

s88udp

Was ist das?

s88udp ist ein C Programm für den Raspberry Pi. Es liest über die GPIOs die Signale eines S88 Bus ein und sendet die Ergebnisse per Netzwerk weiter. Dabei werden die Daten auf Netzwerkebene im Märklin CAN Bus Format gesendet und sind damit kompatibel zu den Meldungen die eine Märklin Central Station 2 versenden würde. Der S88 Bus ist ein Signalbus aus dem Modellbahn Bereich und transportiert zum Beispiel Besetztmeldungen für bestimmte Gleisabschnitte.

Wozu dient das?

Der Raspberry Pi ist ein wunderbar kleiner Computer, der sich hervorragend dazu eignet, als Steuerungsrechner für eine Modellbahn eingesetzt zu werden. Eins der dabei auftretenden Themen, ist die Frage, wie bekomme ich meine Besetztmeldungen in den Rechner. S88udp löst dieses Problem, indem es auf der Hardwareseite die nötigen Signale zum Bedienen des S88 Buses liefert und die Ergebnisse in einem weit bekannten Format an ein mögliches Steuerungsprogramm weiterleitet. Weitverbreitet zur Steuerung ist das Programm "rocrail". Aber s88udp kann auch mit jedem anderen Steuerungsprogramm verwendet werden, das die Netzwerksschnittstelle der Märklin CS2 unterstützt.

Wer braucht sowas?

Alternde Computerfreaks, die mit der Eisenbahn spielen ;-). Ich wollte möglichst einfach die Signale meines S88 Buses in meinen Raspi bekommen, mit dem ich meine kleine Modellbahn steuere. Da mich der ganze Rechner nur 35,- Euro gekostet hatte, sah ich nicht ein, für ein fertiges S88 Interface über 100 Euro auszugeben. Da ich die Zugsteuerung bereits mit einem selbstgebauten CAN Bus Interface und einem kleinen Programm Namens can2udp realisiert hatte, lag es nahe, etwas ähnliches für den S88 Bus zu realisieren.

Wie komme ich da ran?

Indem Ihr [hier](#) das Programm runterladet, auspackt und mit

```
make
make install
```

auf eurem RasPi kompiliert und installiert. Dazu benötigt ihr die Bibliothek bcm2835, die die notwendigen Funktionen zum Zugriff auf die I/O Pins des Raspberry Pi bereit stellt. Information zum Bezug und Anwendung der Bibliothek findet ihr an dieser Stelle <http://www.airspayce.com/mikem/bcm2835>. Unter http://elinux.org/RPi_Low-level_peripherals findet ihr die Stiftbelegung zum Anschluss an den S88 Bus, wie sie in der Datei s88udp.h definiert ist und diverse Beispiele zur Ansteuerung. Selbstverständlich könnt ihr auch andere PINs dafür benutzen und definieren.

Wie benutze ich das?

Ihr startet das Programm am besten erstmal mit der Option -h. Dann wird euch geholfen. Nützlich ist auch die Option -vf für Verbose und Foreground. Dann wird eine Reihe Nullen angezeigt, die den aktuellen Live Zustand der Eingänge des ersten Moduls anzeigt.

Wenn ihr soweit seid, dass das mit der Sensoranzeige funktioniert, kommt der Teil mit der Verbindung zum rocrail Server. Dazu muss in rocrail eine Zentrale vom Typ mcs2 eingerichtet werden und die IP Adresse des Servers auf dem s88udp läuft. Umgekehrt muss dem Programm s88udp mit der Option -d wie Destination die Adresse des Servers mitgeteilt werden, auf dem rocrail läuft. Sind beide Programme auf dem selben Raspberry Pi installiert, ist es in beiden Fällen die 127.0.0.1

Hardware

Der S88 Bus muss im Normalfall über Optokoppler und Pegelwandler am Raspberry Pi angeschlossen werden. Die dazugehörige Schaltung kann ich leider nicht liefern, weil:

- Im Sonderfall, dass die original S88 Schaltung mit ICs vom Typ 4014N und 4044N verwendet wird, kann auf Pegelwandler verzichtet werden. Die CMOS Bausteine funktionieren auch mit den 3,3 Volt des Raspberry Pi.
- Im Sonderfall, dass die S88 Bausteine keinerlei Verbindung zum Rest der Modellbahn und Steuerung hat, kann auf die galvanische Trennung durch Optokoppler verzichtet werden.

Ich verwende den s88-3 Baustein von TAMS und benutze Reed Kontakte als Sensoren. Damit sind für mich beide Sonderfälle erfüllt und ich habe die GPIOs plus Ground plus 3,3 Volt direkt an den S88 Bus angeschlossen. Einfacher gehts nicht.

RasPi	S88N
Steckerleiste	Steckerbelegung
P1-01---3,3V-----	+5V----PIN1
P1-06--Ground-----	GND----PIN3+PIN5
P1-11--GPIO17-----	DATA---PIN2
P1-15--GPIO22-----	CLOCK--PIN4
P1-16--GPIO23-----	LOAD---PIN6
P1-18--GPIO24-----	RESET--PIN7

Bitte auf gar keinen Fall nachmachen, wenn die oben angesprochenen Vorraussetzungen nicht gegeben sind! Der Raspberry verträgt keine Eingangsspannungen > 3,3 Volt und eine Verbindung zwischen Booster und Raspberry kann sowohl die Booster Endstufe, als auch den Pi zerstören!

siggi(at)s88udp.de