design ideas

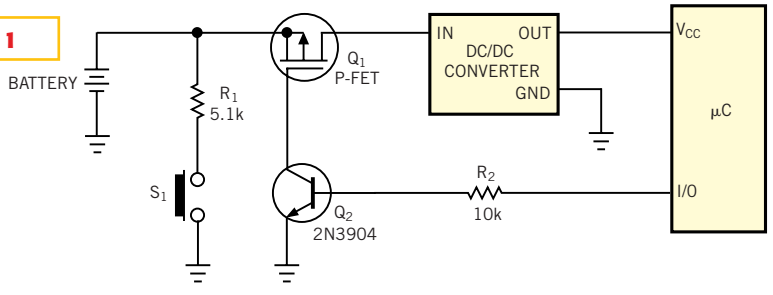
Edited by Bill Travis and Anne Watson Swager

Transistor latch improves on/off circuitry

Eugene Palatnik, SIMS-BCI, Waukesha, WI

Figure 1 shows an example of on/off circuitry commonly used in battery-operated devices. The p-channel MOSFET, Q_1, serves as a power switch. When you push the On button, S_1, Q_1's gate goes low. Q_1 turns on and supplies battery voltage to the dc/dc converter. Depending on the battery voltage in the device, the dc/dc converter might convert the voltage either up or down. In either case, it supplies V_{CC} to the µC. The µC goes through its power-up software sequence and programs one of its general-purpose I/O pins, setting it to logic one. This operation, in turn, causes saturation of the npn transistor, Q_2, which "confirms" the power-up state. Later, when the µC decides to power itself off, the µC simply sets its I/O output to logic zero, and Q_1 returns to its off state. The circuit is simple and reliable but has a significant disadvantage. It usually takes a fraction of a second for the dc/dc converter to reach its stable output voltage. Then, the µC's Reset pulse usually lasts 50 to 200 msec. After the release of Reset, the µC must go through its "housekeeping" start-up code before it has a chance to set its I/O pin to logic one. This delay in some portable systems may be user-unfriendly, because if you don't depress the On button long enough, the system will not power up. The circuit in **Figure 2** eliminates this uncertainty.

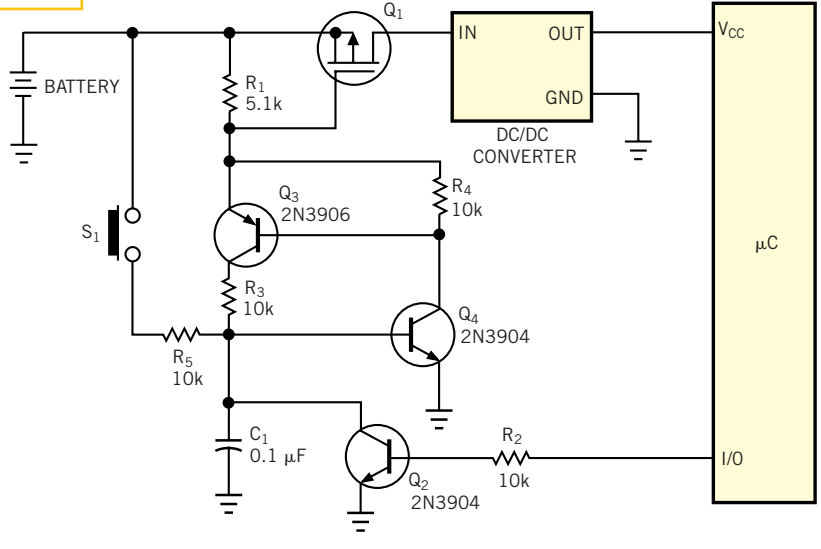
Figure 1



This on/off circuitry is effective but can suffer from turn-on ambiguity.

The circuit includes a simple two-transistor latch, which the On button switches to the on state. As in **Figure 1**, the p-channel MOSFET, Q_1, serves as a power switch. When you push the On button, S_1, it causes saturation of the npn transistor, Q_4, via the base-current-limiting resistor, R_5. The collector current of Q_4 flows through R_1 and the base-emitter junction of pnp transistor Q_3, thereby saturating Q_3. Q_3 redirects some current into the

Figure 2



A two-transistor latch provides positive turn-on when you push the button.

base-emitter junction of Q_4 and finishes the latching process. At this point, both Q_3 and Q_4 are saturated, and the voltage on the gate of Q_1 is a function of the voltage drop across the base-emitter junction of Q_3 and the saturation voltage of Q_4. This voltage is approximately 0.9V. The µC need not confirm the on state of the latch. When the µC powers up and finishes its housekeeping start-up code, it programs the I/O pin to logic zero.

Later, when the µC decides to power itself off, it programs the I/O pin to logic one and stops. Q_2 turns off Q_4, resetting the latch to its initial off state. R_4 lowers the equivalent input impedance of Q_3. This function improves EMI and ESD noise immunity and prevents the circuit from turning itself on in the presence of strong electromagnetic fields. Capacitor C_1 in combination with R_5 protects Q_4 and Q_2 from direct ESD into the pushbutton. Some portable devices use undervoltage-lockout circuitry. This circuitry usually uses a voltage comparator with a built-in voltage reference. If the battery voltage drops below the threshold, the output of the comparator (usually an open-drain type) switches low. If your portable system uses this type of circuitry, you can connect the open-drain output of the comparator in parallel with Q_2, thus preventing the latch from turning on if the battery voltage is too low.

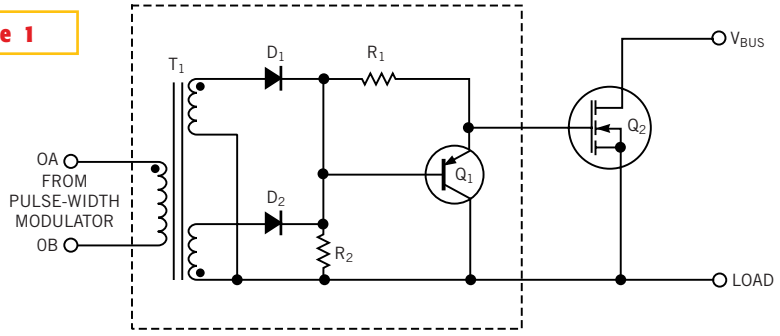
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

High-side driver feeds IGBTs and MOSFETs

Carlisle Dolland, Honeywell Engines and Systems, Torrance, CA

THE LOW- TO MODERATE-POWER system in **Figure 1** provides the interface between a pulse-width modulator and a high-side IGBT (insulated-gate-bipolar-transistor) or MOSFET switch. You can use it to interface TTL or CMOS circuitry to an H-bridge if you buffer it by a FET driver, such as an ICL7667 or an MIC4423. When OA is positive, D_1 conducts, charging the capacitance of the FET through R_1. The value of R_1 and the output impedance associated with the drive signal determine the turn-on time of the FET. After the capacitance is charged, the voltage across R_1 is essentially 0V, and Q_1 is off. During the PWM dead time, the gate capacitance discharges through Q_1 and R_2. R_2 and the h_{fe} of Q_1 determine the turn-off time of the FET. This circuit achieves turn-on and -off times of less than 150 nsec.

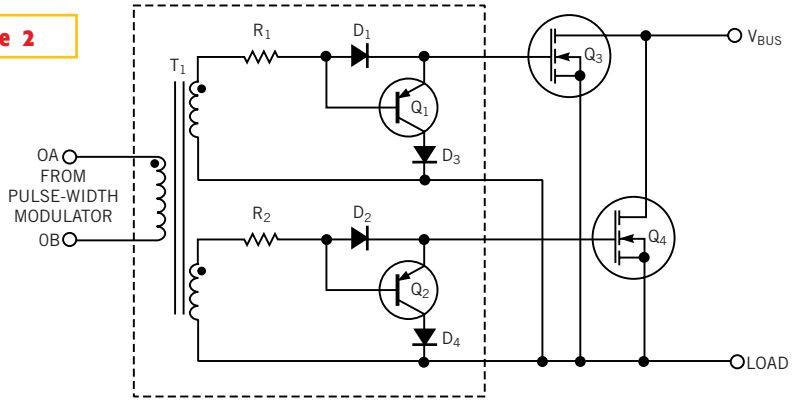
Figure 1



You can use this low-cost circuit to drive high-side FETs or IGBTs.

In systems that require higher power, you can use a dual-FET circuit (**Figure 2**). R_1 and R_2 and the output impedance of the PWM or FET driver determine the turn-on time. R_1 and R_2 and the h_{fe} values for Q_1 and Q_2 determine the turn-off time. The Schottky diodes, D_3 and D_4, prevent current flow through the collector-base junctions of Q_1 and Q_2 when the drive signal is negative. You can obtain switching speeds higher than 50 nsec at turn-on and 100 nsec at turn-off with this circuit, depending on the output impedance of the transformer drive signal.

Figure 2



Two FETs in parallel provide more power than the circuit in Figure 1.

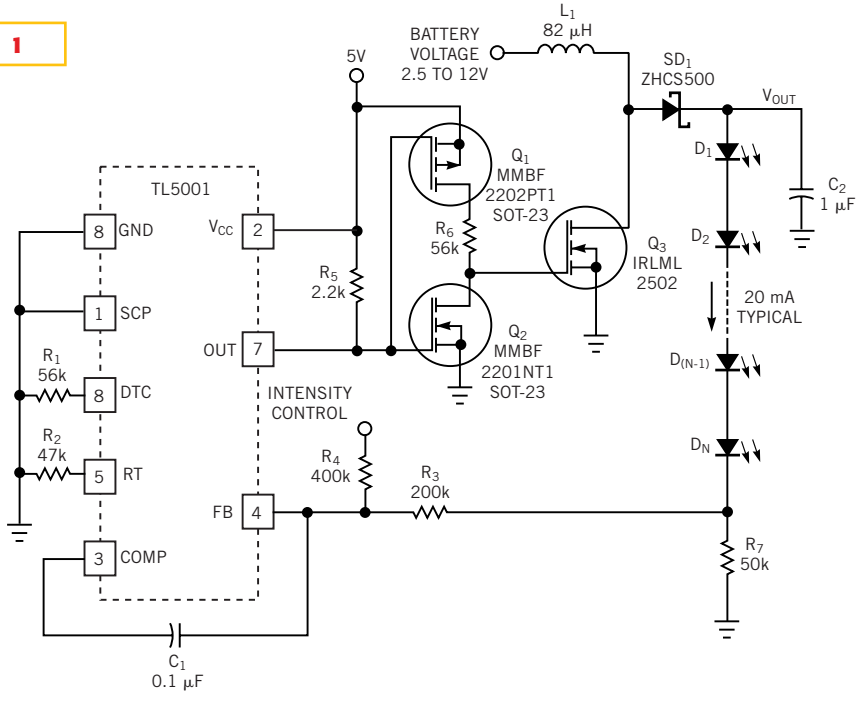
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Economical circuit drives white LEDs

Eddy Wells, Texas Instruments

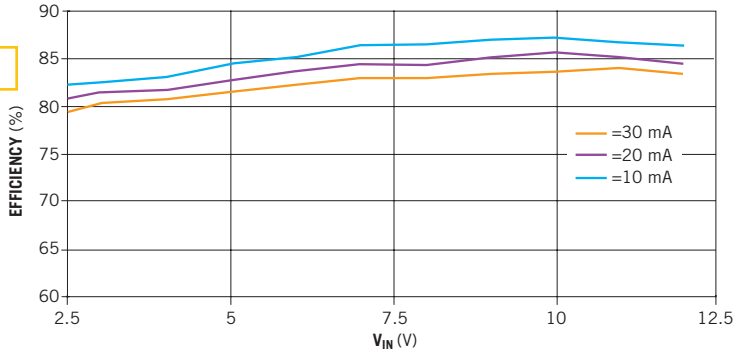
Newly available white LEDs are replacing CCFLs (cold-cathode fluorescent lamps) in handheld applications using a backlit LCD. These applications include PDAs (personal digital assistants), digital cameras, and cellular telephones, to name a few. Advantages of white LEDs over CCFLs include longer life, higher efficiency, and significantly lower operating voltages. Regulating the current in the LED (typically 10 to 30 mA) controls the brightness; the forward voltage in each LED is approximately 3V. The circuit in **Figure 1** provides a means of efficiently controlling LED current in a series-connected string. The TL5001 PWM-controller IC is an older, industry-standard, inexpensive driver. The boost topology of the circuit allows operation from a single or dual lithium-ion cell. The ratings of Q_3, SD_1, and the maximum allowed duty cycle of the IC (programmed with pin DTC) determine the maximum output voltage of the circuit. V_{CC} comes from a separate 5V supply.

Figure 1



White LEDs provide ideal backlighting for small LCDs.

Figure 2



The circuit in Figure 1 provides more than 80% conversion efficiency.

Trade-offs in the selection of inductor L_1 include size, dc resistance, and inductance value. An 82-µH inductor (with 200-mΩ dc resistance) results in continuous inductor current at higher LED currents, but the current becomes discontinuous at lower levels. The RT pin programs the oscillation frequency at approximately 200 kHz. Because the TL5001 has a relatively weak (20-mA) open-collector output driver and is intended to drive a buck-topology circuit, the circuit uses a low-cost inverter stage comprising Q_1 and Q_2 to efficiently drive Q_3. R_6 provides controlled turn-on and fast turn-off for Q_3. The reverse-breakdown voltage of SD_1 must be greater than the C_2-filtered V_{OUT}.

R_7 senses current in the white-LED string; the error amplifier at the FB pin of the IC controls the feedback signal at this pin to 1V. You can control the LEDs' intensity by summing in a control voltage via R_4. **Figure 2** shows the efficiency of a four-LED string. You could obtain approximately 2% higher efficiency by adding a gain-of-5 op-amp stage between R_7 and R_3, resulting in a lower voltage drop across R_7. Of course, this slight efficiency improvement adds to the system cost. For higher power applications, such as notebook computers, you can attach additional LED strings to V_{OUT}. To maintain uniform intensity in each string, you should add a dummy resistor of the same value as R_7 to each string.

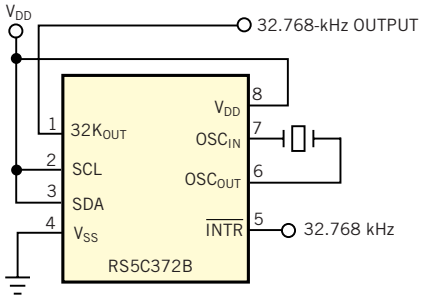
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Real-time-clock chip makes low-power oscillator

Yongping Xia, Teldata Inc, Los Angeles, CA

Many systems use watch-crystal-based, 32.768-kHz oscillators. In battery-powered designs, the 32-kHz oscillator may consume a fairly high percentage of the total power budget. Reduced power consumption equates to longer battery life, smaller batteries, and smaller products. Ricoh (www.ricoh.com) manufactures more than 10 types of real-time-clock chips, including the RS5C372B (**Figure 1**). This device is an eight-pin IC with a built-in oscillator, programmable periodic interrupts, and an I^2C interface to a µC. The only function the device in **Figure 1** uses is the 32-kHz oscillator. Using only the IC and the crystal, the circuit consumes low current over its 1.5 to 6V power-supply range, as the table in **Figure 1** shows. The CMOS-based output delivers a waveform with an amplitude of 0V to V_{DD}.

Figure 1



V_{DD} (V)	CURRENT (µA)
1.5	0.9
2	1.1
2.5	1.3
3	1.6
3.5	2.1
4	2.6
4.5	3.2
5	3.9
5.5	4.7
6	5.6

This ultra-low-power oscillator uses only the oscillator portion of a real-time-clock chip.

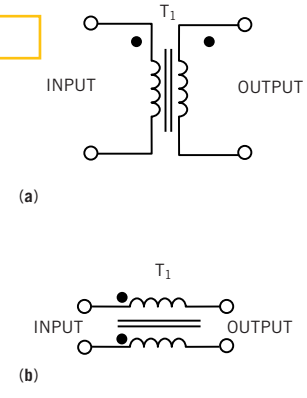
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Everything you wanted to know about wideband, low-frequency transformers

Richard Kurzrok, Queens Village, NY

Wideband, low-frequency transformers are useful components in various passive circuits, such as the return-loss bridge (**Reference 1**, **Figure 2**). **Figure 1a** shows a conventional transformer. If you connect the transformer windings differently, you can configure a transmission-line transformer (**Reference 2**, **Figure 1b**). These transformers use a magnetic core of modest size, and unit cost is reasonable. The basic transformer uses a Fair Rite (www.fair-rite.com) toroid (type 597700601), which has a nominal outer diameter of 0.825 in., a nominal inner diameter of 0.525 in., and a nominal thickness of 0.25 in. The toroid uses number 77 material and has an inductance factor (AL) of 1175. To obtain useful performance at audio frequencies, the transformer uses a 129-turn bifilar winding of number 26 magnet wire. To avoid the use of expensive commercial bifilar wire, you can twist together monofilar red and green windings using a hand drill before winding it on the toroid.

Figure 1



You can use a wideband transformer in conventional mode (a) or in a transmission-line configuration (b).

Table 1 gives the measured performance with 50Ω source and load impedances of the conventional transformer in **Figure 1a**. **Figure 1b** shows the schematic diagram of a one-to-one transmission-line transformer. **Table 2** gives the measured performance with 50Ω source and load impedances. This transformer provides bandwidth enhancement with useful behavior down to dc. You can use the conventional transformer in **Figure 1a** in a passive return-loss bridge (**Figure 2**) or for stand-alone dc isolation. **Table 3** gives the measured performance of the 50Ω

return-loss bridge. At 1 MHz, the return-loss bridge exhibits a forward insertion loss of 12 dB and an open-circuit-to-short-circuit ratio of 0.5 dB. You can use the conventional transformer to isolate a grounded signal from a balanced test piece. We built the circuits for the wideband transformers and return-loss bridge using single-clad vector board and enclosed them in die-cast aluminum boxes with BNC connectors.

References

1. Wetherhold, E, "Design and Construction of a 9-kHz Highpass Filter and Assembly of a Return Loss Bridge for Filter and PLISN," *Interference Technology Engineers' Manual*, pg 220, 1993.
2. Sevick, J "Transmission Line Transformers," American Radio Relay League, 1990.

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

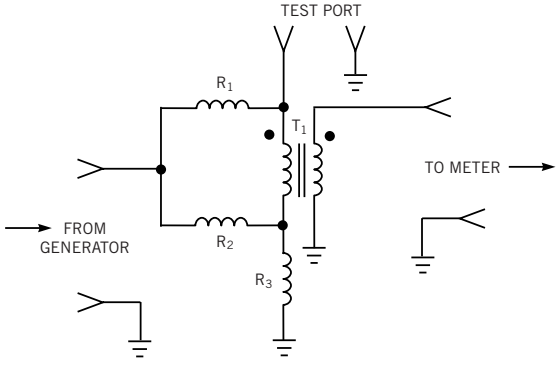
TABLE 1–PERFORMANCE OF A CONVENTIONAL TRANSFORMER

Frequency (MHz)	Insertion loss (dB)	Frequency (kHz)	Insertion loss (dB)
10	7.2	50	<0.1
8	3.8	20	<0.1
7	2.6	10	<0.1
6	1.9	5	<0.1
4	1	2	<0.1
2	0.5	1	0.15
1	0.3	0.5	0.5
0.5	0.15	0.2	1.7
0.2	<0.1	0.15	2.6
0.1	<0.1	0.1	4.1

TABLE 2–PERFORMANCE OF A TRANSMISSION-LINE TRANSFORMER

Frequency (MHz)	Insertion loss (dB)	Frequency (kHz)	Insertion loss (dB)
10	1.7	100	0.15
7	1.2	50	0.15
4	0.9	20	0.1
2	0.6	10	0.1
1.5	0.5	5	0.1
1	0.4	1	0.1
0.5	0.3	0.1	<0.1
0.2	0.2	0.01	<0.1

Figure 2



A return-loss bridge is one application for wideband, low-frequency transformers.

TABLE 3–DIRECTIVITY OF THE RETURN-LOSS BRIDGE

Frequency (kHz)	Directivity (dB)
1000	22
500	27
200	33.5
50	38
20	41.5
10	44.5
5	>46
2	>46
1	>46
0.5	>46

µC makes effective frequency counter

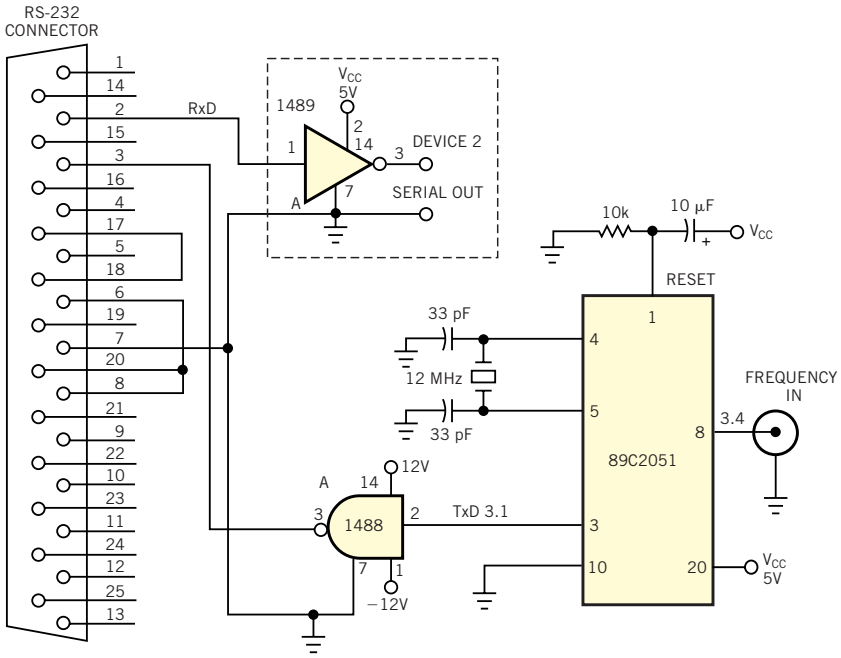
Fazal Pathan, Physical Research Laboratory, Ahmedabad, India

Figure 1 shows an efficient and cost-effective frequency counter using an Atmel 89C2051 µC (**Reference 1**). The design can use any µC of the 8051 family. The circuit counts frequency and sends the count to a PC via the serial port. The signal connects to pin 3.4 of the µC. The TTL-compatible output of the µC drives the 1488, which converts the output to RS-232 voltage levels. The output of the 1488 connects to the RxD pin on the serial port of the PC. In this design, only unidirectional communication exists between the µC and the PC. But you can use the TxD line to control another device. The assembly routine in **Listing 1** shows initialization of the timer, counter, and interrupts. The main program sets up T0 to count external pulses and T1 to count time in the autoreload mode 2 (**Reference 2**). The main program monitors the main-program flag, frqflg, until the flag is set. Then, the main program sends the counted frequency to the PC's port using mode 1. First, it sends the special character "L" to recognize the following 3 bytes as valid data. Then it sends the value of register R2, which is the most significant byte of the counter value. This value increments every time the counter

overflows. Then, the main program sends the values of registers TH0 and TL0 to the PC's port.

After sending the data, the program again jumps to the main routine, in which it clears the timer and counter and reinitializes them, and then starts counting again. The µC counts the value in hexadecimal format; the PC can then convert it to decimal format. In this example, we used a 12-MHz crystal and a baud rate of 2400 bps. However, you can use higher clock frequencies and baud rates, because the µC can operate to 24 MHz. We use the system in **Figure 1** for photon counting in astronomical instrumentation. We give thanks to the Department of Space, Government of India, for material support and BG Anandarao, PhD, for encouragement. You can download **Listing 1** 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 #2616.

Figure 1



An 8051-family µC makes an efficient and cost-effective frequency counter.

References

1. Data Sheet for 89C2051, www.atmel.com.

2. Ayala, Kenneth J, *8051 Microcontroller: Architecture, Programming and Applications* (ISBN 0-314-77278-2), West Publishing Co, St Paul, MN.

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

LISTING 1—FREQUENCY-COUNTER ROUTINE FOR 89C2051

```
; Program counts the pulses and transmits through p3.1 pin only without
handshaking signal.

        .Equ Frqflg,0fh                 ;use a bit flag to signal main program
        .Equ Baudnum,0f3h               ;number loaded in TH1 for 2403.8 baud
Freq:
        .Org 0000h
         Mov SP, #30h                   ;set stack above register/bit area
         Sjmp over                      ;jump over T1 interrupt address

        .Org 000bh
         Inc r2
         Reti

        .Org 001bh                      ;T1 overflow flag interrupt to here
         Setb psw.3                     ;swich to register bank1
         Inc r0                         ;count R0 up until overflow at 00h
         Cjne r0, #00h,Checktime        ;check to see if time is up
         Inc r1                         ;or inc R1 when R0 rolls over
Checktime:
        Cjne r1, #27h, Goback           ;check R1 for terminal count
        Cjne r0, #10h, Goback            ;check R0 for terminal count
        Clr tr0                         ;stop T0
        Clr tr1                         ;stop T1 time before T0 stopped =
        Setb Frqflg                     ;signal main program that T0 =freq.
Goback:
        Clr psw.3                       ;return to bank 0 registers
        Reti                            ;return to main program

Over:
        Mov tcon, #00h                  ;all timers stopped - flags reset
        Set psw.3                       ;select register bank 1 and reset R0, R1
        Mov r0, #00h
        Mov r1, #00h
        Clr psw.3                       ;return to bank 0
        Mov tmod, #25h                  ;T1 a mode 2 timer, T0 mode 1 counter
        Mov tl1, #9ch                   ;start TL1 at 9ch
        Mov th1, #9ch                   ;TH1 = 156d, overflows in 100 clocks
        Mov tl0, #00h                   ;zero T0
        Mov th0, #00h
        Clr Frqflg                      ;reset the freq. measured flag
        Mov tcon, #50h                  ;start timer T1 and counter T0
        Mov ie, #88h                    ;enable global and T1 overflow interrupts
Simulate:
        Jbc Frqflg, Getfrq              ;have main program test "frqflg"
        Sjmp Simulate                   ;loop here until freq.is measured
Getfrq:
        Anl pcon, #7fh                  ;set SMOD bit to 0 for Baud * 32 rate
        Anl tmod, #30h                  ;alter timer T1 configuration only
        Orl tmod, #20h                  ;set timer T1 as an 8 bit autoload
        Mov th1, #Baudnum
        Setb tr1                        ;run T1
        Mov scon, #40h
        Mov sbuf, #'L'
Wait:
        Jbc ti, Next
        Sjmp Wait
Next:
        Mov sbuf, r2                    ;set UART to mode1
Wait1:
        Jbc ti, Next1
        Sjmp Wait1
Next1:
        Mov sbuf, th0                   ;transmit content of the th0
Wait2:
        Jbc ti, Next2                   ;wait for T1set before next transmission
        Sjmp Wait2                      ;else poll flag again
Next2:
        Mov sbuf, tl0                   ;now transmit tl0
                                        ;and wait until the ti flag is set
Wait3:
        Jbc ti, Over
        Sjmp Wait3

       . End
```

Software provides interrupt system for 8051

Deng Yong, Shanghai Jiaotong University, China

By using a "pseudo-RETI" instruction, the program in **Listing 1** provides a three-priority-level interrupt system for the 8051 µC. Among the three interrupt sources in the routine, External Request 0 (INTO) has the highest priority, and Internal Time/Counter 0 (ITO) has the lowest priority. In the ITO interrupt-service routine before the "pseudo-RETI" instruction, the address of the first instruction after the "pseudo-RETI" instruction goes back into the stack. The internal nonaddressable flip-flop of the ITO clears to acknowledge a higher interrupt after execution of the "pseudo-RETI" instruction, while the ITO interrupt-service routine executes continuously until the arrival of the RETI instruction. You can download **Listing 1** 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 #2589.

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

LISTING 1—THREE-PRIORITY-LEVEL INTERRUPT SYSTEM FOR THE 8051

```
        ORG   0000H
        LJMP  START
        ORG   0003H
        LJMP  INT0
        ORG   000BH
        LJMP  IT0
        ORG   0013H
        LJMP  INT1
START:  MOV  SP,#60H
        MOV  IP,#01H            ;INT0 has high priority
        MOV  TMOD,#01H
        MOV  TH0,#00H
        MOV  TL0,#00H
        SETB EA                 ;enable INT0,INT1,IT0
        SETB EX0
        SETB EX1
        SETB ET0
        SETB TR0
          …

INT0:     …  ……..;INT0 interrupt service program
          …
        RETI
INT1:     …                     ;INT1 interrupt service program
          …
        RETI
IT0:    CLR  TR0                ;IT0 interrupt service program
        PUSH DPL
        PUSH DPH
        MOV  DPTR,#GO_ON
        PUSH DPL
        PUSH DPH
        RETI                    ;"pseudo-RETI"
GO_ON:  NOP
          …
          …
        MOV  TH0,#00H
        MOV  TL0,#00H
        POP  DPH
        POP  DPL
        SETB TR0
        RETI
```

Look-up table facilitates bit flipping

Brad Bierschenk, High End Systems, Austin, TX

In certain instances in embedded software, it becomes necessary to flip the order of bits in a byte, so that B7:B0 becomes B0:B7. For example, this feature could be useful with a synchronous serial port that does not allow programmed selection of bit order (MSB first or LSB first) for its shift register. If a device to which the processor sends data expects one bit order but the serial port can provide only the other bit order, you must use a software method to translate the data. One solution to this problem is to provide a look-up table in ROM, in which the value of each byte in the table is offset into the table, but with bit order reversed. In other words, the first byte is offset 0 (00000000b), the second byte is offset 1 (10000000b), the third byte is offset 2 (01000000b), and so on.

LISTING 1—CODE SEGMENT FOR 80XC51 ARCHITECTURE

```
; Load value into accumulator (hex AA)
mov         A, #10101010b
; Load lookup table address into index
mov     DPTR,#InvertTable
; Load "flipped" value into accumulator
movc    A,@A+DPTR
; Accumulator should now hold hex 55
```

The program merely needs to load the value to be translated into a register that can serve as an offset, index the look-up table, and load the corresponding value from the index+offset location. This design uses the Philips 80xC51 architecture as an example. **Listing 1** shows a code segment for the µC. You can use the µC's 16-bit DPTR (data pointer) plus an 8-bit offset in its accumulator to load the accumulator with a byte value. This solution to bit flipping is dynamically more efficient than rotating a byte location through carry bits or other possible solutions. However, it's not the most statically efficient solution, because it requires 256 bytes of ROM for the look-up table. You can download **Listing 1** and the look-up table 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 #2621.

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