

FreeDV Treffen Ende April 2022

1. **Terminplanung, Stellenausschreibung?**
2. wir bauen uns ein eigenes **FreeDV**
 - unter Win10 für Win10
 - unter Linux für Linux (und Windows)
3. offene Diskussion

1. Terminplanung

- FreeDV Runde **jeden Sonntag**
 ab 11:00 Uhr, 40m & 80m
 ab 13:30 Uhr, 40m & 80m
- nächstes Gruppentreffen am **19. Mai**, ab **20:00** Uhr

April							
kw	Mo	Di	Mi	Do	Fr	Sa	So
13					1	2	3
14	4	5	6	7	8	9	10
15	11	12	13	14	15	16	17
16	18	19	20	21	22	23	24
17	25	26	27	28	29	30	

Mai							
kw	Mo	Di	Mi	Do	Fr	Sa	So
17							1
18	2	3	4	5	6	7	8
19	9	10	11	12	13	14	15
20	16	17	18	19	20	21	22
21	23	24	25	26	27	28	29
22	30	31					

Juni							
kw	Mo	Di	Mi	Do	Fr	Sa	So
22			1	2	3	4	5
23	6	7	8	9	10	11	12
24	13	14	15	16	17	18	19
25	20	21	22	23	24	25	26
26	27	28	29	30			

HAMRadio

nächstes Treffen in **3 Wochen**

2. Stellenausschreibung?

Wir haben mit unserer HAMgroup den Punkt erreicht, wo wir anfangen könnten, **aktiv mit zu entwickeln**

(programmieren, testen, validieren, berichten)

Umfrage:

- A. das **traue ich mir zu**, und ich hätte auch Lust drauf
- B. ich würde gerne, aber ich **weiß nicht wie**
- C. das traue ich mir **heute** noch nicht zu
- D. Anwenden reicht mir, entwickeln muss oder mag ich **nicht**

2. Stellenausschreibung?

Sollte ich vielleicht eine Stellenausschreibung

„HAMgroup sucht SW-affine Mitmacher“

in der nächsten CQDL platzieren?

Natürlich mit ein bisschen Augenzwinkern,
es gibt ja auch nichts zu verdienen,
aber dafür viel zu lernen und
neue Hams zum Kennenlernen

FreeDV Treffen Ende April 2022

1. Terminplanung, Stellenausschreibung?

2. wir bauen uns ein eigenes **FreeDV**

- unter Win10 für Win10
- unter Linux für Linux (und Windows)

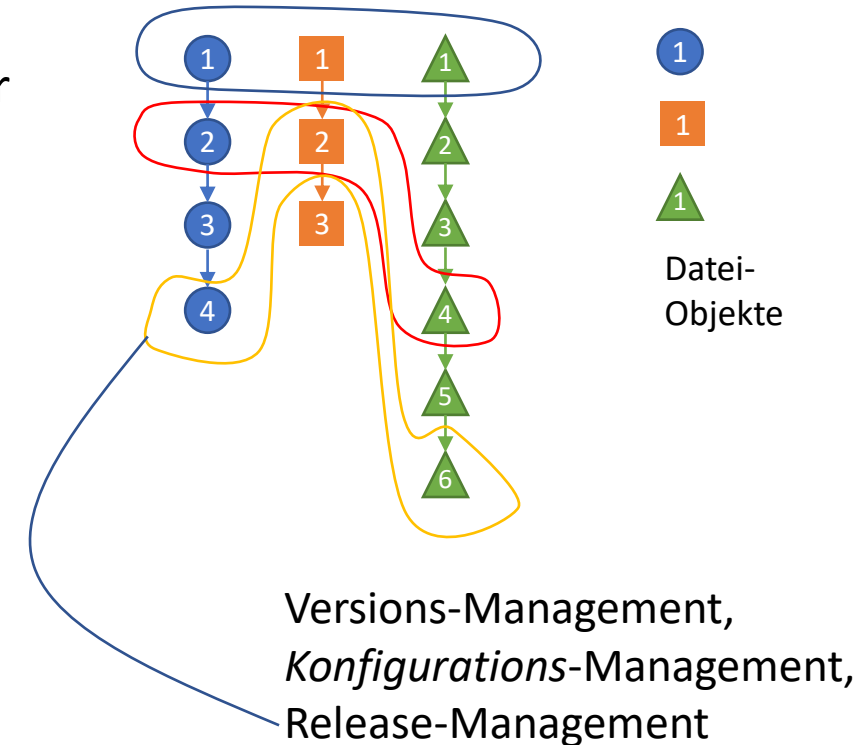
3. offene Diskussion

2. wir bauen unser eigenes FreeDV

Vorbemerkung 1/4: git


- insbesondere OpenSource Projekte verwenden oft **github** oder ähnliche „repositories“ zur Ablage ihrer Quelldaten
 - multi-user fähig
 - direkte Ansicht der Quelldateien
 - oft mit Mehrwert-Funktionen: *issues, discussions, ...*
 - die Verwaltung von Daten findet auf Einzel-Objekt-Ebene statt, Objekte werden dann zu Versionen gebündelt
 - wenn man selbst etwas verändern möchte, funktioniert das mittels *branches* (→ das behandeln wir heute nicht!)
- zu Grunde liegt ein lokal verwendbares Programm **git**, wenn's online ist, dann heißt es **github**
- github gehört heute Microsoft

- man definiert, was man herunterladen möchte, startet den Download der Quell-Dateien (*pull*)
- dann startet man den SW-Bauprozess („compilieren“)



2. wir bauen unser eigenes FreeDV

Vorbemerkung 2/4: cross und nativ

- traditionell *compiliert* man Software „**nativ**“, d.h. auf dem Zielsystem mit dem Compiler des Zielsystems und dem dazu gehörenden Betriebssystem
- typisches Beispiel Microsoft „Visual Studio C++“ auf dem Windows-PC
- Erzeugung auf „fremdem“ Betriebssystem oder CPU Architektur nennt man „**cross**-compilieren“
- typisches Bastel-Beispiel: 

→ wir werden cross-compilieren für Win10

→ vorzugsweise unter **WSL** mit Linux in einer **docker** Umgebung

→ alternativ unter Linux mit Linux in einer **docker** Umgebung

→ wir werden nativ compilieren für Linux unter Linux (ubuntu)



2. wir bauen unser eigenes FreeDV

Vorbemerkung 3/4: docker

- *docker* ist ein SW-Paket-Konzept (Paket=*container*) zur Verteilung von SW-Applikationen
- zu Grunde liegt meist der Linux Kernel eines Computers (Host-Betriebssystem)
- notwendige Erweiterungen werden dazu geladen und ergänzen oder ersetzen Teile des Host-Betriebssystems für die gewünschte Applikation
- das Host-Betriebssystem bleibt dabei aber unverändert, alles findet in Containern statt
- diese Container werden für die Anwendung „zusammen gesetzt“ (*compose*)





2. wir bauen unser eigenes FreeDV

Vorbemerkung 3/4: docker

Auszug aus [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)):

Containers are isolated from one another and bundle their own software... Because all of the containers share the services of a **single operating system kernel**, they use fewer resources than virtual machines....

The Docker software as a service offering consists of three components:

- Docker Daemon, called **dockerd**, is a persistent process that manages Docker containers and handles container objects....
- Objects: container, image, service...
- Registries: A Docker registry is a repository for Docker images. Docker clients connect to registries to download ("pull") images for use ...

Sehr praktisch:

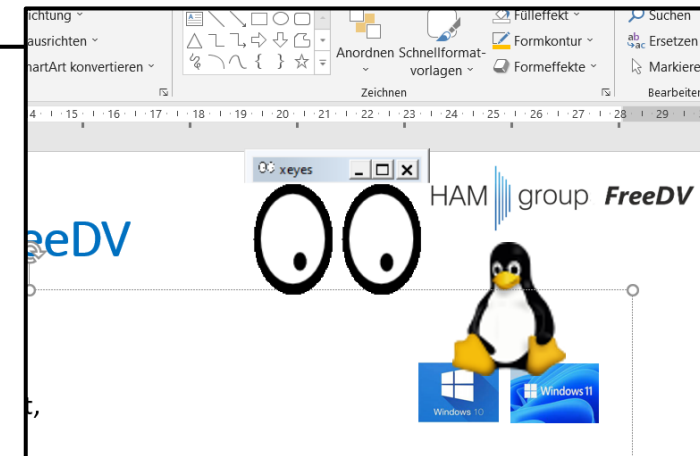
wir können uns eine komplette Compiler- und Bau-Umgebung für Windows auf Basis von docker images fix-und-fertig herunterladen, einen eigenen Container bauen und dann...

2. wir bauen unser eigenes FreeDV

Vorbemerkung 4/4: WSL

Windows Subsystem for Linux

- seit Ende 2016 bietet Windows die Möglichkeit, ein Linux Subsystem zu installieren
- das läuft dann in einem Konsolenfenster (Linux Kommandozeile)
 - mit neueren Win10 und Win11 Versionen werden auch grafische Ausgaben von linux Programmen unterstützt, d.h. original Linux Programme laufen dann unter Windows und erscheinen wie Windows Programme!?
 - alternativ auch mit xming oder ähnlichen Programmen
 - Grafik-Ausgabe brauchen wir aber heute nicht
- WSL ist für uns die „**Abkürzung**“ (=wir brauchen keine „Linux-Kiste“), um auf einem Windows PC eine unter Linux Cross-Compilierte FreeDV Version zur Ausführung unter Win10 zu erzeugen!



2. wir bauen unser eigenes FreeDV

DREI Varianten



1. Unter Windows **mit WSL** für Windows
2. Unter Linux für Windows
3. Unter Linux für Linux



2. wir bauen unser eigenes FreeDV

unter Windows mit WSL für Windows (1/4)

1. WSL installieren (<https://docs.microsoft.com/en-us/windows/wsl/install>):
 - „cmd“ ausführen
 - wsl --install (installiert das WSL Framework)
 - wsl --install --d Ubuntu-20.04 (lädt alles für ein aktuelles Ubuntu nach)
2. danach gibt es im Windows-Start Menü ein „Ubuntu 20.04“, das man direkt ausführen kann
3. *user mit password anlegen, fertig!*

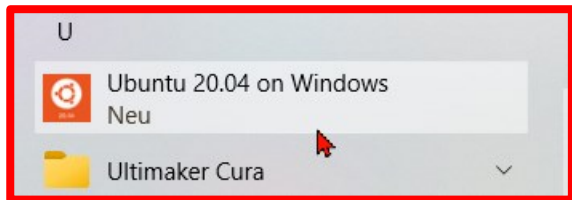
```
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>wsl --install
Installation: VM-Plattform
VM-Plattform wurde installiert.
Installation: Windows-Subsystem für Linux
Windows-Subsystem für Linux wurde installiert.
Herunterladen: WSL-Kernel
Installation: WSL-Kernel
WSL-Kernel wurde installiert.
Herunterladen: Ubuntu
[=====53,3%
```

```
C:\Windows\system32>wsl -l -o
Die folgende Liste enthält die gültigen Distributionen, die inst
Führen Sie die Installation mithilfe des Befehls „wsl --install
```

NAME	FRIENDLY NAME
Ubuntu	Ubuntu
Debian	Debian GNU/Linux
kali-linux	Kali Linux Rolling
openSUSE-42	openSUSE Leap 42
SLES-12	SUSE Linux Enterprise Server v12
Ubuntu-16.04	Ubuntu 16.04 LTS
Ubuntu-18.04	Ubuntu 18.04 LTS
Ubuntu-20.04	Ubuntu 20.04 LTS

```
C:\Windows\system32>wsl --install -d Ubuntu-20.04
Herunterladen: Ubuntu 20.04 LTS
[=====18,0%
```



```
Administrator: Eingabeaufforderung
C:\Windows\system32>wsl -l -o
Die folgende Liste enthält die gültigen Distributionen, die installiert werden können.
Führen Sie die Installation mithilfe des Befehls „wsl --install -d <Distribution>“ aus.

NAME          FRIENDLY NAME
Ubuntu        Ubuntu
Debian        Debian GNU/Linux
kali-linux    Kali Linux Rolling
openSUSE-42   openSUSE Leap 42
SLES-12       SUSE Linux Enterprise
Ubuntu-16.04  Ubuntu 16.04 LTS
Ubuntu-18.04  Ubuntu 18.04 LTS
Ubuntu-20.04  Ubuntu 20.04 LTS

C:\Windows\system32>wsl --install -d Ubuntu-20.04
Herunterladen: Ubuntu 20.04 LTS
Installation: Ubuntu 20.04 LTS
Ubuntu 20.04 LTS wurde installiert.
Ubuntu 20.04 LTS wird gestartet...

C:\Windows\system32>
```

Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to be the same as the Windows account.
For more information visit: <https://aka.ms/wslusers>
Enter new UNIX username:

2. wir bauen unser eigenes FreeDV

unter Windows mit WSL für Windows (2/4)

1. spielen mit dem **Ubuntu Terminal 1**, „df“ zeigt aktuell gemountete Festplatten an, orientieren am Verzeichnisbaum, Verzeichnis ändern („cd“), geeignetes Bauverzeichnis finden oder anlegen („mkdir“), dort hinein wechseln
2. jetzt weiter mit FreeDV Anleitung im Terminal 1
(<https://github.com/drowe67/freedv-gui>)

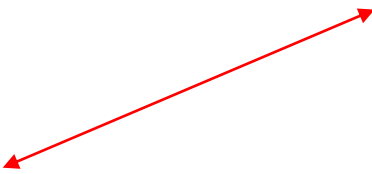
Building on Ubuntu Linux (16-20) with PortAudio

```
$ sudo apt install libspeexdsp-dev libsamplerate0-dev sox git \  
libwxgtk3.0-gtk3-dev portaudio19-dev libhamlib-dev libasound2-dev libao-dev \  
libgsm1-dev libsndfile-dev cmake module-assistant build-essential  
$ git clone https://github.com/drowe67/freedv-gui.git  
$ cd freedv-gui  
$ ./build_linux.sh portaudio
```

3. dann weiter mit Anweisungen aus „FreeDV-GUI docker“ ...

2. wir bauen unser eigenes FreeDV

unter Windows mit WSL für Windows (3/4)

1. dann weiter mit Anweisungen aus „FreeDV-GUI docker“ (https://github.com/drowe67/freedv-gui/blob/master/docker/README_docker.md)
2. `sudo apt install docker docker-compose`
3. **jetzt abweichend:**
 - ein weiteres „ubuntu“ Terminal 2 öffnen
 - dort den Docker Hintergrund-Dienst (Daemon) starten mit „`sudo dockerd`“, 
 - das Terminal 2 nicht schließen, sondern offen lassen!
 - zurück zu Terminal 1
4. `sudo usermod -aG docker $USER` (\$USER: der von Euch angelegte Nutzernamen)
5. `<log out and log in to update groups>` (=Terminal 1 schließen und neu starten via Windows Start-Menü)
6. `groups`
7. `<you should see docker as one of your groups>`
8. `docker info`
9. `docker container run hello-world` (dabei Ausgaben in Terminal1 und in Terminal2 beobachten!)

Docker installation

```
sudo apt install docker docker-compose
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
sudo usermod -aG docker $USER
<log out and log in to update groups>
groups
<you should see docker as one of your groups>
docker info
docker container run hello-world
```

2. wir bauen unser eigenes FreeDV

unter Windows mit WSL für Windows (4/4)

1. dann weiter mit Anweisungen aus „FreeDV-GUI docker“ (https://github.com/drowe67/freedv-gui/blob/master/docker/README_docker.md)
2. wie beschrieben verfahren, 32Bit oder 64Bit Variante wählen
3. im Ergebnis erhalten wir die aktuelle Release-Version „**FreeDV1.7.0**“ als Windows „msi“ Installer

Building the docker images

Building is only required once, or if you modify the docker scripts

```
cd $FREEDV_GUI/docker
docker-compose -f docker-compose-win.yml build
```

Running the docker image to build Windows installers

```
cd $FREEDV_GUI/docker
./freedv_build_windows.sh 64
./freedv_build_windows.sh 32
```

2. wir bauen unser eigenes FreeDV

DREI Varianten



1. Unter Windows **mit WSL** für Windows
2. Unter Linux für Windows
3. Unter Linux für Linux

2. wir bauen unser eigenes FreeDV

unter Linux für Windows (1/3)

1. Terminal (bash) öffnen, Verzeichnis ändern („*cd*“), geeignetes Bauverzeichnis finden oder anlegen („*mkdir*“), dann dort hinein wechseln
2. jetzt weiter mit FreeDV Anleitung (<https://github.com/drowe67/freedv-gui>)

Building on Ubuntu Linux (16-20) with PortAudio

```
$ sudo apt install libspeexdsp-dev libsamplerate0-dev sox git \  
libwxgtk3.0-gtk3-dev portaudio19-dev libhamlib-dev libasound2-dev libao-dev \  
libgsm1-dev libsndfile-dev cmake module-assistant build-essential  
$ git clone https://github.com/drowe67/freedv-gui.git  
$ cd freedv-gui  
$ ./build_linux.sh portaudio
```

3. dann weiter mit Anweisungen aus „FreeDV-GUI docker“ ...

2. wir bauen unser eigenes FreeDV

unter Linux für Windows (2/3)

1. dann weiter mit Anweisungen aus „FreeDV-GUI docker“ (https://github.com/drowe67/freedv-gui/blob/master/docker/README_docker.md)
2. *sudo apt install docker docker-compose*
3. *sudo usermod -aG docker \$USER* (\$USER: der von Euch verwendete Nutzernamen)
4. *<log out and log in to update groups>*
5. *groups*
6. *<you should see docker as one of your groups>*
7. *docker info*
8. *docker container run hello-world*

Docker installation

```
sudo apt install docker docker-compose
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
sudo usermod -aG docker $USER
<log out and log in to update groups>
groups
<you should see docker as one of your groups>
docker info
docker container run hello-world
```

2. wir bauen unser eigenes FreeDV

unter Linux für Windows (3/3)

1. dann weiter mit Anweisungen aus „FreeDV-GUI docker“ (https://github.com/drowe67/freedv-gui/blob/master/docker/README_docker.md)
2. wie beschrieben verfahren, 32Bit oder 64Bit Variante wählen
3. im Ergebnis erhalten wir die aktuelle Release-Version „**FreeDV1.7.0**“ als Windows „msi“ Installer

Building the docker images

Building is only required once, or if you modify the docker scripts

```
cd $FREEDV_GUI/docker
docker-compose -f docker-compose-win.yml build
```

Running the docker image to build Windows installers

```
cd $FREEDV_GUI/docker
./freedv_build_windows.sh 64
./freedv_build_windows.sh 32
```

2. wir bauen unser eigenes FreeDV

DREI Varianten

1. Unter Windows **mit WSL** für Windows
2. Unter Linux für Windows
3. Unter Linux für Linux



Original-Beschreibung befolgen (die braucht keine weiteren Kommentare):

<https://github.com/drowe67/freedv-gui>



2. wir bauen unser eigenes FreeDV

DREI Varianten

1. Unter Windows **mit WSL** für Windows
2. Unter Linux für Windows
3. Unter Linux für Linux

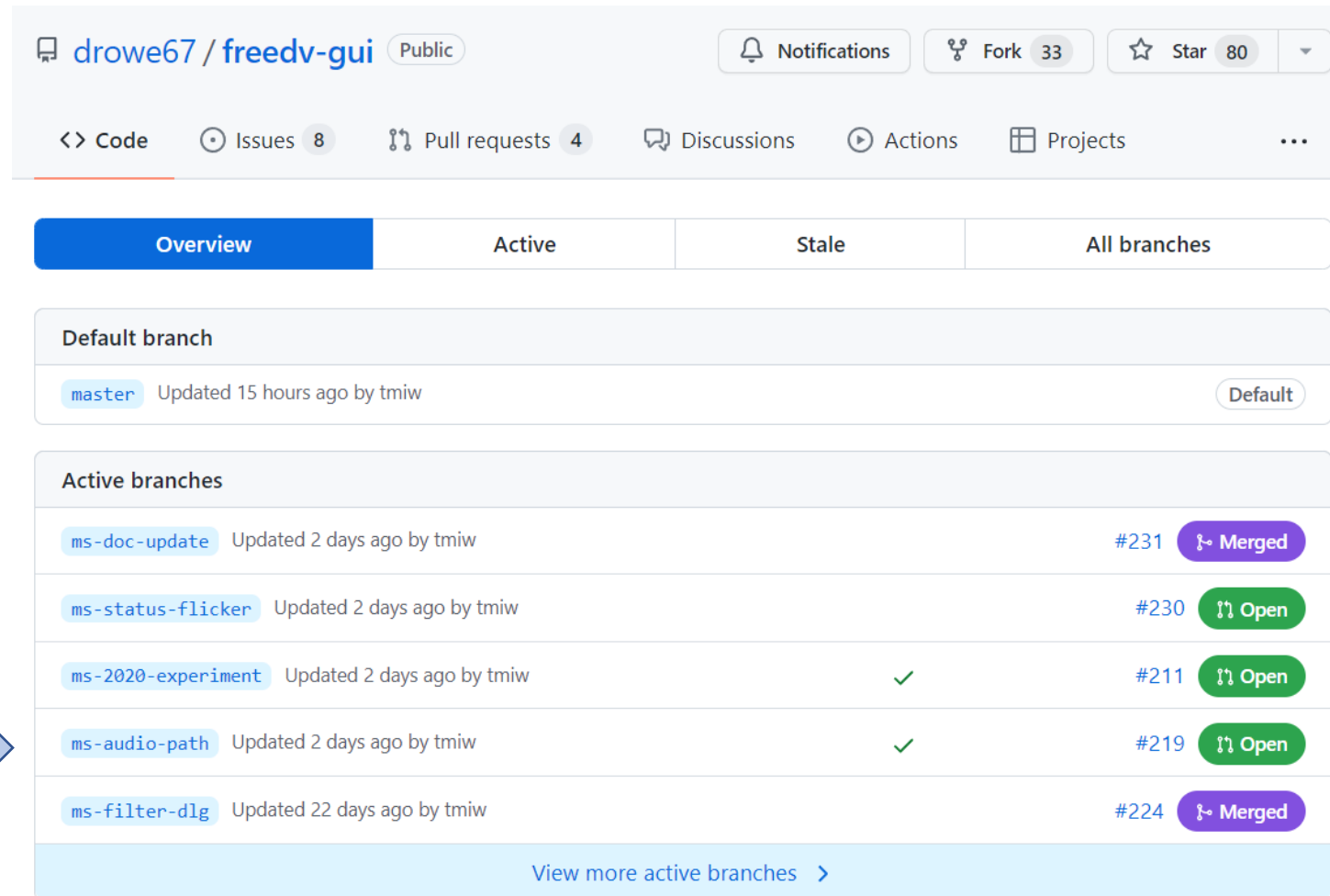
So, und jetzt müssen wir die verwendeten Quellen verändern, d.h. wir müssen uns einen **anderen github-Archiv-Stand** „pull“-en



2. wir bauen unser eigenes FreeDV

So, und jetzt müssen wir die verwendeten Quellen verändern, d.h. wir müssen uns einen **anderen github-Archiv-Stand** „pullen“

<https://github.com/drowe67/freedv-gui/branches>



drowe67 / freedv-gui Public

Notifications Fork 33 Star 80

Code Issues 8 Pull requests 4 Discussions Actions Projects

Overview Active Stale All branches

Default branch

master Updated 15 hours ago by tmiw Default

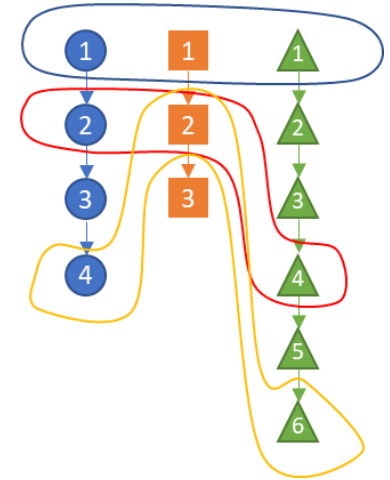
Active branches

ms-doc-update	Updated 2 days ago by tmiw	#231	Merged
ms-status-flicker	Updated 2 days ago by tmiw	#230	Open
ms-2020-experiment	Updated 2 days ago by tmiw	#211	Open
ms-audio-path	Updated 2 days ago by tmiw	#219	Open
ms-filter-dlg	Updated 22 days ago by tmiw	#224	Merged

View more active branches >

2. wir bauen unser eigenes FreeDV

- ins Bau-Verzeichnis wechseln
- versuchsweise mal „*git branch*“ ausführen (liefert z.B. „* *master*“)
- jetzt wollen wir aber etwas anderes als „* *master*“
- vorher: „*git clone https://github.com/drowe67/freedv-gui.git*“
- jetzt: „*git clone --branch ms-2020-experiment https://github.com/drowe67/freedv-gui.git*“
besorgt uns die Quelldateien mit den zum Branch passenden Versionsstand



- **aber Achtung:**
in den Build Scripts taucht auch der Name des Branches auf, mindestens ein Script muss auch aktualisiert werden nach ersten *git clone* des Wunsch-Branches (s. Screenshot)



```
Auswählen andreas@DESKTOP-9MEMEBA: /mnt/d/FUNK/FreeDV/build/win/freedv-gui/docker
GNU nano 4.8                                freedv_build_windows.sh
echo "    --bootstrap-wx    Builds wxWidgets from source (may take significantly longer to complete)"
echo
exit
}

# defaults - these variables are passed to the docker container
FDV_CLEAN=1
FDV_BUILD=0
FDV_REBUILD=0
FDV_GIT_REPO=https://github.com/drowe67/freedv-gui.git
FDV_GIT_BRANCH=ms-2020-experiment
FDV_BOOTSTRAP_WX=0

POSITIONAL=()
while [[ $# -gt 0 ]]; do
^G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo      M-A
^X Exit        ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line  M-E Redo      M-6
```

2. wir bauen unser eigenes FreeDV

- ins Bau-Verzeichnis wechseln
- versuchsweise mal „*git branch*“ ausführen (liefert z.B. „* *master*“)
- jetzt wollen wir aber etwas anderes als „* *master*“
- vorher: „*git clone https://github.com/drowe67/freedv-gui.git*“
- jetzt: „*git clone --branch ms-2020-experiment https://github.com/drowe67/freedv-gui.git*“
besorgt uns die Quelldateien mit den **zum Branch passenden Versionsstand**
- dann mit modifiziertem Build-Script, aber genauso wie vorher starten
- Linux für Linux
- oder für Windows als Docker Build (bei **WSL** braucht es dann wieder den Docker Daemon händisch gestartet im zweiten Terminal!)



```
Building the docker images

Building is only required once, or if you modify the docker scripts

cd $FREEDV_GUI/docker
docker-compose -f docker-compose-win.yml build

Running the docker image to build Windows installers

cd $FREEDV_GUI/docker
./freedv_build_windows.sh 64
./freedv_build_windows.sh 32
```

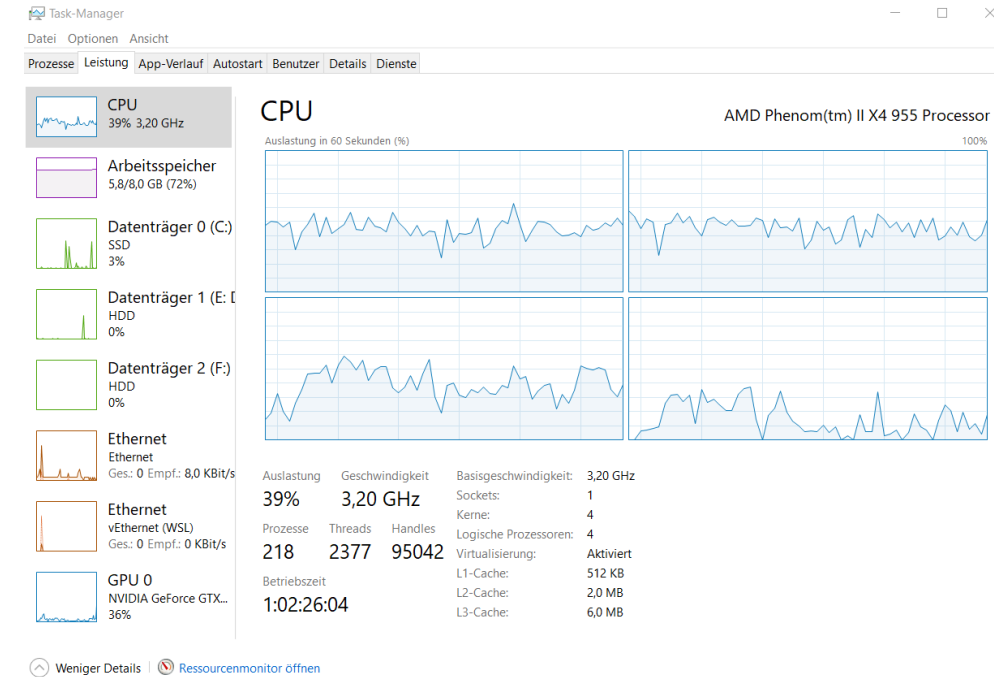

2. wir bauen unser eigenes FreeDV

Verlauf und Ergebnis von FreeDV Builds:

- Dauer Linux-nativ für Linux auf meinem alten Desktop etwa 6 Minuten
- Dauer WSL Build etwa 10 Minuten, Prozessor nicht wirklich ausgelastet (Win10 Task Scheduler Problem?)

WSL Build:

```
andreas@DESKTOP-9MEMEBA:/mnt/d/FUNK/FreeDV/build/win/freedv-gui/docker$ date
Tue Apr 26 22:32:08 CEST 2022
andreas@DESKTOP-9MEMEBA:/mnt/d/FUNK/FreeDV/build/win/freedv-gui/docker$ ./freedv_build_windows.sh 64 && date
Starting fdv_win_fed34_c ... done
/home/build/freedv-gui/build_win64/FreeDV-1.8.0-devel-20220426-75190e1-win64.exe
Tue Apr 26 22:42:30 CEST 2022
andreas@DESKTOP-9MEMEBA:/mnt/d/FUNK/FreeDV/build/win/freedv-gui/docker$ ll
total 31840
drwxrwxrwx 1 andreas andreas      512 Apr 26 22:42 /
drwxrwxrwx 1 andreas andreas      512 Apr 26 22:16 /
-rwxrwxrwx 1 andreas andreas 32056197 Apr 26 22:42 FreeDV-1.8.0-devel-20220426-75190e1-win64.exe*
-rwxrwxrwx 1 andreas andreas      784 Apr 24 16:49 README_docker.md*
-rwxrwxrwx 1 andreas andreas  533878 Apr 26 22:42 build_log.txt*
-rwxrwxrwx 1 andreas andreas      396 Apr 24 16:49 docker-compose-win.yml*
drwxrwxrwx 1 andreas andreas      512 Apr 24 16:49 fdv_win_fedora/
-rwxrwxrwx 1 andreas andreas    2726 Apr 24 21:31 freedv_build_windows.sh*
andreas@DESKTOP-9MEMEBA:/mnt/d/FUNK/FreeDV/build/win/freedv-gui/docker$
```



2. wir bauen unser eigenes FreeDV

Editieren im Quellen-Baum

- die SW Gesamt-Architektur ist etwas kompliziert
- im Rahmen des Build Prozesses wird *LPCNet* und *codec2* gebaut
- das Hauptprogram ist allerdings *freedv-gui*
- „*main.cpp*“ enthält *main()*
- hier könnte man jetzt editieren, aber das funktioniert nicht gut ohne einen eigenen github Branch, und das ist für heute Abend zu kompliziert...

