

# Infrarot Sender

Author DK1RI, Version V02.4, 20161111

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

## Einleitung

Dieses Interface ist eine Vorlage für ein Interface, das ein Gerät, das sich mit Infrarot 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 RC5 RC6 Signale umsetzen.

Im „no\_myc“ Mode arbeitet das Interface als einfacher ASCII – Infrarot (nur RC5) 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.

Das Interface arbeitet als I2C Slave.

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

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

&H01&H01

01 ist der Befehl zur Infrarot RC5 Ausgabe. 01 ist ein Infrarot Code. Es wird immer nur ein Zeichen übertragen.

Im nicht-MYC mode müssen die RC5 Codes als 2 Ziffern von 00 - 63 über die serielle Schnittstelle eingegeben werden. Rückkehr zum MYC mode erfolgt durch Eingabe von &H80. Die I2C Schnittstelle ist vom nicht-MYC mode nicht betroffen.

## 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;IR\_sender;V03.2;1;160;1;14"

,

Announce1:

'Befehl &H01 0-63

'als RC5 Signal senden 1 Zeichen  
'send 1 IR signal  
Data "1;oa,send RC5;b,{&H00 to &H3F}"  
,

Announce2:  
'Befehl &H02 0-255  
'als RC6 Signal senden, 8 bit  
'send as RC6 signal, 8 bit  
Data "2;oa,send RC6;b"  
,

Announce3:  
'Befehl &H03 0-31  
'RC5 Adresse schreiben  
'write RC5 adress  
Data "3;oa,rc5adress;b,{0 to 31}"  
,

Announce4:  
'Befehl &H04  
'RC5 Adresse lesen  
'read RC5 adress  
Data "4;aa,rc5adress,as3"  
,

Announce5:  
'Befehl &H05 0-255  
'RC6 Adresse schreiben, 8 bit  
'write RC6 adress, 8 bit  
Data "5;oa,rc6adress;b"  
,

Announce6:  
'Befehl &H06  
'RC6 Adresse lesen  
'read RC6 adress  
Data "6;oa,rc5adress,as5"  
,

Announce7:  
'Befehl &H07 0,1  
'no\_myc schreiben  
'write no\_myc  
Data "7;oa,write no\_myc;a"  
,

Announce8:  
'Befehl &H07  
'no\_myc lesen  
'read no\_myc  
Data "8;aa,read no\_myc;a"  
,

Announce9:  
'Befehl &HF0<n><m>  
'liest announcements  
'read n announcement lines

Data "240;an,ANNOUNCEMENTS;160;14"

Announce10:

'Befehl &HFC

'Liest letzten Fehler

'read last error

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

Announce11:

'Befehl &HFD

'Geraet aktiv Antwort

'Life signal

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

Announce12:

'Befehl &HFE :

'eigene Individualisierung schreiben

'write individualization

Data "254;oa,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,5,{0 to 127};a,RS232,1;a,USB,1"

Announce13:

'Befehl &HFF :

'eigene Individualisierung lesen

'read individualization

Data "255;aa,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,5,{0 to 127};a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER\_OF\_BITS,8n1;a,USB,1"

## **I2C**

Die Default Adresse ist 5.

Mit dem Befehl &HFE03<n> kann die Adresse in n (1 ... 128) 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.

## **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.

## **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.

## **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.

## **Infrarot Ausgang**

Es muss eine geeignete Infrarot LED an SL4 angeschlossen werden

## **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 mittleren 4 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 relativ großen Kondensatoren (1206).

## **Stromversorgung**

Die Stromversorgung ist 7- 15V, Stromaufnahme ca. 40mA 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),  
R5, R6, C11, T4, SL4

Verwendung von ISP:

JP6

nur mit RS232 Schnittstelle:

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

nur mit USB

UM2102

nur I2C

R8, R9 nach Bedarf, X2, X3, 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

Infrarot (SL4)

1 Infrarot

5 GND

### **Versionen**

Diese Beschreibung gilt für die

Leiterplattenversion 02.0

Bascom Version 03.2

## '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/infrarot\\_tx\\_bascom.zip](http://dk1ri.de/dhw/infrarot_tx_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)