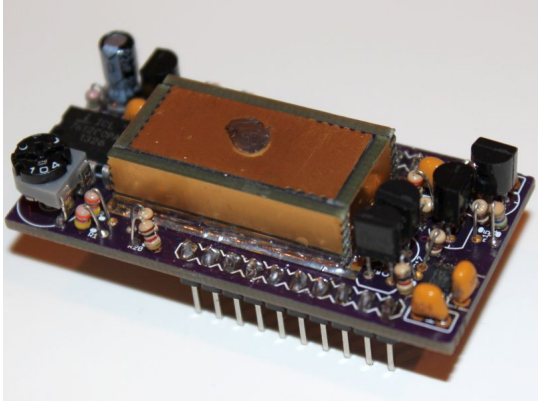

This page documents the design and development of an OCXO and Synthesiser module. There were s

This module is available as a kit from http://www.qrp-labs.com . The AD9850 DDS module is pictured below left. My final module is pictured below centre and right.
Click on the photos to see larger versions
.

{gallery}ocxosynth/1{/gallery}

The AD9850 DDS module is a very useful module but it has limitations. The module is used in the QRP Labs Ultimate3 kit and allows operation on any HF band from 2200m up to 10m. However, getting reliable drift free operation at 10m is slightly challenging. It can be done, but it requires adjustment and patience. People had enquired about operating higher up on the 6m band. A modification for 6m can be done, see here . It allows the comparator to work up over 50MHz and so the U3 kit can operate on 6m. However the spectral purity is poor. This is to be expected, considering the output frequency of the DDS is a large proportion of the 125MHz reference frequency. Filtering cannot remove the close-in DDS spurs.

Another concern about the AD9850 module is drift. Not so important on the lower HF bands, but the higher up the bands you go, the more the drift worsens. The U3 kit has optional GPS drift correction but this operates only between transmissions. If the drift of the output signal is more than a few Hz (3 or 4Hz), then WSPR decodes are not possible. With careful adjustment and use of the U3 kit's Park mode parameters, it is possible to remove the drift even on 10m, but it is not trivial and the results vary from kit to kit.

Finally, availability. The AD9850 DDS modules were available on eBay for under $5, at one time, even down to $4. However prices are now rising and this may be because the modules

are running out. The module price is only a small fraction of the price of the DDS chip alone, so it is possible that the modules are built from some leftover stock, and that may run out.

Hence the desire to develop a replacement module, that may operate in the U3 kit , guarantee future supply, and enhance its capabilities. The Si5351A synthesiser chip is used in this project. It can produce up to three outputs from 8kHz to 160MHz. It's a related chip to the famous (in amateur radio circles) Si570 chip. However, it does not have an onboard crystal oscillator as the Si570 does. This is an opportunity here, because it allows construction of an external ovenised oscillator. Additionally the Si570 is very pricey.

Oven Controlled Crystal Oscillator (OCXO)

So let's start with the OCXO. These are available commercially but expect to pay many $$ for one. Less for second hand ones but that is not useful when we need a supply of hundreds for a kit run. Therefore, let's build our own. The performance may not be as good as a commercial one but it is possible to homebrew a good OCXO, plenty good enough for the requirements here.

The performance of an OCXO is dictated by several things, including how good the thermal characteristics of the oven are, and how good the control circuit. The quality of the crystal is important too, but in this case we're using just an ordinary common AT-cut crystal, to keep the costs down.

The photo below centre shows my oven construction. I believed it to be important to keep the oven as physically small as possible. This will minimise the time taken for the oven to reach thermal equilibrium, as well as reduce the amount of power required to heat the oven. To make the OCXO easily reproducible and inexpensive, I didn't want to include large thermal heatsinks inside the oven, or lots of insulation around it. So it's even more important to keep it small and simple, so the performance can still be reasonable. Yet, the oven should contain more than just the crystal and the heater. The other components of the oscillator are also temperature sensitive. So this oven includes the heater, crystal, temperature sensor, Colpitts oscillator transistor and associated components.

{gallery}ocxosynth/2{/gallery}

The control circuit is the tricky part. In the simplest kind of control circuit, the measured temperature is just compared with the desired target temperature. Then the heater is switched on if the oven is too cold, and off if the oven gets too hot. This kind of controller is common (getting les common) in the mechanical room thermostat, or the thermostat in your kitchen oven or fridge, or the air conditioner in your home or office. The disadvantage is that the thermal mass of the oven takes time to heat and cool, and this means the temperature can vary quite considerably as the heater (or cooler) cycles on and off.

The right photograph above, shows the crystal frequency (received on an HF receiver and plotted in Argo spectrum analysis software on a laptop). The cycles of the heater on/off are very clear here. The cycle duration was around 38 seconds, of which the oven was switched on for 17 seconds and off for 21 seconds. This kind of temperature cycling would be completely unacceptable for QRSS or WSPR operation such as in the U3 kit .

So we come to proportional oven control. There are several good examples of homebrewed ovenised oscillators on the web, but two of my favourites are: Andy G4OEP, about half way down this page and Des M0AYF's extensive page on his QRSS ovens . These two are my inspiration. Both of these ovens use a proportional controller. There's a heater, a temperature sensor, and a control circuit consisting of an op-amp with relatively low gain (not an on/off comparator). The limited gain of the op-amp makes it possible for the heater to be partially on. If the thermal characteristics are right, and the gain is correctly matched to them, then the proportional control circuit makes it possible to control the frequency without the characteristic "hunting" cycles of an on/off oven.

However, I wasn't quite comfortable with proportional control circuits. An ideal control circuit would maintain the oven at a constant temperature regardless of the ambient, environmental temperature. But a proportional circuit does not. The oven is kept at a reasonably constant temperature but there is still some variation, as the environmental temperature varies. Even if everything else is perfect, the circuit inherently produces an error term. To stop "hunting" oscillations, the amplifier gain needs to be low, but the lower the gain, the larger the error term becomes.

So to PID controllers (Proportional Integral Derivative) which are used in industrial automation. Surely the subject of many text books and the nightmare of Electronic Engineering students everywhere. There is plenty to read on this topic, and it seems to quickly get complicated enough. Software programs are nowadays used for the control logic. But here, we use a simple op-amp. The quick summary (after an awful lot of reading and research): it turns out that the

"Derivative" term isn't necessary in a simple temperature controller. The "Proportional" term we have covered already, such as in the circuits above. Best of all, the "Integral" term can be implemented simply by putting a capacitor in the feedback loop of the op-amp; and the "Integral" term removes the error that is inherent in "Proportional" only circuits. Simple and neat, and does everything we need!

So the circuit diagram of my first Oven, is shown in the left of those three images above, and shown again right here below.

{gallery}ocxosynth/2{/gallery}

I used a 10MHz crystal in this development oven. The heater is a single BS170 MOSFET, a device with a maximum power dissipation of 830mW. To save space and keep things simple, the BS170 is connected directly across the 5V supply so it is the entire heating element (no resistors). 830mW is enough to heat this oven from room temperature to the set point, but in the final version I used two BS170 to widen the operating range. The temperature sensor is a BC547 transistor. A 10K trimmer potentiometer sets the operating point (temperature) of the oven. The arrangement is actually a bridge circuit that should theoretically make it relatively insensitive to supply voltage variation; and the less sensitivity you have to anything, surely the better. The use of a TO92 transistor as heater, and another as sensor, also makes a nice easy compact mechanical construction with the crystal (HC49/4H low profile 10MHz crystal) sandwiched between the two, as shown on the diagram. The oscillator is a usual Colpitts followed by another buffer stage. The "oven" was just made from PCB stock cut and soldered to the required box shape.

The behaviour of the oven is shown using the frequency plots below, with different size of capacitor in the feedback loop. From left to right: 100uF, 220uF, 33uF and 47uF. In all the graphs, the vertical lines are spaced at 1 minute intervals. It can be seen that the 33uF capacitor is too small, the oven temperature (indicated by the plotted frequency) is oscillating slowly. The 47uF capacitor also shows oscillations but they die out gradually over time. The 100uF capacitor is pretty nice, in the initial aggressive heating of the oven from cold, there is some overshoot but everything stabilises fast. To allow some safety margin, 220uF seems a good value. It takes longer for the oven to settle but in practice temperature variation is not likely to be extreme and we don't need a very fast initial readiness from switch-on with a cold oven.

{gallery}ocxosynth/3{/gallery}

That's all very encouraging, so now to turn this into an actual kit product, needs a PCB. The PCB design is shown below left.

{gallery}ocxosynth/4{/gallery}

The PCB is about 4 x 3 inches and it is designed to be broken up into 16 smaller PCB's. The main PCB for constructing the circuit is a little larger than the AD9850 DDS module, but it still fits comfortably in the U3 kit . The other 15 PCB's make up three 5-sided boxes, the 6'th side of each, is the main circuit PCB. The three boxes are the top and bottom half of the oven enclosure itself, and then a large box to go over the entire module and give an extra level of thermal insulation. The PCB box design is carefully done so that they have copper covering inside and out, which are insulated from each other, where the "insulation" is the fibreglass of the PCB itself. It's simple but elegant, and it works well.

The circuit diagram (upper right) is similar to the development oven, this time however two BS170 are used as the heater element, in order to leave some safety margin and also allow operation in cold weather without risking burning up a single BS170. In the circuit, the resistors around the op-amp are scaled up by 10, and the capacitor is scaled down by a factor of 10. The main reason for this was physical size of the capacitor, which could not be fit inside the box otherwise. To make the module compatible with the U3 kit the dimensions of the board are strictly limited, and the height of the components above the board cannot exceed 8.4mm. Scaling the components allowed the use of a 22uF capacitor which fits in the available space.

The circuit includes two LM317Z voltage regulators, to provide some isolation against external 5V supply voltage variation, and because the Si5351A is a 3.3V device. The circuit also includes two I2C level converters, to convert between 3.3V at the Si5351A and 5V in the control circuit, assumed to be the U3 kit 's AVR processor.

Now the first results are shown below, these use a 47uF tantalum capacitor in the feedback loop, because Tantalum's have a much lower leakage (higher leakage resistance) compared to standard electrolytic capacitors, and I worried that leakage resistance would impair the purity of the PI control circuit and increase the control error. In these charts the OCXO has a 25MHz crystal as reference for the Si5351A which is set for 10MHz (actually a little over 10MHz). The Si5351A output is beat against my 10MHz frequency reference which actually doesn't have a

GPS connected at the moment, but the important thing is that it is highly stable with a good commercial 10MHz OCXO inside. The "beating" mixer is just two diodes, pretty simple. The audio is fed into the laptop and processed in Argo.

{gallery}ocxosynth/5{/gallery}

The centre chart shows a frequency vs ambient temperature plot over the range 0 to 30C. This was obtained with the use of the kitchen freezer and a large block of polystyrene, with temperature measurement in an Altoids tin using an TMP36 temperature sensor. The temperature sweep took about 1.5 hours and it can be reasonably assumed that the TMP36 temperature reading was representative of a state of thermal equilibrium within the tin, or at least to a reasonable approximation. The frequency change is 100Hz with the oven switched off. With the oven on, and the temperature set to something just over 40C, the frequency variation is only 3Hz over the same temperature range.

The right chart shows the frequency change from switch-on, as the oven heats up from room temperature to its operating set point; on the same chart the heater current (right axis). You can see that an initial surge of current heats the oven then it settles down to a steady value.

On the left, a chart of heater current vs temperature. As you'd expect, the colder ambient temperature requires a greater heater current to heat the oven. The heat loss is supposed to be proportional to temperature, which I remember from my student days. So we expect, and obtain, a nice straight line. Extrapolation of the straight line to the X-axis would show us the set temperature of the oven.

There followed after this, a great many experiments and long slow patient temperature runs in the freezer. The laptop power supply gets hot and was used for the initial heating to 30C, before putting the setup in the polystyrene and the kitchen freezer. Adjustments to the oven operating temperature were tried. Different capacitor values were tried.

A 2.2M resistor was placed in parallel with the capacitor, to simulate an increase in leakage, probably an unrealistically high capacitor leakage, in fact. Interestingly, the 2.2M leakage did make the oven performance much worse (making it more like the Proportional-only controller), but in the error was in the opposite direction to the error that seems to occur even with the

tantalum capacitor. The conclusion is that something else is causing the error. But this is not really a surprise - many parts of the circuit are outside the oven and who knows what their effect is - including the voltage regulators, the resistors associated with them, the op-amp, and the oscillator buffer. So it appears that a little leakage would not be a bad thing, there is some cancellation of errors; which led me back to try the ordinary 22uF electrolytic capacitor. This gave the best result that I was able to measure, see charts below.

{gallery}ocxosynth/6{/gallery}

The above left chart shows the frequency variation for a temperature change from 0 to 30C. The "zero frequency" point is taken to be the frequency at 25C, as an arbitrary reference. The straight line between 6C and 14C is not because there are no measurements in that interval: there are, measurements were made every 1 minute. But the measurements really are almost all the same, hence the straight line! So between about 3C and 28C, the frequency variation is all within a 0.1Hz band. This is a band only 0.01 parts per million (ppm) wide. Really a very nice and good result, considering the simplicity of the oven circuit.

The left hand image shows the measured frequency shift characteristic of the 25MHz crystal, vs temperature. The range is -6C to +62C, which was the highest temperature I dared take the OCXO oven up to without getting uncomfortably close to the BS170's power dissipation rating. The graph shows the expected cubic curve, with turning points just below 0C and just above 40C. According to what can be gleaned from the internet, this is entirely consistent with common cheap AT-cut crystals.

The above right chart shows the current consumption (left axis) and in power consumption terms (right axis) of the oven heater, vs temperature. The range covered is -8C to +32C. A best-fit straight line is also shown on this chart, and this can be extrapolated to calculate the oven set's temperature, which turns out to be 45C.

Overall the OCXO produces an impressive result, considering the simplicity of the construction and the controller circuit. It was found to be stable at all tested temperatures from -8C to +32C. The heater power consumption is just over 1W at -8C which is well within the allowable dissipation of the two parallel BS170's which can dissipate 830mW each. For more typical ambient temperatures the heater power consumption should be less than half a watt. It should be noted that the rest of the module circuit (op-amp, temperature sensor, voltage regulators, Si5351A) consume about 27mA of current at the 5V operating voltage, in addition to the current

consumption of the oven itself.

Si5351A synthesiser

Now that's the oven dealt with. The SI5351A side of things is very interesting too. This is a very neat chip, a digital PLL synthesiser, like the Si570, with three independent (almost) outputs covering 8kHz to 160MHz. The disadvantage is the tiny size, just 3 x 3mm, and its 10 pins have a spacing of only 0.5mm. Rather hard to solder.

{gallery}ocxosynth/7{/gallery}

Above, here's my initial development board. It was built "ugly style" on a piece of copper PCB, which was fixed on top of a dead AD9850 DDS board that was used just for the purposes of a convenient substrate that would plug nicely into the U3 kit . Soldering 0.5mm-spaced pins on that tiny chip is not easy. The wires uses tiny individual strands from multi-core hookup wire. In this development circuit, the 27MHz reference crystal is connected directly to the Si5351A, rather than using an external oscillator as in my final version. I felt that the external oscillator would be an advantage because it would allow buffering between it and the Si5351A, which would further reduce any effects of variation of loading the Si5351A etc. when in an application circuit such as the U3 kit .

Now, see below, some oscilloscope photos showing the output waveform of the Si5351A (far left) compared with the AD9850 DDS (next left) at 10.140MHz. The main noticeable difference is the amplitude, which is 5V for the DDS and 3.3V for the Si5351A.The right two photos show the 27MHz signal at each of the crystal terminals.

{gallery}ocxosynth/8{/gallery}

Comparison with AD9850 DDS Module

Click here to see an extensive set of measurements. The measurements show the Si5351A Spectrum analyser output at various frequencies (always on the left, in the PDF document) and the AD9850 DDS output on the right, at various frequencies. The Si5351A in all cases produces less spurious outputs than the AD9850. The 6m comparison is particularly interesting. Visible on the Si5351A plot, are peaks corresponding to the 20MHz system oscillator in the

U3 kit
and the 27MHz Si5351A reference oscillator. But certainly the Si5351A is cleaner than the AD9850 module so it will be a good improvement.

The comparison PDF document also shows a comparison of the two generators, with the U3 kit
sending "G0UPL" in FSKCW. Again, the Si5351A looks at least as good as the AD9850 DDS.

OCXO/Synthesiser module test in a U3 kit

The Si5351A worked (see above), and the OCXO was stable and nice, when tested individually. The critical test was, would the module work properly and well in actual use in a U3 kit ? Or would the RF, and the heat from the PA transistor(s), cause the frequency to drift on key-down? The tests were carried out with a standard configuration
U3 kit
, single BS170 PA transistor at 5V and without any heatsink, operating at 10MHz into a 47-ohm resistor connected at the board output, as dummy load.

Well, initially I found that there was a drift of about 0.4Hz (at 10MHz output) during a 2-minute key-down. However, there is about 60cm of thin hookup wire between my power supply and the U
3 kit
. I wondered if the extra load of the PA during key-down would cause the voltage to droop a bit, altering the heating of the oven. I used a different power supply (13.8V @ 5V switched mode supply, followed by 7805 +5V voltage regulator) and some thicker wire. The drift in 2-minutes keydown dropped to under 0.05Hz. So indeed, power supply loading is to blame. A wire does not have zero resistance. Furthermore I then connected the PA to one power supply, and the rest of the
U3 kit
to the
other power supply
, to totally isolate them. The drift in 2-minute keydown is now unmeasurable. If there is any, it is less than the 0.01Hz resolution of Argo's peak-frequency reading.

{gallery}ocxosynth/9{/gallery}

Above left: here's the Argo chart, of 2-minute key-down, followed by 2-minutes key-up. A stressful situation indeed for a normal U3 kit with AD9850 DDS. The drift is zero. This means that the OCXO/Synthesiser module is performing really well, and it should allow operation of U3 kit on 6m and hopefully even 4m and 2m.

Above centre: this plot shows the frequency change from cold switch-on. It can be seen that several minutes are required for the system to stabilise (reach thermal equilibirum). Again, 10MHz output was used for convenience of beating against my 10MHz reference . However, the frequency has stabilised to less than 1Hz of the final value, within 2-3 minutes. This is 0.1ppm (parts per million).

Above right: here's a photo showing the new prototype OCXO/synth module sitting comfortably inside U3 kit . Everything was carefully dimensioned so that the lower oven compartment doesn't interfere with the U3 kit 's AVR processor, and there is still space for the LPF module at the left; the height of the components over the board was kept to under 8.4mm so that with the 1.6mm standard PCB thickness of the external box, the overall height is 10mm above the module board, leaving several mm of headroom if you want to install the relay-switched LPF kit .

In conclusion

This OCXO project taught me a huge amount, and its performance surpasses my expectation. In the U3 kit I think it will allow operation on 6m and perhaps even 4m and 2m (to be tested - watch the U3 kit page for announcements). In addition I am sure there will be many more applications for this module.