Teensy Propagation Simulator Project

Rick Muething, KN6KB Feb28, 2020

- This PP presentation describes and initially documents a short project for the development of a HF and VHF propagation simulator based on the Teensy 4.0 processor with audio shield.

Objectives:

1) Evaluate the latest Teensy CPU and Audio components and libraries.

2))Learn C++ and C#

3) Provide an easy to duplicate, easy to use, low cost channel simulator to allow development, optimization, and characterization of HF and VHF modems and protocols.

Background

- Audio simulators have been used extensively in the development, comparison, and enhancement of HF and VHF protocols
- Most are based on modeling the propagation effects (noise, multipath, fade, etc.) using the Waterson model or some variation of it.
- About 1998 Winlink purchased a fairly simple and low cost (~$700) simulator built around an Analog Devices DSP demo board. This project was designed and built by Johan Forrer, KC7WW an engineer at Harris. The product had some limitations (mostly due to the Analog Devices board capability) but was quite useful and modeled basic WGN (white gaussian noise) and Multi path channels ("standardized" HF channels for MPG, MPM and MPP (multipath good, moderate and poor). The simulator was used in the development of SCAMP, WINMOR, Early ARDOP and comparison of various Pactor protocols.
- The simulator finally died around 2015 and the AD board is no longer available.
- There are a few commercial simulators (mostly used/built by Harris, Motorola etc. for evaluation of HF and VHF protocols but theses are typically $8-30K instruments .

Low Cost Channel Simulator Objectives

- Provides simulation for all standard HF & VHF channel models (WGN, FLAT FADING , MPG, MPM, MPP, MPDisturbed)
- 1 or 4 path (ray) modeling options Audio Bandwidth 300-3300 Hz.
- Separate I and Q channel modeling.(Waterson Propagation Model)
- WGN S:N -40 to + 40 dB, 1 dB steps (allows evaluating new protocols for very low power/weak signal (JT modes etc.)
- Tuning offset and slow drift modeling capability to evaluate modems using SSB
- Ability to model a half duplex connection using a single simulator. (Two simulators are required for full duplex or non symmetric path half duplex modeling.)
- Low cost. Easy to build. Easy to learn. Easy to operate.

Teensy Simulator Function and Flexibility Summary

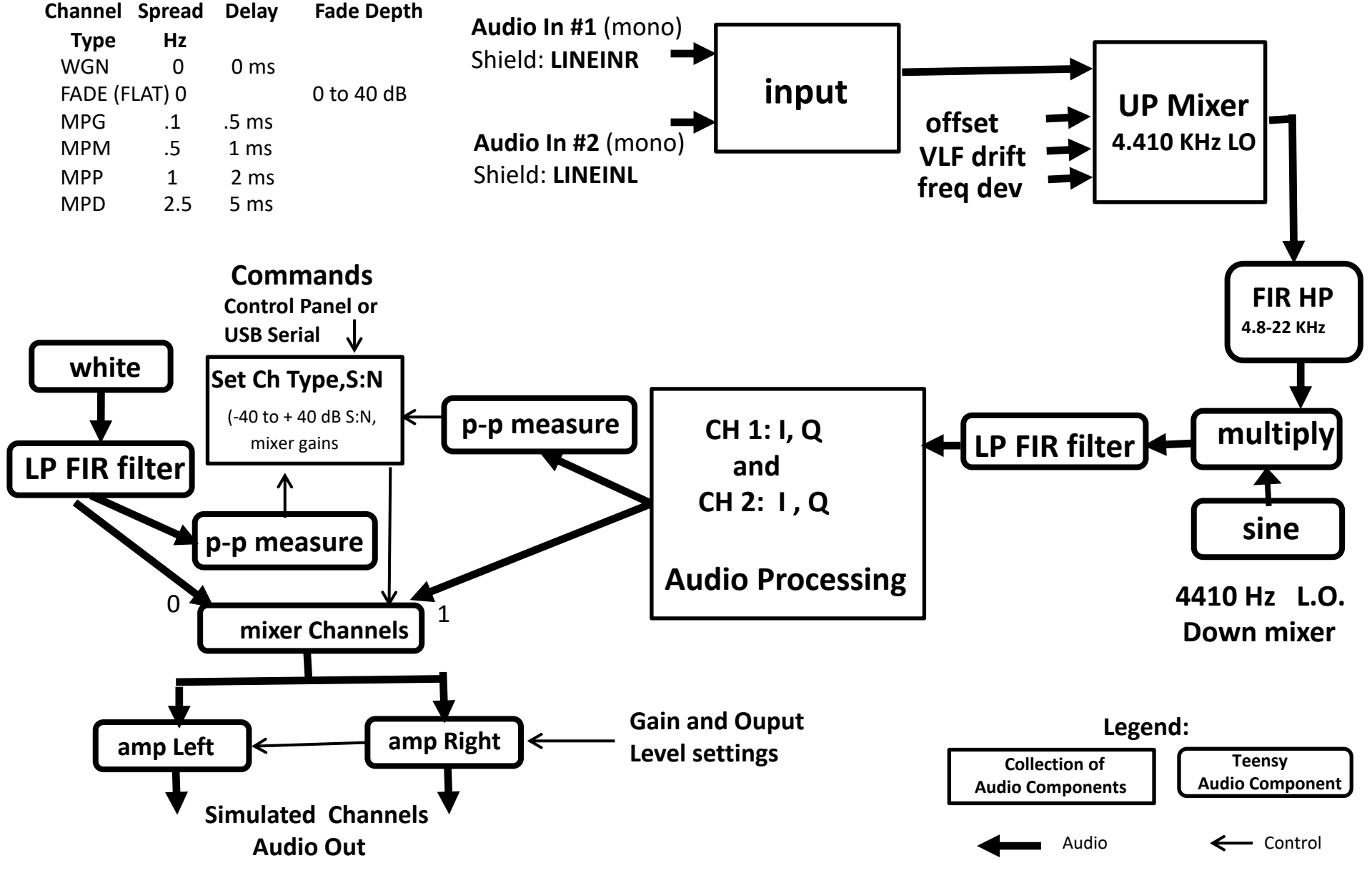
- Common “standardized” modes: WGN(1 path) or MPG,MPM,MPP,MPD(2 paths I & Q, 4 total)
- WGN added (S:N -40 to +40 dB) to all common modes.
- Fading (0-40 dB, 0-100 Hz) can be applied to all common modes (useful for VHF/UHF modeling)
- Static Frequency offset (-200 to +200Hz) can be applied to all common modes. (auto tune modeling)
- VLF FM (0 to +/100Hz deviation, 0-100Hz rate) can be applied to all common modes (drift and doppler modeling)
- Individual gain adjust on all audio inputs (2) and outputs(2)

Half Duplex 4 Path (ray) Channel Simulator

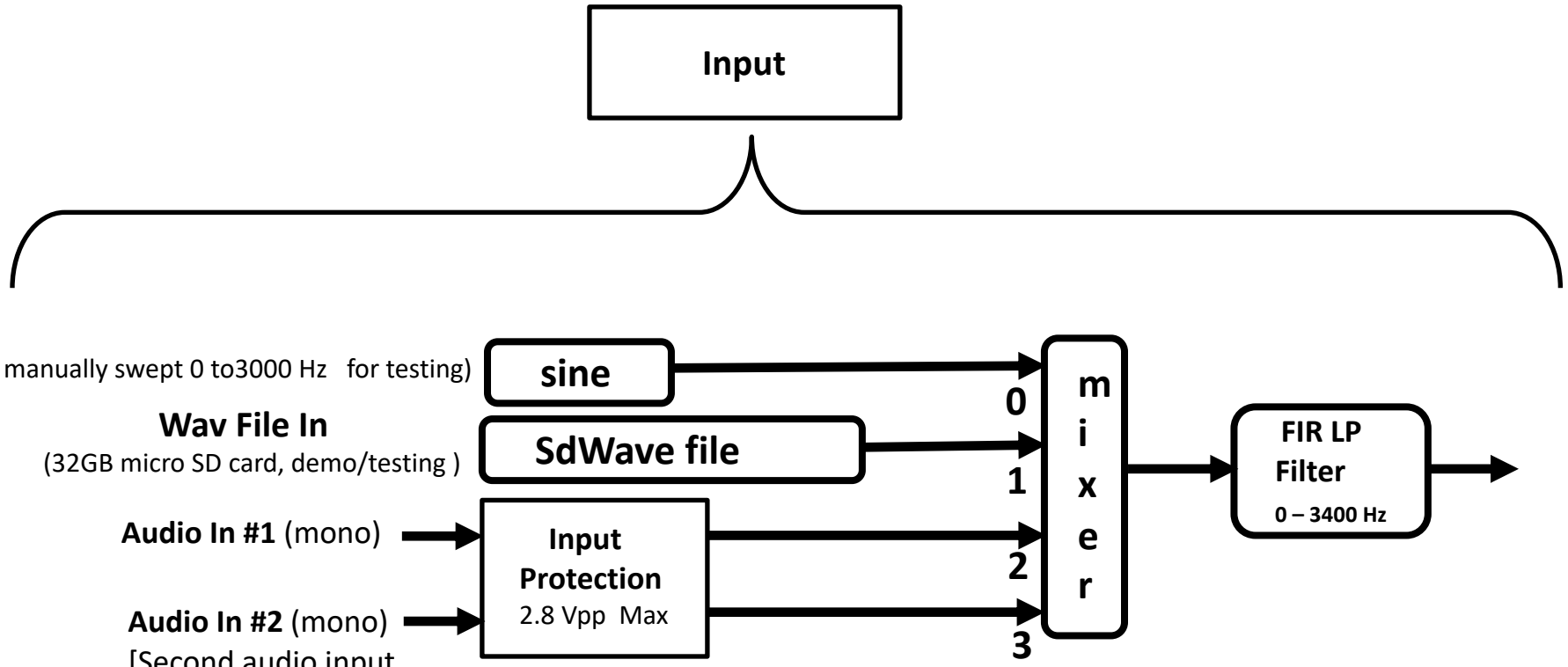
(Based on Arduino Teensy 4.0 + Audio Shield Rev D and Teensy audio library)

Rev5 , 2/27/2020
Rick Muething, KN6KB

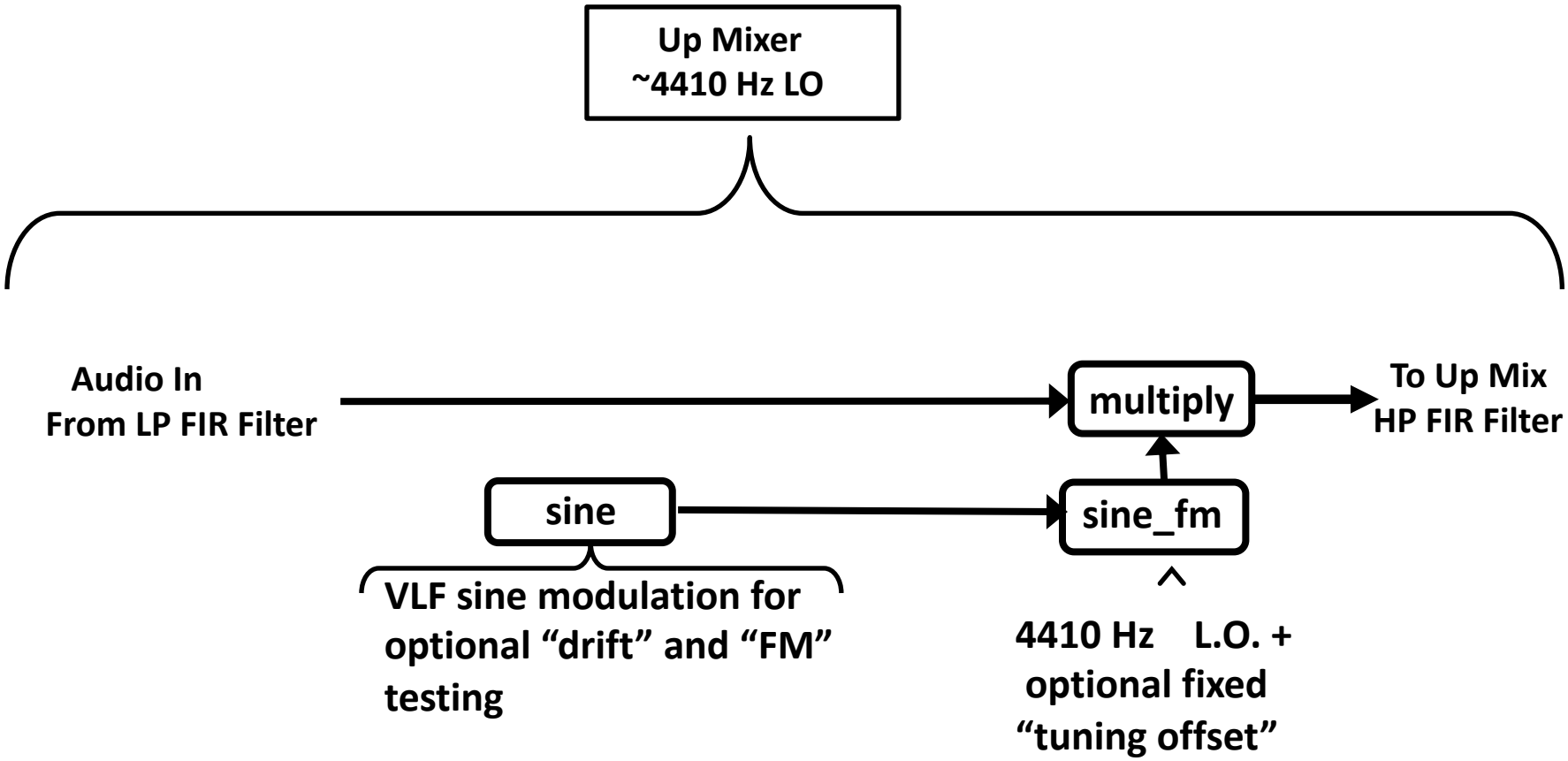
Channel Type	Spread Hz	Delay	Fade Depth
WGN	0	0 ms	
FADE (FLAT)	0		0 to 40 dB
MPG	.1	.5 ms	
MPM	.5	1 ms	
MPP	1	2 ms	
MPD	2.5	5 ms	



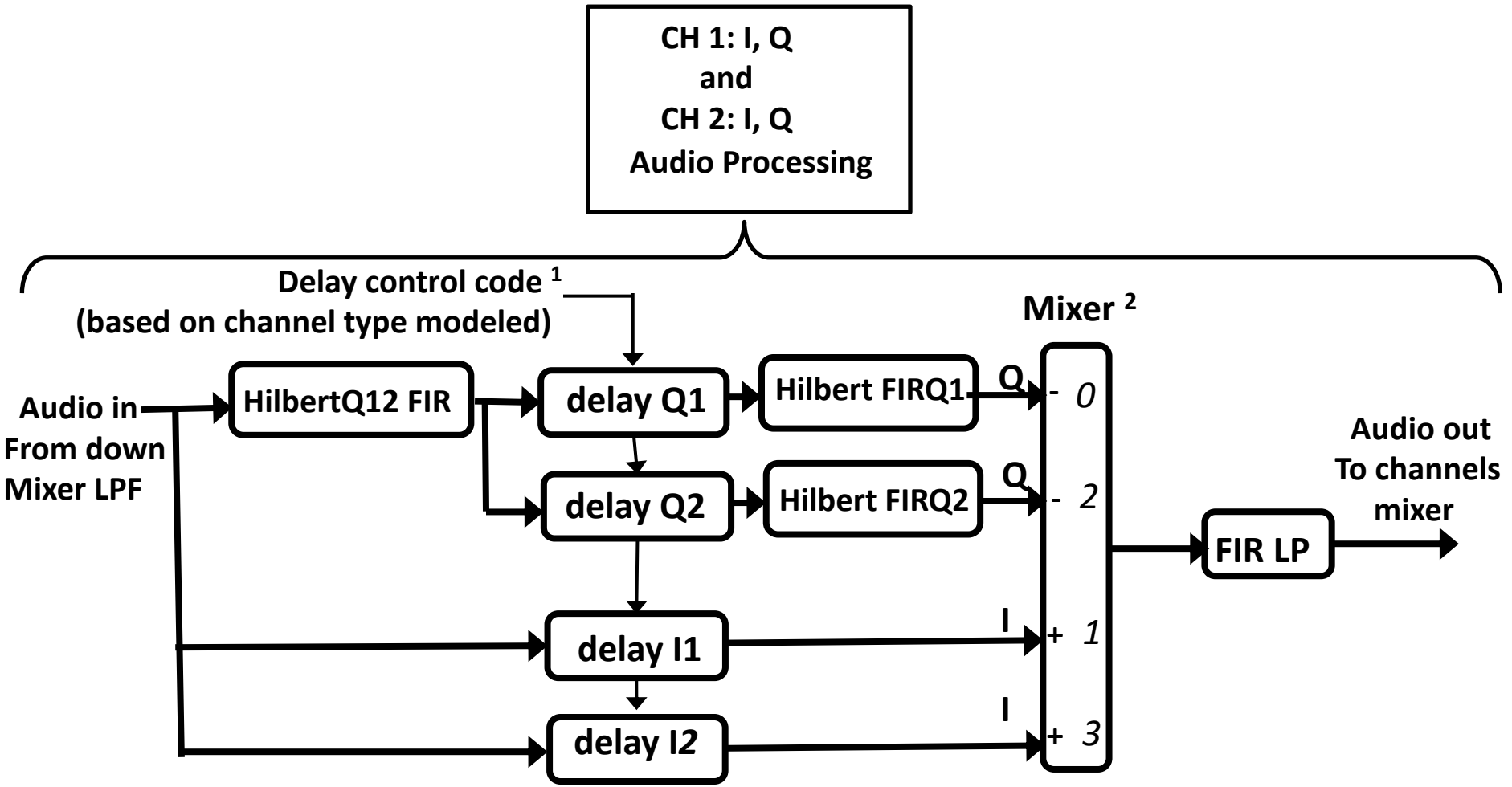
Input Detail



Up Mixer Detail

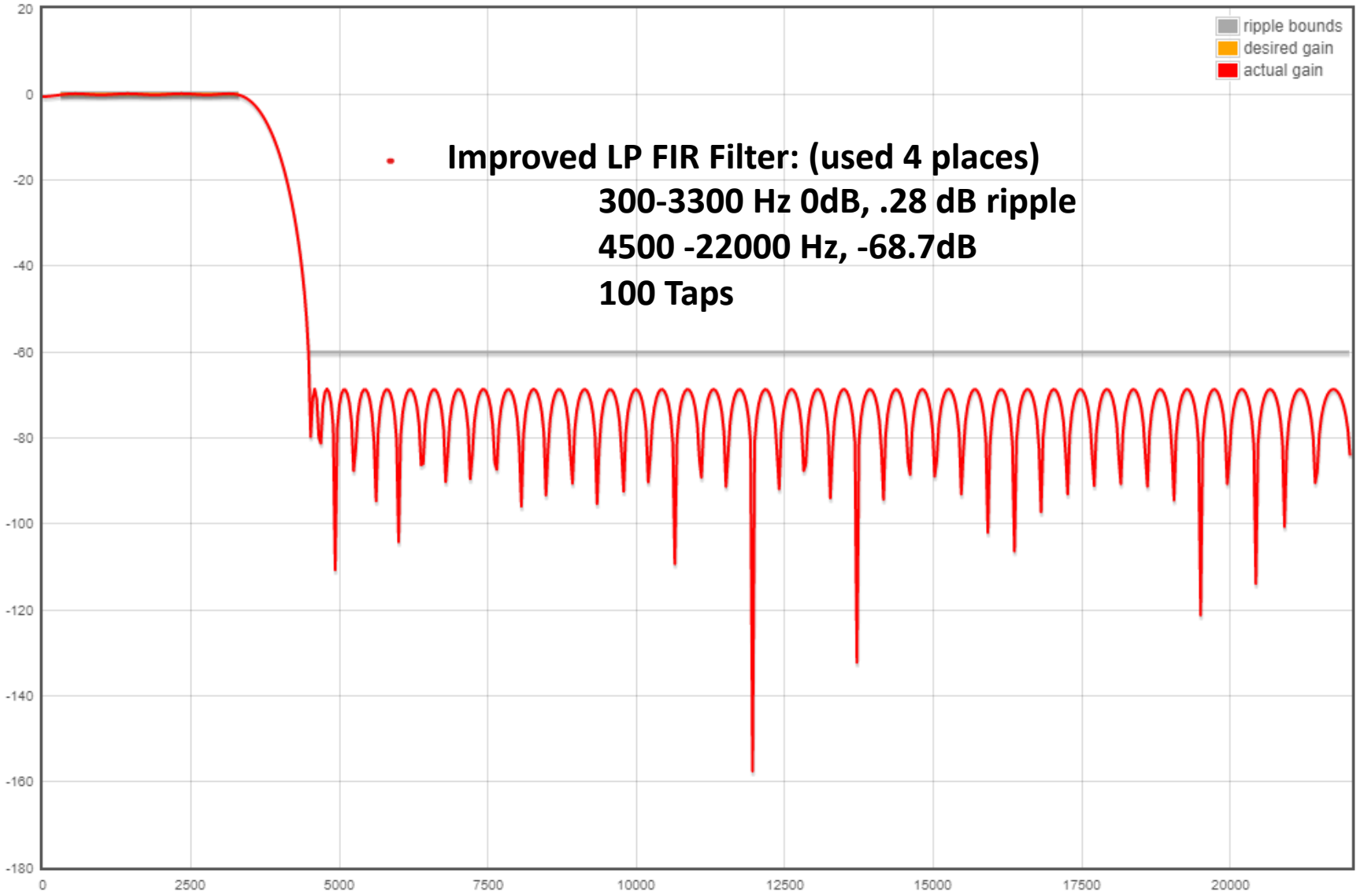


Channels Detail



1 Equalizes I delay path for Hilbert filters delay (~2.72 ms total)
Determines magnitude and rate of changes to I and Q delays
based on channel being modeled.
Modulates delays with random values based on channel being modeled

2 Negative value accommodates 180 deg shift of 2 Hilbert filters: [H(H(x)) = -H(x)]



add passband add stopband !predefined ▼

from	to	gain	ripple/att.	act. rpl
300 Hz	3300 Hz	1	1 dB	0.28 dB
4500 Hz	22000 Hz	0	-60 dB	-68.67 dB

sampling freq.	44100 Hz
desired #taps	100
actual #taps	100

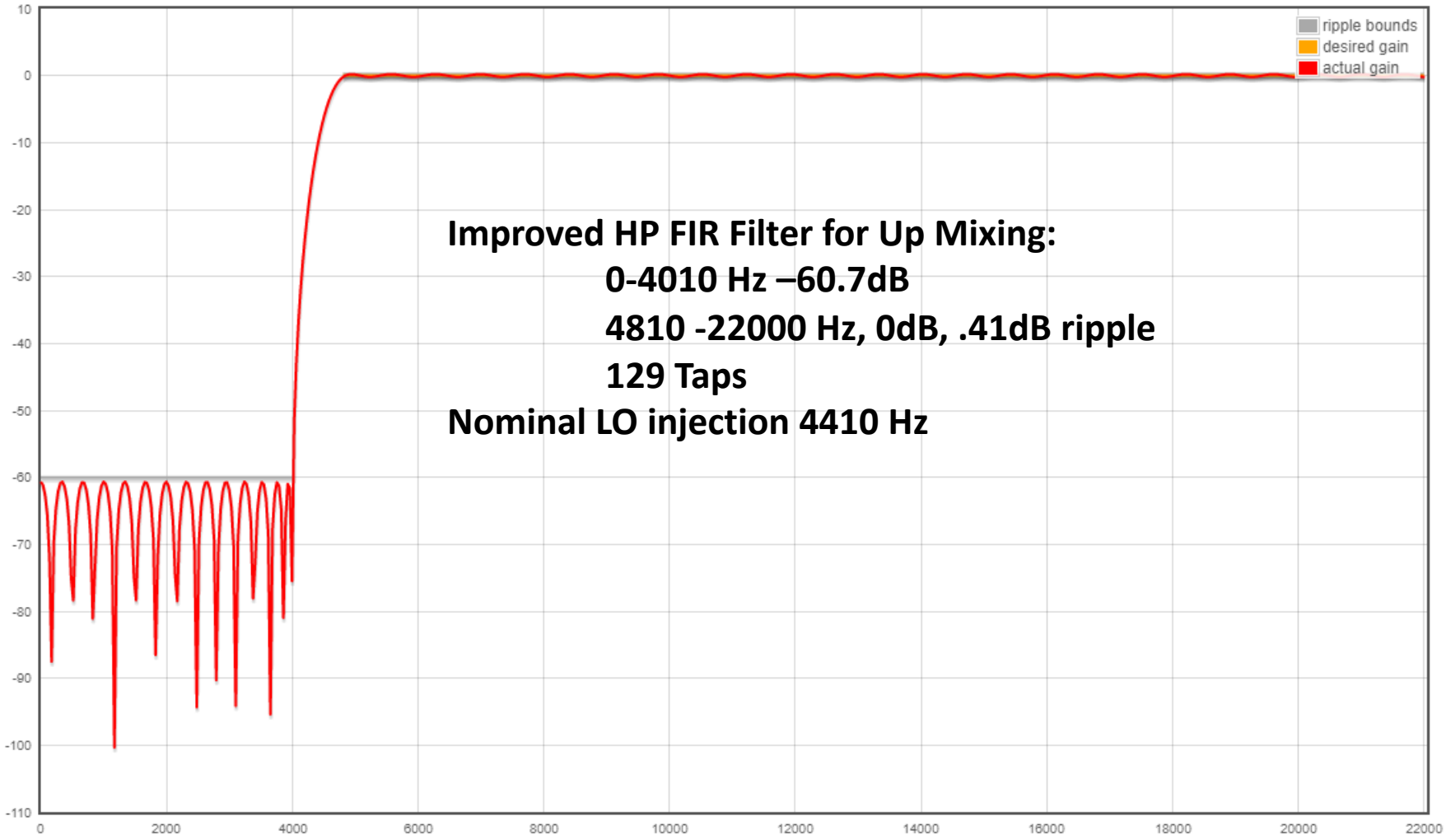
DESIGN FILTER

I am working on **TFilter2**. Screenshot here. Features include CIC (Sinc) filt
effect of quantization, save/load/share, aliasing visualization, and signal chain

If you want to **advertise here**, contact me at peterisza@gmail.com.

TFilter is being used by many tech companies and universities.

Gain vs. Frequency | Impulse Response | Source Code | Feature Request | Enterprise | IIR Design



Improved HP FIR Filter for Up Mixing:

0-4010 Hz –60.7dB

4810 -22000 Hz, 0dB, .41dB ripple

129 Taps

Nominal LO injection 4410 Hz

add passband | add stopband | !predefined

from	to	gain	ripple/att.	act. rpl
0 Hz	4010 Hz	0	-60 dB	-60.68 dB
4810 Hz	22000 Hz	1	0.6 dB	0.41 dB

sampling freq.	44100 Hz
desired #taps	129
actual #taps	129

DESIGN FILTER

I am working on **TFilter2**. Screenshot here. Features include CIC (Sinc) filters, effect of quantization, save/load/share, aliasing visualization, and signal chain.

If you want to **advertise here**, contact me at peterisza@gmail.com.

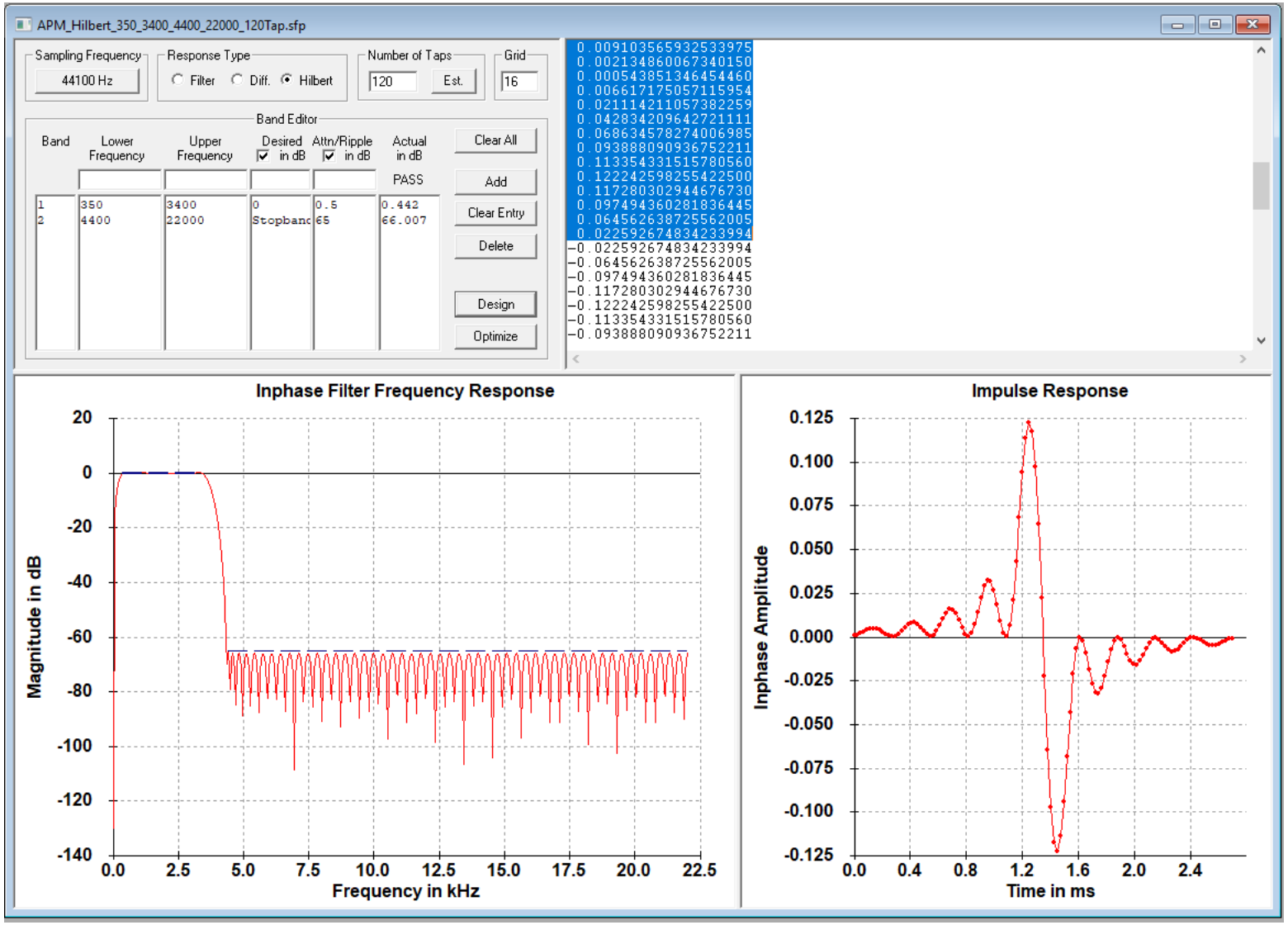
TFilter is being used by many tech companies and universities.

Hilbert FIR Filter: (used 3 places, I and Q processing)

350-3400 Hz .44 dB ripple

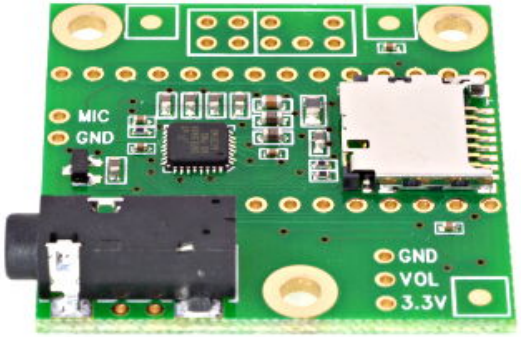
4400-22000 Hz –66 dB

120 taps, 1.3605 mS delay, (Scope FIR: Advanced PM)



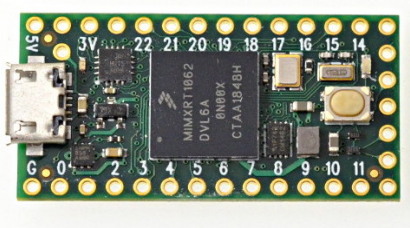
Basic Hardware for first breadboard
(This is all that is required!)

Additional support: LCD display, encoders



Audio Shield Rev D

Sound card chip (16 bit stereo, 44100 Hz sample rate)
Headphone Jack
Micro SD card reader (32GB)
Microphone input
"Piggybacks" to processor
~ $14 assembled



Teensy processor 4.0

ARM Cortex-M7 CPU @ 600Mhz
(~ 3-5 x faster than Teensy3.6)
1024K Ram 2048K Flash
2 USB ports 480 Mbit/sec
31 PWM pins
40 Digital pins
14 analog pins 12 bit A:D
Random Num Generator
Real Time Clock
~$20 assembled

https://www.pjrc.com/store/teensy40.html

User Interface

- Goals: Easy to Learn (1-2 sheet instructions)
  - Self contained and OS independent.
  - Expandable to allow external CPU control/readout via USB serial for more complex or automated testing
- Initial approach
  - Simple user interface (small LCD/matrix screen and rotary encoders for “manual” setup and control.
  - Use existing Teensy USB interface for +5v power, USB serial control and readout
  - Provide for easy common audio connections (stereo plugs and patchcords) for half duplex modeling.

Manual User Interface

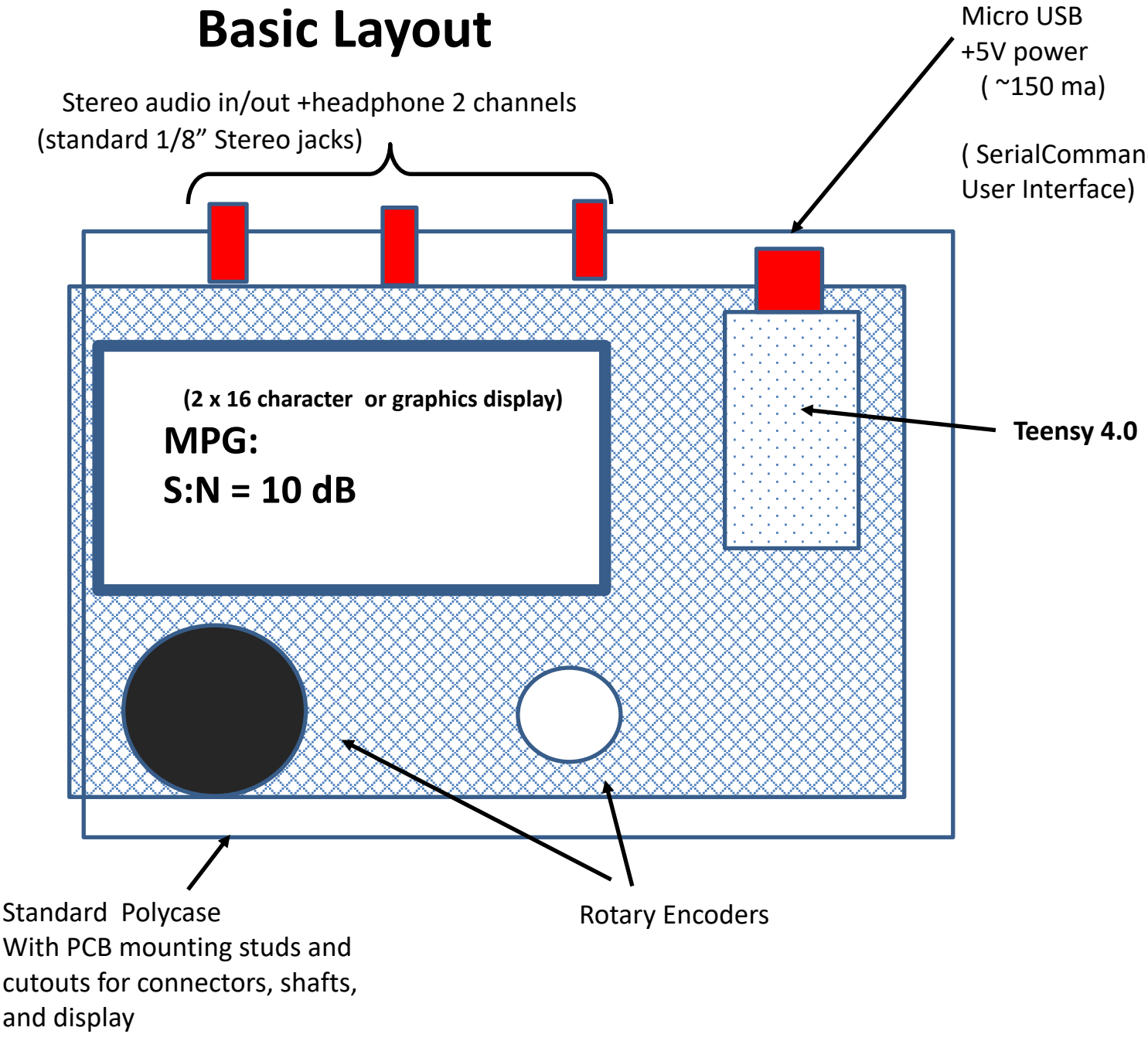
(based on Teensy controlled 24 steps/turn low-cost Rotary encoders)

Encoder 1, Large knob

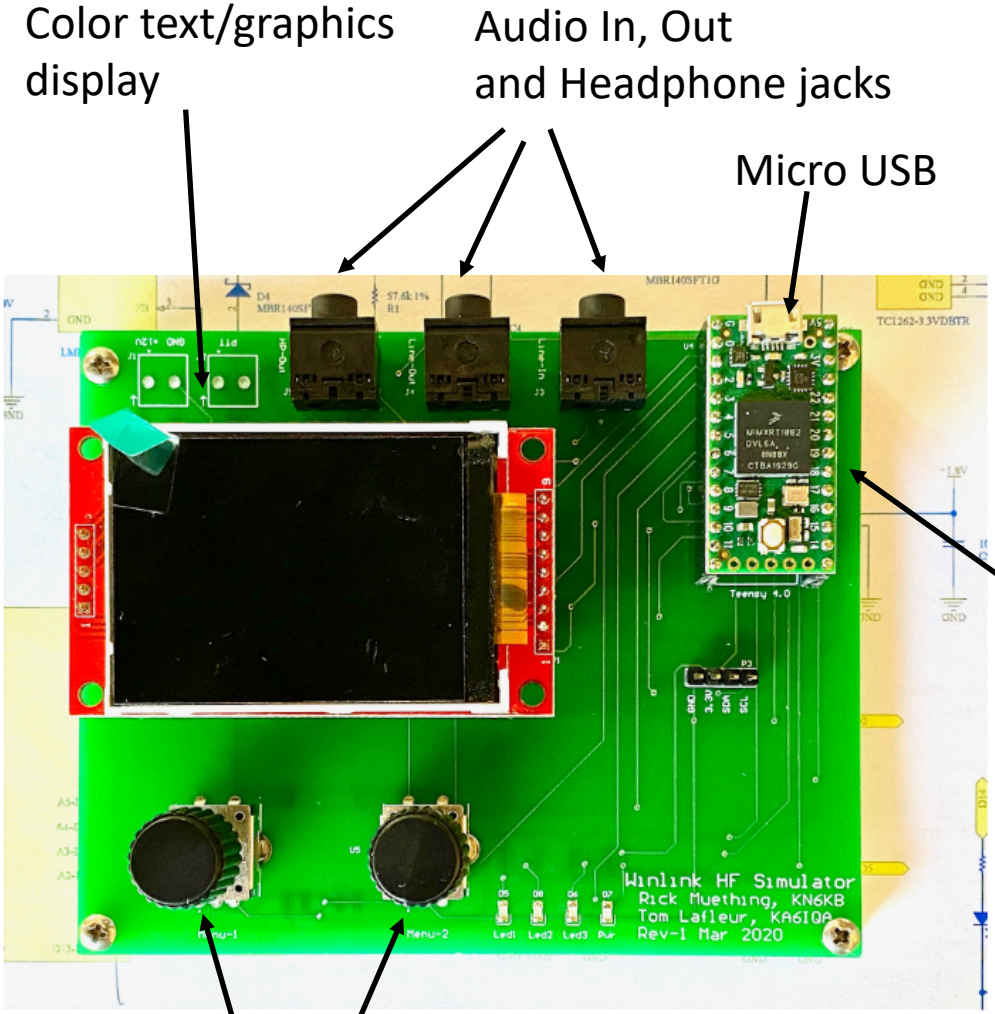
Encoder 2, Small knob

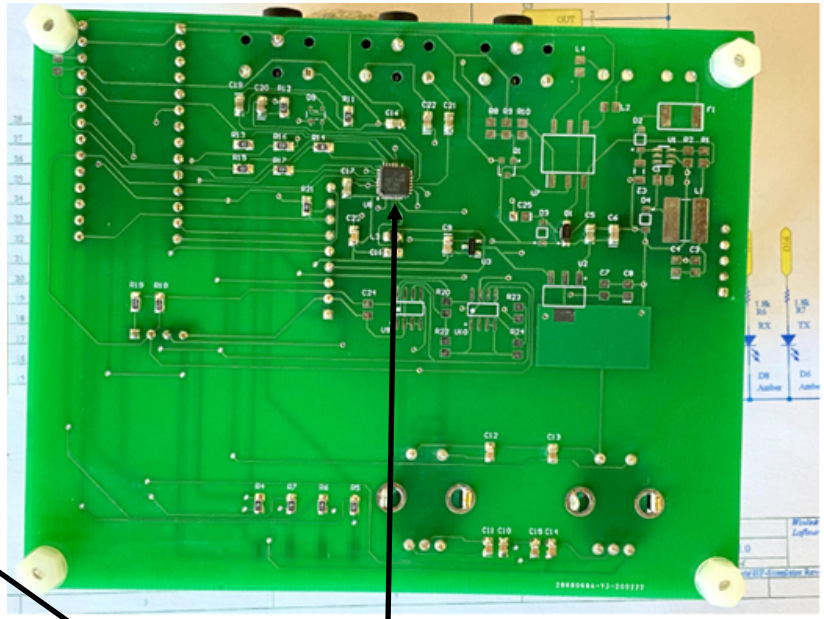
MODE		Parameter / Range /Power On Default	
WGN	Common	S:N /-40 to +40 ,1 dB steps/ +10	
MPG		S:N /-40 to +40 ,1 dB steps/ +10	
MPM		S:N /-40 to +40 ,1 dB steps/ +10	
MPP		S:N /-40 to +40 ,1 dB steps/ +10	
MPD		S:N /-40 to +40 ,1 dB steps/ +10	
FADE Depth		Depth of Fade dB/ 0 to 40 dB/ 0	
FADE Freq		Frequency of Fade/ (0, .1, .3, 1, 3, 10, 30,100)/0	
OFFSET	Less used	Freq Offset/-200 to +200 Hz, 10 Hz steps/0	
FM Deviation (p-p Hz)		FM Deviation/0-100 Hz (0,1,3,10,30,100)/0	
FM Rate (Hz)		FM Rate/0-100Hz (0, .1, .3, 1, 3, 10, 30,100)/0	
CH1in (Gain)		Ch1 Input Gain/0-20/10	Flash storage?
CH2in (Gain)		Ch2 Input Gain/0-20/10	
CH1out (Level)		Ch1 Output Level/0-20/10	
CH2out (Level)		Ch2 Output Level/0-20/10	

Basic Layout



Photos of Tom's PCB and first prototype!





USB Serial User Interface

(preliminary)

USB Serial Commands and Responses for automated testing

(Mimic manual commands from Mode and Parameter Encoders)

Command (case insensitive, End with Cr)	Acknowledge Response/ Interpretation
WGN:20	OK(Cr) or ?(Cr) / WGN +20dB S:N
MPG:20	OK(Cr) or ?(Cr) / MPG +20dB S:N
MPM:-10	OK(Cr) or ?(Cr) / MPM -10dB S:N
MPP:+20	OK(Cr) or ?(Cr) / MPP +20dB S:N
MPD:20	OK(Cr) or ?(Cr) / MPD +20dB S:N
FADE Depth:10	OK(Cr) or ?(Cr) / FADE Depth:10 dB
FADE Freq:10	OK(Cr) or ?(Cr) / FADE Freq: 10 Hz
OFFSET:100	OK(Cr) or ?(Cr) / OFFSET: 100 Hz
FM Deviation:10	OK(Cr) or ?(Cr) / FM Deviation 10 Hz
FM Rate:.1	OK(Cr) or ?(Cr) / FM Rate .1Hz
CH1in:10	OK(Cr) or ?(Cr) / CH1 gain = 10
CH2in:10	OK(Cr) or ?(Cr) / CH2 gain = 10
CH1out:20	OK(Cr) or ?(Cr) / CH1 level = 20
CH2out:20	OK(Cr) or ?(Cr) / CH2 level = 20

Note: 1. All commands case insensitive. “+” signs ignored.

2. All commands and replies terminate with (Cr).
3. “OK” reply indicates command was interpreted OK and executed.
4. “?” reply indicates command was not understood, corrupted or out of range.
5. Commands must be sequenced by sender using Acknowledge Response.

Status 2/28/2020

- All functions of simulator coded and basically tested using patch cord hand wired breadboard.
- Parts received (Teensy, electrical, hardware, displays, PCB, enclosure) for 2-3 prototypes.
- Target for prototype completion (Mar 15)
- Remaining work:
  - mechanical drawings (drill/route guides)
  - silk screens (Case and PCB)
  - user instructions
  - Finishing USB serial command and data interface software
  - testing with several HF/VHF modems.