SvxLink-UDP

svxlink erlaubt die Anbindung von Audio (sowohl als "Mic", wie als "Lautsprecher") über UDP.

Dazu ist etwa folgender Abschnitt in der Konfiguration von svxlink notwendig:

```
[TxUDP]
TX_ID=TU
TYPE=Local
AUDIO_DEV=udp:127.0.0.1:4300
AUDIO_CHANNEL=0
PTT_TYPE=NONE
CARDS_CHANNELS =1
```

Die Daten werden mit zwei Kanälen mit jeweils 16 Bits kodiert. Pro Sekunde werden 48 UDP-Pakete mit einer Länge von 4032 Bytes übermittelt, dh. je Paket 1008 Samples mit einer Sample-Rate von hier ca 48.000 Samples/s, der wohl internen Sample-Rate von svxlink.

```
[2025-03-10 21:12:45.603] Received 4032 bytes from ('127.0.0.1', 53786):
8eff00006eff000037ff0000fafe0000befe000099fe000094fe0000a7fe0000cefe0000fbfe00
001eff000034ff00003cff00003dff000044ff00005bff000086ff0000c2ff0000000000003500
0000510000004800000001f000000deff000094ff00005aff00003fff00004cff000089ff0000e9
ff00005b000000d90000004e010000af010000f80100002602000039020000370200002202000 0
ff010000d5010000a90100008101000063010000500100004901000048010000480100004101 00
00310100001501000 0f1000000ce000000b3000000a6000000a4000000ab000000b4000000a800
0000840000004600000 0e6ff000077ff00000cff0000b0fe000079fe000072fe000095fe0000dc
fe00003dff0000a0ff0000faff000043000000730000008c0000009700000092000000860000 00
730000005400000028000000efff0000a4ff00004fff0000f9fe0000acfe000075fe00005bfe00
0067fe000095fe0000d8fe00002dff000080ff0000c2ff0000eeff000000000000f9ff0000e2ff
0000c5ff0000abff000096ff00008aff000085ff00007fff000076ff00006cff000065ff00006b
ff000086ff0000c4ff00001e00000900000000f01000083010000d6010000fe010000f1010000
b60100005a010000ef000000870000002f000000ebff0000b5ff000084ff000050ff000010ff00
00c9fe000084fe00004afe00002efe000034fe00005afe000097fe0000d8fe00000bff00001cff
000007ff0000d2fe00007ffe000028fe0000e1fd0000b1fd0000a3fd0000b7fd0000e3fd000020
fe000065fe0000affe0000fffe000053ff0000b6ff000022000000920000000050100006b010000
b7010000ec01000002020000fe010000ee010000d9010000c6010000c0010000c4010000c90100
00d7010000e1010000e2010000e7010000ec010000f50100000b0200002c020000520200007a02
000099020000ab020000a702000091020000710200004302000017020000f3010000ce010000ac
0100008901000058010000180100 00ce000000760000002000000 0dbff0000a6ff00008bff0000
8bff000094ff00009fff0000a2ff000087ff000054ff000011ff0000befe00006ffe000037fe00
000efe0000fbfd0000f8fd0000eefd0000d7fd0000abfd000064fd000010fd0000befc00007dfc
00005afc00005bfc00007afc0000abfc0000dcfc000000fd00000efd000006fd0000f4fc0000e6
fc0000e9fc00000cfd00004dfd0000a0fd000004fe000064fe0000b3fe0000f8fe000032ff0000
66ff0000aeff000080000007200000 0f70000007c010000ef010000540200009702000 0b50200
00c5020000c5020000bd020000c2020000cc020000d3020000e2020000e6020000da020000c802
0000aa020000840200005d020000340200000702000 0d8010000a30100006c010000300100 0f8
000000ca000000a00000007f000000670000004200000014000000dcff00008cff000036ff0000
e8fe00009ffe000072fe00006afe000076fe00009bfe0000d1fe000001ff00002cff000050ff00
0066ff00007bff000097ff0000b9ff0000e9ff000023000000600000009a000000cb000000eb00
0000fd000000020100000201000002010000050100001601000 02f0100004c0100007101000091
010000a6010000b3010000b0010000a001000087010000670100004401000023010000050100 00
ea000000cf000000b1000000910000006800000039000000 0a000000d9ff0000adff00008aff00
006dff000058ff000048ff000035ff000020ff00000bff0000effe0000d5fe0000c1fe0000acfe
00009afe000087fe00006afe000040fe00000bfe0000cbfd00008bfd000057fd000039fd00003b
fd000060fd0000a3fd0000f8fd000051fe0000a0fe0000d6fe0000edfe0000e6fe0000c4fe0000
8ffe000056fe00001efe0000eefd0000cefd0000bffd0000bcfd0000cefd0000edfd000014fe00
0048fe00007efe0000acfe0000dcfe000002ff00001bff000036ff000050ff00006aff00008eff
0000b6ff0000ddff0000020000001e00000290000002b00000250000001a0000001e00000033
```

```
000000590000009b000000ee00000047010000a7010000010200004c0200008a020000b6020000
cc020000d1020000c6020000aa02000850200005d0200003b020000260200002602000003d0200
006b020000a7020000ea020000280300005803000077030000810300007d030000720300006103
0000530300004d0300004103000033030000260300000703000de020000b3020000790200003d
02000006020000ca0100009301000066010000390100001201000f9000000dc000000c2000000
ac0000081000000490000000600000a8ff00003cff0000d0fe000059fe0000eafd00008afd00
0029fd0000cffc00007dfc000024fc0000cefb000084fb000043fb00001afb000010fb00001efb
000044fb000078fb0000a8fb0000cefb0000e3fb0000e4fb0000dafb0000cefb0000cafb0000d5
fb0000f0fb00001ffc000056fc000089fc0000c0fc0000f1fc000017fd00004dfd00008dfd0000
d6fd000041fe0000bcfe000038ff0000c5ff000046000000ad0000000a0100004a010000690100
007d0100007d0100006801000059010000460100002e01000290100002a0100002e0100004601
0000660100008701000b8010000ee0100002502000065020000a5020000de0200000c03000024
03000020030000f7020000ad0200004a020000d30100005c010000f7000000aa00000082000000
7f00000097000000c2000000f8000000290100005501000007f010000a3010000cc010000000200
003202000063020000 8d020000910200007002000027020000a90100000a01000062000000bfff
00003dff0000f2fe0000e6fe00001cff00008bff00001d000000c100000060010000e301000041
0200007702000080020000640200002f020000da0100007101000 0fa0000006b000000d2ff0000
3cff0000aefe00003dfe0000fbfd0000f0fd00001cfe000076fe0000eefe000067ff0000c8ff00
00feff000000000000d1ff000077ff000008ff00009bfe000030fe0000d3fd000082fd000029fd
0000c5fc000051fc0000cdfb000049fb0000d7fa000093fa00008afa0000c0fa00003dfb0000ea
fb0000adfc000071fd00001cfe00009afe0000e4fe0000fafe0000e2fe0000a7fe000056fe0000
f7fd0000a1fd00005cfd000031fd00003dfd000082fd000000fe0000cafe0000c8ff0000dc0000
000b020000240300000904000 0c30400003b0500007205000083050000740500005105000 02d05
000006050000db040000b10400007e040000440400001204 0000e8030000ce030000d9030000fe
030000360400007c040000b5040000d4040000bf04000073040000f70300003a0300005a020000
6d0100006500000063ff000073fe000083fd0000a7fc0000eefb000047fb0000cafa000084fa00
0064fa000076fa0000b4fa0000fafa000042fb00007efb00008bfb000076fb00004afb0000fdfa
0000b7fa00008cfa000078fa00008efa0000c8fa000019fb000078fb0000d8fb000040fc0000af
fc000024fd0000c2fd000081fe000056ff0000540000005901000044020000 1d030000c4030000
2c040000710400009204000096040000a7040000be040000d70400000c050000420500006d0500
009e050000c1050000d1050000e4050000ea050000e3050000d7050000b9050000840500003805
0000d30400005d040000d10300003e030000b30200001f020000920100001301000088000000ff
ff00007cff0000e7fe000053fe0000cdfd00003efd0000bcfc000053fc0000e9fb000091fb0000
53fb00000ffb0000d5fa0000abfa000072fa00003ffa000017fa0000ebf90000cef90000c9f900
00d3f90000f5f900002bfa000069fa0000aafa0000e2fa000010fb000033fb00004afb00006ffb
0000a5fb0000f2fb00006efc00000cfd0000bcfd000088fe000051ff000000000000a60000002e
0100008d010000e70100002d020000580200008a020000af020000bd020000d2020000dd020000
da020000e9020000ff0200001a030000540300009c030000e90300004a040000a9040000fc0400
004705000079050000 8e0500007c05000042050000e5040000600400 00c6030000290300008e02
00000b020000ae0100006c0100004d0100004901000048010000 4b0100004f0100004301000034
0100002a0100001401000 0ff000000ed000000bd0000007d0000034000000cbff000060ff0000
0aff0000c2fe0000a6fe0000befe0000f7fe000052ff0000bfff000025000000 7f000000c70000
00f7000000160100002f0100003f0100004c0100005601000 04d0100002e010000f40000009800
00001f00000094ff000001ff000075fe0000fcfd0000a4fd00006bfd00004ffd00004afd00004b
fd000045fd00002bfd0000f5fc0000a2fc000034fc0000bafb000046fb0000defa000092fa0000
67fa000052fa00004cfa000049fa00003cfa000021fa0000fcf90000dbf90000cdf90000dbf900
001afa000082fa000003fb00009afb00002bfc0000a0fc000003fd00004efd000087fd0000d3fd
000034fe0000a9fe00004bff0000fdff0000a5000000510100 00e4010000500200 00bb0200001b
0300007403000 0ea0300006704000 0dc0400005805000 0b9050000f30500001b0600002c060000
2d0600003f0600005f0600008c060000c9060000000700001f0700001c070000ec060000930600
0017060000890500 00ff0400007904000008040000b3030000640300002203000 0ee020000a902
0000620200002002000 0c6010000660100000b0100009500000 01a000000a9ff000022ff0000a6
fe000044fe0000defd00008efd00005dfd000025fd0000fafc0000defc0000b9fc00009dfc0000
94fc000093fc0000a9fc0000dafc000015fd00005cfd0000a7fd0000e3fd000011fe000030fe00
0041fe00004dfe00005dfe00007dfe0000affe0000effe000042ff000098ff0000e3ff00002700
0000560000006c0000007c000000800000007a0000007d0000007d000000710000006300000 03f
00000001000000b6ff00005aff0000f7fe0000a3fe00005ffe00002ffe00001afe000014fe0000
10fe00000bfe0000f8fd0000d8fd0000adfd00007efd000052fd00002afd00000afd0000ebfc00
00c7fc00009afc000066fc000030fc000007fc0000f9fb000016fc000062fc0000d6fc00006bfd
000007fe00008ffe0000fefe000045ff000066ff000079ff00008bff0000aaff0000ebff000048
000000b40000002d0100009a010000ed0100003302000067020000 8f020000cb02000016030000
6b030000d80300004304000 09c040000e7040000160500002805000028050000 1a050000ff0400
00db040000a60400005c040000fa0300007f030000f6020000670200 00e40100007f0100003401
00000c0100000301000000010000fc000000f3000000cc0000009400000056000000 01000000af
ff00006bff00001fff0000e0fe0000
```

Der digitale Kanal kann auch parallel zu einem bestehenden Kanal erstellt werden, etwa durch eine Multi-TRX-Konfiguration:

```
[MultiTX]
TYPE=Multi
TRANSMITTERS=Tx1,TxUDP
```

Einfacher Python-Code zur Ausgabe der UDP-Daten:

```
import socket
from datetime import datetime

def start_udp_server(host='0.0.0.0', port=4300):
    # Create a UDP socket
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    # Bind the socket to the address and port
    server_address = (host, port)
    sock.bind(server_address)

    print(f"Starting UDP server on {host}:{port}. Waiting for data...")

    try:
        while True:
            # Receive data from the client
            data, address = sock.recvfrom(65535)

            # Get the current timestamp with milliseconds
            timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f')[:-3]

            # Convert the data to hexadecimal format
            hex_data = data.hex()

            # Print the timestamp and the received hex data
            print(f"[{timestamp}] Received {len(data)} bytes from {address}: {
hex_data}")

    except KeyboardInterrupt:
        print("\nServer is shutting down.")
    finally:
        sock.close()

if __name__ == '__main__':
    start_udp_server()
```

Einfacher Pyhton-Code um die Daten auf 8000 Samples/s zu konvertieren:

```
import socket
import numpy as np
from scipy import signal  # Correcting the import to use functions directly
import wave

def low_pass_filter(data, cutoff, fs, order=5):
    nyquist = 0.5 * fs
    normal_cutoff = cutoff / nyquist
    b, a = signal.butter(order, normal_cutoff, btype='low', analog=False)
    return signal.lfilter(b, a, data)

def start_udp_server_with_processing(host='0.0.0.0', port=4300, output_file='o
```

```
utput.wav'):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.bind((host, port))

    print(f"Listening on {host}:{port}")

    out_wave = wave.open(output_file, 'wb')
    out_wave.setnchannels(1)
    out_wave.setsampwidth(2)
    out_wave.setframerate(8000)

    try:
        while True:
            data, _ = sock.recvfrom(4032)
            samples = np.frombuffer(data, dtype=np.int16)[::2]

            filtered_samples = low_pass_filter(samples, cutoff=3400, fs=48000)

            downsampled_samples = signal.resample_poly(filtered_samples, up=1,
 down=6)

            out_wave.writeframes(downsampled_samples.astype(np.int16).tobytes(
))

    except KeyboardInterrupt:
        print("\nServer shutting down.")
    finally:
        sock.close()
        out_wave.close()

if __name__ == '__main__':
    start_udp_server_with_processing()
```