

EXDUL-537E

EDV-Nr.: A-375440

EXDUL-537S

EDV-Nr.: A-375420

12 Eingänge bipolar über Optokoppler
8 Ausgänge über Relais je 2A
6 Zähler 32 Bit
Programmierbare Logik
TCP/IP- und Webserver
LCD-Anzeige (nur EXDUL-537E)

wasco®

Handbuch

Copyright® 2021 by Messcomp Datentechnik GmbH

Diese Dokumentation ist urheberrechtlich geschützt. Alle Rechte sind vorbehalten.

Messcomp Datentechnik GmbH behält sich das Recht vor, die in dieser Dokumentation beschriebenen Produkte jederzeit und ohne Vorankündigung zu verändern.

Ohne schriftliche Genehmigung der Firma Messcomp Datentechnik GmbH darf diese Dokumentation in keinerlei Form vervielfältigt werden.

Geschützte Warenzeichen

Windows®, Visual Basic®, Visual C++®, Visual C#® sind eingetragene Warenzeichen von Microsoft.

wasco® ist ein eingetragenes Warenzeichen.

EXDUL® ist ein eingetragenes Warenzeichen. LabVIEW® ist ein eingetragenes Warenzeichen.

Bei anderen genannten Produkt- und Firmennamen kann es sich um Warenzeichen der jeweiligen Inhaber handeln.

Haftungsbeschränkung

Die Firma Messcomp Datentechnik GmbH haftet für keinerlei durch den Gebrauch des Multifunktionsmoduls EXDUL-537 und dieser Dokumentation direkt oder indirekt entstandenen Schäden.

Wichtiger Hinweis:

Dieses Handbuch wurde für die Module EXDUL-537E und EXDUL-537S erstellt. Das EXDUL-537E bietet zusätzlich eine LCD-Anzeige, alle weiteren Funktionen der Module sind identisch. Für das EXDUL-537S sind die Befehle und Funktionen, die das Display betreffen, nicht zutreffend.

Inhaltsverzeichnis

| | |
|--|----|
| 1. Produktbeschreibung | 5 |
| 2. Anschlussklemmen | 6 |
| 2.1 Klemmenbelegung von CN1 | 6 |
| 3. Systemkomponenten | 7 |
| 3.1 Blockschaltbild EXDUL-537E | 7 |
| 3.2 Blockschaltbild EXDUL-537S | 8 |
| 3.3 Digitale Eingänge über Optokoppler | 9 |
| 3.4 Digitale Ausgänge über Relais | 9 |
| 3.5 Zähler | 9 |
| 3.6 Programmierbare Logik | 9 |
| 3.7 TCP/IP-Server | 10 |
| 3.8 Webserver | 10 |
| 3.9 Kommunikations-Watchdog-Timer | 10 |
| 3.10 LCD Anzeige (nur EXDUL-537E) | 10 |
| 4. Inbetriebnahme | 11 |
| 4.1 Anschluss an einen Ethernet-Port | 11 |
| 4.2 Anschluss der Betriebsspannung | 11 |
| 4.3 Integrierte Webpage des EXDUL-537E / EXDUL-537S | 12 |
| 4.4 Passwortschutz - Zugangskennung | 12 |
| 4.5 Grundeinstellung Netzwerk-Konfiguration | 13 |
| 4.6 Zusammensetzung und Aufbau der IP-Adresse | 13 |
| 4.7 Änderung der Netzwerk-Konfiguration | 14 |
| 4.8 Konfiguration mit statischer IP-Adresse (DHCP deaktiviert) | 15 |
| 4.9 Konfiguration mit dynamischer IP-Adresse (DHCP aktiviert) | 17 |
| 4.10 LCD-Anzeige während des Bootvorgangs (nur EXDUL-537E) | 18 |
| 4.11 LCD-Anzeige während des Betriebs (nur EXDUL-537E) | 19 |
| 5. Zugriff auf das EXDUL-537 | 20 |
| 5.1 Zugriff über die EXDUL Webpage | 20 |
| 5.2 Zugriff über TCP/IP-Sockets | 21 |
| 5.3 Host-Namen, IP-Adresse und MAC-Adresse feststellen | 22 |
| 6. 12 Optokopplereingänge | 23 |
| 6.1 Pinbelegung der Eingangsoptokoppler | 23 |
| 6.2 Eingangsbeschaltung | 24 |
| 6.3 Eingangsstrom | 24 |

| | |
|--|----|
| 7. 8 Relaisausgänge | 25 |
| 7.1 Pinbelegung der Ausgangsrelais | 25 |
| 7.2 Relaisdaten | 25 |
| 7.3 Programmierung der Relaisausgänge | 26 |
| 7.4 Relaisausgänge rücklesen | 26 |
| 8. Zähler | 27 |
| 9. Programmierbare Logik | 28 |
| 9.1 Logikzweig | 29 |
| 9.2 Logikeingänge | 29 |
| 9.3 Logikverknüpfung | 30 |
| 9.4 Logikausgänge | 30 |
| 9.5 Message an PC schicken | 31 |
| 9.6 Timing der programmierbaren Logik | 32 |
| 10. Watchdog-Timer (WDT) | 33 |
| 11. Error-Register | 34 |
| 12. Informations-, LCD- und Userregister | 35 |
| 12.1 Register HW-Kennung und Seriennummer | 35 |
| 12.2 Speicherbereiche UserA, UserB, UserLCD1m* und UserLCD2m* | 36 |
| 12.3 Display-Register UserLCD-Zeile1*, UserLCD-Zeile2* und LCD-Kontrast* | 36 |
| 13. Installation des Treibers | 37 |
| 14. Programmierung | 38 |
| 14.1 Einführung | 38 |
| 14.2 Programmierarten | 38 |
| 14.3 Programmierung unter Windows mit der .NET EXDUL.dll Library | 38 |
| 14.4 Programmierung mit TCP-Libraries | 45 |
| 14.5 Receivermodus | 79 |
| 14.6 Modulzugriff über LabVIEW [®] und EXDUL.dll | 80 |
| 15. FAQ - Problembehandlung | 81 |
| 16. Technische Daten | 86 |
| 17. Beschaltungsbeispiele | 88 |
| 17.1 Beschaltung der Optokoppler-Eingänge | 88 |
| 17.2 Beschaltung der Optokoppler-Ausgänge | 90 |
| 18. ASCII-Tabelle | 91 |
| 19. Produkthaftungsgesetz | 94 |
| 20. EG-Konformitätserklärung | 96 |

1. Produktbeschreibung

Das EXDUL-537E verfügt über 12 digitale Eingänge und acht digitale Ausgänge mit galvanischer Trennung. Die Potentialtrennung der Eingänge, welche mit zusätzlichen Schutzdioden abgesichert sind, ist durch hochwertige Optokoppler gegeben. Leistungsfähige Relais, die einen Schaltstrom von 2A bewältigen, gewährleisten die Trennung bei den Ausgängen.

Sechs der 12 Optokoppler-Eingänge können bei Bedarf auch als hardwareunterstützte 32Bit Zählereingänge programmiert werden.

Über eine externe Spannungsquelle wird das Modul mit der notwendigen Betriebsspannung versorgt. Die programmierbare LCD-Anzeige ermöglicht die Darstellung von digitalen I/O-Statusinformationen oder programmierbaren anwenderspezifischen Daten.

Über die integrierte Webpage kann das Modul benutzerfreundlich konfiguriert werden. Zudem ist auch ein einfacher Funktionstest möglich.

























Die Kommunikation zwischen PC und Modul erfolgt über das Senden und Empfangen von Byte-Arrays über eine stabile TCP/IP-Verbindung, welche durch ein Handshakeprotokoll abgesichert ist.

Die programmierbare Logik des Moduls kann sowohl für autarke Aktionen an den Ausgängen, als auch für Meldungen an den PC genutzt werden. Hierdurch ist oft ein Polling der Eingänge nicht mehr nötig und sowohl der Datenverkehr als auch die Rechnerauslastung kann wesentlich verringert werden.

Die Anschlüsse für die Spannungsversorgung sind wie die Anschlüsse des Eingangs- und Ausgangsoptokopplers einer 24poligen Schraubklemmleiste zugeführt. Das kompakte Gehäuse erlaubt den Einsatz als mobiles Modul am Notebook sowie als Steuermodul im Steuerungs- und Maschinenbau mit einfacher Wandmontage oder unkomplizierter Montage auf DIN EN-Tragschienen.

2. Anschlussklemmen

2.1 Klemmenbelegung von CN1

| | | | |
|-----------------|--|--|-----------------|
| DOUT01A | 2  |  1 | DOUT00A |
| DOUT03A | 4  |  3 | DOUT02A |
| DOUT05A | 6  |  5 | DOUT04A |
| DOUT07A | 8  |  7 | DOUT06A |
| DIN00 / Zähler0 | 10  |  9 | DOUT00 .. 07B |
| DIN02 / Zähler2 | 12  |  11 | DIN01 / Zähler1 |
| DIN04 / Zähler4 | 14  |  13 | DIN03 / Zähler3 |
| DIN06 | 16  |  15 | DIN05 / Zähler5 |
| DIN08 | 18  |  17 | DIN07 |
| DIN10 | 20  |  19 | DIN09 |
| DIN_COM | 22  |  21 | DIN11 |
| GND_EXT | 24  |  23 | Vcc_EXT |

Vcc_EXT:

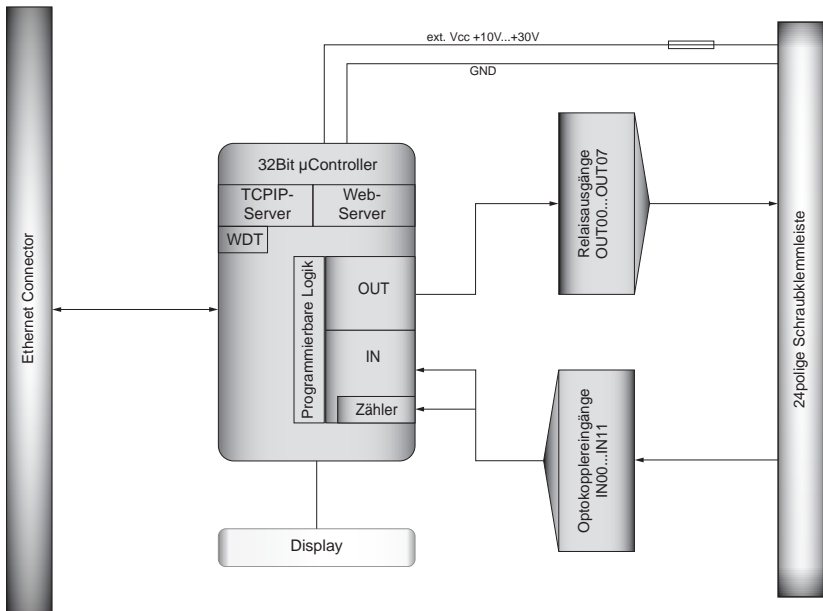
Anschlussklemme für externe Versorgungsspannung

GND_EXT:

Masse-Anschluss

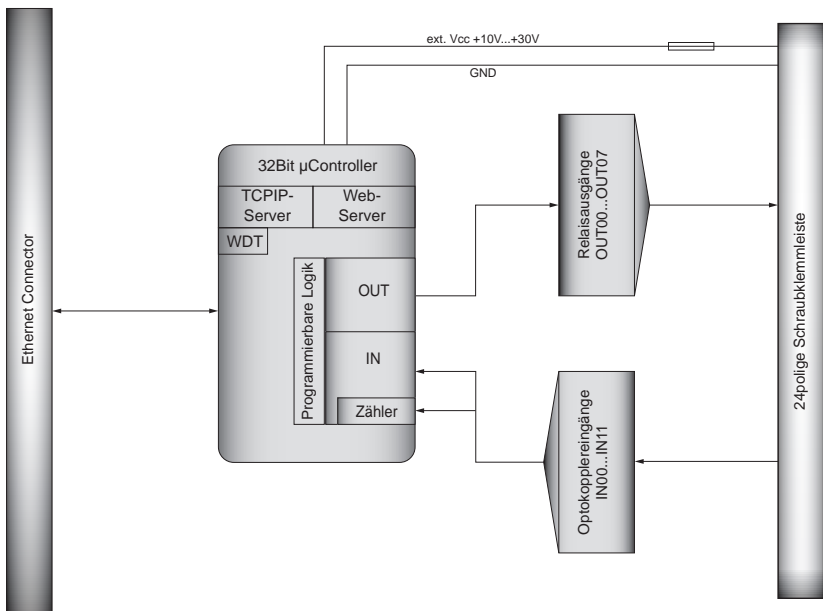
3. Systemkomponenten

3.1 Blockschahtbild EXDUL-537E



Grafik 3.1 Blockschahtbild EXDUL-537E

3.2 Blockschaltbild EXDUL-537S



Grafik 3.2 Blockschaltbild EXDUL-537S

3.3 Digitale Eingänge über Optokoppler

12 bipolare Kanäle mit galvanischer Trennung

Überspannungsschutz-Dioden

Eingangsspannungsbereich

high = 10..30 Volt

low = 0..3 Volt

3.4 Digitale Ausgänge über Relais

8 Kanäle mit galvanischer Trennung über Relais

Kontakt A des Relais mit jeweils einer Anschlussklemme, Kontakt B des Relais mit gemeinsamer Anschlussklemme aller 8 Kanäle

Kontakt: 1 Schließer

Schaltstrom: max. 2A / Kanal

Schaltspannung: max. 30V AC / 30V DC

Schaltleistung: max 60VA / 60W

Mechanische Lebensdauer: mind. $10 * 10^6$ Schaltspiele ohne Last

Kontakt Lebensdauer: 2A, 30VAC am Wechsler ca $2 * 10^5$ Schaltspiele

Schalhäufigkeit mit Last: typ. 10/min

Schalhäufigkeit ohne Last: typ. 300/min

3.5 Zähler

6 programmierbare Zähler 32 Bit (belegen 6 Optokoppler-Eingänge)

Zählfrequenz: max. 5 kHz

Automatische Sicherung der Zählerstände im 10kHz Takt

3.6 Programmierbare Logik

Ausgänge über Verknüpfungen schalten

Meldung an PC bei Zustandsänderung an den Eingängen senden

3.7 TCP/IP-Server

Stabile und einfache Kommunikation mit einfachem Protokoll

3.8 Webserver

Konfiguration

Funktionstest

Update der Firmware

3.9 Kommunikations-Watchdog-Timer

Einstellbares Zeitfenster von 1 bis 2^{32} * 1 ms

3.10 LCD Anzeige (nur EXDUL-537E)

Matrixanzeige mit 2 Zeilen und 16 Spalten zur Darstellung von 16 Zeichen je Zeile

Programmierbar zur Darstellung anwendungsspezifischer Daten oder als I/O-Zustandsanzeige

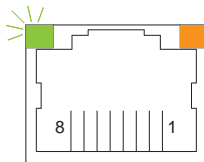
4. Inbetriebnahme

Der Netzwerk- bzw. PC-Anschluss erfolgt einfach und unkompliziert über eine Ethernet-Schnittstelle, die Konfiguration über einen Webbrowser. Für die Versorgung mit der notwendigen Betriebsspannung ist ein externes Netzteil notwendig.

4.1 Anschluss an einen Ethernet-Port

Das EXDUL-537E/EXDUL-537S verfügt über ein 10/100 Fast Ethernet Interface mit RJ45 Anschluss und wird über ein Netzwerkkabel direkt an einen PC, einen Ethernet-Hub oder an einen Ethernet-Switch angeschlossen.

Das Modul bootet nach Anlegen der Betriebsspannung, sobald eine stabile Ethernet-Verbindung besteht, leuchtet die linke LED an der RJ45-Buchse des EXDUL-537 kontinuierlich grün.



4.2 Anschluss der Betriebsspannung

Das EXDUL-537E / EXDUL-537S benötigt für den Betrieb eine Spannungsversorgung von +10 V ... +30 V DC an Klemme 23 (Vcc) und Klemme 24 (GND).

4.3 Integrierte Webpage des EXDUL-537E / EXDUL-537S

Der Zugriff auf die Webpage des EXDUL-537 ist über eine TCP/IP-Verbindung durch einen beliebigen Web-/Internetbrowser wie Mozilla® Firefox®, Chrome®, Edge®, Safari® usw. möglich. Über die Webpage besteht die Möglichkeit, Verbindungsinformationen auszulesen und passwortgeschützt Konfigurationsdaten zu verändern. Vorgenommene Einstellungen werden im internen Speicher des EXDUL-537 gespeichert und beim Booten geladen. Die EXDUL-537-Webpage ermöglicht zudem das Beschreiben, Auslesen und Anzeigen der Anwender-Speicherbereiche UserA, UserB, UserLCD1m und UserLCD2m sowie das Starten und Stoppen des Zählers. Zudem ist das Testen der digitalen Ein- und Ausgänge möglich.

4.4 Passwortschutz - Zugangskennung

Wie bereits aufgeführt, ist über die EXDUL-Webpage die Netzwerk-Konfiguration, die Einstellung der LCD-Anzeige, das Beschreiben der User-Speicherbereiche sowie das Setzen der Ein- und Ausgänge und des Zählers möglich. Um unbefugten Zugriff zu vermeiden, sind diese Einstellungsbereiche durch ein Kennwort geschützt.

Werksmäßig ist folgende Zugangskennung voreingestellt:

Benutzerkennung: admin
Kennwort: 11111111

Groß-/Kleinschreibung beachten!

Falls über diese Zugangsdaten kein Zugriff möglich ist, wurde die Einstellung der Zugangskennung von Ihrem Systemadministrator geändert.

4.5 Grundeinstellung Netzwerk-Konfiguration

In der Grundeinstellung ist das EXDUL-537 auf DHCP (Dynamic Host Configuration Protocol), d.h. auf dynamische IP-Adresse eingestellt. Bei der Inbetriebnahme schickt das EXDUL-537 eine Nachfrage ins Netzwerk (LAN). Im Netzwerk, aktiver DHCP-Dienst vorausgesetzt, wird dem Modul daraufhin automatisch eine IP-Adresse zugewiesen. Diese Einstellung ermöglicht Ihnen einfach und unkompliziert den Anschluss des Moduls zum Anpassen der Konfigurationsdaten nach Ihren Bedürfnissen.

4.6 Zusammensetzung und Aufbau der IP-Adresse

IP4-Adressen bestehen aus 32 Bits = 4 Bytes (Oktetten). Jedes Byte kann einen Wert zwischen 0 und 255 annehmen. Die Darstellung erfolgt als vier Dezimalzahlen durch Punkte getrennt (z.B. 192.168.1.83).

Jede IP-Adresse enthält einen Netzwerk- und Geräteanteil (Hostanteil). Über die Subnetzmaske erfolgt die Trennung zwischen Netz- und Hostteil. Alle Geräte, die sich im gleichen Netzwerkteil befinden, können miteinander kommunizieren.

Beispiel:

Ist der IP-Adresse 192.168.1.83 die Subnetzmaske 255.255.0.0 zugeordnet, so befindet sich das Gerät im Netz 192.168.-.- als Gerät -.-.1.83.

4.7 Änderung der Netzwerk-Konfiguration

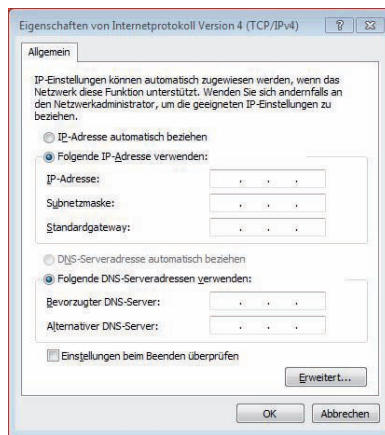
Zum Ändern der werksmäßigen Konfigurationseinstellungen schließen Sie das EXDUL-537 über die RJ45-Buchse und das beiliegende Standard-Netzwerkkabel an ein lokales Netzwerk an. Zudem ist eine Spannungsversorgung (+10 V ...+30 V) an Klemme 23 (Vcc) und Klemme 24 (GND) des Moduls nötig. Das EXDUL bootet jetzt, die linke LED an der RJ45-Buchse des EXDUL-537 leuchtet kontinuierlich grün sobald eine stabile Ethernet-Verbindung besteht. Über einen beliebigen Web-/Internetbrowser ist durch Eingabe des Host-Namens **http://EXDUL-537** der Zugriff auf die EXDUL-537-Webpage möglich. Durch Anklicken des Buttons **TCP/IP Config** öffnet sich folgende Konfigurationsmaske:

The screenshot shows the EXDUL-537 web interface. At the top, there is a navigation bar with 'Home', 'Status', 'Info', 'Config', and 'Peripherie'. The main heading is 'TCP/IP Konfiguration'. Below this, a note states: 'Diese Seite dient zur Einstellung und Änderung der Netzwerkparameter.' A red warning box contains the text: 'Achtung: Beachten Sie dringend die Hinweise im Handbuch, durch unkorrekte Einstellungen geht die Netzwerkkonnektivität verloren.' The configuration form includes the following fields: 'MAC Address' (04:b4:3a:00:00:00), 'Host Name' (EXDUL-537), 'Enable DHCP' (checked), 'IP Address' (192.168.180.4), 'Gateway' (192.168.180.1), 'Subnet Mask' (255.255.255.0), 'Primary DNS' (192.168.180.1), and 'Secondary DNS' (0.0.0.0). A 'Daten speichern' button is located at the bottom of the form. The footer of the page reads 'EXDUL Webpage Copyright © 2021'.

Über das beiliegende Standard-Netzwerkkabel kann das EXDUL-537 auch direkt an einen PC mit Ethernet-Schnittstelle angeschlossen werden. Beim Direktanschluß steht im Normalfall kein DHCP-Dienst zur Verfügung, da weder der PC noch das EXDUL diesen bietet. Für diesen Fall ist das EXDUL-537 über die zuletzt eingestellte statische IP-Adresse ansprechbar. Für EXDUL-Neugeräte ist die IP-Adresse 169.254.1.1 gültig.

Wichtiger Hinweis:

Zum Kommunizieren mit dem angeschlossenen EXDUL muss sich der verwendete PC im gleichen Netzwerk befinden. Dazu muss beim Rechner DHCP deaktiviert und z.B. von der EXDUL-IP-Adresse 192.168.1.83 (Subnetzmaske 255.255.0.0) die ersten zwei Zahlenblöcke (192.168) in die IP-Adresse des Rechners übernommen werden. Die zwei weiteren Blöcke können, jeweils Ihren Bedürfnissen angepasst, mit Werten zwischen 0 und 255 belegt werden.



Der Netzanteil der IP-Adresse ist von der Subnetzmaske abhängig. Von der IP-Adresse des EXDUL-537 müssen alle Oktette, die in der Subnetzmaske mit 255 belegt sind, in die IP-Adresse des Rechners übernommen werden (siehe auch Kapitel 4.6 Zusammensetzung und Aufbau der IP-Adresse).

4.8 Konfiguration mit statischer IP-Adresse (DHCP deaktiviert)

Für die Konfiguration des EXDUL-537 auf die statische IP-Adresse muss DHCP in der EXDUL-537-Konfigurationsmaske deaktiviert werden. Für die Konfigurationsänderung schließen Sie das EXDUL-537 wie unter Kapitel 4.7 (Änderung der Netzwerk-Konfiguration) an ein LAN oder einen PC an.

Sobald eine stabile Ethernet-Verbindung besteht, haben Sie über einen beliebigen Internetbrowser Zugriff auf die Webpage des EXDUL-537. Durch Eingabe des Host-Namens **http://EXDUL-537** sollte der Browser die EXDUL-537-Webpage öffnen, das Anklicken des Buttons TCP/IP Config öffnet folgende Konfigurationsmaske:

The screenshot shows the 'TCP/IP Konfiguration' web page for the EXDUL-537 device. At the top, there is a navigation bar with 'Home', 'Status', 'Info', 'Config', and 'Peripherie'. Below the title, a note states: 'Diese Seite dient zur Einstellung und Änderung der Netzwerkparameter.' A red warning box contains the text: 'Achtung: Beachten Sie dringend die Hinweise im Handbuch, durch unkorrekte Einstellungen geht die Netzwerkkonnektivität verloren.' The main configuration area includes fields for MAC Address (04:24:3e:00:00:00), Host Name (EXDUL-537), IP Address (192.168.100.4), Gateway (192.168.100.1), Subnet Mask (255.255.255.0), Primary DNS (192.168.100.1), and Secondary DNS (0.0.0.0). There is an 'Enable DHCP' checkbox which is currently unchecked. A 'Daten speichern' button is located at the bottom of the form. The footer of the page reads 'EXDUL Webpage Copyright © 2021'.

Sobald Sie DHCP deaktivieren, können Sie Ihre gewünschte IP-Adresse, Subnetzmaske und den gewünschten Host-Namen eintragen. Durch Anklicken des Buttons **Konfiguration speichern** werden die aktuell eingetragenen Daten in den internen Speicher des EXDUL-537 übernommen. Das Modul ist ab diesem Zeitpunkt nur über die hier eingestellte IP-Adresse bzw. über den angegebenen Host-Namen ansprechbar. Der verwendete PC oder das LAN muss sich dazu im gleichen Netz befinden.

Wichtiger Hinweis: Jede IP-Adresse und jeder Host-Name darf nur einem Gerät bzw. Modul in einem Netzwerk zugeteilt werden, eine Doppelvergabe ist nicht zulässig! Der Host-Name darf beliebig gewählt werden, jedoch nur aus den ASCII-Zeichen 0-9 sowie A-Z (Groß/Kleinschreibung egal) und dem - (Bindestrich) bestehen. Bestimmte IP-Adressen sind reserviert bzw. haben eine Sonderfunktion wie z.B.: 127.0.0.1 (local Host) 192.168.1.0 (0 ist Adresse des Netzes) mit 255.255.255.0 (Subnetzmaske)

Bitte erkundigen Sie sich bei Ihrem Netzwerkadministrator, welche IP-Adresse Sie verwenden dürfen. Bei Verwendung unzulässiger IP-Adressen besteht die Möglichkeit, dass der Zugriff auf das Modul nicht mehr möglich ist. Die Einstellung der wichtigsten nicht zulässigen Adressen wird vom Modul blockiert.

4.9 Konfiguration mit dynamischer IP-Adresse (DHCP aktiviert)

Falls Sie das EXDUL-537 in ein bestehendes Netzwerk mit bereits aktiven DHCP-Server einbinden und über eine dynamische IP-Adresse ansprechen möchten, muss DHCP (Dynamic Host Configuration Protocol) in der Konfigurationsmaske der Webpage aktiviert sein. In der werksmäßigen Grundeinstellung ist DHCP bereits aktiviert und keine Umstellung notwendig.

Bei eingestellter statischer IP-Adresse wird DHCP wie folgt aktiviert: Schließen Sie dazu das EXDUL-537 über ein Netzkabel (bei älteren PCs über ein Crossover-Netzkabel) an einen Computer an. Der verwendete PC muss auf „**Folgende IP-Adresse verwenden: (DHCP-deaktiviert)**“ eingestellt sein. In die IP-Adresse des Rechners muss der Netzanteil (siehe dazu Kapitel 4.6, Zusammensetzung und Aufbau der IP-Adresse) aus der IP-Adresse des EXDUL-537 übernommen werden, denn der Computer und das EXDUL müssen sich im gleichen Netzwerk befinden. An den Klemmen 23 (Vcc) und 24 (GND) schließen Sie die Spannungsversorgung (+10 V ...+30 V) für das EXDUL-537 an. Das Modul bootet jetzt, die linke LED an der RJ45-Buchse des EXDUL-537 leuchtet kontinuierlich grün, sobald eine stabile Ethernet-Verbindung besteht. Den Zugriff auf die Webpage des EXDUL-537 ermöglicht Ihnen ein beliebiger Internetbrowser durch Eingabe des Host-Namens oder der IP-Adresse des EXDUL-537. Jetzt sollte der Browser die EXDUL-537-Webpage öffnen. Durch Anklicken des Buttons **TCP/IP Config** öffnet sich die Konfigurationsmaske, in der Sie wie in nachfolgender Abbildung **Enable DHCP** aktivieren:

EXDUL

Home Status Info Config Peripherie

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:

Host Name:

Enable DHCP

IP Address:

Gateway:

Subnet Mask:

Primary DNS:

Secondary DNS:

EXDUL Webpage Copyright © 2021

Wichtiger Hinweis:

Bevor Sie DHCP aktivieren, ist eine Absprache mit Ihrem Netzwerkadministrator dringend notwendig. Sollten Sie mehrere gleiche Module aus der EXDUL-5xx-Serie in ein Netzwerk einbinden, müssen Sie den voreingestellten Host-Namen der einzelnen Module ändern. Jeder Host-Name darf innerhalb eines Netzwerkes nur einem Gerät bzw. Modul zugewiesen werden. Der Host-Name kann beliebig gewählt werden, jedoch nur aus den ASCII-Zeichen 0-9 sowie A-Z (Groß/Kleinschreibung egal) und dem - (Bindestrich) bestehen.

4.10 LCD-Anzeige während des Bootvorgangs (nur EXDUL-537E)

Während des Bootvorgangs des Moduls erscheint im Display eine Infoanzeige. In Zeile 1 wird der Modul-Name angezeigt, in Zeile 2 die Information, dass das Modul initialisiert wird. Sobald der Bootvorgang abgeschlossen ist, erscheint in beiden Fällen, je nach Einstellung, entweder die I/O-Statusanzeige oder die UserLCD-Anzeige.

4.11 LCD-Anzeige während des Betriebs (nur EXDUL-537E)

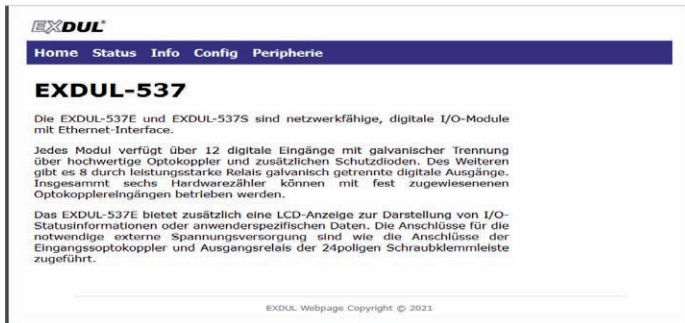
Nach dem Booten schaltet das Display, je nach Einstellung, von der Infoanzeige in die I/O-Statusanzeige oder UserLCD-Anzeige. Während der I/O-Statusanzeige werden in Zeile1 die aktuellen Zustände der Eingänge, in Zeile 2 die Zustände der Ausgänge angezeigt. Falls in der EXDUL-537 Webpage der UserLCD-Modus aktiviert ist, erscheint anstelle der I/O-Statusanzeige die UserLCD-Anzeige mit den Werten aus den Speicherbereichen UserLCD1m und UserLCD2m. Die Daten aus UserLCD1m und UserLCD2m werden solange auf dem LCD-Display angezeigt, bis neue Benutzerdaten über UserLCD-Zeile1 und UserLCD-Zeile2 auf die LCD-Anzeige geschrieben werden. Um einen „Screen-Burn“ zu vermeiden, wechselt die Anzeige im laufenden Betrieb etwa alle 30s ca. fünf Sekunden von der I/O-Statusanzeige oder UserLCD-Anzeige in die Infoanzeige mit der aktuellen IP-Adresse.

5. Zugriff auf das EXDUL-537

Der Zugriff auf die Konfigurationseinstellungen und auf die Ein-/Ausgänge des EXDUL-537 ist wie bereits erwähnt über die EXDUL-537 Webpage sowie über TCP/IP-Sockets möglich. Dazu wird die IP-Adresse, der Host-Name oder die MAC-Adresse benötigt.

5.1 Zugriff über die EXDUL Webpage

Die Webpage des EXDUL-537 ermöglicht es, die Eingänge zu lesen, die Ausgänge zu setzen, die Anwender-Speicherbereiche UserA, UserB und UserLCD sowie die Verbindungs- oder Modulinformationen auszulesen und die Konfigurationsdaten zu verändern. Der Zugriff auf die Webpage ist von jedem mit dem Modul verbundenen Computer über einen beliebigen Internetbrowser möglich. Der verwendete PC muss auf **IP-Adresse automatisch beziehen (DHCP-aktiviert)** eingestellt sein, soweit sich das EXDUL-537 noch im Auslieferungszustand (DHCP aktiviert) befindet und in ein Netzwerk mit aktiven DHCP-Dienst integriert ist. Durch Eingabe des Host-Namens (Im Auslieferungszustand **http://EXDUL-537**, ansonsten den von Ihnen eingestellten Host-Namen, evtl. über ExdulUtility_v2_xx oder spätere Version feststellen) oder der IP-Adresse können Sie die Webpage öffnen. Falls ein Öffnen der EXDUL-537-Webpage nicht möglich ist, überprüfen Sie die Netzwerkverbindung oder den eingegebenen Host-Namen. Nähere Informationen finden Sie im Kapitel FAQ - Problembehandlung.

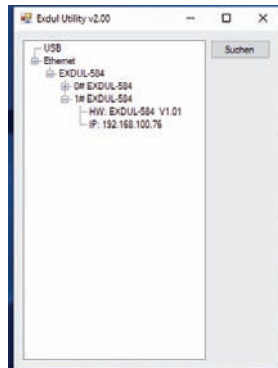


5.2 Zugriff über TCP/IP-Sockets

Mit der Verwendung des TCP-Protokolls wird eine zuverlässige Verbindung zwischen PC und dem EXDUL-537 erreicht. Das Protokoll ergreift selbstständig Maßnahmen bei Datenverlust. Die Adressierung des Moduls findet über eine 4 Byte IP-Adresse (IPv4) bzw. über den vergebenen Host-Namen und einer Portnummer 9760 statt. Der PC versendet über die Verbindung für jeden Befehl ein Byte-Array. Das Modul verarbeitet den Befehl und sendet immer eine Rückantwort. In Kombination mit einer Hochsprache ist über die TCP/IP-Verbindung das Lesen der Eingänge, das Setzen der Ausgänge, das Starten, Stoppen und Auslesen des Zählers, das Beschreiben der User-Speicherbereiche, das Auslesen der Verbindungs- und Modulinformationen sowie das Verändern der Konfigurationsdaten möglich. Insgesamt können 3 TCP/IP-Verbindungen gleichzeitig mit dem Modul geöffnet sein.

5.3 Host-Namen, IP-Adresse und MAC-Adresse feststellen

Falls Sie für EXDUL-5xx-Module weder den Host-Namen noch die IP-Adresse oder MAC-Adresse kennen, ermöglicht Ihnen das Suchprogramm ExdulUtility_v2_xx (oder höher) diese festzustellen. Falls Ihre Firewall die Kommunikation des Suchprogrammes mit den EXDUL-5xx verhindert, ist eine Freigabe für das Programm in der Firewall erforderlich.



Das Suchprogramm **ExdulUtility_v2_xx** (oder höher) befindet sich auf der EXDUL-Software-CD oder steht auf www.messcomp.com im Bereich Support - Software Support zum Download bereitgestellt.

6. 12 Optokopplereingänge

Das EXDUL-537 verfügt über 12 Eingangskanäle, deren galvanische Trennung mittels Optokoppler erreicht wird. Die Isolationsspannung des Optokopplers zwischen Masse des Moduls und Eingang beträgt 500 Volt, während die Spannung zwischen den Eingangskanälen auf 50 Volt begrenzt ist.

6.1 Pinbelegung der Eingangsoptokoppler

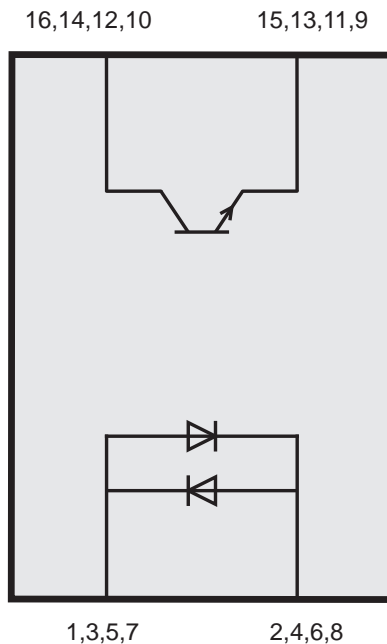


Abb. 6.1 Pinning Eingangsoptokoppler

6.2 Eingangsbeschaltung

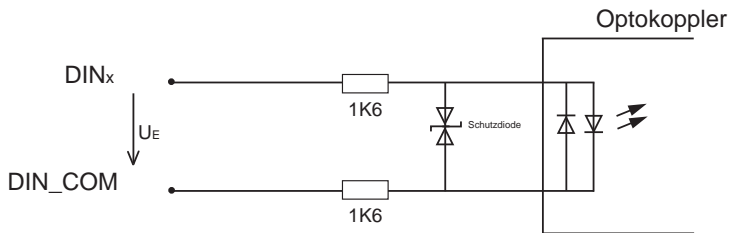


Abb. 6.2 Eingangsbeschaltung

Die Eingänge des Optokopplers sind bipolar ausgeführt. Im Normalfall wird der DIN_COM-Anschluß auf Minus gelegt und am DINx-Anschluss eine Spannung angelegt. Sie können jedoch auch, falls es schaltungs-technisch sinnvoller ist, am DIN_COM-Anschluß die Plus-Spannung und am DINx-Anschluß die Minus-Spannung anlegen. Bitte beachten Sie, dass diese Festlegung aufgrund des gemeinsamen DIN_COM-Anschlusses sämtlicher Eingangsoptokoppler für alle Eingänge gilt.

6.3 Eingangsstrom

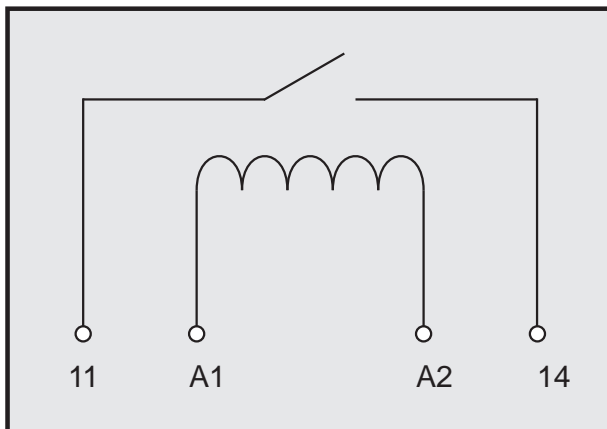
$$I_E \approx \frac{U_E - 1,1V}{3200\Omega}$$

Bei einer Eingangsspannung zwischen DINx und DIN_COM von 24 Volt ergibt sich ein Eingangsstrom von ca. 7mA, bei 12V von ca. 3,4 mA.

7. 8 Relaisausgänge

Das EXDUL-Modul verfügt über 8 Ausgangskanäle, deren galvanische Trennung mittels Relais erreicht werden.

7.1 Pinbelegung der Ausgangsrelais



7.2 Relaisdaten

| | |
|--------------------|------------------------|
| Schaltspannung: | max. 30V AC / 30V DC |
| Schaltstrom: | max. 2A |
| Schaltleistung: | max. 60VA / 60W |
| Kontaktwiderstand: | typ. 30Ω (6VDC, 100mA) |

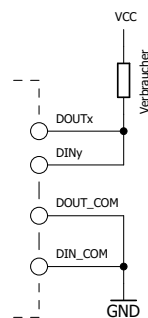
7.3 Programmierung der Relaisausgänge

Für die Programmierung der Relaisausgänge stehen mehrere Zugriffsfunktionen zur Verfügung. So können sowohl alle Kanäle auf einmal angesteuert werden als auch einzeln beschrieben werden. Sollte der aktuelle Zustand der Ausgänge im Anwenderprogramm nicht speicherbar sein, so kann dieser über einen Lese-Befehl rückgelesen werden. Hier ist zu beachten, dass der gelesene Schaltzustand nur dem primären Zustand (auf Prozessorseite) entspricht. Soll der tatsächliche Schaltzustand bzw. der Pegel des Ausgangs rückgelesen werden, muss dafür ein Optokopplereingang verwendet werden (siehe Kapitel Relaisausgänge rücklesen).

Eine detaillierte Beschreibung der Programmierung ist im Kapitel Programmierung zu finden.

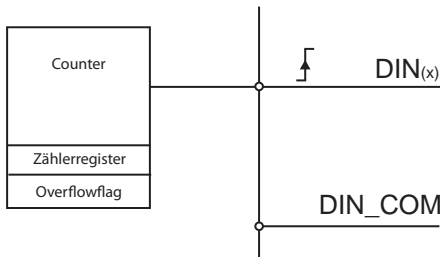
7.4 Relaisausgänge rücklesen

Gelegentlich ist es bei Applikationen nötig, den tatsächlichen Ausgangszustand in dem Programm zu wissen. Ein Beispiel kann z. B. eine Fehlererkennung im Programm sein. Dies kann durch die Rückführung des Relaisausgangs zu einem der Optokopplereingänge erfolgen. Im folgenden Beschaltungsbeispiel wird der Ausgang x mit dem Eingang y verbunden. Beachten Sie bitte, daß in dieser Schaltungsversion das Ergebnis des Eingangs negiert ist. Wenn der Ausgang geschaltet ist, liegt am Transistor keine Spannung an und somit zeigt der Eingang eine „0“ an. Zudem fließt bei nicht geschaltetem Ausgang über den Optokoppler-Eingang und damit auch über den Verbraucher ein geringer Strom (bei 24V ca. 7mA).



8. Zähler

Das Modul stellt an den ersten 6 Optokopplereingängen (DIN0 bis DIN5) jeweils einen eigenständigen, hardwareunterstützten 32bit Zähler zur Verfügung. Bei Bedarf können diese einzeln aktiviert werden und reagieren bei jeder steigenden Flanke durch die Inkrementierung des Zählerstandes. Ein abrufbares Flag signalisiert jeweils einen Overflow. Damit bei einer außerplanmäßigem Unterbrechung der Spannungsversorgung die Zählerstände nicht verloren gehen, werden diese ungefähr alle 100µs gesichert. Liegt dann am Modul wieder eine Spannungsversorgung an, werden die gesicherten Werte automatisch in die Zählerregister geladen.



| Funktionen | Beschreibung |
|--------------------|--|
| Start | Startet den Zähler bzw. gibt den Eingang frei |
| Stop | Stoppt den Zähler, Signal an dem jeweiligen Eingang werden ignoriert |
| Reset | Setzt den Zählerstand auf 0 |
| Zählerstand lesen | Liest den aktuellen Zählerstand |
| Overflowflag lesen | Liest das Overflowflag |
| Clear Overflowflag | Setzt das Overflowflag zurück |

9. Programmierbare Logik

In verschiedenen Anwendungen kann es von Vorteil sein, wenn das Modul selbständig auf Eingangssignale bzw. Änderungen an den Eingängen reagiert. Ohne diese Möglichkeit kann eine Änderung am Eingang nur durch regelmäßige Abfrage der Eingänge mittels Polling erkannt werden. Dieses Polling verursacht eine erhöhte Auslastung des Netzwerkes und des Rechners.

Um diese eigenständige Reaktion des Moduls zu erzeugen, werden vier Logikzweige mit jeweils vier Logikeingängen, einer Verknüpfung sowie einem Logikausgang zur Verfügung gestellt. In Abbildung 9.1 ist einer dieser Zweige zu sehen.

Den Logikeingängen bzw. Ausgängen können einzelne Funktionen wie später näher erläutert zugewiesen werden.

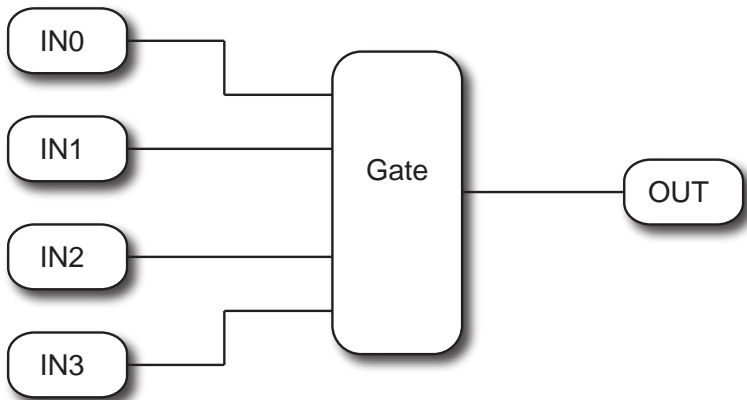


Abb. 9.1

9.1 Logikzweig

Im Modul können vier verschiedene Logikzweige verwendet werden. Jeder Zweig hat vier Eingänge, eine Logikverknüpfung (UND, ODER) und einen Ausgang. Den Logikeingängen sowie Logikausgängen können verschiedene Funktionen zugewiesen werden. Dabei kann die Logikverknüpfung eventgesteuert sein (z.B. Flanke an den Eingängen invertiert Optokopplerausgang oder sendet Nachricht an PC) als auch zustandsgesteuert (z.B. wenn Eingang DIN0 und DIN1 gleich HIGH dann ist DOUT0 ebenfalls HIGH). Bei der Auswahl der Funktionen ist darauf zu achten, dass bei einem Eventausgang (z.B. Message an PC oder Optokopplerausgang DOUT1 setzen) immer ein Eventeingang im Logikzweig vorhanden ist. Ist einem Logikausgang die Funktion NONE zugewiesen, so ist dieser Zweig automatisch deaktiviert und schont damit die Prozessorressourcen.

9.2 Logikeingänge

Jedem Logikzweig stehen vier Logikeingänge zur Verfügung. Diesen können folgende Funktionen zugewiesen werden.

| Eingangsfunktion | Beschreibung | zustand/event |
|------------------|---|---------------|
| TRUE | Eingang liefert immer eine 1 | zustand |
| FALSE | Eingang liefert immer eine 0 | zustand |
| DINxx | Logikeingang wird mit dem jeweiligen Optokopplereingang verknüpft | zustand |
| DINxx_EDGE | Logikeingang erkennt eine steigende Flanke und liefert für einen Zweigtakt (~10ms) eine 1 | event |

Als Ausgangsbasis für die Logikeingänge werden die Optokopplereingänge in regelmäßigen Zeitabständen (ca. 1ms) abgetastet. Hierbei werden Logikpegel oder Pegeländerungen erfaßt und an die programmierbare Logik übergeben.

9.3 Logikverknüpfung

Jedem Logikzweig steht eine Logikverknüpfung zur Verfügung. Dieser können folgende Funktionen zugewiesen werden.

| Ausgangsfunktion | Beschreibung | zustand/event |
|------------------|------------------|---------------|
| AND | UND-Verknüpfung | zustand/event |
| OR | ODER-Verknüpfung | zustand/event |

9.4 Logikausgänge

Jedem Logikzweig steht ein Logikausgang zur Verfügung. Diesem können folgende Funktionen zugewiesen werden.

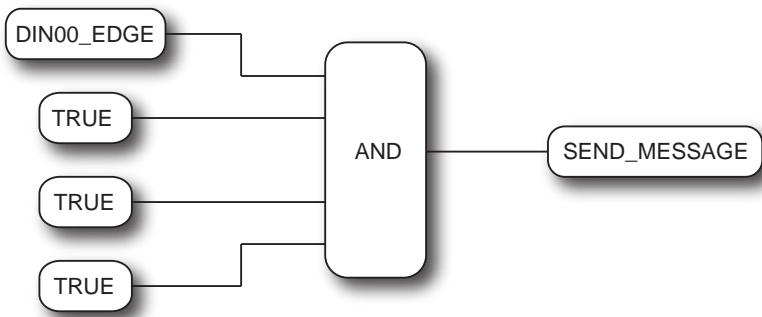
| Ausgangsfunktion | Beschreibung | zustand/event |
|---------------------------|---|---------------|
| NONE | Logikzweig ist deaktiviert | zustand |
| SEND_MESSAGE _x | Modul sendet eine Messages(x) an den PC | event |
| WRITE_DOUT _{xx} | Optokopplerausgang wird mit dem Ergebnis der Logikverknüpfung beschrieben | zustand |
| ,SET_DOUT _{xx} | Bei einer 1 der Verknüpfung wird der jeweilige Optokopplerausgang durchgeschaltet | event |
| CLEAR_DOUT _{xx} | Bei einer 1 der Verknüpfung wird der jeweilige Optokopplerausgang gesperrt | event |
| TOGGLE_DOUT _{xx} | Bei einer 1 der Verknüpfung wird der jeweilige Optokopplerausgang invertiert | event |

9.5 Message an PC schicken

Möchten Sie bei einem Event eine Benachrichtigung an den PC bzw. an die Anwendung senden, geht dies über die Message-Ausgänge. Es gibt insgesamt 4 Message-Funktionsausgänge, von welchen jeder eine eigene individuelle Nachricht an den PC schickt.

Für diese Funktion muss in dem Programm eine weitere TCP/IP-Verbindung mit dem Modul im Receiver-Modus aufgebaut werden. Liegt nun z.B. eine steigende Flanke an einem der zu überwachenden Eingänge der programmierbaren Logik an, wird automatisch eine Message an den Receiver gesendet. Diese Message muss nicht mit einer Rückantwort bestätigt werden.

Folgendes Beispiel veranschaulicht die Konfiguration.



In diesem Beispiel soll bei jeder steigenden Flanke an DIN00 eine Nachricht an den PC gesendet werden. Dafür wird für den Eingangsfunktionsblock IN0 die Funktion DIN00_EDGE ausgewählt und für alle anderen TRUE. Als Gater wird eine UND-Verknüpfung (AND) und als Ausgang SEND_MESSAGE1 gewählt. Liegt nun eine steigende Flanke an DIN00 an, so wird an den TCP/IP Client mit der Receiver-Verbindung ein 12 Byte langes Array gesendet.

Damit der PC unterscheiden kann, von welchem Logikblock die Meldung erfolgt, stehen 4 Message-Arrays zur Verfügung. In folgender Tabelle werden die 4 möglichen Byte-Arrays dargestellt. Die ersten 4 Byte stellen den Befehl, in diesem Fall immer 0x0E, sowie das Längenbyte (immer 2) dar. Byte 4 bis Byte 6 sind reserviert und haben den Wert 0x00. In Byte 7 steht der Message-Index. Wurde als Ausgang die Message1 ausgewählt steht in diesem Byte der Wert 1. Zur Kontrolle der Reihenfolge der gesendeten Meldungen, bzw. um auch das Fehlen einer Meldung erkennen zu können beinhaltet des Byte-Array zusätzlich einen Zähler. Byte 8 bis Byte 11 entsprechen dem Wert des Receivernachrichten-Zählers. Bei jeder Nachricht an den PC wird dieser Wert incrementiert.

Receivernachrichtenzähler:

$$\text{Wert} = \text{dd} * 0x100000 + \text{cc} * 0x10000 + \text{bb} * 0x100 + \text{aa}$$

| 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 Timing der programmierbaren Logik

| | |
|-------------------|---------------------------|
| | typ. Bearbeitungsfrequenz |
| Logikzweig-Update | 100 Hz |
| Input-Abtastung | 1 kHz |

10. Watchdog-Timer (WDT)

Das EXDUL-Modul besitzt einige Sicherheitsmechanismen für eine stabile Kommunikation. Sollte es jedoch z.B. Aufgrund von Routingproblemen zu einer Unterbrechung der Kommunikation kommen und ein Wiederaufbau der Verbindung nicht mehr möglich sein, kann der integrierte Watchdog-Timer des Moduls zu Problembeseitigung verwendet werden.

Zu Beginn der Applikation muss der WDT mit einer ausgewählten Periodendauer initialisiert und gestartet werden. Wird der Timer nicht innerhalb der eingestellten Periodendauer durch einen PC-Befehl zurückgesetzt, so wird ein vollständiger Reset des EXDUL-Moduls durchgeführt. Beim Auslösung des Watchdog-Resets wird das entsprechende Flag im Error-Register gesetzt. Hierdurch kann später durch das Überprüfen dieses Bits der Reset erkannt werden.

11. Error-Register

Kommt es zu außerplanmäßigen Fehlern (z.B. einem Watchdog-Reset), so werden diese in den beiden Error-Registern angezeigt. Tritt ein Fehler auf, so wird das dem Fehler zugewiesene Bit gesetzt. Die Fehlerregister bleiben auch nach einem Reset oder dem Entfernen der Spannungsversorgung erhalten. Die Register können mit dem entsprechenden Befehl zurückgesetzt werden.

| Error-Register 0 | Bit31 .. Bit2 | Bit1 | Bit0 |
|-------------------------|----------------------|-------------|-------------|
| Bedeutung | reserviert | WDT_SW | reserviert |
| Error-Register 1 | Bit31 .. Bit0 | | |
| Bedeutung | reserviert | | |

Beschreibung:

WDT_SW: 1 = Der Watchdog-Reset wurde ausgeführt.

12. Informations-, LCD- und Userregister

12.1 Register HW-Kennung und Seriennummer

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| HW-Kennung | E | X | D | U | L | - | 5 | 3 | 7 | | | V | 1 | . | 0 | 1 |
| | 45 _{hex} | 58 _{hex} | 44 _{hex} | 55 _{hex} | 4C _{hex} | 2D _{hex} | 35 _{hex} | 33 _{hex} | 37 _{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} | | | | | | | | | |

Tabelle 12.1 Register HW-Kennung und Seriennummer

Im Register HW-Kennung ist der Modulname sowie die Version der Firmware abgelegt und kann zur Feststellung der Produkt-Identität vom User gelesen werden. In der o. a. Tabelle sind als Beispiel in der Zeile HW-Kennung jeweils der Hex-Wert und das dazugehörige ASCII-Zeichen für das Modul EXDUL-537 mit Firmware-Version 1.01 dargestellt.

Das Register Serien-Nummer kann vom Anwender lediglich gelesen werden. Die Serien-Nummer in der o. a. Tabelle dient als Formatbeispiel. In der Zeile S/N ist jeweils der Hex-Wert und darüber das dazugehörige ASCII-Zeichen für die Serien-Nummer 1044026 dargestellt.

12.2 Speicherbereiche UserA, UserB, UserLCD1m* und UserLCD2m*

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

In den Registern UserA, UserB, UserLCD1m* und UserLCD2m* können jeweils 16 Stellen (16 Byte) zur eigenen Verwendung genutzt werden. Die Daten bleiben beim Ausschalten erhalten, ein Default-Reset setzt diese Register in die Werkseinstellung (Auslieferungszustand) zurück. Im Auslieferungszustand steht in allen vier User-Speicherbereichen an jeder Stelle der Hex-Wert 20, der im ASCII-Code einem Leer-Zeichen entspricht. In der o. a. Tabelle sind jeweils der Hex-Wert und darüber das dazugehörige ASCII-Zeichen dargestellt.

12.3 Display-Register UserLCD-Zeile1*, UserLCD-Zeile2* und LCD-Kontrast*

Die Register UserLCD-Zeile1 und UserLCD-Zeile2 dienen bei aktivierten UserLCD-Modus zum Beschreiben der beiden LCD-Zeilen mit jeweils 16 beliebigen Zeichen. Mit Übernahme der Daten erfolgt die Anzeige im Display anstelle der Daten aus UserLCD1m* und UserLCD2m*. Die Daten in den Registern UserLCD-Zeile1 und UserLCD-Zeile2 bleiben beim Ausschalten **nicht** erhalten. Über das Register LCD-Kontrast ist der Display-Kontrast einstellbar, der auch beim Ausschalten erhalten bleibt.

*: Nur für EXDUL-537E zutreffend, bei EXDUL-537S ohne Funktion!

13. Installation des Treibers

Für das Ethernetmodul EXDUL-537 ist kein Treiber notwendig. Voraussetzung ist eine bereitgestellte Netzwerkverbindung von PC (Netzwerkkarte mit Treiber) oder mobilen Device. Für den direkten Zugriff auf das Modul werden die bei vielen Hochsprachen wie C, C++, C#, Visual Basic[®], Python[®] oder Java[®] vorhandenen TCP/IP-Libraries benötigt. Beispiele in mehreren Programmiersprachen für den Zugriff befinden sich auf der mitgelieferten CD und auf unserer Website www.messcomp.com.

14. Programmierung

14.1 Einführung

Die Programmierung erfolgt mit Hilfe des Standard TCP/IP-Protokolls und somit über sogenannte TCP/IP-Sockets, für die in vielen gängigen Programmiersprachen Standardbibliotheken vorhanden sind. Eine einfache und schnelle Implementierung ermöglicht die Verwendung des .Net Frameworks von Microsoft. Durch das Verwenden des Standard-Protokolls kann das Modul neben Windows auch mit einer Vielzahl von anderen Betriebssystemen wie Ubuntu[®] (Linux-basierend) oder Android[®] verbunden werden. Verschiedene Programmierbeispiele sind auf der beiliegenden CD und auf unserer Website bereitgestellt.

14.2 Programmierarten

Für den Zugriff auf das EXDUL-Modul gibt es mehrere Möglichkeiten. So kann für die Programmierung unter Windows und .NET die Library EXDUL.dll verwendet werden. Diese ermöglicht einen leichten und schnellen Einstieg, um den Zugriff auf das Modul zu programmieren. Des Weiteren können auch TCP und Socket Libraries verwendet werden, welche bei vielen Programmiersprachen wie C# oder Java[®] vorhanden sind. Sie ermöglichen oft eine breite Einstellmöglichkeit der Schnittstelle. LabVIEW[®]-Anwender können ebenfalls mit Hilfe der EXDUL.dll oder den TCP-Funktionsblöcken leicht auf das Modul zugreifen.

14.3 Programmierung unter Windows mit der .NET EXDUL.dll Library

Wird für den Modul-Zugriff eine .NET-Programmiersprache verwendet (C#, C++.NET oder VB.NET), so kann die Library EXDUL.dll verwendet werden. Sie besitzt einen objektorientierten Aufbau, in welchem jedes EXDUL-Modul durch ein Objekt mit ihren Methoden dargestellt wird.

Bei der Entwicklung der Library wurde auf eine möglichst einheitliche API zwischen den unterschiedlichen EXDUL-Modulen geachtet.

Dies ermöglicht es dem Anwender, bei Bedarf ohne großen Programmieraufwand von z.B. einem USB-EXDUL-Modul auf ein Ethernet-EXDUL-Modul (z.B. EXDUL-337 -> EXDUL-537) zu wechseln.

Befehlsübersicht für EXDUL.dll-Library

Open:

[bool](#) Open()

Rückgabewerte: true wenn erfolgreich / false bei Fehler

Zusammenfassung: Verbindung zu Modul aufbauen

Close

void Close()

Zusammenfassung: Verbindung zu Modul schließen

Schreiben in Inforegister:

void SetModullInfo ([byte](#) type, [string](#) info)

Parameter: type: Info-Typ (siehe Handbuch)

info: Bis zu 16 Zeichen langer Info-String

Zusammenfassung: Beschreibt die Modul-Informationsregister

| Infobereich | Info-Byte |
|-------------|-----------|
| UserA | 0 |
| UserB | 1 |

Lesen aus Inforegister:

[string](#) GetModullInfo([byte](#) type)

Parameter: type: Info-Typ (siehe Handbuch)

Rückgabewerte: Gibt das Register "type" als string zurück

Zusammenfassung: Liest die Modul-Information-Register aus

| Infobereich | Info-Byte |
|-----------------|-----------|
| UserA | 0 |
| UserB | 1 |
| Hardwarekennung | 3 |
| Seriennummer | 4 |

Schreiben in LCD-Register UserLCD:

void SetUserLCD([byte](#) *line*, [string](#) *text*)

Parameter: *line*: 0 = 1. Zeile / 1 = 2. Zeile

text: Bis zu 16 Zeichen langer LCD-Text

Zusammenfassung: Beschreibt die UserLCD-Register. Der Parameter *line* legt die Zeile (0 oder 1) fest und *text* den Text aus 16 Zeichen.

Schreiben in LCD-Register UserLCDm:

void SetUserLCDm([byte](#) *line*, [string](#) *text*)

Parameter: *line*: 0 = 1. Zeile / 1 = 2. Zeile

text: Bis zu 16 Zeichen langer LCD-Text

Zusammenfassung: Beschreibt die UserLCDm-Register. Der Parameter *line* legt die Zeile (0 oder 1) fest und *text* den Text aus 16 Zeichen

Schreiben des LCD-Modes:

void SetLCDMode([byte](#) *mode*)

Parameter: *mode*: LCD-Modus

Zusammenfassung: Setzt den LCD-Modus fest

| LCD-Modus | LCD-Modus-Byte |
|-----------|----------------|
| IO-Mode | 0 |
| User-Mode | 1 |

Lesen des LCD-Modes:[byte](#) GetLCDMode()

Rückgabewerte: LCD-Modus

Zusammenfassung: Liest den LCD-Modus aus

| LCD-Modus | LCD-Modus-Byte |
|-----------|----------------|
| IO-Mode | 0 |
| User-Mode | 1 |

Schreiben LCD-Kontrastwert:void SetLCDContrast([ushort](#) contrast)Parameter: *contrast*: Wert zwischen 0 und 4095 (empfohlen
800 bis 1800)

Zusammenfassung: Legt den LCD-Kontrast fest

Lesen LCD-Kontrastwert:[ushort](#) GetLCDContrast()

Rückgabewerte: LCD-Kontrast

Zusammenfassung: Liest den LCD-Kontrast aus

Relaisausgänge lesen:

[uint](#) GetRelOut()

Rückgabewerte: Zustand der Relaisausgänge

Zusammenfassung: Liest den Zustand der Relaisausgänge

Relaisausgänge schreiben:

void SetRelOut([uint](#) value)

Parameter: *value*: Zustand der Ausgänge

Zusammenfassung: Setzt die Relaisausgänge

Relaisausgang schreiben:

void WriteRelOut(byte channel, [uint](#) value)

Parameter: *channel*: Index des Ausgangskanals

value: Zustand des Ausgangs

Zusammenfassung: Setzt einzelne Relaisausgänge

Optokopplereingänge lesen:

[uint](#) GetOptIn()

Rückgabewerte: Aktueller Zustand der Optokopplereingänge

Zusammenfassung: Liest den aktuellen Zustand an den Optokopplereingängen

Zähler starten:

void StartCounter([byte](#) index)

Parameter: *index*: Counter-Index

Zusammenfassung: Startet den Zähler mit der Nummer index

Zähler stoppen:

void StopCounter([byte](#) index)

Parameter: *index*: Counter-Index

Zusammenfassung: Stoppt den Zähler mit der Nummer index

Zähler resettten:

void ResetCounter([byte](#) index)

Parameter: *index*: Counter-Index

Zusammenfassung: Setzt den Zählerstand des Zählers mit der Nummer index zurück auf 0

Zählerstand lesen:

[uint](#) ReadCounter([byte](#) index)

Parameter: *index*: Counter-Index

Rückgabewerte: Zählerstand

Zusammenfassung: Liest den Zählerstand des Zählers mit der Nummer index aus

Overflow-Flag lesen:

[bool](#) ReadOverflowFlagCounter([byte](#) *index*)

Parameter: *index*: Counter-Index

Rückgabewerte: Overflowflag false = kein Overflow
true = Overflow

Zusammenfassung: Liest das Overflowflag des Zählers mit der Nummer *index* aus

Overflow-Flag rücksetzen:

void ResetOverflowFlagCounter([byte](#) *index*)

Parameter: *index*: Counter-Index

Zusammenfassung: Setzt das Overflowflag des Zählers mit der Nummer *index* zurück

Werksreset:

void DefaultReset()

Zusammenfassung: Setzt das Modul auf die Werkseinstellung zurück. Nach dem Befehl muss die Verbindung zum Modul geschlossen und wieder neu geöffnet werden

14.4 Programmierung mit TCP-Libraries

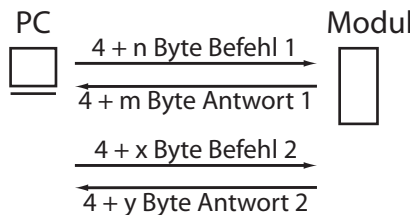
Durch die Möglichkeit, mit Standard TCP/Socket-Libraries auf das Modul zugreifen zu können, kann der Anwender mit einer Vielzahl an Sprachen seine Anwendung auf verschiedenen Plattformen programmieren. So kann unter Windows neben dem .NET-Framework auch Delphi[®] oder Java[®] verwendet werden. Auch können Anwendungen auf vielen Linux basierten Betriebssystemen mit z.B. C oder Java[®] entworfen werden. Dabei ist zu beachten, dass das Modul immer als Server fungiert.

14.4.1 Kommunikation mit dem EXDUL-537

Der Datenaustausch erfolgt durch Senden bzw. Empfangen von Byte-Arrays mit unterschiedlicher Länge über die TCP/IP-Schnittstelle.

Jeder erlaubte Sendestring wird mit einem definierten Ergebnis- bzw. Bestätigungsstring beantwortet.

Vor dem Senden eines Strings muss der letzte Ergebnis- bzw. Bestätigungsstring gelesen werden.



Grafik 14.4 Kommunikationsmodell

14.4.2 Befehls- und Datenformat

Der Datenaustausch erfolgt durch Senden und Empfangen von Byte-Arrays. Jedes zu sendende bzw. zu empfangende Byte-Array besteht aus mindestens 4 Bytes. Dabei stellen die ersten drei Bytes den Befehl und das vierte die Anzahl der noch folgenden 4 Byte-Blöcke dar.

| | | | |
|------------------|------------------|------------------|------------|
| Befehl Byte 0 | Befehl Byte 1 | Befehl Byte 2 | Längenbyte |
|------------------|------------------|------------------|------------|

Die Anzahl der 4-Byte-Blöcke variiert von Befehl zu Befehl und ist zum Teil von der zu sendenden Datenmenge abhängig. Genauere Informationen befinden sich bei den einzelnen Befehlsbeschreibungen.

14.4.3 Passwortschutz

Um das Modul vor unbefugten Zugriffen zu schützen, kann für den Datenaustausch ein einfacher Passwortschutz verwendet werden. Ist dieser aktiviert (siehe Befehl Securitykonfiguration), so muss an jedes gesendete Byte-Array das richtige Passwort (+8 Bytes) angehängt werden. Ist das Passwort falsch oder wurde es nicht dem Befehlsstring hinzugefügt, so wird eine Error-Antwort zurückgeschickt. Die Rückantwort vom Modul bleibt unverändert

Beispiel: Optokopplerausgänge schreiben mit aktiviertem Passwortschutz und dem Default-Passwort „11111111“

| Byte | Senden | Empfangen | Beschreibung |
|------|---|-----------|--|
| 0 | 08 | 08 | Befehlscode 1. Byte |
| 1 | 00 | 00 | Befehlscode 2. Byte |
| 2 | 00 | 00 | Befehlscode 3. Byte |
| 3 | 03 (→ 12Byte) | 00 | Längenbyte |
| 4 | 00 | | r/w Byte |
| 5 | 0w 00 (gesperrt) 01 (durchgeschaltet) | | Optokopplerzustand |
| 6 | 00 | | reserviert |
| 7 | 00 | | reserviert |
| 8 | 31 _{hex} | | Passwort 1. Zeichen 1 _{ascii} |
| 9 | 31 _{hex} | | Passwort 2. Zeichen 1 _{ascii} |
| 10 | 31 _{hex} | | Passwort 3. Zeichen 1 _{ascii} |
| 11 | 31 _{hex} | | Passwort 4. Zeichen 1 _{ascii} |
| 12 | 31 _{hex} | | Passwort 5. Zeichen 1 _{ascii} |
| 13 | 31 _{hex} | | Passwort 6. Zeichen 1 _{ascii} |
| 14 | 31 _{hex} | | Passwort 7. Zeichen 1 _{ascii} |
| 15 | 31 _{hex} | | Passwort 8. Zeichen 1 _{ascii} |

Der normale Befehl ohne Passwortschutz beinhaltet nur 8 Bytes und im Längenbyte steht der Wert 1. Wie in der Tabelle zu sehen, wurden nun 8 weitere Bytes, in welchen das Passwort steht, angehängt. Zusätzlich muss das Längenbyte um den Wert 2 (+8Byte) erhöht werden. Die Rückantwort beinhaltet kein Passwort und entspricht dem normalen Antwortarray ohne Passwort.

14.4.4 Befehlsübersicht

| Hexcode | Beschreibung |
|----------|---|
| 0C 00 00 | Inforegister lesen und schreiben |
| 0C 00 03 | LCD-Register lesen und schreiben |
| 0C 00 08 | Netzwerkkonfigurationen lesen und schreiben |
| 0C 00 0C | Securitykonfigurationen lesen und schreiben |
| 0C 00 0D | Passwort ändern |
| 08 00 00 | Relaisausgänge lesen und schreiben |
| 08 00 01 | Optokopplereingänge lesen |
| 09 00 00 | Zähler0 |
| 09 00 01 | Zähler1 |
| 09 00 02 | Zähler2 |
| 09 00 03 | Zähler3 |
| 09 00 04 | Zähler4 |
| 09 00 05 | Zähler5 |
| 0C 01 01 | Software Watch Dog Timer |
| 0C 02 10 | Initialisierung eines programmierbaren Logikzweiges |
| 0C 03 00 | Receivermodus aktivieren/deaktivieren + Messagezähler lesen |
| FF 00 00 | Fehlerregister auslesen/rücksetzen |

14.4.5 Befehlszusammensetzung

Schreiben in Inforegister

Das EXDUL-Modul stellt mehrere beschreibbare Inforegister zur Verfügung. UserA/B sind zwei 16-Byte-Bereiche für den Anwender, um Informationen in einem nicht-flüchtigen Speicher (FLASH) zu sichern. Die Register sind nur als ganzer 16-Byte-Block beschreibbar.

| Infobereich | Info-Byte |
|-------------|-----------|
| UserA | 0 |
| UserB | 1 |

Beispiel: Schreiben der Zeichenfolge EXDUL-537 in Register UserA und UserB

| Byte | Senden | Empfangen | Beschreibung |
|------|--------------------------|-----------|--|
| 0 | 0C | 0C | Befehlscode 1. Byte |
| 1 | 00 | 00 | Befehlscode 2. Byte |
| 2 | 00 | 00 | Befehlscode 3. Byte |
| 3 | 05 | 00 | Längenbyte → 20 Byte |
| 4 | 00 (UserA) 01 (UserB) | | Info-Byte |
| 5 | 00 | | reserviert |
| 6 | 00 | | reserviert |
| 7 | 00 | | Schreibfunktion Infobereich |
| 8 | 45 | | Daten 1. Zeichen E _{asci} |
| 9 | 58 | | Daten 2. Zeichen X _{asci} |
| 10 | 44 | | Daten 3. Zeichen D _{asci} |
| 11 | 55 | | Daten 4. Zeichen U _{asci} |
| 12 | 4C | | Daten 5. Zeichen L _{asci} |
| 13 | 2D | | Daten 6. Zeichen _ _{asci} |
| 14 | 35 | | Daten 7. Zeichen 5 _{asci} |
| 15 | 33 | | Daten 8. Zeichen 3 _{asci} |
| 16 | 37 | | Daten 9. Zeichen 7 _{asci} |
| 17 | 20 | | Daten 10. Zeichen [Leer] _{asci} |
| 18 | 20 | | Daten 11. Zeichen [Leer] _{asci} |
| 19 | 20 | | Daten 12. Zeichen [Leer] _{asci} |
| 20 | 20 | | Daten 13. Zeichen [Leer] _{asci} |
| 21 | 20 | | Daten 14. Zeichen [Leer] _{asci} |
| 22 | 20 | | Daten 15. Zeichen [Leer] _{asci} |
| 23 | 20 | | Daten 16. Zeichen [Leer] _{asci} |

Lesen aus Inforegister

Das EXDUL-Modul besitzt mehrere 16-Byte breite Infobereiche, in welchen Modulinformationen wie die Seriennummer oder die Hardwarekennung stehen. Des Weiteren kann der Anwender auch die beschreibbaren User-Register auslesen.

| Infobereich | Info-Byte |
|-----------------|-----------|
| UserA | 0 |
| UserB | 1 |
| Hardwarekennung | 3 |
| Seriennummer | 4 |

Info: Alle Infobereiche lassen sich nur als ganzer 16-Byte-Block auslesen.

Beispiel: Infobereich UserA auslesen (User-String = „EXDUL-537“)

Gesendet wird ein 8Byte langer Block und empfangen ein 20Byte langer Block mit Inhalt von UserA bzw. UserB

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------------------------|--------------------------|-----------|---|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4Byte | 04 | Längenbyte → 16Byte |
| 4 | 00 (UserA) 01 (UserB) | Info-Byte | 45 | Daten 1. Zeichen E _{ASCII} |
| 5 | 00 | reserviert | 58 | Daten 2. Zeichen X _{ASCII} |
| 6 | 00 | reserviert | 44 | Daten 3. Zeichen D _{ASCII} |
| 7 | 01 | Lesefunktion Infobereich | 55 | Daten 4. Zeichen U _{ASCII} |
| 8 | | | 4C | Daten 5. Zeichen L _{ASCII} |
| 9 | | | 2D | Daten 6. Zeichen r _{ASCII} |
| 10 | | | 35 | Daten 7. Zeichen 5 _{ASCII} |
| 11 | | | 33 | Daten 8. Zeichen 3 _{ASCII} |
| 12 | | | 37 | Daten 9. Zeichen 7 _{ASCII} |
| 13 | | | 20 | Daten 10. Zeichen [Leer] _{ASCII} |
| 14 | | | 20 | Daten 11. Zeichen [Leer] _{ASCII} |
| 15 | | | 20 | Daten 12. Zeichen [Leer] _{ASCII} |
| 16 | | | 20 | Daten 13. Zeichen [Leer] _{ASCII} |
| 17 | | | 20 | Daten 14. Zeichen [Leer] _{ASCII} |
| 18 | | | 20 | Daten 15. Zeichen [Leer] _{ASCII} |
| 19 | | | 20 | Daten 16. Zeichen [Leer] _{ASCII} |

Beispiel: Infobereich Hardwarekennung auslesen

Gesendet wird ein 8Byte langer Block und empfangen ein 20Byte langer Block mit der Hardwarekennung

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|--------------------------|-----------|---|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4Byte | 04 | Längenbyte → 16Byte |
| 4 | 03 | Info-Byte | 45 | Daten 1. Zeichen E _{ascii} |
| 5 | 00 | reserviert | 58 | Daten 2. Zeichen X _{ascii} |
| 6 | 00 | reserviert | 44 | Daten 3. Zeichen D _{ascii} |
| 7 | 01 | Lesefunktion Infobereich | 55 | Daten 4. Zeichen U _{ascii} |
| 8 | | | 4C | Daten 5. Zeichen L _{ascii} |
| 9 | | | 2D | Daten 6. Zeichen * _{ascii} |
| 10 | | | 35 | Daten 7. Zeichen 5 _{ascii} |
| 11 | | | 33 | Daten 8. Zeichen 3 _{ascii} |
| 12 | | | 37 | Daten 9. Zeichen 7 _{ascii} |
| 13 | | | 20 | Daten 10. Zeichen [Leer] _{ascii} |
| 14 | | | 20 | Daten 11. Zeichen [Leer] _{ascii} |
| 15 | | | 20 | Daten 12. Zeichen [Leer] _{ascii} |
| 16 | | | 20 | Daten 13. Zeichen [Leer] _{ascii} |
| 17 | | | 20 | Daten 14. Zeichen [Leer] _{ascii} |
| 18 | | | 20 | Daten 15. Zeichen [Leer] _{ascii} |
| 19 | | | 20 | Daten 16. Zeichen [Leer] _{ascii} |

Beispiel: Infobereich Seriennummer auslesen

Gesendet wird ein 8Byte langer Block und empfangen ein 20Byte langer Block mit der Seriennummer

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|--------------------------|-----------|-----------------------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4Byte | 04 | Längenbyte → 16Byte |
| 4 | 04 | Info-Byte | 31 | Daten 1. Zeichen 1 _{dez} |
| 5 | 00 | reserviert | 30 | Daten 2. Zeichen 0 _{dez} |
| 6 | 00 | reserviert | 34 | Daten 3. Zeichen 4 _{dez} |
| 7 | 01 | Lesefunktion Infobereich | 34 | Daten 4. Zeichen 4 _{dez} |
| 8 | | | 30 | Daten 5. Zeichen 0 _{dez} |
| 9 | | | 32 | Daten 6. Zeichen 2 _{dez} |
| 10 | | | 36 | Daten 7. Zeichen 6 _{dez} |
| 11 | | | | reserviert |
| 12 | | | | reserviert |
| 13 | | | | reserviert |
| 14 | | | | reserviert |
| 15 | | | | reserviert |
| 16 | | | | reserviert |
| 17 | | | | reserviert |
| 18 | | | | reserviert |
| 19 | | | | reserviert |

Schreiben in LCD-Register

Das EXDUL-Modul stellt mehrere beschreibbare LCD-Register zur Verfügung. UserLCD1 und UserLCD2 entsprechen den beiden Zeilen während der UserMode-LCD-Anzeige. UserLCD1m und UserLCD2m sind zwei 16-Byte-Bereiche, welche direkt in einen nicht-flüchtigen Speicher (FLASH) abgelegt werden und beim Modulstart in die Register UserLCD1m bzw. UserLCD2m geladen werden. Alle Register sind nur als ganze 16-Byte-Blöcke beschreibbar.

| LCD-Befehl | LCD-Befehl-Byte |
|------------|-----------------|
| UserLCD1 | 0 |
| UserLCD2 | 1 |
| UserLCD1m | 2 |
| UserLCD2m | 3 |

Beispiel: Schreiben der Zeichenfolge EXDUL-537 in Register

| Byte | Senden | Empfangen | Beschreibung |
|------|--|-----------|--|
| 0 | 0C | 0C | Befehlscode 1. Byte |
| 1 | 00 | 00 | Befehlscode 2. Byte |
| 2 | 03 | 03 | Befehlscode 3. Byte |
| 3 | 05 | 00 | Längenbyte → 20 Byte |
| 4 | 00 (UserLCD1) 01 (UserLCD2) 02 (UserLCD1m) 03 (UserLCD2m) | | LCD-Befehl |
| 5 | 00 | | reserviert |
| 6 | 00 | | reserviert |
| 7 | 00 | | Schreibfunktion |
| 8 | 45 | | Daten 1. Zeichen E _{asci} |
| 9 | 58 | | Daten 2. Zeichen X _{asci} |
| 10 | 44 | | Daten 3. Zeichen D _{asci} |
| 11 | 55 | | Daten 4. Zeichen U _{asci} |
| 12 | 4C | | Daten 5. Zeichen L _{asci} |
| 13 | 2D | | Daten 6. Zeichen _ _{asci} |
| 14 | 35 | | Daten 7. Zeichen 5 _{asci} |
| 15 | 33 | | Daten 8. Zeichen 3 _{asci} |
| 16 | 37 | | Daten 9. Zeichen 7 _{asci} |
| 17 | 20 | | Daten 10. Zeichen [Leer] _{asci} |
| 18 | 20 | | Daten 11. Zeichen [Leer] _{asci} |
| 19 | 20 | | Daten 12. Zeichen [Leer] _{asci} |
| 20 | 20 | | Daten 13. Zeichen [Leer] _{asci} |
| 21 | 20 | | Daten 14. Zeichen [Leer] _{asci} |
| 22 | 20 | | Daten 15. Zeichen [Leer] _{asci} |
| 23 | 20 | | Daten 16. Zeichen [Leer] _{asci} |

Lesen von LCD-Register

Das EXDUL-Modul stellt mehrere beschreib- bzw. lesbare LCD-Register zur Verfügung. UserLCD1 und UserLCD2 entsprechen den beiden Zeilen während der UserMode-LCD-Anzeige. UserLCD1m und UserLCD2m sind zwei 16-Byte-Bereiche, welche direkt in einen nicht-flüchtigen Speicher (FLASH) abgelegt werden und beim Modulstart in die Register UserLCD1m bzw. UserLCD2m geladen werden. Alle Register sind nur als ganze 16-Byte-Blöcke lesbar.

| LCD-Befehl | LCD-Befehl-Byte |
|-----------------------|-----------------|
| UserLCD1 & UserLCD2 | 0 |
| UserLCD1m & UserLCD2m | 2 |

Beispiel: Lesen der Zeichenfolge EXDUL-537 aus Register

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------------------------------------|--------------------------------|-----------|---|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 03 | Befehlscode 3. Byte | 03 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4 Byte | 08 | Längenbyte → 32 Byte |
| 4 | 00 (UserLCD1&2) 02 (UserLCD1m&2m) | LCD-Befehl | 45 | Daten Zeile1 1. Zeichen E _{asci} |
| 5 | 00 | reserviert | 58 | Daten Zeile1 2. Zeichen X _{asci} |
| 6 | 00 | reserviert | 44 | Daten Zeile1 3. Zeichen D _{asci} |
| 7 | 01 | Lesefunktion von LCD-Registern | 55 | Daten Zeile1 4. Zeichen U _{asci} |
| 8 | | | 4C | Daten Zeile1 5. Zeichen L _{asci} |
| 9 | | | 2D | Daten Zeile1 6. Zeichen - _{asci} |
| 10 | | | 35 | Daten Zeile1 7. Zeichen 5 _{asci} |
| 11 | | | 33 | Daten Zeile1 8. Zeichen 3 _{asci} |
| 12 | | | 37 | Daten Zeile1 9. Zeichen 7 _{asci} |
| 13 | | | 20 | Daten Zeile1 10. Zeichen [Leer] _{asci} |
| 14 | | | 20 | Daten Zeile1 11. Zeichen [Leer] _{asci} |
| 15 | | | 20 | Daten Zeile1 12. Zeichen [Leer] _{asci} |
| 16 | | | 20 | Daten Zeile1 13. Zeichen [Leer] _{asci} |
| 17 | | | 20 | Daten Zeile1 14. Zeichen [Leer] _{asci} |
| 18 | | | 20 | Daten Zeile1 15. Zeichen [Leer] _{asci} |
| 19 | | | 20 | Daten Zeile1 16. Zeichen [Leer] _{asci} |
| 20 | | | 45 | Daten Zeile2 1. Zeichen E _{asci} |
| 21 | | | 58 | Daten Zeile2 2. Zeichen X _{asci} |
| 22 | | | 44 | Daten Zeile2 3. Zeichen D _{asci} |
| 23 | | | 55 | Daten Zeile2 4. Zeichen U _{asci} |
| 24 | | | 4C | Daten Zeile2 5. Zeichen L _{asci} |
| 25 | | | 2D | Daten Zeile2 6. Zeichen - _{asci} |
| 26 | | | 35 | Daten Zeile2 7. Zeichen 5 _{asci} |
| 27 | | | 33 | Daten Zeile2 8. Zeichen 3 _{asci} |
| 28 | | | 37 | Daten Zeile2 9. Zeichen 7 _{asci} |
| 29 | | | 20 | Daten Zeile2 10. Zeichen [Leer] _{asci} |
| 30 | | | 20 | Daten Zeile2 11. Zeichen [Leer] _{asci} |
| 31 | | | 20 | Daten Zeile2 12. Zeichen [Leer] _{asci} |
| 32 | | | 20 | Daten Zeile2 13. Zeichen [Leer] _{asci} |
| 33 | | | 20 | Daten Zeile2 14. Zeichen [Leer] _{asci} |
| 34 | | | 20 | Daten Zeile2 15. Zeichen [Leer] _{asci} |
| 35 | | | 20 | Daten Zeile2 16. Zeichen [Leer] _{asci} |

Schreiben des LCD-Modus

Die LCD-Anzeige des EXDUL-Moduls stellt mehrere Anzeige-Modi bereit. Diese können mit folgendem Befehl eingestellt werden. Der LCD-Modus wird in einem nicht-flüchtigen Speicher abgelegt und wird auch nach einem Neustart des Moduls verwendet

| LCD-Modus | LCD-Modus-Byte |
|-----------|----------------|
| IO-Mode | 0 |
| User-Mode | 1 |

Beispiel: Schreiben des LCD-Modus

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------------------------------|---------------------|-----------|---------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 03 | Befehlscode 3. Byte | 03 | Befehlscode 3. Byte |
| 3 | 02 | Längenbyte → 8 Byte | 00 | Längenbyte → 0 Byte |
| 4 | 04 | LCD-Befehl LCD-Mode | | |
| 5 | 00 | reserviert | | |
| 6 | 00 | reserviert | | |
| 7 | 00 | Schreibfunktion | | |
| 8 | 00 (IO-Mode) 01 (User-Mode) | LCD-Modus | | |
| 9 | 00 | reserviert | | |
| 10 | 00 | reserviert | | |
| 11 | 00 | reserviert | | |

Lesen des LCD-Modes

Die LCD-Anzeige des EXDUL-Moduls stellt mehrere Anzeige-Modi bereit. Der eingestellte LCD-Modus kann mit folgendem Befehl ausgelesen werden.

| LCD-Modus | LCD-Modus-Byte |
|-----------|----------------|
| IO-Mode | 0 |
| User-Mode | 1 |

Beispiel: Lesen des LCD-Modes

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|---------------------|--------------------------------|---------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 03 | Befehlscode 3. Byte | 03 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4 Byte | 01 | Längenbyte → 4 Byte |
| 4 | 04 | LCD-Befehl LCD-Mode | 00 (IO-Mode) 01 (User-Mode) | LCD-Modus |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 01 | Lesefunktion | 00 | reserviert |

Schreiben LCD-Kontrastwert

Über diesen Befehl ist der Display-Kontrast einstellbar. Werte zwischen 0 bis 4095 werden akzeptiert. Der Display-Kontrast verringert sich mit ansteigendem Wert. Eine angenehme Darstellung wird im Bereich 800 bis 1800 erreicht.

Beispiel: Schreiben Display-Kontrast-Wert 800

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|-----------------------------------|-----------|---------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 03 | Befehlscode 3. Byte | 03 | Befehlscode 3. Byte |
| 3 | 02 | Längenbyte → 8 Byte | 00 | Längenbyte → 0 Byte |
| 4 | 0B | LCD-Befehl LCD-Kontrast | | |
| 5 | 00 | reserviert | | |
| 6 | 00 | reserviert | | |
| 7 | 00 | Schreibfunktion | | |
| 8 | 50 | Kontrastwert (Lowbyte - 00...FF) | | |
| 9 | 03 | Kontrastwert (Highbyte - 00...0F) | | |
| 10 | 00 | reserviert | | |
| 11 | 00 | reserviert | | |

Lesen LCD-Kontrastwert

Über diesen Befehl ist der Display-Kontrast auslesbar. Der Wert kann zwischen 0 bis 4095 liegen. Der Display-Kontrast verringert sich mit ansteigendem Wert. Eine angenehme Darstellung wird im Bereich 800 bis 1800 erreicht.

Beispiel: Lesen Display-Kontrast-Wert 800

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|-------------------------|-----------|-----------------------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 03 | Befehlscode 3. Byte | 03 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4 Byte | 01 | Längenbyte → 4 Byte |
| 4 | 0B | LCD-Befehl LCD-Kontrast | 50 | Kontrastwert (Lowbyte - 00...FF) |
| 5 | 00 | reserviert | 03 | Kontrastwert (Highbyte - 00...0F) |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 01 | Lesefunktion | 00 | reserviert |

Schreiben Netzwerkkonfigurationen

Dieser Befehl ermöglicht das Abändern aller Netzwerkkonfigurationen wie IP-Adresse, Subnetzmaske, Hostname, Gateway, DNS-Adressen sowie die Einstellung der DHCP-Client-Funktion.

Beispiel: Schreiben der Netzwerkkonfigurationen

Hostname = „EXDUL-537“, IP = 192.168.0.63, Subnetzmaske = 255.255.255.0,

Gateway = 192.168.0.1, Primary DNS = 192.168.0.1, Secondary DNS = 217.237.151.115

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|--|-----------|---------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 08 | Befehlscode 3. Byte | 08 | Befehlscode 3. Byte |
| 3 | 0B | Längenbyte → 44 Byte | 00 | Längenbyte → 0 Byte |
| 4 | 00 | reserviert | | |
| 5 | 00 | reserviert | | |
| 6 | 00 | reserviert | | |
| 7 | 00 | Schreibfunktion | | |
| 8 | 45 | Hostname 1. Zeichen E _{ascii} | | |
| 9 | 58 | Hostname 2. Zeichen X _{ascii} | | |
| 10 | 44 | Hostname 3. Zeichen D _{ascii} | | |
| 11 | 55 | Hostname 4. Zeichen U _{ascii} | | |
| 12 | 4C | Hostname 5. Zeichen L _{ascii} | | |
| 13 | 2D | Hostname 6. Zeichen _ _{ascii} | | |
| 14 | 35 | Hostname 7. Zeichen 5 _{ascii} | | |
| 15 | 33 | Hostname 8. Zeichen 3 _{ascii} | | |
| 16 | 37 | Hostname 9. Zeichen 7 _{ascii} | | |
| 17 | 20 | Hostname 10. Zeichen [Leer] _{ascii} | | |
| 18 | 20 | Hostname 11. Zeichen [Leer] _{ascii} | | |
| 19 | 20 | Hostname 12. Zeichen [Leer] _{ascii} | | |
| 20 | 20 | Hostname 13. Zeichen [Leer] _{ascii} | | |
| 21 | 20 | Hostname 14. Zeichen [Leer] _{ascii} | | |
| 22 | 20 | Hostname 15. Zeichen [Leer] _{ascii} | | |
| 23 | 20 | Hostname 16. Zeichen [Leer] _{ascii} | | |
| 24 | 3F | IP-Adresse 4. Byte Dezimalwert 63 | | |
| 25 | 0 | IP-Adresse 3. Byte Dezimalwert 0 | | |
| 26 | A8 | IP-Adresse 2. Byte Dezimalwert 168 | | |
| 27 | C0 | IP-Adresse 1. Byte Dezimalwert 192 | | |

| | | | | |
|----|---|--------------------------------------|--|--|
| 28 | 00 | Subnetmaske 4.Byte Dezimalwert 0 | | |
| 29 | FF | Subnetmaske 3.Byte Dezimalwert 255 | | |
| 30 | FF | Subnetmaske 2.Byte Dezimalwert 255 | | |
| 31 | FF | Subnetmaske 1.Byte Dezimalwert 255 | | |
| 32 | 01 | Gateway 4.Byte Dezimalwert 1 | | |
| 33 | 00 | Gateway 3.Byte Dezimalwert 0 | | |
| 34 | A8 | Gateway 2.Byte Dezimalwert 168 | | |
| 35 | C0 | Gateway 1.Byte Dezimalwert 192 | | |
| 36 | 01 | Primary DNS 4.Byte Dezimalwert 1 | | |
| 37 | 00 | Primary DNS 3.Byte Dezimalwert 0 | | |
| 38 | A8 | Primary DNS 2.Byte Dezimalwert 168 | | |
| 39 | C0 | Primary DNS 1.Byte Dezimalwert 192 | | |
| 40 | 73 | Secondary DNS 4.Byte Dezimalwert 115 | | |
| 41 | 97 | Secondary DNS 3.Byte Dezimalwert 151 | | |
| 42 | ED | Secondary DNS 2.Byte Dezimalwert 237 | | |
| 43 | D9 | Secondary DNS 1.Byte Dezimalwert 217 | | |
| 44 | 0w 00 (DHCP disable) 01 (DHCP enable) | DCHP-Client Konfiguration | | |
| 45 | 00 | reserviert | | |
| 46 | 00 | reserviert | | |
| 47 | 00 | reserviert | | |

Lesen Netzwerkkonfigurationen

Dieser Befehl ermöglicht das Lesen aller Netzwerkkonfigurationen wie IP-Adresse, Subnetzmaske, Hostname, Gateway, DNS-Adressen, die Einstellung der DHCP-Client-Funktion sowie der MAC-Adresse.

Beispiel: Lesen der Netzwerkkonfigurationen

Hostname = „EXDUL-537“, IP = 192.168.0.63, Subnetzmaske = 255.255.255.0,

Gateway = 192.168.0.1, Primary DNS = 192.168.0.1, Secondary DNS = 217.237.151.115

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|---------------------|-----------|--|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 08 | Befehlscode 3. Byte | 08 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4 Byte | 0F | Längenbyte → 60 Byte |
| 4 | 00 | reserviert | 45 | Hostname 1. Zeichen E _{ASCII} |
| 5 | 00 | reserviert | 58 | Hostname 2. Zeichen X _{ASCII} |
| 6 | 00 | reserviert | 44 | Hostname 3. Zeichen D _{ASCII} |
| 7 | 01 | Lesefunktion | 55 | Hostname 4. Zeichen U _{ASCII} |
| 8 | | | 4C | Hostname 5. Zeichen L _{ASCII} |
| 9 | | | 2D | Hostname 6. Zeichen - _{ASCII} |
| 10 | | | 35 | Hostname 7. Zeichen 5 _{ASCII} |
| 11 | | | 33 | Hostname 8. Zeichen 3 _{ASCII} |
| 12 | | | 37 | Hostname 9. Zeichen 7 _{ASCII} |
| 13 | | | 20 | Hostname 10. Zeichen [Leer] _{ASCII} |
| 14 | | | 20 | Hostname 11. Zeichen [Leer] _{ASCII} |
| 15 | | | 20 | Hostname 12. Zeichen [Leer] _{ASCII} |
| 16 | | | 20 | Hostname 13. Zeichen [Leer] _{ASCII} |
| 17 | | | 20 | Hostname 14. Zeichen [Leer] _{ASCII} |
| 18 | | | 20 | Hostname 15. Zeichen [Leer] _{ASCII} |
| 19 | | | 20 | Hostname 16. Zeichen [Leer] _{ASCII} |
| 20 | | | 3F | IP-Adresse 4. Byte Dezimalwert 63 |
| 21 | | | 0 | IP-Adresse 3. Byte Dezimalwert 0 |
| 22 | | | A8 | IP-Adresse 2. Byte Dezimalwert 168 |
| 23 | | | C0 | IP-Adresse 1. Byte Dezimalwert 192 |
| 24 | | | 00 | Subnetzmaske 4. Byte Dezimalwert 0 |
| 25 | | | FF | Subnetzmaske 3. Byte Dezimalwert 255 |
| 26 | | | FF | Subnetzmaske 2. Byte Dezimalwert 255 |
| 27 | | | FF | Subnetzmaske 1. Byte Dezimalwert 255 |

| | | | | |
|----|--|--|---|--------------------------------------|
| 28 | | | 01 | Gateway 4.Byte Dezimalwert 1 |
| 29 | | | 00 | Gateway 3.Byte Dezimalwert 0 |
| 30 | | | A8 | Gateway 2.Byte Dezimalwert 168 |
| 31 | | | C0 | Gateway 1.Byte Dezimalwert 192 |
| 32 | | | 01 | Primary DNS 4.Byte Dezimalwert 1 |
| 33 | | | 00 | Primary DNS 3.Byte Dezimalwert 0 |
| 34 | | | A8 | Primary DNS 2.Byte Dezimalwert 168 |
| 35 | | | C0 | Primary DNS 1.Byte Dezimalwert 192 |
| 36 | | | 73 | Secondary DNS 4.Byte Dezimalwert 115 |
| 37 | | | 97 | Secondary DNS 3.Byte Dezimalwert 151 |
| 38 | | | ED | Secondary DNS 2.Byte Dezimalwert 237 |
| 39 | | | D9 | Secondary DNS 1.Byte Dezimalwert 217 |
| 40 | | | 0w 00 (DHCP disable) 01 (DHCP enable) | DCHP-Client Konfiguration |
| 41 | | | 00 | reserviert |
| 42 | | | 00 | reserviert |
| 43 | | | 00 | reserviert |
| 44 | | | | reserviert |
| 45 | | | | reserviert |
| 46 | | | 00 | MAC-Adresse 6.Zeichen |
| 47 | | | 00 | MAC-Adresse 5.Zeichen |
| 48 | | | 00 | MAC-Adresse 4.Zeichen |
| 49 | | | 3E | MAC-Adresse 3.Zeichen |
| 50 | | | B4 | MAC-Adresse 2.Zeichen |
| 51 | | | D4 | MAC-Adresse 1.Zeichen |
| 52 | | | 00 | reserviert |
| 53 | | | 00 | reserviert |
| 54 | | | 00 | reserviert |
| 55 | | | 00 | reserviert |
| 56 | | | 00 | reserviert |
| 57 | | | 00 | reserviert |
| 58 | | | 00 | reserviert |
| 59 | | | 00 | reserviert |
| 60 | | | 00 | reserviert |
| 61 | | | 00 | reserviert |
| 62 | | | 00 | reserviert |
| 63 | | | 00 | reserviert |

Schreiben der Securitykonfiguration

Das EXDUL-Modul stellt für die sichere Kommunikation einen Passwortschutz zur Verfügung. Ist dieser aktiviert, so muss bei jedem Transfer zum Modul das richtige Passwort gesendet werden. Im Default-Zustand ist dieser deaktiviert

| | |
|-----------------------|---------------|
| Securitykonfiguration | Security-Byte |
| Passwort deaktiviert | 0 |
| Passwort aktiviert | 1 |

Beispiel: Schreiben der Securitykonfiguration

| Byte | Senden | Empfangen | Beschreibung |
|------|--|-----------|---------------------------------------|
| 0 | 0C | 0C | Befehlscode 1. Byte |
| 1 | 00 | 00 | Befehlscode 2. Byte |
| 2 | 0C | 0C | Befehlscode 3. Byte |
| 3 | 01 | 01 | Längenbyte → 20 Byte |
| 4 | 00 (Passwort deaktiviert) 01 (Passwort aktiviert) | | Security-Byte |
| 5 | 00 | | reserviert |
| 6 | 00 | | reserviert |
| 7 | 00 | | Schreibfunktion Securitykonfiguration |

Lesen der Securitykonfiguration

Das EXDUL-Modul stellt für die sichere Kommunikation einen Passwortschutz zur Verfügung. Ist dieser aktiviert, so muss bei jedem Transfer zum Modul das richtige Passwort gesendet werden. Im Default-Zustand ist dieser deaktiviert.

| | |
|-----------------------|---------------|
| Securitykonfiguration | Security-Byte |
| Passwort deaktiviert | 0 |
| Passwort aktiviert | 1 |

Beispiel: Lesen der Securitykonfiguration

| Byte | Senden | Empfangen | Beschreibung |
|------|--------|--|------------------------------------|
| 0 | 0C | 0C | Befehlscode 1. Byte |
| 1 | 00 | 00 | Befehlscode 2. Byte |
| 2 | 0C | 0C | Befehlscode 3. Byte |
| 3 | 01 | 01 | Längenbyte → 20 Byte |
| 4 | 00 | 00 (Passwort deaktiviert) 01 (Passwort aktiviert) | Security-Byte |
| 5 | 00 | 00 | reserviert |
| 6 | 00 | 00 | reserviert |
| 7 | 01 | 00 | Lesefunktion Securitykonfiguration |

Ändern des Passwortes

Das EXDUL-Modul stellt für die sichere Kommunikation einen Passwortschutz zur Verfügung. Ist dieser aktiviert, so muss bei jedem Transfer zum Modul das richtige Passwort gesendet werden. Im Default-Zustand lautet das Passwort „11111111“ in ASCII und entspricht dem der Webpage. Mit dieser Funktion kann das Passwort abgeändert werden.

Beispiel: Ändern des Passwortes in „EXDUL537“

| Byte | Senden | Empfangen | Beschreibung |
|------|-------------------|-----------|--|
| 0 | 0C | 0C | Befehlscode 1. Byte |
| 1 | 00 | 00 | Befehlscode 2. Byte |
| 2 | 0D | 0D | Befehlscode 3. Byte |
| 3 | 02 | 00 | Längenbyte → 20 Byte |
| 4 | 45 _{hex} | | Passwort 1. Zeichen E _{ascii} |
| 5 | 58 _{hex} | | Passwort 2. Zeichen X _{ascii} |
| 6 | 44 _{hex} | | Passwort 3. Zeichen D _{ascii} |
| 7 | 55 _{hex} | | Passwort 4. Zeichen U _{ascii} |
| 8 | 4C _{hex} | | Passwort 5. Zeichen L _{ascii} |
| 9 | 35 _{hex} | | Passwort 6. Zeichen 5 _{ascii} |
| 10 | 33 _{hex} | | Passwort 7. Zeichen 3 _{ascii} |
| 11 | 37 _{hex} | | Passwort 8. Zeichen 7 _{ascii} |

Relaisausgänge lesen

Dieser Befehl ermöglicht das Einlesen des aktuellen Zustands der Relaisausgänge.

Relais durchgeschaltet = 1, Relais nicht durchgeschaltet = 0

Beispiel: Auslesen des Relaisausgangszustands

Gesendet wird ein 8Byte langer Block und empfangen ein 8Byte langer Block mit dem Zustand des Relaisausgangs

| Ausgangskanal | DOUT7 | DOUT6 | DOUT5 | DOUT4 | DOUT3 | DOUT2 | DOUT1 | DOUT0 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Schraubklemme | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Schaltzustand | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------------|---------------------|--------------|------------------------|
| 0 | 08 | Befehlscode 1. Byte | 08 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 (→ 4Byte) | Längenbyte | 01 (→ 4Byte) | Längenbyte |
| 4 | 01 | r/w Byte (1→ lesen) | 02 | Zustand Relaisausgänge |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |

Relaisausgänge schreiben

Dieser Befehl ermöglicht dem Anwender, die Ausgangsrelais zu sperren oder durchzuschalten.

Relais durchgeschaltet = 1, Relais nicht durchgeschaltet = 0

Beispiel: Ausgabe des Zustands (0x02_{hex}) an den Relaisausgängen (= DOUT0 gesperrt und DOUT1 durchgeschaltet)

Gesendet wird ein 8Byte langer Block und empfangen ein 4Byte Block als Bestätigung

| Ausgangskanal | DOUT7 | DOUT6 | DOUT5 | DOUT4 | DOUT3 | DOUT2 | DOUT1 | DOUT0 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Schraubklemme | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Schaltzustand | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------------|-------------------------|-----------|---------------------|
| 0 | 08 | Befehlscode 1. Byte | 08 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 (→ 4Byte) | Längenbyte | 00 | Längenbyte |
| 4 | 00 | r/w Byte (0→ schreiben) | | |
| 5 | 02 | Relaiszustand | | |
| 6 | 00 | reserviert | | |
| 7 | 00 | reserviert | | |

Relaisausgänge einzeln schreiben

Dieser Befehl ermöglicht dem Anwender, einzelne Ausgangsrelais zu sperren oder durchzuschalten.

Relais durchgeschaltet = 1, Relais nicht durchgeschaltet = 0

Beispiel: DOUT1 soll unabhängig der anderen Ausgänge geschaltet werden
Gesendet wird ein 8Byte langer Block und empfangen ein 4Byte Block als Bestätigung

| Ausgangskanal | DOUT7 | DOUT6 | DOUT5 | DOUT4 | DOUT3 | DOUT2 | DOUT1 | DOUT0 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Schraubklemme | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Kanal-Index | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------------|----------------------------|-----------|----------------------------|
| 0 | 08 | Befehlscode 1. Byte | 08 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 (→ 4Byte) | Längenbyte | 00 | Längenbyte |
| 4 | 02 | r/w (2→ einzeln schreiben) | | r/w (2→ einzeln schreiben) |
| 5 | 01 | Kanal-Index | | Kanal-Index |
| 6 | 01 | Relaiszustand | | Relaiszustand |
| 7 | 00 | reserviert | | reserviert |

Relaisausgänge setzen

Dieser Befehl ermöglicht dem Anwender, die Ausgangsrelais mit Hilfe einer Maske zu setzen

Relais durchgeschaltet = 1, Relais unverändert belassen = 0

Beispiel: Setzen aller mit der Maske (0x02_{hex}) ausgewählten Relaisausgängen (= DOUT1 durchgeschaltet, alle anderen Ausgänge behalten ihren alten Zustand x)

Gesendet wird ein 8Byte langer Block und empfangen ein 4Byte Block als Bestätigung

| Ausgangskanal | DOUT7 | DOUT6 | DOUT5 | DOUT4 | DOUT3 | DOUT2 | DOUT1 | DOUT0 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Schraubklemme | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Schaltzustand | x | x | x | x | x | x | 1 | x |

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------------|----------------------|-----------|---------------------|
| 0 | 08 | Befehlscode 1. Byte | 08 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 (→ 4Byte) | Längenbyte | 00 | Längenbyte |
| 4 | 03 | r/w Byte (3→ setzen) | | |
| 5 | 02 | Relaismaske | | |
| 6 | 00 | reserviert | | |
| 7 | 00 | reserviert | | |

Relaisausgänge zurücksetzen

Dieser Befehl ermöglicht dem Anwender, die Ausgangsrelais mit Hilfe einer Maske zurückzusetzen

Relais zurücksetzen = 1, Relais unverändert belassen = 0

Beispiel: Zurücksetzen aller mit der Maske (0x02_{hex}) ausgewählten Relaisausgängen (= DOUT1 geschlossen, alle anderen Ausgänge behalten ihren alten Zustand x)

Gesendet wird ein 8Byte langer Block und empfangen ein 4Byte Block als Bestätigung

| Ausgangskanal | DOUT7 | DOUT6 | DOUT5 | DOUT4 | DOUT3 | DOUT2 | DOUT1 | DOUT0 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Schraubklemme | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Schaltzustand | x | x | x | x | x | x | 1 | x |

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------------|----------------------------|-----------|---------------------|
| 0 | 08 | Befehlscode 1. Byte | 08 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 (→ 4Byte) | Längenbyte | 00 | Längenbyte |
| 4 | 04 | r/w Byte (3→ zurücksetzen) | | |
| 5 | 02 | Relaismaske | | |
| 6 | 00 | reserviert | | |
| 7 | 00 | reserviert | | |

Optokopplereingänge lesen

Dieser Befehl ermöglicht das Einlesen der aktuellen Zustände an den Optokopplereingängen.

Beispiel: Einlesen der Zustände an den Optokopplereingängen

Gesendet wird ein 4Byte langer Block und empfangen ein 8Byte langer Block mit den Zuständen.

Voraussetzung für diese Beispiele ist das Anlegen der Eingangspegel (hier 0x1B3) an den einzelnen Eingängen nach folgender Tabelle:

(0 = LOW = 0...3V, 1 = HIGH = 10...30V)

| | HB | | | | LB | | | | | | | |
|----------------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Ausgangs-kanal | DIN11 | DIN10 | DIN9 | DIN8 | DIN7 | DIN6 | DIN5 | DIN4 | DIN3 | DIN2 | DIN1 | DIN0 |
| Schraub-klemme | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| Schaltzu-stand | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|---------------------|--------------|---|
| 0 | 08 | Befehlscode 1. Byte | 08 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 01 | Befehlscode 3. Byte | 01 | Befehlscode 3. Byte |
| 3 | 00 | Längenbyte | 01 (→ 4Byte) | Längenbyte |
| 4 | | | B3 | Zustand Optokopplereingänge LB (DIN7-DIN0) |
| 5 | | | 01 | Zustand Optokopplereingänge HB (DIN10-DIN8) |
| 6 | | | 00 | reserviert |
| 7 | | | 00 | reserviert |

Zähler

Dieser Befehl ermöglicht den Zugriff auf die Zähler. So kann der Zähler gestartet, gestoppt, resettet und gelesen werden. Zudem besteht die Möglichkeit, das Overflow-Flag einzulesen und rückzusetzen. Der Zählerindex wird über das 3.Byte im Befehlscode ausgewählt.

| Code | Zähler-Befehlscode |
|------|--------------------------|
| 00 | Zähler starten |
| 01 | Zähler stoppen |
| 02 | Zähler resettet |
| 03 | Zählerstand lesen |
| 04 | reserviert |
| 05 | Overflow-Flag lesen |
| 06 | Overflow-Flag rücksetzen |

| Befehlscode 3.Byte | Zählerindex |
|--------------------|-------------|
| 00 | Zähler 0 |
| 01 | Zähler 1 |
| 02 | Zähler 2 |
| 03 | Zähler 3 |
| 04 | Zähler 4 |
| 05 | Zähler 5 |

Beispiel für Zähler0:

Zähler0 Start / Stop / Reset

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|----------------------|--|-----------|-------------------------------|
| 0 | 09 | Befehlscode 1. Byte | 09 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte (Zähler0) | 00 | Befehlscode 3. Byte (Zähler0) |
| 3 | 01 | Längenbyte | 01 | Längenbyte |
| 4 | bb 00 01 02 | Zähler Befehlscode Zähler0 starten Zähler0 stoppen Zähler0 resettet | bb | Zähler Befehlscode |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |

Zähler0 lesen

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|-------------------------------|--------------|-------------------------------|
| 0 | 09 | Befehlscode 1. Byte | 09 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte (Zähler0) | 00 | Befehlscode 3. Byte (Zähler0) |
| 3 | 01 | Längenbyte | 02 (→ 8Byte) | Längenbyte |
| 4 | 03 | Zähler Befehlscode | 03 | Zähler Befehlscode |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |
| 8 | | | ww | Zählerstand Byte0 |
| 9 | | | ww | Zählerstand Byte1 |
| 10 | | | ww | Zählerstand Byte2 |
| 11 | | | ww | Zählerstand Byte3 |

Zählerstand = Zählerstand Byte3 * 0x1000000 + Zählerstand Byte2 * 0x10000 + Zählerstand Byte1 * 0x100 + Zählerstand Byte0

Overflow-Flag Zähler0 lesen

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|---|--------------|---|
| 0 | 09 | Befehlscode 1. Byte | 09 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte (Zähler0) | 00 | Befehlscode 3. Byte (Zähler0) |
| 3 | 01 | Längenbyte | 02 (→ 8Byte) | Längenbyte |
| 4 | 05 | Zähler Befehlscode Overflow-Flag lesen | 05 | Zähler Befehlscode Overflow-Flag lesen |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 0f | Overflow-Flag |

Overflow-Flag Zähler0 rücksetzen

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|--|--------------|--|
| 0 | 09 | Befehlscode 1. Byte | 09 | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte (Zähler0) | 00 | Befehlscode 3. Byte (Zähler0) |
| 3 | 01 | Längenbyte | 01 (→ 4Byte) | Längenbyte |
| 4 | 06 | Zähler Befehlscode Overflow-Flag rücksetzen | 06 | Zähler Befehlscode Overflow-Flag rücksetzen |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |

Watchdog-Timer

Dieser Befehl ermöglicht den Zugriff auf die Zähler. So kann der Zähler gestartet, gestoppt, resettet und gelesen werden. Zudem besteht die Möglichkeit, das Overflow-Flag einzulesen und rückzusetzen.

| Code | Zähler-Befehlscode |
|------|--------------------------|
| 00 | Watchdog-Timer starten |
| 01 | Watchdog-Timer stoppen |
| 02 | Watchdog-Timer resettet |
| 03 | WDT Periodendauer setzen |

Watchdog-Timer Start / Stop / Reset

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|----------------------------|---|-----------|---------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 01 | Befehlscode 2. Byte |
| 2 | 01 | Befehlscode 3. Byte | 01 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte | 01 | Längenbyte |
| 4 | bb 00 01 02 03 | WDT Befehlscode WDT starten WDT stoppen WDT resettet WDT Periodendauer setzen | bb | Zähler Befehlscode |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |

Watchdog-Timer Periodendauer in ms einstellen

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|-----------------|--------------------------|-----------|---------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 01 | Befehlscode 2. Byte |
| 2 | 01 | Befehlscode 3. Byte | 01 | Befehlscode 3. Byte |
| 3 | 02 | Längenbyte | 01 | Längenbyte |
| 4 | 03 | WDT Periodendauer setzen | bb | WDT Befehlscode |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |
| 8 | pp ₀ | Periodendauer Byte0 | | |
| 9 | pp ₁ | Periodendauer Byte1 | | |
| 10 | pp ₂ | Periodendauer Byte2 | | |
| 11 | pp ₃ | Periodendauer Byte3 | | |

Periodendauer = Byte3 * 0x1000000 + Byte2 * 0x10000 + Byte1 * 0x100 + Byte0 [ms]

Initialisierung eines programmierbaren Logikzweiges

Dieser Befehl ermöglicht das Initialisieren eines der programmierbaren Logikzweige.

| Code | Eingangsfunktion |
|----------|--------------------------------|
| 00 | NONE |
| 01 | TRUE (logische 1) |
| 02 | FALSE (logische 0) |
| 03 .. 15 | reserviert |
| 16 | Zustand DIN0 |
| 17 | Zustand DIN1 |
| 18 | Zustand DIN2 |
| 19 | Zustand DIN3 |
| 20 | Zustand DIN4 |
| 21 | Zustand DIN5 |
| 22 | Zustand DIN6 |
| 23 | Zustand DIN7 |
| 24 | Zustand DIN8 |
| 25 | Zustand DIN9 |
| 26 | Zustand DIN10 |
| 27 | Zustand DIN11 |
| 28 .. 31 | reserviert |
| 32 | Steigende Flanke DIN0 (event) |
| 33 | Steigende Flanke DIN1 (event) |
| 34 | Steigende Flanke DIN2 (event) |
| 35 | Steigende Flanke DIN3 (event) |
| 36 | Steigende Flanke DIN4 (event) |
| 37 | Steigende Flanke DIN5 (event) |
| 38 | Steigende Flanke DIN6 (event) |
| 39 | Steigende Flanke DIN7 (event) |
| 40 | Steigende Flanke DIN8 (event) |
| 41 | Steigende Flanke DIN9 (event) |
| 42 | Steigende Flanke DIN10 (event) |
| 43 | Steigende Flanke DIN11 (event) |

| Code | Gatterfunktion |
|------|------------------|
| 00 | UND-Verknüpfung |
| 01 | ODER-Verknüpfung |

| Code | Ausgangsfunktion |
|------|--------------------------|
| 00 | NONE (Zweig deaktiviert) |
| 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) |

Beispiel: Initialisierung des ersten Logikzweiges

Dabei soll bei jeder steigenden Flanke am Optokopplereingang DIN0 (event) die Message 1 an den PC geschickt werden. (IN0 = 32, IN1 = TRUE, IN2 = TRUE, IN3 = TRUE, Gate = 0, OUT = 4)

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|-----------------------|-----------|---------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 02 | Befehlscode 2. Byte | 02 | Befehlscode 2. Byte |
| 2 | 10 | Befehlscode 3. Byte | 10 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4 Byte | 01 | Längenbyte → 4 Byte |
| 4 | 00 | Lesen-/Schreibbyte | 00 | Lesen-/Schreibbyte |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 01 | Zweigindex | 00 | reserviert |
| 8 | 20 | Funktion IN0 | | |
| 9 | 00 | reserviert | | |
| 10 | 00 | reserviert | | |
| 11 | 00 | reserviert | | |
| 12 | 01 | Funktion IN1 | | |
| 13 | 00 | reserviert | | |
| 14 | 00 | reserviert | | |
| 15 | 00 | reserviert | | |
| 16 | 01 | Funktion IN2 | | |
| 17 | 00 | reserviert | | |
| 18 | 00 | reserviert | | |
| 19 | 00 | reserviert | | |
| 20 | 01 | Funktion IN3 | | |
| 21 | 00 | reserviert | | |
| 22 | 00 | reserviert | | |
| 23 | 00 | reserviert | | |
| 24 | 00 | Gatterfunktion | | |
| 25 | 00 | reserviert | | |
| 26 | 00 | reserviert | | |
| 27 | 00 | reserviert | | |
| 28 | 04 | Ausgangsfunktion OUT0 | | |
| 29 | 00 | reserviert | | |
| 30 | 00 | reserviert | | |
| 31 | 00 | reserviert | | |

Error-Register auslesen

Dieser Befehl ermöglicht das Auslesen der beiden Error-Register.

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|-------------------------|-----------|-------------------------|
| 0 | FF | Befehlscode 1. Byte | FF | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte | 03 | Längenbyte |
| 4 | 00 | Error-Register auslesen | 00 | Error-Register auslesen |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |
| 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

Error-Register rücksetzen

Dieser Befehl ermöglicht das Rücksetzen der beiden Error-Register.

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|---------------------------|-----------|---------------------------|
| 0 | FF | Befehlscode 1. Byte | FF | Befehlscode 1. Byte |
| 1 | 00 | Befehlscode 2. Byte | 00 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte | 00 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte | 01 | Längenbyte |
| 4 | 01 | Error-Register rücksetzen | 01 | Error-Register rücksetzen |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |

Receivermodus aktivieren

Mit diesem Befehl kann der Receivermodus für die programmierbare Logik aktiviert werden. Nachdem eine zweite TCP/IP-Verbindung mit dem Modul aufgebaut wurde, wird für diese Verbindung der Receivermodus aktiviert. Dieser Befehl bekommt keine Rückantwort. Ist dieser Modus eingeschaltet, funktionieren nur noch die Receiverbefehle über diese TCP/IP-Verbindung. Sämtliche anderen Befehle, wie z. B. Eingänge lesen, müssen über die erste Verbindung erfolgen. Um den Receivermodus zu deaktivieren, muss der Befehl „Receivermodus deaktivieren“ auf der Receiver-Verbindung verwendet oder die Verbindung beendet werden.

| Byte | Senden | Beschreibung |
|------|--------|--------------------------|
| 0 | 0C | Befehlscode 1. Byte |
| 1 | 03 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4 Byte |
| 4 | 00 | Receivermodus aktivieren |
| 5 | 00 | reserviert |
| 6 | 00 | reserviert |
| 7 | 00 | reserviert |

Receivermodus Zähler lesen (Receivermodusbefehl)

Mit diesem Befehl kann der aktuelle Stand des Receivermodus-Zählers gelesen werden. Er kann nur bei der Verbindung in Receivermodus benutzt werden. Mit ihm kann überprüft werden, ob auch alle Messages angekommen sind. Der Wert sollte um 1 größer als der Indexwert der letzten Message sein.

$$\text{Zählerindex} = dd * 0x1000000 + cc * 0x10000 + bb * 0x100 + aa$$

| Byte | Senden | Beschreibung | Empfangen | Beschreibung |
|------|--------|----------------------------|--------------|----------------------------|
| 0 | 0C | Befehlscode 1. Byte | 0C | Befehlscode 1. Byte |
| 1 | 01 | Befehlscode 2. Byte | 01 | Befehlscode 2. Byte |
| 2 | 02 | Befehlscode 3. Byte | 02 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte | 02 (→ 8Byte) | Längenbyte |
| 4 | 02 | Receivermodus-Zähler lesen | 02 | Receivermodus-Zähler lesen |
| 5 | 00 | reserviert | 00 | reserviert |
| 6 | 00 | reserviert | 00 | reserviert |
| 7 | 00 | reserviert | 00 | reserviert |
| 8 | | | aa | Receivermodus-Index Byte0 |
| 9 | | | bb | Receivermodus-Index Byte1 |
| 10 | | | cc | Receivermodus-Index Byte2 |
| 11 | | | dd | Receivermodus-Index Byte3 |

Receivermodus deaktivieren (Receivermodusbefehl)

Mit diesem Befehl kann der Receivermodus für die programmierbare Logik deaktiviert werden. Dieser Befehl bekommt keine Rückantwort und ist nur auf einer aktiven Receiver-Verbindung benutzbar.

| Byte | Senden | Beschreibung |
|------|--------|----------------------------|
| 0 | 0C | Befehlscode 1. Byte |
| 1 | 03 | Befehlscode 2. Byte |
| 2 | 00 | Befehlscode 3. Byte |
| 3 | 01 | Längenbyte → 4 Byte |
| 4 | 01 | Receivermodus deaktivieren |
| 5 | 00 | reserviert |
| 6 | 00 | reserviert |
| 7 | 00 | reserviert |

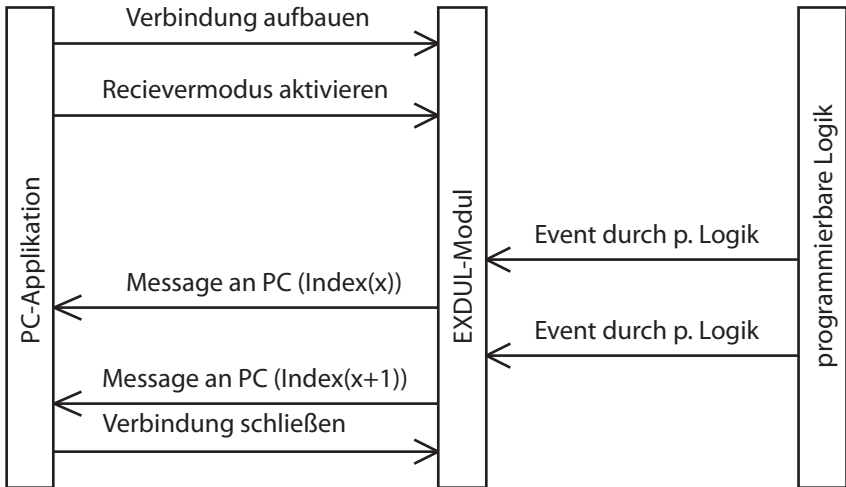
14.5 Receivermodus

Oft kommt es vor, dass in regelmäßigen Abständen eine Änderung z.B. an den Optokopplereingängen überprüft werden muss. Dies erfolgt normalerweise durch das Polling der entsprechenden Eingänge. Werden nun die Eingänge sehr oft abgefragt und sind auch noch mehrere Module gleichzeitig im Einsatz, so kann dies schnell zu einer erheblichen Auslastung des Netzwerkes führen. Um dies zu verhindern, steht eine programmierbare Logik zu Verfügung, welche z. B. bei einer Änderung eines Optokopplereingangs eine Nachricht an den PC schicken kann. Damit entfällt die Notwendigkeit eines regelmäßigen Polling des Optokopplereingangs.

Um diese Nachrichten der programmierbaren Logik empfangen zu können, muss eine extra TCP/IC-Verbindung (NetworkStream) mit dem Modul gestartet werden und diese mittels Befehl in den Receivermodus versetzt werden. Ist dies geschehen kann von dem Anwenderprogramm auf Nachrichten der programmierbaren Logik gewartet werden. Dies kann je nach Bedarf und Möglichkeit synchron bzw. asynchron geschehen oder in einem externen Task ausgelagert werden.

Die Nachricht muss nicht mit einem Acknowledge im Programm bestätigt werden.

Während mit dem Modul mehrere TCP/IP-Verbindungen möglich sind, kann immer nur eine Verbindung im Receivermodus sein. Wird der Receivermodus nicht mehr benötigt, so kann diese Verbindung entweder durch Befehl in eine normale TCP/IP-Verbindung zurückgesetzt werden oder auch durch einfaches Schließen der Verbindung beendet werden.



Für eine genauere Veranschaulichung des Receivermodus verweisen wir auf unsere Programmierbeispiele (z.B. Windows-C#-Beispiele)

14.6 Modulzugriff über LabVIEW® und EXDUL.dll

Dank der EXDUL.dll kann das Modul ohne großen Aufwand in ein LabVIEW-Projekt eingebunden werden. Neben LabVIEW® und der EXDUL.dll-Datei wird zudem auf dem Rechner das .NET-Framework benötigt.

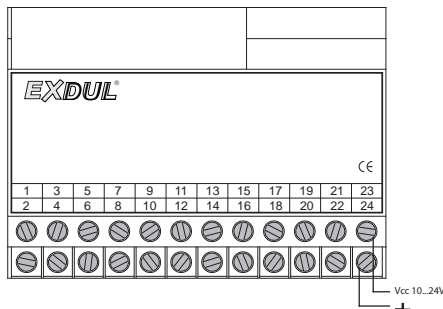
Genauere Informationen finden sie im EXDUL-LabVIEW-Tutorial.

15. FAQ - Problembehandlung

Nachfolgend finden Sie eine kurze Zusammenstellung der häufigsten bekannten Fehlerursachen, die während der Inbetriebnahme oder beim Zugriff auf das EXDUL-537 bzw. auf die EXDUL-Webpage auftreten können. Prüfen Sie bitte zunächst folgende Punkte, bevor Sie mit Ihrem Händler Kontakt aufnehmen.

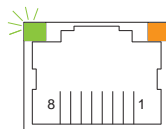
Ist die Versorgungsspannung für das EXDUL-537 korrekt angeschlossen?

Für den Betrieb ist eine Spannungsversorgung von +10V ... +24 V DC an Klemme 23 (Vcc) und Klemme 24 (GND) nötig. Überprüfen Sie die Schraubklemmverbindungen am Modul sowie das Netzteil und die Anschlüsse am Netzteil.



Leuchtet die Connect-LED an der RJ45-Buchse kontinuierlich grün?

Das EXDUL-537 bootet nach Anlegen der Betriebsspannung. Sobald eine stabile Ethernet-Verbindung besteht, leuchtet die Connect-LED an der 8P8C-Modularbuchse (RJ45-Buchse) des EXDUL-537 kontinuierlich grün. Sollte das nicht der Fall sein, überprüfen Sie die direkte Kabelverbindung zwischen EXDUL-537 und Computer (evtl. Crossover-Kabel notwendig), beim Netzwerkbetrieb prüfen Sie die Netzwerkleitungen zwischen EXDUL-537 und der Wand-Netzwerkbuchse, dem aktiven Ethernet-Switch oder Ethernet-Hub.



Besteht eine stabile Ethernet-Verbindung zwischen PC und Netzwerk?

Prüfen Sie das Netzkabel zwischen dem Computer und der Netzwerkbuchse (RJ45-Wandsteckdose), dem aktiven Ethernet-Switch oder Ethernet-Hub. Das Ethernetkabel muss für die Ethernet-Verbindung geeignet, nicht beschädigt und beidseitig korrekt eingesteckt sein. Bei den aktuellen PCs befinden sich an der Netzwerkbuchse des Netzwerkadapters meistens zwei LEDs. Bei bestehender Verbindung mit dem Netzwerk leuchtet die grüne LED kontinuierlich. Befindet sich an der Netzwerkbuchse des PCs nur eine LED, blinkt bzw. flackert diese bei funktionierender Netzwerkverbindung.

Ist die verwendete Netzwerkleitung für den Anschluss geeignet?

Beim Anschluss des EXDUL-537 an einen Switch, Hub oder PC mit Ethernet-Schnittstelle, die Auto-MDI(X) beherrscht, kann ein Standard Netzkabel (Cat. 5 oder höher) verwendet werden. Für ältere Rechner, deren Ethernet-Schnittstelle die Sende- und Empfangsleitungen nicht automatisch kreuzen, ist ein gekreuztes Netzkabel (Crossoverkabel) oder ein Crossover-Adapter notwendig.

Ist die Wandnetzwerkbuchse aktiv?

Falls Sie das EXDUL-537 über eine Wandsteckdose in ein fest installiertes Netzwerk integrieren, prüfen Sie zusammen mit Ihrem Netzwerkadministrator, ob die Wandnetzwerkbuchse aktiv und mit einem aktiven Ethernet-Switch oder Ethernet-Hub verbunden ist.

Ist die Ethernet-Schnittstelle des Computers aktiviert?

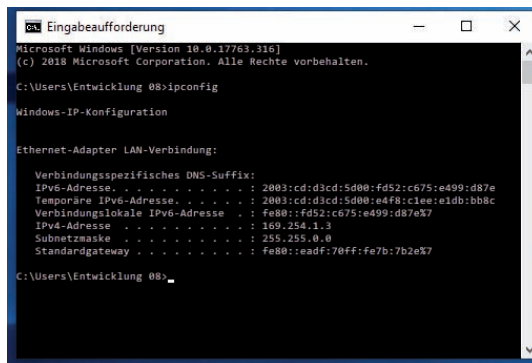
Im BIOS des PCs muss der Ethernetadapter aktiviert sein. Kontrollieren Sie im Windows Geräte Manager, ob dieser unter Netzwerkadapter gelistet ist. Der Eintrag darf nicht mit einem Ausrufezeichen versehen sein!

Sind die Netzwerkeinstellungen des Computers korrekt?

Jedes aktive Gerät in einem TCP/IP-basierten Netzwerk benötigt eine eindeutige IP-Adresse, die in die Netzwerk-ID und Geräte-ID gegliedert ist. Die Subnetzmaske trennt die IP-Adresse in den Netzwerkeil (Netzpräfix, Netzwerk-ID) und den Geräteteil (Geräte-ID). In der Grundeinstellung ist das EXDUL-537 auf DHCP-Client eingestellt.

Wie kann die Netzwerkeinstellung des PCs überprüft werden?

Die TCP/IP-Einstellungen Ihres Rechners können Sie über das Fenster Eigenschaften von Internetprotokoll Version 4 (TCP/IPv4) bzw. Status von LAN-Verbindung (siehe „Wie kann die IP-Adresse des PCs überprüft und geändert werden?“) oder über den einfachen Kommandozeilenbefehl IPCONFIG anzeigen. Wechseln Sie dazu in die MS-DOS-Eingabeaufforderung (siehe „Wie wechsele ich in die MS-DOS-Eingabeaufforderung?“), geben **ipconfig** ein und bestätigen Sie mit der **Eingabetaste** (Enter). Die Rückmeldung sollte ähnlich der folgenden Abbildung aussehen:



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

C:\Users\Entwicklung_00>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_00>
```

Wie kann die IP-Adresse des PCs überprüft und geändert werden?

Windows10:

Start -> Einstellungen -> Netzwerk und Internet -> Status -> Adapteroptionen ändern -> im Fenster Netzwerkverbindungen die gewünschte LAN-Verbindung auswählen (Doppelklick oder rechte Maustaste) -> Eigenschaften -> Internetprotokoll Version 4 (TCP/IPV4) -> Eigenschaften

Hinweis: Zum Ändern der TCP/IP-Einstellungen sind Administratorrechte notwendig!

The screenshot shows the EXDUL TCP/IP Configuration web interface. At the top, there is a navigation bar with 'Home', 'Status', 'Info', 'Config', and 'Peripherie'. The main heading is 'TCP/IP Konfiguration'. Below the heading, a note states: 'Diese Seite dient zur Einstellung und Änderung der Netzwerkparameter.' A red warning box contains the text: 'Achtung: Beachten Sie dringend die Hinweise im Handbuch, durch unkorrekte Einstellungen geht die Netzwerkkonnektivität verloren.' The configuration form includes the following fields: 'MAC Address' (04:14:3a:00:00:00), 'Host Name' (EXDUL-518), 'IP Address' (192.168.180.4), 'Gateway' (192.168.180.1), 'Subnet Mask' (255.255.255.0), 'Primary DNS' (192.168.100.1), and 'Secondary DNS' (0.0.0.0). There is a checkbox for 'Enable DHCP' which is checked, and a 'Daten speichern' button at the bottom of the form. The footer of the page reads 'EXDUL Webpage Copyright © 2021'.

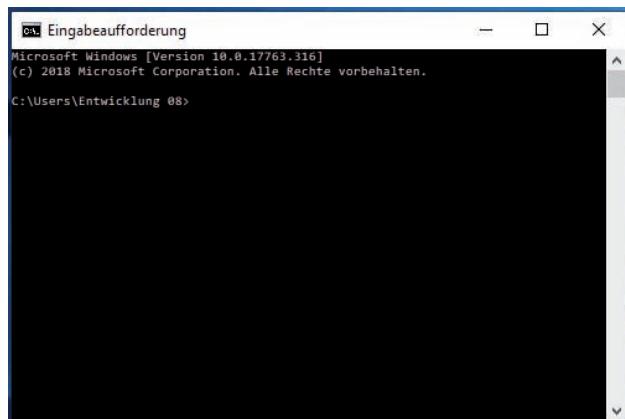
Wie wechsle ich in die MS-DOS-Eingabeaufforderung?

Windows10:

Start (Rechtsklick) -> Suchen -> **cmd** eingeben -> Bestätigung über die Eingabetaste (Enter)

oder

Start -> Windows-System -> Eingabeaufforderung



Ist es möglich EXDUL-5xx-Module in einem bestehenden Netzwerk zu orten und die Netzwerkdaten festzustellen?

Alle EXDUL-5xx-Module versenden in gewissen zeitlichen Abständen Erkennungssignale. Das Programm **ExdulUtility_v2_xx** (oder höher) wertet die Kennungsdaten aus und erstellt eine Liste mit Host-Namen, IP-Adresse und MAC-Adresse. Es eignet sich für ein einzelnes direkt am PC angeschlossenes EXDUL-5xx, wie für ein Netzwerk über Hub oder Switch mit mehreren Modulen. Für das Programm ist eine Freigabe in Ihrer Firewall erforderlich, falls diese die Kommunikation des Suchprogrammes mit den EXDUL-5xx verhindert.

16. Technische Daten

Digitale Eingänge über Optokoppler

12 bipolare Kanäle mit galvanischer Trennung

Überspannungsschutz-Dioden

Eingangsspannungsbereich

high = 10..30 Volt

low = 0..3 Volt

Digitale Ausgänge über Relais

8 Kanäle mit galvanischer Trennung über Relais

Kontakt A der Relais mit jeweils einer Anschlussklemme, Kontakt B mit gemeinsamer Anschlussklemme aller 8 Kanäle

Kontakt: 1 Schließer

Schaltstrom: max 2A / Kanal

Schaltspannung: max 30V AC / 30V DC

Schaltleistung: max. 60VA / 60W

Mechanische Lebensdauer: mind. $10 * 10^6$

Schaltspiele ohne Last

Kontakt Lebensdauer: $2 * 10^5$ Schaltspiele

Schalhäufigkeit mit Last: typ. 10/min

Schalhäufigkeit ohne Last: typ. 300/min

Zähler

6 programmierbare Zähler 32 Bit (belegen 6 Optokoppler-Eingänge)

Zählfrequenz: max. 5 kHz

Automatische Sicherung der Zählerstände im 10kHz Takt

Programmierbare Logik

Ausgänge über Verknüpfungen schalten

Meldung an PC bei Zustandsänderung an den Eingängen senden

TCP/IP-Server

Ausgänge über Verknüpfungen schalten

Meldung an PC bei Zustandsänderung an den Eingängen senden

TCP/IP-Server

Ausgänge über Verknüpfungen schalten

Meldung an PC bei Zustandsänderung an den Eingängen senden

Webserver

Konfiguration

Funktionstest

Update der Firmware

LCD Anzeige (nur EXDUL-537E)

Matrixanzeige mit 2 Zeilen und 16 Spalten zur Darstellung von 16 Zeichen je Zeile

Programmierbar zur Darstellung anwendungsspezifischer Daten oder als I/O-Zustandsanzeige

Modul-Anschlüsse

1 * 24polige Schraubklemmleiste

Ethernet RJ45-Buchse

Stromversorgung

extern

Spannung: 10 .. 30V

Strom: typ. 100mA bei 24V

Ethernet-Anschlussleitung

RJ45 Netzkabel Cat5 oder höher

Abmessungen

105 mm x 89 mm x 59 mm (l x b x h)

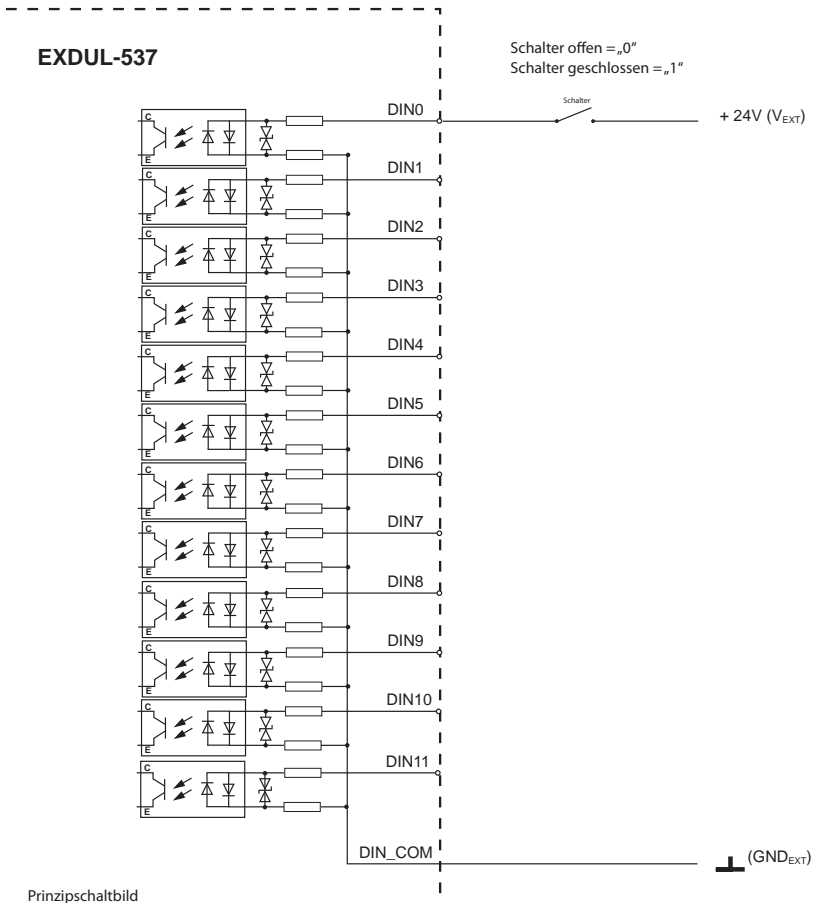
Gehäuse

Isolierstoffgehäuse mit integrierter Schnapptechnik zur DIN-EN-Hutschienenmontage

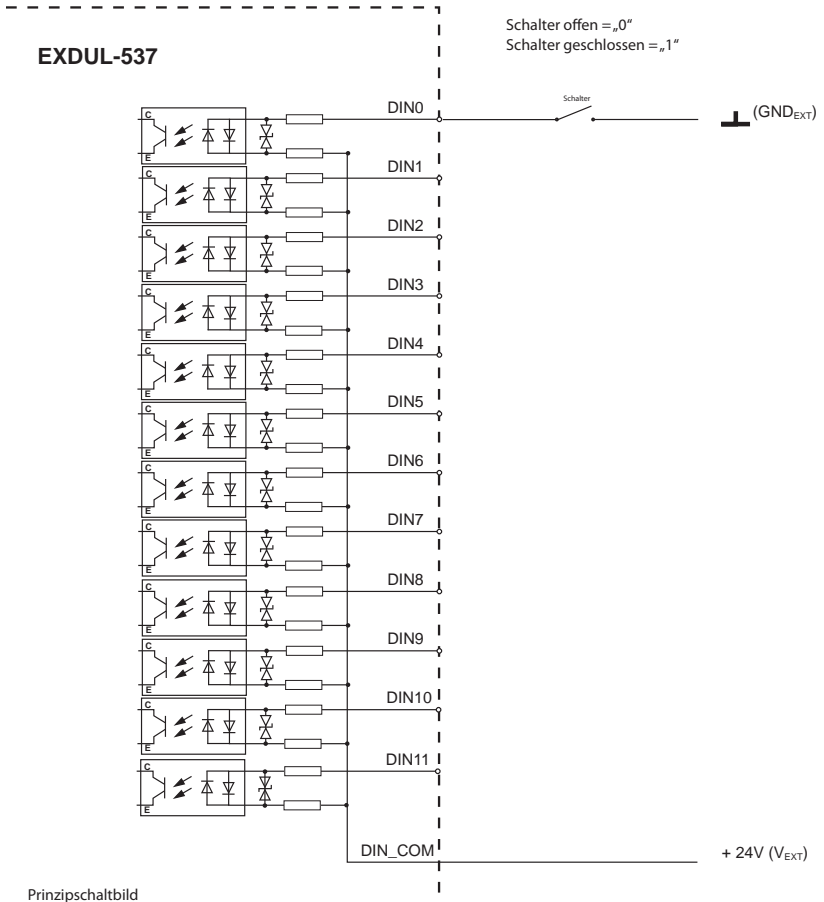
Geeignet für Aufbaumontagen, Schaltschrank- und Verteilereinbau sowie für mobile Tischeinsätze

17. Beschaltungsbeispiele

17.1 Beschaltung der Optokoppler-Eingänge

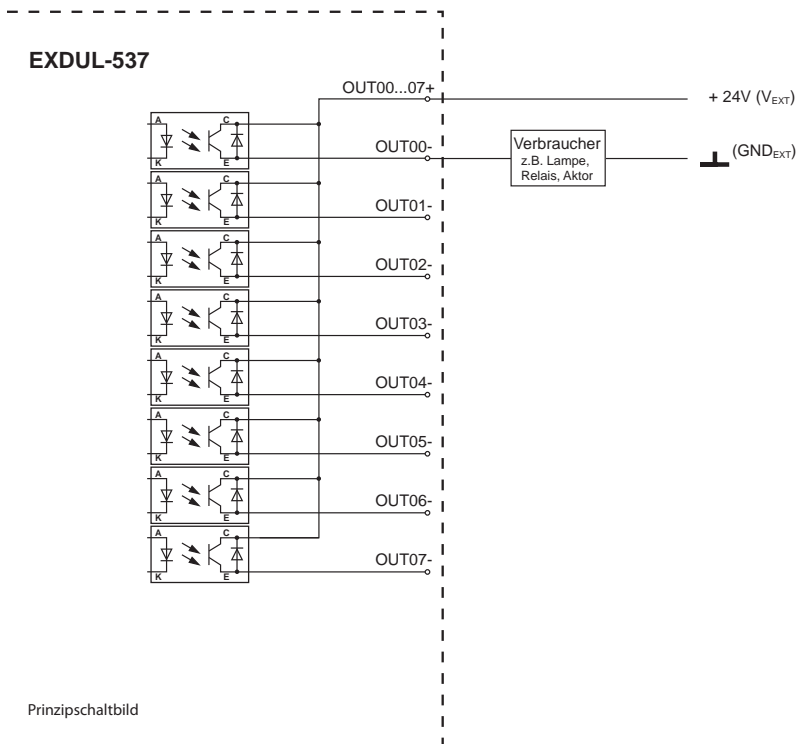


Grafik 17.1.1 Beschaltung der Optokopplereingänge



Grafik 17.1.2 Beschaltung der Optokopplereingänge revers

17.2 Beschaltung der Optokoppler-Ausgänge



Grafik 17.2 Beschaltung der Optokoppler-Ausgänge

18. ASCII-Tabelle

| Hex | Dez | Binär | Zeichen |
|-----|-----|----------|---------|
| 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 | [Leer] |
| 21 | 33 | 00100001 | ! |
| 22 | 34 | 00100010 | " |
| 23 | 35 | 00100011 | # |
| 24 | 36 | 00100100 | \$ |
| 25 | 37 | 00100101 | % |
| 26 | 38 | 00100110 | & |
| 27 | 39 | 00100111 | ' |

| Hex | Dez | Binär | Zeichen |
|-----|-----|----------|---------|
| 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 | Dez | Binär | Zeichen |
|-----|-----|----------|---------|
| 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 | Dez | Binär | Zeichen |
|-----|-----|----------|---------|
| 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 | Dez | Binär | Zeichen |
|-----|-----|----------|---------|
| 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 | Dez | Binär | Zeichen |
|-----|-----|----------|---------|
| 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. Produkthaftungsgesetz

Hinweise zur Produkthaftung

Das Produkthaftungsgesetz (ProdHaftG) regelt die Haftung des Herstellers für Schäden, die durch Fehler eines Produktes verursacht werden.

Die Verpflichtung zu Schadenersatz kann schon gegeben sein, wenn ein Produkt aufgrund der Form der Darbietung bei einem nichtgewerblichen Endverbraucher eine tatsächlich nicht vorhandene Vorstellung über die Sicherheit des Produktes erweckt, aber auch wenn damit zu rechnen ist, dass der Endverbraucher nicht die erforderlichen Vorschriften über die Sicherheit beachtet, die beim Umgang mit diesem Produkt einzuhalten wären.

Es muss daher stets nachweisbar sein, dass der nichtgewerbliche Endverbraucher mit den Sicherheitsregeln vertraut gemacht wurde.

Bitte weisen Sie daher im Interesse der Sicherheit Ihre nichtgewerblichen Abnehmer stets auf Folgendes hin:

Sicherheitsvorschriften

Beim Umgang mit Produkten, die mit elektrischer Spannung in Berührung kommen, müssen die gültigen VDE-Vorschriften beachtet werden.

Besonders sei auf folgende Vorschriften hingewiesen:

VDE0100; VDE0550/0551; VDE0700; VDE0711; VDE0860.

Sie erhalten VDE-Vorschriften beim vde-Verlag GmbH, Bismarckstraße 33, 10625 Berlin.

- * Vor Öffnen eines Gerätes den Netzstecker ziehen oder sicherstellen, dass das Gerät stromlos ist.
- * Bauteile, Baugruppen oder Geräte dürfen nur in Betrieb genommen werden, wenn sie vorher in ein berührungssicheres Gehäuse eingebaut wurden. Während des Einbaus müssen sie stromlos sein.
- * Werkzeuge dürfen an Geräten, Bauteilen oder Baugruppen nur benutzt werden, wenn sichergestellt ist, dass die Geräte von der Versorgungsspannung getrennt sind und elektrische Ladungen, die in im Gerät befindlichen Bauteilen gespeichert sind, vorher entladen wurden.
- * Spannungsführende Kabel oder Leitungen, mit denen das Gerät, das Bauteil oder die Baugruppe verbunden sind, müssen stets auf Isolationsfehler oder Bruchstellen untersucht werden. Bei Feststellen eines Fehlers in der Zuleitung muss das Gerät unverzüglich aus dem Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- * Bei Einsatz von Bauelementen oder Baugruppen muss stets auf die strikte Einhaltung der in der zugehörigen Beschreibung genannten Kenndaten für elektrische Größen hingewiesen werden.
- * Wenn aus den vorgelegten Beschreibungen für den nichtgewerblichen Endverbraucher nicht eindeutig hervorgeht, welche elektrischen Kennwerte für ein Bauteil gelten, so muss stets ein Fachmann um Auskunft ersucht werden.

Im Übrigen unterliegt die Einhaltung von Bau- und Sicherheitsvorschriften aller Art (VDE, TÜV, Berufsgenossenschaften usw.) dem Anwender/Käufer.

20. EG-Konformitätserklärung

Für die Erzeugnisse

EXDUL-537E EDV-Nummer A-375440
EXDUL-537S EDV-Nummer A-375420

wird hiermit bestätigt, dass sie den Anforderungen der betreffenden EG-Richtlinien entsprechen. Bei Nichteinhaltung der im Handbuch angegebenen Vorschriften zum bestimmungsgemäßen Betrieb der Produkte verliert diese Erklärung Ihre Gültigkeit.

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

Diese Erklärung wird verantwortlich für den Hersteller

Messcomp Datentechnik GmbH
Neudecker Str. 11
83512 Wasserburg

abgegeben durch

Dipl.Ing.(FH) Hans Schnellhammer

Wasserburg, 31.01.2019



Referenzsystem-Bestimmungsgemäßer Betrieb

Die Multifunktionsmodule EXDUL-537E und EXDUL-537S sind nicht selbstständig betreibbare Geräte, deren CE-Konformität nur bei gleichzeitiger Verwendung von zusätzlichen Computerkomponenten beurteilt werden kann. Die Angaben zur CE-Konformität beziehen sich deshalb ausschließlich auf den bestimmungsgemäßen Einsatz der Multifunktionsmodule in folgendem Referenzsystem:

| | | |
|--------------------|--------------------------|---|
| Schaltschrank: | Vero IMRAK 3400 | 804-530061C 802-563424J 802-561589J |
| 19" Gehäuse: | Vero PC-Gehäuse | 145-010108L |
| 19" Gehäuse: | Zusatzelektronik | 519-112111C |
| Motherboard: | GA-586HX | PIV 1.55 |
| Floppy-Controller: | auf Motherboard | |
| Floppy: | TEAC | FD-235HF |
| Grafikkarte: | Advantech | PCA-6443 |
| Schnittstellen: | EXDUL-537E EXDUL-537S | A-375440 A-375420 |