Designing a modern digi-mode transceiver: QDX

Introduction

QDX (QRP Labs Digital Xcvr) is a very high performance, feature packed, 4-band QRP Digital modes transceiver that also boasts a very low cost.

Achieving the combination of features, performance and low cost isn't an accident, it is the result of design, innovation and careful production. This article documents some of the process. More details of the QDX design are in the manuals at http://qrp-labs.com/qdx

I will describe several unusual, innovative or interesting features. I really hope some of these will give you ideas to use in your own projects too. Look out for:

- Audio cycle frequency measurement and transmission
- Incorporating USB devices into the radio, for CAT control and noise-free audio
- An interesting and easy NPN/PNP transistor crystal oscillator with near rail-to-rail output
- Novel Class-D broadband push-pull power amplifier delivers 5W from 9V supply
- PIN diode switching of Low Pass Filters
- An embedded Software Defined Radio receiver is easier than you think
- Differential amplifier as differential ADC driver
- Easy firmware update with the radio pretending to be a USB Flash drive
- USB terminal applications for configuration, adjustment, debugging, fun and education
- Reverse polarity protection with low voltage drop using a single MOSFET

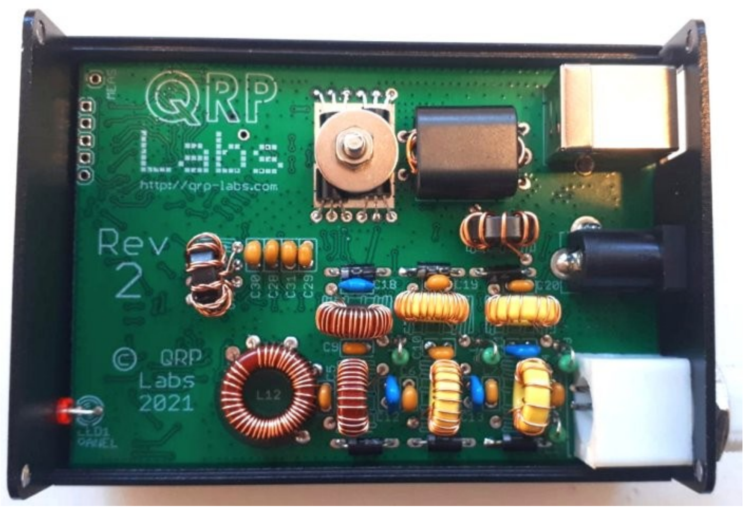
QDX

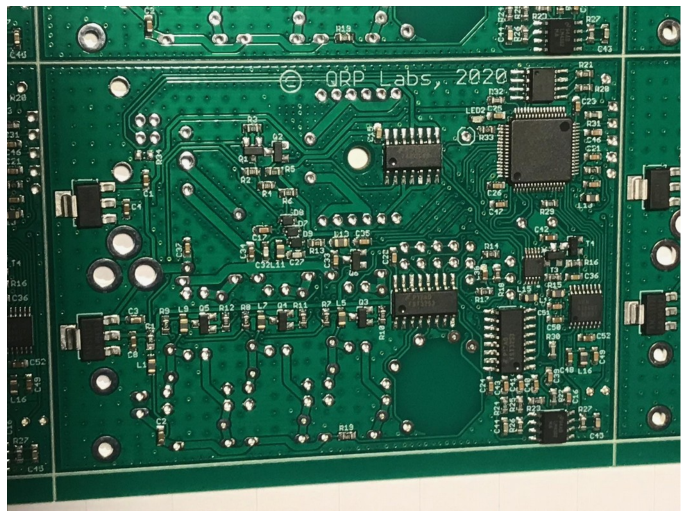
QDX is an innocent-looking little black box measuring just 3.5 x 2.5 x 1 inches.

On the front panel is a single solitary status LED, and on the rear you'll find just three connectors: RF (BNC connector for the antenna), DC power in, and a full-size USB B connector. The Rev 3 PCB also has a 3.5mm jack socket as a configurable PTT output to activate an external amplifier. Let's not talk more about such QRO matters in polite QRP company.

Aside from the RF connection and DC power, that single USB connector is the only thing that connects to the computer. That single USB cable carries both serial data to control the radio (CAT – Computer Aided Transceiver) and digital audio. This is key to the performance and simplicity of QDX, as it allows a complete interface with the computer over a single common cable.

Inside QDX is a single PCB with three low pass filters, filter switching, power amplifier components, band pass filtering and a trifilar receiver input transformer. Construction is easy! The bottom side of the board contains quite a few SMD (Surface Mount Devices) that are, thankfully, factory assembled.





Motivation for QDX

QDX was designed back at the end of 2019. Due to a lot of interruptions and distractions, not forgetting the Covid19 pandemic, it wasn't until 2 years later (October 2021) that it finally went on sale.

During 2019 there were several single band, fixed frequency digital mode kits available. They all used receivers based on a very simple direct conversion, double sideband architecture using my old enemy, the SA602 chip. Convenient and simple, yes – but good performance, not at all! Mostly these kits also had DSB transmitters, wasting half the power on the unwanted sideband and using twice as much bandwidth as really needed.

Frankly I was rather disappointed with what I saw. I felt that it should be perfectly possible to design something very much better, yet not costing more than these existing radio kits. It would take some careful design and quite a bit of innovation along the way. This was going to be great fun!

Digital modes

Arguably the oldest digital mode is Morse code. The carrier is keyed on or off. It's there, or it isn't. 1's and 0's. Or looked at another way, dits and dahs.

But modern digital modes normally involve a host computer running software to encode and decode keyboard entered messages into audio, that is then modulated onto RF by an SSB radio. You type on your keyboard, and the other operator sees the message appear on his screen (and vice versa). There are many types of modulation including frequency shift keying (FSK), phase shift keying (PSK), on/off keying (OOK), and more advanced techniques such as spread spectrum.

RTTY (radio teletype) is an ancient example of an FSK mode that can still be heard on the bands, particularly during RTTY contests. However, despite the many varied ways of transmitting information digitally, it's fair to say that many popular modern modes use multi frequency shift keying (MFSK). A sequence of tones (frequencies) encoded by a computer, modulated to RF.

By far the most popular digital mode today on HF is FT8, which is a good example of an MFSK mode. A message is encoded in very clever ways with lots of built-in error correction mathematics, making it highly resilient to interference. There are 8 tones that encode the message, spaced 6.25Hz apart. The total bandwidth is about 50Hz. The popular WSJT-X software package produces audio that is sent to the radio, and decodes received audio back from the radio. Once you initiate the QSO, the computers at either end will exchange signal reports and 73's and it's all over in a few moments. That attracts some controversy, it tends to be one of those love it or hate it things… but this article is not going to dwell on the details of digital modes.

The other modes in the WSJT-X (WSPR, JT4, JT9, JT65, FT4 etc) are also MFSK modes. A radio transceiver handling MFSK and FSK modes therefore caters for the majority of digital modes in use on the HF bands today.

But MFSK is not SSB!

Traditionally we needed an SSB transceiver, to transmit and receive digital modes. But why is that? FSK is a single tone, shifted in frequency. It is not SSB, it has never been SSB. Most of us will have built CW transceivers, and know that to build an SSB transceiver is a more complex and expensive task.

To be sure, an SSB transceiver could be used to transmit CW! Make a keyed audio oscillator, and feed that into the mic input of the SSB transceiver. But the SSB transceiver is more complex, and more expensive, than a CW transceiver. It also produces a less clean output since try as you will, you will find some residual carrier and residual unwanted sideband remain. Furthermore, since you'll be using a linear power amplifier in the SSB transceiver, having relatively low efficiency, the radio will consume more power than a purpose-built CW transceiver.

All the same arguments apply to MFSK digital modes. It IS just a carrier, whose frequency is shifted. An SSB modulator is not required. A linear power amplifier is not required. A simpler transceiver architecture would have many advantages.

So why, in that case, have people always used SSB transceivers for digital modes?

The answer is simple. The lowest common denominator is audio. Any SSB transceiver necessarily has an audio input (microphone) and an audio output (headphones). If you are a software writer, coding software for a digital mode, the most sensible thing to do is to make your program use audio for the input and output. By doing so, your software is automatically completely compatible with every SSB transceiver in the world. Audio is THE common interface almost every commercial SSB transceiver already supports. If you want your software to be accepted and easy to use and popular – the best way to do that is to be compatible with radios that are already out there in the wild. Trying to sell people a special kind of computer interface and a digital radio, just to be able to try digital modes which may or may not ever become popular, is a non-starter. Fit-the-existing-world is easier than making a new one. So audio it is, and an SSB transceiver to transmit and receive it!

A new digital transceiver strategy for QDX

Despite the universal use of SSB transceivers for digital modes, I realized that there IS a better alternative, which is also compatible with all existing software such as WSJT-X, JS8Call etc.

The Si5351A synthesizer chip, or any DDS chips of the last few decades, can produce a synthesized RF output, both highly accurate and highly stable. My idea here is to measure the audio frequency coming from the PC, and to add it to the "USB Dial Frequency" that the radio's VFO sits on, then command the synthesizer to produce exactly that frequency. The audio frequency has to be measured continuously and keep updating the frequency, to follow the frequency shifts of the audio tones coming from the host PC.

There are a number of significant advantages to this approach, and a few disadvantages too.

Advantages:

- Simple circuits can be used, no SSB exciter is required. Just the synthesizer feeding the power amplifier, as it would in a CW transmitter. This improves performance and lowers cost.
- No linear amplifier is needed – we can use a Class C, D or E amplifier with higher efficiency, simpler and easier to set up, and we won't need a heatsink. No question of linearity performance (for example third order modulation products IMD3 etc), or splatter – they just don't exist.
- Even the transmit/receive switch is very much simplified in this architecture, similar to a CW transceiver.
- There is no question of unwanted sideband or residual carrier that would exist in an SSB transceiver. Here, these unwanted artifacts are, quite simply, completely absent!
- The audio drive level is of no consequence; no need for careful setting of the volume control on the PC (too high will overdrive causing distortion and splatter, too low will reduce the power output). Here, the power output is always at the normal maximum level. You can't overload it, you can't cause splatter, you can't cause low power output by not enough volume.

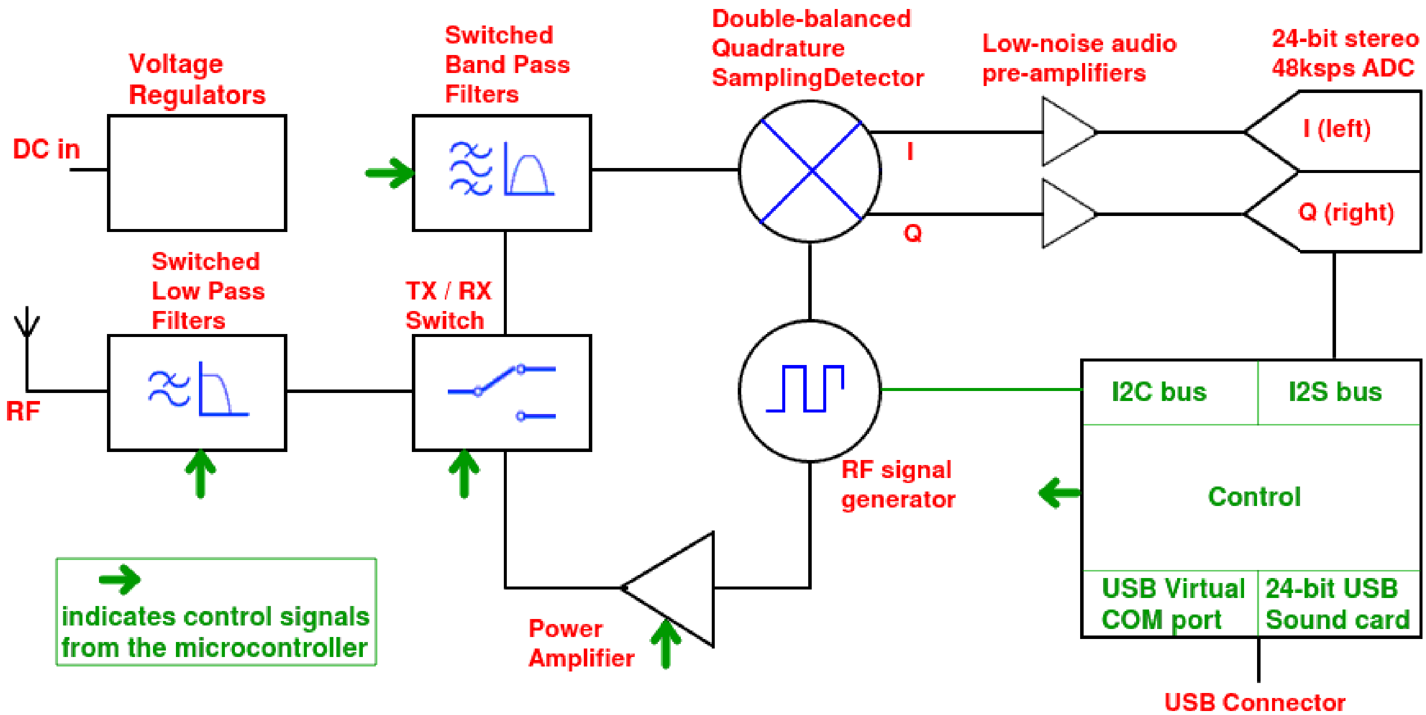
Disadvantages:

- This method won't work for modes other than FSK/MFSK. Phase Shift Keying (PSK) or any mode that transmits multiple tones concurrently, still require a SSB exciter and linear amplifier.
- You can't adjust to a lower power output by reducing the PC sound output volume. You always have maximum power (5W in this case).

Note: A similar technique has been independently developed by QRP Guys and implemented in their digital transceiver kit. In my opinion, it greatly improves the performance of the transmitter side of the kit and was a very good development. Their radio uses the ATmega328 and uses timing between zero crossings to calculate the audio frequency and command the Si5351A.

QDX Block Diagram

Before talking in detail about various interesting aspects of the QDX design, let's take a look at the block diagram of what's on that little PCB.



In summary:

- We have a set of Low Pass Filters, which are switched under control of the microcontroller, using solid state (PIN diode) switching.
- A transmit/receive switch routes the signal from the LPF to the receiver section, or the transmitter section.
- The transmit section consists merely of a 5W power amplifier driven directly by the RF signal generator (Si5351A Synthesizer).
- There are a set of four band pass filters, for the four bands supported, again switched under control of the microcontroller.
- A Quadrature Sampling Detector (QSD a.k.a. Tayloe detector) provides I & Q baseband outputs, which are amplified by a pair of differential amplifiers, feeding a high performance 24-bit stereo Analog to Digital Converter that takes 48,000 samples per second (48ksps).
- Finally at bottom right, a powerful 32-bit Cortex M4 ARM microcontroller running at 72 MHz. It interfaces with all the other sections of the radio. A high performance Software Defined Radio receiver is implemented. The host PC interface is a single USB cable carrying serial data and digitized output audio data.

Computer interface

It makes sense to talk first about the host computer interface, before discussing the transmit side of the QDX transceiver then last, the receiver. In QDX, the computer interface is a key factor in the achieved high performance.

Many digital stations use audio cables to carry analog sound to and from the PC, to feed the SSB transceiver.

There are good audio cables, there are bad ones. You've heard the jokes about audiophilic types with their oxygen-infused, purified silver, double-shielded speaker cables, 100-hours burn-in by the manufacturer, complete with arrows indicating which direction the sound has to travel in… and complete with a $$$ per meter price tag? At the other end of the spectrum when you pop into Dollar Tree (etc.) and buy a $1 audio hookup cable with a 3.5mm stereo jack either end, don't be surprised if you cut it open and find three hair-thin wires side by side without any kind of shielding at all.

Whether one extreme or the other, you are liable to find hum caused by ground loops and these kinds of problems can be hard to resolve completely.

QDX avoids all this by transferring audio from the PC to the QDX transceiver digitally, using the USB Digital Audio device class. In other words, QDX appears to the PC as a USB sound card. There are tremendous advantages to doing this.

On transmit: conventionally, transmitted audio would be generated digitally in the PC, then converted to analog by a Digital to Analog Converter (DAC), passed along an audio cable, then modulated to SSB; or in many cases including ours, converted back to the digital domain for analysis, using an Analog to Digital Converter (ADC). All these conversions and analog connections introduce noise, losses and the potential for hum pickup. Keeping it all digital via USB results in a zero noise, zero loss audio sample transfer. This will be important as you will see later, in the audio frequency measurement.

On receive: the ADC of many laptops is rather poor, and in desktop PC motherboards, not a lot better; there's often little effort to shield the IC from nearby sources of interference and the dynamic range and noise floor performance is really not important. The human ear can't hear beyond about 12-bits of sound resolution (72dB). But in a radio receiver application where large signals may exist next to weak ones, more dynamic range is desirable. Implementing the audio conversion in QDX provides the full benefit of the available dynamic range of the ADC chip used. Particularly if the QDX is used as an SDR front-end providing I & Q baseband to the PC for SDR processing, dynamic range is even more important. Every 1 bit of actual resolution (not noise) from an ADC is equivalent to 6dB in dynamic range terms. A 16-bit sound card is limited to at best, 96dB dynamic range; practically speaking, the lowest few bits are noise so the real dynamic range is less than this. This is why a 24-bit ADC is important.

You can use a high-end USB sound card like the Focusrite Scarlett Solo I use in my lab measurements, or a $3 Amazon dongle. Don't expect the same performance. In fact, the performance of the embedded soundcard in QDX demonstrably exceeds that of the professional Focusrite unit on the left.

In addition to USB digital audio, the QDX transceiver microcontroller also needs to receive commands from the PC host to set the frequency and band. This is provided by a Virtual COM Serial port emulation USB device in the QDX microcontroller. QDX includes a CAT (Computer Aided Transceiver) command interface. CAT is the standard way that computers communicate with amateur radio transceivers. All digital mode software for PCs supports CAT for controlling the radio, and they all support selection of audio devices such as the PC's built-in audio if it has it, or USB sound cards.

Technically, since the single USB cable provides both a Virtual COM Serial port as well as a USB soundcard, there's also the equivalent of a virtual USB hub inside QDX too – with the two other devices (Serial and Audio) plugged into it.

If all that wasn't enough, there's also a way for the QDX owner to upgrade the firmware in QDX; for this, QDX appears as a USB Flash drive and the new firmware file is copied in.

Importantly, all of these USB devices contained within QDX are standard USB devices (audio, serial, hub, flash drive). It means I didn't have to write any USB driver software, or any PC software. It means that QDX works on Windows, Mac or Linux (the exception is windows 8 and earlier, which need installation of an ST Microelectronics Virtual COM driver).

That's a whole lot of USB going on inside QDX's microcontroller! It's key to the performance and features of the QDX.

Audio Frequency measurement

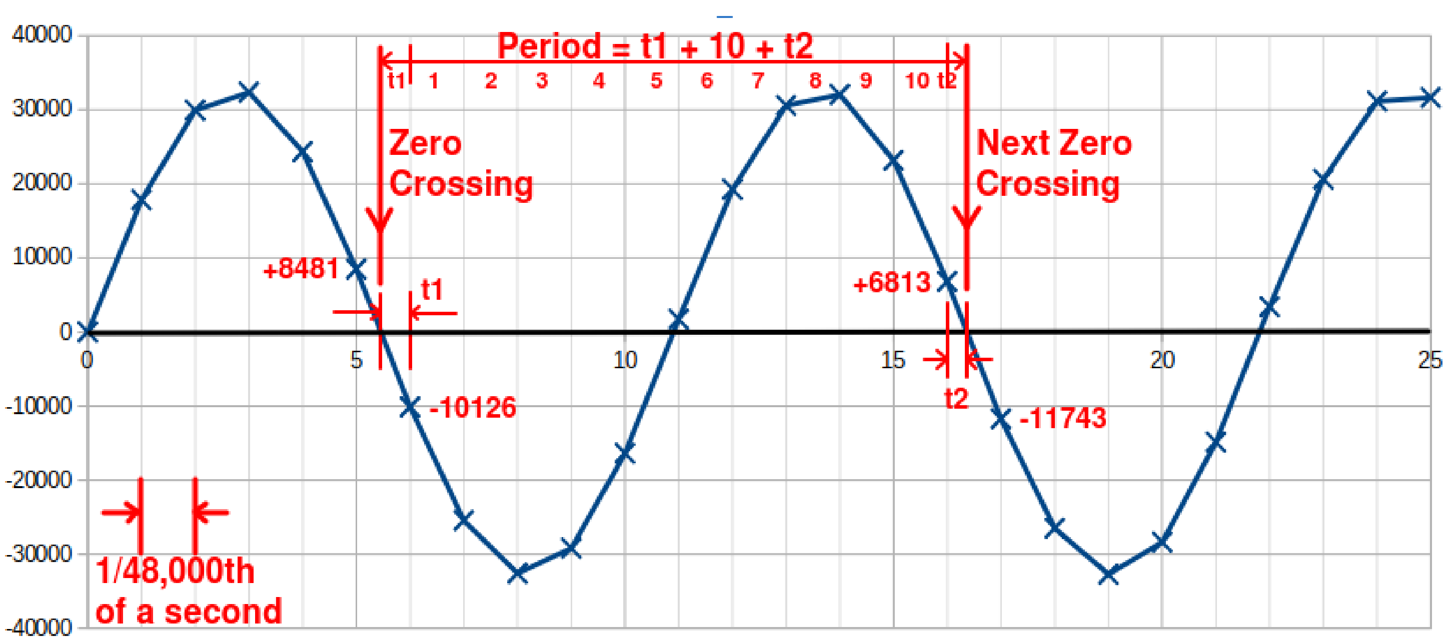
I mentioned earlier that the strategy was going to be, measure the audio tone frequency from the PC host software, add it to the "USB dial frequency" and command the synthesizer to produce that output frequency. This part of the QDX design is right at the heart of what QDX is and why it works so well.

My first thought was to use Fast Fourier Transform (FFT). However I only had to dig into this very slightly, a rather shallow grave for myself, to realize this would be nearly impossible. The WSJT-X modes are highly optimized, with tone spacing which is the reciprocal of the symbol duration. The sizing of the FFT buckets would be critical and it would be necessary to somehow run overlapping FFT to try and synchronize the bucket frequencies to the tones being transmitted. WSJT-X can do this, on a powerful modern PC. But I can't, not on a microcontroller.

Next I wondered whether it would be possible, and accurate enough, to measure the period of the incoming audio sinewave. Frequency counting in the conventional sense is too slow – for example to obtain a 1Hz resolution, you must count pulses for 1 second. Again the required accuracy (smaller than the tone spacing) and requirement for a measurement duration less than the symbol duration, make this impossible.

It turned out that measuring the period of the sinewave was surprisingly accurate, easy and fast. QDX can obtain an accuracy of for example, +/-0.05Hz when measuring a single audio cycle at around 1500Hz. This accuracy, and measurement speed, is plenty enough for any of the digital modes commonly used.

The embedded USB sound card emulation turns out to be very useful here. WSJT-X generates the audio sinewaves using a software-implemented Direct Digital Synthesis (DDS) algorithm and sends those to the QDX USB audio device. WSJT-X uses 16-bit resolution at 48ksps (48,000 samples per second) for its generated sinewaves. The numbers streaming in from the USB interface are EXACTLY the same numbers that WSJT-X generated. There is no noise, no loss, no hum – just perfect digital data transfer of the audio samples. This makes it possible to determine the zero crossing points with high accuracy.



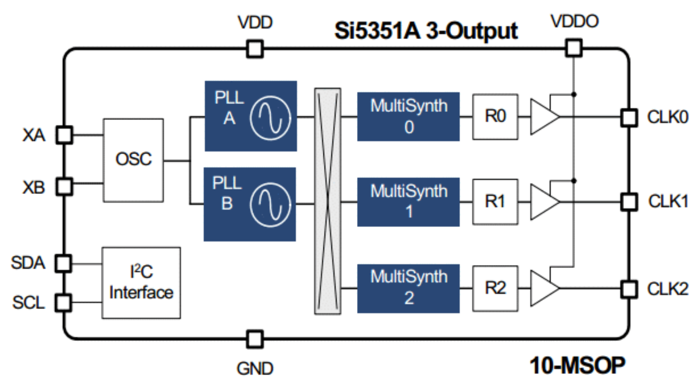
Each of the numeric samples representing the incoming audio sinewave is a 16-bit signed number, between -32768 and +32767 (the WSJT-X software on the PC operates in 16-bit arithmetic). The analysis determines the point of zero-crossing, on the downward slope of the sinewave, by looking for a positive sample followed by a negative one. We know the time interval between each sample is 1/48,000'th of a second because the sample rate is 48ksps (this is the native sample rate output by WSJT-X – even if it was not, the sample rate of the QDX soundcard emulation is 48ksps and the PC would make sure that this is the audio format sent).

In the above example, there are 10 whole sample intervals between the zero crossings, and two partial time intervals named t1 and t2. The period, measured in units of 1/48,000'ths of a second, is therefore t1 + 10 + t2. The partial intervals can be easily calculated using a straight line linear interpolation between the positive sample to the left of the zero crossing, and the negative sample to the right. To calculate the audio frequency, we just take the reciprocal of the period.

I tried playing with other more accurate methods too. Particularly at high audio frequencies where there are fewer samples per cycle, the straight line approximation at the zero crossing point is less accurate. I tried fitting a sinewave. The results were indeed more accurate but it also consumed a **lot** more CPU processing power, and the small benefit in accuracy wasn't really worth it; the simple method outlined above is already highly accurate, really much more accurate than we need.

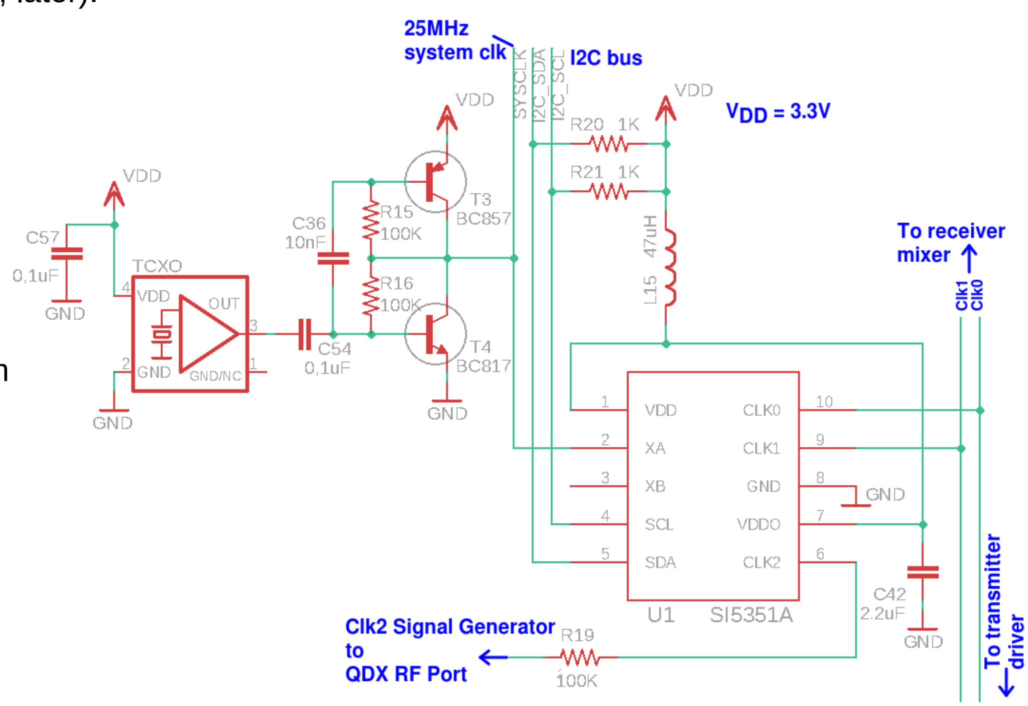
RF synthesizer

QDX uses the popular Si5351A Synthesizer chip, or its equivalent MS5351M. By now, many of you may be familiar with this amazing and inexpensive chip, which is used in many QRP radios and projects. The device contains two configurable PLL subsystems and three division stages. It can produce three independent outputs with frequencies from 3.5kHz to 200MHz, at a frequency stability governed by the reference oscillator.



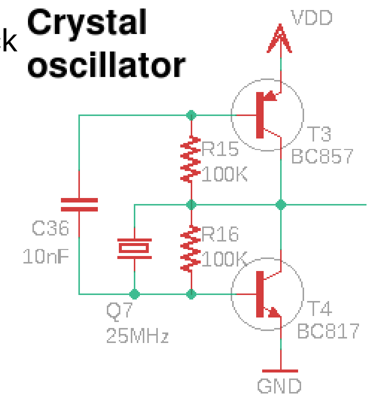
It is also possible to configure two outputs to have the same frequency, but a fixed 90-degree phase offset; a fact which greatly simplifies any radio receiver architecture using the Quadrature Sampling Detector (QSD a.k.a. Tayloe detector). In the QDX two outputs are also used with a 180-degree offset during transmit (more on this, later).

Since excellent frequency stability is needed for many narrow bandwidth digital modes, the reference oscillator in QDX is a 25MHz TCXO (Temperature Compensated Crystal Oscillator). This prevents heating during operation from causing any significant frequency drift.



The output of the TCXO is 25MHz with 1.1V peak-peak amplitude. This is perfect for driving the oscillator input of the Si5351A; however in QDX the 25MHz reference is also used to clock the microcontroller. The reason for this is that it is a very good design rule, to try to minimize the number of different oscillators in a radio design – harmonics of oscillators somehow find a way to mix and create receiver birdies.

It was found that the 1.1V signal was of insufficient amplitude to reliably clock the microcontroller. Which brings us to a quite interesting part of this circuit block: the two-transistor buffer circuit shown at the Si5351A output. I can't remember where I found this circuit, but I have not come across it anywhere else. It's rather rare! I don't even know what it should properly be called. Yet it is certainly very useful and I have used it in a number of projects now. It was originally a crystal oscillator (right).



It's remarkable in so far as:

- It needs one NPN and one PNP transistor, which are not at all critical.
- No change in component values for a wide range of operating frequencies.
- The output is low impedance and quite close to a rail-to-rail squarewave.
- Seems to "just work", with any crystal I have ever tried – and what's not to love, about a circuit that "just works", every time!

After deciding to use the 25MHz TCXO rather than a crystal, I kept the two-transistor circuit, which functions just as well as a rail-to-rail buffer, in this case producing a 3.3V output suitable for clocking the microcontroller and the Si5351A.

<u>An innovative new Class-D power amplifier</u>

Power amplifiers are a whole topic in themselves! For QDX, I developed a new QRP amplifier that is efficient, multi-band, and has a relaxed LPF requirement.

The "class" of an amplifier really depends on what proportion of the RF cycle the transistor conducts for. It's important to select the class of operation appropriate to the application. A key criteria is whether linear operation is required or not; in a linear amplifier, the output waveform is the same shape as the input, just bigger. In other words, there isn't any distortion of the waveform.

- Class A: There's a single transistor that conducts throughout the entire cycle, biased into its linear region. It provides linear amplification, but the significant bias current means it consumes plenty of power even when there is no input signal, and efficiency is rather low.
- Class B: The transistor conducts for half the RF cycle. Normally two transistors are used in a push-pull configuration to amplify the whole RF cycle with low distortion. Class B amplifiers are more efficient than class A but have higher distortion (worse linearity).
- Class AB: somewhere between Class A and Class B – some level of bias is applied so that each of the two transistors conducts for more than half the RF cycle; It has both higher efficiency than class A and lower distortion than Class B and is in many ways the best of both worlds. Ham radio "Linears" are Class AB amplifiers.
- Class C: The transistor conducts for half the time, producing a very distorted amplified version of the input signal, rich in harmonic content. For a CW transmitter, Class C is fine since there is

only a single input frequency, so the distortion doesn't cause within-band mixing products, only harmonics of the operating frequency which can be easily filtered out.

- Class D: The transistor is fully saturated as a switch, either on or off, by a large amplitude input signal. The term Class D normally describes audio amplifiers or motor drivers in which the transistor is switched at a high frequency, using pulses whose width (Pulse Width Modulation, PWM) reconstructs the audio signal. Class D amplifiers are very efficient because the transistor switch is either On (high current, but near zero resistance and therefore voltage) or Off (zero current, high resistance, high voltage across the device). Either the current through the device is zero, or the voltage across the device is zero, so the power lost in the device (power being voltage multiplied by current) is also zero. In practice the transition from On to Off is non-instantaneous and the transistor isn't a perfect switch, so some power is dissipated.
- RF Class D: I am using the "Class D" term in the context of this article, to mean an amplifier which is switched On or Off in saturation at the operating frequency; there is no PWM. It is also quite an efficient amplifier, more than Class C. It is also non-linear so only suited for CW or single tone frequency shift modes, and needs a strong low pass filter to tidy up harmonics.
- Class E: In a Class E amplifier such as the QRP Labs QCX, the load is a resonant circuit at the operating frequency. Like Class D, there is either zero current or zero voltage, so the power loss is low. But in the case of Class E, the amplifier timing is such that the effects of delays in the device switching are minimized. A Class E amplifier is highly non-linear, suitable for CW or FSK only, but can achieve efficiencies over 90%. A disadvantage is that a Class-E amplifier with its resonant load, is inherently single-band unless complex switching is used, that risks degrading the efficiency.
- Classes F, G, H etc.: seriously, let's not even go there…

For the QDX, Class E was ruled out because of its single-band nature (unless more complexity had been introduced). But high efficiency is still very desirable - not least because for portable operations, low power consumption is beneficial; but I was thinking mostly about the need to avoid a large heatsink. So the next best thing is Class-D.

Now a minor digression. an important property of a push-pull amplifier is that even harmonics cancel themselves out, leaving only the odd harmonics. Practically speaking, since devices never have absolutely identical characteristics, there are still even harmonics, but at a quite low level. This is very useful from the perspective of designing Low Pass Filters because it means the first significant harmonic we need to tackle, is the 3rd. Accordingly such a sharp cut-off as needed in Class E for example, is not required.

This is one reason why we often see Low Pass Filtering that follows Class AB linear amplifiers, having a smaller number of filters than the number of supported bands, and having filters with less elements (a more gradual attenuation of harmonics). It is common to see five or six filters cover all ten HF bands because one filter can be shared by two adjacent bands. The other reason is that the Class AB amplifier is supposed to be linear, so harmonic content should be very low in any event.

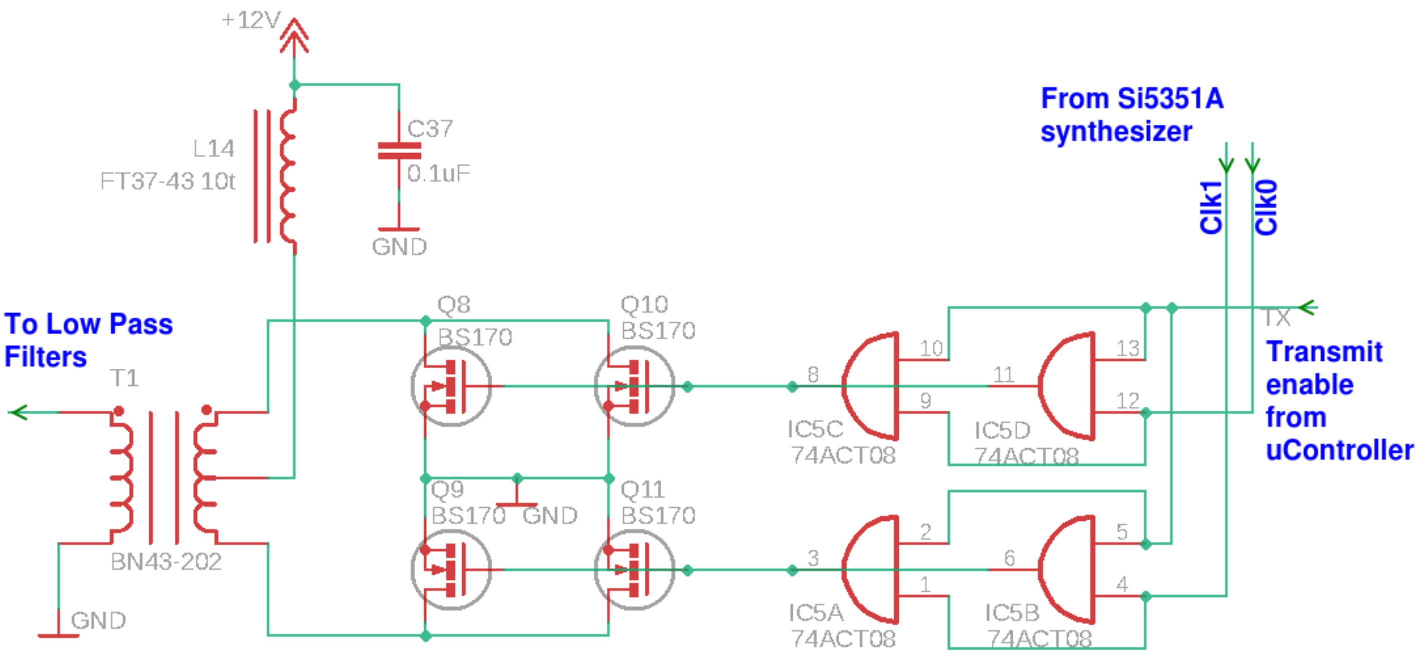
Many QRP transceivers use three BS170 MOSFETs in parallel, either in Class C or Class E (such as QRP Labs QCX-series transceivers). But another factor weighing on my mind was that digital modes can be 100% duty cycle for prolonged transmission duration – for example, WSPR is a 112 seconds continuous carrier transmission. I admit that it makes me ever so slightly nervous, to coerce three

BS170s into producing 5W output power at high duty cycle, particularly if the high efficiency of Class-E is not achieved.

So for QDX I had the idea to combine Class D (saturated switching amplifier) with the push-pull concept. The layout of the amplifier is rather similar to a typical Class AB Linear. Two transistors are driven with 180-degree out of phase input signals. The out-of-phase output signals are combined in a transformer that feeds the load (antenna). The difference, in the case of QDX, is that the transistors are driven to saturation, switching-style. Not only do we not bother to strive for linear operation, we actually don't WANT it because it wastes power as heat. Another difference, is that whereas a Class AB linear has an input transformer to split the signal into the two opposing phases, in QDX this is deleted; the Si5351A can, all by itself, produce two signals in perfect anti-phase (180-degree phase shift).

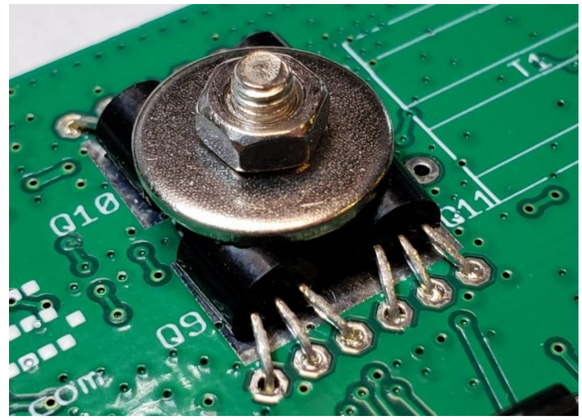
Each "transistor" is actually two BS170 in parallel. The total number of transistors is therefore four, and I could be more relaxed about the dissipated power getting shared between four, rather than three. The gates are driven by pairs of fast 74ACT08 logic AND gates that convert the 3.3V peak-peak square wave from the Si5351A to 5V peak-peak which is sufficient to drive the BS170 transistors to saturation. Everything was starting to work out nicely!

The combining transformer is a binocular core BN202-43 and an FT37-43 RF choke feeds DC power in at its center tap. It's therefore a circuit derived directly from a Push-Pull Class AB Linear amplifier, but with the transistors driven to Class D saturation by logic gates, and no need for an input transformer, since I can, in software, command the Si5351A to produce 180-degree out of phase outputs to drive the push-pull circuit perfectly.

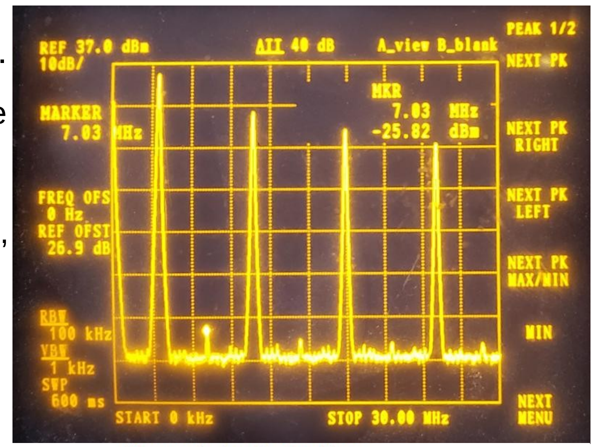


The transistors are bolted against the bare PCB (no solder mask under them) by a 9mm screw and 12mm diameter washer pushing the transistor flats down against the PCB. The PCB therefore acts as something of a heatsink, and no "real" heatsinking is required. Some people mistakenly think that the 12mm washer is the heatsink; it isn't, in reality the contact area between the rounded top of the TO92 transistor package and the washer is a very small area. The point of the washer is to hold the transistors firmly down flat against the PCB.

In practice, this amplifier worked extremely well, I was pleasantly surprised by how well! On my first prototype (40m band), the 2nd harmonic measured -65dBc, with no Low Pass Filtering at all. The amplifier has no problem producing 5W on all four intended bands 20, 30, 40 and 80m. It doesn't mind 112 seconds of continuous duty cycle WSPR transmission. There are no band-dependent parts in the amplifier so nothing prevents easy use in a multi-band transceiver.

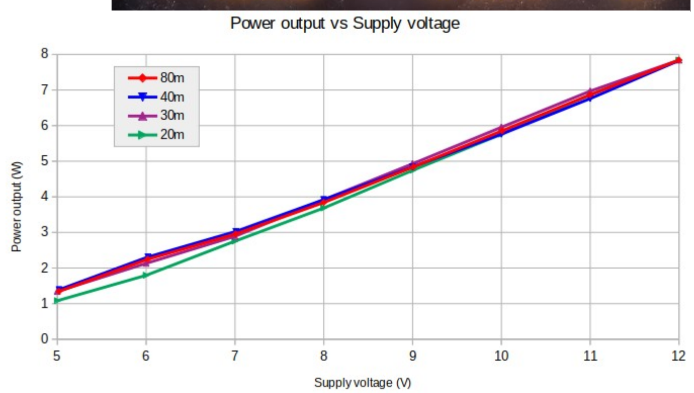


In this spectrum analyzer screenshot of 80m band operation, the 2nd harmonic is seen at -70dBc, again without any low pass filtering. The odd harmonics are at a much higher level.



If T1 is wound as a 3:3 turn transformer with 1.5 turn tap, the circuit produces 5W power output from around 9V DC supply. With a 3:2 turn transformer, the circuit produces 5W from about 12V supply. The amplifier can be built either way, depending on what is most convenient regarding power supplies, for example portable (battery) operation.

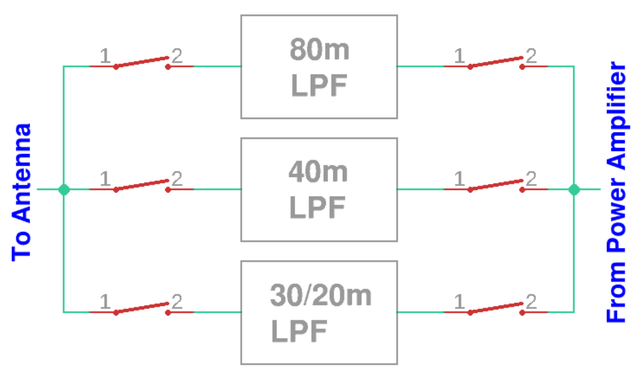
Power output varies according to the supply voltage; the chart (right) shows power output vs supply voltage for a 3:3 wound T1 (intended for 9V supply). It shows 5W at 9V supply and very little variation in power output by band.



It's a really neat QRP amplifier and I expect we will see more use of this design in other people's projects too.

Low Pass Filters

Low Pass Filtering is required in any transmitter, to remove harmonic content of the wanted transmission. Or more accurately put, to attenuate it to levels below that required by regulators. For example in the US the FCC stipulates unwanted harmonic content must be not more than -43dBc.

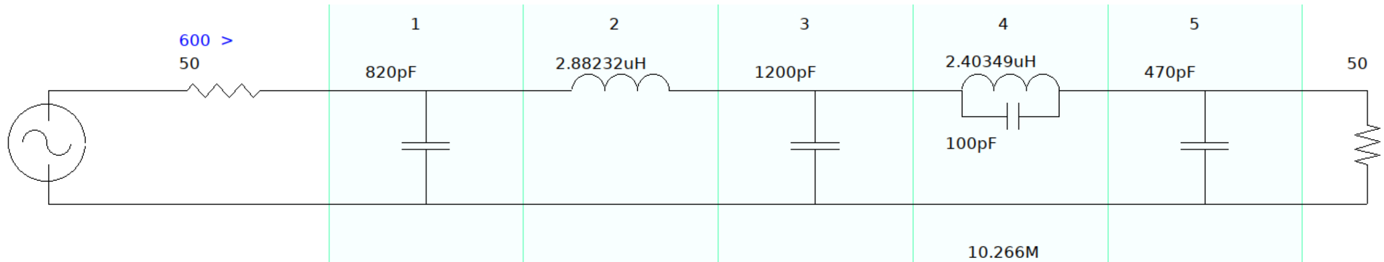


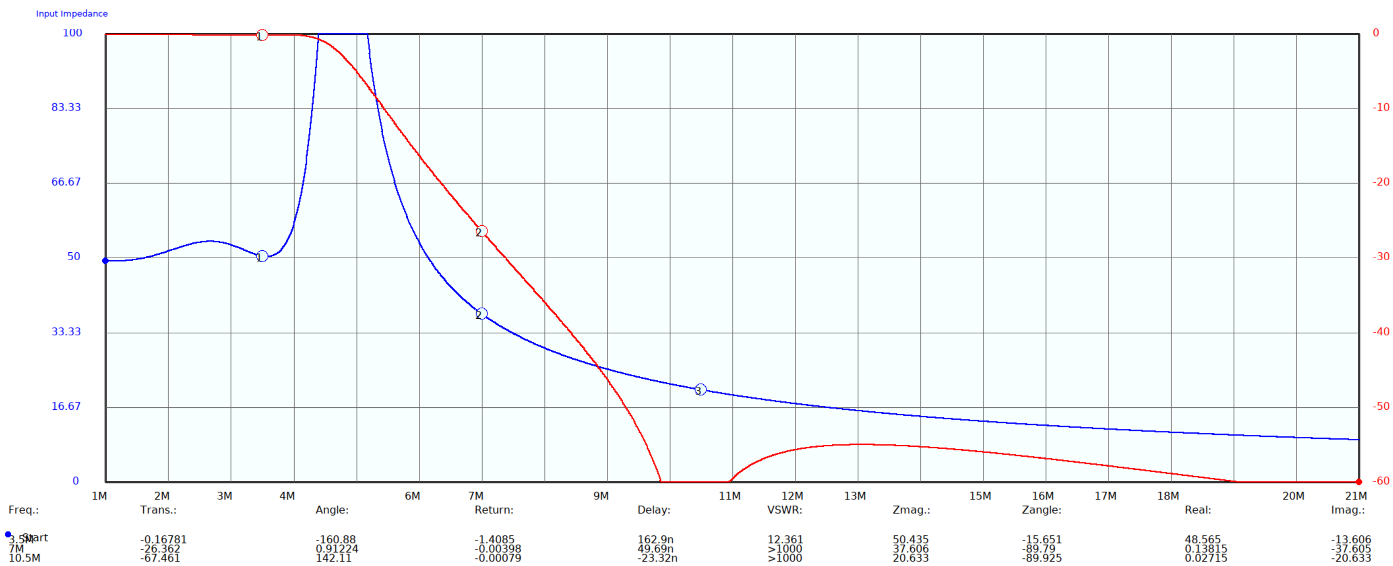
Three filters are used in the 80-40-30-20m QDX. 80m and 40m bands have their own filter, whereas 30m and 20m share one filter.

The filters were designed using the freeware Elsie program by Tonne software (http://tonnesoftware.com/elsie.html). The filter topology is a 5-element filter, with a 6th (capacitor) creating a resonant trap. For the 80 and 40m filters, the design was optimized to satisfy the following criteria:

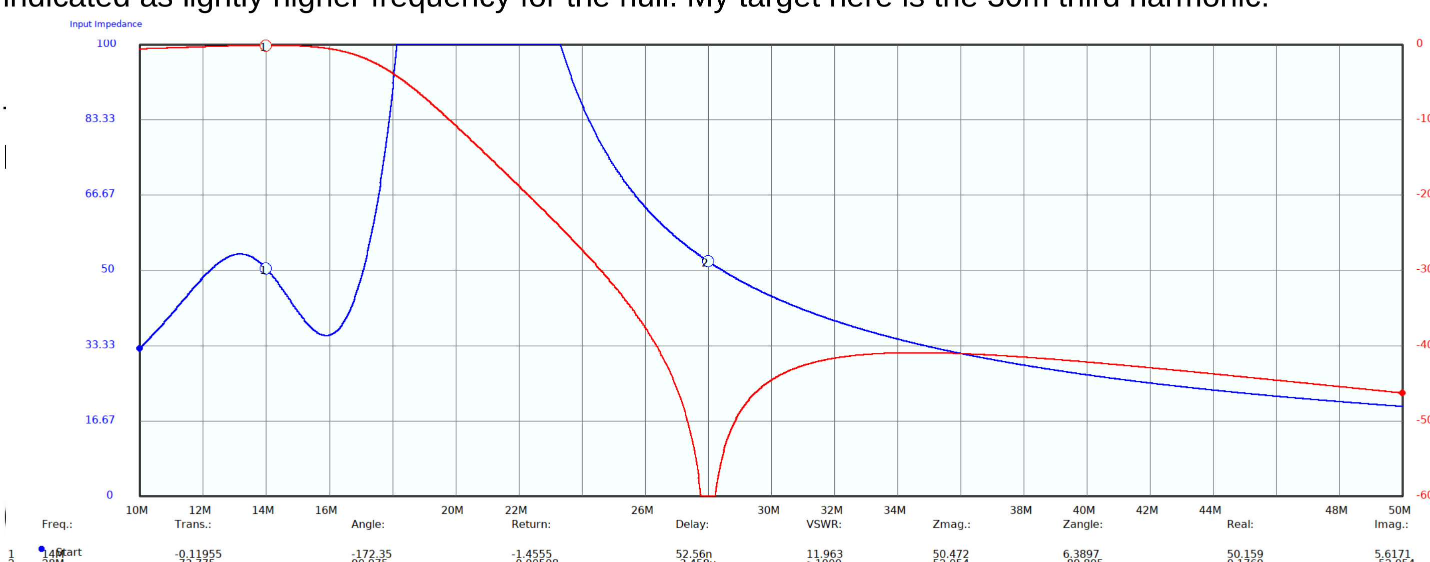
- 50-ohm impedance at the operating frequency.
- Low insertion loss at the operating frequency.
- Null (the resonant trap) at the third harmonic.
- Uses available component values (specifically the capacitors, since the inductors can be wound for any inductance).

This is the design and modeled response of the 80m Low Pass Filter. You can see that the capacitor values are standard, the operating frequency impedance is 50-ohms, and a deep null in the response occurs at the third harmonic (10.5MHz).





The 40m filter design and response is similar. The 30/20m filter design was done slightly differently in order to place the null at 28MHz (according to the model). Measurements of the actual circuit indicated as lightly higher frequency for the null. My target here is the 30m third harmonic:



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

60m band operation

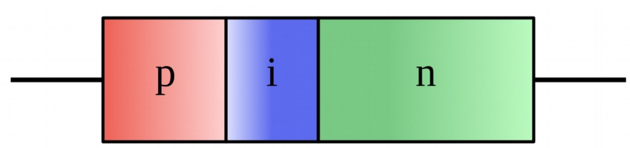
Experiments by John AE5X https://ae5x.blogspot.com/2021/11/spectrum-analysis-qdx-on-60m.html also verified QDX works well on the 60m band; he measured approximately -50dBc 2^{nd} harmonic and -60dBc third harmonic.

Low Pass Filter switching

I really dislike relays for switching, for multiple reasons. They're quick and easy but they have a number of disadvantages, including:

- Bulky and heavy
- Expensive
- Mechanical – sooner or later they may fail or become unreliable

I hadn't tried PIN diode switching for Low Pass Filters before but saw no reason why it shouldn't be a good alternative to relays.



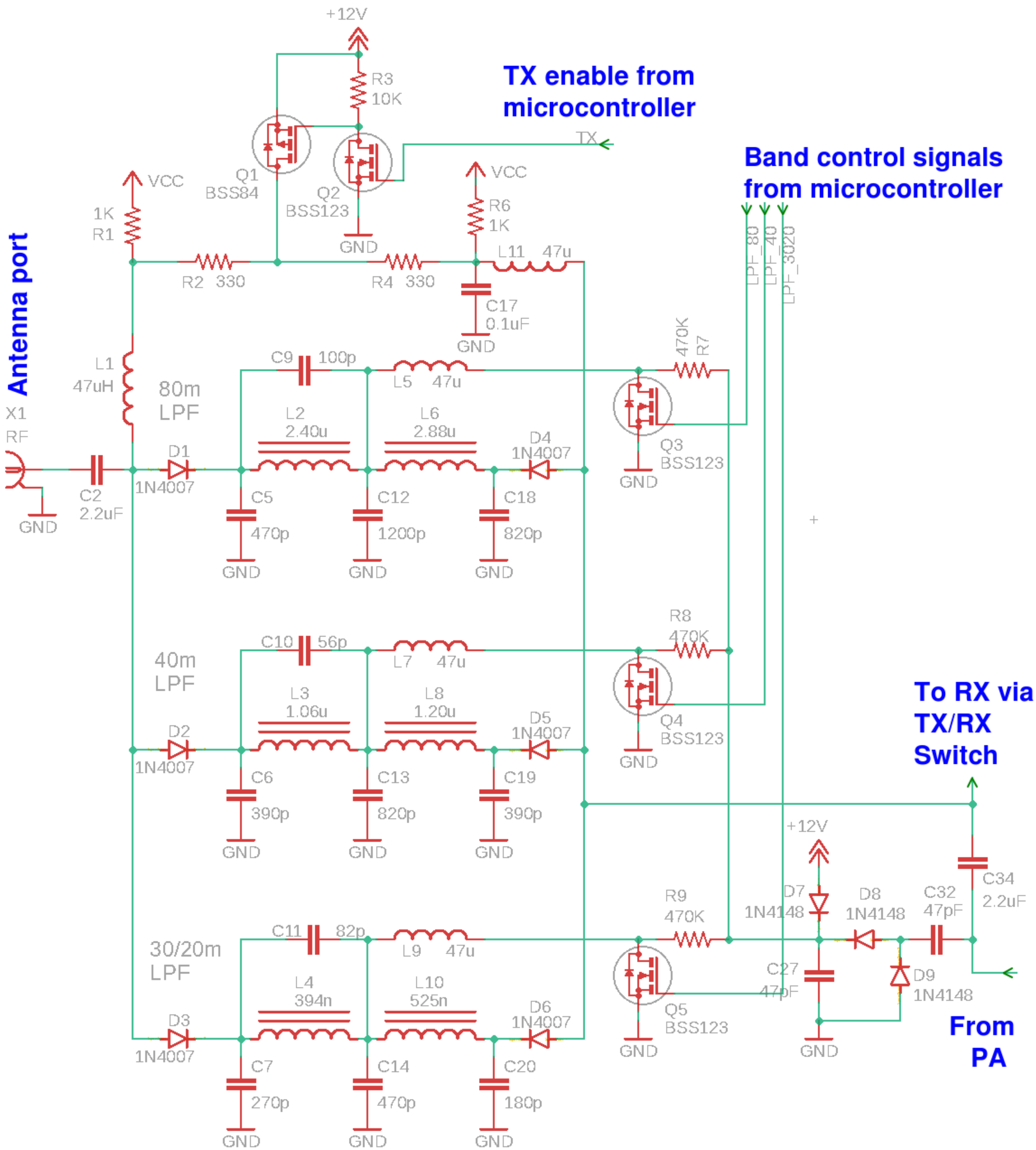
A PIN diode has the normal P-doped and N-doped regions at either and an an "Intrinsic semiconductor" layer in the middle. The effect is to slow down the action of the diode. When acting as a switch, it is not fast enough to rectify RF. PIN diodes can be expensive components and hard to find. But luckily, the common 1N4007 diode, with its 1,000V PIV rating, happens to have a type of construction resembling a PIN diode, and operates very well as a poor-man's PIN diode, from the 630m band to the 10m band inclusive.

Here are the rules for PIN diodes in RF switching applications:

1. Switched ON requires a forward bias current. But it turns out to be slightly more complicated than that. Not only does insertion loss decrease the more current you pass; but the diodes are imperfect and they create 2^{nd} harmonic distortion, and the more power you pass through the diode (larger signal voltage), the worse the harmonics get. You're using filters to get rid of harmonics, and now you're ruining all that by putting harmonics back in, AFTER the filter, using switching diodes! So you have to use a large enough forward bias current, to get those harmonics down to a low level.
2. Switched OFF requires a reverse voltage. The reverse voltage has to be larger than the peak voltage of the RF waveform.
3. Frequencies lower than about 0.5MHz are slow enough that the diode acts as a normal rectifier so doesn't work well as a PIN diode.

4. Attenuation decreases with higher frequency, due to the capacitance of the diode leaking RF through the “Off” state.
5. You have to make sure you block DC on the signal path, because you need to control the state of the diode switch using a DC forward bias current and a DC reverse voltage.
6. You have to block RF from entering the bias control circuits and potentially bypassing (leaking past) the switch.
7. Make sure you use real 1N4007 diodes, it’s a rough world here in 2022 and there are lots of fakes around; some lesser diodes packaged as 1N4007 sometimes have inferior performance.

Here’s the complete schematic of the Low Pass Filters and filter switching section.

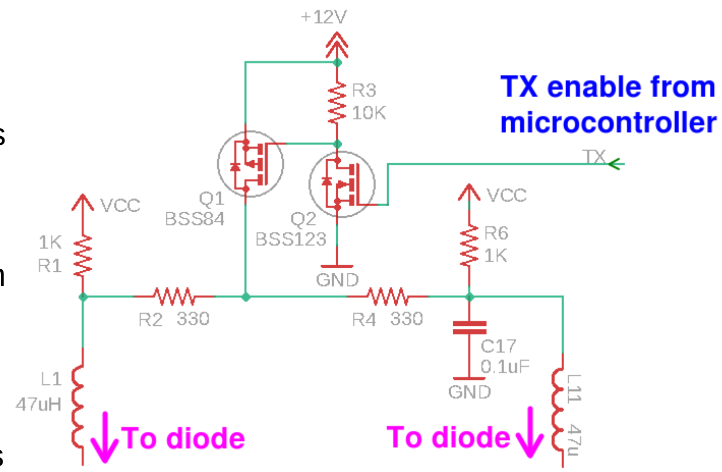


The large reverse bias voltage is generated using D8 and D9 which act as a rectifier and voltage doubler of the RF coming from the PA. They ensure that the reverse voltage that is available to switch OFF the diodes is always more than the peak voltage of the RF signal. The doubler circuit “steals” a very tiny amount of the transmitter power created in the power amplifier. During receive operation, there is no RF from the PA; in this case we take the 12V supply voltage via D7 to give a good amount of reverse bias voltage during receive (we never see signals anywhere near that big).

One of the three transistors Q3, Q4 and Q5 is switched ON by a control signal from the microcontroller, to select one of the Low Pass Filters. The other two low pass filter diode switches are biased OFF by the reverse voltage generated (see above).

The forward bias control circuit is worth of note. I'm referring to Q1 and Q2 at the top of the schematic.

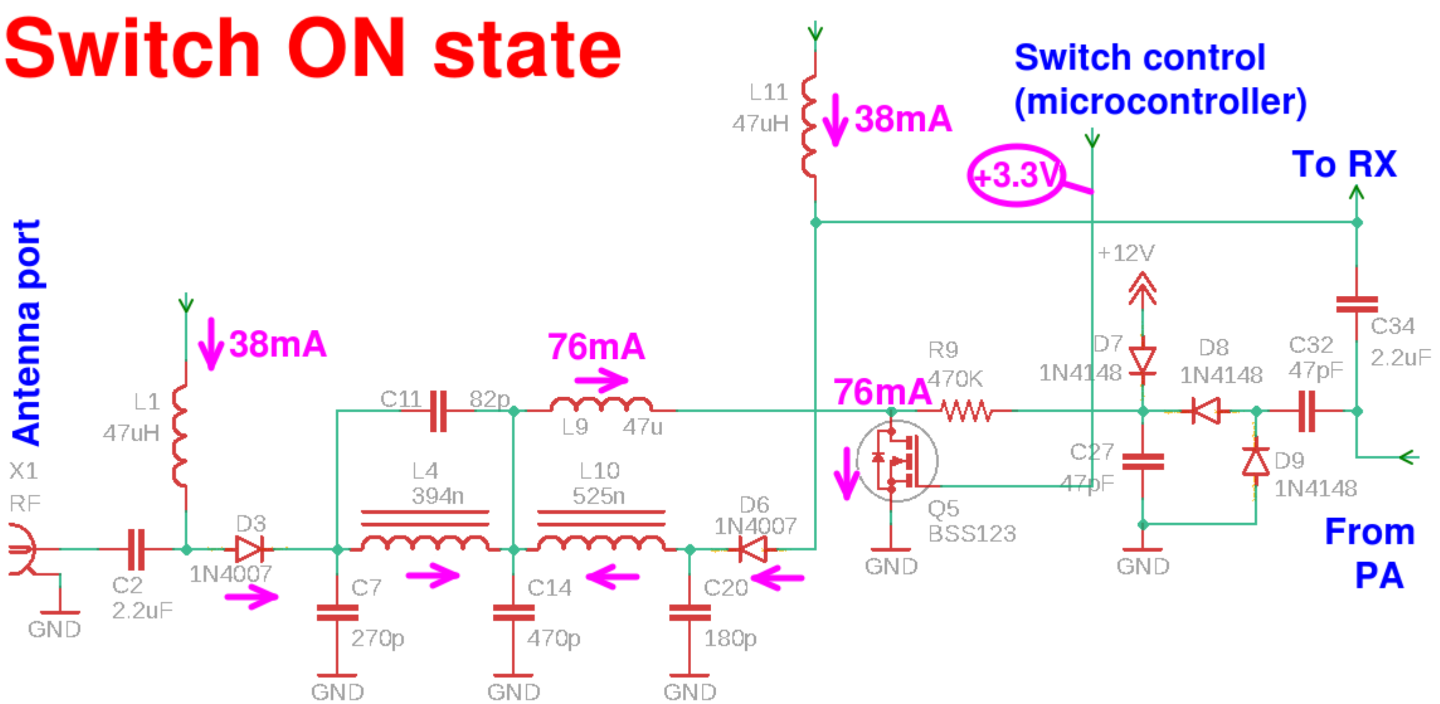
During Receive operation, the PIN-diode switches only have to handle small signals. Large bias currents aren't required. Resistors R1 and R6 pass a bias current of 4mA through each diode.



During Transmit operation, we need a lot more current than that, otherwise an unacceptable level of harmonic content will be inflicted by the diodes. Q2 (N-type MOSFET) is switched on by the TX enable signal from the microcontroller and in turn, enables Q1 (P-type MOSFET) that supplies current through R2 and R4 to the diodes; now the current is around 38mA through each diode (76mA total), which is enough to ensure clean operation.

The reason for having this arrangement to provide a different forward bias current in transmit and receive, is that we generally want to minimize current consumption in receive (for more portable-friendly characteristics). We wouldn't want to unnecessarily consume 70mA more current, during Receive.

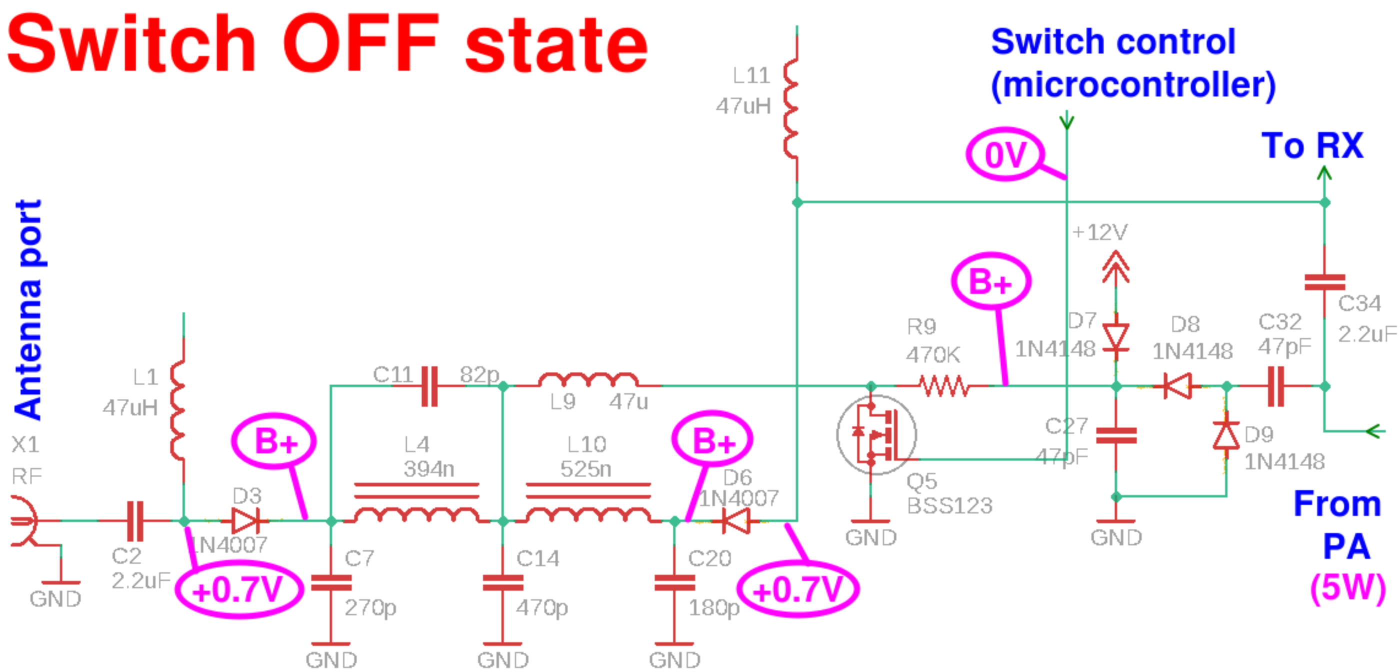
Here's a sub-set of the schematic showing a Low Pass Filter in the ON state during transmit:



The 3.3V (logic "1") signal from the microcontroller switches on Q5, which then pulls down the center point of the Low Pass Filter via inductor L9. Forward bias current flows from the bias control circuit through L1 and L11, passing about 38mA through each diode (assuming 12V supply).

Notice that the signal path is DC-blocked by capacitors C2 on the left, and C34 on the right. The control bias signals are RF-blocked by the 47uH inductors L1, L8 and L11. These components are very important (see “the rules” listed above).

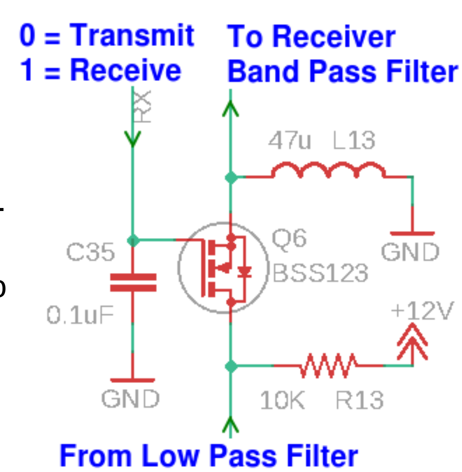
Here’s a sub-set of the schematic showing a Low Pass Filter in the OFF state during transmit:



Now Q5 is switched OFF by the 0V (logic “0”) control signal from the microcontroller. The “B+” voltage is generated by the rectifier/doubler (D8/D9) and passed through R9, L9 and L4/L10 to the diode anodes. The diode cathodes are necessarily at a potential of 0.7V because one other of the Low Pass Filters is switched on, pulling current through L1 and L11 (refer to full LPF switching schematic, above). So the reverse voltage across the diodes is B+ minus 0.7V, which, being more than the peak of the RF waveform, ensures the diodes are firmly in their OFF state.

It may seem like a lot of components, to switch things. Relays would certainly be simpler. But in practice, many of these are compact, ready-installed SMD components costing cents; this turns out to be a much more economic and reliable way of LPF switching than relays.

Solid state transmit/receive switch



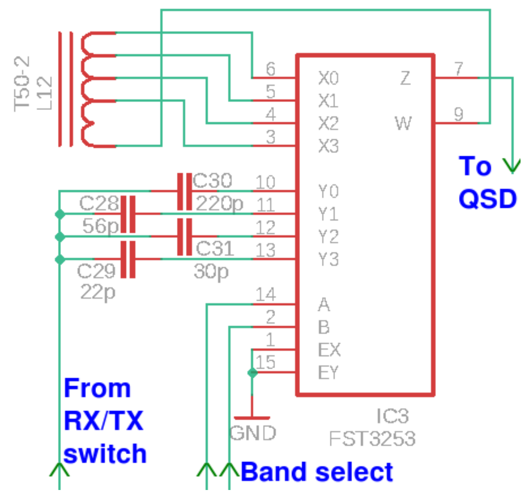
If relays are objectionable for the reasons discussed above, several more reasons exist for why NOT to use a relay as the transmit/receive switch! They are noisy, and they are slow. If it was a CW transceiver, where many experienced operators prefer full break-in operation (QSK), relays would be tortured and potentially wear out quickly. They’ll clack away noisily in time with your keying, and be too slow to permit hearing the other station between your transmitted symbols. QDX is not a CW transceiver, but my objections to relays remain!

The transmit/receive switch circuit is very simple, just a MOSFET that is switched ON to allow signals to reach the receiver, and OFF to block them. Nothing more than that is needed.

Band Pass Filtering (BPF)

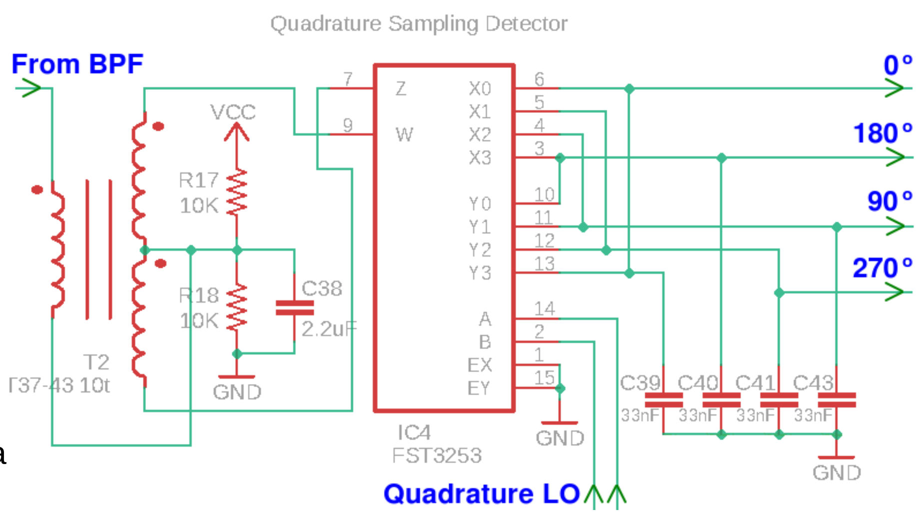
Even though the Quadrature Sampling Detector has a high performance (Dynamic range, IP3) it is still prudent to include some kind of band pass filtering in order to attenuate strong out of band signals.

In QDX, to keep things simple and minimal, a single series-resonant tuned circuit is used as the BPF. Since signals are at a low level, the dual 1:4 multiplexer FST3253 can be used as the switch. One half of it selects the appropriate capacitor to use, the other, selects a tap from an inductor wound on a T50-2 toroid. A 2.5V DC bias on the MUX switch is derived from the trifilar phase splitter in the QSD circuit (see next section).



Quadrature Sampling Detector (QSD)

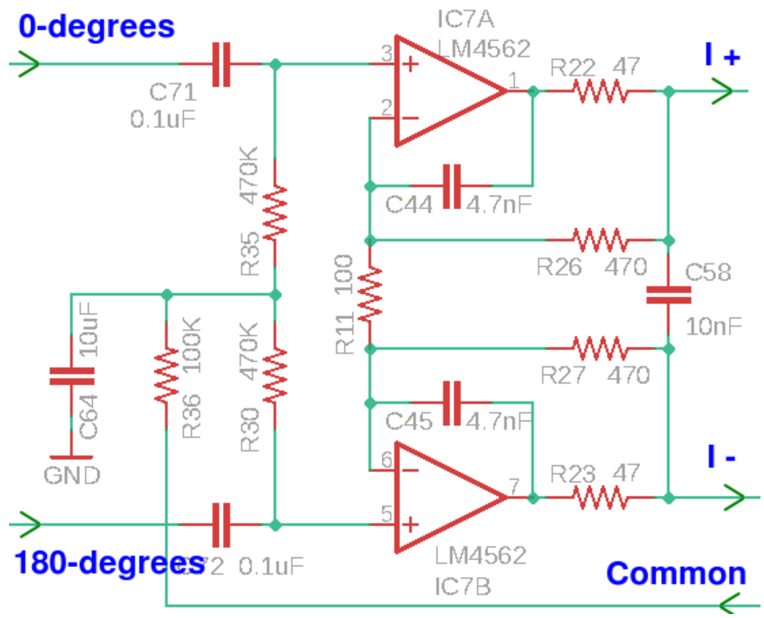
This detector is also known as the Tayloe detector. It is a simple, low cost, and very high performance of mixing RF down to baseband I & Q signals, perfect for a Software Defined Radio of for hardware demodulation such as occurs in the QRP Labs QCX-series transceivers.



QDX employs the high performance double-balanced configuration. The R17/R18 potential divider provides a 2.5V DC bias for both the Band Pass Filter switches and the QSD switches. T2 is a trifilar transformer that efficiently splits the signal into two 180-degrees out of phase signals for the detector (IC4). The integrating capacitors C39/40/41 and 43 at the four outputs need a lower value of capacitance for higher bandwidths. QDX samples baseband at 48ksps and 33nF was determined to be a good and appropriate value for this bandwidth.

At the output of the QSD are four baseband signals having 0, 90, 180 and 270-degree phase shifts relative to each other. It's mathematically sufficient to use 0 and 90-degree phases as I and Q signals, however using the 0 and 180-degrees as I differentially, and 90 and 270-degree phases as Q, has important performance benefits. Common mode noise is greatly attenuated.

Differential ADC driver



The PCM1804 Analog To Digital (ADC) chip used in the QDX is a high performance, 112dB dynamic range, 24-bit stereo converter with differential inputs. To achieve the full performance potentials of this device, it is necessary to feed in positive and negative differential inputs; they are the same signal but 180-degrees out of phase.

For each of the I & Q channels that will be converted as the Left and Right channels respectively of the ADC, we have corresponding 180-degree differential signals. However, the signal level is very low, and needs somewhere between 20 and 30dB of gain to optimally use the enormous dynamic range of this high performance ADC.

Differential amplifier ICs are available but expensive, and less commonly available parts. It is preferable to assemble a differential amplifier from normal operational amplifiers. The LM4562 has been a QRP Labs favourite, offering a good balance between performance (low noise) and price.

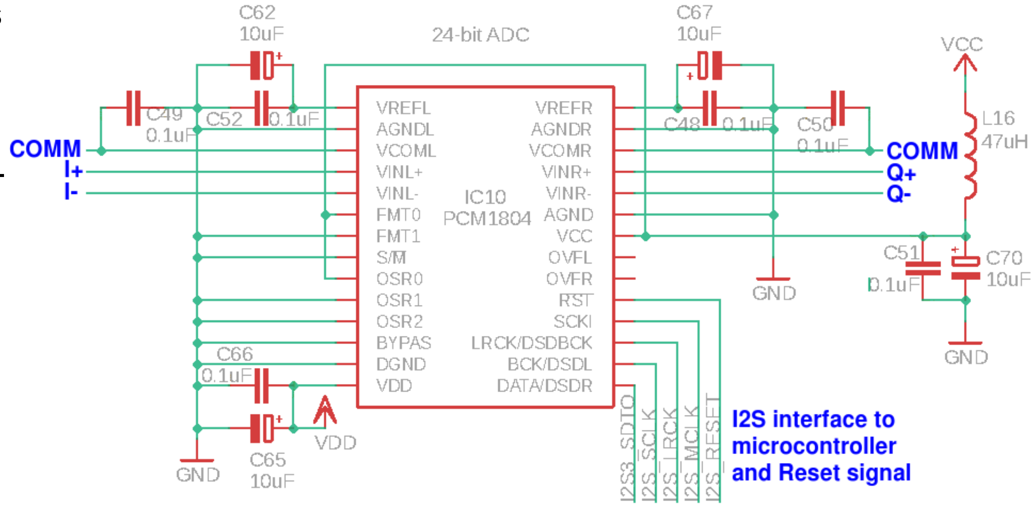
The schematic fragment (above right) shows one of the differential amplifiers used. It has a differential input (the 0-degrees and 180-degrees outputs from the QSD) and a differential output (I+ and I-) to the ADC. The “Common” output of the ADC chip is a 2.5V DC bias level which is used as the DC reference level for the differential amplifier circuit. The circuit is part of an instrumentation amplifier circuit; in an instrumentation amplifier the gain is set by the feedback resistors (in this case R26, R27) and the resistor R11 which couples the two halves. In a real instrumentation amplifier, there is a third op-amp that subtracts the two amplified outputs to produce the difference signal. However in this case, since we still want differential outputs, that final stage is omitted. The instrumentation amplifier has several useful advantages here:

- It has a high input impedance that is independent of the source impedance presented by the QSD and all the other circuits between antenna and QSD.
- Both inputs have the same, high input impedance so there is no imbalance in the load on the QSD integrating capacitors. The only effective load on these sampling capacitors is the source impedance through the BPF etc.
- The gain is determined by R26, R27 and R11. There is no input resistor that forms part of the gain calculation, which would make the gain dependent on the unknown, non-fixed source impedance.
- The configuration has a high common mode noise rejection.

C58 is recommended between the differential inputs of the ADC chip, by its datasheet.

The Q-channel amplifier is identical to the I-channel amplifier shown, and feeds into the ADC “Right” channel.

Analog to Digital Converter

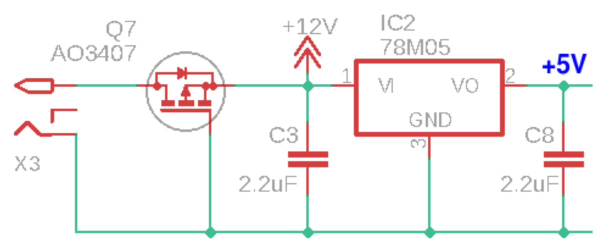


The first and second batches of the QDX transceiver (875 kits in total) used the AK5386 ADC. This chip is a stereo single-ended input 24-bit ADC with 110dB dynamic range. However, the manufacturer suffered a factory fire in Japan that effectively obsoleted this chip. In the global semiconductor crisis we face today, many chips are very hard to find, and expensive when you do. But the AK5386 will never be produced again, and commands ridiculous prices even if you can find small quantities of it. The search for an alternative resulted in PCM1804 by Texas Instruments. It's not easy to find either, but nor is it impossible. The other important criteria to meet or exceed was the dynamic range performance; the PCM1804 datasheet claims a 112dB dynamic range which is sufficient to meet this criteria.

The important difference between the AK5386 and PCM1804 is that the '1804 has a differential input, which required some redesign work in the baseband pre-amplifier/ADC driver circuit (see previous section). In the end the final arrangement significantly outperforms the already excellent performance of the AK5386 first and second QDX batches.

Both AK5386 and PCM1804 have the industry standard I2S audio interface, supported by the STM32F401 microcontroller. No firmware changes were required for the change of ADC chip to PCM1804. In all other regards (tantalum capacitors for decoupling etc.) it is highly advisable to follow the recommendations of the chip manufacturer's datasheet.

Other circuits



QDX is powered at either 9V or 12V (depending on how you chose to build the PA output transformer). Of course the digital circuits require 3.3V or 5V. So we use a 78M05 voltage regulator to provide a regulated +5V rail. Downstream of this is an AMS1117-3.3 low dropout voltage regulator providing a +3.3V rail for STM32 microcontroller, Si5351A synthesizer and ADC chips.

An interesting feature of the power supply circuit (see right) is Q7, a P-channel MOSFET which acts as reverse polarity protection. The addition of a P-channel MOSFET to your project is a very low cost, convenient and simple way to protect you against the inevitable time when a momentary distraction somehow lets you connect the power the wrong way around.

Note that the MOSFET is connected sort-of “backwards”: the Drain is connected to the DC input of the radio, whereas the Source is connected to the downstream circuits. This is absolutely necessary otherwise if the power is reversed, the MOSFET will conduct current through its “body diode”, an internal diode that is created due to the design of a MOSFET. In operation when power is applied the correct way around, current flows through the body diode so both Source and Drain are at +12V potential; the gate is at ground potential so the negative Vgs voltage switches on the MOSFET. If power is applied in reverse, the positive gate voltage keeps the MOSFET switched OFF and no current flows.

The ON-resistance of an AO3407 is around 50-milliohms; even at a 1A current consumption the voltage drop is therefore only 0.05V. It’s a much lower voltage drop than a diode would have. The circuit is therefore a simple and performant means of reverse polarity protection.

If the supply voltage exceeded 20V in either the correct or incorrect directions, then the AO3407 would fail due to exceeding its rated +/-20V Vgs specification. This can be protected against using a series resistor and a 12V (for example) zener diode from the Gate to Source. However, I didn’t bother with this, on the basis that I can’t protect every user against every failure mode, and if you have connected a 20V supply then you’re probably in enough other kinds of trouble anyway.

QDX also has a PTT output, which is configurable as either a positive-going (+5V) or open-drain grounded output.

STM32F401 microcontroller-based SDR receiver

The STM32F401RBT6 processor is a 64-pin QFP 32-bit ARM Cortex M4 microcontroller with 128K Flash memory and 64K RAM, and a useful array of peripherals including USB and I2S. It was chosen for the combination of required resources needed by the project. It has a Floating Point Unit (FPU) and Digital Signal Processing (DSP) instructions making it perfect for this application.

From Wikipedia:

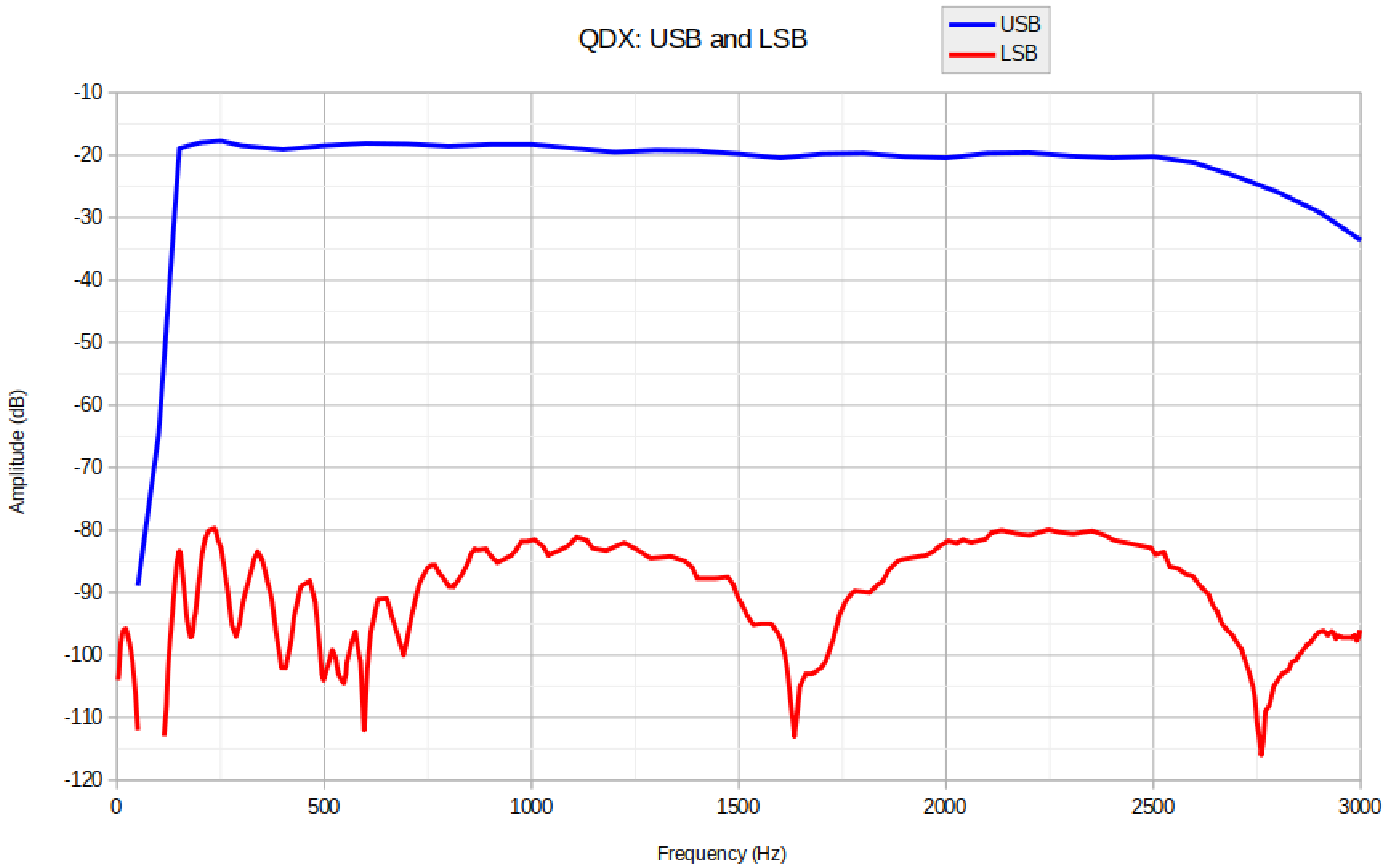
> *“Software-defined radio (SDR) is a radio communication system where components that have been traditionally implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded system.”*

SDR is an enormous topic and I don’t plan to go into details of it here. It was chosen for this project, because it is my believe that a SDR currently offers the best way to optimize performance per dollar, compared to an all-analogue receiver.

QDX implements a superhet receiver with a 12kHz Intermediate Frequency (IF). The reason for a superhet at 12kHz rather than a direct conversion receiver at baseband is that many types of noise, including power line hum, are highest intensity at low frequencies and drop off rapidly with frequency. 12kHz is a convenient choice of IF, being one quarter the 48ksps sampling rate, which makes the mathematics relatively straightforward.

After the 12kHz IF the SDR implements a Hilbert Transform to provide a 90-degree phase shift on one of the I/Q channels. The 90-degree phase shifted audio stream samples are subtracted from the un-shifted channel to give Lower Sideband (LSB), or added to give Upper Sideband (USB). USB is the convention for digital modes, regardless of band, and is the default setting of QDX though it is a configurable option. Finally a filter is applied to provide a final audio band covering 150Hz to 3.2kHz which is a good audio bandwidth for digital modes.

In the current implementation there is no attempt to correct for any amplitude imbalance between the baseband I & Q channels, or phase shift imbalances. Such errors arise naturally due to component tolerances. The unwanted sideband cancellation and audio response was measured, see below.



Despite no attempt to correct for errors, the unwanted sideband suppression is still an impressive -60dB in this example. There will naturally be some variation from unit to unit.

One of the beautiful advantages of SDR is that since large parts of the receiver are software defined, changing the receiver implementation to provide upgraded functionality is only a matter of a firmware update. So in future, I may decide to create an automatic channel amplitude and phase imbalance compensation algorithm that further improves the unwanted sideband cancellation. Many things are possible.

QDX can also be set in “IQ Mode”, in which the raw sample streams from the ADC are delivered straight to the host PC over the USB Audio interface without any kind of processing or demodulation at all. This mode is suitable for use with SDR receivers implemented in software on the host PC, if that is what you wish to use.

QDX has an 8K EEPROM (24C64) where configuration settings are stored.

Finally, the microcontroller has a single status LED on the front panel that indicates the transmit/receive status, or firmware update mode; it can also be used for diagnosing audio level faults.

QRP Labs Firmware Update

Another innovation I have introduced is a very easy firmware update procedure. The QDX can be put into the firmware update mode simply by cycling the power during the first 5 seconds after applying DC power. In firmware update mode, the status LED flashes steadily and slowly.

QDX then pretends to be a USB Flash Drive containing two files, one is the program file (firmware), and the other is the contents of the EEPROM (configuration settings). Firmware updates are published on the QRP Labs website. Installing the firmware update is simply a matter of downloading the file and copying it into the QDX “USB Flash Drive”. Similarly, the configuration file (EEPROM) can be copied (backed up) or written, just by copying from or two the QDX “USB Flash Drive”.

Additionally the firmware implements 256-bit AES encryption. QDX firmware is not open source and this is an important step to protect the commercial viability of the project. Open Source or not, is another huge debate and topic that we’d best not stray into.

The advantages of this firmware update procedure are that it runs on any operating system (Windows, Mac, Linux) because every operating system already contains drivers for USB Flash Drives. It requires no additional programming hardware, everything is done by QDX itself communicating with the PC via the existing USB cable. No software or drivers at all need to be installed on the host PC.

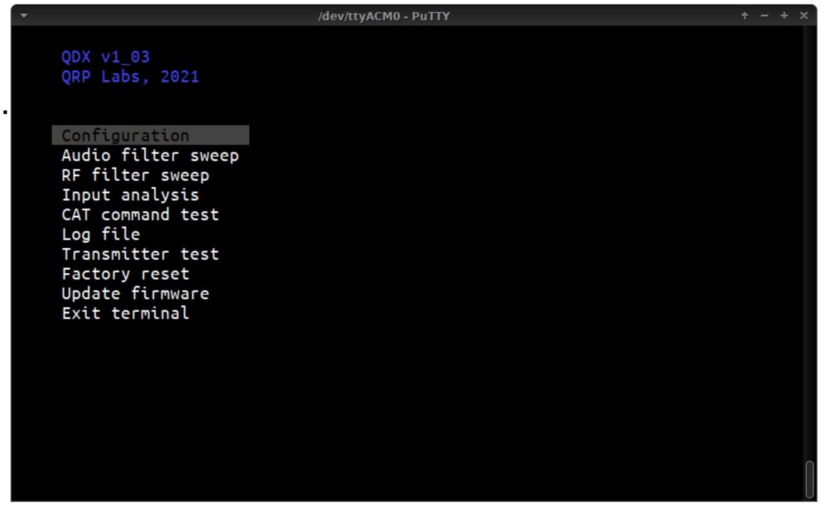
Contrast this with firmware update utilities that are provided by other radio manufacturers, where you have to install some PC software that lets you install the update to the radio, and maybe runs only one one operating system (Windows); or insert an SD-card into the radio with the firmware update copied onto it. This QDX firmware update procedure is ridiculously easy!

QDX configuration and terminal utilities

To configure QDX, experiment with different settings, optimize the band pass filter, debug problems, and LEARN, I provided a set of terminal applications which is accessible via any PC-based terminal emulator, communicating with QDX over the Virtual COM serial port.

Again the advantages here are that anyone can use their favourite terminal emulator (if they have one), or choose from any number of freely available terminal emulators which will all do the job. I use PuTTY which is available for Windows and Linux. I don’t have to write any PC software, and terminal emulators are available for any PC operating system. A disadvantage is that everything in the terminal emulator is text-based (ASCII) so fancy graphics are out of the question.

On connecting the terminal emulator to the QDX serial port, pressing the Enter key enters the terminal applications menu (right).

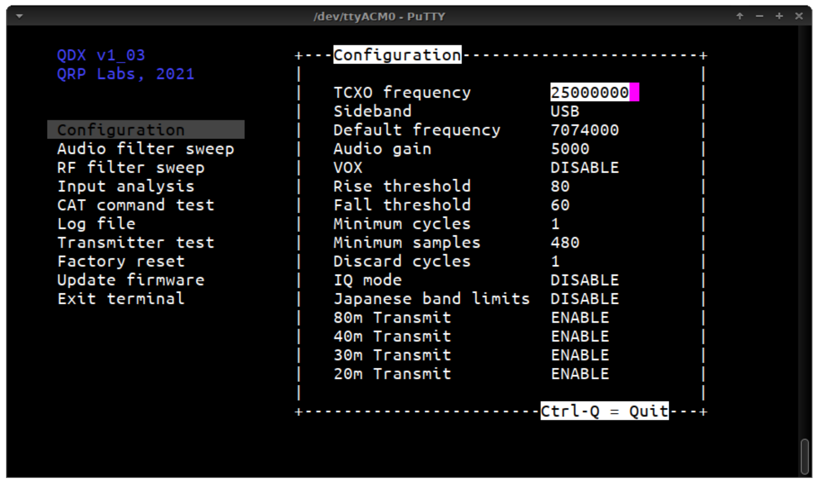


The firmware version is shown, and you can select one of the options using the up and down arrow keys then pressing enter.

A “factory reset” is available which returns the QDX to default factory configuration settings.

You can also initiate a firmware update from this initial menu.

Configuration menu

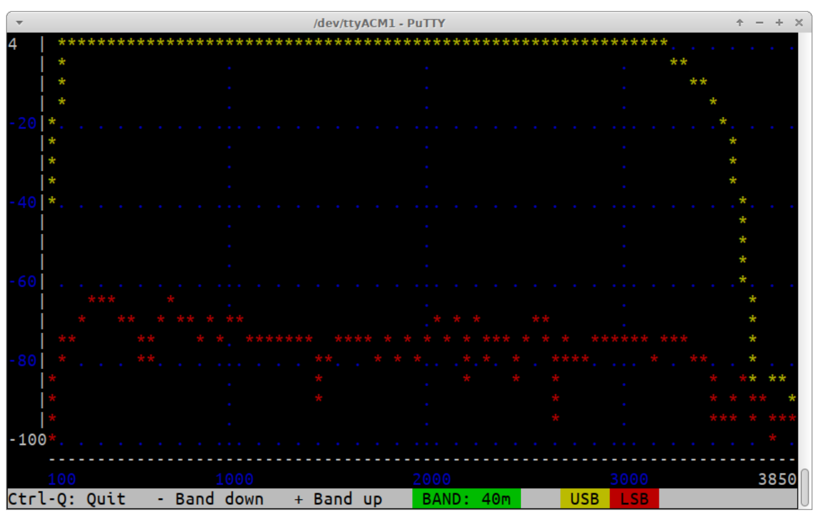


The configuration menu lets you customize the behaviour of the QDX transceiver, including calibration of the synthesizer reference frequency, sideband, audio gain, and other parameters.

Japanese band limits mode forces QDX to only transmit within Japanese band limits. This is a regulatory requirement in Japan and South Korea.

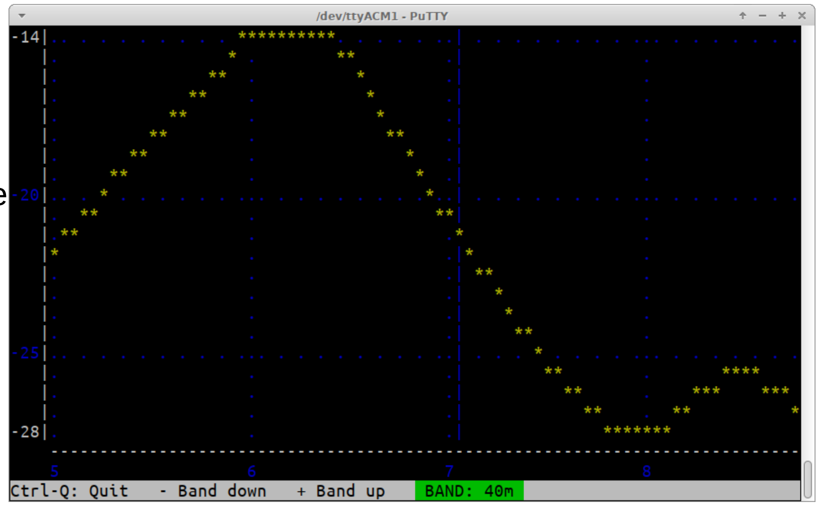
Transmit can also be enabled by band, this could be useful to prevent accidental transmission into high SWR on bands your antenna isn’t optimized for.

Audio filter sweep



Here the Clk2 output of the Si5351A is used to inject a signal into the front end of the QDX receiver, and sweep it across +/- 3850Hz from the dial frequency. You can run this on any of the four bands. It shows the audio response of the wanted sideband (USB) and the unwanted sideband (LSB) as yellow and red “lines” respectively; though as I mentioned, you don’t get graphics, and have to get used to ASCII art. Nevertheless it’s a nice demonstration and validation that the receiver is working correctly, and if you compare this to the performance graph I presented earlier, you will note the similarity.

RF filter sweep



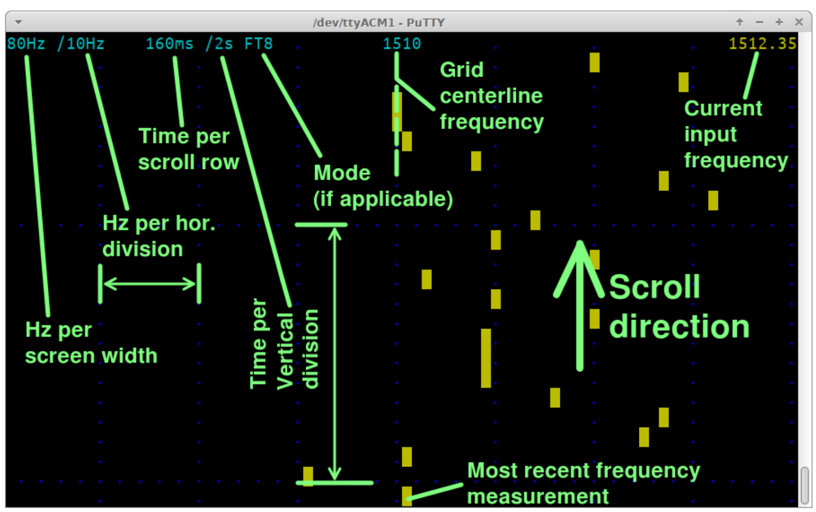
The RF filter sweep moves the receiver's operating frequency across a range and injects a signal using the Si5351A Clk2 output. In this example on the 40m band, the frequency is swept from 5 to 9MHz. The 7MHz (40m) band is indicated by the blue vertical line and the response, by the yellow line.

You can see that in this example, the 40m band pass filter is peaked incorrectly on about 6.2MHz and is around 8dB down on the intended 40m operating frequency.

For the perfectionist, the receiver band pass filters can be adjusted by squeezing or spreading (or even removing or adding) turns on the band pass filter inductor toroid L12.

Input analysis

This one is lots of fun. If a digi mode software package such as WSJT-X is used to send audio over the USB link to QDX, this terminal screen shows the actual decoded (measured) audio frequency in the top right hand corner. It's a poor man's waterfall display but despite the obvious limitations of an ASCII-art waterfall display, it's quite capable of giving you a fascinating insight into the crucial operations of the QDX audio measurement which is right at the heart of the whole principle of QDX.



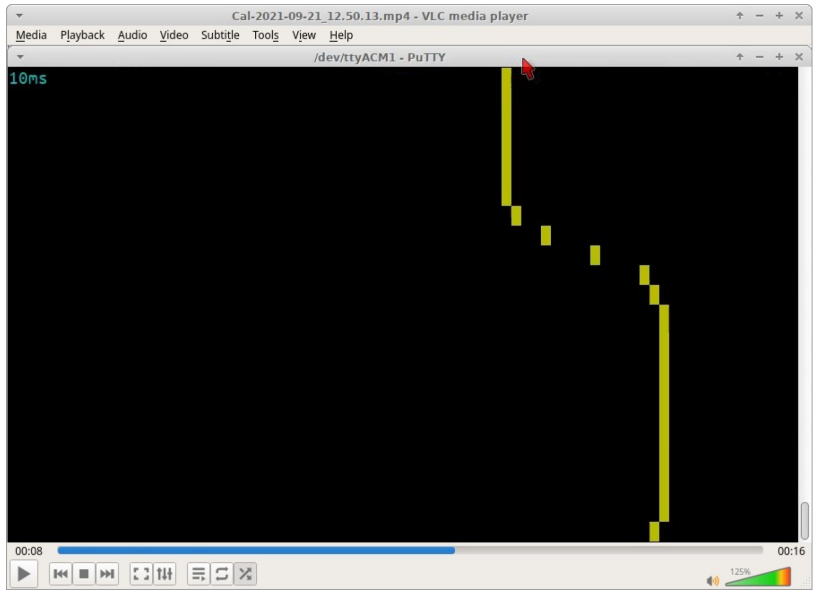
The annotated image (right) explains all the elements of the display.

The scroll rate and screen bandwidth are all configurable and can be changed by pressing the keyboard arrow keys. Settings are provided to scroll though, that suit various common digital modes.

This example shows FT8, the screen scrolls every 160ms which is the FT8 symbol rate, and the screen bandwidth is 80Hz which is sufficient to show all the tone symbols of FT8's 50Hz bandwidth.

Another interesting effect can be noted when the scroll rate is sped up to 100 lines per second (each row is 10ms).

This demonstrates the way the WSJT-X output frequency changes from one tone symbol to another; in this case we a shift of 3 symbols (18.75Hz). WSJT-X slides the frequency from one tone to the next, in a kind of raised-cosine shape to minimize bandwidth. Sudden frequency changes would cause splatter into adjacent channels. QDX also outputs the same sliding frequency by making audio cycle measurements and synthesizer frequency updates fast enough.



CAT test

Another terminal screen allows the operator to type in raw CAT commands directly to test their validity and effect. For example, "FA;" is the command to query the VFO A frequency. The full list of available CAT commands is in the manual.

Error log

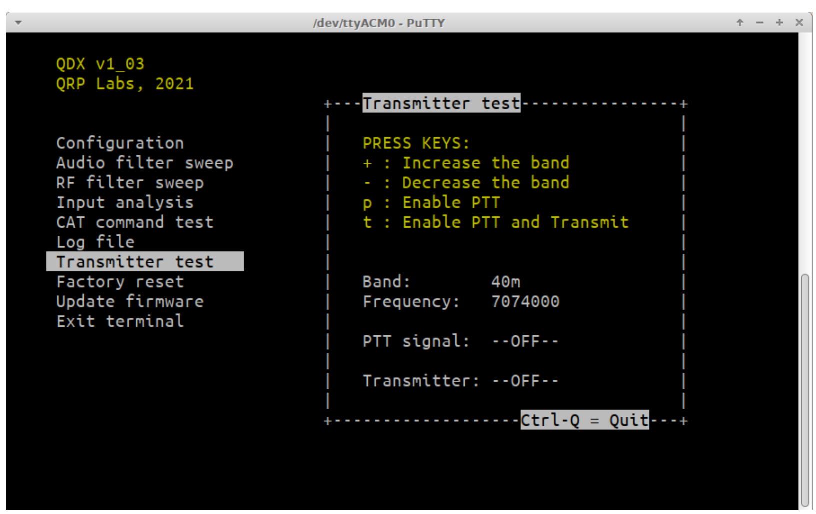
There's an error log file stored on the 8K EEPROM, and a terminal screen to configure and view it. The log file can record all events (including CAT commands) or just error events. This can be useful to debug problems with CAT commands, for example if a new digi mode software application uses unsupported CAT commands, this log file would be able to help find the unsupported commands, so that a firmware change could be made to support them.

Transmitter test

The transmitter test screen is particularly useful as it allows keying either the PTT line (with no key-down), or the PTT line AND key-down.

You can select the band by using the + and – keys.

This terminal screen can be used to verify correct operation of the low pass filter switching and PA, and debug any assembly issues in these areas.



Conclusion

I hope this article has generated some ideas for you to use in your own homebrew projects.

I was very pleased with the outcome of the project, the performance is excellent and there are so many features to explore and enjoy, yet the cost was kept low ($65 for the kit, and $20 for the enclosure).

More information and to purchase QDX:

http://qrp-labs.com/qdx

Happy homebrewing and happy QRP!