EXDUL-519PoE

EDP No: A-374650

EXDUL-519E

EDP No: A-374640

EXDUL-519S

EDP No: A-374620

11 inputs via bidirectional opto-isolators
8 outputs via optocoupler 1A each
6 counter 32bit
Programmable logic
TCP/IP and webserver
LCD display (EXDUL-519E only)
PoE and LCD display (EXDUL-519PoE only)

wasco®

manual

Registered Trademarks

Windows®, Visual Basic®, Visual C++®, Visual C#® are registered trademarks of Microsoft Corporation.
wasco® is a registered trademark.
EXDUL® is a registered trademark**.**
LabVIEW® is a registered trademark.
Other product and company names mentioned herein may be trademarks of their respective owners.

Limitation of Liability

The company Messcomp Datentechnik GmbH is not liable for direct or indirect damage resulting from the use of the multifunction module EXDUL-519 and this documentation.

Important Information:

This manual has been written for the EXDUL-519E, EXDUL-519PoE and EXDUL-519S modules. EXDUL-519E has an additional LCD display, EXDUL-519PoE an LCD display and power via PoE, all other functions of the modules are identical. The commands and functions related to the LCD display do not apply for the EXDUL-519S.

Table of Content

1. Product Details

The EXDUL-519E with Ethernet interface has 11 digital inputs and eight digital outputs with galvanic isolation. The bidirectional inputs are protected by additional overvoltage protection diodes. The outputs, which are protected by reverse polarity protection diodes, can switch a maximum current of 1A per channel due to the downstream FET power switches.

If required, six of the 11 optocoupler inputs can also be programmed to act as hardware-supported 32bit counter inputs. To prevent data loss in the event of a power failure, the counter readings are stored at 100µs intervals. They are automatically loaded into the counter registers when the module is restarted.

The integrated web page allows the module to be configured in a user-friendly manner. It is also possible to perform a simple functional test.

Communication between the PC and the module is achieved by sending and receiving byte arrays via a stable TCP/IP connection. The connection is secured by a handshake protocol.

The programmable logic of the module can be used both for autonomous actions at the outputs and for sending messages to the PC. This often eliminates the need to poll the inputs, and can significantly reduce both data traffic and computer load.

The module is supplied with the necessary operation voltage by means of an external power supply unit. The EXDUL-519PoE can also be powered via Power over Ethernet.

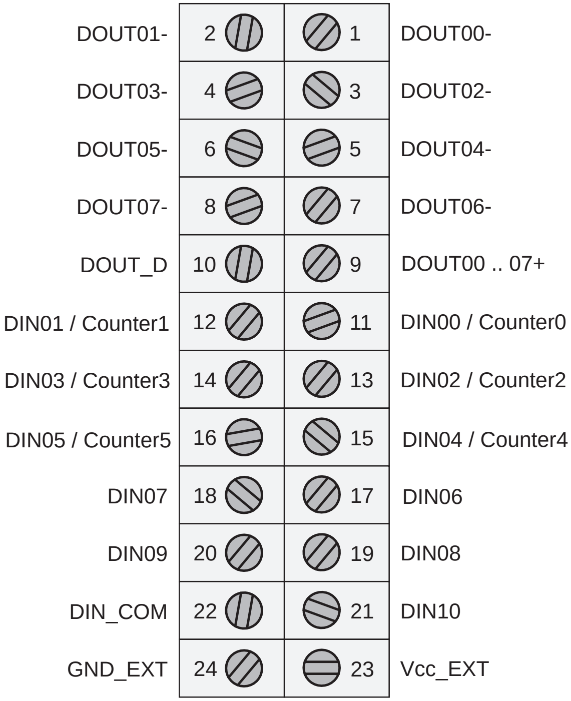
The programmable LCD display on the EXDUL-519E and EXDUL-519PoE alllows digital I/O status information or programmable user specific data to be displayed.

The power supply connections are made on a 24-pin screw terminal strip, as are the input and output optocoupler connections.

The compact package allows use as a mobile module on a notebook computer. It can also be easily mounted on a wall or DIN rail for use in mechanical or control engineering applications.

2. Connection Terminals

2.1 Pin Assignments of CN1



DOUT01-	2	1	DOUT00-
DOUT03-	4	3	DOUT02-
DOUT05-	6	5	DOUT04-
DOUT07-	8	7	DOUT06-
DOUT_D	10	9	DOUT00 .. 07+
DIN01 / Counter1	12	11	DIN00 / Counter0
DIN03 / Counter3	14	13	DIN02 / Counter2
DIN05 / Counter5	16	15	DIN04 / Counter4
DIN07	18	17	DIN06
DIN09	20	19	DIN08
DIN_COM	22	21	DIN10
GND_EXT	24	23	Vcc_EXT

Vcc_EXT:
Connecting terminal for external power supply
GND_EXT:
Ground connection terminal

3. System Components

3.1 Block Diagram EXDUL-519PoE

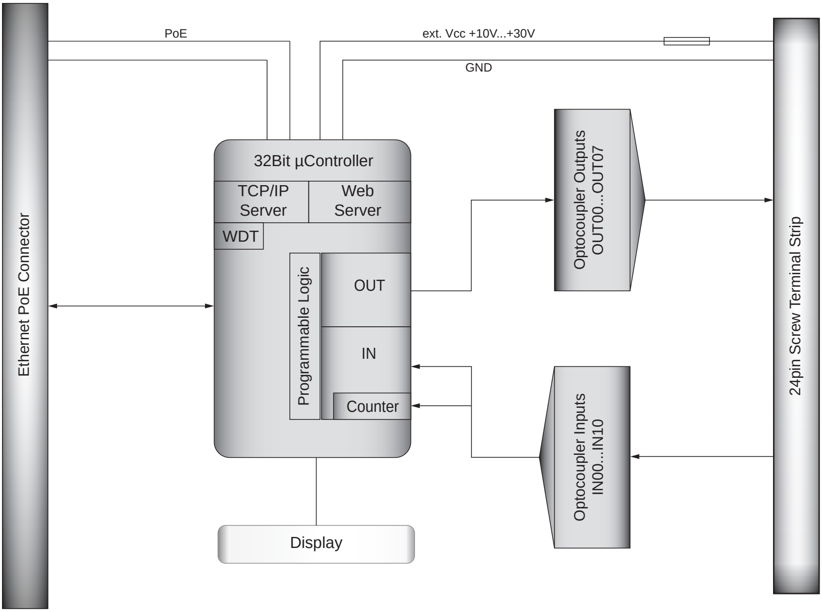


Fig. 3.1 Block diagram EXDUL-519PoE

3.2 Block Diagram EXDUL-519E

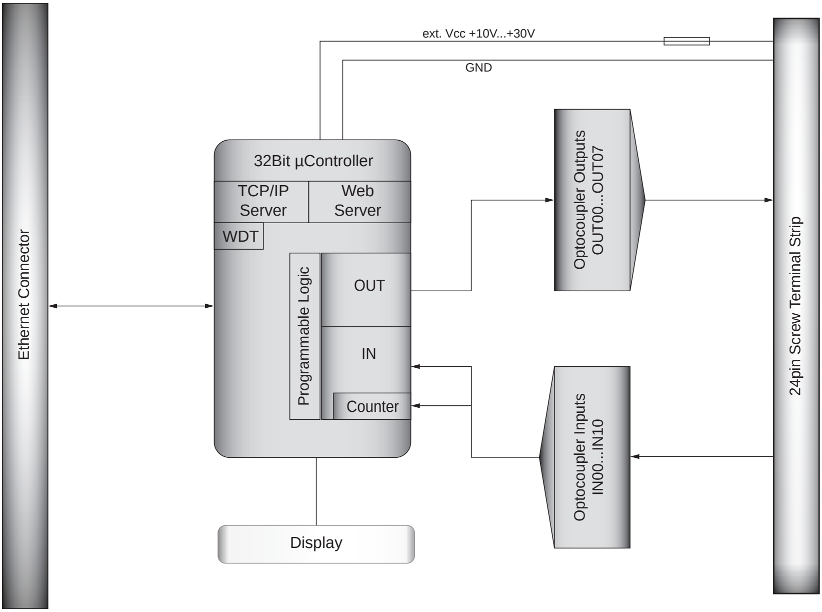


Fig. 3.2 Block diagram EXDUL-519E

3.3 Block Diagram EXDUL-519S

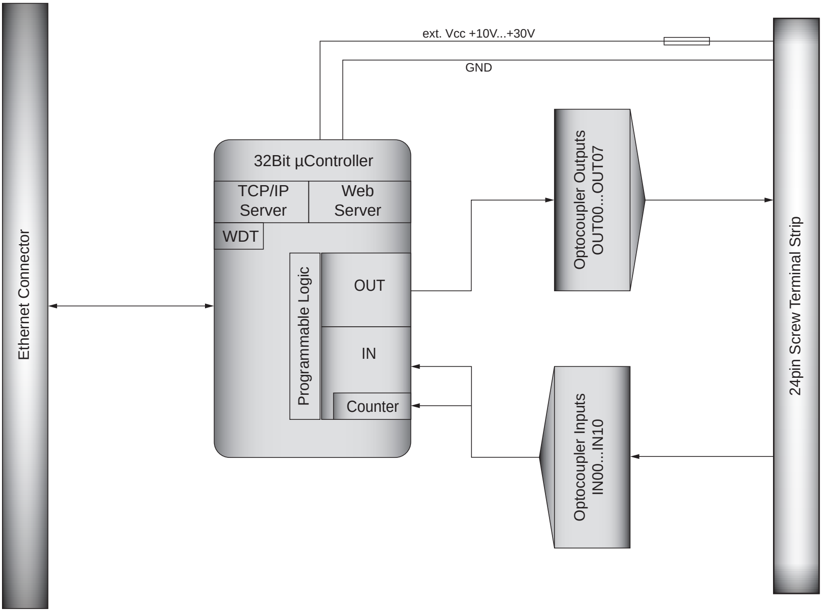


Fig. 3.3 Block diagram EXDUL-519S

3.4 Digital Inputs via Optocoupler

11 bidirectional channels with galvanic isolation
Over voltage protection diodes
Input voltage range
high = 10..30 Volt
low = 0..3 Volt

3.5 Digital Outputs with FET Power Switches

8 channels with galvanic isolation via optocouplers
Contact A of the FET (source, + connection) with one connection terminal each, contact B (drain, - connection) with common connection terminal for all 8 channels
Reverse polarity protection
Switchable free-wheeling diodes for each of the channels
Voltag CE: max. 30V
Output current: max. 1A/channel
Switching time: typ. 60µs (24V, 100mA)
Release time: typ. 250µs (24V, 100mA)

3.6 Counter

6 programmable 32bit counters (using 6 optocoupler inputs)
Counting frequency: max. 5kHz
Automatic counter readings backup every 10kHz

3.7 Programmable Logic

Switch outputs with logic operations
Send messages to PC when input status changes

3.8 TCP/IP Server

Stable and easy communication with a simple protocol

3.9 Web Server

Configuration
Functional test
Firmware Update

3.10 Timer for Communication Watchdog

Time window adjustable from 1 to 2^{32} - 1ms

3.11 LCD Display (EXDUL-519E and EXDUL-519PoE only)

2-line,16-column matrix display with 16 characters per line
Programmable to display application specific data or as I/O status display

3.12 Power over Ethenet (EXDUL-519PoE only)

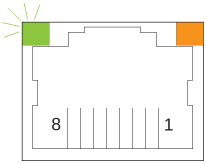
The module supports the IEEE802.3at standard when powered via PoE. The supplied voltage must be between 44V and 57V. IEEE802.3 standard specifies a typical voltage of 48V.

4. Getting started

Connection to a network or to a computer is simple and convenient via an Ethernet interface, and can be configurated from any web browser. An external power supply unit is required to provide the necessary operating voltage..

4.1 Connection to an Ethernet Port

The EXDUL-519 module features a 10/100 Fast Ethernet interface with an RJ-45 connector. It is connected directly to either a PC, Ethernet hub or Ethernet switch via a network cable.
The module boots when power is applied; as soon as a stable connection has been established, the left LED of the RJ-45 socket of the EXDUL-519 is lit continuously in green.



4.2 Power Supply Connection

The EXDUL-519E / EXDUL-519S requires a supply voltage from +10V ... +30V DC at terminal 23 (Vcc) and terminal 24 (GND) to operate.
The EXDUL-519PoE can be powered both via the terminals 23 (Vcc) und 24 (GND) as well as via the Ethernet cable using PoE. The power supply via PoE is only used if no power is available at the terminals. The Switchover is automated.

4.3 Integrated Web page of the EXDUL-519

Any web or internet browser such as Mozilla Firefox, Chrome, Edge, Safari etc. can access the EXDUL-519’s web page via a TCP/IP connection. The web page allows to read connection information and change password protected configuration data. Settings made are stored in the EXDUL-519’s built-in memory and are loaded during the boot process. The EXDUL-519 web page also allows you to write, read and display the user memory areas UserA, UserB, UserLCD1 and UserLCD2 as well as to start and stop the counters. It is also possible to test the digital inputs and outputs.

4.4 Password Protection - Access Code

As mentioned above, the EXDUL web page can be used to configure the network, set the LCD display, write to the user memory areas as well as set the inputs and outputs and the counters. In order to prevent unauthorised access, these setting areas are password protected.

The factory default access code is as follows:

User ID:	admin
Password:	11111111

Note that the password is case sensitive!

If you cannot log in with this password, your system administrator has changed the password settings.

4.5 Default Network Configuration

In the factory default setting, the EXDUL-519 is set to DHCP (Dynamic Host Configuration Protocol) to obtain a dynamic IP address. During power-up, the EXDUL-519 requests an IP address from the network (LAN). A network with an enabled DHCP server automatically assigns an IP address to the module. This setting allows an easy and comfortable connection of the module and a correct adaptation of the configuration data according to the own requirements.

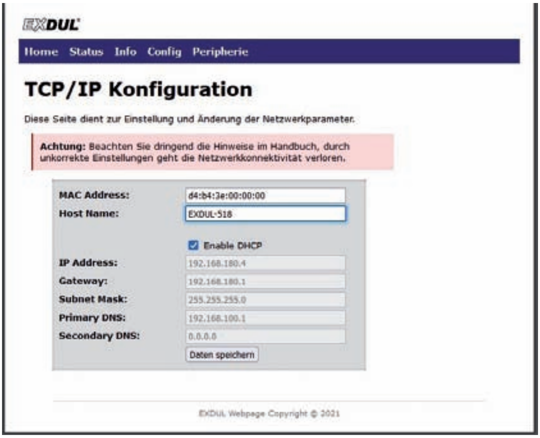
4.6 Composition and Structure of an IP Address

IP4 addresses consist of 32 bits = 4 bytes (octets). Each byte can range from 0 to 255. IP addresses are represented in dot-decimal notation (e.g. 192.168.1.83).
Each IP address contains a network and a device part (the host identifier). A subnet mask divides the network and host parts. All devices on the same network can communicate with each other.

For Example:
If the subnet mask 255.255.0.0 is assigned to the IP address 192.168.1.83, then the device is on the192.168.-.- network as device -.-.1.83.

4.7 Changing the Network Configuration

To change the factory default configuration settings, connect the EXDUL-519 to a local network using the RJ-45 socket and the standard network cable supplied. A power supply (+10V ...+30V) is also required at terminal 23 (Vcc) and terminal 24 (GND) of the module. The EXDUL will now boot. As soon as a stable connection has been established, the left LED of the RJ-45 socket of the EXDUL-519 is lit continuously in green.
The EXDUl-519 web page can be accessed via any web or internet browser by entering the host name **http://EXDUL-519.** Click on the **TCP/IP Config** button to open following configuration screen:



TCP/IP Konfiguration

Diese Seite dient zur Einstellung und Änderung der Netzwerkparameter.

Achtung: Beachten Sie dringend die Hinweise im Handbuch, durch unkorrekte Einstellungen geht die Netzwerkkonnektivität verloren.

MAC Address: d4:b4:3e:00:00:00
Host Name: EXDUL-518

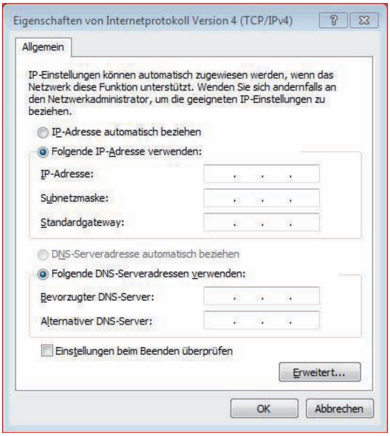
☑ Enable DHCP

IP Address: 192.168.180.4
Gateway: 192.168.180.1
Subnet Mask: 255.255.255.0
Primary DNS: 192.168.100.1
Secondary DNS: 0.0.0.0

EXDUL Webpage Copyright © 2021

You can use the standard network cable supplied to connect EXDUL-519 directly to a PC with an Ethernet interface. With a direct connection, there is normally no DHCP service available, as neither the PC nor the EXDUL provides one. In this case, the last static IP address set can be used to address the EXDUL-519. New EXDUL-519 units are delivered with an IP address of 169.254.1.1.

Important note:
The PC used and the EXDUL connected must be located on the same network to be able to communicate. For this, the computer‘s DHCP must be disabled and, for example, the first two number blocks (192.168) of the EXDUL IP address 192.168.1.83 (Subnet mask 255.255.0.0) must be transferred to the computer‘s IP address. The next two number blocks can be assigned values between 0 and 255 according to your current personal requirements.



Eigenschaften von Internetprotokoll Version 4 (TCP/IPv4)

Allgemein

IP-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IP-Einstellungen zu beziehen.

○ IP-Adresse automatisch beziehen
● Folgende IP-Adresse verwenden:

IP-Adresse: . . .
Subnetzmaske: . . .
Standardgateway: . . .

○ DNS-Serveradresse automatisch beziehen
● Folgende DNS-Serveradressen verwenden:

Bevorzugter DNS-Server: . . .
Alternativer DNS-Server: . . .

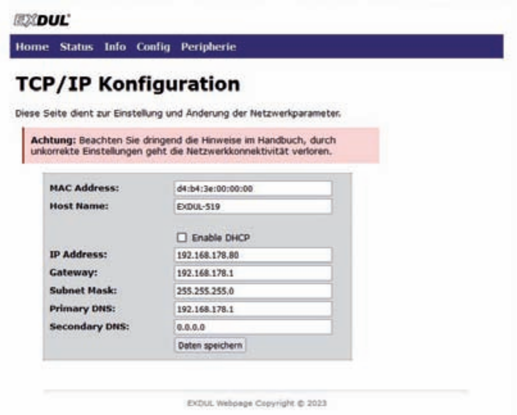
☐ Einstellungen beim Beenden überprüfen

The network part of the IP address depends on the subnet mask. All octets of the IP address of the Exdul-519 which are assigned to 255 in the subnet mask must be transferred to the computer‘s IP address (see chapter 4.6 Composition and Structure of the IP Address)

4.8 Configuration with Static IP Address (DHCP disabled)

To configure EXDUL-519 with a static IP address DHCP must be disabled in the EXDUL-519 configuration mask. To change the configuration, connect EXDUL-519 to a LAN or PC as described in chapter 4.7 (Changing the network configuration).

As soon as a stable Ethernet connection has been established, you can access the EXDUL-519 web page using any Internet browser. By entering the host name **http://EXDUL-519**, the browser should open the EXDUL-519 web page. Click on the **TCP/IP Config** button to open the following configuration screen:



TCP/IP Konfiguration

Diese Seite dient zur Einstellung und Änderung der Netzwerkparameter.

Achtung: Beachten Sie dringend die Hinweise im Handbuch, durch unkorrekte Einstellungen geht die Netzwerkkonnektivität verloren.

MAC Address:	d4:b4:3e:00:00:00
Host Name:	EXDUL-519
	□ Enable DHCP
IP Address:	192.168.178.80
Gateway:	192.168.178.1
Subnet Mask:	255.255.255.0
Primary DNS:	192.168.178.1
Secondary DNS:	0.0.0.0

EXDUL Webpage Copyright © 2023

Once you have disabled DHCP, you can enter your desired IP address, subnet mask and host name. Click on the **Konfiguration speichern** (*save configuration*) button to save all currently registered data in the internal memory of the EXDUL-519. From this point on, the module can only be accessed by the IP address registered here. The PC or LAN used must be located on the same network.

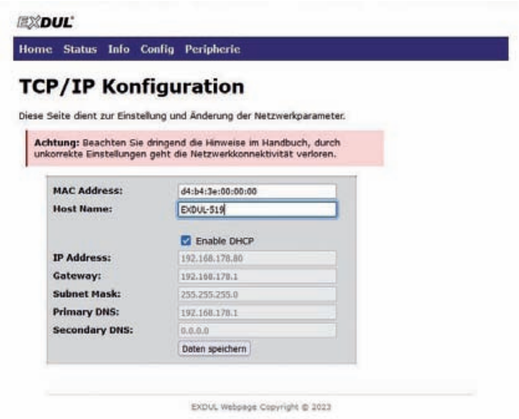
Important note: Each IP address and each host name can only be assigned to one device or module on a network; duplicate assignment is not permitted! Any host name can be chosen, but it must consist of the ASCII characters 0 to 9 and A to Z (not case-sensitive) and the - (hyphen).
Certain IP addresses are reserved or have a special function, e.g. 127.0.0.1 (local host), 192.168.1.0 (0 is the network address) with 255.255.255.0 (subnet mask).

Please ask your network administrator which IP address you are allowed to use. Using an unauthorised IP address may prevent access to the module. The main unauthorised addresses are blocked by the module.

4.9 Configuration with Dynamic IP Address (DHCP enabled)

If you want to integrate EXDUL-519 into an existing network with an already active DHCP server and access it via a dynamic IP address, DHCP (Dynamic Host Configuration Protocol) must be enabled in the configuration screen of the web page. By default, DHCP is already enabled and it is not necessary to change this setting.

If a static IP address is set, enable DHCP as follows:
Connect the EXDUL-519 to a computer using a network cable (older computers may require a crossover cable). Make sure that the computer is set to: **Folgende IP-Adresse verwenden:** (DHCP-deaktiviert) (*Use the following IP address: DHCP disabled*)
The network part of the IP address (see chapter 4.6 Composition and structure of IP Address) must be overwritten by the EXDUL-519 IP address, because the computer and the EXDUL must belong to the same network.
Connect supply voltage (+10V ...+30V) to terminals 23 (Vcc) and 24 (GND) of the module.The EXDUL will start up immediately. As soon as a stable connection has been established, the left LED of the RJ-45 socket of the EXDUL-519 is lit continuously in green.
Access to the EXDUl-519 web page is possible with any web browser pointing to the EXDUL-519‘s host name or IP address**.** The browser should now open the EXDUL web page. If you click on the **TCP/IP Config** button,the configuration screen will open and you can **Enable DHCP** as shown in the following figure:



TCP/IP Konfiguration

Diese Seite dient zur Einstellung und Änderung der Netzwerkparameter.

Achtung: Beachten Sie dringend die Hinweise im Handbuch, durch unkorrekte Einstellungen geht die Netzwerkkonnektivität verloren.

MAC Address:	d4:b4:3e:00:00:00
Host Name:	EXDUL-519
	☑ Enable DHCP
IP Address:	192.168.178.80
Gateway:	192.168.178.1
Subnet Mask:	255.255.255.0
Primary DNS:	192.168.178.1
Secondary DNS:	0.0.0.0

EXDUL Webpage Copyright © 2023

Important note:
We strongly recommend that you consult your network administrator before enabling DHCP. If you want to integrate several identical EXDUL-5xx series modules into a network, the preset 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 chosen, but it must consist of ASCI characters 0 to 9 and A to Z (case insensitive) and the - (hyphen).

4.10 LCD Display during Booting (EXDUL-519E / 519PoE only)

While the module is booting, the display shows information data. Line 1 shows the name of the module, line 2 the information that the module has been started. As soon as the boot process is completed, either the I/O status display or the UserLCD display appears, depending on the configuration.

4.11 LCD Display during Operation (EXDUL-519E / 519PoE only)

After booting, the display switches from information display to I/O status display or the UserLCD display, depending on the setting. When the I/O status is displayed, line 1 shows the current status of the inputs and line 2 shows the status of the outputs. If the UserLCD mode is activated in the EXDUL-519 web page, the display shows UserLCD data with values from the UserLCD1m and UserLCD2m memory areas instead of I/O status display.

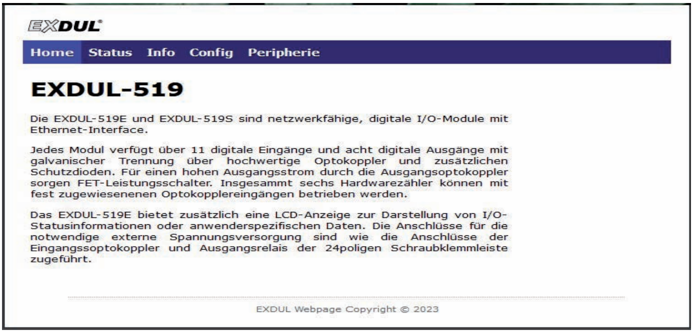
The LCD display will show UserLCD1m and UserLCD2m data values unless you write new user data to UserLCD line1 and line2 of the LCD display. To prevent „screen burn“, the display switches from I/O status display or UserLCD display to the information display for approx. 5 seconds every minute during operation, showing the current IP address.

5. Accessing the EXDUL-519

As already mentioned, access to the configuration settings and to the inputs and outputs of the EXDUL-519 is possible via the EXDUL-519 web page or via the TCP/IP sockets. For this, the IP address, the host name or MAC address is required.

5.1 Access via the EXDUL Web Page

The EXDUL-519 web page allows you to read inputs, set outputs, read UserA, UserB and UserLCD user memory areas, read information about connection or module details or change configuration data. You can access to the web page from any computer connected to the module using any Internet browser. The computer used must be set to „**IP-Adresse automatisch beziehen (DHCP-aktiviert)“** (*Obtain an IP address automatically, where DHCP is turned on)* if the EXDUL-519 is still in the delivery state (DHCP enabled) and if it is integrated in a network with an active DHCP service. You can open the web page by entering the host name (in the delivery state **http://EXDUL-519,** otherwise the name you have set, or you can find it out using the ExdulUtility_v2_xx or a later version) or the IP address.If you can‘t open the EXDUL-519 web page, check the network connections or the host name you have entered. For more details please refer to the FAQ - Troubleshooting chapter.



EXDUL-519

Die EXDUL-519E und EXDUL-519S sind netzwerkfähige, digitale I/O-Module mit Ethernet-Interface.

Jedes Modul verfügt über 11 digitale Eingänge und acht digitale Ausgänge mit galvanischer Trennung über hochwertige Optokoppler und zusätzlichen Schutzdioden. Für einen hohen Ausgangsstrom durch die Ausgangsoptokoppler sorgen FET-Leistungsschalter. Insgesammt sechs Hardwarezähler können mit fest zugewiesenenen Optokopplereingängen betrieben werden.

Das EXDUL-519E bietet zusätzlich eine LCD-Anzeige zur Darstellung von I/O-Statusinformationen oder anwenderspezifischen Daten. Die Anschlüsse für die notwendige externe Spannungsversorgung sind wie die Anschlüsse der Eingangssoptokoppler und Ausgangsrelais der 24poligen Schraubklemmleiste zugeführt.

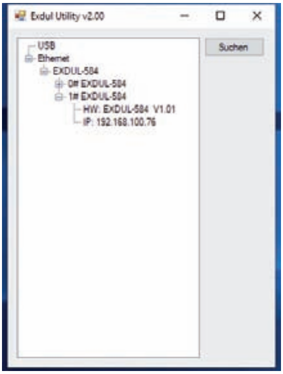
EXDUL Webpage Copyright © 2023

5.2 Access via TCP/IP Sockets

A reliable connection between the PC and the EXDUL-519 is achieved by using the TCP protocol. The protocol automatically takes action in case of data loss. The module is addressed by a 4-byte IP address (IPv4) or the assigned host name and a port number 9760. The computer sends one byte array for each command over the link. The module processes the command and always sends a response. In combination with a high level language, the TCP/IP connection allows to read the inputs, to set the outputs, start, stop and read the counters, write to the user memory areas, read the connection and module details and also modify the configuration setup. Up to three TCP/IP connections can be opened at the same time with the module.

5.3 How to Determine the Host Name, IP Address and MAC Address

If you do not know either the host name or the IP address or the MAC address of the EXDUL-519, you can use the ExdulUtility_v2_xx (or a later version) discovery tool to find them out. If your firewall prevents the discovery tool from communicating with the EXDUL-5xx, ExdulUtility_v2_xx needs to be enabled in the firewall.



```
USB
Ethernet
  EXDUL-584
    0# EXDUL-584
    1# EXDUL-584
      HW: EXDUL-584  V1.01
      IP: 192.168.100.76
```

The **ExdulUtility_v2_xx** (exdul_sp_v04) (or a later version) discovery tool is available for download at www.wasco.de.

6. 11 Optocoupler Inputs

The EXDUL-519 provides 11 input channels, each of which is optically isolated by optocouplers. The isolation voltage between the ground of the computer and the input is 500 volts. The voltage within the input channels is limited to 50 volts.

6.1 Pin Assignment of the Optocouplers on the Inputs

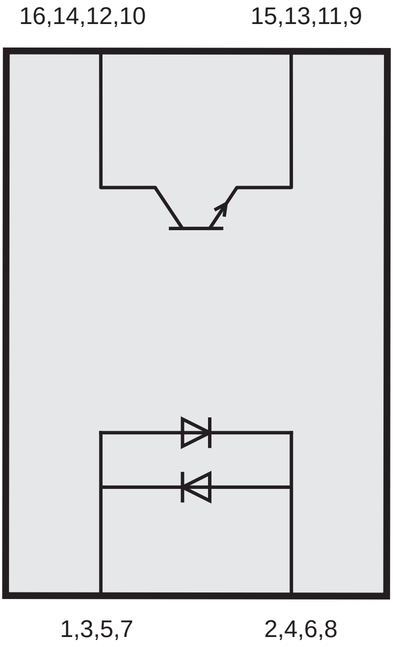


Fig. 6.1 Pinning of the input optocouplers

6.2 Input Circuitry

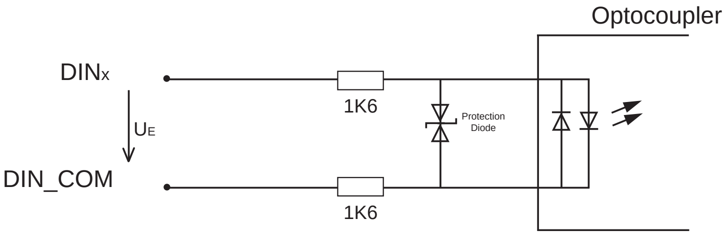


Fig. 6.2 Input circuit

The optocoupler inputs are bidirectional. Normally the DIN_COM connector is connected to minus and a voltage is applied to the DINx connector. However, it is also possible to apply the positive voltage to the DIN_COM connector and the negative voltage to the DINx connector if this makes more sense in terms of circuitry. Note that this setting applies to all inputs, as the DIN_COM connector is common to all input optocouplers.

6.3 Input Current

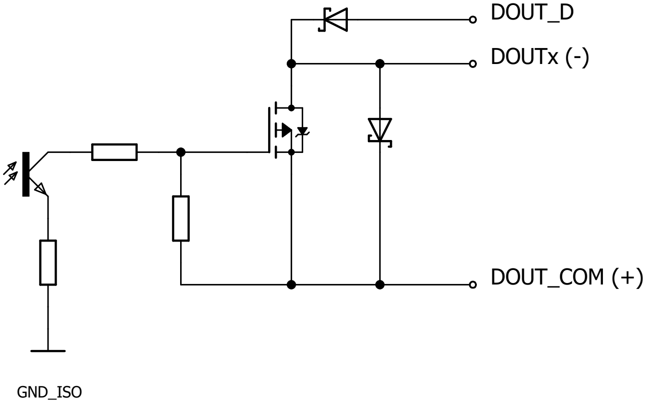
$$I_E \approx \frac{U_E - 1{,}1V}{3200\Omega}$$

An input voltage between DINx and DIN_COM of 24 volts results in an input current of approx. 7mA, at 12V approx. 3,4mA.

7. 8 Optocoupler Outputs

The EXDUL module provides 8 output channels, which are also optically isolated. The optocoupler output is amplified by a high power MOSFET, providing an output current of 1A per channel. A freewheeling diode is available for each channel for use in special situations.

7.1 Optocoupler/FET Output Circuit

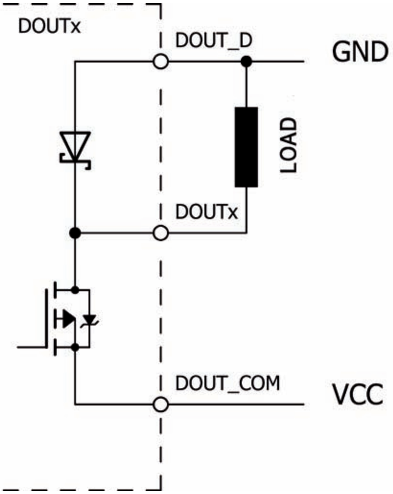


7.2 Output Data

Voltage DS:	max. 30V
Output current :	max.1A/channel
Switching times:	typ.60µs (24V, 100mA)
Release times:	typ. 250µs (24V, 100mA)

7.3 Freewheeling Diode

If the optocoupler outputs are used to switch loads with low inductance and it is not possible to connect a freewhelling diode directly to the inductance, a freewheeling diode is provided in the module for each channel. The wiring should be as shown in the diagram below.
If it is possible to connect a suitable freewheeling diode in parallel directly next to the load, then this should be the preferred option. If a higher inductance (contactor) is also controlled, an external freewheeling diode is essential to prevent damage caused by the voltage spikes that occur when the inductance is switched off. This freewheeling diode should be placed as close as possible to the inductive load (e.g. contactor).

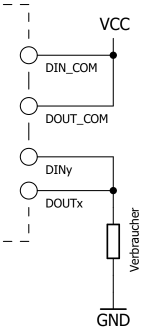


7.4 Programming of the Optocoupler Outputs

For programming the optocoupler outputs, several access functions are available. This allows all channels to be controlled simultaneously or individually. If the current output state cannot be stored in the user application, it can be retrieved using a read command. Please note that the read switching state only corresponds to the primary state (on the processor side). In order to read back the actual switching state or the level of the output, you have to use an optocoupler input (see chapter Reading Optocoupler Outputs).
A detailed description of programming is given in the chapter Programming.

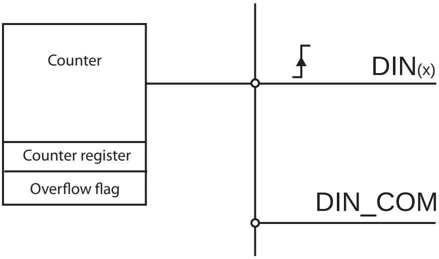
7.5 Reading Optocoupler Outputs

In some applications, it is necessary to know the actual output state in the application. An example of this could be error detection in the program. This can be done by feeding the optocoupler output back to one of the optocoupler inputs. In the following wiring example, output x is connected to input y. Please note that the input result is negated in this circuit version. When the output is connected, there is no voltage across the transitor and therefore the input indicates „0“. Additionally, when the output is not switched, there is a small current flowing through the optocoupler input and therefore also through the load (approx. 7mA at 24V).



8. Counters

The module provides a self-contained, hardware-supported 32bit counter on each of the first 6 optocoupler inputs (DIN0 to DIN5). These can be activated individually, if required. They respond to each rising egde by incrementing the counter value. Any overflow is indicated by a retrievable flag.
The counter readings are backed up approximately every 100µs to prevent them from being lost in the event of an unintentional power interruption. The stored values are automatically reloaded into the counter registers when the power supply returns.



Functions	Notes
Start	Starts the counter or enables the input
Stop	Stops the counter, signals at the corresponding input are ignored
Reset	Resets the counter reading to 0
Read counter reading	Reads the current counter reading
Read overflow flag	Reads the overflow flag
Clear overflow flag	Clears the Overflow flag

9. Programmable Logic

In some applications it may be useful for the module to react autonomously to input signals or input changes. Without this option, the only way to detect a change is by regularly polling the inputs. This type of polling places a increased load on the network and on the computer.
To generate this autonomous reaction of the module, four logic branches are provided, each with four logic inputs, one gate and one logic output. One of these branches is shown in figure 9.1.
You can assign individual functions to the logic inputs and outputs, as explained later.

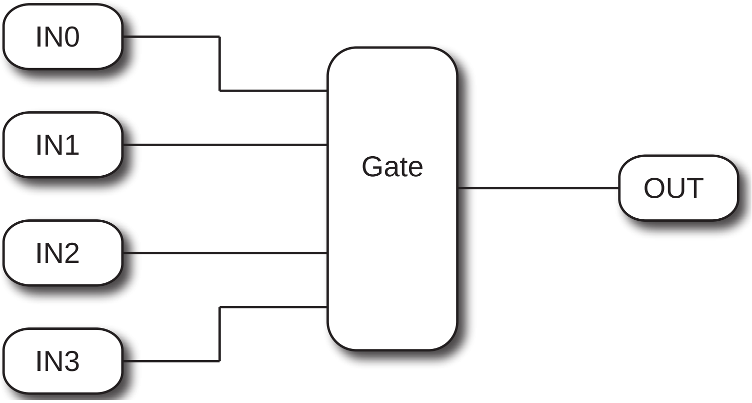


Fig. 9.1

9.1 Logic Branch

Four different logic branches can be used in the module. Each branch has four inputs, one logic operation (AND, OR) and one output. Logic inputs and outputs can be assigned different functions. The logic gate can be event-controlled (e.g. an edge at the inputs inverts the optocoupler output or sends a message to the PC). It can also be state-controlled (e.g. if inputs DIN0 and DIN1 are HIGH, then DOUT0 is also HIGH)
When selecting the functions, make sure that there is always an event input in the logic branch for an event output (e.g. send message to the PC or set optocoupler ouput DOUT1).
If the NONE function is assigned to a logic output, this branch is automatically disabled, thus saving processor resources.

9.2 Logic Inputs

Each logic branch provides four logic inputs. These can be assigned following functions:

Input function	Notes	State/event
TRUE	input always returns 1	state
FALSE	input always returns 0	state
DINxx	logic input is connected to the corresponding optocoupler input	state
DINxx_EDGE	logic input detects a rising edge and returns a 1 for a branch cycle (~10ms)	event

As a starting point for the logic inputs, the optocoupler inputs are scanned at regular intervals (approx. 1ms). This allows logic levels or level changes to be captured and transferred to the programmable logic.

9.3 Logic Operation

Each logic branch has a logic operation. The following functions can be assigned:

Operation	Notes	State/event
AND	AND link	state/event
OR	OR link	state/event

9.4 Logic Output

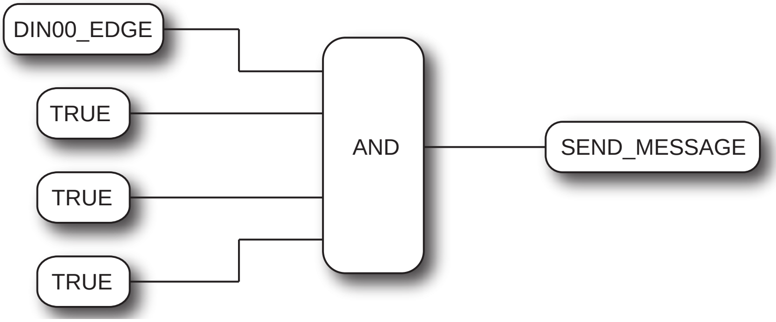
Each logic branch has one logic output. The following functions can be assigned:

Logic output	Notes	state/event
NONE	Logic branch is disabled	state
SEND_MESSAGEx	Module sends message(s) x to the PC	event
WRITE_DOUTxx	Optocoupler output is written with the result of the logic operation	state
‚SET_DOUTxx	The corresponding optocoupler output is switched if the logic operation is 1	event
CLEAR_DOUTxx	The corresponding optocoupler output is disabled if the logic operation is 1	event
TOGGLE_DOUTxx	The corresponding optocoupler output is inverted if the logic operation is 1	event

9.5 Sending Message to PC

If you want to send a message to the PC or application when an event occurs, you can do this via the message outputs. There are a total of 4 message function outputs, each of which sends an individual message to the PC.
This function requires another TCP/IP connection to be established in the program with the module in receiver mode. If, for example, a rising egde is detected on one of the monitored inputs of the programmable logic, a message is automatically sent to the receiver. This message does not require a reply.

The following example illustrates the configuration:



In this example, a message is to be sent to the PC on each rising edge of DIN00. For this, the function DIN00_EDGE must be selected for the input function block IN0 and TRUE for all others. Select an AND operation as the gate and SEND_MESSAGE1 as the output. If there is a rising edge at DIN00, the receiver connection will send a 12byte array to the TCP/IP client.

Four message arrays are provided to allow the PC to distinguish which logic block is sending the message. The following table shows the 4 possible byte arrays. The first 4 bytes represent the command, in this case always 0x0E, and the length byte (always 2). Bytes 4 to byte 6 are reserved and have the value 0x00. Byte 7 is the message index. If the message 1 has been selected as output, this byte shows the value 1. The byte array has an additional counter in order to check the sequence of the messages sent or to detect, for example, a missing message. Bytes 8 to byte 11 correspond to the value of the receiver message counter. Each message to the PC increments this value.

Receiver message counter:
Value = dd*0x100000 + cc*0x10000 + bb*0x100 + aa

Byte index	0	1	2	3	4	5	6	7	8	9	10	11
Message1	0x0E	0x00	0x00	0x02	0x00	0x00	0x00	0x01	aa	bb	cc	dd
Message2	0x0E	0x00	0x00	0x02	0x00	0x00	0x00	0x02	aa	bb	cc	dd
Message3	0x0E	0x00	0x00	0x02	0x00	0x00	0x00	0x03	aa	bb	cc	dd
Message4	0x0E	0x00	0x00	0x02	0x00	0x00	0x00	0x04	aa	bb	cc	dd

9.6 Programmable Logic Timing

	typ. processing frequency
Logic branch update	100 Hz
Input polling	1 kHz

10. Watchdog-Timer (WDT)

The EXDUL module features a number of safety mechanisms to ensure a stable communication. However, if the communication is interrupted, e.g. due to routing problems, and it is not possible to re-establish the connection, you can use the module‘s integrated watchdog timer to solve the problem.

When starting the application, the WDT has to be initialised and started with a defined timeout period. If the timer is not reset by a PC command within the preset timeout period, the EXDUL module is competely reset. When the watchdog reset has been triggered, the corresponding flag is set in the error register. Checking this bit later, you can detect the reset.

11. Error Register

If any unplanned errors occur (e.g. a watchdog reset), they will be indicated in the two error registers. When an error occurs, the bit associated with the error is set. The error registers are maintained even after a reset or power off. The registers can be reset with the appropriate command.

Error register 0	Bit31 .. Bit2	Bit1	Bit0
Meaning	reserved	WDT_SW	reserved

Error register 1	Bit31 .. Bit0
Meaning	reserved

Description:
WDT_SW: 1 = the watchdog reset has been executed

12. Information LCD and User Register

12.1 Register HW Identifier and Serial Number

Byte	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
HW Identifier	E	X	D	U	L	-	5	1	9			V	1	.	0	1
	45$_{hex}$	58$_{hex}$	44$_{hex}$	55$_{hex}$	4C$_{hex}$	2D$_{hex}$	35$_{hex}$	31$_{hex}$	39$_{hex}$	20$_{hex}$	20$_{hex}$	56$_{hex}$	31$_{hex}$	3E$_{hex}$	30$_{hex}$	31$_{hex}$
S/N	1	0	4	4	0	2	6									
	31$_{hex}$	30$_{hex}$	34$_{hex}$	34$_{hex}$	30$_{hex}$	32$_{hex}$	36$_{hex}$									

Table 12.1 Register HW Identifier and serial number

The module name and the firmware version are stored in the HW Identifier register and can be read by the user to determine the product identity. The table above is an example for the module EXDUL-519 with firmware version 1.01. The HW Identifier line shows each hex value and the corresponding ASCII character.

The register Serial Number is a read-only register. The serial number in the table above is an example of the format. The S/N line shows each hex value and the corresponding ASCII character as for the serial number 1044026.

12.2 UserA, UserB, UserLCD1m* und UserLCD2m* Memory Areas

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}

Each of the UserA and UserB registers contains 16 digits (16 bytes) and can be used for your own purposes. The data is retained when you switch off, and the registers can be reset to the factory default settings. On delivery, each digit in all user memory areas is set to the hex value 20, which corresponds to a space in ASCII code.
The table above shows each hex value and the corresponding ASCII character above it.

12.3 UserLCD-line1*, UserLCD-line2* and LCD Contrast* Display Registers

When UserLCD mode is enabled, any 16 characters can be written to the two UserLCD-line1 and UserLCD-line2 registers. When the data are transferred, they are displayed instead of UserLCD1m* and UserLCD2m* data. The data in the UserLCD-line1 and UserLCD-line2 registers are **not** retained when the unit is switched off.
You can adjust the LCD display contrast in LCD contrast register. This setting is retained at power off.

*: Only applicable for EXDUL-519E/PoE, no function for EXDUL-519S!

13. Installing the Driver

The Ethernet module EXDUL-519 does not require a driver.
All that is required, is an existing network connection from a PC (network interface card with driver) or from a mobile device. Direct access to the module requires the TCP/IP libraries available in many high-level languages such as C, C++, C#, Visual Basic or Java.
Examples of how to access the module in several programming languages can be found on our website www.messcomp.com.

14. Programming

14.1 Getting started

You can program using the standard TCP/IP protocol and therefore via TCP/IP sockets, for which standard libraries are available in the most common programming languages. The use of Microsoft‘s .Net Frameworks allows easy and quick implementation. By using the standard protocol, the module can also be connected to a variety of other operating systems besides Windows, such as Ubuntu (Linux-based) or Android. You can find a number of programming examples on our website.

14.2 Programming Methodes

There are several ways to access the EXDUL module. For example, the EXDUL.dll library can be used for programming under Windows and .NET. This allows a quick and easy way of programmming access to the module. You can also use TCP and Socket libraries which are available in a variety of programming languages such as C# or Java. They often provide a wide range of interface settings. LabVIEW users can also easily access the module using the EXDUL.dll or TCP function blocks.

14.3 Programming under Windows using the .NET EXDUL.dll Library

If you use a .NET programming language (C#, C++, .NET or VB.NET) to access to the module, you can use the EXDUL.dll Library. It is structured object-oriented, so each EXDUL module is represented by an object with its methods. During the developing the library, particular attention was paid to ensuring an API between the different EXDUL modules to be as consistent as possible. This enables the user to change, for example, from a USB EXDUL module to an Ethernet EXDUL module (e.g. EXDUL-319 -> EXDUL-519) without extensive programming.

Command Overview for EXDUL.dll Library

Open:
bool Open()
Return values: true on success / false on error
Result connect to the module

- -

Close
void Close()
Result: Close the connection to the module

- -

Write to the info register:
void SetModulInfo (byte type, string info)
Parameter: type: Info type (see manual)
info: Info string of up to 16 characters
Result: writes to the module information registers

Info Area	Info Byte
UserA	0
UserB	1

- -

Read from the info register:
string GetModulInfo(byte type)
Parameter: type: Info type (see manual)
Return values: Returns the ‚type‘ register as a string
Result: Reads the module information registers

Info Area	Info byte
UserA	0
UserB	1
Hardware Identifier	3
Serial Number	4

Write to the UserLCD LCD register:
void SetUserLCD(byte *line*, string *text*)

Parameter: *line*: 0 = 1st line / 1 = 2nd line
text: LCD text of up to 16 characters

Result: Writes the UserLCD registers. The parameter *line* defines the line (0 or 1) and *text* the text of 16 characters.

- -

Write to the UserLCDm LCD register:
void SetUserLCDm(byte *line*, string *text*)

Parameter: *line*: 0 = 1st line / 1 = 2nd line
text: LCD text of up to 16 characters

Result: Writes to the UserLCDm registers. The parameter *line* defines the line (0 or 1) and *text* the text of 16 characters.

- -

Write the LCD mode:
void SetLCDMode(byte *mode*)

Parameter: *mode*: LCD mode

Result: sets the LCD mode

LCD mode	LCD mode byte
IO mode	0
User mode	1

Read the LCD mode:
byte GetLCDMode()

Return values: LCD mode

Result: Reads the LCD mode

LCD mode	LCD mode byte
IO mode	0
User mode	1

- -

Write LCD Contrast Value:
void SetLCDContrast(ushort *contrast*)

Parameter: contrast:Value between 0 and 4095 (recommended 800 to 1800)

Result: Sets the LCD contrast

- -

Read LCD Contrast Value:
ushort GetLCDContrast()

Return values: LCD contrast

Result: Reads the LCD contrast

- -

Read Optocoupler outputs:
uint GetOptoOut()
Return values: State of the optocoupler outputs
Result: Reads the state of the optocoupler outputs

Write Optocoupler outputs:
void SetOptoOut(uint *value*)
Parameter: *value*: state of the outputs
Result: Sets the optocoupler outputs

Write Optocoupler output:
void WriteOptoOut(byte channel, uint *value*)
Parameter: *channel*: Index of the output channel
value: state of the ouput
Result: Sets one optocoupler output

Read Optocoupler inputs:
uint GetOptoIn()
Return values: current state of the optocoupler inputs
Result: Reads the current state of the optocoupler inputs

Start Counter:

void StartCounter(byte *index*)

Parameter: *index*: Counter index

Result: Starts the counter with the number index

Stop Counter:

void StopCounter(byte *index*)

Parameter: *index*: Counter index

Result: Stops the counter with the number index

Reset Counter:

void ResetCounter(byte *index*)

Parameter: *index*: Counter index

Result: Sets the counter reading of the counter with the number index back to 0

Read Counter:

uint ReadCounter(byte *index*)

Parameter: *index*: Counter index

Return values: Counter reading

Result: Reads the counter reading of the counter with the number index

---

Read Overflow Flag:

bool ReadOverflowFlagCounter(byte *index*)

Parameter:	*index*: Counter index
Return values:	Overflow flag false = no overflow true = overflow
Result:	Reads the overflow flag of the counter with the number index

- -

Reset Overflow Flag:

void ResetOverflowFlagCounter(byte *index*)

Parameter:	*index*: Counter index
Result:	Resets the overflow flag of the counter with the number index

- -

Reset to Factory Default:

void DefaultReset()

Result:	Resets the module to factory defaults. After this command, the module must be shut down and restarted.

14.4 Programming with TCP Libraries

By accessing the module using standard TCP/IP socket libraries, you can program your application in a variety of languages on different platforms. On Windows you can use Delphi or Java in addition to the .Net Framework. Applications can also be developed on Linux based operation systems in languages such as C or Java. Please make sure that your module always acts as a server.

14.4.1 Communication with EXDUL-519

Data is exchanged by sending and receiving byte arrays of variable length via the TCP/IP interface.

Each valid transmission string has a defined result string or confirmation string as a response.

Before sending a string, the last result or confirmation string has to be read.

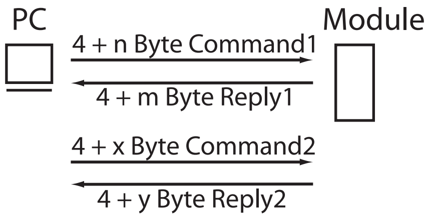


Fig. 14.1 Communication model

14.4.2 Command and Data Format

Data is exchanged by sending and receiving byte arrays. Each byte array to be sent or received consists of at least 4 bytes. The first three bytes represent the command and the fourth byte indicates the number of the 4byte blocks to follow.

Command Byte 0	Command Byte 1	Command Byte 2	Length Byte

The number of the 4byte blocks varies from command to command and depends in part on the volume of data to be sent. More detailed information can be found in the individual command descriptions.

14.4.3 Password protection

In order to protect the module from unauthorised access, a simple password protection can be used for data exchange. If this is enabled (see security configuration command), the correct password (+8 bytes) has to be appended to each byte array sent. If the password is incorrect or has not been added to the command string, an error response is returned. The response of the command remains unchanged.

Example: write optocoupler outputs with password protection enabled and default password „11111111“

Byte	Send	Receive	Notes
0	08	08	Command code 1st Byte
1	00	00	Command code 2nd Byte
2	00	00	Command code 3rd Byte
3	03 (→ 12Byte)	00	Length prefix byte
4	00		r/w Byte
5	0w 00 (locked) 01 (enabled)		Optocoupler status
6	00		reserved
7	00		reserved
8	31$_{hex}$		Password 1st character 1$_{asci}$
9	31$_{hex}$		Password 2nd character 1$_{asci}$
10	31$_{hex}$		Password 3rd character 1$_{asci}$
11	31$_{hex}$		Password 4th character 1$_{asci}$
12	31$_{hex}$		Password 5th character 1$_{asci}$
13	31$_{hex}$		Password 6th character 1$_{asci}$
14	31$_{hex}$		Password 7th character 1$_{asci}$
15	31$_{hex}$		Password 8th character 1$_{asci}$

The regular command without password protection consists of only 8 bytes, and the length prefix byte is 1. As shown in the table above, 8 more bytes have now been added containing the password. Additionally, the length byte has to be increased by 2 (+8 bytes). The response contains no password and corresponds to the usual response array without a password.

14.4.4 Command overview

Hex code	Notes
0C 00 00	Read and write info register
0C 00 03	Read and write LCD register
0C 00 08	Read and write network configuration
0C 00 0C	Read and write security configuration
0C 00 0D	Change password
08 00 00	Read and write optocoupler outputs
08 00 01	Edit optocoupler inputs
09 00 00	Counter0
09 00 01	Counter1
09 00 02	Counter2
09 00 03	Counter3
09 00 04	Counter4
0C 01 01	Software Watch Dog Timer
0C 02 10	Initialisation of a programmable logic branch
0C 03 00	Activate/deactivate receiver mode + read message counter
FF 00 00	Read/reset error register

14.4.5 Command structure

Writing to Info Registers

The EXDUL module provides several writable information registers. UserA/B are two 16-byte areas for the user to store information into a non-volatile memory (FLASH). The registers can only be written as a complete 16-byte block.

Info area	Info byte
UserA	0
UserB	1

Example: Enter the string EXDUL-519 into the UserA and UserB registers

Byte	Send	Receive	Notes
0	0C	0C	Command code 1st byte
1	00	00	Command code 2nd byte
2	00	00	Command code 3rd byte
3	05	00	Lenght prefix byte => 20 byte
4	00 (UserA) 01 (UserB)		Info byte
5	00		reserved
6	00		reserved
7	00		Info area for write function
8	45		Data 1st character E$_{asci}$
9	58		Data 2nd character X$_{asci}$
10	44		Data 3rd character D$_{asci}$
11	55		Data 4th character U$_{asci}$
12	4C		Data 5th character L$_{asci}$
13	2D		Data 6th character -$_{asci}$
14	35		Data 7th character 5$_{asci}$
15	31		Data 8th character 1$_{asci}$
16	39		Data 9th character 9$_{asci}$
17	20		Data 10th character [space]$_{asci}$
18	20		Data 11th character [space]$_{asci}$
19	20		Data 12th character [space]$_{asci}$
20	20		Data 13th character [space]$_{asci}$
21	20		Data 14th character [space]$_{asci}$
22	20		Data 15th character [space]$_{asci}$
23	20		Data 16th character [space]$_{asci}$

Reading the info registers

The EXDUL module provides several 16-byte information areas containing module information such as the serial number or the hardware identifier. Additionally, the user can also read out the writable user registers.

Info area	Info byte
UserA	0
UserB	1
Hardware ID	3
Serial Number	4

Information: All info areas can only be read as a complete 16-byte block.

Example: Read information area UserA (user string = „EXDUL-519“)
An 8-byte block is sent and a 20-byte block is received with the contents of UserA or UserB

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01	Length byte → 4 byte	04	Length byte → 16 byte
4	00 (UserA) 01 (UserB)	Information byte	45	Data 1st character E$_{asci}$
5	00	reserved	58	Data 2nd character X$_{asci}$
6	00	reserved	44	Data 3rd character D$_{asci}$
7	01	Read function info area	55	Data 4th character U$_{asci}$
8			4C	Data 5th character L$_{asci}$
9			2D	Data 6th character -$_{asci}$
10			35	Data 7th character 5$_{asci}$
11			31	Data 8th character 1$_{asci}$
12			39	Data 9th character 9$_{asci}$
13			20	Data 10th character [space]$_{asci}$
14			20	Data 11th character [space]$_{asci}$
15			20	Data 12th character [space]$_{asci}$
16			20	Data 13th character [space]$_{asci}$
17			20	Data 14th character [space]$_{asci}$
18			20	Data 15th character [space]$_{asci}$
19			20	Data 16th character [space]$_{asci}$

Example: Read out information area hardware identifier
An 8-byte block is sent and a 20-byte block containing the hardware identifier is received

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01	Length byte → 4 byte	04	Length byte → 16 byte
4	04	Information byte	45	Data 1st character E$_{asci}$
5	00	reserved	58	Data 2nd character X$_{asci}$
6	00	reserved	44	Data 3rd character D$_{asci}$
7	01	Read function info area	55	Data 4th character U$_{asci}$
8			4C	Data 5th character L$_{asci}$
9			2D	Data 6th character -$_{asci}$
10			35	Data 7th character 5$_{asci}$
11			31	Data 8th character 1$_{asci}$
12			39	Data 9th character 9$_{asci}$
13			20	Data 10th character [space]$_{asci}$
14			20	Data 11th character [space]$_{asci}$
15			20	Data 12th character [space]$_{asci}$
16			20	Data 13th character [space]$_{asci}$
17			20	Data 14th character [space]$_{asci}$
18			20	Data 15th character [space]$_{asci}$
19			20	Data 16th character [space]$_{asci}$

Example: Read out information area serial number
An 8-byte block is sent and a 20-byte block is received with serial number

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd Byte	00	Command code 3rd byte
3	01	Length byte → 4 byte	03	Length byte → 16 byte
4	04	Information byte	31	Data 1st character 1_{dez}
5	00	reserved	30	Data 2nd character 0_{dez}
6	00	reserved	34	Data 3rd character 4_{dez}
7	01	Read function info area	34	Data 4th character 4_{dez}
8			30	Data 5th character 0_{dez}
9			32	Data 6th character 2_{dez}
10			36	Data 7th character 6_{dez}
11				reserved
12				reserved
13				reserved
14				reserved
15				reserved
16				reserved
17				reserved
18				reserved
19				reserved

Writing to LCD registers

The EXDUL module provides several writable LCD registers. UserLCD1 and UserLCD2 correspond to the two lines when the LCD display is used in UserMode. UserLCD1m and UserLCD2m are two 16-byte areas that are stored directly in non-volatile memory (FLASH) and loaded into the registers UserLCD1m or UserLCD2m when the module is started. All registers are writable only as a complete 16-byte block.

LCD command	LCD command byte
UserLCD1	0
UserLCD2	1
UserLCD1m	2
UserLCD2m	3

Example: enter the string EXDUL-519 to the register

Byte	Send	Receive	Notes
0	0C	0C	Command code 1st byte
1	00	00	Command code 2nd byte
2	03	03	Command code 3rd byte
3	05	00	Lenght prefix byte => 20 byte
4	00 (UserLCD1) 01 (UserLCD2) 02 (UserLCD1m) 03 (UserLCD2m)		LCD command
5	00		reserved
6	00		reserved
7	00		write function
8	45		Data 1st character E$_{asci}$
9	58		Data 2nd character X$_{asci}$
10	44		Data 3rd character D$_{asci}$
11	55		Data 4th character U$_{asci}$
12	4C		Data 5th character L$_{asci}$
13	2D		Data 6th character -$_{asci}$
14	35		Data 7th character 5$_{asci}$
15	31		Data 8th character 1$_{asci}$
16	39		Data 9th character 9$_{asci}$
17	20		Data 10th character [space]$_{asci}$
18	20		Data 11th character [space]$_{asci}$
19	20		Data 12th character [space]$_{asci}$
20	20		Data 13th character [space]$_{asci}$
21	20		Data 14th character [space]$_{asci}$
22	20		Data 15th character [space]$_{asci}$
23	20		Data 16th character [space]$_{asci}$

Reading LCD registers

The EXDUL module provides several writable and readable LCD registers. UserLCD1 and UserLCD2 correspond to the two lines when the LCD display is used in UserMode. UserLCD1m and UserLCD2m are two 16-byte areas that are stored directly in non-volatile memory (FLASH) and loaded into the registers UserLCD1m or UserLCD2m when the module is started. All registers are readable only as a complete 16-byte block.

LCD command	LCD command byte
UserLCD1 & UserLCD2	0
UserLCD1m & UserLCD2m	2

Example: reading the string EXDUL-519 from the register

Byte	Send	Representing	Receive	Representing
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	03	Command code 3rd byte	03	Command code 3rd byte
3	01	Length byte → 20 byte	08	Length byte → 20 byte
4	00 (UserLCD1&2) 02 (UserLCD1m&2m)	LCD command	45	Data Line1 1st character E$_{asci}$
5	00	reserved	58	Data Line1 2nd character X$_{asci}$
6	00	reserved	44	Data Line1 3rd character D$_{asci}$
7	01	LCD registers read function	55	Data Line1 4th character U$_{asci}$
8			4C	Data Line1 5th character L$_{asci}$
9			2D	Data Line1 6th character -$_{asci}$
10			35	Data Line1 7th character 5$_{asci}$
11			31	Data Line1 8th character 1$_{asci}$
12			39	Data Line1 9th character 9$_{asci}$
13			20	Data Line1 10th character [space]$_{asci}$
14			20	Data Line1 11th character [space]$_{asci}$
15			20	Data Line1 12th character [space]$_{asci}$
16			20	Data Line1 13th character [space]$_{asci}$
17			20	Data Line1 14th character [space]$_{asci}$
18			20	Data Line1 15th character [space]$_{asci}$
19			20	Data Line1 16th character [space]$_{asci}$
20			45	Data Line2 1st character E$_{asci}$
21			58	Data Line2 2nd character X$_{asci}$
22			44	Data Line2 3rd character D$_{asci}$
23			55	Data Line2 4th character U$_{asci}$
24			4C	Data Line2 5th character L$_{asci}$
25			2D	Data Line2 6th character -$_{asci}$
26			35	Data Line2 7th character 5$_{asci}$
27			31	Data Line2 8th character 1$_{asci}$
28			39	Data Line2 9th character 9$_{asci}$
29			20	Data Line2 10th character [space]$_{asci}$
30			20	Data Line2 11th character [space]$_{asci}$
31			20	Data Line2 12th character [space]$_{asci}$
32			20	Data Line2 13th character [space]$_{asci}$
33			20	Data Line2 14th character [space]$_{asci}$
34			20	Data Line2 15th character [space]$_{asci}$
35			20	Data Line2 16th character [space]$_{asci}$

Writing the LCD mode

The LCD display of the module provides several display modes. These can be set using the following command. The LCD mode is stored in a non-volatile memory and is also used after rebooting the module.

LCD mode	LCD mode byte
I/O mode	0
User mode	1

Example: writing the LCD mode

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Befehlscode 1. Byte
1	00	Command code 2nd byte	00	Befehlscode 2. Byte
2	03	Command code 3rd byte	03	Befehlscode 3. Byte
3	02	Length byte → 8 byte	00	Längenbyte → 0 Byte
4	04	LCD command LCD mode		
5	00	reserved		
6	00	reserved		
7	00	Write function		
8	00 (I/O mode) 01 (User mode)	LCD mode		
9	00	reserved		
10	00	reserved		
11	00	reserved		

Reading the LCD mode

The LCD display of the module provides several display modes. The set LCD mode can be read out using the following command.

LCD mode	LCD mode byte
I/O mode	0
User mode	1

Example: reading the LCD mode

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	03	Command code 3rd byte	03	Command code 3rd byte
3	01	Length byte → 4 byte	01	Length byte → 4 byte
4	04	LCD command LCD mode	00 (I/O mode) 01 (User mode	LCD mode
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	01	read function	00	reserved

Writing LCD contrast value

This command is used to set the contrast of the display. Values from 0 to 4095 are accepted. The higher the value, the less contrast is displayed. A comfortable display is achieved in the range of 800 to 1800.

Example: writing display contrast value 800

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	03	Command code 3rd byte	03	Command code 3rd byte
3	02	Length byte → 8 byte	00	Length byte → 0 byte
4	0B	LCD command LCD contrast		
5	00	reserved		
6	00	reserved		
7	00	write function		
8	50	Contrast value (Lowbyte - 00...FF)		
9	03	Contrast value (Highbyte - 00...0F)		
10	00	reserved		
11	00	reserved		

Reading LCD contrast value

This command is used to read out the display contrast. The values can be from 0 to 4095. The higher the value, the less contrast is displayed. A comfortable display is achieved in the range of 800 to 1800.

Example: read display contrast value 800

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	03	Command code 3rd byte	03	Command code 3rd byte
3	01	Length byte → 4 byte	01	Length byte → 4 byte
4	0B	LCD command LCD contrast	50	Contrast value (Lowbyte - 00...FF)
5	00	reserved	03	Contrast value (Highbyte - 00...0F)
6	00	reserved	00	reserved
7	01	read function	00	reserved

Writing Network Configurations

This command is used to change all network configurations such as IP address, subnet mask, hostname, Gateway, DNS addresses and the setting of the DHCP client function.

Example: writing the network configurations
Hostname = „EXDUL-519“, IP = 192.168.0.63, Subnet mask = 255.255.255.0,
Gateway = 192.168.0.1, Primary DNS = 192.168.0.1, Secondary DNS = 217.237.151.115

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	08	Command code 3rd byte	08	Command code 3rd byte
3	0B	Length byte → 44 byte	00	Length byte → 0 byte
4	00	reserved		
5	00	reserved		
6	00	reserved		
7	00	write function		
8	45	Hostname 1st character E$_{asci}$		
9	58	Hostname 2nd character X$_{asci}$		
10	44	Hostname 3rd character D$_{asci}$		
11	55	Hostname 4th character U$_{asci}$		
12	4C	Hostname 5th character L$_{asci}$		
13	2D	Hostname 6th character -$_{asci}$		
14	35	Hostname 7th character 5$_{asci}$		
15	31	Hostname 8th character 1$_{asci}$		
16	39	Hostname 9th character 9$_{asci}$		
17	20	Hostname 10th character [space]$_{asci}$		
18	20	Hostname 11th character [space]$_{asci}$		
19	20	Hostname 12th character [space]$_{asci}$		
20	20	Hostname 13th character [space]$_{asci}$		
21	20	Hostname 14th character [space]$_{asci}$		
22	20	Hostname 15th character [space]$_{asci}$		
23	20	Hostname 16th character [space]$_{asci}$		
24	3F	4th byte IP address decimal value 63		
25	0	3rd byte IP address decimal value 0		
26	A8	2nd byte IP address decimal value168		
27	C0	1st byte IP address decimal value 192		

28	00	4th byte Subnet mask decimal value 0		
29	FF	3rd byte Subnet mask decimal value 255		
30	FF	2nd byte Subnet mask decimal value 255		
31	FF	1st byte Subnet mask decimal value 255		
32	01	Gateway 4th byte decimal value 1		
33	00	Gateway 3rd byte decimal value 0		
34	A8	Gateway 2nd byte decimal value 168		
35	C0	Gateway 1th byte decimal value 192		
36	01	Primary DNS 4th byte decimal value 1		
37	00	Primary DNS 3rd byte decimal value 0		
38	A8	Primary DNS 2nd byte decimal value 168		
39	C0	Primary DNS 1st byte decimal value 192		
40	73	Secondary DNS 4th byte decimal value 115		
41	97	Secondary DNS 3rd byte decimal value 151		
42	ED	Secondary DNS 2nd byte decimal value 237		
43	D9	Secondary DNS 1st byte decimal value 217		
44	0w 00 (DHCP disable) 01 (DHCP enable)	DCHP client configuration		
45	00	reserved		
46	00	reserved		
47	00	reserved		

Reading network configurations

This command is used to read all network configurations such as IP address, subnet mask, hostname, Gateway, DNS addresses and the setting of the DHCP client function or MAC address.

Example: reading the network configuration
Hostname = „EXDUL-519“, IP = 192.168.0.63, Subnet mask = 255.255.255.0,
Gateway = 192.168.0.1, Primary DNS = 192.168.0.1, Secondary DNS = 217.237.151.115

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	08	Command code 3rd byte	08	Command code 3rd byte
3	01	Length byte → 4 byte	0C	Length byte → 48 byte
4	00	reserved	45	Hostname 1st character E$_{asci}$
5	00	reserved	58	Hostname 2nd character X$_{asci}$
6	00	reserved	44	Hostname 3rd character D$_{asci}$
7	01	Read function	55	Hostname 4th character U$_{asci}$
8			4C	Hostname 5th character L$_{asci}$
9			2D	Hostname 6th character -$_{asci}$
10			35	Hostname 7th character 5$_{asci}$
11			31	Hostname 8th character 1$_{asci}$
12			39	Hostname 9th character 9$_{asci}$
13			20	Hostname 10th character [space]$_{asci}$
14			20	Hostname 11th character [space]$_{asci}$
15			20	Hostname 12th character [space]$_{asci}$
16			20	Hostname 13th character [space]$_{asci}$
17			20	Hostname 14th character [space]$_{asci}$
18			20	Hostname 15th character [space]$_{asci}$
19			20	Hostname 16th character [space]$_{asci}$
20			3F	4th byte IP address decimal value 63
21			0	3rd byte IP address decimal value 0
22			A8	2nd byte IP address decimal value168
23			C0	1st byte IP address decimal value 192
24			00	4th byte Subnet mask decimal value 0
25			FF	3rd byte Subnet mask decimal value 255
26			FF	2nd byte Subnet mask decimal value 255
27			FF	1st byte Subnet mask decimal value 255

28			01	Gateway 4th byte decimal value 1
29			00	Gateway 3rd byte decimal value 0
30			A8	Gateway 2nd byte decimal value 168
31			C0	Gateway 1th byte decimal value 192
32			01	Primary DNS 4th byte decimal value 1
33			00	Primary DNS 3rd byte decimal value 0
34			A8	Primary DNS 2nd byte decimal value 168
35			C0	Primary DNS 1st Bbyte decimal value 192
36			73	Secondary DNS 4th byte decimal value 115
37			97	Secondary DNS 3rd byte decimal value 151
38			ED	Secondary DNS 2nd byte decimal value 237
39			D9	Secondary DNS 1st byte decimal value 217
40			0w 00 (DHCP disable) 01 (DHCP enable)	DCHP client configuration
41			00	Reserved
42			00	Reserved
43			00	Reserved
44				Reserved
45				Reserved
46			00	MAC address 6th character
47			00	MAC address 5th character
48			00	MAC address 4th character
49			3E	MAC address 3rd character
50			B4	MAC address 2nd character
51			D4	MAC address 1st character

Writing the security configuration

The EXDUL module provides a password protection for secure communication. If this is enabled, the correct password needs to be sent with each transfer to the module. It is disabled by default.

Security configuration	Security byte
Password disabled	0
Password enabled	1

Example: writing the security configuration

Byte	Send	Receive	Notes
0	0C	0C	Command code 1st byte
1	00	00	Command code 2nd byte
2	0C	0C	Command code 3rd byte
3	01	01	Length byte → 20 byte
4	00 (Password disabled) 01 (Password enabled)		Security byte
5	00		reserved
6	00		reserved
7	00		Write function security configuration

Reading the Security configuration

The EXDUL module provides a password protection for secure communication. If this is enabled, the correct password needs to be sent with each transfer to the module. It is disabled by default.

Security configuration	Security byte
Password disabled	0
Password enabled	1

Example: Reading the security configuration

Byte	Send	Receive	Notes
0	0C	0C	Command code 1st byte
1	00	00	Command code 2nd byte
2	0C	0C	Command code 3rd byte
3	01	01	Length byte → 20 byte
4	00	00 (Password disabled) 01 (Password enabled)	Security byte
5	00	00	reserved
6	00	00	reserved
7	01	00	Read function security configuration

Changing the Password

The EXDUL module provides a password protection for secure communication. If this is enabled, the correct password needs to be sent with each transfer to the module. By default, the password is „11111111“ in ASCII and corresponds to that of the web page. This function is used to change the password.

Example: changing the password to „EXDUL519“

Byte	Send	Receive	Notes
0	0C	0C	Command code 1st byte
1	00	00	Command code 2nd byte
2	0D	0D	Command code 3rd byte
3	02	00	Length byte → 20 byte
4	45_{hex}		Password 1st character E_{asci}
5	58_{hex}		Password 2nd character X_{asci}
6	44_{hex}		Password 3rd character D_{asci}
7	55_{hex}		Password 4th character U_{asci}
8	$4C_{hex}$		Password 5th character L_{asci}
9	35_{hex}		Password 6th character 5_{asci}
10	31_{hex}		Password 7th character 1_{asci}
11	39_{hex}		Password 8th character 9_{asci}

Reading Optocoupler Outputs

This command reads the current state of the optocoupler outputs.
Optocoupler enabled = 1, optocoupler disabled = 0

Example: reading the optocoupler output state
An 8-byte block is sent and an 8-byte block is received containing the current optocoupler output state

Output channel	DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0
Screw terminal	8	7	6	5	4	3	2	1
Switching state	0	0	0	0	0	0	1	0

Byte	Send	Notes	Receive	Notes
0	08	Command code 1st byte	08	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01 (→ 4byte)	Length byte	01 (→ 4Byte)	Length byte
4	01	r/w byte (1→ read)	01	read function
5	00	reserved	02	optocoupler output state
6	00	reserved	00	reserved
7	00	reserved	00	reserved

Writing Optocoupler Outputs

This command is used to disable or to enable the output optocouplers.
Optocoupler enabled = 1, optocoupler disabled = 0

Example: output of the state (0x02$_{hex}$) at the optocoupler outputs (= DOUT0 disabled and DOUT1 enabled)
An 8-byte block is sent and a 4-byte block is received as an acknowledgement

Output channel	DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0
Screw terminal	8	7	6	5	4	3	2	1
Switching state	0	0	0	0	0	0	1	0

Byte	Send	Receive	Notes
0	08	08	Command code 1st byte
1	00	00	Command code 2nd byte
2	00	00	Command code 3rd byte
3	01 (→ 4Byte)	00	Length byte
4	00		r/w byte (0→ write)
5	02		Optocoupler state
6	00		reserved
7	00		reserved

Writing Optocoupler Outputs Separately

This command is used to disable or enable individual output optocouplers.

Optocoupler enabled = 1, optocoupler disabled = 0

Example: DOUT1 is to be enabled independently of the other outputs
An 8-byte block is sent and an 4-byte block is received as an acknowledgement

Output channel	DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0
Screw terminal	8	7	6	5	4	3	2	1
Channel index	7	6	5	4	3	2	1	0

Byte	Send	Receive	Notes
0	08	08	Command code 1st byte
1	00	00	Command code 2nd byte
2	00	00	Command code 3rd byte
3	01 (→ 4byte)	00	Length byte
4	02		r/w (2→ write individually)
5	01		Channel index
6	01		Optocoupler state
7	00		reserved

Reading the Optocoupler Inputs

This command is used to read the current status of the optocoupler inputs

Example: reading the optocoupler input status
A 4-byte block is sent and an 8-byte block is received containing the optocoupler input states.
These examples assume that the input levels (here 0xB3) are applied to the single inputs according to the following table:
(0 = LOW = 0...3V, 1 = HIGH = 10..30V)

	HB			LB							
Output channel	DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
Screw terminal	21	20	19	18	17	16	15	14	13	12	11
Switching state	0	0	1	1	0	1	1	0	0	1	1

Byte	Send	Notes	Receive	Notes
0	08	Command code 1st byte	08	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	01	Command code 3rd byte	01	Command code 3rd byte
3	00	Length byte	01 (→ 4byte)	Length byte
4			B3	Optocoupler input state LB (DIN7-DIN0)
5			01	Optocoupler input state HB (DIN10-DIN8)
6			00	reserved
7			00	reserved

Counters

This command gives access to the counters. The counter can be started, stopped, reset an read. In addition, the overflow flag can be read and reset

Code	Counter command code
00	start counter
01	stop counter
02	reset counter
03	read counter value
04	reserved
05	read overflow flag
06	reset overflow flag

Example as for counter0:

Start / Stop / Reset counter0

Byte	Send	Notes	Receive	Notes
0	09	Command code 1st byte	09	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01	Length byte	01	Length byte
4	bb 00 01 02	Counter command code Start counter0 Stop counter0 Reset counter0	bb	Counter command code
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	00	reserved

Read counter0

Byte	Send	Notes	Receive	Notes
0	09	Command code 1st byte	09	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01	Length byte	02 (→ 8byte)	Length byte
4	03	Counter command code	03	Counter command code
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	00	reserved
8			ww	Counter reading byte0
9			ww	Counter reading byte1
10			ww	Counter reading byte2
11			ww	Counter reading byte3

Counter reading = counter status byte3 * 0x1000000 + counter status byte2 * 0x10000 + counter status byte1 * 0x100 + counter status byte0

Read overflow flag counter0

Byte	Send	Notes	Receive	Notes
0	09	Command code 1st byte	09	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01	Length byte	02 (→ 8byte)	Length byte
4	05	Counter command code Read overflow flag	05	Counter command code Read overflow flag
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	0f	Overflow flag

Reset overflow flag counter0

Byte	Send	Notes	Receive	Notes
0	09	Command code 1st byte	09	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01	Length byte	01 (→ 4byte)	Length byte
4	06	Counter command code Reset overflow flag	06	Counter command code Reset overflow flag
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	00	reserved

Watchdog Timer

This command gives access to the counters. The counter can be started, stopped, reset an read. In addition, the overflow flag can be read and reset.

Code	Counter command code
00	Start watchdog timer
01	Stop watchdog timer
02	Reset watchdog timer
03	Set WDT time interval

Start / Stop / Reset the watchdog timer

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	01	Command code 2nd byte
2	01	Command code 3rd byte	01	Command code 3rd byte
3	01	Length byte	01	Length byte
4	bb 00 01 02 03	Command code WDT Start WDT Stop WDT Reset WDT Set WDT time interval	bb	Counter command code
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	00	reserved

Set watchdog timer interval in ms

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	00	Command code 2nd byte	01	Command code 2nd byte
2	01	Command code 3rd byte	01	Command code 3rd byte
3	02	Length byte	01	Length byte
4	03	Set WDT time interval	bb	Command code WDT
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	00	reserved
8	pp_0	Time interval byte0		
9	pp_1	Time interval byte1		
10	pp_2	Time interval byte2		
11	pp_3	Time interval byte3		

Time interval = byte3 * 0x1000000 + byte2 * 0x10000 + byte1 * 0x100 + byte0 [ms]

Initialising of a programmable logic branch

This command enables one of the programmable logic branches to be initialised.

Code	Input function
00	NONE
01	TRUE (logic 1)
02	FALSE (logic 0)
03 .. 15	reserved
16	Status DIN0
17	Status DIN1
18	Status DIN2
19	Status DIN3
20	Status DIN4
21	Status DIN5
22	Status DIN6
23	Status DIN7
24	Status DIN8
25	Status DIN9
26	Status DIN10
27 .. 31	reserved
32	Rising edge DIN0 (event)
33	Rising edge DIN1 (event)
34	Rising edge DIN2 (event)
35	Rising edge DIN3 (event)
36	Rising edge DIN4 (event)
37	Rising edge DIN5 (event)
38	Rising edge DIN6 (event)
39	Rising edge DIN7 (event)
40	Rising edge DIN8 (event)
41	Rising edge DIN9 (event)
42	Rising edge DIN10 (event)

Code	Gatter function
00	AND operation
01	OR operation

Code	Output function
00	NONE (branch disabled)
04	Send Message1 (event)
05	Send Message2 (event)
06	Send Message3 (event)
07	Send Message4 (event)
16	Write DOUT0
17	Write DOUT1
18	Write DOUT2
19	Write DOUT3
20	Write DOUT4
21	Write DOUT5
22	Write DOUT6
23	Write DOUT7
32	Set DOUT0 (event)
33	Set DOUT1 (event)
34	Set DOUT2 (event)
35	Set DOUT3 (event)
36	Set DOUT4 (event)
37	Set DOUT5 (event)
38	Set DOUT6 (event)
39	Set DOUT7 (event)
48	Clear DOUT0 (event)
49	Clear DOUT1 (event)
50	Clear DOUT2 (event)
51	Clear DOUT3 (event)
52	Clear DOUT4 (event)
53	Clear DOUT5 (event)
54	Clear DOUT6 (event)
55	Clear DOUT7 (event)
64	Toggle DOUT0 (event)
65	Toggle DOUT1 (event)
66	Toggle DOUT2 (event)
67	Toggle DOUT3 (event)
68	Toggle DOUT4 (event)
69	Toggle DOUT5 (event)
70	Toggle DOUT6 (event)
71	Toggle DOUT7 (event)

Example: Initialising the first logic branch
Message1 is to be sent to the PC on each rising edge of the optocoupler input DIN0 (event).
(IN0 = 32, IN1 = TRUE, IN2 = TRUE, IN3 = TRUE, Gate = 0, OUT = 4)

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	02	Command code 2nd byte	02	Command code 2nd byte
2	10	Command code 3rd byte	10	Command code 3rd byte
3	01	Length byte → 4 byte	01	Length byte → 4 byte
4	00	Read/write byte	00	Read/write byte
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	01	Branch index	00	reserved
8	20	Function IN0		
9	00	reserved		
10	00	reserved		
11	00	reserved		
12	01	Function IN1		
13	00	reserved		
14	00	reserved		
15	00	reserved		
16	01	Function IN2		
17	00	reserved		
18	00	reserved		
19	00	reserved		
20	01	Function IN3		
21	00	reserved		
22	00	reserved		
23	00	reserved		
24	00	Gatter function		
25	00	reserved		
26	00	reserved		
27	00	reserved		
28	04	Output funkcion OUT0		
29	00	reserved		
30	00	reserved		
31	00	reserved		

Reading the error registers

This command is used to read the two error registers

Byte	Send	Notes	Receive	Notes
0	FF	Command code 1st byte	FF	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01	Length byte	03	Length byte
4	00	Read error register	00	Read error register
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	00	reserved
8			aa0	Register0 byte0
9			aa1	Register0 byte1
10			aa2	Register0 byte2
11			aa3	Register0 byte3
12			bb0	Register1 byte0
13			bb1	Register1 byte1
14			bb2	Register1 byte2
15			bb3	Register1 byte3

Error register0 = aa3 * 0x1000000 + aa2 * 0x10000 + aa1 * 0x100 + aa0
Error register1 = bb3 * 0x1000000 + bb2 * 0x10000 + bb1 * 0x100 + bb0

Resetting the error registers

This command is used to reset the two error registers

Byte	Send	Notes	Receive	Notes
0	FF	Command code 1st byte	FF	Command code 1st byte
1	00	Command code 2nd byte	00	Command code 2nd byte
2	00	Command code 3rd byte	00	Command code 3rd byte
3	01	Length byte	01	Length byte
4	01	Reset error register	01	Reset error register
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	00	reserved

Enabling the receiver mode

This command is used to enable the programmable logic receiver mode. After a second TCP/IP connection has been established with the module, the receiver mode is enabled for this connection. There is no respond to this command. When this mode is enabled, then only the receiver commands work over this TCP/IP connection. All other commands, such as reading inputs, must be carried out via the first connection. To disable the receiver mode, use the „disable receiver mode“ command on the receiver connection, or terminate the connection.

Byte	Send	Notes
0	0C	Command code 1st byte
1	03	Command code 2nd byte
2	00	Command code 3rd byte
3	01	Length byte → 4 byte
4	00	Enable receiver mode
5	00	reserved
6	00	reserved
7	00	reserved

Reading the receiver mode counter (receiver mode command)

This command is used to read the current receiver mode counter state. It can only be used when connected in receiver mode. With this command you can check whether all messages have been received. The count should be 1 higher than the last message index count.
Counter index = dd * 0x1000000 + cc * 0x10000 + bb * 0x100 + aa

Byte	Send	Notes	Receive	Notes
0	0C	Command code 1st byte	0C	Command code 1st byte
1	01	Command code 2nd byte	01	Command code 2nd byte
2	02	Command code 3rd byte	02	Command code 3rd byte
3	01	Length byte	02 (→ 8byte)	Length byte
4	02	Read receiver rmode counter	02	Read receiver rmode counter
5	00	reserved	00	reserved
6	00	reserved	00	reserved
7	00	reserved	00	reserved
8			aa	Receiver mode index byte0
9			bb	Receiver mode index byte1
10			cc	Receiver mode index byte2
11			dd	Receiver mode index byte3

Disabling the receiver mode counter (receiver mode command)

This command is used to disable the programmable logic receiver mode. This command does not receive a response and only works with an enabled receiver connection.

Byte	Send	Notes
0	0C	Command code 1st byte
1	03	Command code 2nd byte
2	00	Command code 3rd byte
3	01	Length byte → 4byte
4	01	Disable receiver mode
5	00	reserved
6	00	reserved
7	00	reserved

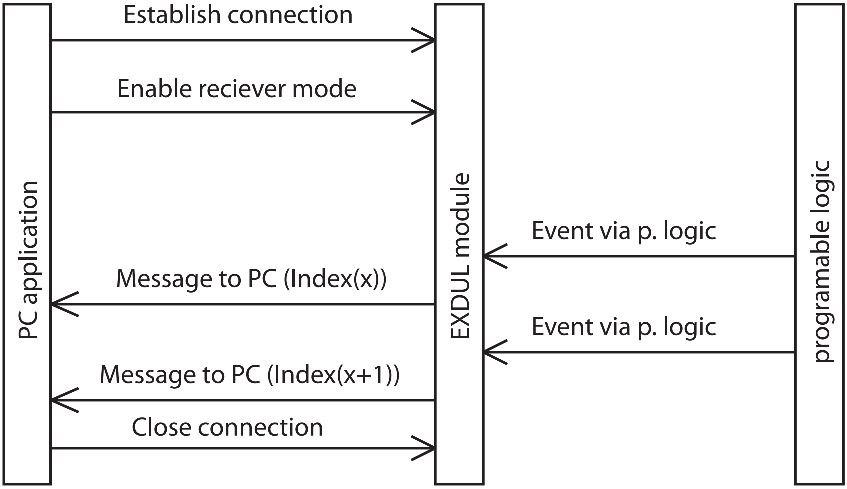
14.5 Receiver mode

It is often the case that a change must be checked at regular intervals, e.g at the optocoupler inputs. This is usually done by polling the respective inputs. This can quickly lead to a considerable load on the network if the inputs are polled frequently and several modules are in use at the same time. To prevent this, programmable logic is available that can send a message to the PC when, for example, an optocoupler input has changed. This eliminates the need for regular polling of the optocoupler input.

To receive these messages from the programmable logic, a separate TCP/IP connection (NetworkStream) must be established with the module. This connection has to be set to the receiver mode using a command. After that, the user application is able to wait for the messages from the programmable logic. This can be synchronous or asynchronous or outsourced to an external task, depending on requirements and capabilities.

The message does not need to be acknowledged in the program.

The module supports multiple TCP/IP connections, but only one of them can be in receiver mode at any one time. When the receiver mode is no longer required, either reset the connection to a normal TCP/IP connection by command or simply close the connection.



Please refer to our programming examples (e.g. Windows-C# examples) for a more detailed illustration of the receiver mode.

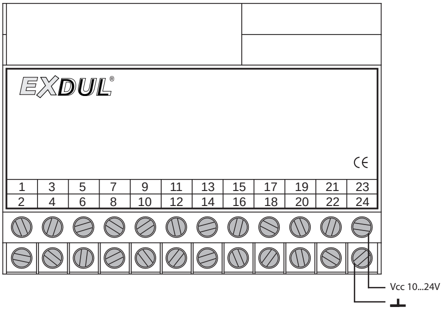
14.6 Module access via LabVIEW and EXDUL.dll

Thanks to the EXDUL.dll file, the module can be implemented in a LabVIEW project without a great deal of effort. Besides LabVIEW and the EXDUL. dll file, the .NET-Framework is required on your PC.
See the EXDUL LabVIEW Tutorial for more information.

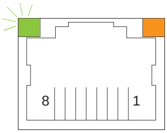
15. FAQ - Troubleshooting

The following is a brief summary of the most common causes of malfunctions that may occur during commissioning or when accessing the EXDUL-519 or the EXDUL website. Please first check the following points before contacting your distributor:

Is the EXDUL-519 supply voltage connected correctly?
A power supply of +10V ... +30V DC via terminal 23 (Vcc) and terminal 24 (GND) or a PoE capable switch for the EXDUL-519PoE is required for operation. Please check the screw terminal connections on the module as well as the power supply unit and the connections on the power supply unit.



Does the Connect LED on the RJ-45 jack light up green constantly?
After the operating voltage is applied, the EXDUL-519 boots. As soon as a stable Ethernet connection is established, the LED on the 8P8C modular connector (RJ-45 jack) will light up green constantly.
If this is not the case, please check the direct cable connection between EXDUL-519 and the computer (crossover cable may be required). For network operation, check the network cables between EXDUL-519 and the wall socket, the active Ethernet switch or the Ethernet hub.



Is there a stable Ethernet connection between the PC and the network?
Check the network cable between the computer and the network outlet (RJ-45 wall socket), active Ethernet switch or Ethernet hub. The Ethernet cable must be suitable for the Ethernet connnection, undamaged and correctly plugged in at both ends. Current computers usually provide two LEDs on the network adapter socket. When connected to the network, the green LED will light up continuously. If the computer‘s network socket has only one LED, it will flash or flicker when the network connection is in operation.

Is the network cable used suitable for the connection?
When connecting the EXDUL-519 to a switch, hub or PC with an Auto-MDI(X) supporting Ethernet interface, a standard network cable (Cat 5 or better) can be used. Older computers whose Ethernet interface does not automatically cross the send and receive lines may require a crossover cable or crossover adapter.

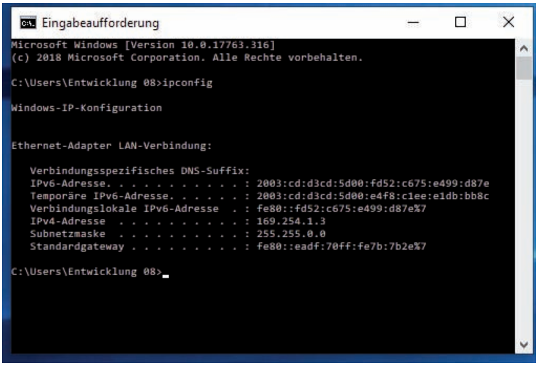
Is the wall outlet active?
If you are connecting the EXDUL-519 to a fixed network via a wall outlet, consult your network administrator to ensure that the wall socket is active and connected to an active Ethernet switch or Ethernet hub.

Is the computer‘s Ethernet interface enabled?
The Ethernet adapter must be enabled in the PC‘s BIOS. In the Windows Device Manager, check that it is listed under Network Adapters. It must not be marked with an exclamation mark!

Are the computer‘s network settings correct?
Each enabled device in a TCP/IP based network requires a unique IP address, which is divided into a network ID and a device ID. The subnet mask separates the IP address into a network part (network prefix, network ID) and a device part (device ID). By default, the EXDUL-519 is set to DHCP client.

How to check your PC‘s network configuration?

You can monitor your computer‘s TCP/IP settings from the Internet Protocol Version 4 (TCP/IPv4) Properties window or the LAN connection status (see „How to check and change your PC‘s IP address“), or by using the simple command line IPCONFIG. To do this, switch to the MS-DOS command prompt (see „How to switch to MS-DOS command prompt“), type **ipconfig** and press **Enter** to confirm. The response should look similar to the following figure:



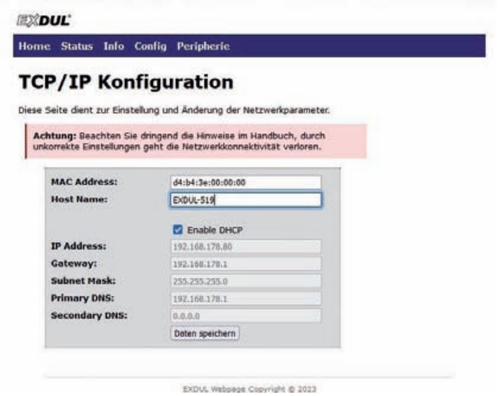
```
Microsoft Windows [Version 10.0.17763.316]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Entwicklung 08>ipconfig

Windows-IP-Konfiguration


Ethernet-Adapter LAN-Verbindung:

   Verbindungsspezifisches DNS-Suffix:
   IPv6-Adresse. . . . . . . . . . . : 2003:cd:d3cd:5d00:fd52:c675:e499:d87e
   Temporäre IPv6-Adresse. . . . . . : 2003:cd:d3cd:5d00:e4f8:c1ee:e1db:bb8c
   Verbindungslokale IPv6-Adresse  . : fe80::fd52:c675:e499:d87e%7
   IPv4-Adresse  . . . . . . . . . . : 169.254.1.3
   Subnetzmaske  . . . . . . . . . . : 255.255.0.0
   Standardgateway . . . . . . . . . : fe80::eadf:70ff:fe7b:7b2e%7

C:\Users\Entwicklung 08>_
```

How to check and change your PC‘s IP address

Windows10:

Start > Control Panel > Network & Internet > Status > Change adapter settings > select the required LAN connection (double click or right click) > Properties > Internetprotocol Version 4 (TCP(IPv4) > Properties

Please note: administrator privileges are required to change TCP/IP settings!



TCP/IP Konfiguration

Diese Seite dient zur Einstellung und Änderung der Netzwerkparameter.

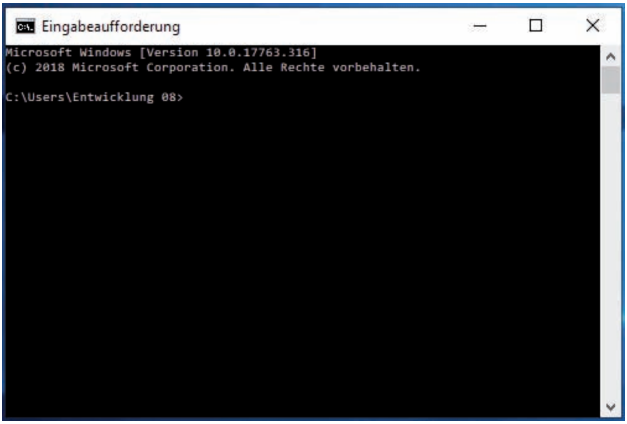
Achtung: Beachten Sie dringend die Hinweise im Handbuch, durch unkorrekte Einstellungen geht die Netzwerkkonnektivität verloren.

MAC Address:	d4:b4:3e:00:00:00
Host Name:	EXDUL-519
	☑ Enable DHCP
IP Address:	192.168.178.80
Gateway:	192.168.178.1
Subnet Mask:	255.255.255.0
Primary DNS:	192.168.178.1
Secondary DNS:	0.0.0.0

EXDUL Webpage Copyright © 2023

How to switch to MS-DOS prompt

Windows10:
Start (right click) > Search programs > type **cmd** > press enter to confirm
or
Start > Windows System > prompt



```
Microsoft Windows [Version 10.0.17763.316]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Entwicklung 08>
```

Is it possible to locate EXDUL-5xx modules in an existing network and to determine the network data?

All EXDUL-5xx modules send identification signals at certain intervals. The program **ExdulUtility**_v2_xx (or later versions) evaluates the identification data and generates a list with the host name, IP address and MAC address. It is suitable for a single EXDUL-5xx connected directly to a PC as well as for a network connected via a hub or switch with several modules. If your firewall prevents the scan program from communicating with the EXDUL-5xx, you will need to enable it.

16. Technical Specification

Optocoupler inputs
11 galvanic isolated bidirectional channels
Over voltage protection diodes
Input voltage range
high = 10..30 Volt
low = 0..3 Volt

Digital Outputs via FET Power Switches
8 channels with opto-coupler galvanic isolation
Contact B of the FET (source, + connection) with one connection terminal each, contact A (drain, - connection) with common connection terminal for all 8 channels
Reverse polarity protection diodes
Switchable recovery diodes
Voltage CE: max. 30V
Output current: max. 1A/channel
Response time: typ. 60µs (24V, 100mA)
Release time: typ. 250µs (24V, 100mA)

Counters
6 programmable 32-bit counters (occupying 6 optocoupler inputs)
Counting frequency: max. 5kHz
Automatic backup of counter readings at 10kHz intervals

Programmable Logic
Switch outputs with logic operations
Send message to PC when input status changes

TCP/IP Server
Switch outputs with logic operations
Send message to PC when input status changes

Web Server
Configuration
Function test
Firmware Update

Timer for Communication Watchdog
Time window adjustable from 1 to 2^{32} - 1ms

LCD Display (EXDUL-519E/PoE only)
2-line, 16-column matrix display with 16 characters per line
Programmable to display application specific data or as I/O status display

Connection Terminals
1 * 24-pin screw terminal block
Ethernet RJ-45 socket

Power Supply
external
Supply voltage: 10 .. 30V
Current: typ. 100mA at 24V
PoE IEEE802.3at 48V typ.

Ethernet connection cable
RJ-45 patch cable Cat5 or better

Dimension
105 mm x 89 mm x 59 mm (l x w x h)

Housing
Insulated housing with integrated snap-on technology for DIN EN top-hat rail mounting.
Suitable for surface mounting, installation in switch and distribution cabinets or as a mobile desk unit.

17. Wiring Examples

17.1 Optocoupler Input Wiring

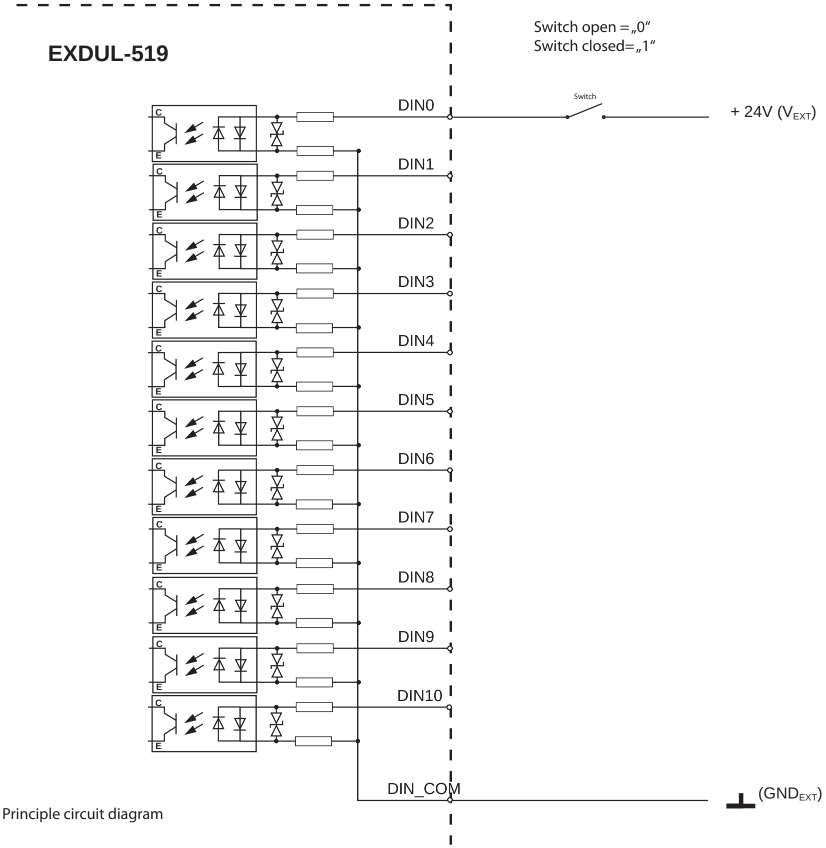


Fig 17.1.1 Optocoupler input wiring

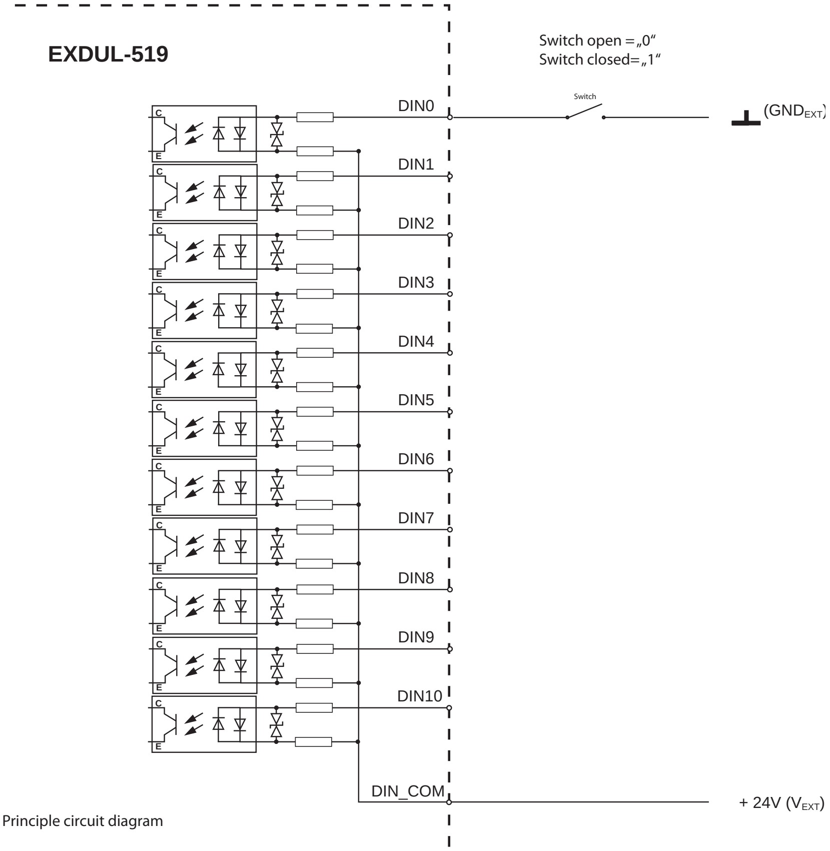


Fig 17.1.2 Optocoupler input wiring reverse

17.2 Optocoupler Output Wiring

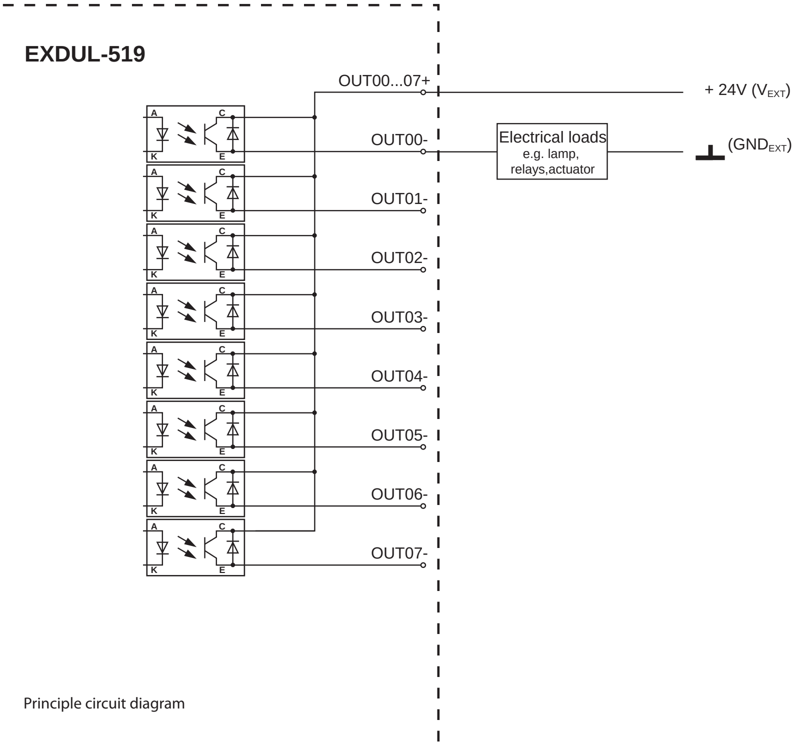


Fig 17.2 Optocoupler output wiring

18. 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	[space]
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	

19. Product Liability Act

Notes on Product Liability

The German Product Liability Act (ProdHaftG) governs the liability of manufacturers for damage caused by defective products.

The obligation to pay compensation may already exist if the appearance of the product could mislead a non-commercial end user into believing that the product is safe, and also if the end-user can be expected to disregard the necessary safety instructions when handling the product.

It must therefore always be possible to prove that the end-user has been made aware of the safety rules.

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

Safety Regulations

The relevant VDE regulations must be observed when handling products that come into contact with electrical voltage.

Particular attention should be paid to the following regulations:
VDE100; VDE0550/0551; VDE0700; VDE0711; VDE0860.
You can obtain the regulations from:
vde-Verlag GmbH
Bismarckstr. 33
10625 Berlin
.

* disconnect the mains plug before opening the unit or ensure, that no power is supplied to the unit.

* You must not put any components, printed circuit boards or devices into service unless they have first been installed in a safe, touch-proof enclosure. There must be no power supply to the equipment during the installation process.

* Ensure that the unit is disconnected from the power supply before using tools on any components, boards or devices. Any electrical charge stored in components inside the device must be discharged first.

* Live cables or wires connected to the unit, components or boards must be inspected for insulation faults or breakage. In case any wire is defective the device must be taken out of service immediately until the defective wire has been replaced.

* When using components or printed circuit boards you must strictly comply with the electrical parameters specified in the relevant description.

* If, as a non-commercial end-user, you are not sure whether the electrical parameters specified in the description supplied are applicable to a component, you must consult an expert.

Furthermore, the user/customer is responsible for the complying with all types of building and safety instructions (VDE, TÜV, Berufsgenossenschaften, etc.) .

20. EC Declaration of Conformity

This is to certify, that the products

EXDUL-519PoE	**EDP number A-374650**
EXDUL-519E	**EDP number A-374640**
EXDUL-519S	**EDP number A-374620**

comply with the requirements of the relevant 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 Class B
IEC 801-2
IEC 801-3
IEC 801-4
EN 50082-1
EN 60555-2
EN 60555-3

This declaration is issued on behalf of the manufactorer

Messcomp Datentechnik GmbH
Odelshamer Str. 2
83512 Wasserburg

by

Dipl.Ing.(FH) Hans Schnellhammer

Wasserburg, 24.06.2024

Reference system - intended use

The EXDUL-519E, EXDUL-519PoE and EXDUL-519S multifunction modules are not intended to be stand-alone devices. CE-conformity can only be assessed if other computer components are in use simultaneously. Therefore, CE conformity can only be confirmed if the following reference system for the intended use of the multifunction modules is used:

Electrical cabinet:	Vero IMRAK 3400	804-530061C 802-563424J 802-561589J
19“ casing:	Vero PC case	145-010108L
19“ casing:	Additional Electronic	519-112111C
Motherboard:	GA-586HX	PIV 1.55
Floppy controller:	on motherboard	
Floppy:	TEAC	FD-235HF
Graphics card:	Advantech	PCA-6443
Interfaces:	EXDUL-519PoE EXDUL-519E EXDUL-519S	A-374650 A-374640 A-374620

Messcomp Datentechnik has been working hard on the topic of sustainability and saving resources and has changed a number of processes.

As a further step in this direction, we have decided to stop including a CD with our manuals.

You will always find all the information and files you need on the Internet at www.messcomp.com.

Software downloads of drivers and sample programs are available free of charge. Please request the password at
www.messcomp.com/php/software_passwort.php.
The password will be emailed to you immediately by the server.