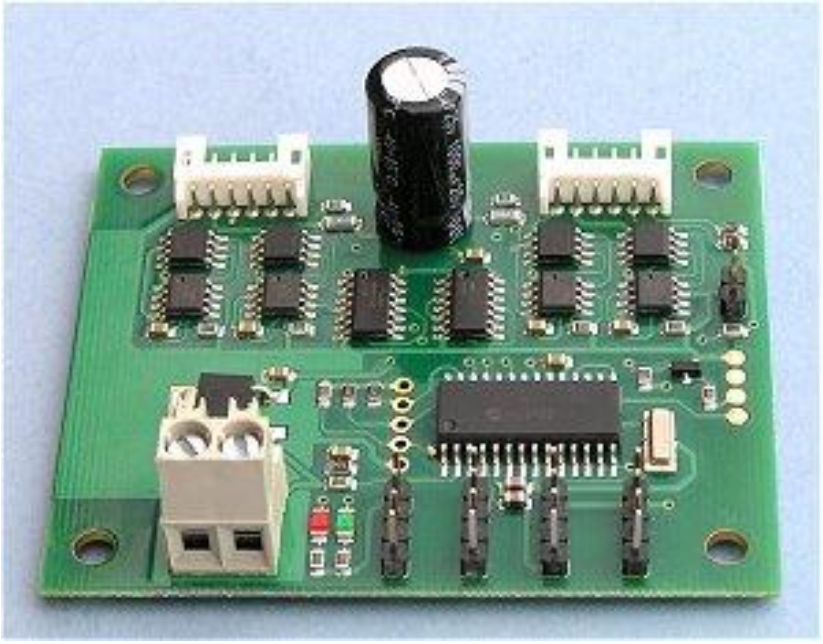
Datenblatt und Doku zu

Motorboard MD23

Inteligentes Motorboard (12V/6A) mit I2C-Bus



Spezielle intelligente Roboter-Motoransteuerung für 2 Motoren mit jeweils bis zu 3A Dauerbelastung. Ausgelegt für eine Betriebsspannung von 12 Volt und ideal passend zu unseren EMG30-Motoren mit eingebautem Drehgeber, Stecker paßt direkt in das Board. Aber auch für andere Motoren geeignet.

Angesteuert wird der Motortreiber über den beliebten I2C-Bus. Dabei sind zwei Betriebsarten möglich.

Entweder die manuelle Methode wobei man beide Motoren unabhängig voneinander über den I2C-Bus ansteuert. Die Zahlimpulse kann man in dieser Betriebsart über Register abrufen.

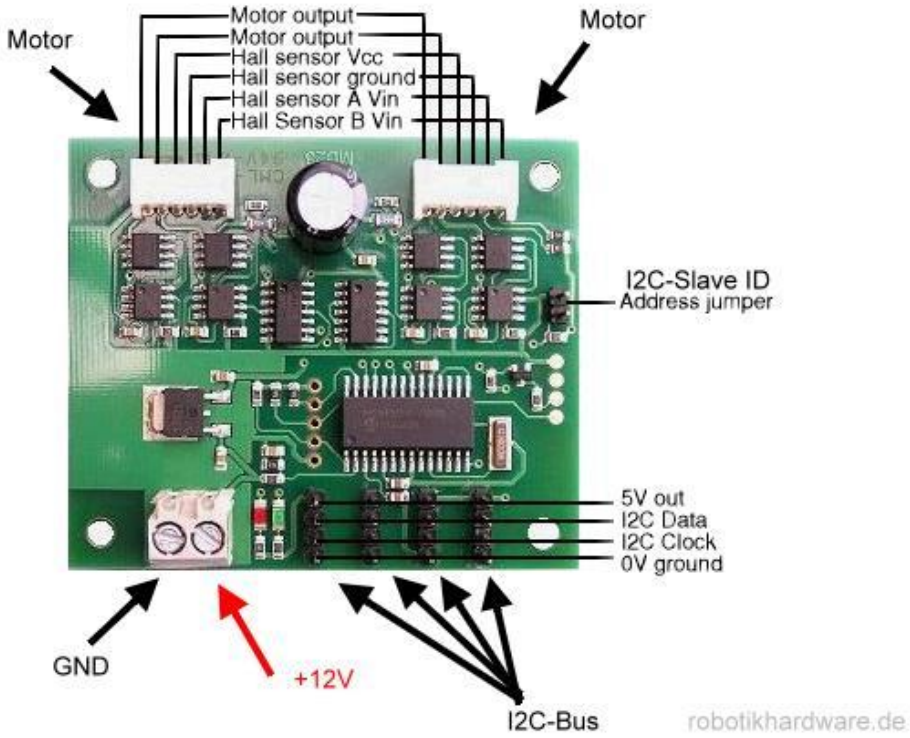
Oder aber eine Art intelligenter Modi bei dem man nur Geschwindigkeit und Lenkinformationen übermittelt und sich der Motor selbst um die Steuerung und Drehzahlregelung kümmert. Dabei sind verschiedene Verzögerungen (Rampen) einstellbar. Auch der Motorstrom kann über den I2C-Bus abgerufen werden.

Die wichtigsten Features:

- Steckerkompatibel zu unseren EMG Motoren mit eingebautem Drehgeber (siehe robotikhardware.de)
- Betriebsspannung: ca. 12V
- 5V Logikspannung über Onboard Spannungsregler
- 5V/300mA (max. 1A Spitze) für externe Komponenten verfügbar
- Ansteuerung von 2 Motoren (bis 3A Dauerstrom)
- Auslesen eines Quadratur-Encoder-Motorgeber
- Drehzahl/Zählergebnisse über I2C-Register auslesbar
- 2 Betriebsarten:
  direkte Steuerung
  oder
  Vorgabe der Geschwindigkeit + Lenkinformation (open loop, closed loop)
- Watchdog, nach 2s ohne I2C Kommunikation wird Motor gestoppt (schaltbar)
- Vier I²C-Anschlüsse (einfache 4er Stifleiste)
- Acht I²C-Adressen über Software wählbar (Standard: 0xB0), Alternativadresse einfach über Jumper setzbar
- Modul fertig aufgebaut und getestet
- Abmessungen: 72mm x 59mm x 25mm

Das Bild zeigt das Board mit angeschlossenen EMG Motoren

Die Anschlüsse des Modules:



I2C Anschluß

Der I2C Anschluß ist als 4 polige Stiftleiste ausgelegt (siehe Bild). Es müssen drei Leitungen SDA, SCL und GND mit dem zu steuernden Microcontrollerboard verbunden werden. Die RN-Boards´s führen diese Leitungen gewöhnlich auf einem 10 poligen Stecker. Man kann die Leitungen aber oft auch manuell leicht verdrahten, siehe das Bild auf den nächsten Seite (RN-Control und MD23).

Das Motorboard bietet vier I2C Anschlüsse, diese sind parallelgeschaltet und können zur Verbindung von weiteren I2C Boards genutzt werden.

Motorspannung

Die Motorsteuerung ist für eine 12V Batterie bzw. 12V Akku ausgelegt. Praktisch darf die Spannung zwischen 9 und 14 Volt liegen. Achtung, Spannungen über 14 Volt zerstören das Board!

Motorentstörung

Es wird empfohlen Motoren durch einen 10n Kondensator zu entstören. Wenn unsere Motoren EMG30 benutzt werden ist dies nicht notwendig , da diese bereits einen 10n Kondensator eingebaut haben. Durch die Entstörung werden Beeinflussungen des Motorboardes oder des Zählvorganges (Drehzahl) verhindert.

Leds

Die rote LED zeigt an das die Betriebsspannung anliegt.

Die grüne LED gibt bei Slave ID Änderungen die eingestellte I2C Adresse bekannt. Weiter hinten in der Doku finden sie näheres dazu.

Ansonsten blinkt die grüne Led kurz auf wenn I2C Befehle empfangen werden.

Automatische Geschwindigkeitsregelung

Benutzt man Motoren mit Drehgeber (wie z.B. unsere EMG30) Motoren, so sorgt die eingebaute Geschwindigkeitsregelung stets für konstante Geschwindigkeit. Das bedeutet das stärker belastete Motoren (z.B. an Steigung) automatisch mit mehr Power versorgt werden.

Automatischer Motorstop

Wenn über 2 Sekunden keine I2C Kommunikation erfolgt, so stoppt das Motorboard aus Sicherheitsgründen automatisch beide Motoren.

Über ein Register kann diese Option jedoch auch abgeschaltet werden.

Steuerung des MD23

Die Steuerung erfolgt über einen Standard I2C Bus wie Ihn zum Beispiel fast jedes RN-Board (aber auch andere) besitzt. Die Standard I2C-Slave Adresse ist Hex B0. Sie läßt sich über einen Jumper oder per Software auch auf BE ändern.

Register	Name	Read/Write	Beschreibung
0	Speed1	Lesen und Schreiben	Motorgeschwindigkeit Motor 1 (oder beider motoren, je nach Mode)
1	Speed2/Turn	Lesen und Schreiben	Motorgeschwindigkeit Motor 2
2	Enc1a	Nur lesen	Byte 1 Drehgeber 1 Position (höchste Byte)
3	Enc1b	Nur lesen	Byte2 Drehgeber 1 Position
4	Enc1c	Nur lesen	Byte3 Drehgeber 1 Position
5	Enc1d	Nur lesen	Byte4 Drehgeber 1 Position
6	Enc2a	Nur lesen	Byte 1 Drehgeber 2 Position (höchste Byte)
7	Enc2b	Nur lesen	Byte 2 Drehgeber 2 Position
8	Enc2c	Nur lesen	Byte 3 Drehgeber 2 Position
9	Enc2d	Nur lesen	Byte 4 Drehgeber 2 Position
10	Battery volts	Nur lesen	Batteriespannung Wert / 10 = Volt
11	Motor 1 current	Nur lesen	Aktueller Strom Motor 1 Wert / 10 = Ampere
12	Motor 2 current	Nur lesen	Aktueller Strom Motor 2 Wert / 10 = Ampere
13	Softw. Revision	Nur lesen	Firmware Version (Revision)
14	Acceleration rate	Lesen und Schreiben	Optionales Beschleunigungs Register
15	Mode	Lesen und Schreiben	Betriebsmodus
16	Command	Lesen und Schreiben	Spezielle Commandos wie Rücksetzen der Encoder-Zähler auf 0 Befehl Hex 20: Encoders auf 0 setzen

Erläuterung der Register

Register 0 (Speed1 Register)

Je nachdem in welchem Mode sich das Board befindet, hat dieses Register unterschiedliche Aufgaben.

Im Betriebsmode 0 und 1: Geschwindigkeit und Richtung von Motor 1. Je höher der Wert je schneller!
Im Betriebsmode 2 und 3: Geschwindigkeit und Richtung beider Motoren

Resister 1 (Speed2/Turn Register Lenkinformation)

Je nachdem in welchem Mode sich das Board befindet, hat dieses Register unterschiedliche Aufgaben.

Im Betriebsmode 0 und 1: Geschwindigkeit und Richtung von Motor 2. Je höher der Wert je schneller!
Im Betriebsmode 2 und 3: Turn-Wert (Lenkinformation)

Turn mode (Lenkregister)

Der Turn-Wert legt fest wann sich ein Motor bei einem geschwindigkeitswert vorwärts oder rückwärts bewegt.

So berechnet sich die Geschwindigkeit bei Vorwärtsrichtung:

motor speed1 = speed - turn

motor speed2 = speed + turn

So berechnet sich die Geschwindigkeit bei Rückwärtsrichtung:

motor speed1 = speed + turn

motor speed2 = speed - turn

Wenn ein Motor nicht in der Lage ist die Höchstgeschwindigkeit für die Wende zu erreichen, dann wird automatisch der zweite Motor entsprechend in der geschwindigkeit gedrosselt, so das das Verhältnis wieder stimmt.

Register 2 bis 9 (die Encoder Register)

Die Umrehungen des Drehgebers werden intern als Long-Wert (4 Byte) gezählt. Es müssen demnach auch alle 4 Bytes ausgelesen werden um eine 32 Bit zahl darzustellen. Man kann die Registe rauf 0 setzen indem man den Wert Hex 20 in das Command Register schreibt.

Register 10 (Batteriespannung)

Dieses Register gibt die Batteriespannung mit einer Nachkommastelle zurück. Der Wert 121 würde beispielsweise 12.1 Volt bedeuten.

Register 11 und 12 (Motorstrom Register)

Die Motorstrom Register geben den Strom in Ampere mit einer Nachkommastelle zurück. Der Wert 25 bedeutet beispielsweise 2,5 Ampere.

Register 13 (Software Version /Revision)

Die Versionsnummer der Firmware im Board. Zum Zeitpunkt als die Doku geschrieben wurde, war das Revision 3 !

Register 14 (Beschleunigungsregister)

Über das Beschleunigungsregister wird festgelegt wie schnell ein Motor eine Geschwindigkeit auf eine neue Geschwindigkeitsvorgabe oder auf einen Richtungswechsel vornimmt. Der Wert kann zwischen 1 und 10 liegen.

Die genaue Berechnung in Millisekunden wird in der englischen Doku näher erläutert. Hier eine Tabelle mit einigen Beispielwerten:

Beschleunigungs-register	Time/step	Aktuelle Geschwindigkeit	Neue Geschwindigkeit	Schritte	Beschleunigungs-zeit
1	18ms	0	255	255	4.59s
2	18ms	127	255	64	1.152s
3	18ms	80	0	27	0.486s
5 (default)	18ms	0	255	51	0.918s
10	18ms	255	0	26	0.468s

Register 15 (Mode Register)

Das Mode Register legt die Betriebsart des Boards fest. Je nach Mode sind einige Register unterschiedlich zu handhaben.

Die Modis:

Mode 0 (Standard-Modus) :
Der Geschwindigkeitswert wird wie folgt intepretiert:
0-127 Rückwärtsdrehung / 128 Stop / 129-255 Vorwärtsdrehung

Mode 1 :

Ist identisch mit mode 0, nur das hier der Geschwindigkeitswert als vorzeichenbehaftet interpretiert wird.

Der Geschwindigkeitswert wird dadurch wie folgt intepretiert:
-1 bis -128 Rückwärtsdrehung / 0 Stop / 1-127 Vorwärtsdrehung

Mode 2:

In dieser Betriebsart werden beide Motoren über das Register 0 (Speed Register) in der Geschwindigkeit gesteuert. Die Lenkbewegung erfolgt für das Register 2.

Der Geschwindigkeitswert wird wie folgt intepretiert:
0-127 Rückwärtsdrehung / 128 Stop / 129-255 Vorwärtsdrehung

Mode 3:

Ist identisch mit mode 2, nur das hier der Geschwindigkeitswert als vorzeichenbehaftet interpretiert wird.

Der Geschwindigkeitswert wird dadurch wie folgt intepretiert:
-1 bis -128 Rückwärtsdrehung / 0 Stop / 1-127 Vorwärtsdrehung

Register 16 (Command Register)

Über das Command-Register können verschiedene Funktionen ausgelöst werden. Einfach das entsprechende Befehlsbyte in das Register schreiben

Befehl		
Dec	Hex	Funktion
32	20	Die Encoder (Drehgeber) werden auf 0 zurückgesetzt
48	30	Die automatische Geschwindigkeitsregelung wird abgeschaltet
49	31	Die automatische Geschwindigkeitsregelung wird eingeschaltet (Voreinstellung)
50	32	Schaltet den automatischen 2 Sekunden I2C Timeout aus
51	33	Schaltet den automatischen 2 Sekunden I2C Timeout ein (Voreinstellung)
160	A0	1 Wert zum Ändern der I2C Slave ID
170	AA	2 Wert zum Ändern der I2C Slave ID
165	A5	3 Wert zum Ändern der I2C Slave ID

Ändern der I2C Slave ID

Die I2C-Sve Adresse kann durch einen Jumper als auch durch schreiben einer Befehlssequenz in das Command-Register geändert werden.

Um die Adresse per Command Register zu ändern, müssen folgende vier Bytes nacheinander in das Command Register geschrieben werden:

A0 / AA / A5 / Neue Slave ID

Um die Adresse Hex B4 hinein zu schreiben, müssten also folgende Bytes an Register 16 gesendet werden:

A0 / AA / A5 / B4

Beim Einschalten des Boards wird die jeweils eingestellte I2C-Slave ID durch ein langes Blinkzeichen und mehrere kleine Blinkzeichen zurück gegeben. In der nachfolgenden Tabelle ersieht man die Bedeutung dieser Blinkzeichen.

I2C Slave (Dezimal)	I2C Slave (hex)	Langes Blinken	Kurzes Blinken
176	B0	1	0
178	B2	1	1
180	B4	1	2
182	B6	1	3
184	B8	1	4
186	BA	1	5
188	BC	1	6
190	BE	1	7

I2C Slave ID Jumper

Man kann die i2C Adresse auch durch entfernen des Jumpers und anschließendem Neustart (Spannung kurz unterbrechen) verändern. Und zwar wird bei jedem neustart ohne Jumper automatisch die nächst höhere Adresse gewählt. Da Standard B0 ist, würde nach dem ersten Neustart B2 eingestellt. Ein weiterer Neustart (ohne Jumper) würde B4 einstellen. Sobald der Jumper wieder gesteckt wird, bleibt die I2C Slave ID auch nach Neustart unveränderlich.

Beispielprogramme

Auf den nachfolgenden Seiten haben wir eine Beispielprogramme bereitgestellt, welche die Steuerung des Boardes mit dem Controllerboard RN-Control zeigt. Natürlich lassen sich diese Bascom Beispiele leicht auch auf andere Controller übertragen.

Die abgedruckten Beispiele als auch einige weitere sind natürlich auch auf der CD von Robotikhardware.de zu finden.

Das erste Beispielprogramm demonstriert einige Grundfunktionen des Boardes. Die tasten von RN-Control werden mit folgenden Funktionen belegt:

Taste 1	Firmware Version des Motorboards MD23 auslesen und über RS232 ausgeben und I2C Timeout abschalten (damit Motoren nicht stoppen wenn keine Befehle gesendet werden)
Taste 2	Spannung am Motorboard MD23 messen und über RS232 ausgeben
Taste 3	Motor 1 volle Geschwindigkeit Mode 0
Taste 4	Motor 2 volle Geschwindigkeit Mode 0
Taste 5	Motor 1 und 2 stoppen

```
'#################################################################
'md23test.bas
'
'Das Programm demonstriert wie das Motorboard MD23
'über I2C von RN-Control angesteuert wird
'
'Die tasten haben in dem Demo folgende Funktion:
'Taste 1: Firmware Version des Motorboards MD23 auslesen und über RS232 ausgeben
'         und I2C Timeout abschalten (damit Motoren nicht stoppen wenn keine Befehle gesendet
werden)
'Taste 2: Spannung am Motorboard MD23 messen und über RS232 ausgeben
'Taste 3: Motor 1 volle Geschwindigkeit Mode 0
'Taste 4: Motor 2 volle Geschwindigkeit Mode 0
'Taste 5: Motor 1 und 2 stoppen
'
'Autor: Frank (roboternetz.de)
'Verwendet wurden: RN-Control & MD23 (robotikhardware.de)
'Weitere Beispiele sind im Roboternetz gerne willkommen!
'#################################################################

$programmer = 12                                            'MCS USB  (Zeile weglassen wenn
anderer Programmer)




'  -------------- RN-Control übliche _____________________
Declare Function Tastenabfrage() As Byte

$regfile = "m32def.dat"
$framesize = 32
$swstack = 32
$hwstack = 64

$crystal = 16000000                                         'Quarzfrequenz
$baud = 9600
Config Scl = Portc.0                                        'Ports fuer IIC-Bus
Config Sda = Portc.1

Config Adc = Single , Prescaler = Auto                      'Für Tastenabfrage und
Spannungsmessung
Config Pina.7 = Input                                       'Für Tastenabfrage
Porta.7 = 1                                                 'Pullup Widerstand ein
Dim Taste As Byte
Dim Ton As Integer

I2cinit
Start Adc
Sound Portd.7 , 400 , 450                                   'BEEP
Sound Portd.7 , 400 , 250                                   'BEEP
Sound Portd.7 , 400 , 450                                   'BEEP
Print
Print "**** RN-CONTROL  V1.4 *****"
Print "Demoprogramm zum Motorboard MD23"
Print "RN-Control als auch MD23 gibts bei robotikhardware.de"
Print
'  -------------- Ende RN-Control übliche _____________________


Declare Function Md23_version() As Byte
Declare Function Md23_batteriespannung() As Single
Declare Sub Md23_writeregister(byval Register As Byte , Byval Wert As Byte)

Const Md23_slaveid = &HB0
Dim S As String * 15
Dim F As Single
```

```
Do
   Taste = Tastenabfrage()
   If Taste <> 0 Then

      Select Case Taste

         Case 1:
                Print "MD23 Firmware Version:" ; Md23_version()

                Md23_writeregister 16 , &H32               'Automatische Timout Abschaltung
deaktivieren


         Case 2
                F = Md23_batteriespannung()
                F = F / 10
                Print "Die gemessene Batteriespannung beträgt:" ; F

         Case 3
                Md23_writeregister 0 , 255


         Case 4
                Md23_writeregister 1 , 255


         Case 5
                Md23_writeregister 0 , 128
                Md23_writeregister 1 , 128


      End Select
      Sound Portd.7 , 400 , 500                            'BEEP
   End If

   Waitms 100
Loop

End



' Diese Unterfunktion fragt die Tastatur am analogen Port ab
' Sollte beim betätigen einer Taste kein Quittungston kommen, dann
' muss die die Tastenabfrage (Select Case Anweisung in Funktion )
' an ihr Board angepaßt werden. Widerstandstoleranzen sorgen in
' Einzelfällen manchmal dafür das die Werte etwas anders ausfallen
' Am besten dann den WS wert mit Print für jede Taste ausgeben lassen

Function Tastenabfrage() As Byte
Local Ws As Word

   Tastenabfrage = 0
   Ton = 600
   Ws = Getadc(7)
'   Print "ws= " ; Ws
   If Ws < 1010 Then
      Select Case Ws
         Case 400 To 455
            Tastenabfrage = 1
            Ton = 550
         Case 335 To 380
            Tastenabfrage = 2
            Ton = 500
         Case 250 To 305
            Tastenabfrage = 3
            Ton = 450
         Case 180 To 220
            Tastenabfrage = 4
            Ton = 400
```

```
         Case 100 To 130
            Tastenabfrage = 5
            Ton = 350
      End Select
      Sound Portd.7 , 400 , Ton                              'BEEP
   End If

End Function




Function Md23_version() As Byte
Local Firmware As Byte
Local I2cread As Byte

   I2cread = Md23_slaveid + 1

   I2cstart
   I2cwbyte Md23_slaveid
   I2cwbyte 13                                               'Leseregister festlegen
   I2cstop

   I2cstart
   I2cwbyte I2cread
   I2crbyte Firmware , Nack
   I2cstop

   Md23_version = Firmware
End Function



Function Md23_batteriespannung() As Single
Local Spannung As Byte
Local I2cread As Byte

   I2cread = Md23_slaveid + 1

   I2cstart
   I2cwbyte Md23_slaveid
   I2cwbyte 10                                               'Leseregister festlegen
   I2cstop

   I2cstart
   I2cwbyte I2cread
   I2crbyte Spannung , Nack
   I2cstop

   Md23_batteriespannung = Spannung
End Function



'Schreibt in Register ein Wert
Sub Md23_writeregister(byval Register As Byte , Byval Wert As Byte)
   I2cstart
   I2cwbyte Md23_slaveid
   I2cwbyte Register                                         'Leseregister festlegen
   I2cwbyte Wert                                             'Leseregister festlegen
   I2cstop
End Sub
```

md23drehzahl.bas

Dieses Testprogramm demonstriert wie man die Drehgeberimpulse ausliest und in umdrehungen umrechnet. Wenn die Radgröße bekannt ist, kann man diesen Wert natürlich sehr einfach in eine Wegstrecke umrechnen.

Die Tasten im Demo sind wie folgt belegt:

Taste 1	Motor 1 stoppen
Taste 2	Motor 1 kleinere Geschwindigkeit
Taste 3	Motor 1 volle Geschwindigkeit Mode 0
Taste 4	Ausgabe der Impulsanzahl über RS232 (Drehgeber) und Umdrehungen
Taste 5	Zähler für Impulse auf 0 setzen

```
'############################################################
'md23drehzahl.bas
'
'Das Programm demonstriert wie das Motorboard MD23
'über I2C von RN-Control angesteuert wird
'
'Die Tasten haben in dem Demo folgende Funktion:
'Taste 1: Motor 1 stoppen
'Taste 2: Motor 1 kleinere Geschwindigkeit
'Taste 3: Motor 1 volle Geschwindigkeit Mode 0
'Taste 4: Ausgabe der Impulsanzahl über RS232 (Drehgeber) und Umdrehungen
'Taste 5: Zähler für Impulse auf 0 setzen
'
'Autor: Frank (roboternetz.de)
'Verwendet wurden: RN-Control & MD23 (robotikhardware.de)
'Weitere Beispiele sind im Roboternetz gerne willkommen!
'############################################################

$programmer = 12                        'MCS USB  (Zeile weglassen wenn anderer Programmer)


'  -------------- RN-Control übliche _____________________
Declare Function Tastenabfrage() As Byte

$regfile = "m32def.dat"
$framesize = 32
$swstack = 32
$hwstack = 64

$crystal = 16000000                                         'Quarzfrequenz
$baud = 9600
Config Scl = Portc.0                                        'Ports fuer IIC-Bus
Config Sda = Portc.1

Config Adc = Single , Prescaler = Auto                      'Für Tastenabfrage und
Spannungsmessung
Config Pina.7 = Input                                       'Für Tastenabfrage
Porta.7 = 1                                                 'Pullup Widerstand ein
Dim Taste As Byte
Dim Ton As Integer
```

```
I2cinit
Start Adc
Sound Portd.7 , 400 , 450                                   'BEEP
Sound Portd.7 , 400 , 250                                   'BEEP
Sound Portd.7 , 400 , 450                                   'BEEP
Print
Print "**** RN-CONTROL  V1.4 *****"
Print "Demoprogramm zum Motorboard MD23"
Print "RN-Control als auch MD23 gibts bei robotikhardware.de"
Print
'  -------------- Ende RN-Control übliche _____________________



Declare Sub Md23_writeregister(byval Register As Byte , Byval Wert As Byte)
Declare Function Md23_impulse(byval Motor As Byte) As Long
Const Md23_slaveid = &HB0
Dim Umdrehungen As Long
Dim Zaehler As Long



Wait 1
Md23_writeregister 16 , &H32                                'Automatische Timout Abschaltung
deaktivieren
Md23_writeregister 15 , &H0                                 'Mode 0
Print "Automatische Timout Abschaltung deaktiviert"



Do
   Taste = Tastenabfrage()
   If Taste <> 0 Then

      Select Case Taste

         Case 1:
                Md23_writeregister 0 , 128


         Case 2
                Md23_writeregister 0 , 170

         Case 3
                Md23_writeregister 0 , 255

         Case 4
                Zaehler = Md23_impulse(1)
                Umdrehungen = Zaehler / 360
                Print "Impulse: " ; Zaehler ; "  Umdrehungen: " ; Umdrehungen

         Case 5
                Md23_writeregister 16 , &H20                'Zähler auf 0 setzen

      End Select
      Sound Portd.7 , 400 , 500                             'BEEP
   End If

   Waitms 100
Loop

End


' Diese Unterfunktion fragt die Tastatur am analogen Port ab
' Sollte beim betätigen einer Taste kein Quittungston kommen, dann
' muss die die Tastenabfrage (Select Case Anweisung in Funktion )
' an ihr Board angepaßt werden. Widerstandstoleranzen sorgen in
' Einzelfällen manchmal dafür das die Werte etwas anders ausfallen
' Am besten dann den WS wert mit Print für jede Taste ausgeben lassen

Function Tastenabfrage() As Byte
```

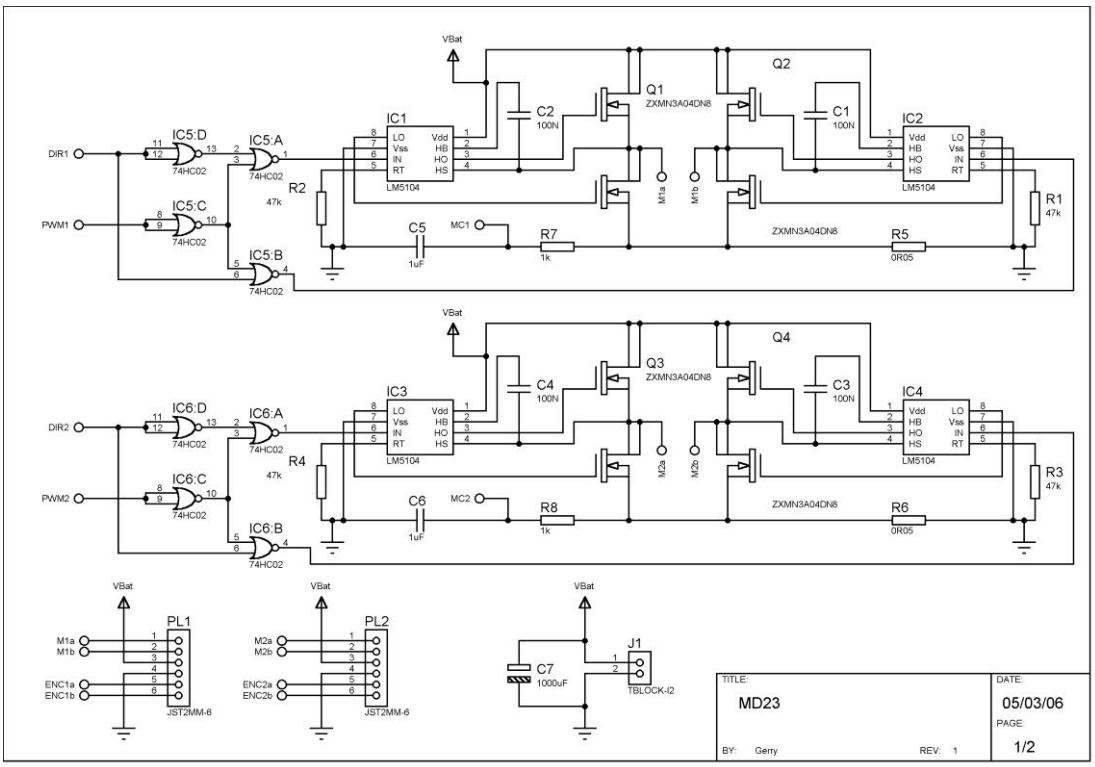
```
Local Ws As Word

   Tastenabfrage = 0
   Ton = 600
   Ws = Getadc(7)
'   Print "ws= " ; Ws
   If Ws < 1010 Then
      Select Case Ws
         Case 400 To 455
            Tastenabfrage = 1
            Ton = 550
         Case 335 To 380
            Tastenabfrage = 2
            Ton = 500
         Case 250 To 305
            Tastenabfrage = 3
            Ton = 450
         Case 180 To 220
            Tastenabfrage = 4
            Ton = 400
         Case 100 To 130
            Tastenabfrage = 5
            Ton = 350
      End Select
      Sound Portd.7 , 400 , Ton                              'BEEP
   End If

End Function




'Schreibt in Register ein Wert
Sub Md23_writeregister(byval Register As Byte , Byval Wert As Byte)
   I2cstart
   I2cwbyte Md23_slaveid
   I2cwbyte Register                                         'Leseregister festlegen
   I2cwbyte Wert                                             'Leseregister festlegen
   I2cstop
End Sub



Function Md23_impulse(byval Motor As Byte) As Long
Local W1 As Byte
Local W2 As Byte
Local W3 As Byte
Local W4 As Byte
Local Impulse As Long
Local I2cread As Byte


  I2cread = Md23_slaveid + 1

  If Motor = 1 Then
    I2cstart
    I2cwbyte Md23_slaveid
    I2cwbyte 2                                              'Leseregister festlegen
    I2cstop
  Else
    I2cstart
    I2cwbyte Md23_slaveid
    I2cwbyte 6                                              'Leseregister festlegen
    I2cstop
  End If

   I2cstart
   I2cwbyte I2cread
   I2crbyte W1 , Ack
   I2crbyte W2 , Ack
```

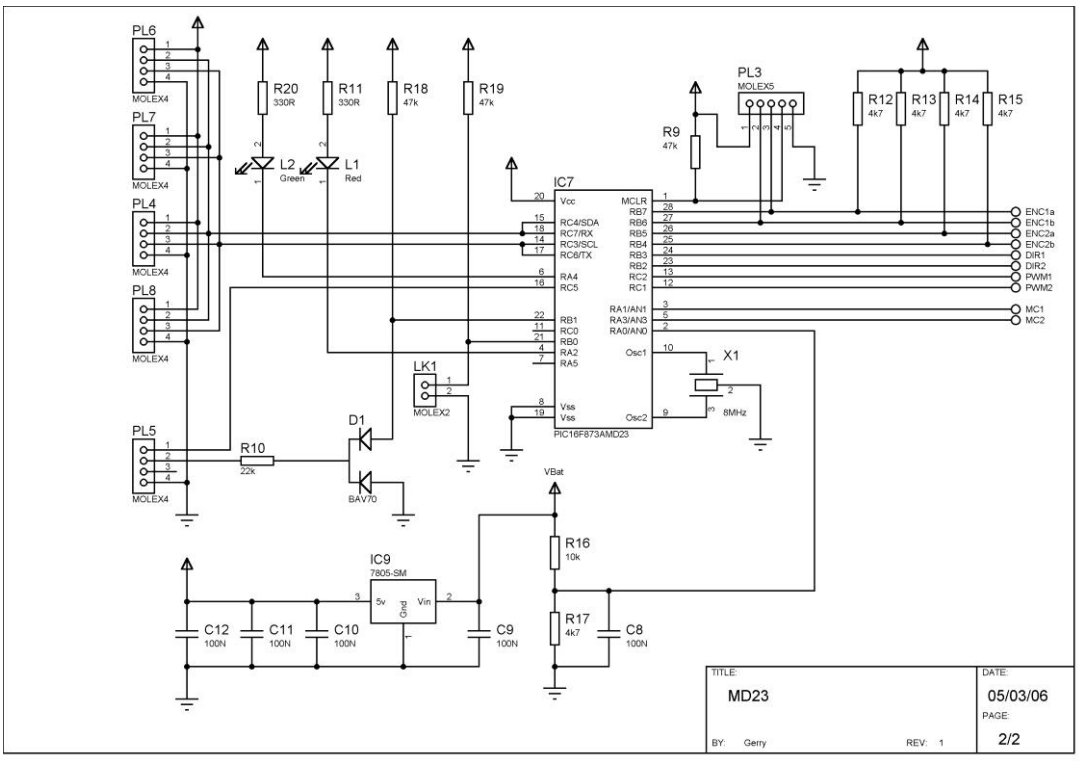
```
    I2crbyte W3 , Ack
    I2crbyte W4 , Nack
    I2cstop

    Impulse = 0
    Impulse = Impulse Or W1
    Shift Impulse , Left , 8
    Impulse = Impulse Or W2
    Shift Impulse , Left , 8
    Impulse = Impulse Or W3
    Shift Impulse , Left , 8
    Impulse = Impulse Or W4


    Md23_impulse = Impulse
End Function
```

Weitere Beispielprogramme auf der CD von Robotikhardware.de.

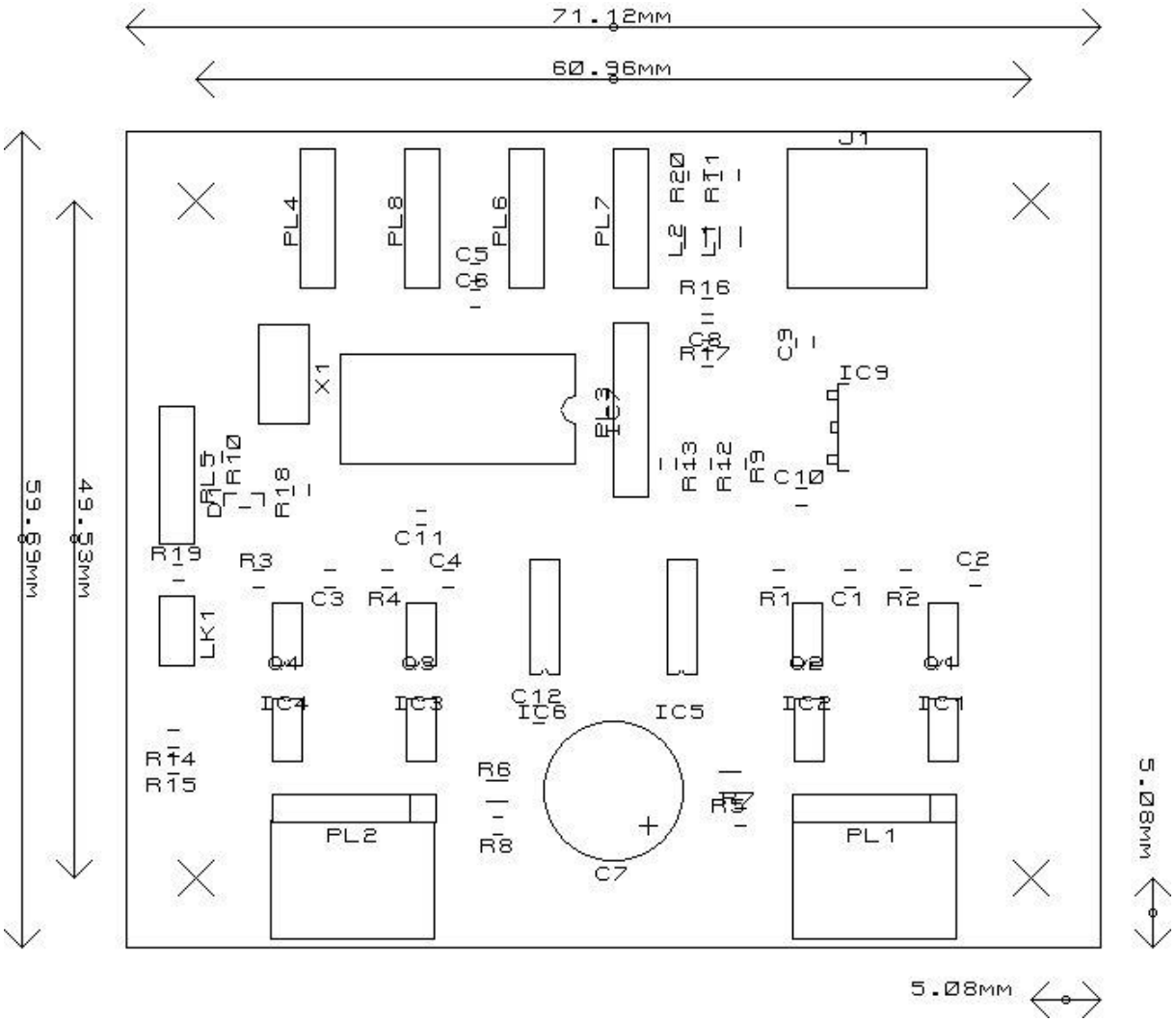
Die CD liegt jeder Boardbestellung bei!

Schaltplan


VBat
Q1
ZXMN3A04DN8
Q2
C2
100N
C1
100N
IC1
IC2
LM5104
IC5:D
IC5:A
IC5:C
IC5:B
74HC02
DIR1
PWM1
R2
47k
R1
47k
C5
1uF
MC1
R7
1k
R5
0R05
M1a
M1b
Q3
Q4
C4
C3
IC3
IC4
IC6:D
IC6:A
IC6:C
IC6:B
DIR2
PWM2
R4
R3
C6
MC2
R8
R6
M2a
M2b
PL1
PL2
J1
C7
1000uF
TBLOCK-I2
JST2MM-6
ENC1a
ENC1b
ENC2a
ENC2b
TITLE:
MD23
DATE:
05/03/06
PAGE:
1/2
BY:
Gerry
REV:
1


PL6
PL7
PL4
PL8
PL5
MOLEX4
R20
330R
R11
330R
R18
47k
R19
47k
L2
Green
L1
Red
PL3
MOLEX5
R9
47k
R12
4k7
R13
4k7
R14
4k7
R15
4k7
IC7
PIC16F873AMD23
LK1
MOLEX2
X1
8MHz
D1
BAV70
R10
22k
R16
10k
R17
4k7
C8
100N
IC9
7805-SM
C12
100N
C11
100N
C10
100N
C9
100N
ENC1a
ENC1b
ENC2a
ENC2b
DIR1
DIR2
PWM1
PWM2
MC1
MC2
TITLE:
MD23
DATE:
05/03/06
PAGE:
2/2
BY:
Gerry
REV:
1

Boardmaße


71.12MM
60.96MM
59.69MM
49.53MM
5.08MM
5.08MM
J1
PL4
PL8
PL6
PL7
PL3
PL5
PL2
PL1
X1
LK1
IC9
IC4
IC3
IC6
IC5
IC2
IC1
C7
C10
C11
C12
R16
R19
R18
R14
R15
R8
R6

MD23 Hersteller:

Devantech Ltd, England

Bezugsquelle Germany:

www.Robotikhardware.de
Bei uns mit CD und Bascom Beispielprogrammen

Aktuellen Link zu einer englischen Original-Dokumentation findet man im Shop von www.robotikhardware.de bei der jeweiligen Artikelbeschreibung!

Hinweise zur beschränkten Haftung

Das Modul ist nicht für Geräte geeignet die direkt oder indirekt medizinischen, gesundheitlichen oder anderen Zwecken, bei denen ein Ausfall / Fehler zu Schäden an Personden oder Sachwerten führen würde. Soll das Modul für einen solchen Fall eingesetzt werden, so ist der Kunde für den test und alle Zulassungen selbst verantwortlich. Robotikhardware.de übernimmt darfür keinerlei Haftung. Die Haftung beschränkt sich in allen Fällen auf den Austausch des fehlerhaften Moduls. Module die durch fehlerhaften Anschluß / Bedienung beschädigt wurden, können nicht ersetzt werden.