Budget GPSDO – Version 0.0

Summary

This paper describes a low cost GPS disciplined oscillator (GPSDO) module that can be used to construct a GPSDO that achieves an accuracy of better than 10MHz ± 0.01Hz. The module is self monitoring, and indicates if it has not reached the desired accuracy (normal for the first few minutes of operation). It also is self configuring, and requires no tuning by the user. It uses low cost parts.

The design was not created for any specific need. The author believes there are people who want to be more ‘hands on’ than buying a ready made black box GPSDO, but are not interested in designing a GPSDO from scratch.

As the design is a module, constructors may supply some of the requirements from equipment they already have. And if the constructor decides at a later stage they want better performance, the module can be retired and other components reused. This may be the lowest cost way to get experience with a GPSDO.

Background

The idea for a Budget GPSDO initially was an exercise to see how few components could make a working GPSDO using an Oven Controlled Crystal (Xtal) Oscillator (OCXO). Cost was not a consideration, but more an outcome. The simpler the design, the less parts, the lower the cost.

Once the design was proven, was it useful? A quick survey online turns up many GPSDO designs. What characteristics of this design make it different?

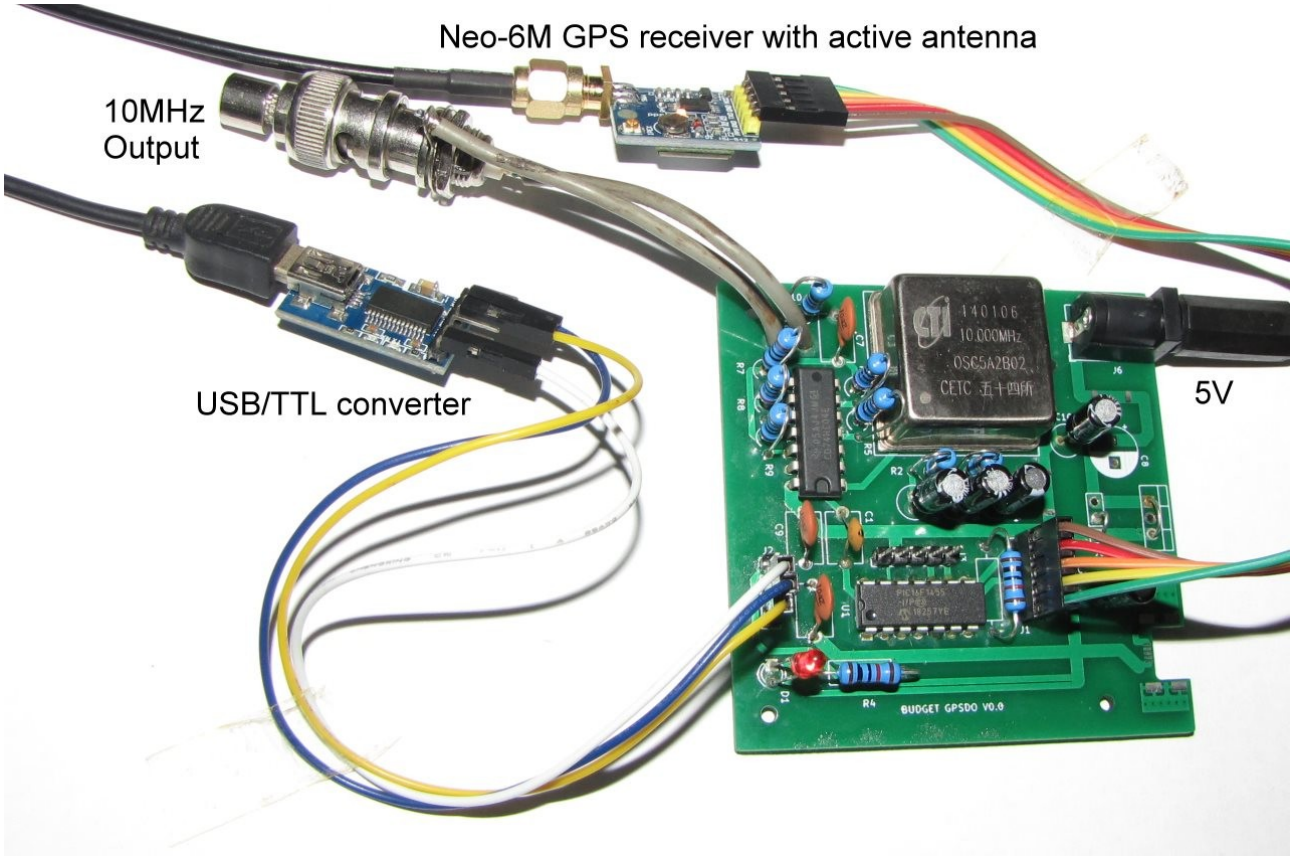
- The most significant difference is the control voltage to the local oscillator is changed at infrequent intervals. This is a benefit when using cheap GPS receivers. It also allows the processor to compare the 1pps with the oscillator over a period of minutes, and determine the inaccuracy of the oscillator without the need for a better reference for comparison.
- The processor uses the disciplined oscillator as its clock. Once the oscillator is close to stable, the processor counts cycles and locks the oscillator long term to the GPS unit. Within the limits of the control voltage, the processor will catch and correct frequency changes of up to 5Hz.
- The control voltage is generated by a dithered Pulse Width Modulator (PWM). This allows changes of less than a microvolt, with no external DAC and fewer components than most other designs.

Another advantage is that a printed circuit board (PCB) design is available.

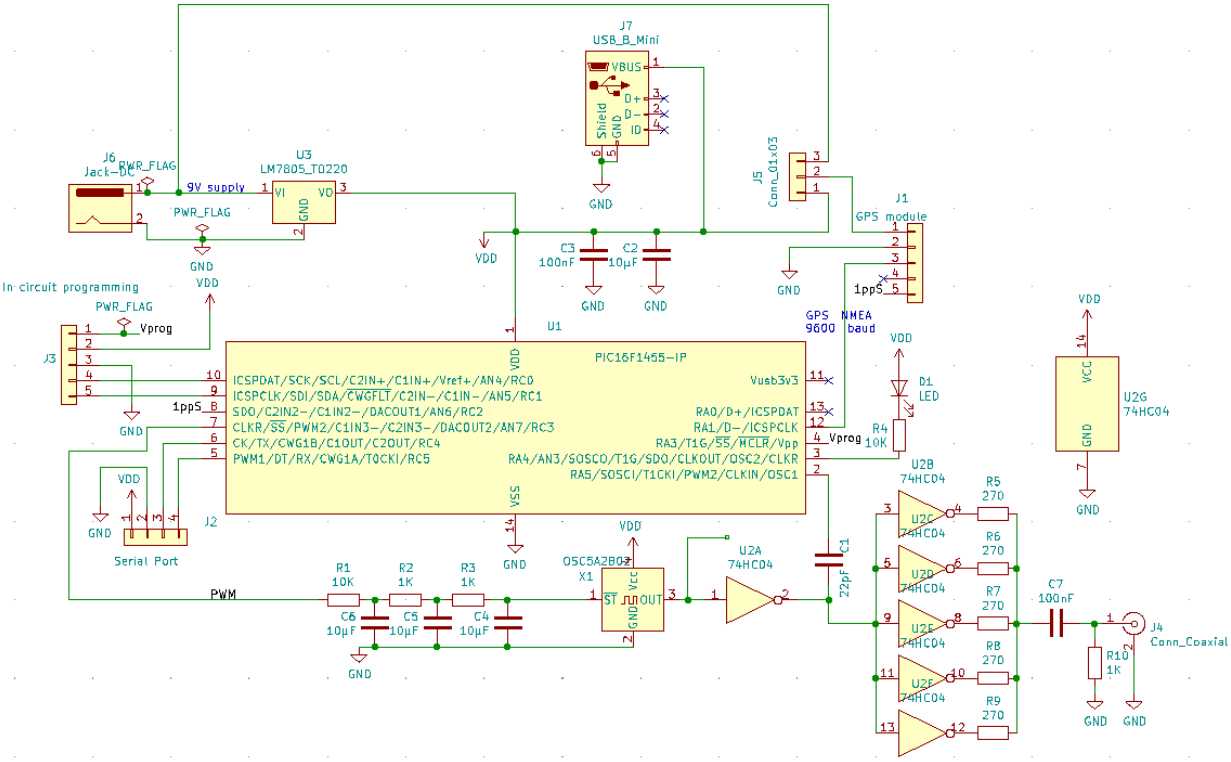
The philosophy is that the design of a GPSDO can be modular. The PCB provides only the disciplined oscillator and a 10MHz square wave output into 50Ω. The power supply, GPS module, optional serial output module, enclosure and any other embellishments are up to the user.

This places the design between commercially available single box designs, and most other DIY GPSDO projects that are provided as circuits. Whether such a market exists is unknown.

Description



A working configuration



The Circuit

The circuit includes multiple power options, required so that the PCB design software (KiCad) includes them.

In the pictured configuration, the supply is 5V from a wall wart (5V 2A) repurposed from obsolete equipment. There is space for a micro USB B socket, suitable for a phone charger. There is also space for a voltage regulator such as a 7805 on the right of the board. It has a bypass link in the picture. The location allows for a suitable heat sink, should it be needed.

Performance

The module estimates accuracy from the changes of control voltage applied to the oscillator. Before the system is used to produce 10MHz, it does a self calibration to determine the oscillator sensitivity to the control voltage. This can be expressed as V/Hz, e.g. 0.1V/Hz – a change of 0.1V changes the frequency by 1Hz. When the system is running, all control voltage changes are corrections. Changes correct two errors: frequency errors and offset errors. Once the system is locked onto the GPS, in the long term the oscillator should produce 10 million cycles each second. An offset error is because the oscillator is running ahead or behind the desired count. The correction applied will change the frequency by as much or more than the frequency error, the extra being to correct the offset error. If the frequency change is 0.01Hz then the estimated error before the change must have been 0.01Hz or less. Using the control voltage change as a monitoring tool allows the system to put an upper bound on the frequency error.

Using this technique, the processor monitors for an offset of less than 2 cycles after 192 seconds. If this is achieved it is indicated by altering the LED flash. This corresponds to a frequency error of less than 10MHz ± 0.01Hz or less than 1 part per billion (10E9) usually abbreviated as 1ppb. The module will usually achieve this accuracy within 15 minutes of powering up. However, there are many factors that affect the performance, and achieving 1ppb is not guaranteed. Some of the factors are quality of the GPS signal, quality of the oscillator, temperature changes, and power supply quality. A well documented GPSDO project is the *Lars DIY GPSDO with Arduino and 1ns resolution TIC* that can be found online at https://www.eevblog.com/forum/projects/lars-diy-gpsdo-with-arduino-and-1ns-resolution-tic/. Lars Walenius has posted several documents covering his experiments and discussing the factors that affect GPSDO operations.

This module uses a used OCXO with modest specifications, the OSC5A2B02. This was chosen initially because it was readily available and low cost. Tests show it should fulfil the expectations of someone who wants a GPSDO as the best frequency standard in their workshop. Very few standalone instruments boast an accuracy of 1ppb, and someone who can afford them is likely to be able to spend the money for a higher specification GPSDO. The OSC5A2B02 is however capable of maintaining an accuracy of better than 10MHz ± 0.001Hz when other factors are well controlled. It is not normally the weak link in the chain.

Most constructors find the requirement of good quality GPS signal is a limiting factor. The cheapest modules are more than adequate if they receive strong signals from a good proportion of the sky. Modules specifically designed for timing perform better in more difficult situations. The modular nature of this GPSDO allows the constructor to choose a solution appropriate to their situation.

Operation

The processor (PIC16F1455) needs to be programmed. This can be done before it is installed on the board, or it can be programmed in-situ using the 5 pin connector on the PCB. The connector pin out is the same as a PICkit3.

On power up, the processor runs on its internal clock. It sets a control voltage for the OCXO then attempts to switch to the 10MHz provided from the OCXO via the 74HC04. It flashes the LED while attempting to do so. If for some reason the 10MHz is not available, the processor tries repeatedly and the LED flashes rapidly.

Once the processor is running on a 10MHz clock, it monitors the NMEA output of the GPS module for up to 10 seconds. If nothing is received the processor will loop if it has calibration data, or will reboot if there is none. The 10MHz output of a recently calibrated system will normally be within 1Hz of accurate after 15 minutes, so the output may be useful. Most GPS receivers by default produce NMEA output as soon as they are powered, so no input is a fault condition.

NMEA data is continually monitored by the processor. Before the GPS module indicates it has a valid fix, the processor reports on satellites in view. Once there is a valid fix, the processor also monitors the 1pps input and enters either a calibration or run state.

In calibration state, the processor determines the oscillator sensitivity to the control voltage and also a default control voltage for subsequent restarts. This takes several hours. [it doesn't need to be so long, but it is only needed once so has not been optimised]. Once calibration is completed, the values are written to non volatile memory for subsequent use, and the system reboots.

In run state, the processor counts the number of quarter cycles (nominal 25ns) between 1pps arrivals. If the OCXO is still warming up, this may be outside the range of the control voltage, so the processor loops waiting for frequency to come within range (for the OSC5A2B02 this is about +-10Hz). Once within range, the processor disciplines the oscillator until either the GPS input is removed or the system is powered off.

User Interface

The module has a LED that provides basic information. It is adequate to verify operations, but for more detailed information the UART (serial interface) can be used. Most designs would use the UART to process the NMEA data. This design uses a software receiver for the NMEA data, allowing the UART to be dedicated to a user interface.

The user interface is primarily to log events – mainly the second to second offset captures, and also the control voltage changes. A control voltage change entry looks like this:

```
Time 070821 UTC. Ctrl 2.056413 6.360 ppb
```

The time is captured from the NMEA data. The Ctrl (control voltage) assumes an exact 5V supply. It is quoted to 7 figures because it is derived from a 24 bit value. The ppb figure is the difference between this control voltage and the previous control voltage converted to a frequency change.

The second by second offsets report looks like this:

```
00000 00 00 00 00 FF FF FF 00 00 00 00 00 00 FF FF 00
00016 00 00 00 00 00 00 01 00 00 00 01 01 00 00 01 00
```

The 5 digit number is seconds since the last voltage change. It is followed by 16 seconds of data. Each two character are a hexadecimal representation of the offset in 25ns amounts.

There is an option is to enable NMEA pass through. The NMEA data is copied from the GPS module to the user interface. This can then be input to a program for analysis and display. One such program is VisualGPS but there are many others.

The user has available some simple commands, e.g. to control logging, to ask for a date stamp.

The user interface is subject to change, and is described elsewhere.

Design philosophy

This was seen as a software design challenge – minimise external components and get the processor to do the work. This results in some unusual choices.

The PIC16F1455 has some features that make it a good choice for the task. It also was the author's choice due to previous experience with PIC processors. Most important, the processor has an internal PLL to multiply the external 10MHz clock to 40MHz. Some peripherals, importantly Timer 1 and the PWM are run at 40MHz.

Timer 1 is used to capture the arrival time of the 1pps to within 25ns. Ideally the arrival time should be known more accurately than this, and almost every other GPSDO has external circuitry to accomplish this. But by using statistics, the significance of the deficiency is reduced.

The solution used here is to accumulate many 1ppS arrivals and work with averages. If there is predictable jitter, the detector does not need to accurately time each pulse. In the case of a cheap GPS, the jitter is caused by quantisation of the 1ppS due to the GPS module clock not being aligned with the GPS signal. This should result in an even distribution of arrival times over the quantisation period. This has been documented for some GPS modules in this paper – https://hamsci.org/sites/default/files/publications/2020_TAPR_DCC/N8UR_GPS_Evaluation_August2020.pdf. If the detection window is a similar period to the jitter distribution (in this case 25nS window and approximately 22nS jitter) then a change of the 1pps distribution by a few nanoseconds will cause some to be detected earlier or later than most. Averaging a large number of arrival times gives a value that is statistically valid to within a few nanoseconds.

The decision to adjust the control voltage at infrequent intervals was partially dictated by the need to collect arrival times over a period. Changing the control voltage during collection invalidates data collected before the change. The infrequent change strategy also allows error estimates without resorting to better standards for comparison,

Averaging is also used to obtain a precise control voltage from a 10 bit PWM. This is achieved by varying the pulse width for every pulse, so that the filtered output is between the values that would be obtained with a constant pulse width. The output emulates a 24 bit DAC, and is accurate enough that it is not a factor in performance (although the supply voltage from which it is derived is a factor).

The program is written in assembler. This was chosen because the processor has to handle a number of tasks, some being quite time critical. Unfortunately this results in a large source file that is not easy to maintain. The assembled code however is quite small and there is room for enhancement. There is not a lot of spare data space.

13 September 2021