Hardware-Based LED Blinking Control Eliminates Software Overhead

KEN WHITELEATHER | Sparton Electronics, Gov't Systems, DeLeon Springs, Fla.
KWhiteleather@sparton.com

LEDs are often used on man-
machine interfaces (MMIs) or device front panels to illuminate switches or backlight text information so that the user knows what the device is supposed to be doing at the time. Occasionally, the designer wants these LEDs to start blinking at a particular rate to alert the user to a condition needing attention, such as a low battery.

Many devices use embedded microprocessors or microcontrollers to handle the MMI, along with all other command and control functions for the device. The simple task of causing the LED to blink may add undesired overhead to the software design. However, this task can easily be offloaded to hardware by using the following technique.

Memory-mapped hardware-control registers are commonly used to turn

multiple LEDs on or off with one instruction. Similarly, memory-mapping a "blink-control" register can set a mask to control which LEDs should be blinking at any particular time, if the output of each bit of that register gates a low-rate (i.e., 1-Hz) clock signal for blink timing. By connecting the gated signal to the LED using a tri-state buffer that's enabled via the LED control signal, the LED can be turned on, off, or on-blinking *(Fig. 1)*. If the blink function isn't needed, an LED can simply be turned on or off using the simplified circuit in Figure 2.
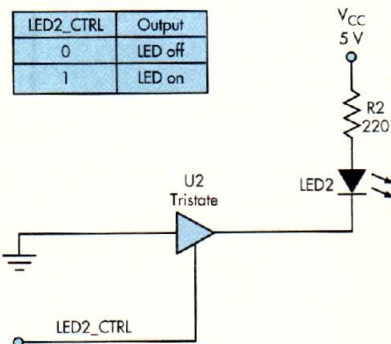
ED ONLINE 16764

KEN WHITELEATHER, *senior electrical engineer, received a BSEE from the University of Florida, Gainesville.*

LED2_CTRL	Output
0	LED off
1	LED on

2. If the blinking function isn't needed, the LED control circuit can be simplified as shown.

LED1_CTRL	BLINK1_CTRL	Output
0	0	LED off
0	1	LED off
1	0	LED on
1	1	LED blink at 1 Hz
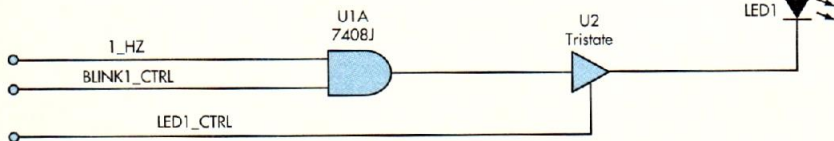
1. Using a simple hardware solution to make an LED blink eliminates the need for software overhead in the device's main control program.