

Linux ist sexy Freiheit!

DAB/UKW/FM-Kanäle scannen
mit dem Raspberry-Pi3

UKW-Tagung
Weinheim (8.9.2018)

Hüfingen Update (24.11.2018)

Hayati Aygün
(UKW/TV-Arbeitskreis)

Email: h_ayguen@web.de



Überblick

- Linux, Skripting und Entwicklung
- Raspberry-Pi 3B+
- DAB und UKW/FM Scanner
- Ausblick des Projekts

Linux und Skripting

- Raspberry-Pi → automatische Wahl von Linux ?
- Nicht ganz! Beschäftigung mit Linux seit ca. 1995
- Linux/Unix Paradigma:
“kleine (schöne) Programme verknüpfen”
- Bash/Shell als Klebstoff

Skripting: Bash/Shell als Klebstoff

Umleitungen:

- < Eingangsdatei
- > Standard-Ausgang an Datei (Ziel überschreiben)
- >> Standard-Ausgang (anhängen)
- 1> Standard-Ausgang
- 2> Fehler-Ausgang
- > Ausgang 2>&1 (Standard + Fehler an Ausgang)
- &> Standard+Fehler-Ausgang

| Standard-Ausgang als Eingang
für weiteres Programm: “Pipe”.
quasi-gleichzeitige Ausführung (Parallelisierung)

Skripting: Beispiel aus dem Projekt

```
cat ${rdy_rec_name}.raw \      # zuvor von rtl_sdr aufgezeichnet
| csdr convert_u8_f \          # 8 Bit nach Fließkomma
| csdr fastdcblock_ff \        # DC entfernen
| csdr shift_addfast_cc 0.123 2>/dev/null \    # Frequenz mischen
| csdr fir_decimate_cc 14 0.0741 HAMMING 2>/dev/null \ # DDC
| csdr fmdemod_quadri_cf \      # FM demodulation (→ MPX)
| csdr convert_f_s16 \          # Fließkomma nach 16 Bit
| redsea --bler \              # RDS Dekoder
>> redsea.107M.txt              # Ergebnis in diese Datei
```

Skripting: Bash/Shell als Programmiersprache

- Variablen:

```
T=123  
STR="789"  
echo "$T  ${STR}"
```
- Arrays:

```
A=( 0 1 "hallo" "welt" )  
echo "${A[2]} ${A[3]}"
```
- Bedingungen:

```
if [ "$T" = "hello" ]; then  
    ..  
fi
```
- Schleifen:

```
for f in $( ls -1 ); do  
    ..  
done
```

Skripting: Bash/Shell als Programmiersprache

- Berechnungen: `SUM=$((3 + 4))`
- Funktionen:

```
foo() { echo $1 $2; }  
foo "Hallo" "Welt!"
```
- Includes: `source file / . file`
- Programmausführung:
 - im Hintergrund: `ls -l &`
 - intern für Variablen: ``ls -l``
 - einfacheres Quoting: `$(ls -l)`

<https://ss64.com/bash/>

Sexiness → Freie Software

- Bash ist also nicht nur eine Eingabeaufforderung
- sondern eine (von vielen) Programmiersprachen
- Die Mächtigkeit kommt von den vielen *freien* Programmen,
die man *verknüpfen* kann!
- Wichtiger Faktor: Software-Lizenz(en)
GPL, LGPL, BSD, MIT, ..

weitere Themen: Software für Behörden
 Code-Review → Sicherheit
 Software-Patente als Waffenarsenal

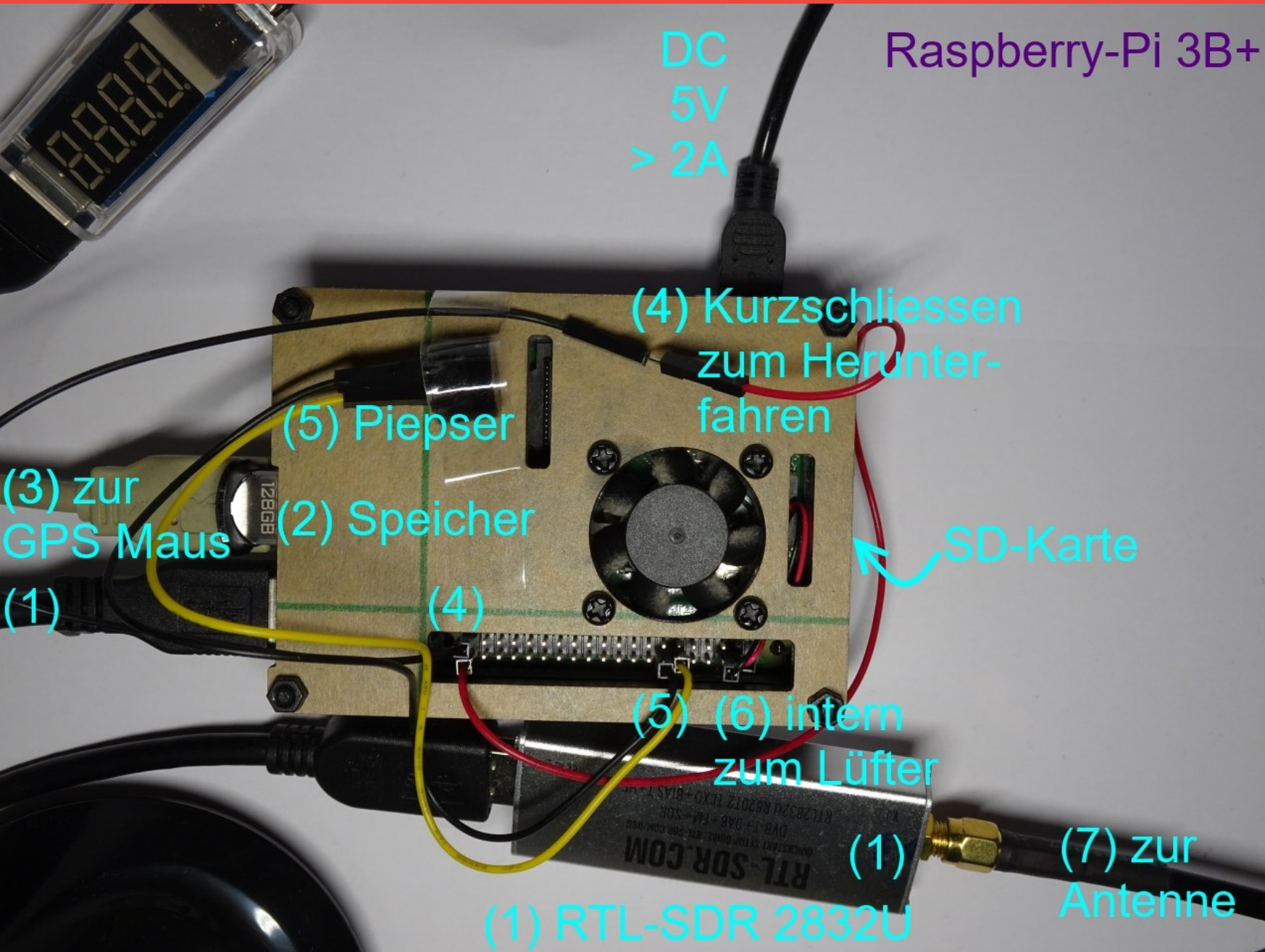
Scanner Ziele (1)

- **Sammeln digitaler Senderinformationen von**
 - UKW/FM (VHF-Band II: 87,5 – 108,0 MHz) → RDS-Dekoder
 - DAB (VHF Band III: 174 – 230 MHz) → Dekoder
 - DVB-T2 (474 – 786 MHz) → mit 8 MHz zu breit für günstigen RTL2832U!
- **Stationär wie auch mobil nutzbar, z.B. im Kfz**
 - Mobil: → GPS, Scan Geschwindigkeit!
- **Einspeisung in die FMLIST-DB des UKW/TV Arbeitskreises**
 - Offen → funktioniert

Scanner Ziele (2)

- **Erkennung
von besonderen Ausbreitungsbedingungen**
→ 2m perfekt zwischen UKW/FM und DAB
- **Relativ günstige Hardware**
- **KEIN DX-Empfang!**

Scanner Basis / Voraussetzungen



Hardware Voraussetzungen

- Günstig → RTL-SDR Empfänger + Antenne
- USB-Speicher (neben SD-Karte) zur Speicherung der Daten
- Uhrzeit (NTP), Benachrichtigungen (per eMail) → Internet
- Mobil → GPS-Maus für Koordinaten + Uhrzeit (Raspberry-Pi!)
- Mobiler Upload (über Mobiltelefon Hotspot) → WLAN
- Mobil → schneller Scan → Raspi 3 B+: 4x 1400 MHz, 1GB RAM, 4xUSB
- Mobil → geeignetes Kfz-Netzteil ! bzw. Powerbank
- Piezo-Piepser für akustische Benachrichtigungen
- Mobil: Drahtbrücken für sauberes Herunterfahren des Raspi → lpie

UKW/FM Software für 87,5 – 108,0 MHz

- **rtl_fm | redsea**

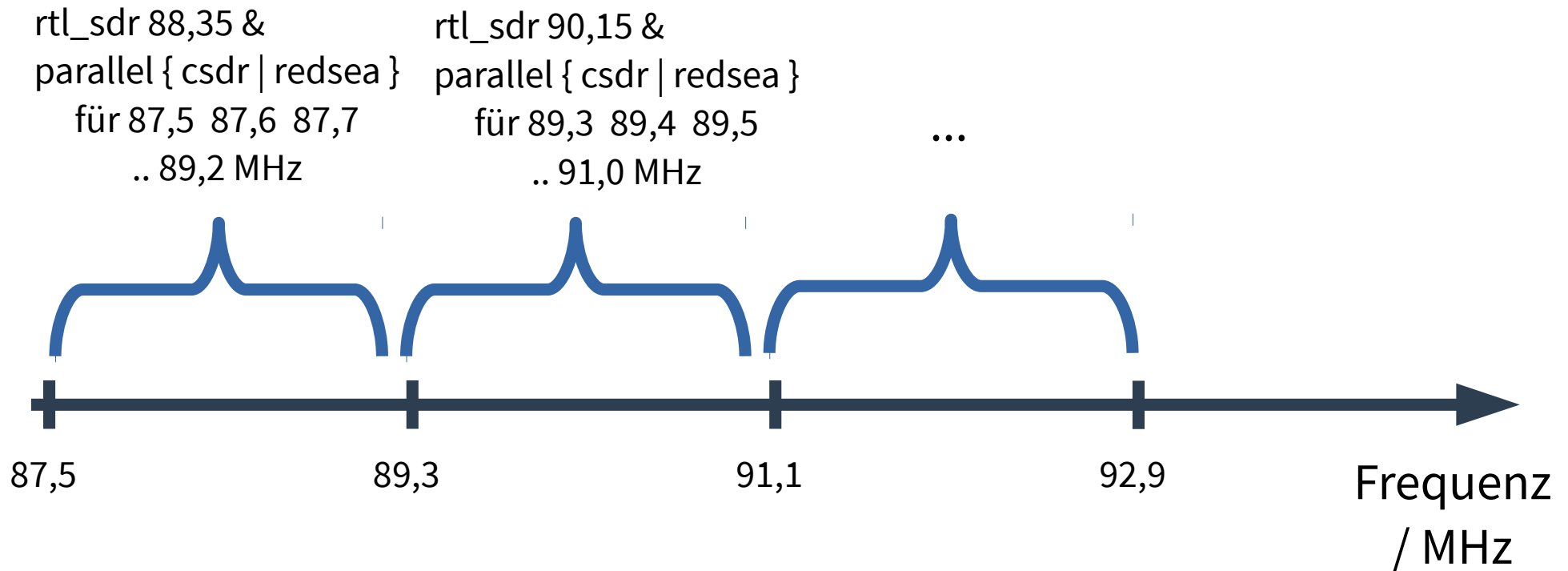
- rtl_fm (verschiedene Autoren, auch rtl_sdr) zur Demodulation je Kanal:
216 FM-Kanäle à 3 Sekunden → 648 Sekunden
- redsea zur RDS-Dekodierung (Oona Räisänen, OH2EIQ)

- **rtl_sdr ; gnu_parallel { csdr | redsea }**

- rtl_sdr zur Aufzeichnung von 2 MHz Bändern (genauer: $> \pm 850$ kHz):
12 Subbänder à 3 Sekunden → 36 Sekunden
- Gnu parallel (Ole Tange) zur parallelen Verarbeitung aller Kanäle
- csdr zur Demodulation je Kanal (Andras Retzler, HA7ILM, OpenWebRX)
- redsea zur RDS-Dekodierung

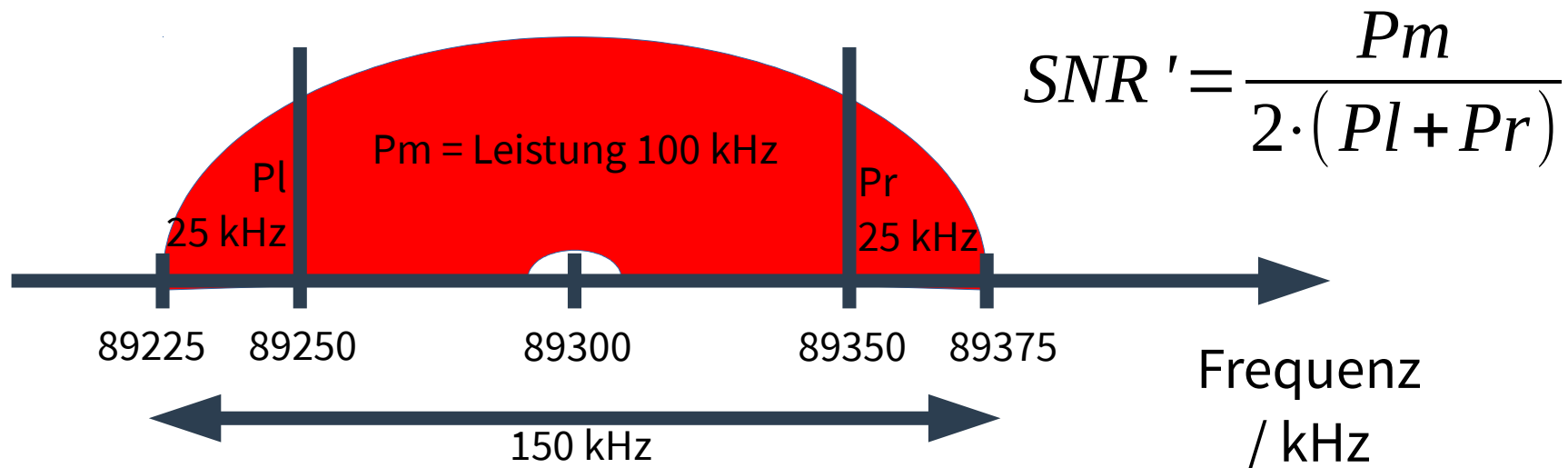
UKW/FM: rtl_sdr + parallel { csdr | redsea }

Diese Variante wird genutzt:



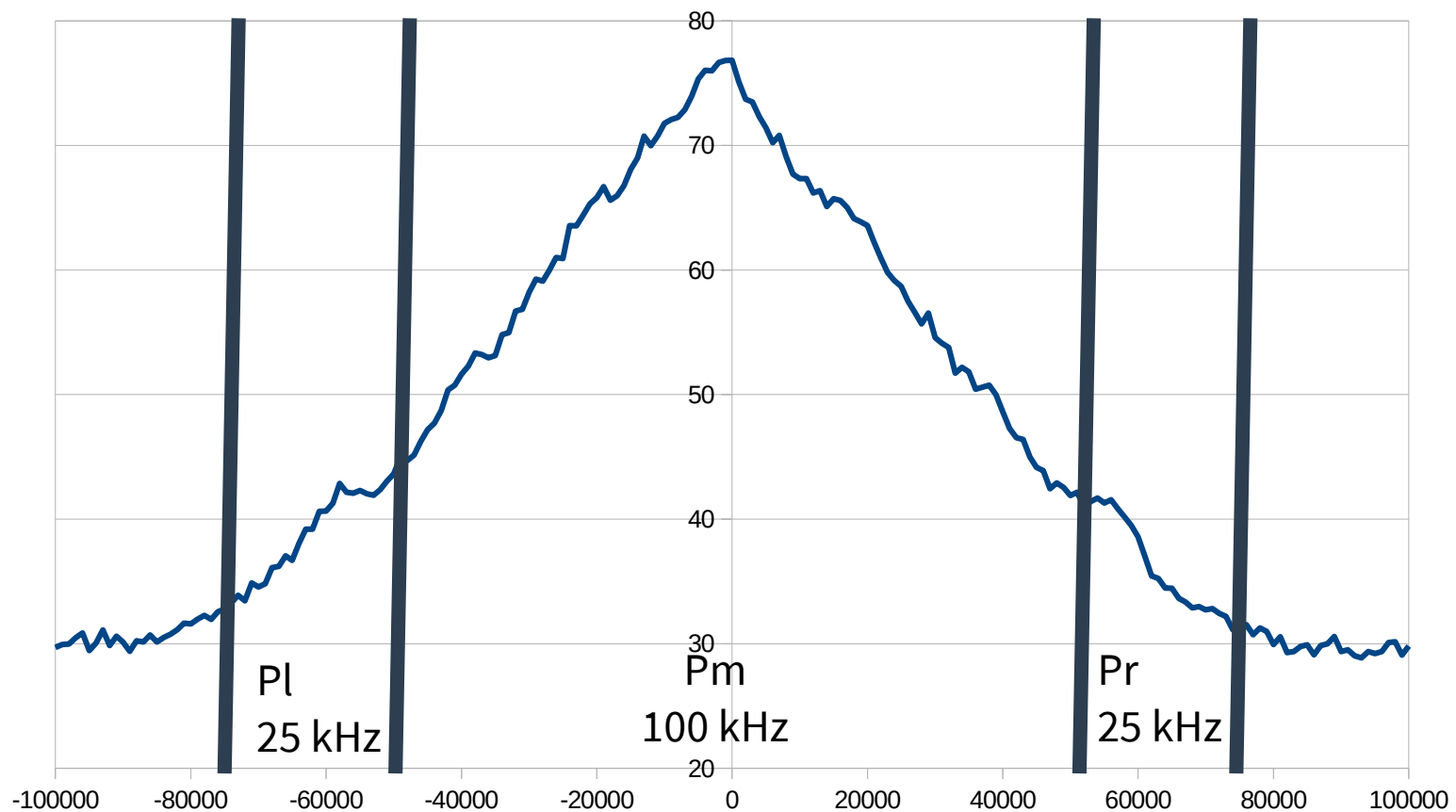
UKW/FM: rtl_sdr + parallel { csdr | redsea }

Weitere Scan-Beschleunigung durch
"spektrale Voruntersuchung" der Kanäle im 2 MHz Band
anhand 200 – 500 ms Signal:



UKW/FM: rtl_sdr + parallel { csdr | redsea }

Spektrale Voruntersuchung (Aufzeichnung)



UKW/FM: rtl_sdr + parallel { csdr | redsea }

Spektrale Voruntersuchung:

- Demodulation/Dekodierung vieler Kanäle kann eingespart werden
- Keine Falsch-Positiv RDS-Dekodierung mehr bei sehr starken Signalen im Nachbarkanal
- Positiver Nebeneffekt:
 - weniger CPU-Abwärme → niedrigere Temperatur
 - geringerer Stromverbrauch → Öko ..
 - oder längere Laufzeit an Powerbank :-)

DAB - Software für 174 – 230 MHz

- **dab-cmdline**

- Kommandozeilen Beispiele zu Qt-DAB (Jan van Katwijk)
- Liest Daten direkt vom RTL-SDR sowie anderen Empfängern
- Keine Parallelisierung, da bereits 1,6 MHz breit
- Kommandozeile bot keinen Scan Betrieb
- Hilfsprogramm um Scan Betrieb ergänzt:
Abbruch nach Sammeln der Ensemble/Senderinformationen

DAB – Software: Scan - Beschleunigung

- **Weiteres Hilfsprogramm zur Scan-Beschleunigung erstellt**
 - DAB nutzt OFDM Modulation
 - OFDM: viele Unterträger: konkret: 1536
 - DAB strahlt im Gleichwellennetz: mehrere Sendestandorte senden auf gleicher Frequenz
 - Autokorrelation im Nutz-Symbol-Abstand T_u : 1 ms bei Mode I
 - Gesamt Symboldauer T_s = Guard Intervall + Nutzdauer
 - Guard Intervall (aka. Cyclic Prefix) = Kopie vom Ende des Nutzsymbols

Ausblick / Offene Arbeiten UKW/FM

- Empfangs-Dynamik: kleinere Bänder statt 1,8 MHz
 - unter Nutzung der RTLSDR internen Bandfilter
 - Erledigt. Über Konfiguration einstellbar.
- Auswertung / Zusammenfassung der JSON Ergebnisse von redsea
 - Erledigt. Bei PI Code, .. wird häufigstes Ergebnis weitergeleitet
- Erkennung des Stereo-Pilotons auf 19 kHz im FM-demodulierten MPX – zur Erkennung ob Aussendung mono/stereo

Ausblick / Offene Arbeiten DAB

- Vorauswahl durch Prüfen
des Null-Symbol-Einbruchs im Frame Abstand
- Vorauswahl durch Kreuzkorrelation
mit dem DAB Phase Reference Symbol (PRS)
- Zusammenfassung der Senderinformationen zu .CSV
→ Erledigt.

Ausblick / Offene Arbeiten Allgemein

- Automatischer Upload an FMLIST: → Erledigt.
- Stationär: Erkennung +Meldung
besonderer Ausbreitungsbedingungen
- Konfiguration abhängig von GPS Koordinaten:
 - UKW Kanalraster: 100 kHz. Für Italien eher 50 kHz?
 - UKW Band: D: 87,5 – 108,0 MHz GB: 88,0 – 107,9 MHz weitere
 - UKW Aufzeichnungsdauer: 3 sec
 - DAB Kanäle (13 C/D)
 - DAB Scan Dauern
Max Synchronisation, Max bis Ensemble Information, Minimum SNR, ..

Neue Funktionen 1

seit 8.9.2018 (Weinheim) bis 24.11.2018 (Hüfingen)

- **GPS Koordinaten werden häufiger erfasst**
 - neu: alle 3 - 4 Sekunden
 - Zuvor: einmal pro Minute
- **Read-Only Dateisystem für SD-Karte bzw. dessen Lebensdauer: layerpie ('sudo lpie [status]') aufgenommen**
- **Piepser-Tonfolgen zur Signalisierung einfach(er) änderbar**
→ scanToneFeedback.sh

Neue Funktionen 2

seit 8.9.2018 (Weinheim) bis 24.11.2018 (Hüfingen)

- **Initialeinrichtung über WiFi/WLAN und SSH**
- ohne Anschluss von Tastatur und Bildschirm an den Raspberry
- **Ohne Tastatur/Bildschirm am Raspberry – am Notebook oder Smartphone – über den USB Speicherstick ..**
 - .. WiFi/WLAN Zugang über erweiterbar
 - .. Komplette Scanner Konfiguration editierbar.
Übernahme beim nächsten Start

Neue Funktionen 3

seit 8.9.2018 (Weinheim) bis 24.11.2018 (Hüfingen)

- **Neubelegung des früheren ‘Reboot’-Tasters (am ATX-Gebilde):**
 - Scan beenden: ‘stopBgScanLoop’
 - Warten: N x ‘screen -ls’
 - Ergebnisse aufbereiten: ‘prepareScanResultsForUpload.sh’
 - Ergebnisse hochladen (versuchen): ‘uploadScanResults.sh’

Bekannte Fehler

Stand 24.11.2018

- Dauerpiepsen nach Stop des Scanners,
z.B. mit 'stopBgScanLoop.sh'
Workaround: 'scanToneFeedback.sh welcome'

Neue Anregungen / Wünsche aus Hüfingen – vom 24.11.2018

- Grafische Oberfläche (X Windowing) mit installieren
.. aber nicht automatisch nutzen
- SDR, wie z.B. gqrx installieren.
ggf. weitere wie fldigi, ..
- Komplett-Image (gparted, ..) bereitstellen
- Anleitung mit Checkliste zum Abhaken versehen

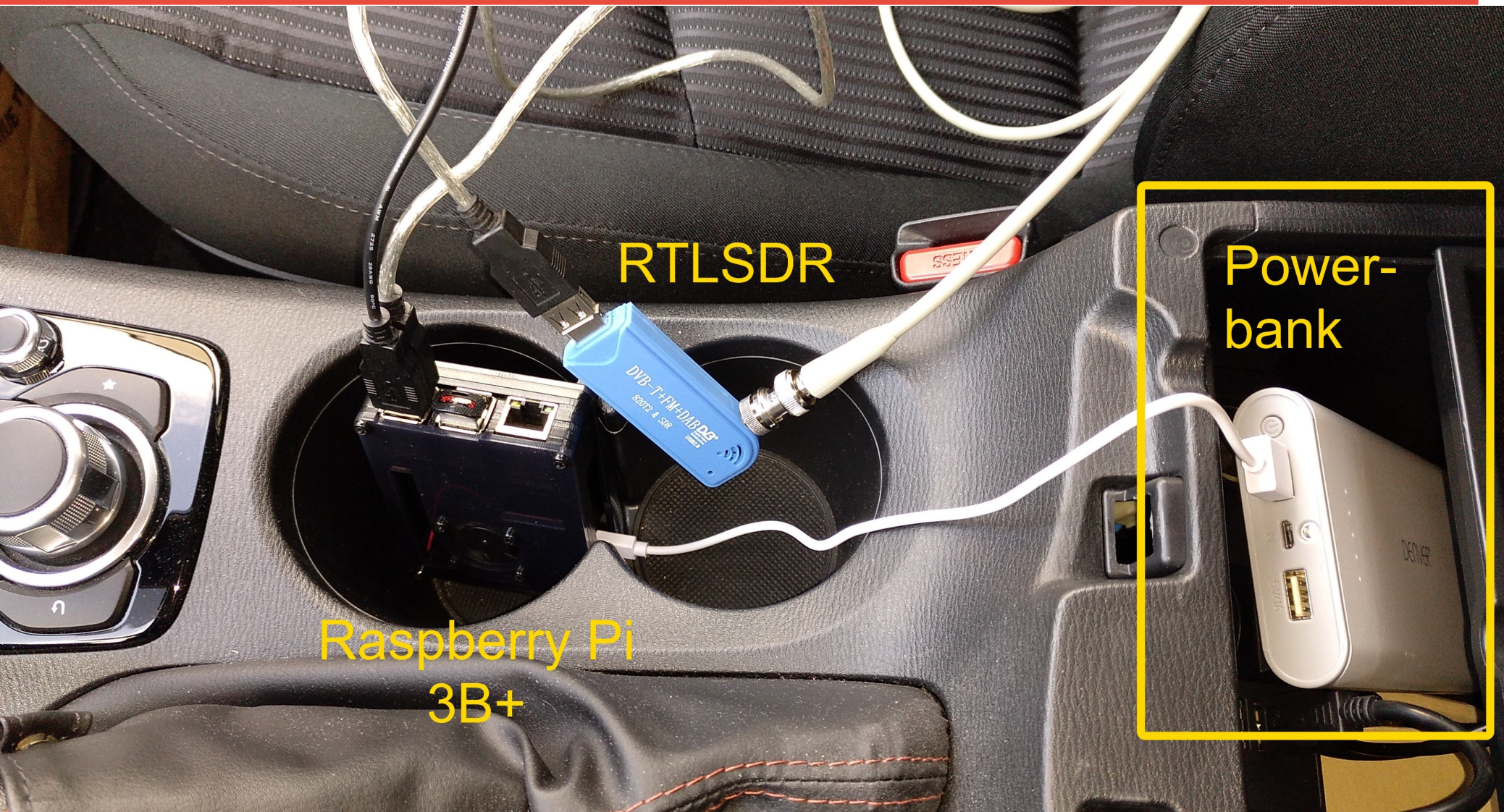
Aus dem Testbetrieb (1)



GPS-
Maus

ANTenne

Aus dem Testbetrieb (2)



Aus dem Testbetrieb (3)



Aus dem Testbetrieb (4)

Erfahrungsbericht(e) von Alpha Testern

Oliver Schmidt

Günter Lorenz

vom UKW/TV – Arbeitskreis

Fazit

Ohne Linux

und die vielen vorhandenen Open-Source Programme

wäre das Projekt

https://github.com/hayguen/fmlist_scan

nicht denkbar gewesen!



ENDE

und

Danke für die Aufmerksamkeit!