ADX UnO – Arduino Uno based HF Digital Modes Transceiver

Build and Operation Manual for ADX UnO PCB Version 1.4

Rev 1.0 11/2022

ADX UnO files and all relevant info can be found in ADX UnO Github page link: https://github.com/WB2CBA/ADX-UnO-V1.4

ADX UnO is an Arduino Uno based HF Digital Modes Transceiver. ADX is abbreviation for **A**rduino **D**igital **X**ceiver.

ADX UnO is based on ADX – Arduino based Digital Modes Transceiver design.

Here is the link for ADX – Arduino based digital modes transceiver for further information:

https://antrak.org.tr/blog/adx-arduino-digital-transceiver/

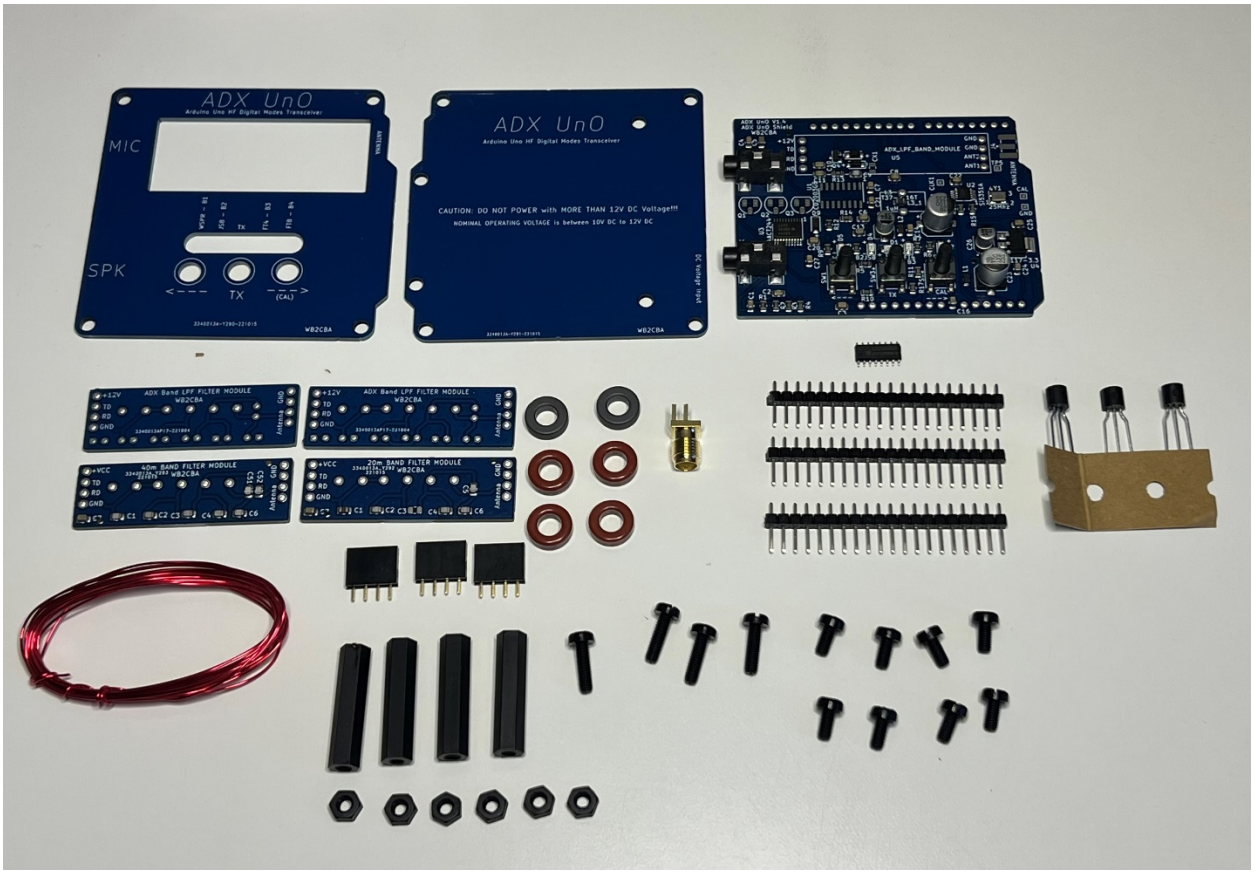
ADX UnO is a mono band digital modes optimized HF transceiver that can be plugged on top of an Arduino Uno board to create a 3.5 watts average RF Output power QRP portable HF Digital Modes capable transceiver. ADX UnO can cover four pre-programmed bands one band at a time by swapping Band LPF Filter Modules. It can operate on any of the pre-programmed four bands of 80m,40m,30m,20m, 17m, 15m and 10m bands and operates on four of the most popular digital modes, FT8, FT4, JS8call and WSPR.

ADX UnO V1.4 Kit contents and BOM(Bill Of Materials):

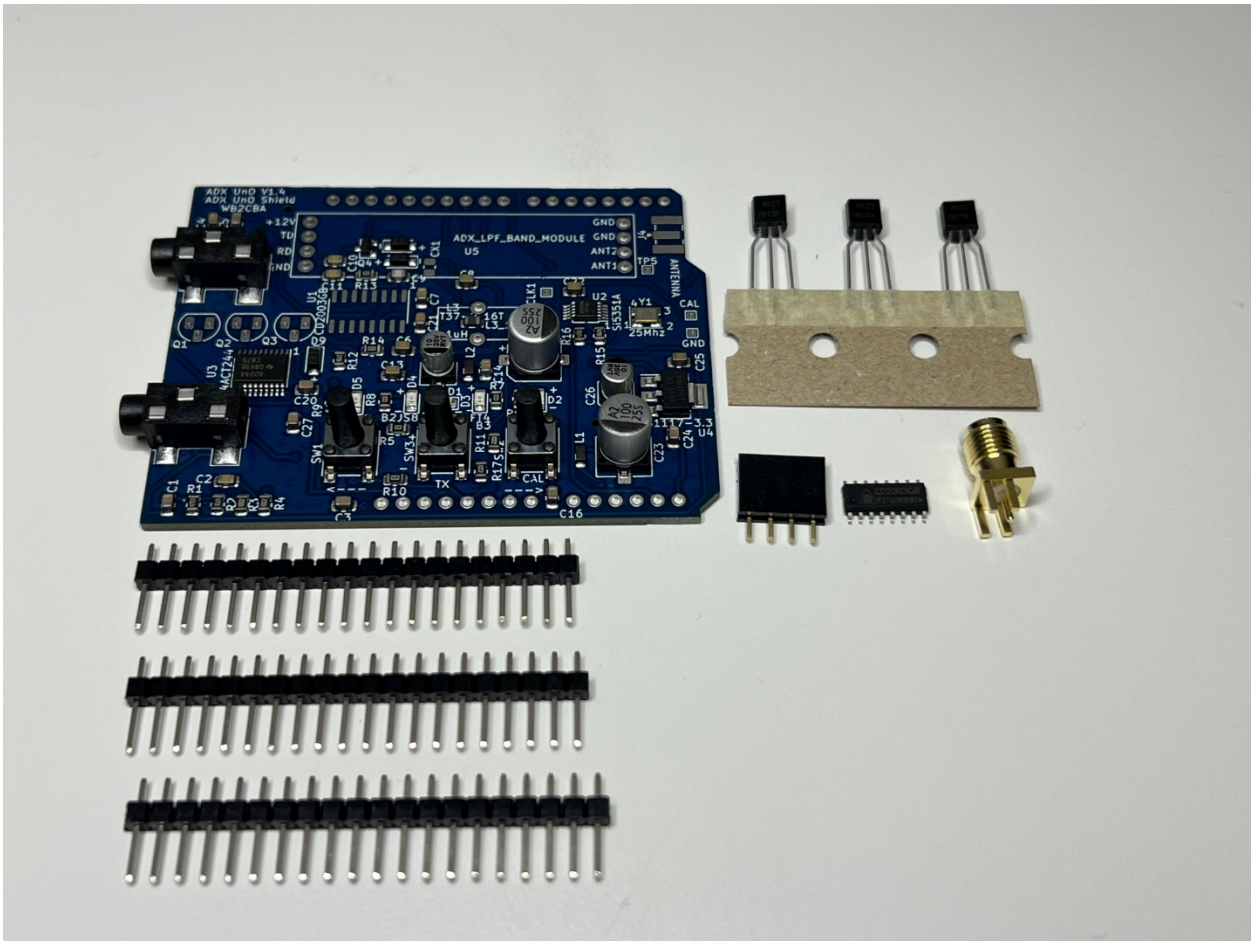
ADX UNO Kit Bill of Materials (parts in the kit)	Each
Populated ADX UNO Board with SMD parts soldered	1
20m LPF module with smd caps soldered	1
40m LPF module smd caps soldered	1
blank LPF module pcb	2
top panel pcb	1
bottom panel pcb	1
CD2003GB receiver IC soldered onto the JLCPCB board	1
BS170 Mosfets	3
SMA pcb edge type RF connector	1
20 pin male header	4
4 pin female header	3
T37-2 red toroid cores (for 2 SMD soldered LPFs)	4
FT37-43 ferrite cores (for 2 SMD soldered LPFs)	2
1.5 m / 0.40 mm (6 feet awg24) enamel copper magnet wire	1
M3-6mm Screw	8
M3-12mm Screw	4
M3-Nuts (used as spacers between Arduino Uno and bottom panel)	6
M3-25mm Female to Female Spacers	4

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

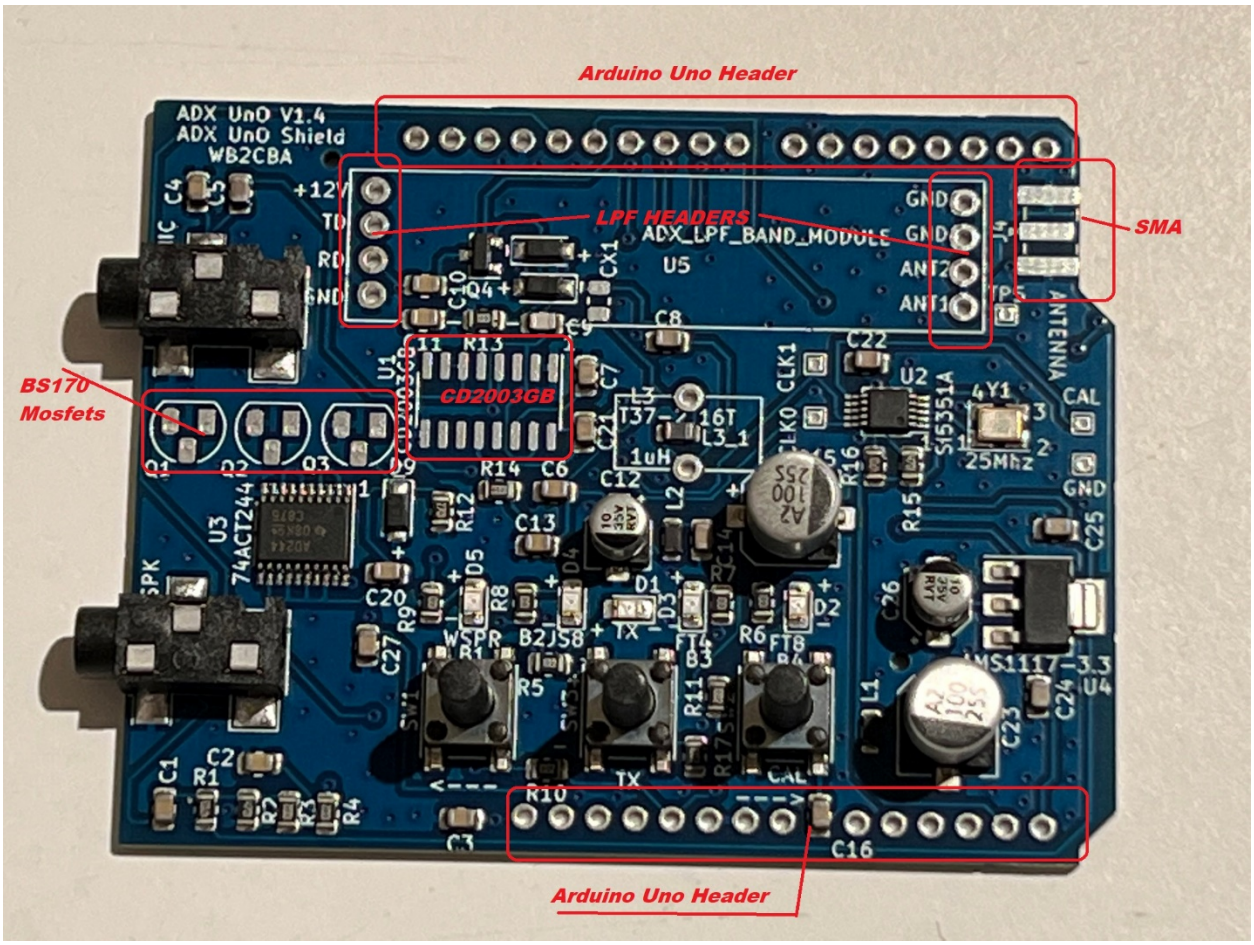
ADX UnO V1.4 Kit parts:



1- Building ADX UnO Main Board:



Parts in above photo are to be soldered on ADX UnO Main Board in order to finish Main Board Build.



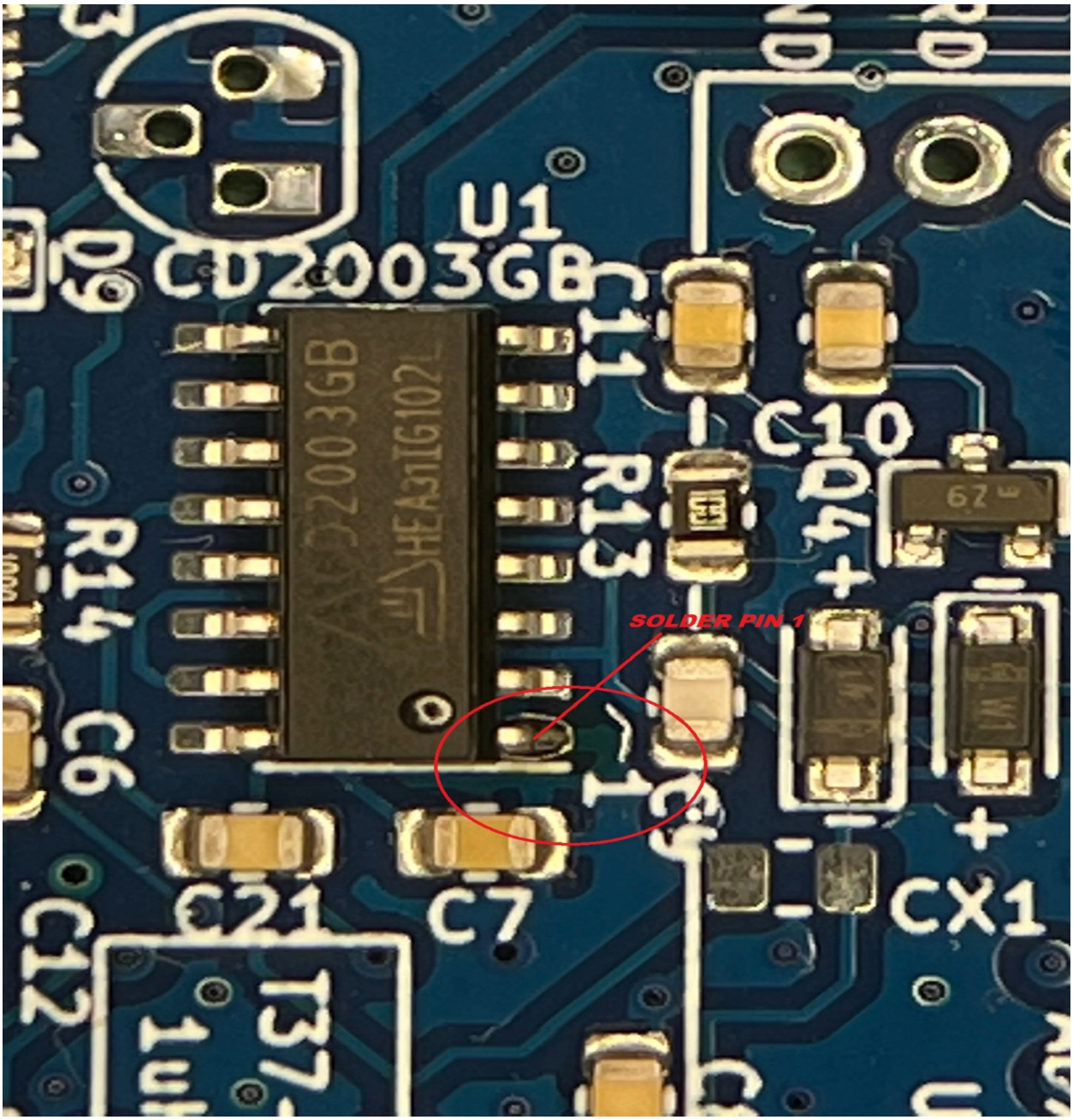
In the above photo all parts and their relative locations on ADX UnO Main board that needs to be soldered are outlined in red.

ADX UnO Main Board Building Steps:

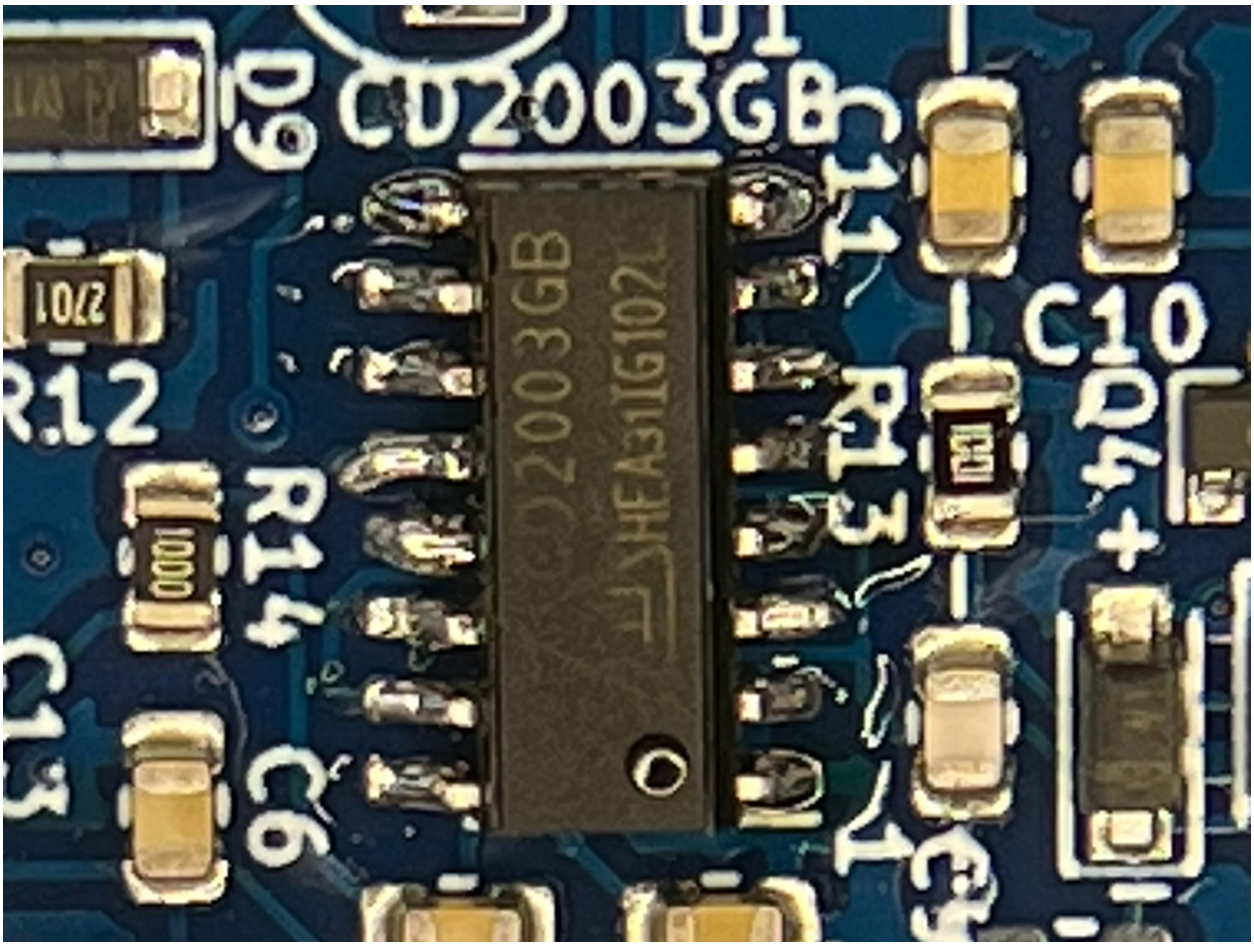
- **CD2003GB Soldering:**

For CD2003GB SMD RX IC soldering on ADX UnO Main Board follow these steps:

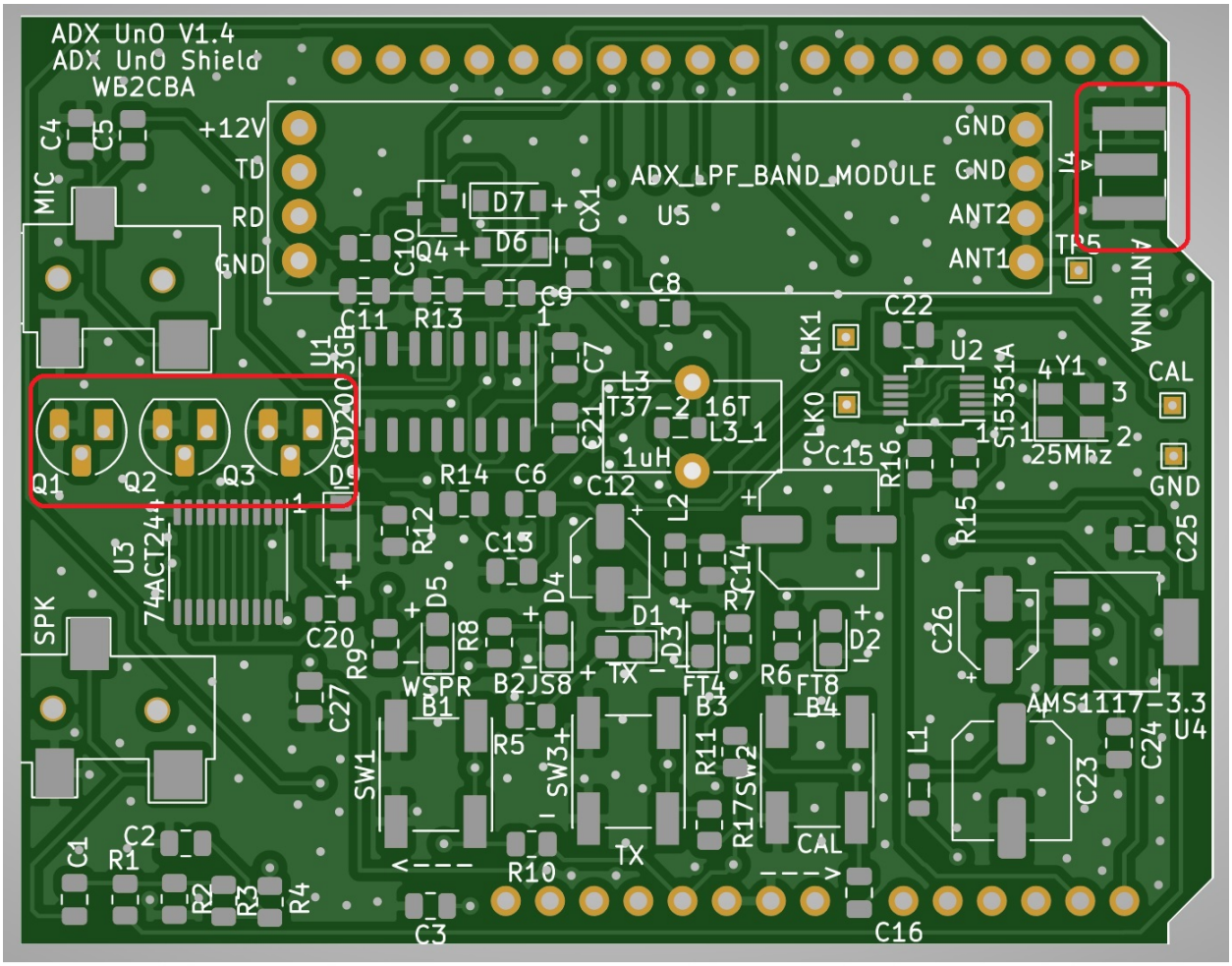
1- Solder pin 1 of CD2003GB with a fine tip soldering iron as shown below:



2- Now solder rest of the pins of CD2003GB. Finished soldering should look like shown below:



STEP 1:

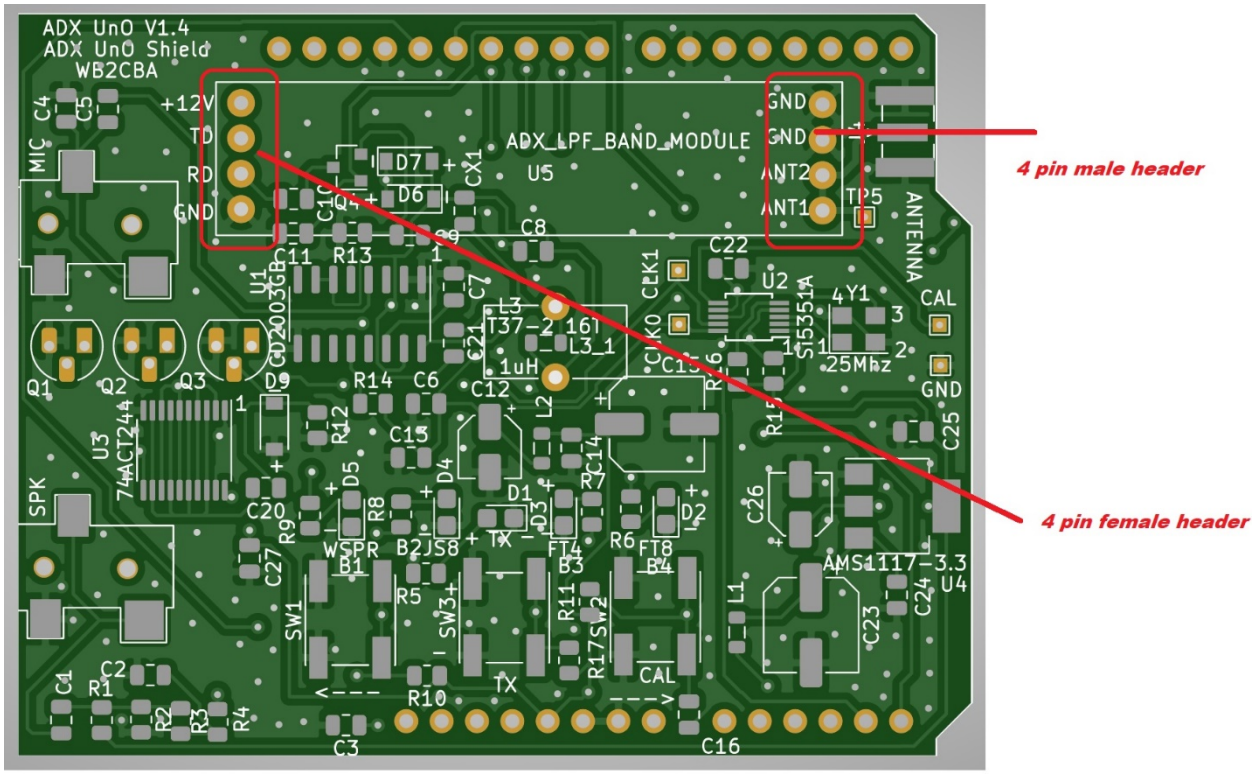


STEP 1 - SOLDER BS170s and SMA Connector

<u>Solder in place these parts. These parts are outlined in red lines:</u>

- 3 x BS170 Mosfets labeled as Q1, Q2, Q3
- Solder J4 SMA Antenna Connector

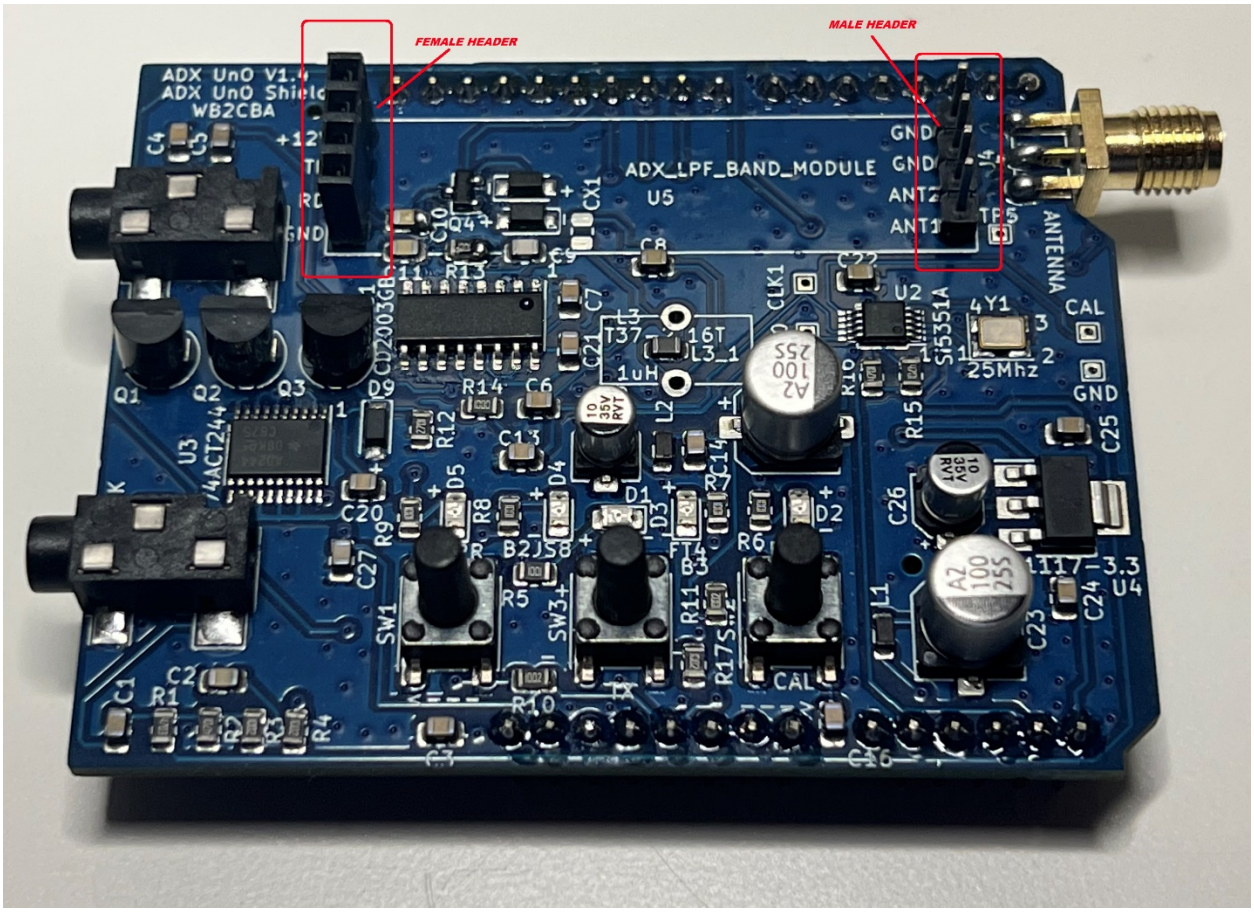
STEP 2:



STEP 2 - SOLDER LPF HEADERS

In step 2 solder male and female 4 pin headers for LPF band Module as shown in the above layout and below photo. Solder female header at mosfet side of the pcb and solder male header on antenna connector side of PCB.

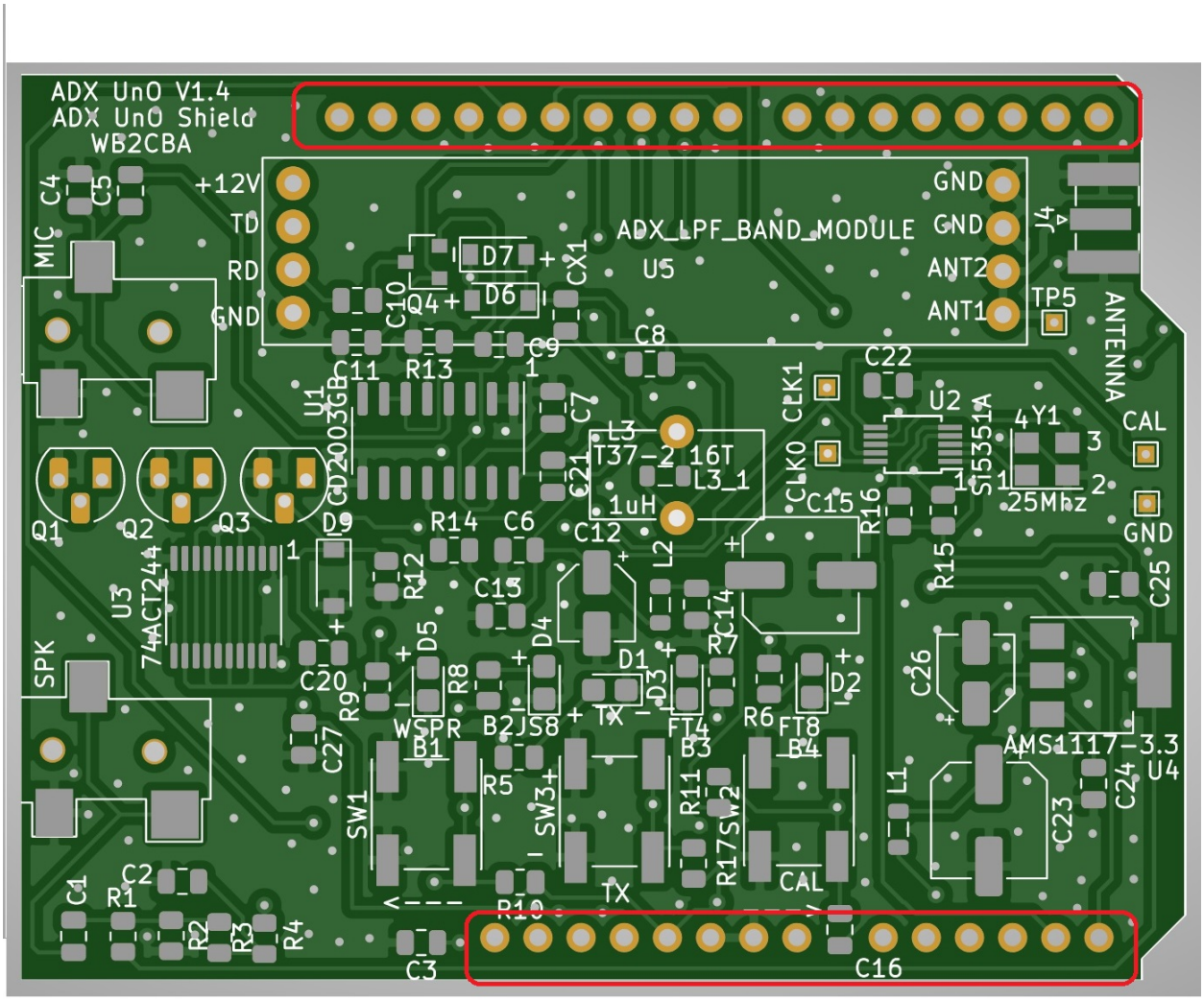
Soldered headers should look like this:



If you have LPF Band Modules from uSDX mOnO or uSDX V1.02 serial resonance modules proejects already build and want to use them then you can arrange LPF Band Module headers to match those LPF Band Modules in hand.

STEP 3:

Step 3 is soldering Arduino Uno Headers. These headers are outlined in red below.



STEP 3 - SOLDER HEADERS for ARDUINO UNO

First cut these headers from Male 20 pin header stacks.

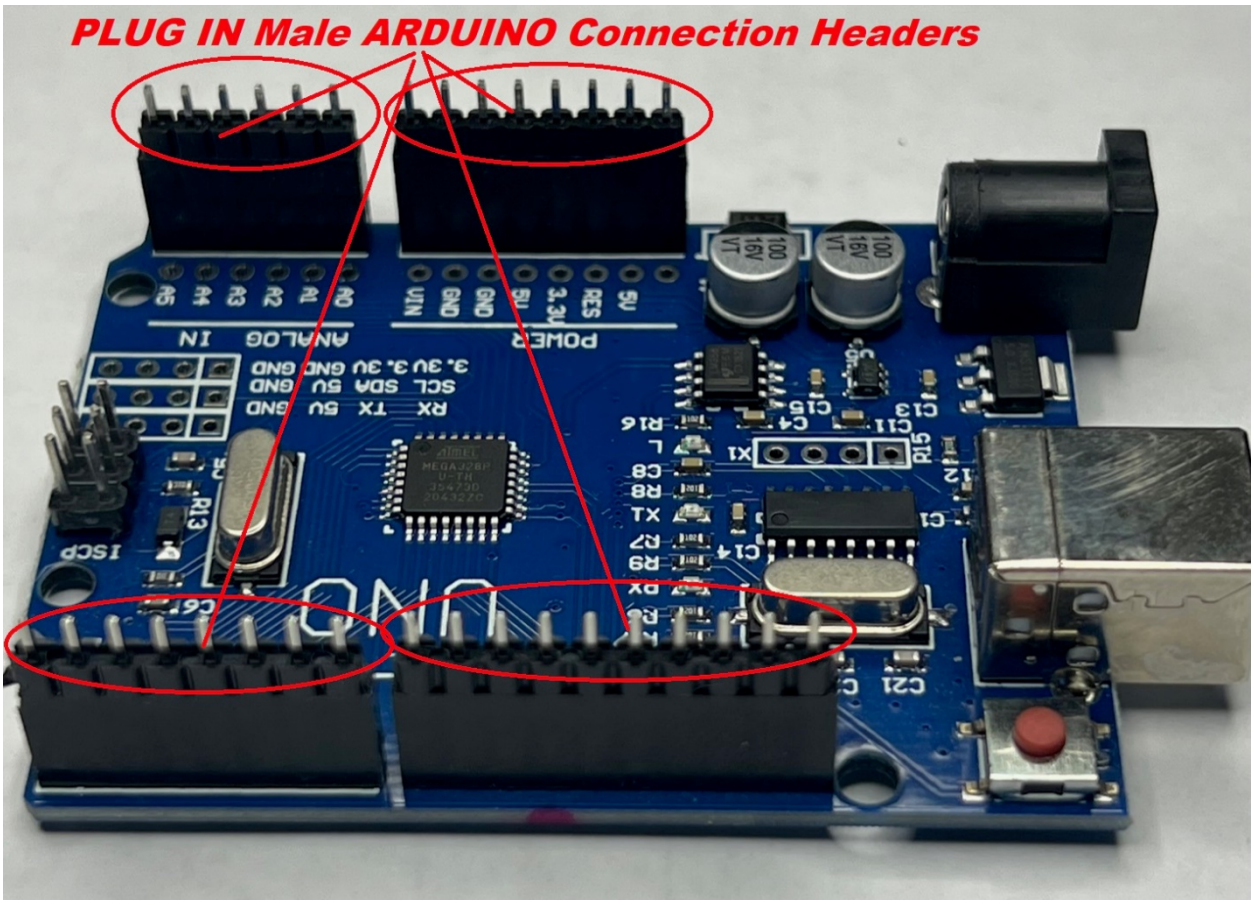
4 male pin headers:

2 x 8 pin male header

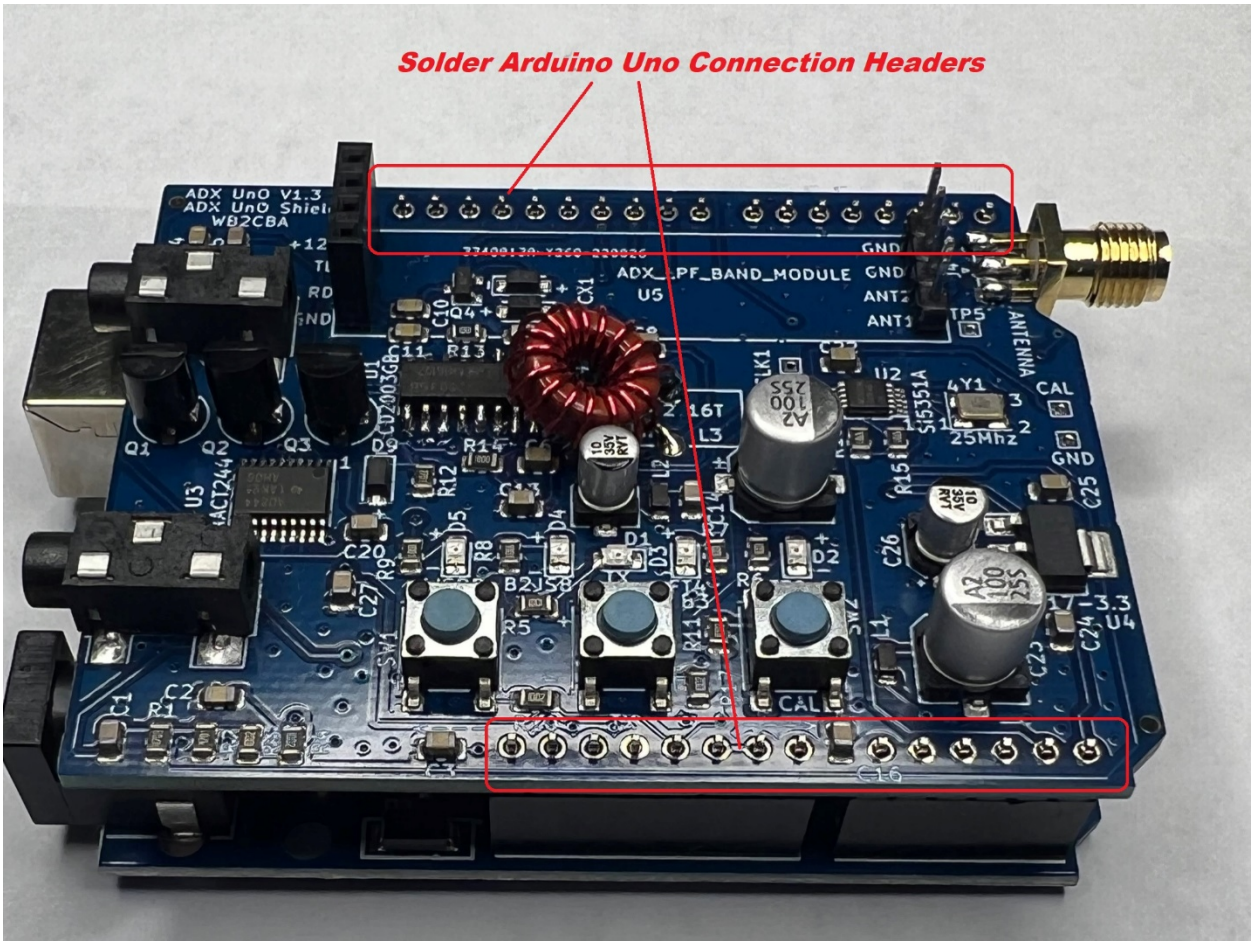
1 x 6 pin male header

1 x 10 pin male header

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

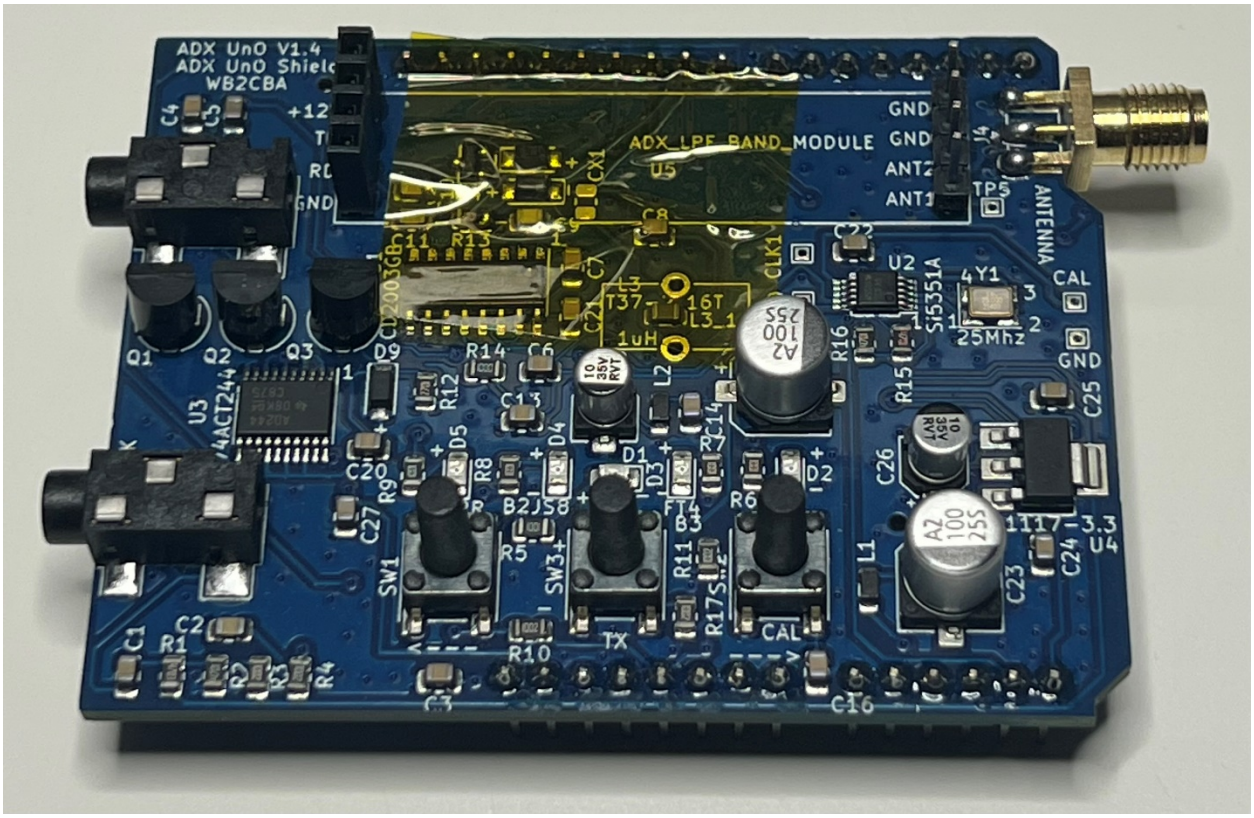


Now insert ADX UnO board on these male headers gently and firmly matching headers and then solder all pins: (L3 in the photo is T37-2 toroid for illustrative purpose. In the actual kit it is fixed smd 1uh inductor)

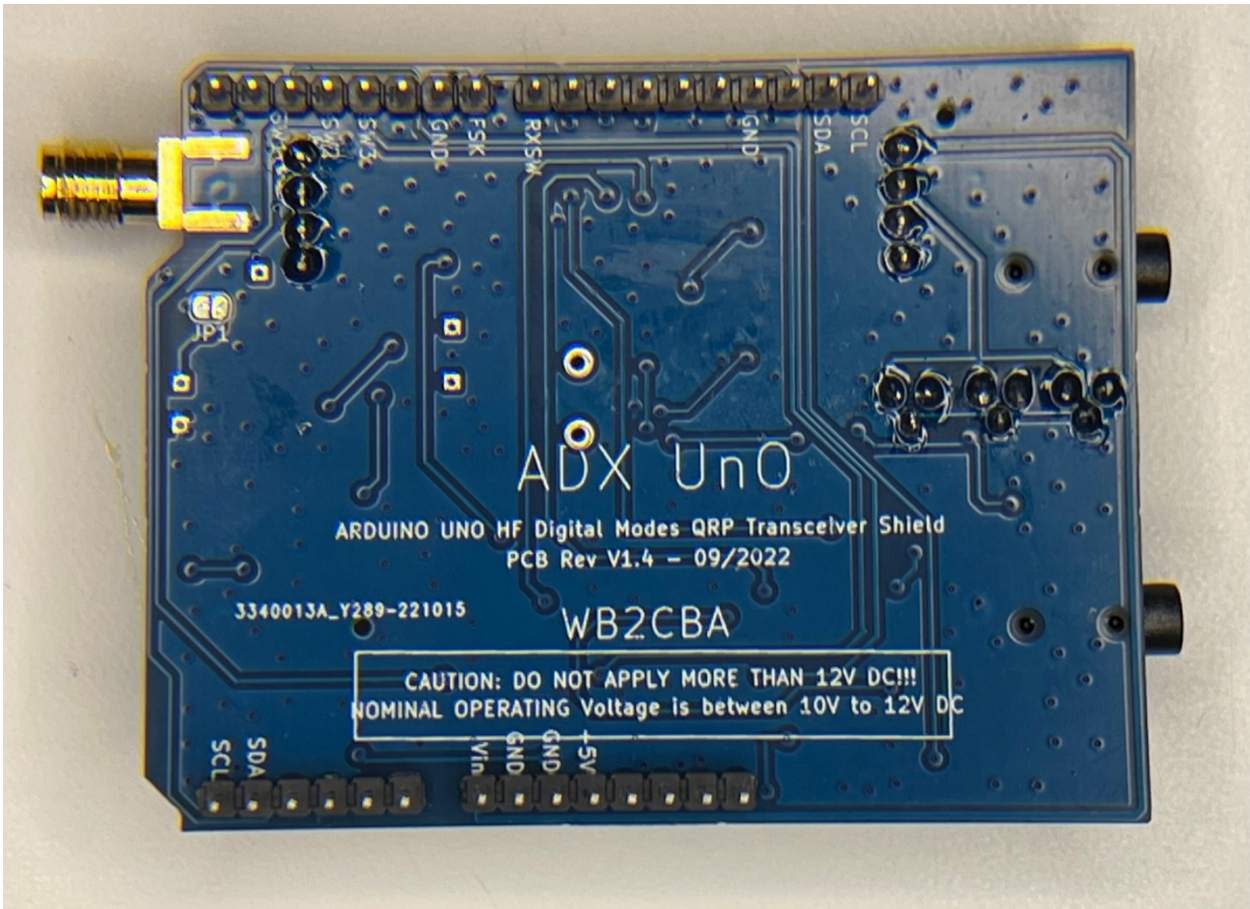


Finished and soldered ADX UnO Main Board should look like this:

Place electrical insulation tape to cover components under LPF band module area to prevent any possible shorts when LPF Band Module is plugged in as shown in the photo below.



And this is bottom view of completed ADX UnO main board showing Arduino Uno male headers:



2- Building ADX UnO SMD and THD LPF Band Modules:

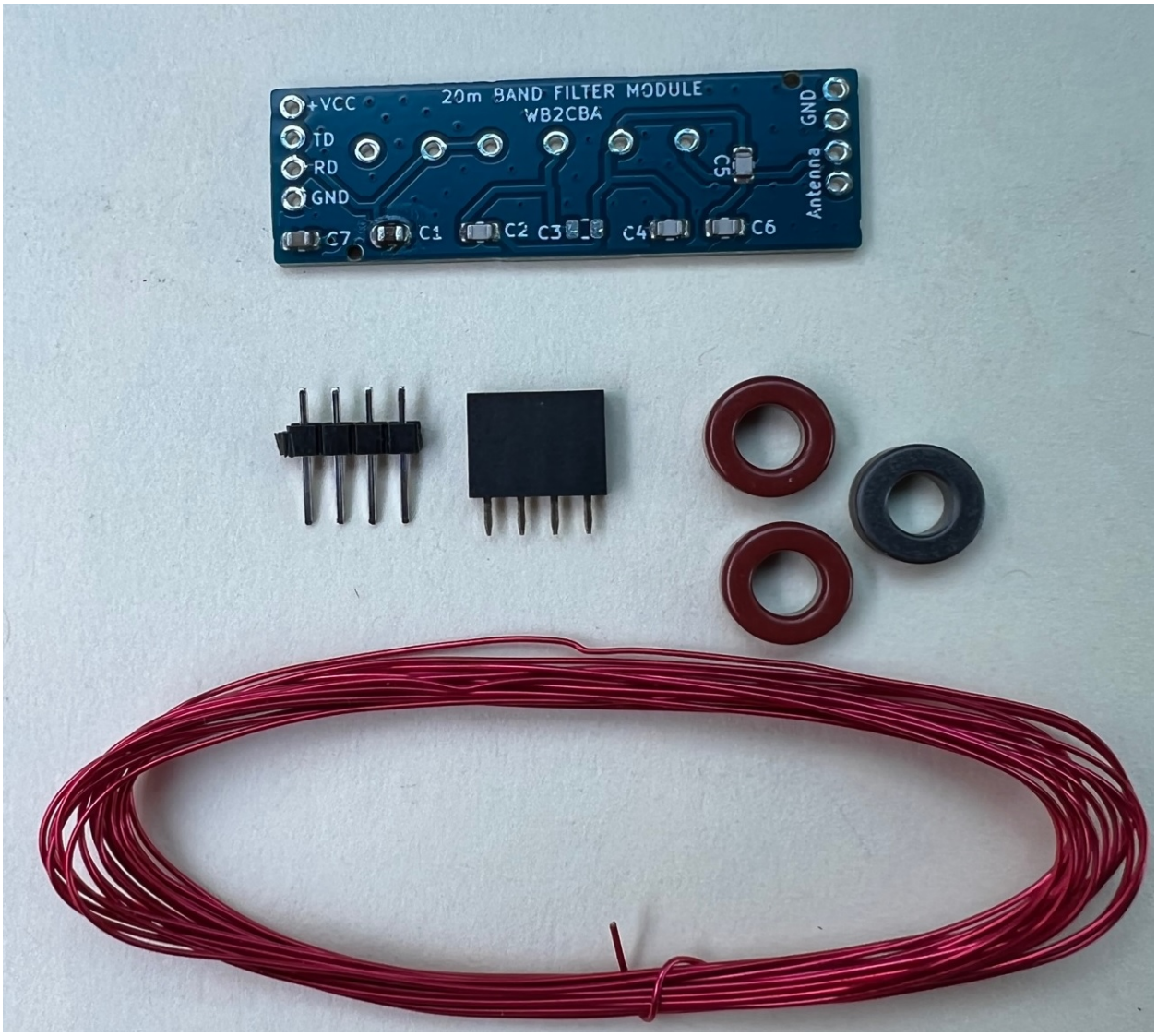
ADX UnO Kit comes with one 20m/14Mhz smd populated LPF Band Module with associated parts and one 40m/7Mhz smd populated LPF Band Module with associated parts. We will call these LPF Band Modules as **SMD LPF Band Modules**, SMD standing for Surface Mount Device.

Also included in the kit two blank LPF Band Module pcbs for the kit builder to build his/her own LPF Band Modules for any band desired with own supplied parts. We will call these LPF band Modules as **THD LPF Band Modules**, THD standing for Through Hole Device.

- **<u>ADX UnO SMD LPF BAND MODULE BUILD:</u>**

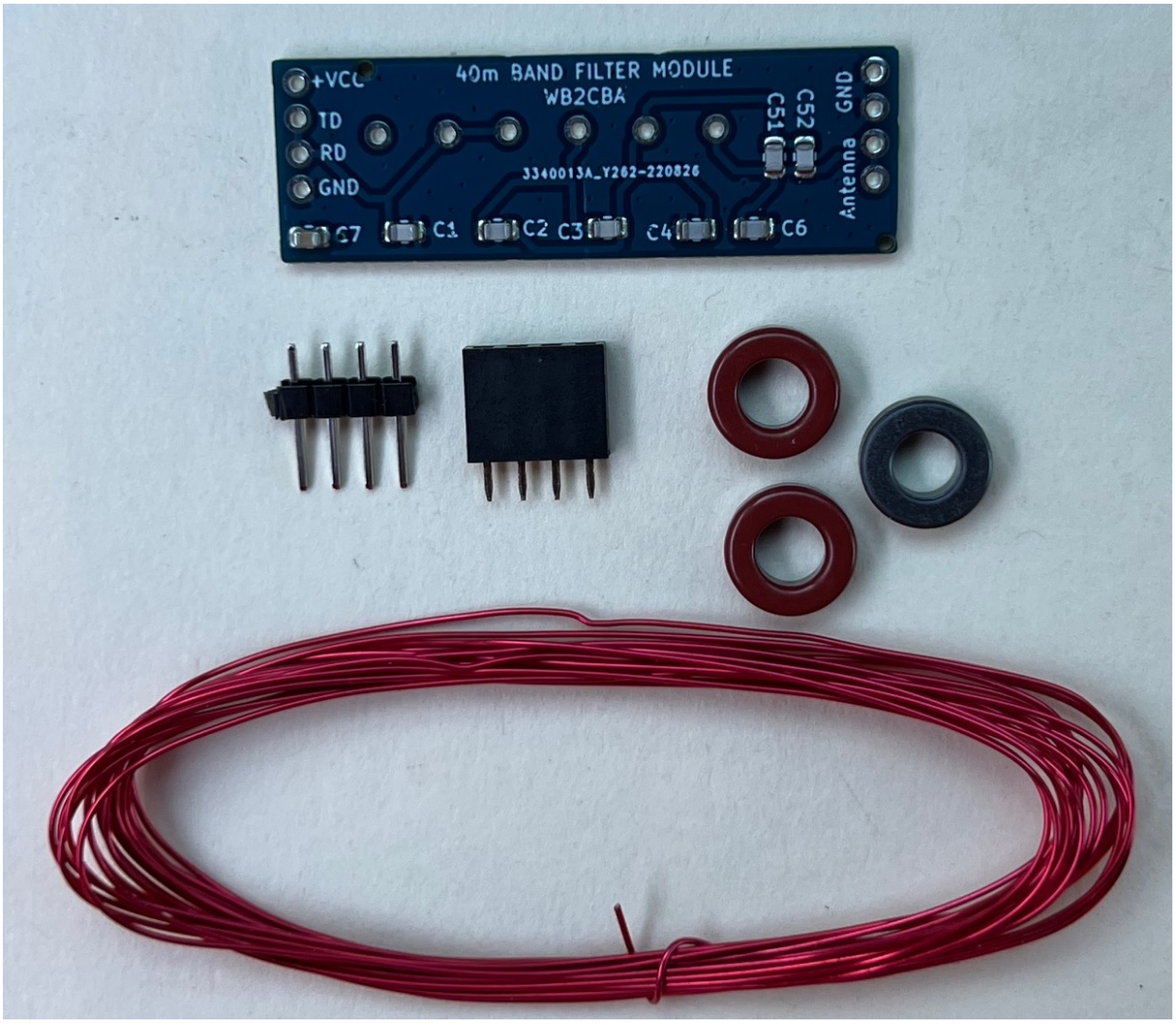
20m/14Mhz LPF BAND MODULE Parts:

- **1 x SMD populated LPF Band Module PCB with 20m Band FILTER MODULE (Designation on the PCB)**
- 1 x 4 pin male header to be cut from 40 pin male header stack
- 1 x 4 pin female header to be cut from 40 pin female header stack
- 2 x T37-2 RED Toroid core
- 1 x FT37-43 BLACK Toroid ferrite core
- 24 AWG enameled magnet wire (Adequate amount of wire to be cut and used to wind toroid coils from supplied 8 feet of AWG24 wire)



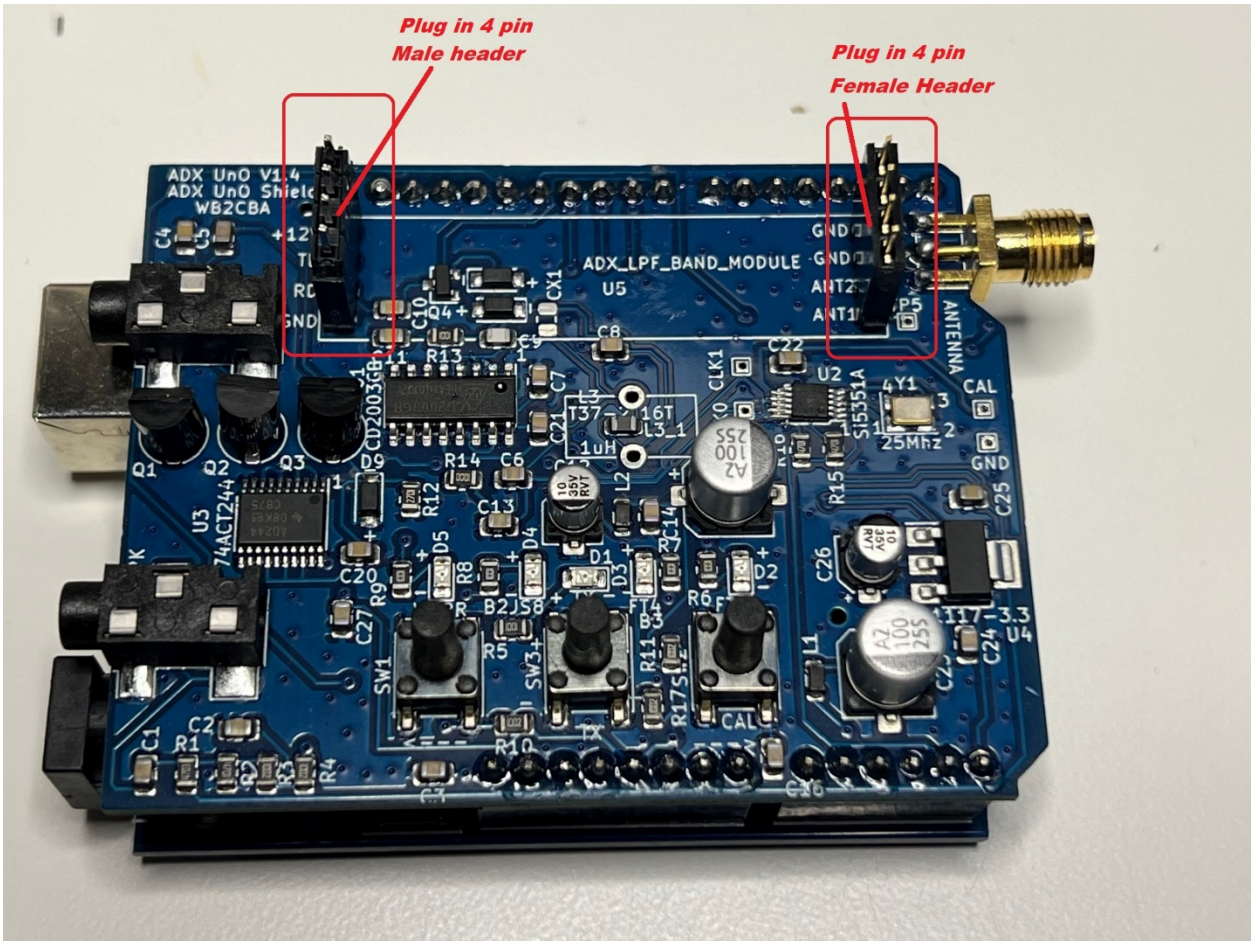
40m/7Mhz LPF BAND MODULE Parts:

- **1 x SMD populated LPF Band Module PCB with 40m Band FILTER MODULE (Designation on the PCB)**
- 1 x 4 pin male header to be cut from 40 pin male header stack
- 1 x 4 pin female header to be cut from 40 pin female header stack
- 2 x T37-2 RED Toroid core
- 1 x FT37-43 BLACK Toroid ferrite core
- 24 AWG enameled magnet wire (Adequate amount of wire to be cut and used to wind toroid coils from supplied 8 feet of AWG24 wire)

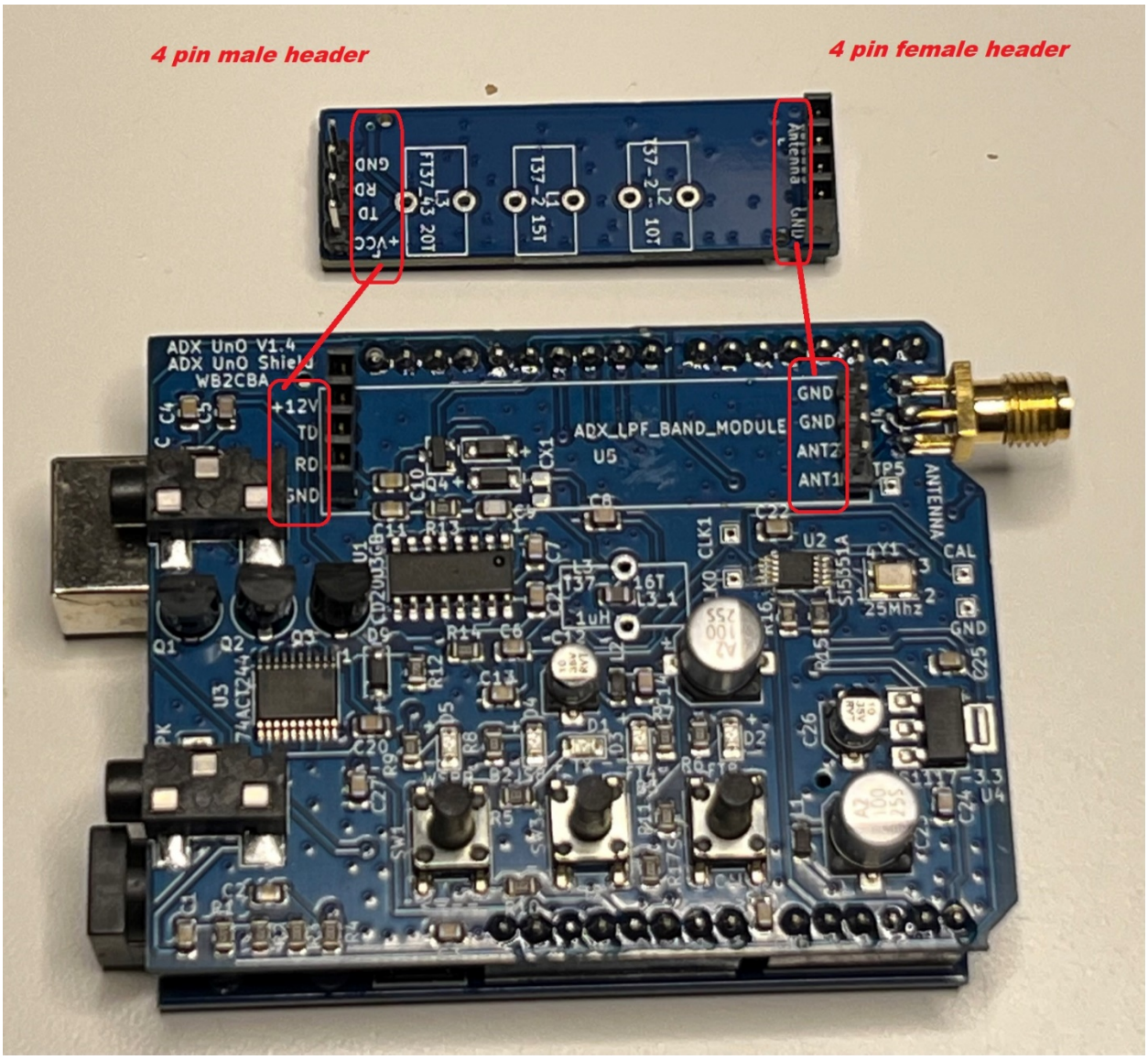


<u>STEP 1: Soldering 4 pin male and female headers:</u>

First thing is to solder 4 pin male and female headers in their right location to match main board 4 pin headers on both 20m and 40m SMD LPF Band Module:

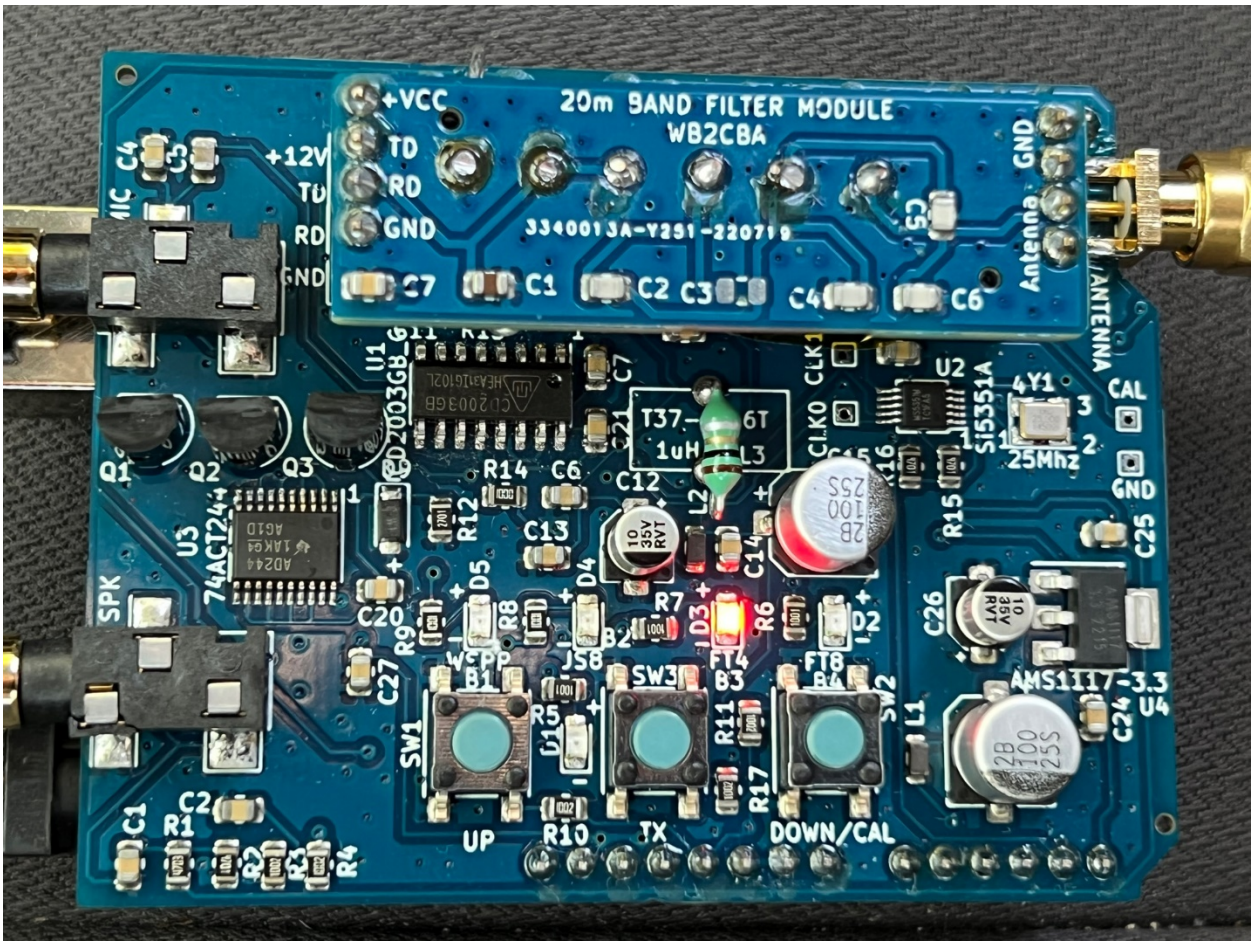


First step is to insert a 4 pin male header to ADX UnO main Board LPF band module female header and then insert a 4 pin female header to ADX UnO Main board LPF Band module male header as shown in the above photo. This will help to hold the LPF Band Module in place in exact position to solder easily.



In STEP 2 orient LPF Band Module as shown in the photo such that Antenna and GND pins coincides with GND and ANT1 and ANT2 pins. Also +12V,RD,TD and GND should be on the same side for bot ADX UnO main board and LPF Band Module as shown in above photo.

Now insert LPF Band Module to coinciding with this position and solder headers.



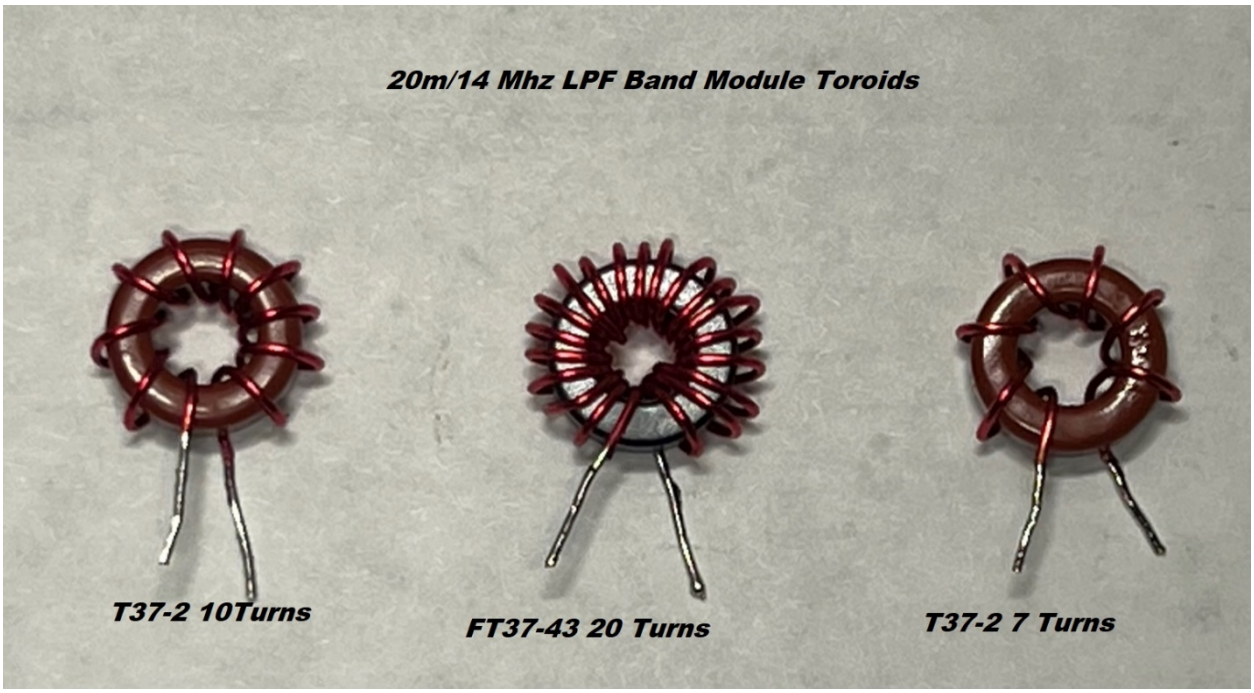
When the LPF Band Module is plugged in it should look like in the above photo.

Repeat these steps for both 20m and 40m LPF band Modules. These header soldering steps also applies to THD LPF Band Modules.

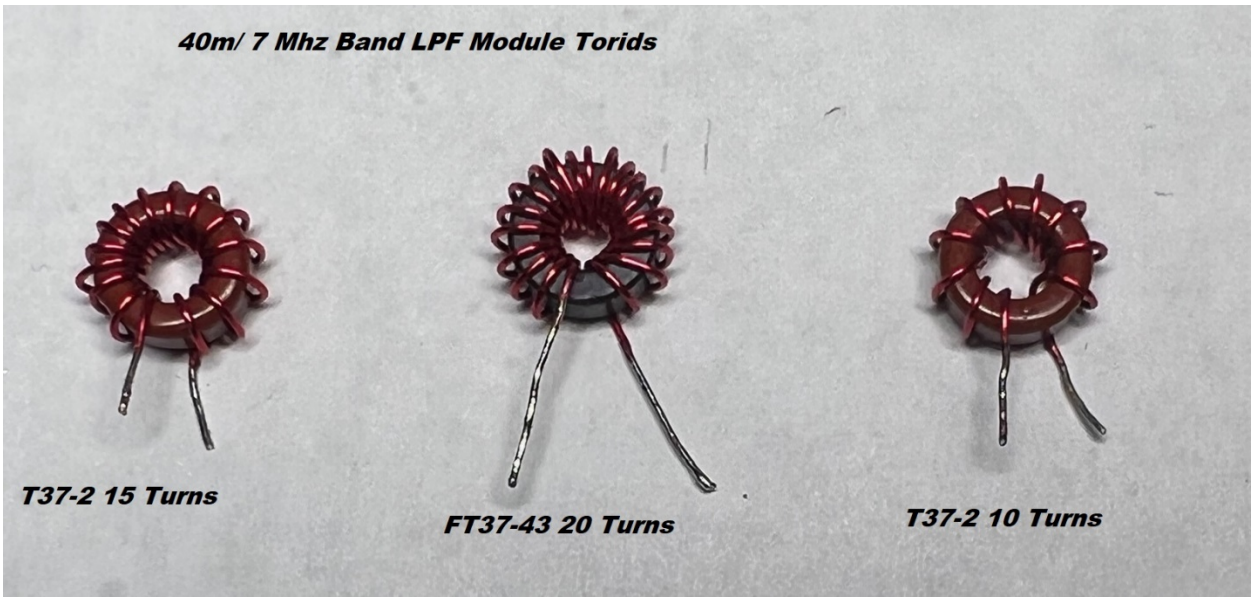
The reason of using alternating female and male 4 pin headers is to prevent any LPF Band Module plug in errors as both sides of LPF Band Module have 4 pin headers.

STEP 2: LPF Band Modules Toroid Build:

SMD 20m LPF Band Module Toroid winding:



SMD 40m LPF Band Module Toroid winding:



Each LPF Band Module has 3 Toroid inductors.

- FT37-43 Toroid has same turns in all LPF Band Modules, 20 Turns from 24 AWG enamel coated magnet wire which is included in the kit.

- 20m LPF Band Module has two T37-2 RED colored Toroids. L1 is 10 Turns and L2 is 7 Turns. Turns and type of Toroid is indicated on the LPF Band Module PCB. These are also wound with included 24 AWG enameled copper wire.

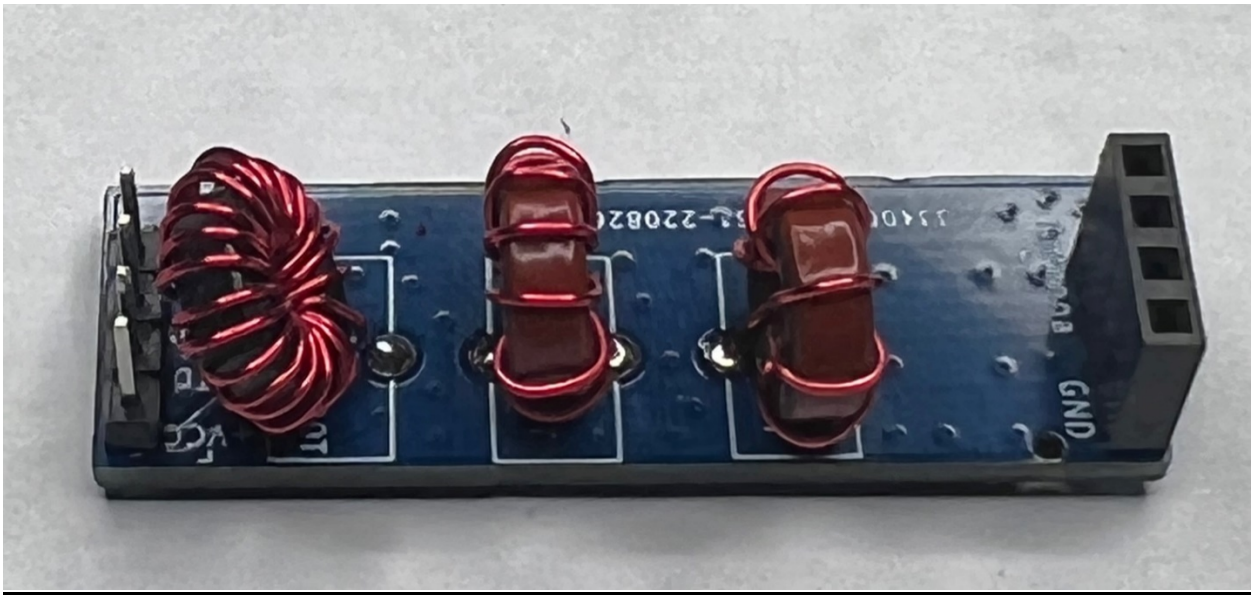
- 40m LPF Band Module has two T37-2 RED colored Toroids. L1 is 15 Turns and L2 is 10 Turns. Turns and type of Toroid is indicated on the LPF Band Module PCB. These are also wound with included 24 AWG enameled copper wire.

TIPS on TOROID Winding:

One important point to consider is that the edges of FT37-43 Toroid are quite sharp which can scrape enamel coating on inductor wire. To fix this issue the easiest method is tapering the edge a little with an oversized drill bit. Use the oversized drill bit to turn with your fingers inside the toroid ring to smoothen out sharp edges. Do not use a drill to turn the drill bit! You will snap the toroid and most importantly will harm your fingers in the process!

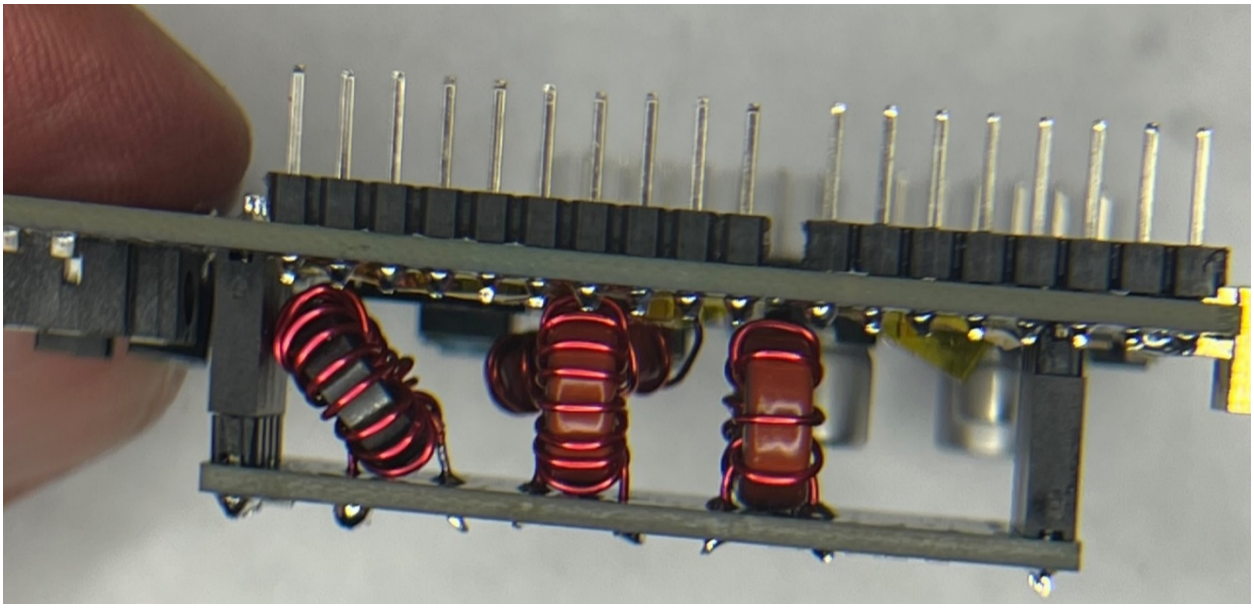
1- One turn means enameled copper wire passes once inside the toroid ring. For example, in 7 turns T37-2 toroid in the above photo wire passes 7 times inside the ring.
2- Always after passing the turn stretch the wire so it is tightly wound around the toroid.
3- When all turns are in count carefully every turn that passes through the inner ring of toroid. The best method to do is to take a close up photo of the toroid and count the turns. This way it is easier to see turns and you won't miss a turn.
4- Never ever let one turn go over another one. Every turn should be side by side! This is important in FT37-43 toroid that has 20 turns and it is easy to overlap turns.
5- Always scrape ends of enamel wire to clean enamel coating for soldering. To clean enamel coating there are two methods which can be used separately or together.
Scraping enamel coating with a cutter knife thoroughly and then apply a bit solder coating. Second method is to heat up wire with soldering iron so that enamel coating just burns out and then cover with a solder coating. Scraping first and then burning the residue enamel coating is the best approach! Then apply a bit solder coating before soldering toroid into pcb.
6- Before soldering wound toroid try to spread as even as possible as in the photos above.
7- After soldering toroid in its place check with a continuity tester if solder joints are okay and connected electrically. This is the most common failure point in kits.

Now solder each toroid to corresponding location paying attention not to solder one in a wrong location. Toroid types and number of turns are indicated on PCB silk screen to make matching easy.



When soldering Toroids in place bend FT37-43 20 turn toroid towards the male header in as in the photo and then solder in place. This to prevent Ft37-43 toroid windings touching smd parts on main board when LPF Band Module is plugged in. This applies both to SMD type and THD type LPF Band Modules.

Here is another photo of LPF Band Module with bent toroid plugged into main board showing clearance:



- **ADX UnO THD LPF BAND MODULE BUILD:**

THD LPF Band filter modules can be built either using surface mount device (SMD)capacitors or through hole device(THD) capacitors.

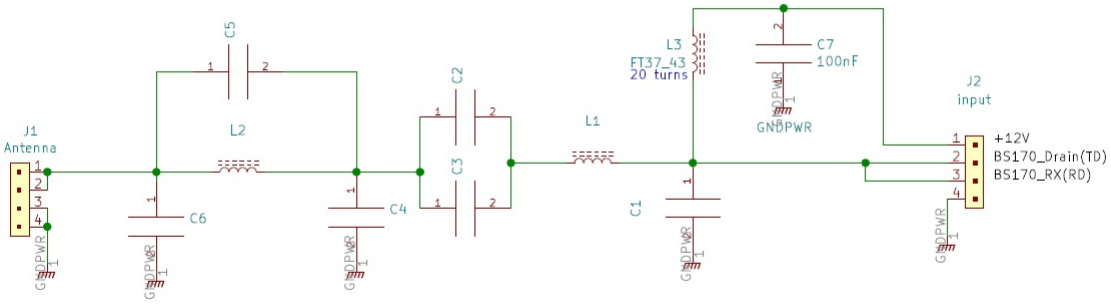
Capacitor footprint on THD LPF Band Module pcb allows to use 0805 or 1206 footprint standard SMD(Surface Mount Device) capacitors or 2.54 mm lead spacing THD(through hole) capacitors.

Toroids used are:

1x FT37-43

2 x T37-2 Red toroid or T37-6 Yellow toroid depending on band selection for each band module. Up to 20m/14Mhz band T37-2 toroids work pretty good. From 17m/18Mhz band to 10m/28Mhz band T37-6 yellow toroids are used.

Schematic of Band LPF Module:



BOM for THD LPF Band Module:

2 x 4 pin header – These headers are male and female headers as explained in detail SMD LPF Band Module build. These band module headers should be opposite mate of RF board 4 pin headers.

1 x C1 – Band Dependent THD or 0805 NPO 100V or more Capacitor

1 x C2 – Band Dependent THD or 0805 NPO 100V or more Capacitor

1 x C3 – Band Dependent THD or 0805 NPO 100V or more Capacitor

1 x C4 – Band Dependent THD or 0805 NPO 100V or more Capacitor

1 x C5 – Band Dependent THD or 0805 NPO 100V or more Capacitor

1 x C6 – Band Dependent THD or 0805 NPO 100V or more Capacitor

1 x C7 – 100 nF THD or 0805 Capacitor (Non band dependent capacitor)

1 x L1 – T37-2 RED or T37-6 YELLOW toroid - Band dependent

1 x L2 – T37-2 RED or T37-6 YELLOW toroid - Band dependent

1 x L3 – FT37-43 Ferrite Toroid – 20 turns. Non band dependent and same detail applies to all band LPF Modules.

AWG 24 or 0.40 mm enameled copper wire can be used for Toroid turns.

This table will help choosing all band dependent component values and details for each band

ADX UnO Serial Resonance LPF Band Module" Band dependent components list

Band	C1	C2	C3	C4	C5	C6	L1	L2	L3
80	660pf	1000pf	1000pf	2000 pf	620pf	2000pf	19Turn/T37-2	14 Turn/T37-2	20Turn/FT37-43
40	130pf	470pf	470pf	1000pf	300pf	1000pf	15Turn/T37-2	10Turn/T37-2	20Turn/FT37-43
30	100pf	330pf	330pf	660pf	220pf	660pf	12Turn/T37-2	8Turn/T37-2	20Turn/FT37-43
20	68pf	470pf		470pf	150pf	470pf	10Turn/T37-2	7Turn/T37-2	20Turn/FT37-43
17	91pF	180pF	180pF	360pF	120pF	360pF	11Turn/T37-6	7Turn/T37-6	20Turn/FT37-43
15	68pF	300pF	15pF	300pf	100pF	300pF	11Turn/T37-6	7Turn/T37-6	20Turn/FT37-43
10	39pF	240pF		240pF	75pF	240pF	9Turn/T37-6	6Turn/T37-6	20Turn/FT37-43

This table is a base for building various band LPF modules. If you have a knack to experiment and need a bit more power here are some useful tips:

- Lower one turn from L1 in each band. This will slightly increase RF output power though might increase TX current too so exercise caution!
- For L3 12 turns also works pretty good so this can be a nice option to play with.

Though caution here! DO NOT EXCEED 5 WATTS RF POWER! Otherwise BS170 voltage limit will be exceeded which will end up in magic smoke!

CLASS E RF POWER AMPLIFIER BASICS and Some useful tinkering knowledge:

ADX UnO Transceiver uses Class E type RF power amplifier scheme. Class E is an efficient class of power amplifier which can go up to 85 to 90 % efficiency which means less thermal heat dissipation on RF power transistors and lower supply current vs more RF power output. These are the advantages of Class E RF amplifiers. Disadvantage of this amplifier type is it is harder to tune and get the required efficiency when compared to other classes like Class A,B or C amplifiers. ADX UnO uses this efficient class E amplifier like uSDX transceiver. 3 x BS170 mosfet transistors parallel when tuned properly can easily give up to 5 watts RF output power without dissipating any excess heat under 12V DC.

ADX UnO LPF band filter board consists of all parts related to fine tune ADX UnO Class E mosfet Power amplifier. Above table has component values of each LPF Band Filter which will give a reasonable power efficiency on all 6 bands covered. Efficiency values for each band can be further tweaked and it is a great way to learn and experiment with class E amplifiers. ADX UnOuses so called serial resonance LPF design for Class E and all values are calculated by WA0ITP's excellent excel Class E design spreadsheet: http://www.wa0itp.com/class%20e%20design.html

Here are some class E readouts for curious ADX kit Builders:

http://www.norcalqrp.org/files/Class_E_Amplifiers.pdf

https://people.physics.anu.edu.au/~dxt103/class-e

3- Programming Arduino Uno with ADX UnO FIRMWARE:

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

ADX UnO uses **Arduino Uno** board as microcontroller. This is a readily available and cheap Arduino development board that can be outsourced easily online.

ADX_UnO_V1.1.ino is the ADX UnO firmware file and can be downloaded from ADX UnO github page:

https://github.com/WB2CBA/ADX-UnO-V1.4 under firmware directory.

There is also **a ADX_UnO_V1.1.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 ADX 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.

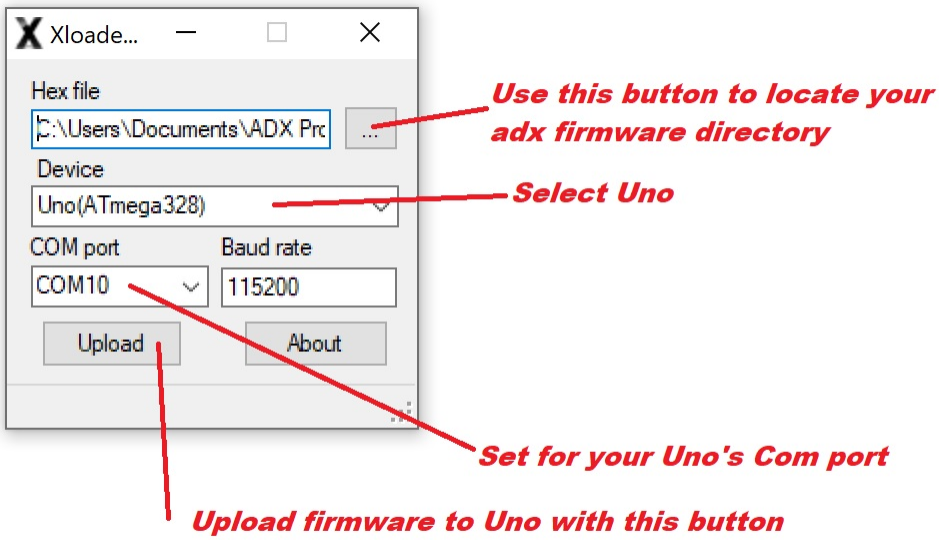
STEPS for Uploading ADX UnO Firmware to Arduino Uno via XLOADER Application:

1- Download Xloader.zip file from ADX UnO Github Link and unzip:
https://github.com/WB2CBA/ADX-UnO-V1.4 under firmware directory.

2- Download **ADX_UnO_V1.1.hex** firmware file from ADX UnO github page under ADX UnO Firmware:
https://github.com/WB2CBA/ADX-UnO-V1.4 under firmware directory.

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

4- Run XLOADER. Below screenshot will be seen:



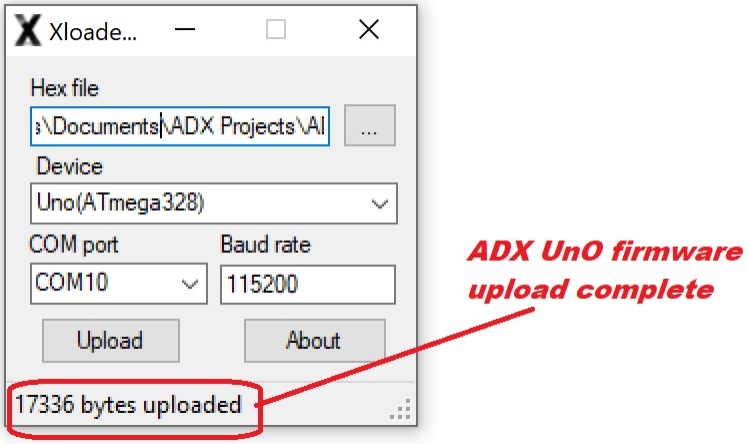
Now locate your ADX_UnO_V1.1.hex file previously downloaded and saved using the file search button of Xloader.

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 ADX UnO firmware to Arduino uno.

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



ADX_UnO_V1.1.hex file is configured for bands 40m, 30m, 20m and 17m as default bands. If band configuration changes are desired for different bands then please use Arduino IDE to do so and compile and upload via Arduino IDE.

Method 2: Arduino IDE Method or Classic method

In this method we will use Arduino IDE to compile and upload **ADX_UnO_V1.1.ino** firmware

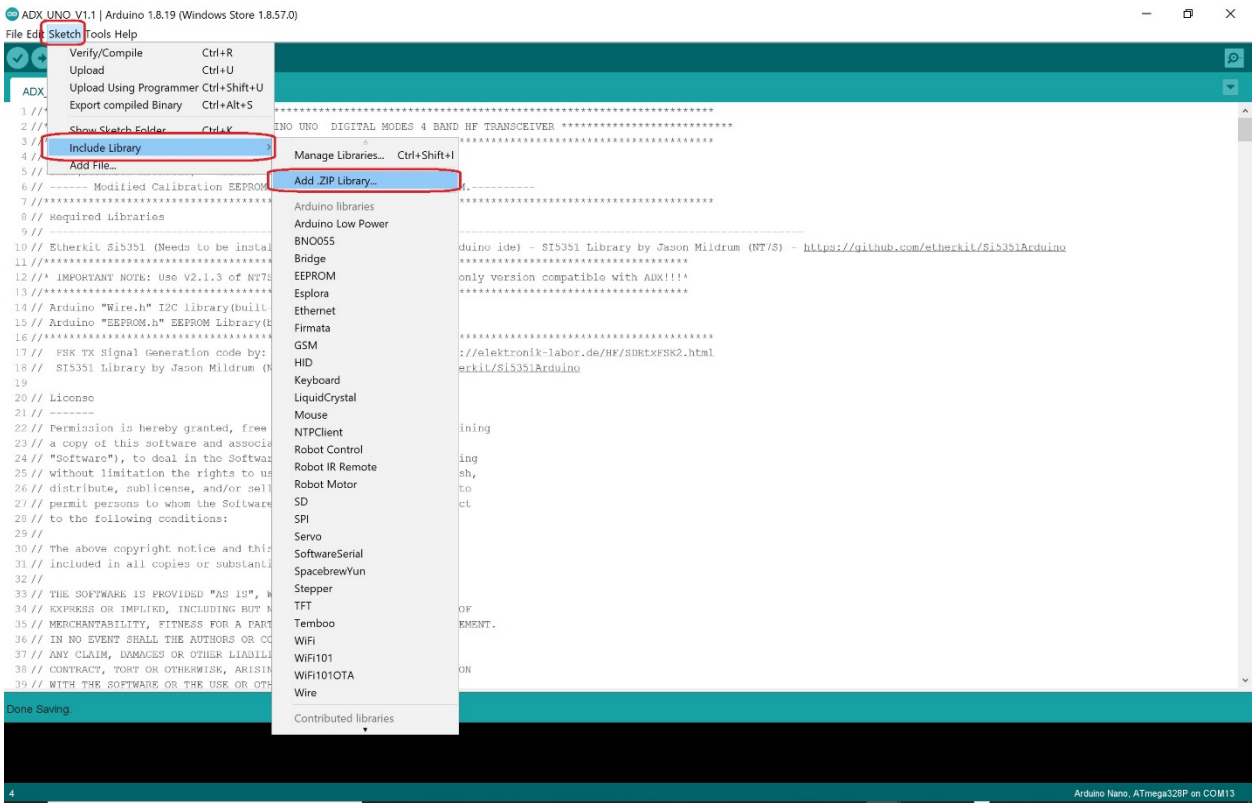
First step is to download and install Arduino IDE from www.arduino.cc

ADX 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 ADX firmware. Use SI5351 Library that is in ADX UnO Github firmware page to avoid any I2C communication problems. ADX 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 ADX Uno github page under ADX Uno firmware directory.
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 **ADX_UnO_V1.1.ino** firmware from ADX UnO Github Page under ADX UnO Firmware directory.

Use " File/Open" menu to open ADX_UnO_V1.1.ino firmware.

Before uploading ADX UnO firmware if a different band setup is preferred then first thing is to setup band assignment in the firmware for our needs! Which is assigning four preferred bands we plan to operate on:

These are band assign code lines from ADX UnO firmware which needs to tailored for our preference:

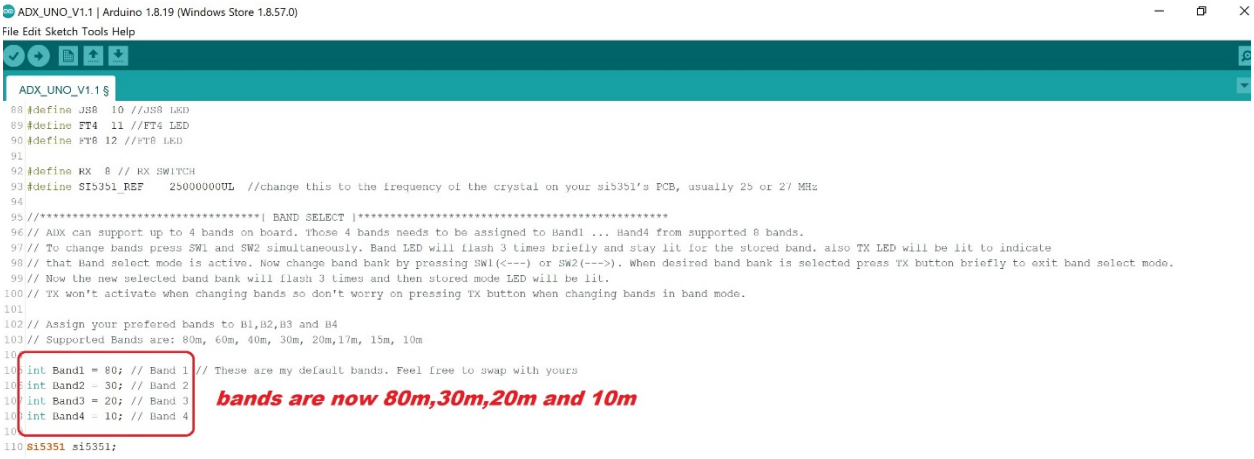
```
//*********************************[ BAND SELECT ]*************************************************
// ADX can support up to 4 bands on board. Those 4 bands needs to be assigned to Band1 ... Band4 from supported 6 bands.
// To change bands press SW1 and SW2 simultaneously. Band LED will flash 3 times briefly and stay lit for the stored band. also TX LED will be lit to indicate
// that Band select mode is active. Now change band bank by pressing SW1(<---) or SW2(--->). When desired band bank is selected press TX button briefly to exit band select mode.
// Now the new selected band bank will flash 3 times and then stored mode LED will be lit.
// TX won't activate when changing bands so don't worry on pressing TX button when changing bands in band mode.

// Assign your prefered bands to B1,B2,B3 and B4
// Supported Bands are: 80m, 40m, 30m, 20m,17m, 15m

int Band1 = 40; // Band 1 // These are my default bands. Feel free to swap with yours
int Band2 = 30; // Band 2
int Band3 = 20; // Band 3
int Band4 = 17; // Band 4
```

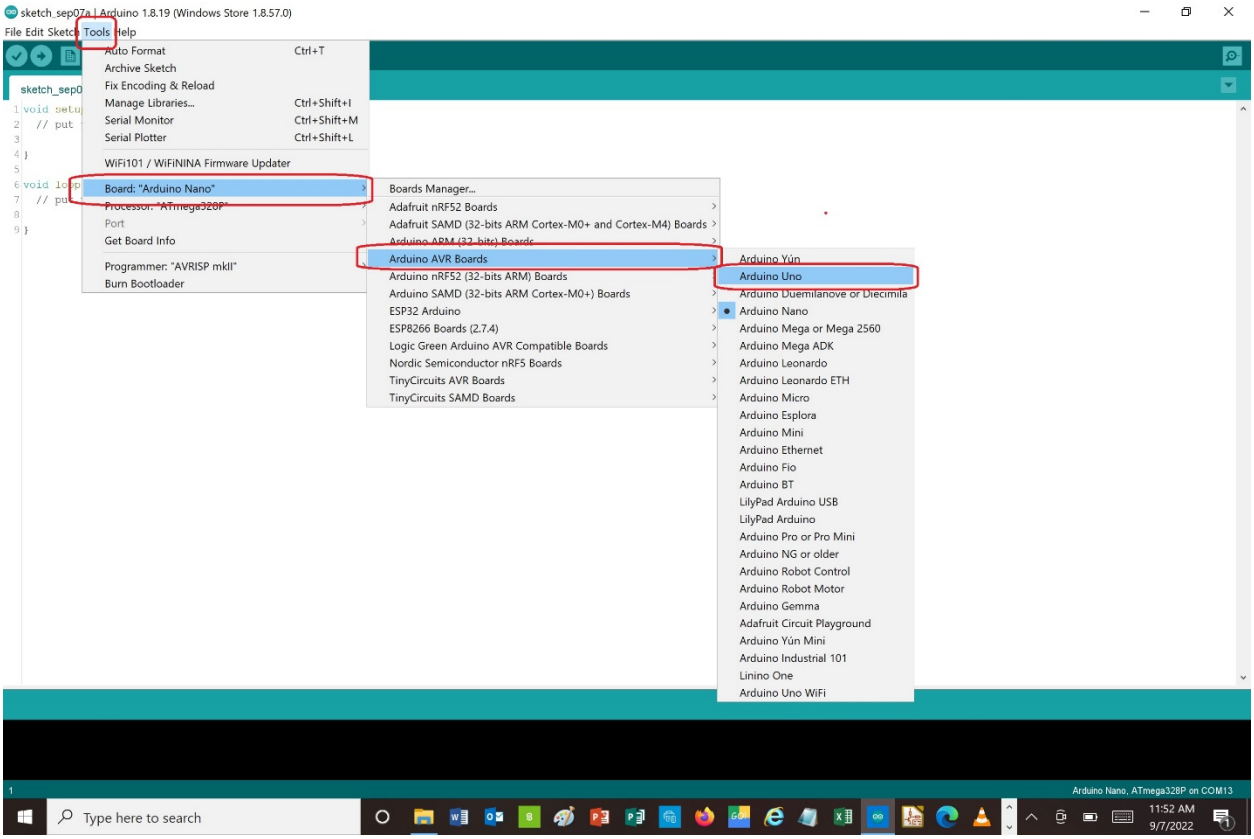
Assign your own preferred bands to those locations in the firmware which is inside the red rectangle above. ADX UnO default bands are 40m for Band1, 30m for Band2, 20m for Band3 and 17m for Band4. If this default band setup works for you then skip this step.

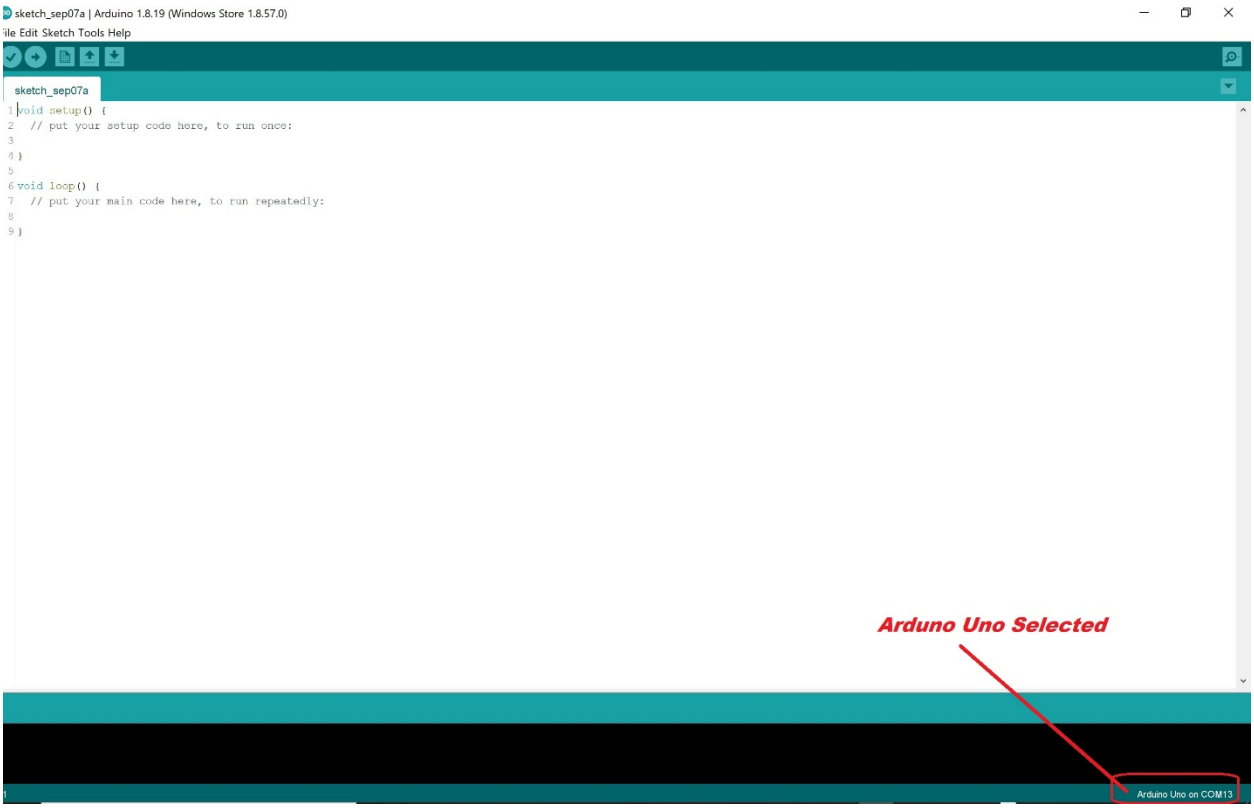
Below is an example for a different band layout. In this example 80m for Band1, 30m for Band2, 20m for Band3 and 10m for Band4 selected.



After this band assignment we can compile and upload ADX UnO firmware to Arduino Uno as uploading to any Arduino board. Nothing special here.

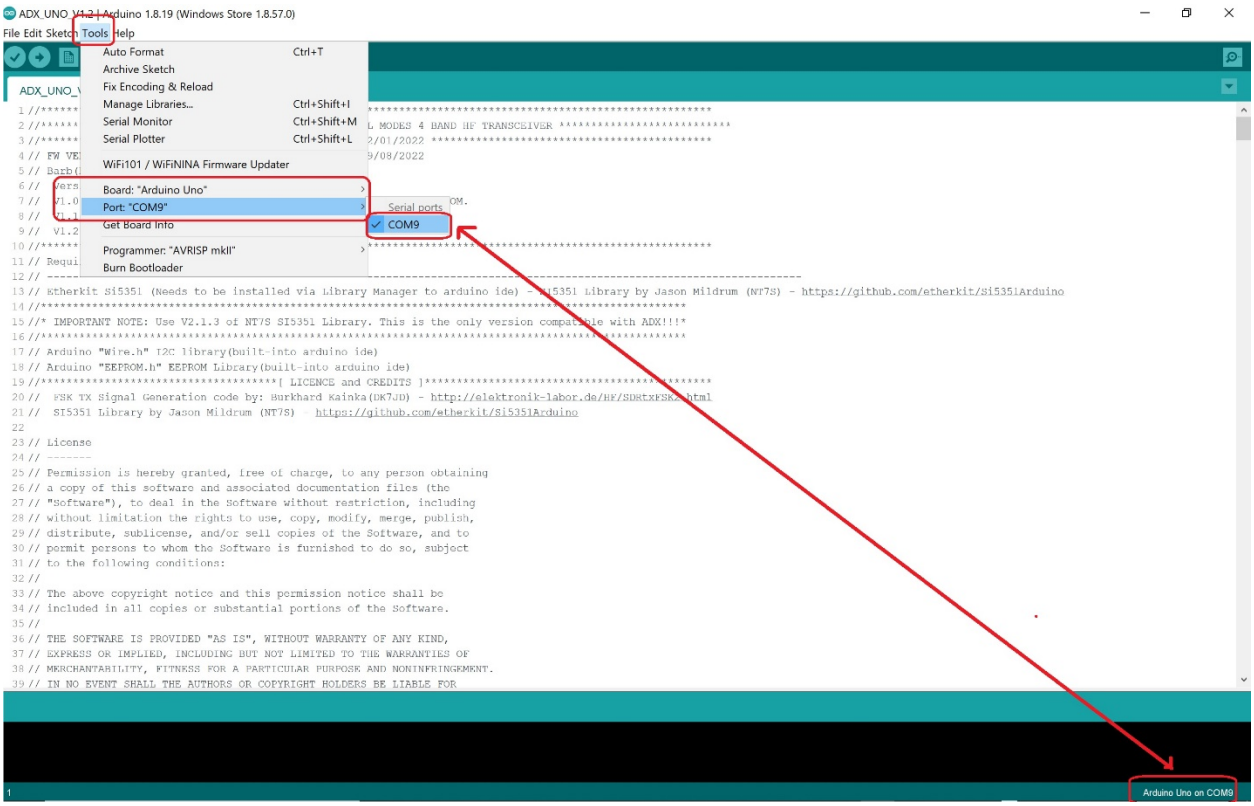
Select Arduino Uno Board following below menu items.





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

Now select Arduino Uno Com port:



- Now upload ADX 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 ADX UnO firmware install.

4 Power up and smoke test!

After firmware upload we are ready for first smoke test! Plug in your ADX UnO to your Arduino Uno. **Do not install LPF Band module at this time.** Connect 12V DC power source.

DO NOT USE More than 12V DC for ADX UnO Transceiver as this pushes RF power output beyond 5 Watts and over 60V which is over max limit of BS170 MOSFETS. This can release magic smoke on mosfets!

ADX UnO draws around 70 mA on RX and 500 mA on TX. If that's what is measured as current consumption under 12V then all is good with power up.

If all is ok in terms of voltage and current now we can test if SI5351 is generating any signal. To do this use a frequency counter or oscilloscope on **CLK1** test point. The default band is Band 1 and default mode is FT8. You should observe your choice of band 1 base frequency for FT8. For example in ADX UnO default band scheme Band1 is 40m and for FT8 it is 7074000 Hz or anything close to that as we didn't yet calibrated the SI5351 VFO. The important point is to see some kind of signal generation which is tell tale of SI5351 is working! SI5351 Signal at CLK1 should be a square wave signal.

Now plug in 40m Band LPF module as default ADX UnO is on 40m. Connect a 10 WATT capable RF Power meter and press TX button to see RF Power output. It should be around 3.5 Watts.

This verifies rf power mosfets are also alive and kicking!

If no smoke and nasty smell and there is signal from SI5351 and there is rf power output then we are in business and passed smoke test! ☺

5. Calibration Procedure for SI5351 VFO:

For SI5351 VFO Calibration Procedure follow these steps:

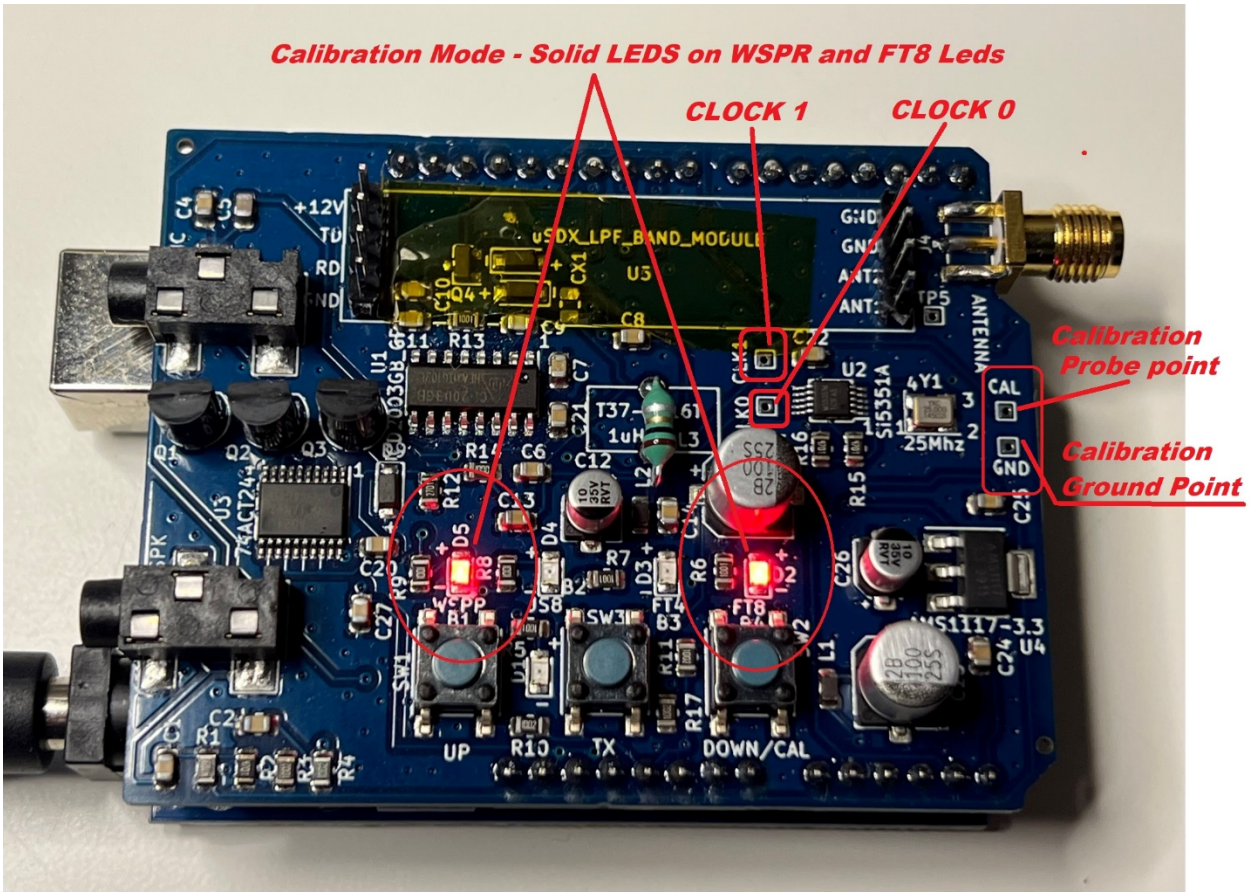
1 - Connect CAL test point and GND test point on ADX UnO to a Frequency Counter or Oscilloscope that can measure 1 Mhz up to 1Hz accurately.

2 - Press SW2 / --->(CAL) pushbutton and hold.

4- Power up with 12V or with 5V by using Arduino Nano USB socket while still pressing SW2 / --->(CAL) pushbutton.

5 – When FT8 LED and WSPR LED flashes 3 times and stay on, release SW2 / --->(CAL).

Now Calibration mode is active.



6 - Using SW1(<---) and SW2(--->) pushbuttons change the Calibration frequency until reading 1 Mhz = 1000000 Hz exact to the nearest hertz on Frequency counter or Oscilloscope.

The waveform is Square wave so frequency calculation can be performed easily.

7 - If you read as accurate as possible 1000000 Hz then calibration is done.

8 - Now save calibration value to EEPROM. In order to save calibration value press TX button briefly. TX LED will flash 3 times which indicates that Calibration value is saved.

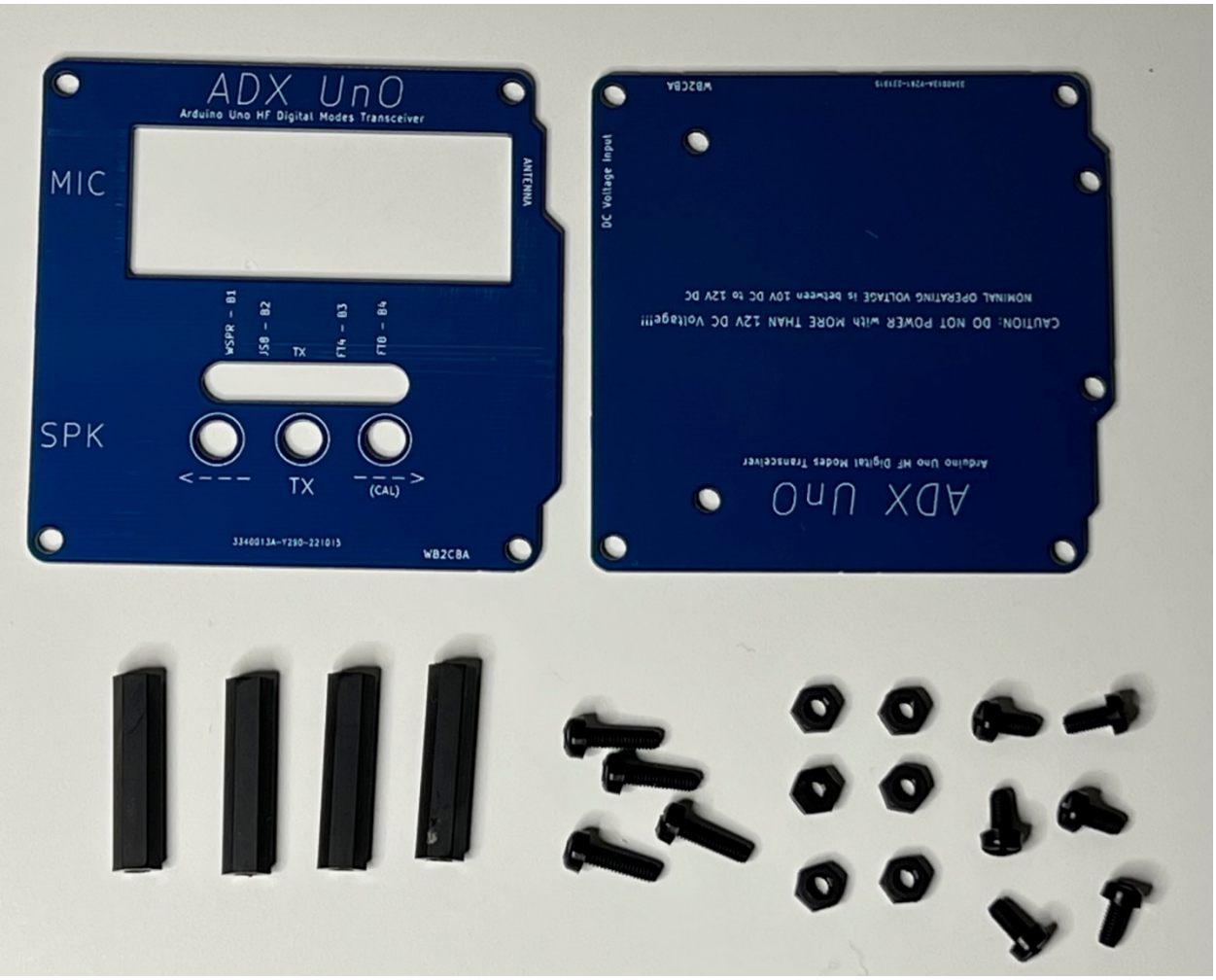
10 – Now Power off ADX.

DO NOT FORGET TO SAVE CALIBRATION VALUE BY PRESSING TX BUTTON BRIEFLY AT THE END OF EACH CALIBRATION SESSION. OTHERWISE CALIBRATION WILL REVERT TO THE DEFAULT VALUE.

Calibration will be loaded back in every power up so no need to repeat calibration once done. Also new firmware update won't reset calibration and no need to re-calibrate again.

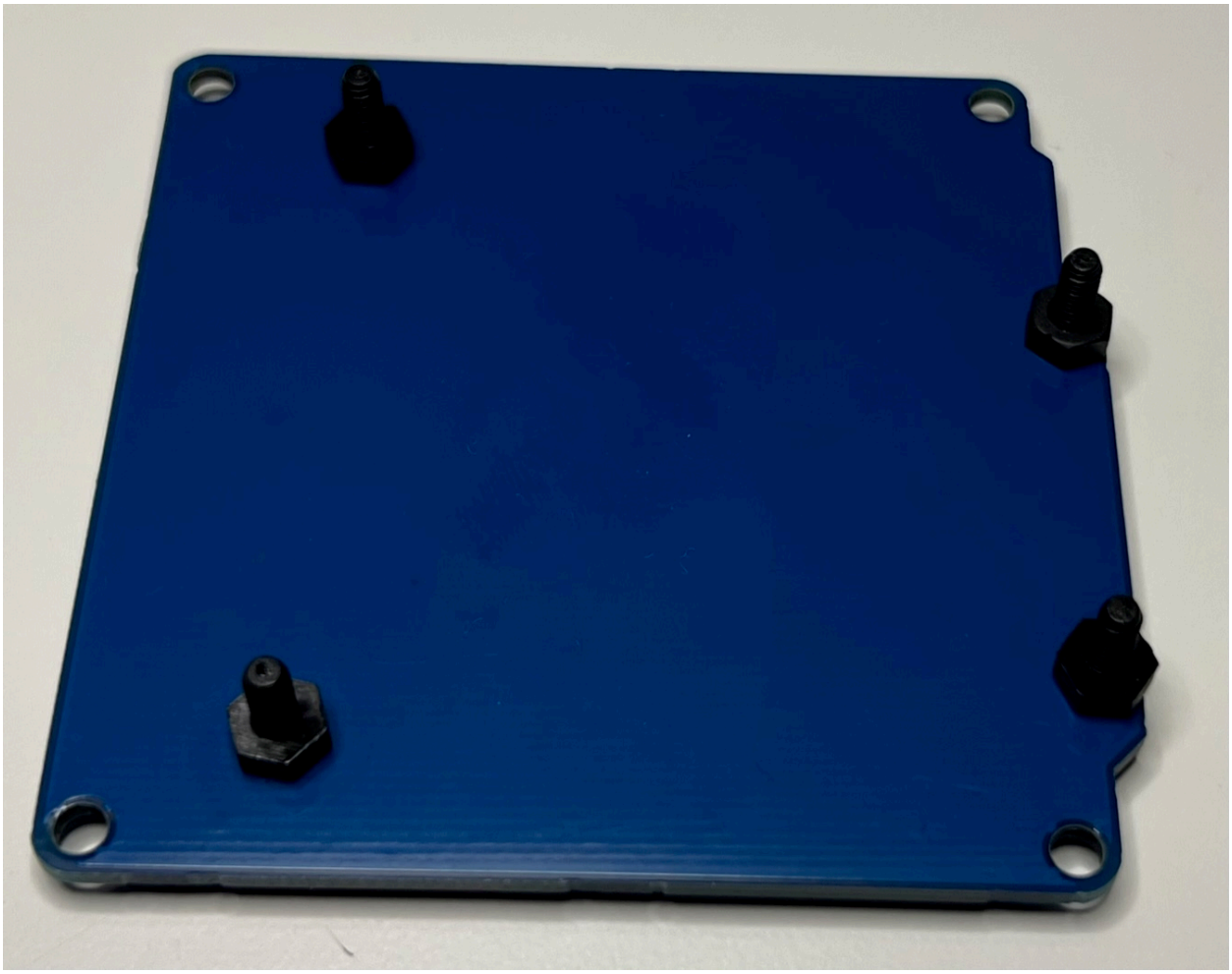
After calibration, plug in desired LPF band module. To activate Transmit for testing purposes we can always use TX button which acts as a PTT(Push To Talk) button and no need to connect to a digital modes software for transmit. This RF outputs a continuous wave signal only for testing RF power. Also this can be used for Antenna Tuner tuning.

Installing Top and Bottom Panels to form ADX UnO Case:



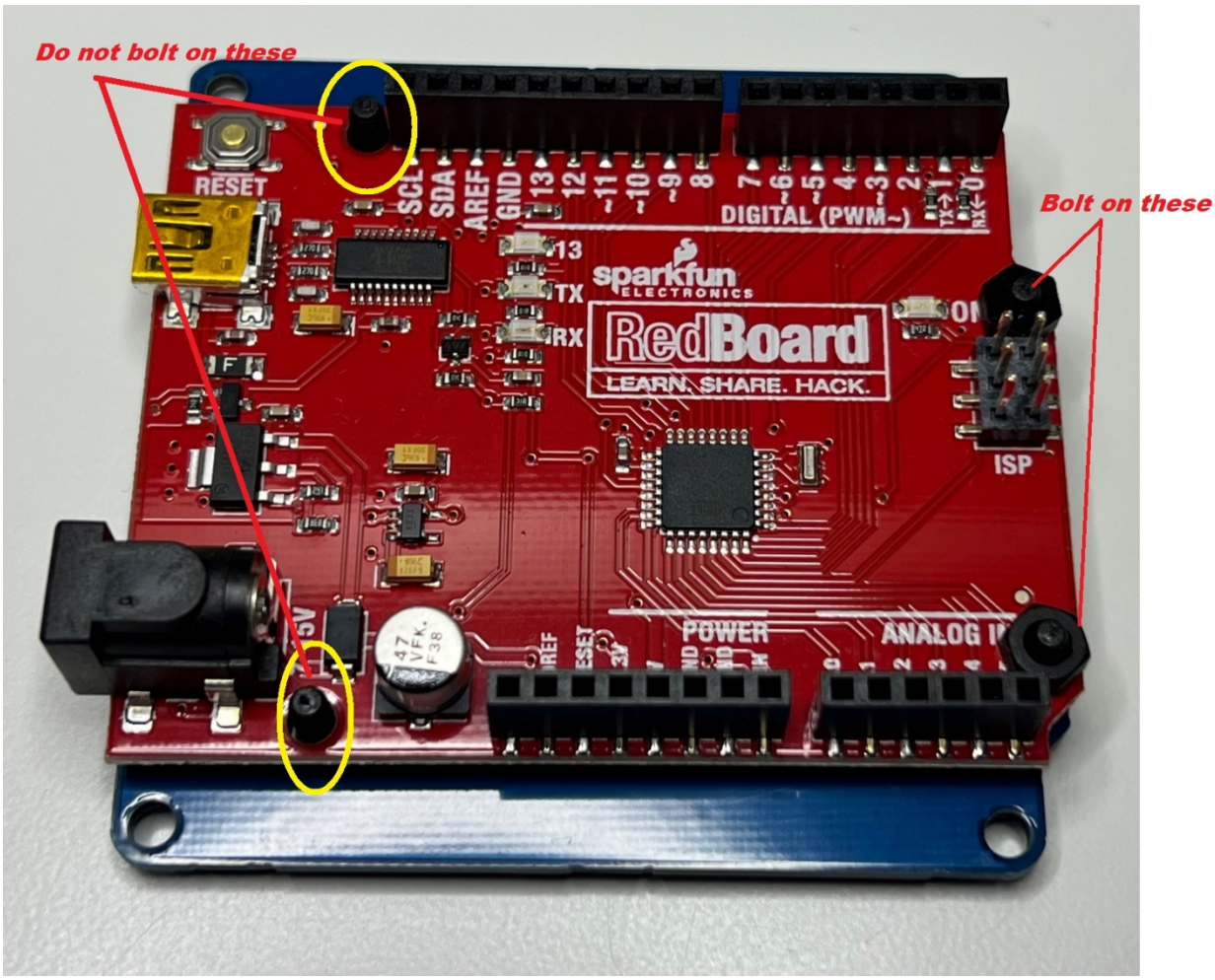
These are Top and Bottom Face plates for ADX UnO and all hardware to put it together. Now that ADX UnO passed smoke test and SI5351 Calibration is done it's time to encase!

STEP 1:



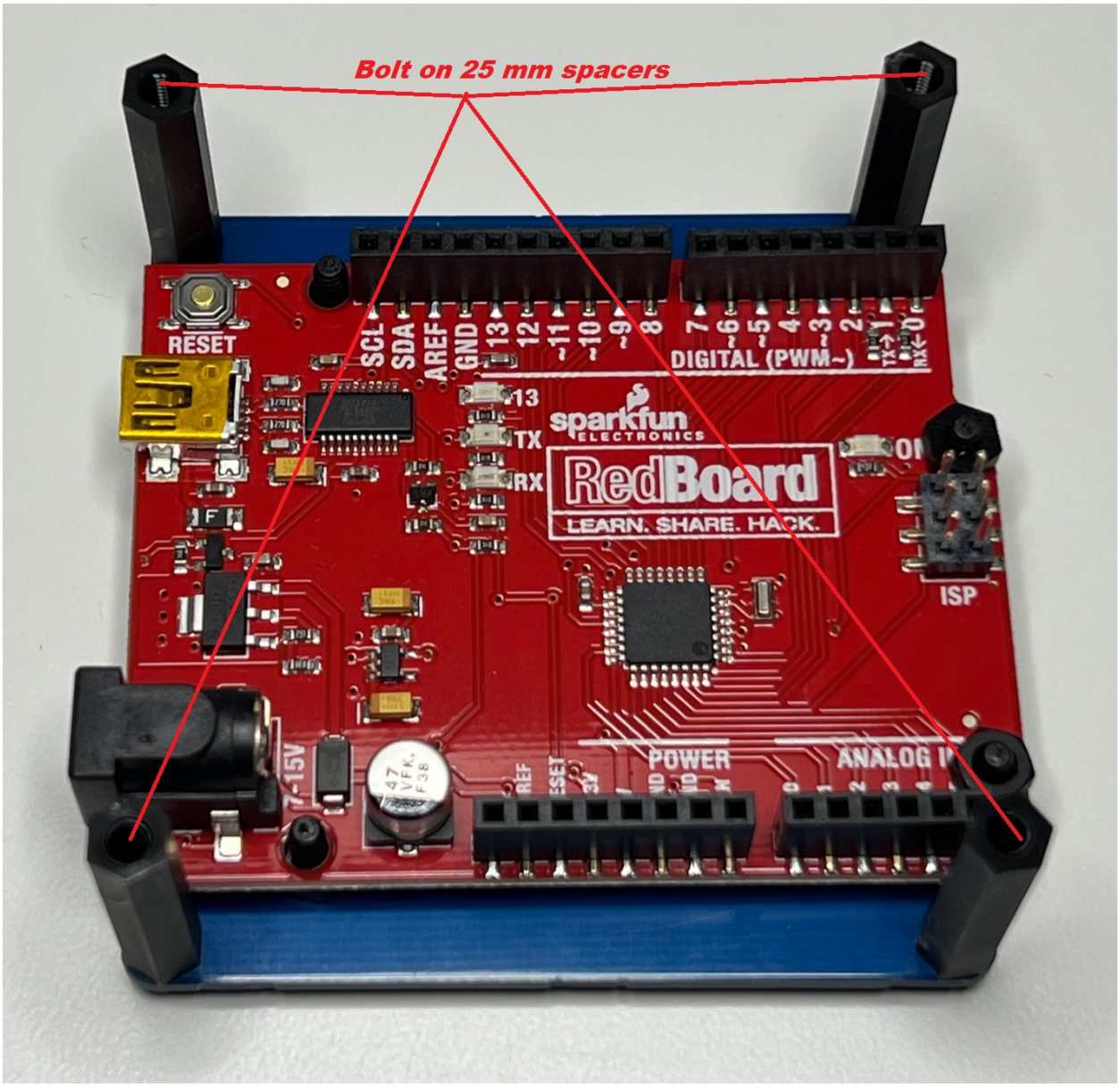
Install four plastic bolts and nuts as shown.

STEP2:



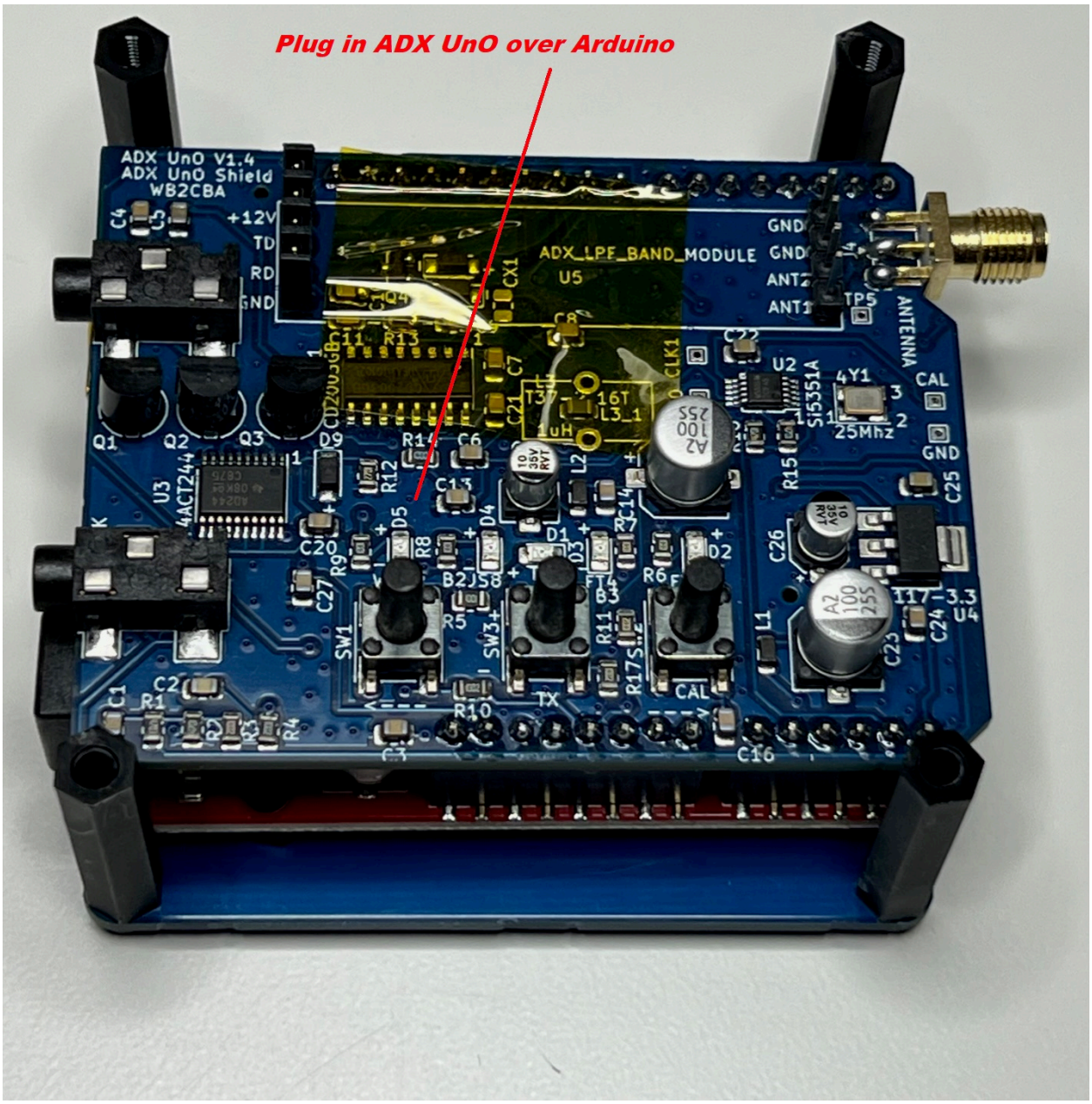
Insert Arduino Uno board as shown and screw in two nuts as shown.

STEP 3:



Bolt on 4 x 25mm stand offs as shown.

STEP 4:



Plug In ADX UnO over Arduino Uno.

STEP 5:

Bolt on Top plate.

This completes ADX UnO Top and bottom plate installation.

Operating Controls and switching bands:

1- When you power on ADX UnO one of the LED's will briefly flash 3 times and then another LED will be on solid. The first LED which was flashing 3 times indicates the active or selected BAND bank. By default, it is Band1.

   For example, if the LED is blinking in BAND1 which is 40m band in ADX UnO as default band now ADX UnO is set for operate on 40m band.
   The second LED which is solid is showing the active MODE. For example, if it is in FT8 then ADX UnO will operate on 40m in FT8.

   So how we will change band?

- Press **< ---** and **--- >** switches simultaneously briefly.

- One of the BAND LED will blink 3 times and stay solid. Also TX LED will be on. This doesn't mean that we are transmitting anything! It is just an indicator to show we are in band switch mode.

- Now using < --- and --- > switches we can turn on LED on different band banks. Of course we need to know which band is assigned to which bank. For that I use labeling tape to label each band bank corresponding to each band setup in the firmware.

- After we select the band we would like to operate on, we can exit Band select mode by briefly pressing TX switch. As soon as we press TX switch the TX LED will turn off and the selected band's LED will flash 3 times to notify us that is the band ADX UnO will operate on. After blinking Band LED turns off, the mode LED will light up showing the mode we are on.

- **WSPR LED also doubles as Band1 bank LED.**

- **JS8 LED also doubles as Band2 bank LED.**

- **FT4 LED also doubles as Band3 bank LED.**

- **FT8 LED also doubles as Band4 bank LED.**

- Now we can scroll between modes with < --- and --- > switches. The LED that is on is the active mode. We can't change band by scrolling left or right. This only changes operation mode such as FT8, FT4, JS8 or WSPR with in that selected band. To change band we

need to simultaneously press two switches briefly. Then we enter band change mode which is indicated by TX LED on.

- **Now the most important part is not to smoke any finals! When we change a band we need to make sure that band's dedicated LPF band module is plugged on before any attempt of TX!!!**

Connecting ADX Transceiver to Computer and firing up ADX UnO:

1- Connecting ADX 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- Connect Soundcard MIC input to ADX UnO MIC input with one of the 3.5mm audio cables. Do the same connection from soundcard SPK output to ADX UnO SPK input with the other 3.5mm audio cable.
3- Run WSJT/X or JSCALL software.
4- Power on ADX UnO.
5- When you power on ADX UnO one of the LED's will briefly flash 3 times and then another LED will be on solid. The first LED which was flashing 3 times indicated the active selected BAND bank. So now we know which band we are going to operate on so double check if that LPF Band Module is plugged in and we are good to go! Enjoy ADX UnO with Digital modes similar to any other HF Transceiver you will operate digital modes on!

For setting WSJT/X to work with ADX UnO:

- Go to Settings/Radio and activate PTT as VOX.
- Choose your soundcard under Settings/Audio menu
- Set Speaker Volume to 100% and then lower while operating TUNE on WSJT/X until there is no flicker on TX LED. If there is a flicker then increase volume. Do not be afraid to increase volume to full. Nothing will break. ADX UnO needs proper volume level to operate reliably.

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.

Building blocks of ADX UnO Transceiver:

- Brains of ADX UnO is an Arduino Uno which takes care of signal generation from audio tones of digital modes, user interface and calibration of SI5351 Module. ADX 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 ADX 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 ADX UnO is inspired from Burkhard Kainka(DK7JD) - http://elektronik-labor.de/HF/SDRtxFSK2.html work.

  So how does this 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 Nano'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. that comes with SSB signal generation technique which 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 and CLK2 calibration signal. SI5351 module works with 400khz I2C speed.

- TX BUFFER: TX buffer is an inverting 74ACT244 octal buffer connected in parallel to drive Class E RF power amplifier consisting of 3 x BS170 mosfets. RF power output - varies from band to band - around 3.5 watts.

- Low Pass Filter Module is a serial resonance class E LPF filter similar to that one used in uSDX designs.

- RECEIVER is CD2003GB – A Chinese clone of Toshiba's TA2003GP AM/FM radio IC. Only AM section is used as a direct conversion receiver.

- 5 LEDs and three tactile pushbutton switches for user visual interface and controls.

- **LIMITATIONS and PRECAUTIONS while operating ADX UnO:**
  1. **When you change bands DO NOT Forget to change LPF band filter modules to match that selected band! Otherwise your PA mosfets might smoke!**
  2. **Do not power ADX UnO more than 12V DC. It can take up to 13 volts though there will be a danger of damaging PA mosfets as this will bring mosfets to MAX value of 60V that they can handle. Exercise with caution.**
  3. **Do not operate TX or press tune on WSJT/X without any antenna connection! RF PA mosfets will be damaged.**
  4. **Always use a tuned SWR antenna or an antenna tuner when operating ADX UnO. Not doing so might damage RF Power output mosfets!**

- ADX 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.

- ADX UnO is not an SSB or CW rig!

- ADX UnO has a DSB Double side band receiver configured as direct conversion receiver. It is not an SDR or SSB receiver.

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

- ADX UnO RF output should be around 3.5 Watts average depending on band. As the frequency gets higher RF power output will decrease. For example, 17m band RF output will be around 2.5 watts.

Schematics and PCB Layouts of ADX UnO TRX Design:

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

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.
- Scott Baker, KJ7NLA for designing ADX UnO 3D printed Case.
- Burcak Cubukcu, TA2EE for extensively testing ADX UnO prototype and for his inspiring suggestions.

Enjoy your Digital QSOs!

Barbaros Asuroglu aka Barb, WB2CBA

11/15/2022

wb2cba@gmail.com