. System Components

3.1 Block Diagram EXDUL-516E



3.2 Block Diagram EXDUL-516S



3.3 Input Optocoupler

10 channels, galvanically isolated
Common ground connection (cathodes of the optocouplers connected)
1 of the channels progammable to be a digital counter
Optocouplers with integrated schmitt trigger function
Overvoltage protection diodes
Input voltage range: low = 0....3 Volt high = 10.....30 Volt
Input frequency: max. 10 kHz

3.4 Output Optocoupler

8 channels, galvanically isolated
Common plus connection (collectors of the optocouplers connected)
High capacity optocouplers
Reverse polarity protection
Output current: max. 150mA
Switching voltage: max. 50 V

3.5 Counter

1 programmable 16-bit counter
(one of the input optocouplers is assigned)
Counting frequency: max. 5 kHz

3.6 LCD Display (EXDUL-516E only)

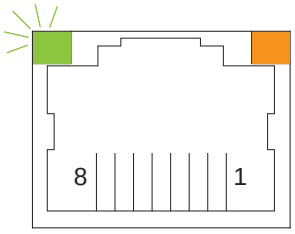
Matrix display with 2 lines and 16 columns performing 16 signs each line
Info display during booting-up
I/O status display or user LCD display during operation

4. First Start-up

Connection to a network or to a computer is made easily and conveniently via an Ethernet interface, the configuration is made via any web browser. An external power supply unit is required for the necessary current supply.

4.1 Connection to an Ethernet Port

The module EXDUL-516E / EXDUL-516S provides a 10Base-T Ethernet interface with RJ45 connection (8P8C modular connector) and you can connect directly to either a PC, Ethernet hub or Ethernet switch using a network cable. If you connect the module to a switch, hub or PC with Ethernet interface able to control Auto MDI(X) you can use a standard straight through network cable such as Cat5 or higher. Older computers whose Ethernet interfaces do not automatically crossover transmitting and recieving lines may require a crossover cable or crossover adapter.
After the operating voltage is applied the module will boot up. Once a stable connection is established, the green LED on the left side of the RJ45 jack is lit continuously.



4.2 Connecting Supply Voltage

The EXDUL-516E / EXDUL-516S requires a voltage supply from +10 V ... +24 V DC across terminal 23 (Vcc) and terminal 24 (GND).

4.3 Integrated ModPage of EXDUL-516E / EXDUL-516S

Any web or internet browser such as Mozilla Firefox, Internet Explorer, Safari etc. can access to the ModPage of EXDUL-516 by TCP/IP connection. The ModPage enables to read connection information and modify configuration data under password protection. Modifications made will be stored in the built-in EEPROM of EXDUL-516 and will be loaded during booting process. The EXDUL-516 ModPage enables you to write, read-out and show user specific memory area UserA, UserB, UserLCD1 and UserLCD2 as well as to start and stop the counter or to test digital inputs and outputs.

4.4 Password Protection - Access Code

As mentioned before using the EXDUL ModPage enables to configure the network, to set up the LCD display, to write into user specific memory areas as well as to set up inputs and outputs or counter. To prevent unauthorised access these setting sections are password protected.

Following default access code is preset:

User name: admin
Password: 11111111

Please regard upper and lower case!

If it is not possible to access with this access code, your system administrator changed settings of the access code.

4.5 Basic Settings of Network Configuration

In the basic setting the EXDUL-516 is set to DHCP (Dynamic Host Configuration Protocol), i.e. it creates a dynamic IP address. During commissioning the EXDUL-516 sends a request to the network (LAN). A network with active DHCP service will automatically assign an IP address to the module. This setting allows an easy and convenient connection of the module and a proper adjustment of the configuration data to your own requirements.

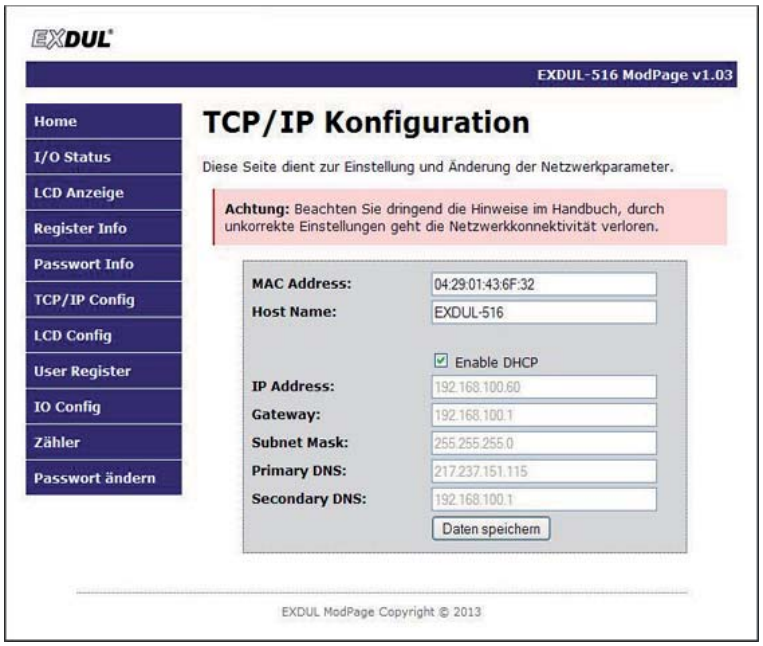
4.6 Composition and Structure of IP Address

IP4 addresses consist of 32 Bits = 4 Bytes (octets). Each Byte can range from 0 through 255. It is shown as dotted four decimal number blocks (e.g. 192.168.1.83).
Each IP address contains a network number portion and a device number portion (host identifier). A Subnet mask divides network part and host part. All devices located in the same network can communicate with each other.

Example:
If Subnet mask 255.255.0.0 is allocated to the IP address 192.168.1.83, so the device is located in network 192.168.-.- and is named -.-.1.83.

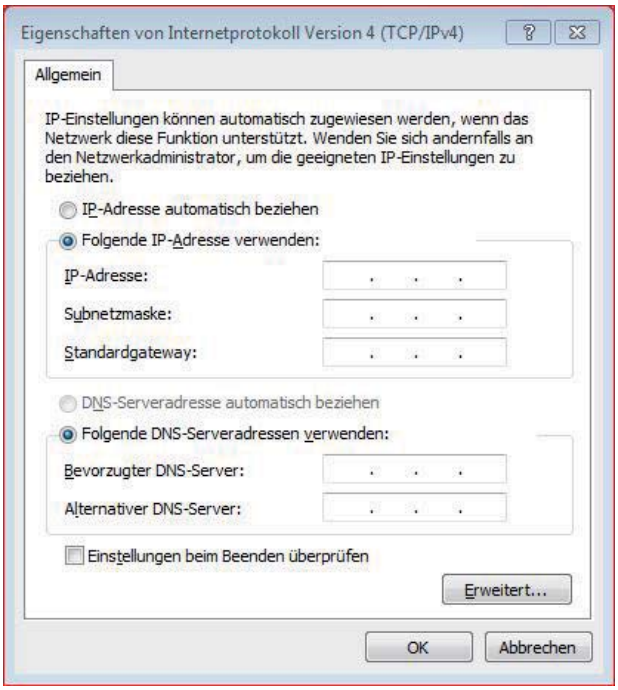
4.7 Change of Network Configuration

To change the configuration set-up by default plug the enclosed standard network cable of the EXDUL 516 into a local network and connect voltage supply (+10 V ...+24 V) across terminal 23 (Vcc) and terminal 24 (GND) of the module.The EXDUL will boot up immediately. Once a stable connection is established, the green LED on the left side of the RJ45 jack is lit green continuously. Access to the EXDUl-516 ModPage is possible using any web or internet browser. Host name then is **http://EXDUL-516.** If you click the button **TCP/IP Config** following configuration screen will open:



You can use the enclosed standard network cable to connect EXDUL-516 directly to a PC with an Auto MDI(X) controlling Ethernet interface. Older computers may need a crossover cable or crossover adapter.
If you connect directly, normally no DHCP service is at your disposal, for neither the PC nor the EXDUL provides one. In this case the last set static IP address can be used for adressing the EXDUL-516. Factory-new devices of EXDUL are shipped with an IP Address of 169.254.1.1.

Important Information:
The PC used and the cable-connected EXDUL are to be located in the same network to be able to communicate. For this purpose the computer‘s DHCP has to be deactivated and the first two number blocks (192.168) of EXDUL IP address 192.168.1.83 (Subnet mask 255.255.0.0) are to be transferred to the computer‘s IP address. The next two number blocks can be assigned with values between 0 through 255 adjusted to your current personal requirements.

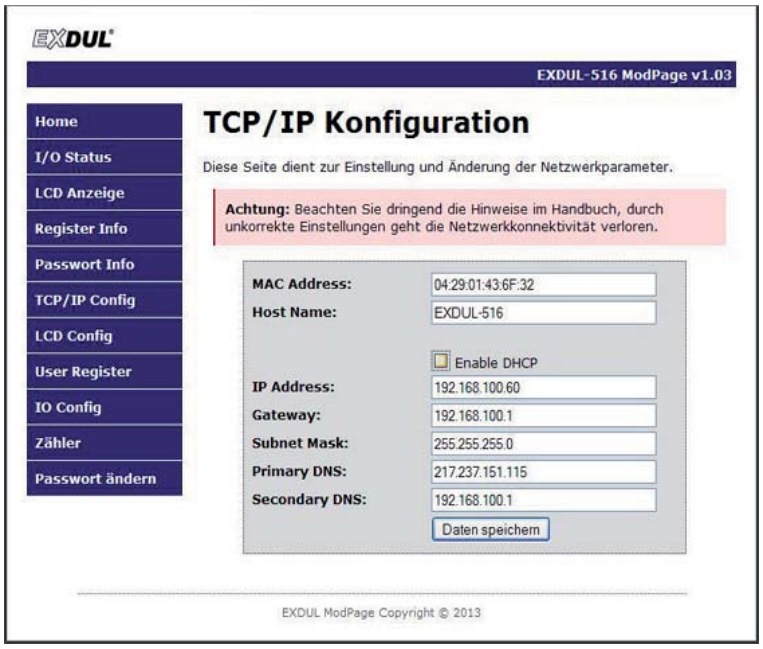


The net part of the IP address depends on the Subnet mask. All octets of the IP address of Exdul-516, which in the Subnet mask are assigned with 255, are to be transferred to the IP address of the computer (see chapter 4.6 Composition and structure of IP address)

4.8 Configuration with Static IP Address (DHCP deactivated)

To configure EXDUL-516 to the static IP address DHCP has to be deactivated in EXDUL-516 configuration screen. To change configuration connect the EXDUL-516 to a LAN or PC as per description in chapter 4.7 (Change network configuration).

Once a stable Ethernet connection is established you can access the ModPage of the EXDUL-516 via any Internet browser. When you enter the host name **http://EXDUL-516**, the browser should open the ModPage. Click on the button **TCP/IP Config** to open following configuration screen:



Once having disabled DHCP you can enter IP address, Subnet mask and desired host name. A click on the button **Konfiguration speichern** takes over all currently registered data in the internal memory of the EXDUL-516. From that point the module only can be addressed by the IP address registered here, or by the indicated host name respectively. PC used or LAN are to be located within the same network, too.

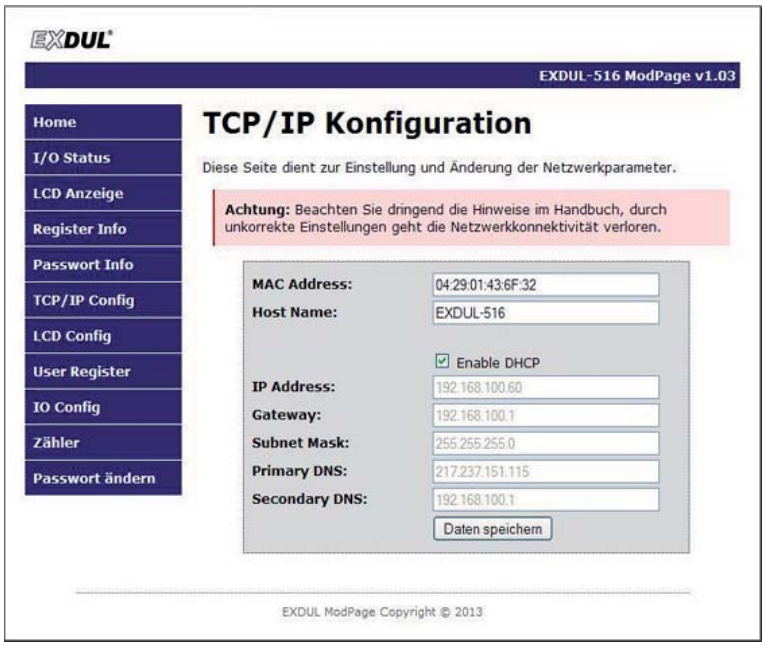
Important information: Each device or module in a network must have its individual IP address or host name, double assignments are not permitted! Any host name can be choosen, but it has to be composed of ASCII characters 0 to 9 as well as A to Z (leave out upper and lower case) and - (hyphen).
Some IP addresses are reserved or have special functions as for example 127.0.0.1 (local host)
192.168.1.0 (0 defines the network address) with 255.255.255.0 (Subnet mask).
Please consult your network administrator which IP address you may use.

If you use an impermissible IP address, the access to the module might be not possible any more. The setting of the most important impermissible addresses will be blocked by the module.

4.9 Configuration with Dynamic IP Address (DHCP activated)

If you want to embed EXDUL-516 in an existing network with already active DHCP server and to access via a dynamic IP address DHCP (Dynamic Host Configuration Protocol) has to be activated in the configuration screen of the ModPage. In basic setting by default DHCP is activated already and no re-setting is necessary.

If the IP address is set statically activate DHCP as follows:
Connect EXDUL-516 to a computer using a network cable (older computers may need a crossover cable). Please make sure that the computer is set on: **Folgende IP-Adresse verwenden:** (DHCP-deaktiviert)
The network part of the IP address (see chapter 4.6 Composition and structure of IP address) has to be overtaken from the IP address of the EXDUL-516, for the computer and the EXDUL are to be located in the same network.
Connect voltage supply (+10 V ...+24 V) across terminal 23 (Vcc) and terminal 24 (GND) of the module.The EXDUL will boot up immediately. Once a stable connection is established, the LED on the left side of the RJ45 jack is lit green continuously.
Access to the EXDUl-516 ModPage is possible using any internet browser entering host name or IP address of EXDUL-516**.** The browser now should open the EXDUL ModPage. If you click the button **TCP/IP Config** the configuration screen will open and you can acitivate DHCP by setting the checkmark ☑ **Enable DHCP**



Important Information:
We strongly recommend to consult your network administrator before activating DHCP. If you want to embed several modules of the series EXDUL-5xx in one network, the presetted host names are to be changed. Each host name within a network must be assigned to only one device or module. Any host name can be choosen, but it has to be composed of ASCI characters 0 to 9 as well as A to Z (leave out upper and lower case) and - (hyphen).

4.10 LCD Display during the Booting Process (EXDUL-516E only)

During booting-up the module the display shows information data. Line 1 indicates the module name. If the IP address is set dynamically in TCP/IP configuration (DHCP activated = basic setting by default) line 2 temporarily indicates the last set static IP address, subsequently the dynamic adress assigned by the DHCP server.
If the IP address is set statically in TCP/IP configuration line 2 indicates the effectively set address.
After having finished the booting process in both cases the display shows either I/O status information or UserLCD information depending upon your configuration.

4.11 LCD Display during Operation (EXDUL-516E only)

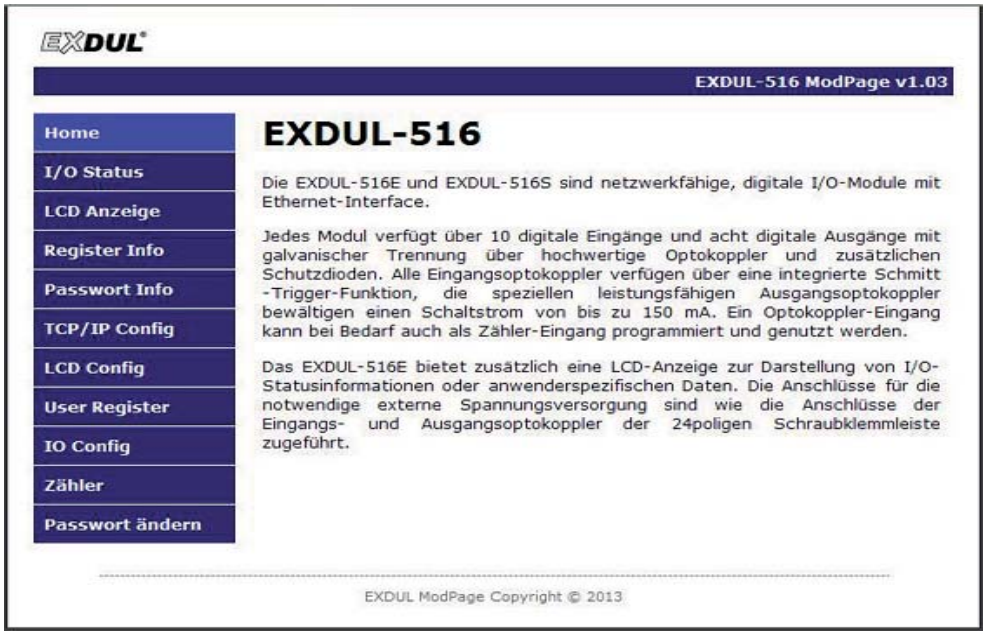
After having finished the booting process the display switches from information display to I/O status display or UserLCD display. If the I/O status is displayed line 1 indicates the current states of the inputs, line 2 those of the outputs. If UserLCD mode is activated in EXDUL-516 ModPage the display shows UserLCD data with values from memory areas UserLCD1m and UserLCD2m instead of I/O status display.
You can see the values of UserLCD1m and UserLCD2m on the LCD display unless you write new user data in UserLCD line1 and line2 of the LCD display. To avoid „screen-burn“ while in operation the display alternates from I/O status display to UserLCD display for 5 seconds every minute.

5. Access to EXDUL-516

As mentioned before the access to the configuration set-up and to the inputs and outputs of the EXDUL-516 is possible via EXDUL-516 ModPage or TCP/IP socket. Therefore you need to know IP address, host name or MAC address.

5.1 Access via EXDUL ModPage

The ModPage of the EXDUL-516 allows to read inputs, to set outputs, to read out UserA, UserB and UserLCD area as well as to read out information about details of the connection or the module, or to change configuration data. You can access to the ModPage via any internet browser with any computer connected to the module. The computer in use has to be set to „**IP-Adresse automatisch beziehen (DHCP-aktiviert)“** if the module still is in delivery status (DHCP activated) and if it is integrated in a network with effective DHCP service. Entering the host name (at delivery status **http://EXDUL-516,** alternatively the name you choosed, or you might determine it via Exdul Ethernet Discoverer) you can open the ModPage. If you don‘t succeed in opening please check the network connections or the entered host name. More details see chapter FAQ - trouble shooting.

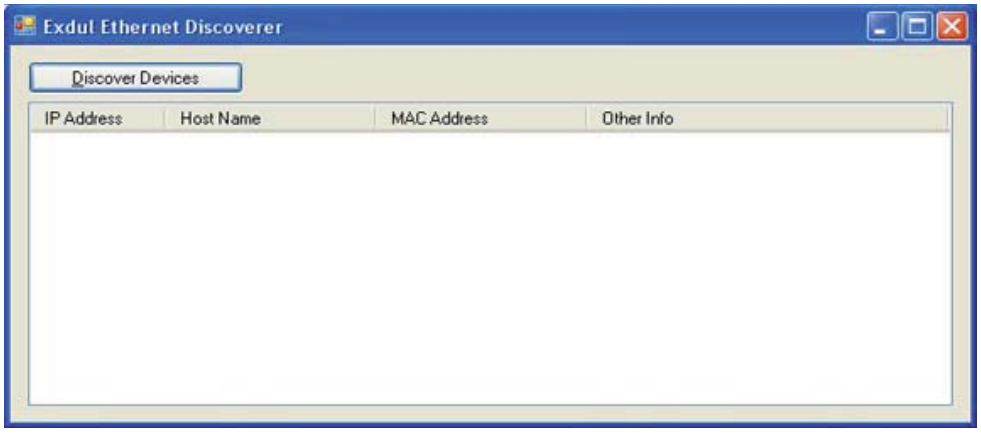


5.2 Access via TCP/IP Sockets

Using TCP protocol a reliable connection between computer and EXDUL-516 will be achieved. The protocol will take independent actions in case of data loss. The module is addressed by a 4 Byte IP address (IPV4) or the allocated host name respectively and a port number x. The computer transmits a 52-byte string for every command via this connection. The module processes the command and always will send a response. In combination with a high level language the TCP/IP connection allows to read the inputs, to set the outputs, to start, to stop and to read-out the counter, to write user specific memory areas and to change configuration setting as well.

5.3 How to Identify Host Name, IP Address and MAC Address

If you don‘t know neither host name nor IP address or Mac address of the EXDUL-516, the search tool Exdul Ethernet Discoverer enables you to detect these addresses. If your Firewall prevents communication between the search program and EXDUL-5xx it is necessary to grant access to the program.



The search tool **Exdul Ethernet Discoverer** is provided on the EXDUL software CD or available for download at www.wasco.de.

6. Programming

You can program EXDUL under Windows via so-called TCP/IP sockets, for which many commonly used programming languages provide standard libraries. To use .Net Frameworks of Microsoft permits a quick and easy implementation. You can find several programming examples on the enclosed CD and on our website.

6.1 Register HW Identification and Serial Number

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HW identification	E	X	D	U	L	-	5	1	7	v	1	.	0	2		
	45_{hex}	58_{hex}	44_{hex}	55_{hex}	$4C_{hex}$	$2D_{hex}$	35_{hex}	31_{hex}	37_{hex}	76_{hex}	31_{hex}	$3E_{hex}$	30_{hex}	32_{hex}	20_{hex}	20_{hex}
S/N	1	0	4	4	0	2	6									
	01_{hex}	00_{hex}	04_{hex}	04_{hex}	00_{hex}	02_{hex}	06_{hex}	FF_{hex}	FF_{hex}	FF_{hex}	FF_{hex}	FF_{hex}	FF_{hex}	FF_{hex}	FF_{hex}	FF_{hex}

In the register HW identification the module name as well as the firmware version are stored and can be read by the user to verify the product identity. Hardware identifier ends with a blank. The table above shows every Hex value and the corresponding ASCII character in the row HW identification, in this example for EXDUL-516 with firmware version 1.02.

The register serial number only can be read by the user. The serial number in the table above serves as a format example. Row S/N shows every Hex value and the corresponding ASCII character, here for serial number 1044026.

6.2 Memory Areas UserA, UserB, UserLCD1m* and UserLCD2m*

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
UserA																
	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}
UserB																
	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}
UserLCD1m*																
	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}
UserLCD2m*																
	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}	20_{hex}

In each register UserA, UserB, UserLCD1m* and UserLCD2m* 16 digits (16 Byte) can be used for own purposes. The data remains stored when turning off. Factory settings (delivery status) can be restored by a default reset. In factory setting in all of the four user memory areas the Hex value is 20, corresponding to a blank in ASCII characters. The top table shows every Hex value and the corresponding ASCII character in the row above.

After having started the module in activated UserLCD mode the data from the memory register UserLCD1m* and UserLCD2m* is displayed as long as the user doesn‘t write new information into UserLCD line1 and UserLCD line2 in LCD display.

6.3 Display Register UserLCD line1*, UserLCD line2* and LCD Contrast*

In UserLCD activated mode the register UserLCD line1 and UserLCD line2 are used to write any 16 characters into each of the two lines of the LCD display. When you take over the data, the display shows it instead of the data from UserLCD1m and UserLCD2m. The data in the register UserLCD line1 and UserLCD line2 remains **not** stored when turning off.
In register LCD contrast you can set display contrast. Setting will be stored when turning off.

*: EXDUL-516E only

6.4 Command and Data Format

The data exchange is achieved by transmitting and receiving descrete strings.
Each transmitting or receiving string consists of 52 elements in ANSI format (1 Byte each character), beginning with a „!“ (Byte 0) and ending with a „$“ (Byte 51). For each transmitting string the header (Byte 0 ... 20) will show the number of digits to be transferred (Byte 1 and 2), the Job-ID User (Byte 3 and 4) and password / access code (Byte 11..18). Bytes 21 through 24 contain the command code, while Bytes 32 through 47 are reserved for data transfer.

6.5 Structure of the Header

Byte 0	Start code „!“
Byte 1	Length of string (High-Byte)
Byte 2	Length of string (Low-Byte)
Byte 3	Job-ID User (High-Byte)
Byte 4	Job-ID User (Low-Byte)
Byte 5..10	Reserved
Byte 11..18	Password / access code
Byte 19..20	Reserved

*: EXDUL-516E only

Length of string: number of elements (Bytes) which are sent within a transmitting or receiving string, for the EXDUL-516 52 Bytes are fixed.
Job-ID User: In larger networks not all of the strings may arrive in the same correct order as they were sent. In each string there are two Bytes intended for Job-ID User to allocate the strings.

6.6 Command Index

Hexcode	Description
0C 00 00 00	write UserA
0C 00 00 01	read UserA
0C 00 00 02	write UserB
0C 00 00 03	read UserB
0C 00 03 07	write UserLCD1m
0C 00 03 09	read UserLCD1m
0C 00 03 08	write UserLCD2m
0C 00 03 0A	read UserLCD2m
0C 00 03 00	write UserLCD line1
0C 00 03 02	read UserLCD line1
0C 00 03 01	write UserLCD line2
0C 00 03 03	read UserLCD line2
0C 00 03 04	enable UserLCD Mode
0C 00 03 05	read status UserLCD Mode
0C 00 03 0B	write LCD contrast
0C 00 03 0C	read LCD contrast
0C 00 04 01	read HW identier
0C 00 05 01	read serial number
0C 00 0C 01	change password
0C 00 0E 00	write IP address + Subnet mask
0C 00 0E 01	read IP address + Subnet mask
0C 00 0E 04	write host name
0C 00 0E 05	read host name

*: EXDUL-516E only

0C 00 0E 06	write Gateway + DNS
0C 00 0E 07	read Gateway + DNS
0C 00 0E 08	read MAC address
0C 00 0E 09	DHCP enable / disable
0C 00 0E 0A	DHCP state
08 00 01 01	read Optocoupler input port
08 00 00 00	write Optocoupler output port
08 00 00 01	read Optocoupler output port (status request)
09 00 00 00	start counter0
09 00 00 01	stop counter0
09 00 00 02	read state of counter0
09 00 00 03	read counter0
0C 00 0C 0E	reset (User)
0C 00 0C 0F	reset factory setting (set delivery status)

6.7 Command Composition

6.7.1 Write in User Area A and B

Example: write the string EXDUL-516 to register UserA and UserB

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	00	00	Command code 3rd Byte
24	00 (UserA) 02 (UserB)	00 (UserA) 02 (UserB)	Command code 4th Byte
25...31			Reserved
32	45	45	Data 1st digit E_{asci}
33	58	58	Data 2nd digit X_{asci}
34	44	44	Data 3rd digit D_{asci}
35	55	55	Data 4th digit U_{asci}
36	4C	4C	Data 5th digit L_{asci}
37	2D	2D	Data 6th digit $-_{asci}$
38	35	35	Data 7th digit 5_{asci}
39	31	31	Data 8th digit 1_{asci}
40	36	36	Data 9th digit 6_{asci}
41	20	20	Data 10th digit $[blank]_{asci}$
42	20	20	Data 11th digit $[blank]_{asci}$
43	20	20	Data 12th digit $[blank]_{asci}$
44	20	20	Data 13th digit $[blank]_{asci}$
45	20	20	Data 14th digit $[blank]_{asci}$
46	20	20	Data 15th digit $[blank]_{asci}$
47	20	20	Data 16th digit $[blank]_{asci}$
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.2 Read out User area A und B

Example: read the string EXDUL-516 in register UserA and UserB

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	00	00	Command code 3rd Byte
24	01 (UserA) 03 (UserB)	01 (UserA) 03 (UserB)	Command code 4th Byte
25...31			Reserved
32	xx	45	Data 1st digit E_{asci}
33	xx	58	Data 2nd digit X_{asci}
34	xx	44	Data 3rd digit D_{asci}
35	xx	55	Data 4th digit U_{asci}
36	xx	4C	Data 5th digit L_{asci}
37	xx	2D	Data 6th digit $-_{asci}$
38	xx	35	Data 7th digit 5_{asci}
39	xx	31	Data 8th digit 1_{asci}
40	xx	36	Data 9th digit 6_{asci}
41	xx	20	Data 10th digit $[blank]_{asci}$
42	xx	20	Data 11th digit $[blank]_{asci}$
43	xx	20	Data 12th digit $[blank]_{asci}$
44	xx	20	Data 13th digit $[blank]_{asci}$
45	xx	20	Data 14th digit $[blank]_{asci}$
46	xx	20	Data 15th digit $[blank]_{asci}$
47	xx	20	Data 16th digit $[blank]_{asci}$
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.3 Write in UserLCD1m* and UserLCD2m*

Example: Write the character string EXDUL-516 in register UserLCD1m* and UserLCD2m*

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	03	03	Command code 3rd Byte
24	07 (UserLCD1m) 08 (UserLCD2m)	07 (UserLCD1m) 08 (UserLCD2m)	Command code 4th Byte
25...31			Reserved
32	45	45	Data 1st digit E_{asci}
33	58	58	Data 2nd digit X_{asci}
34	44	44	Data 3rd digit D_{asci}
35	55	55	Data 4th digit U_{asci}
36	4C	4C	Data 5th digit L_{asci}
37	2D	2D	Data 6th digit $-_{asci}$
38	35	35	Data 7th digit 5_{asci}
39	31	31	Data 8th digit 1_{asci}
40	36	36	Data 9th digit 6_{asci}
41	20	20	Data 10th digit $[blank]_{asci}$
42	20	20	Data 11th digit $[blank]_{asci}$
43	20	20	Data 12th digit $[blank]_{asci}$
44	20	20	Data 13th digit $[blank]_{asci}$
45	20	20	Data 14th digit $[blank]_{asci}$
46	20	20	Data 15th digit $[blank]_{asci}$
47	20	20	Data 16th digit $[blank]_{asci}$
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.4 Read out UserLCD1m* and UserLCD2m*

Example: read the string EXDUL-516 from the register UserLCD1m* and UserLCD2m*

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	03	03	Command code 3rd Byte
24	09 (UserLCD1m) 0A (UserLCD2m)	09 (UserLCD1m) 0A (UserLCD2m)	Command code 4th Byte
25...31			Reserved
32	xx	45	Data 1st digit E_{asci}
33	xx	58	Data 2nd digit X_{asci}
34	xx	44	Data 3rd digit D_{asci}
35	xx	55	Data 4th digit U_{asci}
36	xx	4C	Data 5th digit L_{asci}
37	xx	2D	Data 6th digit $-_{asci}$
38	xx	35	Data 7th digit 5_{asci}
39	xx	31	Data 8th digit 1_{asci}
40	xx	36	Data 9th digit 6_{asci}
41	xx	20	Data 10th digit $[blank]_{asci}$
42	xx	20	Data 11th digit $[blank]_{asci}$
43	xx	20	Data 12th digit $[blank]_{asci}$
44	xx	20	Data 13th digit $[blank]_{asci}$
45	xx	20	Data 14th digit $[blank]_{asci}$
46	xx	20	Data 15th digit $[blank]_{asci}$
47	xx	20	Data 16th digit $[blank]_{asci}$
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.5 Write in UserLCD1* and UserLCD2*

Example: Write the string EXDUL-516 in UserLCD1* resp. UserLCD2*

Byte	Transmit	Reply	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	03	03	Command code 3rd Byte
24	00 (UserLCD1) 01 (UserLCD2)	00 (UserLCD1) 01 (UserLCD2)	Command code 4th Byte
25...31			Reserved
32	45	45	Data 1st digit E_{asci}
33	58	58	Data 2nd digit X_{asci}
34	44	44	Data 3rd digit D_{asci}
35	55	55	Data 4th digit U_{asci}
36	4C	4C	Data 5th digit L_{asci}
37	2D	2D	Data 6th digit $-_{asci}$
38	35	35	Data 7th digit 5_{asci}
39	31	31	Data 8th digit 1_{asci}
40	36	36	Data 9th digit 6_{asci}
41	20	20	Data 10th digit $[blank]_{asci}$
42	20	20	Data 11th digit $[blank]_{asci}$
43	20	20	Data 12th digit $[blank]_{asci}$
44	20	20	Data 13th digit $[blank]_{asci}$
45	20	20	Data 14th digit $[blank]_{asci}$
46	20	20	Data 15th digit $[blank]_{asci}$
47	20	20	Data 16th digit $[blank]_{asci}$
48...50			Reserved for error code/error detection
51	24	24	end identifier $\$_{asci}$

*: EXDUL-516E only

6.7.6 Read from UserLCD1* and UserLCD2*

Example: read the string EXDUL-516 from UserLCD1* resp. UserLCD2*

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	03	03	Command code 3rd Byte
24	02 (UserLCD1) 03 (UserLCD2)	02 (UserLCD1) 03 (UserLCD2)	Command code 4th Byte
25...31			Reserved
32	xx	45	Data 1st digit E_{asci}
33	xx	58	Data 2nd digit X_{asci}
34	xx	44	Data 3rd digit D_{asci}
35	xx	55	Data 4th digit U_{asci}
36	xx	4C	Data 5th digit L_{asci}
37	xx	2D	Data 6th digit $-_{asci}$
38	xx	35	Data 7th digit 5_{asci}
39	xx	31	Data 8th digit 1_{asci}
40	xx	36	Data 9th digit 6_{asci}
41	xx	20	Data 10th digit $[blank]_{asci}$
42	xx	20	Data 11th digit $[blank]_{asci}$
43	xx	20	Data 12th digit $[blank]_{asci}$
44	xx	20	Data 13th digit $[blank]_{asci}$
45	xx	20	Data 14th digit $[blank]_{asci}$
46	xx	20	Data 15th digit $[blank]_{asci}$
47	xx	20	Data 16th digit $[blank]_{asci}$
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.7 Write UserLCD-Mode

Example: Enable UserLCD mode

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	03	03	Command code 3rd Byte
24	04	04	Command code 4th Byte
25...31			Reserved
32	01	01	01 = enable / 00 = disable
33...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.8 Read UserLCD-Mode

Example: UserLCD mode is set enable

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	03	03	Command code 3rd Byte
24	05	05	Command code 4th Byte
25...31			Reserved
32	xx	01	01 = enable / 00 = disable
33...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.9 Read Hardware Identification

Example: Read of hardware identification EXDUL-516V1.02

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	04	04	Command code 3rd Byte
24	01	01	Command code 4th Byte
25...31			Reserved
32	xx	45	Data 1st digit E_{asci}
33	xx	58	Data 2nd digit X_{asci}
34	xx	44	Data 3rd digit D_{asci}
35	xx	55	Data 4th digit U_{asci}
36	xx	4C	Data 5th digit L_{asci}
37	xx	2D	Data 6th digit $-_{asci}$
38	xx	35	Data 7th digit 5_{asci}
39	xx	31	Data 8th digit 1_{asci}
40	xx	36	Data 9th digit 6_{asci}
41	xx	76	Data 10th digit V_{asci}
42	xx	31	Data 11th digit 1_{asci}
43	xx	2E	Data 12th digit $._{asci}$
44	xx	30	Data 13th digit 0_{asci}
45	xx	32	Data 14th digit 2_{asci}
46	xx	20	Data 15th digit $[blank]_{asci}$
47	xx	20	Data 16th digit $[blank]_{asci}$
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.10 Read the Serial Number

Example: read the serial number 1044026

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	05	05	Command code 3rd Byte
24	01	01	Command code 4th Byte
25...31			Reserved
32	xx	01	Data 1st digit 1_{dez}
33	xx	00	Data 2nd digit 0_{dez}
34	xx	04	Data 3rd digit 4_{dez}
35	xx	04	Data 4th digit 4_{dez}
36	xx	00	Data 5th digit 0_{dez}
37	xx	02	Data 6th digit 2_{dez}
38	xx	06	Data 7th digit 6_{dez}
39...47	xx	20	Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.11 Read Optocoupler Input Port

Example: read the inputs at the optocoupler input port. Precondition for this example are the voltage input levels (0 = Low = 0...3 V; 1 = High = 10...30 V) to be set on each input according following table:

Input channel	IN09	IN08	IN07	IN06	IN05	IN04	IN03	IN02	IN01	IN00
Screw terminal	20	19	18	17	16	15	14	13	12	11
Input level	1	0	1	1	1	1	0	0	1	1
Display*	E	A	E	E	E	E	A	A	E	E

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	08	08	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	01	01	Command code 3rd Byte
24	01	01	Command code 4th Byte
25...31			Reserved
32	xx	02	Read value (Highbyte - 00...03)
33	xx	F3	Read value (Lowbyte - 00...FF)
34...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.12 Write Optocoupler Output Port

Example: interconnection of optocouplers to channel OUT02, OUT03, OUT04 and OUT06, (optocouplers connected through = 1; optocouplers not connected through = 0)

Output channel	OUT07	OUT06	OUT05	OUT04	OUT03	OUT02	OUT01	OUT00
Screw terminal	8	7	6	5	4	3	2	1
Switching state	0	1	0	1	1	1	0	0
Display*	A	E	A	E	E	E	A	A

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	08	08	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	00	00	Command code 3rd Byte
24	00	00	Command code 4th Byte
25...31			Reserved
32	5C	5C	Transfer value (00...FF)
33...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.13 Readback Optocoupler Output Port (Status Request)

Example: interconnection of optocouplers to channel OUT02, OUT03, OUT04 und OUT06, (optocouplers connected through = 1; optocouplers not connected through = 0)

Output channel	OUT07	OUT06	OUT05	OUT04	OUT03	OUT02	OUT01	OUT00
Screw terminal	8	7	6	5	4	3	2	1
Switching state	0	1	0	1	1	1	0	0
Display*	A	E	A	E	E	E	A	A

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	08	08	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	00	00	Command code 3rd Byte
24	01	01	Command code 4th Byte
25...31			Reserved
32	xx	5C	Transfer value (00...FF)
33...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.14 Start Counter

Every start command resets the counter to 0 and it then will begin to count upwards, with a counting range of 0...65535.

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	09	09	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	00	00	Command code 3rd Byte
24	00	00	Command code 4th Byte
25...31			Reserved
32...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.15 Readback Start of Counter (Status Request)

Example: counter started

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	09	09	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	00	00	Command code 3rd Byte
24	02	02	Command code 4th Byte
25...31			Reserved
32	xx	01	01 = counter started (00 = counter stopped)
33...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.16 Stop Counter

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	09	09	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	00	00	Command code 3rd Byte
24	01	01	Command code 4th Byte
25...31			Reserved
32...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.17 Read Counter Status

Example 1: Read counter status 2047 (without overflow)

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	09	09	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	00	00	Command code 3rd Byte
24	03	03	Command code 4th Byte
25...31			Reserved
32	xx	00	Flag for overflow (set when counting range is overrun)
33	xx	07	Read value (Highbyte - 00...FF)
34	xx	FF	Read value (Lowbyte - 00...FF)
35...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

Counter status = read value High-Byte x 256 + read value Low-Byte

Example 2: Read counter status 2047 counting range overrun (with overflow)

Byte	Transmit	Response	Description
32	xx	01	Flag for overflow (set when counting range is overrun)
33	xx	07	Read value (Highbyte - 00...FF)
34	xx	FF	Read value (Lowbyte - 00...FF)

Counter status = read value High-Byte x 256 + read value Low-Byte

6.7.18 Write LCD Contrast Value*

Calling this command you can adjust display contrast. Values are accepted from 0 up to 4095, the display contrast will reduce the more the value increases. Comfortable display contrast will be achieved with values ranging from 800 up to 1800.

Example: Display contrast value 800

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	03	03	Command code 3rd Byte
24	0B	0B	Command code 4th Byte
25...31			Reserved
32	03	03	Read value (Highbyte - 00...FF)
33	50	50	Read value (Lowbyte - 00...FF)
34...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

Contrast value = transfer value High-Byte x 256 + transfer value Low-Byte (03 50 = 800)

Example: Display contrast value 1800

Byte	Transmit	Response	Description
32	07	07	Transfer value (High-Byte - 00...0F)
33	08	08	Transfer value (Low-Byte - 00...FF)

Contrast value = transfer value High-Byte x 256 + transfer value Low-Byte (07 08 = 1800)

*: EXDUL-516E only

6.7.19 Read LCD Contrast Value*

Example: Display contrast value 800

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	03	03	Command code 3rd Byte
24	0C	0C	Command code 4th Byte
25...31			Reserved
32	xx	03	Read value (Highbyte - 00...0F)
33	xx	50	Read value (Lowbyte - 00...FF)
34...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

Contrast value = transfer value High-Byte x 256 + transfer value Low-Byte (03 50 = 800)

Example: Display contrast value 1000 (factory setting at delivery)

Byte	Transmit	Response	Description
32	xx	03	Transfer value (High-Byte - 00...0F)
33	xx	E8	Transfer value (Low-Byte - 00...FF)

Contrast value = transfer value High-Byte x 256 + transfer value Low-Byte (03 E8 = 1000)

*: EXDUL-516E only

6.7.20 Write Host Name

Example: write host name EXDUL-516

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	04	04	Command code 4th Byte
25...31			Reserved
32	45	45	Data 1st digit E_{asci}
33	58	58	Data 2nd digit X_{asci}
34	44	44	Data 3rd digit D_{asci}
35	55	55	Data 4th digit U_{asci}
36	4C	4C	Data 5th digit L_{asci}
37	2D	2D	Data 6th digit $-_{asci}$
38	35	35	Data 7th digit 5_{asci}
39	31	31	Data 8th digit 1_{asci}
40	36	36	Data 9th digit 6_{asci}
41	20	20	Data 10th digit $[blank]_{asci}$
42	20	20	Data 11th digit $[blank]_{asci}$
43	20	20	Data 12th digit $[blank]_{asci}$
44	20	20	Data 13th digit $[blank]_{asci}$
45	20	20	Data 14th digit $[blank]_{asci}$
46	20	20	Data 15th digit $[blank]_{asci}$
47			Reserved
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

This command requires a reboot / reset of the module.

*: EXDUL-516E only

6.7.21 Read Host Name

Example: read host name EXDUL-516

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	05	05	Command code 4th Byte
25...31			Reserved
32	xx	45	Data 1st digit E_{asci}
33	xx	58	Data 2nd digit X_{asci}
34	xx	44	Data 3rd digit D_{asci}
35	xx	55	Data 4th digit U_{asci}
36	xx	4C	Data 5th digit L_{asci}
37	xx	2D	Data 6th digit $-_{asci}$
38	xx	35	Data 7th digit 5_{asci}
39	xx	31	Data 8th digit 1_{asci}
40	xx	36	Data 9th digit 6_{asci}
41	xx	20	Data 10th digit $[blank]_{asci}$
42	xx	20	Data 11th digit $[blank]_{asci}$
43	xx	20	Data 12th digit $[blank]_{asci}$
44	xx	20	Data 13th digit $[blank]_{asci}$
45	xx	20	Data 14th digit $[blank]_{asci}$
46	xx	20	Data 15th digit $[blank]_{asci}$
47			Reserved
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.22 Write IP Address and Subnet Mask

Example: Write IP address 192.168.0.83

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	00	00	Command code 4th Byte
25...31			Reserved
32	C0	C0	1st Byte IP address decimal value 192
33	A8	A8	2nd Byte IP address decimal value168
34	00	00	3rd Byte IP address decimal value 0
35	53	53	4th Byte IP address decimal value 83
36	FF	FF	1st Byte Subnetmask decimal value 255
37	FF	FF	2nd Byte Subnetmask decimal value 255
38	FF	FF	3rd Byte Subnetmask decimal value 255
39	00	00	4th Byte Subnetmask decimal value 0
40...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

This command requires a reboot / reset of the module.

6.7.23 Write IP Address and Subnet Mask

Example: read IP address 192.168.0.83

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	01	01	Command code 4th Byte
25...31			Reserved
32	xx	C0	1st Byte IP address decimal value 192
33	xx	A8	2nd Byte IP address decimal value168
34	xx	00	3rd Byte IP address decimal value 0
35	xx	53	4th Byte IP address decimal value 83
36	xx	FF	1st Byte Subnetmask decimal value 255
37	xx	FF	2nd Byte Subnetmask decimal value 255
38	xx	FF	3rd Byte Subnetmask decimal value 255
39	xx	00	4th Byte Subnetmask decimal value 0
40...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.24 Write Gateway and DNS

Example:
Write Gateway 192.168.0.1, Primary DNS 192.168.0.1, Secondary DNS 217.237.151.115

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	00	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	06	06	Command code 4th Byte
25...31			Reserved
32	C0	C0	Gateway 1th Byte decimal value 192
33	A8	A8	Gateway 2nd Byte decimal value 168
34	00	00	Gateway 3rd Byte decimal value 0
35	01	01	Gateway 4th Byte decimal value 1
36	C0	C0	Primary DNS 1st Byte decimal value 192
37	A8	A8	Primary DNS 2nd Byte decimal value 168
38	00	00	Primary DNS 3rd Byte decimal value 0
39	01	01	Primary DNS 4th Byte decimal value 1
40	D9	D9	Secondary DNS 1st Byte decimal value 217
41	ED	ED	Secondary DNS 2nd Byte decimal value 237
42	97	97	Secondary DNS 3rd Byte decimal value 151
43	73	73	Secondary DNS 4th Byte decimal value 115
44...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

This command requires a reboot / reset of the module.

*: EXDUL-516E only

6.7.25 Read Gateway und DNS

Example:
Read Gateway 192.168.0.1, Primary DNS 192.168.0.1, Secondary DNS 217.237.151.115

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	00	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	07	07	Command code 4th Byte
25...31			Reserved
32	xx	C0	Gateway 1st Byte decimal value 192
33	xx	A8	Gateway 2nd Byte decimal value 168
34	xx	00	Gateway 3rd Byte decimal value 0
35	xx	01	Gateway 4th Byte decimal value 1
36	xx	C0	Primary DNS 1st Byte decimal value 192
37	xx	A8	Primary DNS 2nd Byte decimal value 168
38	xx	00	Primary DNS 3rd Byte decimal value 0
39	xx	01	Primary DNS 4th Byte decimal value 1
40	xx	D9	Secondary DNS 1st Byte decimal value 217
41	xx	ED	Secondary DNS 2nd Byte decimal value 237
42	xx	97	Secondary DNS 3rd Byte decimal value 151
43	xx	73	Secondary DNS 4th Byte decimal value 115
44...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

*: EXDUL-516E only

6.7.26 Read MAC Address

Example: Read MAC address 00:04:A3:C0:BE:AF

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	08	08	Command code 4th Byte
25...31			Reserved
32	xx	00	MAC address 1st Byte 00_{hex}
33	xx	04	MAC address 2nd Byte 04_{hex}
34	xx	A3	MAC address 3rd Byte $A3_{hex}$
35	xx	C0	MAC address 4th Byte $C0_{hex}$
36	xx	BE	MAC address 5th Byte BE_{hex}
37	xx	AF	MAC address 6th Byte AF_{hex}
38...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.27 DHCP Enable/Disable

Example: DHCP status

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	09	09	Command code 4th Byte
25...31			Reserved
32	01 (enable) 00 (disable)	01 (enable) 00 (disable)	01 = activate DHCP 00 = deactivate DHCP
33...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

This command requires a reboot / reset of the module.

6.7.28 DHCP Status

Example: DHCP effective

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0E	0E	Command code 3rd Byte
24	0A	0A	Command code 4th Byte
25...31			Reserved
32	xx	01	01 = DHCP activated 00 = DHCP deactivated
33...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.29 How to Change Access Code (Password)

To prevent unauthorisized access to the module it is highly recommended to change the factory presetted password. The new access code should consist of an 8 digit combination of letters and numbers using upper and lower case. Capital letters (A-Z), lower case letters (a-z) and numbers (0-9) are permitted. Presetted login name (admin) is not changeable.

Example: New password Exdul516

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0C	0C	Command code 3rd Byte
24	01	01	Command code 4th Byte
25...31			Reserved
32	45	45	1st digit E_{asci}
33	78	78	2nd digit X_{asci}
34	64	64	3rd digit D_{asci}
35	75	75	4th digit U_{asci}
36	6C	6C	5th digit L_{asci}
37	35	35	6th digit 5_{asci}
38	31	31	7th digit 1_{asci}
39	36	36	8th digit 6_{asci}
40...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

6.7.30 Reset

Description: User settings keep reserved

Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0C	0C	Command code 3rd Byte
24	0E	0E	Command code 4th Byte
25...31			Reserved
32...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

Caution: the module does not send a response!

6.7.31 Factory Resetting

Description: restores the basic status

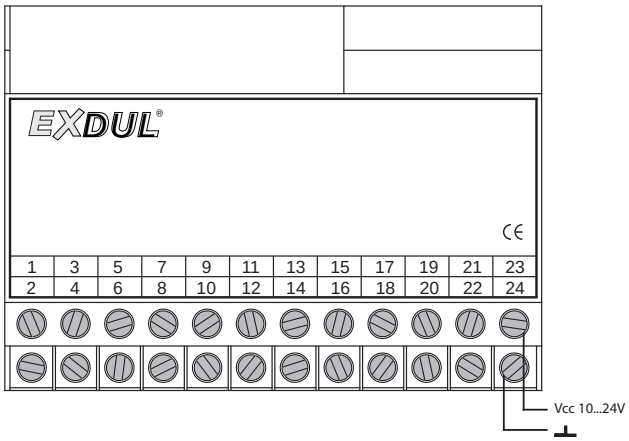
Byte	Transmit	Response	Description
0	21	21	Start identifier $!_{asci}$
1	0	0	Length of the string (High-Byte)
2	52	52	Length of the string (Low-Byte)
3	xx	xx	Job-ID (High-Byte)
4	xx	xx	Job-ID (Low-Byte)
5...10			Reserved
11...18	xx	xx	Password / access code
19...20	xx	xx	Reserved
21	0C	0C	Command code 1st Byte
22	00	00	Command code 2nd Byte
23	0C	0C	Command code 3rd Byte
24	0F	0F	Command code 4th Byte
25...31			Reserved
32...47			Reserved, without relevance for this command
48...50			Reserved for error code/error detection
51	24	24	End identifier $\$_{asci}$

Caution: the module does not send a response!

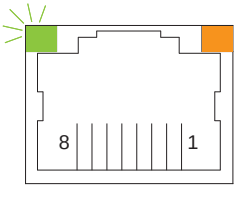
7. FAQ - Trouble Shooting

You can find below a short summary of the most and common causes of error, which can appear during commissioning or access to the EXDUL-516 or to the EXDUL ModPage. Please firstly check following points before contacting your distributor:

Is the power voltage supplyof the EXDUL516 connected properly?
For operating a voltage supply from +10 V ... +24 V DC across terminal 23 (Vcc) and terminal 24 (GND) is required. Please check the screw terminal connections on the module as well as the power supply unit and connections to the power supply.



Is the connection LED lit green continuously on the RJ45 jack?
Having applied the operating voltage, the EXDUL-516 will boot up. Once a stable Ethernet connection is established, the LED on 8P8C module jack (RJ45 jack) is lit green continuously.
If not, please check the direct cable connection between EXDUL-516 and your computer (a crossover cable might be necessary), if network is operating, check network connections between EXDUL-516 and wall-mounted network socket, effective Ethernet switch or Ethernet hub.



Is there a stable connection between computer and network ?
Check network cable between EXDUL-516 and network socket (wall-mounted RJ45 jack), effective Ethernet switch or Ethernet hub. The Ethernet cable has to be suitable for the Ethernet connnection, it may not be damaged and properly connected onto both sides. Current computers mostly provide two LED‘s on the network jack of the network adapter. If connection to the network is established, the green LED is lit continuously. If the network jack of the computer provides only one LED it will flash or flicker when network connection is working.

Is the used network cable suitable for the connection?
If you connect EXDUL-516 to a switch, hub or PC with an Auto MDI-X commanding Ethernet interface, a standard network cable (Cat5 or higher) can be used. Older computer whose Ethernet interface doesn‘t automatically cross transmitting and receiving lines, may need a crossover cable or crossover adapter.

Is the wall-mounted network jack effective?
If you connect the EXDUL-516 to a permanent installed network via a wall socket please check together with your network administrator, if the wall socket is active and connected with an effective Ethernet switch or Ethernet hub.

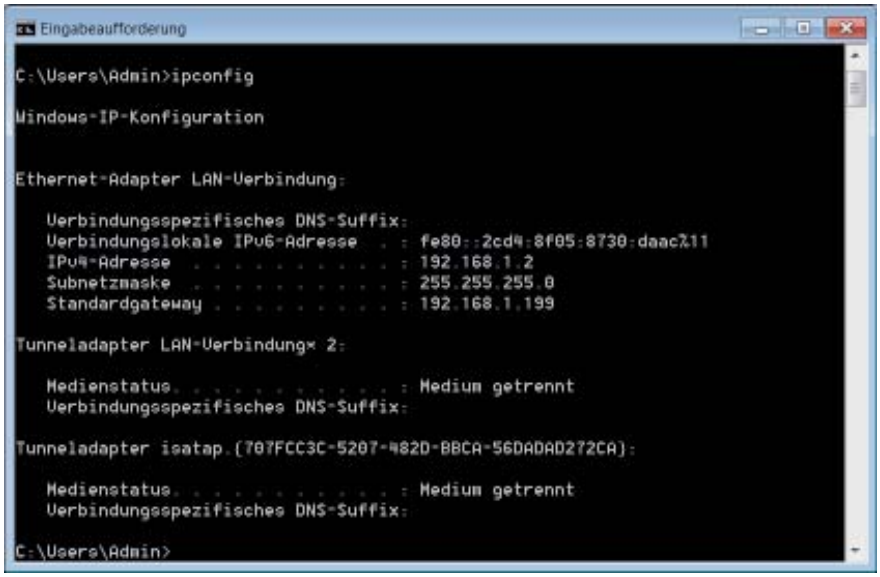
Is the computer‘s Ethernet interface activated?
The Ethernet adapter has to be activated in the BIOS of your computer. Please check the Windows Device manager whether the adapter is listed under network adapters. The entry must not be tagged with an exclamation mark!

Is the computer‘s network configuration set correctly?
Each effective device in a TCP/IP based network needs an unique IP address, which is composed of a network ID and a device ID. The Subnet mask separates the IP address in a network portion (network prefix, network ID) and a device portion (device ID). In the basic setting the EXDUL-516 is adjusted to DHCP client. To access the EXDUL-516 the PC in use has to be set to **IP-Adresse automatisch beziehen (DHCP-aktiviert)** and the network has to provide a DHCP server (mostly integrated in routers). The DHCP server automatically allocates a Subnet mask and an IP address to the EXDUL-516 and to the computer. If the basic setting was changed to static IP address, the used computer has to be set to static address (**Folgende IP-Adresse verwenden**), too.

Example: The static IP address for the EXDUL-516 is registered as 192.168.1.199 (network ID: 192.168.1., device ID 199). To access the EXDUL-516 the used PC has to be set to Subnet mask 255.255.255.0 and to static IP address ranging from 192.168.1.**1** through 192.168.1.**244**

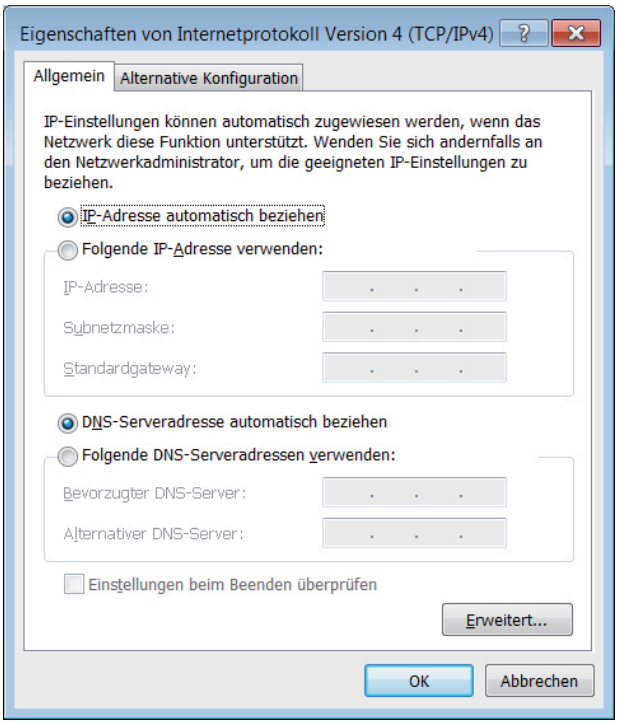
How to check network configuration of the PC

You can figure out the TCP/IP settings of your computer via the window Internet Protocol Version 4 (TCP/IPv4) or status of the LAN connection respectively (see „How to check or to change IP address of the PC“). Alternatively type the simple command IPCONFIG to the command-line. Thereto change to MS-DOS prompt (see „How to change to MS-DOS prompt“), type **ipconfig** and press enter to confirm. The response should look similar to the picture as shown below:



How to check or to change IP address of the PC

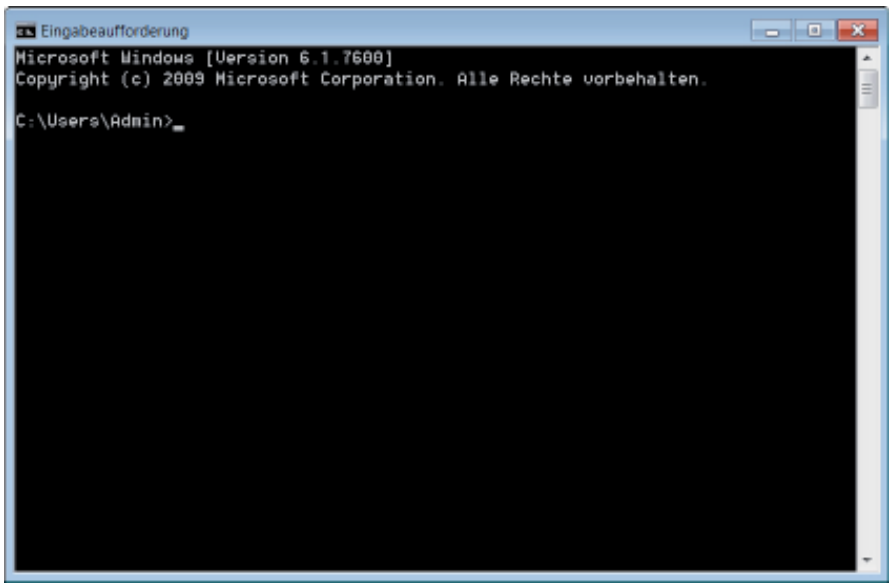
Windows7:
Start -> control panel -> Network and sharing center (display network status and tasks) -> change adapter settings -> select required LAN connection in the window network connections (double-click or right mouse button) -> properties -> highlight Internet Protocol Version 4 (TCP(IPV4) -> properties
Please note: you have to own administrator privileges for changing TCP/IP settings!



WindowsXP:
Start -> control panel -> network connections (network and Internet connections) -> select required LAN connection (double-click or right mouse button) -> properties -> highlight Internet Protocol (TCP/IP) -> properties
Please note: you have to own administrator privileges for changing TCP/IP settings!

How to change to MS-DOS prompt
Windows7:
Start -> type in **cmd** in input box (program and file searching) -> press enter to confirm
or
Start -> All Programs -> Accessories -> prompt

WindowsXP:
Start -> Run -> type in **cmd** in run box-> OK
or
Start -> All Programs -> Accessories -> prompt



Is it possible to locate modules EXDUL-5xx in an existing network and to detect the network data?
All EXDUL-5xx modules transmit detection signals in certain temporal intervals. The program **Exdul Ethernet Discoverer** evaluates the identification data and compiles a list with Host name, IP address and MAC address. It is suitable for any single EXDUL-5xx directly connected to a PC as well as for a network with several modules connected via Hub or Switch. In case the firewall inhibits the communication between searching program and EXDUL-5xx an approval to your firewall is necessary.

8. Specifications

Digital inputs by optocoupler

Channels	10 inputs galvanically isolated common ground connection (cathodes connected) 1 of the channels programmable as counting input
galvanic isolation	optocoupler with integrated Schmitt-Trigger function
over voltage protection	diodes
input voltage range	high = 10 .... 30 V low = 0 .... 3 V
input frequency	max. 10 kHz

Digital outputs by optocoupler

Channels	8 outputs galvanically isolated common ground connection (emitter connected)
galvanic isolation	high-capacity optocoupler
reverse polarity protection	diodes
output current	max. 150mA
switching voltage	max. 50 V

Counter

Channel	1 programmable counter 16 Bit (1 of the 10 input optocoupler is assigned)
counting frequency	max. 5 kHz

LCD display (EXDUL-516E only)

Display	Matrix display with 2 lines and 16 columns displaying 16 characters on each line
display modes	Info display during booting process I/O status display or UserLCD display during operation

Operating voltage

External voltage source:	+10 V...+24 V

Ethernet Port

10Base-T Ethernet Interface

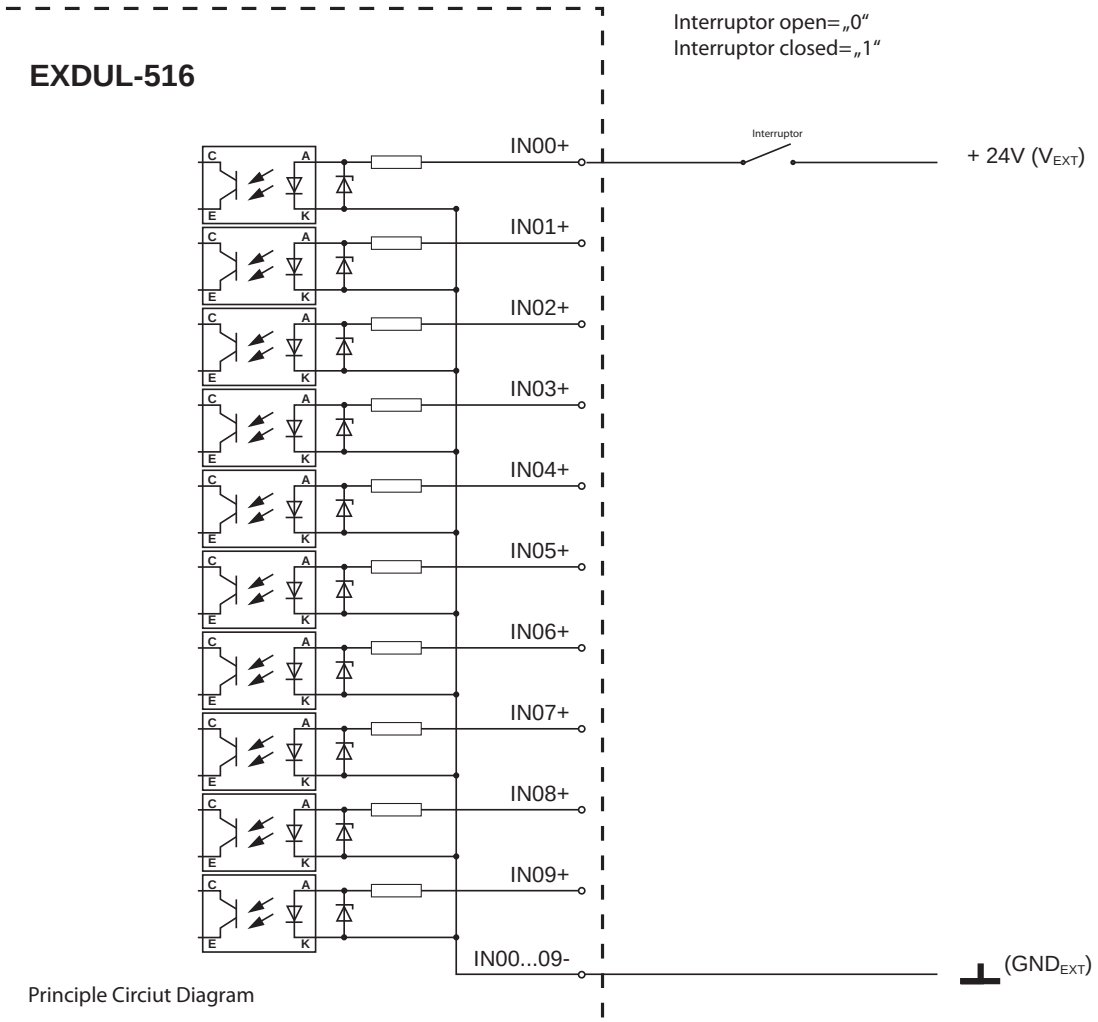
Module circuit points

1 * 24pin screw terminal strip
1 * RJ45 jack

Network cable	RJ45 network cable Cat5 or higher
Dimensions	105 mm x 89 mm x 59 mm (l x b x h)
Casing	Plastic casing with integrated snap-on technology or DIN EN rail mounting Suitable for control and engineering technology mounted to control and distribution boxes, surface mounting or mobile use on a desk.

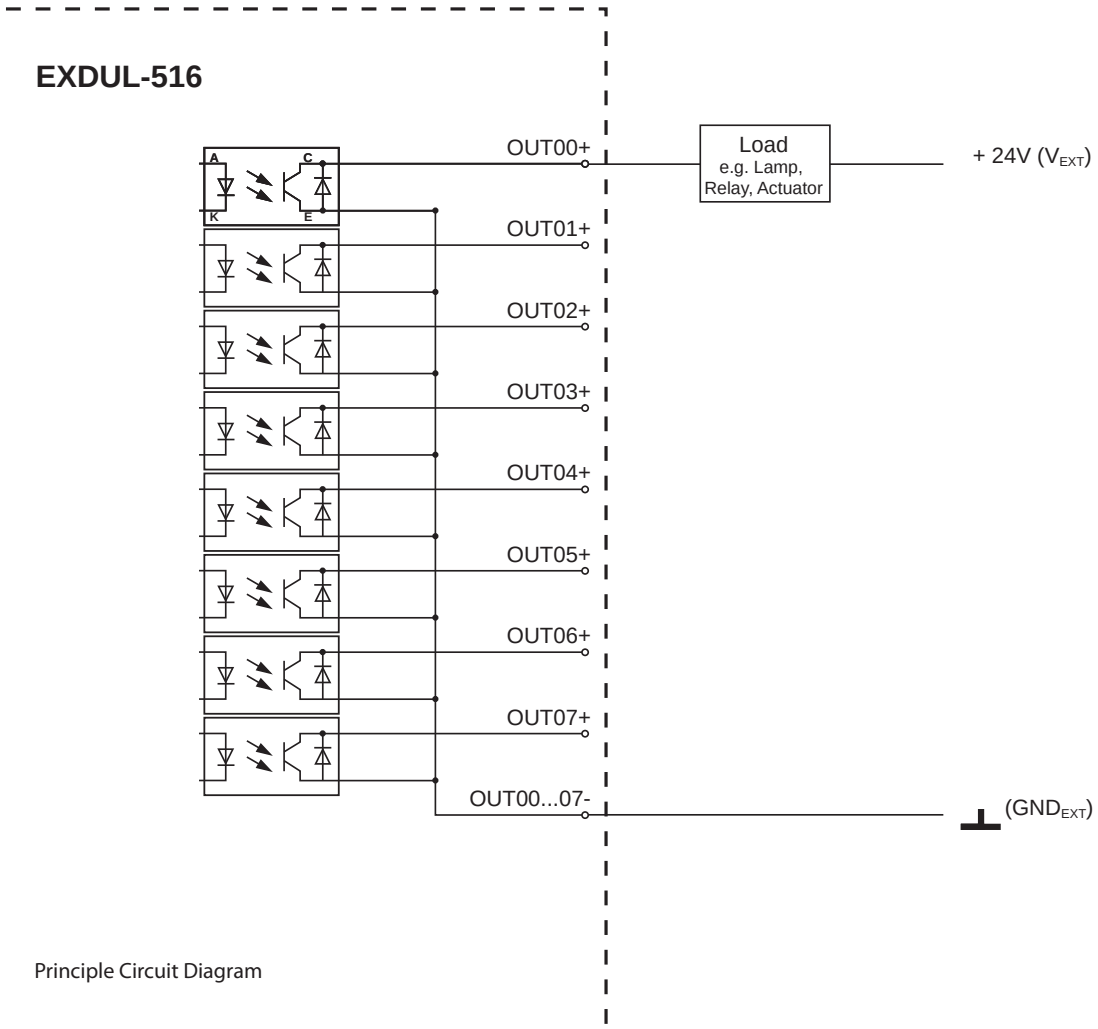
9. Examples of circiutry

9.1 Input Wiring



Principle Circiut Diagram

9.2 Output Wiring



Principle Circuit Diagram

10. ASCII Table

Hex	Dec	Binary	Character
00	0	00000000	
01	1	00000001	
02	2	00000010	
03	3	00000011	
04	4	00000100	
05	5	00000101	
06	6	00000110	
07	7	00000111	
08	8	00001000	
09	9	00001001	
0A	10	00001010	
0B	11	00001011	
0C	12	00001100	
0D	13	00001101	
0E	14	00001110	
0F	15	00001111	
10	16	00010000	
11	17	00010001	
12	18	00010010	
13	19	00010011	
14	20	00010100	
15	21	00010101	
16	22	00010110	
17	23	00010111	
18	24	00011000	
19	25	00011001	
1A	26	00011010	
1B	27	00011011	
1C	28	00011100	
1D	29	00011101	
1E	30	00011110	
1F	31	00011111	
20	32	00100000	[blank]
21	33	00100001	!
22	34	00100010	”
23	35	00100011	#
24	36	00100100	$
25	37	00100101	%
26	38	00100110	&
27	39	00100111	’

Hex	Dec	Binary	Character
28	40	00101000	(
29	41	00101001	)
2A	42	00101010	*
2B	43	00101011	+
2C	44	00101100	,
2D	45	00101101	-
2E	46	00101110	.
2F	47	00101111	/
30	48	00110000	0
31	49	00110001	1
32	50	00110010	2
33	51	00110011	3
34	52	00110100	4
35	53	00110101	5
36	54	00110110	6
37	55	00110111	7
38	56	00111000	8
39	57	00111001	9
3A	58	00111010	:
3B	59	00111011	;
3C	60	00111100	<
3D	61	00111101	=
3E	62	00111110	>
3F	63	00111111	?
40	64	01000000	@
41	65	01000001	A
42	66	01000010	B
43	67	01000011	C
44	68	01000100	D
45	69	01000101	E
46	70	01000110	F
47	71	01000111	G
48	72	01001000	H
49	73	01001001	I
4A	74	01001010	J
4B	75	01001011	K
4C	76	01001100	L
4D	77	01001101	M
4E	78	01001110	N
4F	79	01001111	O

Hex	Dec	Binary	Character
50	80	01010000	P
51	81	01010001	Q
52	82	01010010	R
53	83	01010011	S
54	84	01010100	T
55	85	01010101	U
56	86	01010110	V
57	87	01010111	W
58	88	01011000	X
59	89	01011001	Y
5A	90	01011010	Z
5B	91	01011011	[
5C	92	01011100	
5D	93	01011101	]
5E	94	01011110	^
5F	95	01011111	_
60	96	01100000	`
61	97	01100001	a
62	98	01100010	b
63	99	01100011	c
64	100	01100100	d
65	101	01100101	e
66	102	01100110	f
67	103	01100111	g
68	104	01101000	h
69	105	01101001	i
6A	106	01101010	j
6B	107	01101011	k
6C	108	01101100	l
6D	109	01101101	m
6E	110	01101110	n
6F	111	01101111	o
70	112	01110000	p
71	113	01110001	q
72	114	01110010	r
73	115	01110011	s
74	116	01110100	t
75	117	01110101	u
76	118	01110110	v
77	119	01110111	w
78	120	01111000	x
79	121	01111001	y
7A	122	01111010	z
7B	123	01111011	{

Hex	Dec	Binary	Character
7C	124	01111100	|
7D	125	01111101	}
7E	126	01111110	
7F	127	01111111	
80	128	10000000	
81	129	10000001	
82	130	10000010	
83	131	10000011	
84	132	10000100	
85	133	10000101	
86	134	10000110	
87	135	10000111	
88	136	10001000	
89	137	10001001	
8A	138	10001010	
8B	139	10001011	
8C	140	10001100	
8D	141	10001101	
8E	142	10001110	
8F	143	10001111	
90	144	10010000	
91	145	10010001	
92	146	10010010	
93	147	10010011	
94	148	10010100	
95	149	10010101	
96	150	10010110	
97	151	10010111	
98	152	10011000	
99	153	10011001	
9A	154	10011010	
9B	155	10011011	
9C	156	10011100	
9D	157	10011101	
9E	158	10011110	
9F	159	10011111	
A0	160	10100000	
A1	161	10100001	
A2	162	10100010	
A3	163	10100011	
A4	164	10100100	
A5	165	10100101	
A6	166	10100110	
A7	167	10100111	

Hex	Dec	Binary	Character
A8	168	10101000	
A9	169	10101001	
AA	170	10101010	
AB	171	10101011	
AC	172	10101100	
AD	173	10101101	
AE	174	10101110	
AF	175	10101111	
B0	176	10110000	
B1	177	10110001	
B2	178	10110010	
B3	179	10110011	
B4	180	10110100	
B5	181	10110101	
B6	182	10110110	
B7	183	10110111	
B8	184	10111000	
B9	185	10111001	
BA	186	10111010	
BB	187	10111011	
BC	188	10111100	
BD	189	10111101	
BE	190	10111110	
BF	191	10111111	
C0	192	11000000	
C1	193	11000001	
C2	194	11000010	
C3	195	11000011	
C4	196	11000100	
C5	197	11000101	
C6	198	11000110	
C7	199	11000111	
C8	200	11001000	
C9	201	11001001	
CA	202	11001010	
CB	203	11001011	
CC	204	11001100	
CD	205	11001101	
CE	206	11001110	
CF	207	11001111	
D0	208	11010000	
D1	209	11010001	
D2	210	11010010	
D3	211	11010011	

Hex	Dec	Binary	Character
D4	212	11010100	
D5	213	11010101	
D6	214	11010110	
D7	215	11010111	
D8	216	11011000	
D9	217	11011001	
DA	218	11011010	
DB	219	11011011	
DC	220	11011100	
DD	221	11011101	
DE	222	11011110	
DF	223	11011111	
E0	224	11100000	
E1	225	11100001	
E2	226	11100010	
E3	227	11100011	
E4	228	11100100	
E5	229	11100101	
E6	230	11100110	
E7	231	11100111	
E8	232	11101000	
E9	233	11101001	
EA	234	11101010	
EB	235	11101011	
EC	236	11101100	
ED	237	11101101	
EE	238	11101110	
EF	239	11101111	
F0	240	11110000	
F1	241	11110001	
F2	242	11110010	
F3	243	11110011	
F4	244	11110100	
F5	245	11110101	
F6	246	11110110	
F7	247	11110111	
F8	248	11111000	
F9	249	11111001	
FA	250	11111010	
FB	251	11111011	
FC	252	11111100	
FD	253	11111101	
FE	254	11111110	
FF	255	11111111	

11. Product Liability Act

Information for Product Liability

The Product Liability Act (Act on Liability for Defective Products - ProdHaftG) in Germany regulates the manufacturer‘s liability for damages caused by defective products.

The obligation to pay compensation can be given, if the product’s presentation could cause a misconception of safety to a non-commercial end-user and also if the end-user is expected not to observe the necessary safety instructions handling this product.

It must therefore always be shown, that the end-user was made familiar with the safety rules.

In the interest of safety, please always advise your non-commercial customer of the following safety instructions:

Safety instructions

The valid VDE-instructions must be observed, when handling products that come in contact with electrical voltage.

Especially the following instructions must be observed:
VDE100; VDE0550/0551; VDE0700; VDE0711; VDE0860.
The instructions are available from:
vde-Verlag GmbH
Bismarckstr. 33
10625 Berlin

* unplug the power cord before you open the unit or make sure, there is no current to/in the unit.

* You only may start up any components, boards or equipment, if they are installed inside a secure touch-protected casing before. During installation there must be no current to the equipment.

* Make sure that the device is disconnected from the power supply before using any tools on any components, boards or equipment. Any electric charges saved in components in the device are to be discharged prior.

* Voltaged cables or wires, which are connected with the unit, the components or the boards, must be tested for insulation defects or breaks. In case of any defect the device must be immediately taken out of operation until the defective cables are replaced.

* When using components or boards you must strictly comply with the characteristic data for electrical values shown in the corresponding description.

* As a non-commercial end-user, if it is not clear whether the electrical characteristic data given in the provided description are valid for a component you must consult a specialist.

Nevertheless, the compliance with building and safety instructions of every kind (VDE, TÜV, industrial injuries corporation, etc.) is duty of the user/customer.

12. CE Declaration of Conformity

This is to certify, that the products

EXDUL-516E EDV-Number A-374340
EXDUL-516S EDV-Number A-374320

comply with the requirements of the EC directives. This declaration will lose its validity, if the instructions given in this manual for the intended use of the products are not fully complied with.

EN 5502 Klasse B
IEC 801-2
IEC 801-3
IEC 801-4
EN 50082-1
EN 60555-2
EN 60555-3

The following manufacturer is responsible for this declaration:

Messcomp Datentechnik GmbH
Neudecker Str. 11
83512 Wasserburg

given by

Dipl.Ing.(FH) Hans Schnellhammer

Wasserburg, 14.12.2011

Reference system for intended use

The multi functional modules EXDUL-316E and EXDUL-316S are not stand-alone devices. The CE-conformity only can be assessed when using additional computer components simultaneously. Thus the CE conformity only can be confirmed when using the following reference system for the intended use of the multi functional modules:

Control Cabinet:	Vero IMRAK 3400	804-530061C 802-563424J 802-561589J
19“ Casing:	Vero PC-Casing	145-010108L
19“ Casing:	Addition Electronic	519-112111C
Motherboard:	GA-586HX	PIV 1.55
Floppy-Controller:	on Motherboard	
Floppy:	TEAC	FD-235HF
Grafic Card:	Advantech	PCA-6443
Interface:	EXDUL-516E EXDUL-516S	A-374340 A-374320