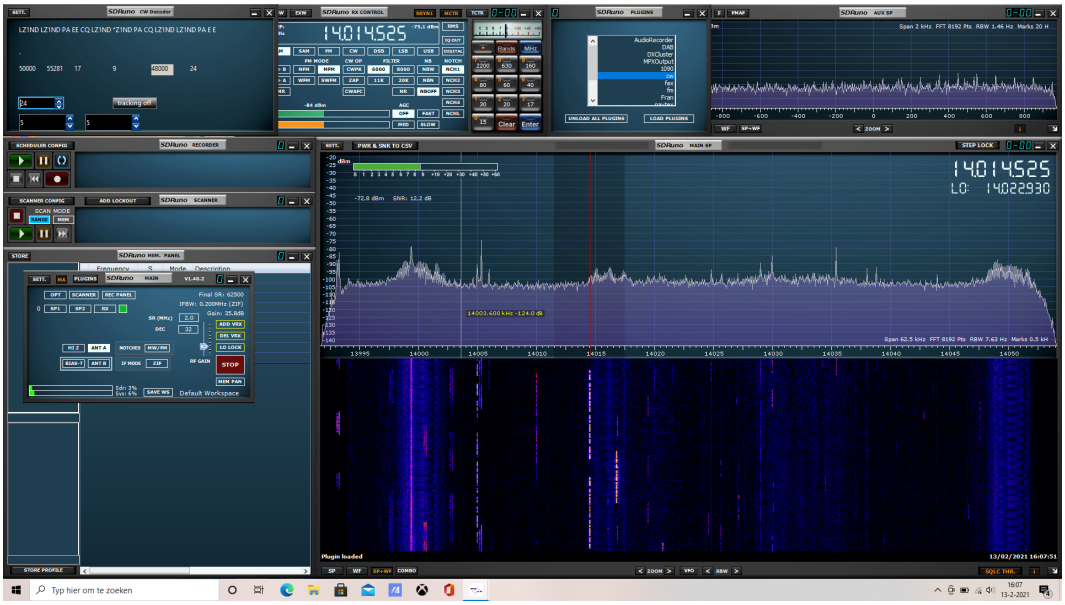
A simple CW plugin for SDRuno

Jan van Katwijk
Lazy Chair Computing
The Netherlands
J.vanKatwijk@gmail.com

February 14, 2021



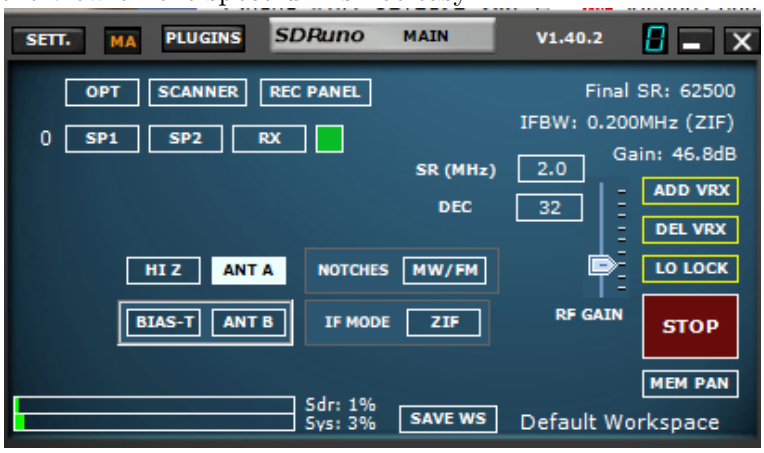
1 Introduction

The SDRuno cw plugin is a simple plugin to decode CW (continuous Waves) signals. CW is in amateur bands still very popular, I am usually looking at the 14 MHz band.

2 Settings

CW (basically carrier on/carrier off) is a signal with a small footprint, the width used on the band is less than 50 Hz. The decoder therefore works with an intermediate samplerate of 2000 samples/second. Since the minimal samplerate for the SDRplay family is 2000000, a lot of decimation (and filtering) has to be done.

This implementation requires an input sample rate of 62500 samples/second, this requires the setting of the mainwidget to a samplerate of 2000000, and a decimation of 32, as shown in the picture. One should realize that the SDRuno spectrum display shows a band of 62.5 KHz, the advantage is that one sees a lot of signals, the disadvantage is that precise tuning, based on the view on the spectrum is not easy.

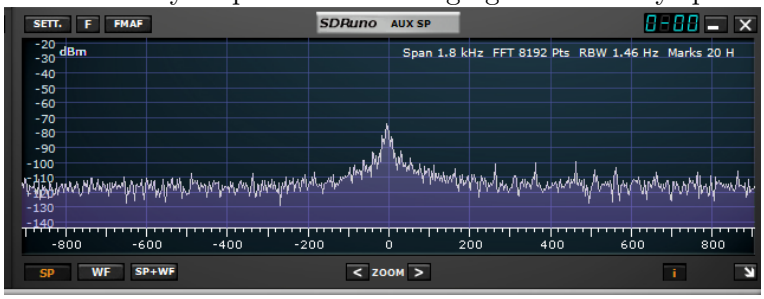


The plugin generates an audiotone of 800 Hz + the tuning offset, At least for me, some sound helps with tuning. The sound is output with a rate of 48000, setting "AM" in the RX control window will set this rate.

3 Tuning

As said, CW is a signal with a small footprint, and since many of the amateur transmissions are brief messages (such as CQ CQ ...), tuning requires some training.

What is really helpful here is enlarging the auxiliary spectrum display, and zooming in.



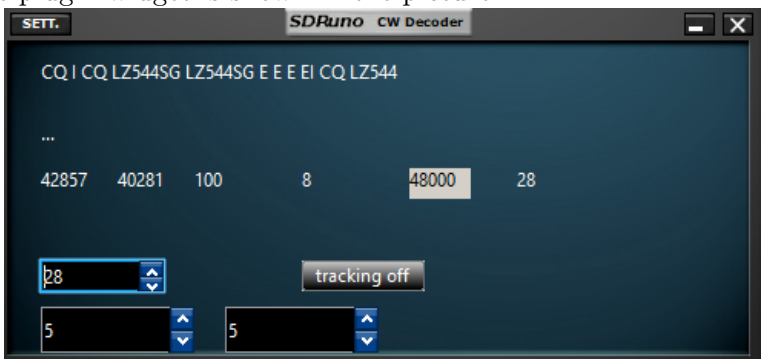
The way to tune is in two steps

- coarse tuning with the mouse on the main spectrum display;
- fine tuning with the mouse controlled numerical display until the CW signal is above the '0' in the auxiliary spectrum display.

The drawback of course is that CW transmissions are only a few seconds long, so by the time tuning is - more or less - correct, the transmission stopped.

4 The plugin

The plugin widget is shown in the picture



The main issue with CW decoding is that different operators use different transmission speeds. The way the plugin works is that the *duration* of the signal above a certain noise level is measured. As known, a dot and a space within the encoding of a letter have the same length, while a dash is supposed to take 3 times as much time. Experience shows that most operators transmit with app 25 to 30 words per minute. The spinbox - in the picture set to "24" - can be used to adapt the wpm estimate.

The top line in the widget shows the received text, the second line - in the picture only containing three dots - displays the dots and dashes of the letter received. The third line contains 6 number displays, from left to right

- the number of micro seconds *assumed* as duration for a dot. The number is computed by looking at the setting for the words per minute;
- the number of micro seconds *measured* as (average) duration of the spaces between dots and dashes;
- the strength of the signal;
- the strength of the noise floor;
- the audio output rate;
- the current words per minute setting.

As said, the next row contains a selector for the assumed words per minute. It furthermore contains a button labeled *tracking off*. If a transmission is seen with a duration longer than a few seconds, a tracker can be activated that adapts the assumed WPM. However, for transmissions of 3, 4 seconds as usually seen on the 14 MHz band, it is not advisable to set the tracker on.

Finally, the bottom row contains two spinboxes:

- the filter depth. Before attempting to decode the on/off appearance of the carrier, a lowpass filter is applied to the signal. The actual degree of the (FIR) filter is twice the amount on the selector plus 1.
- the squelch level. To distinguish signal from noise, a squelch level can be set.