

# Infrarot Sender für TV Philips 22PFL2908H/12

Author DK1RI, Version V02.4 20161111

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

## Einleitung

Dieses Interface setzt das MYC Protokoll in Infrarot (RC5) Signale um.  
Das Interface bedient die Befehle des Philips Fernsehers 22PFL2908H/12.  
Es ist möglich, dass das Interface auch für andere Modelle dieser Serie funktioniert.  
Dieses Interface kann aber auch als Vorlage für andere Infrarotsender benutzt werden.  
Die Infrarotcodes wurden mit dem Infrarot receiver ermittelt.

## 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 und kann auch über die serielle Schnittstelle bedient werden.  
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;philips_tv_2908;V03.2;1;160;1;47"  
,
```

```
Announce1:  
'Befehl &H01  
'an / aus  
'on / off  
Data "1;ou,An/Aus;0;1,An/Aus"  
,
```

```
Announce2:  
'Befehl &H02  
'o  
'o  
'Data "2;ou,o;0;1,o"  
,
```

```
Announce3:  
'Befehl &H03  
||  
||
```

```

Data "3;ou,||;0;1,||"
,

Announce4:
'Befehl &H04
'<<
'<<
Data "4;ou,<<;0;1,<<"
,

Announce5:
'Befehl &H05
'>
'>
Data "5;ou,>;0;1,>"
,

Announce6:
'Befehl &H06
'>>
'>>
Data "6;ou,>>;0;1,>>"
,

Announce7:
'Befehl &H07
'source
'source
Data "7;ou,source;0;1,source"
,

Announce8:
'Befehl &H08
'tv
'tv
Data "8;ou,tv;0;1,tv"
,

Announce9:
'Befehl &H09
'format
'format
Data "9;ou,format;0;1,format"
,

Announce10:
'Befehl &H0A
'home
'home
Data "10;ou,home;0;1,home"
,

Announce11:
'Befehl &H0B
'list
'list
Data "11;ou,list;0;1,list"
,

```

Announce12:  
'Befehl &H0C  
'info  
'info  
Data "12;ou,info;0;1,info"  
,

Announce13:  
'Befehl &H0D  
'adjust  
'adjust  
Data "13;ou,adjust;0;1,adjust"  
,

Announce14:  
'Befehl &H0E  
'options  
'options  
Data "14;ou,options;0;1,options"  
,

Announce15:  
'Befehl &H0F  
'auf  
'up  
Data "15;ou,up;0;1,up"  
,

Announce16:  
'Befehl &H10  
'links  
'left  
Data "16;ou,left;0;1,left"  
,

Announce17:  
'Befehl &H11  
'rechts  
'right  
Data "17;ou,right;0;1,right"  
,

Announce18:  
'Befehl &H12  
'ab  
'down  
Data "18;ou,down;0;1,down"  
,

Announce19:  
'Befehl &H13  
'zurueck  
'back  
Data "19;ou,back;0;1,back"  
,

Announce20:  
'Befehl &H14

'ch-  
'ch-  
Data "20;ou,ch-;0;1,ch-"  
,  
Announce21:  
'Befehl &H15  
'ch+  
'ch+  
Data "21;ou,ch+;0;1,ch+"  
,  
Announce22:  
'Befehl &H16  
'laut-  
'loud-  
Data "22;ou,loud-;0;1,loud-"  
,  
Announce23:  
'Befehl &H17  
'Ton aus  
'loud off  
Data "23;ou,loud off;0;1,loud off"  
,  
Announce24:  
'Befehl &H18  
'laut+  
'loud+  
Data "24;ou,loud+;0;1,loud+"  
,  
Announce25:  
'Befehl &H19  
'rot  
'red  
Data "25;ou,red;0;1,red"  
,  
Announce26:  
'Befehl &H1A  
'gruen  
'green  
Data "26;ou,green;0;1,green"  
,  
Announce27:  
'Befehl &H1B  
'gelb  
'yellow  
Data "27;ou,yellow;0;1,yellow"  
,  
Announce28:  
'Befehl &H1C  
'blau  
'blue

Data "28;ou,blue;0;1,blue"

,

Announce29:

'Befehl &H1D

'1

'1

Data "29;ou,1;0;1,1"

,

Announce30:

'Befehl &H1E

'2

'2

Data "30;ou,2;0;1,2"

,

Announce31:

'Befehl &H1F

'3

'3

Data "31;ou,3;0;1,3"

,

Announce32:

'Befehl &H20

'4

'4

Data "32;ou,4;0;1,4"

,

Announce33:

'Befehl &H21

'5

'5

Data "33;ou,5;0;1,5"

,

Announce34:

'Befehl &H22

'6

'6

Data "34;ou,6;0;1,6"

,

Announce35:

'Befehl &H23

'7

'7

Data "35;ou,7;0;1,7"

,

Announce36:

'Befehl &H24

'8

'8

Data "36;ou,8;0;1,8"

,

Announce37:  
'Befehl &H25  
'9  
'9  
Data "37;ou,9;0;1,9"  
,

Announce38:  
'Befehl &H26  
'0  
'0  
Data "38;ou,0;0;1,0"  
,

Announce39:  
'Befehl &H27  
'subtitle  
'subtitle  
Data "39;ou,0;subtitle;1,subtitle"  
,

Announce40:  
'Befehl &H28  
'text  
'text  
Data "40;ou,0;text;1,text"  
,

Announce41:  
'Befehl &H29  
'ok  
'ok  
Data "41;ou,0;ok;1,ok"  
,

Announce42:  
'Befehl &HF0<n><m>  
'liest announcements  
'read n announcement lines  
Data "240;an,ANNOUNCEMENTS;160;47"  
,

Announce43:  
'Befehl &HFC  
'Liest letzten Fehler  
'read last error  
Data "252;aa,LAST ERROR;20,last\_error"  
,

Announce44:  
'Befehl &HFD  
'Geraet aktiv Antwort  
'Life signal  
Data "253;aa,MYC INFO;b,ACTIVE"  
,

Announce45:  
'Befehl &HFE :

'eigene Individualisierung schreiben

'write individualization

Data "254;oa,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADDRESS,15,  
{0 to 127};a,RS232,1;a,USB,1"

Announce46:

'Befehl &HFF :

'eigene Individualisierung lesen

'read individualization

Data "255;aa,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADDRESS,15,  
{0 to 127};a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER\_OF\_BITS,8n1;a,USB,1"

## **RC5 Adresse.**

Die RC5 Adresse des TV Gerätes ist 1.

## **I2C Adresse**

Die Default Adresse ist 15.

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.

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.

## **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 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.

### **Infrarot Ausgang**

Es muss eine geeignete Infrarot LED 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 V02.0

Bascom Version V03.2

## **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\\_philips\\_tv\\_2908\\_bascom.zip](http://dk1ri.de/dhw/infrarot_tx_philips_tv_2908_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)