

I2C zu RS232 / USB Interface

Author DK1RI, Version V05.1, 20200501

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

Einleitung

Diese beiden Interfaces sollen im wesentlichen zum Test von MYC Devices und für Softwaretests dienen.

Mit einem Interface als Master und einem als Slave können die Grundfunktionen eines MYC Systems demonstriert werden.

Das Interface als Master ist auch als seriell - I2C Konverter verwendbar.

Als serielle Schnittstelle kann je nach Bestückung RS232 oder USB verwendet werden; sowohl für Master als auch Slave.

Das Interface als Slave kann als Prototyp für andere Interfaces mit I2C Device - Schnittstelle verwendet werden. Als Interface für andere Geräte muss das Interface andere / weitere Befehlskonvertierungen machen.

Beschreibung

Es gibt eine Firmwareversion als I2C Master und eine als Slave.

Die Eagle Daten für die Leiterplatte stehen unter [1].

Die Stromversorgung ist 7- 10V, Stromaufnahme ca. 20mA max oder über USB.

Einbindung in das MYC System

Dieses Interface kann als I2C Master für die Bedienung anderer MYC Devices ohne command-router verwendet werden.

Dabei wird an der RS232 Schnittstelle (19,2kB, 8N1) oder an der USB Schnittstelle ein Rechner mit Terminalprogramm angeschlossen .

Wenn dann das gleiche Interface als Slave angeschlossen wird, arbeitet das Master Interface (fast) wie ein command-router mit einer seriellen Schnittstelle für eine Benutzerschnittstelle oder übergeordnetes MYC System. Auf der seriellen Seite werden die MYC Befehle eingegeben, so wie die Benutzerschnittstelle in einem MYC System das tun muss.

Das Interface nimmt auf der Befehlsschnittstelle (I2C beim Slave, seriell beim Master) Daten entgegen und interpretiert das erste Zeichen als Befehl und handelt entsprechend. Wird ein Befehl oder Parameter als ungültig erkannt, wird das nächste Zeichen als neuer Befehl interpretiert.

Der Befehl wird ausgeführt, wenn er mit Parametern komplett und gültig erkannt wurde.

Ein nicht vollständiger Befehl wird nach ca 1 s gelöscht.

Die Strings – maximal 250 Zeichen - müssen mit der Zahl der Bytes - 1 Byte - beginnen!

Interface als Slave:

Dieses Interface ist ein einfaches Device mit (neben den reservierten Befehlen) nur 2 Befehlen an der I2C Schnittstelle: der Schreibbefehl gibt den von der I2C Schnittstelle empfangenen String (ohne Länge) auf die serielle Schnittstelle. Der Lesebefehl kopiert die Daten des seriellen Eingabepuffers in den I2C Puffer, wo der I2C Master die Daten abholen kann. Dabei wird die Länge des aktuellen Strings hinzugefügt. Die Abholung kann in mehreren Paketen erfolgen. Werden mehr Daten abgerufen als vorhanden, sind diese ein &H80 (ein nicht gültiges Kommando). Weitere

Lesebefehle werden ignoriert, bis die Daten abgerufen wurden; MULTI_ANSWER = 0 (siehe [6])
Schreibbefehle sind möglich, ohne dass die Lesedaten abgeholt werden.

Um Blockaden zu vermeiden, müssen Daten von Lesebefehlen innerhalb von 10 Sekunde abgeholt werden. Danach werden sie gelöscht. Die Firmware kann einfach auf eine Zeit von 1 Sekunden geändert werden.

Eingaben auf der seriellen Schnittstelle werden geecho und in der seriellen Puffer kopiert. Wenn kein Lesebefehl erfolgt, werden nach 252 Zeichen weitere Zeichen ignoriert. Weitere Funktionen sind an der seriellen Schnittstelle nicht möglich.

Beispiele für die Eingabe an der I2C Schnittstelle im HEX Format:

&H010474657374 01 ist der Befehl des Slave zur Ausgabe auf der seriellen Schnittstelle.
Gibt 4 Zeichen des Strings „test“ ohne CRLF auf der seriellen Schnittstelle aus.

Ist die Länge größer als 252 Byte wird der Befehl ignoriert.

Der Befehl wird ausgeführt, sobald er komplett ist. Während der Ausführung wird die I2C Schnittstelle nicht abgefragt.

Werden mehr Zeichen übertragen als in der Länge angegeben, wird der Rest anschließend als neuer Befehl interpretiert.

&H010474657374

test Ausgabe an der seriellen Schnittstelle

test Eingabe an der seriellen Schnittstelle

&H02 ist der Lesebefehl vom Master

&H020474657374 wird in den I2C Buffer kopiert, und der Master abrufen kann

oder

&H02

&H02037465737480 Antwort an den Master, wenn der 7 Bytes abruf

Interface als Master:

Normalerweise ist in einem MYC System an einem I2C Bus der Command-router der Master. Das Interface soll also einen manuell bedienbaren command-router emulieren.

Ein richtiger commandrouter kennt die Befehle des angeschlossenen Devices und die Adresse (bei I2C). Das Interface hat daher zwei Betriebsarten:

- für den Anschluss eines gleichen Interfaces als Slave, No_myc = 0 (default)

- für den Anschluss beliebiger (MYC -) Devices oder als Protokollwandler, No_myc = 1

Einige Befehle funktionieren nur abhängig vom der gewählten Betriebsart.

Anschluss eines gleichen Interfaces als Slave, der Master ist "commandrouter"

Dieser Mode wird durch den Befehl &H0B00 eingestellt. Die Befehle &H01, &H02 gelten nicht. Dann verhält das Interface wie ein command-router – es kennt die Slave Befehle - und setzt die Befehle für das Slave Interface richtig um.

Die Eingabe an der seriellen Schnittstelle hat ein Echo. (ein command-router allerdings nicht)

Die Ausgabe erfolgt ohne Abschluss (CR LF).

Wie bei einem richtigen command-router werden die Individualisierungs und announce commands des Slave nicht an den Slave weitergegeben und der Befehl &H00 des Slave vom Master selbst beantwortet.

An der seriellen Schnittstelle müssen alle Werte als Bytewerte (ein Byte von 0 bis 255) eingegeben werden!!. Es erfolgt keine Umwandlung (zB vom Hex auf Binär). Eingabe auf der Windows

Tastatur mit <ALT> und 3 Ziffern des ASCII Codes. Einfacher ist es, wenn das Terminalprogramm die Eingabe in Hexformat zulässt.

Bei den folgenden Beispielen werden zwei gleiche Interfaces angenommen; eins als Master und eins als Slave konfiguriert.

Entsprechend den MYC Regeln lautet ein Schreibbefehl zum Beispiel (gezeigt im HEX Format):
&H040474657374

&H04 ist der Befehl an das Master-Interface zum Schreiben, der auf den I2C Bus als Befehl &H01 weitergegeben wird,

Die 04 ist die Länge des folgenden Strings. Danach folgt der zu übertragende Text (test).

Bei dem gleichen Interface als Slave wird der Text -test- auf der seriellen Schnittstelle des Slave ausgegeben.

Ein Lesebefehl (ebenso im HEX Format)

&H0503

&H12 ist der Befehl zum Lesen. Der Befehl (weitergegeben als &H02) bewirkt beim Slave, dass der String im Eingabepuffer mit vorangestellter &H02 und Länge des String in den Ausgabepuffer geschrieben wird. Danach liest der Master den String und gibt ihn an die serielle Schnittstelle.

Wurden auf der Slave Seite nicht genügend Daten eingegeben, wird nur statt der Daten ein ungültiger Commandtoken (&H80) ausgegeben.

Die angegebene Länge (252) im announcement ist die Maximallänge des String, die der Master behandeln kann..

Der Befehl

&H03

liefert das basic announcement des Slave. Die Daten liefert der Master, ohne den Slave zu fragen; so wie es der command-router machen würde.

Der Befehl &H06 liefert den letzten Fehler des Slave.

Anschluss eines beliebigen Devices:

Dieser Mode wird durch &H0B01 eingestellt. Die Befehle &H03 - &H07 gelten nicht. Das Interface verhält sich wie ein MYC device.

Mit &H08xx wird die I2C Adresse des Slave eingestellt.

Beispiel zum Senden &H01

&H0106010474657374

Nach dem Befehl &H01 beginnt der Sendestring mit der Länge des folgenden String (06). Das folgende (dritte) Zeichen kann der Befehlstoken an ein angeschlossenes Device sein. 01 gibt folgenden String der Länge 04 am gleichen Interface als Slave aus: test

Bei gleichem Interface als Slave wird zum Lesen zunächst der Lesebefehl 02 gesendet. Dieser Befehl kopiert zum Beispiel 4 Bytes des RS232 Puffers in den I2C Puffer des Slave:

&H010102

Danach werden 6 Bytes abgeholt:

&H0206

Das Ergebnis ist:

&H0206020474657374

Das erste Byte ist der eigene Token, dann die eigene Länge und dann die 6 Bytes der Antwort.

Da in diesem Fall sich der Master wie ein device verhält, erscheint der Token und die Länge doppelt.

Andere Befehle geben ggf andere Datentypen und andere Datenlängen zurück.

Die I2C Adresse des Slave kann mit &H08xx eingegeben werden. Die Adresse xx muss im Bereich

von 1 – 127 sein.

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:

Master:

Announce:

'Befehl &H00

'basic annouement wird gelesen

'basic announcement is read

Data "0;m;DK1RI;RS232_I2C_interface Master;V06.1;1;120;1;19;1-1"

,

'Announce1:

'Befehl &H01 <s>

'string an angeschlossenes device schicken, Myc_mode = 0

'write string to device

Data "1;oa;250"

,

'Announce2:

'Befehl &H02

'string von angeschlossenenem device lesen, Myc_mode = 0

'read string from device

Data "2;aa,as1"

,

'Announce3:

'Befehl &H03

'übersetzes 0 des slave Myc_mode = 1

'translated 0 of slave

Data "3;m;DK1RI;RS232_I2C_interface Slave;V06.0;1;90;1;8;1-1"

,

'Announce4:

'Befehl &H04 <s>

'übersetzes 1 des slave Myc_mode = 1 I2C nach RS232

'translated 1 of slave I2C to RS232

Data "4,oa;250"

,

'Announce5:

'Befehl &H05

'übersetzes 2 des slave Myc_mode = 1 , RS232 nach I2C

'translated 2 of slave, RS232 to I2C

Data "5,aa,as4"

,

'Announce6:

'Befehl &H67

'übersetzes 252 des slave Myc_mode = 1

'translated 252 of slave,

```

Data "6;aa,LAST ERROR;20,last_error"
,
'Announce7:
'Befehl &H07
'Uebersetztes 253 des slave Myc_mode = 1
'translated 253 of slave,
Data "7;aa,MYC INFO;b,ACTIVE"
,
'Announce8:
'Befehl &H08 <0..127>
'Adresse zum Senden speichern
'write send adress
Data "8;oa,I2C adress;b,{0 to 127}"
,
'Announce9:
'Befehl &H09
'Adresse zum Senden lesen
'read send adress
Data "9;aa,as8"
,
'Announce10:
'Befehl &H0A 0|1
'no_myc speichern
'write no_myc
Data "10;oa,no_myc;a"
,
'Announce11:
'Befehl &H0B
'MYC_mode lesen
'read myc_mode
Data "11;aa,as10"
,
'Announce12:
'Befehl &HF0<n><m>
'liest announcements
'read n announcement lines
Data "240;ln,ANNOUNCEMENTS;120;19"
,
'Announce13:
'Befehl &HFC
'Liest letzten Fehler
'read last error
Data "252;aa,LAST ERROR;20,last_error"
,
'Announce14:
'Befehl &HFD
'Geraet aktiv Antwort
'Life signal
Data "253;aa,MYC INFO;b,ACTIVE"
,

```

```

'Announce15:
'Befehl &HFE <n><data>
'eigene Individualisierung schreiben
'write individualization
Data "254;ka,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,0;b,ADRESS,0,{0
to 127};a,SERIAL,1"
'

'Announce16:
'Befehl &HFF <n> :
'eigene Individualisierung lesen
'read individualization
Data "255;la,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,0;b,ADRESS,0,{0
to 127};a,SERIAL,1;b,BAUDRATE,0,{19200};3,NUMBER_OF_BITS,8n1"
'

'Announce17:
Data "R !($1 $2) IF $11=1"
'

#Announce18:
Data "R !($3 $4 $5 $6 $7) IF $11=0"

Slave:

Announce:
'Befehl &H00
'basic announcement wird gelesen
'basic announcement is read to I2C
Data "0;m,DK1RI;Rs232_i2c_interface Slave;V06.1;1;200;1;8;1-1"
'

'Announce1:
'Befehl &H01 <s>
'Sendet Daten von I2C nach RS232
'read data from I2C, write to RS232 (write to device)
Data "1,oa;250"
'

'Announce2:
'Befehl &H02
'liest Daten von RS232 nach I2C
'read data from RS232, write to I2C (read from device)
Data "2,aa,250"
'

'Announce3:
'Befehl &HF0<n><m>
'announcement aller Befehle lesen
'read m announcement lines
Data "240;ln,ANNOUNCEMENTS;200;8"
'

'Announce4:
'Befehl &HFC
'Liest letzten Fehler
'read last error

```

Data "252;aa,LAST ERROR;20,last_error"

'Announce5:

'Befehl &HFD

'Geraet aktiv antwort

'Life signal

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

'Announce6:

'Befehl &HFE :

'eigene Individualisierung schreiben

'write individualization

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

'Announce7:

'Befehl &HFF :

'eigene Individualisierung lesen

'read individualization

Data "255;la,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,1,{0 to 127};a,SERIAL,0;b,BAUDRATE,0,{19200};3,NUMBER_OF_BITS,8n1"

Regeln

Es gibt zwei Zeilen mit Regeln für den Master.

Abhängig vom No_myf funktionieren einige Befehle nicht.

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 I2C reset (Slave)

Software

Die Steuerung übernimmt ein AVR Mikrocontroller Atmega8 oder größer.
Das aktuelle Bascom Programm verwendet einen Atmega8.
Die Software wurde in BASCOM geschrieben [2]
Um das Programm zu kompilieren, muss das Verzeichnis common_1.7 [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 1 für den Slave
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 (<=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 (Slave).

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.

Verwendung von ISP:

JP6

Mit I2C:

Siehe I2C oben.

Mit serieller Schnittstelle:

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

Anschlüsse

Power

Tip 12V

Ring GND

RS232 (Buchse)

5 GND

2 Jumper

3 Jumper

I2C

1 GND

2 SCL

3 SDA

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
JP8/JP9 RS232 Umschaltung

Versionen

Diese Beschreibung gilt für die
Leiterplattenversion V05.1
Bascom Version V06.0 für Master und Slave

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] https://www.dk1ri.de/dhw/i2c_rs232_interface_eagle.zip
- [2] https://www.dk1ri.de/dhw/i2c_rs232_interface_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