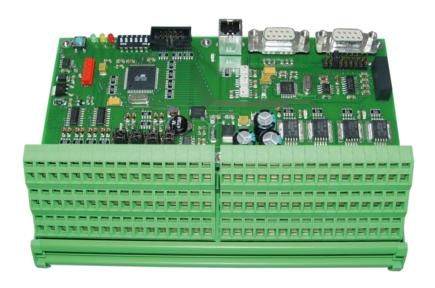
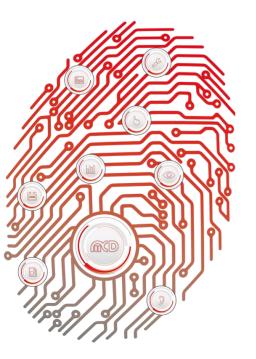


Bedienungsanleitung

Steuereinheit MIO1616







Softline -

Modline -

Conline

Boardline

Avidline

Pixline —

Applikation

MCD Elektronik GmbH

Hoheneichstr. 52 75217 Birkenfeld

Telefon 0 72 31/78 405-0
Telefax 0 72 31/78 405-10
info@mcd-elektronik.de

www.mcd-elektronik.de

Sitz: Birkenfeld

HRB 505692

Geschäftsführer: Bruno Hörter Registergericht Mannheim



Inhalt

1. ALLGEM	EIN	4
1.1. Auss	TATTUNG UND FUNKTIONEN	4
2. LIEFERU	MFANG	
	IEITSHINWEISE	
4. SERIELLI	ES 16 BIT EIN - / AUSGABEMODUL FÜR HUTSCHIENE	5
4.1. ANSC	CHLUSSBELEGUNG 3 - STOCK KLEMMLEISTE MIO1616 UND ABMARE	5
4.1.1.	ST3	5
4.1.2.	ST4	5
4.2. STEC	KERBELEGUNGEN	6
4.2.1.	ST3: 3 - Stock - Klemmenblock	6
4.2.2.	ST4: 3 - Stock - Klemmenblock	7
4.2.3.	RS232 - IN: D - SUB 9 pol BU	7
4.2.4.	RS232 - OUT: D - SUB 9 pol BU	8
4.2.5.	USB Form B	8
4.2.6.	IO Stiftwanne 10 pol	8
4.3. PLAT	INENKONFIGURATION	9
5. BEFEHLS	SSATZ	10
5.1. ALLG	EMEINE FESTLEGUNG	10
5.1.1.	Allgemein Befehl Standard Syntax	10
5.1.2.	Allgemeine Antwort	10
5.1.3.	Maximale Datenlänge der Übertragung	11
5.1.4.	Typdefinitionen	11
5.1.5.	ERROR - Code	11
5.1.6.	MIO1616 Firmwareupdate über serielle Schnittstelle	12
5.1.7.	Befehlsimplementierung bei verschiedenen SW - Versionen	13
5.1.8.	Befehl 10: EINGÄNGE + AUSGÄNGE lesen 16 Bit	14
5.1.9.	Befehl 11: ADW'S lesen 8 Bit / 10 Bit	15
5.1.10.	Befehl 12: EINGÄNGE: Triggerbedingung auf Eingangsmuster setzen	17
5.1.11.	EINGÄNGE: Trigger auf Eingangsmuster ein / aus 16 Bit über EEPROM Zelle	18
5.1.12.	Befehl 15: COUNTER – EINGÄNGE	19
5.1.13.	AUSGABE – Zustand nach Power Up über EEPROM konfigurieren	20
5.1.14.	Befehl 20: 24 V AUSGABE byteweise setzen 16 Bit	21

6.

5.1.15.	Befehl 21: 24 V AUSGABE bitweise setzen 16 - Bit	21
5.1.16.	Befehl 22: 24 V AUSGABE Bitweise blinkend setzen 16 Bit	22
5.1.17.	Befehl 24: 24 V AUSGABE Kanal 1 + 2 PWM ausgeben	24
5.1.18.	Befehl 25: 24 V INPUT Triggerschwelle mit DAW setzen 8 Bit	25
5.1.19.	Befehl 28: Mechanischen Ablauf über Timeout überwachen	25
5.1.20.	Befehl 30: M5450 MCD Busschaltung ansteuern 34 Bit	28
5.1.21.	Befehl 32: 3 - Wire - Datenausgabe über I ² C EXTERN Ports	29
5.1.22.	Befehl 34: LCD Initialisieren	30
5.1.23.	Befehl 35: LCD POS * TEXT ausgeben	30
5.1.24.	Befehl 3A: EEPROM 1 schreiben max. 00FF = 256 BYTES (auf Controller)	31
5.1.25.	Befehl 3B: EEPROM 1 lesen max. 00FF = 256 BYTES (auf Controller)	32
5.1.26.	Befehl 3F: SW VERSION Firmware lesen	32
5.1.27.	I ² C Bedienung allgemein	33
5.1.28.	Befehl 40: I ² C EXTERN START - Bedienung ausgeben	33
5.1.29.	Befehl 41: I ² C EXTERN STOP - Bedingung ausgeben	34
5.1.30.	Befehl 42: I ² C EXTERN DATENBYTES ausgeben	34
5.1.31.	Befehl 43: I ² C EXTERN DATENBYTES einlesen	35
5.1.32.	Befehl 44: I ² C EXTERN universelle Ein - / Ausgabe (Execute)	36
5.1.33.	Befehl 46: PCF8574 I ² C EXTERN bitweise setzen 8 Bit	37
5.1.34.	Befehl 47: PCF8574 I ² C EXTERN lesen 8 Bit	38
5.1.35.	Befehl 48: I ² C EXTERN TASTENDECODER MATRIX 5 x 4 einlesen	39
5.1.36.	Befehl 4A: EEPROM universal schreiben (nicht kompatibel)	41
5.1.37.	Befehl 4B: EEPROM universal lesen (nicht PIC kompatibel)	42
TECHNISO	CHE DATEN	44



1. Allgemein

Die MCD Steuereinheit MIO1616 wird zur Ansteuerung und Kontrolle von pneumatisch oder elektrisch angetriebenen Komponenten eingesetzt.

Die Ansteuerung erfolgt wahlweise über USB, RS232 oder RS485.

Jedes Interface kann als Masterumsetzer von USB auf RS232 bzw. auf RS485 verwendet werden.

1.1. Ausstattung und Funktionen

- 16 x Ein / Ausgänge
- 4 x Eingänge als schnelle Ereigniszähler / Trigger bis 10 kHz
- 2 x Ausgänge als PWM Generator von 10 bis 250 Hz, 0 bis 100 %
- 4 x ADW Eingänge, Bereich 0 bis 10 V, 10 Bit
- · Programmierbare Triggerfunktion auf Eingangsmuster
- Getrennte Zuführung der Betriebsspannungen für Ausgang 1 bis 8, Ausgang 9 bis 16 und Versorgung der Steuerelektronik
- Versorgungs und EA Spannung 12...24 V möglich
- Universelle IO Schnittstelle zum Ansteuern von I²C Komponenten, DOT Matrix Anzeigen
- Speicherung von Werten in nicht flüchtigen Speicher (EEPROM)
- Ansteuerung von externen Modulen über I2C Schnittstelle
- Textausgabe auf DOT Matrix Anzeigen optional möglich
- Betrieb von mehreren Geräten über adressierbare, serielle Schnittstelle
- Montage auf Hutschiene möglich

Bestellnummer: #7587 - V3

2. Lieferumfang

Ausführung mit 16 IO's, AD - Wandler, I2C Bus und LCD - Schnittstelle:

- Steuerung mit Klemmleiste für Ausgänge, Sensoren, Versorgungsspannung, etc.
- · Konfektioniertes Anschlusskabel mit Netzteil lieferbar
- Steuerungssoftware zur Inbetriebnahme verfügbar
- Universal Com Server (UCS) Software für komplexe Anwendungen verfügbar

3. Sicherheitshinweise



Die Steuereinheit MIO1616 dient zur Ansteuerung und Kontrolle von pneumatisch oder elektrisch angetriebenen Komponenten. Das Produkt wurde mit größter Sorgfalt erstellt und getestet.

Dennoch wird ausdrücklich eine Haftung für alle Schäden, die durch den Einsatz des Produkts entstehen, ausgeschlossen. Alle in der Dokumentation genannten Warenzeichen bzw. Schutzmarken sind urheberrechtlich geschützt, und sind Eigentum der jeweiligen Inhaber.



4. Serielles 16 Bit Ein - / Ausgabemodul für Hutschiene

4.1. Anschlussbelegung 3 - Stock Klemmleiste MIO1616 und Abmaße

4.1.1. ST3

GND	Analog	GND	Analog	GND	Analog			GND:	24V			Us Out 18	Us Out 916	UB Co + Inpu	
	IIN I		IIN Z		IIN 3		IIN 4	GND	+24V	+24V	+24V	+24V	+24V	+24 V	+24V
1.3	2.3	3.3	4.3	5.3	6.3	7.3	8.3	9.3	10.3	11.3	12.3	13.3	14.3	15.3	16.3
OUT1	OUT2	OUT3	OUT4	OUT5	OUT6	OUT7	OUT8	OUT 9	OUT 10	OUT 11	OUT 12	OUT 13	OUT 14	OUT 15	OUT 16
1.2	2.2	3.2	4.2	5.2	6.2	7.2	8.2	9.2	10.2	11.2	12.2	13.2	14.2	15.2	16.2
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
1.1	2.1	3.1	4.1	5.1	6.1	7.1	8.1	9.1	10.1	11.1	12.1	13.1	14.1	15.1	16.1

4.1.2. ST4

| +24V |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1.3 | 2.3 | 3.3 | 4.3 | 5.3 | 6.3 | 7.3 | 8.3 | 9.3 | 10.3 | 11.3 | 12.3 | 13.3 | 14.3 | 15.3 | 16.3 |
| IN1 | IN2 | IN3 | IN4 | IN5 | IN6 | IN7 | IN8 | IN9 | IN10 | IN11 | IN12 | IN13 | IN14 | IN15 | IN16 |
| 1.2 | 2.2 | 3.2 | 4.2 | 5.2 | 6.2 | 7.2 | 8.2 | 9.2 | 10.2 | 11.2 | 12.2 | 13.2 | 14.2 | 15.2 | 16.2 |
| GND |
| 1.1 | 2.1 | 3.1 | 4.1 | 5.1 | 6.1 | 7.1 | 8.1 | 9.1 | 10.1 | 11.1 | 12.1 | 13.1 | 14.1 | 15.1 | 16.1 |

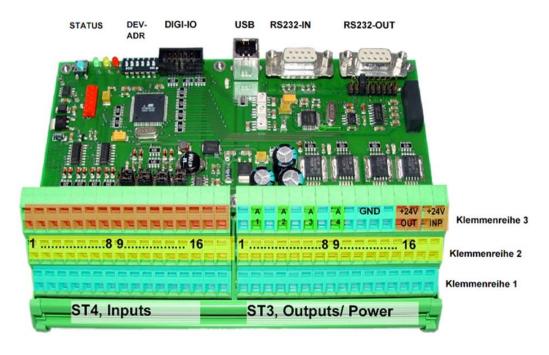


Abbildung 1: Mechanische Abmaße und Detailbild des MIO1616



4.2. Steckerbelegungen

4.2.1. ST3: 3 - Stock - Klemmenblock

Klemme	Signal	Klemme	Signal	Klemme	Signal
1.1		1.2	OUT1	1.3	GND
			PWM1 (10 Hz250 Hz) 8 Bit		
			Frequenz PWM1 = PWM2		
2.1		2.2	OUT2	2.3	Analog IN1 (8/10Bit) 010V
			PWM2 (10 Hz250 Hz) 8 Bit		Alternativ Widerstandskodierung
			Frequenz PWM1 = PWM2		010 kOhm
3.1		3.2	OUT3	3.3	GND
4.1		4.2	OUT4	4.3	Analog IN2 (8 / 10 Bit) 010 V
					Alternativ Widerstandskodierung 010 kOhm
5.1		5.2	OUT5	5.3	GND
6.1		6.2	OUT6	6.3	Analog IN3 (8 / 10 Bit) 010 V
	GND				Alternativ Widerstandskodierung 010 kOhm
7.1	OND	7.2	OUT7	7.3	GND
8.1		8.2	OUT8	8.3	Analog IN4 (8 / 10Bit) 010 V
					Alternativ Widerstandskodierung 010 kOhm
9.1		9.2	OUT9	9.3	
10.1		10.2	OUT10	10.3	GND (von Versorgung Netzteil)
11.1		11.2	OUT11	11.3	(1011 101001guilg Holztoll)
12.1		12.2	OUT12	12.3	
13.1		13.2	OUT13	13.3	Nutzen für Ausgänge 18 (1032 V / 5 A)
14.1		14.2	OUT14	14.3	Nutzen für Ausgänge 916 (1032 V / 5 A)
15.1		15.2	OUT15	15.3	Ub Versorgung UB Inputs und
16.1		16.2	OUT16	16.3	Steuerung Ausgänge (1032 V / 5 A)



4.2.2. ST4: 3 - Stock - Klemmenblock

Klemme	Signal	Klemme	Signal	Klemme	Signal
1.1		1.2	IN1 (Counter max 5 KHz / Trigger Input1)	1.3	
2.1		2.2	IN2 (Counter max 5 KHz / Trigger Input2)	2.3	
3.1		3.2	IN3 (Counter max 5 KHz / Trigger Input3)	3.3	
4.1		4.2	IN4 (Counter max 5 KHz / Trigger Input4)	4.3	
5.1		5.2	IN5	5.3	
6.1		6.2	IN6	6.3	UBout Versorgung
7.1		7.2	IN7	7.3	Initiatoren (1032 V / 5 A)
8.1	GND	8.2	IN8	8.3	Strombegrenzung
9.1	JOND	9.2	IN9	9.3	durch Polyswitch
10.1		10.2	IN10	10.3	nach 15 sec. Überlast. Erholzeit
11.1		11.2	IN11	11.3	<= 1 min
12.1		12.2	IN12	12.3	
13.1		13.2	IN13	13.3	
14.1		14.2	IN14	14.3	
15.1		15.2	IN15	15.3	
16.1		16.2	IN16	16.3	

4.2.3. RS232 - IN: D - SUB 9 pol BU

Über "*Ubextern"* kann hier 24 V für die Steuerung eingespeist werden (nur bei kleinen Lastströmen erlaubt, Lötbrücken schließen).

Zum Versorgen der Ausgänge am Klemmenblock ST3 13.1,14.2,15.1 verbinden.

1	UB extern
2	TXD - RS232 / P - RS485
3	RXD - RS232 / M - RS485
4	GND UB
5	GND RS232
6	UB extern
7	UB extern
8	GND UB
9	GND UB



4.2.4. RS232 - OUT: D - SUB 9 pol BU

1	UB extern
2	RXD - RS232 / P - RS485
3	TXD - RS232 / M - RS485
4	GND UB
5	GND RS232
6	UB extern
7	UB extern
8	GND UB
9	GND UB

4.2.5. USB Form B

1	5 V USB (for reset only)
2	D -
3	D+
4	GND USB = RS232

4.2.6. IO Stiftwanne 10 pol

Der LCD Kontrast muss über eine gesonderte negative Spannung erzeugt werden.

LCD Beleuchtung nicht über "+ 5 V out" erlaubt → "UB ext. OUT" mit entsprechendem Vorwiderstand verwenden.

DSUB 9 pol BU	10pol	LCD	MCDBUS	IIC	μP Port
1	1	DATA7	A2		PD5
6	2	DATA6	A1		PD4
2	3	LCDEN	DATA	SDA	PD0
7	4	LCDRS	CLOCK	SCL	PD1
3	5	DATA4	DLEN		PD2
8	6	DATA5	A0		PD3
4	7	UB ext. out (24 V)			
9	8	+ 5 V out (0.2 A)			
5	9	GND	GND	GND	GND



4.3. Platinenkonfiguration

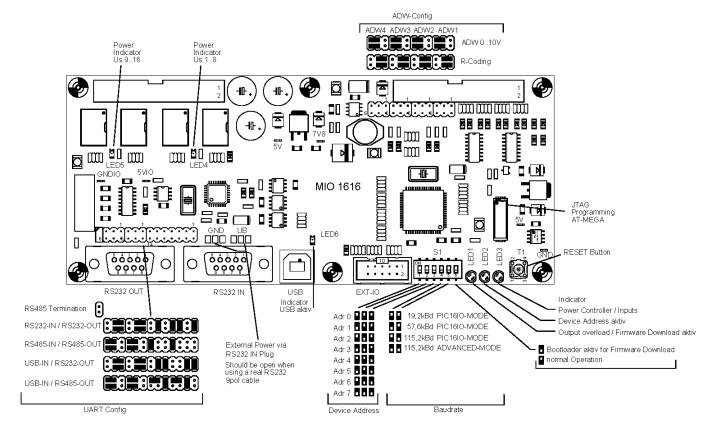


Abbildung 2: Layout MIO1616



5. Befehlssatz

5.1. Allgemeine Festlegung

Baudrate: 19200Bd / 57600Bd / 115200Bd, 8 Daten ,1 Stop, keine Parität, kein HW Handshake

Als Startzeichen wird \$12 gesendet.

Address Nibble = '0..7' ist per Jumper einstellbar.

Längenbyte ist die Anzahl der gesendeten ASCII Zeichen (Char) beginnend mit dem CMD - Byte.

Als Ende und Terminierungszeichen wird \$0D gesendet.

5.1.1. Allgemein Befehl Standard Syntax

Befehl: "?" entspricht einem ASCII Zeichen

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			Triggerzeichen für Interface
?	Addr Nibble	Deviceadresse		\$20\$7E	Alle druckbaren ASCII Zeichen Default ASCII "07"
??	LENbyte	Längenbyte [u08]		2255	Anzahl der ASCII Zeichen beginnend mit CMD - Byte bis Daten - Byte n
??	CMDbyte	Befehlscode [u08]			Befehlscode sh. Befehl
??	Dbyte1				
\$0D	Term.	Terminierung			Abschlusszeichen

5.1.2. Allgemeine Antwort

Antwort: Bei korrekter Befehls - Parametrisierung

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			Triggerzeichen für Interface
?	Addr Nibble	Deviceadresse		\$20\$7E	Wiederholung der Deviceadresse in der Antwort
??	CMDbyte	Befehlscode [u08]			Wiederholung des Befehlscode in der Antwort
??	Dbyte1				
\$0D	Term.	Terminierung			Abschlusszeichen



Antwort: Bei Fehler in Befehls - Parametrisierung

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			Triggerzeichen für Interface
?	Addr Nibble	Deviceadresse		\$20\$7E	Wiederholung der Deviceadresse in der Antwort
FF	CMDbyte	ERROR [u08]			Default "FF" für Fehler
??	Fehlerbyte	Errorcode [u08]			Errorcode
\$0D	Term.	Terminierung			Abschlusszeichen

5.1.3. Maximale Datenlänge der Übertragung

Befehle ohne feste Länge wie z.B. LCD Textausgabe dürfen nicht mehr als 127 Datenbytes (254 ASCII Zeichen high - Byte / low - Byte) nach dem Befehlsbyte (CMDbyte) folgen.

5.1.4. Typdefinitionen

Datentyp	Abkürzung
unsigned char	u08
signed char	s08
unsigned short	u16
signed short	s16
unsigned long	u32
signed long	s32
float	f32

5.1.5. **ERROR - Code**

Datentyp	Abkürzung
ERROR	0 x FF
NOERROR	0 x 00
NOCMD	0 x 01
SYNTAX	0 x 02
PARAMS	0 x 03
WERTEBEREICH	0 x 04
CMDLEN	0 x 05
CHECKSUM	0 x 06
TIMEOUT	0 x 07



5.1.6. MIO1616 Firmwareupdate über serielle Schnittstelle

Das MIO1616 besitzt eine Bootloader Funktion. Hierdurch können bei Bedarf über sie serielle Schnittstelle Firmwareupdates aufgespielt werden.

Vorgehensweise:

- 1. Mit Schalter S1 Bit 6 auf "ON" stellen. → LED1 (rot) leuchtet
- 2. Anschließen RESET Taste betätigen → LED2 (orange) leuchtet → Bootloader Modus aktiv
- 3. Die serielle Schnittstelle des MIO1616 muss nun an eine COM von 1..4 verbunden sein.
- 4. Programm "AVRprog.exe" starten. Dieses sucht nun im Bereich von COM 1..4 nach einer MIO1616 Steuerung.
- 5. Die neue Firmware über die Bedienoberfläche auswählen (Browse...).
- 6. Neue Firmware Flashen: Taste "Program" betätigen.
- 7. Es erscheint nun ein Fortschrittsbalken, der den Status der Programmierung anzeigt.
- 8. Nach erfolgreicher Programmierung kann das Tool geschlossen werden.
- 9. Mit Schalter S1 Bit 6 auf "OFF" stellen. → LED1 (rot) deaktiviert
- 10. Anschließen RESET Taste betätigen \rightarrow LED2 (orange) blinkt 3 x \rightarrow Bootloader Modus deaktiviert, Programmierung wurde erfolgreich abgeschlossen.



5.1.7. Befehlsimplementierung bei verschiedenen SW - Versionen

Befehl	Beschreibung	SW - Version			
		3.02			
10	EINGÄNGE + AUSGÄNGE lesen 16 Bit	Х			
11	ADW'S lesen 8 Bit / 10 Bit	Х			
12	EINGÄNGE: Triggerbedingung auf Eingangsmuster setzen	Х			
15	COUNTER - EINGÄNGE: Zähler / Pulszeit Kanal 14 initialisieren / lesen				
20	24 V AUSGABE byteweise setzen 16 Bit	Х			
21	24 V AUSGABE bitweise setzen 16 Bit	Х			
22	24 V AUSGABE bitweise blinken setzen 16 Bit				
24	24 V AUSGABE Kanal 1 + 2 PWM ausgeben	Х			
25	24 V INPUT Triggerschwelle mit DAW setzen 8 Bit	Х			
28	Mechanischen Ablauf über Timeout überwachen				
30	M5450 MCD serielle Karten steuern				
32	Universelle 3 - Wire Datenausgabe mit Chip - Select				
34	LCD initialisieren				
35	LCD Textausgabe	Х			
3A	EEPROM 1 schreiben max. 00FF = 256 BYTES (auf Controller)	Х			
3B	EEPROM 1 lesen max. 00FF = 256 BYTES (auf Controller)	Х			
3C	EEPROM universal schreiben (nicht PIC kompatibel)				
3D	EEPROM universal lesen (nicht PIC kompatibel)				
3F	SW - Version abfragen	Х			
40	I ² C extern Startbedingung ausgeben	Х			
41	I ² C extern Stoppbedingung ausgeben	Х			
42	I ² C extern Daten senden	Х			
43	I ² C extern Daten lesen	Х			
44	I ² C extern universelle Ein - Ausgabe (Execute)	Х			
46	PCF - PIO 8574 I ² C extern Daten senden	Х			
47	PCF - PIO 8574 I ² C extern Daten lesen	Х			
48	I ² C extern Tastendecoder Matrix 4 x 5 lesen	Х			
4A	EEPROM universal schreiben				
4B	EEPROM universal lesen	Х			
	Ausgänge auf Defaultwert nach Power Up setzen. Wird über entsprechende EEPROM Zellen gesetzt	Х			



5.1.8. Befehl 10: EINGÄNGE + AUSGÄNGE lesen 16 Bit

Die Eingänge werden zyklisch alle 10 ms erfasst.

ACHTUNG:

Kanal 1..4 immer feste Schwelle von High = 3,8 V; Low = 2,8 V, reagieren auf Flankenänderung und sind immer aktuell.

Befehl: Standard

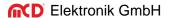
Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
02	LENbyte	Längenbyte [u08]		2255	
10	CMDbyte	Befehlscode [u08]			
\$0D	Term.	Terminierung			

Antwort: Standard

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse			
10	CMDbyte	Befehlscode			
??	Dbyte1	INPUTS [u16]	highByte	065535	
??	Dbyte2		lowByte		
??	Dbyte3	OUTUTS [u16]	highByte	065535	
??	Dbyte4		lowByte		
??	Dbyte5	OUTPUT - STATUS		0 = io	Status Diagnoseausgang der High
		[u08]		1 = Overload	Side Schalter Output 116
\$0D	Term.	Terminierung			

Befehl: Erweitert

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
04	LENbyte	Längenbyte [u08]		2255	
10	CMDbyte	Befehlscode [u08]			
??	Dbyte1	Config [u08]		1	tbd
\$0D	Term.	Terminierung			



Antwort: Erweitert

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse			
10	CMDbyte	Befehlscode			
??	Dbyte1	INPUTS [u16]	highByte	065535	
??	Dbyte2		lowByte		
??	Dbyte3	OUTUTS [u16]	highByte	065535	
??	Dbyte4		lowByte		
??	Dbyte5	OUTPUT - STATUS [u08]		0 = io 1 = Overload	Status Diagnoseausgang der High Side Schalter Output 116
??	DByte6	MERKERBITS [u16]	highByte	065535	tbd
??	DByte7		lowByte		
\$0D	Term.	Terminierung			

5.1.9. Befehl 11: ADW'S lesen 8 Bit / 10 Bit

Ist am MIO1616 die R - Codierung für den ADW eingestellt, so berechnet sich der Widerstandswert wie folgt:

Befehl: Standard ADW'S mit 8 Bit - Auflösung lesen

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		Alle druckbaren ASCII Zeichen	
02	LENbyte	Längenbyte [u08]		2255	
11	CMDbyte	Befehlscode [u08]			
\$0D	Term.	Terminierung			



Antwort: Standard ADW'S mit 8 Bit - Auflösung lesen

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
11	CMDbyte	Befehlscode			
??	Dbyte1	ADW1 [u08]		0255	8 Bit 0255 = 010 V
??	Dbyte2	ADW2 [u08]		0255	
??	Dbyte3	ADW3 [u08]		0255	
??	Dbyte4	ADW4 [u08]		0255	
\$0D	Term.	Terminierung			

Befehl: Erweitert ADW'S mit 10 Bit - Auflösung lesen

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse			
04	LENbyte	Längenbyte [u08]		2255	
11	CMDbyte	Befehlscode [u08]			
??	Dbyte1	Config [u08]		1	Es werden die ADW'S mit 10 Bit Auflösung ausgegeben
\$0D	Term.	Terminierung			

Antwort: Erweitert ADW'S mit 10 Bit - Auflösung lesen

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse			
11	CMDbyte	Befehlscode			
??	Dbyte1	ADW1 [u16]	highByte	01023	10 Bit 01023 = 010 V
??	Dbyte2		lowByte		
??	Dbyte3	ADW2 [u16]	highByte	01023	
??	Dbyte4		lowByte		
??	Dbyte5	ADW3 [u16]	highByte	01023	
??	DByte6		lowByte		
??	DByte7	ADW4 [u16]	highByte	01023	
??	DByte8		lowByte		



5.1.10. Befehl 12: EINGÄNGE: Triggerbedingung auf Eingangsmuster setzen

Das MIO1616 kontrolliert zyklisch die Eingänge und setzt nach Übereinstimmung mit der Triggerbedingung die entsprechenden Ausgangszustände bzw. das entsprechende Merkerflag [u16].

Die Priorität fällt mit steigendem Index. Das bedeutet, dass Index 0 immer den Ausgabezustand von z. B. Index 1 überschreibt.

Trigger Index 0 wird nach dem Power Up aus dem EEPROM geladen und hat höchste Priorität. Er kann mit Befehl 12 überschrieben werden, jedoch nicht dauerhaft im EEPROM abgelegt werden.

Permanentes ändern von Index 0 geschieht nur durch Beschreiben der entsprechenden EEPROM Zellen. Für schnelle Trigger (Impulse < 20 ms) sollten Kanal 1..4 verwendet werden (bis >= 100 µs).

ACHTUNG: Kanal 1..4 immer feste Schwelle von High = 3,8 V Low = 2,8 V.

Befehl: Standard

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
16	LENbyte	Längenbyte [u08]		2255	
12	CMDbyte	Befehlscode [u08]			
??	Dbyte1	TRIGGER_NR [u08]		07	Index zum Auswählen der Triggerbedingung
??	Dbyte2	INPUT - TRIGGER [u16]	highByte	065535	Zustand der zu bewertenden Eingänge
??	Dbyte3		lowByte		
??	Dbyte4	INPUT - MASK [u16]	highByte	065535	Auswahl der zu bewertenden Eingänge
??	Dbyte5		lowByte		
??	Dbyte6	OUTPUT - TRIGGER [u16]	highByte	065535	Zustand der zu stellenden Ausgänge
??	Dbyte7		lowByte		
??	Dbyte8	OUTPUT - MASK [u16]	highByte	065535	Auswahl der zu stellenden Ausgänge
??	Dbyte9		lowByte		
??	Dbyte10	STATUS [u08]		0 = off 1 = on	Triggerbedingung aktivieren / deaktivieren
\$0D	Term.	Terminierung			



Antwort: Standard

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
12	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			

5.1.11. EINGÄNGE: Trigger auf Eingangsmuster ein / aus 16 Bit über EEPROM Zelle

Die Steuerung der Triggerung auf Eingangsmuster ist aus Sicherheitsgründen für Trigger Index 0 über EEPROM Inhalte steuerbar. Bei aktiv gesetzter Triggerung (z.B. auf Notaus) wird dieser Zustand nun sofort nach dem Power Up aktiviert.

Für das Schreiben der EEPROM Zellen kann Befehl 3A oder 4A verwendet werden.

EEPROM Zellen 24c64 auf MIO1616 I²C-BUS (8 kByte)

EEPROM Adresse	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
0x0033	1	INPUT - DATA [u16]	highByte	0x00-0xFF	Zustand der zu bewertenden Eingänge
0x0034	2		lowByte	0x00-0xFF	
0x0035	3	INPUT - MASK [u16]	highByte	00x00-0xFF	Auswahl der zu bewertenden Eingänge
0x0036	4		lowByte	0x00-0xFF	
0x0037	5	OUTPUT - DATA [u16]	highByte	0x00-0xFF	Zustand der zu stellenden Ausgänge
0x0038	6		lowByte	0x00-0xFF	
0x0039	7	OUTPUT - MASK [u16]	highByte	0x00-0xFF	Auswahl der zu stellenden Ausgänge
0x003A	8		lowByte	0x00-0xFF	
0x003B	9	STATUS [u08]		0 = off 1 = on	Triggerung inaktiv nach Power Up Triggerung aktiv nach Power Up



5.1.12. Befehl 15: COUNTER - EINGÄNGE

Dieser Befehlt dient zum Initalisieren / Lesen von Zähler / Pulszeit Kanal 1..4, d.h. Messen von Zählerereignissen / Pulszeit für Kanal 1..4 bis 10 kHz.

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
10	LENbyte	Längenbyte [u08]		2255	
15	CMDbyte	Befehlscode [u08]			
??	Dbyte1	INIT[u08]		0 = initialisieren 1 = Status abfragen	Beim Status Abfragen werden Dbyte37 ignoriert, wegen Längebyte alle auf 0 x 00 setzen, werden nicht ausgewertet
??	Dbyte2	INPUT - CHANEL[u08]		03	Eingangskanal 14 auswählen
??	Dbyte3	MESSART [u08]	highNibble	0 = Standard- zähler 1 = Ereignis-	Zählerstand wird nach Power UP mit jedem Flankenwechsel um 1 erhöht Erst nach Freischalten wird gezählt
			Lavo A Clada	zähler	
			IowNibble	0 = all 1 = high	Flankenauswertung bei jedem Flankenwechsel
				2 = low	Flankenauswertung bei low → high Flankenwechsel
					Flankenauswertung bei high → low Flankenwechsel
??	Dbyte4	GATE - TIME [u32]	highByte	00xFFFFFF FF	Torzeit für Messung 32 Bit
??	Dbyte5				
??	Dbyte6				
??	Dbyte7		lowByte		
\$0D	Term.	Terminierung			



Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
12	CMDbyte	Befehlscode			
??	Dbyte1	STATUS [u08]		0 = ready 1 = running	Messung wurde gültig beendet Messung läuft noch, Startflanke wurde erkannt
??	Dbyte2	VALUE [u32]	highByte	00xFFFFFF FF	Zählerstand in [n] / Pulsdauer in [ms]
??	Dbyte3				
??	Dbyte4				
??	Dbyte5		IowByte		
\$0D	Term.	Terminierung			

5.1.13. AUSGABE – Zustand nach Power Up über EEPROM konfigurieren

Der Zustand der Ausgänge nach dem Power Up kann aus dem EEPROM geladen werden. Für das Schreiben der EEPROM Zellen kann Befehl 3A oder 4A verwendet werden.

EEPROM Zellen 24c64 auf MIO1616 I²C - BUS (8 kByte)

EEPROM	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
Adresse					
0x003D	1	OUTPUT - DATA	highByte	065535	Zustand der zu stellenden
		[u16]			Ausgänge
0x003E	2		lowByte		
0x003F	3	STATUS [u08]		0 = off	Inaktiv nach Power Up
				1 = on	Aktiv



5.1.14. Befehl 20: 24 V AUSGABE byteweise setzen 16 Bit

Alle Ausgänge gemeinsam stellen:

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
06	LENbyte	Längenbyte [u08]		2255	
20	CMDbyte	Befehlscode [u08]			
??	Dbyte1	OUTPUT [u16]	highByte	065535	Zustand der zu stellenden Ausgänge
??	Dbyte2		lowByte		
\$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
20	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			

5.1.15. Befehl 21: 24 V AUSGABE bitweise setzen 16 - Bit

Ausgänge Einzelbits stellen:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
08	LENbyte	Längenbyte [u08]		2255	
21	CMDbyte	Befehlscode [u08]			
??	Dbyte1	OUTPUT [u16]	highByte	065535	Auswahl der zu stellenden Ausgänge
??	Dbyte2		lowByte		
??	Dbyte3	STATUS [u08]		0 = off 1 = on	Zustand der zu stellenden Ausgänge
\$0D	Term.	Terminierung			



Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
21	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			

5.1.16. Befehl 22: 24 V AUSGABE Bitweise blinkend setzen 16 Bit

Standard:

Die Blinkfrequenz wird über EEPROM Zelle übergeben (in Intervallen von 10 ms einstellbar 10 ms..2,55 sec = 1..255)

Befehl: Standard

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
08	LENbyte	Längenbyte [u08]		2255	
22	CMDbyte	Befehlscode [u08]			
??	Dbyte1	OUTPUT [u16]	highByte	065535	Auswahl der zu blinkenden Ausgänge
??	Dbyte2		lowByte		
??	Dbyte3	STATUS [u08]		0 = off 1 = on	Blinken aktivieren / deaktivieren
\$0D	Term.	Terminierung			

Antwort: Standard

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
22	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			

EEPROM Zellen 24c64 auf MIO1616 I²C - BUS (8 kByte)

EEPROM Adresse	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
0x003C	1	Blinkrate [u08]	Byte	0x00-0xFF	Blinkfrequenz in 10 ms Intervall (02550 ms)



Erweitert:

Die Blinkfrequenz wird direkt im Befehl mit Parameter übergeben.

Befehl: Erweitert

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
0C	LENbyte	Längenbyte [u08]		2255	
22	CMDbyte	Befehlscode [u08]			
??	Dbyte1	OUTPUT [u16]	highByte	065535	Auswahl der zu blinkenden Ausgänge
??	Dbyte2		lowByte		
??	Dbyte3	BLINKRATE [u16]	highByte	065535	Hightime bzw Lowtime der Blinkfrequenz in [1ms]
??	Dbyte4		lowByte		
??	Dbyte5	STATUS [u08]		0 = off 1 = on	Blinken aktivieren / deaktivieren
\$0D	Term.	Terminierung			

Antwort: Erweitert

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
22	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			



5.1.17. Befehl 24: 24 V AUSGABE Kanal 1 + 2 PWM ausgeben

Kanal 1 + 2 besitzen die gleiche Zeitbasis und haben somit die gleiche Ausgangsfrequenz. Die PWM kann getrennt eingestellt werden.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
0C	LENbyte	Längenbyte [u08]		2255	
24	CMDbyte	Befehlscode [u08]			
??	Dbyte1	OUTPUT [u08]		0 = Ausgang1 1 = Ausgang2	Ausgang setzen
??	Dbyte2	DUTY [u16]	highByte	0100	0100 % Duty - Cycle in 1 % Steps
??	Dbyte3		lowByte		
??	Dbyte4	FREQUENZ [u16]	highByte	0250	0250 Hz (Begrenzt wegen HIGH Side Switches) in 1 Hz Steps
??	Dbyte5		lowByte		
??	Dbyte6	STATUS [u08]		0 = off 1 = on	PWM für gewählten Ausgang aktivieren / deaktivieren
\$0D	Term.	Terminierung			

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
24	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			



5.1.18. Befehl 25: 24 V INPUT Triggerschwelle mit DAW setzen 8 Bit

Triggerschwelle = 3,3 x DAW Wert (max. Schwelle von 10 V für Eingänge möglich)

ACHTUNG: Kanal 1..4 immer feste Schwelle von High = 3,8 V; Low = 2,8 V

Befehl: Standard

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
04	LENbyte	Längenbyte [u08]		2255	
25	CMDbyte	Befehlscode [u08]			
??	Dbyte1	DAW [u08]		0255	DAW Wert * 3,3 = Schaltschwelle für Bereich <= 10 V
\$0D	Term.	Terminierung			

Antwort: Standard

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
25	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			

5.1.19. Befehl 28: Mechanischen Ablauf über Timeout überwachen

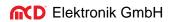
Die Bewegungsdaten müssen zuvor im EEPROM eingetragen werden (max. 32 Bewegungen, siehe EEPROM Liste unten).

Befehl 4A verwenden!

Hier wird nun die Reihenfolge übergeben, nach der die Bewegungen ausgeführt werden (max. 32 am Stück). **Ob** der komplette Ablauf ausgeführt wurde muss gepollt werden.

Ist Timeout = 0(0x0000) gesetzt, so wird der Eingangszustand nicht abgefragt

Ist Timeout = 65535(0xFFFF) gesetzt, so wird unendlich auf den Eingangszustand gewartet



Befehl:

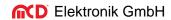
Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
08	LENbyte	Längenbyte [u08]		2255	
28	CMDbyte	Befehlscode [u08]			
??	Dbyte1	MODE [u08]		0 = initialisieren 1 = Status abfragen 2 = Ablauf abbrechen	Für Polling verwenden
??	Dbyte2	INDEX [u08]		032	Maximal 32 Einzelabläufe möglich
??					
??	DbyteN	INDEX [u08]			Insgesamt nicht mehr als 32 Indexe möglich
\$0D	Term.	Terminierung			

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
22	CMDbyte	Befehlscode			
??	Dbyte2	INDEX [u08]			Aktuelle laufender Index
??	Dbyte1	STATUS [u08]		0 = ready n = Abläufe offen 0xFF Timeout	Ablauf ohne Fehler beendet n Abläufe noch unbearbeitet
\$0D	Term.	Terminierung			



EEPROM Zellen 24c64 auf MIO1616 I²C-BUS (8 kByte)

EEPROM Adresse	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
0x0100	Index 1	Delay [u16]	highByte	065535 [ms]	Delay vor Ausführen der Bewegung
0x0101	Index 1		lowByte		
0x0102	Index 1	Timeout [u16]	highByte	065535 [ms]	Timeout für das Ausführen der Bewegung 65535 (0xFFFF) bedeutet unendlich lange warten
0x0103	Index 1		lowByte		
0x0104	Index 1	INPUT - DATA [u16]	highByte	065535	Zustand der zu bewertenden Eingänge
0x0105	Index 1		lowByte		
0x0106	Index 1	INPUT - MASK [u16]	highByte	065535	Auswahl der zu bewertenden Eingänge
0x0107	Index 1		lowByte		
0x0108	Index 1	OUTPUT - DATA [u16]	highByte	065535	Zustand der zu stellenden Ausgänge
0x0109	Index 1		lowByte		
0x010A	Index 1	OUTPUT - MASK [u16]	highByte	065535	Auswahl der zu stellenden Ausgänge
0x010B	Index 1		lowByte		
0x010C	Index 2	Delay [u16]			
0x0190	Index 12	Delay [u16]			
0x0280	Index 32	Delay [u16]			



5.1.20. Befehl 30: M5450 MCD Busschaltung ansteuern 34 Bit

Ansteuerung einer MCD - Systemkarte mit adressierbarer Busanschaltung.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
0E	LENbyte	Längenbyte [u08]		2255	
30	CMDbyte	Befehlscode [u08]			
??	Dbyte1	KARTEN - ADRESSE [u08]		063	MCD - BUS Kartenadresse
??	Dbyte2	DATA [u40]	highByte	02	Datenbyte 5 34 Bit Schieberegister Daten M5450
??	Dbyte3			0255	Datenbyte 4
??	Dbyte4			0255	Datenbyte 3
??	Dbyte5			0255	Datenbyte 2
??	Dbyte6		lowByte	0255	Datenbyte 1
\$0D	Term.	Terminierung			

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
30	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			



5.1.21. Befehl 32: 3 - Wire - Datenausgabe über I²C EXTERN Ports

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
08+ n*2	LENbyte	Längenbyte [u08]		2255	
32	CMDbyte	Befehlscode [u08]			
??	Dbyte1	CONFIG [u08]	highNibble	Chip Select	Data/WWW\
				0 =	CS\/
				1 =	CS/\
				2 =	CS_/
				3 =	CS/-\
				4 =	CS/_/
				5 =	CS\/-\
				6 =	CSlo
				7 =	CShi
??	Dbyte2	DATA [u08]	lowNibble	Direction 0 = MSB first 1 = LSB first	Bei gedrehter Datenausgabe (MSB oder LSB first) werden nur die Einzelbytes gedreht, nicht die kompletten Ausgabedaten bei 8+n Bits Aus 0 x 12 wird dann 0 x 48
??	Dbyte3	ANZBITS [u08]		1255	Anzahl Bits zum senden
??	Dbyte4			0255	
??				0255	
??	DbyteN		е	0255	
\$0D	Term.	Terminierung			

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
32	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			



5.1.22. Befehl 34: LCD Initialisieren

Das LCD wird mit der Hersteller - Grundinitialisierung eingestellt und kann anschließend beschrieben werden.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
02	LENbyte	Längenbyte [u08]		2255	
34	CMDbyte	Befehlscode [u08]			
\$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
34	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			

5.1.23. Befehl 35: LCD POS * TEXT ausgeben

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
04 + n	LENbyte	Längenbyte [u08]		2255	
35	CMDbyte	Befehlscode [u08]			
??	Dbyte1	COMMAND [u08]		0255	Kommando Byte für LCD siehe Datenblatt
?	Dbyte2	DATA [u08]		0x200x7F	Druckbares Zeichen
?		DATA [u08]		0x200x7F	
?	DbyteN	DATA [u08]		0x200x7F	
\$0D	Term.	Terminierung			



Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
35	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			

5.1.24. Befehl 3A: EEPROM 1 schreiben max. 00..FF = 256 BYTES (auf Controller)

Es wird auf das interne EEPROM 24c64 der MIO1616 geschrieben.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
06	LENbyte	Längenbyte [u08]		2255	
3A	CMDbyte	Befehlscode [u08]			
??	Dbyte1	ADRESSE [u08]		0255	EEPROM Adresse
??	Dbyte2	DATA [u08]		0255	
\$0D	Term.	Terminierung			

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
3A	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			



5.1.25. Befehl 3B: EEPROM 1 lesen max. 00..FF = 256 BYTES (auf Controller)

Es wird vom internen EEPROM 24C64 der MIO1616 gelesen.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
04	LENbyte	Längenbyte [u08]		2255	
3B	CMDbyte	Befehlscode [u08]			
??	Dbyte1	ADRESSE [u08]		0255	EEPROM Adresse
\$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
3B	CMDbyte	Befehlscode			
??	Dbyte1	DATA [u08]			EEPROM Inhalt der Adresse
\$0D	Term.	Terminierung			

5.1.26. Befehl 3F: SW VERSION Firmware lesen

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
02	LENbyte	Längenbyte [u08]		2255	
3F	CMDbyte	Befehlscode [u08]			
\$0D	Term.	Terminierung			



Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
3F	CMDbyte	Befehlscode			
??	Dbyte1	DATA [u08]			Textstring der Version
??		DATA [u08]			
??	Dbyte1	DATA [u08]			
\$0D	Term.	Terminierung			

5.1.27. I²C Bedienung allgemein

Das Mechanik - Interface arbeitet als Master. Der I²C - Bus wird auf der MIO1616 über Open - Collector - Pull - Up 4k7 Ohm an 5 V geklemmt. Clock - Stretching durch den Slave > 1 ms wird nur mit Befehl 44 unterstützt.

5.1.28. Befehl 40: I²C EXTERN START - Bedienung ausgeben

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
02	LENbyte	Längenbyte [u08]		2255	
40	CMDbyte	Befehlscode [u08]			
\$0D	Term.	Terminierung			

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
40	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			



5.1.29. Befehl 41: I²C EXTERN STOP - Bedingung ausgeben

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
02	LENbyte	Längenbyte [u08]		2255	
41	CMDbyte	Befehlscode [u08]			
\$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
41	CMDbyte	Befehlscode			
\$0D	Term.	Terminierung			

5.1.30. Befehl 42: I²C EXTERN DATENBYTES ausgeben

Max 255 Bytes auf den I²C - Bus schreiben.

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
04 + 2*n	LENbyte	Längenbyte [u08]		2255	
42	CMDbyte	Befehlscode [u08]			
??	Dbyte1	DATA [u08]			Max. 255 Bytes Daten
??		DATA [u08]			
??	DbyteN	DATA [u08]			
\$0D	Term.	Terminierung			



Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
42	CMDbyte	Befehlscode			
??	Dbyte1	EERORSTATUS [u08]		0 = io 1 = ACK - Error 2 = Clockstretch Timeout	Error Status für I ² C schreiben
\$0D	Term.	Terminierung			

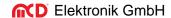
5.1.31. Befehl 43: I²C EXTERN DATENBYTES einlesen

Zwischen den Bytes wird I²C - ACK ausgegeben. Nach dem letzten Byte "NOACK".

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
02	LENbyte	Längenbyte [u08]		2255	
43	CMDbyte	Befehlscode [u08]			
??	Dbyte1	ANZAHL [u08]		1255	Anzahl Bytes zum Lesen
\$0D	Term.	Terminierung			

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
43	CMDbyte	Befehlscode			
Opti onal	Dbyte1	EERORSTATUS [u08]		0 = io	Error Status nicht ausgewertet
??	Dbyte2	DATA [u08]		0255	
??				0255	
??	DbyteN	DATA [u08]		0255	
\$0D	Term.	Terminierung			



5.1.32. Befehl 44: I²C EXTERN universelle Ein - / Ausgabe (Execute)

Universelles Bedienen der I²C - Schnittstelle. Die Kommandozeile wird interpretiert und entsprechend ausgegeben. Dieser Befehl sollte für **neue Applikationen** verwendet werden und **ersetzt Befehl 40, 41, 42, 43.**

Folgende Parameter stehen zur Verfügung:

Mit Data:='ISA000SA1Q07RP' bewirkt das Kommando folgendes:

I SDA, SCL Ports initialisieren

Cn Clockstretch einstellen: 0 = 1 ms; 1 = 10 ms; 2 = 100 ms; 3 = 1000 ms; 4 = 3000 ms

Nach dem Ausführen von Befehl 44 ist Clockstretch wieder 1 ms

S Startbedingung erzeugen

Nach Startbedingung senden von Daten

A0 Byte A0 senden (Adressierung des I²C - Bausteines - Schreibzugriff)

00 Byte 00 senden (zweiter Teil der Adresse)

S Repeated Startbedingung erzeugen

Nach Repeated Startbedingung senden von Daten

A1 Byte A1 senden (Adressierung des I²C - Bausteines - Lesezugriff)

Qnn Byte lesen mit Ack (insgesamt nn mal)

R Byte lesen ohne Ack - signalisiert Ende des Lesens

P Stoppbedingung erzeugen

Mit Data: = 'ISA000SA1Q07RP' bewirkt das Kommando folgendes:

Die Executefunktion liest 8 Byte vom I²C Baustein mit Adresse A0 ein.

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
??	LENbyte	Längenbyte [u08]		2255	
44	CMDbyte	Befehlscode [u08]			
??	Dbyte1	[u08]			
??		[u08]			
??	DbyteN	[u08]			
\$0D	Term.	Terminierung			



Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
12	CMDbyte	Befehlscode			
	Dbyte1	EERORSTATUS [u08]		0 = io 1 = ACK - Error 2 = Clockstretch Timeout	I ² C Errorstatus
	Dbyte N	[u08]			Je nach Kommandozeile erfolgt die Antwort
\$0D	Term.	Terminierung			

5.1.33. Befehl 46: PCF8574 I²C EXTERN bitweise setzen 8 Bit

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
08	LENbyte	Längenbyte [u08]		2255	
46	CMDbyte	Befehlscode [u08]			
??	Dbyte1	HW ADRESSE [u08]		0255	PCF8574 I ² C HW Adresse HW Adresse A0, A1, A2 * 2 multiplizieren bei Übergabe
??	Dbyte2	OUTPUT [u08]		0255	Auswahl der zu stellenden Ausgänge
??	Dbyte3	STATUS [u08]		0 = clr 1 = set Bit 2 = set Byte direkt	Zustand der zu stellenden Ausgänge Einzelne Bits setzen / löschen Das komplette Ausgabebyte wird gesetzt wie in OUTPUT angegeben
\$0D	Term.	Terminierung			



Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
46	CMDbyte	Befehlscode			
??	Dbyte1	EERORSTATUS [u08]		0 = io 1 = ACK - Error 2 = Clockstretch Timeout	Error Status für I ² C schreiben
\$0D	Term.	Terminierung			

5.1.34. Befehl 47: PCF8574 I²C EXTERN lesen 8 Bit

Aus Kompatibilitätsgründen wird der I²C Errorstatus hier nicht ausgegeben.

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
04	LENbyte	Längenbyte [u08]		2255	
47	CMDbyte	Befehlscode [u08]			
??	Dbyte1	HW - ADRESSE [u08]		0255	PCF8574 I ² C HW Adresse HW Adresse A0, A1 ,A2 * 2 multiplizieren bei Übergabe
\$0D	Term.	Terminierung			

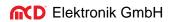


Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
47	CMDbyte	Befehlscode			
??	Dbyte1	EERORSTATUS [u08]		0 = io 1 = ACK - Error 2 = Clockstretch Timeout	Error Status für I ² C schreiben
??	Dbyte2	DATA [u08]		0255	
??				0255	
??	DbyteN	DATA [u08]		0255	
\$0D	Term.	Terminierung			

5.1.35. Befehl 48: I²C EXTERN TASTENDECODER MATRIX 5 x 4 einlesen

Tastendecoder mit PCF8574

```
+---+
7+---(x)------+
 6+---( x)------+
 5+---( x)-----+
 4+---( x)-----+ ;
       |20 |16 |12 |8
 3+---( x)----o-----o----o-
    |19 |15 |11 |7 |3
 2+---(x)----o----o----o-
       1+---( x)----o----o-
       17
           ¦13
              ¦ 9
                  |5 |1
| 0+---(x)----o----o----o-
PCF8574 GND
```



Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
04	LENbyte	Längenbyte [u08]		2255	
48	CMDbyte	Befehlscode [u08]			
??	Dbyte1	HW - ADRESSE [u08]		0255	PCF8574 I ² C HW Adresse HW Adresse A0, A1, A2 * 2 multiplizieren bei Übergabe
\$0D	Term.	Terminierung			

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
48	CMDbyte	Befehlscode			
??	Dbyte1	EERORSTATUS [u08]		0 = io 1 = ACK - Error 2 = Clockstretch Timeout	Error Status für I ² C schreiben
??	Dbyte2	TASTENCODE[u08]		0x000x20	
\$0D	Term.	Terminierung			



5.1.36. Befehl 4A: EEPROM universal schreiben (nicht kompatibel)

Es können das interne EEPROM des ATMEL Controllers (4K), sowie das Expansions - EERPOM 24C64 (8K) auf der MIO1616 angesprochen werden. Zusätzlich kann auf ein beliebiges am externen I²C - BUS angeschlossenes EEPROM zugegriffen werden. Max. 32 Byte am Stück schreiben. Pagesize des verwendeten EEPROM beachten.

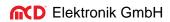
Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
0A +2*n	LENbyte	Längenbyte [u08]		2255	
4A	CMDbyte	Befehlscode [u08]			
??	Dbyte1	CONFIG [u08]	highNibble	0 = ext 1 = int 2 =	EEPROM 24c am externen I ² C - BUS EEPROM 24c64 auf MIO1616 I ² C
				Controller	- BUS (8 kByte) 16 Byte Page
					EEPROM ATMEL (8 kByte) 32 Byte Page
			lowNibble	0 = 8 Bit I ² C Adresse	Ab 24c64 zu verwenden
				1 = 16 Bit I ² C Adresse	
??	Dbyte2	HW - ADRESSE [u08]		0255	I ² C HW Adresse des EEPROM (0x00 für ATMEL EEPROM setzen)
					EEPROM 24c64 auf MIO1616 I ² C - BUS (8 kByte) 16 Byte Page = 0 x A0
??	Dbyte3	HighByte - ADRESSE [u08]		0255	
??	Dbyte4	LowByte - ADRESSE [u08]		0255	
??	Dbyte5	DATA [u08]		0255	
??				0255	
??	DbyteN	DATA [u08]		0255	
\$0D	Term.	Terminierung			



Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
4A	CMDbyte	Befehlscode			
??	Dbyte1	EERORSTATUS [u08]		0 = io 1 = ACK - Error 2 = Clockstretch Timeout	Error Status für I ² C EEPROMS
\$0D	Term.	Terminierung			

5.1.37. Befehl 4B: EEPROM universal lesen (nicht PIC kompatibel)

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		0x200x7F	
0C	LENbyte	Längenbyte [u08]		2255	
4B	CMDbyte	Befehlscode [u08]			
??	Dbyte1	CONFIG [u08]	highNibble	0 = ext 1 = int 2 = Controller	EEPROM 24c am externen I ² C - BUS EEPROM 24c64 auf MIO1616 I ² C - BUS (8 kByte) 16 Byte Page
					EEPROM ATMEL (8 kByte) 32 Byte Page
			lowNibble	0 = 8 Bit I ² C Adresse 1 = 16 Bit I ² C Adresse	Ab 24c64 zu verwenden
??	Dbyte2	HW - ADRESSE [u08]		0255	I ² C HW Adresse des EEPROM (0 x 00 für ATMEL EEPROM setzen)
					EEPROM 24c64 auf MIO1616 I ² C - BUS (8 kByte) 16 Byte Page = 0 x A0
??	Dbyte3	HighByte - ADRESSE [u08]		0255	
??	Dbyte4	LowByte - ADRESSE [u08]		0255	



??	Dbyte5	ANZAHL [u08]	1255	Anzahl Bytes zum Lesen
\$0D	Term.	Terminierung		

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
\$12	Trig.	Triggerzeichen			
?	Addr Nibble	Deviceadresse		\$20\$7E	
4B	CMDbyte	Befehlscode			
??	Dbyte1	EERORSTATUS [u08]		0 = io 1 = ACK - Error 2 = Clockstretch Timeout	Error Status für I ² C EEPROMS
??	Dbyte2	DATA [u08]		0255	
??				0255	
??	DbyteN	DATA [u08]		0255	
\$0D	Term.	Terminierung			



6. Technische Daten

Allgemein		
Betriebsspannung	1032 V	Polarität beachten!
		Getrennte Zuführung der Betriebsspannungen für Ausgang 1 bis 8, Ausgang 9 bis 16, Versorgung der Steuerelektronik
Ruhestromaufnahme	<= 100 mA	Zusätzlich Laststrom
Schnittstellen		
Serielles Protokoll	RS232 19200; 57600; 115200; Baud, 8 Daten, No Parity, 1 Stop	Mit S1 einstellbar
Adressierung	Max. 8 Interfaces adressierbar	Adr. 07 mit S1 einstellbar
Serielle Daten INPUT	Serielle Daten vom PC	
USB	Serielle Daten vom PC	FTDI Serielles COM - Device
Serielle Daten OUTPUT	Weiterleitung der Daten an zusätzliche IO - Karten	Konfigurierbar RS232 / RS485
LCD / IO Anschluss	LC - Displays, IIC BUS, 3 - Wire	Textausgabe max. 40 Charakter pro Befehl Max. Spannungsbereich an den IO Ports 5 V
Daten Ausgänge		
Maximale Summen Ausgangsstrom Kanal 18	4 A	Gesichert über Polyfuse 4A
Maximale Summen Ausgangsstrom Kanal 916	4 A	Gesichert über Polyfuse 4A
Maximale Ausgangsstrom pro Kanal	0,7 A	Kanal 116 Kurzschluss / Überlastfest
2 x PWM Ausgänge alternativ	10500 Hz / 0100 %	Kanal 1+2
		Standard Ausgabe alle 1 ms
Daten Eingänge		
Eingangsspannungsbereich	Triggerschwelle 110 V	Triggerschwelle 010 V einstellbar
Kanal 116		Werden alle 10 ms eingelesen
		Eingangswiderstand = 27 k Ω
Schnelle Triggereingänge Kanal 14		Eingänge als schnelle Ereigniszähler / Trigger bis 10 kHz Triggerschwelle Vih = 3,8 V
		ViI = 2,8 V
		Eingangswiderstand = 20 k Ω
Versorgung der Initiatoren	Zusammen mit Steuerelektronik	Gesichert über Polyfuse 4 A
Analogeingänge	Konfigurierbar als	Mit Jumper J1J4 einstellbar
	Konstantstromquelle	Auflösung 8 Bit Standard,
	(1 mA \pm 5 %) zur Codierung mit Widerständen (max. 22 k Ω).	10 Bit mit erweitertem Befehlssatz
	2. Analogeingang 010 V	Eingangswiderstand = 200 k Ω
Abmessungen		
Breite / Tiefe / Höhe	175 mm / 125 mm / 70 mm	
Befestigung	Auf Hutschiene 35 mm	
5 5		