

PRELIMINARY INSTRUCTION MANUAL
FOR THE INFOTEK SYSTEMS
RT-30 REAL TIME CLOCK

Copyright (c) Infotek Systems

INFOTEK SYSTEMS CORPORATION
1400 North Baxter Street
Anaheim, California 92806

REV. II-August 1979

TABLE OF CONTENTS

	<u>PAGE</u>
INTRODUCTION.....	1
INSPECTION AND UNPACKING.....	1
WARRANTY.....	1
EQUIPMENT SUPPLIED.....	3
SPECIFICATIONS.....	3
INSTALLATION.....	3
DISPLAY	4
BATTERY	4
EXTERNAL 5MHz SIGNAL SOURCE.....	4
OPERATION.....	
SET STATEMENT.....	6
INT STATEMENT.....	9
PIC STATEMENT.....	10
KEY STATEMENT.....	11
ENTER STATEMENT.....	13



INTRODUCTION

The RT-30 Real Time Clock provides the H-P 9830A/B desk-top computer the first interrupt capability on a real time basis. The RT-30 interrupt capability may be on a fixed basis, computed basis, or a preset time. Upon interrupt, the RT-30 will execute up to 32 key strokes which have been previously stored into its buffer. An optional 1,024 key stroke buffer significantly enhances the procedures that can be executed upon interrupt.

The RT-30 accumulates time in days, hours, minutes, seconds, and hundredths of a second. The ENTER statement of the Extended I/O ROM allows data to be easily transferred from the RT-30 to the 9830.

The 9830 user can also specify the precision of the time which is to be output from the RT-30. For example; in applications where the time to be measured is nominally less than a second, the RT-30 can be instructed to suppress days, hours, minutes and output only the seconds and hundredths.

The optional LED display can be ordered as an integral part of the RT-30 or at the end of a 4 foot cable attaching to the front of the 9830 via a self adhesive strip. The display enables a user to view the current RT-30 time. A button on the RT-30 will cause the next interrupt time to be displayed.

INSPECTION AND UNPACKING

The RT-30 Real Time Clock has been thoroughly inspected and tested prior to shipment, and is ready for operation when received. However, before installation and use, the RT-30 must be inspected for any possible damage incurred during transient.

Carefully remove the RT-30 from the shipping carton and inspect the unit for possible damage. If there is any damage file a claim with the carrier, and retain the original packing carton for use in the event that it becomes necessary to reship the unit.

WARRANTY

The RT-30 is fully warranted for one year from date of ship, to be free of defect in materials or workmanship. During the warranty period the RT-30 will be repaired free of charge when returned to the nearest Infotek Repair Depot

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

in the original shipping carton. The customer is responsible for shipping and insurance cost to and from the point of repair, and all risk of shipment rest with the customer. The warranty does not cover service required by reason of abuse or unauthorized modifications. Returning the RT-30 to Infotek with the BATTERY ON-OFF switch in the ON position will void the warranty in regards to the RT-30 power supply.

TABLE 1
EQUIPMENT SUPPLIED

<u>Item</u>	<u>Description</u>	<u>Qty.</u>
1	Real Time Clock, Model RT-30	1
2	Interface Cable for 9830A/B factory preset for select code 1 (unless otherwise specified in the purchase order)	1
3	AC Power Cord	1
4	Instruction Manual	1
5	Spare Fuse for 120 volt operation 1 Amp	1
6	Fuses for 240 volt operation 1/2 Amp	2

SPECIFICATIONS

Dimension: Depth: 29.05 cm (11.44 in.)
 Height: 12.70 cm (5.00 in.)
 Width: 17.78 cm (7.00 in.)



Weight: 4.20 Kg (9.25 lbs.)

Input Power: 60 Hz units 110 or 220 volts AC, 70VA
 Option 02: 50 Hz units 220 volts AC, 70VA

INSTALLATION

- Step 1: Turn off the power to the 9830 and turn off any peripheral device which may be connected such as the printer, outboard cassette, etc.
- Step 2: Install the interface cable into an available I/O slot at the back of the 9830.
- Step 3: If the RT-30 was purchased without a display or if the internal display was ordered go to Step 4. The following steps detail the placement of the RT-30 external display.
 - a. Remove power and cables from 9866 printer and the FD-30 floppy disk if these peripherals are installed on the top of the 9830.

- b. Place the 9866 and FD-30 to one side of the 9830.
- c. Place the flat ribbon cable on top of the 9830.
- d. Remove the backing from the adhesive on the back of the display.
- e. Place the display along the top edge of the 9830.
- f. Re-install the 9866 and FD-30 Floppy Disk. Install the power cords.

Step 4: Turn on the power to the computer, any peripherals and move the back panel BATTERY switch to the ON position on the RT-30. Install the power cord.

This completes installation of the RT-30.

DISPLAY

The RT-30 optional display will indicate the current time of the RT-30. When the INTERRUPT TIME DISPLAY button is pressed, the RT-30 will display the next interrupt time.

BATTERY

CAUTION

Ensure that the BATTERY ON-OFF switch is set to ON whenever the RT-30 is plugged in. Failure to do so will prevent battery back-up for time setting in the event of power failure. When power is to be removed for more than 6 hours, when the RT-30 is to be stored, or whenever it is shipped, set the BATTERY ON-OFF switch to OFF. Failure to do so will cause battery failure due to excessive discharge time and will void the warranty.

The battery in the RT-30 will maintain the RT-30 time for approximately 6 hours. The battery is activated by a slide switch at the rear of the RT-30. Whenever the RT-30 is operating from the battery the display will be blank and will display the time only when the INTERRUPT TIME DISPLAY button is pressed.

EXTERNAL 5MHz SIGNAL SOURCE

The RT-30 is accurate to 30 seconds per month, however some applications may require greater accuracy. For this reason

a BNC connector is available on the rear panel of the RT-30 for connection to a suitable time reference. A slide switch allows the user to select either the internal or external signal source. The external source must be capable of driving two standard TTL loads.

OPERATION

SET STATEMENT

The SET statement enables the RT-30 to be set to an initial time. The time may be the present day, hour, minute, second, or may be any other time that is desired by the user.

SYNTAX

```
WRITE(SELECT CODE,*)"SETDDDHHMMSSTT"
```

The characters, SET, instruct the RT-30 that the following numbers are the initialization time. The numerals following the SET characters are interpreted as follows: DDD, a three digit number corresponding to the Julian Day; HH, a two digit number corresponding to the hour (24 hour); MM, a two digit number corresponding to the minute; SS, a two digit number which corresponds to the second, and TT, a two digit number corresponding to the number of hundredths of a second. Leading zeroes must appear, e.g. 10 days must appear as 010, 1 second must appear as 01.

Example;

```
WRITE(SC,*)"SET10013301510"
```

Results in a setting in the RT-30 to 100 days, 13 hours, 30 minutes, 15.10 seconds.

After recognizing the SET command the RT-30 will ignore any non-numeric character. When used within quote marks, characters may be used to clarify the setting. The two examples that follow set the RT-30 to 100 days, 13 hours, 30 minutes, 15.10 seconds.

```
WRITE(SC,*)"SET100,13,30,15,10"  
WRITE(SC,*)"SET 100 DAYS - 12:30:15.01"
```

The flexibility of the output to the RT-30 can be seen from the last example.

Special Considerations

The RT-30 accepts data in the following manner: The RT-30 display is a 12 digit field very similar to a string variable that is 12 characters long. For the purpose of clarification we will use the following convention:

```
D1,D2,D3 - Days  
H1,H2 - Hours  
M1,M2 - Minutes
```

S1,S2 - Seconds
T1,T2 - Hundredths

The clock stores this data as: D1 D2 D3 H1 H2 M1 M2 S1
S2 T1 T2. The RT-30 places the first number from the
9830 in position D1, the second number in D2 and so on
until T2 is filled. Try the following from the
keyboard;

```
WRITE(1,*)"SET"1
```

The 1 (one) appears in position D1. Now suppose you wanted
to set the clock to 000 days and 12 hours;

```
WRITE(1,*)"SET"00012
```

Results in a reading of 12 in position D1 and D2. Remember
the 9830 suppresses leading zeroes, except when in literals
(quote marks) or strings! Now try the following;

```
WRITE(1,*)"SET"10012
```

Now the clock is set to 12 hours but it is also set to 100
days. Now replace the 1 (one) with a 0 (zero) by executing
the following:

```
WRITE(1,*)"SET"0
```

Now to set the clock to any initial time the following
routine is used:

```
10  REM SET RT-30 TIME
20  DISP "DAYS";
30  INPUT D
40  DISP "HOURS" '
50  INPUT H
60  DISP "MINUTES";
70  INPUT M
80  DISP "SECONDS" '
90  INPUT S
100 DISP "HUNDREDTHS" '
110 INPUT T
120 A=D*1E+08+H*1E+06+M*1E+04+S*1E+02+T
130 IF A>1E+10 THEN 180
140 A=A+1E+10
150 WRITE(1,*)"SET"A
160 WRITE(1,*)"SET"0
170 GO TO 190
180 WRITE(1,*)"SET"A
190 END
```

Line 10 through 110 inputs Day, Hours, Minutes, Seconds and Hundredths. Line 120 builds the variable A by multiplying each of the input values by the appropriate number. Line 130 checks to see if A is greater than 1E10. This check insures that the RT-30 position D1 has a number greater than 0. If position D1 is less than 0, then line 140 adds 1E10 to A. This insures that all characters will be in the correct location. Line 150 outputs A and line 160 outputs a 0 to eliminate the 1 added by line 140.

The RT-30 will accept the first three numbers after the SET statement as the day, the next two numbers as the hour, the next two as the minutes, the next two numbers as the second and the next as the hundredths. Important - The last two numbers which follow the seconds will become the fractional part of a second.

For Example;

```
WRITE(SC,*)"SET 100,12,30,15,19,78"
```

will result in a time set to 100 days, 12 hours, 30 minutes, 15 seconds and 78 hundredths of a second. The 19 is not stored into the fractional part of a second.

Usage Example

```
10 DIM D$(3),H$(2),M$(2),S$(2),T$(2)
20 DISP"# OF DAYS";
30 INPUT D$
40 DISP"HOUR";
50 INPUT H$
60 DISP"MINUTE";
70 INPUT M$
80 DISP"SECOND";
90 INPUT S$
100 DISP"TENTHS";
110 INPUT T$
120 WRITE(1,*)"SET"D$,H$,M$,S$,T$
130 END
```

The above example allows the variables D\$, H\$, M\$, S\$, and T\$ to be input as variables and output to the RT-30. Strings may be used instead of the program on page 5 to guarantee the correct number of places in each digit. Days are represented by three digits e.g., three days are input 003 in the above example. Variables can be used along with format statements if the variables have the correct number of digits per location. Remember the 9830 suppresses leading zeroes in a format statement.

INT STATEMENT

The INT statement sets the time for the RT-30 interrupt procedure. Interrupts on the RT-30 can be specified only to the nearest second. The same basic syntax used in the SET instruction is used in the INT instruction. The interrupt feature of the RT-30 is activated by the SERVICE statement of the FB-III.

SYNTAX

```
WRITE(1,*)"INTDDDHHMMSS"
```

Where DDD equals the day, HH equals the hours, MM equals the minute and SS equals the second at which the interrupt is to occur.

The RT-30 ignores any non-numeric characters which follow INT. Since the RT-30 can be set to interrupt only to the nearest second, any numbers which follow the two digit second designation will become the seconds.

```
WRITE(1,*)"INT100,12;30;15.17"
```

This instruction will result in an interrupt at the 100th day, 12:30 p.m. and 17 seconds. The 15 is not recognized by the RT-30 as seconds.

The following example can be used to set the interrupt time via variables. This method can be valuable when computing new interrupts based on existing conditions.

Interrupts can be computed with the RT-30 read via the ENTER statement of the Extended I/O ROM as follows:

```
10 DISP "SELECT CODE OF RT-30";
20 INPUT R
30 WRITE (R,*)"SET 1"
40 DISP "INTERRUPT IN HOW MANY SECONDS";
50 INPUT I
60 SERVICE R
70 ENTER (R,80)D,H,M,S,T
80 FORMAT F3.0,4F2.0
90 IF I < 60 THEN 160
100 I=I-60
110 M=M+1
120 IF M < 60 THEN 150
130 M=0
140 H=H+1
150 GOTO 90
160 IF S+I < 60 THEN 260
170 S=(S+I)-60
180 M=M+1
```



```

190 IF M < 60 THEN 160
200 M=M-60
210 H=H+1
220 IF H < 24 THEN 190
230 H=0
240 D=D+1
250 GOTO 190
260 S=S+I
270 A=D*(1 E+06)+H*(1E+04)+M*(1E+02)+S
280 IF A > 1E+08 THEN 330
290 A=A+1E+08
300 WRITE (R,*)"INT"A
310 WRITE (R,*)"INT"0
320 GOTO 340
330 WRITE (R,*)"INT"A
340 WRITE (R,*)"KEY CONT 380"WBYTEll
360 DISP "NEXT INTERRUPT AT"A
370 END
380 FOR N=1 TO 5
390 BEEP
400 WAIT 250
410 NEXT N
420 GOTO 70

```

The service statement in line 60 allows the RT-30 (select code R) to input data to the 9830. The ENTER statement in line 70 reads the current time from the RT-30. Lines 90 through 260 increment the seconds variable (S) by the amount entered in line 50. These lines also adjust the minutes, hours, and days variables in the event of overflow. Line 270 forms the variable A so that A is in the format DDD HH MM SS. As the 9830 suppresses leading zeroes, lines 280 through 320 adds 100 00 00 00 to A if A is less than that value, sets the interrupt time to IDD HH MM SS in line 300, then removes the leading 1 in line 310. In line 340, the characters which are to be provided to the 9830 by the RT-30 at the interrupt time are entered into the RT-30 buffer. WBYTE ll is the binary value of the EXECUTE key.

PIC STATEMENT

The PIC statement allows the user to specify the precision of the data which is to be output from the RT-30. For example, if the intervals required are nominally less than a minute, the RT-30 can be instructed to suppress the days and hours and only minutes, seconds and hundredths will be output from the RT-30.

SYNTAX

```
WRITE(1,*)"PICD"
```

Where D is equal to days.

```
WRITE(1,*)"PICH"
```

Where H equals hours.

```
WRITE(1,*)"PICM"
```

Where M equals minutes.

```
WRITE(1,*)"PICS"
```

Where S equals seconds.

```
WRITE(1,*)"PICF"
```

Where F equals fractional part of a second.

In all the above examples, the character following the PIC statement must be one of the following above five choices or else the command is ignored and the last PIC statement remains in effect.

The PIC statement is not affected by any characters following the parameter.

For Example;

```
WRITE(1,*)"PICHOURS"
```

In the above example, the PIC statement would indicate to the RT-30 that the output precision is to be changed and the character H indicates that everything above the hours is to be suppressed. This becomes a very convenient way of reminding the user in a program what the precision of the clock is.

KEY STATEMENT

The KEY statement allows the instructions which are to be executed upon interrupt to be output to the RT-30 buffer. The 9830 executes the KEY instructions from the RT-30 as if a user is pressing the keys on the 9830 keyboard. With this in mind, operations which are normally not programmable, such as TLIST, LIST, XREF, LXREF, can be easily executed via the RT-30 key mode. NOTE: A 'STOP' cannot be output from the RT-30. The reasons are twofold: 1) The 'STOP' key does not have a decimal code and 2) Typing in the characters 'S" T" O" P' while a program is running will have no effect. The STOP key is a hardware interrupt. The only way that a running program can be broken into is to use the MDUMP statement in Fast Basic III.

SYNTAX

```
WRITE(1,*)"KEY(ANY STATEMENT)"
```

Following the KEY statement any instruction or statement can be stored into the RT-30. Upon interrupt the characters between the parenthesis will be executed as if you were pressing the keys on the 9830. Outside the quotation marks the decimal equivalent of any 9830 key may be used. Appendix B of the Fast Basic III ROM instruction manual contains the decimal codes for all keys (except STOP).

Example;

1. WRITE(1,*)"KEY RUN",WBYTE 11
2. WRITE(1,*)"KEY"W BYTE82,WBYTE85,WBYTE78,WBYTE11
3. WRITE(1,*)"KEY"WBYTE15,WBYTE11

All three examples will begin execution of a program in the 9830. The EXECUTE key is represented by the WBYTE11 in all three examples. Example (2) spells RUN with the binary equivalent of the letters R, U, & N, followed by a WBYTE11 which is the EXECUTE key. Example (3) uses the WBYTE code 15 which corresponds to the RUN key on the 9830. A FORMAT B statement could have been used to reduce the WBYTE's to the decimal equivalent as in the following Usage Example.

Usage Example

```
10 WRITE (1,*)"KEY PRINT"34"RT-30 REAL TIME  
CLOCK",34,11 20 FORMAT 3B
```

Upon interrupt the 9866 Printer would print RT-30 REAL TIME CLOCK. Since a quote cannot be contained in a quote, it is necessary to output a quote as the ASCII equivalent (34).

Executing the above instruction from the RT-30 would be the same as typing PRINT"RT-30 REAL TIME CLOCK" and then pressing EXECUTE. Remember, the RT-30 outputs characters to the 9830 as if you are pressing the keys. The 9830 does not know the difference between an RT-30 issuing instructions and a user pressing keys on the keyboard. For this reason, any characters which are not allowed under normal 9830 syntax will cause an ERROR, both from the keyboard and when executed via RT-30.

Many times it is necessary to execute multiple commands in one KEY statement. When this is required, a delay parameter is required at the end of the KEY statement. To signify end of key storage, place a zero followed by a comma, then the delay time in seconds enclosed in quotes. The delay time can be between 1 and 255 seconds. These delays are represented by 001 and 255.

Example;

```
10 WRITE (1,*)"KEY LOAD#5,0"WBYTE 11,LIST" WBYTE
11,0,"030"
```

The above statement would load in file 0 from the floppy disk (select code #5), wait 30 seconds and then list the program.

Usage Example

```
10 DIMD$(3), H$(2), M$(2), S$(2), T$(2)
20 SERVICE 1
30 D$="000"
40 H$="12"
50 M$="00"
60 S$="00"
70 WRITE(1,*)"SET000,00,00,00,00"
80 WRITE(1,*)"INT"D$,H$,M$,S$
90 WRITE(1,*)"KEY LOAD #5,1"WBYTE11,WBYTE15,WBYTE11,
0,"030"
100 END
```

This program initializes the RT-30 to 0 in line 60. Line 70 sets the interrupt to be in 12 hours. Line 80 then stores the keystrokes which are to be executed upon interrupt. In this case, a program from file 1 is to be loaded, a delay parameter 30 seconds insures that the program is loaded. Decimal 15 (RUN) is executed followed by decimal 11 (EXECUTE).

ENTER STATEMENT

The ENTER statement is from the Extended I/O ROM and is used to read the RT-30. For a more detailed discussion of the ENTER statement refer to the HP Extended I/O Manual.

SYNTAX

```
10 ENTER(1,20)D,H,M,S,T
20 FORMAT F3.0, 4F2.0
```

Where the variable D equals days, H equals hours, M equals minutes, S equals seconds and T equals tenths.

The RT-30 is read via the ENTER statement. The appropriate days is read into the first variable, hours into the second, minutes into the third, seconds into the fourth and tenths into the fifth.

Usage Example

```
10 WRITE(1,*)"SET 100-12;30;15.15"  
20 ENTER(1,30)D,H,M,S,T  
30 FORMAT F3.0, 4F2.0
```

The variables will be as follows, D equals 100, H equals 12, M equals 30, S equals 15 and T equals 5.

The RT-30 can be read in its entirety via one variable as follows:

```
10 ENTER(1,*)D  
OR  
10 ENTER(1,20)D  
20 FORMAT F11.0
```

This results in the value of D equal to 10012301505. By the same token if the statement is;

```
10 ENTER(1,20)D,H  
20 FORMAT F3.0, F8.0
```

D would be set equal to 100, H would be set equal to 12301505.

The PIC statement used in conjunction with the ENTER statement allows the RT-30 to inhibit the output of characters.

For Example;

```
10 WRITE(1,*)"SET100-12;30;15.05"  
20 WRITE(1,*)"PICM"  
30 ENTER(1,35)D,H,M,S,F  
40 FORMAT 3F2.0  
50 END
```

D will not be equal to the number of days, but will be equal to 30 minutes. H will be equal to 15 seconds. M will be equal to .05 tenths. S and F will be undefined.