THE DESIGN OF A MICROPROCESSOR-BASED DATA LOGGER

by Kenneth J. Leap, Jr. and Lee A. Dedini

U.S. Geological Survey

Open-File Report 82-167

Prepared as part of a continuing study of the
San Francisco Bay estuary

January 1982

UNITED STATES DEPARTMENT OF THE INTERIOR

JAMES G. WATT, Secretary

GEOLOGICAL SURVEY

OPEN-FILE REPORT

For additional information write to:

Regional Hydrologist
Water Resources Division
U.S.Geological Survey
345 Middlefield Road
Menlo Park, CA 94025

CONTENTS

TABLES

FIGURES

ABSTRACT

The design of a microprocessor-based data logger, which collects and digitizes analog voltage signals from a continuous-measuring instrumentation system and transmits serial data to a magnetic tape recorder, is discussed. The data logger was assembled from commercially-available components and can be user-programmed for greater flexibility. A description of the data logger hardware and software designs, general operating instructions, the microprocessor program listing, and electrical schematic diagrams are presented.

INTRODUCTION

This report describes the multi-channel data logger (DL) which records signals from the continuous water-sampling and -analysis systems on the U.S.Geological Survey-operated vessels: R/V Polaris and R/V Estero (Schemel and Dedini 1979; Dedini and Schemel 1980). The DL is an adaptation of the microprocessor based wind-prospecting system designed by Leap (1980). Up to 16 channels of analog signals are sampled and converted to digital data by the DL. A serial interface receives digital data from a fathometer. The DL transmits data and time from an internal clock to a tape recorder (Md 4923, Tektronix, Inc., Beaverton, Oregon)1/ via a serial interface (RS-232) at user selected time intervals. The tape recorder is compatible with certain computer systems.

At the time of design and fabrication, no commercially available data logger could meet our requirements without extensive modification. In particular, specialized hardware-interfaces are required to match the outputs of the water-analysis systems and to resolve the required number of significant digits. The user-programmable microprocessor allows for greater flexibility for future modification as our requirements change.

The text of this report is limited to a description of the hardware and software designs; additional relevant information can be found in Leap (1980). Operating instructions, a glossary, a program listing, and circuit diagrams make up the Appendix.

1/ The mention of brand names is for identification purpose and does not constitute endorsement by the authors or the U.S. Geological Survey.

HARDWARE CONFIGUATION

The data logger utilizes a microprocessor (CDP 1802, RCA Solid State, Somerville, NJ). The microprocessor combined with memory and input-output controllers form a microcomputer. The 16 analog channels are multiplexed, digitized by a 12-bit A/D converter, then read by the microcomputer for further processing (Fig.1) The serial input port converts serial data from a fathometer to parallel data for input to the microcomputer. Data sampling rates and data processing are determined by the microcomputer software and switch settings on the front panel of the logger (Fig.2). Data are transmitted via a RS-232 output port to the recorder. The format and frequency of the data transmissions are also determined by software and switch settings.

The data logger was fabricated in modules that are functionally and physically on separate circuit cards. Figure 3 shows how the modules are integrated into the system. Each of the 10 modules shown in Fig.3, is described in the text of this report. The hardware is contained in a metal cabinet that is 12" wide by 12" deep by 7" high. The following is a brief description of the front panel switches and displays (Fig.2) and the rear panel connections and their functions. Connections to the front panel components are detailed at the end of the hardware configuration section.

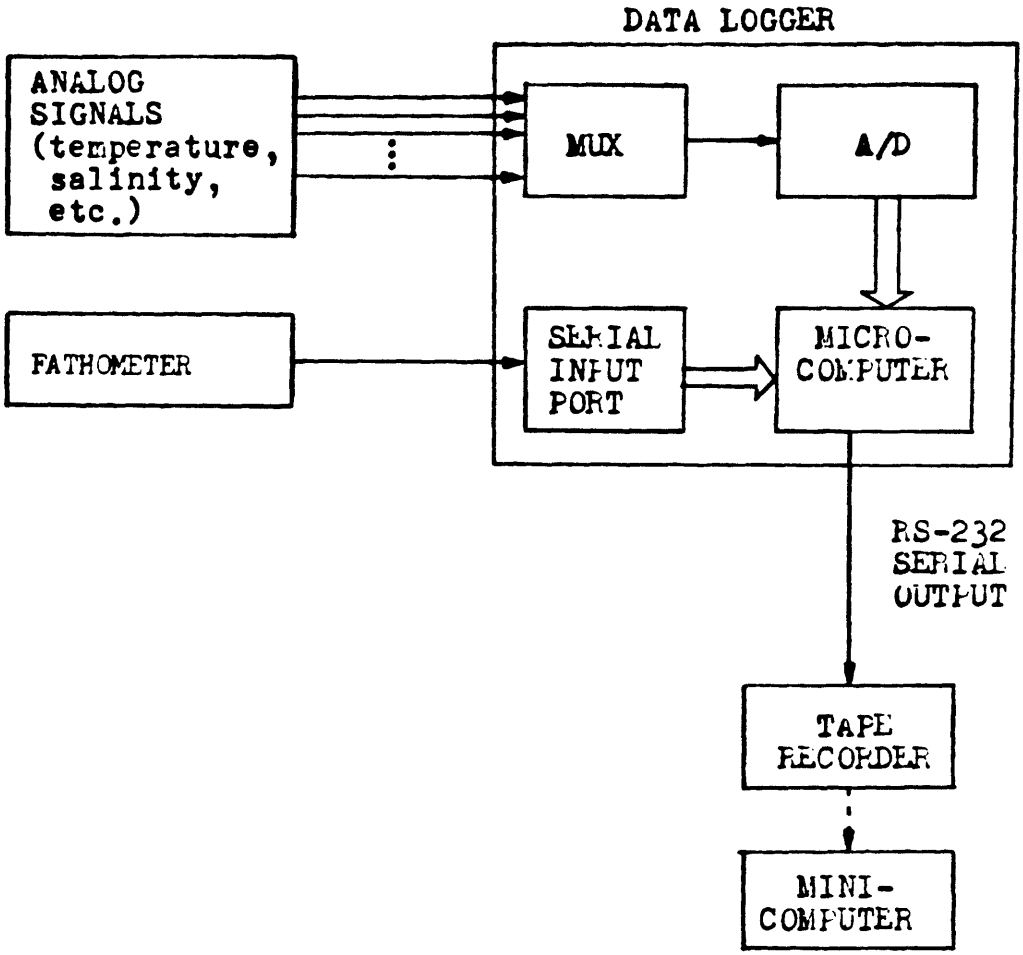


Figure 1

Block Diagram of the Data Logger in a Water-Sampling System

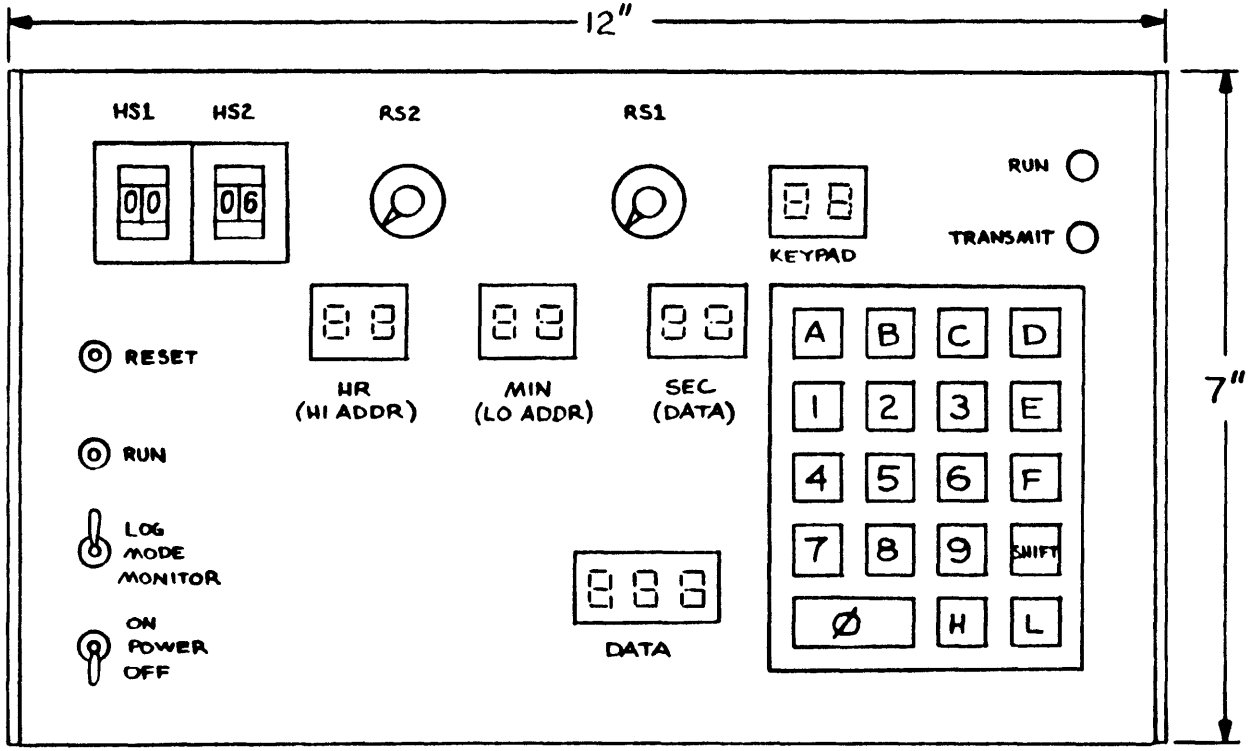


Figure 2

Data Logger Front Panel

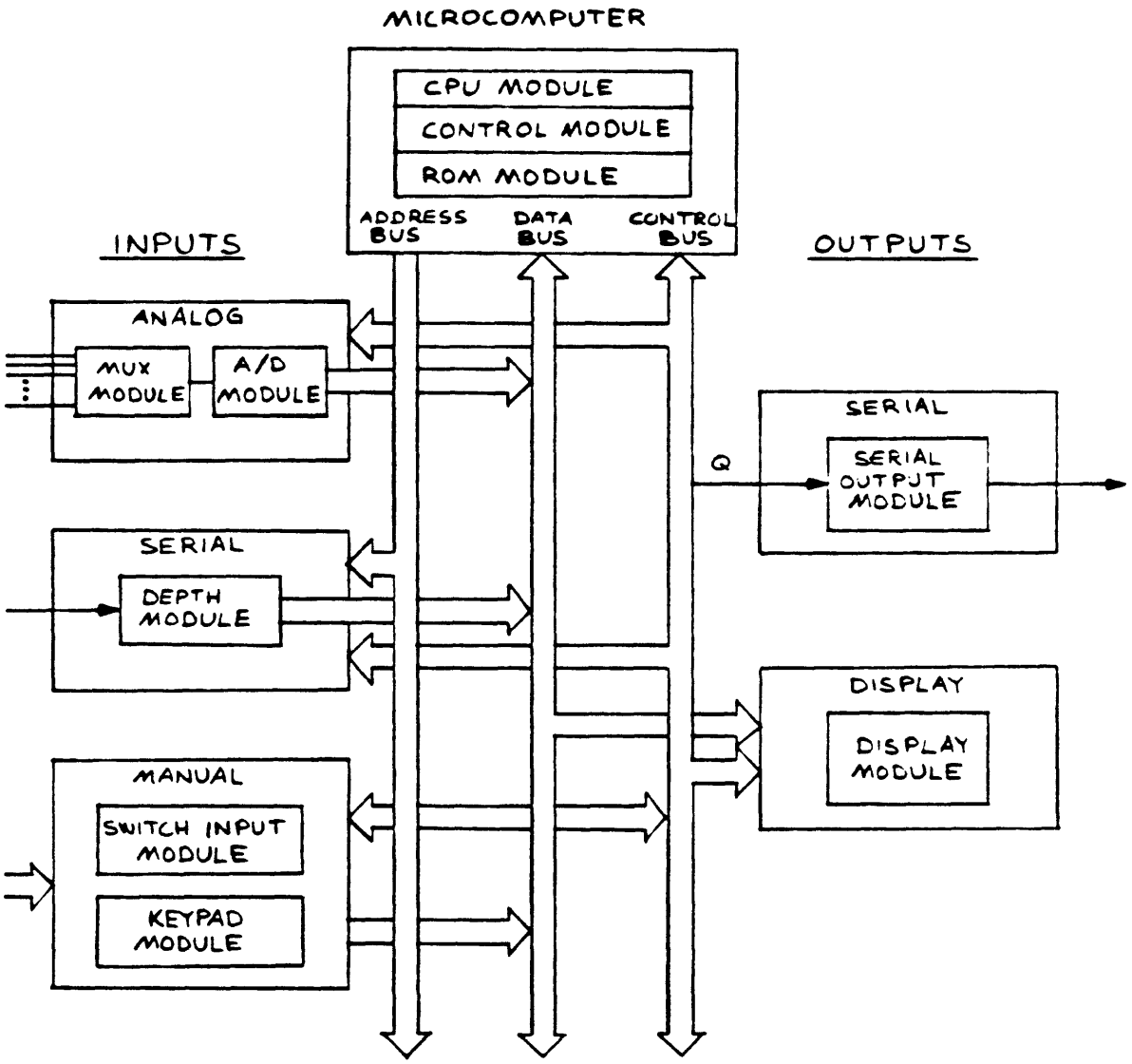


Figure 3

Block Diagram of the Data Logger

DESCRIPTION OF FRONT PANEL

MODE SWITCH - The DL has two modes of operation determined by a switch on the front panel. In the LOG mode the logger collects data and transmits them over the serial interface. In the MONITOR mode the microcomputer executes a program determined by the user. In the present configuration a program which sets the real-time clock is executed when the DL is placed in the MONITOR mode.

RUN SWITCH - Pressing the "RUN" switch executes the program when the DL is in the MONITOR mode.

RESET SWITCH - The "RESET" switch will manually reset the DL when running a program in the MONITOR mode. The following digits will then appear on the panel display: 03 00 30.

SCAN INTERVAL SWITCH - Hexadecimal switch HS2 selects the scan interval, which is the frequency at which the DL transmits data and time to the tape recorder. The scan interval (in seconds) is equal to the HS2 switch setting multiplied by 5. For example, if HS2 is set on "06", then data are transmitted every 30 seconds.

KEYPAD - In the MONITOR mode the keypad is used to enter address and data information. The keypad can also be used for the operation of an executed program. In the LOG mode the keypad is used for selecting an analog data channel to be displayed in the DATA display (see below).

TRANSMIT LIGHT - The "TRANSMIT" light is on when serial data are transmitted to the tape recorder.

RUN LIGHT - The "RUN" light indicates that the microcomputer is in the RUN mode. This light will also blink once every five seconds when the logger is in in the LOG mode, because the microcomputer is in a RESET mode.
DISPLAYS - In the MONITOR mode, the DL high address, low address, and data will appear on the HR/HI ADDR, and MIN/LO ADDR, and SEC/DATA displays, respectively. In the LOG mode, the real-time clock will be displayed. The keypad display shows the last two keys (0-F) that were pressed. The DATA display shows, the signal from one of the analog channel inputs in 12-bit hexadecimal notation; this channel (0-15) is selected by pressing one of the keys (0-F). Data appear every five seconds.

DESCRIPTION OF REAR PANEL

POWER SOURCE - The DL requires a 115VAC, 60Hz power source. The case is earth grounded.
CONNECTIONS - A 25-pin male terminal connector J1 (type DB-25P), is the 16 analog-signal input. The analog-signal input voltage range to the DL is -0.5V to +4.5V. A 25-pin female terminal connector J2 (type DB-25S), is the RS-232 serial output to the tape recorder. A 4-pin female terminal connector J3 (Molex type), is the connection to the fathometer.

BUS STRUCTURE

Most of the connections between the 10 modules of the data logger are made via a 44-pin bus. Bus pin numbers and connections to the microprocessor chip are shown in Table 1, along with the corresponding 1802 signal name.

Following are brief functional descriptions of some of the 1802 signals that are used in this system.

1802 Signals:

$\overline{EF1}$ - $\overline{EF4}$	These are four inputs to the microprocessor, controlled by keypad inputs and tested by branch instructions.
A0 - A7	These are the eight memory-address lines. They provide the 16-bit memory-address in the form of two successive 8-bit bytes. These bus lines are buffered by the CPU module.
TPA	This is a timing signal; one function is to latch one byte of the multiplexed address.
TPB	One of the functions of this timing signal is to take memory data from the bus during an output instruction.
N0 - N2	This 3-bit code is generated by the input/output instruction and is used by this system to select one of seven input or output ports.
D0 - D7	These eight lines form the 8-bit bidirectional data bus.
$\overline{MWR}$, $\overline{MRD}$	The state of these two lines determines whether a byte is to be read from the address memory location, written into it, or neither.
$\overline{INT}$	This is the interrupt line. An interrupt is used to advance the software-controlled real-time clock.
SC0, SC1	These signals form the 2-bit state code, which permits synchronization of I/O circuits with the microprocessor operating instructions.
Q	This output flag is toggled (having either one of two states at a given time) under program control to provide serial data output.

Table 1

Bus-Signal Pin Assignments

Signal name	Bus pin number	CPU pin number	Signal name	Bus pin number	CPU pin number
$\overline{\text{EF4}}$	1	21	NO	A	19
$\overline{\text{EF3}}$	2	22	N1	B	18
$\overline{\text{EF2}}$	3	23	N2	C	17
$\overline{\text{EF1}}$	4	24	VBAT	D	
A0	5	25	D0	E	15
A1	6	26	D1	F	14
A2	7	27	D2	H	13
A3	8	28	D3	J	12
A4	9	29	D4	K	11
A5	10	30	D5	L	10
A6	11	31	D6	M	9
A7	12	32	D7	N	8
TPB	13	33	$\overline{\text{MRD}}$	P	7
TPA	14	34	SCO	R	6
$\overline{\text{MWR}}$	15	35	SC1	S	5
$\overline{\text{INT}}$	16	36	Q	T	4
$\overline{\text{DMA OUT}}$	17	37	$\overline{\text{STROBE}}$	U	
$\overline{\text{DMA IN}}$	18	38	VSTB	V	16, 40
XTAL	19	39	CLOCK	W	1
$\overline{\text{EXT CLEAR}}$	20			X	
$\overline{\text{WAIT}}$	21	2		Y	
GND	22	20	GND	Z	20

Clock — A single-phase clock input on this line determines the operating speed of the microprocessor. a 1.78 MHz signal from the control module provides the clock input.

The following external signals are not 1802 signals but are signals peculiar to this system that are also found on the system bus.

External Signals:

$\overline{\text{EXT CLEAR}}$ — This signal directly controls the RUN and RESET mode of the microprocessor. When $\overline{\text{EXT CLEAR}}$ =1, the microprocessor is in the RUN mode. When $\overline{\text{EXT CLEAR}}$ =0, the microprocessor is in the RESET mode. $\overline{\text{EXT CLEAR}}$ is also used to "enable" the RAM.

VBAT — A continuous 5 volts DC from the power module is supplied on this line.

$\overline{\text{STROBE}}$ — This signal is generated by the control module. When $\overline{\text{STROBE}}$ is low, 5 volts is applied to VSTB. (Not used in present configuration.)

VSTB — The CPU module is power-strobed by VSTB; it is activated when $\overline{\text{STROBE}}$ is low. (Not used in present configuration.)

GND — GND is the common voltage for all of the circuitry and is isolated from earth ground and chassis.

CONTROL MODULE

The control module controls the LOG and MONITOR modes of the data logger and the RUN and RESET modes of the microprocessor, provides a clock input for the microprocessor, and issues interrupts.

The data logger has two modes of operation determined by a switch on the front panel. In the LOG mode the logger collects data and transmits them over the serial interface. In the MONITOR mode the microcomputer executes programs chosen by the user. In the present configuration the program which sets the real-time clock is executed when the logger is in the MONITOR mode.

LOG mode. In the LOG mode the microcomputer operates in five-second cycles. A cycle begins with the positive transition of the START signal (refer to Fig.4). Two cycles (of the 60 Hz clock) after the START signal goes high $\overline{\text{EXT CLEAR}}$ goes high. (The microcomputer is in the RUN mode when $\overline{\text{EXT CLEAR}}$=1 and in the RESET mode when $\overline{\text{EXT CLEAR}}$=0.) The microcomputer executes the data-logging program while in the RUN mode (i.e., when $\overline{\text{EXT CLEAR}}$=1) until the execution of the OUT 6 instruction. This instruction triggers the INTERNAL RESET signal, which puts the microcomputer back in the RESET mode ($\overline{\text{EXT CLEAR}}$=0). The microcomputer remains in the RESET mode until the next positive transition of the START signal, then the cycle is repeated.

The $\overline{\text{CLR}}$ signal is inserted after the positive transition of the START signal. This is to make sure that the computer is reset every five seconds; it prevents the microcomputer from

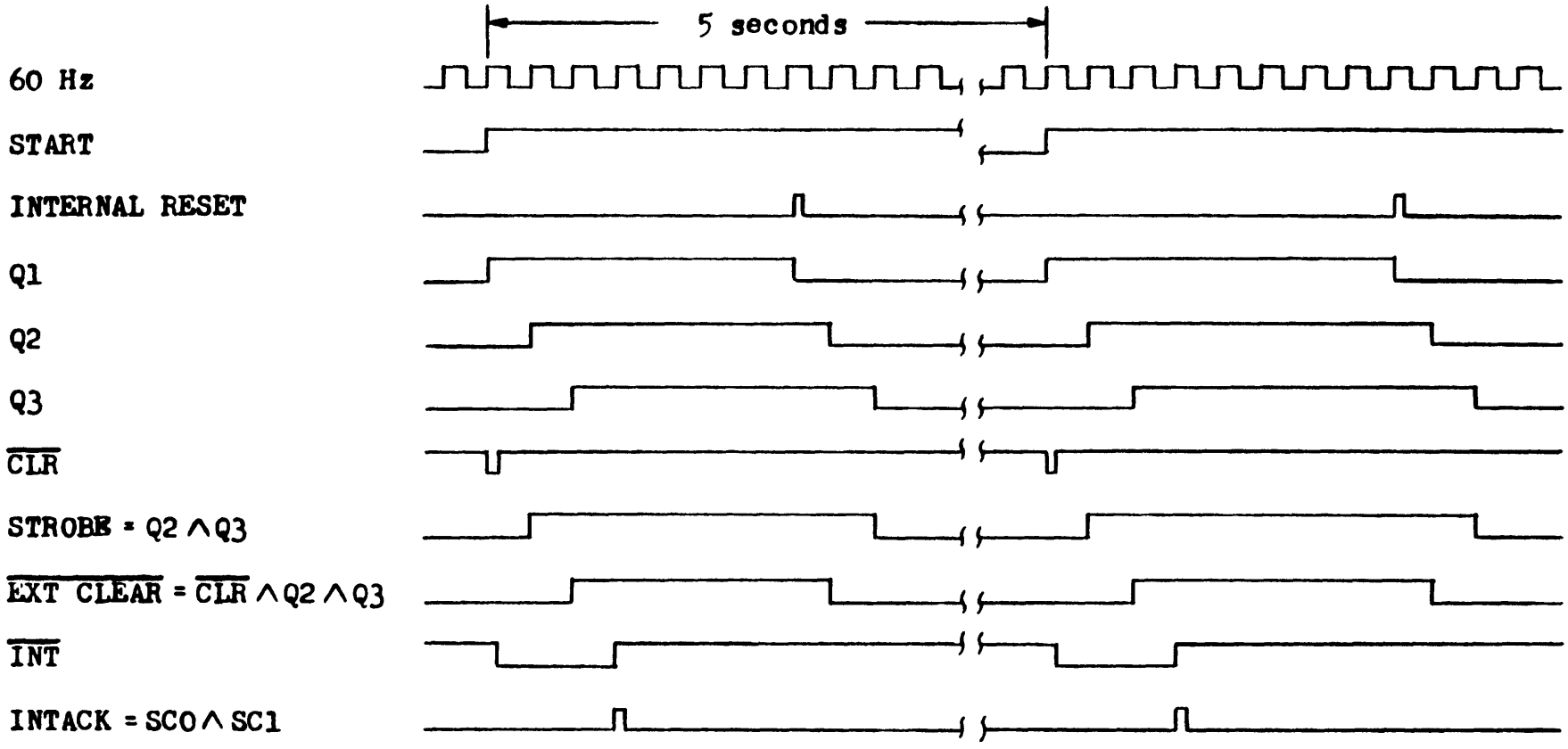


Figure 4

Timing Diagram for the Control Module
in the LOG Mode

being "hung" in a loop due to erroneous program execution. The $\overline{CLR}$ signal has no effect during a normal cycle because the microcomputer is already in the RESET mode when $\overline{CLR}$ is asserted.

The $\overline{INT}$ signal (interrupt request) is also asserted during the positive transition of the START signal. $\overline{INT}$ is held low until the interrupt is serviced (i.e., when INTACK goes high).

The $\overline{STROBE}$ signal is not used in the present configuration of the data logger. The signal is used in low power-consumption applications requiring power strobing of the CPU and other modules.

MONITOR mode. The only "hardware" difference between the MONITOR and LOG modes is that in the MONITOR mode the $\overline{CLR}$ signal is disabled. This allows the microcomputer to be in the RUN mode continuously without being reset every five seconds. The microcomputer can be reset with the INTERNAL RESET signal (under software control) but this is not typically done in the MONITOR mode. Interrupts occur at five-second intervals as in the LOG mode. This allows the interrupt driven clock to function in both the LOG and MONITOR modes. (See Fig.5 for a timing diagram for the control module in the MONITOR mode.)

CPU AND ROM MODULES

The CPU (Central Processing Unit) module contains the microprocessor, 2048 bytes of EPROM, and 256 bytes of RAM. The program stack, the real-time clock, and a "scratchpad" storage are maintained in this RAM.

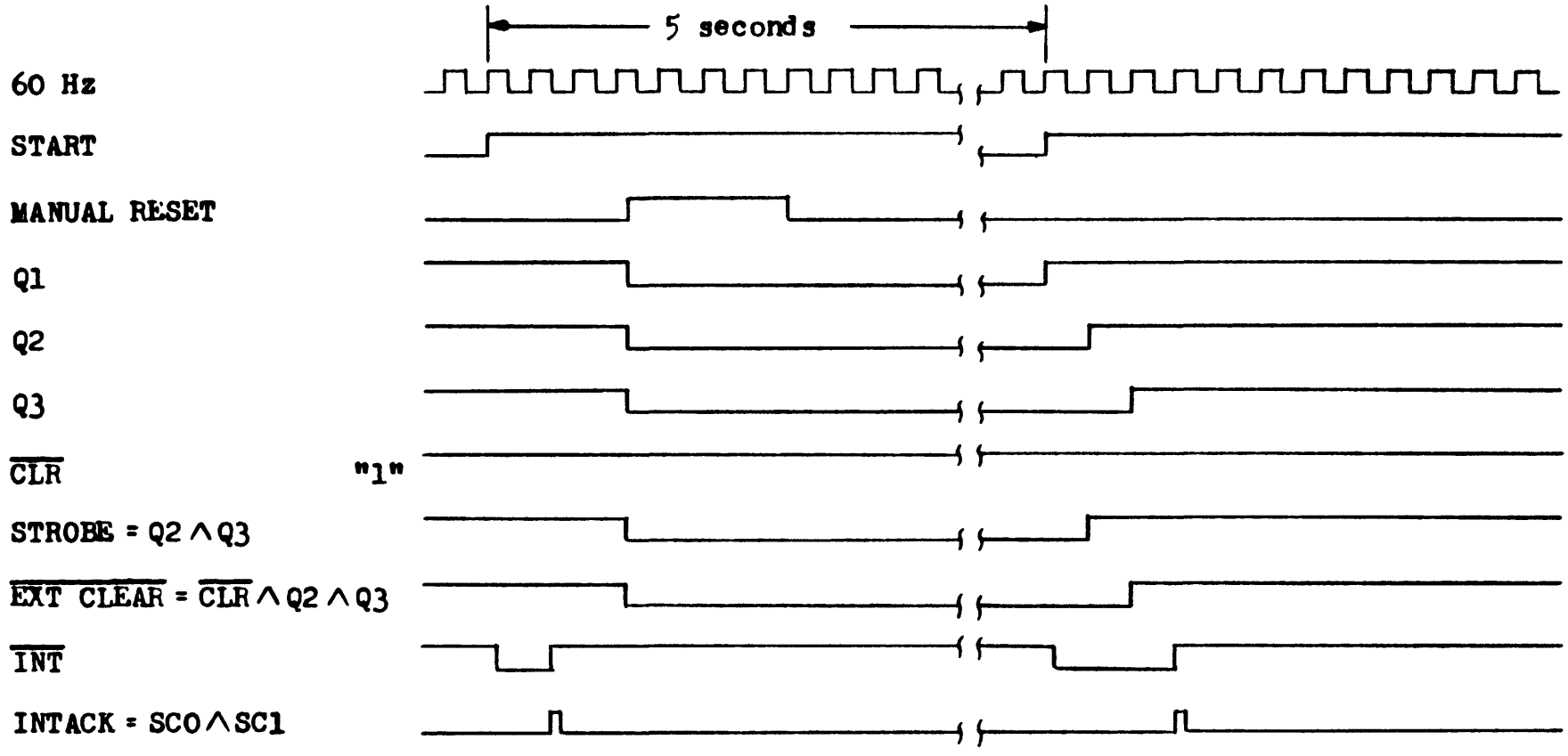


Figure 5

Timing Diagram for the Control Module
in the MONITOR Mode

The 16-bit memory address is multiplexed in the form of two 8-bit bytes. The high address byte appears on the address bus followed by the low byte. The high byte is strobed into an 8-bit register by the TBA timing signal so that the full 16-bit address is available to the memory devices as soon as the low byte appears on the address bus. The upper eight address bits are encoded to give the 256 bytes of RAM a starting address of 8C00. The upper five address bits are encoded to give the 2048 bytes of EPROM a starting address of 0000.

The ROM module consists of only a ROM chip and a power by-pass capacitor. The ROM (CDPR582, RCA Corporation, Somerville, NJ) provides subroutines used in programing the microprocessor (See RCA CDP 1802 Microprocessor User Manual). Pin assignments for the ROM are shown in Table 2.

MULTIPLEXER AND A/D MODULES

The multiplexer module selects one of the 16 analog signals from the data cable. The selected signal is offset by 0.5V and filtered. A 4-bit output port addresses the multiplexer for each one of the 16 analog signals. The output port latches the four most significant bits (D7 - D4 on the bus) from the OUT 5 instruction.

A three-stage op-amp circuit buffers, filters, and offsets the incoming signals. The circuit is a second-order low pass filter with a time constant of 1.0 microsecond. The filter rejects high frequency electronic noise. The time constant is small so that scan rates can be achieved. The output signal settles to 99.95 % of its final value 10 microseconds after the

Table 2

CDPR582 (ROM) Pin Assignments

CDPR582		Data Logger Bus	
Name	Pin	Name	Pin
MA7	1	A7	12
MA6	2	A6	11
MA5	3	A5	10
MA4	4	A4	9
MA3	5	A3	8
MA2	6	A2	7
MA1	7	A1	6
MA0	8	A0	5
$\overline{\text{MUX}}$	23	$\overline{\text{TPA}}$	14
$\overline{\text{MRD}}$	19	$\overline{\text{MRD}}$	P
BUS7	17	D7	N
BUS6	16	D6	M
BUS5	15	D5	L
BUS4	14	D4	K
BUS3	13	D3	J
BUS2	11	D2	H
BUS1	10	D1	F
BUS0	9	D0	E
VSS	12	GND	22,Z
VDD	24	VBAT	D
CS1	21	VBAT	D
CS2	20	VBAT	D
CL0	18	N.C.	
CLI	22	N.C.	

(step) input signal is applied. The signal is offset by 0.5V. The steady state response of the filter is

$$V_{out} = V_{in} + 0.5\ V\ .$$

The A/D module takes the analog signal selected by the multiplexer module and performs a 12-bit analog-to-digital conversion in response to the OUT 7 instruction. The 12-bit word is read into the microcomputer with the INP 4 and INP 5 instructions. Table 3 shows how the module responds to the microprocessor's I/O instruction.

DEPTH MODULE

The depth module takes serial data from a fathometer (Md 2700, Datamarine, Pocasset, MA.) and converts them to parallel data for input to the microcomputer. A functional circuit diagram of the depth module is shown in Fig.6.

Input Signals. The $\overline{\text{CLOCK}}$ and $\overline{\text{DATA}}$ signals from the depth sounder are optically isolated from the depth module. The $\overline{\text{CLOCK}}$ signal directly drives the optical isolator, U1, with current-limiting resistor R1. The $\overline{\text{DATA}}$ signal is buffered with transistor Q1 to drive optical isolator U2.

A timing diagram of the $\overline{\text{CLOCK}}$ and $\overline{\text{DATA}}$ signals is shown in Fig.7. Serial data bits (inverted) are valid for one clock cycle following each negative transition of the clock. The $\overline{\text{CLOCK}}$ and $\overline{\text{DATA}}$ signals are isolated and inverted giving the CS and DS signals, respectively (refer to Fig.7). CS and DS lag behind the $\overline{\text{CLOCK}}$ and $\overline{\text{DATA}}$ signals because of rise and fall times of several microseconds at the outputs of the optical isolators. The times by which CS and DS lag the $\overline{\text{CLOCK}}$ and $\overline{\text{DATA}}$ signals,

Table 3

The Use of I/O Instructions
with the A/D Module

instruction	action
OUT 7	start A/D conversion
INP 4	4 most significant bits → $D_{3:0}$
INP 5	8 least significant bits → $D_{7:0}$

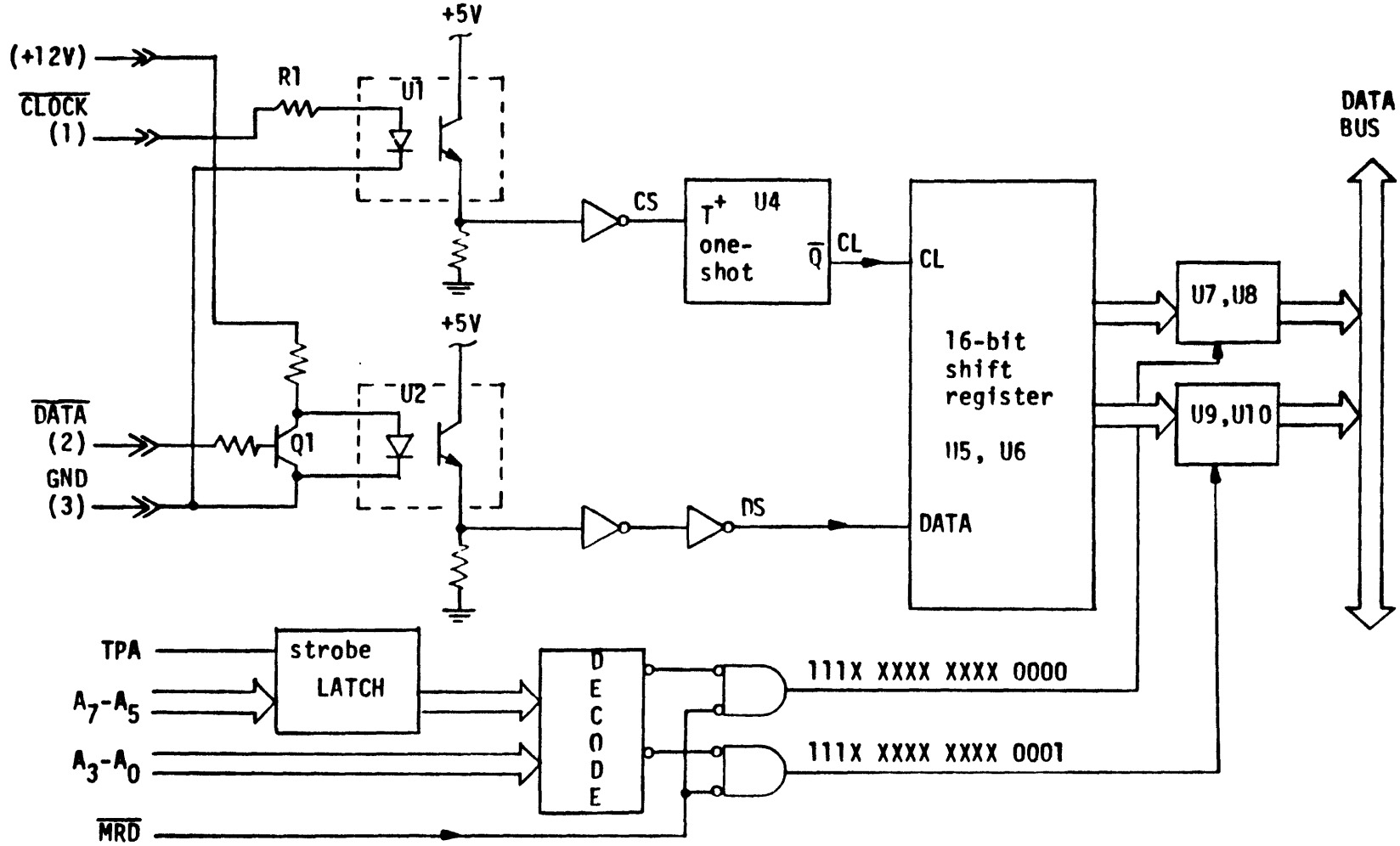


Figure 6

Depth Module Functional
Circuit Diagram

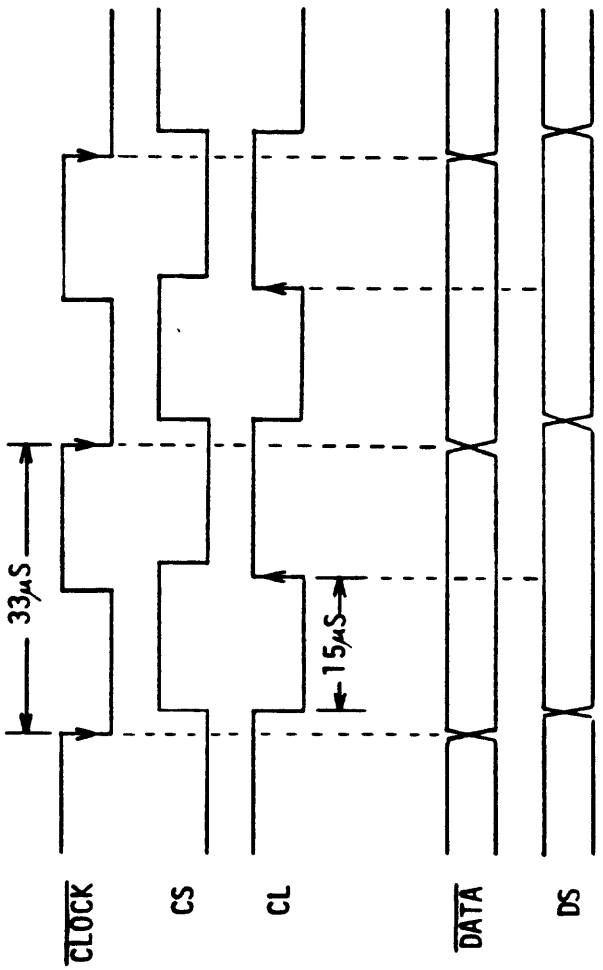


Figure 7

Timing Diagram for Depth Module
Input Signals

respectively, depend upon the characteristics of each optical isolator. The lag times of CS and DS cannot be assumed to be the same; therefore, CS cannot be used to clock data from DS into the shift register. This problem is corrected by shifting CS to approximate center of the time interval for which the corresponding data bit on DS is valid. The rising edge of CS is delayed by about 15 microseconds by a one-shot. The resulting signal, CL, is used to clock data from DS into the 16-bit shift register (See Fig.7).

Outputs. The 16-bit parallel data at the outputs of shift registers U5 and U6 are connected to bilateral switches U7-U10 (refer to Fig.6). Memory-mapped I/C is used to connect these 16 bits of data to the data bus eight at a time. Table 4 shows the information provided at each memory-mapped address.

DISPLAY AND I/O MODULES

In this section, the display, switch-input, keypad, and serial-output modules are described. These modules provide the data logger with its basic I/O capabilities.

Display Module. The display module is made up of five output ports which display hexadecimal or decimal numbers. Figure 8 indicates which output instructions provide the data for the front-panel displays.

Switch-Input Module. The switch-input module is used to read the positions of the rotory and hexadecimal switches on the front panel. The rotory switch positions are encoded by the module. Table 5 shows the correspondence between switches and input ports.

Table 4

Depth Module Outputs

Memory-Mapped Address																Output-Data Bits*							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	1	X	X	X	X	X	X	X	X	X	0	0	0	0	AI	FM	FT	DP	D2	C2	B2	A2
1	1	1	X	X	X	X	X	X	X	X	X	0	0	0	1	D1	C1	B1	A1	D0	C0	B0	A0

X = don't care (either 1 or 0).

*Meaning of output-data bits:

AI	Alarm
0	off
1	on

DP	Decimal Point
0	off
1	on

FM	FT	Depth Units
1	0	fathoms
0	1	feet
0	0	meters

BCD Bits				Digit
D2	C2	B2	A2	100's
D1	C1	B1	A1	10's
D0	C0	B0	A0	1's

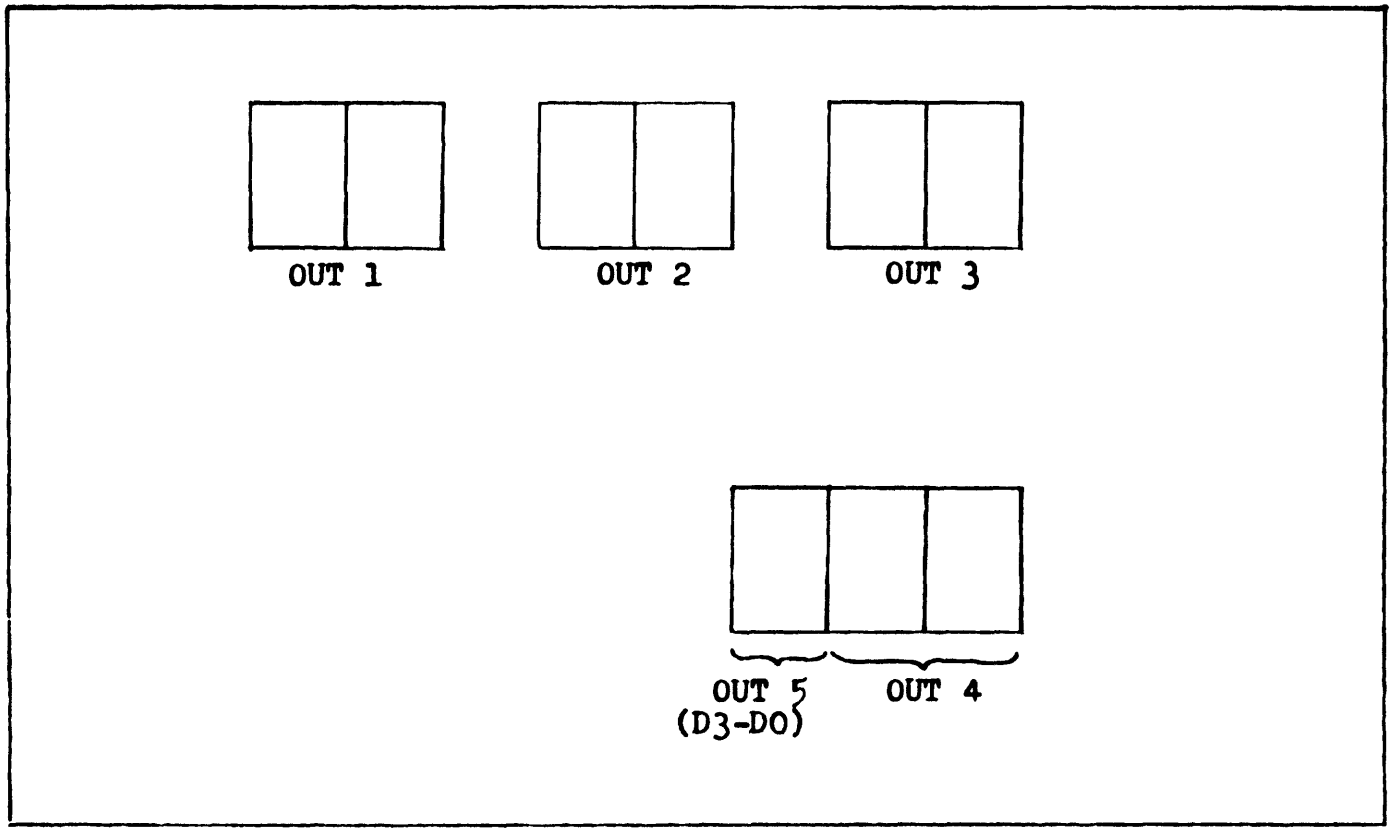


Figure 8

Locations of Display Output Ports on
the Front Panel

Keypad Module. The keypad module provides two kinds of inputs to the microcomputer: (1) a 16-key encoder allows parallel data input with a hexadecimal keypad and (2) closures of the "H", "L", and and "SHIFT" keys are debounced and connected to three input flags of the microprocessor. The INP 3 instruction reads the eight bits of data created by two successive key strokes on the hexadecimal keypad. The "H", "L", and "SHIFT" keys are connected to the $\overline{EF1}$, $\overline{EF2}$, and $\overline{EF4}$ input flags, respectively.

Serial-Output Module. The serial-output module consists of a transistor circuit which shifts voltage levels of the microprocessor's output flag, Q, in accordance with the EIA RS-232 standard for serial communication. The parallel-to-serial conversion of the output is done with software, eliminating the need for a UART (Universal Asynchronous Receiver and Transmitter) integrated circuit.

POWER AND PANEL CONNECTIONS

This section describes the elements of the data logger that are not plugged into the bus. The locations of these elements and their interconnection are shown.

Front Panel. Hexadecimal thumbwheel switches HS1 and HS2, and rotory switches RS1 and RS2 (Fig.2) are connected to input ports on the switch-input module (rotory switches RS1 and RS2 are not used in the present configuration). The RESET button and MODE switch are connected to the control module.

The keypad, keypad display, RUN button, and RUN and TRANSMIT lights are all connected to the keypad module. Circuitry for the keypad display is mounted on the back of the front panel (a circuit diagram is in the Appendix).

Table 5

Rotary and Hexadecimal Switch
Input Ports

Switch	input instruction	data bits
RS1 (9 position rotary switch)	INP 1	D3-D0
RS2 (5 position rotary switch)	INP 1	D7-D4
HS1 (hexadecimal switch)	INP 6	D7-D0
HS2 (hexadecimal switch)	INP 7	D7-D0

The HR, MIN, SEC, and DATA display LED's are mounted on the display module, located directly behind the front panel.

The POWER switch connects the hot AC line to the two power supplies.

Power Supplies. A line-operated +5V, 2 ampere power module (UPM-5/2000, Datel Systems, Inc., Mansfield, MA.) powers the displays and all of the digital hardware. A line-operated +15V, 200 milliampere power module (Md 920, Analog Devices, Norwood, MA.) powers the analog circuitry. One 1/2 ampere slow-blow line fuse is used for both supplies.

Interconnections. A terminal strip is used for all power connections and connections between modules which are not on the bus. Table 6 and Fig.9 describe the terminal connections. Wires labled 1 thru 8 correspond to terminals labled T1 thru T8. Figures 10 and 11 show the locations of the power modules and the terminal strip.

Table 6

Terminal Connections

Terminal	Signal name	Connections
T1	ANALOG IN	multiplexer module to A/D module
T2	REF OUT	A/D module to multiplexer module
T3	+15 V	power module to A/D module and multiplexer module
T4	VBAT (+5V)	power module to bus pin D
T5	COMMON	power module, bus pins 22 and Z, data cable
T6	-15 V	power module to A/D module, multiplexer module, and serial output module
T7	AC HOT	AC line to power modules
T8	AC NEUTRAL	AC line to power modules
(chassis)	AC GROUND	AC line to chassis

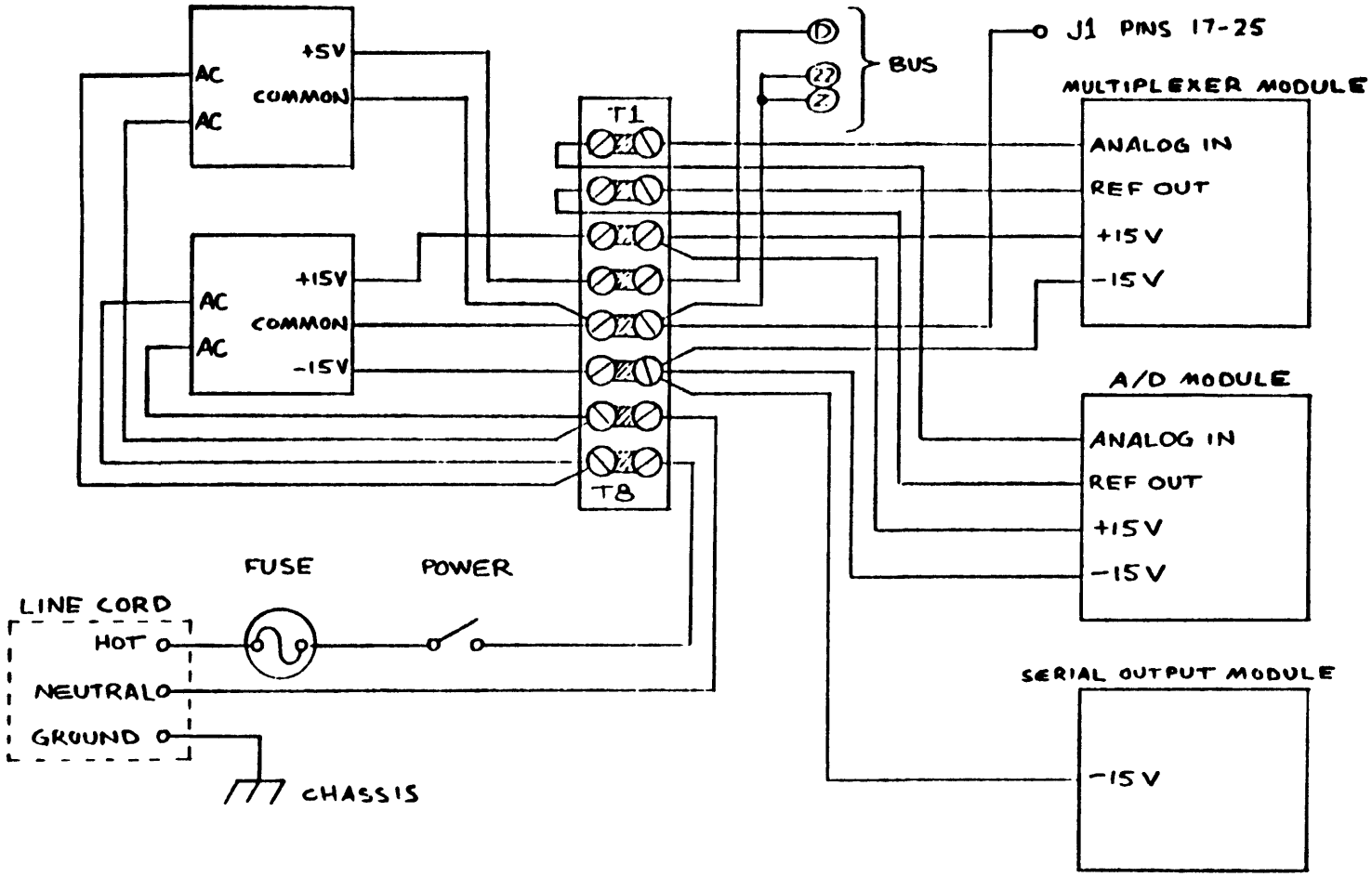


Figure 9

Terminal Connections Diagram

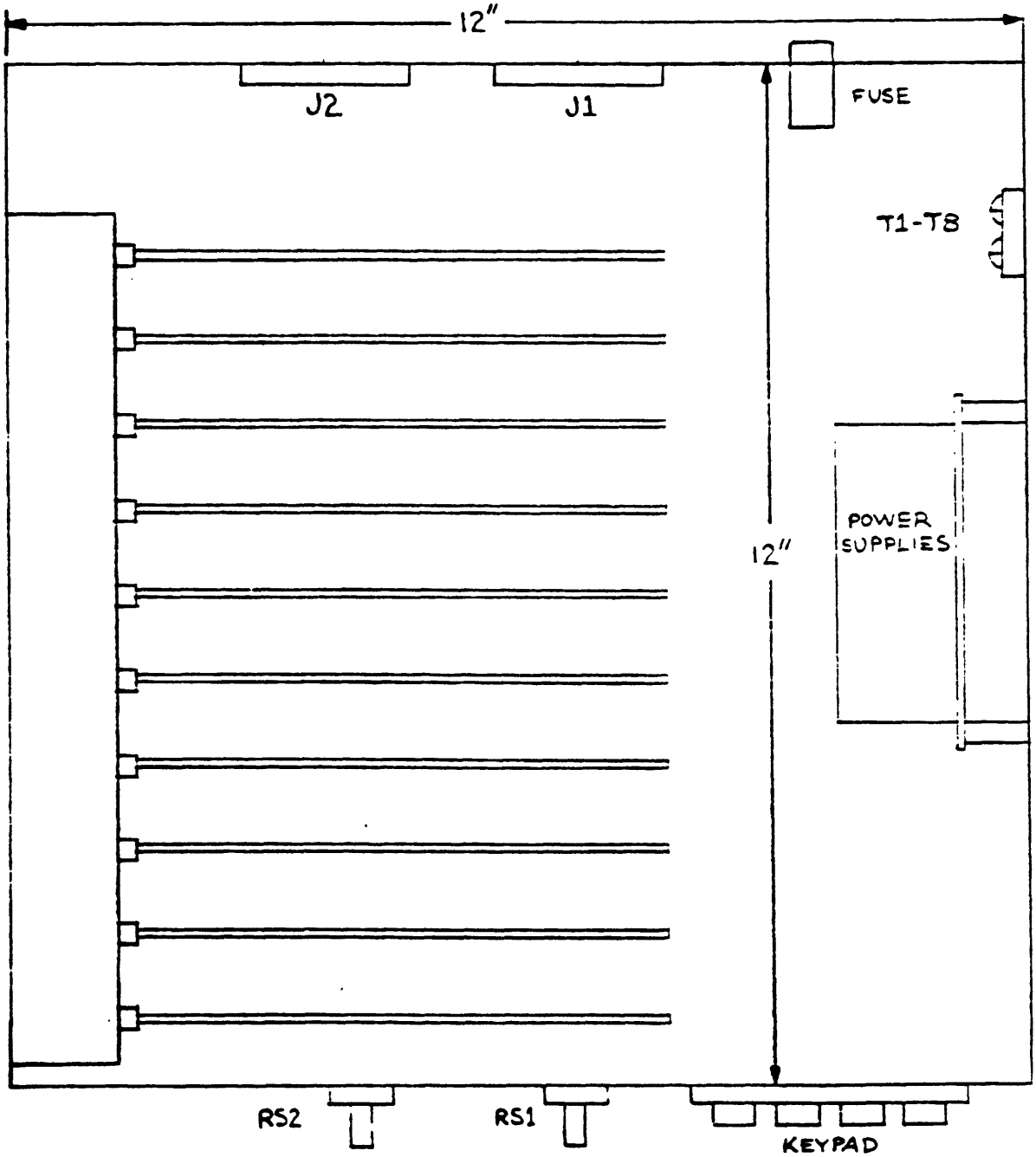


Figure 10

Data Logger Top View

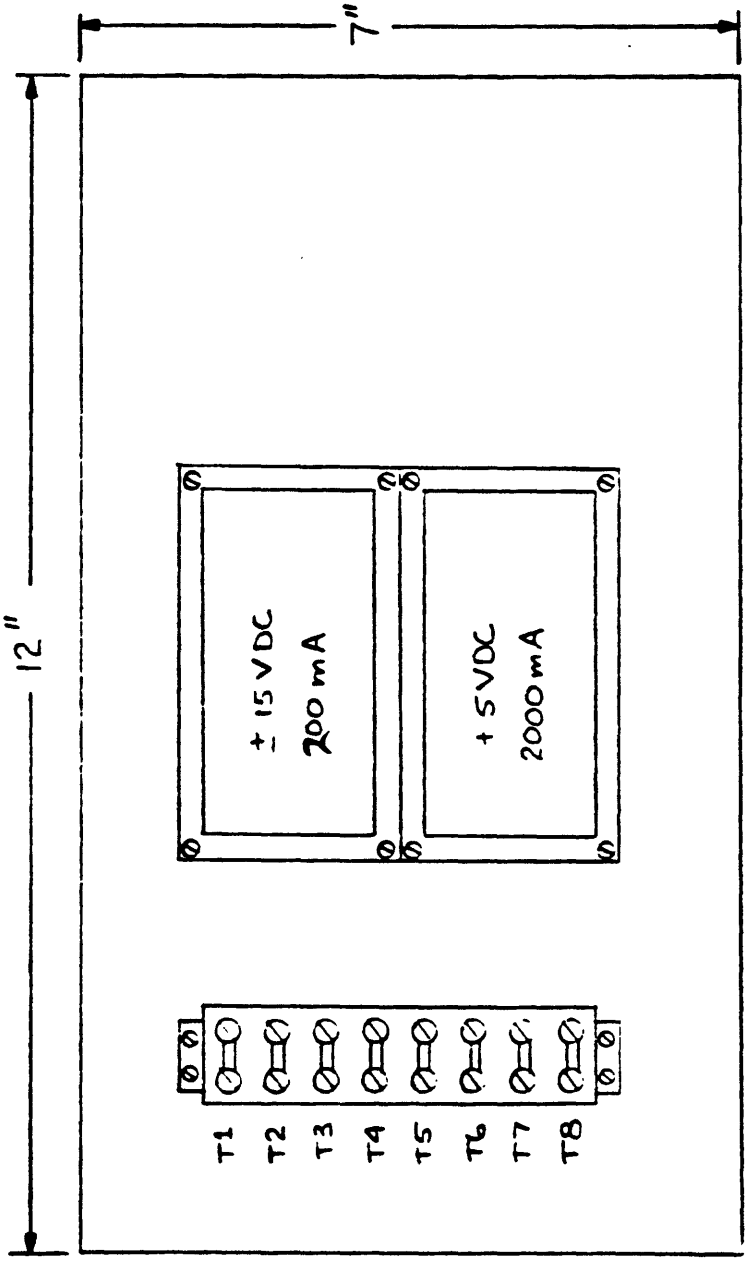


Figure 11

Data Logger Side Panel

SOFTWARE CONFIGURATION

The program used in the data logger is described in this section. An overview of the entire program is presented first, followed by a more detailed description of the main routines within the program. A list of programs and subroutines, a map of RAM storage, a table of I/O ports, and a complete program listing is in the Appendix.

A flow chart of the program is shown in Fig.12. Program execution begins at memory address 0000 after a RESET. Four types of RESETs are possible: (1) a "power on RESET", (2) a manual RESET from a push button on the front panel, (3) a program controlled INTERNAL RESET, and (4) a RESET from the insertion of the $\overline{CLR}$ signal (see Fig.4). After a RESET, interrupts are disabled and initialization takes place. Initialization consists Bf setting up various program counters, stack pointers, and data-memory pointers. After initialization the MODE switch (front panel) is read, resulting in a conditional branch to the LOG routine or MONITOR routine.

LOG ROUTINE

The LOG routine is executed when the LOG mode is selected by the front-panel switch. This routine samples data every five seconds and displays one of the 16-channels selected by the keypad. Data are transmitted over the serial-output port at regular time intervals determined by a panel switch.

At the start of the LOG routine, a flag (INTF) is reset to "0" and interrupts are enabled (refer to Fig.12). The LOG routine then waits for an interrupt to be serviced by polling

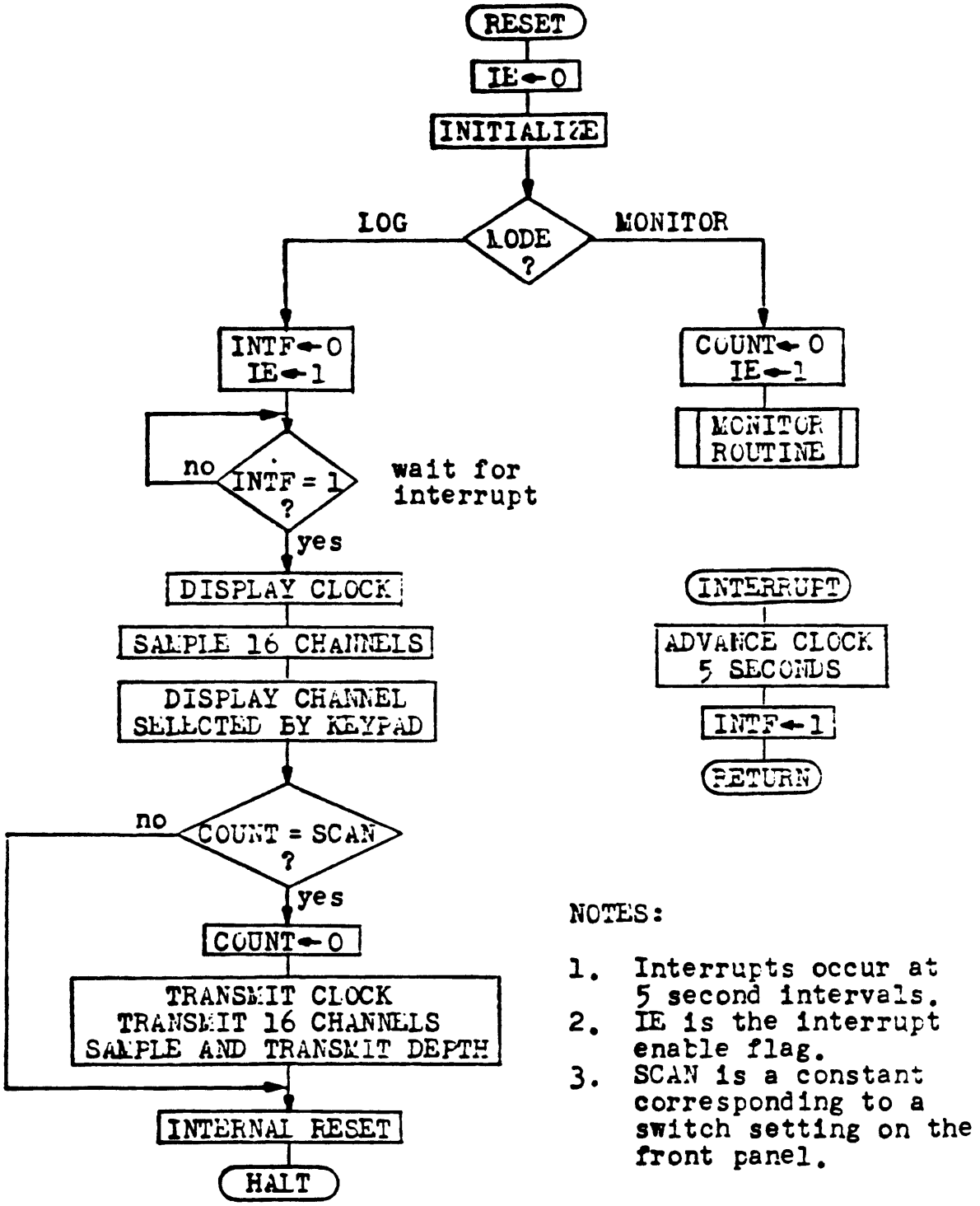


Figure 12

Program Flow Chart

the INTF flag (the interrupt-service routine sets this flag to "1"). The real-time clock, which is updated by the interrupt-service routine is then displayed on the front panel. Next, 16 channels with analog data are sampled and stored in memory. One of these 16 channels is selected by the least-significant hexadecimal digit in the keypad register for display on the front panel.

The constant, SCAN, determines the scan interval. When COUNT (which is incremented every five seconds) equals SCAN, the scan interval has elapsed and data are transmitted over the serial interface. Data transmission begins after clearing COUNT. The real-time clock data are transmitted first, followed by the data from the 16 analog channels. Next, the depth is sampled and transmitted.

The LOG routine terminates with the execution of the OUT 5 instruction, which causes an INTERNAL RESET. At the start of the next five-second cycle, the control module puts the CPU back into the RUN mode ($\overline{\text{EXT CLEAR}}$ =1) and execution starts again at location 0000. This cycle repeats as long as the data logger is in the LOG mode.

MONITOR ROUTINE

The MONITOR routine is used to start the execution of a program at any address, load data into RAM, and inspect any memory location. This routine is typically used to enter and execute test programs.

Interrupts are enabled and COUNT is initialized prior to execution of the MONITOR routine (see Fig.12). The MONITOR

routine can be re-entered by pressing the RESET button (while the data logger is still in the MONITOR mode).

INITIALIZATION

Initialization takes place after each RESET, which occurs every five seconds. The program resets itself after each LOG cycle with an INTERNAL RESET. If the program fails to assert this INTERNAL RESET due to an error in execution, the $\overline{CLR}$ signal on the control module will force the microcomputer into the RESET mode at the start of the next five-second cycle. This "guaranteed" RESET followed by initialization prevents the microcomputer from being "hung" in a loop or halted indefinitely.

Not all variables can be initialized in the MONITOR mode, however, the real-time clock must be set in the MONITOR mode.

The software makes use of the Standard Call and Return Technique (SCRT) for handling subroutines (see RCA CDP1802 Microprocessor Manual).

The CPU registers are used as memory pointers. Register R(1) contains the address of the interrupt-service routine. Register R(2) points to a free memory area for use as a stack. Register R(3) is used as the program counter for the main program and subroutines. Registers R(7) and R(8), and R(9) point to scratch-pad storage locations used by the MONITOR program.

INTERRUPTS

The interrupt-service routine is used to advance the real-time clock. The control module issues an interrupt to the CPU every five seconds. The interrupt routine updates the clock by adding five seconds to a count of hours, minutes, and seconds

in memory. These numbers are stored in hexadecimal form. The numbers are converted to decimal for display on the front panel for transmission over the serial interface.

The clock functions in both the MONITOR and LOG modes. The clock can be set in the MONITOR mode with a utility program (see operating instructions in Appendix).

Interrupts are disabled during initialization, because part of the initialization involves setting up the interrupt-service routine.

SERIAL OUTPUT PROGRAMMING

Data gathered by the data logger are transmitted to the tape recorder via the RS-232 serial interface. The Q output flag of the microprocessor is switched to either of two stable states at a given time, under program control, to provide the appropriate ASCII-coded data. A programmed delay in the software determines the transmission rate. Any rate up to 1200 BAUD is possible with the present software.

OBTAINING DEPTH DATA

The depth module converts serial data from the fathometer to a 16-bit parallel word. The word is read one byte (8 bits) at a time. Because the depth sounder and data logger operate asynchronously, the logger may read the 16 bits as they are being shifted into the register, resulting in an erroneous reading. This problem is overcome by performing multiple reads. The data logger reads the two bytes, delays for a short time, and then reads the two bytes again. If these two pairs of bytes do not form the same 16-bit word, two more bytes are read until two

consecutive pairs of bytes form the same word. When two consecutive pairs of bytes compare, the 16-bit word formed by either pair is taken as a valid reading.

RECOMMENDATIONS

We have constructed a more-compact battery powered version of the data logger, which consumes much less power. This allows the logger to be used in remote self-contained units, such as bouys. The keypad and display circuits, which are not necessary in the logging mode and the LED (light emitting diode) displays, which are the major power consumers, are now in a separated module that is connected to the CPU module through the 44-pin bus only when the system is started up or when testing in the monitor mode. An integrated circuit voltage inverter (ICL7660CPA, Intersil, Cupertino, CA.) provides the negative power to the operational amplifiers, allowing the system to be operated from a single 12VDC power source. Hardware modifications that lower the power consumption are: 1) the logic +5 volt power is strobed (off) to the CPU module (except to the 256 bytes of RAM; Fig.33) and 2) a lower-power analog to digital converter with an extremely low quiescent power consumption in the interrupt power mode (ADC-HC12B, Datel Inc., Mansfield, MA.) replaces the model ADC-HX12B.

ACKNOWLEDGEMENT

Special thanks is given to Laurence E. Schemel for assisting in the preparation and review of this report.

REFERENCES

Dedini, L. A., and Schemel, L. E., 1980, A continuous water sampling and multiparameter-measurement system for estuaries: an improved system for small vessels: U.S. Geological Survey Open-File Report 80-1293. 49 p.

Leap, K.J., 1980, The design of a microprocessor-based wind-prospecting system: Master's thesis, Department of Electrical Engineering, San Jose State University, CA. 90 p.

RCA Corporation, 1977. User manual for the CDP 1802 COSMAC Microprocessor. U.S.A. 117 p.

Schemel, L. E., and Dedini L. A., 1979, A continuous water-sampling and multiparameter system for estuaries: U.S. Geological Survey Open-File Report 79-273. 92 p.

APPENDIX

TABLES

FIGURES

- LOGGING DATA (In Present Configuration) -

1. Set MODE switch to MONITOR.
2. Wait for "03 00 30" to appear on the panel display. This is the starting address and data of the program SET, which is for setting the real-time clock.
3. Press the "RUN" switch.
4. Press the "H" key for fast advance. Press the "L" key for slow advance. Press the "SHIFT" key for single (5 second) advance.
5. Switch to the LOG mode to start logging data.
6. Select scan interval by hexadecimal switch HS2. Scan interval = HS2 setting x 5 seconds.

- ENTERING AND EXECUTING TEST PROGRAMS -

1. Set MODE switch to MONITOR.
2. Select the high address (HI ADDR) by pressing any two keys (0-F) on the keypad. While pressing the "SHIFT" key down, press and release the "H" key.
3. Select the low address (LO ADDR) by

pressing any two keys (0-F) on the keypad. While pressing the "SHIFT" key down, press and release the "L" key.

4. To increment the address by one, press the "L" key. The corresponding data will appear in the SEC/DATA display.
5. To store the data on the keypad register (keypad display), press the "H" key. The data will be stored at the current address, and the address will be incremented by one.
6. To execute a program, select the programs starting address (HI ADDR) and (LO ADDR), and press the "RUN" switch.

GLOSSARY

Bus - A number of conductors used for transmitting signals or power from sources to destinations.

Byte - One piece of 8-bit parallel binary information.

Control bus - Control lines for special signals generated and sent out by the microprocessor.

CPU - (Central Processing Unit): That part of a microcomputer that controls the interpretation and execution of instructions.

Data bus - Eight separate lines connected to the 8-bit microprocessor which can transfer data to or from memory and input/output (I/O) ports.

EPROM - A form of ROM which is erasable and electrically reprogramable.

I/O port - A connection to a CPU which is configured (or programmed) to provide an input or output data path between the CPU and external devices, such as keypad, display, etc.

Memory-address lines - Input lines (A0-A7) to memory integrated-circuit chips from the microprocessor, that are used to select the memory word that is being written or read.

RAM - Random-access read/write memory integrated-circuit chip. Data stored in RAM is lost when power is turned off.

Registers - A fast-access circuit used to store words in a microprocessor.

ROM - Read-only memory, where software instructions are permanently contained during manufacturing of the integrated-circuit chip.

Table 7

List of Programs and Subroutines

Address HI	Address LO	Name	Description
00	00-39	RESET	Start of initialization.
00	40-85	MON	Monitor program.
01	00-0E	LOG	Calling program for LOG mode.
02	00-3B		Hexadecimal to decimal look-up table.
03	00-50	SET	Subroutine to set the real-time clock.
04	00-1C	ENTRY	Routine to initialize registers R1, R2, R3, R4, and R5 for SCRT.
04	27-58	B1200	Subroutine for the serial transmission of the byte in M(R(7)).
04	60-69	WAIT	Subroutine to wait for INTF = 1 (wait for interrupt).
04	70-7D	CLOCK	Subroutine to display the real-time clock.
04	90-BB	SAMPLE	Subroutine to sample the 16 analog channels and store in RAM.
04	D0-D7	DELAY1	80 microsecond delay.
04	D8-DF	DELAY2	7 millisecond delay.
04	E0-F7	ANIBBLE	Subroutine to convert the low 4 bits of M(R(7)) to ASCII and send.
05	00-0F	ABYTE	Subroutine to convert the two hexadecimal digits in M(R(7)) to ASCII and send.
05	18-24	SP	Subroutine to send ASCII space.
05	25-37	CRLF	Subroutine to send ASCII carriage return and line feed.
05	40-57	DISPLAY	Subroutine to display the channel selected by the keypad.
05	60-71	TXCLOCK	Subroutine to send the real-time clock data.
05	78-9B	TXDATA	Subroutine to send the data stored by SAMPLE.
05	B0-CE	TRANS	Subroutine to transmit data if COUNT = SCAN.
06	00-30	TXDEPTH	Subroutine to sample depth and send.
06	40-5D	CC	Subroutine to convert hexadecimal clock data to decimal.
07	00-2E	INTRPT	Interrupt-service routine. (code starts at 0700, entry point is 0701)
07	B0-DE	PDUMP	Program to dump the contents of the EPROM.

Table 8

Map of RAM Storage

Address HI LO	Name	Description
8C 00		Beginning address of RAM.
8C A0	LDAT(0)	Low 8 bits (D7-D0) of channel 0.
8C A1	LDAT(1)	Low 8 bits of channel 1.
.	.	.
.	.	.
.	.	.
8C AF	LDAT(F)	Low 8 bits of channel F (15).
8C B0	HDAT(0)	High 4 bits (D11-D8) of channel 0.
8C B1	HDAT(1)	High 4 bits of channel 1.
.	.	.
.	.	.
.	.	.
8C BF	HDAT(F)	High 4 bits of channel F (15).
8C C0	INTF	Flag set by interrupt-service routine.
8C C1	COUNT	Counts number of samples taken.
8C F0	BYTE0	Low byte from depth sounder.
8C F1	BYTE1	High byte from depth sounder.
8C FA	T0	Hexadecimal hours.
8C FB	T1	Hexadecimal minutes.
8C FC	T2	Hexadecimal seconds.
8C FD	HR	Decimal hours.
8C FE	MIN	Decimal minutes.
8C FF	SEC	Decimal seconds.
8C FF		Ending address of RAM.

Table 9

Table of I/O Ports

Instruction	Location of I/O port	Action
OUT 1	display module	D7-D0 → HR/HI ADDR display.
OUT 2	display module	D7-D0 → MIN/LO ADDR display.
OUT 3	display module	D7-D0 → SEC/DATA display.
OUT 4	display module	D7-D0 → low 8 bits of DATA display.
OUT 5	display module	D3-D0 → high 4 bits of DATA display.
	multiplexer module	D7-D4 → multiplexer address latch.
OUT 6	control module	trigger INTERNAL RESET.
OUT 7	A/D module	trigger A/D conversion.
INP 1	switch input module	encoded switch position (RS1) → D3-D0.
		encoded switch position (RS2) → D7-D4.
INP 2	control module	mode switch position → D0.
INP 3	keypad module	keypad shift register → D7-D0.
INP 4	A/D module	4 high bits of A/D conversion → D3-D0.
INP 5	A/D module	8 low bits of A/D conversion → D7-D0.
INP 6	switch input module	hexadecimal switch (HS1) → D7-D0.
INP 7	switch input module	hexadecimal switch (HS2) → D7-D0.

PROGRAM LISTING

```
                        ..INITIALIZATION ROUTINE
                        ..EXECUTED AFTER A RESET
                        ..
00 00 7100      RESET:  #71         ..DISABLE INTERRUPTS.
   02 C00400            LBR ENTRY   ..SET UP SCRT.
   05 F88C      START:  LDI #8C     ..SET UP POINTERS
   07 B7                PHI 7       ..FOR SCRATCH PAD
   08 B8                PHI 8       ..STORAGE.
   09 C4                NOP
   0A F8F7              LDI #F7
   0C A7                PLO 7
   0D F8F8              LDI #F8
   0F A8                PLO 8
   10 C4C4C4            NOP;NOP;NOP
   13 F803              LDI #03     ..POINT TO "SET"
   15 B9                PHI 9       ..ROUTINE TO SET
   16 F800              LDI #00     ..REAL-TIME CLOCK.
   18 A9                PLO 9
   19 E7                SEX 7       ..READ FUNCTION SWITCH.
   1A 6A                INP 2
   1B F6                SHR         ..BRANCH TO LG IF DF IS 1.
   1C 332D              BDF LG
   1E F88C              LDI #8C     ..ELSE...
   20 B0                PHI 0
   21 F8C1              LDI #C1
   23 A0                PLO 0
   24 F800              LDI #00
   26 50                STR 0       ..PUT 0 IN COUNT.
   27 E37003            #E37003     ..ENABLE INTERRUPTS.
   2A C00040            LBR MON     ..BRANCH TO MONITOR.
   2D F88C      LG:     LDI #8C
   2F B0                PHI 0
   30 F8C0              LDI #C0
   32 A0                PLO 0
   33 F800              LDI #00
   35 50                STR 0       ..PUT 0 IN INTF.
   36 E37003            #E37003     ..ENABLE INTERRUPTS.
   39 C00100            LBR LOG     ..BRANCH TO LOG ROUTINE.
                        ..
                        ..MONITOR ROUTINE
                        ..
                        ..THIS ROUTINE IS USED WITH THE HEX KEYPAD
                        ..TO INSPECT ANY MEMORY LOCATION, LOAD
                        ..DATA IN RAM, AND START THE EXECUTION OF
                        ..A PROGRAM AT ANY LOCATION.
                        ..
                        ..KEY FUNCTIONS:
                        ..  H  INSERT BYTE AT PRESENT MEMORY LOCATION
                        ..     AND ADVANCE.
                        ..  L  ADVANCE.
                        ..  H(SHIFTED) LOAD HI ADDRESS BYTE.
                        ..  L(SHIFTED) LOAD LO ADDRESS BYTE.
                        ..  S  SHIFT KEY.
                        ..  RUN  START EXECUTION AT PRESENT MEMORY
                        ..       LOCATION.
                        ..  RESET  RESET MICROCOMPUTER.
```

```
00 40 E7        MON:    SEX 7       ..READ KEYPAD
   41 6B                INP 3       ..VIA M(R(7)).
   42 E9                SEX 9       .. DISPLAY
   43 63                OUT 3       ..VIA M(R(9)).
   44 29                DEC 9
   45 89                GLO 9       ..DISPLAY
   46 58                STR 8       ..R(9).0
   47 E8                SEX 8       ..VIA
   48 62                OUT 2       ..M(R(8)).
   49 28                DEC 8
   4A 99                GHI 9       ..DISPLAY
   4B 58                STR 8       ..R(9).1
   4C 61                OUT 1       ..VIA M(R(8)).
   4D 28                DEC 8
   4E 375F              B4 SP       ..BRANCH IF "SHIFT" PRESSED.
   50 3C58      HP:     BN1 LP      ..BRANCH IF "H" NOT PRESSED.
   52 3452      HR:     B1 HR       ..BRANCH IF "H" PRESSED.
   54 07                LDN 7       ..MOVE M(R(7))
   55 59                STR 9       ..TO M(R(9)).
   56 305C              BR INC
   58 3D5F      LP:     BN2 SP      ..BRANCH IF "L" NOT PRESSED.
   5A 355A      LR:     B2 LR       ..BRANCH IF "L" PRESSED.
   5C 19        INC:    INC 9       ..INCREMENT ADDRESS.
   5D 3040              BR MON      ..READ KEYPAD AGAIN.
   5F 3F71      SP:     BN4 RP      ..BRANCH IF "SHIFT"
                                    ..NOT PRESSED.
   61 3C69      HP2:    BN1 LP2     ..BRANCH IF "H" NOT PRESSED.
   63 3463      HR2:    B1 HR2      ..BRANCH IF "H" PRESSED.
   65 07                LDN 7       ..MOVE M(R(7))
   66 B9                PHI 9       ..TO R(9).1
   67 3040              BR MON      ..READ KEYPAD AGAIN.
   69 3D71      LP2:    BN2 RP      ..BRANCH IF "L" NOT PRESSED.
   6B 356B      LR2:    B2 LR2      ..BRANCH IF "L" PRESSED.
   6D 07                LDN 7       ..MOVE M(R(7))
   6E A9                PLO 9       ..TO R(9).0
   6F 3040              BR MON      ..READ KEYPAD AGAIN.
   71 3E        RP:     BN3 MON     ..BRANCH IF "RUN"
                                    ..NOT PRESSED.
   73 F810              LDI #10     ..START DELAY...
   75 B0                PHI 0
   76 10        LOC1:   INC 0
   77 90                GHI 0
   78 3A76              BNZ LOC1    ..END DELAY.
   7A F800              LDI # 00    ..BRANCH TO M(R(9))
   7C B0                PHI 0       ..WITH PC = R(3)...
   7D F881              LDI # 81
   7F A0                PLO 0
   80 D0                SEP 0
   81 99        T:      GHI 9
   82 B3                PHI 3
   83 89                GLO 9
   84 A3                PLO 3
   85 D3                SEP 3       ..EXECUTE PROGRAM.
```

```
                    ;;
                    ;;MAIN CALLING ROUTINE FOR THE LOG MODE.
                    ;;
01 00 D40460  LOG:      SEP 4;WAIT    ;;CALL WAIT.
   03 D40470            SEP 4;CLOCK   ;;CALL CLOCK.
   06 D40490            SEP 4;SAMPLE  ;;CALL SAMPLE.
   09 D40540            SEP 4;DISPLAY ;;CALL DISPLAY.
   0C D405B0            SEP 4;TRANS   ;;CALL TRANS.
                    ;;
                    ;;ROUTINE TO SET REAL-TIME CLOCK.
                    ;;
03 00 302B    SET:      BR DIS        ;;GO DISPLAY CLOCK.
                                      ;;
   02 F88CBA  ADV:      LDI #8C;PHI A ;;ADVANCE CLOCK
   05 F8FCAA            LDI #FC;PLO A ;;(SEE INTRPT)
   08 0A                LDN A
   09 FC05              ADI #05
   0B 5A                STR A
   0C FF3C              SMI #3C
   0E 3B2B              BM DIS
   10 F8005A            LDI #00;STR A
   13 2A                DEC A
   14 0A                LDN A
   15 FC01              ADI #01
   17 5A                STR A
   18 FF3C              SMI #3C
   1A 3B2B              BM DIS
   1C F8005A            LDI #00;STR A
   1F 2A                DEC A
   20 0A                LDN A
   21 FC01              ADI #01
   23 5A                STR A
   24 FF18              SMI #18
   26 3B2B              BM DIS
   28 F8005A            LDI #00;STR A ;;END ADVANCE CLOCK.
                                      ;;
   2B D40470  DIS:      SEP 4; CLOCK  ;;DISPLAY CLOCK
   2E 3C35              BN1 DELAY     ;;BRANCH IF "H" NOT PRESSED.
   30 E37153            #E37153       ;;DISABLE INTERRUPTS.
   33 3002              BR ADV        ;;ADVANCE CLOCK - NO DELAY.
   35 3D43    DELAY:    BN2 SNGL      ;;BRANCH IF "L" NOT PRESSED.
   37 E37153            #E37153       ;;DISABLE INTERRUPTS.
   3A F8F8BF            LDI #F8;PHI F       ;;DELAY
   3D 1F9F3A3D LOC1:    INC F;GHI F;BNZ LOC1  ;;DELAY
   41 3002              BR ADV        ;;GO ADVANCE CLOCK.
   43 3F4C    SNGL:     BN4 EN        ;;BRANCH IF "SHIFT"
                                      ;;NOT PRESSED.
   45 E37153            #E37153       ;;DISABLE INTERRUPTS.
   48 3748    REL:      B4 REL        ;;WAIT FOR "SHIFT"
                                      ;;RELEASED.
   4A 3002              BR ADV        ;;GO ADVANCE CLOCK.
   4C E37023  EN:       #E37023       ;;ENABLE INTERRUPTS.
   4F 302B              BR DIS        ;;GO DISPLAY CLOCK.
```

```
                    ..SUBROUTINE TO INITIALIZE
                    ..R(2),R(3),R(4),R(5)
                    ..FOR STANDARD CALL AND RETURN TECHNIQUE
                    ..(SCRT)
                    ..
04 00 F88C  ENTRY:  LDI #8C
   02 B2            PHI 2
   03 F8EF          LDI #EF
   05 A2            PLO 2       ..STACK AT 8CEF.
   06 F800          LDI #00     ..A(START).1
   08 B3            PHI 3
   09 F805          LDI #05     ..A(START).0
   0B A3            PLO 3       ..R(3) IS PROGRAM COUNTER.
   0C F8C3          LDI #C3
   0E B4            PHI 4
   0F B5            PHI R5
   10 F85E          LDI #5E     ..CALL ROUTINE AT C35E.
   12 A4            PLO 4
   13 F86E          LDI #6E     ..RETURN ROUTINE AT C36E.
   15 A5            PLO 5
   16 F807          LDI #07     ..INTERRUPT ROUTINE
   18 B1            PHI 1       ..STARTS AT
   19 F801          LDI #01     ..0701.
   1B A1            PLO R1
   1C D3            SEP 3       ..GO TO START.
```

```
                    ..SUBROUTINE B1200
                    ..
                    ..THIS SUBROUTINE ACCOMPLISHES THE
                    ..SERIAL TRANSMISSION OF THE BYTE
                    ..IN M(R(7)) BY "TOGGLING" THE
                    ..OUTPUT FLAG Q.  TRANSMISSION IS
                    ..AT 1200 BAUD.  ONE START BIT, 8
                    ..INFORMATION BITS, AND 2 STOP BITS
                    ..ARE SENT.
                    ..
04 27 F88CB8B9 B1200:   LDI #8C;PHI 8;PHI 9 ..SET UP
   28 F8F8A8            LDI #F8;PLO 8       ..SCRATCH
   2E F8F9A9            LDI #F9;PLO 9       ..STORAGE.
   31 7B                SEQ           ..START BIT.
   32 F800              LDI #00       ..CLEAR
   34 58                STR 8         ..BITSUM   M(R(8)).
   35 08        INC:    LDN 8         ..INCRE-
   36 FC01              ADI #01       ..MENT
   38 58                STR 8         ..BITSUM.
   39 F8FF              LDI #FF       ..HIGH TIMING BYTE.
   3B BC                PHI C
   3C F8E7              LDI #E7       ..LOW TIMING BYTE.
   3E AC                PLO C         ..(E7 FOR 1200 BAUD,
   3F 1C        LOC1:   INC C         .. 8B FOR 300 BAUD).
   40 9C                GHI C         ..DELAY
   41 3A3F              BNZ LOC1      ..DELAY.
   43 08                LDN 8         ..SUBRACT 11
   44 FD0B              SDI #0B       ..FROM BITSUM AND TEST.
   46 3A49              BNZ CONT      ..CONTINUE IF BITSUM IS NOT
                                      ..EQUAL TO 11.
   48 D5                SEP 5         ..RETURN.
   49 08        CONT:   LDN 8         ..SUBTRACT 9 FROM
   4A FF09              SMI #09       ..BITSUM AND TEST.
   4C 3356              BGE RQ        ..BRANCH IF BITSUM .GE. 9.
   4E 07                LDN 7         ..SHIFT
   4F 76                SHRC          ..M(R(7))
   50 57                STR 7         ..RIGHT.
   51 3356              BDF RQ        ..BR IF DF .EQ. 1.
   53 7B                SEQ           ..SEND "0"
   54 3035              BR INC        ..BRANCH.
   56 7A        RQ:     REQ           ..SEND "1"
   57 3035              BR INC        ..BRANCH.
```

```
                    ..SUBROUTINE TO WAIT UNTIL INTF EQUALS 1.
                    ..
04 60 F88CBA   WAIT:    LDI #8C;PHI A  ..POINT TO
   63 F8C0AA            LDI #C0;PLO A  ..INTF  M(8CC0).
   66 0A        LOOK:   LDN A
   67 3266              BZ LOOK        ..BRANCH UNTIL
                                       ..INTF IS NONZERO.
   69 D5                SEP 5          ..RETURN.
                    ..
                    ..SUBROUTINE TO DISPLAY CLOCK.
                    ..
04 70 D40640   CLOCK:   SEP 4;CC    ..CALL CC TO CONVERT
                                    ..TO DECIMAL.
   73 F88CBA            LDI #8C;PHI A  ..POINT TO
   76 F8FDAA            LDI #FD;PLO A  ..HOURS.
   79 EA                SEX A
   7A 616263            OUT 1;OUT 2;OUT 3  ..OUTPUT
   7D 2A2A2A            DEC A;DEC A;DEC A  ..HRS,MIN,SEC.
   80 D5                SEP 5          ..RETURN.
                    ..
                    ..SUBROUTINE TO SAMPLE 16 CHANNELS
                    ..AND STORE IN RAM.
                    ..
04 90 F88C     SAMPLE:  LDI #8C                ..SET
   92 BABBB7            PHI A;PHI B;PHI 7      ..UP
   95 F8A0AA            LDI #A0;PLO A          ..MEMORY
   98 F8B0AB            LDI #B0;PLO B          ..DATA
   9B F8F7A7            LDI #F7;PLO 7          ..POINTERS.
   9E 8A        MUX:    GLO A                  ..OUTPUT
   9F FEFEFEFE          SHL;SHL;SHL;SHL        ..MUX
   A3 57                STR 7                  ..ADDRESS
   A4 E7                SEX 7                  ..TO
   A5 6527              OUT 5;DEC 7            ..PORT 5.
   A7 D404D8            SEP 4;DELAY2   ..DELAY 7 MILLISEC.
   AA E7                SEX 7
   AB 6727              OUT 7;DEC 7    ..START A/D CONV.
   AD D404D0            SEP 4;DELAY1   ..DELAY 80 MICROSEC.
   B0 EB                SEX B
   B1 6C                INP 4          ..GET HI 4 BITS.
   B2 1B                INC B          ..BUMP POINTER.
   B3 EA                SEX A
   B4 6D                INP 5          ..GET LO 8 BITS.
   B5 1A                INC A          ..BUMP POINTER.
   B6 8A                GLO A          ..CONTINUE UNTIL
   B7 FA0F              ANI #0F        ..ALL 16 CHANNELS HAVE
   B9 3A9E              BNZ MUX        ..BEEN SAMPLED.
   BB D5                SEP 5          ..RETURN.
```

```
                        ..
                        .80 MICROSECOND DELAY.
                        ..
04 D0 F8FE      DELAY1: LDI #FE
   D2 AD                PLO D
   D3 1D        LOC1:   INC D
   D4 8D                GLO D
   D5 3AD3              BNZ LOC1
   D7 D5                SEP 5
                        ..
                        ..7 MILLISECOND DELAY.
                        ..
04 D8 F800      DELAY2: LDI #00
   DA AD                PLO D
   DB 1D        LOC2:   INC D
   DC 8D                GLO D
   DD 3ADB              BNZ LOC2
   DF D5                SEP 5
                        ..
                        ..CONVERT LOW 4 BITS OF M(R(7))
                        ..TO ASCII AND TRANSMIT.
                        ..
04 E0 07        ANIBBLE: LDN 7       ..SAVE
   E1 AA                 PLO A       ..BYTE.
   E2 FA0F               ANI #0F     ..MASK OUT 4 HI BITS.
   E4 BA                 PHI         ..SAVE MASKED BYTE.
   E5 FF0A               SMI #0A     ..BRANCH IF NIBBLE
   E7 33EE               BGE ALPHA   .. IS .GE. TO 0A.
   E9 9A                 GHI A       ..ADD
   EA FC30               ADI #30     ..30.
   EC 30F1               BR SEND     ..GO SEND CHARACTER.
   EE 9A        ALPHA:   GHI A       ..ADD
   EF FC37               ADI #37     ..37.
   F1 57        SEND:    STR 7       ..TRANSMIT
   F2 D40427             SEP 4;B1200..CHARACTER.
   F5 8A                 GLO A       ..RESTORE
   F6 57                 STR 7       ..M(R(7)).
   F7 D5                 SEP 5       ..RETURN.
                        ..
                        ..CONVERT TWO HEXADECIMAL DIGITS IN
                        ..M(R(7)) TO ASCII AND TRANSMIT.
                        ..
05 00 07        ABYTE:   LDN 7       ..SAVE
   01 AB                 PLO B       ..BYTE.
   02 F6F6F6F6           SHR;SHR;SHR;SHR ..GET LEFT NIBBLE.
   06 57                 STR 7
   07 D404E0             SEP 4;ANIBBLE   ..SEND LEFT NIBBLE.
   0A 8B                 GLO B
   0B 57                 STR R7
   0C D404E0             SEP 4;ANIBBLE   ..SEND RIGHT NIBBLE.
   0F D5                 SEP 5       ..RETURN.
```

```
                        ;;
                        ;;SUBROUTINE TO SEND ASCII SP (SPACE).
                        ;;
05 18 F88CB7    SP:     LDI #8C;PHI 7   ;;STORE
   1B F8F7A7            LDI #F7;PLO 7   ;;20 IN
   1E F82057            LDI #20;STR 7   ;;M(R(7)).
   21 D40427            SEP 4;B1200     ;;CALL B1200.
   24 D5                SEP 5           ;;RETURN.
                        ;;
                        ;;SUBROUTINE TO SEND ASCII CR AND LF
                        ;;(CARRIAGE REUTRN AND LINE FEED).
                        ;;
05 25 F88CB7    CRLF:   LDI #8C;PHI 7   ;;STORE
   28 F8F7A7            LDI #F7;PLO 7   ;;OD IN
   2B F80D57            LDI #0D;STR 7   ;;M(R(7)).
   2E D40431            SEP 4;B1200     ;;CALL B1200.
   31 F80A57            LDI #0A;STR 7   ;;STORE 0A IN M(R(7)).
   34 D40427            SEP 4;B1200     ;;CALL B1200.
   37 D5                SEP 5           ;;RETURN.
                        ;;
                        ;;SUBROUTINE TO DISPLAY CHANNEL SELECTED BY
                        ;;LEAST SIGNIFICANT HEX DIGIT OF KEYPAD.
                        ;;
05 40 F88C      DISPLAY: LDI #8C                ;;SET UP
   42 B7B8B9            PHI 7;PHI 8;PHI 9       ;;MEMORY DATA
   45 F8F7A7            LSI #F7;PLO 7           ;;POINTERS.
   48 E7                SEX 7
   49 6B                INP 3           ;;READ KEYPAD.
   4A 07                LDN 7           ;;MASK OUT
   4B FA0F              ANI #0F         ;;HIGH NIBBLE.
   4D FCA0              ADI #A0         ;;HI NIBBLE IS A.
   4F A8                PLO 8           ;;R(8)   POINTS TO LOW 8
                                        ;;BITS OF A/D CONVERSION.
   50 FC10              ADI #10         ;;HI NIBBLE IS B.
   52 A9                PLO 9           ;;R(9)   POINTS TO HI 4
                                        ;;BITS OF A/D CONVERSION.
   53 E9                SEX 9           ;;
   54 65                OUT 5           ;;DISPLAY HI 4 BITS.
   55 E8                SEX 8           ;;
   56 64                OUT 4           ;;DISPLAY LO 8 BITS.
   57 D5                SEP 5           ;;RETURN.
                        ;;
                        ;;SUBROUTINE TO SEND CLOCK DATA.
                        ;;
05 60 D40640    TXCLOCK: SEP 4;CC       ;;CONVERT TO DECIMAL.
   63 F88CB7            LDI #8C;PHI 7   ;;POINT TO
   66 F8FDA7            LDI #FD;PLO 7   ;;HOURS.
   69 D40500            SEP 4;ABYTE     ;;SEND HOURS.
   6C 17                INC 7           ;;POINT TO MINUTES.
   6D D40500            SEP 4;ABYTE     ;;SEND MINUTES.
   70 17                INC 7           ;;POINT TO SECONDS.
   71 D40500            SEP 4;ABYTE     ;;SEND SECONDS.
   74 D5                SEP 5           ;;RETURN.
```

```
                        ..
                        ..SUBROUTINE TO TRANSMIT DATA STORED
                        ..BY SAMPLE.
                        ..
05 78 F88CB7    TXDATA: LDI #8C;PHI 7
   7B F800AE            LDI #00;PLO E
   7E D40518    LOOP:   SEP 4;SP    ..SEND SPACE.
   81 8E                GLO E       ..LOW POINTER
   82 FCB0              ADI #B0     ..IS B0 PLUS R(E).0.
   84 A7                PLO 7
   85 D404E0            SEP 4;ANIBBLE  ..SEND LO NIBBLE.
   88 C4C4C4            NOP;NOP;NOP;NOP
   8B 8E                GLO E       ..LOW POINTER
   8C FCA0              ADI #A0     ..IS A0 PLUS R(E).0.
   8E A7                PLO 7
   8F D40500            SEP 4;ABYTE    ..SEND BYTE.
   92 1E                INC E       ..INC LOW ADDRESS.
   93 8E                GLO E       ..TEST LOW NIBBLE
   94 FA0F              ANI #0F     ..OF LOW ADDR.
   96 3A7E              BNZ LOOP    ..LOOP UNTIL DONE.
   98 D40518            SEP 4;CRLF  ..SEND CR LF.
   9B D5                SEP 5       ..RETURN.
                        ..
                        ..SUBROUTINE TO TRANSMIT DATA IF
                        ..COUNT EQUALS SCAN.
                        ..COUNT IS INCREMENTED EVERY 5 SECONDS.
                        ..SCAN IS READ FROM A FRONT PANEL SWITCH.
                        ..
05 B0 F88CB7B8  TRANS:  LDI #8C;PHI 7;PHI 8
   B4 F8C1A7            LDI #C1;PLO 7 ..COUNT IS IN M(R(7)).
   B7 F8F8A8            LDI #F8;PLO 8 ..SCAN IS IN M(R(8)).
   BA 07                LDN 7       ..INCREMENT
   BB FC01              ADI #01     ..COUNT.
   BD 57                STR 7
   BE E8                SEX 8
   BF 6F                INP 7       ..COMPARE
   C0 07                LDN 7       ..COUNT WITH
   C1 F7                SM          ..SCAN.
   C2 3BD0              BNF HALT    ..BR IF COUNT .LT. SCAN.
   C4 F800              LDI #00     ..CLEAR
   C6 57                STR 7       ..COUNT.
   C7 D40560            SEP 4;TXCLOCK  ..TRANSMIT CLOCK DATA.
   CA D40578            SEP 4;TXDATA   ..TRANSMIT DATA.
   CD D40600            SEP 4;TXDEPTH  ..TRANSMIT DEPTH.
   D0 66        HALT:   OUT 6       ..INTERNAL RESET.
   D1 00                HLT         ..HALT.
```

```
                    ;;
                    ;;SUBROUTINE TO TRANSMIT 2 BYTES OF
                    ;;DEPTH DATA FOLLOWED BY CR AND LF.
                    ;;
06 00 F8E0B8    TXDEPTH:  LDI #E0;PHI 8 ;;E000 IS MEMORY MAPPED
   03 F800A8              LDI #00;PLO 8 ;;ADDRESS FOR DEPTH DATA.
   06 F88CB7              LDI #8C;PHI 7 ;;8CF0 IS TEMP. STORAGE
   09 F8F0A7              LDI #F0;PLO 7 ;;FOR DEPTH DATA.
   0C 08                  LDN 8       ;;GET FIRST BYTE.
   0D 57                  STR 7       ;;STORE FIRST BYTE.
   0E 1718                INC 7;INC 8   ;;POINT TO SECOND BYTE.
   10 08                  LDN 8       ;;GET SECOND BYTE.
   11 57                  STR 7       ;;STORE SECOND BYTE.
   12 F8FCBDAD            LDI #FC;PHI D;PLO D ;;DELAY
   16 1D9D      LOC1:     INC D;GHI D         ;;DELAY
   18 3A16                BNZ LOC1            ;;DELAY.
   1A E7                  SEX 7
   1B 08                  LDN 8           ;;COMPARE M(R(8))
   1C F3                  XOR             ;;WITH M(R(7)).
   1D 3A00                BNZ TXDEPTH     ;;BR IF NOT EQUAL.
   1F 2728                DEC 7;DEC 8     ;;POINT TO FIRST BYTE.
   21 08                  LDN 8           ;;COMPARE M(R(8))
   22 F3                  XOR             ;;WITH M(R(7)).
   23 3A00                BNZ TXDEPTH     ;;BR IF NOT EQUAL.
   25 17                  INC 7       ;;POINT TO SECOND BYTE.
   26 D40500              SEP 4;ABYTE ;;SEND SECOND BYTE.
   29 27                  DEC 7       ;;POINT TO FIRST BYTE.
   2A D40500              SEP 4;ABYTE ;;SEND FIRST BYTE.
   2D D40525              SEP 4;CRLF  ;;SEND CR LF.
   30 D5                  SEP 5       ;;RETURN.
                    ;;
                    ;;SUBROUTINE TO CONVERT HOURS, MINUTES, AND
                    ;;SECONDS (STORED AT 8CFA, FB, AND FC) FROM
                    ;;HEXADECIMAL TO DECIMAL AND STORE DECIMAL
                    ;;RESULT AT 8CFD, FE, AND FF.
                    ;;A LOOK-UP TABLE IS USED FOR THE CONVERSION.
                    ;;
06 40 F88CBABB  CC:       LDI #8C;PHI A;PHI B
   44 F8FAAA              LDI #FA;PLO A  ;;R(A) POINTS TO HR(HEX).
   47 F8FDAB              LDI #FD;PLO B  ;;R(B) POINTS TO HR(DEC).
   4A F802BC              LDI #02;PHI C  ;;R(C) POINTS TO
   4D 0AAC                LDN A;PLO C    ;;LOOK-UP TABLE.
   4F 0C5B                LDN C;STR B    ;;STORE DEC. HR.
   51 1A1B                INC A;INC B    ;;POINT TO MIN.
   53 0AAC                LDN A;PLO C
   55 0C5B                LDN C;STR B    ;;STORE DEC. MIN.
   57 1A1B                INC A;INC B    ;;POINT TO SEC.
   59 0AAC                LDN A;PLO C
   5B 0C5B                LDN C;STR B    ;;STORE DEC. SEC.
   5D D5                  SEP 5      ;;RETURN.
```

```
                    ..
                    ..INTERRUPT SERVICE ROUTINE.
                    ..ADVANCE REAL*TIME CLOCK BY 5 SECONDS.
                    ..SET INTF FLAG.
                    ..
07 00 70            EXIT:     RET  ..RETURN POINT.
   01 22            INTRPT:   DEC 2       ..ENTRY POINT.
   02 78                      SAV         ..SAVE X,P.
   03 22                      DEC 2
   04 73                      STXD        ..SAVE D
   05 76                      SHRC
   06 73                      STXD        ..SAVE DF
   07 8A                      GLO A       ..SAVE R(A)
   08 73                      STXD
   09 9A                      GHI A
   0A 73                      STXD
                    ..
                    ..START INTERRUPT TASK.
                    ..
   0B F88CBA                  LDI #8C;PHI A  ..POINT TO
   0E F8FCAA                  LDI #FC;PLO A  ..SECONDS.
   11 0A                      LDN A       ..ADD 5
   12 FC05                    ADI #05     ..TO
   14 5A                      STR A       ..SECONDS.
   15 FF3C                    SMI #3C     ..BR IF SECONDS
   17 3B34                    BM DIS      ..LT. 60.
   19 F8005A                  LDI #00;STR A  ..CLEAR SECONDS.
   1C 2A                      DEC A       ..POINT TO MINUTES.
   1D 0A                      LDN A       ..ADD 1
   1E FC01                    ADI #01     ..TO
   20 5A                      STR A       ..MINUTES.
   21 FF3C                    SMI #3C     ..BR IF MINUTES
   23 3B34                    BM DIS      ..LT. 60.
   25 F8005A                  LDI #00;STR A  ..CLEAR MINUTES.
   28 2A                      DEC A       ..POINT TO HOURS.
   29 0A                      LDN A       ..ADD 1
   2A FC01                    ADI #01     ..TO
   2C 5A                      STR A       ..HOURS.
   2D FF18                    SMI #18     ..BR IF HOURS
   2F 3B34                    BM DIS      ..LT. 24.
   31 F8005A                  LDI #00;STR A  ..CLEAR HOURS.
   34 F8C0AA        DIS:      LDI #C0;PLO A  ..SET
   37 F801                    LDI #01        ..INTF
   39 5A                      STR A          ..FLAG.
                    ..RESTORE REGISTERS.
   3A 12                      INC 2       ..RESTORE R(A).
   3B 42                      LDA 2
   3C BA                      PHI A
   3D 42                      LDA 2
   3E AA                      PLO A
   3F 42                      LDA 2       ..RESTORE DF.
   40 FE                      SHL
   41 42                      LDA 2       ..RESTORE D.
   42 3000                    BR EXIT     ..GO RETURN.
```

```
                        ..
                        ..PROGRAM TO DUMP
                        ..CONTENTS OF EPROM.
                        ..
07 B0 E37153    PDUMP:  #E37153     ..DISABLE INTERRUPTS.
   B3 F800BDAD          LDI #00;PHI D;PLO D ..CLEAR ADDR.
   B7 F800AE    A:      LDI #00; PLO E  ..CLEAR WORD COUNT.
   BA 9D57              GHI D;STR 7     ..SEND
   BC D40500            SEP 4;ABYTE     ..HI ADDR.
   BF 8D57              GLO D;STR 7     ..SEND
   C1 D40500            SEP 4;ABYTE     ..LO ADDR.
   C4 D40518            SEP 4;SP        ..SEND SPACE
   C7 D40518    B:      SEP 4;SP        ..SEND SPACE.
   CA 0D57              LDA D;STR 7     ..GET MEMORY DATA.
   CC 1D                INC D           ..INCR. ADDR.
   CD D40500            SEP 4;ABYTE     ..TRANSMIT DATA.
   D0 1E                INC E           ..INCR. WORD COUNT.
   D1 8EFB10            GLO E;XRI #10   ..WORD COUNT .EQ. 16 ?
   D4 3AC7              BNZ B       ..IF NO, BR TO B.
   D6 D40525            SEP 4;CRLF  ..IF YES, END LINE.
   D9 9DFB08            GHI D;XRI #08   ..HI ADDR. .EQ. 08?
   DC 3AB7              BNZ A       ..IF NO, CONTINUE.
   DE 00                HLT         ..IF YES, HALT.
```

CIRCUIT DIAGRAMS

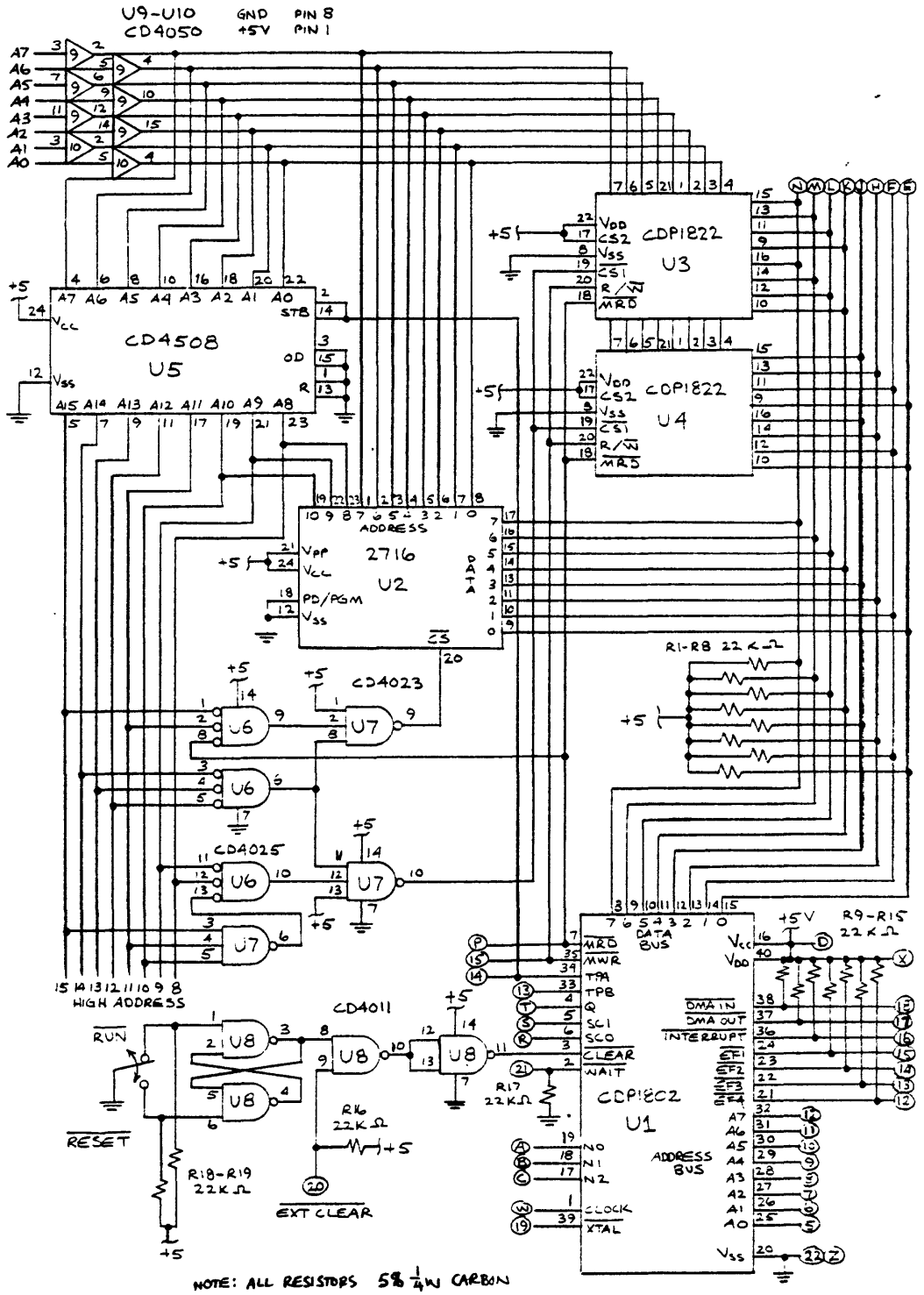


Figure 13

CPU Module Circuit Diagram

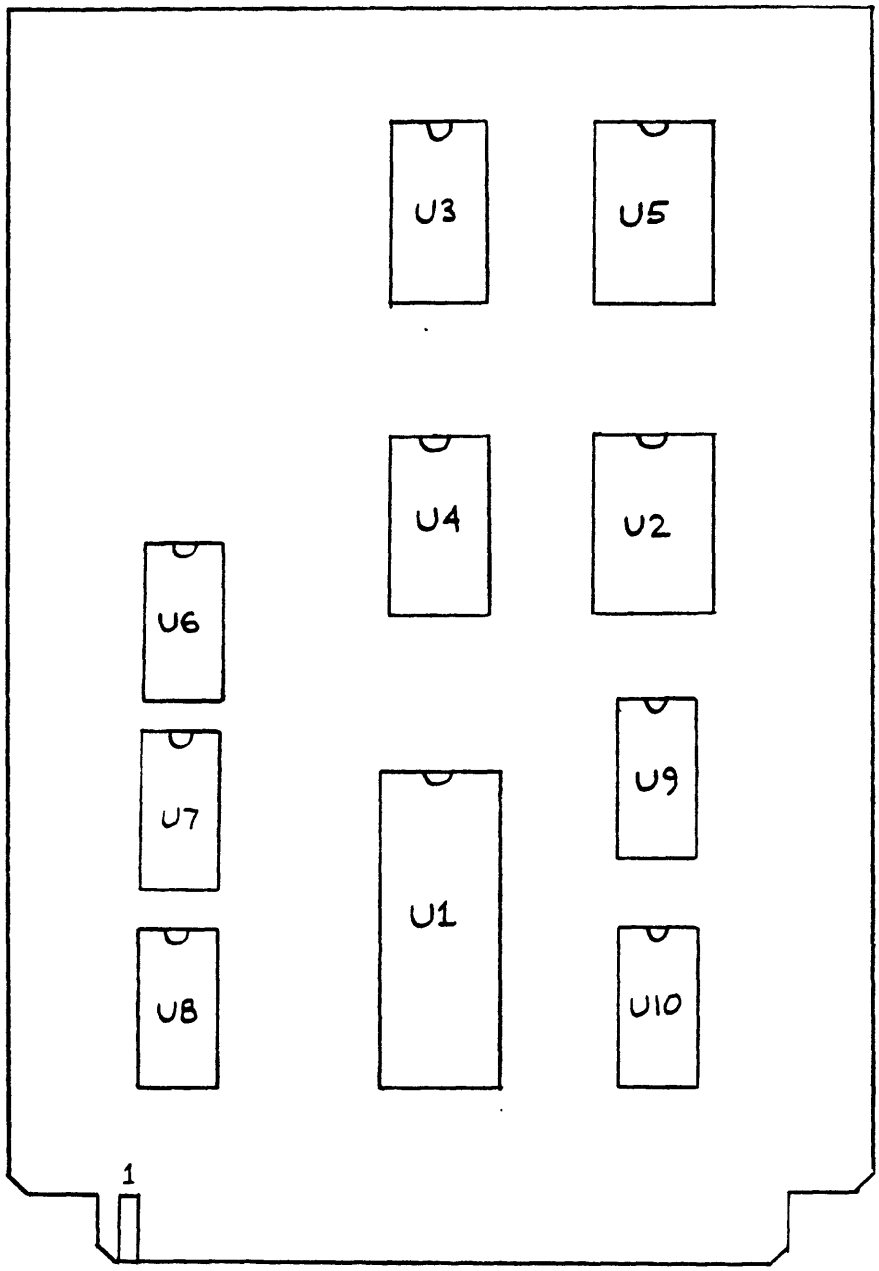
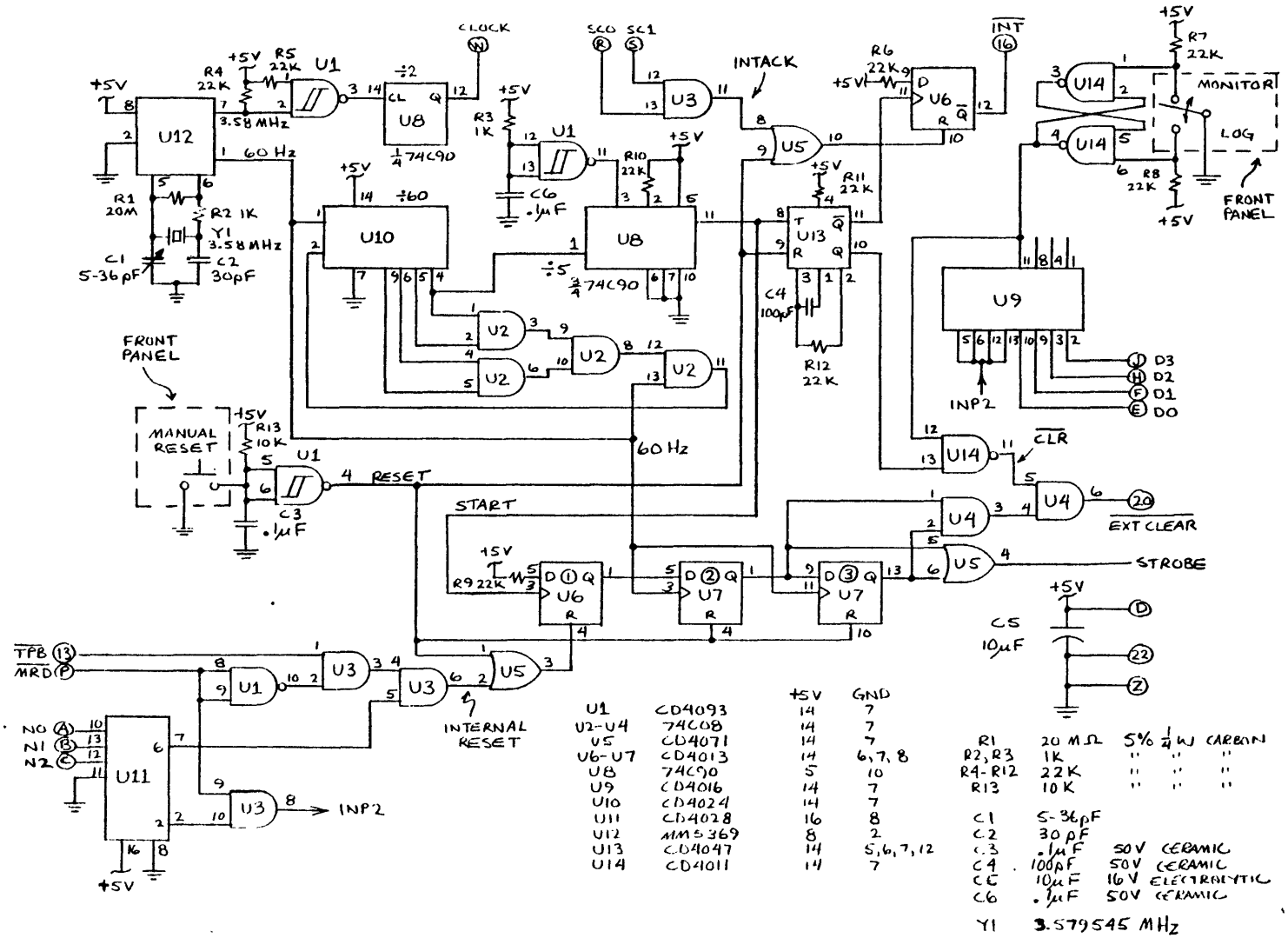


Figure 14

CPU Module IC Layout



		+5V	GND
U1	CD4093	14	7
U2-U4	74C08	14	7
U5	CD4071	14	7
U6-U7	CD4013	14	6, 7, 8
U8	74C90	5	10
U9	CD4016	14	7
U10	CD4024	14	7
U11	CD4028	16	8
U12	MM5369	8	2
U13	CD4047	14	5, 6, 7, 12
U14	CD4011	14	7

R1	20 MΩ	5% ¼ W CARBON
R2, R3	1K	" " "
R4-R12	22K	" " "
R13	10K	" " "
C1	5-36pF	
C2	30 pF	
C3	.1μF	50V CERAMIC
C4	100pF	50V CERAMIC
C5	10μF	16V ELECTROLYTIC
C6	.1μF	50V CERAMIC
Y1	3.579545 MHz	

Figure 15

Control Module Circuit Diagram

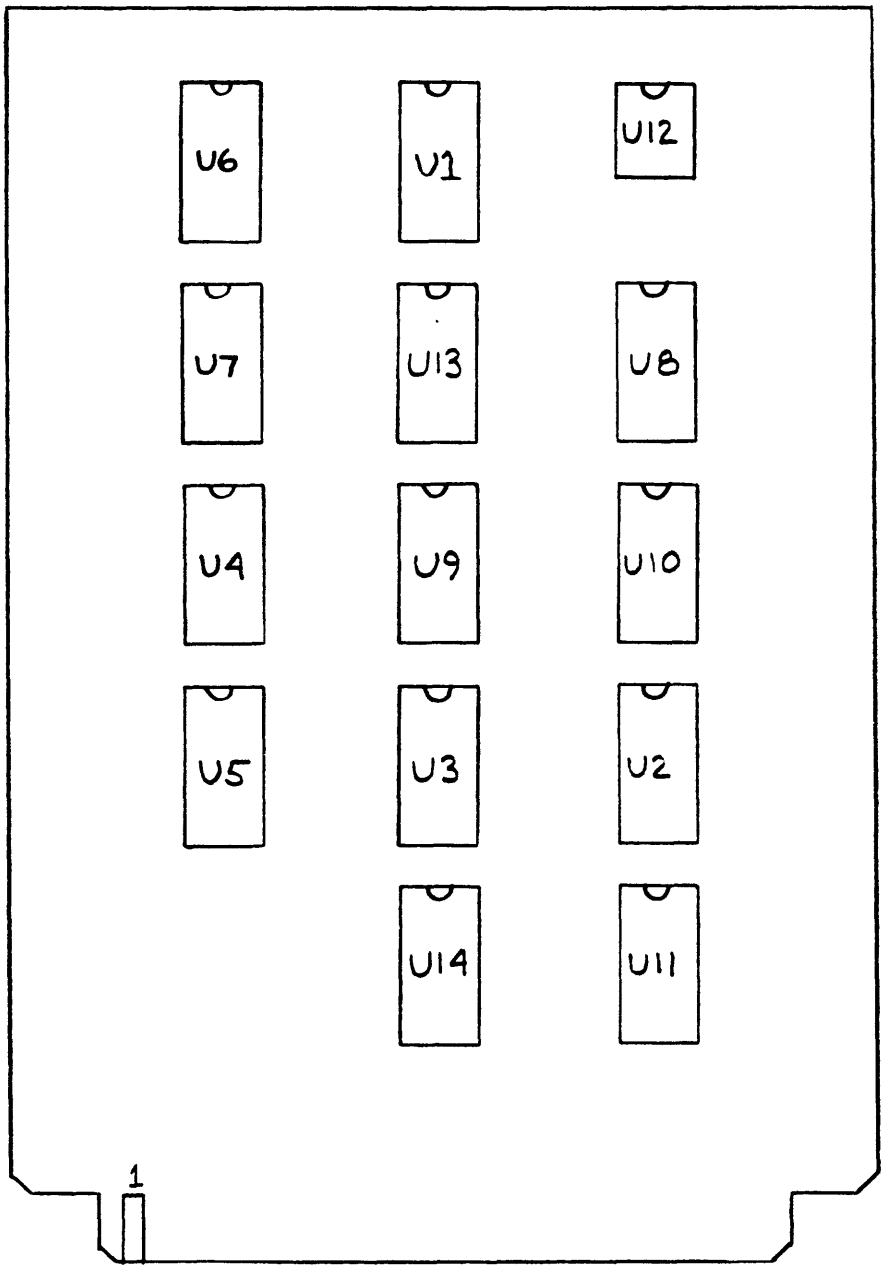


Figure 16

Control Module IC Layout

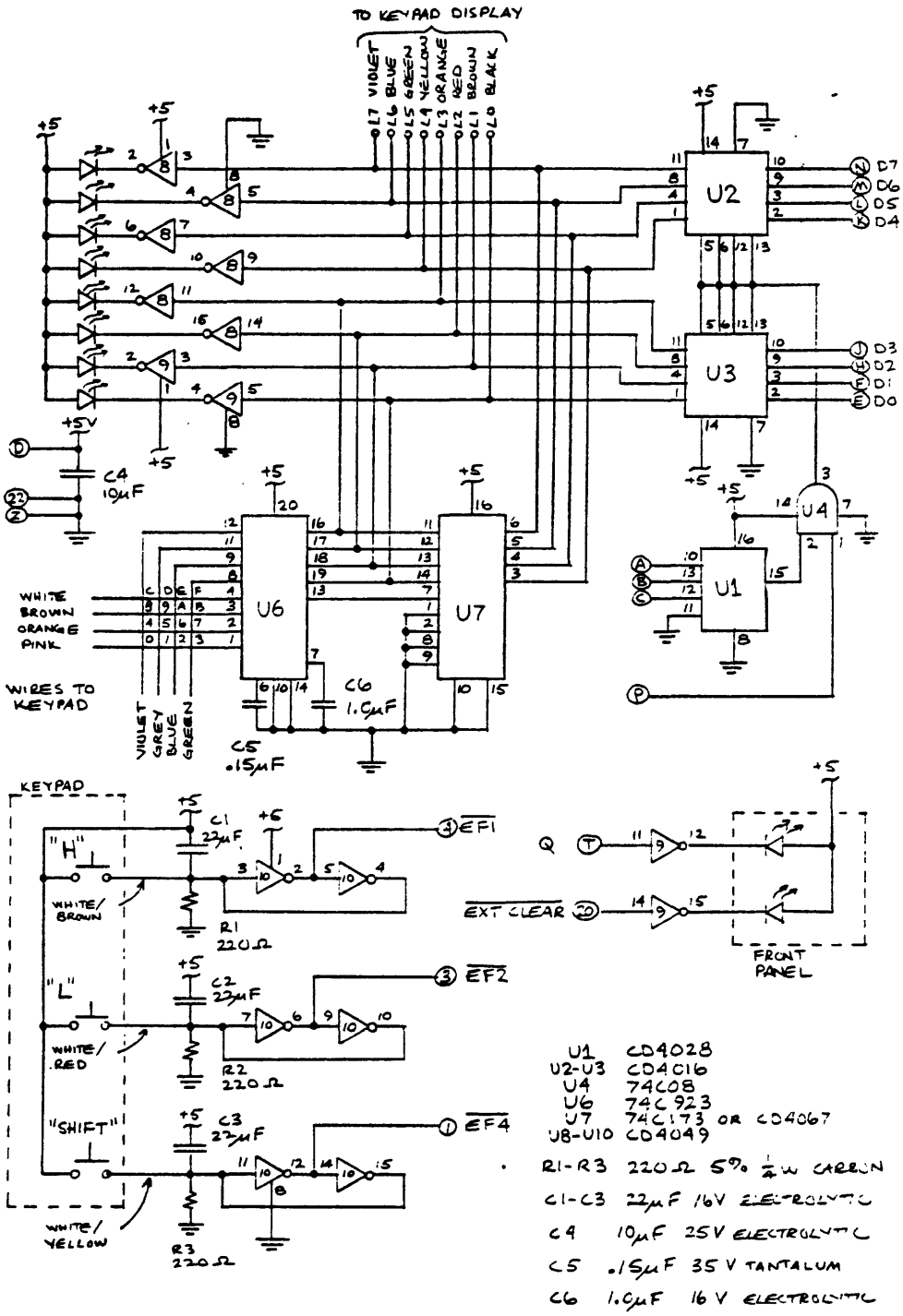


Figure 17

Keypad Module Circuit Diagram

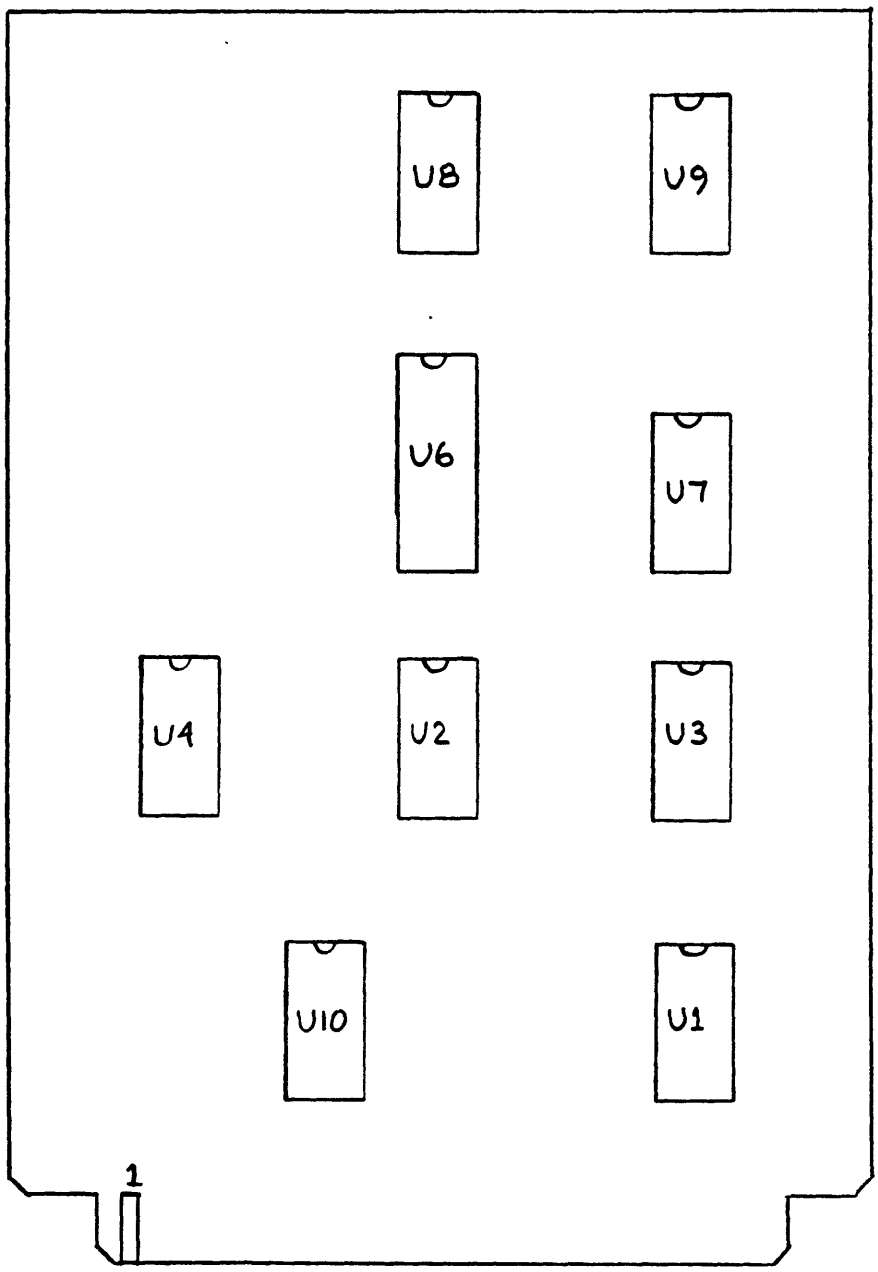


Figure 18

Keypad Module IC Layout (Estero)

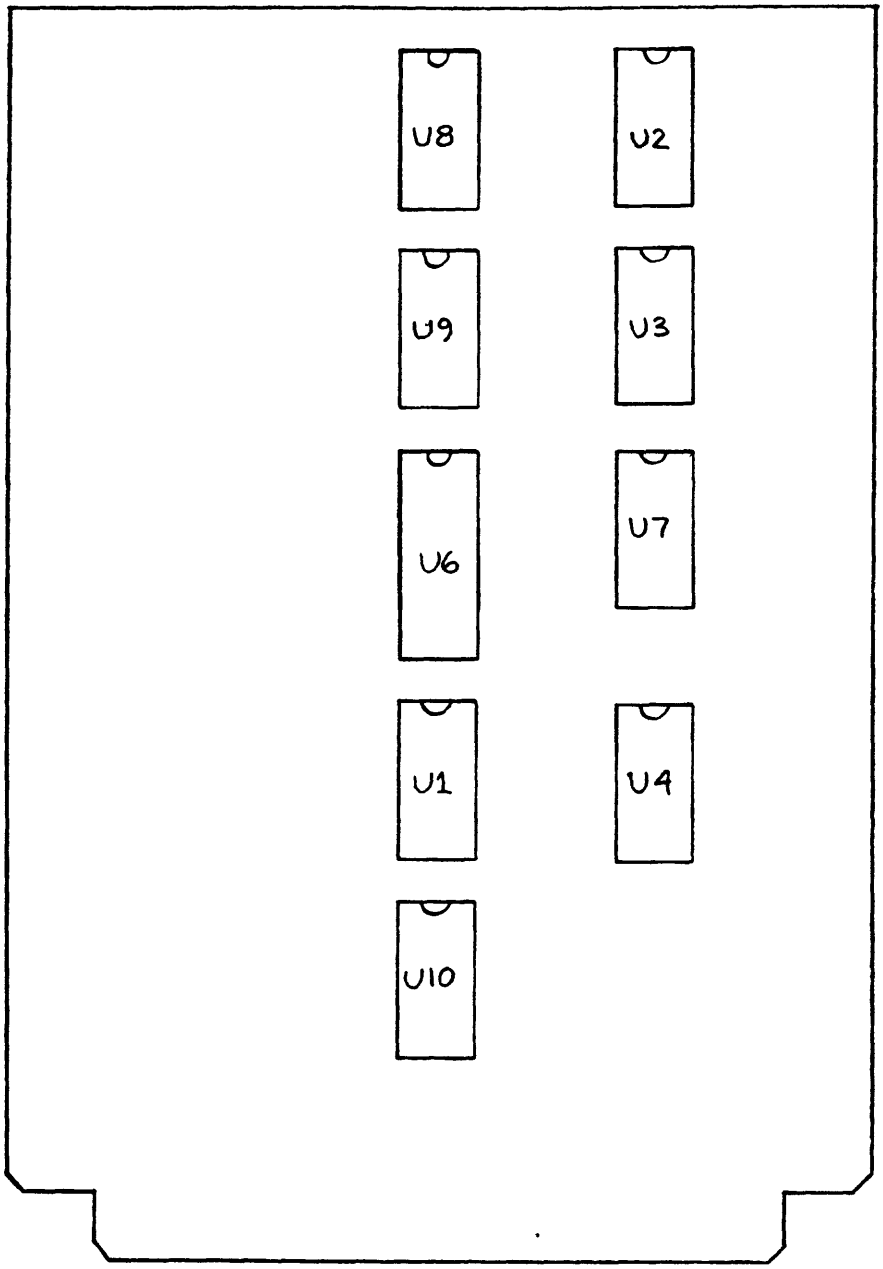


Figure 19

Keypad Module IC Layout (Polaris)

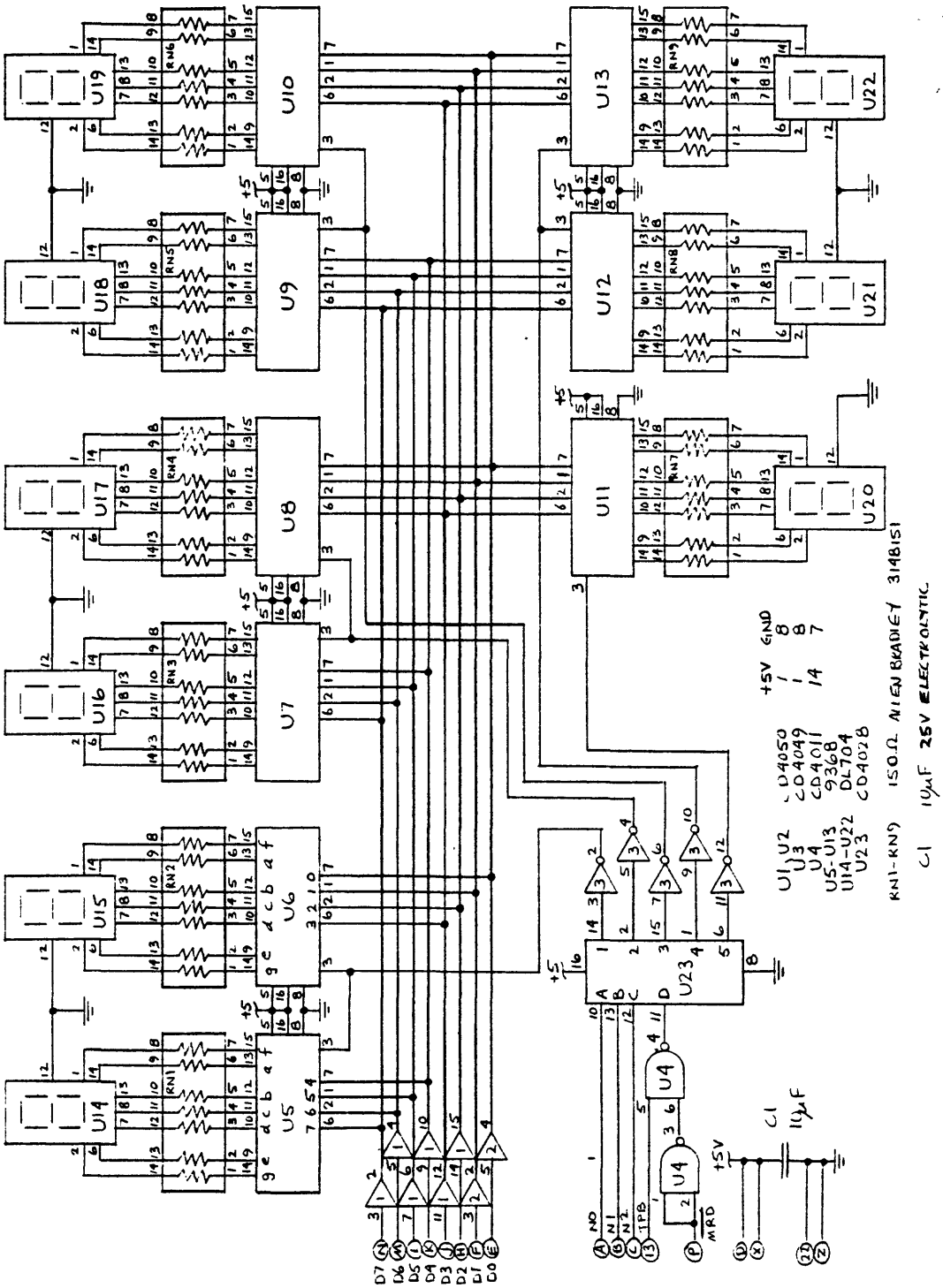


Figure 20

Display Module Circuit Diagram

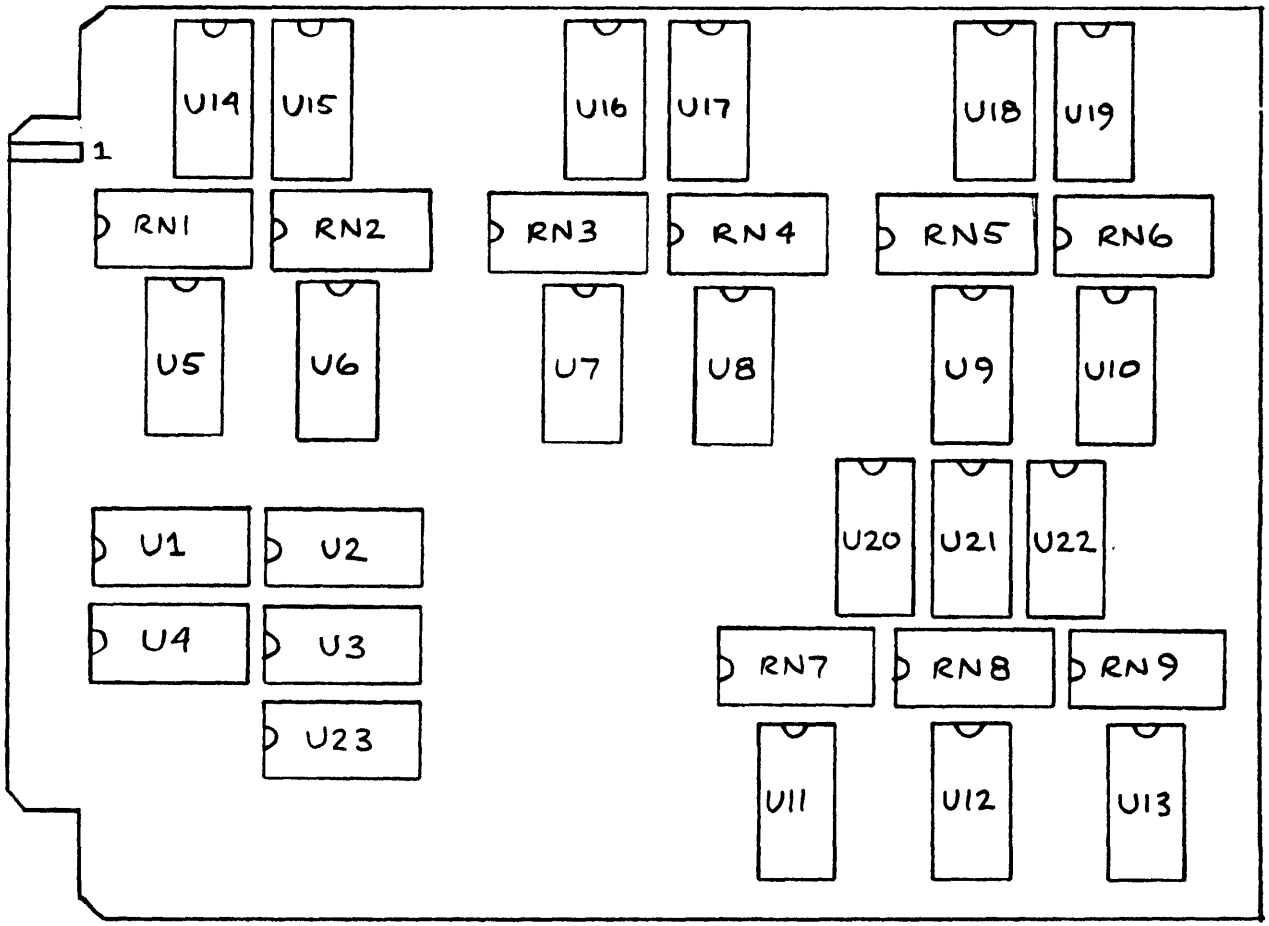


Figure 21

Display Module IC Layout

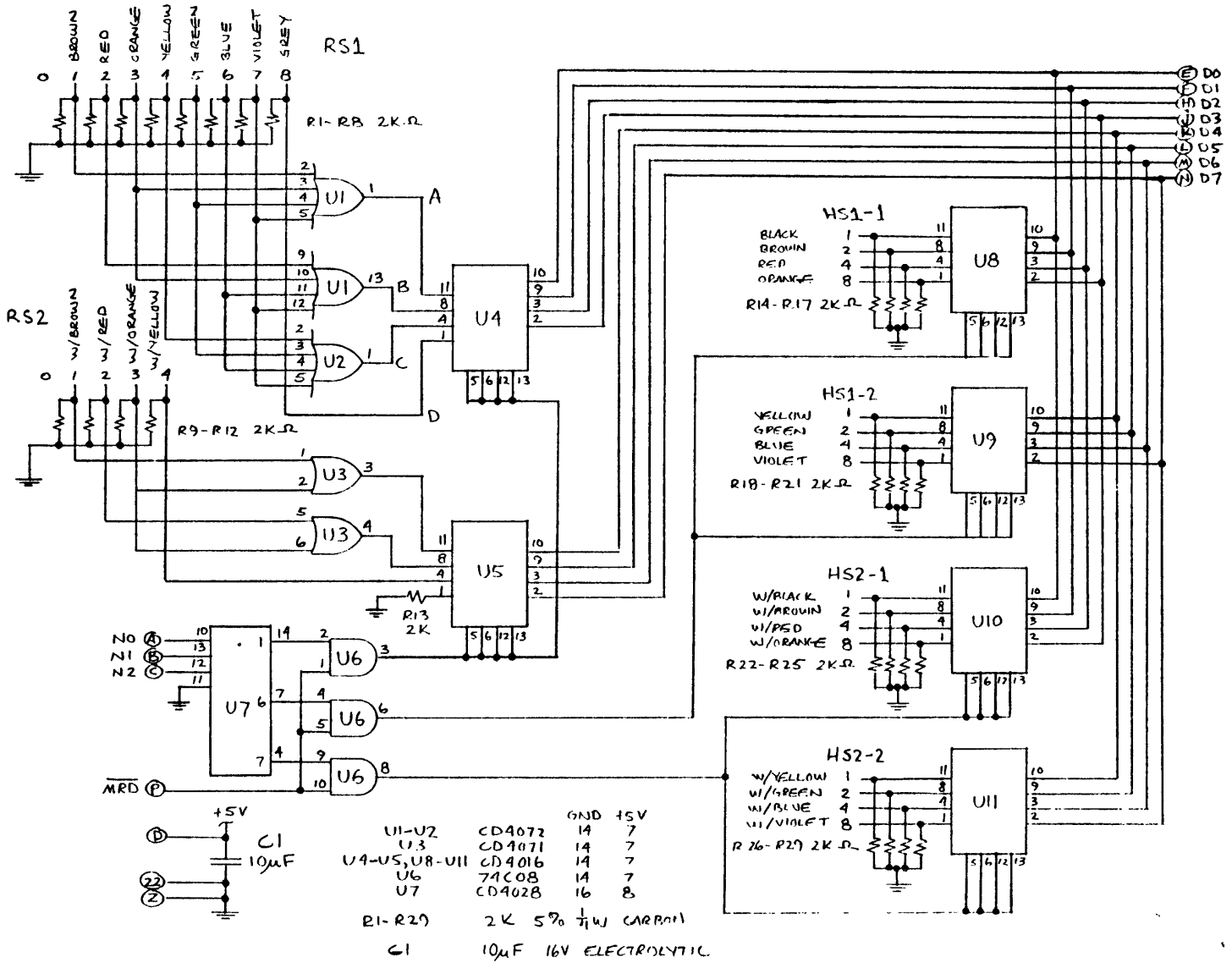


Figure 22

Switch Input Module Circuit Diagram

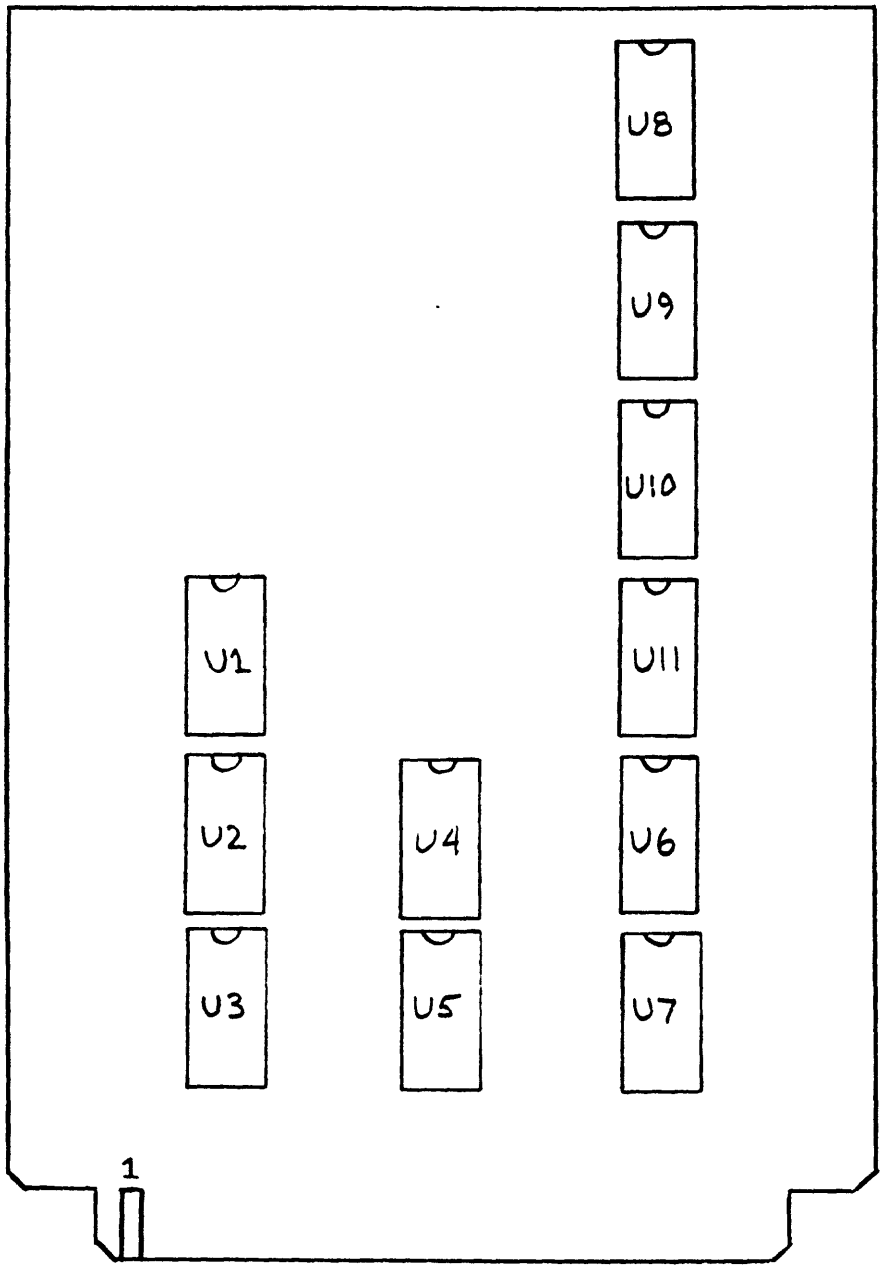
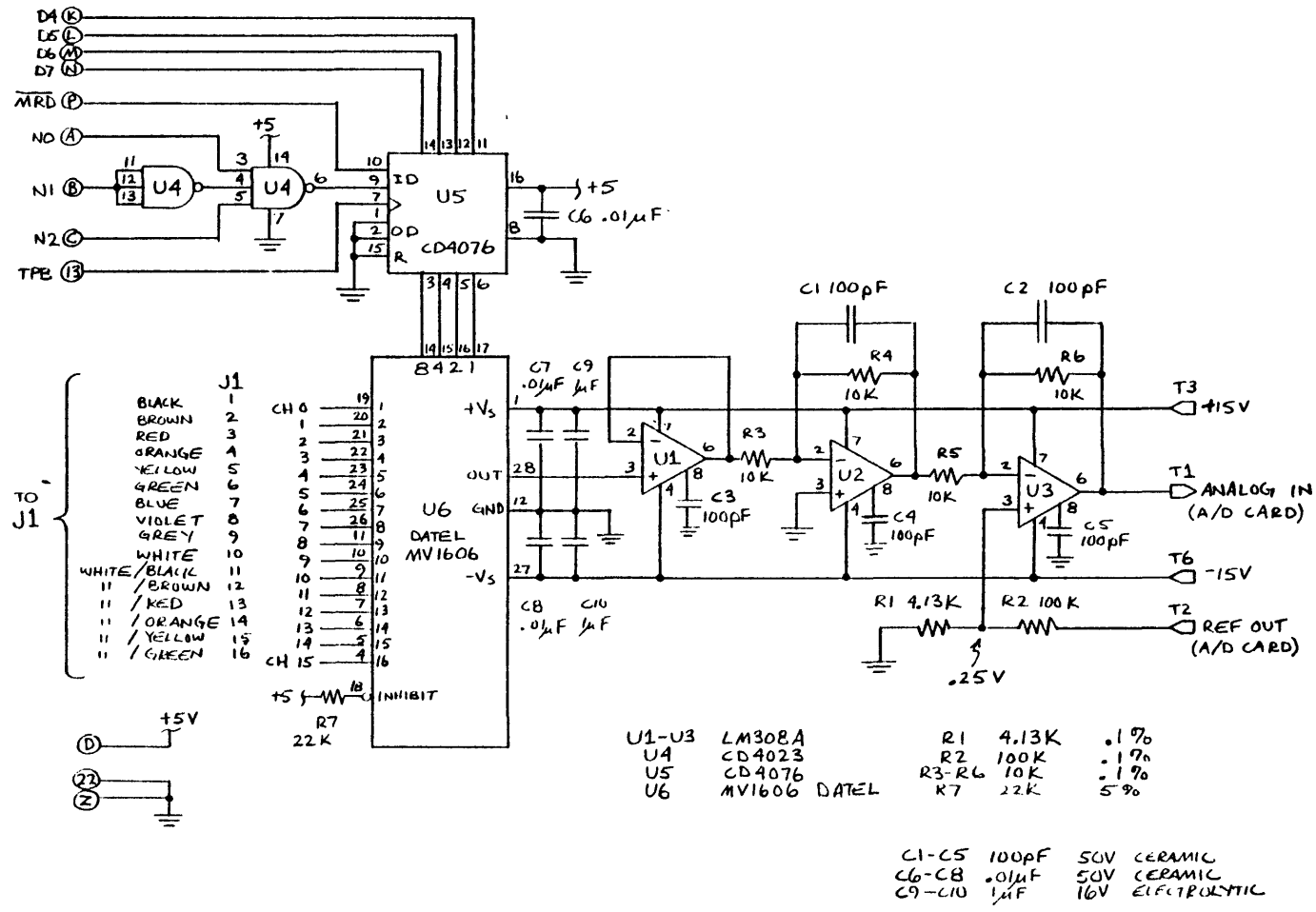


Figure 23

Switch Input Module IC Layout



U1-U3	LM308A		R1	4.13K	.1%
U4	CD4023		R2	100K	.1%
U5	CD4076		R3-R6	10K	.1%
U6	MV1606	DATEL	R7	22K	5%

C1-C5	100pF	50V	CERAMIC
C6-C8	.01µF	50V	CERAMIC
C9-C10	1µF	16V	ELECTROLYTIC

Figure 24

Multiplexer Module Circuit Diagram

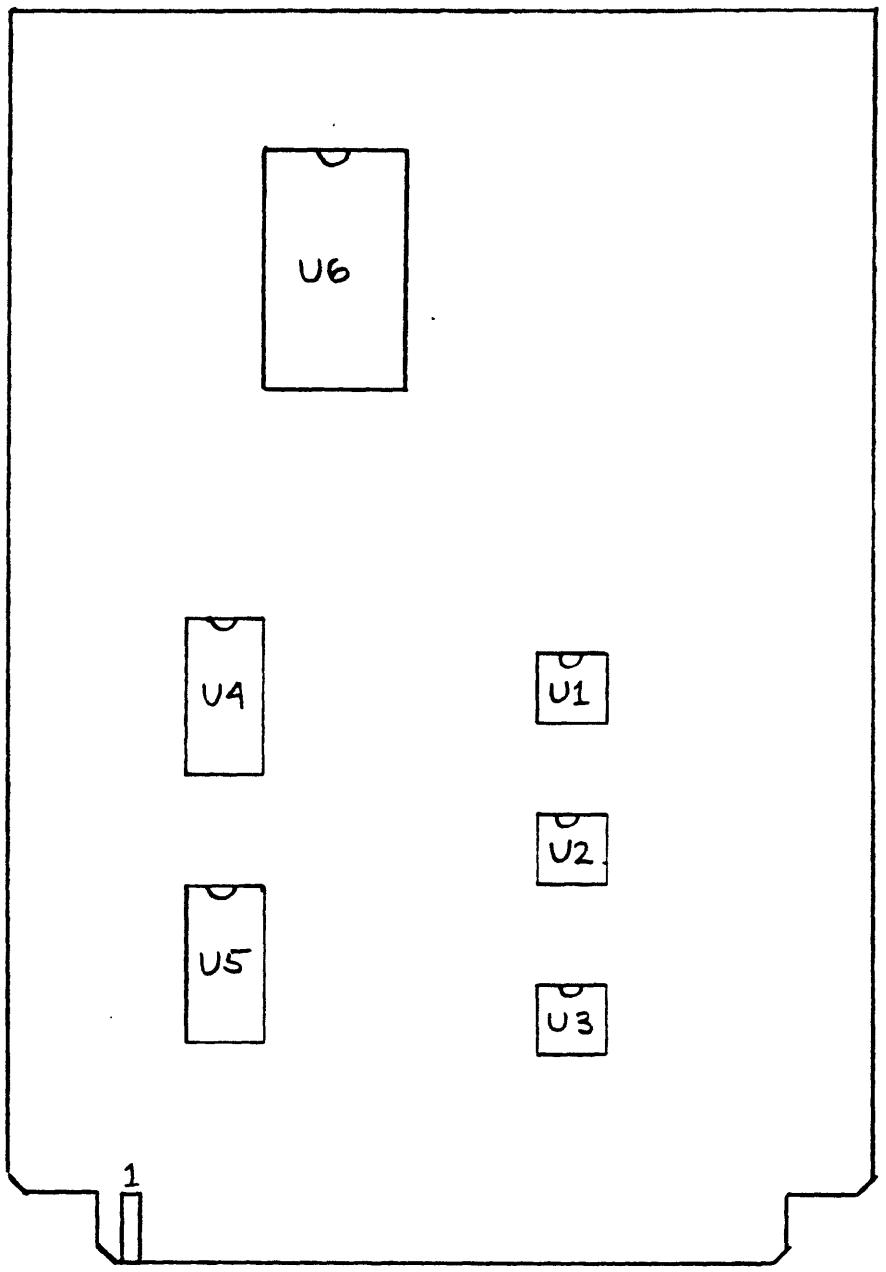


Figure 25

Multiplexer Module IC Layout

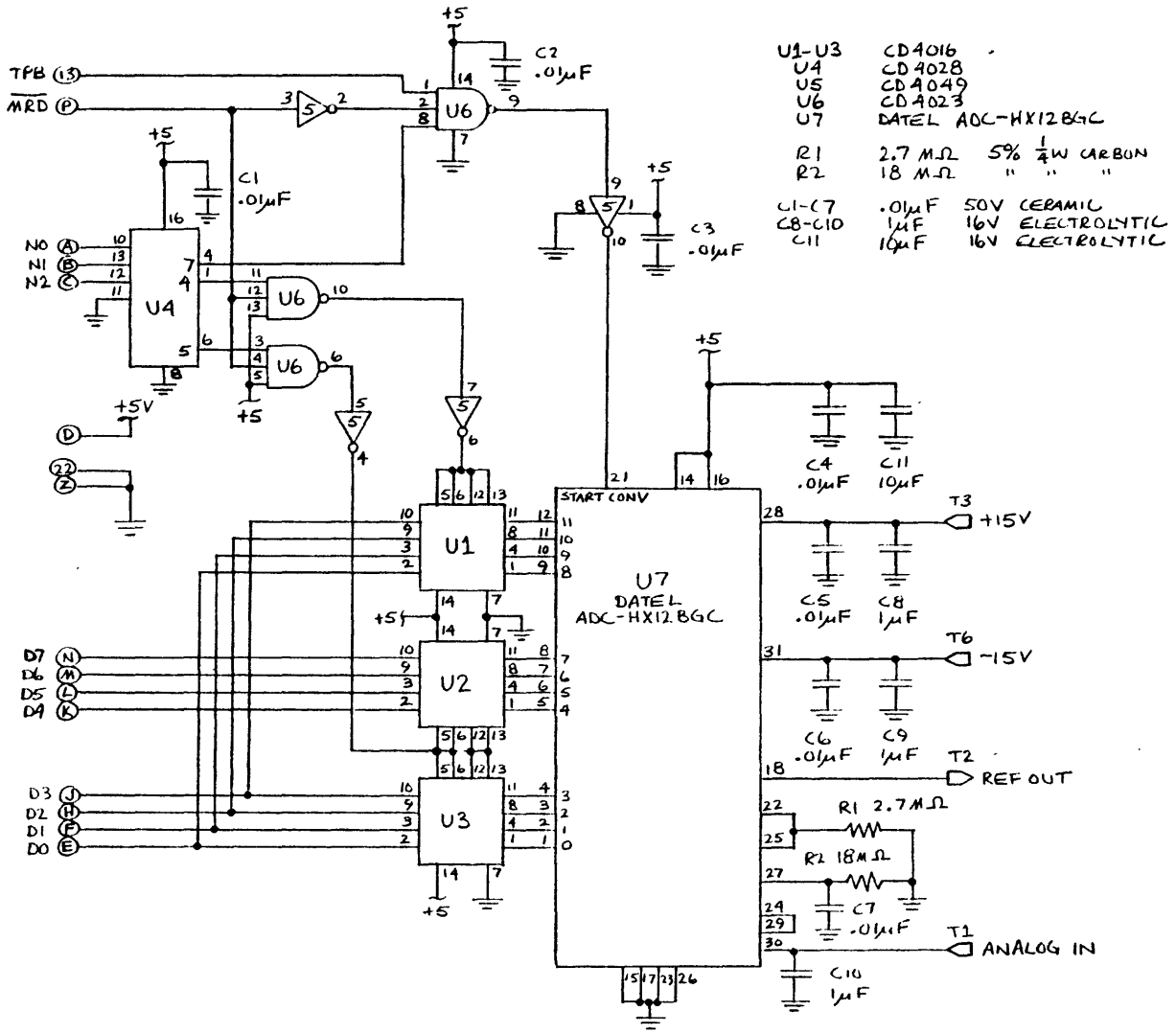


Figure 26

A/D Module Circuit Diagram

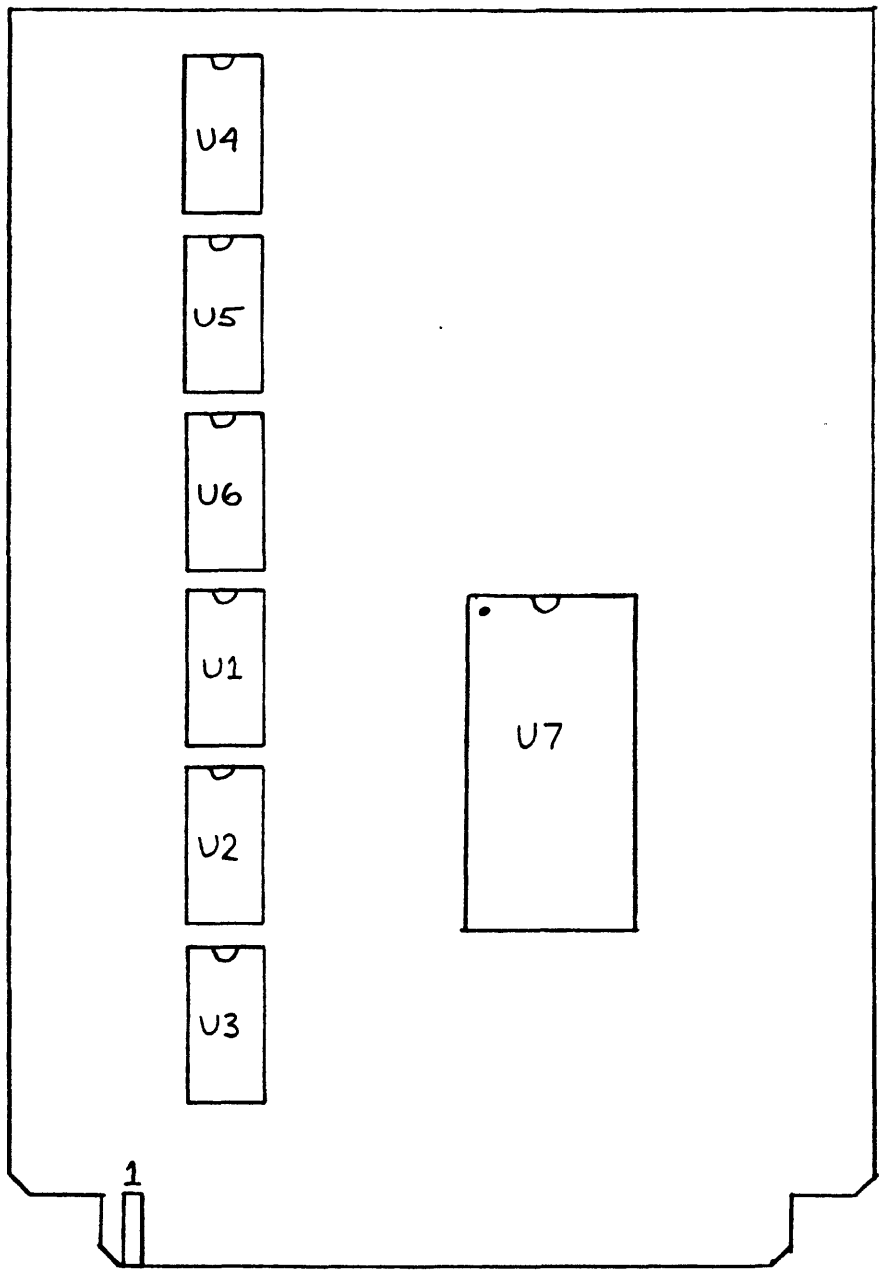


Figure 27

A/D Module IC Layout (Estero)

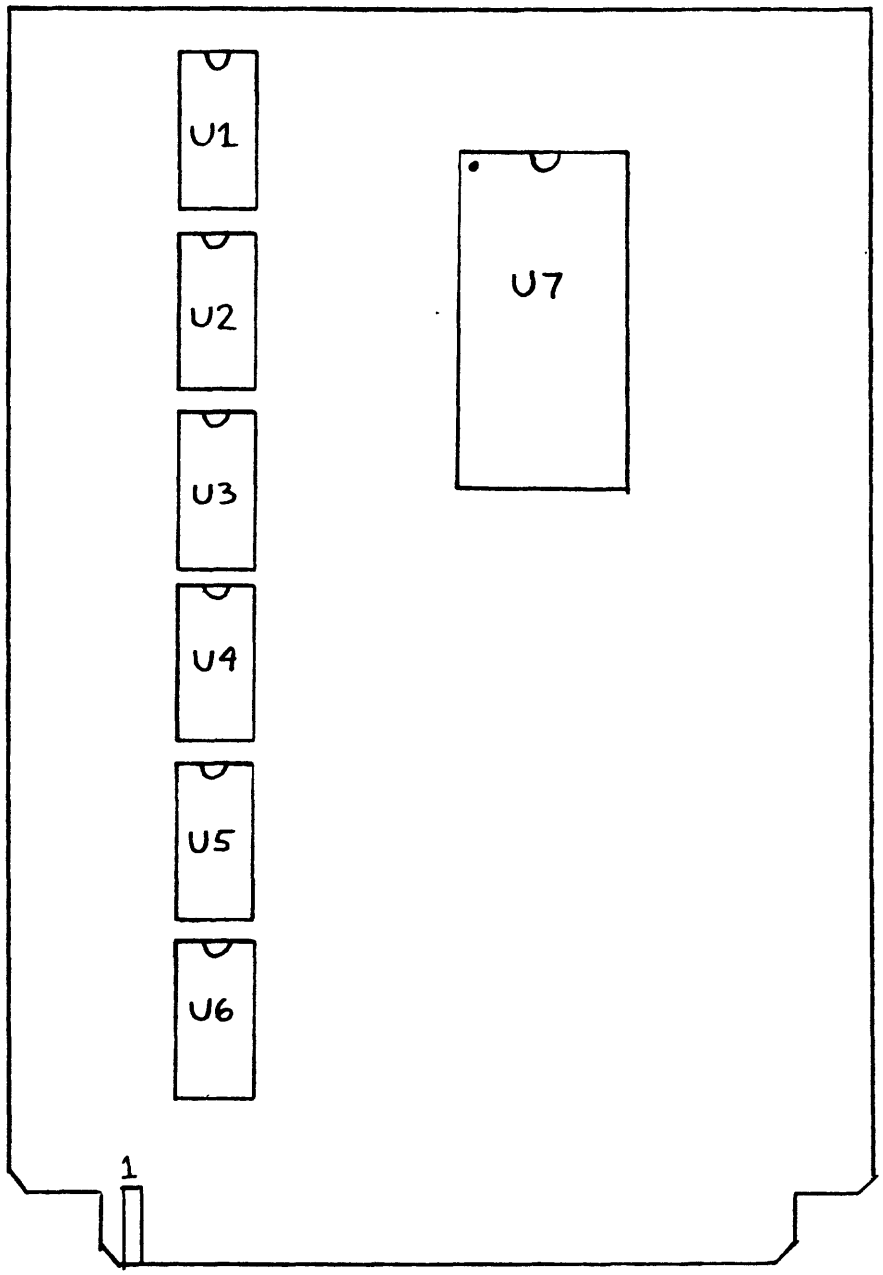


Figure 28

A/D Module IC Layout (Polaris)

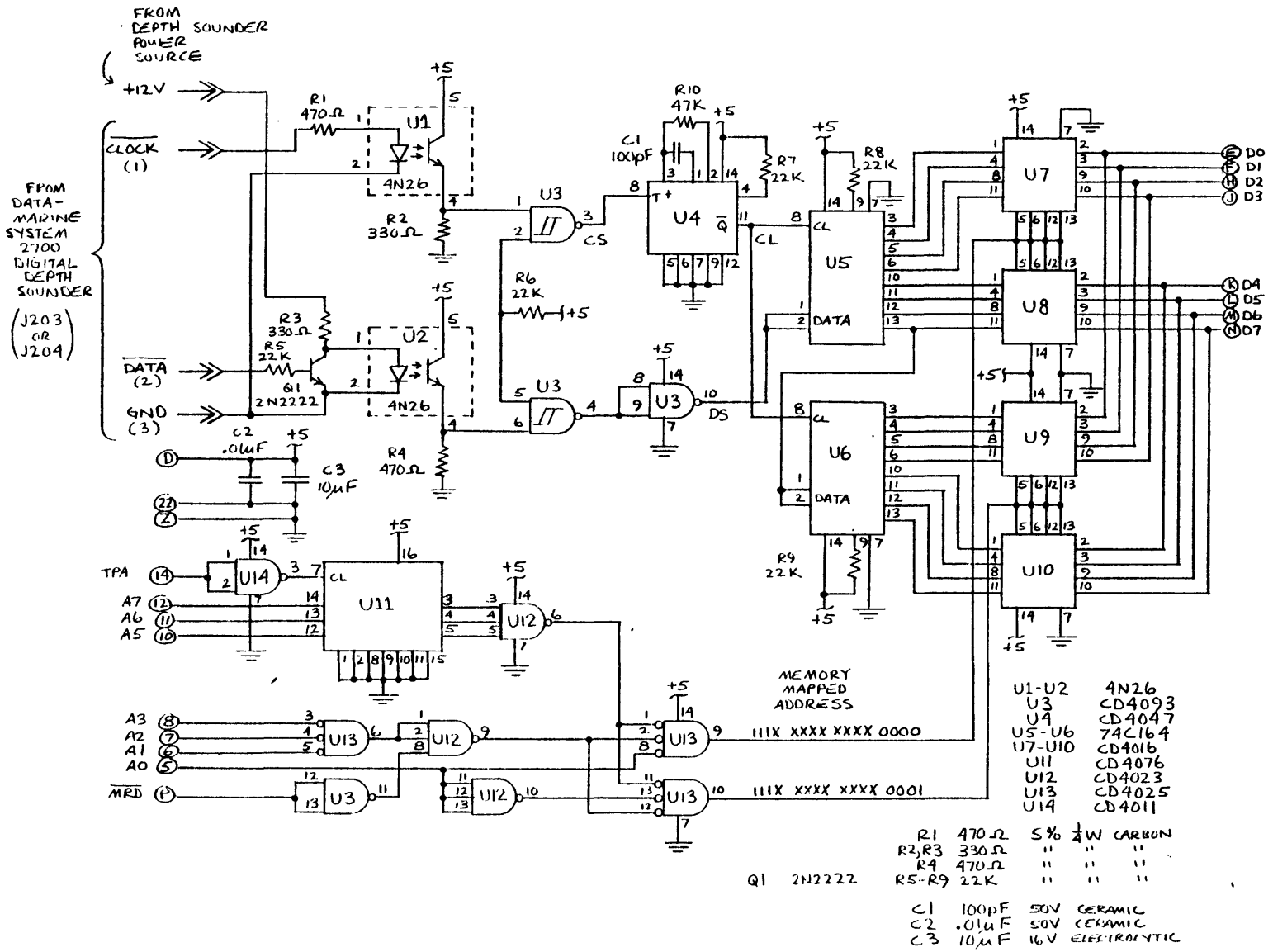


Figure 29

Depth Module Circuit Diagram

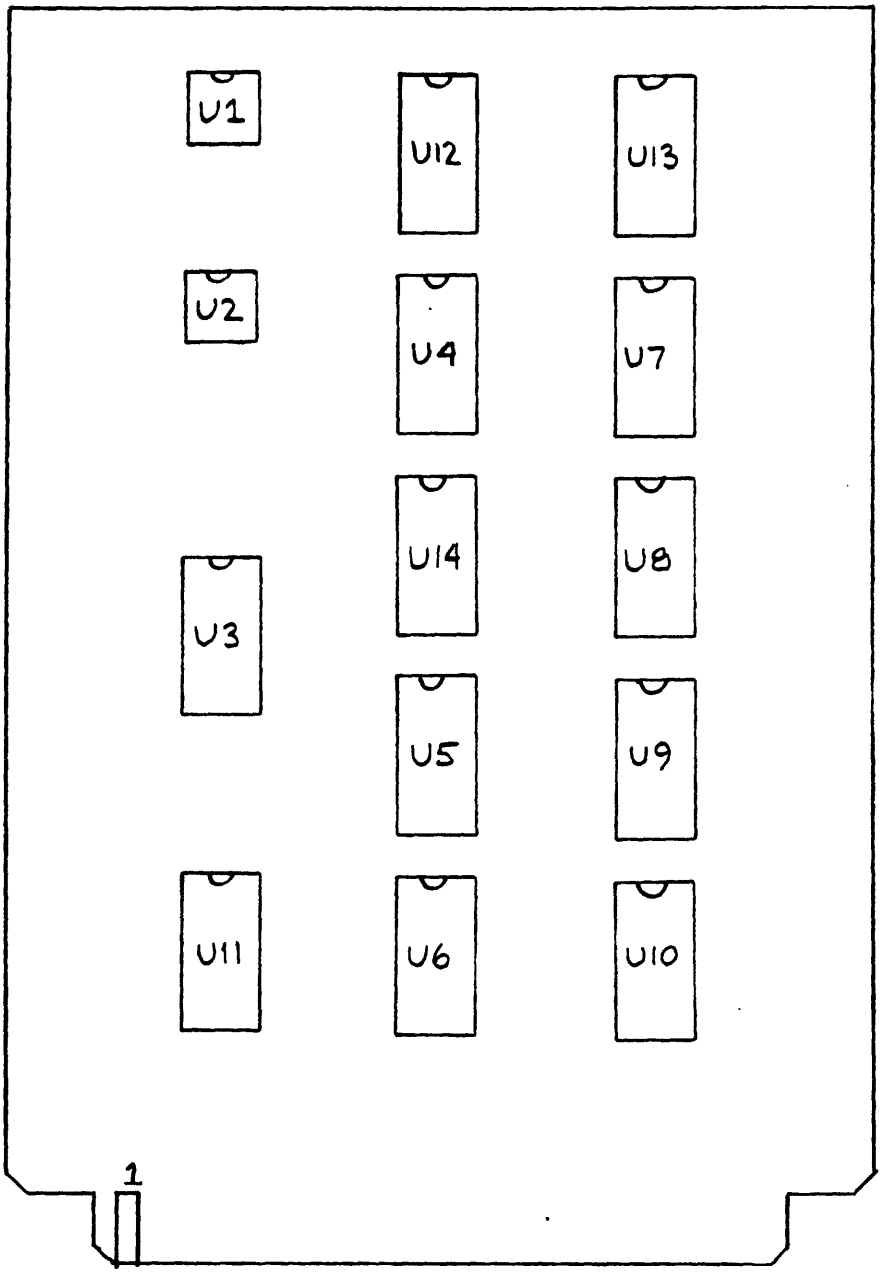
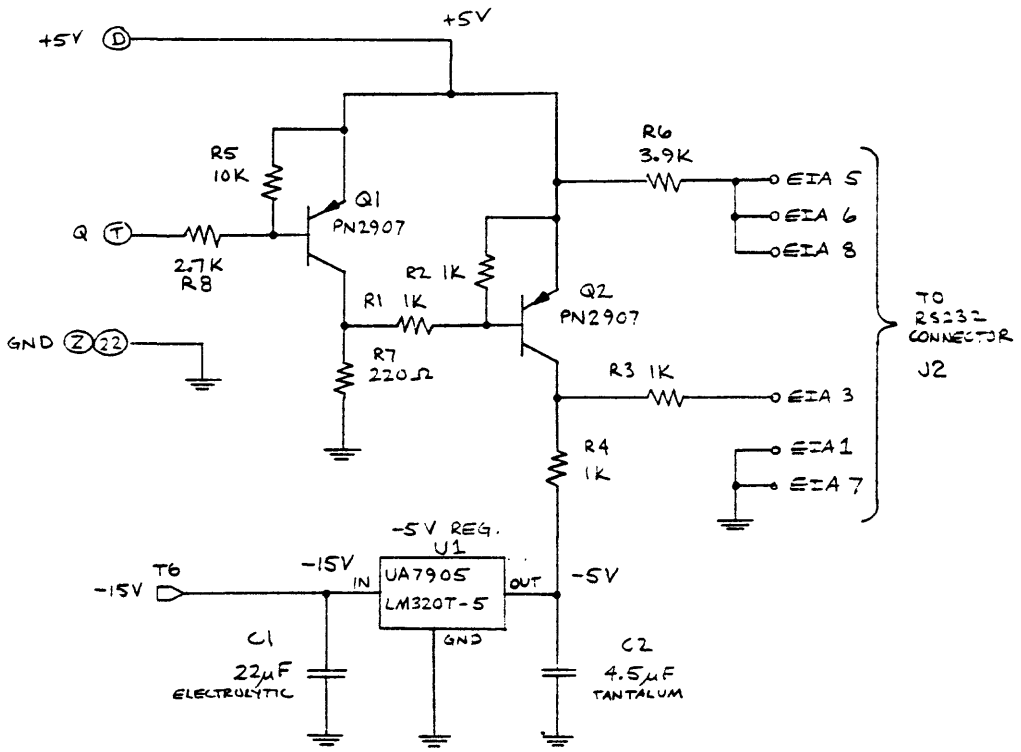


Figure 30

Depth Module IC Layout

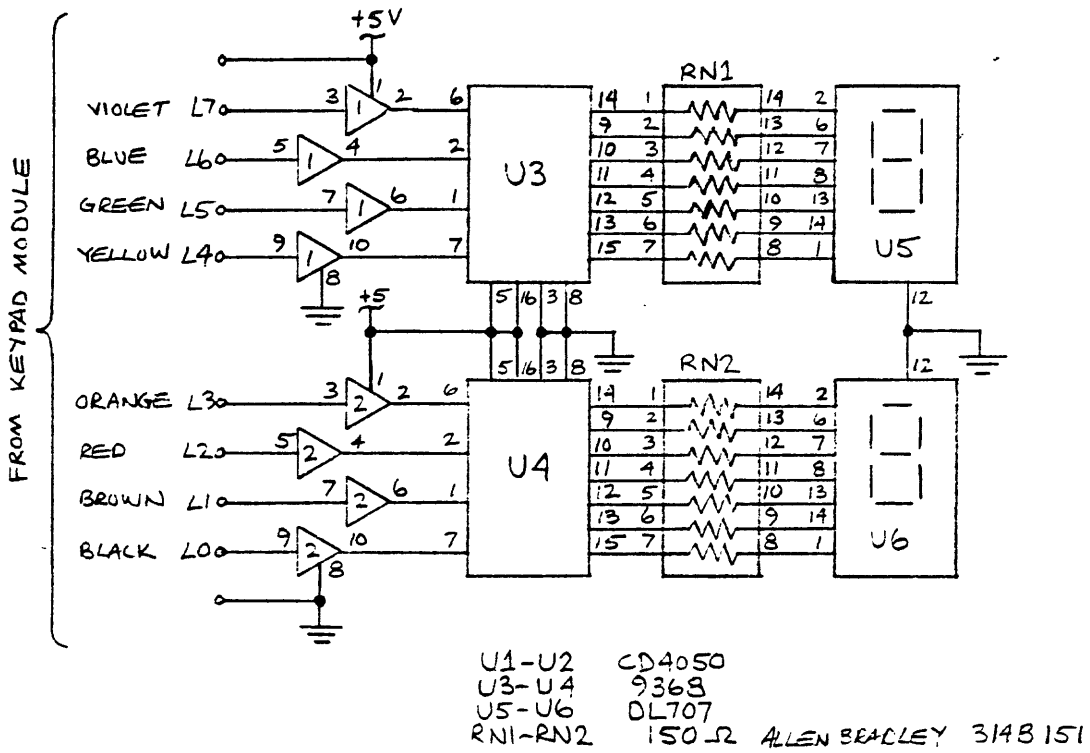


NOTE : ALL RESISTORS 5% ¼W CARBON

Q1,Q2	PN2907	
U1	UA7905 OR LM320T-5	
R1-R4	1K	
R5	10K	
R6	3.9K	
R7	220Ω	
R8	2.7K	
C1	22μF	16V ELECTROLYTIC
C2	4.5μF	25V TANTALUM

Figure 31

Serial-Output Module Circuit Diagram



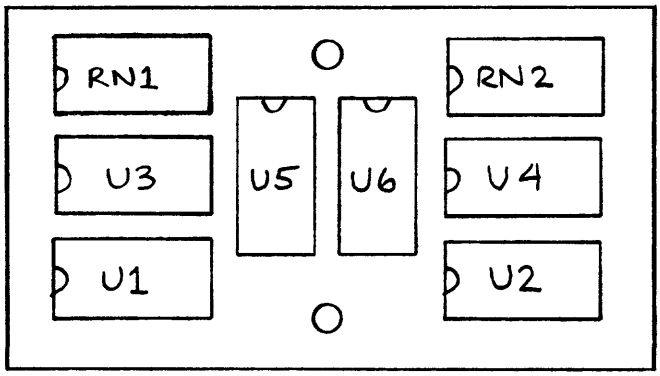
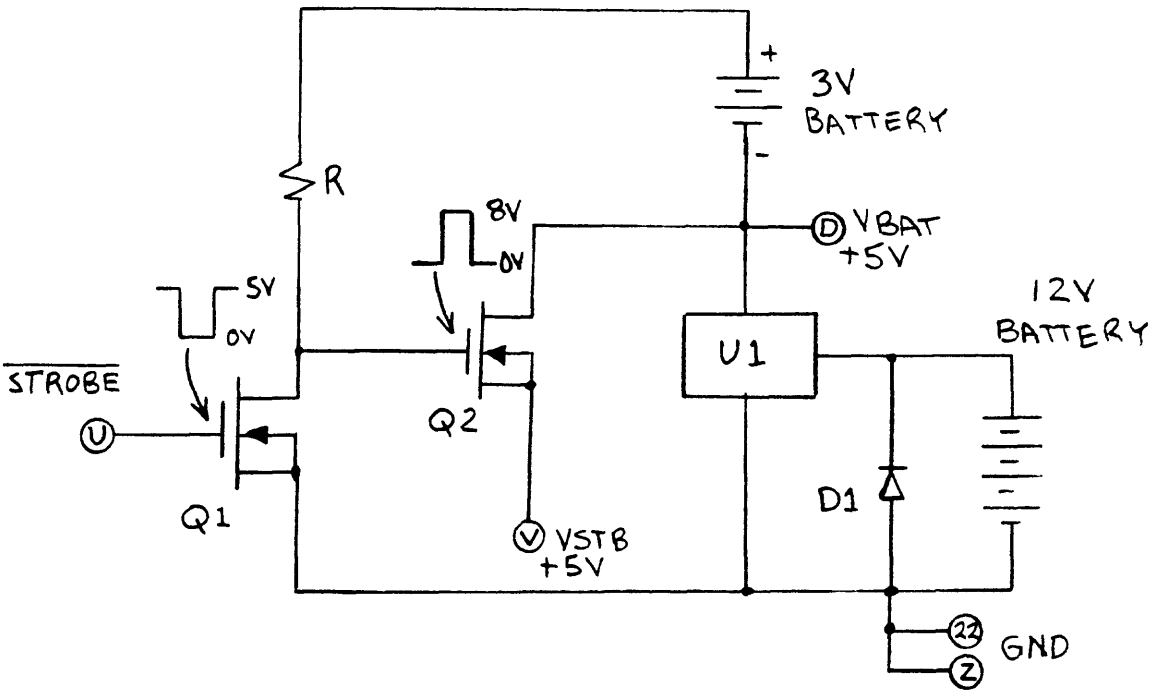


Figure 32

Keypad Display Circuit
and IC Layout



Q1 3N171 MOTOROLA
n-channel enhancment mode FET

Q2 VN46AF SILICONIX
VMOS power transitor

R1 1MΩ

U1 7805 +5V Regulator

D1 IN4003 protection diode

Figure 33

Battery Operated Power Circuit Diagram