

## DTMF Sender

Author DK1RI, Version V03.1, 20180116

This paper is published in <https://github.com/dk1ri> as well.

### Einleitung

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

Das Interface arbeitet als Slave am I2C Bus.

Das Interface kann auch bei entsprechender Bestückung der Leiterplatte Eingaben an der RS232 oder USB Schnittstelle in DTMF Signale umsetzen.

Im „no\_myc“ Mode arbeitet das Interface als einfacher ASCII - DTMF Wandler.

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. 40mA max oder über USB.

Das Programm ist nur das Framework, weitere MYC Befehle des Gerätes müssen nach Bedarf eingefügt werden. Man kann aber DTMF Codes direkt entsprechend den MYC Regeln eingeben.

Die Daten müssen binär eingegeben werden; es erfolgt keine Wandlung, hier im HEX Format:  
010448495051

01 ist der Befehl zur DTMF Ausgabe. 04 ist die Länge des nachfolgenden String. Die folgenden Zeichen müssen als ASCII Zeichen 0-9, \*,#, A-D (oder entsprechende HEX Codes) eingegeben werden.

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 DTMF Signale übertragen wurden.

Andernfalls gehen Eingabedaten verloren.

RS232 /USB und I2C können „gleichzeitig“ verwendet werden. Allerdings wird beim jedem Wechsel der Befehlpuffer gelöscht.

Das Gerät kann auch als ASCII - DTMF Wandler verwendet werden (no\_MYC Mode). Dies funktioniert nur mit der seriellen Schnittstelle. Die Zeichen müssen einzeln eingegeben werden. Jedes Zeichen wird sofort gesendet. Ein weiteres Zeichen wird noch zwischengespeichert, weitere aber nicht mehr sicher.

Mit der Eingabe des Leerzeichens kommt man in den normalen MYC Befehlsmode.

### 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  
'basic annouement wird gelesen  
'basic announcement is read  
Data "0;m;DK1RI;DTMF_sender;V04.0;1;145;1;13;1-1"  
,
```

```
Announce1:  
'Befehl &H01  
'liest string von I2C oder serial und sendet als DTMF  
'read string from I2C or serial and send as DTMF  
Data "1;oa,send dtmf;250,{0 to 9,*,#,A to D}"  
,
```

```
Announce2:  
'Befehl &HEA  
'DTMF Länge schreiben  
'write DTMF length  
Data "234;ka,DTMF Duration;b"  
,
```

```
Announce3:  
'Befehl &HEB  
'Dtmf duration lesen  
'read DTMF Länge  
Data "235;la,as234"  
,
```

```
Announce4:  
'Befehl &HEC  
'Dtmf Pause schreiben  
'write DTMF Pause  
Data "236;ka,DTMF Pause;b"  
,
```

```
Announce5:  
'Befehl &HED  
'Dtmf Pause lesen  
'read DTMF Pause  
Data "237;la,as236"  
,
```

```
Announce6:  
'Befehl &HEE  
'nomyc schreiben  
'write nomyc  
Data "238;ka,no_myc;a"  
,
```

```
Announce7:  
'Befehl &HEF  
'nomyc lesen
```

```

'read nomyc
Data "239;la,as238"
'
Announce8:
'Befehl &HF0<n><m>
'announcement aller Befehle lesen
'read m announcement lines
Data "240;ln,ANNOUNCEMENTS;145;13"
'
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 <n> <n>
'Individualisierung schreiben
'write individualization
Data "254;ka,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,7,{0
to 127};a,RS232,1;a,USB,1"
'
Announce12:
'Befehl &HFF
'Individualisierung lesen
'read individualization
Data "255;la,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,7,{0
to 127};a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER_OF_BITS,8n1;a,USB,1"
'

```

## I2C

Die Default Adresse ist 7.

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), muss dieses Interface auch mit 3.3V versorgt werden oder die Pullup Widerstände dürfen 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.

Nach 254 korrekten Befehlen wird der Fehlereintrag gelöscht.

## **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.

## **Watchdog**

Die Befehlseingabe und Ausführung muss in weniger als 1 Sekunde beendet sein. Danach werden die bereits empfangenen Daten gelöscht. Dies soll falsche Eingaben vermeiden. Mit dem &HFC "letzten Fehler" Befehl kann man Eingabefehler sehen.

Bei einem Lesebefehl müssen die Daten innerhalb von 10 Sekunden vom I2C Master abgeholt werden – wenn die I2C Schnittstelle gerade verwendet wird. Danach werden die Daten gelöscht. Diese Zeit kann mit dem Wert Tx\_factor im Bascom Programm geändert werden. Neue Befehle können erst eingegeben werden, wenn alle Daten abgeholt wurden. Wird die RS232 / USB Schnittstelle verwendet, werden die Daten sofort ausgegeben.

Es gibt einen kompletten Reset, wenn die Hauptschleife länger als 2 Sekunde dauert, zum Beispiel, wenn die I2C Schnittstelle nicht korrekt arbeitet.

## **Software**

Die Steuerung übernimmt ein AVR Mikrocontroller Atmega8  
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.

## **DTMF Ausgang**

Der DTMF Ausgang des Prozessors kann direkt herausgeführt werden (R7 =0 Ohm, C12 entfällt) oder über einen Tiefpass (330Ohm/330nF). 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. Mit Tiefpass ergibt sich ein recht brauchbares Signal, allerdings ist der Pegel für den DTMF Empfänger noch zu hoch.

## **RS232 Schnittstelle**

Bei Bedarf und entsprechender Bestückung kann auch die RS232/USB Schnittstelle zur Befehlseingabe verwendet werden. Die Erkennung der aktiven Schnittstelle (I2C / seriell) erfolgt automatisch. Sofern ein Befehl nicht komplett gesendet wurde und die andere Schnittstelle sendet Daten, wird der alte Befehl abgebrochen. Daher kann eine gleichzeitige Verwendung beider Schnittstellen zu Fehlern führen.

Schnittstellenparameter: 19k2 8N1

Es muss bei Jumper JP7 und JP8 ist jeweils Pin1 und Pin2 überbrückt werden.

## **USB Schnittstelle**

Das Interface kann alternativ mit der USB Platine UM2102 von ELV bestückt werden. Die USB Platine wird plan auf der Oberseite der Interfaces verlötet: der USB Stecker zeigt seitlich nach außen. Die Pins des Verbinders ST2 sind mit dem 4 poligen Verbinder JP9 auf dem Interface zu verbinden. USB Platine und Interface 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 zwei relativ großen Kondensatoren (1206).

## **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(10MHz !!!), C3 – C6, JP1 (muss für Normalbetrieb überbrückt werden), SL4, ohne Tiefpass:R7=0

Tiefpass:

R7, C12

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

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

DTMF (LP SL6)  
1 DTMF  
5 GND

## Versionen

Diese Beschreibung gilt für die  
Leiterplattenversion V02.1  
Bascom Version V04.1

## 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/dtmf\\_tx.zip](http://dk1ri.de/dhw/dtmf_tx.zip)
- [3] [dk1ri.de/myc/MYC.pdf](http://dk1ri.de/myc/MYC.pdf)
- [4] <https://dk1ri.de/myc/Description.txt> oder <https://dk1ri.de/myc/Description.pdf> (englisch)
- [5] <https://dk1ri.de/myc/Definitions.txt> oder <https://dk1ri.de/myc/Definitions.pdf> (englisch)