

# Infrarot (RC5) Empfänger

Author DK1RI, Version V02.3, 20161111

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

## Einleitung

Dieses Interface ist eine Konverter von Infrarot (RC5) Signalen auf MYC Protokoll über I2C (Slave) / RS232 / USB

Im „no\_myc“ Mode arbeitet das Interface als einfacher RC5 - ASCII Wandler.

## Beschreibung

Die Eagle Daten stehen unter [1].

Die Befehle zur Steuerung müssen binär eingegeben werden; er erfolgt keine Wandlung.

*MYC mode (default oder mit Befehl &H14H00):*

Im MYC mode erfolgt die Ausgabe der empfangenen RC5 Signale als Antwort auf den Befehl &H01:

zB im Hex Format:

0401020304

04: 4 Bytes 01, 02, 03 und 04)

Es werden nur RC5 Befehle an die eingestellte RC5 Adresse (Default 1, änderbar mir dem Befehl &H02) gespeichert.

Mit dem Lesebefehl &H01 wird der Rc5buffer gelöscht.

*Nicht MYC mode (mit Befehl &H14&H01):*

Im nicht\_MYC mode werden RC5 Adresse und RC5 Befehl dezimal zweistellig auf die serielle Schnittstelle ausgegeben.

Die MYC Befehle funktionieren nicht. Zur Rückkehr in den MYC mode muss an der seriellen Schnittstelle &H14 oder an der I2C Schnittstelle &H14&H00 eingegeben werden.

Der nicht MYC mode funktioniert nur mit der seriellen Schnittstelle.

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

## 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;Infrared (RC5) receiver;V03.2;1;160;1;11"

,

Announce1:  
'Befehl &H01  
'kopiert RC5 Daten in den Ausgang  
'copies RC5 data to output  
Data "1;aa,RC5 buffer;252,{&H00 to &H3F}"  
,

Announce2:  
'Befehl &H02 0-31  
'RC5 Adresse schreiben  
'write RC5 adress  
Data "2;oa,rc5adress;b,{0 to 31}"  
,

Announce3:  
'Befehl &H03  
'RC5 Adresse lesen  
' read RC5 adress  
Data "3;aa,as3"  
,

Announce4:  
'Befehl &H14  
'schaltet MYC / no\_MYC mode  
'switches MYC / no\_MYC mode  
Data "20;oa,no\_myc;a"  
,

Announce5:  
'Befehl &H15  
'liest MYC / no\_MYC mode  
'read MYC / no\_MYC mode  
Data "21;aa,as20"  
,

Announce6:  
'Befehl &HF0<n><m>  
'liest announcements  
'read n announcement lines  
Data "240;an,ANNOUNCEMENTS;160;11"  
,

Announce7:  
'Befehl &HFC  
'Liest letzten Fehler  
'read last error  
Data "252;aa,LAST ERROR;20,last\_error"  
,

Announce8:  
'Befehl &HFD  
'Geraet aktiv Antwort  
'Life signal  
Data "253;aa,MYC INFO;b,ACTIVE"  
,

Announce9:  
'Befehl &HFE :

'eigene Individualisierung schreiben

'write individualization

Data "254;oa,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,6,{0 to 127};a,RS232,1;a,USB,1"

Announce10:

'Befehl &HFF :

'eigene Individualisierung lesen

'read individualization

Data "255;aa,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,6,{0 to 127};a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER\_OF\_BITS,8n1;a,USB,1"

## **I2C**

Die Default Adresse ist 6

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 Eingang**

Am Infrarot Eingang muss ein geeigneter Empfänger SFH506 oder ähnlich angeschlossen werden. Bei mir funktionierte der TSOP1835, allerdings auch nur mit 22uF zwischen + und – und 47k Pullup direkt am Empfänger. Der Speisewiderstand ist 270 Ohm. Eine Begründung für dieses Verhalten fehlt.

### **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

Für RS232 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 große Kondensatoren (1206).

### **Stromversorgung**

Die Stromversorgung ist 7- 15V, Stromaufnahme ca. 30mA. Bei Verwendung des USB Moduls erfolgt die Stromversorgung darüber. Die RS232 Buchse, die Stromversorgungsbuchse und der Spannungsregler dürfen dann nicht bestückt werden.

### **LEDs**

Die LEDs müssen nicht bestückt werden. LED1 zeigt nur, daß der Prozessor läuft und blinkt. LED2 zeigt, daß ein Infrarot Signal empfangen wurde.

### **Bestückung der Leiterplatte**

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

IC1, Q1 (8MHz), C3 – C6, JP1 (muss für Normalbetrieb überbrückt werden),  
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)

2 Infrarot Daten

4 5V

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) 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) 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\\_rx\\_bascom.zip](http://dk1ri.de/dhw/infrarot_rx_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)