Read me file LU7DZ ADX FIRMWARE with CAT and CW Implementation

LU7DZ Pedro E.Colla modified stock ADX firmware V1.1 adding CAT and experimental CW capability.

Below listed posts are from Pedro's ucx iogroups posts about his firmware and how to use it plus invaluable experiences and tips. I would like thank Pedro for his contribution to ADX project.

Ucx IOgroups address of this post:

https://groups.io/g/ucx/message/16311

Hi,

After having built 3 different ADX transceivers (1 by wire and 2 using PCBs), and having completed the craft of version 1.2e of the firmware including all factors of my interest at

this point I believe it's time to share some conclusions.

The operation is smooth and it performs extremely well, I've used it on 40m so far with global coverage, my transceiver is set for 2W outputs (blew the finals of one when trying

to extract too much power with very bad SWR). I'm using it with WSJT-Z running at a Raspberry Pi model 3B.

I did changed my mind dramatically since I started with the project; at the beginning I started by thinking in lots of features and functions starting with the "base" design, as an

example adding an OLED, better interface, a controlled fan to keep it cool and so on. As I progressed with the project and build several units I realize that extending the

capabilities would make either an oversized-ADX or a downsized-uSDX; an I don't need any additional uSDX as I have several. I started to grasp that the true beauty and virtue

of the ADX transceiver is how capable and performing it is within it's extreme simplicity and easy of building. I re-baselined my entire approach to the project after that.

I've added CAT support because I found it a more reliable way to operate the transceiver, specially for my remote station; and I added CW because using it on the open would be good to have that capability for an ocassional operation in CW, and the watchdog again for my remote station and the EEPROM conservative use because a continuous

write cycle wipes out Arduino boards sooner than realized, specially with a tunning capability such as the one provided for CW usage.

For the CW usage I provide more a RIT capability than a tune capability, much in line with what is available on small xtal based QRP rigs such as the Pixie or a 49er than a full fledged transceiver. I found myself using the rig mostly for FT8 and ocassionally for WSPR to evaluate antennas.

On the hardware implementation I think it's extremely well conceived, so it's difficult to produce change recommendations which aren't at the same time against the minimalist

spirit of the device, but will try with some.

The tip and sleeve connectors for the SPKR should be at least separated (today are both tied and ran to the same Arduino pin). If at all possible routed to different Arduino pins, if not a wire connection can be run. Perhaps to mimic the interface of the uSDX would be good and that solve the CW interface.

All test pins holes should be separated for a std 2.54mm, this will allow to solder few directly from the comb, otherwise the soldering becomes more difficult.

The filter is very well designed not to feed finals without it, +12v is run thru it, so if by mistake We transmit without it We don't blow the finals. This should be extended a step further and allow one line to run into an unused pin to read if the filter is connected or not, being able to provide an error condition if not.

Copper stripes around the finals should be made stronger and wider, to be more resilient to at least one soldering-desoldering cycle, BS170 is an inexpensive device but to replace them can produce difficulties with the board.

A variant of the above is to run 2 lines but perform a binary coding at the filter by connecting or not two jumpers (00,01,10,11) in a fixed way on each filter programmed, therefore the firmware can check if the filter is the one appropriate for the band being selected (or even refuse the band change).

Some form of SWR protection. If at all possible use a rudimentary power read at the output connected to analog pins (then program the firmware to show some reading using the existing LED as a rudimentary meter). This can be invaluable for a mobile operation.

Make the SPKR filter enter a micro potentiometer to fine tune the level, I found necessary to custom pick a different resistor to optimize the VOX operation.

I made a 3D printer model for the case but it still is in the horrid phase with many inaccuracies making advisable not to publish it yet, but a good case design would be a very nice feature is somebody is willing to step in.

73 de Pedro, LU7DZ

CW support for ADX by LU7DZ

UCX IOgroups post address:

https://groups.io/g/ucx/message/16325

CW support involves several modifications to the firmware when enabled at compilation time (#define CW 1). First a mod to allow the change into CW mode (and out), this is accomplished by introducing meaning to the "long push" of a button, long pushing UP and it will enter CW mode and long pushing DN out of it.

The second change is how to show your position in the band without adding an OLED/LCD, so the UP/DN buttons will move in 500 Hz increments either up or down the initial frequency (which is by program configuration the QRP calling frequency of each band), being at "exactly" that frequency will make both central LED to be on, while shifting up or down will make the LED pattern to change every 5 KHz or so. Going beyond that would make the extreme LED to flash. There is a configuration restriction (not activated by default) to avoid going beyond 15 KHz at either side. Without it you might risk to go off band, perhaps some feature for the future is to guard for band limits.

When thinking on implementing this mode I have a Pixie transceiver in mind, so it's more a RIT than a tuning aid and it's inherently an auditive tuning process rather than a clearly marked one.

When commanded into transmission, i.e. with the TX button, the VFO frequency is shifted +600Hz (configurable by compilation) and returned back when in RX mode, I didn't implemented a CW-R mode (going down instead of up when keying).

I have no reports for clicks with the tests QSOs I did, but I can observe and hear them myself, so probably some further work needs to be done on the raising envelop of the finals. At this point the keying is performed with my keyer in paralell with the TX button and a small trace cut on the ADX board to use the "tip" of the SPKR jack to plug my keyer.

A horrid and untested way to do CW would be to use what is called "Audio CW" where the tone out of the PC speaker is the keying signal, both FLDigi and MixW support that (in order for it to work with my mod the shift needs to be cancelled out in the firmware).

Regarding the reception I found the CD2003 a good performer, no filter is on the ADX as there is no or little filtering on other QRPp rigs such as, again, the Pixie transceiver. Personally I don't use filters, working pile ups during contest I just sort out signals on my head, but without that you can easily perform casual QSO

that way if the band isn't crowded. The kind of supplemental QSO you might want to do ocassionally with a QRP rig.

I don't think the primary purpose of the ADX transceiver is to perform CW, it's just a handy feature that you might want to exercise ocassionally and be glad to have it if you want to use it.