

# FreeDV Treffen Ende Juni 2022 (Teil 2)

1. kurzes Willkommen, Status letzte Funkerfahrten (10 Minuten)
2. **Fragen** und Antworten zu FreeDV (max. 20 Minuten)
3. nächster Termin

## 4. FreeDV Entwicklung

git

**github**

Code-Editoren

Kommando-Zeile

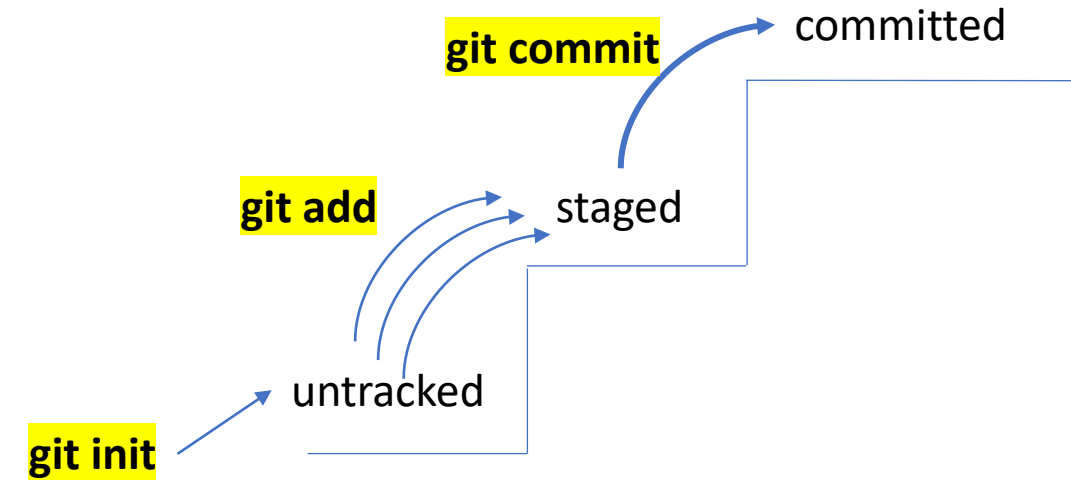
web-site

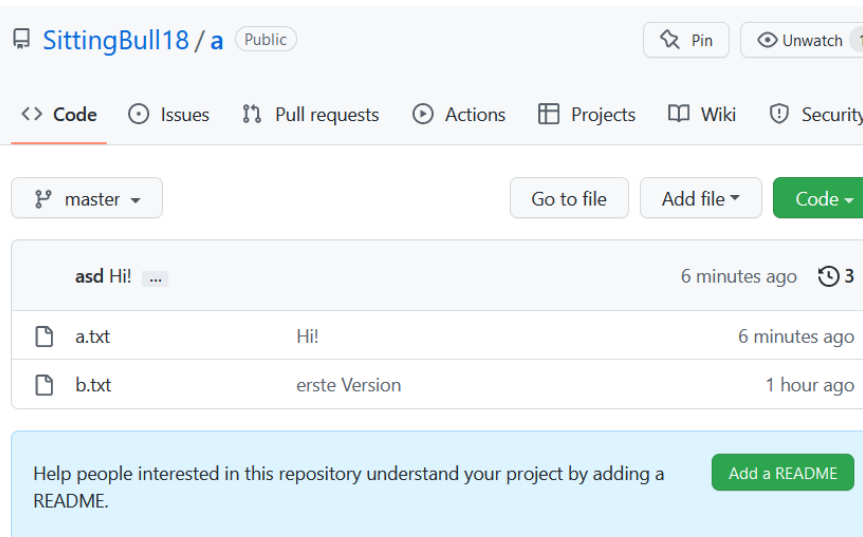
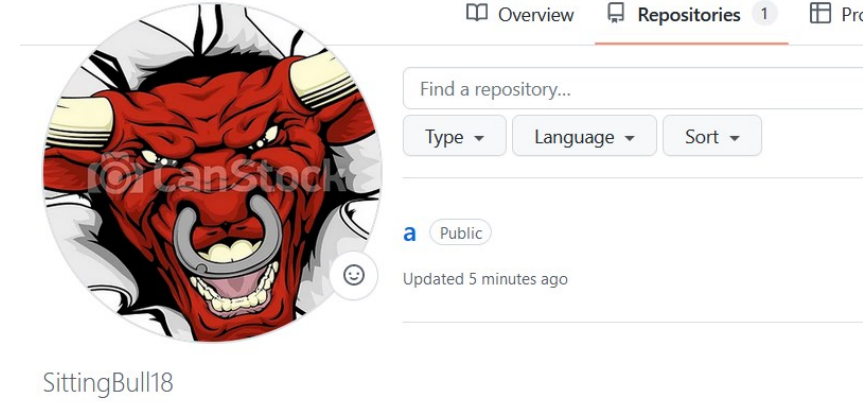
## 4. git: lokal und online

- git beobachtet alle Dateien im Verzeichnis und kennt:

- untracked files
- staged files (=files to be committed)
- committed files

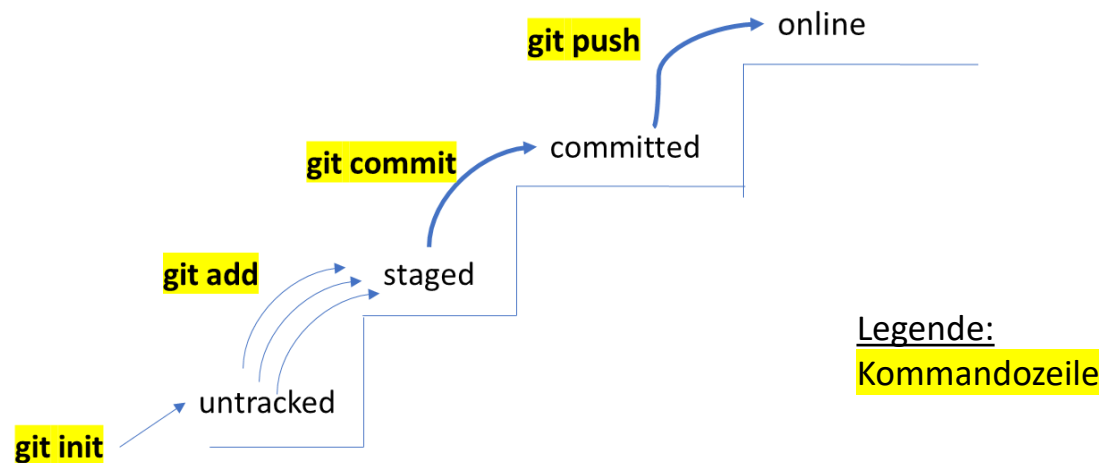
- „**git add**“ übernimmt den aktuellen Dateiinhalt in einen „Vor-Freigabe-Bereich“ (staging area)
- „git add“ kann mehrmals ausgeführt werden, um weitere Dateien in die staging area zu übernehmen
- ein erneutes „git add“ mit einer bereits hinzugefügten Datei aktualisiert diese in der staging area
- „**git commit -m „Kommentar“**“ nimmt alle Dateien gemeinsam aus der staging area und fügt sie zu den freigegebenen Dateien hinzu, nur sinnvoll mit einem Kommentar und **Nutzernamen**
- „**git log**“  
„**git log -p**“  
„**git status**“  
sagt etwas über den aktuellen Verzeichnis-Status,  
liefert mehr Details über die Änderungen an einzelnen Dateien,  
ist immer auch eine gute Idee





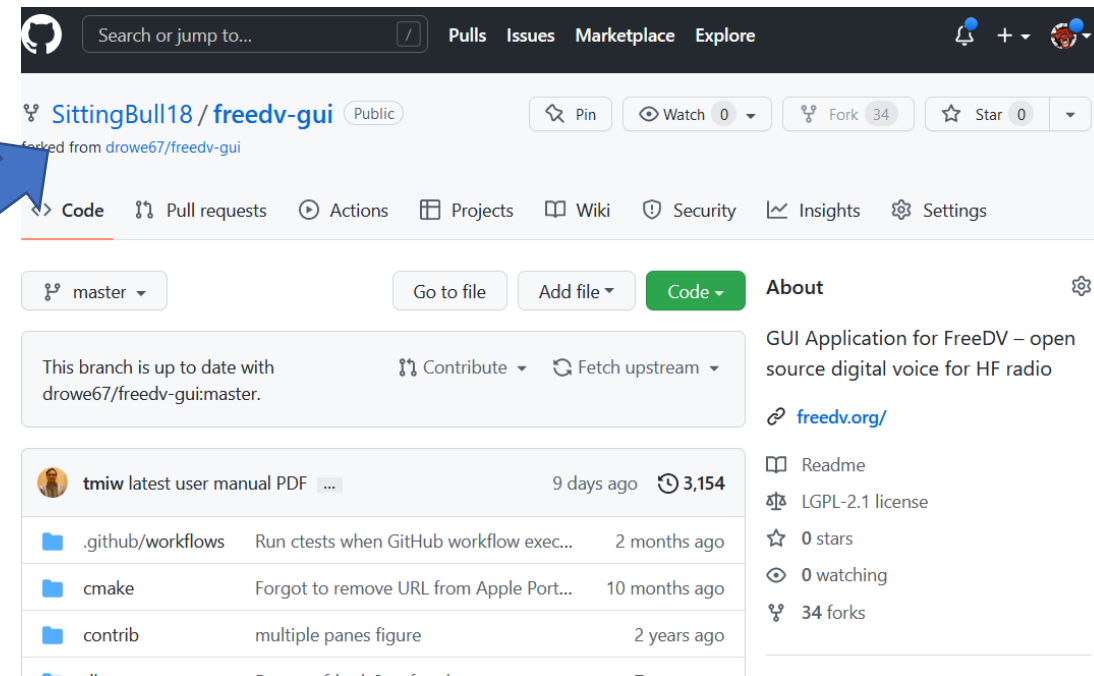
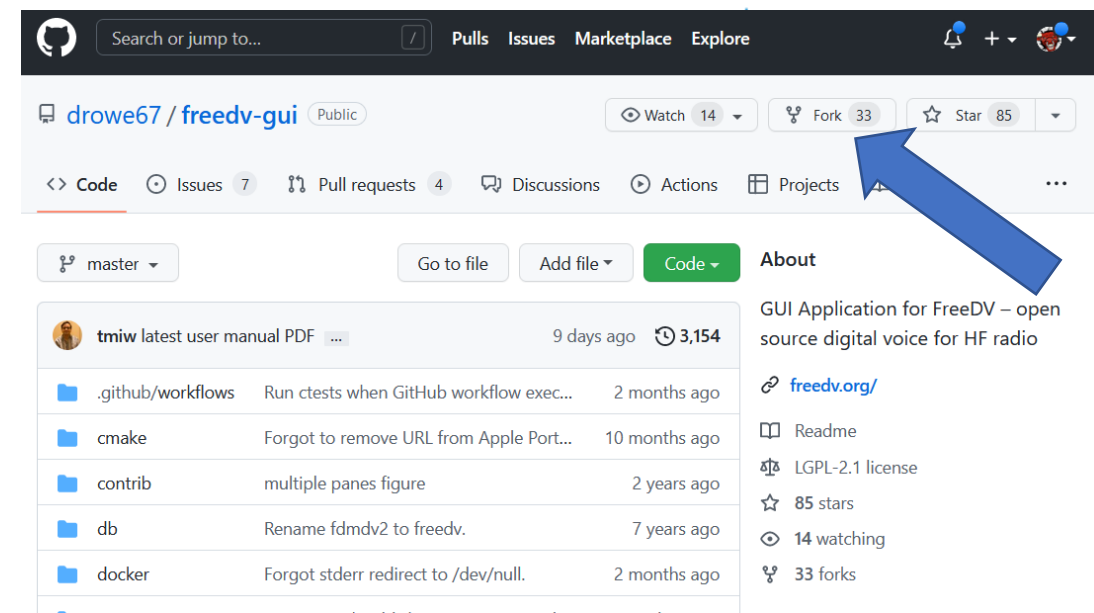
## 4. git: lokal und online

- „**git push**“ erzeugt einen Datentransfer zum angegebenen Server (hier github)
- „**git clone**“ ist das Pendant dazu und transferiert das angegebene Repo auf den lokalen Rechner (legt ein neues Verzeichnis mit dem entsprechenden **Github Account** an)
- das ge-clonte Verzeichnis funktioniert wie zuvor („git status“, ...)
- wer Schreibrechte für das Repo hat, kann direkt veränderte Dateien ins Repo push-en
- das probieren wir direkt mal aus



## 4. git: lokal und online

- „**git clone**“ funktioniert mit allen öffentlichen Repos (es gibt auch private Repos)
- „**git push**“ funktioniert nur bei Repos mit eingeräumten Schreibrechten, braucht Authentifizierung, und das macht auch Sinn, sonst geht ja alles drunter und drüber!
- wie arbeitet man dann zusammen?
- indem man ein öffentliches Repo „**forked**“
- auf „meinen **fork**“ habe ich beliebigen Zugriff, kann mir meinen **fork** klonen, editieren und pushen
- ge-push-te Änderungen gebe ich dann mit einem **Pull-Request** zurück an den Verwalter des Quell-Repos zur Durchsicht und Freigabe



## 4. git: lokal und online



Nutzername

Was wir beim letzten mal noch lernten:


```
>> git commit -m "zweiter Commit"
```

\*\*\* Bitte geben Sie an, wer Sie sind.

Führen Sie

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

aus, um das als Ihre standardmäßige Identität zu setzen.  
Lassen Sie die Option "--global" weg, um die Identität nur  
für dieses Repository zu setzen.



```
git commit -m "zweiter Commit"
[master 0373265] zweiter Commit
1 file changed, 2 insertions(+), 1 deletion(-)
```

### Lösung:

```
>> git config --global user.email "dm4ab@darcc.de"
>> git config --global user.name "cybork42"
```

wird gespeichert in...

```
cat ~/.gitconfig
[credential]
    helper =
    helper = /usr/local/share/gcm-core/git-credential-manager-core
    credentialStore = cache
[credential "https://dev.azure.com"]
    useHttpPath = true

[user]
    email = dm4ab@darcc.de
    name = cybork42
```

## 4. git: lokal und online



Github Account



GCM

Wie funktioniert das mit dem Datentransfer und der Authentifizierung?

- entweder arbeitet man mit einem personal access token (PAT), den man sich auf der github Web-Seite besorgt → geht, läuft aber nach < 90 Tagen ab ☹️
- oder man installiert sich den **git credential manager (GCM)**  
git verwendet diesen (verdeckt),  
um bei Sitzungsbeginn eine Authentifizierung mit github durchzuführen und sich diese vorübergehend zu merken
  - wird bei Windows automatisch installiert
  - braucht für Linux einige Extra-Aktivitäten
- <https://github.com/GitCredentialManager/git-credential-manager>
- das neueste passende Download-Package finden und herunterladen, z.B.:  
`wget https://github.com/GitCredentialManager/git-credential-manager/releases/download/v2.0.696/gcmcore-linux\_amd64.2.0.696.deb`
- und installieren z.B. mit `sudo dpkg -i gcmcore-linux_amd64.2.0.696.deb`
- danach:  

```
>> git-credential-manager-core configure
Configuring component 'Git Credential Manager'...
Configuring component 'Azure Repos provider'...
```

## 4. git: lokal und online



**Github Account**



**GCM**

- wir versuchen:

```
>> git push --set-upstream https://github.com/cybork42/test master
```

```
fatal: No credential store has been selected.
```

```
Set the GCM_CREDENTIAL_STORE environment variable or the credential.credentialStore Git config
```

```
secretservice : freedesktop.org Secret Service (requires graphical interface)
gpg            : GNU `pass` compatible credential storage (requires GPG and `pass`)
cache         : Git's in-memory credential cache
plaintext     : store credentials in plain-text files (UNSECURE)
```

```
See https://aka.ms/gcm/credstores for more information.
```

```
Username for 'https://github.com': dm4ab@darc.de
Password for 'https://dm4ab@darc.de@github.com':
fatal: No credential store has been selected.
```

- Abhilfe: >> *git config - -global credential.credentialStore cache*
- danach:

```
>> git push --set-upstream https://github.com/cybork42/test master
```

```
Zähle Objekte: 3, Fertig.
```

```
Schreibe Objekte: 100% (3/3), 256 bytes | 256.00 KiB/s, Fertig.
```

```
Total 3 (delta 0), reused 0 (delta 0)
```

```
To https://github.com/cybork42/test
```

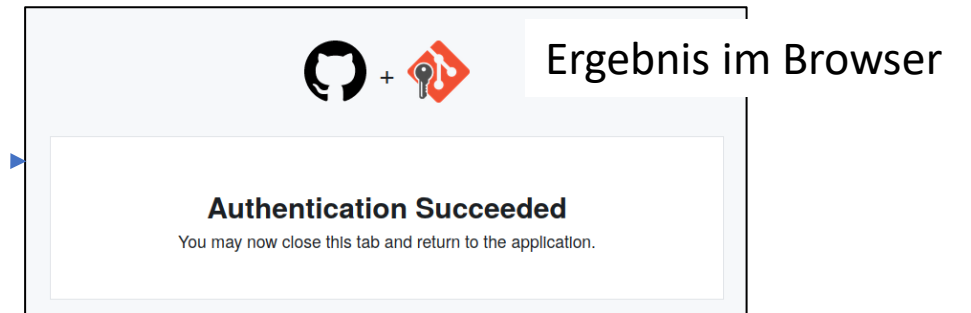
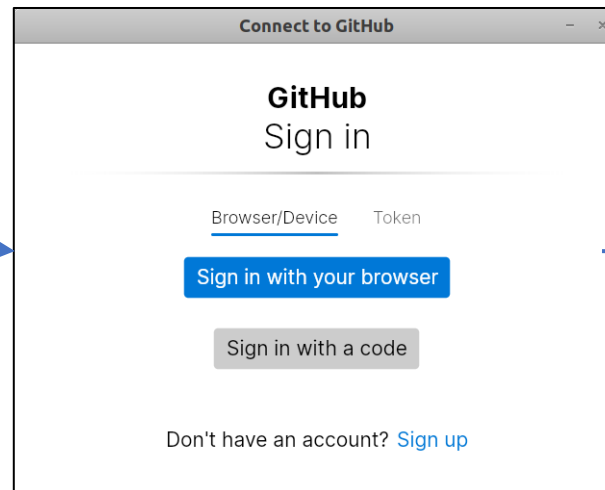
```
be07b40..0373265 master -> master
```

## 4. git: lokal und online

Github Account

GCM

- erneut: `>> git push --set-upstream https://github.com/cybork42/test master`



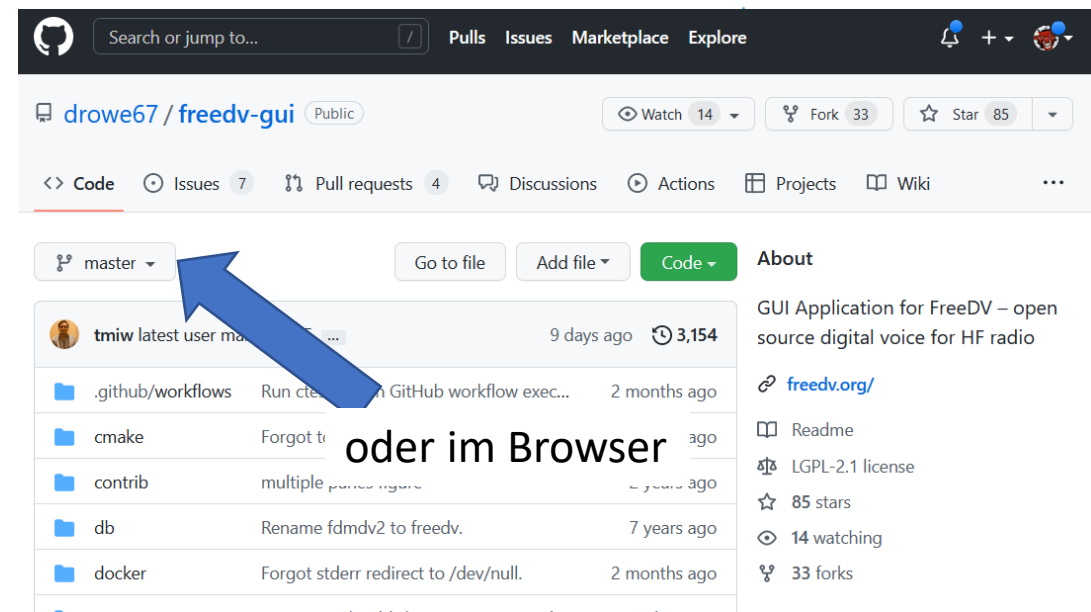
einmal nach dem Booten  
oder nach Sitzungsbeginn

Zähle Objekte: 3, Fertig.  
Schreibe Objekte: 100% (3/3), 256 bytes | 256.00 KiB/s, Fertig.  
Total 3 (delta 0), reused 0 (delta 0)  
To https://github.com/cybork42/test  
be07b40..0373265 master -> master



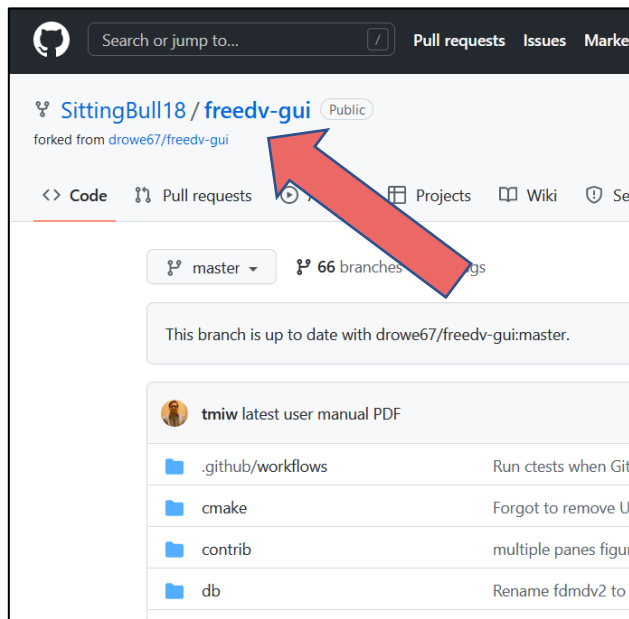
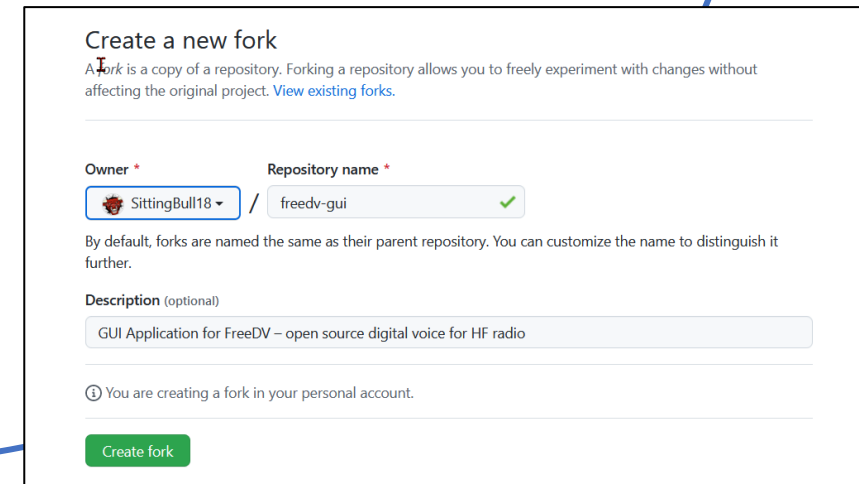
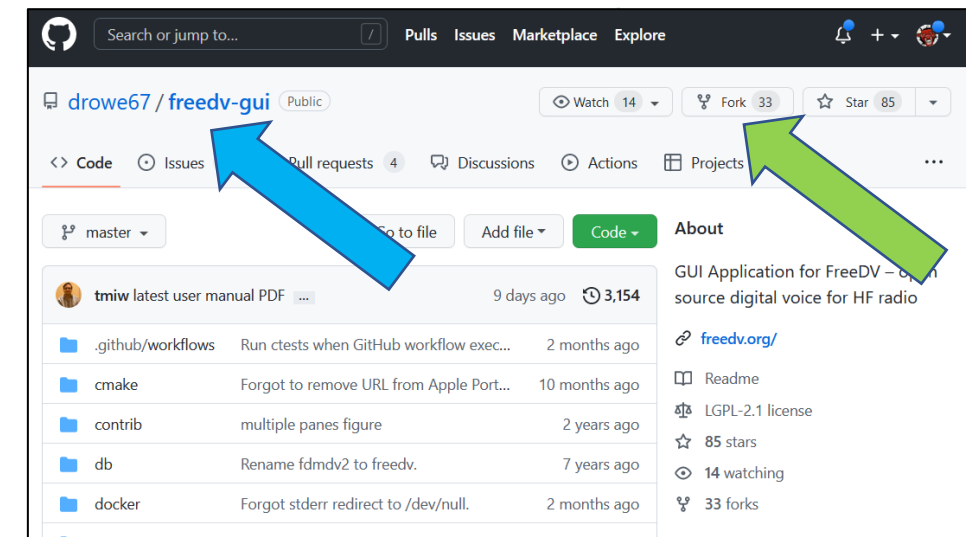
## 4. git: lokal und online

- „**branch**“ gibt es auch noch
- das ist hilfreich für verschiedene, parallel laufende Entwicklungen in einem Repo, die sich nicht direkt beeinflussen sollen
- „**git branch - - list**“ zeigt die verfügbaren Branches des gewählten Repos an
- oder auch sichtbar auf **github**
- „**git checkout branch**“ wählt einen bestimmten Branch aus, danach braucht es wieder ein „**git clone**“ um die entsprechenden Dateistände herunter zu laden



## 4. git: lokal und online

- an einem **Original Repo** habt Ihr normalerweise keine Schreibrechte, sondern nur an eigenen **Repos**
- um aber an einem Original Repo mit zu entwickeln, gibt es die Möglichkeit, dieses zu „**forken**“
- dann habt Ihr eine **eigene Kopie**



## 4. git: lokal und online

- Ihr push Eure Änderungen in Euren eigenen Fork
- danach könnt Ihr eine Reihe von Änderungen zusammen packen und einen **Pull Request** anlegen (=damit bittet Ihr den Owner des Repositories, Eure Änderungen zu prüfen und nach Möglichkeit zu übernehmen).

```
>> git clone https://github.com/SittingBull18/freedv-gui
```

(Änderungen durchführen)

```
>> git status
```

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

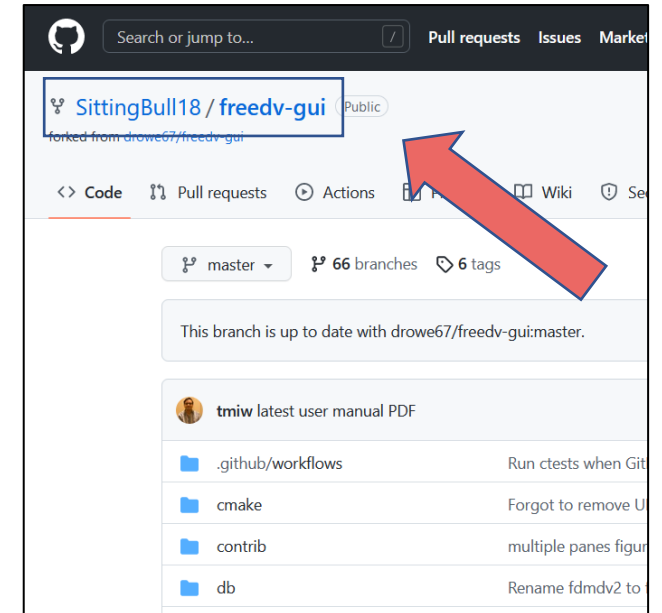
modified: **README.md**

no changes added to commit (use "git add" and/or "git commit -a")

```
>> git add README.md
```

```
>> git commit -m "Readme.md editiert"
```

```
>> git push https://github.com/SittingBull18/freedv-gui
```



# 4. git: lokal und online

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: drowe67/freedv-gui base: master ← head repository: SittingBull18/freedv-gui compare: master

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) [Create pull request](#)

1 commit 1 file changed 1 contributor

Commits on Jun 30, 2022

**Readme.md editiert**  
asd committed 4 minutes ago

SittingBull18 / freedv-gui Public  
forked from drowe67/freedv-gui

<> Code Pull requests Actions Projects Wiki Security

master 66 branches 6 tags

This branch is 1 commit ahead of drowe67:master.

asd Readme.md editiert

.github/workflows

cmake

Showing 1 changed file with 2 additions and 0 deletions.

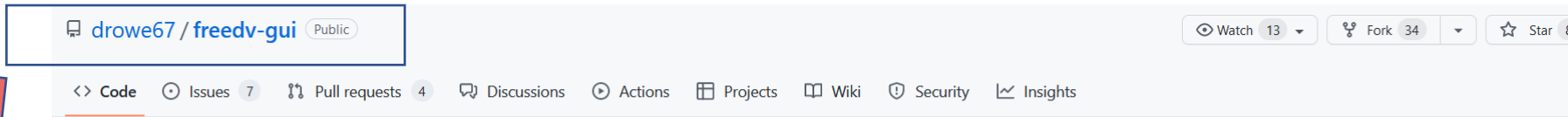
2 README.md

```
@@ -1,3 +1,5 @@
1 + Andreas hat hier etwas editiert!
2 +
1 3 # Building FreeDV GUI
2 4
3 5 This document describes how to build the FreeDV GUI program for various operating systems. FreeDV GUI is developed on Ubuntu Linux, and then cross compiled for Windows using Fedora Linux (Fedora has great cross compiling support) and Docker.
```

initiative des



## 4. git: lokal und online






























### Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: drowe67/freedv-gui base: master ← head repository: SittingBull18/freedv-gui compare: master

✓ **Able to merge.** These branches can be automatically merged.

 Readme.md editiert


Write Preview H B I                          

**\*\*Just a test!\*\***

Attach files by dragging & dropping, selecting or pasting them.

☒ **Allow edits by maintainers** ?



[Helpful resources](#)  
[GitHub Community Guidelines](#)

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

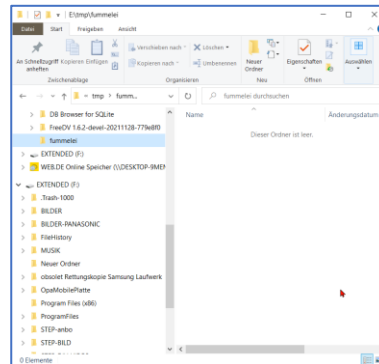
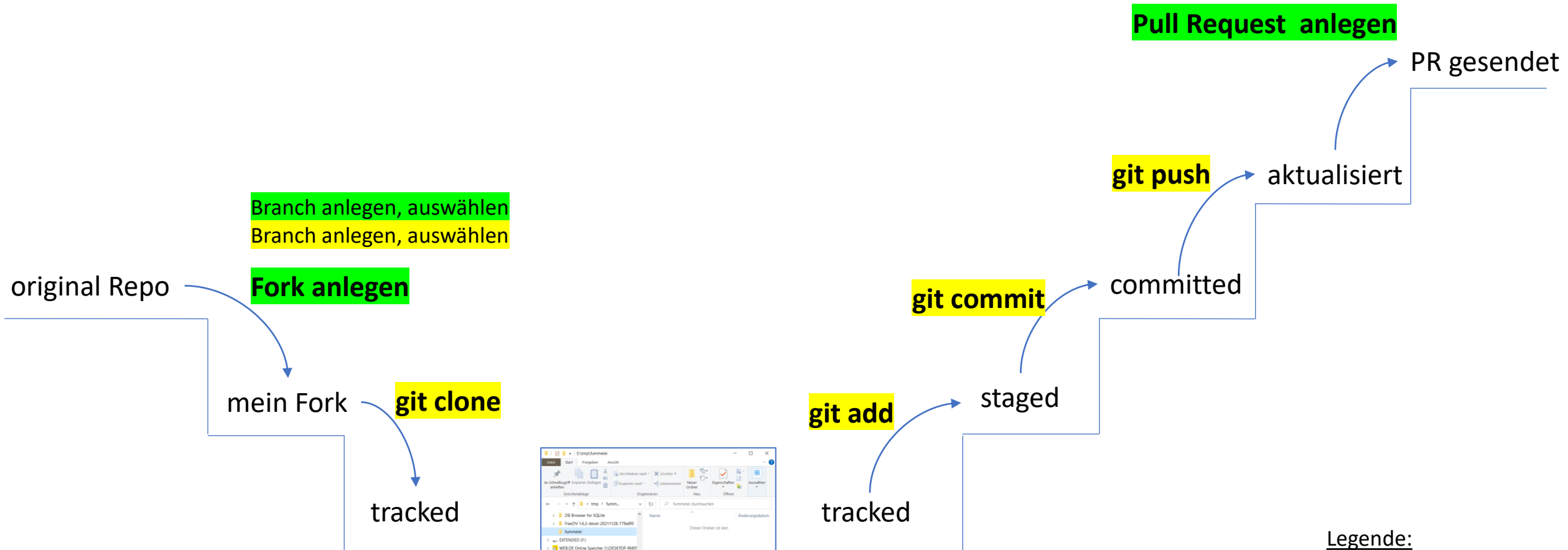
1 commit 1 file changed

Commits on Jun 30, 2022

**Readme.md editiert**  
asd committed 5 minutes ago

 a70652b 

## 4. git: lokal und online



Legende:  
github  
Kommandozeile