

Time Keeping Using a COP800 Microcontroller

National Semiconductor
Application Note 823
Mike Werstlein
November 1997



Timekeeping Using a COP800 Microcontroller

ABSTRACT

Many applications require the time of day be available to the system so that a time stamp can be associated with an event. This application note discusses how National Semiconductor's COP820C microcontroller can be used to implement a timekeeping function. In addition, this paper demonstrates how two COP472 LCD display drivers are cascaded together to drive an 8 digit LCD display, which is used to display the time of day.

INTRODUCTION

Most often applications which use microcontrollers require some way of obtaining the time of day. Examples of these such applications are security systems, automatic sprinkler systems, telephone answering machines, automatic coffee makers, automatic thermostats, etc. One way of keeping track of the time and date is to use a dedicated real time clock IC. Although this is an easy and accurate option to implement, it is not very cost effective. The most cost effective method of timekeeping is to have the microcontroller itself perform the timekeeping function.

Timekeeping using the COP820C microcontroller is a relatively simple task. The system described here consists of a COP820C microcontroller, two COP472 LCD display drivers, and a Hamlin 8 digit LCD display (model #4216). The COP820C uses a 32.768 kHz crystal.

The timekeeping system described in this application note has many capabilities. The display can be toggled between two modes. One mode displays the time in the following format: HH-MM-SS (hours-minutes-seconds). The other mode displays the date in the following format: YR-MTH-DAY (year-month-day). Each part of the time or date can be individually set. In other words, one doesn't have to increment the seconds past 59 in order to increment the minutes by 1. Also, the real time clock automatically determines leap years and adjusts itself accordingly.

The following sections will detail the hardware and software portions of the real time clock.

HARDWARE

As previously stated, the hardware consists of a COP820C microcontroller, two COP472 LCD display drivers, a Hamlin 8 digit LCD display, and four normally open pushbuttons. These buttons are used to set the time/date and to select the display mode, either time or date. The schematic for the system is shown in *Figure 1*.

The four pushbuttons are interfaced to the COP820C via the lower four pins of the I port. 1 k Ω pullup resistors are used to keep the I lines high until a button is pressed, in which case the corresponding pin is pulled low.

G0 and G1 are used as chip select lines for the two COP472's. G0 is used for the slave while G1 is used for the master COP472. The technique used to cascade two COP472's together is detailed in the COP472 data sheet.

SOFTWARE

The software portion of the real time clock consists of several subroutines used to perform special functions. The program uses approximately 600 bytes of ROM. Included in the program are the routines to display the time/date to the LCD. There are also some overhead routines used for the LCD. This means that the clock portion of the program is actually less than 600 bytes.

The program relies on the 32.768 kHz crystal oscillator as its time base. The 16-bit timer is loaded with a value that will correspond to one underflow per second. This underflow will cause an interrupt to occur that will increment the seconds value by one.

The beginning of the program is used to initialize certain RAM locations and also to configure the proper I/O ports and timers.

The first subroutine to be called is labeled SYNC. This routine is used to synchronize two COP472 LCD displays. One COP472 can drive 4½ digits. Therefore, in order to display more data, additional COP472's can be cascaded together to drive more digits.

The next section of the program is used to set the time/date information. The first step is to read the value of port I. Since only I0–I3 are used, the value of port I is AND'ed with 0F hex to mask out the upper four bits. If a button is pressed, a status bit in the CLKSTAT register is checked to see if the clock is displaying the time or the date. The correct part of the time/date is then set accordingly. The program keeps scanning the buttons until one of them is pressed or a timer interrupt is encountered, at which time the appropriate routines are executed.

When the time underflow interrupt occurs (once per second) the program jumps to location 0FF hex. Here the program jumps to the timer interrupt service routine if the timer interrupt pending bit is set. The timer interrupt service routine simply increments the seconds value by one. It also takes care of incrementing the minutes, hours, days, months, and years as needed. Leap years are also accounted for during this routine.

The next subroutine to be called is labeled ADJUST and is used to account for the clock inaccuracies. This is necessary because the instruction cycle time is 1/3276.8 seconds instead of the ideal time of 1/3277 seconds. An integer value for the instruction cycle time is considered ideal. The reason it is ideal is because if this value is loaded into the timer, then the underflow will occur exactly once per second. This inaccuracy can be fixed easily. By using a generic counter, one second can be added to the time whenever the clock becomes one second behind.

The next subroutine to be called is BCDDIG. This routine is used to convert the time/date information into a format that is usable by the COP472's. Memory locations 01–0A contain the COP472's digit data.

AN-823

The subroutine CONVRT is then called. This routine is used to convert the data contained in memory locations 01–0A into the proper format for the Hamlin LCD. This conversion routine might vary depending on the particular display being used.

The final subroutine to be called is DISP. This routine is used to send the contents of memory locations 01–0A to the COP472's so that the new time/date information can be updated on the display. The program then returns to the main loop at the beginning.

CONCLUSION

As can be seen, the COP820C can easily incorporate time-keeping functions into virtually any existing program. This approach can save the user the cost and time of adding additional circuit components to the overall system. Although this program takes about 600 bytes of memory, a significant portion of the program is overhead for the COP472's and the LCD display. Various parts of the program might not be necessary in a specific application.

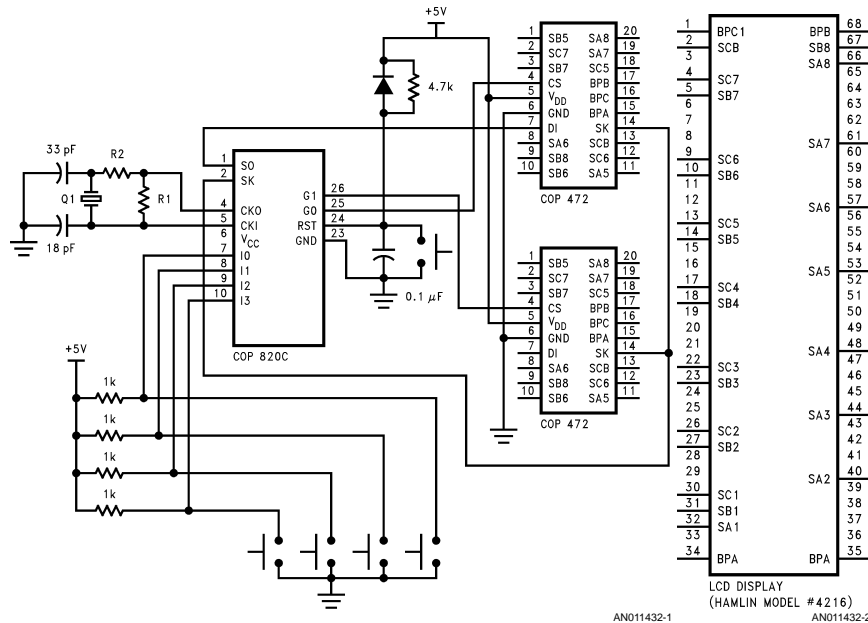


FIGURE 1. System Schematic

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com

www.national.com

National Semiconductor Europe

Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 1 80-530 85 85
English Tel: +49 (0) 1 80-532 78 32
Français Tel: +49 (0) 1 80-532 93 58
Italiano Tel: +49 (0) 1 80-534 16 80

National Semiconductor Asia Pacific Customer Response Group

Tel: 65-2544466
Fax: 65-2504466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.

Tel: 81-3-5639-7560
Fax: 81-3-5639-7507