I2C zu RS232 / USB Interface

Author DK1RI, Version V03.6, 20180112
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.
Als Testinterface wird an der RS232 Schnittstelle (19,2kB, 8N1) oder an der USB Schnittstelle ein Rechner mit Terminalprogramm angeschlossen .

Einbindung in das MYC System

Dieses Interface kann als I2C Master für die Bedienung eines MYC Devices ohne command-router verwendet werden.
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 commands eingeben, 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!

I2C 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. 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])

Eingaben auf der seriellen Schnittstelle werden geechot und in der seriellen Puffer kopiert. Wenn kein Lesebefehl erfolgt, werden nach 252 Zeichen weitere Zeichen ignoriert.
Die Länge des Strings wird dann hinzugefügt, wenn der String in den I2C Puffer übertragen wird.
Schreibbefehle sind auch möglich, ohne dass die Lesedaten abgeholt werden.
Um Blockaden zu vermeiden, müssen Daten von Lesebefehlen innerhalb einer Sekunde abgeholt werden. Danach werden sie gelöscht. Die Firmware kann einfach auf eine Zeit von 10 Sekunden (für Tests) geändert werden.

Beispiele im HEX Format:

010474657374	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.

02	ist der Lesebefehl (kopiert den RS232 Puffer in den I2C Puffer)
020474657374	Antwort an den Master: der ruft 4 gültige Bytes ab
oder	
02	Es gibt aber nur 3 gültige Bytes:
020374657380	Antwort mit 3 gültigen Bytes (80 ist ein ungültiger commandtoken)

I2C 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, Myc_mode = 1
- für den Anschluss beliebiger (MYC -) Devices oder als Protokollwandler, Myc_mode = 0 (default)
Einige Befehle funktionieren nur abhängig vom gewählten Myc_mode.

<u>Anschluss eines gleichen Slave-Interfaces, der Master ist "commandrouter"</u>

Dieser Mode wird durch den Befehl &HEE01 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 RS232 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 die Benutzerschnittstelle des Master weitergegeben und der Befehl &H00 des Slave vom Master selbst beantwortet.
An der RS232 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):
110474657374
11 ist der Befehl an das Master-Interface zum Schreiben, der auf den I2CBus als Befehl 01 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 RS232 Schnittstelle ausgegeben.
Ein Lesebefehl (ebenso im HEX Format)
12 03
12 ist der Befehl zum Lesen. Der Befehl (weitergegeben als 02) 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 RS232 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
&H10
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 &H13 liefert den letzten Fehler des Slave.

<u>Anschluss eines beliebigen Devices:</u>

Dieser Mode wird durch &HEE00 eingestellt und ist der Default Mode. Die Befehle &H10 - &H13 gelten nicht.
Da so beliebige devices angeschlossen werden können, kennt der Master auch das Datenformat der Antworten nicht. Abweichend vom MYC Protokoll in „richtigen“ MYC Systemen muss daher beim Lesebefehl auch die Zahl der gewünschten Bytes angegeben werden.

Beispiel zum Senden &H01
0106010474657374
Nach dem Befehl 01 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 I2C - RS232 Interface (Slave) aus: test
Bei gleichem Interface als Slave muss der Lesebefehl 02 mit Zahl der gewünschten Bytes gesendet werden. Dieser Befehl kopiert 4 Bytes des RS232 Puffers in den I2C Puffer des Slave. Danach werden 6 Bytes abgeholt:
01020204
0206
Das Ergebnis ist:
0206020474657374
Das erste Byte ist der eigene Token, dann die eigene Länge und dann die 6 Byte 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 &HECxx 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:

```
Announce0:
'Befehl &H00
'basic annoumement wird gelesen
'basic announcement is read
Data "0;m;DK1RI;RS232_I2C_interface Master;V05.0;1;120;1;20;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 angeschlossenem device lesen, Myc_mode = 0
'read string from device
Data "2;aa,as1"
'
Announce3:
'Befehl &H10
'übersetzes 0 des slave Myc_mode = 1
'translated 0 of slave
Data "16;m;DK1RI;RS232_I2C_interface Slave;V04.0;1;90;1;8;1-1"
'
Announce4:
'Befehl &H11 <s>
'übersetzes 1 des slave Myc_mode = 1 I2C nach RS232
'translated 1 of slave I2C to RS232
Data "17,oa;250"
'
Announce5:
'Befehl &H12
'übersetzes 2 des slave Myc_mode = 1 , RS232 nach I2C
'translated 2 of slave, RS232 to I2C
Data "18,aa,as17"
'
Announce6:
'Befehl &H13
'übersetzes 252 des slave Myc_mode = 1
'translated 252 of slave,
Data "19;aa,LAST ERROR;20,last_error"
'
```

```
Announce7:
'Befehl &H14
'übersetzes 253 des slave Myc_mode = 1
'translated 253 of slave,
Data "20;aa,MYC INFO;b,ACTIVE"
'
Announce8:
Data "I;DK1RI;RS232_I2C_interface Master;V05.0;Device 1;1;DK1RI;Rs232_i2c_interface
Slave;V05.0;Device 1;1"
'
Announce9:
'Befehl &HEC <0..127>
'Adresse zum Senden speichern
'write send adress
Data "236;oa,I2C adress;b,{0 to 127}"                                    '
'
Announce10:
'Befehl &HED
'Adresse zum Senden lesen
'read send adress
Data "237;aa,as236"
'
Announce11:
'Befehl &HEE 0|1
'MYC_mode speichern
'write myc_mode
Data "238;oa,MYC mode;a"                                    '
'
Announce12:
'Befehl &HEF
'MYC_mode lesen
'read myc_mode
Data "239;aa,as238"
'
Announce13:
'Befehl &HF0<n><m>
'liest announcements
'read n announcement lines
Data "240;ln,ANNOUNCEMENTS;120;20"
'
Announce14:                                    '
'Befehl &HFC
'Liest letzten Fehler
'read last error
Data "252;aa,LAST ERROR;20,last_error"
 '
Announce15:                                    '
'Befehl &HFD
'Geraet aktiv Antwort
'Life signal
```

```
Data "253;aa,MYC INFO;b,ACTIVE"
'
Announce16:
'Befehl &HFE <n><data>
'eigene Individualisierung schreiben
'write individualization
Data "254;ka,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,RS232,1;a,USB,1"
'
Announce17:
'Befehl &HFF <n> :
'eigene Individualisierung lesen
'read individualization
Data "255;la,INDIVIDUALIZATION;20,NAME,Device
1;b,NUMBER,1;a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER_OF_BITS,8n1;a,USB,1"
'
Announce18:
Data "R !($1 $2) IF $239=1"
'
Announce19:
Data "R !($16 $17 $18 $19 $20) IF $239=0"
'
```

Slave:

```
Announce0:
'Befehl &H00
'basic annoumement wird gelesen
'basic announcement is read to I2C
Data "0;m;DK1RI;Rs232_i2c_interface Slave;V05.0;1;90;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;90;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 <0..3> <data>
'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}"
'
Announce7:
'Befehl &HFF <0 .. 3> :
'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}"
'
```

Regeln

Es gibt vier Zeilen mit Regeln für den Master.
Abhängig vom Myc_mode funktionieren einige Befehle nicht.
Weitere Regeln beschreiben, was passiert, wenn die Befehlsschnittstellen abgeschaltet werden: beim Master lassen sich nicht bei die Schnittstellen gleichzeitig abschalten.

I2C Adresse

Die Default Adresse ist 1 für den Slave.
Mit dem Befehl &HFE03<n> kann die Adresse in n (1 … 127) 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.
Der Master ändert die Adresse, an die er sendet, mit dem Befehl &HEExx.

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. Der Reset hat eine Verzögerungszeit von 1 Sekunde.

Watchdog

Die Befehlseingabe und Ausführung muss in weniger als 1 Sekunde beendet sein. Danach werden die bereits empfangenen Daten gelöscht. Dies soll falsche Eingaben vermeiden. Mit dem &HFC "letzten Fehler" Befehl kann man Eingabefehler sehen.Die Befehlseingabe und Ausführung muss in weniger als 1 Sekunde 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 Lesebefehl müssen die Daten innerhalb von 10 Sekunden vom I2C Master abgeholt werden – wenn die I2C Schnittstelle gerade verwendet wird. Danach werden die Daten gelöscht. Diese Zeit kann mit dem Wert Tx_factor im Bascom Programm geändert werden. Neue Befehle können erst eingegeben werden, wenn alle Daten abgeholt wurden. Wird die RS232 / USB Schnittstelle verwendet, werden die Daten sofort ausgegeben.
Es gibt einen kompletten Reset, wenn die Hauptschleife länger als 2 Sekunde dauert, zum Beispiel, wenn die I2C Schnittstelle nicht korrekt arbeitet.

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.

RS232 Schnittstelle

Schnittstellenparameter: 19k2 8N1
Es muss bei Jumper JP7 und JP8 ist jeweils Pin1 und Pin2 (die jeweils äußeren pins) überbrückt werden.

USB Schnittstelle

Das Interface kann auch 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 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 gesamte 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).
Es gibt einen nicht getesteten Leiterplattenentwurf für eine recht kleine Leiterplatte mit Minimalbestückung und TQFP Prozessor [4] Die muss mit dem ELV USB Modul bestückt werden.

Stromversorgung

Die Stromversorgung ist 7- 15V, Stromaufnahme ca. 20mA 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 (ATMega168), Q1 (20MHz), C3 – C6, X2 X3 (2.5mm Klinke), JP1 (muss für Normalbetrieb überbrückt werden), JP4

mit RS232 Schnittstelle:
IC2, IC3 (7805 oder pinkompatible Schaltregler), D1, C1, C2, C7 – C10, JP7, JP8 (jeweils Pin1 und Pin 2 überbrücken), X1 Buchse fur Hohlstecker), X4 (DB9 Buchse)

Verwendung von ISP:
JP6

mit I2C:
X2, X3, R8, R9 nach Bedarf

Mit USB Schnittstelle (alternativ zu RS232):
UM2102, JP10

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

Versionen

Diese Beschreibung gilt für die
Leiterplattenversion V02.1

Bascom Version V05.0 für Master und Slave

Copyright

Referenzen

[1] dk1ri.de/dhw/i2c_rs232_interface_eagle.zip
[2] dk1ri.de/dhw/i2c_rs232_interface.zip
[3] dk1ri.de/myc/MYC.pdf
[4] dk1ri.de/myc/Description.pdf (englisch)
[5] dk1ri.de/dhw/i2c_rs232_interface_eagle_minimal.zip
[6] dk1ri.de/myc/Reseved_tokens.pdf
[7] dk1ri.de/myc/Definitions.pdf