

# DTMF Sender für Remote Shack von MFJ (TM)

Author: DK1RI

Version V01.5 20180116

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

## Einleitung

Dieses Interface setzt MYC Befehle in DTMF Befehle für das MFJ Remote Shack (TM) um. So kann ein Remote Shack in ein MYC System eingebunden werden.

Dies ist ein Beispiel für die Verwendung des DTMF Senders als Template durch einfache Ergänzungen.

Das Interface arbeitet als Slave am I2C Bus.

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

Defaultmäßig sind alle Schnittstellen aktiv. Mit dem Initialisierungsbefehl können diese aber deaktiviert werden. Der Initialisierungsbefehl funktioniert aber immer.

## Beschreibung

Die Eagle Daten der Leiterplatte stehen unter [1].

Die Stromversorgung ist 7- 15V, Stromaufnahme ca. 40mA max oder über USB.

Die Daten müssen binär eingegeben werden; es erfolgt keine Wandlung, RS232 / USB und I2C können „gleichzeitig“ verwendet werden. Allerdings wird beim jedem Wechsel der Befehlspeicher gelöscht.

Für das Programm stand die Befehlsliste, die mit dem RBC212-IC2 geliefert wurde, zur Verfügung. Ich konnte nicht die Bedeutung aller Befehle herausfinden; einige sind offenbar für andere Zusatzgeräte gedacht. Die Befehle wurden zwar mit einem DTMF Empfänger getestet, aber nicht mit einem realen RBC.

Das Remote Shack liefert keine Datenrückmeldung. Bei Eingabe von Daten, die das Funkgerät nicht versteht, sind die Anzeigen an der Benutzerschnittstelle möglicherweise falsch.

Die Initialisierung dauert ca 10 Sekunden, da der RBC erst eingeschaltet wird und dann die Ansage abgewartet wird.

## Befehle / Announcements

Folgende Befehle werden von der I2C / RS232 / USB Schnittstelle akzeptiert:

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

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;MFJ RBC Interface(TM);V03.0;1;145;1;61;1-1"
```

```
,
```

Announce1:  
'Befehl &H01  
'setzt FrequenzVFO A, command \* VFOA \*  
'set frequency  
Data "1;op,set frequency;1;10000000,{0 to 999999900};lin;Hz"  
,'

Announce2:  
'Befehl &H02  
'setzt Frequenz VFO B, command \* VFOB #  
'set frequency  
Data "2;op,set frequency;1;10000000,{0 to 999999900};lin;Hz"  
,'

Announce3:  
'Befehl &H03  
'gibt frequenz ausVFO A (als Sprache), command \*\*  
'play frequency VFOA  
Data "3;ou,frequency VFO A;1;0;idle;1 play"  
,'

Announce4:  
'Befehl &H04  
'gibt frequenz aus (als Sprache), command ##  
'play frequency VFOB  
Data "4;ou,frequency VFOB;1;0;idle;1 play"  
,'

Announce5:  
'Befehl &H05 0 to 3  
'Aendert Frequenz um 1 Step; command #1 1 4 8 0  
'change frequency 1 step  
Data "5;os,change frequency;1;0,+100;1,-100;2,+500;3,-500"  
,'

Announce6:  
'Befehl &H06 0 to 4  
'startet scan, command #1 2 3 5 6 0  
'start scan  
Data "6;os,scan;0,medium up;1;1,fast up;2,medium down;3,fast down;4,stop"  
,'

Announce7:  
'Befehl &H07 0 to 99  
'setzt memory, command #1 7 x x  
'set memory  
Data "7;op,set memory;1;100,{0 to 99};lin;-"  
,'

Announce8:  
'Befehl &H08 0 to 99  
'recall memory, command #1 9 x x  
'recall memory  
Data "8;op,recall memory frequency;1;100,{0 to 99};lin;-"  
,'

Announce9:  
'Befehl &H09

```

'Ant 1 ein, command #2 1
'Ant 1 on
Data"9;ou,Ant1;1;0,idle;1,Ant1"
,
Announce10:
'Befehl &H0A
'Ant 2 ein command #2 2
'Ant 2 on
Data "10;ou,Ant2;1.0,idle;1,Ant2"
,
Announce11:
'Befehl &H0B
'Tuner an, command #2 3
'Tuner on
Data"11;ou,Tuner on;1;0,idle;1,tuner on"
,
Announce12:
'Befehl &H0C
'Tuner aus, command #2 6
'Tuner off
Data"12;ou,Tuner off;1;0,idle;1,tuner off"
,
Announce13:
'Befehl &H0D
'Aux1 an, command #2 7
'Aux1 on
Data"13;ou,Aux1 on;1;0,idle;1,aux1 on"
,
Announce14:
'Befehl &H0E
'Aux1 aus, command #2 8
'Aux1 off
Data"13;ou,Aux1 off;1;0,idle;1,aux1 off"
,
Announce15:
'Befehl &H0F
'Tune Antenne command #2 9
'Tune Antenna
Data"15;ou,Tune Antenna on;1;0,idle;1,tune antenna on"
,
Announce16:
'Befehl &H10
'0 to 359 dreht Antenne1, command #2 4 x x x
' Rotate Ant1
Data "16;op,rotate Antenna 1;1;360,{0 to 359}"
,
Announce17:
'Befehl &H11
'0 to 359 dreht Antenne2, command #2 5 x x x
'Rotate Ant2

```

```

Data "17;op,rotate Antenna 2;1;360,{0 to 359}"
,
Announce18:
'Befehl &H12
'stoppt Rotor, command #2 0
'stops rotation
Data"18;ou,stop rotation;1;0,idle;1,stop antenna"
,
Announce19:
'Befehl &H13 0,1
'Abschwächer, command #3 1 or #3 0 1
'Attenuator
Data"19;ou,Attenuator;1;0.idle;1,attenuator"
,
Announce20:
'Befehl &H14 0,1
'Vorverstärker, command #3 2 Or #3 0 2
'Preamp
Data"20;ou,Preamp;1;0.idle;1,preamp"
,
Announce21:
'Befehl &H15 0,1
'Noiseblanker,command #3 3 Or #3 0 3
'Noiseblanker
Data"21;ou,Noise blanker;1;0.idle;1,noise blanke"
,
Announce22:
'Befehl &H16 0,1
'Rauschunterdrückung, command #3 4 Or #3 0 4
'Noise reduction
Data"22;ou,Noise reduction;1;0.idle;1,noise reduction"
,
Announce23:
'Befehl &H17 0,1
'Auto Notch #3 5 Or #3 0 5
'Auto notch
Data"23;ou,Auto Notch;1;0.idle;1,auto notxh"
,
Announce24:
'Befehl &H18 0 to 2
'setzt filter, command #3 7.. 8... 9
'set filter
Data"24;os,Filter;1;0,narrow;1,medium;2,wide"
,
Announce25:
'Befehl &H19
'alle Function aus, command #3 0 0
'all functions off
Data"25;ou,all filter functions off;1;0,idle;1,all filter off"
,

```

Announce26:  
'Befehl &H1A 0 to 4  
'setzt Betriebsart, command #3 6 1, 2, 3, 4, 5  
'set mode  
Data"26;os,Mode;1;0,LSB;1,USB;2,AM;3,CW;4,FM"  
,'

Announce27:  
'Befehl &H1B 0,1  
'Sprachkompressor, command #4 2 Or #4 0 2  
'speech compressor  
Data"27;ou,spech compressor;1;0.idle;1,speech compressor"  
,'

Announce28:  
'Befehl &H1C 0,1  
'VOX, command #4 3 Or #4 0 3  
'VOX  
Data"28;ou,VOX;1;0.idle;1,vox"  
,'

Announce29:  
'Befehl &H1D 0,1  
'Tone, command #4 4 Or #4 0 4  
'Tone  
Data"29;ou,Tone;1;0.idle;1,tone"  
,'

Announce30:  
'Befehl &H1E 0,1  
'split, command #4 5 Or #4 0 5  
'split  
Data"30;ou,split;1;0.idle;1,split"  
,'

Announce31:  
'Befehl &H1F 0,1  
'Vertärker #4 6 Or #4 0 6  
'Amplifier  
Data"31;ou,Amplifier;1;0.idle;1,amplifier"  
,'

Announce32:  
'Befehl &H20  
'alle Sendefunctionen aus, command #4 0 0  
'all tx functions off  
Data"32;ou,all TX functions off;1;0.idle;1,tx function off"  
,'

Announce33:  
'Befehl &H21 0 to 9999  
'setzt tone Frequenz, command #4 7 x x x x  
'set tone frequency  
Data "33;op,tone frequency;1;1000,{0 to 999,9};lin;Hz"  
,'

Announce34:  
'Befehl &H22 0 to 2

```

'setzt shift,command #4 8 x
'set shift
Data"34;os,Shift;1;0,simplex;1,+;2,-"
,
Announce35:
'Befehl &H23 0 to 3
'Ausgangsleistung, command #4 9 x
'set power
Data"35;os,power;1;0,25%;1,50%;2,75%;3,100%"
,
Announce36:
'Befehl &H24 0,1
'AUX2 an au, command #6 2 0 Or #6 2 1
'AUX2 on off
Data"36;ou,AUX2;1;0.idle;1,aux2"
,
Announce37:
'Befehl &H25 0,1 #
'AUX3 an au, command #6 3 0 Or #6 3 1
'AUX3 on off
Data"37;ou,AUX3;1;0.idle;1,aux3"
,
Announce38:
'Befehl &H26 0,1
'AUX4 an au, command #6 4 0 Or #6 4 1
'AUX4 on off
Data"38;ou,AUX4;1;0.idle;1,aux4"
,
Announce39:
'Befehl &H27 0,1
'AUX5 an au, command #6 5 0 Or #6 5 1
'AUX5 on off
Data"39;ou,AUX5;1;0.idle;1,aux5"
,
Announce40:
'Befehl &H28 0,1
'AUX6 an au, command #6 6 0 Or #6 6 1
'AUX6 on off
,
Announce41:
'Befehl &H29 0,1
'AUX7 an au, command #6 7 0 Or #6 7 1
'AUX7 on off
Data"41;ou,AUX7;1;0.idle;1,aux7"
,
Announce42:
'Befehl &H2A 0,1
'AUX8 an au, command #6 8 0 Or #6 8 1
'AUX8 on off
Data"42;ou,AUX8;1;0.idle;1,aux8"

```

```

,
Announce43:
'Befehl &H2B 0,1
'AUX9 an au, command #6 9 0 Or #6 9 1
'AUX9 on off
Data"43;ou,AUX9;1;0.idle;1,aux9"
,
Announce44:
'Befehl &H2C
'reset, command #5 5
'reset
Data"44;ou,reset;1;0.idle;1,reset"
,
Announce45:
'Befehl &H2D
'Sprachlautstärke auf, command #5 8
'voice volume up
Data"45;ou,voice volume up;1;0.idle;1,volume up"
,
Announce46:
'Befehl &H2E
'Sprachlautstärke ab, command #5 0
'voice volume down
Data"46;ou,voice volume down;1;0.idle;1,volume down"
,
Announce47:
'Befehl &H2F 0 to 9
'Zahl der Ruftone, command#5 7 x
'number of ring
Data"47;op,number of ring;1;10,{0 to 9}"
,
Announce48:
'Befehl &H30 0 to 9999
'passwort festlegen, command #5 4 x x x x
'set password
Data"48;om,set password;L,{0 to 9999}"
,
Announce49:
'Befehl &H31
'Sende ein, command #4 1
'transmit
Data"49;ou,transmit;1;0.idle;1,transmit"
,
Announce50:
'Befehl &H32
'Spielt Ch1 beim Senden, command #4 1
'play Ch1
Data"50;ou,play Ch1;1;0.idle;1,play ch 1"
,
Announce51:

```

```

'Befehl &H33
'Start, command *
' start
Data"51;ou,start;1;0.idle;1,start"
,

Announce52:
'Befehl &HEA
'DTMF Länge schreiben
'write DTMF length
Data "234;ka,DTMF Duration;b"
,

Announce53:
'Befehl &HEB
'Dtmf duration lesen
'read DTMF Länge
Data "235;la,as234"
,

Announce54:
'Befehl &HEC
'Dtmf Pause schreiben
'write DTMF Pause
Data "236;ka,DTMF Pause;b"
,

Announce55:
'Befehl &HED
'Dtmf Pause lesen
'read DTMF Pause
Data "237;la,as236"
,

Announce56:
'Befehl &HF0<n><m>
'liest announcements
'read m announcement lines
Data "240;ln,ANNOUNCEMENTS;145;61"
,

Announce57:
'Befehl &HFC
'Liest letzten Fehler
'read last error
Data "252;aa,LAST ERROR;20,last_error"
,

Announce58:
'Befehl &HFD
'Geraet aktiv Antwort
'Life signal
Data "253;aa,MYC INFO;b,ACTIVE"
,

Announce59:
'Befehl &HFE :
'eigene Individualisierung schreiben

```

'write individualization

Data "254;ka,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,14, {0 to 127};a,RS232,1;a,USB,1"

,

Announce60:

'Befehl &HFF :

'eigene Individualisierung lesen

'read individualization

Data "255;la,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,14, {0 to 127};a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER\_OF\_BITS,8n1;a,USB,1"

,

## I2C

Die Default Adresse ist 14.

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 Atmega168

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 V03.0

## 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/dtmf\\_rbc.zip](http://dk1ri.de/dhw/dtmf_rbc.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)