

# SvxLink-PTT

Ausgabe:  
05.07.2025

Dieses Dokument wurde erzeugt mit  
BlueSpice

Seite von

## SvxLink-PTT

Um Svxlink mit einem Transceiver zu verbinden ist neben den Audiosignalen (RX, TX) auch die Sende-Empfangsumschaltung (PTT) notwendig. Der Squelch wird üblicherweise in Software erkannt (CTCSS).

Es gibt mehrere Möglichkeiten die PTT zu verbinden:

### GPIO

Auf Raspberrys ist die einfachste Variante die Verwendung eines GPIO-Pins. Typischerweise wird bei der PTT ein Pin gegen des Funkgeräts gegen Masse geschaltet. Ein Transistor mit Basis über einen Spannungsteiler am GPIO und Open-Collector am PTT-Pin des Funkgeräts ist eine einfache Lösung.

Hier ein kurzer Test-Code:

```
import RPi.GPIO as GPIO
import time

# Set up the GPIO pin for PTT
PTT_PIN = 17 # Change this to your pin number
GPIO.setmode(GPIO.BCM)
GPIO.setup(PTT_PIN, GPIO.OUT)

# Function for PTT
def ptt_on():
    GPIO.output(PTT_PIN, GPIO.LOW) # Pulling to LOW to key the radio

def ptt_off():
    GPIO.output(PTT_PIN, GPIO.HIGH) # Releasing to HIGH to unkey

try:
    while True:
        ptt_on() # Key the radio
        time.sleep(5) # Duration for which to key
        ptt_off() # Unkey the radio
        time.sleep(5) # Pause before the next key

except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```

### USB\-\-Adapter

Sofern ein gewöhnlicher PC verwendet wird, kann eine über USB hinzugefügte serielle Schnittstelle als PTT verwendet werden. Dabei wird die Handshake-Leitung "RTS" (Request to Send) verwendet. RTS ist Pin 7 am 9-poligen SubD-Stecker. Masse (GND) findet sich auf Pin 5. Die Verbindung zum

Funkgerät erfolgt wie bei der Verwendung eines GPIO-Pins. RTS wechselt bei USB-Serial-Adapttern typischerweise zwischen 0 V ("off") und 7 V ("on").

Hier ein kurzer Test-Code:

```
#!/usr/bin/python3

# pip install pyserial

import serial
import time

# Configure your serial port and baud rate
serial_port = '/dev/ttyUSB0' # Change this to your serial port
baud_rate = 9600

# Open the serial port
with serial.Serial(serial_port, baudrate=baud_rate, timeout=1) as ser:
    try:
        while True:
            # Set RTS (key the radio)
            ser.setRTS(True)
            print("PTT ON")
            time.sleep(5) # Keep PTT on for 5 seconds

            # Clear RTS (unkey the radio)
            ser.setRTS(False)
            print("PTT OFF")
            time.sleep(5) # Wait for 5 seconds before the next key

    except KeyboardInterrupt:
        print("\nStopped by User")
```

### **PTT-Pins auf USB-Sound-Karten**

Manche Sound-Chips (z.B. CM108) haben unbenutzte GPIO-Pins, welche als PTT verwendet werden können. Allerdings ist es dazu -abhängig vom Leiterplatten-Design -notwendig, einen Draht direkt am Sound-Chip anzulöten.