



Übertragungsprotokolle + Registerbelegung

2023 März

INDEX

<u>1. Einleitung</u>	7
1.1. Was sind überhaupt Register ?	8
1.2. Registerzugriff mit 8/ 16/ 32 oder 64 Bit-Datenbreite	8
<u>2. Registerbelegung</u>	10
2.1. Allgemeines	11
2.1.1. Lesen der Anzahl von Kanälen	11
2.2. Ansteuerung Ein-/Ausgänge	12
2.2.1. Register Übersicht	12
2.2.2. Register für digitale Eingänge	13
2.2.2.1. Lesen der Eingänge	13
2.2.2.2. Eingangszustandsänderungen lesen	14
2.2.2.3. Eingangszähler lesen	16
2.2.2.4. Eingangszähler lesen und zurücksetzen	17
2.2.2.5. Modus Counter-Modul (nur bei O8-R8)	18
2.2.2.6. Digitaler Eingangsfiler	19
2.2.3. Register für digitale Ausgänge	20
2.2.3.1. Ausgänge setzen und zurücksetzen	20
2.2.3.2. Ausgänge zurücklesen	23
2.2.3.3. Ausgänge setzen	24
2.2.3.4. Ausgänge zurücksetzen	27
2.2.4. Register für digitale PWM Ausgänge (RO-PWM)	29
2.2.4.1. Beispiele für das Setzen eines PWM Ausgangs	30
2.2.5. Register für 48-Bit Zähler Eingänge	32
2.2.5.1. 48-Bit Eingangszähler Adressbereich	32
2.2.5.1.1. 48-Bit Eingangszähler Register	33
2.2.5.1.1.1, Zähler Modus	34
2.2.5.1.1.1, Submodi für Zähler Modus COUNT_RISING_EDGE	35
2.2.5.1.1.2, Eingangsfiler	36
2.2.5.1.2. 48-Bit Eingangszähler Latchregister (Komplett für ein RO-CNT8 Modul)	37
2.2.6. Register für Pulsgenerator Ausgänge (für RO-CNT8 Modul)	38
2.2.6.1. Pulsgenerator Ausgang Adressbereich	38

INDEX

2.2.6.1.1. Pulsgenerator Ausgang Register	39
2.2.6.1.2. Beispiele für das Setzen eines Pulsgenerator Ausgangs	40
2.2.7. Register für die Richtungssetzung der Ein-/Ausgänge	44
2.2.7.1. Byteweise Richtungssetzung der Ein-/Ausgänge	44
2.2.7.2. Bitweise Richtungssetzung der Ein-/Ausgänge	47
2.2.8. Register für analoge Eingänge (A/D)	50
2.2.8.1. A/D Wandler Adressbereich	50
2.2.8.2. A/D Wandler Register	52
2.2.8.2.1. A/D - Modus	52
2.2.9. Register für Temperatur Eingänge	53
2.2.9.1. Temperatur Kanal Adressbereich	53
2.2.9.1.1. Temperatur Eingang Register	55
2.2.9.1.2. Beispiel für das Abfragen eines Temperatur Eingangs	56
2.2.10. Register für analoge Ausgänge (D/A)	57
2.2.10.1. D/A Wandler Adressbereich	57
2.2.10.2. D/A Wandler Register	59
2.2.10.2.1. D/A - Modus	60
2.2.11. Register für Stepper	61
2.2.11.1. Stepper Adressbereich	61
2.2.11.2. Stepper-Kommandos	62
2.2.11.2.1. Command-Liste	64
2.2.11.2.2. Erklärung zu par 1, 2 und 3	65
2.2.11.3. Stepper-Status lesen	66
2.2.11.3.1. Erklärung des Befehls GET_POSITION	66
2.2.11.3.2. Erklärung der Bits für Befehl GET_ACTIVITY	67
2.2.11.3.3. Erklärung der Bits für Befehl GET_SWITCH	68
2.2.12. Timeout-Register für digitale und analoge Ausgänge (D/A)	69
2.2.12.1. Timeout-Register Command setzen	71
2.2.12.2. Timeout-Register Status lesen	72
2.2.12.3. Timeout Register Wert schreiben	73
3. CAN Protokoll	74
3.1. Geschwindigkeit und Schnittstellenparameter	75

INDEX

3.2. Aufbau „Sendepaket“	75
3.2.1. COMMAND Write	75
3.2.2. COMMAND Read	76
3.2.3. ADRESSE	77
3.2.4. JOB-ID	77
3.2.5. DATA	77
3.3. Beispiel für ein Schreibbefehl:	78
3.4. Beispiel für das Beschreiben der ersten 16 Relais	79
3.5. Aufbau „Antwortpaket“	80
3.5.1. „OK“ - Antwortpaket	80
3.5.2. „Fehler“-Antwortpaket	81
3.5.2.1. Fehlercode „0xfd“ – JOB-ID doppelt	81
3.5.2.2. Fehlercode „0xfe“ – ungültiger Befehl	82
3.6. Aufbau der CAN-Pakete TX/RX Modus	83
3.6.1. Digitale Eingänge	83
3.6.2. Digitale Ausgänge	84
3.6.3. Digitale Eingangszähler (16-Bit)	85
3.6.4. Digitale Eingangszähler (48-Bit) - 32-Bit Paket	86
3.6.5. Digitale Eingangszähler (48-Bit) - 64-Bit Paket	87
3.6.6. Analoge Ein- / Ausgänge	88
3.6.6.1. Analoge Eingänge	88
3.6.6.2. Analoge Ausgänge	90
3.6.6.3. Beispiele	91
3.6.7. Temperatur Eingänge	94
3.6.8. Stepper	96
3.6.8.1. Command-Liste 2	96
3.6.8.2. Werte für par 1 zu Befehl SET_MOTORCHARACTERISTIC	98
3.6.8.3. Werte für par 1 zu Befehl GO_REFSWITCH	100
3.6.8.4. Beispiel	101
3.6.9. PWM Ausgänge	102

INDEX

<u>4. Ethernet (TCP) Protokoll</u>	103
4.1. Das TCP-Protokoll	104
4.1.1. IP-Adresse	104
4.1.2. Netzmaske	104
4.1.3. Port	104
4.2. TCP Sende/Empfangsstring	104
4.2.1. Aufbau des "Sendestrings"	104
4.2.1.1. JOB-ID	107
4.2.1.2. Länge des Sendestrings	108
4.2.1.3. Extended Address Mode	108
4.2.1.4. COMMAND	108
4.2.1.5. WIDTH	109
4.2.1.6. ADDRESS	110
4.2.1.7. DATA	110
4.2.2. Aufbau des „Antwortstrings“	111
4.2.2.1. Fehlercode	111
4.2.2.2. JOB-ID	112
4.2.2.3. Länge des Antwortstrings	112
4.2.2.4. DATA	112
<u>5. Serielles Protokoll</u>	114
5.1. Geschwindigkeit und Schnittstellenparameter	115
5.2. Aufbau des „Sendestrings“	115
5.2.1. Anfangszeichen	116
5.2.2. MODUL-NR	116
5.2.3. JOB-ID	117
5.2.4. COMMAND	117
5.2.5. WIDTH	118
5.2.6. ADDRESS	118
5.2.7. DATA	119
5.2.7.1. CHECKSUM	119
5.2.8. END	119

INDEX

5.3. Aufbau des „Antwortstrings“	120
5.3.1. Antwortstring „O“ (OK)	120
5.3.2. Antwortstring „D“ (DATA)	122
5.3.3. Startzeichen mit Antwort „E“ (ERROR)	124
<u>6. Anhang</u>	125
6.1. Revisionen	126
6.2. Urheberrechte und Marken	127

Einleitung



1. Einleitung

1.1. Was sind überhaupt Register ?

Register sind kleine Zwischenspeicher, die die zu schreibenden oder zu lesenden Daten zwischenpuffern. Die Daten verbleiben solange gespeichert, bis sie überschrieben werden oder deren Stromversorgung abgeschaltet wird. Sie besitzen einen Adressbereich um sie anzusprechen. Mit Hilfe der Adresse kann man von Registern lesen oder auf ihnen schreiben. Der Registerzugriff erfolgt also adressiert.

Eine Besonderheit bilden die Register der Eingangszustandsänderungs-Merker. Werden diese ausgelesen, so werden deren Daten zugleich zurückgesetzt.

1.2. Registerzugriff mit 8/ 16/ 32 oder 64 Bit-Datenbreite

Der Registerzugriff kann in unterschiedlicher Datenbreite erfolgen. Die Datenbreite kann wahlweise 8 (Byte), 16 (Word), 32 (Long) oder 64 (eXtralong) Bit breit sein. Mit der Adresse wird der Bereich ausgewählt, auf dem man Zugriff erlangen möchte. Eine Adresse weist auf 8 Bits.

Erfolgt beispielsweise ein 32 Bit-Zugriff auf Adr. 0004, so werden 4 x 8 Byte (als ein Datenblock) beginnend mit Adresse 0004 bis Adresse 0007 ausgelesen bzw. beschrieben.

Tabelle mit Aufteilung in 8,16,32,64 Bit Registerzugriff

Zugriffs- breite	Adresse	00	01	02	03	04	05	06	07
8 Bit	0000 hex	X							
16 Bit	0000 hex	X	X						
32 Bit	0000 hex	X	X	X	X				
32 Bit	0004 hex					X	X	X	X
64 Bit	0000 hex	X	X	X	X	X	X	X	X

Beim Schreiben sowie beim Lesen werden die Daten Byte für Byte beschrieben bzw. ausgelesen. Dabei ist auf die Byte-Reihenfolge zu achten. Begonnen wird mit den niederwertigen Bytes (Byteorder: Little Endian).

Byte-Reihenfolge der Daten im Register

Zugriffs- breite	Adresse	Wert	00	01	02	03	04	05	06	07
8 Bit	04	1a					1a			
16 Bit	06	1a1b							1b	1a
32 Bit	00	01020304	04	03	02	01				
32 Bit	04	01020304					04	03	02	01
64 Bit	00	0102030405060708	08	07	06	05	04	03	02	01

Registerbelegung



2. Registerbelegung

2.1. Allgemeines

Hier finden Sie hilfreiche Statusregister der Grundkonfiguration. Hier kann die Hardwareausstattung des Moduls gelesen werden.

2.1.1. Lesen der Anzahl von Kanälen

Adresse	r/w	Bit	Beschreibung
FF00 hex	Read	16	Lesen der Anzahl von digitalen Ausgangskanälen
FF02 hex	Read	16	Lesen der Anzahl von digitalen Eingangskanälen
FF04 hex	Read	16	Lesen der Anzahl von digitalen Ein-/Ausgangskanälen
FF06 hex	Read	16	Lesen der Anzahl von analogen Ausgangskanälen
FF08 hex	Read	16	Lesen der Anzahl von analogen Eingangskanälen
FF0A hex	Read	16	Lesen der Anzahl von Stepperkanälen
FF0E hex	Read	16	Lesen der Anzahl von Eingangszählern
FF10 hex	Read	16	Lesen der Anzahl von Temperaturkanälen
FF12 hex	Read	16	Lesen der Anzahl von 48-Bit Eingangszählern
FF14 hex	Read	16	Lesen der Anzahl von Pulsgenerator Ausgangskanälen
FF16 hex	Read	16	Lesen der Anzahl von PWM Ausgangskanälen

2.2. Ansteuerung Ein-/Ausgänge

2.2.1. Register Übersicht

Start- adresse [hex]	End- adresse [hex]	Beschreibung
0000	000F	Register für digitale Ausgänge
0020	002F	Register für digitale Eingänge
0040	004F	Register für digitale Eingangszustandsänderung
0100	0101	Register für Richtungssetzung (Byteweise)
0100	011F	Register für digitale Eingangszähler
0120	012F	Register für Richtungssetzung (Bitweise)
01F0	01F7	Register für digitale Eingangszählermodi (RO-O8-R8)
0200	021F	Register für digitale Eingangszähler (RO-O8-R8)
0800	087F	Register für digitale PWM-Ausgänge
1000	11FF	Register für analoge Eingänge
2000	21FF	Register für analoge Ausgänge
3000	31FF	Register für Stepper
4000	40FF	Register für Temperatur Eingänge
5000	507F	Register für 48 Bit Eingangszähler
5800	58FF	Register für Pulsgenerator Ausgänge (RO-CNT8)

2.2.2. Register für digitale Eingänge

Die logischen Zustände der Spannungspegel an den Eingangsmodulen werden über die Register beginnend ab der Adresse 0020 hex ausgelesen.

2.2.2.1. Lesen der Eingänge

Adresse [hex]	r/w	Bit	Beschreibung
0020	Read	0-7	Lesen der Eingänge 1-8
0021	Read	0-7	Lesen der Eingänge 9-16
0022	Read	0-7	Lesen der Eingänge 17-24
0023	Read	0-7	Lesen der Eingänge 25-32
0024	Read	0-7	Lesen der Eingänge 33-40
0025	Read	0-7	Lesen der Eingänge 41-48
0026	Read	0-7	Lesen der Eingänge 49-56
0027	Read	0-7	Lesen der Eingänge 57-64
0028	Read	0-7	Lesen der Eingänge 65-72
0029	Read	0-7	Lesen der Eingänge 73-80
002A	Read	0-7	Lesen der Eingänge 81-88
002B	Read	0-7	Lesen der Eingänge 89-96
002C	Read	0-7	Lesen der Eingänge 97-104
002D	Read	0-7	Lesen der Eingänge 105-112
002E	Read	0-7	Lesen der Eingänge 113-120
002F	Read	0-7	Lesen der Eingänge 121-128

2.2.2.2. Eingangszustandsänderungen lesen

Adresse [hex]	r/w	Bit	Beschreibung
0040	Read	0-7	Lesen der Eingangszustandsänderung 1-8
0041	Read	0-7	Lesen der Eingangszustandsänderung 9-16
0042	Read	0-7	Lesen der Eingangszustandsänderung 17-24
0043	Read	0-7	Lesen der Eingangszustandsänderung 25-32
0044	Read	0-7	Lesen der Eingangszustandsänderung 33-40
0045	Read	0-7	Lesen der Eingangszustandsänderung 41-48
0046	Read	0-7	Lesen der Eingangszustandsänderung 49-56
0047	Read	0-7	Lesen der Eingangszustandsänderung 57-64
0048	Read	0-7	Lesen der Eingangszustandsänderung 65-72
0049	Read	0-7	Lesen der Eingangszustandsänderung 73-80

Adresse [hex]	r/w	Bit	Beschreibung
004A	Read	0-7	Lesen der Eingangszustandsänderung 81-88
004B	Read	0-7	Lesen der Eingangszustandsänderung 89-96
004C	Read	0-7	Lesen der Eingangszustandsänderung 97-104
004D	Read	0-7	Lesen der Eingangszustandsänderung 105-112
004E	Read	0-7	Lesen der Eingangszustandsänderung 113-120
004F	Read	0-7	Lesen der Eingangszustandsänderung 121-128

ACHTUNG!

Beim Lesen der Eingangszustandsänderungs-Merker werden diese direkt zurückgesetzt (auf 0)

Die Register der Eingangszustandsänderungs-Merker beginnen ab Adresse 0040 hex.

2.2.2.3. Eingangszähler lesen

Adresse [hex]	r/w	Bit	Beschreibung
0100	Read	16	Lesen des Eingangszählers von Eingang 1
0102	Read	16	Lesen des Eingangszählers von Eingang 2
0104	Read	16	Lesen des Eingangszählers von Eingang 3
0106	Read	16	Lesen des Eingangszählers von Eingang 4
0108	Read	16	Lesen des Eingangszählers von Eingang 5
010A	Read	16	Lesen des Eingangszählers von Eingang 6
010C	Read	16	Lesen des Eingangszählers von Eingang 7
010E	Read	16	Lesen des Eingangszählers von Eingang 8
0110	Read	16	Lesen des Eingangszählers von Eingang 9
0112	Read	16	Lesen des Eingangszählers von Eingang 10
0114	Read	16	Lesen des Eingangszählers von Eingang 11
0116	Read	16	Lesen des Eingangszählers von Eingang 12
0118	Read	16	Lesen des Eingangszählers von Eingang 13
011A	Read	16	Lesen des Eingangszählers von Eingang 14
011C	Read	16	Lesen des Eingangszählers von Eingang 15
011E	Read	16	Lesen des Eingangszählers von Eingang 16

2.2.2.4. Eingangszähler lesen und zurücksetzen

Adresse [hex]	r/w	Bit	Beschreibung
0200	Read	16	Lesen / zurücksetzen des Zählers von Eingang 1
0202	Read	16	Lesen / zurücksetzen des Zählers von Eingang 2
0204	Read	16	Lesen / zurücksetzen des Zählers von Eingang 3
0206	Read	16	Lesen / zurücksetzen des Zählers von Eingang 4
0208	Read	16	Lesen / zurücksetzen des Zählers von Eingang 5
020A	Read	16	Lesen / zurücksetzen des Zählers von Eingang 6
020C	Read	16	Lesen / zurücksetzen des Zählers von Eingang 7
020E	Read	16	Lesen / zurücksetzen des Zählers von Eingang 8
0210	Read	16	Lesen / zurücksetzen des Zählers von Eingang 9
0212	Read	16	Lesen / zurücksetzen des Zählers von Eingang 10
0214	Read	16	Lesen / zurücksetzen des Zählers von Eingang 11
0216	Read	16	Lesen / zurücksetzen des Zählers von Eingang 12
0218	Read	16	Lesen / zurücksetzen des Zählers von Eingang 13
021A	Read	16	Lesen / zurücksetzen des Zählers von Eingang 14
021C	Read	16	Lesen / zurücksetzen des Zählers von Eingang 15
021E	Read	16	Lesen / zurücksetzen des Zählers von Eingang 16

2.2.2.5. Modus Counter-Modul (nur bei 08-R8)

Adresse [hex]	r/w	Bit	Beschreibung
01F0	r/w	8	Modus des Counter-Modules 0 lesen/schreiben
01F1	r/w	8	Modus des Counter-Modules 1 lesen/schreiben
01F2	r/w	8	Modus des Counter-Modules 2 lesen/schreiben
01F3	r/w	8	Modus des Counter-Modules 3 lesen/schreiben
01F4	r/w	8	Modus des Counter-Modules 4 lesen/schreiben
01F5	r/w	8	Modus des Counter-Modules 5 lesen/schreiben
01F6	r/w	8	Modus des Counter-Modules 6 lesen/schreiben
01F7	r/w	8	Modus des Counter-Modules 7 lesen/schreiben

Werte für Counter - Modus

Wert	Zähler	Modus
0	8 x 16 Bit	8 Zähler (zählt nur hoch)
1	4 x 16 Bit up/down	4 Zähler für Encoder (zählt hoch & runter)

2.2.2.6. Digitaler Eingangsfiler

Adresse [hex]	r/w	Bit	Beschreibung
FD10	r/w	8	FlipFlop Eingangsfiler lesen/schreiben
FD11	r/w	8	Digitaler Eingangsfiler lesen/schreiben

Der übertragene Wert entspricht der Zeit in Millisekunden.

Im Handbuch der DELIB Treiberbibliothek finden Sie am Ende von Kapitel 2 eine Übersichtstabelle.

Diese zeigt, welches Modul die oben genannte Funktion unterstützt und welche Werte gültig sind.

2.2.3. Register für digitale Ausgänge

2.2.3.1. Ausgänge setzen und zurücksetzen

Adresse	r/w	Bit	Beschreibung
0000 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 1-8
0001 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 9-16
0002 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 17-24
0003 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 25-32
0004 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 33-40
0005 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 41-48
0006 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 49-56
0007 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 57-64
0008 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 65-72
0009 hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 73-80

Adresse	r/w	Bit	Beschreibung
000A hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 81-88
000B hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 89-96
000C hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 97-104
000D hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 105-112
000E hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 113-120
000F hex	Write	0-7	Setzen/Zurücksetzen der Ausgänge 121-128

Dieses Register dient zum Setzen und Zurücksetzen von digitalen Ausgängen. Hat das Bit einen Wert von 0 wird der Ausgang zurückgesetzt, beträgt der Wert 1 wird der Ausgang gesetzt.

Beispiel:

Auf Adresse 0x0000 wird der Wert 0x02 geschrieben. Kanal 2 wird gesetzt, alle anderen Kanäle werden zurückgesetzt.

Auf Adresse 0x0000 wird der Wert 0x03 geschrieben. Kanal 1 und 2 werden gesetzt, alle anderen Kanäle werden zurückgesetzt.

Auf Adresse 0x0000 wird der Wert 0x04 geschrieben. Kanal 3 wird gesetzt, alle anderen Kanäle werden zurückgesetzt.

Auf Adresse 0x0000 wird der Wert 0x08 geschrieben. Kanal 4 wird gesetzt, alle anderen Kanäle werden zurückgesetzt.

Falls Sie Kanäle nur setzen oder zurücksetzen möchten, benutzen Sie bitte folgende Register:

Ausgänge setzen**Ausgänge zurücksetzen**

Die Register zum Ansteuern der Ausgangsmodule können nicht nur beschrieben, sondern auch ausgelesen werden. Somit läßt sich der aktuelle Ausgangszustand eines (oder mehrerer Module) zu einem späteren Zeitpunkt abfragen. Beim versehentlichen Beenden der Software oder gar eines Absturzes ist der Zustand der Ausgangsmodule erfassbar. Ein laufender Prozess muß somit nicht zurückgesetzt werden.

2.2.3.2. Ausgänge zurücklesen

Adresse	r/w	Bit	Beschreibung
0000 hex	Read	0-7	Zurücklesen der Ausgänge 1-8
0001 hex	Read	0-7	Zurücklesen der Ausgänge 9-16
0002 hex	Read	0-7	Zurücklesen der Ausgänge 17-24
0003 hex	Read	0-7	Zurücklesen der Ausgänge 25-32
0004 hex	Read	0-7	Zurücklesen der Ausgänge 33-40
0005 hex	Read	0-7	Zurücklesen der Ausgänge 41-48
0006 hex	Read	0-7	Zurücklesen der Ausgänge 49-56
0007 hex	Read	0-7	Zurücklesen der Ausgänge 57-64
0008 hex	Read	0-7	Zurücklesen der Ausgänge 65-72
0009 hex	Read	0-7	Zurücklesen der Ausgänge 73-80
000A hex	Read	0-7	Zurücklesen der Ausgänge 81-88
000B hex	Read	0-7	Zurücklesen der Ausgänge 89-96
000C hex	Read	0-7	Zurücklesen der Ausgänge 97-104
000D hex	Read	0-7	Zurücklesen der Ausgänge 105-112
000E hex	Read	0-7	Zurücklesen der Ausgänge 113-120
000F hex	Read	0-7	Zurücklesen der Ausgänge 121-128

2.2.3.3. Ausgänge setzen

Adresse	r/w	Bit	Beschreibung
0080 hex	Write	0-7	Setzen der Ausgänge 1-8
0081 hex	Write	0-7	Setzen der Ausgänge 9-16
0082 hex	Write	0-7	Setzen der Ausgänge 17-24
0083 hex	Write	0-7	Setzen der Ausgänge 25-32
0084 hex	Write	0-7	Setzen der Ausgänge 33-40
0085 hex	Write	0-7	Setzen der Ausgänge 41-48
0086 hex	Write	0-7	Setzen der Ausgänge 49-56
0087 hex	Write	0-7	Setzen der Ausgänge 57-64
0088 hex	Write	0-7	Setzen der Ausgänge 65-72
0089 hex	Write	0-7	Setzen der Ausgänge 73-80

Adresse	r/w	Bit	Beschreibung
008A hex	Write	0-7	Setzen der Ausgänge 81-88
008B hex	Write	0-7	Setzen der Ausgänge 89-96
008C hex	Write	0-7	Setzen der Ausgänge 97-104
008D hex	Write	0-7	Setzen der Ausgänge 105-112
008E hex	Write	0-7	Setzen der Ausgänge 113-120
008F hex	Write	0-7	Setzen der Ausgänge 121-128

Dieses Register dient zum Setzen von digitalen Ausgängen.

Hat das Bit einen Wert von 0 wird er ignoriert, beträgt der Wert 1 wird der Ausgang gesetzt.

Beispiel:

Auf Adresse 0x0080 wird der Wert 0x02 geschrieben. Kanal 2 wird gesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x0080 wird der Wert 0x03 geschrieben. Kanal 1 und 2 werden gesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x0080 wird der Wert 0x04 geschrieben. Kanal 3 wird gesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x0080 wird der Wert 0x08 geschrieben. Kanal 4 wird gesetzt, alle anderen Kanäle bleiben unverändert.

Falls Sie Kanäle gleichzeitig setzen und zurücksetzen möchten, nutzen Sie bitte folgendes Register:

Ausgänge setzen und zurücksetzen

2.2.3.4. Ausgänge zurücksetzen

Adresse	r/w	Bit	Beschreibung
00A0 hex	Write	0-7	Zurücksetzen der Ausgänge 1-8
00A1 hex	Write	0-7	Zurücksetzen der Ausgänge 9-16
00A2 hex	Write	0-7	Zurücksetzen der Ausgänge 17-24
00A3 hex	Write	0-7	Zurücksetzen der Ausgänge 25-32
00A4 hex	Write	0-7	Zurücksetzen der Ausgänge 33-40
00A5 hex	Write	0-7	Zurücksetzen der Ausgänge 41-48
00A6 hex	Write	0-7	Zurücksetzen der Ausgänge 49-56
00A7 hex	Write	0-7	Zurücksetzen der Ausgänge 57-64
00A8 hex	Write	0-7	Zurücksetzen der Ausgänge 65-72
00A9 hex	Write	0-7	Zurücksetzen der Ausgänge 73-80
00AA hex	Write	0-7	Zurücksetzen der Ausgänge 81-88
00AB hex	Write	0-7	Zurücksetzen der Ausgänge 89-96
00AC hex	Write	0-7	Zurücksetzen der Ausgänge 97-104
00AD hex	Write	0-7	Zurücksetzen der Ausgänge 105-112
00AE hex	Write	0-7	Zurücksetzen der Ausgänge 113-120
00AF hex	Write	0-7	Zurücksetzen der Ausgänge 121-128

Dieses Register dient zum Zurücksetzen von digitalen Ausgängen.

Hat das Bit einen Wert von 0 wird er ignoriert, beträgt der Wert 1 wird der Ausgang zurückgesetzt.

Beispiel:

Auf Adresse 0x00A0 wird der Wert 0x02 geschrieben. Kanal 2 wird zurückgesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x00A0 wird der Wert 0x03 geschrieben. Kanal 1 und 2 werden zurückgesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x00A0 wird der Wert 0x04 geschrieben. Kanal 3 wird zurückgesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x00A0 wird der Wert 0x08 geschrieben. Kanal 4 wird zurückgesetzt, alle anderen Kanäle bleiben unverändert.

Falls Sie Kanäle gleichzeitig setzen und zurücksetzen möchten, nutzen Sie bitte folgendes Register:

Ausgänge setzen und zurücksetzen

2.2.4. Register für digitale PWM Ausgänge (RO-PWM)

Adresse	r/w	Bit	Beschreibung
0800 hex	Write	0-7	Setzen des PWM-Ausgangs 1
0801 hex	Write	0-7	Setzen des PWM-Ausgangs 2
0802 hex	Write	0-7	Setzen des PWM-Ausgangs 3
0803 hex	Write	0-7	Setzen des PWM-Ausgangs 4
0804 hex	Write	0-7	Setzen des PWM-Ausgangs 5
0805 hex	Write	0-7	Setzen des PWM-Ausgangs 6
0806 hex	Write	0-7	Setzen des PWM-Ausgangs 7
0807 hex	Write	0-7	Setzen des PWM-Ausgangs 8
0808 hex	Write	0-7	Setzen des PWM-Ausgangs 9
0809 hex	Write	0-7	Setzen des PWM-Ausgangs 10
080A hex	Write	0-7	Setzen des PWM-Ausgangs 11
080B hex	Write	0-7	Setzen des PWM-Ausgangs 12
080C hex	Write	0-7	Setzen des PWM-Ausgangs 13
080D hex	Write	0-7	Setzen des PWM-Ausgangs 14
080E hex	Write	0-7	Setzen des PWM-Ausgangs 15
080F hex	Write	0-7	Setzen des PWM-Ausgangs 16
...
087D hex	Write	0-7	Setzen des PWM-Ausgangs 126
087E hex	Write	0-7	Setzen des PWM-Ausgangs 127
087F hex	Write	0-7	Setzen des PWM-Ausgangs 128

2.2.4.1. Beispiele für das Setzen eines PWM Ausgangs

Beispiel 1

Am digitalen PWM Ausgang 2 eines RO-PWM-Moduls soll ein PWM-Verhältnis von 50% (50% high / 50% low) ausgegeben werden:

Werte der Parameter

Parameter	Wert [dez]	Wert [hex]
PWM-Verhältnis [%]	50	32
Ausgang 2	1	1

Inhalt des Registers

Adresse [hex]	Bit	Wert [hex]
0801	0-7	32

Beispiel 2

Am digitalen PWM Ausgang 67 eines RO-PWM-Moduls soll ein PWM-Verhältnis von 20% (20% high / 80% low) ausgegeben werden:

Werte der Parameter

Parameter	Wert [dez]	Wert [hex]
PWM-Verhältnis [%]	20	14
Ausgang 67	66	42

Inhalt des Registers

Adresse [hex]	Bit	Wert [hex]
0842	0-7	14

2.2.5. Register für 48-Bit Zähler Eingänge

2.2.5.1. 48-Bit Eingangszähler Adressbereich

Adresse	Bit	Beschreibung
5000 hex	64	Eingangszähler 1
5008 hex	64	Eingangszähler 2
5010 hex	64	Eingangszähler 3
5018 hex	64	Eingangszähler 4
5020 hex	64	Eingangszähler 5
5028 hex	64	Eingangszähler 6
5030 hex	64	Eingangszähler 7
5038 hex	64	Eingangszähler 8
5040 hex	64	Eingangszähler 9
5048 hex	64	Eingangszähler 10
5050 hex	64	Eingangszähler 11
5058 hex	64	Eingangszähler 12
5060 hex	64	Eingangszähler 13
5068 hex	64	Eingangszähler 14
5070 hex	64	Eingangszähler 15
5078 hex	64	Eingangszähler 16

2.2.5.1.1. 48-Bit Eingangszähler Register

Offset [hex]	r/w	Bit	Beschreibung
0	Read	0-7	48-Bit Counter Wert
1	Read	8-15	
2	Read	16-23	
3	Read	24-31	
4	Read	32-39	
5	Read	40-47	
6	r/w	0-3	Zähler Modus
6	r/w	4-7	Submodus
7	r/w	0	Eingangszustand (0/1)
7	-	1-3	Nicht benutzt
7	r/w	4-7	Eingangsfiter

2.2.5.1.1.1. Zähler Modus

Zähler Modus	Wert [hex]
COUNT_RISING_EDGE	0
T	D
FREQUENCY	E
PWM	F

Submodus	Wert [hex]
Zählen ohne Reset	0
Zählen mit Reset bei Lesen	1
Reset, wenn an Kanal 7 = 1 ist	7
Latch common*	8

* Der Submodus "Latch common" kann mit anderen Submodi logisch verknüpft werden.

2.2.5.1.1.2. Eingangsfilter

Eingangs- filter	Wert [hex]
20 ns	0
100 ns	1
250 ns	2
500 ns	3
1 us	4
2,5 us	5
5 us	6
10 us	7
25 us	8
50 us	9
100 us	A
250 us	B
500 us	C
1 ms	D
2,5 ms	E
5 ms	F

2.2.5.1.2. 48-Bit Eingangszähler Latchregister (Komplett für ein RO-CNT8 Modul)

Adresse	Beschreibung
5002 hex	Write latch common 1-8
5003 hex	Write reset common 1-8
5042 hex	Write latch common 9-16
5043 hex	Write reset common 9-16

2.2.6. Register für Pulsgenerator Ausgänge (für RO-CNT8 Modul)

2.2.6.1. Pulsgenerator Ausgang Adressbereich

Adresse	Bit	Beschreibung
5800 hex	128	Pulsgenerator-Ausgang 1
5810 hex	128	Pulsgenerator-Ausgang 2
5820 hex	128	Pulsgenerator-Ausgang 3
5830 hex	128	Pulsgenerator-Ausgang 4
5840 hex	128	Pulsgenerator-Ausgang 5
5850 hex	128	Pulsgenerator-Ausgang 6
5860 hex	128	Pulsgenerator-Ausgang 7
5870 hex	128	Pulsgenerator-Ausgang 8
5880 hex	128	Pulsgenerator-Ausgang 9
5890 hex	128	Pulsgenerator-Ausgang 10
58A0 hex	128	Pulsgenerator-Ausgang 11
58B0 hex	128	Pulsgenerator-Ausgang 12
58C0 hex	128	Pulsgenerator-Ausgang 13
58D0 hex	128	Pulsgenerator-Ausgang 14
58E0 hex	128	Pulsgenerator-Ausgang 15
58F0 hex	128	Pulsgenerator-Ausgang 16

2.2.6.1.1. Pulsgenerator Ausgang Register

Offset [hex]	r/w	Bit	Beschreibung
0-3	Write	0-31	Modus
4-7	Write	0-31	Par0 (Anzahl der Impulse) Par0 = 0 -> unendlich viele Impulse
8-B	Write	0-31	Par1 Low-Zeit (t [ns] / 10 - 1)
C-F	Write	0-31	Par2 High-Zeit (t [ns] / 10 - 1)

Beispiel für das Einstellen der low-/high-Zeit (Par1/Par2)

500 ns -> $500 / 10 - 1 = 49(\text{dez})$

7 μs -> $7000 / 10 - 1 = 699(\text{dez})$

2.2.6.1.2. Beispiele für das Setzen eines Pulsgenerator Ausgangs

Beispiel 1

An Pulsgenerator-Ausgang 3 sollen 5 Impulse mit einer low-Zeit von 400ns und einer high-Zeit von 600ns erzeugt werden.

Werte der Parameter

Parameter	Wert [dez]	Wert [hex]
Modus	0	0
Par0	5	5
Par1	$400 / 10 - 1 = 39$	27
Par2	$600 / 10 - 1 = 59$	3B

Inhalt der Register

Adresse [hex]	Bit	Wert [hex]
5820	0-7	00
5821	0-7	00
5822	0-7	00
5823	0-7	00
5824	0-7	05
5825	0-7	00
5826	0-7	00
5827	0-7	27
5828	0-7	00
5829	0-7	00
582A	0-7	00
582B	0-7	00
582C	0-7	3B
582D	0-7	00
582E	0-7	00
582F	0-7	00

Beispiel 2

An Pulsgenerator-Ausgang 12 sollen 1.000.000 Impulse mit einer low-Zeit von 2,5ms und einer high-Zeit von 3ms erzeugt werden.

Werte der Parameter

Parameter	Wert [dez]	Wert [hex]
Modus	0	0
Par0	$1.000.000 / 10 - 1 = 99.999$	1869F
Par1	$2.500.000 / 10 - 1 = 249.999$	3D08F
Par2	$3.000.000 / 10 - 1 = 299.999$	493DF

Inhalt der Register

Adresse [hex]	Bit	Wert [hex]
58B0	0-7	00
58B1	0-7	00
58B2	0-7	00
58B3	0-7	00
58B4	0-7	9F
58B5	0-7	86
58B6	0-7	01
58B7	0-7	00
58B8	0-7	8F
58B9	0-7	D0
58BA	0-7	03
58BB	0-7	00
58BC	0-7	DF
58BD	0-7	93
58BE	0-7	04
58BF	0-7	00

2.2.7. Register für die Richtungssetzung der Ein-/Ausgänge

Zu beachten ist, dass die Richtungssetzung nur für Ein-/Ausgangsmodule verwendbar ist, dessen Ein-/Ausgaberrichtung veränderbar ist.

Die Register der programmierbaren Ein-/Ausgänge ermöglichen es die Richtung der Kanäle, je nach Bedarf, bitweise (einzeln) oder byteweise (in 8 Bit Blöcken) als Ein- oder Ausgang zu setzen. Im Auslieferungszustand sind sie als Eingang gesetzt.

Bitwertigkeit für die Zuweisung der Richtung

Bitwertigkeit	Beschreibung
0	Richtungssetzung Eingang
1	Richtungssetzung Ausgang

2.2.7.1. Byteweise Richtungssetzung der Ein-/Ausgänge

Adresse	r/w	Bit	Beschreibung
0100 hex	Read/Write		Richtungssetzung Ein- und Ausgänge
		0	Ein-/Ausgang 1-8
		1	Ein-/Ausgang 9-16
		2	Ein-/Ausgang 17-24
		3	Ein-/Ausgang 25-32
		4	Ein-/Ausgang 33-40
		5	Ein-/Ausgang 41-48
		6	Ein-/Ausgang 49-56
0101 hex	Read/Write	7	Ein-/Ausgang 57-64
0101 hex	Read/Write		Richtungssetzung Ein- und Ausgänge
		0	Ein-/Ausgang 65-72
		1	Ein-/Ausgang 73-80

Adresse	r/w	Bit	Beschreibung
		2	Ein-/Ausgang 81-88
		3	Ein-/Ausgang 89-96
		4	Ein-/Ausgang 97-104
		5	Ein-/Ausgang 105-112
		6	Ein-/Ausgang 113-120
		7	Ein-/Ausgang 121-128

Beispiel um Ein-/Ausgänge in folgender Richtung zu setzen:

Richtung	r/w	Ein-/Ausgangsnummer	Wertigkeit (bit)
Eingang	Read	9-16	0
Eingang	Read	41-48	0
Ausgang	Write	49-56	1

Im Register 0100 hex ist das 2. Bit auf 0, das 5. Bit auf 1 und das 6. Bit auf 1 zu setzen. Das resultierende Bitmuster ist: x10x xx0x. Dabei steht "x" für den im Register bereits vorhandenen Wert.

2.2.7.2. Bitweise Richtungssetzung der Ein-/Ausgänge

Adresse	r/w	Bit	Beschreibung
0120 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 1-8
0121 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 9-16
0122 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 17-24
0123 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 25-32
0124 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 33-40
0125 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 41-48
0126 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 49-56
0127 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 57-64
0128 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 65-72
0129 hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 73-80
012A hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 81-88
012B hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 89-96
012C hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 97-104

Adresse	r/w	Bit	Beschreibung
012D hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 105-112
012E hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 113-120
012F hex	Write	0-7	Richtungssetzung Ein- und Ausgänge 121-128

Beispiel um die ersten 8 Ein-/Ausgänge in folgender Richtung zu setzen:

Kanal 1: Eingang

Kanal 2: Eingang

Kanal 3: Ausgang

Kanal 4: Ausgang

Kanal 5: Eingang

Kanal 6: Eingang

Kanal 7: Ausgang

Kanal 8: Ausgang

Adresse 0120 hex ist auf 11001100 zu setzen.

2.2.8. Register für analoge Eingänge (A/D)

2.2.8.1. A/D Wandler Adressbereich

Adresse [hex]	Beschreibung
1000-1003	Ansprechen A/D Kanal 1
1004-1007	Ansprechen A/D Kanal 2
1008-100B	Ansprechen A/D Kanal 3
100C-100F	Ansprechen A/D Kanal 4
1010-1013	Ansprechen A/D Kanal 5
1014-1017	Ansprechen A/D Kanal 6
1018-101B	Ansprechen A/D Kanal 7
101C-101F	Ansprechen A/D Kanal 8
1020-1023	Ansprechen A/D Kanal 9
1024-1027	Ansprechen A/D Kanal 10
1028-102B	Ansprechen A/D Kanal 11
102C-102F	Ansprechen A/D Kanal 12
1030-1033	Ansprechen A/D Kanal 13
1034-1037	Ansprechen A/D Kanal 14
1038-103B	Ansprechen A/D Kanal 15
103C-103F	Ansprechen A/D Kanal 16
...	...

Adresse [hex]	Beschreibung
10F8-10FB	Ansprechen A/D Kanal 63
10FC-10FF	Ansprechen A/D Kanal 64
1100-1103	Ansprechen A/D Kanal 65
...	...
1178-117B	Ansprechen A/D Kanal 95
117C-117F	Ansprechen A/D Kanal 96
1180-1183	Ansprechen A/D Kanal 97
...	...
11F4-11F7	Ansprechen A/D Kanal 126
11F8-11FB	Ansprechen A/D Kanal 127
11FC-11FF	Ansprechen A/D Kanal 128

2.2.8.2. A/D Wandler Register

Offset [hex]	r/w	Bit	Beschreibung
0, 1	Read	0-15	A/D Wert lesen
2	Write	0-7	Nicht benutzt
3	Read / Write	0-7	"A/D Modus" Schreiben/Zurücklesen

2.2.8.2.1. A/D - Modus

Modus [hex]	Beschreibung
00	0-10V
01	0-5V
40	$\pm 10V$
41	$\pm 5V$
80	0..20 mA (Nur bei RO-AD16 mit Strom-Modus)
81	4..20 mA (Nur bei RO-AD16 mit Strom-Modus)
82	0..24 mA (Nur bei RO-AD16 mit Strom-Modus)

2.2.9. Register für Temperatur Eingänge

2.2.9.1. Temperatur Kanal Adressbereich

Adresse [hex]	Beschreibung
4000-4007	Ansprechen Temperatur Kanal 1
4008-400F	Ansprechen Temperatur Kanal 2
4010-4017	Ansprechen Temperatur Kanal 3
4018-401F	Ansprechen Temperatur Kanal 4
4020-4027	Ansprechen Temperatur Kanal 5
4028-402F	Ansprechen Temperatur Kanal 6
4030-4037	Ansprechen Temperatur Kanal 7
4038-403F	Ansprechen Temperatur Kanal 8
4040-4047	Ansprechen Temperatur Kanal 9
4048-404F	Ansprechen Temperatur Kanal 10
4050-4057	Ansprechen Temperatur Kanal 11
4058-405F	Ansprechen Temperatur Kanal 12
4060-4067	Ansprechen Temperatur Kanal 13
4068-406F	Ansprechen Temperatur Kanal 14
4070-4077	Ansprechen Temperatur Kanal 15

Adresse [hex]	Beschreibung
4078-407F	Ansprechen Temperatur Kanal 16
4080-4087	Ansprechen Temperatur Kanal 17
4088-408F	Ansprechen Temperatur Kanal 18
4090-4097	Ansprechen Temperatur Kanal 19
...	...
40C8-40CF	Ansprechen Temperatur Kanal 26
40D0-40D7	Ansprechen Temperatur Kanal 27
40D8-40DF	Ansprechen Temperatur Kanal 28
40E0-40E7	Ansprechen Temperatur Kanal 29
40E8-40EF	Ansprechen Temperatur Kanal 30
40F0-40F7	Ansprechen Temperatur Kanal 31
40F8-40FF	Ansprechen Temperatur Kanal 32

2.2.9.1.1. Temperatur Eingang Register

Adresse [hex]	r/w	Bit	Beschreibung
base + 0, 1	Read	0-14	Read Bit 0-14 der aktuell gemessenen Temperatur (Wert = Temp*Faktor)
	Read	15	Vorzeichen (0 = positiv, 1 = negativ)
base + 2	Read	0-7	Faktor (0 [dez] = illegaler Wert / Sensor nicht verbunden, 1 [dez] = Faktor 10, 2 [dez] = Faktor 100)
base + 3	Read	0	Status PT100 Sensor (1 = verbunden, 0 = nicht verbunden)

Berechnung der Temperatur

Temperatur = (Vorzeichen) Wert [dez] / Faktor

2.2.9.1.2. Beispiel für das Abfragen eines Temperatur Eingangs

Beispiel 1

Abfragen der Temperatur an Temperatur Eingang 3:

Adresse [hex]	Bit	Wert [dez]	Beschreibung
4010, 4011	0-14	123	Dezimaler Wert 123
	15	0	Vorzeichen positiv
4012	0-7	1	Faktor 10
4013	0	1	PT100 Sensor verbunden

Temperatur = $+123 / 10 = +12,3 \text{ }^{\circ}\text{C}$

Beispiel 2

Abfragen der Temperatur an Temperatur Eingang 1:

Adresse [hex]	Bit	Wert [dez]	Beschreibung
4000, 4001	0-14	2263	Dezimaler Wert 2263
	15	1	Vorzeichen negativ
4002	0-7	2	Faktor 100
4003	0	1	PT100 Sensor verbunden

Temperatur = $-2263 / 100 = -22,63 \text{ }^{\circ}\text{C}$

2.2.10. Register für analoge Ausgänge (D/A)

2.2.10.1. D/A Wandler Adressbereich

Adresse [hex]	Beschreibung
2000-2007	Ansprechen D/A Wandler 1
2008-200F	Ansprechen D/A Wandler 2
2010-2017	Ansprechen D/A Wandler 3
2018-201F	Ansprechen D/A Wandler 4
2020-2027	Ansprechen D/A Wandler 5
2028-202F	Ansprechen D/A Wandler 6
2030-2037	Ansprechen D/A Wandler 7
2038-203F	Ansprechen D/A Wandler 8
2040-2047	Ansprechen D/A Wandler 9
2048-204F	Ansprechen D/A Wandler 10
2050-2057	Ansprechen D/A Wandler 11
2058-205F	Ansprechen D/A Wandler 12
2060-2067	Ansprechen D/A Wandler 13
2068-206F	Ansprechen D/A Wandler 14
2070-2077	Ansprechen D/A Wandler 15
2078-207F	Ansprechen D/A Wandler 16
...	...

Adresse [hex]	Beschreibung
20F0-20F7	Ansprechen D/A Wandler 31
20F8-20FF	Ansprechen D/A Wandler 32
2100-2107	Ansprechen D/A Wandler 33
...	...
2170-2177	Ansprechen D/A Wandler 47
2178-217F	Ansprechen D/A Wandler 48
2180-2187	Ansprechen D/A Wandler 49
...	...
21E8-21EF	Ansprechen D/A Wandler 62
21F0-21F7	Ansprechen D/A Wandler 63
21F8-21FF	Ansprechen D/A Wandler 64

2.2.10.2. D/A Wandler Register

Offset [hex]	r/w	Bit	Beschreibung
0, 1	Write/Read	0-15	D/A Wert setzen (Register rücklesbar)
2	Write/Read	0-7	"D/A – Modus" Schreiben/Zurücklesen
3	Write/Read	0-7	Output enable (0=enable / 1=disable)
4	Write/Read	0-7	Slew rate Step (Nur bei RO-DA2-ISO)
5	Write/Read	0-7	Slew Rate Update Clock (Nur bei RO-DA2-ISO)
6	Write/Read	0-7	Slew Rate enable (1=enable / 0=disable) (Nur bei RO-DA2-ISO)
7	Write	0-7	Konfiguration als Standard im EEPROM speichern (bei data = 10 [hex])
7	Write	0-7	Konfiguration aus EEPROM laden (bei data = 11 [hex])
7	Write	0-7	Auslieferungszustands-Konfiguration laden (bei data = 12 [hex])
7	Read	0-7	Letzte D/A Aktion kann hier zurückgelesen werden. 01 = D/A Daten wurden neu gesetzt 10 = Konfiguration als Standard im EEPROM wurde gespeichert. 11 = Konfiguration aus EEPROM wurde geladen 12 = Auslieferungszustands-Konfiguration wurde geladen

2.2.10.2.1. D/A - Modus

Modus [hex]	Beschreibung
00	0-10V
01	0-5V
40	$\pm 10V$
41	$\pm 5V$
80	0..20 mA (Nur bei RO-DA2-ISO)
81	4..20 mA (Nur bei RO-DA2-ISO)
82	0..24 mA (Nur bei RO-DA2-ISO)

2.2.11. Register für Stepper

2.2.11.1. Stepper Adressbereich

Adresse [hex]	Beschreibung
3000-303F	Ansprechen Stepper 1
3040-307F	Ansprechen Stepper 2
3080-30BF	Ansprechen Stepper 3
30C0-30FF	Ansprechen Stepper 4
3100-313F	Ansprechen Stepper 5
3140-317F	Ansprechen Stepper 6
3180-31BF	Ansprechen Stepper 7
31C0-31FF	Ansprechen Stepper 8

2.2.11.2. Stepper-Kommandos

Adresse [hex]	Para- meter	Bit	Beschreibung
base + 0010	par 3	0-7	Read/Write Bit 0-7 von par 3
base + 0011	par 3	8-15	Read/Write Bit 8-15 von par 3
base + 0012	par 3	16-23	Read/Write Bit 16-23 von par 3
base + 0013	par 3	24-31	Read/Write Bit 24-31 von par 3
base + 0014	par 2	0-7	Read/Write Bit 0-7 von par 2
base + 0015	par 2	8-15	Read/Write Bit 8-15 von par 2
base + 0016	par 2	16-23	Read/Write Bit 16-23 von par 2
base + 0017	par 2	24-31	Read/Write Bit 24-31 von par 2

Adresse [hex]	Para- meter	Bit	Beschreibung
base + 0018	par 1	0-7	Read/Write Bit 0-7 von par 1
base + 0019	par 1	8-15	Read/Write Bit 8-15 von par 1
base + 001A	par 1	16-23	Read/Write Bit 16-23 von par 1
base + 001B	par 1	24-31	Read/Write Bit 24-31 von par 1
base + 001C	comman d	0-7	Read/Write Bit 0-7 von command
base + 001D	comman d	8-15	Read/Write Bit 8-15 von command
base + 001E	comman d	16-23	Read/Write Bit 16-23 von command
base + 001F	comman d	24-31	Read/Write Bit 24-31 von command

2.2.11.2.1. Command-Liste

Kommando DAPI_STEPPER_CMD_	mit Wert (hex)	Bedeutung
SET_MOTORCHARACTERISTIC	1 (hex)	Setzen der Motor Konfiguration
GET_MOTORCHARACTERISTIC	2 (hex)	Abfrage der Motor Konfiguration
SET_POSITION	3 (hex)	Setzen der Motorposition
GO_POSITION	4 (hex)	Anfahren einer bestimmten Position
GET_POSITION	5 (hex)	Abfrage einer bestimmten Position
SET_FREQUENCY	6 (hex)	Einstellung der Motorsollfrequenz
SET_FREQUENCY_DIRECTLY	7 (hex)	Einstellung der Motorfrequenz
GET_FREQUENCY	8 (hex)	Abfrage der Motorfrequenz
FULLSTOP	9 (hex)	Sofortiges Anhalten des Motors
STOP	10 (hex)	Anhalten des Motors (Bremsrampe wird eingehalten)
GO_REFSWITCH	11 (hex)	Anfahren einer Referenzposition
DISABLE	14 (hex)	Ein-/Ausschalten des Motors
MOTORCHARACTERISTIC_LOAD_DEFAULT	15 (hex)	Setzen der Motorcharakteristik auf Defaultwert
MOTORCHARACTERISTIC_EEPROM_SAVE	16 (hex)	Speichern der Motorcharakteristik im EEPROM

Kommando DAPI_STEPPER_CMD_	mit Wert (hex)	Bedeutung
MOTORCHARACTERISTIC_EEPROM_LOAD	17 (hex)	Laden der Motorcharakteristik aus dem EEPROM
GET_CPU_TEMP	18 (hex)	Abfrage der Temperatur des CPU
GET_MOTOR_SUPPLY_VOLTAGE	19 (hex)	Abfrage der Versorgungsspannung des Motors
GO_POSITION_RELATIVE	20 (hex)	Anfahren einer relativen Position

2.2.11.2.2. Erklärung zu par 1, 2 und 3

Die Werte für par1, par2, par3 entnehmen Sie bitte aus der DELIB Anleitung (hier gehts zum DELIB Manual).

2.2.11.3. Stepper-Status lesen

Adresse [hex]	Parameter	Bit	Beschreibung
base + 0028	GET_POSITION	0-7	Abfrage von Bit 0-7 einer Position
base + 0029	GET_POSITION	8-15	Abfrage von Bit 8-15 einer Position
base + 002A	GET_POSITION	16-23	Abfrage von Bit 16-23 einer Position
base + 002B	GET_POSITION	24-31	Abfrage von Bit 24-31 einer Position
base + 002E	GET_ACTIVITY	0-7	Abfrage von Bit 0-7 der Statusinformationen
base + 002F	GET_SWITCH	0-7	Abfrage von Bit 0-7 des Zustands der Schalter

2.2.11.3.1. Erklärung des Befehls GET_POSITION

Dieser Befehl gibt die aktuelle Position des Motors in 1/16 Schritteinheiten zurück.

2.2.11.3.2. Erklärung der Bits für Befehl GET_ACTIVITY

Bit	Command	Beschreibung
0	DISABLE	Motor darf nicht verfahren
1	MOTORSTROMACTIV	Motorstrom ist aktiv
2	HALTESTROMACTIV	Haltestrom ist aktiv
3	GOPOSITIONACTIV	GoPosition ist aktiv
4	GOPOSITIONBREMSSEN	GoPosition Bremsung ist aktiv
5	GOREFERENZACTIV	GoReference ist aktiv
6	nicht benutzt	nicht benutzt
7	nicht benutzt	nicht benutzt

2.2.11.3.3. Erklärung der Bits für Befehl GET_SWITCH

Bit	Schalter	Beschreibung
0	Endschalter 1	Endschalter 1 ist geschlossen
1	Endschalter 2	Endschalter 2 ist geschlossen
2	Referenzschalter 1	Referenzschalter 1 ist geschlossen
3	Referenzschalter 2	Referenzschalter 2 ist geschlossen
4	nicht benutzt	nicht benutzt
5	nicht benutzt	nicht benutzt
6	nicht benutzt	nicht benutzt
7	nicht benutzt	nicht benutzt

2.2.12. Timeout-Register für digitale und analoge Ausgänge (D/A)

Beschreibung

Es gibt seit 2021 drei unterschiedliche Timeout-Methoden.

"normalen" Timeout

Dies ist der Timeout, den unsere Module schon seit 2009 besitzen.

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

Findet dann ein sogenanntes Timeout-Ereignis statt(Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "2".
- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind dann weiterhin möglich, aber der Timeout ist nicht weiter aktiv. Erst wieder, wenn er wieder aktiviert wurde.

"auto reactivate" Timeout

Dies ist ein seit 2021 implementierter Timeout-Modus, der nach Auftreten des Timeout-Ereignisses den Timeout automatisch wieder aktiviert.

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

Findet dann ein sogenanntes Timeout-Ereignis statt(Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "4".
- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind dann weiterhin möglich. UND der Timeout ist weiter aktiv. Bei erneuter Zeitüberschreitung der Timeout-Zeit werden die Ausgänge wieder ausgeschaltet.

"secure outputs" Timeout

Dies ist ein seit 2021 implementierter Timeout-Modus, der nach Auftreten des Timeout-Ereignisses einen Schreibenden Zugriff auf die Ausgänge verhindert. Somit wird sichergestellt, dass die Software erst einmal einen "sicheren" Zustand der Ausgänge wiederherstellen muss, da der Timeout-Mechanismus des Moduls die Ausgänge auf vordefinierte Werte verändert hat.

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

Findet dann ein sogenanntes Timeout-Ereignis statt(Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "6".
- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind NICHT möglich. Erst nach erneutem Aktivieren des Timeouts oder Deaktivieren des Timeouts können die Ausgänge beschrieben werden.

2.2.12.1. Timeout-Register Command setzen

Ein Timeout-Ereignis hat dann stattgefunden, wenn der Timeout aktiviert wurde und für mindestens die Zeit des Timeout-Wertes kein Datenverkehr stattfand.

Adresse [hex]	r/w	Bit	Beschreibung
FD00	Write	0-7	Timeout Command setzen

Wert = 0 (Timeout deaktivieren)

Wert = 1 ("normalen" Timeout aktivieren)

Wert = 3 ("auto reactivate" Timeout aktivieren)

Wert = 5 ("secure outputs" Timeout aktivieren)

2.2.12.2. Timeout-Register Status lesen

Ein Timeout-Ereignis hat dann stattgefunden, wenn der Timeout aktiviert wurde und für mindestens die Zeit des Timeout-Wertes kein Datenverkehr stattfand.

Adresse [hex]	r/w	Bit	Beschreibung
FD01	Read	0-7	Timeout Status lesen

Return = 0 (Timeout ist deaktiviert)

Werte für "normalen" Timeout

Return = 1 (Timeout "normal" ist aktiviert)

Return = 2 (Timeout "normal" hat stattgefunden)

Werte für "auto reactivate" Timeout

Return = 3 (Timeout "auto reactivate" ist aktiviert)

Return = 4 (Timeout "auto reactivate" hat ein oder mehrmals stattgefunden)

Werte für "secure" Timeout

Return = 5 (Timeout "secure outputs" ist aktiviert)

Return = 6 (Timeout "secure outputs" hat stattgefunden. In diesem Status wird ein schreiben auf die Outputs verhindert)

2.2.12.3. Timeout Register Wert schreiben

Adresse [hex]	r/w	Bit	Beschreibung
FD02-FD03	Write	0-15	Setzen des Timeout-Wertes (Wert * 100 ms)

Der Timeout-Wert ist das Produkt aus 16 Bit-Wert * 100 Millisekunden (die Einheit ist also 100 ms).

CAN Protokoll



3. CAN Protokoll

3.1. Geschwindigkeit und Schnittstellenparameter

Mit dem CAN Bus kann eine Übertragungsgeschwindigkeit von 10Kbit/sec bis hin zu 1Mbit/sec erreicht werden. Andere Bitraten sind per DIP Schalter einstellbar.

3.2. Aufbau „Sendepaket“

Das erste Byte eines Paketes enthält das "COMMAND".

Zeichen Nr.	Bedeutung	Erklärung
1.Byte	COMMAND	COMMAND gibt das Kommando und die Bit Breite an.
2.Byte	ADRESSE	Bit 0-7
3.Byte	ADRESSE	Bit 8-15
4.Byte	JOB-ID	Die JOB-ID kennzeichnet das Paket mit einer Kennnummer.
5.Byte	DATA	Bit 0-7
6.Byte	DATA	Bit 8-15
7.Byte	DATA	Bit 16-23
8.Byte	DATA	Bit 24-31

3.2.1. COMMAND Write

Mit COMMAND Write werden die Zeichen "0x20" zum Beschreiben von der Adressen gekennzeichnet. Dazu besteht die Möglichkeit, die Datenbreite zu bestimmen.

COMMAND	Datenbreite (Bits)
	Es werden Daten zum Gerät geschickt werden.
0x21	Bit (1 Bit)
0x22	Byte (8 Bit)
0x23	Word (16 Bit)
0x24	Long (32 Bit)

3.2.2. COMMAND Read

Mit COMMAND Read werden die Zeichen "0x10" zum Auslesen von Adressen gekennzeichnet. Dazu besteht die Möglichkeit, die Datenbreite zu bestimmen.

COMMAND	Datenbreite (Bits)
	Es werden keine Daten zum Gerät geschickt. Das Gerät schickt Daten als Antwort zurück.
0x11	Bit (1 Bit)
0x12	Byte (8 Bit)
0x13	Word (16 Bit)
0x14	Long (32 Bit)

3.2.3. ADRESSE

Eine Adresse ist 16 Bit breit und wird mit 2 hexadezimalen Zeichen übertragen.

3.2.4. JOB-ID

Die JOB-ID kennzeichnet das aktuelle Sendepaket. Zwei hintereinanderliegende Sendepakete dürfen hierbei nicht die gleiche JOB-ID erhalten.

TIP:

Die Senderoutine sollte die JOB-ID nach erfolgreicher Übertragung eines Sendepaket um "1" erhöhen. So werden nacheinander die Nummern "0" dann "1" ... "255" und dann wieder "0" verwendet.

3.2.5. DATA

Das Datenfeld ist bei einem Schreibkommando vorhanden. Bei einem Lesekommando fällt dieses weg.

3.3. Beispiel für ein Schreibbefehl:

Im folgenden Beispiel wird auf die Adresse 0012 hex das Datenwort 0F hex geschrieben. Die JOB-ID ist hierbei 0x34.

Zeichen Nr.	Bedeutung	Bits	Daten [hex]
1.Byte	COMMAND	0-7	22
2.Byte	ADRESSE	0-7	12
3.Byte	ADRESSE	8-15	00
4.Byte	JOB-ID	0-7	34
5.Byte	DATA	0-7	0f
6.Byte	DATA	8-15	00
7.Byte	DATA	16-23	00
8.Byte	DATA	24-31	00

3.4. Beispiel für das Beschreiben der ersten 16 Relais

Relais 1-13 sollen ausgeschaltet werden. Relais 14 soll angeschaltet werden und Relais 15,16 wird ausgeschaltet.

Zeichen Nr.	Bedeutung	Bits	Daten [hex]
1.Byte	COMMAND	0-7	23
2.Byte	ADRESSE	0-7	00
3.Byte	ADRESSE	8-15	00
4.Byte	JOB-ID	0-7	35
5.Byte	DATA	0-7	00
6.Byte	DATA	8-15	20
7.Byte	DATA	16-23	00
8.Byte	DATA	24-31	00

3.5. Aufbau „Antwortpaket“

Das Antwortpaket besteht aus einem 7. Byte CAN-Paket. Solch ein Paket enthält eine Antwort, eine Prüfung, die die Gültigkeit der empfangenen Daten überprüft, eine JOB-ID und 4 DATA Bytes.

3.5.1. „OK“ - Antwortpaket

Dies bedeutet, dass die Antwort "OK" war und teilt dem Paket mit, dass der Schreibbefehl erfolgreich ausgeführt wurde.

OK-Antwort zum Sendepaket Beispiel aus Kapitel 3.2

Zeichen Nr.	Bedeutung	Zeichen (Bits)	Modus
1.Byte	Antwort.	1	OK
2.Byte	Prüfung, auf Gültigkeit der Daten	0 1 2 3 4	Keine Daten 1 Bit 1 Byte 1 Word 1 Long
3.Byte	JOB-ID		
4.Byte	DATA	0-7	
5.Byte	DATA	8-15	
6.Byte	DATA	16-23	
7.Byte	DATA	24-31	

3.5.2. „Fehler“-Antwortpaket

3.5.2.1. Fehlercode „0xfd“ – JOB-ID doppelt

Ein Antwortpaket beginnend mit einem „0xfd“ besagt, dass die JOB-ID schon einmal vorhanden ist.

Zeichen Nr.	Bedeutung
1	„0xfd“ für JOB-ID schon einmal vorhanden ist.
2	JOB-ID

Kommando vom Sendepaket und die Antwort-0xfd

Zeichen Nr.	Bedeutung	
COMMAND	Datenbreite (Bits)	Anzahl Bytes
0x11	1	1
0x12	8	1
0x13	16	2
0x14	32	4

3.5.2.2. Fehlercode „0xfe“ – ungültiger Befehl

Der zuletzt gesendete „Sendepaket“ besitzt einen ungültigen Befehl.

„0xfe“-Format

Zeichen Nr.	Bedeutung
1	„0xfe“ für ungültiges Befehl.

3.6. Aufbau der CAN-Pakete TX/RX Modus

Die folgenden Seiten zeigen, wie die CAN-Pakete der unterschiedlichen I/O-Module aufgebaut sind. Ebenfalls wird dargestellt, wie A/D- und D/A-Werte berechnet werden können.

Die Daten der CAN-Pakete sind jeweils 8 Byte groß. Welche Daten an welcher Stelle zu finden sind, entnehmen Sie bitte den folgenden Seiten.

3.6.1. Digitale Eingänge

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Inhalt
1	DI channel 1-8 (Bit 0-7)
2	DI channel 9-16 (Bit 0-7)
3	DI channel 17-24 (Bit 0-7)
4	DI channel 25-32 (Bit 0-7)
5	DI channel 33-40 (Bit 0-7)
6	DI channel 41-48 (Bit 0-7)
7	DI channel 49-56 (Bit 0-7)
8	DI channel 57-64 (Bit 0-7)

3.6.2. Digitale Ausgänge

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Inhalt
1	DO channel 1-8 (Bit 0-7)
2	DO channel 9-16 (Bit 0-7)
3	DO channel 17-24 (Bit 0-7)
4	DO channel 25-32 (Bit 0-7)
5	DO channel 33-40 (Bit 0-7)
6	DO channel 41-48 (Bit 0-7)
7	DO channel 49-56 (Bit 0-7)
8	DO channel 57-64 (Bit 0-7)

3.6.3. Digitale Eingangszähler (16-Bit)

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Inhalt
1	DI counter 1 (Bit 0-7)
2	DI counter 1 (Bit 8-15)
3	DI counter 2 (Bit 0-7)
4	DI counter 2 (Bit 8-15)
5	DI counter 3 (Bit 0-7)
6	DI counter 3 (Bit 8-15)
7	DI counter 4 (Bit 0-7)
8	DI counter 4 (Bit 8-15)

3.6.4. Digitale Eingangszähler (48-Bit) - 32-Bit Paket

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Inhalt
1	CNT8 counter 1 (Bit 0-7)
2	CNT8 counter 1 (Bit 8-15)
3	CNT8 counter 1 (Bit 16-23)
4	CNT8 counter 1 (Bit 24-31)
5	CNT8 counter 2 (Bit 0-7)
6	CNT8 counter 2 (Bit 8-15)
7	CNT8 counter 2 (Bit 16-23)
8	CNT8 counter 2 (Bit 24-31)

Hinweis:

Beachten Sie, dass jeweils nur die ersten 32-Bit der beiden 48-Bit Eingangszähler automatisch in einem CAN-Paket versendet werden können.

3.6.5. Digitale Eingangszähler (48-Bit) - 64-Bit Paket

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Bit	Inhalt
1	0-7	CNT8 counter 1 (Bit 0-7)
2	0-7	CNT8 counter 1 (Bit 8-15)
3	0-7	CNT8 counter 1 (Bit 16-23)
4	0-7	CNT8 counter 1 (Bit 24-31)
5	0-7	CNT8 counter 1 (Bit 31-39)
6	0-7	CNT8 counter 1 (Bit 40-47)
7	0-3	CNT8 counter 1 Zähler-Modus
7	4-7	CNT8 counter 1 Sub-Modus
8	0	CNT8 counter 1 Eingangszustand (0/1)
8	1-3	Nicht benutzt
8	4-7	CNT8 counter 1 Eingangsfilter

3.6.6. Analoge Ein- / Ausgänge

3.6.6.1. Analoge Eingänge

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Inhalt
1	A/D channel 5 (Bit 0-7)
2	A/D channel 5 (Bit 8-15)
3	A/D channel 6 (Bit 0-7)
4	A/D channel 6 (Bit 8-15)
5	A/D channel 7 (Bit 0-7)
6	A/D channel 7 (Bit 8-15)
7	A/D channel 8 (Bit 0-7)
8	A/D channel 8 (Bit 8-15)

Der Wertebereich eines A/D Wandlers gibt an, in welchem Bereich analoge Signale (z.B. im Bereich 0-5V), digital umgesetzt werden. Die Einstellungen bezüglich des Wertebereiches können im CAN-Configuration-Utility vorgenommen werden.

Anmerkung:

Der hexadezimal Wert FFFF kennzeichnet immer die obere Grenze eines Wertebereiches, der Wert 0000 die untere.

Formel (A/D-Modus +/-10V, +/-5V oder +/-2,5V)

Spannung = $(\text{Wert} * (\text{max. Spannungswert} * 2) / 0xFFFF) - \text{max. Spannungswert}$

Formel (A/D-Modus 0..10V, 0..5V oder 0..2,5V)

Spannung = $\text{Wert} * \text{max. Spannungswert} / 0xFFFF$

Formel (A/D-Modus 0..20mA, 4..20mA oder 0..24mA)

Stromstärke = $\text{Wert} * 25 / 0xFFFF$

3.6.6.2. Analoge Ausgänge

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Inhalt
1	D/A channel 1 (Bit 0-7)
2	D/A channel 1 (Bit 8-15)
3	D/A channel 2 (Bit 0-7)
4	D/A channel 2 (Bit 8-15)
5	D/A channel 3 (Bit 0-7)
6	D/A channel 3 (Bit 8-15)
7	D/A channel 4 (Bit 0-7)
8	D/A channel 4 (Bit 8-15)

Der Wertebereich eines D/A Wandlers gibt an, in welchem Bereich digitale Signale, analog (z.B. im Bereich 0-5V) umgesetzt werden. Die Einstellungen bezüglich des Wertebereiches können im CAN-Configuration-Utility vorgenommen werden.

Anmerkung:

Der hexadezimal Wert FFFF kennzeichnet immer die obere Grenze eines Wertebereiches, der Wert 0000 die untere.

Hinweis:

Die Ausgabe in einen Strombereich ist nur bei Modulen möglich, die diesen Modus auch unterstützen.

3.6.6.3. Beispiele

Beispiel Spannungsbereich $\pm 10\text{V}$

Wert (hex)	Spannung
FFFF	+10 V
8000	0 V
0000	-10V

Beispiel zur Berechnung:

CAN-Data-Byte0, 1 = 0x4711[hex]

Wertebereich = +/-10 V

Berechnung:

Spannung = $(0x4711 * (10 * 2) / 0xFFFF) - 10 = -4,45 \text{ V}$

Beispiel Spannungsbereich 0-5V

Wert (hex)	Spannung
FFFF	+5 V
8000	+2,5 V
0000	0 V

Beispiel zur Berechnung:

CAN-Data-Byte0, 1 = 0x4711[hex]

Wertebereich = 0-5 V

Berechnung:

Spannung = $0x4711 * 5 / 0xFFFF = 1,38 \text{ V}$

Beispiel Strombereich

Wert (hex)	Stromstärke
FFFF	25 mA
8000	12,5 mA
0000	0 mA

Beispiel zur Berechnung:

CAN-Data-Byte0, 1 = 0x4711[hex]

Wertebereich = 0..20 mA, 4..20 mA oder 0..24 mA

Berechnung:

Stromstärke = $0x4711 * 25 / 0xFFFF = 6,94 \text{ mA}$

Hinweis:

Bitte beachten Sie, dass sich bei einem Auto-TX-Paket mit eingestelltem Strombereich, der Wert eines A/D-Kanals im CAN-Paket auf den Modus 0..25mA bezieht.

3.6.7. Temperatur Eingänge

Dieses Beispiel zeigt den Aufbau eines Auto-TX-Paketes mit den Einstellungen für PT-100 Kanal 1 und 2.

CAN-Data-Byte	Bit	Inhalt
1	0-7	Wert Kanal 1 (Bit 0-7)
2	0-6	Wert Kanal 1 (Bit 8-14)
	7	Vorzeichen Kanal 1 (0 = positiv, 1 = negativ)
3	0-1	Faktor Kanal 1 (0 [dez] = illegaler Wert / Sensor nicht verbunden, 1 [dez] = Faktor 10, 2 [dez] = Faktor 100)
	2-7	nicht benutzt
4	0	Status PT100 Sensor Kanal 1 (0 = nicht verbunden, 1 = verbunden)
	1-7	nicht benutzt
5	0-7	Wert Kanal 2 (Bit 0-7)
6	0-6	Wert Kanal 2 (Bit 8-14)
	7	Vorzeichen Kanal 2 (0 = positiv, 1 = negativ)
7	0-1	Faktor Kanal 2 (0 [dez] = illegaler Wert / Sensor nicht verbunden, 1 [dez] = Faktor 10, 2 [dez] = Faktor 100)
	2-7	nicht benutzt
8	0	Status PT100 Sensor Kanal 2

CAN-Data-Byte	Bit	Inhalt
		(0 = nicht verbunden, 1 = verbunden)
	0-7	nicht benutzt

Berechnung der Temperatur

Temperatur = (Vorzeichen) Wert [dez] / Faktor

3.6.8. Stepper

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Inhalt
1	COMMAND
2	PAR1 (Bit 0-7)
3	PAR1 (Bit 8-15)
4	PAR1 (Bit 16-23)
5	PAR1 (Bit 24-31)
6	PAR2 (Bit 0-7)
7	PAR2 (Bit 8-15)
8	PAR3 (Bit 0-7)

3.6.8.1. Command-Liste 2

Kommando DAPI_STEPPER_CMD_	mit Wert (hex)	Bedeutung
SET_MOTORCHARACTERISTIC	1 (hex)	Setzen der Motor Konfiguration
GET_MOTORCHARACTERISTIC	2 (hex)	Abfrage der Motor Konfiguration
SET_POSITION	3 (hex)	Setzen der Motorposition
GO_POSITION	4 (hex)	Anfahren einer bestimmten Position
GET_POSITION	5 (hex)	Abfrage einer bestimmten Position
SET_FREQUENCY	6 (hex)	Einstellung der Motorsollfrequenz
SET_FREQUENCY_DIRECTLY	7 (hex)	Einstellung der Motorfrequenz
GET_FREQUENCY	8 (hex)	Abfrage der Motorfrequenz

Kommando DAPI_STEPPER_CMD_	mit Wert (hex)	Bedeutung
FULLSTOP	9 (hex)	Sofortiges Anhalten des Motors
STOP	10 (hex)	Anhalten des Motors (Bremsrampe wird eingehalten)
GO_REFSWITCH	11 (hex)	Anfahren einer Referenzposition
DISABLE	14 (hex)	Ein-/Ausschalten des Motors
MOTORCHARACTERISTIC_LOAD_DEFAULT	15 (hex)	Setzen der Motorcharakteristik auf Defaultwert
MOTORCHARACTERISTIC_EEPROM_SAVE	16 (hex)	Speichern der Motorcharakteristik im EEPROM
MOTORCHARACTERISTIC_EEPROM_LOAD	17 (hex)	Laden der Motorcharakteristik aus dem EEPROM
GET_CPU_TEMP	18 (hex)	Abfrage der Temperatur des CPU
GET_MOTOR_SUPPLY_VOLTAGE	19 (hex)	Abfrage der Versorgungsspannung des CPU
GO_POSITION_RELATIVE	20 (hex)	Anfahren einer relativen Position

3.6.8.2. Werte für par 1 zu Befehl SET_MOTORCHARACTERISTIC

Parameter (DAPI_STEPPER_MOTORCHAR_P AR_ ...)	Werte (dez)	Beschreibung
STEPMODE	1	Stepmode (Full-, 1/2-, 1/4-, 1/8-, 1/16-step)
GOFREQUENCY	2	Speed [Full-step / s] - related to full-step
STARTFREQUENCY	3	Startfrequency [Full-step / s]
STOPFREQUENCY	4	Stopfrequency [Full-step / s]
MAXFREQUENCY	5	Maximum frequency [Full-step / s]
ACCELERATIONSLOPE	6	Acceleration slope [Full-step / ms]
DECELERATIONSLOPE	7	Deceleration slope [Full-step / 10ms]
PHASECURRENT	8	Phase current [mA]
HOLDPHASECURRENT	9	Phase current for motor hold [mA]
HOLDTIME	10	Time in that the hold goes to motorstop [ms]
STATUSLEDMODE	11	Mode of the Status-LED
INVERT_ENDSW1	12	Invert endswitch1
INVERT_ENDSW2	13	Invert endswitch2
INVERT_REFSW1	14	Invert referenceswitch1
INVERT_REFSW2	15	Invert referenceswitch2

Parameter (DAPI_STEPPER_MOTORCHAR_P AR_ ...)	Werte (dez)	Beschreibung
INVERT_DIRECTION	16	Invert all direction details
ENDSWITCH_STOPMODE	17	Setting of the stop behaviour (0=Fullstop / 1=Stop)
GOREFERENCEFREQUENCY_TOEND SWITCH	18	Frequency before endswitch [Full-step / s]
GOREFERENCEFREQUENCY_AFTERE NDSWITCH	19	Frequency after endswitch [Full-step / s]
GOREFERENCEFREQUENCY_TOOFF SET	20	Frequency to optional offset [Full-step / s]

3.6.8.3. Werte für par 1 zu Befehl GO_REFSWITCH

Parameter (DAPI_STEPPER_GO_REFSWITCH_PAR_ ...)	Wert (dez)	Bezeichnung
REF1	1	Anfahren des Referenzschalter 1
REF2	2	Anfahren des Referenzschalter 2
REF_LEFT	4	Anfahren der linken Kante des Referenzschalters
REF_RIGHT	8	Anfahren der rechten Kante des Referenzschalters
REF_GO_POSITIVE	16	Start des Motors nach rechts
REF_GO_NEGATIVE	32	Start des Motors nach links
SET_POS_0	64	Nullen der Motorposition

3.6.8.4. Beispiel

Der Befehl

<code>DapiStepperCommand(handle, DAPI_STEPPER_CMD_GO_POSITION, 3200, 0, 0, 0);</code>	1,
---	----

wird in einem 8 Byte CAN Paket versendet.

Das Paket hat folgende Struktur:

CAN-Byte	Type	Wert	Byte
1	COMMAND	4	04
2	PAR1 (Bit 0-7)	3200	80
3	PAR1 (Bit 8-15)		0C
4	PAR1 (Bit 16-23)		00
5	PAR1 (Bit 24-31)		00
6	PAR2 (Bit 0-7)	0	00
7	PAR2 (Bit 8-15)		00
8	PAR3 (Bit 0-7)	0	00

3.6.9. PWM Ausgänge

Aufbau eines 8 Byte langen CAN-Paketes:

CAN-Data-Byte	Inhalt
1	PWM channel 1 (Bit 0-7)
2	PWM channel 2 (Bit 0-7)
3	PWM channel 3 (Bit 0-7)
4	PWM channel 4 (Bit 0-7)
5	PWM channel 5 (Bit 0-7)
6	PWM channel 6 (Bit 0-7)
7	PWM channel 7 (Bit 0-7)
8	PWM channel 8 (Bit 0-7)

Ethernet (TCP) Protokoll



IV

4. Ethernet (TCP) Protokoll

4.1. Das TCP-Protokoll

Die Kommunikation erfolgt über die Ethernet-Schnittstelle mit Hilfe des TCP/IP-Protokolls. Eine entsprechende Netzwerkverbindung zwischen Modul und PC wird vorausgesetzt.

4.1.1. IP-Adresse

Die IP-Adresse läßt sich über den integrierten Webserver modifizieren. Dies ist nötig, wenn die IP-Adresse bereits belegt ist.

4.1.2. Netzmaske

Die Netzmaske muß über den integrierten Webserver modifiziert werden, wenn die IP-Adresse sich in einem anderen IP-Adressraum befindet bzw. der IP-Adressraum anders aufgeteilt ist.

4.1.3. Port

Die TCP-Übertragung findet über den "Standard-Port 9912" für das Senden und Empfangen statt. Sollte sich dieser als problematisch erweisen, z.B. bedingt durch eine Firewall, so läßt er sich ändern (à s.a. Manual RO-Serie).

4.2. TCP Sende/Empfangsstring

Zuerst wird der Aufbau des Sendestrings mit einem Sendebeispiel detailliert erläutert. Anschließend wird der Antwortstring mit einem zugehörigen Antwortbeispiel beschrieben.

4.2.1. Aufbau des "Sendestrings"

16-Bit Adressmodus

Beispiel für einen Schreibbefehl im 16-Bit Adressmodus:

Im folgenden Beispiel wird auf die Adresse 0012 hex das Datenwort 0f hex geschrieben (der Sendestring besteht aus 11 Werten). Die JOB-ID ist 01.

Zeichen Nr.	Bedeutung	Wert	Hexadezimal
1	Packet_ID_0		63
2	Packet_ID_1		9a
3	TCP_CMD_RO_1		01
4	JOB_ID		01
5	Länge des Sendestrings (Bit 8-15)	11 dez.	00
6	Länge des Sendestrings (Bit 0-7)	(Anzahl aller Zeichen)	0b
7	COMMAND	ASCII "W"	57
8	WIDTH (Datenbreite)	ASCII "B"	42
9	ADDRESS Bit 8-15	0012 hex	00
10	ADDRESS Bit 0-7		12
11	DATA (= 0f hex)	0f hex	0f

Hier geht es zu dem 32-Bit Adressmodus:
32-Bit Adressmodus

32-Bit Adressmodus

Beispiel für einen Schreibbefehl im 32-Bit Adressmodus:

Im folgenden Beispiel wird auf die Adresse 80010012 hex das Datenwort 0f hex geschrieben (der Sendestring besteht aus 14 Werten). Die JOB-ID ist 01.

Zeichen Nr.	Bedeutung	Wert	Hexadezimal
1	Packet_ID_0		63
2	Packet_ID_1		9a
3	TCP_CMD_RO_1		01
4	JOB_ID		01
5	Länge des Sendestrings (Bit 8-15)	14 dez. (Anzahl aller Zeichen)	00
6	Länge des Sendestrings (Bit 0-7)		0e
7	Extended Address Mode	ASCII "X"	58
8	COMMAND	ASCII "W"	57
9	WIDTH (Datenbreite)	ASCII "B"	42
10	ADDRESS Bit 24-31	80010012 hex	80
11	ADDRESS Bit 16-23		01
12	ADDRESS Bit 8-15		00
13	ADDRESS Bit 0-7		12
14	DATA (= 0f hex)	0f hex	0f

Hier geht es zu dem 16-Bit Adressmodus:

16-Bit Adressmodus

4.2.1.1. JOB-ID

Die JOB-ID kennzeichnet den aktuellen Sendestring. Zwei hintereinanderliegende Sendestrings dürfen hierbei nicht die gleiche JOB-ID erhalten.

TIP:

Die Senderoutine sollte die JOB-ID nach erfolgreicher Übertragung eines Sendestrings um "1" erhöhen. So werden nacheinander die Nummern "0" dann "1" ... "255" und dann wieder "0" verwendet.

4.2.1.2. Länge des Sendestrings

Die Länge des Sendestrings besteht aus 2 Bytes und gibt an, aus wievielen Bytes ein Sendestring besteht. Die höherwertigen Bits werden zuerst übertragen. Im Beispiel beträgt die Länge 11 Bytes.

4.2.1.3. Extended Address Mode

Der Extended Address Mode (32-Bit Adressen) wird bei Produkten verwendet, welche über 32-Bit Register verfügen. Ein X im Sendestring zeigt, ob der 32-Bit Modus verwendet wird.

4.2.1.4. COMMAND

Mit COMMAND wird das Zeichen "W" (write) zum Beschreiben von Registern oder "R" (read) für das Auslesen von Registern gekennzeichnet.

4.2.1.5. WIDTH

Abhängig vom verwendeten "COMMAND" und der benutzten Datenbreite "WIDTH" werden unterschiedlich lange Daten versendet.

COMMAND	WIDTH	Datenbreite (Bits)	Anzahl Bytes
W	B	8	1
W	W	16	2
W	L	32	4
W	X	64	8
R		0	0

Beim COMMAND = "R" hat die das WIDTH Zeichen keinen Einfluss auf den Sendestring. Es steuert hierbei die Datenlänge das Empfangsstrings.

4.2.1.6. ADDRESS

Die Register sind entweder 16-Bit oder 32-Bit breit und werden jeweils mit 2 oder 4 hexadezimalen Zeichen dargestellt. Die Übertragung erfolgt zuerst mit dem höherwertigen Byte.

Beim 16-Bit Adressmodus werden zuerst die Bits 8-15, dann Bits 0-7 übertragen. Bei 32-Bit Adressmodus werden zuerst die Bits 24-31, 16-32, 8-15 und schlussendlich Bits 0-7 übertragen.

4.2.1.7. DATA

Das Datenfeld ist nur bei einem Schreibkommando vorhanden. Bei einem Lesekommando fällt dieses weg.

4.2.2. Aufbau des „Antwortstrings“

Der Antwortstring ist hexadezimal kodiert und beginnt mit 2 Bytes, die mit PACKET_ID_0 (hex 63) und PACKET_ID_1 (hex 9a) gekennzeichnet sind. Anschliessend folgt ein TCP_ANSWER_RO_1 (hex 81).

Nun folgt ein Fehlercode und weitere hexadezimale Zeichen. Sofern der Sendestring ein Lesebefehl ist, werden dem Antwortstring die ausgelesenen Daten am Ende angehängt.

OK-Antwort zum Sendestring Beispiel aus Kapitel 4

Zeichen Nr.	Bedeutung	Wert	Hexa- dezimal
1	DEDITEC_PACKET		63
2	_ID_0		9a
	DEDITEC_PACKET		
	_ID_1		
3	DEDITEC_TCP_AN		81
	SWER_RO_1		
4	OK (OK = 0x00)		00
5	JOB-ID		01
6	Länge des		00
7	Antwortstrings	7	07
	(Bit 8-15)		
	Länge des		
	Antwortstrings		
	(Bit 0-7)		

4.2.2.1. Fehlercode

Der Fehlercode gibt mit 0 (hex) = ok, an, dass die Daten erfolgreich empfangen wurden. Alle anderen Codes sind Fehlermeldungen.

4.2.2.2. JOB-ID

Die vom Sendestring empfangene JOB-ID wird zurückgesendet. Zwei hintereinander liegende Sendestrings dürfen hierbei nicht die gleiche JOB-ID erhalten.

4.2.2.3. Länge des Antwortstrings

Die Länge des Antwortstrings besteht aus 2 Bytes und gibt an, welche Bytelänge der Antwortstring hat. Die höherwertigen Bits werden zuerst übertragen. Im Beispiel beträgt die Länge 7 Bytes.

4.2.2.4. DATA

Das Datenfeld ist nur bei einem Lesekommando des Sendestrings vorhanden. Bei einem Schreibkommando fällt dieses weg.

Serielles Protokoll



5. Serielles Protokoll

5.1. Geschwindigkeit und Schnittstellenparameter

Die Serielle Schnittstelle ist mit 115200 Baud, 8 Datenbits, 1 Stoppbit und keine Parität konfiguriert (8,n,1).

5.2. Aufbau des „Sendestrings“

Der Beginn eines Pakets wird mit dem Zeichen <SOH> (“Start of Header”, 01 hex) gekennzeichnet und das Ende mit dem Zeichen <CR> (“Carriage-Return”, 0D hex).

Mit Ausnahme vom Anfangs- und Endzeichen, sowie COMMAND und WIDTH werden hexadezimalen Daten ASCII-kodiert übertragen. Wird beispielsweise die Adresse 0012 hex angesprochen, so werden von links nach rechts die 4 ASCII-Zeichen “0012” nacheinander übertragen. Die ASCII-kodierten hexadezimalen Werte können nur aus Zahlen von 0-9 und aus Buchstaben von A-F gebildet werden.

Beispiel für ein Schreibbefehl:

Im folgenden Beispiel wird auf die Adresse 0012 hex das Datenwort 0F hex geschrieben (der Sendestring besteht aus 16 ASCII-Zeichen).

Zeichen Nr.	Bedeutung	ASCII-Zeichen	Hexadezimal
1	Start Of Header (Markiert start eines Packets)	<SOH>	01
2, 3	MODUL-NR (=34 hex)	34	33, 34
4, 5	JOB-ID (= 12 hex)	12	31, 32
6	COMMAND	W	57
7	WIDTH (Datenbreite)	B	42
8, 9, 10, 11	ADDRESS (= 0012 hex)	0012	30, 30, 31, 32
12, 13	DATA (= 0F hex)	0F	30, 46
14, 15	CHECKSUM (= 29D hex, es werden aber nur die unteren 8 Bit übertragen)	9D	39, 44
16	END (Carriage Return)	<CR>	0D

5.2.1. Anfangszeichen

Mit dem Zeichen <SOH> wird der Beginn einer Datenübertragung gekennzeichnet.

5.2.2. MODUL-NR

Die Modul-Nr kennzeichnet das ausgewählte Modul. (Die Modulnummern können auf den Modulen mit Hilfe von DIP-Schaltern eingestellt werden.)

5.2.3. JOB-ID

Die JOB-ID kennzeichnet den aktuellen Sende-String. Zwei hintereinanderliegende Sendestrings dürfen hierbei nicht die gleiche JOB-ID erhalten.

TIP:

Die Sende-Routine sollte die JOB-ID nach erfolgreicher Übertragung eines Sendestrings um "1" erhöhen. So werden nacheinander die Nummern "0" dann "1" ... "255" und dann wieder "0" verwendet.

5.2.4. COMMAND

Mit COMMAND wird das Zeichen "W" (write) zum Beschreiben von Register oder "R" (read) für das Auslesen von Register gekennzeichnet.

5.2.5. WIDTH

Abhängig vom verwendeten "COMMAND" und der benutzten Datenbreite "WIDTH" werden unterschiedlich lange Daten versendet.

COMMAND	WIDTH	Datenbreite (Bits)	Anzahl Zeichen	ASCII
W	B	8	2	
W	W	16	4	
W	L	32	8	
W	X	64	16	
R		0	0	

Beim COMMAND = "R" hat die das WIDTH Zeichen keinen Einfluss auf den Sendestring. Es steuert hierbei die Daten-Länge des Empfangs-Strings.

5.2.6. ADDRESS

Die Register sind 16 Bit breit und werden mit 4 hexadezimalen Zeichen dargestellt. Da die Übertragung jedes hexadizimalen Zeichens mit einem ASCII-Zeichen repräsentiert wird, werden 4 ASCII-Zeichen benötigt.

5.2.7. DATA

Das Datenfeld ist nur bei einem Schreibkommando vorhanden. Bei einem Lesekommando fällt dieses weg.

5.2.7.1. CHECKSUM

CHECKSUM ist eine Prüfsumme. Es ist eine hexadezimale Aufsummierung aller Zeichen (einschliesslich <SOH>) bis zur Prüfsumme. Der Empfänger vergleicht seine Aufsummierung mit der übertragenen Prüfsumme. Sind diese gleich, gilt die Übertragung als fehlerfrei.

In hexadezimal:

$01 + 33 + 34 + 31 + 32 + 57 + 42 + 30 + 30 + 31 + 32 + 30 + 46 = 29D$

-> CHECKSUM = 9D hex (2 ist der Überlauf und fällt somit weg).

5.2.8. END

Das Ende des Sende-Strings wird mit einem <CR> gekennzeichnet (Carriage Return, 0D hex).

5.3. Aufbau des „Antwortstrings“

Der Antwortstring startet mit dem Zeichen („O“, „D“ oder „E“) gefolgt von ASCII Zeichen und wird mit einem <CR> (Carriage Return, 0d hex) beendet.

5.3.1. Antwortstring „O“ (OK)

Dies bedeutet, dass die Antwort „OK“ war und teilt dem Sender mit, dass der Schreibbefehl erfolgreich ausgeführt wurde.

„O“-Format

Zeichen Nr.	Bedeutung
1	ASCII-„O“ für OK
2	JOB-ID (Higher Byte)
3	JOB-ID (Lower Byte)
4	CHECKSUM (Higher Byte)
5	CHECKSUM (Lower Byte)
6	END (Carriage Return)

OK-Antwort zum Sendestring Beispiel aus Kapitel 3.2

Zeichen Nr.	Bedeutung	ASCII-Zeichen	Hexa- dezimal
1	OK	O	4F
2, 3	JOB-ID (= 12 hex)	12	31, 32
4, 5	CHECKSUM (= B2 hex)	B2	42, 32
6	END (Carriage Return)	<CR>	0D

5.3.2. Antwortstring „D“ (DATA)

Ein Antwortstring beginnend mit einem „D“ kennzeichnet Antwortdaten, die aus einem Lesebefehl resultieren.

„D“-Format

Zeichen Nr.	Bedeutung
1	„D“ für DATA (Antwort des Lesebefehls)
2	JOB-ID (Higher Byte)
3	JOB-ID (Lower Byte)
4,5 ...	n-Daten
4 + n	CHECKSUM (Higher Byte)
5 + n	CHECKSUM (Lower Byte)
6 + n	END (Carriage Return)

Mit $n = 2, 4, 8$ oder 16 Zeichen (abhängig von WIDTH des Sendestrings)

Kommando + WIDTH vom Sende-String und die Antwort-Daten

Lese-Anfrage		Antwort-Daten	
Command	WIDTH	Datenbreite (Bits)	Anzahl Zeichen ASCII-
R	B	8	2
R	W	16	4
R	L	32	8
R	X	64	16

5.3.3. Startzeichen mit Antwort „E“ (ERROR)

Der zuletzt gesendete "Sende-String" wurde fehlerhaft empfangen.

"E"-Format

Zeichen Nr.	Bedeutung
1	"E" für Error
2	Fehlercode
3	END (Carriage Return)

Fehlercode

Hexa-dezimal	Bedeutung
31	Kommando ungültig; es wurde ein ungültiges COMMAND benutzt
32	Ungültige Datenlänge; die Datenlänge des Sende-Strings stimmt nicht
33	Checksum error; Prüfsummenfehler

Anhang



6. Anhang

6.1. Revisionen

Rev 3.01	DEDITEC Design Update 2022
Rev 3.00	DEDITEC Design Update 2021
Rev 2.12	"Auto Reactivate" und "Secure" Timeout Commands hinzugefügt
Rev 2.11	CAN-RX Modus für PWM Ausgänge hinzugefügt
Rev 2.10	Extended Adress Mod für Ethernet Protokoll hinzugefügt
Rev 2.00	Designänderung Zusammenlegung aller Protokoll-Handbücher und der Registerbelegung

6.2. Urheberrechte und Marken

Linux ist eine registrierte Marke von Linus Torvalds.

USB ist eine registrierte Marke von USB Implementers Forum Inc.

LabVIEW ist eine registrierte Marke von National Instruments.

Intel ist eine registrierte Marke von Intel Corporation.

AMD ist eine registrierte Marke von Advanced Micro Devices, Inc.

ProfiLab ist eine registrierte Marke von ABACOM Ingenieurbüro GbR.

ispVM System ist eine registrierte Marke von Lattice Semiconductor Corporation

Windows, Visual-C/C++, -C#, -Basic, -Basic.NET und Visual-Studio sind registrierte Marken von Microsoft Corporation.

Delphi ist eine registrierte Marke von Borland Software Corporation.

Java ist eine registrierte Marke von Oracle Corporation.