

Wenn Sie über Links in unseren Artikeln einkaufen, erhalten wir eine kleine Provision. Das hat weder Einfluss auf unsere redaktionelle Unabhängigkeit noch auf den Kaufpreis.

Home / How-To / Hardware

HOW-TO

Modellbahnsteuerung mit dem Raspberry Pi

Wie Sie eine automatische Modellbahnsteuerung mit günstigen Alternativen zu kommerziellen Produkten einrichten können, lesen Sie in diesem Beitrag.

Von Stephan Richter

PC-WELT | 16.10.2014 08:30 UHR



Bild: Fotolia

Stephan Richter ist derzeit Promotionsstudent in Bioinformatik an der Uni Jena in Zusammenarbeit mit dem Umweltforschungszentrum in Leipzig. Dort forscht er am biologischen Abbau von Umweltschadstoffen durch verschiedene Mikroorganismen. Sein Hobby sind Modelleisenbahnen. Eine steht noch bei seinen Eltern – und inzwischen auch zum Verkauf, weil ihm die Zeit fehlt, sich darum weiter zu kümmern. In seinem [Blog](#) können Sie mehr dazu lesen. Aber: Nach dem Besuch einer Modellbahn-Ausstellung im Frühjahr 2014 wollte er erneut eine eigene Anlage aufbauen.

diesem Beitrag beschreibt er, wie er die Aufgabe gelöst hat.

Eine im Bau befindliche Modellbahn will ich digital steuern. Die für die Automatisierung nötige Technik will ich möglichst günstig umsetzen. Konkret greife ich dabei an folgenden Punkten auf "selbstgemachte" Alternativen zurück.

- An Stelle einer teuren Digitalzentrale setze ich einen preiswerten Raspberry Pi ein
- Statt relativ teurer S88-Rückmelde-Encoder verwende ich billige Schieberegister
- Anstelle von herkömmlichen Weichenantrieben benutze ich KACO-Relais
- Auf einem Desktop-PC steuere ich die Modellbahn mittels RocView

Wenn Sie meiner Bauanleitung folgen wollen, brauchen Sie

- Lötkenntnisse
- einen Raspberry Pi
- einen SRCP-Client wie jman auf dem Rechner (SRCP = Simple Railroad Command Protocol) oder SRCPC auf dem Smartphone

Digitale Modellbahn und die Computersteuerung

Wenn man heutzutage eine Modellbahn aufbaut, kommt man ziemlich früh zu der Frage: Analog oder digital? Es gibt für beides gute Argumente.

Das Argument schlechthin für die digitale Modellbahn ist die Möglichkeit, auf ein und demselben Gleis mehrere Züge vollkommen unabhängig voneinander steuern zu können. Dazu muss ein kleines Stück Elektronik in jedem Triebfahrzeug sein: der Digitaldecoder. Dieses fingernagel- bis münzgroße elektronische Wunderwerk eröffnet unzählige Möglichkeiten. Zum Beispiel liegt es sehr nahe, die digitale Modellbahn auch gleich mit einem Computer zu steuern und zu automatisieren.

Gleichzeitig sind die Decoder oder vielmehr ihr Preis auch ein gutes Argument gegen die Digitaltechnik. Man muss ja jede Lok mit einem Decoder ausstatten. Und wenn die Signale und Weichen auch digital gestellt werden sollen, ist auch hier der Kauf vieler Decoder erforderlich. Eine Modellbahn ist somit nicht nur ein



Wenn man die Modellbahn mit dem PC steuern will, müssen dessen Anweisungen auch in die Sprache der Decoder übersetzt werden. Das geschieht im Regelfall mittels einer mehr oder weniger leistungsfähigen Digitalzentrale, die Befehle vom Computer entgegennimmt und das digitale Gleissignal erzeugt. Einfache Zentralen gibt es ab etwa 50 Euro, gängige Geräte mit PC-Schnittstelle kosten auch gern mal 100 Euro und mehr.

Doch es gibt eine billige Alternative: Man kann das Gleissignal auch direkt von einem Computer erzeugen lassen.

Das geht mit der Software SRCPD, einem Open-Source-Programm, das auf einer einfachen seriellen Schnittstelle das Steuersignal für Züge und Zubehör erzeugt. Dieses Programm bekommt die Befehle, die es umsetzen soll, über das sogenannte Simple Railroad Command Protocol – ein offener Standard für die Kommunikation von Modellbahnbefehlen. Damit kann man dann sein eigenes Programm schreiben oder Modellbahnsteuerungs-Programme wie RocRail die Anlage steuern lassen.

Haken dabei: Aktuelle PCs werden kaum noch mit seriellen Schnittstellen ausgestattet.

Die Lösung für dieses Problem bringt der Raspberry Pi. Dabei handelt es sich um eine kreditkartengroße Platine, die neben einem Prozessor, Arbeitsspeicher, einem Grafikchip und einer Soundkarte auch noch jede Menge Anschlussmöglichkeiten bietet. Obwohl es sich im Prinzip um einen fast vollwertigen PC handelt, kostet ein Raspberry Pi nur etwa 35 Euro. Ihn kann man nutzen, um das Digitalsignal mittels SRCPD zu erzeugen.

Dazu ist folgendes vorzubereiten:

- Der Raspberry muss mit dem Booster der Modellbahnanlage verbunden werden
- Die Software muss auf dem Raspberry installiert werden
- Für eine Automatisierung der Modellbahnanlage sind Rückmeldekontakte anzuschließen

Damit das Signal aus dem Raspberry in den Booster gefüttert werden kann, müssen wir dessen Spannung noch vom Logikpegel des Rasperrys auf das Level des Boosters umsetzen. Das geschieht mit einem sogenannten Pegelumsetzer, einem kleinen integrierten Schaltkreis, der aus dem 3-V-Logiklevel des PCs ein 5-V-Signal für den Booster macht. Der Chip hört auf den Namen Max232ECPE und übernimmt diese Aufgabe für uns. Er wird wie abgebildet mit einem passenden Stecker an die GPIO-Schnittstelle des Rasperrys angeschlossen.

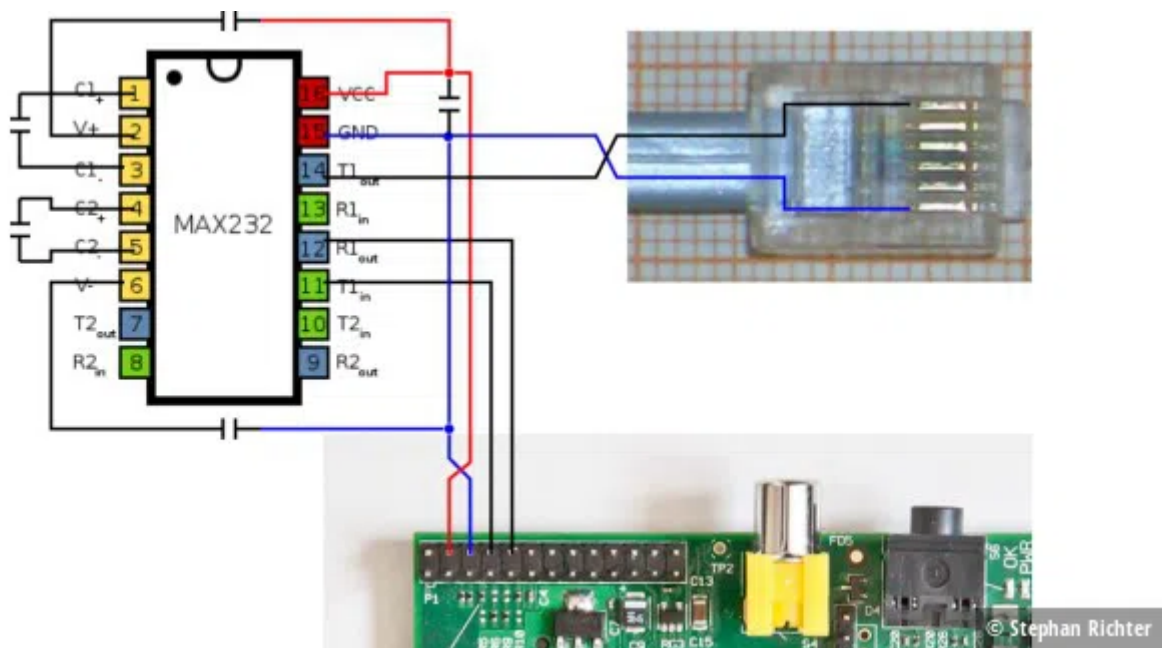
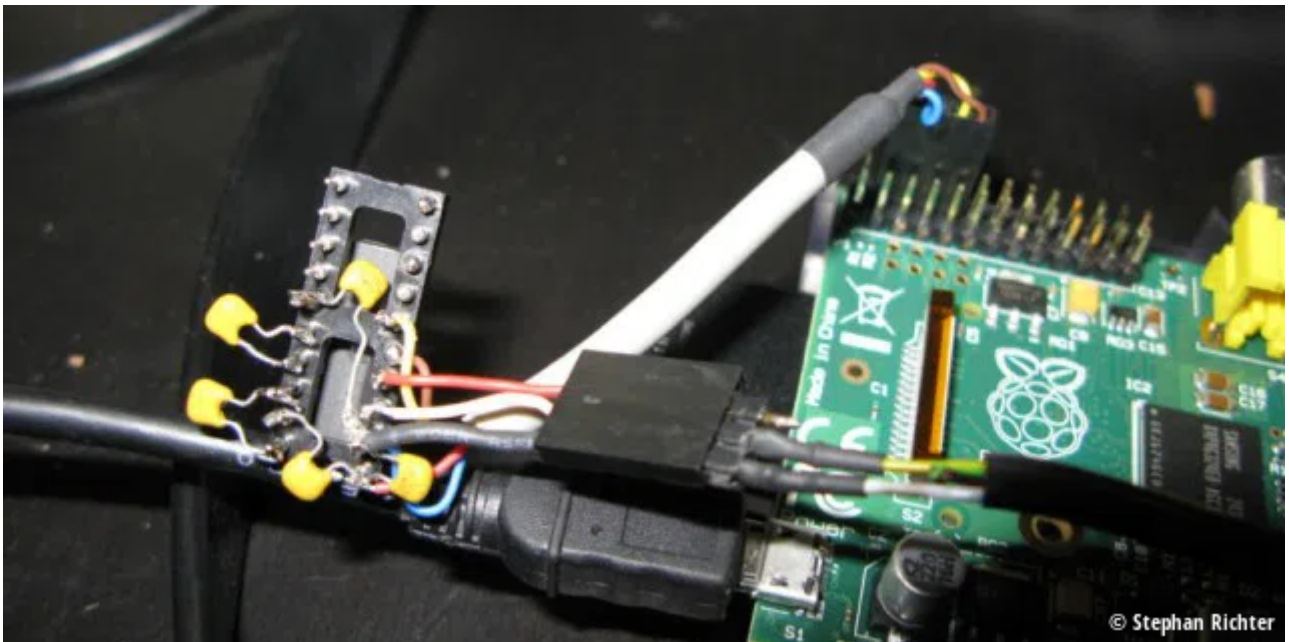


Bild: ©Stephan Richter

Booster, der über einen Westernstecker verbunden ist. Hierfür habe ich ebenfalls eines der Kabel aus der Startpackung verwendet, mit dem normalerweise die Lokmäuse mit dem Booster verbunden werden.

Im Schaubild sieht man zusätzlich zu den paar Leitungen vom und zum Chip noch 4 Keramikkondensatoren mit $0,1\ \mu\text{F}$. Die bekommen Sie in jedem Elektronikfachgeschäft, ich habe sie aus einem alten Radio ausgelötet. Für meinen ersten Aufbau habe ich mir für den Chip eine Sockelleiste besorgt und diese Kondensatoren direkt an die Beine der Sockelleiste gelötet.

Dabei ist darauf zu achten, dass man den Chip erst nach dem Lötens aufsteckt, da er durch die Hitze des Lötens zerstört werden kann!



Der Chip wird direkt aus dem Raspberry mit Strom versorgt

Bild: ©Stephan Richter

Installation der Software

Jetzt muss man die Software auf dem Raspberry Pi zum Laufen bringen. Dazu sind folgende Schritte nötig:



- kompilieren und installieren des SCP-Servers
- Einrichtung
- Test
- Automatischer Start

Installation des Betriebssystems

Damit auf dem Raspberry Pi überhaupt ein Programm ausgeführt werden kann, müssen wir zunächst ein Betriebssystem installieren. Das ist einfach: Man lädt sich von www.berryterminal.com/doku.php/berryboot einen sogenannten Bootloader als Zip-Datei (berryboot-YYYYMMDD.zip) herunter und entpackt diese auf eine leere SD-Karte. Danach steckt man diese Karte in den Raspberry Pi, steckt diesen an Tastatur, Maus, Monitor und Internetzugang an und verbindet die Stromzuführung.

Nach wenigen Sekunden sollte ein Einrichtungsdialog (englisch) erscheinen. Hier wird man gefragt, ob man grüne Balken an den Bildschirmrändern sieht. Entsprechend wählt man "Yes" (ja) oder "No" (nein). Die Netzwerk-Voreinstellungen kann man so lassen, auch die Regions- und Spracheinstellungen sollten automatisch erkannt werden.



© Stephan Richter

Einrichtungsdialog bei der Installation des Betriebssystems

Bild: ©Stephan Richter

Im nächsten Schritt wird man gefragt, ob man auf die SD-Karte oder auf ein anderes Medium installieren will. Diesen Schritt können wir mit den Voreinstellungen (SD-Karte) hinter uns lassen. Durch Klicken auf "Format" wird die SD-Karte eingerichtet und der Bootloader installiert. Danach kommt die Betriebssystem-Übersicht. Hier wählt man das "Debian Wheezy Raspbian" und klickt auf "OK". Das Betriebssystem wird nun heruntergeladen und installiert, dies kann einige Minuten dauern. Anschließend kommt man in den BerryBoot Menü-Editor. Zum Verlassen einfach die Taste Esc drücken, und nach ein paar Sekunden startet das Betriebssystem.

Beim ersten Start sind noch ein paar Einstellungen vorzunehmen:

- Change User Password – hier kann man ein neues Passwort setzen (das Standardpasswort ist "raspberrry")
- Enable Boot to Desktop/Scratch – hier sollte man "Console" einstellen: Wir wollen keine Desktop-Umgebung, da dies das System nur ausbremsen würde.
- Internationalization – hier sollte man de_DE.UTF8 einstellen, wenn man ein deutsches Tastaturlayout haben will
- Enable Camera – kann man abgeschaltet lassen

- Advanced Options – hier sollte man ggf. SSH aktivieren, wenn man über Netzwerk auf das Gerät zugreifen will. Auch sollte man einmal die Option "Update" hier ausführen, um auf den neuesten Stand zu kommen

Anschließend speichern wir mit "Finish" die Einstellungen und werden in ein Terminal (eine Eingabeaufforderung) entlassen. Jetzt geht es ans Eingemachte.

Abschalten der Logmeldungen auf dem Terminal

Von Haus aus gibt das installierte Debian auf dem seriellen Port Statusmeldungen aus, die für Fehlersuche hilfreich sein können. In unserem Anwendungsfall stören diese Statusmeldungen jedoch und werden deaktiviert. Dazu muss man in der Datei /etc/inittab die Zeile

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

auskommentieren oder entfernen und dann den Raspberry neu starten. Wer sich mit Linux auskennt, kann das einfach auf der Konsole machen.

Alternativ kann man auch

```
sudo startx
```

eingeben und Enter drücken. Damit wird eine Desktop-Umgebung mit Admin-Rechten gestartet. Im Startmenü unter Zubehör findet sich der Texteditor Leafpad. Diesen starten wir und klicken auf "Datei > Öffnen". Im erscheinenden Dialog wählen wir links "Dateisystem" und navigieren anschließend zu "/etc". Dort doppelklicken wir auf die Datei "inittab". Sie wird geöffnet, wir entfernen besagte Zeile. Nach dem Speichern gehen wir übers Startmenü zu "Abmelden" und verlassen diese Sitzung wieder.

Kompilieren des SRCP-Servers

Um das Programm zur Erzeugung des Digitalsignals auf dem Raspberry auszuführen, müssen wir es zunächst herunterladen und installieren. Das geht auch



startx

lassen diesmal aber das vorangestellte "sudo" weg.

Auf dem Desktop gibt es eine Verknüpfung "Midori" – das ist ein einfacher Webbrowser. Den benötigten Quellcode bekommt man unter <https://sourceforge.net/projects/srcpd/files/srcpd/> . Dort wählt man eine Version und dann in der Dateiliste eines der Quellcode-Archive, beispielsweise srcpd-2.1.2.tar.gz oder srcpd-2.1.2.zip.

Wichtig: Version 2.1.1 funktioniert auf dem Raspberry nicht. Sofern es keine neuere Version gibt, sollte man 2.1.2 nutzen! Diese lädt man auf den Raspberry herunter. Dann öffnet man im Startmenü unter Zubehör den Dateibrowser und sucht die gepackte Datei – wahrscheinlich wird sie unter "/home/pi" liegen. Klickt man mit der rechten Maustaste darauf, hat man unter anderem die Option "Hier entpacken". Die wählen wir, und nach ein paar Sekunden gibt es einen Ordner "srcpd-2.1.2". Wir öffnen ihn per Doppelklick und drücken danach "F4".

Es öffnet sich ein Terminal. Dort führen wir

```
./configure
```

aus. Dieses Hilfsprogramm prüft, ob alle Werkzeuge vorhanden sind, die wir brauchen, um das Programm für den Raspberry Pi zusammenzustellen und zu installieren.

Der Aufruf von ./configure auf einer "jungfräulichen" Installation führte bei mir zu folgender Fehlermeldung:

```
checking for libxml – version >= 2.4.0... no
```

[...]

```
configure: error: no matching library found
```

Hier fehlte also noch ein Hilfsprogramm, das ich dann über den Befehl



dev telnets als Administrator ausgerollt werden soll. `apt-get install` als Administrator gestartet bedeutet dann wieder so viel wie "installiere die folgenden Pakete". `telnets`, `libxml2` und `libxml2-dev` sind dann die Paketnamen der zu installierenden Programme.

Nach der Installation dieser Programme kann man nochmal

`./configure`
starten. Bei der von mir genutzten Version von Debian und SRCP lief dieses Vorbereitungsprogramm ohne Fehler durch, zu erkennen an der letzten Zeile:

Run 'make' to continue

Das tun wir:

`make`
eingeben, Enter drücken.

Nun wird das Programm zusammengestellt und die Installation vorbereitet. Das kann eine Weile dauern. Währenddessen werden Unmengen kryptischer Meldungen angezeigt – davon sollte man sich nicht verwirren lassen. Wenn als letzte Zeile

make[1]: Leaving directory `/home/pi/srcpd-2.1.2'

stehenbleibt, hat das Erstellen geklappt, und das Programm kann mit

`sudo make install`
installiert werden.

Zusammengefasst:

`sudo apt-get install libxml2 libxml2-dev` = installiere das XML2-Hilfspaket

`./configure` = bereite Programmerstellung vor



`make` – erstelle das Programm

`sudo make install` = führe Installation (`make install`) als Administrator aus

Einrichtung

Jetzt ist es Zeit, das Programm einzurichten. Dafür müssen wir zunächst die Konfigurationsdatei des Programms anpassen. Wir starten wieder die grafische Oberfläche mittels

```
sudo startx
```

Wir öffnen "Leafpad" und in Leafpad die Datei `"/usr/local/etc/srcpd.conf"`. Hier ändern wir die Zeilen

3

zu

0

Den Abschnitt

...

ersetzen wir durch

```
<bus number> 1 </bus number>
<ddl>
<enable_maerklin>no</enable_maerklin>
<enable_nmradcc>yes</enable_nmradcc>
<enable_usleep_patch>yes</enable_usleep_patch>
</ddl>
<use_watchdog>no</use_watchdog>
<verbosity>0</verbosity>
<auto_power_on>no</auto_power_on>
<device>/dev/ttyAMA0</device>
<nmradcc_translation_routine>2</nmradcc_translation_routine>
</bus>
```

© Stephan Richter

Konfiguration in Leafpad anpassen

Bild: ©Stephan Richter

Gegebenenfalls kann man maerklin aktivieren/DCC deaktivieren. Die Verbosity-Einstellungen (also das Abschalten von Logmeldungen) ist wichtig, da sonst später Timing-Probleme auftreten können.

Nach dem Speichern ist das Programm fertig konfiguriert, wir können uns aus der grafischen Oberfläche abmelden.

Test

Jetzt ist es Zeit, zu testen, ob wir das Programm ausführen können. Dazu tippen wir den Befehl

```
/usr/local/sbin/srcpd -v
```

in der Konsole ein und drücken Enter.

Jetzt sollte das Programm seine eigene Version anzeigen. Ich bekam als Ausgabe

```
srcpd V2.1.2; SRCP 0.8.4; SRCPOTHER 0.8.3
```

So weit so gut. Der nächste Befehl, den wir probieren können, ist



nichts. Man kann aber mit Strg+Alt+F2 in eine zweite Konsole wechseln und sich dort mit Nutzernamen (pi) und Passwort (raspberrypi) anmelden. Dann gibt man

```
ip a
```

ein, um die aktuelle Adresse des Minicomputers zu erfahren. Da gibt es dann eine ähnliche Ausgabe wie folgt:

```
2: eth0: mtu 1500 qdisc >pfifo_fast state UP qlen 1000 link/ether  
b8:27:eb:d9:b1:e5 brd ff:ff:ff:ff:ff:ff inet 192.168.0.133/24 brd 192.168.0.255  
scope global eth0 valid_lft forever preferred_lft forever
```

wobei die 192.168.0.133 (bei mir) die gewünschte Adresse ist. Nun gibt man

```
telnet 4303
```

ein (also bei mir telnet 192.168.0.133 4303). Damit meldet man sich am Programm an (nur für Testzwecke) und müsste als Antwort die Ausgabe

Escape character is '^['.

srcpd V2.1.2; SRCP 0.8.4; SRCPOTHER 0.8.3

bekommen. Das bedeutet, dass der SRCP-Server funktioniert und auf Clients wie jman oder SRCPC wartet. Man kann diese Verbindung wieder trennen, indem man zunächst Strg+AltGr+] drückt und dann "quit" eingibt.

Man sollte also zu diesem Zeitpunkt mit einem beliebigen Programm, das das Simple Railroad Command Protocol (SRCP) spricht, zum Raspberry unter der oben ermittelten Adresse verbinden können. Hat man den Anschluss wie zuvor beschrieben vorgenommen, kann man ab jetzt schon mit einem Computer die Modellbahnfahrzeuge fahren lassen.

Automatischer Start



```
sudo startx
```

die Desktop-Umgebung und öffnen mit Leafpad die Datei `"/etc/rc.local"`. Vor der Zeile

```
exit 0
```

fügt man die Zeile

```
/usr/local/sbin/srcpd
```

ein und speichert die Datei. Dann meldet man sich wieder ab. In der Konsole führt man

```
sudo reboot
```

aus, um den Raspberry neu zu starten.

Die Software RocRail

RocRail ermöglicht es, per virtuellem Gleisbildstellpult eine Modellbahnanlage zu steuern und zu automatisieren.

Installation von RocRail

Nach dem Neustart kommt man beim Raspberry wieder auf der Konsole an. Nach der Eingabe von `"pi"` und `"raspberrypi"` müssen wir für RocRail zunächst ein paar weitere Hilfspakete installieren. Wir führen zunächst

```
sudo -s
```

aus, um uns als Administrator anzumelden. Nun steht am Anfang der Eingabezeile so etwas wie

```
root@raspberrypi:/home/pi#
```

Wir führen



Administrator-Modus über

```
exit
```

Mit

`git clone --depth 1 https://github.com/rocrail/Rocrail.git` Rocrail
holen wir uns den Quellcode von RocRail. Der Download kann eine Weile dauern.
Ist er fertig, wechseln wir mit

```
cd Rocrail
```

ins heruntergeladene Verzeichnis und führen

```
make server
```

aus, um die Server-Software zu erstellen. Auch hier ist Geduld gefragt, und es
werden unzählige kryptische Meldungen auf dem Bildschirm angezeigt, bis
schließlich

*make[2]: Leaving directory /home/pi/Rocrail/rocnethnode' make[1]: Leaving
directory/home/pi/Rocrail/rocrail'*

erscheint.

Nun ist es Zeit, das soeben erstellte Programm zu installieren. Nach dem Aufruf
von

```
sudo make install
```

kann man das Serverprogramm mit

```
/opt/rocrail/rocrail.sh
```

starten.

Wenn man nun auf einem anderen Rechner im Netzwerk RocView ausführt, kann
man sich über "Datei" > "Verbinden mit..." mit dem soeben gestarteten RocRail-
Server verbinden.



```
sudo apt-get install screen
```

geladen wird. Dann geht man wieder als Administrator mit

```
sudo startx
```

in die grafische Oberfläche, öffnet mit Leafpad nochmal die Datei `"/etc/rc.local"` und fügt die zwei Zeilen

```
sleep 10 su pi -c 'screen -d -m -S RocRail'
```

zwischen die Zeilen

```
/usr/local/sbin/srcpd
```

```
exit 0
```

Dann startet man

```
leafpad .screenrc
```

um eine neue Konfigurationsdatei für das Programm "Screen" zu erzeugen. In diese Datei schreiben wir folgendes:

```
startup_message off  
altscreen on  
screen bash -c 'sleep 7; /opt/rocrail/rocrail.sh'  
detach
```

Nach einem Neustart sollten nun sowohl der SRCP-Server als auch RocRail auf dem Raspberry starten.

Einrichtung von RocRail

Als nächstes richten wir auf dem Desktop die Anbindung von RocRail an den zuvor installierten SRCP-Server ein. Dazu gehen wir in RocView auf "Datei" > "Rocrail Eigenschaften" (nur verfügbar, wenn RocView mit dem Server verbunden ist) und im sich öffnenden Dialog auf den Reiter "Zentrale".



Klicken auf "Eigenschaften". Hier müssen folgende Werte eingetragen werden:

Schnittstellenkennung: srcp

Hostname: 127.0.0.1

Port: 4303

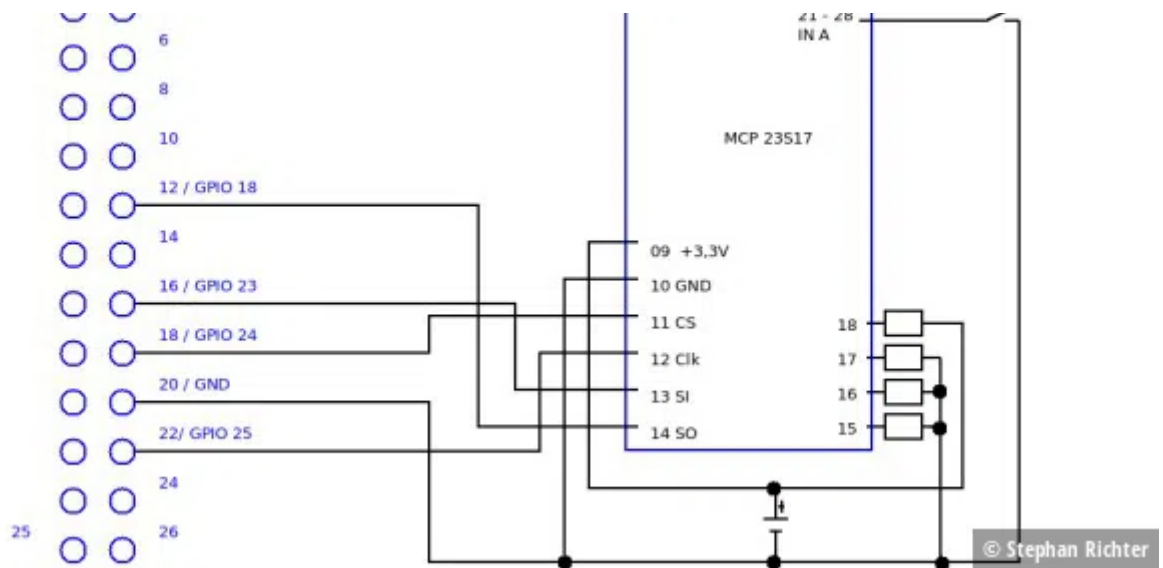
Gerät: /dev/ttyAMA0

Sub-Bibliothek: TCPIP

Im gleichen Dialog müssen wir oben noch auf "Busse" gehen und alle Werte auf 1 setzen. Einzig der Server wird auf 0 gesetzt. Dann alles speichern. Diese Änderungen werden erst nach einem Neustart von RocView aktiv. Danach sollte es in RocView möglich sein, Lokomotiven anzulegen und diese händisch zu steuern. Auf eine Erklärung dieser Schritte verzichte ich hier, das wird ausführlich unter <https://wiki.rocrail.net/doku.php> erklärt.

Verbindung Rückmeldekontakte => Raspberry Pi

Hier ist wieder etwas Elektronik zu löten. Zunächst muss man sich bei einem Elektronikversand oder Elektronikfachhändler mindestens einen MCP23S17-Chip, eine Lochrasterplatine und eine Sockelleiste besorgen. Der Schaltplan für den Anschluss des Chips ist hier:



Schaltplan für den Chip-Anschluss

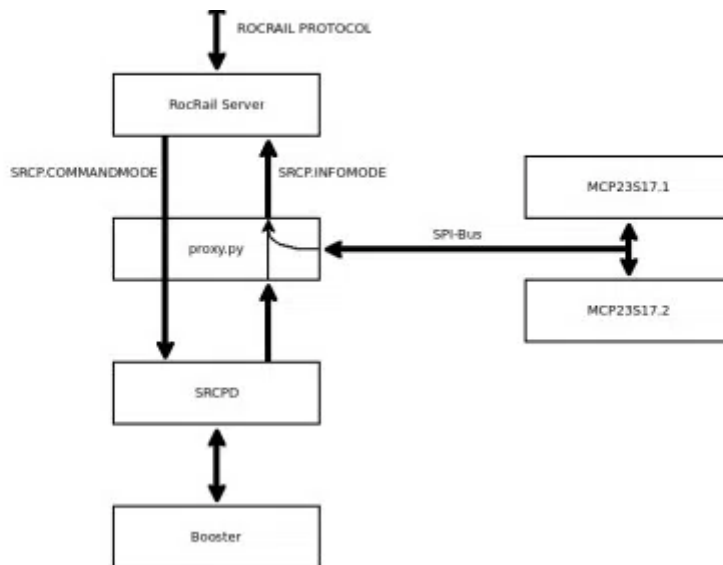
Bild: ©Stephan Richter

Man unterbricht als erstes die Leiterbahnen auf der Platine und lötet die Sockelleiste über die Trennstelle. Dann setzt man die Widerstände und Drahtbrücken, so dass letztlich die Verbindungen wie im ersten Bild hergestellt sind. An den Pins 21 bis 28 des Chips können wir Gleiskontakte anschließen, die gegen Masse (GND) schließen. Die Stromversorgung (3 bis 3,3 V) lässt sich mit zwei Batterien oder mit einem Netzteil anschließen. Damit ist die Hardware komplett.

Was man nun noch braucht, ist ein bisschen Software, die die Kontaktinformationen zur Verfügung stellt.

Reaktion auf Rückmeldeereignisse

Damit SRCP-fähige Programme auch Informationen über Kontakt-Ereignisse bekommen, habe ich das Python-Script "model-rail-control" (<https://github.com/StephanRichter/model-rail-control>) erschaffen. Dieses schaltet sich zwischen das SRCP-Programm und ein Client-Programm. Ich habe es mit RocRail – übrigens einer großartigen Modellbahnsteuerungs-Software – und der Oberfläche RocView getestet.



© Stephan Richter

Das Python-Script "model-rail-control" schaltet sich zwischen das SRCP- und ein Client-Programm

Bild: ©Stephan Richter

Die nächsten Schritte sind

- Installation des model-rail-control-Scripts
- Anpassung der RocRail-Eigenschaften
- Test mit RocRail

Installation des model-rail-control-Scripts

Wir melden uns wieder als Benutzer "pi" mit dem passwort "raspberry" an. Danach führen wir

`git clone https://github.com/StephanRichter/model-rail-control.git`
aus. Damit wird das Script heruntergeladen und im Ordner /home/pi/model-rail-control abgelegt. Auch hier soll ein automatischer Start eingerichtet werden. Diesmal machen wir es allerdings über die Konfiguration von Screen. Wir starten

`leafpad .screenrc`
und ergänzen



zwischen die schon vorhandenen Zeilen

```
altscreen on
```

```
screen bash -c 'sleep 7; /opt/rocrail/rocrail.sh'
```

Anpassung der RocRail-Eigenschaften

Nach einem weiteren Neustart des Rasperrys (und etwa 30 Sekunden warten...) öffnen wir wieder RocView, stellen im Menü "Datei" die Serververbindung zu raspberrypi her, falls nicht schon automatisch geschehen, und gehen dann im Menü "Datei" auf "Rocrail Eigenschaften". Es öffnet sich der bekannte Dialog, wo wir auf "Zentrale" klicken und den SRCP-Server markieren. Ein Klick auf "Eigenschaften" und hier die Portnummer von 4303 nach 4304 ändern. Speichern, und schon sollte Kontakt zu den angeschlossenen Rückmeldern bestehen.

Test mit RocRail

Nun kann man die Funktion wie folgt testen: Man loggt sich, falls nicht schon geschehen, am Raspberry ein und führt

```
screen -r
```

aus. Nun sieht man die Ausgaben von RocRail. Löst man jetzt einen der an den MCP23S17 angeschlossenen Rückmelder händisch aus, sollte eine entsprechende Meldung erscheinen, vorausgesetzt, man hat in den RocRail-Einstellungen (in RocView auf "Datei > Rocrail Einstellungen" gehen, dann auf "Fehlerprotokoll") das Monitoring angeschaltet.

Bei mir kommt beispielsweise

```
1000 sensor 0:31=1 ident=
```


Eigenschaft "Schnittstelle" den Wert 0 für den Bus (die Zahl vorm Doppelpunkt) und die 31 für den Sensor (Zahl nach dem Doppelpunkt) eingeben.

The screenshot shows a software window titled 'Rückmeldung R_51.6' with a tabbed interface. The 'Schnittstelle' tab is active. It contains the following fields and options:

- 'Schnittstellenkennung': An empty text input field.
- 'UID-Name': An empty text input field.
- 'Adresse' section:
 - 'Bus': A dropdown menu showing '0'.
 - 'Adresse': A dropdown menu showing '31'.
- 'Kurzschluss' section:
 - 'Bus': A dropdown menu showing '0'.
 - 'Adresse': A dropdown menu showing '0'.
- 'Typ' section with radio buttons:
 - ☒ Sensor
 - ☐ Lissy
 - ☐ Transponding
 - ☐ Barcode
 - ☐ Railcom
 - ☐ RFID
 - ☐ Wheel counter
- 'Optionen' section with checkboxes:
 - ☐ Aktiv niedrig
 - ☒ Zurücksetzen

At the bottom are three buttons: 'Übernehmen', 'Abbrechen', and 'ok'.

© Stephan Richter

Einrichtung eines Rückmeldekontaktes

Bild: ©Stephan Richter

Dann sollte in RocView der Sensor-Gleisabschnitt blinken, wenn der Sensor ausgelöst wird.

Und damit haben wir nun alle Voraussetzungen geschaffen, um in RocView eine automatische Modellbahnsteuerung zu programmieren.

Diese Programmierung ist jedoch nicht mehr Teil meiner Anleitung. Informationen dazu finden Sie in der RocRail-Dokumentation.



**ÜBER
UNS**

PC-WELT INTERNATIONAL

ABONNEMENT

Copyright © 2022 IDG Communications, Inc.

Explore the Foundry Network +

SEITENANFANG

Datenschutzeinstellungen