# **Telegram CLI auf Eingebetteten Systemen**

### Definition Eingebettete Systeme

Bei mir kommen die beiden Plattformen Raspberry Pi in unterschiedlichen Versionen und das System Cubietruck (aka. Cubieboard 3) zum Einsatz.

### Quellen:

- Telegram Homepage
- GitHub Project Homepage for Telegram CLI
- Patch, siehe unten
- Telegram messenger CLI (getestet 04.2022)
- How to send yourself a Telegram message from BASH

### Raspberry Pi B, Rev. 1

### Vorbereitungen

Die folgenden Installationsschritte habe ich auf einem Raspberry Pi B Rev. 2 mit einem Raspbian GNU/Linux 9 (Stretch) durchgeführt.

```
sudo su
# notwendige Tools installieren
apt-get update
apt-get upgrade
apt-get install git libreadline-dev libconfig-dev libssl-dev lua5.2
liblua5.2-dev libevent-dev libjansson-dev libpython-dev make libssl1.0-dev
```

### Sourcen von Git kopieren

### git clone --recursive https://github.com/vysheng/tg.git

# In der Datei tgl/mtproto-utils.c, die Zeilen 101 und 115 auskommentieren. Quelle [3] # assert (0); // As long as nobody ever uses this code, assume it is broken. # by # //assert (0); // As long as nobody ever uses this code, assume it is broken. # (or delete the lines) and run make again.

### **Telegram CLI kompilieren**

cd tg

# ./configure make

Im Unterverzeichnis ./bin liegt dann das ausführbare Binary telegram-cli.

### Cubietruck aka. Cubieboard 3

Cubietruck Informationen:

- https://www.cubietruck.com/
- http://docs.cubieboard.org/
- https://linux-sunxi.org/Cubietruck

#### Vorbereitungen

Die folgenden Installationsschritte habe ich auf einem Cubietruck aka. Cubieboard 3 mit einem Debian GNU/Linux 8 (jessie) durchgeführt.

sudo su
# notwendige Tools installieren
apt-get update
apt-get upgrade
apt-get install git libreadline-dev libconfig-dev libssl-dev lua5.2
liblua5.2-dev libevent-dev libjansson-dev libpython-dev make libssl-dev

### Sourcen von Git kopieren

```
git clone --recursive https://github.com/vysheng/tg.git
```

# In der Datei tgl/mtproto-utils.c, die Zeilen 101 und 115 auskommentieren. Quelle [3] # assert (0); // As long as nobody ever uses this code, assume it is broken. # by # //assert (0); // As long as nobody ever uses this code, assume it is broken. # (or delete the lines) and run make again.

### Telegram CLI kompilieren

cd tg
./configure
make

Im Unterverzeichnis ./bin liegt dann das ausführbare Binary telegram-cli.

Christophs DokuWiki - https://von-thuelen.de/

## Bedienung (allgemein)

```
Telegram im interaktiven Modus starten:
bin/telegram-cli -k tg-server.pub
# Kontaktliste ausgeben:
./bin/telegram-cli -k tg-server.pub -e "contact_list"
# eine Textnachricht versenden:
./bin/telegram-cli -k tg-server.pub -W -e "msg <EMPFÄNGER> \"Nachricht mit
Leerzeichen\""
```

### Installation

```
cd tg
cp ./bin/telegram-cli /bin/
mkdir /etc/telegram-cli/
cp tg/tg-server.pub /etc/telegram-cli/
telegram-cli -W -e "msg <EMPFÄNGER> \"Nachricht mit Leerzeichen\""
```

From: https://von-thuelen.de/ - **Christophs DokuWiki** 

Permanent link: https://von-thuelen.de/doku.php/wiki/linux/messaging/uebersicht

```
Last update: 2022/04/04 19:53
```

