We hope that these simplifications will encourage more instructors to introduce their students to the interesting physics observable with real-time HI.

---

[1] Keith Lubell and Robert Prigo, Am. J. Phys. **55**, 823 (1987).

[2] Nils Abramson, *The Making and Evaluation of Holograms* (Academic, New York, 1981).

Discount interfacing with the IBM parallel printer port

Ray Wisman and Kyle Forinash
Natural Science Division, Indiana University Southeast, New Albany, Indiana 47150

IBM PC's with parallel printer ports are widely available in many teaching laboratories. This note describes a convenient and inexpensive means for interfacing a PC to experimental apparatus using the parallel printer port and thus avoiding the expense of specialized interfacing boards. Using this method we are able to measure frequency, period, and virtually any low-speed time interval. A simple extension allows the measurement of resistance and therefore any variable resistance-dependent measurements such as temperature with thermistors and light intensity with photocells.

While a variety of types of data may be collected using the printer port, one of the most useful measurements that can be taken is the length of time a signal is high or low. Determining this requires a relatively accurate time base and a means of monitoring the signal. The computer system time-of-day counter provides a means for timing events and the printer status input lines (pins 11, 12, or 13 with pin 25 as common ground) the means for monitoring signal voltage changes between 0 and + 5 V.

Since only standard hardware is required, it is software that provides the special ingredient for interfacing. Although TURBO PASCAL version 4 has features that make writing interfacing software much simpler, any compiled language that provides for interrupt service routines should suffice. A copy of a sample program in TURBO PASCAL may be obtained from the authors.

The function of the software is as follows: (1) Initialize the computer system time-of-day counter to generate an interrupt at precise intervals; (2) count the number of times the time-of-day counter generates an interrupt causing a service routine to be called; (3) monitor the logic level of a printer status input, outputing the duration the signal was high or low as (number of interrupts)/(interrupt per second).

The 16-bit time-of-day counter on the IBM PC is normally set to interrupt every 55 μs, which is the longest available time between interrupts and corresponds to a counter value of 65 536 (the largest 16-bit number). This can be changed by using smaller counter values. The 1.1938-MHz time-of-day clock can be made to generate interrupt values from $1.193\,18\times 10^6/65\,536 = 18.2$ interrupts/s to $1.193\,18\times 10^6$ interrupts/s (Ref. 1). Actual usable values are limited to about 10 000 interrupts/s due to processing overhead. Using this value we have determined a theoretical accuracy of better than 10% is possible for a signal of 0.002-s duration with a 4-MHz CPU. This was verified by comparison with a commercially available frequency counter. It is possible to measure frequencies more accurately with different programming techniques; however, the flexibility in measuring longer time durations (for example, photogate timing) would be lost.

The printer port may be connected directly to a photogate to determine the length of time the gate is on or off (see Ref. 2 for a diagram on building your own photogate). A simple microphone circuit using a Schmitt trigger to change analog signals into digital signals can also be connected directly to the printer port to measure audio frequencies.[2] In fact, the frequency or duration of any digital (TTL) signal alternating between 0 and + 5 V can be measured in this fashion, regardless of duty cycle.

As an example of the data gathering potential of the printer port, consider the circuit shown in Fig. 1. The frequency of the 555 timer is determined by[3] $f = 1.44/(2R_2 + R_1)C$. The 7473 flip–flop serves to produce a 50/50 duty cycle from the 555 while halving the frequency and was added only for convenience. When the output (Q) of the 7473 flip–flop is connected to the printer port, the time the output is low and the time it is high may be measured. The sum of these is the period and the inverse gives the frequency which is 1/2 that of the timer chip. An unknown R_1 can thus be determined from this frequency measurement and the circuit becomes an (expensive) ohmmeter. Here, R_2 and C may be adjusted initially to put the frequencies in a conveniently measurable range. (We found an R_2 of 100 kΩ and C of 0.1 μF to be suitable for use with the thermistor used below.)

In order to measure temperature, a thermistor (for example, Radio Shack 271-110) can be substituted for R_1. The relationship between resistance and temperature is exponential for most thermistors so that $R = Ae^{-B/T}$, where B and A are constants that can be determined from the data sheet of the thermistor or by calibrating with known tem-

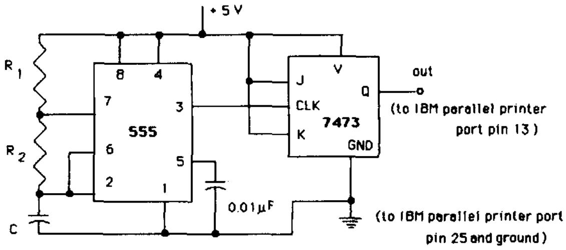

Fig. 1. Circuit used to measure resistance of an unknown resistor R_1.

peratures. A plot of $\ln(1/R)$ versus $1/T$ will give a straight line with slope equal to B and Y intercept of $\ln(1/A)$. In this way a measurement of frequency by the computer gives a resistance value from which temperature can be determined.

The method described above is very flexible (it is the basis of many gameport interfaces) and costs less than $10 to set up. This is quite a savings over the expense of other interface boards and we hope it will be of some use to PC users in the laboratory.

[1] D. Willen and J. Krantz, *8088 Assembly Language Programming: The IBM PC* (Howard W. Sams, Indianapolis, IN, 1983).

[2] D. L. Vernier, *How To Build a Better Mousetrap* (Vernier Software, Portland, 1986).

[3] A. J. Diefenderfer, *Principles of Electronic Instrumentation* (Saunders, New York, 1979).