ADX-S Kit Assembly Manual

CRKITS.COM
2024/1/20 V1.6: Formal Release with FAQ

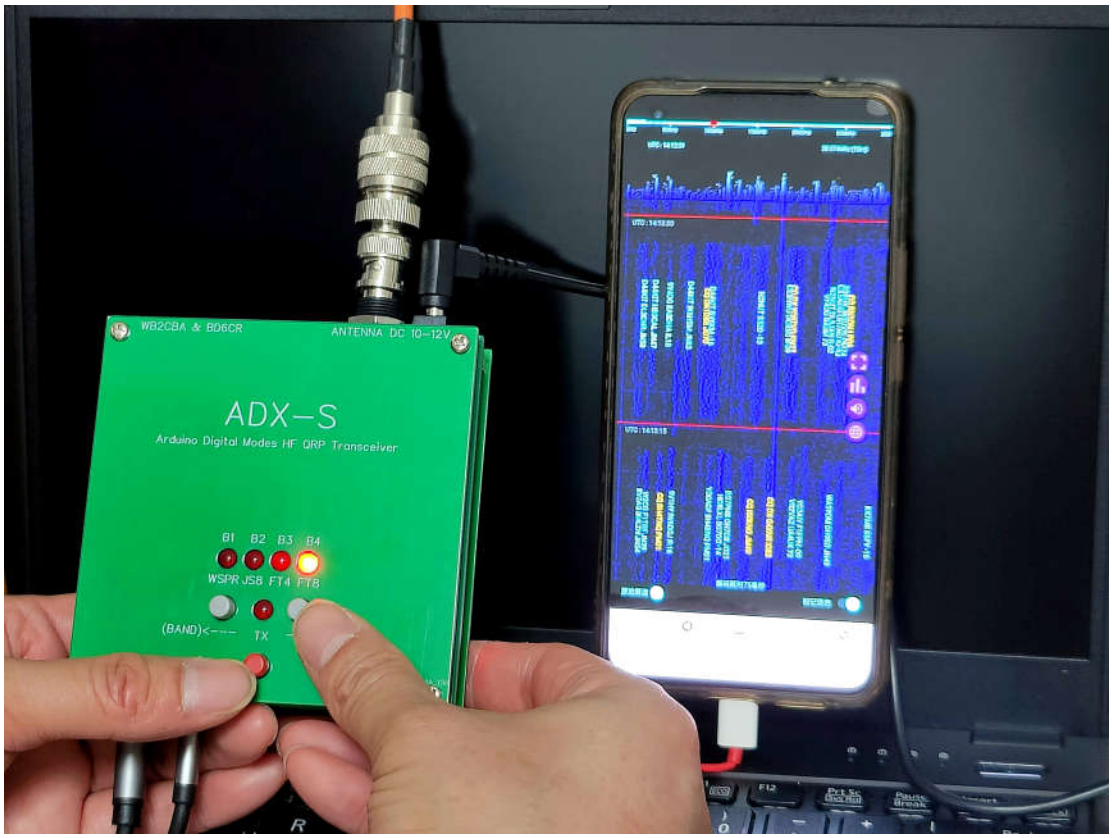


Photo: Jack Liang, BG7IKK who achieved DX contact of over 18,000 km with his EFHW antenna

Change history:
2023/3/26 V1.0: Formal Release
2023/3/29 V1.1: Formal Release with issue fixes
2023/4/2 V1.2: Formal Release with minor change
2023/4/29 V1.3: Formal Release with CAT and other minor change
2023/7/24 V1.4: Formal Release with modifications
2023/8/13 V1.5: Formal Release with 7 band upgrade
2024/1/20 V1.6: Formal Release with FAQ

Contents

ADX-S Kit Assembly Manual 1
- Contents 2
- Specifications 3
- Part List 3
- Theory of Operation 5
- Step by Step Kit Assembly 8
  - Step 1: Diodes 10
  - Step 2: +12V Power 12
  - Step 3: Arduino Nano Module and +5V and +3.3V Power 13
  - Step 4: Power supply for CD2003 15
  - Step 5: 5351 Module 16
  - Step 6: LEDs and Buttons 18
  - Step 7: TX Driver 74ACT244 20
  - Step 8: Complete the TX 21
  - Step 9: Audio Jacks 24
  - Step 10: CD2003 RX 25
  - Step 11: LPF band filter board 31
  - Step 12: Box the ADX-S 33
- Settings and Calibration 36
- Use the ADX-S 38
- Open Source Code Modification and Update 40
- CAT Support 41
- Modifications 42
- From 4 to 7-Band Upgrade 44
- Frequently Asked Questions and Answers 45
- Further readings 46
- Reference and Acknowledgement 46

The ADX (Arduino Digital Xcvr) is an open source Arduino based digital modes HF QRP transceiver designed by Barb, WB2CBA, who is known as one of the designers of uSDX. The ADX-S is a superhet modification by BD6CR. The receiver performance is much improved and the design details are refined to become a kit good for beginners. The normal building time is 3 hours, and the minimum requirement for test instruments is a frequency counter or just your commercial ham radio transceiver. You can use ADX-S with WSJT-x or JTDX on PC or FT8CN on your Android phone or tablet. For WSPR TX, you can also use WSPR TX Android App.

Specifications

Dimensions: 97 mm x 97 mm x 22 mm, not including the protruding parts
Weight: 150 grams, including one LPF band filter board
Power supply: 10-12V from DC IN Jack or 5V from USB Port
Receive current: about 60mA at 12V
Transmit current: about 600mA at 12V (lower on 15 and 10 meters)
Transmit RF output: about 3.5W at 12V, about 0.3W at 5V (about half on 15 and 10 meters)
Spurious suppression: about -45dBc
Modes supported: Digital modes including WSPR, JS8, FT4 and FT8 on Upper Side Band
Bands supported: 40, 20, 15 and 10 meter by plugging in each band filter board
Audio jacks: MIC (to MIC jack of sound card) and SPK (to PHONE or SPK jack of sound card)
USB port: Type C

Part List

Please follow the part list to do parts inventory first. Report missing parts to your seller.

The Main Board

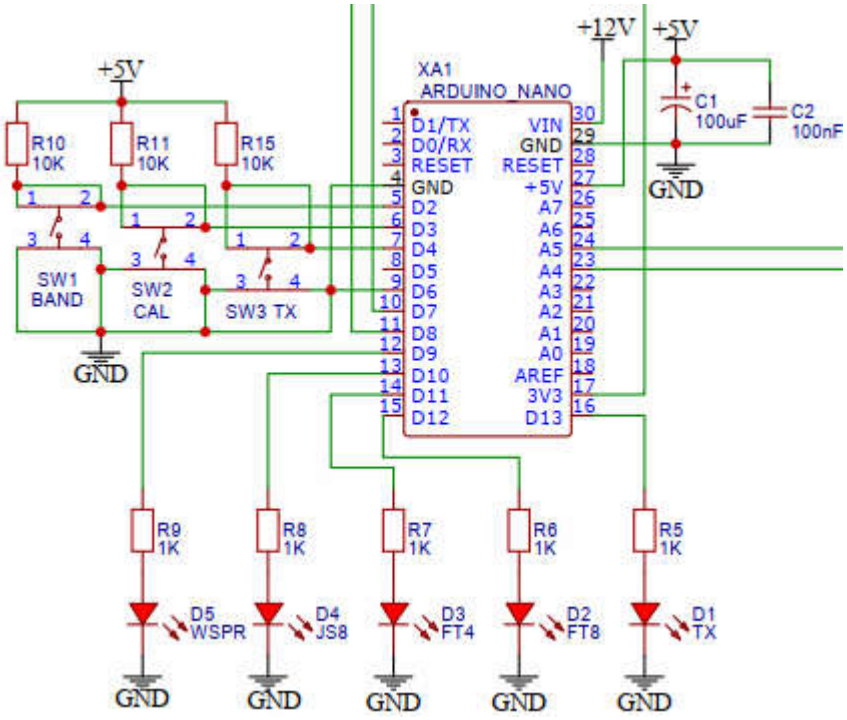
Part	Qty	Designator
100uF/16V	3	C1,C16,C24
10uF/16V	2	C13,C18
560pF, 620pF or 680pF	1	C11
10nF	4	C4,C5,C12,C17
100nF	13	C2,C3,C6,C8,C9,C10,C14,C15,C19,C20,C21,C22,C23
470nF or 1uF	1	C7
5.1pF	1	C25
PFB455JR Ceramic Filter	1	FL1
SPK 3.5mm Audio Connector	2	CON1
MIC 3.5mm Audio Connector		CON2
TX 5mm Red LED	5	D1
FT8 5mm Red LED		D2
FT4 5mm Red LED		D3
JS8 5mm Red LED		D4
WSPR 5mm Red LED		D5
1N4148	3	D6,D7,D8
1N5817	1	D9
1N4756A Zener Diode	1	D10, between Test Points RFO and GND
BNC	1	J1
DC-005	1	J2
1uH	1	L1
100uH	1	L2
BS170	4	Q1,Q2,Q3,Q4
1M	1	R1
2.7k	1	R12
100	1	R14
4.7k	1	R2
10k	7	R3,R4,R10,R11,R15,R16,R17
1k	6	R5,R6,R7,R8,R9,R13
UP(BAND) Tactile Switch With Grey Cap	2	SW1
DOWN(CAL) Tactile Switch With Grey Cap		SW2
TX Tactile Switch With Red Cap	1	SW3
CD2003GB_GP	1	U1
SI5351_Module or Compatible	1	U2
74ACT244	1	U3
LPF_BAND_MODULE Sockets	2	U4_A,U4_B
Arduino_Nano or Compatible (Pre-Programmed)	1	XA1
Main PCB	1	
Top Panel PCB	1	
Bottom Panel PCB	1	
Screws	4	
Standoff M3x12+6	4	
Standoff M3x6+6	4	
Standoff M3x5	4	

The LPF Band Module

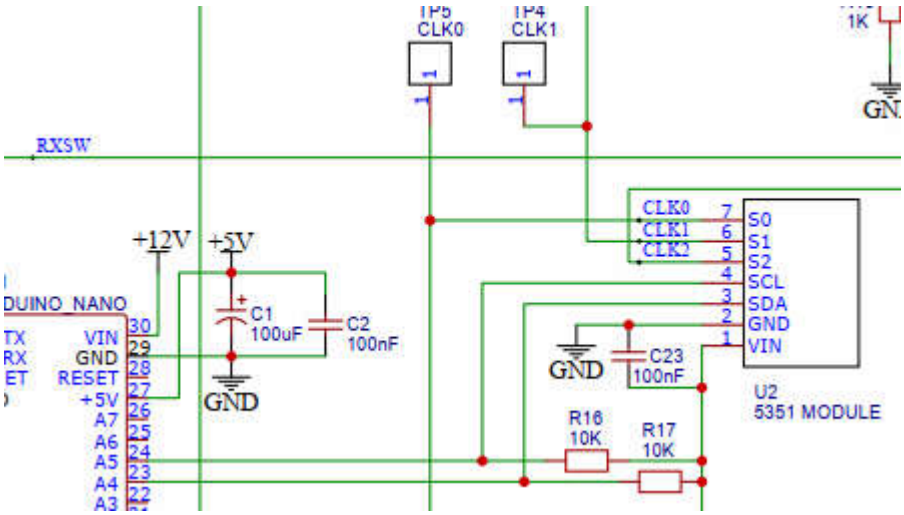
LPF BAND MODULE										
Band	C1	C2	C3	C4	C5	C6	C7	L1	L2	L3
40m	120+10	470	470	1000	300	1000	100nF	12T/BLACK	15T/RED	10T/RED
30m	100	330	330	680	220	680	100nF	12T/BLACK	12T/RED	8T/RED
20m	68	470		470	150	470	100nF	12T/BLACK	10T/RED	7T/RED
17m	91	360		360	120	360	100nF	12T/BLACK	11T/YELLOW	7T/YELLOW
15m	68	300	15	300	100	300	100nF	12T/BLACK	11T/YELLOW	7T/YELLOW
12m	47	270		270	62	270	100nF	12T/BLACK	10T/YELLOW	6T/YELLOW
10m	39	240		240	75	240	100nF	12T/BLACK	9T/YELLOW	6T/YELLOW
Unit: pF by default										

Theory of Operation

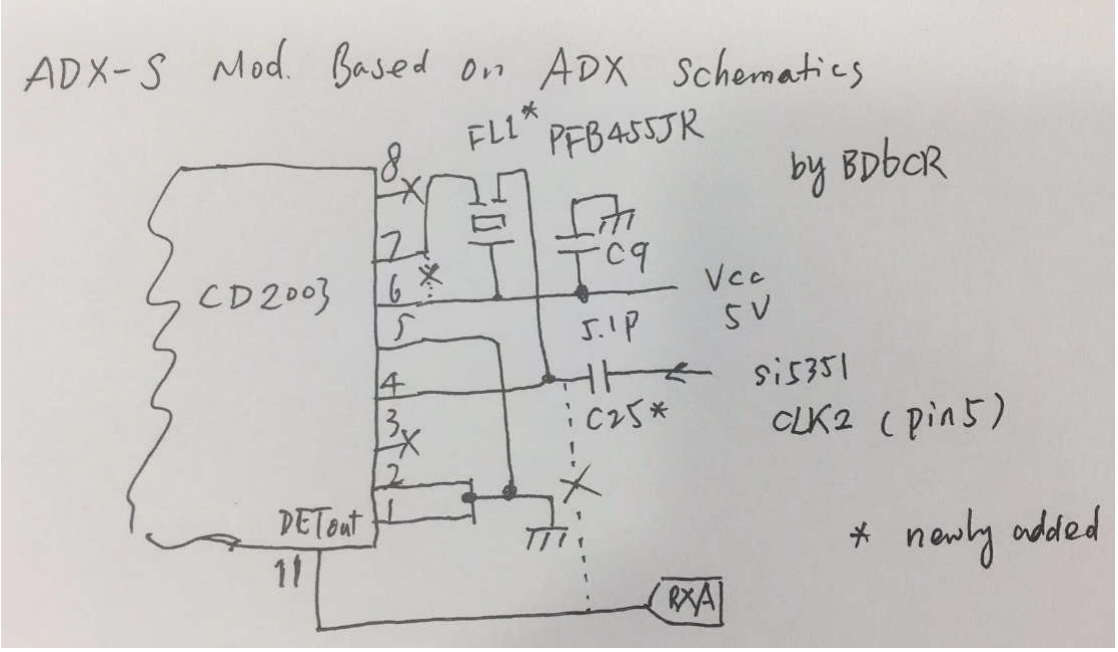
The Arduino Nano module (XA1) is the control center. Besides the handling of buttons, LEDs and 5351 frequency generation module, it counts the incoming audio frequency and adds to the carrier frequency to complete the FSK modulation. So, ADX-S transmits single signal instead of DSB signals like D4D. The Arduino Nano does not process receiving signal, so it is purely analog. The Arduino Nano also provides regulated 5V to the main board.

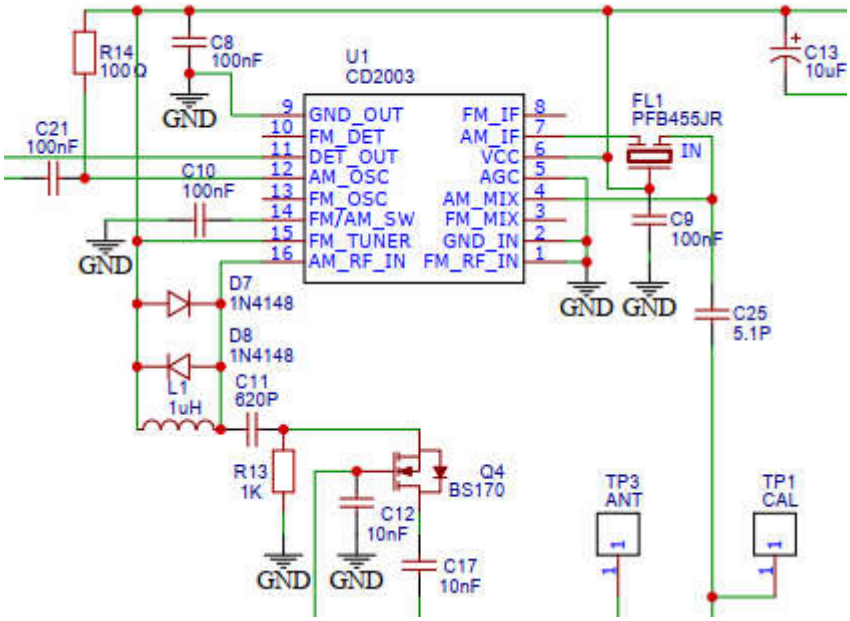


The 5351 module (U2) is controlled by the Arduino Nano module through the I2C interface. The module can generate 3 different square wave signals from the S0 to S2 ports. The S0 is for transmitting, the S1 is for receiving local oscillator, and the S2 is for receiving beat frequency. In calibration mode, the S2 port outputs 1MHz signal for calibration at the test point CAL.

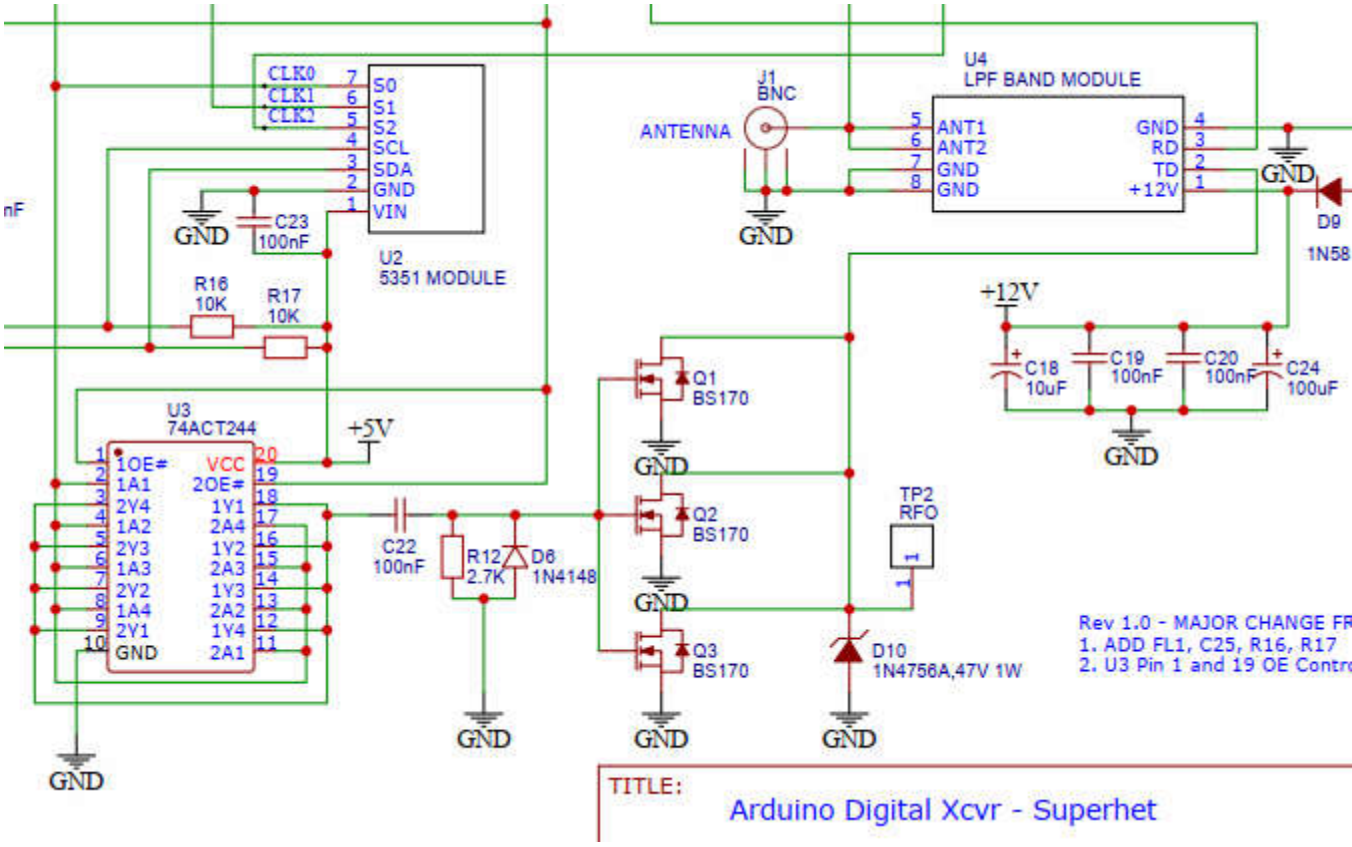


The CD2003 (U1) is a compatible Chinese model of Toshiba TA2003, which is a mature AM/FM radio receiver chip. It uses ceramic filters instead of IF transformers as the IF filter. The original design of ADX uses the mixer stage for direct conversion receiver, while ADX-S adds a ceramic filter and injects beat frequency to make a simple superhet receiver thanks to the design ideas from JA9TTT. The PFB455JR filter by Murata is a 2-element ceramic filter to provide better selectivity although the bandwidth is wider than 3kHz. The receiver front end is not tuned considering the multiple band design, so it is recommended to add a tuned ATU like Z-match tuner in between the ADX-S and antenna to somehow reject the strong BCI if necessary. A simple Z-match tuner can be made by two variable capacitors and a toroid, see https://www.qsl.net/g3vgr/zmatch.html. The following hand drawing schematic shows the receiver side modification from the ADX.





The transmitter of the ADX-S is the same as the original ADX design. The CLK0 from the 5351module (U2) is amplified by the driver stage of 74ACT244 (U3), fed to the final class E amplifier stage of 3 BS170 in parallel, filtered by the LPF band module (U4), and finally fed to the antenna through J1. The T/R switching is implemented by firstly switching CLK0 or CLK1+CLK2 outputs, and also switching by the RXSW line.



The ADX-S keeps the same 5 LEDs and 3 buttons like ADX. Among them, 4 LEDs are indicating both band (B1-B4) and mode (WSPR, JS8, FT4 and FT8). The band is shown by blinking at the

power on period, while the mode is shown by a solid LED on. By switching the mode, you also switch the frequency based on the fixed frequency table. The TX LED is on in both TX and BAND setting statuses. In normal working mode, TX button will enable manual transmitting, while in calibration or BAND setting mode, it is used to confirm and save the current setting. The major difference of ADX-S from ADX is the band switching way. Now you only can switch band by pressing and holding the BAND button at power on, instead of holding two buttons in normal working mode.

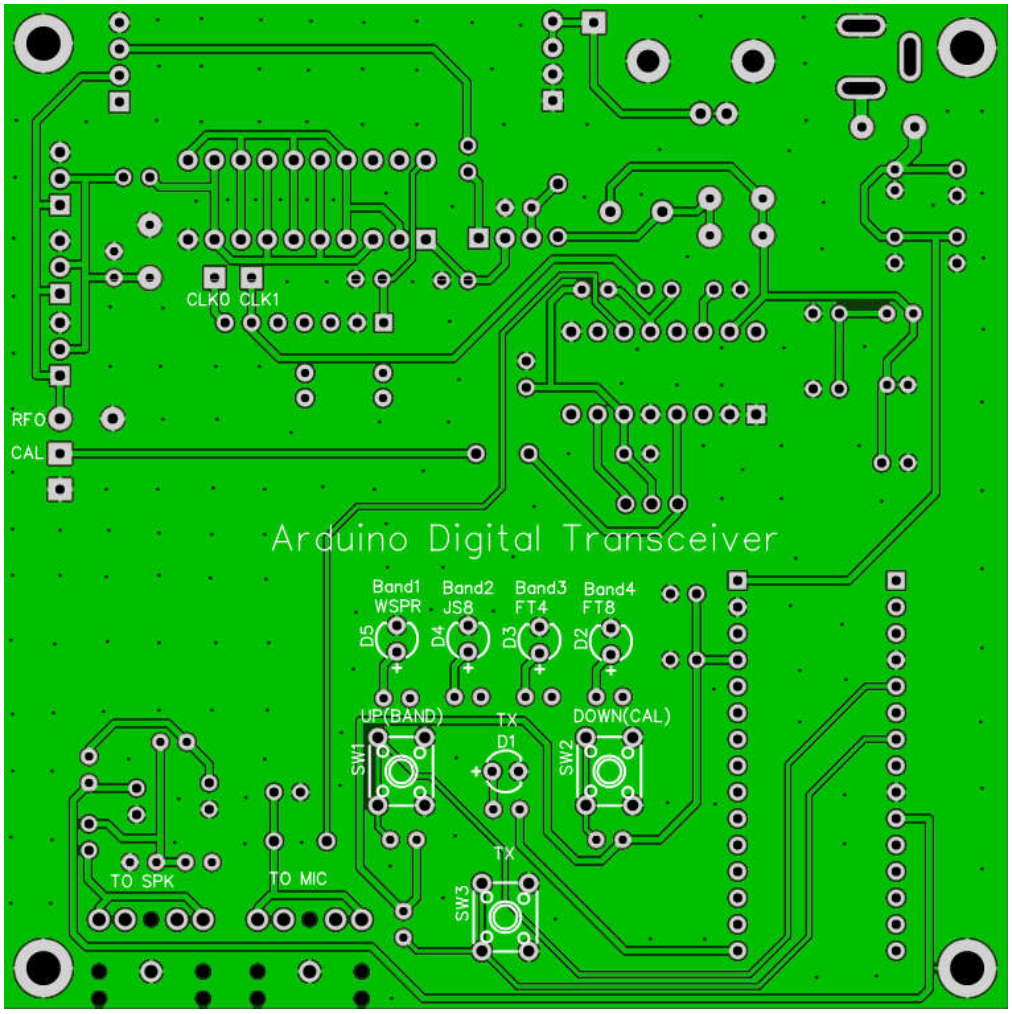
Step by Step Kit Assembly

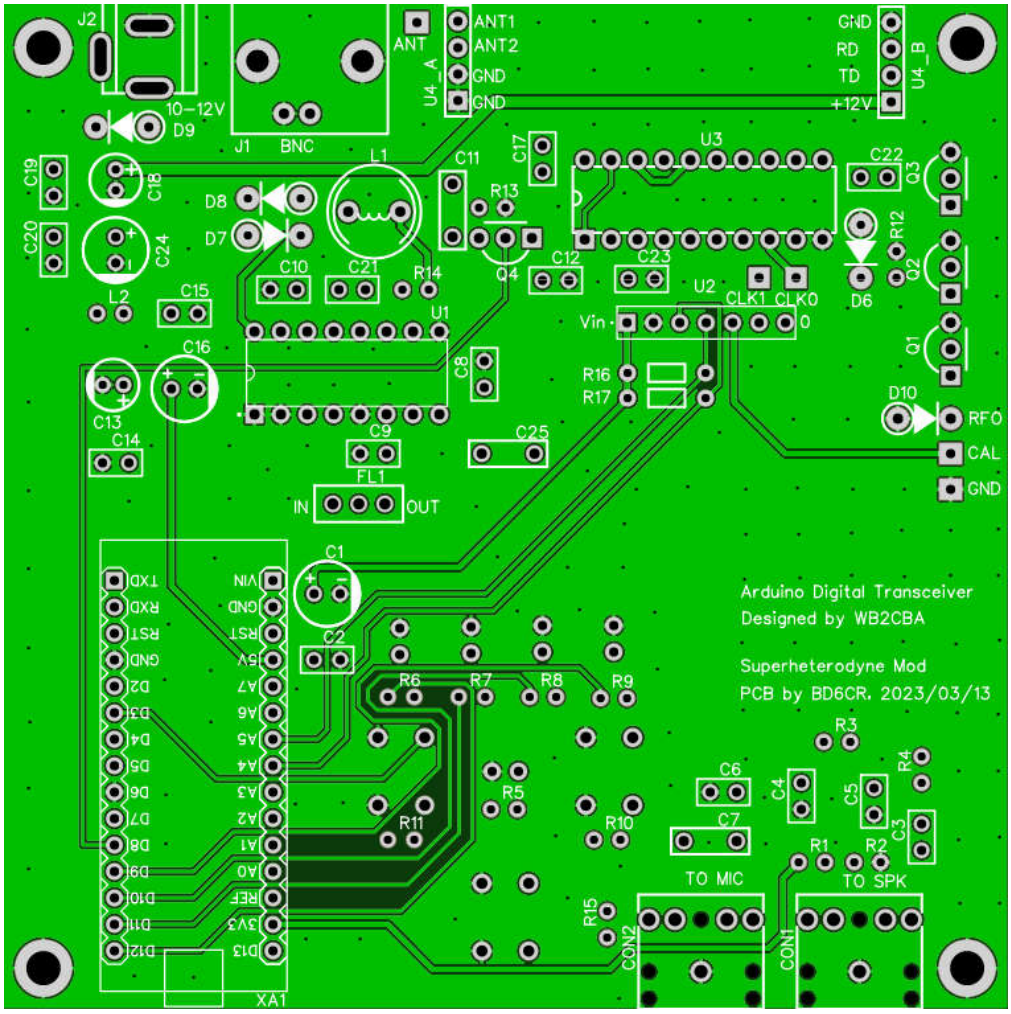
Let's start to build the kit. It is a kit of all though-hole components, so it is much easier for your eyes. It is recommended you use a fine tipped soldering iron since the pads are not very big. You will need to supply solder wire by yourself. 0.8mm diameter, 63% Sn and 37% Pb with flux is recommended.

The kit comes in a plastic container. In the container, you will see main PCB, top cover PCB and bottom cover PCB, a bag of parts for main board, and a bag of parts for 4 LPF band filters and audio cables.

Let's start with the main board. For the complete schematic and BOM, please download from Files section of our group. If you are not yet a member, please subscribe first.
https://groups.io/g/crkits/files/ADX-S

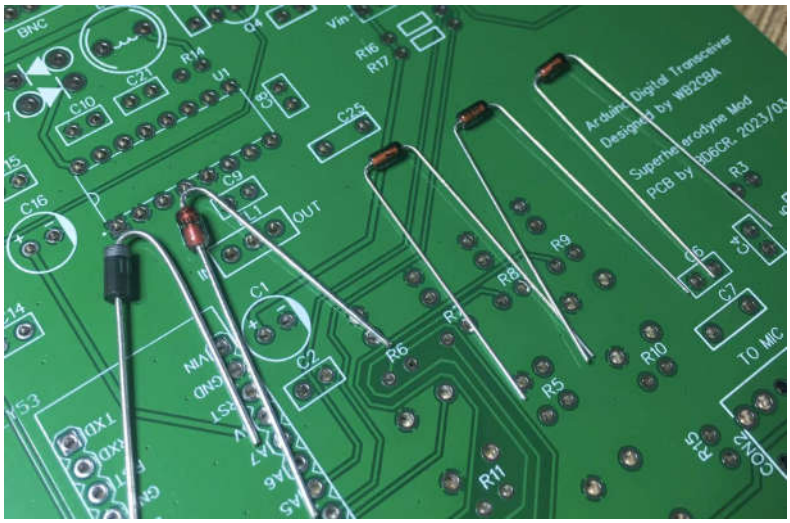
Here are the current PCB photos. Check the date for revisions.


CLK0 CLK1
RFO
CAL
Arduino Digital Transceiver
Band1
WSPR
Band2
JS8
Band3
FT4
Band4
FT8
D5
D4
D3
D2
UP(BAND)
TX
D1
DOWN(CAL)
SW1
SW2
SW3
TO SPK
TO MIC

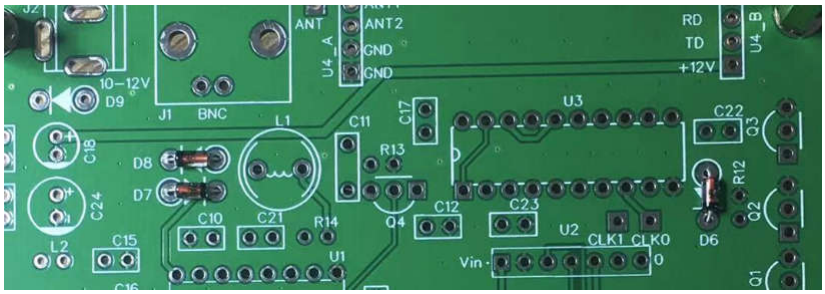


Step 1: Diodes

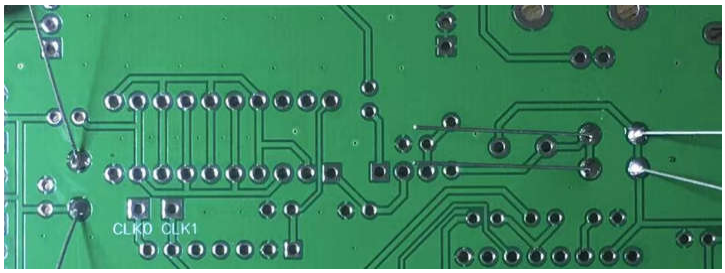
Bend the leads of diodes like this. From left to right: D9 – 1N5817, D10 – 1N4756A, D6-D8 – 1N4148.



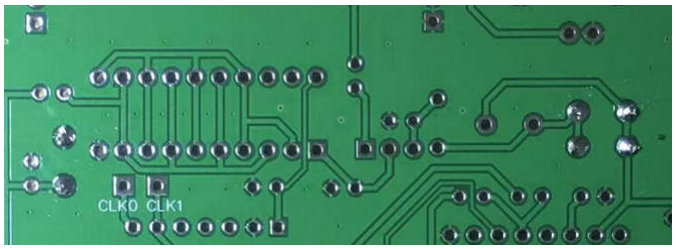
Populate D6-D8 as shown. The diodes have polarity so never swap the two leads.



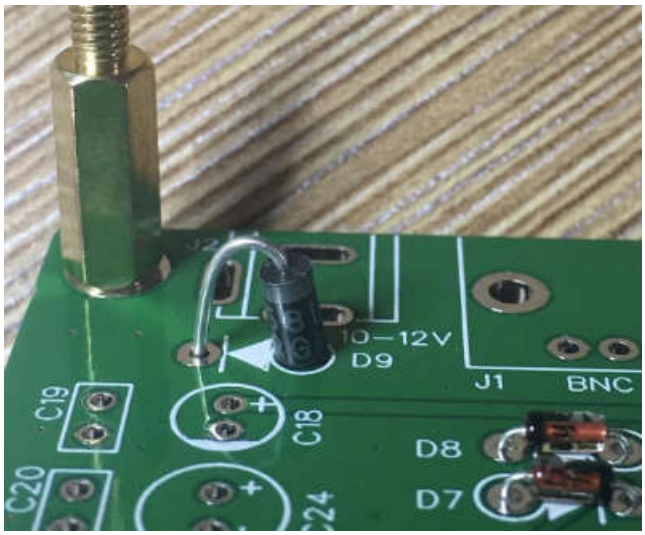
Bend the leads so they will not drop after you turn the PCB for soldering. If your soldering station supports temperature adjustment, you can adjust to 350C. It is good for both the leaded or lead-free solder wire. Solder about 3 seconds for each pad. Don't solder too long time.



Trim the extra leads using a wire cutter.



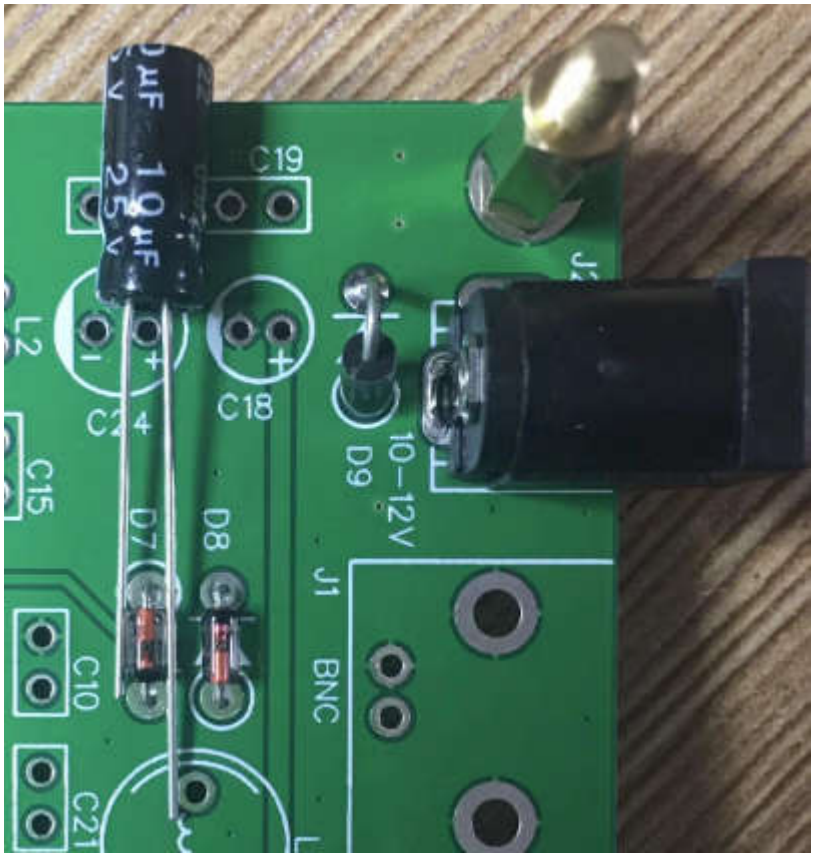
Populate D9 and solder. Never swap the two leads



Populate D10 and solder. Never swap the two leads.

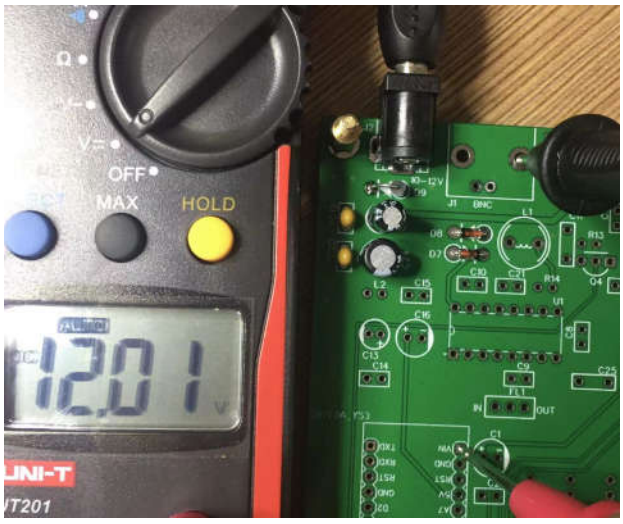
Step 2: +12V Power

The photo shows C18 – 10uF. The long lead goes to the pad with + marking. Never swap the leads.



Likewise for C24 – 100uF. Solder C18, C19-100nF, C20-100nF, and C24.

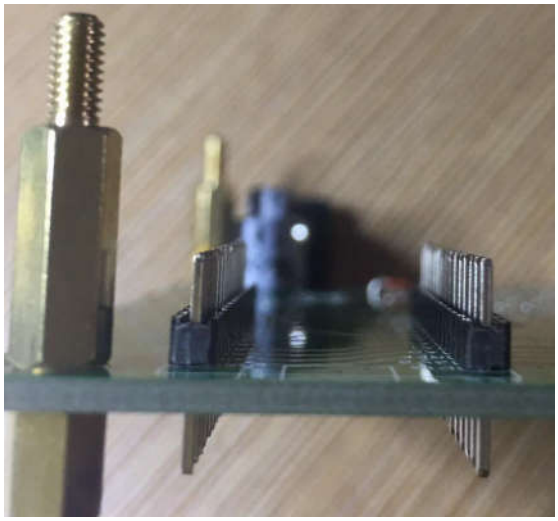
Now let's do a test to make sure the 12V does exist on VIN pin of XA1 – Arduino Nano module. Use the DVM for voltage measurement of about 12V. The red probe goes to the VIN pad, while the black probe goes to ground pad of the BNC connector. Plug in the 12V power supply (center positive) and measure. Looks good so far.



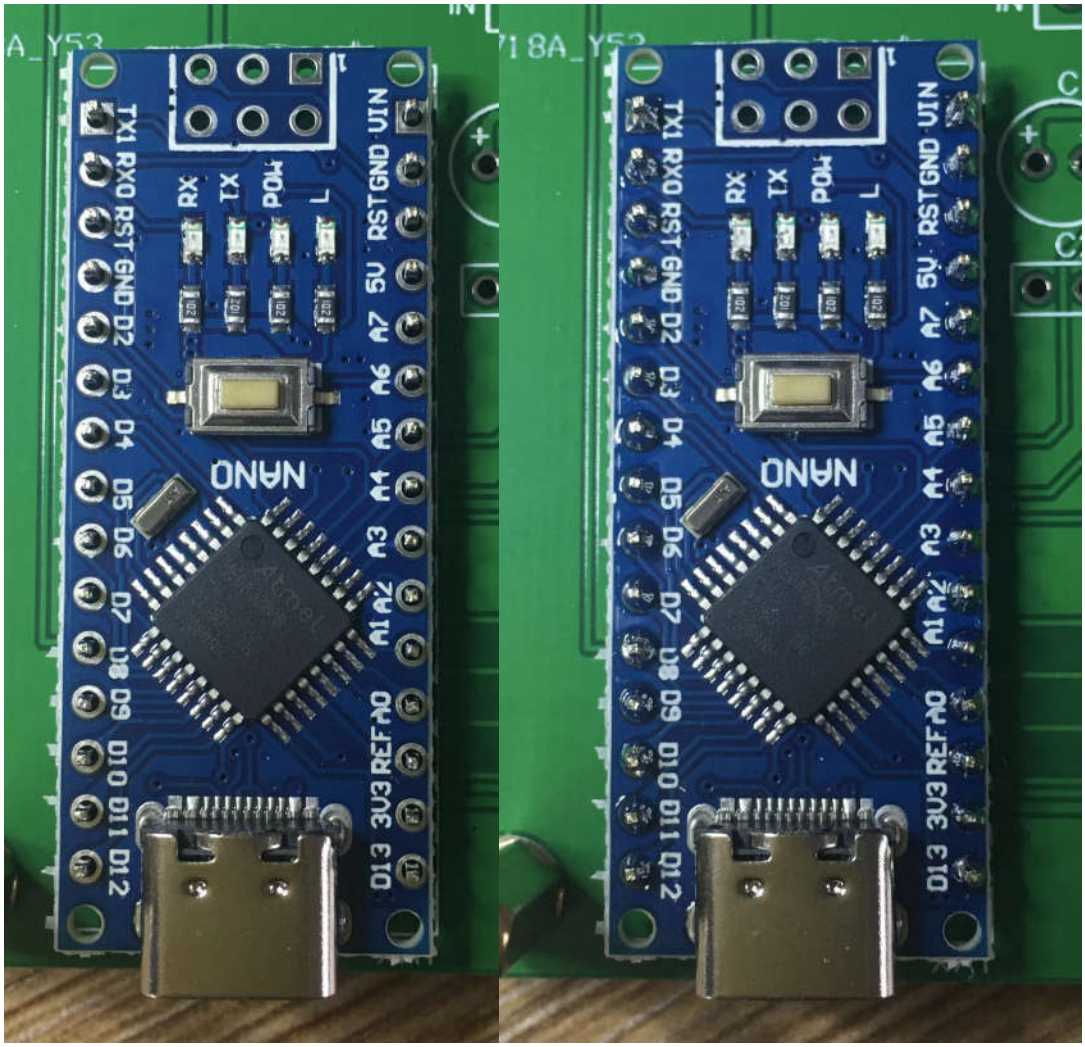
Step 3: Arduino Nano Module and +5V and +3.3V Power

The XA1 - Arduino Nano Module serves as the +5V and +3.3V power supply for the main board, so we will measure them after soldering the Nano module.

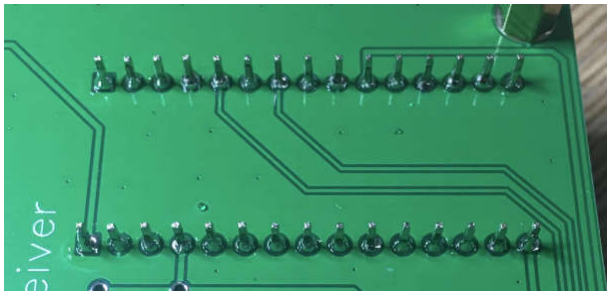
Use the pin headers to solder the Nano module on the main board. Populate both 15-pin headers, shorter lead side on the top.



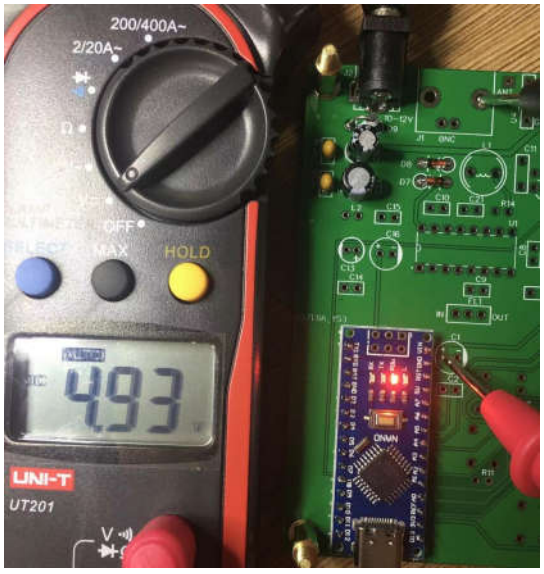
Plug in the Nano module. Make sure the USB port is at the main board edge. Watch out the holes in the four corners. Solder all 30 pins.



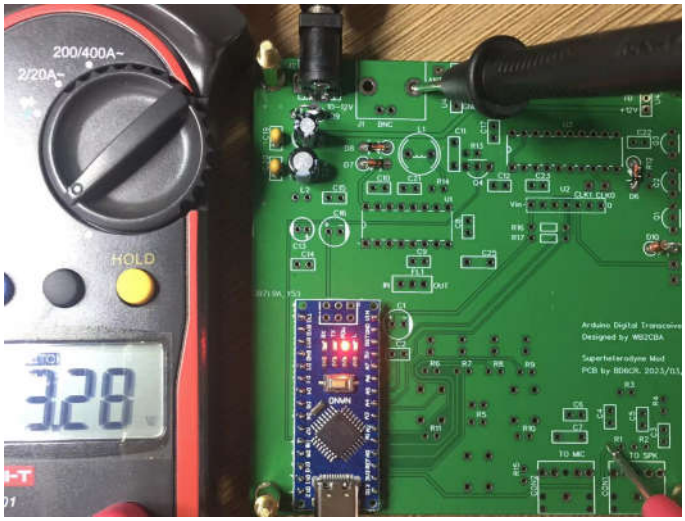
Turn the main board and solder all the leads of the pin headers. Don’t need to trim the extra leads.



Now it is test time. Plug in the 12V power supply (center positive) and measure the voltage on the + pad of C1 – 100uF with the red probe. 4.93V falls well in the tolerance range of 5V regulator on the Nano module. Don't short circuit in this step, or the regulator will be damaged and it is very difficult to desolder the Nano module.



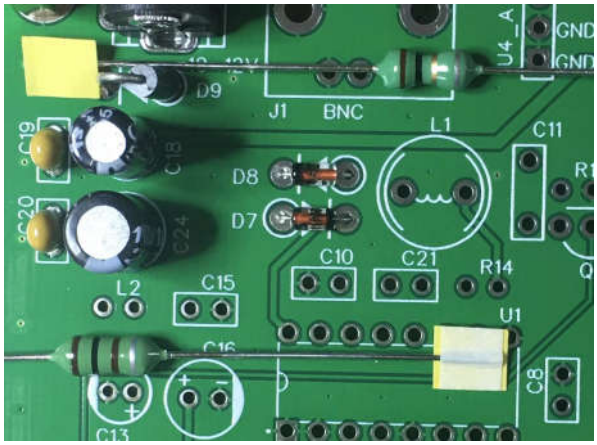
Measure 3.3V at the left pin of R1 – 1M. Looks good. Don't short circuit in this step, or the regulator will be damaged and it is very difficult to desolder the Nano module.



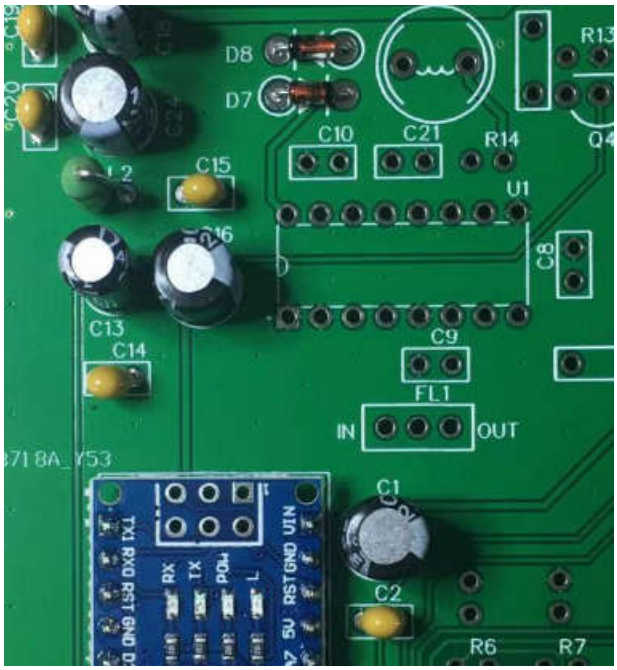
Step 4: Power supply for CD2003

The LC filter on the power supply line is necessary for an analog device like CD2003 receiver

chip. The photo shows the L1 – 1uH and L2 – 100uH fixed inductors.



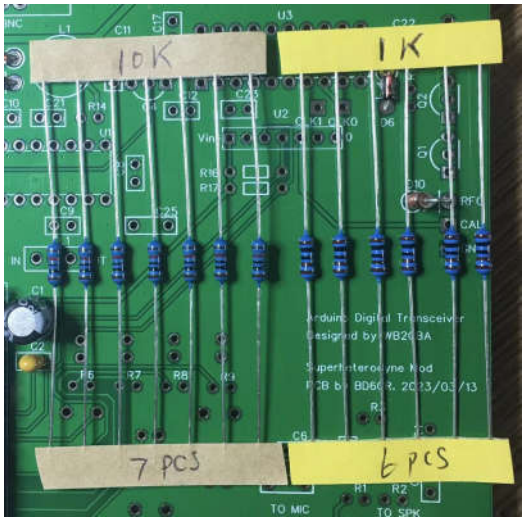
Solder L2, C13-10uF, C1, C16-100uF, C2, C14, C15-100nF. Beware of the polarity of C1, C13 and C16.



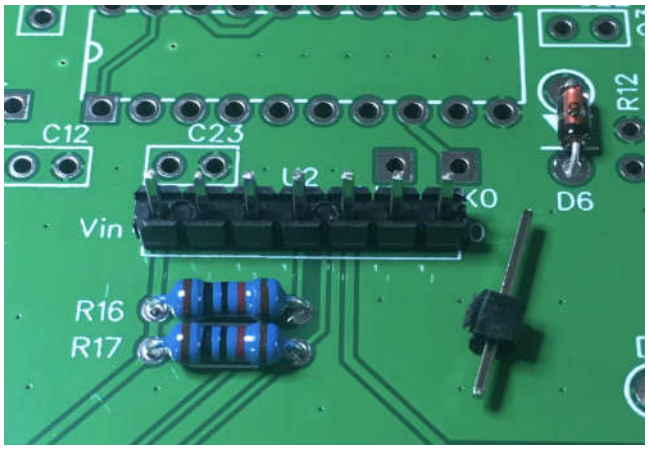
You can plug in the 12V power supply (center positive) again and test on pin 15 of U1-CD2003 to make sure the +5V power supply for CD2003 exist.

Step 5: 5351 Module

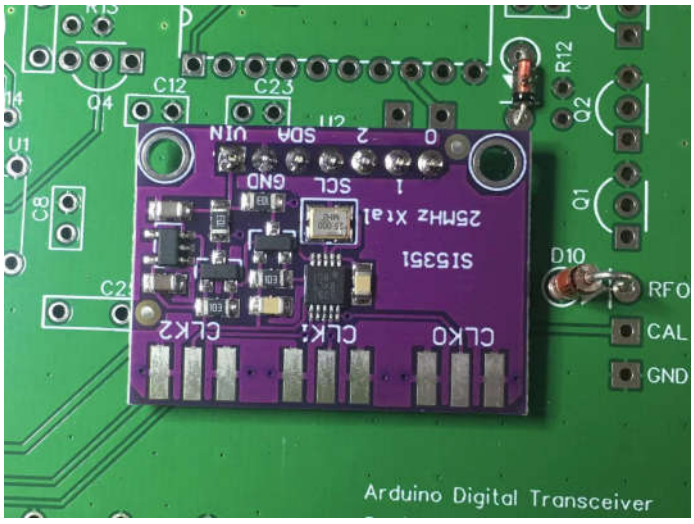
Let’s identify color codes of the blue resistors, and pick 2 pcs 10K resistors for this step.



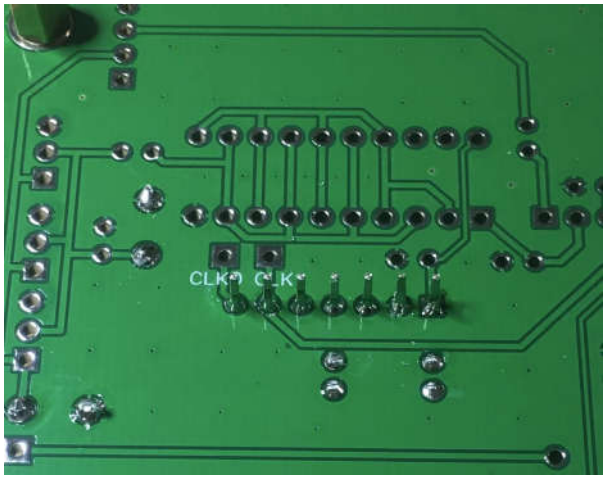
Populate R16, R17-10K as close to the PCB as possible and solder. The 5351 module has a 8-pin header but it only needs 7 pins, so you will need to trim 1 pin first, and plug in the 7-pin header, again, shorter lead side on the top.



Plug in the 5351 module on the pin header and solder. The chip on the module might be a Chinese compatible chip with Silicon Labs SI5351, but it works just fine.



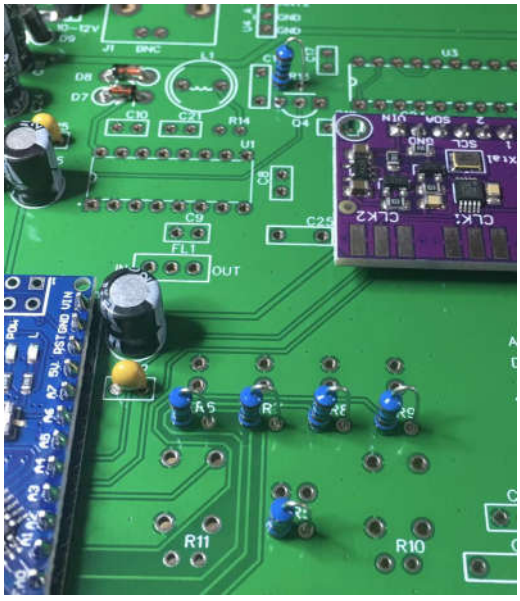
Turn the main board and solder the pin headers. Again, don't need to trim the extra leads.



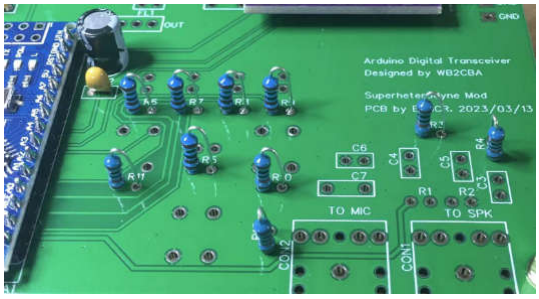
Let’s proceed to the next step then we can perform test for this module.

Step 6: LEDs and Buttons

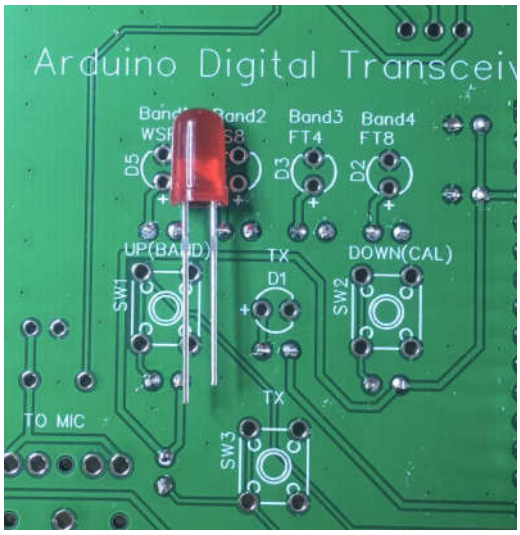
Populate all the 1K resistors R5-R9 and R13 and solder.



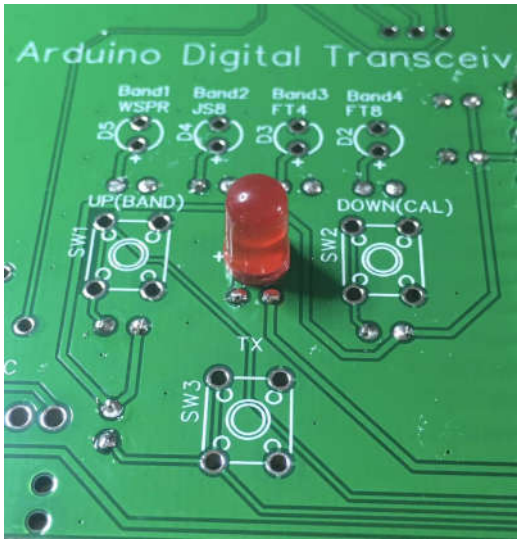
Populate all the 10K resistors R3, R4, R10, R11, R15 and solder.



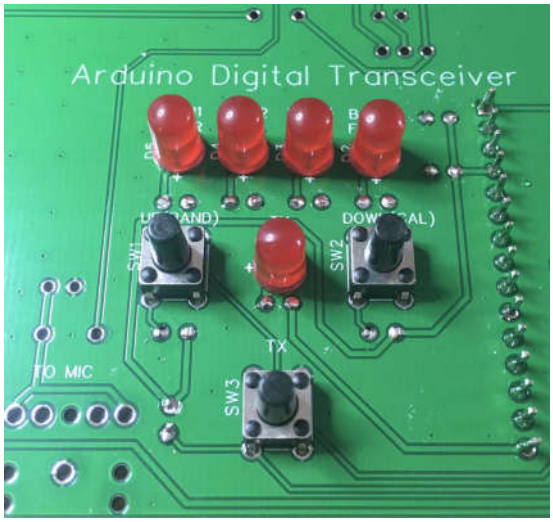
The long lead of the red LED goes to the + marking pad of D1.



Populate D1 as close to the PCB as possible and solder.

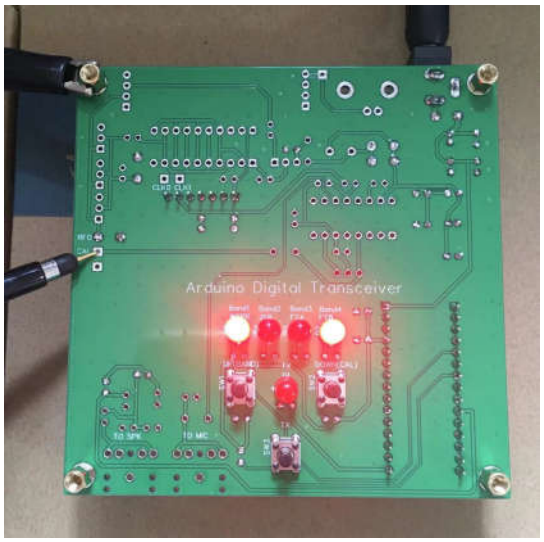


Populate D2-D5 and SW1-SW3 as close to the PCB as possible and solder.



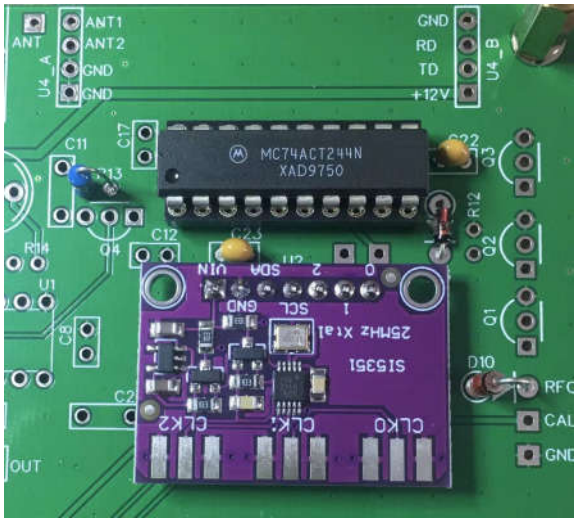
Now press and hold the SW2-DOWN(CAL) button and plug the power supply until you see both B1 and B4 LEDs are on. Now you can probe on the test point CAL using a frequency counter or other test instruments like an oscilloscope to see if you can get a 1MHz square wave. Please note,

you should clip on the ground first before you probe on the CAL test point. You can perform coarse calibration here if the frequency has big tolerance. Press SW1-UP(BAND) or SW2-DOWN(CAL) several times to get 1MHz as close as possible, then press SW3-TX to save the setting. Please note, the frequency counter on oscilloscope might not be very accurate, and some frequency counters might need higher amplitude to be able to read something.

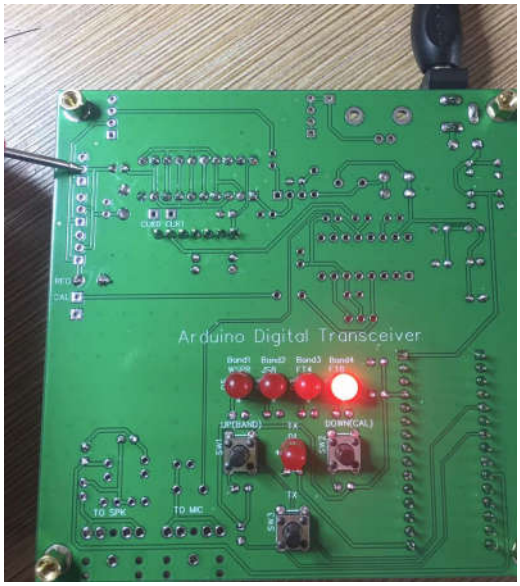


Step 7: TX Driver 74ACT244

Populate C22, C23-100nF, and U3-74ACT244 and solder. It is recommended you use a socket. The new chip comes with the leads not very vertical. You can carefully hold the chip with both hands and press on a table to make them vertical, so you can easily plug into the socket. Don't bend the pins one by one.



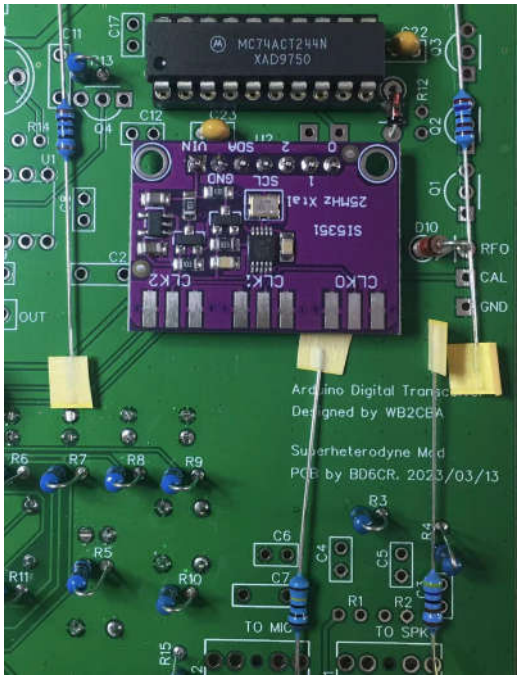
Now let's do a radiation TX test. Turn the board and power on. Use a probe of DVM to touch the center pin of Q3-BS170 as a radiation antenna.



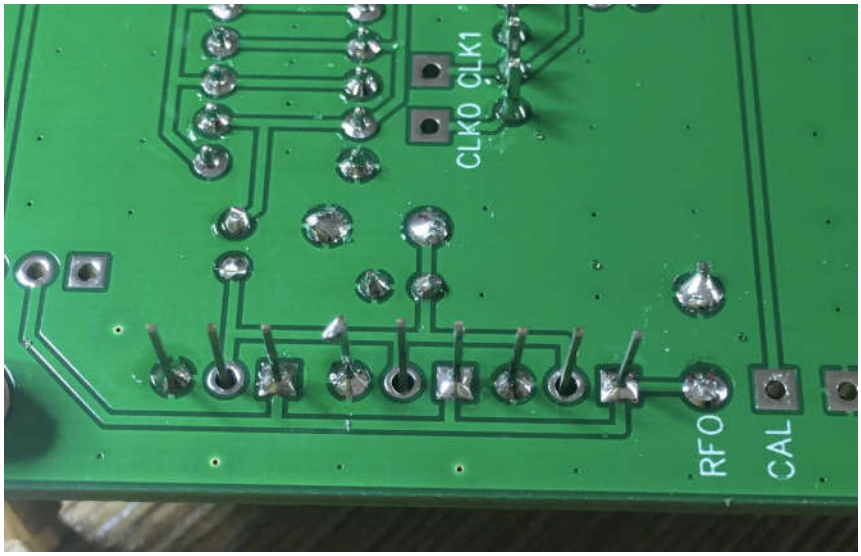
Press SW3-TX button to transmit, the nearby receiver will hear a quite strong signal. It is S9 plus 60dB on my TS-590 transceiver. I tuned to 14.074MHz because I have identified from the LED that the band is 20m and mode is FT8.

Step 8: Complete the TX

Identify the color codes of the rest 4 resistors. R1-1M, R2-4.7K, R12-2.7K, and R14-100R. Populate them and solder.

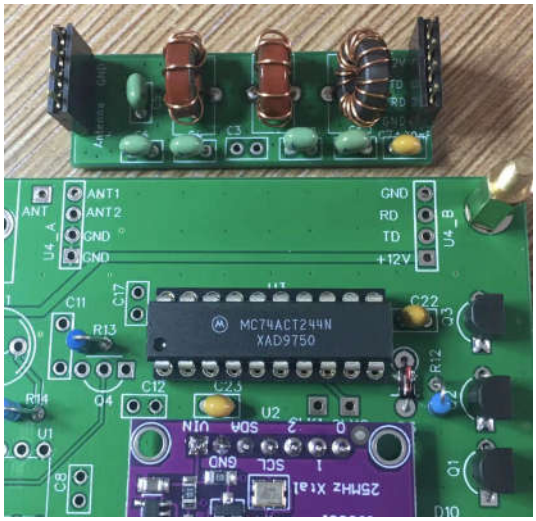


Populate Q1-Q3 – BS170 and solder all the leads but the center leads. Don't touch the leads with your hands as they are static sensitive.



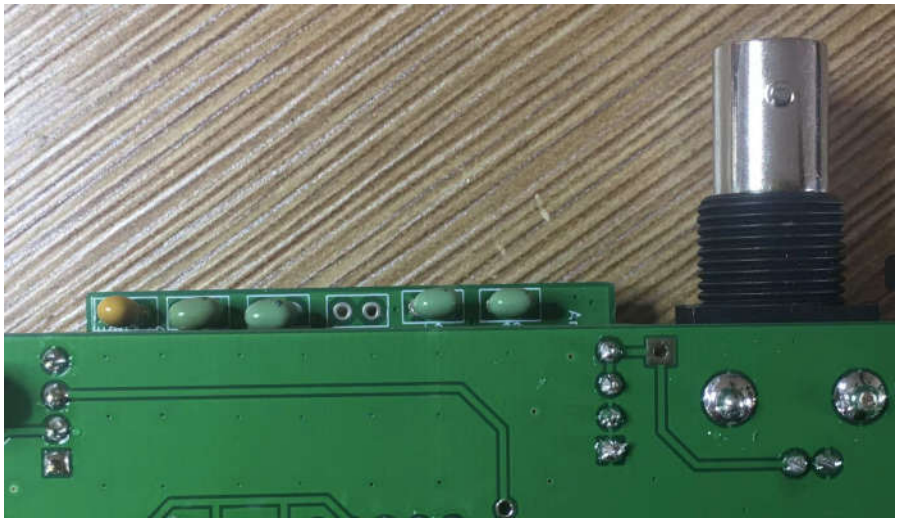
Make sure your solder iron is well grounded. If not, you can unplug the power of your solder iron and solder the 3 center leads.

Suppose you already have the LPF band filter board ready. It is recommended you plug in the pin header sockets on the LPF band filter board first to make sure they are matched.



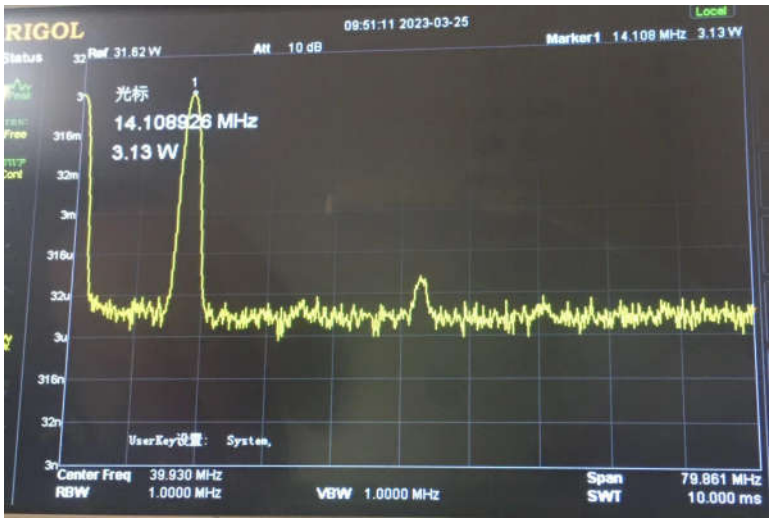
Plug the pin header sockets to the main board and solder.

Populate J1-BNC in place and solder. Please make sure it is not floating as it sets the height limitation of the board.



Now you have done the assembly of the TX portion. Before the TX test, please make sure the LPF band filter module and the band setting of Nano is matched. If not, please press and hold SW1-UP(BAND) button and power on so you can adjust. To adjust the band, you can press

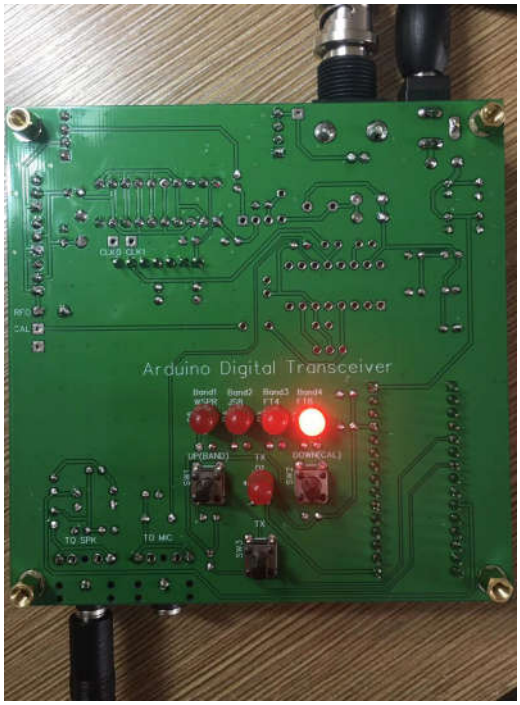
SW1-UP(BAND) or SW2-DOWN(CAL). Remember the default setting is B1 – 40, B2 – 20, B3 – 15, B4 – 10. Press SW3-TX to confirm the band setting. Now let's connect the TX to a RF power meter or a spectrum analyzer if you have. Make sure you have proper 50 ohm load for the TX. In my case, I have a -45dB power attenuator inserted between the TX under test and the spectrum analyzer, so I can measure the RF output and spurious by pressing the SW3-TX button. It looks good that the RF output is 3.13W and the spurious is -50dBc or so. The RF output varies with the voltage of the power supply. If your spectrum analyzer reads 35dBm or around, it equals to 3W or around on a 50 ohm load. Please double check if your overall current is around 600mA.



Step 9: Audio Jacks

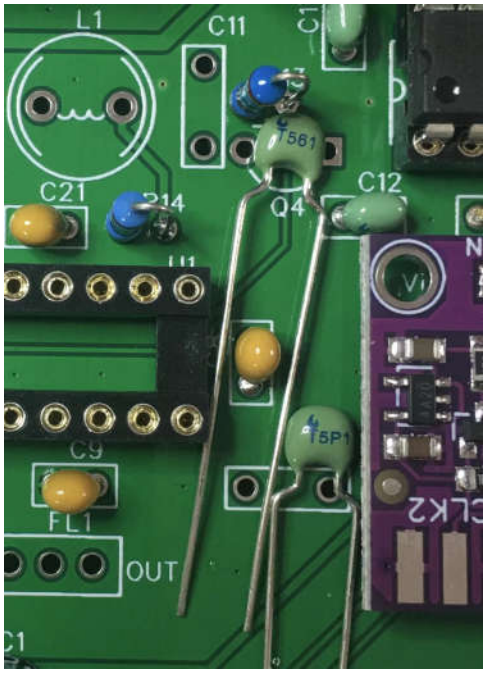
Populate C3, C6-100nF, C4, C5-10nF, C7-1uF, CON1-TO SPK and CON2-TO MIC, and solder.

Use an audio signal generator like Sonic Tools App on iPhone to inject audio to CON1-TO SPK jack. Turn the volume to MAX and it should trigger TX, and the TX LED should be on. When you perform this test, please make sure the 50 ohm load is still connected to J1-BNC.

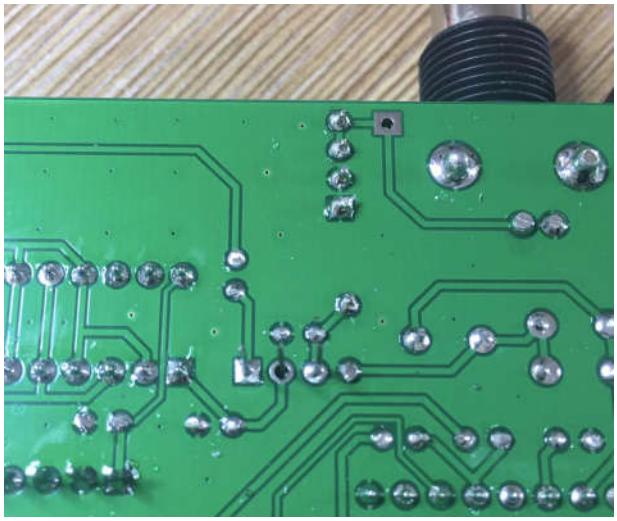


Step 10: CD2003 RX

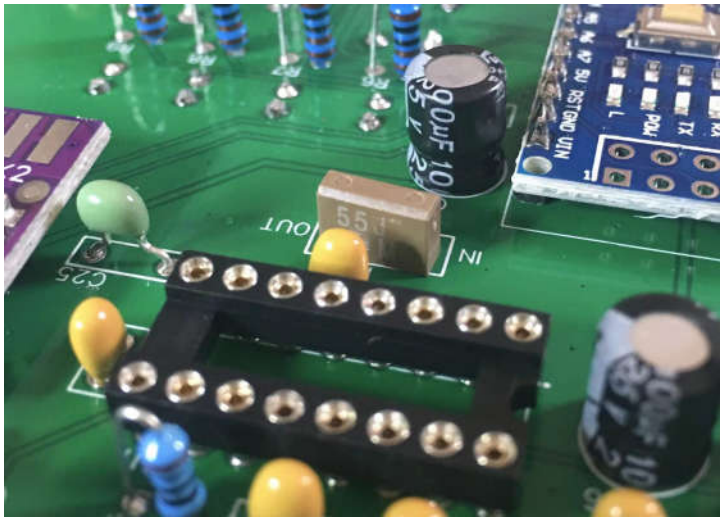
The photo shows C11-560pF, 620pF or 680pF and C25-5.1pF. They look quite similar.



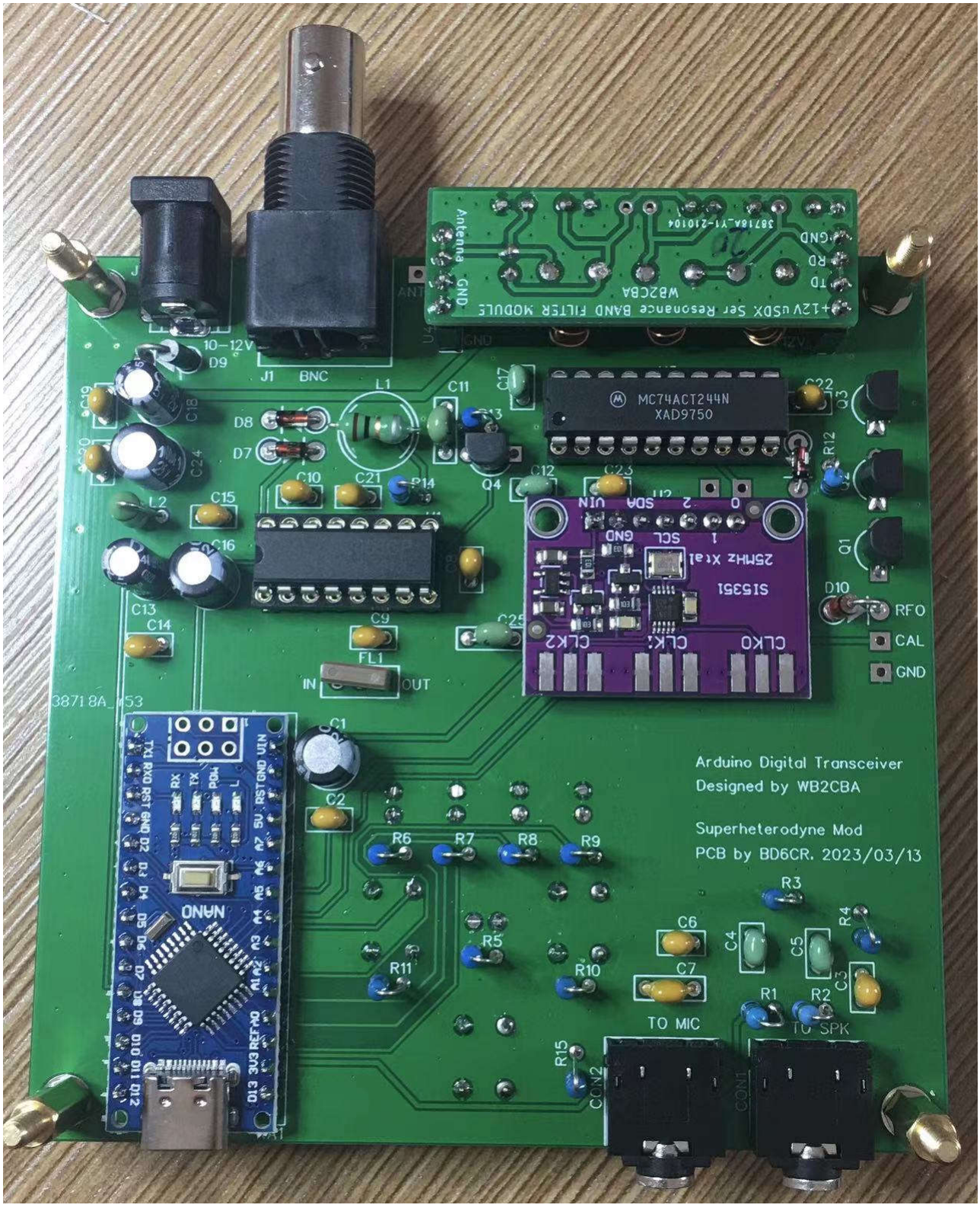
The photo shows the center pin of Q4-BS170 to make sure it is safe from static.



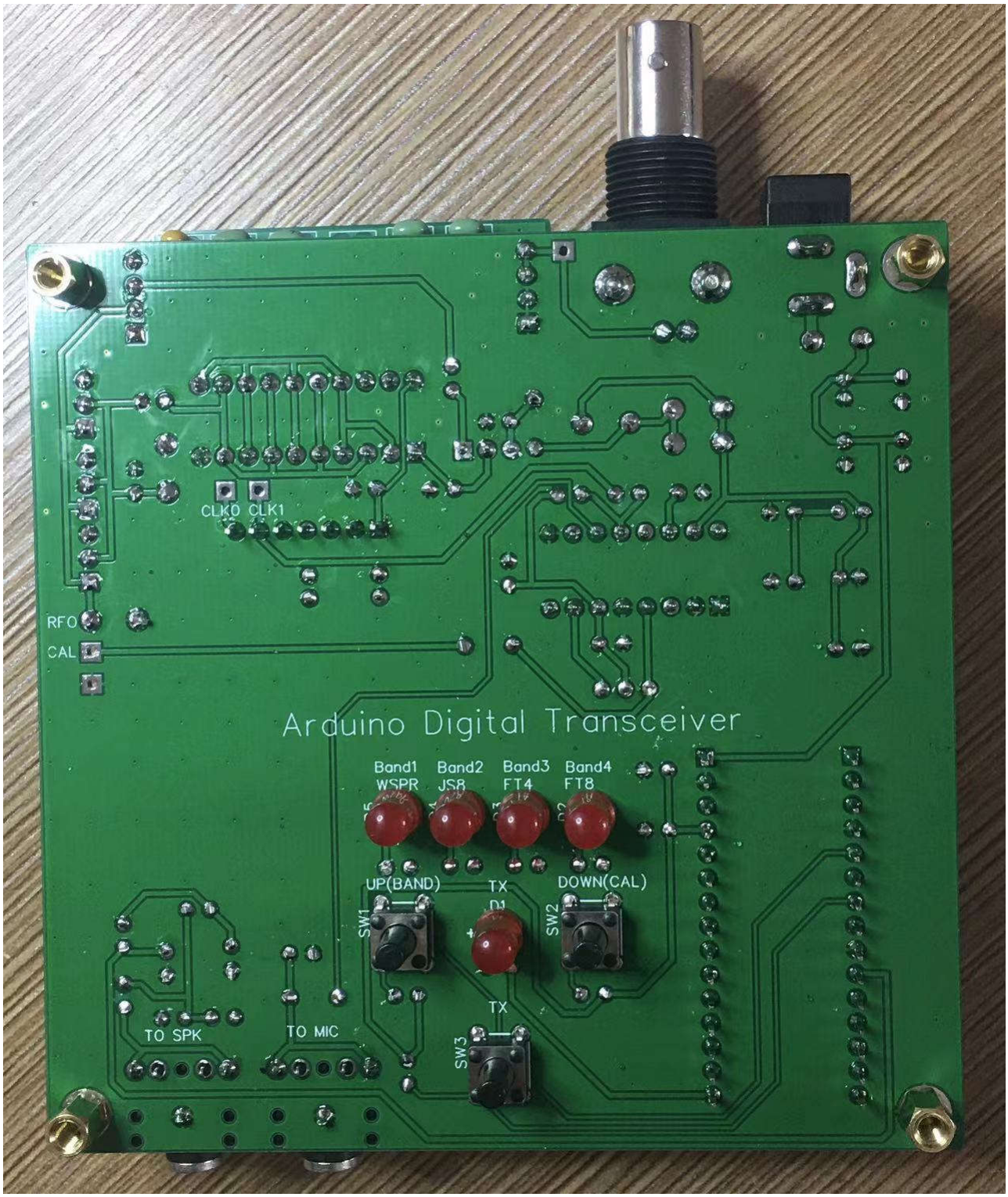
The photo shows the marking of FL1 is facing CD2003.



Populate all the parts L1-1uH, Q4-BS170, FL1-PFB455JR, U1-CD2003, C8,C9,C10,C21-100nF, C12,C17-10nF, C11-560pF or 620pF or 680pF, and C25-5.1pF and solder to complete the soldering of the main board.


Arduino Digital Transceiver
Designed by WB2CBA
Superheterodyne Mod
PCB by BD6CR. 2023/03/13
MC74ACT244N
XAD9750
SI5351
25MHz Xtal
+12V uSDX Ser Resonance BAND FILTER MODULE
WB2CBA
Antenna
GND
RD
TD
J1
BNC
10-12V
D9
D8
D7
L1
L2
C11
C10
C21
C15
C16
C13
C14
C9
FL1
IN
OUT
C1
C2
C6
C7
C4
C5
C3
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R15
TO MIC
TO SPK
CON2
CON1
D10
RFO
CAL
GND
Q1
Q3
Q4
R12
C22
C12
C23
C25
C17
C18
C19
C24
C30
NANO

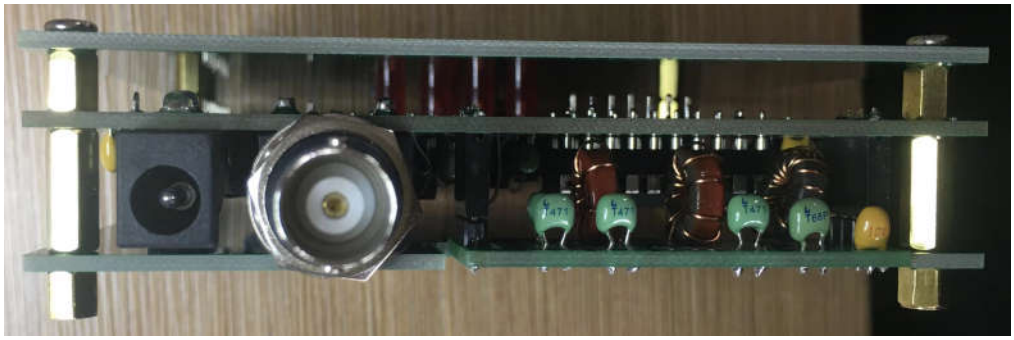
And the solder side.



Now you can put on the top cover and fasten the screws. Put on the caps of the switches. Red for SW3-TX.

And put on the bottom cover and fasten with M3x5 standoffs.

Check out the stack of the sandwich and the standoffs to make sure they are right. If you identify the risks of potential short circuit, please correct them before you apply power supply.

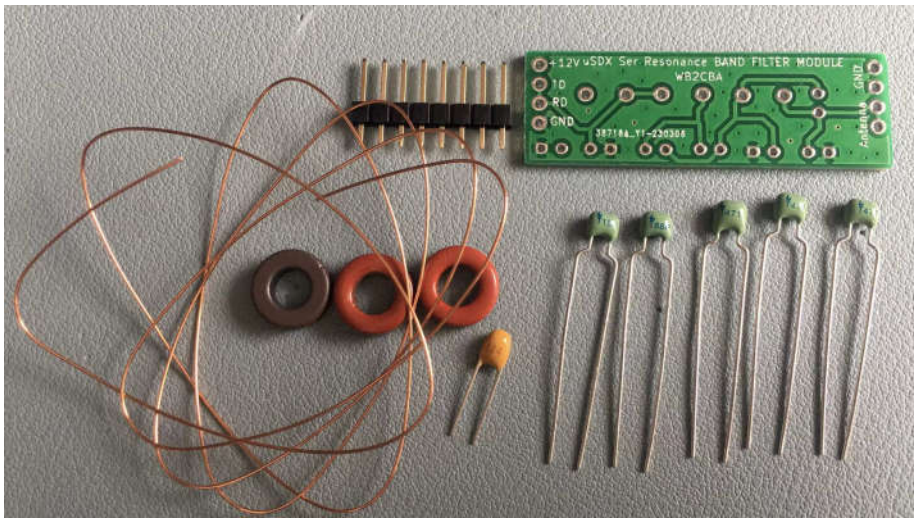


Now, the main board assembly is completed.

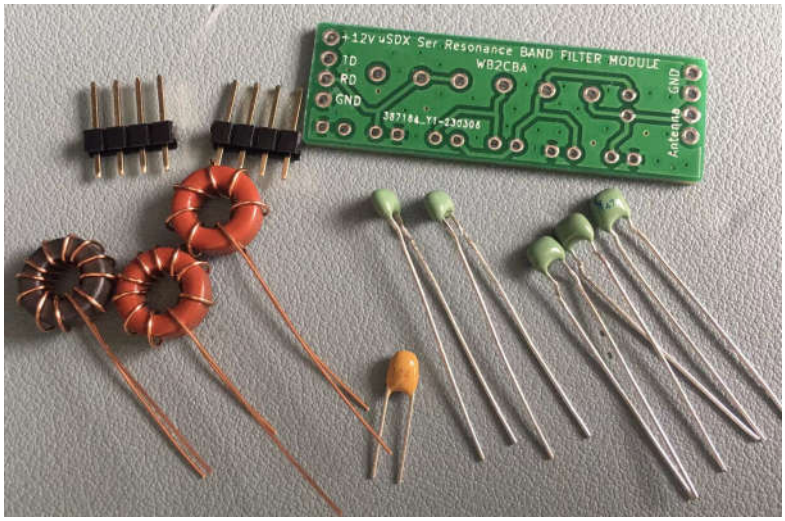
WB2CBA & BD6CR
ANTENNA DC 10–12V
ADX–S
Arduino Digital Modes HF QRP Transceiver
B1
B2
B3
B4
WSPR
JS8
FT4
FT8
(BAND)<---
TX
--->(CAL)
SPK
MIC
Arduino USB
3871 8A_Y50

Step 11: LPF band filter board

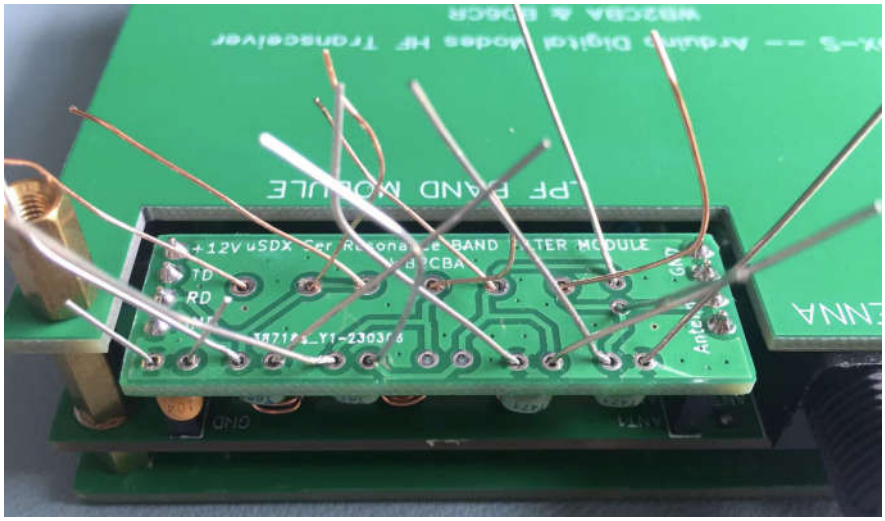
Let's take 20m LPF band filter board as an example. You have parts like this. For 10 and 15 meters, you will have yellow toroids instead of red.



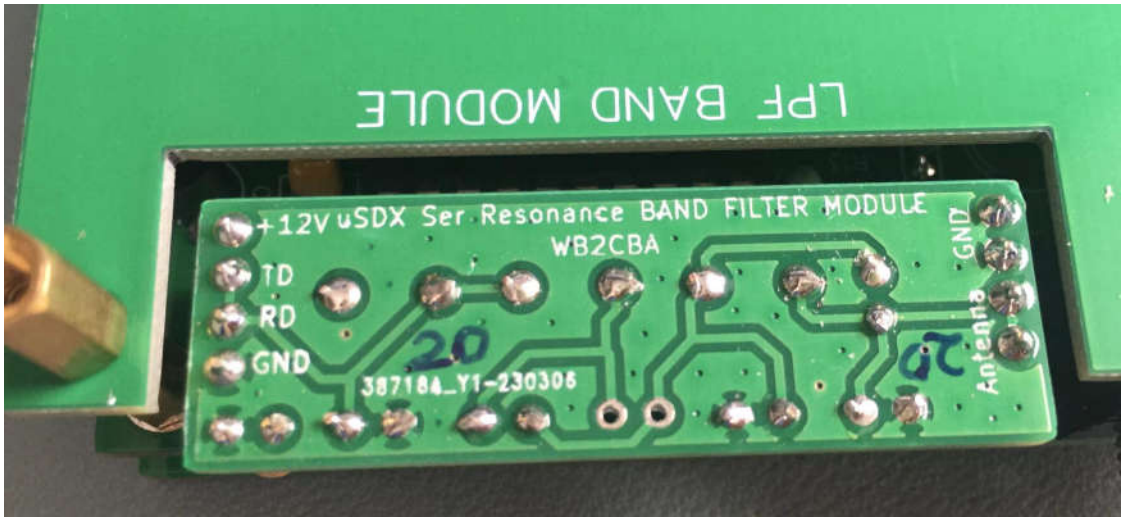
Wind the toroids based on the BOM, and prepare the pin headers and capacitor leads. The black toroid has 12 turns, and the red toroids have 10 turns and 7 turns. Remove the enamels of the wires ends before soldering. Some people use sandpapers. Wire length: 1.5 cm per turn plus 5 cm. For 12 turns, the length is 23 cm.



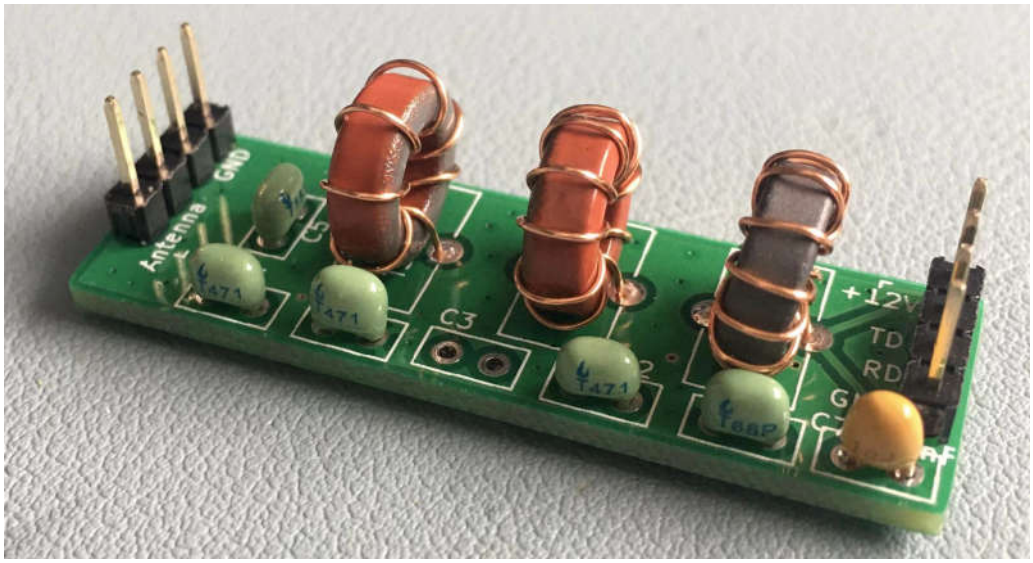
It is recommended you use the sockets on the main board as support. Populate the parts correctly and solder. Always remember, C7-100nF is near to the black toroid, and the red toroid with more turns is in the middle of 3 toroids. Some schematic and BOM of the original design ADX does not match the PCB very well, so please double check the ADX-S BOM.



Use a marker pen or label to mark the LPF band filter board.



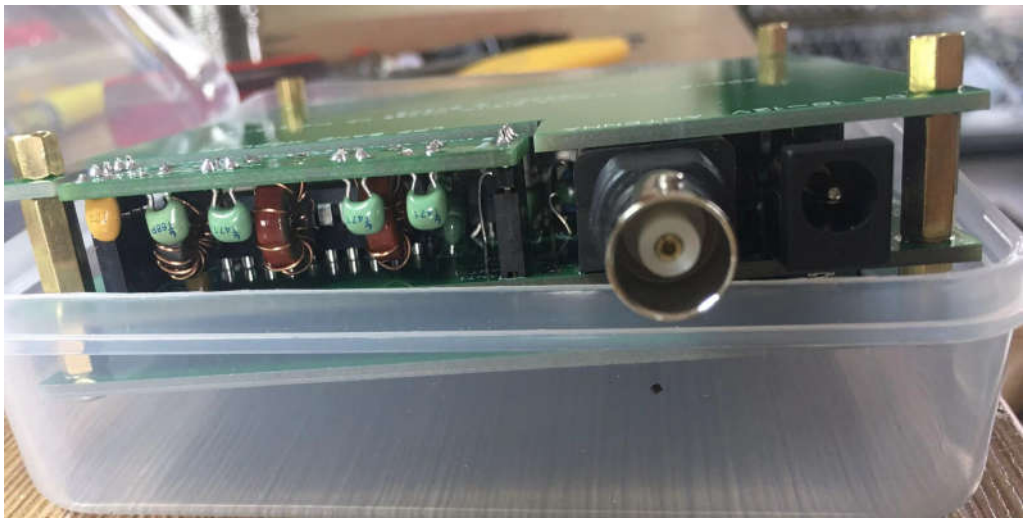
The completed 20m LPF band filter board looks like this.



Step 12: Box the ADX-S

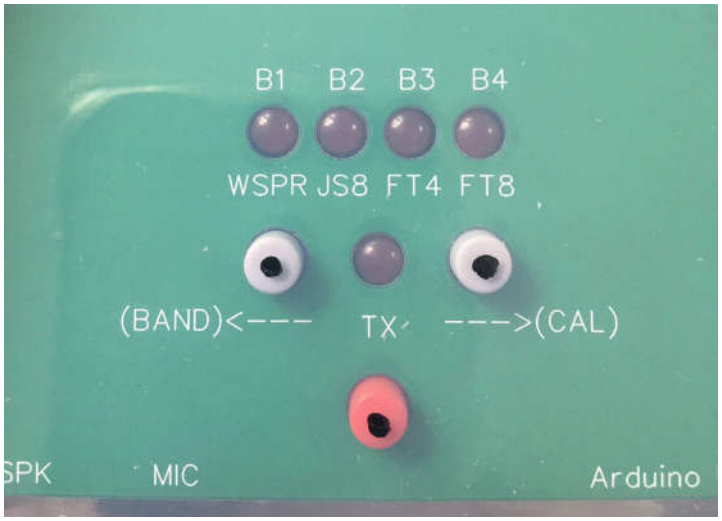
If your plastic container looks still nice after shipping, you can consider using it as the chassis, although the sandwich PCBs look great too. If your container is broken, please don't hate us as it is not even included in the BOM.

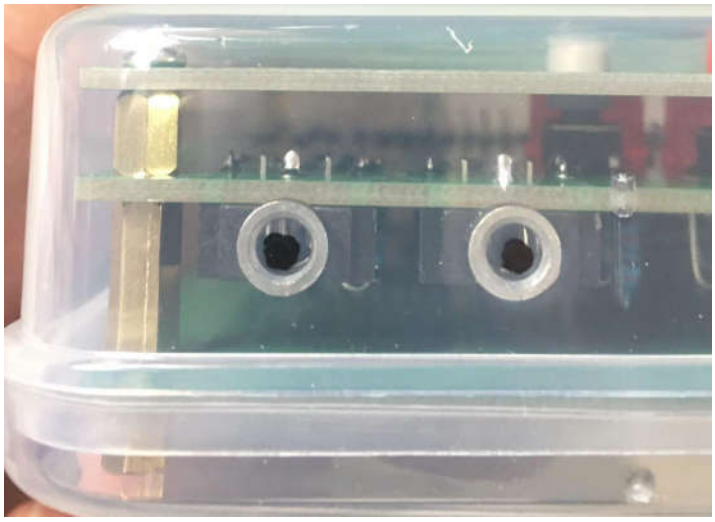
Align the BNC hole location and mark it with a dot. The belt line of the container defines the upper limit of the BNC hole. It is recommended you install the ADX-S boards upside down so you can easily swap the LPF band filter board.

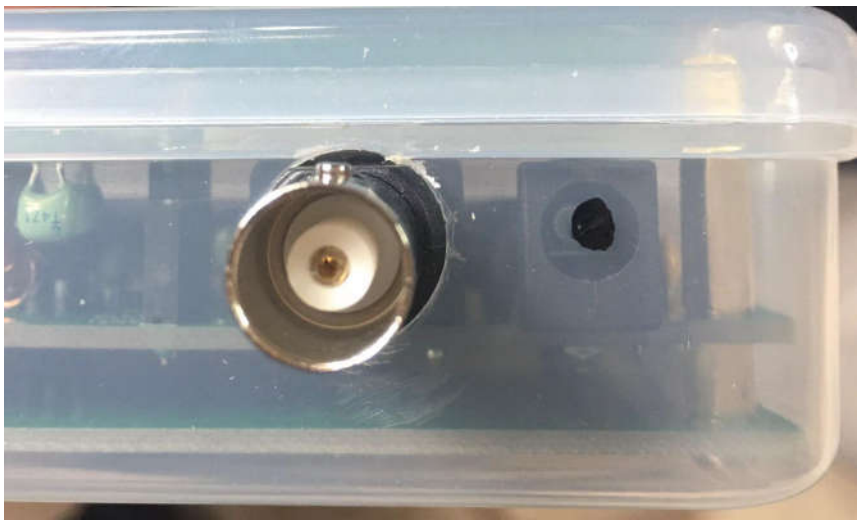


Drill a small hole, enlarge it gradually so you can protrude the BNC connector. The electric drill sold by IKEA for screw fastening will do the job here.

Press the board down so you can get correct locations for audio jacks, power supply jack and USB port on container sides, or the buttons will lift the board. Now mark the locations.







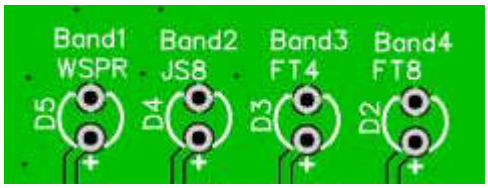
You can mark the USB port location with a short line, drill two holes, enlarge them and join them. The finished chassis looks like this.

WB2CBA & BD6CR
ANTENNA DC 10–12V
ADX–S
Arduino Digital Modes HF QRP Transceiver
05
PP
B1 B2 B3 B4
WSPR JS8 FT4 FT8
(BAND)<---
TX
--->(CAL)
38718A Y50
SPK
MIC
Arduino USB

Settings and Calibration

There are two settings: BAND setting and MODE setting. Please note, BAND setting is a design change from the original design ADX.

Before any TX, please make sure the LPF band filter module and the band setting of Nano is matched. To adjust the BAND setting, please press and hold SW1-UP(BAND) button and power on. After LED flashes are done, you can press SW1-UP(BAND) or SW2-DOWN(CAL) to adjust. Remember the default setting is Band1 – 40, Band2 – 20, Band3 – 15, Band4 – 10. Press SW3-TX to confirm the band setting.



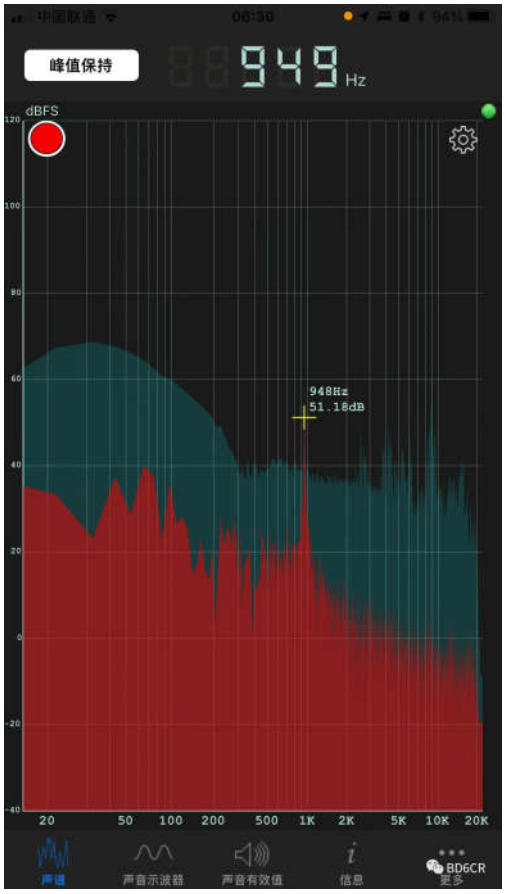
MODE setting can be made during normal working mode. The LED shows the current MODE, from left to right: WSPR, JS8, FT4 and FT8. You also change frequency when you change MODE. The combination of the BAND and MODE setting defines the carrier frequency. For example, 20m band and FT8 mode defines the carrier frequency 14.074MHz. The frequency table in the firmware is as follows.

(kHz)	40 meter	20 meter	15 meter	10 meter
WSPR	7038.6	14095.6	21094.6	28124.6
JS8	7078	14078	21078	28078
FT4	7047.5	14080	21140	28180
FT8	7074	14074	21074	28074

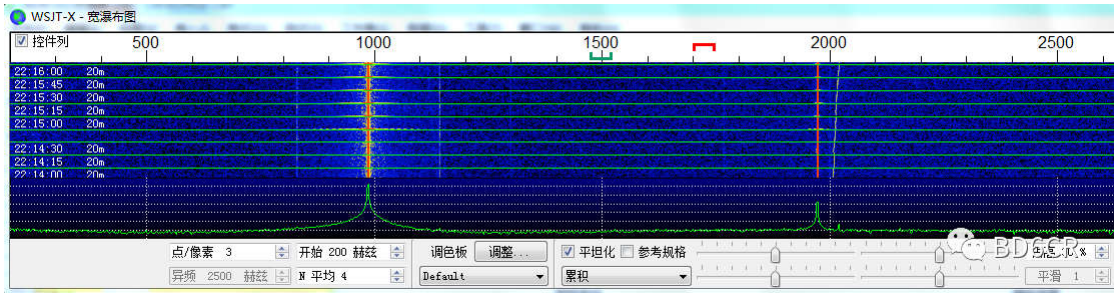
The necessary calibration is the frequency calibration. The oscillating frequency of the 25MHz crystal on the 5351 module has tolerance, so you can press and hold SW2-DOWN(CAL) button and power on to enter the calibration mode in which both B1 and B4 LEDs are lit. You need to wait for at least 2 minutes to warm up. The standard calibration method is that you probe the test point CAL with a frequency counter of 1Hz resolution, press SW1-UP(BAND) or SW2-DOWN(CAL) to adjust to as close to 1MHz as possible, then press SW3-TX to save the setting. You can exit the calibration mode only by recycling the power.

If you don't have a frequency counter, you can consider a commercial ham radio transceiver with frequency accuracy. It is advised to choose 10 meter band to perform the frequency calibration, so the frequency of the lower bands will be accurate automatically. Let's say we choose FT8 mode on 10 meter band, so you can tune to 28074kHz USB to monitor. Please make sure you have inserted 10 meter LPF band filter module, and set band correctly. Connect a 50 ohm dummy load of at least 5 watts or a resonant antenna to the BNC connector. Press SW3-TX button, and you should hear a tone of about 1kHz. Use an audio spectrum analyzer app, like Sonic Tools for iPhone, to measure the frequency of the tone. If you use an audio cable, you can download from

http://www.qsl.net/zl1an/Software/Spectrum3.zip and run on PC. Enter the calibration mode at power up, and adjust 10 steps in one direction, press SW3-TX to save the setting, recycle the power and redo the frequency measurement. You will need a few steps to calibrate the frequency to accurate, so you will hear exact 1kHz tone.



If you have access to the tinySA, you can use the signal generator function to inject 28075kHz, -85dBm together with a fixed attenuator (-30 or -40dB) in between, and monitor the receiver spectrum of ADX-S with WSJT-x or JTDX. Do the calibration at power up, until you see spectrum like this.



If you have access to the nanoVNA, you can carefully adjust the LPF band filter module, so the RF output is maximized and the harmonics are minimized. The default is not bad, so it is optional only for the experienced builders.

Use the ADX-S

You can change band by inserting the correct LPF band module to ADX-S first, then setting by the BAND button at power on. It is critical to insert the LPF band module in the correct orientation, especially when the bottom cover is not yet installed, or the board could get damaged. The capacitors on the LPF band module should be on the outer side of the main board. After properly inserting the band module, you will need to press and hold the BAND button and power on, wait for a while until you see the current band setting, switch the band by UP or DOWN key, and confirm the setting by TX button.

Please power the ADX-S with 10-12V. The maximum is 3 Li-Ion battery in series or 12.6V. The RF output is in QRP range. It draws about 400-600mA. QRP power is good for JS8, FT4 and FT8, but not quite good for WSPR, as the heat generated by transmitter might cause too much frequency drift.

You can switch to QRPP range by powering by the USB port at 5V. Never power by both 12V and USB ports, or the Arduino Nano module will get damaged. The output is about 1/10 of the QRP power. Less heat is generated in QRPP and much more stable for WSPR. Some people also use QRPP to pursue more challenging DX communication and it is a lot of fun.

Because ADX-S has built in VOX code, only two audio cables are required to operate. One audio cable is connected between the jack marked TO MIC or MIC on the ADX-S and the MIC jack on your sound card. The other audio cable is connected between the jack marked TO SPK or SPK on the ADX-S and the PHONE or SPEAKER jack on your sound card. If you swap them, you will see strange lines on the spectrum and the audio output cannot trigger transmitter. The ADX-S prefers higher audio input level, and you don't have to worry about over modulation. If the

ADX-S audio output is too low for your sound card, you can adjust the MIC gain if your system supports. Take WSJT-x as example, you need to set **Radio as None**, **PTT method as VOX**. In the main window, you need to manually choose the correct band, frequency and mode, for correct logging purpose.

In setting window, choose Radio as None, PTT method as VOX, and click OK to confirm

In main window, manually choose band the mode to display the correct working frequency. For example, if you choose WSPR and 20m band, the frequency will be 14.095600 and it is the same as the setting of the ADX-S firmware

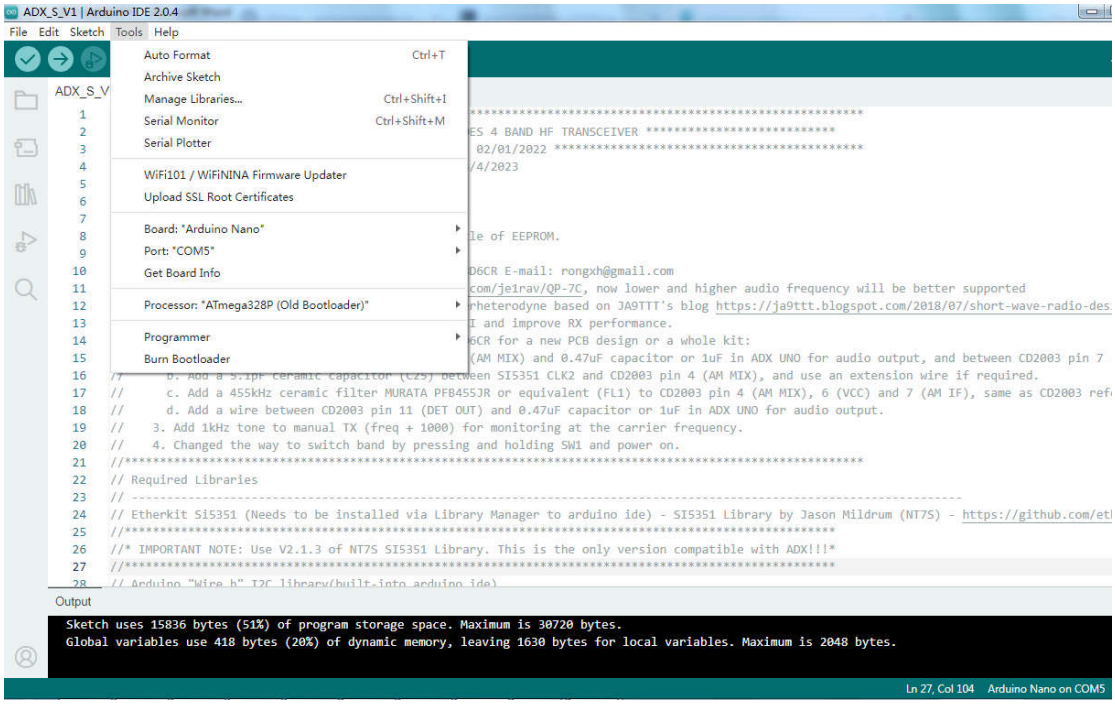
To make a successful QSO, you will need to calibrate time and frequency. Type time.is in your browser to make sure your computer time is accurate (1 sec accuracy is fine). Frequency calibration is recommended before you put the ADX-S in actual QSO. Since the power level is QRP, you can start with calling a strong station, at least -5dB SNR. Later, you can try to call weaker stations. It is not difficult to achieve DX contacts if you have a full sized antenna.

If your receiver suffers from strong broadcast interference, you can try a tuned antenna like magnet loop antenna or add a Z-match ATU in between the ADX-S and the antenna. If it does not help, you can temporarily add a fixed attenuator, say -20dB, in between the ADX-S and the antenna to see if it helps. If it does help, you can consider permanently modifying the receiver front end to add the attenuator, and add a switch to enable or disable it.

Open Source Code Modification and Update

The Arduino code ADX_S_Vx.ino is open source, so you can modify and compile by yourself. Normally the Arduino Nano module is pre-programmed and the B1-B4 setting is 40, 20, 15 and 10m bands. If you wish to browse the code and customize your own, you will need to download the Arduino IDE from https://www.arduino.cc/en/software

Remove the 12V power supply first, then connect the Arduino Nano module to the USB port of your computer by your TypeC USB cable. The board can be powered by your USB port. You will choose Arduino Nano in board manager of the Arduino IDE, choose the correct port like COM5, then choose ATmega328P(Old Bootloader) in processor if your Nano module is blue. After loading the code to the Arduino IDE, you can compile and upload the code to the Nano module. If you get an error message saying the 5351 library is missing, you will need to load the library Si5351Arduino-master through menu Sketch – Include Library – Add .ZIP Library. You need to download both the source code and library file in the same place to avoid compatibility issues.



Since the ceramic filter has tolerance on bandwidth and center frequency, it is better that you fine tune your own beat frequency, so you can get higher SNR and less interference. In order to do so, you do a search in the source code and you will find two different beat frequencies. One is 464570, the other is 447430. For now, 447430 is only for 20 meter band. Adjust them at 500-1kHz step and compare with the default setting, so you will get the best value for your kit. This is the fun part of the open source kit. **Optional Modification**: If you have a 3pF capacitor, you can replace the 5.1pF and tune to get the best. Normally, the beat frequencies for 3pF should be closer to 455kHz.

SPECIFICATIONS

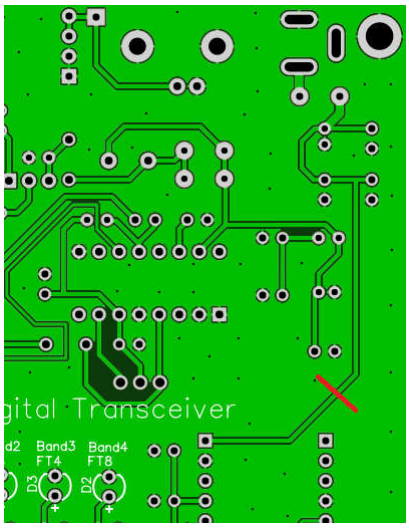
Part Number	3dB Band Width (kHz)	Selectivity +9kHz off (dB) min.	Insertion Loss (dB) max.	Composition
PFB455JR▼	5.5 ± 1.5	17 (23)	6 (3)	2 Elements

CAT Support

CAT is the abbreviation of Computer Aided Tuning. With this function, the PC software can automatically tune the radio. Although the shipping firmware in the Arduino module might not directly support CAT, you can re-program V1.2 firmware or later (Download the source code from https://groups.io/g/crkits/files/ADX-S) to try CAT. The CAT commands support is limited, but it should work with WSJT-x or JTDX. The implementation of CAT is well considered. The ADX-S will stay in the original normal mode if CAT commands are not received. Once CAT commands are received, the ADX-S will turn on both B2 and B3 LED's to indicate CAT mode. Now, the buttons are disabled and you can only control by CAT unless you recycle the power.

The firmware detects the CAT frequency and compares with the band set manually. If they are different band, it can receive but cannot transmit. This is a newly added protection to the finals.

Again, never power by both 12V and USB ports, or the Arduino Nano module will get damaged. Since USB port is always in use in CAT mode, either you can live with QRPP with USB 5V, or you can consider cutting a PCB trace so you can power by both 12V and USB ports and transmit at 3-5W QRP level.



WSJT-x or JTDX radio settings: Kenwood TS-2000, Correct Serial Port, Baud Rate 115200, CAT, others are default. Press Test CAT to confirm it is working. You still need audio to trigger TX, so Test PTT should not work.

Known frequency sync issue: You might see the red background in frequency display area to show out of sync at the beginning. You can switch MODE and switch back to sync.

If the CAT is disconnected when you transmit, it might be RFI and you can consider using some ferrite cores on the USB cable, or move away from the antenna.

Normally, you don't need to calibrate the frequency again after re-programming the Arduino.

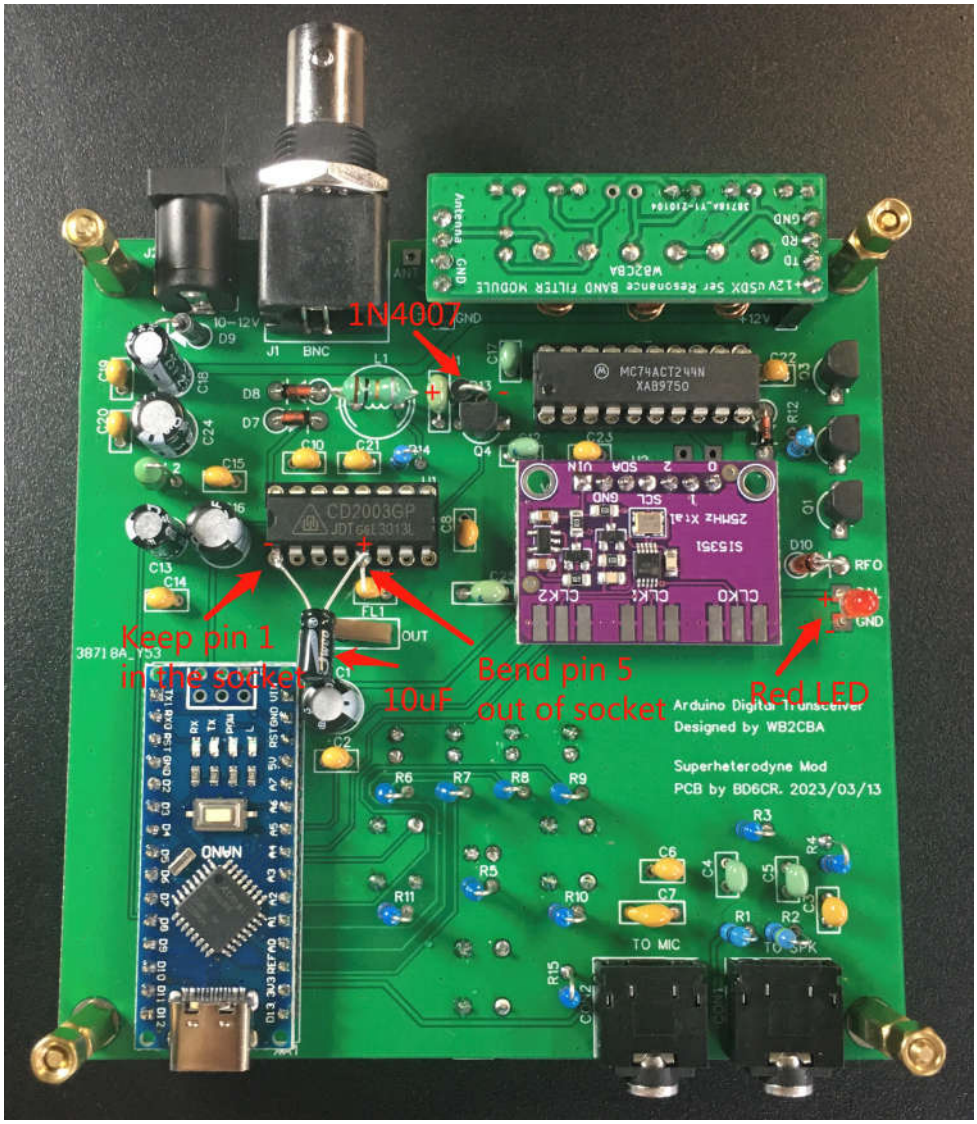
Modifications

It is recommended that you apply the following receiver modifications to improve decoding while building. You can also do it after your assembly. The benefit of the modifications:

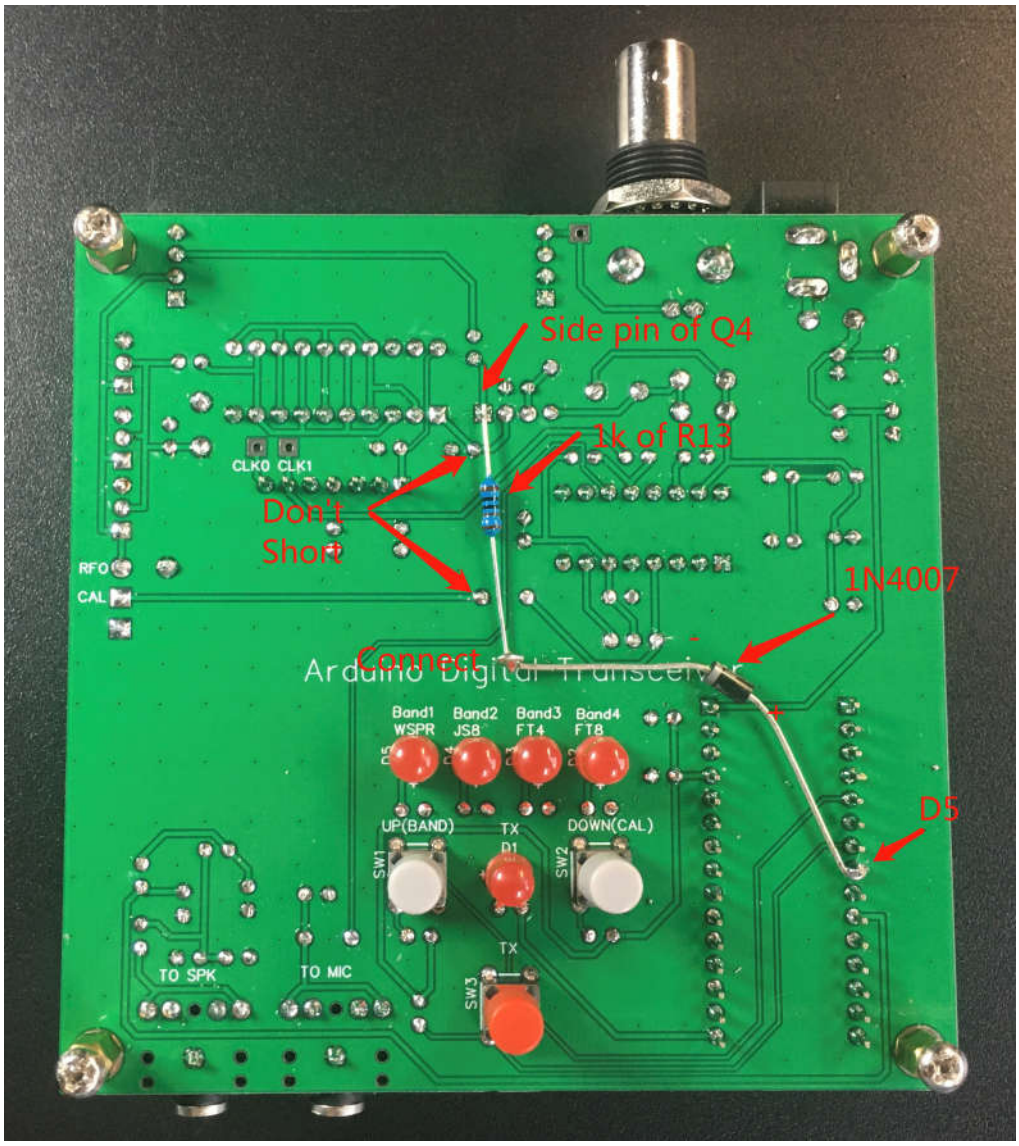
- Add a button controlled fixed attenuator in the RX front end
- Enable AGC of CD2003
- Reduce the amplitude of BFO and change firmware settings to improve the RX sensitivity
- You can also listen to AM broadcast in CAT mode (high impedance headphone is required)

The part list for the modification:

- 2 pcs 1N4007 diode
- 1 pcs 10uF capacitor
- 1 pcs red LED (3mm and 5mm are both okay)



The red LED will reduce the amplitude of CLK2 (BFO) and also indicate the existence of CLK2 (BFO). The 10uF will enable AGC of CD2003. The 1N4007 in place of R13 is used as an attenuator in the RX front end.



The 1k resistor is originally used for R13. Once you press both UP and DOWN buttons, the firmware will toggle D5 pin and it will act as a switch for ATT.

You will need to update firmware to V1.2RX or later to enable all the features. Download the firmware from https://groups.io/g/crkits/files/ADX-S and update by Arduino IDE (see above)

How to use:

	SW1 (UP)	SW2 (DOWN)	SW3 (TX)	Both SW1 and SW2
Normal Mode	Mode change	Mode change	TX	ATT toggle
CAT Mode	-5kHz	+5kHz	AM mode toggle (See the red LED added in this mod to tell if the BFO is on or off)	ATT toggle

You will need to add your AM broadcast frequencies into the frequency list of your WSJT-x.

From 4 to 7-Band Upgrade

The standard 4 bands are 40, 20, 15 and 10m. Now we are offering an upgrade kit to add 30, 17 and 12m. The firmware needs to be updated to V1.3 or higher. In the new firmware, you need to add the receiver modification as well, as explained in the previous chapter, or the receiver sensitivity will be degraded. Fortunately, the upgrade kit will include the modification parts.

The frequency table in the firmware is updated as follows.

(kHz)	40 m	30 m	20 m	17 m	15 m	12 m	10 m
WSPR	7038.6	10138.7	14095.6	18104.6	21094.6	24924.6	28124.6
JS8	7078	10130	14078	18104	21078	24922	28078
FT4	7047.5	10140	14080	18104	21140	24919	28180
FT8	7074	10136	14074	18100	21074	24915	28074

The blinking LEDs at power up represent bands like this.
B1 – 40m No change
B1&B2 – 30m New
B2 – 20m No change
B2&B3 – 17m New
B3 – 15m No change
B3&B4 – 12m New
B4 – 10m No change

The BOM of the band filters are updated as follows.

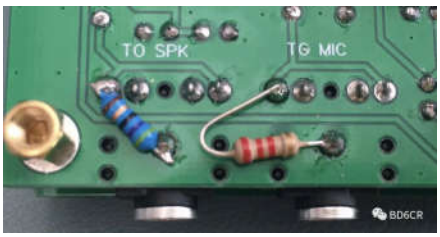
LPF BAND MODULE										
Band	C1	C2	C3	C4	C5	C6	C7	L1	L2	L3
40m	120+10	470	470	1000	300	1000	100nF	12T/BLACK	15T/RED	10T/RED
30m	100	330	330	680	220	680	100nF	12T/BLACK	12T/RED	8T/RED
20m	68	470		470	150	470	100nF	12T/BLACK	10T/RED	7T/RED
17m	91	360		360	120	360	100nF	12T/BLACK	11T/YELLOW	7T/YELLOW
15m	68	300	15	300	100	300	100nF	12T/BLACK	11T/YELLOW	7T/YELLOW
12m	47	270		270	62	270	100nF	12T/BLACK	10T/YELLOW	6T/YELLOW
10m	39	240		240	75	240	100nF	12T/BLACK	9T/YELLOW	6T/YELLOW
Unit: pF by default										

Frequently Asked Questions and Answers

1. **Can I add sockets to the Arduino Nano module and SI5351 module?**
   a) Yes. If you use sockets, please change the length of the standoffs, and you will need to find another case instead of the plastic case.

2. **I am afraid the Arduino Nano module is damaged, what should I do?**
   a) Try to reflash the firmware.
   b) Determine if it only fails while at 12V. If it is true, the on board 5V regulator is damaged, you can consider adding an external 5V regulator. More details can be found on the wechat blog in Chinese https://mp.weixin.qq.com/s/wO0uJCJHtiufBroOgAXmQQ
   c) Find another Arduino Nano module (any version will do) and flash the firmware, desolder from the Arduino Nano module side instead of the main PCB side to remove the damaged module, and replace it. If you need the pre-programmed Arduino Nano module, contact your seller directly.

3. **Can I directly connect the ADX-S to the audio connector of an Android phone or tablet?**
   a) Yes, FT8CN App is very popular, but you will need some modifications to the main PCB. The major problem you might have is the Android device cannot detect the audio connection as a microphone. If it happens, please add a 2.2k ohm resistor to the MIC connector on the main board side. Also, it is recommended to add a 33 – 100 ohm resistor to the SPK connector on the main board side, as shown. More details can be found on the wechat blog in Chinese
   https://mp.weixin.qq.com/s/h7nkQAuqB0pQcVu1M0ixhg



4. **The receiver decodes much less stations than my other radios. How can I improve?**
   a) If you use a regular antenna at home or in the field, most likely you will suffer strong interference due the broad band design in the receiver front end. The recommended solution is you add the RX modification and enable the ATT by pressing both left and right buttons.

5. **How can I receive Engineering Change Notice?**
   a) Please monitor the posts in the group and the web page. So far, the only engineering change is https://groups.io/g/crkits/message/4639

Further readings

This document will be updated in the folder, together with other documents like schematic, part list and source code. If you are not yet a member, please subscribe first.
https://groups.io/g/crkits/files/ADX-S
or, scan this QR code with your cell phone.

Reference and Acknowledgement

Barb, WB2CBA
https://antrak.org.tr/blog/adx-arduino-digital-transceiver/
https://github.com/WB2CBA/ADX

Burkhard Kainka, DK7JD
https://www.elektronik-labor.de/HF/FT8QRP.html
http://elektronik-labor.de/HF/SDRtxFSK2.html

JE1RAV
https://github.com/je1rav/QP-7C

JA9TTT
https://ja9ttt.blogspot.com/2018/07/short-wave-radio-design-2.html

QRPGUYS
https://qrpguys.com/qrpguys-digital-fsk-transceiver-iii

Jason Mildrum, NT7S
https://github.com/etherkit/Si5351Arduino

Peter Parker, VK3YE
https://www.youtube.com/watch?v=FGH0kUvkqX0

BD4AHS, BG7IKK, BH3RQW, BD6EC, BG7IBQ, BD8TE, BG8NSK, BA7MPY, BI4TMT, BA1RB and more…