

Externes Zubehör / Adapter

- USB -> Floppy Drive Adapter

USB -> Floppy Drive Adapter

Beschreibung

Möchte man 3,5" Zoll Disketten lesen und schreiben am heutigen Computer. Wird die Auswahl eng. Während es für 1,44 MB Disketten USB Laufwerke gibt sieht es bei 5,25" Zoll schlecht aus. Aber es gibt Adapter um ein C64 Floppy Drive anzuschließen.

Vorraustzungen

C64 Laufwerk mit DIN Anschluss



Din Anschluss Anschlusskabel :



USB Adapter



Zu kaufen bei : <https://www.ebay.de/itm/164239154127>

Installation unter Linux

Abhängigkeiten installieren

```
sudo apt-get install libusb-dev build-essential linux-headers-generic git libncurses-dev
```

Die eigentlichen Programme installieren

```
cd ~
git clone https://github.com/cc65/cc65.git
cd cc65
make

Ausgabe:
gcc -O3 -Wall -Wextra -Wno-char-subscripts -o ataricvt ataricvt.c
gcc -O3 -Wall -Wextra -Wno-char-subscripts -o gamate-fixcart gamate-fixcart.c
make[2]: Für das Ziel „zlib“ ist nichts zu tun.
make[1]: Für das Ziel „all“ ist nichts zu tun.
Diese Ausgabe ist okay

sudo PREFIX=/usr make install
```

Nun das CBM Paket downloaden.

<https://sourceforge.net/projects/opencbm/files/opencbm/opencbm-0.4.99.99/opencbm-0.4.99.99.tar.bz2/download>

Nun ins Downloadverzeichnis gehen

```
cd ~/Downloads
tar -xf opencbm-0.4.99.99.tar.bz2

#ggf falls versionsnummer abändern
cd ~/opencbm-0.4.99.99
make -f LINUX/Makefile opencbm plugin-xum1541
sudo make -f LINUX/Makefile install install-plugin-xum1541
sudo ln -s /usr/local/lib/libopencbm.so.0 /usr/lib/libopencbm.so.0
```

Testen

Mit dem Befehl testen wir die Verbindung

```
sudo cbmctrl detect
```

Ausgabe, die Nummer ist die Laufwerksnummer mit der wir das Laufwerk ansprechen:

```
8: SpeedDOS 1541
```

Bedienung

Parameter

```
sudo cbmctrl <Befehl> <Laufwerksnummer>
```

Befehle

```
dir    output the directory of the disk in the specified drive
change wait for a disk to be changed in the specified drive
detect detect all drives on the IEC bus
```

Eine Disk kopieren

```
d64copy <quelle> <ziel>
```

Eine Reale Disk auf den Computer (8 ist die ID des Laufwerks, herauszubekommen mit

```
sudo cbmctrl detect
```

```
d64copy 8 mein64image.img
```

Ein Disk image auf eine reale Disk kopieren

```
d64copy mein64image.img 8
```

Ein Programm von einer Disk kopieren

Dateien anzeigen

```
sudo cbmctrl dir 8
```

Ausgabe:

```
0 ."SPIELE2      " fp 2a
1  ", "         seq
```

```
65 "boulderdash"   prg
66 "boulder dash ii" prg
64 "montezuma"     prg
34 "clowns"        prg
67 "olympic skier" prg
345 blocks free.
00, ok,00,00
```

Wir wollen als beispiel boulderdash.prg downloaden

Befehl:

```
sudo cbmread 8 boulderdash
```

Ausgabe

```
[Info] reading BOULDERDASH -> boulderdash.prg
[Info] identified a SpeedDOS 1541 drive
.....\[Info] 00, OK,00,00
```

Ein Programm auf eine Disk kopieren

Falls die Dateienung .prg noch dran ist diese entfernen. Denn diese wir beim upload wieder angefügt

```
mv boulderdash.prg boulderdash
```

```
sudo cbmwrite 8 boulderdash
```

Ausgabe

```
[Info] writing boulderdash -> BOULDERDASH.PRG,P,W
[Info] identified a SpeedDOS 1541 drive
.....|
[Info] 00, OK,00,00
```

GUI-Tool

Download GUI tool von <https://c64.mvgrafx.net/qtcbm/LinuxQtCBM.zip>

Entpacken , ausführbar machen, starten

```
unzip LinuxQtCBM.zip
```

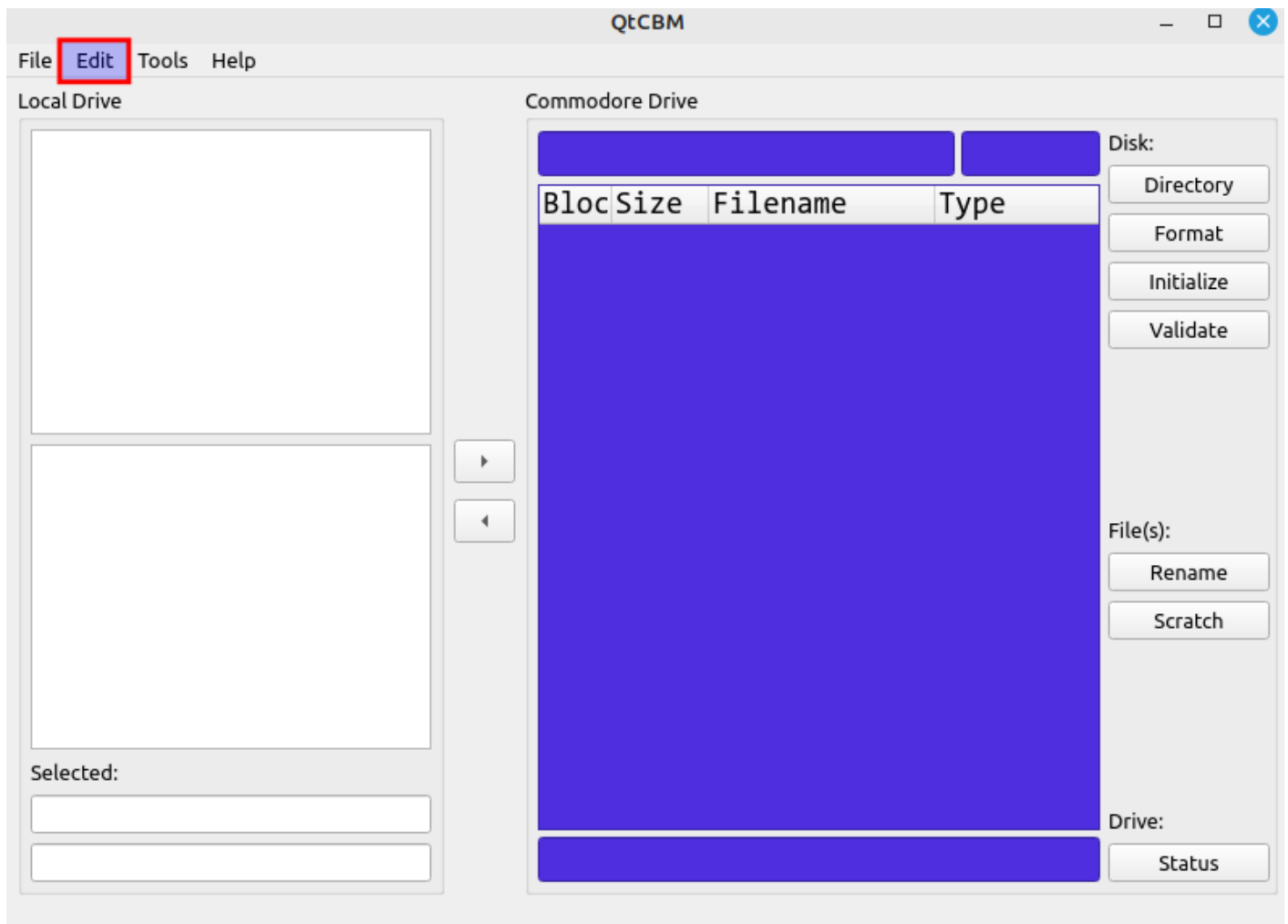
```
chmod +x QtCBM
```

Starten mit

```
sudo ./QtCBM
```

Nachdem das Programm gestartet ist unter

Edit -> Preferences gehen



Preferences...

Nun diese Pfade eintragen:

```
cbmctrl : /usr/local/bin/cbmctrl
```

```
cbmfornng : /usr/local/bin/cbmfornng
```

```
d64copy : /usr/local/bin/d64copy
```

```
cbmcopy : /usr/local/bin/cbmcopy
```

Haken bei Preview cbm command

und auf Save drücken

Application Preferences

File Locations

cbmctrl	/usr/local/bin/cbmctrl	Browse...
cbmforng	/usr/local/bin/cbmforng	Browse...
d64copy	/usr/local/bin/d64copy	Browse...
cbmcopy	/usr/local/bin/cbmcopy	Browse...
morse		Browse...

Other Settings

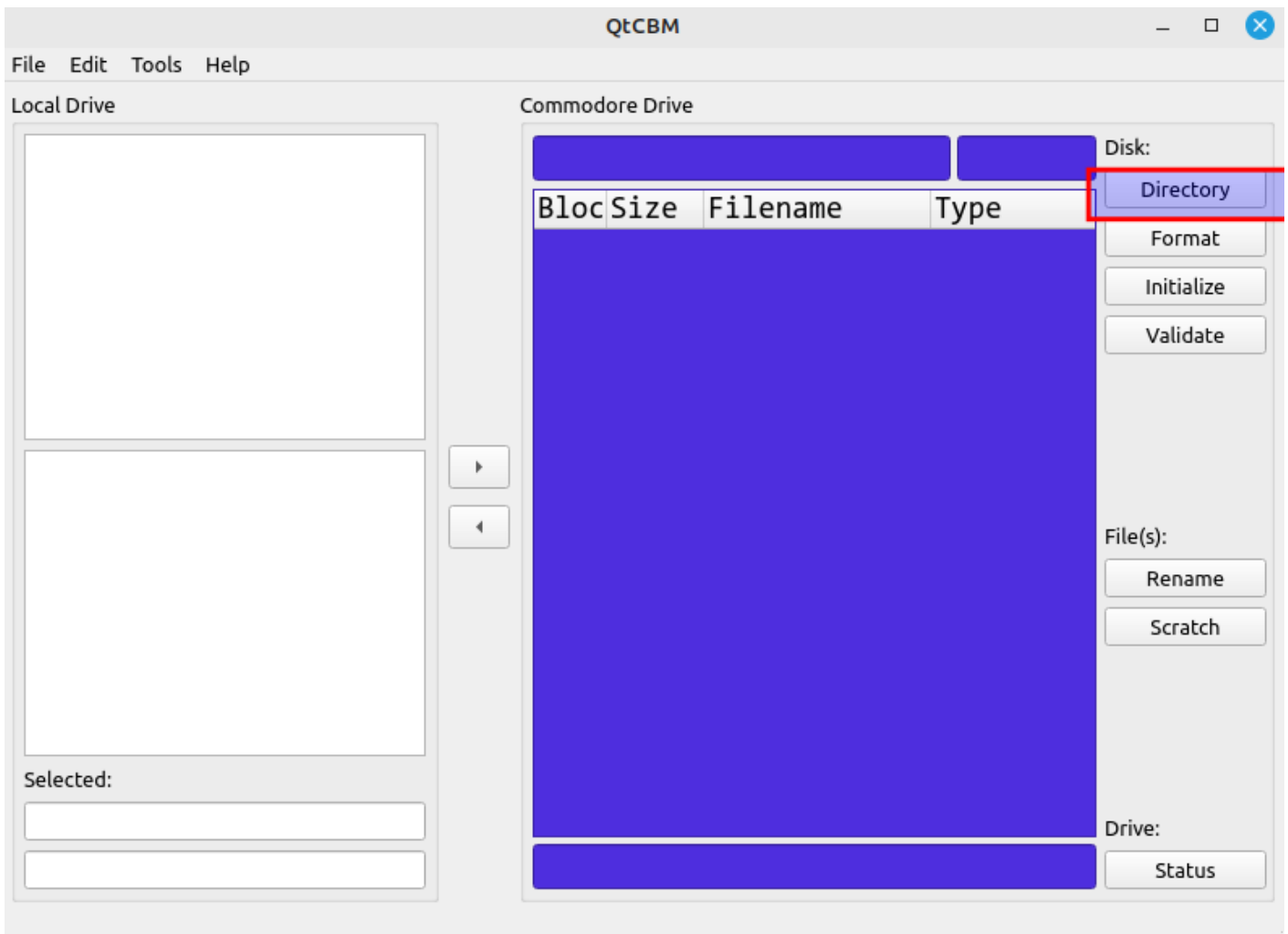
Device ID: 8

Transfer Mode:
 Original (Very Slow)
 Serial 1 (Slow)
 Serial 2
 Parallel
 Auto (Recommended)

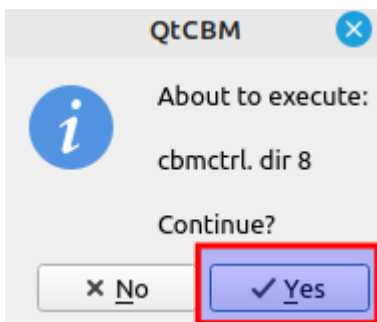
Preview CBM commands
 Automatically refresh after write operation
 Use classic font in CBM output

× Cancel **Save**

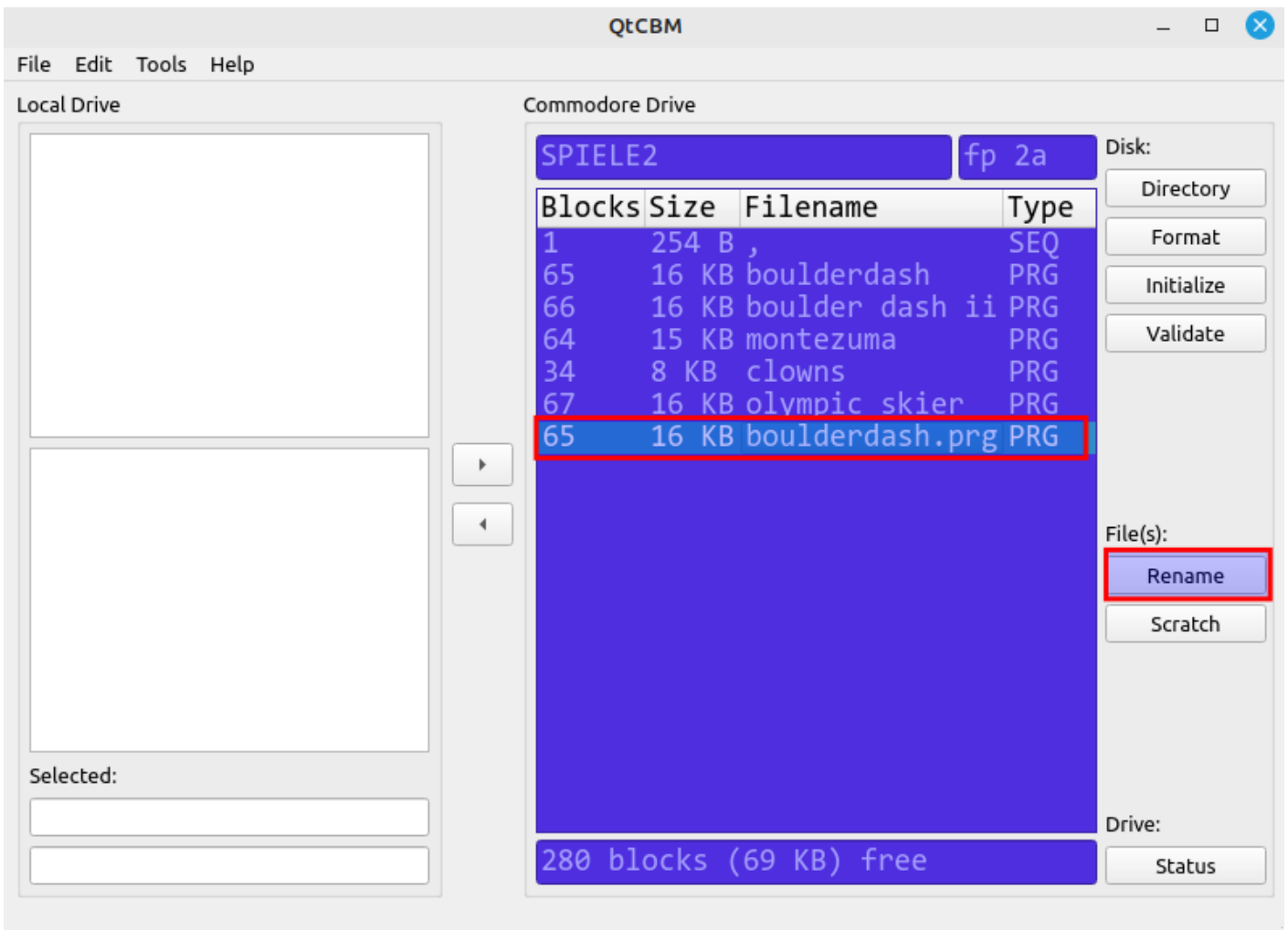
Nun mit Dir das Verzeichnis auflisten



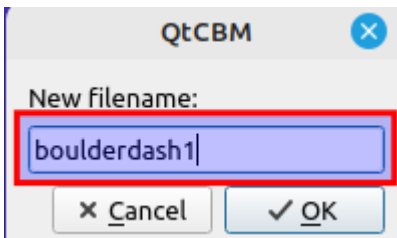
Mit Yes bestätigen



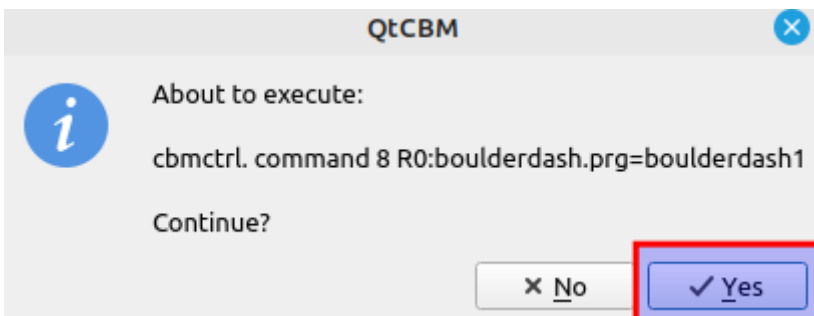
Die Auflistung, nun können wir eine Datei umbenennen, dazu die Datei anklicken und auf rename klicken



Neuen namen eingeben und OK klicken



Nun die Frage wieder mit yes beantworten



Kleines Schellscripte dei das Leben vereinfachen sollen

Script um alle Prgramme von einem Laufwerk in einem Zielverzeichnis zu speichern

```
#!/bin/bash
kopiereSpiele() {
    local OUTPUT="$1"
    local ZIELVERZEICHNIS="$2"
    #Ausgabe der Dateiliste auf den Monitor
    echo "$OUTPUT"
    # Überprüfen, ob der Diskettenname vorhanden ist
    if echo "$OUTPUT" | grep -q "2a"; then
        # Den Namen der Diskette aus der ersten Zeile extrahieren
        DISKETTENNAME=$(echo "$OUTPUT" | awk -F'"' 'NR==1 {print $2}' | tr -d ' ')
    else
        # Wenn kein Diskettenname vorhanden ist, einen Zeitstempel verwenden
        DISKETTENNAME=$(date +"%Y-%m-%d_%H-%M-%S")
    fi

    # Überprüfen, ob das Verzeichnis bereits existiert und ggf. einen Suffix hinzufügen
    SUFFIX=0
    ORIGINAL_DISKETTENNAME="$DISKETTENNAME"
    while [[ -d "$ZIELVERZEICHNIS/$DISKETTENNAME" ]]; do
        SUFFIX=$((SUFFIX+1))
        DISKETTENNAME="${ORIGINAL_DISKETTENNAME}-${SUFFIX}"
    done

    # Den Namen des Diskette ausgeben
    echo "Kopiere Diskette: $DISKETTENNAME"
    # Ein Verzeichnis für die Diskette anlegen
    mkdir -p "$ZIELVERZEICHNIS/$DISKETTENNAME"

    # Jede Zeile des Outputs durchgehen
    echo "$OUTPUT" | while IFS= read -r line; do
        # Überprüfen, ob die Zeile ein PRG enthält
```

```

if echo "$line" | grep -q "prg"; then
    # Den Programmnamen extrahieren
    ORIGINAL_PROGRAMMNAME=$(echo "$line" | awk -F'"' '{print $2}')
    SAFE_PROGRAMMNAME=$(echo "$ORIGINAL_PROGRAMMNAME" | tr '\\' '--')

    # In das Zielverzeichnis wechseln
    pushd "$ZIELVERZEICHNIS/$DISKETTENNAME" > /dev/null

    # Das Programm mit cbmread speichern
    cbmread 8 "$ORIGINAL_PROGRAMMNAME" > "$SAFE_PROGRAMMNAME.prg"

    # Zurück zum ursprünglichen Verzeichnis
    popd > /dev/null
fi
done
}

# Überprüfen, ob die notwendigen Parameter angegeben wurden
if [ -z "$1" ] || [ -z "$2" ]; then
    echo "Usage: $0 -l <laufwerksnummer> -d <zielverzeichnis>"
    exit 1
fi

#reset bus
cbmctrl reset

while getopts ":l:d:" opt; do
    case $opt in
        l) LAUFWERKSNUMMER="$OPTARG";;
        d) ZIELVERZEICHNIS="$OPTARG";;
        \?) echo "Invalid option -$OPTARG" >&2; exit 1;;
        :) echo "Option -$OPTARG requires an argument." >&2; exit 1;;
    esac
done

# Endlosschleife
while true; do
    # Dateiliste von cbmctrl abrufen
    OUTPUT=$(cbmctrl dir $LAUFWERKSNUMMER)

```

```

# Spiele kopieren
kopiereSpiele "$OUTPUT" "$ZIELVERZEICHNIS"

# Den Benutzer fragen, ob er weitermachen oder beenden möchte
echo "1. Nächste Diskette drücken Sie 1"
echo "2. Beenden drücken Sie 2"
read -p "Wählen Sie eine Option: " OPTION

if [ "$OPTION" == "2" ]; then
    break
fi
if [ "$OPTION" == "1" ]; then
    cbmctrl change $LAUFWERKSNUMMER
fi

done

cbmctrl reset
echo "Script beendet."

```

Fehlerbehebung:

Drive not Ready

Sollte das Drive den Fehler bei folgendem Befehl schmeißen:

```
74,drive not Ready,00,00
```

Software Fehler : Laufwerk wurde nicht geschlossen. Sollte Diskette gewechselt werden change Befehl benutzen.

Sollte Fehler bestehen bleiben, nochmals versuchen.

Befehl für Reset:

```
cbmctrl reset
```

Befehl für change :

```
cbmctrl change laufwerknummer
```

Laufwerknummer in der Regel

```
cbmctrl change 8
```

Floppy Image converter

Unter : <https://style64.org/release/dirmaster-v3.1.5-style> gibt es einen Manager der dank Wine auch unter Linux funktioniert.

Damit kann man prg aus T64 Files exportieren und noch viel mehr.

