

# W0EB/W2CTX DSP Audio Filter

## Operating Manual V1.11

Manual and photographs Copyright© W0EB/W2CTX, January 09, 2019.  
This document may be freely copied and distributed so long as no changes are made and the authors are given full credit for their efforts.

---



W0EB's build of the filter with the 1.8" Color touch Screen showing the adjustable CW filter. Green dot - CW Zero Beat indicator when incoming tone matches the filter's Center Frequency.

This project got started late last year when TSW's master programmer, Ron - W2CTX came across an article on the internet by Gareth, G11MIC. Gareth used a small Teensy 3.2 development MPU from PJRC.com, noting the Teensy 3.2 has a reasonably decent on-board ADC (analog to digital converter)/DAC (digital to analog converter) and is fast enough, with enough memory to be able to do DSP audio processing and so had the capability of creating tunable audio filters with varying low and high skirt frequencies which gives variable bandwidth capability and also once the

bandwidth is established, the filter's center frequency can be easily moved to align it to the incoming signal. Gareth's initial focus was on creating a built-in, tunable CW audio filter for his Yaesu FT-817.

Originally, we used much of GI1MIC's published, open-source Teensy code and his original wiring to test the concept. Ron, W2CTX, in working with the original code and achieving some success, began refining the code to fit our own concept of what types of filtering we would most like to have. We eventually came up with 3 filter types, CW, SSB and RTTY (since changed to "DIG" for Digital) as it is versatile enough to handle most of the current digital modes being used. Ron found that he could remove almost all routines dealing with control by the FT-817 that were not really pertinent to the stand-alone concept we were working on.

We first used a 1.44" TFT Color Display that did not have touch capability. We subsequently found a much better, slightly larger 1.8" TFT Color Resistive Touch Screen and attempted to use that in place of the original "non-touch" display. There were a few growing pains with the 1.8" display and, at first we could not get the touch feature to work with the Teensy 3.2. Both Ron and I were beating our heads on the table trying to figure out what was happening.

We purchased the displays from [www.buydisplay.com](http://www.buydisplay.com) and it's their ER-TFTM018-2 with the "pin header, 4-wire SPI controller", "3.3V power" and "Resistive Touch Screen installed" Asking for help from the supplier, they said they didn't have a Teensy 3.2 to test it with so essentially we were on our own. It came down to the fact that we needed to use the same "clock" pulse to set the timing for both the display "Chip Select" and Touch "Chip Select" functions.

The Teensy 3.2 apparently has a much faster and narrower pulse output for this function. We had to parallel the two "Clocks" and if I disconnected the clock from the "Touch" controller the display would work, but not the

touch. Reconnecting that clock, nothing worked. It appeared to be related to the timing of the two "Chip Select" pulses relative to the clock pulses. I finally got out my oscilloscope and while checking to see how close together both "CS" pulses were occurring, that the extra capacitance introduced by the scope probes and cables made things work. Taking an approximate guess at what value to use, I added a 47 picofarad capacitor to ground at each of the chip select pins on the display and BINGO! The display and touch now worked!

Once that hurdle was passed, Ron rapidly got the filters actually working and tunable. Over the next couple of weeks things got refined and a "Zero Beat" indicator was conceived and after a few more hurdles (the input to the Teensy 3.2's counter section had to be conditioned to make detecting the incoming tone reliable on lower levels of audio and limited so it wouldn't damage the Teensy's 3.3V (but 5V tolerant) digital input limits. Ron had the neat idea of making the actual indicator graphic look similar to one of the old "Magic Eye" indicators and so it starts out as a green ring and closes down to a solid green circle when the incoming tone matches the selected center frequency of the CW filter.

Finally, it was up to me (W0EB) as project coordinator and test engineer to create an accurate schematic from the working breadboard prototype. Once I had that finished, N5IB, the TSW PC layout engineer got the project and after a few emails back and forth, a viable PCB layout was agreed upon and boards ordered. Once the initial order of boards arrived I built up several to ensure that they worked correctly - I found a minor specification difference between the 74HC14 through hole part I used on the breadboard and the SMD version used on the PCB and wound up having to delete one resistor from the original design to get the 2nd stage of the zero beat audio conditioner to work reliably but other than that everything functioned as planned.

The final stage was to actually build one into an enclosure, complete with all input/output jacks, digital encoder for adjustments, an audio output amplifier (The Teensy's output DAC doesn't have sufficient current output to drive headphones or speakers directly) and the 1.8" touch screen display.

Compiling The Firmware with the Arduino IDE:

We originally used the Arduino IDE Version 1.8.5 in development of this firmware but they have progressed to 1.8.8 as of this writing. We had to modify one of the libraries to make the 1.8 inch display specified for the project operate correctly. That modified library must be downloaded from the TSW website [www.w0eb.com](http://www.w0eb.com) and is available in the MANDATORY\_LIBRARIES directory or via a link near the bottom of the page explaining the history & operation of the filter. The name of the library is TFT\_ILI9163C and the unmodified version may already be in your version of the IDE's hardware/teensy/avr/libraries directory. You will need to download the modified library from the TSW website, unzip it into a separate directory on your desktop and copy it into the hardware/teensy/avr/libraries directory of your IDE. When it asks you if you want to overwrite the existing files answer YES or the modified portion won't be properly installed.

In the IDE's "Tools" menu you need to set the "Board" to Teensy 2.0/3.2 and also set "Serial" to "Serial+MIDI+Audio" before you compile and upload the sketch to the Teensy 3.2.

Operation:

First time use of the Touch Screen:

You will have to calibrate the 1.8" TFT Color Touch Screen the first time you use it as the X-Y coordinates for the screen are stored in the Teensy's memory and "Touch" won't work until an initial calibration is performed.

To enter "Screen Calibration" mode, hold in the encoder's pushbutton and apply power to the unit. Once it passes the "Splash Screen" it will enter "Screen Cal" mode and display the following screen.



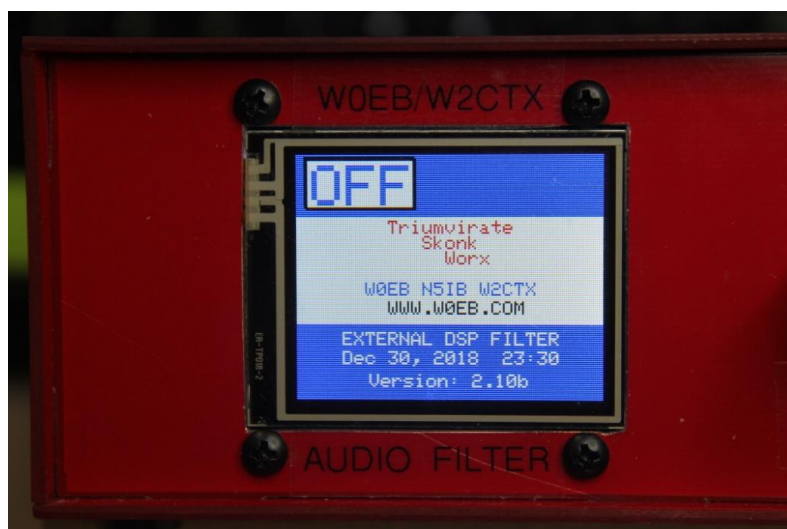
Using a blunt wood or plastic stylus, press and hold the dot in the upper left corner of the screen until the program averages and saves the X-Y coordinates for the upper left corner of the screen and the dot jumps to the lower right corner as in the following image.



Once you have done step 2 properly, the following screen will appear,



Tap the stylus on the screen and the Touch “X-Y” calibration will be complete. Now the “Splash Screen” display shown below will appear.



While this screen is displayed, the filter’s effective bandwidth is 60-3000 Hz

The software version number (in this case 2.10b) may differ depending on the version of software actually loaded. At the time of this writing, we were still in “Beta” testing and the small b at the end indicates a Beta version. Release versions will end with a capital “R”.

**NOTE: If you are re-programming the unit to a later version of the software you will have to apply power through the main power connector and turn the unit on to activate the USB connector on the back. As indicated in the**

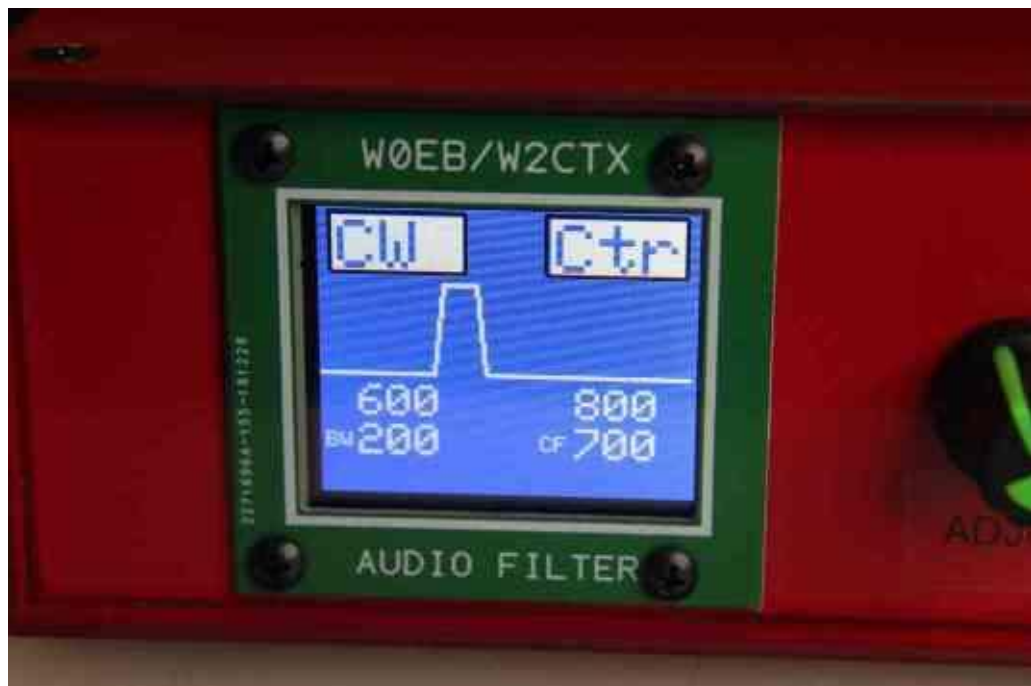
construction manual, the “VIN/USB” power jumper MUST be cut which negates powering the Teensy through the USB connector.

OFF: (Splash Screen)

This is the condition in which the DSP filter starts after turned on and the boot sequence completes. While this screen is active, the filter is basically bypassed but in this condition, it does have a bandwidth of 60-3000Hz.

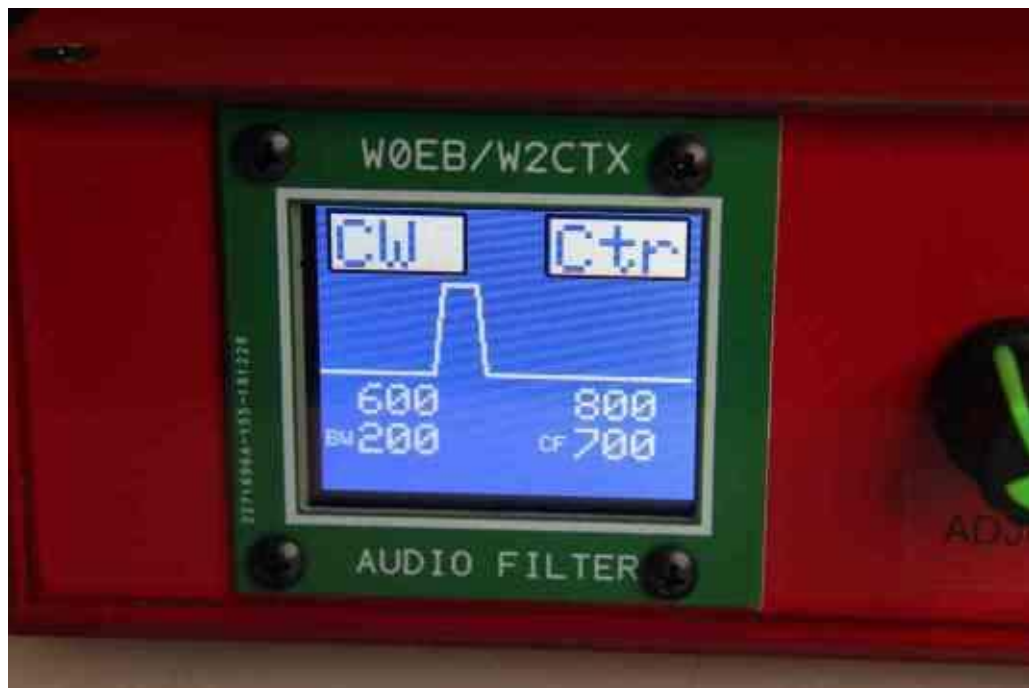
CW filter:

There are two ways to select the three available (CW, SSB, DIG) filters. While the “Splash” screen (previous picture) is showing, give a short press on the encoder’s push button (or press the “OFF” box on the touch screen briefly with a small plastic or wooden stylus (blunt point please) and the CW filter’s display will appear.



Note: If you give the encoder button a fairly long press you will start a list of options which are: “Next Filter”, “Reset Filter”, “Save Filter” and “No

Change”. You need to keep holding the encoder button to cycle through these options and release it when you see the one you want on the bottom of the screen. Once you have your filter set up the way you want it (BW and CTR) you need to use the “Save Filter option to write the current filter data to EEPROM. You can also cycle through these options by using a long press of the stylus on the CW or SSB filter button on the screen.



You can now adjust the center frequency to anything you want within the range of 100Hz to 1690Hz by either turning the encoder knob OR pressing lightly on the screen. If you press on the left side of the screen the center frequency will decrease and if you hold the stylus there it will continue decreasing until it reaches the lower limit. The same goes for increasing the center frequency, press & hold the stylus on the right side of the screen and if you continue to hold it, it will increase until the upper limit is reached.

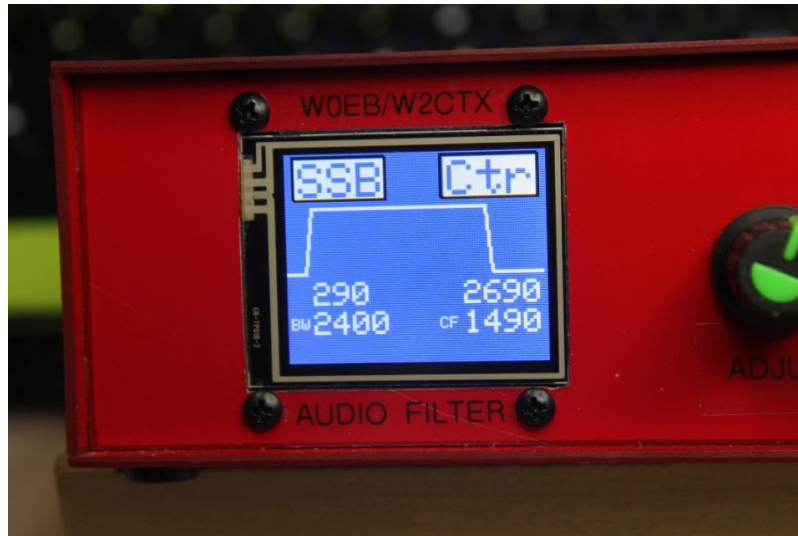


Once you get to the desired Center frequency, release stylus pressure or stop turning the encoder.

Tap the “CTR” block or give a short press on the encoder knob and “Hi” (high side of the skirt’s frequency) will be selected. The same tuning instructions apply. Tap the “Hi” block or give another short press to the encoder button and “Lo” will appear. Here you can tune the lower side of the filter skirt’s frequency and another press or tap will get “BW” where you can set the bandwidth of the filter (minimum BW is 200Hz, max is 1900Hz). Once you have your CW filter set up the way you want it, press and hold the encoder button (or press and hold the “CW” on-screen block with your stylus) until you see “Save Filter” on the bottom line. As soon as “Save Filter” appears, release the button or stylus and the configured filter will be saved to EEPROM. The next time you select the CW filter, all of your settings should be preserved.

#### SSB Filter:

After boot up and you have pressed the “OFF” button to bring up the CW filter menu, pressing the encoder or touching the CW button will bring up the CW filter as previously described. A longer press (until “Next Filter shows on the bottom line) of the encoder or CW box on the screen will switch to the SSB filter. All tuning instructions explained previously for the CW filter apply to the SSB Filter. Play with these adjustments until your received SSB audio (tune the signal in on the radio first before using the filter) sounds the way you want it to. The SSB filter’s parameters can also be saved with a long press of the encoder button to show “Save Filter” and will be recalled the next time you select the SSB filter.



### DIGital Filter:

The last available filter is designed for use with most digital programs; RTTY, PSK, FT8 and others. All tuning adjustments explained previously apply to the DIG(ital) filter and the same goes for saving the adjustments once you have things set up the way you want.



That's about all there is to it. Experiment with the various bandwidth & center frequencies for each filter and save the one you like of each type.

Jim, W0EB and Ron, W2CTX.