JOURNAL OF RESEARCH of the National Bureau of Standards
Volume 83, No. 5, September–October 1978

A Microprocessor Controlled Potentiostat for Electrochemical Measurements*

M. I. Cohen and P. A. Heimann**

National Measurement Laboratory, National Bureau of Standards, Washington, D. C. 20234

(April 21, 1978)

A system, utilizing a microprocessor, intended for the control and unattended operation of a standard laboratory potentiostat is described. The system consists of a central processing unit, 16 kilobytes of random access memory, peripheral interfacing, a timer and digital to analog and analog to digital converters. It allows flexible operation of the potentiostat by programming of the central processor.

Key words: Control; converter (digital and analog); electrochemical measurements; microcomputer; microprocessor; potentiostat.

1. Introduction

The advent of the microprocessor (μP) [1][1] has opened up a relatively low cost approach to digital techniques for control and data handling in the laboratory. Although somewhat limited in throughput and "number crunching" ability (as compared to minicomputers), these devices are more than adequate for many applications in instrument control.

For our purposes, the device is considered as a Central Processing Unit (CPU) on a single chip capable (at maximum speed) of handling 250,000 instructions per second (estimated by utilizing an average of 2 bytes per instruction with 2 cycles per byte at a clock rate of 1 MHz). With the addition of suitable input-output interfaces and peripherals and of memory [both Read-Only-Memory (ROM) and Random-Access-memory (RAM)], a microcomputer is constructed. By the use of appropriate Digital to Analog Converters (DAC) and Analog to Digital Converters (ADC), we can interface this microcomputer with analog instruments for operational control and data acquisition.

The primary goal of this work was to design and implement a relatively low cost control package to be utilized with a standard laboratory potentiostat. The bulk of this paper is concerned with the implementation of this concept for the operational control of and data acquisition from a potentiostat used in electrochemical measurements. However, Section VII will be concerned with some planned uses of the equipment.

2. The Potentiostat

A potentiostat is an electronic device used by an electrochemist to control the potential of a cell in an accurate predetermined manner. The primary quantity that the electrochemist measures is the current as a function of the potentiostatically controlled cell potential and the rate of change of that potential. The cell potential, of course, varies with the input voltage which may be either AC or DC. The waveshape and frequency response are governed by the characteristic impedances of the overall system.

It is useful at this point to review the characteristics of a potentiostat [2] for the purpose of defining the factors that must be controlled by the microcomputer. The instrument can be modeled (fig. 1) as two amplifiers, one an error amplifier and the other a power amplifier. The electrochemical cell with which the potentiostat is used can be represented by the impedances Z_s and Z_c (for simplicity we assume that the cell contains only resistive components). Z_s is the lumped series impedance counter electrode, cell electrolyte, and the current leads, while Z_c is the impedance between the reference electrode and the working electrode (sample) across which the potential is developed that is to be controlled by the potentiostat. The operation of the system can be outlined as the comparison of the potential across Z_c with some input potential, amplification of the resulting difference signal and application of this amplified potential to the cell. The resulting current changes the potential across Z_c and the process continues until the difference signal is essentially zero.

* Partially supported by DoE.

** Present Address: Singer Corp., Kearfott Division, 1225 McBride Ave., Little Falls, N.J. 07424.

Note: Specific identification of companies in this paper cannot be construed as recommendations or endorsements by the National Bureau of Standards.

[1] Figures in brackets indicate the literature references at the end of the paper.

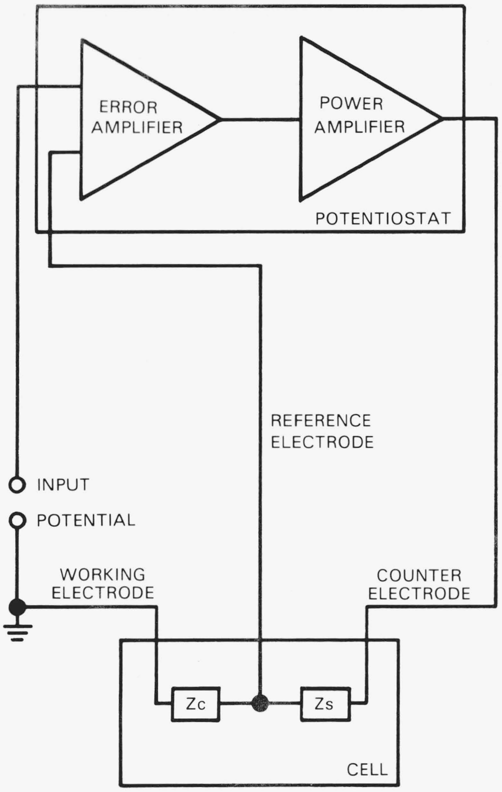


FIGURE 1. *Block diagram of potentiostat and cell.*

3. Basic Experimental Set-up and Design Criteria

A very common experimental arrangement consists of a motorized potentiometer used to generate an input potential to the potentiostat which varies linearly with time (ramp or triangular waveshape). The output of the instrument is the cell current which is plotted against cell potential on an X-Y recorder. Going beyond this experimental minimum, we find varying arrangements for input and output. The source potential is generated by oscillators, stepping-motor-driven sources, pulse generators, etc. These circuits allow the use of DC, ramps, stepped sweeps, waves, (sine, square and triangle), and pulse inputs. Compensation for changes in potentials due to IR drop across the cell's internal impedance is often added. Output data is often acquired by 2 pen recorders, logarithmic recorders, digital voltmeters, etc.

From this, we see that ideally, a digital control system for potentiostatic measurements should be capable of generating DC and AC potentials whose waveform must be easily controlled. Data acquisition should be high speed and the data should be of high resolution. The output of the acquired data must be flexible in format. To be useful, the instrument must be able to make at least simple computations using the acquired data so that results will be in the most desirable form.

Additional operational requirements can be imposed. The equipment should operate unattended (except for start-up), and must be able to terminate the experiment according to criteria established by the operator. If the device cannot perform the required data analysis by itself, it must have provision for communication with a larger machine. Finally, operation and programming of the instrument must be relatively simple and fast.

All of these functions can be performed by a system consisting of appropriate DAC's and ADC's combined with a digital computer. A minicomputer dedicated to this purpose would be under-utilized and was therefore not considered. The use of a communication line to a central computer with dedicated hardware at the instrument was considered but not used since cost estimates indicated it would be essentially no cheaper than the use of a microcomputer. A desk calculator could be used but is more expensive.

4. The Microcomputer [3]

The system designer must first evaluate the trade-offs necessary to determine which microprocessor matches the application best. Many factors enter into this decision and they must be carefully evaluated before proceeding. We have stressed the following:

1. Word size—The word length of a microprocessor at the present time can be 4, 8, 12, 16 bits, etc. In general, the capability of the unit increases with increasing word length, but so does the overall system cost.
2. Speed—The speed of a μP is not only a function of clock frequency, but also depends upon other factors such as the instruction cycle and the instruction set.
3. Instruction set—In general, the larger the instruction set, the more capable the resulting microcomputer. We must note, however, that in some cases, one microprocessor might require special instructions (e.g., input-output instructions) that would not be needed by other microprocessors.
4. Chip family—This factor is of great importance to the scientist. A well integrated and compatible set of circuits that reduce the time and effort necessary for design and construction of a system can more than offset other factors.
5. Available software—If we examine the overall costs of a system, programming is by far the most expensive element. If a body of software already exists, then the overall expense of the system is effectively reduced.
6. Available hardware—Obviously, the less time and effort taken from the scientist for design and construction, the better. New devices and hardware are contin-

ually appearing and it is important to be aware of these.

Details necessary for evaluation of many of these factors are available from the μP manufacturer's data sheets. The relative importance of each factor is dependent upon the particular project, the potentiostat used and the individual designer.

Our intended implementation of microprocessor control of a potentiostat relied upon the use of computer controlled digital to analog converters for generation of the input signals to the potentiostat. Measurement of the output of the potentiostat was to be performed by analog to digital conversion. The entire system was to be capable of interactive control with the operator determining the initial parameters and the extent of computation necessary. For purposes of speed and program simplicity we felt that direct access of the computer to the peripherals without the need for special instructions, was a necessity. The resulting design followed the general scheme:

1. The computer (based upon initial parameters interactively determined with the operator) outputs a signal to the potentiostat.
2. After a period of time the computer reads the output of the potentiostat and then (again based on initial parameters) either stores the data and goes on to the next point, or compares the data with previous data and goes on to the next point only when previously established criteria are met.
3. The process continues until either the computer determines that the experiment is complete or the operator terminates it prematurely.
4. Depending on the time interval per point, computations can be made either during the wait period or after all data is acquired.

At the time this instrument was initially designed, we determined that the 6800 Microprocessor was the best fit to our requirements. It is an 8 bit μP, with a clock frequency of up to 1 MHz and a minimum instruction time of 2 cycles. It has a reasonably extensive instruction set with several addressing modes giving great flexibility. Peripherals are treated as memory locations, thus requiring no special instructions. (This is probably the most important reason for selecting the 6800. This feature, desirable for the potentiostat, was not, in general, found in other microprocessors at similar cost.) The accessory chips are appropriate allowing relatively simple and easy access to peripherals with a minimum of design. Although the available software is not as extensive as that available for the 8080, it is adequate for our purposes. There is a considerable variety of peripheral hardware available for microcomputer systems based on the 6800 μP, enabling the designer to add on functions to his system with minimal problems of design and construction.

The 6800 based microcomputer we have constructed is based on the manufacturer's evaluation board [5] which has been somewhat modified. This board provides the microprocessor with its support circuits (clock, gating circuits, etc.). It also provides interface adaptors for 3 peripherals [2 parallel using Peripheral Interface Adaptors (PIA) and one serial using an Asynchronous Communications Interface Adaptor (ACIA)]. One of the parallel ports is dedicated to communication with a teletype or low speed RS-232-C device. (A detailed discussion of all the circuit boards will follow later on in this paper). In our system, communication with a central computer, if desired, is provided through a terminal with a modem and an acoustic coupler. However, if we wish, data can be recorded on cassette tape at the terminal (at 30 or 120 characters per second) for later processing by the central computer.

The rest of the microcomputer consists of 5 circuit boards. There are two 8192 byte (one byte equals an 8 bit word) RAM boards, one converter (DAC's, ADC) and timer board, one board for signal distribution, and one variable speed serial interface board. The two memory boards feature data buffers and fully decoded address circuitry and include switching operations to place their base address at any 4K boundary in the address field. The variable speed, serial interface provides interfacing for RS-232-C communication levels at 8 speeds ranging from 10 characters per second to 960 characters per second.

The converter and timer board contains three PIA's with interfacing, one of which is dedicated to a programmable timer. This has a range of microseconds to 40 minutes per timing point and generates an interrupt signal at the conclusion of the period. A second PIA is dedicated to two 10 bit digital to analog converters which have an output range of -5.12 to $+5.11$ volts. The third PIA interfaces with a 10 bit tracking ADC.

The signal distribution board provides connection points for input and output to the potentiostat or other device. Of the two DAC outputs one is normally connected to the input potential point on the potentiostat. The other output is connected to a voltage controlled oscillator (3×10^{-5} to 3×10^{6} Hz). There are three ADC connectors. One provides an output of the analog voltage generated by a DAC which output is subsequently compared with the input (identified in fig. 12 as ECHO), while the other two connectors are inputs which are connected thru a switch to a precision voltage clipper (discussed below). The output of the clipper provides the input to the ADC. These two inputs have a common return. (One of these inputs carries the full signal, while the other input carries half of this potential.)

Communication between operator and equipment occurs via a teletype or RS-232-C interface to a terminal. There is a monitor program in the ROM which obviates the need for panel switch programming. In fact, the only switches needed (other than power on) are a reset and communication speed set switch. Power requirements are +5V @ 6A, ±15 V @ 50 mA, and ±12 V @ 100 mA. A small cooling fan is necessary if the microcomputer is enclosed in a cabinet. Figure 2 shows the overall block diagram of the system.

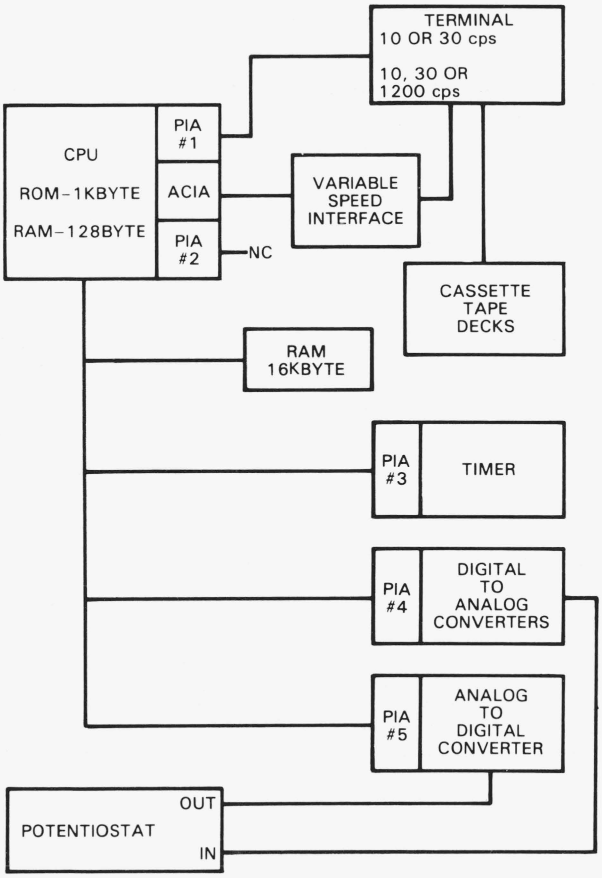


FIGURE 2. *Block diagram of complete system.*

5. Board-by-Board Details

The CPU of the microcomputer is a somewhat modified version of the manufacturer's evaluation board #1. The original board was designed to be a stand-alone unit with limited memory. Addressing is redundant and expansion of the memory is possible only by removing the redundancy. This is easily accomplished by removing all memory from the evaluation board except for the monitor ROM (address E000 hex) and its associated scratch-pad RAM (address A000 hex). An additional redundancy exists for the peripheral addresses. Since it is desirable for the peripheral addressed to be separated from main memory, we added an address line to the scratch pad enabling several blocks of addresses in the memory region that is identified by Axxx hex. An additional line was run from the PIA used for a low speed terminal. This line allows switch selection of 10 or 30 character per second operation. A schematic of the resulting board, including only operational circuits, is shown in figure 3. A map of available memory is shown in figure 4.

The monitor ROM (MIKBUG*), which is included in the evaluation kit, allows for loading from and generating of tapes, displaying and changing the contents of memory locations, displaying the contents of the primary registers and entering a user written program. Since most of the commercially available software is written using MIKBUG subroutines for control of input and output, we have retained it. Although limited in scope, it is adequate for our purposes.

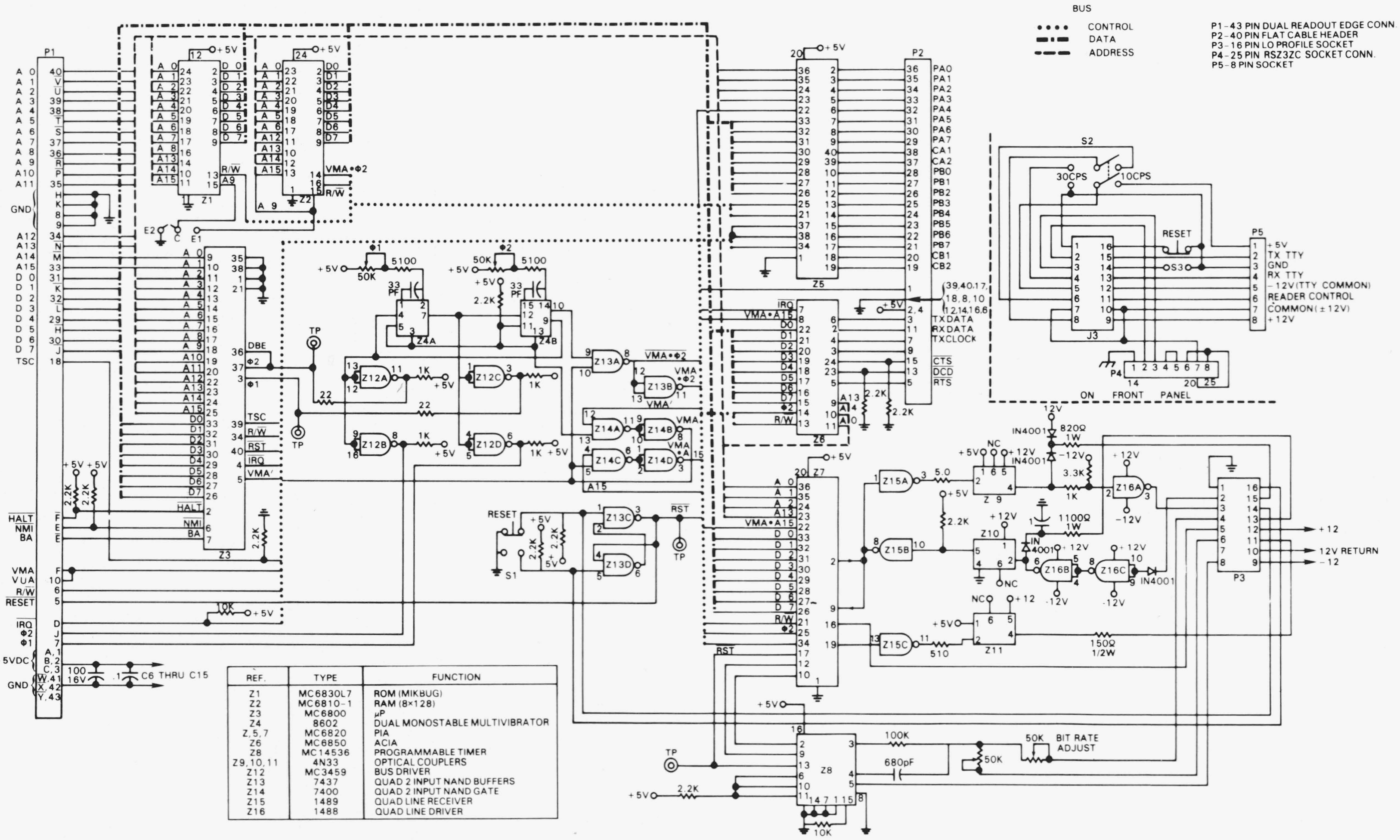
The 2 phase multivibrator clock oscillates at a frequency of 1 MHz for operation of the microcomputer. Three buses (data, address and control) are brought out through a dual 43 pin edge connector. As previously mentioned, there are 3 on-board peripheral adaptors. One of these adaptors (a PIA) is reserved to interface with a planned addition of a floppy disc unit.

The 8K memory boards (fig. 5) are based on 1024 × 1 bit static memory Integrated Circuits (IC). These memories have access times of 500 nanosecond (nsec) or better. Address and data lines are buffered and base address selection is made by means of a hexadecimal coded switch. Connection to the bus is via a dual 43 pin edge connector. Only one board contains the Power-on Reset [6] circuit shown in figure 5.

Figure 6 is a schematic of the variable speed interface together with the low speed terminal connections. RS-232-C interface connections are shown for mating with an Automatic Send Receive (ASR) terminal that has the capability for 120 characters per second operating speed when recording or playing back with its built-in cassette tape drives. If desired, data rates other than those shown can be obtained through the use of the MC 14411 bit rate generator [5, 8]. These data rates can be achieved by wiring pins 22 and 23 (Rate Select B and Rate Select A) according to Table I or by connecting them to a peripheral interface and programming them. There are also 8 other speeds directly available from the chip at any rate select configuration. (Note that the output frequency for each of the positions is found by multiplying the characters/sec by 160.)

The timer-converter board schematics are shown in figure 7 through figure 11. The interval timer [7] section (fig. 8) is software programmable and runs off the system clock (1 MHz). Thus, its accuracy is dependent upon the stability and frequency of that signal. Since we are not using a crystal

* Registered Trademark Motorola (Firmware ROM program)



REF.	TYPE	FUNCTION
Z1	MC6830L7	ROM (MIKBUG)
Z2	MC6810-1	RAM (8×128)
Z3	MC6800	μP
Z4	8602	DUAL MONOSTABLE MULTIVIBRATOR
Z,5,7	MC6820	PIA
Z6	MC6850	ACIA
Z8	MC14536	PROGRAMMABLE TIMER
Z9,10,11	4N33	OPTICAL COUPLERS
Z12	MC3459	BUS DRIVER
Z13	7437	QUAD 2 INPUT NAND BUFFERS
Z14	7400	QUAD 2 INPUT NAND GATE
Z15	1489	QUAD LINE RECEIVER
Z16	1488	QUAD LINE DRIVER

FIGURE 3. *Schematic of CPU Board as modified.* Also shown are components mounted on front panel.

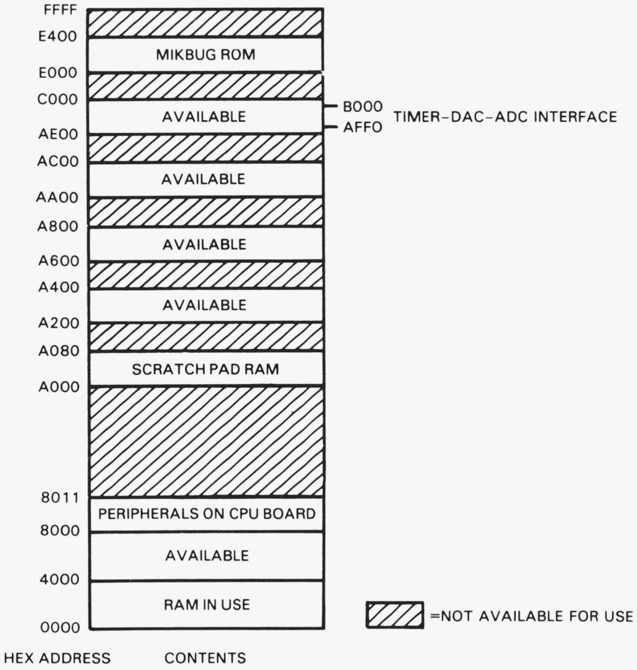


FIGURE 4. *Memory map of system.*

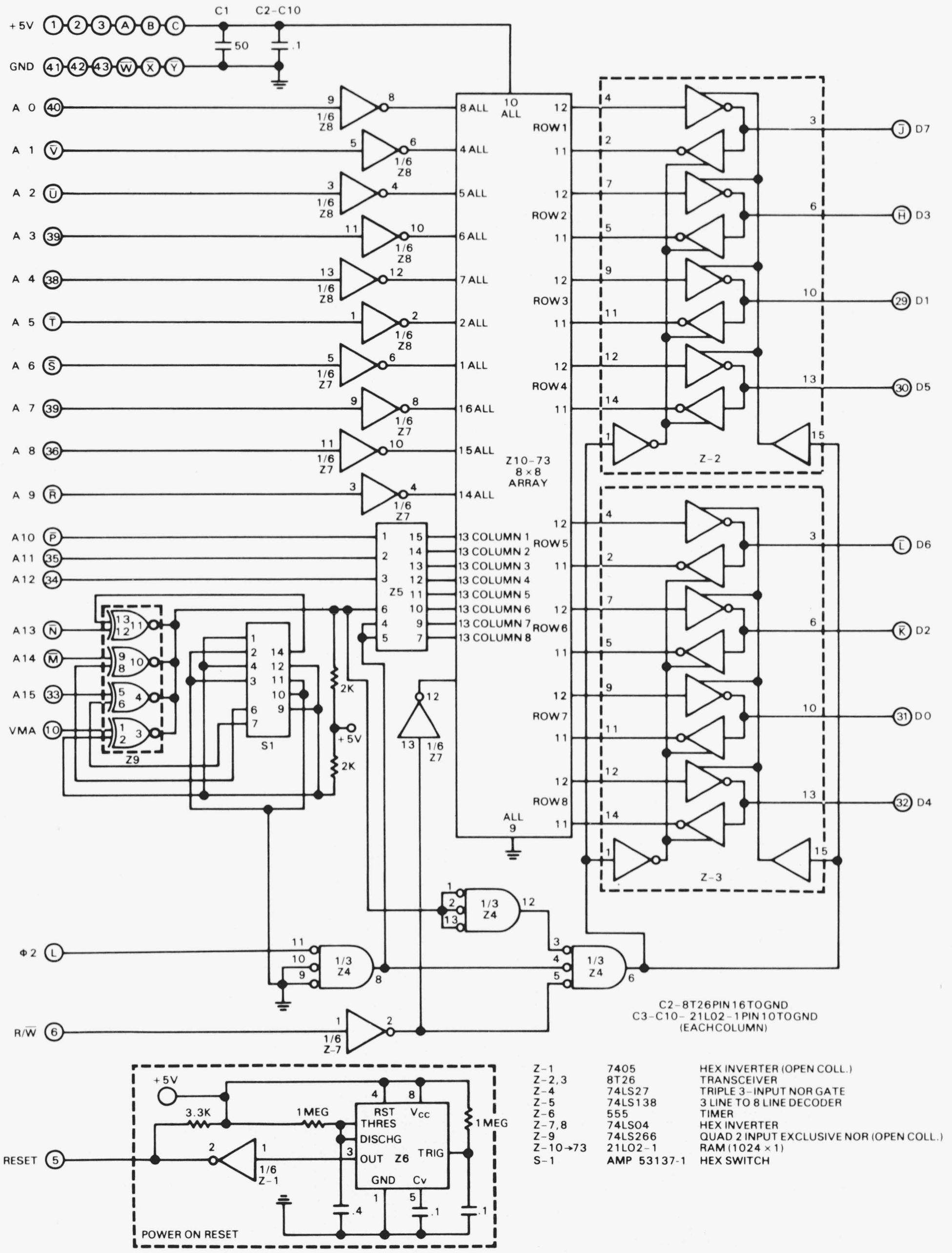
controlled oscillator the stated set times must be considered to be nominal values. At the present time, they are sufficiently precise for our application. Installation of a two phase crystal controlled clock is not difficult and can be done when necessary.

The setting of the time interval is done in two steps. First a code, defining which output from the 7490-74452 decade divider chain is to be fed to the presettable counter (74455), is loaded, via the PIA A side, into the 9312 multiplexer. Then, the 74455 is preset with the number of counts desired (0–256) from the B side of the PIA, and counting (down count) is started. When the desired number of timing pulses has been counted, the 74455 puts out an interrupt pulse to the CPU. The minimum interval time for the timer is 1 Microsecond (μs). The maximum interval is 42.6 minutes. However, the maximum repetition rate is dependent on the interrupt handling routine in the microprocessor. We have not directly determined this elapsed time for the shortest program but estimate it to be on the order of 30 μs. The program we are using takes on the order of 250 μs. As will be shown, this places a lower limit on the effective pulse time period for the input to the potentiostat.

There are two identical DAC's (fig. 9) on this board. [9] Under program control, each one can be independently loaded from the PIA utilizing the 7475 latches. As presently formatted, the input to the DAC remains the same until changed by the CPU. Thus, the output of the DAC, which is fed to the potentiostat, remains at a constant level until the next value is placed on the latches. A simple addition of circuitry (shown as fig. 10) will remove this feature and allow the input pulse width to the potentiostat to closely follow the timer period.

As a check on performance, a program for generating a triangular wave (in steps of 10 millivolts (mV)) was written. An effort was made to minimize program steps. When run, this program generates a DAC output triangular wave of 25 Hz. The time per point works out to be 20 μs. From this we can estimate that a waveform generated from a lookup table would take about 50 μs per step (based on a minimum number of instructions). It is obvious that were we to generate individual points of a waveform and output them individually, as generated by the program, we would have much longer times per point. Pulses of variable duty cycles would be limited in repetition rate by the interrupt handler (see timer discussion) to about 30 μs, although "on" time could be adjusted by the technique discussed before (fig. 10 and text of previous paragraph).

Repetitive waveforms of normal complexity (sine, square, triangle, ramp) for input to the potentiostat can be generated under program control in an alternative manner. The output of one of the DAC's can act as input for a Voltage Controlled Oscillator (VCO) (the VCO we use has a frequency deviation range of 30 μHz to 3 MHz) allowing program selection of output frequency. The inverted CB_2 output of the timer PIA is used as a trigger signal for the VCO. This oscillator puts out one cycle of the selected frequency when triggered. Since



C2-8T26PIN16TOGND
C3-C10- 21L02-1PIN10TOGND
(EACHCOLUMN)

Z-1	7405	HEX INVERTER (OPEN COLL.)
Z-2,3	8T26	TRANSCEIVER
Z-4	74LS27	TRIPLE 3-INPUT NOR GATE
Z-5	74LS138	3 LINE TO 8 LINE DECODER
Z-6	555	TIMER
Z-7,8	74LS04	HEX INVERTER
Z-9	74LS266	QUAD 2 INPUT EXCLUSIVE NOR (OPEN COLL.)
Z-10→73	21L02-1	RAM (1024 × 1)
S-1	AMP 53137-1	HEX SWITCH

FIGURE 5. *Schematic of 8 kilobyte RAM Board.*

Also shown is Power-on Reset circuit which is included on one RAM Board only.

peak-to-peak voltage, offset voltage, and starting phase are all controllable, we have considerable flexibility for the generation of waveforms.

One final comment is in order. The DAC section of the board contains logic for limited multiplexing (although not fully implemented) of the input and output signals. With the installation of some reed relays we can choose between two distinct analog sources and simultaneously between four outputs. Some minor circuitry changes would allow all of these lines to be dedicated to some other arrangement of signals or to extend the multiplexing range. Addition of this multiplexing would allow measurement of both the potentiostat output voltage and current (essentially a simultaneous measurement). Also outputs to other peripherals (such as X-Y recorders) would be available.

The ADC section (fig. 11) of the board reflects some major

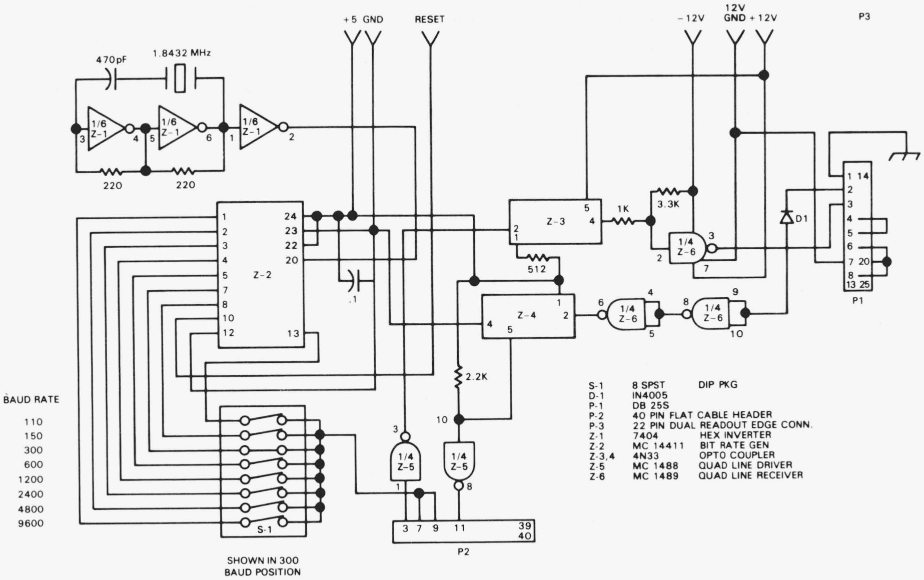


FIGURE 6. *Schematic of Variable speed serial interface.*

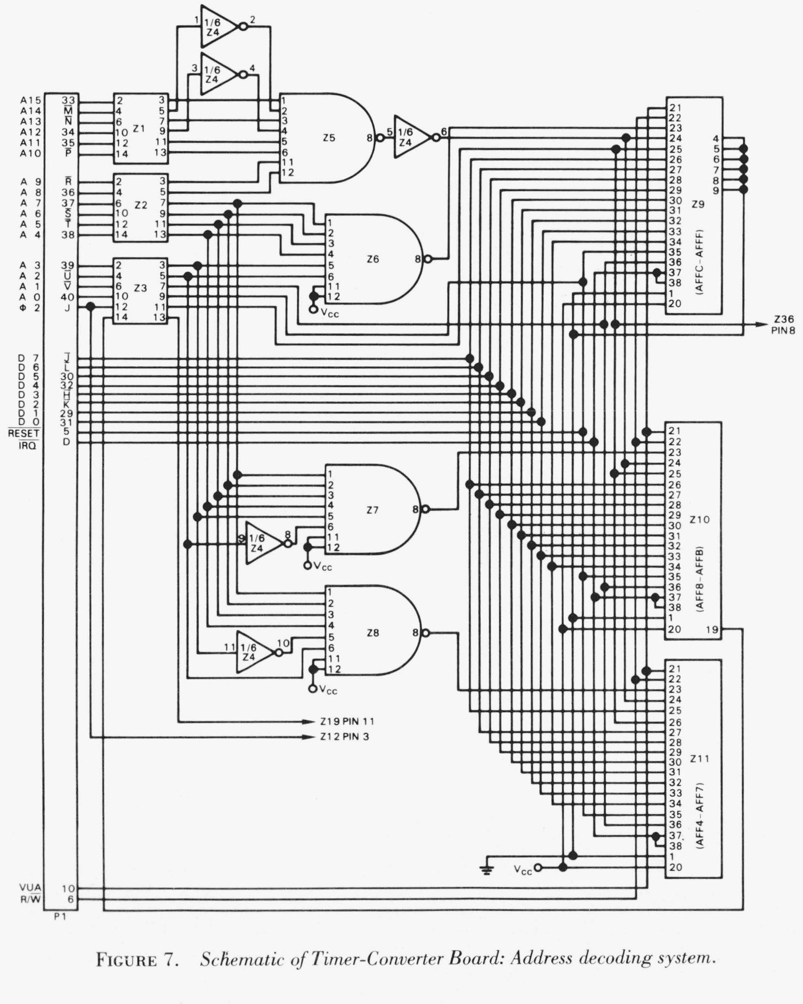


FIGURE 7. *Schematic of Timer-Converter Board: Address decoding system.*

TABLE I. *Data Clock Rate Selection*

Rate Multiplier	Rate Select A Connection	Rate Select B Connection
X1	0	0
X8	1	0
X16 ----------------	0 -------------------	1 (present connections)
X64	1	1

where: 0 = GND
1 = +5*V*

TIMER-CONVERTERCOMPONENTS		
REF.	TYPE	FUNCTION
Z1,2,3	8T97	BUFFER
Z4,12,19,45	7404	INVERTER
Z5,6,7,8	7430	8 INPUT NAND GATE
Z9,10,11	MC6820	PIA
Z13	9312	MULTIPLEXER
Z14	MC74455	BINARY UP/DOWN COUNTER
Z15	7490	DECADE COUNTER
Z16,17,18	MC74452	DUAL DECADE COUNTER
Z20,21,22,23	8T26	BUS TRANSCEIVER
Z24,25,26,27,28,29,30	7475	4 BIT LATCH
Z31,32,35	DAC-100	10 BIT DAC
Z33,34,40	741	OP AMP
Z36,37,38	MC8284	BINARY UP/DOWN COUNTER
Z39	7474	FLIP-FLOP
Z41	CMP-01	VOLTAGE COMPARATOR
Z42,43,44	7408	2 INPUT AND GATE
Z46	7432	EX OR GATE
D1-12	IN914	DIODE
DZ1-3	IN754	ZENER DIODE

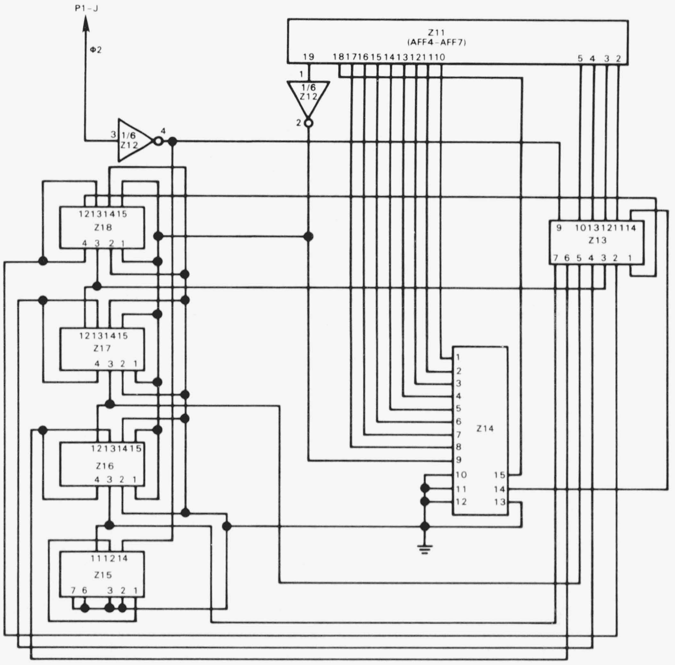


FIGURE 8. *Timer-Converter Board: Timer section.*

compromises that were made after careful trade-off considerations of system speed requirements and costs. Ideally, the analog to digital converter would have a resolution of one millivolt or better and would acquire the data in one μsec or less. Such a device is not feasible at this time. Examination of the experimental needs allows relaxation of these requirements particularly in the area of high-speed operation. The experiment as planned calls for the input to the potentiostat to be stepped through a cycle in which the input potential is changed by a fixed value, and then sustained for a period of time. At this time the current is measured and after this reading the cycle is continued. As indicated in the timer discussion the fastest rate at which the input can be changed is 30 μs (or 33.33 khz). This establishes an upper limit on the operating speed. Resolution requirements can be minimized by amplification of the input signal (preferably under program control.) In our case, automatic amplification is not performed, our choice being a manual change in output amplification on the potentiostat. In addition to the factors already noted, cost and circuit complexity are vital considerations.

We should also note here that the output current from the potentiostat (our analog signal source) will often be slowly falling with time due to the charging of the reactive components of the cell. It is desirable for the ADC to follow this fall closely. (When this exponential fall-off has reached a quasi-constant level, a "steady state" condition has been reached in the cell, and this is a region of interest.)

After weighing all these points, we decided to use a 10 bit tracking ADC (also known as counter-ramp ADC) [10, 11]. This type of ADC is normally thought of as a "slow" converter, since the time to acquire the signal is $(2^n - 1)$ times the step rate of the counter for *full scale* changes (n is the number of bits in the DAC component of the converter).

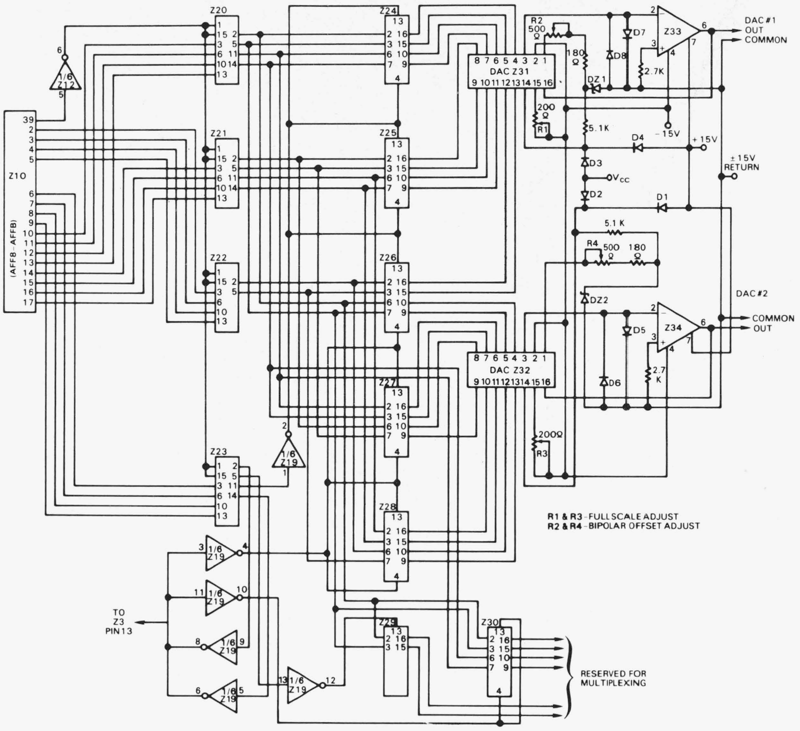


FIGURE 9. *Timer-Converter Board: DAC section.*

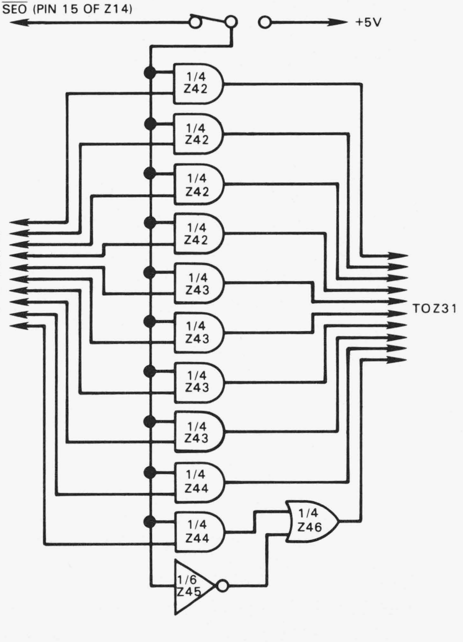


FIGURE 10. *Timer-Converter Board: modification to DAC section proposed for Interrupt-Triggered return to zero output.*

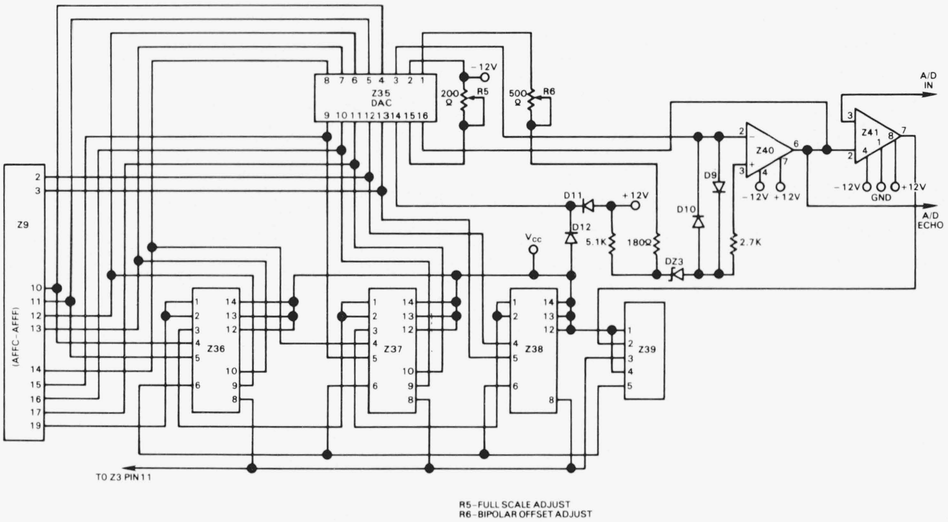


FIGURE 11. *Timer-Converter Board: ADC section.*

In our case, this is about three milliseconds. However, this number is valid only for full scale excursions. For less than full scale the time is reduced proportional to the change in the number of counts needed. Indeed for a change in input potential of 100 mV (10 counts) the time is 30 μs. This is estimated for a step change. For a slow change the speed would be even better. The overall limitation on ADC conversion (for small analog change) rate is the conversion speed of the component DAC.

The tracking ADC converter follows the input potential continuously, and the digital output is always available. It is a relatively simple circuit [10] consisting of an up/down counter, a DAC, an amplifier, a comparator and a gate. We are using a 10 bit DAC that has a conversion rate of ~40 kHz. Our clock source for the counter is the CPU clock of 1 MHz. The ADC could be run at a maximum frequency of 2.5 MHz if desired. This would allow an ADC conversion rate of 800 Hz for full scale excursions and proportionately reduced times for small changes. Overranging is a problem with this converter. If the input potential exceeds the full scale potential, the counter will continue to count indefinitely and the digital data will be in error. To avoid this, a precision voltage limiter is employed on the input to the ADC [12]. This is physically located on the signal distribution board (fig. 12) in close proximity to the potentiostat. The full scale input to the ADC ranges from −5 to +5 volts.

6. Programming [13, 14]

To no one's surprise, initial programming is a major fraction of the time needed to develop an operating system. Fortunately, the commercial availability of a higher level language (BASIC) has reduced the size of this effort considerably. There are compromises needed, however, which must be carefully weighed.

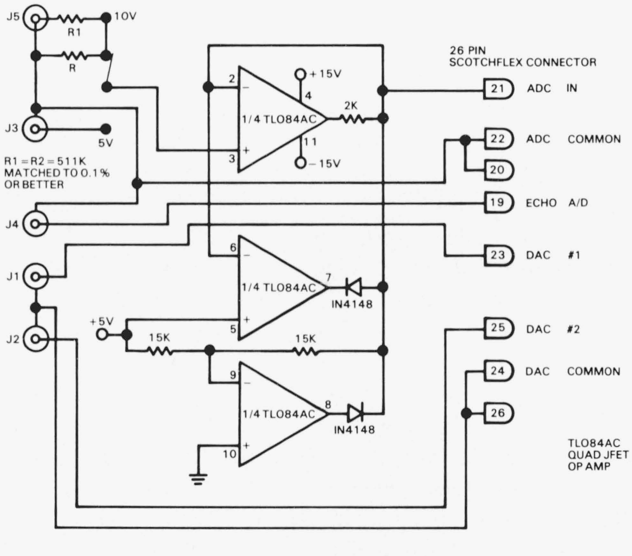


FIGURE 12. *Schematic of Signal Distribution Board.*

Assembly language programming yields a more concise and faster running code than the use of a higher level language, especially when that language is implemented as an interpreter. Since relatively high speed data acquisition is an integral part of our experimental concept, alternatives to BASIC must be considered. Two approaches are immediately obvious. First, we can implement in hardware a Direct Memory Access (DMA) system. This would have the advantage of providing the fastest possible storage of data. However, flexibility would be reduced. The second approach (the one we adopted) would involve the calling of a machine language program as a subroutine while the overall program would be in BASIC. Under this concept, we would maintain the ease and flexibility of programming in a higher level language while maintaining reasonable data acquisition speeds. This requires that the BASIC program (written or purchased) include a command to call a machine language program, a means to return from that program and the ability to handle the data in the form generated by the data acquisition program. There are several commercial programs available (one of which we have acquired) [15] that satisfy these requirements.

For generation of machine language programs the use of an assembler is suggested. This can either be "resident" in the microcomputer or can be available via time-sharing on a large computer (cross-assembler). An example of a cross assembled program is shown in appendix A. This particular program is of interest since it is the data acquisition program we currently are using. (Some comments are in order). First, this program utilizes only one of the two DAC's. Second, initial clearing of the data storage area is accomplished by the BASIC control program (automatically on RUN). Third, this is probably *not* the fastest possible program (as indicated earlier it uses about 250 μs per point). Fourth, proper organization of the BASIC program will enable this program to be utilized for signal averaging. For this purpose, the data storage allotted per point, limits the maximum number of sweeps to 63. Addition of one more byte of data storage per point would allow more data accumulation by allotting memory space for the addition of 256 more sweeps.

The required memory for the BASIC interpreter and the machine language program is approximately 8K byte. The BASIC control program and its required buffer memory takes an additional 4K byte leaving 4K byte for data storage. Since 4 bytes are required per point, only 1000 points can be stored without installing additional RAM. In the future, we expect to place these programs into ROM thereby freeing the 16K bytes of RAM for additional data storage. At present, however, they are recorded on cassette tape and are loaded into RAM when needed.

7. System Use

We have described the implementation of a microprocessor control and fast data acquisition system for use with a standard laboratory potentiostat. Appendix B shows a result from one of the experiments we are conducting using this instrument. Using this as an example we can illustrate several features. First, we can accumulate digital data at selected rates. If we sweep at fast rates, the data can be plotted in a manner yielding the equivalent of a potentiodynamic scan. If we use long delay times per point—long enough for steady state to be reached—we can generate semilogarithmic plots (commonly known as Tafel plots [16]) which yield information about the kinetics of the electrode system. (Alternatively the machine language program can be changed to one that will check the raw data for steady state conditions after which the data is plotted).

Second, we can perform most of the necessary computations without an operator. This means that the instrument will not only give us the data as acquired, but will perform the necessary mathematical steps to generate a more usable form of information (e.g., calculate the log of the current and then plot those values). The BASIC program in the μP, although slow compared to a big machine, is capable of most of the operations (except Matrix functions in particular) that BASIC in a larger computer could perform. Computational speed is quite adequate, and for those occasions when it is not, communication with a larger machine is provided for.

Third, once programmed and started, the instrument can run unattended. The length of time a run takes does not have to be fixed. The microcomputer can run an experiment for as long as it needs to run. It can be programmed, if necessary, to take emergency measures when conditions exceed limits.

This instrument has, as indicated before, Alternating Current (AC) capabilities. For discussion purposes, we will separate pulse techniques from other methods. In general, in pulse techniques we observe the effects of an impulse input on the cell by observing the decay of its effect. Pulses can be derived either from the timer (minimum width 1 μs) or from the voltage controlled oscillator (minimum width 200 ns). Other waveforms can be provided by the DAC or the VCO (which depend upon the frequency desired—see DAC discussion). If needed, one DAC can provide a DC level which can be summed with the VCO output (controlled by DAC #2) and directed to the potentiostat input.

Direct measurement of AC potentials is possible using the existing ADC for a limited frequency range. To obtain all the information in an AC signal, the sampling rate must be at least twice the highest signal frequency [17]. Our sampling rate for full scale excursions is 300 Hz and thus our maximum frequency would be 150 Hz. In line with our previous discussion, this established maximum would increase as the AC peak to peak voltage decreases. As a practical consideration, however, the increase in bandwidth is not wide enough in general, and another technique should be considered. This can either take the form of a different type of ADC with wider bandwidth or some form of AC to DC converter whose output can be measured by the microcomputer ADC. At present, the response is adequate for our purpose but we are actively pursuing the resolution of this problem for the future.

8. Cost of Implementation

The system described here was constructed for a parts cost of slightly less than $1000 (exclusive of terminal). Table II shows the breakdown of costs per board. Depending upon desired parameters, systems similar to this can be constructed that range in parts cost from about $800 on up. The lowest system price is based on the use of minimum memory hardware (no BASIC). Decisions as to memory size, other peripherals, and/or purchase of assembled boards from commercial sources will have a significant effect on overall expense. There is, at present, no commercial source for a tracking type analog to digital converter (similar to the one used here) that mates with this bus structure. Successive approximation type converters are available (usually with multiplexed inputs) but they are more expensive, are relatively limited in throughput, and do not contain the necessary timer circuit.

TABLE II. *Circuit Board Parts Cost and Commercial Availability*

PC Board	Prototype Parts Cost	Similar Board Available Commercially*	Commercial Pricing
Microcomputer	$200	Yes	Same or higher
8K Byte RAM	180	Yes	Higher
8K Byte RAM	180	Yes	Higher
Variable Speed Serial Interface	25	Yes	Higher
Converter & Timer Board (including Signal Distribution)	250	No	—
Power Supplies & Motherboard	150	Yes	Same or higher
Total	$985		

* Partial listing of possible sources for directly compatible boards: Motorola, Creative Micro Systems, Electronic Product Associates, and many others.

9. Conclusions

In conclusion, an instrument has been constructed that offers improvement in measurement techniques over manual methods. This device incorporates one of the recent advances in instrumentation (the microprocessor) which allows rapid, concise and consistent measurements not before realized by manual methods.

The authors wish to thank H. P. R. Frederikse, A. D. Franklin, and L. H. Bennett for useful discussions. A very special thanks to U. Bertocci without whose advice and encouragement this work would not have been completed.

10. REFERENCES

[1] Torrero, E. A., ed. Microprocessors-New Directions for Designers. New Jersey: Hayden Book Co., Inc.; 1975. 135 p.

[2] Fraunhofer, J. A., Banks, C. H. Potentiostat and its Applications. London: Butterworth & Co., 1972. 254 p. ISBN: 0-408-70239-7.

[3] Peatman, John B. Microcomputer-Based Design. New York: McGraw-Hill Book Co., 1977. 540 p. ISBN: 0-07-049138-0.

[4] Motorola Semiconductor Products, Inc. M6800 Microcomputer System Data. Phoenix, Arizona; 1976. 164 p.

[5] Motorola Semiconductor Products, Inc. Type No. MEK6800 D1, M6800 Microcomputer Design Kit.

[6] Motorola Semiconductor Products, Inc. M6800 Microprocessor Application Manual. Phoenix, Arizona; 1975. p. 4–42.

[7] Ibid; p. 4–46.

[8] Motorola Semiconductor Products, Inc. MC14411 Data Sheet. Phoenix, Arizona; 1975. 4 p.

[9] Precision Monolithics, Inc. Technical Specification AIM DAC100, 8 and 10 Bit I.C. Digital-to-Analog Converter Series. Santa Clara, CA; 1973. 8 p.

[10] Precision Monolithics, Inc. Application Note AN-6, A Low Cost, High Performance Tracking A/D Converter. Santa Clara, CA; 1973. 8 p.

[11] Hoeschele, D. F., Jr. Analog-to-Digital/Digital-to-Analog Conversion Techniques. New York: J. Wiley and Sons, Inc.; 1968. p. 392. ISBN: 471-40575-2.

[12] Graeme, J. B. Designing with Operational Amplifiers-Application Alternatives. New York: McGraw-Hill Book Co., 1977. p. 44. ISBN: 0-07-023891-x.

[13] Motorola Semiconductor Products, Inc. M6800 Programming Reference Manual. Phoenix, Arizona; 1976. 311 p.

[14] Wester, J. B., Simpson, W. D. Software Design for Microprocessors. Dallas, TX: Texas Instrument Learning Center; 1976. 378 p.

[15] 8K BASIC Version 2.0 [Computer Program]. Uiterwyk, R. H. San Antonio, TX: Southwest Technical Products Corp.; 1976.

[16] Bauer, H. H. Electrodics-Modern Ideas Concerning Electrode Reactions. New York: J. Wiley and Sons, Inc.; 1972. p. 15. ISBN: 0-470-05920-6.

[17] Magrab, E. B., Blomquist, D. S. The Measurement of Time-Varying Phenomena-Fundamentals and Applications. New York: Wiley-Interscience; 1971. p. 233. ISBN: 0-471-56343-9.

11. Appendix A. Cross Assembled Data Acquisition Program

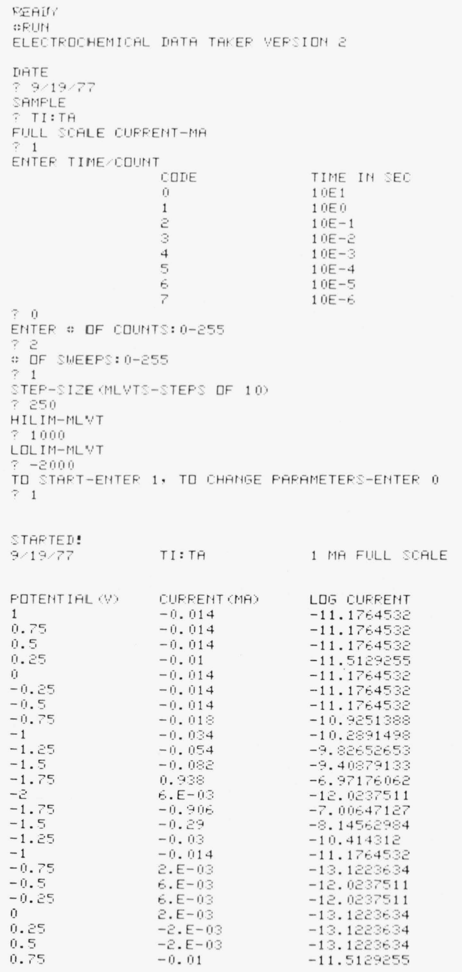
```
LINE    ADDR  B1 B2 B3 E    DATA ACQUISITION

   1                          NAM DATA ACQUISITION

   3                          ◆THIS PROGRAM OUTPUTS A POTENTIAL
   4                          ◆BETWEEN -5 AND +5 VOLTS FOR A
   5                          ◆FIXED TIME PERIOD, AT THE
   6                          ◆END OF WHICH, A POTENTIAL IS READ.
   7                          ◆THE UPPER AND LOWER LIMITS CAN BE
   8                          ◆SET EXTERNALLY, AS CAN BE THE
   9                          ◆STEP SIZE AND CLOCK PERIOD.

  11                          ORG $1EBF
  12    1EBF  1E AF          CLKCT EQU $1EAF HI=TIME/COUNT:LO=#COUNTS
  13    1EBF  1E B3          STEPL EQU $1EB3
  14    1EBF  1E B2          STEPH EQU $1EB2
  15    1EBF  1E B4          HILIM EQU $1EB4
  16    1EBF  1E B6          LOLIM EQU $1EB6
  17    1EBF  1E B8          DIREG EQU $1EB8 0=DOWN:1=UP
  18    1EBF  1E B9          TREG EQU $1EB9 POINTER TO TOP OF DATA STACK
  19    1EBF  1E BB          RREG EQU $1EBB POINTER TO BOTTOM OF STACK
  20    1EBF  AF F4          TIMER EQU $AFF4
  21    1EBF  AF F8          DAC EQU $AFF8
  22    1EBF  AF FC          ADC EQU $AFFC
  23    1EBF  A0 00          IOINT EQU $A000 INTERRUPT VECTOR
  24    1EBF  A0 14          TEMPR EQU $A014
  25    1EBF  1E BD          TREGW EQU $1EBD
  26    1EBF  A0 16          INREG EQU $A016
  27    1EBF  A0 18          SREG EQU $A018
  28    1EBF  1E B1          SWEEP EQU $1EB1 SAVE #OF SWEEPS


  30    1EBF  36              PSH A  SAVE REGISTERS
  31    1EC0  37              PSH B
  32    1EC1  07              TPA
  33    1EC2  36              PSH A
  34    1EC3  FF A0 16        STX INREG
  35    1EC6  BF A0 18        STS SREG


  37    1EC9  7F 1E B8        CLR DIREG  SET DIRECTION, DOWN
  38    1ECC  FE 1E B9        LDX TREG
  39    1ECF  FF 1E BD        STX TREGW
  40    1ED2  BE 1E B4        LDS HILIM
  41    1ED5  AF 00           STS 00,X LOAD LIMIT INTO T REGISTER
  42    1ED7  CE AF F4       SETUP LDX #TIMER SET UP PERIPHERALS
  43    1EDA  4F              CLRA
  44    1EDB  A7 02           STAA 02,X
  45    1EDD  A7 03           STAA 03,X
  46    1EDF  A7 06           STAA 06,X
  47    1EE1  A7 07           STAA 07,X
  48    1EE3  43              COMA
  49    1EE4  A7 00          STAA 00,X
  50    1EE6  A7 01          STAA 01,X
  51    1EE8  A7 04          STAA 04,X
  52    1EEA  A7 05          STAA 05,X
  53    1EEC  86 04          LDAA #$04
  54    1EEE  A7 02          STAA 02,X
  55    1EF0  A7 0A          STAA 0AH,X
  56    1EF2  86 2D          LDAA #$2D
  57    1EF4  A7 03          STAA 03,X
  58    1EF6  86 3C          LDAA #$3C
  59    1EF8  A7 06          STAA 06,X
  60    1EFA  A7 0B          STAA 0BH,X
  61    1EFC  86 2C          LDAA #$2C
  62    1EFE  A7 07          STAA 07,X
  63    1F00  0E            DATAT CLI
  64    1F01  CE 1F 1C       LDX #SERV
  65    1F04  FF A0 00       STX IOINT
  66    1F07  FE 1E BD       LDX TREGW
  67    1F0A  AE 00          LDS 00,X
  68    1F0C  BF AF F8       STS DAC  OUTPUT-ANALOG VOLTAGE
  69    1F0F  FE 1E AF       LDX CLKCT
  70    1F12  FF AF F4       STX TIMER OUTPUT TIME PERIOD
  71    1F15  BE A0 18       LDS SREG
  72    1F18  3E             WAI WAIT FOR INTERRUPT
  73    1F19  01             NOP
  74    1F1A  20 0F          BRA CONT
  75    1F1C  86 08         SERV LDAA #$08
  76    1F1E  B7 AF F4       STAA TIMER  STOP TIMER
  77    1F21  B6 AF F5       LDAA TIMER+1  READ OUT TIMER
```

```
LINE    ADDR  B1 B2 B3 E    DATA ACQUISITION

  78    1F24  FE AF FC        LDX ADC  READ ADC
  79    1F27  FF A0 14        STX TEMPR
  80    1F2A  3B              RTI
  81    1F2B  FE 1E BD       CONT LDX TREGW
  82    1F2E  A6 03           LDAA 03,X  RECALL PREVIOUS VALUE
  83    1F30  E6 02           LDAB 02,X
  84    1F32  BB A0 15        ADDA TEMPR+1  ADD NEW VALUE
  85    1F35  F9 A0 14        ADCB TEMPR
  86    1F38  A7 03           STAA 03,X  STORE SUM
  87    1F3A  E7 02           STAB 02,X
  88    1F3C  A6 01           LDAA 01,X
  89    1F3E  E6 00           LDAB 00,X
  90    1F40  7D 1E B8        TST  DIREG UP OR DOWN?
  91    1F43  27 19           BEQ DOWN
  92    1F45  B0 1E B3       UP SUBA STEPL  UP!THEN DECREMENT
  93    1F48  F2 1E B2        SBCB STEPH
  94    1F4B  A7 05           STAA 05,X
  95    1F4D  E7 04           STAB 04,X  STORE NEW D-A DATA IN
  96                                     ◆NEXT LOCATION
  97    1F4F  FF A0 14        STX TEMPR  SAVE INDEX
  98    1F52  EE 04           LDX 04,X
  99    1F54  BC 1E B4        CPX HILIM COMPARE NEW X TO UPPER LIMIT
 100    1F57  27 2D           BEQ END GREATER? THEN GO TO END
 101    1F59  FE A0 14        LDX TEMPR  ELSE? RESTORE INDEX
 102    1F5C  20 17           BRA NEXT
 103    1F5E  BB 1E B3       DOWN ADDA STEPL  DOWN!THEN INCREMENT
 104    1F61  F9 1E B2        ADCB STEPH
 105    1F64  A7 05           STAA 05,X
 106    1F66  E7 04           STAB 04,X  STORE NEW X
 107    1F68  FF A0 14        STX TEMPR  SAVE INDEX
 108    1F6B  EE 04           LDX 04,X
 109    1F6D  BC 1E B6        CPX LOLIM COMPARE NEW X TO LO LIMIT
 110    1F70  27 0C           BEQ REVER  LESS? THEN TURN AROUND
 111    1F72  FE A0 14        LDX TEMPR  ELSE, RESTORE INDEX
 112    1F75  08             NEXT INX
 113    1F76  08              INX
 114    1F77  08              INX
 115    1F78  08              INX
 116    1F79  FF 1E BD        STX TREGW  SAVE NEW POINTER
 117    1F7C  20 82           BRA DATAT  NEXT POINT
 118    1F7E  7C 1E B8       REVER INC DIREG  CHANGE DIRECTION
 119    1F81  FE A0 14        LDX TEMPR
 120    1F84  20 EF           BRA NEXT  NEXT POINT
 121    1F86  FE 1E BD       END LDX TREGW  SAVE POINTER TO LAST DATA
 122                                              ◆POINT
 123    1F89  08              INX
 124    1F8A  08              INX
 125    1F8B  08              INX
 126    1F8C  08              INX
 127    1F8D  FF 1E BB        STX RREG
 128    1F90  FE 1E B9        LDX TREG  RESTORE ALL REGISTERS
 129    1F93  FF 1E BD        STX TREGW RESTORE WORKING T REGISTER
 130    1F96  BE A0 18        LDS SREG
 131    1F99  FE A0 16        LDX INREG
 132    1F9C  32              PUL A
 133    1F9D  06              TAP
 134    1F9E  33              PUL B
 135    1F9F  32              PUL A
 136    1FA0  39              RTS
 137    1FA1                  END

 TOTAL ASSEMBLER ERRORS =     0


                          DATA ACQUISITION


                               SYMBOL TABLE

◆  1

ADC       AFFC    CLKCT     1EAF    CONT      1F2B    DAC       AFF8
DATAT     1F00    DIREG     1EB8    DOWN      1F5E    END       1F86
HILIM     1EB4    INREG     A016    IOINT     A000    LOLIM     1EB6
NEXT      1F75    REVER     1F7E    RREG      1EBB    SERV      1F1C
SETUP     1ED7    SREG      A018    STEPH     1EB2    STEPL     1EB3
SWEEP     1EB1    TEMPR     A014    TIMER     AFF4    TREG      1EB9
TREGW     1EBD    UP        1F45
```

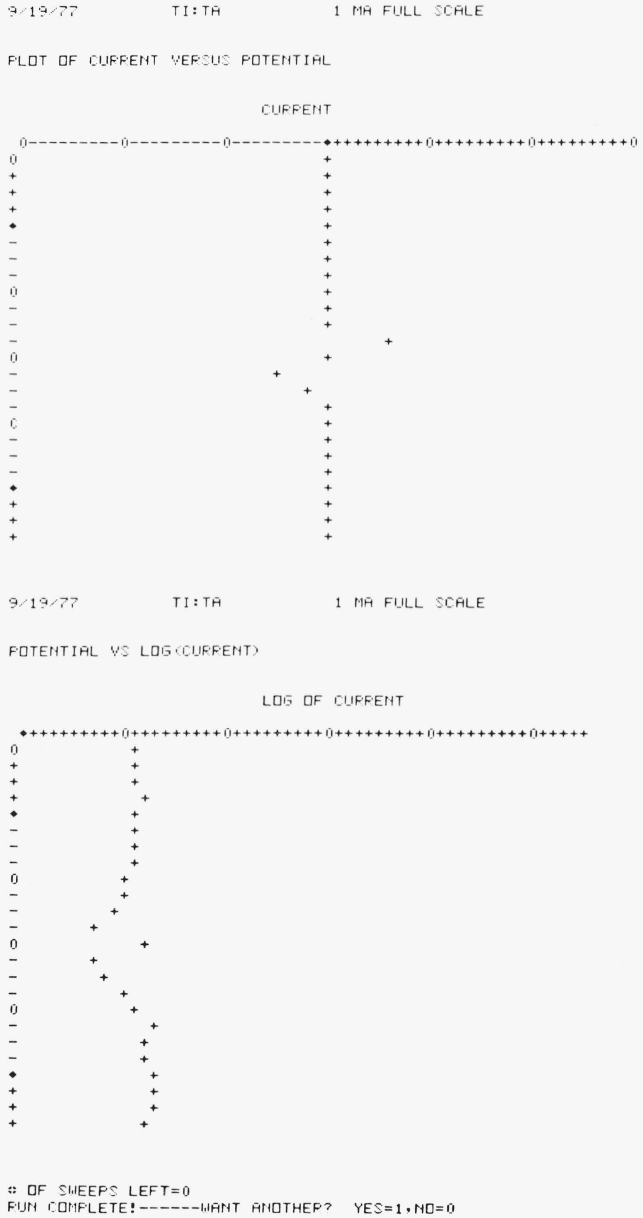
12. Appendix B. Sample of System Output.

Note: The full scale current referred to in the sample is the Potentiostat Full Scale setting and does not refer to the output data. Also note that the log of current listed in the table is the natural log of the absolute value of the measured current in Amperes and that the value plotted is the negative of this. The excessive number of significant figures shown is intended as an illustration of the range of precision of the BASIC program and does not reflect accuracy.

The actual scale for the current axis of the current versus potential plot runs from −5 to +5 mA, while the log of current scale on the next plot covers the range of zero to 55. The diamond character on the axes of both plots indicates zero. It should also be noted here that the reader should not expect the curves to be symmetrical about zero, since the asymmetry reflects the differences in the non-linear character of the charge transfer and transport processes at the electrodes.



```
READY
#RUN
ELECTROCHEMICAL DATA TAKER VERSION 2

DATE
? 9/19/77
SAMPLE
? TI:TA
FULL SCALE CURRENT-MA
? 1
ENTER TIME/COUNT
                CODE            TIME IN SEC
                0               10E1
                1               10E0
                2               10E-1
                3               10E-2
                4               10E-3
                5               10E-4
                6               10E-5
                7               10E-6
? 0
ENTER # OF COUNTS:0-255
? 2
# OF SWEEPS:0-255
? 1
STEP-SIZE(MLVTS-STEPS OF 10)
? 250
HILIM-MLVT
? 1000
LOLIM-MLVT
? -2000
TO START-ENTER 1, TO CHANGE PARAMETERS-ENTER 0
? 1


STARTED!
9/19/77         TI:TA           1 MA FULL SCALE


POTENTIAL(V)    CURRENT(MA)     LOG CURRENT
1               -0.014          -11.1764532
0.75            -0.014          -11.1764532
0.5             -0.014          -11.1764532
0.25            -0.01           -11.5129255
0               -0.014          -11.1764532
-0.25           -0.014          -11.1764532
-0.5            -0.014          -11.1764532
-0.75           -0.018          -10.9251388
-1              -0.034          -10.2891498
-1.25           -0.054          -9.82652653
-1.5            -0.082          -9.40879133
-1.75           0.938           -6.97176062
-2              6.E-03          -12.0237511
-1.75           -0.906          -7.00647127
-1.5            -0.29           -8.14562984
-1.25           -0.03           -10.414312
-1              -0.014          -11.1764532
-0.75           2.E-03          -13.1223634
-0.5            6.E-03          -12.0237511
-0.25           6.E-03          -12.0237511
0               2.E-03          -13.1223634
0.25            -2.E-03         -13.1223634
0.5             -2.E-03         -13.1223634
0.75            -0.01           -11.5129255
```



```
9/19/77         TI:TA           1 MA FULL SCALE

PLOT OF CURRENT VERSUS POTENTIAL

                         CURRENT

 0---------0---------0---------◆+++++++++0+++++++++0+++++++++0
0                              +
+                              +
+                              +
+                              +
◆                              +
-                              +
-                              +
-                              +
0                              +
-                              +
-                              +
-                                    +
0                              +
-                         +
-                            +
-                              +
0                              +
-                              +
-                              +
-                              +
◆                              +
+                              +
+                              +
+                              +


9/19/77         TI:TA           1 MA FULL SCALE

POTENTIAL VS LOG(CURRENT)

                         LOG OF CURRENT

 ◆+++++++++0+++++++++0+++++++++0+++++++++0+++++++++0+++++
0           +
+           +
+           +
+            +
◆           +
-           +
-           +
-           +
0          +
-          +
-         +
-       +
0            +
-       +
-        +
-          +
0           +
-             +
-            +
-            +
◆             +
+             +
+             +
+            +


# OF SWEEPS LEFT=0
RUN COMPLETE!------WANT ANOTHER?  YES=1,NO=0
```