

Characterizing Temperature Drift of Oscillators

All oscillators exhibit output frequency that drifts with changing temperature. Coils expand when heated, capacitors can increase *or* decrease capacitance when heated, even crystals change frequency a little, sometimes in very non-linear ways. In an effort to minimize temperature effects, we often start with components that are inherently temperature stable, and hope that the resultant oscillator doesn't wander too far as the ambient temperature changes. What is needed is an oven so that the oscillator under test can be swept over a temperature range, along with a stable frequency counter to monitor frequency. This note describes:

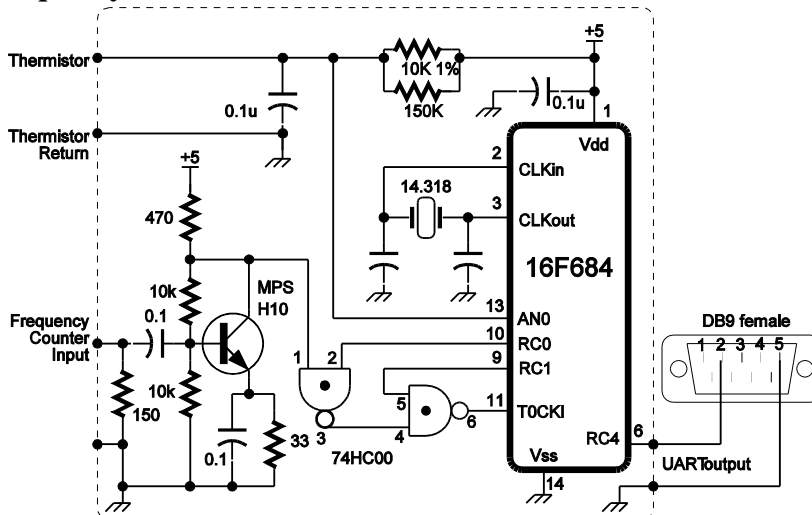
- A small oven
- A frequency counter plus preamp
- A thermistor temperature monitor
- Temperature compensation methods.

The oven



We don't need much here – a heat source (an incandescent lamp) and some insulation (a Styrofoam container). The insulation keeps the heat inside, but also helps to ensure that everything inside eventually settles to the same temperature. Even so, a circulating fan can make temperature runs go a little quicker. A more sophisticated oven might employ a triac-controlling lamp dimmer so that temperature is more finely controlled.

Frequency counter

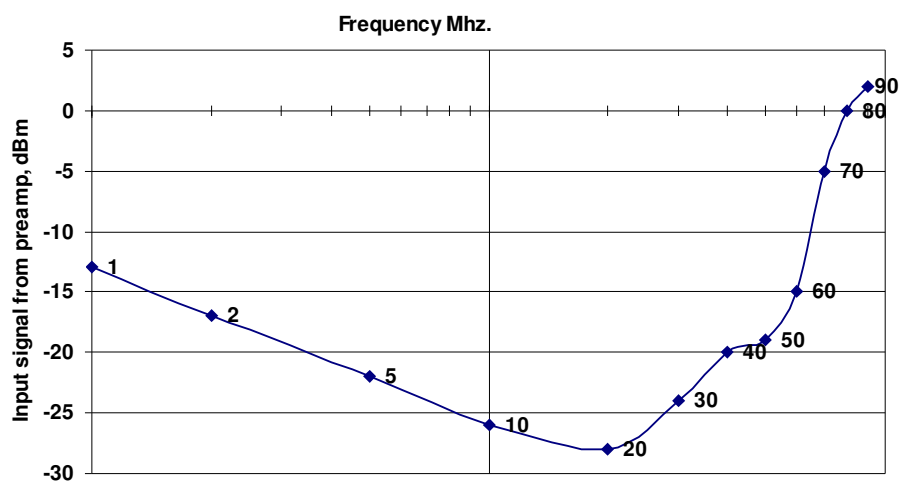


The counter is a simple microcontroller-based frequency counter, using its own 14.318 Mhz. clock crystal as a time base. Input frequency from the oscillator-under-test inside the oven is counted over a ten-second period, and then sent as ASCII numerical text over a serial interface to a personal computer's serial

port. This gives a frequency resolution of 0.1 Hz. This counter can measure input frequency from about 1 MHz. up to nearly 100 MHz. A counter like this (having no built-in display) is a very simple device that can be built inexpensively.

To keep temperature from affecting the counter's own time base, the counter is placed outside the oven, with transmission-line wires connecting to a preamp inside the oven. It is important that the preamp be close to the oscillator-under-test, so that it can be connected with short wires – the preamp shouldn't add extra impedances that load the test oscillator unduly.

A single VHF transistor converts the small signal from the input transmission line to a larger signal close to the +2.5v threshold level of the 74HC00 logic gate. This transistor also has useful gain. At the input side, a 130 ohm resistor provides most of the resistive load for the transmission line. The transmission line could be a 100 ohm twisted pair similar to those found in RJ-45 ethernet cabling. Or it could be a ground-signal-ground triple of a flat-wire cable. Signal input to the transmission line that gives a valid count is shown below (large negative numbers indicate better sensitivity):



Measuring Temperature

Over the temperature range that an oscillator is likely to be exercised (-10 to +50 degrees Celsius), a thermistor can measure temperature. About the simplest way to employ a thermistor is by adding a series resistor to a fixed, stable DC supply. The voltage on the thermistor (one end of which goes to ground) is read by a voltmeter – in our case an analog-to-digital converter inside the microcontroller.

The thermistor chosen is a small surface-mount NCPXH103J03RB from Murata, costing about \$0.16. At nominal room temperature (25 C), its resistance is 10K ohms. One slight disadvantage is voltage output that requires linearizing to get proper temperature. A third-order equation is calculated inside the microcontroller to get temperature:

$$T = -1.414*v^3 + 11.99*v^2 - 54.97*v + 111.5$$

Here, **T** is temperature in degrees Celsius, and **v** is voltage measured across thermistor. The thermistor is fed from a +5.0v DC supply through a 9.3K series resistor. The result is converted to ASCII digits and sent serially to a personal computers' serial port input.

For both frequency and temperature measurements, it is more important that results sent to the personal computer be more *precise and repeatable* than *accurate*. For example, the crystal time base of the frequency counter need not be exactly adjusted to its stated frequency of 14.318 MHz. But it should not drift from ageing or from ambient temperature changes while an oven temperature run proceeds. For this reason, no

variable capacitor is employed to allow exact frequency netting of the microcontroller's master clock. Unless a very high quality variable capacitor was used, its temperature coefficient could cause errors.

For similar reasons, a 1% 10K resistor is used in series with the thermistor. A 1% resistor is more stable than a 5% resistor, even if the 5% resistor happens to have better accuracy. To aid temperature stability, the 10K resistor is mounted outside the oven, along with the microcontroller.

At the personal computer end, a terminal program, such as Hyperterminal (for Windows), or Minicom (for Linux) is appropriate to read frequency and temperature results. Serial communication setup should be set for 9600 baud, eight data bits, no parity, and one stop bit.

Since the interface is a very simple two wire connection (serial in, and ground), **no flow control** is best. Only ASCII text characters are sent, so **XON/XOFF** protocol is acceptable too. **Hardware flow control** likely won't work, with hanging caused by DTR or DSR lines being inactive.

Once the ASCII text is sent to a file, a spreadsheet can import the data easily. Each frequency-temperature data set is separated with a carriage return (ASCII hex 0x0D) followed by line feed (ASCII hex 0x0A). Frequency is separated from its associated temperature by white space (ASCII hex 0x20). It should be easy to plot frequency vs. temperature, and calculate the slope (temperature coefficient).

Upon finding the temperature coefficient of the oscillator-under-test, the value of compensating components can be determined. Once the oscillator is modified, another temperature run in the oven can verify that adequate temperature immunity has been achieved. The oven can reveal an oscillators' temperature drift that you'd not notice in actual use, such as drift of crystal oscillators that seem to be rock-stable.

Micro controller

A PIC16F684 microcontroller was chosen for this project. A few others have the required internal components: an 8-bit counter with prescaler, a 10-bit analog-to-digital converter. No internal UART is needed, since the serial output stream is generated with software. The 16F684 has two counters available (TMR0 and TMR1). TMR0 is chosen over TMR1 for the counter. TMR1 has restriction of input period that limits its upper frequency range.

When the 8-bit prescaler is used with TMR0, the upper frequency limit is higher (above 100 MHz.), since the prescaler is a fast asynchronous counter. A disadvantage is that the prescaler value cannot be read directly, once the 10-second count period ends. An external logic gate 74HC00 is added that allows the microcontroller to pulse the prescaler. The prescale value is found by sending a pulse through the NAND gate, incrementing the counter. TMR0 won't change value until the prescaler overflows, at which point TMR0 increases by one count. By counting the number of pulses sent into the prescaler, its initial value can be found.

Even with the 8-bit prescaler feeding the 8-bit TMR0 (resulting in a 16-bit counter), the count value will overflow before the 10-second count period ends. A 32-bit count value is needed. The 16-bit counter is extended with software (adding two eight-bit general-purpose registers) to 32 bits. Every time that TMR0 overflows, a flag is raised in the interrupt control register (INTCON). This flag indicates when to increment the upper 16-bit registers - then the flag is reset.

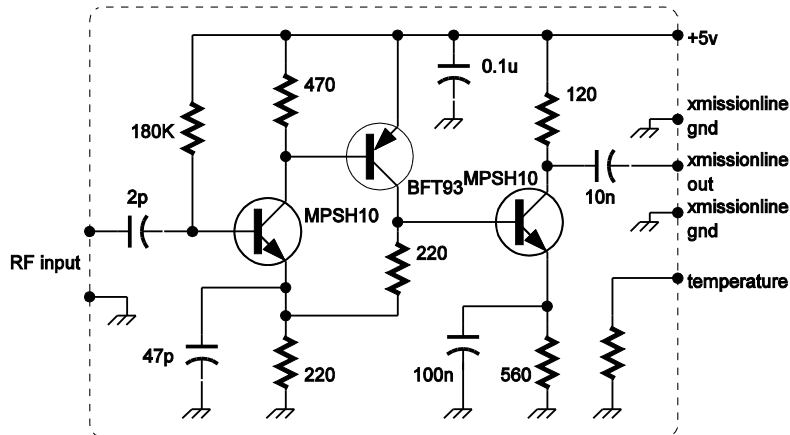
The 10 second count period is determined by simply counting clock cycles used by the program. Careful track is kept of cycles used, during which all code uses the same number of clock cycles, no matter which path is taken. In this way, the microcontroller's own master crystal clock determines the timing gate period – any drift of this master clock will result in frequency error of the result.

During the 10 second time gate, four equally-spaced temperature measurements are made, and averaged (by adding together). The result is a 12-bit integer that represents temperature. 16-bit integer math is used to apply the linearizing equation, ending up with a binary result 40 times the actual temperature. This result is rounded to the nearest 10th of a degree, and printed as a signed decimal number, with resolution of 0.1 degrees Celsius.

TMR1 generates interrupts to set the 9600 baud data rate. The interrupt handling routine for TMR1 is the heart of the software serial generator (UART). Serial transmission proceeds while other routines calculate temperature or frequency. Every ten seconds (actually, a little more) a new frequency/temperature data set is transmitted out.

The serial interface between microcontroller and personal computer does not meet EIA specifications. Normally a RS-232 driver chip such as a MAX232 accepts TTL logic levels from the microcontroller, and generates RS-232 signal levels. I have found that most modern personal computers will accept TTL levels with no errors. So the extra MAX232 is left out, and the microcontroller drives serial output directly. Keep in mind that this simple short-cut is not guaranteed to work.

Frequency Counter Preamp



This preamp sits inside the oven, and drives a transmission line to the external counter. It doesn't have much gain, but acts more as a buffer so that the oscillator-under-test sees a high impedance, rather than the low impedance transmission line. A critical spec for this preamp is that its input impedance be high over the expected frequency range of 1Mhz – 40 MHz. A little gain doesn't hurt – allowing the preamp to be more lightly coupled to the oscillator-under-test. Most usually, the coupling will be a small value capacitor – perhaps 2 pf.

Surface-mount components are used for a few reasons. Small thermal mass means that temperature changes settle out quickly, and temperature variations inside the oven are resolved. From an electrical point of view, surface mount components tend to operate better at high frequency, with less stray coupling and less stray reactance.

Since the preamp is mounted close to the oscillator-under-test, it is an appropriate place to mount the temperature-measuring thermistor. Try to keep the preamp and oscillator close together.

Keep in mind that the ground system is continuous right from the oscillator-under-test, to counter preamp inside the oven, to the microcontroller counter, to the personal computer's ground. The PC's ground is most likely connected through the power cord to your house-wiring electrical ground system. It is a good idea to power the counter and oscillator-under-test from a floating power supply such as a 9v battery. Otherwise, ground loops could cause problems.

Connecting to a PC should be regarded as rendering it vulnerable to electrical damage to the serial port or even the PC's power supply. Be especially certain that the lamp heater inside the oven is electrically isolated from the preamp – oscillator-under-test combination.