59309A Digital Clock

HP-IB/HP 1000 Programming Example

HEWLETT PACKARD
HP-IB
DATA SYSTEMS DIVISION

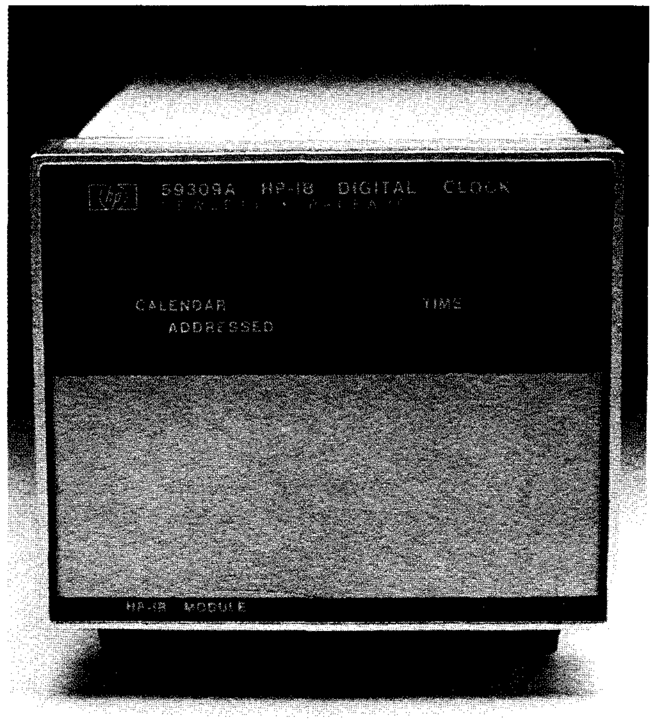
Application Note 401-8

Device Introduction

The HP 59309A Digital Clock provides a front-panel display of the date and time on a 24-hour basis. When used in a system, the 59309A is fully programmable and can output the date and time onto the HP-IB for printout or other systems use. The display is a row of digits (figure 8-1). The year is not included with the time and date information.

The 59309A can be used as a stand-alone digital clock, or as a system time-of-day source. For example, although the RTE operating system has its own timekeeping mechanism, its memory is volatile and the time must be reset using an operator request when the system is rebooted. The 59309A digital clock can operate from its own standby battery to maintain continuous time in the event of a power failure. User programs can be written which obtain the current date and time from the 59309A at RTE boot up. Similiarly, the 59309A can be set from an interactive program in the HP 1000.

This note should be used in conjunction with the 59309A Operating and Service Manual (59309A-90004) and Application Note 401-1 (5953-2800).



Addressing

Setting the 59309A address is straightforward. First, the left two switches are always set "up" for use with the HP 1000. The remaining five switches are set to the binary device address. A typical setting is shown in figure 8-2.

	Month	Day of Month	Hour	Minute	Second
digits:	MM	DD	HH	MM	SS
example:	01	25	09	54	26

Figure 8-1. 59309A Output Format

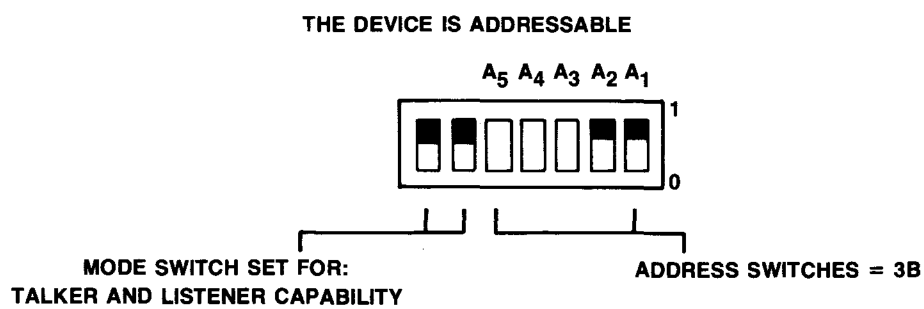


Figure 8-2. Typical Address Settings for the 59309A

Bus Output Format Selection

The 59309A contains a switch assembly (Figure 8-3) near the top edge of board A5 (the 59309A must be opened) to provide selection of various output formats to suit different applications.

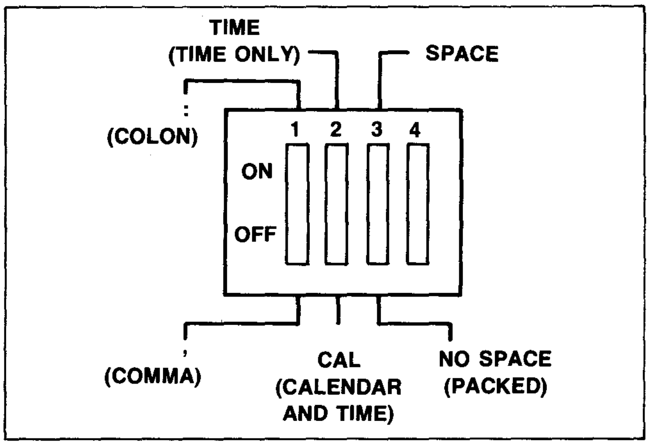


Figure 8-3. 59309A Switch Assembly

The settings of the switch assembly allow selection of delimiters or no delimiters, colons or commas, calendar and time of day or just time of day. The output formats in figure 8-4 may be selected.

For example, time is usually delimited by colons to provide a legible printout for most people. Depending on the programming task, the use of commas for formatting, or numerical computations may be more convenient.

Output format is a significant consideration for the programmer. Software library subroutines in the HP 1000 are designed to typically parse (pick out ASCII numeric values between) commas or colons, so usually, one of these two output formats should be selected.

Two relocatable library subroutines are available which will facilitate parsing of the input string:

1. "PARSE" parses information delimited by commas, and converts it to binary.[1]

SPACE (not packed)

TIME (Time Only)

: (Colon)	(Status)*	SP	1	1	:	2	3	:	1	4	CR	LF
, (Comma)	(Status)*	SP	1	1	,	2	3	,	1	4	CR	LF

CAL (Calendar and Time)

: (Colon)	(Status)*	SP	1	2	:	2	8	:	1	1	:	2	3	:	1	4	CR	LF
, (Comma)	(Status)*	SP	1	2	,	2	8	,	1	1	,	2	3	,	1	4	CR	LF

NO SPACE (packed)

TIME (Time Only)

: (Colon)	(Status)*	SP	1	1	2	3	1	4	CR	LF
, (Comma)	(Status)*	SP	1	1	2	3	1	4	CR	LF

CAL (Calendar and Time)

: (Colon)	(Status)*	SP	1	2	2	8	1	1	2	3	1	4	CR	LF
, (Comma)	(Status)*	SP	1	2	2	8	1	1	2	3	1	4	CR	LF

*The ASCII character in this position of the data output string will be either ? or SP depending on the error status.

Figure 8-4. 59309A Output Format Selection Switches

[1] *Available from the RTE Relocatable Library manual (24998-90001).*

2. "NAMR" parses information delimited by commas, having subparameters delimited by colons.[1] For example,

```
A:B:C,F:G:H,M:N:R
```

will be parsed into three groups,

```
A:B:C
F:G:H
M:N:R
```

and each of the subparameters will be converted to a binary value when it is applicable.

These parse subroutines allow the user program to put the hours, minutes, seconds, etc., each into separate variables so that they may later be manipulated by the user program.

System Preparations

LU Assignment

One LU number is needed for the 59309A clock. As shown under "Addressing," a typical clock address is "00 011" binary or 3 octal. If LU 30 were to be assigned to the clock, the operator command would be,

```
:SYLU,30,10,3B
```

assuming equipment table 10 represented the bus.

Buffering

Buffering may be used once the device has been checked out, but the EQT should be unbuffered during initial setup. The request can be made from File Manager,

```
:SYEQ,10,UN
```

to unbuffer EQT 10.

Time-out

Time-out usually means that something is wrong with the 59309A or the bus. Device problems are so infrequent with the 59309A, that errors can usually be handled by the operating system.

Configuration

The device configuration word defaults to the correct mode when DMA is not allocated. Generally, system operation is more efficient when DMA is not allocated for the 59309A. All other configuration bits default to the correct mode.

Remote

The bus should be in remote for 59309A programming.

Programming

Programming the 59309A is straight forward. The clock accepts ASCII command characters which program it to start, stop, increment and reset. The current time and date is also returned in ASCII. To set the clock it must first be reset (1:0:0:0:0), and then incremented to the proper date and time. See the command codes in Table 8-1 for more details.

A complete clock programming scheme is shown in Figure 8-5. Most of the details concerning clock programming can be found in the FORTRAN program.

The 59309A is best read using the method shown (figure 8-5) on line 90. The time values are read into an ASCII input buffer "INBUF". Note that the 59309A hardware is set up to delimit ASCII time data with commas so that the subroutine "PARSE" can put numeric time values into IPBUF. (See the EQUIVALENCE statement for IPBUF at the beginning of the program.)

In some cases, the system level subroutines 'REIO' and "EXEC" are more efficient for ASCII data than using the formatter, since ASCII information which has been input to the HP 1000 is normally displayed on a line printer or CRT terminal. (No conversion to binary is required.)

Notice that the computer handles the date in terms of the Julian date while the clock tracks each month and day. The subroutine 'IDAYS' adjusts the Julian date in accordance with the year. The year must be obtained from the run parameters or from the computer clock (if the command is 'SC').

Table 8-1. 59309A Programming Codes

Function	ASCII Character	Octal Code	Binary DIO Lines						
			7	6	5	4	3	2	1
Resets the clock to: 01:01:00:00:00 and clears output register	R	122	1	0	1	0	0	1	0
Stops the clock	P	120	1	0	1	0	0	0	0
Starts the clock	T	124	1	0	1	0	1	0	0
Updates the counting chain 1 second (for more than 1 up-date repeat entry desired times)	S	123	1	0	1	0	0	1	1
Updates the counting chain 1 minute (for more than 1 min. repeat entry desired times)	M	115	1	0	0	1	1	0	1
Updates the counting chain 1 hour (for more than 1 hr. repeat entry desired times)	H	110	1	0	0	1	0	0	0
Updates the counting chain 1 day (for more than 1 day, repeat entry desired times)	D	104	1	0	0	0	1	0	0
Commands the clock to store time value in the output register but does not output it. Time value is output when the 59309A is addressed to talk.	C or BS	103 10	1 0	0 0	0 0	0 1	0 0	1 0	1 0
*Unlisten	?	077	0	1	1	1	1	1	1
*Untalk	—	137	1	0	1	1	1	1	1

*Universal command (effective when ATN is low)

```
FTN4,L
      PROGRAM TIME(3),08-02-78 (GWG) SET THE SYSTEM TIME
C
C :RU,TIME,INPUT,YEAR,59309A LU,COMMAND
C
C COMMAND=0 BUT 59309A LU NONZERO DEFAULTS TO CS AND TERMINATE
C
C IF 59309A LU IS ZERO THE PROGRAM IS INTERACTIVE
C CS        59309A SETS THE SYSTEM TIME
C SC        SYSTEM SET THE 59309A CLOCK
C OS        OPERATOR SETS 59309A CLOCK WHICH SETS THE SYSTEM CLOCK
C OC        OPERATOR SET THE 59309A CLOCK
C
      COMMON IYEAR
      DIMENSION IBUF(14),ILEN(12)
      DIMENSION MESBF(15),IPRM(5),INBF(8),IREG(2),IPBUF(33)
      EQUIVALENCE (REG,IREG,IA),(IREG(2),IB)
      EQUIVALENCE (MONTH,IPBUF(2)),(IDAY,IPBUF(6)),(IHR,IPBUF(10))
      EQUIVALENCE (MIN,IPBUF(14)),(ISEC,IPBUF(18))
C IPMT IS THE PROMPT FLAG
      DATA IPMT/0/
      CALL RMPAR(IPRM)
      ILU=IPRM
      IF(ILU.EQ.0)ILU=1
      IYEAR=IPRM(2)
      IF(IYEAR.EQ.0)IYEAR=1978
      LU=IPRM(3)
      ICM=IPRM(4)
C IF LU WAS SPECIFIED IN THE RUN STATEMENT, NO INTERACTION
      IF(LU.NE.0) GO TO 200
      IF(IPMT.EQ.0)WRITE(ILU,311)
311   FORMAT(" TIME:  ':RU,TIME,INPUT,YEAR,59309A LU,COMMAND'"/,
     &      "         '??' FOR MORE INFO 'EN' TO END."/)
      IPMT=1
      WRITE(ILU,1000)
1000  FORMAT(" TIME: ENTER 59309A LU #: ")
      READ(ILU,*)LU
C     ISSUE PROMPT
 800  WRITE(ILU,1001)
1001  FORMAT("\")
      READ(ILU,1002)ICM
1002  FORMAT(A2)
C OPERATOR SET SYSTEM CLOCK AND 59309A CLOCK
C OPERATOR SET 59309A CLOCK
 200  IF(ICM.NE.2HOS.AND.ICM.NE.2HOC) GO TO 550
 903  WRITE(ILU,501)
 501  FORMAT(" TIME: ENTER DATE AND TIME: MNTH(1-12), DAY",
     &" HOUR(1-24), MIN, SEC")
      READ(ILU,*)MONTH,IDAY,IHR,MIN,ISEC
      IF(IYEAR.GE.0.AND.IDAY.GE.0.AND.IHR.GE.0.AND.MIN.GE.0.AND.
     &ISEC.GE.0.AND.MONTH.GE.0)GO TO 900
 902  WRITE(ILU,901)
```

Figure 8-5. 59309A System Clock Program

```
 901  FORMAT(" TIME: TIME AND DATE VALUES OUT OF BOUNDS!")
      GO TO 903
 900  IF(IYEAR.GT.2000.OR.IDAY.GT.366.OR.IHR.GT.24.OR.MIN.GT.60.OR.
     &ISEC.GT.60.OR.MONTH.GT.12)GO TO 902
C FIGURE OUT THE JULIAN DATE AND LEAP YEAR POSSIBILITIES
      CALL IDAYS(MONTH,IDAY)
C RESET AND STOP THE CLOCK
 552  WRITE(LU,503)
 503  FORMAT("RP")
      IF(IDAY.EQ.1)GO TO 520
C UPDATE THE MONTH AND DAY
      DO 502 I=1,IDAY-1
 502  WRITE(LU,504)
 504  FORMAT("D")
 520  IF(IHR.EQ.0)GO TO 521
C UPDATE THE HOUR
      DO 505 I=1,IHR
 505  WRITE(LU,506)
 506  FORMAT("H")
 521  IF(MIN.EQ.0)GO TO 522
C UPDATE THE MINUTE
      DO 507 I=1,MIN
 507  WRITE(LU,508)
 508  FORMAT("M")
 522  IF(ISEC.EQ.0)GO TO 523
C UPDATE THE SECOND
      DO 509 I=1,ISEC
 509  WRITE(LU,510)
 510  FORMAT("S")
C START UP THE CLOCK
 523  WRITE(LU,511)
 511  FORMAT("T")
C NOW THE THE 59309A IS SET GO SET THE SYSTEM CLOCK
      IF(ICM.EQ.2HOS) ICM=2HCS
C 59309A SETS THE SYSTEM TIME
 550  IF(ICM.NE.0.AND.ICM.NE.2HCS) GO TO 600
C NOTE: ICM=0 AT RUN TIME MEANS '\CS'!
      REG=REIO(1,LU,INBF,100)
      INBUF=IAND(377B,INBUF)
      CALL PARSE(INBF,IB*2,IPBUF)
      CALL IDAYS(MONTH,IDAY)
      CALL CODE
      WRITE (MESBF,50) IYEAR,IDAY,IHR,MIN,ISEC
  50  FORMAT ("TM,"I6",","I4",","I2",","I2",","I2)
      IERR=MESSS(MESBF,23,ILU)
      IF (IERR.EQ.0) GO TO 5
      GO TO 999
   5  CALL TODAY(IBUF)
      CALL EXEC(2,ILU,IBUF,-25)
C SYSTEM SETS 59309A TIME
 600  IF(ICM.NE.2HSC) GO TO 650
      MESBF=2HTI
      LNTH=MESSS(MESBF,2,ILU)
 551  MESBF(3)=IAND(MESBF(3),377B)+26000B
      MESBF(5)=IAND(MESBF(5),377B)+26000B
```

Figure 8-5. 59309A System Clock Program (Continued)

```
      MESBF(7)=IAND(MESBF(7),377B)+26000B
      MESBF(9)=IAND(MESBF(9),377B)+26000B
      CALL PARSE(MESBF,20,IPBUF)
      ICM=2HDD
      GO TO 552
C USER INFO
 650  IF(ICM.NE.2H??)GO TO 750
      WRITE(ILU,9005)
9005  FORMAT(,
     &"                     \CS ... 59309A SETS THE SYSTEM TIME",
     &/,"                    \SC ... SYSTEM SETS 59309A CLOCK",
     &/,"                    \OS ... OPERATOR SETS 59309A AND",
     &" THE SYSTEM TIME",
     &/,"                    \OC ... OPERATOR SETS 59309A CLOCK",
     &/,"                    \EN ... END THIS PROGRAM",
     &/)
 750  IF(IPMT.EQ.0) GO TO 999
C IGNORE INVALID INPUTS
      IF(ICM.NE.2HEN) GO TO 800
  999 END
C
C
C
      SUBROUTINE IDAYS(MONTH,IDAY),08-03-78 (GWG) ADJUST FOR LEAP YEAR
      COMMON IYEAR
      DIMENSION MONBF(24),ILEN(12)
      DATA MONBF/1,2,3,4,5,6,7,8,9,10,11,12/
      DATA ILEN/0,31,28,31,30,31,30,31,31,30,31,30/
      ILEN(3) = 28
      IF (MOD(IYEAR,4).EQ.0) ILEN(3) = 29
      ISUM = 0
      DO 30 I=1,12
      ISUM = ISUM + ILEN(I)
      IF (MONTH.NE.MONBF(I)) GO TO 30
      GO TO 40
   30 CONTINUE
   40 IDAY = ISUM + IDAY
      RETURN
      END
      END$
```

Figure 8-5. 59309A System Clock Program (Continued)

Notice how the subroutine "PARSE" is used in line 110 of figure 8-5. The input buffer from the system message processor is modified to include commas.[1] 'Parse' is then called to convert the ASCII to integer binary numbers. The information is then reformatted and sent back out to the 59309A.

The subroutine 'CODE' allows a FORTRAN 'WRITE' statement to modify a memory buffer (line 94).[1]

The program "TIME" in figure 8-5 can either be copied or obtained from the contributed library.[2]

The subroutine "TODAY" is required, and may be obtained from the contributed library also.[3] Note that when "TIME" is ordered, "TODAY" will be automatically included with the supplied software.

Error Checking

Whenever the 59309A is read, the second character of the input will either be a space or a question mark (?). The question mark occurs when there is an error condition (Operating and Service Manual, 59309-90004). This may be checked in FORTRAN (figure 8-6).

Performance

The 59309A has no service request ability and in most applications, the clock is used in such a way that detailed performance documentation is unnecessary. Performance testing can be conducted, however, using the programs documented in Chapters 4 and 5 of AN 401-1.

```
            REG = REIO (1,LU,INBUF,20)
            IF(IAND(INBF,377B).NE.77B) GO TO 20
            WRITE (ILU,10)
        10  FORMAT("POWER FAIL ON THE 59309A HAS OCCURRED!")
            GO TO 100
   20       CONTINUE
```

Figure 8-6. 59309A Error Checking Example

[2]The contributed library part number for "TIME" is 22683-13307.
[3]The contributed library part number for "TODAY" is 22683-13308.