

Die **trx-control** System Architektur

Den Kern von **trx-control** bildet `trxd(8)`, ein hocheffizienter Daemon-Prozess für Linux (oder Windows mit WSL). `trxd(8)` ist in der C Sprache geschrieben und nutzt einen modernen multi-thread Ansatz.

Treiber für die Transceiver- oder Gerätesteuerung hingegen sind in der Skript-Sprache Lua geschrieben, was es einfach macht, das Verhalten von `trxd(8)` zu ändern oder neue Funktionen hinzuzufügen. Weiter setzt das Hinzufügen neuer Treiber etc. keine Neukompilierung der Software voraus.

Jeder Kommunikationskanal (TCP/IP Socket, WebSocket) wird durch einen Thread beobachtet, der einen Service-Thread für jeden Client startet, der sich mit `trxd(8)` verbindet.

Jeder Client wird durch seinen eignen Service-Thread bedient und der Zugriff auf alle Ressourcen ist sorgfältig synchronisiert, so dass nie zwei Clients gleichzeitig auf die gleiche Ressource zugreifen.

Transceiver können automatisch in einem bestimmten Intervall für Status-Updates abgefragt werden (Polling) oder, falls sie selbst automatische Updates senden, kann dies auch ohne Polling-Thread verwendet werden.

Die Kommunikation wird nicht nur durch den Client, sondern auch durch `trxd(8)` angestossen, z.B. für Status-Updates.

Wieso `trxd(8)`?

Linux verfügt über ein umfangreiches Handbuch, das in nummerierte Kapitel eingeteilt ist.

Kapitel 8 des Handbuchs ist das "System Manager's Manual".

Es ist gebräuchlich, Kommandos mit der Kapitelnummer, in der sie dokumentiert sind, in Klammern zu versehen.

`trxd(8)` ist in Kapitel 8 des Handbuchs dokumentiert, siehe `$ man trxd`.

Die Lösung zur Shack-Automatisierung

trx-control ist ein modernes und erweiterbares Software-System, das die folgenden Anwendungsfälle abdeckt:

- Steuerung von Transceivern.
- Steuerung anderer Geräte, z.B. GPIO Adapter.
- Zu Datenbanken verbinden, z.B. PostgreSQL.
- Kommunikation zwischen verschiedenen Clients.
- Drittsysteme wie QRZ.com, DXCluster etc. abfragen.

Vorteile einer offenen Lösung

trx-control basiert auf einem völlig offenen Design und ist Open Source unter der sehr liberalen MIT Lizenz. Dank des modularen Aufbaus ist es leicht zu erweitern, das einzige Limit ist die Phantasie. `trxd(8)`, der Daemon-Prozess, der im Kern des Systems läuft, ist hocheffizient und durch sein multi-thread Ansatz nutzt er alle verfügbaren CPU-Kerne.

trx-control in existierende Software wie Logbücher oder Contest-Software zu integrieren, ist sehr einfach dank einem gut dokumentierten JSON basierten Client/Server-Protokoll, das über Sockets oder WebSockets benutzt werden kann.



trx-control website

micro systems
Radio Communications
Landstrasse 66
CH-5073 Gipf-Oberfrick

<https://trx-control.msys.ch>
info@hb9ssb.ch
[#trx-control:matrix.org](https://matrix.org)
062 871 45 65



Ein modernes Software-System für Linux zur Steuerung von Transceivern und anderen Geräten über das Netzwerk.

trx-control ist ein erweiterbares Software-System, um Amateurfunk Transceiver und andere Hardware wie GPIO-Pins etc. zu steuern sowie Drittsysteme durch anwendungsspezifische Erweiterungen anzubinden. Eine umfangreiche und komplette Dokumentation steht zur Verfügung.

Projekt-Ziele

- Eine Lösung aus einem Guss für die Shack-Automatisierung bereitstellen.
- Einen einfachen Weg finden, um neue Transceiver und andere Geräte hinzuzufügen.
- Das System in Lua erweiterbar machen, einer Skript-Sprache, die einfach zu erlernen und leicht zu benutzen ist.
- Ein klar definiertes Protokoll basierend auf dem Austausch von JSON-Datenpaketen zur Kommunikation zwischen **trx-control** und Clients einsetzen.
- Der Kern der Software basiert auf einem modernen asynchronen multi-thread Ansatz.

trx-control wird in der Schweiz hergestellt.



trx-control für Endanwender

Installation

trx-control kann ab Quellcode gebaut und installiert werden oder es können vorgefertigte Pakete genutzt werden.

Konfiguration

Für die Konfiguration muss nur die Datei `/etc/trxd.yaml` im gut verständlichen YAML Format angepasst werden.

Anwendung

Nachdem die `/etc/trxd.yaml` Konfigurationsdatei angepasst wurde, wird `trxd(8)` mit `systemctl` gestartet:

```
# systemctl start trxd
```

Lesen Sie auch die `trxd(8)`, `trxctl(1)` und `trx-control(7)` Handbuchseiten.

Das `trxctl(1)` Kommando

Um Ihre Installation zu testen oder Transceiver zu steuern, könne Sie das `trxctl(1)` Kommando verwenden, das sich mit `trxd(8)` verbindet und Ihnen erlaubt, Kommandos zu senden oder Status-Updates zu empfangen und anzuzeigen.

Sagen Sie es weiter

Wenn Ihnen **trx-control** gefällt, dann sagen Sie es weiter. Erzählen Sie von dem Projekt oder stellen Sie es in Ihrem Verein vor. Fragen Sie uns nach Präsentationen und Flyern.



User guide
<https://trx-control.msys.ch/trx-control-user-guide.html>

trx-control für System-Integratoren

Mit `trxd(8)` verbinden

Ihre Software verbindet sich mit `trxd(8)` über einen Socket oder WebSocket. Die Kommunikation ist bi-direktional.

Das Client/Server Protokoll

Das Client/Server Protokoll basiert auf dem Austausch JSON-formatierter Meldungen. Bei regulären Sockets muss dabei das NDJSON Format verwendet werden.

Das Protokoll ist wie die Software erweiterbar. Erweiterungen definieren ihre eigenen Anfragen und Antworten.

Anfrage-Ziele

Jedes Gerät und jede Erweiterung stellt ein Ziel dar, an das Anfragen gestellt werden können. Die Liste der Ziele kann abgefragt werden, die Namen der Ziele werden in der `/etc/trxd.yaml` Konfigurationsdatei festgelegt.

Automatische Status-Updates

Wenn Sie automatische Status-Updates einschalten oder auf Ereignisse von Erweiterungen warten, muss die Software jederzeit in der Lage sein, JSON Pakete zu empfangen.

Lassen Sie es uns wissen

Wenn Sie Unterstützung für **trx-control** zu Ihrer Software hinzufügen, dann lassen Sie es uns wissen. Wir fügen Ihr Projekt zu einer entsprechenden Liste hinzu.



Integration guide
<https://trx-control.msys.ch/trx-control-integration-guide.html>

trx-control für Entwickler

Übersicht

trx-control ist in der Lua Programmiersprache erweiterbar. Lua ist einfach zu erlernen und leicht zu benutzen, ist aber trotzdem sehr leistungsfähig und schnell. Siehe <https://lua.org> für Details.

Gerätetreiber

Gerätetreiber verbinden `trxd(8)` mit Hardware. Hier werden Anfragen vom standardisierten JSON Format in das gerätespezifische Format umgewandelt und umgekehrt werden die Antworten behandelt. So sind z.B. Frequenzen immer in Hertz, unabhängig vom Transceiver.

Erweiterungen

Erweiterungen fügen anwendungsspezifische Funktionen zu **trx-control** hinzu und machen diese so für Clients verfügbar. Sie definieren ihre eigenen Anfragen und Antworten.

Lua Module

trx-control kommt mit einer Reihe von Lua Modulen: `yaml`, `json`, `postgres`, `curl`, `sqlite`, `linux`, `expat`. Weitere, auch selbstgeschriebene, können leicht hinzugefügt werden.

Hilfe erhalten

Es gibt den Issue Tracker auf GitHub unter <https://github.com/hb9ssb/trx-control>. Sie können aber auch dem Matrix Raum `#trx-control:matrix.org` beitreten.



Developer guide
<https://trx-control.msys.ch/trx-control-developer-guide.html>