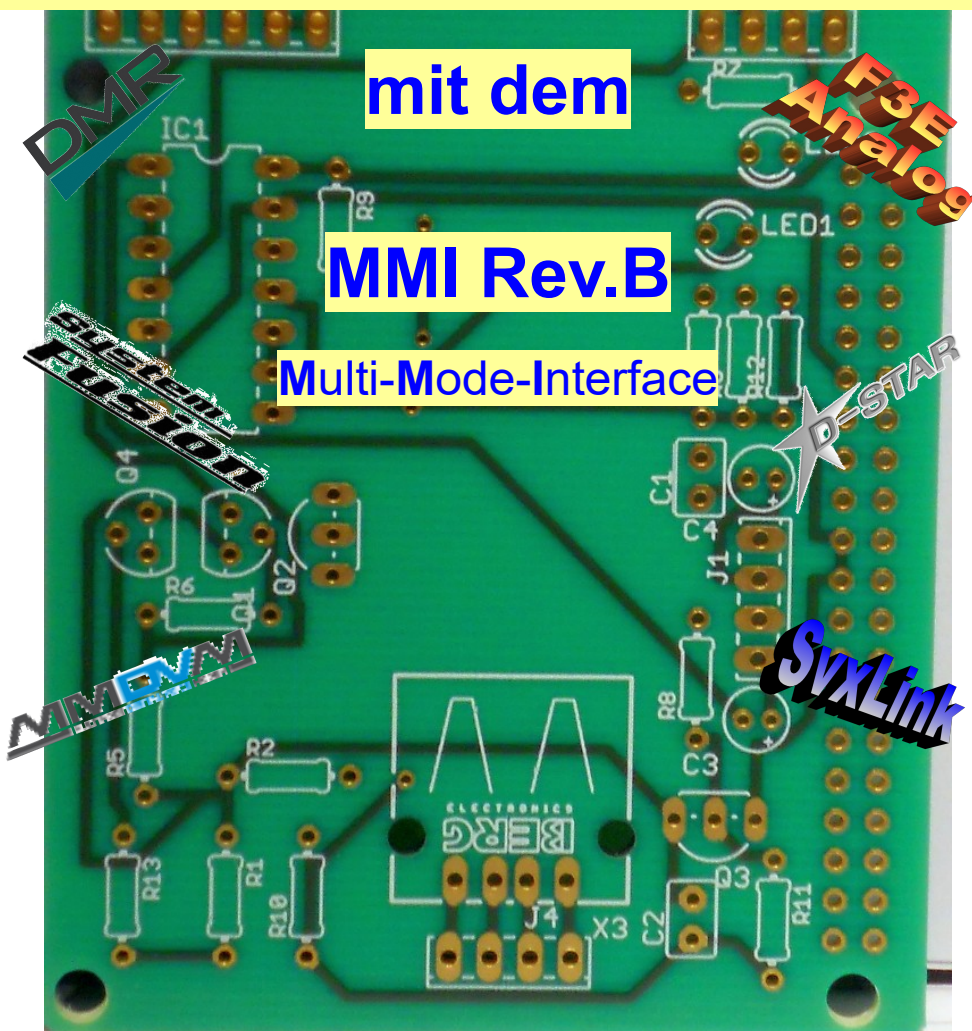


Kleines Kochbuch zum Betrieb



Inhaltsverzeichnis

1. Was wird alles benötigt.	3
2. Blockschaltbild der Steuerung	3
3. Das Multi-Mode-Interface	4
4. Hinweis zur USB-Soundkarte	10
5. SvxLink bei digitalem Signal auf der Eingabe absteuern	11
6. Nun wird ein Shell-Script benötigt welches die Logdatei vom MMDVM-Host prüft	11
7. Nun fehlt noch das Einrichten der GPIO's für SvyLink mit einem weiteren Shell-Script	12
8. Die erstellten Files müssen beim starten von Linux aufgerufen werden	12
9. Unter crontab muss noch folgender Eintrag hinzugefügt werden	13
10. Datei Logik.tcl von SvxLink modifizieren	13
11. Die RepeaterLogik.tcl von SvxLink modifizieren	15
12. Anpassen der svxlink.conf Datei	17

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

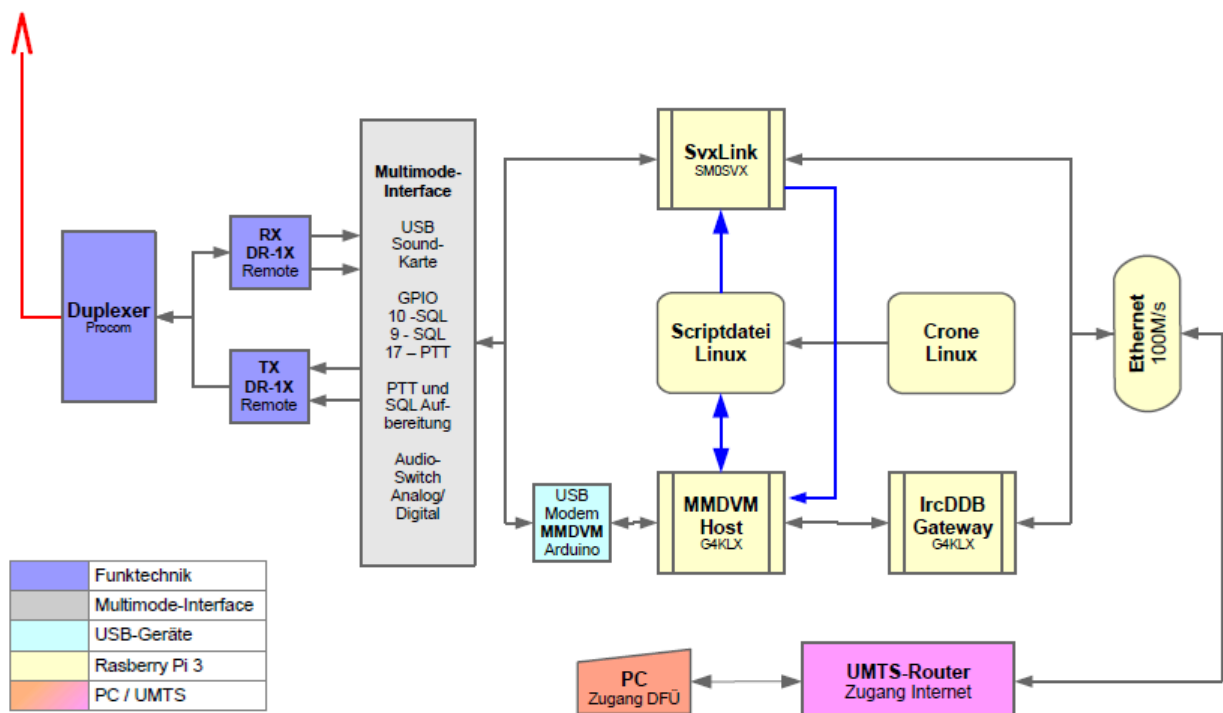
1. Was wird alles benötigt.

Raspberry Pi 2 oder 3
MMI-Platine für den Raspi (Multimode-Interface)
SvxLink Software
Arduino DUE mit MMDVM-Shield
Software für den Arduino
MMDVM Software für den Raspi
Linux-Editor mc (Midnight-Commander) oder nano

Ausserdem müssen ein paar Dateien angelegt werden um eine effektive Steuerung zu schaffen. In der SvxLink-Software müssen zwei TCL-Skripte bearbeitet werden. Natürlich müssen auch die INI-Dateien angepasst werden. Hier muss darauf geachtet werden, wo das MMDVMHost Log abgelegt wird.

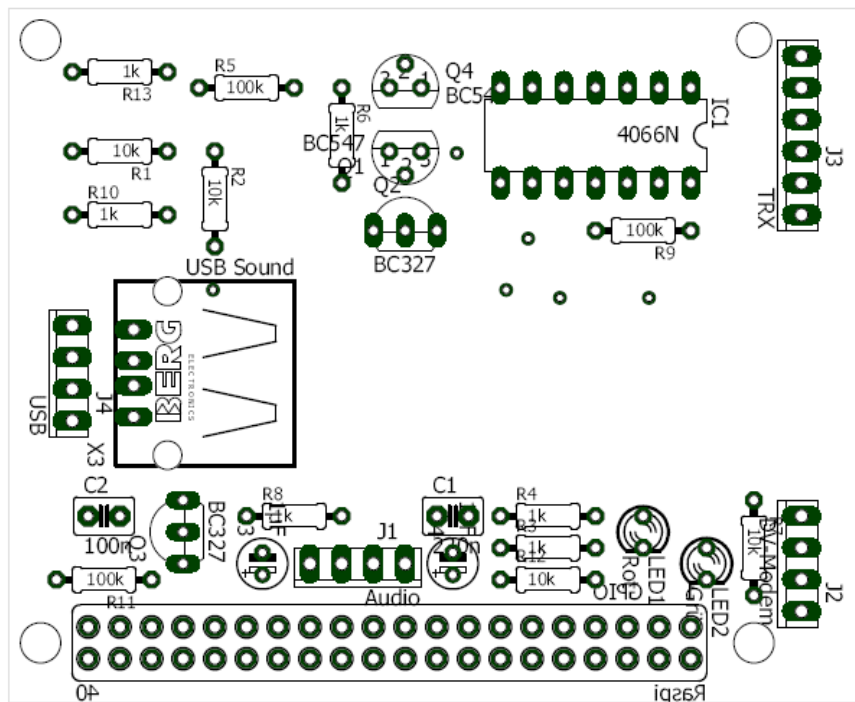
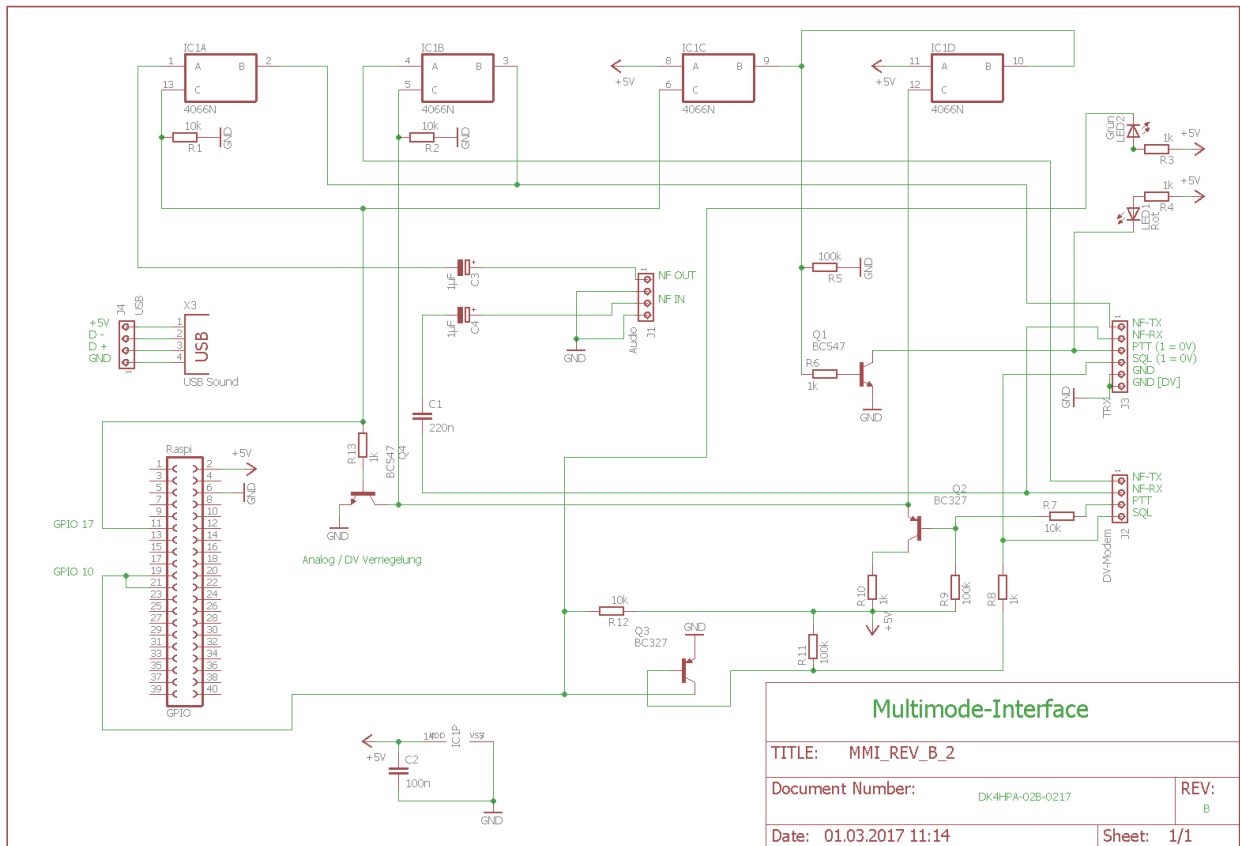
Noch ein kleiner Hinweis, das Ganze Projekt ist nur als Vorschlag zu verstehen. Eine kommerzielle Nutzung der Schaltung bzw. Veröffentlichung ist ohne meine Zustimmung jedoch ausdrücklich untersagt.

2. Blockschaltbild der Steuerung



Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

3. Das Multi-Mode-Interface



Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

Stückliste

MMI_REV_B

Part	Value	Device	Package	Library
C1	220n	C2,5-3	C2.5-3	capacitor-wima
C2	100n	C2,5-3	C2.5-3	capacitor-wima
C3	1µF	CPOL-EUE1.8-4	E1,8-4	rcl
C4	1µF	CPOL-EUE1.8-4	E1,8-4	rcl
GPIO	Raspi	FE20-2	FE20-2	con-ista
IC1	4066N	4066N	DIL14	40xx
J1	Audio	MTA04-100	10X04MTA	con-amp
J2	DV-Modem	MTA04-100	10X04MTA	con-amp
J3	TRX	MTA06-100	10X06MTA	con-amp
J4	USB	MTA04-100	10X04MTA	con-amp
LED1	Rot	LED3MM	LED3MM	led
LED2	Grün	LED3MM	LED3MM	led
Q1	BC547	BC547	TO92	transistor-npn
Q2	BC327	BC327	TO92-EBC	transistor-pnp
Q3	BC327	BC327	TO92-EBC	transistor-pnp
Q4	BC547	BC547	TO92	transistor-npn
R1	10k	R-EU_0204/7	0204/7	rcl
R2	10k	R-EU_0204/7	0204/7	rcl
R3	1k	R-EU_0204/7	0204/7	rcl
R4	1k	R-EU_0204/7	0204/7	rcl
R5	100k	R-EU_0204/7	0204/7	rcl
R6	1k	R-EU_0204/7	0204/7	rcl
R7	10k	R-EU_0204/7	0204/7	rcl
R8	1k	R-EU_0204/7	0204/7	rcl
R9	100k	R-EU_0204/7	0204/7	rcl
R10	1k	R-EU_0204/7	0204/7	rcl
R11	100k	R-EU_0204/7	0204/7	rcl
R12	10k	R-EU_0204/7	0204/7	rcl
R13	1k	R-EU_0204/7	0204/7	rcl
X3	USB Sound	PN87520	PN87520	con-berg

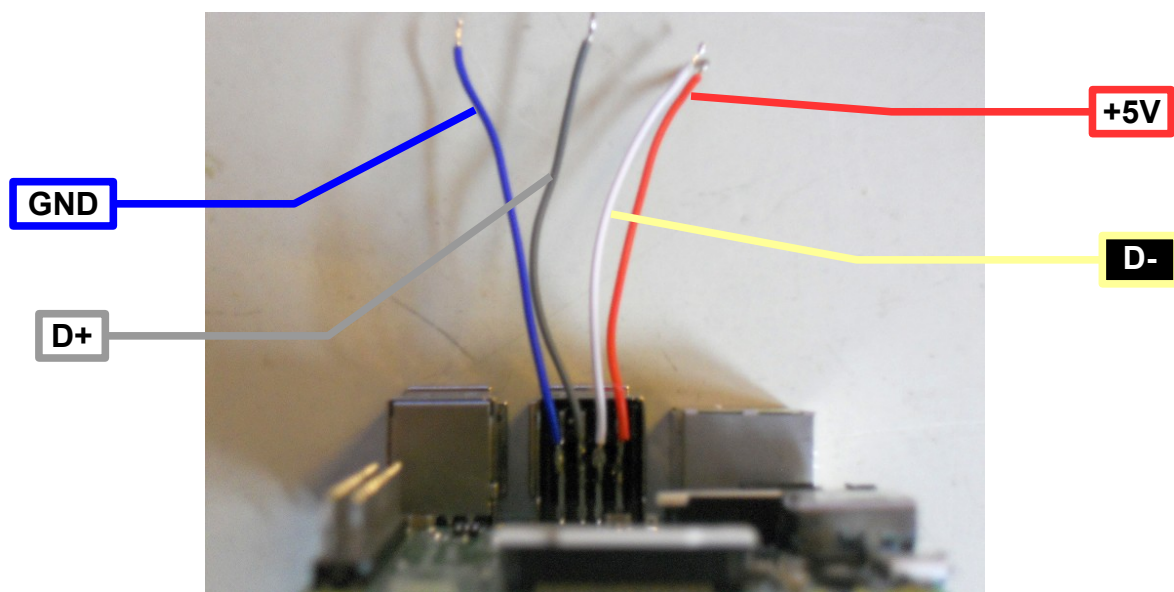
Einfache USB Soundkarte (z.B. 3D Sound)

Modifikation zum Betreiben eines Multimode-Repeater mit SvxDLink und MMDVM

Modifikation des Raspberry Pi für den USB Soundkartenanschluss

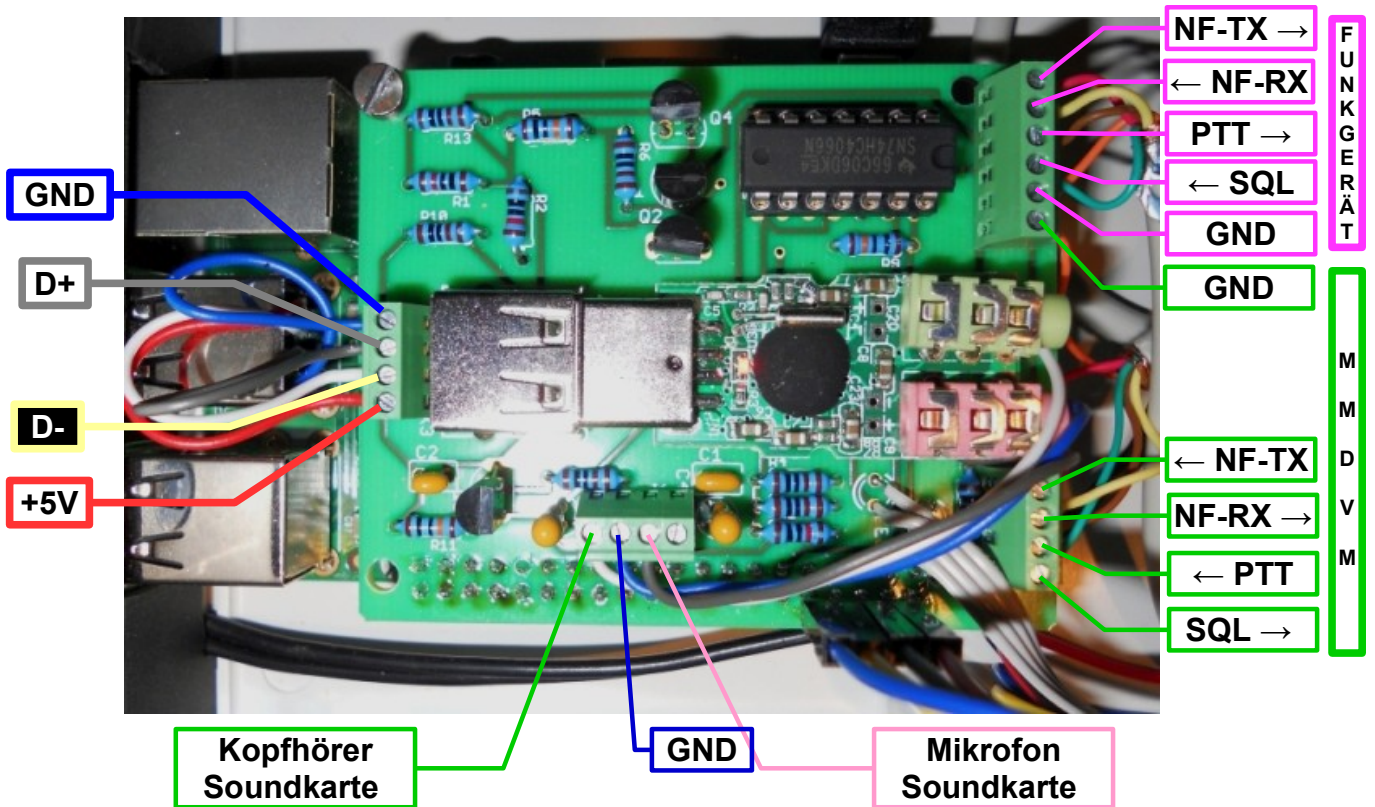


Entfernen des Abschirmdeckels aus der Rückseite des USB-Anschlusses



Löten der Anschlussleitung für die MMI-Platine

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM



An Stelle eines Soundchips für die Audiodigitalisierung, für das Programm SvxLink, wurde hier die USB-Buchse zur Aufnahme einer USB-Soundkarte vorgesehen. Die Soundkarte ist wesentlich Preiswerter (ca. 2 bis 3 Euro) als ein Soundchip. Somit fallen für die Bauteile incl. USB-Soundkarte Kosten von ca. 15 Euro an.

Die kleine Aufsteck-Platine für den Raspberry Pi beinhaltet einen Audioschalter 4066N, vier Transistoren für die Steuerung und einen Steckplatz für die USB-Soundkarte. Der Audioschalter sorgt für die Durchschaltung der Modulation-Signale für die analoge Aussendung (SvxLink) und die digitale Aussendung (MMDVM).

```

Links Datei Befehl Optionen Rechts
<- /home/pi .[^> <- /home/pi/.config .[^>
.n Name Größe Modifikations .n Name Größe Modifikations
/.. ÜBERVZ. 04. Mär 2016 /.. ÜBERVZ. 02. Mär 22:17
/.cache 4096 06. Mär 18:17 /autostart 4096 02. Mär 16:59
/.config 4096 31. Okt 19:46 /dconf 4096 04. Sep 2016
/.cpan 4096 21. Feb 18:09 /enchant 4096 03. Mär 2016
/.dbus 4096 26. Feb 2016 /epiphany 4096 10. Mär 2016
/.fontconfig 4096 26. Feb 2016 /filezilla 4096 04. Mär 2016
/.gconf 4096 06. Mär 18:17 /geany 4096 30. Okt 11:58
/.gstreamer-0.10 4096 26. Feb 2016 /gtk-2.0 4096 31. Okt 19:48
/.local 4096 26. Feb 2016 /gtk-3.0 4096 26. Feb 2016
/.oracle~e_usage 4096 05. Mär 2016 /leafpad 4096 05. Mär 2016
/.themes 4096 26. Feb 2016 /libfm 4096 04. Mär 2016
/.thumbnails 4096 04. Mär 2016 /libreoffice 4096 05. Mär 2016
/.vnc 4096 27. Aug 2016 /lxpanel 4096 26. Feb 2016
/Desktop 4096 08. Okt 12:57 /lxsession 4096 26. Feb 2016
/Documents 4096 09. Feb 12:26 /lxterminal 4096 03. Mär 2016

ÜBERVZ. ÜBERVZ.
3244M/7338M (44%) 3244M/7338M (44%)
Hint: Do you want lynx-style navigation? Set it in the Configuration dialog.
root@DB0HEW:/home/pi#
1|ilfe 2|Menü 3|Ans-ht 4|Bea-en 5|Kop-en 6|Ver-en 7|Mkdir 8|Lös-en 9|Menüs 10|Bee-er
    
```

Die analogen Audiosignale werden mit der USB-Soundkarte verbunden. Die Erfahrung hat gezeigt, dass die Signale per Lötverbindung von der USB-Soundkarte entnommen werden sollten und nicht mit den Klinkesteckern. Mit der Klemmleiste J1 werden die Signale der Soundkarte fest verbunden.

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

Mit der Klemmleiste J4 werden die USB-Signale direkt von einer modifizierten USB-Buchse vom Raspberry Pi verbunden. Dieses sorgt für einen kompakten Aufbau.

An der Klemmleiste J2 werden die Signale NF-TX, NF-RX, PTT und SQL mit dem MMDVM verbunden.

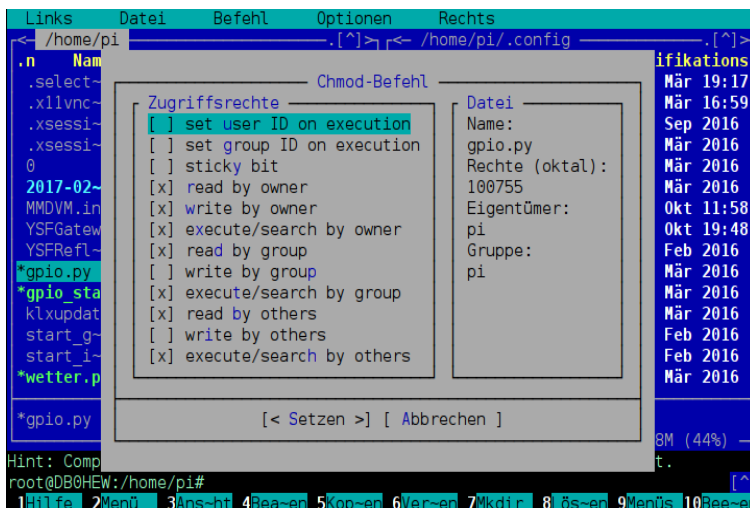
An der Klemmleiste J3 wird eine Verbindung zur Funktechnik für die Signale NF-TX, NF-RX, PTT und SQL hergestellt. Zusätzlich sind zwei Klemmen für die GND-Verbindungen der Funktechnik und dem MMDVM vorgesehen.

Der Transistor Q1 steuert die PTT des Funkgerätes. Als Signal wird hier 0V durchgeschaltet, was im allgemeinen bei vielen Geräten der Fall sein sollte. Der Q2 sorgt für eine Pegelwandlung vom MMDVM um den Anlogschalter entsprechend ansteuern zu können. Q4 sorgt für einen Vorrang der analogen Seite. Sollte vom MMDVM nur Netzbetrieb übertragen werden und die analoge Bake läuft, so wird der Eingang PIN 5 vom 4066N blockiert. Somit wird das Gehör der analogen Teilnehmer nicht strappaziert.

Q3 sorgt für die Steuerung des SQL-Signals für die SvxLink-Software. Sollte das SQL-Signal 0V betragen, so muss in der svxlink.conf das Signal invertiert werden (siehe Anpassen der svxlink.conf Datei Seite 16).

Für die Funktion muss einiges an Programmierarbeit durchgeführt werden. Unter Linux lässt es sich am besten mit dem Programmteil MC arbeiten. Sollten Scripte unter Windows erzeugt werden, so ist auf das CR am Ende der Zeilen zu achten. Hier kann es vorkommen, dass die Scripte nicht funktionieren.

Ob man nun Nano oder MC benutzt ist jedem selbst überlassen. Beim Aufruf des entsprechenden Editors habe ich stets mit Root Rechten gearbeitet (z.B. ~ sudo mc).



```
Links Datei Befehl Optionen Rechts
~/home/pi .[^> |< /home/pi/.config .[^>
.n Nam
.select
.x11vnc~
.xsessi~
.xsessi~
0
2017-02~
MMDVM.in
YSFGatew
YSFRefl~
*gpio.py
*gpio_sta
klxupdat
start_g~
start_i~
*wetter.p
*gpio.py
Chmod-Befehl
Zugriffsrechte
[ ] set user ID on execution
[ ] set group ID on execution
[ ] sticky bit
[x] read by owner
[x] write by owner
[x] execute/search by owner
[x] read by group
[ ] write by group
[x] execute/search by group
[x] read by others
[ ] write by others
[x] execute/search by others
Datei
Name:
gpio.py
Rechte (oktal):
100755
Eigentümer:
pi
Gruppe:
pi
Mär 19:17
Mär 16:59
Sep 2016
Mär 2016
Mär 2016
Mär 2016
Okt 11:58
Okt 19:48
Feb 2016
Mär 2016
Mär 2016
Mär 2016
Feb 2016
Feb 2016
Mär 2016
BM (44%)
t.
Hint: Comp
root@DB0HEW:/home/pi#
1 Hilfe 2 Menü 3 Ans-ht 4 Bea-en 5 Kop-en 6 Ver-en 7 Mkdir 8 Lös-en 9 Menüs 10 Bee-en
```

Außerdem muss auf die Rechtevergabe geachtet werden. Diese lässt sich am einfachsten mit mc überprüfen und anpassen.

Die Pfade werden je nach Installation oder verwendetes Image angelegt. Die kleine Pfadauswahl dient also nur als Beispiel.

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

/home/pi

- dstar
- etc
- home
 - pi
 - Desktop
 - Documents
 - Downloads
 - dv4mini
 - MMDVM_app
 - perl5
 - svxlink
 - test
 - wiringPi

Name	Größe	Geändert	Rechte	Besitzer
wiringPi		03.03.2016 08:33:30	rw-r--r--	pi
0	0 KB	02.03.2017 08:18:07	rw-r--r--	root
2017-02-11-123255_1364x76...	140 KB	11.02.2017 12:32:55	rw-r--r--	pi
gpio.py	1 KB	06.02.2017 20:45:39	rw-r--r--	pi
gpio_start.sh	1 KB	10.02.2017 13:43:14	rw-r--r--	pi
kxupdate	8 KB	25.08.2015 15:56:14	rw-r--r--	pi
MMDVM.ini	3 KB	28.02.2017 20:20:39	rw-r--r--	root
wetter.pl	1 KB	21.02.2017 17:08:04	rw-r--r--	pi
YSFGateway.ini	1 KB	20.02.2017 16:19:27	rw-r--r--	root
YSFReflector.ini	1 KB	16.10.2016 17:48:03	rw-r--r--	root

/etc/svxlink

- sgml
- skel
- sound
- ssh
- ssl
- subversion
- sudoers.d
- svxlink
 - svxlink.d

Name	Größe	Geändert	Rechte	Besitzer
..		02.03.2017 17:00:41	rw-r--r--	root
svxlink.d		27.08.2016 21:19:36	rw-r--r--	pi
remotetrx.conf	4 KB	27.08.2016 20:59:07	rw-r--r--	root
svxlink.conf	6 KB	22.02.2017 13:31:40	rw-r--r--	pi
TclVoiceMail.conf	2 KB	27.08.2016 20:59:06	rw-r--r--	root

/usr/share/svxlink/events.d/local

- srv
- sys
- tmp
- usr
 - share
 - svxlink
 - events.d
 - local
- var

Name	Größe	Geändert	Rechte	Besitzer
..		22.02.2017 10:07:30	rw-r--r--	pi
checkdvstart.sh	1 KB	18.12.2016 14:27:23	rw-r--r--	pi
logbuch_delete.sh	1 KB	12.10.2016 14:23:27	rw-r--r--	pi
Logic.tcl	17 KB	22.02.2017 13:31:28	rw-r--r--	pi
RepeaterLogic.tcl	10 KB	05.11.2016 15:03:41	rw-r--r--	pi
SimplexLogic.tcl	7 KB	19.02.2017 11:22:16	rw-r--r--	pi
svxlink_start_stop.sh	2 KB	06.02.2017 20:51:37	rw-r--r--	pi

/media/DVrepeater

- etc
- home
- lib
- lost+found
- media
 - DVrepeater
 - pi
 - SvxLog
- mnt

Name	Größe	Geändert	Rechte	Besitzer
..		29.08.2016 19:37:29	rw-rw-rw-	root
abfrage.txt	1 KB	07.03.2017 12:53:01	rw-rw-rw-	root
Headers.log	345 KB	07.03.2017 12:53:20	rw-r--r--	root
ircDDBGateway-2016-10-10.log	4 KB	10.10.2016 22:17:21	rw-r--r--	root
ircDDBGateway-2016-10-11....	88 KB	11.10.2016 23:27:11	rw-r--r--	root

Hinweis:

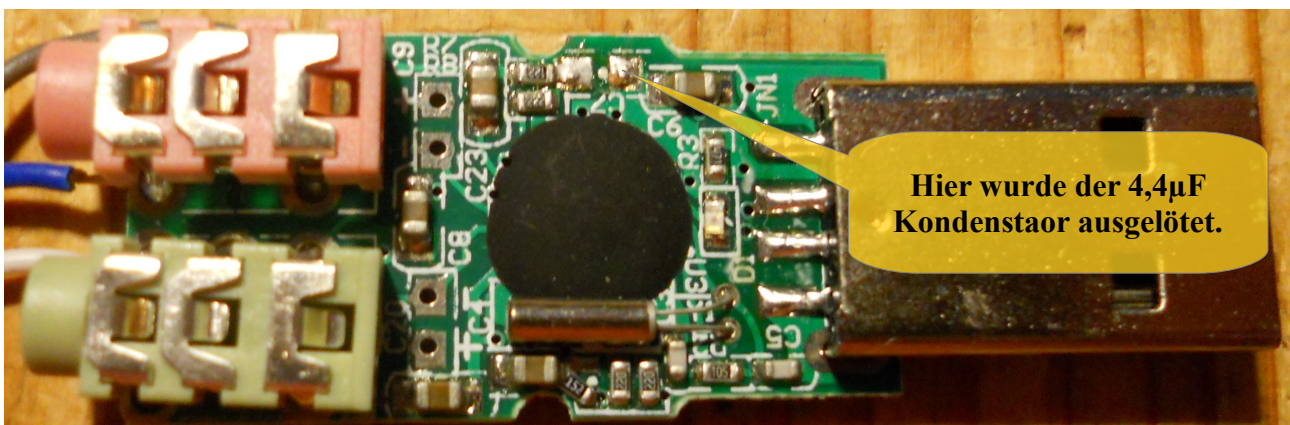
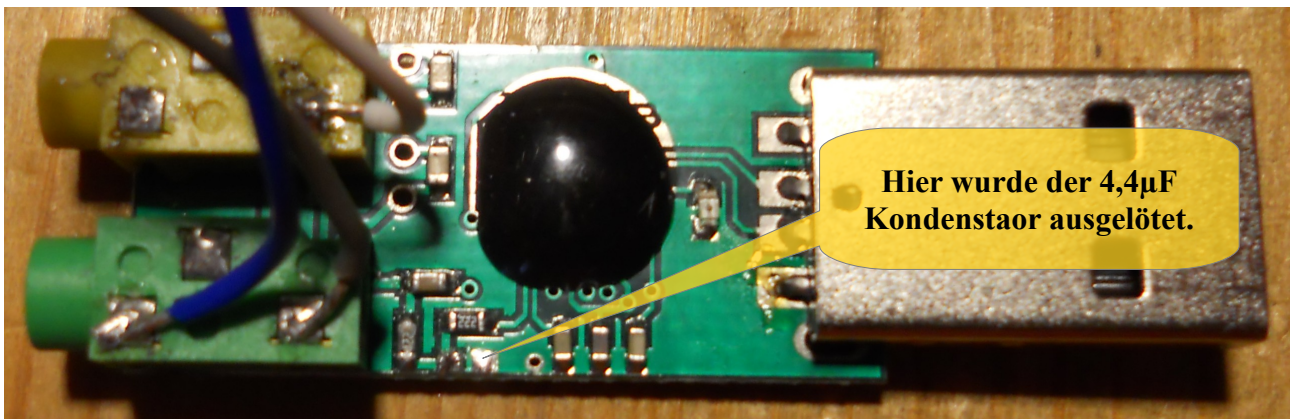
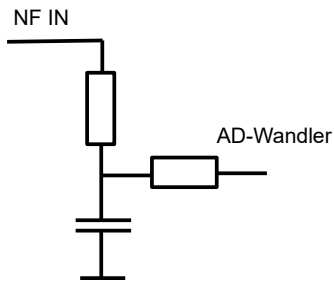
Nun folgt das Ändern und Einrichten der verschiedenen Scripte. Alles was fett gedruckt ist, muss eingefügt oder erstellt werden. Bei den Zeilen die eingefügt werden sollen, ist der vorangestellte Text und der Nachfolgende kursiv dargestellt.

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

4. Wichtiger Hinweis zur USB-Soundkarte

Einige Soundkarten besitzen im Mikrofoneingang einen passiven „Tiefpassfilter“. Dieser hat leider einen geringen Einfluss auf das zu verarbeitende NF-Signal vom RX. Die analoge Verarbeitung wird nicht beeinträchtigt. Da aber der Eingang vom MMDVM das Signal auch verarbeiten muss, kommt es bei DMR zu einem Decodierproblem bei dem Duplexbetrieb mit dem DR1-X von Yaesu. Nach dem Tausch einer Soundkarte ohne „Tiefpass“ war die Funktion von DMR einwandfrei vorhanden. Wenn man sich die Mühe macht, kann der entsprechende Kondensator von der USB-Soundkarte entfernt werden.

Nach einem Test wird die analoge Seite weiterhin voll bedient und DMR läuft. Die Betriebsarten D-Star und C4FM wurden durch den „Tiefpass“ nicht beeinflusst.



Zwei Beispiel für das entfernen des Kondensators.

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

5. SvxLink bei digitalem Signal auf der Eingabe absteuern

Hierzu wird ein Python-Script benötigt um an Hand von dem SQL-Signal via GPIO9 ein Interupt auszuführen.

Hier nun das Script gpio.py

Diese Script liegt unter /home/pi

```
import time
import RPi.GPIO as GPIO
import os

# Abfrage ob DV-Betrieb via Repeater
def Interrupt(channel):
    time.sleep(3.0)
    os.system("sudo/usr/share/svxlink/events.d/local/svxlink_start_stop.sh")

# RPi.GPIO Layout verwenden (wie Pin-Nummern)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(21, GPIO.IN)
# Interrupt ausfuehren
GPIO.add_event_detect(21, GPIO.FALLING, callback = Interrupt)

# Dauerschleife
while True:
    # Warte 100 ms
    time.sleep(0.01)
```

6. Nun wird ein Shell-Script benötigt welches die Logdatei vom MMDVM-Host prüft

Pfade die mit XXXXX gekennzeichnet sind müssen natürlich angepasst werden. Eine leere Datei **abfrage.txt** und **log.txt** müssen im ausgewählten Pfad angelegt werden. Die Datei svxlink_start_stop.sh habe ich unter /usr/share/svxlink/events.d/local abgelegt.

Hier nun das Script svxlink_start_stop.sh

```
#!/bin/bash

# Startet und stopt SvxLink durch auswerten des MMDVM Logfiles

# Variablen festlegen

PIDMMH=$(pidof MMDVMHost)

if [ $PIDMMH > "0" ]; then
    LOGFILE=$(date +" /XXXXX/MMDVM-%Y-%m-%d.log")
    PIDFILE=$(pidof svxlink)
    ZEITABLAGE="/XXXXX/abfrage.txt"
    ALT=$(tail -n 1 $ZEITABLAGE)
    LOGDAT="/XXXXX/log.txt"
    DIFF="180"
    ABFRAGE=$((date +%s) - $DIFF)
    RF1="0"
    RF2="0"
```

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

```
# Prüfen ob DV-Betrieb auf der Eingabeseite vorhanden ist
# und SvxLink stoppen.

RF1=$(tail -n 10 $LOGFILE | grep -c "DMRslotRX")
RF2=$(tail -n 10 $LOGFILE | grep -c "RF")
if [ $RF1 -ne "0" -o $RF2 -ne "0" ]; then
    if [ "$(tail -n 1 $LOGDAT)" != "$(tail -n 1 $LOGFILE)" ]; then
        if [ $PIDFILE > "0" ]; then
            sudo systemctl stop svxlink.service
        fi
        stat -c %Z $LOGFILE>$ZEITABLAGELAGE
        ALT=$(tail -n 1 $ZEITABLAGELAGE)
        echo "STOP SvxLink"
    fi
    tail -n 1 $LOGFILE>$LOGDAT
fi

# Wenn kein RF DV-Betrieb nach einer Zeit (DIFF) SvxLink starten.

if [ $ALT -lt $ABFRAGE ]; then
    stat -c %Z $LOGFILE>$ZEITABLAGELAGE
    if [ $PIDFILE > "0" ]; then
        echo "SvxLink läuft"
    else
        sudo systemctl start svxlink.service
        echo "START SvxLink"
    fi
fi
fi
```

7. Nun fehlt noch das Einrichten der GPIO's für SvyLink mit einem weiteren Shell-Script

Die Datei trägt den Namen gpio_start.sh und liegt unter /home/pi.

```
#!/bin/bash

# GPIO anlegen für SvxLink

echo "10" > /sys/class/gpio/export
sudo chmod o+rw /sys/class/gpio/gpio10/direction
sudo chmod o+rw /sys/class/gpio/gpio10/active_low
sudo chmod o+rw /sys/class/gpio/gpio10/value
echo "in" > /sys/class/gpio/gpio10/direction
echo "1" > /sys/class/gpio/gpio10/active_low

echo "17" > /sys/class/gpio/export
sudo chmod o+rw /sys/class/gpio/gpio17/direction
sudo chmod o+rw /sys/class/gpio/gpio17/value
echo "out" > /sys/class/gpio/gpio17/direction
```

8. Die erstellten Files müssen beim starten von Linux aufgerufen werden

In der Datei /etc/rc.local am Ende folgendes einfügen:

```
sudo /home/pi/gpio_start.sh
sudo python /home/pi/gpio.py
```

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

9. Unter crontab muss noch folgender Eintrag hinzugefügt werden

Aufruf mit sudo crontab -e und am Ende folgendes eintragen:

```
*/1 * * * * /usr/share/svxlink/events.d/local/svxlink_start_stop.sh >/dev/null 2>&
```

10. Datei Logik.tcl von SvxLink modifizieren

/usr/share/svxlink/events.d/Logik.tcl nach /usr/share/svxlink/events.d/local kopieren
Bei einem update von SvxLink werden die Dateien im Pfad local nicht überschrieben und die Anpassungen bleiben erhalten.

Folgende Zeilen ergänzen und einfügen:

```
..... Ab Zeile ~123
#
# Executed when a short identification should be sent
# hour - The hour on which this identification occur
# minute - The hour on which this identification occur
#
proc send_short_ident {{hour -1} {minute -1}} {
    global mycall;
    variable CFG_TYPE;
    global tx_on;

# Abfrage ob DV-Betrieb via Netz vorhanden?

    if {$tx_on == "0"} {
        set datum [clock format [clock seconds] -format "-%Y-%m-%d.log"];
        set zeit [clock seconds];
        set a [file mtime "/XXXXX/MMDVM$datum"];
        set b 0;
        set b [expr {$zeit-$a}];
        if {$b < "120"} {
            puts "DV-Betrieb / keine Bake";
            return;
        }

        spellWord $mycall;
        if {$CFG_TYPE #== "Repeater"} {
            playMsg "Core" "repeater";
        }
    }
}
.....

..... Ab Zeile ~141
#
# Executed when a long identification (e.g. hourly) should be sent
# hour - The hour on which this identification occur
# minute - The hour on which this identification occur
#
proc send_long_ident {hour minute} {
    global mycall;
    global loaded_modules;
    global active_module;
    variable CFG_TYPE;
    global tx_on;

# Abfrage ob DV-Betrieb via Netz vorhanden?
```

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

```
if {$tx_on == "0"} {
    set datum [clock format [clock seconds] -format "-%Y-%m-%d.log"];
    set zeit [clock seconds];
    set a [file mtime "/XXXXXX/MMDVM$datum"];
    set b 0;
    set b [expr {$zeit-$a}];
    if {$b < "120"} {
        puts "DV-Betrieb / keine Bake";
        return;
    }
    spellWord $mycall;
    if {$CFG_TYPE == "Repeater"} {
        playMsg "Core" "repeater";
    }
}
.... Ab Zeile ~335
#
# Executed each time the squelch is opened or closed
# rx_id - The ID of the RX that the squelch opened/closed on
# is_open - Set to 1 if the squelch is open or 0 if it's closed
#
proc squelch_open {rx_id is_open} {
    variable sql_rx_id;
#
# -----MMDVM Modifikation-----
#
    global sql_time_div;
    global dvdowntime;
    variable akttime;

    set akttime [clock seconds];

    puts "The squelch is $is_open on RX $rx_id";
    if {$is_open } {
        set sql_time_div $akttime;
    }
    if {$is_open == 0 && ([clock seconds] - $sql_time_div > 30)} {
        set dvdowntime [expr [clock second]+120];
    }

    set sql_rx_id $rx_id;
}
....

.... Ab Zeile ~403
#
# Executed once every whole minute. Don't put any code here directly
# Create a new function and add it to the timer tick subscriber list
# by using the function addTimerTickSubscriber.
#
proc every_minute {} {
    variable timer_tick_subscribers;
#
# -----MMDVM-Modifikation-----
#
    global dvmute;
    global dvdowntime;
    global tx_on;
    variable akttime;
```

Modifikation zum Betreiben eines Multimode-Repeater mit SvxDLink und MMDVM

```
set aktttime [clock seconds];

if {$dvdmute == 1 } {
  if {$tx_on == 1 } {
    set dvddowntime [expr [clock second]+60];
  } else {
    if {[clock second] > $dvddowntime } {
      set dvdmute 0;
      exec sudo systemctl start mmdvmhost.service &;
    }
  }
}

#puts [clock format [clock seconds] -format "%Y-%m-%d %H:%M:%S"];
foreach subscriber $timer_tick_subscribers {
  $subscriber;
}
}.....
```

Fertig und speichern nicht vergessen....

11. Die RepeaterLogik.tcl von SvxDLink modifizieren

/usr/share/svxdlink/events.d/RepeaterLogik.tcl nach /usr/share/svxdlink/events.d/local kopieren

Folgende Zeilen ergänzen und einfügen:

```
..... Ab Zeile ~28
#
# Executed when the SvxDLink software is started
#
proc startup {} {
  global logic_name;
#
# -----MMDVM Modifikation-----
#
  global dvddowntime;
  global dvdmute;
  global tx_on;
  set dvddowntime 0;
  set dvdmute 0;
  set tx_on 0;

  append func $logic_name "::checkPeriodicIdentify";
  Logic::addTimerTickSubscriber $func;
  Logic::startup;
}
.....
..... Ab Zeile ~169
#
# Executed each time the transmitter is turned on or off
#
proc transmit {is_on} {
#
# MMDVM-Modifikation
#
  set tx_on $is_on;
}
```

Modifikation zum Betreiben eines Multimode-Repeater mit SvxLink und MMDVM

```
Logic::transmit $is_on;
}
.....
..... Ab Zeile ~193
#
# Executed when the repeater is activated
# reason - The reason why the repeater was activated
#         SQL_CLOSE      - Open on squelch, close flank
#         SQL_OPEN       - Open on squelch, open flank
#         CTCSS_CLOSE    - Open on CTCSS, squelch close flank
#         CTCSS_OPEN     - Open on CTCSS, squelch open flank
#         TONE           - Open on tone burst (always on squelch close)
#         DTMF           - Open on DTMF digit (always on squelch close)
#         MODULE         - Open on module activation
#         AUDIO          - Open on incoming audio (module or logic linking)
#         SQL_RPT_REOPEN - Reopen on squelch after repeater down
#
proc repeater_up {reason} {
    global mycall;
    global active_module;
    variable repeater_is_up;
#
#         -----MMDVM-Modifikation-----
#
    variable uptime;
    global dvmute;
    global downtime;

    set repeater_is_up 1;

    set uptime [clock seconds];
    set downtime $uptime;

    if { $dvmute == 0 } {
        set dvmute 1;
        set downtime [expr $uptime+60];
        exec sudo systemctl stop mmdvmhost.service &
        puts "MMDVM Stop $uptime $downtime";
    }

    if {($reason != "SQL_OPEN") && ($reason != "CTCSS_OPEN") &&
        ($reason != "SQL_RPT_REOPEN")} {
        set now [clock seconds];
        if {$now-$Logic::prev_ident < $Logic::min_time_between_ident} {
            return;
        }
    }
.....
```

Fertig und speichern nicht vergessen....

12. Anpassen der svxlink.conf Datei

Die Datei befindet sich im Verzeichnis /etc/svxlink und muss für die eigenen Bedürfnisse angepasst werden. Um die die GPIO's zu nutzen, muss das hier berücksichtigt werden.

```
..... Ab Zeile ~175
[Rx1]
TYPE=Local
AUDIO_DEV=alsa:plughw:0
AUDIO_CHANNEL=0
#SQL_DET=CTCSS
#SQL_DET=SERIAL
SQL_DET=GPIO
.....
.....
#CTCSS_BPF_HIGH=270
#SERIAL_PORT=/dev/ttyUSB0
#SERIAL_PIN=CTS
GPIO_SQL_PIN=gpio10
#SERIAL_SET_PINS=DTR!RTS
#EVDEV_DEVNAME=/dev/input/by-id/usb-SYNIC_SYNIC_Wireless_Audio-event-
.....
```

Sollte das SQL 1-Signal nicht 0V sondern 12V betragen, so ist die Zeile
GPIO_SQL_PIN=gpio10 wie folgt zu invertieren: **GPIO_SQL_PIN=!gpio10**.

```
..... Ab Zeile ~230

[Tx1]
TYPE=Local
AUDIO_DEV=alsa:plughw:0
AUDIO_CHANNEL=1
#PTT_TYPE=SerialPin
#PTT_PORT=/dev/ttyUSB0
#PTT_PIN=RTS
PTT_TYPE=GPIO
PTT_PIN=gpio17
#SERIAL_SET_PINS=DTR!RTS
#PTT_HANGTIME=1000
TIMEOUT=7200

.....
```

Die Installationen der Software auf den Raspberry PI sind entsprechend den Anleitungen der jeweiligen Autoren zu entnehmen.

Viel Spaß beim Programmieren und Basteln wünscht

*Peter
DK4HPA*

E-Mail: dk4hpa@darcd.de