AMSTRAD

6128/CPC 464

REAL TIME CLOCK

dk'tronics

CONTENTS

PAGE

INTRODUCTION 1

INSTALLATION 2

RSX's |SETTIME 3
|ASKTIME 4
|PEEKRTC 4
|POKERTC 5
|TIMEON 5
|TIMEOFF 6
|ALARMON 6
|ALARMOFF 7

BASIC DEMO - SIMPLE TIMER 6

USE UNDER CP/M PLUS 7

8-WAY I/O PORT 8

BATTERY 9

MACHINE CODE PROGRAMMERS 10

ADDRESS MAP 10

INTRODUCTION

The Real Time Clock module uses the 146818 chip which gives the user a complete time-of-day clock with alarm, calendar, a programmable periodic interrupt generator and 50 bytes of low power static RAM.

Features

Time-of-day clock and calendar

Binary or BCD representation of time, calendar and alarm

12 or 24 hour clock with AM and PM in 12-hour mode

Automatic end of month recognition

Automatic leap year compensation

Battery-backed. Automatic trickle-charge when computer is on.

8-way bidirectional I/O port for external control.

Supplied with 8 RSX's for ease of use under BASIC.

The additional commands are :-

|SETTIME,a,b,c,d,e,f,g	set the time, years to secs.
|ASKTIME,@a,@b,@c,@d,@e,@f,@g	ask for the time years to secs.
|PEEKRTC,a,@b	read a location within the RTC.
|POKERTC,a,b	write to a location within the RTC.
|TIMEON,a,b,c	enable automatic time display.
|TIMEOFF	turn off time display function.
|ALARMON,a,b,c,@d	turn alarm function on.
|ALARMOFF	turn alarm function off.

CP/M PLUS utility for setting up the RTC and updating the internal software clock.

AMSTRAD CPC REAL TIME CLOCK (RTC)

The interface has a through-connector to allow other peripherals to be connected at the same time, thus eliminating the need for continually removing peripherals.

INSTALLATION

WARNING

Ensure that the power to your Amstrad computer is switched off before you fit the interface to the expansion socket. Failure to comply with these instructions may cause permanent damage to the interface or the computer.

Power down your computer. Plug the RTC into the expansion socket on the back of the computer so that the I/O connector is on the left hand side as viewed from the front of the computer.

The computer should now power up as normal. If it fails to do so, check that the RTC is correctly fitted. Note that all DK'tronics products have a keyway location on the connector to ensure that there can be no alignment problems, other interfaces may not have this keyway, the Amstrad DDI-1 Interface being an example.

SOFTWARE DETAILS

The RTC module is supplied with various programs which allow the user to use the RTC, these are:-

RSX.BAS
RSX.BIN
CLOCK.COM

The RSX.BAS program, when run, adds 8 new commands to BASIC which allows the user to read/modify the RTC time/data, or display the time automatically.

The RSX's must first be installed before they can be used, this is done by typing in RUN "RSX" [ENTER/RETURN]., When you have done this you will be asked where you want the RSX program to run in memory. Obviously it should be loaded as high as possible so that BASIC can have as much memory as possible. The actual RSX program (RSX.BIN) requires 986 bytes of memory so you must take this into account when you reply to the prompt "run address ?". If you are not sure where to put it then simply press the RETURN/ENTER key in reply and the program will place the RSX code as high as possible, it will also inform you where it has placed the code.

When this is done the program modifies the code so that it can run at the specified address,it then removes itself as it is no longer needed.

Care should be used in, running the "RSX.BAS" program as it is not protected for multiple insertions, it should only be used once to insert the RSX code.

Once inserted, the available RSX's are as follows :-

|SETTIME.

This can take 1 TO 7 parameters. Thus the command in full is,

```
|SETTIME,YEAR,MONTH,DATE,DAY,HOURS,MINUTES,SECONDS
```

Note that all the parameters need not be used at once, EG if you only want to change the secondsto 25 then the command would be :-

```
    SECONDS% = 25                 [ENTER/RETURN]

    |SETTIME,SECONDS%             [ENTER/RETURN]
or
    |SETTIME,25                   [ENTER/RETURN]
```

However if you wish to change the YEAR then all the other parameters must follow. Thus if you want to set up the clock completely, then you might type in the following :-

```
|SETTIME,87,3,19,5,11,26,45     [ENTER/RETURN]
```

You are now prompted with :

```
PRESS ANY KEY TO SET THE TIME
```

At the exact time press any key and the RTC will be set to the selected time.

** NOTE **

You must not update the clock parameters when the |TIMEON facility is being used. See |TIMEOFF command following.

|ASKTIME.

As above this can take 1 TO 7 parameters. Thus the command in full is,

```
|ASKTIME,@YEAR%,@MONTH%,@DATE%,@DAY%,@HOURS%,@MINUTES%,@SECONDS%
```

Note that all the parameters need not be used at once, EG if you only want to obtain the seconds the command would be :-

```
SECONDS%=0                      [ENTER/RETURN]
```

Note that variables must be initialised before use otherwise BASIC will return the "IMPROPER ARGUMENT" error message.

```
|ASKTIME,@SECONDS%              [ENTER/RETURN]

PRINT SECONDS%                  [ENTER/RETURN]
```

However if you wish to obtain the HOURS value then the parameters after the HOURS must be used. Thus to obtain the HOURS you would type in the following :-

```
SECONDS%=0 : MINUTES%=0 : HOURS%=0        [ENTER/RETURN]

|ASKTIME,@HOURS%,@MINUTES%,@SECONDS%      [ENTER/RETURN]

PRINT HOURS%, MINUTES%, SECONDS%          [ENTER/RETURN]
```

The variables, HOURS%,MINUTES%,SECONDS% would be updated accordingly. Note that when you wish to return data from machine code back to BASIC variables then the '@' symbol must be used, and in this case the variables must be INTEGER, ie followed by the '%' symbol.

|PEEKRTC.

This command is similar to the BASIC PEEK command in that it is used to inspect a particular memory location. Here the command is used to inspect any of the RTC locations. There are 64 locations in all and are numbered 0 - 63 where 0 is the first location and 63 is the last.
The first 14 locations are related to the clock function of the RTC chip, the remaining 50 locations can be used for general purpose data storage. See Address Map, page 10, for full details

The command takes the following format :-

```
|PEEKRTC,ADDRESS%,@DATA%                  [ENTER/RETURN]
```

Both parameters must be used where ADDRESS% is in the range 0 - 63, and DATA% is the variable where the data is returned to. Say you wish to inspect location 25 then the commands would simply be :-

DONT USE DATA USE SOMTHING ELSE

```
DATA% = 0                              [ENTER/RETURN]

|PEEKRTC,25,@DATA%                     [ENTER/RETURN]

PRINT DATA%                            [ENTER/RETURN]
```

|POKERTC.

This command is similar to the BASIC POKE command in that it is used to write to a particular memory location. Here the command is used to write to any of the RTC locations. See Adress Map,page 10, for full details.

The command takes the following format :-

```
|POKERTC,ADDRESS%,DATA%
```

Both parameters must be used where ADDRESS% is in the range 0 - 63, and DATA% contains the value to be written. Say you wish to write 78 to location 25 then the command would simply be :-

```
ADDRESS%=25 : DATA%=78                 [ENTER/RETURN]

|POKERTC,ADDRESS%,DATA%                [ENTER/RETURN]
or
|POKERTC,25,78                         [ENTER/RETURN]
```

NOTE - the first 10 locations in the RTC chip are related to time functions where the information is in BCD format. Thus during the |PEEKRTC and |POKERTC commands the data is converted to BCD format for these locations. The exception to this is the alarm locations. If the alarm times are set as normal then they are converted to BCD format, however if the alarms are set to the 'DON'T CARE' codes then they are not converted to BCD.

|TIMEON.

This command can take either 0 or 3 parameters and is used to print the time on the screen at a given location at a particular frequency. Once used the time will continually be displayed until the |TIMEOFF command has been executed. The command in full is :-

```
|TIMEON,COLUMN,ROW,RATE          [ENTER/RETURN]
```

Say you wish to display the time at the top right-hand side of the screen in mode 1 every second then the following would be entered:-

```
|TIMEON,30,1,50                    [ENTER/RETURN]
```

In mode 2 this would be :-

```
|TIMEON,70,1,50                    [ENTER/RETURN]
```

The RATE parameter is in units of 1/50th of a second, thus a value of 50 would result in the time being displayed every second, A RATE value of 500 would result in the time being displayed every 5 seconds.

If the command is used without parameters then the previous values are used.

|TIMEOFF.

This command is used in conjunction with the |TIMEON command and simply turns off the automatic time display function. Simply type :-

```
|TIMEOFF                           [ENTER/RETURN]
```

BASIC DEMO - SIMPLE TIMER.

The following program shows how you could use the RTC as a simple timer. It does this by using the seconds, from the RTC, to count how long a program loop takes to execute. Try the following program :-

```
10 a%=0
20 |settime,a%
30 for n=1 to 9000 : next n
40 |asktime,@a%
50 print "time taken ";a%;" seconds"
```

In the above program the seconds value of the RTC is set to zero in line 20. A simple delay loop is the executed in line 30 after which the elapsed time is obtained in line 40. The result is then printed in line 50. Try altering the value '9000' in line 30 for different results.

|ALARMON

This routine uses the time of day alarm facility within the RTC chip. It takes the following format:-

```
|ALARMON,HOURS%,MINUTES%,SECONDS%,@F%
```

All 4 parameters must be used, where the HRS%, MINS%, SECS% is the alarm time and the Variable F% is the alarm Flag.

Thus if you wish to set the alarm time to 9am exactly then you would enter the following:-

```
    F%=0
    |ALARMON,9,0,0,@F%
```

The variable F% will remain at 0 until the alarm time is reached when it will be changed to -1.

The alarm can be set to operate from once per day, as in the above example, to once per second. This is achieved by setting a 'DON'T CARE' code, (255), in the appropriate locations.

An alarm every hour would be selected by setting the hours value to 255 with the minutes and seconds set as required. Similarly an alarm every minute would be selected by setting the hours and minutes to 255 with the seconds set as required. Finally an alarm every second would be selected by setting the hours, minutes and seconds to 255, as in the example below.

```
    10 F%=0
    20 |ALARMON,255,255,255,@F%
    30 IF F%=0 THEN GOTO 30
    40 SOUND 1,50
    50 GOTO 10
```

In the above program line 10 resets the alarm flag, F%, to 0. Line 20 enables the alarm function. Line 30 waits until the alarm time has been reached, in this case every second. Line 40 makes a sound and line 50 starts the whole process again.
Note that the alarm function is a 'one-shot' operation. Every time an alarm is required then the |ALARMON RSX must be used.

Try changing line 20 to:-

```
    20 |ALARMON,255,255,13,@F%
```

This has the effect of giving an alarm every minute at precisely 13 seconds into the minute.

|ALARMOFF

This utility simply turns off the ALARMON function.

CP/M PLUS UTILITY

The CLOCK.COM program is for use under CP/M PLUS. It allows the user to modify the RTC time/date and also update the internal time/date locations. Thus if the user wishes to set up the internal clock, as used for date stamping, then all that is necessary is to type in CLOCK then press the [RETURN] key.
Note that the internal clock is updated immediately after the CLOCK routine is used, the internal software clock then takes over by updating

the time once per second. Clearly, as with all software clocks, the time will gradually go wrong but since DATE STAMPING does not use seconds then a few seconds either way does not matter.

If the user wishes to alter the time/date to:-

5th Jan. 1987   12 Hrs 15 Mins 35 Secs

then the following should be entered:

```
CLOCK 05/01/87 12/15/35               [ENTER/RETURN]
```

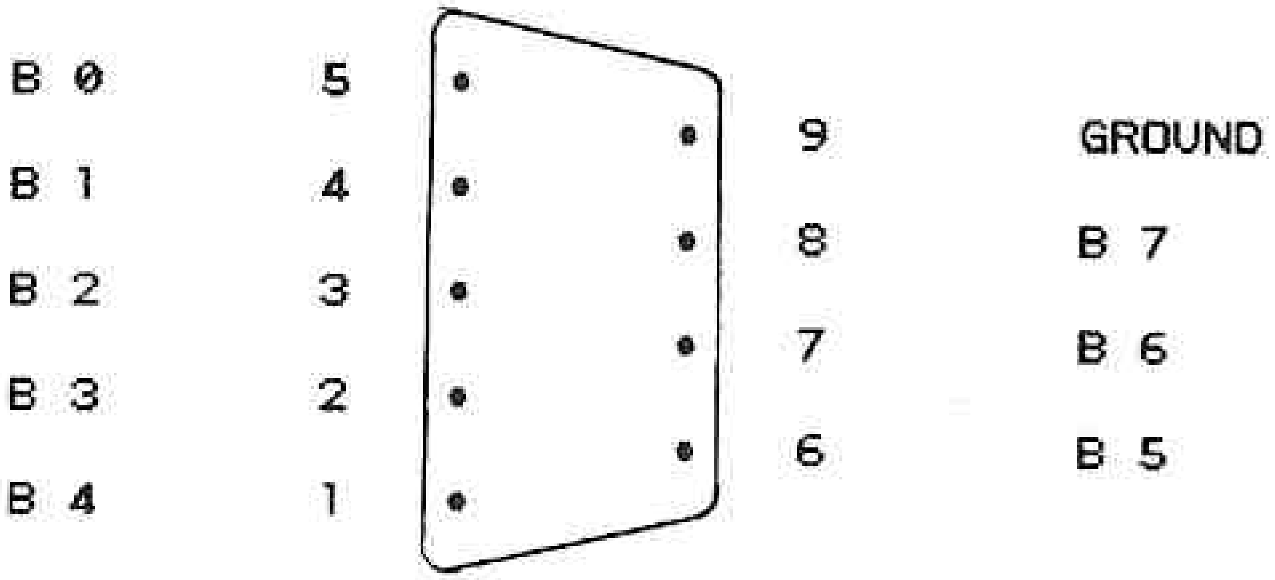
You are now asked to 'PRESS ANY KEY TO SET THE TIME'. When it is the exact time press any key. When the key is pressed the RTC time is updated to the time given then the internal clock locations as used by CP/M PLUS are updated to the correct time.

It is possible to alter your system disc so that every time you 'BOOT' up CP/M PLUS the CLOCK.COM program is run automatically. To do this you have to create a file called PROFILE.SUB. This file simply would be an ascii file with the words CLOCK.COM in it, which is created by using a text editor program such as ED.COM.
Thus every time CP/M PLUS is entered the internal clock locations will be set up to the real time from the RTC module. If the RTC module is not fitted then the appropriate error message will be given.

8-WAY I/O PORT

The RTC module contains a Z80 PIO chip to control the RTC chip via port A. **Do not use port A as this will affect the RTC operation.**

Port B is connected directly to a standard 9-way 'D' connector on the left of the interface. The pins are assigned as follows:-



The PIO is mapped in the I/O space at the following locations:-

0FBE0 Hex	Port A data, used for RTC chip
0FBE2 Hex	Port A mode control register
0FBE8 Hex	Control port for RTC chip (4 bit latch)

0FBE1 Hex	Port B data, used for external control
0FBE3 Hex	Port B mode control register

Here we are only interested in the PIO port B. The PIO can be set to various modes, here we will discuss mode 3 in detail. Users are advised to obtain details of the Z80 PIO if the other modes are to be used.

Mode 3 is used because individual bits can be set to act as either input or output. You might want to select some bits for input and other bits as outputs, this is achieved by first writing a control word to the control port followed by the data for the various input/output bits. If the data bit is '0' then that bit will be set to output mode, conversely if the bit is '1' then that bit will be set to input mode.
For example if you wish to select bits 0-3 for output use and bits 4-7 as inputs then the following program would achieve this.

```
    10 OUT &FBE3,255          :  REM PORT B CONTROL - MODE 3
    20 OUT &FBE3,240          :  REM BITS 0-3 OUTPUT ('0')
                                     BITS 4-7 INPUT  ('1')
```

Now if you want to set bits 1 and 3 high then you would write to the data port, ie:-

```
    30 OUT &FBE1,10
```

Note that this does not affect the bits that are set as inputs.
Now we may wish to read in the data for bits 4-7. This is achieved by the following:-

```
    40 A=INP(&FBE1)
```

Here all the bits 0-7 are read, but the bits that are set to act as outputs will return meaningless data, thus the program should ignore these bits. Eg

```
    50 D=A AND 240            :  REM THIS WILL FORCE BITS 0-3 TO ZERO
```

Or the program could simply test for the individual bits 4-7.

BATTERY

The unit contains a Ni-cad battery which is trickle-charged while the computer is switched on. All the data in the RTC chip will be retained for several months even if the computer is not used in this time.

MACHINE CODE PROGRAMMERS

The control of the RTC chip is fairly complex in that the PIO port A is used as both input and output configurations to communicate with the RTC data BUS. Also a further I/O port is used to give the RTC the correct signals during read/write operations. This is because the RTC chip was not designed to run with the Z80 directly.

Machine code programmers may however access the RSX routines to obtain the desired effects. These are as follows:-

SETTIME	START + 162
ASKTIME	START + 165
PEEKRTC	START + 168
POKERTC	START + 171
TIMEON	START + 174
TIMEOFF	START + 177
ALARMON	START + 180
ALARMOFF	START + 183

Where START is the address where the RSX code has been located, this address is printed during the installation program.
The entry conditions are the same as described in the firmware manual under RSX's. That is the ACCUMULATOR holds the number of parameters passed and the IX register points to their address. It is strongly suggested that the machine code programmer understands RSX's before attempting to use the above routines outside BASIC.

ADDRESS MAP

As mentioned earlier the RTC has 64 memory locations where the first 14 are used by the clock function and the remainig 50 are for general purpose use. These are organised as follows:-

LOCATION	FUNCTION	VALUE	
0	SECONDS COUNT	0 - 59	
1	SECONDS ALARM VALUE	0 - 59	
2	MINUTES COUNT	0 - 59	
3	MINUTES ALARM VALUE	0 - 59	
4	HOURS COUNT	0 - 23	
5	HOURS ALARM VALUE	0 - 23	
6	DAY OF THE WEEK	1 - 7	1 = SUNDAY
7	DATE OF MONTH	1 - 31	
8	MONTH	1 - 12	
9	YEAR	0 - 99	
10	CONTROL REGISTER A		
11	CONTROL REGISTER B		
12	CONTROL REGISTER C		
13	CONTROL REGISTER D		
14 - 63	50 BYTES OF RAM		

NOTES