Hinweis für die Textformat-Version; um diese Version im PDF Format zu lesen, laden Sie bitte https://www.dk1ri.de/dhw/cr_test.pdf

# Commandrouter / Logicdevice Tests für das MYC Protokoll

Author DK1RI, Version V02.1, 20260530
This paper is published in https://github.com/dk1ri as well

## Definitionen

https://dk1ri.de/myc/Definitions .txt oder https://dk1ri.de/myc/Definitions .pdf (englisch)

## Einleitung

Diese zwei Geräte (FU) sind zum Test des Commandrouters mit Logicdevice ab Version V02.03.00 (CR) vorgesehen. Die Steuerung kann nur mit USB erfolgen. Weitere Schnittstellen sind mit Hardwareergänzungen möglich.
Zur Steuerung mit einem Browser: https://dk1ri.de/myc/webserver.pdf oder https://dk1ri.de/myc/webserver.txt (wird aber nicht empfohlen).
Die Tests müssen manuell durchgeführt werden und das Verständnis der Syntax der announcements ist Voraussetzung. Die Test sind nicht „100%“, dürften aber die meisten Funktionen abdecken. Die Firmware der FUs kann auch bei Bedarf ergänzt werden.

## Beschreibung und Bedienung

Zunächst muss das MYC System installiert werden: CR: Beschreibung: https://dk1ri.de/myc/commandrouter.pdf oder https://dk1ri.de/myc/commandrouter.txt oder das Programm direkt:https://dk1ri.de/myc/commandrouter.zip.

In die Datei commandrouter_config/connection_of_devices müssen die COM Schnittstellen der beiden Testdevices eingetragen werden. Eine Beschreibung dazu steht dort.

*Bedienung mit Terminalprogramm (Console oder Hterm oder ähnliches)*
*Diese Bedienung setzt voraus, dass man das MYC System kennt und weiß wie die announcements in Befehle umzusetzen sind!*
Nach dem Aufruf des CR mit commandrouter.py kopiert der die announcefiles der beiden FUs.
Da der CR neue Befehlsnummern erzeugt, muss man die Befehle in check_output/v_announcelist_full_240 nachlesen oder den „0xF0<n>,01“ Befehl verwenden.

Wenn der Commandrouter läuft kann man Befehle im HEX Format (0 -9, A-F) eingeben.
Bei falschen Befehlen antwortet das device mit „not ok“ bei blockierten mit „not allowed“.
Falsche Befehle soll aber der CR ohne Fehlermeldung blockieren blockierte Befehle mit „blocked“ beantworten (testmode = 1 in der CR Konfiguration). Der CR funktioniert,richtig, wenn die FUs keine Fehlermeldungen liefern.
Wenn bei einem Befehl Daten fehlen, gibt es die Fehlermeldung „watchdog“.
Die Rules sind so aufgebaut, dass beim Start kein Befehl blockiert ist.
Nach dem Test einer Blockierung müssen wieder alle Blockierungen abgeschaltet werden.

*Mit Browser*
Die Bedienung mit dem Browser ist weitestgehend selbsterklärend.
Der Browser zeigt nicht alle Meldungen des Commandrouters, insbesondere sieht man nicht, wenn

ein Command geblockt ist. Man sieht aber Fehlermeldung im Terminalfenster. Ein weiterer Nachteil ist, dass der Browser nur „richtige“ Daten liefert; man sieht also nicht das Verhalten bei fehlerhaften Eingaben.

**Random Test**

Zusätzlich gibt es ein Python Testprogramm, das zufällige Daten an die File Schnittstelle (FU oder SK) des CR schickt. Dazu muss der CR laufen und in einem weiteren Terminalfenster läuft test_random.py sk oder test_random.py dev.
test_random.py dev.verhält sich beim Start wie eine FU: Beim ersten Start wird die announce-Datei geladen. Das funktioniert manchmal nicht richtig. Das hängt wahrscheinlich mit dem Windows Task switching zusammen. Das Programm, das Daten empfängt, muss manchmal länger werten, bis der Sender die Daten geschrieben hat, manchmal fehlt auch eine Datei.
Danach sieht man im Terminalfenster des CR – mit Testmode = 1 – viele Fehler, da der CR die meisten Eingaben als falsch erkennt und blockiert. Das Programm darf aber nicht abstürzen.
Das Programm test_random.py macht zusätzlich eine Messung des Befehlsdurchsatzes.
Das Ergebnis sollte auf einem einigermaßen schnellen Rechner 1000 Commands in 5 – 10 Sekunden sein.

**Einbindung in das MYC System**

Details zum MYC System stehen in https://www.dk1ri.de/myc/MYC.pdf .
Die komplette Befehlsliste steht als announcements in der Datei announcements.bas im Bascom Programm.

**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.

**Hardware Reset**

Ist der Reset Jumper JP5 beim Anlegen der Versorgungsspannung überbrückt, werden wieder die Defaultwerte eingelesen.

**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 „Testbetrieb“ nach ca 10 Sekunden ansprechen. Für den „nicht Test 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.

**Firmware**

Die Steuerung übernimmt ein AVR Mikrocontroller Atmega1284.

Die Software wurde in BASCOM geschrieben https://www.dk1ri.de/dhw/cr_test_bascom.zip .
Dort gibt es einen Hex Datei zur Programmierung des ATMEGA
Um das Programm zu kompilieren, muss das Verzeichnis common_V01.15 in das Verzeichnis mit dem Programm kopiert werden. Die aktuelle gültige Version steht im Bascom Programm.

**Programmierung des Prozessors**

Zur Programmierung des Prozessors ist ein 6poliger ISP Stecker JP6 vorgesehen.
Die Fuses müssen möglicherweise programmiert werden (JTAG abschalten!!). Prozessortyp und Frequenz müssen gegebenenfalls angepasst werden.
Der Jumper JP1 und sollte während der Programmierung entfernt werden.

**Serielle (RS232 / USB) Schnittstelle**

Es 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 6 pins des Verbinders ST2 sind mit den 6 Anschlusspunkten JP7 / JP8 auf dem Interface zu verbinden. USB Platine und Interface müssen voneinander isoliert werden.
Die Stromversorgung kann dann über USB erfolgen.

**I2C Schnittstelle**

ist nicht vorgesehen

**SMD**

Die Leiterplatte ist teilweise mit SMD bestückt

**Stromversorgung**

Die Stromversorgung erfolgt über USB.

**Bestückung der Leiterplatte**

Es ist nur die Minimalbestückung mit USB Schnittstelle vorgesehen
Die Leiterplatte ist für verschiedene Konfigurationen ausgelegt und wird daher daher nur teilweise bestückt. Für genau diese Anwendung kann die Leiterplatte verkleinert werden.

Verwendung von ISP:
JP6

Außer den bereits erwähnten Bauteilen wird immer bestückt: Q1, IC1, U1, C2 – C5, C10 - C12, JP1, JP5,

**Anschlüsse**
-

**Jumper**

JP5 Reset

| | |
|---|---|
| JP6 | ISP |
| JP7/8 | Anschluss für USB Modul |

**Versionen**

Diese Beschreibung gilt für die Leiterplattenversion (wireless interface) V05.0
https://www.dk1ri.de/dhw/Wireless_interface_eagle.zip
Bascom Version V01.0 https://www.dk1ri.de/dhw/cr_test_bascom.zip .

**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.