

# Morse Sender

Author DK1RI, Version V01.5, 20170815

This project can be found in <https://www.github.com/dk1ri> also.

## Einleitung

Dieses Interface ist eine Vorlage für ein Interface, das ein Gerät, das sich mit Morse Signalen steuern lässt, in ein MYC System einbindet.

Das Interface arbeitet als Slave am I2C Bus oder kann über RS232 /USB gesteuert werden. Defaultmäßig sind alle Schnittstellen aktiv. Mit dem Initialisierungsbefehl können diese aber deaktiviert werden. Der Initialisierungsbefehl funktioniert aber immer.

## Beschreibung

Es wird die gleiche Leiterplatte wie für das I2C\_RS232\_Interface verwendet.

Die Eagle Daten stehen unter [1].

Die Stromversorgung ist 7- 15V, Stromaufnahme ca. 20mA max oder über USB.

Das Programm ist nur das Framework, weitere MYC Befehle des Gerätes müssen nach Bedarf eingefügt werden. Man kann aber Strings direkt entsprechend den MYC Regeln eingeben, die dann in Morsesignale umgesetzt werden.

Die Tonhöhe des Morsesignals ist von 100Hz bis 2kHz einstellbar, die Geschwindigkeit von 5 bis 100 Wpm. Diese Werte sind nicht sehr genau.

Es gibt drei Betriebsarten; die Betriebsart bleibt nach dem Ausschalten erhalten:

### *MYC\_mode, default (&H0600)*

Die Befehlsdaten müssen binär eingegeben werden; es erfolgt keine Wandlung, hier als Beispiel im HEX Format:

0104474657374

01 ist der Befehl zur Morse Ausgabe. 04 ist die Länge des nachfolgenden Strings test

Die gesendeten Zeichen werden auch immer über die serielle Schnittstelle so ausgegeben.

Es können 252 Zeichen maximal am Stück übertragen werden; die Länge des Strings wird mit einem Byte übertragen.

Nach Eingabe eines Strings muss man warten, bis die Morse Signale übertragen wurden.

Andernfalls gehen Eingabedaten verloren.

Serieller Eingang und I2C können „gleichzeitig“ verwendet werden. Allerdings wird beim jedem Wechsel der Befehlspeicher gelöscht.

### *Direkt\_mode, nicht MYC Mode (&H0601):*

Alle erlaubten Morsezeichen, die an der **seriellen Schnittstelle** in ASCII eingegeben werden, werden in Morsezeichen umgesetzt, sobald ein CR (&H0D) empfangen wurde.

An der seriellen Schnittstelle kann dieser Mode mit &H0600 oder &H0603 ... 08 verlassen werden. Andere Befehle werden ignoriert.

### *Verschiedene 5 er Gruppen (&H0602 bis &h0608):*

Es gibt die Möglichkeit, 5er Gruppen mit verschiedenen Zeichensätzen auszugeben

Die Eingabe von Befehlen ist eingeschränkt möglich: Die Befehlsauswertung erfolgt nach jedem Morsezeichen. Auch hier ist zu beachten, dass während des Sendens eines Morsezeichens nur 1 Byte

gepuffert werden kann. Ein Wechsel des Modes zum Beispiel &H0600 funktioniert.

## Befehle

Zu Details zum MYC Protokoll und zur Bedienung siehe [3] und [4] (aktuell).

Folgende Befehle werden von der I2C / RS232 / USB Schnittstelle akzeptiert; dies ist eine Kopie aus dem Bascom Programm:

Announce0:

'Befehl &H00

'eigenes basic announcement lesen

'basic announcement is read to I2C or output

Data "0;m;DK1RI;morse sender;V03.0;1;145;1;15"

,

Announce1:

'Befehl &H01 <s>

'I2C / serial String als Morse senden

'send I2C / serial String as Morse

Data "1;oa,send morse;252,{.,\,,:,\,;?,-,\_,(,)',=,+/,@,0 to 10,a to z, A to Z}"

,

Announce2:

'Befehl &H02 0 to 19

'Geschwindigkeit schreiben

'write speed

Data "2;op,morse speed;20,{5 to 100};lin;Wpm"

,

Announce3:

'Befehl &H03

'Geschwindigkeit lesen

'read speed

Data "3;ap,as2"

,

Announce4:

'Befehl &H04

'Frequenz schreiben 100 - 2000Hz

'write frequency

Data "4;op,morse frequency;20,{100 to 2000};lin;Hz"

,

Announce5:

'Befehl &H05

'Frequenz lesen

'read frequency

Data "5;ap,as4"

,

Announce6:

'Befehl &H06

'Mode einstellen, Myc, direkteingabe, 5er Gruppen

'set mode

Data "6;os,mode;0,myc mode;1,morse input;2,0 to 9;3,a to f;4,g to l;5,m to s;6,t to z;7,special;8,all"

,

Announce7:

'Befehl &H07

'Morse mode lesen

'read morse mode

Data "7;as,as6"

'

Announce8:

'Befehl &HF0<n><m>

'liest announcements

'read n announcement lines

Data "240;an,ANNOUNCEMENTS;145;15"

'

Announce9:

'Befehl &HFC

'Liest letzten Fehler

'read last error

Data "252;aa,LAST ERROR;20,last\_error"

'

Announce10:

'Befehl &HFD

'Geraet aktiv Antwort

'Life signal

Data "253;aa,MYC INFO;b,ACTIVE"

'

Announce11:

'Befehl &HFE :

'eigene Individualisierung schreiben

'write individualization

Data "254;ka,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,18,  
{0 to 127};a,RS232,1;a,USB,1"

'

Announce12:

'Befehl &HFF :

'eigene Individualisierung lesen

'read individualization

Data "255;aa,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,18,  
{0 to 127};a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER\_OF\_BITS,8n1;a,USB,1"

Announce13:

Data "R !\* IF \$6&1"

'

Announce14:

Data "R &6 IF \$6&1"

'

## Regeln

Es gibt zwei Zeilen mit Regeln für den Master und eine für den Slave.

Während des Morse\_mode = 1 kann nur der Morse\_mode geändert werden.

## **I2C**

Die Default Adresse ist 18 (&H12).

Mit dem Befehl &HFE03<n> kann die Adresse in n (1 ... 127) geändert werden.

Pullup Widerstände für den I2C Bus (R8/R9) können bei Bedarf bestückt werden. Der Gesamtwiderstand am Bus sollte zwischen 1 und 10 kOhm liegen.

Wenn Geräte am I2C Bus nur 3.3V vertragen (zB der Raspberry), dürfen die Pullup Widerstände nicht bestückt werden.

## **Fehlermeldungen**

Der Befehl &HFC liefert den letzten Fehler im Format:

aktuelle Befehlsnummer - Fehler - Befehlsnummer beim Auftritt des Fehlers

Dazu werden die empfangenen Befehle von 0 bis 255 umlaufend gezählt.

## **Reset**

Ist der Reset Jumper JP4 beim Anlegen der Versorgungsspannung überbrückt, werden wieder die Defaultwerte eingelesen. Dies ist hilfreich, wenn die aktuelle I2C Adresse verloren gegangen ist. Der Reset hat eine Verzögerungszeit von 1 Sekunde.

## **Watchdog**

Die Befehlseingabe muss in weniger als 1 Sekunde beendet sein. Danach werden die bereits empfangenen Daten gelöscht. Dies soll falsche Eingaben vermeiden. Mit dem "letzten Fehler" Befehl kann man Eingabefehler sehen. Er zeigt die aktuelle Befehlsnummer und die des Fehlers. Nach jedem beendeten Befehl wird der gesamte Befehlpuffer gelöscht. Weiterhin müssen zu sendende I2C Daten innerhalb einer Sekunde abholt werden. Es gibt einen allgemeinen Reset, wenn die Abfrageschleife länger als 2 Sekunde dauert.

## **Software**

Die Steuerung übernimmt ein AVR Mikrocontroller Atmega168 oder größer.

Die Software wurde in BASCOM geschrieben [2]

## **Programmierung des Prozessors**

Zur Programmierung des Prozessors ist ein 6poliger ISP Stecker vorhanden.

Um der Prozessor von der Stromversorgung der übrigen Schaltung zu trennen, muss der Jumper JP1 entfernt werden.

Die Fuses müssen möglicherweise programmiert werden (sh Bascom Programm) !! Prozessortyp und Frequenz müssen ggf angepasst werden.

## **RS232 Schnittstelle**

Schnittstellenparameter: 19k2 8N1

Es muss bei Jumper JP7 und JP8 ist jeweils Pin1 und Pin2 (die jeweils äußeren pins) überbrückt werden.

I2C und RS232 / USB können zur Befehlseingabe nicht gleichzeitig verwendet. Der Befehlpuffer

wird gelöscht, wenn die Schnittstelle gewechselt wird.

## **USB Schnittstelle**

Das Interface kann alternativ mit der USB Platine UM2102 von ELV bestückt werden. Die USB Platine wird plan auf der Oberseite des Morse Senders verlötet: der USB Stecker zeigt seitlich nach außen. Die mittleren 4 pins des Verbinders ST2 sind mit dem 4 poligen Verbinder JP9 auf dem Interface zu verbinden. USB Platine und Morse Sender müssen voneinander isoliert werden. Die Stromversorgung erfolgt dann über USB.

## **SMD**

Die Leiterplatte ist teilweise mit SMD bestückt. Bei den nötigen Bauteilen sind das aber nur relativ großen Kondensatoren (1206).

## **Morse Ausgang**

Das Morsesignal kann direkt herausgeführt werden ( $R7 = 0 \text{ Ohm}$ , C12 entfällt) oder über einen Tiefpass ( $1k/100nF$ ). Das Ausgangssignal ist ein Rechtecksignal und auch nach der Filterung nicht gleichspannungsfrei. Der Pegel im Ruhezustand ist unbestimmt. Ein Koppelkondensator und eine Anpassung des Pegels an den Empfänger wird daher empfohlen.

## **Stromversorgung**

Die Stromversorgung ist 7- 15V, Stromaufnahme ca. 20mA max. Bei Verwendung des USB Moduls erfolgt die Stromversorgung darüber.

## **Bestückung der Leiterplatte**

Da die Leiterplatte auch für andere Anwendungen eingesetzt werden kann, brauchen nur folgende Bauteile bestückt werden:

IC1, Q1, C3 – C6, JP1 (muss für Normalbetrieb überbrückt werden), ohne Tiefpass:  $R7=0$   
SL4

Verwendung von ISP:  
JP6

RS232 Schnittstelle:

IC2, IC3, D1, C1, C2, C7 – C10, JP7, JP8 (jeweils Pin1 und Pin 2 überbrücken), X1, X4 (Buchse)

USB (alternativ zu RS232)  
UM2102

I2C

R8, R9 nach Bedarf, X2, X3, (ohne USB außerdem: X1, IC3, D1, C1, C2

Weitere Bestückung erleichtert Tests..

## **Anschlüsse**

Power  
Tip 12V  
Ring GND

RS232 (Buchse)  
5 GND  
2 Jumper  
3 Jumper

I2C (2 x 3,5mm Klinke, Stereo)  
Sleeve GND  
Ring SDA  
Tip SCL

Morsesignal (LP SL6)  
1 Morsesignal  
5 GND

## Versionen

Diese Beschreibung gilt für die  
Leiterplattenversion 02.0  
Bascom Version 03.0

## Copyright

Die Ideen in diesem Dokument unterliegen der GPL (Gnu Public Licence V2) soweit keine früheren, anderen Rechte betroffen sind.

Die Verwendung der Unterlagen erfolgt auf eigene Geafahr; es wird keinerlei Garantie übernommen.

The ideas of this document can be used under GPL (Gnu Public License V2) as long as no earlier other rights are affected.

The usage of this document is on own risk, there is no warranty.

## Referenzen

- [1] [dk1ri.de/dhw/i2c\\_rs232\\_interface\\_eagle.zip](http://dk1ri.de/dhw/i2c_rs232_interface_eagle.zip)
- [2] [dk1ri.de/dhw/morse\\_sender\\_bascom.zip](http://dk1ri.de/dhw/morse_sender_bascom.zip)
- [3] [dk1ri.de/myc/MYC.pdf](http://dk1ri.de/myc/MYC.pdf)
- [4] [dk1ri.de/myc/Description.pdf](http://dk1ri.de/myc/Description.pdf) (englisch)
- [5] [dk1ri.de/myc/Definitions.pdf](http://dk1ri.de/myc/Definitions.pdf) (englisch)