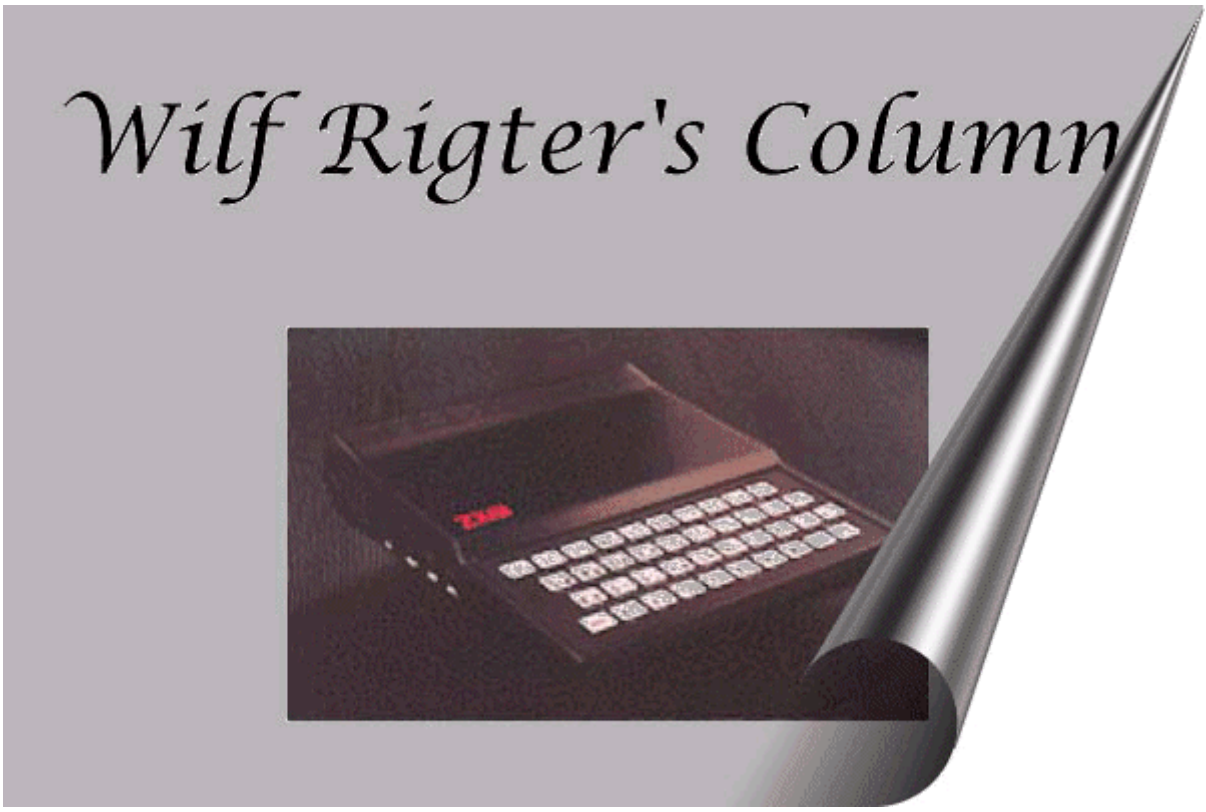
Tech/Worlde Presents



Welcome!

As an exclusive to readers of this launch edition of this brand new column, Wilf has given permission for the following article on his latest hardware project to be published here in its entirety.

He's literally just finished writing it and posted it over to me whereupon, I have edited it for HTML.

By the way, I was going to put up a MAJOR new article that Wilf had sent over to me in BETA form and given permission for me to post here - but due to the subjects significance and importance (I don't think there has ever been a work like this before), I have contacted Wilf - and decided not to post it until it is all finished.

It will be worth the wait! ...

Anyway, I know you will all find this articles contents fascinating and remember, you can't get it anywhere else - so without further delay - over to Wilf...

ZX81 REAL TIME CLOCK PROJECT

Written and developed by Wilf Rigter

Last revised: October, 1996

Hello there!

Here is a simple Real Time Clock (RTC) project I have recently built and tested on a ZX81 but should be compatible with the Spectrum and other Z80 machines. Cost and simplicity are the main reasons for choosing the DALLAS DS1287. These RTC modules were used on some 286 AT motherboards which can often be had for free and contain other useful parts for ZX81 projects. The DS1287 is functionally equivalent to the more common MC146818 but integrates all external components including crystal and battery in a single 24 pin DIP module. Check the date code on the unit to determine the remaining life of the internal lithium battery, which is normally good for 10 years or more. An internal flag can also be used to verify a good battery. The DS1287 is designed to work with multiplexed address/data bus MPU's like the 6805 or the 8088. The databook shows an example of a 68000 application but there are no Z80 application examples given. This may be the reason it is seldom used in Z80 designs although the interface is straight forward. Rather than multiplexing the data and address, I used two separate IO addresses: one for the address port and one for the data port.

The DS1287/Z80 interface programmer model is a block of 80 bytes, each of which can be selected by writing a byte address (0 to 79 decimal) to the address port and reading or writing data for that byte through the data port. The 74HC138 decodes IO addresses 1F,3F,.,FF any two of which (except FF) can be assigned to the RTC. In this example I have used 9F for the address register and BF for the data register. The AS address strobe and the DS data strobe are active high and CE is active low. The R/W pin is connected to Z80 inverted RD line for timing purposes. CS is active only when selecting data port. The reset pin may also be connected to the VCC line.

In this application I have connected an LED to the SQW output pin to provide a 2 Hz blinking indication light. The SQW frequency is controlled with REGISTER A bits 0-3.

The IRQ pin is connected to a piezo buzzer which turns on when the RTC bytes are equal to the ALARM bytes. The AIE alarm interrupt must be enabled in REGISTER B and when the IRQ output is active it can be reset by reading the data of REGISTER C. The unused 74HC14 Schmitt trigger inverters can be used as oscillators, latches or push-button conditioning for other applications.

I have included a brief introduction to the DS1287 specifications and some programming examples to get you started, however you should consult the MC146818 or the DS1287 data manual for more complex applications.

```
DS1287 IC pins:

Pin 1      Mode=1 for Motorola control bus timing used in this application.
Pin 4-11   AD0-7 used to load the internal address and data.
```

```
Pin 12     Vss connected to powersupply common.
Pin 13     CS active low chip select enables data read and write cycles.
Pin 14     AS active high loads internal address from AD0-7.
Pin 15     R/W select read or write cycle inverted RD is used to enable.
           write cycle : inverter required for minimum hold time.
Pin 17     DS active high read or write data to internal address.
Pin 18     RESET active low to initialize flags etc. but not RTC data.
Pin 19     IRQ active low programmable interrupt request (i.e. piezo alarm).
Pin 23     SQW user programmable frequency output (i.e. flashing LED).
Pin 24     Vcc connected to +5V powersupply.

Note:      M146818 uses additional pins for external crystal etc.
           Consult data book for complete details.


DS1287 and MC146818 programmer model:

The byte addresses and their functions are as follows:

00     seconds                 01     sec alarm
02     minutes                 03     min alarm
04     hours                   05     hr alarm
06     day of week             07     day of month
08     month                   09     year
10     Register A              11     Register B
12     Register C              13     Register D
14-79  user bytes
```

The RTC bytes (byte 00,02,04,06-09) are updated with current data once a second at which time the alarm bytes (01,03,05) are compared to the RTC. The alarm bytes can be loaded with a don't care code (192 decimal) to generate alarms more than once a day (i.e. once a minute or once an hour).

```
REGISTER A is a read/write control byte with the following functions:

bit0-3     timebase divider selection i.e. 1111 = 2 Hz square wave
bit4-6     oscillator start code = 010 and any other code to stop
bit7       UIP update in progress bit : clock data is valid when low
```

Register A can be initialized by with a value of 47 to start the oscilator and to select a 2 Hz square wave output. The RTC is updated once per second and the UIP bit goes high 244 us before the RTC data is changed. To avoid errors of reading changing data during the update, read data only if the UIP bit is low.

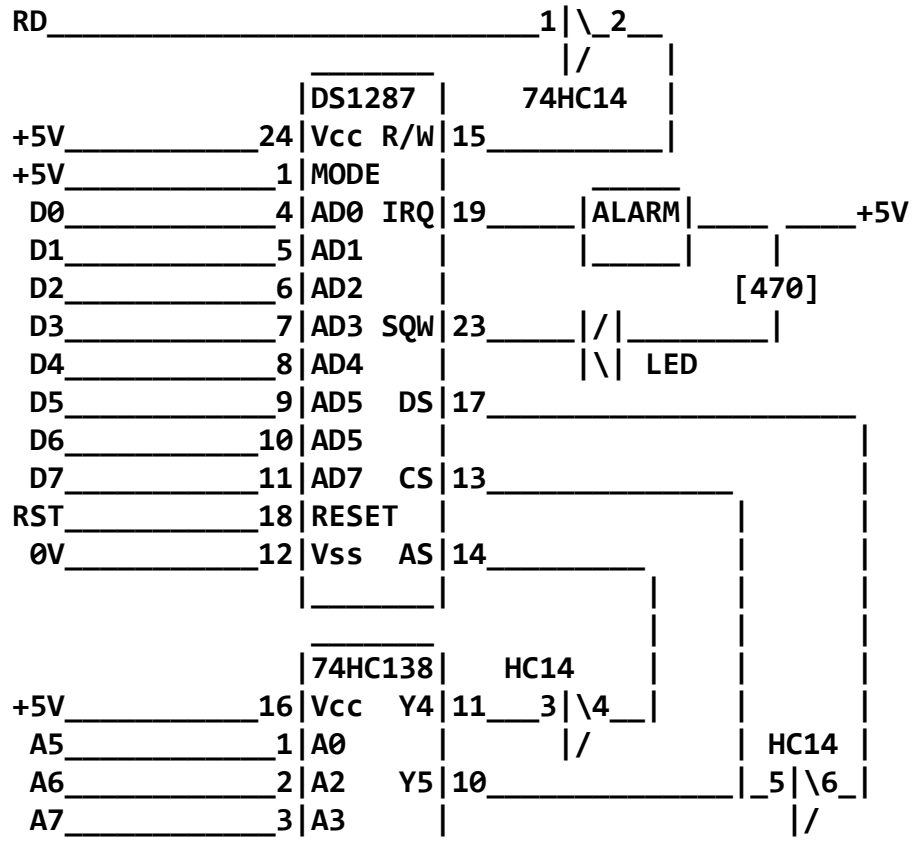
```
REGISTER B is a read/write control byte with the following functions:

bit0       DSE  =1 daylight saving time in effect
bit1       24/12=1 enable 24 hour format
bit2       DM   =1 RTC data in binary / DM =0 RTC data in BCD
```

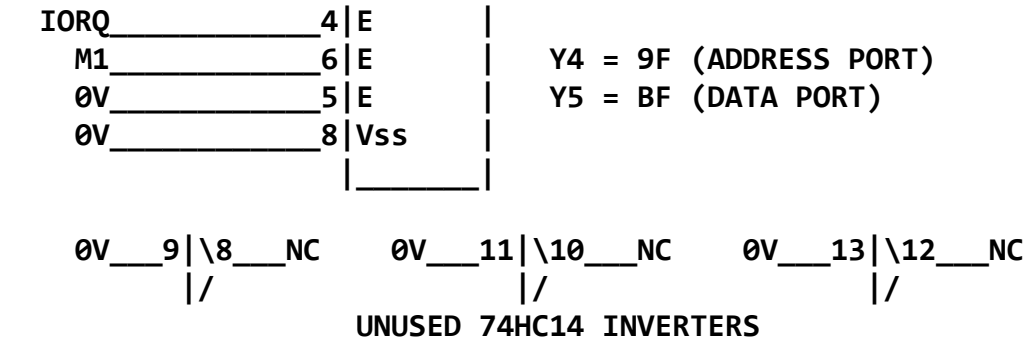
```
bit3      SQWE =1 enable square wave output
bit4      UIE  =1 enable IRQ generated at the end of the update cycle
bit5      AIE  =1 enable IRQ generated by RTC alarm byte comparator
bit6      PIE  =1 enable IRQ generated from timebase. (ie 2 Hz)
bit7      SET  =1 used to freeze RTC updates. Normally  SET =0

REGISTER B is initialized to 14 decimal in our simple example

REGISTER C is a read only byte with the following functions :

bit0-3    N/C  =0  bits not used
bit4      UF   =1  update flag indicates update cycle is finished
bit5      AF   =1  alarm flag indicates alarm comparator match
bit6      PF   =1  periodic flag set every new square wave period
bit7      IRQF =1 IRQ generated by any of the other status flags
```

The ZX81 does not provide user defined interrupts but the DS1287 IRQ pin may be used to drive a small piezo beeper for alarms etc. Reading register C causes the IRQ pin to be reset.

```
Register D is read only  status byte with the following function:

Bit0-6    N/C  =0 bits not used
bit7      VRT  =1 indicates the internal lithium battery is good
```

DS1287 REAL TIME CLOCK CIRCUIT FOR THE ZX81 AND SPECTRUM





ASSEMBLY CODE:

Note: in this simple example, RTC data is read regardless of update status

```
                        ;STARTS AT +16516
ADRS    LD A,XX         ;POKE +16517,INTERNAL BYTE ADDRESS
        OUT 9F          ;WRITE TO ADDRESS PORT
        RET

WR      LD A,XX         ;POKE +16522, BYTE DATA
        OUT BF          ;WRITE TO DATA PORT
        RET

RD      IN A,BF         ;READ DATA PORT
        LD C,A          ;PASS DATA VIA REGISTER BC
        LD B,00         ;BACK TO BASIC
        RET
```

DECIMAL CODE STARTING IN 1 REM LINE STARTING AT 16514:

118,118.

BASIC CODE: RUN 1000 TO INITIALIZE: RUN TO ENTER DATA ETC.

```
10      PRINT "ENTER W OR R FOLLOWED BY ADDRESS 0 TO 79 (IE W2=MINUTES)"
20      INPUT A$
30      CLS
40      POKE 16516,VAL (A$(2 TO))
50      RAND USR 16516
60      IF A$(1) = "R" THEN PRINT "BYTE " ; A$ (2 TO) ; " = " ; USR 16526
70      IF A$(1) <> "W" THEN GOTO 10
80      PRINT "ENTER DATA FOR BYTE" ; A$ (2 TO)
90      INPUT A
100     POKE 16522,A
110     RAND USR 16526
120     GOTO 10

REM     REGISTER A INITIALIZED FOR 2 Hz ON SQW PIN
1000    POKE 16517,10
1010    RAND USR 16516
1020    POKE 16522,47
1030    RAND USR 16521
```

```
REM     REGISTER B  UPDATE/BINARY DATA/24HR FORMAT/ALARM ENABLED
1040    POKE 16517,11
1050    RAND USR 16516
1060    POKE 16522,10
1070    RAND USR 16521
1080    GOTO 10
```

END OF ARTICLE
****

If you have any questions or comments on this site to pass onto Wilf, you can contact him below.

Thank You.

e-mail: rigter@cafe.net

Please check this page regularly for updates! - There is a lot planned for this site...

Main Menu - What's New - The Sinclair ZX81 Support Page - Windows 2000 - Links