

Morse Sender

Author DK1RI, Version V02.3, 20200517

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

Einleitung

Dieses Gerät kann in einem MYC System verwendet werden, aber auch unabhängig davon mit (binären) Befehlen gesteuert werden. Die Befehle sind im Kapitel „Einbindung in das MYC System“ beschrieben.

Das Interface arbeitet als Slave am I2C Bus oder kann über RS232 /USB gesteuert werden. Defaultmäßig sind alle Schnittstellen aktiv. Mit dem Initialisierungsbefehl können diese aber deaktiviert werden. Der Initialisierungsbefehl funktioniert aber immer.

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

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. 20mA max oder über USB.

Das Programm ist nur das Framework, weitere MYC Befehle zur Steuerung anderer Geräte mit Morsecodes müssen nach Bedarf eingefügt werden. Man kann aber Strings direkt entsprechend den MYC Regeln eingeben, die dann in Morsesignale umgesetzt werden.

Das Interface kann auch als selbstständiger Morsetrainer verwendet werden.

Die Tonhöhe des Morsesignals ist von 100Hz bis 2kHz einstellbar, die Geschwindigkeit von 5 bis 100 Wpm. Diese Werte sind nicht sehr genau.

Es gibt drei Betriebsarten; die Betriebsart bleibt nach dem Ausschalten erhalten:

MYC_mode, default (&H0600)

Die Befehlsdaten müssen binär eingegeben werden; er erfolgt keine Wandlung, hier als Beispiel im HEX Format:

010447465737

01 ist der Befehl zur Morse Ausgabe. 04 ist die Länge des nachfolgenden Strings test

Die Zeichen werden auf der seriellen Schnittstelle ausgegeben. Werden weitere Befehle eingegeben, bevor alle Zeichen ausgegeben wurden, gehen diese verloren.

Es können 250 Zeichen maximal am Stück übertragen werden; die Länge des Strings wird mit einem Byte übertragen.

Direkt_mode, nicht MYC Mode (&H0601):

Alle erlaubten Morsezeichen werden in Morsezeichen umgesetzt, sobald ein LF (&H0A) empfangen wurde.

Die Zeichen werden auf der seriellen Schnittstelle ausgegeben.

Dieser Mode wird durch einen gültigen Befehlscode (&H00 bis &H07, >= &H240 sofort verlassen..

Verschiedene 5er Gruppen (&H0602 bis &H0608):

Es gibt die Möglichkeit, 5er Gruppen mit verschiedenen Zeichensätzen auszugeben

Der Mode wird durch eine beliebige weitere Eingabe beendet.

Einbindung in das MYC System

Details zum MYC System stehen in [3].

Folgende Befehle werden akzeptiert::

Announce:

'Befehl &H00

'eigenes basic announcement lesen

'basic announcement is read to I2C or output

Data "0;m;DK1RI;morse sender;V04.2;1;190;1;15;1-1"

,

'Announce1:

'Befehl &H01 <s>

'I2C / serial String als Morse senden

'send I2C / serial String as Morse

Data "1;oa,send morse;250,{.,\,,:;\;?,-,_,(,)',=,+/,@,0 to 10,a to z, A to Z}"

,

'Announce2:

'Befehl &H02 0 to 19

'Geschwindigkeit schreiben

'write speed

Data "2;op,morse speed;1;20,{5 to 100};lin;Wpm"

,

'Announce3:

'Befehl &H03

'Geschwindigkeit lesen

'read speed

Data "3;ap,as2"

,

'Announce4:

'Befehl &H04

'Frequenz schreiben 100 - 2000Hz

'write frequency

Data "4;op,morse frequency;1;20,{100 to 2000};lin;Hz"

,

'Announce5:

'Befehl &H05

'Frequenz lesen

'read frequency

Data "5;ap,as4"

,

'Announce6:

'Befehl &H06

'Mode einstellen, Myc, direkteingabe, 5er Gruppen

'set mode

Data "6;os,mode;1;0,myc mode;1,morse input;2,0 to 9;3,a to f;4,g to l;5,m to s;6,t to

z;7,special;8,all"

,

```

'Announce7:
'Befehl &H07
'Morse mode lesen
'read morse mode
Data "7;as,as6"
,

'Announce8:
'Befehl &HF0<n><m>
'liest announcements
'read n announcement lines
Data "240;ln,ANNOUNCEMENTS;190;15"
,

'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 :
'eigene Individualisierung schreiben
'write individualization
Data "254;ka,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,18,
{0 to 127};a,SERIAL,1"
,

'Announce12:
'Befehl &HFF :
'eigene Individualisierung lesen
'read individualization
Data "255;la,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,18,{0
to 127};a,SERIAL,1;b,BAUDRATE,0,{19200};3,NUMBER_OF_BITS,8n1"
,

'Announce13:
Data "R !* IF $6&1"
,

'Announce14:
Data "R &6 IF $6&1"
,

```

Regeln

Während des Morse_mode = 1 kann nur der Morse_mode geändert 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 JP5 beim Anlegen der Versorgungsspannung überbrückt, werden wieder die Defaultwerte eingelesen. Dies ist hilfreich, wenn die aktuelle I2C Adresse verloren gegangen ist.

Watchdog

Es gibt einen kompletten Hardware-reset, wenn die Hauptschleife länger als 2 Sekunde dauert. Zusätzlich gibt es drei weitere Watchdogs, die in der vorliegenden Firmware für Tests und „nicht_MYC Betrieb“ nach ca 10 Sekunden ansprechen. Für „MYC Betrieb“ sollte der Wert auf 1 Sekunde gesetzt werden.

Die Befehlseingabe und Ausführung muss in dieser Zeit 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 I2C Lesebefehl müssen die Daten innerhalb dieser Zeit vom I2C Master abgeholt werden. Danach werden die Daten gelöscht. Neue Befehle können erst eingegeben werden, wenn alle Daten abgeholt wurden oder die Watchdog Zeit abgelaufen ist. Wird die RS232 / USB Schnittstelle verwendet, werden die Daten sofort ausgegeben.

Bei einem I2C BusLock (SDA pin auf 0) erfolgt auch ein kompletter Restart. Das bedeutet, dass bei einem Kurzschluss auf der I2C Leitung das Interface nicht funktioniert!

Software

Die Steuerung übernimmt ein AVR Mikrocontroller Atmega168.

Das aktuelle Bascom Programm verwendet einen Atmega328.

Die Software wurde in BASCOM geschrieben [2]

Um das Programm zu kompilieren, muss das Verzeichnis common_1.10 [6] in das Verzeichnis mit dem Programm kopiert werden

Programmierung des Prozessors

Zur Programmierung des Prozessors ist ein 6poliger ISP Stecker JP6 vorgesehen.

Die Fuses müssen möglicherweise programmiert werden (siehe Bascom Programm) !! Prozessortyp und Frequenz müssen gegebenenfalls angepasst werden.

Der Jumper J1 sollte während der Programmierung entfernt werden.

Serielle (RS232 / USB) Schnittstelle

Schnittstellenparameter: 19k2 8N1

Bei Verwendung der RS232 Schnittstelle muss bei Jumper JP8 und JP9 ist jeweils Pin1 und Pin2 überbrückt werden.

Alternativ zur RS232 Schnittstelle kann die USB Platine UM2102 von ELV verwendet werden. Die USB Platine wird plan auf der Oberseite der Interfaces verlötet: der USB Stecker zeigt zum Rand. Die mittleren 4 pins des Verbinders ST2 sind mit dem 4 Anschlusspunkten JP7 auf dem

Interface zu verbinden. USB Platine und Interface müssen voneinander isoliert werden.
Die Stromversorgung erfolgt dann über USB.

I2C Schnittstelle

Die Default Adresse ist 18 (&H12).

Mit dem Befehl &HFE03<n> kann die Adresse in n (1 ... 127) geändert werden.

Pullup Widerstände R3 / R4 müssen immer bestückt werden (1k - 10k).

Mit JP2 kann festgelegt werden, ob der I2C Bus mit 3V oder 5V betrieben wird.

Bei anderer I2C Spannung als 3V kann R5 / R6 angepasst werden.

Wenn auf den 3V Betrieb völlig verzichtet werden soll, kann IC3 (PCA9517), R1, R2, R5, R6, JP2 entfallen und alternativ wird JP3 und JP4 bestückt.

Ganz ohne I2C kann auch SL1, SL2, JP3, JP4 entfallen.

Der Gesamtwiderstand am I2C Bus sollte bei 1 bis 10 kOhm je nach Leitungslänge liegen

Mit IC3 muss R1 / R2 ($\leq 10k$) bestückt werden. Wenn auf IC3 verzichtet wird und JP3 / JP4 verwendet wird, muss berücksichtigt werden, dass R1 / R2 parallel zu R3 / R4 liegt. R1 / R2 kann also gegebenenfalls entfallen.

SL1 und SL2 sind parallel geschaltet. Ein Anschluss kann zur Weitergabe des I2C Signals an das nächste Gerät verwendet werden.

Um Buslocks zu vermeiden, wird circa alle 200ms geprüft, ob das SDA Signal auf „0“ liegt.

Ist das 50 mal hintereinander der Fall, wird die I2C Schnittstelle neu gestartet.

Morse Ausgang

Das Morsesignal kann direkt herausgeführt werden (R15 = 0 Ohm, C12 entfällt) oder über einen Tiefpass (1k/100nF). 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.

SMD

Die Leiterplatte ist teilweise mit SMD bestückt.

Stromversorgung

Die Stromversorgung ist 7- 15V, Stromaufnahme ca. 20mA max.

Alternativ erfolgt die Stromversorgung über USB

Bestückung der Leiterplatte

Da die Leiterplatte auch für andere Anwendungen eingesetzt werden kann, brauchen nur folgende Bauteile bestückt werden: X1, D1, IC1, IC2, Q1, C1 – C5, C10, JP1(muss für Normalbetrieb überbrückt werden), R3, R4, SL4, ohne Tiefpass: R15=0

Verwendung von ISP:

JP6

Mit I2C:

Siehe I2C oben.

Mit serieller Schnittstelle:

Bei Verwendung der RS232 Schnittstelle wird IC4 und C6 – C9, JP8, JP9 bestückt. Alternativ dazu kann der USB Modul UM2102 verwendet werden. Dann darf auch X1, D1, C1, IC2 nicht bestückt werden!

Anschlüsse

Power

Tip 12V

Ring GND

RS232 (Buchse)

5 GND

2 Jumper

3 Jumper

I2C (SL1, SL2)

1 GND

2 SCL

3 SDA

Morsesignal (SL4)

1 Morsesignal

5 GND

Jumper

JP1 Power

JP2 I2C: 3V/5V Umschaltung

JP3 SDA Überbrückung (ohne IC3)

JP4 SCL Überbrückung (ohne IC3)

JP5 Reset

JP6 pin1 – 6: ISP, pin10 : SS für den SPI Anschluss

JP7 Anschluss für USB Modul

Versionen

Diese Beschreibung gilt für die

Leiterplattenversion V05.1

Bascom Version V04.3

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](https://www.dk1ri.de/dhw/i2c_rs232_interface_eagle.zip)
- [2] [dk1ri.de/dhw/morse_sender_bascom.zip](https://www.dk1ri.de/dhw/morse_sender_bascom.zip)
- [3] <https://www.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)
- [6] https://dk1ri.de/dhw/common_1.10.zip