**[Updating Firmware on uSDX+ – KC1SLE](https://kc1sle.com/updating-firmware-on-usdx/)**

**Updating Firmware on uSDX+**

[March 3, 2024](https://kc1sle.com/updating-firmware-on-usdx/) by Trevan

I recently acquired a used –and seemingly a little faulty– uSDX+. These are the little SDR transceivers all over eBay based on the tr(uSDX) devleopment and produced by a plethora of vendors of varying quality. I believe mine is what is called the ‘white buttons’ model — in reference to the different variations that turn up– this has white buttons.

The one I have seems to have some hardware faults that give it almost too much noise to hear a signal over. Somewhere I came across a page the suggested that some of the firmware used on these is very outdated and could benefit from updating. There’s a process documented elsewhere but almost always involves using an Arduino Nano or similar as a programmer.

I thought I’d share the process I used instead in case anyone else is hoping to bypass the Arduino as programmer as and just use an existing programmer they may have. I didn’t find anyone who had done this. It’s actually a fairly straightforward process, but as with anything DIY, I am not responsible if this process does not work for you or causes damage to your device.

1. Locate the firmware you want to use – in my case, I went with the latest release of [uSDXOpen](https://github.com/GW8RDI/uSDXOpen) for my “white buttons” model. Simply download the .ino file for your model.
2. Install (if you don’t already have) the [Arduino IDE](https://www.arduino.cc/en/software). You can now open the .ino file with the Arduino IDE and read the code and the various notes included. **\*Read it carefully.\* There are notes about removing a capacitor from the radio and disabling the internal mic before programming otherwise the MCU may be damaged.**
3. Open up the top cover on the uSDX and locate the JTAG pins
4. Locate a suitable programmer – I used a cheap USBasp programmer from Amazon.
5. Connect up the programmer to the JTAG pins, making sure to get the direction correct. (My radio included no indication of which pins were which and I had to try twice to get it right- thankfully no damage occured from my error.)
6. Plug in the programmer to your USB on your computer. In my case, at this time the radio turned on and made very loud noise.
7. Download, install and open the program [Zadig](https://zadig.akeo.ie/) (a driver utility).
8. Under Options, select List All Devices.
9. Select “USBasp” (or your programmer) from the list of USB devices then, under target driver select “linusb(K)”. Then click ‘Replace Driver’
10. Once that completes, open up the .ino file you downloaded with the Arduino IDE.
11. Read through the code and make necessary configurations (setting your call sign as desired and various functional options). Once you’re happy, under Tools, select Boards > Amtel atmega328p Xplained mini. If you do not have this board, you may need to install it under Boards Manager.
12. Then Tools > Programmer > USBasp (or your programmer)
13. Then go to Sketch > Upload Using Progammer. The code will then compile and if successful begin uploading. You may also receive one or more errors you will need to investigate, including a problem with the size of the code (you’ll need to shrink it by disabling unwanted features), trouble starting the programmer (if the drivers or somethign else is wrong), or a number of other warnings or errors.
14. If successfully completed, your radio can be disconnected and closed up.
15. I recommend on the first boot to do a full reset — in my case this was done by holding the first button (Menu) during power on.
16. Enjoy your new firmware!

Unfortunately, in my case, the firmware did not solve the noise issue- so I’ll be tinkering a bit more in the future. Looking at the PCB, there are numerous missing components from the board based on the silk screen, so I suspect one or more of these missing parts may account for the poor performance….