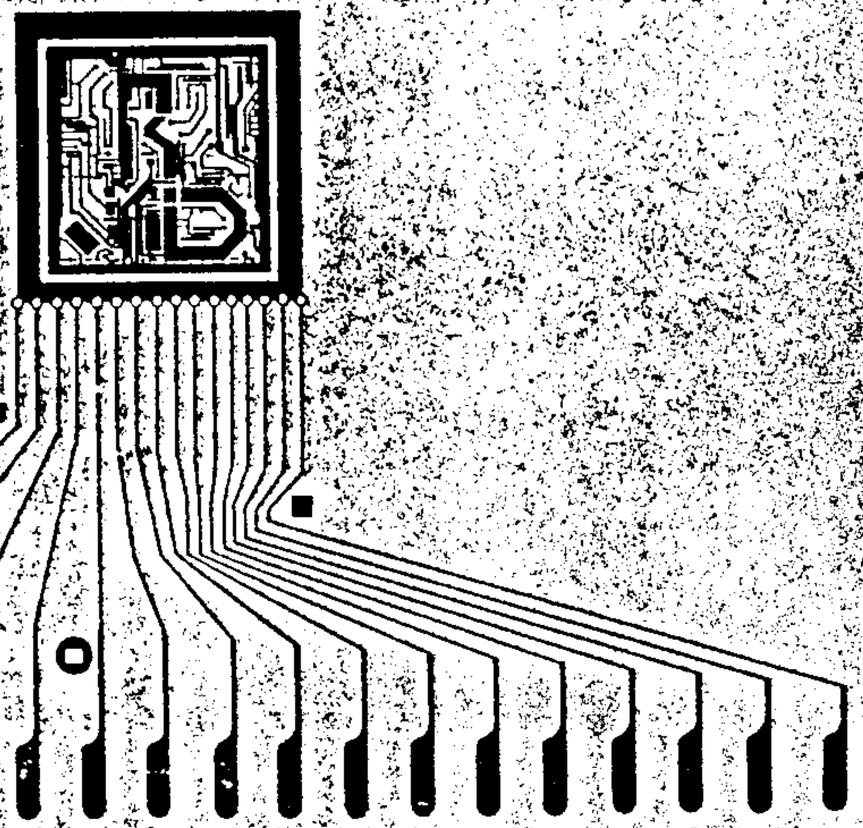
MARKETING AREAS : SOFTWARE PUBLISHING 1986

Ryte Data

210 MOUNTAIN STREET
HALIBURTON, ONTARIO K0M 1S0
CANADA

Super Clock Support



Program Manual

INTRODUCTION:

Super Clock Support is a program designed to facilitate easier use of your CORCOMP 9900 CLOCK or TRIPLE-TECH card. This software package includes the following :

1) Three independent timers which can be set and read from EXTENDED BASIC.

2) Reading of the date in text form rather than the standard 00/00/00 format.

3) Reading of the time in either 12 or 24 hour mode with the 12 hour mode including an AM/PM function.

4) Allows day of the week, date, and time to be set independent of each other rather than having to set all of them at once.

5) Two interrupt driven utilities to allow constant display of the time or user called time display.

LOADING PROCEDURE:

You can load the support utilities either in immediate mode or in a program in EXTENDED BASIC.

A) Immediate mode:

```
CALL INIT
CALL LOAD("DSK.SUPPORT.SUPCLKSUP")
```

B) EXTENDED BASIC program:

```
10 CALL INIT :: CALL LOAD("DSK.SUPPORT.SUPCLKSUP")
```


HOW IT WORKS:

Super Clock Support accesses the ROM of the clock card directly rather than going through a DSRLNK call. This means that Peripheral Access Blocks do not need to be opened in VDP RAM and that it does not have to execute the DSRLNK routine. Also the program does not have to execute the full code stored in the ROM of the card. By accessing in this way the program saves time and does not interfere with standard EXTENDED BASIC operation. Watch for future programs by Monty Schmidt which operate in the same fashion!

USING SUPER CLOCK SUPPORT:

Now that you own Super Clock Support it is time to learn how to use its routines to their full extent. Here is a list of the routines, examples of how to use them, and suggestions for using them in your own programs.

Timer Routines:

```
CALL LINK("TIMER1"),

CALL LINK("ELAPS1",X$)

CALL LINK("TIMER2"),

CALL LINK("ELAPS2",X$)

CALL LINK("TIMER3"),

CALL LINK("ELAPS3",X$)
```

These 6 call link statements allow you to set 3 independent timers and see how long they have

been running at any time. To start a timer running use the statement CALL LINK("TIMERX"), where X is replaced by a number 1 through 3. To find out how long one of the three timers has been running use the statement CALL LINK("ELAPSX",X$) where X is a number 1 through 3. The variable X$ will hold the elapsed time in hours, minutes, and seconds. The variable will be in the form of "00:00:00" which is read by your program and displayed on the screen - or used as a decision point to be acted upon by your program.

Examples:

Here is a program which asks you to set the length of time you wish to time something (say a cake in the oven), displays the time on the screen and sets off a bell when the time has elapsed.

```
10 CALL INIT :: CALL LOAD("DSK.SUPPORT.SUPCLKSUP"):: CALL CLEAR
20 INPUT "Time please? (00:00:00) ":A$ :: CALL LINK("TIMER1") :: CALL CLEAR
30 CALL LINK("ELAPS1",B$):: IF B$<A$ THEN DISPLAY AT(12,11):B$ :: GOTO 30
40 CALL SOUND(1000,880,1) :: END
```

A perfect use for these subroutines would be in the limiting of time that a person could use a program such as a Bulletin Board System. When the person logged on, the program could use timer 1 to check how long the user had been on. He then could use timer 2 to see how long he had been in a certain section of the board such as a download section. A message could be sent based upon this length of time or some action taken to allow another user to log on etc.

READING TIME AND DATE ROUINTES:

CALL LINK("DATE",X$)

When reading the date ofthe CORCOMP 9900 CLOCK using the standard ' INPUT #1:' statement of Basic, the day of the week anddate, are returned in the familiar format "0","00/00/00" that you have already seen in use. In this format the first string is a number from 0 through 6 which corresponds to the day of the week and the second string is the date in the form month/day/year.

Super Clock Support returns the date in a more esthetic format. This format is "day of week, month, day, year." Here is an example to make this a bit clearer. If the clock had the following data in it: 3, 03/14/86,, and you used CALL LINK("DATE",X$) this is what would be returned in X$. X$="Wednesday, March 14, 1986". A more human oriented format!

Super Clock Support equates the following numbers to the following strings.

0= Sunday
1= Monday
2= Tuesday
3= Wednesday
4= Thursday
5= Friday
6= Saturday
7= Sunday

Years are returned as 1980-1999 or 2000-2079.

This should allow you plenty of time to use Super Clock Support in all your programs!

```
CALL LINK("TIME12",X$)
CALL LINK("TIME24",X$)
```

The time in the clock is stored in 24 hour format where the time varies from 00:00:00 through 23:59:59. The standard 'INPUT #1:' statement only allows you to read it in this format. Super Clock Support allows you to read it in 2 formats.

CALL LINK("TIME24",X$) returns the standard 24 hour time in X$ where X$= "00:00:00" - "23:59:59".

CALL LINK("TIME12",X$) returns the time in a 12 hour format with an A.M. or P.M. indicator. X$= either "12:00:00 A.M."- "11:59:59 A.M." or "12:00:00 P.M." - "11:59:59 P.M.".

Examples:

This program prints the date on the screen

```
10 CALL INIT :: CALL LOAD("DSK.SUPPORT.SUPCLKSUP")
20 CALL LINK("DATE",X$):: PRINT X$
```

(Just for reference, if your application program has already done CALL INIT, the statement does not have to be repeated.)

This program displays the time in 24 hour mode in the upper right corner of the screen.

```
10        CALL        INIT        ::        CALL
LOAD("DSK.SUPPORT.SUPCLKSUP"):: CALL CLEAR
20 CALL LINK("TIME24",X$):: DISPLAY AT(1,21):X$  ::
```



```
GOTO 20
```

This program displays the time in 12 hour mode in the upper right corner of the screen

```
10        CALL        INIT        ::        CALL
LOAD("DSK.SUPPORT.SUPCLKSUP"):: CALL CLEAR
20  CALL LINK("TIME12",X$):: DISPLAY AT(1,16):X$ ::
GOTO 20
```

This program functions as an alarm clock. It first asks you to input the time you want to wake up and then continually displays the time on the screen. When the time limit is reached it sounds a tone until a key is pressed.

```
10        CALL        INIT        ::        CALL
LOAD("DSK.SUPPORT.SUPCLKSUP"):: CALL CLEAR
20  PRINT "Input the time you want to " :: PRINT
"wake up in 24 hour format.  00:00:00 "  ::  INPUT
:X$
30 CALL CLEAR
40  CALL  LINK("TIME24,B$):: IF B$<X$ THEN CALL
LINK("TIME12",B$):: DISPLAY AT(1,16):B$ :: GOTO 40
50 CALL SOUND(1000,440,0):: CALL KEY(0,K,S):: IF
S=0 THEN 50 ELSE END
```

Again an excellent use for these routines would be in a Bulletin Board System. If you wanted to display the time and date when a user logged on to the BBS all you would have to do is use the CALL LINK("DATE",X$) and CALL LINK("TIME12",Y$) and print them out!

WRITING TIME AND DATE ROUTINES:

When writing to the CORCOMP 9900 CLOCK using the standard PRINT #1: statement you must reset all the variables in the card at the same time. With Super Clock Support you can change any one of the three variables independent of the others!

```
CALL LINK("SETDAY",X$)
```

To change only the day of the week all you have to do is use the utility CALL LINK("SETDAY",X$) where X$ is a number from 0 through 6 (corresponding to the table listed below) passed as a string variable. You must use a string variable to pass the actual day in text format.

Super Clock Support equates the following strings to these days:

```
"0"= "Sunday"
"1"= "Monday"
"2"= "Tuesday"
"3"= "Wednesday"
"4"= "Thursday"
"5"= "Friday"
"6"= "Saturday
```

```
CALL LINK("SETDAT",X$)
```

To change only the date use the utility CALL LINK("SETDAT",X$) where X$ is a string passed in the format "00/00/00". The string is interpreted as month/day/year. You do not have to use the slashes between numbers. Super Clock Support will accept "00-00-00" or any other delimiter, however you must have one character between each set of two numbers.


```
CALL LINK("SETTIM",X$)
```

To change only the time use the utility CALL LINK("SETTIM",X$) where X$ is a string passed in the format "00:00:00". The string is interpreted as hours:minutes:seconds. Again you do not have to use the ":" symbol in between numbers but must have one character between each one of the number values.

EXAMPLES:

This program sets the day of the week to Sunday but does not affect the date or time.

```
10 CALL INIT :: CALL LOAD("DSK.SUPPORT.SUPCLKSUP")
20 CALL LINK("SETDAY","0")
```

If the clock originally held:

Friday, April 7, 1986 12:25:06

The clock would now hold:

Sunday, April 7, 1986 12:25:06

This program sets the date but does not affect the day of the week or the time.

```
10 CALL INIT :: CALL LOAD("DSK.SUPPORT.SUPCLKSUP")
20 CALL LINK("SETDAT","01/25/02")
```

If the clock originally held:

Friday, April 7, 1986 12:25:06

The clock would now hold:

Friday, January 25, 2002 12:25:06

This program sets the time but does not affect the day of the week or the date.

```
10 CALL INIT :: CALL LOAD("DSK.SUPPORT.SUPCLKSUP")
20 CALL LINK("SETTIM","15:23:34")
```

If the clock originally held:

Friday, April 7, 1986 12:25:06

The clock would now hold:

Friday, April 7, 1986 15:23:34

INTERRUPT RELATED ROUTINES:

There are two interrupt routines which are established by placing their addresses at >83C4 in Console Ram. This is of no concern unless you are trying to run them with a program that is already using interrupt routines. This should not be a problem in most cases. You will have to watch the use of >83C4 (scratchpad RAM) to make sure other interrupt driven routines do not use this location. Only one of these routines may be running at any

one time. If you turn on the other it will disable the previously set routine.

```
CALL LINK("INTON1")
```

This utility turns on interrupt utility one. When interrupt utility one is running and you press both the Control key and the T key at the same time the current time in 24 hour format will be displayed in the upper right corner of your screen.

```
CALL LINK("INTON2")
```

This utility turns on interrupt utility two. When interrupt utility two is running the current time in 24 hour format is constantly displayed in the upper right corner of your screen.

```
CALL LINK("INTOFF")
```

This utility disables both of the interrupt utilities. To turn an interrupt utility back on after using CALL LINK("INTOFF") all you have to do is turn it back on using the previous two CALL LINK statements.

EXAMPLES

```
CALL LINK("INTON1")
```

Interrupt routine one is now running. User presses Control T and the time appears in the upper right of the screen.

```
CALL LINK("INTOFF")
```

Interrupt routine one is now disabled. User presses Control T and nothing happens!

CALL LINK("INTON2")

Interrupt routine two is now running. Time is constantly updated and displayed in upper right of screen.

CALL LINK("INTOFF")

Interrupt routine two is now disabled. Time is no longer updated and displayed in upper right of screen.

SPECIAL NOTES:

All the utilites described can be called in a program or in immediate mode. The interrupt routines will also continue to work while accessing peripherals! WARNING: DO NOT use the CALL INIT statement without first disabling the interrupt routines because this can cause erratic behavior in the system!

Since Super Clock Support uses the standard character set used by EXTENDED BASIC you should not redefine the characters used by the interrupt routine. If you redefine the characters 0-9 or the ":" character they will be shown as you defined them. I.E. if you redefine the character 0 to be a colored block and then enable the interrupt routine. When SCS prints the time all 0 characters will show on the screen as colored blocks!

Two files are included on the disk you have received. The first one SUPCLKSUP is the complete

Super Clock Support system. The second file SUPCLKLIM is a limited version including only the utilities: DATE, TIME12, TIMER1, TIMER2, TIMER3, ELAPS1, ELAPS2, ELAPS3, INTON2, and INTOFF. This was done to allow for a smaller memory usage should you want to use Super Clock Support with a BBS system.

SUPCLKSUP occupies approximately 1.7K of memory while SUPCLKLIM occupies approximately 1.3k of memory. Super Clock Support is not copy protected in any way shape or form to allow you to add it to other programs without problems. All rights are reserved - please do not include these routines in programs for others! The author of this software hopes that you will have respect for his time and effort and not distribute this software.

* CORCOMP 9900 CLOCK and TRIPLE-TECH card are trademarks of CORCOMP INC. EXTENDED BASIC is a trademark of TEXAS INSTRUMENTS INC.

Ryte Data warrants the Super Clock Support program, manufactured by Ryte Data, to be free from defects in materials and workmanship for a period of 120 days from the date of purchase. During the 120 day warranty Ryte Data will replace any defective disks at no additional charge. Owners must return the defective product, shipped prepaid to:

Ryte Data,
210 Mountain St.
Haliburton, Ontario K0M 1S0 (705) 457-2774

The author does not warrant that the programs and manual will be free from error. The consumer assumes complete responsibility for decisions made or actions taken based on information obtained from this manual and programs. Statements made concerning this program are not to be construed as