USING GPS TO ENHANCE AMATEUR COMMUNICATIONS

Andy Talbot G4JNT

What is GPS, and how does it work?

The Global Positioning Satellite system (GPS) is designed to allow a user anywhere on (or above) the Earth's surface to determine their position to an accuracy of a few metres. It does this by maintaining a constellation of satellites in very accurately determined orbits, such that at least four are visible from any location at any one time. Each satellite transmits a spread spectrum code so that observers are able to determine the time of flight of the signal, and hence their distance, from each satellite observed. Once these distances are known, trigonometry and triangulation allow the observers position to be calculated.

In order that the exact distance from each can be determined, the satellites have to be precisely locked together in time. To ensure this, each satellite carries a caesium frequency standard (plus a rubidium one, and crystal oscillators, as back up in the event of failure). So each and every satellite now transmits a code to the ground that allows the exact time to be determined. To measure distance accurately to a few metres, timing accuracy has to be of the order of nano-seconds (3ns = 1 metre of signal travel in free space)

Any GPS receiver locks to the codes, measures the time difference-of-arrival and calculates the distances to its location from each satellite, determining the user's location as described above. But, as a by-product of calculating the location fix, the receiver also knows the exact time. Although, strictly speaking, this need only be relative or 'GPS System time', the GPS network was designed for timing use and time dissemination as well as position fixing, and the final result is that Universal Co-ordinated Time or UTC, is available from any GPS receiver.

> The result is that anyone, anywhere on the surface of the Earth knows the exact time, to within a few nano-seconds, of anyone else with another GPS receiver, which opens up a huge range of possibilities for experimentation in radio communications.

Getting and Using the Data

First of all, we have to get access to the timing information. The usual portable GPS receiver used for walking or in cars is of no use to us here – the only timing information it supplies is usually a clock display - and we need something more fundamental than this , to get inside the GPS receiver hardware. Enter the GPS module, a small self-contained unit that has a raw GPS receiver to do the time extraction and a processor to calculate the location – that is all, and in fact many GPS end user products often contain a similar module as their 'front end' The data is generated in a computer readable form, usually as serial data in RS232 format, and the interface is used to communicate with, and set-up the module. The time of day and the date is included within the serial data stream, along with the observer's latitude and longitude and a host of other useful information on satellite status, signal quality and accuracy of the position fix. As well as the serial data stream, a separate one Pulse-Per-Second (1 PPS) output is provided for precision timing purposes; the leading edge of this corresponds to the exact UTC second. The data frames on the serial interface appear once per second, corresponding to each pulse appearing on the PPS signal line and these are transmitted immediately AFTER the pulse to which they refer

An example of the serial data, in a standardised format from most GPS modules, is shown in Figure 1. This is known as the National Marine Electronics Association (NMEA) format and is standardised at 4800 baud, 8 bit, although other signalling speeds are available as a set-up option on some GPS modules. The data format should be the same from all manufacturers, but there are often subtle differences, so be aware of the exact structure when making use of this data. An example of this ‘non-standard standard’ can be seen in Figure 1 where the reported time can now include fractions of a second. As an alternative, each manufacturer also allows the data to be sent and received on the serial interface in their own native binary format, this option usually provides greater flexibility and a wider range of options than does NMEA.

A (by no means comprehensive) list of some GPS modules suitable for amateur use is shown in Table 1 A photograph of the Jupiter-T module is shown Figure 2

Figure 1 - Typical NMEA data format from a Garmin GPS25 module.

$GPRMC,212132,A,5054.5876,N,00117.4041,W,000.0,000.0,141202,003.5,W*7B
$GPGSA,A,3,,11,14,,28,31,,,,,,,3.7,2.4,2.7*38
$GPGSV,2,1,06,03,23,146,,11,64,276,40,14,33,083,44,20,21,215,36*74

The sentence starting $GPRMC contains the most useful data

Time	Latitude	Longitude	Date
HHMMSS	DDmm.mmmm	DDDmm.mmmm	DDMMYY
21:21:32 UTC	50° 54.5876’N	1 ° 17.4041’W	14 December 2002

The same data as sent from a Motorola Oncore module with the NMEA option set..

$GPRMC,212132.0000,A,5054.5876,N,00117.4041,W,000.0,000.0,141202,003.5…..

Table 1 - Some GPS Modules suitable for amateur use		
Manufacturer	Model type(s)	Data formats available
Motorola	Oncore UT, GT, M12	Binary, NMEA
Garmin	GP25, GP35	NMEA (RS232 levels)
Navman/Rockwell/Connexant	Jupiter, Jupiter-T	Binary (Mot. compatible), NMEA
Trimble	Lassen	Binary, NMEA

Figure 2 – The Jupiter-T GPS receiver

Accuracy

Although the GPS receiver has to be able to determine relative time to within a few ns internally, this accuracy, unfortunately, does not apply to the 1 PPS output. Due to hardware limitations, and the need to synchronise to its own internal clock oscillators, the 1 PPS output signal often has a random jitter on it that is many times worse than the internal timekeeping needed for position fixing. Most standard modules specify a pulse-to-pulse timing accuracy in the region of 50ns to 500us, but some modules designed specifically for timekeeping can manage appreciably better than this. For precision timekeeping at a fixed location, a position hold mode is usually available as a set-up option.

The measured jitter from a Jupiter-T module averaged over a 5 second interval is shown in Figure 3. Here the vertical scale was limited to 0 to 100us by the measurement technique employed to obtain the data, and a few instances of wrap-round can be seen, where the time appears to jump nearly 100us in one step. The jitter appears completely random in nature and by averaging over a longer time period, the errors reduce in magnitude. Figure 4 shows the same data averaged over a five minute rolling period, where it can be seen that all the short term variations have been ironed out, although there are still several tens of ns timing error over periods of tens of minutes. Longer averaging periods will progressively reduce the size of these errors.

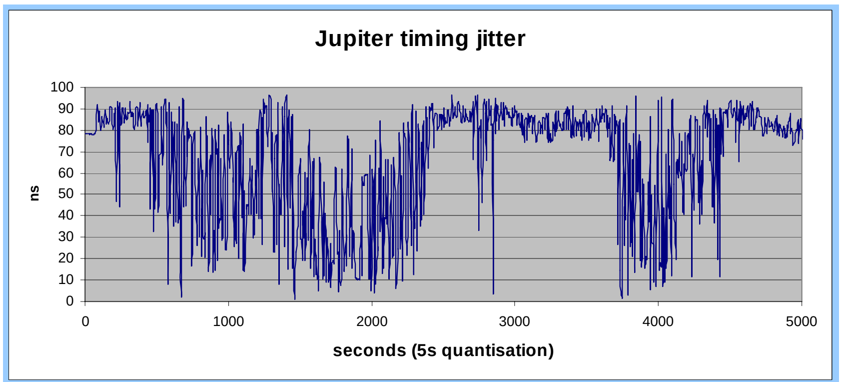

Figure 3 - Timing Jitter from a Jupiter-T GPS module, averaged over 5s timing periods

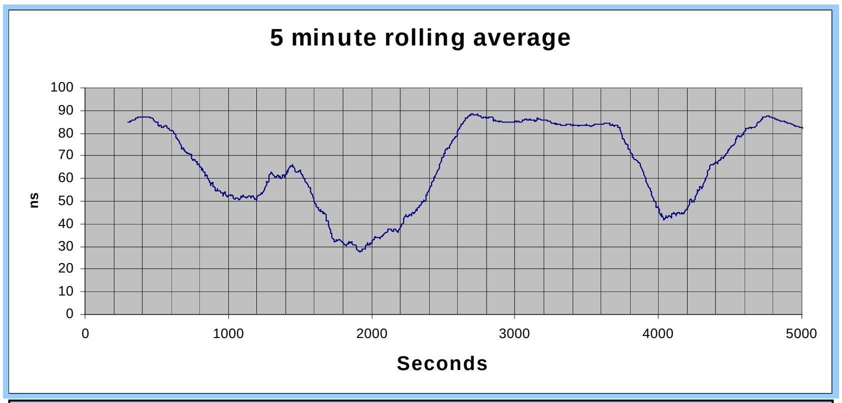

Figure 4 – Jupiter-T GPS Module timing data, averaged over 5 minute rolling period.

Frequency Control

Now that we have time to a high level of accuracy, we can use this to measure and control frequency. The simplest way is to use the 1 PPS signal to gate a frequency counter, but we now immediately run into problems caused by the timing jitter. 100ns of timing uncertainty in one second corresponds to a timing error of 1 part in 10^{-7} and this, taken with the +/- 1 count inherent in any digital counting scheme means that we cannot measure frequency by using the 1 PPS to control the gate to a frequency counter with an accuracy any better than about 0.3 parts per million – which is only as good as a low cost temperature compensated crystal oscillator could achieve. For higher accuracy, we could divide the 1 PPS signal down and employ a gate time of, say, 1000 seconds. The resolution is now in the order of parts in 10^{-10}, but we now have to wait 17 minutes for each reading to update. A long winded process, but useable if all we want to do is check the frequency of a master oscillator.

The next trick is to use the 1 PPS signal to automatically correct an oscillator. Voltage control is added to a crystal oscillator by the addition of a varicap diode that allows it to be tuned over a few parts-per-million – giving a Voltage Controlled Crystal Oscillator (VCXO). Typically, the crystal may be a high specification 5 or 10 MHz device in order to form part of a master frequency reference. The output from the oscillator is divided down in simple logic circuitry to give a 1 Hz signal which is compared in phase with the 1 PPS from the GPS receiver. The output from the phase detector is low pass filtered and fed back to control the VCXO, with the result that this is now, in theory, locked to the GPS signal. The complication is, once again, the jitter and timing uncertainty of the 1 PPS signal, so averaging once more comes to the fore. By making the Phase Locked Loop time constant long enough to smooth out the worst of the jitters, the oscillator can be made stable enough to form quite a good master source for most amateur signalling requirements. But, in order to do this properly, we need to step back and take a look at how much averaging is really needed to give a good result.

If we assume the 1 PPS signal jitters from pulse to pulse typically by 50 - 200ns, and we want a final frequency stability in the region of 10^{-10}, this immediately suggests that we will probably require a PLL time constant that averages over 1000 to 10000 seconds (17 minutes to 2.8 hours). In fact, the situation is even worse than this, as the number of samples used to obtain the average and the resultant stability improvement approximates to a square root relationship. So, in practice we need to average over many hours, or days, to achieve a guaranteed sub-10^{-10} result. Also, GPS induced errors themselves – such as changes in the satellites' orbits and RF propagation anomalies - mean that the recovered signal itself varies over a few parts in 10^{-10} over periods of tens to hundreds of seconds. In fact this system based variation can be seen in the graph of Figure 4 which has removed the 1 PPS jitter by the 5 minute averaging. Any attempt to make an analogue phase locked loop filter to just tack onto the end of the otherwise quite straightforward PLL hardware just described is infeasible – try building and using an analogue CR filter with a time constant of a day!

Instead, digital methods have to be employed using a microcontroller and D/A converters to synthesise a PLL with very long time constant. This is not too difficult, and an excellent design by Brooks Sheera, W5OJM, for the radio amateur has been around for some years now. See Reference 1 for details. Such designs are usually referred to as a GPS Disciplined Oscillator (GPSDO) rather than as being 'Locked', as the GPS signal continuously controls the oscillator rather than locks it. The point is rather moot, but the term GPSDO has entered the literature and become firmly established.

The need to have smoothing times of many hours means that the underlying VCXO has to itself be stable over this period so that its own internal 'wobbles' that may occur faster than the PLL could deal with them do not appear in the output. The biggest cause, by far, of such variations are temperature changes and a good quality ovenned oscillator is usually essential. These are not cheap when new, although there are plenty to be had at reasonable prices on the surplus markets these days. A more modest design of GPSDO, specifically targeted at LF low data rate signalling where jitter on the output did not matter over periods of several seconds was published in Reference 2. This made use of a simple crystal oscillator for the VCXO, with a PIC microcontroller for a modern version of the long-established 'Huff and Puff' stabiliser, rather than adopting a true phase locked solution.

The Simple GPSDO

But, help is at hand. One specific GPS receiver module, the Navman Jupiter-T, has an additional output available from it at 10kHz specifically designed for simple frequency locking. The signal still suffers internally induced timing jitter due to clock edge synchronisation, but is now randomly spread over each cycle of 10kHz rather than at 1Hz, making averaging by an analogue filter a much simpler prospect. A VCXO can be phase locked to this 10kHz signal, and even with a PLL time constant of just a few seconds the effects of the hardware-produced internal receiver induced jitter can be reduced to insignificant proportions, allowing a lower cost crystal oscillator to be used. There are still the GPS system variations present at a few parts in 10^{-10} (worst case a few parts in 10^{-9}), but the result is still satisfactory for most amateur and commercial radio communication needs. A simple GPSDO using the Jupiter-T module can be made with just four CMOS ICs and a packaged 10MHz crystal oscillator. Details of one design can be found at Reference 3. And a spectrum plot of the output of this simple GPSDO, multiplied up to 1GHz can be seen in Figure 5. A random frequency wobble that does peak up to around one Hz maximum, but is more typically around 0.2Hz, can be observed, with a period measured in tens of seconds.

Once an accurate reference signal has been generated, which is usually at 10 or 5MHz, this can be distributed to all radio equipment that needs it, ensuring everything is locked together. Some amateur transceivers have the option for an external reference input (all professional ones do); others can often be modified to accept one – see the sidebar for an example. Local oscillators for VHF and microwave transverters can be phase locked to the reference signal, eliminating the tedious long warm up and steady drift often accompanying high frequency operation and making frequency setting an exact art. Various techniques for phase locking can be employed, and several designs have appeared in the journals. Reference 4 gives a few ideas.

So, now that we can generate an accurate frequency reference locally, and we can get a timing reference that gives UTC to a few tens of ns accuracy, what else can we do with these?

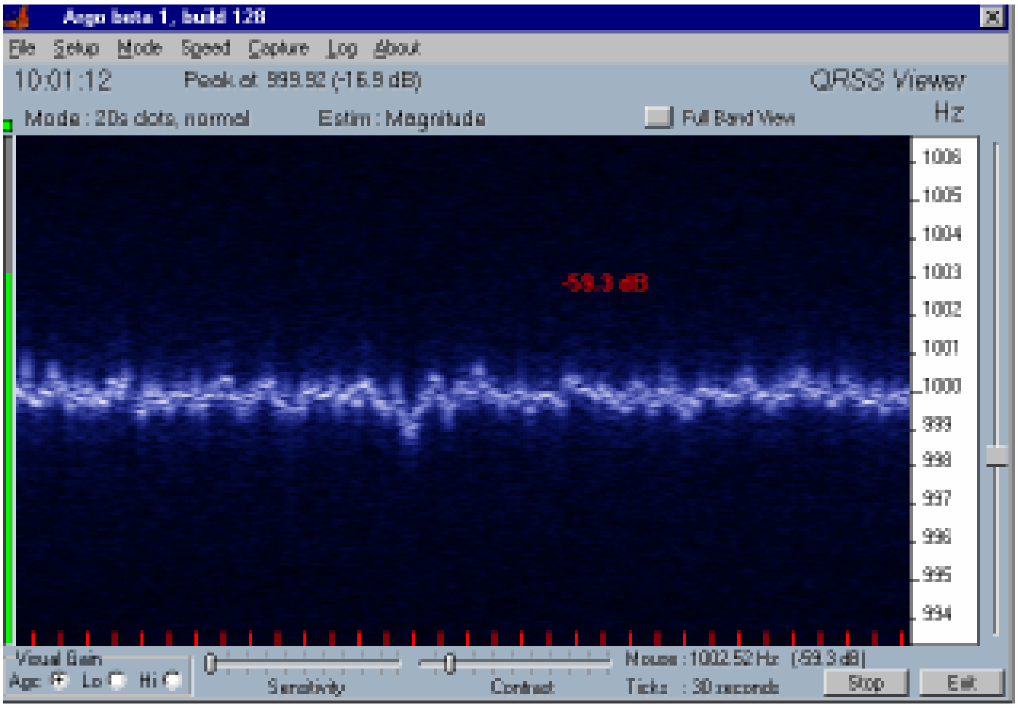

Figure 5 – Output from the Simple GPSDO multiplied up to 1GHz

Synchronising data modes

All datamodes have to obtain, and keep, synchronisation between Tx and Rx. Apart from having to recover the transmitted bits, whether this is by means of frequency or phase shift keying (FSK or PSK), or something more complex, the software has to keep track of the correct transition point where one data bit merges into the next. This synchronisation issue is a bigger problem, by far, than merely deciding if a ‘1’ or ‘0’ was sent. The timing recovery process often uses non-linear processing, such as edge detection and envelope recovery, and is always the first to suffer when the signal gets weak and noisy. What if we could derive the timing information independently, and let the decoding software just get on with extracting the data? A solution is to adopt GPS timing. Any user of a simple low cost GPS module has access to timing that is accurate to within a microsecond of UTC, so if the two ends of a digital link agree to start their transmissions at a precise point in time, and the timing can be maintained for as long as required, there should be no further need to extract any timing information from the noisy signal itself.

Things are not quite this simple as there is an inevitable time delay in the transmission path between the two stations (which could be up to 140ms for long path HF QSOs right around the world, but will more typically be a few milli-seconds to tens of ms), as well as filter group-delay in the transmitters and receivers which also contribute a few ms. But such matters can be resolved. On LF, where slow data rates and very low bandwidths are employed, the one pulse-per-second can even be used directly. If signalling is done at one bit-per-second or slower, then the bit transitions can be timed using the 1 PPS output from a GPS receiver directly. Provided both stations agree to start at a known time – say the exact 15 minute interval – everything ought to carry on seamlessly. At this data rate, typical propagation delays of tens of milliseconds (10ms is equivalent to 3000km) will be insignificant and no timing correction will be called for.

Coherent Communications

Using GPS to independently derive timing information can be carried a stage further when coherent modes such as PSK are in use. Here, not only does the symbol timing information have to be extracted from the noisy received signal, but the carrier itself has to be regenerated so the phase of each symbol can be determined. Since the phase is being keyed, usually by 0/180° for the data transmission, the carrier phase has to be regenerated *without knowing in advance whether is should be inverted for any particular bit*. This can lead to some wonderful and clever techniques for carrier recovery. A favourite simple carrier recovery technique for BPSK is to square the signal, and generate the second harmonic of the received tone which turns a 180° shift into 360°. Now this has the same phase as for a 0° symbol, so by dividing by two again to generate the original tone an invariant reference results. Any non-linear processing always suffers in the presence of noise, the squaring technique just described is exactly the same as AM detection, and looked at like this, it seems wasteful to degrade a highly efficient coherent signalling mode by having to tack on an inefficient AM receiver to recover the reference. Many implementations of PSK31 use the shape of the amplitude envelope to define the data transition points – another use of an AM detector to steer the otherwise coherent demodulation process.

Differential BPSK (D-BPSK) is often adopted to remove the need to generate a local coherent carrier reference. Here, each data bit recovered depends on the *change* of phase from one symbol to the next rather than its absolute phase, eg. no change = ‘1’, 180° phase shift = ‘0’. Carrier coherence is now only needed for the duration of one symbol to the next, and often implemented with a digital delay line, or just by letting the demodulation software ‘free run’. The downside to differential keying is that if a bit is received in error, then the next one be wrong too.

The accurate frequency setting possible using a GPS Disciplined Oscillator can, with a bit of care in setting up, mean that there is no need to recover the carrier from the signal itself. An example best illustrates this:

Eg. BPSK signalling at 137kHz.

A simple GPSDO on a bad day can give 10^{-9} absolute frequency accuracy, so, assuming both ends achieve this and Murphy's law places the errors in opposite directions, the frequency setting error between both could be as high as $137000 * 10^{-9} * 2 = 0.000274$Hz. We can assume that to resolve a weak BPSK signal with 180 degree phase keying, we need to know the reference phase to an accuracy of 45 degrees. A 0.000274Hz frequency error will result in a drift in phase of 45 degrees in 456 seconds, or a 7.6 minutes. In practice, the frequency error will be much better than this, especially when averaged over longer periods, and it ought to be possible to expect carrier coherence at LF over an hour or so, even when using a simple GPSDO. By defining a fixed character or word length and declaring that the start phase at, say, every minute or ten minute boundary is always zero degrees, even this drift can be corrected before it becomes noticeable and coherent signalling can carry on for ever.

Communications at HF are complicated by the uncertainties in the ionosphere as rapidly changing propagation paths shift the phase randomly and unpredictably, but more complex signalling methods are often used here to compensate; in effect they correct the phase reference on a more regular basis. In any case, GPS will always help the timing issues even if carrier coherence suffers.

Getting the Timing information into the PC

Once we’ve made the decision to use the data from a GPS receiver to assist our digital communications software, how do we get the timing information into the PC? This problem is particularly acute for the 1 PPS signal with its (potential) sub-microsecond resolution? For widespread acceptance of any datamode (at least in the modern amateur radio community) it must not be a requirement to have to construct, or interface, any significant hardware other than the GPS module itself. So a separate processor or logic interfacing hardware is not an option. One route that is possible is the other soundcard channel. Nearly all datamode software uses just the left channel of the soundcard (the tip of the 3.5mm jack) leaving the right-hand channel lying idle. If the 1 PPS signal from the GPS receiver is attenuated down to a level of around 0.5 Volts peak with a pair of resistors, it can be used directly The datamode software can then be updated to recognise the appropriate timing edge, then use this to synchronise the Tx and Rx coding, and possibly control carrier coherency. Furthermore, by doing everything in software, slight tuning errors can be recognised and corrected automatically

This use of the right channel was first used for GPS derived timing by G0TJZ in his chirpsounder monitoring software, details are available from Reference 5. Actual timing information, such as which PPS edge is which, is available from the NMEA data stream which can fed to the PC’s serial port – leaving the datamode software to sort out the correct timing points.

Next generation Beacon design.

Amateur beacons, on all bands, have traditionally tended to use free running oscillators both for frequency setting, and for data and ident timing. This has resulted in a degree of uncertainty for listeners as to the exact tuning point, and what any beacon transmission should be sending. These issues were not usually significant when all listening was done by ear, but now that ultra-narrowband techniques such as waterfall displays are in widespread use, the frequency errors, even on HF beacons, can become embarrassing.

The IARU beacon chain on 10/14/21/28MHz adopted GPS timing from their inception to allow multiple transmissions to occupy a single frequency within each band – each beacon occupies an exact 10 second time slot every three minutes, and the transmission sequence is defined precisely, to align with the UTC second. BUT, the frequencies are not controlled, and many of the beacons within the chain are several hundred Hz in error, and drift, making any attempt at automatic reception infinitely more complex.

Several microwave beacons are now employing GPS locked local oscillators. Usually these are derived from the simple GPSDO design based around the Jupiter-T module, and so the microwave frequency can vary by a few Hz – but the signal can shift this much anyway just due to Doppler shift caused by antenna movements, so its not too important! GB3SCX in Dorset has a transmission frequency of 10368.9050068MHz and GB3SCF, co-located with it, is on 3400.905000000004MHz – or at least would be if the GPS receiver induced errors were zero! The non-exact frequencies (they all should be on xxxx.905MHz) are due to the frequency setting resolution in the Direct Digital Synthesiser (DDS) defining the RF source frequency within the beacon hardware. GB3SCX uses an AD9851 32-bit DDS yielding errors of a few Hz at the final 10GHz output frequency. GB3SCF uses the later AD9852 DDS chip with a 48 bit frequency setting accumulator for a setting error of a few micro-Hz. At the moment GPS timing information is not used for signalling on these beacons, as there is little need for it – yet. An accurate carrier is all that is needed by most microwave operators.

The next generation of VHF beacons will make use of both the time and frequency available from a GPS module. Using a DDS source controlled by a GPSDO delivering micro-Hz accuracy, they can transmit a data sequence using the JT44 or JT65 signalling protocol. These data modes are both a very narrow bandwidth multi-FSK signalling scheme used extensively by the VHF community, for weak signal tropo and EME working. Both are characterised by having to maintain a timing accuracy to within a few hundred milli-seconds to a couple of seconds at each end. While achieving this time setting is no big deal for most manned stations (the radio pips are good enough for setting the PC clock!), its use on beacons means that only a GPS receiver can ensure there is insignificant drift over many months of unattended operation. A programmable DDS driver allows the multi-FSK waveform to be generated directly at RF - the AD9852 will generate up to 80MHz and higher frequencies are achieved by multiplication. So, a single integrated beacon driver can be built that consists of a GPS receiver, a phase locked master oscillator, DDS signal source and a micro-controller such as a PIC looking after everything.

On 26 February a new version of the GB3VHF 144MHz beacon went on-air from Wrotham in Kent. The transmit chain starts off with an AD9852 DDS device to directly generate the RF output frequency (via a doubler) and the controller makes extensive use of GPS to control both frequency and timing. CW, JT65 and BPSK modes are generated directly by the DDS chip, all precisely timed at 30 second intervals. The Frequency is controlled by a GPSDO built around a Jupiter-T GPS module. Further details of the driver can be found in [3]

What is already out there?

Well… Very little at the moment! At this stage of writing few soundcard based datamode software authors have started looking at adding GPS timing options on to their packages, but things will change and several are now talking about it. With the increasing availability of GPS modules it is inevitable that the accurate timing will be used to overcome the system losses inherent in bit timing recovery and carrier coherence. The G0TJZ chirpsounder software has already been mentioned, and is a good route to demonstrating the value of high-precision GPS locked timing.

Bill de Carle, VE2IQ, has added GPS timing control to his ‘Coherent’ and ‘Africa’ BPSK signalling packages, and tests appear to show great promise, particularly in the areas of signal timing and lock-up behaviour.

Beacon hardware is already being slowly migrated to use GPS derived time and frequency, and more will be converted as the listeners expect better and better – particularly at VHF and microwave frequencies. As automatic monitoring and logging software comes into wider use there will be further demand for high stability from beacons – so things can only get better!

References

Ref 1 W5OJM GPS Disciplined Oscillator
- http://www.rt66.com/~shera/index_fs.htm

Ref 2 GPS Locked Frequency Standard for LF Signalling
G4JNT RadCom October 2002

Ref 3 Simple GPSDO - www.scrbg.org/g4jnt/

Ref 4 A Simple Way of Phase Locking Microwave Oscillators
G4JNT, RSGB Microwave Newsletter, April 2004

Ref 5 G0TJZ Chirpsounder Monitoring
- http://www.jcoppens.com/radio/prop/g3plx/tjz/conn.en.php