This modification turns the Ultimate3 kit into an accurate GPS-disciplined frequency reference (approx 0.03Hz accuracy). **The firmware has NOT yet been updated to operate with the Si5351A synthesiser module therefore this modifcation cannot be done with the U3S kit.**

The input frequency to the ATmega series of Atmel AVR microcontrollers limited to 40% of the system frequency. With the U3's 20MHz system frequency, this means the upper limit for direct counting is 8MHz. If you want to turn a U3 into a frequency reference with no more than 8MHz output, then all you need is a special firmware version. If you want to generate an output reference frequency higher than 8MHz then you will need to make a hardware modification as described below.

Firmware

This modification requires a special, beta (maybe we should even say Alpha), firmware chip. You can program your own ATmega328 if you wish, using the files in the QRP Labs forum. Or I can provide a programmed ATmega328 chip (email me , scroll to the bottom of the contact page). This beta firmware chip is not available from QRP Labs .

On power up you will see the following screen:

The v00003.08 is nothing at all related to the normal Ultimate3 firmware versions, it is simply a Beta firmware ID. Please think nothing more about that.

Parameter set-up

The set up of the parameters is done in the same way as the U3, but the set of parameters is greatly simplified. There are only three parameters, as follows.

"Frequency" sets the output frequency, your desired GPS-disciplined frequency reference. It can be anywhere in the DDS's available output range, 0 - 40MHz. But if you want an output frequency higher than 8MHz, then you will need to make a hardware modification as described in the section lower down this page.

"Divider" sets the external division ratio. If you are using an output frequency of not more than 8MHz in an un-modified U3 , then you leave this at its default value of 1. If however, you are using an external divider chip as described in the hardware modification section below, then you must set this parameter to whatever the external division ratio is. More on this later.

Method of GPS discipline action

The firmware applies a continual correction, measuring the output frequency once every second using the GPS 1pps signal as the counter gate. This is unlike the U3 , where the frequency correction only occurs in the calibration period that takes place in between transmission cycles. The firmware applies two methods for the frequency correction. An initial "Coarse" method to get the frequency locked on target, then a "Fine" method to keep it there during gentle temperature-induced frequency drift.

The "Course" method measures the DDS output frequency and compares it to the set target output frequency (that you entered in the Frequency parameter). The error is added or subtracted to the Ref. Frq. setting and the DDS tuning word is recalculated. Imagine for simplicity that your desired reference frequency is 1/20th the 125MHz DDS reference oscillator, i.e. 6.25MHz. A measured error of 10Hz is added to the 125MHz setting, and the DDS word re-calculated for the next second. Of course this correction is only 1/20th of what would really be required to compensate for the 10Hz error, due to the dividing action of the DDS. This means even the coarse mode correction is quite gentle.

(Note that the system does not currently write anything back to the EEPROM. Next time you power up, or next time you cycle through the menu, everything starts again from the beginning).

Eventually the measured error (between actual measured frequency and target frequency) becomes zero, and at this point the system switches over to the "Fine" method.

The "Fine" method stops adjusting the Ref Frq and re-calculating the DDS tuning word. Instead, it directly increases or decreases the DDS tuning word itself, by an amount equal to the measured error. Now this is very fine because if the measured error is 2Hz, then the applied correction will be 2 * the DDS tuning resolution of 0.029Hz, which means only 0.058Hz. In the practical case the actual error will be 1Hz or 0Hz, assuming there are no sudden temperature changes. The system therefore will keep adjusting the DDS tuning word, at every 1 second, by 0.029Hz in one direction or the other. A measured frequency of greater or equal than the target

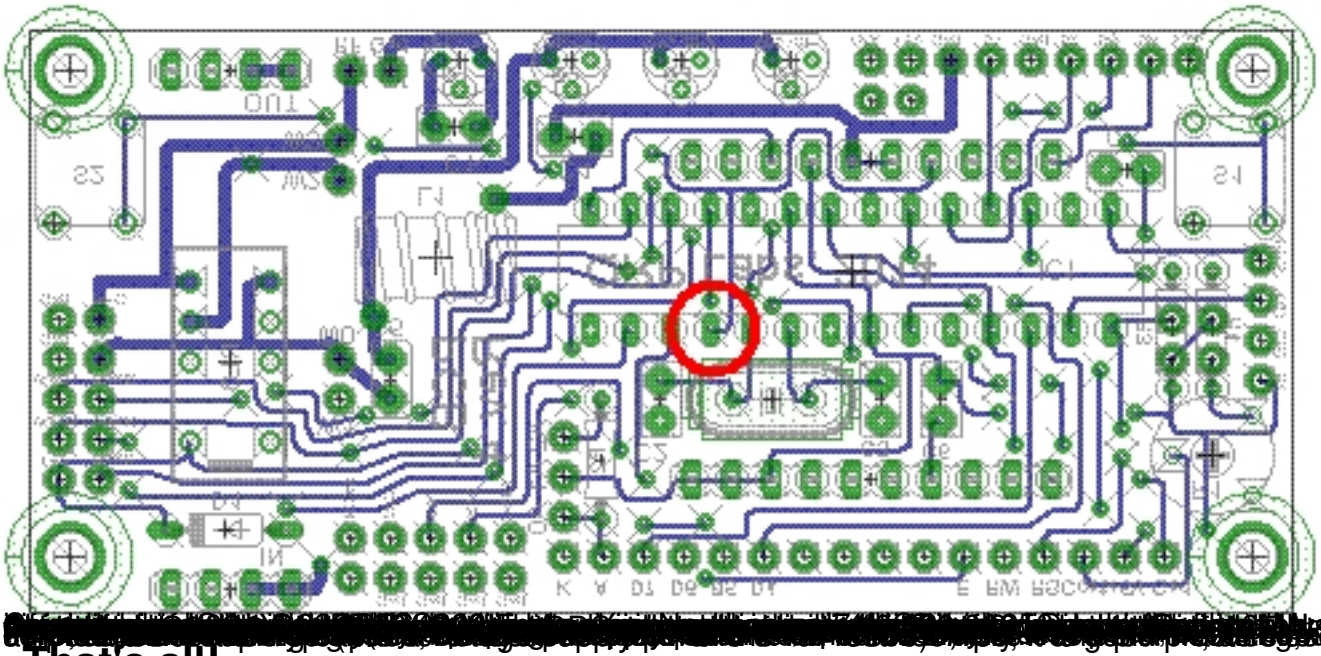
results in a down-step, a measured frequency less than the target results in an up-step. This 0.029Hz minimum DDS step is the accuracy of the GPS-disciplined frequency reference: +/- 0.03Hz (roughly speaking).

Note that if you are using external hardware division, for an output frequency of more than 8MHz, then the error measurement is on the divided value. For example, if you have chosen to have an output frequency of 10.000000MHz, and an external division ratio of 2, then the frequency that hits the processor is 5MHz, and the error measurement will be against a target value of 5MHz also.

Hardware modification for more than 8MHz output

This hardware modification, as I mentioned, is only required if you want to use an output frequency of more than 8MHz, because 8MHz is the maximum frequency that these AVR microcontrollers can count. You therefore must arrange for division to occur, between the DDS output and the AVR frequency counter input, to ensure that the 8MHz limit is not broken. For example: if using a disciplined output frequency of 10MHz, you could use a divide-by-2 chip, resulting in 5MHz to the AVR. If using an output frequency of 20MHz, then you could divide by 3. However generally powers of 2 are more practical, so a division ratio of 4 would be easier.

The AVR frequency counter input is at pin 11 in the U3 . You need to cut the trace at this pin and wire in the external divider. Turn over the
U3
PCB and look at the underside (NOT the component side). You can find the track at pin 11, circled in RED in the image below. This shows a Revision 4 PCB but the same applies to any of the PCB revisions. Check you have the right pin! On the chip itself, there is a little dimple next to pin 1 of the chip. Count down the left side of the chip to pin 11. My recommendation is to cut the track very carefully as close to the pin as possible. Be careful not to damage any nearby track or connection. Then you can wire the external divider in. I don't know where you want to physically put the divider chip, this is for you to work out, but it's always a good idea to keep wire lengths as short as possible. If you're careful you could fit the chip and a small piece of board, between the LCD module and the main PCB.



That's all!

Any suggestions to improve the firmware, contact me (scroll to the bottom of the contact page). The firmware files are available in the QRP Labs forum. If you want an actual chip though, contact me (scroll to the bottom of the contact page). The chip is not available from QRP Labs, so no point contacting them, please.