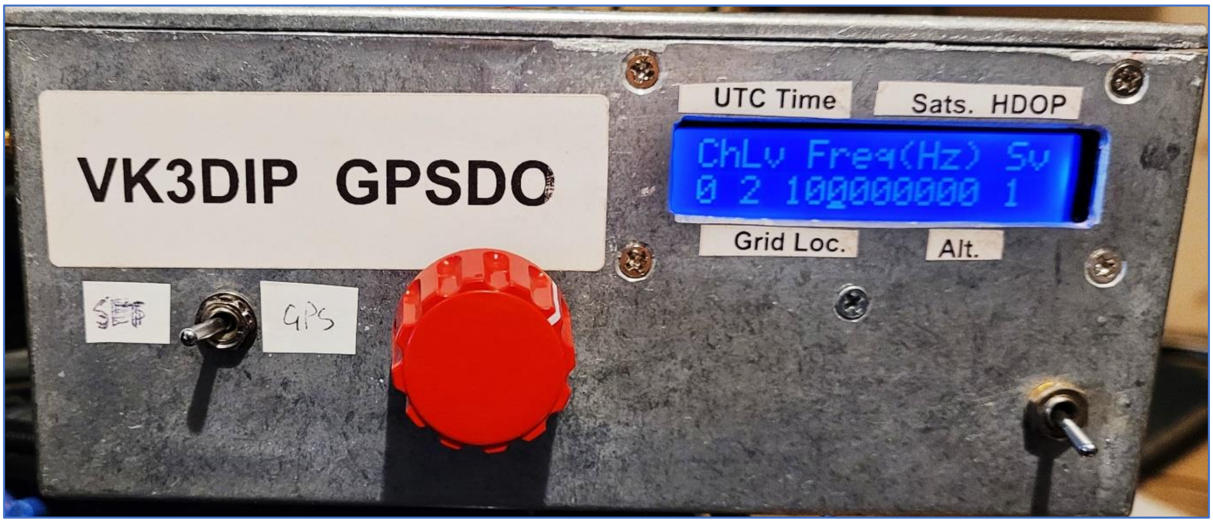
A GPS Disciplined Oscillator

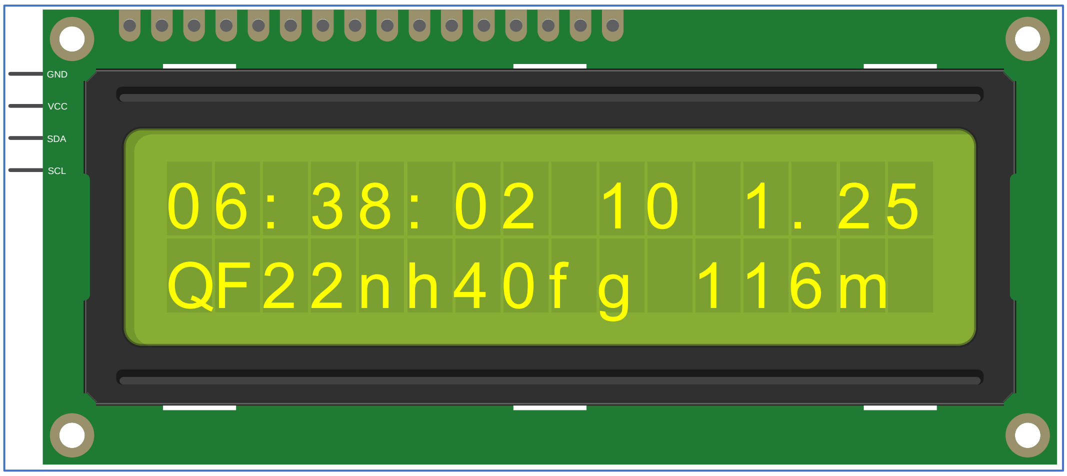
By Paul McMahon VK3DIP

Part 4. Yet More Mods.



10 Character Grid Locator.

The first addition is the adding of the next two digits to the maidenhead grid locator display as shown in Figure 1.



```
06:38:02 10 1.25
QF22nh40fg  116m
```

Figure 1 - 10 Char Grid locator display.

On the top line we still have 8 characters of the UTC time, a space, the number of Satellites used, another space, and the DOP (Dilution Of Precision) which is basically a measure of how good the actual physical arrangement of the satellites is. For DOP as mentioned in part 3 a value less than 1 is considered ideal, 1-2 is excellent, 2-5 is good, and so on. The second line starts with the, now ,10-character location grid, a space, and an altitude value now rounded up to the nearest metre. This altitude even with the lesser precision will still vary a bit especially when the satellites are low on the horizon, so I don't think the decimal points were doing much here. The 10-character grid square however does give quite a bit more accurate idea of your location, and there are some small number of higher band contests etc. where you are asked for this higher precision. This higher precision also brings the possibility of any non-ideal satellite arrangements causing the last two characters to vary (bobble) especially if your antenna happens to be close to a grid boundary. As an example consider the case shown in Figure 1. This is actually a display that might be shown on the display of the GPSDO that is sitting in the shack room at the back righthand corner of the NERG clubrooms in Briar Hill. If we look at this on a map with the grid squares overlaid (Figure 2.) , then we can see that with the GPS antenna positioned on the right hand wall, near the back of the building it is actually quite close to the border of the next grid square ending with ff rather than fg. So even with one metre accuracy of the GPS, bobbling between QF22nh40fg and QF22nh40ff would not be unusual.

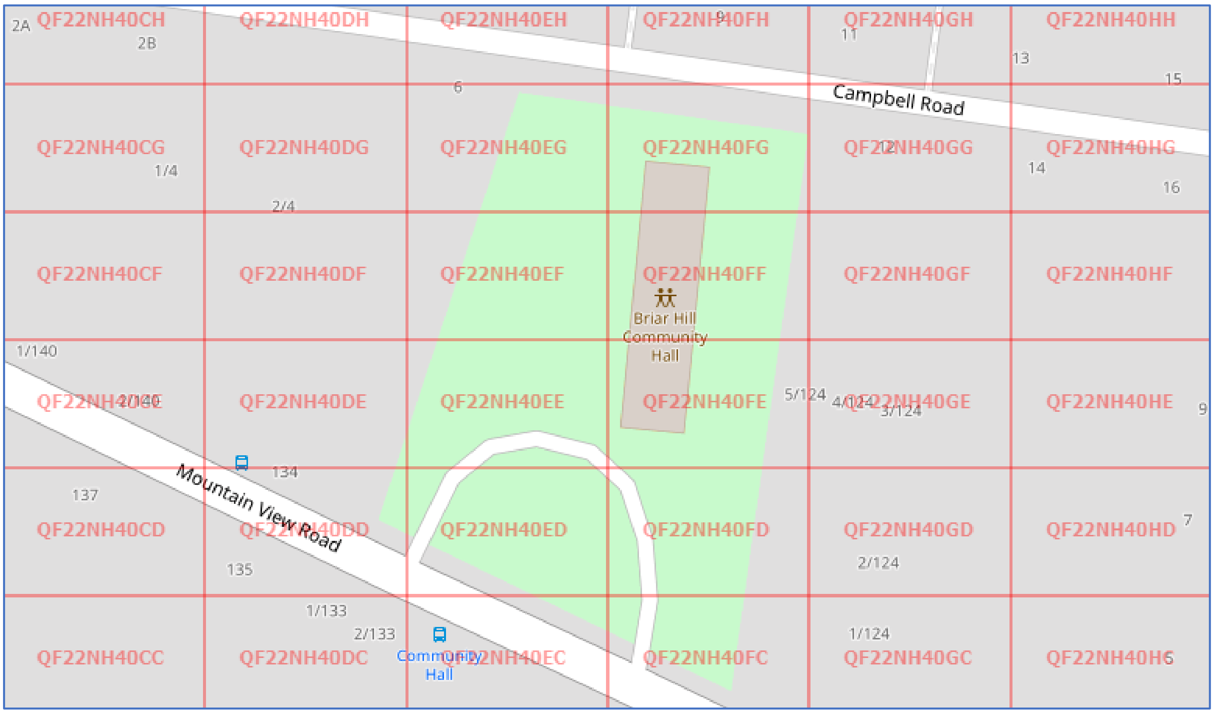


Figure 2 - NERG Clubrooms from https://dxcluster.ha8tks.hu/hamgeocoding/

If you want to update to just this mod then all that is required is a new flash of firmware to a version 2.3 or greater. The V2.3 .INO file can be downloaded from here : http://yagicad.com/Projects/VK3DIPGPSDOV2_3.zip

GPS Locked not quite Triple Signal Generator

In part 3. We discovered that with a few small mods it was possible to use the 10MHz from the GPSDO as the reference frequency for a SI5351A three channel PLL module. Just as a reminder the mods consisted of removing the on-board SM crystal oscillator and feeding in the logic level 10MHz signal where the bottom left crystal pin was via a 100nF cap. See Figure 3.

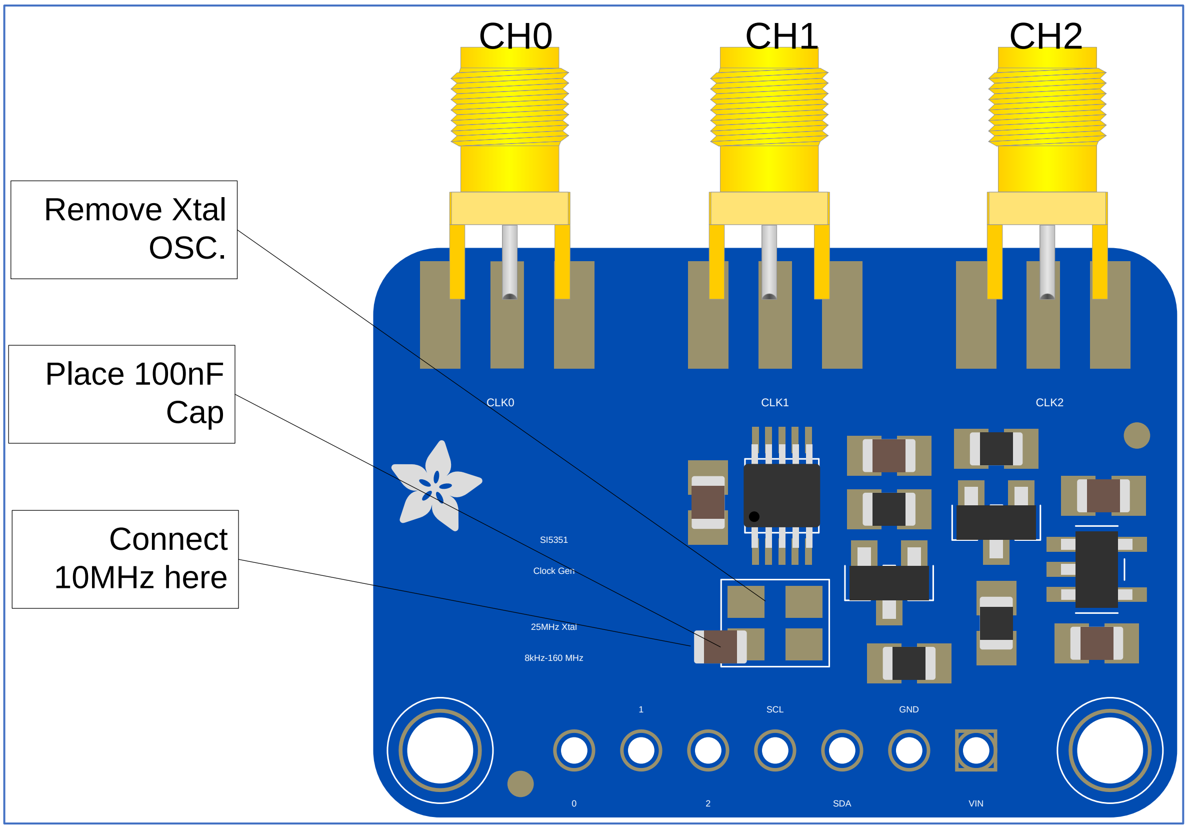
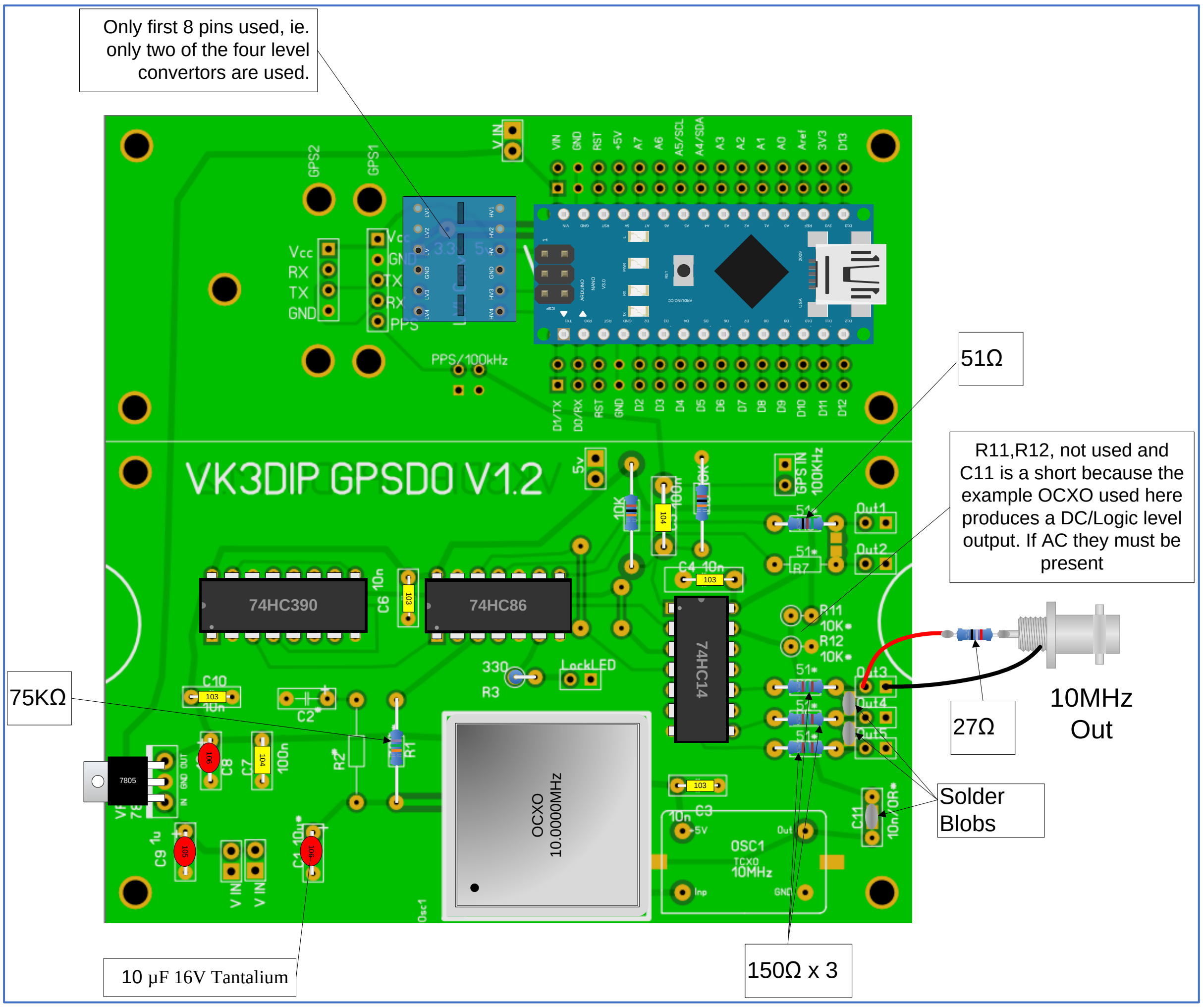


Figure 3 - Modified SI5351A module.

In the older versions the three frequencies for the three channels of the SI5351 were hard coded in the .INO file and you needed to recompile and flash to change them. With the newer firmware (4.2 and up) here: http://yagicad.com/Projects/VK3DIPGPSDOV4_2.zip , and with the addition of some further hardware, ie. a rotary encoder and switch (or jumper), you can change the frequencies and power levels for each of the channels at will, and save them to EEPROM so that they will be remembered at next start up.

As an example and refresher of how to build a version 1.2 PCB using the commonly available Chinese OCXOs which has a logic level output, have a look at Figure 4. Apart from those values indicated values and orientations are as per the silkscreen on the board.

In lieu of a circuit diagram, and in honour of Dave's presentation on Fritzing diagrams some months back , a pseudo Fritzing diagram of the full setup showing how it all connects up is shown in Figure 5.



Only first 8 pins used, ie. only two of the four level convertors are used.

51Ω

R11,R12, not used and C11 is a short because the example OCXO used here produces a DC/Logic level output. If AC they must be present

75KΩ

27Ω

10MHz Out

Solder Blobs

10 µF 16V Tantalium

150Ω x 3

Figure 4 - An example fit-out of a version 1.2 board.

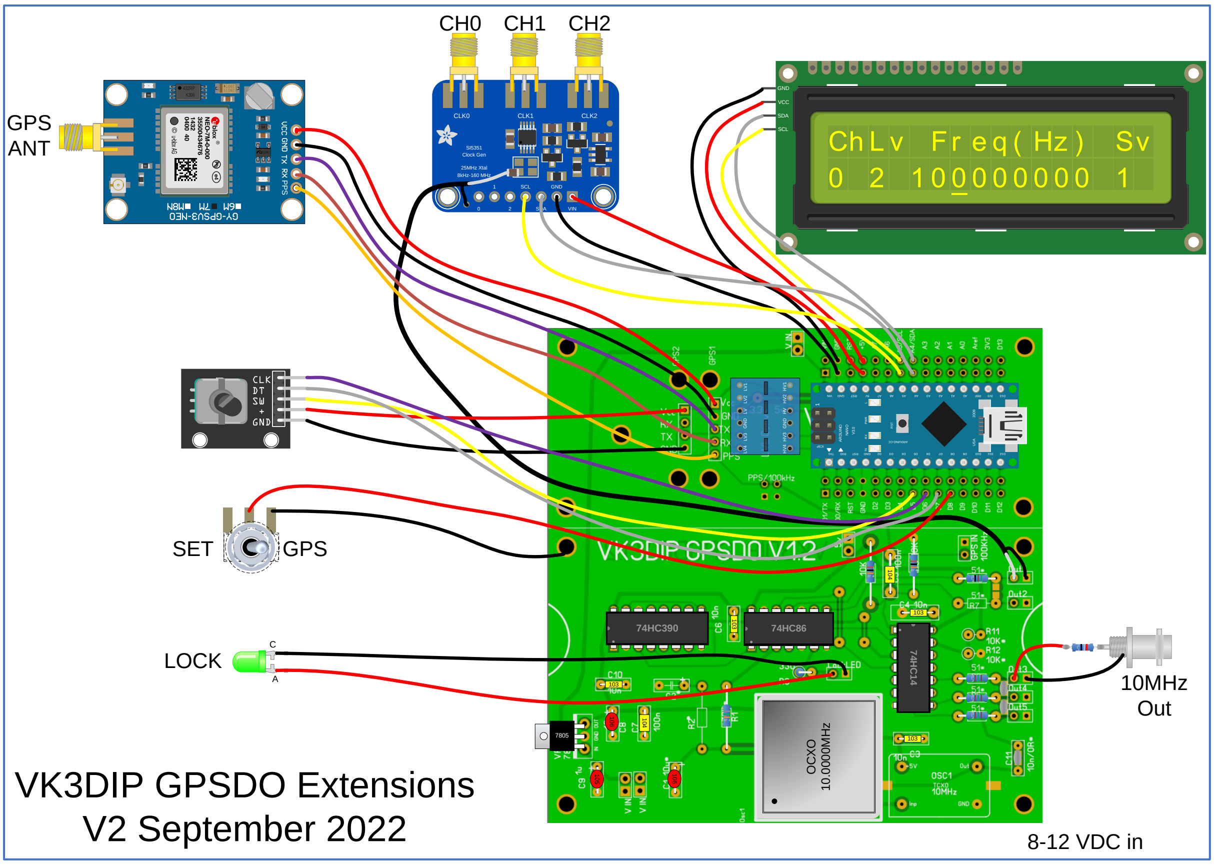


Figure 5 - Overall Fritzing Diagram

If you are having trouble seeing where all the wires go a larger (A3) size PDF of Figure 5 can be found here: http://yagicad.com/Projects/GPSDOV1_2StuffingA3.pdf .

I used the commonly available Rotary encoder Module from Ebay/ALI-Express, search "Rotary Encoder Module Arduino" usually less than $3 including postage and often a knob. This control can be rotated Clock-Wise (CW) or Counter-Clock-Wise (CCW) as well as being a push button switch by pushing down on the knob. The other hardware was a simple single pole double throw switch (or jumper link if you only want the setting functionality) which can be whatever type you have to hand. Not a momentary type though.

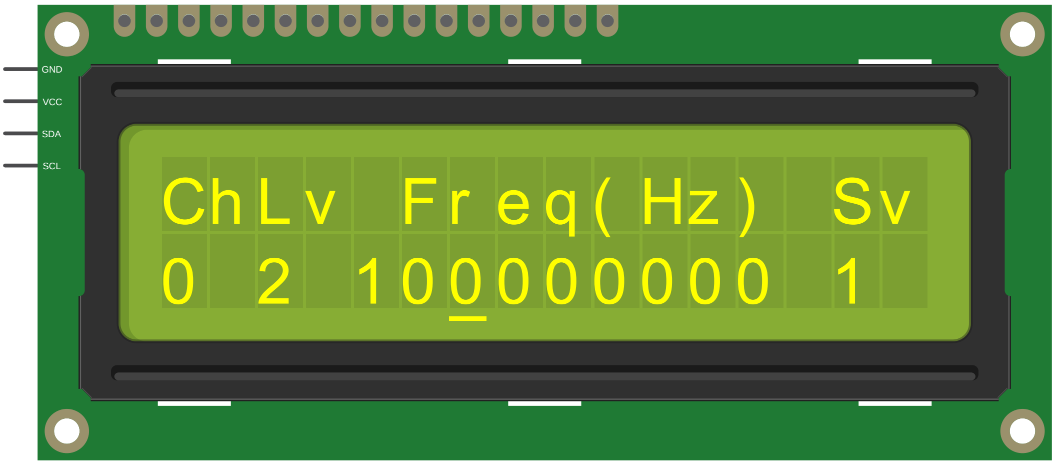
In words the Rotary Encoder module and switch are connected as follows:

- Rotary Pin CLK -> Nano D6 on V1.2PCB
- Rotary Pin DT –> Nano D7 on V1.2PCB
- Rotary Pin SW –> Nano D5 on V1.2PCB
- Rotary Pin + –> any convenient +5V/VCC on V1.2PCB
- Rotary Pin GND –> any convenient GND on V1.2PCB
- Centre pin of Switch –> Nano D8 on V1.2PCB
- Outer pin of Switch –> any convenient GND on V1.2PCB

I have as much as possible made V4.2 of the firmware backwards compatible. Ie. if you leave off the switch and rotary encoder (or any of the other optional bits like the display and the SI5351A board) it all should still work fine as a simple GPSDO. As an example that is why the new switch, which is used to indicate if you want to go into setup mode, can be left off. In the code I set an internal pullup to volts on that pin so that with nothing attached the software will read it as always high. High indicates normal GPSDO and display operation, whereas a short to ground (Low) indicates a desire to go to setup mode. If you only ever want to use this as a GPS locked pseudo triple signal generator then as mentioned before you can replace this switch with a short/jumper , but be aware the GPSDO will always start up in standard GPSDO mode /display and stay there until it has found at least three satellites and it is presumed the 10MHz will be well locked, once that is achieved the system will then obey the setting switch/jumper and go into setting mode if so requested. You can come back to normal GPS mode/display at any time by just setting the switch/jumper open.

On start up the Arduino will read its EEPROM to recover the frequencies/levels previously set, if no values have been saved then typically the EEPROM will return all bits high, or FF for a byte. The returned values are compared to the min and max values set in the code, and if outside those values (as they would usually be if never set) the code sets the frequencies to the default frequencies as per the previous firmware versions. You could of course be unlucky and be using an Arduino that had previously saved something in those locations that coincidently fell within the limits, in that case the frequency etc. will be set to those values.

As previously mentioned, even if set mode is requested (switch or jumper closed) the box will, after going through the startup messages, initially display the GPS screen as per Figure 1 - 10 Char Grid locator display. Once the number of satellites exceeds 3 the display will change over to something like Figure 6



```
ChLv Freq(Hz)  Sv
0 2 100000000  1
```

Figure 6 - Set channel data display.

Here the first/top line is basically labels for the data on the second/lower line. Going from left to right the first item is the Channel or clock output number, (Heading Ch, value 0 in above example). In the case of the SI5351A module there are three outputs so in this case we are indicating the first output (0) and the other two would be 1 and 2. The next item along is the output level of that channel, (Heading Lv, value 2 in above example). A Level of 0 corresponds to off, and 1,2,3,and 4, correspond to 2, 4,6,and 8mA of drive respectively, just multiply the number by 2 and call it mA's if you want a way to remember it. The next number is the current frequency in Hertz (Heading Freq(Hz) value 100,000,000 in the example above) this frequency can be set anywhere between the min and max values specified in the code, currently 8KHz and 150MHz respectively. Any particular SI5351A may have different actual limits so you can vary this if you wish in the code, but these are the typical figures from the data sheet. The final value is a flag to show if this current data frequency and level, has been saved in EEPROM or not, (Heading SV, value 1 in the example above), a value of 1 indicates the data matches that saved, a 0 indicates that one or the other (or both) of level or frequency does not match what is saved.

The final feature of Figure 6 to note is the cursor or underline character, shown in the example as being under the 1 MHz digit of frequency, the position of this cursor determines what the rotary encoder knob will do. The possible things you can do with the knob are:

- Rotate Clock-Wise (CW), ie. increment something.
- Rotate Counter-Clock-Wise (CCW), ie. decrement something.
- Push in the knob (and release), without rotating the knob (PB), ie. do an action.
- Rotate Clock-Wise while holding the knob pushed in, (Shift CW or SCW).
- Rotate Counter-Clock-Wise while holding the knob pushed in, (Shift CCW or SCCW).

Note: the rotary encoder has a detent mechanism on the shaft so that as you rotate it you feel and possibly hear one or more clicks. In an ideal world each click would correspond to a single step.

If the cursor is under the Channel Value then a single CW turn will increase the channel value and a CCW turn will decrease it, nominally one per click. The value will wrap at 2 going back to 0 for CW and vice versa for CCW. As the channel changes the Level , Frequency , and Saved status will also change to reflect the current values for the indicated channel. PB will have no effect.

If the cursor is under the Level Value similarly to the channel case a CW click will increment the level and a CCW one will decrement the level. The new level will be set to the SI5351 immediately but not saved to EEPROM, and any effect this has will be reflected in the Saved state, typically it would go from a 1 to a 0 indicating that the value was as per the EEPROM but is not now. Level wraps at 0-4. As mentioned previously 0 corresponds to off, and 1,2,3,and 4 , correspond to drive levels of 2,4,6,and 8 mA respectively. 0 is thus the lowest output level and 4 the highest. Again a PB here does nothing.

If the cursor is under one of the frequency digit positions then a CW click will increase or increment the frequency by the amount corresponding to that digit position, for example in the figure the cursor is under the 1MHz digit so a CW click will increment the frequency by 1000000 (1 MHz). Note the frequency is shown in a right justified / fixed width format, so even if the indicated frequency is say less than 1000000, then with leading zeros being suppressed the 1MHz cursor position will be still shown under a space where the MHz digit would have been if there was one. All cursor positions under frequency are valid from 1 Hz to 100000000 Hz, with relevel increments. CCW does the opposite to CW and decrements the frequency by the equivalent value. In all cases like level changes are set to the SI5351 immediately and any effects on saved status are reflected. Frequency will limit (not wrap) at max freq for CW and min freq for CCW. Once again PB does nothing.

The final cursor position of note is under the Saved Status value. The AtMega328 which is the processor on the Nano board has 1K of EEPROM which is rated at 100,000 writes but with no particular limit to reads. While 100,000 sounds like a lot if we saved stuff every time there was a click on the rotary encoder we would soon be making a significant dent in it. So reading the EEPROM is fine, but if the info is already saved we don't want to save it again, similarly unless we explicitly want to have the GPSDO start up with new values we shouldn't bother saving them. To limit unnecessary saves, neither CW or CCW clicks do anything here, only pressing and releasing the knob (PB) will actually do a save. Ie. change a 0 to a 1.

Cursor positions under the spaces between values do nothing CW, CCW, or PB.

That leaves the question of how to move the cursor? That is where the Shift operations come in. Doing a CW click while pushing down and holding the knob (SCW) will move the cursor one to the right, and similarly a SCCW click will move the cursor to the left. The cursor will wrap around at the ends.

Things to do with a GPS Locked not quite Triple Signal Generator

The original rationale for incorporating the GPS locked SI5351 was to generate frequencies that could be used as replacements for the reference oscillators in rigs that don't have external reference provisions built in. That use of course still stands, it is just easier to set the frequencies etc. without changing code. There are however other things that can be done.

Yes, you could use one of the SI5351 outputs for a GPS locked say 96MHz for an old microwave transverter, or similarly replace the 1MHz crystal oscillator in a frequency counter to make it also effectively GPS locked, but you could also just use the outputs like a signal generator.

There are a number of cases where the square wave nature of the output is actually helpful such as when wanting a low powered signal source to test a microwave receive convertor , or receiver front end. Even with the maximum output being say 150MHz, the square wave nature of this would make it rich in harmonics, for example the third harmonic of this would cover up to 450MHz (ie. 70cm), the 9th cover 23cm, and so on . Remember the harmonics are just as GPS locked as the fundamental. With an accurate signal source tuneable in one HZ increments and matching wideband or tuneable receiver, you could measure frequency edges of filters or 3dB points with great frequency accuracy.

Why have three sources though? One suggestion is that you could use one output for a signal source, and the other two for the first and second local oscillator of a receiver. Another utilizes Fourier and sees a single signal being made up of a number of discrete frequencies. If you set up say Ch0 with an output of 3.599000MHz you could see that as a CW carrier at 3.599000MHz, you could also see that as a lower sideband 1Kz tone at 3.600000MHz and check out how accurate your rigs dial is. If you were to use two of the outputs one at level 4 at 3.600MHz and one at 3.599MHz level 2 and combine the two signals using say a hybrid combiner of some sort (see the ARRL handbook for simple designs of these, or repurpose one of those Return Loss Bridges made from my old AR articles on the subject) then the resultant output will be what the ITU designate as H3E or single sideband full carrier (classed as a type of AM) . Most amateur boxes would pretty much see this as a close to 100% modulated AM 1KHz tone at 3.6MHz. Purists could add the third oscillator at 3.601MHz level 2 for the other sideband but that would need a three port combiner which is harder to build, plus the SI5351 can't do all three outputs with the same PLL so there would be some small phase differences that would probably show up as some distortion. Speaking of distortion, two signals at the same level through a combiner at close but non-harmonically related frequencies is a common method for doing receiver intermod tests. Basically there are lots you can do with multiple sources of accurate frequency.

Maybe this time this was the last part on the GPSDO but you never know, 73

Paul VK3DIP.