

EXDUL-392E

EDV-Nr.: A-382220

EXDUL-392S

EDV-Nr.: 382210

4 A/D-Eingänge 16 Bit (single-ended) oder
2 A/D-Eingänge 16 Bit (differentiell)
2 Strom A/D-Eingänge 15 Bit
1 Eingang über Optokoppler
1 Ausgang über Optokoppler
3 Temperaturmesseinheiten
1 Zähler 32 Bit
LCD-Anzeige (nur EXDUL-392E)

wasco[®]

Handbuch

Copyright® 2019 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-392 und dieser Dokumentation direkt oder indirekt entstandenen Schäden.

Wichtiger Hinweis:

Dieses Handbuch wurde für die Module EXDUL-392E und EXDUL-392S erstellt. Das EXDUL-392E bietet zusätzlich eine LCD-Anzeige, alle weiteren Funktionen der Module sind identisch. Für das EXDUL-392S 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-392E	7
3.2 Blockschaltbild EXDUL-392S	8
3.3 A/D-Eingänge Spannung	9
3.4 A/D-Eingänge Strom	9
3.5 Digitaler Eingang über Optokoppler	9
3.6 Digitaler Ausgang über Optokoppler	10
3.7 Zähler	10
3.8 3 Temperaturmesseinheiten PT100	10
3.9 LCD Anzeige (nur EXDUL-392E)	10
4. Inbetriebnahme	11
4.1 Anschluss an einen USB-Port	11
4.2 Spannungsversorgung über den USB-Port	11
4.3 Spannungsversorgung über externe Spannungsquelle	11
4.4 LCD-Anzeige während der Inbetriebnahme (nur EXDUL-392E)	11
4.5 LCD-Anzeige während des Betriebs (nur EXDUL-392E)	12
5. A/D-Eingänge Spannung-/Strommessung	13
5.1 Spannungsmessung	13
5.2 Strommessung	18
5.3 Messmodi	19
5.4 Abgleich der A/D-Eingänge	21
6. 3 Temperaturmesseinheiten PT100	22
6.1 Beschaltung	22
6.2 Messmöglichkeiten	23
6.3 Fehlererkennung	23
7. 1 Optokopplereingang	25
7.1 Pinbelegung des Eingangsoptokopplers	25
7.2 Eingangsbeschaltung	26
7.3 Eingangsstrom	26

8. 1 Optokopplerausgang	27
8.1 Pinbelegung des Ausgangsoptokopplers	27
8.2 Optokopplerdaten	27
8.3 Ausgangsbeschaltung	27
9. Informations-, LCD- und Userregister	28
9.1 Register HW-Kennung und Seriennummer	28
9.2 Speicherbereiche UserA, UserB, UserLCD1m* und UserLCD2m*	29
9.3 Display-Register UserLCD-Zeile1*, UserLCD-Zeile2* und LCD-Kontrast*	29
10. Installation der Treiber	30
10.1 Windows-Treiber	30
10.2 Linux-Treiber	30
11. Programmierung unter Windows[®]	31
11.1 Einführung	31
11.2 Programmierarten	31
11.3 Programmierung unter Windows mit der .NET EXDUL.dll Library	31
11.4 Programmierung mit seriellen COM-Port-Libraries	44
11.5 Modulzugriff über LabVIEW und EXDUL.dll	74
12. Programmierung unter Linux[®]	75
12.1 Einführung	75
12.2 Programmierung mit seriellen COM-Port-Libraries	75
13. Technische Daten	76
14. Beschaltungsbeispiele	78
14.1 Beschaltung des Optokoppler-Eingangs	78
14.2 Beschaltung des Optokoppler-Ausgangs	79
14.3 Beschaltung der A/D-Eingänge single-ended (Spannung)	80
14.4 Beschaltung der A/D-Eingänge differentiell (Spannung)	81
14.5 Beschaltung der A/D-Eingänge Strommessung	82
14.6 Beschaltung der A/D-Eingänge Spannung-/Strommessung	83
15. ASCII-Tabelle	84
16. Produkthaftungsgesetz	87
17. EG-Konformitätserklärung	89

1. Produktbeschreibung

























Das EXDUL-392 verfügt über vier massebezogene oder zwei differentielle 16-Bit A/D-Spannungseingangskanäle mit bipolaren Eingangsspannungsbereichen ($\pm 0,63$ V, $\pm 1,27$ V, $\pm 2,55$ V, $\pm 5,1$ V, $\pm 10,2$ V), sowie über 2 bipolare Stromeingänge (± 20 mA), welche auf $4..20$ mA abgeglichen sind. Für die Temperaturmessungen mit PT100-Sensoren besitzt das Modul 3 Messeinheiten mit jeweils eigener Stromquelle und Messeingängen. Die Wandlungsauslösung inklusive der damit verbundenen Konfiguration der A/D-Komponenten (Bereich/Kanalauswahl) erfolgt per Software-Befehl. Zusätzlich verfügt das Modul über einen digitalen Eingang und einen digitalen Ausgang mit galvanischer Trennung über hochwertige Optokoppler und zusätzlichen Schutzdioden. Der Optokopplereingang kann bei Bedarf als 32-Bit-Zählereingang programmiert und genutzt werden. Spezielle leistungsfähige Ausgangsoptokoppler bewältigen einen Schaltstrom von bis zu 150 mA.

Die programmierbare LCD-Anzeige beim EXDUL-392E ermöglicht die Darstellung von digitalen und analogen I/O-Statusinformationen oder programmierbaren anwenderspezifischen Daten.

Über USB oder eine externe Spannungsquelle wird das Modul mit der notwendigen Betriebsspannung versorgt. 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

AINU1+	2 	 1	AINU0+
AINU3+	4 	 3	AINU2+
AINI0-	6 	 5	AINI0+
AINI1-	8 	 7	AINI1+
FORCE0+	10 	 9	AGND
FORCE0-	12 	 11	RTDIN0+
RTDIN1+	14 	 13	FORCE1+
FORCE2+	16 	 15	FORCE1-
FORCE2-	18 	 17	RTDIN2+
DOUT0-	20 	 19	DOUT0+
DIN0-	22 	 21	DIN0+
GND_EXT	24 	 23	Vcc_EXT

Vcc_EXT:

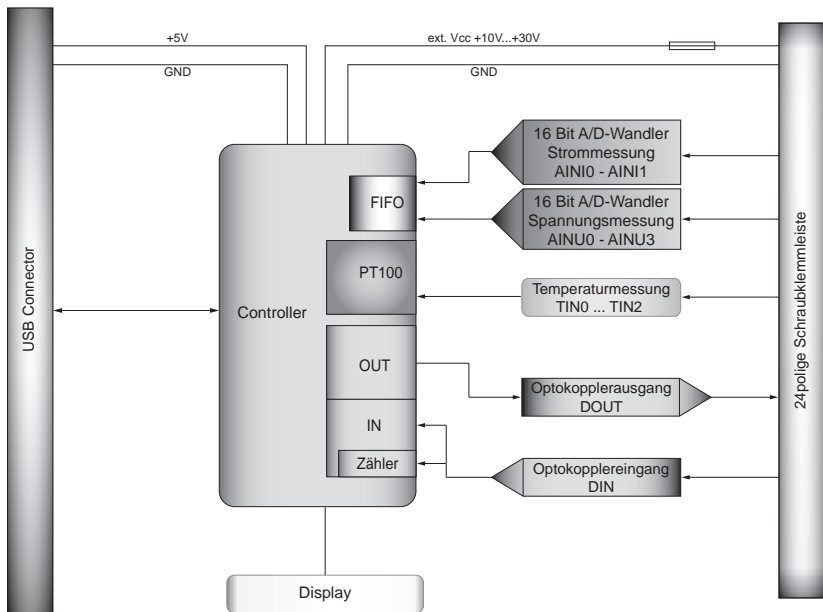
Anschlussklemme für externe Versorgungsspannung

GND_EXT:

Masse-Anschluss bei Verwendung einer externen Versorgungsspannung

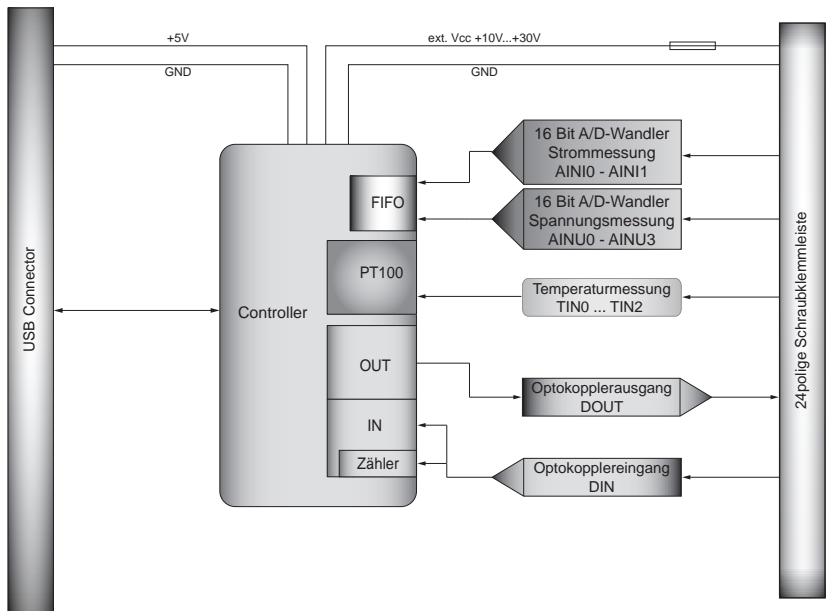
3. Systemkomponenten

3.1 Blockschahtbild EXDUL-392E



Grafik 3.1 Blockschahtbild EXDUL-392E

3.2 Blockschaftbild EXDUL-392S



Grafik 3.2 Blockschaftbild EXDUL-392S

3.3 A/D-Eingänge Spannung

4 Eingänge single-ended (se)
oder 2 Eingänge differentiell (diff)
oder kombiniert se/diff per SW wählbar
Auflösung: 16 Bit

Eingangsspannungsbereich bipolar

+/-0,63 Volt, +/-1,27 Volt, +/-2,55 Volt, +/-5.1 Volt, +/-10.2 Volt
+/-20.4V (nur Differenzeingänge)

FIFO: 10000 Messwerte

Eingangswiderstand: > 500 M Ω

Überspannungsschutz: +/- 50V

max. Abtastrate: 100 kS/s

3.4 A/D-Eingänge Strom

2 Eingänge

Auflösung 15 Bit

Messbereich +/-20mA

Abgeglichen auf 4..20mA

Eingangswiderstand 120 Ω

FIFO: 10000 Messwerte

max. Abtastrate: 100kS/s

3.5 Digitaler Eingang über Optokoppler

1 bipolarer Kanal

Überspannungsschutz-Dioden

Eingangsspannungsbereich

high = 10..30 Volt

low = 0..3 Volt

3.6 Digitaler Ausgang über Optokoppler

1 Kanal

Leistungsoptokoppler

Verpolungsschutz-Dioden

Ausgangsstrom: max. 150 mA

Spannung-CE: max. 50 V

3.7 Zähler

1 programmierbarer Zähler 32 Bit (belegt den Optokoppler-Eingang)

Zählfrequenz: max. 5 kHz

3.8 3 Temperaturmesseinheiten PT100

3-Leiteranschluss

3.9 LCD Anzeige (nur EXDUL-392E)

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 PC-Anschluss erfolgt einfach und unkompliziert Plug & Play über eine USB-Schnittstelle. Über USB oder eine externe Spannungsquelle wird das Modul mit der notwendigen Betriebsspannung versorgt.

4.1 Anschluss an einen USB-Port

Das EXDUL-392E / EXDUL-392S verfügt über ein USB 2.0 Interface und wird über die beiliegende USB-Anschlussleitung direkt an einen PC oder an einen USB-Hub angeschlossen. Der Anschluss erfolgt hotpluggable, d.h. das Modul ist auch im laufenden Betrieb anschließbar.

4.2 Spannungsversorgung über den USB-Port

Das Modul EXDUL-392 kann bei Bedarf ohne Einschränkungen ausschließlich über die USB-Schnittstelle versorgt werden. Dafür muss sichergestellt werden, dass der PC über das USB-Interface 500mA liefern kann.

4.3 Spannungsversorgung über externe Spannungsquelle

Die Firmware des EXDUL-392E / EXDUL-392S erkennt selbständig die Spannungsversorgung über eine externe Spannungsquelle. Wird an den Klemmen Vcc_EXT und GND_EXT (siehe Klemmenbelegung) eine Spannung von +10 V...+30 V DC angelegt, schaltet das Modul sofort auf Betriebsspannung „extern“ um. Die Spannungsversorgung über den USB-Port wird automatisch unterbrochen.

Achtung: Die Spannungsversorgung des Moduls darf während des Betriebs nicht mehr gewechselt werden!

4.4 LCD-Anzeige während der Inbetriebnahme (nur EXDUL-392E)

Während der Inbetriebnahme bzw. Start des Moduls erscheint im Display eine Infoanzeige in Form des Modulnamens. Nach fünf Sekunden wird der Modulname je nach LCD-Anzeigen-Konfiguration entweder durch die I/O-Statusanzeige oder UserLCD-Anzeige ersetzt.

4.5 LCD-Anzeige während des Betriebs (nur EXDUL-392E)

Bei der Inbetriebnahme des Moduls schaltet das Display nach ca. fünf Sekunden, je nach Einstellung, von der Infoanzeige in die I/O-Statusanzeige oder die UserLCD-Anzeige. Während der I/O-Anzeige werden in Zeile1 die aktuellen Zustände der Eingänge, in Zeile2 die Zustände der Ausgänge angezeigt. Falls beim letzten Betrieb des Moduls mit vorgesehenem Befehl der UserLCD-Modus aktiviert wurde, erscheint anstelle der I/O-Statusanzeige die UserLCD-Anzeige mit den Werten aus den Speicherbereichen UserLCD1m und UserLCD2m. Die Daten aus den beiden Registern werden solange angezeigt, bis neue Benutzerdaten über UserLCD-Zeile1 und UserLCD-Zeile2 auf die Anzeige geschrieben werden. Um einen „Screen-Burn“ zu vermeiden, wechselt die Anzeige im laufenden Betrieb etwa jede Minute für ca. fünf Sekunden von der I/O-Statusanzeige oder UserLCD-Anzeige in die Infoanzeige.

5. A/D-Eingänge Spannung-/Strommessung

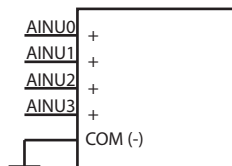
Das EXDUL-392 verfügt über 4 gemultiplexte single-ended oder 2 differenzielle 16-Bit-A/D-Spannungseingangskanäle mit programmierbarem Eingangsspannungsbereich sowie 2 16-Bit-A/D-Stromeingangskanäle. Die Konfiguration für die Wandlung (Kanal, Bereich) wird in Form von zwei Bytes mit der Wandlungsauslösung durch den PC übergeben. Der Messwert wird durch das Modul nach Fehlerkorrekturen (z.B. Offsetfehler) und einer Transformation in einen Spannungswert in μV bzw. einen Stromwert in μA als Antwort übermittelt oder in ein FIFO abgelegt.

5.1 Spannungsmessung

Zur Spannungsmessung werden bis zu 4 Eingangskanäle mit einstellbaren Eingangsspannungsbereichen und Messmodi zur Verfügung gestellt.

5.1.1 Single-ended Betrieb

Im Single-Ended Betrieb stehen max. 4 Eingangskanäle zur Verfügung. Alle Eingangsspannungen werden gegen die Masse (ADGND) der A/D-Komponenten gemessen (siehe Grafik 5.1.1). Eine genauere Beschreibung der Beschaltung ist in Kapitel 12.3 zu finden.



Grafik 5.1.1 A/D-Wandler Single-ended

Wie zuvor erwähnt wird dem Befehl zum Messen der Spannung ein Byte zur Kanalauswahl hinzugefügt. Welcher Wert für welchen Kanal einer single-ended Messung verwendet werden muss, ist aus der Tabelle 5.1 zu entnehmen.

Kanal-Byte	Single-ended Kanalauswahl				
	1	2	3	4	ADGND
0 _{dez}	+				-
1 _{dez}		+			-
2 _{dez}			+		-
3 _{dez}				+	-

Tabelle 5.1.1 A/D-Wandler Single-ended Messung

So muss für eine single-ended Messung an Kanal 3 der Pluspol der Spannungsquelle an AINU2 und der Minuspol an ADGND angeschlossen werden. Das Kanalbyte des Befehls besitzt den Wert 2_{dez}.

5.1.2 Differenz-Betrieb

Im Differenz-Betrieb stehen max. 2 Eingangskanäle zur Verfügung. In der Differenz-Betriebsart gibt es für jeden Kanal jeweils einen Plus- und einen Minus-Eingang (siehe Grafik 5.1.2-1). Bitte beachten Sie, dass für alle Kanäle ebenfalls ein Bezug zur Masse (ADGND) hergestellt werden muss. Eine genauere Beschreibung der Beschaltung ist in Kapitel 12.4 zu finden. Durch die Differenzmessung können allgemein auftretende Störspannungen auf beiden Signalleitungen und der Analogmasse reduziert werden.

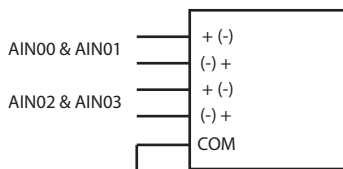


Tabelle 5.1.2-1 A/D-Wandler differenzielle Messung

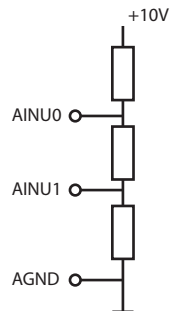
Auch hier findet die Kanalauswahl über das Kanalbyte im Befehl zur Spannungsmessung statt. Die entsprechenden Werte sind aus der folgenden Tabelle zu entnehmen.

Kanal-Byte	Differentielle Kanalauswahl				
	1	2	3	4	ADGND
8 _{dez}	+	-			
9 _{dez}	-	+			
10 _{dez}			+	-	
11 _{dez}			-	+	

Tabelle 5.1.2-2 A/D-Wandler differenzielle Messung

Als Beispiel soll nun die Differenz zwischen zwei Spannungen an den Eingängen AINU0 und AINU1 gemessen werden. Hierfür schließen sie die erste Spannung an AINU0 und die zweite an AINU1 an (siehe Grafik 5.1.2-2).

Nun kann als Kanalbyte entweder der Wert 8_{dez} (AINU0+ / AINU1-) oder 9_{dez} (AINU0- / AINU1+, Ergebnis ist eine negative Differenzspannung!) verwendet werden.



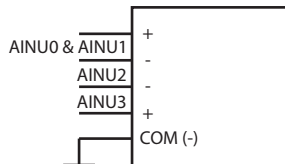
Grafik 5.1.2-2

Achtung:

Achten Sie darauf, dass die Differenz zwischen den Eingängen innerhalb des Eingangsspannungsbereiches liegen muss. Eine Eingangsspannung an AINU0 von +10V und einer Eingangsspannung an AINU1 von -10V ergibt eine Differenz von +20V, d.h. es muss ein Eingangsspannungsbereich von +/- 20.4V gewählt werden (siehe Kap. 5.1.4)

5.1.3 Kombination von Single-ended und Differenz Messung

Bei Bedarf können die Messvarianten wie in Grafik 5.1.3 auch von Kanal zu Kanal variiert werden oder sogar „on the fly“ zwischen den einzelnen Messungen geändert werden.



Grafik 5.1.3

5.1.4 Eingangsspannungsbereich

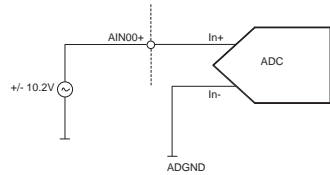
Für die Spannungsmessung stehen mehrere Eingangsspannungsbereiche zur Verfügung (+/-0.63 V, +/-1.27 V, +/-2.55 V, +/-5.1 V, +/-10.2 V). So kann der Messbereich an das Eingangssignal angepasst und somit die Messgenauigkeit optimiert werden. Für die Auswahl des Bereichs wird mit dem Messbefehl durch den PC ein Bereichsbyte an das Modul mitgesendet. Folgend sind zu den einzelnen Bereichen die dazugehörigen Bytewerte aufgelistet.

Eingangsspannungsbereich	
Bytewert	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1.27V
5	+/- 0.63V

Tabelle 5.1.4 A/D-Wandler Eingangsspannungsbereiche

a) Single-Ended Messung

Wie in Grafik 5.1.4.1 gezeigt, wird bei einer Single-Ended-Messung das Eingangssignal im Vergleich zur Masse gemessen. Die maximal bzw. minimal zu messende Spannung bei einem Spannungsbereich von +/- 10.2V beträgt +10.2V bzw. -10.2V.

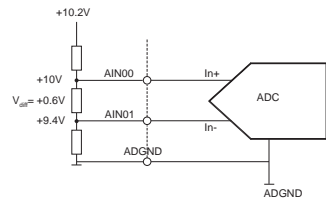


Grafik 5.1.4.1

Achtung: da die maximal zu messende Spannung am Analogeingang (z.B. AIN00+) 10.2V beträgt, ist der Spannungsbereich +/- 20.4V bei einer Single-Ended-Messung nicht vorhanden!

b) Differenzmessung

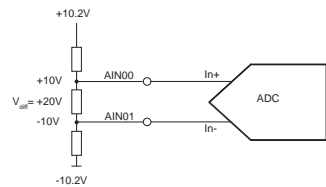
Bei Differenzmessungen entspricht der verwendete Eingangsspannungsbereich der maximalen Eingangsdifferenz zwischen den gewählten Eingängen. Dabei kann wie in Grafik 5.1.4.2 gezeigt ein Eingangsspannungsbereich von +/- 0.63V gewählt werden, obwohl an den Eingängen eine Spannung von bis zu +/- 10.2V anliegt.



Grafik 5.1.4.2

Bei der Differenzmessung gibt es im Gegensatz zur Single-Ended Messung zudem einen Eingangsspannungsbereich von +/- 20.4V.

Achtung: Für den Eingangsspannungsbereich +/- 20.4V gilt die maximale bzw. minimale Eingangsspannung von +10.2V bzw. -10.2V. Nur die Differenz zwischen zwei Eingängen darf +20.4V bzw. -20.4V betragen. (Z.B. AIN00 = +10.2V und AIN01 = -10.2V, $V_{diff} = 20.4V$)



Grafik 5.1.4.3

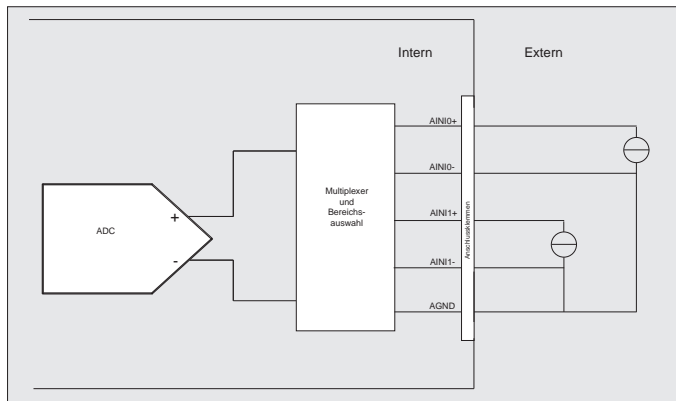
5.2 Strommessung

Neben den Spannungsmesseingängen gibt es 2 Strommeseinheiten mit einem Eingangsbereich von +/-20mA. Die Kanalauswahl erfolgt bei der Befehlsübergabe durch das Kanalbyte (siehe Tabelle 5.2). Werden Strom und Spannung gemessen, so wird bei der Spannungsmessung eine Differenzmessung für ein genaueres Ergebnis empfohlen.

Kanal-Byte	Kanalauswahl AINix		
	X	+	-
12 _{dez}	0	AINI0+	AINI0-
14 _{dez}	1	AINI1+	AINI1-

Tabelle 5.2 A/D-Wandler Single-ended Messung

Beschaltung:



5.3 Messmodi

Das EXDUL-392 besitzt mehrere Messmodi, um die Anwendung zu erleichtern.

5.3.1 Einfache Spannungsmessung

Bei der einfachen Spannungsmessung führt das Modul nach Erhalt des entsprechenden Befehls eine Messung an dem gewählten Eingang durch, gleicht diese ab und liefert den Wert in μV als Antwort an den Benutzer.

5.3.2 Einfache Spannungsmessung mit Mittelwertbildung

In diesem Messmodus führt das Modul an dem vom Benutzer gewählten Eingang 32 Messungen in einem Abstand von jeweils $10\ \mu\text{s}$ durch, bildet einen Durchschnitt, gleicht die Messung ab und liefert das Ergebnis in μV an den Anwender.

Dieser Messmodus eignet sich vor allem für kleinere Eingangsspannungsbereiche, um Störungen wie Rauschen zu unterdrücken.

5.3.3 Blockmessung mit Mittelwertbildung

Dieser Messmodus ist für Anwendungen gedacht, in welchen Spannungen an mehreren Eingängen möglichst genau und zeitnah gemessen werden sollen. Dabei werden bei der Übergabe des Befehls an das Modul die gewünschten Kanäle (bis zu 8) mit dem jeweiligen Spannungsbereich übergeben. Nach Erhalt des Befehls beginnt das Modul jeden gewünschten Kanal nacheinander 32 mal in $10\ \mu\text{s}$ -Schritten abzutasten.

Dauer = Kanalanzahl*32* $10\ \mu\text{s}$

Nach Abschluss werden die Werte abgeglichen und in μV an den Anwender zurückgeschickt.

Beispiel:



Grafik 5.3

In diesem Beispiel sollen drei Kanäle abgetastet werden (z.B. AINU1+, AINU2+, AINU3+). Diese Kanäle werden mit dem Befehl übergeben und das Modul beginnt mit den 32 Messungen des ersten Kanals (hier AINU1+). Sobald die Messungen des ersten Kanals abgeschlossen sind, wird mit der Abtastung des 2. Kanals begonnen. Wurden alle Kanäle fertig abgetastet, (hier nach $960\mu\text{s} = \text{Kanalanzahl} * 32 * 10\mu\text{s}$), werden Offset und Gain-Fehler abgeglichen und die Spannungen in μV übergeben.

5.3.4 Mehrfachmessung

Beim Messmodus Mehrfachmessung können bis zu 8 Kanäle mehrfach (bis zu 65535 mal) abgetastet werden. Dabei werden bei der Übergabe des Befehls neben der gewünschten Abtastrate (1 - 100kS/s) die gewünschten Kanäle mit dem jeweiligen Spannungsbereich übergeben. Nach Erhalt des Befehls führt das Modul die Messungen durch und speichert abgeglichenen Werte in μV in das FIFO ab. Aus dem FIFO können diese Werte jederzeit abgeholt werden. Dabei ist darauf zu achten, dass das FIFO nicht überläuft. Zudem darf in diesem Zeitraum kein EXDUL-Info-Register beschrieben werden.

5.3.5 Dauermessung

Beim Messmodus Dauermessung können bis zu 8 Kanäle mit beliebigem Messbereich und mit bis zu 100kS/s im Dauerbetrieb abgetastet werden. Hierfür gibt es einen Start- und einen Stop-Befehl. Die abgeglichenen Messwerte werden in μV und/oder μA in ein FIFO geschrieben und können von dort jederzeit abgeholt werden. Dabei ist darauf zu achten, dass das FIFO nicht überläuft. In dem Zeitraum der Dauermessung darf kein EXDUL-Info-Register beschrieben werden.

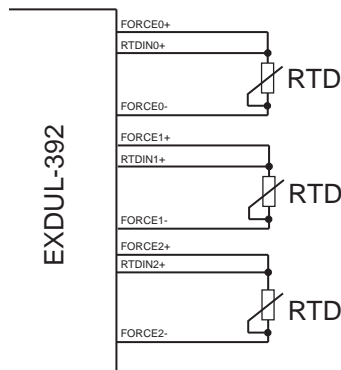
5.4 Abgleich der A/D-Eingänge

Das Modul wird beim Endtest unserer Produktion bei einer Umgebungstemperatur von ca. 20°C abgeglichen. Sollten bei der Endanwendung größere Temperaturabweichungen vorhanden sein, kann die A/D-Komponente des Moduls mittels nachträglichem Abgleich an die Umgebung angepasst werden. Die benötigte Software steht auf der CD bzw. im Internet zur Verfügung.

6. 3 Temperaturmesseinheiten PT100

Das Modul EXDUL-392 besitzt zur Temperaturmessung mit PT100-Sensoren (IEC 751 $\alpha = 0.00385$) drei Messeinheiten, welche zur Bestimmung der Temperatur eine 3-Leiterschaltung verwendet. Die 3-Leiterschaltung sorgt für eine automatische Messfehlerkompensation der Sensorleitungen und führt zu einer genaueren Temperaturmessung. Dabei ist darauf zu achten, dass die Leitungen zum Sensor den selben Widerstand besitzen müssen (identische Länge, Querschnitt und Material). Jede Messeinheit liefert während der Temperaturmessung den nötigen Messstrom und übergibt dem Anwender nach abgeschlossener Messung den Temperaturwert und bei Bedarf den Widerstandswert.

6.1 Beschaltung



6.2 Messmöglichkeiten

Bei der Durchführung einer Messung gibt es mehrere Modi, welche bestimmen, wie das Messergebnis verarbeitet oder als Befehlsantwort an den Anwender geschickt werden soll (z.B. mΩ oder °C).

6.2.1 Widerstandsmessung

In diesem Modus wird der angeschlossene Widerstand gemessen und in mΩ an den PC übergeben. Der zu messende Widerstand darf im Bereich von 0 bis 370Ω liegen.

6.2.2 Temperaturmessung PT100 IEC 751

Wird dieser Modus verwendet, so misst das Modul den Sensorwiderstand und berechnet die daraus resultierende Temperatur mit Hilfe der aus IEC 751 ($\alpha=0.00385$) vorgegebenen Kennlinie des Sensors. Die Temperatur wird mit dem Faktor 100 an den PC zurück geschickt.

Verwendete Callendar-Van Dusen Koeffizienten:

$$a = 3.908030 \times 10^{-3}$$

$$b = -5.7750 \times 10^{-7}$$

$$c = -4.18301 \times 10^{-12}$$

6.3 Fehlererkennung

Um Fehler bei der Temperaturmessung erkennen zu können, gibt es eine Möglichkeit, einen Fehlertest durchzuführen. Hier können Fehler wie Leiterbruch, Kurzschlüsse und Über-/Unterspannung erkannt werden.

Wird mit Aufruf des entsprechenden Befehls ein Fehlertest durchgeführt, wird nach einigen wenigen ms ein Fehlerbyte an den Anwender gesendet, welches der Fehleranalyse dient.

Während des Fehlertests kann keine Temperaturmessung durchgeführt werden.

Codes bei Fehlermeldung

Fehler-Bit	mögliche Fehlerursache	Fehlerbeschreibung
D7	reserviert	
D6	reserviert	
D5	Fehler bei der Verdrahtung	
D4	Fehler bei der Verdrahtung	
D3	Fehler bei der Verdrahtung	
D2	Overvoltage oder Under-voltage	evtl. externe Spannung eingespeist

Steht in einem Bit eine 1, so liegt entsprechend dessen Bedeutung ein Fehler vor.

7. 1 Optokopplereingang

Das EXDUL-392 verfügt über 1 Eingangskanal, dessen galvanische Trennung mittels Optokoppler erreicht wird. Die Isolationsspannung zwischen Masse des Moduls und Eingang beträgt 500 Volt.

7.1 Pinbelegung des Eingangsoptokopplers

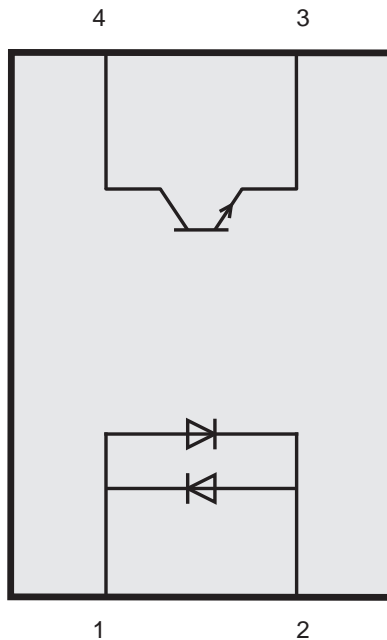


Abb. 7.1

7.2 Eingangsbeschaltung

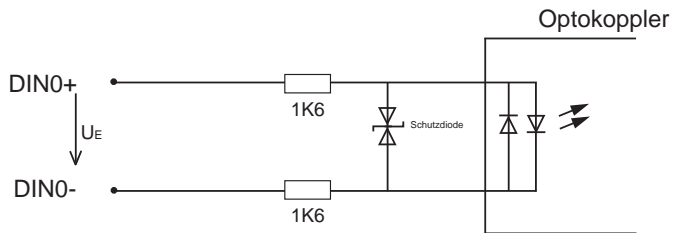


Abb. 7.2

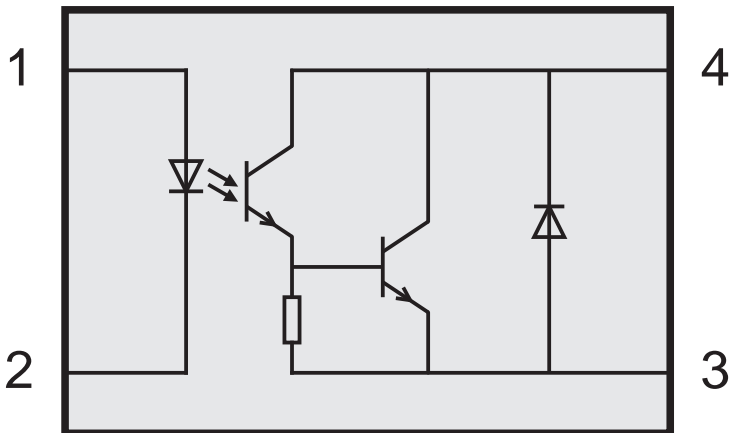
7.3 Eingangsstrom

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

8. 1 Optokopplerausgang

Das EXDUL-Modul verfügt über einen Ausgangskanal, dessen galvanische Trennung ebenfalls mittels Optokoppler erreicht werden. Die Isolationsspannung zwischen Masse des Moduls und Ausgang beträgt 500 Volt.

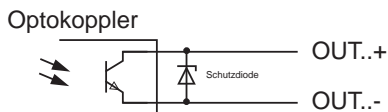
8.1 Pinbelegung des Ausgangsoptokopplers



8.2 Optokopplerdaten

Spannung-CE:	max. 50V
Spannung-EC:	0,1V
Strom-CE:	150 mA

8.3 Ausgangsbeschaltung



9. Informations-, LCD- und Userregister

9.1 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	-	3	9	2			V	1	.	0	1
	45 _{hex}	58 _{hex}	44 _{hex}	55 _{hex}	4C _{hex}	2D _{hex}	33 _{hex}	39 _{hex}	32 _{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 9.1 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-392 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.

9.2 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.

9.3 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-392E zutreffend, bei EXDUL-392S ohne Funktion!

10. Installation der Treiber

10.1 Windows-Treiber

Achtung: ab Windows10 muss kein extra Treiber für das Modul installiert werden!

Sobald das USB-Modul EXDUL-392E / EXDUL-392S das erste Mal am PC angeschlossen wird, erkennt Windows automatisch ein neues Gerät und sucht nach einem passenden Treiber.

Geben Sie zur Treiberinstallation dem Windows-Hardwareassistenten den Ordner bzw. das Verzeichnis und den Namen der Setup-Datei „wascoxmfe_v0x.inf“ (anstelle von x die Versions-Nr. der INF-Datei eintragen z.B. wascoxmfe_v06.inf) an.

Nach der Aktualisierung der Treiberdatenbank informiert Sie der Hardwareassistent über die erfolgreiche Installation des Treibers.

Im Windows-Gerätmanager wird das EXDUL-392E / EXDUL-392S im Verzeichnis Anschlüsse (COM/LPT) als Wasco-USB-Kommunikationsport oder Serielles USB-Gerät COMx geführt. Jedes Windowsprogramm kann auf die virtuelle Schnittstelle so zugreifen, als handle es sich um einen echten COM-Port.

10.2 Linux-Treiber

Das EXDUL-392 verwendet einen virtuellen Standard COM-Port-Treiber, welcher bei den meinsten gängigen Linux-Distributionen bereits installiert ist.

Wird das Modul an die USB-Schnittstelle angeschlossen, wird das Modul im dev-Ordner aufgelistet (z.B. als ttyACM0 unter Ubuntu).

11. Programmierung unter Windows®

11.1 Einführung

Nach erfolgreicher Installation wird das EXDUL-392E / EXDUL-392S im Windows-Gerätemanager als Wasco-Communications-Port COMx bzw. als serielles USB-Gerät COMx (ab WIN10) geführt. Es handelt sich hierbei um ein CDC-Device (Communications Device Class), das über einen virtuellen COM-Port angesprochen wird.

Der Softwarezugriff auf diesen virtuellen COM-Port erfolgt wie über eine normale COM-Schnittstelle über Standard-Windows®-Treiber, eine Installation eines zusätzlichen Treibers ist nicht notwendig.

11.2 Programmierarten

Für den Zugriff auf das EXDUL-Modul gibt es mehrere Möglichkeiten. So kann für die Programmierung unter .NET die Library EXDUL.dll verwendet werden. Diese ermöglicht eine leichten und schnellen Einstieg, um den Zugriff auf das Modul zu programmieren.

Des Weiteren können auch serielle COM-Port-Libraries verwendet werden, welche bei vielen Programmiersprachen wie C oder Delphi vorhanden sind. Sie ermöglichen oft eine breite Einstellmöglichkeit der Schnittstelle und teilweise auch eine Eventprogrammierung (Lesepuffer muss nicht gepollt werden).

LabVIEW-Anwender können ebenfalls mit Hilfe der EXDUL.dll oder den VISA-Funktionsblöcken (Serial Port) leicht auf das Modul zugreifen.

11.3 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-392 -> EXDUL-592) zu wechseln.

Open: Verbindung zu Modul aufbauen
[bool](#) Open()
Rückgabewerte: true wenn erfolgreich / false bei Fehler

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

Optokopplerausgänge lesen:

[uint](#) GetOptoOut()

Rückgabewerte: Zustand der Optokopplerausgänge

Zusammenfassung: Liest den Zustand der Optokopplerausgänge

Optokopplerausgänge schreiben:

void SetOptoOut([uint](#) value)

Parameter: *value*: Zustand der Ausgänge

Zusammenfassung: Setzt die Optokopplerausgänge

Optokopplereingänge lesen:

[uint](#) GetOptoIn()

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 resettet:

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

AD-Einzelmessung:

`int` GetADC(`byte` channel, `byte` range)

Parameter: *channel*: Kanal
range: Messbereich

Rückgabewerte: Messwert in μV

Zusammenfassung: Führt eine ADC-Messung durch.

Kanal:

Kanal	Kanalbyte
Single-ended	
AIN00	0
AIN01	1
AIN02	2
AIN03	3
AIN04	4
AIN05	5
AIN06	6
AIN07	7
Differenzmessung	
AIN00+ / AIN01-	8
AIN00- / AIN01+	9
AIN02+ / AIN03-	10
AIN02- / AIN03+	11
AIN04+ / AIN05-	12
AIN04- / AIN05+	13
AIN06+ / AIN07-	14
AIN06- / AIN07+	15

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1,27V
5	+/- 0.63V

AD-Einzelmessung mit Mittelwertbildung aus 32 Messungen:

public [int](#) GetADC_Mean([byte](#) channel, [byte](#) range)

Element von [EXDUL.EXDUL381](#)

Parameter: *channel*: Kanal
range: Messbereich

Rückgabewerte: Messwert in μV

Zusammenfassung: Führt eine ADC-Messung mit einer Mittelwertbildung aus 32 Einzelmessungen durch.

Kanal:

Kanal	Kanalbyte
Single-ended	
AIN00	0
AIN01	1
AIN02	2
AIN03	3
AIN04	4
AIN05	5
AIN06	6
AIN07	7
Differenzmessung	
AIN00+ / AIN01-	8
AIN00- / AIN01+	9
AIN02+ / AIN03-	10
AIN02- / AIN03+	11
AIN04+ / AIN05-	12
AIN04- / AIN05+	13
AIN06+ / AIN07-	14
AIN06- / AIN07+	15

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2.55V
4	+/-1.27V
5	+/- 0.63V

AD-Blockmessung mit Mittelwertbildung:

`int[]` GetADC_BlockMean([EXDUL.ADC_CHANNEL_CONFIG_1\[\]](#) *config*)

Parameter: *config*:

Rückgabewerte: Messwerte in μV

Zusammenfassung: Führt eine ADC-Blockmessung über mehrere Kanäle aus (siehe Handbuch)

Kanal:

Kanal	Kanalbyte
Single-ended	
AIN00	0
AIN01	1
AIN02	2
AIN03	3
AIN04	4
AIN05	5
AIN06	6
AIN07	7
Differenzmessung	
AIN00+ / AIN01-	8
AIN00- / AIN01+	9
AIN02+ / AIN03-	10
AIN02- / AIN03+	11
AIN04+ / AIN05-	12
AIN04- / AIN05+	13
AIN06+ / AIN07-	14
AIN06- / AIN07+	15

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1.27V
5	+/- 0.63V

ADC-FIFO Reset:

void ResetFIFO()

Zusammenfassung

Mit diesem Befehl wird ein Reset des FIFOs durchgeführt. Dies sollte nach einem Überlauf durchgeführt werden.

ADC-FIFO Overflowflag lesen:

bool ReadOverflowFlagFIFO()

Rückgabewert:

Overflowflag false = kein Overflow / true = Overflow

Zusammenfassung:

Liest das Overflowflag des ADC-FIFOs aus. Mit dem Auslesen wird das Flag automatisch zurückgesetzt

ADC-FIFO auslesen:

int[] ReadFIFO()

Rückgabewerte:

Gibt ein Array mit den Messwerten zurück. Die Größe des Arrays ist von der Messanzahl abhängig

Zusammenfassung:

Liest das ADC-FIFO aus

AD-Mehrfachmessung:

`int[] GetADC_Multi(ushort counts, uint samplerate, EXDUL_ADC_CHANNEL_CONFIG_1[] config)`

Parameter: *counts*: Anzahl der Messungen
samplerate: Abtastrate
config: Kanalkonfigurationen

Rückgabewerte: Messwert in μV

Zusammenfassung: Führt eine ADC-Mehrfachmessung über einen oder mehrere Kanäle durch. Die Messwerte können über die Funktion ReadFIFO abgeholt werden.

Kanal:

Messbereich:

Kanal	Kanalbyte	Bereichsbyte	Spannung
Single-ended		0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
AIN00	0		
AIN01	1	1	+/-10.2V
AIN02	2	2	+/- 5.1V
AIN03	3	3	+/-2,55V
AIN04	4	4	+/-1.27V
AIN05	5	5	+/- 0.63V
AIN06	6		
AIN07	7		
Differenzmessung			
AIN00+ / AIN01-	8		
AIN00- / AIN01+	9		
AIN02+ / AIN03-	10		
AIN02- / AIN03+	11		
AIN04+ / AIN05-	12		
AIN04- / AIN05+	13		
AIN06+ / AIN07-	14		
AIN06- / AIN07+	15		

AD-Dauerabtastung starten:

void StartADC([uint samplerate](#),
[EXDUL_ADC_CHANNEL_CONFIG 1\[\]](#)config)

Parameter: *samplerate*: Abtastrate
config: Kanalkonfiguration

Zusammenfassung Startet eine ADC-Dauerabtastung über einen oder mehrere Kanäle. Die Messwerte können über die Funktion ReadFIFO abgeholt werden. Zum Stoppen der Dauerabtastung wird die Funktion StopADC benötigt.

Kanal:

Kanal	Kanalbyte
Single-ended	
AIN00	0
AIN01	1
AIN02	2
AIN03	3
AIN04	4
AIN05	5
AIN06	6
AIN07	7
Differenzmessung	
AIN00+ / AIN01-	8
AIN00- / AIN01+	9
AIN02+ / AIN03-	10
AIN02- / AIN03+	11
AIN04+ / AIN05-	12
AIN04- / AIN05+	13
AIN06+ / AIN07-	14
AIN06- / AIN07+	15

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1.27V
5	+/- 0.63V

AD-Dauerabtastung stoppen:

void StopADC()

Zusammenfassung: Beendet eine ADC-Dauerabtastung

Temperatur messen:

[int](#) GetTemperature_PT([byte](#) channel)

Parameter: *channel*: Kanalauswahl

Rückgabewerte: Temperatur in °C * 100

Zusammenfassung: Misst die Temperatur durch den angeschlossenen PT-Sensor

Temperatur-Messeinheit kalibrieren:

void CalibrateTempUnit_PT([byte](#) channel)

Parameter: *channel*: Kanalauswahl

Zusammenfassung: Mit der Funktion kann die Messeinheit zur Temperaturmessung kalibriert werden. Dazu muss ein im Handbuch festgelegter Widerstand angelegt werden

Werksreset:

void DefaultReset()

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

11.4 Programmierung mit seriellen COM-Port-Libraries

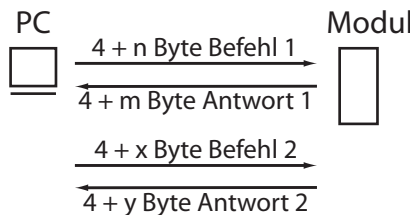
Durch die Möglichkeit mit Standard-COM-Port-Libraries auf das Modul zugreifen zu können, kann der Anwender mit einer Vielzahl an Sprachen seine Anwendung programmieren. So kann neben dem .NET-Framework auch Delphi oder C verwendet werden. Auch können Anwendungen auf vielen Linux basierten Betriebssystemen (falls ein virtueller COM-Port-Treiber vorhanden ist) entworfen werden.

11.4.1 Kommunikation mit dem EXDUL-392

Der Datenaustausch erfolgt durch Senden bzw. Empfangen von Byte-Arrays mit unterschiedlicher Länge über die virtuelle COM-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 11.2 Kommunikationsmodell

11.4.2 Windows[®]-Funktionen für die Programmierung

Die Programmierung des EXDUL-392E / EXDUL-392S erfolgt entweder über WIN32API Funktionen oder sehr komfortabel über ein bereits vorhandenes SerialPort Object in einer Programmiersprache. Beispielprogramme hierzu finden Sie nach der Installation der Software im Installationsverzeichnis auf Ihrem Rechner.

Windows-Funktionen für die Programmierung:

- CreateFile
- GetCommState
- SetCommState
- WriteFile
- ReadFile
- DCB-Struktur (beschreibt die Kontroll-Parameter des Devices)

11.4.3 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.

11.4.4 Befehlsübersicht

Hexcode	Beschreibung
0C 00 00	Inforegister lesen und schreiben
0C 00 03	LCD-Register lesen und schreiben
08 00 00	Optokopplerausgänge lesen und schreiben
08 00 01	Optokopplereingänge bearbeiten
0A 00 00	AD-Einzelmessung
0A 00 01	AD-Einzelmessung mit Mittelwertbildung
0A 00 02	AD-Blockmessung mit Mittelwertbildung
0A 00 06	AD-FIFO Reset
0A 00 07	AD-FIFO Overflowflag auslesen
0A 00 08	AD-FIFO auslesen
0A 00 09	AD-Mehrfachmessung
0A 00 0A	AD-Dauerabtastung starten
0A 00 0B	AD-Dauerabtastung stoppen
0A 04 00	Temperaturmesseinheit Temperaturmessung
0A 04 01	Temperaturmesseinheit Fehlererkennung durchführen
09 00 00	Zähler0

11.4.5 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-392 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	33		Daten 7. Zeichen 3 _{asci}
15	39		Daten 8. Zeichen 9 _{asci}
16	32		Daten 9. Zeichen 2 _{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-392“)

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			33	Daten 7. Zeichen z _{ASCII}
11			39	Daten 8. Zeichen g _{ASCII}
12			32	Daten 9. Zeichen a _{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	04	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			33	Daten 7. Zeichen 3 _{ascii}
11			39	Daten 8. Zeichen 9 _{ascii}
12			32	Daten 9. Zeichen 2 _{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	03	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-392 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	33		Daten 7. Zeichen 3 _{asci}
15	39		Daten 8. Zeichen 9 _{asci}
16	32		Daten 9. Zeichen 2 _{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-392 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 → 20 Byte	08	Längenbyte → 20 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			33	Daten Zeile1 7. Zeichen 3 _{asci}
11			39	Daten Zeile1 8. Zeichen 9 _{asci}
12			32	Daten Zeile1 9. Zeichen 2 _{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			33	Daten Zeile2 7. Zeichen 3 _{asci}
27			39	Daten Zeile2 8. Zeichen 9 _{asci}
28			32	Daten Zeile2 9. Zeichen 2 _{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-Modes

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-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	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 und 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		reserviert
11	00	reserviert		reserviert

Lesen LCD-Kontrastwert

Über diesen Befehl ist der Display-Kontrast auslesbar. Der Wert kann zwischen 0 und 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

Optokopplerausgang lesen

Dieser Befehl ermöglicht das Auslesen des aktuellen Zustands des Optokopplerausgangs

Beispiel: Auslesen des Optokopplerausgangszustands

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

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)	0w 00 (LOW an DIN0) 01 (HIGH an DIN0)	Zustand Optokopplerausgang
5	00	reserviert	00	reserviert
6	00	reserviert	00	reserviert
7	00	reserviert	00	reserviert

Optokopplerausgang schreiben

Dieser Befehl ermöglicht dem Anwender, den Ausgangsoptokoppler zu sperren oder durchzuschalten

Beispiel: Ausgabe eines Zustands am Optokopplerausgang

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

Byte	Senden	Empfangen	Beschreibung
0	08	08	Befehlscode 1. Byte
1	00	0	Befehlscode 2. Byte
2	00	00	Befehlscode 3. Byte
3	01 (→ 4Byte)	00	Längenbyte
4	00		r/w Byte
5	0w 00 (gesperrt) 01 (durchgeschaltet)		Optokopplerzustand
6	00		reserviert
7	00		reserviert

Optokopplereingang lesen

Dieser Befehl ermöglicht das Auslesen des aktuellen Zustands am Optokopplereingang

Beispiel: Auslesen des Zustands am Optokopplereingang

Gesendet wird ein 4Byte langer Block und empfangen ein 8Byte langer Block mit dem Zustand am Optokopplereingang

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	00	Befehlscode 3. Byte
3	00	Längenbyte	01 (→ 4Byte)	Längenbyte
4			0w	Zustand Optokopplereingang
5			00	reserviert
6			00	reserviert
7			00	reserviert

Zähler

Dieser Befehl ermöglicht den Zugriff auf den 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	Zähler starten
01	Zähler stoppen
02	Zähler resettet
03	Zählerstand lesen
04	reserviert
05	Overflow-Flag lesen
06	Overflow-Flag rücksetzen

Zähler 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	00	Befehlscode 3. Byte
3	01	Längenbyte	01	Längenbyte
4	bb 00 01 02	Zähler Befehlscode Zähler0 starten Zähler0 stoppen Zähler0 resetten	bb	Zähler Befehlscode
5	00	reserviert	00	reserviert
6	00	reserviert	00	reserviert
7	00	reserviert	00	reserviert

Zähler 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	00	Befehlscode 3. Byte
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 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	00	Befehlscode 3. Byte
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 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	00	Befehlscode 3. Byte
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

AD-Einzelmessung

Der Befehl AD-Einzelmessung führt an einem gewünschten analogen Eingangskanal eine Spannungsmessung/Strommessung durch und liefert den Wert abgeglichen im Integerformat in $\mu\text{V}/\mu\text{A}$ an den PC zurück. Dem Befehl muss der gewünschte Kanal sowie der Messbereich übergeben werden.

Kanal:

Kanal	Kanalbyte
Spannung Single-ended	
AINU0	0
AINU1	1
AINU2	2
AINU3	3
Spannung Differenzmessung	
AINU0+ / AINU1-	8
AINU0- / AINU1+	9
AINU2+ / AINU3-	10
AINU2- / AINU3+	11
Strommessung	
AINI0+ / AINI0-	12
AINI1+ / AINI1-	14

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1.27V
5	+/- 0.63V
Strom	
	+/-20mA

Beispiel zum Messen der Spannung an einem Eingangssignal

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	00	Befehlscode 3. Byte	00	Befehlscode 3. Byte
3	01 (→ 4Byte)		01 (→ 4Byte)	Längenbyte
4	cc	Kanalbyte	ww	Messwert Byte0
5	bb	Bereichsbyte	ww	Messwert Byte1
6	00		ww	Messwert Byte2
7	00		ww	Messwert Byte3

Spannung = (integer) (Byte3 * 0x1000000 + Byte2 * 0x10000 + Byte1 * 0x100 + Byte0) [μV]

Strom = (integer) (Byte3 * 0x1000000 + Byte2 * 0x10000 + Byte1 * 0x100 + Byte0) [μA]

AD-Einzelmessung mit Mittelwertbildung aus 32 Messungen

Der Befehl AD-Einzelmessung mit Mittelwertbildung führt an einem gewünschten analogen Eingangskanal 32 Spannungs- bzw. Strommessungen mit einer Geschwindigkeit von 100kS/s durch, bildet den Mittelwert und liefert den Wert abgeglichen im Integerformat in μV bzw. μA an den PC zurück.

Dem Befehl muss der gewünschte Kanal sowie der Messbereich übergeben werden.

Kanal:

Kanal	Kanalbyte
Spannung Single-ended	
AINU0	0
AINU1	1
AINU2	2
AINU3	3
Spannung Differenzmessung	
AINU0+ / AINU1-	8
AINU0- / AINU1+	9
AINU2+ / AINU3-	10
AINU2- / AINU3+	11
Strommessung	
AINI0+ / AINI0-	12
AINI1+ / AINI1-	14

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1.27V
5	+/- 0.63V
Strom	
	+/-20mA

Beispiel zum Messen der Spannung oder dem Strom an einem Eingangssignal

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	01	Befehlscode 3. Byte	01	Befehlscode 3. Byte
3	01 (→ 4Byte)	Längenbyte	01 (→ 4Byte)	Längenbyte
4	cc	Kanalbyte	ww	Messwert Byte0
5	bb	Bereichsbyte	ww	Messwert Byte1
6	00	reserviert	ww	Messwert Byte2
7	00	reserviert	ww	Messwert Byte3

Spannung = (integer) (Byte3 * 0x1000000 + Byte2 * 0x10000 + Byte1 * 0x100 + Byte0) [μV]

Strom = (integer) (Byte3 * 0x1000000 + Byte2 * 0x10000 + Byte1 * 0x100 + Byte0) [μA]

AD-Blockmessung mit Mittelwertbildung

Mit diesem Befehl können bis zu 8 Kanäle kurz hintereinander abgetastet werden. Dabei wird jeder zu messende Kanal 32mal abgetastet, jeweils ein Mittelwert (siehe Kapitel 5.3) gebildet und als Antwort im Integerformat in μV an den PC zurückgeschickt.

Kanal:

Kanal	Kanalbyte
Spannung Single-ended	
AINU0	0
AINU1	1
AINU2	2
AINU3	3
Spannung Differenzmessung	
AINU0+ / AINU1-	8
AINU0- / AINU1+	9
AINU2+ / AINU3-	10
AINU2- / AINU3+	11
Strommessung	
AINI0+ / AINI0-	12
AINI1+ / AINI1-	14

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V \rightarrow GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1.27V
5	+/- 0.63V
	Strom
	+/-20mA

Befehlsaufbau $n = 1 \dots 8$

Byte	Senden	Beschreibung
0	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte
2	02	Befehlscode 3. Byte
3	(n*4)	Längenbyte (n = Kanalanzahl)
4	00	reserviert
5	00	reserviert
6	C ₀ C ₀	Kanalbyte
7	b ₀ b ₀	Bereichsbyte
	:	
	:	
3 + n*4	C _{n-1} C _{n-1}	Kanalbyte
4 + n*4	b _{n-1} b _{n-1}	Bereichsbyte

Byte	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte
2	02	Befehlscode 3. Byte
3	(n*4)	Längenbyte (n = Kanalanzahl)
4	w ₁ w ₁	Messwert: Byte0 ₁
5	w ₁ w ₁	Messwert: Byte1 ₁
6	w ₁ w ₁	Messwert: Byte2 ₁
7	w ₁ w ₁	Messwert: Byte3 ₁
	:	
	:	
3 + n*4	w _n w _n	Messwert: Byte0 _n
4 + n*4 + 1	w _n w _n	Messwert: Byte1 _n
4 + n*4 + 2	w _n w _n	Messwert: Byte2 _n
4 + n*4 + 3	w _n w _n	Messwert: Byte3 _n

Beispiel:

In folgendem Beispiel sollen AINU1, AINU2 und AINI0 abgetastet werden. Der Messbereich soll bei den Spannungsmessungen +/- 10.2V und bei der Strommessung +/-20mA betragen

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	02	Befehlscode 3. Byte	02	Befehlscode 3. Byte
3	03 (→ 12Byte)	Längenbyte	03 (→ 12Byte)	Längenbyte
4	00	reserviert	w ₁ w ₁	Messwert AINU1 Byte0 ₁
5	00	reserviert	w ₁ w ₁	Messwert AINU1 Byte1 ₁
6	01	Kanalbyte AINU1	w ₁ w ₁	Messwert AINU1 Byte2 ₁
7	01	Bereichsbyte +/- 10.2V	w ₁ w ₁	Messwert AINU1 Byte3 ₁
8	00	reserviert	w ₂ w ₂	Messwert AINU2 Byte0 ₂
9	00	reserviert	w ₂ w ₂	Messwert AINU2 Byte1 ₂
10	0C	Kanalbyte AINU2	w ₂ w ₂	Messwert AINU2 Byte2 ₂
11	01	Bereichsbyte +/- 10.2V	w ₂ w ₂	Messwert AINU2 Byte3 ₂
12	00	reserviert	w ₃ w ₃	Messwert AINI0 Byte0 ₃
13	00	reserviert	w ₃ w ₃	Messwert AINI0 Byte1 ₃
14	04	Kanalbyte AINI0	w ₃ w ₃	Messwert AINI0 Byte2 ₃
15	03	Bereichsbyte +/- 20mA	w ₃ w ₃	Messwert AINI0 Byte3 ₃

Messwert AINU1 = (integer) (Byte3₁ * 0x1000000 + Byte2₁ * 0x10000 + Byte1₁ * 0x100 + Byte0₁) [µV]

Messwert AINU2 = (integer) (Byte3₂ * 0x1000000 + Byte2₂ * 0x10000 + Byte1₂ * 0x100 + Byte0₂) [µV]

Messwert AINI0 = (integer) (Byte3₃ * 0x1000000 + Byte2₃ * 0x10000 + Byte1₃ * 0x100 + Byte0₃) [µA]

ADC-FIFO Reset

Mit folgendem Befehl kann ein Reset des ADC-FIFOs durchgeführt werden. Dies sollte nach einem Überlauf erfolgen.

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	06	Befehlscode 3. Byte	06	Befehlscode 3. Byte
3	00	Längenbyte	00	Längenbyte → 0 Bytes

ADC-FIFO Overflowflag lesen

Mit folgendem Befehl kann das Overflowflag des ADC-FIFOs ausgelesen werden. Mit dem Auslesen wird das Overflowflag zurückgesetzt.

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	07	Befehlscode 3. Byte	07	Befehlscode 3. Byte
3	00	Längenbyte	01	Längenbyte → 4 Bytes
4			0w	Overflowflag 00 FIFO nicht übergelaufen 01 FIFO übergelaufen
5			00	reserviert
6			00	reserviert
7			00	reserviert

ADC-FIFO auslesen

Einige Befehle liefern die Messergebnisse nicht direkt mit dem Antwortbefehl zurück, sondern speichern die Messwerte in ein FIFO. Als Befehlsbeispiel sind die AD-Mehrfachmessung oder die Dauermessung zu nennen. Mit dem ADC-FIFO-Auslesebefehl kann das FIFO ausgelesen werden. Dabei werden die sich im FIFO befindlichen Werte direkt an die Antwort des Befehls angehängt (bis zu 255 Messwerte). Sind keine Daten im FIFO vorhanden, so wird nur eine 4Byte-Antwort an den PC zurückgeschickt.

Befehlsaufbau

Zu senden sind 4Byte, zu empfangen sind, je nach Datenmenge n im FIFO, $4 + n \cdot 4$ Bytes.

$n = 1 \dots 8$

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	08	Befehlscode 3. Byte	08	Befehlscode 3. Byte
3	00	Längenbyte	nn	Längenbyte → $n \cdot 4$ Bytes
4			ww ₁	Messwert ₁ Byte0 ₁
5			ww ₁	Messwert ₁ Byte1 ₁
6			ww ₁	Messwert ₁ Byte2 ₁
7			ww ₁	Messwert ₁ Byte3 ₁
			:	
			:	
$n \cdot 4$			ww _n	Messwert _n Byte0 _n
$n \cdot 4 + 1$			ww _n	Messwert _n Byte1 _n
$n \cdot 4 + 2$			ww _n	Messwert _n Byte2 _n
$n \cdot 4 + 3$			ww _n	Messwert _n Byte3 _n

Beispiel 1:
Im FIFO sind keine Daten vorhanden

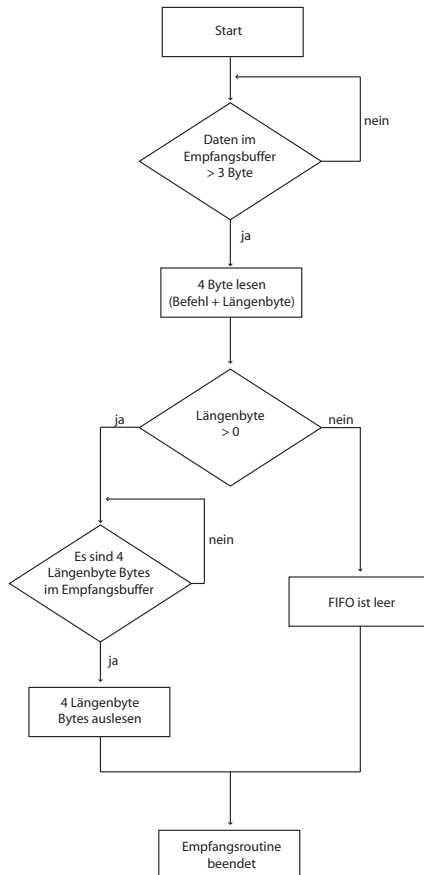
Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	08	Befehlscode 3. Byte	08	Befehlscode 3. Byte
3	00	Längenbyte	00	Längenbyte

Beispiel 2:
Im FIFO sind zwei Messwerte vorhanden

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	08	Befehlscode 3. Byte	08	Befehlscode 3. Byte
3	00	Längenbyte	2	Längenbyte → 8 Bytes
4			ww ₁	Messwert: Byte0 ₁
5			ww ₁	Messwert: Byte1 ₁
6			ww ₁	Messwert: Byte2 ₁
7			ww ₁	Messwert: Byte3 ₁
8			ww ₂	Messwert: Byte0 ₂
9			ww ₂	Messwert: Byte1 ₂
10			ww ₂	Messwert: Byte2 ₂
11			ww ₂	Messwert: Byte3 ₂

Programmierung:

- Senden: um Daten aus dem FIFO auszulesen, muss der 4Byte lange Befehl an das Modul gesendet werden.
- Empfangen der Daten: Da die Array-Länge der zu empfangenden Daten variieren kann, muss das Empfangen des gesamten Datenblocks aufgeteilt werden.



AD-Mehrfachmessung

Die AD-Mehrfachmessung ermöglicht dem Anwender, einen oder mehrere Kanäle in einem einstellbaren Takt (1 - 100000kHz) mehrfach abzutasten (bis zu 65535 mal). Die Messwerte werden vom Modul im internen FIFO abgelegt und können dort während und nach dem Abtastvorgang abgeholt werden. Die Werte werden solange im FIFO zwischengespeichert, bis sie entweder abgeholt wurden, oder ein neuer AD-Abtastbefehl aufgerufen wurde.

Achtung: Es muss sichergestellt werden, dass das FIFO schnell genug geleert werden kann, da das FIFO auf 10000 Messwerte begrenzt ist. Des Weiteren dürfen während des Vorgangs keine EXDUL-Info-Register (z.B. UserA, UserB) beschrieben werden.

Kanal:

Kanal	Kanalbyte
Spannung Single-ended	
AINU0	0
AINU1	1
AINU2	2
AINU3	3
Spannung Differenzmessung	
AINU0+ / AINU1-	8
AINU0- / AINU1+	9
AINU2+ / AINU3-	10
AINU2- / AINU3+	11
Strommessung	
AINI0+ / AINI0-	12
AINI1+ / AINI1-	14

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1,27V
5	+/- 0.63V
	Strom
	+/-20mA

Befehlsaufbau

n = 1 8

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	09	Befehlscode 3. Byte	09	Befehlscode 3. Byte
3	n + 2	Längenbyte	00	Längenbyte
4	ff	Abtastrate Byte0		
5	ff	Abtastrate Byte1		
6	ff	Abtastrate Byte2		
7	00	reserviert		
8	aa	Messanzahl Byte0		
9	aa	Messanzahl Byte1		
10	00	reserviert		
11	00	reserviert		
12	00	reserviert		
13	00	reserviert		
14	cc _n	Kanalbyte ₁		
15	bb _n	Bereichsbyte ₁		
	:			
	:			
n*4 + 8	00	reserviert		
n*4 + 9	00	reserviert		
n*4 + 10	cc _n	Kanalbyte _n		
n*4 + 11	bb _n	Bereichsbyte _n		

Abtastrate = Byte2 * 65536 + Byte1 * 256 + Byte0

Messanzahl = Byte1 * 256 + Byte0

AD-Dauerabtastung starten

Die AD-Dauerabtastung ermöglicht dem Anwender, einen oder mehrere Kanäle in regelmäßigen Zeitabständen (1s - 10µs) abzutasten. Die Messwerte werden vom Modul im internen FIFO abgelegt und können dort während und nach dem Abtastvorgang abgeholt werden. Die Werte werden solange im FIFO zwischengespeichert, bis sie entweder abgeholt wurden, oder ein neuer AD-Abtastbefehl aufgerufen wurde.

Um die Dauerabtastung zu stoppen, muss der Befehl „AD-Dauerabtastung stoppen“ an das Modul gesendet werden.

Achtung: Es muss sichergestellt werden, dass das FIFO schnell genug geleert werden kann, da das FIFO auf 10000 Messwerte begrenzt ist. Des Weiteren dürfen während des Vorgangs keine EXDUL-Info-Register (z.B. UserA, UserB) beschrieben werden.

Kanal:

Kanal	Kanalbyte
Spannung Single-ended	
AINU0	0
AINU1	1
AINU2	2
AINU3	3
Spannung Differenzmessung	
AINU0+ / AINU1-	8
AINU0- / AINU1+	9
AINU2+ / AINU3-	10
AINU2- / AINU3+	11
Strommessung	
AINI0+ / AINI0-	12
AINI1+ / AINI1-	14

Messbereich:

Bereichsbyte	Spannung
0	+/- 20.4V (nur bei Differenzmessung max +/- 10.2V → GND)
1	+/-10.2V
2	+/- 5.1V
3	+/-2,55V
4	+/-1.27V
5	+/- 0.63V
Strom	
	+/-20mA

Befehlsaufbau

n = 1 8

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	0A	Befehlscode 3. Byte	0A	Befehlscode 3. Byte
3	n + 1	Längenbyte	00	Längenbyte
4	ff	Abtastrate Byte0		
5	ff	Abtastrate Byte1		
6	ff	Abtastrate Byte2		
7	00	reserviert		
8	aa	reserviert		
9	aa	reserviert		
10	cc ₁	Kanalbyte ₁		
11	bb ₁	Bereichsbyte ₁		
	:			
	:			
n*4 + 4	00	reserviert		
n*4 + 5	00	reserviert		
n*4 + 6	cc _n	Kanalbyte _n		
n*4 + 7	bb _n	Bereichsbyte _n		

$$\text{Abtastrate} = \text{Byte2} * 65536 + \text{Byte1} * 256 + \text{Byte0}$$

AD-Dauerabtastung stoppen

Mit diesem Befehl wird die AD-Dauerabtastung gestoppt.

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	00	Befehlscode 2. Byte	00	Befehlscode 2. Byte
2	0B	Befehlscode 3. Byte	0B	Befehlscode 3. Byte
3	00	Längenbyte	00	Längenbyte

Temperaturmessung

Mit diesem Befehl können Temperaturmessungen an den Temperaturmesseinheiten durchgeführt werden. Dabei muss der Kanal sowie die Art der Messung (z.B. Widerstandsmessung oder Temperaturmessung mit PT100 IEC 751 Sensor) mit an das Modul gesendet werden.

Kanal:

Kanal	Kanalbyte
TIN0	0
TIN1	1
TIN2	2

Messbereich:

Messfunktionsbyte	Funktion
0	Widerstandsmessung in mΩ
1	Temperaturmessung PT100 IEC 751

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	04	Befehlscode 2. Byte	04	Befehlscode 2. Byte
2	00	Befehlscode 3. Byte	00	Befehlscode 3. Byte
3	01	Längenbyte → 4 Bytes	02	Längenbyte → 8 Bytes
4	cc	Kanalbyte	cc	Kanalbyte
5	mm	Messfunktionsbyte	00	reserviert
6	00	reserviert	00	reserviert
7	00	reserviert	00	reserviert
8			ww	Messwert Byte0
9			ww	Messwert Byte1
10			ww	Messwert Byte2
11			ww	Messwert Byte3

Widerstand = (integer) (Byte3 * 0x1000000 + Byte2 * 0x10000 + Byte1 * 0x100 + Byte0) [mΩ]

Temperatur = (integer) (Byte3 * 0x1000000 + Byte2 * 0x10000 + Byte1 * 0x100 + Byte0) [°C * 10²]

Temperaturmesseinheit Fehlererkennung durchführen

Mit diesem Befehl kann eine Fehlererkennung an den Temperaturmesseinheiten durchgeführt werden. Dies kann einige ms benötigen. Als Rückantwort wird ein Fehlerbyte an den PC gesendet. Die Bedeutung der einzelnen Bits wird in Kapitel 6.3 erörtert

Byte	Senden	Beschreibung	Empfangen	Beschreibung
0	0A	Befehlscode 1. Byte	0A	Befehlscode 1. Byte
1	04	Befehlscode 2. Byte	04	Befehlscode 2. Byte
2	01	Befehlscode 3. Byte	00	Befehlscode 3. Byte
3	01	Längenbyte → 4 Bytes	02	Längenbyte → 8 Bytes
4	cc	Kanalbyte	cc	Kanalbyte
5	00	reserviert	00	reserviert
6	00	reserviert	00	reserviert
7	00	reserviert	00	reserviert
8			ww	Fehlerbyte
9			00	reserviert
10			00	reserviert
11			00	reserviert

11.5 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.

Für genauere Informationen lesen Sie sich bitte das EXDUL-LabVIEW-Tutorial durch.

12. Programmierung unter Linux[®]

12.1 Einführung

Nach dem erfolgreichen Erkennen des EXDUL-381E / EXDUL-381S durch das Betriebssystem wird das Modul im Ordner /dev als ttyACM* Gerät gelistet. Es handelt sich hierbei um ein CDC-Device (Communications Device Class), das über einen virtuellen COM-Port angesprochen wird. Der Softwarezugriff auf diesen virtuellen COM-Port erfolgt wie über eine normale COM-Schnittstelle über einen Standard-Treiber, eine Installation eines zusätzlichen Treibers ist nicht notwendig.

12.2 Programmierung mit seriellen COM-Port-Libraries

Wurde das Modul erkannt, kann über die Standard Libraries für die serielle Schnittstellen mit ihm kommuniziert werden. Für genauere Informationen lesen Sie ab Kapitel 10.4 weiter.

13. Technische Daten

A/D-Eingänge Spannung

4 Eingänge single-ended (se)
oder 2 Eingänge differentiell (diff)
oder kombiniert se/diff per SW wählbar
Auflösung: 16 Bit

Eingangsspannungsbereich bipolar

+/-0,63 Volt, +/-1,27 Volt, +/-2,55 Volt, +/-5.1 Volt, +/-10.2 Volt
+/-20.4V (nur Differenzeingänge)

FIFO: 10000 Messwerte

Überspannungsschutz: +/- 50V

Messzyklus: max. 10 μ s

Abtastrate: max 100 kS/s

A/D-Eingänge Strom

2 Eingänge

Auflösung 15 Bit

Messbereich +/-20mA

Abgeglichen auf 4..20mA

Eingangswiderstand 120 Ω

FIFO: 10000 Messwerte

max. Abtastrate: 100kHz

Digitaler Eingang über Optokoppler

1 bipolarer Kanal

Überspannungsschutz-Dioden

Eingangsspannungsbereich

high = 10..30 Volt

low = 0..3 Volt

Digitaler Ausgang über Optokoppler

1 Kanal

Leistungsoptokoppler

Verpolungsschutz-Dioden

Ausgangsstrom: max. 150 mA

Spannung-CE: max. 50 V

3 Temperaturmessenheiten PT100

3-Leiteranschluss

Zähler

1 programmierbarer Zähler 32 Bit (belegt den Optokoppler-Eingang)
Zählfrequenz: max. 5 kHz

LCD Anzeige (nur EXDUL-392E)

Matrixanzeige mit 2 Zeilen und 16 Spalten zur Darstellung von 16 Zeichen je Zeile
Programmierbar zur Darstellung anwendungsspezifischer Daten oder als I/O-Zustandsanzeige

USB-Schnittstelle

USB 2.0 kompatibel
USB-Anschluss Plug&Play (hotpluggable, auch im laufenden Betrieb anschließbar)

Modul-Anschlüsse

1 * 24polige Schraubklemmleiste
1 * USB-Buchse Typ B

Abmessungen

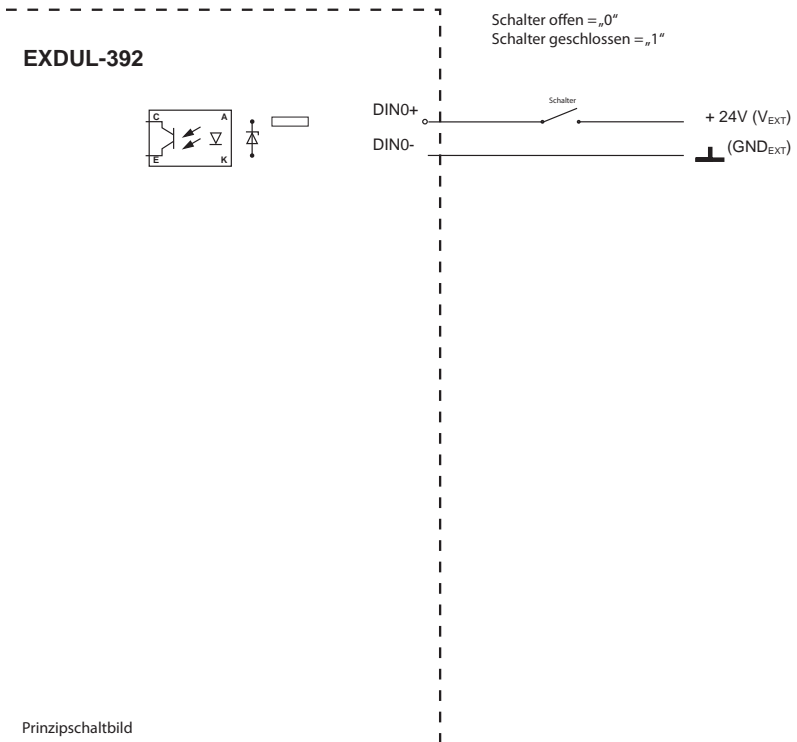
105 mm x 89 mm x 59 mm (l x b x h)

Gehäuse

Isolierstoffgehäuse mit integrierter Schnapptechnik zur DIN EN Hut-schienen-Montage
Geeignet für Aufbaumontagen, Schaltschrank- und Verteilereinbau sowie für mobile Tischeinsätze

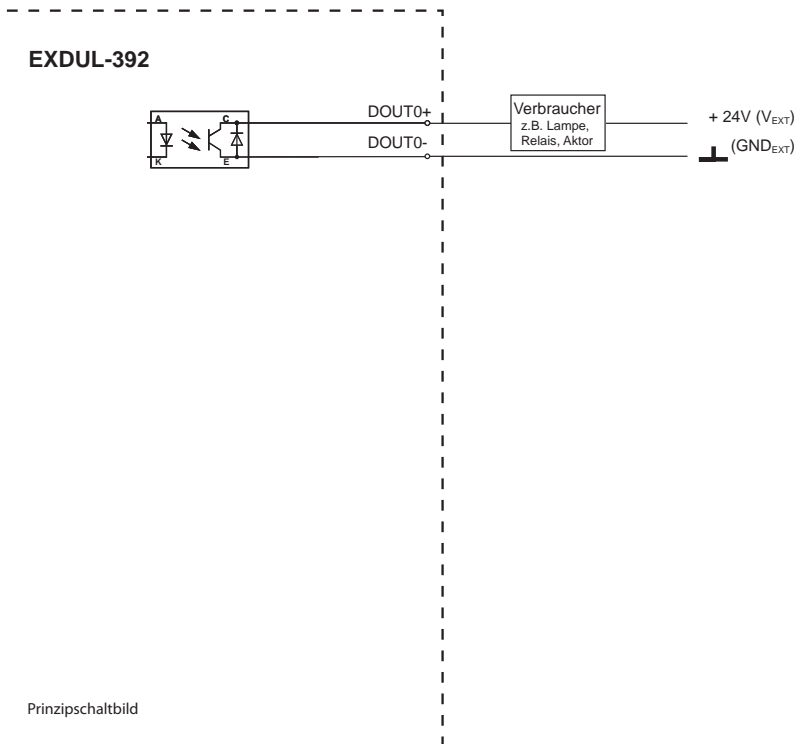
14. Beschaltungsbeispiele

14.1 Beschaltung des Optokoppler-Eingangs



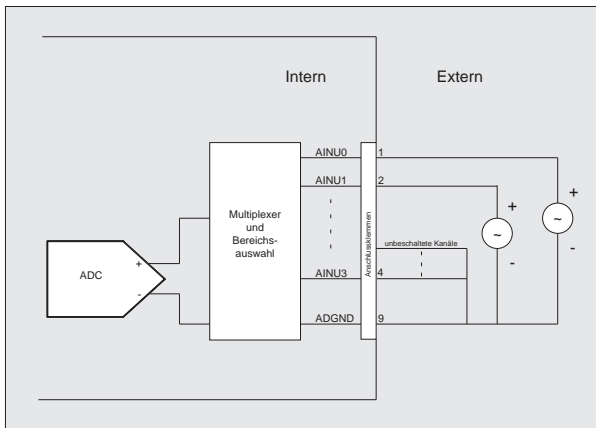
Grafik 14.1 Beschaltung des Optokopplereingangs

14.2 Beschaltung des Optokoppler-Ausgangs



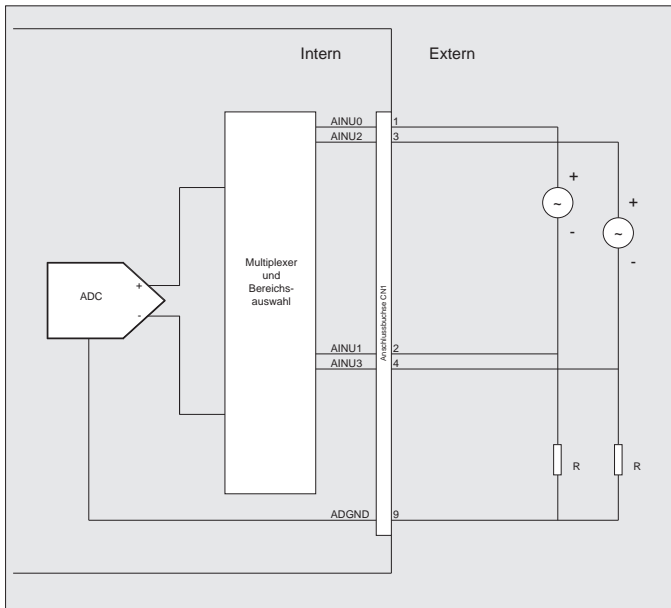
Grafik 14.2 Beschaltung des Optokoppler-Ausgangs

14.3 Beschaltung der A/D-Eingänge single-ended (Spannung)



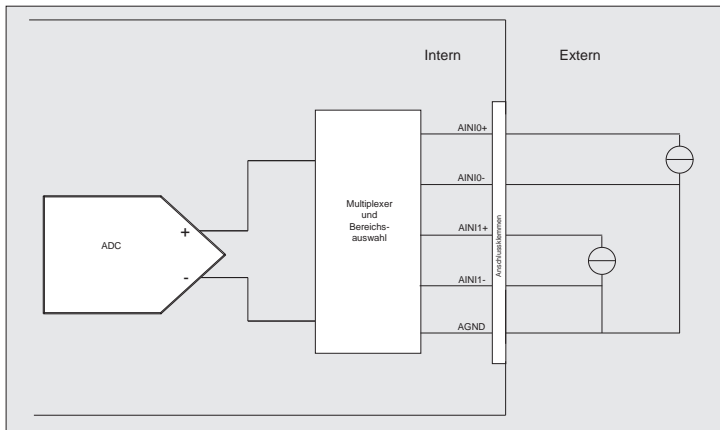
Grafik 14.3 Beschaltung der AD-Eingänge (Single-ended)

14.4 Beschaltung der A/D-Eingänge differentiell (Spannung)



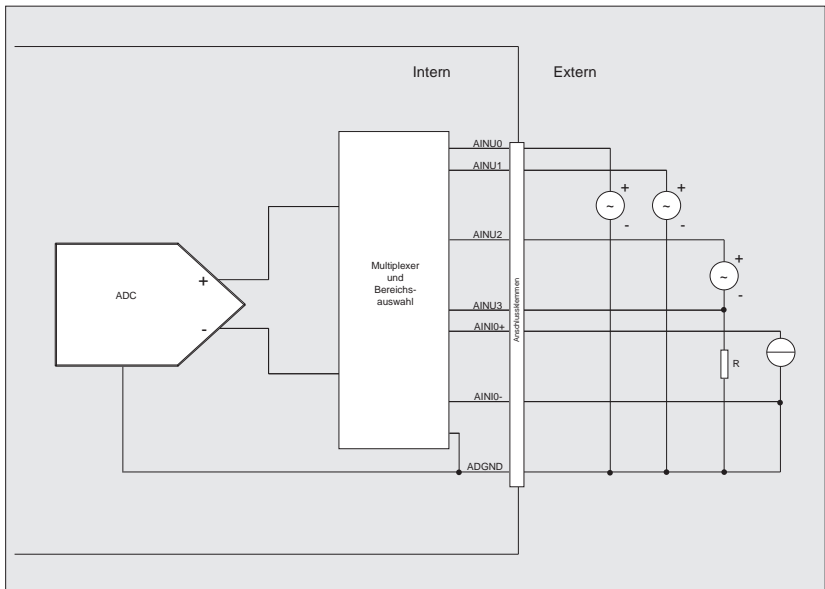
Grafik 14.4 Beschaltung der AD-Eingänge (differenziell)

14.5 Beschaltung der A/D-Eingänge Strommessung



Grafik 14.5 Beschaltung der AD-Eingänge (Strommessung)

14.6 Beschaltung der A/D-Eingänge Spannung-/Strommessung



Grafik 14.6 Beschaltung der AD-Eingänge (Strom/Spannungsmessung)

15. 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	

16. 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.

17. EG-Konformitätserklärung

Für die Erzeugnisse

EXDUL-392E EDV-Nummer A-382220
EXDUL-392S EDV-Nummer A-382210

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-392E und EXDUL-392S sind nichtselbstständig betreibbare Geräte, dessen 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-392E EXDUL-392S	A-382220 A-382210