Final Commit: f022f11 at 6:36 pm March 27, 2025.

Mono: Fully functional

Stereo: Fully functional

RDS: Basic version (Single Pass) works in python, block version fully functional until CDR

RDS: C++ version works until CDR as well with threading, outputting RRC properly with Manchester decoding and differential decoding reading appropriately.

Threading: Fully functional for all paths.

Anuja:

- Worked on Mono implementation (Resampler, Signal Flow) (All modes)
- Worked on Stereo Implementation (Threading, Mode 0 and Mode 1)
- Worked on Unit Testing for the Resampler and Convolution
- Worked on Stereo modelling
- Worked on RDS implementation until Clock and Data Recovery

Hunter:

- Worked on mono modelling with group
- Unit tested convolution functions
- Implemented PLL and RRC blocks in C++
- Modelled RDSBlock before CDR
- Worked on Stereo pre multi-threading (Mode 2 and 3) and RDS implementation in C++

Ryan:

- Worked on mono modelling and mono implementation with group
- Worked on most of stereo modelling and RDS modelling,
- Ceating fully functional stereobasic.py, stereoblock.py, and rdsbasic.py.
- Helped create C++ RDSsupporting functions such as Manchester, paritycheck, getsamples
- Created frame synchronization functions in fmParityMatrix.py, as well as CDR, Manchester, and differential decoding functions in python

Jack:

- Worked on mono mode 0 modelling and mono implementation mode 1
- Worked on stereo implementation mostly with modes 2, and 3
- Worked on RDS implementation supporting Hunter up till RRC filtering
- Worked on post RRC filtering, getting normalization to work, logging data and creating plots, and had functional data up to RRC outputs.
- Implemented getting samples, Manchester, and differential with help of Ryan fixing functions
- Added pointers for ThreadSafeQueueing and created dual queue system for RDS and Audio threads to receive data