

FreeDV Treffen Mitte Juli 2022

1. kurzes Willkommen, Status letzte Funkerfahrten 40m & 80m (10 Minuten)
2. **Fragen** und Antworten zu FreeDV (max. 20 Minuten), nächster Termin

3. **FreeDV Entwicklung**

SW Struktur: Bauprozedur, github Archive, Änderungen compilieren
wxWidgets, multi-threading, Fenster-Aufbau

3. Terminplanung

- FreeDV Runde **jeden Sonntag**
ab 11:00 Uhr, 40m & 80m
ab 13:30 Uhr, 40m & 80m
- für unsere Gruppentreffen ist der typische Wochentag **DONNERSTAG**

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

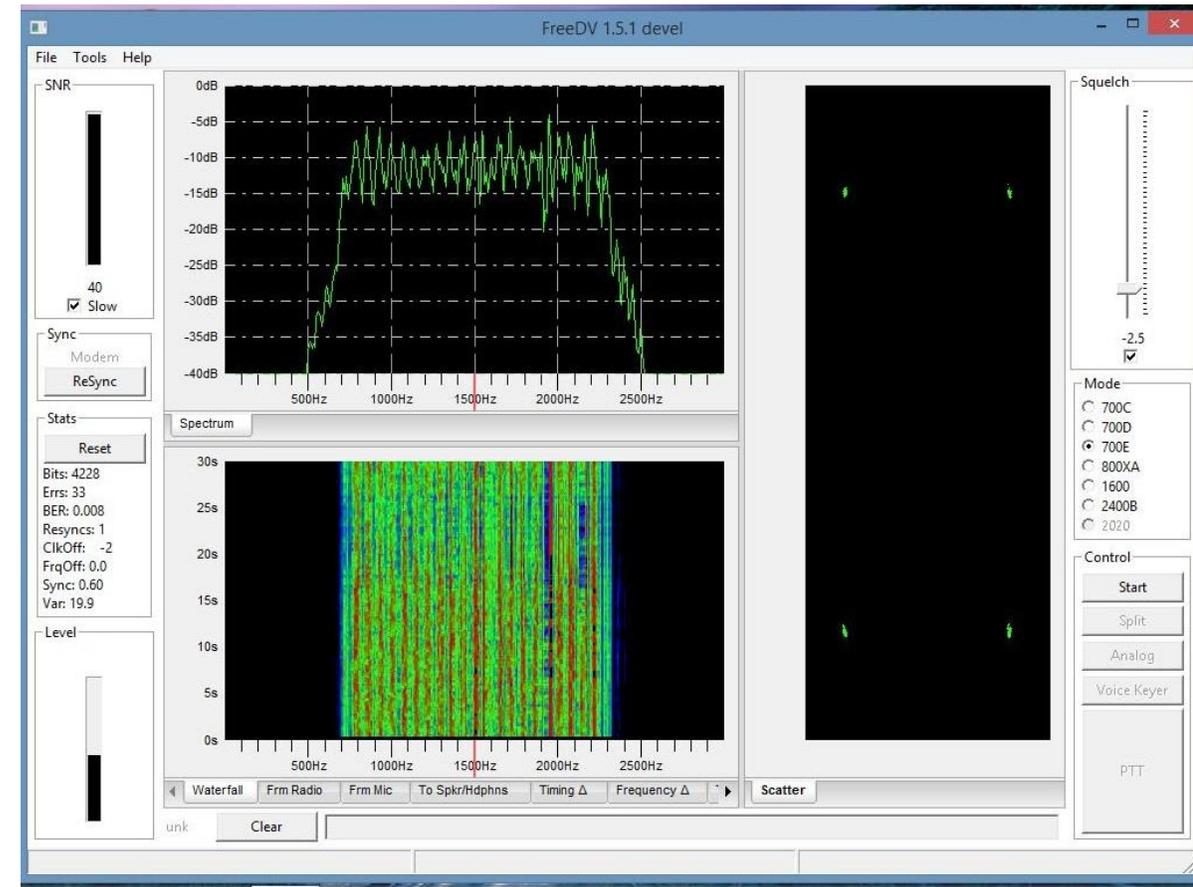
August							
kw	Mo	Di	Mi	Do	Fr	Sa	So
31	1	2	3	4	5	6	7
32	8	9	10	11	12	13	14
33	15	16	17	18	19	20	21
34	22	23	24	25	26	27	28
35	29	30	31				

nächstes Treffen in **3 Wochen**

→ **Donnerstag, 4. August 2022, 20:00 Uhr**

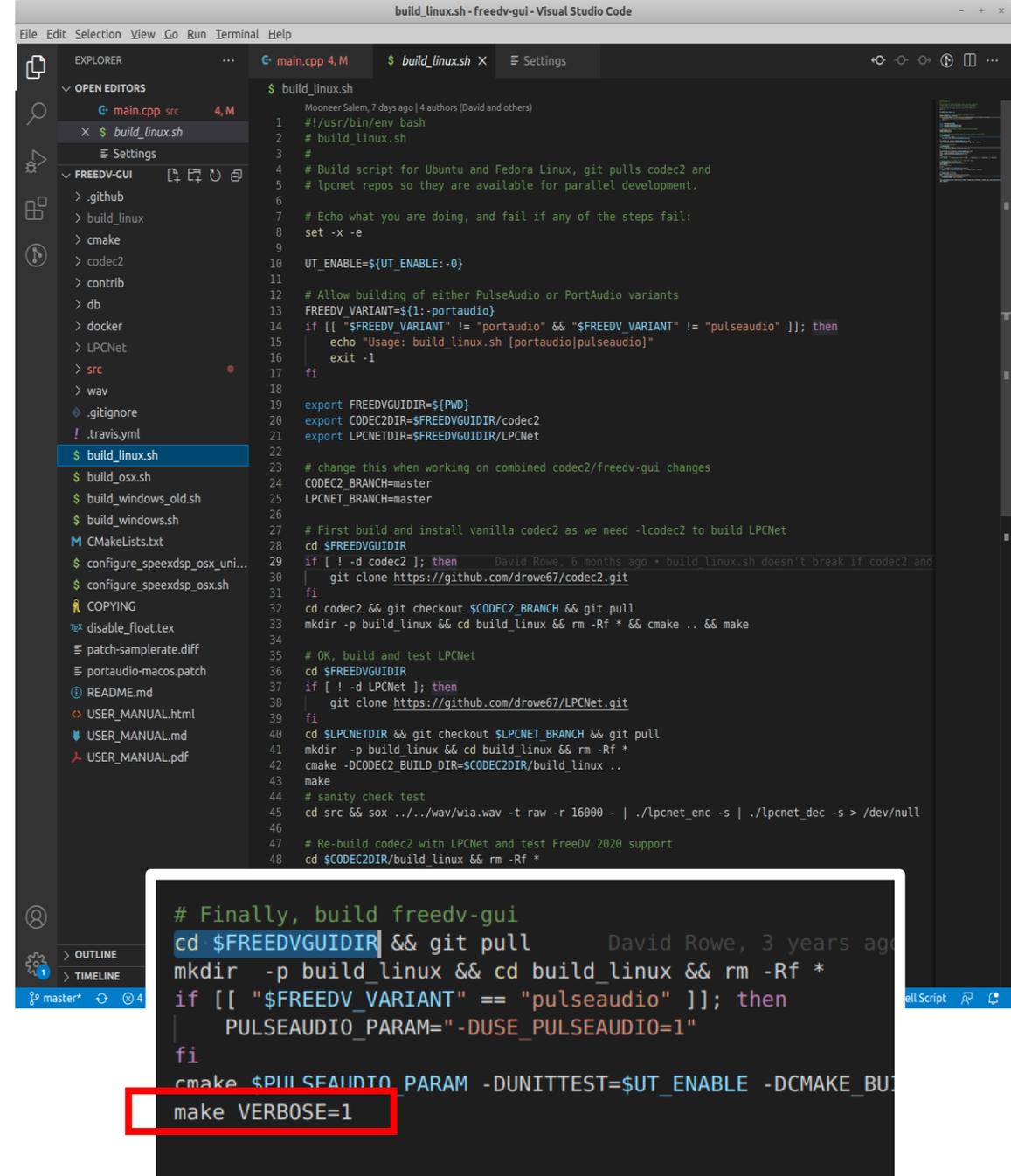
4. SW Struktur

- **Start** mit dem „git clone“ der freedv-gui Quellen
- weiter mit einem sympatischen Editor („nano“ im Terminal, sublime Text, Visual Studio Code)
- werft mal einen Blick in „*Readme.md*“
- das ist nur Text mit wenig Formatierungs-Hinweisen und sieht im Ergebnis erstaunlich gut aus!
- und lest man „*USER_MANUAL.md*“ mit einer sehr guten Zusammenfassung zu FreeDV



4. SW Struktur

- check von *build_linux.sh*
- zusätzlich „git clone“ von *codec2* und *LPCNet*
- falls das Verzeichnis nicht existiert:
 - git clonewenn das Verzeichnis schon existiert:
 - *git checkout && git pull*
(aktualisiert ohne lokale Änderungen zu überschreiben)
- „./*build_linux.sh*“ führt man typischerweise nur einmal aus
- einen erneuten Compilier-Vorgang startet am besten im richtigen Unterverzeichnis mit „*make*“



4. SW Struktur

wie funktioniert eine GUI basierte Software? (1/2)

- Event-basiert
- definierte Fenster und Dialoge mit einzelnen Schaltflächen und Anzeige-Elementen (widgets)
- mit den Elementen sind Funktionen verbunden (wird ein Element aktiviert, wird die Funktion ausgelöst)

- Dialog-Fenster (Pop-Ups) sind fertig definiert und werden nur angezeigt oder wieder „versteckt“
- das Hauptprogramm ist im Prinzip eine leere Schleife,
alles passiert nach einem **Maus-Klick** oder nach Ablauf eines **Timers**

- wichtig: mehrere Prozesse innerhalb eines Programms (oder auch Threads)
Prozess: „parallele Ausführung“, gleiches Programm aber **isolierter** Speicherbereich
Thread: „parallele Ausführung“, gleiches Programm im **gleichen** Speicherbereich

4. SW Struktur

wie funktioniert eine GUI basierte Software? (2/2)

- Fenster-Inhalte sind aus einzelnen kleinen Inhalten platziert und werden miteinander angeordnet; verschiedene Toolkits haben hier jeweils eigene Methoden
- FreeDV verwendet wxWidgets; hier werden verschiedene „sizer“ verwendet
- Beispiel: dlp_ptt.cpp
- Code-Modifikation für Modus 2020
- gdb zum Debuggen
- ...