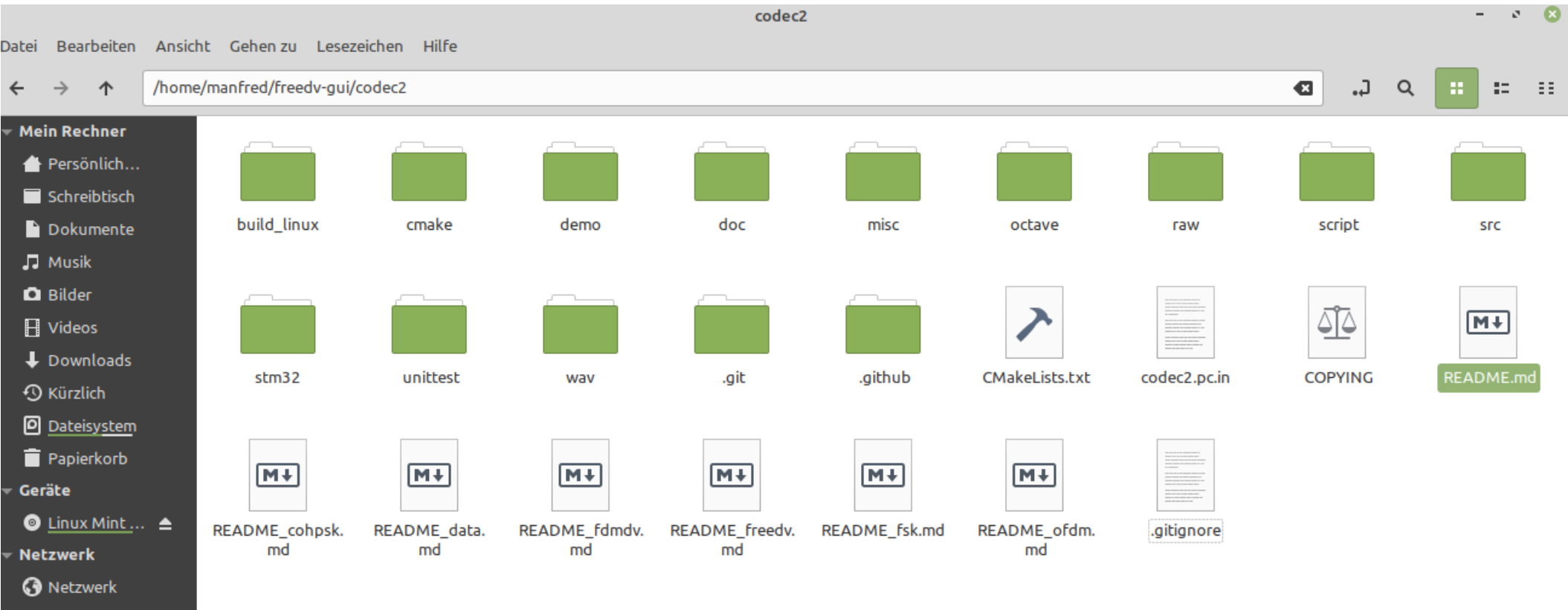


# Codec2 Versuche



1. Listen to Codec 2:

...

```
cd codec2/build_linux
./demo/c2demo ../raw/hts1a.raw hts1a_c2.raw
aplay -f S16_LE ../raw/hts1a.raw
aplay -f S16_LE hts1a_c2.raw
```

...

1. Compress, decompress and then play a file using Codec 2 at 2400 bit/s:

...

```
./src/c2enc 2400 ../raw/hts1a.raw hts1a_c2.bit
./src/c2dec 2400 hts1a_c2.bit hts1a_c2_2400.raw
```

...

which can be played with:

...

```
aplay -f S16_LE hts1a_c2_2400.raw
```

...

Or using Codec 2 using 700C (700 bits/s):

...

```
./src/c2enc 700C ../raw/hts1a.raw hts1a_c2.bit
./src/c2dec 700C hts1a_c2.bit hts1a_c2_700.raw
aplay -f S16_LE hts1a_c2_700.raw
```

...

1. If you prefer a one-liner without saving to files:

...

```
./src/c2enc 1300 ../raw/hts1a.raw - | ./src/c2dec 1300 - - | aplay -f S16_LE
```

...

1. Or you can use your microphone and headphones to encode and listen to the result on the fly:

...

```
br=1300; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
```

...

```
manfred@manfred-virtual-machine: ~/freedv-gui/build_linux
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ ./demo/c2demo ../raw/htsla.raw htsla_c2.raw
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ aplay -f S16_LE ../raw/htsla.raw
Eingabe: Rohdaten '../raw/htsla.raw' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ aplay -f S16_LE htsla_c2.raw
Eingabe: Rohdaten 'htsla_c2.raw' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ ./src/c2enc 2400 ../raw/htsla.raw htsla_c2.bit
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ ./src/c2dec 2400 htsla_c2.bit htsla_c2_2400.raw
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ aplay -f S16_LE htsla_c2_2400.raw
Eingabe: Rohdaten 'htsla_c2_2400.raw' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ ./src/c2enc 700C ../raw/htsla.raw htsla_c2.bit
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ ./src/c2dec 700C htsla_c2.bit htsla_c2_700.raw
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ aplay -f S16_LE htsla_c2_700.raw
Eingabe: Rohdaten 'htsla_c2_700.raw' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=1300; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br -
| ./src/c2dec $br - - | aplay -f S16_LE -
Eingabe: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Eingabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 138,466 ms)
Abbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=1600; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br -
| ./src/c2dec $br - - | aplay -f S16_LE -
Eingabe: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Eingabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 135,988 ms)
Abbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=2400; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br -
| ./src/c2dec $br - - | aplay -f S16_LE -
Eingabe: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Eingabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 263,074 ms)
Abbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=3200; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br -
| ./src/c2dec $br - - | aplay -f S16_LE -
Eingabe: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Eingabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 266,967 ms)
[A^CAbbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=700; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br -
| ./src/c2dec $br - - | aplay -f S16_LE -
Error in mode: 700. Must be 3200, 2400, 1600, 1400, 1300, 1200, 700C, 450, or 450PWB
Error in mode: 700. Must be 3200, 2400, 1600, 1400, 1300, 1200, 700C, 450, 450PWB or WB
Eingabe: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
```

e/manfred/freedv-gui/codec2/build\_linux

Name	Größe	Dateityp	Änderungsdatum
Testing	1 Objekt	Ordner	Do 11 Aug 2022
src	56 Objekte	Ordner	Do 04 Aug 2022
CMakeFiles	12 Objekte	Ordner	Do 04 Aug 2022
demo	10 Objekte	Ordner	Mi 20 Jul 2022
codec2	1 Objekt	Ordner	Mi 20 Jul 2022
htsla_c2_700.raw	48,0 kB	Bild	Sa 13 Aug 2022
htsla_c2.bit	300 Bytes	Binär	Sa 13 Aug 2022
htsla_c2_2400.raw	48,0 kB	Bild	Sa 13 Aug 2022
htsla_c2.raw	48,0 kB	Bild	Sa 13 Aug 2022

c2enc erzeugt aus einem mit 8kHz abgetasteten Signal mit 48kByte Größe ein 300Byte großes File (\*.bit). Dieses kann nicht über das Mikrophon eines TRX ausgesendet werden! Es ist noch kein Modulator vorhanden.

Microphone and headphones to encode and listen to the result on the fly !!! Nur Codec, noch kein Modulationssignal das als NF auf den Mikrofoneingang gegeben werden kann !!!

fred@manfred-virtual-machine: ~/freedv-gui/codec2/build\_linux


Datei Bearbeiten Ansicht Suchen Terminal Hilfe

```
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=1300; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 138,466 ms)
^CAbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=1600; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 135,988 ms)
^CAbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=2400; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 263,074 ms)
^CAbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=3200; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 266,967 ms)
^[A^CAbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=700; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Error in mode: 700. Must be 3200, 2400, 1600, 1400, 1300, 1200, 700c, 450, or 450PWB
Error in mode: 700. Must be 3200, 2400, 1600, 1400, 1300, 1200, 700c, 450, 450PWB or WB
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
aplay: read header:2861: Lesefehler
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=700c; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Error in mode: 700c. Must be 3200, 2400, 1600, 1400, 1300, 1200, 700c, 450, 450PWB or WB
Error in mode: 700c. Must be 3200, 2400, 1600, 1400, 1300, 1200, 700c, 450, or 450PWB
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
aplay: read header:2861: Lesefehler
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=700c; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 136,007 ms)
^CAbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=450; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 132,181 ms)
^CAbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=450PWB; arecord -f S16_LE -c 1 -r 8000 | ./src/c2enc $br - - | ./src/c2dec $br - - | aplay -f S16_LE -
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
^CAbruch durch Signal Unterbrechung ...
```

# Demo receive program for FreeDV API 700D mode.


Python-Version (zwei Punkte)

cd ~/codec2/build\_linux

cat ../raw/ve9qrp\_10s.raw | ./demo/freedv\_700d\_tx |  ./demo/freedv\_700d\_rx.py | aplay -f S16\_LE

```
manfred@manfred-virtual-machine:~/freedv-gui/codec2$ cd build_linux/  
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ cat ../raw/ve9qrp_10s.raw | ./demo/freedv_700d_tx | ./demo/freedv_700d_rx.py | aplay -f S16_LE  
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono  
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$
```

C-Version (ein Punkt)

cat ../raw/ve9qrp\_10s.raw | ./demo/freedv\_700d\_tx |  ./demo/freedv\_700d\_rx | aplay -f S16\_LE

```
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ cat ../raw/ve9qrp_10s.raw | ./demo/freedv_700d_tx | ./demo/freedv_700d_rx | aplay -f S16_LE  
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono  
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$
```

# NF FreeDV TX-Signal aus ve9qrp\_10s.raw erzeugen

```
cat ../raw/ve9qrp_10s.raw | ./demo/freedv_700d_tx - - | aplay -f S16_LE -
```

```
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ cat ../raw/ve9qrp_10s.raw | ./demo/freedv_700d_tx - - | aplay -f S16_LE -  
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono  
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$
```

# NF TX-Signal aus Mikrofon erzeugen

Mikrofon (Sprache) ==> FreeDV\_700\_tx ==> Lautsprecher (Mikrofoneingang Funkgerät)

```
arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_700d_tx - - | aplay -f S16_LE -
```

```
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_700d_tx - - | aplay -f S16_LE -
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 690,926 ms)
Unterlauf!!! (mindestens 3860,234 ms)
Unterlauf!!! (mindestens 5087,326 ms)
^CAbbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$
```

# Test der ganzen Kette:

Mikrofon (Sprache) ==> FreeDV\_TX ==>FreeDV\_RX ==> Lautsprecher

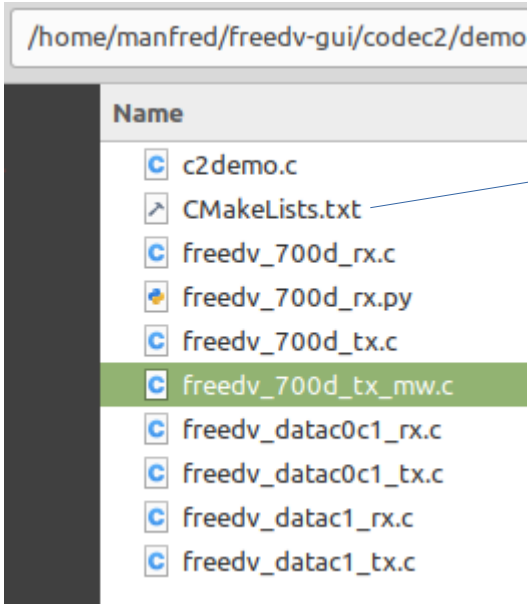
```
arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_700d_tx | ../demo/freedv_700d_rx.py | aplay -f S16_LE
```

```
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ br=3200; arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_700d_tx | ../demo/freedv_700d_rx.py | aplay -f S16_LE
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 4552,668 ms)
^CAbbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
Traceback (most recent call last):
  File "../demo/freedv_700d_rx.py", line 46, in <module>
    demod_in = sys.stdin.buffer.read(nin*2)
KeyboardInterrupt
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_700d_tx | ../demo/freedv_700d_rx.py | aplay -f S16_LE
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
^CAbbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
Traceback (most recent call last):
  File "../demo/freedv_700d_rx.py", line 46, in <module>
    demod_in = sys.stdin.buffer.read(nin*2)
KeyboardInterrupt
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$
```



Jetzt wird „selbergebaut“.

1. freedv\_700d\_tx.c wird kopiert zu freedv\_700d\_tx\_mw.c
2. CmakeLists.txt wird um die zwei grün markierten Zeilen erweitert



The image shows a text editor window titled `*CMakeLists.txt (~/.freedv-gui/codec2/demo)`. The editor contains the following CMake code:

```
add_definitions(-DFLOATING_POINT -DVAR_ARRAYS)
include_directories(../src)

add_executable(c2demo c2demo.c)
target_link_libraries(c2demo codec2)
add_executable(freedv_700d_tx freedv_700d_tx.c)
target_link_libraries(freedv_700d_tx codec2)
add_executable(freedv_700d_rx freedv_700d_rx.c)
target_link_libraries(freedv_700d_rx codec2)
add_executable(freedv_datac1_tx freedv_datac1_tx.c)
target_link_libraries(freedv_datac1_tx codec2)
add_executable(freedv_datac1_rx freedv_datac1_rx.c)
target_link_libraries(freedv_datac1_rx codec2)
add_executable(freedv_datac0c1_tx freedv_datac0c1_tx.c)
target_link_libraries(freedv_datac0c1_tx codec2)
add_executable(freedv_datac0c1_rx freedv_datac0c1_rx.c)
target_link_libraries(freedv_datac0c1_rx codec2)
add_executable(freedv_700d_tx_mw freedv_700d_tx_mw.c)
target_link_libraries(freedv_700d_tx_mw codec2)
```

The last two lines are highlighted in green, indicating the new code added to the file.

3. im freedv-gui/codec2/build\_linux/demo Verzeichnis „make“ ausführen.

```
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux/src$ cd ..
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ cd demo
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux/demo$ make
[ 2%] Built target generate_codebook
[ 82%] Built target codec2
[ 82%] Linking C executable c2demo
[ 84%] Built target c2demo
Scanning dependencies of target freedv_700d_tx
[ 84%] Building C object demo/CMakeFiles/freedv_700d_tx.dir/freedv_700d_tx.c.o
[ 86%] Linking C executable freedv_700d_tx
[ 86%] Built target freedv_700d_tx
Scanning dependencies of target freedv_700d_rx
[ 86%] Building C object demo/CMakeFiles/freedv_700d_rx.dir/freedv_700d_rx.c.o
[ 88%] Linking C executable freedv_700d_rx
[ 88%] Built target freedv_700d_rx
Scanning dependencies of target freedv_datac0c1_rx
[ 91%] Building C object demo/CMakeFiles/freedv_datac0c1_rx.dir/freedv_datac0c1_rx.c.o
[ 91%] Linking C executable freedv_datac0c1_rx
[ 91%] Built target freedv_datac0c1_rx
Scanning dependencies of target freedv_datac1_tx
[ 93%] Building C object demo/CMakeFiles/freedv_datac1_tx.dir/freedv_datac1_tx.c.o
[ 93%] Linking C executable freedv_datac1_tx
[ 93%] Built target freedv_datac1_tx
Scanning dependencies of target freedv_700d_tx_mw
[ 93%] Building C object demo/CMakeFiles/freedv_700d_tx_mw.dir/freedv_700d_tx_mw.c.o
[ 95%] Linking C executable freedv_700d_tx_mw
[ 95%] Built target freedv_700d_tx_mw
Scanning dependencies of target freedv_datac0c1_tx
[ 97%] Building C object demo/CMakeFiles/freedv_datac0c1_tx.dir/freedv_datac0c1_tx.c.o
[ 97%] Linking C executable freedv_datac0c1_tx
[ 97%] Built target freedv_datac0c1_tx
Scanning dependencies of target freedv_datac1_rx
[100%] Building C object demo/CMakeFiles/freedv_datac1_rx.dir/freedv_datac1_rx.c.o
[100%] Linking C executable freedv_datac1_rx
[100%] Built target freedv_datac1_rx
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux/demo$
```

/home/manfred/freedv-gui/codec2/demo

Name

- c2demo.c
- CMakeLists.txt
- freedv\_700d\_rx.c
- freedv\_700d\_rx.py
- freedv\_700d\_tx.c
- freedv\_700d\_tx\_mw.c
- freedv\_datac0c1\_rx.c
- freedv\_datac0c1\_tx.c
- freedv\_datac1\_rx.c
- freedv\_datac1\_tx.c

4. Jetzt wird das ausführbare „freedv\_tx\_mw“ erzeugt

/home/manfred/freedv-gui/codec2/build\_linux/demo

Name

- CMakeFiles
- c2demo
- cmake\_install.cmake
- freedv\_700d\_rx
- freedv\_700d\_tx
- freedv\_700d\_tx\_mw
- freedv\_datac0c1\_rx
- freedv\_datac0c1\_tx
- freedv\_datac1\_rx
- freedv\_datac1\_tx
- Makefile

```
arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_700d_tx_mw | ../demo/freedv_700d_rx.py | aplay -f S16_LE
arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_700d_tx_mw | ./demo/freedv_700d_rx | aplay -f S16_LE
```

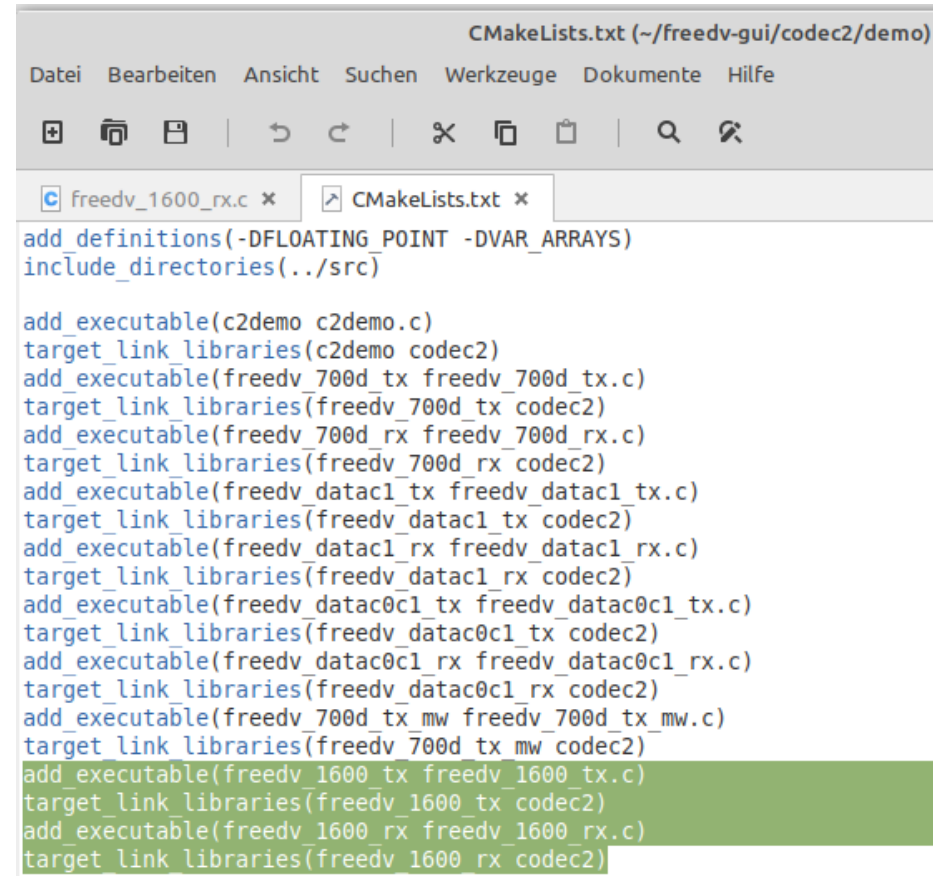
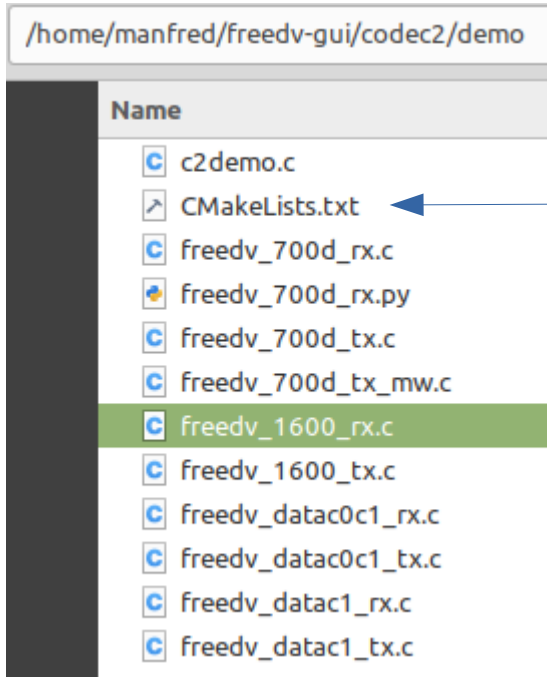
```
anfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ cd demo
anfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux/demo$ make
 2%) Built target generate_codebook
82%) Built target codec2
82%) Linking C executable c2demo
84%) Built target c2demo
anning dependencies of target freedv_700d_tx
84%) Building C object demo/CMakeFiles/freedv_700d_tx.dir/freedv_700d_tx.c.o
86%) Linking C executable freedv_700d_tx
86%) Built target freedv_700d_tx
anning dependencies of target freedv_700d_rx
86%) Building C object demo/CMakeFiles/freedv_700d_rx.dir/freedv_700d_rx.c.o
88%) Linking C executable freedv_700d_rx
88%) Built target freedv_700d_rx
anning dependencies of target freedv_datac0c1_rx
91%) Building C object demo/CMakeFiles/freedv_datac0c1_rx.dir/freedv_datac0c1_rx.c.o
91%) Linking C executable freedv_datac0c1_rx
91%) Built target freedv_datac0c1_rx
anning dependencies of target freedv_datac1_tx
93%) Building C object demo/CMakeFiles/freedv_datac1_tx.dir/freedv_datac1_tx.c.o
93%) Linking C executable freedv_datac1_tx
93%) Built target freedv_datac1_tx
anning dependencies of target freedv_700d_tx_mw
93%) Building C object demo/CMakeFiles/freedv_700d_tx_mw.dir/freedv_700d_tx_mw.c.o
95%) Linking C executable freedv_700d_tx_mw
95%) Built target freedv_700d_tx_mw
anning dependencies of target freedv_datac0c1_tx
97%) Building C object demo/CMakeFiles/freedv_datac0c1_tx.dir/freedv_datac0c1_tx.c.o
97%) Linking C executable freedv_datac0c1_tx
97%) Built target freedv_datac0c1_tx
anning dependencies of target freedv_datac1_rx
100%) Building C object demo/CMakeFiles/freedv_datac1_rx.dir/freedv_datac1_rx.c.o
100%) Linking C executable freedv_datac1_rx
100%) Built target freedv_datac1_rx
anfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux/demo$ cd ..
anfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_700d_tx_mw | ../demo/freedv_700d_rx.py | aplay -f S16_LE
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 4552,626 ms)
Unterlauf!!! (mindestens 1224,891 ms)
```

Test ob es funktioniert



Neuen Demo Mode erzeugen ( Mode 1600).

- Dazu zuerst die Files freedv\_700d\_tx.c und freedv\_700d\_rx.c kopieren und als freedv\_1600\_tx.c und freedv\_1600\_rx.c umbenennen.
- Danach Inhalt in freedv\_1600\_tx.c und freedv\_1600\_rx.c modifizieren.
- CmakeList.txt um grünen Block erweitern



```
C *freedv_700d_tx.c x
```

```
FILE.....: freedv_700d_tx.c
AUTHOR.....: David Rowe
DATE CREATED: April 2021
Demo transmit program using the FreeDV API (700D mode).
usage:
cd ~/codecs2/build_linux
cat ../raw/ve9qrp_10s.raw | ./demo/freedv_700d_tx | ./demo/freedv_700d_rx | aplay -f S16_LE
Listen to the modulated Tx signal:
cat ../raw/ve9qrp_10s.raw | ./demo/freedv_700d_tx | aplay -f S16_LE
/*-----*/
/*
Copyright (C) 2021 David Rowe. All rights reserved.
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU Lesser General Public License version 2.1, as
published by the Free Software Foundation. This program is
distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
License for more details.
You should have received a copy of the GNU Lesser General Public License
along with this program; if not, see <http://www.gnu.org/licenses/>.
*/
#include <assert.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include "freedv_api.h"

int main(int argc, char *argv[]) {
    struct freedv *freedv;

    freedv = freedv_open(FREEDV_MODE_700D);
    assert(freedv != NULL);

    /* handy functions to set buffer sizes */
    int n_speech_samples = freedv_get_n_speech_samples(freedv);
    short speech_in[n_speech_samples];
    int n_nom_modem_samples = freedv_get_n_nom_modem_samples(freedv);
    short mod_out[n_nom_modem_samples];

    /* OK main loop ----- */

    while(fread(speech_in, sizeof(short), n_speech_samples, stdin) == n_speech_samples) {
        freedv_tx(freedv, mod_out, speech_in);
        fwrite(mod_out, sizeof(short), n_nom_modem_samples, stdout);
    }

    freedv_close(freedv);

    return 0;
}
```

```
C *freedv_700d_rx.c x
```

```
FILE.....: freedv_700d_rx.c
AUTHOR.....: David Rowe
DATE CREATED: April 2021

Demo receive program for FreeDV API (700D mode), see freedv_700d_tx.c for
instructions.
/*-----*/
/*
Copyright (C) 2021 David Rowe

All rights reserved.

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU Lesser General Public License version 2.1, as
published by the Free Software Foundation. This program is
distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
License for more details.

You should have received a copy of the GNU Lesser General Public License
along with this program; if not, see <http://www.gnu.org/licenses/>.
*/
#include <assert.h>
#include <stdlib.h>
#include <stdio.h>

#include "freedv_api.h"

int main(int argc, char *argv[]) {
    struct freedv *freedv;

    freedv = freedv_open(FREEDV_MODE_700D);
    assert(freedv != NULL);

    /* note API functions to tell us how big our buffers need to be */
    short speech_out[freedv_get_n_max_speech_samples(freedv)];
    short demod_in[freedv_get_n_max_modem_samples(freedv)];

    size_t nin,nout;
    nin = freedv_nin(freedv);
    while(fread(demod_in, sizeof(short), nin, stdin) == nin) {
        nout = freedv_rx(freedv, speech_out, demod_in);
        nin = freedv_nin(freedv); /* call me on every loop! */
        fwrite(speech_out, sizeof(short), nout, stdout);
    }

    freedv_close(freedv);

    return 0;
}
```

```
freedv_1600_tx.c (~/.freedv-gui/codec2/demo)
Datei Bearbeiten Ansicht Suchen Werkzeuge Dokumente Hilfe
freedv_1600_tx.c x
along with this program; if not, see <http://www.gnu.org/licenses/>.
*/
#include <assert.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "freedv_api.h"
int main(int argc, char *argv[]) {
    struct freedv *freedv;
    freedv = freedv_open(FREEDV_MODE_1600);
    assert(freedv != NULL);
    /* handy functions to set buffer sizes */
    int n_speech_samples = freedv_get_n_speech_samples(freedv);
    short speech_in[n_speech_samples];
    int n_nom_modem_samples = freedv_get_n_nom_modem_samples(freedv);
    short mod_out[n_nom_modem_samples];
    /* OK main loop ----- */
    while(fread(speech_in, sizeof(short), n_speech_samples, stdin) == n_speech_samples) {
        freedv_tx(freedv, mod_out, speech_in);
        fwrite(mod_out, sizeof(short), n_nom_modem_samples, stdout);
    }
    freedv_close(freedv);
    return 0;
}
```

```
freedv_1600_rx.c (~/.freedv-gui/codec2/demo)
Datei Bearbeiten Ansicht Suchen Werkzeuge Dokumente Hilfe
freedv_1600_rx.c x
License for more details.
You should have received a copy of the GNU Lesser General Public License
along with this program; if not, see <http://www.gnu.org/licenses/>.
*/
#include <assert.h>
#include <stdlib.h>
#include <stdio.h>
#include "freedv_api.h"
int main(int argc, char *argv[]) {
    struct freedv *freedv;
    freedv = freedv_open(FREEDV_MODE_1600);
    assert(freedv != NULL);
    /* note API functions to tell us how big our buffers need to be */
    short speech_out[freedv_get_n_max_speech_samples(freedv)];
    short demod_in[freedv_get_n_max_modem_samples(freedv)];
    size_t nin, nout;
    nin = freedv_nin(freedv);
    while(fread(demod_in, sizeof(short), nin, stdin) == nin) {
        nout = freedv_rx(freedv, speech_out, demod_in);
        nin = freedv_nin(freedv); /* call me on every loop! */
        fwrite(speech_out, sizeof(short), nout, stdout);
    }
    freedv_close(freedv);
    return 0;
}
```

im freedv-gui/codec2/build\_linux/demo Verzeichnis „make“ ausführen.



Test ob der neue Demo Mode 1600 funktioniert:

In freedv-gui/codec2/build\_linux/ folgenden Befehl eingeben:

```
record -f S16_LE -c 1 -r 8000 | ./demo/freedv_1600_tx | ./demo/freedv_1600_rx | aplay -f S16_LE
```

```
manfred@manfred-virtual-machine: ~/freedv-gui/codec2/build_linux
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ cd demo
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux/demo$ make
[ 2%] Built target generate_codebook
[ 78%] Built target codec2
[ 80%] Built target c2demo
[ 82%] Built target freedv_700d_rx
[ 84%] Built target freedv_datac0c1_rx
Scanning dependencies of target freedv_1600_rx
[ 84%] Building C object demo/CMakeFiles/freedv_1600_rx.dir/freedv_1600_rx.c.o
[ 86%] Linking C executable freedv_1600_rx
[ 86%] Built target freedv_1600_rx
[ 89%] Built target freedv_datac1_tx
[ 91%] Built target freedv_700d_tx_mw
[ 93%] Built target freedv_datac0c1_tx
[ 95%] Built target freedv_700d_tx
Scanning dependencies of target freedv_1600_tx
[ 95%] Building C object demo/CMakeFiles/freedv_1600_tx.dir/freedv_1600_tx.c.o
[ 97%] Linking C executable freedv_1600_tx
[ 97%] Built target freedv_1600_tx
[100%] Built target freedv_datac1_rx
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux/demo$ cd ..
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_1600_tx | ./demo/freedv_1600_rx | aplay -f S16_LE
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 418,469 ms)
^Z
[9]+  Angehalten          arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_1600_tx | ./demo/freedv_1600_rx | aplay -f S16_LE
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$
```

Neuen Demo Mode erzeugen ( Mode 2020). **Dieser wird mit 16kHz abgetastet!**

- Dazu zuerst die Files freedv\_700d\_tx.c und freedv\_700d\_rx.c kopieren und als freedv\_2020\_tx.c und freedv\_2020\_rx.c umbenennen.
- Danach Inhalt in freedv\_2020\_tx.c und freedv\_2020\_rx.c modifizieren.
- CmakeList.txt um grünen Block erweitern

```
freedv_2020_tx.c (~/.freedv-gui/codec2/demo)
Datei Bearbeiten Ansicht Suchen Werkzeuge Dokumente Hilfe

freedv_2020_tx.c x
along with this program; if not, see <http://www.gnu.org/licenses/>.
*/

#include <assert.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include "freedv_api.h"

int main(int argc, char *argv[]) {
    struct freedv *freedv;

    freedv = freedv_open(FREEDV_MODE_2020);
    assert(freedv != NULL);

    /* handy functions to set buffer sizes */
    int n_speech_samples = freedv_get_n_speech_samples(freedv);
    short speech_in[n_speech_samples];
    int n_nom_modem_samples = freedv_get_n_nom_modem_samples(freedv);
    short mod_out[n_nom_modem_samples];

    /* OK main loop ----- */

    while(fread(speech_in, sizeof(short), n_speech_samples, stdin) == n_speech_samples) {
        freedv_tx(freedv, mod_out, speech_in);
        fwrite(mod_out, sizeof(short), n_nom_modem_samples, stdout);
    }

    freedv_close(freedv);

    return 0;
}

C Leerzeichen: 4 Z. 3, Sp. 27 EINF
```

```
freedv_2020_rx.c (~/.freedv-gui/codec2/demo)
Datei Bearbeiten Ansicht Suchen Werkzeuge Dokumente Hilfe

freedv_2020_rx.c x
distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
License for more details.

You should have received a copy of the GNU Lesser General Public License
along with this program; if not, see <http://www.gnu.org/licenses/>.
*/

#include <assert.h>
#include <stdlib.h>
#include <stdio.h>

#include "freedv_api.h"

int main(int argc, char *argv[]) {
    struct freedv *freedv;

    freedv = freedv_open(FREEDV_MODE_2020);
    assert(freedv != NULL);

    /* note API functions to tell us how big our buffers need to be */
    short speech_out[freedv_get_n_max_speech_samples(freedv)];
    short demod_in[freedv_get_n_max_modem_samples(freedv)];

    size_t nin, nout;
    nin = freedv_nin(freedv);
    while(fread(demod_in, sizeof(short), nin, stdin) == nin) {
        nout = freedv_rx(freedv, speech_out, demod_in);
        nin = freedv_nin(freedv); /* call me on every loop! */
        fwrite(speech_out, sizeof(short), nout, stdout);
    }
}

C Leerzeichen: 4 Z. 38, Sp. 42 EINF
```

```
CMakeLists.txt (~/.freedv-gui/codec2/demo)
Datei Bearbeiten Ansicht Suchen Werkzeuge Dokumente Hilfe

CMakeLists.txt x
add_definitions(-DFLOATING_POINT -DVAR_ARRAYS)
include_directories(../src)

add_executable(c2demo c2demo.c)
target_link_libraries(c2demo codec2)
add_executable(freedv_700d_tx freedv_700d_tx.c)
target_link_libraries(freedv_700d_tx codec2)
add_executable(freedv_700d_rx freedv_700d_rx.c)
target_link_libraries(freedv_700d_rx codec2)
add_executable(freedv_datac1_tx freedv_datac1_tx.c)
target_link_libraries(freedv_datac1_tx codec2)
add_executable(freedv_datac1_rx freedv_datac1_rx.c)
target_link_libraries(freedv_datac1_rx codec2)
add_executable(freedv_datac0c1_tx freedv_datac0c1_tx.c)
target_link_libraries(freedv_datac0c1_tx codec2)
add_executable(freedv_datac0c1_rx freedv_datac0c1_rx.c)
target_link_libraries(freedv_datac0c1_rx codec2)
add_executable(freedv_700d_tx_mw freedv_700d_tx_mw.c)
target_link_libraries(freedv_700d_tx_mw codec2)
add_executable(freedv_1600_tx freedv_1600_tx.c)
target_link_libraries(freedv_1600_tx codec2)
add_executable(freedv_1600_rx freedv_1600_rx.c)
target_link_libraries(freedv_1600_rx codec2)
add_executable(freedv_2020_tx freedv_2020_tx.c)
target_link_libraries(freedv_2020_tx codec2)
add_executable(freedv_2020_rx freedv_2020_rx.c)
target_link_libraries(freedv_2020_rx codec2)
```

```
arecord -f S16_LE -c 1 -r 16000 | ./demo/freedv_2020_tx | ./demo/freedv_2020_rx | aplay -f S16_LE -r 16000
```



Datei Bearbeiten Ansicht Suchen Terminal Hilfe

manfred@manfred-virtual-machine:~/freedv-gui/codec2/build\_linux\$ cd demo

manfred@manfred-virtual-machine:~/freedv-gui/codec2/build\_linux/demo\$ make

```
-- codec2 version: 1.0.4
-- freedv-gui current git hash: v1.0.0-425-g30c6b8ba
-- Codec2 current git hash: v1.0.0-425-g30c6b8ba
-- liblpcnetfreedv found in build tree.
-- Build type is: Debug
-- Compiler Flags: -Wall -Wno-strict-overflow -std=gnu11 -fPIC-g -O2 -DDUMP
-- Libraries linked: m
-- Compilation date = XX20220814XX
-- Configuring done
-- Generating done
-- Build files have been written to: /home/manfred/freedv-gui/codec2/build_linux
```

[ 2%] Built target generate\_codebook

[ 74%] Built target codec2

[ 76%] Built target c2demo

[ 78%] Built target freedv\_700d\_rx

[ 80%] Built target freedv\_datac0c1\_rx

[ 82%] Built target freedv\_1600\_rx

[ 85%] Built target freedv\_datac1\_tx

Scanning dependencies of target freedv\_2020\_rx

[ 87%] Building C object demo/CMakeFiles/freedv\_2020\_rx.dir/freedv\_2020\_rx.c.o

[ 87%] Linking C executable freedv\_2020\_rx

[ 87%] Built target freedv\_2020\_rx

[ 89%] Built target freedv\_700d\_tx\_mw

[ 91%] Built target freedv\_datac0c1\_tx

[ 93%] Built target freedv\_700d\_tx

[ 95%] Built target freedv\_1600\_tx

[ 97%] Built target freedv\_datac1\_rx

Scanning dependencies of target freedv\_2020\_tx

[100%] Building C object demo/CMakeFiles/freedv\_2020\_tx.dir/freedv\_2020\_tx.c.o

[100%] Linking C executable freedv\_2020\_tx

[100%] Built target freedv\_2020\_tx

manfred@manfred-virtual-machine:~/freedv-gui/codec2/build\_linux/demo\$ cd ..

manfred@manfred-virtual-machine:~/freedv-gui/codec2/build\_linux\$ arecord -f S16\_LE -c 1 -r 16000 | ./demo/freedv\_2020\_tx | ./demo/freedv\_2020\_rx | aplay -f S16\_LE -r 16000

Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 16000 Hz, mono

Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 16000 Hz, mono

Unterlauf!!! (mindestens 424,033 ms)

^Z

[13]+ Angehalten arecord -f S16\_LE -c 1 -r 16000 | ./demo/freedv\_2020\_tx | ./demo/freedv\_2020\_rx | aplay -f S16\_LE -r 16000

manfred@manfred-virtual-machine:~/freedv-gui/codec2/build\_linux\$

```
freedv_1600mw_tx.c (~/.freedv-gui/codec2/demo)
Datei Bearbeiten Ansicht Suchen Werkzeuge Dokumente Hilfe
freedv_1600mw_tx.c x
#include <assert.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include "freedv_api.h"

int main(int argc, char *argv[]) {
    struct freedv *freedv;

    freedv = freedv_open(FREEDV_MODE_1600mw);
    assert(freedv != NULL);

    /* handy functions to set buffer sizes */
    int n_speech_samples = freedv_get_n_speech_samples(freedv);
    short speech_in[n_speech_samples];
    int n_nom_modem_samples =
    freedv_get_n_nom_modem_samples(freedv);
    short mod_out[n_nom_modem_samples];

    /* OK main loop ----- */

    while(fread(speech_in, sizeof(short), n_speech_samples,
    stdin) == n_speech_samples) {
        freedv_tx(freedv, mod_out, speech_in);
        fwrite(mod_out, sizeof(short), n_nom_modem_samples,
        stdout);
    }

    freedv_close(freedv);

    return 0;
}
C Leerzeichen: 4 Z. 47, Sp. 44 EINF
```

```
freedv_1600mw_rx.c (~/.freedv-gui/codec2/demo)
Datei Bearbeiten Ansicht Suchen Werkzeuge Dokumente Hilfe
freedv_1600mw_rx.c x
Public License
along with this program; if not, see <http://www.gnu.org/licenses/>.
*/

#include <assert.h>
#include <stdlib.h>
#include <stdio.h>

#include "freedv_api.h"

int main(int argc, char *argv[]) {
    struct freedv *freedv;

    freedv = freedv_open(FREEDV_MODE_1600mw);
    assert(freedv != NULL);

    /* note API functions to tell us how big our buffers need
    to be */
    short speech_out[freedv_get_n_max_speech_samples(freedv)];
    short demod_in[freedv_get_n_max_modem_samples(freedv)];

    size_t nin, nout;
    nin = freedv_nin(freedv);
    while(fread(demod_in, sizeof(short), nin, stdin) == nin) {
        nout = freedv_rx(freedv, speech_out, demod_in);
        nin = freedv_nin(freedv); /* call me on every loop! */
        fwrite(speech_out, sizeof(short), nout, stdout);
    }

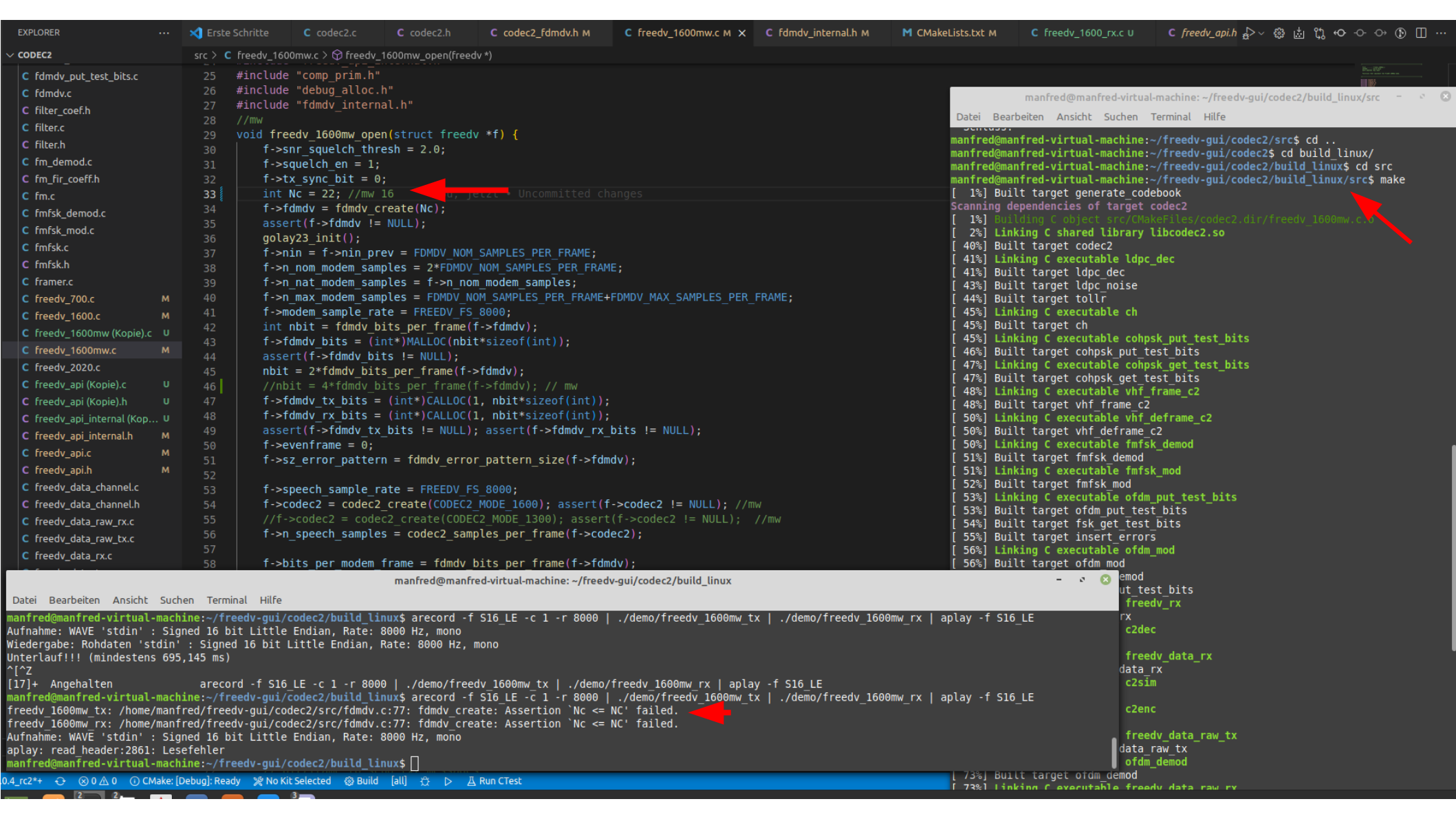
    freedv_close(freedv);
    return 0;
}
C Leerzeichen: 4 Z. 38, Sp. 44 EINF
```

```
CMakeLists.txt (~/.freedv-gui/codec2/demo)
Datei Bearbeiten Ansicht Suchen Werkzeuge Dokumente Hilfe
CMakeLists.txt x
add_definitions(-DFLOATING_POINT -DVAR_ARRAYS)
include_directories(..src)

add_executable(c2demo c2demo.c)
target_link_libraries(c2demo codec2)
add_executable(freedv_700d_tx freedv_700d_tx.c)
target_link_libraries(freedv_700d_tx codec2)
add_executable(freedv_700d_rx freedv_700d_rx.c)
target_link_libraries(freedv_700d_rx codec2)
add_executable(freedv_datac1_tx freedv_datac1_tx.c)
target_link_libraries(freedv_datac1_tx codec2)
add_executable(freedv_datac1_rx freedv_datac1_rx.c)
target_link_libraries(freedv_datac1_rx codec2)
add_executable(freedv_datac0c1_tx freedv_datac0c1_tx.c)
target_link_libraries(freedv_datac0c1_tx codec2)
add_executable(freedv_datac0c1_rx freedv_datac0c1_rx.c)
target_link_libraries(freedv_datac0c1_rx codec2)
add_executable(freedv_700d_tx_mw freedv_700d_tx_mw.c)
target_link_libraries(freedv_700d_tx_mw codec2)
add_executable(freedv_1600_tx freedv_1600_tx.c)
target_link_libraries(freedv_1600_tx codec2)
add_executable(freedv_1600_rx freedv_1600_rx.c)
target_link_libraries(freedv_1600_rx codec2)
add_executable(freedv_2020_tx freedv_2020_tx.c)
target_link_libraries(freedv_2020_tx codec2)
add_executable(freedv_2020_rx freedv_2020_rx.c)
target_link_libraries(freedv_2020_rx codec2)
add_executable(freedv_1600mw_tx freedv_1600mw_tx.c)
target_link_libraries(freedv_1600mw_tx codec2)
add_executable(freedv_1600mw_rx freedv_1600mw_rx.c)
target_link_libraries(freedv_1600mw_rx codec2)
CMake Leerzeichen: 4 Z. 28, Sp. 1 EINF
```

Das ist jetzt der Demo Mode für **1600mw**

```
arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_1600mw_tx | ./demo/freedv_1600mw_rx | aplay -f S16_LE
```



```
arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_1600mw_tx | ./demo/freedv_1600mw_rx | aplay -f S16_LE <==3200bit/s
arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_1600_tx | ./demo/freedv_1600_rx | aplay -f S16_LE <==1300bit/s
```

```
src > C freedv_1600mw.c > freedv_1600mw_open(freedv *)
33 int Nc = 20; //mw 16
34 f->fdmdv = fdmdv_create(Nc);
35 assert(f->fdmdv != NULL);
36 golay23_init();
37 f->nin = f->nin_prev = FDMDV_NOM_SAMPLES_PER_FRAME;
38 f->n_nom_modem_samples = 2*FDMDV_NOM_SAMPLES_PER_FRAME;
39 f->n_nat_modem_samples = f->n_nom_modem_samples;
40 f->n_max_modem_samples = FDMDV_NOM_SAMPLES_PER_FRAME+FDMDV_MAX_SAMPLES_PER_FRAME;
41 f->modem_sample_rate = FREEDV_FS_8000;
42 int nbit = fdmdv_bits_per_frame(f->fdmdv);
43 f->fdmdv_bits = (int*)MALLOC(nbit*sizeof(int));
44 assert(f->fdmdv_bits != NULL);
45 nbit = 2*fdmdv_bits_per_frame(f->fdmdv);
46 //nbit = 4*fdmdv_bits_per_frame(f->fdmdv); // mw
47 f->fdmdv_tx_bits = (int*)CALLOC(1, nbit*sizeof(int));
48 f->fdmdv_rx_bits = (int*)CALLOC(1, nbit*sizeof(int));
49 assert(f->fdmdv_tx_bits != NULL); assert(f->fdmdv_rx_bits != NULL);
50 f->evenframe = 0;
51 f->sz_error_pattern = fdmdv_error_pattern_size(f->fdmdv);
52
53 f->speech_sample_rate = FREEDV_FS_8000;
54 f->codec2 = codec2_create(CODEC2_MODE_3200); assert(f->codec2 != NULL); //mw
55 //f->codec2 = codec2_create(CODEC2_MODE_1300); assert(f->codec2 != NULL); //mw
56 f->n_speech_samples = codec2_samples_per_frame(f->codec2);
57
58 f->bits_per_modem_frame = fdmdv_bits_per_frame(f->fdmdv);
59 f->bits_per_codec_frame = codec2_bits_per_frame(f->codec2);
60 f->n_codec_frames = 1;
61 f->tx_payload_bits = MALLOC(f->bits_per_codec_frame); assert(f->tx_payload_bits != NULL);
62 f->rx_payload_bits = MALLOC(f->bits_per_codec_frame); assert(f->rx_payload_bits != NULL);
63
64
65
```

```
manfred@manfred-virtual-machine: ~/freedv-gui/codec2/build_linux/src
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux/src$ make
[ 1%] Built target generate_codebook
Scanning dependencies of target codec2
[ 1%] Building C object src/CMakeFiles/codec2.dir/freedv_1600mw.c.o
[ 2%] Linking C shared library libcodec2.so
[ 40%] Built target codec2
[ 41%] Linking C executable ldpc_dec
[ 41%] Built target ldpc_dec
[ 43%] Built target ldpc_noise
[ 44%] Built target tollr
[ 45%] Linking C executable ch
[ 45%] Built target ch
[ 45%] Linking C executable cohpsk_put_test_bits
[ 46%] Built target cohpsk_put_test_bits
[ 47%] Linking C executable cohpsk_get_test_bits
[ 47%] Built target cohpsk_get_test_bits
[ 48%] Linking C executable vhf_frame_c2
[ 48%] Built target vhf_frame_c2
[ 50%] Linking C executable vhf_deframe_c2
[ 50%] Built target vhf_deframe_c2
[ 50%] Linking C executable fmsk_demod
[ 51%] Built target fmsk_demod
[ 51%] Linking C executable fmsk_mod
[ 52%] Built target fmsk_mod
[ 53%] Linking C executable ofdm_put_test_bits
[ 53%] Built target ofdm_put_test_bits
[ 54%] Built target fsk_get_test_bits
[ 55%] Built target insert_errors
[ 56%] Linking C executable ofdm_mod
[ 56%] Built target ofdm_mod
[ 61%] Built target fdmdv_demo
```

```
manfred@manfred-virtual-machine: ~/freedv-gui/codec2/build_linux
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_1600_tx | ./demo/freedv_1600_rx | aplay -f S16_LE
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 687,367 ms)
^CAbbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$ arecord -f S16_LE -c 1 -r 8000 | ./demo/freedv_1600mw_tx | ./demo/freedv_1600mw_rx | aplay -f S16_LE
Aufnahme: WAVE 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Wiedergabe: Rohdaten 'stdin' : Signed 16 bit Little Endian, Rate: 8000 Hz, mono
Unterlauf!!! (mindestens 271,973 ms)
Unterlauf!!! (mindestens 3723,947 ms)
^CAbbruch durch Signal Unterbrechung ...
Abbruch durch Signal Unterbrechung ...
manfred@manfred-virtual-machine:~/freedv-gui/codec2/build_linux$
```

**Funktioniert mit 3200bit/s**

Interessante Beobachtung:

Das Demo Testprogramm mit dem Mode 1600mw (codec2 3200bit/s 20 Träger) funktioniert mit guter Audioqualität, aber in der Oberfläche ./freedv hackt die Modulation

==> eventuell zu wenig Rechenleistung in der virtuellen Maschine

==> Installation dieser Version in richtigem Linux:

./freedv hackt auch

Demoprogramm mit 3200bits/s ist ok.

????????????????????????????????????