

By Jan Skirrow

A 60MHz AD9851 DDS Variable Frequency Oscillator

Tech Talk 12 discussed the construction of a VFO based on direct digital synthesis using an AD9850 chip. This article discusses a version of the same circuit using an AD9851 chip. I will assume you have read the first article, and will not repeat the information on construction or calculating digit weights.

The AD9850 is capable of clocking speeds up to 125MHz, and can be used to generate a good sine wave up to probably a third of this rate. However, to achieve maximum performance requires a 125MHz crystal driving the chip, which may be difficult to find.

The AD9851 is pin for pin compatible with the AD9850, so that the same PCB can be used. The AD9851 can clock to 180MHz, so that good output to 60MHz is possible.

The main advantage of the AD9851 is that it has a built-in x6 multiplier so that the external clock frequency need only be 30MHz to reach the maximum internal clock frequency of 180MHz.

The only changes needed to implement this design was to alter the software to engage the multiplier, and change the digit weights to reflect the 180MHz clock frequency achieved with a 30MHz external oscillator.

In addition, because the DDS chip outputs not only the desired fundamental, but also sum and difference frequencies of the clock and fundamental, the output filter was changed to suppress the unwanted higher frequency components.

The digit weights assuming an external 30MHz crystal oscillator and with the x6 multiplier enabled are shown in Table 1.

Table 1 - Digit Weights

<i>LCD_Digit</i>	<i>Digit Weight (hex)</i> <i>180MHz Ref</i>
10's	EF
100's	0952
1k	5D35
10k	03A411
100k	24688AA
1M	016C16A3
10M	0E38E25D

The software was modified to reflect the new maximum frequency of 60MHz and so that the power-up start frequency would be 35MHz.

The software changes can be followed on the PDF file mentioned below - which includes line numbers.

To alter the start-up frequency, the Start routine at line 647 is altered. The start-up LCD digits are entered on lines 654, 656, 658, 660, 662, 664 and 666. The first is the 10MHz digit (3), the second is the 1MHz digit (5). The rest are zero, although they could be set to any valid number.

The on-board multiplier is enabled by lines 520 and 521. Changing the constant in line 520 from 1 to 0 would disable the multiplier and the AD9851 would function as an AD9850.

To alter the maximum frequency requires changes in the BCDInc routine starting at line 226. This routine takes an encoder direction to increase the frequency, checks the new frequency to see if it exceeds the limit and, if it does, sets the DDS frequency at that limit until the encoder indicates a decrease in frequency.

Specifically, line 241 sets the maximum limit of the 10MHz digit - six in this case. Only even

decade limits can be accommodated with the current software, and PIC.

The routine SetMax at line 252 is used to load the LCD data bytes with the limit values. Line 263 sets the 10MHz digit, and the remaining digits are zeroed.

The software listing with numbered lines is in the PDF file at the first URL. The second file is an unformatted text file that can be assembled. The third URL points to a schematic. The values on the output filter are for the original 20MHz version. I will add a note on the output filter once I'm through playing with the unfiltered output!

<http://skirrow.org/Boatanchors/docs/pic2.pdf>
<http://skirrow.org/Boatanchors/docs/pic2.txt>
<http://skirrow.org/Boatanchors/docs/sch.pdf>

LCD Issues

The AD9850 version had bugs affecting, apparently, the LCD only - See Tech Talk 12. This version had similar problems. Most were resolved by putting a 0.1uF ceramic capacitor across pins 1 and 2 of the LCD module. This should be done on the module itself, but adding it on the PCB is also useful.

It is obvious that the LCD modules are tricky, due partly to timing issues, but also to temperature sensitivity and the apparent vulnerability to hash on the Vcc supply.

The software inserts wait intervals during LCD initialization, and always checks the 'LCD Busy' flag before attempting to write to it. The latter is foolproof - I think. The former may need some change if you are having problems with the display not starting properly. On my units, once in awhile the LCD will start with one digit garbled. It corrects as soon as the encoder is turned, so I haven't tracked this down. If you have more problems, you may have to!