

Icom Interfaces für das MYC Protokoll

Author DK1RI, Version V02.1, 20200315

This paper is published in <https://github.com/dk1ri> as well.

Einleitung

Dies ist die Beschreibung eines Interfaces für einige ICOM (TM) Transceiver; siehe auch: *Details zu den einzelnen Tranceivern*

Dieses Interface kann in einem MYC System verwendet werden, aber auch unabhängig davon mit (binären) Befehlen gesteuert werden.

Beschreibung

Das Interface setzt MYC Befehle in das CIV Protokoll um.

Es werden die meisten CIV Befehle des Radios direkt unterstützt.

Es gibt zwei Versionen:

Software (für Tranceiver mit USB Schnittstelle):

Das Programm nimmt die MYC Befehle über die Tastatur oder Ethernet Schnittstelle über Telnet entgegen und gibt die CIV Befehle über den USB Port aus. Der Transceiver wird direkt an die USB Schnittstelle oder über einen USB -> RS232Wandler → CIV Pegelwandler an die Remote Buchse angeschlossen.

Hardware:

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.

Der Transceiver wird an der Remote Buchse angeschlossen.

Hinweise zur Bedienung

Die Befehle sind meist eine direkte Übersetzung der CIV Befehle. Daraus ergeben sich Einschränkungen bei der Bedienung:

Einige Befehle arbeiten abhängig von den aktuellen Einstellungen: Dies ist für in den Regeln abgebildet, und die Befehle wirken dann auch nicht (soweit ich das erkannt habe :)).

Beim Start des Interfaces muss das Radio an sein und mit dem Interface verbunden sein!

Die Befehlseingabe erfolgt binär. Der Wertebereich der Befehlsparameter ist bei den Befehlsbeschreibungen angegeben.

Softwareversion:

Zum Betrieb unter Windows muss Python 3.6 installiert werden.

Der Aufruf des Programms erfolgt dann in der Powershell (Die Verzeichnisnamen müssen ggf angepasst werden):

```
cd c:\Users\xx\Desktop\IC7300
```

```
>>> C:\Users\xx\AppData\Local\Programs\Python\Python36-32\python.exe Ic7300_interface.py
```

Hardwareversion:

Das Interface unterstützt keine Kollisionserkennung auf dem ICOM Bus. Daher darf nur 1 Radio an den CIV-Bus angeschlossen werden. Während des Remote Betriebs sollten daher auch keine manuellen Einstellungen am Gerät vorgenommen werden. Die CIV Schnittstelle verwendet 19200 Baud 8N1.

Einbindung in das MYC System

Details zum MYC System stehen in [3].

Die Befehlslisten sind für die Softwareversion und Hardwareversion leicht unterschiedlich und stehen bei den Programmen in v_announcelist..py beziehungsweise _announcements.bas der einzelnen Transceiver.

Fehlermeldungen

Der Befehl &HFFFC 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 (Hardwareversion)

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 &HFFFC "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 Softwareversion wurde mit Pycharm 18.03 entwickelt [1]

Bei der Hardwareversion übernimmt die Steuerung ein AVR Mikrocontroller Atmega1284

Die Software dazu wurde in BASCOM geschrieben [8]

Programmierung des Prozessors (Hardwareversion)

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 (Hardwareversion)

Schnittstellenparameter: 19k2 8N1

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 und RS232 / USB können nicht gleichzeitig verwendet. Der Befehlspeicher wird gelöscht, wenn die Schnittstelle gewechselt wird.

I2C Schnittstelle (Hardwareversion)

Die Default Adresse ist 23 (&H17)

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 ein reset gestartet. Einige I2C-Master setzen nach der Datenübertragung den Bus nicht entsprechend der Spezifikation auf „Idle“ zurück, was zu regelmäßigen resets führt.

SMD (Hardwareversion)

Die Leiterplatte ist teilweise mit SMD bestückt.

Stromversorgung (Hardwareversion)

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

Alternativ erfolgt die Stromversorgung über USB

Bestückung der Leiterplatte (Hardwareversion)

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!

SL3 wird nicht bestückt.

Anschlüsse (Hardwareversion)

Power

Tip 12V

Ring GND

RS232 (Buchse)

5 GND

2 TX (PC Eingang)

3 RX (PC Ausgang)

I2C

1 GND

2 SCL

3 SDA

Jumper (Hardwareversion)

JP1 Power

JP2 I2C: 3V/5V Umschaltung

JP3 SDA Überbrückung (ohne IC3)

JP4 SCL Überbrückung (ohne IC3)

JP5 Reset

JP6 ISP

JP7 Anschluss für USB Modul

Details zu den verschiedenen Transceivern

Dieses Programm wurde ursprünglich für den IC7300 geschrieben. Die Versionen für die anderen Transceiver wurden daraus abgeleitet.

IC7300

Das Interface verwendet die Befehle der Version des Handbuchs A7292-4EX-6a (2018). Es gibt leichte Änderungen gegenüber älteren Versionen. Daher sollte das Radio auch die neueste Softwareversion haben.

Das Handbuch enthält einige Fehler und Ungenauigkeiten. Details dazu und Abweichungen dazu bei den MYC Befehlen stehen in [7]

Diese Version wurde getestet.

Der Befehl 798 / Ausgabe einer Sprachdatei (IC7300) funktioniert nur, wenn eine entsprechende Karte mit Dateien vorhanden ist. Dies ist durch Regeln mit darstellbar, gibt aber einen CIV Fehler.

Softwareversion:

Bei Betrieb über die USB Schnittstelle muss Menu → Set → Connectors → CI-V → CI-V USB Port auf „Unlink from [REMOTE]“ gesetzt werden.

Weiterhin muss Menu → set → Connectors → USB Serial Function auf CI-V stehen.

Diese Einstellung ist per remote nur lesbar.

Als USB Baudrate muss AUTO oder 19200 oder kleiner gewählt werden. Es ist nicht geklärt, warum höhere Baudraten nicht funktionieren

Ändert man das im Betrieb, ist die USB Schnittstelle nicht mehr verwendbar.

Die CIV Adresse sollte auf 0x94 stehen. Bei anderer Adresse muss die Konfigurationsdatei geändert werden.

Bei der Initialisierung wird der VFO Mode eingeschaltet, USB echo abgeschaltet und alle leeren Memoryplätze beschrieben.

Hardwareversion:

Es muss Menu → Set → Connectors → CI-V → CI-V USB Port auf „Link to [REMOTE]“ gesetzt werden.

Außerdem sollte Menu → Set → Connectors → CI-V CIV transceive off sein

IC7000

Grundlage für die Befehlsliste ist das Benutzerhandbuch A-6478H-1EX. Die Beschreibung der Datenformate einiger Befehle fehlt. Sie wurden wie bei IC7300 gelassen. Es ist teilweise nicht beschrieben, welche Befehle auch als Lesebefehle gelten. Auch hier wurde es wie beim IC7300 gelassen. Die Befehlscodes 06 bis 09 auf Seite 146 sind teilweise doppelt und wahrscheinlich falsch. Sie wurden als 1A06 .. programmiert.

Diese Version wurde nicht getestet, da die CIV Schnittstelle meines Transceivers defekt ist.

ICx71

Grundlage für die Befehlsliste ist eine Liste von DF4OR (Tnx!)

Da die Parameter unterschiedlich sind, gibt es für jeden Tnsceiver eine Firmware.

Diese Firmware wurde noch nicht getestet, ein IC1271 ist aber noch vorhanden.

IC9700:

Grundlage für die Befehlsliste ist das CIV Reference Guide von 2020 (ohne Nummer).

Auch hier gibt es einige Fehler und Ungereimtheiten.

Da ich diesen Tranceiver nicht besitze, wurde das Programm nicht getestet.

Versionen

Diese Beschreibung gilt für die

Leiterplattenversion V03.2

Bascom Version V01.2 für alle Transceiver

Softwareversion (Python) V01.0

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 Gefahr; es wird keinerlei Garantie / Gewährleistung / Produkthaftung 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/ic7300_interface_python.zip
- [2] https://www.dk1ri.de/dhw/MYC_icom_eagle.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/myc/common_1.8.zip
- [7] https://dk1ri.de/dhw/7300_bugs.txt
- [8] https://dk1ri.de/dhw/IC7300_interface_bascom.zip
- [9] https://dk1ri.de/dhw/IC7000_interface_bascom.zip
- [10] https://dk1ri.de/dhw/IC9700_interface_bascom.zip
- [11] https://dk1ri.de/dhw/IC1271_interface_bascom.zip
- [12] https://dk1ri.de/dhw/IC471_interface_bascom.zip
- [13] https://dk1ri.de/dhw/IC271_interface_bascom.zip