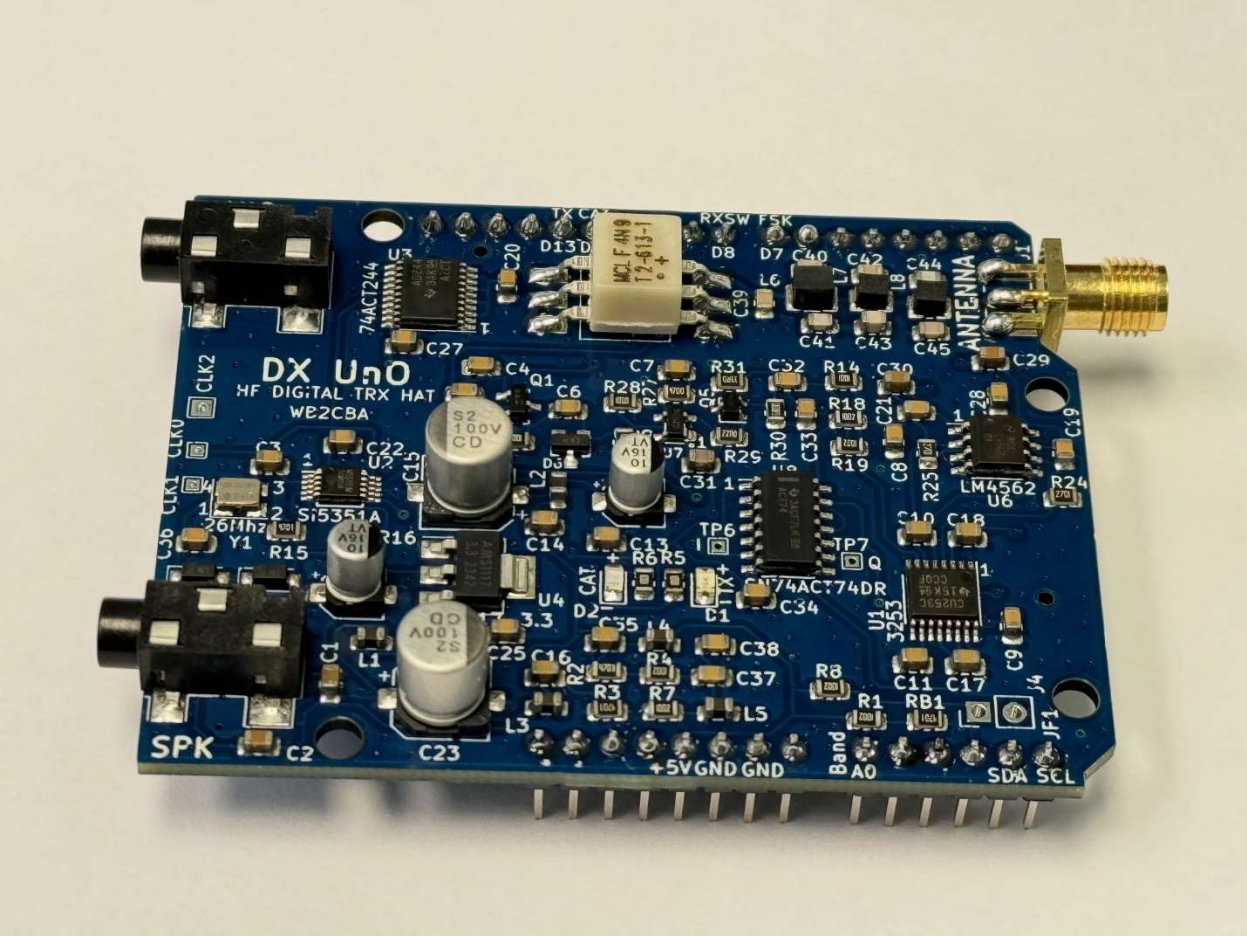
DX UnO – Arduino Uno based Multiband QRPp HF Digital Modes Transceiver

Build and Operation Manual for DX UnO PCB Version 1.0

Manual Rev 1.0 - 06/2024

DX UnO files and all relevant info can be found in DX UnO Github page link:

https://github.com/BluQRP/DX_UnO/tree/main



DX UnO is an Arduino Uno based HF Digital Modes QRPp Multiband Transceiver. DX is abbreviation for **D**igital **X**ceiver.

DX UnO is a 5 band digital modes optimized HF QRPp transceiver that can be plugged on top of an any Arduino Uno board with Atmega328p microcontroller to create a 400 milliwatts average RF Output power QRPp portable HF Digital Modes capable transceiver. QRPp means the DX UnO Transceiver operates below 1 watt RF output power.

DX UnO can operate on any of these five bands of 20m, 17m, 15m, 12m and 10m bands and operates on popular digital modes such as FT8, FT4, JS8call and WSPR and similar tonal modes.

DX Uno is fully CAT controlled and can be used with any Software that emulates KENWOOD TS-2000 with 9600bps.

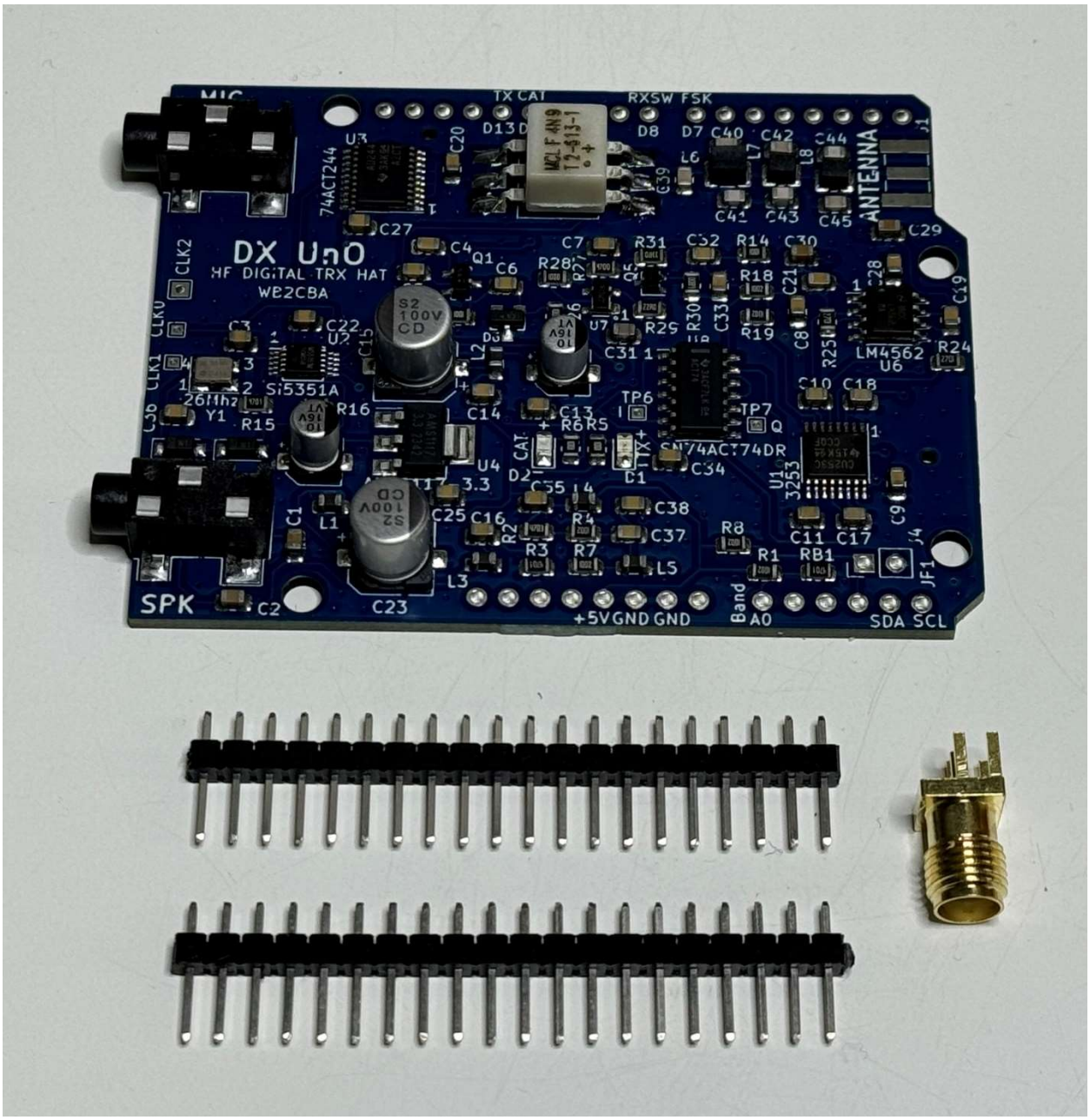
DX UnO uses a TTL based RF PA, not PA mosfets which are prone to failure. The advantage of this TTL Logic Chip based RF Power Amplifier is that it is extremely resistant to SWR mishaps such as high SWR cases and antenna mishaps as antenna short or TX without antenna. These conditions won't damage the RF PA section of this rig. This makes this tiny transceiver due to its resilience and small size an ideal portable rig for SOTA/POTA Activations and for beginner ham radio operators with varying antenna conditions which are prone to errors.

There is no external power supply to power DX UnO Transceiver. PC/Tablet USB connector powers the transceiver through USB cable which is also CAT control data connection. MIC, Microphone and SPK, Speaker outputs go to relevant Microphone and Speaker outputs of PC or Tablet where the digital modes software operates.

DX UnO kit is designed to be an easy transceiver kit to solder, operate and experiment with. It is a suitable kit for a beginner and for seasoned ham operators to carry in their backpack for SOTA/POTA activations or for travel. The advantage of portability comes from simplicity of no need for external power supply and resilient antenna conditions RF PA.

It can be used with other microcontroller families also such as Raspberry pi or pi pico etc for experimentation. The audio output jack which is denoted as MIC has both I and Q outputs of TAYLOE Quadrature Sampling detector which can be used to experiment with SDR SSB decoding firmware or with SDR PC applications like HDSDR.

DX UnO Kit contents and BOM(Bill Of Materials):

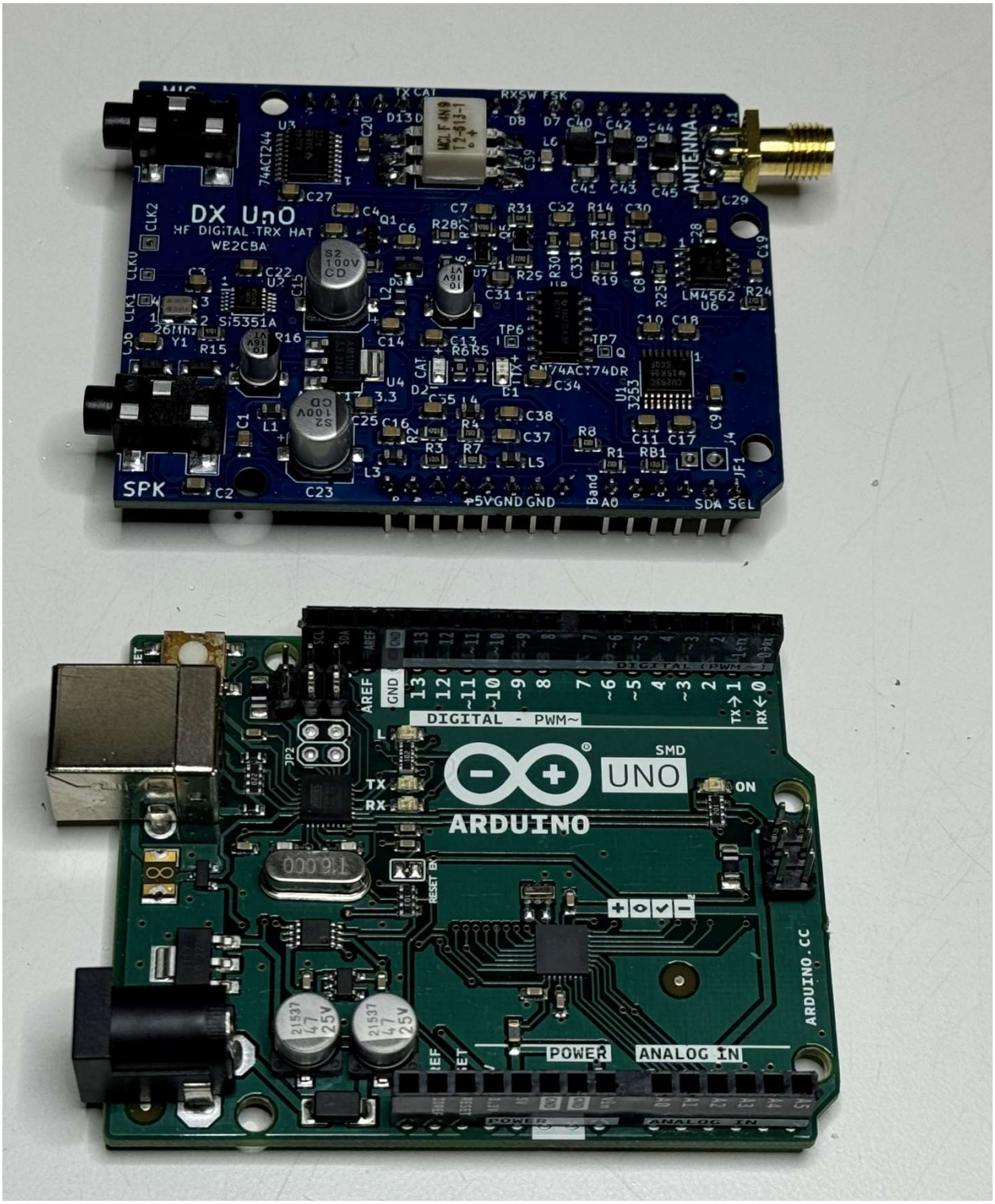


1x DX UnO SMD populated PCB Arduino shield.

2 x 20 pin 2.54 mm spacing male headers.

1 x PCB edge type solder SMA Female connector.

To operate DX UnO an Arduino Uno board is required which will be supplied by the builder/User. Arduino Uno is not included in the DX UnO kit.



1- Building DX UnO Main Board:

DX UnO Main Board Building Steps:

STEP 1:

First cut these headers from Male 20 pin header stacks supplied.

4 male pin headers to be cut from 2 x 20 pin male headers:

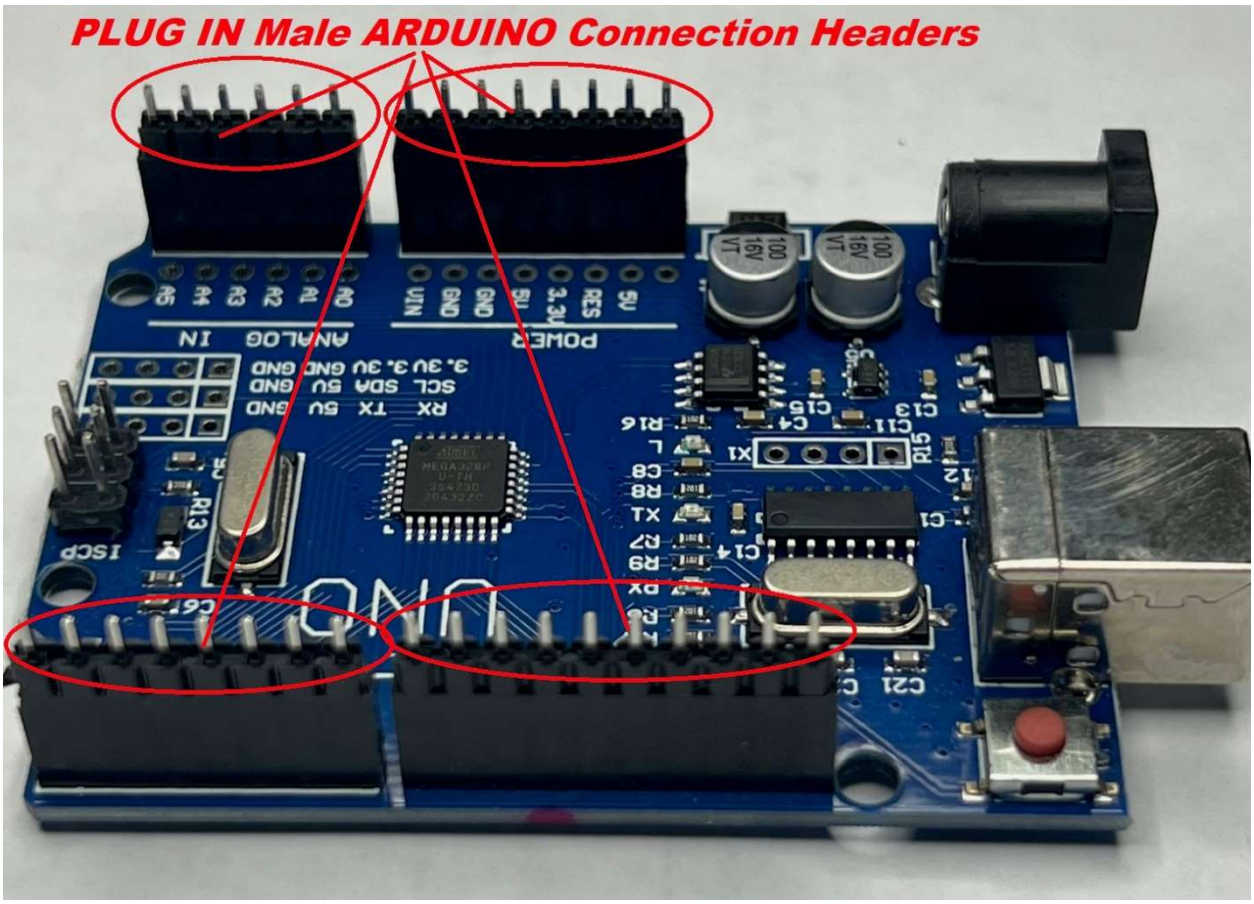
2 x 8 pin male header

1 x 6 pin male header

1 x 10 pin male header

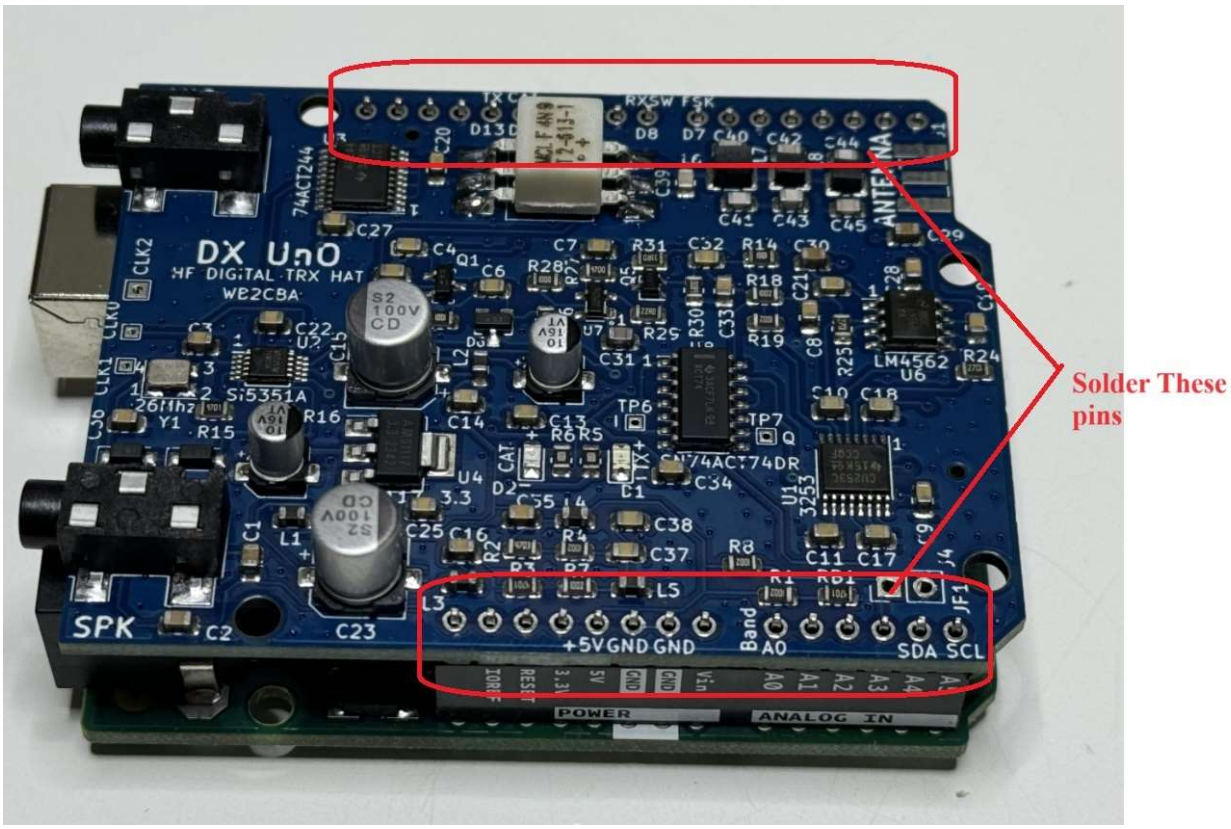
STEP 2:

Now plug these male pin headers in corresponding female pin headers on your Arduino Uno Board:

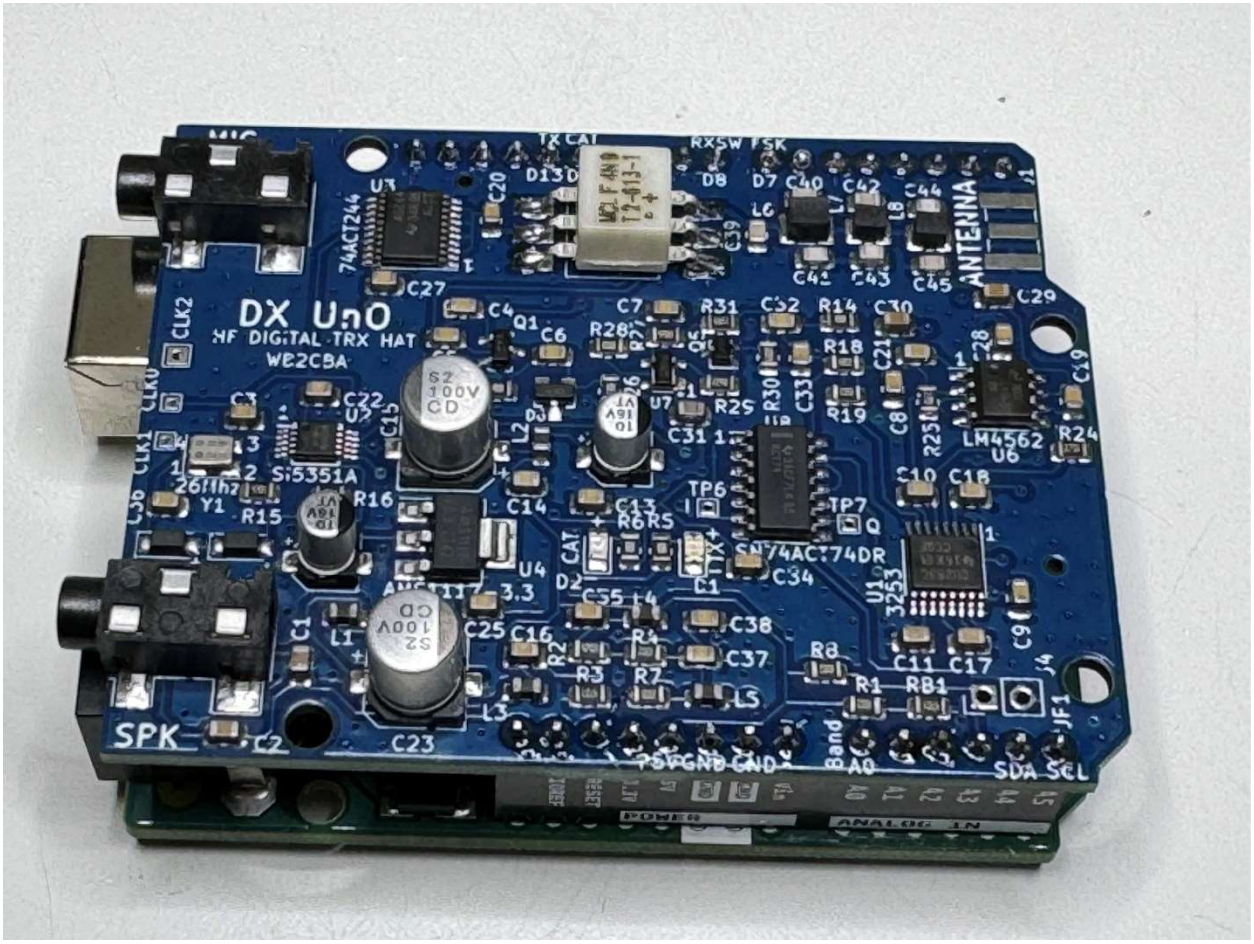


STEP3:

Now insert DX UnO board on these male headers gently and firmly matching headers and then solder all pins:



Finished and soldered DX UnO Main Board should look like this:



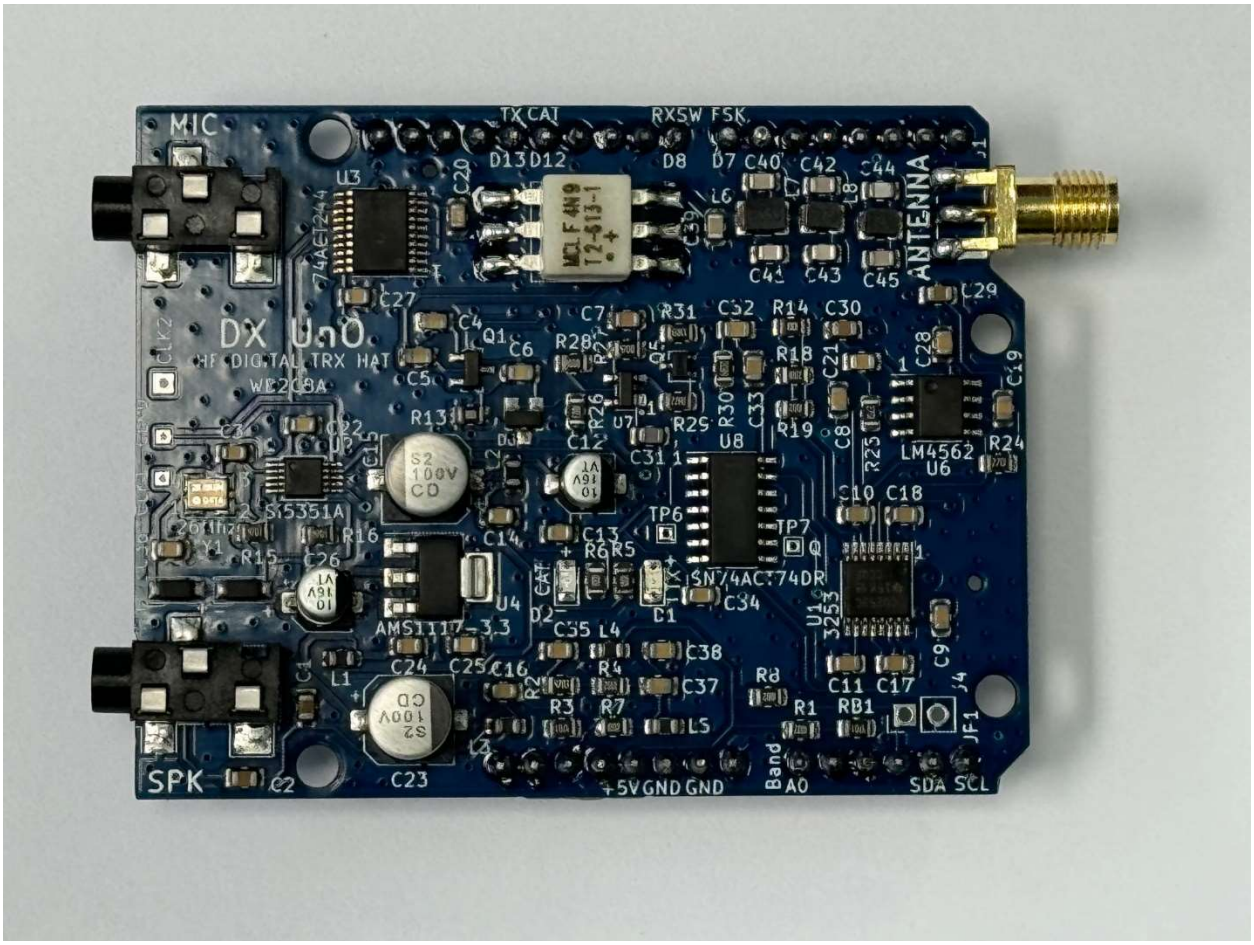
STEP 4:

Now solder PCB edge solder type SMA female connector from top:

Don’t forget to solder the bottom side of SMA connector too! This is important to create a mechanically sturdy connector.

This completes all soldering steps of DX UnO and now DX UnO is ready to be plugged on Arduino UnO.

The completed and soldered DX UnO from top should look like this:



And this is the bottom view of completed DX UnO main board showing Arduino Uno male headers and SMA connector:

DX UnO plastic transparent case can be used for storing DX UnO/Arduino Uno combo! Dx UnO plugged to an Arduino Uno can fit perfectly to this transparent case and can be used to carry DX UnO in a backpack and protect the DX UnO TRX from any damage.

2- Programming Arduino Uno with DX UnO FIRMWARE:

DX UnO firmware is developed in Arduino C language and Arduino IDE can be used for visualization and experimenting with it.

DX UnO uses **Arduino Uno** board as microcontroller. This is a readily available and cheap Arduino development board with an Atmega328P microcontroller that can be outsourced easily online. DX UnO works with Arduino Uno boards that has Atmega328P microcontroller.

DX_UnO_H_V1.0.ino is the DX UnO firmware file and can be downloaded from DX UnO github page Firmware Directory:

https://github.com/BluQRP/DX_UnO/tree/main/DX_UnO_Firmware

There is also **a DX_UnO_H_V1.0.hex** file which can be downloaded from same github link. This file is to be used with Xloader uploading method.

There are two methods to upload DX UnO firmware to Arduino Uno:

1- **XLOADER Method (Easy Method)**
2- **Arduino IDE Method or Classic method**

Method 1: XLOADER Method:

Xloader is an application to upload ".hex" extension firmware files to Arduino controller boards without the need of using Arduino IDE.

Hex files are already compiled firmware files. The advantage of these type of file is user does not need to install IDE environment and compile etc. All taken care of.

It is just a matter of uploading to Arduino microcontroller via **XLOADER** upload application.

Xloader runs only on Windows operating system.

STEPS for Uploading DX UnO Firmware to Arduino Uno via XLOADER Application:

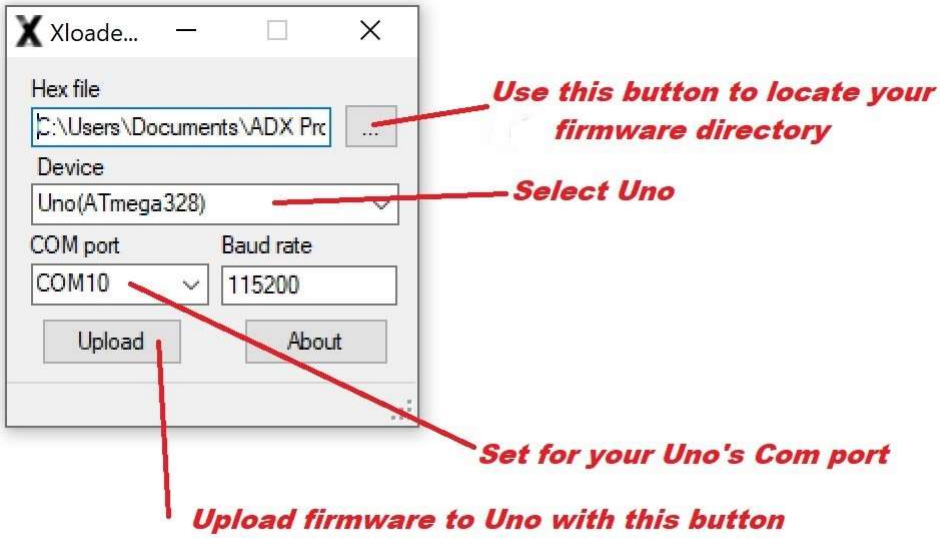
1- Download Xloader.zip file from DX UnO Firmware Github Link and unzip:

https://github.com/BluQRP/DX_UnO/tree/main/DX_UnO_Firmware

2- Download **DX_UnO_H_V1.0.hex** firmware file from DX UnO github page under DX UnO Firmware:
https://github.com/BluQRP/DX_UnO/tree/main/DX_UnO_Firmware

3- Now plug in Arduino Uno to PC. DO NOT plug in DX UnO on top of Arduino Uno for this process of uploading firmware.

4- Run XLOADER. Below screenshot will be seen:



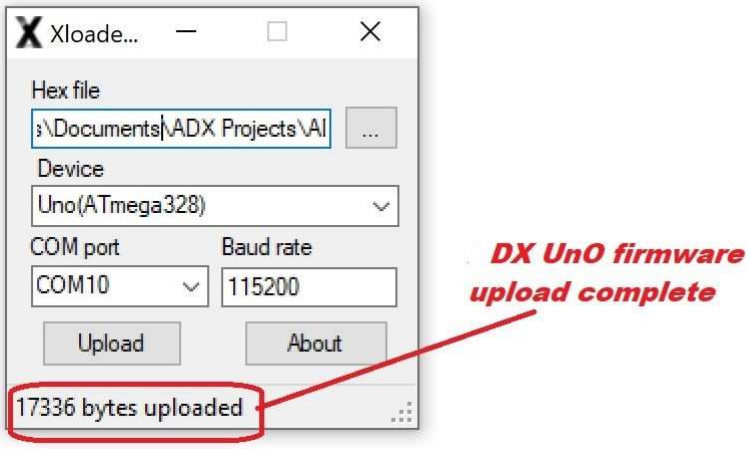
Now locate your DX_UnO_H_V1.0.hex file previously downloaded and saved using the file search button of Xloader.

As Device set **Uno(ATmega328)** from drop down menu.

Select COM Port of Arduino Uno from Com port drop down menu.

5- Now click on Upload to upload DX UnO firmware to Arduino uno.

Upload should be pretty fast. Below screenshot shows successful Upload:



DX_UnO_H_V1.0.hex file is configured for bands 20m, 17m. 15m, 12m and 10m operation as default bands.

Method 2: Arduino IDE Method or Classic method

In this method we will use Arduino IDE to compile and upload **ADX_UnO_H_V1.0.ino** firmware

DO NOT plug in DX UnO on top of Arduino Uno for this process of uploading firmware.

First step is to download and install Arduino IDE from www.arduino.cc

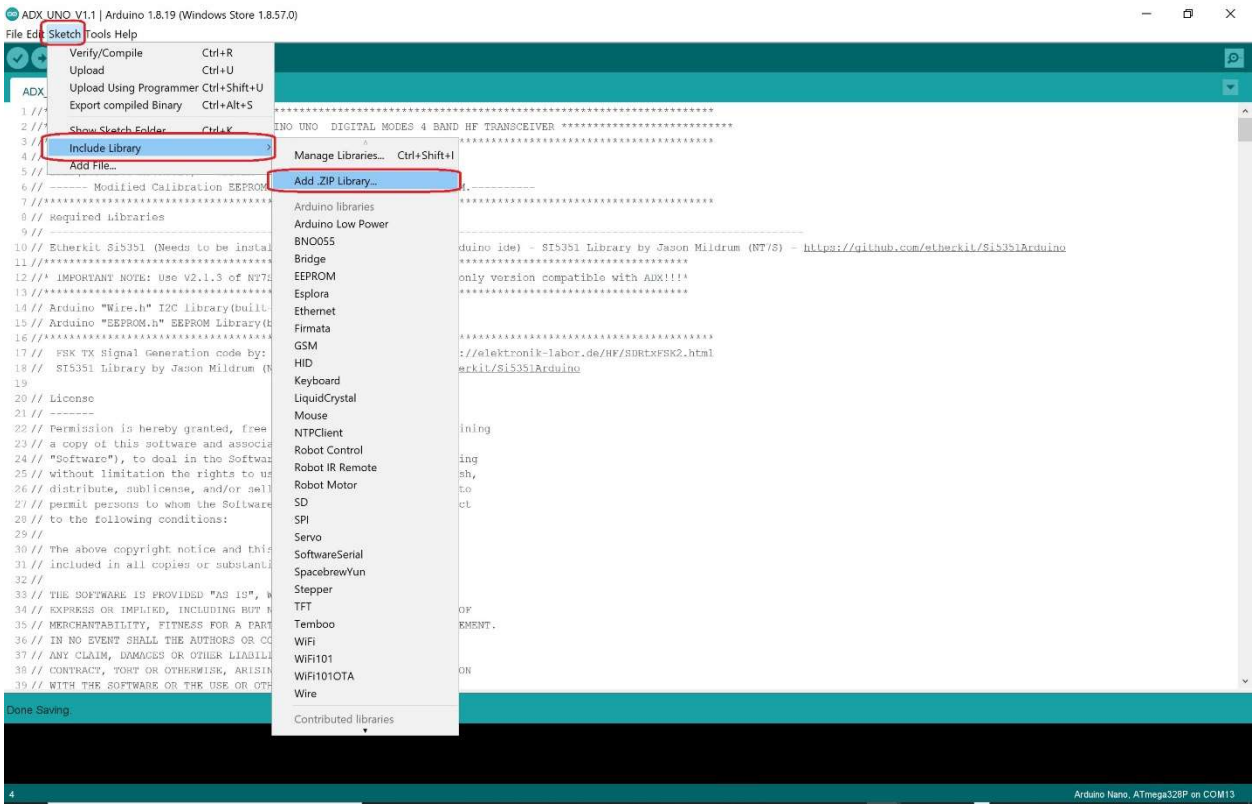
DX UnO PLL VFO, SI5351 needs an Arduino library to work with. This library is Jason Mildrum, NT7S's SI5351 library with Version 2.1.3, **Si5351Arduino-master.zip** .

DO NOT USE Jason Mildrum NT7S's latest version of SI5351 library from his github site as it is not compatible anymore with DX firmware. Use SI5351 Library that is in DX UnO Github firmware page to avoid any I2C communication problems. DX UnO is using version 2.1.3 of NT7S SI5351 Library.

To install SI5351 Arduino Library Follow these steps:

1- Download **SI5351Arduino-master.zip** from DX Uno github page under DX Uno firmware directory: https://github.com/BluQRP/DX_UnO/tree/main/DX_UnO_Firmware

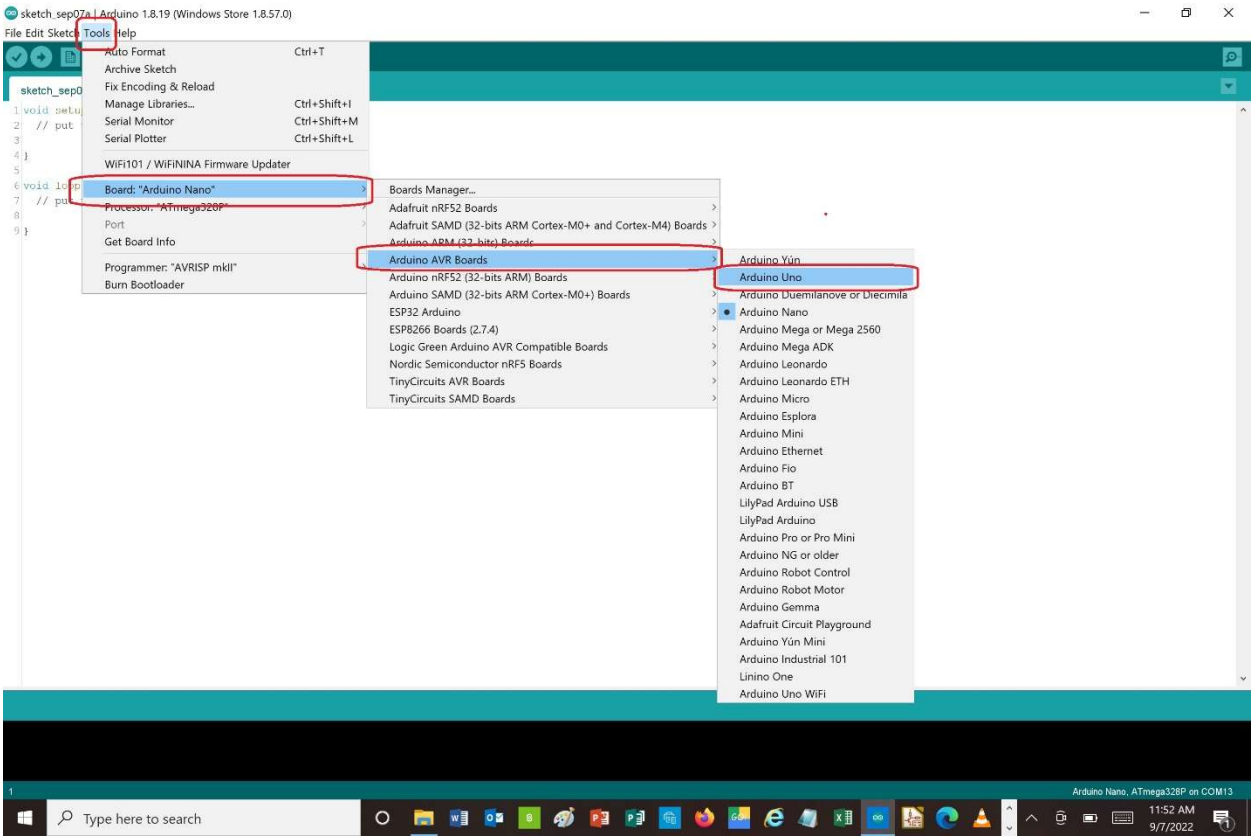
2- Use Sketch/Include Library/Add .ZIP Library menu. Just select **SI5351Arduino-master.zip** file and it installs. **Do not unzip the file!**

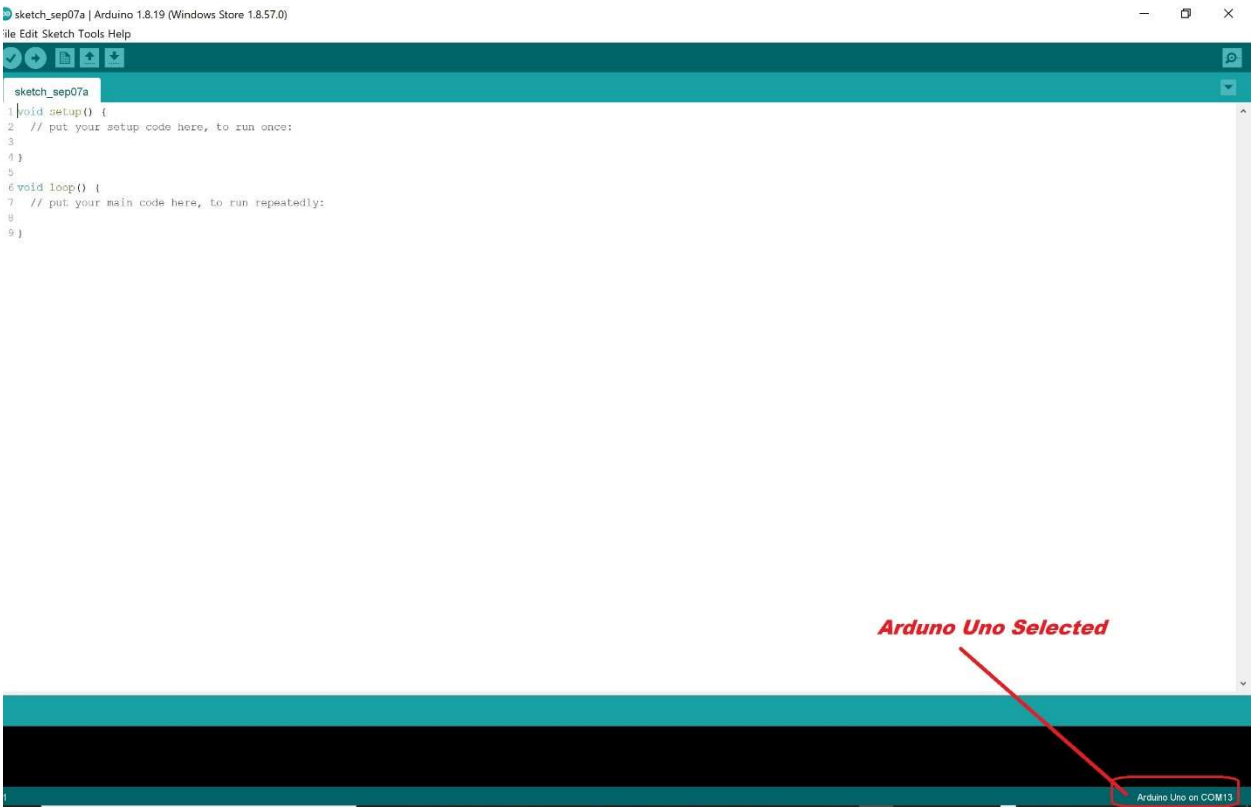


- Now Download **DX_UnO_H_V1.0.ino** firmware from DX UnO Github Page under DX UnO Firmware directory: https://github.com/BluQRP/DX_UnO/tree/main/DX_UnO_Firmware

Use “ File/Open” menu to open DX_UnO_H_V1.0.ino firmware.

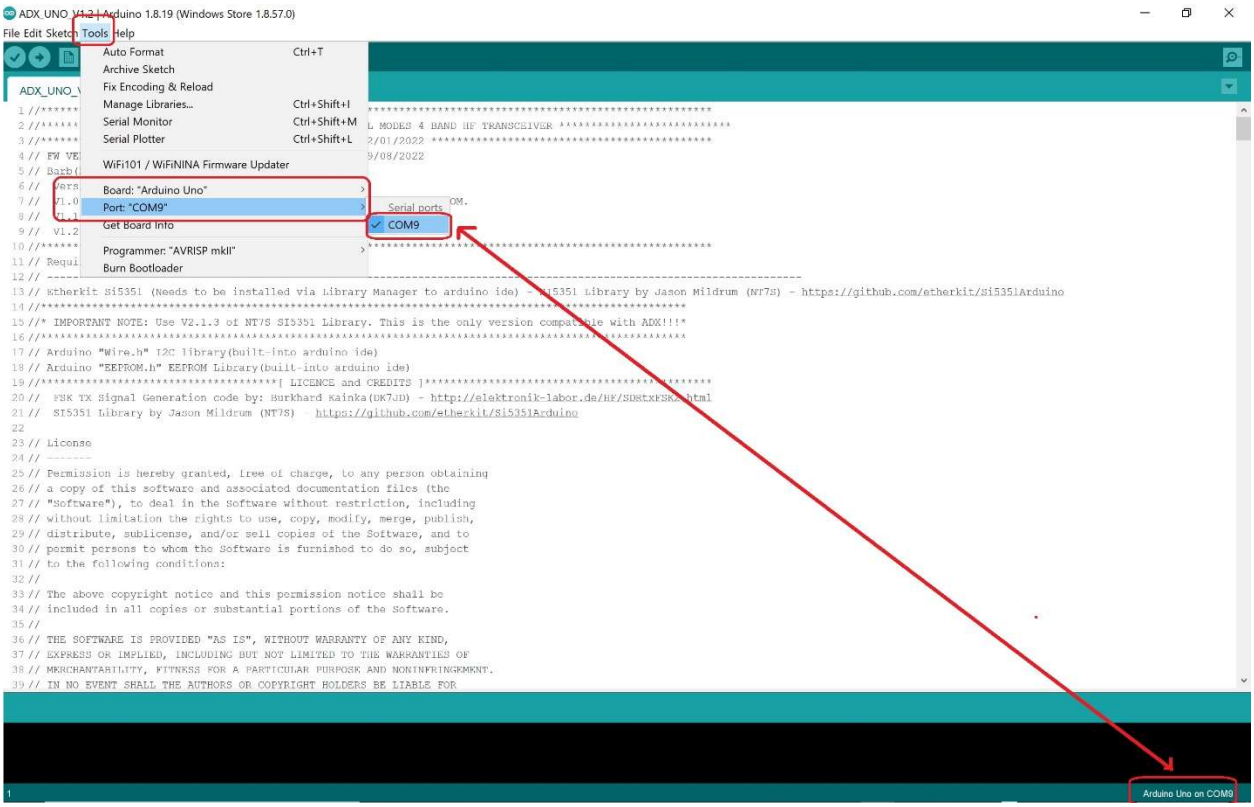
Select Arduino Uno Board following below menu items.





If Arduino Uno is selected, then at the bottom right corner it should be indicated.

Now select Arduino Uno Com port:



- Now upload DX UnO firmware by using → button which is the second button on top left corner.
- Arduino IDE will first compile firmware which will be indicated with a green progress bar at the lower right corner.
- When upload is done there will be a message at the lower left corner as "**Upload Done!"**

This ends DX UnO firmware install.

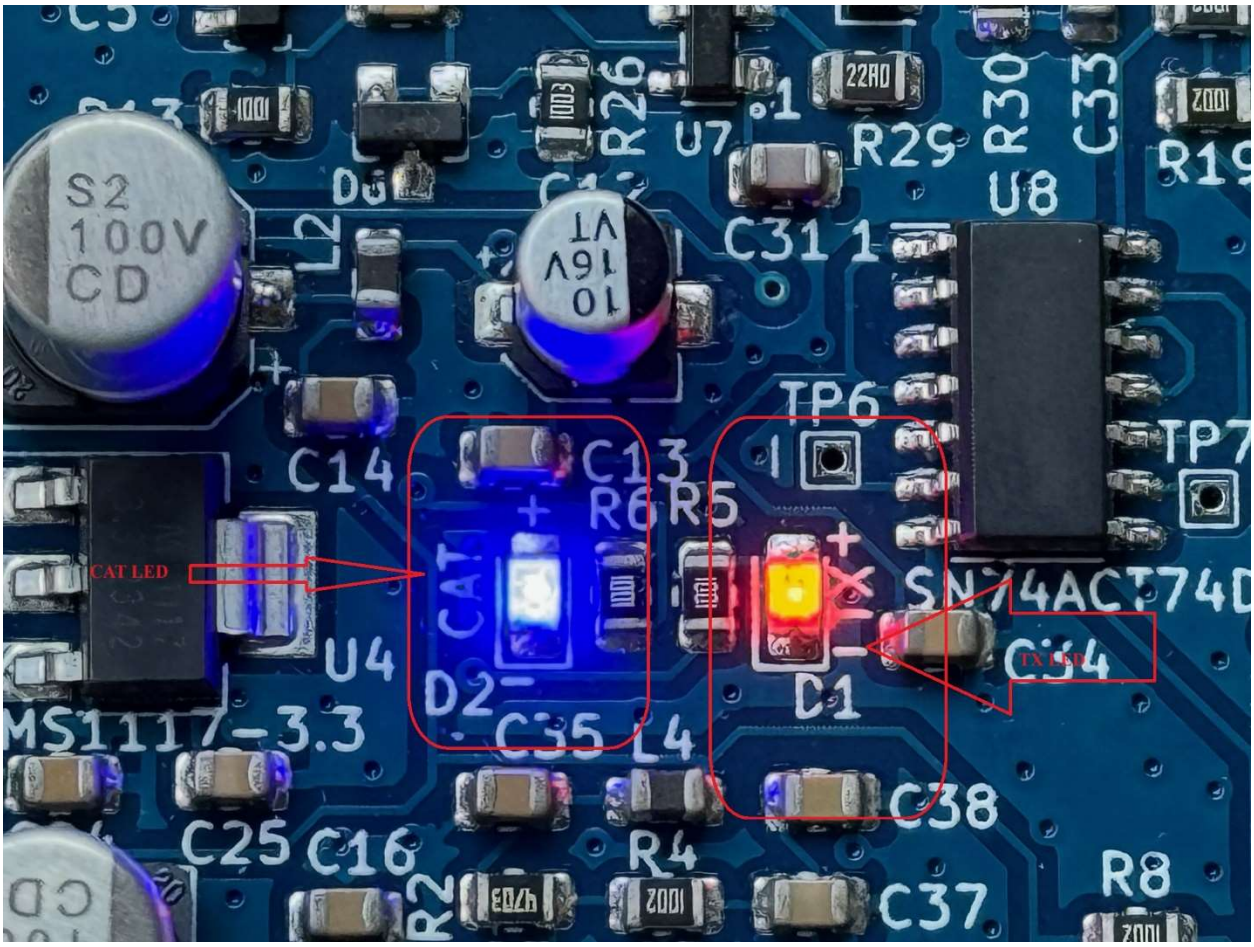
3 Power up and smoke test!

After firmware upload we are ready for first smoke test! Plug in your DX UnO to your Arduino Uno.

Now connect your USB cable to your Arduino Uno with DX UnO plugged on top of it.

The signature of life to show your programming is successful is the flashing Blue LED which is CAT LED, 3 consecutive times back to back. After blinking 3 times it will be off.

If this is the case then all is good and your DX UnO is alive and kicking! If no blinking Blue led then the programming is not successful so revisit programming steps whichever method is used.



DX UnO draws less than 500 mA on RX and TX from USB +5V. This current consumption is allowed by the specs of USB port of any PC or Tablet.

DX UnO works with +5V supply from USB port of PC or Tablet and do not need any external power to operate!

4 Connecting DX Transceiver to Computer and firing up DX UnO:

1- Connecting DX UnO Transceiver to any computer is pretty straight forward. We need a MIC which is Microphone input and SPK which is Speaker input or headphone input on the PC or Laptop. We can either use PC built in soundcard or one of those cheap USB Soundcard adapters. I suggest to use a USB soundcard adapter for couple of reasons! This way if anything goes wrong built in soundcard or PC won't be damaged.

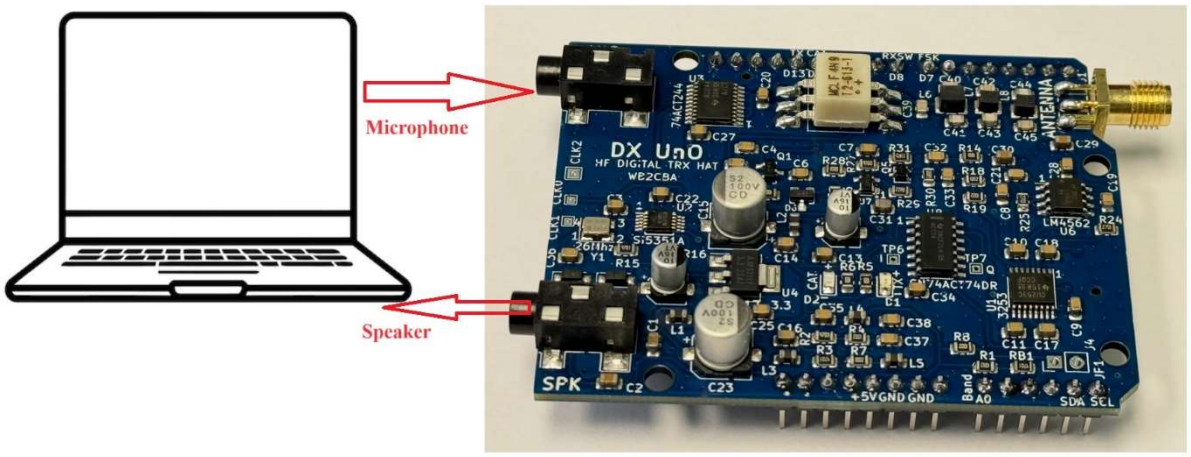
   For USB isolation and peace of mind from ground loop dangers you can use one of these before USB soundcard plug in:
   https://www.amazon.com/GeeekPi-Isolator-ADUM3160-Isolation-Protection/dp/B07QKYYCD8/ref=asc_df_B07QKYYCD8/?tag=hyprod-20&linkCode=df0&hvadid=366402536789&hvpos=&hvnetw=g&hvrand=10336854439692763421&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9003562&hvtargid=pla-814018215075&psc=1&tag=&ref=&adgrpid=75347436439&hvpone=&hvptwo=&hvadid=366402536789&hvpos=&hvnetw=g&hvrand=10336854439692763421&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9003562&hvtargid=pla-814018215075

   For USB soundcard adapter I use this sound card from amazon:
   https://www.amazon.com/Sabrent-External-Adapter-Windows-AU-MMSA/dp/B00IRVQ0F8/ref=asc_df_B00IRVQ0F8/?tag=hyprod-20&linkCode=df0&hvadid=312824707815&hvpos=&hvnetw=g&hvrand=7034552265858110987&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9004160&hvtargid=pla-563309581845&psc=1

- We also need 3.5mm audio jack male to 3,5 mm audio jack male extension adapter cables. Actually two of them!
  Again from amazon I use this:
  https://www.amazon.com/Syncwire-Braided-Auxiliary-Adapter-Headphones/dp/B01I0SI1SG/ref=sr_1_10?crid=33ELZO9OEPEH2&keywords=5mm+audio+cable&qid=1649788399&s=electronics&sprefix=5mm+audio+ca%2Celectronics%2C994&sr=1-10

  These are just examples. You can get anything similar to these.

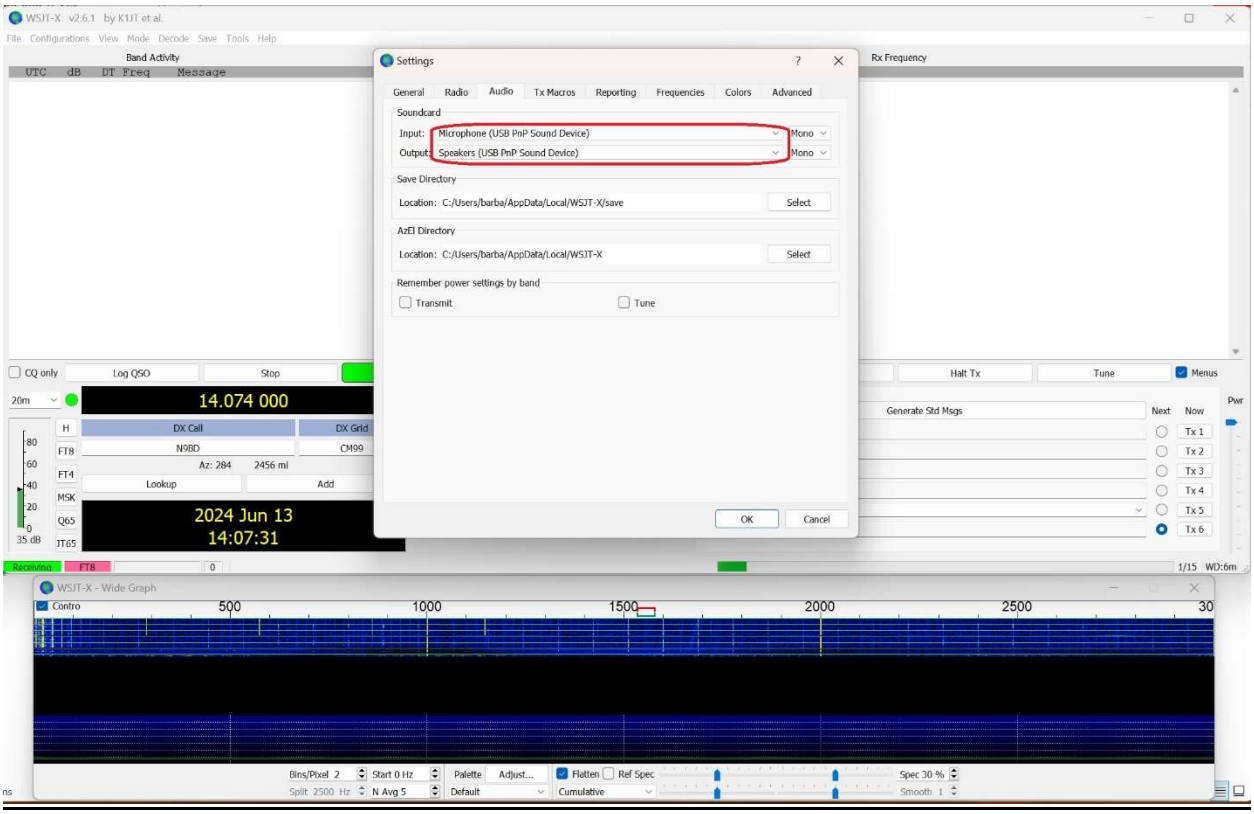
2- Plug in Arduino Uno USB to PC USB port of your choice.
3- Connect Soundcard MIC – Microphone input to DX UnO MIC input with one of the 3.5mm audio cables. Do the same connection from soundcard SPK - Speaker output to DX UnO SPK input with the other 3.5mm audio cable.



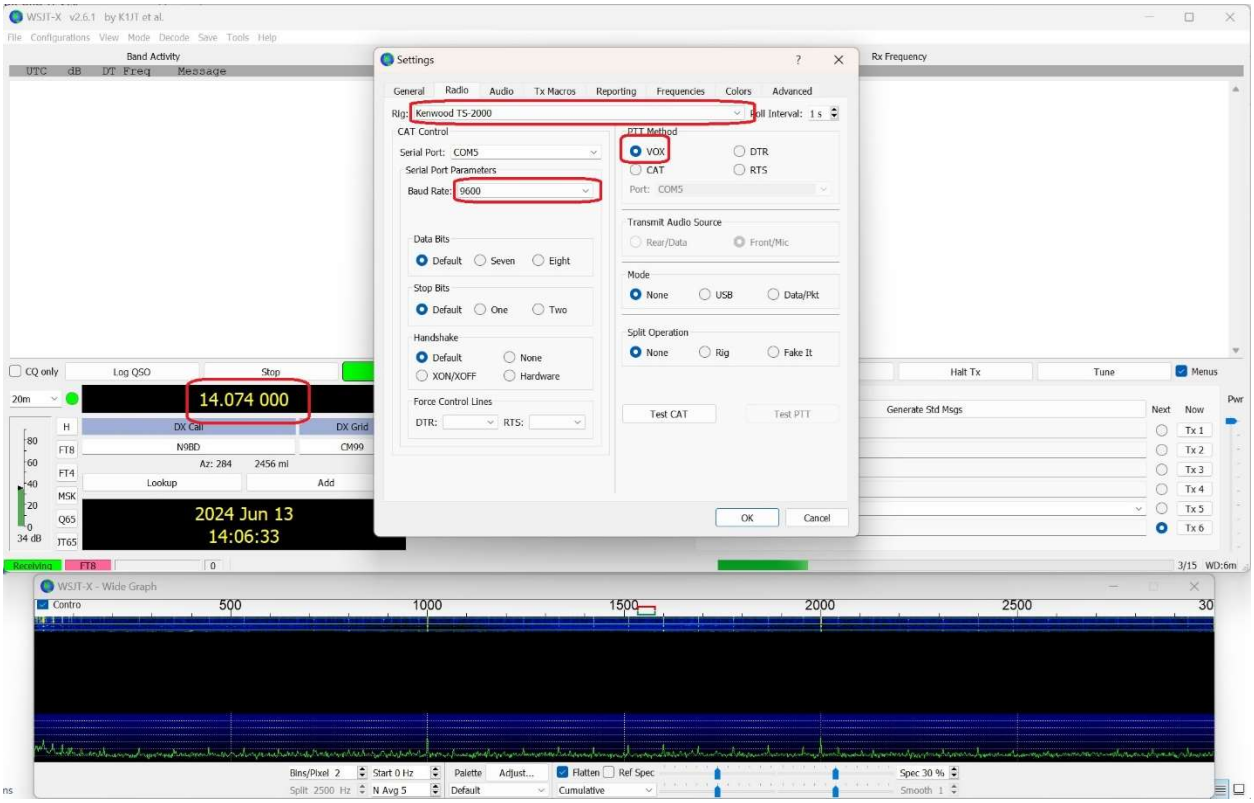
4- Run WSJT/X or any tonal digital modes application software.
5- When you plug in DX UnO CAT LED will briefly flash 3 times and will be off.

For setting WSJT/X to work with DX UnO:

- Choose your soundcard under Settings/Audio menu:



- Now to setup CAT Control Go to Settings/Radio:



- For Rig selection DX UnO emulates KENWOOD TS-2000. Select this from Rig drop down menu.
- Now choose your Serial port from serial port drop down menu.
- For PTT method select VOX.
- The rest will be default.

Now to test if CAT control works properly press Test CAT button. If the button appears in GREEN then CAT control is working properly. If the button appears in RED and there is an error message then check your serial port selection first by changing to a different port and trying test cat again. If there is still an error message it might be related to improper firmware upload so proceed to firmware upload and try again. If all is good and CAT is working then press OK to save CAT settings.

- When CAT is active Blue Led which is CAT LED will be on.

- Set Speaker Volume to **<u>100%</u>**

- DX UnO starts default with 14.074.000 Mhz which is 20 m FT8. User can select whichever mode or band they wish to operate from PC Software such as WSJTX.

- Now connect an antenna or a dummy load to ANTENNA connector of DX UnO and press TUNE on WSJT/X. TX LED(RED LED) will be on. If there is a flicker of TX LED then increase volume. Do not be afraid to increase volume to full. Nothing will break. DX UnO needs proper volume level to operate reliably. The safe level is 100%!

That's all you need to start working a QSO with WSJT/X. This applies pretty much to other software such as JS8Call or WSJT/Z etc.

5 Building blocks of DX UnO Transceiver:

- Brains of DX UnO is an Arduino Uno which takes care of signal generation from audio tones of digital modes, CAT control interface and frequency management of SI5351 Module. DX UnO uses direct FSK Signal generating method.

- Audio interface consists of a Speaker input from PC sound card which is actually an input to Arduino Uno analog comparator for tone frequency detection and sampling. Before inputting signal to analog comparator of Arduino it passes through a band pass filter which has an audio band pass frequency range of 500 Hz to 3500 Hz.

  This input acts like a VOX so no need for any PTT or serial PTT CAT input. When DX UnO hears a tone from for example WSJT/X audio output it starts transmitting. And when the tone of digital mode stops it stops TX. It's that simple. The AFP-FSK (Audio frequency processed Frequency shift keying) technique used in DX UnO is inspired from Burkhard Kainka(DK7JD) - http://elektronik-labor.de/HF/SDRtxFSK2.html work.

  ### So how does DX UnO actually work?

  Any digital tonal mode consists of varying audio tones that change frequency in relation to the data they correlate to. This audio tone generated for example for FT8 with WSJT/X software is passed through an audio band pass filter and then it is compared with Arduino Uno's Atmega328P processor A/D comparator for start and stop zero cross detection to determine period of that tone. From that period, frequency of that particular tone that is calculated and added to base transmit frequency of that mode, for example, 14074000 Hz for 20m FT8. If the tone let's say is 1000 Hz then the carrier TX is now 14074000 Hz + 1000 Hz = 14075000 Hz.

  As it is summed it will be in USB frequency range of any SSB receiver though the signal is not SSB signal still any SSB TRX set to USB can't tell the difference! This tone frequency detection and adding to base frequency is continuously repeated 400 times per second and refreshed until the FT8 tone transmission generation is over. In that case TX stops.

FSK signal generation technique is to generate a signal without any SSB signal mixing or filtering which eliminates problems of IMD or phase differences etc. which are unavoidable with SSB signal generation technique. FSK Signal Generation allows such a simple Transceiver to be conceived with minimal parts.

- VFO: VFO is a SI5351 PLL VFO IC which generates CLK0 TX signal, CLK1 RX signal. SI5351 module works with 400khz I2C speed. DX UnO uses a TCXO crystal so no calibration is needed.

- TX PA: TX PA is a non-inverting 74ACT244 octal buffer connected in parallel to drive a Bifilar transformer configured in auto transformer configuration to boost TTL PA output. RF power output - varies from band to band - around 400 milliwatts which is enough for decent digital modes QSOs. DX UnO RF PA is resilient to high SWR and antenna mishaps such as short in antenna and no antenna connected conditions. It is a forgiving RF PA design almost indestructible!

- Low Pass Filter Module is a 10m/28 Mhz 7 pole smd low pass filter.

- DX UnO RECEIVER is a TAYLOE detector – Quadrature sampling detector. There is also a mosfet RF preamplifier stage with diode peak limiting. This improves Receiver gain significantly. The advantage of QSD is I it's narrow bandwidth. This allows DX UnO to operate on 5 bands with one lowpass filter. There is no SSB decoding or any form of Hilbert transform. This receiver scheme is still a direct conversion receiver with a much sensitive and improved narrow band compared to ADX and ADX UnO.

- DX UnO is designed only to operate on digital modes that are using audio tones to communicate. It won't work on phase mode digital modes like PSK31.

- DX UnO is not an SSB or CW rig!

- RF signal output of DX UnO is FSK generated signal which will correspond to USB of an SSB Receiver. It is not a DSB signal and will not occupy both USB and LSB band where it operates.

- DX UnO is a QRPp rig, RF output is around 400 milliwatts average depending on band. As the frequency gets higher RF power output will decrease. For example, 10m band RF output will be around 300 milliwatts. Although this kind of QRPp output power seems

low it is sufficient to have decent continental QSOs in digital modes and especially on WSPR. Please refer to section 6 of this manual.

- Although CAT control of DX UnO limits Transmit to 5 bands, 20m, 17m, 15m ,12m and 10m, receive is not limited with these 5 bands. 30m, 40m and even 80m receiving can be accomplished. TX on 80m, 40m and 30m bands are prohibited due to being outside of FCC regulations envelope.

Schematics and PCB Layouts of DX UnO TRX Design:

Schematics and other related info uploaded to DX UnO Github page not to clutter this build manual.

https://github.com/BluQRP/DX_UnO/tree/main

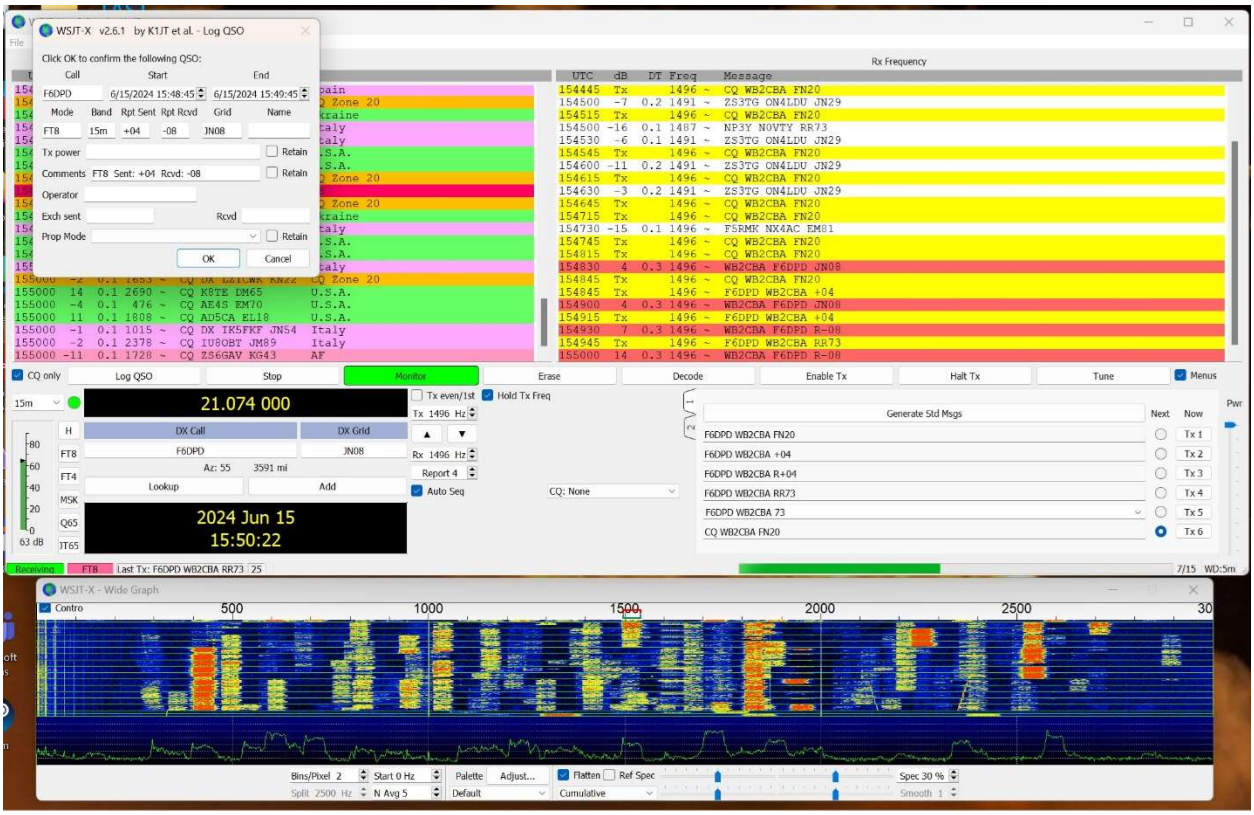
6 How effective is DX UnO for long distance QSOs as a QRPp 400 mW RF Output Transceiver?

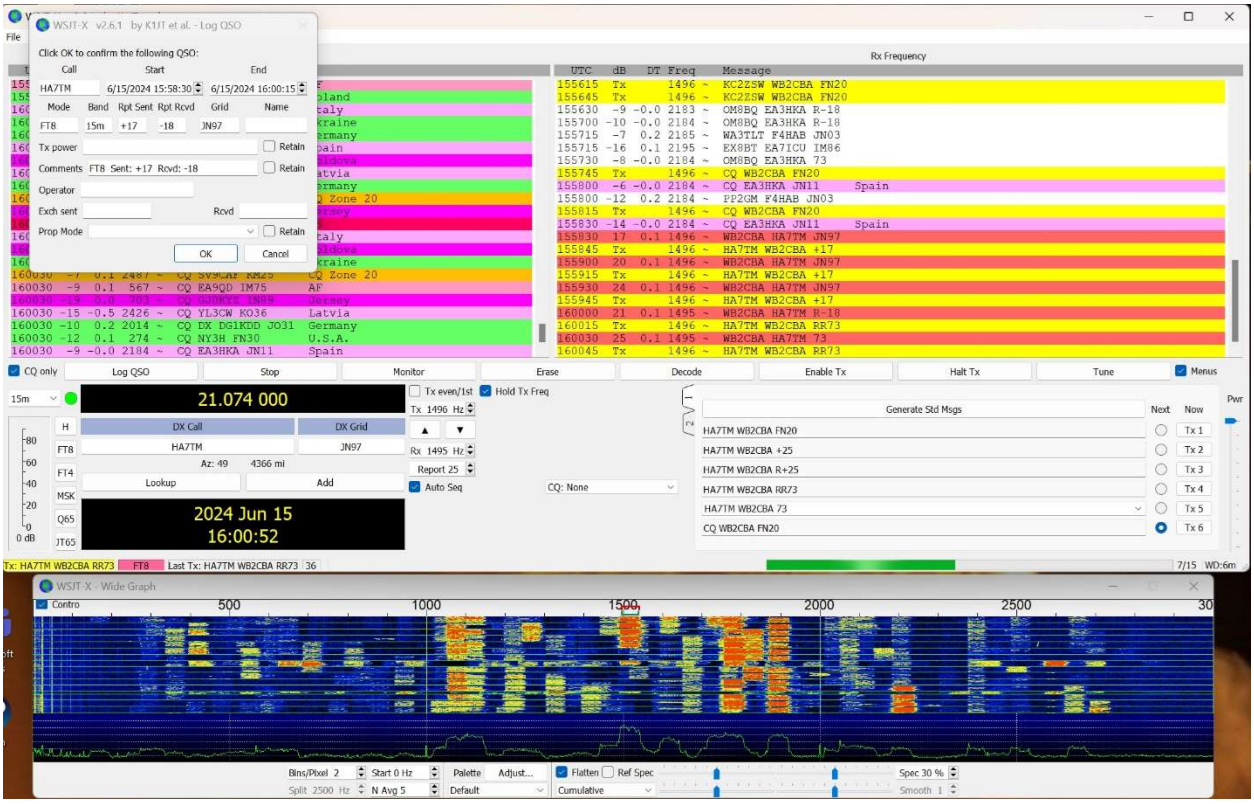
Following is two example QSOs that are carried out with DX UnO.

QTH Location: Liberty State Park, Jersey City, NJ, U.S.A

Antenna : MFJ 10m Hamstick Antenna tuned to 15m.

RF Output Power = 400 mW.





Acknowledgments:

I would like to thank:

- Burkhard Kainka, DK7JD for his inspirational Audio zero cross detection FSK generation code for Arduino. Without it I would be still scratching my head how to solve this problem!
- Jason Mildrum, NT7S for his excellent SI5351 Arduino Library.

Enjoy your Digital QRPp QSOs!

Barbaros Asuroglu aka Barb, WB2CBA

06/15/2024

wb2cba@gmail.com