

# Additions and suggestions to SunSDR2 DX

*This article is a stub. You can help by expanding it providing more/new ideas and suggestions. Don't hesitate to borrow a good idea from an older proven radio.*

## CW - method of tuning

Still missing the “opposite sideband” on CW mode. The ESDR Software knows DIGL and DIGU. Why it does NOT know CWL and CWU? The current CW mode is marked on other radios mostly as CWR (CW reversed) – when tuning UP the beat note goes LOW. More natural is the opposite tuning, ie. when tuning UP the beat note goes UP. This is marked as CW (= NORMAL CW) on most radios.

It is related to DX operation on low band edges where the ZERO BEAT denotes the actual frequency, ie. dial at 3500.000 kHz and ZERO BEAT note means that you operating exactly on 3500.000 kHz. Remember, please that older radios does not have the exact band limits built-in.

I would expect in next version actual style of tuning marked as CWR and newly introduced option CW in opposite mean, ie. when tuning UP the beat note goes UP. This should be the standard.

## Sidetone vs. monitor

Another essential thing for the CW operator is the method of listening to own sent signal. There are two options:

**1. SIDETONE** – the resulting beat note does not need to be exactly the same as the offset between the exact signal frequency and the frequency to tune the receiver to obtain an audible beat note, however it is a BIG ADVANTAGE when the offset is identical to the resulting beat note. MORE IMPORTANT is that there should be ZERO LATENCY between paddle touch and sidetone audio output. The main purpose of this is NOT monitoring the actual signal quality, it is to provide the necessary feedback from hand to ear (brain) while composing the character. To send out perfect, error-free Morse code, a high level of synchronization is needed. Every CW operator has his internal “clock” which must be synchronized with resulting signal (ie. frequency of dits). Any latency presumes to be subtracted from the audio output heard. As a result of observation of hundreds of CW operators (any knowledge level) is that any latency between a paddle touching and audio output > 10 ms is a disadvantage at speeds above 35 WPM, the operating become to be tiresome and exhausting.

**2. MONITOR** – the actual state. The actual output signal is MONITORED concerning frequency, shape and sidebands however the is some latency. My impression from operating at speeds > 35 WPM is that the output signal is “sampled”, ie. should be exactly synchronized with a “clock” which is independent on the used speed (this is only option possible when

using an external keyer). It sounds like the resulting dits does not have an uniform length but are "variable" depending on the used speed. Like how the speed matches the internal "clock", in some cases the dits CAN have an uniform length. When using the internal keyer this problem disappears.

It seems that the ideal solution will be the pure MONITOR when using the internal keyer but when using an external keyer a SIDETONE with zero latency between key-on state and audio output. Of course, there should be two selectable options - MONITOR and SIDETONE. Anyway if the relationship is "sampled" in either way use the highest possible sampling frequency, always > 1 kHz (remember that 100 Hz means reciprocally 10 ms which is TOO MUCH!).

This latency also badly affects the SSB operation. The resulting delay may force the operator to "unnatural voice performances" like yelling, speaking with artificial, strained voice and even falter (stutter) speech. **The MONITOR latency seems to be one of the worst**, badly and deeply affecting the operator's performance.

## Keying modes

Your presumption that the keyer built into the ExpertSDR2 1.3.1 update8 SunSDR2 DX Linux is using CURTIS A MODE is definitely wrong. It is MODE B. I was wrong in my previous statement that this is mode A because I reproduced Roman's comment before my SunSDR2 DX arrived. Putting my SunSDR2 DX onto air confirmed that it is definitely mode B! TNX SM5IMO for notice!

There are many reasons for use of an external keyer. It can be the ULTIMATIC MODE, see <https://la3za.blogspot.com/2013/05/is-ultimatic-morse-keyer-really-that.html>

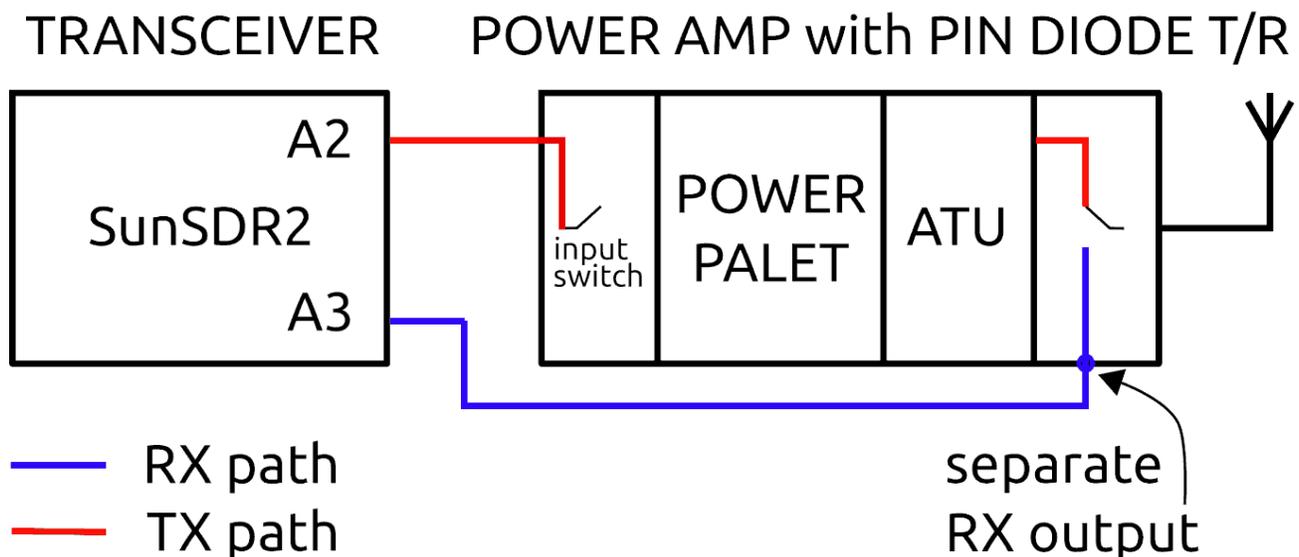
or many other references on the Net. Another good reason is keyer interaction with particular program, ie. *Autosend* option in TR4W or just the simple fact that the user is well accustomed to his keyer used for years. Anyway, the external keyer complicates the way to (almost) perfect keying but has some advantages why the user prefers it and refuses to use the internal keyer.

**Warning** – the quite new TCI protocol would need years to cover all the subtle nuances of operating techniques. Although it is full of great new ideas, it can't turn back to the years proven techniques which became domestic among DXers and contesters. Even full implementation of all 3 modes (Curtis A, Curtis B and Ultimatic) would not eliminate the need of an external keyer under particular circumstances.

I would urge to minimize the latencies! To be in proper picture, a latency of units of milliseconds can be acceptable however 10 ms and more is TOO LONG!

## TWO HOLER CW RIG

A very simple addition would turn the SunSDR2 (DX, Pro or QRP) to an almost perfect CW rig! Simply use the A2 antenna connector as TX output and A3 as RX input and disable the annoying antenna relay (deactivate, prevent from switching). We can rely on the AMP with PIN Diode T/R switching, there will be an independent RX path. The AMP modification will be very easy (if it is not already made) and amps like KPA500, KPA1500, RF2K-S (by RF Kit in Germany) and many others (some with QSK-1500 add-on board) may provide the T/R switching. Unless 100 watt barefoot operation, the SunSDR2 does NOT need any T/R antenna switching. Let's hope that the antenna relay is the ONLY mechanical relay involved in the T/R switching procedure.



## E-coder - chainable commands assigned to the buttons (-> instant SPLIT)

(already published <https://eesdr.com/en/forum-en/e-coder/9112-e-coder-chainable-commands-assigned-to-the-buttons-instant-split#18418>)

A very useful idea borrowed from the Elecraft K3 - the commands assigned to each E-coder button should be chainable, ie. there should be possible to assign more commands to a single button. This will form a macro. Very useful for split operation if:

- 1) VFO B QSY to VFO A frequency + 1 kHz (UP 1 is a good starting point)
- 2) The E-coder main tuning knob assigned to VFO B
- 3) SUB RX ON and assigned to VFO B

I used similar arrangement with my K3 for years. It saved lot of work and lot of errors....

## **MAIN TUNING KNOB ASSIGNED TO RIT OR XIT**

Current Linux version of ExpertSDR2 1.3.1 does not accept RIT/XIT operations via E-Coder. It allows to switch RIT or XIT on but does not allow any tuning. This is a bug.

A possibility to switch on the RIT (XIT) ON and simultaneously assign the E-Coder main tuning knob to RIT (XIT) with a single key on the E-Coder panel would help a lot. The current VFO frequency will be fixed (locked) to the momentary value. A second touch to the RIT or XIT ON key will switch the RIT (XIT) OFF, unlock the VFO and reassign the main tuning knob to tuning (its main function).