

3Dfix

by Tommy Sullivan
W1AUV

I enjoyed and was intrigued by the paper “A Simple GPS Stabilized Oscillator” by N1JEZ, Mike Sequin, published in the Proceedings of the 31st Eastern VHF/UHF Conference. At the conference in April 2005 I picked up a PLL board and an oscillator from Paul Wade, W1GHZ. Soon after the conference, I ordered parts and got ready to assemble one of my own.

The following is from Mike’s write-up:

One problem with the Jupiter engine is that from the moment it’s powered, it outputs 10 kHz and 1 PPS. Until I had a valid 3D GPS fix (I recommend at least 4 satellites) the output was not accurate. I needed some way to be able to verify that I had a valid 3D GPS fix before I could trust the 10 MHz output. The GPS engine does not provide any type of output such as a line that might go high/low to indicate lock, so I had to think of another solution.

By the time the parts arrived I had decided build what Mike said the GPS lacked: a small microcontroller to generate a lock indication by monitoring the NMEA serial stream. As suggested in the article, I keyed my software on the GPGSA sentence. The sentence looks something like this:

\$GPGSA,A,3,01,20,19,13,,,,,,,,,40.4,24.4,32.2*0A

After the second delimiter (comma) in the sentence there is a three. A three in this position means that a 3D position has been established.

The microcontroller I chose for this application is a Microchip PIC. I have experience with a wide range of microcontrollers and microprocessors but, I chose the PIC because of its popularity in ham applications. The firmware development tools, for assembly language at least, are inexpensive as are chip programmers.

The PIC16F688 is a Flash based microcontroller that has a built-in UART. A smaller controller could be used with a bit-banging serial routine. I’ve done this before but I prefer the hardware approach. The hardware UART helps save code space and CPU cycles for enhancements and updates.

The 16F688 is available in a 14 pin DIP package which makes it ideal for hand assembly and prototype boards. The 100 mil spacing of the pins is easier on my eyes too.

The software is straight forward. The PIC just monitors the serial data looking for our target string. If the three is found in the right spot a pin goes high on the PIC and this

drives an LED. If the field in the GPS string doesn't have a 3 then the LED goes out as the pin goes low in response.

It is possible to reverse the LED so that it is on until a 3D fix is acquired, then off. This will save a little power. Every little bit of power counts: especially in battery powered applications. The overall power required for the PLL and GPS is so much higher that it swamps out the small amount of power for the LED. I mention this option though because it is basically free.

I soldered a prototype together and tested it with the assembled PLL and GPS. It works great. All that is needed is to tap the TX line and power from the GPS. The PIC can run from 3.3 or 5 volt power. This means this design will work with either a 3.3 or a 5 volt GPS.

I use the PIC's internal oscillator to drive the CPU and UART. This oscillator's frequency will vary with temperature. With the addition of a small 32 KHz oscillator the software can be changed to compensate for changes in frequency that may cause errors in decoding the serial data. I plan to add this capability to the software in the future.

The source code is available (C or assembly) to anyone who wants it and has the equipment to program the chip themselves. If you want a programmed chip let me know. They cost \$3 or less (plus postage). Contact me for details.

tpsully@verizon.net
6/22/2005

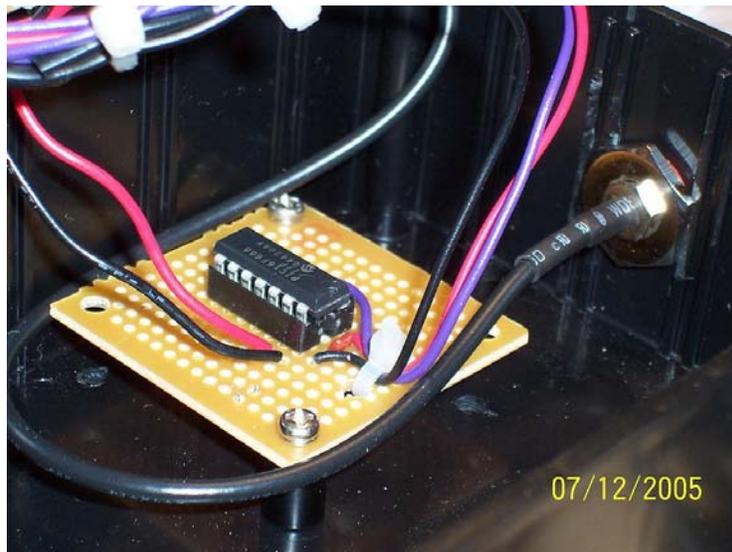


Figure 1 The prototype board version of 3Dfix.

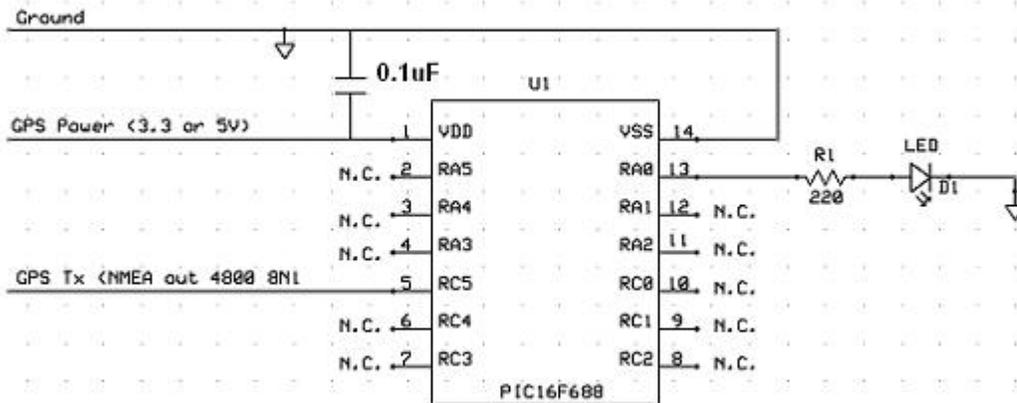


Figure 2 The Schematic



Figure 3 Front View