Data Acquisition Theory

There are several ways to get analog data into a computer, The usual approach is to use an analog-to-digital converter; such devices have at least one analog input channel, and a digital data bus for output. The binary number appearing at the data bus is proportional to the voltage at the analog input. Converters are available to give a resolution anywhere between 8 and 14 bits (15 bit with a minus sign is present "state of the art"; that's 4 significant figures. ed.), depending on the accuracy required, Eight-bit converters are relatively easy to implement on an eight-bit CPU like the Z80, since the data can be impressed directly onto the CPU's data bus. For higher-resolution A-D's the acquisition has to be done in two "chunks," using some decoding scheme to first look at the first 8 bits of the output bus, then at the remaining bits. In either case, the page to start a conversion can be either I/O or memory-mapped. The resulting data is then loaded into memory for subsequent display and other processing.

With all its advantages (mainly speed) this approach does have a few drawbacks, First among these is that it requires connection to the computer's data bus (8 lines) as well as to at least part of the address and control busses (lines like MREQ, IORQ, RAMCS, etc). Another drawback is that matters start getting quite complicated if greater than 8-bit resolution is required. Because of all the circuitry needed, most folks would prefer simply buying a commercial module rather than mess with all those wires, decoding, timing circuits, etc. See the Ener-Z Report Generator review elsewhere in this issue for an overview of one such package. Another is the Computer Continuum "(8+8)*8" A-D, D-A board (reviewed in a future issue.)

There is another route we can take if high speed (faster than about 300 per second on the ZX-TS) is not of the essence, It is an old method, but still very useful, In this method, the voltage is first converted into a stream of on-off pulses whose frequency is proportional to the input voltage. Such a device is called, aptly enough, a "voltage-to-frequency converter," or more simply a "V-F." Now all that is required is a single inport bit to get the stream of pulses into the computer, It is a serial rather than parallel approach.

Your ZX/TS has a dedicated input port already - the EAR jack. A single 280 IN A,(FE) command gets the state of the EAR jack input in bit 7 (the most significant bit;) from there it is a simple matter to write software to count how many pulses (state changes) appear during a fixed time span - the resulting count will be proportional to the input voltage. By appropriately modifing the IN command, this approach will work with any computer with an IN port such as a cassette input. Calibration can be done entirely in software, making the hardware aspect simple indeed.

V-F converters almost always use a circuit called an "integrator," which provides an output voltage proportional to the integral of the input voltage. This means that the higher the voltage to the input, the faster the output from the integrator will ramp up (increase), This signal is applied to a comparator, and when it exceeds the comparators upper trip point the comparator toggles (changes state), at the same time resetting the integrator. Result - the higher the input voltage, the higher the frequency of the pulses at the output of the comparator, See Figure 1. This is called "single-slope V-F conversion." Also possible is what is termed "dual-slope conversion," in which the conversion is done in two steps; a known reference voltage is used to ramp the integrator up, and the unknown voltage is used to ramp the signal back down. The ratio of the ramp-down time to the ramp-up time equals the ratio of the reference voltage to the unknown voltage, from which the unknown voltage can be easily derived. The dual-slope approach is more immune to drift and other inaccuracies, and allows "auto-calibration" in dedicated devices like the ubiquitous Digital Volt Meter (DVM). (Virtually all inexpensive DVM's use some variant on this approach - as I mentioned earlier, we're not dealing with anything new and esoteric here.)

When using V-F data acquisition, accuracy depends on the "acquisition interval." This is how long we are counting, and therefore how many counts we read for a given input voltage. If "full-scale" (the highest voltage we wish to measure) only returns 256 counts, we get 8-bit accuracy; but if it returns 16384 counts at full scale we have 14-bit accuracy. So you see that the more accurate you want it, the longer you'll have to sample the input pulses. How fast you can go depends on computer speed; on the ZX-TS it takes about 1 second to get maximum resolution (better than 14 bit, or about .005%.) At the other extreme, we can get 8-bit resolution (about .5%) with acquisition times down to a few milliseconds. In this program we'll experiment with the relationship between speed and accuracy by making the acquisition interval equal to the sampling interval; as soon as one sample is acquired, the program goes right on to the next, with no additional timing pauses. Timing is varied by changing the acquisition interval; as a result, the full-scale count and the number of counts per volt changes with sample-rate, and has to be compensated for when converting the counts into voltage or other units.

Next time we'll add programmable pauses so the sampling interval is greater than the acquisition interval for longer time-bases (minutes, hours, days). This will also allow time to update a continuous display during acquisition. You are encouraged to customize and improve these programs from there, to come up with exactly what you need for your application.

What applications can it be put to? Well, just briefly mentioning some of the possible input devices should help suggest plenty of ideas. You could use a potentiometer for position sensing (simple robotics, wind or antenna rotor direction indication, etc.), a photocell (light meter, etc.) or a strain transducer (for forces, stresses.) Use a microphone as the heart of a decibel (sound level) meter, Let's not forget about temperature probes, for clinical / meteorological "thermographs," solar heating control, or heat-transfer studies. For some uses you won't even need the V-F. The obvious example is as a frequency counter (this program will handle frequencies from about 2 Hz to about 20 kHz.) Similarly, you won't need the V-F for such things as radiation monitors (just run a Geiger counter output to the EAR jack and have the computer count the pulses, Use Tom's moving-trend routine to smooth the data as required.)

A final point before we get on to brass tacks is that the V-F approach lends itself very well to remote sensing; you can locate the V-F with the sensing device, and bring the data to the computer along a single digital line (use a comparator for cleanup at the receiving end if needed.) This is much more accurate and free of glitches than running a long analog line, and is cheaper and easier than bringing the data in on parallel data + control lines, But now that I've firmly convinced you that "you NEED this program," let's take care of the hardware aspect, and then briefly go over the system software (the fun part).

FIGURE 1 - PRINCIPLES OF V-F DATA ACQUISITION