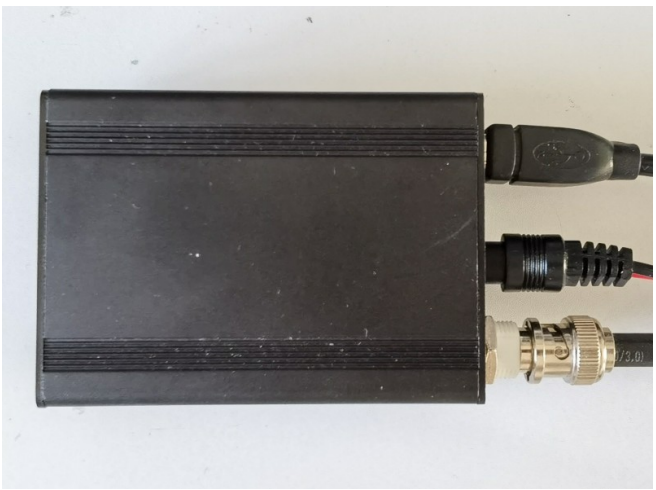
QRP Labs

QDX-M

QDX: QRP Labs Digital Xcvr - Monoband

Assembly, design and performance manual

--- PCB Rev 5M ---



Contents

1. Introduction

The QDX-M is a high performance, single-band 5W Digital modes transceiver with CAT control and built-in USB sound card. QRP Labs presents QDX-M, a digital transceiver with a ratio of performance to price not available until now.

- 5 W from 9 – 10V supply
- Clean single signal output (zero residual carrier, zero unwanted sideband)
- Solid state transmit/receive switching
- High performance embedded-SDR SSB receiver using 112dB 24-bit stereo ADC chip
- Built-in USB sound card: 48ksps 24-bit stereo
- Built in USB Virtual COM port serial for CAT control
- Si5351A Synthesized local oscillator with better than 0.001Hz resolution and high precision 25MHz TCXO reference as standard
- Built-in signal generator
- Built-in suite of configuration and analysis tools
- Lifetime free firmware upgrades with QRP Labs Firmware Update (QFU) bootloader for easy firmware update on any OS with no extra software, or drivers, or programming hardware
- All SMD components pre-installed by factory, only through-hole component soldering by the kit constructor
- Receive current: 150mA; Transmit current 1.0 – 1.1A (9V supply, 5W output)
- Only four connectors: USB (audio and serial for CAT), Power, PTT (external amp) and RF
- Optional smart aluminium extruded enclosure measuring just 89 x 63 x 25mm

No test equipment is required to build, align and operate this digi modes transceiver. There are no alignment tasks.

We hope you enjoy building and operating this kit! Please read this manual carefully, and follow the instructions step by step in the recommended order. Later in the manual the circuit design is described in detail and we recommend reading and understanding this section too, to get the maximum enjoyment and education from your new radio.

Typical performance measurements are shown in the measurements section.

The operating manual is a separate document and will get you started with QDX-M and your WSJT-X or other digi modes software in minutes.

PLEASE READ THE BASIC ASSEMBLY AND USE INSTRUCTIONS IN THIS MANUAL VERY CAREFULLY BEFORE APPLYING POWER TO THE BOARD!

Additional connections:

The Rev 5M PCB has additional pin header pads on the top right corner of the PCB (as pictured) which provide access to the VGA monitor and PS/2 keyboard ports, and serial port #2. The PTT connection can be re-purposed as serial port #3. Details of these functions are in the operating manual. They're just easier to access now on the Rev 5 PCB than on earlier PCB revisions.

IMPORTANT!

QDX-M can be built for 9V or 12V operation! You need to decide NOW!

Note: higher bands produce lower power output: for example: 3.5-4W would be normal for 10m or 11m.

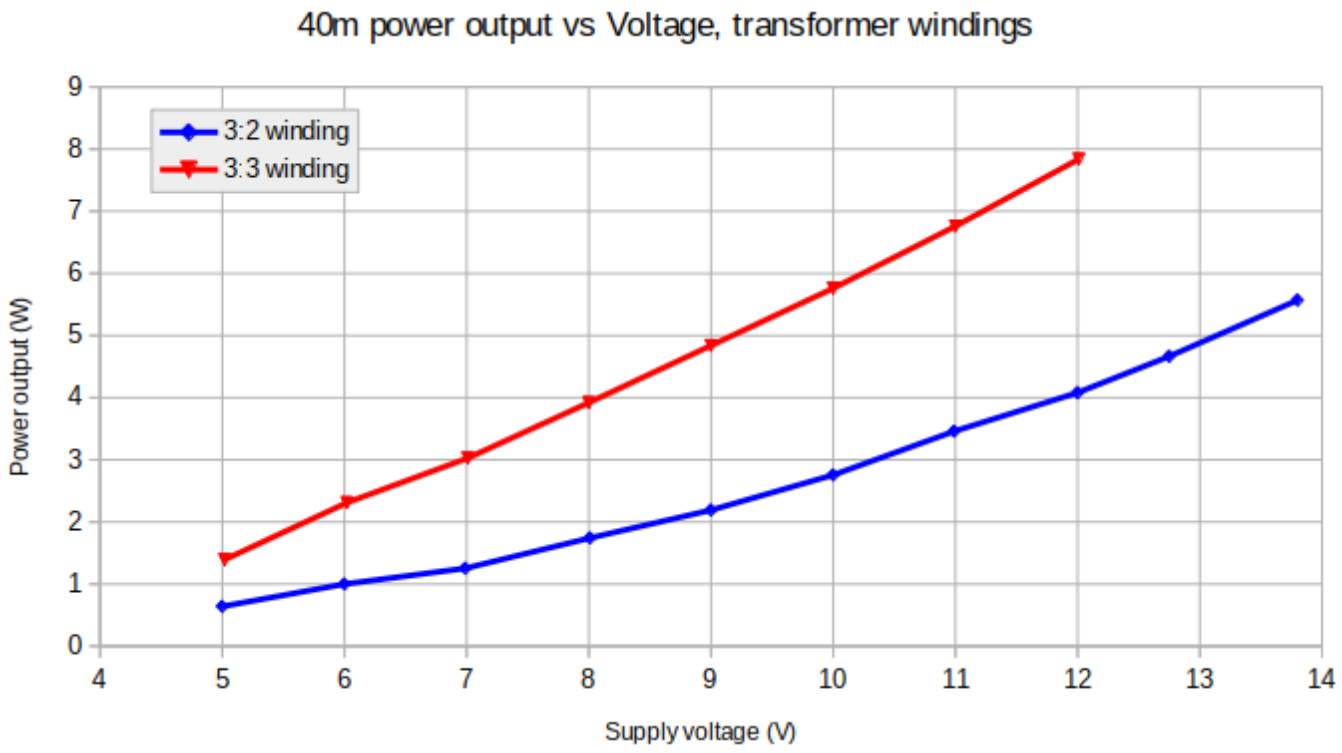
A 9V QDX-M produces 5 W power output from a supply voltage of 9V or a little over. At 12V, a QDX-M built for 9V could be producing 8 W power output which is likely to cause over-heating and perhaps failure of the BS170 final transistors. Do not power a QDX-M with a higher voltage than you built it for.

Operation of QDX-M at more than 5W power output is NOT RECOMMENDED.

If you wish to operate using a 12 V supply, you may use a two turn secondary winding on the output transformer T1, so a 3:2 ratio instead of the 3:3 turns ratio documented in this manual. Remember this when you come to the assembly step for preparing and installing the output transformer T1. The “primary” is still 3 turns, with a tap half way at 1.5 turns. The secondary (no tap) will now be only two turns.

The chart below shows the measured power output vs supply voltage for the standard 3:3 winding (Red line); at 12 V supply the output power of around 8 W is too high and likely to cause over-heating or failure of the power amplifier transistors. If you wish to use a supply of 12 - 13 V the 3:2 winding style is more suitable and will produce 4 – 5 W output for 12 – 13 V supply. The graph shows 40m but other bands are very similar.

Operation at more than 12V will reduce the protection safety margin available in the event of very long key-downs, hot environment, antenna mismatch, cable faults etc. I don’t recommend more than 12V. It isn’t worth pushing more just for a fractioSNR`n of a dB. It is acceptable to use diodes in series on the power line, each one will drop the voltage 0.6V or so. Beware that a 12V nominal batter will potentially be several volts higher when fully charged.



2. Assembly

2.1 General guidelines

Assembly of this kit is quite straightforward, most components are SMD and have already been pre-assembled by the PCB factory. The usual kit-building recommendations apply: work in a well-lit area, with peace and quiet to concentrate. S**ome of the other semiconductors in the kit are sensitive to static discharge. Therefore, observe Electrostatic discharge (ESD) precautions**.

And I say it again: **FOLLOW THE INSTRUCTIONS!!** Don't try to be a hero and do it without instructions!

A jeweler's loupe is really useful for inspecting small components and soldered joints. You'll need a fine-tipped soldering iron too. It is good to get into the habit of inspecting every joint with the magnifying glass or jeweler's loupe (like this one I use), right after soldering. This way you can easily identify any dry joints or solder bridges, before they become a problem later on when you are trying to test the project.

You could also take photos with a mobile phone, and use the phone's zoom features to view the board in detail.

Triple check every component value and location BEFORE soldering the component!

It is easy to put component leads into the wrong holes, so check, check and check again! It is difficult to de-solder and replace components, so it is much better to get them correctly installed the first time. In the event of a mistake, it is always best to detect and correct any errors as early as possible (immediately after soldering the incorrect component). Again, a reminder: removing a component and re-installing it later is often very difficult!

Please refer to the layout diagram and PCB tracks diagrams below, and follow the steps carefully.

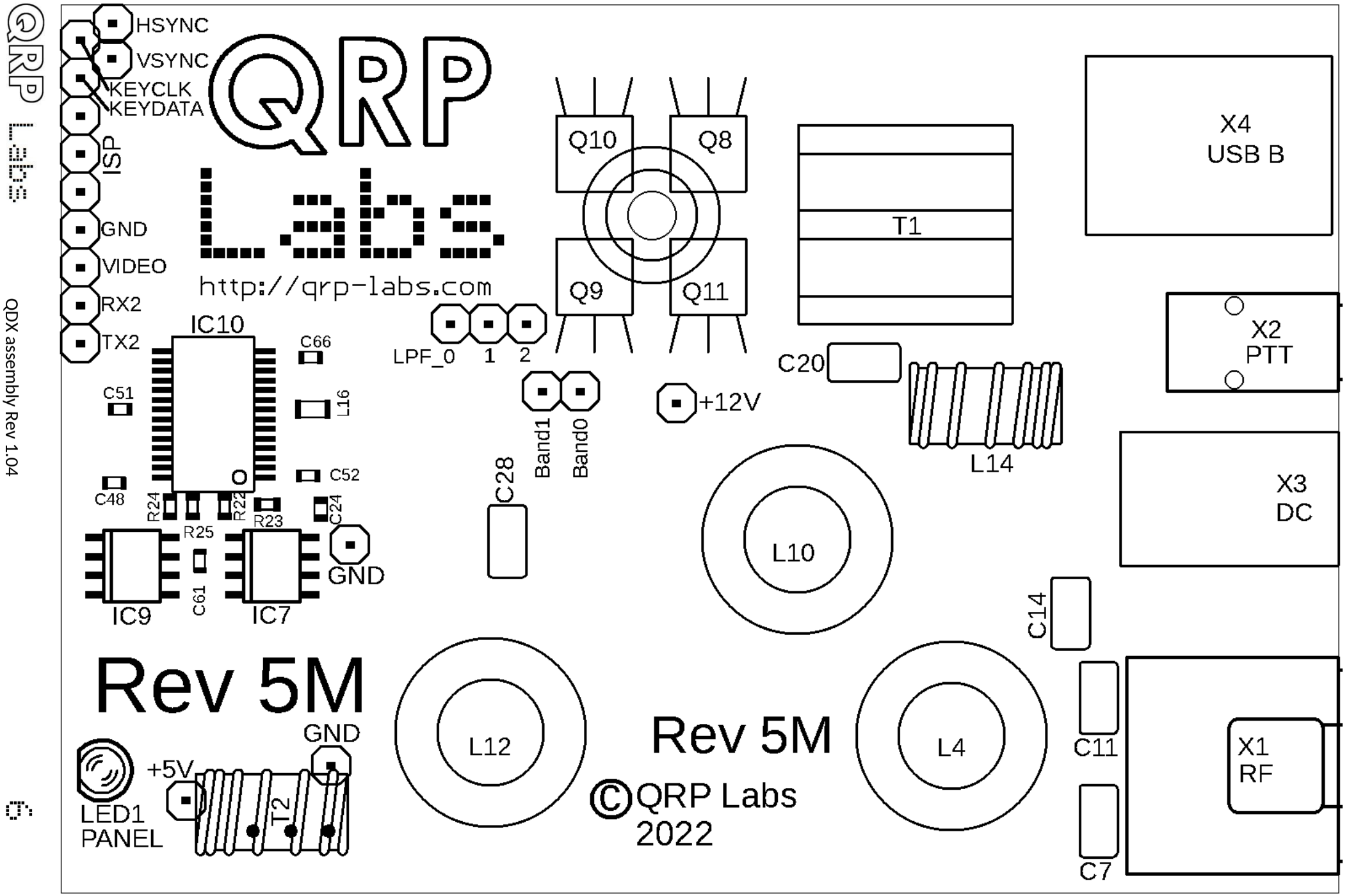
Use of a good quality soldering iron and solder is highly recommended for best results!

The following diagrams show the PCB layout and track diagrams of the QDX.

Tracks shown in BLUE are on the bottom layer. Tracks shown in RED are on the top layer. There are only two layers (nothing is hidden in the middle). Not shown in this diagram are the extensive ground-planes, on both sides of the board. Practically everything on both layers that isn't a RED or BLUE track, is ground-plane! The two ground-planes are connected at frequent intervals (not more than 0.1-inches) by vias.

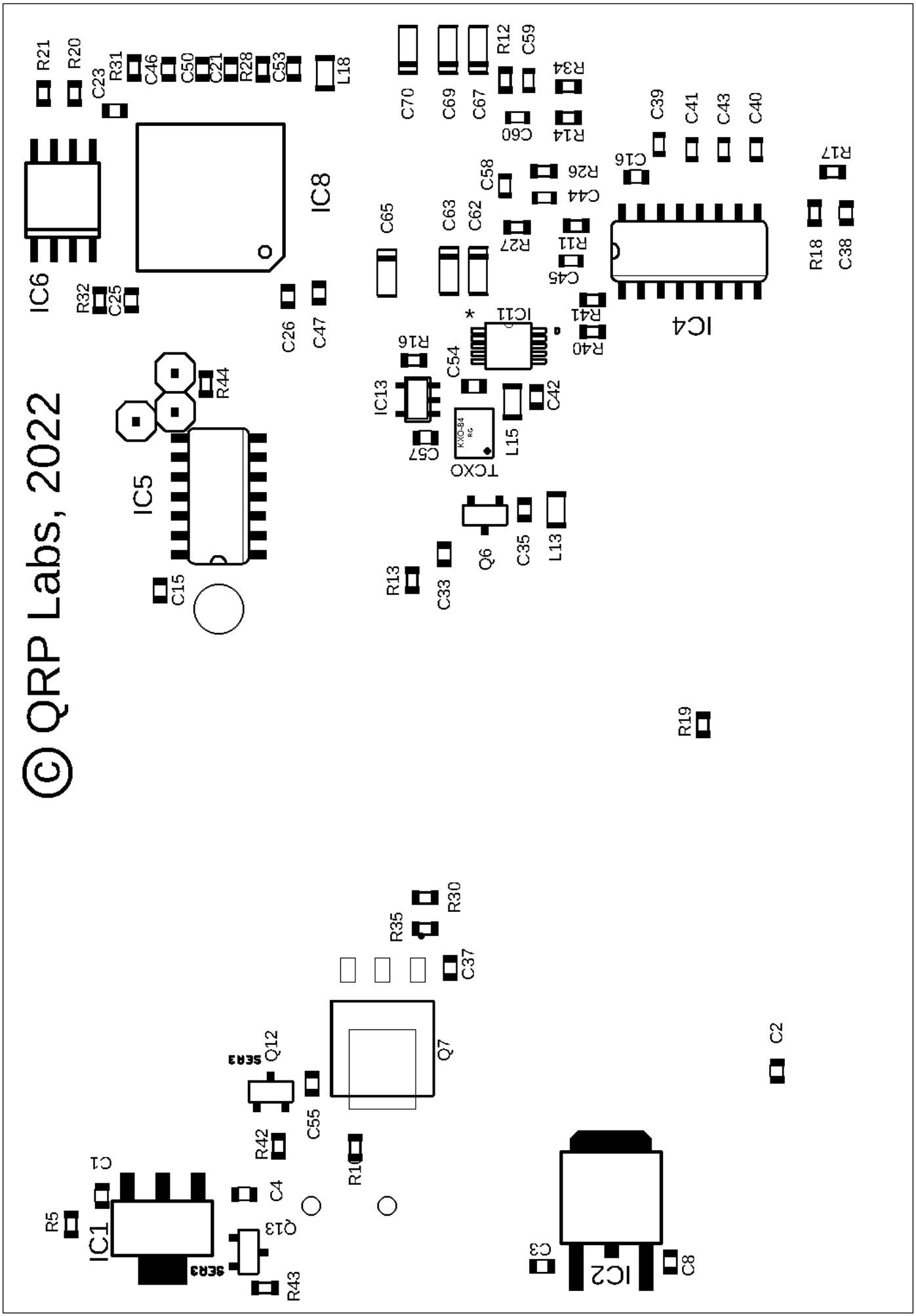
NOTE: the capacitor lead spacing on the PCB is 0.1-inches (2.54 mm) and most of the capacitors are sized appropriately for this. From time to time, due to availability constraints, we may have to use capacitors with 0.2-inch lead spacing (5.08 mm); this is not a mistake, it is just due to component availability. In this case simply use a pair of long-nosed pliers (etc) to straighten out the wires and make them spaced for the 0.1-inch pads.

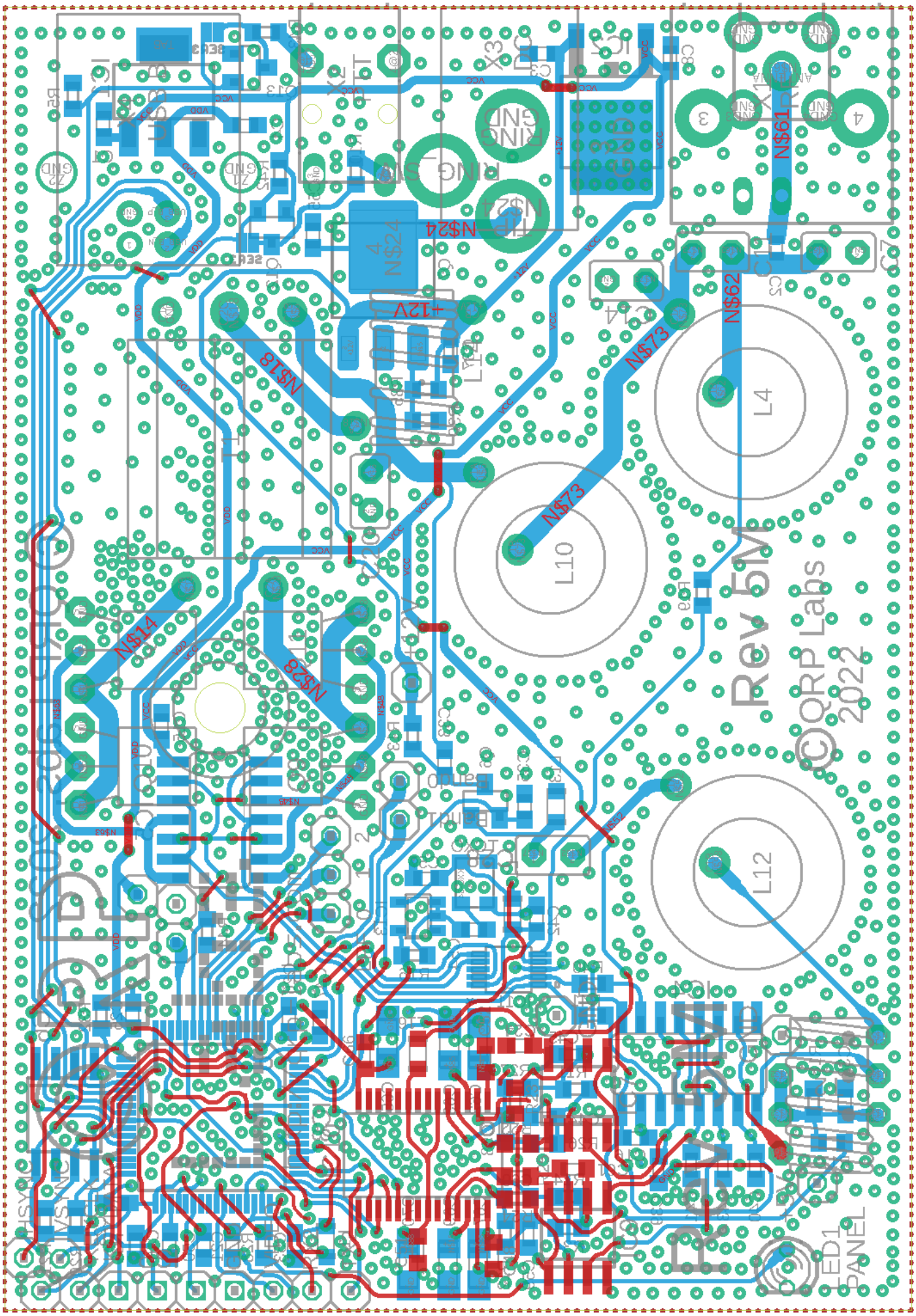
NOTE: PCB Rev 3a is different from Rev 3 only in that the trifilar transformer T2 was moved slightly further from the PCB edge, and some components nearby were moved slightly too. **Rev 3a has a short-circuit from a via to ground, which must be resolved – see section 2.3a below.**



QRP Labs

QDX assembly Rev 1.04


© QRP Labs, 2022
IC1
IC2
IC4
IC5
IC6
IC8
IC11
IC13
TCXO
Q6
Q7
Q12
Q13
L13
L15
L18
C1
C2
C3
C4
C8
C15
C16
C23
C25
C26
C33
C35
C37
C38
C39
C40
C41
C42
C43
C44
C45
C46
C47
C50
C53
C54
C55
C57
C58
C59
C60
C62
C63
C65
C67
C69
C70
R5
R10
R11
R12
R13
R14
R16
R17
R18
R19
R20
R21
R26
R27
R28
R30
R31
R32
R34
R35
R40
R41
R42
R43
R44
SER3
SER3
KXO-84


N$24
+12V
N$18
N$14
N$28
N$73
N$62
N$61
L10
L4
L12
Rev 5M
QRP Labs
2022

2.2 Parts list

Many components are SMD, pre-soldered to the PCB in the factory. Only through-hole components need to be installed by the constructor. SMD components in the parts list are identified in the Description column and by the text colour being purple.

Resistors

Qty	Value	Description	Component numbers
2	0-ohms	SMD 0603	R31, 32
6	47-ohms	SMD 0603	R22, 23, 24, 25, 40, 41
2	100-ohms	SMD 0603	R11, 34
7	470-ohms	SMD 0603	R10, 12, 14, 26, 27, 28, 43
2	1K	SMD 0603	R20, 21
5	10K	SMD 0603	R13, 17, 18, 30, 35
4	100K	SMD 0603	R5, 19, 42, 44
1	470K	SMD 0603	R16

Capacitors (50V, Multi-layer Ceramic capacitors)
(see section 2.4 below for listing of band-version dependent capacitors)

Qty	Value	Description	Component numbers
4	4.7nF	SMD 0603	C44, 45, 59, 61
2	10nF	SMD 0603	C58, 60
4	33nF	SMD 0603	C39, 40, 41, 43
20	0.1uF	SMD 0603	C15, 16, 21, 23, 24, 25, 26, 33, 35, 37, 46, 47, 48, 50, 51, 52, 53, 54, 57, 66
2	1uF	SMD 3216 tantalum	C63, 69
8	2.2uF	SMD 0603	C1, 2, 3, 4, 8, 38, 42, 55
4	10uF	SMD 3216 tantalum	C62, 65, 67, 70
1	22uF	SMD 0405 electrolytic	C36
1	470uF	SMD 0607 electrolytic	C49

Semiconductors

Qty	Description	Component numbers
1	SMD: AMS1117-3.3	IC1
1	SMD: 78M05	IC2
1	SMD: FST3253 SOIC	IC4
1	SMD: 74ACT08	IC5
1	SMD: 24C64	IC6
2	SMD: LM4562	IC7, 9
1	STM32F401RBT6	IC8
1	PCM1804	IC10
1	SMD: Si5351A / MS5351M	IC11
1	SMD: SN74LVC1G00DBVR	IC13

Qty	Description	Component numbers
1	SMD: BSS84 MOSFET	Q12
2	SMD: BSS123 MOSFET	Q6, 13
4	BS170: TO92 MOSFET	Q8, 9, 10, 11
1	SMD: AOD403 (TO252)	Q7
1	3mm Red LED	PANEL
1	SMD: 25MHz TCXO module	TCXO

Inductors

(see sections 2.8 and 2.9 below for listing of band-version dependent inductors)

Qty	Description	Component numbers
5	SMD: 47uH	L13, 15, 16, 18
1	FT37-43 10t	L14
1	BN43-202 binocular, 3:3	T1
1	FT37-43 10t trifilar	T2

Miscellaneous

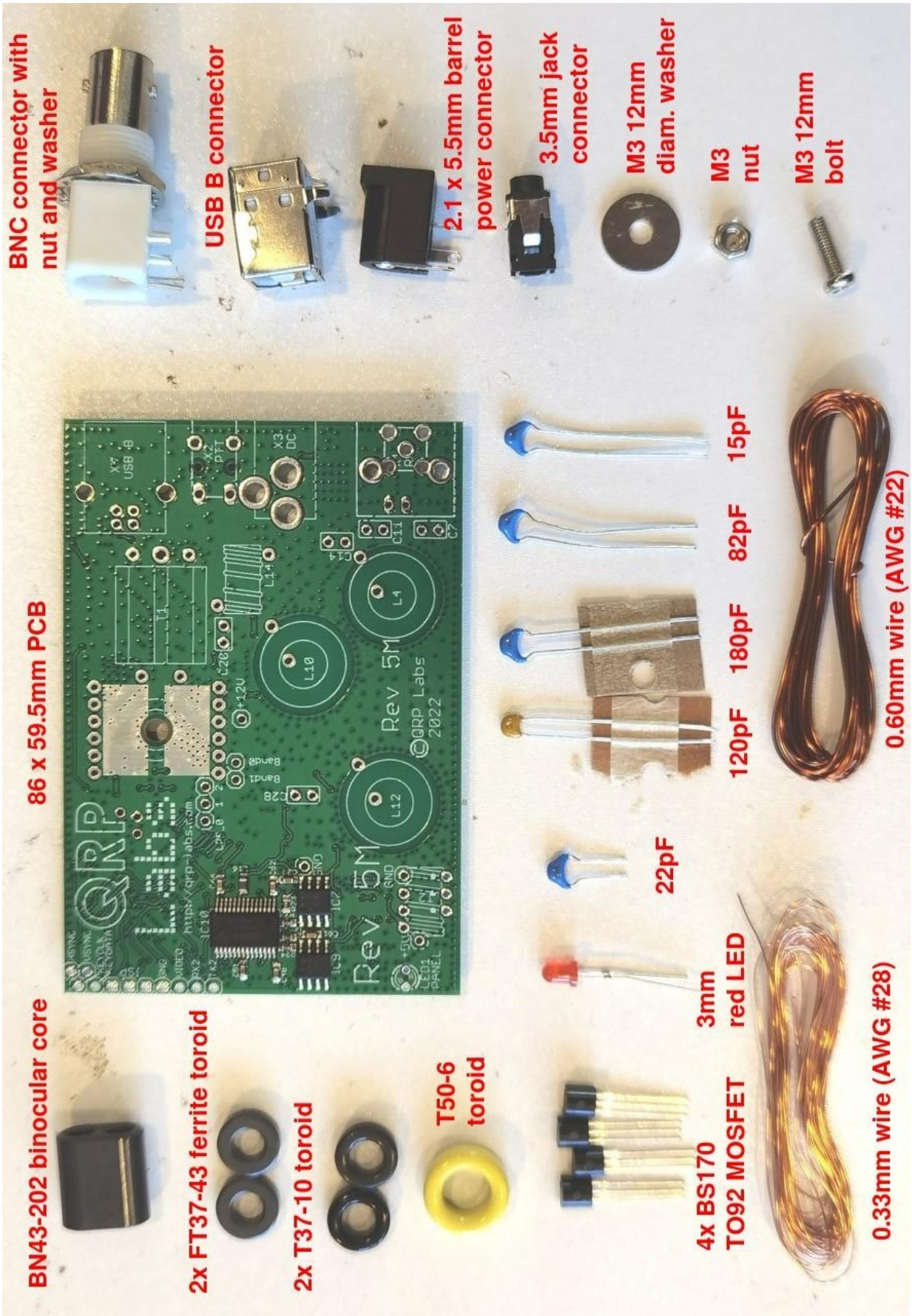
Qty	Value	Description
1	2.1 mm DC	2.1 x 5.5 barrel DC power connector
1	USB B	USB type B connector
1	BNC	RF connector
1	3.5mm jack	3.5mm Stereo connector (PTT)
1	PCB	Main PCB, 86 x 59.5 mm
1	310 cm (80-20m version) 50cm (20-10m version)	0.33 mm diameter wire (AWG #28)
1	50 cm (80-20m version) 175cm (20-10m version)	0.60 mm diameter wire (AWG #22) **Note: new kit batches will be supplied with ONLY #28 wire, which slightly improves performance.**
1	M3 12 mm	Steel 12mm long M3 screw
1	M3	Steel M3 nut
1	M3 12 mm	Steel 12mm diameter M3 washer

Enclosure (OPTIONAL)

Qty	Value	Description
2	Top, Bottom	Extruded aluminium top and bottom cover
1	Front panel	Laser etched, drilled for 3mm LED
1	Rear panel	Laser etched, drilled for three connectors
8	M2.5 machine screw	Screws to secure end panels
4	Rubber foot	Self-adhesive rubber foot

2.3 Inventory parts

Note: Photo shows parts for PCB Rev 5M, 12/11/10m version of the QDX-M.
Note: Capacitor colours vary depending on availability. All are NP0/C0G Class-I RF dielectric.
Note: Recent kit batches supply only 0.33mm wire for everything, it improves performance.



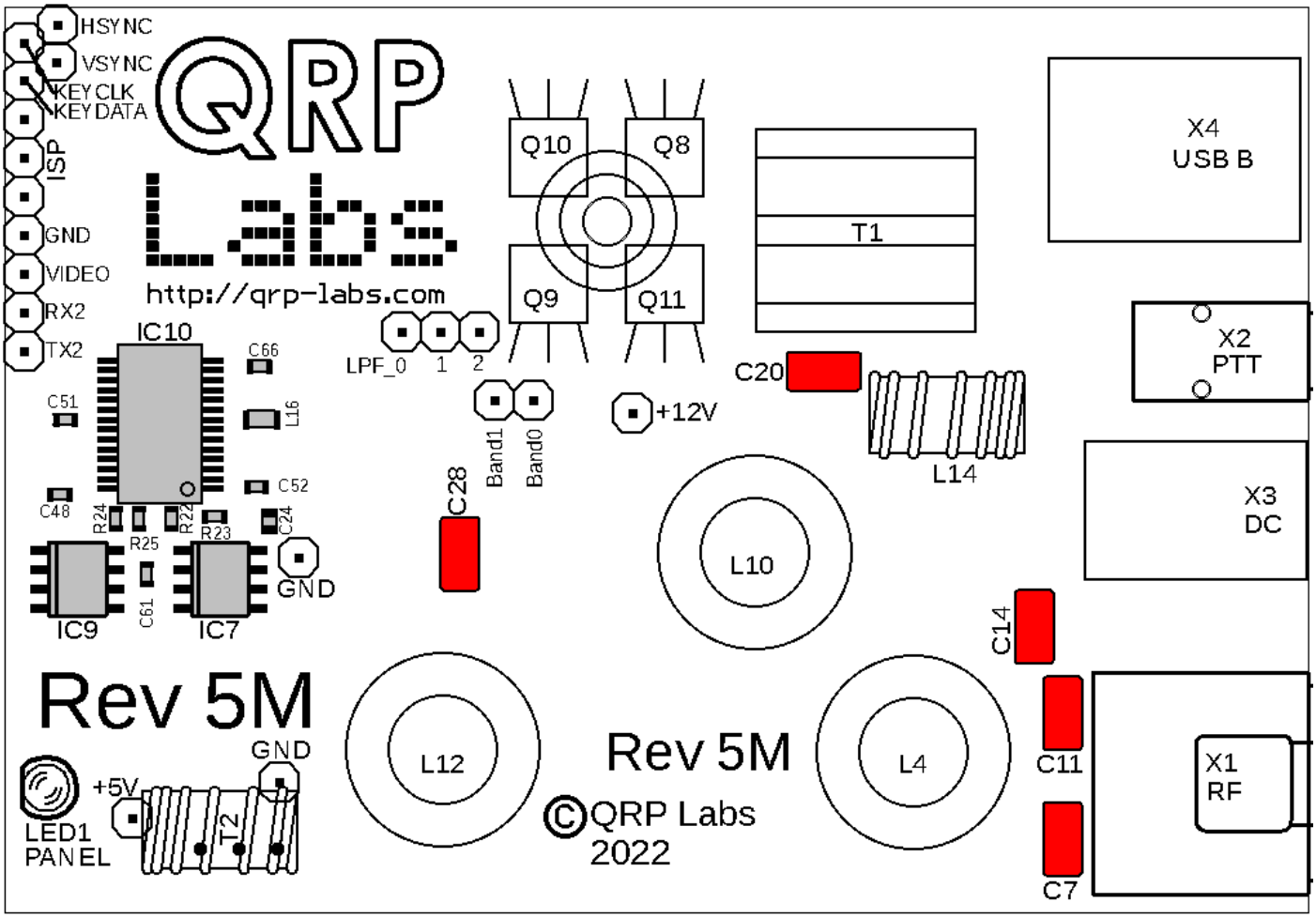
2.4 Install all the ceramic capacitors

Install all 5 through-hole capacitors in accordance with the following diagram. The procedure is so easy that I have included all capacitors in a single assembly step rather than doing one diagram for each capacitor value. Nevertheless be very careful to insert the correct value capacitors in the correct places. Mistakes are hard to correct later.

In the diagram, the component label (capacitor body inscription) is written inside the capacitor body, which **may be supplied coloured yellow, or blue, according to availability**. Check the values carefully, and double-check, using optical magnification if necessary (printed values are rather small). Note that the leads of some capacitors may need to be bent to fit the 2.5mm holes.

Capacitor values depend on the band being built. Please refer to the table below. Note that capacitor values below 100pF may be labeled either as a 2-digit value + 1-character tolerance (for example “56J” for 56pF), or as a 3-digit value in the conventional 2-digits value + 1-digit for the number of zeroes (for example “560” for 56pF). Be VERY careful to put the right capacitors in the right positions!

Band	C28	C20	C14	C11	C7
10/11/12m	33pF, “330”/”33J”	120pF, “121”	180pF, “181”	15pF “150”/”15J”	82pF, “820”/“82J”

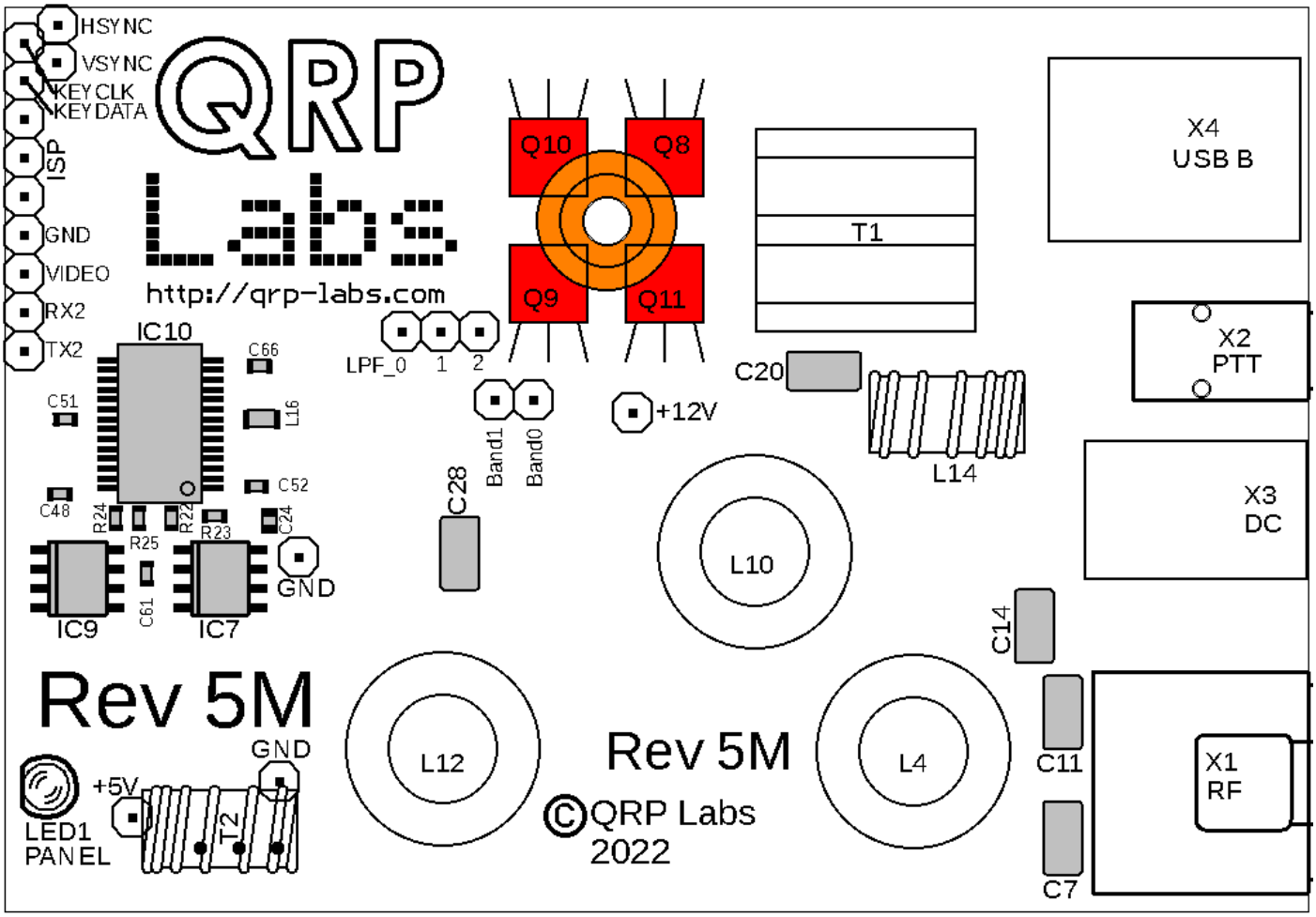


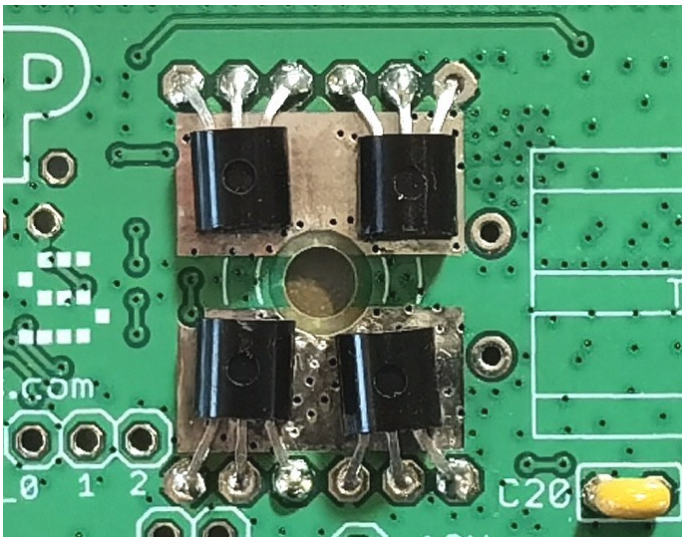
2.5 Install PA transistors

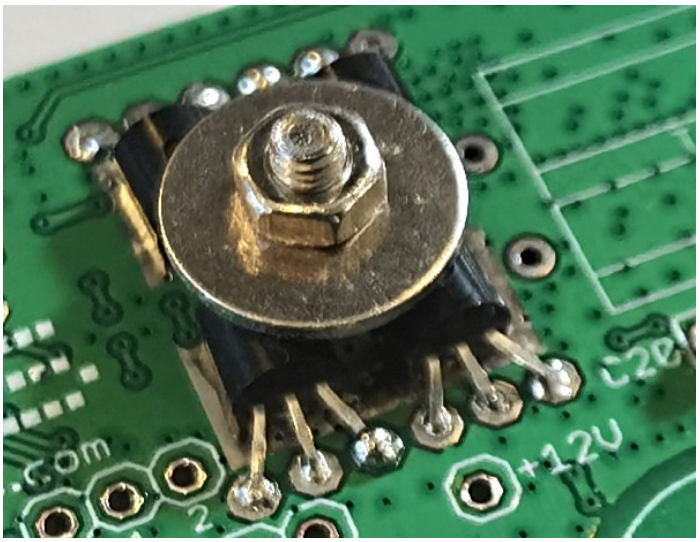
Install the transistors in the positions shown, with their flat faces flush against the PCB.

Use the 12mm M3 bolt, washer and nut to firmly press the transistor faces against the PCB, as shown.

Refer to the diagram and photographs below.

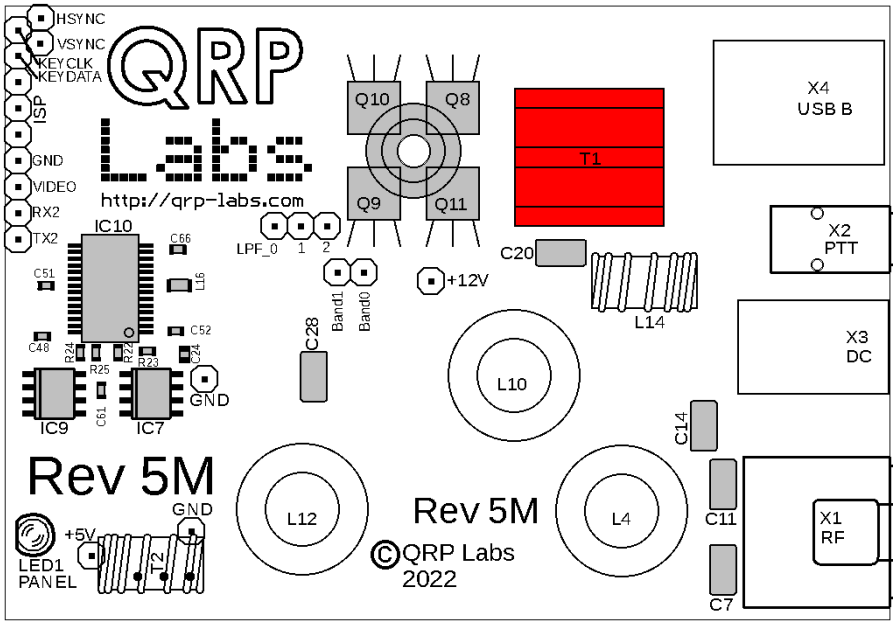


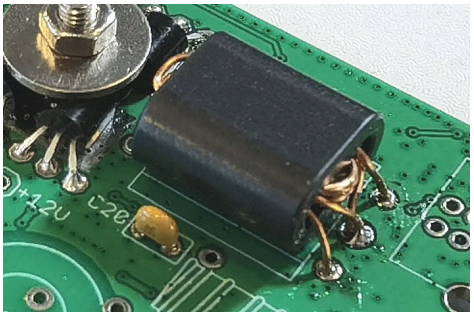




2.6 Assemble and install transformer T1

Transformer 1 is wound on the BN43-202 binocular ferrite former, using the thick 0.6mm (AWG #22) wire, or 0.33mm (AWG #28) wire if no 0.6mm is supplied (0.33mm has better performance). This wire is also used for L14 so do not use all of it on the transformer. Please refer to the separate document that describes the preparation of this transformer.

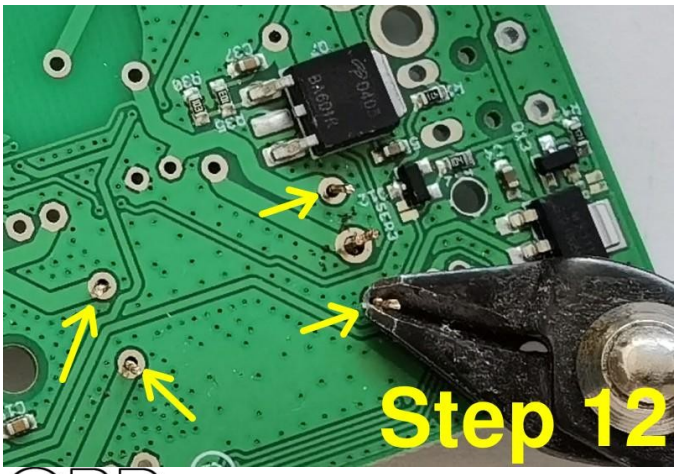


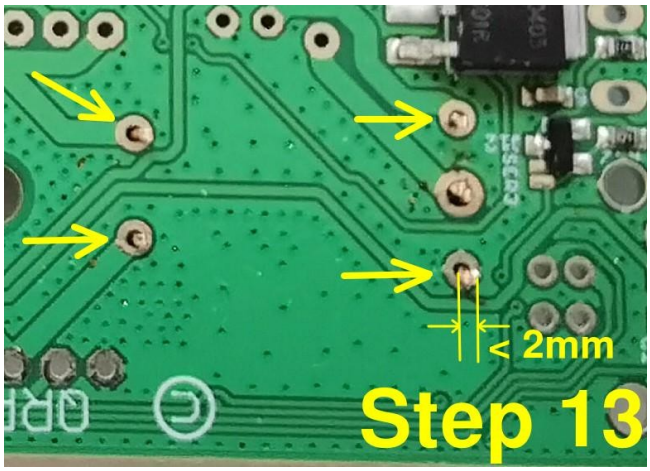


When the transformer is ready, insert the wires into the holes on the PCB, and continue with the installation steps listed here. For historical reasons the diagrams start at step 12... because the transformer winding instructions were moved to a separate document.

Step 12: Now we need to get the enamel off the wire. Usually on thinner wire, I hold the soldering iron to the wire until the enamel burns off. But that doesn't work so well on thicker wire such as this. So my technique here is to scrape the enamel off, at least partially, using wire cutters. The correct pressure needs to be applied to the wire cutters, so as NOT to actually cut the wire. I hold the wire cutter as close to the PCB as possible, then gently but firmly pull the cutter away from the PCB, scratching off the enamel. Turn the cutter to a different angle and scrape again, 2 or 3 times. It is not necessary to remove ALL the enamel, if you get a few good scrapes on, enough enamel will be removed that the soldering iron heat will burn off the rest of the enamel and a good joint will be achieved. Do this for the 4 winding ends first, leave the two center-tap for later, to make it easier.

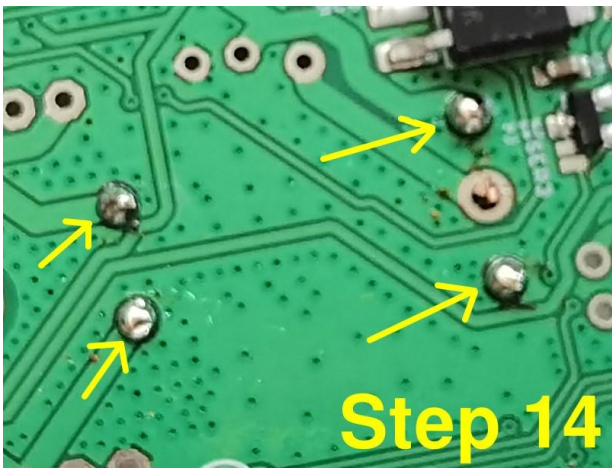
Step 13: Cut each of the four wire-endings, leaving about 2mm or less, sticking up from the PCB.

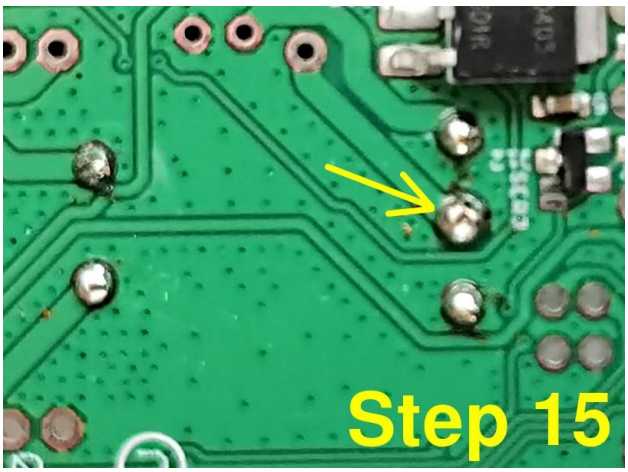




Step 14: Solder the four wire endings. At each pad, hold the soldering iron firmly, apply plenty of solder, and wait for about 10 seconds. This will ensure a good connection, and any remaining enamel will be burned away.

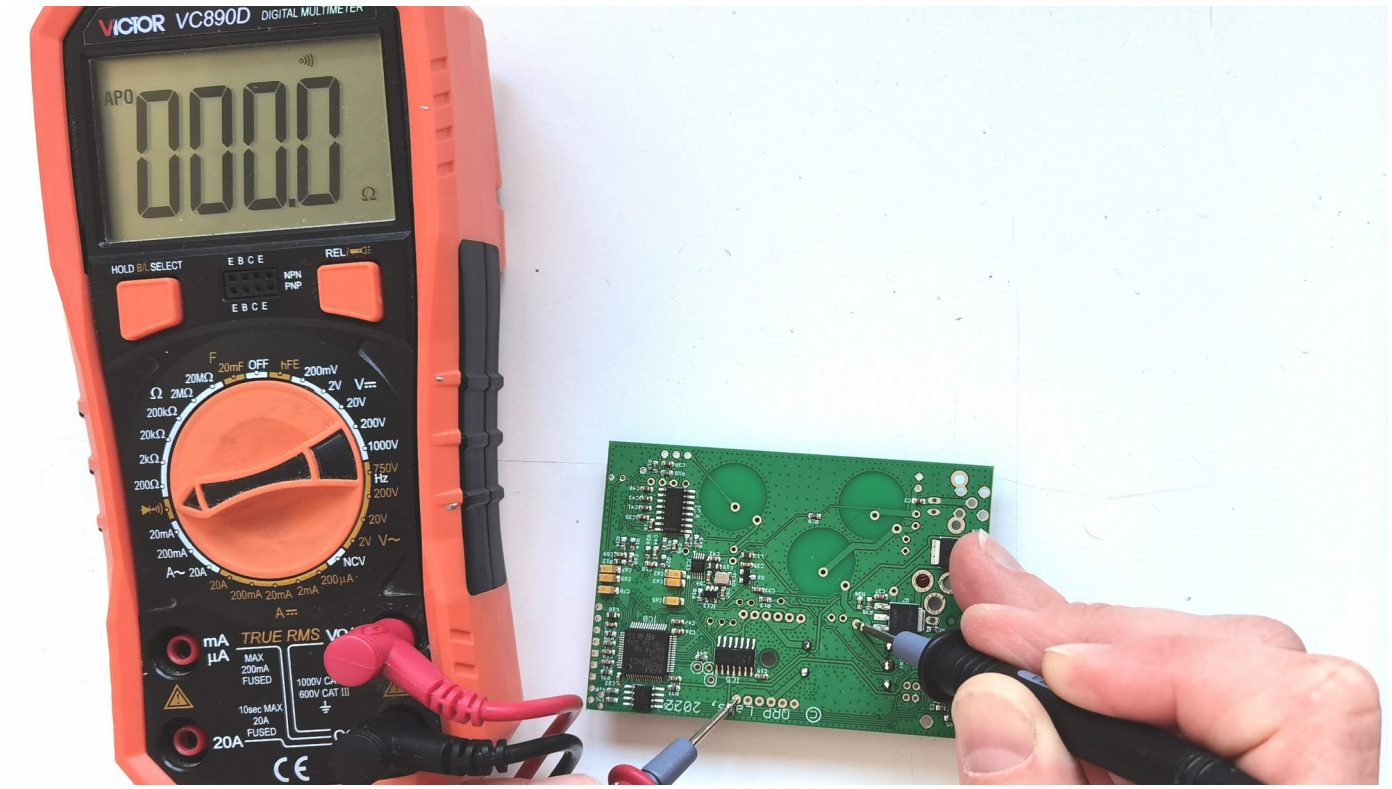
Step 15: Finally repeat the procedure with the two center-tap wires that came through the large center hole. Scrape them, cut them to 2mm, and solder them. Apply plenty of solder and hold the soldering iron tip to the joint for maybe 15 seconds, to really be sure of a good joint and burning away any remaining enamel.





Step 16: Check that none of the wires protrude more than 2mm from the surface of the PCB, since if you are using the optional aluminium enclosure, there are only a few mm clearance.

Step 17: Verify good connections for all five soldered jtoints of T1 using a DMM set to resistance or continuity mode. My cheap DMM hasn't got a continuity mode and I'm using the 2000-ohm resistance mode; in this mode when there is continuity, the reading on mine shows 001 (not zero-ohms; this is just a DMM thing, of no significance).

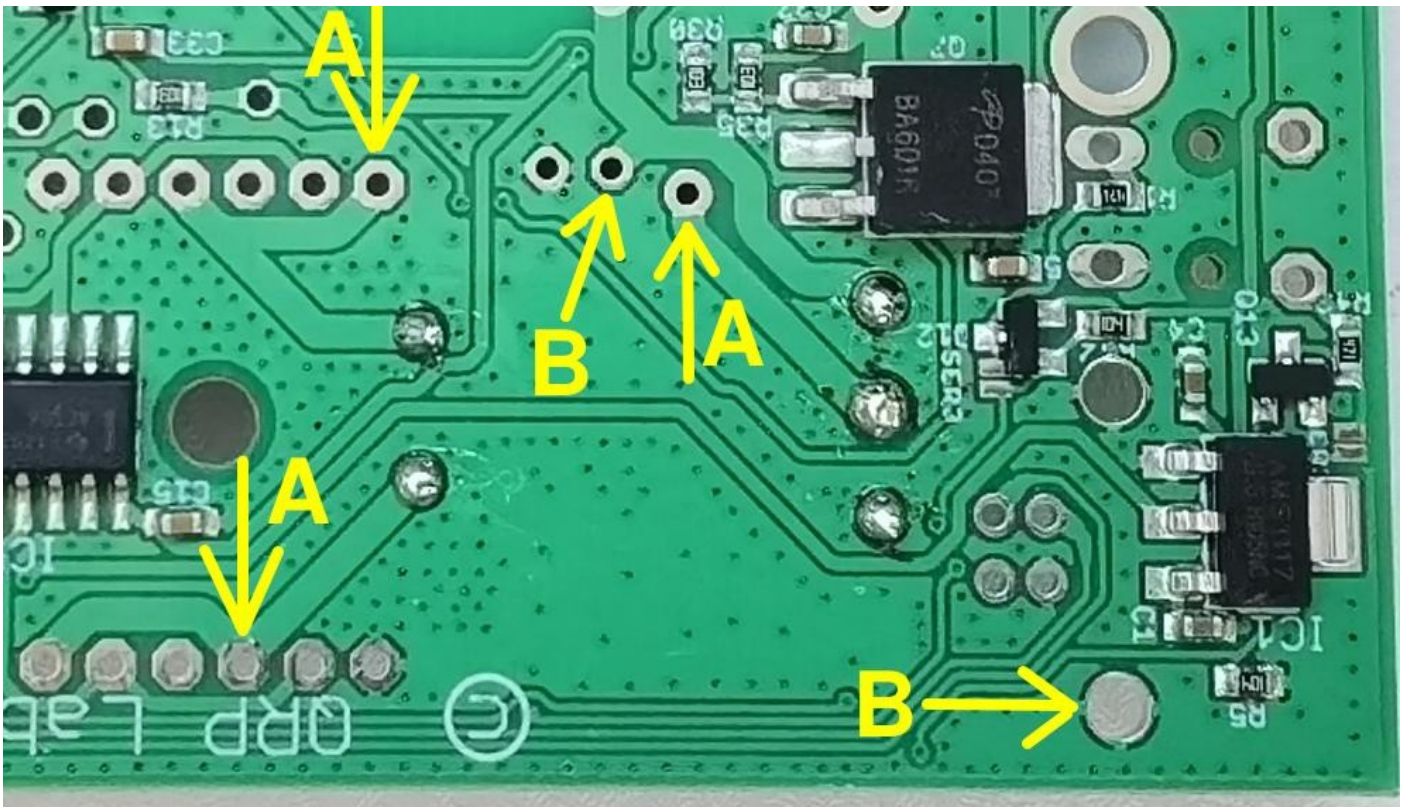


You should see continuity between all the points labeled A in the photo below. Touch the probes to pairs of these points, to verify continuity.

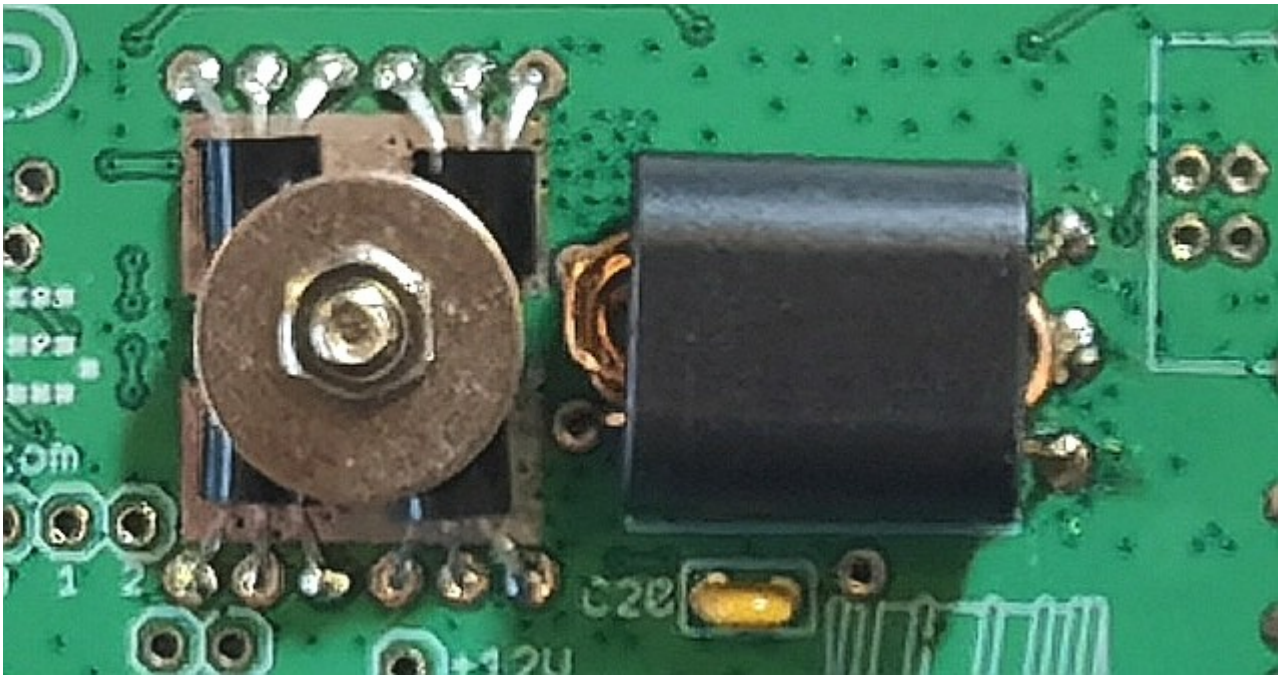
You should see continuity between the points labeled B also. Touch the two probes to the two pads labeled B and check for continuity.

Finally, there should be NO connection between A and B. Hold one probe on any A point and the other probe on a B point. That should read infinite resistance (no continuity).

If any of these tests fail, then you have a soldering problem somewhere, or the wrong wire in the wrong hole, or some short-circuit somewhere etc.



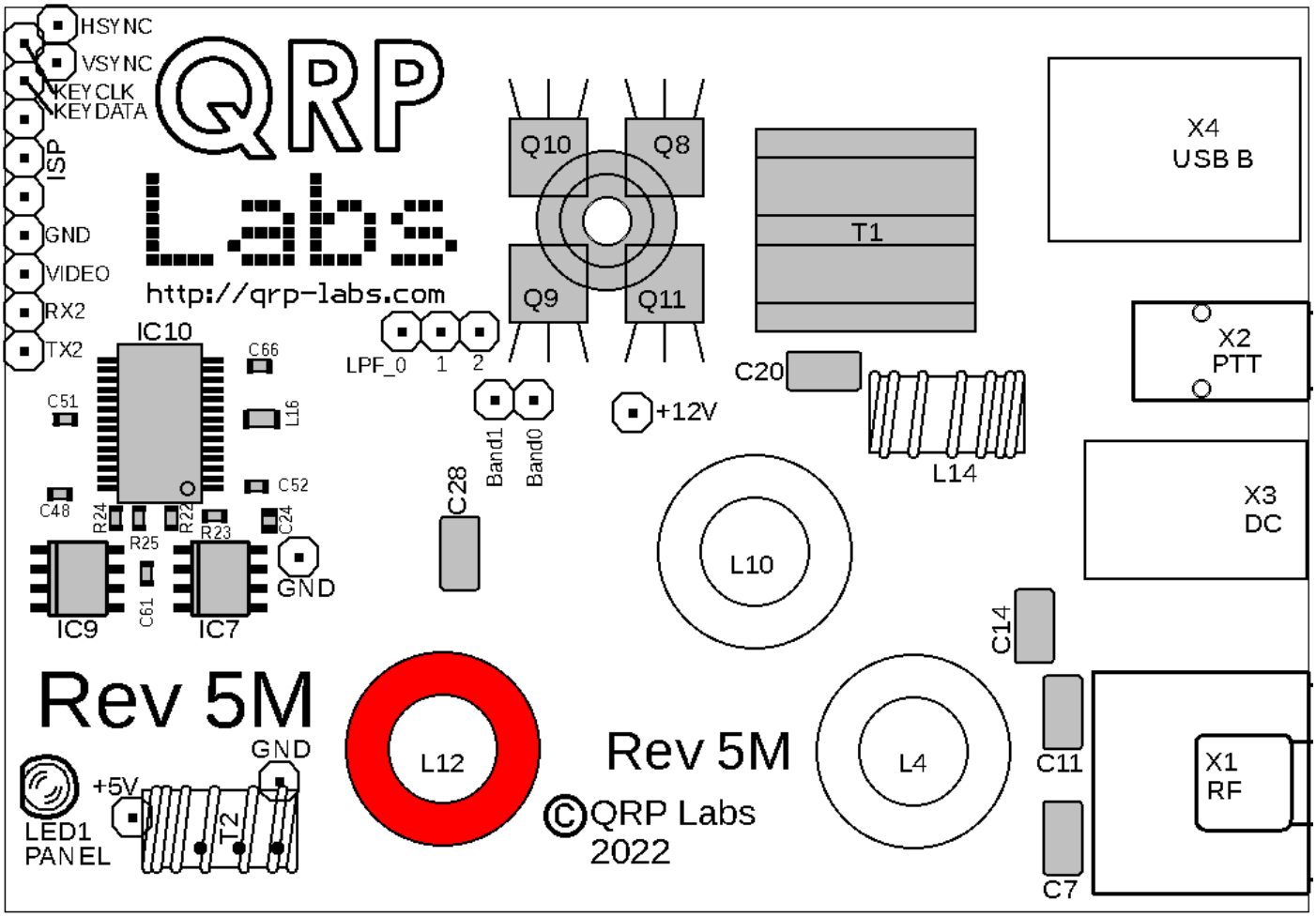
This is the final appearance of T1 when the installation is complete:



2.7 Prepare and install tapped inductor L12

Toroidal core, wire size and number of turns varies depending on the band being built. Please refer to the table below (core type, number of turns, inductance, and wire length required):

Band	Wire mm	L12 specification Core, turns (inductance), wire length
10/11/12m	0.33	T50-6 (Yellow), 16 turns (1.02uH), 32cm

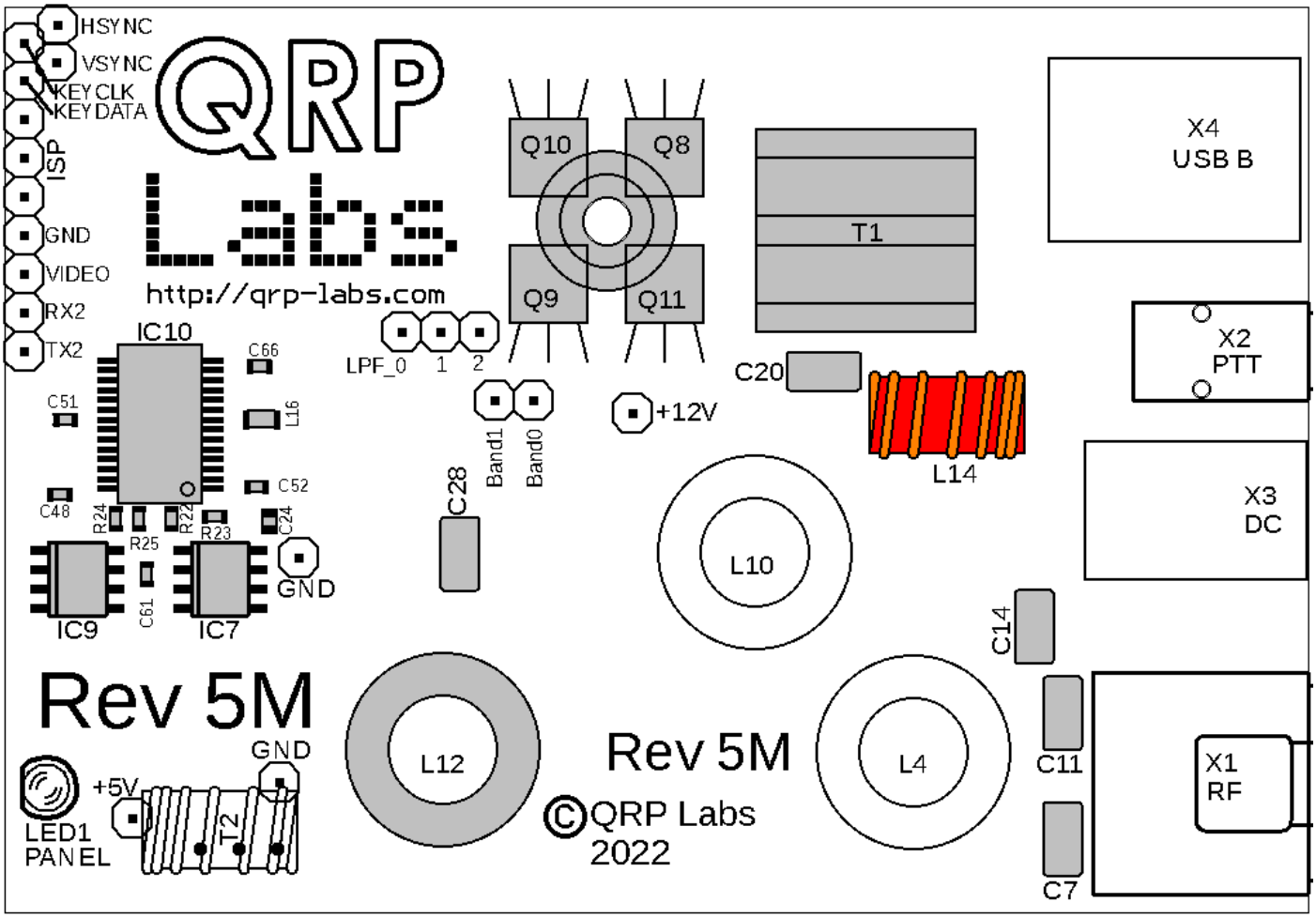


2.8 Wind and install L14

L14 consists of 10 turns of 0.6mm (AWG #22) wire, or 0.33mm (AWG #28) wire if no 0.6mm is supplied, wound on an FT37-43 toroid (**dull black colour**). Firstly, you have to understand that there's a right way and a wrong way to wind a toroidal inductor, too. The terms "right" and "wrong" are probably not as appropriate here, as "handedness" or "winding direction". There are two directions you can wind the toroid. If you choose the right one, then all the wires will automatically end up near the holes they are to be soldered into. If you do it wrongly, it will be a bit messy.

For all the toroidal inductors in the QDX-M kit, you will get it right, if you start as shown (photo, above right) and pass the wire through the toroid from the top side down through the hole, out and then around and over again; accumulating turns in the anticlockwise direction.

With this in mind, wind 10 turns and install the toroid. As with the output transformer T1, I recommend scratching the enamel with a knife or wire-cutter, and then cutting it to about 2mm length on the underside of the board. Then solder, applying the soldering iron for at least 10 seconds and plenty of solder, to ensure a good connection. Do NOT tin the wires before inserting into the holes: the holes are not large enough to accommodate tinned wires.



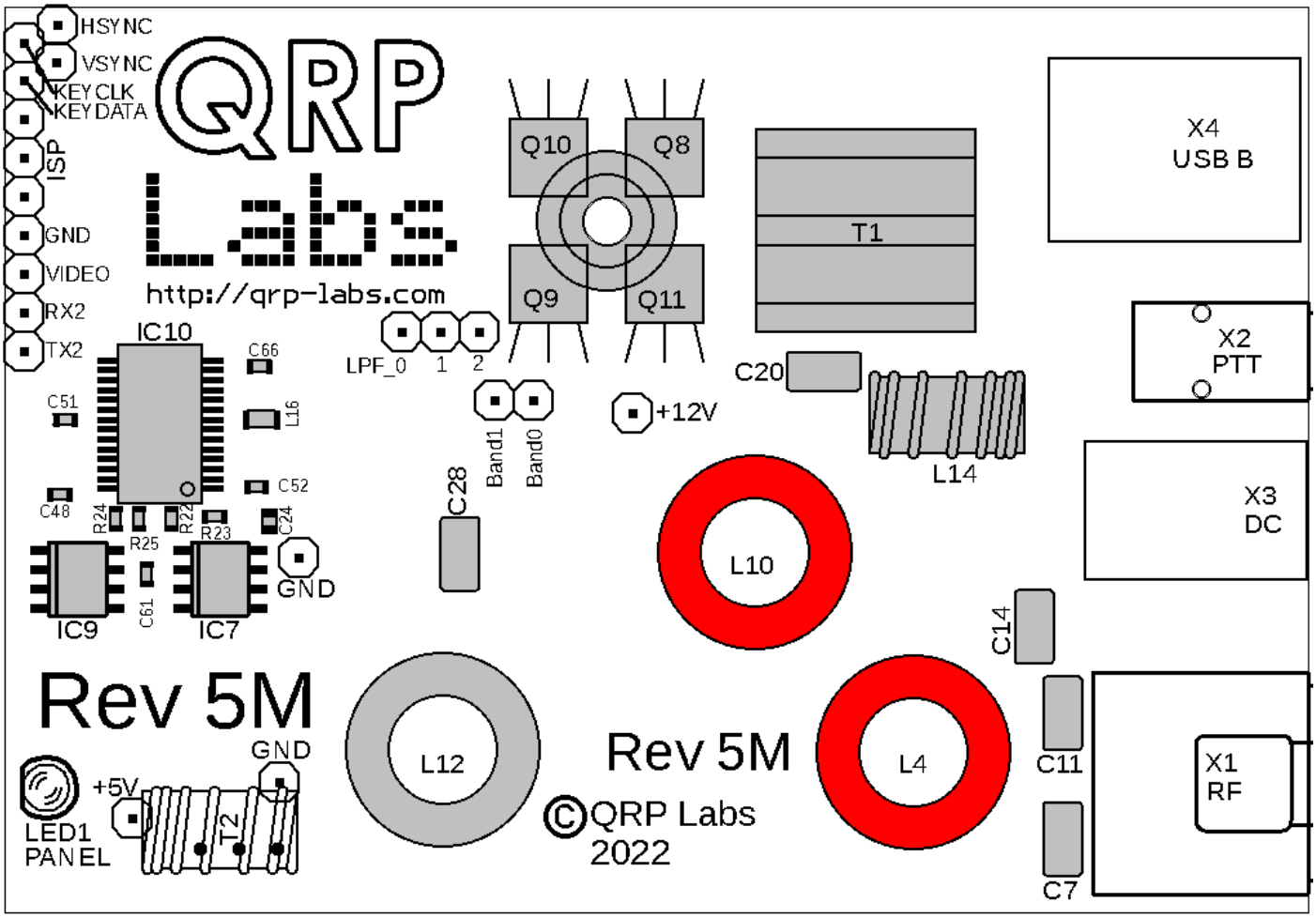
2.9 Install Low Pass Filter toroids

Now wind and install the two Low Pass Filter toroids. As per the previous direction, using the right winding direction will make the toroids much easier to fit to the PCB. Make sure to count carefully and install the toroids in the correct places! Remember that each time the wire passes through the center of the toroid counts as one turn. Spread the turns evenly with a small gap. Winding the toroids as tightly as possible will slightly improve the performance of the Low Pass Filters. But don't pull so tightly that you break the wire!

Toroidal core, wire size and number of turns varies depending on the band being built. Please refer to the table below (core type, number of turns, inductance, and wire length required):

Band	Wire mm	L4 specification Core, turns (inductance), wire length	L10 specification Core, turns (inductance), wire length
10/11/12m	0.33 or 60mm	T37-10 (Black), 10 turns (250nH), 18cm	T37-10 (Black), 11 turns (302nH), 20cm

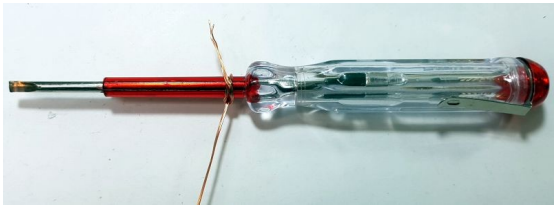
NOTE: Recent kit batches may be supplied with only 0.33mm wire. We found it improves performance.

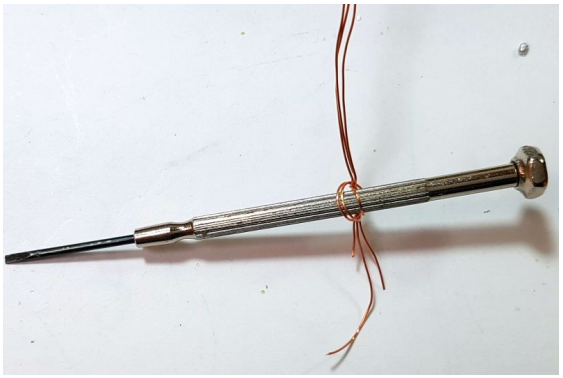


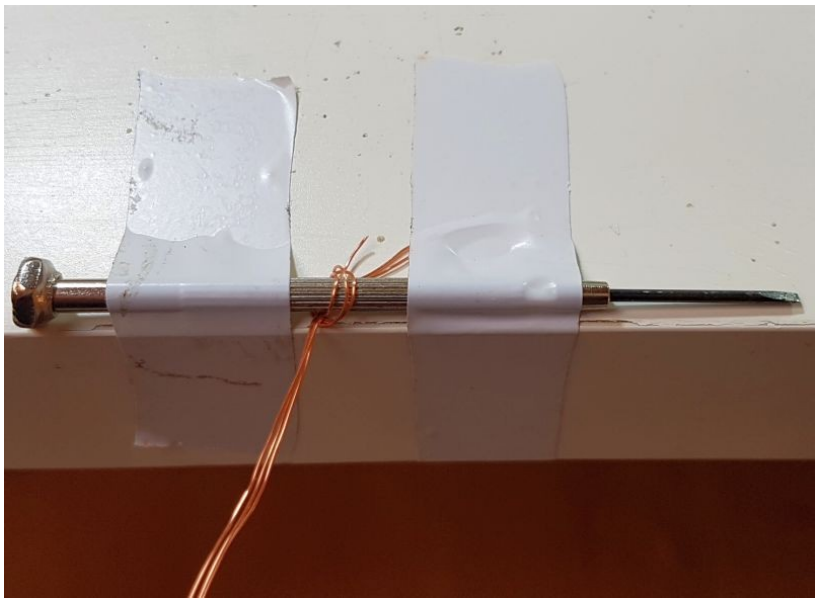
2.10 Wind and install trifilar toroid T2

This toroid needs some care so please follow these instructions very carefully.

Divide the remaining 0.33mm (AWG #28) wire into three equal lengths. These three pieces now need to be tightly twisted together to make the trifilar wire. My method for this is to tie one end in a knot around a small screwdriver shaft. Similarly tie the other end around another small screwdriver. Now clamp one end somehow to something solid. You could use a vise, if you have one. If you don't, then you have to get creative and think of something. Here I taped it to the edge of the desk. Now you can twist the screwdriver at the free end, repeatedly until you twist the three wires together thoroughly. You need to keep the wire under a little tension to keep the twists evenly spaced.





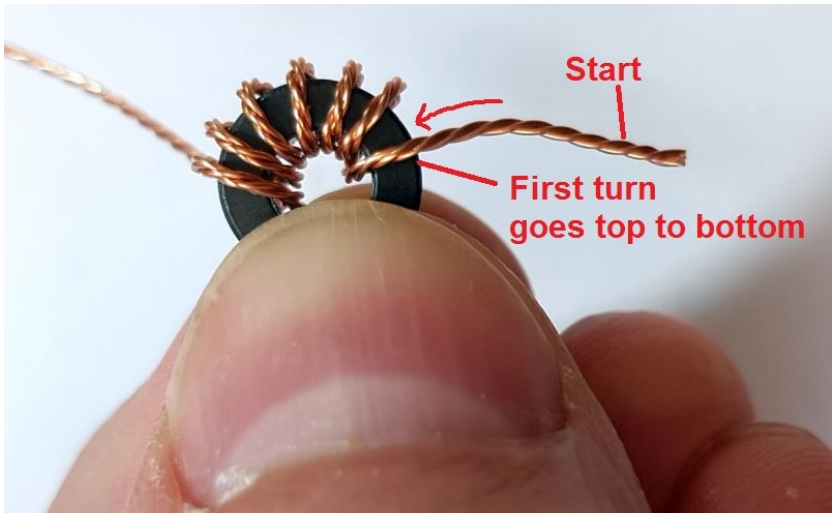


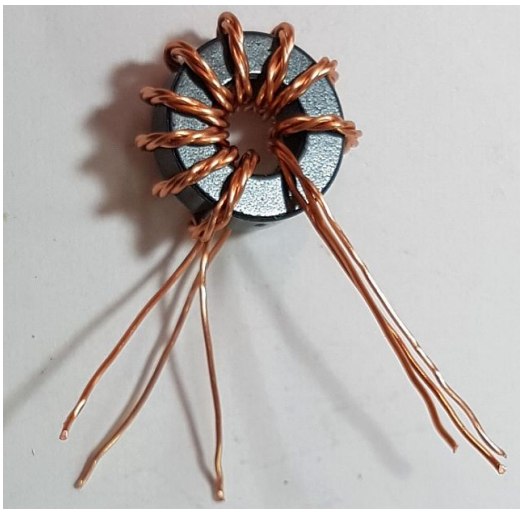
A 20cm length of wire should have about 60 twists. It is not critical.

The end result is something like the photo (right). The measurement scale is in cm.

Now cut off the untidy ends, and this is the piece of wire that will be used to wind the FT37-43 toroidal core as a trifilar transformer.

Hold the core between thumb and finger. Pass the wire first from above, to below. Then take the wire from below, and bring it around to pass through the toroid again to form the second turn. After each turn, ensure the wire is fitting snugly around the toroidal core. Wind 10 turns on the core. Each time through the toroid's central hole counts as one turn. Cut off the excess wire, leaving about 2.5cm remaining.

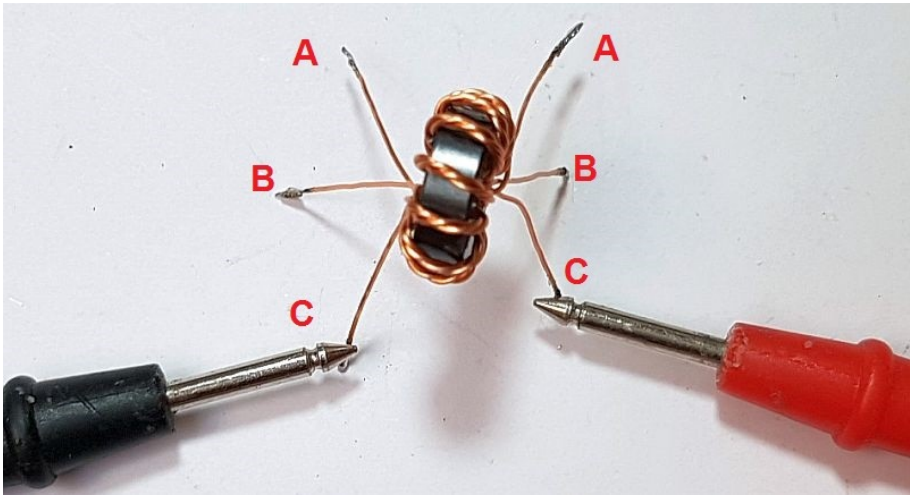




Now it's necessary to identify which wire belongs to which winding. You have three windings twisted together, they all use the same wire. The only way to do this is with a DMM as continuity tester. First, untwist and straighten the wire ends that are not wound around the toroidal core.

Now tin the last few mm at the ends of each wire. You can do this by scraping off the enamel then tinning with the soldering iron; or, if your soldering iron is powerful enough, hold the wire end in a blob of molten solder for a few (maybe 10) seconds, until the enamel burns off.

Now use a DMM to test for continuity. Re-arrange the wires so that there is continuity from A-A, B-B, and C-C in this photo.



Carefully keep this orientation of wires and insert the transformer this way into the PCB. BE CAREFUL not to lose the orientation of the wires! The right wires must be in the right holes, so that the windings are connected correctly in the circuit!

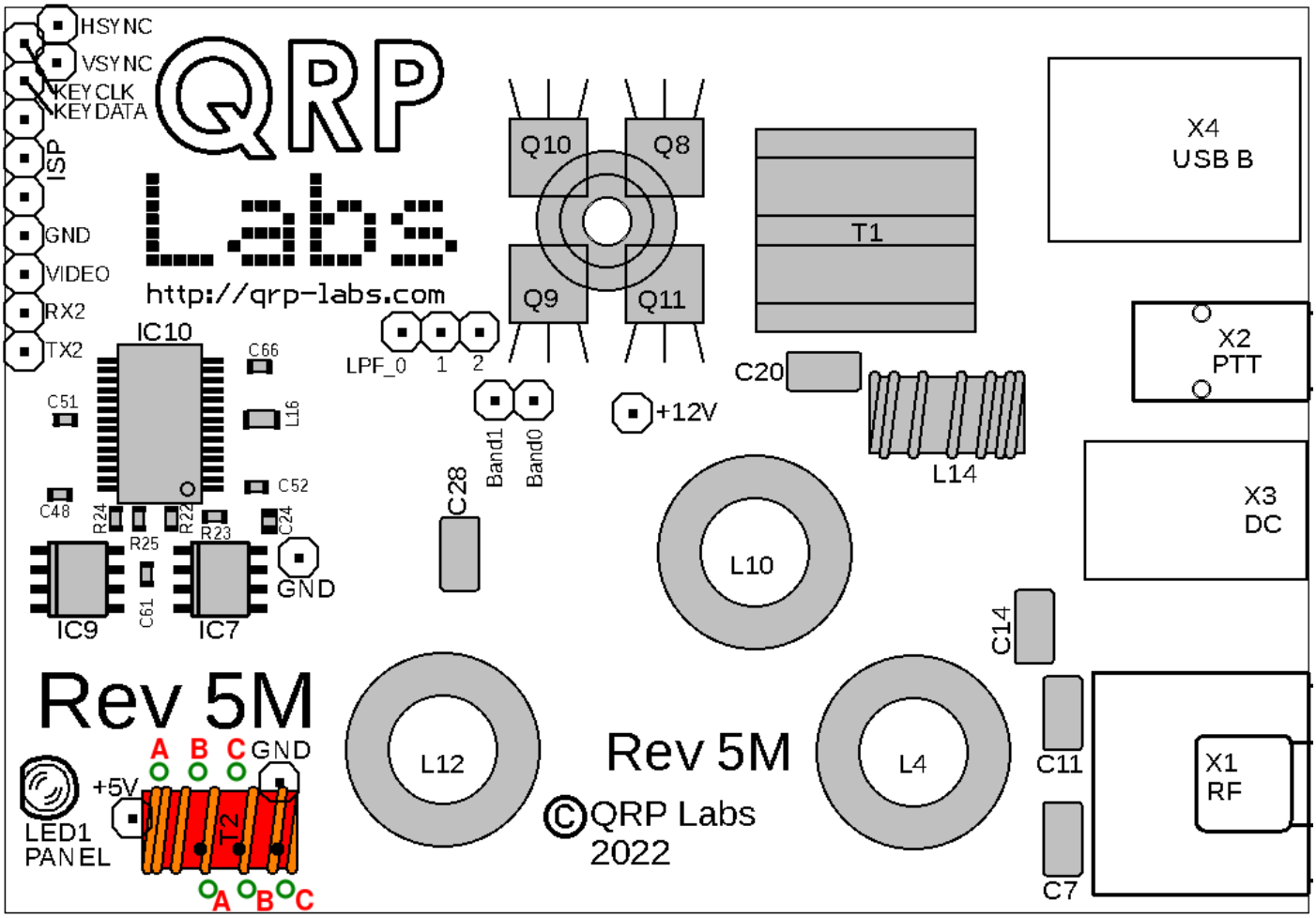
Refer to the diagram on the following page to see the correct orientation.

Once the wires are inserted through the correct holes in the PCB, and pulled tight, check for continuity between the pairs of wires in holes A, B and C once AGAIN. It is much easier to get this right first time, than it is to make repairs later!

Be sure NOT to solder any wire into the hole marked "GND"...

When you are satisfied that the wires are all in the correct holes, you can cut them to a length of about 2mm and solder them. It is best to cut-and-solder one wire at a time, since if you cut all the

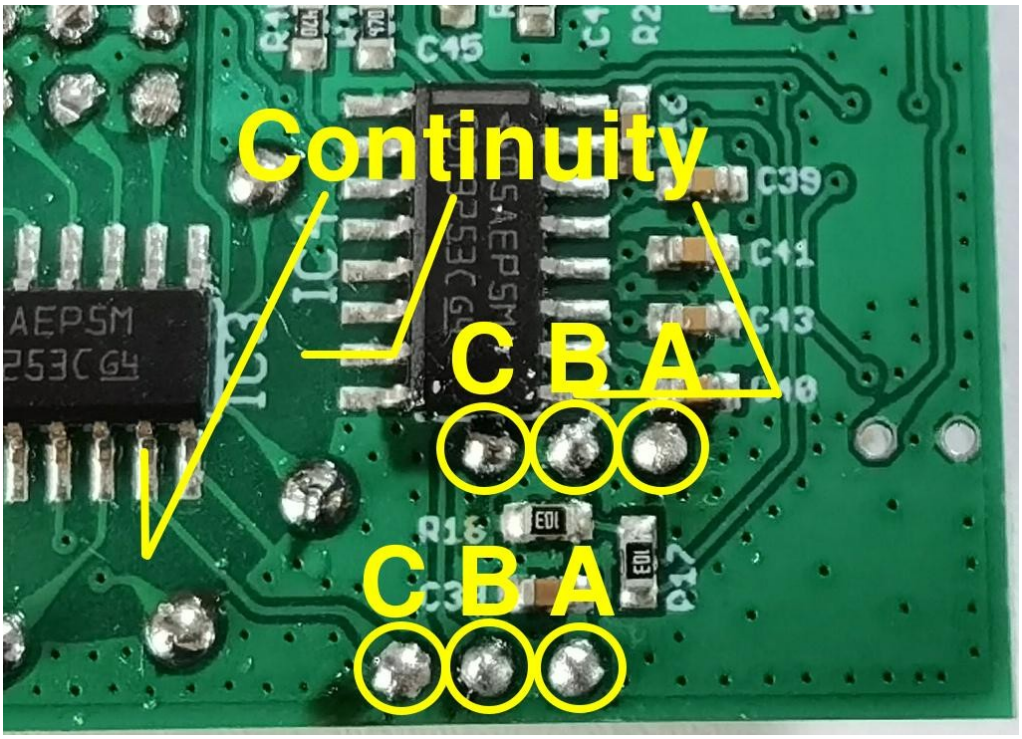
wires to 2mm length then the toroid is more likely to fall out before you've had a chance to solder any of the wires. If that happens, it will be tough to get all the wires back in the correct holes again.



There are numerous SMD components in the vicinity so be VERY careful when soldering the wires of toroid T2, NOT to touch any of the nearby SMD components!

Once again my method involves holding the iron and plenty of solder on the joint for at least 10 seconds to make sure the enamel burns off completely.

The photograph (right) shows the trifilar wire ending holes labeled A B C on the lower side of the PCB. You should check for continuity between each pair of pins A-A, B-B and C-C; and the triplet of pins IC4 pins 7 and 9, and IC3 pin 7.



2.13 Install the 3mm red status LED

Identify the shorter of the two wires of the 3mm red LED and bend it 90-degrees right at the body of the LED.

Bend the other wire 90-degrees but at a distance of approximately 2.5mm (0.1-inches) from the body of the LED (see photograph, right).

Insert the LED leads into the holes in the PCB. As an additional check, the LED will have a flat part on its body, which will sit flat on the surface of the PCB. Solder the LED In place but be careful that it is accurately aligned in position, protruding over the board edge at right-angles. This will ensure that later if you are using the optional enclosure, the LED will fit without difficulty.

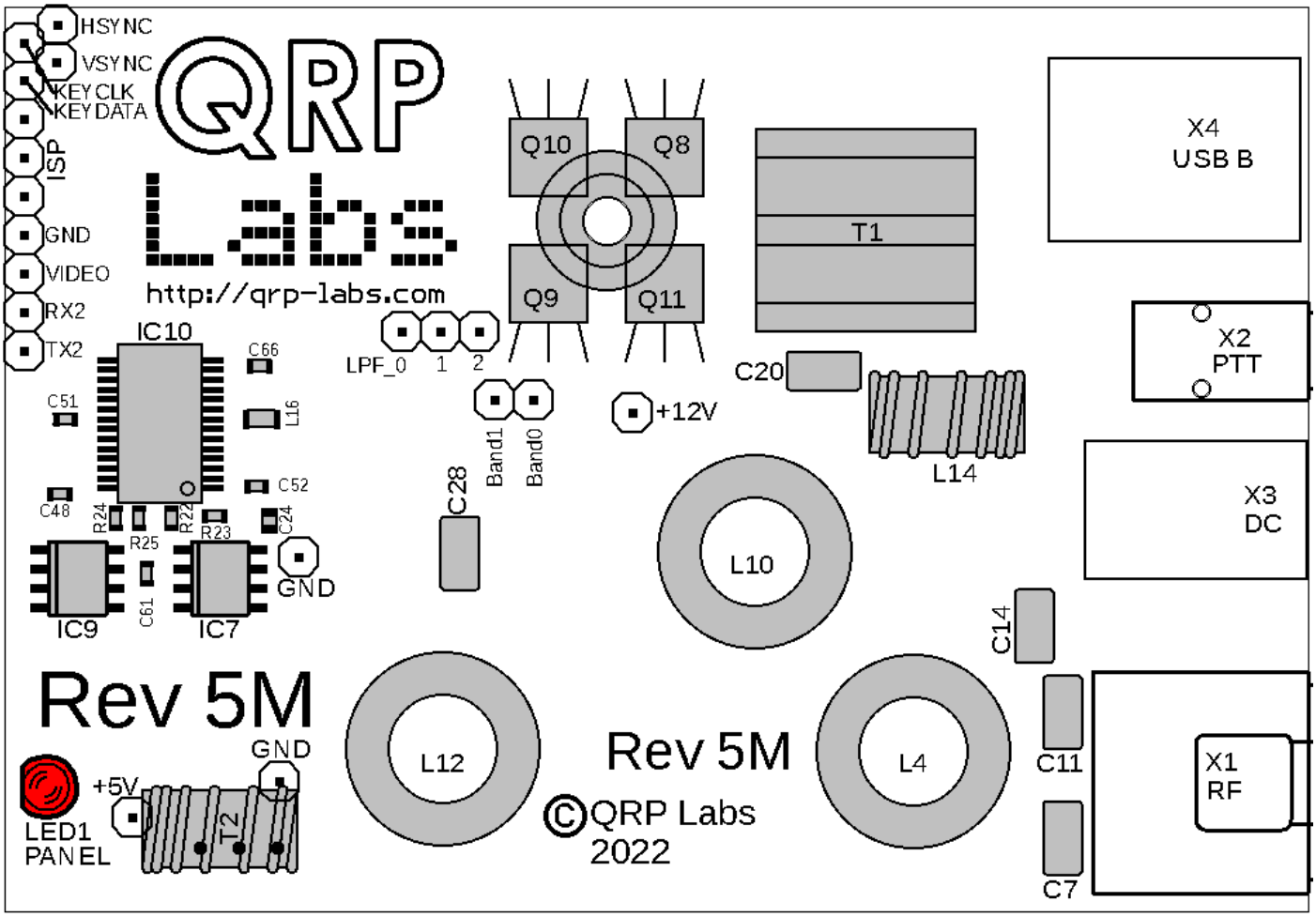
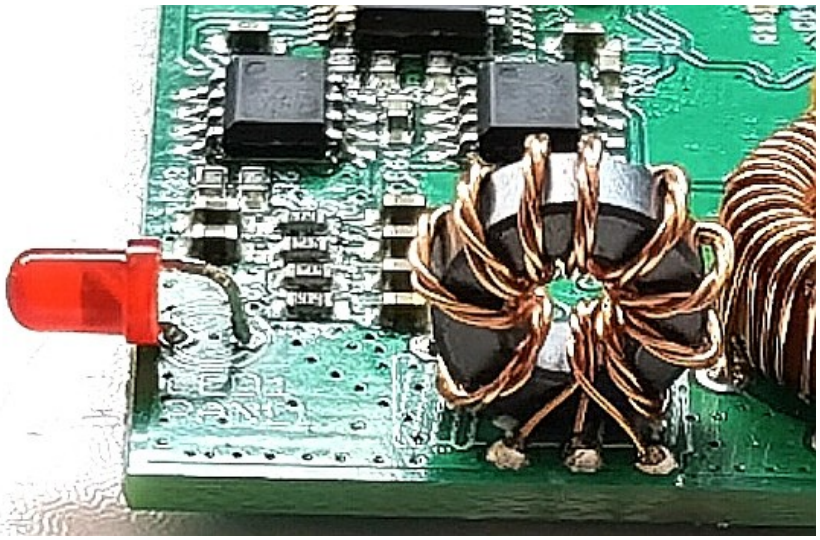


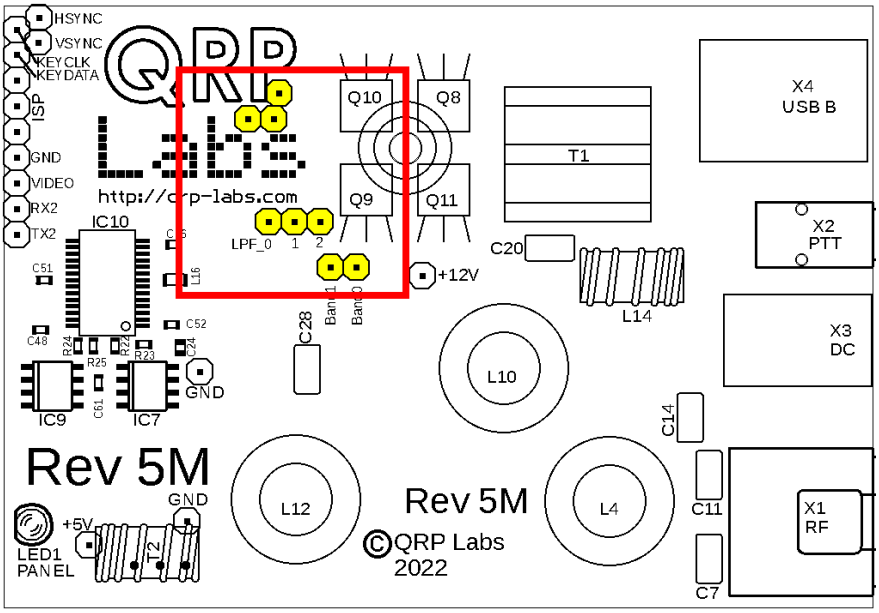
Photo (right) shows the LED installed correctly (Rev 3 PCB pictured)



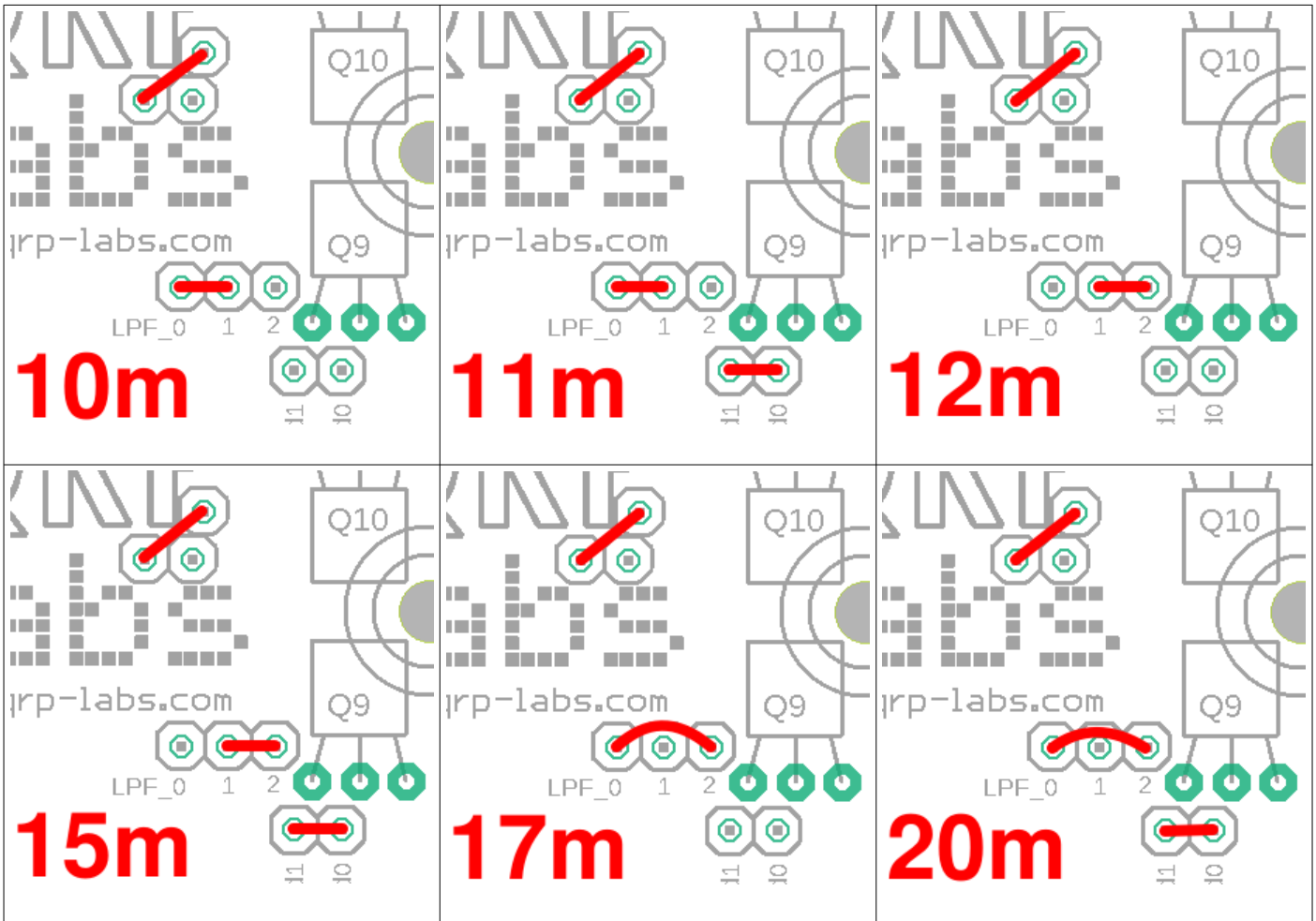
2.14 Install jumper wires

The jumper wire installations are necessary to inform the firmware to set up the band configuration parameters according to the band you are building for. The firmware will automatically configure your QDX-M the first time you power-up, with the correct parameters for the band you specify.

It's critical to install these jumper wires correctly for your desired band of operation.



The jumper wire locations are to the left and below the BS170 transistors and QRP Labs logo. Use pieces of component wire (capacitor off-cuts) formed into an inverted 'U', and install them according to the correct image below, for your band.

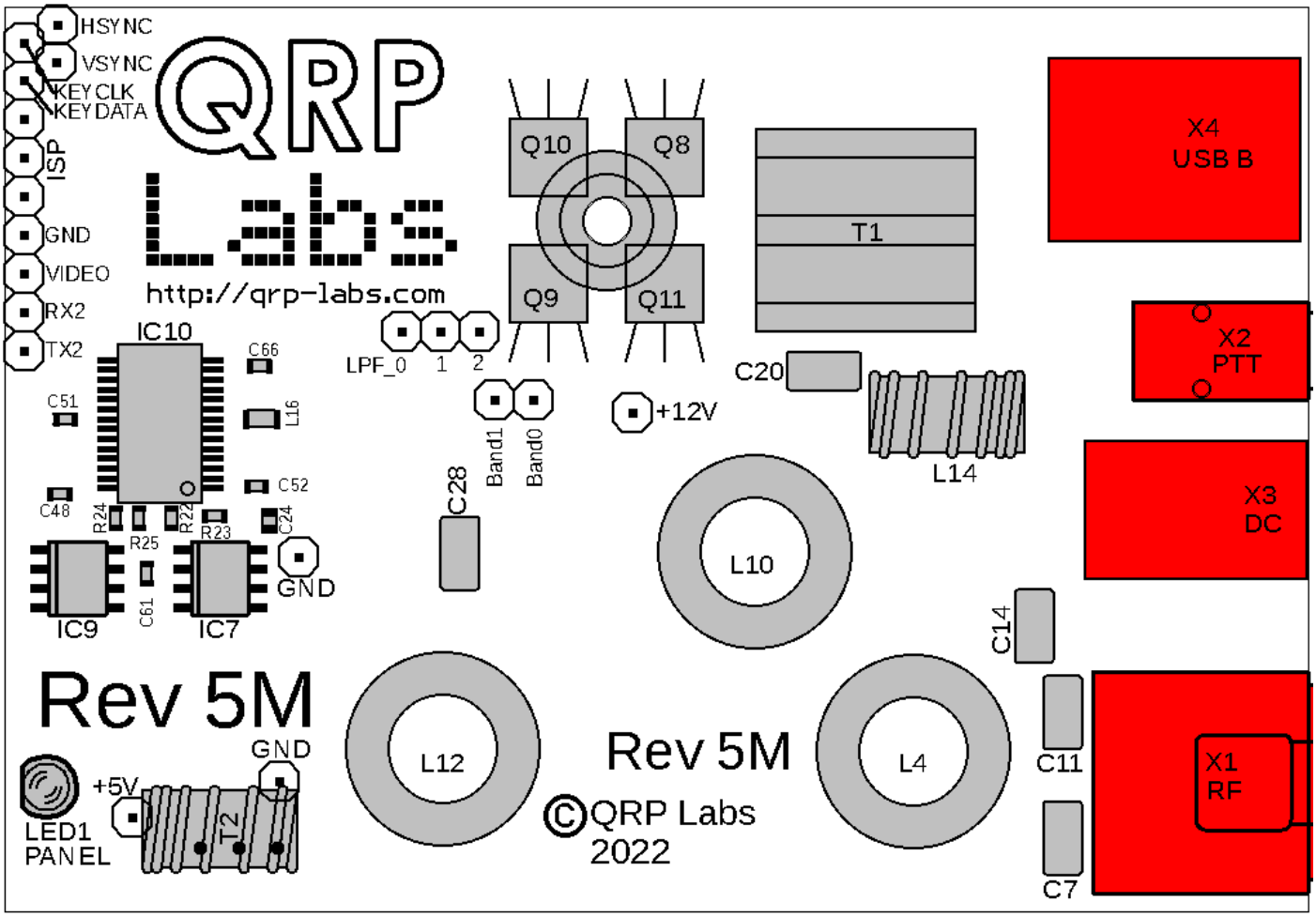


2.15 Install connectors

The final components to install are the connectors, from top right to bottom right:

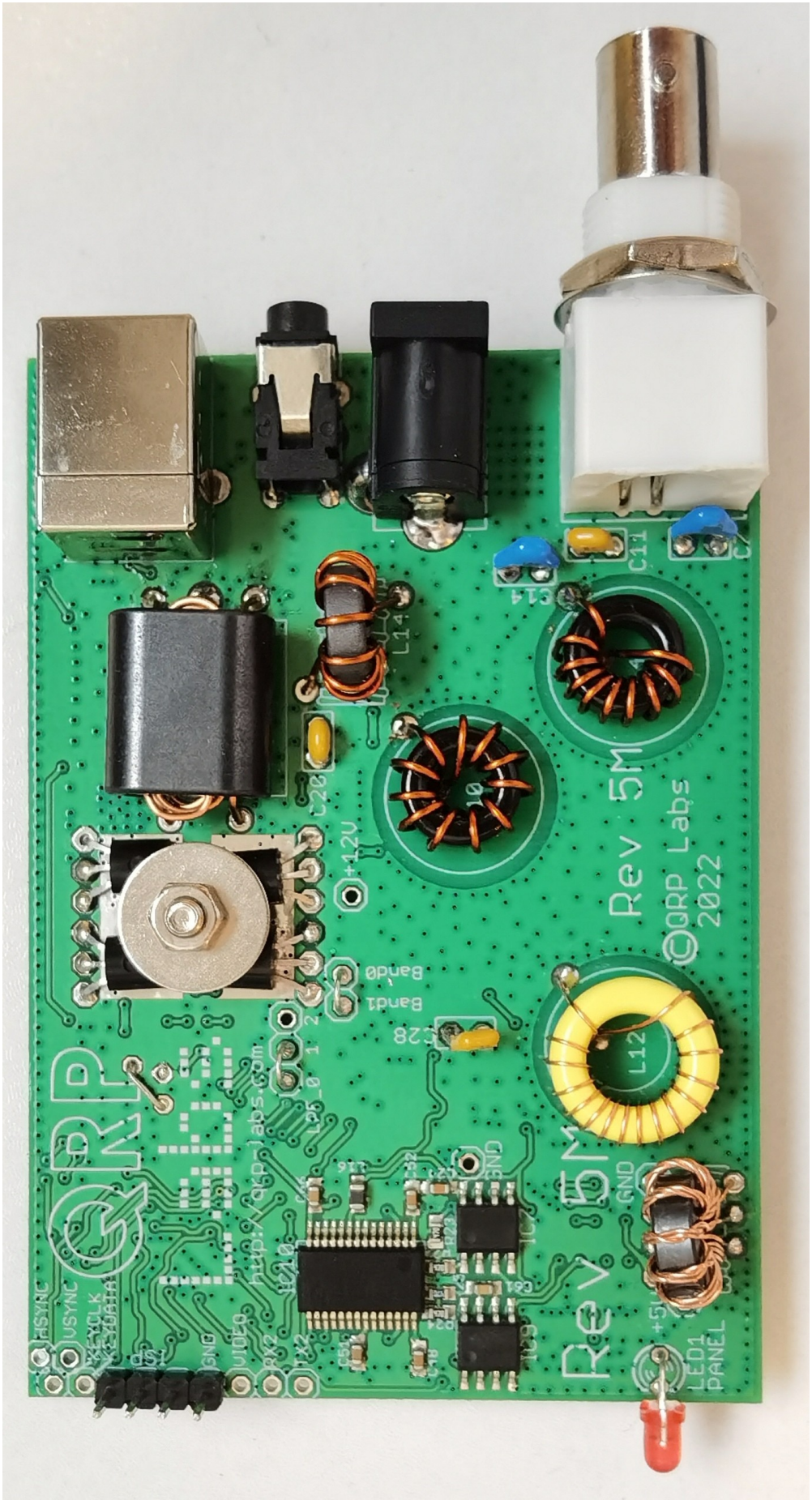
- X1: USB B connector
- X2: DC Power connector
- X2: PTT (mis-named X2, should be X3)
- X4: BNC RF connectors – NOTE: there is a footprint for an SMA connector that you may install if you wish; however this will have gaps around it as it pokes through the enclosure.

It is very important to install these carefully aligned so that they are straight and at right-angles with respect to the PCB. The connector body should not protrude beyond the edge of the PCB. When the connectors are well-aligned, you will have no trouble fitting the PCB into the optional enclosure. Clip off excess leads and tabs to make sure nothing protrudes more than 1.5mm.



Assembly of the QDX-M kit PCB is now complete!

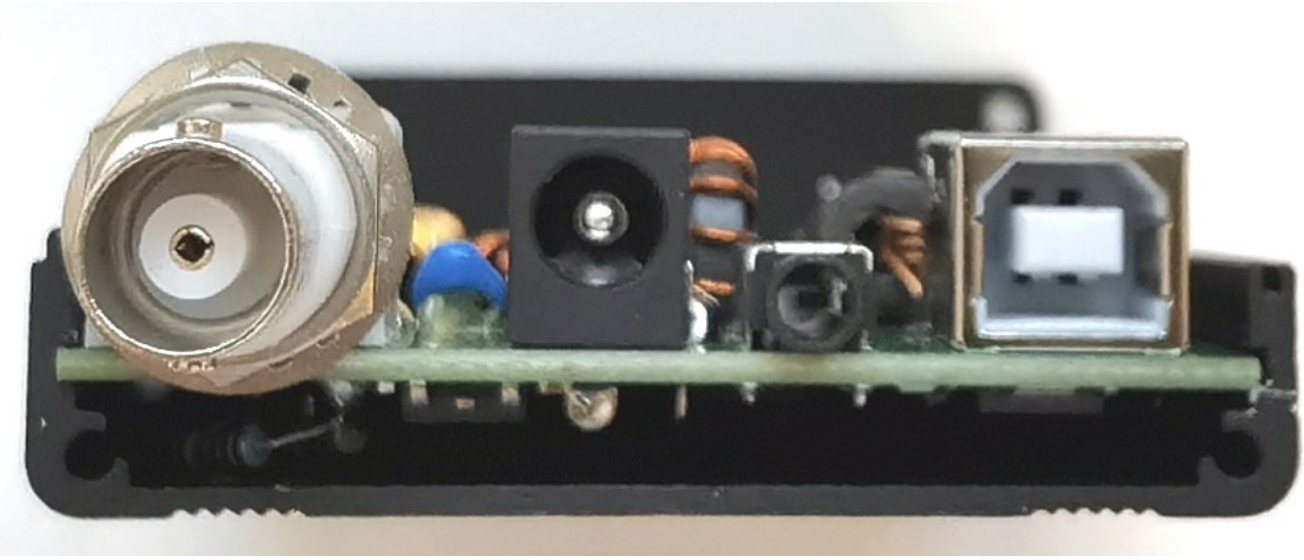
(Rev 5M PCB, 11m version shown – pin header at bottom left is on this particular board only. Not yours.).



2.16 Optional enclosure

Installation of the QDX-M PCB in the optional enclosure is very straightforward and self-explanatory.

1. Install the screws at the bottom left and bottom right of the front panel, fixing it to the bottom half of the enclosure. Note that the enclosure top and bottom are identical.
2. If the short edges (front and back, i.e. LED status edge, and connectors edge) of the PCB feel rough, it can be a good idea to gently file them so that the PCB edge is square.
3. Gently slide the PCB into the guide-rail grooves in the enclosure bottom.



4. Install the two screws at the bottom left and bottom right of the rear panel, fixing it to the bottom half of the enclosure.
5. Fit the top half of the enclosure. Note that one side has an “I-tongue” and the other side has a “U-groove” and this is repeated on both halves (top and bottom) of the enclosure. Therefore the top half will only fit the bottom half, “one-way-round”. If it does not appear to sit correctly, rotate the top half 180-degrees. Secure the top with the remaining screws.
6. Finally you may, if you wish, fit the nut to the BNC connector. I generally leave off the washer. The nut may be omitted too, it is not really required and should in any event not be over-tightened since there is a small gap between the back surface of the rear panel, and the BNC connector body. The washer can be installed on BNC connector the INSIDE of the enclosure to fill some of that gap. It’s a matter of trying it out and seeing what fits best in your case and gives you a result you’re most happy with.
7. If desired, attach the self-adhesive rubber feet to the bottom of the enclosure in the corners.


QRP
Rev 5M
QRP Labs
2022
+12V
Band0
Band1
L14
L13
C11
C20
C28
GND
+5V
LED1
PANEL
dsp
VIDEO
RX2
TX2
HSYNC
VSYNC
http://qrp-labs.com

3 Design

3.1 QDX-M Summary

Traditionally it was normally assumed that transmission of digital modes requires a computer, to generate the audio tones, and an SSB transceiver, to modulate those to RF and transmit them via an antenna.

Yet – the widespread assumption that an SSB transceiver is REQUIRED is in fact absolutely incorrect, at least for a wide category of digital modes transmitting a single tone frequency shift keyed signal. This class of transmissions are NOT SSB, they are a single carrier, frequency modulated. Potentially this can provide simplification of the radio transceiver design, delivering both higher performance and lower cost.

Why is this important? An SSB transmitter design is necessarily relatively complex and a non-trivial undertaking for several reasons. The audio signal must be converted to RF by an SSB exciter. An exciter may commonly take the form of a superheterodyne architecture having a modulator, IF filter (crystals) and another mixer to convert to the final RF operating frequency. It then requires a linear amplification chain to raise the signal amplitude to the required power output at the antenna port. An alternative architecture is a phasing network exciter which makes use of phasing techniques to null the unwanted sideband, and converts directly from baseband to RF. Again it is followed by a linear amplification chain. In both cases considerable attention to the design is required, to minimize the residual carrier frequency, and the unwanted sideband.

Linearity must be maintained throughout the amplifier chain to the antenna port, to avoid splattering onto adjacent frequencies. Even though the audio signal is theoretically a single audio tone, harmonics of the audio tone may be present particularly if the audio level is too high, and the carrier suppression and unwanted sideband are never perfect; so there is plenty of potential for generation of intermodulation products, hence the need for good linearity throughout. A linear amplifier (for example, Class AB) has low efficiency compared to non-linear amplifiers such as Class C. Low efficiency means higher current draw for a given power output (less battery-friendly in portable scenarios), and higher heat production, probably requiring heatsinking of the PA transistors – increasing size, cost and weight.

Double Sideband (DSB) is even worse – the designs are simpler but here we have an equal amplitude unwanted sideband emission. Half the transmitter's power is wasted on this unwanted sideband, which potentially can interfere with users of adjacent channels. On receive, we suffer a 3dB degradation in noise floor at best, and at worst, interference from signals in the unwanted sideband. With the strong double tone (both sidebands) the potential for intermodulation products due to non-linearities in the amplification chain is higher. Except under optimal conditions, and even then with difficulty, two DSB transceivers cannot QSO with each other! A DSB transceiver can only QSO with an SSB transceiver at the other end. DSB is the only mode which cannot even communicate with itself! Some would say, an abomination of a mode.

If there were a way to convert the audio tones from the computer, to RF, without going through the SSB modulation process and linear amplification, it could solve many problems.

Back in 2012 the QRP Labs Ultimate QRSS/WSPR/Digi transmitter kit was the first kit to generate and transmit digital signals standalone with no PC required, and without an SSB transmitter. It used just a microcontroller calculating the required frequency shifts and commanding an RF signal generator – then the AD9850 Direct Digital Synthesis (DDS) chip – to produce the required frequencies. Several iterations of the kit evolved over the years, to the current Ultimate3S http://qrp-labs.com/ultimate3/u3s which can transmit quite a list of digital modes in the standalone beacon application.

However for QSO digital modes, using computer (PC) software to encode and decode the audio, SSB transceivers remained the norm.

In the second half of 2019, I was thinking about FT8, and a number of simple low cost FT8 transceiver kits that were available (DSB transceivers), none of which offered a particularly good performance. I realized that there IS a better way to do this, that will dramatically increase the performance without increasing the price. Straight away I designed QDX and built the prototype.

The fact that the computer is generating the audio, does not need to stop us from generating a single RF tone from an RF signal generator, just as the Ultimate3S QRSS/WSPR/Digi kit does. A microcontroller can analyze the audio arriving from the PC, and determine the audio frequency of the tone; then add that to an RF base frequency (which on an SSB radio, we would call the "USB Dial Frequency"), to determine the correct RF carrier frequency to generate for transmission. The rest is then a matter of commanding an RF generator to produce the calculated frequency, and amplifying it for presentation to the antenna. The generated signal is absolutely clean, having only harmonically related unwanted spurious emissions which can be taken care of in the conventional way by Low Pass Filtering.

The importance of this technique cannot be understated.

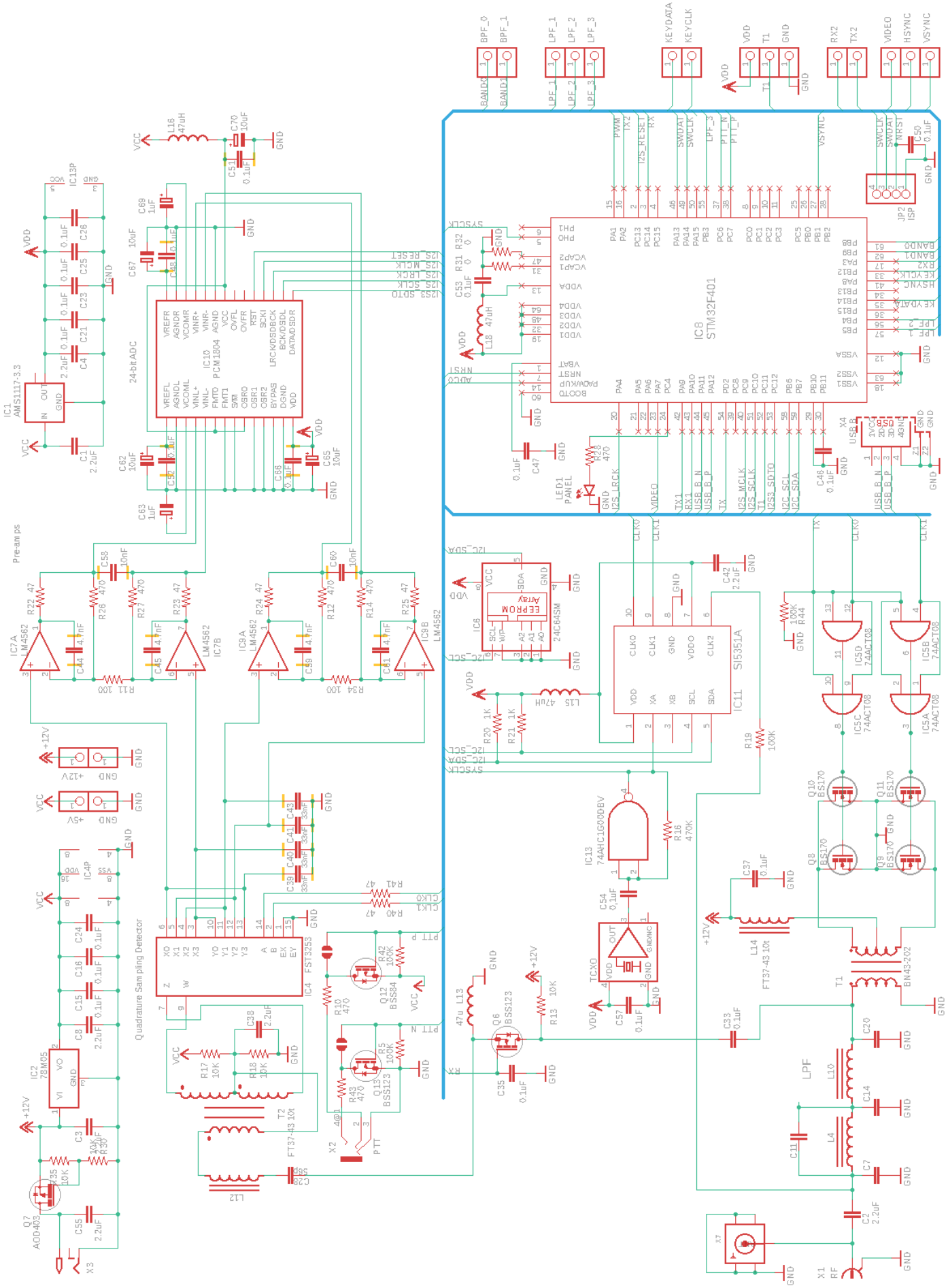
- We no longer need a complex SSB modulator involving multiple mixers, IF filters (or phasing in the case of the direct method).
- No more linear amplification chain – we can use higher efficiency non-linear amplifier classes, reducing the need for heatsinking, and lowering transmit current, very battery friendly.
- The unwanted sideband simply DOES NOT EXIST. It isn't just "there but attenuated" depending on how well the transceiver was designed and aligned - it just isn't there at all.
- Similarly the carrier also does not exist. Carrier suppression in a modulator depends on how well the mixer is balanced and is never perfect. But here, we generate a single tone directly – and there's NO SSB carrier at all. Zero.
- It is impossible to overdrive this design by turning up the audio volume too high on the PC. Therefore it is impossible to generate a messy signal that splatters on to adjacent frequencies.

With this in mind, the rest of the QDX transceiver design falls into place, around this central concept.

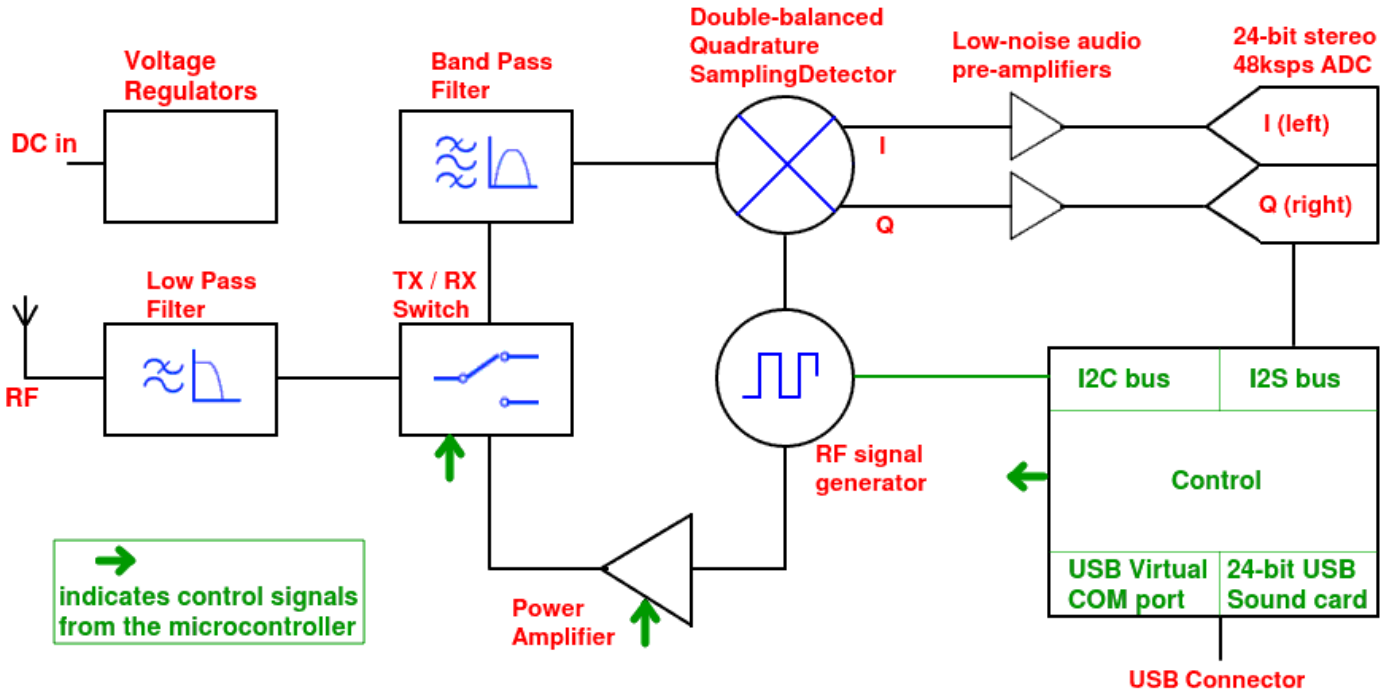
QDX-M – the mono-band QDX

QDX-M evolved from the multi-band QDX, to cater for the MF/LF bands 160m, 630m and 2200m. It was found through experiment – with not unexpected results – that the the PIN diode switching in QDX does not work well enough at such low frequencies, 2MHz and below. This is a normal result for PIN diodes. Accordingly the QDX-M was designed, based on the original QDX, but simply deleting the band-switching (Low and Band pass filters).

The rest of this theoretical section is based on the QDX but edited for QDX-M by removing all references to band-switching to simplify it. If you would like to understand the original band-switching in the QDX, then please refer to the QDX manual at http://qrp-labs.com/qdx. In the rest of this design section, the terms QDX and QDX-M are used interchangeably.



3.2 Block Diagram



QDX-M consists of the following circuit blocks:

1. A synthesized Local Oscillator using the famous Si5351A Digital PLL signal generator chip. A 0.25ppm 25MHz TCXO is included as standard as the PLL reference.
2. Low Pass Filtering to attenuate emissions at harmonics of the operating frequency. The filters is built using component values appropriate to the band of operation.
3. A solid state transmit/receive switch – no bulky, noisy, expensive, unreliable relays here!
4. A band pass filter to provide some degree of protection to the mixer against out of band signals – again built for the band of operation
5. A high performance, double balanced Quadrature Sampling Detector to mix the incoming RF signals to baseband and produce I and Q signals.
6. A low noise pre-amplifier for the I and Q signals.
7. High performance 24-bit 48ksps (Kilo Samples Per Second) stereo Analog To Digital Converter (ADC) chip with 112dB of dynamic range.
8. An embedded Software Defined Radio (SDR) receiver which implements, digitally, a 12kHz Intermediate Frequency superheterodyne receiver with excellent performance and unwanted sideband suppression, also implementing a sharp digital filter.
9. An embedded 24-bit 48ksps stereo USB Sound card – no more noisy, cluttered audio cables!
10. A CAT control serial interface, also over the same USB cable, to allow the PC software to control the radio (frequency, transmit/receive switching etc) in a standard way.
11. The microcontroller performs single cycle frequency analysis of the audio tone, and commands the Si5351A Signal generator to produce the necessary RF frequency.
12. A Class-D Push-Pull power amplifier that is small, low cost, high efficiency, and produces very low even harmonic output levels, reducing the demands on the Low Pass Filter.

13. Voltage regulation and supply decoupling.

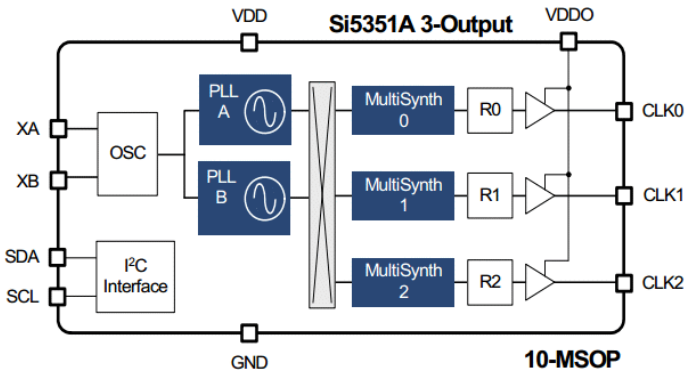
In subsequent sections, each of these blocks will be described in detail.

3.3 Synthesized local oscillator

Generating a stable, precise oscillator signal was one of the most challenging aspects of transceiver design. Modern semiconductors however, make it one of the easiest building blocks of the radio.

The synthesizer used in the QDX-M is the same as in the QCX 5W CW transceiver, and various other QRP Labs products: the Si5351A. This is a Digital Phase Locked Loop (PLL or DPLL) synthesizer which provides three separate frequency outputs, each having a frequency range spanning 3.5kHz to 200MHz. The frequency stability is governed by the a crystal reference.

Due to unavailability of the Si5351A, the equivalent MS5351M may be used. For test details on the performance of Si5351A vs MS5351M demonstrating the suitability of the MS5351M (in fact, slight superiority in many regards), please see http://qrp-labs.com/synth/ms5351m.html



The block diagram (right) is taken from the SiLabs Si5351A datasheet. Briefly, the 27MHz reference oscillator is multiplied up to an internal Voltage Controlled Oscillator in the range 600-900MHz (the PLL), then divided down to produce the final output frequency. The multiplication up and the division down are both fractional and so the frequency resolution is extremely finely controlled. The chip has two PLLs and three output divider units.

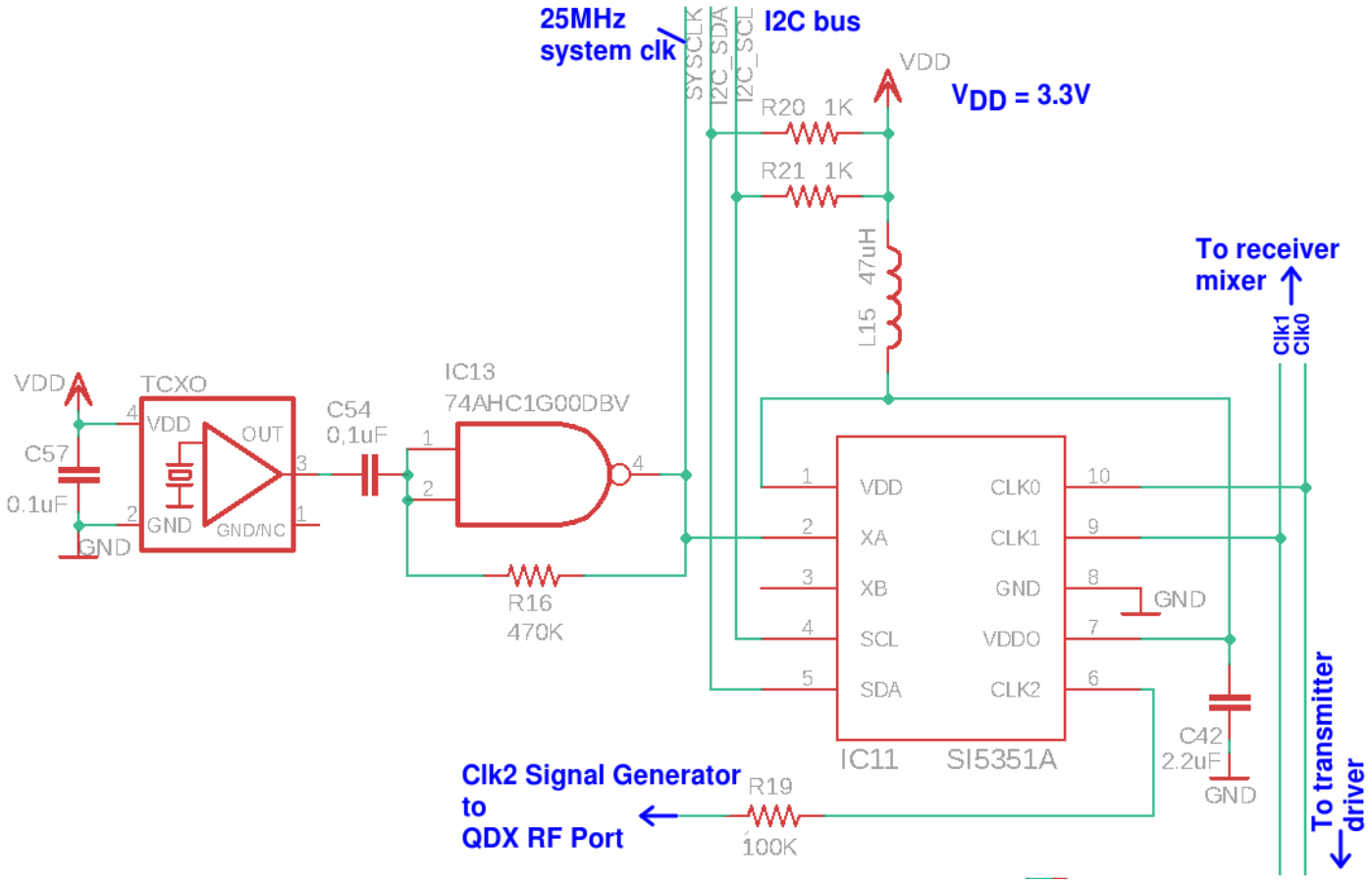
For best jitter performance, the Si5351A datasheet recommends the use of even integer dividers (no fractional component) in the MultiSynth dividers and in the QDX-M transceiver design, this recommendation is followed.

The Si5351A datasheet dictates the use of a 25 or 27MHz crystal or oscillator. In the QDX a 25MHz crystal was originally used, together with an unusual NPN/PNP transistor oscillator circuit, which produces a near rail-to-rail (3.3V peak-peak) output that is perfect for driving both the Si5351A and the STM32 processor. Temperature stability of the crystal frequency was found to be inadequate. Therefore a high performance TCXO is used. A single NAND logic gate chip is used to buffer the 1.1Vpp signal from the 25MHz TCXO to a 3.3V squarewave.

The Si5351A has a large number of internal 8-bit registers to control the synthesizer configuration and output frequency, and these are programmed by the microcontroller using the I2C serial protocol. 1K resistors R20 and R21 are pull-ups required for the operation of the bus at 400kHz.

There are three outputs of the Si5351A synthesizer and these are all used to good advantage. The Clk0/1 outputs are used to drive the Quadrature Sampling Detector (QSD) during receive.

A feature of the Quadrature Sampling Detector is that either the RF input, or the LO input, must provide two paths in 90-degree quadrature. This is normally applied at the Local Oscillator where it can be easily controlled for best performance. So, two oscillator signals are required, with the same frequencies but a precise 90-degree phase offset.



Generating this quadrature Local Oscillator signal is always difficult. Analogue phase shift circuits have limited accuracy. Often a divide-by-4 circuit is used, to produce quadrature oscillator outputs from an oscillator input at 4x the reception frequency. This also creates challenges particularly as you try to increase the reception frequency to cover higher bands. For example, on 10m e.g. 30MHz, a local oscillator at 120MHz is required and the divide-by-4 circuit must be able to operate at such a high frequency. Devices such as the 74AC74 can do so, but pushing it higher into the 6m band cannot be done with the 74AC74.

The Si5351A has a phase offset feature, which is not really very clearly described in the SiLabs documentation. However, QRP Labs has perfected the technique to put two of the Si5351A outputs into precise 90-degree quadrature, which is maintained without tuning glitches as the frequency is altered. It's a nice development because it eliminates one more circuit block (the 74AC74 divide-by-4 circuit), again reducing complexity and cost. To the best of my knowledge this is the first time the Si5351A has been implemented in a product directly driving a QSD with two outputs in quadrature (no divide-by-4 circuit).

The Clk0/1 Si5351A outputs are also routed to the transmit driver. During transmit, Clk0 and Clk1 are configured with a 180-degree phase difference, this allows them to cleanly drive the QDX RF power amplifier, which is a Push-Pull configuration. It's a very neat way to obtain the necessary push-pull driver signals in a highly symmetric and low parts-count way.

Finally the Clk2 output is used as a separate signal generator, it is routed back to the QDX transceiver's RF port via the 100K resistor R19. This signal generator can be used by the QDX diagnostic and self-performance measurement tools, to sweep the audio filter and sweep the RF bandpass input filter. Clk2 is normally switched off, and when these functions are needed, it is enabled in conjunction with Clk0/Clk1 operating the receiver mixer.

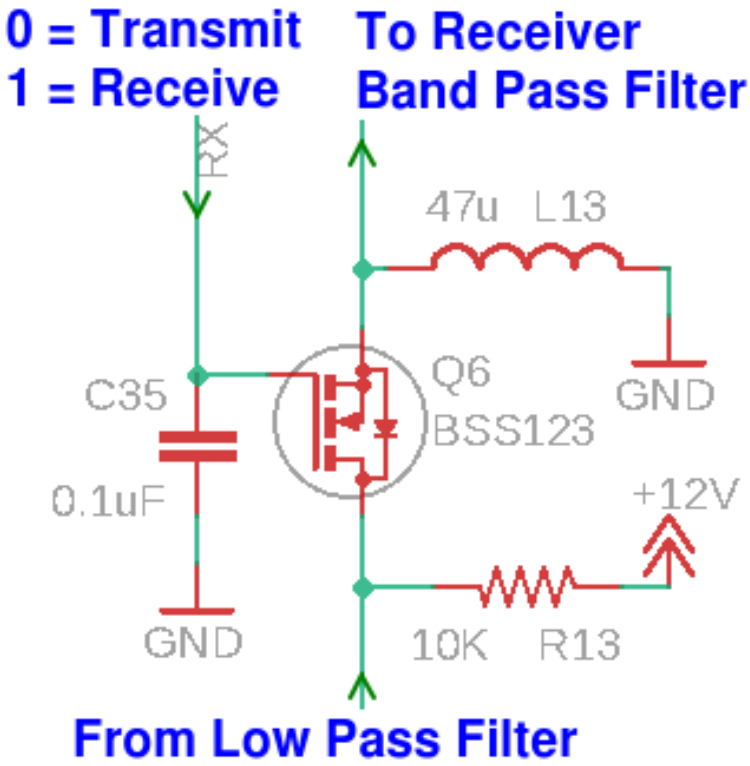
An additional function of the 25MHz TCXO and its NAND gate buffer amplifier, is to provide the precise 25MHz system clock for the STM32. The STM32 microcontroller contains its own internal PLL clock generation systems that are highly configurable. In QDX, the 25MHz reference oscillator is fed to the STM32 external clock input, and the STM32 is configured to provide:

- 24.576 MHz clock for the 512 fs input to the 24-bit stereo I2S ADC chip (24.576 / 512 = 48,000) – fs is the number of clock pulses for each analog to digital conversion.
- 48 MHz clock for the STM32's USB peripheral to enable it to provide USB functionality for the USB sound card and the CAT Virtual COM serial port.
- 72 MHz for the main CPU clock

None of these signals are absolutely precise but they are very close to the target values and well within specified allowable errors for the protocols involved (USB, I2S). The 72 MHz CPU clock is a little below the maximum allowable 84 MHz for this processor, but this is acceptable since the performance at 72 MHz is plenty adequate to achieve all desired functionality for QDX.

3.4 Solid state transmit/receive switch

During transmit, the receiver must be disconnected from the RF port of QDX because a 5W signal is 45V peak-peak which would be too much for the receiver input mixer IC3.



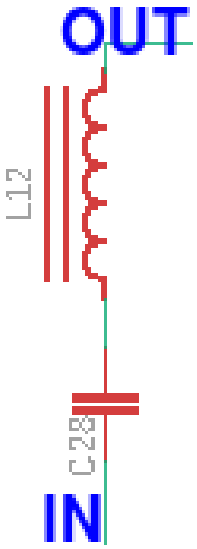
A spectacularly high "Off" isolation is not required, only sufficient attenuation to protect the mixer. During transmit, the Digital Signal Processing of the I & Q channels is suspended and there is no audio output of the receiver. In the QDX application a totally clean click-free transition between Receiver and Transmit states is not needed – a welcome relaxation compared to the demands of a CW transceiver.

The transmit/receive switch is implemented by a single BSS123 MOSFET. The source is at DC ground (via 47 uH inductor L4). The control signal from the microcontroller switches the MOSFET on or off. Capacitor C35 close to the MOSFET gate is found to be necessary to prevent inductive pickup of the 5W RF from partially switching on the MOSFET.

During receive, the transmitter drivers are all at zero potential, so the transmitter power amplifier transistors are in a high impedance state and do not affect the receive operation.

This simple transmit/receive switch very conveniently avoids the need for relays. Relays may be great for some applications but if a relay was used here for transmit/receive switching it would add bulk, weight, cost, and reduce reliability.

3.5 Receiver band pass filter



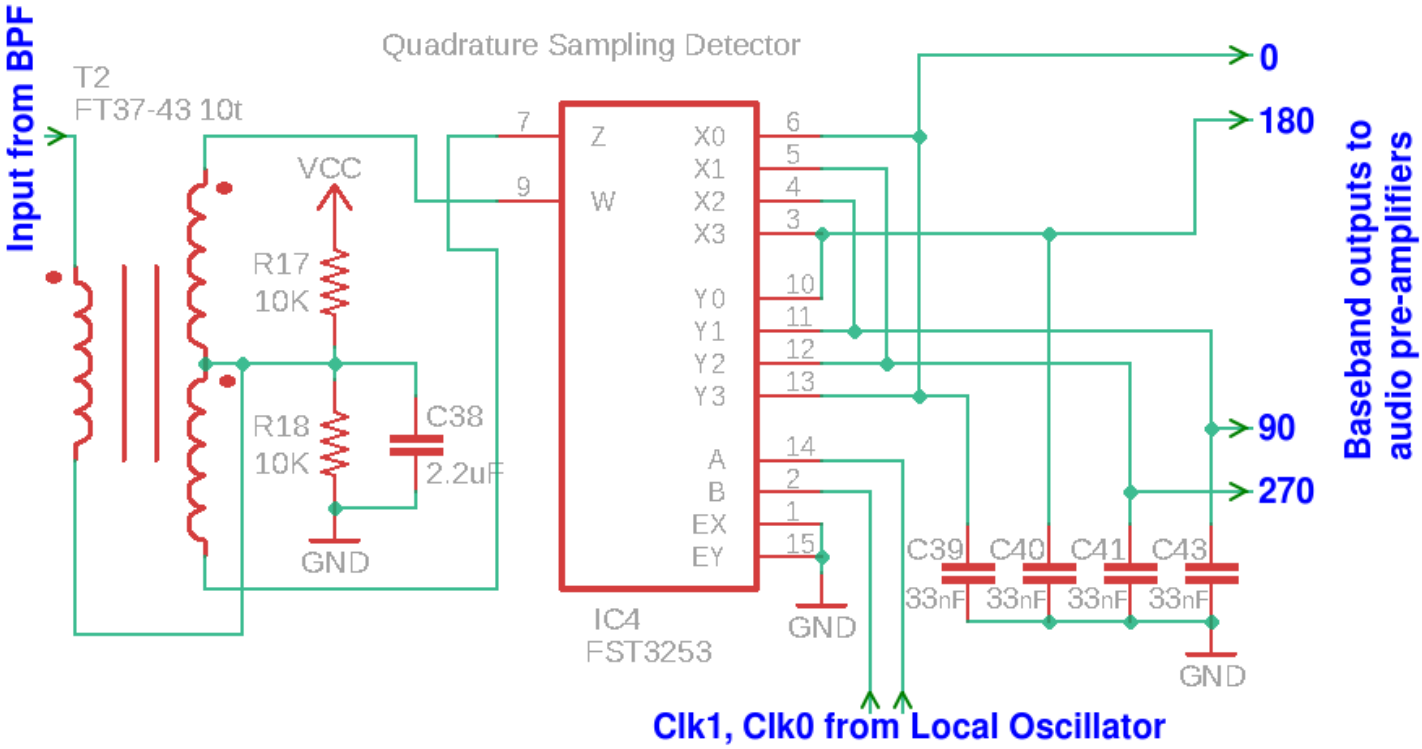
In any receiver, performance is improved by excluding strong out of band signals using a band pass filter. In QDX-M a rather simple series resonant L-C filter is used, which although limited, still provides some useful protection to the mixer. The mixer is a Quadrature Sampling Detector which is a high dynamic range, high IP3 mixer and therefore has excellent intermodulation performance. Together, the BPF and QSD provide a high performance receiver front end.

Inductor L and capacitor C values are chosen such that the LC resonance is centered on the digi modes section of each band.

The whole BPF is DC-biased to 2.5V midrail via the trifilar phase splitting transformer T2 of the Receive mixer.

More sophisticated Band Pass Filters could be used but would come at the expense of complexity, board area, and cost.

3.6 Double-balanced Quadrature Sampling Detector



This circuit implements a double-balanced Quadrature Sampling Detector, mixing from RF to baseband. The FST3253 is a dual 1:4 multiplexer which is often seen in QSD circuits. It has fast switching times and very low on resistance of only a few ohms. The input signal is switched by the quadrature LO to each of the four integrating capacitors C39, C40, C41 and C43 in turn, for 90-degrees of the RF cycle each. The result is that the audio difference (beat) between the RF input and LO input appears across each of the four integrating capacitors, with four phases at 0, 90, 180 and 270 degrees.

The 33nF capacitors and the low source resistance results also provide a relatively fast roll-off of the audio response. This is effectively a narrow band pass filter since any incoming RF more than a few 10's of kHz away from the LO frequency is greatly attenuated. The QSD is inherently a very

high performance mixer design with high third order intercept and dynamic range, and low loss (0.9dB).

The FST3253 dual switch is often connected with the two switches simply paralleled together (which does halve the switch ON resistance). But I prefer the double-balanced mixer configuration which provides higher performance. The double-balanced configuration requires two RF inputs 180-degrees out of phase (opposite to each other). This is provided here by trifilar-wound transformer T2.

R17, R18 and C38 provide a 2.5V DC bias to the RF signal through the mixer and audio pre-amplifiers. It also provides mid-rail DC bias to the Band Pass Filter component switches in the previous section. This simple bias does not source or sink any significant current due to the balanced nature of the system, therefore no buffering is required.

Note that the Clk0 and Clk1 signals from the local oscillator have 47-ohm resistors in series; some people say this reduces ringing artifacts in the detector though I have not been able to find any evidence for this.

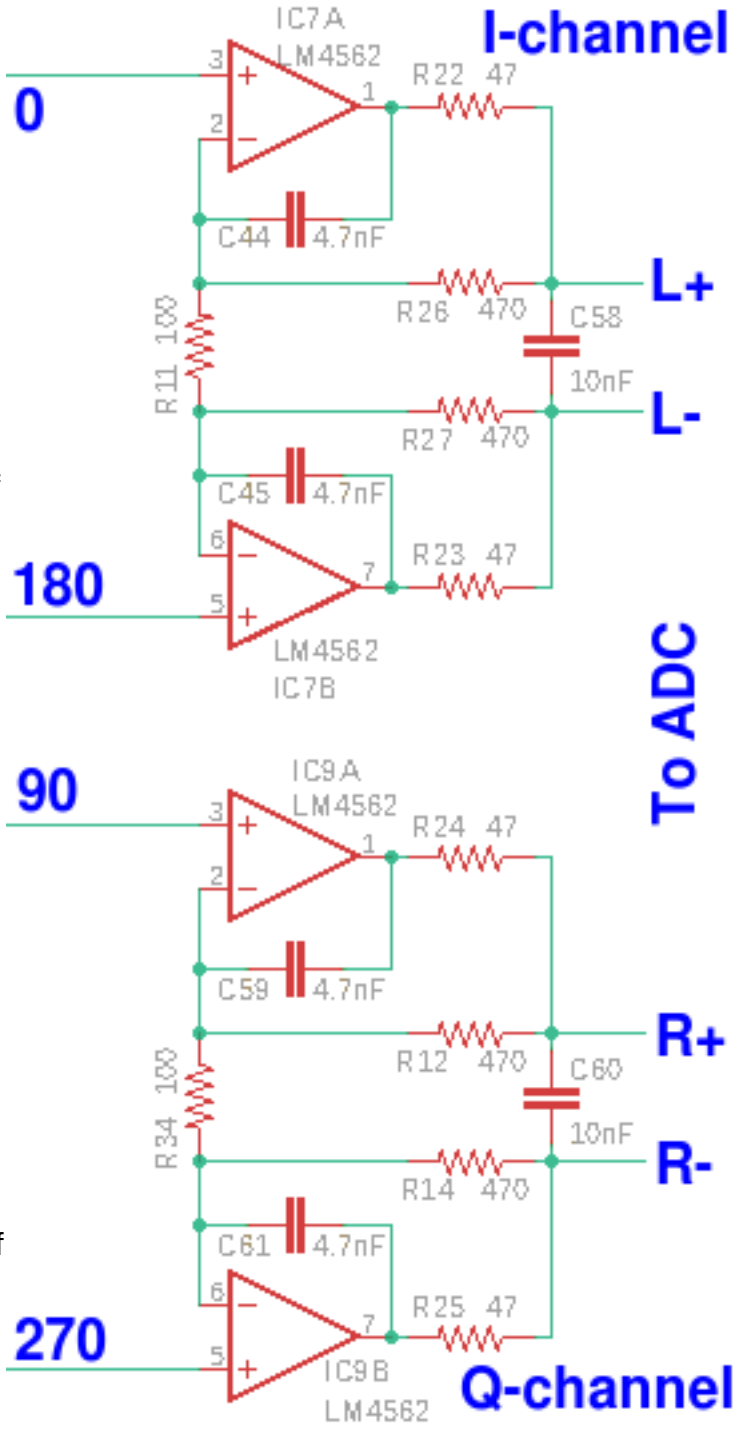
3.7 Low noise balanced differential pre-amplifiers

The same LM4562 dual operational amplifier chip is used in QDX as is used in the QCX-series CW transceivers and QRP Labs receiver module. This is a low noise op-amp providing an excellent combination of high performance and low cost.

Low noise is important for the pre-amplifier because it determines the overall sensitivity of the receiver (in conjunction with losses of the RF front end).

This part of the circuit was substantially redesigned for the Rev 3 QDX PCB, due to the fact that the formerly used AK5386 ADC chip had become obsolete. The chosen replacement, PCM1804 claims even higher performance than the AK5386 but the PCM1804 has a differential input. One input could be ignored, feeding the I and Q signals into only one terminal of each ADC (left and right); however, this would sacrifice 6dB of the potential 112dB claimed dynamic range of the PCM1804.

The circuit used here is derived from the instrumentation amplifier topology that usually has three op-amps; the third op-amp being a unity-gain difference amplifier of the outputs of the first two. An instrumentation amplifier arrangement brings several key benefits in this application:



- Excellent balance means common mode noise signals are canceled out
- Very high input impedance makes the loading on the Quadrature Sampling Detector capacitors very light, as well as balanced
- Provides positive and negative phase I and Q signals suitable for feeding the differential PCM1804 ADC chip

The disadvantage is that we require twice the number of op-amps and more resistors and capacitors – but you know what they say about free lunches.

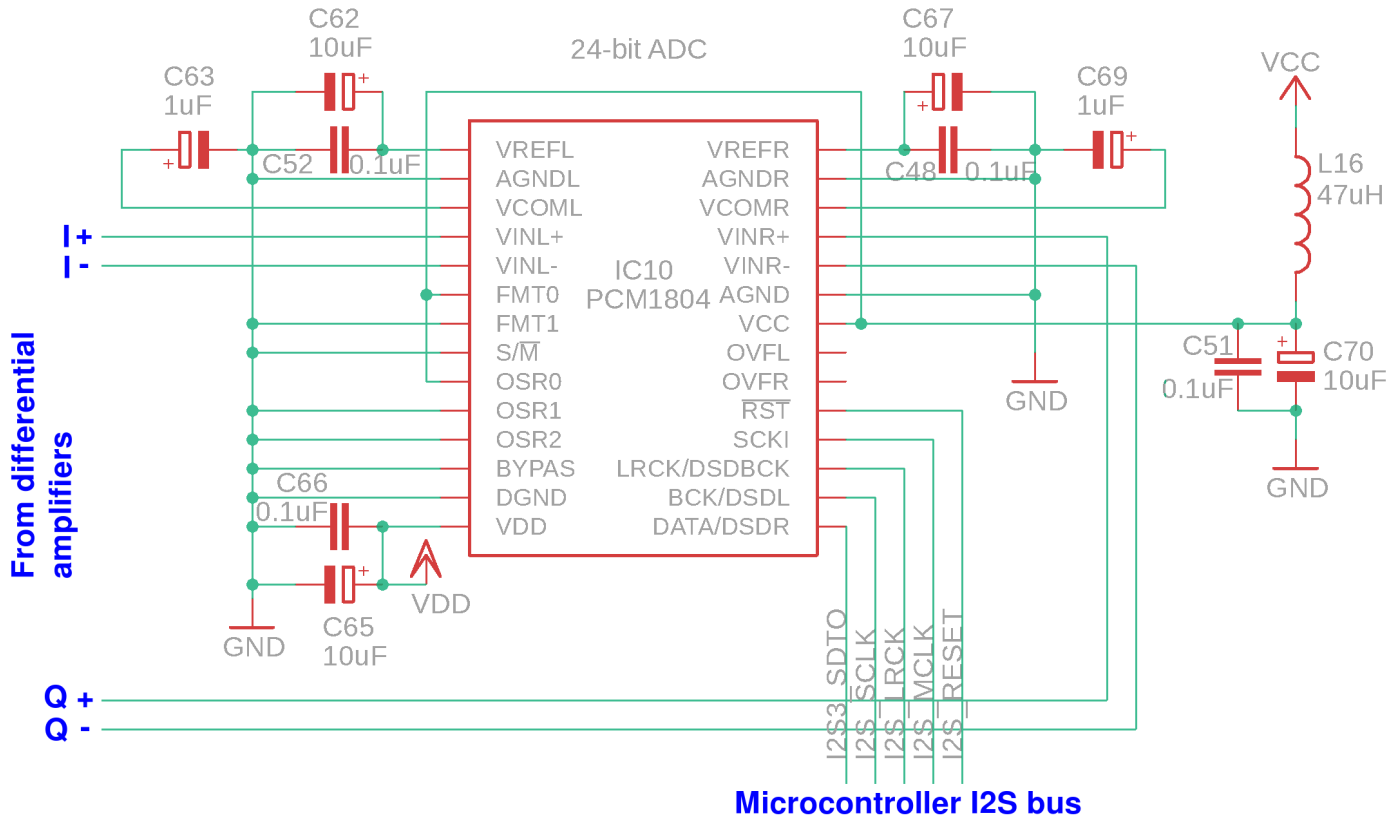
In the circuit shown above, the ADC chip itself, having differential input, performs the difference function so no 3rd op-amp is required in the instrumentation amplifier. The “Common” input was used with a resistor network to provide DC bias to the instrumentation amplifier inputs. However it was found that in circumstances the LM4562 op-amps could “lock up”, particularly IC7, with its output near the supply rail. On Rev 4 PCBs the DC blocking capacitors were shorted out post-production; on Rev 5 and above, the DC blocking capacitors and the bias network are removed completely; the DC mid-rail bias is fed through all the way from T2.

The 4.7nF capacitors provide high frequency attenuation beyond the 48kHz sampling bandwidth. The 10nF capacitors C58 and C60 are recommended by Texas Instruments in the PCM1804 datasheet and who am I to argue.

The gain of the instrumentation amplifiers is only about 20dB and is chosen to best match the dynamic range of the PCM1804 ADC in the application.

The I and Q channel contain all necessary phase and amplitude information to allow the following SDR to demodulate any mode, although only single sideband is used in QDX-M.

3.8 Analog to Digital Converter



The PCM1804 Analog to Digital Converter (ADC) chip is a high performance stereo ADC chip with differential input. Its purpose is to convert the analog I and Q-channel signals to a digital

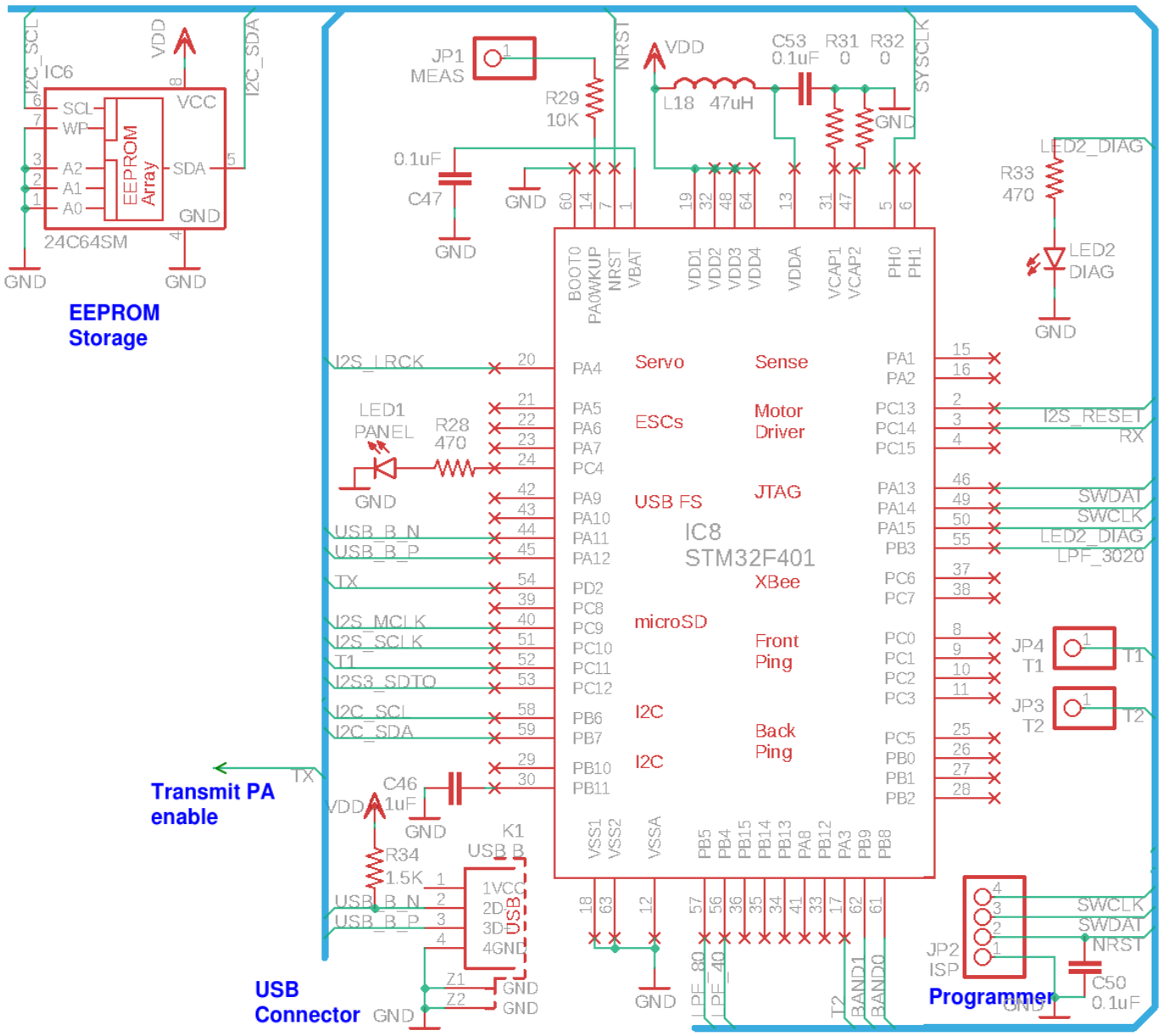
representation to allow digital signal processing by the microcontroller, which implements a Software Defined Radio (SDR). The PCM1804 specifies a 112dB dynamic range and is used at 48 ksps (kilo samples per second).

All the Texas Instruments PCM1804 datasheet recommendations in regard to layout and supply decoupling with tantalum capacitors, are followed (they most probably had good reasons for writing all that stuff and even if they didn't, I'm not about to take risks here).

In this design, the PCM1804 chip acts as Master of the I2S interface and the STM32 I2S peripheral acts as Slave. The STM32 generates a 24.576 MHz clock signal (I2S_MCLK) which is 512 x the sample rate (i.e. 48,000 x 512 = 24,576,000). From this, the PCM1804 generates the bit clock (I2S_SCLK) at 3.072 MHz (64 bits per sample – two 32-bit fields, for Left and Right channels, of which 24-bits are populated). The processor also generates the Frame clock (LRCK) at 48 kHz. The I2S_RESET signal is supplied by the microcontroller and must be activated once at power-up to reset the ADC.

3.9 Embedded Software Defined Radio

This schematic fragment contains the whole microcontroller section around the STM32F401RB microcontroller.



The microcontroller section of the schematic includes a 24C64 (8K x 8-bit) serial I2C EEPROM chip. Unlike the ATmega328 processors used in many other QRP Labs kits, the STM32 used here does not contain internal EEPROM so for storage of configuration parameters it is necessary to either use Flash storage or EEPROM. The problem with Flash storage is a limited 10,000 write-cycle specification which may be a little low. Therefore an external EEPROM chip is used.

The 4-pin programming header at the bottom right of the diagram is for factory use only to install the QFU bootloader on the QDX board. An onboard diagnostic LED is also used only for factory validation. LED1 is a front panel LED on the QDX which shows basic status information such as whether or not the unit is transmitting or receiving, or in bootloader (firmware update) mode. A transmit-enable output switches on the transmit driver to enable the RF power amplifier.

Other than that, all the magic goes on in the firmware, which is not open source.

The Software Defined Radio receiver (SDR) implements digitally a superhet receiver having a 12kHz Intermediate Frequency. The reason for doing this rather than processing directly at baseband as a direct conversion receiver, is that it eliminates any issues with power line hum – the harmonics of 50Hz or 60Hz (depending on your country) and other noise which exists around 0Hz.

The receive signal path processing contains the following stages implemented in the Digital Signal Processing (DSP):

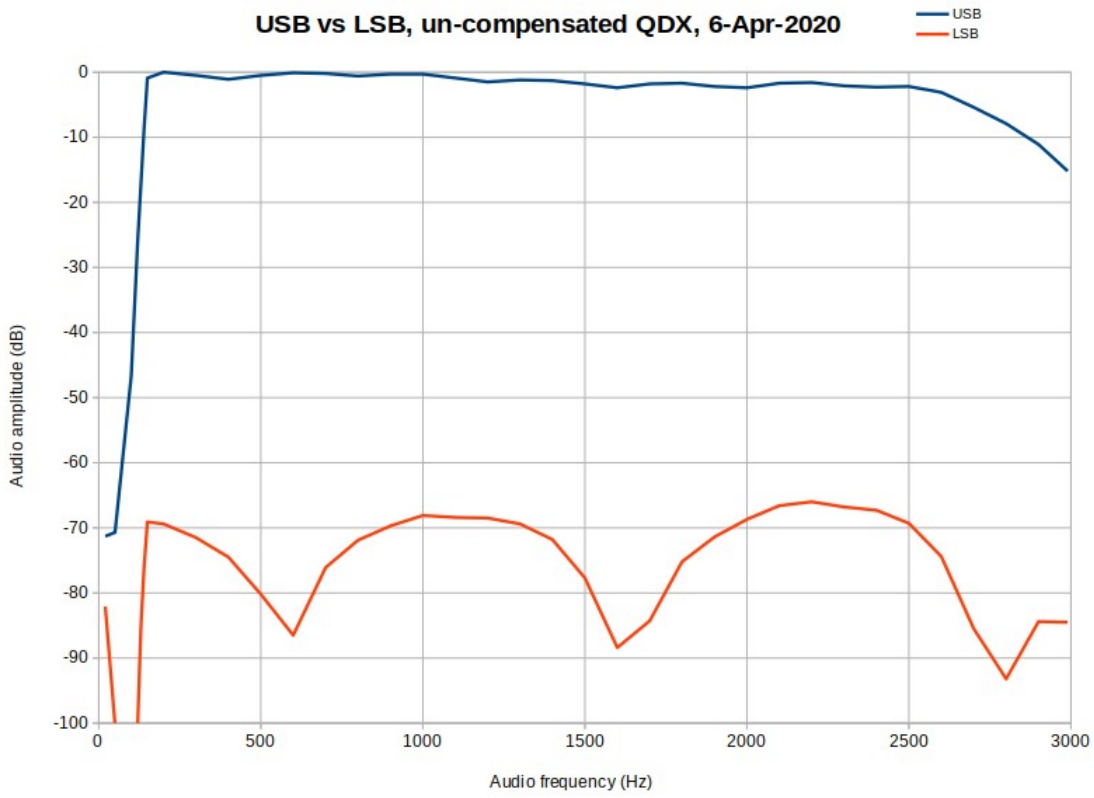
- Retrieve a block of 32 24-bit audio I and Q samples from the ADC chip via the I2S bus
- “Mix” to 12 kHz Intermediate frequency, including a Low Pass Filter in the factor-of-four decimation process; the reason for the choice of 12 kHz as the intermediate frequency is that the mathematics becomes relatively easy, as it is exactly one quarter of the 48 kHz sample rate.
- A Hilbert Transform applies a 90-degree phase shift to one of the I and Q paths relative to the other; this is the equivalent of the analog all-pass phase shift network implemented using op-amps in the QCX CW transceiver.
- Sum or subtract the two paths to produce either the Upper or Lower sideband; QDX is capable of demodulating either sideband but digital operations normally use upper side band (USB) which is therefore the default operating mode.
- Apply a digital audio filter, which has a passband from 150Hz to 3.2kHz
- Interpolate back to 48 ksps audio
- Send the 48ksps audio samples to the PC over the USB Digital Audio link.

The performance of the receiver was tested using the internal signal generator to inject a signal into the QDX input at a defined offset; Argo software was used on the PC to determine the received signal amplitude.

Two different measurement runs were performed, with quite some elapsed time between them (17 months!). In both cases, there is no change to the firmware but the measurements were performed on different prototypes. There is no attempt at amplitude or phase compensation, which does not appear to be needed (the results are already excellent).

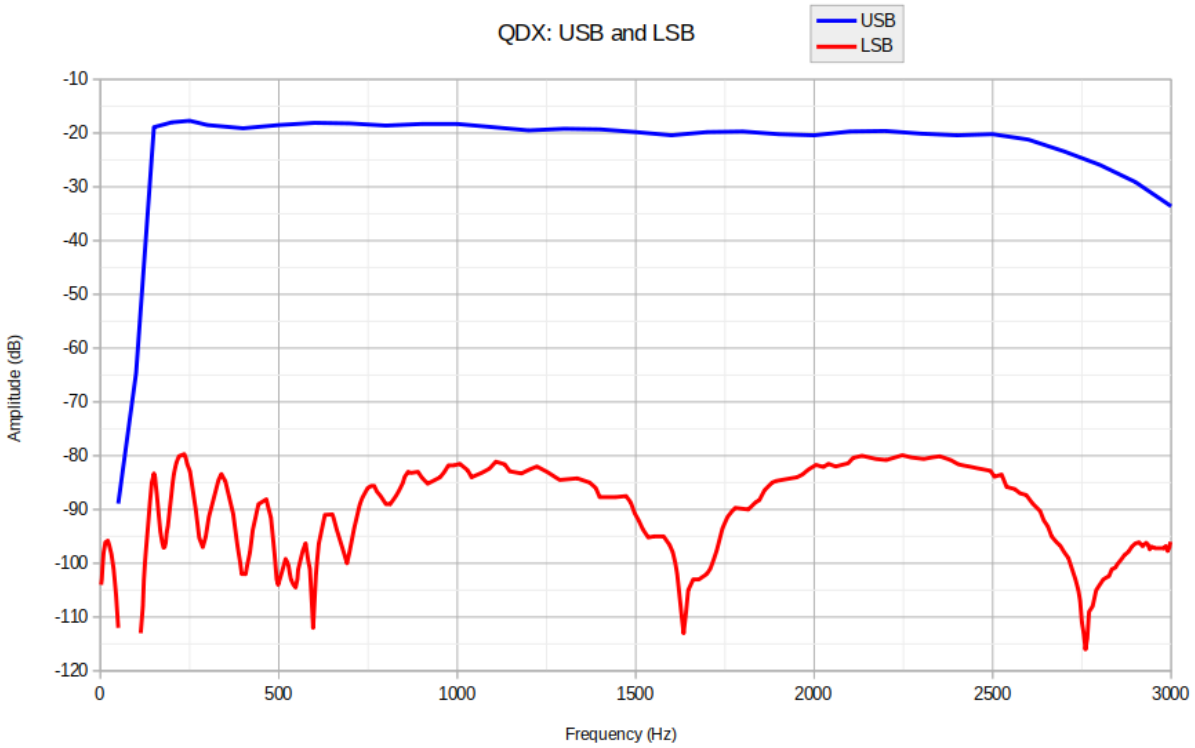
The vertical scale in dB has an arbitrary zero reference; the important point is the difference between the upper sideband (blue line), and the unwanted lower sideband (red line).

In the first example (shown below) dating from 06-Apr-2020, the worst case unwanted sideband suppression is at -64dB relative to the wanted sideband, which occurs at about 2,200Hz audio.



The second measurement (September 2021) shown below, shows a worst case unwanted sideband -60dB relative to the wanted sideband, again at about 2200Hz audio. The difference of 4dB between the two prototypes will have been due to component tolerances and build-style variations. In both cases, this is really good performance!

The second measurement was painstakingly made with much finer resolution (lower frequency steps) than the earlier measurement, which shows more clearly the behaviour of the Digital Signal Processing.



3.10 Embedded 24-bit 48ksps stereo USB Sound card

QDX has its own embedded 24-bit 48ksps stereo USB sound card! But there is not a real physical little piece of hardware strapped inside, like the PCB from a sound card dongle or something. Instead, the USB sound card is actually coded in firmware directly by the STM32 microcontroller, the same single microcontroller that is also doing everything else, in the design.

This is an extremely important design feature of QDX.

A radio transceiver having an audio input has a number of disadvantages:

- An audio cable is one more cable to have to plug in between your radio and your PC.
- There may be level mis-matches between the PC generating the audio signal, and the audio input of the radio. The PC will need to be carefully set up to set the volume such that the radio produces the expected RF proper power level, but does so without distortion (clipping, intermodulation distortion, splatter, however you want to refer to it). With the built-in QDX sound card, no level adjustment is necessary; set the volume to the max and rest assured that the QDX will never be overdriven (in fact it CANNOT be).
- Audio cables are a recipe for picking up noise, particularly in most of our lab environments where we power our equipment from grid power. You will often see 60 Hz (US) or 50 Hz (most of the world ex-US) harmonic lines on spectrum waterfalls. Ground loops can be hard to eliminate. With QDX and its built-in sound card, numeric audio sample data is transferred back and forth between the PC and QDX with zero loss, zero distortion, no hum or other noise. The quantization error due to sampling at 48ksps is minimal but is in any case matched by WSJT-X so a higher sample rate would not achieve an increase in performance. The benefit of direct digital numeric transfer between PC and QDX, without any conversion errors or noise, is enormous!

You can buy cheap USB dongles and connect them between your radio and your PC. But the dynamic range IS limited, and not just by the 6dB/bit theoretical relationship between bit depth and dynamic range, but also by the actual noise floor of the converter (the least significant bits are in other words, just noise).

In contrast, QDX uses a 24-bit stereo ADC chip with 112dB dynamic range specification. If you look at the specification of any expensive USB external sound card, and the price of getting anything like 112dB, you will appreciate the true significance of this.

All Digital Signal Processing in QDX is carried out at full floating point resolution, not limited to 16-bits as many SDR implementations do. In its communication with the PC this is rounded to 24-bits and transferred via the USB Digital Audio device class. WSJT-X only uses 16-bits so the PC sound system will automatically re-sample but still, the 24-bit USB sound is sent by QDX to support any future PC applications which use a higher resolution.

The result is a very high performance radio receiver, and equally, an excellent performance transmit side due to the zero-loss, zero-noise transfer of audio digitally from the WSJT-X synthesis straight to the QDX frequency analysis.

3.11 CAT control serial interface

QDX includes an emulated virtual COM (serial) port, all in the same STM32 processor that does everything else. The STM32 processor therefore actually includes THREE USB device classes:

1. Compound device class (a container for more than one other device classes)
2. Digital audio
3. Virtual COM (serial) port

The introduction of the compound device class makes QDX effectively, contain a virtual USB hub, to which is connected a Digital Audio USB device (the USB sound card) and a Virtual COM Serial Port. This allows a single USB cable to be used between QDX and the PC, the USB cable transfers both digital audio and serial data seamlessly between the two.

An important benefit of the CAT control interface is that it allows QDX to have a flexible tunable VFO and band switching, all without any actual physical controls. No physical controls are at all necessary because everything is done under the control of the WSJT-X software as CAT controller. Elimination of display and physical controls is not just a matter of lowering costs, it also makes the QDX extremely easy to use. Want to change bands? Simply click the drop-down in WSJT-X and QDX will automatically change bands. Want to use a different frequency (for example, change mode to WSPR, or perhaps even change software application to JS8Call)? The software application will automatically switch you to the correct frequency for the mode you select.

Another advantage of CAT control is that WSJT-X contains an option to effect the transmit/receive switch either by VOX (Voice activation, or in other words, the presence of audio) or by an actual CAT command to enable transmission. QDX also supports both methods, there is an entry in the configuration screen that allows VOX to be enabled or disabled. The default setting is “disabled”, which means a CAT command is necessary to switch from Receive to Transmit (and back).

The danger of VOX is that it is all too easy to accidentally have the QDX selected as the sound card for system noise outputs etc… then when a WhatsApp message from someone arrives, and the PC does its fancy “pling” sound, well that’ll be faithfully converted to a series of audio tones by QDX and transmitted accordingly. So disabling VOX and insisting on a proper CAT command to switch on to transmit, is the recommended modus operandi.

QDX also has a rich user interface via the Virtual COM port, by connecting a terminal emulator on a PC, to the virtual COM port. This is a whole blissful and enormous topic by itself, it allows interaction with all the built-in analysis tools in QDX as well as the configuration screen and features such as factory reset and firmware update.

3.12 Audio frequency analysis

Determination of the audio frequency arriving from the PC software (for example, WSJT-X) is key to the whole operation.

My first thought was a Fast Fourier Transform (FFT). However that quickly turns out to be a lot more difficult than it at first sounds. For the FT8 mode for example, the audio range is a 3kHz band. It isn’t so trivial to set up a FFT with 6.25Hz buckets, that wide, and would require a significant amount of processing power, such that I would need to use a more powerful (and expensive!) microcontroller than I’d really like. Furthermore WSJT-X modes are heavily optimized such that the period is the inverse of the tone spacing, so I might need to do TWO interleaved FFT analysis runs offset by half the tone spacing. Then there’s the problem of the frequency slide when

a tone changes. Furthermore, the QDX would have to know which mode was being transmitted, in order to configure the size (and hence speed) of the FFT buckets. I started to shudder and shake!

I soon realized that FFT would be the sledge hammer approach and there is a much simpler and faster way, which also offers higher accuracy.

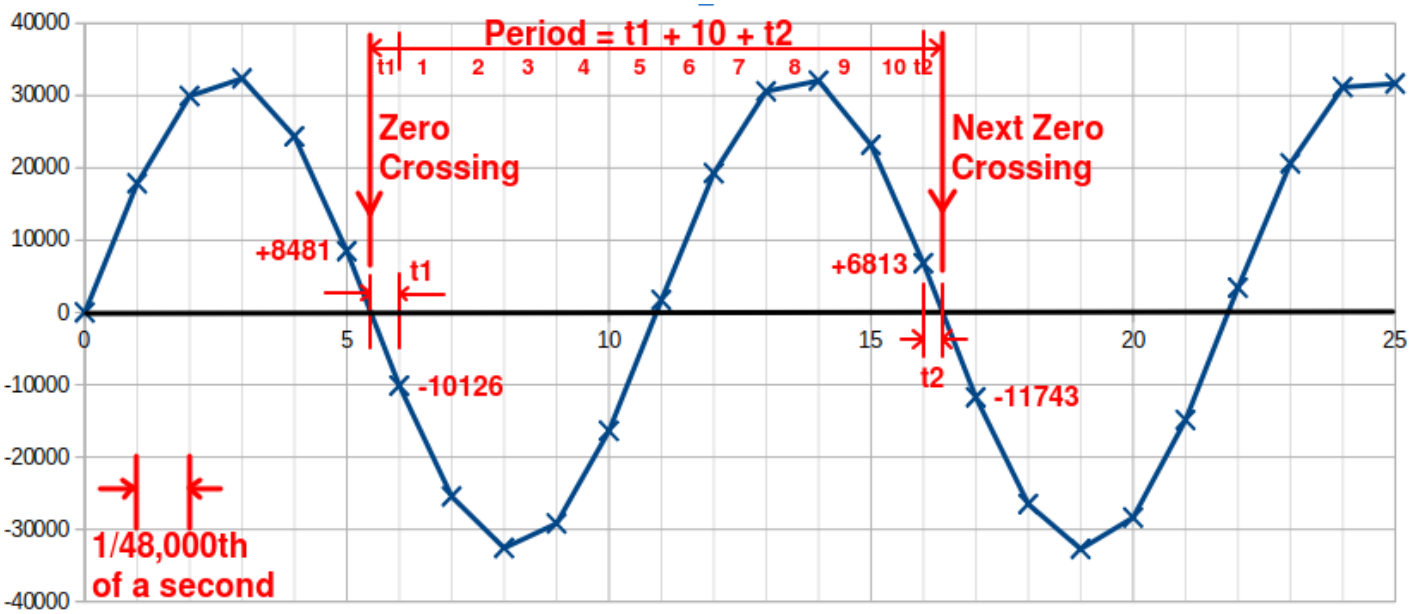
The simple approach is to detect the zero crossings of the sinewave, and carefully time the period between zero crossings. Frequency is the reciprocal of Period, so once you have the period, a moment later you know the frequency too, and the problem is solved. It turned out to be very easy to implement this method, and the accuracy is rather impressive. Experimentally, a SINGLE CYCLE measurement of the frequency at 1500Hz audio frequency leads to frequency accuracy of around +/- 0.05 Hz. More than enough!

NOTE: 1500Hz is an unfair frequency to measure at, because each cycle fits into an exact number of audio samples; 32 audio samples at 48,000 cycles per second make up one 1500Hz audio cycle. This sets us up for an extremely low error cycle measurement. So to make things fair, I actually offset the frequency for the test, to around 1520 Hz, so that the cycle doesn't contain an exact number of samples.

In actual fact we don't really need to update the frequency of the Si5351A Synth 1500 times per second… that's excessive. QDX has configuration options that allow you to specify an averaging period, where a specified number of sound samples AND a specified number of audio cycles must both be exceeded, in order to terminate the audio cycle measurement. The default parameters are set for a minimum of one cycle, but a minimum of 480 samples. 480 samples is 0.01 seconds and since we never go as low as 100Hz, the dominant criteria is therefore a measurement of at least 0.01 seconds.

With this default configuration, we get 100 audio frequency measurements per second, and experimentally the frequency accuracy is better than +/- 0.002 Hz. Amazing.

As for the details of the analysis, study the following hypothetical sinewave:



Remember that we have a built-in 24-bit 48ksps stereo USB sound card. Accordingly we get 48,000 samples per second. In this example I am assuming WSJT-X is used; WSJT-X outputs 16-bit samples, so the extent of the sinewave data is from -32,768 to + 32,767. The sound samples are received from the PC via the USB cable and are absolutely FREE of any noise. It's a lossless and zero noise transfer between the WSJT-X software on the PC, and the QDX's firmware. The

sequence of sample numbers received are EXACTLY what the WSJT-X software generates in its audio software-DDS.

We also know that the sound samples arrive at exact equal intervals of one 48-thousandth of a second. This is effectively the time base of our period measurement. It is not necessary to run any kind of actual timer inside the microcontroller, because in this instance we already have the luxury of knowing we get exactly 48,000 samples per second from the PC.

In the graph, the sine wave is shown starting at sample 0, with amplitude zero. In general, except in special cases (such as 1500Hz as mentioned above), we won't have a subsequent zero crossing indicated by a sample with value zero. We will have a sample ABOVE zero, followed by a subsequent sample BELOW zero. It is easy to detect this situation in the firmware code, just by looking for two successive samples, the first one being positive and the second one negative.

In QDX the firmware detects positive-to-negative zero crossings but it could equally well be done on negative-to-positive zero crossings, it is of no importance. It would even be possible to measure the frequency of half-cycles, with half the accuracy. But this kind of extremism doesn't seem warranted, unless in future some kind of mode with very fast measurements is needed or perhaps, we need to quickly measure the frequency of very low frequency audio signals.

Once we have a positive value followed by a negative value, and we know that they are 1/48000th of a second apart, we now want to find the exact time of the zero-crossing itself. This is also quite easy with a straight-line (linear) approximation. For small angles (in radians) the well known approximation holds: x = sin(x).

The period of the measurement is then given by the fractional times calculated by interpolation, at the start and end of the sample, plus the number of complete sound samples that occur BETWEEN the zero crossings. This is a "number of samples", and we know that there are 48,000 samples per second, so now it is easy to calculate the time in actual seconds which is the Period of the audio cycle; then the frequency is 1/Period, and we're done.

An example

To give a worked example, consider again the above simulated sinewave samples (generated in a spreadsheet). For ease of viewing, there are not many samples per cycle, in other words, this is a much higher frequency than we will be dealing with in practice; but it serves to make the principles relatively easy to understand.

We see audio sample #5, which happens to have a value of +8481 (according to the spreadsheet). Now the next sample, audio sample #6, has a value of -10126. We want to know the fractional interval (in samples), that is at the start of this sinewave cycle. I have called this time t1 in the diagram. Time t1 (as counted in samples), is a fraction 10126 / (8481 + 10126). This is a straightforward linear interpolation between the values at sample #5 and sample #6. The result calculates to be 0.544203794 samples.

The next zero crossing occurs between sample #16 (value +6813) and sample #17 (value -11743). Now we want to know the time t2, which is the fraction of a sample that occurs between the time of sample #16 and the zero crossing. I hope that you can see that this fraction is given by, in this case, 6813 / (6813 + 11743) = 0.367158870.

Next we can see that there are 10 complete sample periods in the cycle, between our two fractional pieces t1 and t2. The complete period, measured in audio samples, is therefore t1 + 10 + t2, which is 0.544203794 + 10 + 0.367158870 = 10.911362664.

By now, we have determined that our audio cycle being measured, has a period of 10.911362664 audio samples. We also know that there are 48,000 samples per second. Therefore the Period of the cycle in seconds, is 10.911362664 / 48000 = 0.00022732.

The frequency of the audio tone is therefore 1 / 0.00022732 = 4399.08 Hz.

All without anything too complex, no complicated DSP, Fast Fourier Transform, etc. Love it!

Error considerations

Now like any good scientist, we should consider the possible sources of error in this measurement technique and try to understand if it is good enough for our purposes.

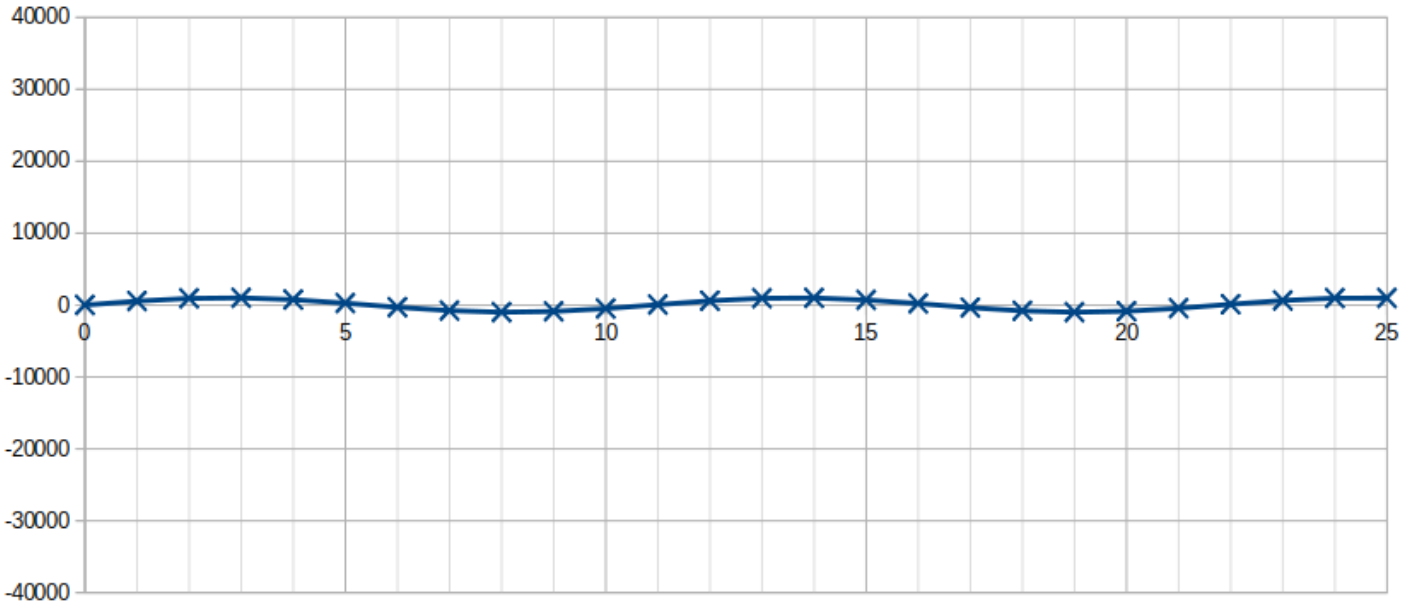
Thinking about it, I could envisage three possible sources of error: audio amplitude error, timing error, and interpolation error (the straight line linear interpolation). Let's consider each of these in turn.

1) Audio amplitude error:

WSJT-X produces samples with 16-bit resolution. Other digimodes software may offer 24-bit resolution but I have not come across any popular modes or software that do this. Neither do I think there are any that provide BELOW 16-bit resolution. Since most users will use WJST-X, and other software anyway has similar facilities, I will use WSJT-X for the purposes of discussion.

A 16-bit resolution provides 65,536 amplitude values per sinewave cycle. That's a theoretical (and practically achieved, since we have the built-in USB soundcard and therefore a lossless, noiseless transfer mechanism) 96dB dynamic range in the information transfer between WSJT-X and QDX, a very accurate transfer indeed.

And yet – there are two scenarios where this can be dramatically reduced.



The first scenario is where the operator uses the output amplitude slider at the bottom right of the WSJT-X screen, to something less than maximum. On a more traditional digital modes station involving a SSB transceiver, this would be done in order to avoid over-driving the SSB transceiver's audio input, causing splatter. There is NO POINT to doing this on the QDX, since the QDX RF output simply cannot be overdriven. The QDX RF output simply **is**, it is the perfection of a

RF signal without any spurious residual carrier or attenuated unwanted sideband, without any audio harmonics due to over-driving and clipping.

But perhaps the operator wants to reduce the transmission output power, and uses the amplitude slider to do this? Again there is NO POINT, because QDX does not transmit at reduced output power. It's all or nothing. There is no in-between.

Therefore the operator should be advised for best accuracy, to leave the volume slider at the maximum value, as there is no point to any lower value choice anyway. I found I was able to move the slider to very low volumes anyway before the frequency measurement accuracy significantly deteriorated. So this is not a particularly serious concern. But the recommendation still remains, just leave the slider at maximum.

The second scenario where lower resolution sinewaves can be generated by WSJT-X, is during the key-down and key-up instants, where WSJT-X applies (I believe), a raised cosine amplitude envelope to the beginning and end of the transmission. For the sake of design simplicity and because it is much less critical in digital FSK modes, than on/off keyed modes such as CW, amplitude envelope shaping has been omitted from the QDX design.

So there is no point to WSJT-X sending reduced amplitude sine waves at the beginning and end of the transmission key-down, since QDX cannot implement that anyway. Of course, WSJT-X does not know this, and there is no way to turn it off, so it will happen anyway. We can't "advise" WSJT-X software in the same way as we could advise the operator to leave the slider at maximum setting, in the previous discussed scenario.

At very low amplitudes, there would be a much more significant quantization error due to the limited amplitude resolution, that has the potential to increase the frequency estimation error substantially at very low amplitudes such as when the raised cosine envelope shape starts. This could cause a frequency estimation with a significant amount of error, and the Si5351A Synthesizer to start producing RF at an incorrect frequency.

To avoid this possibility completely, the QDX has a configuration ("Rise threshold") which allows it to ignore any audio cycles until a certain percentage of full amplitude has been reached. This can be set to a high percentage value such as 80 or 90%. This will ensure that there is no initial frequency value with a high error. Transmission will not occur until the 80% threshold (for example) is reached.

As a side-benefit, this will also FORCE the operator to keep the volume slider at the maximum value, or there will be no RF output! The "advice" will become binding.

The threshold should not be set to a very high value such as 99% because at high frequencies, where the cycle is made up of a smaller number of sample values, there may be no sample value near the actual peak of the sinewave, and so it may take several cycles for perchance, a value to exceed the threshold.

There is a similar "Fall threshold" setting to switch off the transmission at a defined amplitude threshold, before the raised cosine trailing edge of the amplitude envelope generated by WSJT-X takes the amplitude to a very small number and causes a potentially inaccurate frequency measurement.

"Rise threshold" has a default value of 80% and "Fall threshold" has a default value of 60%; it is unlikely the operator would ever want or need to change these from the default values. The default values are not critical but they should be neither very small, nor very large; and furthermore the falling edge threshold should be somewhat smaller than the rising edge threshold, otherwise key-up will be effected prematurely.

When the operator uses the maximum volume setting, and thresholds are implemented to avoid the possibility of high calculation error at very low amplitudes which could occur during the raised cosine leading and trailing edges, the effects of amplitude error are practically speaking, of no consequence.

2) Timing error:

USB audio devices transfer data to and from their PC host at an agreed rate. In the case of QDX, it appears to the PC as a 48ksps (48,000 samples per second) 24-bit stereo USB sound card. However, in common with all such USB audio systems, there is no facility for synchronization between the PC USB host, and the USB device. In other words, there is no 48kHz clock signal between the two. In the Full Speed USB protocol, the host and device communicate via data packets sent 1000 times every second.

Now it can get all rather complicated… because the host may send data to the device at not exactly 48kHz, and the device may send data to the host at not exactly 48kHz either. There are specifications in the USB standards that define the worst allowable error percentage. Now both ends of the connection have to try to deal with the fact that there is ALWAYS going to be a difference between them. Even if both 48kHz crystals derive their sample rates from a quartz crystal reference, the crystals will not be at exactly the same frequency.

In the case of QDX, there is a 25MHz TCXO that operates as the CPU clock input. All signals are derived from this clock, including the 72.something CPU clock, and the 48MHz USB peripheral clock (well, CLOSE enough to 48MHz), and the I2S peripheral master clock which is at 512 times the sample rate (each sample has 512 clocks); or 48kHz x 512 = 24.576 MHz.

This 24.576 MHz signal is derived internally in the QDX, by a PLL multiplication factor of 117, then division factors of 17 and 7. So 25 * 117 / (7 * 17) = 24.579832 MHz. Dividing by 512 gives an audio sample rate of 48007.5 samples per second! As well as this synthesis accuracy limitation there's also the smaller imprecision in the 25MHz TCXO oscillation value.

Luckily, this small 7.5 samples per second difference is well within the allowable percentage error in the USB sound device specification. On the PC side, the built-in USB driver has to deal with this difference, in addition to the fact that its own master clock will also not be truly at precisely 48kHz; it does so, I believe, not by digitally re-sampling to what it thinks is exactly 48ksps, but by some kind of software phase lock mechanism.

On the incoming data from the PC over the USB interface, we're also not going to be getting data samples at exactly what WE believe is 48ksps (our belief also being slightly inaccurate, itself). So what do we do… certainly I'm not going to dabble in software PLL clock recovery; an option could be do occasionally drop a sample if there are too many, or to duplicate a sample from time to time when there are, in my judgement, too few.

But in the end, after all this waffle, none of it is necessary. It is perfectly acceptable to do NOTHING about the lack of synchronization, just nothing at all! It is fine to pretend that we get exactly 48,000 samples per second, and analyze them all perfectly as though everything was exact. If there is any sample rate error, it will simply produce in the calculation result, a proportionate error in calculated frequency. Even a 1Hz error, if it happened (and frankly, I have never seen even a 0.01Hz error), would be of no consequence to operation of the transceiver, alongside much more significant errors such as the precision of the RF synthesis.

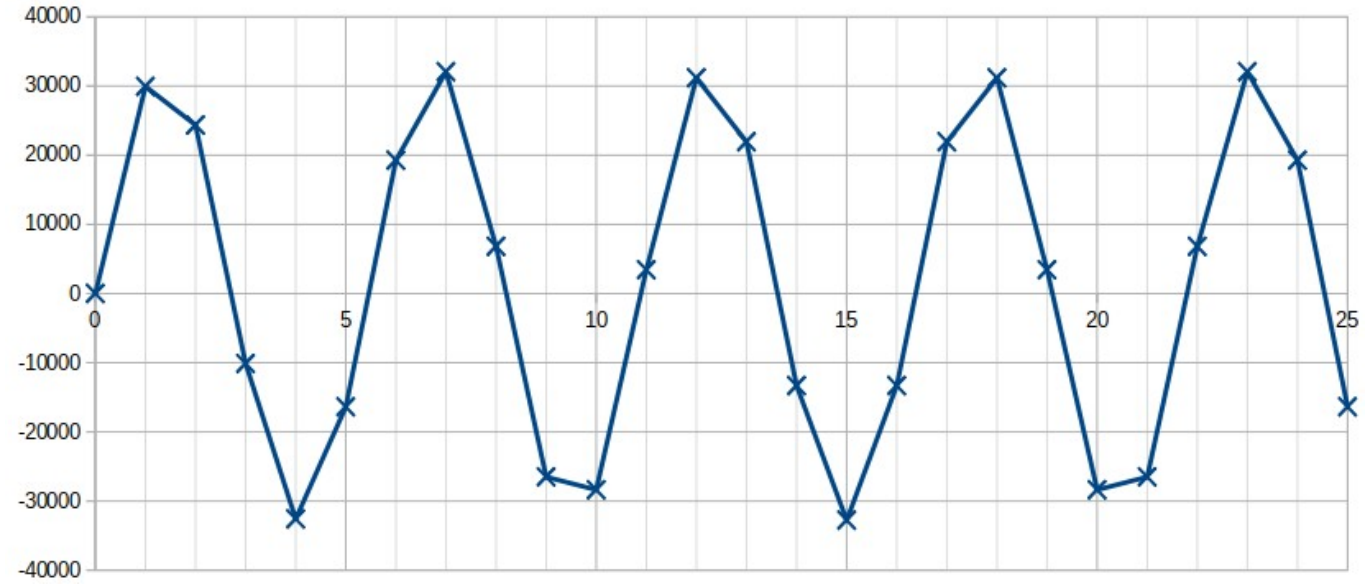
In the end therefore, the conclusion is that timing errors are of no consequence.

3) Interpolation error

Remember the approximation we have used for the zero-crossing interpolation that determines the exact moment of the zero-crossing event? Mathematically, the approximation is that for small values of x, then sin(x) equals x. Put another, less mathematical way, we can draw a straight line between the two points we want to interpolate, which makes the calculation easy.

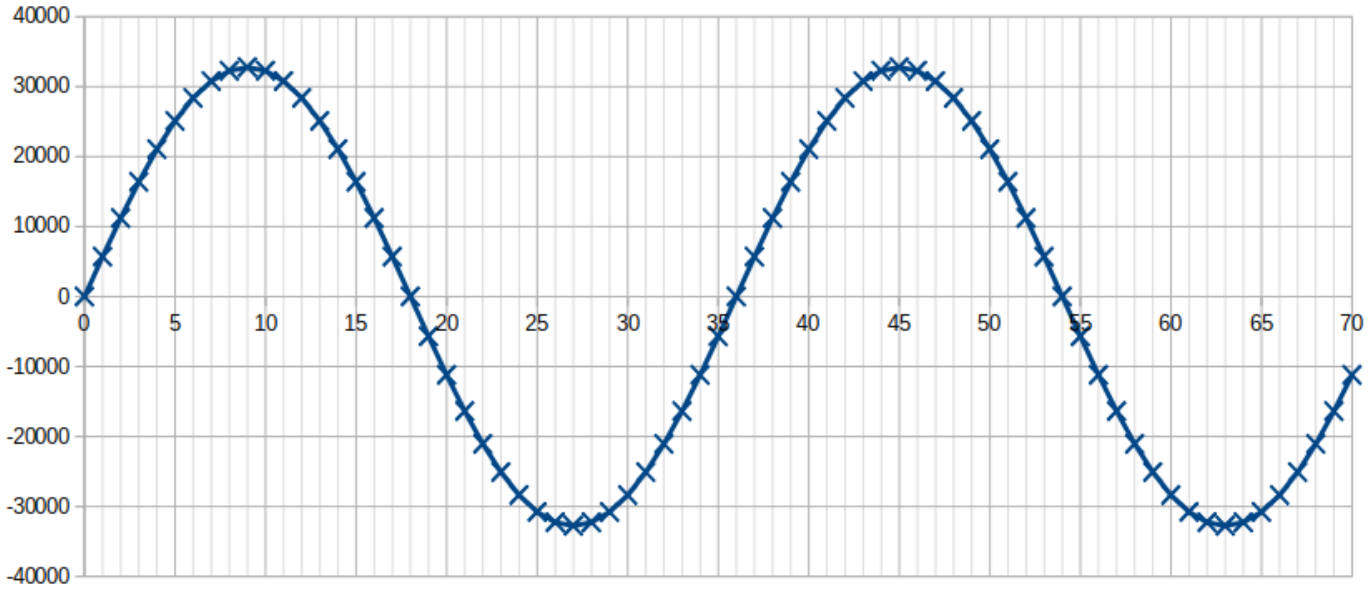
What can cause this to generate error? Well, when we are no longer dealing with “small values of x”. Practically speaking in our case, this occurs when there are a lower number of samples per audio cycle.

Consider an extreme example, for the sake of demonstration – here is a an 8.4kHz sinewave (much higher than we will ever be dealing with):



You can see that the sample points, indicated by the X’s, are quite far apart. The zero-crossing time, or the time between the two samples adjacent to the zero-crossing, is a relatively large fraction of the cycle period. Our “x” is therefore not so small as it was before. The straight lines that previously didn’t look at all bad, now look a lot less like the sinewaves that are being approximated.

Now consider this sinewave which happens to be about 1.27 kHz:



Obviously it LOOKS a lot better but we can also see that using straight lines at the zero crossing, will be a very much better approximation to the sinewave, than the high frequency case.

We can therefore say that when the sample rate is fixed at 48ksps, a low frequency audio sinewave is a lot more accurately interpolated, than a higher frequency audio sinewave. In other words, the errors get bigger, and quickly so, as we increase the audio frequency. This can also easily be seen experimentally by generating a signal in WSJT-X and then observing the measured frequencies (this can be done using QDX's own built-in tools, as described elsewhere in this manual).

The error in frequency measurement is also reduced by measuring over more than one cycle. Since we are adding a fraction of a sample at the beginning, and a fractional of a sample at the end of each cycle, and counting the number of samples in between – any error on the fractional estimations (due to linear interpolation error) will be reduced by counting multiple samples. For example, if we measure for 2 cycles, the error will be halved. If we measure for 10, the error will be reduced by a factor of 10.

As mentioned earlier, it is quite unnecessary to count the frequency so often as a SINGLE cycle. If we are measuring say, an audio signal of around 3kHz – do we really need to measure it 3000 times per second? Not practically speaking, no.

So the QDX has a configuration built in, which specifies a number of cycles that must be measured, and a number of samples that must be measured. BOTH of these parameters must be satisfied, before the measurement activity completes. The default values are 1 cycle, and 480 samples. 480 samples is a time duration of 0.01 seconds. In this configuration, a very low frequency signal, below 100Hz (0.01 seconds period) would be measured properly. And at any higher frequencies than 100Hz, the minimum samples criteria will dominate, and ensure that the averaging period is automatically increased for higher frequencies! For example at 3kHz, 480 samples will contain 15 cycles, so the error in the fractional period estimation due to interpolation inaccuracy, will be reduced by a factor of 15. In this way, high frequencies do more averaging than low frequencies; we get 100 frequency measurements per second.

Again these configuration parameters are not particularly critical, and again, the default values will suffice for all imaginable purposes and most likely the QDX operator will never want or need to change these, other than perhaps for his own curiosity of investigation.

Sinewave interpolation

As an aside – for interest only – I did investigate the possibility of more advanced interpolation methods than linear (straight line). I spent some enjoyable time experimenting with a method which measures the amplitude of the sinewave (looking at the incoming samples) and then fits the two samples either side of the zero crossing to an actual sinewave rather than a straight line.

It did provide some improvement, some observable reduction in error. However, the error reduction was not as great as I had expected, and I suspect this is due to limitations in the floating point accuracy of the arc-sin implementation, perhaps (speculation) a limitation to the number of terms of a polynomial expansion? Furthermore, although the STM32 processor used has a floating point unit, the calculations were still too slow at the 72MHz CPU clock speed used. In experimental measurements the calculations were too slow to be able to perform single cycle frequency estimation at the upper end of the frequency range, which is exactly where we most need the improved accuracy; multiple cycles were therefore necessary and it was found that the improvement in accuracy of averaging simple straight line interpolation over several cycles, was more than the improvement in accuracy of the slow sinewave interpolation!

Furthermore other inaccuracies could creep in, such as the estimation of the maximum amplitude; or we'd have to assume the maximum amplitude sinewave, which would work less well on the rising and falling keying edges.

I toyed briefly (very) with the idea of trying to implement a three (or more) point sinewave fit, but dismissed it as it would involve even more calculation and be even slower. Polynomial fits were another possibility considered.

But frankly – in the end – the simple linear interpolation is very reliable at the frequency range of interest, and is already SO good, that the idea of trying to pursue more advanced approaches, seemed, particularly after a day's worth of investigation, to constitute an unnecessarily frivolous indulgence in mathematical decadence.

Errors conclusion

The frequency estimation by cycle timing method employed in QDX is exceptionally accurate, far more precise than needed for any digi modes likely to employed on HF bands during QDX operation (I cannot imagine any exceptions). There are a few boundary cases where accuracy could conceivably be impaired, and though this is actually still unlikely to be of any practical significance, configuration parameters are provided to mitigate and control the errors, with suitable default values that will be excellent for all conceivable use cases.

As a final word on audio frequency estimation, I want to also mention that WSJT-X is said to transition evenly between two frequency symbol tones, by implementing a raised cosine falling envelope shaping on the finishing tone, at the same time as a raised cosine envelope shape on the newly starting tone. As discussed previously, QDX just doesn't do envelope shaping. However, I believe that to all intents and purposes, that this fancy raised cosine tone envelope shaping is equivalent to a gentle "frequency slide" between the old and new tone frequencies. Perhaps it is even exactly mathematically equivalent, I don't know.

At any rate – for any mode WSJT-X generates, QDX will continuously measure the frequency and update the transmitter output frequency accordingly; at 100 updates per second (the default configuration) this is plenty sufficient to achieve smooth transitions between tones; additionally the frequency synthesis method used in the Si5351A itself causes a "slide" between frequencies not a sudden discontinuous transition, because the PLL takes a finite time to settle on the new value.

QDX contains its own test and analysis tools built-in; elsewhere in this manual is a description of how you can observe these frequency transmissions yourself, using the built-in tools.

Si5351A fine control

Historically QRP Labs products have set up the Si5351A with an integer frequency, except for the Ultimate3S QRSS/WSPR transmitter which has an algorithm during WSPR operation to optimize the frequency steps for very accurate 1.46 Hz tone spacing.

The problem with QDX is that the QDX must faithfully transmit any frequency given to it. The tone spacing may be very fine, or it may be coarse. QDX does not know. QDX knows nothing of the transmission mode. All QDX does is receive audio tones from a PC and effectively modulate them to RF. The tones may be 1.46 Hz spaced in the case of WSPR, or 6.25 Hz in the case of FT8, or many other possibilities. Therefore a fixed algorithm for WSPR such as Ultimate3S has, or an integer frequency configuration such as used in the QCX CW transceiver, will not be optimum here.

As an example, suppose we want an output frequency of 14,097,002.123 Hz and the firmware has to decide the parameters to configure the Si5351A chip. So let's say we chose a MS Synth divider of 62 (abiding obediently by the even integer MS Synth divider rule). Now that means the internal VCO must operate at 870,014,131.626 Hz. The multiplication factor from 25 MHz is therefore 870,014,131.626 / 25,000,000 = 34.960565265039996. Now that has to be represented as a + b / c where all are integers, b is in the range 0 to 1,048,575 and c is in the range 1 to 1,048,575 (b and c are 20-bit numbers – so the max value is 2 ^ 20 – 1).

The question therefore is the choice of b and c to get the fraction 0.960565265039996. Conventionally one may simply set c to 1,048,575 and set b to 0.960565265039996 x 1,048,575 = 1,007,224. Now we can calculate the output frequency:

25,000,000 x (34 + 1,007,224 / 1,048,575) / 62 = 14,097,001.845 Hz

The error in the output frequency is 14,097,002.123 – 14,097,001.845 = 0.228 Hz. It's rather larger than we might like; remember also that the error will be different every time and could be better or worse than this.

Well it turns out we took the cowardly way out by choosing the denominator (c) to be 2 ^ 20 – 1 (1,048,575) and that there are better choices which will bring the ratio b / c very much closer to the desired fraction 0.960565265039996. In fact the choice b = 416,333 and c = 433,425 works very well… look, now the output frequency is:

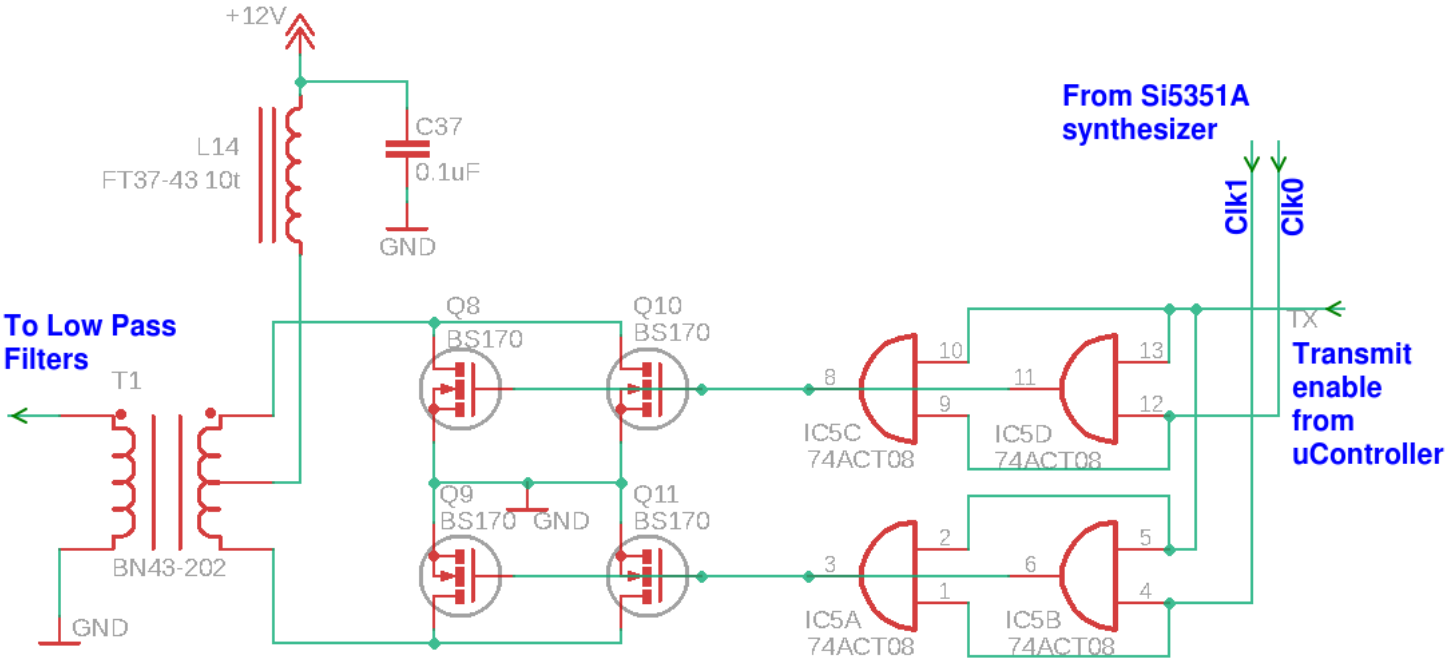
25,000,000 x (34 + 416,333 / 433,425) / 62 = 14,097,002.123 Hz

The error, to three decimal places, is now 0.000. Bingo!

However the question is STILL, how to choose b and c to get this magical and wonderful result. The answer is "best rational approximation" which delves into an area of mathematics called "continuing fractions". Rather than spoil (and lengthen) this manual with my own garbled interpretation, allow me to just refer you to Google and Wikipedia https://en.wikipedia.org/wiki/Continued_fraction .

The implementation in QDX is a simple and fast iterative algorithm based on the information found here, which rapidly converges on values for b and c which provide a very optimal ratio.

3.13 Class-D Push-pull Power amplifier



An interesting power amplifier circuit was designed specially for this transceiver. There are four BS170 transistors. Two are connected in parallel for the “push” side, and two connected in parallel for the “pull” side. Each side is driven by two AND gates of a 74ACT08 quad AND-gate IC, in parallel to ensure strong and fast drive.

The AND gate drivers are clocked from two outputs of the Si5351A Synthesizer which are arranged to have 180-degree phase shift. This provides the necessary phasing for the “push” and “pull” sides of the amplifier, without requiring an input transformer. All the 74AC08 AND gates are enabled by a “transmit” or in other words, a “key down” signal from the microcontroller; only when this signal is high, are the oscillator signals passed on to the MOSFET “gate” connections – this is the operation of the logical AND function.

The outputs of the “Push” and “Pull” sides are combined by transformer T1.

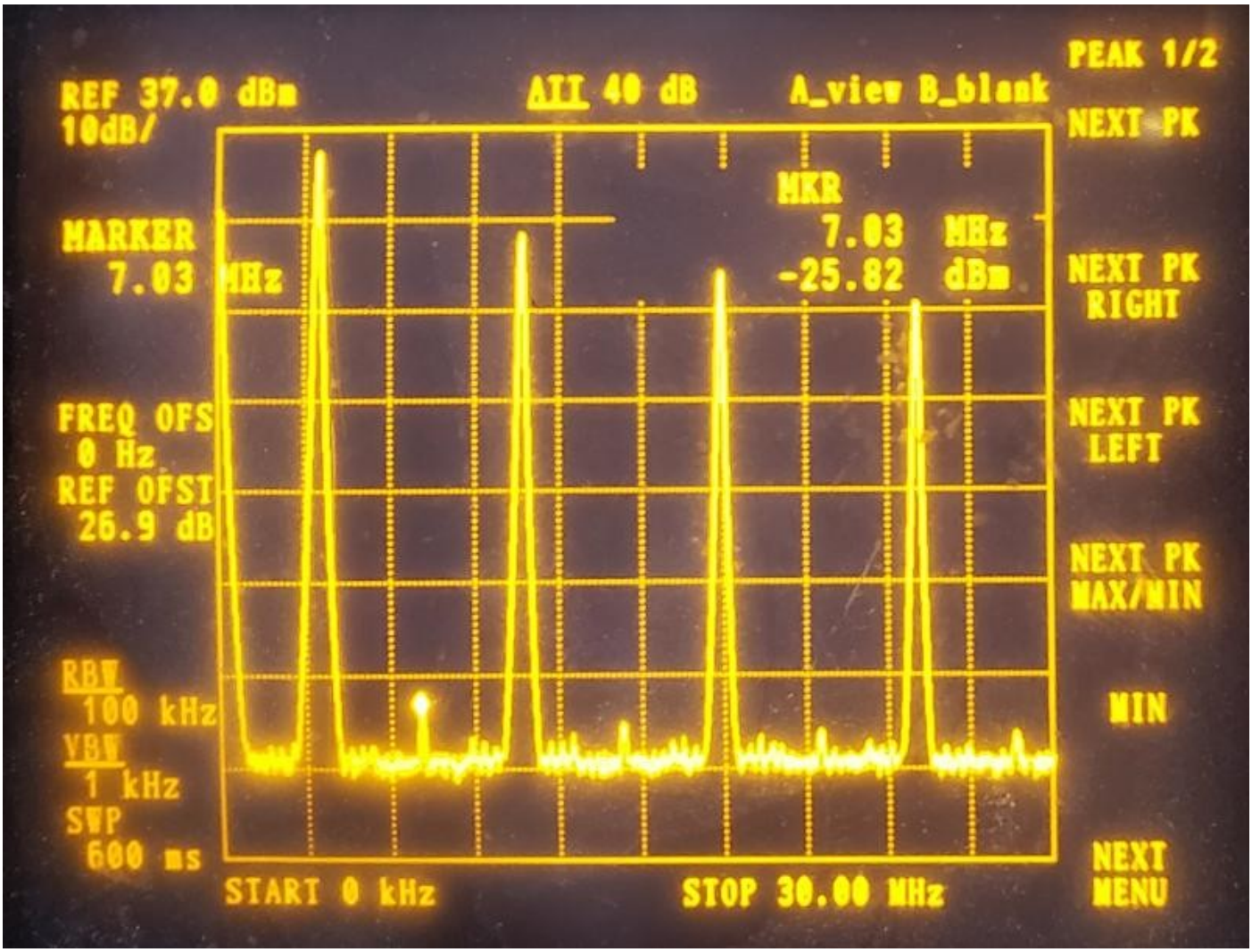
This power amplifier design provides a number of useful features:

- Class-D operation is efficient compared to Class C, or even worse, Class A/B linear which as discussed previously, is overkill for this application. Class-D is not as efficient as Class-E (as used in the QCX CW transceiver), but it better suited to a multi-band transceiver, since a Class-E amplifier has a resonant load circuit and this must be switched by band.
- Push-pull operation provides 5 W output power at a lower voltage than would be needed in a single-ended amplifier (9 V supply produces about 5 W output).
- Push-pull operation cancels even harmonics which are therefore at a very low level (due only to imperfect cancellation due to slight differences in transistor characteristics). Importantly the 2nd harmonic level is very low, which greatly simplifies the demands on the Low Pass Filters that will subsequently clean up the signal after the amplifier.
- Push-pull operation allows us to use four BS170 transistors to spread the load, but they are only connected in pairs so the driver only has to drive two paralleled gate capacitances not four.
- Using an antiphase drive (the precise 180-degree out of phase signals from the Si5351A) eliminates the need for an input transformer that is normally required in push-pull power amplifiers.

- All components are low cost and easy to replace in the unlikely event of failure.

As an example spectrum analyzer output, consider the following trace showing an 80m QDX. For this measurement, a 50-ohm dummy load was connected directly at the output of transformer T1, with no Low Pass Filtering.

The second harmonic is seen, about -70 dBc (70 dB below the fundamental) due to the cancellation effect of the Push-Pull amplifier. The subsequent even harmonics are even lower amplitude. Only the odd harmonics are problematic and require filtering.



3.14 Low Pass Filters

Every radio transmitter always requires low pass filters at its output to attenuate any out of band spurious emissions to sufficient level to comply with regulatory requirements (for example FCC in United States).

In QDX-M the Low Pass Filters are carefully optimized to provide excellent performance with low parts count. This can be achieved because the Class-D push-pull power amplifier design produces a very low level of even harmonics.

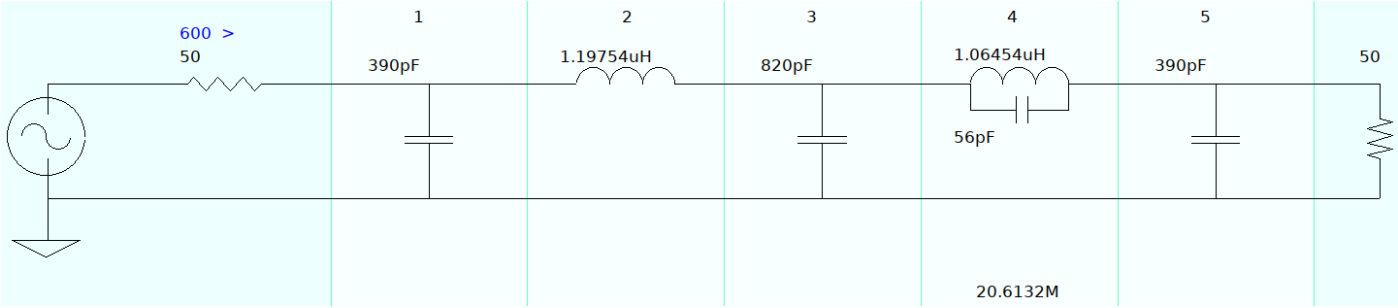
It is very important to reiterate that Low Pass Filter and Power Amplifier design must be considered carefully, at an overall SYSTEM LEVEL. In fact this applies to the entire transceiver design, but nowhere more importantly than the transmitter output stages, where the performance

is critical to your regulatory compliance. The requirements on Low Pass Filtering are determined by the level of spurious emissions produced by the final Power Amplifier as well as levels in earlier stages feeding into the power amplifier. One cannot simply transplant a circuit block from one transceiver design to another, and expect the performance to be satisfactory. System level design is critical.

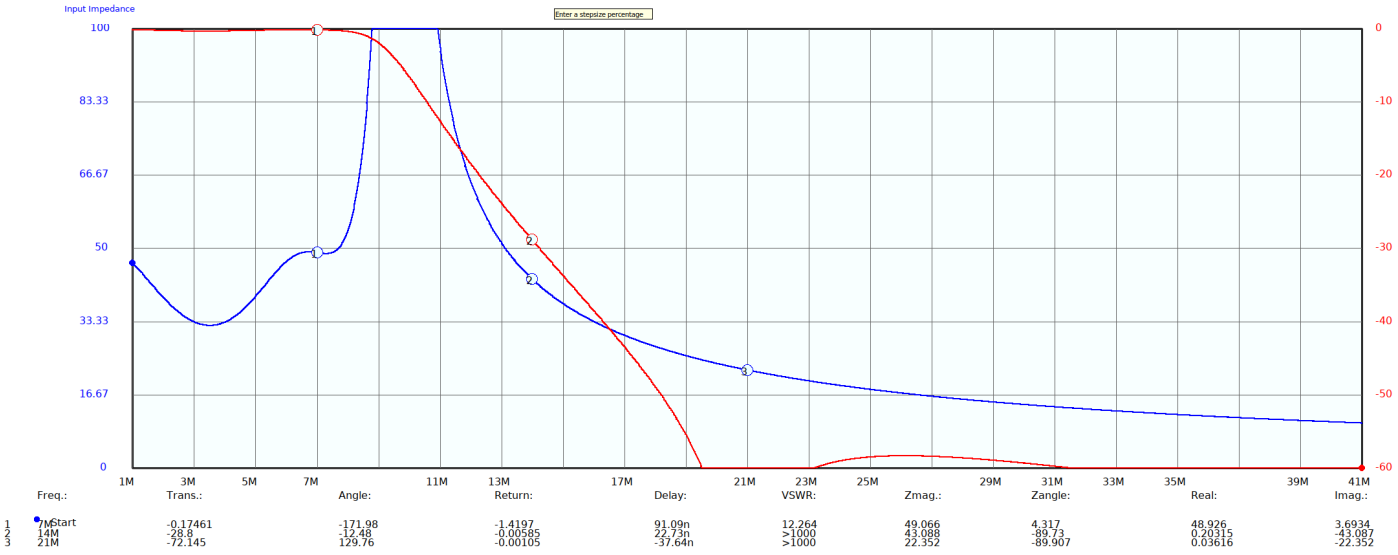
The Low Pass Filter design was done using the Elsie program by Tonne Software. It is freeware and can be downloaded from here: http://tonnesoftware.com/elsie.html

40m example:

For 40m this is the final schematic, having a resonant trap at the third harmonic:



The simulation shows:



Fundamental:	7 MHz	-0.17 dB
Second harmonic:	14 MHz	-28.8 dB
Third harmonic:	21 MHz	-72.1 dB

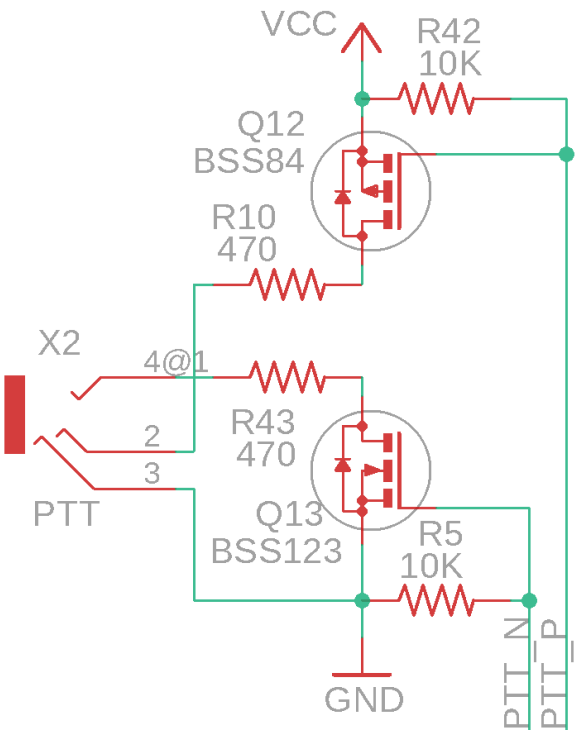
3.15 Configurable PTT (Push To Talk) output

The Rev 3 QDX PCB has a 3.5mm stereo jack as PTT output.

The “Tip” connection is pulled to ground during transmit, when Q13 is switched on by the microcontroller.

The “Ring” connection is pulled to +5V during transmit when Q12 is switched on by the microcontroller.

A common standard 3.5mm stereo audio cable with 3.5mm plugs at each end may be used without modification between the QDX and the QRP Labs 50W PA (which uses the positive-going +5V signal on the “ring” and makes no connection to “tip”.

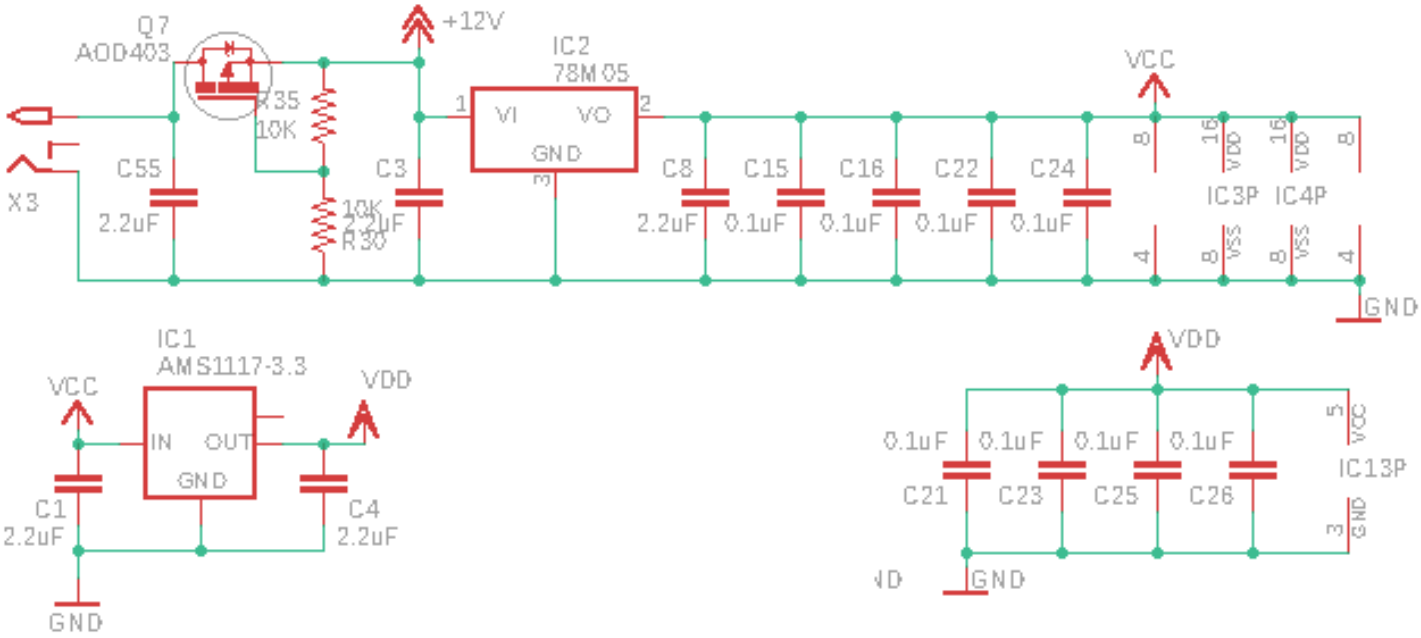


During transmit, the microcontroller activates ONE of the PTT signals; you may configure which one, according to your needs.

It should be noted that:

- The QRP Labs 50W PA may not be used in high duty cycle modes such as FT8, JS8 etc at full power. It should be de-rated to half power operation at maximum, by using a 12V or 13.8V supply. Caution is advised.
- The 50W PA kit has a built-in Low Pass Filter and is designed for signal band operation only.
- Firmware 1.04 or above is required to use the PTT feature.

3.16 Voltage regulation and supply decoupling



The two fixed voltage regulators in the QDX design produce +3.3V and +5.0V. The +5V voltage regulator type is 78M05 and the 3.3V type is AMS1117.

The 78M05 voltage regulator is robust, virtually indestructible. The AMS1117-3.3V regulator is a much more delicate IC, but in this application it is only regulating voltage down from +5V from the 78M05, so it is not being stressed and has a very low risk of failure. Note that this is a design change with respect to the Rev 1 QDX PCB where both voltage regulators were AMS1117, necessitating an additional 220uF capacitor at the power input; with the 78M05 used in Rev 2 and above, this 220uF capacitor is not needed.

Additionally a reverse polarity protection feature is provided by the P-Channel MOSFET type AOD403. This transistor may appear, at first sight, to be connected “in reverse”, since the drain is connected to the incoming DC voltage from the DC connector, and the source is connected to the downstream circuits. However, it is in fact correct and necessary.

In normal operation, at power-up a little current will flow through the forward-biased “body diode” of the AOD403, raising the voltage at the source pin (on the right hand side of the transistor as drawn in the diagram). The gate is connected to half the supply voltage via R30/R35, hence the gate voltage is 6V (say) below the source voltage, and the transistor switches on, as desired. 2.2uF capacitor C55 provides smoothing for any small spikes that may be induced in the supply wiring or from the power supply.

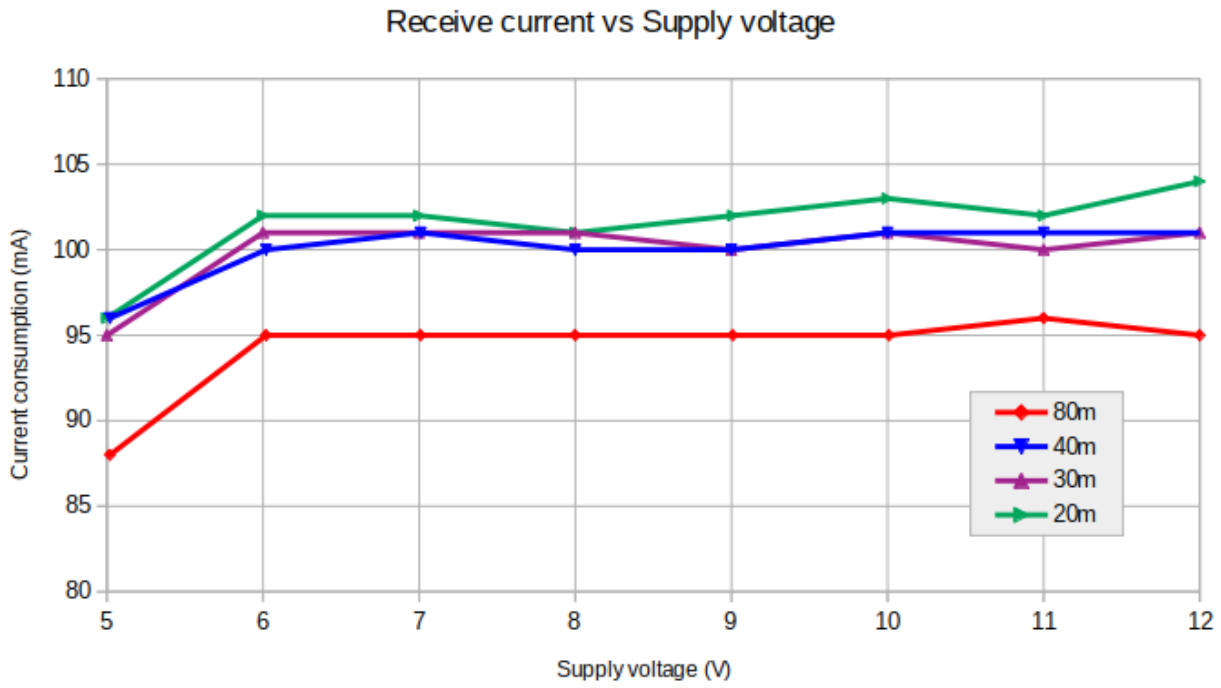
If the operator should happen to accidentally connect QDX in reverse, then the gate-source voltage will be positive, which will ensure the P-channel MOSFET is switched OFF and no current flows through it. Since the transistor was connected “backwards”, if the voltage is applied in reverse, no current flows through the “body diode” either. It therefore achieves the desired reverse polarity protection function.

4 Performance measurements

Several QDX units have been measured; there are of course variations from one to another, due to component tolerances, toroid winding style, etc. The results presented here are for a sample QDX; all measured QDX were reasonably similar.

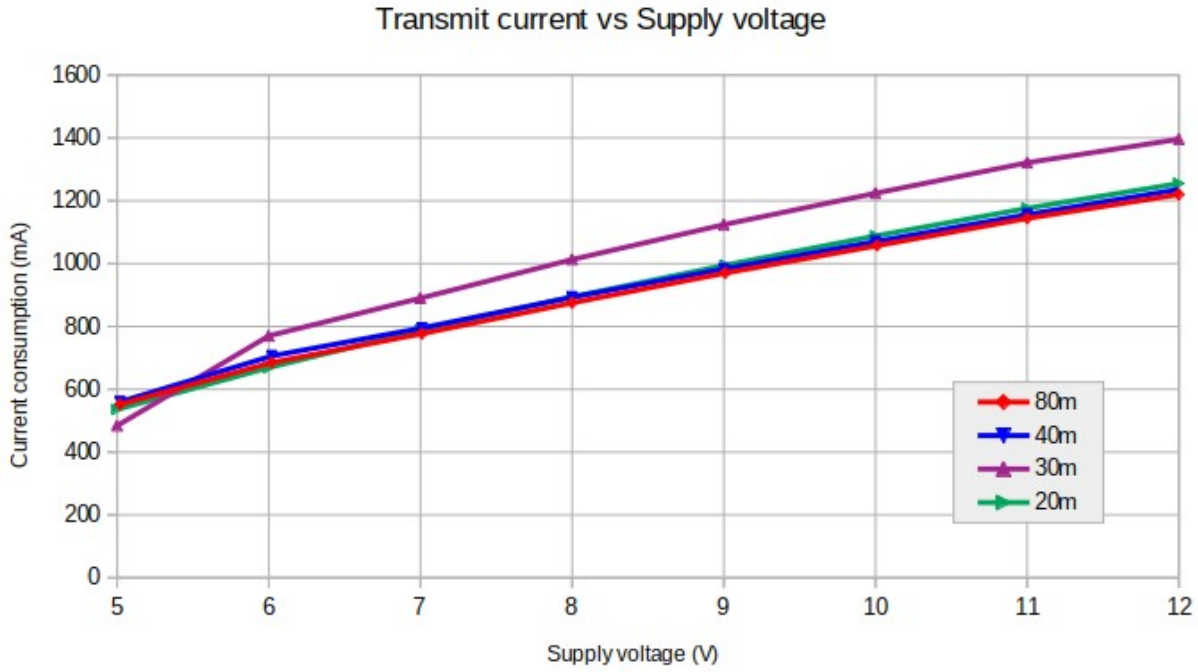
4.1 Receive current consumption

The following chart shows receive current vs supply voltage, per band. There is not much variation and it is fair to quote 100mA as the receive current. **For QDX-M, 150mA is normal.**



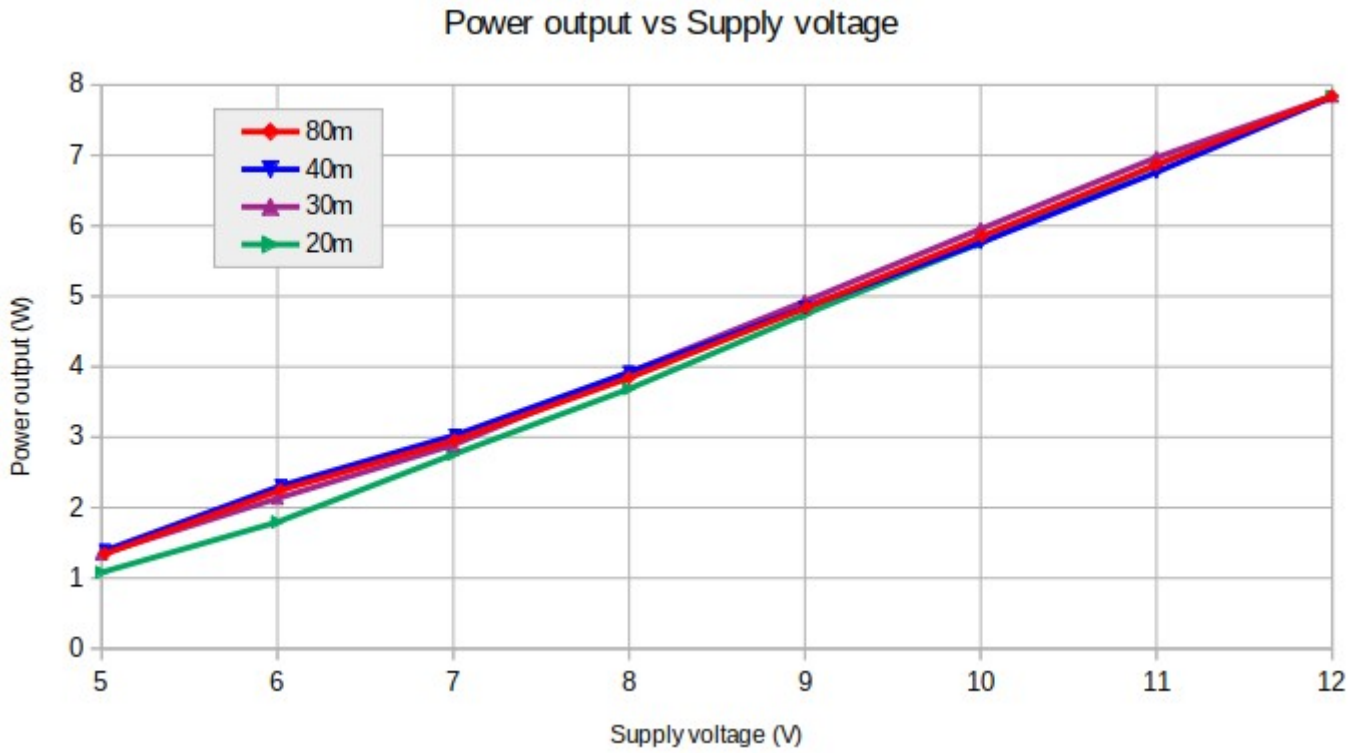
4.2 Transmit current consumption

The following chart shows transmit current vs supply voltage, per band.

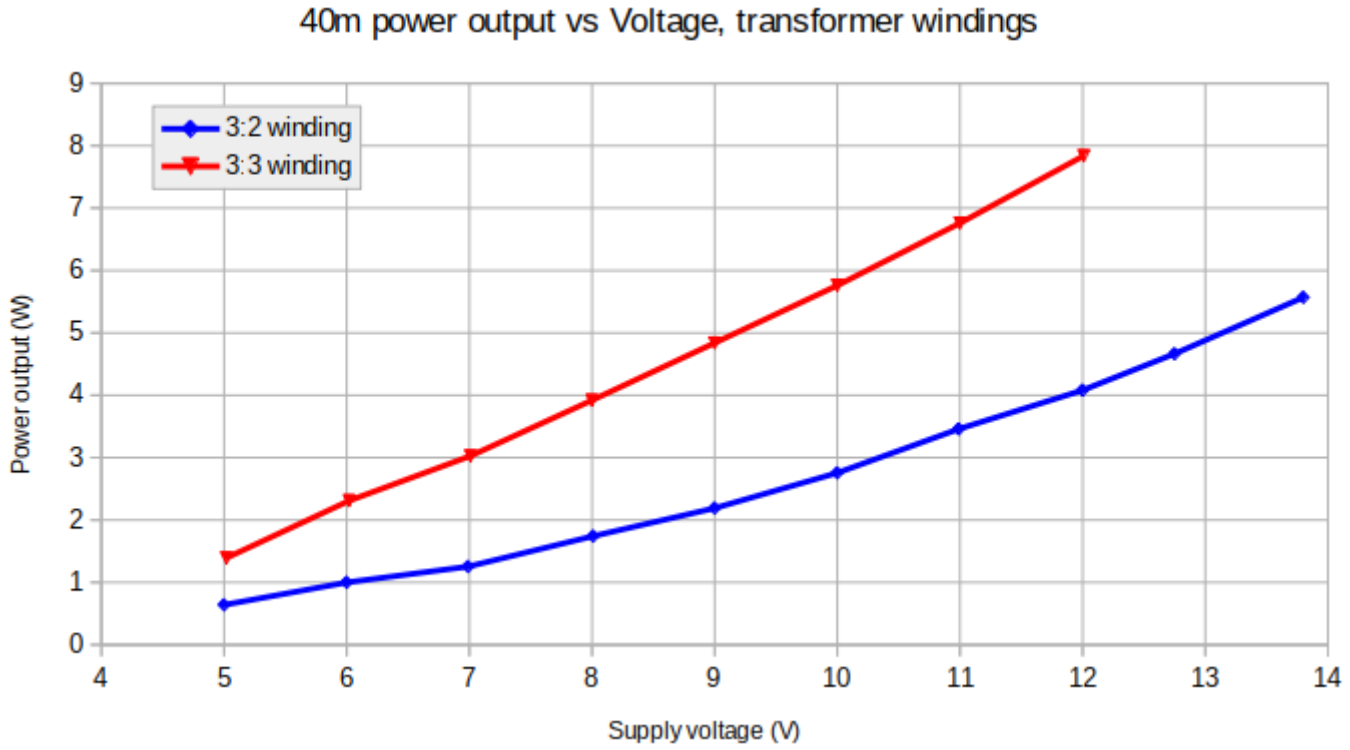


4.3 Power output vs supply voltage

Power output is consistent across bands. **Operation at more than 6W output is NOT RECOMMENDED and may destroy the power amplifier transistors.**



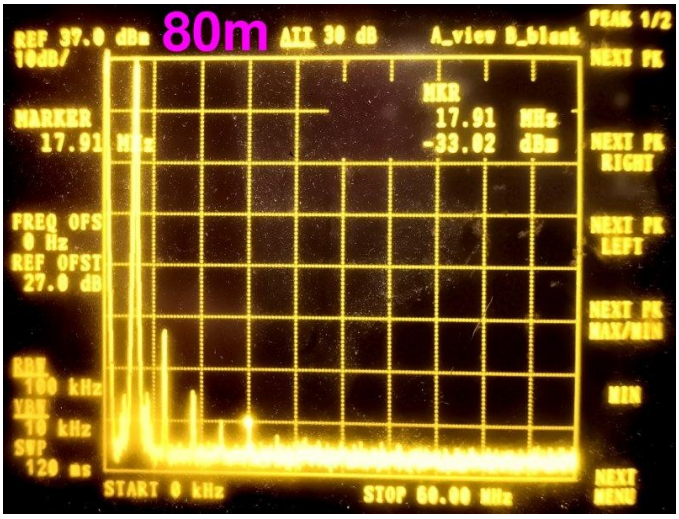
The following graph shows power output vs supply voltage for 40m operation, with the two alternative output transformer winding arrangements. 3:2 is less efficient but more suitable if you need to operate from a 12 – 13 V supply.

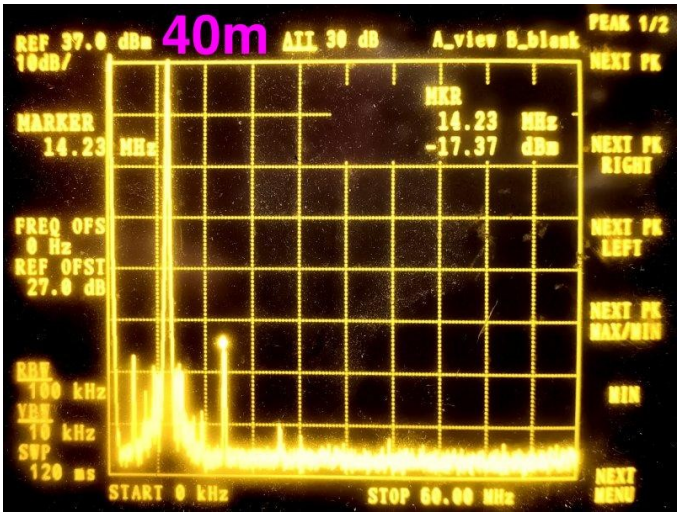


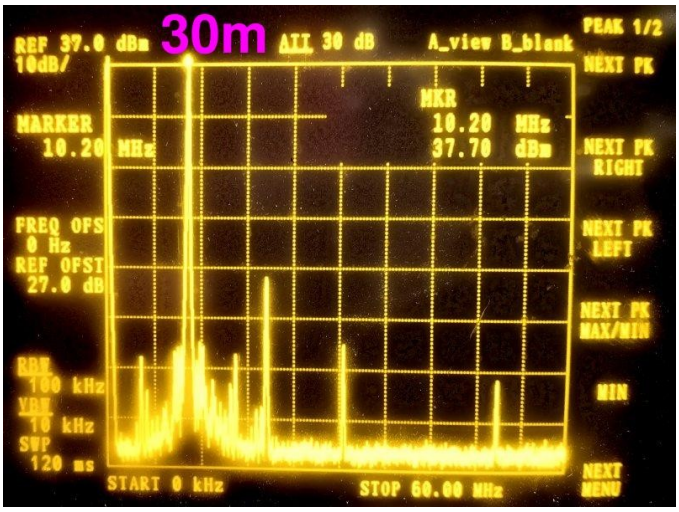
4.4 Output harmonic content

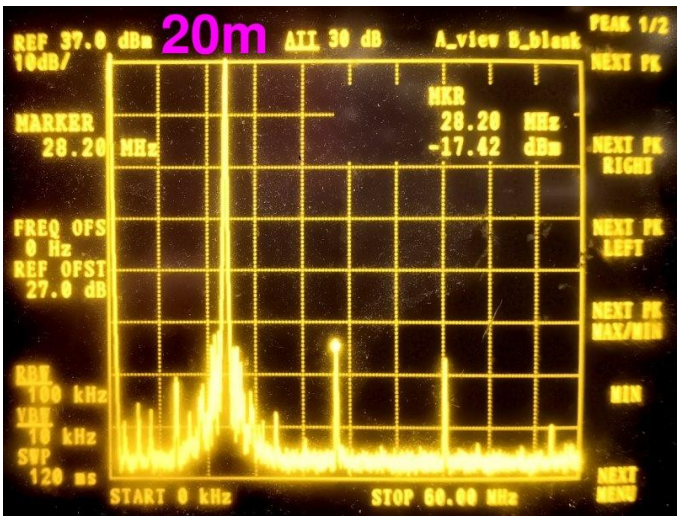
Equipment:
QDX > QRP Labs 50-ohm dummy load kit > attenuator > Advantest R3361C Spectrum Analyzer.

Band	80m	40m	30m	20m
2nd harmonic	-53 dBc	-55 dBc	-43 dBc	-55 dBc
3rd harmonic	-64 dBc	-71 dBc	-56 dBc	-57 dBc
4th harmonic	-70 dBc	Undetectable	Undetectable	-70 dBc
5th harmonic	-70 dBc	-73 dBc	-62 dBc	Undetectable



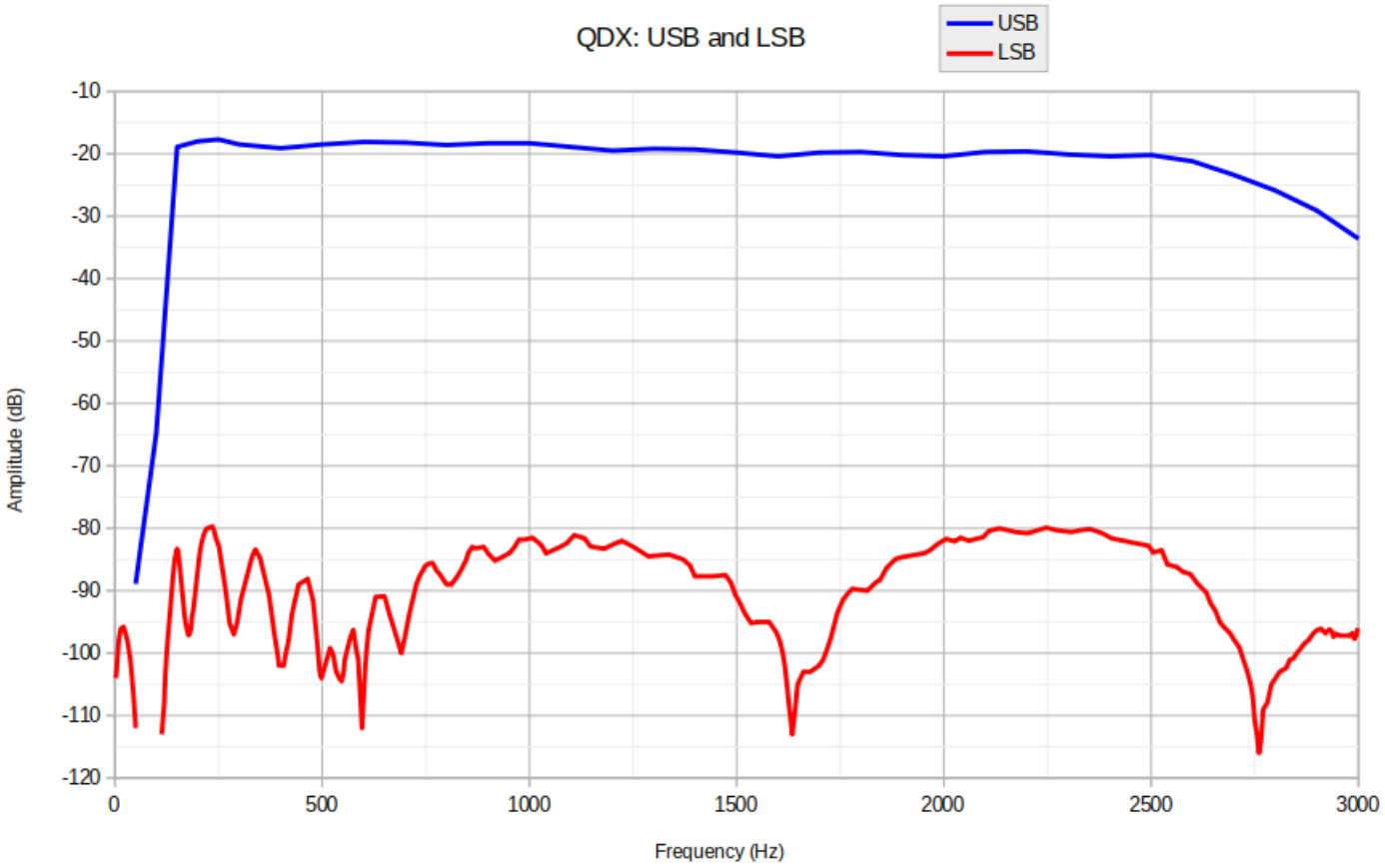




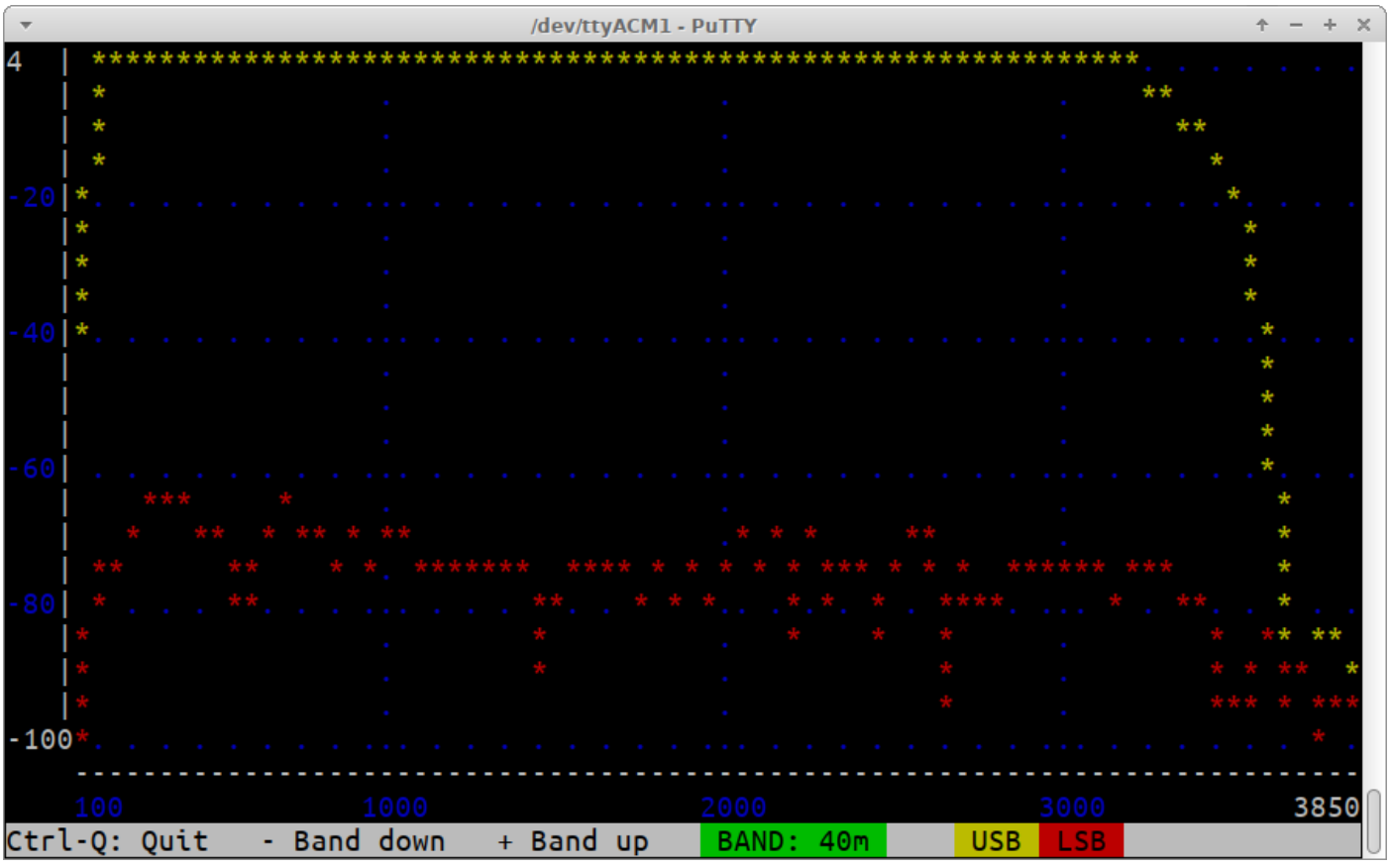


4.5 Unwanted sideband suppression

Measured using the QDX's internal signal generator; unwanted sideband is 60-70 dB down.



This is also seen on the QDX's own terminal application, Audio sweep tool:



5. Resources

- For updates and tips relating to this kit please visit the QRP Labs QDX-M kit page http://qrp-labs.com/qdx-m
- For any questions regarding the assembly and operation of this kit please join the QRP Labs group, see http://qrp-labs.com/group for details

6. Document Revision History

1.00	12-Jan-2023	First version version 1.00, derived from QDX manual version 1.17
1.01	17-Jan-2023	Fix typos on capacitor values
1.02	08-Feb-2023	Fix typo in section 2,9 (L2 should be L4)
1.03	12-Sep-2023	Moved output transformer to separate document
1.04	19-Dec-2024	Added notes that new kit batches are supplied only with 0.33mm wire which improves performance compared to 0.60mm.