SOME USES OF A MICROPROCESSOR IN TRANSDUCER SYSTEMS

by

DWIGHT R. PALMER

B. S., Kansas State University, 1979

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1981

Approved by:

M S P Lucas

Major Professor

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERENTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

SPEC
COLL
LD
2668
.T4
1981
P34
c.2

Table of Contents

List of Figures

Acknowledgements

I would like to express my sincere appreciation to Dr. M. S. P. Lucas for his guidance, wise counsel, and helpful suggestions.

Thanks are also due my wife, Kathryn, for her understanding and moral support.

I. Introduction

The advent of microprocessors has made radical changes in almost every area of electronics. Instrumentation is no exception. The availability of cheap computing and controlling power makes it possible to implement complex measurement systems that are otherwise unfeasible due to a variety of economic, reliability, power, and size restrictions. The result has been a marriage of the measurement techniques of transducer arrangements with the "thinking ability" of a microprocessor. Such a union is termed a "smart sensor", and this thesis is concerned with classifying and describing the various types.

A basic measuring device contains the following elements (see Figure 1): a stimulus, the transducer, signal conditioning, and an interface between the device and the real world. The heart of the device is, of course, the transducer. It produces an electrical output which is correlated with some physical phenomenon that is to be measured. The stimulus provides the energy needed for the transducer to operate properly (e.g. a power supply). Signal conditioning puts the transducer output in an acceptable form for the interface between the device and the real world, such as a display or a data link to a computer. A microcomputer can fit into this diagram in basically two ways. If it is used "passively" as a calculating machine, a block diagram similar to that shown in Figure 2 results. As the diagram illustrates, the computer performs some type of transformation on the raw data from the transducer to produce a reading that is improved in some way over the raw transducer output. Benefits

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

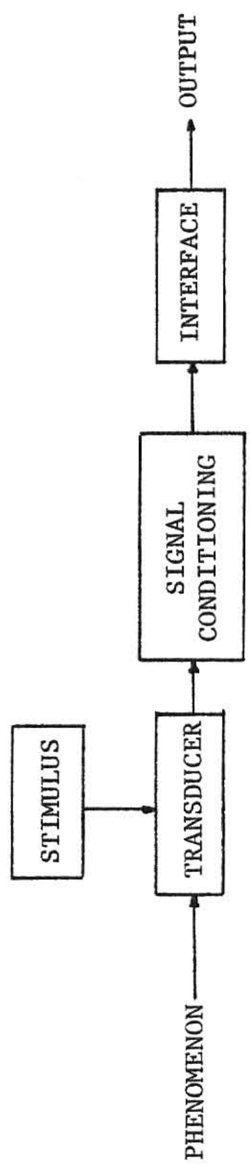

Figure 1. Basic Transducer System

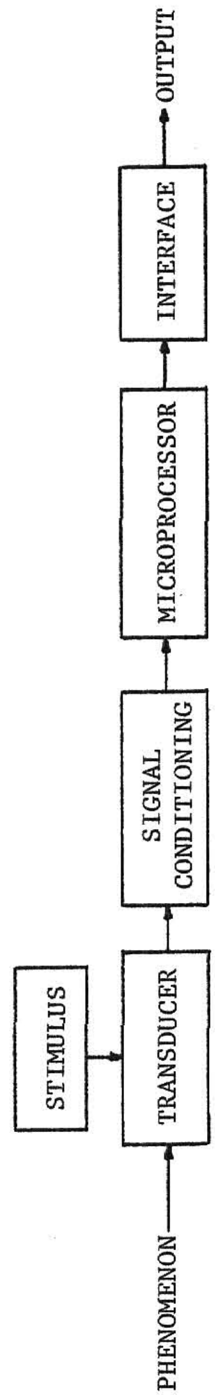

Figure 2. Passive Transducer System

resulting from passive microprocessor use include:

1) Automatic scaling of results to desired units

2) Reduction of nonlinearities and dependence upon unwanted parameters, such as temperature

3) Calculation of a measurement based on a combination of the outputs of a variety of sensors, with easy modification to include different sensors

4) Reduction of systematic errors

5) Increased reliability from statistical analysis of the outputs of redundant sensors

6) Use of complex measurement calculations

7) Simplification of signal conditioning circuitry

8) Data formatting for transmission to a larger computer

Figure 3 shows the second way in which a microprocessor can be used in a transducer system. Here, the computer is no longer just "massaging" raw transducer output. Instead, the controlling or "active" aspect of the microcomputer comes into play. It can, based on some algorithm which may include previous readings, change all the characteristics of the transducer system: the stimulus, the transducer itself, and the signal conditioning circuitry. Benefits include:

1) Increased effective measurement range

2) Use of complicated stimulus signals

3) Increased reliability through the use of standby sensors

4) Relaxation of transducer requirements and use of new transducer types

5) Automation of complicated measurement procedures

6) Use of esoteric principles of measurement, which may measure the desired quantity more directly than other methods

What follows is a discussion of the details of passive and active smart

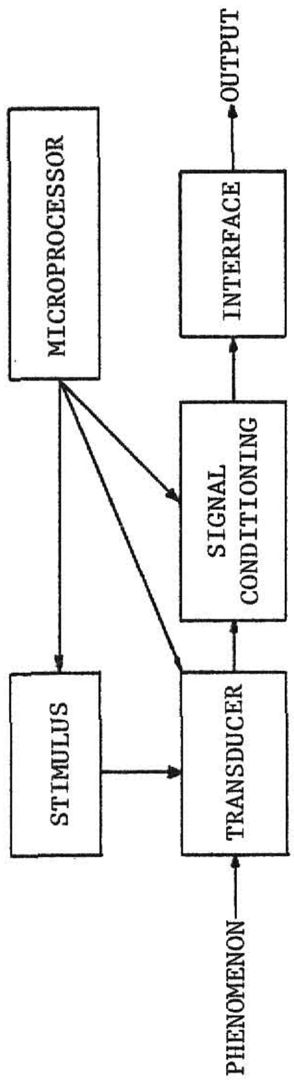

Figure 3. Active Transducer System

transducer systems, with examples taken from the many microprocessor measuring systems that have been designed in recent years. A description of an experimental impedance measuring system which illustrates in detail both passive and active techniques is included in the Appendices.

II. Passive Systems

Passive techniques have at least one common characteristic: in each, the output of the system is calculated from an equation involving the transducer output(s) and possibly several correcting factors. Consider the voltage output of a iron-constantan thermocouple. It bears a complicated relationship to the temperature difference between the two junctions of the thermocouple, which is tabulated by the manufacturer. To take a temperature reading, the operator must measure the voltage output, then look up the corresponding temperature in the manufacturer's manual. A microprocessor would have no trouble storing that table in its memory and looking up the value corresponding to the transducer's output. It could even interpolate values between the ones in its table. This is the simplest application of a microcomputer to a measuring system -- the scaling of the transducer output to proper units.

As an example which makes fuller use of the microprocessor's capabilities, consider a relative humidity sensor [1] whose transfer equation is

$$R.H. = A \exp(B \cdot T) \ln(g/g_o)$$

where g is the sensor output,

T is the absolute temperature,

A,B, and g_o are known constants.

Note the dependence on temperature. In order to get a reasonable relative humidity reading from the sensor output, one must change the scaling factor by the amount $(\exp(B \cdot T))^{-1}$. In the thermocouple example, the microprocessor's function could have been performed with a properly marked scale on an analog meter. In this case, however, the markings must be changed as the temperature changes. That task is not easily accomplished without human intervention, unless one has a microcomputer. A possible solution which doesn't require a computer adds with the system a device whose response to temperature opposes that of the transducer. This extra device is connected in such a way that the output of the combination exhibits little or no dependence on temperature. However, the implementation can be difficult, and one must be concerned with the possibility that the correcting effect of the added device and connecting circuitry may change with time and environmental factors. Also, the correcting device and/or its connecting circuitry must be carefully chosen to match the particular transducer so it does not overcorrect the temperature dependence.

A passive microprocessor can remove the temperature dependence by reading the temperature from a separate sensor and using the appropriate scaling table based on the result. Alternately, it could calculate the output directly from the above transfer equation, since both g and T are measured. Either way, the microcomputer system makes the temperature dependence of the actual transducer transparent to the user. Obviously, this technique can be extended to include any number of unwanted parameters, not only temperature. Such an extension can be a problem without a microcomputer, however, since adding a new device to the transducer assembly makes the hardware more complex, and minimizing the interaction between the correcting devices becomes difficult. The microprocessor system keeps the sensors separate and combines their results through calculation. As long as the relationship between the desired output and the transducer response can be written in the form of an equation involving quantities that can be measured by the system, the microprocessor can, through a process of calculation and/or table look-up, produce a corrected output value which negates the effects of the transducer's nonlinearities and dependence on unwanted parameters. The amount of memory needed to store the tables and the length of time needed to compute the corrected value depend on the complexity of the relationship between the desired output and the transducer response, as well as on the required accuracy and precision of the system.

When dependence on an unwanted parameter is removed by measuring the parameter and then doing a calculation, what is really being done is a measurement based on two parameters: the transducer output and the "unwanted" parameter. If the problem is viewed as a two-parameter measurement rather than as a correction of a single-parameter one, the category of multi-parameter measurements is revealed. From this view, the sensor does not try to eliminate measurement dependence on one parameter to improve its dependence on another so much as it combines the information gathered from all sources to calculate the measured value. An example of a multi-parameter system is one developed by Fisher and Lielevik [2] which predicts the severity of fungus attacks on apple trees based on three parameters: air temperature, leaf wetness, and relative humidity.

Systematic errors are those which are inherent in the measuring system and usually affect all measurements in the same way. For example, a D.C. offset is a systematic error that raises or lowers all values by a constant amount. These errors can be greatly reduced by first measuring the system response to a standard input and comparing the unknown response to the standard response. A D.C. offset can be eliminated by first measuring the system response with no input and subtracting that value (which is, in fact, the offset) from the actual response to an unknown phenomenon. Similar techniques can reduce the other systematic errors that produce the same error in all readings. A microprocessor can measure the standard system response and store that value in its memory for comparison with unknown responses. It also can update the stored value periodically, to correct for changes in the system parameters that occur as measurements progress.

Reliability is another area in which a passive microcomputer can be useful. A perfectly reliable measurement system produces readings which correctly reflect what is actually happening in the environment. If there is some prior knowldge of what a correct reading is like in terms of possible range of values, rate of change, and so forth, then the microprocessor can detect gross errors in measurement because the erroneous reading does not fit what previous readings have shown.

Less obvious errors can be detected and corrected by using redundant sensors. If one includes several sensors to measure a single quantity, chances are the majority will agree on the correct value. A microprocessor can be programmed to check all the sensors and report the majority decision. It can also flag a defective sensor for replacement at the next service call. Thus, by using redundant sensors, the failure of one need not result in an erroneous reading.

Much theoretical work has been done in the area of reliability. As a result, decision algorithms have been developed which are more complex than the simple majority voter described above, and are based on statistical analysis of previous readings. These more complicated methods may be too elaborate to be economically constructed using discrete logic, but the computing power of a microcomputer can easily handle very elaborate voting schemes with very little hardware and may let the theoretical methods of ensuring reliability be more widely used.

The complexity of the measurement algorithm used to calculate the measurement from the transducer response is limited only by the available program and data memory of the microcomputer. This presents intriguing possibilities for new types of sensors whose outputs on the surface may seem unrelated to a measurable quantity, but which produce an accurate measurement after the microprocessor has processed their data. For example, a microprocessor could perform a Fourier transform on the output signal and detect the presence and magnitudes of various frequencies. The full range of digital filtering techniques may be used to extract useful information from the data.

Howerter, et. al., [3] implemented digital filtering in a microprocessor system commissioned by the Federal Railroad Administration to detect internal rail defects in railroad track. The complete mechanism traveled along the section of track to be tested, using several ultrasonic transducers to probe the area beneath its wheels. Abnormal echoes (or the absence of expected ones) signaled an error. The microprocessor analyzed the results to determine what type of defect had been noted, if indeed one existed. Digital filtering techniques were used in the analysis to create a "yes-no" answer out of the data stream presented to the microcomputer. The system was much faster and more objective than the previous method of human interpretation of the echo traces, and resulted in improved performance of the detection process.

Related to the above is a seventh benefit of passive microcomputer use. The computer can do much of the signal conditioning in software, thus simplifying the hardware design. It can ignore a transient noise spike, for example, or wait until some steady state is reached after turn-on. More complicated digital filtering to remove noise generated in the input stages may also be implemented. Thus the complexity of external hardware may be greatly reduced and passed on to the internal software, which can reduce the cost of the sensing device. The availability of microprocessor filtering may also allow the use of cheaper transducers whose output is "cleaned up" by the smart part of the measuring system.

Under contract to Sandia Laboratories, Nickel [4] developed a microprocessor system which used a sophisticated digital filtering technique first presented by Widrow, et. al. [5]. Its purpose was to indicate the presence of an intruder into secure areas. The microcomputer obtained its input from Magnetic Interference Line Sensors placed around the secure area. They detected ground disturbances, such as would be caused by footsteps, vehicular traffic, and so forth. Widrow's adaptive digital prediction algorithm then produced a signal which predicted the input and changed itself to match gradual changes in the ground disturbances. The algorithm could adapt to something as repetitive and slow to change as an automobile or a passing train. It could not adapt, however, to fast changes such as those caused by the footsteps of someone trying to illegally enter the secure area. The predicted estimate (which theoretically did not include the intruder signal) was subtracted from the actual input. If there were too large a difference, the presence of an intruder was indicated. Thus, the sensor adapted itself to background noises and sounded the alarm only when it received data that had a high probability of being generated by an intruder. This is an excellent example of the degree to which a smart sensor can improve the characterstics of a given transducer. Raw output from the sensors would not have distinguished between loud background noise and an intruder. This system used a microprocessor to discriminate between the two and thus produced an improved output. Plus, the implementation was fairly simple, with little hardware and only moderately complicated software.

A side benefit of using a microcomputer in the transducer system is the ability to format and store the sensor data. Many measuring systems are part of a larger data acquisition and manipulation system, which perhaps uses a minicomputer to analyze the sensor data in detail. The microprocessor can convert the transducer output into the proper code for easy input to the larger computer. The translation problem can thus be distributed to the sensor level, relieving the larger computer of that chore. It is also possible for the microcomputer to store the sensor data temporarily in its memory. This way, if the minicomputer were occupied when a sensor was ready with data, that data would probably not be lost, since it could be stored until the larger computer had time to receive it. Of course, more buffer space at the sensor means more memory there, and that increases costs. However, the use of a microprocessor can simplify the interface between the sensor and the analysis computer, which may justify increased cost to an extent.

In summary, a passive microcomputer system manipulates the transducer output(s) to improve the quality of the measurement. Since the microprocessor can perform complicated calculations, new types of transducers may be used, and old transducer circuits may be simplified. However, the computer is using only half of its power; it is also a controlling device. The next section is concerned with applications utilizing this aspect of a smart sensor.

III. Active Systems

In the previous section, the computer power was used after the fact. The transducer was set up separately and its output was then massaged by the microprocessor. This allowed easy scaling, linearization, increase of reliability, and other benefits, each of which made use of the microprocessor as a calculating machine which applied various corrections to the transducer output to bring it closer to the correct value. There was no control over the transducer output itself.

Active systems, on the other hand, put the microcomputer right in the middle of the operation, instead of only at the end. As mentioned in the introduction, it can be used to influence every facet of the measuring system: the signal conditioning, the stimulus which allows the transducer to operate, and even the transducer itself. In each case, the microprocessor changes the characteristics of the measuring system so that the measured quantity can be interpreted with maximum fidelity and ease. The relatively minor increase in hardware complexity due to the presence of a microcomputer and its support chips is more than offset, in many cases, by the large increase in measurement procedure complexity that can be easily implemented in software. In active systems, the microprocessor manages the measurement process. It is used as a controlling device instead of a calculating device. The end result of this is a simplification of the external circuitry needed to implement a given measurement technique, which can either facilitate existing systems or make new methods of measurement more practical. Each of the areas mentioned above can benefit in both ways.

First consider the signal conditioning part of the measurement system. Perhaps the most obvious application of a controller to this area is in range switching. Signal conditioning, which converts the sensor output into something the microcomputer can understand, inherently has a limit to the spread of values it can convert. (This corresponds to the full-scale specification of an analog meter.) This usually limits the range of possible phenomena that can be measured by the given system. The microprocessor, however, can automatically change the gain of the signal conditioning circuitry (analogous to changing the scale of an analog meter) as needed to produce a good reading. If, for example, the output is so large that an overflow exists, the computer can reduce the gain so that an accurate reading is obtained. Similarly, it can increase the gain of small signals to achieve the maximum possible precision.

To see how this is accomplished, consider Figure 4, a simple signal conditioning circuit that uses an operational amplifier to amplify the transducer signal before sending it on to an analog-to-digital converter (A-D) for input to the microprocessor. The range of sensor output values that can be measured without overflowing the A-D is determined by the gain of the amplifier. This gain is given by

$$V_{amp} = -(R_2/R_1)V_{xdcr}$$

So a microprocessor that has control over the size of R_1 or R_2, can control the amplification and so control the allowable range of V_{xdcr}. One method of changing R_2 is to have several resistors in parallel with it (see Figure 5). By properly controlling the switches, the microcomputer can set R_2 to a wide range of values. For an even wider range of values, digitally-controlled potentiometers are available in integrated circuit (I.C.) form (for example, Analog Devices AD7525). The range of effective resistance available and the high reliability of monolithic devices make this type of I.C. a good choice for applications needing a wide range of resistance values. For those applications which require only a few dif-

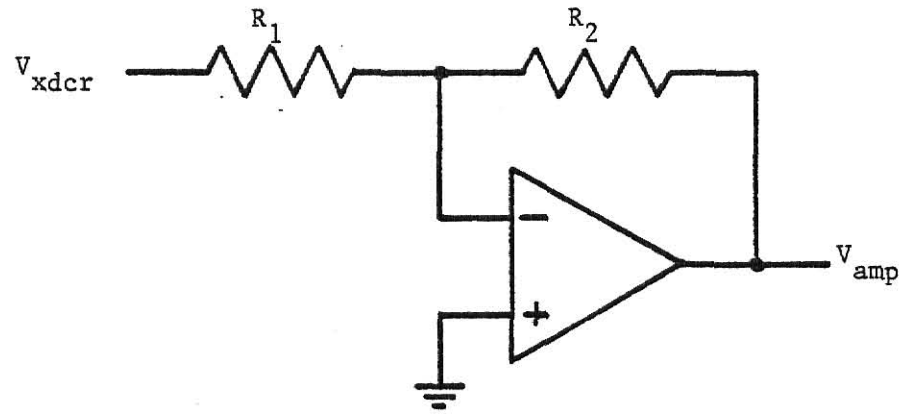

Figure 4. Simple Signal Conditioning Circuit

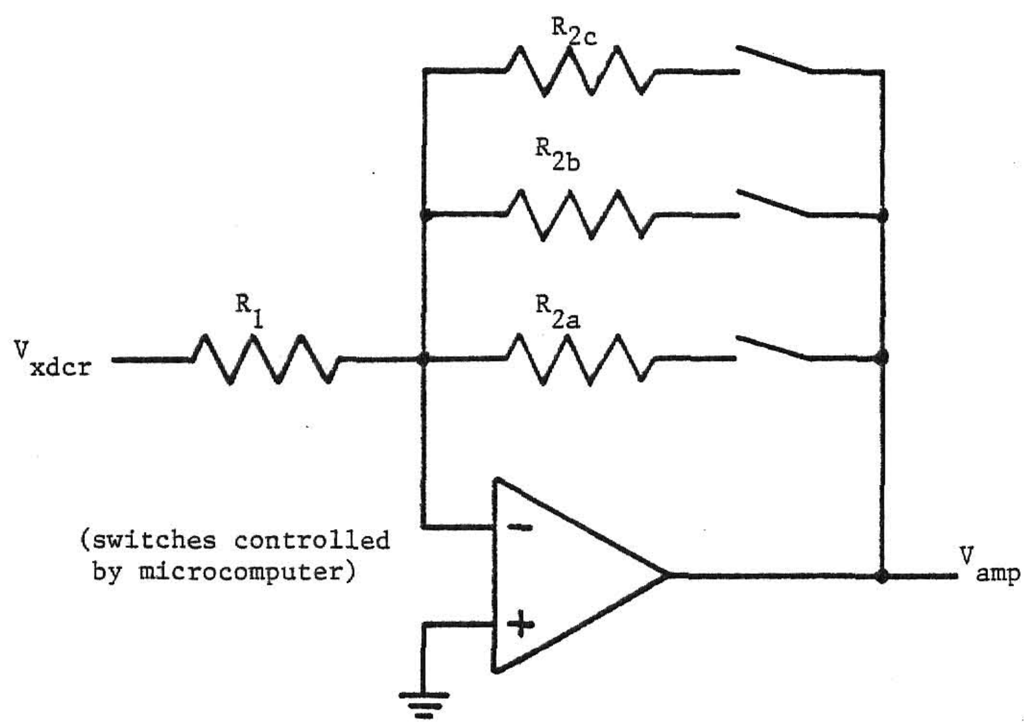

Figure 5. Computer-Controlled Signal Conditioning Circuit

ferent gain values, discrete resistors will work as well.

Conceivably, any change in the conditioning circuitry is possible. Consider Figure 6, which represents a simple digitally-controlled filter [6]. The voltage developed across the capacitor is determined by how long it is allowed to charge. Thus, by varying the duty cycle of the control signal of the switch, the computer can vary the effective capacitance and hence the bandwidth (cutoff frequency) of the filter. Of course, the allowable input frequencies must be much lower than the frequency of switch oscillation so as not to be overshadowed by commutation noise. Within that restriction, the sensor system can change itself to measure a different range of frequencies, if something in the measurement procedure indicates the range needs to be switched. The variable filter may also be used if the input signal contains different information in different frequency bands. A microprocessor can easily switch between bands to scan all the information available in the signal.

The stimulus circuitry may also come under computer control. As explained earlier, the stimulus is the energy that is supplied to the transducer so that it may operate properly. Many sensors derive their operational energy from the measured phenomenon itself; in this case, the stimulus cannot be manipulated without possibly falsifying the measurement. Many others, however, require an external power source which may be manipulated to advantage by a microcomputer. The possibilities exist for new measuring techniques, because of the versatility of an "intelligent" sensor.

A simple use of stimulus-control is in range switching. As discussed above in the case of signal conditioning, the magnitude of signal output must be scaled as large as possible for maximum precision, yet small enough to be measurable by a limited A-D. Active signal conditioning

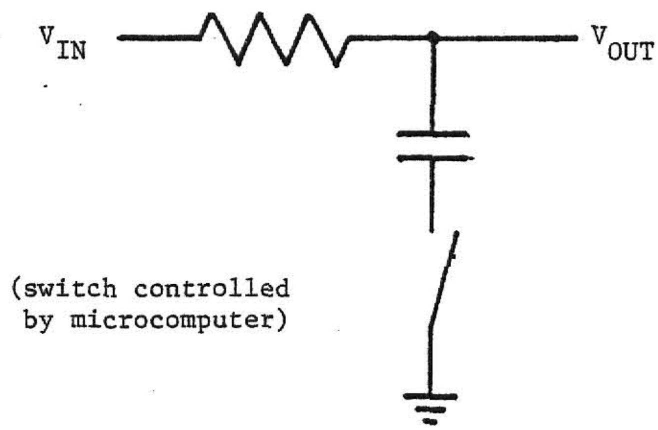

Figure 6. Simple Digitally-Controlled Filter

accomplishes this by changing the gain of the conditioning circuitry. The same effect can be achieved, in many cases, by changing the magnitude of the stimulus in a transducer system with a fixed-gain conditioning circuit. For example, consider a thin-film strain gauge, whose resistance changes with applied strain. The current through the gauge is proportional to the applied voltage ($V_{stimulus}$) and to the gauge resistance (i.e. the strain). The output current can thus be scaled simply by scaling $V_{stimulus}$. If a more complicated relationship exists, the microprocessor can calculate (or look up) the required change in $V_{stimulus}$ to produce the desired change in the output.

The versatility of microprocessor systems shows itself here, as well. Virtually any desired stimulus signal can be easily generated from tables stored in the microcomputer's memory. There is almost no need any longer for expensive hardware to generate bizarre signals. (The hardware is still needed for high-speed signals, due to the limited speed of the microprocessor.) The advantage of microprocessor-generated stimuli is that they can be easily tailored to the particular transducer. In fact, the stimulus may be changed to a different form if the system requires it, even during a measurement. Thus, a sensor can be "babied along" by the microprocessor. The computer can coax it into correct operation by giving it just the right stimulus at just the right time. This can make usable some types of sensors that are too particular to run on just a sine wave, for example.

Perhaps the most exciting area of active microprocessor application is that of the transducer itself. Certainly it is the area with the most promise of significant innovation. Computer control of the transducer implies the measurement-taking procedure itself is run by the microcomputer. Not only does the computer modify the circuitry to best handle the incoming signal, it is now involved in the acquisition of that signal. The tedious and often unfruitful search for a transducer that responds properly to the desired phenomenon can be considerably shortened. Complicated responses need not be contained within the physics or chemistry of the transducer, but can be transferred to the operating software of the microprocessor, which constructs the desired response by manipulating the outputs of more mundane sensors. As an example, consider the radiation monitor designed by Keefe, McDowell, and Groer [7]. They needed a sensor to detect ^{222}Rn-daughter concentrations in remote areas of uranium mines. Instead of attempting to find a substance that responded electrically to the amount of Radon daughter elements in the remote area, they used a fan to collect a sample of radioactive particles from the air on filter paper. Conventional transducers then measured the numbers of alpha and beta particles emitted from the paper. A microprocessor operated the fan as well as the motor to supply fresh filter paper for every measurement, and it ensured the radiation measurement was taken in the proper sequence at the correct time. They were able to use a relatively common sensor to respond to a complicated phenomenon because of the "intelligence" of the operating device itself.

Another example can be found in relative humidity measurements. These are typically taken by measuring the temperature of wet-bulb and dry-bulb thermometers, then performing a calculation. One problem associated with automatic systems is that the water used to keep the wet-bulb moist dries up quickly, especially in dry environments, thus invalidating the humidity measurement. A microprocessor system could use a sealed syringe filled with water to inject the proper amount of water on the wet-bulb thermometer whenever a reading was taken. The syringe's supply of water should last quite a while, since only the amount of water needed to take the readings is evaporated; the rest is kept sealed in the syringe until time for use.

Reliability may also be improved with active techniques. The system shown in Figure 7 uses redundant sensors and the microcomputer determines the correct response by some statistical analysis, as was done in the passive system discussed earlier. However, in the active case the microprocessor can disconnect a faulty sensor and attach a standby, which should work better. This can improve the reliability tremendously and keep the system operating when a totally passive system would have failed. The strategies for switching to standby sensors have been the subject of several papers in the literature [8,9]. A microprocessor can implement these strategies very easily and several other improvements as well, with minimal hardware investment.

In summary, an active smart transducer system uses the microcomputer as a controlling device, which changes the system parameters as required. The computer can produce the desired stimulus signals without the need for complicated hardware; it can extend the range of measurement automatically; and it can modify the characteristics of the measuring device, if needed. An active system has "intelligence" in the feedback loop between the sensor output and the measuring device, and so can measure more efficiently and reliably.

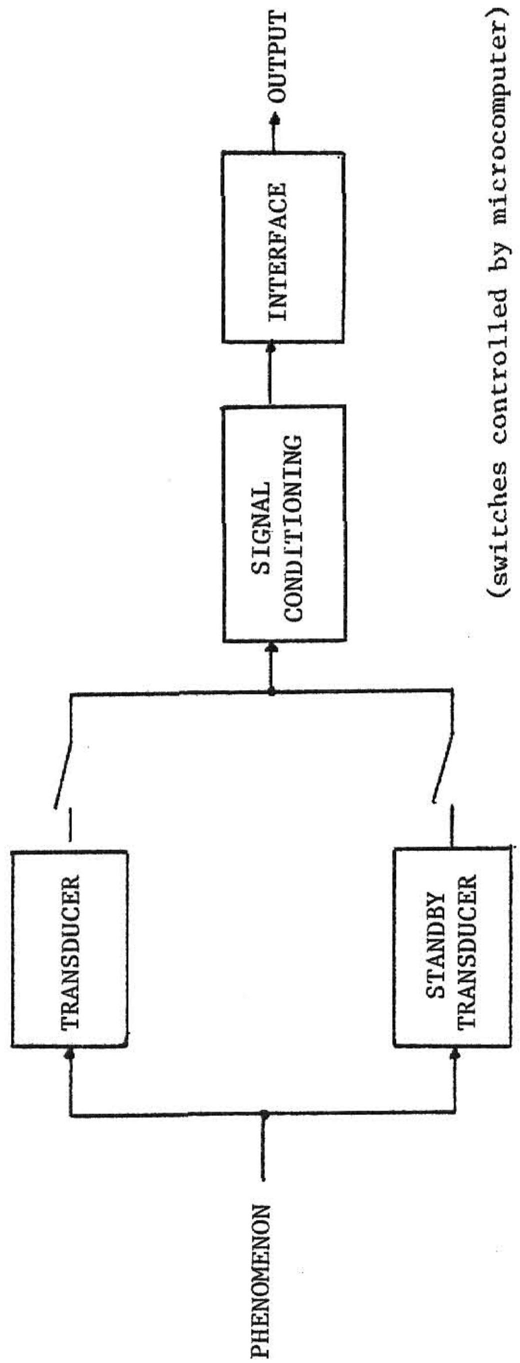

Figure 7. Simple Standby System

IV. Limitations and Trade-offs

The preceding paragraphs have presented the microcomputer as an instrumentation panacea. It is not, of course, and has drawbacks of its own. One of the most severe limitations of a microprocessor system is that of speed. The average microcomputer can execute an instruction in approximately one microsecond. If the processor must execute several hundred instructions to take a single measurement, then the maximum measuring frequency is in the low kilohertz range. That slow a measuring frequency is no problem for many phenomena, but it may be too slow for some of the faster ones. The speed problem is compounded by the necessary presence of some sort of A-D. Even the fastest ones today can sample only every 100 nanoseconds or so, and the slower ones can take tens of milliseconds. Obviously, this will limit the maximum measuring frequency even further.

If the sensor is to be operated with an internal power source, a microcomputer and its support circuitry may consume more power than it is feasible to provide via a battery or other depletable source. Although low-power microprocessors are being developed (a few are already on the market), the extra hardware will always require some power to operate. The trade-off between measurement complexity and energy requirements must be made during the design of any sensor.

Another requirement of the smart sensor is the manpower needed to write the control software. This problem is increasing as programs increase in complexity. Someone has to write and debug the instructions that give the sensor its intelligence. If no one in the design team has this ability, someone must be hired or trained to do it, and the cost may make the first few sensors very expensive. On the other hand, microprocessors are becoming more popular and versatile. The initial investment to obtain a software engineer should be offset fairly quickly by the increased use of microprocessors in many areas.

Cost is a fourth limitation of microcomputer systems. Although microprocessors can be obtained for about twenty-five dollars or so, this cost may be prohibitive in very small systems. This can be especially true if one includes the increased cost of a more complex hardware board, and so forth. Of course, a microprocessor reduces the amount of hardware needed to perform the same logical functions as a discrete logic design, and in that case should justify its cost. But in small, simple systems, a microcomputer may just be added expense that is really not justified by a need for complexity.

Physical constraints may also make a smart sensor unsuitable for a particular application. The microprocessor board may be too large to fit in the prescribed space, or need to dissipate too much heat. It may also be unable to meet ruggedness or environmental standards.

In summary, a microprocessor-based transducer system is usually justified when a complex measuring procedure is needed. It can reduce the parts count, board size, and power consumption. However, if the procedure is not very detailed, or the microcomputer cannot meet environmental or physical constraints, a discrete hard-wired implementation may be best.

V. The Total Smart Sensor

For clarity, each of the above microprocessor applications has been discussed separately, as if it were the only element in the system. In an actual sensor, though, one could include several of the above techniques, depending upon the nature of the system and upon economic constraints. Following are two examples of a smart sensor that use several of the microcomputer functions discussed earlier.

The first is a thermocouple psychrometer arrangement that measures relative humidity, according to a principle first presented by Spanner [10]. Its block diagram is given in Figure 8. In operation, the current source is turned on, which cools the thermocouple junction due to the Peltier effect. Since the junction is cooler than the surrounding air, water condenses on its surface. The voltage output of the thermocouple is now related to the wet-bulb temperature and can be used with the dry-bulb temperature (obtained before applying the cooling current) to determine the relative humidity. Wiebe, et. al., [11] reported a manually operated system, with switches to alternately connect the transducer with the cooling current and the measuring circuit. Output was on a strip-chart recorder and the relative humidity was calculated from the graphical data. This sometimes required graphical interpolation of the dry junction temperature, and was a fairly time-consuming process due to the incompleteness of the strip-chart output. Such a system can be automated using a microprocessor. It can flip the switches at the appropriate time, and also analyze the data, noting the time when the wet thermocouple temperature reaches a minimum. Thus, the microprocessor has control over the stimulus (cooling current) and performs calculations to determine the relative humidity and eliminate offsets and drift. It can also produce this reading a lot faster than the manual method of graphical analysis and

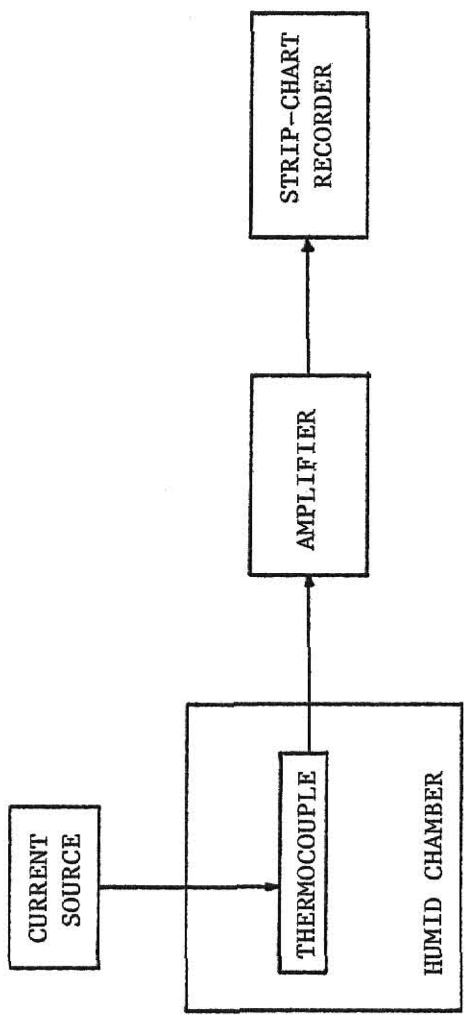

Figure 8. Humidity Sensor Block Diagram

error correction. This automation was attempted experimentally by the author, with inconclusive results. However, part of the problem may have been the use of reed relays in place of the switches. The electromagnetic field generated when the relays were activated may have induced voltages and/or currents into the system, causing the inconclusive results. The effect could have been very pronounced due to the large amplification required (the thermocouple output was in the microvolt range). A future attempt should use semiconductor switches which do not have the electromagnetic field associated with relays.

A flow velocity sensor developed by Tsukamato, et. al., [12] provides the second smart sensor example. They developed a correlation technique to measure the velocity of a cryogenic fluid. Conventional velocity techniques were not satisfactory because of the difficulties associated with the extremely low temperature of the fluid. To avoid these problems, they inserted a heating device into the flow stream, which applied random amounts of heat to the fluid. Downstream from the heater, a thermistor detected temperature rises in the fluid (see Figure 9). Ideally, the only difference between the heater input and the thermistor output was a time delay corresponding to the time for temperature fluctuations at the heater to reach the thermistor. Therefore, the thermistor output was statistically correlated with the heater input, and a peak corresponding to the time delay occurred in the correlation function. The flow velocity was then calculated by dividing the distance between the heater and the thermistor by the time at which the peak occurred in the correlation function.

A microprocessor generated the random heater signal and performed the correlation function, as well calculating the flow velocity and driving a display. Also included in the measurement algorithm were correction

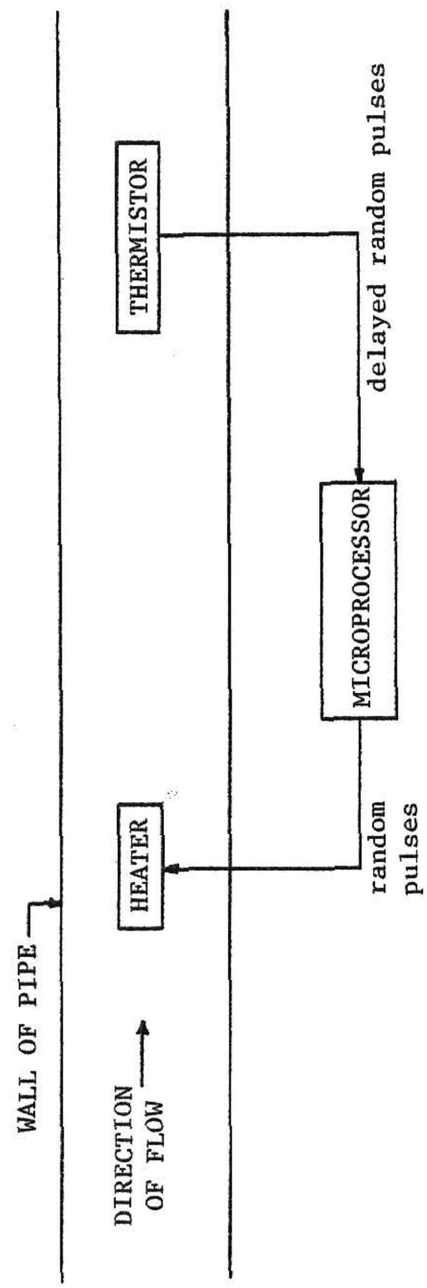

Figure 9. Correlation Flow Meter Block Diagram

factors to compensate for the non-ideal nature of the fluid, heater, and thermistor. A relatively simple hardware setup was used in combination with a moderately complex measurement procedure including stimulus-control and digital signal processing to create a smart sensor which measured cryogenic fluid velocities from 20 cm/sec to 2 m/sec. The microprocessor was able to extract meaningful information out of essentially random temperature fluctuations in the fluid. No other method could have implemented this novel measurement procedure as easily.

Smart sensors apply the large resources of a computer to the problem of measuring physical phenomena. Depending on the nature of the specific case, a microprocessor system may not be the best solution, due to the limitations discussed above. However, in cases such as the ones outlined in this section, the capabilities of an intelligent measuring system outweigh the disadvantages and can produce better results than the conventional methods.

References

1. M. S. P. Lucas, J. Troughton, and W. Hydes, "Thick Film Humidity Sensor", Author's Manuscript.

2. P. D. Fisher and S. Lillevik, "Monitoring System Optimizes Apple-Tree Spray Cycle," Electronics, Vol. 50, No. 24, pp. 125-127, November 1977.

3. E. D. Howerter, et. al., "An Automated Approach to Ultrasonic Rail Flaw Detection," Proceedings of Southeastcon '77 (April 4-6, 1977), pp. 303-306.

4. D. J. Nickel, An Evaluation of Various Microprocessor Implementations of an Adaptive Digital Predictor for Intrusion Detection, A Master's Report, Kansas State University, 1979.

5. B. Widrow, et. al., "Adaptive Noise Cancelling: Principles and Applications," Proceedings of the IEEE, Vol. 63, No. 12, pp. 1692-1716, December 1975.

6. L. P. Huelsman and P. E. Allen, Introduction to the Theory and Design of Active Filters, New York: McGraw-Hill, 1980, pp. 360-378.

7. D. J. Keefe, W. P. McDowell, and P. G. Groer, "Use of a Microprocessor in a Remote Working Level Monitor," IEEE Transactions on Nuclear Science, Vol. NS-23, No. 1, pp. 704-707, February 1976.

8. N. G. Dennis, "Reliability Analysis of Combined Voting and Standby Redundancies," IEEE Transactions on Reliability, Vol. R-23, No. 2, pp. 66-75, June 1974.

9. A. C. Carver and P. J. Spreadbury, "The Application of a Microprocessor to Provide Accurate and Reliable Temperature Measurement," Proceedings of the Conference on Programmable Instruments (November 22-24, 1977), pp. 203-210.

10. D. C. Spanner, "The Peltier Effect and its Use in the Measurement of Suction Pressure," Journal of Experimental Botany, Vol. 2, pp. 145-168, 1951.

11. H. H. Wiebe, et. al., "Measurement of Plant and Soil Water Status," Utah Agricultural Experiment Station Bulletin 484, Utah State University, May 1971.

12. O. Tsukamato, K. Harada, T. Fujikawa, "Correlation Flow Measurement of a Cryogenic Fluid by Use of a Microprocessor," IECI Proceedings -- Industrial and Control Applications of Microprocessors (March 19-21, 1979), pp. 250-255.

Appendix 1

An Experimental Impedance Measuring System

An impedance-measuring device has been designed which clearly demonstrates some of the problems encountered in designing a smart sensor. It measures the conductance and capacitance of a passive element connected to the test apparatus. This device is novel in that it does not measure impedance directly, but rather measures the currents developed through the element when sinusoidal voltages of two different frequencies are impressed upon it. The phasor equations of these two currents are

$$I_1(jw_1) = V_1(jw_1)(G + jw_1C)$$

$$I_2(jw_2) = V_2(jw_2)(G + jw_2C)$$

where G is the conductance of the element,

C is its capacitance,

w_1 and w_2 are the two frequencies involved,

$V_1(jw_1)$ and $V_2(jw_2)$ are the two applied voltages,

and $I_1(jw_1)$ and $I_2(jw_2)$ are the two measured currents.

These two equations can be solved simultaneously to find G and C. In terms of the amplitudes of the currents and voltages,

$$C = \frac{(V_2^2 I_1^2 - V_1^2 I_2^2)^{\frac{1}{2}}}{V_1 V_2 (w_1^2 - w_2^2)^{\frac{1}{2}}}$$

$$G = \frac{(V_1^2 I_2^2 w_1^2 - V_2^2 I_1^2 w_2^2)^{\frac{1}{2}}}{V_1 V_2 (w_1^2 - w_2^2)^{\frac{1}{2}}}$$

The block diagram of a system which implements this measurement procedure appears in Figure 10. As can be seen, there is very little hardware needed, only a D-A, an A-D, plus a filter and a current-to-voltage converter; the latter two are implemented with a single operational amplifier. The microcomputer concurrently outputs values (obtained via table look-up) to the D-A to produce the sine wave stimulus voltage, and inputs values from the A-D (which are subsequently stored in memory) to measure that applied voltage as well as the current through the element. Several cycles of the sine wave are performed, then the stored values are searched to find the magnitudes of the current and voltage at that frequency. This procedure is done again with the frequency doubled, and the four resulting magnitudes are then used in the above equations to calculate the capacitance and conductance of the element. Figure 11 shows a flow chart of this procedure.

The system has been implemented on a Z-80 development system (see Figure 12). It requires 603 bytes of program memory and uses 271 bytes of data memory. An Analog Devices AD7524 and two AD741 operational amplifiers form the eight-bit D-A. The filter to smooth out the quantization noise from the D-A output and produce a good sine wave is provided by a simple R-C network and an AD741 used as a buffer amplifier. The current-to-voltage conversion is performed by a fourth AD741 and associated resistors. The 10 kohm potentiometer connected to that amplifier provides offset adjustment for the eight-bit A-D, a Teledyne 8703. The +10 volt reference for the D-A and the -5 volt reference for the A-D are derived using AD584 voltage reference chips.

The Z-80 development system communicates to the rest of the circuit via an eight-bit input port and an eight-bit output port. The eight data lines of the 8703 A-D are connected directly to the computer's input port.

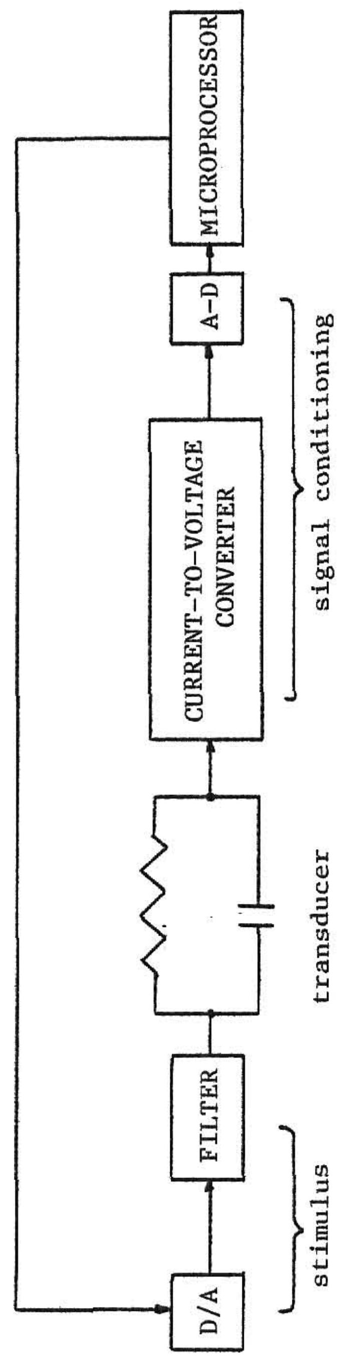

Figure 10. Impedance Meter Block Diagram

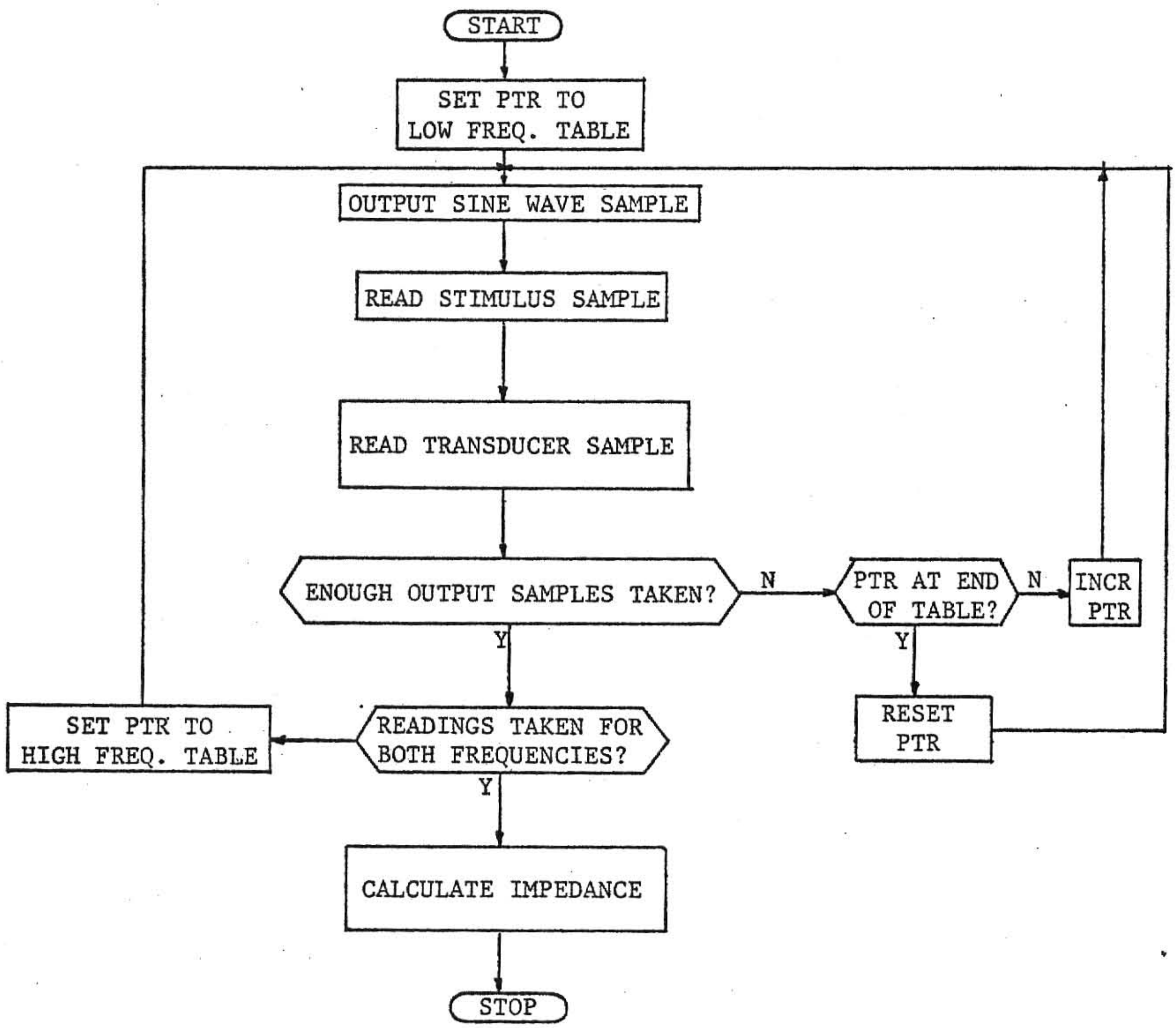

Figure 11. Impedance Meter Flow Chart

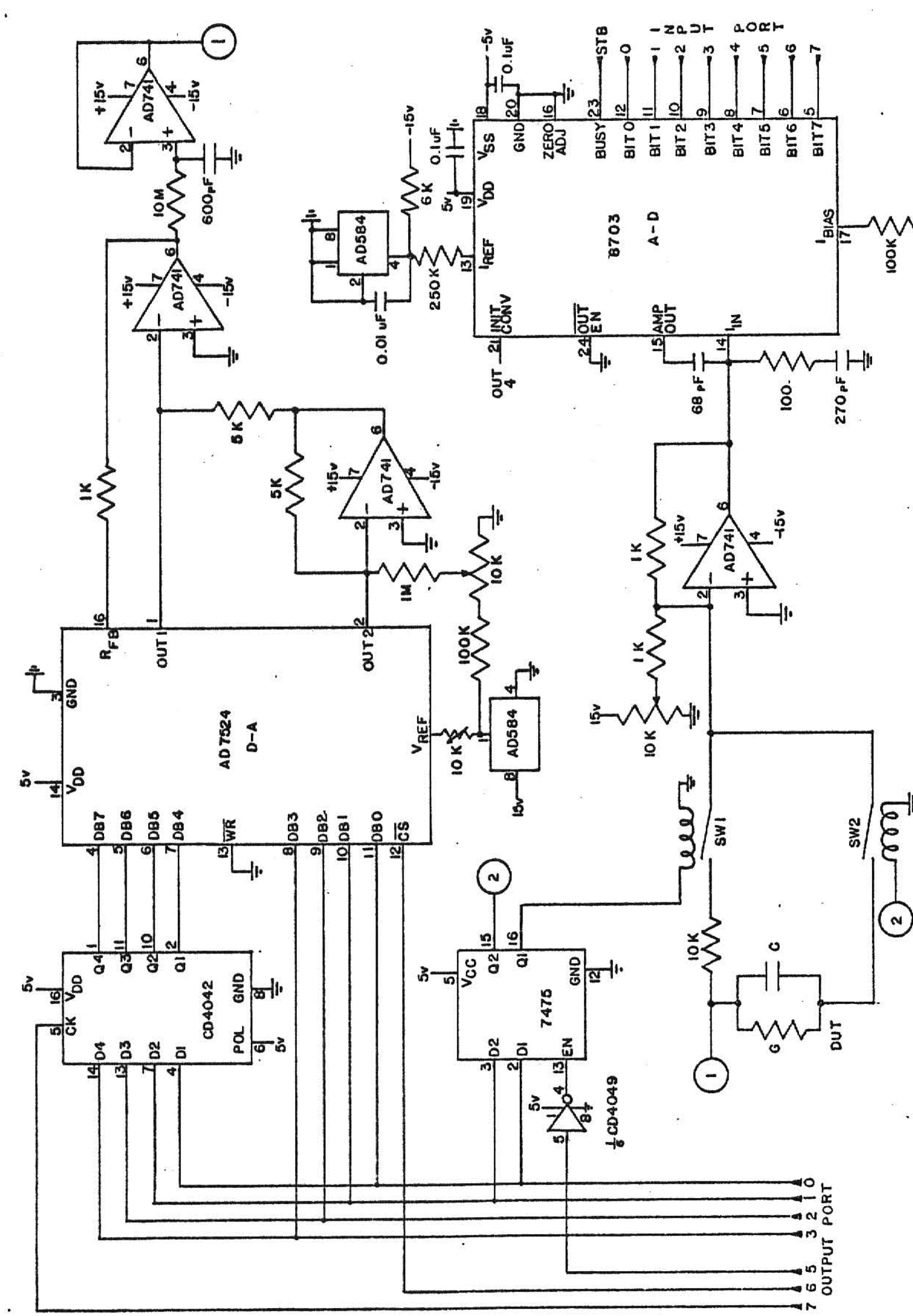

Figure 12. Impedance Meter Schematic

A conversion is begun by placing a "1" on line four of the output port; completion of a conversion is signaled over the Z-80's "Input Strobe" line, which is connected to the "Busy" line of the 8703.

The AD7524 D-A has an internal latch to store the most recent value received from the computer. Unfortunately, the top four bits of the Z-80 output port must be used for chip selection, so only the lower four output port bits may be used for data. The eight bits corresponding to a D-A value are loaded by first storing the high-order four bits of data in an external CD4042 four-bit latch. The low-order four bits are then placed on the output port lines and the chip select of the AD7524 is enabled, latching all eight data bits into its internal latch.

The output port also controls the two reed relays SW1 and SW2 which alternately connect the A-D input with the applied voltage (SW1 closed) and with the output current of the device under test (SW2 closed). The condition of the relays ("1" = closed, "0" = open) is stored in another four bit latch, a 7475 (a CD4042 does not have the current driving capability to activate the relays).

To summarize, the output port bits are used as follows:

Bit 7: Clock of CD4042, latches on "1" to "0" transition.

Bit 6: Chip select of AD7524, latches on "0" to "1" transition.

Bit 5: Enable of 7475 (signal inverted), latches on "0" to "1" transition.

Bit 4: Initiate conversion line of 8703, conversion starts on "0" to "1" transition.

Bits 0-3: Data bits.

The program to operate the sensor is reproduced in Appendix 2. It differs slightly from the flow chart of Figure 11 in that eight readings of I_1, I_2, V_1, and V_2 are taken and the averages are used in the capacitance and conductance calculations. Those calculations are performed using fixed-point arithmetic. Routines for doing the 16-bit multiply, divide, and square root operations are also included in Appendix 2. Following them is the BASIC routine used to generate the sine wave tables.

Another difference lies in the calculation itself. Since all readings are taken from an A-D instead of a voltmeter, the values used in the calculations are only proportional to the currents and voltages, instead of being the actual values. For simplicity, the calculations used in the algorithm also omitted the proportionality constant $(w_1^2-w_2^2)^{1/2}$ present in the denominator of the theoretical equations. Hence, the calculated capacitance and conductance values are not the actual values, but are instead something proportional to them. The calculated values C' and G', which are proportional to the actual C and G values, are plotted in Figures 13 and 14, respectively. As can be seen, there is a proportional relationship between the measured value and the actual one, indicating that the measuring premise is probably sound. However, the calculated values lack accuracy. In an effort to determine the causes of this inaccuracy, the experiment has been repeated on a PDP-11 minicomputer, which has a 12-bit D-A and two 12-bit A-D's (eliminating the need for reed relays), plus floating-point arithmetic. The extra precision of the D-A and A-D help to make the measurements more reliable. The FORTRAN program to calculate the values of capacitance and conductance appears in Appendix 2. An assembly language subroutine operates the D-A and A-D, and follows the FORTRAN program. Results are summarized in Figure 15 for conductance and Figure 16 for capacitance. As can be seen, the calculated values are much more accurate, within 4% for conductance and 3% for capacitance.

The essential differences between the two sensors are the size of the D-A and A-D and the precision of the mathematical calculations. The type

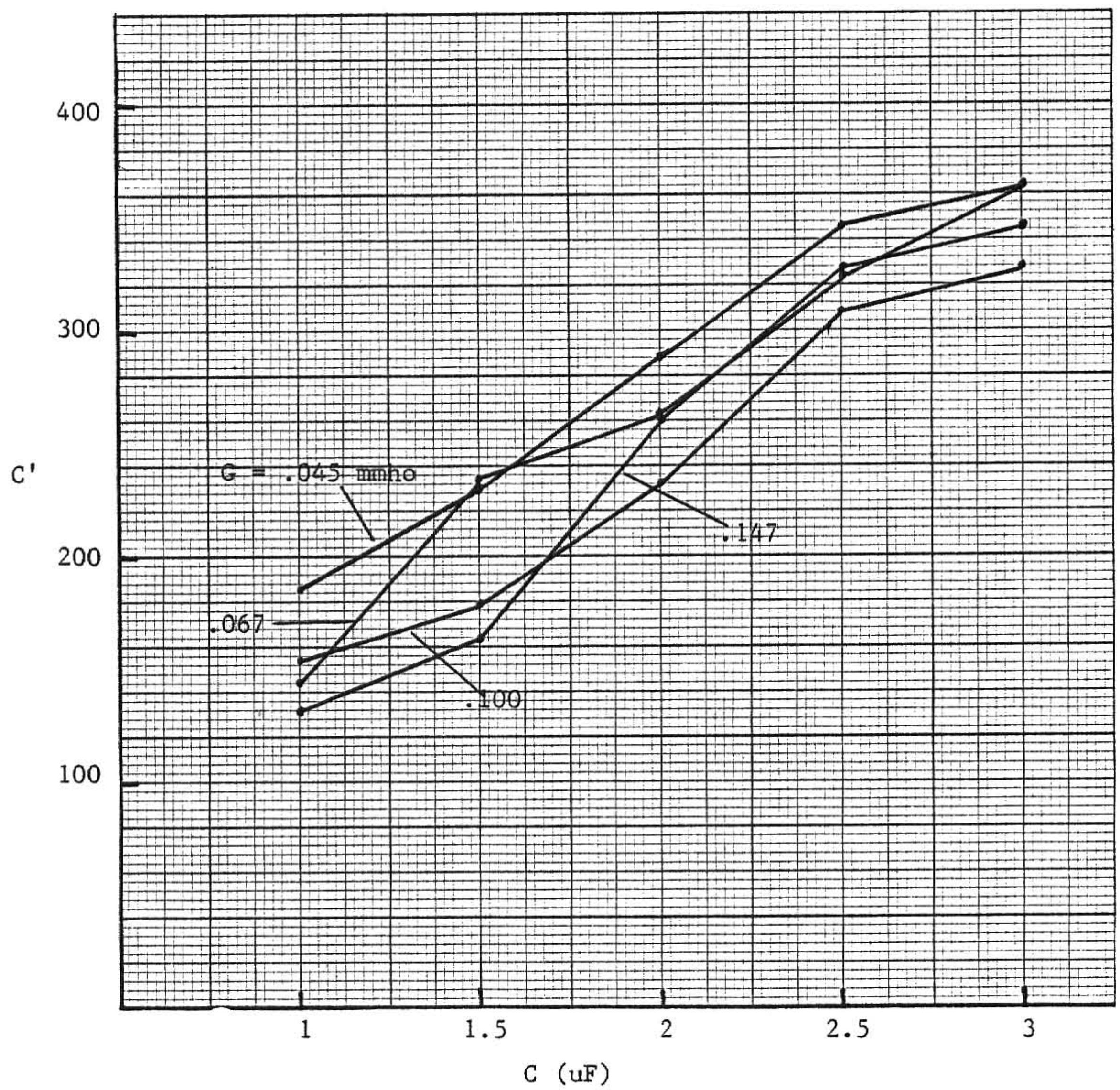

Figure 13. Measured Capacitance vs. Actual Capacitance,
8-bit System

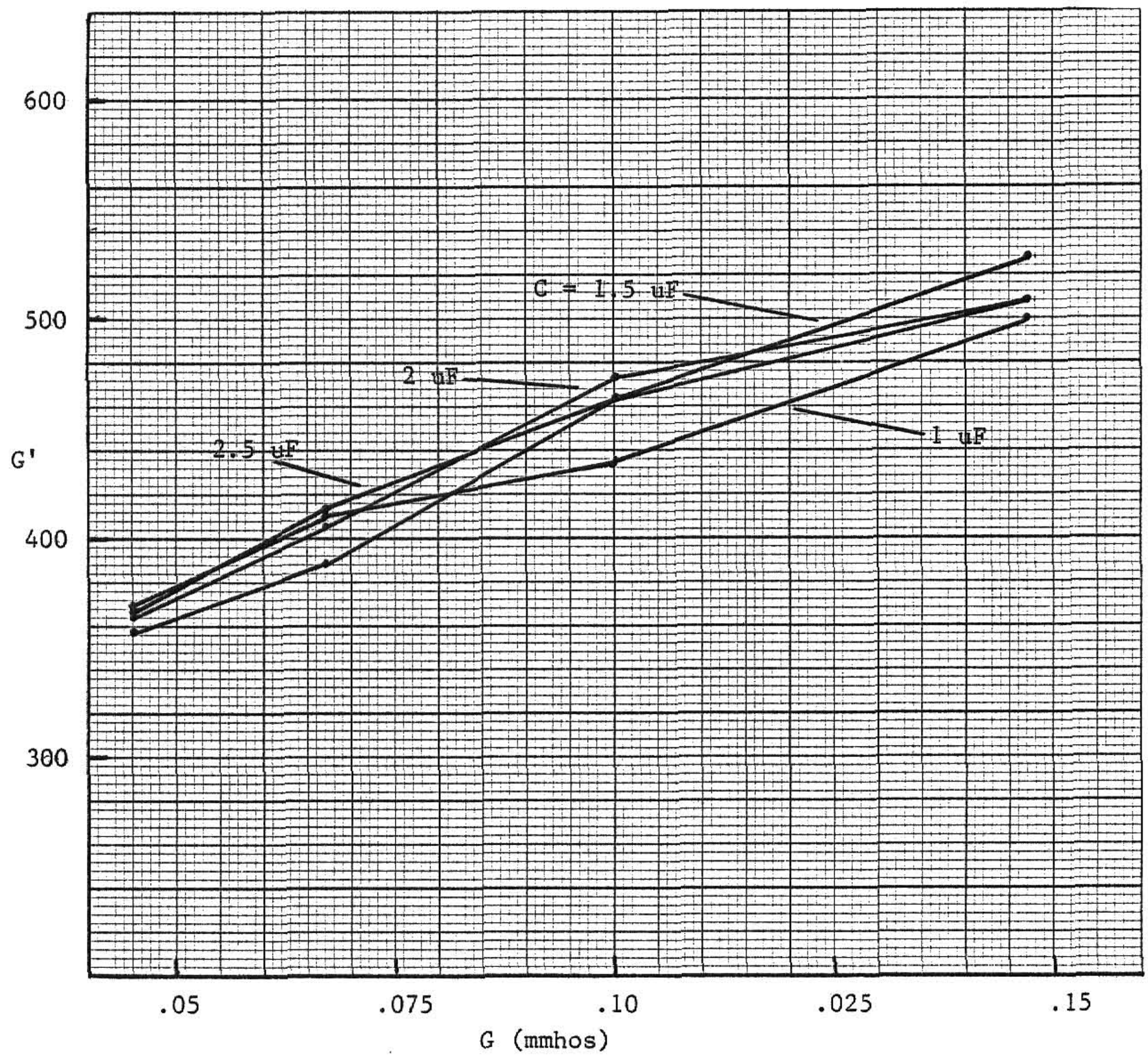

Figure 14. Measured Conductance vs. Actual Conductance
8-bit System

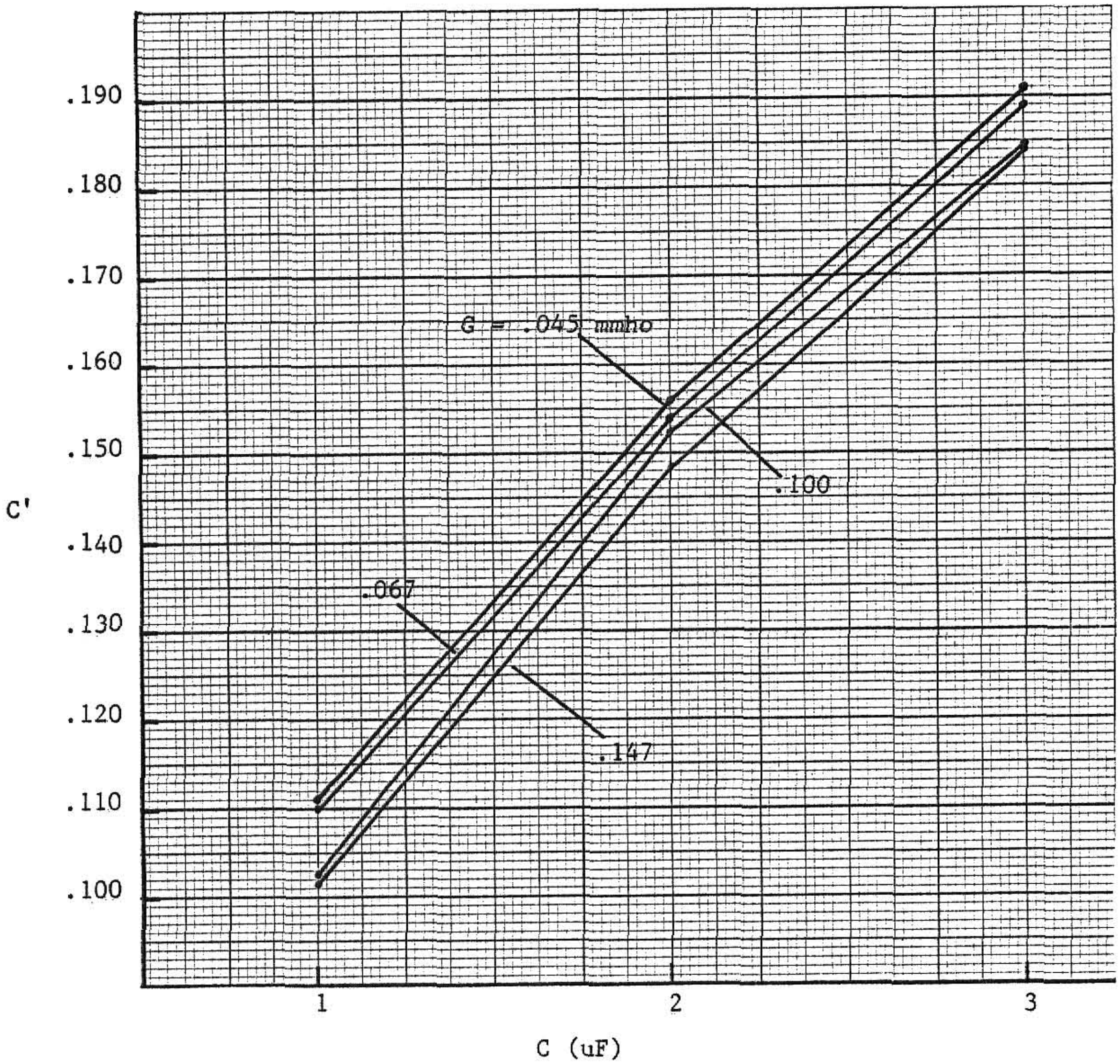

Figure 15. Measured Capacitance vs. Actual Capacitance, 12-bit System

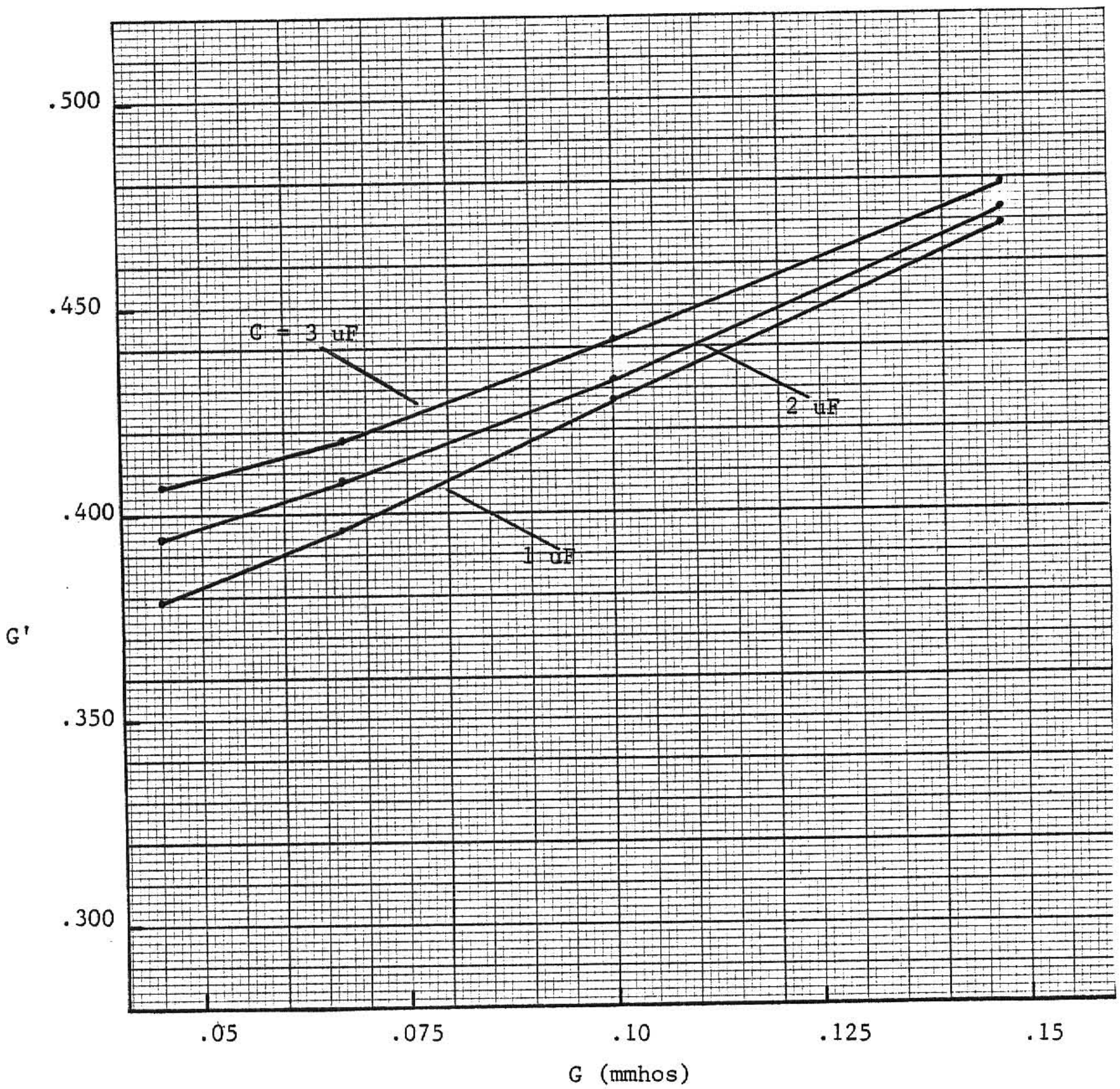

Figure 16. Measured Conductance vs. Actual Conductance, 12-bit System

of calculation used requires the extra accuracy of the 12-bit values and floating-point arithmetic to produce reliable results. This was one of the setbacks in producing the desired output: the accuracy required of the output, in combination with a complex calculation which enhanced any precision errors, necessitated more precise measuring equipment than was originally thought. Still, the experiment indicated that the measurement algorithm does indeed work, and impedance can be calculated from current measurements, given enough precision of the system components.

Some sources of the errors that do exist in the 12-bit system are low measuring frequency and inaccuracies in amplitude measurement. The frequency of the sine wave stimulus voltage is limited by the speed of an A-D conversion. In the eight-bit system, the maximum stimulus frequency is 7.5 Hz, while the 12-bit system (having a much faster A-D) operates at 106 Hz. Capacitance is normally measured at 1 kHz, however, and since real capacitances vary with frequency, an error may be introduced here. Another source of error may be the peak detection method. A sample value of the output current is taken every time a new sine wave value is loaded into the D-A. Theoretically, this should be often enough to be sure that a measurement is taken when the current is at its peak. However, it is possible that the signal droops before the A-D samples it, thus producing slightly inaccurate responses.

Solutions for these problems would involve the use of more hardware. A voltage-controlled oscillator or similar device could be used to provide computer-controlled (instead of computer-generated) frequencies in the 1 kHz range. A hardware peak-detector would solve the A-D drooping signal problem. Further investigations into this technique for measuring impedance should look into these areas.

The experiment is illustrative of the types of problems that must be solved in the implementation of a smart transducer system. One problem is determining the precision of the transducer and related parts that is required to produce the desired accuracy in the final result. Another is in deriving the stimulus signal. A good deal of time can be spent obtaining the proper method of generating the look-up tables for the two sine waves. To create a very good signal, one must know a great deal about digital synthesis of analog signals, which is the subject of many books and papers. Other areas of difficulty may be the design of interface circuitry between the transducer and the microprocessor, and in making the choice of which A-D or other device to use. There appear to be about the same number of problems in getting this system to work as there are in getting any new piece of hardware to operate properly. While it may not be the best method of measuring impedance, the experimental system does illustrate some of the techniques needed in combining the intelligence of a computer with the measuring abilities of a transducer.

Appendix 2

Programs for Impedance Measuring System

```
;
;               MAIN PROGRAM
;    AVERAGES EIGHT READINGS AND CALCULATES G AND C FROM THE
;    AVERAGE.  STORES CAPACITANCE IN LOCATION "CVAL" AND
;    CONDUCTANCE IN LOCATION "GVAL".
;
5000 21 00 00          LD   HL,0000H        CLEAR STORAGE
5003 22 02 61          LD   (DATA1),HL
5006 22 04 61          LD   (DATA2),HL
5009 06 08             LD   B,08H           # OF READINGS TO AVERAGE
500B 21 00 74    LP0:  LD   HL,TABLE1       PASS LOW FREQ. TABLE ADDR.
500E 22 00 61          LD   (TABADR),HL
5011 CD FE 6F          CALL MEASURE         TAKE READING INTO HL

5014 C5                PUSH BC              DIVIDE BY NUMBER OF READINGS
5015 06 03             LD   B,03H               "         "
5017 CB 3C       LP1:  SRL  H                   "         "
5019 CB 3D             SRL  L                   "         "
501B 10 FA             DJNZ LP1                 "         "
501D C1                POP  BC                  "         "

501E 3A 02 61          LD   A,(V1)          UPDATE ACCUMULATED AVERAGES
5021 85                ADD  A,L                FOR V1 AND I1
5022 32 02 61          LD   (V1),A              "         "
5025 3A 03 61          LD   A,(I1)              "         "
5028 84                ADD  A,H                 "         "
5029 32 03 61          LD   (I1),A              "         "

502C 21 20 74          LD   HL,TABLE2       PASS HIGH FREQ. TABLE ADDR.
502F 22 00 61          LD   (TABADR),HL         "         "
5032 CD FE 6F          CALL MEASURE         HL = READING

5035 C5                PUSH BC              DIVIDE BY NUMBER OF READINGS
5036 06 03             LD   B,03H               "         "
5038 CB 3C       LP2:  SRL  H                   "         "
503A CB 3D             SRL  L                   "         "
503C 10 FA             DJNZ LP2                 "         "
503E C1                POP  BC                  "         "

503F 3A 04 61          LD   A,(V2)          UPDATE ACCUMULATED AVERAGES
5042 85                ADD  A,L                FOR V2 AND I2
5043 32 04 61          LD   (V2),A              "         "
5046 3A 05 61          LD   A,(I2)              "         "
5049 84                ADD  A,H                 "         "
504A 32 05 61          LD   (I2),A              "         "

504D 10 BC             DJNZ LP0             CONTINUE FOR ALL READINGS
504F 00
```

```
;
;          CAPACITANCE CALCULATION
;
5050 ED 5B 03 61     LD   DE,(DATA1+1)   DE = I1
5054 16 00           LD   D,00H            "
5056 2A 04 61        LD   HL,(DATA2)     HL = V2
5059 62              LD   H,D              "
505A CD C0 60        CALL MULT           HL = I1 X V2
505D 54              LD   D,H            HL',HL = (I1 X V2)^2
505E 5D              LD   E,L                 "         "
505F CD C0 60        CALL MULT                "         "
5062 D9              EXX                 SAVE H.O. 16 BITS
5063 E5              PUSH HL'                 "         "
5064 D9              EXX                      "         "

5065 ED 5B 02 61     LD   DE,(DATA1)     DE = V1
5069 16 00           LD   D,00H            "
506B 2A 05 61        LD   HL,(DATA2+1)   HL = I2
506E 62              LD   H,D              "
506F CD C0 60        CALL MULT           HL = V1 X I2
5072 54              LD   D,H            HL',HL = (V1 X I2)^2
5073 5D              LD   E,L                 "         "
5074 CD C0 60        CALL MULT                "         "
5077 D9              EXX                 SAVE H.O. 16 BITS
5078 E5              PUSH HL'                 "         "
5079 D9              EXX                      "         "

507A E1              POP  HL             HL = (V1 X I2)^2
507B D1              POP  DE                  - (V2 X I1)^2
507C 37              SCF                      "         "
507D 3F              CCF                      "         "
507E ED 52           SBC  HL,DE               "         "
5080 CB 7C           BIT  07H,H          CHECK SIGN OF ANSWER
5082 28 0A           JR   Z,ARD          AND TAKE ABSOLUTE VALUE
5084 7C              LD   A,H                 "         "
5085 2F              CPL                      "         "
5086 67              LD   H,A                 "         "
5087 7D              LD   A,L                 "         "
5088 2F              CPL                      "         "
5089 6F              LD   L,A                 "         "
508A 11 01 00        LD   DE,0001H            "         "
508D 19              ADD  HL,DE               "         "
508E 22 06 61   ARD: LD   (ANSWER),HL         "         "
```

```
5091 CD 60 60       CALL SQRT         (F) = SQRT(ANSWER)

5094 ED 5B 02 61    LD   DE,(DATA1)   HL = V1 X V2
5098 2A 04 61       LD   HL,(DATA2)        "
509B 16 00          LD   D,00H             "
509D 62             LD   H,D               "
509E CD C0 60       CALL MULT              "

50A1 EB             EX   DE,HL        HL = DIVISOR
50A2 5A             LD   E,D          MSB OF HL = (F)
50A3 16 00          LD   D,00H             "
50A5 2A 07 61       LD   HL,(F-1)          "
50A8 2E 00          LD   L,00H             "

50AA CD 80 61       CALL DIVIDE       HL = (F) / (V1 X V2)
50AD 22 0B 61       LD   (CVAL),HL         "         "
50B0 C3 00 51       JP   GCALC
```

```
;
;                    CONDUCTANCE CALCULATION
;
5100 ED 5B 03 61     LD   DE,(DATA1+1)    DE = I1
5104 16 00           LD   D,00H              "
5106 2A 04 61        LD   HL,(DATA2)      HL = V2
5109 62              LD   H,D                "
510A CD C0 60        CALL MULT            HL = V2 X I1

510D 54              LD   D,H             HL',HL = (V2 X I1)^2
510E 5D              LD   E,L                  "          "
510F CD C0 60        CALL MULT                 "          "
5112 D9              EXX
5113 29              ADD  HL',HL'         MULTIPLY H.O. 16 BITS
5114 29              ADD  HL',HL'         BY FOUR, AND SAVE IT
5115 E5              PUSH HL'                  "          "
5116 D9              EXX                       "          "

5117 ED 5B 02 61     LD   DE,(DATA1)      DE = V1
511B 16 00           LD   D,00H              "
511D 2A 05 61        LD   HL,(DATA2+1)    HL = I2
5120 62              LD   H,D                "
5121 CD C0 60        CALL MULT            HL = V1 X I2

5124 54              LD D,H               HL',HL = (V1 X I2)^2
5125 5D              LD E,L                    "          "
5126 CD C0 60        CALL MULT                 "          "
5129 D9              EXX                  SAVE H.O 16 BITS
512A E5              PUSH HL'                  "          "
512B D9              EXX                       "          "

512C E1              POP  HL              HL = (V1 X I2)^2
512D D1              POP  DE                   - (V2 X I1)^2
512E 37              SCF                       "          "
512F 2F              CCF                       "          "
5130 ED 52           SBC  HL,DE                "          "

5132 CB 7C           BIT  07H,H           CHECK SIGN OF ANSWER
5134 28 0C 00 00     JR   Z,ARD           AND TAKE ABSOLUTE VALUE
5138 7C              LD   A,H                  "          "
5139 2F              CPL                       "          "
513A 67              LD   H,A                  "          "
513B 7D              LD   A,L                  "          "
513C 2F              CPL                       "          "
513D 6F              LD   L,A                  "          "
513E 11 01 00        LD   DE,0001H             "          "
5141 19              ADD  HL,DE                "          "
5142 22 06 61  ARD:  LD   (ANSWER),HL          "          "
```

```
5145 CD 60 60       CALL SQRT          (F) = SQRT(ANSWER)

5148 ED 5B 02 61    LD   DE,(DATA1)    HL = V1 X V2
514C 2A 04 61       LD   HL,(DATA2)         "
514F 16 00          LD   D,00H              "
5151 62             LD   H,D                "
5152 CD C0 60       CALL MULT               "

5155 ED 5B 08 61    LD   DE,(F)        DIVISOR INTO DE
5159 16 00          LD   D,00H              "
515B CD 80 61       CALL DIVIDE
515E 22 0D 61       LD   (GVAL),HL     STORE CONDUCTANCE VALUE
5161 C3 00 0E       JP   MONITOR       RETURN TO MONITOR
```

```
;
;              SQUARE ROOT ROUTINE
;    BASIC PRINCIPLE: THE SUM OF N ODD NUMBERS IS N SQUARED
;    ROUTINE SUMS N ODD NUMBERS UNTIL THE VALUE IS GREATER THAN
;    THE ARGUMENT. THE RESULTING VALUE OF N IS TAKEN AS THE SQUARE
;    ROOT.
;    ENTRY: ARGUMENT X^2 IN LOCATION "ANSWER"
;    EXIT:  SQUARE ROOT X IN LOCATION "F"
;
6060 C5          SQRT: PUSH BC              SAVE STATUS
6061 D5                PUSH DE                  "
6062 E5                PUSH HL                  "
6063 ED 5B 06 61       LD   DE,(ANSWER)     DE = X^2
6067 21 00 00          LD   HL,0000H        HL = APPROX. TO X^2
606A 01 01 00          LD   BC,0001H        BC = NTH ODD NUMBER

606D 09          CONT: ADD HL,BC            UPDATE APPROXIMATION
606E 37                SCF                  IS IT GREATER THAN X^2?
606F 3F                CCF                      "          "
6070 E5                PUSH HL                  "          "
6071 ED 52             SBC  HL,DE               "          "
6073 E1                POP  HL                  "          "
6074 30 04             JR   NC,LARG         IF YES, JUMP
6076 03                INC  BC              NO, SO GET NEXT ODD NUMBER
6077 03                INC  BC                  "          "
6078 18 F3             JR   CONT

607A 03          LARG: INC BC               CONVERT NTH ODD NUMBER
607B CB 38             SRL  B                   TO N ITSELF
607D CB 19             RR   C                   "          "
607F 79                LD   A,C             STORE N
6080 32 08 61          LD   (F),A             "

6083 E1                POP  HL              RESTORE STATUS
6084 D1                POP  DE                  "
6085 C1                POP  BC                  "
6086 C9                RET
```

```
;
;              16-BIT MULTIPLY ROUTINE
;
;    ENTRY: MULTIPLER IN DE, MULTIPLICAND IN HL
;    EXIT:  H.O. RESULT IN HL', L.O. RESULT IN HL
;
60C0 C5          MULT: PUSH BC               SAVE STATUS
60C1 F5                PUSH AF                  "
60C2 06 10             LD   B,10H            # OF BITS TO BE MULTIPLIED
60C4 4A                LD   C,D              MULTIPLIER INTO C AND A
60C5 7B                LD   A,E                 "           "
60C6 EB                EX   DE,HL            MULTIPLICAND INTO DE
60C7 21 00 00          LD   HL,0000H         CLEAR L.O. RESULT
60CA D9                EXX                   CLEAR H.O. RESULT
60CB 21 00 00          LD   HL',0000H           "           "
60CE 55                LD   D',L'            CLEAR DE'
60CF 5D                LD   E',L'               "
60D0 D9                EXX

60D1 CB 39        TOP: SRL  C                SHIFT MULTIPLIER RIGHT
60D3 1F                RRA                   (NEXT BIT INTO CARRY)
60D4 30 05             JR   NC,SHIFT         JUMP IF THAT BIT IS ZERO
60D6 19                ADD  HL,DE            ELSE, ADD L.O. MULTIPLICAND
60D7 D9                EXX
60D8 ED 5A             ADC  HL',DE'          ADD H.O. MULTIPLICAND + CARRY
60DA D9                EXX

60DB EB         SHIFT: EX   DE,HL            SHIFT L.O. MULTIPLICAND LEFT
60DC 29                ADD  HL,HL                    ONE BIT
60DD EB                EX   DE,HL               "           "
60DE D9                EXX
60DF EB                EX   DE',HL'          SHIFT H.O. MULTIPLICAND LEFT
60E0 ED 6A             ADC  HL',HL'                  ONE BIT
60E2 EB                EX   DE',HL'             "           "
60E3 D9                EXX

60E4 10 EB             DJNZ TOP              CONTINUE FOR 16 BITS
60E6 F1                POP  AF               RESTORE STATUS
60E7 C1                POP  BC                  "
60E8 C9                RET
```

```
;              16-BIT DIVIDE ROUTINE
;
;    ENTRY: DIVISOR IN DE, DIVIDEND IN HL
;    EXIT:  QUOTIENT IN HL
;
6180 C5          DIVIDE: PUSH BC            SAVE STATUS
6181 DD E5               PUSH IX              "
6183 06 10               LD   B,10H         # OF BITS TO DIVIDE
6185 22 09 61            LD   (TEMP),HL     IX = DIVIDEND (FOR STORAGE)
6188 DD 2A 09 61         LD   IX,(TEMP)       "         "
618C 21 00 00            LD   HL,0000H      CLEAR QUOTIENT
618F DD 29           LP: ADD  IX,IX         SHIFT DIVIDEND AND QUOTIENT
6191 ED 6A               ADC  HL,HL             LEFT ONE BIT
6193 37                  SCF                CLEAR CARRY BIT
6194 3F                  CCF                  "
6195 ED 52               SBC  HL,DE         SUBTRACT DIVISOR
6197 38 04               JR   C,ADD         IF BORROW, JUST SHIFT
6199 DD 23               INC  IX            ELSE, ADD ONE TO QUOTIENT
619B 18 01               JR   DECR
619D 19             ADD: ADD  HL,DE         RESTORE QUOTIENT
619E 10 EF         DECR: DJNZ LP            CONTINUE FOR 16 BITS

61A0 DD 22 09 61         LD   (TEMP),IX     PUT ANSWER INTO HL
61A4 2A 09 61            LD   HL,(TEMP)       "         "
61A7 DD E1               POP  IX            RESTORE STATUS
61A9 C1                  POP  BC              "
61AA C9                  RET
```

```
;
;                  MEASURING ROUTINE
;     ENTRY: ADDRESS OF SINE WAVE TABLE STORE IN "TABADR"
;     EXIT:  PEAK VALUE OF CURRENT IN H
;            PEAK VALUE OF VOLTAGE IN L
;
6FFE C5     MEASURE: PUSH BC             SAVE STATUS
6FFF D5              PUSH DE                 "
7000 2A 00 61        LD   HL,(TABADR)    COPY SINE WAVE TABLE TO
7003 11 00 73        LD   DE,7300H       WORKING MEMORY (ALLOWS RESET
7006 01 20 00        LD   BC,0020H       WITHOUT DATA LOSS)
7009 ED B0           LDIR

700B 01 00 75        LD   BC,7500H       INIT READINGS PTR
700E DD 21 1E 71     LD   IX,711EH       INIT PTR TO COUNTERS IN RAM
7012 16 01           LD   D,01H          INIT COUNTER OF SINE VALUES/READING
7014 21 00 73   TOP: LD   HL,7300H       INIT PTR TO SINE VALUES TABLE

7017 3E 60     CONT: LD   A,60H          OUTPUT HIGH FOUR BITS OF
7019 ED 6F           RLD                 SINE VALUE TO D-A
701B D3 00           OUT  0,A                "         "
701D F6 E0           OR   E0H                "         "
701F D3 00           OUT  0,A                "         "
7021 E6 AF           AND  AFH            OUTPUT LOW FOUR BITS OF
7023 ED 6F           RLD                 SINE VALUE TO D-A
7025 D3 00           OUT  0,A                "         "
7027 F6 E0           OR   E0H                "         "
7029 D3 00           OUT  0,A                "         "
702B ED 6F           RLD                     "         "

702D 15              DEC  D              TIME TO TAKE A READING?
702E CA 48 70        JP   Z,READ         YES, SO JUMP
7031 DD 36 00 01     LD   (IX+0),01H     DELAY EQUAL TO A-D
7035 DD 36 01 03 LP1: LD  (IX+1),E3H       CONVERSION TIME
7039 DD 35 01   LP2: DEC  (IX+1)             "         "
703C C2 39 70        JP   NZ,LP2             "         "
703F DD 35 00        DEC  (IX+0)             "         "
7042 C2 35 70        JP   NZ,LP1             "         "
7045 C3 66 70        JP   REST
```

```
7048 3E C2      READ: LD   A,C2H        SET CONTROLS TO MEASURE
704A D3 00            OUT  0,A          STIMULUS VOLTAGE
704C F6 E0            OR   E0H              "          "
704E D3 00            OUT  0,A              "          "
7050 CD 70 70         CALL A/D          MEASURE IT
7053 3E C1            LD   A,C1H        SET CONTROLS TO MEASURE
7055 D3 00            OUT  0,A          OUTPUT CURRENT
7057 F6 E0            OR   E0H              "          "
7059 D3 00            OUT  0,A              "          "
705B CD 70 70         CALL A/D          MEASURE IT

705E 79               LD   A,C          ENOUGH READINGS TAKEN?
705F FE 00            CP   00H              "          "
7061 CA 00 72         JP   Z,LARGEST    YES, FIND MAGNITUDE
7064 16 01            LD   D,01H        NO, RESET VALUES/READING COUNTER
7066 23         REST: INC  HL           MOVE SINE VALUE PTR
7067 7D               LD   A,L          AT END OF SINE VALUES TABLE?
7068 FE 20            CP   20H              "          "
706A CA 14 70         JZ   TOP          YES, SO RESTART
706D C3 17 70         JP   CONT         NO, SO CONTINUE
```

```
;
;              A-D ROUTINE
;     PERFORMS A CONVERSION, THEN STORES RESULT
;     ENTRY: BC REGISTER PAIR POINTS TO CURRENT DATA STORAGE LOCATION
;     EXIT:  A-D READING STORED, BC INCREMENTED
;
7070 DD 36 00 01 A/D: LD  (IX+0),01H      SWITCH-SETTLING DELAY
7074 DD 36 01 E3      LD  (IX+1),E3H         "          "
7078 DD 35 01         DEC (IX+1)             "          "
707B 20 FB            JR  NZ,A/D+8           "          "
707D DD 35 00         DEC (IX+0)             "          "
7080 20 F2            JR  NZ,A/D+4           "          "
7082 3E F0            LD  A,F0H           INITIATE A-D CONVERSION
7084 D3 00            OUT 0,A                "          "
7086 E6 EF            AND EFH                "          "
7088 D3 00            OUT 0,A                "          "
708A DB 06     WAIT:  IN  A,6             CONVERSION DONE?
708C E6 02            AND 02H                "          "
708E CA 8A 70         JP  Z,WAIT          NO, SO WAIT
7091 DB 00            IN  A,0             INPUT READING
7093 02               LD  (BC),A          STORE IT
7094 3E 30            LD  A,30H           RESET CONVERSION-DONE FLAG
7096 DE 06            OUT 6,A                "          "
7098 03               INC BC              MOVE READINGS POINTER
7099 C9               RET
```

```
;
;                  MAGNITUDE-FINDING ROUTINE
;     ENTRY: BC POINTS AFTER LAST DATA PAIR STORED
;     EXIT:  H CONTAINS LARGEST CURRENT VALUE
;            L CONTAINS LARGEST VOLTAGE VALUE
;
7200 21 00 00         LD   HL,0000H          INIT STORAGE
7203 0B          UP:  DEC  BC               POINT TO NEXT DATA
7204 79               LD   A,C              ALL DONE?
7205 FE C1            CP   C1H                   "
7207 28 10            JR   Z,OUT            YES, SO RETURN TO MAIN PROG.
7209 0A               LD   A,(BC)           NO, BRING IN DATA
720A BD               CP   L                IS DATA LARGER?
720B D9 0F 72         JP   C,DN             NO, SO JUMP
720E 6F               LD   L,A              YES, UPDATE "LARGEST" VOLTAGE
720F 0B          DN:  DEC  BC               POINT TO NEXT DATA
7210 0A               LD   A,(BC)           BRING IN DATA
7211 BC               CP   H                IS DATA LARGER?
7212 D9 03 72         JP   C,UP             NO, SO JUMP
7215 67               LD   H,A              YES, UPDATE "LARGEST" CURRENT
7216 C3 03 72         JP   UP
7219 D1          OUT: POP  DE               RESTORE STATUS
721A C1               POP  BC                   "
721B C9               RET
```

```
;
;               DATA STORAGE MAP
;
6100 __ __      TABADR: (2 BYTES)
6102 V1 I1      DATA1:  (2 BYTES)
6104 V2 I2      DATA2:  (2 BYTES)
6106 __ __      ANSWER: (2 BYTES)
6108 __         F:      (1 BYTE)
6109 __ __      TEMP:   (2 BYTES)
610A __ __      CVAL:   (2 BYTES)
610C __ __      GVAL:   (2 BYTES)
;
;
;               SINE WAVE TABLES
;
;    TABLE1     (1 CYCLE OF LOW FREQUENCY)
7400 80 99 B1 C7  DA EA F5 FD  FF FD F5 EA  DA C7 B1 99
7410 80 67 50 3A  26 17 0B 03  01 03 0B 16  26 39 4F 67
;
;    TABLE2     (2 CYCLES OF HIGH FREQUENCY)
7420 80 B1 DA F5  FF F5 DA B1  80 50 26 0B  01 0B 26 4F
7430 80 B0 D9 F5  FF F6 DA B1  8A 50 27 0B  01 0A 26 4F
```

BASIC PROGRAM TO

GENERATE SINE WAVE VALUES

```
10 INPUT "NUMBER OF LEVELS: ",B
12 B=B/2-1
15 INPUT "NUMBER OF TIME STEPS: ",N
17 L=FND("7400")
18 INPUT "NUMBER OF CYCLES: ",C
20 FOR I=0 TO N-1 STEP 1
30 Q=INT(B*SIN(6.28*C*I/N)+.5)
40 FILL L+I,Q+128
50 NEXT
60 END
```

```
C       THIS PROGRAM CALLS THE MACHINE LANGUAGE SUBPROGRAM MEAS
C       AND MEASURES THE IMPEDANCE OF THE TEST CIRCUIT, WHICH IS
C       ASSUMED TO BE A PARALLEL COMBINATION OF R AND C.
C
        DIMENSION IARRAY(1024),ARRAY(1024)
C
        TYPE 49
49      FORMAT(' ENTER NUMBER OF MEASUREMENTS, <RETURN>, THEN IDELAY')
        ACCEPT 50,N
        ACCEPT 50,IDELAY
50      FORMAT (I4)
C
C       TAKE INPUT AND OUTPUT READINGS
C
        CALL MEAS (IARRAY,N,IDELAY,IERR)
        IF (IERR.GT.0) TYPE 100
C
C       CONVERT TO VOLTAGE
C
        DO 10 I=1,2*N
        ITEMP=IARRAY(I).AND.'7777
        CALL DIGVLT (ITEMP,TEMP)
        ARRAY(I)=TEMP
10      CONTINUE
C
        TYPE 199
199     FORMAT ('$OUTPUT READINGS? (Y/N) ')
        ACCEPT 201,A
201     FORMAT (A1)
        IF (A.EQ.'Y') TYPE 200,((IARRAY(I),ARRAY(I)),I=1,2*N)
C
C       FIND MAXIMUM AND MINIMUM VALUES OF INPUT AND OUTPUT
C
        V1=0
        V2=0
        C1=0
        C2=0
        MV1=0
        MV2=0
        MC1=0
        MC2=0
        DO 20 I=1,N,2
        IF (ARRAY(I).GT.V1) V1=ARRAY(I)
        IF (ARRAY(I).LT.MV1) MV1=ARRAY(I)
        IF (ARRAY(I+1).GT.C1) C1=ARRAY(I+1)
        IF (ARRAY(I+1).LT.MC1) MC1=ARRAY(I+1)
20      CONTINUE
C
        DO 25 I=N+1,2*N,2
        IF (ARRAY(I).GT.V2) V2=ARRAY(I)
        IF (ARRAY(I).LT.MV2) MV2=ARRAY(I)
        IF (ARRAY(I+1).GT.C2) C2=ARRAY(I+1)
        IF (ARRAY(I+1).LT.MC2) MC2=ARRAY(I+1)
25      CONTINUE
```

```
C
C       REMOVE D.C. OFFSET
C
        V1=(V1-MV1)/2
        V2=(V2-MV2)/2
        C1=(C1-MC1)/2
        C2=(C2-MC2)/2
C
C       CALCULATE CAPACITANCE, RESISTANCE, AND CONDUCTANCE
C
        TYPE 300,V1,C1,V2,C2
        IF (V1**2*C2**2-2*V2**2*C1**2.EQ.0) GOTO 26
        IF (V1*V2.EQ.0) GOTO 26
        R=V1*V2/(SQRT(ABS(V1**2*C2**2-4*V2**2*C1**2)))
        G=1/R
        C=SQRT(ABS(V2**2*C1**2-V1**2*C2**2))/(V1*V2)
        TYPE 400,C,R,G
        STOP
26      STOP 'DIVISION BY ZERO'
100     FORMAT(' ERROR IN READING')
200     FORMAT(' ',I6,F10.2,4X,I6,F10.2)
300     FORMAT(' LOW VOLTAGE:',F10.2,'  LOW CURRENT:',F10.2,
        */,' HIGH VOLTAGE:',F10.2,'  HIGH CURRENT:',F10.2)
400     FORMAT(' CAPACITANCE',F10.4,'  RESISTANCE',F10.4,'  CONDUCTANCE',
        *F10.4)
        END
```

```
;       PROGRAM TO OPERATE IMPEDANCE SYSTEM
;       USES A/D 0 FOR INPUT VOLTAGE
;            A/D 1 FOR OUTPUT CURRENT
;            D/A 0 FOR SINEWAVE OUTPUT
;
;       CALLING FORMAT:
;
;               CALL MEAS(IARRAY,N,IDELAY,IERR)
;
;       WHERE   IARRAY IS AN ARRAY IN WHICH THE READINGS WILL BE STORED
;                   (IN PAIRS, INPUT VOLTAGE FIRST, THEN OUTPUT CURRENT)
;               N IS THE NUMBER OF READINGS (TWO ELEMENTS IN IARRAY EACH)
;               IDELAY IS A LOOP COUNTER TO SET SINE WAVE FREQUENCY
;               IERR IS A/D ERROR FLAG: IERR=1 MEANS ERROR
;
        .GLOBL  MEAS
MEAS:   MOV     2(R5),R0                ;R0=START OF IARRAY
        MOV     @4(R5),R1               ;R1=NUMBER OF READINGS
        CLR     @8.(R5)                 ;CLEAR IERR
;
        MFPS    R4                      ;DISABLE INTERRUPTS
        BIS     #200,R4
        MTPS    R4
;
        MOV     #TABLE1,TABADR          ;MEASURE AT LOWER FREQUENCY
        JSR     PC,DOIT
        MOV     @4(R5),R1               ;R1=NUMBER OF READINGS
        MOV     #TABLE2,TABADR          ;MEASURE AT HIGHER FREQUENCY
        JSR     PC,DOIT
;
        MFPS    R4                      ;ENABLE INTERRUPTS
        BIC     #200,R4
        MTPS    R4
        RTS     PC
;
DOIT:   MOV     TABADR,R3               ;R3 IS PTR TO SINE TABLE
CONT:   MOV     (R3)+,@DAREG            ;OUTPUT SINE VALUE
;
        MOV     @6(R5),R2               ;DELAY
1$:     DEC     R2
        BGE     1$
;
        CLR     @ADCSR                  ;SELECT INPUT VOLTAGE
        INC     @ADCSR                  ;START CONVERSION
2$:     TSTB    @ADCSR                  ;CONVERSION DONE?
        BPL     2$                      ;IF NOT, WAIT
        TST     @ADCSR                  ;WAS THERE AN ERROR?
        BMI     OOPS
;
        MOV     @ADDBR,(R0)+            ;STORE VALUE
        DEC     R1                      ;ENOUGH VALUES TAKEN?
        BLE     OUT                     ;JUMP IF TRUE
;
        MOV     #400,@ADCSR             ;SELECT OUTPUT CURRENT
        INC     @ADCSR                  ;START CONVERSION
3$:     TSTB    @ADCSR                  ;CONVERSION DONE?
        BPL     3$                      ;IF NOT, WAIT
        TST     @ADCSR                  ;WAS THERE AN ERROR?
        BMI     OOPS                    ;JUMP IF SO
```

```
;
        MOV     @ADDBR,(R0)+            ;STORE VALUE
        DEC     R1                      ;ENOUGH VALUES TAKEN?
        BLE     OUT                     ;JUMP IF TRUE
;
        CMP     R3,#TABLE2+64.          ;ALL SINE VALUES OUTPUT?
        BEQ     DOIT                    ;IF SO, START AT BEGINNING
        CMP     R3,#TABLE1+64.          ;ALL SINE VALUES OUTPUT?
        BEQ     DOIT                    ;IF SO, START AT BEGINNING
        BR      CONT                    ;IF NOT, CONTINUE
OUT:    RTS     PC
OOPS:   INC     @8.(R5)                 ;SET IERR = 1
        RTS     PC
;
        .EVEN
TABLE1: .WORD   4000,4617,5417,6161,6647,7245,7543,7727,7777
        .WORD   7730,7544,7247,6651,6163,5422,4622,4003,3164
        .WORD   2364,1622,1133,534,237,51,1,47,233,527,1125,1612,2353,3152
TABLE2: .WORD   4000,5417,6647,7543,7777,7544,6651,5422,4003
        .WORD   2364,1133,237,1,233,1125,2353,3771,5411,6642,7540
        .WORD   7777,7546,6656,5430,4012,2372,1140,241,1,230,1120,2345
TABADR: .BLKW   1
ADCSR:  170400
ADDBR:  170402
DAREG:  170440
        .END
```

SOME USES OF A MICROPROCESSOR IN TRANSDUCER SYSTEMS

by

DWIGHT R. PALMER

B. S., Kansas State University, 1979

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1981

ABSTRACT

The power of a microprocessor can be applied to the problem of measurement with excellent results. The uses of computing power in transducer systems can be classified into two groups: passive, in which the microcomputer is used as a calculating device, and active, in which it is used as a controlling device.

Passive systems perform calculations on the transducer output, which allow scaling of results, reduction of non-linearities and dependence on unwanted parameters, the use of digital filtering techniques, and reduction of systematic errors, among other benefits. Active systems put the microcomputer in charge of the measurement process, with the ability to change nearly every system parameter. As a result, esoteric measurement procedures, which before were too complicated to be implemented in an automatic system, can now be used with a minimum of effort. Other benefits include automatic range switching, the use of complicated stimulus signals, and increased reliability. Both techniques can make possible new types of transducer systems, those that are too complex to implement economically in hardware alone, as well as those that are impossible to build without a microcomputer.

These attributes of "smart" sensor systems are discussed in detail, with example systems taken from the literature. The design and operation of an experimental impedance-measuring device using a novel measurement algorithm and illustrating both passive and active methods is explained in the Appendices.