SMITH SYSTEMS INC.
215 Joliet St. Suite 1
Dyer, Indiana 46311

Custom Computer Systems & Services
Engineering Manufacturing Design
Programming Consulting

LLOYD L. SMITH
President

219-865-6010
219-865-6011 24 Hr.

REAL TIME CLOCK

OWNER'S MANUAL

SciTronics Inc.

RTC-100 Real-Time Clock

USER'S MANUAL

SciTronics, Inc.
P O Box 5344
523 S. Clewell Street
Bethlehem, PA 18015

(215) 868-7220

SciTronics RTC-100 Real-Time Clock

CONTENTS

FIGURES

TABLES

APPENDIX

-- Note --

All clocks have an on-board, fully charged lithium battery. Serious damage to the battery may result if the circuit board is placed on a conducting surface.

INTRODUCTION

The SciTronics Inc. S-100 Real Time Clock offers computer enthusiasts an extremely accurate time piece for their system. It employs the latest microprocessor clock chip available and is crystal-controlled to assure .002% accuracy. A lithium battery provides clock power in the event that system power is removed. The RTC-100 may be used to generate interrupts while operating in the background, providing time resolutions as fine as every second. The clock provides year, month, date, day of week, hour, minute, and second data. The RTC-100 is made to be fully compatible with the SciTronics Remote Controller for real-time control of a-c operated appliances. Interactive software packages written in BASIC provide the user with easy-to-use programs to set and read the RTC-100. There are two programs required to use your clock, RTCSET, to set the clock, and RTCREAD to read and display the clock data. In addition to Basic versions of each of these programs, an assembly language program is supplied as an example of how to use the clock in a non-Basic interrupt-driven environment.

Before the clock is installed in your system, you will have to set the base port address and, if you plan to use program interrupt, the interrupt restart address.

SELECTING THE BASE PORT ADDRESS

The S-100 clock uses four port addresses; one base address, and three additional consecutive addresses. These four ports are used to read and write various data registers within the Peripheral Interface Adaptor chip. In order to select your base address, you must set or reset the six port-select switches located on the circuit board. These switches are part of the 8-position dip switch, PORT, shown in Figure 1. The general form of the base port address is:

A0 A1 A2 A3 A4 A5 A6 A7, where A2 through A7 are user defined address bits. Putting any switch into the "ON" ("OFF") position causes that address bit to be valid when high (low).

As an example, to set the base address to 144 decimal (220 octal/90 hexadecimal), the PORT switches should be set as follows:

ORIGINAL

```
        A2          A7
         |          |
       xx00       1001

      |0Hex|     |9Hex|
```

Note that address bits A1 and A0 do not get set because they are used internally to select among the clock functions. It is important that you set the switches with reference to the markings on the circuit board and the diagram of Figure 1, below. Do not use the switch markings as a guide to the identification of the address bits as they may vary from one production run to another. The only valid base addresses are those which are multiples of 4, such as: 0, 4, 8, 12, 16, 20, ... 244, 248, 252. Table I is a listing of valid port addresses in binary and decimal. Since the Basic clock software will ask you for your base port address in decimal, choose one that does not conflict with any I/O ports already assigned in your system and note it here.

Base Port Address = _____ (decimal)

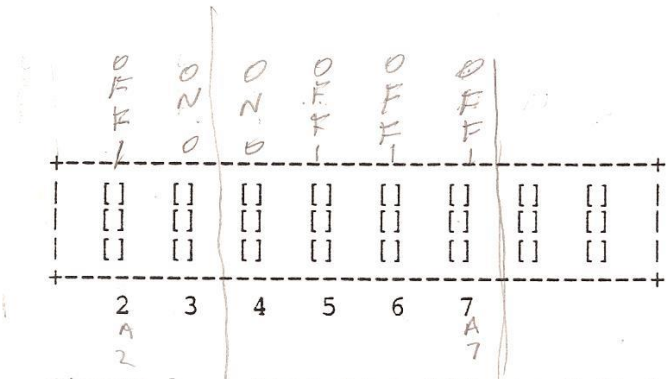


Figure 1. - Base Port Address Switch

Table I. Valid Base Port Addresses

Decimal	Binary	Decimal	Binary	Decimal	Binary
00	000000	88	011010	172	110101
04	100000	92	111010	176	001101
08	010000	96	000110	180	101101
12	110000	100	100110	184	011101
16	001000	104	010110	188	111101
20	101000	108	110110	192	000011
24	011000	112	001110	196	100011
28	111000	116	101110	200	010011
32	000100	120	011110	204	110011
36	100100	124	111110	208	001011
40	010100	128	000001	212	101011
44	110100	132	100001	216	011011
48	001100	136	010001	220	111011
52	101100	140	110001	224	000111
56	011100	144	001001	228	100111
60	111100	148	101001	232	010111
64	000010	152	011001	236	110111
68	100010	156	111001	240	001111
72	010010	160	000101	244	101111
76	110010	164	100101	248	011111
80	001010	168	010101	252	111111
84	101010				

Note: the binary digits are listed in the same order as they would be set into the PORT switch with the S-100 connector towards you.

READING THE CLOCK

Your Scitronics Real-Time clock keeps time continuously; powered by your system when it is on, and by its onboard lithium battery when your system is off. You may read the clock at any time when your system is running; an example of a Basic program to read and display the clock data is given in the Appendix. This program, RTCREAD, reads and displays the clock data on a continuous basis. The program elements, and, in particular, the clock read subroutine, may be incorporated as part of your own software.

SEQUENCE TO ENABLE CLOCK READ no interrupts active:

```
port # = dec = hex = 76 543 2 10 - binary bit order

port 1 = 240 = F0H = 11 110 0 00 - to PIA 'A' ctrl port
                                   CA2 = low = hold high =
                                   start of stop clock pulse

port 0 =  15 = 0FH =   0000 1111 - to PIA 'DIR' data input register
                       |||| ||||   direction information to be used
                       |||| ||||   later by F4H instruction
                       |||| ++++--> sets b0-b3 to output
                       ++++-------> sets b4-b7 to input

150us delay

port 1 = 244 = F4H = 11 110 1 00 - to PIA 'A' ctrl port
                                   sets CA2  low & port direction
                                   CA2 = low = hold high =
                                   clock stopped

port 3 = 248 = F8H = 11 111 0 00 - to PIA 'B' ctrl port
                                   sets CB2 high
                                   CB2 = high = start of read pulse
```

SEQUENCE TO READ CLOCK (repeat as needed):

```
digit address out to PIA port 'A'
6us delay
read PIA port 'A'
clear lower four bits
rotate upper four bits into lower four bits
save result
do next digit

AND 3 to month tens digit - eliminate leap year indicator
AND 3 to hours tens digit - eliminate AM/PM indicator
```

SEQUENCE TO RETURN CLOCK TO RUN MODE:

```
port 3 = 240 = F0H = 11 110 0 00 - to PIA 'B' ctrl port
                                   sets CB2 low
                                   CB2 = low = read pulse off

port 1 = 248 = F8H = 11 111 0 00 - to PIA 'A' ctrl port
                                   sets CA2 high
                                   CA2 = high = hold low =
                                   end of clock stop pulse =
                                   clock run mode active
```

SETTING THE CLOCK

Once you have installed your clock and are able to read it, you will want to set it to correspond to your local time zone. A Basic program, RTCSET, is provided for this purpose. You must tell the program your base port address (in decimal), and the current date and time. You will be asked to set the clock to the next full minute because the clock automatically resets its seconds to zero when it is set. You will be asked for the last two digits of the year, the one or two digits of the month, the date, the day of the week (Sunday = 0, etc.) and the time on a 24-hour basis. The clock is capable of handling Leap Year, although that feature is not included in the program as released.

SEQUENCE TO ENABLE CLOCK SETTING no interrupts active:

port # = dec = hex = 76 543 2 10 - binary bit order

port 0 = 255 = FFH = 1111 1111 - to PIA 'A' data input register
port 2 = 255 = FFH = 1111 1111 - to PIA 'B' data input register
data to preset both PIA ports
to output

port 1 = 244 = F4H = 11 110 1 00 - to PIA 'A' ctrl port
sets CA2 to low & DIR of port
CA2 = low = hold high
start of stop clock pulse

port 3 = 244 = F4H = 11 110 1 00 - to PIA 'B' ctrl port
sets CB2 to low & DIR of port
CB2 = low = read off

150us delay

Continued on next page.

Continued from previous page.

SEQUENCE TO SET CLOCK (repeat as needed):

port Ø = xxx = xxH = xxxx xxxx - digit address & data to PIA port 'A'

```
                     |||| ||||
                     |||| |||+--> bØ = aØ = clock digit
                     |||| ||+---> b1 = a1   select address
                     |||| |+----> b2 = a2   bits
                     |||| +-----> b3 = a3
                     ||||
                     |||+--------> b4 = dØ = data to clock
                     ||+---------> b5 = d1   bits
                     |+----------> b6 = d2
                     +-----------> b7 = d3
```

port 2 = Ø = ØØH = ØØØØ ØØØØ - to PIA 'B' data input register
turns write pulse on

port 2 = 1 = Ø1H = ØØØØ ØØØ1 - to PIA 'B' data input register
turns write pulse off

do next digit

SEQUENCE TO RETURN CLOCK TO RUN MODE:

port 1 = 248 = F8H = 11 111 Ø ØØ - to PIA 'A' ctrl port
sets CA2 high end of stop pulse
CA2 = high = hold low = run

INTERRUPT MODE OPERATION

As an alternative to reading the clock upon demand, you may set up your system to read the clock on an interrupt basis. If you read the clock under interrupt, the process of reading and displaying the data are separate functions. The clock is read every second in response to an interrupt request generated by the clock. At the conclusion of the instruction cycle in progress when the interrupt request was generated, the processor (if interrupt was enabled) initiates the machine interrupt cycle. This interrupt cycle resembles an ordinary instruction fetch cycle, except that the INTA (interrupt acknowledge) status bit is set high. The clock will respond during T3 of M1 by "jamming" an instruction onto the processor's data bus. This instruction, the RST (RESTART) instruction is of the form:

```
B0      B7
|       |
11lxxx11
```

where xxx is the binary representation of the interrupt number selected for your clock. Once the processor receives the RST, control is transferred to the instruction located at the address eight times the decimal equivalent of the binary interrupt number xxx.

For example, suppose the processor receives RST 2 upon interrupt; that is, the clock places 11010111 on the data buss. Then control will be transferred to 8 X 2 = 16.

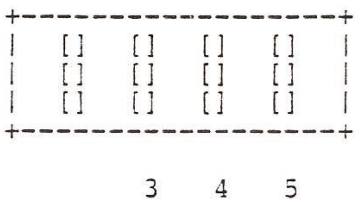


Figure 2. - Interrupt Address Switch

The three bits xxx allow control to be transferred to any one of eight fixed locations; namely, 0, 8, 16, 24, 32, 40, 48, and 56. To give the user maximum flexibility in selecting any one of these eight locations, a 4-position dip switch, INT, is provided as shown in Figure 2. Please note the markings on the diagram. Also note that the switch is negative logic, that is, ON for 0 and OFF for 1. It is recommended that you avoid having all switches on (RST 0) or all switches off (RST 7) as, frequently, these addresses are used by other devices. Table II lists legal interrupt codes and their decimal equivalents.

Note your INT code here = _____ (decimal)

Table II. - Valid Interrupt Addresses

Interrupt Code	Switch Setting	Decimal Equivalent
B0 xxx B7		
111 000 11	ON ON ON	0
111 100 11	OFF ON ON	8
111 010 11	ON OFF ON	16
111 110 11	OFF OFF ON	24
111 001 11	ON ON OFF	32
111 101 11	OFF ON OFF	40
111 011 11	ON OFF OFF	48
111 111 11	OFF OFF OFF	56

SOFTWARE DESCRIPTION

The two Basic programs, RTCREAD and RTCSET, have been described previously in brief. This section will discuss the format of the clock data as read by RTCREAD and as used by RTCSET to set the clock. This format is also used by the Scitronics a-c controller software to schedule and control a-c appliances and lights. The clock data block consists of a 13-byte area, each byte of which contains a single clock digit in its least significant 4 bits. The clock digits are in the order: year tens, year units, month tens (Leap Year flag is included with this digit), month units, day-of-week digit, hour tens (24-hour flag is included with this digit), hour units, minute tens, minute units, second tens, and second units. The read subroutine in RTCREAD places the clock board in read mode with Digit Hold on (so the digits will not change while you read them) and fills the clock data block.

The actual clock read operation requires transmitting a digit code to the clock (0=seconds units, 1= seconds tens, etc.), and then reading the clock port. At the conclusion of the read cycle, the clock is released from Hold. RTCREAD does not enable clock interrupts because it does not operate under interrupt, but reads the clock continuously. Because it cycles through the read and display portions of the program continuously, and because the speed of the various Basic programs vary considerably, the program may display the time more or less often than once per second on your system.

RTCSET uses a similar clock data block to set the clock; however, the format of the data actually used for setting the clock is somewhat different. The first part of RTCSET asks the user for the clock base port address and the clock setting information. Next the information is formatted with the data in the high order 4 bits and the digit code in the low order four bits. The digit code is the same as that used in reading the clock, that is 0=seconds units, 1=second tens, etc. The clock is set by the sequence of port operations shown in the program which transmit the data-digit/code-digit bytes to the clock and then place the clock back into normal mode. Once again, the clock interrupt is not enabled in the Basic programs.

The Appendix also contains a listing of a program demonstrating interrupt-driven use of the RTC100. This program is divided into two major sections: first, a section which is user-dependent; and second, the clock routines to set, read, start, and stop the clock. These routines, SETTER, INTR, STARTS, and STOPS should form part of any interrupt-driven software.

SETTER expects to find the clock data in the low four bits of the bytes in the clock area. It shifts each to the high four bits, inserts the proper code digit and sends the digits to the clock. SETTER must be run with interrupt off to avoid having INTR affect the data being set up; also, it does not enable interrupt mode after the clock is set. It is important to realize that the clock chip keeps time even when interrupt is off.

INTR is the interrupt service routine which is invoked automatically each second to read the clock data into the data array, CLOCK. The data is placed as individual digits into the array; the Leap Year and 24-hour flags are stripped off prior to storage. As shown in the first part of the demonstration program, the clock data may be printed from the data array by adding ASCII zero to each digit and sending them to an ASCII printer. INTR restores interrupt mode when it has finished reading the clock data.

Once the clock has been set, it will keep time and may be read; however, it will not be enabled to interrupt back ground programs until the program STARTS is run. This program enables the seconds 'tick' from the clock chip through the PIA chip to cause a program interrupt request. You must have set the INT switch on your RTC100 board to a suitable address and that address must be used for the program constant, RSTART. STARTS will set up the interrupt vector at RSTART to point to the interrupt service routine, INTR, before enabling interrupt.

You must have an interrupt vector at RSTART and the service routine INTR active in memory whenever the clock is generating interrupts. If not, your computer will get 'lost' and crash whatever program you are running. The program, STOPS, is provided to disable the clock interrupts so you may use your computer in the normal manner. STOPS disables the clock interrupts in two ways; first, it disables ALL interrupts by the command, DI, and second, it programs the PIA chip to not pass on the clock 'ticks'. If you are running other interrupt-driven devices besides the RTC100, you may wish to remove the DI command from the beginning of STOPS. Once again, STOPS stops the interrupts, not the time-keeping!

SOFTWARE

APPENDIX

I. RTCREAD.BAS

```
10      REM **READ SCITRONICS REAL-TIME CLOCK**
20      REM **COPYRIGHT 1980 - SCITRONICS, INC.**
30      DIM TIME(13),W$(7)
40      DATA SUN,MON,TUE,WED,THU,FRI,SAT
50      FOR X=1 TO 7 : READ W$(X) : NEXT X
60      PRINT
70      PRINT "PROGRAM TO DISPLAY CLOCK DATA"
80      PRINT "SCITRONICS, INC., BETHLEHEM, PA"
90      PRINT
100     REM THIS IS THE NON INTERRUPT-DRIVEN VERSION
110     PRINT "ENTER CLOCK BASE PORT ADDRESS"
120     PRINT "IN DECIMAL INTEGER FORM - 0,4,8,12,16,20,ETC"
130     PRINT "MAXIMUM VALUE = 252"
140     INPUT P
150     REM CHECK FOR PROPER VALUE AND RANGE
160     IF P>=256 THEN 60
170     M=INT(P/4)
180     IF (4*M-P)<>0 THEN 60
190     P1=P+1
200     P2=P+2
210     P3=P+3
220     REM DISPLAY LOOP STARTS HERE
230     GOSUB 1000
240     Y=TIME(1)*10+TIME(2)
250     MO=TIME(3)*10+TIME(4)
260     D=TIME(5)*10+TIME(6)
270     W=TIME(7)+1
280     H=TIME(8)*10+TIME(9)
290     MI=TIME(10)*10+TIME(11)
300     S=TIME(12)*10+TIME(13)
310     PRINT Y;":";MO;":";D;" ";
320     PRINT W$(W);". ";
330     PRINT H;":";MI;":";S;" "
340     PRINT
350     GOTO 230
360     REM
```

Continued on next page.

Continued from previous page.

```
1000    REM ***************************
1010    REM THIS READS THE CLOCK DATA *
1020    REM ***************************
1030    OUT P1,240
1040    OUT P,15
1050    OUT P3,252
1060    OUT P1,244
1070    FOR X=1  TO 13
1080    OUT P,13-X
1090    TIME(X)=INP(P) : NEXT X
1100    REM PUT CLOCK BACK INTO RUNNING MODE
1110    OUT P1,248
1120    OUT P,15
1130    OUT P3,248
1140    OUT P1,252
1150    OUT P,15
1160    FOR X=1 TO 13
1170    TIME(X)=INT(TIME(X)/16) : NEXT X
1180    REM THERE ARE SPECIAL MARKS ON HOURS AND MONTH
1190    REM STRIP THEM OFF
1200    TIME(3)=TIME(3) AND 3
1210    TIME(8)=TIME(8) AND 3
1220    RETURN
1230    REM **LATEST REVISION - 801107**
```

II. RTCSET.BAS

```
10      REM PROGRAM TO SET SCITRONICS CLOCK - 801030
20      REM **COPYRIGHT 1980 - SCITRONICS, INC**
30      DIM TIME(13)
40      PRINT "PROGRAM TO SET THE SCITRONICS CLOCK"
50      PRINT "SCITRONICS, INC., BETHLEHEM, PA 18015"
60      PRINT
70      PRINT "ENTER CLOCK BASE PORT ADDRESS"
80      PRINT "IN DECIMAL INTEGER FORM - 0,4,8,12,16,20,ETC"
90      PRINT "MAXIMUM VALUE = 252"
100     INPUT P
110     REM CHECK FOR PROPER VALUE AND RANGE
120     IF P>=256 THEN 60
130     M=INT(P/4)
140     IF(4*M-P)<>0 THEN 60
150     P1=P+1
160     P2=P+2
170     P3=P+3
180     REM GET THE TIME DATA
190     PRINT
200     PRINT "ENTER DATE IN THE FORM: YR,MO,DA "
210     PRINT "ONLY LAST 2 DIGITS FOR YR!"
220     INPUT YR,MO,DA
230     PRINT
240     PRINT "ENTER DAY-OF-THE-WEEK - 0=SUNDAY "
250     INPUT W
260     PRINT
270     PRINT "ENTER TIME ON 24HR CLOCK IN THE"
280     PRINT "FORM:  HR,MIN"
290     PRINT "CLOCK ALWAYS STARTS ON 00 SECONDS, SO"
300     PRINT "BE SURE TO ENTER THE NEXT MINUTE."
310     INPUT HR,MIN
320     REM PREPARE THE DATA FOR ENTRY
330     TIME(1)=16*INT(YR/10)+12
340     TIME(2)=16*(YR-10*INT(YR/10))+11
350     TIME(3)=16*INT(MO/10)+10
360     TIME(4)=16*(MO-10*INT(MO/10))+9
370     TIME(5)=16*INT(DA/10)+8
380     TIME(6)=16*(DA-10*INT(DA/10))+7
390     TIME(7)=16*W+6
400     TIME(8)=16*INT(HR/10)+5+128
410     TIME(9)=16*(HR-10*INT(HR/10))+4
420     TIME(10)=16*INT(MIN/10)+3
430     TIME(11)=16*(MIN-10*INT(MIN/10))+2
440     TIME(12)=1
450     TIME(13)=0
```

Continued on next page.

Continued from previous page.

```
460     REM DATA IS READY, ASK FOR
470     REM START SIGNAL FROM USER
480     PRINT
490     PRINT "ENTER 1 AND CARRIAGE RETURN TO SET CLOCK"
500     PRINT "PUSH CR ON THE ZERO SECOND"
510     INPUT J
520     REM THIS IS THE SETTING SEQUENCE
530     OUT P3,244
540     OUT P2,255
550     OUT P3,240
560     OUT P2,255
570     OUT P1,240
580     OUT P,255
590     OUT P3,244
600     OUT P1,244
610     FOR X=1 TO 13
620     OUT P,TIME(X)
630     OUT P2,0
640     OUT P2,1
650     NEXT X
660     OUT P1,248
670     OUT P3,240
680     PRINT "THE CLOCK IS SET"
690     END
```

III. RTC.ASM

```
;****************************************
;  SciTronics S-100 Real-Time Clock     *
;                                       *
;       Date       -    801024          *
;       Revision   -    801202          *
;       Programmer -    A. I. Larky     *
;                                       *
;****************************************
;    COPYRIGHT 1980 - SCITRONICS, INC   *
;****************************************
;
        ORG     100H
;
BDOS    EQU     5H      ;DISK SYSTEM ETP
CONIN   EQU     1H      ;READ CONSOLE DEVICE
PRINT   EQU     9H      ;PRINT LINE BUFFER
;
CR      EQU     13
LF      EQU     10
;
;****************************************
;            TEST PROGRAM               *
;  THE CODE WHICH FOLLOWS IS AN EXAMPLE *
;  OF THE INITIALIZATION AND DISPLAY OF *
;  CLOCK INFORMATION.                   *
;  IT SHOULD BE CUSTOMIZED FOR YOUR OWN *
;  APPLICATION.  THE CODE WHICH FOLLOWS *
;  THE DIVIDING LINES IS COMMON TO ALL  *
;  CLOCK PROGRAMS.                      *
;****************************************
;
        LXI     SP,STACK
        CALL    MSG
        DW      ID
        CALL    STOPS   ;STOP THE CLOCK
        CALL    SETUP   ;SET UP DATE AND TIME
        CALL    STARTS  ;START THE CLOCK
AGAIN:  CALL    DISPLA
        JMP     AGAIN
;
ID      DB      '(801202) - SCITRONICS REAL-TIME CLOCK'
        DB      CR,LF,'$'
```

Continued on next page.

Continued from previous page.

```
;************************************************
;  THE DISPLAY ROUTINES BUILD AN OUTPUT STRING  *
;  AT "LINE" WHICH CAN BE PRINTED TO SHOW DATE  *
;  AND TIME.  THE USER SHOULD INSERT HIS OWN    *
;  PRINT ROUTINE IN PLACE OF THE BDOS CALL IN   *
;  THE ROUTINE "MSG".                           *
;************************************************
;
DISPLA: LXI     H,OLD
        MOV     A,M     ;EXAMINE FLAG
        ANA     A
        JZ      DISPLA  ;SAME OLD DATA
        DI              ;DON'T INTERRUPT WHILE
                        ;WE'RE MAKING THE LINE IMAGE
        MVI     M,0     ;CLEAR THE FLAG
        LXI     B,6
        LXI     D,LINE  ;OUTPUT BUFFER
        LXI     H,CLOCK
        CALL    DISP8   ;DISPLAY DATE
        INX     H       ;SKIP DAY OF THE WEEK
        CALL    BLANK
        MVI     C,6
        CALL    DISP8   ;SET UP TO DISPLAY TIME
;
; IF YOU HAVE A Z-80 CPU YOU MUST SELECT MODE 0
; INTERRUPTS BY PUTTING
;       DB      0EDH
; AND . . . . . .
;       DB      46H
; IN PLACE OF THE NOP'S BELOW.
        NOP
        NOP
;
        EI              ;ITS OK TO TURN INTERRUPT BACK ON
        CALL    CRLF
        CALL    MSG     ;PRINT THE LINE IMAGE
        DW      LINE
        RET
;
DISP8:  CALL    DISP9
        CALL    DISP9   ;PLANT 2ND DIGIT
        MVI     A,':'
        CALL    DISP9A  ;PLANT COLON
        JNZ     DISP8
        RET
```

Continued on next page.

Continued from previous page.

```
;
DISP9:  MOV     A,M
        INX     H
        ANI     0FH
        ADI     '0'
        DCR     C
DISP9A: STAX    D
        INX     D
        RET
;
BLANK:  MVI     A,' '
        JMP     DISP9A
;
CRLF:   MVI     A,13
        CALL    DISP9A
        MVI     A,10
        CALL    DISP9A
        MVI     A,'$'
        JMP     DISP9A
;
SETUP:  CALL    MSG
        DW      DOYOU
        CALL    READ    ;GET ANSWER
        CPI     'Y'
        STA     OLD     ;UNKNOWN TIME
        RNZ             ;NO
        CALL    MSG
        DW      ASKFOR  ;PROMPT FOR DATA
        CALL    NUMIN   ;GET DATE AND TIME
        JMP     SETTER  ;SET THE CLOCK
;
NUMIN:  LXI     D,CLOCK
        MVI     B,4
        CALL    NUMCON  ;GET DATE INFO
        DCX     D       ;ONLY ONE DIGIT TO DAY-OF-WEEK
        LDAX    D       ;GET IT
        DCX     D       ;BACKUP
        STAX    D       ;PLANT IT
        INX     D
        MVI     B,2
        JMP     NUMCON  ;GET TIME INFO (HR. & MIN. ONLY)
```

Continued on next page.

Continued from previous page.

```
;
NUMCON: MVI     L,Ø       ;CLEAR THE NUMBER
NUMC2:  CALL    READ      ;GET ONE DIGIT
        CPI     ' '
        JZ      NUMC4     ;BLANK IS AN ENDING
        CPI     ','
        JZ      NUMC4     ;SO IS COMMA
        CPI     '.'
        JZ      NUMC4     ;SO IS PERIOD
        SUI     'Ø'       ;NOW WE ASSUME A DIGIT
        DAD     H         ;BEGIN THE *16
        DAD     H
        DAD     H
        DAD     H
        ADD     L         ;INSERT NEW DIGIT
        MOV     L,A
        JMP     NUMC2     ;GET MORE DIGITS
;  THE CODE ABOVE ENSURES THAT WE TREAT ONE-DIGIT
;   AND TWO-DIGIT ENTRIES ALIKE.
NUMC4:  MOV     A,L
        RRC               ;MOVE HIGH STUFF LOW
        RRC
        RRC
        RRC
        ANI     ØFH
        STAX    D
        INX     D
        MVI     A,ØFH
        ANA     L
        STAX    D         ;PLANT A DIGIT
        INX     D
        DCR     B         ;MORE PAIRS ?
        JNZ     NUMCON    ;YES
        RET
;
;  THE USER SHOULD INSERT HIS OWN OUTPUT
;  ROUTINE CALL IN PLACE OF THE CALL TO BDOS
;
;  NOTE:  $ IS THE END-OF-PRINT-LINE FLAG
;  FOR BDOS, NOT A PRINTABLE CHARACTER.
```

Continued on next page.

Continued from previous page.

```
MSG:    XTHL
        MOV     E,M
        INX     H
        MOV     D,M
        INX     H
        XTHL
        MVI     C,PRINT
        JMP     BDOS
;
;  THE USER SHOULD INSERT HIS OWN CHARACTER INPUT
;  ROUTINE IN PLACE OF THE CALL TO BDOS.
;
READ:   PUSH    B
        PUSH    D
        PUSH    H
        MVI     C,CONIN
        CALL    BDOS
        POP     H
        POP     D
        POP     B
        RET
;
ASKFOR: DB      '  ENTER DATE AND TIME',CR,LF
        DB      '  (YR,MON,DAY,DATE,HR,MIN.)'
        DB      CR,LF,' IN THE FORM: 8Ø,1Ø,25,6,18,12.'
        DB      CR,LF,'  (SUNDAY = DAY Ø)',CR,LF,'$'
;
DOYOU:  DB      CR,LF,'  SET NEW TIME ? - $'
;
LINE:   DS      25
OLD:    DS      1       ;OLD CHECKSUM KEPT HERE
;
;****************************************
;       END OF DISPLAY EXAMPLE          *
;****************************************
;  THE ROUTINES WHICH FOLLOW SHOULD BE  *
;  PART OF ANY CLOCK SOFTWARE.  WHILE   *
;  THE READING OF THE CLOCK IS DONE VIA *
;  PRIORITY INTERRUPT, THE USER MAY READ*
;  THE CLOCK AT ANY TIME BY CALLING THE *
;  ROUTINE "INTR".  IF THE CLOCK IS NOT *
;  TO BE READ UNDER INTERRUPT, REMOVE   *
;  THE 'EI' INSTRUCTION FROM THE END OF *
;  THE "INTR" AND "STARTS" ROUTINES.    *
;****************************************
```

Continued on next page.

Continued from previous page.

```
;
PORT    EQU     24      ;BASE PORT ADDRESS
;
;  THE VALUE FOR 'PORT' MUST MATCH THAT WHICH YOU HAVE
; SELECTED BY THE SWITCHES ON YOUR CARD.  THE ONE SHOWN
; ABOVE CORRESPONDS TO SWITCH SETTINGS: 0110000
;
PORT1   EQU     PORT+1
PORT2   EQU     PORT+2
PORT3   EQU     PORT+3
;
;  THE VALUE FOR 'RSTART' MUST MATCH THAT WHICH YOU HAVE
; SELECTED BY THE SWITCHES ON YOUR CARD.  THE ONE SHOWN
; BELOW CORRESPONDS TO SWITCH SETTINGS: ON OFF OFF
;
RSTART  EQU     48      ;RESTART ADDRESS
;
;  THIS IS THE INTERRUPT SERVICE ROUTINE WHICH IS
; INVOKED BY THE CLOCK, ONCE PER SECOND.
;
INTR:   PUSH    PSW     ;SAVE ALL REGISTERS
        PUSH    B
        PUSH    D
        PUSH    H
        MVI     A,0F0H  ;SEQUENCE TO ENABLE THE CLOCK
        OUT     PORT1   ;FOR READING
        MVI     A,0FH
        OUT     PORT
        MVI     A,07FH
LOOP1:  INR     A
        JNZ     LOOP1
        MVI     A,0FCH
        OUT     PORT3
        MVI     A,0F4H
        OUT     PORT1
        LXI     B,NDATA
        LXI     H,CLOCK+NDATA-1
```

Continued on next page.

Continued from previous page.

```
LOOP2:  MOV     A,B
        OUT     PORT    ;CODE DIGIT TO SELECT CLOCK ELEMENT
        NOP             ;WE NEED A BIT OF DELAY HERE TOO
        NOP
        NOP
        NOP
        NOP
        NOP
        IN      PORT    ;READ CLOCK ELEMENT
        ANI     0F0H    ;DROP CODE DIGIT
        RRC             ;MOVE DATA TO LOW NIBBLE
        RRC
        RRC
        RRC
        MOV     M,A     ;SAVE DATA DIGIT
        DCX     H
        INR     B
        DCR     C
        JNZ     LOOP2
;   NOW WIPE OUT THE LEAP YEAR AND 24HR MARKS
        LXI     D,HOUR-CLOCK+1
        DAD     D       ;OFFSET TO HOUR TENS
        MOV     A,M
        ANI     3
        MOV     M,A
        DCX     H       ;BACK UP TO MONTH
        DCX     H
        DCX     H
        MOV     A,M
        ANI     3
        MOV     M,A     ;WIPE OUT LEAP YEAR MARK
INTR4:  MVI     A,0F8H  ;BEGIN SEQUENCE TO RETURN
        OUT     PORT1   ;CLOCK TO INTERRUPT MODE
        MVI     A,0FH
        OUT     PORT
        IN      PORT2   ;CLEAR MINUTES INTERRUPT
```

Continued on next page.

Continued from previous page.

```
;NOTE ** IF YOU WANT TO INTERRUPT EVERY MINUTE, CHANGE
; THE TWO INSTRUCTIONS BELOW TO THE VALUES IN PARENTHESES.
        MVI     A,0F8H  ;(MVI A,0F9H)  ******
        OUT     PORT3   ;DISABLE (ENABLE) MINUTES INTERRUPT
        MVI     A,0FDH  ;(MVI A,0FCH) ******
        OUT     PORT1   ;ENABLE (DISABLE) SECONDS INTERRUPT
        MVI     A,0FH
        OUT     PORT
        STA     OLD     ;SET DISPLAY FLAG
        POP     H       ;RESTORE ALL REGISTERS
        POP     D
        POP     B
        POP     PSW
; IF YOU HAVE A Z-80 CPU, SELECT INTERRUPT MODE 0 BY
; PUTTING . . . . .
;       DB      0EDH
;AND . . . . . .
;       DB      46H
; IN PLACE OF THE NOP'S WHICH FOLLOW
        NOP
        NOP
;
        EI              ;RE-ENABLE INTERRUPT MODE
        RET
;
SETTER: DI              ;TURN OFF INTERRUPT
        MVI     A,0F4H
        OUT     PORT3
        MVI     A,0FFH
        OUT     PORT2
        MVI     A,0F0H
        OUT     PORT3
        MVI     A,0FFH
        OUT     PORT2
        MVI     A,0F0H
        OUT     PORT1
        MVI     A,0FFH
        OUT     PORT
        MVI     A,0F4H
        OUT     PORT3
        MVI     A,0F4H
        OUT     PORT1
;
```

Continued on next page.

Continued from previous page.

```
;************************************************
;  THE CLOCK IS PROGRAMMED BY OUTPUTTING THE    *
; DESIRED DIGITS, ONE AT A TIME.  EACH DIGIT    *
; MUST BE IN THE UPPER HALF OF A BYTE WITH      *
; A DIGIT IN THE LOWER BYTE TO IDENTIFY WHICH   *
; CLOCK ELEMENT IS TO BE SET.  0=SECONDS, 1=TENS*
; OF SECONDS, 2=MINUTES, 3=TENS OF MINUTES, 4=  *
; HOURS, 5=TENS OF HOURS, 6=DAY-OF-THE-WEEK, 7= *
; DATE, 8=TENS OF DATE, 9=MONTH, 10=TENS OF MON-*
; TH, 11=UNITS OF YEAR, 12=TENS OF YEAR.        *
;                                               *
;  THE LOOP BELOW SHIFTS THE INFORMATION DIGITS *
; TO THE HIGH NIBBLE, AND ADDS THE CODE DIGITS. *
;                                               *
;  NOTE: THE SECONDS MUST BE ACCESSED, EVEN     *
; THOUGH THEY CAN'T BE SET, IN ORDER TO RESET   *
; THEM.  A MARKER IS SET ON TENS OF HOURS       *
; FOR 24-HR. TIME.                              *
;************************************************
;
        LXI     H,CLOCK ;SET UP CONTROL DIGITS
        MVI     C,NDATA
START2: MOV     A,M     ;GET ONE DIGIT
        ADD     A       ;MOVE TO HIGH END
        ADD     A       ; *4
        ADD     A       ; *8
        ADD     A       ; *16
        ADD     C       ;INSERT CODE DIGIT
        DCR     A       ;CORRECT IT
        MOV     M,A
        INX     H
        DCR     C
        JNZ     START2
        LXI     D,HOUR-CLOCK-NDATA      ;OFFSET TO HOUR
        DAD     D
        MOV     A,M
        ORI     128     ;SET MILITARY TIME
        MOV     M,A
```

Continued on next page.

Continued from previous page.

```
; NOW THAT WE HAVE THE PROPER FORM OF DATA IN 'CLOCK',
; WE CAN DO THE ACTUAL CLOCK SETTING OPERATION.
        LXI     H,CLOCK ;INITIAL DATA AREA
        MVI     B,NDATA
LOOP4:  MOV     A,M     ;SET UP CLOCK
        OUT     PORT
        MVI     A,0
        OUT     PORT2   ;CREATE A STROBE PULSE
        NOP             ;WE NEED SOME DELAY
        NOP
        NOP
        NOP
        NOP
        MVI     A,1
        OUT     PORT2   ;END OF STROBE
        INX     H
        DCR     B
        JNZ     LOOP4
        MVI     A,0F8H
        OUT     PORT1
        MVI     A,0F0H
        OUT     PORT3
        RET
;
STARTS: DI
        PUSH    PSW     ;DUMMY PUSHES TO ENABLE US
        PUSH    B       ;TO USE THE TAIL END OF
        PUSH    D       ;THE "INTR" ROUTINE TO
        PUSH    H       ;ENABLE INTERRUPTS
;  SET UP THE INTERRUPT VECTOR
        LXI     H,RSTART
        MVI     M,JMP
        INX     H
        MVI     M,INTR AND 0FFH
        INX     H
        MVI     M,INTR SHR 8
        JMP     INTR4   ;NOW DO THE ENABLING
;
STOPS:  DI
        MVI     A,0
        OUT     PORT1
        OUT     PORT3
        RET
```

Continued on next page.

Continued from previous page.

```
;
;CLOCK DATA AREA
;
CLOCK:  DB      8CH,0BH ;YEAR DIGITS     (80)
        DB      0AH,99H ;MONTH           (09)
        DB      18H,87H ;DATE            (18)
        DB      26H     ;DAY-OF-WEEK     (2 = TUES.)
HOUR    EQU     $       ;OFFSET TO HOURS
        DB      05H,94H ;HOURS           (09)
        DB      13H,2H  ;MINUTES         (10)
        DB      1,0     ;SECONDS         (00)
;
NDATA   EQU     $-CLOCK
;
        DS      50
STACK:  DS      0
;
        END
```

IV. TIME.BAS

```
10  DEFINT A-Z
20  DIM TIME(13),W$(7),MO$(12)
30  P0=208   'CLOCK PORT 0D0H change to your port address
40  P1=P0+1
50  P2=P0+2
60  P3=P0+3
70  W$(0)   ="Sunday "
80  W$(1)   ="Monday "
90  W$(2)   ="Tuesday "
100 W$(3)   ="Wednesday "
110 W$(4)   ="Thursday "
120 W$(5)   ="Friday "
130 W$(6)   ="Saturday "
140 '
150 MO$ (1) =" January "
160 MO$ (2) =" February "
170 MO$ (3) =" March "
180 MO$ (4) =" April "
190 MO$ (5) =" May "
200 MO$ (6) =" June "
210 MO$ (7) =" July "
220 MO$ (8) =" August "
230 MO$ (9) =" September "
240 MO$(10) =" October "
250 MO$(11) =" November "
260 MO$(12) =" December "
270 '
280 PRINT CHR$(26)'clears screen on TeleVideo 912/920 terminal
290 PRINT
300 PRINT "Enter <T> for time display ONLY"
310 PRINT "Enter <S> for time setting"
320 PRINT "Enter <E> for EXIT"
330 X1$=INKEY$
340 IF X1$< CHR$(32) THEN 330
350 IF X1$="T" THEN 1000
360 IF X1$="t" THEN 1000
370 IF X1$="E" THEN END
380 IF X1$="e" THEN END
390 IF X1$="S" THEN 420
400 IF X1$="s" THEN 420
410 GOTO 330
```

Continued on next page.

Continued from previous page.

```
420 PRINT CHR$(26)
430 PRINT
440 PRINT "Enter DATE in the form:"
450 PRINT "  YY,MM,DD"
460 INPUT YR,MO,DA
470 PRINT
480 PRINT "Enter DAY-OF-THE-WEEK"
490 PRINT
500 PRINT "  0 = Sunday"
510 PRINT "  1 = Monday"
520 PRINT "  2 = Tuesday"
530 PRINT "  3 = Wednesday"
540 PRINT "  4 = Thursday"
550 PRINT "  5 = Friday"
560 PRINT "  6 = Saturday"
570 PRINT
580 INPUT W
590 PRINT
600 PRINT "Enter TIME on 24HR clock in the form of:"
610 PRINT "  HH,MM"
620 INPUT HR,MIN
630 'PREPARE THE DATA FOR ENTRY
640 TIME (1) =16*INT(YR/10)+12
650 TIME (2) =16*(YR-10*INT(YR/10))+11
660 TIME (3) =16*INT(MO/10)+10
670 TIME (4) =16*(MO-10*INT(MO/10))+9
680 TIME (5) =16*INT(DA/10)+8
690 TIME (6) =16*(DA-10*INT(DA/10))+7
700 TIME (7) =16*W+6
710 TIME (8) =16*INT(HR/10)+5+128
720 TIME (9) =16*(HR-10*INT(HR/10))+4
730 TIME(10) =16*INT(MIN/10)+3
740 TIME(11) =16*(MIN-10*INT(MIN/10))+2
750 TIME(12) =1
760 TIME(13) =0
770 'DATA IS READY, ASK FOR
780 'START SIGNAL FROM USER
790 PRINT
800 PRINT "Press any key to set and start clock with zero seconds"
810 X$=INKEY$
820 IF X1$< CHR$(32) THEN 810
830 IF LEN(X$)=0 THEN 810
```

Continued on next page.

Continued from previous page.

```
840 'THIS IS THE SETTING SEQUENCE
850 OUT P3,244
860 OUT P2,255
870 OUT P3,240
880 OUT P2,255
890 OUT P1,240
900 OUT P0,255
910 OUT P3,244
920 OUT P1,244
930 FOR X=1 TO 13
940    OUT P0,TIME(X)
950    OUT P2,0
960    OUT P2,1
970 NEXT X
980 OUT P1,248
990 OUT P3,240
1000 ' READ CLOCK
1010 PRINT CHR$(26)
1020 '
1030 ' THIS READS THE CLOCK DATA
1040 '
1050 OUT P1,240
1060 OUT P0, 15
1070 OUT P1,244
1080 OUT P3,252
1090 FOR X=1 TO 13
1100    OUT P0,13-X
1110    TIME(X)=INP(P0)
1120 NEXT X
1130 ' PUT CLOCK BACK INTO RUNNING MODE
1140 OUT P3,240
1150 OUT P1,248
1160 FOR X=1 TO 13
1170    TIME(X)=INT(TIME(X)/16)
1180 NEXT X
1190 ' THERE ARE SPECIAL MARKS ON HOURS AND MONTH
1200 ' STRIP THEM OFF
1210 TIME(3) = TIME(3) AND 3
1220 TIME(8) = TIME(8) AND 3
1230 '
```

Continued on next page.

Continued from previous page.

```
1240 Y$  = CHR$(TIME (1)+48)+CHR$(TIME (2)+48)
1250 MO  = TIME(3)*10+TIME(4)
1260 IF TIME(5)>0 THEN D$=CHR$(TIME(5)+48)+CHR$(TIME(6)+48):GOTO 1280
1270 D$  = CHR$(TIME (6)+48)
1280 W   =      TIME (7)
1290 H$  = CHR$(TIME (8)+48)+CHR$(TIME (9)+48)
1300 MI$ = CHR$(TIME(10)+48)+CHR$(TIME(11)+48)
1310 S$  = CHR$(TIME(12)+48)+CHR$(TIME(13)+48)
1320 '
1330 X$=INKEY$
1340 IF X1$< CHR$(32) THEN 1330
1350 IF LEN(X$)<>0 THEN 280
1360 IF S1$=S$ THEN 1050 ELSE S1$=S$
1370 '
1380 PRINT CHR$(30);'Cursor home command
1390 PRINT "                    ";
1400 PRINT W$(W);
1410 PRINT H$;":";MI$;":";S$;
1420 PRINT MO$(MO);D$;" ";"19";Y$
1430 PRINT CHR$(30);
1440 GOTO 1050
```

V. READ.ASM

--- Note ---
Interrupts not used

```
;PGM by Harry Kaemmerer 810220
                                        ;
PROM    EQU     xxxxH                   ;ADD START OF PROM ADDRESS
CONOUT  EQU     xxxxH                   ;ADD YOUR CONOUT CALL ADDRESS
PORT    EQU     xxH                     ;ADD BASE PORT ADDRESS FOR CLOCK
                                        ;
PORT1   EQU     PORT+1                  ;
PORT2   EQU     PORT+2                  ;
PORT3   EQU     PORT+3                  ;
                                        ;
SECU    EQU          0                  ;SECONDS UNITS
SECT    EQU     SECU+1                  ;SECONDS TENS
MINU    EQU     SECT+1                  ;MINUTES UNITS
MINT    EQU     MINU+1                  ;MINUTES TENS
HRSU    EQU     MINT+1                  ;HR'S    UNITS
HRST    EQU     HRSU+1                  ;HR'S    TENS
DAYW    EQU     HRST+1                  ;DAY OF THE WEEK
DAYU    EQU     DAYW+1                  ;DAY     UNITS
DAYT    EQU     DAYU+1                  ;DAY     TENS
MONU    EQU     DAYT+1                  ;MONTH   UNITS
MONT    EQU     MONU+1                  ;MONTH   TENS
YEAU    EQU     MONT+1                  ;YEAR    UNITS
YEAT    EQU     YEAU+1                  ;YEAR    TENS
                                        ;
ASCII   EQU     30H                     ;OFFSET TO MAKE CHAR. ASCII
                                        ;
        ORG     PROM                    ;
                                        ;
TIME:   CALL    TSTART                  ;DO CLOCK INIT SUB
        CALL    DAY0                    ;DISPLAY TIME
        CALL    TIMED                   ;RESET CLOCK BOARD
        RET                             ;
                                        ;
```

Continued on next page.

Continued from previous page.

```
;---------------------------------;
;     SUB ROUTINES START HERE     ;
;---------------------------------;
                                    ;
DAY0    LXI     H,STR3              ;POINTS TO LOOKUP TABLE FOR DAY
        LXI     B,0H                ;CLEAR B&C REGISTERS
        MVI     A,DAYW              ;DAY READ INSTRUCTION
        CALL    GETDIG1             ;GET IT
        MOV     C,A                 ;MOVE DAY OF WEEK POINTER IN C.
        DAD     B                   ;ADD B&C TO H&L REGISTERS
        MOV     C,M                 ;GET OFFSET ADDRESS
        LXI     H,STR4              ;GET DAY NAME
        DAD     B                   ;POINT TO NAME
        CALL    MESAG               ;SEND IT TO TERMINAL
                                    ;
HRS1    MVI     A,HRST              ;SELECT HR'S TENS DIGIT
        CALL    GETDIG1             ;GET DIGIT
        ANI     3                   ;ELIMINATE EXTRA BIT
        CALL    SENDCH              ;SEND IT TO TERMINAL
                                    ;
HRS2    MVI     A,HRSU              ;SELECT HR'S UNITS DIGIT
        CALL    GETDIG              ;SEND IT TO TERMINAL
                                    ;
HRS3    MVI     A,':'               ;LOAD ":" SEPERATOR
        CALL    SENDIT              ;SEND IT TO TERMINAL
                                    ;
MIN1    MVI     A,MINT              ;SELECT MINUTES TENS DIGIT
        CALL    GETDIG              ;SEND IT TO TERMINAL
                                    ;
MIN2    MVI     A,MINU              ;SELECT MINUTES UNITS DIGIT
        CALL    GETDIG              ;SEND IT TO TERMINAL
                                    ;
MIN3    MVI     A,':'               ;LOAD ":" SEPERATOR
        CALL    SENDIT              ;SEND IT TO TERMINAL
                                    ;
SEC1    MVI     A,SECT              ;SELECT SECONDS TENS DIGIT
        CALL    GETDIG              ;SEND IT TO TERMINAL
                                    ;
SEC2    MVI     A,SECU              ;SELECT SECONDS UNITS DIGIT
        CALL    GETDIG              ;SEND IT TO TERMINAL
                                    ;
```

Continued on next page.

Continued from previous page.

```
MON1    MVI     A,MONT          ;SELECT MONTH TENS DIGIT
        CALL    GETDIG1         ;GET DIGIT
        ANI     3               ;ELIMINATE EXTRA BIT
        LXI     H,STR5          ;POINT TO OFFSET LOOKUP TABLE
        CPI     1               ;SEE IF IT IS A MONTH AFTER SEPT.
        CZ      LAB1            ;IF IT IS A ONE THEN CALL
                                ;
MON2    MVI     A,MONU          ;SELECT MONTH UNITS DIGIT
        CALL    GETDIG1         ;
        DCR     A               ;DECR. MONTH VALUE BY ONE JAN.=Ø THEN
        LXI     B,Ø             ;CLEAR B&C REGISTERS
        MOV     C,A             ;PUT CLOCK DIGIT IN C REGISTER
        DAD     B               ;ADD B&C TO H&L REGISTERS
        MOV     C,M             ;PUT OFFSET NUMBER IN C REGISTER
        LXI     H,STR6          ;LOAD MONTH NAME POINTER
        DAD     B               ;ADD OFFSET TO H&L REGISTERS
        CALL    MESAG           ;SEND MESSAGE
                                ;
DAY1    MVI     A,DAYT          ;SELECT DAY TENS DIGIT
        CALL    GETDIG1         ;
        CPI     Ø               ;TEST FOR ZERO
        JZ      DAY2            ;JUMP TO DAY IF ZERO
        PUSH    PSW             ;SAVE RESULT ON STACK
        ADI     ASCII           ;MAKE IT ASCII
        MOV     C,A             ;PUT IN C REGISTER
        CALL    CONOUT          ;SEND IT
        POP     PSW             ;RESTORE A REGISTER
                                ;
DAY2    MVI     A,DAYU          ;SELECT DAY UNITS DIGIT
        CALL    GETDIG          ;SEND IT TO TERMINAL
                                ;
YEAØ    LXI     H,STR7          ;LOAD FIRST TWO DIGITS OF YEAR
        CALL    MESAG           ;SEND IT TO TERMINAL
                                ;
YEA1    MVI     A,YEAT          ;SELECT YEAR TENS DIGIT
        CALL    GETDIG          ;SEND IT TO TERMINAL
                                ;
YEA2    MVI     A,YEAU          ;SELECT YEAR UNITS DIGIT
        CALL    GETDIG          ;SEND IT TO TERMINAL
        RET                     ;
                                ;
```

Continued on next page.

Continued from previous page.

```
                                    ;SEQUENCE TO ENABLE THE CLOCK
TSTART  MVI     A,0F0H              ;
        OUT     PORT1               ;
        MVI     A,0FH               ;
        OUT     PORT                ;
                                    ;
        MVI     A,0F0H              ;LOAD DELAY CONST.
        CALL    DELAY               ;DO SOME DELAY
                                    ;
        MVI     A,0FCH              ;
        OUT     PORT3               ;
        MVI     A,0F4H              ;
        OUT     PORT1               ;
        RET                         ;
                                    ;
                                    ;RETURN CLOCK TO RUN MODE
TIMED   MVI     A,0F8H              ;
        OUT     PORT1               ;
        MVI     A,0FH               ;
        OUT     PORT                ;
        MVI     A,0F8H              ;
        OUT     PORT3               ;
        MVI     A,0FCH              ;
        OUT     PORT1               ;
        MVI     A,0FH               ;
        OUT     PORT                ;
        RET                         ;
                                    ;
MESAG   MOV     A,M                 ;GET CHARACTER FROM MEMORY
        CPI     0                   ;TEST FOR ZERO
        JZ      MSS1                ;IF ZERO EXIT
        CALL    CONOUT              ;SEND CHARACTER
        INX     H                   ;MOVE POINTER TO NEXT CHARACTER
        JMP     MESAG               ;DO IT AGAIN UNTILL DONE
MSS1    RET                         ;RETURN
                                    ;
GETDIG  CALL    GETDIG1             ;READ CLOCK
SENDCH  PUSH    PSW                 ;SAVE FLAGS AND ACC.
        ADI     ASCII               ;MAKE IT ASCII
        MOV     C,A                 ;PUT IN ACC
        CALL    CONOUT              ;SEND IT TO TERMINAL
        POP     PSW                 ;RESTORE ACC. AND FLAGS
        RET                         ;RETURN TO CALLER
                                    ;
```

Continued on next page.

Continued from previous page.

```
SENDIT  PUSH    PSW             ;SAVE FLAGS AND ACC.
        MOV     C,A             ;PUT IN ACC
        CALL    CONOUT          ;SEND IT TO TERMINAL
        POP     PSW             ;RESTORE ACC. AND FLAGS
        RET                     ;RETURN TO CALLER
                                ;
GETDIG1 OUT     PORT            ;CODE DIGIT TO SELECT CLOCK ELEMENT
        MVI     A,0FAH          ;DELAY FACTOR
        CALL    DELAY           ;DELAY SOME TIME HERE
        IN      PORT            ;READ CLOCK ELEMENT
        ANI     0F0H            ;DROP CODE PART OF DIGIT
        RRC                     ;MOVE DATA TO LOW NIBBLE
        RRC                     ;
        RRC                     ;
        RRC                     ;NUMBER RETURNED IN ACC.
        RET                     ;
                                ;
LAB1    LXI     D,0             ;CLEAR D&E REGISTERS
        MVI     E,0AH           ;IF THE RESULT WAS OCTOBER OR LATER
        DAD     D               ;ADD X10 OFFSET
        RET                     ;LOAD POINTER OFFSET AND RETURN
                                ;
DELAY:  INR     A               ;
        JNZ     DELAY           ;NOT DONE DELAY SOME MORE
        RET                     ;
                                ;
STR3    DB      00H,08H,10H,19H ;OFFSET LOOKUP TABLE
        DB      24H,2EH,36H     ;FOR DAY OF THE WEEK
                                ;
STR4    DB      'Sunday ',0     ;
        DB      'Monday ',0     ;
        DB      'Tuesday ',0    ;
        DB      'Wednesday ',0  ;
        DB      'Thursday ',0   ;
        DB      'Friday ',0     ;
        DB      'Saturday ',0   ;
                                ;
```

Continued on next page.

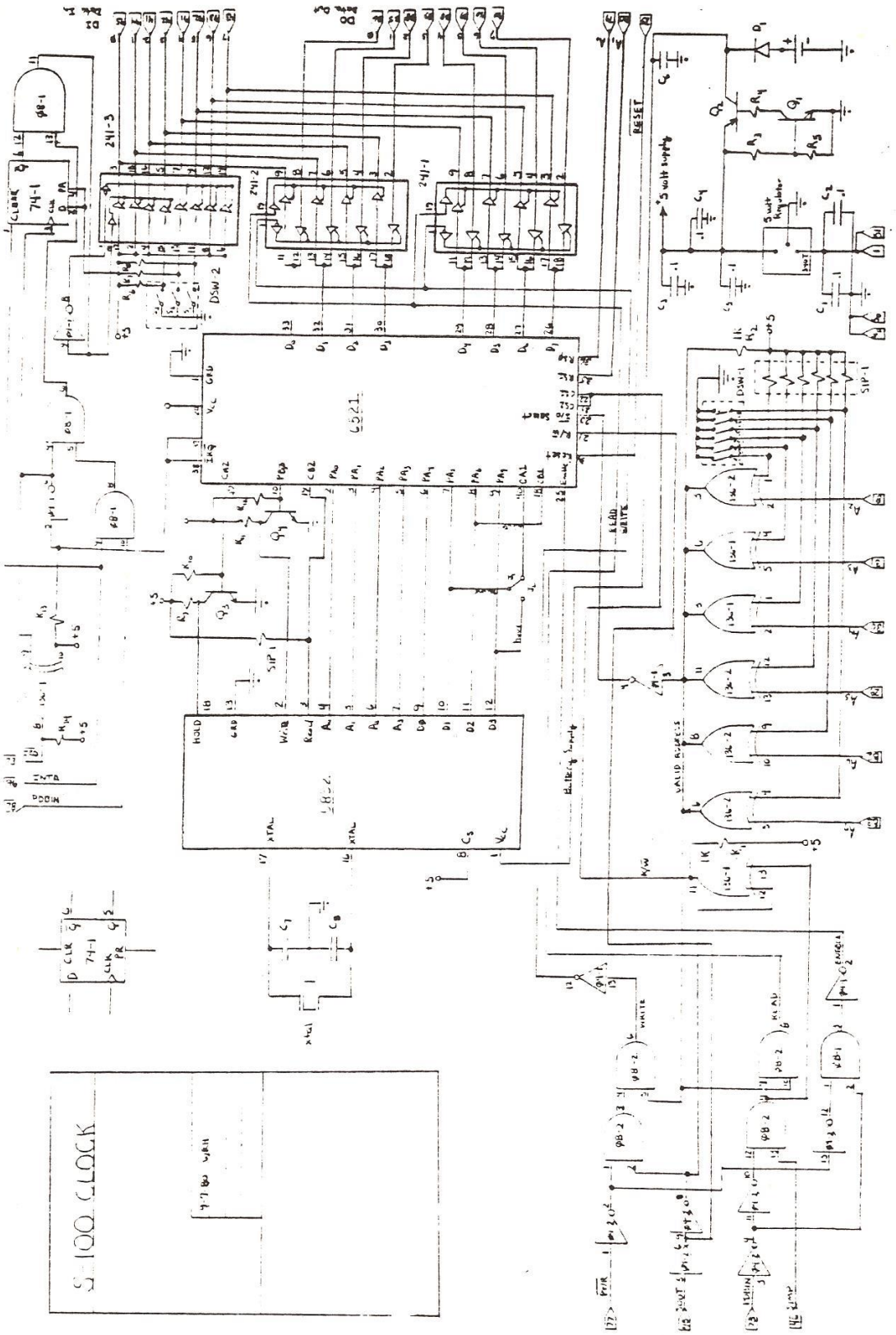
Continued from previous page.

```
STR5    DB      00H,0AH,15H,1DH  ;OFFSET LOOKUP TABLE
        DB      25H,2BH,32H,39H  ;FOR MONTH NAME
        DB      42H,4FH,59H,64H  ;
                                 ;
STR6    DB      ' January ',0    ;
        DB      ' February ',0   ;
        DB      ' March ',0      ;
        DB      ' April ',0      ;
        DB      ' May ',0        ;
        DB      ' June ',0       ;
        DB      ' July ',0       ;
        DB      ' August ',0     ;
        DB      ' September ',0  ;
        DB      ' October ',0    ;
        DB      ' November ',0   ;
        DB      ' December ',0   ;
                                 ;
STR7    DB      ' 19',0          ;YEAR LEADIN MESS.
        END     PROM
```

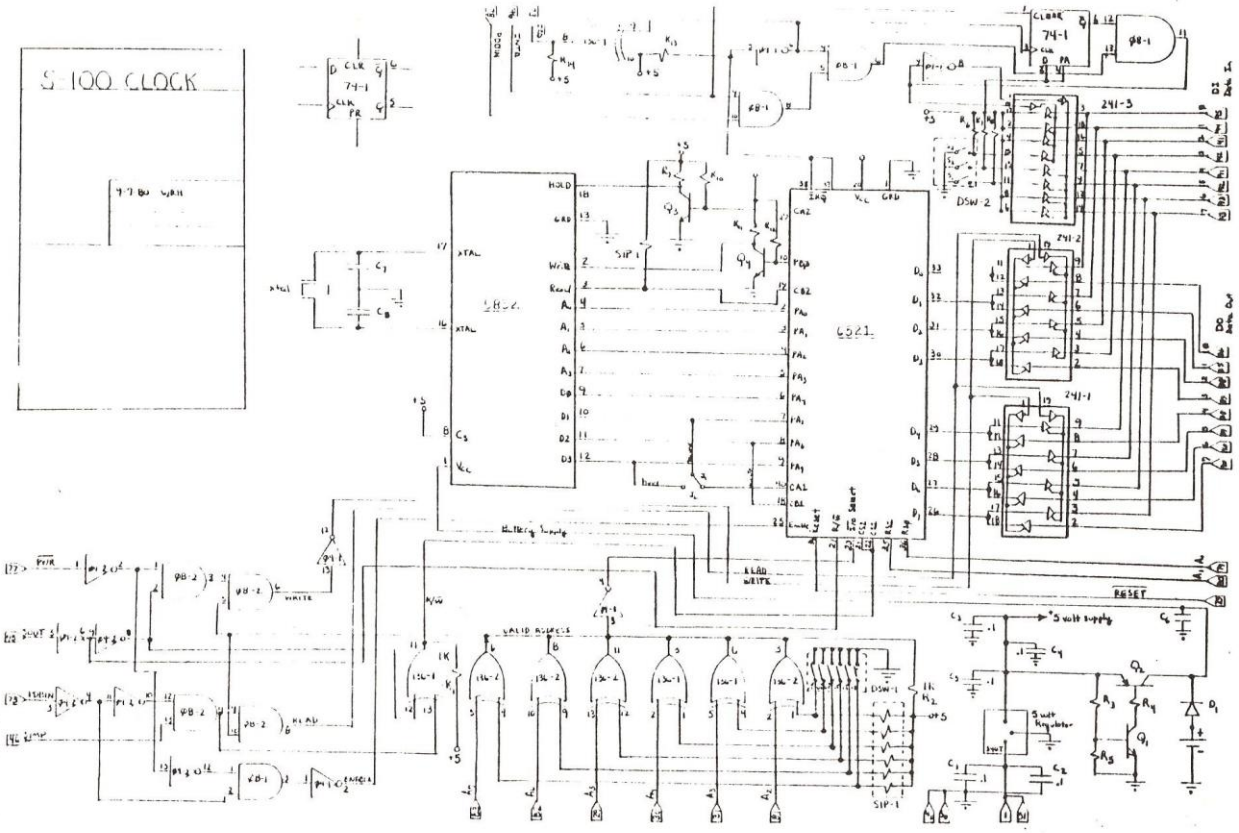
```
10 REM*********RTCSET*****************************
20 REM****NORTH STAR BASIC************************
30 REM PROGRAM TO SET SCITRONICS CLOCK - 801003*
40 REM **COPYRIGHT 1980 - SCITRONICS, INC*******
50 REM****CONV BY RICE COMMUNICATIONS INC*******
60 !"PROGRAM TO SET THE SCITRONICS CLOCK"\DIMT(13)
70 !"SCITRONICS, INC., BETHLEHEM, PA 18015"\!
80 !"ENTER CLOCK BASE PORT ADDRESS"
90 !"IN DECIMAL INTER FORM - 0,4,8,16,20,ETC"
100 !"MAXIMUM VALUE = 252"\INPUT P
110 REM CHECK FOR PROPER VALUE AND RANGE
120 IFP>=256THEN80\M=INT(P/4)\IF(4*M-P)<>0THEN80\P1=P+1\P2=P+2\P3=P+3
130 REM GET THE TIME DATA
140 !\!"ENTER DATE IN THE FORM: YR,MO DA"\!"ONLY LAST TWO DIGITS FOR YR!"
150 INPUT Y,M,D\!\!"ENTER DAY-OF-THE-WEEK - 0=SUNDAY"\INPUTW\!
160 !"ENTER TIME ON 24HR CLOCK IN THE"\!"FORM: HR,MIN"
170 !"CLOCK ALWAYS STARTS ON 00 SECONDS, SO"
180 !"BE SURE TO ENTER THE NEXT MINUTE."\INPUTH,M1\REM PREPARE THE DATE
190 T(1)=16*INT(Y/10)+12\T(2)=16*(Y-10*INT(Y/10))+11\T(3)=16*INT(M/10)+10
200 T(4)=16*(M-10*INT(M/10))+9\T(5)=16*INT(D/10)+8\T(6)=16*(D-10*INT(D/10))+7
210 T(7)=16*W+6\T(8)=16*INT(H/10)+5+128\T(9)=16*(H-10*INT(H/10))+4
220 T(10)=16*INT(M1/10)+3\T(11)=16*(M1-10*INT(M1/10))+2\T(12)=1\T(13)=0
230 REM DATA IS READY, ASK FOR START SIGNAL FROM USER
240 !\!"ENTER 1 AND CARRIAGE RETURN TO SET CLOCK"
250 !"PUSH CR ON THE ZERO SECOND"\INPUTJ\REM THIS IS TO SET SEQUENCE
260 OUTP3,244\OUTP2,255\OUTP3,240\OUTP2,255\OUTP1,240\OUTP,255\OUTP3,244
270 OUTP1,244\FORX=1TO13\OUTP,T(X)\OUTP2,0\OUTP2,1\NEXTX\OUTP1,248\OUTP3,240
280 !"THE CLOCK IS SET"\CHAIN"RTCREAD"
```

```
10 REM **********RTCREAD**************************
20 REM THIS PROGRAM IS IN NORTH STAR BASIC********
30 REM ****CONV BY RICE COMMUNICATIONS INC********
40 REM ** READ SCITRONICS REAL-TIME CLOCK********
50 REM *COPYRIGHT 1980 - SCITRONICS, INC.********
60 REM ******PRESS (ESC) TO STOP READ OUT********
70 DIM T(13),W$(21)\W$="SUNMONTUEWEDTHRFRISAT"
80 !"PROGRAM TO DISPLAY CLOCK DATA"
90 !"SCITRONICS, INC., BETHLEHEM, PA"\!
100 REM THIS IS THE NON INTERRUPT-DRIVEN VERSION
110 !"ENTER CLOCK BASE PORT ADDRESS"
120 !"IN DECIMAL INTER FORM - 0,4,8,12,16,20,ETC"
130 !"MAXIMUM VALUE = 252"\INPUT P
140 REM CHECK FOR PROPER VALUE AND RANGE
150 IFP>=256THEN80\M=INT(P/4)\IF(4*M-P)<>0THEN80\P1=P+1\P2=P+2\P3=P+3
160 IFINP(2)=155THEN80\GOSUB200\REM DISPLAY LOOP STARTS HERE
170 Y=T(1)*10+T(2)\M0=T(3)*10+T(4)\D=T(5)*10+T(6)\W=T(7)+1\H=T(8)*10+T(9)
180 M1=T(10)*10+T(11)\S=T(12)*10+T(13)
190 !Y,":",M0,":",D," ",\!W$(W*3-2,W*3),". ",\!H,":",M1,":",S," "\!\GOTO160
200 REM*****************************
210 REM THIS READS THE CLOCK DATE *
220 REM*****************************
230 OUT P1,240\OUTP,15\OUTP3,252\OUTP1,244\FORX=1TO13\OUTP,13-X
240 T(X)=INP(P)\NEXT X
250 REM*****************************************
260 REM PUT CLOCK BACK INTO RUNNING MODE********
270 REM*****************************************
280 OUT P1,248\OUTP,15\OUTP3,248\OUTP1,252\OUTP,15\FORX=1TO13
290 T(X)=INT(T(X)/16)\NEXT X\T(3)=T(3)AND3\T(8)=T(8)AND3\RETURN
```

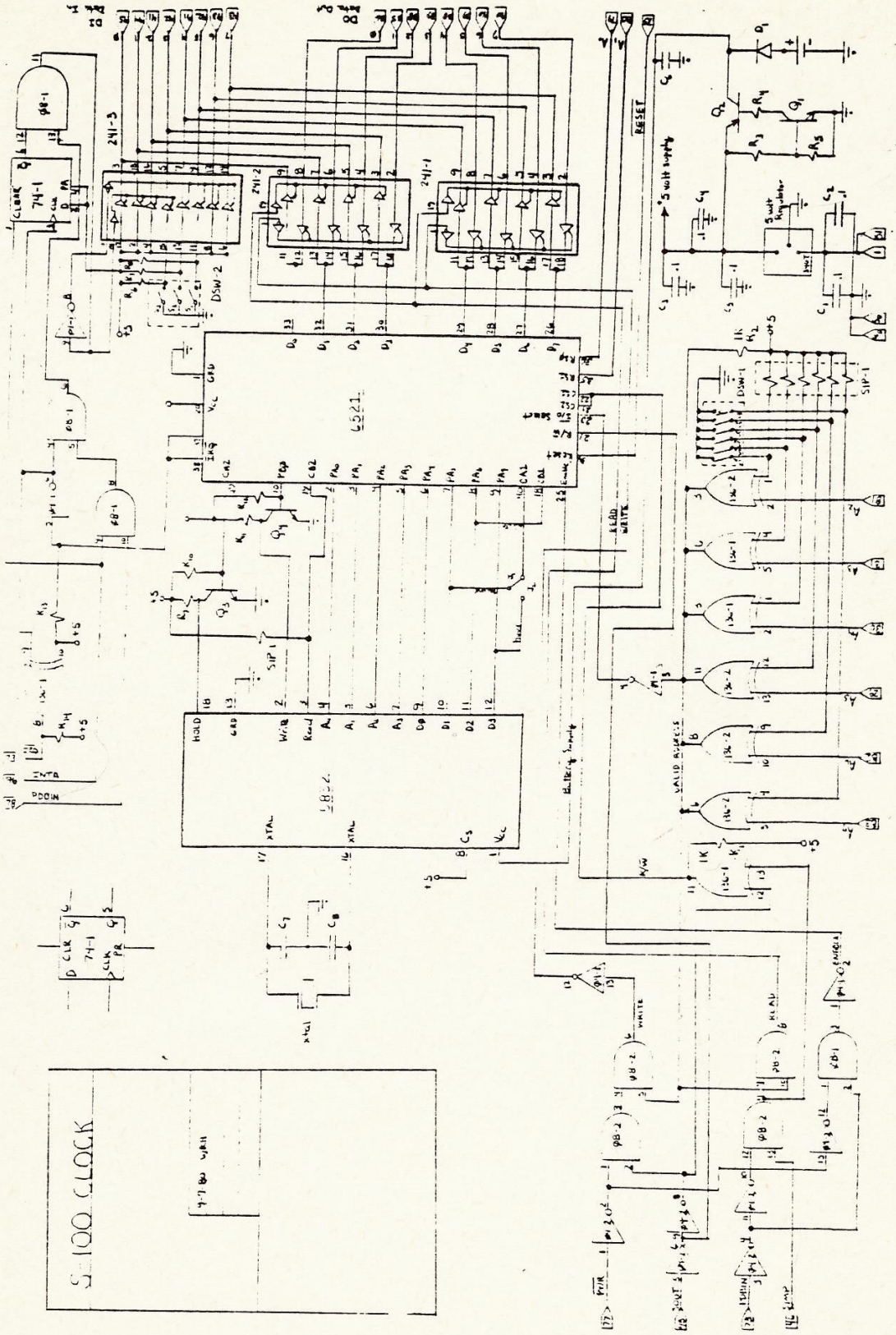
SCHEMATIC
&
PARTS LIST



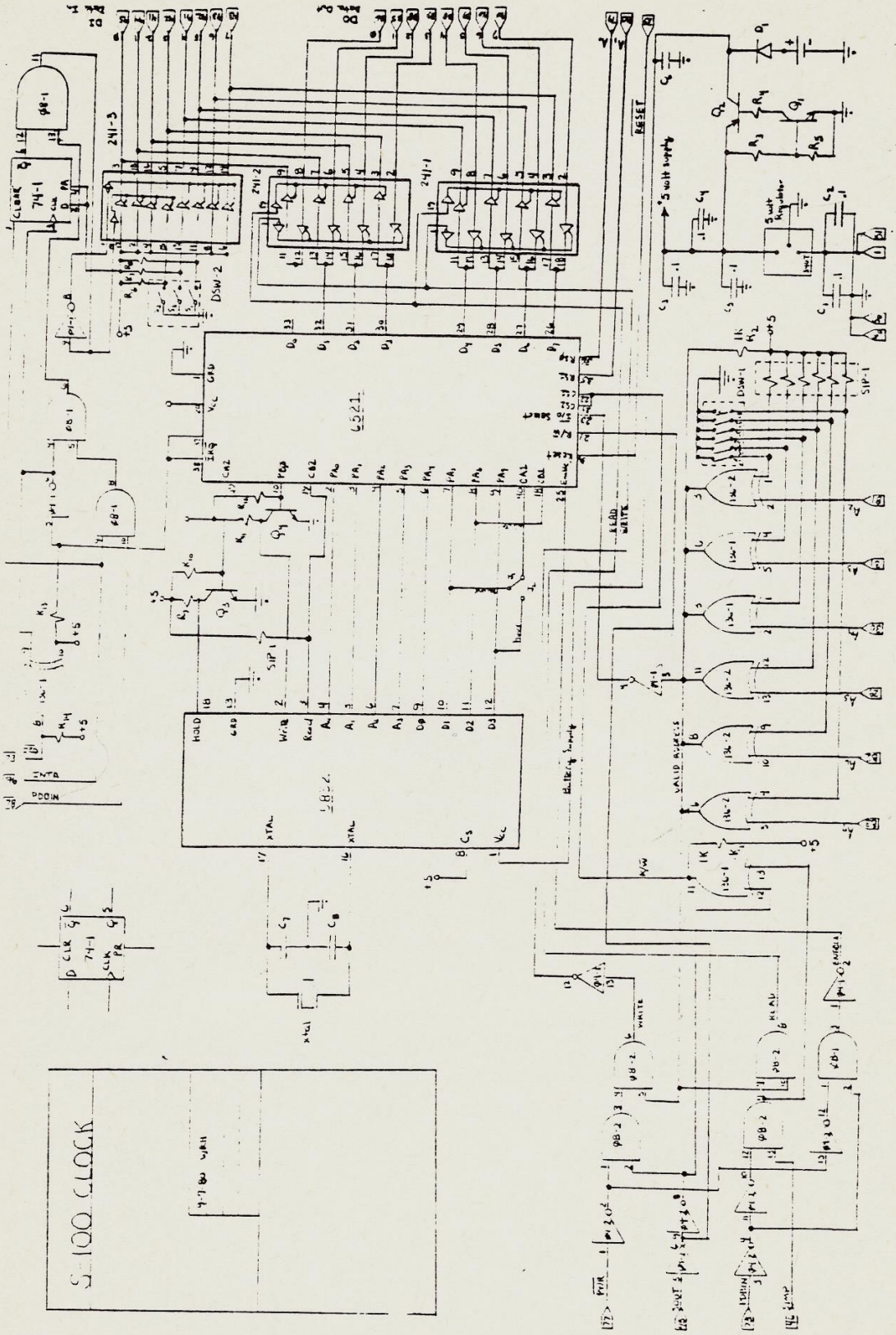
RTC-100 Schematic

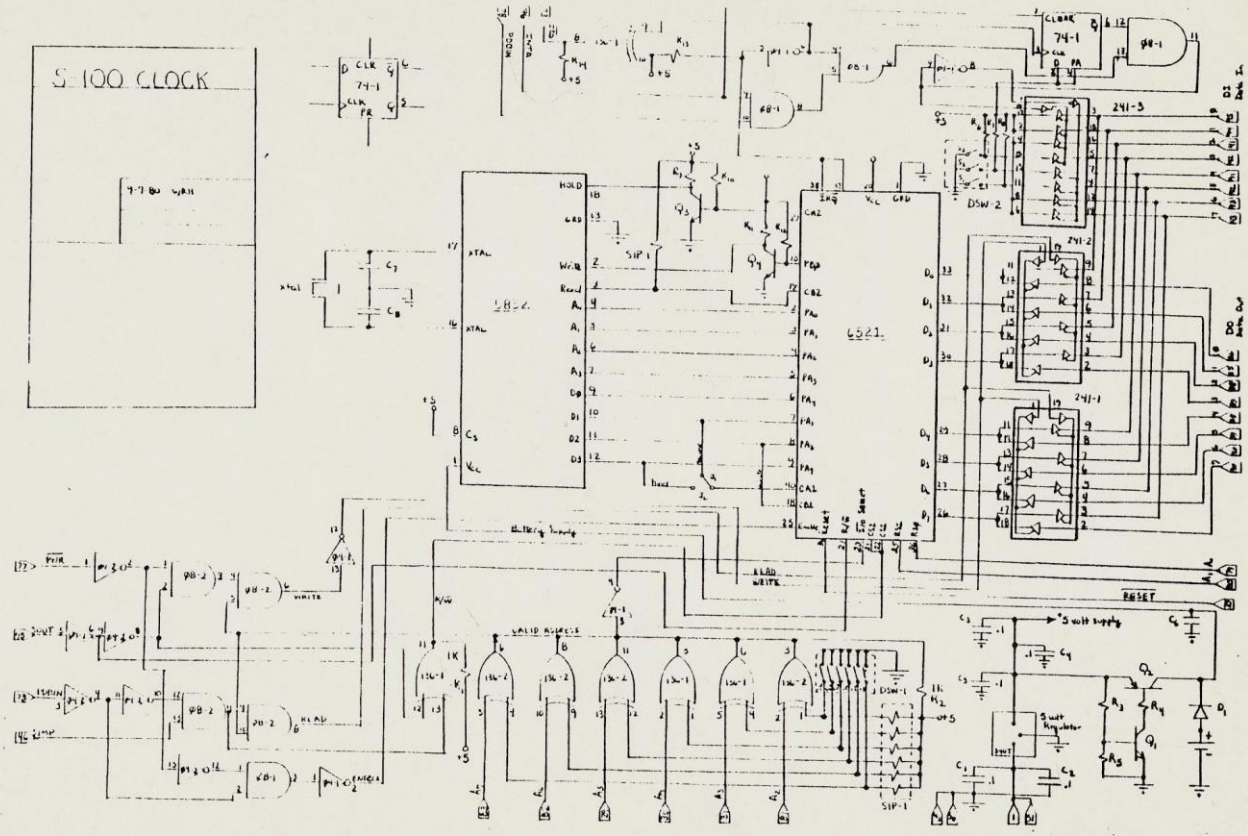


RTC-100 Schematic

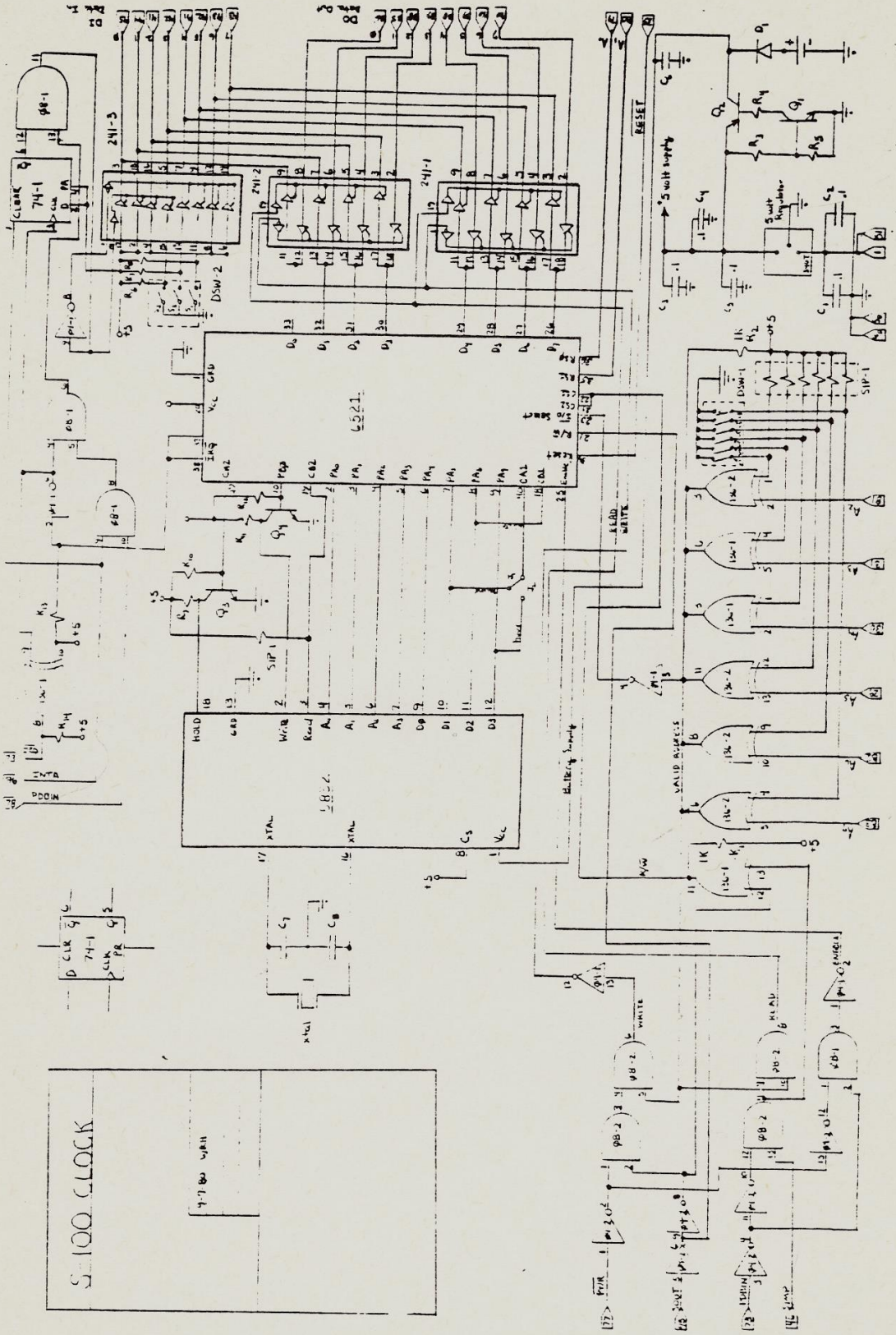


RTC-100 Schematic

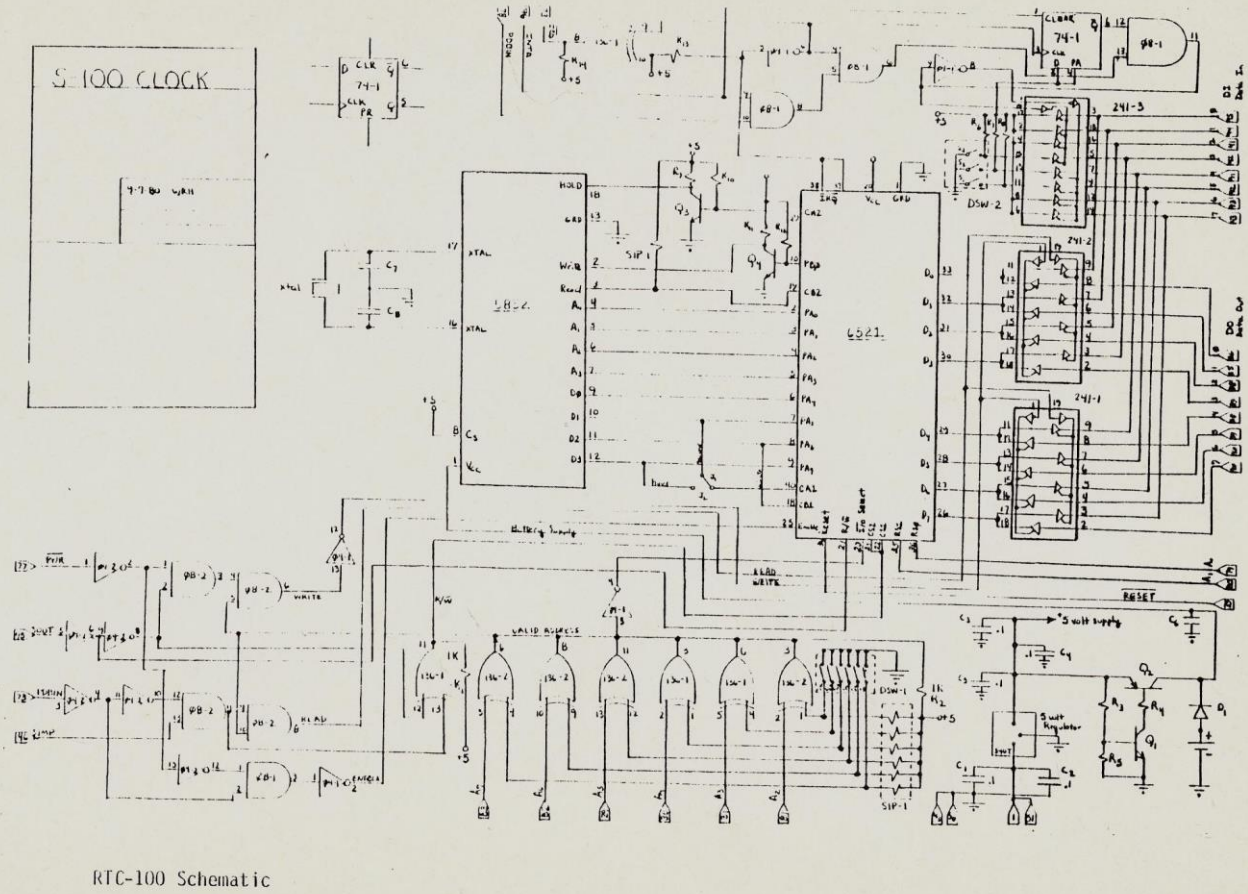

S-100 CLOCK
RESET
6521
74-1
241-3
241-2
241-1
DSW 2
SIP-1
HOLD
R/W
PDOIN
XTAL
RTC-100 Schematic



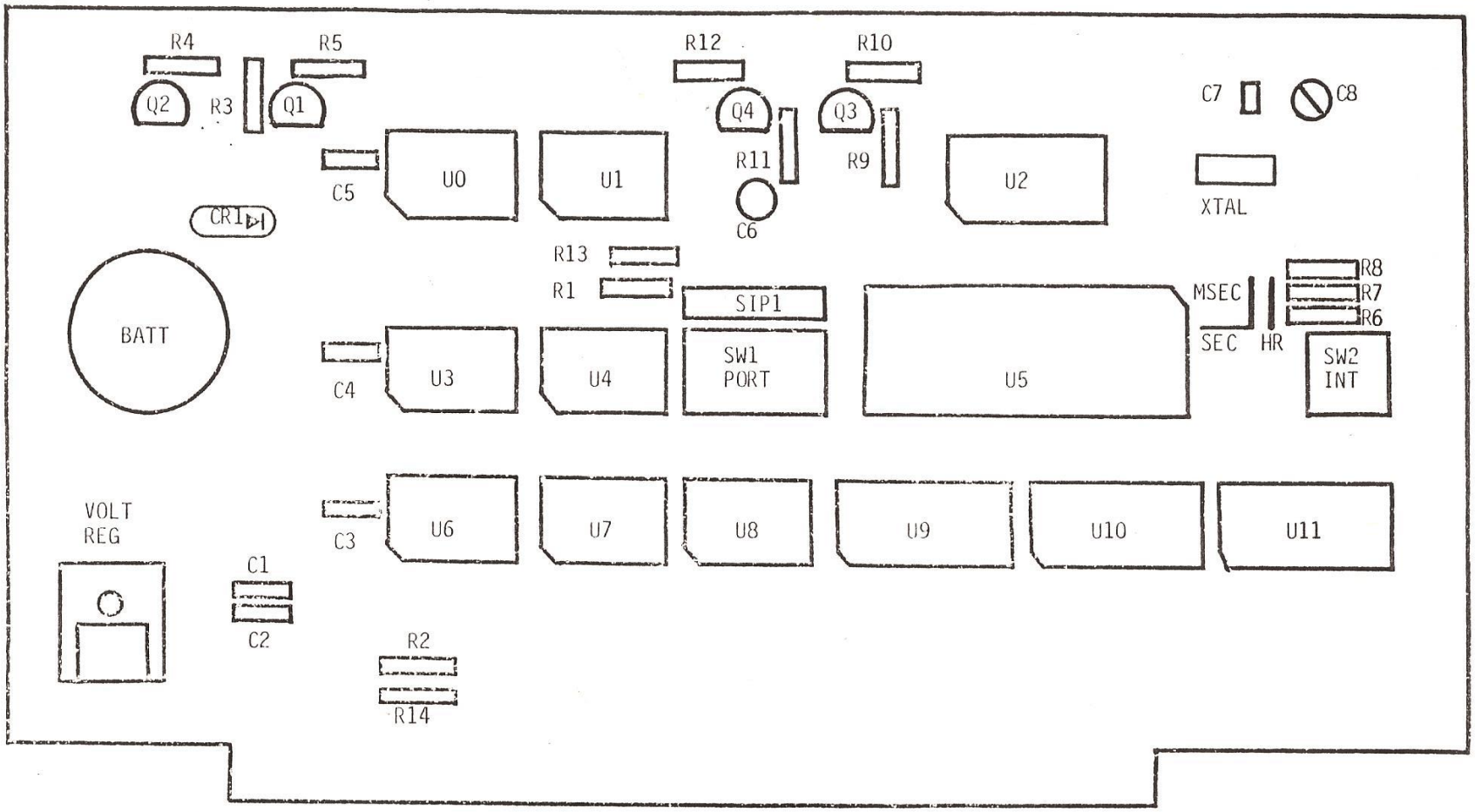
RTC-100 Schematic



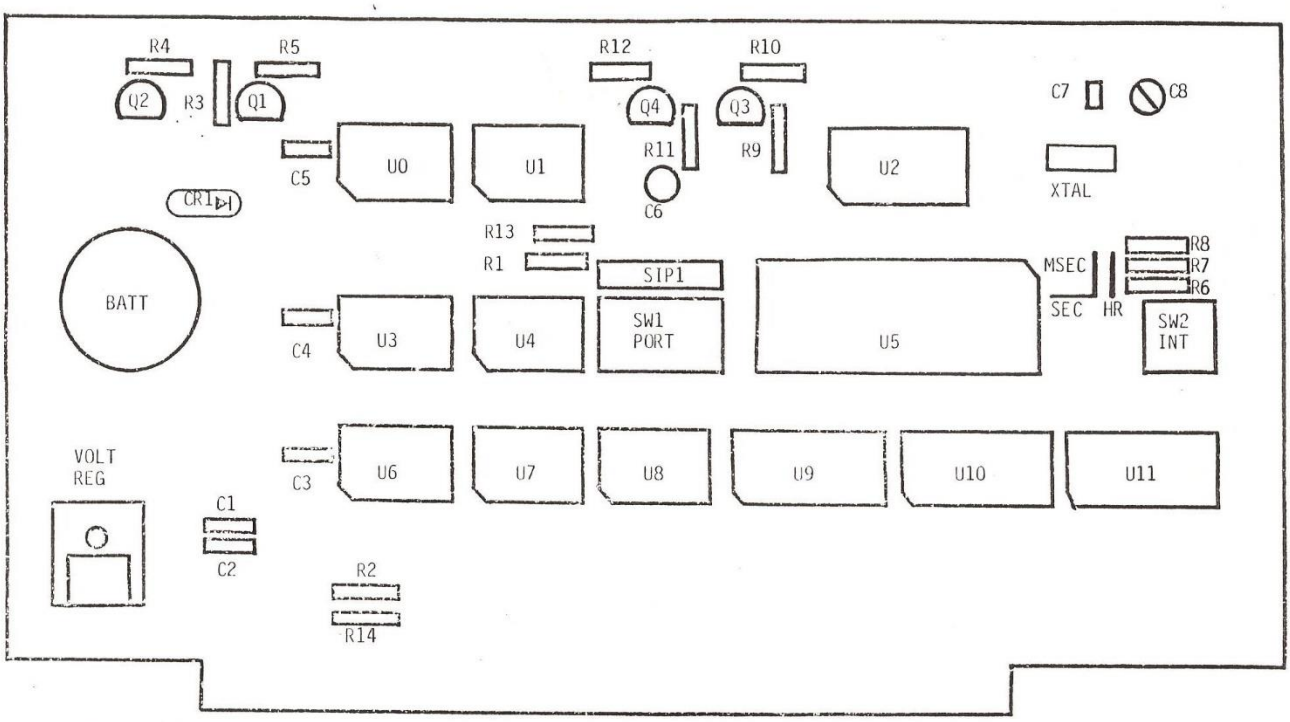
RTC-100 Schematic



RTC-100 Schematic



RTC-100 PARTS LAYOUT: Refer to Parts List for parts description



RTC-100 PARTS LAYOUT: Refer to Parts List for parts description

PARTS LIST --

PRODUCT: RTC-100

LAYOUT #	SCITRONICS P/N	DESCRIPTION	
B 1	17031	3 V Lithium Battery	
C 1	12024	.1mF @ 12V Ceramic Disc Capacitor	
C 2	12024	.1mF @ 12V Ceramic Disc Capacitor	
C 3	12024	.1mF @ 12V Ceramic Disc Capacitor	
C 4	12024	.1mF @ 12V Ceramic Disc Capacitor	
C 5	12024	.1mF @ 12V Ceramic Disc Capacitor	
C 6	12014	4.7mF @ 10V Solid Tantallum Capacitor	
C 7	12031	15pF Ceramic Disc Capacitor	
C 8	12031	15pF Ceramic Disc Capacitor	
C 9	12032	5-30pF Variable Ceramic Capacitor	
CR 1	13011	IN6263 Diode	
D 1	20079	8 Position DIP Switch	
D 2	20084	4 Position DIP Switch	
DS 0	17012	14 Pin DIP Socket	
DS 1	17012	14 Pin DIP Socket	
DS 2	17032	18 Pin DIP Socket	
DS 3	17012	14 Pin DIP Socket	
DS 4	17012	14 Pin DIP Socket	
DS 5	17034	40 Pin DIP Socket	
DS 6	17012	14 Pin DIP Socket	
DS 7	17012	14 Pin DIP Socket	
DS 8	17012	14 Pin DIP Socket	
DS 9	17033	20 Pin DIP Socket	
DS 10	17033	20 Pin DIP Socket	
DS 11	17033	20 Pin DIP Socket	

PARTS LIST --

PRODUCT: RTC-100

LAYOUT #	SCITRONICS P/N	DESCRIPTION	
HS 1	17030	Heat Sink for Voltage Regulator	
J 1	22010	Jumper (seconds)	
Q 1	18001	2N4124 Transistor	
Q 2	18003	2N4126 Transistor	
Q 3	18006	MPSA13 Transistor	
Q 4	18006	MPSA13 Transistor	
VR 1	14041	+5 V Voltage Regulator	
R 1	11013	1K ¼w Carbon Film Resistor	
R 2	11013	1K ¼w Carbon Film Resistor	
R 3	11022	51K ¼w Carbon Film Resistor	
R 4	11001	10K ¼w Carbon Film Resistor	
R 5	11001	10K ¼w Carbon Film Resistor	
R 6	11028	3.3K ¼w Carbon Film Resistor	
R 7	11028	3.3K ¼w Carbon Film Resistor	
R 8	11028	3.3K ¼w Carbon Film Resistor	
R 9	11012	20K ¼w Carbon Film Resistor	
R 10	11012	20K ¼w Carbon Film Resistor	
R 11	11012	20K ¼w Carbon Film Resistor	
R 12	11012	20K ¼w Carbon Film Resistor	
R 13	11028	3.3K ¼w Carbon Film Resistor	
R 14	11028	3.3K ¼w Carbon Film Resistor	
SR 1	11033	3.3K SIP Resistor Network	
U 0	14047	7474 Dual D Flip Flop	
U 1	14046	7408 Quad 2-in AND Gate	
U 2	14045	MSM5832 OKI Clock Chip	

PARTS LIST --

PRODUCT: RTC-100

LAYOUT #	SCITRONICS P/N	DESCRIPTION	
U 3	14046	7408 Quad 2-in AND Gate	
U 4	14030	74LS04 Hex Inverter	
U 5	14048	6821 Peripheral Interface Adaptor	
U 6	14030	74LS04 Hex Inverter	
U 7	14049	74LS136 Quad Exclusive OR Gate (O.C.)	
U 8	14049	74LS136 Quad Exclusive OR Gate (O.C.)	
U 9	14050	74LS241 Octal Bus/Line Driver	
U 10	14050	74LS241 Octal Bus/Line Driver	
U 11	14050	74LS241 Octal Bus/Line Driver	
HW 1	21001	6-32 ¼" Screw	
HW 2	21030	6-32 Hex Nut	
PC 1	17012	S-100 Real Time Clock PC Board	
X 1	17035	Crystal (32.768KHz)	

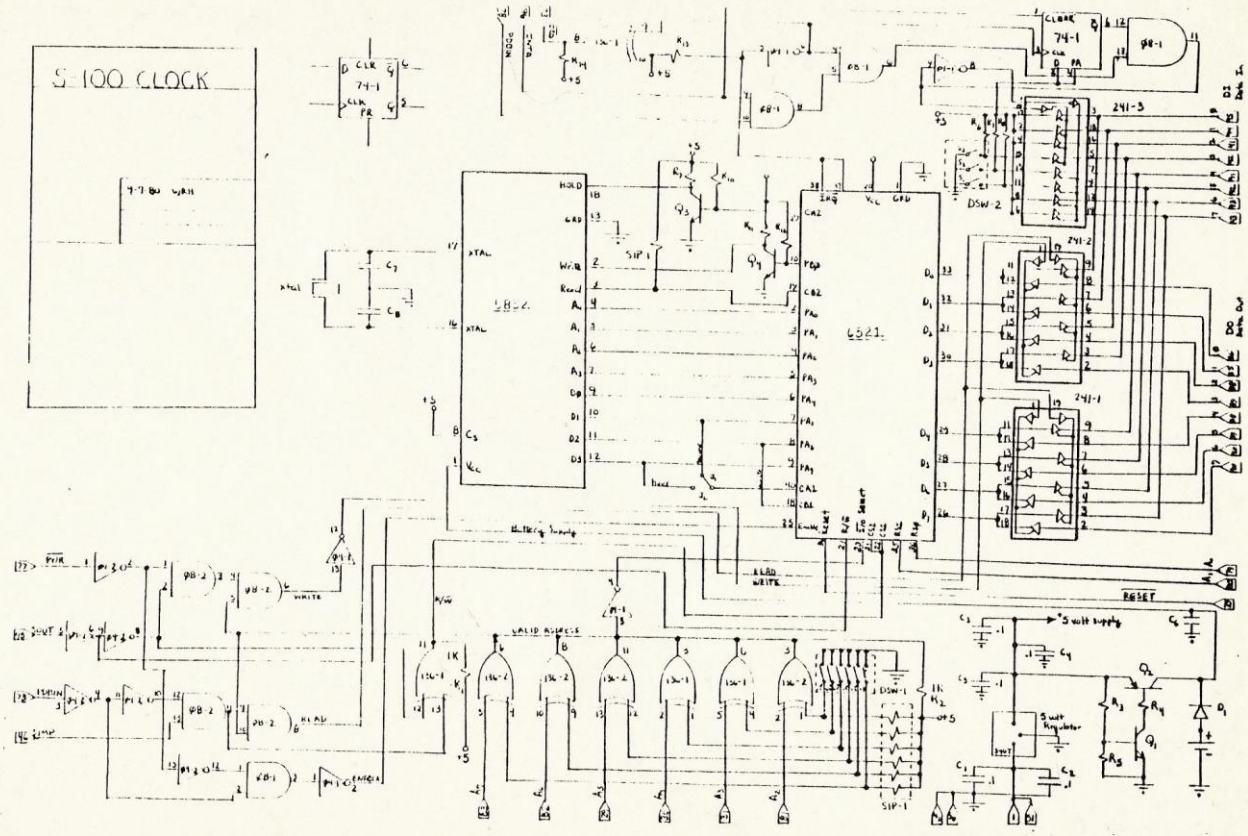
WARRANTY

All SciTronics Inc. computer boards are assembled and fully tested at the factory before shipment. Your Real Time Clock is warranted against defects in material and workmanship for a period of six (6) months from date of delivery. SciTronics will repair or replace products found to be defective during the warranty period, provided they are returned to SciTronics Inc. No other warranty is expressed or implied. We are not liable for improper use or inconsequential damages. We reserve the right to refuse to repair any product that in our opinion has been subjected to abnormal electrical or mechanical abuse. Products out of warranty are subject to a minimum service fee. Contact SciTronics before sending your unit in for repair. Please fill out the enclosed self-addressed warranty card to validate your warranty.

For your records:

Date Purchased _____

Serial # _____
(located on board)



RTC-100 Schematic