

The

ZL1BPU GPSCLOCK

USER MANUAL

A **Single-Chip** GPS Disciplined Clock and Reference!

This manual refers to firmware Version 0.3 (GPSCLK3g.xxx)

Murray Greenman ZL1BPU
February 2005

Introduction

This project is a practical application of precision timing. Time and frequency measurement are closely related, indeed one is the reciprocal of the other, and so an accurate frequency reference can make a very accurate clock. This unit contains both a frequency reference and a clock with performance performance.

Accuracy of frequency standards is usually quoted as parts in 10^6 (ppm)¹ or 10^9 (ppb)². The time accuracy of clocks is rarely quoted, but it works out that one second per 11.5 days, or three seconds per month, is the same as one part in 10^6 . Three seconds per year is about the same as one part in 10^7 .³ One second per 100 years is about three parts in 10^{10} . The clock in this project can (depending on the reference oscillator used) provide this level of performance or better! Of course this order of performance is unnecessary in a clock, since it will outlive its owner, and clocks displaying civil time (with or without “Daylight Saving”) require attention twice a year anyway. Such accuracy is however very useful in a frequency standard.

Electronic clocks and watches typically use a quartz crystal reference, oscillating at either 32.768kHz or 4194.304kHz. While the performance of these isn't too bad, typically within a few seconds per week, it is easy and relatively inexpensive to do much better. In the *Superclock*⁴ a TCXO⁵ was used to provide good temperature stability, resulting in an error of a few seconds per year. TCXOs are widely used in UHF radio equipment, such as cellular phones, which need to stay on frequency over extended periods and a wide temperature range. They are also used in GPS equipment, frequency counters, high performance signal generators and other precision equipment. This *GPSClock* project uses a type of TCXO called the VTXO, or Voltage controlled TCXO. While there are better high performance devices to use for precision references, especially the OCXO or Oven Controlled Crystal Oscillator, devices which have less noise and lower “ageing”, the VTXO has the advantage of reasonable availability, modest cost, and very low power. The performance is good enough for all but the most critical applications.

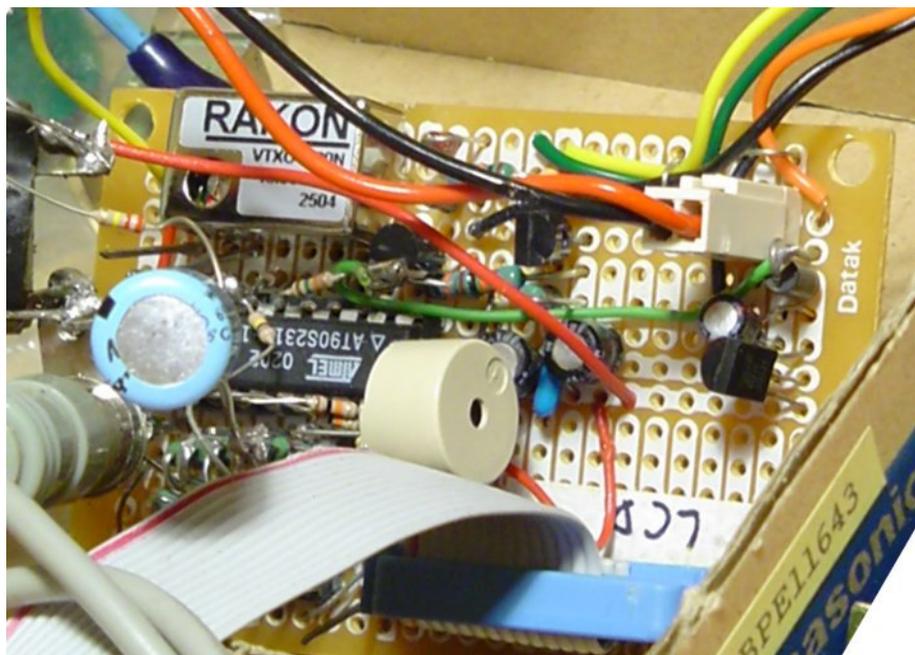


Fig. 1. The *GPSClock* prototype during development

¹ Parts per million and parts per billion

² There is a very good tutorial on Timekeeping at www.allanstime.com/Publications/DWA/Science_Timekeeping/TheScienceOfTimekeeping.pdf

³ See also Appendix “About Frequency References”

⁴ See www.qsl.net/zl1bpu/micro/CLOCK

⁵ Temperature Compensated (or controlled) Crystal Oscillator

The picture of the first prototype (Fig. 1) shows just how little there is to the unit. The project uses an inexpensive micro controller (centre left), a Rakon VTXO (top left), and an LCD display (out of the picture), recycled from an old office telephone. The unit is built on a small Dick Smith Electronics H5608 project board. As well as UTC time⁶ and local time, the **GPSClock** provides three reference signals, a 1 kHz 50% square wave and a 1 Hz (1pps) pulse 1 ms long from the micro, and a 10MHz signal from the VTXO. These can be used as references for other devices. The **GPSClock** also provides time and other information as a message via an RS232 serial port. This information can be used in the calibration procedure.

Long term accuracy of references such as this depends on either routine or automatic calibration to some external reference, which must be at least one order⁷ better in performance. The **GPSClock** uses continuous compensation to remove long term reference errors, and also to remove small daily variations.

This unit automatically calibrates itself to a simple external reference 1Hz pulse, (commonly known as the **1pps pulse**) from a GPS receiver. Some (but by no means all) GPS receivers have this option. The pulse is 5V logic level, and rises from 0V at the exact moment of the UTC second, within typically 100ns. It drops back to 0V from 5V some time later, typically after 25ms or so. It will also operate equally well from other precision sources, provided the source is an exact integer fraction of 1kHz⁸.

While each individual pulse may have a little jitter, as time accumulates with successive pulses, the jitter is averaged out and time is accumulated with great accuracy. The technique of locking a high frequency oscillator to 1pps is known as a **GPS Disciplined Oscillator**. It is not a simple technique to implement, for a range of technical reasons, and equipment of this type is typically very complex and expensive. Now, for the first time, you can have a **SINGLE CHIP GPS Disciplined Oscillator**, with all the complexity inside the chip.

When you build this unit, you will also need a suitable GPS unit and antenna. Most of the suitable GPS units with 1pps are of the "OEM" type, little modules that are designed to be built into other equipment. The author has used the Rockwell Jupiter and the CMC Superstar, but other modules such as the Trimble Lassen SQ⁹ and the Ashtech A12¹⁰ (pictured below) would be equally suitable¹¹. The GPS unit and its antenna will probably cost more than the rest of this project. Fortunately there are many other uses for these modules.

You will be more than impressed by the performance of the **GPSClock** - it always tells the correct time, down to fractions of a second, and is unaffected by temperature or power glitches, and even keeps good time when the GPS reference is lost. Plus, it uses only one inexpensive chip!

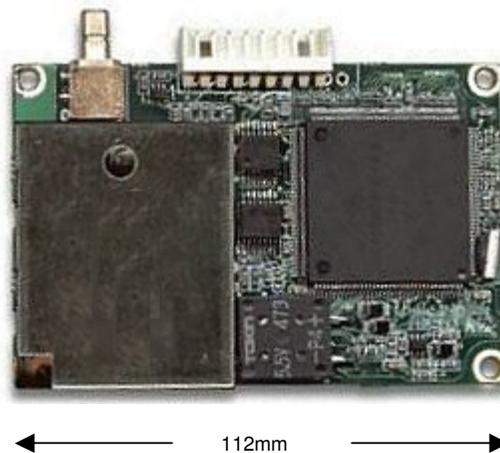


Fig. 2. A typical GPS module, the Ashtech A12

⁶ Universal Time Coordinated (most people will know it as "GMT")

⁷ One power of 10

⁸ It therefore works with 1000Hz, 500Hz, 250, 100, 50, 25, 10Hz etc

⁹ See article by Dale Hughes VK2DSH in Amateur Radio, June 2004 (WIA)

¹⁰ See products.thalesnavigation.com/assets/datasheets/A12receiver.pdf

¹¹ If you have a choice, use a unit which turns off the 1pps pulse when the GPS fix is lost – the Rockwell Jupiter does not, and results in the clock following the unlocked GPS unit rather than going into "hold".

spends most of its time asleep, and the average current drawn from the power supply is low, just 10 milliamps, including the LCD display and the VTXO.

Power for the **GPSClock** is provided by a small AC adaptor, and is backed up by a small NiCd or NiMH battery (which can be a recycled cellular phone battery), to ride out power failures. The unit should operate for three days from typical batteries. Since the current drawn is so low, the clock can also operate from the low current provided by the serial port of a PC (connect a diode from J2 pin 4 to J4). The battery pack is charged slowly by a resistor across the battery's isolating diode D3 (see Fig. 4).

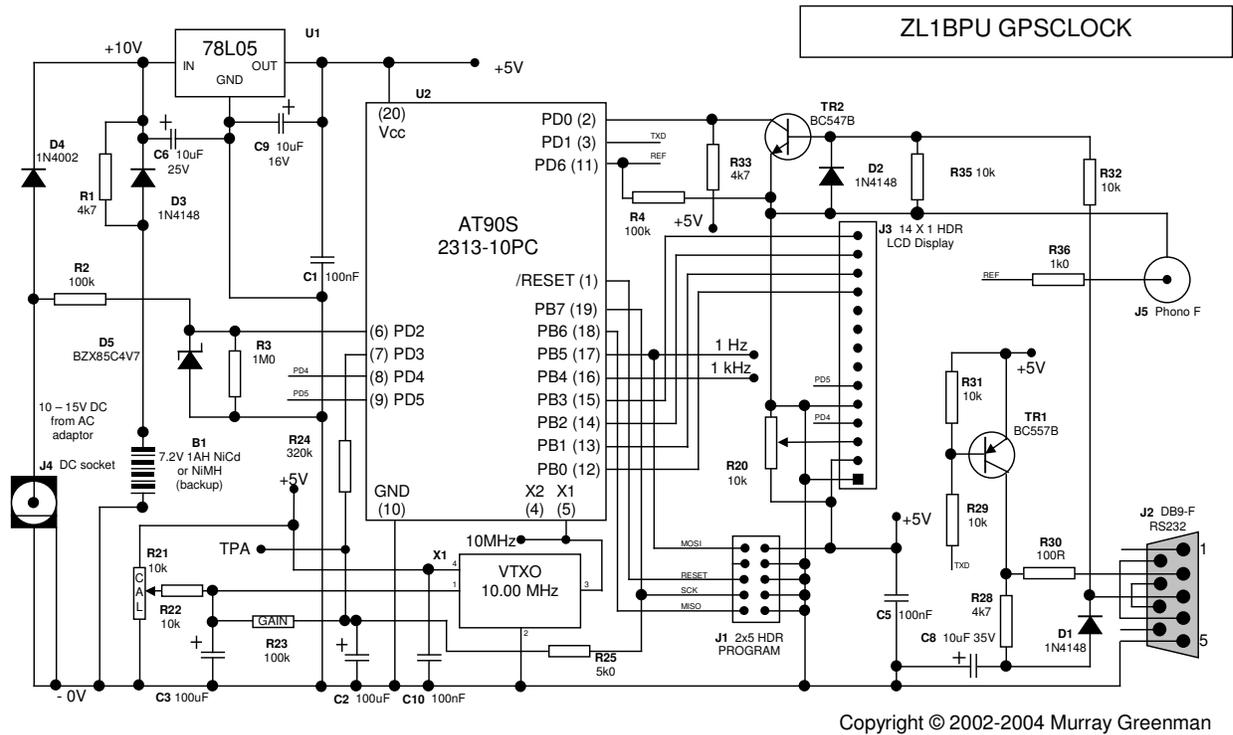


Fig. 4. The **Superclock** schematic

In the schematic, note the power supply arrangements on the left, with the diode isolated backup supply and AC supply, which consists of a simple AC adaptor or plug-pack. When AC power fails, the input micro input PD2 (pin 6) goes to zero, and the micro displays “BATT” on the display. If an external DC battery supply is used, it might be useful to know when the supply drops below the back-up battery rather than when it drops to zero, so to achieve this, replace R2, D3 and R3 with a comparator.

Next in the schematic are the 5V regulator U1, and the micro U2 with the VTXO X1 below. The micro is an inexpensive ATMEL AT90S2313, available in New Zealand and Australia from DSE, Polykom or Jaycar, and in the USA from Digi-key and Mouser. To the right are the RS232 interface, TR1 and TR2, the LCD connector J3 and the programming header J1. A discrete transistor RS232 interface is used to reduce supply current drain. Current is drawn from the supply only when transmitting data.

The REF input J5 is where the 5V-level 1Hz (1pps) reference pulses from the GPS receiver enter. If the GPS is disconnected, it is important that the input be grounded to prevent noise-generated interrupts, and thus false phase measurements. This is achieved by R4, while R36 provides ESD¹³ protection. The use of this GPS phase detector is described later. Other spare pins are outputs and can be left open.

There is of course considerable similarity between the circuit of the **GPSClock** and previous projects, especially the **Superclock**. See Fig. 5 for a detailed view of the circuit board, part way through assembly. The LCD connects at the top left. Display routines have been specially written for operation at up to 15 MHz.

¹³ Electrostatic Damage – the series resistor limits the energy dissipated in the chip protection diodes when static electricity appears on the input. All external inputs are protected in this way.

The trim pot top centre (Fig. 5) sets the LCD contrast. To the right of the pot are the programming header and some unused resistors, left over from a previous project. The micro is centre right, and the power and serial connections lower left.

The two-pin connector bottom left is for the backup battery. The wiring underneath is a combination of solder-bridged pad connections and short lengths of fine insulated wire. The two centre tracks under the micro are 0V (lower) and +5V power (upper).

The same low cost RS232 interface used in other projects is in the lower centre of the board. At the far left centre is the little 5V regulator. The silver box lower right is the VTXO, and its adjustment trimmer can be clearly seen. This reference has through-hole pins, making it easier to mount on the board than a surface-mount device.

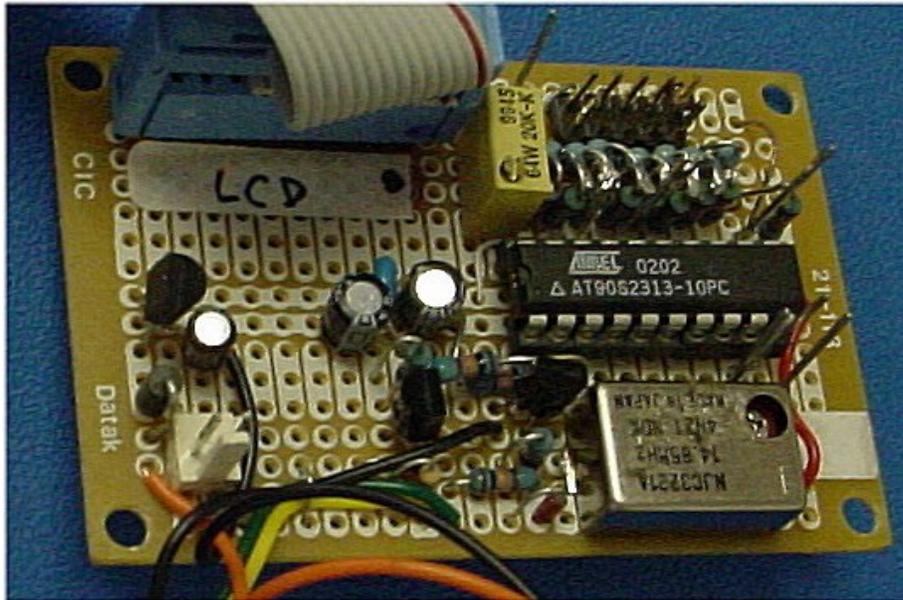


Fig. 5 Circuit board detail (partly assembled)

One interesting feature of the *GPSClock* is the way in which time is counted. Unlike most micro controller projects, which operate in binary (e.g. a frequency counter or digital voltmeter), this clock operates in BCD,¹⁴ in other words, it counts seconds 0-9 and from 0 tens to 5 tens. In fact, the data is stored as "packed BCD" where the information about seconds, minutes and hours are stored in one register each.¹⁵

Every time a mathematical calculation is made, the result must be checked or adjusted to ensure that it is true packed BCD. Seconds and minutes are also checked to see that they do not exceed 59, and a carry and correction made if they do. This may seem a clumsy way to count, but it has the advantage that displaying the result is very straightforward - displaying both nibbles of the register as two ASCII characters simply displays 00 to 59. Time is displayed as "HH:MM:SS UTC" (see Fig. 6).

¹⁴ Binary Coded Decimal

¹⁵ Packed BCD keeps two decades in one byte. In this case, units (right 4 bits) and tens (left 4 bits).



Fig. 6. The LCD display

An extra check is required after adding the hours, to return the count to zero when 24 is reached. Further checks are involved in the addition of the local time offset, and allow the offset to be either negative (west of GMT) or positive (east) with accurate results. For example, for New Zealand Daylight Time the offset is +13 hours. It might seem no big issue, but performing calculations like this in BCD using machine code is not always straightforward.

The local time is displayed as "HH:MM am" (or "pm") followed by $\pm nn$, the local time offset. For negative offsets, the "2's complement" of the number is stored; "FF" for -1, "FE" for -2 etc., but the user need not worry about that, as adjustment is made using only "UP" and "DOWN" commands.



Fig. 7. The 2x24 LCD display with "engineering" data

Depending on the use intended for the **GPSClock**, the user can choose to have a 16-column display (Fig. 6), and simply display time, or the larger 24-column display (Fig. 7), and show the frequency reference management data as well. In Fig. 7, which shows the **GPSClock** operating with a 24 column display, you can see extra data on the screen to the right. The first 16 columns are exactly the same as appear on a 16-column display (as in Fig. 6), but more information about the status of the **GPSClock** frequency reference is visible to the right. There are four blocks of information; Reference Phase, Status & Mode, Feedback Control Voltage, and various Status Announcements. In Fig. 8, these are identified.

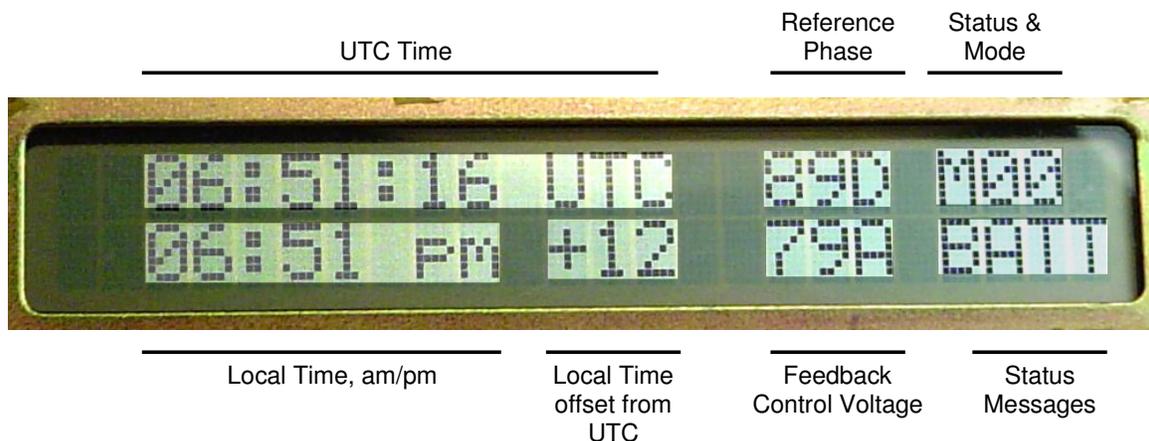


Fig. 8. The display data fields are identified

The Reference Phase is a 12-bit HEX value giving the current 10MHz Reference phase relative to GPS. It is in fact a measure of the count of 10MHz RF cycles at the last GPS pulse, with the most significant bits omitted (these are irrelevant for control purposes).

The Feedback Control Voltage is the 12-bit HEX value sent to the PWM D-A converter, and represents the Control Voltage in volts. The actual Feedback Control Voltage output will be close to $5 \times (1 - \text{Feedback}/4096)$ Volts.

The Status and Mode data has two parts, (hey, Status and Mode, well done!). Ignoring the “M”, which is simply “M” for “Mode”, the first character is the Status, which contains three bits of information:

POWER GOOD	Bit 7	Zero if power has been lost (set with first Mode command)
BATT OPERATION	Bit 6	Set when AC power has failed (and unit runs from battery)
GPS OK	Bit 5	Set when GPS pulses arrive regularly

The second character displays the operating mode, in the range 0 – 7. The Modes are set by the user using the “C” Command. There are eight operating commands. Those listed in **GREEN** are phase locked, while the others are used only for setup. These commands also set the “POWER GOOD” bit.

- C00** Low gain, low noise phase locked control mode, with shift register. (Default mode)
- C01** Medium gain, low noise phase locked control mode, gain=5.
- C02** High gain, quick-locking phase locked control mode, gain=15.
- C03** Minimum frequency, used for range checking.
- C04** Centre frequency, used for reference calibration.
- C05** Maximum frequency, used for range checking.
- C06** Ramp up from current point, used for A-D checking.
- C07** Ramp down from current point, used for A-D checking.

The secret to Mode C00 is the DSP technique using a 16-bit shift register, which acts as a high-tech phase accumulator. It allows phase comparisons to be made 32 seconds apart, rather than just one second apart, meaning that much higher gain and phase resolution are possible. Because a shift register is used, a new 32-second result is available *every second*, important for accurate and stable control. With a 10MHz reference, phase resolution is thus in effect about 3ns, not 100ns as you’d expect. The gain of Mode C00 is user-adjustable using the integration time function.

Modes C01 and C02 lock quickly (just a few minutes) but not with any great accuracy. They are provided mostly so you can see the difference between a high-tech phase accumulator and conventional proportional control as used by others. The fastest way to achieve lock is to use C04 for a few seconds, to centre the frequency close to the right point, then C00 to lock. If you can, watch the fascinating process on the PC using the RECORD program and/or an HF receiver and Spectrogram. If lock is lost and the unit left to recover lock in mode C00, depending on the integration time, it takes about 5 - 15 minutes.

Improved gain can be achieved by decreasing the integration time.¹⁶ If the reference performs better in Mode C01 than C00, or with low integration times, i.e. with higher gain, it might be better to adjust the GAIN setting resistor R23 lower in value. Some noise is added to the reference by the feedback system, so lower gain and longer integration times inject less noise. Thus the gain and integration setting are always a compromise between the ability to correct the oscillator’s own noise and inducing more system noise.

¹⁶ Details of the integration value in the C00 command are found on page 13

precision, by listening to them both at the same time with a pair of headphones. The only problem here is that there is a marked propagation delay in the radio signal arrival. Frequently WWV, WWVH and other timer services will be heard giving multiple ticks due to different propagation times. In Fig. 10, both WWVH and WWV ticks can be identified, having 30 and 45ms delay respectively after the **GPSClock** 1Hz pulse.

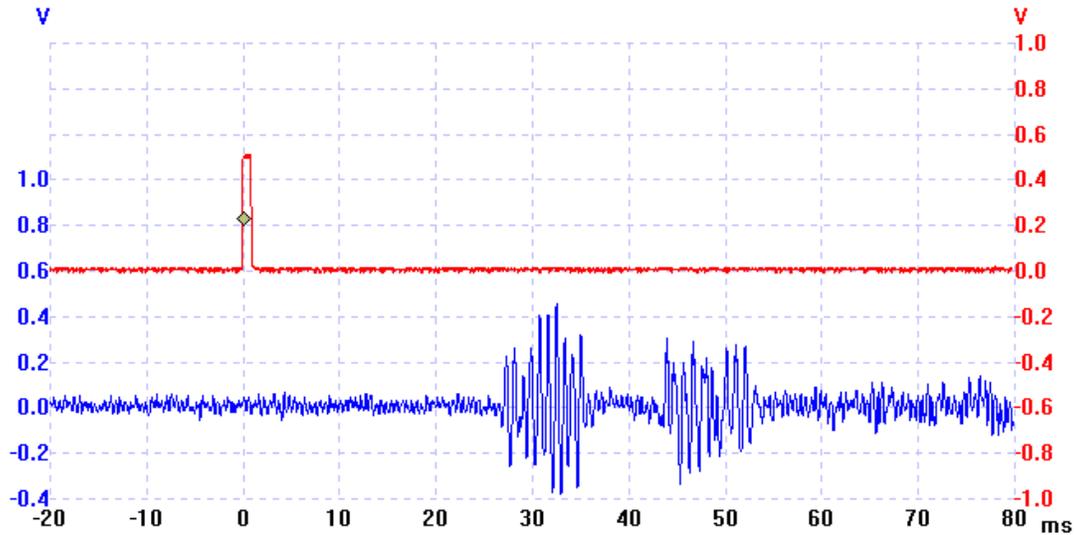


Fig. 10. The **GPSClock** pulse (red) compared with WWVH and WWV (blue)

If your reference is not 1pps (say it's 1kHz), you can't use the "X" command to reliably synchronize seconds. As a last resort, simply adjust with the "A" and "R" command so that the seconds tick lines up visually with WWV (Fig. 10). If the "R" command is used too much, causing overshoot, another 100 presses are required to get close to the correct seconds phase again (and time will slip one second). The "A" command works very slowly, and due to the timing relationships in the clock, will not advance time every time you send the command. 1ms is the finest resolution possible because this is the smallest time interval counted.

After checking at the minute event, use the "+" or "-" command to correct the seconds display. It should change to "00" exactly as the WWV minute tone sounds. The ">" and "<" commands to adjust local time can also be used any time, which is convenient, so stepping forward or back for "daylight saving" is a breeze. These four commands, plus the "A", "R" and "X" commands are the only ones permissible while the **GPSClock** is keeping time. They do not normally affect time keeping at all, although slippage can occur if you happen to send a command just as a second event occurs. The slippage will be less than 10ms. Finally press "X" to resynchronize in case there was any slippage in setting.

The program source code for this project is available from the author at a modest charge, so the constructor can make whatever modifications are desired. These are best made in the main loop where timekeeping will be unaffected. There is plenty of code space, so small adaptations could for example provide:

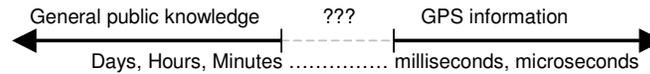
- Time signal "pips", or even time codes as used by time stations (WWV, DCF77, JJJ etc)
- Additional or alternative reference outputs, such as 440 Hz, 50 Hz or 1 hour
- Time announcements (in Morse!) on the hour, or even voice announcements
- An alarm or time controlled output function
- A timer or stopwatch
- A clock that keeps astronomical time, or a tide clock
- Push-button controlled display back-lighting with adjustable brightness

Front-panel controls could be added, provided they were scanned in the main program and did not stop the clock during their action. Using front panel controls for time setting is not recommended, for obvious reasons. The executable code is designed for any reference frequency from 5 to 15 MHz that is a multiple of 2 kHz. The only changes necessary for a different VTXO frequency are made in the EEPROM during setup. Five parameters are stored - the frequency division ratio in two bytes (divide the VTXO frequency in Hz by

2000, then subtract one and convert to HEX), the serial port baud rate (also depends on the reference frequency), the default local time offset, and the default error feedback integration time.

Notes about Setting Time

It is an interesting fact of life that while time is widely known to within a minute (as every broadcast station will tell you), and is easy to discover in a traceable manner to within minute fractions of a second (using the GPS 1pps), actually knowing **which second is which**, i.e. which UTC second the 1pps pulse refers, to is not so easily determined. In other words, there's a grey area in the time setting business between one minute and fractions of a second.



For a start, time disseminated by GPS is **NOT** UTC – it is GPS time, as defined by the US Navy, which is an accumulation of seconds since 1980 without the UTC leap seconds¹⁷. Fortunately the length of GPS seconds is closely controlled to be the same as UTC seconds, and the time offset from UTC is transmitted with the GPS data.

Even though the GPS – UTC time difference is broadcast, there's still the problem of latency through the GPS receiver to the NMEA data stream, through the serial communications link, and finally through the PC to the display. Even if you use a stand-alone GPS receiver with time display, the time display is usually unreliable to better than one minute. The actual GPS seconds pulse sent by the GPS unit is very good – even with propagation variations, it is likely to be within 1us, no matter what GPS receiver you use, and is frequently much better.

In reality, the only way to know for sure which UTC second a 1pps tick refers to is to use a UTC time broadcast, such as WWV, a telephone voice time service (speaking clock) or a local radio station which broadcasts time pips on the hour. WWV is reasonably painless to use because it transmits continuously, and provided propagation is good enough, you get a clear "beep" on the minute which identifies second 00. To use hourly pips from a broadcast station you need to be more patient – they only happen once/hour.

WWV and other services also transmit time codes, which are another useful way to identify the exact minute. Using software such as RADIOCLOCK, you can receive the time codes and set the PC clock to UTC. The PC time will then be close enough for perhaps an hour or so. There are also equivalent internet services, which (if used with care, remembering that propagation delays can well exceed one second) can also be used to set PC time and identify seconds.

Some cellular phone services, corporate computer networks and other data services send time to the user. Once again, the only way to be sure that the time they send is traceable to UTC is to check it yourself against a known traceable source. If possible, stick with WWV for a sure fire method, and check other identification methods against WWV. The propagation delay from WWV will rarely exceed 100ms¹⁸.

Remember that **GPSClock** is a portable device – just disconnect the GPS input, replace the AC supply with a 12V battery, and away you go! It will keep time accurately for months without the GPS connection, so you could transport the unit to another traceable clock, set it there, and then return home.

¹⁷ See <http://gpsinformation.net/main/gpstime.htm>. Also, *The Science of Timekeeping* by David Allan and published by Hewlett Packard is well worth reading. See http://www.allanstime.com/Publications/DWA/Science_Timekeeping/TheScienceOfTimekeeping.pdf

¹⁸ See Fig. 10

Calibration

Setting Up

The purpose of the calibration process is to set the natural operating frequency of the VTXO or V-OCXO, which allows the correction circuitry to have maximum range, and therefore the longest interval between calibrations. For a first-time adjustment, it is also important to check the operating range of the VTXO, since not all devices have the same frequency control sensitivity. Both adjustments are made when the unit is first assembled, and the natural operating frequency is also best checked every six months or so. The time between rechecks can then be modified according to the observed ageing rate of the VTXO. *Setting up must be done before the time is set, as it upsets time keeping.*

The GPSClock is self-calibrating – no special equipment is required for setup.

The process is easy, and requires only the GPS receiver, and a PC with nothing more than a Terminal program, (or preferably the special monitoring software **RECORD**, supplied with the **GPSClock** firmware). If you use a 24-column display, you don't even need the PC, although watching the phase plot is half of the fun.

With the GPS receiver running and giving a reliable fix, connect the 1PPS output (which must be positive going 5V logic level on the UTC second, DC coupled, and preferably 10µs to 100ms wide) to the **GPSClock**, connect the clock to the PC comm port via a serial cable, and run a terminal program on your PC (Windows Terminal, Hyperterminal, even DOS Term) at 9600-N-8-1. Check that telemetry is being received from the unit. Type the command **C04** on the terminal, and note that the last digit of the telemetry on the PC screen display shows a **4**. (It also shows at the top right of the LCD screen on the 2x24 display).

VTXO Centre Frequency

The next action is to adjust the 10k trimpot R21 so the frequency is exactly 10.000MHz. Rather than use a receiver or frequency counter, which will likely be inaccurate, we use the GPS and the built-in phase detector! Watch the first four hex numbers after the time in the telemetry, or the phase on the (2x24) LCD display. If they increase in value with each consecutive reading, lower the frequency. If they decrease, increase it. You will need to adjust in tiny steps very slowly once you get close, as the changes may not happen for many seconds. A change of one every 10 seconds is good enough. You can also adjust by watching the phase on the smaller graph in the RECORD program. This adjustment does not affect the final accuracy, only the calibration operating values, and how long the unit will survive between calibrations.

```
10:44:34 %0123456504
10:44:35 %0122456304
10:44:36 %0122456104
10:44:37 %0121455804
10:44:38 %0121455504
10:44:39 %0120455204
10:44:40 %0120454904
10:44:41 %011F454704
10:44:42 %011F454604
10:44:43 %011E454504
10:44:44 %011E454404
10:44:45 %011D454304
10:44:46 %012C454104
```

The data above is an example of telemetry received on the PC during frequency adjustment in mode C04. Remember that there is a new line of telemetry each second. See how the **PHASE** (artificially coloured **RED**) is slowly decreasing in value, about every other second (i.e. with an error of about 0.5Hz), nearly close enough for this adjustment. Note also that the last digit (the Mode) is **4** (artificially coloured **GREEN**).

Operating Gain

To check the operating gain of the VTXO, watch the phase on the LCD display or telemetry. Send the command **C03**, and check the frequency. (In other words, how big the steps in **PHASE** are each second - the above example is about 0.5Hz). Then repeat with **C05**, and check again. The range should be about +/- 2Hz to 4Hz at 10MHz, and reasonably balanced about the correct frequency. By watching the phase on the computer or LCD display, you can estimate by how many the numbers change per second (4Hz means the

PHASE changes by four each second). If you have the equipment, you can easily check the gain using a very stable communications receiver and Spectrogram PC program (see next section).

You can change this frequency range by changing the GAIN resistor R23. Higher gain (lower resistor) gives wider control range, but poorer noise performance. +/-4Hz range represents only about 0.5ppm range, and you may need to reduce the gain resistor if the clock experiences wider than normal temperature range. If you do change R23, it will be worthwhile repeating the VTXO Centre Frequency adjustment.

It is important to have enough gain to correct the VTXO at its operating extremes, otherwise lock will be lost.

Accurate Ramp

Next you should if possible check that your feedback components are accurate, and that the 4096 steps provided are linear, with no unexpected jumps. Use a very stable communications receiver, and a spectrogram program such as SPECTRAN¹⁹ to monitor the signal at 10MHz. Set the width of the spectrogram to 20Hz, and the speed of the spectrogram as slow as possible. Enter the command **C04** to centre the oscillator. Find the signal on the receiver, and centre it on the spectrogram screen using the spectrogram adjustments. Reduce the spectrogram span to 5Hz, ensuring that the signal stays centred. Monitor for 30 minutes or so to ensure that the receiver is stable. Then send the command **C06** (ramp up). This will make the oscillator frequency ramp up each second, one step at a time. Continue for 10 minutes (about 600 steps), and check on the spectrogram that the ramp is smooth and straight, with no steps of jumps. (Make sure that the jumps are not from the receiver!) Send the command **C07**, and watch it come back down. Continue past in the opposite direction and check again. High performance depends on this highly linear response.

Without a stable receiver and spectrogram, you could manually plot the phase response. Write down the phase value at a constant rate (say exactly every minute) for the duration of the above test, convert the values to decimal, and plot the results. There should be a straight line up and down again. A spread sheet program with HEX maths capability will make this process moderately painless.

Setting the GPSCLOCK into Operation

If all is well after the above checks, send the command **C04** to centre the oscillator, then a few seconds later command **C00** (gain of three), and you'll be in business! The oscillator will lock in just a few minutes, and the phase will no longer change once it is locked. If you are impatient, and want a faster lock, try **C01** (gain of four) or **C02** (gain of 16), and then after a few minutes revert to **C00**. The higher gain settings cause more noise, because they use larger steps, but have faster lock. The default mode (**C00**) is optimized for minimum noise, but will not lock quickly and will not track excessive reference drift.

Once the reference is operating correctly, set the clock (see previous section) and you're done. Just set and forget!

Integration Time

Because in mode C00 the phase comparison can be made over a long period, the gain is very high. Each phase comparison results in a signed (+ or -) error value which is added to the VTXO control voltage. Adding these phase errors every second would result in extremely high gain, and probably to instability, since the phase correction would take place faster than it can be measured. The gain of this feedback process can be adjusted by the simple means of reducing the frequency at which the VTXO control voltage is updated by the phase comparison result. This is called the *integration time*, and is set as the first value in the Cn0 command (up to now referred to as "C00"). The time in seconds is related to the value "n", which is a hexadecimal number 0 – F.

$$\text{Integration time} = 2n + 1 \text{ seconds}$$

Very high performance reference oscillators (double-ovened high performance OCXOs) can benefit from values as high as $n = C$ (25 seconds) or more, since they are very stable. Simpler less stable oscillators such as VTXOs will normally work best with $n = 4$ (9 seconds) or less. The setting is a compromise between correction of short-term jitter (small n) and accuracy of control (larger n). Check your oscillator performance with a Spectrogram and adjust n for the narrowest trace width. Fig. 17. clearly shows the difference in stability and noise performance of two different references at 10MHz.

¹⁹ Obtained from <http://www.weaksignals.com/>

Although system gain is affected by the integration time, it is important to still have the hardware gain set so that the oscillator can be controlled correctly accurately over its whole range – don't just rely on the software integration setting. The integration time also affects all the other modes, so for example if you set C46 (rather than C06) the VTXO control voltage will ramp up every nine seconds, rather than every second. Remember, the integration time is the **first** character in the command, and the mode itself is the **second** character!

Routine Calibration

Every six months or so (more often at first, less often as you learn how much the oscillator ages), you will need to check the natural operating frequency of the reference. ***It is important to do this without stopping the clock or losing any time.***

The process is therefore quite different to initial calibration, since it must work without changing mode (and thus upsetting the timekeeping). With the GPS receiver running and giving a reliable fix, and the 1PPS output connected to the **GPSClock**, connect the clock to the PC and run the terminal program. (If you use a 24-column display, just read the value off the display – the PC won't be required).

Note the Feedback Control Voltage. If the value is significantly different to 0x800 (say below 0x400 or above 0xC00), adjust the 10k trimpot R21 **slightly** one way or the other, wait 10 minutes, and then note the value again. If it is closer to 0x800, you're going in the right direction. If not, make a similar change in the other direction. (Mark the trimpot direction so that you will know which way to go next time). Make one or two further adjustments, each time leaving at least 10 minutes before reading the Feedback Control Voltage again (this allows the adjustment you made to be corrected). This process may take half an hour or so, and can be most conveniently done as you change the daylight saving offset (i.e. every six months). No clock time is lost or gained provided you stay away from the **H**, **M**, **S** and **C** commands.

The idea is to move the Feedback Control Voltage value to the opposite end of the control range. For example, if the value when you started was 0x06FE, then aim to set it to around 0x0A00 or thereabouts. The actual value is not important, as accuracy is unaffected, but offsetting it the other way maximizes the time between calibrations.

In between routine calibrations, you can assess the reference calibration status by monitoring the Feedback Control Voltage. It is a good idea to write this value and the date on a sticker on the back of the clock every month or so. The nominal value when calibrated is 0x800 (800_{hex}), and the range is 0x000 to 0xFFFF (0 to 4095 decimal). So long as the value stays near the middle of the range, all is well. If at calibration time the Feedback Control Voltage is near 0x800, then you do not need to calibrate at all. If the ageing rate of the VTXO is very high (or if you used an ordinary crystal oscillator), and you forgot to check the change in Feedback Control Voltage, it is possible for the control range to be exceeded and lock will be lost. If this occurs, repeat the initial calibration and then reset the time.

Remember that the calibration **ONLY** affects the Feedback Control Voltage, and the available control range. By manually adjusting the median voltage applied to the VTXO, you are maximizing the electronic control range from the micro. You are not adjusting the accuracy of the clock or its reference, which are assured by the GPS. With GPS connected there is **NO ERROR**²⁰, and even when the GPS is disconnected, the error will be very small, since the **GPSClock** remembers the Feedback Control Voltage it used when the GPS was last connected.

Phase Comparison

Much the best source of phase reference is the GPS receiver 1PPS signal, as assumed throughout this manual. There are other suitable references, and the **GPSClock** is able to lock to a remarkably wide range of signals. Provided the signal is a 5V pulse, it will lock to 0.1, 0.2, 0.5, 1, 2, 4, 5, 8, 10, 16, 20, 25, 50, 100, 200, 250, 400, 500, or 1000Hz! Thus the output of some other stable reference source (such as a Rubidium Standard) can be used. Few other sources will have the long term accuracy that the GPS can provide.

Some television networks use an accurate Rubidium-locked or GPS-locked timing generator for some or all their reference timing. PAL TV is a convenient and much under-rated as a source of precision frequency. The colour subcarrier (4433.61875kHz), line frequency (15.625kHz), and frame frequency (50.00Hz) are

²⁰ Unless the oscillator has an excessive ageing or thermal drift rate, which is why a VTXO or better is used

typically all locked to a common reference, which can be 1 part in 10^{10} accuracy or better. There remain two problems with using TV as a reference source: (a) knowing which network to use, when the network is using its reference, and when is it not; and (b) receiving an accurate local version of the pulses, given the various propagation problems. For the purposes of the **GPSClock**, the most appropriate (and generally the most stable) signal is the 50Hz frame frequency²¹.

The well-respected LM1881 sync separator chip²² is an ideal source of the TV frame frequency 50 Hz reference. Use the video output of a TV receiver or video recorder to drive the chip. Operate the LM1881 from the **GPSClock**'s 5V supply, and feed it with TV composite video from a Rubidium or GPS referenced TV station, and this simple little 8-pin chip will generate very stable and precise pulses with the correct timing and levels to drive the J5 input of the **GPSClock** directly. See Figs. 11 and 4.

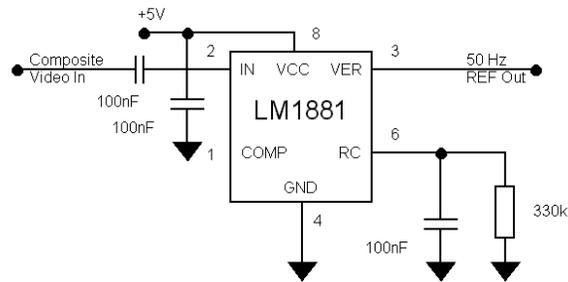


Fig. 11. 50 Hz Reference from TV Sync

The line sync output of the LM1881 is not useful for phase detection purposes in this design, for two reasons. First, it is a composite sync output, which contains twice-frequency equalizing pulses and frame pulses; and second, the **GPSClock** is designed for low frequency references (1kHz or below).

²¹ This option is generally not available in countries using the NTSC TV standard, where, even if locked to a suitable reference, the TV frame frequency is close to, but not exactly 60Hz – it's 59.940059940059940 Hz! In the PAL system the frequency is exactly 50.00Hz.

²² See <http://www.national.com/ds/LM/LM1881.pdf>

Serial Data

Commands are sent to the **GPSClock** in ASCII format, RS232, 9600 bps, N-8-1.²³ Lower case is interpreted as upper case. The commands are of two types – those that stop time keeping, and those that do not. Those that stop time keeping cause the **GPSClock** to stop until sufficient data is received for the command to be processed, after which time keeping is resumed. In all cases two extra characters are expected. These are the **H**, **M**, **S** and **C** commands.

Commands that do not stop time keeping do not need the **GPSClock** to wait for data – these are the single character commands **+**, **-**, **<**, **>**, **A**, **R** and **X**.

The clock transmits four separate data products. During normal operation, the UTC time, the reference divider phase, the feedback control voltage and mode are transmitted. The message format is as follows:

HH:MM:SS %PPPPFFFFMM<CR><LF>

Where **HH:MM:SS** is the UTC Time, “ ” is a space character, “%” is the **GPSClock** model identifier, **PPPP** is the Reference Phase, **FFFF** is the Feedback Control Voltage, and **MM** is the Status and Mode data. Other (debug or engineering) characters may follow between **MM** and **<CR><LF>**.

Whenever a command (one which stops time keeping) is sent to the **Superclock**, it replies with a message. The data format is also 9600-N-8-1. The message is also terminated by **<CR><LF>**.²⁴

UTC Time

The UTC time message is transmitted every second, starting just after the second event. The format is **HH:MM:SS**, in eight bytes (characters 1 – 8) and is followed by a space. The time sent is the same as shown on the LCD screen. It would be quite practical to build and use the **GPSClock** without an LCD display. The full time sequence is available within 10ms of UTC time. The clock can be advanced in 1ms steps so that this message can be received on time if an application requires it.

Reference Phase

Following the model identifier, the reference phase is transmitted as four ASCII characters **PPPP** (characters 11-14). The value can be from 0x0000 (zero_d) to 0x1387 (4999_d) assuming a 10MHz reference frequency is used. The maximum value is the same value programmed into the EEPROM for the reference frequency divider during setup. This phase value can be manually tracked, or using a simple BASIC computer program, graphed over time. The value is the same as seen on the 24-column LCD screen, where only the last three characters are displayed.

Feedback Control Voltage

Following the phase, the Feedback Control Voltage is transmitted as four ASCII characters **FFFF** (characters 15-18). The value can be from 0x0000 (zero_d) to 0x0FFF (4095_d), and directly represents the feedback pulse width, which is $(1 - \text{FFFF}/5000) \times 100\%$. Hence the actual DC control voltage is $5 \times (1 - \text{FFFF}/5000)\text{V}$. The value is the same as seen on the 24-column LCD screen, where only the last three characters are displayed.

Status and Mode

Character 19 is the Status byte, in HEX-ASCII, containing the three status bits outlined on page 6. Character 20 is the Mode byte, and has an ASCII value 0 – F representing the current mode. The table of modes is shown on page 8. Both values are the same as seen on the 24-column LCD screen. Other undocumented characters may follow.

Command Responses

When a command that suspends time keeping is received, the UTC time and reference phase message sent every second is suspended along with time keeping. These commands (**H**, **M**, **S** and **C**) are echoed back to the PC, along with the command data, in the same format as the commands.

²³ No parity, 8 data bits, 1 stop bit

²⁴ “Carriage Return” (ASCII 0x0D) and “Line Feed” (ASCII 0x0A)

If a command is sent to the **GPSClock** which is not understood, or if the data accompanying the command is not understood, the message is either ignored, or the clock sends the error message "?", followed by CR/LF. Time keeping then continues, although a time-keeping delay will have been incurred. Commands that do not suspend time keeping (+, -, <, >, **A**, **R** and **X**) do not invoke command responses.

Programming and Setup

This procedure assumes the constructor uses the same programming tool as the author.²⁵ Since this is free, it's a reasonable assumption! If other tools are used, familiarity with them will be required, as the following procedure is specific to ISP V2.65.

The following are required:

- The ATMEL ISP AVR programmer, ISP.EXE V2.65 or similar.
- A programming cable.²⁶
- The **GPSClock** firmware, available from the author.
- A PC running Windows 3.1, 95 or later to run the programming software.

Connect the DB25 end of the programming cable to the PC printer (parallel) port, and the 10-way header to the **GPSClock** programming header. Make sure the cable is the right way round on the header – no damage will be done if it is wrong, but nothing will work.

If two computers are available, or the computer can run two applications at once, connect up the **GPSClock** serial port to a PC serial port. You will need a “modem” serial cable (pin-to-pin with no crossover), NOT a “null modem” cable. Run a Terminal Emulation program (Windows 3.1 Terminal is ideal), setting it to the correct port, and to 9600 bps. This will allow progress to be monitored during programming and later allows the reference to be calibrated and the clock to be set.

Run the ISP software ISP.EXE. There is also a DOS version; the most recent version supports 32 bit operating systems. The author recommends V2.65, which will work on a 486 PC running Windows 3.1, as well as later computers, and is required for the parallel port programming cable. Make a project (from the menu, **Project/New Project**), selecting AT90S2313 as the device.

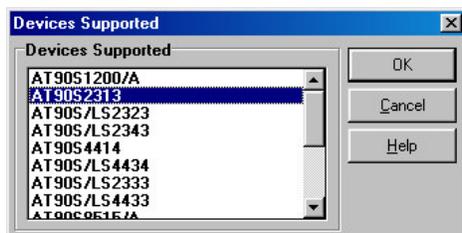


Fig. 12. Select the device

Press OK and then on the “Manager” tab of the next screen, give the project a title, and then from the main program menu select **Project/Save Project** to save the project file.

The next step is to check for communications with the micro. Select **Options/Change Printer Port...** and a “Port available” message will be observed. If all is well, select **CANCEL**. If not, try changing the port in case it isn't LPT1.

Next, select **Program/Read EEPROM**, which should be something safe to do (it won't matter since the micro is already blank). If all is well, the data from the micro will fill the “Data EEPROM Memory” window:

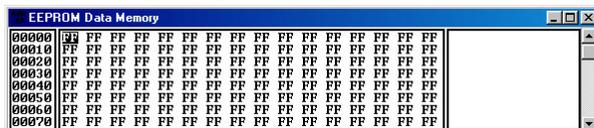


Fig. 13. The “Data EEPROM Memory” window

²⁵ ISP.EXE V2.65, available from user.cs.tu-berlin.de/~sirhenri/sides/up_avr/avrISP.zip.

The latest version is available from www.atmel.com/atmel/products/prod203.htm

²⁶ See www.qsl.net/zl1bpu/micro/ for a suitable cable. The Jaycar KC-5340 kit will also do the trick, as will the ATMEL ISP serial programmer.

This window can be seen by clicking on it, or from the menu **Window/EEPROM Data Memory**.

If the cable is faulty or not connected, an error message like this will be displayed:

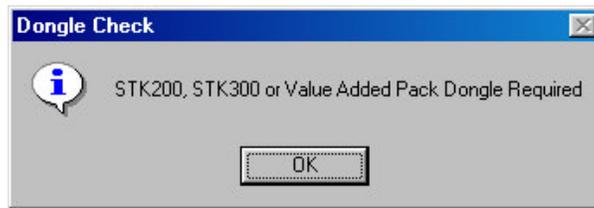


Fig. 14. Oh dear, no / faulty programming cable!

If the cable is connected, but there is a fault at the **GPSClock** (perhaps the programming header is wired incorrectly, is plugged in backwards, or there is no power applied to the clock), there will be a different message - "AT90S2313 Part not Detected":

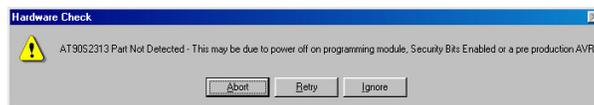


Fig. 15. No communications with device to be programmed

This message also appears if the wrong device has been selected in the Project Manager.

If everything has worked fine so far, the system is ready for programming. The software implicitly loads files into the current window, so in order to load the executable firmware, select the "Program Memory" window, by clicking on it, or from the menu **Window/Program Memory**.

Next select **File/Load...** and locate and select the executable file, which will likely have the name GPSClKnn.HEX. The file is in "Intel HEX" format. The computer is now ready to program the device.

From the menu, select **Program/Erase** to erase the device, then **Program/Program Device** to program it. It should take 10 – 20 seconds, depending on the speed of the computer. If the device is faulty or not blank, it may take longer and will report errors. It may also take ages and report errors if the **GPSClock** power supply is low. At this point the micro starts talking to the serial port, and possibly garbled time messages should be seen on the Terminal screen.

The final step is to set the EEPROM parameters, which are contained in the first few bytes of EEPROM. The example shown below is for a 10.000MHz reference, 9600 bps and a +13 hour default local time reference.

First row of EEPROM memory:

FF FF FF FF	First four bytes not used
40	Baud rate (9600 baud with 10.000MHz)
13 87	T1 reload for 2 kHz interrupt with 10.000MHz VTXO
13	NZDT offset (+13)
10	Integration time (example is 16 seconds, maximum is 31)
FF FF FF FF FF FF FF	Rest not used

The "EEPROM Data Memory" window will probably show all values as "FF", so leave them as they are (it won't matter what they contain). Simply plug the values shown into the first row of numbers. If the EEPROM in the device is blank, reading it will restore all values in the window to 'FF'. Next, count along to the 5th byte in the first row, and change it to your required data rate. The next two bytes set the reference division, and the 8th byte the local time offset. Finally set a default integration time. So, for example, the first line of the EEPROM memory when ready to program should read:

FF FF FF FF 40 13 87 13 04 FF FF FF FF FF FF

Then program the EEPROM. From the menu, select **Program/Program EEPROM** to load the data. It should take just a couple of seconds. If all is well, you should see time messages on the Terminal screen. If the message from the **GPSClock** is still garbled, check your maths again, and check the PC Terminal data rate.

Hints and Adaptations

In its VTXO form, **GPSClock** is very portable. It will run for about 10 days from a 1.2AH SLA battery small enough to hold in your hand. You might consider making the case big enough to hold such a battery, rather than the suggested 450mAH NiCd pack, but you'll need to arrange some other means of charging it. Perhaps a solar panel?

GPSClock can be operated with or without an LCD display. Several versions have been built for embedded frequency control applications with no display. The time is still available from the serial telemetry if you need it, and you could write a little PC application to display the time or simulate the LCD display if you wish – all the data on the display is available from the telemetry.

As an embedded device, you don't need any outside world connections except the GPS 1pps. Just build it into the host equipment that needs to be controlled. For example, build it into your frequency counter or communications receiver! Use a small tin-plate box, plenty of chokes and feed-through capacitors and you won't be bothered by signal leakage. The author has a GPS locked HF transceiver with embedded controller. This version has a LOCK indicator LED in place of the 1Hz output signal.

The author has built several embedded versions. There is a special version for use with high performance OCXO references, and even a version which will lock a crystal at fractional frequencies – in the author's example, a HF transceiver reference at 7,133,333.333Hz (hint – it samples the phase every three seconds).

The design concept slowly loses performance below 5MHz as the phase resolution drops off. With high quality but low frequency references, multiply the reference frequency and then drive the micro from the result. For example the author operates one from 2MHz x 5 = 10MHz and another from 5MHz x 3 = 15MHz.²⁷

The 2MHz version just mentioned is an embedded device for a precision transmitter – as well as generating carrier frequencies of 10, 5 and 2.5MHz, it also uses a direct digital synthesizer in the same micro to generate a complete time code sequence for modulating the transmitter, using the VNG format. It has 1kHz time-coded seconds ticks and a 125Hz sinusoidal sub-carrier, and all products are seconds synchronous. A portable version of this device (which essentially operates at 10MHz) would be perfect for anyone wishing to tape record events along with a time code²⁸.

Yes, you can use a PLED (polymer LED) display with **GPSClock**. The software is completely compatible. A few things to consider – (a) the current drawn by a PLED display is quite a bit higher. Overall, the current will be about 40mA instead of 10mA, so size the power supply accordingly. If you operate the brightness pot (connects the same place as the LCD contrast pot) from the unregulated AC supply rather than the 5V regulated supply, the display will go dim and save you current when the AC power fails. The brightness voltage required will be about 3V, rather than the typical 0.6V contrast voltage for an LCD. It would be a good idea to put a 4.7V zener across the pot, with a series resistor to the unregulated supply. PLED displays tend to fade somewhat when the same data is displayed long term, so a strategy such as scrolling or blanking the display is generally used. The **GPSClock** was not intended for a PLED display, and so the leading zero digits may fade after some time. Maybe a push-button switch or a photo-resistor to adjust the brightness on demand?

Yes, you can use a back-lit LCD display. The trouble is the HUGE current drawn by the back-light. One the author built draws about 400mA at even modest brightness. It is best to operate the backlight from the unregulated supply, otherwise the back-up battery will go flat in no time when there is a power failure. The back-light could perhaps be powered via its own switch-mode regulator for best efficiency. The author pulse-width modulates the back-light using the 1kHz output from the micro to save current.

²⁷ See http://www.wenzel.com/pdf/files/hc_cmos.pdf for an excellent and very simple CMOS multiplier.

²⁸ Any amateur astronomers out there? Contact the author if you are interested.

GPSClock can be operated with a non-adjustable reference (i.e. mechanically adjusted). Simply set **GPSClock** in mode C04, and use the telemetry or phase display to adjust the reference against GPS occasionally. You can use the telemetry to measure the reference performance. Once calibrated, the GPS can be disconnected. Of course, there's no compensation for ageing or thermal variations.

The **GPSClock** can be operated for months on end without GPS if necessary, without losing time or having any great effect on the reference frequency. However, before using the reference frequency for any important calibration, connect a good GPS 1pps signal to it for 10 minutes or so so that it can self-calibrate.

If the reference frequency is to be used regularly, add a buffer amplifier to the VTXO output (for example a 74HC14). For bullet-proof results, use a high speed opto-coupler after the buffer, and follow it with another buffer.

You can make the **GPSClock** unit do double duty as a clock and general purpose frequency reference with multiple frequency outputs. Because the dividers and buffers draw a lot more current than the clock, it's a good idea to operate them from an independent supply that won't drain the backup battery. Here we have a problem – if the AC supply is off, the backup battery could be drained through the buffer output and attempt to power the dividers via the clamp diodes in the CMOS devices. The answer is to once again use an opto-coupler! Then you can keep the **GPSClock** and frequency divider/buffer power supplies completely independent. Use two CMOS gates in parallel as output buffers – or better yet, use a bust driver device such as the 74HC244, and connect each the output to a BNC connector via a series 100 Ohm resistor and a short length of coax cable. Any simple attempt to switch the outputs will likely lead to serious cross-talk. If you wish to switch outputs, use a CMOS digital multiplexer device (not an analog multiplexer).

Using oven-controlled oscillators is well worth while, and **GPSClock** is fully capable of making good use of the improved performance, which (especially noise and thermal stability) is better by a considerable margin (see Fig. 17). You will need to provide a very good regulated heater supply at the correct voltage (typically 5V or 12V) and at sufficient power (typically 2 – 5W, depending on the size and style of oscillator). The micro power supply will also need to be very clean, noise-free and stable. Use an LM723 regulator in preference to a 3-terminal one. Performance and current consumption can often be improved further by lagging the oven externally to prevent heat loss. Oven heat will need to be on all the time, and it's a good idea to provide some automatic cut-off in case the temperature controller fails. You'd rather replace a fuse than have to throw away a cooked reference! You should be able to squeeze a thermistor inside the oven somewhere. If not, tape it to the outside of the oven and then lag the oven. Set the trip point just above the normal operating temperature.

Look for oven-controlled oscillators from old telecom gear, test equipment and old satellite navigation (Transit) receivers. Don't be put off by the age – very often older oscillators are better because they have much lower ageing rate, however, they may be out of calibration and need some tweaking. The more readily available and cheaper oscillators are not electrically adjustable, so you may need to steam the oscillator open with a propane torch and work out a strategy for frequency control. Very often the oscillator has its own internal voltage regulator, and pushing this regulator up and down by about 1V will be enough to move the oscillator by $1e-7$. Another method is to add a bias resistor to the oscillator transistor to alter its operating point and thus its Miller capacitance. In general these methods are more stable and easier to set up than adding varicap diodes, since you won't need to change any carefully selected temperature compensated capacitors – and the last thing you want to do is compromise the performance. You will need to add a buffer device if you wish to control the oscillator power supply, although a bias resistor can be controlled directly.

If you are considering buying an oscillator, be it an OCXO or VTXO/TCXO, look on Ebay. They are for sale in small quantities all the time, and the prices are good. You'll pay big money for the more famous HP oscillators, and for the **GPSClock**, the difference in performance isn't worth it. It would be a different matter if you were building a complex higher performance unit such as the Shera controller. VTXOs are available to order for US\$30 or so, but few frequencies are available off-the-shelf. You can't realistically add frequency control to a TCXO, and besides, the smaller ones are too small to work on.

Performance

The level of performance your oscillator achieves depends on several factors:

- The native performance of the oscillator (phase noise and thermal performance in particular). These depend on the technology used – double oven or single oven OCXO, TCXO, and on the cut and quality of crystal. In general, you get what you pay for!
- Power supply noise and stability, plus quality of other components such as feedback resistors and capacitors.
- Appropriate choice of gain and integration time.

Actually measuring the performance can be very difficult! You need a superb reference standard that is much better than the oscillator under test, and some technique to make comparisons. A high quality ratiometric counter can be used, and the results plotted, but this approach is labour-intensive. The most painless method is to use an SSB receiver locked to the reference standard, which then provides audio to a Spectrogram program on a computer. An ordinary AM receiver will also do the job if you are able to generate (either from your reference standard or from the oscillator) an audio tone that can beat with the other source in the receiver.

For the graphs shown below, the author used a Harris RF-505A communications receiver in USB mode. It has a single 5MHz reference for all its synthesized oscillators, and this reference was provided externally by an HP5065A Rubidium Standard, itself calibrated to 1e-12 against GPS. Measurements were made on harmonics at 30MHz (receiver tuned to 29999.0kHz) to improve resolution. The Spectrogram was the precision sound card program SBSPECTRUM, by Peter Martinez G3PLX, and has a minimum span (vertical direction) of 2.5Hz, giving a resolution of 10mHz (yes, milliHz), or 3e-10 at 30MHz.

Two references (using different technology) were received at the same time to allow comparison to be made. One was the GPSClock operating with a Rakon 10MHz VTXO200N VTXO, the other an older design by the author which uses an ERC 5MHz oven-controlled oscillator (pulled from an old satellite navigation receiver), and phase locked to GPS using thermal control (pulse width modulation of the oven heater).

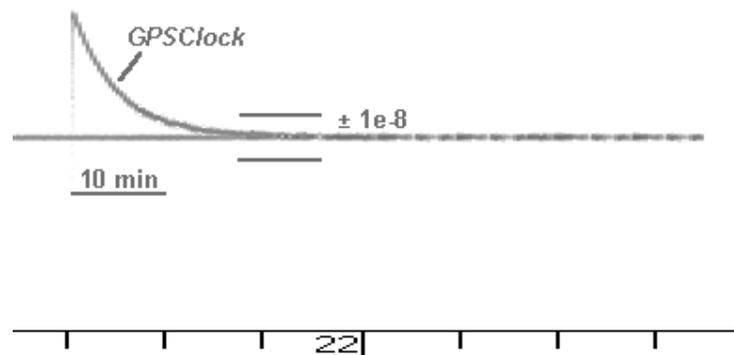


Fig. 16. Cold start performance – **GPSClock** is within 1 part in 10^8 in 10 minutes

In Fig. 16, the 5MHz OCXO GPS locked reference provides a continuous comparison baseline. It is much less noisy than the inexpensive VTXO, but uses 3W of heater power. The GPSClock had power removed for an hour prior to the start of the graph. The marks along the bottom are at 10 minute intervals. You can see that the GPSClock pulls the VTXO to within 1e-8 within 10 minutes, and the performance is close to 1e-9 (give or take some noise) within 30 minutes.

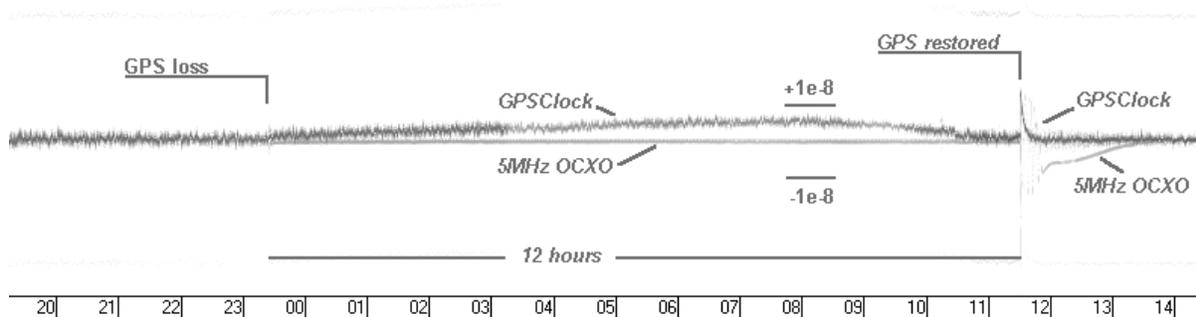


Fig. 17. 12 hour “holdover” performance

In Fig. 17, both references were running for more than 24 hours, locked to GPS. At the start of the test the GPS reference was unplugged from **both** devices, and after 12 hours, the GPS was restored.

Several things are obvious in this graph, which has a horizontal time scale in hours:

1. First, the OCXO is much more stable and has much less noise than the VTXO, and stays much closer to the correct frequency when the GPS reference is removed. It is after all operating at a constant temperature.
2. The VTXO has a thermal variation when not GPS locked. While it holds the last known feedback value for the duration of the “holdover” period, the frequency wanders off. Longer tests show a periodic variation over 24 hours of a thermal nature. The maximum excursion was well under $1e-8$, and more importantly (not shown here) the average excursion over many days is much less (i.e. the clock will be very accurate if no GPS is connected for weeks on end).
3. Note the recovery performance when GPS was restored. After an initial transient, the VTXO recovers to better than $1e-8$ in 10 minutes. The OCXO however, with older software lacking the “holdover” function, and a much slower control system, has a very large transient on restoration of GPS, and takes nearly two hours to recover its $1e-10$ performance.

Unless you have a $1e-10$ or better reference and locked receiver, you will not be able to replicate these graphs! Using this technique, the author can resolve $3e-10$. For higher resolution (such as calibration of the Rubidium Standard) a long term computer plotting technique is used. It takes three days to measure to $1e-12$ in this way. The RECORD program and some simple maths can do the job.

GPSClock Specifications

Clock –

UTC time **00:00:00 UTC to 23:59:59 UTC** within 1ms
Local time **00:00 am to 11:59 pm**
Local offset adjustable ± 13 hours
Independent setting of offset, hours, minutes, seconds via serial link
Incremental time trimming in 10ms steps via serial link
Automatic synchronization to GPS to within 1ms (typically within 500us)
No front panel controls, two internal adjustments (reference centre frequency, LCD contrast)

Reference –

Low power VTXO at 10.000MHz, e.g. Rakon VTXO4080N 10.000MHz or higher performance V-OCXO device
Design will operate at **any** frequency from 5 – 15MHz that is an integer multiple of 2kHz
Typical offset 1 part in 10^6 or better at centre of control range (without GPS)
Typical offset 3 part in 10^9 or better (with GPS)
Typical error over temperature range 1 part in 10^6 (without GPS), 1 part in 10^8 or better (with GPS)
Typical ageing rate 1 part in 10^6 per year
Calibration using continuous GPS reference
 <3 parts in 10^9 short term (0.03Hz @ 10MHz)
 1 part in 10^{11} long term (1 sec in 1000 years)
Holdover (error when GPS lost)
 Approx 1 part in 10^8 over 24 hours (clock error <1ms)
 Recovery when GPS restored reaches 1 part in 10^8 in under 10 minutes
Cold start performance (with GPS) 1 part in 10^8 in under 10 minutes

Outputs –

UTC time, reference phase via RS232, 9600-N-8-1
10.000MHz 2V p-p semi-sinewave, phase locked to GPS (see Reference, above – frequency depends on reference chosen)
1000.0 Hz, 50% square wave, CMOS 5V level
1.000 Hz, 1ms wide pulse, CMOS 5V level, positive true
12-bit PWM feedback at 31.25Hz
HOUR CHIME output, logic low for 1sec at second 00 of minute 00
ALARM CHIME every second 00 of every minute when AC power has failed

Reference –

GPS 1PPS from any GPS receiver supporting this option, 5V logic level, 1Mohm input impedance - GPS is the preferred option
Any pulse frequency which is a common factor of 2kHz from 0.5 to 1000Hz, and has the long term accuracy of GPS
Any other reference with long term accuracy, such as Rubidium or TV network frame pulse meeting the pulse frequency requirement
Glitch-free no-break operation: current control value is held when reference is lost, and new value reacquired when reference returns

LCD Display –

16 column x 2 line standard HD44780 display, in 4 bit mode (24x2 optional)
Display can be back-lit, but not supported by backup battery or DTR signal supply
Display can be PLED, but higher backup battery capacity will be required
UTC time, **00:00:00 UTC to 23:59:59 UTC**
Local time, **00:00:00 am to 11:59:59 pm**
Local offset -13 to $+13$ hours
Reference phase, three character (only on 24x2 display)
Control voltage, three character (only on 24x2 display)
Status and Mode, three character (only on 24x2 display)
Power failure message "BATT", GPS OK message "GPS" or blank if battery OK but no GPS (only on 24x2 display)

Controls –

All via serial command. Eleven commands. No front panel controls.
Four initial setting commands which affect time keeping
Seven trimming commands which do not affect time keeping, inc/decrement seconds, inc/decrement UTC offset, time delay increment
Eight operating modes, three phase locked, five for setup and adjustment

Power Supply –

AC supply via 10 – 15V DC adaptor, 200mW max
Backup from 7.2V 450mAH NiCd/NiMH battery 10 - 15mA (about 3 days) charged in-circuit
External 12V DC supply operation 10 - 15V DC
Can be powered by DTR signal from serial port of PC
True glitch-free no-break operation, power failure detection and warning

Firmware –

Written in machine code for ATMEL WAVRASM compiler. Source and executable code are copyright and separately available at a nominal charge (no publishing or distributing patched code without permission). Code size about 1k bytes.
Plenty of room for adaptations and improvements
Phase monitoring "chart recorder" software supplied with code

Appendix - About Frequency References

Not everyone is interested in precision frequency references, but those of us that are, seem to have a passion for the subject. Here are some simple techniques for making and calibrating references that offer good performance without breaking the bank.

Using a Frequency Reference

Most Amateurs are content with the calibration (or lack of it) of their HF transceiver, perhaps use a simple crystal calibrator with older or homebrew gear, and never even think about the accuracy of their VHF equipment. Amateurs who enjoy constructing, maintaining or adapting equipment often invest in or build a frequency counter, but how well is it calibrated? What do they check it against? A frequency reference is what is required.

The first thing most folk think about when considering frequency references is to use a Standard Frequency HF transmission, such as from WWV or WWVH. Those keen on the Frequency Measuring Contest will also know about HD2IOA (Ecuador) which transmits on 3810 kHz. However, while these standards may be very precise, the signals received from them in New Zealand can be inconvenient (no propagation when you need it) and quite inaccurate, due to propagation effects. Fig. 15, which is a spectrogram, a graph of signal frequency versus time, 20 Hz wide and about an hour long, shows how bad reception can be. The wobbly line that splits in two in places records the received carrier frequency of VNG (now off the air) on 16 MHz. The reasonably straight line is my own local frequency reference, GPS locked. VNG would move in frequency up to ± 5 Hz as a result of Doppler, caused by movement of the ionosphere's reflecting layers.

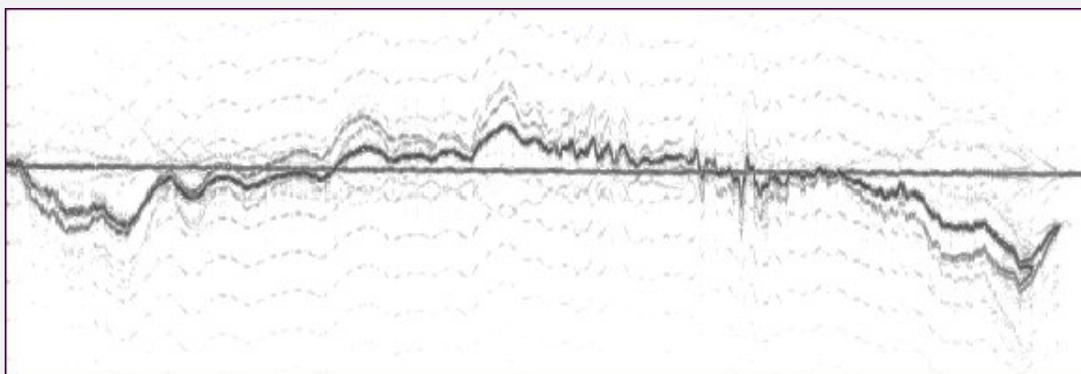


Fig. 1. VNG reception on 15 MHz

For low accuracy calibrators, beating the oscillator against WWV may suffice, but for anything more accurate, this technique is quite inadequate, as can be surmised from the picture above.

There are other frequency references freely available in New Zealand with much better performance. The frame, line and colour burst frequencies of the TV1 and TV2 networks were for many years controlled by a Rubidium Standard, and now by high accuracy GPS standards.²⁹ The carrier frequencies of JJY on 40 and 60 kHz are very accurate, and readily received in New Zealand at night. The seconds ticks available from some GPS units are also very accurate, and are the preferred source, since these pulses are directly traceable to international standards.³⁰ There are several ways to use these references, ranging from simple to complex, which I will describe. The technique chosen largely depends on the quality of results required.

Defining Frequency Accuracy

Frequency reference performance is not just a matter of quoting how far off frequency it is (the offset). The offset also changes over time (ageing rate), and has short-term variations (perturbations). These variations are caused by temperature, varying load and supply voltage, and sideband noise effects. Offset is usually adjustable, if you have something better to calibrate against, and the calibration reference needs to be at least one decade better. The better the reference gets, the harder it is to measure and calibrate. Here are some reference examples:

Reference Type	Ageing Rate	Perturbations
Inexpensive TCXO ³¹	Not quoted	0.5 in 10^6
Good OCXO ³²	1 in 10^9 per week	1 in 10^{10}
Rubidium Standard	1 in 10^{11} per month	1 in 10^{11}
GPS locked VTXO	Compensated - 1 in 10^{11}	3 in 10^9
GPS locked OCXO	Compensated - 1 in 10^{11}	1 in 10^{10}

²⁹ See NZART Call Book, page 8-1. See also www.irl.cri.nz/teams/msl/tiservi.htm. In Australia, use the ABC. TVNZ now use Trimble Thunderbolt GPS units.

³⁰ The NMEA time message may be correct on average, but will wander in time. The second tick provided by some units however is frequently held within 1us or better of UTC.

³¹ Temperature Compensated Crystal Oscillator

³² Oven Controlled Crystal Oscillator

Suitable References

Good quality TCXOs can be purchased inexpensively,³³ or recovered from defunct cellular phones. TCXOs are little silver boxes, the size of a postage stamp or smaller, typically on useful frequencies such as 10.000 MHz or 12.800 MHz. Some cellular phone units are on 9.60 or 14.85 MHz, which may not seem very useful, but using a micro controller, you can easily generate an accurate and useful reference. The micro controller project described in this manual uses an electrically adjustable 10.000MHz version of one of these devices.

For higher performance, an oven controlled reference is necessary. These can be home-built, but suitable crystals are very difficult to find and expensive to buy; the oscillator has to be very good, and the oven control very accurate, all requiring considerable experimentation and years of experience. The best solution is to hunt for surplus equipment that may contain a suitable reference. Look for old Omega receivers, pre-GPS satellite navigation receivers, telephone transmission equipment and communications test equipment. The older the better, to some extent, as the oscillator ageing rate will be lower. The two very high performance units I have come from Magnavox MX4102 Satellite Navigation receivers made in the mid 1980s. Unfortunately few of these devices are electrically adjustable without modification.

If you need high accuracy, but cannot locate a suitable high performance reference, consider phase locking a lower performance voltage controlled oscillator (VCXO) to an external reference. The TV line frequency is a popular choice³⁴ as the circuitry is simple, but from experience lock tends to be unreliable. An excellent micro controller design³⁵ from the Philips Research Labs, which uses an LF reference, would adapt well to TV line frequency. GPS is quite complex to use due to the low reference frequency (1 Hz) but has proved effective using a micro controller approach.³⁶ The project described in this manual is a further up-to-date example of this technology, which is much simpler and easier to use than previous designs.

It is my experience that if you have a really good reference (such as a top quality OCXO), there is no point in attempting to phase lock it to another modest reference, such as the TV. The phase locked reference is likely to have much degraded short-term stability, and of course is quite dependent on having the external reference connected for it to be any use. Voltage controlled OCXOs or TCXOs depend on very low noise control voltages, something not too easy to achieve. Unless a lot of trouble is taken³⁷, it is preferable to operate a manually adjusted local reference continuously, and simply monitor its drift against a TV reference every few weeks. When the offset becomes bothersome, you can tweak the calibration so it is off in the other direction and drifts through over the next few months. It is not usually possible to set an oscillator to an offset much under 1 in 10^9 , so there will always be an offset. Since the offset can be known precisely using phase comparison, frequency measurements and calibrations can be made with high reliability.

TV Phase Comparator

Calibrating a local reference which is a multiple of 15.625 kHz (250 kHz, 500 kHz, 1 MHz etc) is really simple, and costs virtually nothing. Simply include a divider from the reference which generates a 15.625 kHz signal (1 MHz divided by 64). A CMOS HC4020 device is ideal for the purpose. Calibration can be achieved by injecting this signal into the antenna of a TV set tuned to TV1 or TV2 and receiving a good signal. Look closely at the screen, and you should see one or two very narrow vertical lines superimposed on the picture. One should be white, the other black. Decide (according to picture content) which is easiest to see, put a sticky label on the screen to mark where the line is, start your stopwatch, and wait. From the time it takes the line to move off the screen to the left or right, round the back (so to speak!) and return to where it started from the other side, you can work out the offset of your reference. One minute is about 1 in 10^6 , 10 minutes is about 1 in 10^7 , and so on. If the line moves to the right, your oscillator is low in frequency. In under an hour you can easily adjust the oscillator as close as you need.

This technique is called phase comparison, and in doing this you are comparing the phase of your local reference with the TV line frequency at 15.625 kHz. This manual technique becomes tedious if your local reference is very good, or you wish to calibrate frequently. In that case, you need a more automated approach.

A simple phase comparator can be made using a flip-flop and a meter, as in Fig. 2. Using video from a TV receiver (a video recorder tuner is ideal), the sync separator, for example an LM1881, is used to recover the Line Sync. The leading edge of the 15.625 kHz Line Sync is used to set an RS flip-flop. The local reference is also divided down to 15.625 kHz, and the leading edge of the local reference used to reset the flip-flop. Thus the flip-flop is set for the period between one leading edge and the other, so is a measure of the phase difference. Since the inertia of a meter is too high to follow the 15.625 kHz signal, a simple 5V DC meter indicates the phase. As in the TV set approach, you can watch the meter and time the phase rotation, but a much better way is to use a chart recorder or logging DVM to record the phase drift over several hours. This technique easily allows measurement to one part in 10^8 .

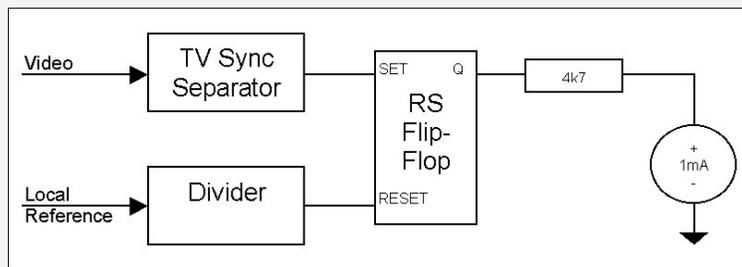


Fig. 2. Simple Phase Comparator

³³ See www.rakon.co.nz

³⁴ Electronics Australia, Nov 1986 page 34.

³⁵ Martin Ossmann, EDN, August 3, 1998 page 115. Also see <http://www.e-insite.net/ednmag/archives/1998/080398/16di.htm>

³⁶ Brooks Shera W5OJM, QST, July 1998. Also see http://www.rt66.com/~shera/index_fs.htm

³⁷ As in this project – the results using a good OCXO will be very good

Micro Phase Comparator

Although it is possible to compare the phase of a local reference to the TV Line or Frame frequency digitally, for example using digital counters and gates, using a micro controller is really much simpler. See Fig. 3. The "Divider" is a timer inside the micro, where high resolution is available. Use an AT90S2313 Atmel device, and drive Timer 1 with 1 MHz derived from the local 5 MHz precision oscillator. The timer is set to divide by 20,000, and thus resets automatically every 20 milliseconds. Use the TV Frame Sync to sample the value in the timer every 20 milliseconds. Using the Input Capture (ICP) feature of the micro, this is achieved without using any interrupts, so the precision stuff is not at risk of timing errors. See Fig. 3.

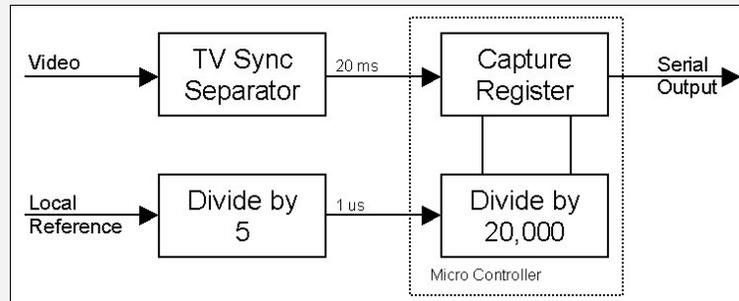


Fig. 3. Micro controller phase comparator

The Capture Register captures the number in the timer at the TV Frame sample point, and so represents the phase of the local reference, a count of 0 to 19,999, with a resolution of 1 microsecond. This register can be read and the data sent to a PC, which plots the phase and calculates the frequency offset. Using this technique, offsets as small as 1 part in 10^{10} can be detected, and running the PC software for an hour or so every few weeks will keep track of the ageing rate with ease. Fig. 4 shows part of a PC software display, which plots about 9 hours of reference phase (the upward sloping line). The plot is magnified because only the least significant eight bits of the 16 bit count are displayed. In 9 hours the phase has slipped 64 microseconds, so the frequency offset is -0.00083 parts in 10^9 (indicated on top line), or 8.3 parts in 10^{10} . An error of this order is equivalent to losing one second in 38 years!

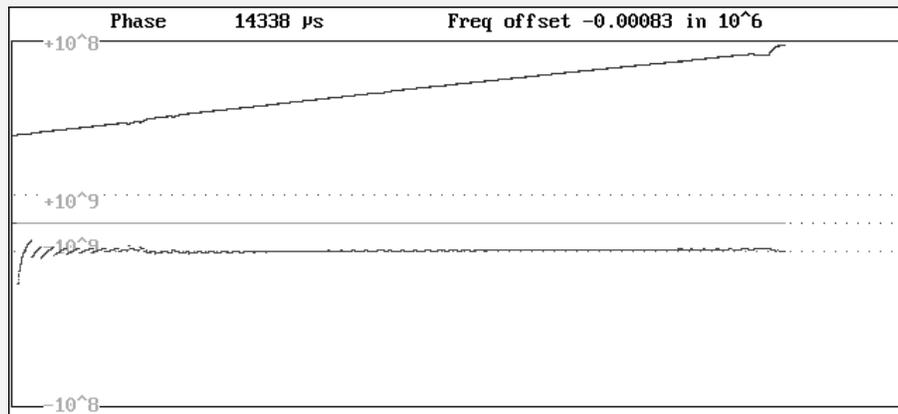


Fig. 4. PC phase comparison display

The phase plot has a couple of kinks in it, which I attribute to occasions where the TV network is not using its precision reference. The best time to run these tests is at night, and the worst during the weekend, when outside broadcasts are frequently used and the TV reference can be degraded.

The other line on the display in Fig. 4 is the frequency offset. It is essentially a horizontal line at about -1 part in 10^9 , and it is to this line that the calibration marks refer. Apart from the interesting effect at the start, when the data has insufficient resolution to resolve the offset, the offset shows as a dead straight line, indicating that the short term perturbations are well within 1 part in 10^{10} . Over a period of weeks this line moves up slowly as the offset changes with crystal ageing. My best reference (illustrated here) moves about 1 part in 10^{10} per week.

Other Improvements

A number of additional features can be easily added to a micro controlled device such as that just described. An obvious thing to do is to add an LCD display, to indicate phase and frequency offset digitally, and since there is plenty of room for software and plenty of available processing time, to add a super-stable clock as well (remember - one second in 38 years). The clock could also provide time to the PC, via the serial output, along with the phase and frequency offset. This is the micro project described in this manual!

A system such as has been described need not have TV video or GPS supplied all the time, and of course the PC software need only be operated when a reading is needed. Thus the system is reasonably economical of power use.

In order to compare phase over long periods, or include a clock, you must have a backup power source. Good crystal ovens use about 1W of heater power, and typically operate from 12V, so a 1 AH sealed lead-acid battery makes an appropriate backup supply. For high precision, the oven supply needs to be closely regulated, and when AC power is lost, the reference phase will move slightly for the duration of the failure. Use a 13.8 V regulated charger for the battery, capable of supplying the complete load. The battery drives a low dropout 12V regulator for the oven supply, and a conventional 5V regulator for the logic supply. Provided the power outages are short, the oven supply stays in regulation as the battery voltages drops down from 13.8 to about 12.5V. Outages more than an two or three hours long run the risk of phase error, although the clock would not noticeably lose time.

Using a Frequency Reference

Probably the best use of a local precision reference is as a direct reference for a frequency counter. The frequencies generally used are 1 MHz or 10 MHz. The well known Dick Smith Frequency / Period Counter, and others based on the ICL7216 chip, can use either frequency. My frequency counter design³⁸ uses 4 MHz, but adapting it to 1 MHz, 5 MHz or 10 MHz would be relatively straightforward.

A frequency reference can also be used to calibrate the oscillator in a frequency counter. Simply read the reference with the highest precision possible, for example reading the 4.43361875 MHz TV colour burst frequency with a 10 second gate time (resolution 1 part in 10^7), and adjust the counter to read the known reference frequency (including any known offset). To do the best possible job, run the counter continuously, and check it every week or so. Write down the calibration date and offset on a label attached to the counter, so you'll know when to calibrate next time, and learn what the drift performance is like.

A good frequency reference makes an ideal source for a frequency synthesizer, such as a phase locked loop (PLL) type or a direct digital synthesis VFO. Unfortunately DDS type synthesizers generally require very high reference frequencies, although these can of course be phase locked to a lower frequency. I use a 12.8 MHz TCXO as the reference in my LF Exciter,³⁹ which can generate any frequency from zero to 400 kHz with 85 mHz (milliHertz) resolution.

Bob ZL2CA and I have experimented with precision frequency transmissions on 80m. I used an old 20W AM transmitter, crystal controlled on 3750 kHz. An output from the reference was devised at 1250 kHz (5 MHz divided by four), and since this is also 3750 kHz divided by three, by injecting a small amount of this signal into the transmitter crystal oscillator, it could be locked to the reference. It would pull in from several Hz away, and provided a transmission with extremely high precision.

³⁸ See Break-In September/October 2002, page 4.

³⁹ See www.qsl.net/zl1bpu/micro/EXCITER