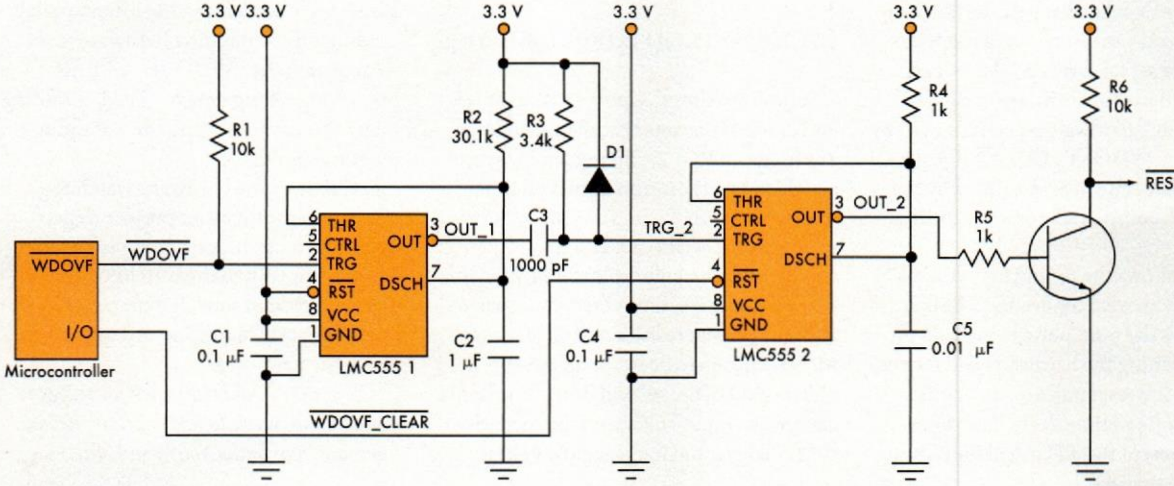
LMC555 Timers Delay Hardware Reset To Collect Debug Data

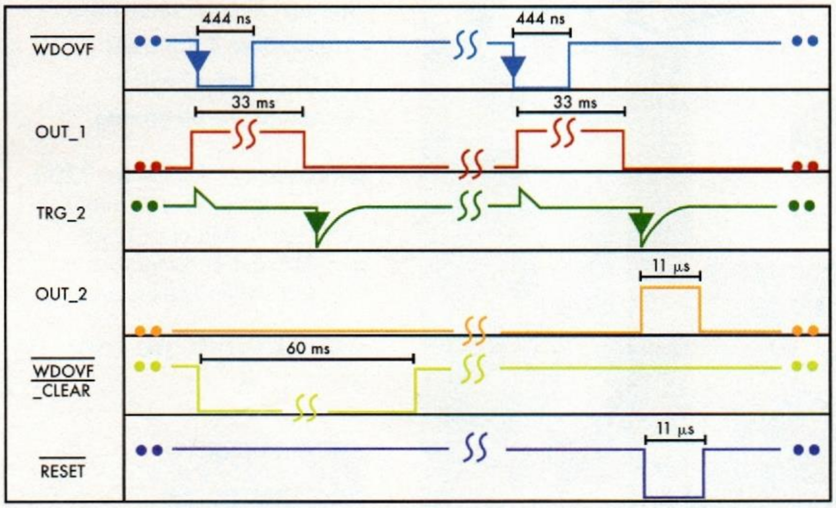
KEN TUROCY
DIEBOLD INC., NORTH CANTON, OHIO
ken.turocy@diebold.com

ED ONLINE 18419



1. The first LMC555 detects the watchdog-overflow trigger from the microcontroller and provides a delay. The second LMC555 allows the firmware to disable the impending hardware reset.

Most microprocessors have some type of hardware watchdog that can be used to reset the microprocessor and the surrounding hardware to a known state if firmware inadvertently stops executing code. But it may be advantageous to record the watchdog-overflow event in nonvolatile memory (NVM) for debug purposes prior to resetting the hardware.



2. This timing diagram for two separate watchdog events shows the reset pulse being held off and not resetting the hardware for the first event and then a reset pulse that occurs after a pre-determined delay for the second event.

The circuit in Figure 1 uses two LMC555 timers that delay a hardware reset in order to save valuable system data. It employs the microprocessor's dedicated watchdog-overflow pin, $\overline{\text{WDOVF}}$, and one I/O pin.

The LMC555 timers are configured in their monostable or one-shot mode, in which the timer's output is driven low until a negative pulse is sensed on the trigger-input pin. When triggered, the output pin goes high for a time, T_H, defined by a single resistor and capacitor—which is the following:

$$T_H = 1.1 \times RC$$

For proper timer operation, the trigger input must transition high again before the end of the output-high period. After T_H, the output goes low and remains there until

KEN TUROCY, principal electrical engineer, received a BSEE from the University of Akron, Ohio.

another negative pulse occurs on the trigger pin. The timer also has an active low-reset pin that, when asserted, holds the output low regardless of the trigger pin's input level. It also resets the timer's internal flip-flop for the next timing cycle.

The first LMC555 detects the watchdog-overflow trigger from the microcontroller and provides a delay that permits the firmware to record various system parameters before the reset takes place. The second LMC555 allows the firmware to disable the impending hardware reset. It does so by holding the $\overline{\text{WDOVF\_CLEAR}}$ signal low for at least the time period defined by the external RC components used by the first timer IC.

Figure 2 shows the timing for two separate watchdog overflow events. The first event shows the reset pulse being held off and not resetting the hardware, and the second shows the reset pulse occurring after a pre-determined time delay. The times shown represent the TH calculations using the RC component values from Figure 1 and the duration of the watchdog-overflow pulse from an Atmel microcontroller. TH for the first timer is:

$$(1.1)R2C2 = (1.1)(30.1\ k\Omega)(1\ \mu F) = 33\ ms$$

Similarly, T_H for the second timer is:

$$(1.1)R4C5 = (1.1)(1\ k\Omega)(0.01\ \mu F) = 11\ \mu s$$

Series capacitor C3, pull-up resistor R3, and diode D1 ensure that the second timer's trigger, TRG_2, will remain high until a high-to-low transition occurs on timer one's output, OUT_1. The pull-up resistor recharges the series capacitor to the supply voltage after the high-to-low transition on OUT_1. The RC time constant associated with the pull-up resistor and series capacitor was chosen to be less than the output-high period of the second timer. The diode clamps the input voltage on the second timer's trigger pin to the supply voltage.

In this example, when the watchdog-overflow event occurs, the microcontroller is configured to send a relatively short active-low pulse to a dedicated pin and create an internal interrupt. Once the event takes place, the code jumps to the interrupt-service routine and performs several specific lines of code. These include clearing the interrupt flag, holding off the pending hardware reset by asserting the $\overline{\text{WDOVF\_CLEAR}}$ I/O pin, saving debug data to NVM, disabling interrupts, and resetting the watchdog-overflow timer.

In this case, the hardware-watchdog event, along with other project-dependent information, is stored in NVM for later review. When the watchdog timer expires a second time, it bypasses the interrupt-service routine and ultimately resets the hardware.

The author would like to acknowledge the contributions made to this article by Bryan Simmons, Samantha Yoder, and Nick Harper.