design ideas

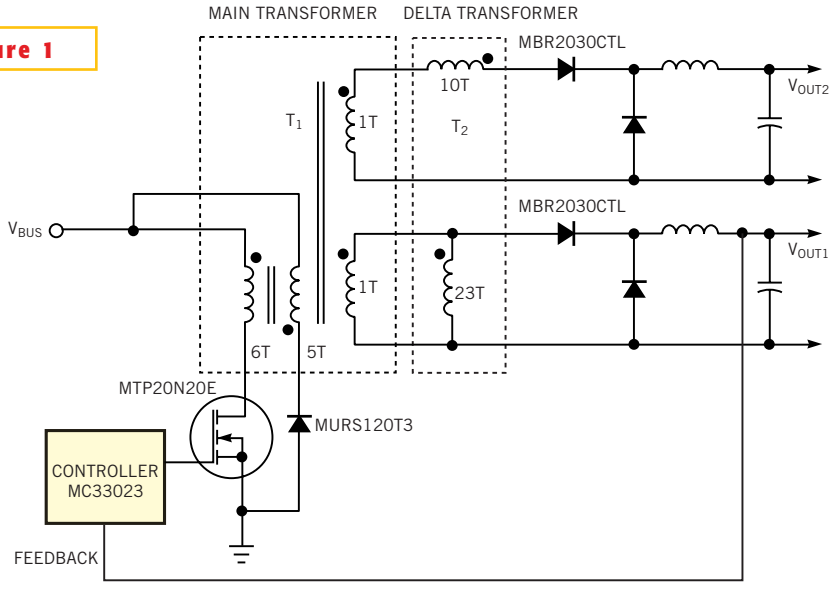
Edited by Bill Travis and Anne Watson Swager

Method sets voltage in multiple-output converters

Robert Bell, On Semiconductor, Phoenix, AZ

When you design a transformer for any power converter, you face several compromises. You must trade off core size against the number of primary turns and flux density. Another trade-off is the number of turns and winding resistance versus the associated losses. After making these trade-offs, you usually arrive at a good compromise that involves the primary and secondary turns. However, if the converter has more than one output, you face a new set of compromises. For high-power, low-output-voltage converters, the number of secondary turns is often very low. In a forward-converter topology, it is common for a 3.3V transformer to have one turn in its main secondary winding. This one-turn configuration is ideal for lowering winding resistance and associated power losses. For this design, the average output voltage is 3.3V per turn. So, if you need another output from the converter, that output is a multiple of 3.3V. For a multiple-output power converter, the ratio between the output voltages is often not a whole number (a problem known as "turns granularity"). Referring to this example, if the main output is 3.3V and the desired auxiliary output is 5V, two secondary turns yield 6.6V—a 32% error. A linear regulator could drop 6.6 to 5V but with the penalty of a power loss. **Figure 1** shows an approach to solving the granularity problem if the regulation requirement is not particularly tight (5 to 15%).

Figure 1



A delta transformer eliminates the problem of turns-ratio granularity.

Transformer T_1 is a normal forward transformer. Each secondary winding has one turn. The control loop regulates the main output, V_{OUT1}, to 3.3V. The objective is for the auxiliary output to be 5.5V. With only one secondary turn, that output will also be 3.3V. Consequently, you need a simple way to increase the voltage. You can add another transformer, T_2, dubbed a delta transformer, to the secondaries (**Figure 1**). The primary of the delta transformer is parallel with the V_{OUT1} winding, and the secondary of the delta transformer is in series with the V_{OUT2} winding. This connection has the effect of adding a portion of the main output voltage, V_{OUT1}, to the auxiliary output, V_{OUT2}. (The turns ratio determines the portion.) In the example above, suppose that the main transformer operates at a 50% duty cycle, and assume that the rectifiers have 0.6V forward voltage drop. Then, the equation relating V_{OUT1} and the transformer secondary voltage, V_{T1}, during the on time is: $3.3=(V_{T1}-0.6)(0.5)-(0.6)(0.5)$. Thus, $V_{T1}=7.8V$.

Now, you need to solve for the desired total V_T (V_{T2}) of the slave output, V_{OUT2}: $5=(V_{T2}-0.6)(0.5)-(0.6)(0.5)$. Thus, $V_{T2}=11.2V$. V_{T2} is the sum of the main-transformer secondary voltage and the delta-transformer secondary voltage. The

desired delta-transformer secondary voltage is 11.2−7.8=3.4V. Because the primary voltage of the delta transformer is also 7.8V, the turns ratio of the delta transformer must be $^{7.8}/_{3.4}$=2.3. In this example, you can use 10 and 23 turns for the delta transformer. The main-transformer secondary output delivers current only during its on time, and an internal resistive-voltage drop exists in the secondary output. Therefore, the volt-time product of the main transformer's secondary output is not exactly zero, which is a required condition for the delta transformer's primary to reset. So, you should make the primary winding of the delta transformer resistive to add a small voltage drop in the forward direction or use a small core gap. You can use this approach in all buck regulators to fine-tune an auxiliary output.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

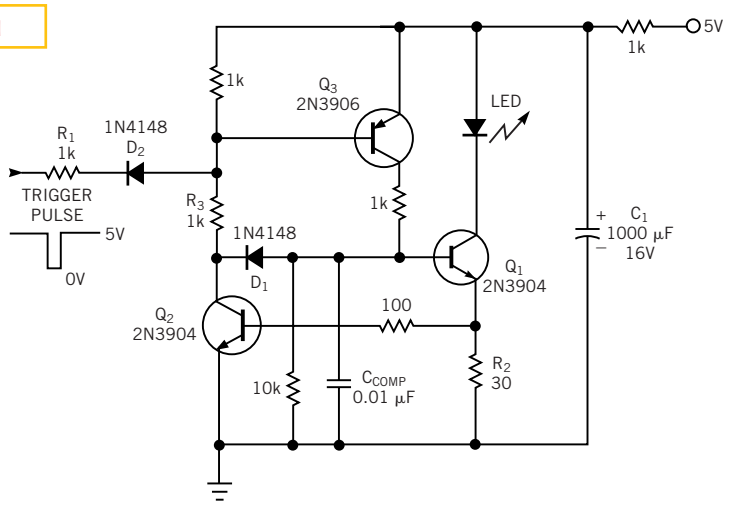
Circuit forms constant-current SCR

Robert Buono, Ringwood, NJ

A TYPICAL SCR (silicon-controlled rectifier) requires a trigger current, which causes the SCR structure to latch on. Once the device latches, the current through the SCR is solely a function of external component values. The SCR has no inherent ability to limit the current flow once it latches on. Current continues to flow, as long as the current exceeds a minimal value known as the holding current. The circuit in **Figure 1** is similar to an SCR, because it also requires a trigger current to latch into its on state. However, once latched, the circuit conducts a constant current. The constant current continues to flow, as long as the external circuitry can provide it, and the minimum compliance voltage of the SCR circuit is satisfied. When these conditions are no longer valid, the circuit latches off. The circuit in **Figure 1** provides a constant-current pulse to drive an LED with current sourced from a capacitor. You trigger the circuit with a narrow, negative-going pulse. The pulse, coupled through R_1 and D_2, turns Q_3 on. Q_3 provides base drive to Q_1. As Q_1 turns on, current begins to flow through the LED and current-sense resistor R_2.

When 0.6V develops across R_2, the current-limiting transistor, Q_2, begins to turn on and shunt base current from Q_1, through diode D_1. Q_2 thus maintains the current through R_2 at a constant level (~0.6V/R_2) by controlling the base current to Q_1. At the same time, because the collector voltage of Q_2 must be one diode drop lower than the base voltage of Q_1 while in constant-current mode, Q_2 also draws current through R_3. Q_2 thus maintains Q_3 in the on state (providing base current to Q_1), even after the trigger pulse disappears. The circuit maintains the constant-current mode, with Q_1 drawing a constant current through the LED, the storage capacitor C_1, and R_2 until Q_1 can no longer sustain the constant current. This situation occurs when the voltage across C_1 drops low enough to be unable to maintain 0.6V across R_2. Then, Q_2 begins to turn off, which allows Q_3 to turn off, thereby depriving Q_1 of base current. Q_1 turns off, which results in a constant-current (flat-topped) pulse through the LED with sharply rising and falling edges. By choosing the proper values of R_2 and C_1, you can easily control pulse width and amplitude.

Figure 1



Resembling an SCR, this circuit provides a constant current of controlled pulse width and amplitude to a load.

An apt application for this circuit is constant-current battery charging. Once you trigger the circuit, it provides constant current to charge a battery. When the battery charges to a point where the charging current falls below the constant-current level, the circuit latches off. Note that the circuit does *not* provide a continuous trickle charge, which could overcharge some batteries.

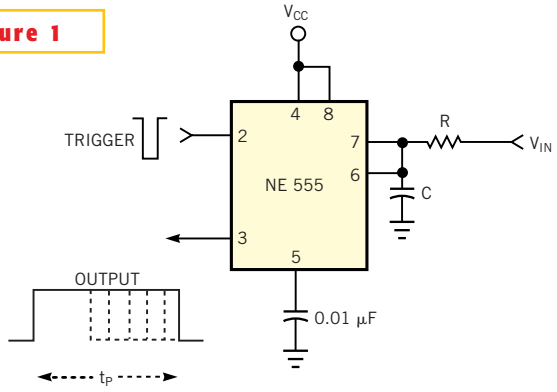
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

555 makes handy voltage-to-time converter

J Jayapandian, IGCAR, Tamil Nadu, India

THE CIRCUIT IN **Figure 1** is a simple, low-cost voltage-to-time converter using the ubiquitous 555 timer chip. You can use the IC's monostable multivibrator as a voltage-to-time converter by connecting the analog-voltage input to the charging resistor, R, instead of connecting R to V_{CC}. With this modification, the timer chip's output-timing cycle, t_p, is proportional to the input voltage, V_{IN}. When you apply an input voltage, the voltage across capacitor C charges exponentially according to the formula $V_C = V_{IN}(1-e^{t/RC})$, where RC is the time constant of the circuit, with C in farads and R in ohms. During one time constant, the voltage across the capacitor changes by approximately 63% of V_{IN}. The output timing of the monostable multivibrator is $t_p = 1.1$ RC. By keeping RC constant with fixed R and C values and varying the input voltage, V_{IN}, you obtain variable output timing. The output pulse width in this circuit is inversely proportional to the input voltage.

Figure 1



A voltage-controlled monostable multivibrator makes a handy voltage-to-time converter.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Program predicts VSWR-mismatch RF uncertainties

Steve Hageman, Agilent Technologies, Santa Rosa, CA

HEWLETT-PACKARD (now Agilent Technologies) once offered a useful little cardboard slide rule for calculating the uncertainty in RF measurements stemming from VSWR (voltage-standing-wave-ratio) mismatch. Unfortunately, this handy device is no longer available. A Visual Basic program accomplishes the same function on a PC, however. You can download the executable program and its associated setup utilities on a blind page at www.sonic.net/~shageman/vswr.html. Mismatch uncertainty is one of the most common calculations an RF engineer makes when determining the uncertainty of RF power measurements. The source and load VSWR interact along an unknown length of line to produce some uncertainty in the power measurement. This uncertainty stems from the fact that, at high frequencies, the length of a transmission line connecting a source and load may be sufficient to transform the impedance at one end of the line to another value at the other end.

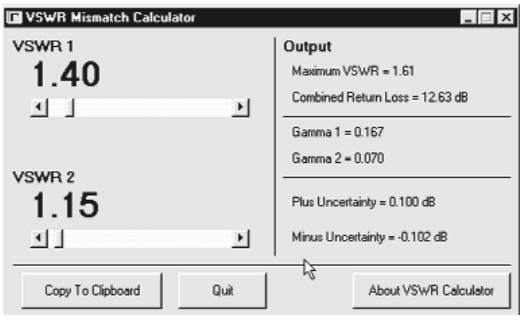


Figure 1 **Given the source and load VSWRs, the VSWR Calc program quickly calculates RF measurement uncertainties.**

System specifications usually include the VSWR values, which lack phase information. So, one certainty about a measurement is that it lies between some range of values. In reality, even the connectors and the transmission line in the measurement path add uncertainty because their true electrical length and, hence, phase is unknown. So, the true power at the load may be higher or lower than the measured value. The conservative way to account for this error is to assume that the phase is unknown and assume the worst case: The incident and reflected signals interact in the worst possible way—in other words, at the peaks and valleys. You express this scenario as $VSWR = E_{MAX}/E_{MIN}$, where E_{MAX} and E_{MIN} are the maximum and minimum voltages along the line. VSWR is a common specification in data sheets for RF devices,

such as amplifiers, sources, and power meters. VSWR relates to the absolute value of the reflection coefficient γ in the expression

$$\gamma = \frac{1 - \mathrm{VSWR}}{1 + \mathrm{VSWR}},$$

and, in turn γ relates to the return loss in decibels in the expression $R_L = -20\log_{10}\gamma$. Because the source and load each have a VSWR, the product of the two gives the maximum VSWR: $\mathrm{VSWR}_{MAX} = \mathrm{VSWR}_1 \bullet \mathrm{VSWR}_2$. The two VSWRs produce a combined return loss, as follows:

$$\text{COMBINED } R_L = -20 \bullet \mathrm{LOG}\left(\frac{\mathrm{VSWR1} \bullet \mathrm{VSWR2} - 1}{\mathrm{VSWR1} \bullet \mathrm{VSWR2} + 1}\right)\mathrm{dB}.$$

The uncertainty in the total measurement stemming from the source and load VSWRs is Uncertainty(+) = $20\log_{10}(1+\gamma_1 \bullet \gamma_2)$ dB, and Uncertainty(−) = $20\log_{10}(1-\gamma_1 \bullet \gamma_2)$ dB.

As a result, you have a range of either plus or minus uncertainty. At small VSWRs, the plus and minus converge to the same value. At higher VSWRs, the plus and minus uncertainties diverge, so you need to calculate both. As an example, consider a Hewlett-Packard ESG-3000 microwave source operating at 900 MHz. Its VSWR is specified at 1.4 to 1. Then, assume that you measure the source's output power with a Hewlett-Packard E4412A power sensor that has a specified VSWR of 1.15 to 1. If you input these figures into the VSWR Calc program, you obtain the screen shown in **Figure 1**. The "Copy to Clipboard" function transfers the VSWRs and the calculated data to the Windows clipboard so that documenting the calculations is easy in any Windows application. (The cardboard slide rule cannot perform this function.) **Figure 1** shows the clipboard data of this example. The uncertainty in the example is +0.100 to −0.102 dB. You should know the measurement uncertainty, because it is relatively easy to obtain totally uncertain measurements at high frequencies if the VSWRs are uncontrolled or unknown. The VSWR Calc program is a Microsoft Visual Basic 32-bit application that runs on Windows 95, 98, and NT 4.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

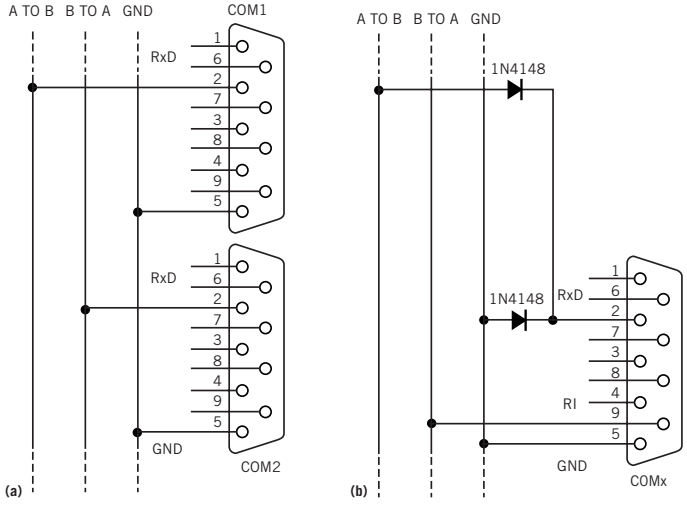
PC monitors two-way RS-232 transmission

Jerzy Chrzaszcz, Warsaw University of Technology, Poland

THE GOAL OF MONITORING transmission in a data link is obvious: You want to know the contents of the data, when it was sent, and by whom. If one of the communicating parties is a PC or another user-programmable controller, then you can modify parameter settings or, at worst, change transmission routines to generate log files or perform other actions. This approach, however, may be inconvenient or impossible to apply in some cases. As an alternative approach, you can use a PC with two serial ports and a monitor program to observe the link itself. The method in **Figure 1a** needs no access or knowledge of the communicating devices. A C program opens two COM ports and installs interrupt-service routines for IRQ4 and IRQ3. Upon the reception of an interrupt, the routine stores a byte in a common circular buffer with the COM identifier and error flags. The main program displays the contents of the buffer, indicating time intervals in milliseconds between consecutive transfers. Although the program simplifies the time measurement, it preserves the original byte order and correctly reflects time relationships as long as the main program keeps up with transmission speed. If you need greater precision, you can easily modify the program to record time stamps, along with the data and status bytes, in the circular buffer.

Unfortunately, not all PCs offer two COM ports. This deficiency is a common drawback of notebook computers, which use a second UART controller for IrDA communication. But you can use even

Figure 1



You can eavesdrop on RS-232 transmissions by using two COM ports (a); a simple modification (b) adapts the method to PCs with only one COM port.

these computers with another version of C to monitor the bidirectional link, provided that the transmission is not full-duplex. A simple interface mixes both data streams onto the receiver input (**Figure 1b**). One channel connects to the RI (ring indicator) input of the UART. Whatever the byte value, the start bit guarantees that the RITD (ring-indicator trailing edge) bit in the modem-status register is set. The interrupt-service routine reads the register, clears the RITD flag, and stores its value in a buffer. Thus, the interface is ready for another byte to come from an arbitrary direction. The main program can identify the data source by checking respective bits. You can download the C listings and executable files from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2661. The programs are simple and accept 9600, 8, E, and 1 transmission parameters. You can easily adapt the programs to other formats.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

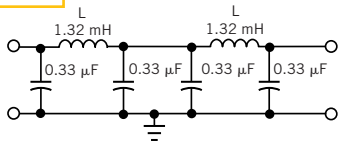
Passive filters fill the bill at audio frequencies

Richard Kurzrok, Queens Village, NY

Low-frequency filters, particularly at audio frequencies, usually take the form of active filters. These filters eliminate expensive inductors with windings of many turns. Both analog and digital active filters are most compatible with large-scale integration at the subsystem and system levels. However, passive filters remain a viable option when you quickly need low-cost prototypes and test pieces (**Reference 1**). These filters use no external dc excitation and require no complex pc boards. You can easily wind some filter inductors using manual techniques. Moreover, inductors can handle greater power levels than small-signal active devices. You can construct a simple lowpass filter with a 3-dB cutoff frequency of 10 kHz; a source/load impedance of 50Ω; five poles; and 0.02-dB-ripple, Chebyshev response. **Figure 1** shows the filter's schematic; **Table 1** provides the parts list.

Table 2 shows the measured frequency response with 50Ω source and load impedances. The extremely low passband-insertion loss indicates that the inductors' unloaded Q is greater than 100. You could use smaller inductors, such as toroids with 0.5- or 0.625-in. diameters with acceptable insertion losses (**Reference 2**). Note that expensive Litz wire is unnecessary. Lowpass filters need much lower inductor unloaded Q values than do most bandpass filters. At very low frequencies, both inductors and capacitors can become large. By using moderate filter-impedance levels, such as 50 or 75Ω at kilohertz frequencies, inductor values can be lower than 10 mH. With high-permeability inductor cores, fewer turns are required, and hand-winding is usually feasible. However, capacitors become large for lower filter impedances. For the traditional 600Ω impedance used at audio frequencies, inductors are larger by an order of magnitude. If you reduce the cutoff frequency from 10 to 1 kHz, the inductor values also increase by an order of magnitude.

Figure 1



A five-pole passive lowpass filter yields sharp cutoff characteristics and low ripple.

Acknowledgment

I acknowledge Ed Wetherhold (Annapolis, MD) for three decades' worth of significant work on low-frequency passive filters and related circuits.

References

1. Wetherhold, Ed, "Audio Filters for EN 55020 Testing," *Interference Engineers' Master*, 1998.

2. DeMaw, MF, *Ferromagnetic Core Design and Application Handbook*, Chapter 3, Prentice-Hall, 1981.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

TABLE 1–PARTS LIST FOR FIVE-POLE LOWPASS FILTER

Function	Value	Realization	Quantity
Inductors	1.32 mH	28 turns No. 26 on Fair Rite toroid No. 597700601-0.825-in. outer diameter ×0.525×0.25 in. thick	Two
Capacitors	0.33 µF	Polypropylene with 2% tolerance	Four
Connectors	BNC female	Four-hole panel receptacle	Two
Enclosure	Aluminum box	Hammond 1590B/Bud CU-124	One
Board	Cut by hand	Vector board 169PP44C1	One
Standoffs	Male/female	Amatom 9794-SS-0440	Six

TABLE 2–MEASURED FREQUENCY RESPONSE FOR LOWPASS FILTER

Frequency (kHz)	Insertion loss (dB)	Frequency (kHz)	Insertion loss (dB)
1	0.1	11	6.5
3	0.1	13	15
5	0.1	15	22.6
7	0.15	20	36.5
8	0.25	30	Greater than 50
9	0.6	To 1 MHz	Greater than 50
10	3.1		

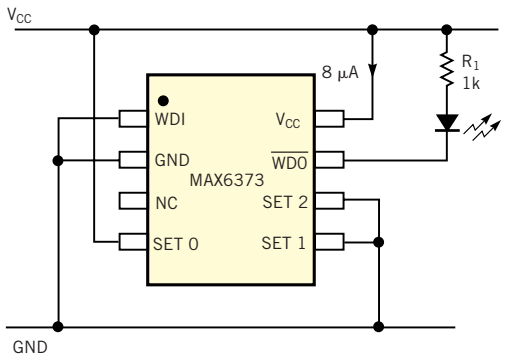
Watchdog timer assumes varied roles

Terry Millward, Maxim Integrated Products, Lambourn Hungerford, UK

The MAX6369-74 series of pin-selectable watchdog timers are designed to supervise µP activity and indicate when a system is working improperly. During normal operation, a µP should repeatedly toggle the WDI (watchdog input) before the selected watchdog-time-out period elapses to indicate that the system is properly executing code. If it fails to do so, the supervisor IC asserts a watchdog output $\overline{WDO}$ to signal that a problem exists. The cited family of watchdog supervisors are available in SOT23-8 packages and have selectable watchdog-time-out periods and delays of 1.7 msec to 104 sec in seven steps. The ICs also have selectable output-pulse widths of 1.7 or 170 msec, depending on part selection and the state of the: Set 0, Set 1, and Set 2 pins. You can use these devices for general-purpose timing functions, especially when low current consumption is important. The ICs consume only 8 mA typical and 20 mA maximum over temperatures from a 2.5 to 5.5V supply. With WDI connected to ground or V_{CC}, the internal timer cycles, pulsing $\overline{WDO}$ low upon time-out. In addition to the lower current (20 versus 120 mA), the watchdog-timer IC takes less board space and uses no timing resistors or capacitors. The following circuits represent a few examples.

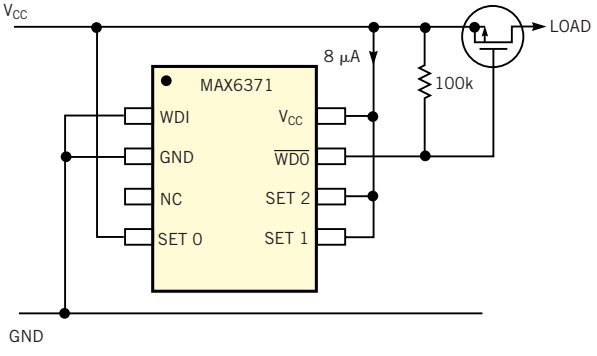
The circuit in **Figure 1** uses a MX6373 to pulse $\overline{WDO}$ low for 170 msec every 5.2 sec. The load is a front-panel power-on LED with a 1-kΩ current-limiting resistor. By pulsing the LED rather than powering it continuously, the average current decreases by a factor of 30 (88 µA versus 2.4 mA). The LED thus indicates that the equipment is on while minimizing battery drain. By changing the Set pins to Set 0=0V, Set 1=Set 2=V_{CC}, you can extend the off time to 17 sec, thus reducing the average current to 32 µA. The circuit in **Figure 2** is similar to the one in **Figure 1** but uses a MAX6371 to turn on a load for 170 msec every 104 sec. The load can be a battery-powered monitoring circuit that remains idle, saving power and then wakes up to make a measurement. The circuit in **Figure 3** uses a MAX6373 with its Set inputs configured for timer disabled. If you hold Set 1 low for longer than the watchdog period (5.2 sec), then $\overline{WDO}$ pulses low. You can use this circuit in applications in which a reset button is on a front panel, for example. You must deliberately depress the button for at least 5.2 sec to trigger a reset. This feature can prevent an accidental reset when someone inadvertently presses the button.

Figure 1



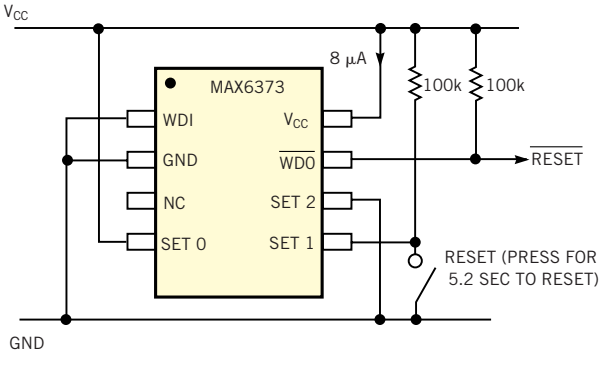
A blinking LED allows a 30-to-1 average-current reduction in a power-on indicator.

Figure 2



This circuit wakes up every 104 sec to turn on a load for 170 msec.

Figure 3



You must press the reset button for at least 5.2 sec for the reset to take effect.

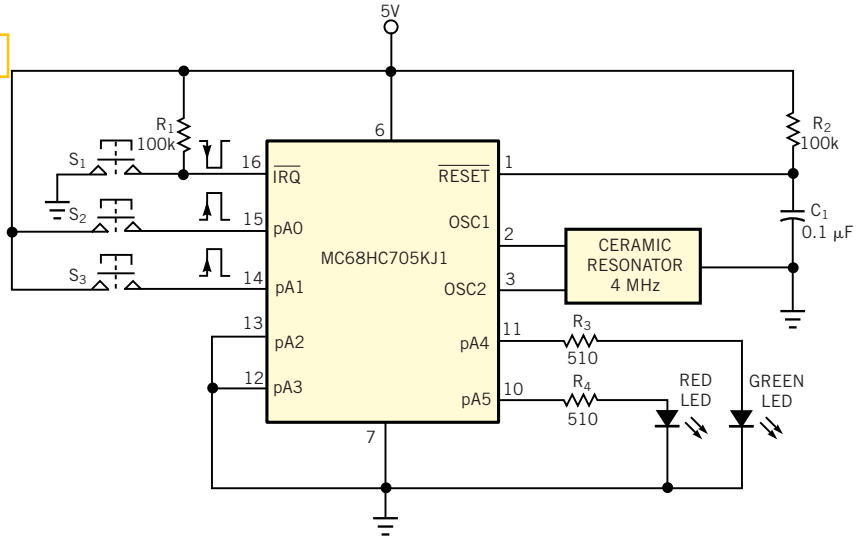
Is this the best Design Idea in this issue? Vote at www.edmag.com/ednmag/vote.asp.

One microcontroller serves multiple external interrupts

Abel Raynus, Armatron International, Melrose, MA

In designing µC-based systems, you often face a situation in which the µC has to respond to an external event happening at an uncertain moment in time. One example is receiving an echo from an object in a pulse-range measuring system. In these situations, you would usually use an external interrupt. Unfortunately, low-end, small, inexpensive µCs have only one external-interrupt vector address, so the µC can execute only one interrupt-service routine. What do you do if the design objectives call for the µC to react to several in-

Figure 1



You can use an inexpensive µC to handle multiple external interrupts.

LISTING 1—MULTIPLE-INTERRUPT TEST ROUTINE

```
                        1  ******** Multiple Interrupt Test
**********************
 0000                   2  $include "std-j1a.asm" ;
 0000                   3  $PAGEWIDTH 160
 07F1                   4          org    MOR;resistor osc,input pulldown,
 07F1      24           5          fcb    %00100100;pA0-pA3 ExtInt enable
                        6  *********  I/O PORTS
******************************
 07F2                   7  GrnLED  equ 4 ;pA4
 07F2                   8  RedLED  equ 5 ;pA5
                        9  *********  VARIABLES
******************************
 00C0                  10          org RAM
 00C0                  11  Disp    rmb 1 ;Reg. to set Int.Subroutine address
                       12  *********  INITIALIZATION
**************************
 0300                  13          org ROM
 0300 [02] A6F0        14  init    lda #%11110000 ;set I/O prtA
 0302 [04] B704        15          sta ddrA
 0304 [05] 3F00        16          clr prtA       ;LED's off
 0306 [05] 3FC0        17          clr Disp
 0308 [02] 9A          18          cli            ;Interrupt enable
                       19
*****************************************************
 0309 [05] 0900FD      20  main    brclr GrnLED,prtA,* ;wait for SW1 interrupt
 030C [02] A601        21          lda   #1           ;set address SW2
interrupt
 030E [04] B7C0        22        sta   Disp
 0310 [05] 0B00FD      23         brclr RedLED,prtA,* ;wait for SW2 interrupt
 0313 [02] A602        24          lda   #2           ;set address SW3
interrupt
 0315 [04] B7C0        25          sta   Disp
 0317 [05] 0A00FD      26          brset RedLED,prtA,* ;wait for SW3 interrupt
 031A [05] 3FC0        27          clr   Disp         ;set address SW1
```

```
interrupt
 031C [03] 20EB        28          bra   main         ;
                       29
*****************************************************
 031E [03] B6C0        30  ExtInt  lda   Disp
 0320 [03] 270E        31        beq   s1 ;execute subroutine 1
 0322 [02] A101        32        cmp   #1
 0324 [03] 270E        33        beq   s2 ;execute subroutine 2
 0326 [02] A102        34          cmp   #2
 0328 [03] 270E        35        beq   s3 ;execute subroutine 3
 032A [05] 120A        36  e0      bset  IRQR,ISCR ;ExtInt reset
 032C [06] CD033C      37          jsr   dly200  ;switch debouncing
 032F [09] 80          38          rti
 0330 [05] 1800        39  s1      bset  GrnLED,prtA ;Green LED on
 0332 [03] 20F6        40        bra   e0
 0334 [05] 1A00        41  s2      bset  RedLED,prtA ;Red LED on
 0336 [03] 20F2        42        bra   e0
 0338 [05] 3F00        43  s3      clr   prtA         ;LED's off
 033A [03] 20EE        44        bra   e0
                       45
*****************************************************
 033C [03] 4F          46  dly200  clra               ;delay for 200 ms
 033D [03] 5F          47  lp1     clrx
 033E [03] 5A          48  lp2     decx
 033F [03] 26FD        49        bne    lp2
 0341 [03] 4A          50         deca
 0342 [03] 26F9        51        bne    lp1
 0344 [06] 81          52        rts
                       53
*****************************************************
 07FA                  54          org    vectors+2
 07FA      031E        55          fdb    ExtInt
 07FE                  56          org    VECTORS+6
 07FE      0300        57          fdb    init
```

terrupts coming from different sources and to process each of them in a different way? **Figure 1** shows a design technique that solves the problem. The method is applicable to any µC, such as the 16-pin OTP MC68HC705KJ1 from Motorola. This µC has two options for handling external interrupts: via the IRQ pin triggered by a negative edge or via the pins pA0 to pA3 triggered by a positive edge. You can choose these options as well as the capability to have edge or edge-and-level triggering by setting the proper bits in the MOR (mask-option register).

When you set pins pA0 to pA3 as external-interrupt inputs, they connect inside the µC as an OR gate. Hence, you can trigger this µC from five external-interrupt sources. If the number of sources exceeds five, you can wire them through an OR gate to any of the external-interrupt pins. To illustrate the method in the simplest way, assume only three interrupt sources, represented by pushbutton switches S_1 to S_3 (**Figure 1**). You can simplify the interrupt-service routines to operate with only two LEDs. The use of the LEDs provides the opportunity to visualize and verify the interrupt process. After initialization, both LEDs turns off. The system waits for the first interrupt from S_1. As a result of the interrupt, the green LED turns on. The system again waits for the next interrupt from S_2, and the red LED turns on. During the waiting period, the µC can perform some function, which can differ for different projects. Service routines in real applications are much more complicated than just lighting LEDs. But those details are unimportant for illustrating this method.

The third external interrupt from S_3 switches off both LEDs, and the µC again waits for an interrupt from S_1. The limitation of this method is that the sequence of incoming interrupts must be known, but this constraint is unproblematic for most applications. **Listing 1** shows the µC program. The key to the method is to prepare the number-address of the interrupt-service routine for the next expected interrupt in the special register Disp (dispatcher). In this case, the µC executes every external interrupt with its own individual interrupt routine. The routine adds a delay of 200 msec for debouncing the switches; you can eliminate the delay if it is unnecessary for the interrupt signals. You can download **Listing 1** and associated assembly software from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2650.

Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.