Electronic Design New Design Ideas

January 4, 2001 * pag. 6
Preheat starter for electronic ballast
Low-cost circuit programs EEPROMs
Circuit yields ultralow-noise VGA
Sequential channel selector simplifies software
Microcontroller provides timer function

January 18, 2001
Open-loop power supply delivers as much as 1W
Four-way remote control uses series transmission
Analyze LED characteristics with PSpice
Programmable-gain amplifier is low-cost
PC hardware monitor reports the weather

February 1, 2001 pag. 20
Method sets voltage in multiple-output converters
Circuit forms constant-current SCR
555 makes handy voltage-to-time converter
Program predicts VSWR-mismatch RF uncertainties
PC monitors two-way RS-232 transmission
Passive filters fill the bill at audio frequencies
Watchdog timer assumes varied roles
One microcontroller serves multiple external interrupts

February 15, 2001
Keep the heat down in power op amps
Buck converter works efficiently from phone line
Sine reference is synchronous with ac line
Cascade bandpass filters for higher Q
C program calculates checksums
One-wire bus powers water-level sensor
Ideal transformers aid in balanced-line analysis

March 1, 2001 * pag. 35
Circuit senses high-side current
Adjustable filter provides lowpass response
Monitor high-side current without an external supply
Noncontact device tests power supplies
Single chip detects optical interruptions
Programmable source powers dc micromotors

design ideas

Edited by Bill Travis and Anne Watson Swager

Preheat starter for electronic ballast

Arthur E Edang, Don Bosco Technical College, Mandaluyong City, Phillipines

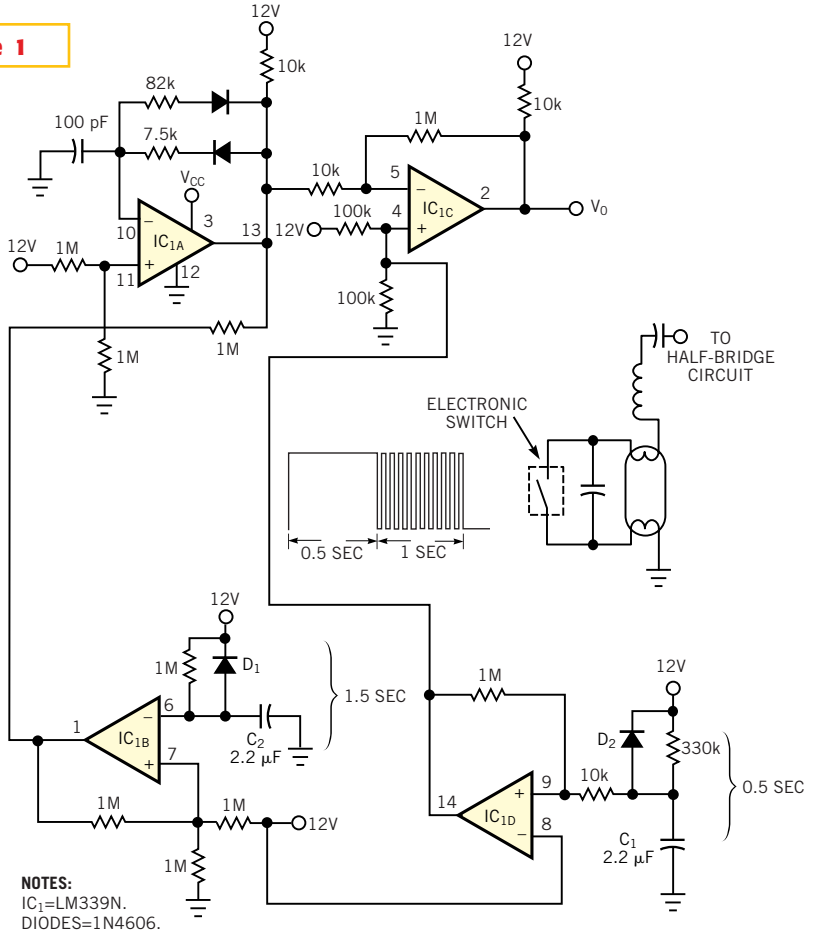
Electronic ballasts for fluorescent lamps use various techniques to turn on the bulbs. The design usually involves a compromise between turn-on voltage and lamp life because the two are inversely related. One way to reach a reasonable compromise is to initially allow a momentary inrush current to warm the filaments, followed by a series of interrupted short circuits across the lamp that generate the required high voltage to trigger the fluorescent. With a preheated filament, the necessary strike potential reduces to half.

The trigger circuit in **Figure 1** controls the electronic switch across the bulb. At start-up, IC_{1D}'s output is low as C_1 and C_2 charge toward V_{CC}. IC_{1D}'s low output pulls IC_{1C}'s inverting input low, which causes V_O to clamp high. A high level at V_O closes the switch and forces current through the filaments. After approximately 0.5 sec, IC_{1D}'s output changes state and allows IC_{1C} to accept the high-frequency signal at its noninverting input. IC_{1A} is a square-wave oscillator, which causes V_O to be a high-frequency-pulse series that lasts for approximately 1 sec. At the end, C_2 reaches a high enough voltage to force IC_{1B} to pull down IC_{1A}'s noninverting pin to ground. With a grounded IC_{1A} output, V_O clamps low.

The high-frequency switching strikes the preheated lamp. In case the bulb fails to start, the circuit turns off and then on again. Residual charges on the capacitors discharge through D_1 and D_2 to ensure precise timing.

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

Figure 1



This trigger circuit generates a high-frequency-pulse series to strike a preheated lamp.

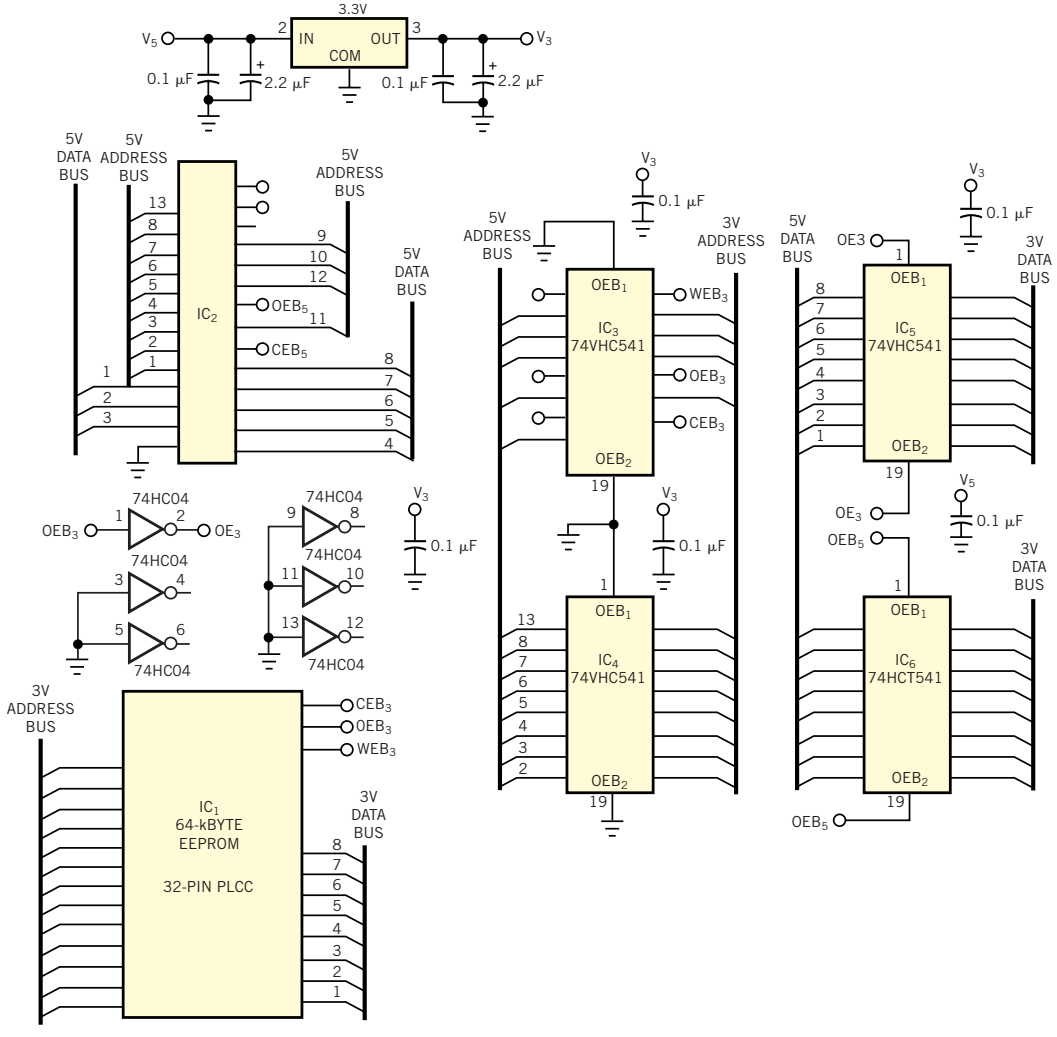
Low-cost circuit programs EEPROMs

Jarrod Eliason, Ramtron, Colorado Springs, CO

When you migrate to 3.3V system supplies, you must usually replace your old, reliable EEPROM programmer with a new, overly flexible and expensive universal programmer. We could not find a 3.3V programmer for less than $1000. For less than $100, the circuit in **Figure 1** extends the functional life of any 5V EEPROM programmer. You can apply the circuit to any bidirectional 5 to 3.3V level-translating application. The key to the circuit lies in choosing the correct logic families. The 74VHC and 74LVC families handle the 5-to-3.3V conversion better than previous logic families, such as the 74HC series. The 74HC family accommodates 3.3V operation, but the input-protection diodes clamp the input voltage within a diode drop of V_{DD} (**Figure 2a**). So, applying 5V to the input of a 74HC part powered from 3.3V results in much undesired current. An external resistor could limit this current, but this fix would impact bus speed. The 74VHCT and 74LVC families do not use a reverse-biased diode to V_{DD} (**Figure 2b**), so the input voltage can safely rise to 5.5V, regardless of the supply level.

The 74HCT family handles the 3.3-to-5V conversion. This 5V CMOS logic family uses input switching levels skewed to accommodate TTL-level inputs. The low and high levels are 0.8 and 2.4V, respectively, in comparison with the typical CMOS levels of 1.5 and 3.5V. Because the inputs receive high levels of 3.3V at most, CMOS-optimized 74HC logic would not guarantee recognition of logic 1 inputs. On the other hand, to a 74HCT powered from 5V, a 3.3V input level represents a solid logic 1. We selected the tristatable

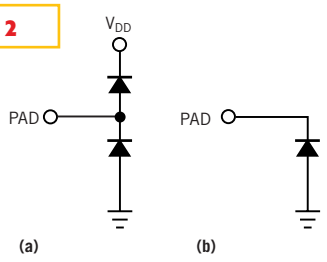
Figure 1



For less than $100, this circuit adapts a 5V EEPROM programmer for 3.3V operation.

buffer function for the EEPROM-programmer level translation. The circuit in **Figure 1** programs a 3.3V, 64-kbit EEPROM, using a 5V programmer. For the address and control pins, the output-enable pin of the 74VHC chips is constantly active. For the bidirectional data bus, the OEB5 and OE3 signals control the in/out selection. When OEB5 is low and OE3 is high, a read operation takes place, and the EEPROM has control of the data bus. When OEB5 is high and OE3 is low, a write operation takes place, and the programmer drives the data bus.

Figure 2



The 3.3V-powered 74HC-logic inputs are not amenable to 5V inputs (a); 74HVC and 74LVC inputs have no such problem (b).

A 28-pin DIP socket, IC_2, connects to the 5V EEPROM programmer. The circuit uses an additional adapter to interface to the 32-pin PLCC target device, IC_1. The 74VHC and 74LVC logic parts are not readily available in DIP form, so you can use SOIC-to-DIP adapters for breadboarding. If the 74HCT541 is not available, you can use the alternate-pinout 241 or 244.

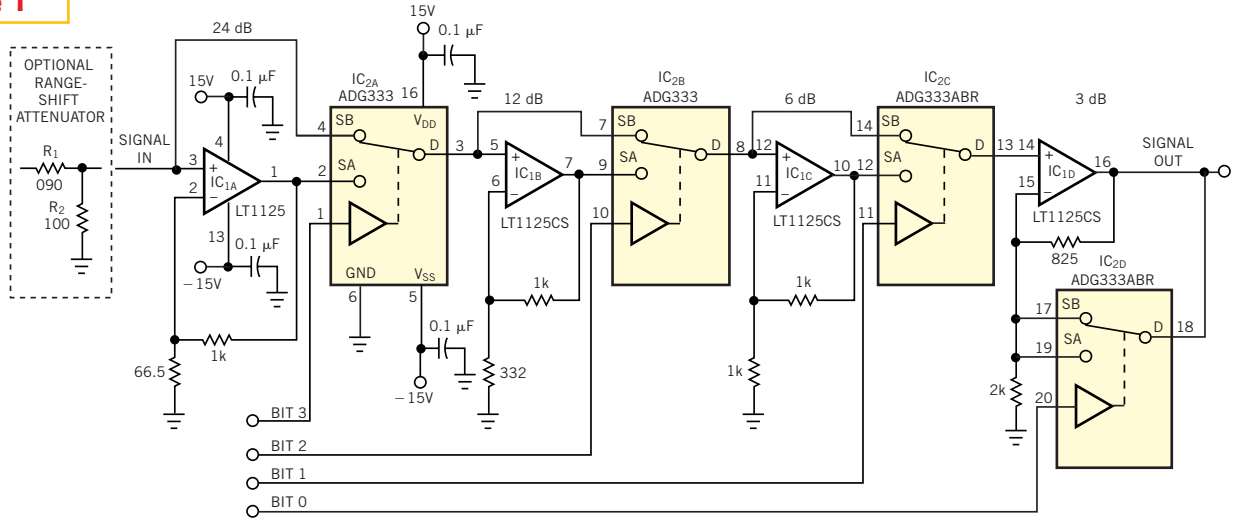
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Circuit yields ultralow-noise VGA

Dale Ouimette, California Institute of Technology, Pasadena, CA

A number of single-chip VGAs variable-gain amplifiers are available today. Unfortunately, they all have drawbacks, such as high noise, 55V limit, low input impedance, or nonlinear gain/frequency characteristics. The circuit in **Figure 1** is a 16-step, ultralow-noise VGA that solves many of these problems. IC_1 is a low-noise quad op amp, and IC_2 is a quad SPDT CMOS switch. The stages switch in successive multiplication (gain) factors using a TTL binary code. The values shown provide 0- to 45-dB gain in 3-dB steps. For best low-noise performance, the higher gain stages precede the lower gain stages. The circuit exhibits approximately 3 nV/$\sqrt{\text{Hz}}$, referred to the input, for most gain settings. The highest noise is 4.5 nV/$\sqrt{\text{Hz}}$ at a gain of 9 dB. Distributing the total gain across multiple stages increases the overall bandwidth. The output stage has a different configuration to yield a low-output-impedance output driver

Figure 1



This VGA offers ultralow noise, a wide dynamic range, and high bandwidth.

at all gain settings.

If you need to remotely control the gain, you must concern yourself with ground loops that can compromise the low-noise characteristics of the circuit. One solution is to place optoisolators in the four digital-control lines, so that no ground connection exists between the two ends of the cable except through the power supply. The method you use is an analog differential-control voltage using an ADC to generate the 4 bits. **Figure 2** shows a circuit that performs this function well. IC_1 is a differential receiver, and IC_2 is an 8-bit ADC. In some applications, you could get away with using only the ADC, because it already has a differential input. However, you must take care not to exceed the narrow common-mode range of the ADC's input. A more robust solution is to place a differential receiver in front of the ADC, as shown. R_1 and C_1 form a lowpass filter for the control voltage to the ADC. The 4 high-order bits from the ADC control the CMOS switches. As shown, the ADC operates in a self-clocking mode and needs no other controls.

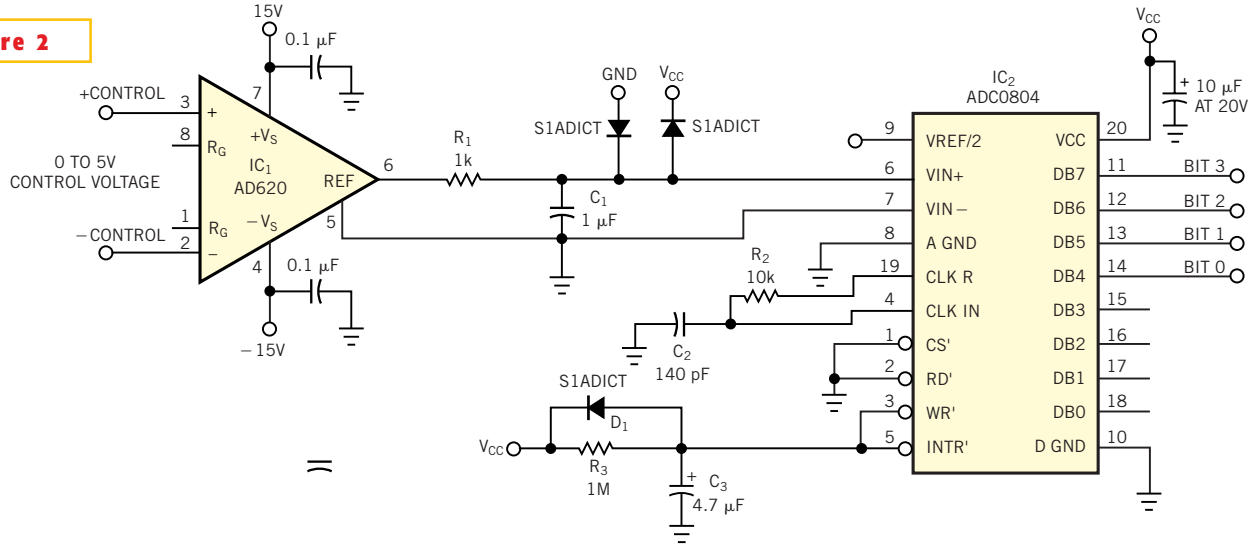
TABLE 1–PERFORMANCE VERSUS GAIN

Step	Gain (dB)	V/V	Noise (referred to input) (nV/√Hz)	3-dB bandwidth (MHz)
0	0	1	3.1	10.5
1	3	1.4	3.8	7.7
2	6	2	4.4	5.1
3	9	2.8	4.5	4.6
4	12	4	3.6	2.7
5	15	5.6	3.6	2.7
6	18	7.9	3.7	2.6
7	21	11.2	3.7	2.6
8	24	15.8	3	0.88
9	27	22.4	3	0.89
10	30	31.6	3	0.94
11	33	44.7	3	0.96
12	36	63.1	3	0.97
13	39	89.1	3	0.97
14	42	125.9	3	1.04
15	45	177.8	3	1.02

R_2 and C_2 control the sampling frequency, approximately 640 kHz for the values shown. D_1, R_3, and C_3 provide power-up initialization for the ADC's clocking function. The control-voltage steps are 310 mV apart, providing ample noise immunity. **Table 1** shows the performance of the overall circuit with analog control. You can use R_1 and R_2 in **Figure 1** to shift down the overall gain range with little sacrifice of noise characteristics. You can obviously alter the individual gain stages to yield other ranges and step sizes, such as 0 to 30 dB in 2-dB steps. At the expense of circuit simplicity, you could replace the quad op amp with four ultra-low-noise op amps, such as the LT1128 or AD797. This replacement lowers the noise to approximately 1.4 nV/√Hz. You could also increase the number of stages, thereby providing a wider dynamic range, finer gain steps, or both. The benefits of this circuit over commercially available single-chip VGAs include ultra-low noise, high bandwidth, ±13V range, high input impedance, ground-loop immunity, and user-defined dynamic range and step size.

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

Figure 2



An ADC controls the gain-setting codes for the circuit in Figure 1.

Sequential channel selector simplifies software

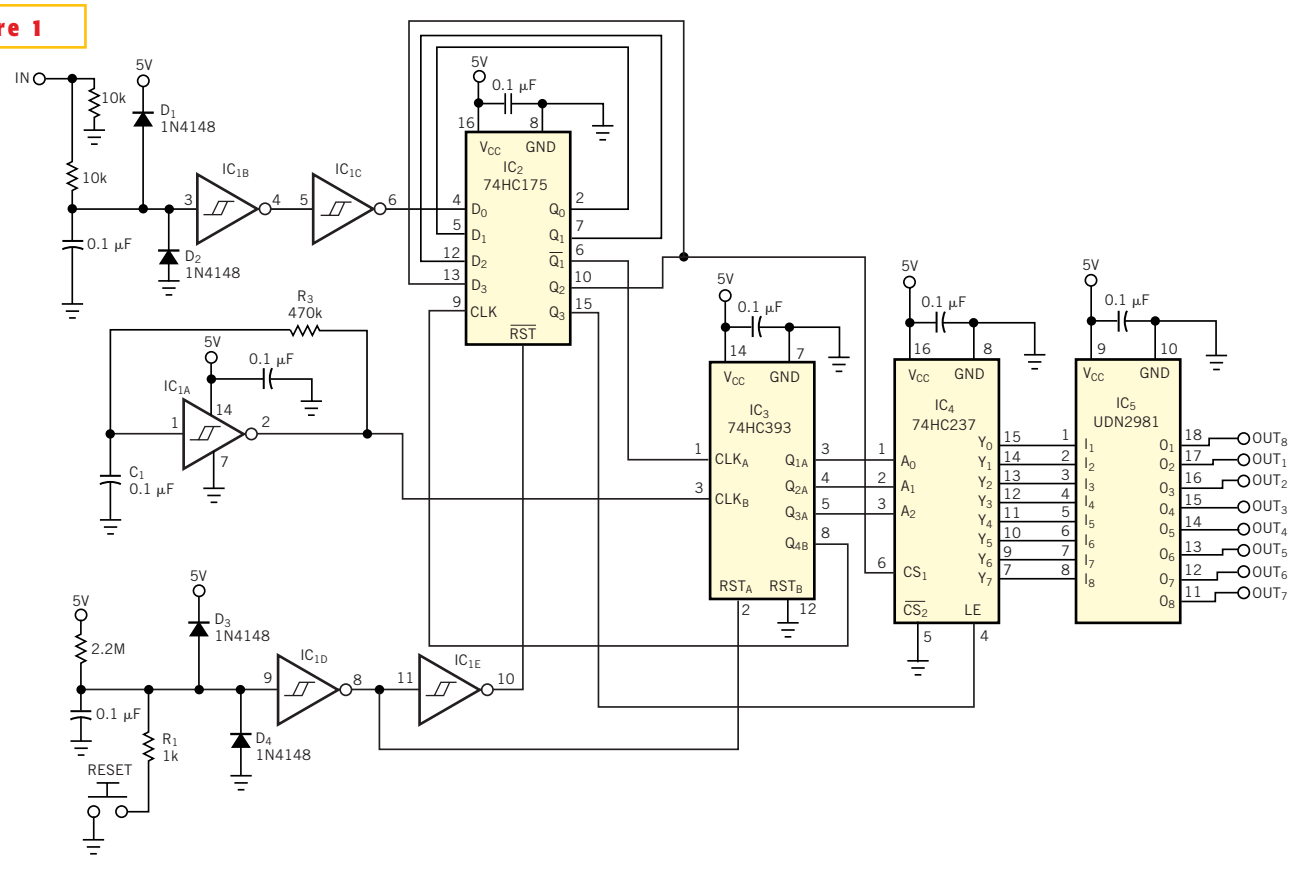
Alex Knight, Cummins Engine Co, Columbus, IN

An efficient but powerful circuit is useful for a variety of applications with limited I/O and for which you want to use one input to sequentially select a different output channel (**Figure 1**). When the software changes the state of only one input, the circuit sequentially selects one output channel at a time for test purposes. Because the test-application environment is potentially harsh, the circuit must have relatively high noise immunity and transient protection at the inputs. You must also be able to reset the circuit to resynchronize the hardware with a test program after any interruption in testing.

Although the resulting circuit may seem simple and standard, it is distinctly robust. The delayed reset signals at IC_2's Pin 1 and IC_3's Pin 2 return the counter and flip-flop ICs to their initial state so that OUT_1 is the first channel active at the first count. The power-on and switch-activated reset circuit includes R_1, D_3, and D_4 to protect against ESD that could arc over the switch contacts when someone first touches the switch. The IN signal input circuit has similar transient protection with R_2, D_1, and D_2. A simple RC oscillator generates the clock signal at IC_2's Pin 9, and the second four-stage binary-ripple counter, IC_3, divides this clock by 16. The oscillator frequency is approximately 21 Hz, but you can change R_3 and C_1 to produce the desired frequency, which is approximately $1/R_3C_1$. You can also use a potentiometer in place of R_3 to make the frequency adjustable. Keep in mind that the flip-flop clock-cycle period should be much less than the expected active and inactive periods of the IN signal but long enough to produce adequate debouncing of the input signal to maintain good noise immunity. The circuit serves a low-speed application, so the clock at IC_2's Pin 9 is 1.3 Hz.

The circuit filters and buffers the IN signal before sending it to the flip-flop

Figure 1



A robust circuit uses one input to sequentially select one output channel at a time.

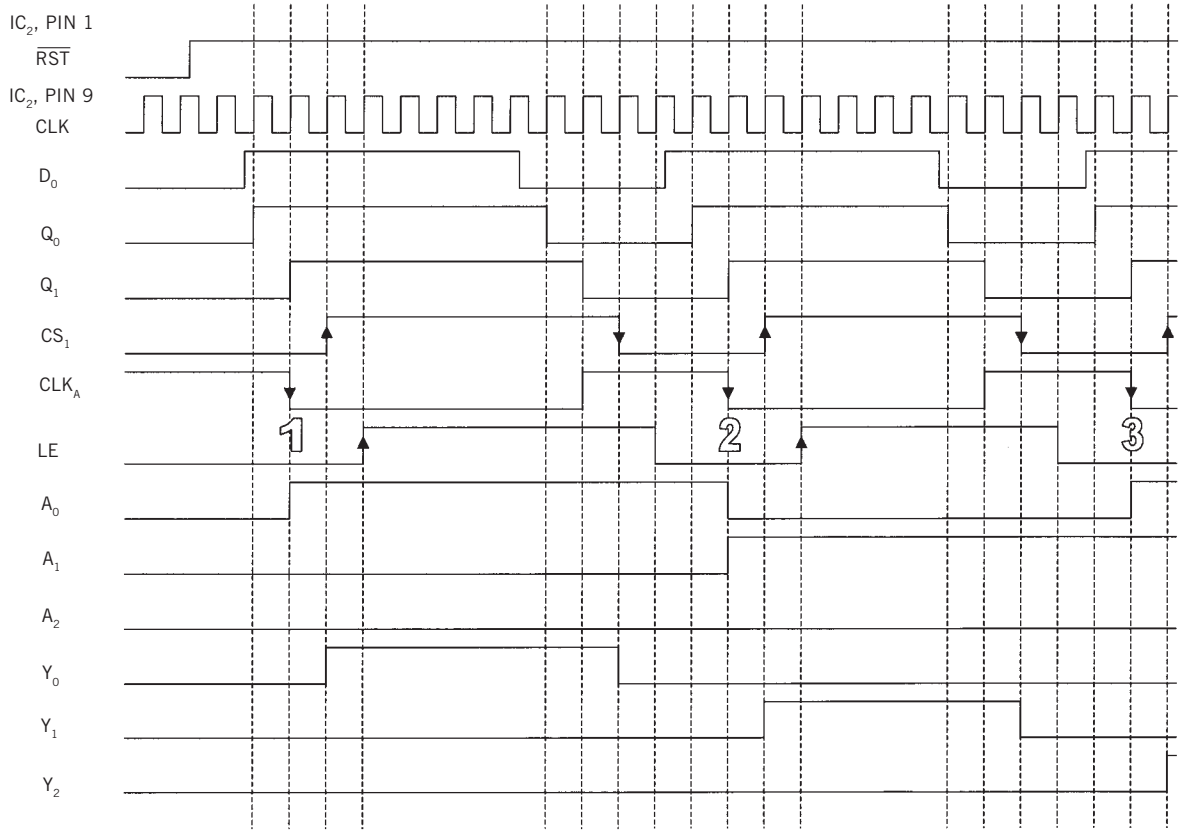
input at IC_2's pin 4. The Schmitt inverter, IC_1 with its built-in hysteresis and the cascaded flip-flop circuit provide high immunity to noise, and the cascaded flip-flop ignores any glitches on the input signal that occur asynchronously to the flip-flop clock signal's positive-going transitions. The circuit uses the $\overline{Q_1}$ output signal as the CLK_A clock input to the first four-stage binary ripple counter, IC_3. Negative-going transitions increment the counter as the timing diagram indicates at counts 1, 2, and 3 (**Figure 2**). The circuit uses the Q_2 output to select the active-high CS_1 chip-select input of IC_4's one-of-eight decoder, which allows plenty of time for the ripple counter outputs to stabilize, even at high flip-flop clock speeds. These outputs do not simultaneously change states. With CS_1 high, the positive-going Q_3 output signal at the LE input of the decoder (IC_4, pin 4) latches the output channel that the state of the A_0-to-A_2 address inputs select. Latching the output channel ensures you that any subsequent noise-induced counter-output state changes will not affect the output-channel states. While CS_1 is low, the Y_0-to-Y_7 outputs from IC_4 are also low. This design maintains a similar off-time for all of the output channels, as reflected in the input signal, although the circuit delays any change of state for each of the outputs by approximately two cycles of the flip-flop clock period.

IC_5 can drive loads that sink as much as 350 mA at room temperature, such as relays, solenoids, dc motors, and lamps. This eight-channel source-driver IC is unnecessary if CMOS outputs suffice as the channel-select signals. The IC_5 source voltage can climb to 35V if you add a separate supply. IC_5 has internal diodes on all of the outputs to clamp inductive spikes.

The circuit includes a switch for generating a reset signal, which you can use in addition to or instead of an external reset signal. The input can also be an external analog signal or non-TTL, as long as you properly compensate for any dc offset necessary to work at the switching thresholds of the Schmitt inverter. You can cascade additional ripple-counter stages and add decoders and output drivers to select from more output channels.

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

Figure 2



Negative-going transitions increment the counter at counts 1, 2, and 3.

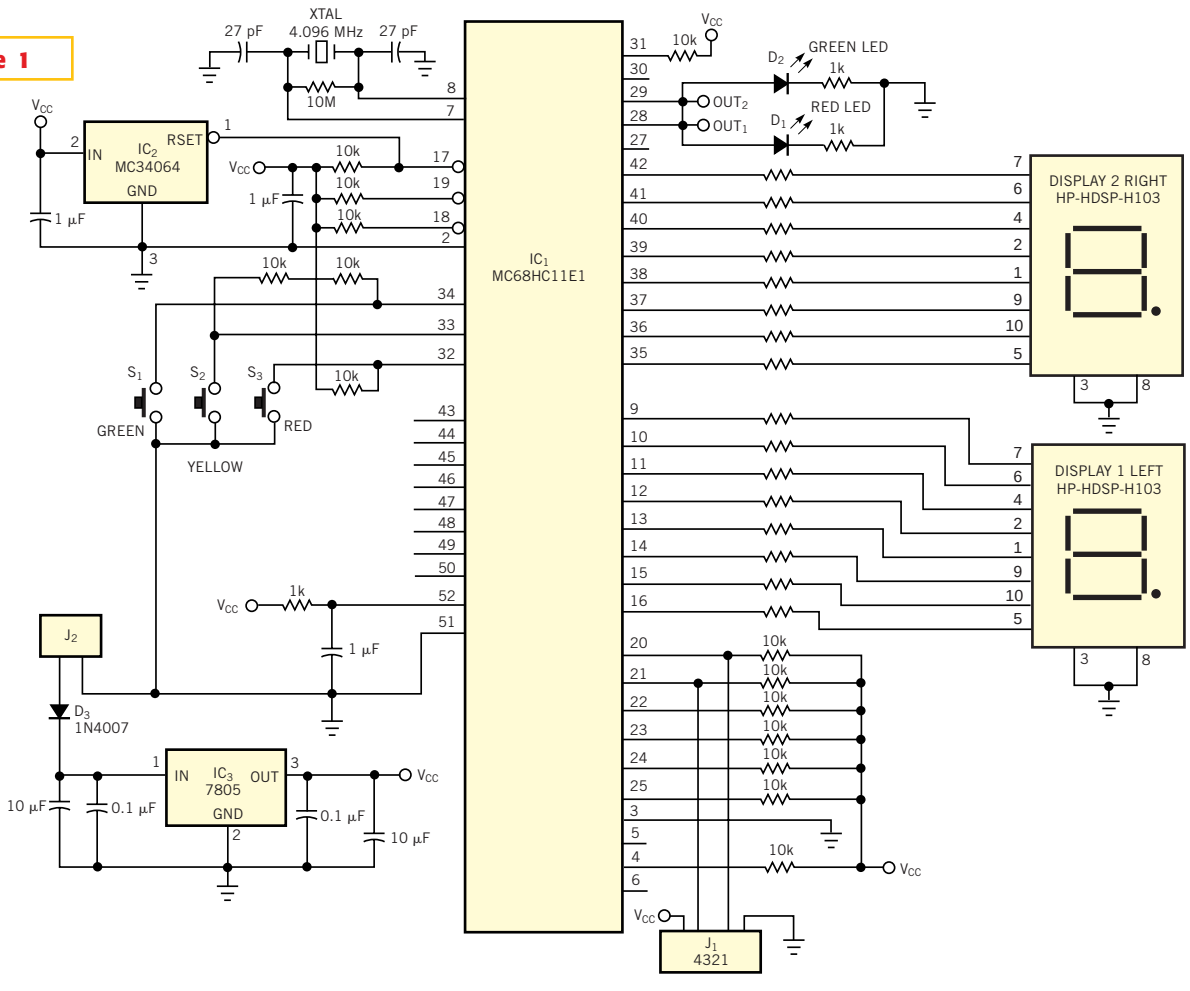
µC provides timer function

Tito Smailagich, ENIC, Belgrade, Yugoslavia

The circuit in Figure 1 is a µC-based programmable timer with two output channels. The first channel, activated by pressing the red switch, S_3, has a red LED at its output. This channel is active until it reaches its desired time-out point. The second output channel connects to a green LED and is active after a preselected time-out period. The second channel remains active until the next depression of the red switch. You can deactivate both channels at any time by pressing the green switch, S_1. In normal mode, the display shows the current remainder of the desired time in seconds. The display decrements by 1 until it reaches 0. The timebase in seconds derives from the main oscillator of the µC, which generates a real-time interrupt every 8 msec. The µC multiplies the 8 msec by 125, yielding a timebase of 1 sec. You program the desired time interval by pressing the yellow switch, S_2; the display shows the programmed value. If you need to change the programmed value, pressing the red switch decrements the value by 1 until it reaches 0, after which it starts with 99. If you need to put the timer interval into memory, press the green switch, and the µC writes the value in its internal EEPROM. You can download the software for the MC68HC11E1 µC 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 #2629.

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

Figure 1



An MC68HC11 µC provides flexible timing functions.

design ideas

Edited by Bill Travis and Anne Watson Swager

Open-loop power supply delivers as much as 1W

Christophe Basso, On Semiconductor, Toulouse, Cedex, France

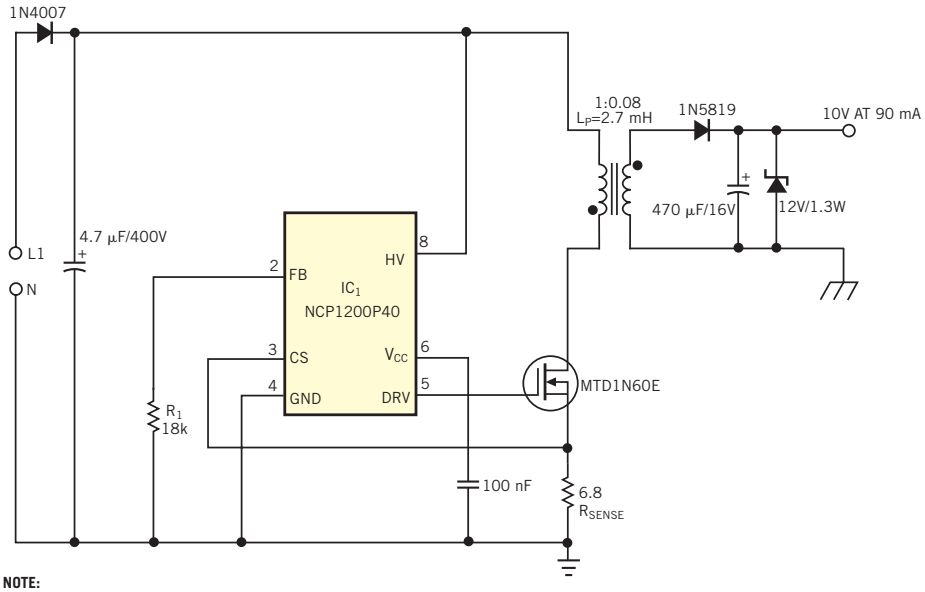
For VCRs, TVs, and other equipment that requires a standby mode, you must supply power to a µP when other components are asleep to receive and interpret any wake-up signal from the remote control or from the broadcasting company. These types of systems have rather low power consumption, and classical switch-mode power-supply ICs represent a clear overkill for less-than-1W output levels. Any active power-supply circuit also needs to be more cost- effective than the standard structure using a metallic transformer. The circuit in **Figure 1** reduces the cost by eliminating the use of the optocoupler.

IC_1 directly drives an external 600V MOSFET. The lack of an auxiliary winding greatly simplifies the overall application circuitry; the controller's integrated dynamic self supply provides V_{CC}. IC_1 works as a peak-current PWM controller, combining fixed-frequency operation at 40, 60, or 100 kHz and the skip-cycle method for low standby-power consumption. IC_1 regulates the peak current and allows operation over universal mains. Because the circuit operates at constant output power, the following formula determines the necessary peak current:

$$I_P = \sqrt{\frac{2 \bullet P_{OUT}}{L_P \bullet F_{OSC}}}.$$

With an internal error amplifier that

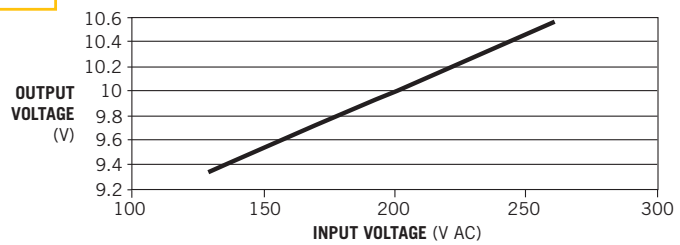
Figure 1



NOTE:
THE TRANSFORMER IS AVAILABLE FROM ELDOR (ELDOR@ELDOR.IT, REF 2262.0058C) AND FROM COILCRAFT (INFO@COILCRAFT.COM REF Y8844-A).

IC_1 regulates the peak current and allows this 1W supply to operate from universal mains.

Figure 2



The input-voltage rejection stays within 1V from 130 to 260V ac.

clips at 1V maximum, R_{SENSE} is equal to $1/I_p$(maximum). In this example, a 40-kHz circuit and a 6.8Ω sense element deliver as much as 1W of continuous power with L_p=2.8 mH. You can recompute R_{SENSE} for lower or higher output-power requirements. The 12V zener diode prevents the circuit from generating overvoltages. R_1 deactivates the internal short-circuit protection, which normally reacts upon feedback-path loss.

Thanks to its avalanche capability, the MTD1N60E requires no clipping network, which further eases the design. The efficiency measured 64% (low line, P_{OUT}=866 mW) and 61% (high line, P_{OUT}=1.08W). **Figure 2** plots the input-voltage rejection, which stays within 1V from 130 to 260V-ac mains. This **figure** illustrates current mode's inherent audio susceptibility.

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

Four-way remote control uses series transmission

JM Terrade, Clermont-Ferrand, France

A simultaneous four-way remote-control system adheres to size, cost, and reduced-complexity constraints and uses a series transmission to drive parallel loads (**Figure 1**). You can use this system as long as the time constant of the load is much larger than the total transmission time for all data. With these considerations, this design can drive any object with four simultaneous controls as motors.

The design uses a 9-bit data packet. The emitter side of the design converts 4 data bits and a 5-bit ID code from parallel to serial. The data packet continuously transmits, and the total information arrives at the HF 433-MHz emitter. The receiver side converts the 9-bit serial data to parallel data. Then, the design compares the received ID code to the local code. The comparison result clocks the 4 data bits for the D latch. This configuration actually controls a small, battery-powered boat with two-way, remote-control switches. The switches are mom-off-mom types, which give front-stop-rear and left-center-right commands. The boat has two dc motors for propulsion and direction. The transmission uses two 433-MHz, AM-radio modules for the HF link.

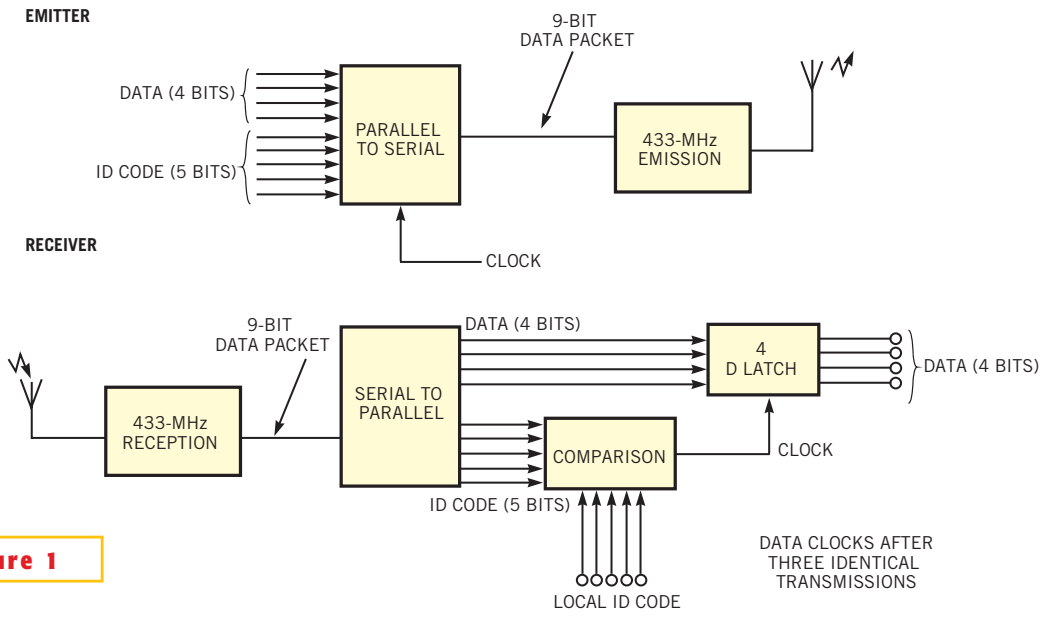


Figure 1

The emitter converts 4 data bits and 5 ID-code bits to serial data and continuously transmits the resulting data packet. The receiver compares the received ID code with the local code three times before clocking in new data.

Power consumption is 10 mA during emission, so the emitter circuit can use a 9V battery (**Figure 2a**). D_1 protects the device against polarity inversion. S_1 and S_2 are three-position, mom-off-mom switches. Only the center, or null, position is static. The user must push the switch in one direction and maintain it to keep the desired action. When released, the switch returns to its null position. With no action on S_1 and S_2, the logic levels on data inputs D_6 to D_9 of IC_1 are low due to R_3 to R_6. When an action occurs on S_1 or S_2, the corresponding data input of IC_1 is close to 5V. You can activate S_1 and S_2 at same time. Voltage-divider pairs R_1 and R_3 or R_1 and R_4 and R_2 and R_5 or R_2 and R_6 produce acceptable levels for IC_1 inputs.

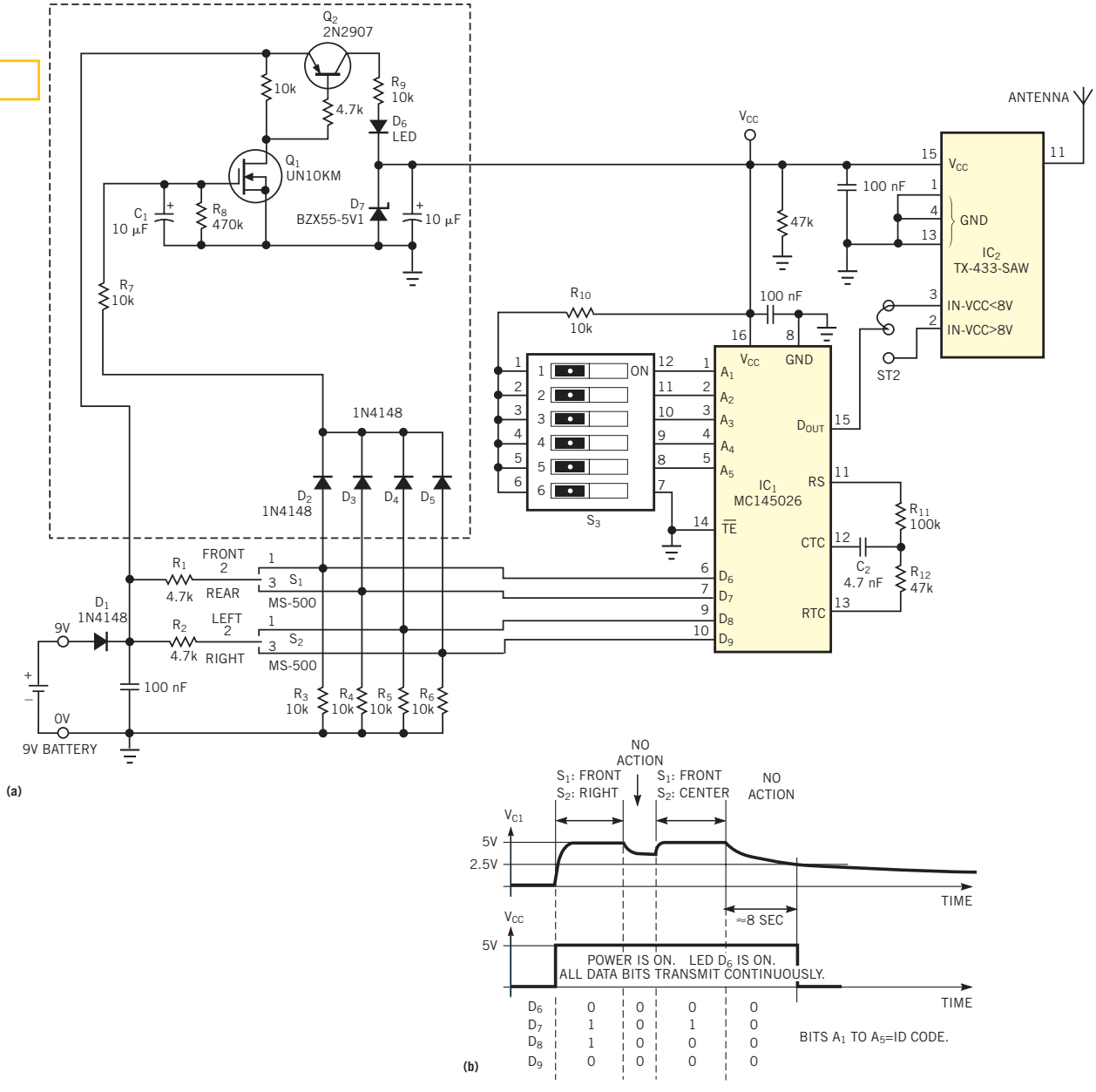
Diodes D_2 to D_5 permit C_1 to charge through R_7. Then, Q_1 conducts, and Q_2 is on. D_6 acts as a power-on indicator. The voltage drop across D_6, R_9, and zener-diode D_7 results in a 5V supply for IC_1 and IC_2. C_1 continuously charges until S_1 and S_2 return to the null position. Then, C_1 discharges through R_8, and Q_1 switches off after approximately 8 to 10 sec (**Figure 2b**).

Inputs A_1 to A_5 of IC_1 are three-state inputs: low, high, and unconnected level. Thus, 243 combinations (3^5) are possible. However, three-state DIP switches are expensive, and 64 possibilities are enough for many applications. If Pin 6 of S_3 provides a low level, A_1 to A_5 can be either low levels or unconnected. If Pin 6 of S_3 provides a high level through R_{10}, A_1 to A_5 can be either high levels or unconnected. This arrangement gives 64 combinations.

R_{11}, R_{12}, and C_2 form the local oscillator. The output of IC_1 at Pin 15 provides the 9-bit data packet to the HF emitter, IC_2. The HF module uses amplitude modulation. The antenna is a 17-cm wire that attaches directly to the pc board. When the power is on, transmission always occurs. After a user releases S_1 and S_2, the emitter continues to transmit the null-position information until power goes off, which takes approximately 8 sec.

On the receiver side (**Figure 3a**), the antenna is also a 17-cm wire attached directly to the pc board. The incoming sig-

Figure 2



In the emitter circuit, two three-position switches, S_1 and S_2, determine the voltage on C_1 (a) and the voltage levels of data bits D_6 to D_9 of IC_1 (b).

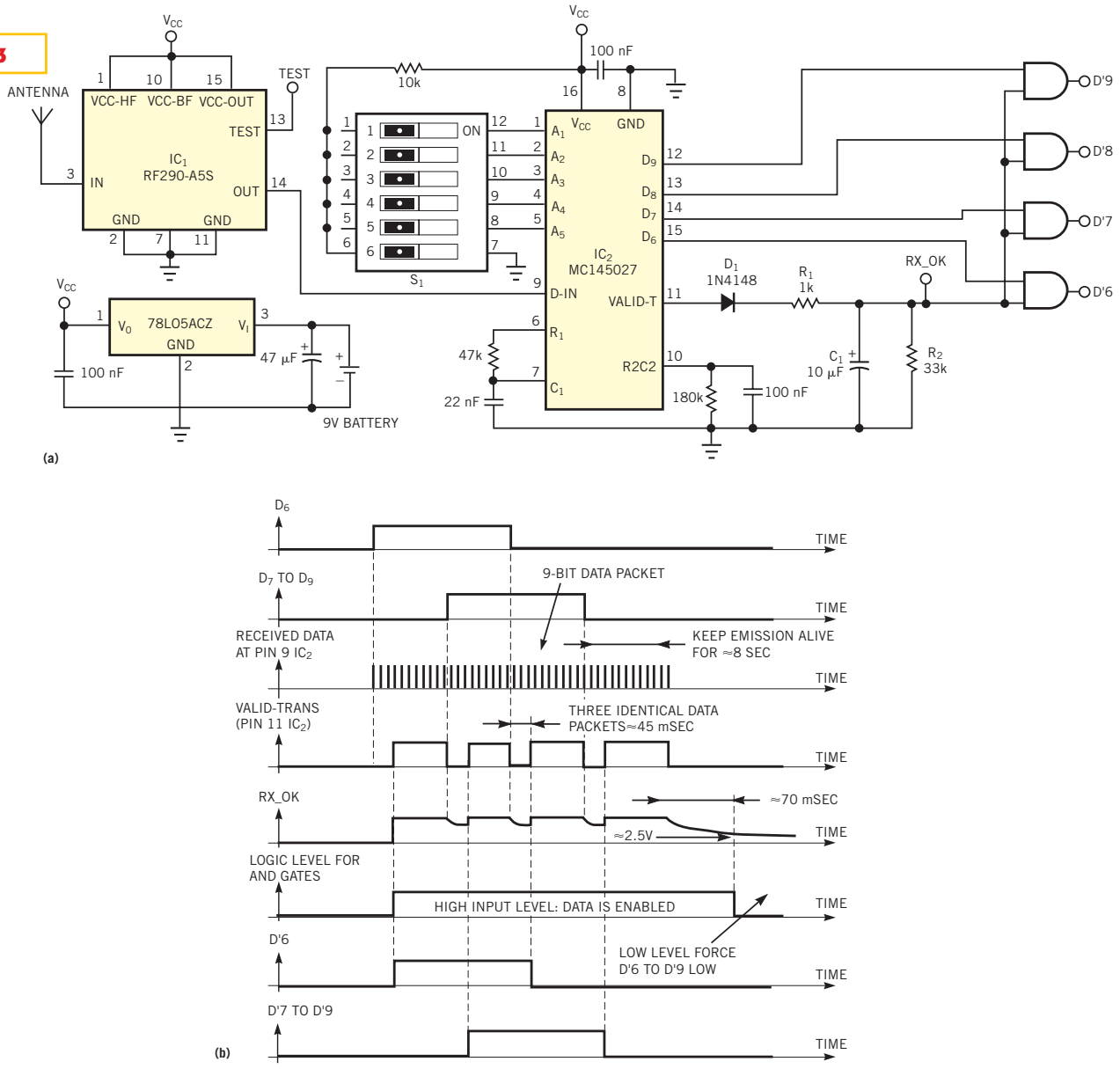
nal arrives at the HF module, IC_1, which has a stable 5V power source. The 9-bit data packet is available at the output, or Pin 14, of the module. Just as for the emitter, DIP switch S_1 provides as many as 64 possibilities for the ID code, and the setting must be the same combination as the emitter.

The 4 data bits are available at outputs D_6 to D_9 of IC_2. When a valid transmission arrives at the receiver, Pin 11 of IC_2 goes high. But each time a user changes the position of the commands on the emitter, the Valid-T signal goes low until the new transmission is valid. Three correct transmissions are necessary. Therefore, the design needs a stable RX_OK signal, and, for this reason D_1, R_1, R_2, and C_1 create a time constant. The RX_OK signal goes low only when the transmission stops or when the ID code is invalid, which can happen if the emitter has no supply and stops emitting or if another transmitter is in the same area (**Figure 3b**).

The internal D latch, IC_2, clocks new output levels only when the circuit receives a new data packet. In this way, when only one transmitted bit changes, the other bits keep their previous level. When the ID code is not valid or when the HF link is lost, which implies that the distance between the emitter and the receiver is too long, D_6 to D_9 keep their previous levels. However, RX_OK goes low after 70 msec and forces D_6 to D_9 to go low.

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

Figure 3



In the receiver (a), three correct transmissions must occur before Pin 11 of IC_2 goes high (b).

Analyze LED characteristics with PSpice

Sam Mollet, GE Harris Harmon Railway Technology, Grain Valley, MO

Recent advances in LED technology have lead to LEDs' widespread use in outdoor-signal applications, such as in traffic and railroad signals. A typical LED signal consists of an LED array and a power supply. When a low-voltage power supply is either desirable or mandatory, series/parallel combinations of LEDs become inevitable. However, analyzing and optimizing series/parallel combinations of LEDs with varying forward characteristics can be complicated. Using the parametric and Monte Carlo capabilities of PSpice greatly simplifies this task.

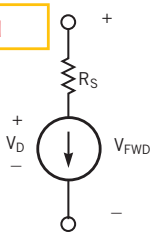
To model an LED in PSpice, use the diode model. You can set the IFK and ISR parameters in the diode model to zero; **Figure 1** shows the resultant PSpice diode forward-current model and corresponding equations. As the equations in the **figure** show, you can express the forward voltage across the diode model, or V_{FWD}, as the sum of the voltage across the series resistance and the voltage across the intrinsic diode.

The dominant term in the V_{FWD} equation of **Figure 1**, assuming R_S is less than 10Ω, is the logarithmic term. Therefore, if you vary the model parameter N in Monte Carlo or parametric analyses, then the V_{FWD} varies accordingly. A helpful hint: When creating an LED model using programs such as Parts (www.microsim.com), use curve-tracer plots or an enlarged photocopy of the VI curve from data books to extrapolate data points along the VI curve.

Figure 2 shows an example for which N varies linearly between 2.07 and 2.53, or 2.3±10%. The forward voltage at 20 mA varies from 1.59 to 1.94V, or 1.765±9.9%. By editing the "N=2.3299" statement in the LED model to "N=2.3299 DEV 10%" assigns a 10% device tolerance to the LED model. Therefore, when you execute a Monte Carlo analysis, the forward characteristics of each LED in the circuit vary randomly. **Figure 3**'s example performs 20 Monte Carlo sweeps at 1V/sec, with N set for a 10% tolerance.

The final example is the analysis of a simple circuit (**Figure 4a**). The input consisted of a 60-mA

Figure 1



$$I_{FWD}=I_S\cdot(e^{V_D/N\cdot V_T}-1).$$

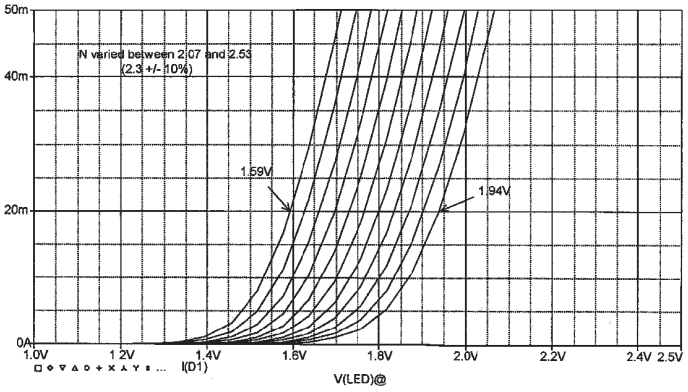
$$V_D=N\cdot V_T\cdot\ln\left(\frac{I_{FWD}}{I_S}+1\right).$$

$$V_{FWD}=I_{FWD}\cdot R_S+N\cdot V_T\cdot\ln\left(\frac{I_{FWD}}{I_S}+1\right).$$

This simple model and equations are the result of setting PSpice's diode-model parameters IFK and ISR to zero.

Figure 2

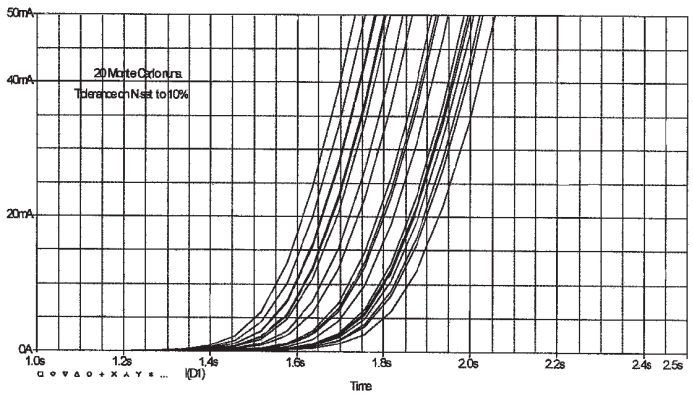
```
LED Model:
.MODEL LED_RED
IS=5.5526E-15
N=2.3299
RS=2.2974
CJO=2.9700E-12
M=.3333
VJ=.75
BV=4
IBV=10.000E-6
TT=4.3200E-6
```



This simulation run varies N linearly from 2.07 to 2.53.

Figure 3

```
LED Model:
.MODEL LED_RED
IS=5.5526E-15
N=2.3299 DEV 10%
RS=2.2974
CJO=2.9700E-12
M=.3333
VJ=.75
BV=4
IBV=10.000E-6
TT=4.3200E-6
```

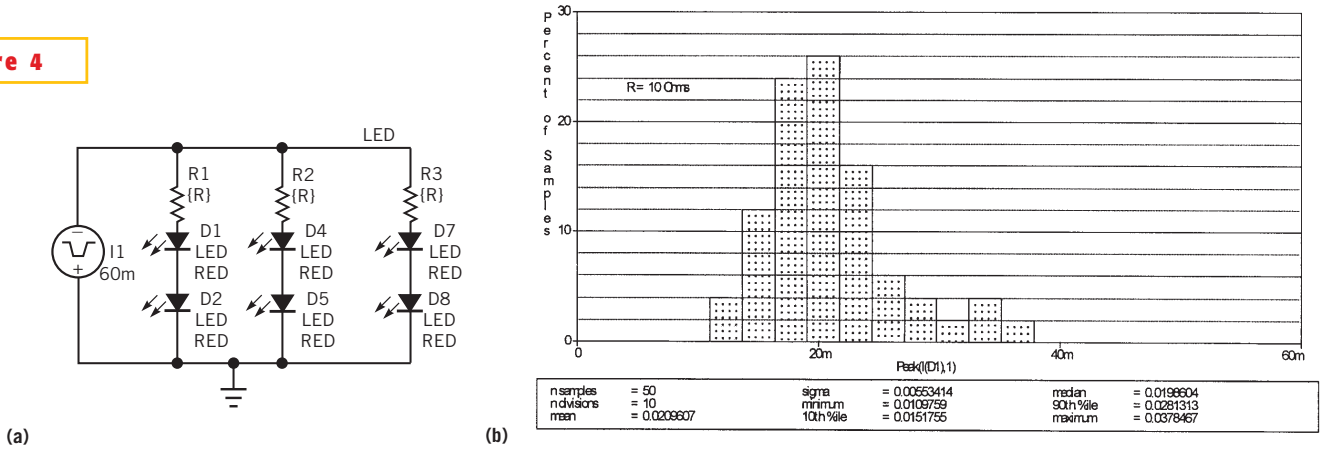


Assigning a 10% tolerance to N causes the forward characteristics of each LED in the circuit to vary randomly during Monte Carlo analysis.

pulse, and the simulations determine the peak current through D_1 for 0, 10, and 100Ω resistance values. The model statement assigned a 10% tolerance to N, and the example executes 50 Monte Carlo runs. The results for R=0 reveal a large standard deviation of 10 mA. The results for R=10 reveals a smaller standard deviation of about 5 mA (**Figure 4b**). The results for R=100 reveals a small standard deviation of only 1 mA.

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

Figure 4



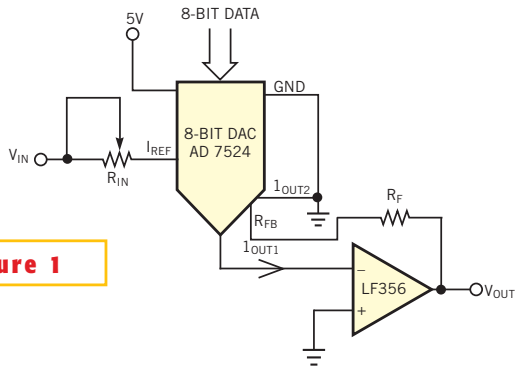
To analyze a simple circuit (a), simulations determine the peak current through D1 for three resistance values. The results for R=10Ω reveal a standard deviation of approximately 5 mA (b).

Programmable-gain amplifier is low-cost

J Jayapandian, Indira Gandhi Centre for Atomic Research, Kalpakkam, India

Numerous programmable-gain amplifiers are available, but a simple solution provides the option of using 256 gain steps with an 8-bit DAC and higher steps with higher bit DACs (**Figure 1**). According to the inverting-amplifier configuration of an op amp, the output voltage is $V_{OUT}=V_{IN}(R_F/R_{IN})$, where R_F is the feedback resistance, R_{IN} is the input resistance, and V_{IN} is the input voltage of the amplifier circuit. Generally, by changing the feedback resistance, you can get the desired gain.

Figure 1



A DAC in series with an op amp attenuates the input signal to achieve the variable-gain factor.

In this design, the 8-bit DAC in the input stage acts as a programmable attenuator for the input signal and permits a maximum full-scale I_{OUT1} of 1 mA. The value of I_{OUT1} is proportional to the input-voltage signal. The shunt feedback resistance, R_F, converts I_{OUT1} to a voltage. Thus, the input signal, V_{IN}, acts as a reference input to the DAC. Instead of increasing the value of the feedback resistor for higher gain, this circuit uses the DAC in series with the op amp to attenuate the input signal and achieve the desired variable-gain factor. You calculate the current output, I_{OUT1}, from the DAC as follows, where D_0 through D_7 are the digital inputs to the DAC:

$$I_{OUT1}=\frac{V_{IN}}{R_{IN}}\left(\frac{D_0}{2}+\frac{D_1}{4}+\frac{D_2}{8}+\frac{D_3}{16}+\frac{D_4}{32}+\frac{D_5}{64}+\frac{D_6}{128}+\frac{D_7}{256}\right).$$

For example, if all of the bits are ones, the 8-bit digital image is FF, and the corresponding amplifier full-scale output is:

$$V_{OUT}=I_{OUT1}\bullet R_F=\frac{V_{IN}}{R_{IN}}\left(\frac{255}{256}\right)\bullet R_F.$$

In an actual application, keep the value of R_F fixed for the maximum gain. By varying the digital image pattern from 00 to FF, you can get the variable amplifier gain according to your requirements.

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

PC hardware monitor reports the weather

Sean Gilmour, Analog Devices, Limerick Ireland

You usually use PC hardware monitors to keep a close eye on power-supply voltage levels, the speed of system cooling fans, and even the temperature of the CPU. Until fairly recently, this level of system monitoring was reserved for high-end servers running mission-critical applications. However, now that low-cost hardware monitoring ASICs are available, advanced hardware monitoring has become a standard feature in most new PCs. And hardware monitors are now finding their way into diverse applications, such as weather stations (**Figure 1**).

IC_1 has two external temperature-measurement channels. One channel connects to a resistive humidity sensor, and a second channel uses a 2N3906 transistor to sense the outdoor temperature. The internal temperature sensor measures the indoor temperature. One of the tachometer inputs connects to the output of a wind-speed meter. For each of the measurement inputs, you can set limits that warn the user of changing weather conditions. IC_1 uses a switching-current-measurement scheme, so you can mount the sensors hundreds of feet from the IC and still maintain a high SNR.

IC_1 connects to a parallel printer port using a 74HC07 open-drain noninverting buffer. Pin 2 of the parallel port is the serial clock. Pin 3 writes configuration data into IC_1, and Pin 13 reads data from IC_1.

The necessary software is simple, and the parallel-printer port is easily accessible using freeware drivers and DLLs that you can find on the Internet. You can bit-bang the SCL and SDATA lines using a programming language such as Visual Basic or Visual C++.

The temperature-measurement channels use a thermal diode, such as that on Intel's Pentium processors (PII+), or a discrete npn or pnp transistor. These channels use a two-wire scheme that supplies switching current levels to the transistor. IC_1 measures the difference in V_{BE} between these two currents and calculates the temperature according to the following well-known relationship:

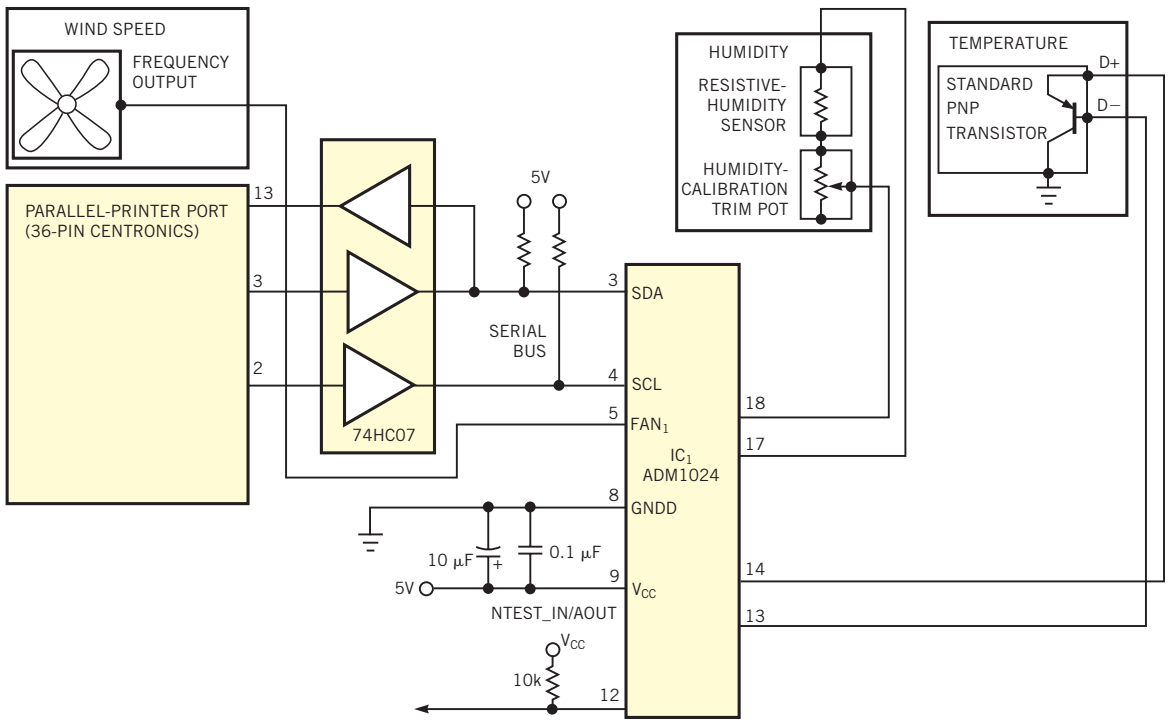
$$\Delta V_{BE} = KT/q \times \ln(N),$$

where K is Boltzmann's constant, q is the charge of an electron, T is the absolute temperature in Kelvin, and N is the ratio of the two currents.

You can also use the CPU temperature-monitoring channels to measure changes in resistance, making them useful for most resistive sensors, including photo diodes, photo resistors, gas sensors, and resistive-humidity sensors.

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

Figure 1



A PC hardware-monitor IC can also monitor weather-station characteristics.

design ideas

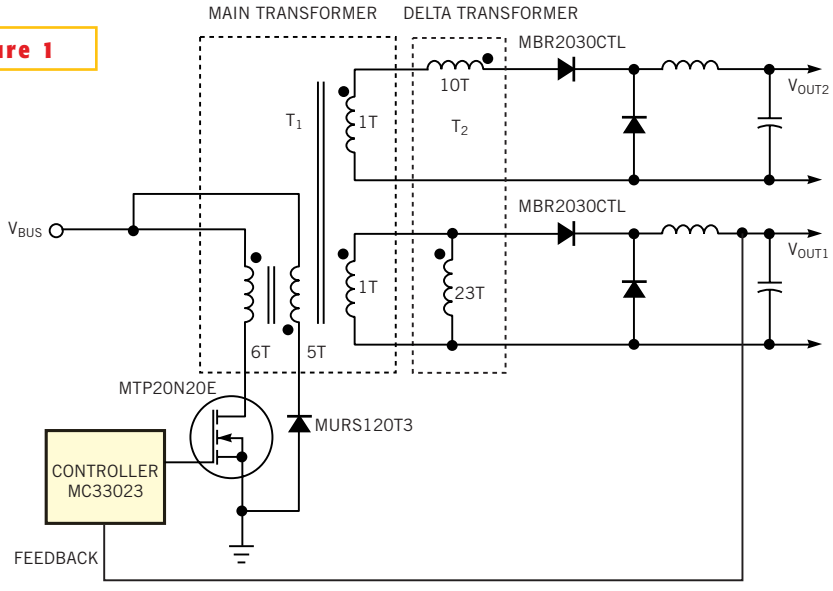
Edited by Bill Travis and Anne Watson Swager

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.

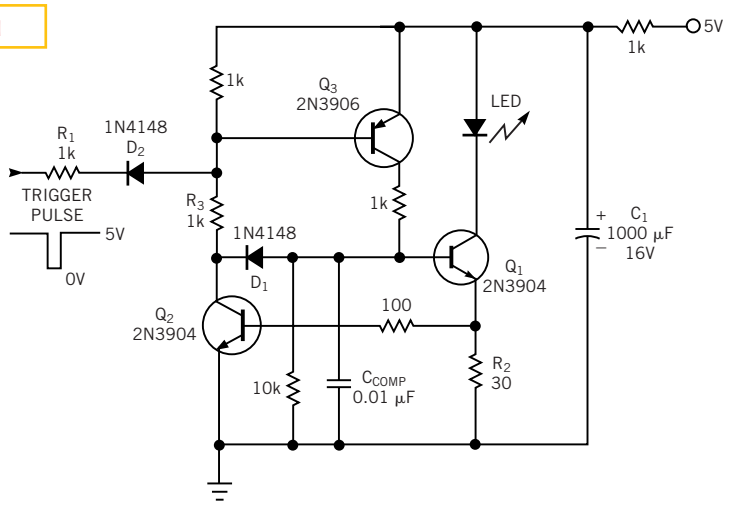
Circuit forms constant-current SCR

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.

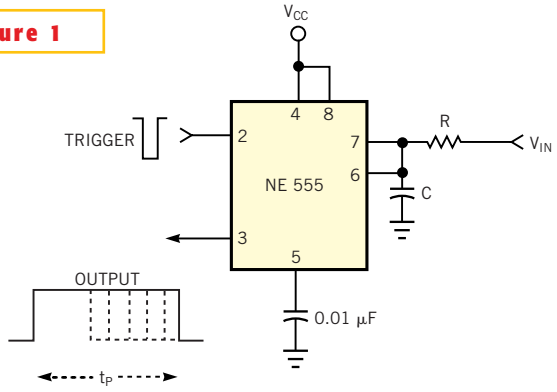
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

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.

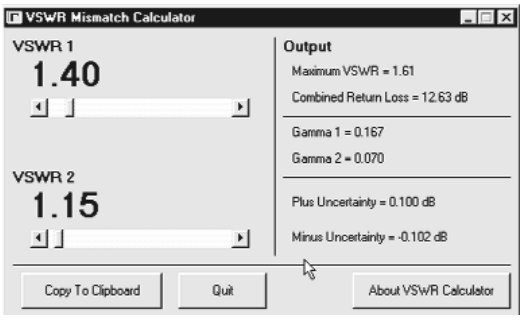


Figure 1 **Given the source and load VSWRs, the VSWR Calc program quickly calculates RF measurement uncertainties.**

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.

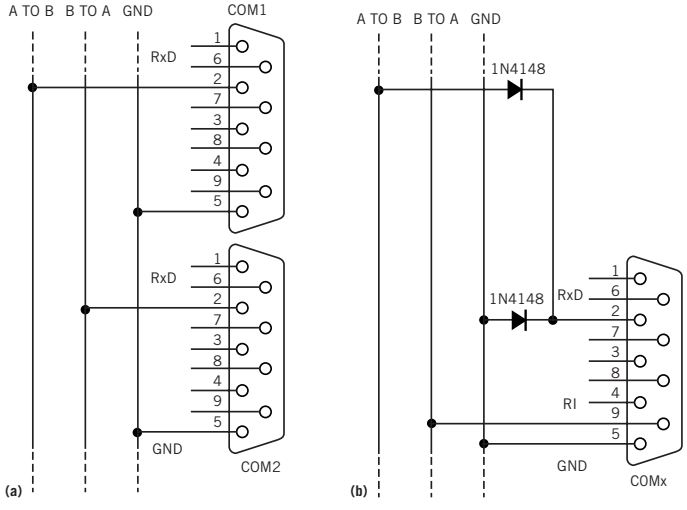
PC monitors two-way RS-232 transmission

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.

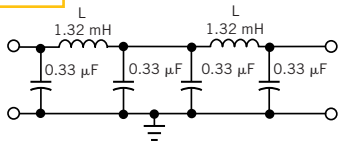
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

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.

Figure 1



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

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.

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		

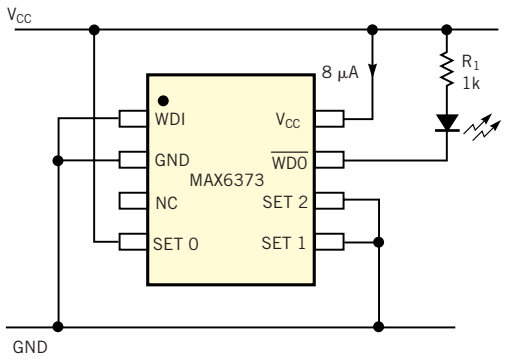
Watchdog timer assumes varied roles

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.

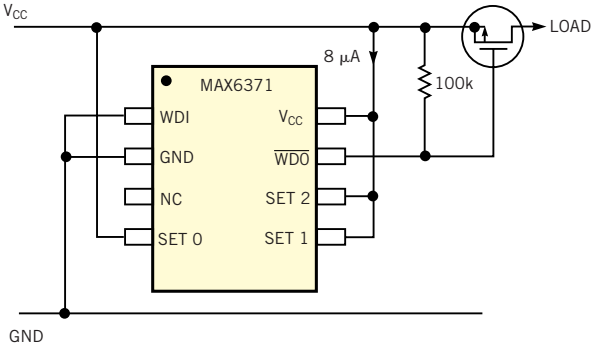
The circuit in **Figure 1** uses a MX6373 to pulse $\overline{WDO}$ low for 170 msec every 5.2 sec. The load is a front-panel power-on LED with a 1-kΩ current-limiting resistor. By pulsing the LED rather than powering it continuously, the average current decreases by a factor of 30 (88 µA versus 2.4 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



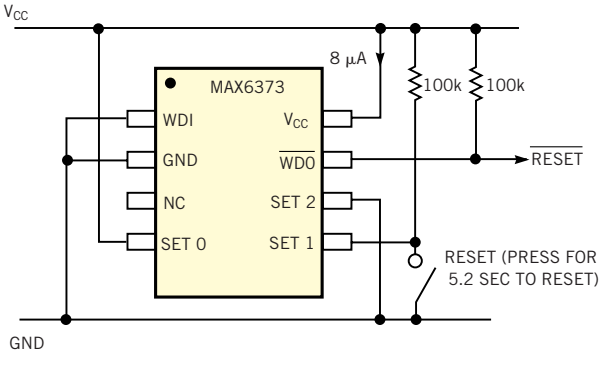
A blinking LED allows a 30-to-1 average-current reduction in a power-on indicator.

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.

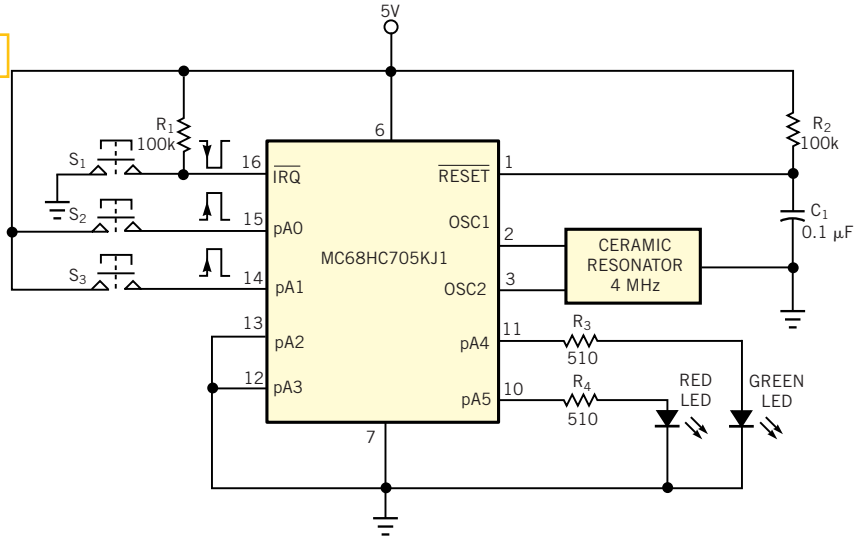
Is this the best Design Idea in this issue? Vote at www.edmag.com/ednmag/vote.asp.

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.

design ideas

Edited by Bill Travis and Anne Watson Swager

Keep the heat down in power op amps

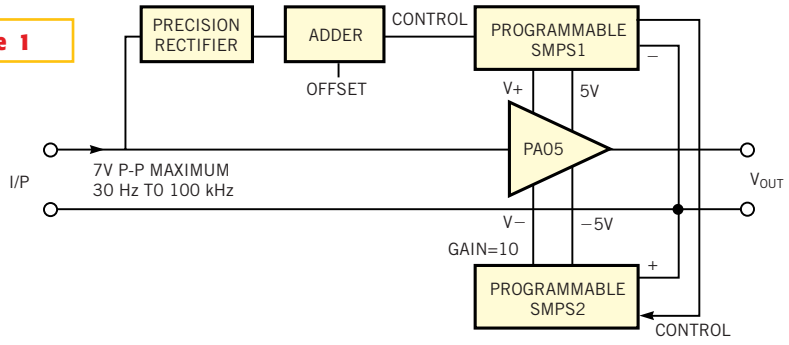
Vijay Damle, Digitronics, Pune, India

When you include a power op amp, such as PA05 from Apex Microtechnology, in your design, it is desirable to minimize the supply-to-output differential to a minimum to reduce power dissipation and to fully exploit the amplifier's output range. Our goal was to design a power amplifier to yield 70V p-p output at 10A with a fixed gain of 10 and a frequency of 30 Hz to 100 kHz. To obtain ±35V swing entailed dc supplies of approximately ±38V and two ±5V supplementary supplies. To derive the full 10A at lower voltage, you must reduce the supply voltage in proportion to the output voltage to decrease dissipation. In this case, the gain is fixed at 10. So, you can control the dc voltage proportional to the input voltage (**Figure 1**). SMPS1 and SMPS2 are identical voltage-programmed supplies (except for the ±5V supplementary supplies). The precision rectifier generates dc output proportional to the ac-input amplitude. To obtain approximately ±6V when no input signal is present, the circuit adds offset voltage to the signal.

As the input increases, the SMPS output increases from 6 to 38V for a 7V p-p ac input. To control the negative-side SMPS, you must transfer information from SMPS1 to SMPS2. The circuit in **Figure 2**, which generates current proportional to the input voltage, effects the transfer. Thus, SMPS2 generates an equal-value but opposite-polarity voltage to that of SMPS1. If you need higher output current, you may need to increase the voltage headroom, depending on the power amplifier you choose. Otherwise, you may experience output clipping. Note that for fast-changing input signals, the output may clip for a short time until the power-supply voltage rises. This phenomenon depends on the precision rectifier and the power supplies' response time.

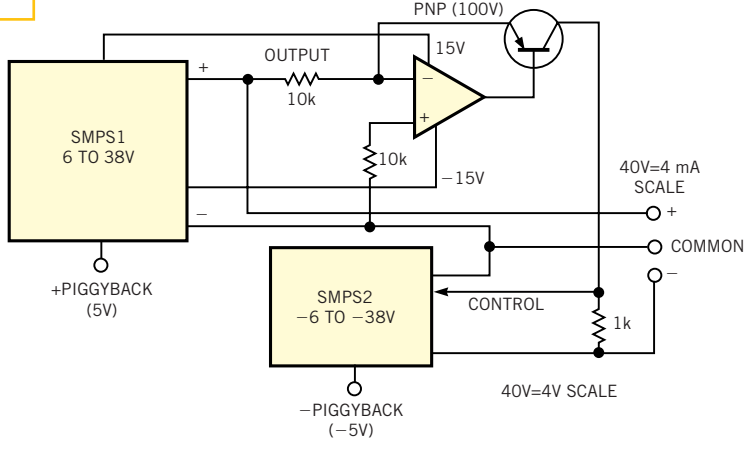
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Figure 1



Power supplies track the input signal in bootstrap fashion, thus reducing power dissipation.

Figure 2



The switch-mode power supplies track each other with opposite-polarity outputs.

Buck converter works efficiently from phone line

Wayne Rewinkel, National Semiconductor, Santa Clara, CA

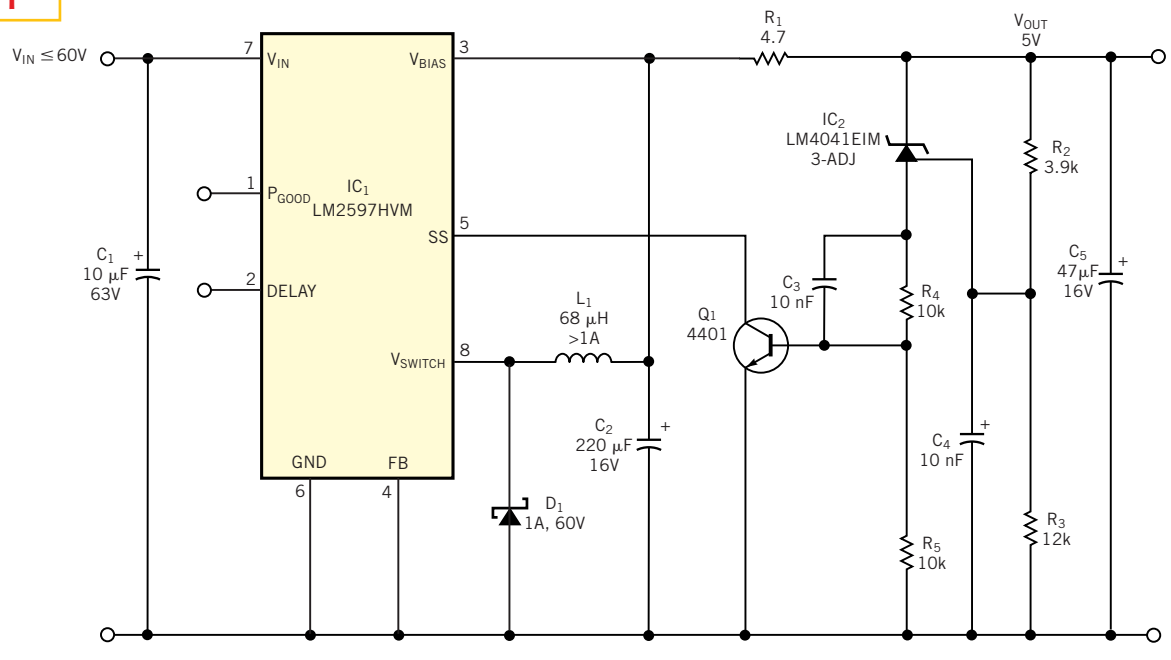
A switching converter provides an inexpensive way to generate 5V, 18 mA (48V, 5 mA maximum) directly from a standard phone line (**Figure 1**). The high input voltage and low available current require a unique design approach to achieve high efficiency. The circuit uses the LM2597HVM, which has a 60V rating and power-saving features. Its VB_{IAS} pin permits bootstrapping bias power whenever the output is higher than 4.4V. This feature reduces the bias current by at least a factor of four to a typical current of 1 mA. Additionally, the IC has a shutdown/soft-start pin that, when pulled low, shuts the regulator off (with 10-μA maximum quiescent current). When you release this pin, the IC starts switching with an increasing duty cycle. These features combine with an external comparator and voltage reference to provide the basis for a low-power switching regulator.

The circuit uses the LM4041 adjustable-voltage reference with Q_1 to form a small, low-power comparator. Its 1.23V threshold controls the regulator's output voltage according to the following expression: $V_{OUT}=1.23V\cdot(1+R_3/R_2)$. If the divided output voltage is below the threshold, IC_1 turns on, causing V_{OUT} to ramp up until the voltage crosses the threshold. Then Q_1 pulls Pin 5 low, forcing IC_1 into shutdown. IC_1 stays in this state until the output voltage again decays to the comparator's threshold and the cycle repeats. In this always-stable, discontinuous-switching mode, the inductor current is many times the load current, so its stored energy at turn-off always forces a slight overshoot that has an effect similar to hysteresis. This overshoot is the ripple voltage (ignoring capacitor ESR); you can estimate it from the energy transferred to C_2+C_5 during a switching cycle. The following equation gives ripple: $V_{RIPPLE}=\sqrt{((V_{OUT}\cdot V_{OUT})+L_1/(C_2+C_5))}$.

R_1 and C_5 remove the large ESR-induced spike from C_2 caused by the 1A peak charging current. This lowpass filter is small and inexpensive, and it allows the use of capacitors with almost any ESR ratings. C_3 and C_4 speed the comparator action and thus reduce output ripple. R_5 ensures that the 4041 receives sufficient bias current, and R_4 sets the maximum bias current. L_1 should have a value lower than 100 μH and must have a saturation rating exceeding 1A. Coilcraft's DO3316683 inductor fills the bill nicely. Smaller inductors degrade efficiency, because the 2597 goes into its pulse-current-limited protection mode. Larger values also degrade efficiency by using additional switching cycles. The circuit in **Figure 1** provides satisfactory results with input voltages of 10 to 60V. For an input of 48V, 5 mA, the available output current measures 34 mA. The output regulation is less than 2-mV output variation for inputs of 10 to 60V and load currents of 0 to 100 mA.

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

Figure 1



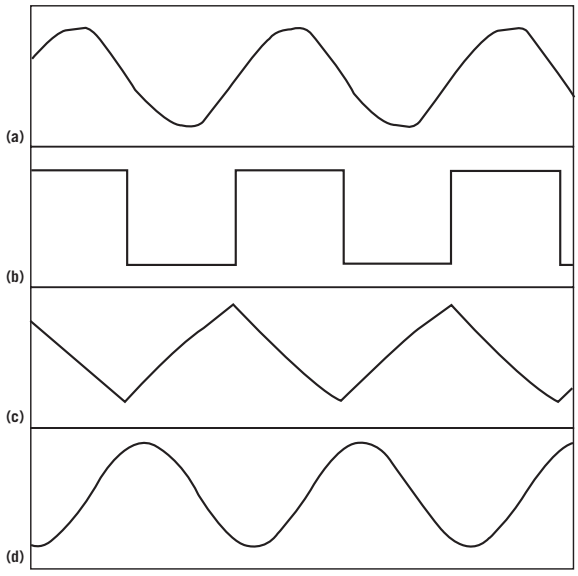
This inexpensive switching regulator derives its power directly from the phone line.

Sine reference is synchronous with ac line

P Seshanna, PhD, Assumption University, Bangkok, Thailand

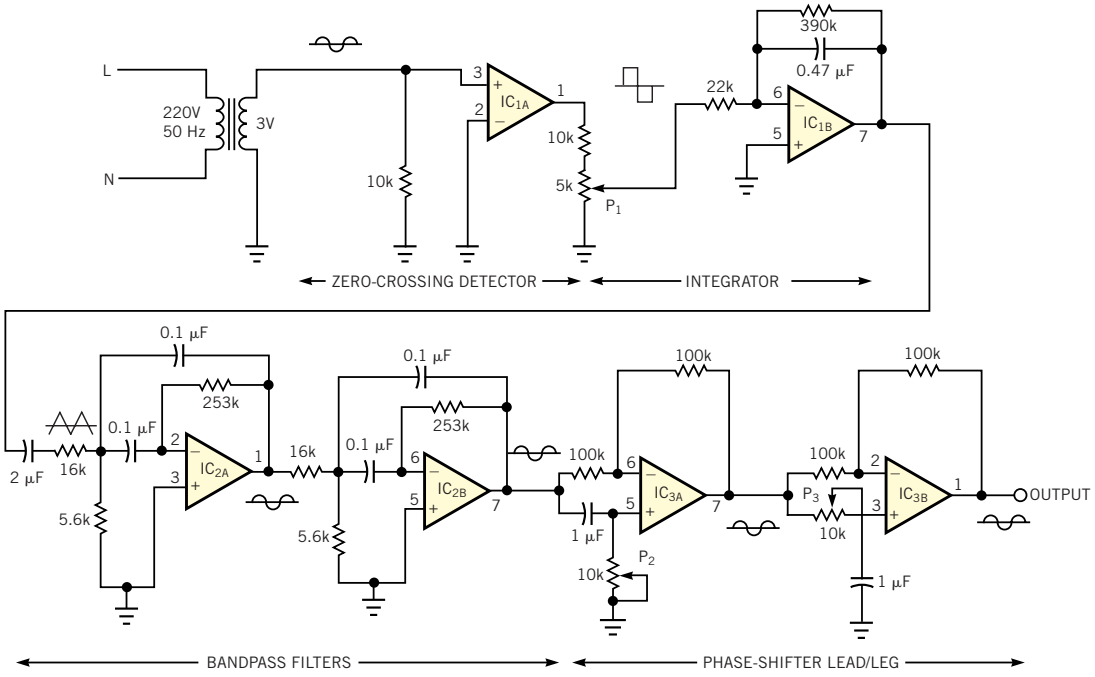
Many applications require a sinusoidal reference voltage synchronized to the ac line voltage. You cannot derive such a reference voltage directly from the ac line because the waveform of the ac line is distorted because of nonlinear loads connected to the line and because the amplitude of the line signal varies. Hence, you cannot use a simple step-down transformer to derive the reference signal. The circuit in **Figure 1** develops a line-synchronized reference signal whose phase and amplitude you can adjust using potentiometers P_1, P_2, and P_3. The circuit steps down and converts the line voltage to a square wave using the IC_{1A} zero-crossing detector. The integrator block, IC_{1B}, then integrates the square wave to produce a triangular wave. The triangular wave passes through two narrow bandpass filters comprising IC_{2A} and

Figure 2



Distorted line voltage (a) produces a square wave (b). An integrator creates a triangular wave (c), and bandpass filters produce a pure sine wave (d).

Figure 1



An op-amp circuit uses only resistors and capacitors to generate a line-synchronized sine wave.

IC_{2B}. The center frequency of the bandpass filters matches the 50-Hz line frequency. The sine-wave output from the filters passes through two phase shifters to set the phase at either lead or lag. The allpass-filter configuration comprising IC_{3A} and IC_{3B} sets the phase of the reference signal. You can adjust the amplitude of the output using potentiometer P_1, and you can adjust the phase with P_2, P_3, or both. The circuit uses three low-power LM358 dual op amps. **Figure 2** shows the measured waveforms at different points in the circuit. The reference sine wave has total harmonic distortion of 0.7%. Note that the circuit requires no heavy, bulky inductors.

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

Cascade bandpass filters for higher Q

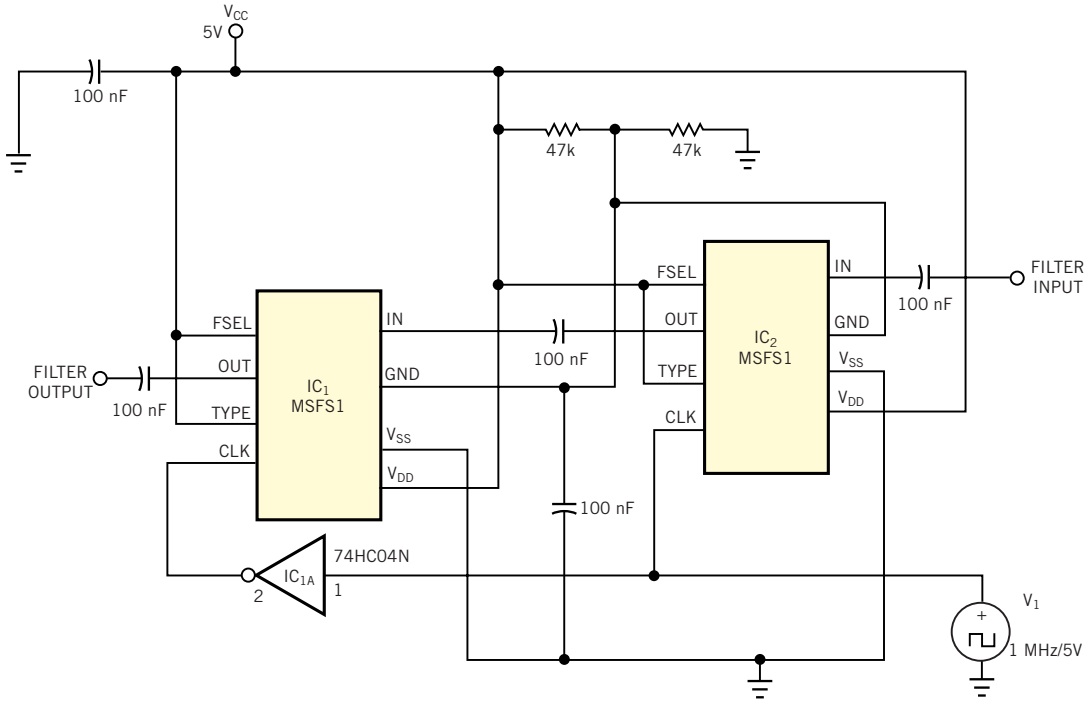
Trong Huynh and John Ambrose, Mixed Signal Integration Corp, Santa Clara, CA

Switched-capacitor filters that are preset for a given bandwidth sometimes do not deliver the bandwidth or Q an application requires. By inverting the clock between two switched-capacitor bandpass filters, such as the MSFS1 from Mixed Signal Integration Corp, you can configure a high-Q filter (**Figure 1**). The MSFS1 is a selectable, seventh-order, lowpass/bandpass, switched-capacitor filter. Using the FSEL pin, you can select either a lowpass or a bandpass response. With the TYPE pin, you can select a Butterworth, a Bessel, or an elliptic response if FSEL is low. When FSEL is high, you can select full-, third-, or sixth-octave bandpass response. The circuit in **Figure 1** shows both ICs set for sixth-octave bandpass response.

The clock-to-corner ratio of the MSFS1 is 50-to-1. With switched-capacitor filters, changing the clock frequency moves the center or corner frequency of the response by the same amount. For example, if the input clock to the MSFS1 is 1 MHz, the corner frequency is 20 kHz. Cutting the clock frequency to 500 kHz results in a corner frequency of 10 kHz. By inverting the clock between the two filter ICs, you obtain a 10th-octave filter. If you simply cascade the filters, without the clock inversion, the signal has a delay to the second filter equal to the clock period. The result is an increase in passband ripple and no change in the Q of the filter.

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

Figure 1



By inverting the clock to one switched-capacitor filter, you obtain a cascaded filter with enhanced Q.

C program calculates checksums

Ken Levine, Airshow Pacific Systems, Kirkland, WA

To ensure data integrity, it's wise to frequently calculate file checksums. The C program in **Listing 1** calculates the checksum of a file using a 16-bit CRC (cyclic redundancy check). The program assumes an 8-bit byte size. The routine reads the file as binary and processes one byte at a time. The CRC formula the program uses is X16+X12+X5+1, with a starting value of hexadecimal FFFF. The program displays the number of bytes processed. The file calccrc.c starts by including the header files needed. Next, the routine defines and initializes the constants. (The program does not use the C++ keyword const; therefore, a C compiler can compile the program.) Inside main(), the program defines variables and issues the initial starting value for the CRC. The routine performs a check to verify that at least two arguments passed to main(). If only one argument passes, the program terminates with a message that you need to supply a file name.

The program reads and processes one byte at a time until it reaches the end of the file. Each time the program reads a byte, the byte counter increments. At the end of the file, the program displays the CRC of the file and the number of bytes read. The routine calculates the 16-bit CRC, byte by byte, using the calcCRC16 function. This function passes the byte to be processed and the current value of the CRC and returns the new CRC value. The program calculates the CRC for each bit in the byte. The variable temp is assigned the current CRC value right-shifted 15 times, XORed with the current byte value right-shifted seven times. This operation XORs the MSB of the CRC with the MSB of the byte, so temp will have the value zero or one. Note that this operation does not change the values of the CRC or the byte. The CRC then left-shifts one place. If temp is 0, nothing happens. If temp is 1, the program XORs the CRC with hexadecimal 1021 (the X12+X5+1

LISTING 1–C PROGRAM FOR CALCULATING CRC

```
/* file calccrc.c
 * calculates the 16 bit CRC of a file
 * in Borland C++, an int is 16 bit
 *
 * file should be read in binary as this will read every character,
 * when a file is read in binary mode, the carriage return (CR) and
 * line feed (LF) are both read; in text mode, only the LF is read
 * should a file have CR in part of the data, text mode may not read
 * it
 */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>

unsigned int INIT = 0xFFFF;  /* initial value of CRC */
unsigned int CRC16 = 0x1021; /* bits 12, 5 and 0 */
int SHIFT_CRC = 15;          /* how far to right shift crc */
int SHIFT_BYTE = 7;          /* how far to right shift byte */
int BYTE_SIZE = 8;           /* number of bits in a byte */


unsigned int calcCRC16(unsigned int, unsigned char);

int main(int argc, char *argv[]){

      unsigned int crc = INIT;
      long byteCounter = 0;
      FILE *Data;
      char ch;

if(argc  < 2){
            printf("\nNeed to supply file name for CRC calculation.");
            printf("\nProgram terminated.");
            exit(1);
      }

/* see if file can be opened*/
      if(NULL == (Data = fopen(argv[1], "rb"))){
            printf("\nUnable to open file %s, program terminated.", argv[1]);
            exit(1);
      }

      while(!feof(Data)){
            /* The end of file character is read and processed
             * by these statements.
            */
            ch = fgetc(Data);
            crc = calcCRC16(crc, ch);
            byteCounter++;
      }

      fclose(Data);
      /* Subtract one from byteCounter to compensate for the
       * end of file character being read. This character is

             * not counted when the operating system shows file
             * size.
            */
         byteCounter--;

            printf("\ncrc of %s is %x",argv[1], crc);
            printf("\n(read %ld bytes)", byteCounter);

            return 0;
      }


      unsigned int calcCRC16(unsigned int crc, unsigned char byte){

      /* Algorithim XOR's bit 15 (the MSB) of the current CRC with the current
       * MSB of byte.  The current CRC and byte are then left shifted by one.
       * This value is then XOR'ed with bits 12, 5 and 0 of the current CRC.
       * This is done until all bits in byte have been processed.
       */
```

(continued on pg 152)

term). Next, the byte left-shifts one place, so the program can process the next bit. This process repeats until the routine processes all eight bits of the byte.

If you wish to use a different formula to calculate the CRC, you need only change the variable CRC16, assuming that the formula still starts with the X16 term). If you wish to calculate a 32-bit CRC, the variables INIT and CRC16 cannot be of type int. You can set the variables SHIFT_CRC, SHIFT_BYTE, and BYTE_SIZE to other values to accommodate various byte and CRC sizes. This program is compiled using Borland C++ 3.0, Borland C++4.5, and Microsoft Visual C++ 1.0. 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 #2674.

LISTING 1–C PROGRAM FOR CALCULATING CRC (*continued*)

```
        unsigned int temp;
        int index;

        for(index = 0; index < BYTE_SIZE; index++){
              temp = (crc >> SHIFT_CRC) ^ (byte >> SHIFT_BYTE);  /*temp is now MSB
of CRC X or'd

         with MSB of byte */
              crc <<= 1;  /* left shift one space */

              if(temp){  /* if temp is 1, then XOR bits 12, 5 and 0 with 1
                                    * if temp is 0, no need to XOR because XOR
with 0
                                    * does not change value */
                    crc ^= CRC16;
              }

              byte <<= 1; /* left shift one to get to next bit */
        }

        return crc;
}
```

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

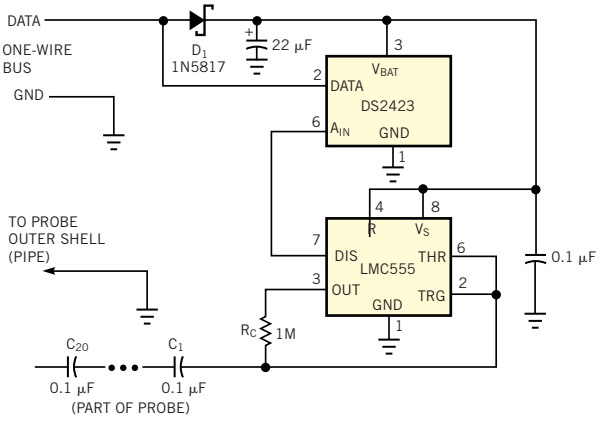
One-wire bus powers water-level sensor

Dale Litwhiler, Lockheed Martin, Newtown, PA

You can use the simple sensor circuit in **Figure 1** to remotely monitor the level of liquid water in a vessel such as a swimming pool. The LMC555 sensor oscillator provides an output-signal frequency that is a function of the water level. This signal drives a DS2423 pulse counter. A host PC or µC reads the output of the pulse counter via the Dallas Semiconductor one-wire bus (**Reference 1**). The circuit uses approximately 150 µA of current, allowing the circuit to steal its power from the bus via the Schottky diode, D_1. Because the circuit is sensing water that is part of the electrical circuit, you should use an ac-coupled signal to avoid polarization of the water and plating of the electrodes. One approach is to have the water in a circuit branch that is in series with some capacitance. In this sensor circuit, the water is in the branch containing the timing capacitance of a CMOS 555 timer, configured as a free-running oscillator with a 50% duty cycle. The sensor provides a capacitance that varies with water level.

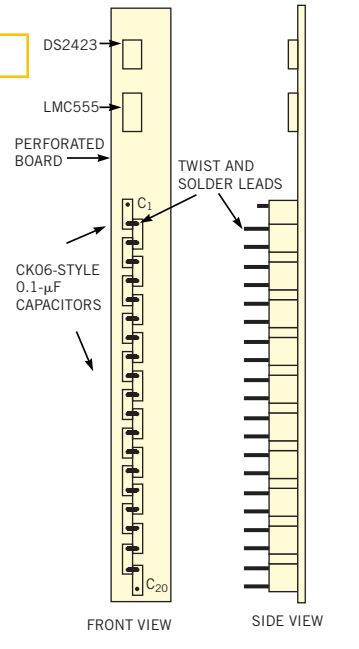
Figure 2 shows one method of fabri-

Figure 1



A simple 555-type timer and a counter form the heart of a water-level sensor.

Figure 2



A string of series-connected capacitors provides an indication of water level.

cating the sensor. You epoxy-bond a series string of N radial-leaded ceramic capacitors of equal value C underneath a pc board. Twist the leads of adjacent capacitors, solder them together, and trim them to serve as electrodes to contact the water. The outer shell of the sensor, a piece of 0.75-in. copper pipe, forms another electrode. If you place this assembly vertically in a vessel, the capacitance between the terminals of the sensor (the uppermost lead of C_1 and the outer shell) increases in steps as the water rises and covers more of the capacitors. The water effectively short-circuits the capacitors to the outer-shell electrode. Because the capacitors are in series, the total capacitance changes according to: $C_{TOTAL}=C/(N-n)$, where n is the number of capacitors with both leads covered by water. When you insert this expression in the equation for the 555 oscillation frequency, you obtain $f_{OSC}=(N-n)/1.4R_CC$. Note that the frequency changes linearly with water level.

This application uses 20 0.1-µF, CK06-style capacitors. The lead spacing of these capacitors is 0.2 in. These dimensions provide a measurement range of 4 in. with a resolution of 0.2 in. This design uses 1 MΩ for the oscillator timing resistor, R_C, because the timing resistance must be much larger than the impedance of the water to minimize timing error. (Measurements of several municipal and residential well-water samples revealed impedance values of approximately 5 kΩ in the frequency of 5 Hz to 1 kHz with 0.5-in. probe spacing.) Another reason for 1-MΩ oscillator timing is that the DS2423 counter has a maximum input frequency of approximately 2 kHz. Also, you must minimize the power stolen from the bus. Finally, you must maintain maximum isolation of the water from the bus in the event of a catastrophic bus fault.

With these values of capacitance and resistance, the sensor's output-frequency range is approximately 7 to 142 Hz in steps of approximately 7 Hz. In practice, you might read the counter at intervals of several seconds to several minutes to obtain an averaging effect. This sensor has found application for two summers in a residential swimming pool. Users noticed no change in performance from corrosion or plating of the electrodes.

REFERENCE

1. Awtrey, Dan, "Transmitting data and power over a one-wire bus," *Sensors*, February 1997.

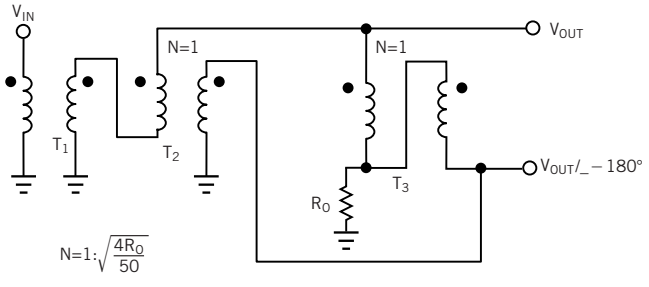
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Ideal transformers aid in balanced-line analysis

Alan Victor, IBM Microelectronics, Research Triangle Park, NC

TRANSMISSION-LINE transformers combined with appropriate resistor values are useful in hybrid applications over limited bandwidths. One such device, the 180° hybrid combiner, is useful in CAD analysis for verifying the performance of balanced and differential circuits. The completed transformer provides matched signal levels, 180° phase-shifted, and all ports at an impedance Z_0. You don't need transmission-line transformers in the construction of this hybrid for analysis. Instead, the circuit uses an ideal 1-to-1 transformer. Combined with the appropriate termination resistor and one additional transformer, an ideal differential-excitation source is available. Extensive bandwidth, balanced and equal impedance at all three ports, and infinite isolation provide an excellent circuit block to assist in analyzing differential and balanced circuits. The circuit block is borrowed from power-amplifier hybrids where it is useful for signal splitting and combining (**Figure 1**).

Figure 1



A simple transformer circuit aids in the analysis of differential and balanced circuits.

After you add a third transformer, T_1, to the two transformers you use, and with R_0 set at 25Ω, a 50Ω, three-port unit becomes available. If you replace the termination resistor, R_0, with a transmission line and a variable resistor, then a 180° phase-shifted variable impedance becomes available at the excitation ports. You then have a tool for studying balanced-line performance sensitivity to source-impedance variations. You can study high-speed, RF, and microwave using differential or balanced topologies using the arrangement in **Figure 1**. You can also turn the circuit around end-to-end and then use it for combining the output of the balanced circuit. Tests show that the circuit yields a perfectly flat frequency response from 1 to greater than 22 GHz and phase shifts of exactly 0° and 180° (depending on the port).

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

design ideas

Edited by Bill Travis and Anne Watson Swager

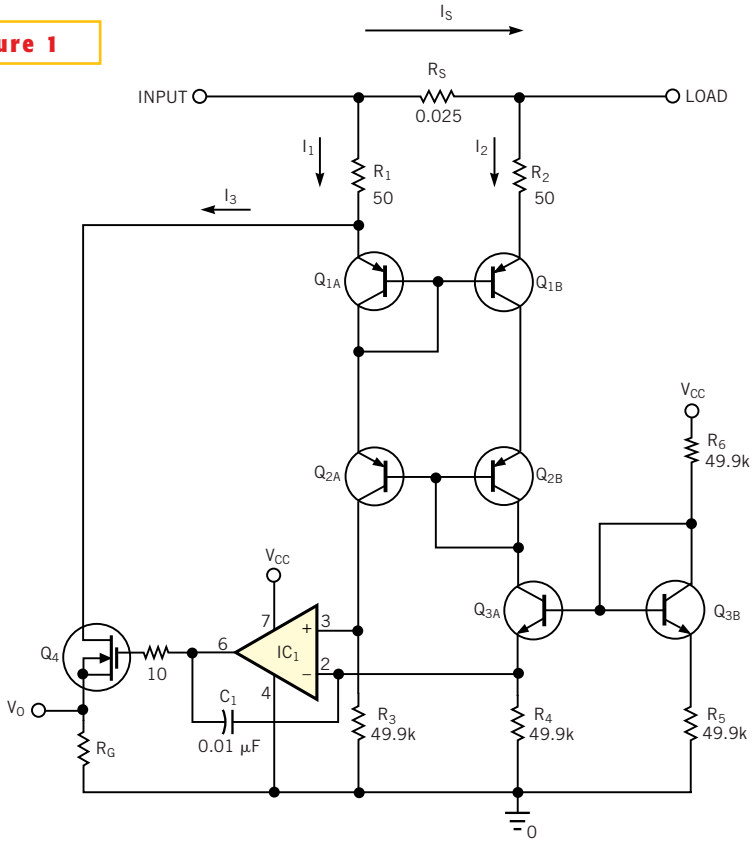
Circuit senses high-side current

Bob Bell and Jim Hill, On Semiconductor, Phoenix, AZ

THE ACCURATE, HIGH-SIDE, current-sense circuit in **Figure 1** does not use a dedicated, isolated supply voltage, as some schemes do. Only the selected transistors limit the common-mode range. The circuit measures the voltage across a small current-sense resistor, R_S. The operation of the circuit revolves around the high-side current mirror comprising Q_1 and Q_2. All the circuit components have one overall function: to make the collector currents equal in Q_1 and Q_2. The additional current mirror using Q_3 sets the values of the collector currents. The collector current is $(V_{CC}-0.7)/(R_5+R_6)\simeq 100\ \mu A$. You can best calculate the gain of the circuit by analyzing the loop formed by R_1, R_S, R_2, Q_{1B} (emitter base), and Q_{1A} (base emitter). In **Figure 1**, the currents are I_S, the high-side measurement current; I_1 and I_2, the mirror currents of Q_{1A} and Q_{1B}; and I_3, a branch current from the emitter of Q_{1A}.

When you sum the currents around the loop, $(I_S \cdot R_S) + (I_2 \cdot R_2) + VQ_{1B}(e-b) - ((I_1+I_3)\cdot R_1) = 0$. Because $I_1=I_2$, $R_1=R_2$, and the emitter-base voltages are equal, $I_3=I_S \cdot R_S/R_1$. Looking at the remaining circuitry, the op amp keeps the transistors' collector currents equal by controlling I_3 through Q_4. Therefore, the overall transfer function is $V_{OUT}=I_S \cdot R_S \cdot R_G/R_1$. For $R_G=1\ k\Omega$, the transfer function is $V_{OUT}=0.5 \cdot I_S$. The circuit can operate over a common-mode input range of approximately 10V to several hundred volts, limited by the selected transistors.

Figure 1



NOTES: IC_1 IS AN MC33202 RAIL-TO-RAIL OP AMP.
Q_1 AND Q_2 ARE SC-88 MBT3906 DUAL PNPs.
Q_3 COMPRISES MBT3904 SC-88 DUAL NPNs.
Q_4 IS A 2N7002 SOT-23 FET.

This circuit measures high-side currents without the need for auxiliary power supplies.

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

Adjustable filter provides lowpass response

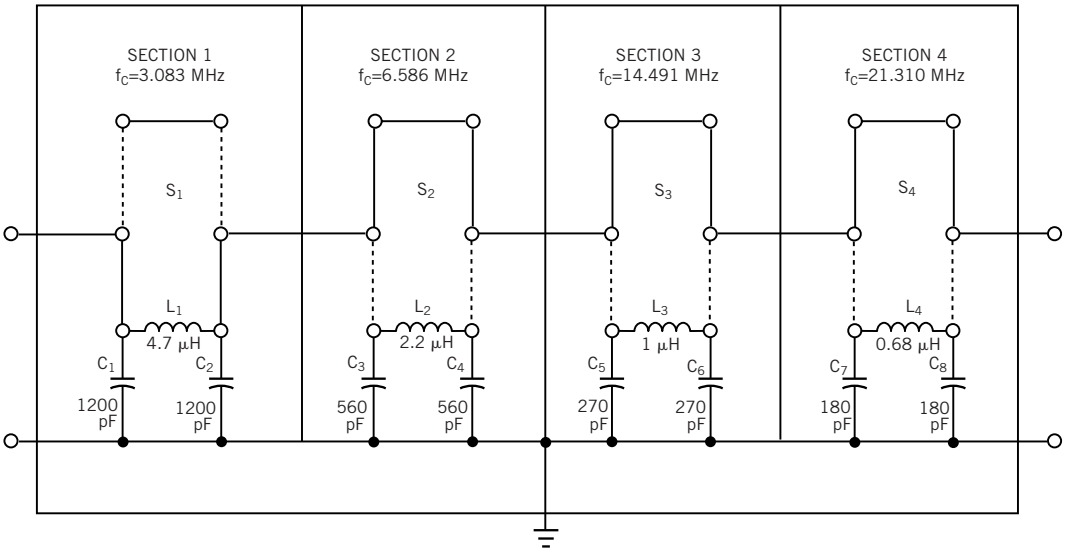
Richard Kurzrok, Queens Village, NY

You can configure simple lowpass filters as pi sections with nominal three-pole, 0.1-dB Chebyshev response to provide a moderate amount of stopband selectivity. You can put four of these filters into one enclosure and then select discrete-filtering steps by using toggle switches. Manufacturers of commercially available stepped attenuators and adjustable baseband equalizers commonly use this technique (**Reference 1**). In an adjustable lowpass filter, each filter section uses commonly available components (**Figure 1**). This example uses filter-section cutoff frequencies for standard

TABLE 1–MEASURED AMPLITUDE RESPONSE OF ADJUSTABLE LOWPASS FILTER

Frequency (MHz)	Box insertion loss (dB)	Filter 1 insertion loss (dB)	Filter 2 insertion loss (dB)	Filter 3 insertion loss (dB)	Filter 4 insertion loss (dB)
1	<0.1	0.1	<0.1	<0.1	<0.1
2	<0.1	0.3	0.1	0.1	0.1
2.5	<0.1	0.6	0.1	0.1	0.1
2.9	<0.1	1.7	0.1	0.1	0.1
3.1	<0.1	2.5	0.15	0.1	0.1
3.3	<0.1	3.3	0.15	0.1	0.1
4	<0.1	7.3	0.2	0.1	0.1
5	<0.1	13	0.45	0.2	0.1
6	<0.1	17.9	1.3	0.25	0.15
6.5	<0.1	20	2.1	0.25	0.2
7	<0.1	21.8	3.1	0.25	0.2
9	<0.1	28.2	8.3	0.4	0.2
12	<0.1	33.4	15.3	1.3	0.25
14	<0.1	34.8	19.4	2.9	0.4
17	<0.1	35.2	24.5	6.4	0.9
20	<0.1	35.3	26.4	10.2	2.1
23	<0.1	>35	28.8	14.5	4
30	0.1	>35	31.7	19.2	9.9
50	0.2	>34	>34	23.5	20
100	0.5	>28	>28	>24	>24

Figure 1



NOTE: ALL SWITCHES ARE DOUBLE-POLE, DOUBLE-THROW TOGGLE SWITCHES.

A switchable lowpass filter provides a choice of four distinct cutoff frequencies.

inductors and capacitors without the need for any extra components in series or parallel. Fixed inductors are Coilcraft 90 series axial-lead chokes with ±10% tolerance. Fixed capacitors are polypropylene units, available from any distributor, with ±2% tolerance for the 1200-pF devices and ±5% tolerance for the other values.

The adjustable lowpass filter is in a 3.625-in.-long×1.5-in.-wide×1.0625-in.-high Bud CU-123 die-cast aluminum box with input and output BNCs. Miniature toggle switches for the individual filter sections are accessible at the enclosure's exterior. Internal ground returns use solder lugs. The four filter sections have 50Ω characteristic impedance and nominal 3-dB cutoff, from left to right in **Figure 1**, of 3.083, 6.586, 14.491, and 21.310 MHz. **Table 1** shows the measured amplitude response for the box alone and for the four individual filter sections. The low-cost, adjustable lowpass filter delivers reasonable performance. As the frequency approaches 100 MHz, the transmission performance of the enclosure deteriorates with all filter sections switched to the off position. The interconnections between switched sections use available bus wire without any precautions to minimize parasitic circuit elements. Stray series inductance, estimated at approximately 55 nH, arises from the 3.5-in. physical length of the enclosure, plus 2 in. for the four switches.

REFERENCE

1. Kurzrok, Richard, "Adjustable-amplitude equalizer provides small discrete steps," *Electronic Design*, May 31, 1999, pg 76.

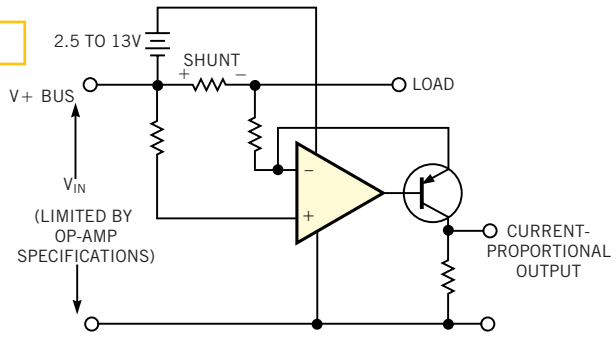
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Monitor high-side current without an external supply

Vijay Damle, Digitronics, Pune, India

TYPICAL HIGH-SIDE current-sensing circuits require a dc source that is 2.5 to 13V greater than the V+ high-bus voltage (**Figure 1**). Generating this supply is painful in many situations. For example, in power supplies for TV transmitters, the main SMPS (switch-mode power supply) output supplies the power amplifier, and a series switching regulator steps down the main SMPS output to drive the exciter. The system must remotely display the currents of both of these supply outputs, with 0 to 50A corresponding to 0 to 5V referred to sense V−. Because of the presence of a series switch, the V− lines of both outputs are common. Thus, you cannot use shunts in the V− line and amplify. Shunts are necessary on the positive bus of both the outputs. The main output supplies 30 to 45V at 30A, and the exciter supply outputs 22 to 26V at 10A. You need costly Hall-effect sensors to achieve the proportional output, though isolation is not required.

Figure 1



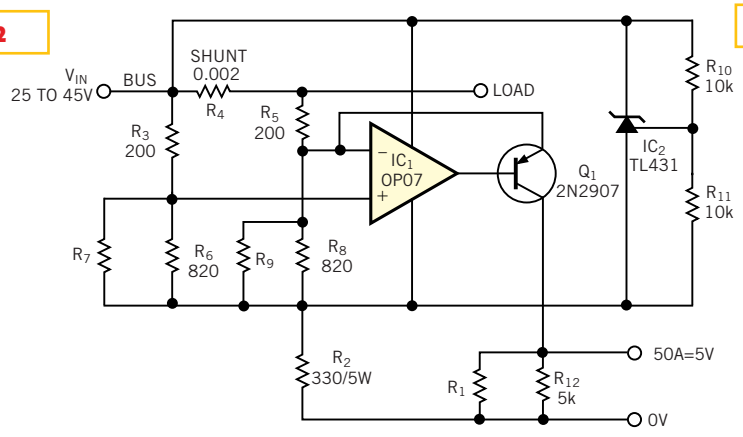
Typical high-side current-sensing circuits require a dc source that is 2.5 to 13V greater than the V+ high-bus voltage; generating this supply can be difficult.

An alternative approach for this application takes advantage of low-offset op-amp characteristics to design a circuit that works with a wide voltage range and needs no other supply. The V+ and inverting and noninverting terminals of the OP07 op amp need a minimum of approximately 2 to 2.5V to function properly. Thus, you can pull the op amp's inputs by more than 2.5V below the positive-supply connection and tie the op amp's V+ pin to shunt V+ (**Figure 2**).

In the circuit, IC_2 with R_{10} and R_{11} generate a 15V output. The R_3 and R_6 pair and R_5 and R_8 pair form dividers such that the op amp's inverting and noninverting inputs are approximately 3V less than the V+ supply of the op amp. You can use R_7 and R_9 to trim the offset to avoid the need for potentiometers. Op amp IC_1 and Q_1 generate a current that is proportional to the shunt voltage. R_{12} generates a voltage that is proportional to the drop across shunt R_4. R_1 trims the gain.

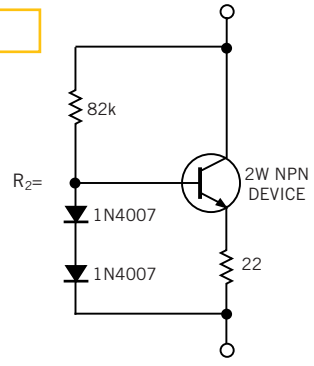
If you use this circuit at less than 25V, then you can delete IC_2, R_{10}, and R_{11}. You should also ground IC_1's V− pin by shorting R_2, and you can replace R_2 with a constant-current source to reduce the power due to bus-voltage variation (**Figure 3**.)

Figure 2



A modified current-monitoring circuit pulls the op-amp inputs below the positive supply voltage.

Figure 3



At voltages less than 25V, you can replace R_2 with a constant-current source.

This circuit was tested for 0 to +55°C, and it maintained proportional output within ±1% for a bus-voltage variation of 25 to 45V over this temperature range.

This approach has many advantages. An external supply is unnecessary. The circuit is suitable for bus voltages of 5 to 60V with component changes. Other circuits have limitations due to op-amp absolute-maximum voltage ratings. The circuit acts as the minimum load that SMPS outputs normally require, which eliminates or reduces high-wattage resistance across the output. You can easily scale the circuit for different proportional outputs. You can add a buffer amplifier to reduce the output impedance, and the buffer can derive its supply across R_2, which increases its operating supply range by 15V or more. One limitation is that, in the case of a short circuit, the current-proportional output drops to zero.

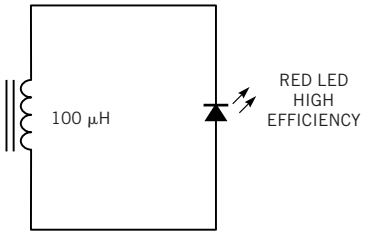
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Noncontact device tests power supplies

Alberto Ricci Bitti, Eptar, Imola, Italy

THE PROBELIKE DEVICE in **Figure 1** comes in handy as a quick go-no-go test for step-down power supplies. You can build it using a very bright surface-mount LED and an inductor of the same type as in the power supply, which in this case is 100 µH. Placing this probe close to a working step-down power-supply coil lights the LED. The probe lights when the distance from the step-down coil is as much as 1 cm, making the probe capable of testing even plastic-encased or epoxy-filled power supplies. Industrial engineers will particularly appreciate the capability of not touching the circuit, which is also a useful feature when testing boards that operate without insulation from the mains.

Figure 1



An LED and an inductor make a simple probe for testing power-on of the supply.

For optimum performance, use a very bright-red LED. Other colors feature greater forward voltages, which reduce the sensitivity. You are not restricted to surface-mount LEDs, although this type helps by keeping the probe small and rugged.

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

Single chip detects optical interruptions

Frederick M Baumgartner, FM Broadcast Services, Parker, CO

Setting up a light beam and detector to count objects on a conveyer belt, sense security intrusions, or drive a tachometer is simple. However, the task is no longer trivial if you add ambient light or multiple beams, limit optical power, or extend the distance of the light beam more than a few inches. You can use optical lenses and filters and high-power optical sources on the light-path side to improve performance. On the electronic side, servo-bias control of the detector and electronic modulation and filtering of the light beam can add considerable range. The circuit in **Figure 1**, which you can use with these performance improvements, economically provides a minimal-parts-count circuit with negligible power requirements to achieve approximately a foot of useful range even under varying ambient-light conditions.

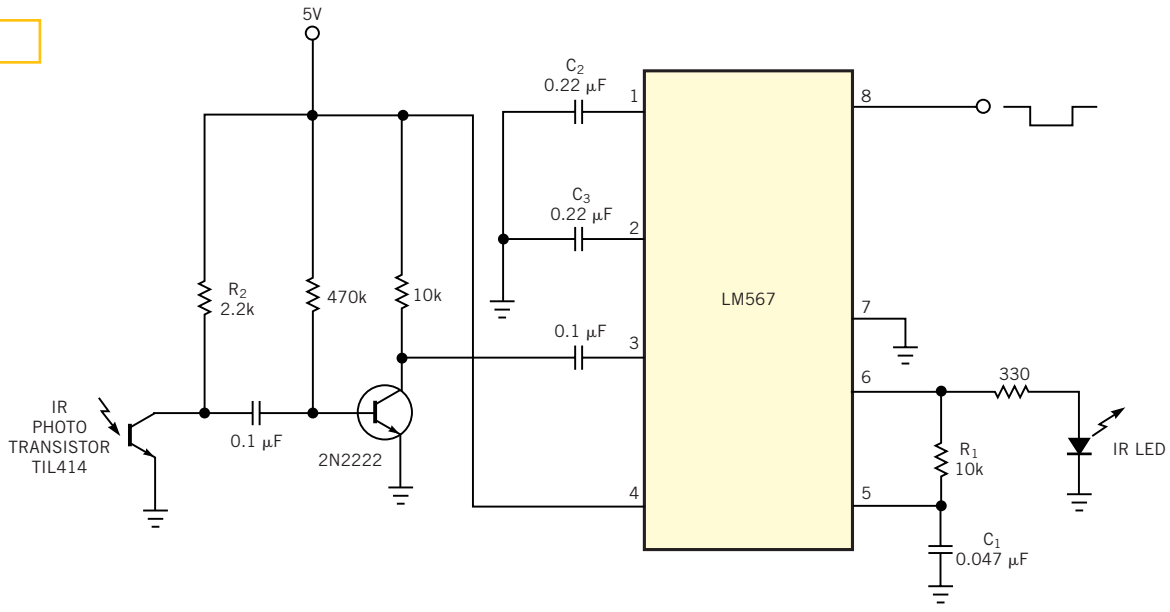
The venerable LM567 PLL is the only IC in the circuit. The 567's oscillator directly drives an infrared LED on the optical-transmitter end. When the pulsed light returns to the IR phototransistor, a single-stage 2N2222 transistor amplifies the resultant signal to drive Pin 3 of the LM567. Thus, the circuit essentially directs the PLL to lock to itself, which makes Pin 8 go low. The values of R_1 and C_1 provide operation of approximately 3 kHz, and the filters set by C_2 and C_3 provide a clean output from the 567. Operation from 2 to 5 kHz works best. Lower frequencies require more conditioning and thus larger and more critical values of C_2 and C_3, resulting in longer response times and possible jitter. Higher frequencies result in lower efficiencies for the cheap LED and phototransistor. However, tachometers may require higher frequencies. IR components are unnecessary. Two same-color LEDs (one for the photo detector) also work to a degree.

Ambient light or another beam breaker's IR light doesn't false-trigger the circuit unless significant near-frequency light content exists. However, ambient light can swamp the detector, so you may need to adjust the R_2 bias for your application. Of course, using a self-adjusting module with IR filters can easily increase the range by two orders of magnitude.

One interesting variation of the circuit is to use two or more devices on the same frequency, forming a ring. All devices lock, and both ends detect a break in any beam or a modulation of the frequency of any device for communication.

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

Figure 1



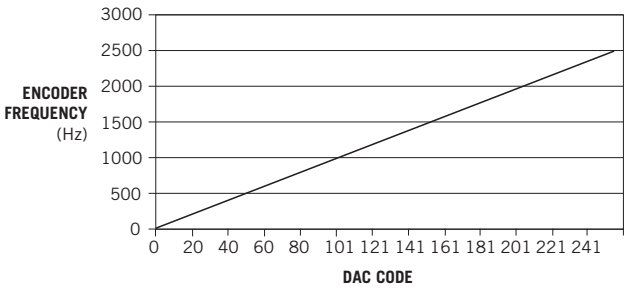
A light-beam-breaker detector uses just one IC and a few external components.

Programmable source powers dc micromotors

VK Dubey, JP Rao, and P Saxena, Centre for Advanced Technology, Indore, India

THE CIRCUIT in **Figure 1** is a simple, economic, compact, and tricky way of using the LM723 as a programmable voltage source to drive dc micromotors. Because of the µPs' accurate positioning and control, these motors are useful in applications such as optical mounts and flexible shaft control, which take advantage of the higher speed and fast movement of servo controls compared with stepper motors. These designs require a stable, programmable dc-voltage source.

Figure 2



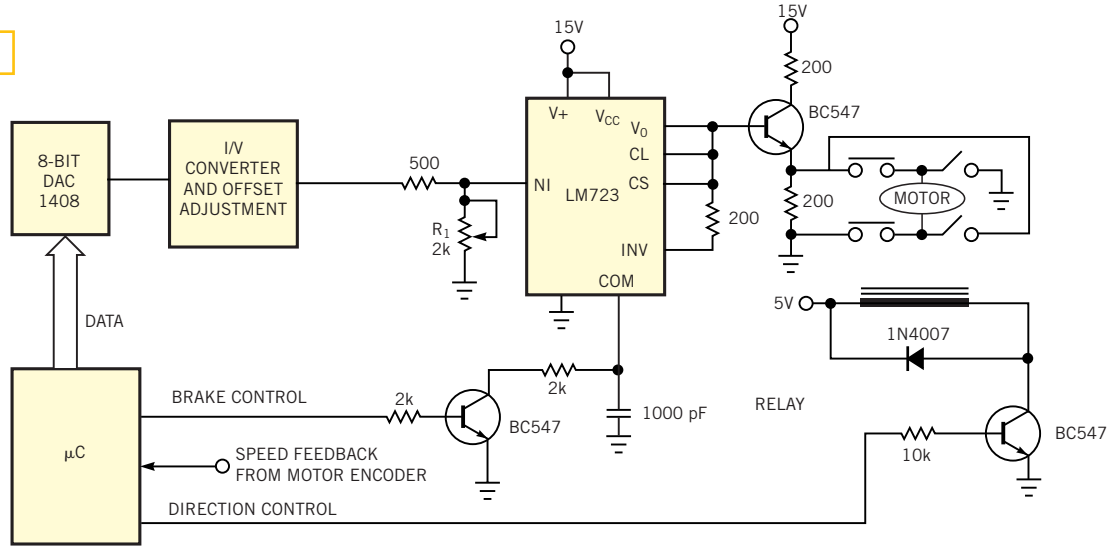
The DAC-code versus encoder-frequency, or speed, curve is linear.

The LM723 is a fixed linear regulator, but this application configures the regulator as a programmable voltage source. You can set the output to a value of 200 mV to 6V. The output, an emitter-follower type, provides low output impedance. The circuit limits the maximum output current to the load, or the motor, at 75 mA. The output of an 8-bit DAC and a current/voltage converter provide a variable reference voltage. At the noninverting input of the LM723, you need to adjust the value of R_1 so that the maximum reference voltage does not exceed 8.5V. Because the reference voltage comes from an external source, the circuit doesn't use the internal voltage reference of the LM723. The circuit also incorporates short-circuit current limiting and remote shutdown. Varying the output voltage changes the speed of the motor that connects across the output.

You adjust the minimum output voltage of 200 mV by offsetting the DAC output with zero data, and successive DAC input codes increase the voltage-source output to 6V. You can use a single-chip µC for controlling the speed through the DAC, the direction, and the brake. The no-load maximum speed is 15,100 rpm. By attaching a reduction gear-head with a ratio of 529-to-1, the maximum frequency from the magnetic encoder in response to maximum speed is 2.8 kHz. The circuit feeds back this signal to the µC to measure the speed. The linearity of the voltage source is good over a voltage, temperature, and speed range (**Figure 2**). With only slight modifications in component values and ratings, you can use this same LM723 configuration in other similar applications for higher output voltages.

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

Figure 1



NOTES:
RELAY=TWO-CHANGEOVER REED RELAY.
MOTOR=FAULHABER DC MICROMOTOR TYPE 1219-006 G, MICRO-ENCODER TYPE 30B.

Configuring an LM723 as a programmable voltage source provides a variable dc source for driving dc micromotors.

Optocoupler extends high-side current sensor to 1 kV

Roger Griswold, Maxim Integrated Products, Sunnyvale, CA

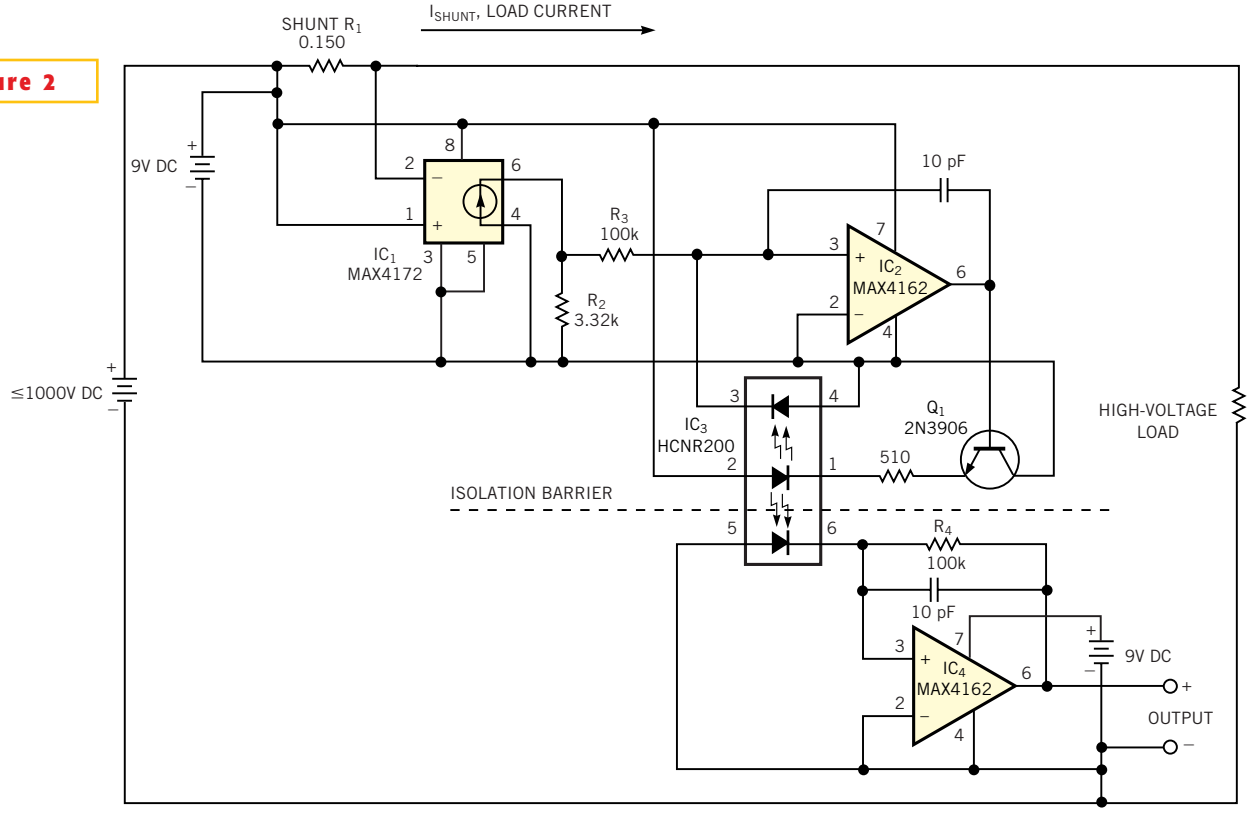
THE TASK OF SENSING dc current at high voltage is often problematic. Most high-side current-sensing ICs available off the shelf are good only to 30 or 40V. Combining an optocoupler with such an IC yields a sensing circuit in which the only limitation of the high-side voltage is the optocoupler's standoff voltage (**Figure 1**).

A precision, high-side current-sense amplifier, IC_1, and a high-linearity analog optocoupler, IC_3, extend the high-side working voltage to 1000V dc. IC_3 supports a continuous 1000V dc. Its UL rating is 500V rms for 1 minute, and its transient surge rating is 8000V dc for 10 seconds. You should follow all proper safety precautions when working with high voltage.

The circuit has a floating section and a grounded section, each requiring a local low-voltage supply. The floating section detects load current and drives the high-voltage side of the optocoupler. The grounded section monitors the optocoupler's low-voltage side and outputs a voltage proportional to the high-side load current. IC_3 has a feedback photodiode on the high-voltage side that virtually eliminates the LED's nonlinearity and drift characteristics. In addition, IC_3's two closely matched photodiodes ensure a linear transfer function across the isolation barrier.

During operation, the load current passes through shunt R_1 and produces a small voltage. IC_1 monitors this voltage and outputs a proportional current of 10 mA/V. This proportional output current routes through R_2, which produces a voltage proportional to the main load current. The rest of the circuit generates a copy of the voltage across R_2 but on the low-voltage side of the optocoupler. IC_2 monitors the voltage across R_2 and drives the optocoupler's LED via Q_1. The LED generates light that impinges equally on the high- and low-side photodiodes. IC_4 monitors the low-side photodiode and outputs a voltage proportional to the high-side load current. A graph shows the

Figure 2



The ground-referenced output voltage, $V_{OUT}=I_{SHUNT}$ (4.80V/A), is proportional to the high-side load current. As configured, the circuit measures load currents to 1A.

output voltage as a function of shunt current (**Figure 2**).

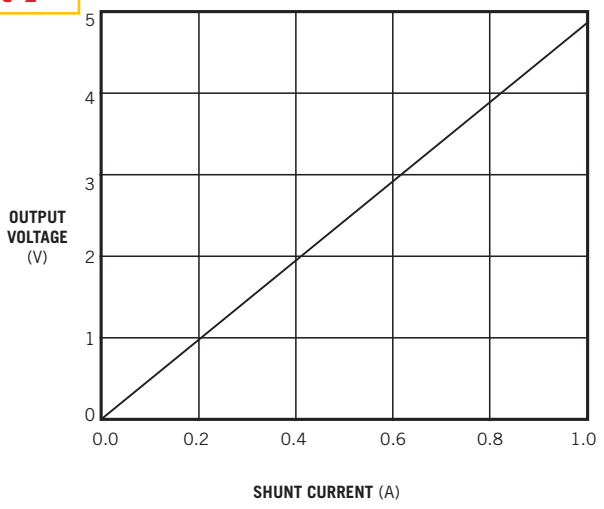
If R_3 and R_4 are equal, the overall transfer function is:

$$\frac{V_{OUT}}{I_{SHUNT}} = 0.01 \bullet R_1 \bullet R_2.$$

Three parameters let you modify the circuit to monitor other maximum load currents and output a different voltage range. The maximum IC_1 output current is 1.5 mA, so the maximum allowed shunt voltage is 150 mV. Also, the maximum allowed photodiode current is 50 µA. Choose an R_1 value that produces 150 mV at the maximum load current that the circuit monitors. Then, choose an R_2 value that produces the desired corresponding maximum output voltage at 1.5 mA. Match R_3 and R_4, and choose a value that allows less than 50 µA through the photodiode at the maximum desired output voltage, or

$$R_3 \geq \frac{V_{OUT\_MAX}}{50 \times 10^{-6}}.$$

Figure 2



The output voltage versus shunt current is linear.

The circuit output then faithfully reproduces the voltage across R_2. The MAX4162 op amp is a good choice for this circuit because of its input-bias current of 1 pA, its rail-to-rail input and output swings, and its ability to operate from one 9V battery. With $R_1 = 150$ mΩ and $R_2 = 3.32$ kΩ, the output voltage for $I_{SHUNT} = 1$A is 4.80V using the given transfer function. Experimental results at $I_{SHUNT} = 1.00$A give $V_{OUT} = 4.84$V with an error less than 1%.

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

design ideas

Edited by Bill Travis and Anne Watson Swager

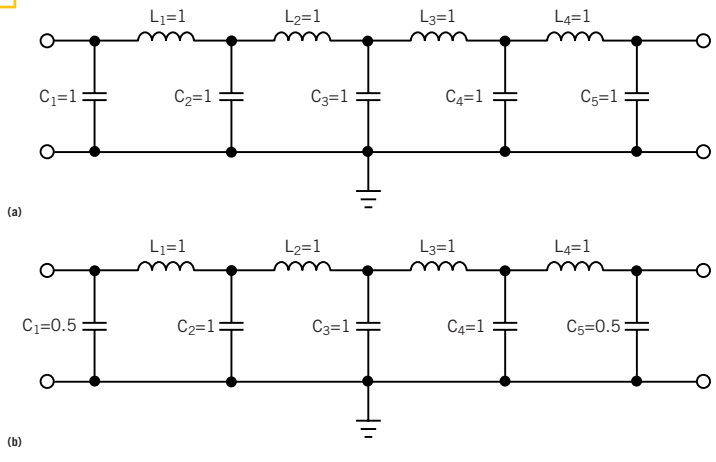
Equal-element filter improves passband performance

Richard M Kurzrok, RMK Consultants, Queens Village, NY

Designers originally conceived equal-element filters as all-pole microwave bandpass filters that provide minimum center-frequency insertion losses for specific values of resonator-unloaded Q (**Reference 1**). All resonators of the equal-element bandpass filter operate at the same loaded Q. For LC filters, the equal-element filter has another advantage. In the lowpass prototype, all inductors have the same value, and all capacitors have the same value. This minimum number of circuit elements provides design simplicity and reduces filter cost.

However, the equal-element filter's response shape has one severe shortcoming. Passband amplitude ripples, due to reflection, are unacceptable for some applications. In minimum-phase-shift filter circuits, group-delay ripples that preclude equalization accompany the amplitude ripples. At microwave frequencies, modifying the central resonator of a five-pole bandpass filter leads to improved performance (**Reference 2**).

Figure 1



The equal-element lowpass filter can feature unacceptable passband ripple for some applications (a). A modified filter realizes substantial improvement in passband performance with some reduction in stopband selectivity (b).

Figure 1a shows the schematic of a nine-pole, equal-element, lowpass-filter prototype. **Figure 1b** shows a comparable schematic of a modified lowpass-filter prototype. By altering the filter input and output capacitors, the modified filter realizes substantial improvement in passband performance with some reduction in stopband selectivity. In the modified equal-element design, all filter inductors are still equal in value, and you need only two capacitor values in a convenient 2-to-1 ratio. **Table 1** shows comparative theoretical amplitude responses for the nine-pole, equal-element lowpass filter and a nine-pole, modified, equal-element, lowpass filter with inductor-unloaded Qs of 100.

The reference frequency, at normalized frequency x=1.0, is not the 3-dB cutoff frequency for the filters in **Figure 1**. The 3-dB cutoff frequency occurs close to x=1.9. This feature differs from Butterworth and Chebyshev filters, for which x can equal 1.0 at 3-dB cutoff frequencies. You use this normalization for equal-element and modified equal-element designs to calculate the values of the circuit elements.

A nine-pole, modified, equal-element lowpass filter was designed at a reference frequency F_R of 4.681 MHz, for which

TABLE 1–THEORETICAL AMPLITUDE RESPONSES FOR NINE-POLE, EQUAL-ELEMENT AND MODIFIED, EQUAL-ELEMENT, LOWPASS FILTER

Normalized frequency	Frequency (MHz)	Insertion loss of equal-element lowpass filter (dB)	Insertion loss of modified equal-element lowpass filter (dB)
0	0	0.172	0.172
0.1	0.468	0.185	0.172
0.2	0.936	0.202	0.174
0.3	1.404	0.201	0.176
0.4	1.872	0.212	0.178
0.5	2.341	0.315	0.182
0.6	2.809	0.532	0.184
0.7	3.277	0.739	0.186
0.8	3.745	0.74	0.192
0.9	4.213	0.475	0.215
1	4.681	0.259	0.27
1.1	5.149	0.707	0.345
1.2	5.617	1.777	0.386
1.3	6.085	2.546	0.337
1.4	6.553	2.167	0.258
1.5	7.022	0.664	0.462
1.6	7.49	1.849	1.191
1.7	7.958	5.52	1.84
1.8	8.426	5.91	1.124
1.9	8.894	3.574	2.887
2	9.362	19.255	12.533
2.1	9.83	28.937	20.678
2.2	10.298	36.285	27.288
2.3	10.766	42.426	32.953
2.4	11.234	47.794	37.977
2.5	11.703	52.612	42.53
2.6	12.171	57.011	46.715
2.7	12.639	61.075	50.604

Note: Data is for inductor-unloaded Qs of 100.

x=1. For 50Ω input and output impedances Z_0, you calculate the normalizing inductance, L_0, and normalizing capacitance, C_0, as follows:

$$L_0 = \frac{Z_0}{2\pi \bullet F_R} = 1.7\ \mu H;$$

$$C_0 = \frac{1\times 10^{-6}}{2\pi \bullet F_R Z_0} = 680\ pF.$$

You then use these values of L_0 and C_0 to denormalize the filter to actual circuit-element values. Filter inductors L_1, L_2, L_3, and L_4 are equal to L_0=1.7 µH. Interior filter capacitors C_2, C_3, and C_4 are equal to C_0=680 pF. The filter input and output capacitors, C_1 and C_5, are equal to $0.5\times C_0$=340 pF. In the actual filter, the input and output capacitors are standard 330-pF values. The nine-pole, modified, equal-element, lowpass filter was constructed in a die-cast aluminum box with BNCs. The filter circuit was fabricated using vector board. All capacitors were 5%-tolerance polypropylene units. All inductors used 18 turns of number 26 magnet wire on Micro Metals' T37-2 toroids. **Table 2** shows the measured amplitude-response data. The measured data provides reasonable correlation between theory and experiment and shows substantial improvement in amplitude response over most of the filter passband with some degradation in stopband performance.

TABLE 2–MEASURED AMPLITUDE RESPONSE FOR NINE-POLE, MODIFIED, EQUAL-ELEMENT, LOWPASS FILTER

Frequency (MHz)	Insertion loss (dB)
2	0.1
4	0.15
5	0.2
5.5	0.25
6	0.3
6.5	0.4
7	0.6
7.25	0.85
7.5	1.15
7.75	1.4
8	1.5
8.1	1.1
8.2	0.8
8.3	1.5
8.4	1.1
8.5	0.8
8.6	1.25
8.7	0.75
8.8	1.25
8.9	2.4
9	4
9.25	9.2
9.5	13.7
10	21.9
11	34.6
12 to 30	Greater than 45

You can transform the modified, equal-element, lowpass prototype into useful highpass and bandpass filters with similar design features.

REFERENCES

1. Taub, JJ, "Design of Minimum Loss Band-Pass Filters," *Microwave Journal*, Volume 6, pg 67, November 1963.

2. Bawer, R, and G Kefalas, "A Modified Equal-Element Band-Pass Filter," *IRE Trans MTT*, Volume MTT-5, pg 175, July 1957.

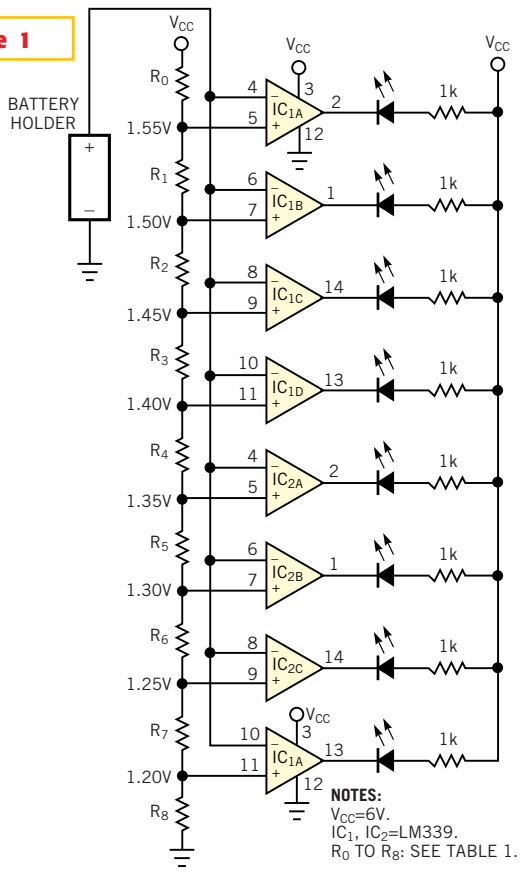
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Test batteries without a voltmeter, part 2

Harry Gibbens Jr, PowerStream Technology, Orem, UT

From a high-volume-production point of view of the Design Idea "Test batteries without a voltmeter" (*EDN*, Nov 9, 2000, pg 167), it is a time-consuming and laborious task to tweak the large number of potentiometers on each of the comparator input references. An alternative to this onerous adjustment chore is to replace all potentiometers with 1%-tolerance fixed resistors (**Figure 1**). Before calculating each voltage-reference resistance value, you should use a reasonable selected total resistance value, R_{TOT}, ranging from 100 kΩ to 1 MΩ. You can usually obtain each of the resistance-divider values from off-the-shelf fixed resistors. After calculating all the resistance-divider values (with R_{TOT} = 100 kΩ), if the closest reasonable stock values are unavailable, just increase the value of R_{TOT}. For the example in **Figure 1**, you calculate the values using a spreadsheet to obtain quick results.

Figure 1



Fixed, 1%-tolerance resistors eliminate the need to trim potentiometers in this battery-testing circuit.

This example uses R_{TOT} = 182 kΩ. All the calculations use Kirchoff's law:

$$V_X = V_{TOT}\left(\frac{R_X}{R_1 + R_2}\right).$$

Rearranging the terms in the formula, you determine the resistance value, R_X:

$$R_X = R_{TOT}\left(\frac{V_X}{V_{TOT}}\right),$$

where V_X is each comparator's reference voltage, V_{REFX}; $V_{TOT} = V_{CC}$; and R_{TOT} is the manually selected value. The first step is to calculate the "Formulated R" in **Table 1**, using the rearranged formula. An example follows:

$$R_{FORM1} = 182\,k\Omega\left(\frac{1.55V}{6V}\right)$$
$$= 182\,k\Omega(0.2583)$$
$$= 47.01\,k\Omega.$$

Next, calculate the first resistance value, R_0, by using the formula

$$R_X = R_{TOT} - R_{TOT}\left(\frac{V_X}{V_{TOT}}\right) = R_{TOT} - R_{FORMX}$$

$$R_0 = 182\,k\Omega - 182\,k\Omega\left(\frac{1.55V}{6V}\right)$$
$$= 182\,k\Omega - 182\,k\Omega(0.2583)$$
$$= 182\,k\Omega - 47.01\,k\Omega$$
$$= 134.99\,k\Omega.$$

Calculate the rest of the resistor values, R_1 through R_7, by using the formula

$$R_{X+1} = R_{TOT}\left(\frac{V_X}{V_{TOT}}\right) - R_{TOT}\left(\frac{V_{X+1}}{V_{TOT}}\right) =$$
$$R_{FORMX} - R_{FORMX+1}.$$

For example, for R_1:

$$R_1 = 100\,k\Omega\left(\frac{1.55V}{6V}\right) - 100\,k\Omega\left(\frac{1.5V}{6V}\right)$$
$$= 1.51\,k\Omega.$$

Finally, use the rearranged formula to

TABLE 1–CALCULATED VALUES FOR BATTERY TESTER

Reference voltage	Voltage (V)		Formulated R kΩ	RX	Calculated R kΩ	Fixed R kΩ
V_{TOT}	6			R_{TOT}	182	
V_{REF1}	1.55	0.26	47.01	R_0	134.99	134.5
V_{REF2}	1.5	0.25	45.5	R_1	1.51	1.5
V_{REF3}	1.45	0.24	43.99	R_2	1.51	1.5
V_{REF4}	1.4	0.23	42.46	R_3	1.53	1.54
V_{REF5}	1.35	0.23	40.95	R_4	1.51	1.5
V_{REF6}	1.3	0.22	39.44	R_5	1.51	1.5
V_{REF7}	1.25	0.21	37.91	R_6	1.53	1.54
V_{REF8}	1.2	0.2	36.4	R_7	1.51	1.5
				R_8	36.4	36.4

find the last resistance value, R_8.

$$R_8 = 182\ k\Omega\left(\frac{1.2V}{6V}\right) = R_{FORM8}$$
$$= 36.4\ k\Omega.$$

In **Table 1**, you match the rounded-off calculated resistance value (Calculated R) to the nearest available fixed-resistor value (Fixed R) as shown in the last two columns. Note that R_0 consists of two series-connected resistors. You can download the table of 1%-tolerance resistor values from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the .gif file for Design Idea #2654.

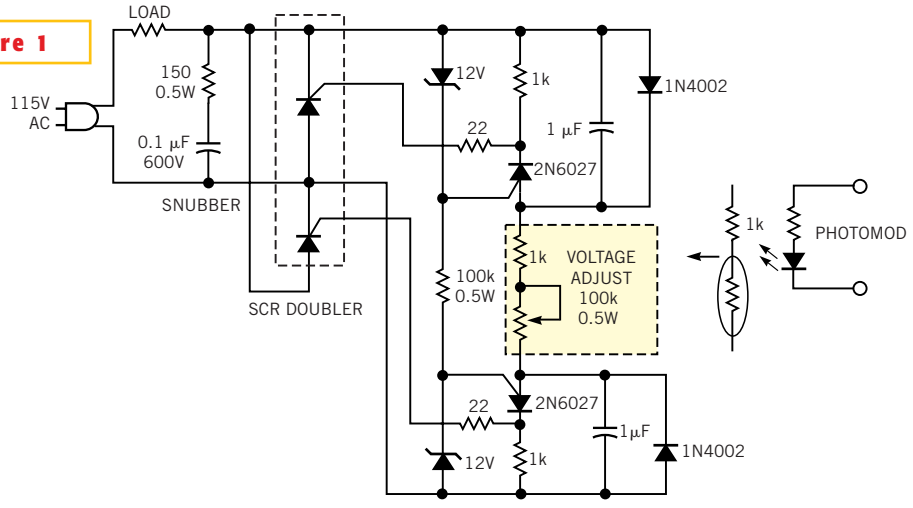
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

SCR phase control yields solid-state switch

James Keith, York, PA

SCRs (silicon-controlled rectifiers), or thyristors, have higher current and voltage ratings, lower conduction losses, and more robustness than triacs. For these reasons, SCRs are better suited to high-power applications. For example, you can use two SCRs to configure a 100 or 200A, 460V control circuit. **Table 1** lists some of the advantages of SCRs over triacs. The main challenge is driving the SCRs: You now have two gates, rather than one, to drive. Furthermore, the gates are referenced to opposite polarities with a significant voltage difference. The circuit in **Figure 1** solves the problem with two PUTs (programmable unijunction transistors), one connected to each SCR. Performance is good because the circuit does not "fold back" as inexpensive triac phase controls (dimmers) do. The PUT fires when the timing-capacitor voltage exceeds the PUT reference voltage by one diode drop and dumps the capacitor's charge into the SCR gate.

Figure 1



This SCR phase-control circuit uses one potentiometer and no pulse transformers.

You adjust the phase delay by varying the charge rate of the capacitor via the potentiometer. You can enhance the balance by matching both the zener-diode voltages and the values of the charging capacitance. You can obtain 230V operation by using 1.5- to 2-µF timing capacitors. You can achieve 460V-ac operation by using 1200V SCRs; 3-µF timing capacitors; and a 100-kΩ, 25W potentiometer. Using the circuit as a solid-state switch is practical; simply replace the voltage-adjust block in the broken lines in **Figure 1** with a *photomod*. A photomod is a unique type of photocoupler that has an LED or incandescent-light source and a cadmium-sulphide photocell. Unique properties of these photocells are their high dark resistance and high voltage ratings.

TABLE 1—SCR-VERSUS-TRIAC COMPARISON

Device	SCR	Triac
Current rating	>>50A	<40A
Conduction voltage drop	~1V	~1.5V
Junction temperature	125°C	115°C
Thermal resistance	Low	Medium
Maxium voltage	1400V	600V
Surge current	High	Limited
Robustness	High	Low
Package	Doubler or "hockey puck"	¼-in. stud
Isolation	Isolated	Nonisolated
Number required	Two	One

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

Circuit facilitates video fading

JM Terrade, Clermont-Ferrand, France

When you're copying videotapes, it's sometimes desirable to suppress some passages. Using the pause control of the recorder does not yield satisfactory results. Another method produces better results (**Figure 1**). The video source connects to the video-in plug, and the recorder connects to the video-out plug. Turning potentiometer P_1 adjusts the image brightness from normal video to a black image. With the P_2 potentiometer ganged to P_1, the sound also varies accordingly. The objectives in building this circuit are to use inexpensive, readily available components and to obtain batteryless operation. The video signal follows two paths (**Figure 2**). In the first path, the signal undergoes amplification by a factor of two and connects to one end of a potentiometer. In the second path, the synchronization pulse, separated from the input signal, connects to the other end of the potentiometer. The wiper of the potentiometer connects to the second video amplifier, which provides the video output.

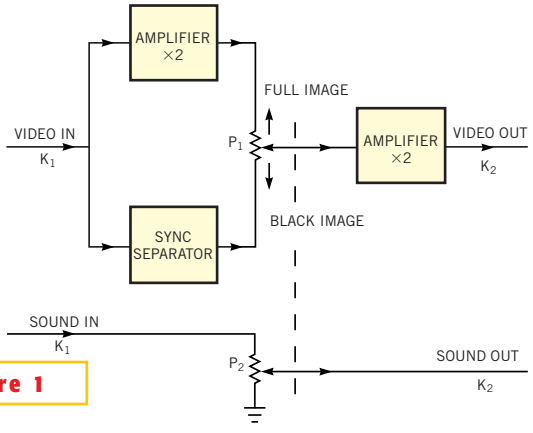
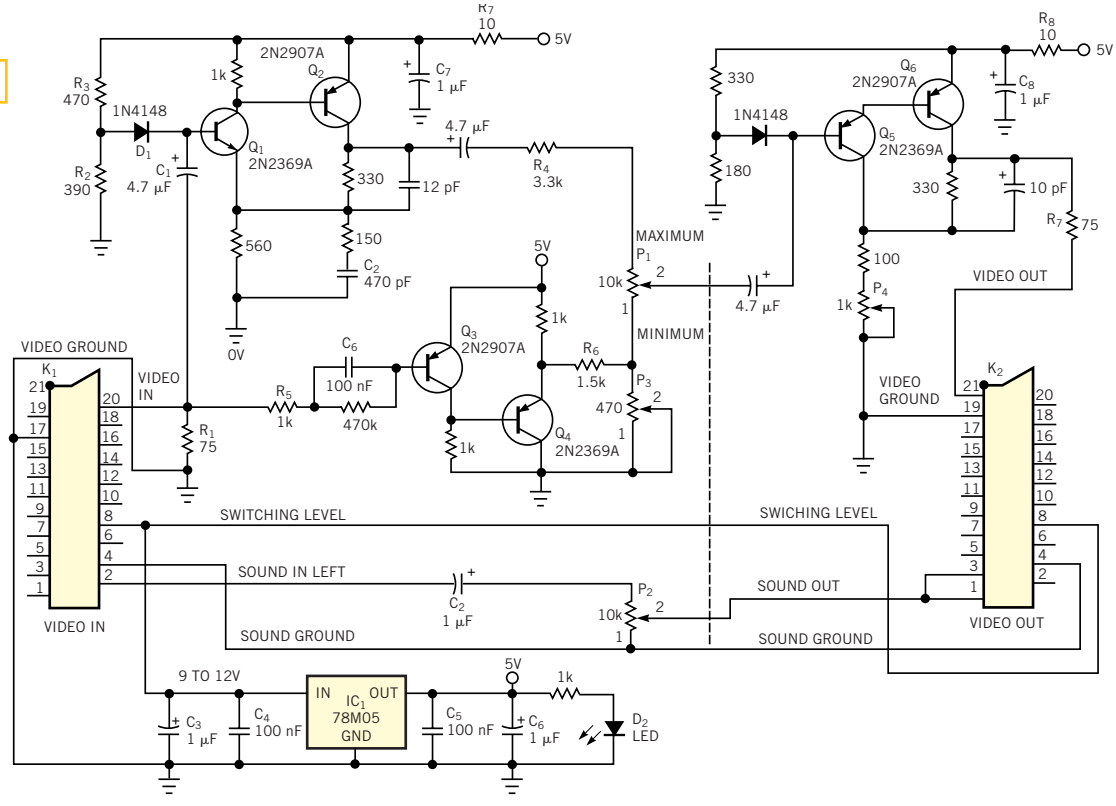


Figure 1

A simple circuit provides for effective video and audio fading when you're recording source material.

When you adjust P_1's wiper from one end to the other, the video image disappears and fades to a black screen. Because P_1 and P_2 are ganged, the sound follows the image brightness. The circuit could have used triple integrated video amplifiers, such as an AD813, and a video sync separator, such as an LM1881. However, these ICs are expensive (approximately $25) compared with the six standard transistors shown in **Figure 2**. R_1 sets the input impedance at 75Ω. Q_1, Q_2, and associated components form a video amplifier with an approximate gain of two. R_2, R_3, and D_1 set the dc voltage, and C_1 blocks any dc voltage from the source. The amplified video signal connects

Figure 2



A handful of transistors and associated components yields a professional-quality video fader.

to P_1 through R_4 and the C_1 dc-blocking capacitor. R_5, Q_3, Q_4, and associated components form a sync separator. The sync pulse connects to the bottom of P_1 through R_6 and P_3, an adjustable voltage divider. The wiper of P_1 connects to the second video amplifier comprising Q_5, Q_6, and associated components. You can adjust the amplification with P_4. R_7 sets the output impedance at 75Ω.

P_2, ganged to P_1, is a simple voltage divider, using C_2 to block dc voltages. The sound input uses the left channel, and the output goes to both the left and right channels. With a video source connected to video in, a dc voltage of 9 to 12V appears at Pin 8 of the video plug. IC_1 and the C_2 through C_6 capacitors derive power from this pin and provide a stable 5V for the circuit. D_2 is a high-brightness LED that indicates that a video source is present. R_7, C_7, R_8, and C_8 provide decoupled supplies for the amplifiers. Video cables are often of poor quality. For that reason, the circuit in **Figure 2** provides for amplification of the video signal. Also, compensation of the first amplifier provides amplification of color burst with a concomitant improvement of video quality. To adjust the circuit, first turn P_1 fully clockwise and then adjust P_4 for a good video image. Then, turn P_1 fully counterclockwise and adjust P_3 to obtain a stable black image.

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

Time-delay relay uses proximity control

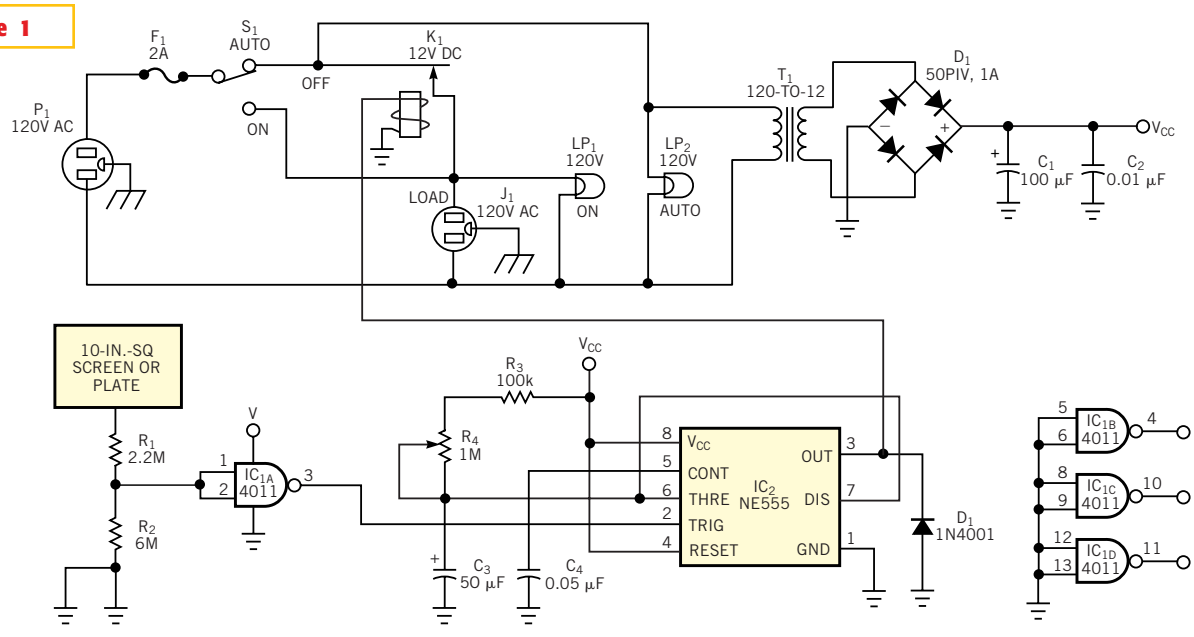
Dennis Eichenberg, Parma Heights, OH

You can build a circuit that allows a passerby to briefly operate model trains in a display window (**Figure 1**). The design uses a proximity detector rather than a pushbutton switch to eliminate the need to mount and wire any equipment outdoors. The circuit worked well in this application and other applications. The heart of the circuit is the quad CMOS NAND gate, IC_1. A human hand near the sensor induces 60-Hz power-line noise into IC_{1A}, and this IC triggers IC_2. IC_2 is configured as a monostable multivibrator, with a period equal to $1{\cdot}1(R_3+R_4)(C_3)$. The period is adjustable from approximately 0.5 to 50 sec. The sensitivity of the circuit depends on the size of the sensor. A 10-in.-sq piece of screen mounted inside the display window works well for this application, because it permits complete visibility through the sensor. The circuit triggers several inches away from the window in this application.

You can manually or automatically operate the circuit through the single-pole, double-throw, center-off switch, S_1. In automatic operation, receptacle J_1 is active when IC_2 drives relay K_1 on. Lamp LP_1 indicates when the load is active. Lamp LP_2 indicates when the circuit is in automatic mode. J_1 is active whenever S_1 is in the on position, as LP_1 indicates. V_{CC} for the circuit comes from transformer T_1, rectifier D_1, and filter capacitors C_1 and C_2. The load rating for the circuit depends on the selection of fuse F_1, switch S_1, relay K_1, and receptacle J_1.

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

Figure 1



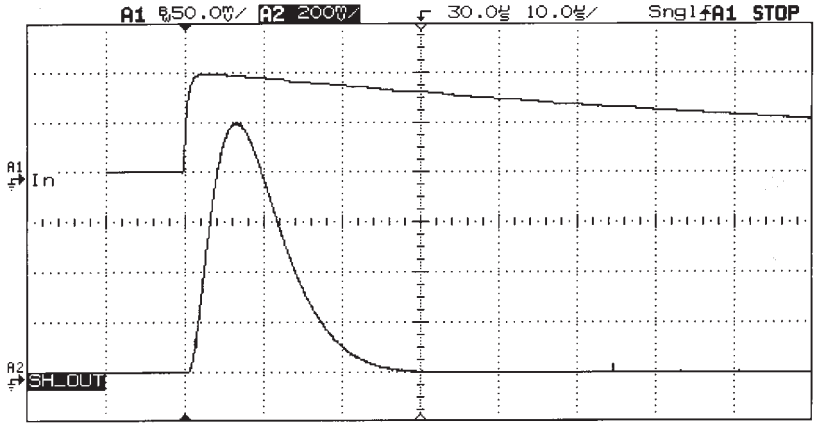
A proximity sensor turns a load on when a human hand comes near the sensor screen.

Circuit forms shaping amp and amplitude detector

Elio Rossi, ITESRE-CNR, Bologna, Italy

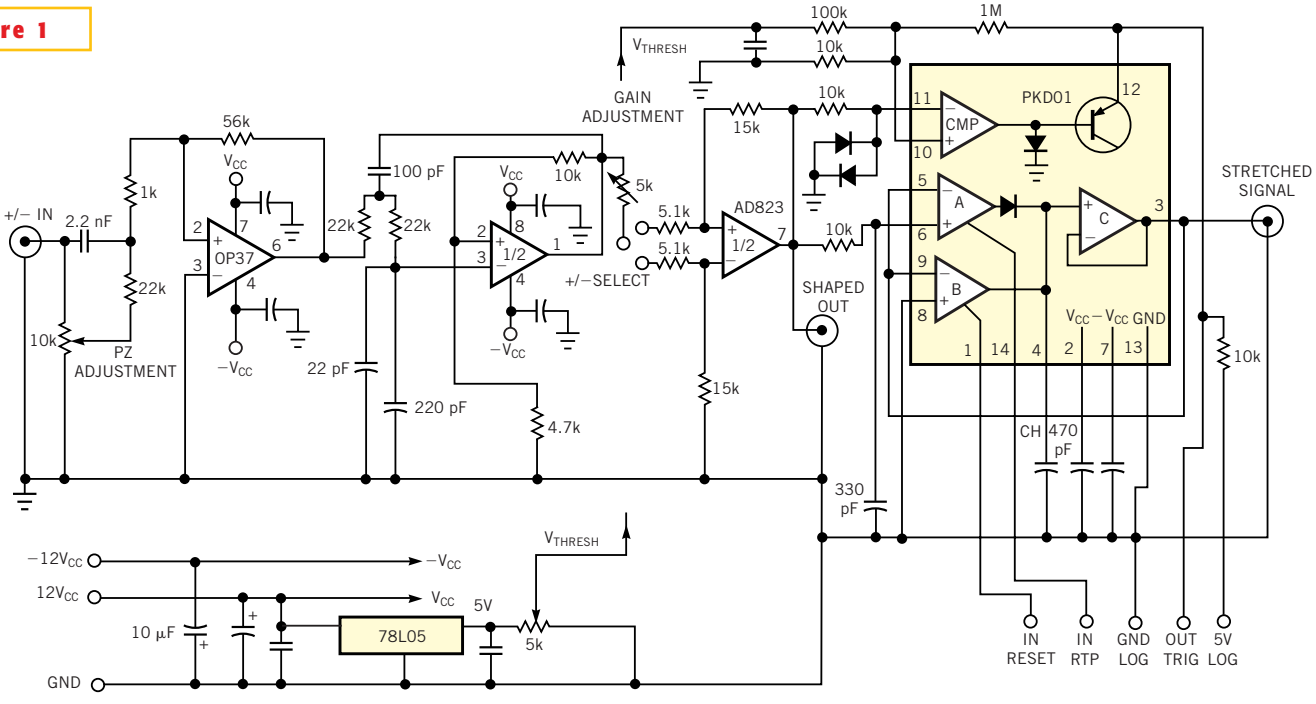
THE USE OF SOLID-STATE detectors connected to charge amplifiers requires appropriate conditioning of the output signals, because of the signals' long decay time. Moreover, you must "stretch" the peaks of the shaped pulses for a period sufficient for A/D conversion. For a single detector, you can use a relatively expensive module. For a large detector array, you need to develop an ASIC. For a moderate number of detectors, you can use an inexpensive circuit that handles an array of 19 drift diodes connected to a scintillating crystal of cesium-iodide (**Figure 1**). The two ICs, an OP37 and an AD823, provide the correct gain and the semigaussian conditioning of the input pulses. The conditioning involves a differentiating input with pole-zero adjust in the gain-of-35 first stage, a lowpass Sallen-

Figure 2



At the AD823 output in Figure 1, the signal peaks in 6 msec. A peak detector then "stretches" the signal.

Figure 1

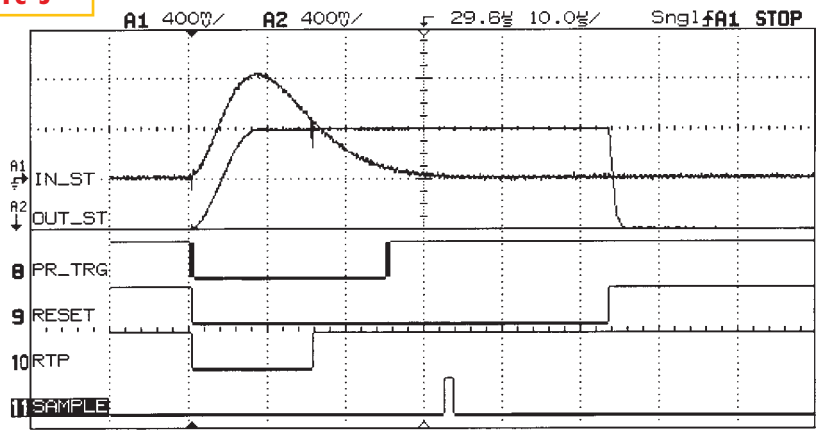


A few linear ICs form the basis of a shaping amplifier and peak-amplitude detector.

Key filter in the unity-gain second stage, and a gain-adjustable third stage. **Figure 2** shows a peaking time of 6 µsec with a gain of 50 to 100V/V. The circuit can use either a positive or a negative input signal.

The third IC, a PKD01, acts as a "stretcher" circuit with a built-in trigger discriminator. To measure the performance of the circuit, the design uses four external one-shot multivibrators, a precision pulse generator, and a multichannel analyzer in sample mode. The discriminator output triggers the sequence of the one-shots, which in turn open the in reset and close the in rise-time-protection, gated amplifiers, A and B (**figures 1** and **3**). This action allows the hold capacitor to reach the peak of the integrated (via the 10-kΩ, 330-pF network) input pulse, so that an A/D converter can begin its conversion. The circuit must maintain the stretched signal via the reset pulse until the end of conversion. Input integration is necessary to generate a delay between the aperture of the reset command and the peak of the input pulses to maintain the linearity of the low-level signal. Also, the approximately 0.7V/µsec slew rate of the stretched output requires at least 8 µsec in the peaking time of the input signal to obtain a stretched signal output greater than 5V.

Figure 3



In the second waveform down, the peak detector provides a sample/hold function and preserves the peak value of the signal.

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

design ideas

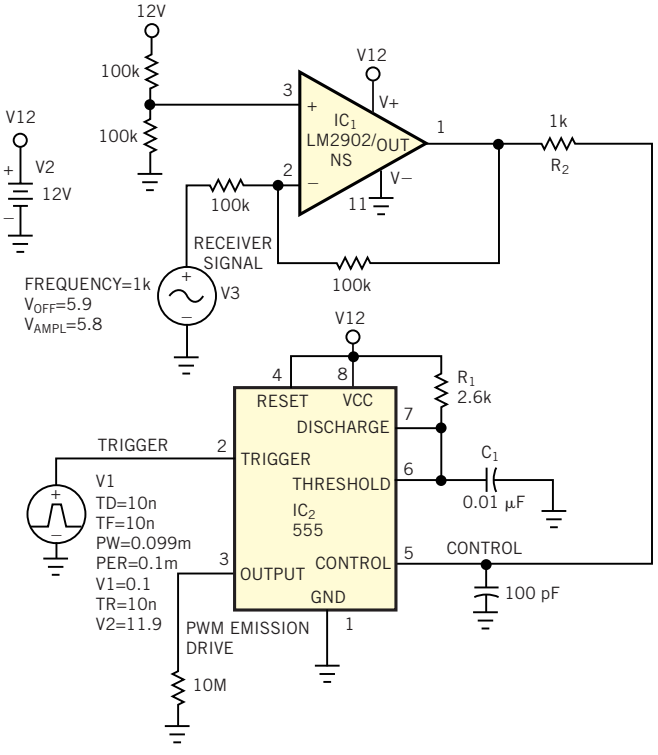
Edited by Bill Travis and Anne Watson Swager

PWM circuit controls sensor's AGC

Dongjie Cheng, Allegro Microsystems, Willow Grove, PA

All electronic sensors have their limits on working distances and environmental tolerances. *Dynamic range* defines a sensor's maximum allowable variations in the signal amplitude. ACG (automatic gain control) finds widespread use in systems to extend the dynamic range. Applications using photoelectric or ultrasonic techniques involve both emission and detection energy. In many cases, the emission first establishes a background receiver signal as a reference, and the receiver monitors the signal and detects any changes against this reference. It is often desirable to maintain the background signal at a moderate level so that the sensor works within its limits. A signal that is too weak cannot produce a significant SNR, and a signal that is too strong can disable the sensor by saturation or produce overheating of the sensor. Designers use different techniques to achieve satisfactory signal handling. The simplest method might be adjusting emission power or the receiver's gain by manually configuring jumpers or switches. An example of an AGC solution is to let a µP constantly adjust emission to a suitable level. **Figure 1** shows a simple PWM-based technique for maintaining the reference signal at an ideal level.

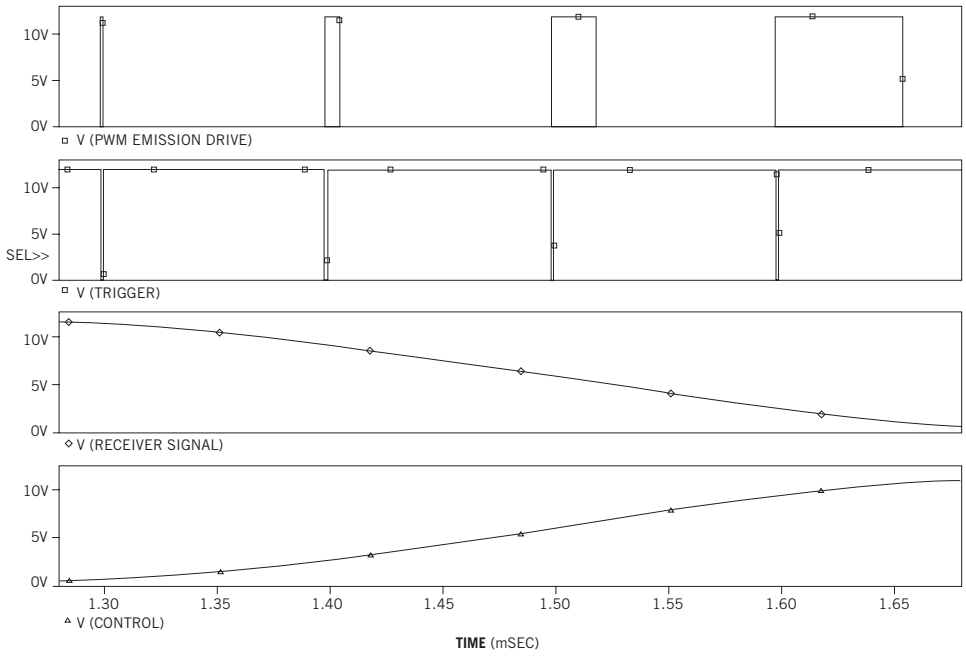
Figure 1



A PWM circuit provides an AGC function to increase a sensor's dynamic range.

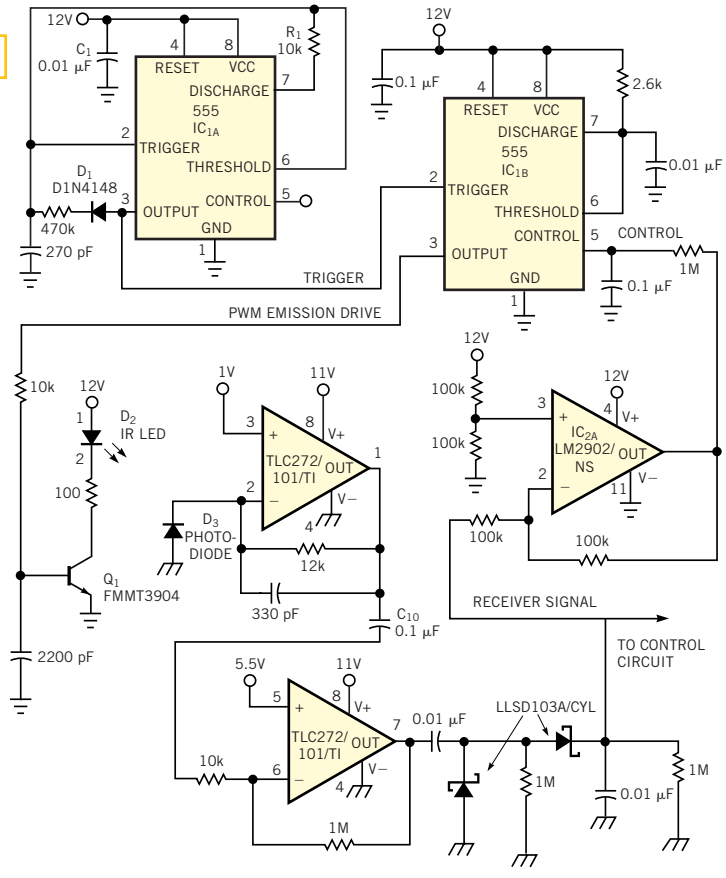
The concept embodied in **Figure 1**'s technique is that, if the sensor sees a strong background signal, the circuit slowly reduces the emission intensity at the fundamental frequency by reducing the pulse width of the emission drive signal. We verified the idea by using OrCAD/Cadence PSpice modeling, followed by experimental verification. The voltage source, V3, generates a 1-kHz sinusoidal signal named receiver signal. **Figure 2** plots this signal for a half-cycle in the third panel down. The model simulates the receiver's filtered, slowly varying background signal. IC_{2A} functions as an inverter. Its output passes through the lowpass filter R_2/C_2 to generate the Control signal in **Figure 2**, fourth panel down. IC_2, a 555 timer, acts as a pulse-width modulator. Its output is a pulse train named PWM emission drive (**Figure 2**, first panel). The frequency of the PWM emission drive follows the 10-kHz trigger signal, modeled with V1 and plotted in **Figure 2** in the second panel down. The modeled results in **Figure 2** indicate a negative correlation between the width of PWM emission drive and the level of receiver signal. As the receiver signal declines, the control signal widens the puls-

Figure 2



The PWM emission-drive signal gains width (top panel) in response to a declining receiver signal (third panel down).

Figure 3



es of PWM emission drive to boost the sensor's emission power at the fundamental frequency.

The process maintains the width of the PWM signal with the width fluctuating around a stable point. This point depends on the signal strength. If a sensor works near its upper range limit, the signal remains weak despite the increased pulse width, so the stable point shifts toward the maximum pulse width. In this Design Idea, the sensor is sensitive to the 10-kHz fundamental frequency because of a bandpass filter. Therefore, a 50% PWM duty cycle yields the maximum signal amplitude. On the other hand, if a sensor operates near its lower range limit, the pulse width converges to a very narrow width for a medium background signal. The time constant R_1C_1 in **Figure 1** also affects the location of the stable point. A longer time constant pushes the PWM stable point to a higher duty cycle and vice versa. In this case, the R_1C_1 time constant is half the trigger period (0.05 msec). You usually need to perform a test to determine R_1, C_1, or both. The test ensures that 50% is the maximum duty

This circuit proves the validity of the PSpice model for the PWM AGC circuit.

cycle from the modulator. The Trigger signal must be a narrow, negative pulse train.

Figure 3 shows a hardware setup to test the PSpice model, using an infrared sensor. IC_{1A} and IC_{1B} are 555 timers (the dual TLC556 or LM556). IC_{1A} is a free-running oscillator that supplies the 10-kHz Trigger pulses for the IC_{1B} PWM circuit. IC_{1B} drives the IR LED and Q_1. The 11, 1, and 5.5V voltages come from filtered voltage sources. A transimpedance amplifier first amplifies the 10-kHz photocurrent from the IR photodiode. The signal then goes through a second-stage amplifier and then undergoes filtering and peak detection to generate the quasi-dc receiver signal. IC_{2A} then inverts the signal to provide the PWM control. The PWM emission drive drives the IR LED in such a way that the IR photodiode receives medium-radiation intensities. Because the receiver's signal amplitude is proportional to the pulse width of PWM emission drive, the sensor constantly tries to reverse any trend of the background signal. In varying the distance between the IR LED and the photodiode, an oscilloscope showed signal behavior as predicted by the model.

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

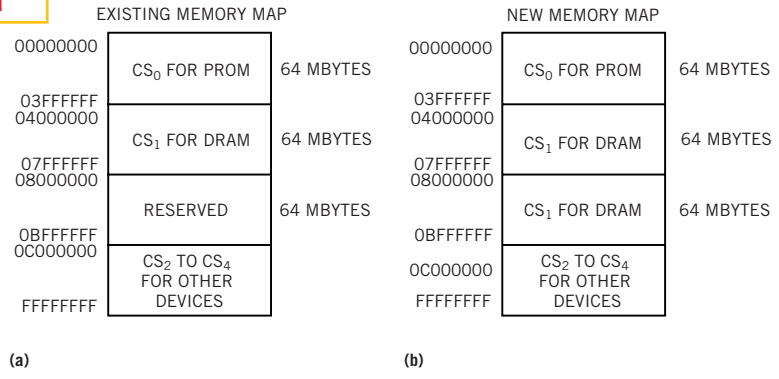
Two gates expand ASIC's memory-decoding range

Vinh Hoang, Ericsson Inc, Brea, CA

Many electronic circuits implement chip-select lines on an ASIC. From the beginning of the design cycle, the chip selects, CS_0 to CS_4, have defined bases on the memory map (**Figure 1**). Adding functions to the product requires increasing the DRAM space. Now, you must redesign the ASIC so the chip select, CS_1, can accommodate the new memory space of 04000000 to 0BFFFFFF.

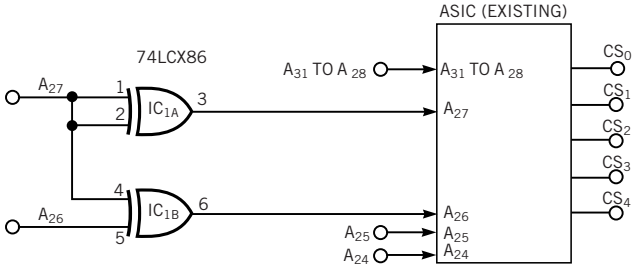
The circuit in **Figure 2** uses two external exclusive-OR gates to expand memory-address-decoding space for the CS_1 signal from the initial range of 04000000 to 07FFFFFF to 04000000 to 0BFFFFFF. When address line A_{27} is low, exclusive-OR gates IC_{1A} and IC_{1B} allow A_{27} and A_{26} signals to pass through unchanged. In this case, the ASIC decodes the address range 04000000 to 07FFFFFF, according to the existing memory map. However, when address line A_{27} is high, which accesses the address range 08000000 to 0BFFFFFF, both IC_{1A} and IC_{1B} act as inverters. Thus, the circuit inverts lines A_{26} and A_{27}, so that the ASIC now sees the address range 04000000 to 07FFFFFF instead of 08000000 to 0BFFFFFF. The function of the exclusive-OR gates is to map the CS_1 selected address range of 08000000 to 0BFFFFFF to 04000000 to 07FFFFFF.

Figure 1



As you modify a product, you may have to expand the product's memory space (a) to accommodate more DRAM (b).

Figure 2



Two exclusive-OR gates expand the memory-address-decoding space for CS_1.

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

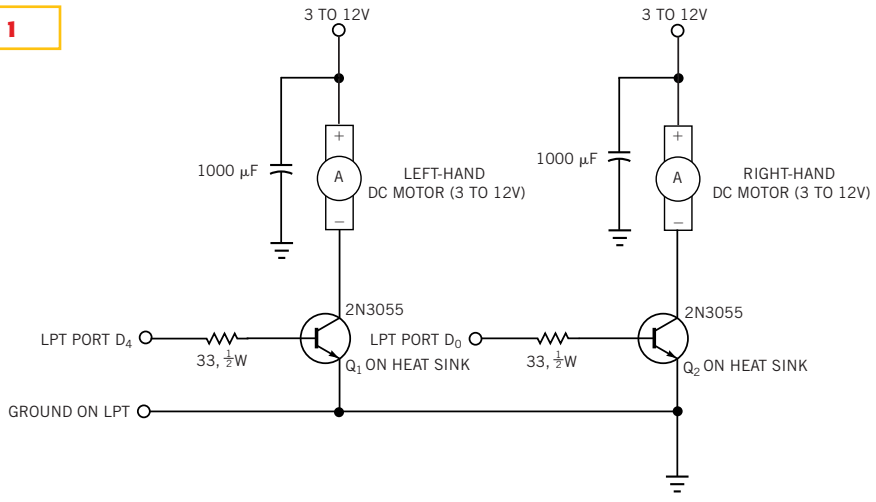
Dual dc-motor-speed controllers navigate robots

Shyam Sunder Tiwari, Robotics Software Pvt Ltd, Gwalior, India

Mobile robots need simple and lightweight dc-motor speed-control hardware that can work with low-power batteries. Flip-flop type navigation systems allow only one motor to operate at a time while the other motor remains off. Navigation direction changes alternatively even when the robot has to navigate through a straight path. Line-tracker robots are of this type.

Zigzag motion lets the robot look at the track toward its left and right such that it can correct its path if necessary. You can drive both motors if this type of motion is undesirable. The circuit in **Figure 1** has two independent motor-speed-control channels: one for a righthand-side motor and the other for a lefthand-side motor. Power to each motor is pulse-width-modulated using a Basic computer program (**Listing 1**). The power-driver circuit uses npn power transistors, Q_1 and Q_2. These transistors have high-power-kicking ability that the robotics require. The PC's parallel port directly controls the base of these transistors. LPT port data bit D_0 operates a righthand-side motor, and data bit D_4 operates a lefthand-side motor. Level one at the port pin turns on the motor power, and level zero turns off the motor power. If both D_0 and D_4 are set to one, then both motors operate together. Reverse control does not occur. Thus, only one motor needs to operate to turn the robot backward until rotation is complete. You can add feedback sensors to the hardware. These sensors are necessary to know the position of the robot. The circuit works for small dc motors operating from a power source in the range of 3 to 12V.

Figure 1



Independent control channels drive righthand and lefthand dc motors.

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

LISTING 1—MOTOR-SPEED CONTROL

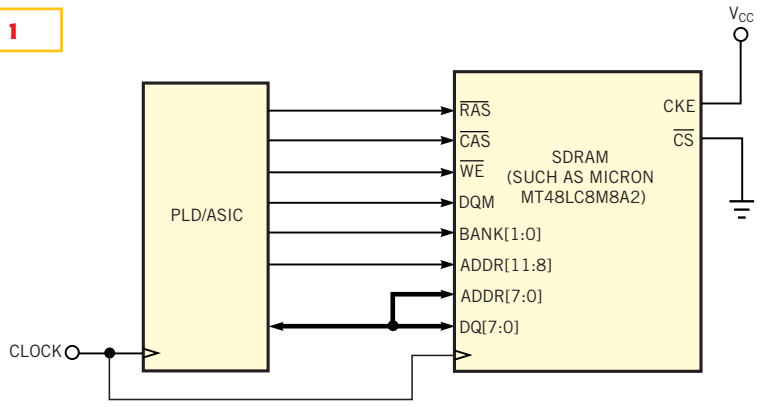
```
        CLS
        REM LPT1 port is located at address decimal 888
        REM
start:  INPUT "motor under action (1= left 2=right 3=both) =", m
        IF m > 3 GOTO start
        IF m < 1 GOTO start
        REM
        REM select the motor to be operated
        IF m = 1 THEN w = &H0F0 : REM left motor
        IF m = 2 THEN w = &H0F : REM right motor
        IF m = 3 THEN w = &H0FF : REM both motors
        REM
        REM Get the power factor for speed control
        REM
power:  INPUT "motor speed power factor (scale 10 to 80) =", p
        IF p > 80 GOTO power : REM check for upper limit
        IF p < 10 GOTO power : REM check for lower limit
period: INPUT "motor power on time (range 1-1000 seconds) =", s
        IF s > 1000 GOTO period : REM maximum time in seconds
        IF s < 1 GOTO period
        REM
        REM compute the PWM parameters here
        REM
        H = 20 * p
        REM H is high level PWM control output
        L = 20 * (80 - p)
        REM L is low level PWM control output
        CLS
        LOCATE 10, 5
        PRINT "
        LOCATE 11, 1
        FOR i = 1 TO 8
        PRINT ":";
        FOR j = 1 TO 9
        PRINT ".";
        NEXT
        NEXT
        LOCATE 10, 1
        FOR i = 1 TO p
        PRINT "-";
        NEXT
        PRINT ">": k = 0
        ON TIMER(1) GOSUB time
        TIMER ON
        REM

repeat: IF k > s GOTO done
        GOSUB control: REM Endless loop
        GOTO repeat
done:   STOP
time:   LOCATE 12, 25
        k = k + 1
        PRINT "(power="; p; " time="; k; "seconds)"
        RETURN
        REM Motor power switching subroutine
control: FOR i = 1 TO H
        OUT 888, w: REM 888 is the address of LPT1 data port
        NEXT i
        FOR i = 1 TO L
        OUT 888, 0
        NEXT i
        RETURN
```

SDRAM interface slashes pin count

Tim Hellman, M&M Consulting, Concord, MA

Many designs need deep buffering but don't require ultra-high-memory bandwidth. Examples include image and audio processing, as well as some deep-FIFO applications. These designs often use a single ×8 SDRAM device that connects to an FPGA or ASIC. This approach solves the buffering problem but also burns a lot of valuable pins, which can be as many as 27 for a single SDRAM device. The design in **Figure 1** takes advantage of the burst counter inside the SDRAM to reduce this pin count to 18 by multiplexing the lower eight address lines with the data. The efficiency loss is low; the design requires only one extra clock during the write burst. **Figure 1** uses an 8Mx8, 125-MHz SDRAM, but this technique works with any SDRAM.

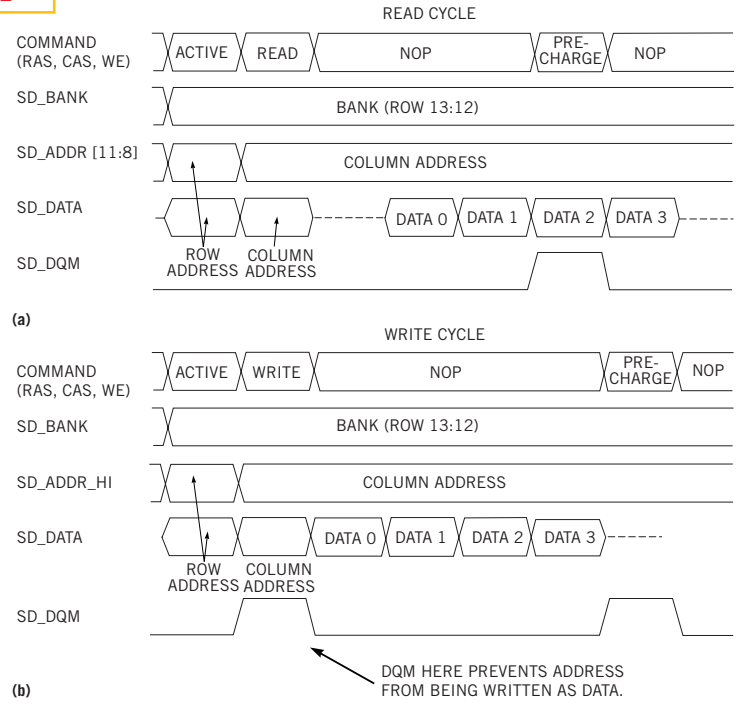
Figure 1



To reduce the interface-pin count to 18, you can take advantage of the burst counter inside the SDRAM by multiplexing the lower address and data lines.

The read- and write-cycle timing diagrams reveal the secret (**Figure 2**). The **figure** shows a burst of 4, but any power-of-2 burst works. These diagrams assume a 50-MHz system clock, a read latency of 2, and a full-page burst. During the read cycle (**Figure 2a**), the data bus is inactive during the initial portion of the cycle (when the row and column addresses are presented), so there's no problem with using the data bus to carry address data. A precharge command ends the burst and prepares the RAM for the next access.

Figure 2



For the write cycle, however, some trickery is necessary (**Figure 2b**). Normally, the first byte of write data is presented to the SDRAM with the Write command, along with the starting column address for that burst. By asserting the DQM (data-mask) signal, the SDRAM ignores the data lines during this phase, thus allowing them to be used for the column address.

Note that the DQM signal does not prevent the internal column address from incrementing, however. Thus, the write-column address presented with the Write command must be one less than the desired burst starting address. For FIFO designs, this requirement is trivial because you can initialize the write-address col-

During the read cycle (a), the data bus is inactive during the initial portion of the cycle so the data bus can carry address data. During the write cycle (b), asserting the DQM command causes the SDRAM to ignore the data line during this phase, which allows the data lines to carry the column address.

umn counter to −1 rather than 0. The column-address counter in the SDRAM wraps around at the end of the column, so this approach works even at the beginning of a column.

You can download a simplified version of a FIFO controller that uses this technique, described in the Verilog language, 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 #2659. The listing omits some of the details, such as SDRAM refresh/init cycles, and FIFO flags, to highlight the portions relevant to this design. This controller uses a simple eight-state finite-state machine to generate the SDRAM control signals and uses a pair of row/column counters to keep track of the FIFO put/get pointers. The special initialization and incrementing of the write row and column pointers satisfies the requirement that the write column start off one behind the desired write address. The code occupies 35% of a small Xilinx SpartanXL-S10 device, and runs at 50 MHz. For the sake of example, all of the outputs are combinatorial, but a true high-speed design should use registered I/O.

You can extend this idea to ×16 SDRAMs and to multiplex a few more of the address lines while getting a boost in memory bandwidth. If you do extend the idea, be careful with SDRAM line A_{10} because this line has special meaning during some SDRAM commands. You can also use this technique with double-data-rate SDRAMs.

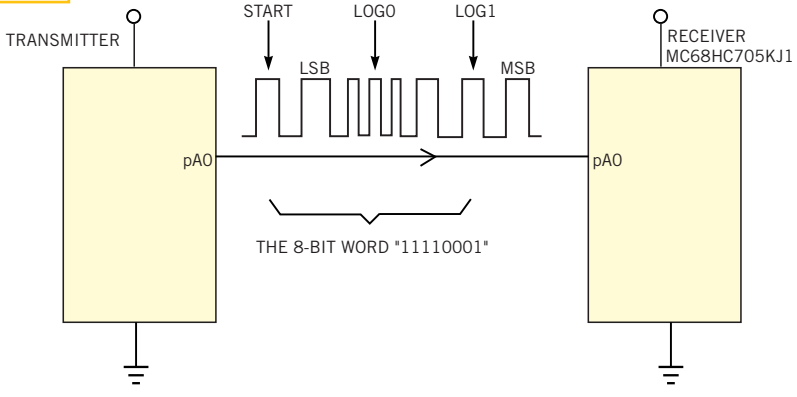
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Minimize communication time between small µCs

Abel Raynus, Armatron International Inc, Melrose, MA

When designing a multicontroller system, it is convenient to organize the communication between µCs via one wire line. Unfortunately, low-end µCs have no serial-interface capabilities like their more expensive counterparts. Low-end µCs have no SCI (serial-communication-interface, SPI (serial-peripheral interface), or SIOP (simple serial-I/O port). Thus, a designer needs to use software tools to create a serial interface (**Reference 1**). One approach, which uses the external interrupt for message receiving, results in message duration of 13.5 to 21.5 msec. For many applications, this time is unimportant. However, some applications may require you to minimize the message time as much as possible, especially when both the timer and external interrupts are in use. The µC cannot simultaneously execute these interrupt requests, and the external interrupt has a priority. Hence, a message and, therefore, the external-interrupt-service routine that are too long can affect the program-process timing related to timer interrupt.

Figure 1



The data word from the transmitting µC connects to Pin pA0 of the receiving µC, which you program as an external-interrupt input.

You cannot minimize the message time by just changing some number in the program. You must instead use a time-measurement concept that leads to modification of the whole program (**Figure 1**). The data word from one µC to another comes to Pin pA0, which you program as an external-interrupt input. When an external interrupt occurs, the µC can process this interrupt only when the current instruction execution is complete. This waiting time is always unpredictable and can range from zero to the time of the longest instruction in the pro-

gram (**Figure 2**). So, this time differs for each user program. Nevertheless, if a program has no exotic instructions, such as SWI (software interrupt) and MUL (multiplication), then the longest instruction can execute in six cycles, which takes 3 µsec for an oscillation frequency of 4 MHz (**Reference 2**). Saving the contents of the CPU registers on the stack and loading the program counter with an external-interrupt vector address take nine more cycles.

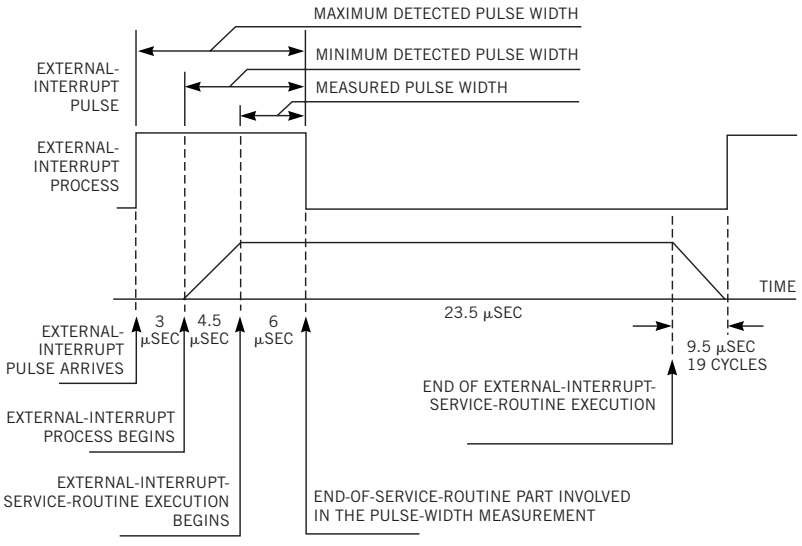
Only at this point can the interrupt-service routine begin the pulse-width measurement. (You can download the message receiver and transmitter programs *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2682.) After executing three instructions (lines 36 to 38 of the message-receiver program), this process completes in 12 cycles, or 6 µsec. Hence, the width of an external interrupt pulse cannot be less than 10.5 µsec.

From another side, it is more convenient for the transmitter part of the program to use only one subroutine, called Pulse, to generate all the pulses with different widths by changing the number of the loop repetitions (X) only. The loop duration is 5 µsec (lines 43 to 46 of the message-receiver program), so the generated pulse is equal to 5X+2.5 µsec. You can choose the pulse-width values according to your project objectives. In this case, for LOG0 X=2, width=12.5 µsec; for START X=4, width=22.5 µsec; and for LOG1 X=6, width=32.5 µsec. However, as you can see from **Figure 2**, the pulses that the receiver measures are shorter by 4.5 µsec than transmitted pulses. So, the measured pulses have widths of 8, 18, and 28 µsec. The µC's internal free-running timer/counter is the clock. You can at any time read the value of the first eight stages of this counter from the (TCR) time-counter register. One time count is equal to four machine cycles, or 2 µsec. Note that the pulse-width measurement must be complete before the TCR overflow that happens every 2×256, or 512 µsec. In the program, the unit of pulse-width value calculated and put into register pW is one timer count (clk). So the widths of LOG0, START, and LOG1 pulses are 4, 9, and 14 clk, respectively. Remember that the moment of beginning and external interrupt processing could be at any time in the range of 3 µsec, or 2 clk, which causes the same variations in the measured pulse width. To overcome this problem, you select the proper pulse width within a fork, which should greater than or equal to 2 clk. In this case, the fork is equal to 4 clk. Finally, you use the following pulse selection logic: 3 clk< LOG0<7 clk, 8 clk<START<12 clk, and 13 clk<LOG1<17 clk.

The minimal interval between the two consecutive external-interrupt pulses should be more than the number of interrupt-service-routine cycles plus 19 cycles (**Reference 2**). The service routine's longest path occurs when the program is processing the widest incoming pulse (LOG1), which takes 47 cycles. Thus, the interval between pulses should be more than 47+19=66 cycles, or 33 µsec. Choosing the number of time-loop repetitions of X=10 results in a 52.5-µsec pause between pulses (lines 48 to 52 of the message-receiver program). With all the above chosen values, the longest message, $\$ff_{HEX}$, lasts 0.84 msec.

The design in **Figure 1** uses a one-time-programmable Motorola MC68HC705JK1 (Motorola µC, but this idea is applicable to any µC. You would need to recalculate the timed values according to the µC's technical data.

Figure 2



The µC can service an external interrupt only when it finishes executing the current instruction, and the waiting time can range from zero to the time of the longest instruction in the program.

References

1. Raynus, Abel, "Single wire connects microcontrollers," *EDN*, Oct 22, 1998, pg 102.

2. MC68HC705KJ1/DTechnical Data, Revision 1.0, Motorola.

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

design ideas

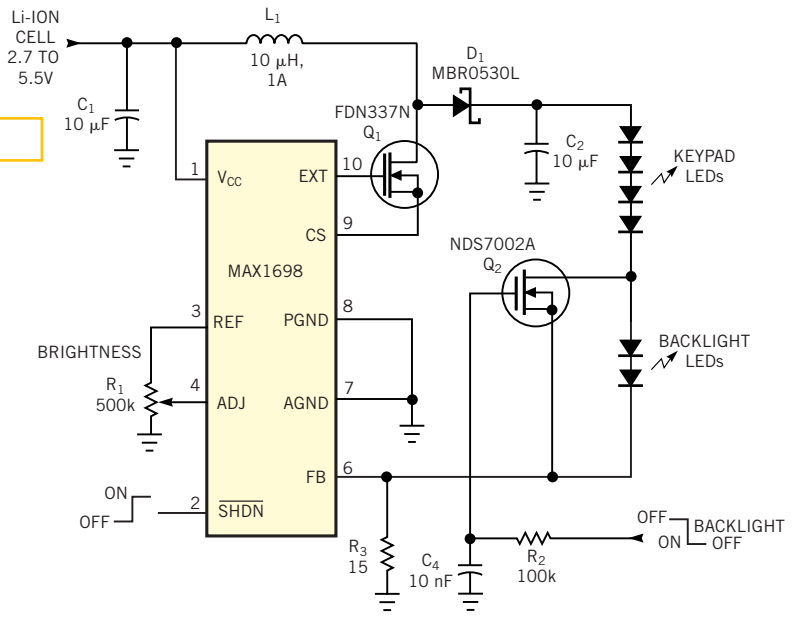
Edited by Bill Travis and Anne Watson Swager

Single FET controls LED array

Len Sherman, Maxim Integrated Products, Sunnyvale, CA

White-LED backlights are gaining acceptance because they offer higher reliability and simpler drive circuitry than backlights based on CCFL (cold-cathode-fluorescent-lamp) and EL (electroluminescent) technology. As a result, white-LED backlights are increasingly common in PDAs (personal digital assistants), cell phones, digital cameras, and other portable devices. A design in which the display requires backlighting for extended periods needs an efficient circuit that drives the LEDs with a controlled current and eliminates the wasted power associated with current-limiting resistors. **Figure 1** shows a switch-mode boost design that regulates current instead of voltage. Because all the LEDs are connected in series, they all receive the same current without the need for ballasting resistors. Identical currents help achieve uniform intensity. And, because the output current is low (20 mA in this case), the output-filter capacitance, C_2, can be smaller than for a load consisting of parallel-connected LEDs.

Figure 1



When this circuit turns off the backlight LEDs, the keypad LEDs remain on with no change in intensity.

The circuit's 90% conversion efficiency offers a distinct power-saving advantage over resistor-limited and linearly regulated designs. It might appear that a series-LED connection is unsuitable for applications in which some (but not all) LEDs must be off. A cell phone, for example, sometimes needs that capability for occasions when the display is off but the keypad remains lit. Or, a PDA might need to play a sound file while maintaining illumination in the buttons but not the display. In the circuit of **Figure 1**, switching off individual LEDs or groups of LEDs is not a problem, even with series drive. Applying a logic-high level to the gate of a simple MOSFET switch, Q_2, turns off a subset of LEDs by shunting their current. The remaining LEDs (for the keypad, for example) remain on, and their intensity remains constant because IC_1 regulates their current, by sensing the voltage across R_2 (300 mV at full brightness). When the circuit turns the LEDs on and off, the R_2-C_4 network at the gate of Q_1 slows the load changes sufficiently to prevent transients in the LED drive current. Other features include adjustable intensity via the ADJ pin and full shutdown via the $\overline{\text{SHDN}}$ pin.

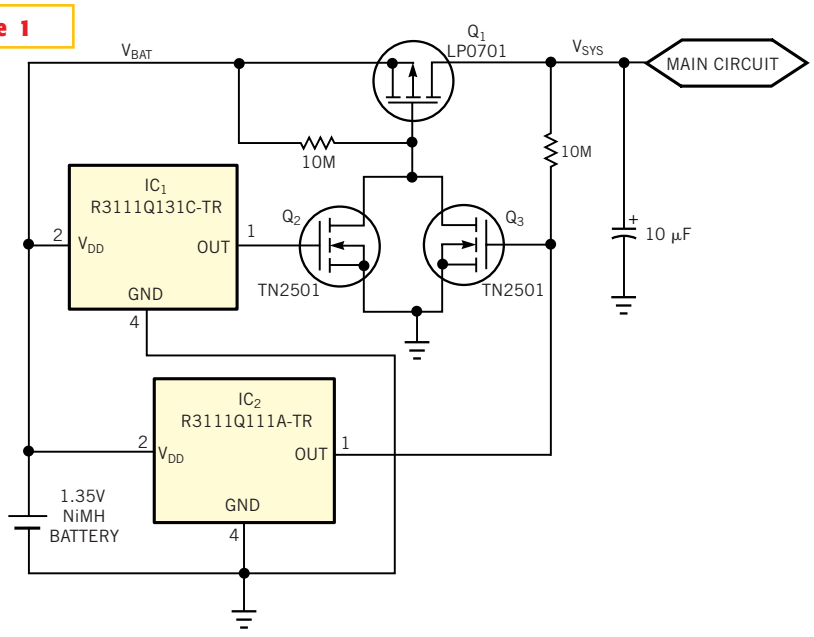
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Circuit protects battery from overdischarge

Martin Wuzik, Implex AG Hearing Technology, Ismaning, Germany

All electronic circuits powered by a battery discharge the battery. In some applications, it is undesirable to overdischarge the battery, because it could irreversibly reduce the battery's capacity and the number of discharge/charge cycles. The circuit in **Figure 1** protects a single NiMH (nickel-metal-hydride) cell by disconnecting the load from the battery. **Figure 2** shows the output voltage, V_{SYS}, versus the input voltage, V_{BAT}. For this NiMH battery, the switching points are 1.1 and 1.3V. If the battery discharges and V_{BAT} drops below 1.1V, Q_1 switches off ,and the node Main Circuit disconnects from the battery. In that case, the battery's only load is the pair of voltage detectors IC_1 and IC_2 from Ricoh (www.ricoh-usa.com). The load current of one detector is typically 800 nA, so the battery drain is 1.6 µA. The user must now charge the battery. Once the battery charges and the voltage reaches 1.3V, the load reconnects to the battery and remains connected as long as V_{BAT} stays above 1.1V.

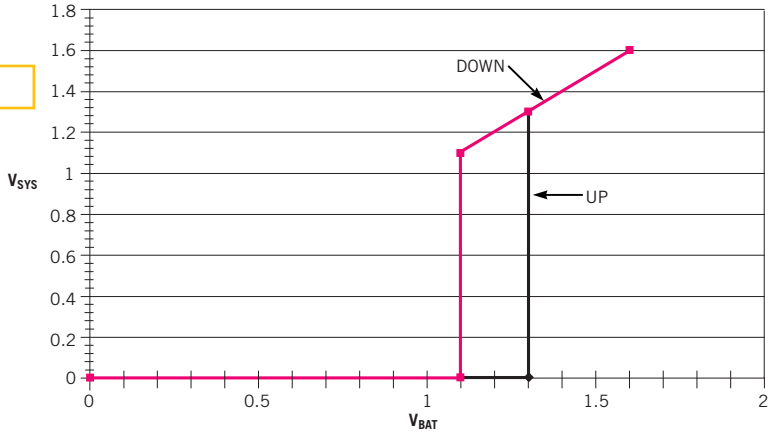
Figure 1



A simple circuit prevents excessive discharge of NiMH cells.

IC_1 is a voltage detector with a 1.3V setpoint and a push-pull output. IC_2 has a 1.1V setpoint. An important difference between the two detectors is that IC_2 has an open-drain output. If the battery voltage drops but remains within the 1.1 to 1.3V range, IC_1's output is low, and Q_2 switches off. Q_3 switches on because IC_2's output is still in the high-impedance state. If V_{BAT} drops below 1.1V, IC_2's output switches low, Q_3 turns off, and, as a result, Q_1 also switches off. As soon as V_{BAT} drops below 1.1V, the load disconnects from the battery. The load reconnects to the battery only when the battery charges to a voltage higher than 1.3V. At voltages of 1.1 to 1.3V, IC_2 cannot switch on Q_3 because the IC's output is an open-drain type and V_{SYS} is low. IC_1's output must assume a high state to switch on Q_2 and to finally switch on Q_1 on. The transistors are low-threshold MOSFETs from Supertex (www.supertex.com). The circuit uses no trimming resistors. You can select IC_1 and IC_2 off the shelf with 100-mV steps and 2% switching-point accuracy. You can adapt the circuit for the higher voltages of Li-ion batteries by selecting the voltage detectors.

Figure 2



The load disconnects from the battery when the voltage drops below 1.1V and reconnects when the battery charges above 1.3V.

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

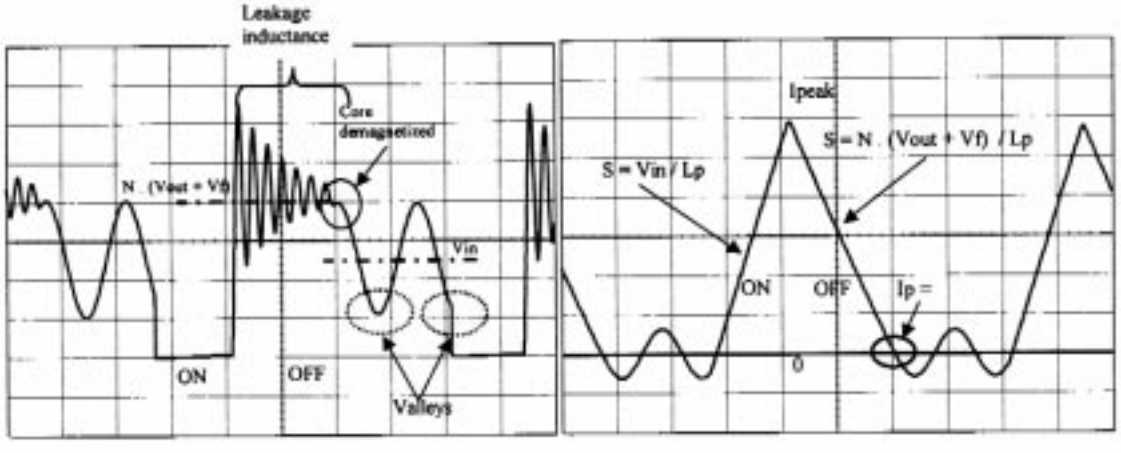
Two diodes change demagnetization-signal polarity

Christophe Basso, On Semiconductor, Toulouse, Cedex, France

Power-supply designers usually like flyback converters to operate in DCM (discontinuous-conduction mode) rather than in CCM (continuous-conduction mode). In DCM, the flyback converter is a first-order system at low frequencies, which eases the feedback-loop compensation. You can use a low-cost secondary rectifier, thanks to soft blocking conditions. In DCM, I_P goes to zero, and the diode stops conducting, whereas the power-switch turn-on event in CCM forces the diode to brutally stop conducting. Also in DCM, valley switching ensures minimum switching losses that C_{OSS} and all the parasitic capacitances bring.

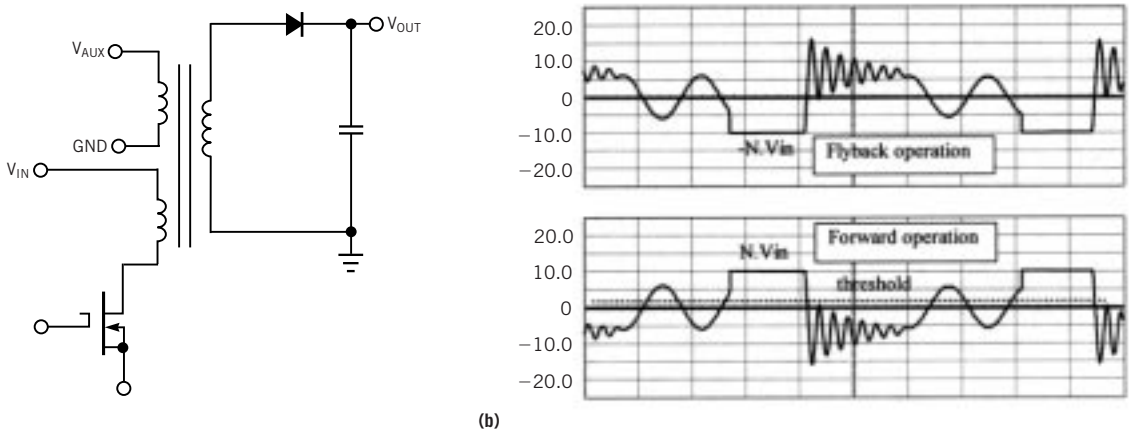
In valley switching, or QR (quasiresonant) operation, the curve of the drain-source voltage, V_{DS}, of a typical flyback converter, shows that when the power switch closes, you observe a low level due to the $R_{DS}(ON)\times I_P$ product (**Figure 1a**). At the switch opening, V_{DS} rises quickly and starts to ring at a high frequency because of the leakage-inductance presence. During this time, the primary current transfers to the secondary, and a reflected level of $N\times(V_{OUT}+V_F)$ appears on the MOSFET drain, where N is the secondary-to-primary turns ratio, V_{OUT} is the output voltage, and V_F is the secondary-diode forward drop. As soon as the primary current has fallen to zero in DCM operation, the transformer core is fully demagnetized (**Figure 1b**). The drain

Figure 1



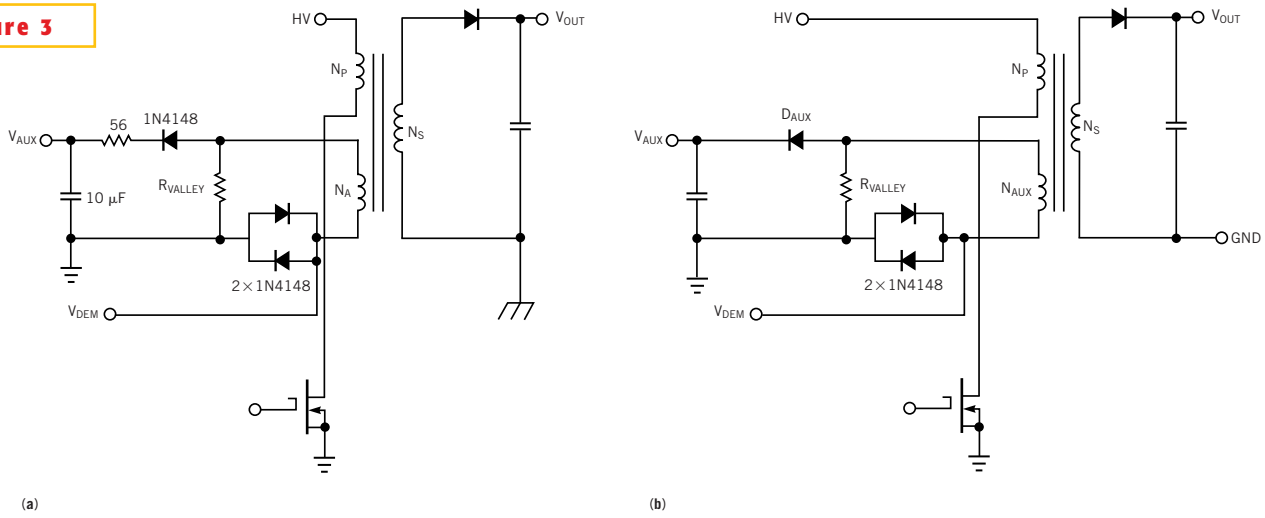
A typical drain-source waveform of a flyback converter shows high-frequency ringing (a). In DCM operation, the primary current ramps up and down to zero (b).

Figure 2



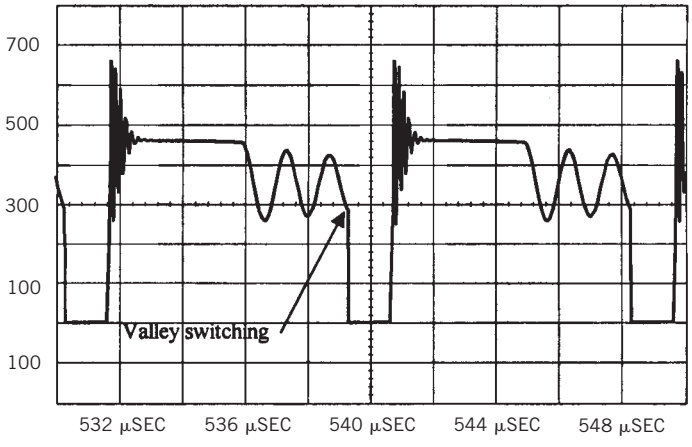
An auxiliary winding (a) lets you observe the flux image in the transformer's core for both flyback and forward operation (b).

Figure 3



A simple component arrangement allows forward-mode detection with a flyback-like PWM controller (a) or flyback-mode detection with a forward-like controller (b).

Figure 4



When you properly adjust the time constant using R_{VALLEY}, the switch restarts in the middle of the valley.

branch starts to ring but at a lower frequency than in **Figure 1a** because the primary inductance, L_p, is now involved.

This natural oscillation exhibits the following frequency value, where C_{LUMP} represents all of the circuit's parasitic capacitances, such as C_{OSS} and the stray capacitance from the transformer.

$$F_{RING} = \frac{1}{2\pi\sqrt{L_P \bullet C_{LUMP}}}.$$

As with any sinusoidal signal, there are peaks and valleys. When you restart the switch in the valley, all the parasitic capacitance values are at their lowest possible levels. Also, the capacitive losses, which are equal to $1/2 \times C_{LUMP} \times V_{DS}^2 \times F_{SW}$, are small because the MOSFET is no longer the seat of turn-on losses, which removes the usual turn-on parasitics. That is the secret of QR operation.

You can easily observe the core flux through an auxiliary winding (**Figure 2a**). Thanks to the coupling between the windings, the auxiliary section delivers a voltage image of the core's flux through the following formula:

$$V_{AUX} = N \bullet \frac{d\phi}{dt}.$$

Now, you can wire the winding either in flyback operation, as the power winding, or in forward operation. The observed signals look the same but have different polarity (**Figure 2b**). Note that both signals center about ground. The problem lies in the fact that most PWM controllers accept only the flyback polarity. Typical examples include the MC-33364 and MC44608 (www.onsemi.com). In battery-charger applications, you usually wire the auxiliary winding—the one that self-supplies the controller and gives the demagnetization signal—in forward mode. The reason is simple: When the battery you charge is close to 0V, the auxiliary windings are also nearly 0V because both windings are coupled in flyback mode. By operating in forward mode, whatever happens on the secondary side is invisible, and the voltage is always there to supply the controller. However, the demagnetization signal now has the wrong polarity, and the controller doesn't restart at the core's reset event.

Figure 3a shows a way around this problem. You still wire the winding for forward operation, but you add two extra diodes in series with the winding. At

the switch closing, you apply $N \times V_{HV}$, where N is the ratio between the auxiliary winding, N_A, and the primary winding, N_P. You clamp V_{DEM} to −0.6V, and the current circulates through R_{VALLEY}. At the switch opening, the voltage reverses and becomes positive but clamped to 0.6V on V_{DEM}. When this level collapses, the PWM controller reactivates the power switch.

You can implement this same type of circuit for PWM controllers that need a forward demagnetization signal but for which you would like to operate the auxiliary winding in flyback mode (**Figure 3b**). The problem and the cure are similar.

When you properly select R_{VALLEY}, this resistance naturally combines with sense-pin internal capacitance to add switch delay right in the middle of the wave (**Figure 4**).

Some controllers exhibit different demagnetization threshold levels. The MC33364 starts at around 1V, and the MC44608 toggles at 65 mV. Because of the diodes, you clamp V_{DEM} between ±600 mV, which could not trigger the MC33364. A small offset from the internal reference to the demagnetization pin brought by a 150-kΩ resistor and a typical R_{VALLEY} of 10 kΩ have provided good circuit operation.

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

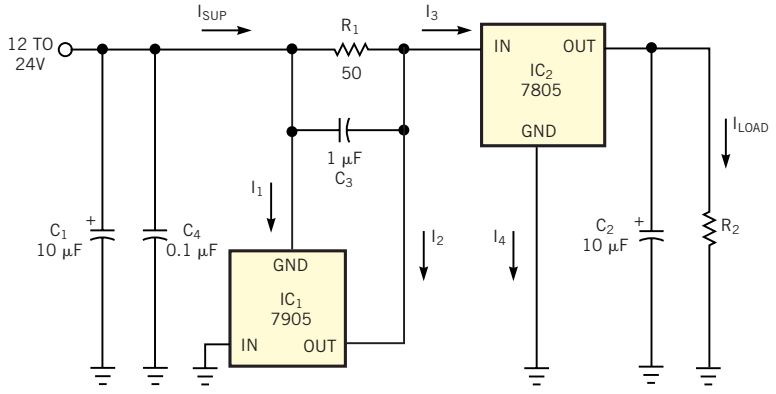
Simple scheme keeps current drain constant

Peter Güttler, APS Software Engineering GmbH, Cologne, Germany

It is sometimes advantageous to keep the overall current consumption of an electronic device constant. A large, seven-segment display, for example, draws nearly zero current when no segment is on to hundreds of milliamps when fully lit. This heavily varying current can cause EMI problems when a device receives its power through long cables from a remote power supply. The low-parts-count circuit in **Figure 1** keeps current consumption constant. IC_2 is an ordinary three-terminal regulator that supplies 5V to the load, R_2. IC_2 draws a total current $I_3 = I_{LOAD} + I_4$. (I_4 is approximately 8 mA, the quiescent current of IC_2). The negative three-terminal voltage regulator, IC_1, maintains 5V across R_1. The current through R_1 is $I_2 + I_3$. So, $I_2 = 5V/R_1 - I_3$, and total supply current $I_{SUP} = I_1 + 5V/R_1$. I_1 is approximately 2 mA, the quiescent current of IC_1. If the load draws more current, IC_1 reduces I_2 and vice versa.

This regulation works well as long as I_3 is smaller than $5V/R_1$. If the load draws more current, IC_1 stops regulating and the voltage drop across R_1 rises above 5V. This example sets R_1 at 50Ω, setting the supply current, I_{SUP}, to approximately 102 mA. C_1 and C_4 are input-filter capacitors, C_2 improves ripple rejection, and C_3 provides stability. Note that R_1 dissipates $(5V)^2/R_1$ and must have an adequate power rating. IC_1 and IC_2 may require heat-sinking. The minimum supply voltage for this circuit is 12V. (The minimum input voltage for $IC_2 = 7V + IC_1$'s reference voltage.) If your application cannot tolerate the 5V drop across R_1, try using an LM337 with a 1.25V reference voltage for IC_1.

Figure 1



This circuit maintains a constant supply current of approximately 102 mA.

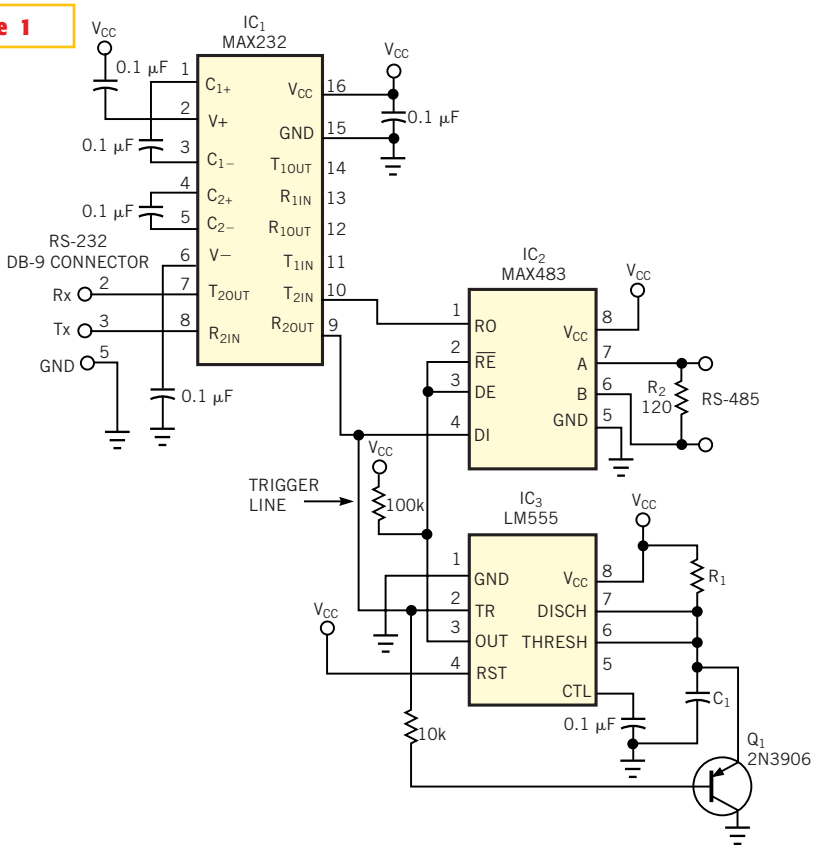
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

RS-232/485 converter has automatic flow control

John Howard, Kw Aware, Ventura, CA

RS-485 communications can provide longer range and better noise immunity than RS-232, as well as multidrop capability. Because it does not have separate transmit and receive lines, RS-485 requires flow control. RS-232/485 converters often use one of the RS-232 handshaking lines to control direction, but several communications-software packages do not support flow control. The circuit in **Figure 1** is an RS-232/485 converter that uses the transmitted signal itself to control the flow. The circuit uses MAX232 and MAX483 interface circuits, IC_1 and IC_2 from Maxim Integrated Products (www.maxim-ic.com) to convert between the ICs' respective signal levels and logic levels. Because both ICs invert the signal, the circuit preserves the original sense of the signal. The MAX483 is normally in the receiving mode. When transmission begins, the signal triggers IC_3, the LM555 timer, which in turn toggles IC_2's DE and RE lines, putting the chip into the transmitting mode.

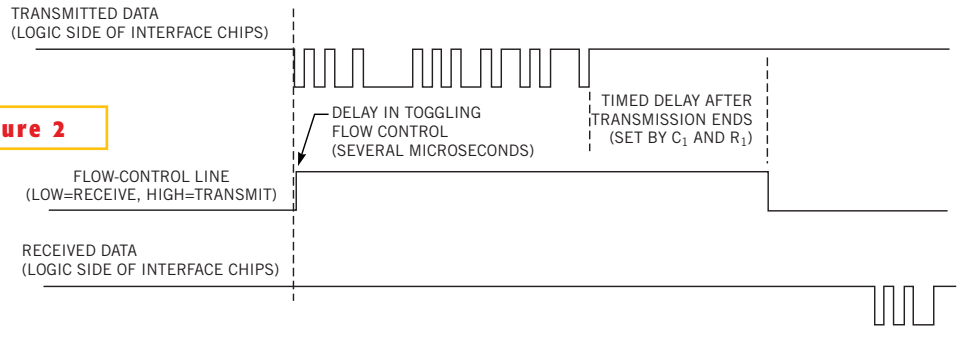
Figure 1



Automatic flow control makes RS-232/485 conversion easy.

Q1, the 2N3906, fully discharges C_1 each time the trigger line goes low, restarting the timing cycle. The values of R_1 and C_1 determine how long IC_3 maintains the transmitting mode after transmission ends. This interval should be long enough such that the converter doesn't switch directions while sending characters containing long sequences of zeros. On the other hand, it shouldn't be so long that the converter misses received characters. The interval T in seconds is $T = 1/R_1C_1$, where R_1 is in ohms, and C_1 is in farads. The flow control responds within a few microseconds after transmission commences, so the converter does not miss any bits at low and medium data rates. The application for this circuit operates at 14,400 bps. **Figure 2** shows the timing of the serial and flow-control lines. The entire circuit can fit into a DB-25 (or even a DB-9) back shell.

Figure 2



R_1 and C_1 determine how long the transmitting mode lasts.

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

Circuit provides accurate RTD measurements

Tito Smailagich, ENIC, Belgrade, Yugoslavia

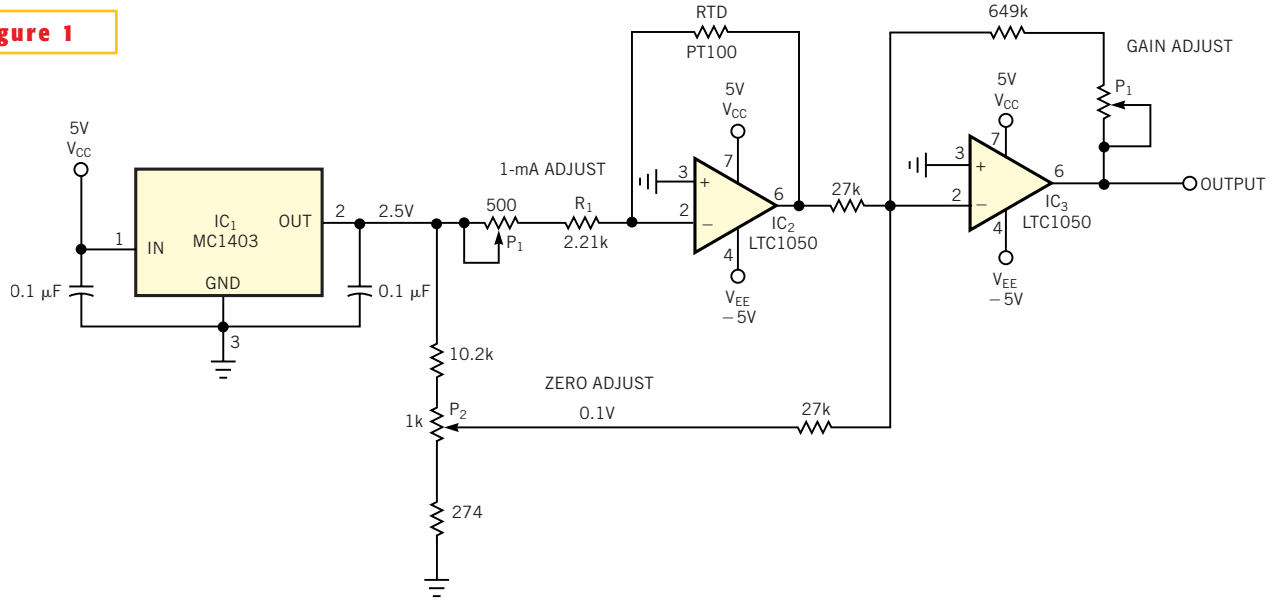
THE CIRCUIT IN **Figure 1** is an efficient measuring circuit for PT100 RTD elements. IC_1 provides an accurate 2.5V output and, together with P_1 and R_1, also provides a stable 1-mA current to the RTD element. The output of IC_2 is −0.1V. P_2 provides a zero adjustment (at 0°C) for IC_3, an amplifier with a gain of 25. P_3 provides a gain adjustment. If, for example, you replace the RTD element by a fixed resistance of 124.78Ω (the RTD's resistance in **Table 1** at 64°C), you would trim P_3 to obtain 0.64V output. Tolerance values for Class A and B elements are ±0.35 and ±0.8°C, respectively. You can use the standard values from **Table 1** or, if you need more accuracy, you can calibrate the PT100 element in a controlled-temperature environment.

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

TABLE 1–RESISTANCE VERSUS TEMPERATURE FOR PT100 RTD ELEMENT

°C	Ω	°C	Ω	°C	Ω	°C	Ω	°C	Ω
−20	92.16	6	102.34	32	112.45	58	122.47	84	132.42
−19	92.55	7	102.73	33	112.83	59	122.86	85	132.8
−18	92.95	8	103.12	34	113.22	60	123.24	86	133.18
−17	93.34	9	103.51	35	113.61	61	123.63	87	133.57
−16	93.73	10	103.9	36	114	62	124.01	88	133.95
−15	94.12	11	104.29	37	114.38	63	124.39	89	134.33
−14	94.52	12	104.68	38	114.77	64	124.78	90	134.71
−13	94.91	13	105.07	39	115.15	65	125.16	91	135.09
−12	95.3	14	105.46	40	115.54	66	125.54	92	135.47
−11	95.69	15	105.85	41	115.93	67	125.93	93	135.85
−10	96.09	16	106.24	42	116.31	68	126.31	94	136.23
−9	96.48	17	106.63	43	116.7	69	126.69	95	136.61
−8	96.87	18	107.02	44	117.08	70	127.08	96	136.99
−7	97.26	19	107.4	45	117.47	71	127.46	97	137.37
−6	97.65	20	107.79	46	117.86	72	127.84	98	137.75
−5	98.04	21	108.18	47	118.24	73	128.22	99	138.13
−4	98.44	22	108.57	48	118.63	74	128.61	100	138.51
−3	98.83	23	108.96	49	119.01	75	128.99	101	138.88
−2	99.22	24	109.35	50	119.4	76	129.37	102	139.26
−1	99.61	25	109.73	51	119.78	77	129.75	103	139.64
0	100	26	110.12	52	120.17	78	130.13	104	140.02
1	100.39	27	110.51	53	120.55	79	130.52	105	140.4
2	100.78	28	110.9	54	120.94	80	130.9	106	140.78
3	101.17	29	111.29	55	121.32	81	131.28	107	141.16
4	101.56	30	111.67	56	121.71	82	131.66	108	141.54
5	101.95	31	112.06	57	122.09	83	132.04	109	141.91
								110	142.29

Figure 1



This circuit provides accurate temperature measurements using a PT100 RTD element.

Edited by Bill Travis and Anne Watson Swager

Delay line aids in one-shot simulations

Christophe Basso, On Semiconductor, Toulouse, France

Many designers use small pulse generators to delay signals, open timing windows, drive sample/hold circuits, and other functions. Though the hardware implementation of these generators does not pose any problems, the lack of dedicated circuitry sometimes puzzles the Spice simulation of the system. A common approach to this problem is to implement a time constant involving a resistor, a capacitor, and a comparator. Unfortunately, each time you need a time constant, you must recalculate the resistor value, the capacitor value, or both. Despite the fact that inline equations can do this job for you, delay lines can often offer a smarter solution. **Figure 1** shows the implementation of a small pulse generator. The operating principle of the circuit lies in applying two "1" levels to the AND-gate input before the delay line switches high. **Figure 2** shows the signals associated with the circuit in **Figure 1**. **Listing 1** shows netlists for Intusoft's IsSpice4 and Cadence's PSpice.

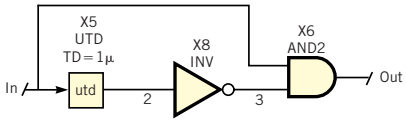


Figure 1 A one-shot multivibrator (shown with Spice nomenclature) makes a simple short-pulse generator.

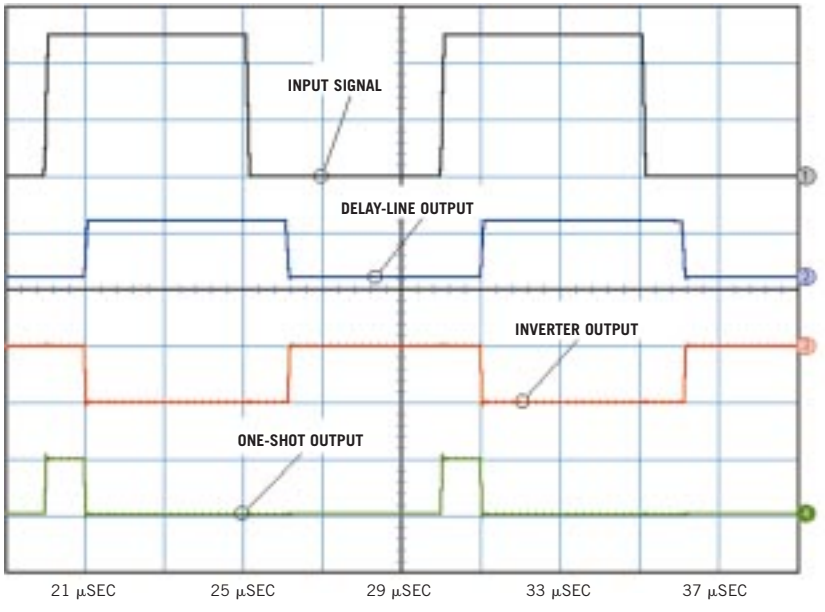


Figure 2 This plot shows timing details for Figure 1's Spice model.

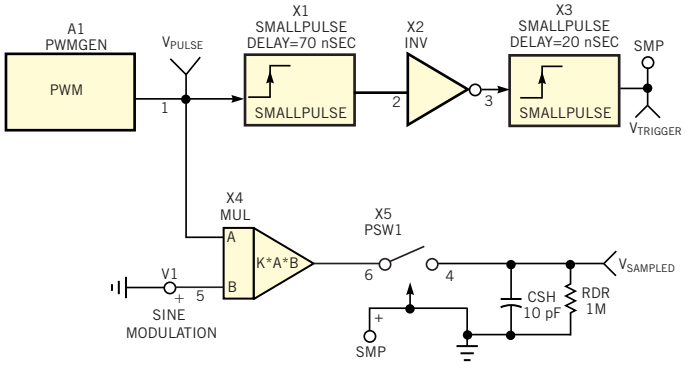


Figure 3 This PWM application is a sample/hold circuit in Spice-simulation nomenclature.

Figure 3 shows a typical application circuit for the one-shot multivibrators. You can use IsSpice4 or PSpice to simulate this sample/hold circuit. **Figure 4** shows the waveforms associated with the circuit in **Figure 3**. A PWM signal (top waveform) generates a kind of arbitrary staircase signal. The multiplier, X4, sinusoidally modulates the PWM signal. The circuit cascades two small pulse generators (SMALLPULSE). One creates a delay signal to sample at a given time (X1, 70 nsec); the other calibrates the width (X3, 20 nsec) of the sampling signal (second waveform). The third waveform in **Figure 4** shows the sinusoidally modulated signal; the fourth waveform is the sampled signal. You can download the IsSpice4 and PSpice listings for three one-shot types 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 #2680.

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

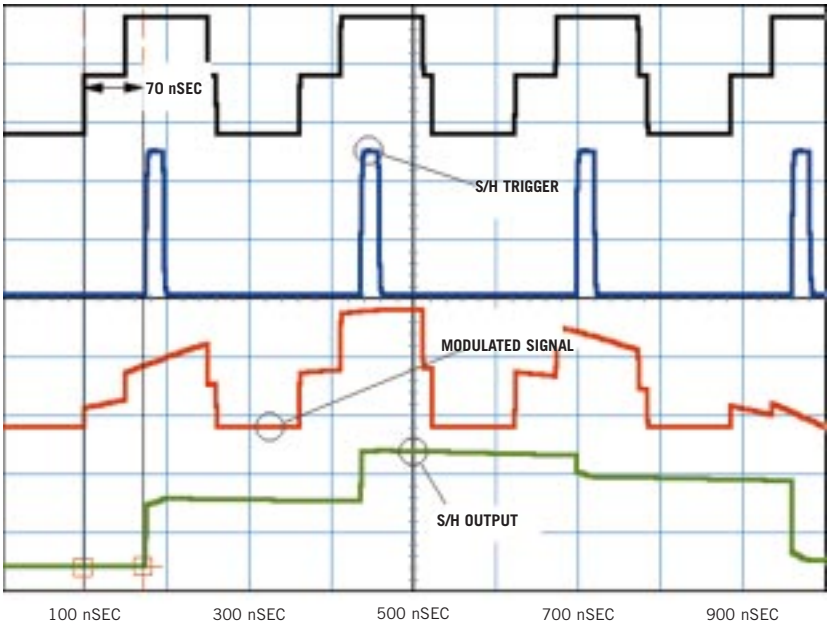


Figure 4 These timing signals illustrate the operation of Figure 3's circuit.

LISTING 1—NETLIST FOR MONOSTABLE SHORT-PULSE GENERATOR

IsSpice4

```
.SUBCKT SMALLPULSE In Out {DELAY=3u}
X5 In 2 UTD PARAMS: TD=DELAY
X6 In 3 Out AND2
X8 2 3 INV
.ENDS
*INCLUDE MNFLOPS.LIB
*******
****** MODELS ****
.SUBCKT UTD 1 2 {TD=???}
*
RIN 1 0 1E15
E1 3 0 1 0 1
T1 3 0 2 0 Z0=1 TD={TD}
R1 2 0 1
.ENDS
*******
.SUBCKT INV 1 2
B1 4 0 V= V(1)>800M ? 0 : 5V
RD 4 2 100
CD 2 0 10P
.ENDS INV
*******
.SUBCKT AND2 1 2 3
B1 4 0 V= (V(1)>800M) & (V(2)>800M) ? 5V : 100m
RD 4 3 100
CD 3 0 10P
.ENDS AND2
*******
```

PSpice

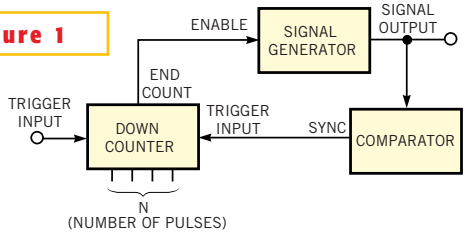
```
SUBCKT SMALLPULSE In Out PARAMS: DELAY=3u
X5 In 2 DL PARAMS: TD={DELAY}
X6 In 3 Out AND2
X8 2 3 INV
.ENDS
*******
****** MODELS ****
.SUBCKT DL 1 2 PARAMS: TD=500n
*
RIN 1 0 1E15
E1 3 0 1 0 1
1 3 0 2 0 Z0=1 TD={TD}
R1 2 0 1
ENDS DL
*******
**** 1 INPUT INVERTER ****
.SUBCKT INV 1 2
EB1 4 0 VALUE = { IF ( V(1)>800M, 0, 5V ) }
RD 4 2 100
CD 2 0 10P
.ENDS INV
*******
**** 2 INPUT AND CIRCUIT ****
.SUBCKT AND2 1 2 3
EB1 4 0 VALUE = { IF ( (V(1)>800M) & (V(2)>800M), 5V, 0 ) }
RD 4 3 100
CD 3 0 10P
ENDS AND2
*******
```

Sine-wave generator outputs precise periods

JM Terrade, Clermont-Ferrand, France

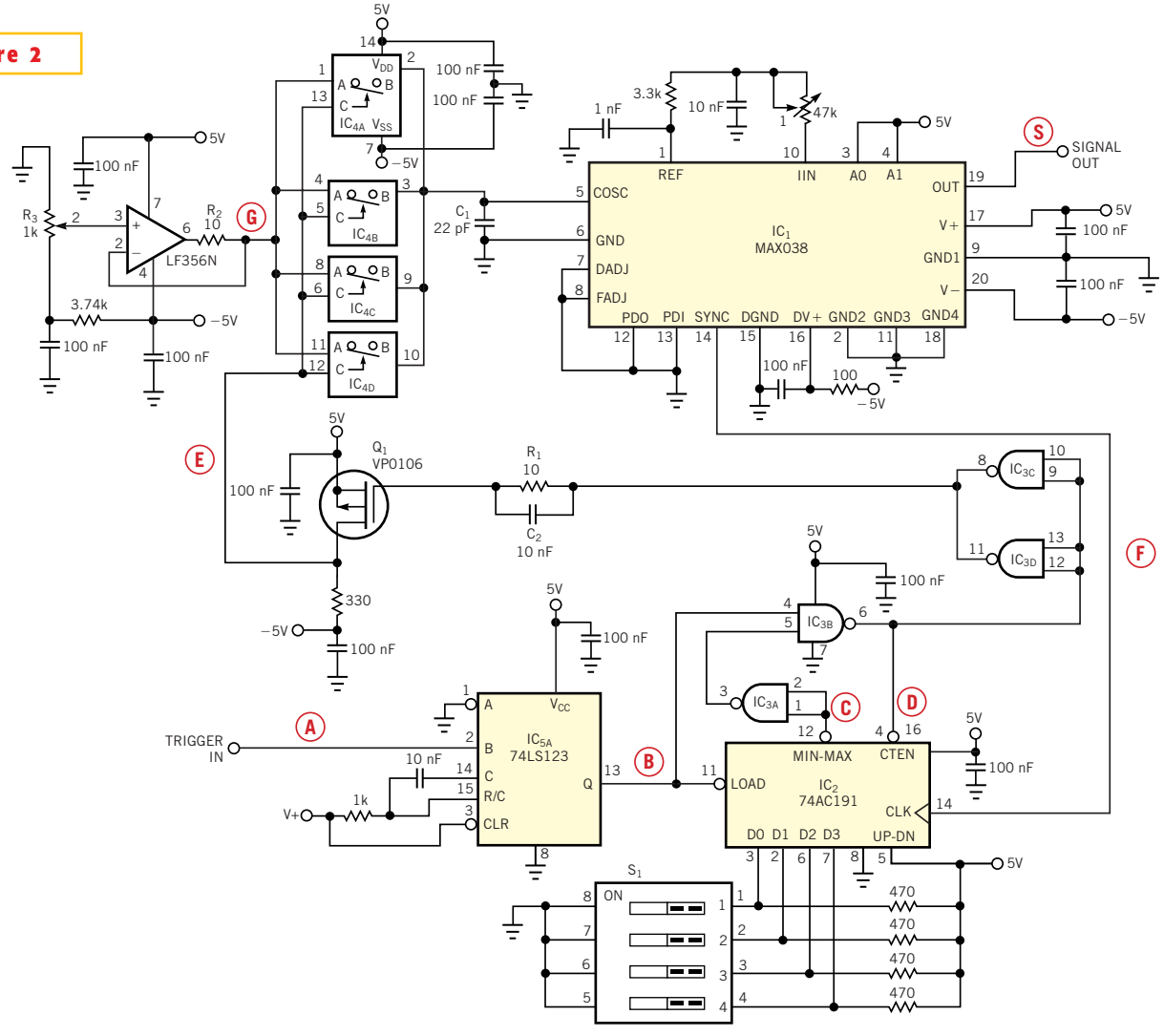
Rectangular pulse generators, even at high frequencies, are easy to design. However, the design becomes more difficult if you need a signal that contains a precise number of periods with a sinusoidal shape. Although it is easy to produce a good sine wave, the difficulty is producing a signal with a precise number of periods. The signal has to start and stop exactly at 0V. The scheme in **Figure 1** can produce one to 15 periods of a 20-MHz sinu-

Figure 1



A signal generator, down counter, and comparator can produce an output signal that contains a precise number of sine-wave periods.

Figure 2



IC_1 contains the generator and comparator. The sync signal at point F drives the counter, IC_2. The setting of S_1 determines the end of count and thus the number of periods, N, at the output.

soidal wave. The scheme has two main characteristics: The positive edge of a 5V signal triggers the input, and the output is one to 15 periods of a 20-MHz signal, adjustable to within ±10%.

Initially, the trigger input is inactive, the generator is disabled, and the counter is loaded with the number N. When the trigger input becomes active, the counter waits, and the end-count output enables the generator. A sine wave appears at the output. At the end of each period of the sine wave, the comparator produces a sync signal that drives the counter's clock input. When N periods of the output wave have occurred, the end-count signal disables the generator.

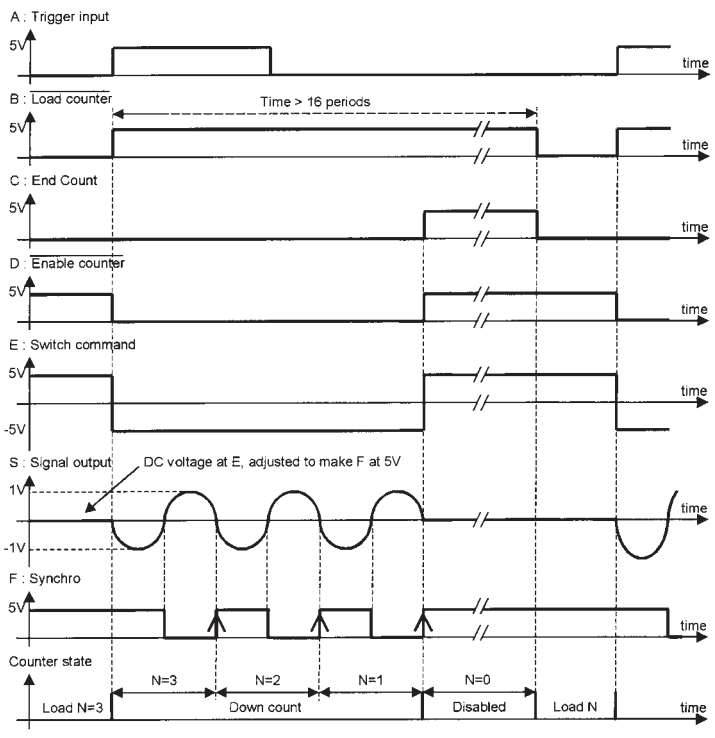
In the actual circuit, a MAX038, IC_1, is the generator (**Figure 2**). This IC contains the sine-wave generator and the comparator. The sync signal is available at Pin 14. The counter has to be fast enough to stop the generator before the next output period starts. The 74AC191, IC_2, is a 4-bit up/down counter with preset inputs. NAND gates IC_{3A} and IC_{3B} disable the counter after the end-count goes active. A one-shot circuit, IC_{5A}, ensures that the input trigger pulse is long enough to allow 15 pulses. A MOS switch, IC_4, short-circuits the oscillator capacitor, C_1, to stop IC_1's generator. If the circuit simply grounds C_1 to stop the generator, the output voltage is not zero. To obtain a zero output voltage, the circuit connects input Pin 5 of IC_1 to a negative 0.5V-dc voltage generator comprising an LF356N and associated components.

Because the signal at Pin 5 of IC_1 goes positive and negative, IC_4's switch requires a ±5V supply. The level for the command signal also has to swing positive and negative. MOS transistor Q_1 provides the level-shifting from 0 to 5V logic levels to ±5V, or 4016, logic levels. NAND gates IC_{3C} and IC_{3D} allow a fast drive for Q_1.

The circuit's operation consists of three timing periods: load N with trigger input inactive, down-count with trigger input high, and disabled (**Figure 3**). When the trigger input is inactive, one-shot IC_{5A} is inactive. The level at Point B in the circuit is low, and counter IC_2 is continuously loading the number N that you program using S_1. The level at D is high, and the counter can't run. The voltage at C is low. The circuit connects NAND gates IC_{3C} and IC_{3D} in parallel to provide more current to drive Q_1 faster during switching. C_2 is also necessary to drive Q_1 faster. These NAND gates invert the level at D, and Q_1 is on, driven with 0V through R_1. Thus, a 5V level is present at E, and the IC_4 switches are on. Potentiometer R_3 controls the voltage at G; the LF356N acts as a voltage follower. The 10Ω resistor, R_2, prevents oscillations during switching. C_1 and the MAX038 input represent the charge impedance. The four switches of IC_4 connect in parallel to present a lower resistance of 200Ω/4, or approximately 50Ω, of total resistance. The switches apply the voltage at G to Pin 5 of IC_1, which stops the internal oscillator. The levels at the signal output and at the sync output, F, depend on the voltage at Pin 5. The voltage at F needs to be 5V, and the voltage at Signal Out needs to be as close to 0V as possible. You need to carefully adjust R_3 to match these conditions. The output voltage is just over 0V when G is close to −0.5V.

When the trigger input goes high, one-shot IC_{5A} starts running. The voltage at B also goes high for 10 µsec. This delay must be longer than 16 periods of the output signal. The voltage at D now goes low and enables the counter. As before, IC_{3C} and IC_{3D} invert the level at D, and the 5V drive turns off Q_1. A −5V level is present at E, and IC_4 switches are off. The internal oscillator of IC_1 is now running, and a signal is present at the output, S. Each time the output signal is positive, the sync output at F is also positive. At the

Figure 3



A timing diagram that corresponds to three periods shows the precise sine-wave output at S.

TABLE 1–WAVEFORM-SHAPE SETTINGS

A_0	A_1	Waveform
X	1	Sine
0	0	Square
1	0	Triangle

end of each period, a positive-going edge appears at F. Each positive edge at F makes counter IC_2 count down by one.

When the circuit has produced N periods at S, the voltage at C goes high, which indicates end of count. The voltage at D goes high and disables the counter. IC_{3C} and IC_{3D} invert the voltage at D, and the resulting 0V drive turns on Q_1. A 5V level is present at E, and IC_4 switches are on. The internal oscillator of IC_1 stops, and the output signal at F returns to zero. Before returning to the original state, the signal at B should return to zero, which happens after the end of the delay that one-shot IC_{5A} produces.

All is now ready for another train of pulses. Using S_1, you can program the circuit for one to 15 pulses. The circuit can produce other signal shapes, depending on how you connect A_0 and A_1 of IC_1 (**Table 1**). You can also replace S_1 with a µC to produce any pattern of pulses.

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

High-voltage current-feedback amplifier is speedy

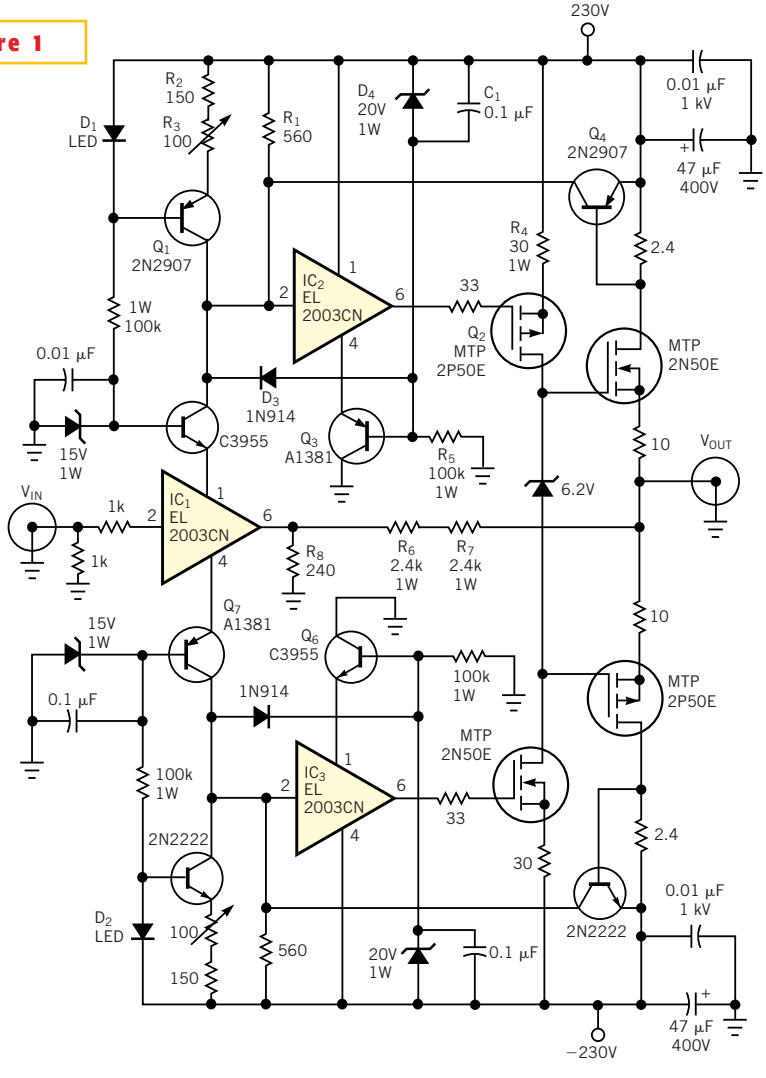
Joseph Ting, Institute of Atomic and Molecular Sciences of the Academia Sinica, Taipei, Taiwan

THE CIRCUIT IN FIGURE 1 powers a microparticle and nanoparticle ion trap through a 1-to-5-turns-ratio, high-voltage transformer. It also works successfully as a driver for a piezo-tube scanner and in a near-field scanning optical microscope. The circuit is robust and works with supplies ranging from ±50 to ±230V. The measured parameters at ±230V supply voltage are gain of 26-dB from dc to −3-dB point at 7 MHz; output swing of ±200V, rise and fall times of 70 nsec for an output step of 350V, slew rate of 4100V/µsec, and supply current of 56 mA.

The red LEDs, D_1 and D_2, in **Figure 1** provide a 1.8V drop; the LEDs are more rugged than precision IC voltage references. The current supply for IC_1 comes from R_1 and the source comprising D_1, R_2, R_3, and Q_1. R_3's trimmed value is such that Q_2's quiescent current is approximately 15 mA. You can determine this current by measuring the voltage drop across R_4. The same adjustment also controls the output-voltage offset. IC_2 is a unity-gain, high-current driver for Q_2. D_3 prevents IC_2's input from going more negative than its negative supply. Q_3, D_4, C_1, and R_5 provide the negative bias for IC_2. Q_4 is an output-current limiting switch. Q_4 starts to turn on at I_{OUT}=290 mA. You can replace the bipolar transistors C3955 (npn, Q_2 and Q_6) and A138 (pnp, Q_3 and Q_7) by equivalents as long as they have the following minimum specs: V_{CEO}≥250V; I_C≥100 mA, and f_T≥100 MHz.

You should mount all the power tran-

Figure 1



This high-voltage, current-feedback amplifier slews at 4100V/µsec.

sistors in individual finned heat sinks with an overhead 3-in. fan for cooling. The pc-board layout is not critical and needs no ground plane. However, you must use single-point grounding to minimize ringing. For the component values shown, the circuit is very stable and needs no compensation capacitors. **Figure 2** shows a large-signal response for a ±9V, 1-MHz square-wave input. This circuit has a fixed gain of 20. For higher gains, you can increase the values of R_6 and R_7. For lower values, it is better to insert an attenuator at the input, because smaller values of R_6 and R_7 may result in excessive dissipation. Do not change the value of R_8, because it is optimized for speed. Be cautious when measuring and using this circuit, because it harbors lethal voltages. The National Science Council of Taiwan sponsored this project.

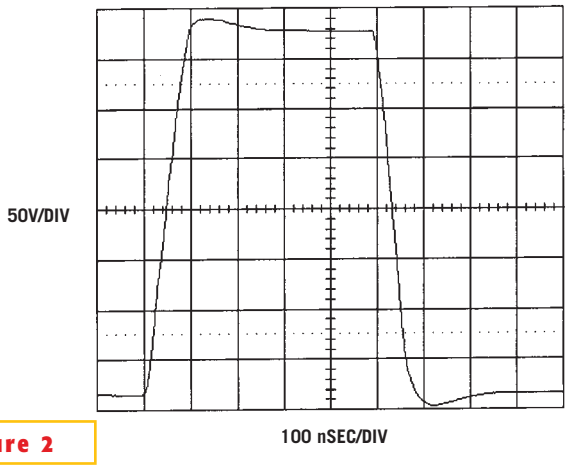


Figure 2

The circuit has a clean square-wave response with minimal overshoot and no ringing.

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

AC-power monitor uses remote sensing

Sanjay R Chendvankar, Tata Institute of Fundamental Research, Colaba, Mumbai, India

The detection circuit in the Design Idea "Circuit monitors ac-power loss" (*EDN*, Nov 24, 1999, pg 172) requires a physical connection with the mains to sense the power loss. The circuit in **Figure 1** senses the power loss through the radiated power-line signal. The battery-operated circuit has a quiescent-current drain of approximately 2 µA. The antenna, which is either a telescopic antenna or simply an approximately 2-ft-long wire, intercepts the radiated power-line signal. The CMOS inverters, IC_{1A} and IC_{1B}, amplify this weak signal and convert it into a digital signal. D_1 and C_1 generate a steady dc voltage at the input of IC_{1C}. D_1 prevents discharge of C_1 through the output of IC_{1B} when the square wave at this output periodically goes to a low level. Inverters IC_{1D}, IC_{1E}, and IC_{1F} connected in parallel enhance the current-sink capacity for sinking the piezo-buzzer current. When the ac mains is present, the output of IC_{1C} is low; hence, the levels of IC_{1D}, IC_{1E}, and IC_{1F} are high, and the buzzer is off. When the ac power fails, the output of IC_{1B} goes low; C_1 discharges through R_1; and IC_{1D}, IC_{1E}, and IC_{1F} go low. This level activates the piezo-buzzer and warns of ac-line failure. Switching off the battery power deactivates the buzzer. You can turn S_1 on after ac power resumes.

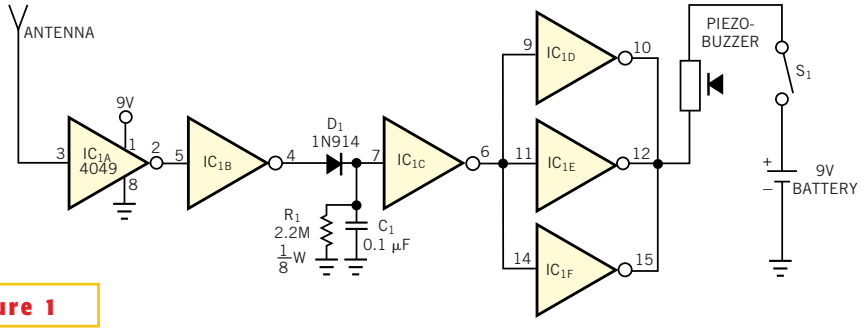


Figure 1

A low level at the outputs of IC_{1D}, IC_{1E}, and IC_{1F} activates the piezo-buzzer and warns of ac-line failure.

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

design ideas

Edited by Bill Travis and Anne Watson Swager

Circuit detects first event

Kelly Flaherty, National Semiconductor, San Mateo, CA

THE CIRCUIT IN **Figure 1** is a "first-event" indicator, like a game show's "who's first to answer" detector. It indicates which of the two momentary switches, S_1 or S_2, closes first by latching the corresponding channel, IC_A or IC_B, to a high state. As either of the outputs latches high and lights its respective LED, it locks out the other channel and prevents it from triggering. The other momentary switch, S_3, resets either of the latched outputs to its initial low (LED-off) state. At the initial condition, the positive input of each comparator is approximately at 0V, because both outputs are low. The negative inputs are at $V_{BAT}/11$, as set by voltage divider R_4 and R_5. In this initial condition, assume that S_1 is momentarily closed. The positive input of IC_A becomes $^5/_6(V_{BAT})$, as set by voltage divider R_{2A} and R_{3A}. Because $^5/_6(V_{BAT})$ is greater than VBAT/11, the output of IC_A goes high, and the positive input of IC_A latches its threshold to approximately $V_{BAT}/6$.

Correspondingly, the negative input of IC_B latches to approximately V_{BAT}, thus preventing S_2 from triggering IC_B's output high. S_3 resets (turns off) either of the active outputs by pulling the inverting inputs one diode drop below V_{BAT}. Both channels are then in their initial condition and ready to go again. The LMC6762 dual comparator is a good fit for this application, because it draws only 7-µA quiescent current, and it has rail-to-rail inputs and outputs. The comparator's sourcing capability allows it to easily drive an LED. **Figure 2** shows how you can cascade two first-event detectors to obtain more channels.

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

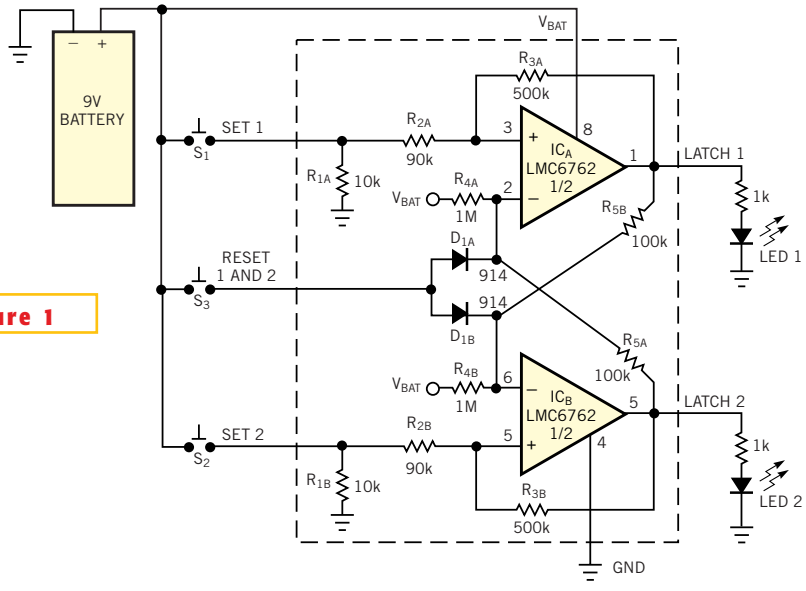


Figure 1

Which button did you push first? This circuit reveals the answer.

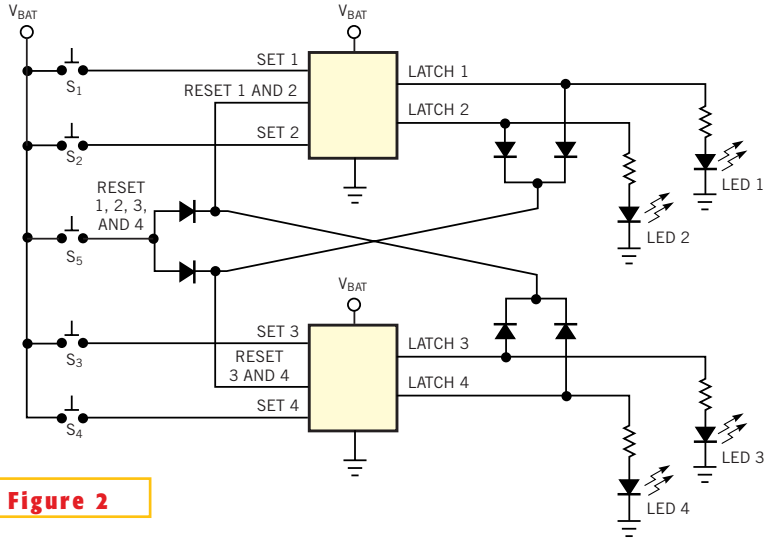


Figure 2

With the addition of six diodes, you can cascade two first-event detectors.

High-speed pulse generator has programmable levels

John Guy, Maxim Integrated Products, Sunnyvale, CA

Lilliputian dimensions associated with the submicron geometries of most digital and many analog processes result in much faster circuit operation. As ICs speed up, the rise and fall times of most pulse and function generators, which are typically 5 nsec, become inadequate for measuring time intervals lower than 20 nsec. You can overcome this limitation with analog comparators or advanced CMOS logic gates, which create faster digital edges. Their rise and fall times are fast enough, but the signal levels include ground and V_{CC} only.

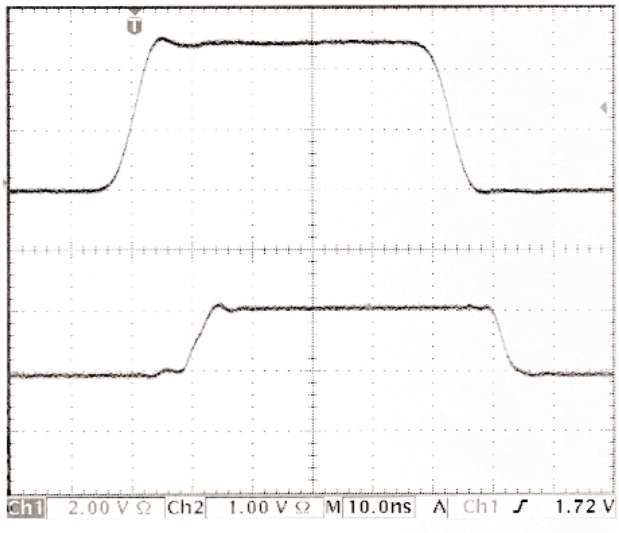


Figure 2

The input (lower) and output (upper) traces illustrate fast output transitions and settable output levels.

Designers have applied the submicron processes that high-speed digital circuits use to analog switches as well, so the turn-on and turn-off times for these switches also produce fast rise and fall times. What's more, an SPDT (single-pole, double-throw) switch can create pulses whose high and low levels are programmable.

A feature of analog switches that hinders their use as pulse generators is the intrinsic built-in delay—the break-before-make time—that guarantees that an SPDT switch does not short the two switched terminals together during a transition. Unfortunately, this delay and the switches' finite turn-on time also extends the rise and fall times. You can avoid this effect by adding a dynamic pullup and a dynamic pulldown to the circuit (**Figure 1**). A sufficiently low pullup and pulldown impedance can drastically improve the corresponding rise and fall times.

The input clock signal, Φ_1, controls an

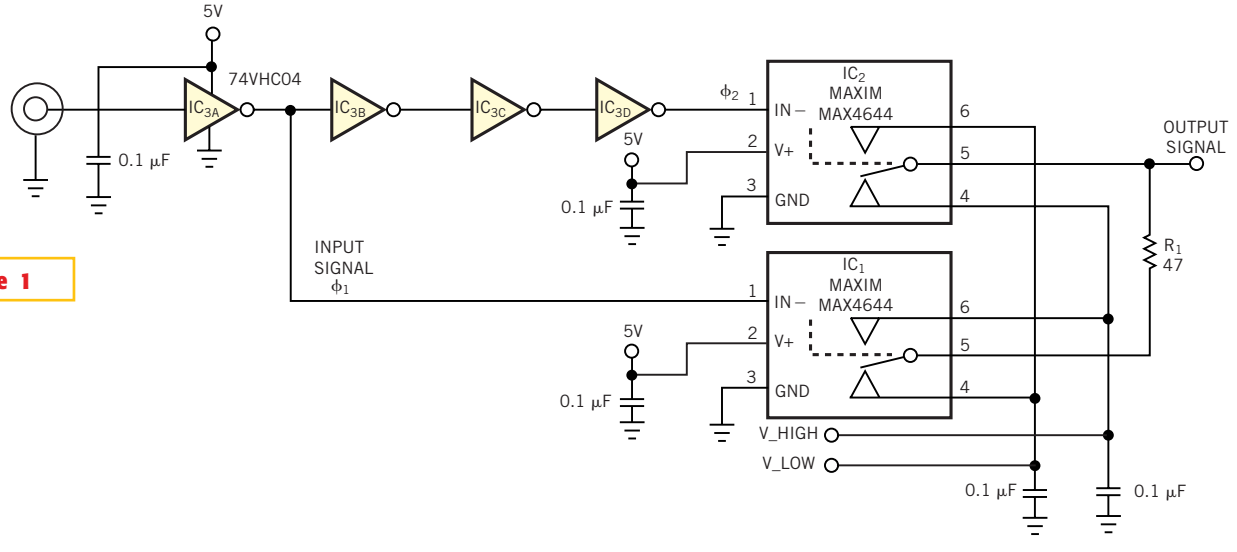


Figure 1

Analog switches provide dynamic pull-up and pull-down at the output of this pulse generator to ensure fast rise and fall times.

SPDT analog switch, IC_1, which the circuit configures as the pullup/pulldown driver. The input clock signal also drives a high-speed CMOS inverter, IC_3, to create a delayed clock signal, Φ_2. The delayed clock drives an SPDT analog switch, IC_2, which the circuit configures as the output driver.

Consider the steady-state condition in which Φ_1 is low and Φ_2 is high. IC_1's COM pin and IC_2's COM pin connect to V_LOW, and a rising edge on Φ_1 causes IC_1 to pull the output signal high. Because the series resistor, R_1, is large with respect to the MAX4644's on-resistance, or 47Ω versus 2.5Ω typical, the immediate effect on output voltage is minimal. However, when Φ_1 propagates through the inverter string, the falling edge of Φ_2 causes IC_2 to transition from V_LOW to V_HIGH. The presence of a low-impedance pullup, R_1, provides drive for the signal transition, and the closing of IC_2 quickly follows.

The input signal is 5V logic, and the output swings from 1V to 2V (**Figure 2**). You can set V_LOW and V_HIGH to any level within the supply range for IC_1 and IC_2. The circuit's quiescent current is essentially zero, with brief peaks only during the output transitions. Rise and fall times at the output are approximately 4 nsec, and the output impedance is 2.5Ω.

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

Low-battery indicator has high efficiency

Joe Neubauer, Maxim Integrated Products, Sunnyvale, CA

THE USUAL METHOD for implementing the low-battery warning featured in most battery-operated equipment is to illuminate an LED. However, the LED exacerbates the low-battery condition. You can greatly reduce the LED's power consumption by operating it at a low frequency and a low duty cycle. An existing LBO (low-battery output) like that found in dc/dc converters offers a convenient way to light the LED (**Figure 1**). IC_1 is a small, inexpensive comparator with shutdown capability, housed in a six-pin SC70 package. It remains in shutdown condition while the battery is at normal operating levels but asserts LBO when the battery voltage falls below a preset threshold. Active-high LBO is usable as shown, but an active-low warning, $\overline{LBO}$, requires the optional circuitry shown. IC_1 turns on, causing the LED to flash according to the following analysis: First, you want to keep the duty cycle low: $DC=t_{ON}/(t_{ON}+t_{OFF})$. You derive the on-time from the equation for time-varying voltage across a charging capacitor: $V(t)=V(1-e^{-t/RC})$, so $t_{ON}=-R_5Cln(_{TRIPHI}1-V_{TRIPHI}/V_{OUT})$. You then derive the off-time from the equation for time-varying voltage across a discharging capacitor: $V(t)=Ve^{-t/RC}$, so $t_{OFF}=-R_4Cln(V_{TRIPLO}/V_{OUT})$. Use Kirchoff's current laws to find the comparator's high and low trip levels: $V_{TRIPHI}=V_{OUT}[R_3(R_1+R_2)]/[R_3(R_1+R_2)+R_1R_2]$, and $V_{TRIPLO}=V_{OUT}[R_3R_2]/[R_3(R_1+R_2)+R_1R_2]$. Assuming a 2.5% duty cycle and assuming that the LBO trips the comparator on when the battery voltage equals 3V, the resulting trip levels are 1V for low and 2V for high. The standard component values corresponding to this performance are: $C_1=0.1\ \mu F$, $R_1=R_2=R_3=1\ M\Omega$, $R_4=3.6\ M\Omega$, and $R_5=91\ k\Omega$.

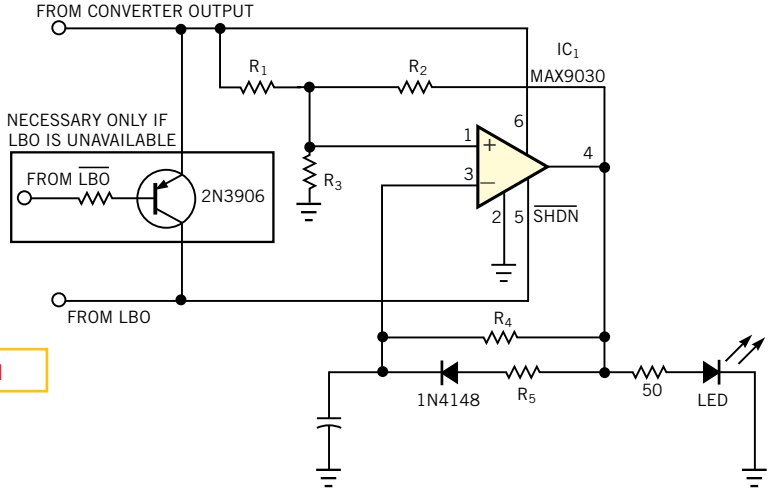


Figure 1

Operating the low-battery LED at low duty cycle saves power and extends battery life.

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

LCD-bias supply provides precise tracking

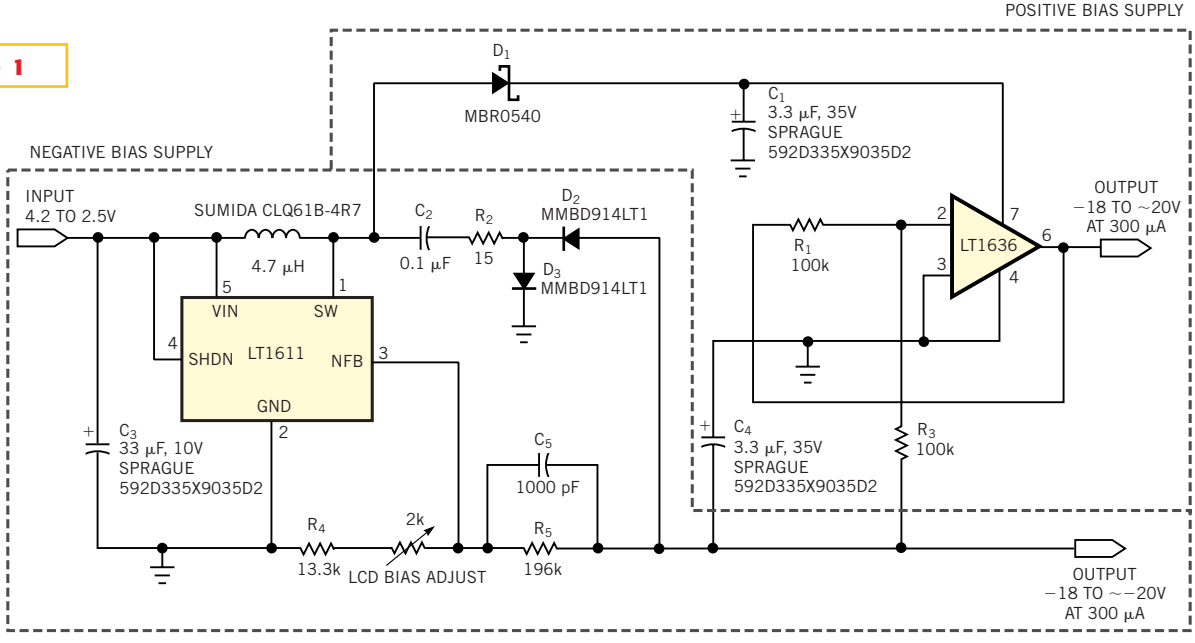
David Kim, Linear Technology Corp, Milpitas, CA

Small monochrome LCD systems often require split (dual)-bias supplies with precise voltage tracking to prevent plating of the LCD. The circuit in **Figure 1** provides ±18 to ±20V adjustable LCD bias voltages with 1% tracking accuracy. The circuit operates from a single 4.2 to 2.5V Li-ion cell for portable monochrome LCD applications. The circuit comprises two blocks: a negative-bias supply using the LT1611 inverting switching regulator and a positive-bias supply using the LT1636 rail-to-rail op amp. The LT1611 converts the Li-ion battery input voltage to a negative output voltage. The combination of 1.4-MHz switching frequency and a 36V internal switch results in small, low-profile circuit. LCD bias requires high voltage at low current. A charge pump consisting of C_2, C_4, D_2, and D_3 generates the negative output voltage. Some benefits of this circuit topology include zero output power during shutdown and low output ripple.

The LT1636 rail-to-rail op amp generates the positive LCD-bias output. The large capacitive-load capability, low quiescent current, and high-impedance input stage make the LT1636 suitable in this application. The LT1636 inverts the LT1611's output to provide the positive LCD-bias voltage. To meet the 1% tracking requirement, you should use a precision-resistor network, such as the 664 series from BI Technologies (www.bitechnologies.com), for R_1 and R_3. The unique input stage of the LT1636 allows you to generate the V_{CC} of the inverting op amp from the rectified switching waveform (using D_1 and C_1) of the LT1611 switching regulator. You adjust both LCD-bias supplies by varying the 2-kΩ potentiometer at the feedback node of the LT1611 regulator.

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

Figure 1



This LCD-bias supply provides better than 1% tracking of the positive and negative outputs.

Rolling-code generator uses flash µC

Wallace Ly, National Semiconductor Corp, Santa Clara, CA

Many security-alarm systems require the use of a random number. A computer program uses this random number to create a sequence of random numbers to prevent unwanted visitors from gaining entry into a protected facility. You can use a "rolling-code generator" to produce random numbers. To implement such a generator, you would typically need a microcontroller with external memory. Instead, you can use National Semiconductor's COP8SBR flash

μC with "virtual-EEPROM" technology. This technology allows you to use a section of flash memory as if it were EEPROM. Because this μC is a true-flash device, the maximum number of erase/write cycles is typically 100,000 cycles. The flow chart in **Figure 1** and the C code in **Listing 1** show the adaptation of a textbook LFSR (linear-finite shift register) to the COP8 flash μC.

An initial "seed" first drives the input. The seed then traverses several exclusive-OR stages. The routine then saves the result to a virtual-EEPROM location. This approach allows an embedded-system designer to easily create a highly secure system without incurring the cost of an external nonvolatile memory, such as a dedicated serial EEPROM. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases," then enter the Software Center to download the file for Design Idea #2704. You can find additional information about the COP8SBR and its virtual-E^2 feature at www.national.com/cop8.

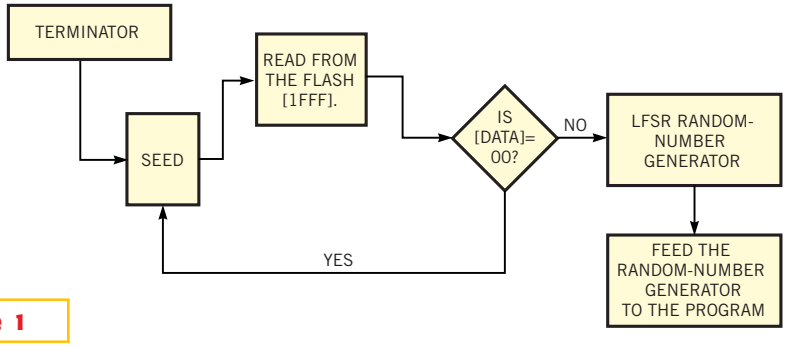


Figure 1

This random-number generator uses the virtual-E^2 feature of the COP8 μC.

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

LISTING 1—SOURCE CODE FOR A VIRTUAL-EEPROM BASED RANDOM-NUMBER GENERATOR

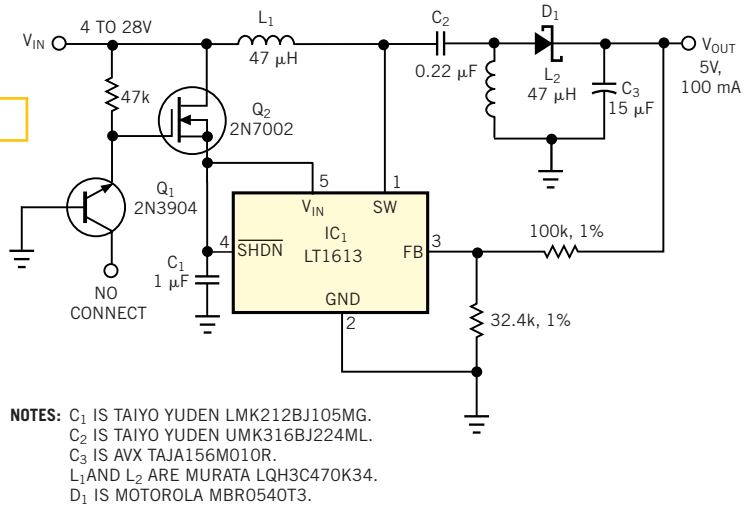
```
#include <8cbr.h>
#include <flash_op.h>


//*****************************************************************
//
// Description: The following is a random number generator,
//              its implementation is the use of a 16 bit
//              LFSR, by XORing the correct bits we arrive
//              at a pseudo random number
//
// Implementation: The pseudo number is created by using the
//                 feedback equation (for a 16 bit word)
//                 bit16=bit5 XOR bit4 XOR bit3 XOR bit0
//
//*****************************************************************

void random(){

        unsigned int seed_upper;  // The seed variables
        unsigned int seed_lower;

        // Seed the random number generator
        // Choose any random numbers for the MSB and
        // LSB of the initial random numbers

        readbf(0x1FFF);    // Do a read at the high location
        seed_upper=ISPRD;

        readbf(0x2000);    // Do a read at the low location
        seed_lower=ISPRD;

        if (!seed_upper && !seed_lower){      // If it is zero then continue

           }

        else {

          seed_upper=10;  // Otherwise give it a seed
          seed_lower=20;

          }

        bits temp1;             // Some Xoring Bits
        bits temp2;


        char flag;
        bits carry;

        temp1=(bits)seed_upper;             // Assign the upper & lower
        temp2=(bits)seed_lower;

        carry.0=temp2.5^temp2.4;     // Now do the Xoring
        carry.0=carry.0^temp2.3;
        carry.0=carry.0^temp2.0;
        if (temp2.0)
            flag=1;

            else flag=0;

        temp1=temp1>>1;                          // Bit shift left
        temp2=temp2>>1;


        if (flag)
            temp2.7=1;


        temp1.7=carry.0;                         // Use the Carry Bit


        seed_upper=temp1;
        seed_lower=temp2;

        page_erase(0x1FFF);          // Erase the location
        cwritebf(0x1FFF,seed_upper);       // Save the upper byte
        cwritebf(0x2000,seed_lower);       // Save the lower byte

} // end of the random routine


void main(){

        unsigned int i;             // A counter variable
        unsigned int random_arr[2]; // A random number array

                // Set the clock timer
                PGMTIM=0x7B;


                                        // copy it back to the random numbers
                readbf(0x1FFF);    // Read the result
                random_arr[0]=ISPRD;     // From the ISPRD register

                readbf(0x2000);
                random_arr[1]=ISPRD;     // Both the Upper and Lower
                                // Seed
        while (1){


             page_erase(0x1FFF);             // Page Erase

             cwritebf(0x1FFF,random_arr[0]);  // Write the byte back
             cwritebf(0x2000,random_arr[1]);  // Write the byte back


             // Get a number ramdomly generated through 100 iterations
             for (i=0;i<100;i++)          // Call the random number
                 {                        // generator a hundred times
                  random();
             }

                        // Application code goes over here

                }


} // end of main
```

SEPIC generates 5V at 100 mA

Dongyan Zhou, Linear Technology Corp, Milpitas, CA

Some applications require an input voltage higher than the breakdown voltage of the IC supply pin. In boost converters and SEPICs (single-ended primary-inductance converters), you can separate the V_{IN} pin of the IC from the input inductor and use a simple zener regulator to generate the supply voltage for the IC. **Figure 1** shows a SEPIC that takes a 4 to 28V input and generates 5V at 100 mA.

Figure 1



NOTES: C_1 IS TAIYO YUDEN LMK212BJ105MG.
C_2 IS TAIYO YUDEN UMK316BJ224ML.
C_3 IS AVX TAJA156M010R.
L_1AND L_2 ARE MURATA LQH3C470K34.
D_1 IS MOTOROLA MBR0540T3.

Q_1 stands in for a zener diode in this SEPIC with a wide input-voltage range.

In this application, Q_1 and Q_2 generate the supply voltage for IC_1 because the supply voltage exceeds IC_1's maximum input voltage. The circuit uses Q_1 in place of a zener diode to save cost. The emitter-to-base breakdown voltage gives a stable 6V reference. The follower, Q_2, provides the supply voltage for the IC. This circuit demonstrates an inexpensive way to extend the input range of the IC.

This SEPIC can step up or step down the input voltage. Because the flying capacitor, C_2, breaks the input-to-output dc path, the output disconnects from the input when you shut down the device, inhibiting any possible load current in shutdown mode, which is important for portable applications and which prevents the input voltage from appearing at the output.

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

Look-up table helps bit flipping

Brad Bierschenk, High End Systems, Austin, TX

In certain instances in embedded software, a programmer needs to flip the order of the bits in a byte so that B_7 to B_0 become B_0 to B_7. Bit flipping is useful, for example, when a synchronous serial port does not allow programmatic selection of bit order, such as MSB first or LSB first, for its shift register. You need a software method to translate data if the processor sends data to a receiving device that expects a certain bit order, but the serial port can provide only the other bit order.

One solution is to provide a look-up table in ROM in which the value of each byte in the table is its offset into the table but with a reversed bit order. 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. The program needs only to load the value to be translated into a register that you can use as an offset, index the look-up table, and load the corresponding value from the index+offset location (**Listing 1**). Using the Philips 8×C51 architecture as an example, you can use the 16-bit DPTR (data pointer) plus an 8-bit offset in accumulator (A) to load the accumulator with a byte value.

LISTING 1—BIT-FLIPPING CODE SEGMENT

```
CodeSegment:
        ; 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
```

This approach is dynamically more efficient than rotating a byte location through carry bits, but it is not the most statically efficient approach because the look-up table requires 256 bytes of ROM. You can download the inversion 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.

design ideas

Edited by Bill Travis and Anne Watson Swager

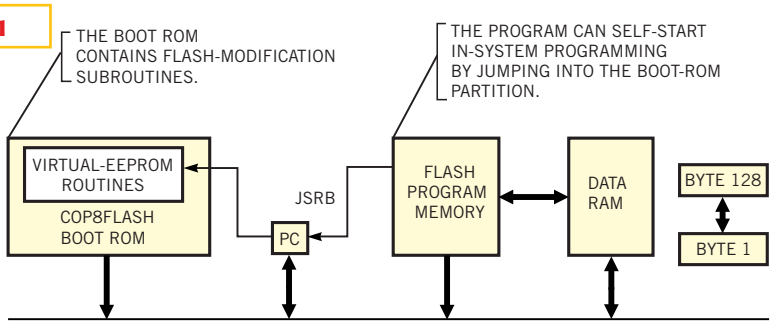
µC eliminates offboard EEPROM with "virtual-EEPROM" routines

Wallace Ly, National Semiconductor Corp, Santa Clara, CA

COP8flash µCs let you allocate some of their internal flash memory as a nonvolatile storage space. This line of flash µCs accomplishes this task by accessing some memory functions built into the boot ROM. In addition to some of these memory functions, the COP8flash µC can initiate and control in-system programming under software. You can use "virtual EEPROM," although not truly EEPROM, to mimic the behavior of the storage space of an offboard EEPROM. Note that the flash inside the COP8flash µC is not just renamed E^2. This fact has significant implications because the flash µC is rated for 100,000 erase/write cycles and 100-year data retention.

Figure 1 depicts how you can allocate a 128-byte virtual EEPROM. Although users may not write over the same byte twice, they may modify RAM contents. Additionally, you can "shadow" the flash µC with RAM. If users want to permanently save the contents of the virtual EEPROM, then they may execute a "flush" to dump the contents of the modified RAM into the flash µC.

Figure 1

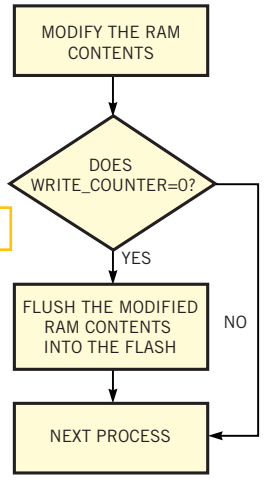


This example of a 128-byte virtual EEPROM shows how you can mimic the behavior of the storage space of an offboard EEPROM.

A system designer may bring up the issue that the "dumping" of the RAM into the flash µC may take too long. However, the opposite is true if you take the following into consideration: A typical write to the flash µC takes a few microseconds, and a page erase takes 8 msec, independently of the µC system clock. So, the dumping process may take less than 10 msec. However, a designer is hard-pressed to find an offboard serial EEPROM that has less than 100 msec of access time.

Figure 2 shows how to visualize the sample virtual-EEPROM code. As you shadow the flash µC, you write bytes to the RAM, and a WRITE_COUNTER decrements. When the WRITE_COUNTER equals 0, the routine flushes the contents of the RAM. The flash µC then saves the contents of the RAM.

Figure 2



When the WRITE_COUNTER equals 0, the routine flushes the contents of the RAM.

A system designer can set the WRITE_COUNTER to 0 to immediately copy

the contents of the RAM into the flash µC. You can download the Virtual EEPROM C code 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 #2688.

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

Medical timer warns when pills are due

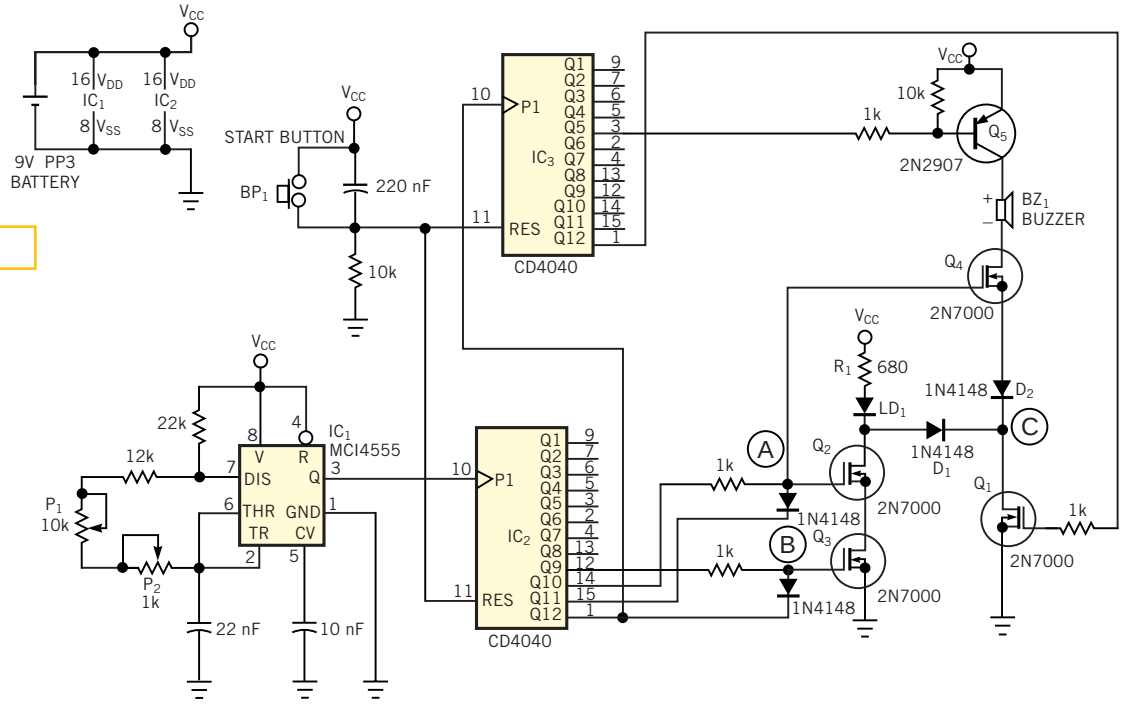
JM Terrade, Clermont-Ferrand, France

Some people need to take medication at precise, regular intervals. When you're in a hospital, a medical staff is present to ensure that you take your medication on time. But when you're taking medication at home, you must frequently look at the clock—a clear annoyance. When my wife was pregnant, she needed to take medication every two hours from 8 am to 10 pm. To help her time the two-hour intervals, I built the battery-powered circuit in **Figure 1**. The circuit derives its power from a 9V PP3 battery. All the ICs are CMOS-based; the low power consumption of the circuit allows one-week autonomy. When you

TABLE 1—TIMING DETAILS FOR CIRCUIT IN FIGURE 1

	IC_2		IC_3	
	Positive edge	Period	Positive edge	Period
Q1	0.858 msec	1.17 msec	3.51 sec	7.03 sec
Q2	1.717 msec	3.43 msec	7.03 sec	14.06 sec
Q3	3.433 msec	6.866 msec	14.06 sec	28.12 sec
Q4	6.866 msec	13.73 msec	28.12 sec	56.25 sec
Q5	13.73 msec	27.5 msec	56.25 sec	112.5 sec
Q6	27.46 msec	55 msec	112.5 sec	225 sec
Q7	54.93 msec	110 msec	225 sec	7.5 minutes
Q8	109.86 msec	220 msec	7.5 minutes	15 minutes
Q9	219.7 msec	439 msec	15 minutes	30 minutes
Q10	439 msec	878 msec	30 minutes	One hour
Q11	879 msec	1.758 sec	1 hour	Two hour
Q12	1.758 sec	3.51 sec	2 hour	Four hour

Figure 1



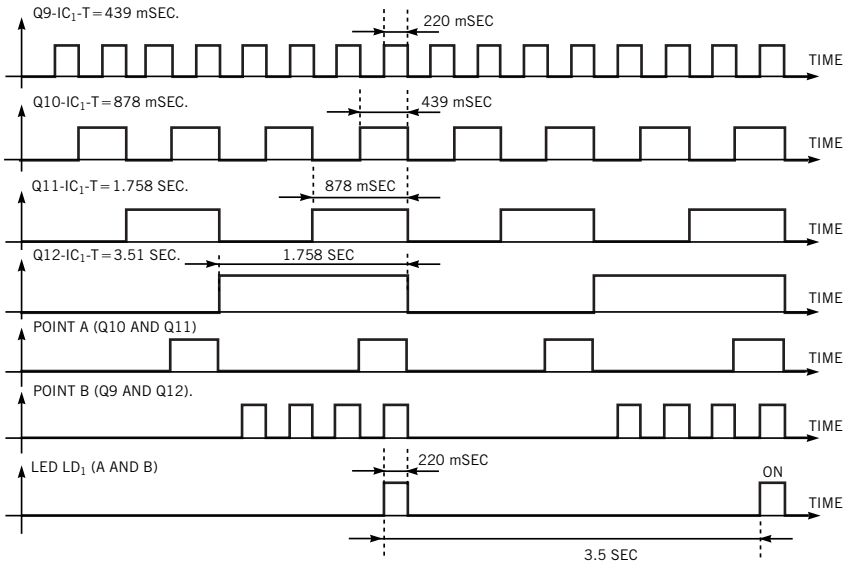
The buzzer in this circuit sounds at precise two-hour intervals.

press the start button, BP_1, a two-hour delay commences. During this delay, LED LD_1 flashes to indicate the delay is in progress and the battery voltage is satisfactory. After the two-hour delay elapses, the buzzer, BZ_1, emits short beeps for one minute If you don't press the start button again, LD_1 stays lit continuously to indicate that the delay has elapsed. Pressing the start button initiates a new two-hour delay cycle.

IC_1, a 555 timer, operates as an astable multivibrator and produces a rectangular waveform at its output. The period should be equal to 858.3 µsec. P_1 and P_2 permit gross and fine adjustments of the period. For the two-hour delay, P_1 yields a ±20-minute adjustment; P_2 produces a ±2-minute adjustment. For a precise adjustment, observe the Q7 output of IC_2 and trim to obtain a period of 110 msec. IC_2 is a 2^{12} divider whose Q12 output has a 3.51-second period. This output connects to IC_3's clock input. IC_3 then yields a 14,400-second period, or four hours. The circuit detects the positive edge of Q12, which occurs after two hours. Changing the connection to IC_3 to another output or trimming P_1 and P_2 allows you to set different delays. **Table 1** shows the timing details for the various outputs of IC_2 and IC_3.

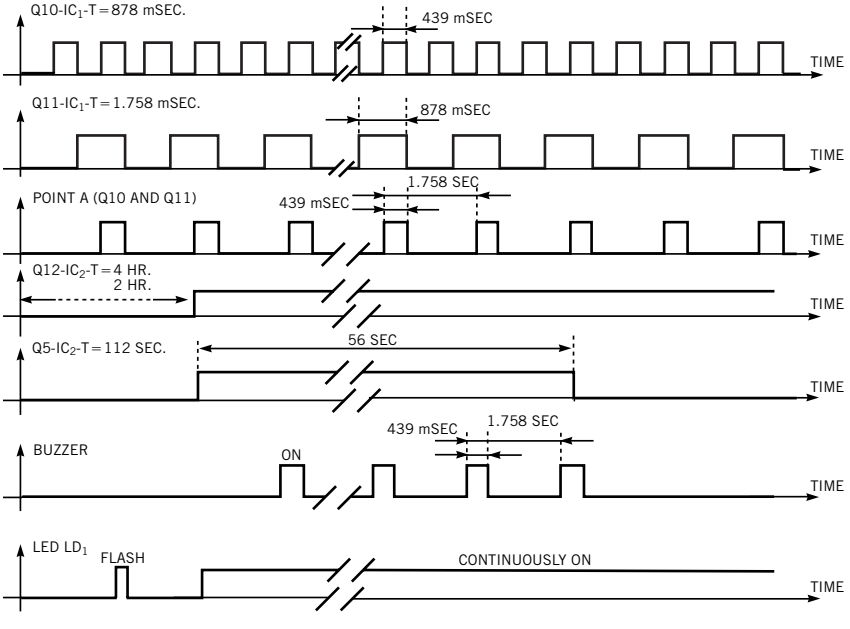
Figure 2 shows timing details for the first two hours after asserting start for the circuit in **Figure 1**. Q12 of IC_3 is at a low level, and transistor Q_1 is off. Thus, D_1 and D_2 cannot conduct. To light LD_1, Q_2 and Q_3 must be on, which is the case when Q9 to Q12 of IC_2 are at a high level. This high-level condition occurs for 220 msec every 3.5 seconds. This time period might seem short, but R_1's low value and the choice of a high-luminosity LED makes LD_1's flashing clearly visible. **Figure 3** shows timing details after the two hour time-out delay. Q12 of IC_3 assumes a high level, and Q_1 grounds Point C. As a result, D_1 conducts, and LD_1 stays continuously on, indicating that the delay has terminated. D_2 conducts when Q_4 and Q_5 are on, and the buzzer sounds. These transistors turn on when Q5 of IC_3 is at a low level and Q10 and Q12 of IC_2 are at a high level. These conditions occur for 439 msec every 1.8 seconds, for a total duration of 56 seconds. This time period might seem short, but it is long enough to create an audible beep. The start button restarts the cycle.

Figure 2



These waveforms show what elapses in the circuit of Figure 1 during the two-hour time-out.

Figure 3



The buzzer sounds, and the LED stays on after the two-hour time-out.

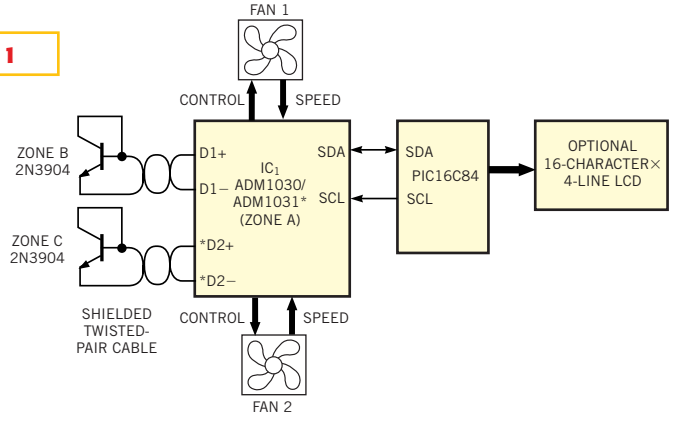
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Temperature monitor and fan controller reduce fan noise

David Hanrahan, Analog Devices Inc, Limerick, Ireland

THE SCHEME IN **Figure 1** reduces system acoustic noise by running system fans at their optimum speeds for a given temperature. IC_1 combines ±1°C-accurate temperature measurement of three temperatures with automatic fan-speed control of two channels. A two-wire serial interface allows you to oversee critical temperature and fan-speed data. NPN transistors, such as the 2N3904, can measure temperatures in remote locations. The microcontroller interface allows you to connect an LCD to display all monitored parameters.

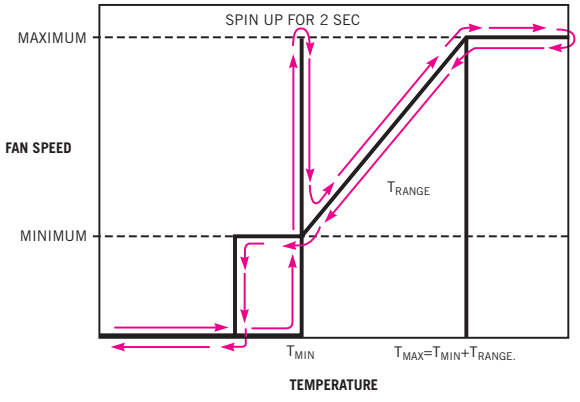
Figure 1



A scheme with multiple temperature-measurement and fan-control channels can run system fans at the optimum speeds for a given temperature.

You program IC_1's automatic-fan-speed-control function using T_{MIN} and T_{RANGE} (**Figure 2**). T_{MIN} is the temperature at which the fan automatically turns on and runs at minimum speed. T_{RANGE} is the temperature-to-fan speed-control slope, with options of 5, 10, 20, 40, and 80°C. Choosing one of the T_{RANGE} options allows you to define how the fan reacts to temperature variation.

Figure 2



The values of T_{MIN} and T_{RANGE} set the parameters of the automatic-fan-speed control loop.

An example of programming IC_1 follows. Setting Configuration Register 1 (Reg 0x00) to 0x99 starts the device in automatic-fan-speed-control mode, with the $\overline{\text{FANFAULT}}$ function enabled. A setting of Remote Temp 1 T_{MIN}/T_{RANGE} (Reg 0x25) = 0x63 sets T_{MIN} for fan 1 to 48°C and T_{RANGE} to 40°C. The Fan reaches full speed at 88°C. A setting of Remote Temp 2 T_{MIN}/T_{RANGE} (Reg 0x26) = 0x62 results in a T_{MIN} for Fan 1 of 48°C and a T_{RANGE} of 20°C. The fan reaches full speed at 68°C.

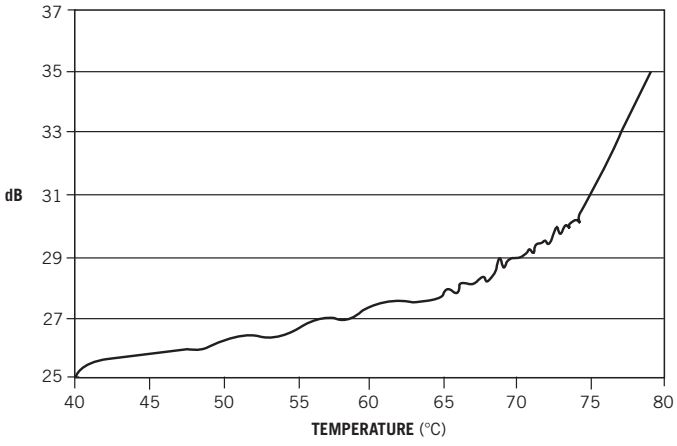
The fan runs at low speed until the system requires a higher level of cooling. The fan speed responds automatically to temperature variation, reducing acoustic noise (**Figure 3**). You can program additional features, such as fan spin-up time and minimum fan speed. The automatic-fan-speed-control mode allows flexibility over which temperature channel controls each fan. You can also decide that the maximum speed calculated for all temperature channels controls the fans.

Figure 4 shows the circuit diagram for the temperature-monitor/fan-control circuit. The PIC16C84 writes the configuration settings to IC_1 and optionally drives an LCD. The µC bit-bangs pins 2 and 3 to provide serial clock and data for IC_1. The µC reads and displays all temperatures and fan speeds. The LEDs that connect to the $\overline{\text{THERM}}$ and $\overline{\text{FANFAULT}}$ outputs illuminate whenever an overtemperature condition occurs or a fan fails. The $\overline{\text{THERM}}$ output is a fail-safe output that goes low if a preprogrammed overtemperature $\overline{\text{THERM}}$ limit is exceeded. In the event of an overtemperature condition, both fans automatically run at full speed. If some external device pulls the $\overline{\text{THERM}}$ pin low, the fans also run at full speed. The $\overline{\text{FANFAULT}}$ pin signals catastrophic fan failure. If one fan fails, the second fan automatically spins at full speed to compensate for the loss

of airflow. Should the failing fan recover or be replaced, both fans automatically return to their normal operating speeds.

You can extend this idea for multiple fans with redundant cooling using two ICs to monitor six temperature zones (**Figure 5**). This scheme cross-connects the $\overline{\text{THERM}}$ and $\overline{\text{FANFAULT}}$ signals. The same pin drives fans A_1 and A_2, which will run at the same speed. Likewise, fans B_1 and B_2 connect in parallel to a common FET. Fans C and D are redundant coolers that the system uses only if it gets excessively hot or a fan fails. If either fan A_1 or A_2 fails, fans B_1 and B_2 automatically run at full speed. The $\overline{\text{FANFAULT}}$ output of IC_1 asserts low, pulling $\overline{\text{THERM}}$ of IC_2 low,

Figure 3



Automatic-fan-speed control reduces acoustic noise.

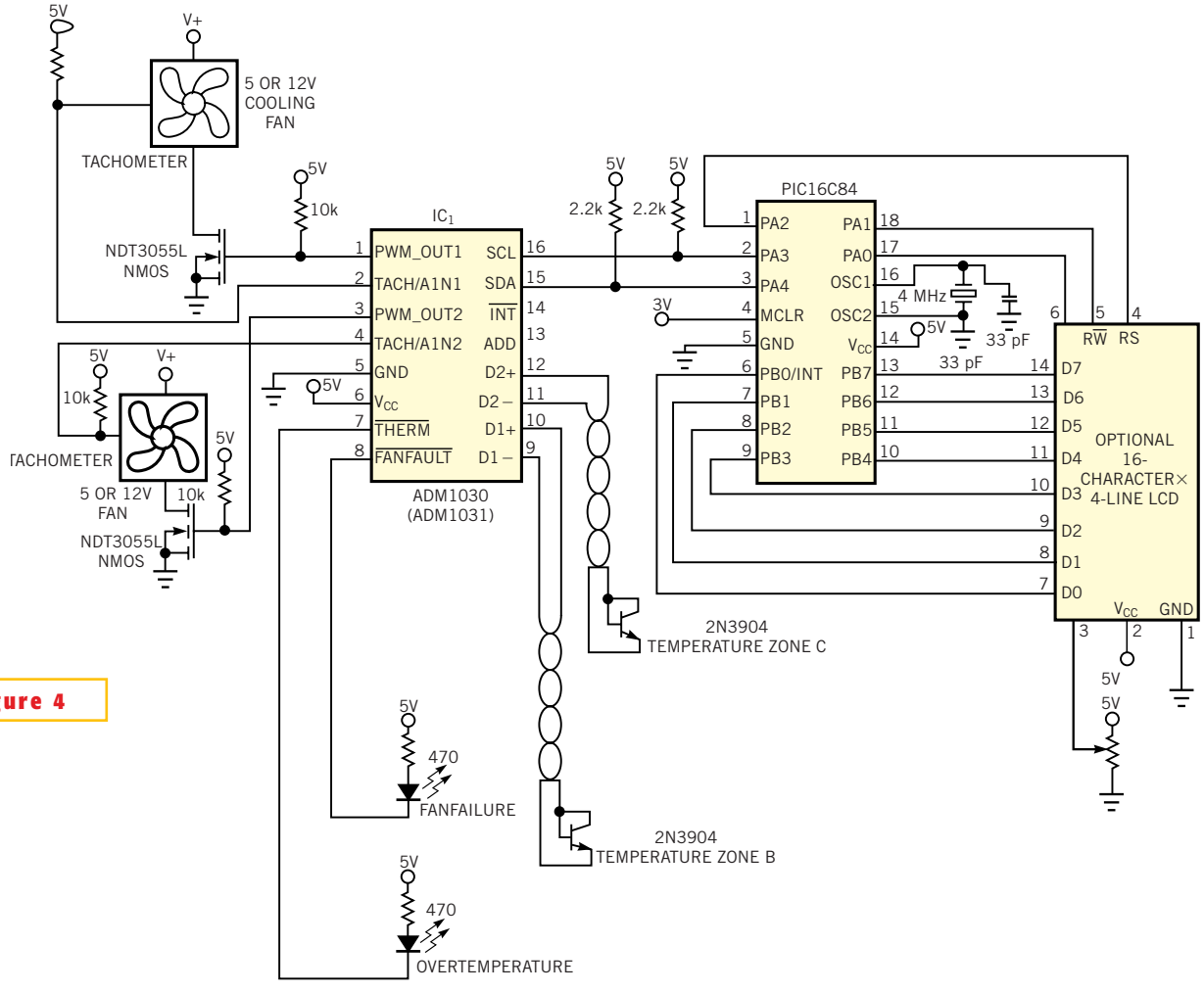


Figure 4

The μC bit-bangs pins 2 and 3 to provide serial clock and data for IC_1 and reads temperatures and fan speeds.

which causes redundant fans C and D to run at full speed. If either fan B_1 or B_2 fails, fans A_1 and A_2 automatically run at full speed. The $\overline{\text{FANFAULT}}$ output of IC_2 pulls $\overline{\text{THERM}}$ of IC_1 low, causing fans A_1 and A_2 to run at full speed. Fans C and D also automatically run at full speed. If the faulty fan is replaced, all fans return to their normal operating speeds. $\overline{\text{FANFAULT}}$ signals alert the system to fan failure.

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

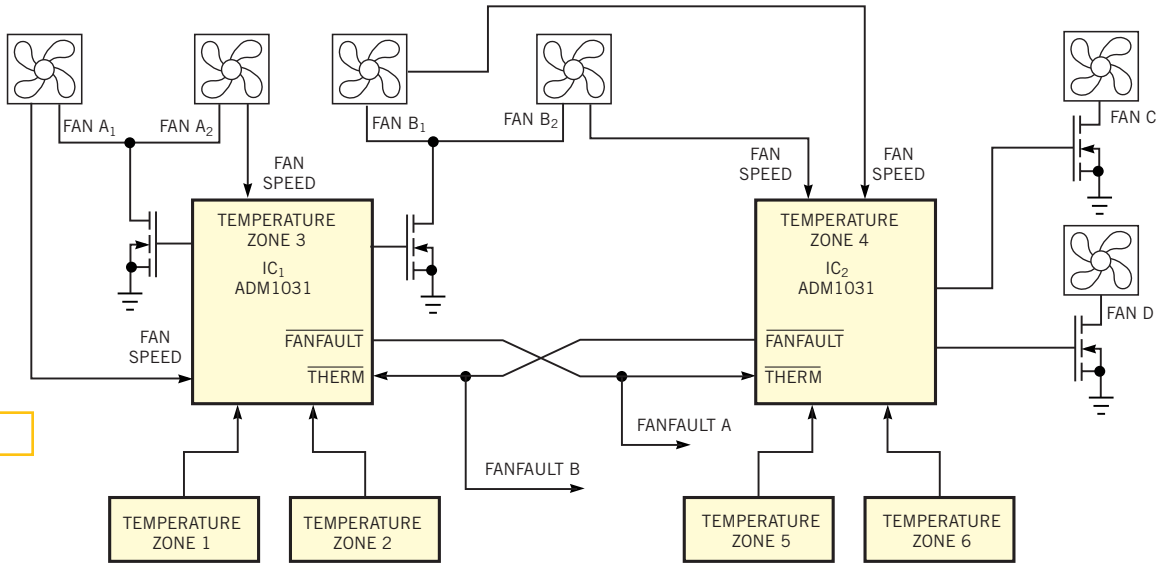


Figure 5

Using two ICs provides for monitoring six temperature zones.

Circuit forms random-bit-sequence generator

Przemyslaw Krehlik and Lukasz Sliwczynski,
University of Mining and Metallurgy, Krakow, Poland

A RANDOM-BIT-SEQUENCE generator is basic equipment for prototyping and testing any data-transmission system. You use such a generator when measuring BER (bit-error rate) and pattern-dependent effects in a transmission system. Such effects can include baseline wander, pattern-dependent data jitter, and recovered-clock jitter. Most sequence generators yield a PRBS (pseudorandom bit sequence) from a shift register with appropriate feedback. Thus, the sequence has limited length, and the generator continuously repeats the same pattern. The generator in **Figure 1** overcomes these limitations by using random noise to form the outgoing data stream. The circuit uses the ECLinPS logic family (**Reference 1**). The MC10EL16 liner receiver converts the incoming noise to a digital signal. Next, a rising clock edge of the first MC10EL31 flip-flop samples this random signal. Ideally, this flip-flop should provide a random bit sequence.

Unfortunately, when the data at the flip-flop's input changes simultaneously with the clock's rising edge, the flip-flop may fall into a metastable state. Thus, the resulting output state is indeterminate, and a significantly extended propagation delay may result, producing a jitter in the generated bit sequence (**Reference 2**). The second MC10EL31 flip-flop eliminates the jitter problem. We tested the generator with clock frequencies to 1 GHz and observed no anomalies in the output eye pattern or frequency spectrum. Note that the ECLinPS devices are ultrafast ICs, so you need to exercise special care in your pc-board design. You should terminate the generator's inputs and outputs with 50Ω, keep all connections short, and decouple all ICs with local capacitors. You can use a circuit from **Reference 3** for a noise source. The voltage of the noise source should be 100 mV to 1V rms, and

its frequency spectrum should be at least equal to the clock frequency.

References

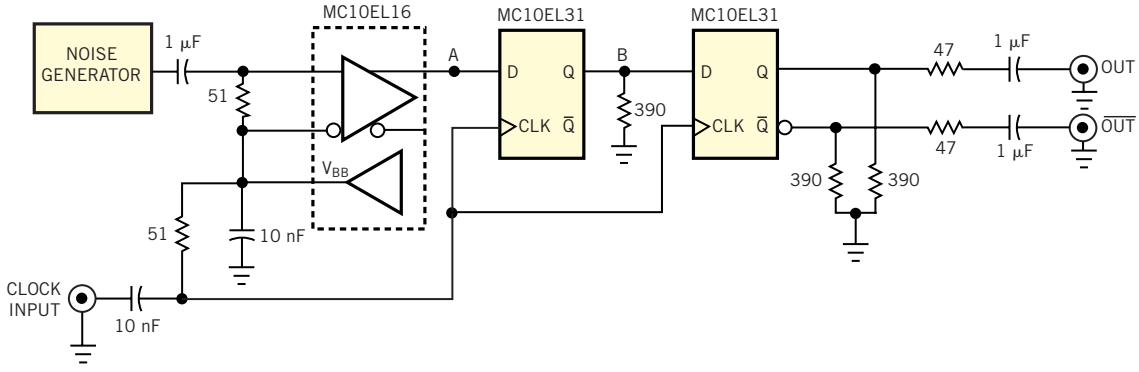
1. High Performance ECL Data, Motorola, 1995.

2. Metastability and the ECLinPS Family, Motorola Application Note AN1504.

3. Sliwczynski, Lukasw, "Zener diode and MMICs produce true broadband noise," *EDN*, Oct 14, 1999, pg 158.

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

Figure 1



This generator produces truly random bit sequences at frequencies to 1 GHz.

design ideas

Edited by Bill Travis and Anne Watson Swager

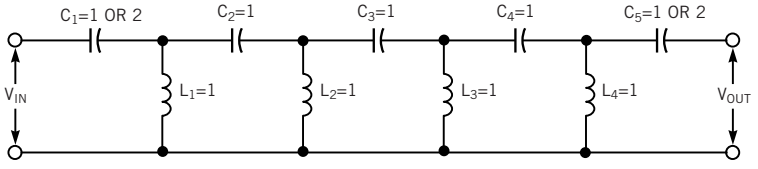
Highpass filters use modified equal-element design

Richard Kurzrok, Queens Village, NY

USING A MODIFIED equal-element design for a lumped-circuit lowpass filter has several advantages over the well-known equal-element design (**references 1** and **2**). The modified design exhibits superior passband performance with only modest degradation of stopband selectivity. Moreover, the modified design is simple and easy to manufacture. You can extend the modified equal-element design to highpass LC filters. Both equal-element and modified equal-element filters use the normalized highpass prototype (**Figure 1**). For the equal-element filter, the normalized value of the outside capacitors C_1 and C_5 is 1; for the modified equal-element filter, it's 2. The design reference frequency, at normalized frequency x=1.0, is not the 3-dB cutoff frequency for the filters in **Figure 1**. The 3-dB cutoff frequency occurs close to x=1.9. In contrast, in Butterworth and Chebyshev filters, x can equal 1.0 at the 3-dB cutoff frequencies. You use this different normalization method to calculate the values of the circuit elements.

We designed, assembled, and tested two nine-pole, modified equal-element filters. The filters used vector boards in die-cast aluminum boxes with BNCs. All

Figure 1



Doubling the values of the outside capacitors in an equal-element filter improves passband performance.

TABLE 1–MEASURED RESPONSE FOR 3.56-MHz MODIFIED HIGHPASS FILTER

Frequency (MHz)	Insertion loss (dB)	Frequency (MHz)	Insertion loss (dB)
2.9	39	6	0.4
2.95	34	6.5	0.25
3.1	29.8	7	0.2
3.3	19.8	8	0.2
3.5	9.8	10	0.1
3.6	4.5	15	0.15
3.7	1.2	20	0.1
3.8	1	25	0.15
3.9	1.7	30	0.2
4	2	40	0.2
4.2	1.7	50	0.25
4.5	0.8	60	0.35
5	0.25	80	0.2
5.5	0.4	100	0.4

TABLE 2–MEASURED RESPONSE FOR 11.17-MHz MODIFIED HIGHPASS FILTER

Frequency (MHz)	Insertion loss (dB)	Frequency (MHz)	Insertion loss (dB)
9	37.8	13	1.4
9.5	30	13.5	0.85
10	22	14	0.4
10.5	15	14.5	0.25
11	7	15	0.25
11.2	2.8	17.5	0.4
11.4	1	19	0.45
11.6	0.8	20	0.4
11.8	1.2	25	0.3
12	1.5	30	0.2
12.2	1.8	40	0.2
12.4	2	50	0.2
12.6	1.8	60 to 100	Less than 0.6
12.8	1.6		

capacitors were ±5% polypropylene units. We selected the filter design frequencies based on available capacitor values. The first filter had a reference frequency of 6.773 MHz. Assuming a ratio of 1.902-to-1, this figure corresponds to a cutoff frequency of 3.561 MHz. After denormalizing the filter to actual circuit values (as in **Reference 1**), we determined all inductor values, L_1 through L_4, to be 1.175 µH. The interior filter capacitors, C_2, C_3, and C_4, are equal to 470 pF. The filter's input and output capacitors, C_1 and C_5, are then equal to 940 pF. To obtain this value, we connected standard 820- and 120-pF capacitors in parallel. All the inductors used 15 turns of number 26 magnet wire wound on Micro Metals T37-2 toroids. **Table 1** shows the measured amplitude response.

The second filter had a reference frequency of 21.22 MHz. Assuming a ratio of 1.902-to-1, this figure corresponds to a cutoff frequency of 11.168 MHz. After denormalizing the filter to actual circuit values (as in **Reference 1**), we determined all inductor values, L_1 through L_4, to be 0.375 µH. The interior filter capacitors, C_2, C_3, and C_4, are equal to 150 pF. The filter's input and output capacitors, C_1 and C_5, are then equal to 300 pF. To obtain this value, we connected two standard 150-pF capacitors in parallel. All the inductors used 10 turns of number 26 magnet wire wound on Micro Metals T25-6 toroids. **Table 2** shows the measured amplitude response. Assuming inductor unloaded Q of approximately 100, the measured data shows good correlation with calculated values. You can cascade the modified equal-element highpass filter with a similar lowpass filter to obtain a bandpass filter of high bandwidth. This technique provides an alternative to using image parameters (**Reference 3**).

References

1. Kurzrok, Richard, "Equal-element filter improves passband performance," *EDN*, March 15, 2001, pg 123.
2. Taub, JJ, "Design of Minimum Loss Bandpass Filters," *Microwave Journal*, November 1963, pg 67.
3. Kurzrok, Richard, "Wideband filter uses image parameters," *EDN*, Oct 26, 2000, pg 174.

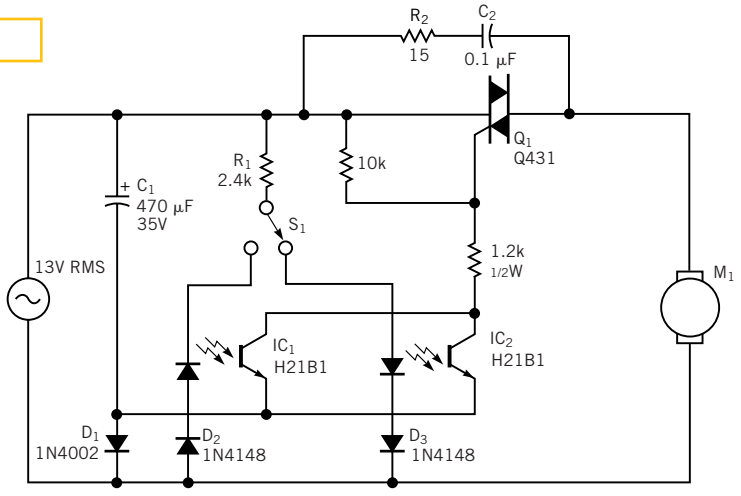
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Position detectors provide motor-control logic

Steve Pomeroy and Russell Hedges, Elgar Corp, San Diego, CA

In the circuit of **Figure 1**, assume that a brush-type dc motor must drive a load back and forth between two endpoints on a lead screw. Optical sensors determine end of travel, and an SPDT switch selects to which end to send the load. The sensors themselves supply all the necessary directional logic, and a triac powers the motor with the necessary polarity of half-wave pulses from the 13.5V-ac input. Starting with the load parked at the south end, when you set switch S_1 to north, the ac input connects to the LED in the north-side sensor, IC_1, through current-limiting resistor R_1 and reverse-polarity-protection diode D_2. The phototransistor output from IC_1 then supplies firing pulses to the gate of triac Q_1 during the northbound half-cycle, and the load proceeds toward that detector.

Figure 1



The optocouplers provide both position-control logic and triac gate drive to the motor.

Similarly, IC_2 drives the motor during the other half-cycle to push the load south, and stops when it reaches the south endpoint. This scheme works even when you change the direction switch while the load is in motion. Power for the gate drives comes from the ac input and the half-wave rectifier comprising D_1 and C_1. You might need snubber R_2 and C_2 if the motor's inductance causes spurious firings of Q_1 during undesired half-cycles. The motor's stall current is approximately 2A. You can easily scale the design for larger or smaller motors.

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

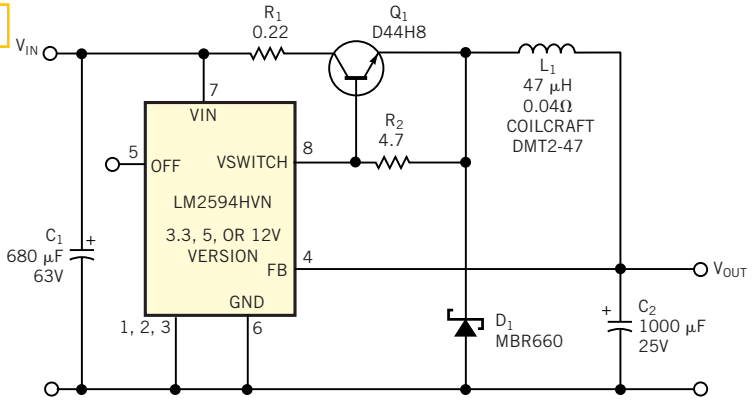
Bipolar transistor boosts switcher's current by 12 times

Wayne Rewinkel, National Semiconductor Corp, Santa Clara, CA

THE CIRCUIT IN **Figure 1** uses a minimal number of external parts to raise the maximum output current of a 0.5A buck switching-regulator IC to more than 6A. The circuit accommodates input voltages of 15 to 60V and delivers output voltages of 3.3, 5, or 12V, depending on your choice of IC. **Figure 2** provides a graph of conversion efficiency for the three standard output voltages, plotted over a range of input voltages extending to 60V. The circuit is useful in applications requiring higher input voltage, higher current, or both than is available from standard ICs. The LM2594HVN is a buck regulator that switches an internal 0.5A device at 150 kHz. This current suffices to feed the base of Q_1 and the bias resistor, R_2. The function of R_2 is to quickly turn off Q_1, a fast npn switch with a beta greater than 10 at 6A. The purpose of R_1 may not be obvious without some knowledge of the internal workings of the 2594. Its value is such to produce sufficient voltage drop at peak current so Q_1 begins to saturate. The saturation causes Q_1's beta to drop, and, as the transistor's base current rises to more than 0.5A, the 2594 drops into its pulse-by-pulse limited-protection mode, followed by a reduction in clock frequency if the overload is severe.

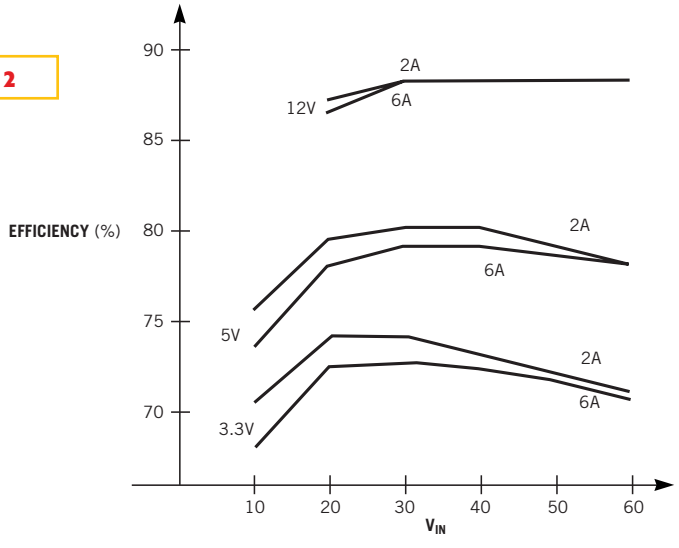
This design example uses through-hole components, because low-ESR capacitors and inductors in through-hole form are inexpensive and easy to find. Worst-case line and load conditions cause Q_1 and D_1 to dissipate 3W each, so you must choose a heat-sink size to keep the temperature rise within acceptable limits. A heat-sink rating of 6 to 7°C/W can accommodate both devices for operation to 85°C ambient temperature. The capacitors are low-ESR types from Nichicon's PL series (www.nichicon-us.com). R_2 dissipates less than 0.25W, but, at full load current, R_1 can dissipate 1W at high V_{IN} and more than 5W at low V_{IN}. You should locate R_1 away from the regulator IC to minimize heating. The DIP version of the 2594 dissipates as much as 0.5W at high V_{IN}; you should solder leads 1, 2, 3, and 6 to a ground-plane area greater than 2 in.^{2} to avoid thermal shutdown. If you don't need the IC's on/off feature, then you should also solder Pin 5 to the ground plane.

Figure 1



Less expensive than a "brick" converter, this circuit accommodates high input voltages and output currents.

Figure 2



The circuit of Figure 1 delivers good efficiency for 15 to 60V inputs.

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

Negative resistor cancels op-amp load

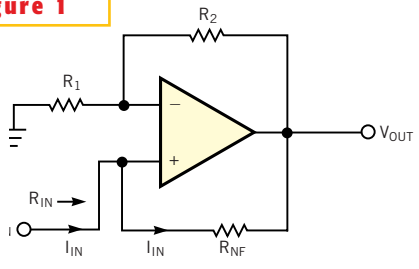
Elliott Simons, Maxim Integrated Products, Sunnyvale, CA

Accurate op amps have high open-loop gain, low offset voltage and current, low voltage and current noise, and low distortion. However, they often lack the ability to provide high output currents while maintaining all the other high-accuracy specifications. In other words, high-accuracy op amps have a problem driving low-impedance loads. One solution to the problem is to "cancel" the load. If your resistive load is RΩ and you connect it in parallel with a negative resistor of −RΩ, the resistance of the parallel combination is infinite. The circuit of **Figure 1** can generate negative resistance at its input: $R_{IN}=-R_{NF}(R_1/R_2)$. You derive this value as follows:

$V_{OUT}=V_{IN}(1+R_2/R_1)$;
$I_{IN}=(V_{IN}-V_{OUT})/R_{NF}=-(V_{IN}/R_{NF})(R_2/R_1)$; and
$R_{IN}=V_{IN}/I_{IN}=-R_{NF}(R_1/R_2)$.

Figure 2 presents a practical application of the concept. The first op amp is an accurate, unity-gain buffer, and the second op amp is a high-current, high-bandwidth, gain-of-2 driver. Because $R_1=R_2$ in this negative-resistance stage, its input resistance is $-R_{NF}=-200\Omega$, which matches the magnitude of the accurate buffer's load resistance. If these magnitudes match perfectly, the buffer sees an open circuit at its output. The buffer drives the positive input of the second amplifier, and the second amplifier, via its negative-resistance input, drives the load. Gain error, output-current limits, and resistor mismatches limit the minimum resistance the circuit can drive, but driving a 200Ω load is easy. That load is an order of magnitude lower than the load the unassisted accurate amplifier can handle without suffering degradation of performance. Note that the second op amp's gain error, offset voltage, and offset current do not affect the first op amp's accuracy. The step response of this circuit is well-behaved and exhibits no ringing.

Figure 1

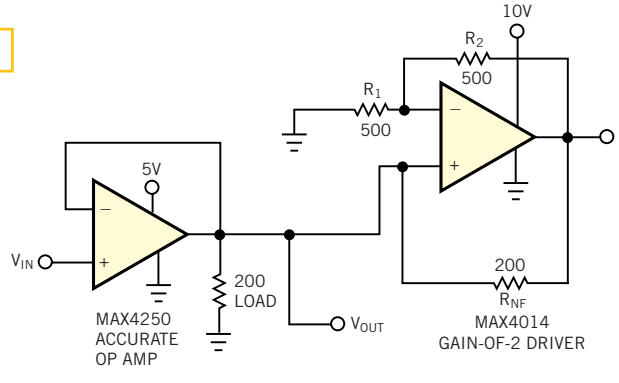


This circuit exhibits negative resistance at its input.

The negative-resistance approach works equally well with dual-supply op amps, because the negative-resistance portion can both source and sink current. If the driver op amp does not have built-in gain-setting resistors, you can set its noninverting gain closer to unity, thereby allowing both op amps to share a power supply. This approach limits the output swing of the accurate op amp, but that restriction may be acceptable in a given application. To ensure full bandwidth for the accurate op amp, the driver op amp should have much higher bandwidth.

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

Figure 2



Connecting a negative resistance in parallel with the load enables a precision op amp to drive 200V.

Op amps make JFET circuits repeatable

Glen Brisebois, Linear Technology Corp, Milpitas, CA

Because they use practically no bias current (a useful feature in itself), JFETs also have practically no current noise. This feature means that you can use JFETs in very-high-resistance circuits and obtain good noise performance. JFETs are also fast devices, with garden-variety types specified in the hundreds of megahertz. On the flip side, JFETs are difficult to exploit in a manufacturing environment because they have widely varying dc specifications. With a simple resistor-based bias network, the same part number can give results that differ by several volts from device to device. One way to use JFETs in a repeatable and manufacturable manner is to use the topology that **Figure 1** shows. The pur-

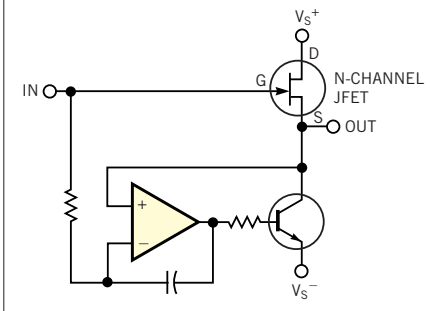


Figure 1

The op amp biases the JFET at I_{DSS}, with V_{GS} 50V.

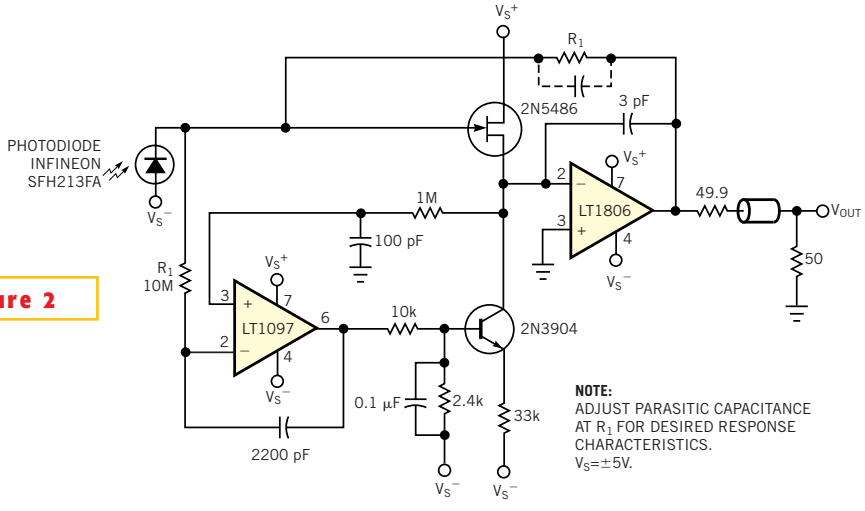


Figure 2

This fast, high-gain photodiode amplifier uses Figure 1's scheme to bias the JFET.

pose of the op amp is to bias the JFET at V_{GS}=0V and, therefore, at I_D=I_{DSS}. It meets this goal by increasing the current in the bipolar transistor until V_{GS}=0V and I_D=I_{DSS}.

In this condition, the JFET operates at its highest gain (g_m) and lowest voltage-noise condition. The JFET operates as a follower with zero offset. The only requirement for the op amp is that it have ultralow bias current. A variety of op-amp types satisfy this criterion, including JFET-input op amps, such as the LT1462; superbeta-input op amps, such as the LT1097; and micropower op amps, such as the LT1494. **Figure 2** shows an implementation of the topology of **Figure 1**, using an inexpensive, fast 2N5486 JFET. This device specifies I_{DSS} at 8 to 20 mA at room temperature. The LT1097 maintains the gate-source voltage at 0V by adjusting the JFET's drain current. The source of the JFET connects to the inverting input of the 325-MHz, low-noise LT1806 op amp. R_F closes the loop back to the JFET's gate. In this application, the circuit serves as a transimpedance amplifier for a fast photodiode.

Selecting a high value, 10 MΩ for R_1 maintains low noise gain, but you could reasonably reduce it to a few times larger than R_F. The values of the other resistors and capacitors in the LT1097 loop to attenuate the noise and shape the noise bandwidth of the slow loop. Measurements show the output-noise spectral density is 9 nV/$\sqrt{\text{Hz}}$ with R_F=0Ω, so resistor noise dominates with R_F greater than approximately 10 kΩ. **Table 1** shows the rise time and bandwidth achieved for several transimpedance gains (as set by R_F). To obtain optimum speed characteristics, you make "parasitic-capacitance adjustments" (the capacitor with broken lines in **Figure 2**) by adjusting the proximity of R_F's leads to its body. **Figure 3** shows the time-domain pulse response with R_F=1 MΩ. Connecting two 499-kΩ resistors in series improves the response.

TABLE 1–RESULTS FOR VARIOUS R_F WITH 1.2V OUTPUT STEP

R_F	10 to 90% rise time (nsec)	3-dB bandwidth (MHz)
100 kΩ	64	6.8
200 kΩ	94	4.6
499 kΩ	154	3
1 MΩ	263	1.8

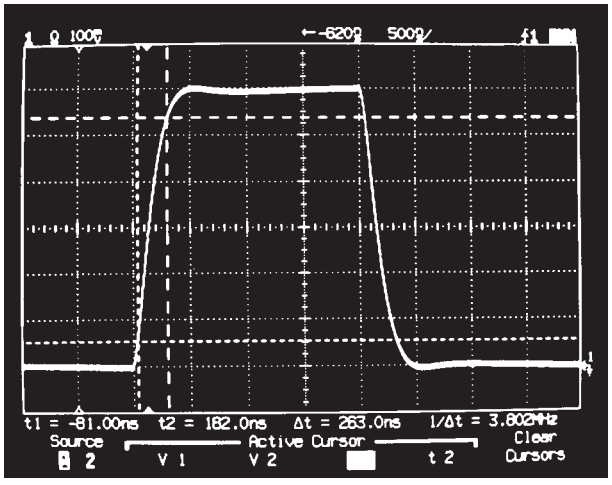


Figure 3

The circuit of Figure 2 exhibits clean pulse response with little overshoot or ringing.

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

design ideas

Edited by Bill Travis and Anne Watson Swager

Neat idea nets $1500 in *EDN*'s yearly Design Ideas contest

EDN is pleased to announce this year's $1500 grand-prize winner for Design Ideas: Art Hogrefe of State College, PA, a consultant who specializes in communications and analog-circuit design. He takes the honor for "Inverted bipolar transistor doubles as a signal clamp," which was published in *EDN*'s Nov 9, 2000, issue. In case you missed it the first time around, you can read all about it below.

In Hogrefe's innovative idea, he uses a bipolar transistor in an inverted configuration as a rectifier, or clamp, with low forward voltage. He offers a meticulously thorough comparison of the characteristics of the transistor versus those of a germanium diode. The idea of using bipolar transistors in an inverted mode is not new: these devices were popular as low-offset switches in R/2R ladder networks in D/A converters in the 1960s and 1970s. In his eminently useful Design Idea, Hogrefe takes the application a step further and thoroughly characterizes the bipolar transistor's attributes as a low-offset clamp and rectifier. Check it out for yourself.

And be sure to send us your own Design Idea. *EDN* publishes about 150 Design Ideas per year, and each one is automatically entered into a best-of-issue and year-end competition. You can find guidelines at www.ednmag.com/ednmag/write_di.htm along with a coupon to email submissions or use the coupon on pg 152 to submit your ideas by mail.

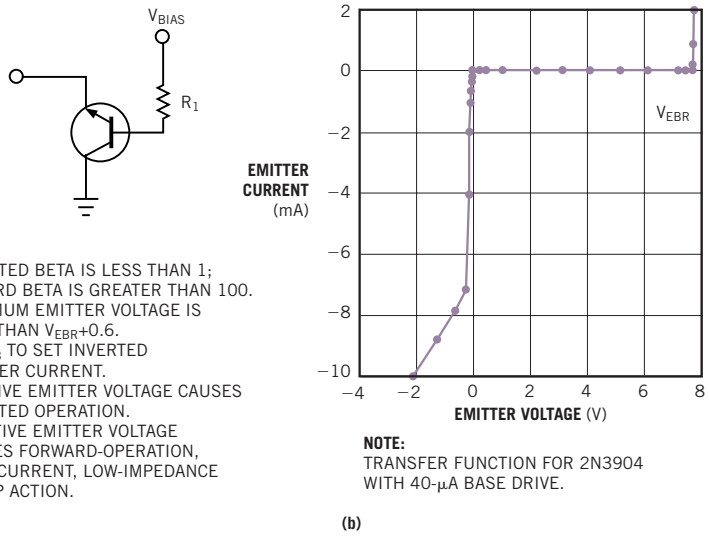
Inverted bipolar transistor doubles as a signal clamp

Art Hogrefe, Puma Instrumentation, State College, PA

A number of circuits, such as level detectors and AM demodulators, benefit from a rectifier with a low offset voltage. Silicon diodes have an offset of approximately 0.6V and do not work well in low-level circuitry. A Schottky diode is a bit better with an offset of approximately 0.4V. A few germanium diodes are still available, but they do not tolerate the temperature range of silicon. Also, you can't include a germanium diode in an IC. A superior configuration uses a bipolar transistor for these applications.

Figure 1 shows the bipolar-inverted-clamp circuit and a typical transfer function. The collector connects to ground or any other desired reference voltage. A fixed current drives the base. In the absence of any external drive, the emitter voltage is near zero. Driving the emitter with an external voltage produces the transfer function in **Figure 1**.

Figure 1

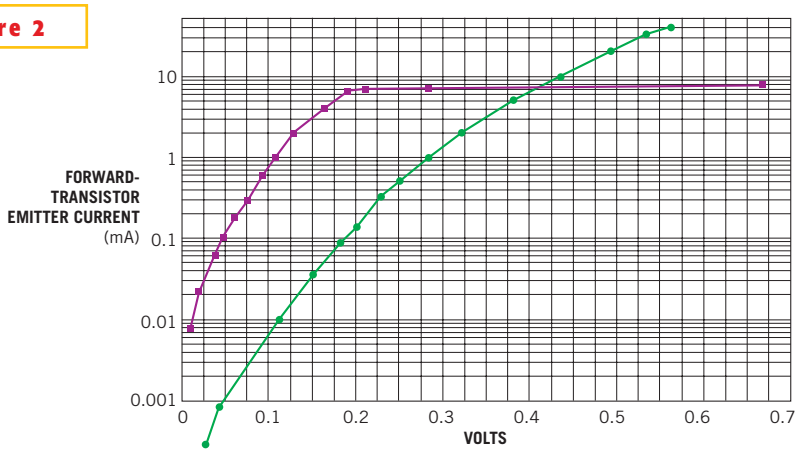


The bipolar inverted clamp (a) has an excellent rectification characteristic (b) because of the 2N3904's large forward-beta-to-reverse-beta ratio.

The circuit achieves this excellent rectification characteristic by using a transistor with a large forward-beta-to-reverse-beta ratio. Many of these transistors are still available. The 2N3904 provides excellent characteristics at a low cost. The reverse beta of the 2N3904 is only 0.25, so that for positive voltage on the emitter, and, with 40 μA of base drive, the emitter, current is around 10 μA. This current is sufficient in most level-detector applications for which the ac input amplitude changes slowly.

The emitter current at even small negative voltages is much greater than in the inverted region because the forward beta of the 2N3904 is greater than 100. Impedance is low up to the beta-limited forward current, at which point the impedance increases to approximately the value of R_L/beta. **Figure 2** shows the forward-transistor emitter current of the 2N3904 and the forward current of the 1N34 germanium point-contact diode. The logarithmic current scale shows the impressive response of the 2N3904 at small voltages.

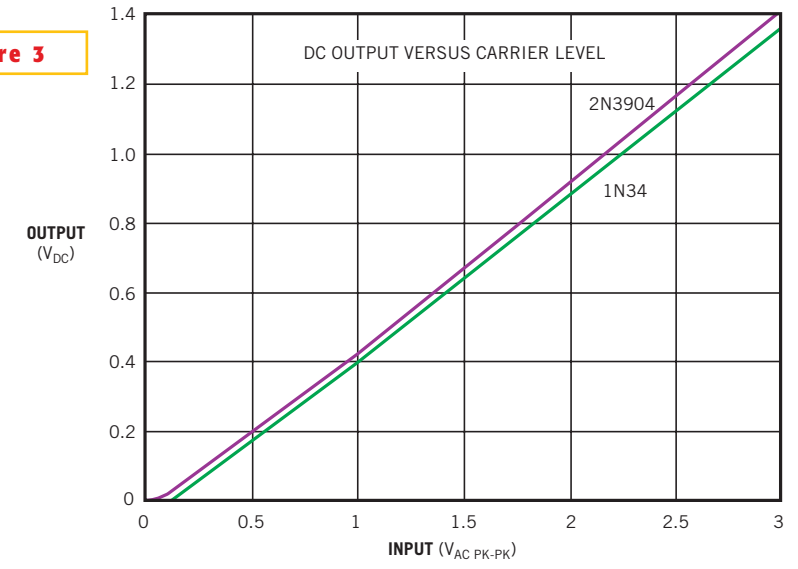
Figure 2



A logarithmic scale of the 2N3904's forward-transistor emitter current and the forward current of the 1N34 show the impressive response of the 2N3904 at small voltages.

Figure 3 shows the output as a level detector for the two clamps. The transistor circuit that produced these results is similar to the demodulator in **Figure 4** except the base drive is 40 μA. For the 1N34, the anode connects to ground, and the cathode connects to the input capacitor in place of the transistor's emitter. **Figure 3** shows that the two configurations have similar responses to input levels and that the 2N3904 has a bit less offset, as you would expect from **Figure 2**. The output can drive a signal level meter or following electronics as part of an automatic-level-control or automatic-gain-control loop.

Figure 3



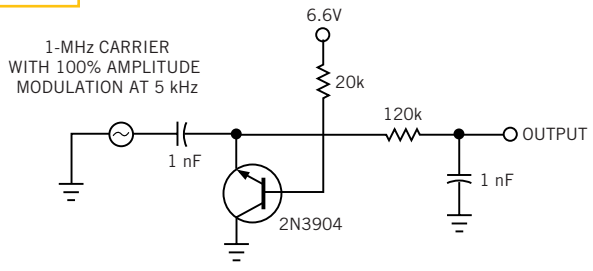
When operating as demodulators, the two configurations have similar input-level responses.

The transfer function in **Figure 1** also shows a sudden increase in inverted current at approximately 7.6V, which occurs at the reverse breakdown voltage for the emitter-to-base junction. Because you know in this case that the base is near 0.6V, the breakdown voltage for the tested part is near 7V. Production circuits would have an input limit of 6.6V p-p because of the minimum specified breakdown voltage of 6V. Note that, for a small production, such as for test equipment, it is practical to select individual transistors to slightly increase the dynamic range. A 6V p-p input dynamic range is sufficient in many applications.

The RF demodulator in **Figure 4** has a base drive current of 300 μA. This current is necessary to track the RF-modulation envelope and depends on the size of the input capacitor, modulation frequency, and maximum signal amplitude. The reverse current, which is I_{BASE} times

the reverse beta, must be large enough to discharge the input capacitor at the highest modulation frequency and amplitude to prevent distortion in the output waveform. **Figure 5** shows the running demodulator with the upper trace at the emitter node and the lower trace at the output.

Figure 4



NOTES:
BASE-DRIVE CURRENT IS 300 μA, EMITTER REVERSE CURRENT IS APPROXIMATELY 75 μA.
2N3904 FORWARD BETA IS GREATER THAN 100; REVERSE BETA IS APPROXIMATELY 0.25.
EMITTER-BASE REVERSE-BREAKDOWN VOLTAGE IS GREATER THAN 6V.
AC-DRIVE VOLTAGE IS LESS THAN 6.6V P-P.

Using the inverted clamp as an AM demodulator requires a base current of 300 μA to track the RF-modulation envelope.

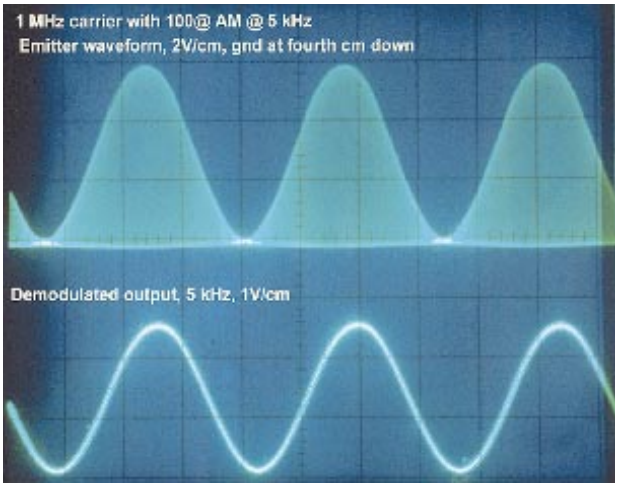


Figure 5 **A scope photo shows the running demodulator; the upper trace is the emitter node, and the lower trace is the output.**

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors. Find the entry blank at http://www.ednmag.com/ednmag/di_entry.htm to email submissions.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02458

I hereby submit my Design Ideas entry. (Please print clearly)

Name _____

Title _____

Phone _____

E-mail _____ **Fax** _____

Company _____

_____

Address _____

_____

Country _____ **Zip** _____

Design Idea Title _____

_____

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to *EDN*, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit text, software listings, and all other computer-readable documentation on a PC disk by mail or by e-mail to b.travis@cahners.com.

Exclusive publishing rights remain with Cahners Business Information unless entry is returned to author, or editor gives written permission for publication elsewhere. The author must be willing to sign and return our publication agreement if the Design Idea is accepted for publication and must complete a W-9 tax form (W-8 for non-US residents) before payment can be processed.

Signed _____

Date _____

Stepdown converter uses a ceramic output capacitor

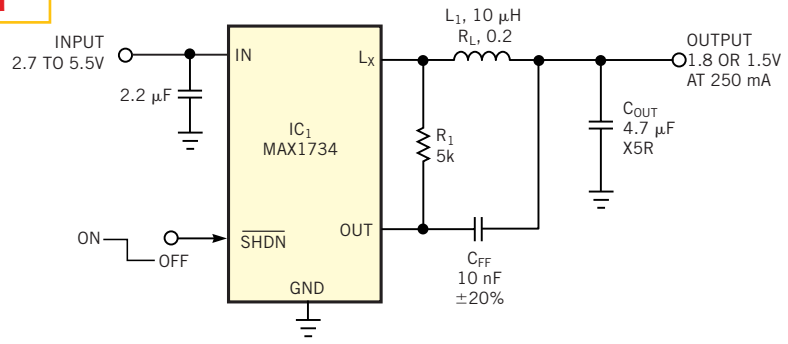
Karl R Volk, Maxim Integrated Products, Sunnyvale, CA

Many stepdown (buck) dc/dc-converter ICs incorporate a voltage-mode-control algorithm. As a result, for stable operation in continuous-conduction mode, the application circuit's output capacitor is normally a high-ESR tantalum type for two reasons. The portion of output ripple due to ESR provides the current-mode signal that's necessary for cycle-to-cycle stability. In the frequency domain, this capacitor also provides a zero that cancels a pole in the buck converter's second-order LC filter, thereby shifting operation back to the stable region by reducing the ripple's phase shift to less than 90°.

The circuit in **Figure 1**, however, allows the use of an inexpensive ceramic output capacitor. To remove the effects of phase lag in the feedback loop, the circuit derives feedback from the L_X pin, via the first-order RC filter comprising R_1 and C_{FF} instead of the output. Connecting the tail of C_{FF} to the output node instead of to ground, as you would for a normal filter, provides a fast "feedforward" load-transient response.

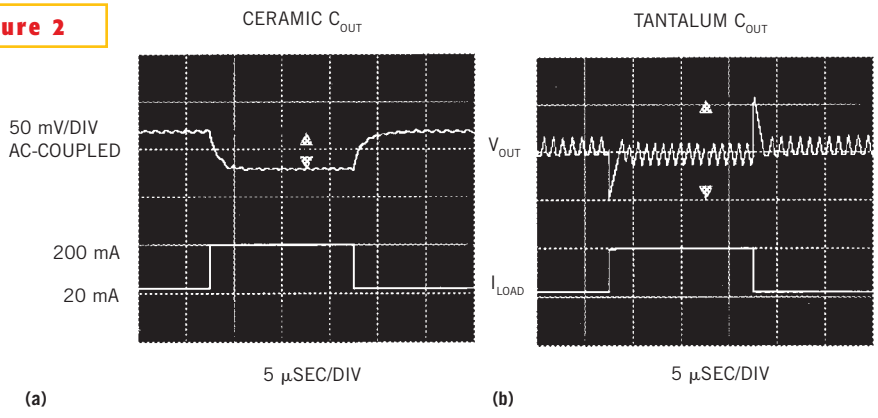
A ceramic-capacitor circuit offers several benefits over a standard application circuit. First, ceramic capacitors are more reliable than tantalum capacitors. Second, ceramic capacitors are more readily available than tantalum types. Third, ceramic capacitors cause output ripple of less than 5 mV p-p versus more than 20 mV p-p (**Figure 2**). For this circuit, the load-transient overshoot is also lower: less than 50 mV p-p versus more than 100 mV p-p.

Figure 1



A stepdown dc/dc converter, which normally requires a tantalum output capacitor, can operate with a ceramic output capacitor by deriving feedback from the L_x pin via the R_1/C_{FF} filter.

Figure 2



Load-transient response waveforms show that using a ceramic output capacitor produces lower output ripple and less overshoot (a) than the standard application circuit with a tantalum output capacitor (b).

IC_1, a stepdown dc/dc converter with an internal synchronous rectifier that supplies a fixed 1.8 or 1.5V output at 250 mA from an input range of 2.7 to 5.5V, needs 20 mV p-p or more at its output pin for stable operation under load. To meet this requirement, calculate the value of R_1:

$$R_1 \cong \left(\frac{20\,mV}{2 \times V_{OUT}}\right)\left(\frac{L_1}{T_{MIN}}\right)\left(\frac{I_{LOADMAX}}{2 \times I_{OUTSENSE}}\right).$$

Per the data sheet for the MAX1734, V_{OUT} is 1.5 or 1.8V, L_1 is 10 µH, T_{MIN} is 0.4

µsec, $I_{LOADMAX}$ is 250 mA, and $I_{OUTSENSE}$ is 4 µA. The result is R_1=4.3 kΩ for V_{OUT}=1.8V, and R_1=5.2 kΩ for V_{OUT}=1.5V. You can therefore round R_1 to 5 kΩ.

Next you calculate the feedforward capacitor value:

$$C_{FF} \le \left(\frac{2 \times V_{OUT}}{20\,mV}\right)\left(\frac{T_{MIN}}{R_1}\right).$$

If R_1=5 kΩ and V_{OUT}=1.5V, then $C_{FF} \le 12$ nF. Select C_{FF}=10 nF. Choosing a much smaller value causes excessive load-transient overshoot, and choosing a larger value causes instability under loaded conditions. For optimized load transients, the inductor series resistance should be as follows:

$$R_L \cong \frac{L_1}{R_1 \times C_{FF}}.$$

Note that this expression is the typical, not the maximum, inductor resistance. In this case, the value of R_L should be approximately 200 mΩ, which allows you to use a small inductor and causes an approximate efficiency drop of only 3% at maximum loads and much less at lighter loads. Because the inductor time constant, L_1/R_L, matches the feedback time constant, $R_1 \times C_{FF}$, the short-term load-transient response equals the dc load regulation (**Figure 2**). If R_L is less than 200 mΩ, the peak-to-peak load-transient voltage increases, but the dc-load regulation decreases.

Finally, choose C_{OUT} large enough for stability:

$$C_{OUT} \ge 2 \times \left(\frac{\Delta I_L}{20\,mV}\right) \times T_{MIN},$$

where ΔI_L is approximately 100 mA when the MAX1734 operates with a 10-µH inductor. In this case, C_{OUT} should be greater than 4 µF.

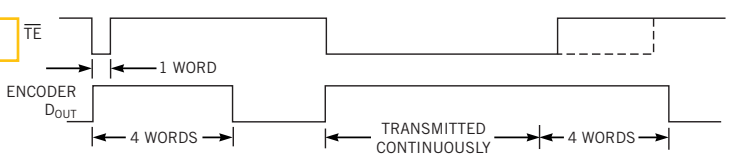
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Single printer-port pin acts as an encoder output

Haobin Dong, Huazhong University of Science and Technology, China

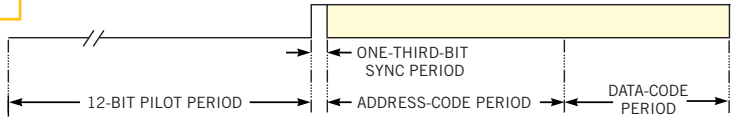
Encoders and decoders are common elements in alarm, remote-control, and measurement systems. However, most of these devices require many I/O lines when under microprocessor or PC control. For example, the HT-12E encoder has eight address pins, four data pins, and one transmit-enable-control pin. As an alternative, you can simulate the HT-12E using a single pin of a PC's printer port as the encoder output. Software determines the functions of the encoder.

Figure 1



At power-up, the HT-12E encoder begins a 4-word transmission cycle.

Figure 2



One complete transmission period includes a pilot period, a bit-sync period, an address-code period, and a data-code period.

The HT-12E is a CMOS IC. This encoder serially transmits data as defined by the state of the A_0 to A_7 and D_0 to D_3 input pints. On power-up, the D_{OUT} pin is low. The HT-12E begins a 4-word transmission cycle on receipt of a transmission enable, or TE, signal, which is active low. The cycle repeats as long as the TE signal is low. When TE goes high, the encoder completes its final 4-word transmission cycle and then stops (**Figure 1**).

You can preset the status of each address or data pin independently to logic high or low. If the TE signal is low, the encoder scans and sequentially transmits the status of the 12 bits of address and data in the order of A_0 to A_8 and D_0 to D_3 (**Figure 2**).

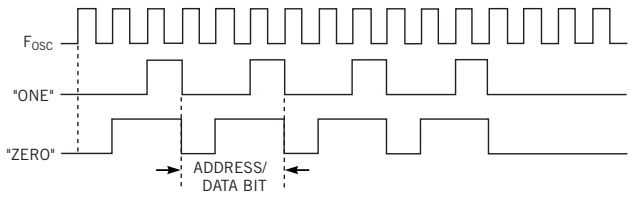
The IC encodes each logic high or low into pulses (**Figure 3**). The encoder represents a logic low as a long pulse (011) and a logic high as a short pulse (001). Every logic bit takes three OSC periods. The information sequentially transmits via the D_{OUT} pin.

Figure 4 shows the test circuit for a virtual encoder that includes an optocoupler and the decoder, IC_1. The encoder output comes from printer Port Pin 2 of the DB25 connector. The data port is at address 0x378h of the PC's LPT. R_1 limits drive current from the PC's printer port. IC_2's Schmitt trigger shapes the optocoupler's output. The D_0 pin of IC_1 connects to R_2 and an LED.

According to **Figure 2** and **Figure 3**, one complete transmission period consists of 73 OSC periods. The pilot period, which is 12 bits, is all logic low (0, 36

OSC periods) followed by a one-third-bit sync period (one OSC). After those periods follows 8 bits of address and 4 bits of data, all of which need 36 OSC periods. Every bit of address or data is either a "001" encoded pulse for a logic high or a "011" for all other cases. A time interval exists between two pulses, and a software loop controls the interval. Also, the interval must be in accordance with the OSC periods of the decoder.

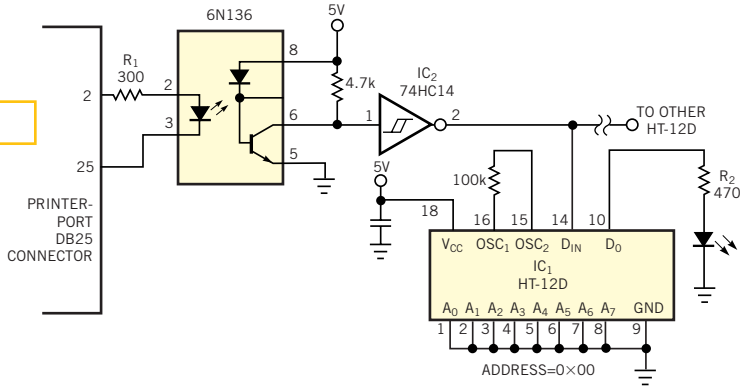
Figure 3



A short pulse of "001" represents a logic high, or a "one," and a long pulse of "011" represents a logic low, or a "zero."

Software listings consist of implementation routines in Turbo C, the main code for determining the delay time, and a test-program site. You can download the listings 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 #2713. The delay time,T_delay, is a global variable in the codes. The variables T_min, T_max, and T_step are properly predefined based on the PC. After you compile and execute the delay-time program, the T_delay value is read when the LED starts to flash. The value for the current design and the PC is 1000. Compile and run the test program, and you will observe that the LED continues to flash until you hit any key to stop the code.

Figure 4



A test circuit includes an optocoupler and a decoder IC.

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

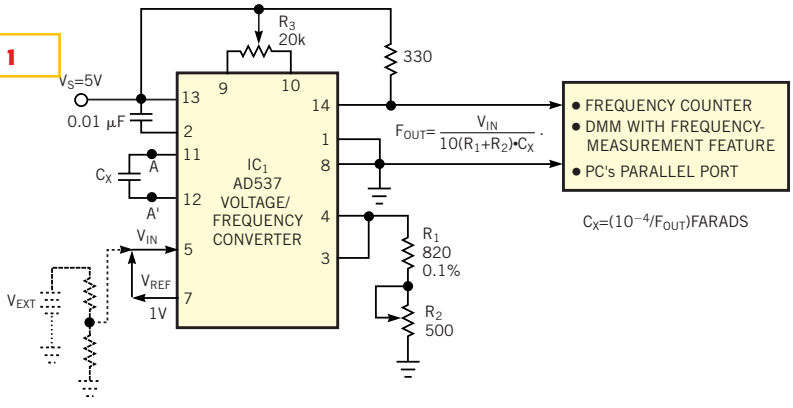
VFC makes simple capacitance meter

K Suresh, Indira Gandhi Centre for Atomic Research, Kalpakkam, India

When you develop the prototype of a circuit using capacitors or when you replace an old, defective capacitor with a new one, you often need to know the value of the capacitor you use or replace. At times, the values printed on the capacitors are no longer readable. Also, the wide tolerance band of the capacitors can leave you making a wild guess about the capacitor value. In these situations, you normally go looking for an LCR bridge of a DMM with a capacitance-measurement facility. Not all DMMs have this feature, and finding the capacitance value from an LCR bridge is a cumbersome process.

An alternative is a simple, low-cost VFC (voltage/frequency converter) with a few inexpensive components (**Figure 1**). This circuit can measure capacitance values of nanofarads to tens of microfarads. The output transfer function is

Figure 1



A simple VFC uses its internal reference to produce a serial pulse train whose frequency is inversely proportional to unknown capacitance C_X. Options for measuring the VFC's output include: using a frequency counter, using a DMM with a frequency-measurement feature, and hooking the output to a PC parallel port through simple counters and buffers.

$$F_{OUT} = \frac{V_{IN}}{10(R_1 + R_2) \times C} Hz,$$

where V_{IN} is in volts, R_1 and R_2 are in ohms, and C is in farads. For IC_1, with an input range of 0 to 10V, the output of the

VFC is a serial pulse train in the frequency range of 0 to 150 kHz with a nonlinearity error of less than 0.05%.

In normal operation of the VFC, you provide a V_{IN} of 0 to 10V and choose R_1 and R_2 such that you get $V_{IN}/(R_1+R_2)$, a current range of 0 to 1 mA to obtain a good linearity between V_{IN} and F_{OUT}. However, in this design, there is no external input to the VFC. Instead, the design exploits an internal reference voltage, V_{REF}, by simply connecting the V_{REF} output to the V_{IN} input. To avoid the possibility of loading the reference source, you can also apply V_{REF} to V_{IN} through a buffer. Also, you can use an external voltage input from a constant voltage source, such as a battery, and connect it, as the dotted lines in the **figure** indicate.

To measure an unknown capacitance value, you connect the capacitor between terminals A and A' very close to the VFC. With $V_{IN}=V_{REF}=1.00V$ and R_1+R_2 trimmed to 1 kΩ, the resulting output of the VFC is a serial pulse train whose frequency varies inversely with the value of the unknown capacitance C_X as follows:

$$C_X = \frac{10^{-4}}{F_{OUT}} \text{FARADS.}$$

You use the R_3 and R_2 trims to obtain calibrations at the higher and lower ends of the capacitance range, respectively. Thus, after due calibration, a 1-Hz output of the VFC indicates the unknown capacitance as 100 μF, and the maximum output of the VFC, 150 kHz, indicates that the capacitance is approximately 0.6 nF. If you want to increase the measurable capacitance range, you use a VFC with a wider output-frequency range, such as 0 to 1 MHz. In this case, you must take care of the parasitic capacitances.

You can measure the output of **Figure 1**'s circuit in a number of ways. One simple and direct approach is to use a simple frequency counter or a low-cost DMM with a frequency-measuring feature. Thus, the simple VFC becomes a handy capacitance interface for your DMM to enable you to measure the capacitance. Alternatively, you can also use a programmable counter, such as the Intel 8254, which is available in most PC add-on cards. One more approach is to attach a simple 16-bit counter, such as the CD4040 and CD4520, to your printer port using the necessary buffering and control (**Reference 1**). In the last two cases, you can exploit a special BIOS interrupt, INT1Ch of your PC, without affecting its normal service routine to provide a measuring window of 1 second. During the measuring window, the serial output of the VFC drives a counter. A the end of the measuring window, the counter contents transfer to the PC, and you manipulate the data to display the unknown capacitance value directly on the PC's screen.

REFERENCE

1. "Use your printer port as a high-current ammeter," *EDN*, July 6, 2000, pg 144.

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

design ideas

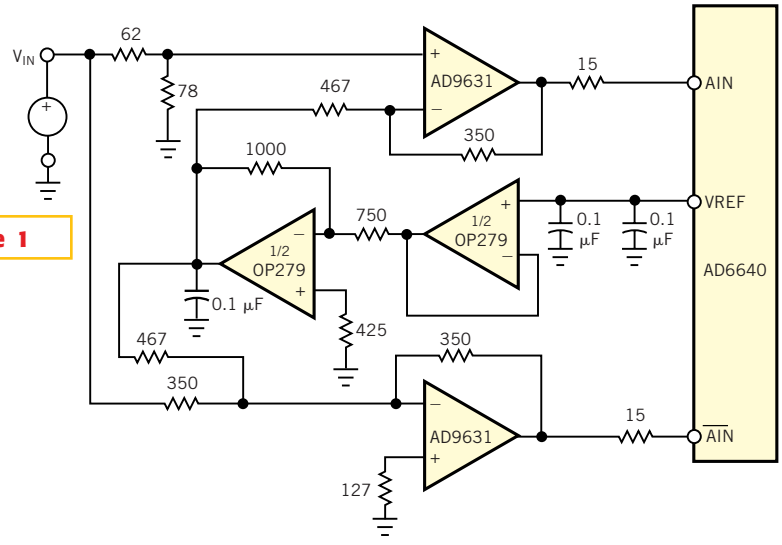
Edited by Bill Travis and Anne Watson Swager

Differential amp drives high-speed ADC

Chau Tran, Analog Devices Inc, Wilmington, MA

THE SCHEMATIC IN **Figure 1** is the discrete-element version of an A/D-converter drive circuit. The circuit converts a single-ended input to a differential output. The ADC's reference voltage determines the common-mode range of the differential outputs. The circuit contains two AD9631 amplifiers—one connected in noninverting mode, and the other connected in inverting mode. The OP279 amplifiers buffer and scale the ADC's reference voltage to set the common-mode range of the two outputs. The circuit in **Figure 1** requires many resistors. The two 15Ω resistors help prevent oscillation arising from the capacitive inputs of the ADC. The circuit has several disadvantages, such as poor gain accuracy, high distortion, and limited speed. The circuit in **Figure 2** is an improved ADC driver.

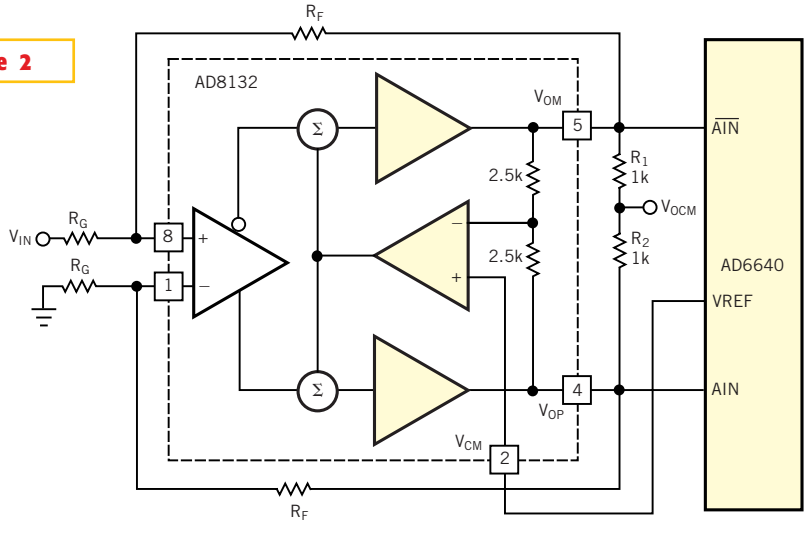
Figure 1



This ADC driver uses many resistors and is prone to errors.

The circuit consists of one AD8132 amplifier and four resistors. You can set the gain of the system by adjusting the ratio of R_F to R_G. The input accommodates both single-ended and dif-

Figure 2



An integrated amplifier provides improved accuracy and higher speed.

ferential signals. The circuit in **Figure 2** is a low-distortion, high-speed (300-MHz-bandwidth) driver. You can also use it to drive precision delta-sigma ADCs. **Figure 3** shows the performance at 10 MHz and unity gain ($R_F=R_G=499\Omega$). **Figure 4** shows the gain error and the low distortion in the circuit of **Figure 2**. The waveform at the node V_{OCM} indicates the output-balance error. The topology of **Figure 2**'s circuit also improves the common-mode rejection ratio, because it provides level-shifting to the reference voltage of the ADC.

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

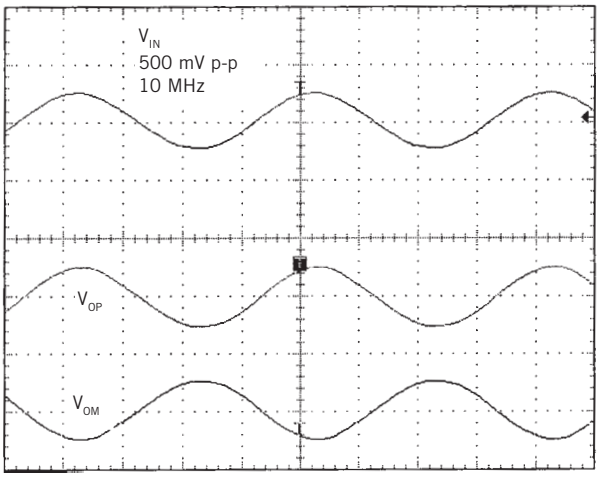


Figure 3 At 10 MHz and unity gain, the out-of-phase outputs have low distortion.

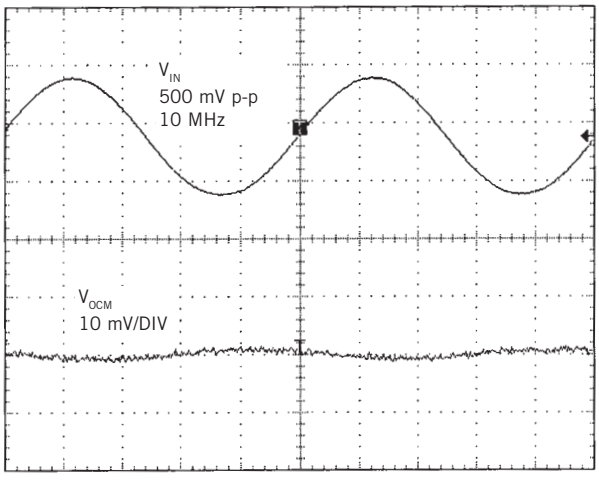


Figure 4 The output-balance error in Figure 2's circuit is lower than 1 mV with a 500-mV, 10-MHz input signal.

Circuit provides flexible gain ranges

Luo BenCheng, Chinese Academy of Sciences, Beijing, China

Certain designs need a programmable-gain amplifier with a wide gain range and high accuracy and common-mode rejection. Usually, it's wise to exploit a programmable-gain instrumentation amplifier, such as an AD625. Unfortunately, the gain range of such standard parts is fixed at certain values, limiting their flexibility. **Figure 1** shows a multichannel, eight-level-programmable-difference-amplifier circuit. IC_1, an AD623, operates from a single supply. This amplifier is a low-power, low-cost instrumentation amplifier that offers good accuracy. A single external resistor sets the gain from 1 to 1000. IC_2, a CD4051, is a programmable, low-voltage 1-of-8 analog multiplexer, which connects to eight weighting resistors, R_0 to R_7, to increase the gain range of the circuit. The overall gain of the circuit depends on the value of the selected weighting resistor. You can compute the weighting resistors, R_0 to R_7, for a given input-output signal range as follows: $V_{OUT}=V_{IN}(1+2R_K/-(R_X+R_{ON}))$, where R_{ON} is the on resistance of the CD4051, typically 125Ω. R_K is the 50-kΩ internal feedback resistor of the AD623, and R_X is one of the selected weighting resistors. IC_3, a CD4052, is a 2-of-8 programmable-difference-input IC. You can control the port-select pins, Z_0 and Z_1, of IC_3 and Z_2 to Z_4 of IC_2 with a µC, such as an AT89C51 or an 80C196. With the aid of some software, the circuit can provide self-adjusting gain.

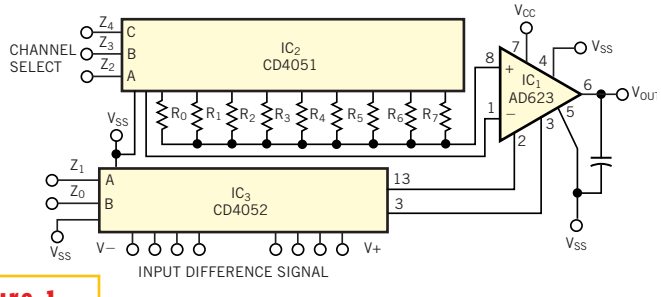


Figure 1 You choose the weighting resistors to obtain the optimum gain ranges for your application.

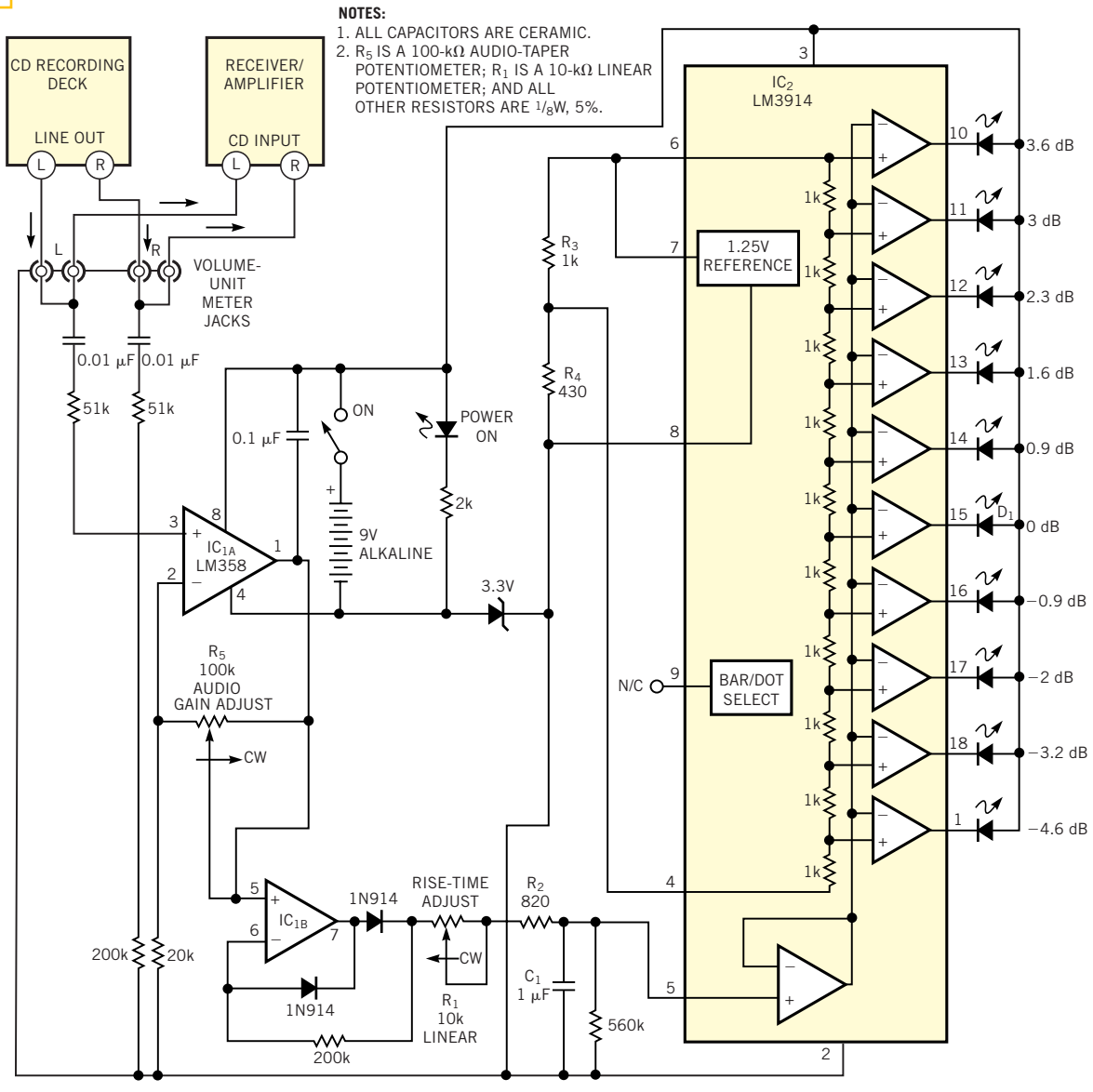
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

High-resolution volume-unit meter simplifies CD recording

Chester Simpson, National Semiconductor, Santa Clara, CA

Digitally recorded music on CDs offers superior quality to that recorded on vinyl records or tape, but most prerecorded CDs have an annoying characteristic: The average volume levels of the recorded signal can vary by as much as 14 dB from disk to disk. Significant variances, such as 4 dB, can occur from track to track on a single CD. This variation can be a problem when recording your own CDs because you can end up with large variations in loudness between songs.

Figure 1



A high-resolution, average- (not peak-) reading volume-unit meter produces an accurate reading of loudness.

CD digital-recording decks typically have peak-reading-only volume-level meters. This feature is adequate to prevent clipping but does a poor job of reading the average volume, or loudness, level. If you set the recording levels just below the peak clip level, average volume variances of 6 to 8 dB can result due to varying levels of dynamic-range compression in recording the original material.

A high-resolution, average- (not peak-) reading volume-unit meter produces an accurate reading of loudness (**Figure 1**). The meter connects to the line outputs from the recording deck to monitor signal level. IC_{1A} sums the left and right channels and adjusts gain. This adjustment allows you to calibrate the 0-dB LED, D_1, to the signal level of your CD recorder.

IC_{1B} is a precision rectifier, which sends the positive portion of the signal to the input of IC_2, an LM3914 dot/bar-display IC. R_1, R_2, and C_1 filter the signal to the input of IC_2 to provide an averaging effect. R_1 adjusts the rise time of the signal, which increases or decreases the meter's ability to track a fast rising signal. Increasing R_1 slows rise time, making the meter more average-reading than peak-reading.

IC_2 contains all the necessary circuitry to drive a 10-LED string as well as an internal reference. R_3 and R_4 bias up the bottom of the resistor string to 0.4V above the ground pin, which effectively reduces the total decibel range of the meter and increases the resolution and accuracy of each LED step.

Another feature of IC_2 is that the resistors connected to Pin 7 externally program the LED current. The resistor values in **Figure 1** result in an LED current of approximately 8 mA. To conserve power, the circuit leaves the bar/dot-select pin, Pin 9, open to select "dot" mode. IC_2 has some built-in overlap, so that at least three LEDs are on at any time during typical operation of the meter, which eliminates flicker and makes the meter easier to view.

You use this volume-unit meter with the peak-reading meter in the recording deck. You should initially set up the recording levels by finding the loudest portion of the music and adjusting the record level of the deck to approximately 4 to 5 dB below the peak clip level the meter shows on the CD deck. The CD-deck makers advise against any clipping because it causes distortion. Backing off by 4 to 5 dB from the clip level allows some headroom in case later-recorded tracks have higher dynamic range (the ratio of peak signal level to average signal level). You should then set the gain-adjust potentiometer, R_5, so that the 0-dB LED lights on the loudest sound. Adjust subsequent track-record levels to this same level, based on the average loudness, to ensure that the deck's recording meter does not exceed the clip level.

Adjust R_1, the rise-time adjust, so that the meter will respond to average level changes but miss fast peaks. This control is user-adjustable, and the optimum setting depends on the music type.

The meter's bandwidth is approximately 250 Hz to 2 kHz; the upper limit depends on R_1's setting. This bandwidth selects the range of sound that most reflects perceived loudness based on the ear's frequency response. Lower bass signals typically have large signal levels in absolute magnitude, but the ear hears these signals less due to the Fletcher-Munson effect. If they register on the meter display, the bass-frequency signals would indicate higher loudness than you actually hear.

The ear also perceives very-high-frequency sounds less loudly than absolute magnitude would suggest, although the effect is not as pronounced as the low-frequency roll-off. So, to get an accurate approximation of loudness, the meter measures the middle range where the ear is most sensitive.

The battery current was measured and found to be 22 mA while operating, which means the expected battery life of a 9V alkaline battery should be at least 20 hours. Because the meter sets only record levels, you can switch it off during recording to conserve the battery.

The circuit connections to the external stereo system are in the line coming from the line outputs of the CD-recording deck and going to the input of the receiver/amp. The meter effectively connects in parallel with the input of the receiver/amp. For this reason the volume-unit meter's input impedance is very high to prevent loading the line output.

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

Small, portable altimeter operates from a single cell

Todd Owen, Linear Technology Corp, Milpitas, CA

Some sports enthusiasts want to know altitude changes from an initial elevation. A small, lightweight, portable altimeter is easy to design using modern micromachined pressure transducers. Inverting barometric pressure and compensating for nonlinearities in air-pressure changes with respect to altitude produces a reasonably accurate altimeter.

Figure 1 shows a small, handheld altimeter based on a micromachined pressure transducer. The circuit takes advantage of the inverse relationship between air pressure and altitude. The aim of this circuit is to be small, lightweight, and portable. Accuracy is not paramount; errors as high as 3%, such as a 300-ft error

at 10,000-ft altitude, are acceptable. The speed of the circuit is also not critical: Extreme changes in altitude in milliseconds may prove fatal to whoever is attempting to read the output.

The heart of the altimeter is an NPC-1220-015-A-3L pressure transducer. This 5-kΩ bridge provides 0 to 50 mV of output voltage for a 0- to 15-psi pressure range. To power the transducer and signal-conditioning circuitry, a micropower dc/dc converter, IC_1, generates 5V from a single AA battery, and a charge pump generates a −5V supply.

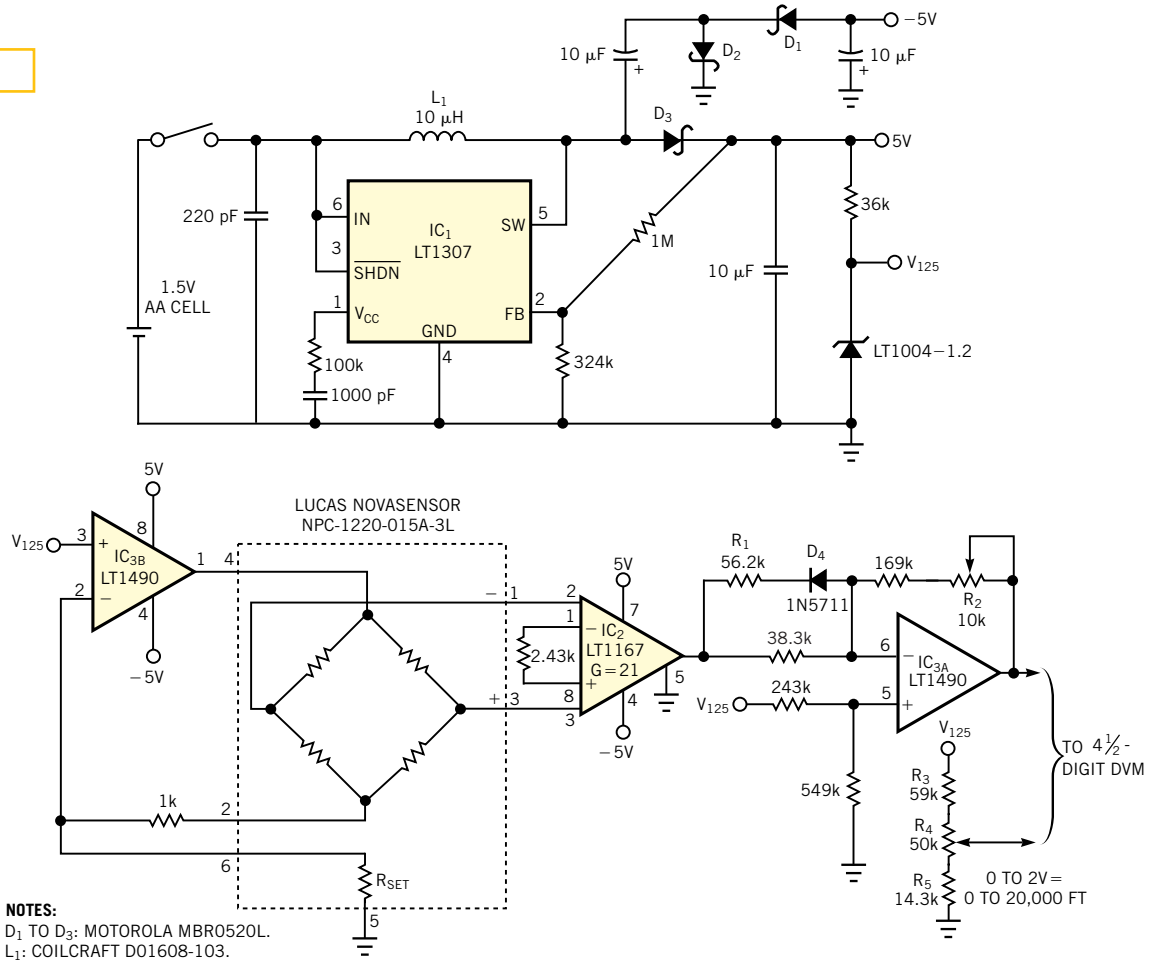
The output of the transducer drives an instrumentation amplifier, IC_2, which provides an initial gain of 21. A nonlinear gain stage comprising IC_{3B} and associated components then inverts the output of the instrumentation to provide a voltage that is inversely proportional to air pressure. D_4 and R_1 introduce the nonlinear gain, and the final output is directly proportional to altitude.

R_2 performs gain calibration in the signal-conditioning circuitry. This potentiometer calibrates out any normal variations in part tolerances and sets the altimeter for a 100-mV change in output for every 1000 ft of altitude. The circuit has some initial offset, as well as an offset that is determined by barometric-pressure variations. You can use R_3 to R_5 to null this offset, giving a 0 to 1V output for 0 to 10,000 ft of altitude.

Altimeter testing was performed using a DeHavilland DHC-6 Twin Otter for an ascent to 13,000 ft, followed by free descent—limited by the engineer's parasitic drag—to 3000 ft. Subsequent deployment of an aerodynamic decelerator (Precision Aerodynamics Icarus Omega 190) prevented engineer injury or circuit damage. Aircraft rental for testing is available at many local airports. Extensive instruction in free descent and the use of aerodynamic decelerators are highly recommended before undertaking testing of this nature. Contact USPA at 1-703-836-3495 for further information.

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

Figure 1



To produce a reasonably accurate altimeter, conditioning circuitry inverts the barometric pressure of a micromachined pressure transducer and compensates for nonlinearities in air-pressure changes with respect to altitude.

Circuit breaker monitors leakage current

Sharath Kumar, Gemplus R&D Centre, Dubai, United Arab Emirates

The residual-current circuit breaker in **Figure 1** continuously monitors the supply lines for any leakage current and immediately disconnects the supply if necessary. Load-supply wires, both live and neutral, pass through the magnetic core of the CR4311-5 transducer (www.crmagnetics.com), which monitors the supply current. Under normal circumstances, because the current flowing in both conductors is equal and opposite, no flux is generated in the transducer core. However, under faulty conditions, the current in the live wire exceeds the current in the neutral wire, which catalyzes the production of flux in the core. This transducer core has a secondary winding that generates a voltage based on the produced flux. The generated voltage ranges from 0 to 10V and is directly proportional to the sensed ac currents.

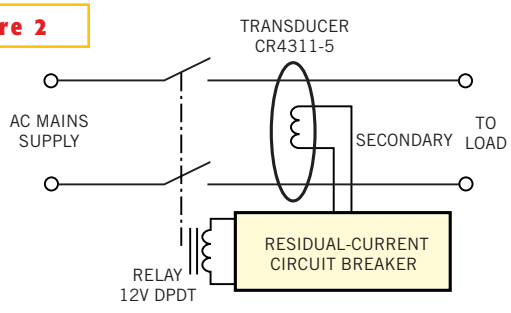
A high-speed comparator, IC_{1B}, detects this generated voltage and compares it with a set reference. If the detected voltage is within a tolerable range, the relay remains active, and the load remains connected to the mains supply. However, if this voltage exceeds tolerable limits, the circuit immediately deactivates the relay, thereby disconnecting the faulty load. Any further or repeated attempts to restart the device with the faulty load result in repeated tripping of the relay. You have to manually disconnect the faulty load and restart the device.

The circuit configures IC_{1B} as a precision, fast-acting voltage comparator. IC_{1A} provides a stable 6V reference to IC_{1B}. When the voltage on the noninverting input of IC_{1B} rises above the preset reference voltage on its inverting input, the output goes high. This output controls Q_1. When Q_1 turns on, Q_2 turns off, which deactivates the relay.

Two trim potentiometers facilitate tripping at user-preset levels. R_2 controls the coarse setting, and R_1 provides for finer adjustments. Typically, the muscles in the human body can tolerate current up to 20 mA. Hence, R_1 and R_2 must have settings that cause the relay to trip at leakage currents of greater than 15 mA that the transducer senses from the load-mains supply wires. R_3 allows control over the hysteresis. D_1 to D_3 provide protection. C_1 and C_2 are decoupling and charge-pump capacitors, respectively.

A 12V, 0.5A mains power-supply unit is sufficient to effectively run the circuit. The relay contacts must have a rating suitable for the load. **Figure 2** shows the wiring layout for attaching the circuit to an ac-mains circuit. All components are standard industrial grades and are commonly available.

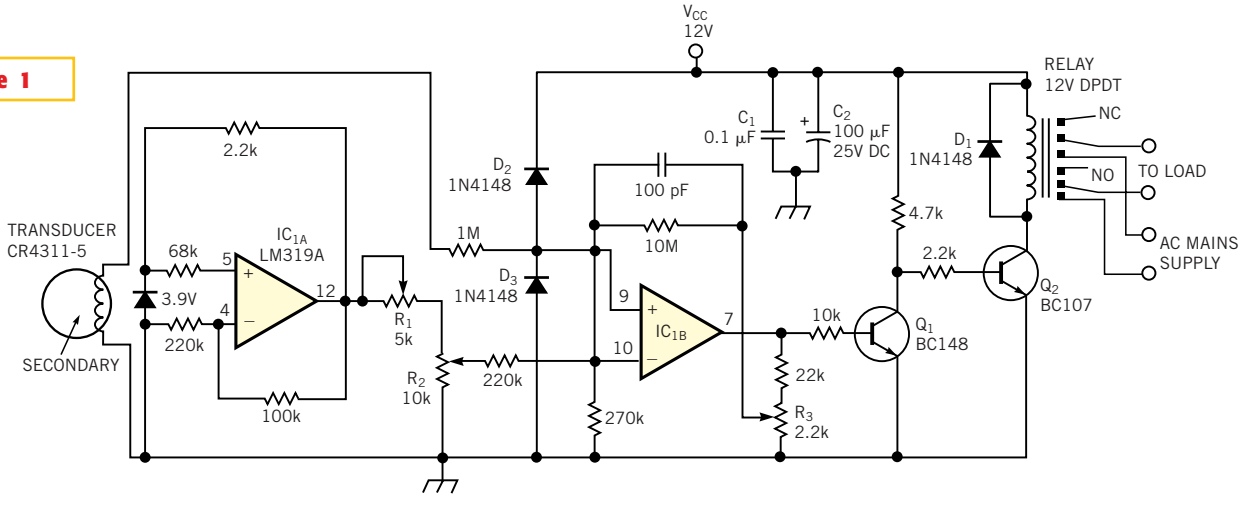
Figure 2



The residual-current circuit breaker uses a transducer to monitor the supply current and a relay to disconnect the mains from the load.

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

Figure 1



IC_{1B} is a precision, fast-acting comparator and controls whether the relay is active based on a preset reference-voltage level.

design ideas

Edited by Bill Travis and Anne Watson Swager

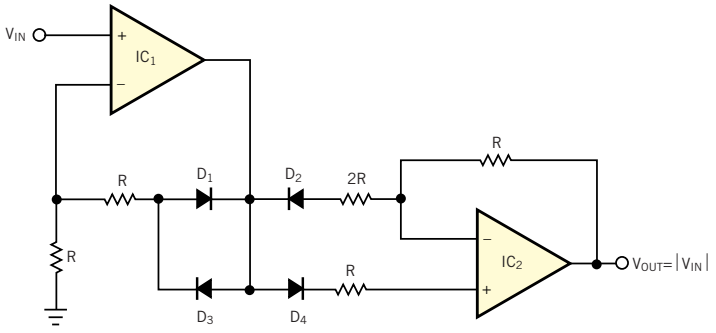
Circuit yields accurate absolute values

Marco Pisani, Istituto di Metrologia G Colonnetti, Turin, Italy

THE CIRCUIT IN FIGURE 1 delivers the absolute value of the input signal with an accuracy better than 10 ppm of the full-scale range. The circuit has low zero-crossing error. You can use it as an asynchronous demodulator, as a source for logarithmic amplifiers, or simply as a demonstration of the wonders of feedback. The circuit uses two op amps; five identical resistors, R; one double-value resistor, 2R; and four thermally matched diodes. When the input, V_{IN}, is positive, IC_1's output is $2V_{IN}+V_{D3}$ (the voltage drop across D_3). D_2 is reverse-biased, thus IC_2 behaves as a voltage follower, yielding $V_{OUT}=R(V_{IC1}-V_{D4})/2R$. Because the same amount of current (V_{IN}/R) flows in D_3 and D_4, assuming their characteristics are the same, $V_{D3}=V_{D4}$, and $V_{OUT}=V_{IN}$. When V_{IN} is negative, IC_1's output is $2V_{IN}-V_{D1}$. D_4 is reverse-biased, and IC_2 is an inverting amplifier, yielding $V_{OUT}=R(V_{IC1}+V_{D2})/2R$. Again, the current flowing in D_1 and D_2 is V_{IN}/R, and $V_{OUT}=-V_{IN}$. For good performance,

- The six resistors must match closely to guarantee symmetrical gain.
- The diode pairs, D_1-D_2 and D_3-D_4, must have tight thermal coupling to minimize errors at low input voltages. (It's best if the pairs are on the same chip.)
- The op amps must have low offset.

Figure 1



This full-wave rectifier circuit yields the accurate absolute value of the input signal.

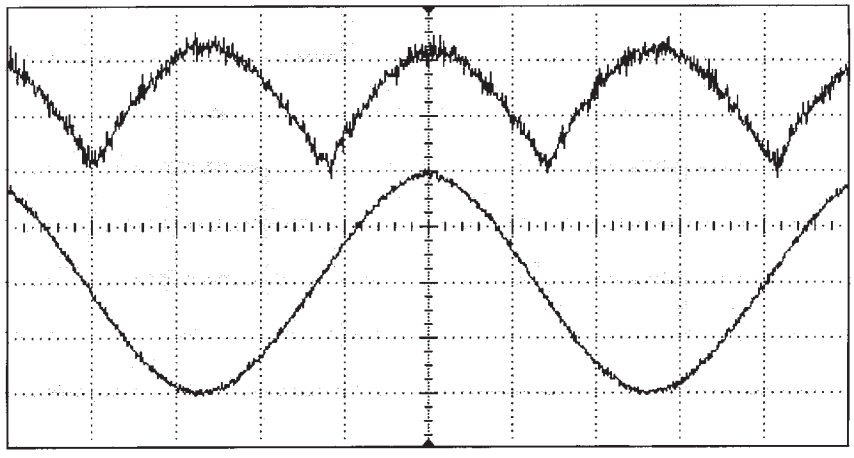


Figure 2

Even with a 40-mV p-p signal, the circuit in Figure 1 yields an accurate absolute value.

In a practical configuration, you can configure D_1 through D_4 using base-collector junctions of a monolithic transistor array, such as an MPQ6700. The resistors are 10- and 20-kΩ, 1% metal-film units. You can use optional 100Ω trimmers in series with the resistors in the circuit to trim for optimum performance. The op amps are OP27 devices, with their offset trimmed. After adjusting the op amps' offset and tweaking the resistors, the residual error is within 100 µV p-p over the 13V p-p operating range. **Figure 2** shows the behavior of the circuit with a 40-µV p-p input signal (bottom trace).

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

ADC enables temperature-compensated weigh-scale measurements

Albert O'Grady, Analog Devices, Limerick, Ireland

You can provide temperature compensation in weigh-scale applications by simultaneously measuring both the temperature of the bridge and the primary output of the bridge transducer. Traditionally, an integrated multiplexer connects multiple input variables to a single sigma-delta ADC. Each time the multiplexer switches the input, the ADC must flush the digital filter of all data pertaining to the previous channel. Before the new data becomes valid, the system must account for the settling time and latency, reducing the maximum throughput rate. For example, with an ADC containing a second-order sigma-delta modulator and a third-order digital filter, the output-settling time for a step input is three times the period of the data rate. Switching from the primary to the secondary channel can reduce the primary channel's throughput by a factor of six when you need to monitor primary and secondary variables together. In many cases, you can monitor the secondary variable only intermittently and thus minimize the reduction in throughput. **Figure 1** shows a solution to the throughput problem that uses the two independent channels of an AD7719 dual sigma-delta ADC.

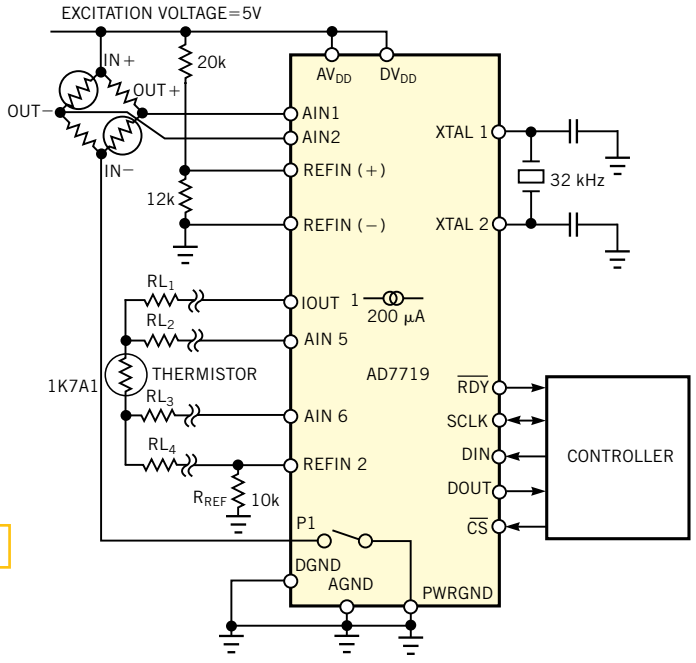


Figure 1

Two independent ADCs eliminate the throughput limitations of multiplexed measuring systems.

The ADCs convert in parallel, so you can simultaneously measure both the bridge output and the bridge temperature. The output data from both measurements is available in parallel, thereby removing the latency associated with multiplexed data-acquisition systems. The main channel monitors the bridge transducer, and the secondary channel monitors the bridge temperature. The bridge transducer develops a differential output voltage between the Out(+) and Out(–) terminals. A bridge sensitivity of 3 mV/V produces a full-scale output of 15 mV when a 5V excitation source powers the bridge. The ADC's reference voltage can assume any value between and including the supply voltages, so you can use the bridge-excitation voltage to provide the reference to the ADC. A resistive divider, however, allows you to use the full dynamic range of the input. This implementation is fully ratiometric, so variations in the excitation voltage do not introduce errors in the system.

The resistor values of 20 and 12 kΩ in **Figure 1** yield a 1.875V reference voltage for the AD7719, with a 5V excitation voltage. The main-channel (programmable) gain is 128, resulting in a full-scale input span of the ADC equal to the full output span of the transducer. A low-side switch disables the transducer to save power in standby mode. The AD7719 features factory calibration, and its signal chain uses a chopping scheme to reduce gain and offset drifts, eliminating the need for field calibration. A key requirement in weigh-scale applications is the ability to reject line-frequency components (50 and 60 Hz). You can achieve simultaneous 50- and 60-Hz rejection by programming the AD7719 for an output data rate of 19.8 Hz. With a gain of 128, the ADC achieves 13-bit resolution at this data rate. You can increase the resolution by reducing the update rate or by providing additional digital filtering in the controller.

The secondary channel of the AD7719 monitors the bridge temperature with the aid of a thermistor. An on-chip current source excites the thermistor and generates the reference voltage for the AD7719. As a result, excitation signals do not affect performance, and the configuration is fully ratiometric. The circuit uses a four-wire force/sense configuration to reduce the effects of lead resistance. Lead resistance of the drive wires shifts the common-mode voltage but does not degrade the performance of the circuit. Lead resistance of the sense wires is immaterial because of the high impedance of the AD7719's analog inputs. The

reference-setting resistor, R_{REF}, must have a low temperature coefficient. The AD7719 achieves 16-bit resolution in the secondary channel, using a 19.8-Hz update rate. The thermistor determines the operating range of the circuit. The maximum voltage on the auxiliary input is REFIN 2 or 2V. With a Betatherm 1K7A1 thermistor (www.betatherm.com) and 200-µA excitation current, the operating range is −26 to +70°C.

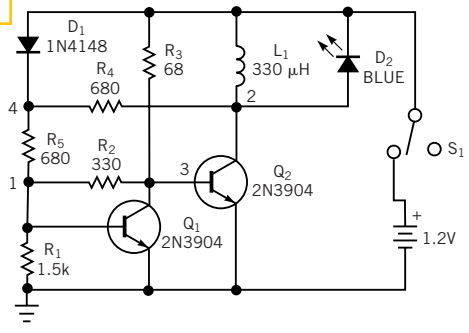
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Single cell lights any LED

Al Dutcher, Al Labs, West Deptford, NJ

The circuit in **Figure 1** allows you to light any type of LED from a single cell whose voltage ranges from 1 to 1.5V. This range accommodates alkaline, carbon-zinc, NiCd, or NiMH single cells. The circuit's principal application is in LED-based flashlights, such as a red LED in an astronomer's flashlight, which doesn't interfere with night vision. White LEDs make handy general-purpose flashlights. You can use the circuit in **Figure 1** with LEDs ranging from infrared (1.2V) to blue or white (3.5V). The circuit is tolerant of the varying LED voltage requirements and delivers relatively constant power. It provides compensation for varying battery voltage. The circuit is an open-loop, discontinuous, flyback boost converter. Q_2 is the main switch, which charges L_1 with the energy to deliver to the LED. When Q_2 turns off, it allows L_1 to dump the stored energy into the LED during flyback.

Figure 1



A simple circuit provides drive from a single cell to an LED of any type or color.

Q_1, an inverting amplifier, drives Q_2, an inverting switch. R_4, R_5, and R_2 provide feedback around the circuit. Two inversions around the loop equal noninversion, so regeneration (positive feedback) exists. If you replace L_1 with a resistor, the circuit would form a classic bistable flip-flop. L_1 blocks dc feedback and allows it only at ac. Thus, the circuit is astable, meaning it oscillates. Q_2's on-time is a function of the time it takes L_1's current to ramp up to the point at which Q_2 can no longer stay in saturation. At this point, the circuit flips to the off state for the duration of the energy dump into the LED, and the process repeats. Because inductors maintain current flow, they are essentially current sources as long as their stored energy lasts. An inductor assumes any voltage necessary to maintain its constant-current flow. This property allows the circuit in **Figure 1** to comply with the LED's voltage requirement.

Constant-voltage devices, such as LEDs, are happiest when they receive their drive from current sources. The LED in **Figure 1** receives pulses at a rapid rate. The inductor size is relatively unimportant, because it determines only the oscillation frequency. If, in the unlikely case the inductor value is too large, the LED flashes too slowly, resulting in a perceivable flicker. If the inductor value is too small, switching losses predominate, and efficiency suffers. The value in **Figure 1** produces oscillation in the 50-kHz neighborhood, a reasonable compromise. D_1 provides compensation for varying cell voltage. By the voltage-division action at Node 4, D_1 provides a variable-clipping operation. The higher the supply voltage, the higher the clipping level, and the result is correspondingly less feedback. Q_1 inverts this clipping level to reduce the turn-on bias to Q_2 at higher cell voltages. We chose 2N3904s, but any small-signal npn works. Q_2 runs at high current at the end of the charging ramp. Internal resistance causes its base-voltage requirement to rise. The R_2-R_1 divider at Q_1's base raises the collector voltage to match that requirement and thus controls Q_2's final current.

The LED's drive current is a triangular pulse of approximately 120 mA peak, for an average of approximately 30 mA to a red LED and 15 mA to a white one. These levels give a reasonable brightness to a flashlight without unduly stressing the LED. The supply current for the circuit is approximately 40 mA. A 1600-mAhr NiMH AA cell lasts approximately four hours. L_1 must be able to handle the peak current without saturating. The total cost of the circuit in **Figure 1** is less than that of a white LED. You can use higher current devices and larger cells to run multiple LEDs. In this case, you can connect the LEDs in series. If you connect them in parallel, the LEDs need swamping (ballast) resistors. You can also rectify and filter the circuit's output to provide a convenient, albeit uncontrolled, dc supply for other uses.

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

Lowpass filter uses only two values

Richard M Kurzrok, Queens Village, NY

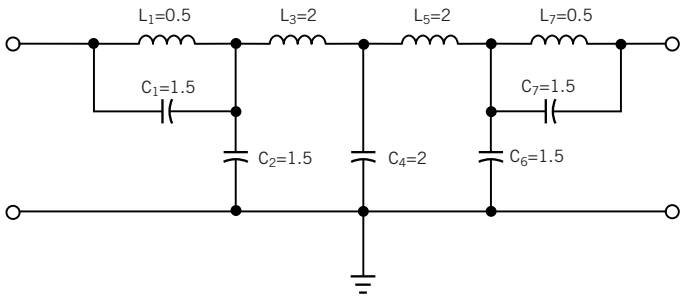
In recent years, image-parameter design of LC filters has received new consideration (**references 1** and **2**). The composite lowpass filter uses interior constant-k full sections terminated by m-derived half-sections. For best passband response, you usually select m to equal 0.6. However, m=0.5 can still give useful filter performance while reducing the number of component values. A low number of component values is advantageous for low-cost manufacturing. The design technique is also applicable to highpass and wideband filters (**references 3** and **4**). **Figure 1** shows a schematic of the composite lowpass filter. The filter uses four inductors of two different values and five capacitors of two different values. **Figure 2** shows the schematic of an equivalent composite lowpass filter. This filter uses judicious combinations of components in series, parallel, and series-parallel. The filter uses eight inductors and 14 capacitors of only one value each.

TABLE 1–AMPLITUDE RESPONSE FOR FILTER IN FIGURE 2

Frequency (MHz)	Insertion loss (dB)	Frequency (MHz)	Insertion loss (dB)
1	0.1	4	5.2
1.5	0.1	4.1	12.7
2	0.15	4.2	20.8
2.5	0.1	4.3	31.2
2.9	0.2	4.4	Greater than 40
3.2	0.25	5	Greater than 40
3.4	0.3	10	Greater than 40
3.6	0.4	15	Greater than 40
3.7	0.45	20	Greater than 40
3.8	0.6	25	Greater than 40
3.9	1.4	30	Greater than 40

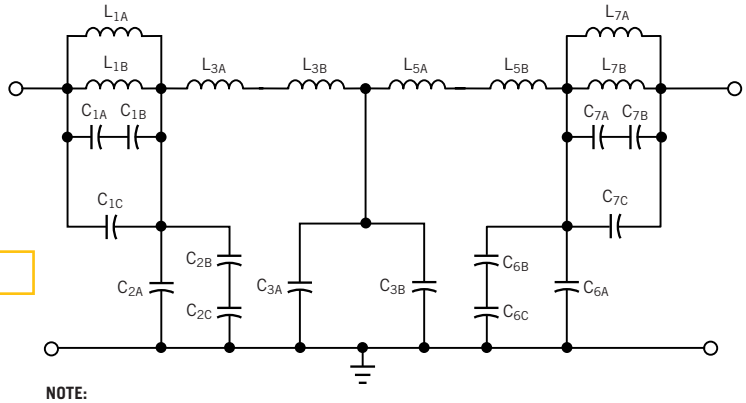
Table 1 gives test results for the lowpass filter of **Figure 2**. We constructed the filter on Vector board in a die-cast aluminum enclosure with BNC input and output connectors. The 3-dB cutoff design frequency is 3.88 MHz with source and load impedances of 50Ω. All capacitors are polypropylene units of 820 pF±5%. All inductors are 2.05 µH and made of 20 turns of number 26 AWG magnet wire on Micro Metals T37-2 toroids. The toroidal inductors, with unloaded Q exceeding 100, provide low passband-insertion loss. Surface-mount inductors with unloaded Q of 10 to 20 yield higher passband losses, which are acceptable in many applications.

Figure 1



A composite lowpass filter uses four component values.

Figure 2



NOTE:
ALL NORMALIZED INDUCTORS AND CAPACITORS EQUAL 1.

By judicious connection of components, this filter uses only one value each for the inductors and capacitors.

References

1. Kurzrok, Richard M, "Low cost lowpass filter design using image parameters," *Applied Microwave & Wireless*, February 1999, pg 72, and correction May 1999, pg 12.

2. Kurzrok , Richard M, "Update the design of image-parameter filters," *Microwaves & RF*, May 2000, pg 119.

3. Kurzrok, Richard M, "Filter design uses image parameters," *EDN*, May 25, 2000, pg 111.

4. Kurzrok, Richard M, "Wideband filter uses image parameters," *EDN*, Oct 26, 2000, pg 174.

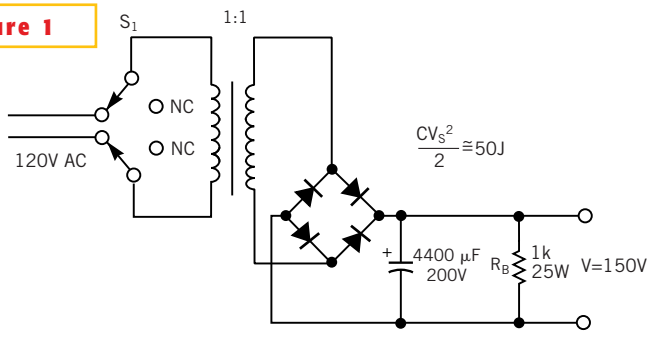
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Quickly discharge power-supply capacitors

Stephen Woodward, University of North Carolina, Chapel Hill, NC

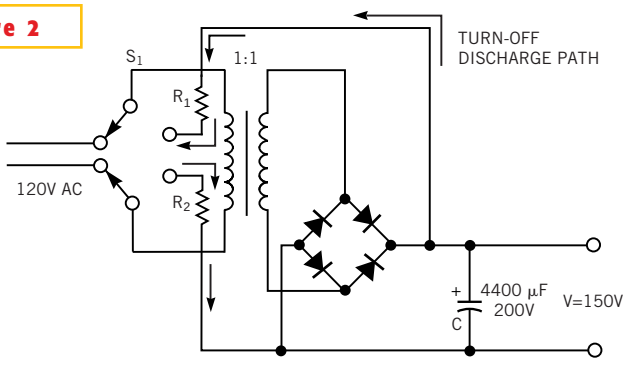
A PERENNIAL CHALLENGE in power-supply design is the safe and speedy discharge, or "dump," at turn-off of the large amount of energy stored in the postrectification filter capacitors. This energy, $CV^2/2$, can usually reach tens of joules. If you let the capacitors self-discharge, dangerous voltages can persist on unloaded electrolytic filter capacitors for hours or even days. These charged capacitors can pose a significant hazard to service personnel or even to the equipment itself. The standard and obvious solution to this problem is the traditional "bleeder" resistor, R_B (**Figure 1**). The trouble with the R_B fix is that power continuously and wastefully "bleeds" through R_B, not only when it's desirable during a capacitor dump, but also constantly when the power supply is on. The resulting energy hemorrhage is sometimes far from negligible.

Figure 1



A bleeder resistor ensures safety but wastes much power.

Figure 2



Otherwise unused switch contacts can dump energy while not wasting power.

Figure 1 offers an illustration of the problem, taken from the power supply of a pulse generator. The $CV^2/2$ energy stored at the nominal 150V operating voltage is $150^2 \times 4400\ \mu F/2$, or approximately 50J. Suppose you choose the R_B fix for this supply and opt to achieve 90% discharge of the 4400-µF capacitor within 10 sec after turning off the supply. You then have to select R_B to provide a constant RC time no longer than 10/ln(10), or 4.3 sec. R_B, therefore, equals 4.3 sec/4400 µF, or approximately 1 kΩ. The resulting continuous power dissipated in R_B is $150^2/1\ k\Omega$, or approximately 23W. This figure represents an undesirable power-dissipation penalty in a low-duty-cycle pulse-generator application. This waste dominates all energy consumption and heat production in what is otherwise a low-average-power circuit. This scenario is an unavoidable drawback of bleeder resistors. Whenever you apply the 10%-in-10-sec safety criterion, the downside is the inevitable dissipation of almost half the $CV^2/2$ energy during each second the circuit is under power.

Figure 2 shows a much more selective and thrifty fix for the energy-dump problem. The otherwise-unused off-throw contacts of the DPDT on/off power switch create a filter-capacitor-discharge path that exists only when you need it: when the supply is turned off. When the switch moves to the off position, it establishes a discharge path through resistors R_1 and R_2 and the power transformer's primary winding. The result is an almost arbitrarily rapid dump of the stored energy, while the circuit suffers zero power-on energy waste. Use the following four criteria to optimally select R_1, R_2, and S_1:

- The peak discharge current, $V/(R_1+R_2)$, should not exceed S_1's contact rating.
- The pulse-handling capability of R_1 and R_2 should be adequate to handle the $CV^2/2$ thermal impulse. A 3W rating for R_1 and R_2 is adequate for this 50J example.
- The discharge time constant, $(R_1+R_2)C$, should be short enough to ensure quick disposal of the stored energy.
- S_1 must have a break-before-make architecture that ensures breaking both connections to the ac mains before making either discharge connection, and vice versa. Otherwise, a hazardous ground-fault condition may occur at on/off transitions.

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

design ideas

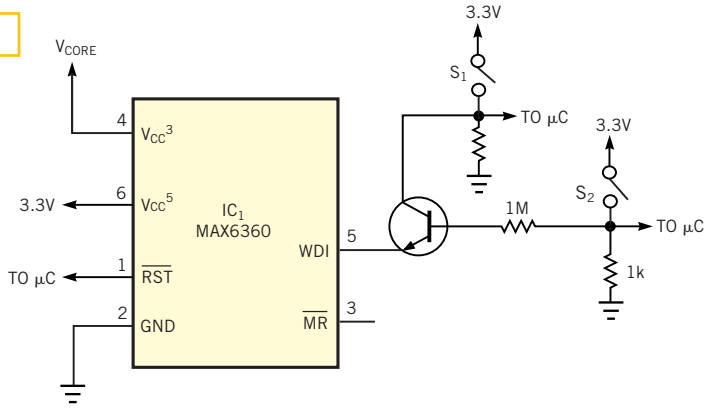
Edited by Bill Travis and Anne Watson Swager

Watchdog timer allows entry to test mode

Greg Sutterlin and Larry Barnes, Maxim Integrated Products, Sunnyvale, CA, and Craig Gestler, Mining Safety Appliances Co, Mars, PA

MANUALS FOR microprocessor-based devices often include instructions for entering a "secret mode" in which you can test or reset the device. These instructions typically ask you to depress and hold one or two switches for a minimum time interval. You can adopt several measures to avoid accidentally triggering the test mode: Depress two keys, hold both keys simultaneously and continuously through the minimum interval, and make the interval two to five seconds long. You can implement such designs with a handful of resistors, capacitors, and diodes and a comparator or two. Such circuitry may remain useless or lack customer appreciation, but it raises cost and complexity while lowering system reliability. A better alternative is to implement the function with minimal additional components (**Figure 1**).

Figure 1



You can activate IC_1 only by depressing S_1 and S_2 simultaneously for at least 2.9 sec.

Microprocessor-based devices usually include a voltage supervisor that monitors the V_{CC} level and, in some cases, the core voltage. When either voltage drops below its threshold, the supervisor issues a reset to the microprocessor. IC_1 in **Figure 1** monitors two voltages, provides a power-on reset, and includes a watchdog timer. If you don't need the watchdog function elsewhere in the system, it can help to implement the switch-delay circuitry. Fortunately, the watchdog timer self-resets when the WDI (watchdog input) Pin 5 is floating, an indication that the circuit is disabled. Thus, you can implement the switch-delay circuitry by enabling and disabling the watchdog timer. Simply provide an interface between WDI and the switches that enable the test mode. The nominal time-out period for IC_1's watchdog timer is 2.9 sec.

WDI must float when both switches are open and when you depress only one. When you depress both, WDI must assume either the active-high or the active-low state as specified in the data sheet. Note that WDI can remain floating while sinking 5 µA. The npn transistor isolates WDI from the switches' pulldown resistors. When S_2 closes, current flows through the base of the transistor to WDI, which remains floating because the 1-MΩ resistor limits the current to less than 5 µA. The transistor turns on and forces WDI to the active-high level only when you depress S_1 as well. If you depress the switches in reverse order, WDI switches high only when you close S_2. You must close both switches to activate the watchdog timer. After the timer activates, IC_1 imposes a 2.9-sec delay before asserting a reset (Pin 1). If you release either or both switches during this 2.9-sec period, the timer resets. Thus, for the minimal cost of an npn transistor and three resistors, the supervisor IC can monitor two voltages, provide a power-on-reset signal, and implement a dual-switch delay function. To monitor one voltage, you can replace IC_1 with a MAX823.

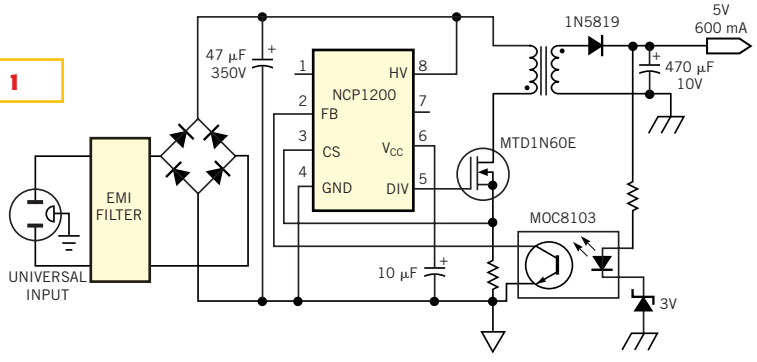
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Transistors offer thermal protection for controller

Christophe Basso, On Semiconductor, Toulouse, France

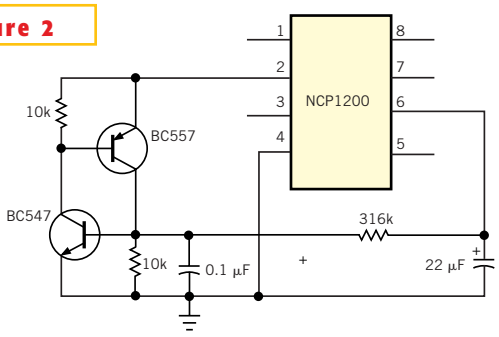
When a switch-mode power-supply controller, such as On Semiconductor's NCP1200, operates at a high ambient temperature, you should protect the entire power supply against lethal thermal runaway. The NCP1200 operates directly from the power mains without an auxiliary winding; therefore, the die in the IC dissipates power (**Figure 1**). Unfortunately, the internal temperature-shutdown circuitry cannot perform its protection function because the die is not at ambient temperature but at a temperature that's higher than ambient by a few tens of degrees. To overcome this problem, you can implement a thermistor-based design, but this solution compromises the system's cost. Fortunately, you can use standard bipolar transistors to implement a low-cost thermal-shutdown circuit. **Figure 2** shows how to build a classic thyristor circuit using two inexpensive bipolar transistors: a BC557B pnp and a BC547B npn. The idea is to use the negative-temperature coefficient of the silicon (~−2 mV/°C) to fire the thyristor.

Figure 1



A controller IC implements a low parts-count offline power supply.

Figure 2



Two bipolar transistors configure a thyristor-based temperature-shutdown circuit.

In the inactive state, both the upper and the lower transistors in **Figure 2** are in the off state because of the presence of the 10-kΩ resistors. The thyristor structure connects between the feedback pin, FB, and ground. One feature of the NCP1200 is to skip unwanted switching cycles when the power demand diminishes. The IC performs this function internally by constantly monitoring the FB pin. When the voltage on this pin falls below a certain level, the IC internally blanks the cycles, and the power transistor turns off. If the thyristor permanently pulls the FB pin to ground, the NCP1200 no longer delivers pulses. Once latched, the thyristor prevents any restart, until you disconnect the power supply from the power mains. The 316-kΩ resistor combines with the 10-kΩ resistor to form a voltage divider from the V_{CC} rail. This rail, on average, varies from lot to lot from 10.3 to 10.6V, for a total ΔV of 300 mV. This variation translates to less than 10 mV at the transistor's base. When the temperature rises, the BC547's turn-on V_{BE} diminishes until it reaches the divider voltage on its base. (This voltage is approximately 320 mV, but you can alter it to accommodate other temperature levels.) At this point, the BC547 conducts current, and the BC557's base voltage starts toward ground. The BC557's collector current further biases the BC547's base, and the thyristor latches, thereby permanently stopping the NCP1200's pulses. Once you remove the supply from the power mains, the thyristor resets. The 0.1-µF capacitor prevents spurious noise from triggering the thyristor.

We conducted tests on the thyristor-based temperature-protection scheme using BC547B and BC557B transistors. The "B" extension is important because it corresponds to a narrow h_{FE} range of 200 to 450. This design uses transistors in TO-92 packages, mounted close to each other. If only one transistor heats up, thermal results vary. Therefore, you should mount these two components close to each other on the pc-board-component side so that they will operate at approximately the same junction temperatures. From 20 bipolar-transistor combinations, you can obtain the results shown in **Table 1**. You can see that the latch-off threshold temperature varies by only approximately 5°C for all combinations of transistors (**Reference 1**).

TABLE 1–TEMPERATURE SHUTDOWN VERSUS V_{BE}

V_{BE} (mV, npn)	V_{BE} (mV, pnp)	$T_{LATCH-OFF}$ (°C)
665	654	110 to 115
666	656	110 to 115
667	656	110 to 115
666	657	110 to 115
670	659	110 to 115
664	653	115
666	652	115
667	655	110 to 115
667	657	110
669	653	110

Reference

1. "Bipolars provide safe latch-off against opto failures," *EDN*, Dec 7, 2000, pg 190.

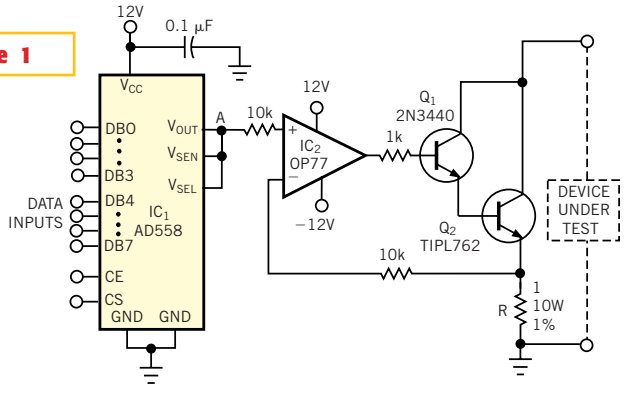
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Variable load tests voltage sources

Michele Frantisek, Brno, Czech Republic

THE CIRCUIT IN **Figure 1** serves as a variable, current-sink load for testing voltage sources. You use digital commands to set the load current of the device under test over a wide range, independently of the device under test's output voltage. The circuit comprises an AD558 DAC, IC_1, which provides a reference voltage at Point A. Practically any type of DAC converter works well in this application. The AD558 is a single-supply type with an internal reference; these features simplify the design. IC_1 generates an output voltage of 0 to 2.55V (**Table 1**). The control inputs CE and CS in IC_1 allow you to control the DAC from a microprocessor bus. If your application does not involve a data bus, connect CE and CS to ground to obtain direct access to the DAC's data inputs.

Figure 1



A simple circuit allows digital control of current, independent of voltage.

TABLE 1–DAC OUTPUT VOLTAGE VERSUS INPUT CODE

Digital input of IC_1		
Binary	**Hexadecimal**	**Voltage at Point A (V)**
0000 0000	00	0
0000 0001	01	0.01
0000 1111	0F	0.15
0001 0000	10	0.16
1000 0000	80	1.28
1111 1111	FF	2.55

The second part of the circuit in **Figure 1** consists of an op amp, IC_2, driving transistors Q_1 and Q_2. IC_2 compares the reference voltage at Point A with the voltage across resistor R. IC_2's output voltage controls Q_1 and Q_2 such that the voltage across R equals the reference voltage at point A. The voltage across R is proportional to the current from the device under test and is independent of the output voltage of the device under test. The value of R in **Figure 1** is 1Ω; thus, the circuit provides a sink current of 1A when the voltage at point A is 1V. With the values shown in **Figure 1**, you can control currents of 0 to 2.55A over a device under test voltage range of 5 to 250V. Be sure to limit the power dissipation in Q_2 to 120W.

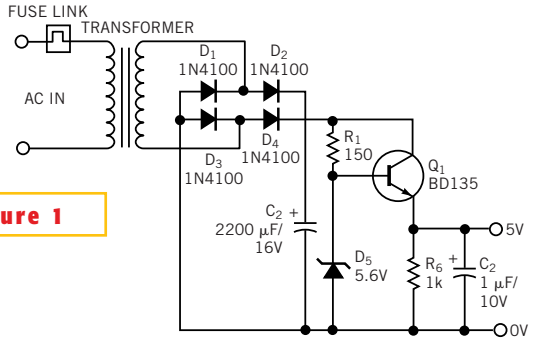
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Linear supply uses switch-mode regulation

David Magliocco, CDPI, Scientrier, France

YOU CAN USE simple circuits to implement small, regulated plug-in power supplies. In **Figure 1**, a basic and versatile 5V supply uses a zener diode and an emitter-follower transistor. You must calculate and design the transformer such that Q_1 is close to saturation at low mains voltages and nominal output current. Additionally, you must choose R_1 to ensure proper bias for the zener diode. The transistor dissipates appreciable power when the mains voltage is high. **Figure 2** shows that Q_1 must handle nearly 0.75W at nominal input voltage and output current. A TO-92 small-signal transistor, such as a BC337, is adequate for 300 mA, but a medium-power device, such as a BD135, is a better choice. To cut costs, this design uses no heat

Figure 1



This simple plug-in power supply is effective but has no current limiting.

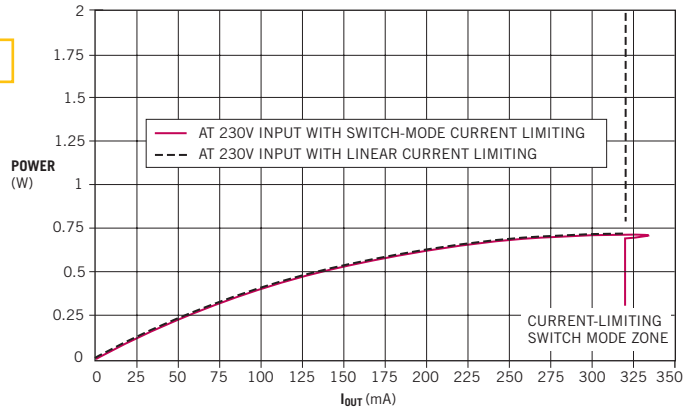
sink. Inasmuch as the circuitry is inside a nonventilated plug-in plastic case, the junction temperature of the BD135 attains temperatures greater than 100°C.

Assuming that you need current limiting, if you use a linear current-limiting circuit, Q_1 needs to dissipate nearly 2.5W in the case of a short-circuited output. The probable result is the melting of the plastic case and the failure of Q_1. To avoid that disaster, you can use switch-mode current limiting. **Figure 3** shows the circuit of **Figure 1** with some additional components, and **Figure 2** shows the benefits of the limiting. Q_1 and Q_2 act as one emitter follower, but with lower base current. A small-signal Schottky diode, BAT85, which receives its bias from R_3 and R_2, provides an approximate 0.25V reference voltage for comparator IC_{1A}'s noninverting input. The inverting input reads the voltage drop created by the output current across R_3. As long as the output current remains less than 300 mA, the output of the comparator is in a high-impedance state (open-collector), and the circuit works like a linear regulator.

If the output current reaches 300 mA, the comparator's output switches low and thus turns off Q_1 and Q_2. The current through L_1 decreases exponentially and flows through Schottky diode D_1. As the emitter of Q_1 is then at approximately 0.5V, the bias current in D_2 decreases. The voltage drop across D_2 drops by approximately 10%. As a consequence, the output current decreases until it reaches 270 mA. Then, the comparator switches back to a high-impedance state, turning on Q_1 and Q_2 and again biasing D_2 with R_1 and R_2. The current in L_1 increases exponentially until it again reaches 300 mA. L_1 is a 300-µH inductor using a powdered-iron core. **Figure 4** illustrates the current-limiting action of the switch-mode circuitry.

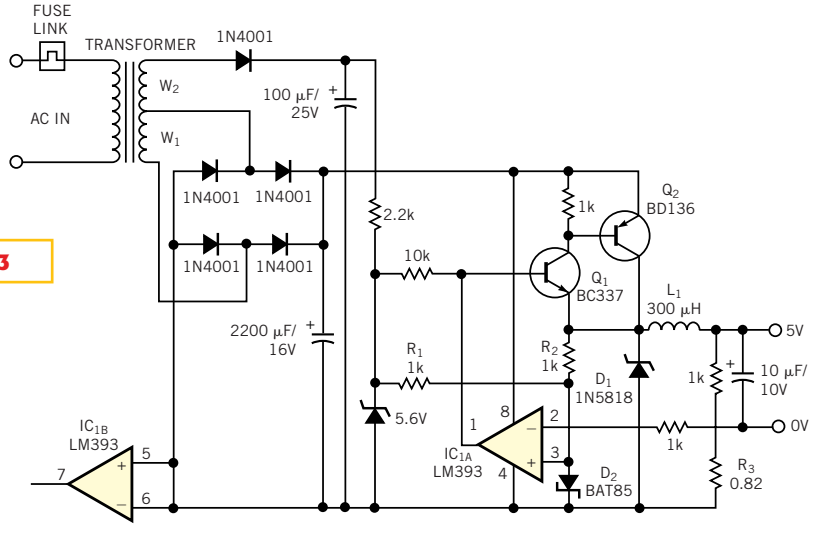
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Figure 2



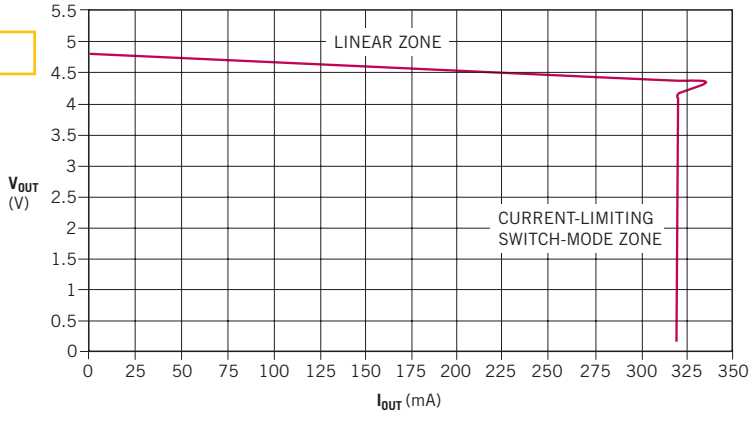
Without switch-mode current limiting, the circuit of Figure 1 can crash and burn.

Figure 3



The addition of a few components protects Figure 1's circuit against overcurrent conditions.

Figure 4



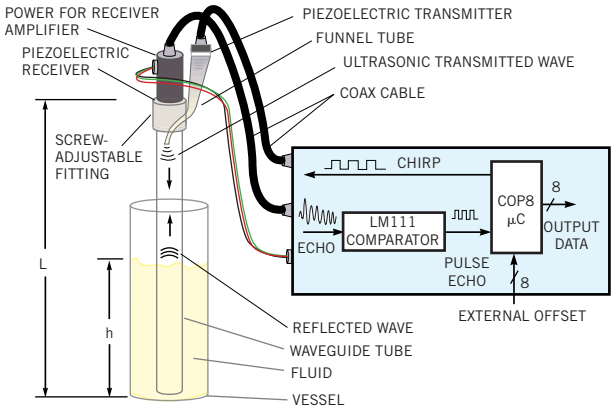
Switch-mode current limiting restricts current in Figure 3's circuit to about 320 mA.

Sonarlike method detects fluid level

Robert LeBoeuf and Eric Masse, National Semiconductor, Salem, NH

FIGURE 1 ILLUSTRATES a simple, cost-effective method of measuring the height of fluid in a column by using ultrasonic waves. Two piezoelectric transducers generate and listen to the ultrasonic acoustical wave. First, the transmitter piezo element receives stimulation from a square wave that lasts three or four cycles. This technique produces the most efficient transfer of electrical energy to acoustical energy. The acoustical wave produced at the transmitter funnels down into the waveguide tube. The wave then concentrates at the tip of the funnel tube and disperses into the waveguide tube. Part of the wave, or "crosstalk," travels up toward the receiver. The other part travels down toward the fluid. The receiver detects and ignores the crosstalk component of the wave. Part of the crosstalk wave is absorbed in the reflection from the receiver, and the remaining energy propagates down toward the fluid. The receiver then listens to the wave that echoes off the surface of the fluid. The first of these waves that strikes the receiver is the component that has propagated toward the fluid. This component is the *primary echo*. The crosstalk wave arrives shortly thereafter with a reduction in amplitude; this wave constitutes the *secondary echo*.

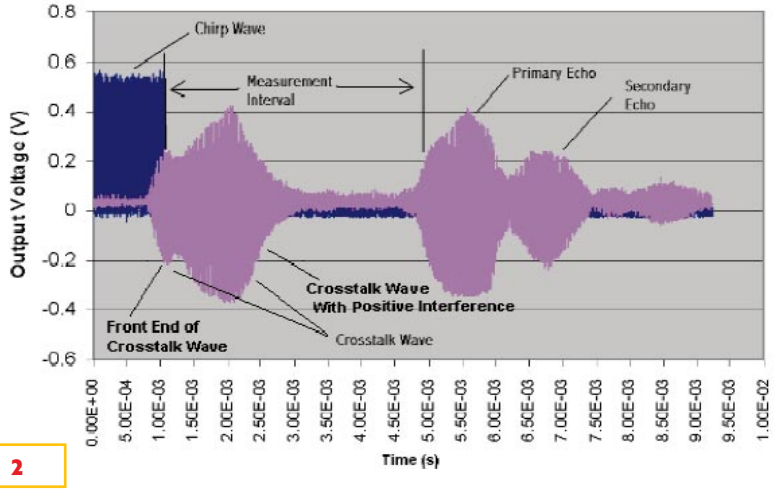
Figure 1



A fluid-level measurement system uses ultrasonic sonar principles.

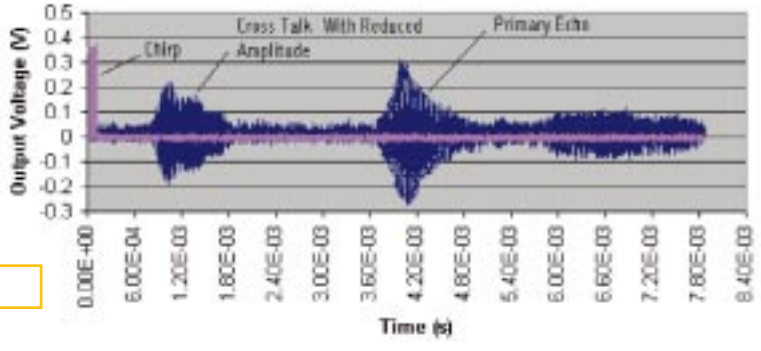
The COP8 µC measures the time it takes to propagate a wave from an arbitrary point on the surface of the fluid to the receiver. You can choose any arbitrary point; the µC can offset, or compensate for, the point with the aid of external data. The mechanical system in **Figure 1** has two major advantages. The first advantage is a cost savings in using two separate piezo elements. Piezo elements for only receivers and only transmitters are less expensive in the ensemble than a piezo element designed for both. The other advantage is the crude "mechanical diode" that the funnel tube forms. A high-intensity wave propagates from the tunnel, but the receiver recaptures only a small fraction of it. **Figure 2** shows a typical waveshape using Panasonic (www.panasonic.com) EFRTSB40KS and EFRRSB40KS piezo elements. Upon application of the

Figure 2



To make a measurement, the receiver must ignore the crosstalk wave and concentrate on the primary echo.

Figure 3



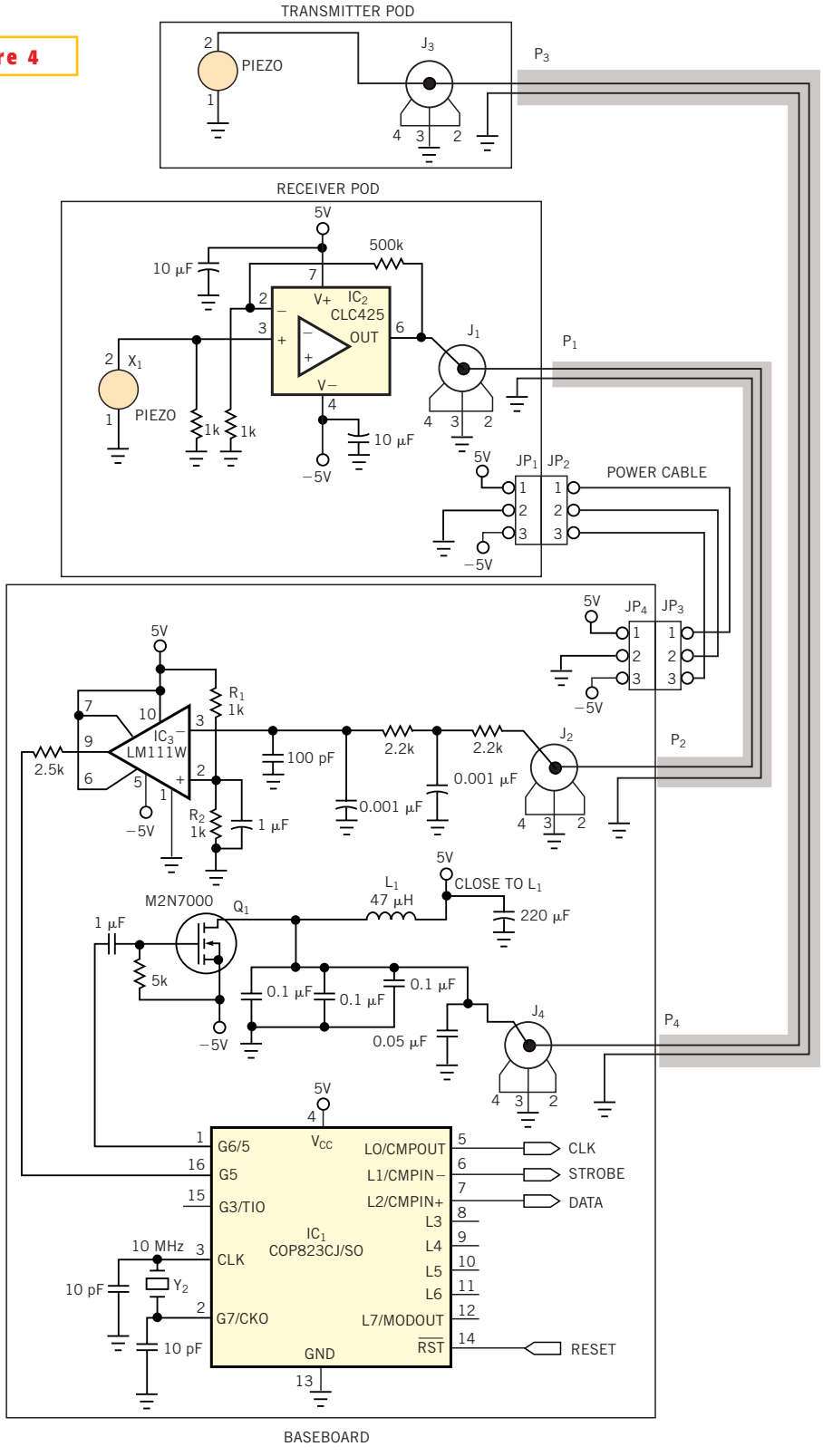
Some adjustments in the system produce a more distinct primary echo.

chirp wave, the receiver immediately detects the crosstalk wave. After an interval, the receiver detects the primary echo, followed closely by the secondary echo.

Reducing the amplitude of the crosstalk wave is convenient (**Figure 2**) This reduction allows shorter waveguide tubes, increasing the versatility of the design. One simple way of reducing crosstalk is to adjust the receiver's distance from the funnel tip, thus moving the receiver to an antinode. Another possible adjustment is to reduce the length of the chirp. **Figure 3** shows the acoustical waves after making the cited adjustments. In the circuit of **Figure 4**, the COP8, IC_1, pulses I/O port G5 at the piezo element's resonant frequency of approximately 40.3 kHz. This signal switches Q_1 and energizes the series-LC tank circuit, which has the same resonant frequency as the piezo element. This tank circuit boosts the voltage across the piezo element from 5 to 25V p-p. The receiver then detects and discards the crosstalk wave. After a short interval, the crosstalk vanishes, and the COP8 begins listening to the output of the comparator and begins counting program cycles and, thus, time.

When the echo reaches the piezo receiver X_1, IC_2 amplifies the echo 1000 times. The amplified signal then routes to the LM111 comparator, IC_3. The first echo component to reach the amplitude of the threshold set by R_1 and R_2 trips IC_3 to a logic one, and the COP8 stops counting. **Figure 5** illustrates this process. The COP8 starts listening at the rising edge of the trace labeled "COP8 Timed Interval," just after the crosstalk becomes silent. When the echo's amplitude crosses the comparator's trip voltage, the

Figure 4



The ultrasonic fluid-level measurement system uses standard, readily available components.

comparator's output switches high, and the COP8 stops timing. Armed with this timing data, converting the figures to a distance is trivial. One constraint is that the leading edge of the echo must occur after the disappearance of the crosstalk. This constraint demands an offset distance or an origin that meets the constraint. Assuming the constraint is satisfied, the distance is $d=V_{AIR}(t/2)$. The velocity of sound in air at room temperature is 345m/sec, or 0.345 mm/μsec. Solving for distance, $d=0.172t$ mm, with t in microseconds.

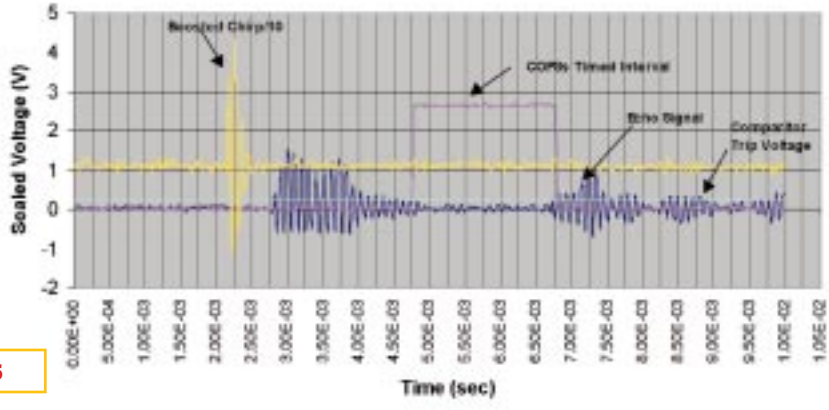


Figure 5

The COP8 doesn't start listening until its timed interval arrives after the disappearance of the crosstalk wave.

If you apply this distance to the waveform in **Figure 5**, the distance from the origin to the fluid's surface is 0.172 mm/μsec×(6750 μsec−4750 μsec)−=344 mm, or 13.54 in. The COP8 μC controls the fluid-level detector. Its first function in producing the waveform is to "chirp" the piezo transmitter. The μC uses its 16-bit programmable timer to perform this step. The timer's clock speed is the same as that of an instruction cycle, which is one-tenth of the oscillator speed (10 MHz). You set the appropriate timer control for PWM. This process produces a square wave of 40 kHz with 50% duty cycle. This figure represents the resonant frequency of the piezo elements (**Figure 5**). You should minimize the crosstalk because it is unusable for the measurement. The system should generate only three or four pulses to perform this step. More pulses create longer crosstalk and problems if the piezo receiver receives the crosstalk and primary echo at the same time. The timer in the μC starts and then experiences a short delay. During this delay, port G3 generates the square wave. The short delay limits the amount of pulses to the piezo transmitter.

While the waveform from the transmitter propagates down the tube, a delay ensures that the COP8 does not trigger during the crosstalk. The μC's timer characteristic then goes into input-capture mode. Because the LM111 has a common-emitter transistor output, the programming of port G2 uses the weak internal pullup, and then an interrupt occurs on a negative edge. At some predetermined time, the crosstalk is finished, and the timer starts and counts down until the interrupt occurs. When the negative edge arrives on the port pin, the results of the timer transfer to the upper and lower R1B. It remains on the port pin to convert the number in the R1B registers. Because the timer counts down, you need to subtract the number from the starting point to determine the actual time. The difference in fluid level determines whether you need a prescaler. Because the speed of the timer is 1 μsec, any value greater than 1 msec shows up as an over-range condition on the display. This example with a 2-ft change in fluid level uses a prescaler with a factor of four, meaning that the resolution decreases by a factor of four.

Once you determine the prescaler, you need to convert the data in the two 8-bit registers to a three-digit decimal equivalent for the three seven-segment LED displays. The conversion process is similar to counting. You set up three 8-bit registers and increment them each time the two 8-bit registers decrement. Consider the lower 8 bits of the hexadecimal number. Each time the lower 8-bit register reaches zero, the upper register decrements, and the lower bit begins at 0xFFh again. This process repeats until the hexadecimal registers equal zero. Each time a hexadecimal number decrements, a decimal-digit register increments. When the decimal-digit register reaches nine, the next decimal register increments, and so on. This operation produces three registers with a decimal-equivalent number.

The COP8 can then send the data to the display panel. You set the flags in the control register for microwire/plus serial I/O. This setting configures ports G4 and G5 as data out and clock, respectively. You configure ports G0 and G1 as output enable and strobe, respectively. You then use a look-up table to send the correct data to the display driver. The data loads into the serial-I/O register. The busy flag in the PSW register initiates the data transfer. The transfer continues at clock speed until all 8 bits complete the transfer. The process occurs for all three digits, starting with the least significant digit. You can download the assembly code for the COP8 from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases," then enter the software center to download the file for Design Idea #2718. You can find additional information about the COP8 at www.national.com/cop8.

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

design ideas

Edited by Bill Travis and Anne Watson Swager

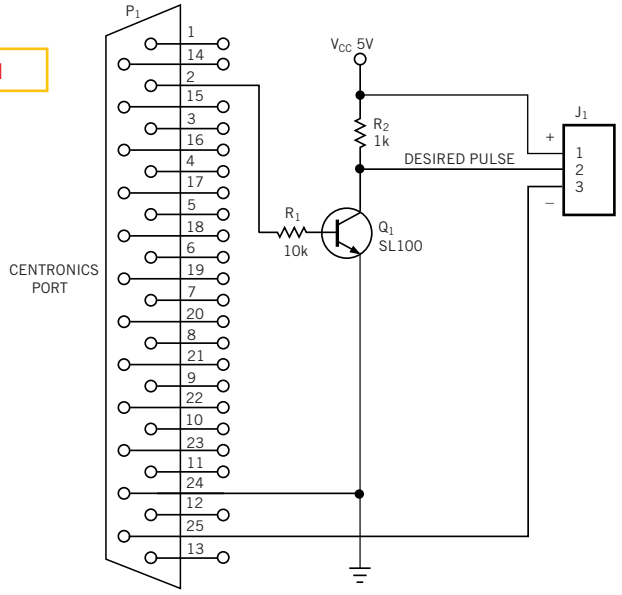
Centronics port generates narrow pulse widths

Darvinder Oberoi, CEDTI, Jammu, India

VARIABLE-PULSE-WIDTH signals are useful in control circuitry for positioning and holding purposes in robotics and power electronics. Frequently, the need arises for pulses with width less than 1 msec. Delays less than 1 msec are usually not available in most programming languages, so generating such pulses can be a problem. To generate a fractional-millisecond delay you can use a PC's 8254 16-bit timer (Counter 2), which normally controls the PC's speaker. The desired pulse is available at the PC's Centronics port (**Figure 1**) through a buffer stage, which protects the port from overload damage. Counter 2 operates at a clock frequency of 1.193181 MHz. To generate a pulse width less than 1 msec, you operate Counter 2 in mode 2 as a rate generator. You do this by setting the control-word value to $0B4_h$ and by writing this data to the control-register port, address 43_h. Initially, Counter 2 contains $FFFF_h$ at address 042_h. Bit 0 of port 61_h is at logic 1 to enable the counter, and bit 0 of the printer port (in this case, 0378_h, printer_port) is at logic 0. **Listing 1** contains the software necessary for controlling the pulse-generation process.

Setting bit 0 of port 61_h enables Counter 2. The counter decrements by

Figure 1



You can use the Centronics port on your PC to generate narrow pulse widths.

LISTING 1—TURBO C11 LISTING FOR PULSE GENERATION

(Continued on next page)

```
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#pragma inline

/*** All values are in milli Second  ***/
#define Max_Time      0.750   /* Maximum Pulse Width Time(mS)       */
#define Neutral_Time  0.500   /* Neutral Width Time(mS)             */
#define Min_Time      0.250   /* Minimum Pulse Width Time(mS)       */
#define Time_Step     0.050    /* Time(mS) Steps for Pulse Change      */
#define Counter_Count_Time 0.0008380958 /* Counter2's single count period(mS) */
#define Off_Time  10              /* Pulse OFF Duration (mS)
*/

/* Address of Printer Port    */
#define Printer_Port 0x378
void screen_display();        /* Routine for displaying Instruction */
/***** Main Program Starts Here *****/
main()
{
      char ch;
      unsigned counter_data, count_elapsed,required_count;
      float desired_time,counter_time;
      desired_time=Neutral_Time;   /* Initial time               */
      clrscr();
      screen_display();
      /* Counter2 is set in Mode 2, control word  data is b4        */
      asm mov al, 0b4h      /* Set the counter2 operation */
      asm out 043h,al
      for (;ch != 'q';)   /* Loop Till q/Q key is pressed */
      {
       if (kbhit())       /* Check, if any key has been pressed or not */
         ch = getch();
         switch (ch)        /* And take action depending upon the key pressed */
         {
```

one every 0.8380958 µsec. Before the generation of the pulse, it's necessary to compute Counter 2's required count (required_count) to generate a pulse of desired duration (desired_time). When the counter is enabled, a "while" loop reads back the counter's count through port 42_h in two read cycles. This count (counter_data) data helps to compute the counts that have elapsed (count_elapsed), and, once the required count arrives, the software exits this read loop. The software disables the counter by setting bit 0 of port 61_h to logic 0. The pulse goes low for the desired time (for example, 10 msec Off_Time). This cycle of pulse generation with a desired width repeats until you press the "Q" or "q" key. The software performs all its calculations during the pulse's off (low) time.

The software in **Listing 1** uses Turbo C++, Version 3. You can download **Listing 1** from the Web version of this article at www.ednmag.com. Using this software, you can increase or reduce the pulse width (desired_time) in variable time steps (for example, 50 µsec per Time_Step) by using the numeric keypad's keys 6 and 4, respectively. By pressing Key 5, you can fix the pulse's duration at a nominal value (for example, 500 µsec Neutral_Time). Key 8 fixes the pulse's duration at the maximum desired pulse width (for example, 750 µsec Max_Time). Key 2 fixes the pulse's minimum desired width (for example, 250 µsec Min_Time). The hardware we used for testing the software is a P-II system running at 400 MHz, with 32 Mbytes of RAM, operating in MS-DOS mode.

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

LISTING 1—TURBO C11 LISTING FOR PULSE GENERATION (Continued)

```
                case 'Q': ch = 'q';break; /* If Q key is     pressed */
                case 52: desired_time -=Time_Step; /* If 6 NUM Key is pressed */
                   ch=' ';break;
                case 54: desired_time += Time_Step;/* If 4 NUM Key is pressed */
                   ch=' ';break;
                case 53: desired_time=Neutral_Time; /* If 5 NUM Key is pressed */
                   ch=' ';break;
                case 56: desired_time =Max_Time; /* If 8 NUM Key is pressed */
                   ch=' ';break;
                case 50: desired_time =Min_Time; /* If 2 NUM Key is pressed */
                   ch=' ';break;
        }
/* Check the extreme positions and reset them in case of violation  */
        if (desired_time > Max_Time)  desired_time = Max_Time;
        if (desired_time < Min_Time)  desired_time = Min_Time;
/* Finds how many counter2 counts are required for the  desired time interval */
        required_count = desired_time/Counter_Count_Time;
        count_elapsed = 0;

/* Counter2 is loaded with initial with  count of ffff            */
         asm mov al, 0ffh          /* Load the initial count as ffff */
         asm out 042h, al          /* Done in two cycles          */
         asm out 042h,al
         asm mov al, 01h     /* Enable the counter to Start the ON period */
         asm out 061h, al
/* Desired Pulse is available at printer port's pin no.-2 */
/* The signal is  inverted by the buffer stage            */
         asm mov al, 0
         asm mov dx, Printer_Port
         asm out dx, al
/* The read the data in this loop, till desired delay is obtained        */
         while (count_elapsed < required_count)
         {
             asm in al, 042h       /*  Get LSB        */
             asm mov cl, al
             asm in al, 042h           /*  Get the MSB    */
             asm mov ah, al
             asm mov al, cl     /* Two bytes, combined to get 16 bit data */
             asm mov counter_data , ax
             count_elapsed = (0xffff - counter_data);  /* Calculate the elapsed
counts  */
         }
/*  Make pulse signal low here, after the desired counts have elapsed  */
             asm mov al, 1             /* Signal is inverted */
             asm mov dx, Printer_Port
             asm out dx, al
/* Disable the counter2 here */
             asm mov al, 00h
             asm out 061h, al
/* Print the information on the screen */
             counter_time=count_elapsed*Counter_Count_Time; /* Calculate counter
time */
             gotoxy(8,12); cprintf("%f",desired_time);
             gotoxy(26,12); cprintf("%f",counter_time);
             gotoxy(45,12); cprintf("       ");
             gotoxy(60,12); cprintf("       " );
             gotoxy(45,12); cprintf("%u", count_elapsed);
             gotoxy(60,12); cprintf("%u",required_count );
/* OFF period taken as 20ms*/
             delay (Off_Time);
      }                  /* End of FOR Loop */
      textcolor(LIGHTGRAY);   /* Set the text color and clear the screen */
      clrscr();
}                        /* END of the MAIN program here */

/**** Routine Which display the instructions & Other Information   ****/
void screen_display()
{
      textcolor(YELLOW);
      gotoxy(26,2); cprintf("Make Sure Num Lock is ON");
      gotoxy(5,11); cprintf("Desired Time(mS)  Counter Time(mS)  Count Elapsed
Required Count");
      textcolor(YELLOW);
      gotoxy(25,19); cprintf("< Press Q/q key to Quit the Program > ");
      gotoxy(15,21); cprintf("Press 4 NUM Key to Decrease Time        (By %f
ms)",Time_Step);

       gotoxy(15,22); cprintf("Press 6 NUM Key to Increase Time        (By %f
ms)",Time_Step);
      gotoxy(15,23); cprintf("Press 8 NUM Key For Maximum Pulse Width ( %f ms
)",Max_Time);
      gotoxy(15,24); cprintf("Press 2 NUM Key For Minimum Pulse Width ( %f ms
)",Min_Time);
      gotoxy(15,25); cprintf("Press 5 NUM Key For Neutral  Position   ( %f ms
)",Neutral_Time);
      textcolor(WHITE);
}
/*pulse.c*/
```

Preprocessor for rotary encoder uses PAL

David Rathgeber, Alles Corp, Toronto, ON, Canada

Rotary encoders usually provide quadrature pulses that indicate both the amount of rotation and the direction (**Figure 1**). A microcontroller can calculate the rotation direction and keep track of angular movement. Many microcontrollers' interrupt inputs, such as those on the Zilog Z86C90, can detect only a falling edge. Some with programmable edge detection, such as the Zilog Z86E30, can function with either rising or falling edges but not both. However, for maximum resolution, it is desirable to look at each rising and falling edge. In these cases, you need four inputs to read the encoder. Further, when you read the two inputs, you can assess the direction of rotation only by referring to the previous read operation. The microcontroller thus must keep track of the previous state of each input. You can use a simple PAL or other programmable-logic chip to precondition the encoder outputs to produce a single falling-edge output for each rising and falling edge for each rotation direction.

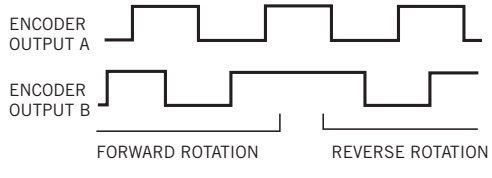
LISTING 1—PAL SOFTWARE FOR ENCODER PREPROCESSOR

```
"INPUTS
STROBE          PIN     1;
A               PIN     2;      " ENCODER
B               PIN     3;      " ENCODER

"OUTPUTS
                "DELAYED ENCODER INPUTS
ADEL    PIN 14  istype 'reg_d';
BDEL    PIN 15  istype 'reg_d';
                "FORWARD AND REVERSE INTERRUPTS
FORWARD         PIN 18  ISTYPE 'REG_D';         "FORWARD IRQ TO CPU
REVERSE         PIN 19  ISTYPE 'REG_D';         "REVERSE IRQ TO CPU

EQUATIONS
                "1. LATCH 2 INPUTS ON RISING EDGE OF STROBE
ADEL.CLK        = STROBE;
ADEL.D          = A1;
BDEL.CLK        = STROBE;
BDEL.D          = B1;

                "2. COMBINE TO FORM !RQ
!FORWARD.CLK = STROBE;

!FORWARD.D =    (A & ADEL & !B & BDEL)
                # (!A & ADEL & B &!BDEL)
                # (!A & !ADEL & B &  !BDEL)
                # (A & !ADEL & B &  BDEL)

!REVERSE.CLK = STROBE;

!REVERSE.D =    (!A & ADEL & B & BDEL)
                # (!A & !ADEL & !B & BDEL)
                # (A & !ADEL & !B & !BDEL)
                # (A & ADEL & B &  !BDEL)

END
```

This design uses an AMD PALCE-16V8Z that requires minimal supply current and costs less than $1. In addition to the two encoder inputs, the scheme requires a strobe, which occurs much more frequently than the encoder edges. This design uses the $\overline{DS}$ pin on the microcontroller, but you could probably use the crystal clock as well. Two PAL outputs are the negative-going direction outputs, and two other outputs serve as D registers. A single chip can thus condition two encoders with inputs and outputs left over for miscellaneous encoding tasks. The design works as follows: The two extra outputs serve as D-type registers, with an encoder input as the data and the strobe serving as the clock. When an input (A or B in **Figure 2**) changes, the corresponding delayed output (ADEL or BDEL) changes on the next strobe. Therefore, the delayed outputs mirror the inputs, with the delay depending on the strobe's frequency. An examination of the diagram in **Figure 2** reveals four unique patterns that relate to forward encoder rotation and four that relate to reverse rotation. **Listing 1** shows the PAL software, in Synario format. You can download the PAL software from the Web version of this article at www.ednmag.com.

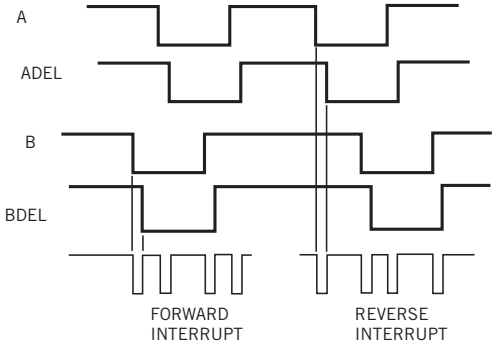
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Figure 1



To determine the direction of rotation, it's necessary to detect both rising and falling edges of the encoder outputs.

Figure 2



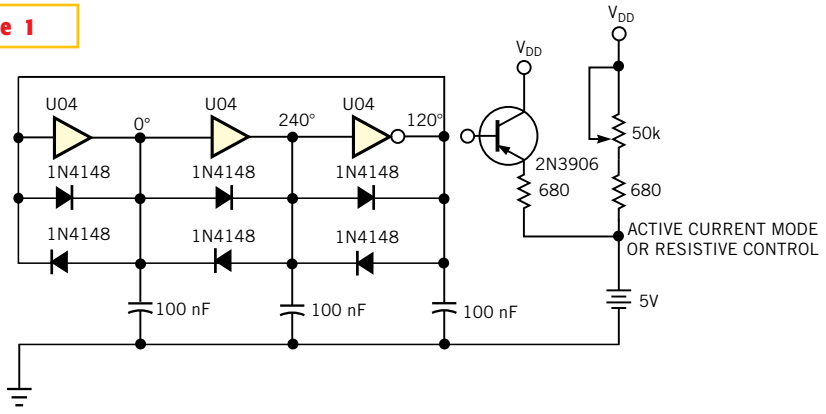
A PAL produces unique patterns for forward and reverse rotation.

Inverters form three-phase VCO

Al Dutcher, AL Labs, West Deptford, NJ

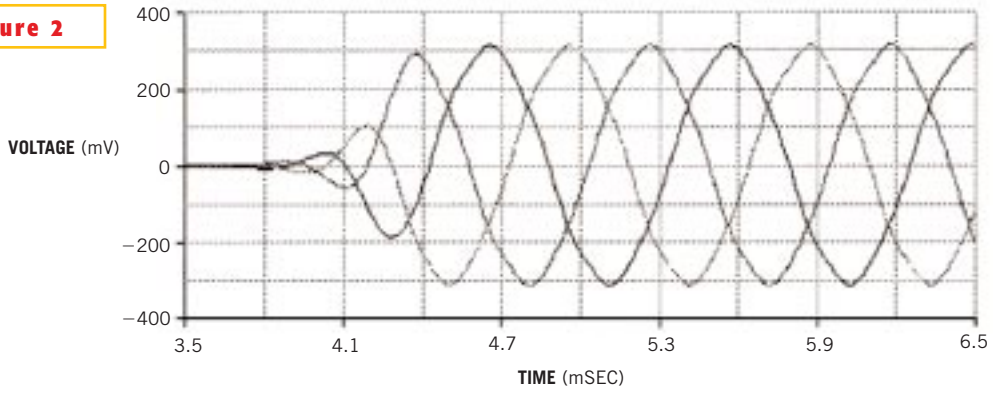
You sometimes need an inexpensive VCO that can produce evenly spaced three-phase outputs over a wide frequency range. You could use tracking all-phase filters with only one oscillator, but this technique is difficult to implement and offers limited range. Other methods, such as using a DSP, are feasible, but they're complex and expensive. The inspiration for the VCO in **Figure 1** came from Texas Instruments' application notes of years ago, detailing the use of unbuffered U-type inverters for use in ring oscillators. The application note's circuit consists of only the inverters. The circuit generates relatively squarish waveforms. Any ring oscillator's operation depends on the fact that an odd number of inversions exists around the loop. Any odd number of inverters would work. The feedback is inverting, or negative. The feedback creates an initial bias equilibrium at the transition voltage for the gates.

Figure 1



A ring-type oscillator generates 3-phase outputs over a wide frequency range.

Figure 2



The circuit in Figure 1 generates 120°-spaced, 600-mV p-p outputs.

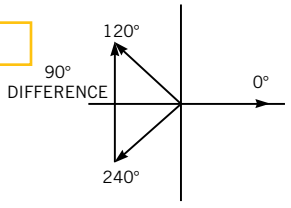
Loop gain greater than unity is a necessary condition for oscillation. Unbuffered inverters typically have a gain of 15 or thereabouts at dc and approximately 7 with capacitive loads. Three inverters thus have a total gain of more than 340, which is plenty for oscillation. At high frequencies, the inverters exhibit a lagging phase shift arising from propagation delay. Enough lagging shift added to the inversions ultimately turns the total inversion into noninversion. The circuit of **Figure 1** starts out with the 180° inversion and adds 60° of lag for 240° per stage. Three stages of 240° works out to 720° total. This figure represents two complete trips around the phase circle for noninversion. Noninversion implies regeneration, which begets oscillation. With no added capacitors, the circuit of **Figure 1** can operate at frequencies as high as tens of megahertz using 74ACU04 gates. Added capacitors can drop the frequency to usable levels. The frequency equates to $I_{DD}/3C$. For lower frequency applications, 74HCU04 gates are suitable, because they're less sensitive to layout considerations.

The diodes in **Figure 1** perform two tasks. First, they limit the excursions of the gates to 600 mV p-p, so that the gates always operate in their linear region. Second, they allow the gates to operate as current diverters, alternately charging and discharging the capacitors. The rate at which the capacitors discharge depends on the common supply current to the inverters. This rate and, hence, the oscillation frequency is proportional to the operating current. The range of frequencies over which the circuit can operate is more than 1000-to-1 (supply current from 10 μA to 10 mA). Note that at the low-current, low-frequency end of the range, the circuit cannot supply much

signal current and might need buffering. **Figure 2** shows the three-phase waveforms. The concept doesn't work well with normal ac- or HC-buffered gates, because they have far too much gain and would drive the nodes into square waves. The ACU and HCU types are somewhat obscure, but they nonetheless have multiple sources. Don't forget to ground the inputs of the remaining three gates of these hex devices. Floating inputs are verboten with all CMOS devices.

The circuit generates three equally spaced outputs. Because it generates substantially sinusoidal outputs, you can easily obtain quadrature-spaced outputs by trigonometric means (**Figure 3**). You can

Figure 3



Vector subtraction of two outputs produces a 90° quadrature signal.

connect a differential amplifier to the 120 and 240° outputs. It rejects the common-mode, 180° components. The difference between these two outputs is at 90°, and it tracks well over the range of the oscillator. You can use the same type of differential amplifier to amplify the 0° output so that the amplifier delays track at higher frequencies. In principle, you can extract any set of phase angles by the judicious adjustment of component amplitudes in the external circuits. The circuit's main drawback is that it does not have a high-Q resonator attached, so phase noise could be a problem. When you incorporate the circuit into a relatively tight PLL circuit, the performance improves considerably. The capabilities of this oscillator allow it to lock over a wide range of frequencies.

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

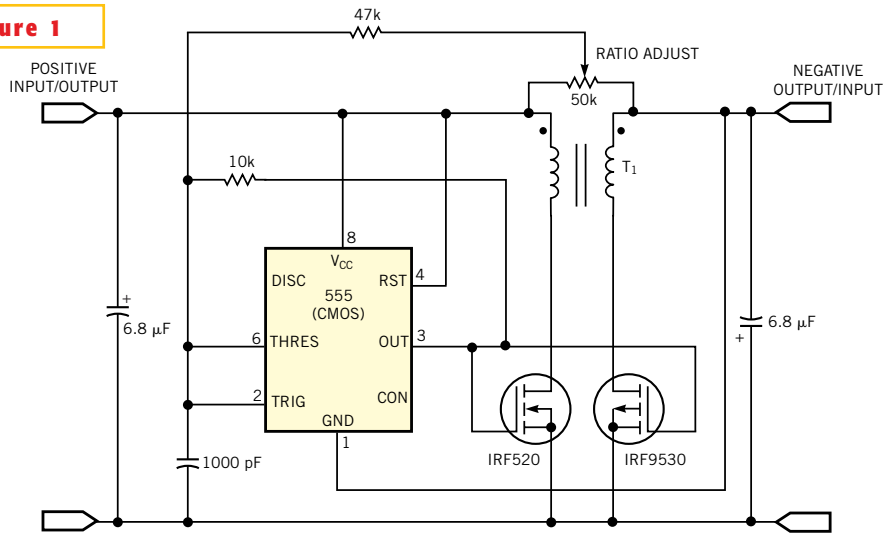
Power inverter is bidirectional

Tom Napier, North Wales, PA

If you want to swap charge in either direction between unevenly loaded positive and negative battery buses, you need an inverting dc transformer. One implementation is the symmetrical flyback converter shown in **Figure 1**. The circuit can generate a negative output from a positive supply or a positive output from a negative supply. When the circuit starts up, the substrate diode of the output FET bootstraps the output voltage to the point where synchronous switching takes over. When the gate-switching signal is symmetrical, the output voltage is approximately −95% of the input voltage, and the efficiency is greater than 80%. You can obtain voltage step-up or step-down by adjusting the switching ratio.

When I used the circuit between two 4V lead-acid batteries, a comparator adjusted the switch ratio to drive charge in the desired direction. The circuit automatically replaces charge drained from one battery to the other. In a short-battery-life application, the 2.5-mA standby current from each battery may be negligible. Using lower-gate-capacitance, FETs can reduce losses. Alternatively, you

Figure 1



An inverter circuit swaps charges between opposite-polarity batteries.

can add gates to the drive circuit to turn off both FETs whenever the battery voltages balance. The minimum input voltage is a function of the gate thresholds of the FETs. The ±9V rating of the CMOS 555 timer sets the maximum voltage. My prototype supplies approximately 100 mA.

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

design ideas

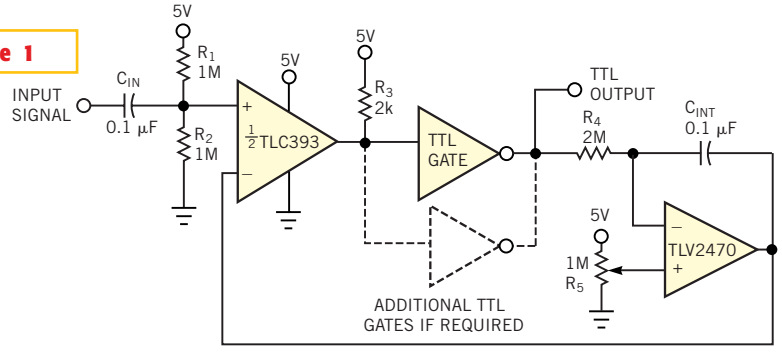
Edited by Bill Travis and Anne Watson Swager

Convert periodic waveforms to square waves

Ron Mancini, Texas Instruments, Bushnell, FL

Converting periodic waveforms to square waves is an integral part of extracting a clock signal from data, creating waveform generators, and making timing-pulse generators. Any square-wave-conversion circuit is more valuable when the square wave's duty cycle is variable and controllable. **Figure 1** shows a circuit that has these attributes and can drive several TTL-compatible loads. C_{IN} couples the input signal onto a dc level set by R_1 and R_2 (the level is $V_{CC}/2$ when $R_1=R_2$). Thus, the periodic signal at the noninverting comparator input rises above and falls below $V_{CC}/2$. The parallel value of R_1 and R_2 (R_P) and C_{IN} form a highpass filter with a −3-dB frequency of $1/(2\pi R_P C_{IN})$. Increasing R_P or C_{IN} lowers the cutoff frequency for low-frequency applications. If high-frequency noise riding on the signal causes problems, add a capacitor in parallel with R_2; this addition eliminates the high-frequency noise by creating a low-frequency filter. If the input signal is a square wave, the added capacitor integrates the square wave, thus increasing its rise and fall times. The longer rise and fall times give the circuit more control range.

Figure 1



You can obtain a square wave with 2 to 98% duty cycle with this simple circuit.

When the input signals are symmetrical, setting the dc level at $V_{CC}/2$ produces the maximum pulse-width control and duty-cycle range. Asymmetrical input signals require a different dc level, because the time durations of the positive and negative portions (with respect to $V_{CC}/2$) of the coupled signal are not equal. R_P's value must be low to prevent input-bias current from developing an appreciable offset voltage. The comparator in this design is a CMOS TLC393 version of the industry-standard LM393. The comparator weighs the dc-referenced input signal against a reference voltage from the integrator output. The comparator's output waveform is a square wave. The comparator drives a gate (or several gates if you need more output drive) through R_3. R_3 must have a low value to quickly charge the gate input during the low-to-high transition. The current the comparator can sink limits R_3's lower value.

The TLV2470 integrator integrates and inverts the output square wave and feeds it back as the reference voltage for the input comparator. If the voltage on the positive-integrator input is $V_{CC}/2$, the output square wave must be symmetrical for its average value to be $V_{CC}/2$. Adjusting R_5 to its center point yields a 50%-duty-cycle square wave. Adjusting R_5 close to ground yields a square wave that is low for most of the period, and adjusting R_5 close to V_{CC} yields a square wave that is high for most of the period. The integrator pole is at $f_P=1/(2\pi R_4 C_{INT})$. With the values shown in **Figure 1**, the 0-dB crossover frequency is 0.8 Hz. The gain of the integrator circuit is unity at 0.8 Hz, and the gain rolls off at 20 dB per decade, so the comparator's small-signal gain is not high enough to cause oscillation. The selection of the integrator pole is a trade-off between stability and control-response time. The circuit in **Figure 1** does not oscillate or multiple-switch under any conditions. It produces a square wave that's adjustable from 2 to 98% duty cycle, and it responds to 20-mV input signals.

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

Circuit improves on first-event detection

Lawrence Arendt, Oak Bluff, MN, Canada

THE CIRCUITS in **figures 1** and **2** exhibit certain advantages over the circuit shown in the Design Idea in *EDN*, "Circuit detects first event," May 3, 2001, pg 89. The n-player first-event-detection circuit offers several improvements:

- It has fewer passive components. It needs only n diodes instead of $(n^2+n)/2$ for three or more players. And, excluding the LEDs' current-limiting resistors, the circuit needs only n+1 resistors instead of 5n.
- The circuit uses less expensive ICs. The 74F74 or 4013 costs only 25% of the price of a LMC6762 (DigiKey catalog).
- The circuit offers inexpensive and simple scalability. You can easily add any number of additional player-event-detection channels to an event-detection configuration. All that's needed is that you connect the additional circuits to a common five-wire bus consisting of V_{CC}, ground, the Reset, the SwitchBus signal, and the CaptureInhibit signal. Thus, the wiring complexity is independent of n; in other words, it is O(1). Expanding the number of players for the original event-detection circuit requires additional diode-connected reset signals from each channel to all other channels, resulting in a wiring complexity that scales as (n^2-2n), or $O(n^2)$.
- The improved circuit uses D flip-flops having reset and clear pins: either 74F74s for regulated 5V supplies or, with minor circuit changes, 4013s for unregulated 9V-battery supplies. For a 74F74 implementation, the D input of flip-flop FF0 connects to logic 1. The Q output of this flip-flop drives the SwitchBus signal. The Q outputs of FF! through FFn have a diode-OR connection to the CaptureInhibit signal, which clocks the clock input of FF0. All the $\overline{\text{Set}}$ inputs for FF1 through FFn are connected through resistors to logic 1. Upon power-up or after you press the reset button, all the flip-flops' Q outputs are at logic 0 because of a pulse on the flip-flops' $\overline{\text{Reset}}$ inputs. The reset forces the SwitchBus signal to logic

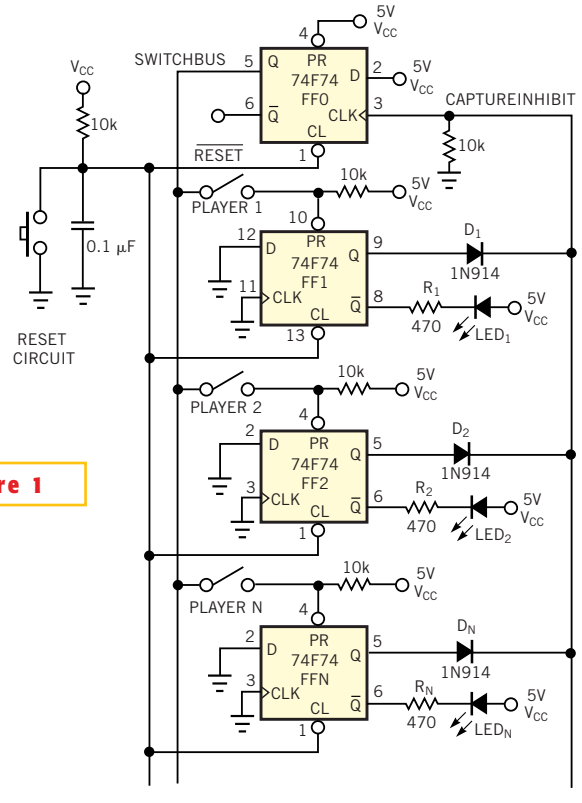


Figure 1

This first-event-detection circuit uses standard 74F74 flip-flops with a 5V supply.

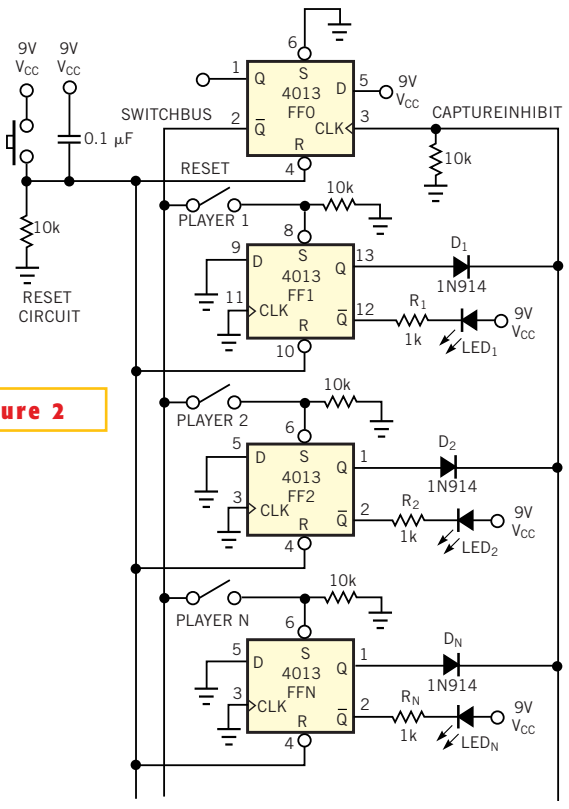


Figure 2

You can use 4013 flip-flops in a 9V-battery-powered system.

0. When you press player-event switch m, the logic-0 SwitchBus signal connects to the $\overline{\text{Set}}$ of the mth flip-flop, forcing Q_M to logic 1. Q_M now clocks FF0, forcing its Q output (SwitchBus) to logic 1. Because SwitchBus is now at logic 1, and applying logic 1 to the $\overline{\text{Set}}$ input of a 74F74 has no effect, any further switch closures by player m or any other player now have no effect.

For a 4013 implementation, the flip-flop connections are the same as for the 74F74 circuit. Upon power-up, or after you press the reset button, all the flip-flops' Q outputs are at logic 0, because of a pulse on the flip-flops' Reset inputs. The Reset signal forces the SwitchBus signal to logic 1. When you press player-event switch m, the logic-1 SwitchBus signal connects to the Set input of the mth flip-flop, forcing Q_M to logic 1. Q_M now clocks FF0, forcing its $\overline{Q}$ output (SwitchBus) to logic 0. Because SwitchBus is now at logic 0, and applying a logic 0 to the Set input of a 4013 has no effect, any further switch closures by player m or any other player now have no effect.

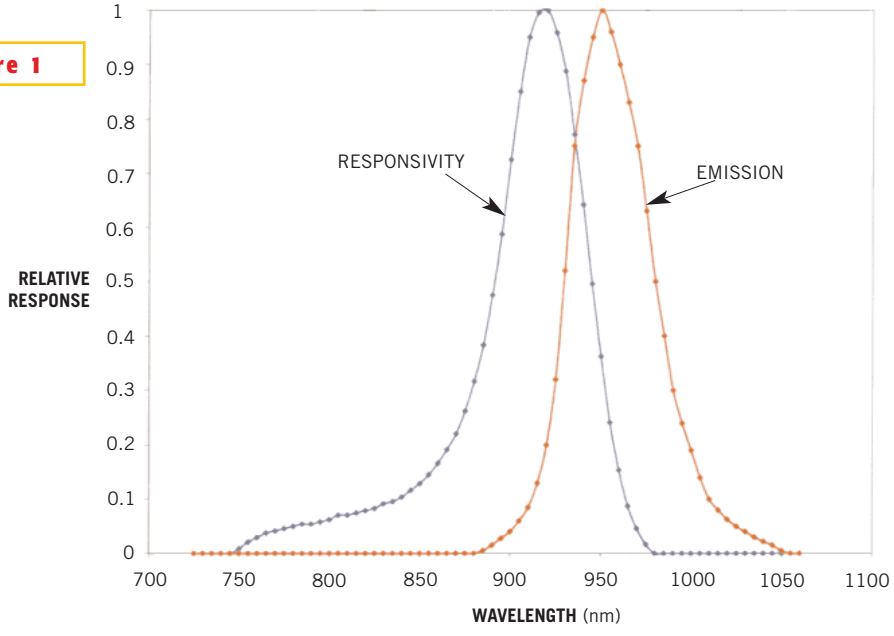
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

LED doubles as emitter and detector

Kyle Holland, LI-COR Inc, Lincoln, NE

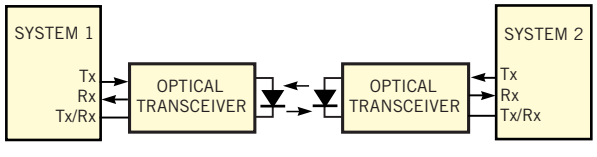
Every junction diode exhibits some degree of photosensitivity when it receives light comprising an appropriate range of wavelengths. The spectral response of a junction diode depends on a variety of factors, including material chemistry, junction depth, and packaging. The packaging of most devices aims to inhibit sensitivity to radiant flux to maintain the intended function of the device. However, some devices' packaging and construction techniques allow convenient exposure to light. The most common light-sensitive devices, photodiodes and phototransistors, sense and measure light from a variety of sources. Other light-sensitive diodes, which don't usually come to mind for light-sensing applications, are LEDs. LEDs, packaged to emit radiant flux, can serve as narrowband photodetectors. The devices lend themselves to applications in which they serve as spectrally selective photodetectors or to applications in which they act as transducers. **References 1** through **4** provide further information on optoelectronic devices.

Figure 1



Enough overlap exists between the responsivity and emission curves to make a LED useful for both transmission and reception.

Figure 2

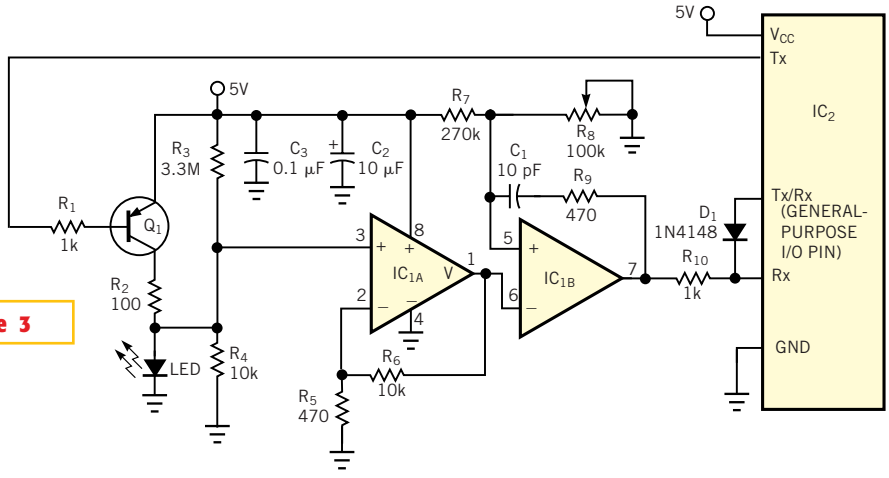


This half-duplex application uses a LED for both transmission and reception.

An LED's sensitivity to light and particularly to its emission wavelength depends primarily on the device's bulk material absorption and junction depth. For LEDs that have low bulk absorption,

photosensitivity at or near peak wavelength is low, and, as a result, the creation of hole-electron pairs is low. GaAs-based emitters with emission wavelengths of 940 nm have relatively good sensitivity at or near their peak emission wavelength, thanks to high bulk absorption. **Figure 1** shows the relative responsivity and emission spectra for an Infineon (www.infineon.com) SFH409 LED. This infrared GaAs LED has peak emission at 940 nm with a half-peak bandwidth of 50 nm. In detection mode, it peaks at 920 nm with a half-peak bandwidth of 55 nm. As **Figure 1** shows, the wavelength of peak responsivity is shorter than the peak-emission wavelength, and a fair amount of overlap exists between the two curves. Thanks to this overlap, the LED is useful as a transducer. **Figure 2** shows a half-duplex application that exploits the LED.

Figure 3



One dual op amp and a handful of components turn a LED into a dual-purpose device.

Here, the LED links two embedded systems via a fiber-optic cable or a short-distance, line-of-sight coupling path. **Figure 3** shows the transceiver circuit used in **Figure 2**'s application. The circuit can send half-duplex data between two embedded systems at rates as high as 250 kbps. The circuit comprises the LED driver, the preamplifier, and the output comparator. The LED driver drives the LED during data transmission and unhooks the Tx pin from the LED during data reception. The Tx pin connects to transistor Q_1 via base resistor R_1. When the Tx pin is in the idle state (logic 1), the quiescent current of the LED driver is zero, because Q_1 is off. Activating the Tx pin (logic 0) causes Q_1 to turn on. Resistor R_2 sets the LED's output-power level. You should set the power level to compensate for transmission losses through the communication medium and to minimize pulse-distortion phenomena in the communication link. R_2 should be 50 to 220Ω when the circuit operates from a 5V supply.

The preamplifier consists of resistor R_4 acting as a shunt current-to-voltage converter and a high-speed, noninverting voltage amplifier (IC_{1A}). Resistor R_3 provides a slight bias of a few millivolts to R_4 to keep IC_{1A} in its linear region. The op amp in this design is the high-speed dual OPA2350 from Texas Instruments' Burr-Brown division (www.ti.com). The device has rail-to-rail inputs and outputs and a gain-bandwidth product of 38 MHz, and it can operate from a single 2.7 to 5V supply. The transconductance gain of the preamplifier is a function of the values of R_4, R_5, and R_6. In the circuit of **Figure 3**, the resistor values produce a transconductance gain of approximately 220,000. The output comparator, IC_{1B}, converts the preamplifier's output signals to logic-level voltages. You set the input threshold of the comparator by adjusting resistor R_8. By properly adjusting the threshold, you can obtain good pulse symmetry for a variety of input-power conditions. The combination of R_9 and C_1 provides ac hysteresis and additional overdrive for improved comparator switching. Moreover, R_9 limits the switching current on input Pin 5 of IC_{1B} for logic one-to-zero transitions.

During data transmission, the transmitter circuitry drives the preamplifier and comparator. To prevent locally transmitted data from causing UART overrun errors, the Tx/Rx line should be at a logic-1 level. The combination of R_{10} and D_1 blocks any transition occurring on the Rx line from activating the UART's receiver. When switching between transmitting and receiving modes, the software should include a time delay to allow for preamplifier recovery. The preamplifier-recovery delay is typically less than 10 to 20 µsec. Note that for 8051-class microcontrollers with depletion-mode pullup resistors on their I/O pins, you should replace D_1 with a pnp transistor and an associated base resistor.

References

1. "The Radiometry of Light Emitting Diodes," Technical Guide, Labsphere Inc, North Sutton, NH.
2. "GE/RCA Optoelectronic Devices," GE/RCA Corp, 1987.
3. Gage, S, D Evans, M Hodapp, and H Sorensen, *Optoelectronics Applications Manual*, McGraw-Hill, 1977.
4. Sze, S, *Semiconductor Devices—Physics and Technology*, Wiley, 1985.

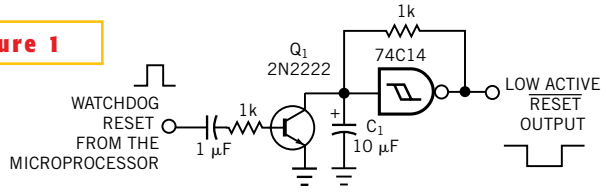
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Watchdog circuit uses ac triggering

Shyam Tiwari, Sensors Technology Private Ltd, Gwalior, India

A DC-TRIGGERED RESET of a watchdog circuit is prone to failure. If the watchdog program hangs up, then the reset signal becomes activated continuously, and the microprocessor has no way to escape the situation. We found that a simple solution uses an ac trigger to reset the watchdog circuit (**Figure 1**). We used an RC oscillator consisting of a 74C14 gate to generate active-low reset signals to the microprocessor at approximately 10-msec intervals. High-level pulses at the base of the transistor switch Q_1 reset the charging capacitor C_1 to a low level. If the watchdog trigger remains in a high state for a longer period than you want, the oscillator generates an active-low reset pulse. You may believe that a reset signal from a watchdog circuit to a microprocessor is equivalent to a power-on reset, but it is not. The warm-boot and cold-boot programs in embedded microprocessors significantly differ. Warm-boot watchdog signals are prone to hang-up. The circuit in **Figure 1** can activate the microprocessor even if the watchdog signal hangs up.

Figure 1



This watchdog circuit uses ac triggering to avoid watchdog-signal hang-up problems.

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

Use power line for baud-rate generation

Joseph Julicher, Microchip Technology Inc, Chandler, AZ

ONE COST-SAVING measure associated with 8-bit embedded microcontrollers is to use a resistor-capacitor oscillator. These RC oscillators are inexpensive, but the trade-off is low stability with temperature and voltage. In many applications, the low cost of an RC oscillator is alluring, but the application requires a stable clock source for baud-rate generation or event timing. In these cases, you can find a low-frequency, stable clock source and use it to calibrate a baud-rate generator or event timer. One source of a low-frequency, stable clock is the line voltage. This voltage is a good source 50- or 60-Hz frequency that you can easily interface to the microcontroller's 16-bit timer. By counting CPU cycles for a half cycle of the external clock, you can determine the frequency of the microcontroller's internal RC oscillator and calibrate the baud rate.

TABLE 1—SAMPLE NUMBERS FOR BAUD-RATE GENERATION

F_{osc} (MHz)	Reference frequency	Timer 1 prescale	Timer 1 counts	Calculated frequency	Desired baud rate	BRGH	SPBRG	Actual baud rate	Percent of error
4	60	16	4166.667	4,000,000	9600	1	25	9615.38	0.16
4.4	60	16	4583.333	4,400,000	9600	1	28	9482.75	−1.22
3.6	60	16	3750	3,600,000	9600	1	22	9782.6	1.90

The PIC16F627 flash microcontroller can benefit from this technique. This device has an internal 4-MHz RC oscillator. You can create a simple capacitor-coupled circuit to allow the microcontroller to see the pulses from the power line. You can power this PIC microcontroller from a separate circuit that includes voltage regulation (see Tech Brief 008 at www.microchip.com concerning a transformerless power supply). The power line supplies a solid 50- or 60-Hz reference frequency. You can use the 16-bit Timer 1 to time the internal oscillator. As the internal RC time constant drifts, the timer count changes, and you can use the value to determine new values for the baud-rate generator. If you adjust the baud-rate generator appropriately, you can maintain the baud rate to ±2% of the desired value. The incremental cost may be less than the cost of a crystal or ceramic resonator. You can also use this technique to periodically learn the value of the internal RC oscillator to calibrate time captures of external events. The following is an example of the technique: The base INTRC value is 4 MHz, and the power-line frequency is 60 Hz. Set BRGH for a baud clock 16 times the baud rate, and set Timer 1 prescale to 16. The desired baud rate is 9600.

The formula we use is ((Timr1 value×prescaler×reference frequency×baud multiplier))−1=SPBRG value. We used a spreadsheet to run some sample numbers, and the results are in **Table 1**. To use the technique with a PIC microcontroller, simplify the math to (Timr1/160)−1=SPBRG. You will find that 9600/(16×60)=160. This simplification causes a slight error; the rounded off value of SPBRG becomes 27 instead of 28. The error at 4.4 MHz becomes 2.3% instead of −1.22%.

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

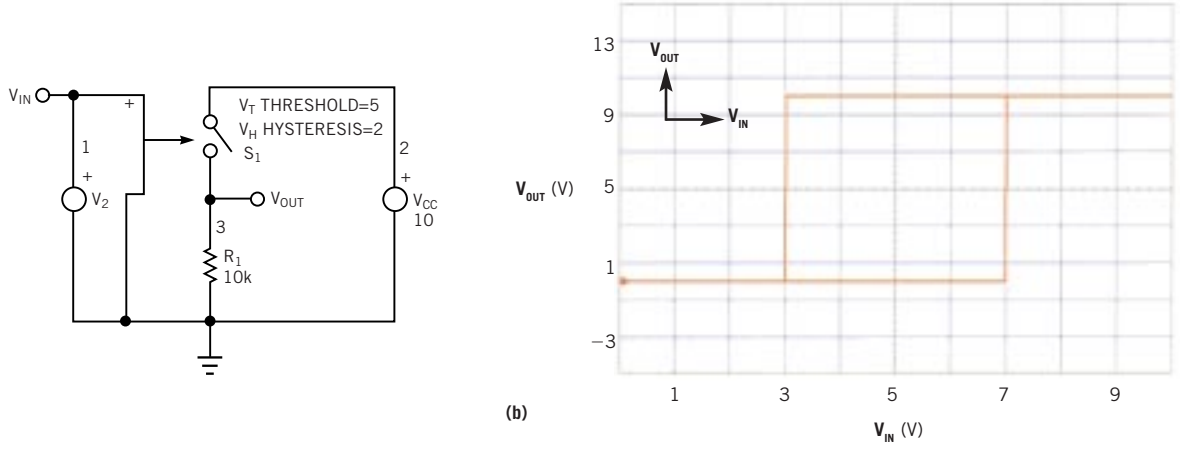
Inline equations offer hysteresis switch in PSpice

Christophe Basso, On Semiconductor, Toulouse, France

Smooth-transition switches are convenient devices in many Spice-based simulators. Their action can greatly ease the convergence process. Unfortunately, these devices lack inherent hysteresis, a helpful feature used to build UVLO (undervoltage-lockout) systems, oscillators, and other systems. Intusoft's (www.intusoft.com) IsSpice4 not only provides users with smooth-transition devices, but also adds the Berkeley Spice primitive switch featuring adjustable hysteresis. **Figure 1a** shows how to wire this component in a simple comparator architecture. **Figure 1b** shows the resulting V_{OUT} versus V_{IN} curve. In Cadence's (www.cadence.com) PSpice, the S_1 primitive switch implements a soft transition between R_{ON} and R_{OFF}, but V_{ON} and V_{OFF} are simply the final levels that reflect the specifications for R_{ON} and R_{OFF}. To incorporate some hysteresis, you just need to add some analog-behavioral-model sources to help tailor our switching events. **Figure 2** shows how you can derive the new device with adjustable hysteresis.

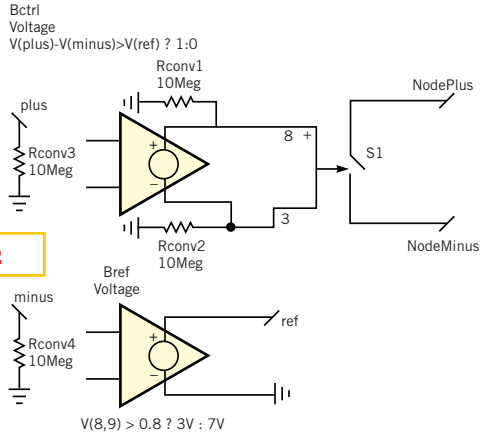
Listing 1 is the complete PSpice netlist for the switch. The inputs V(plus) and V(minus) route the switch-control signal to the Bctrl behavioral element. (A b element in IsSpice4 becomes a "value" E source in PSpice.) When the switch is open (Bctrl delivers logic 0), the reference node (node *ref*) becomes armed at the highest toggling point (7V in **Figure 2**'s

Figure 1



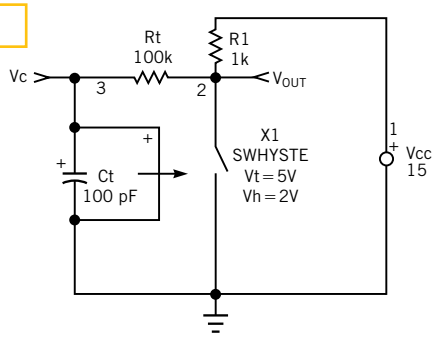
This switch circuit (a) toggles at 7V and resets at 3V, yielding 4V hysteresis (b).

Figure 2



Some analog-behavioral-model sources and a primitive S switch yield a switch with adjustable hysteresis.

Figure 3



These results clearly show the 4V hysteresis in Figure 2's subcircuit.

example). When the input voltage increases, it crosses the *ref* level, and Bctrl switches high. Switch S_1 closes and authorizes the current to flow in its terminals through R_{ON}. At this time, the Bref source has detected that S_1 is closed and now modifies its reference node to the second level (3V in the example). When the input voltage drops, it crosses the 7V level, but no action takes place, because *ref* has changed to 3V. When the input voltage finally crosses 3V, S_1 opens and applies R_{OFF} between its terminals. **Figure 3** shows the PSpice simulation results, confirming the 4V hysteresis. **Figure 4** shows how to build a simple RC test oscillator, using the PSpice-derived hysteresis. You can download the PSpice netlist from the Web version of this article at www.ednmag.com.

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

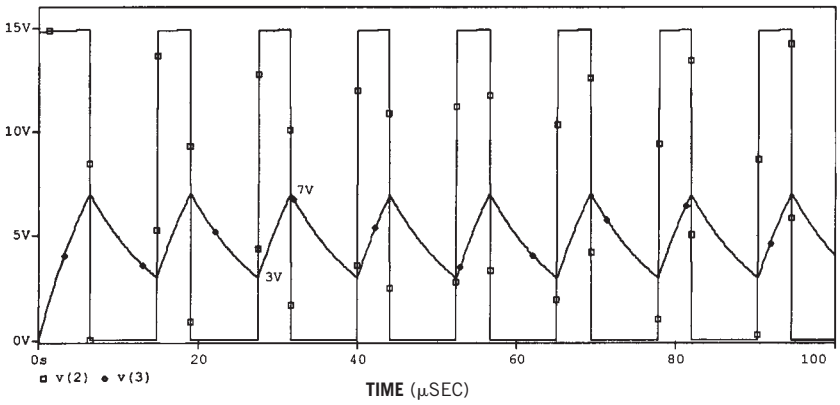


Figure 4 A switch with hysteresis and a few passive elements produce an RC oscillator.

LISTING 1–PSPICE NETLIST FOR SWITCH WITH HYSTERESIS

```
.subckt SWhyste NodeMinus NodePlus Plus Minus PARAMS: RON=1 ROFF=1MEG VT=5 VH=2
S5 NodePlus NodeMinus 8 0 smoothSW
EBcrtl 8 0 Value = { IF ( V(plus)-V(minus) > V(ref), 1, 0 ) }
EBref ref1 0 Value = { IF ( V(8) > 0.5, {VT-VH}, {VT+VH} ) }
Rdel ref1 ref 100
Cdel ref 0 100p  IC={VT+VH}
Rconv1 8 0 10Meg
Rconv2 plus 0 10Meg
Rconv3 minus 0 10Meg
.model smoothSW VSWITCH (RON={RON} ROFF={ROFF} VON=1 VOFF=0)
.ends SWhyste
```

design ideas

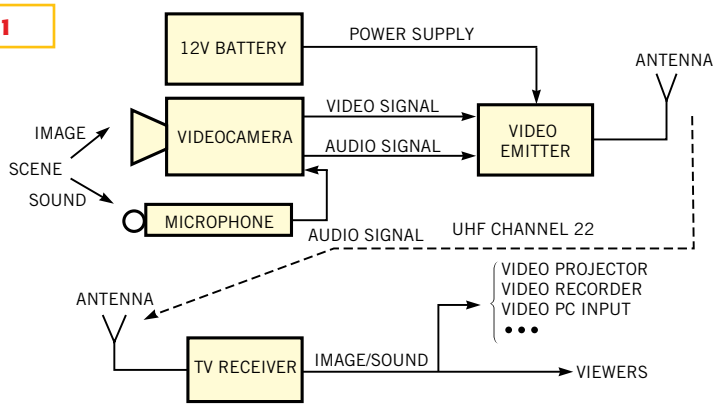
Edited by Bill Travis and Anne Watson Swager

Video emitter uses battery power

JM Terrade, Clermont-Ferrand, France

THE BLOCK DIAGRAM in **Figure 1** shows how to make a cable-free, direct-video system. The system allows users to walk from booth to booth at an exhibition to interview people and to display the interviews in real time on three screens at key locations. You can use the small and simple system each time you need to capture image and sound on the run. **Figure 2** (pg 96) shows a detailed schematic diagram of the video system. The system provides no stereo audio but rather mixes together right and left sources. IC_{2A} acts as an inverter/adder. At Point C, the ac signal represents the sum of the left and right channels: $V_C = -R_1(V_A/R_2 + V_B/R_3)$. With the same value for the three resistors, $V_C = (V_A + V_B)$. C_8 and C_9 block any dc voltage at points A and B. IC_{2A} works from a single 12V supply but needs a continuous bias voltage to provide a positive and negative swing around 6V. R_4 and R_5 create a 6V bias source for both IC_{2A} and IC_{2B}. IC_{2B} acts as an inverting voltage amplifier with a gain of P_1/R_8. With the values shown, you can adjust the gain as high as 4.7. You can adjust P_1 for audio gain as high as 13 dB. C_{10} blocks the 6V dc at Point D, so only the ac audio voltage is present at the audio input of IC_4. The LM358N works well from a single supply, but when the output voltage is close to 0V, it needs some help to avoid signal distortion. Pulldown resistors R_6 and R_7 minimize the distortion.

Figure 1



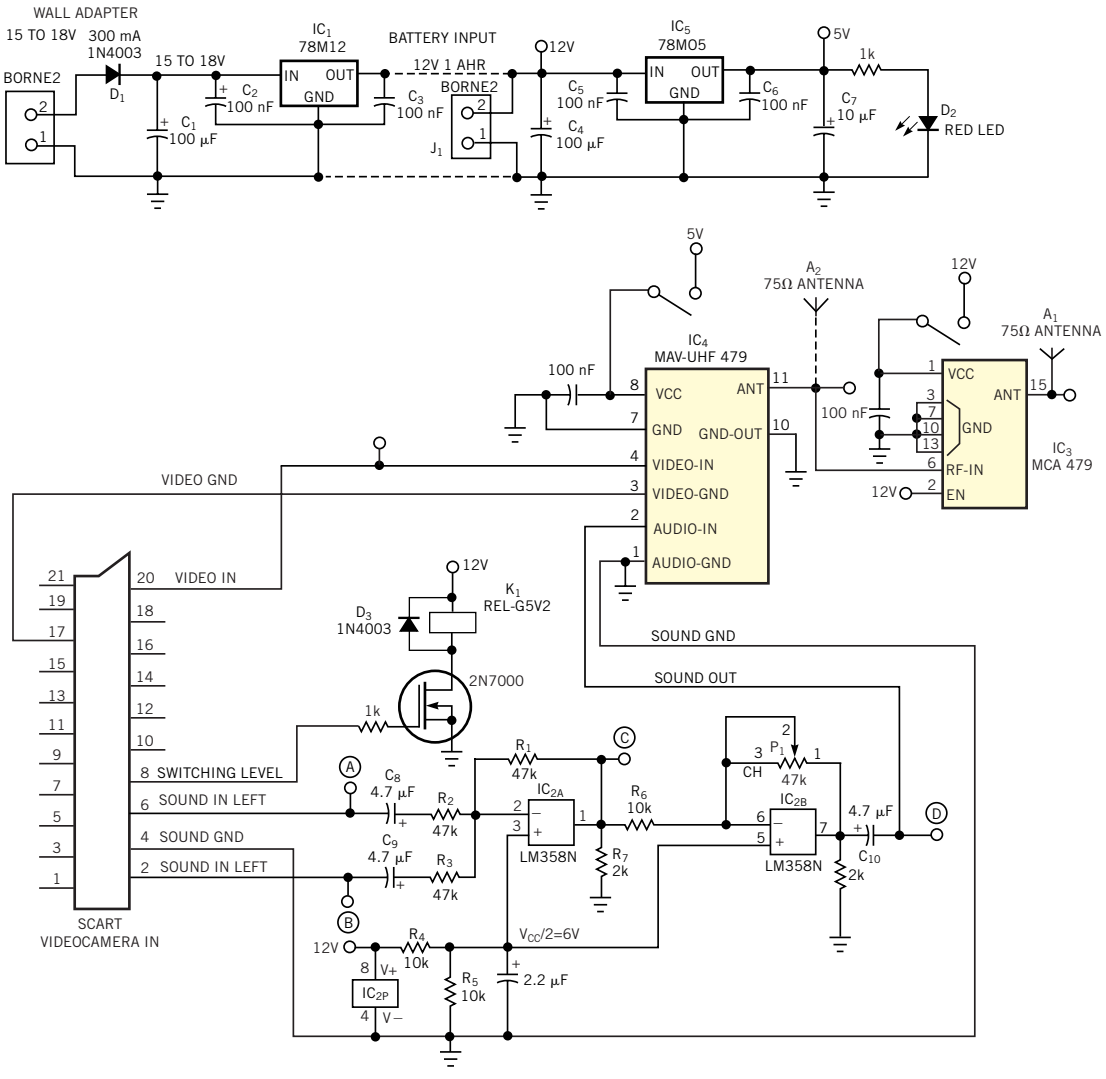
A wireless, battery-powered video system uses UHF Channel 22 to transmit signals to video receivers.

IC_4 is a video-emitter IC from Aurel (www.aurel.it) that works in the UHF band at 479.5 MHz (UHF Channel 22). Its output power is 2 mW to a 75Ω antenna, A_2. Typical supply current is 90 mA. The signal from the videocamera connects directly to IC_4's input. If you need more power, you can add IC_3, also from Aurel. This IC works in the same frequency band as IC_4 and boosts power to 19 dBm in the 75Ω antenna. Both IC_3 and IC_4 are available in small, single-in-line packages. A Switching Level signal from Pin 8 of the SCART video connector is present when the videocamera is on. The current consumption of IC_3 and IC_4 is 90 (5V) and 100 mA (12V), respectively. To reduce power consumption, a dual-contact relay, K_1, connects IC_3 and IC_4 to the supplies only when the camera is on. When the videocamera is off or disconnected, the supply current decreases to only a few milliamperes.

IC_5 and C_4 through C_7 provide a 5V supply to IC_4. IC_1 and C_1 through C_3 provide a stable 12V supply to IC_3 and the LM358. You can connect a 12V battery directly to J_1. IC_3's data sheet specifies a supply level of 11.4 to 12.6V, but tests show that the IC works properly if the supply is higher than 11V. The total 200-mA current consumption yields approximately three-hour battery life with 12 AA cells. If you use a switching power supply, you would obtain longer battery life, but you need to take filtering measures to avoid interference with the video path. If an ac outlet is available, you could use an 18V, 300-mA wall adapter to replace the battery.

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

Figure 2



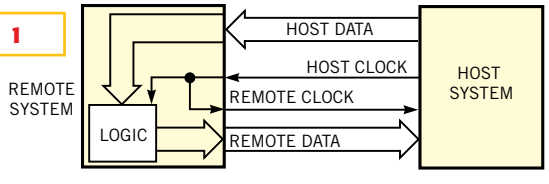
In this video transmitter, 12 AA cells provide approximately three hours of operation.

Circuit avoids metastability

Jonathan Eckrich, Adaptivation, Sioux Falls, SD

Consider a computer system that has a host processor connected to a remote-I/O subsystem (**Figure 1**). The host clock treats the I/O system, which is located far from the main hardware, as a slave. Because of the transmitters, receivers, remote-system logic, and cable length, the data the host receives has a dramatic latency. This latency can be larger than the clock period. If the length of the cable is indeterminate, then the latency is also indeterminate. The problem with such latency is that receiving registers in the host system might clock in the data from the remote system while some bits are changing. The result is that some data may be corrupt, or, worse, the input registers may go into a metastable state. The circuit in **Figure 2** prevents clocking bad or changing data. It does so using only general-purpose, "jellybean" logic. The key is to remote-clock back to host. This action allows XOR gate IC_{1A} to compare the phase difference between the host clock and the delayed clock.

Figure 1



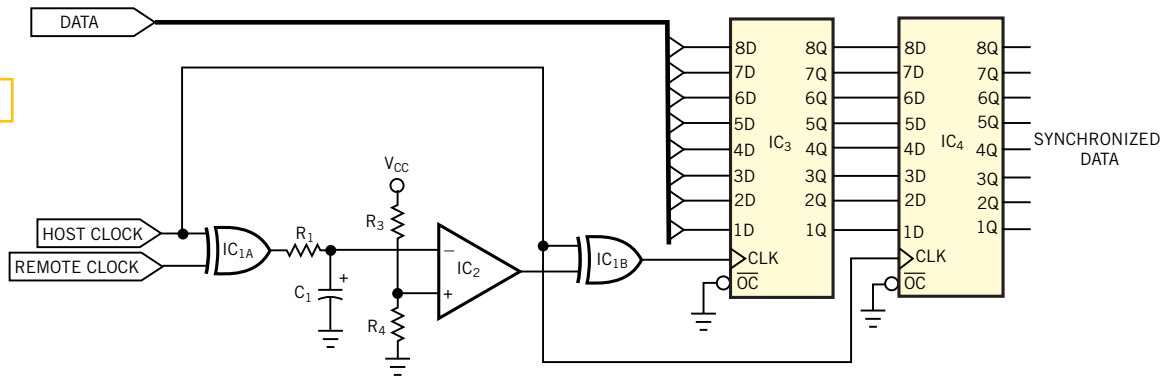
At or near 360 or 180° phase difference between the two clocks, this remote-I/O system is subject to metastability.

When the two clocks are nearly in phase, the duty cycle of IC_{1A}'s output is close to 0%. When the two clocks are close to 180° out of phase, the duty cycle approaches 100%. Whatever the duty cycle is, it is constant during normal operation. The only way it can change is for the cable length between the two systems to change. R_1 and C_1 form a lowpass filter. Set R_3 equal to R_4 so that the reference voltage is at midpoint. IC_2 and IC_{1B} then select whether to clock register IC_3 on the rising or falling edge of the host clock. IC_4 ensures that the data changes consistently with the rest of the host system. **Figure 3** shows a (delayed) remote clock that is nearly 360° out of phase with the host clock. If the host were to clock in the data on the rising edge of its clock, metastability would become a concern. You can simply clock in the data on the falling edge of the host clock, but this solution yields the same problem if you choose a new cable with a different length.

Without any analytical effort on the designer's part, the circuit in **Figure 2** automatically selects which clock edge to use. Note that comparator IC_2 can be a low-speed part, because it operates at dc only. Note also that if the two clocks are 360±90° out of phase, the circuit uses the falling edge of the clock. If they are 180±90° out of phase, the circuit uses the rising edge. If the R_1C_1 time constant is too low, the resulting ripple can cause the output of the comparator to be unstable. You could use a comparator with hysteresis to reject the ripple. Some instability of the comparator's output is acceptable, because you can safely use either the rising or the falling edge for most latencies. You need stability only when the clock is near 360 and 180° out of phase, so you have little to lose by using a large R_1C_1 time constant to present a dc voltage to the comparator's input.

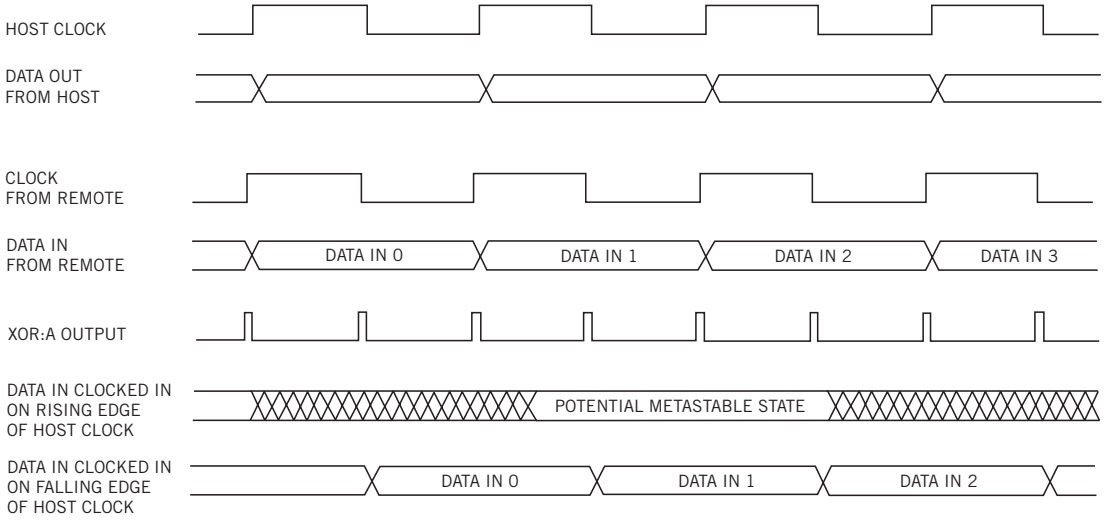
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Figure 2



This circuit automatically chooses between the rising and the falling edge on the host clock for clocking in data.

Figure 3



Clocking data on the wrong edge can result in metastability; the circuit in Figure 2 selects the right edge.

Microphone uses "phantom power"

Bruce Trump, Texas Instruments, Tucson, AZ

THE ELECTRET MICROPHONE capsule is similar to those commonly used in telephones, cassette recorders, and computers. The element functions as a capacitor with a fixed trapped charge. Sound pressure moves a diaphragm, producing variations in the capacitance. This action produces an ac-output voltage with an extremely high source impedance. A FET inside the capsule uses an external-resistor drain load (**Figure 1**). R_1 and R_2 provide an appropriate load impedance and voltage from the 10V supply. The basic performance of this simple capsule is excellent, but it requires further signal processing to conform to professional phantom-powered-microphone standards.

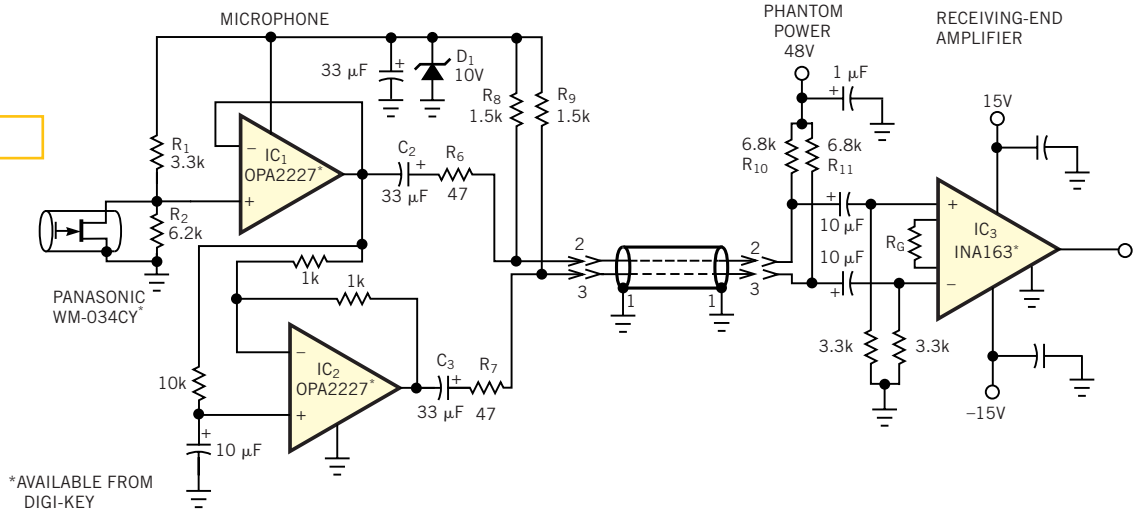
The output of a phantom-powered microphone is a low-impedance differential signal. IC_1 is a simple voltage buffer that provides low-impedance drive for one output. IC_2 is a unity-gain inverter that derives its drive from the output of IC_1. Bias for the noninverting input of IC_2 comes from a heavily filtered output of IC_1. We selected the dual op-amp IC_1/IC_2 for its low noise and low distortion properties. R_6 and R_7 provide immunity from long-line capacitance, RF interference, and transients that occur when you "hot-plug" the microphone into a live phantom-power source. The amplifier outputs use ac coupling, C_2 and C_3, to the microphone's output terminals to block the dc phantom-power voltage on the audio lines. Differential-output voltage capability is limited to approximately 2V p-p because of the limited power supply available to drive the op-amp output currents. This level is adequate, because it corresponds to an extraordinary sound level beyond the linear range of the capsule.

Phantom-powered microphones derive power for their active circuitry from the receiving-end circuit through the same leads that transmit the audio signal. The 48V phantom-power supply couples through two 6.8-kΩ resistors, R_{10} and R_{11}, to both signal lines. This coupling allows the microphone's low output impedance to drive a differential ac signal on the relatively "soft" impedance of the phantom supply voltage. In the microphone, power comes from the signal lines through resistors R_8 and R_9. Zener diode D_1 regulates the voltage. These resistors also provide a soft impedance on the balanced line, allowing the outputs of IC_1 and IC_2 to inject their differential ac-output signal. You can locate the microphone hundreds of feet from the receiving-end phantom power and amplifier and still obtain excellent performance.

The receiving-end amplifier, IC_3, is a low-noise instrumentation amplifier with three internal op amps. Its configuration and laser-trimmed resistors provide excellent CMR (common-mode-rejection) properties. The high CMR rejects noise and power-line hum that appear equally in both signal lines. Low noise (1 $nV/\sqrt{Hz}$), though unnecessary for high-output microphones such as those described here, is necessary in professional-audio equipment to accommodate the use of low-output ribbon and dynamic microphones. These microphone types are strictly passive electromechanical generators and do not require a power source. Phantom power earns its name from the fact that these microphone types "float" at 48V without harm. The electret capsules are available in various sizes and physical configurations. They include both omnidirectional and directional (cardioid) types. Directional capsules have a vent in the rear; you must mount them with free access to both the front and the back to obtain proper characteristics.

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

Figure 1



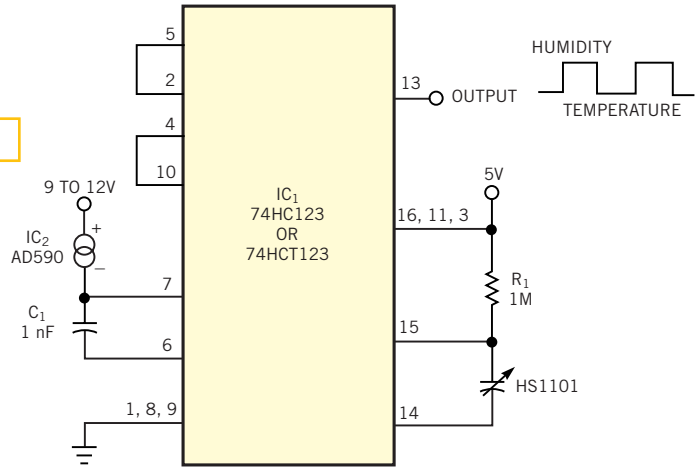
This microphone system derives its power from the receiving-end circuitry through the leads that carry the audio signal.

Measure humidity and temperature on one TTL line

Shyam Tiwari, Sensors Technology Pvt Ltd, Gwalior, India

By combining the responses of an Analog Devices (www.analog.com) AD590 temperature sensor and a Humirel (www.humirel.com) HS1101 humidity sensor, you can generate a single TTL-level signal containing information from both sensors (**Figure 1**). This design uses a 74HC123 monostable multivibrator, IC_1, to form a free-running oscillator. The AD590 current source (1 µA/K), IC_2, and a fixed 1-nF capacitor, C_1, control the timing of the first monostable multivibrator in the 74HC123. Another monostable multivibrator uses a fixed 1-MΩ resistor along with the capacitive output of the HS1101 (172 pF at 0% relative humidity and 222 pF at 100% relative humidity) for its timing. Combining the two monostable multivibrators creates a free-running oscillator that produces a single-line signal from both sensors. The high- and low-level pulse widths carry the information related to the sensor signals. The AD590 circuit displays pulse-width reduction with rising temperature, because of its increased output current with higher temperatures. The HS1101 circuit displays increased pulse width with rises in humidity levels. The circuit in **Figure 1** represents a simple method of transmitting signals from analog sensors by digital rather than analog means. The technique eliminates noise in signal transmission over long distances. You could add an optoisolator in the output path if you need, say, 1500V isolation.

Figure 1



A monostable-multivibrator IC provides temperature and humidity information in one TTL signal.

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

Low-cost anemometer fights dust

Jim Christensen, Maxim Integrated Products, Sunnyvale, CA

As higher levels of power dissipation underscore the need for cooling, more and more fans are finding their way into small electronic enclosures. The dust that fans pull into these enclosures can, however, cause major problems for high-reliability systems. By coating heat sinks and electrically charged components, the dust acts as a blanket that raises the effective thermal impedance between the components and the air. A simple way to combat this problem is to place a disposable filter on the air intake. If you fail to replace the filter on a regular basis, however, it can become clogged and act as an air dam, a condition that is worse than the original problem. Trying to sense a clogged filter by sensing the fan's rotation with tachometer signals is useless, because fan rotation is not directly related to airflow. You can detect poor filter maintenance by determining the actual airflow with a "hot-wire" anemometer, but most electronic anemometers are costly and bulky. As an alternative, you can create an SMBus/I²C anemometer using an I/O expander, a few inexpensive switches, and a low-cost remote-temperature sensor (**Figure 1**).

TABLE 1—FAN VOLTAGE VERSUS COOLING TIME

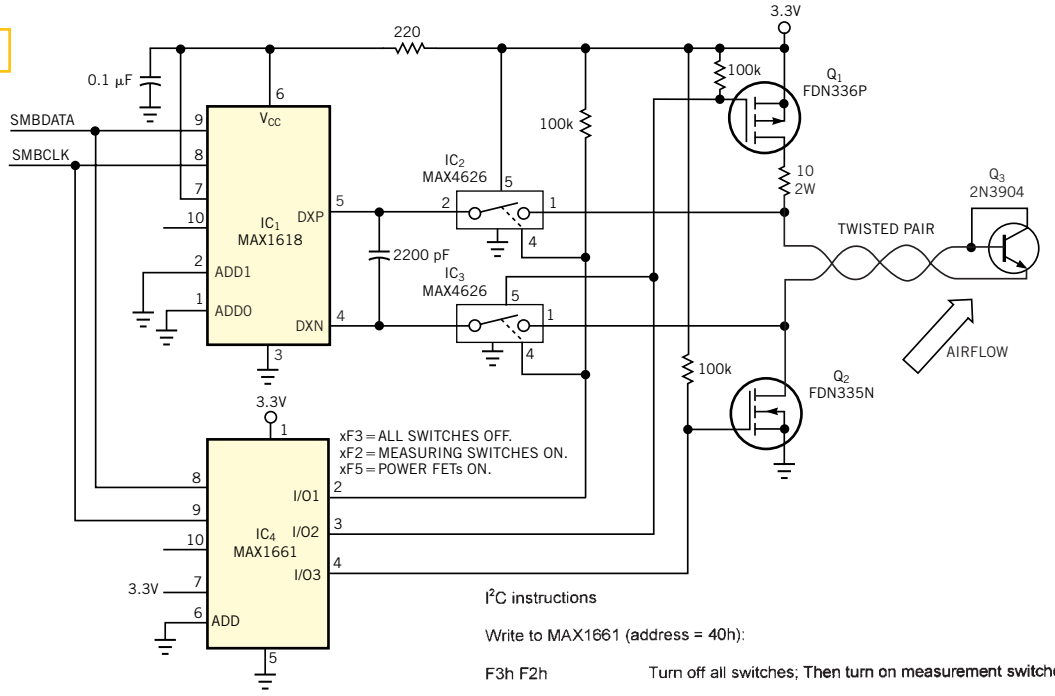
Fan voltage (V)	Cooling time (sec)
12	30
8	47
6	60
0 (no fan)	84

Use the SMBus I/O expander, IC_4, to turn off MOSFETs Q_1 and Q_2 and to turn on the analog switches IC_2 and IC_3. Measure the ambient air temperature with no preheating of Q_3. Then, to apply current for heating Q_3, turn off IC_2 and IC_3 and turn on Q_1 and Q_2. Allow an approximate five-minute "soak" to reach temperature equilibrium. (The exact heating time necessary for equilibrium depends on the setup; you must determine it by experiment.) At equilibrium, remove power from Q_3 by turning off Q_1 and Q_2, and turn on analog switches IC_2 and IC_3 to make temperature measurements. Airflow directly relates to the rate at which the temperature drops; you can determine it by noting the time required for the transistor to return to within 1° of its original temperature. The temperature sensor injects a small current into the base junction, so careful layout is important to keep noise off the DXP and DXN lines.

If you mount the remote transistor in an air channel, the use of twisted-pair wire allows distances to 12 ft. **Table 1** shows fan voltage (airflow) versus cooling time for a sensor placed approximately 12 in. away from a fan running at full speed (12V), medium speed (8V), low speed (6V), and zero speed. Soak times as long as 30 minutes do not significantly alter the times. The circuit draws approximately 200 mA when Q_3 is heating. If this power dissipation poses a problem, you can lower the measurement frequency to hourly or even daily cycles, because changes in airflow occur slowly over time. You can also schedule the measurements during times of low system activity, when overall power use is low.

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

Figure 1



I²C instructions

Write to MAX1661 (address = 40h):

F3h F2h	Turn off all switches; Then turn on measurement switches

Write to MAX1618 (address = 30h):

09h 48h	Write Configure; One shot mode
0Fh	Write one shot command
W 01 R ??h	Read ambient temperature

Write to MAX1661

F3h F5h	Turn off all switches. Then turn on power FETs

Soak for 5 minutes

Write to MAX1661

F3h F2h	Turn off all switches; Then turn on measurement switches

Write to MAX1618

0Fh	Write one shot command
W 01 R ??h	Read temperature

Loop on last two statements and count the time it takes for the temperature to drop to the initial ambient temperature +1°

This anemometer measures airflow by heating Q_3 and then noting the time for Q_3 to return to its original temperature.

design ideas

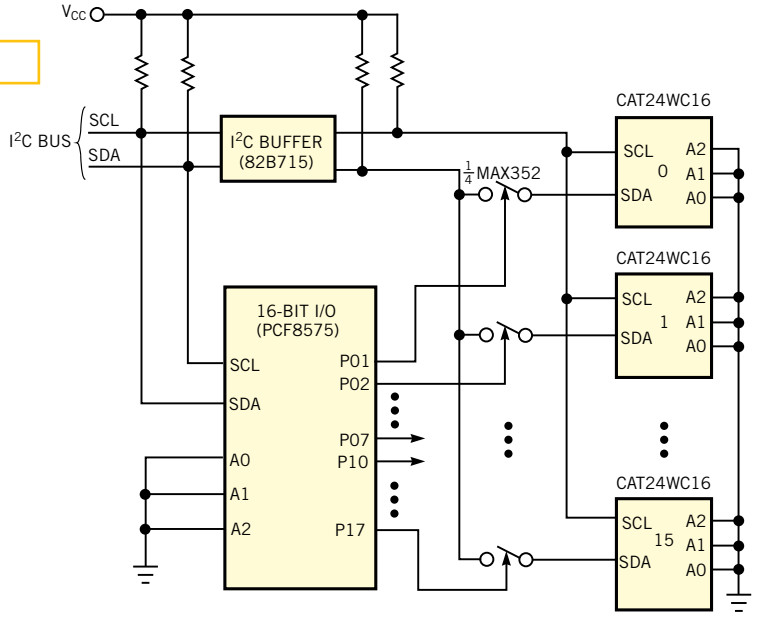
Edited by Bill Travis and Anne Watson Swager

Circuit gang-programs EEPROMs over I²C bus

Denisa Stefan, Catalyst Semiconductor, Sunnyvale, CA

You use the fully controlled circuit in **Figure 1** to parallel-program two-wire serial EEPROMs via the I²C bus. Gang programmers must address all memory devices during a write operation. To verify the memory contents, however, the system must address only one memory at a time during read operations. Therefore, the system in **Figure 1** addresses the memory devices either in parallel or one at a time. Information transfer between devices connected to the I²C bus system requires a SDA (serial-data) and SCL (serial-clock) signals. A device connected to the bus can operate as a transmitter or a receiver. A master device initiates a data transfer on the bus, generates clock signals, and terminates the transfer. The master addresses a slave device. To connect devices on an I²C multimaster bus, the SDA and SCL lines must be bidirectional and must connect to a positive supply voltage through pullup resistors.

Figure 1



An I²C expander and analog switches provide gang programming and serial read access for multiple EEPROMs.

In I²C-bus addressing, the first byte after a Start condition determines the slave that the master selects. A slave address is seven bits long and usually comprises a fixed part and a programmable part. The eighth bit, or LSB, determines the direction of the transfer, either read or write. The programmable part of the slave's address allows you to connect the maximum possible number of identical devices to the I²C bus. This number depends on the number of address-input pins the I²C device has. In **Figure 1**, the serial-data line, SDA, connects to each CAT24WC16 EEPROM via bidirectional Maxim (www.maxim-ic.com) MAX-352 quad SPST analog switches. The switches derive their control from a 16-bit Philips (www.semiconductors.philips.com) PCF8575 I/O expander for the I²C bus. The clock line, SCL, connects to all memory devices. For driving the large capacitive loads required, a Philips 82B715 I²C-bus extender serves as a buffer. The software sequence for parallel writing to all memory devices is to set the port pins by writing to the PCF8575 to command closing all switches and then send an I²C-bus command to write to the CAT24WC16 EEPROMs.

The software flow for reading the contents of one memory device is to set the port pins by writing to the PCF8575 to close the switch associated with the memory to read, set all the other switches to open, and then send an I²C command to read the selected CAT24WC16 EEPROM.

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

LFSR provides encryption

Antonella Di Lillo and Giovanni Motta, Brandeis University, Waltham, MA

LFSRs (linear-feedback shift registers) find extensive use in cryptography. For example, the cryptographic algorithms in the GSM (Global System for Mobile communications) mobile-phone system rely on the use of LFSRs. An LFSR comprises a register containing a sequence of bits and a feedback function. In general, this function is an XOR (exclusive-OR) operation on certain bits in the register. The list of these bits is a "tap sequence." You use an LFSR to generate a pseudorandom sequence of bits that undergo an XOR operation. The XOR result then connects to the input of the LFSR. Repeating the process at the decoder side returns the original sequence of bits. **Listing 1** presents the encryption and decryption process. To generate a pseudorandom sequence, you load the register with a nonzero content, and the software then computes the XOR of the taps and shifts all bits in the register one bit to the left. Finally, the routine inserts the results of the XOR operation in the rightmost position (**Figure 1**). The program adds this 1-bit result to the sequence and repeats the procedure to generate other bits.

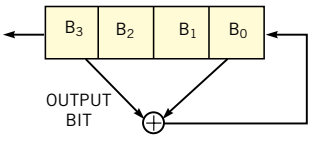


Figure 1

A linear-feedback shift register, combined with an XOR operation, is ideal for encryption.

LFSRs are well-suited to hardware implementations, but their use in software programs unfortunately often suffers from inefficient implementations. LFSRs with few taps, or sparse LFSRs, are easier to use because you need calculate the XOR of only a few bits. On the other hand, for cryptographic purposes, you must avoid sparse polynomials because the resulting algorithms are easy to break. The C program in **Listing 1** has the advantage of implementing the XOR of a 32-bit integer with an efficient algorithm, thus making an efficient way to implement the software of an LFSR. The program encrypts a standard input into a standard output one character at a time. You use an LFSR to generate eight random bits at a time, and the routine XORs the random bits to the current character. The encryption key is a nonzero integer that you use as the initial status of the register. The key is the same for both encoding and decoding. The variable taps represent the 32 binary coefficients of a primitive polynomial of degree 31. Several other choices are possible; **Listing 1** suggests five of them in the comments. You can safely remove the code that describes the number of characters encrypted and the running time to make the program even smaller. You can download the software from the Web version of this article at www.ednmag.com.

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

LISTING 1—LFSR ENCRYPTION AND DECRYPTION

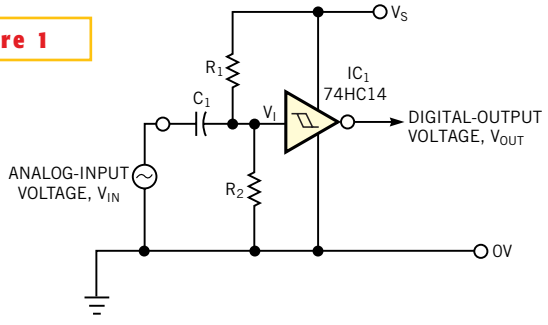
```
//  Usage: LSFR <Key>
//  Key is a non-zero 32 bit integer
//  Alternative taps : 0xBF75CC1F, 0xC679E105,
//    0x844EC703, 0xBE4F7253, 0xDF3B0B11,...
#include <stdio.h>
#include <time.h>

//  XOR of the bits in a 32 bit integer
int xor_32(int a) {
  a ^= a >> 1;
  a ^= a >> 2;
  a ^= a >> 4;
  a ^= a >> 8;
  a ^= a >> 16;
  return(a & 0x1);
}

int main(int argc, char *argv[]) {
  int LSFR, c_char, i, cnt = 0, kcps;
  int taps = 0xE04D11E7;          // Polynomial
  clock_t start;
  double interval;

  if(argc < 2) {                  // Usage
    fprintf(stderr, "Usage:\n");
    fprintf(stderr, "\t%s <Key>\n", argv[0]);
    return -1;
  }
  if(!(LSFR = atoi(argv[1]))) { // LSFR Status
    fprintf(stderr, "Key must be non zero\n");
    return -1;
  }

  start = clock();
  while((c_char = fgetc(stdin)) != EOF) {
    for(i=0; i<8; i++)
      LSFR = (LSFR<<1) | xor_32(LSFR & taps);
    fputc((LSFR & 0xFF) ^ c_char, stdout);
    cnt++;
  }

  interval = (double)(clock()-start) /
                  CLOCKS_PER_SEC;
  kcps = (int)(cnt / interval);
  fprintf(stderr, "Encoded %d chars ", cnt);
  fprintf(stderr, "in %f seconds ", interval);
  fprintf(stderr, "(%d char/sec)\n", kcps);
  return 0;
}
```

Resistor network extends Schmitt trigger's reach

Anthony Smith, Scitech, Biddenham, UK

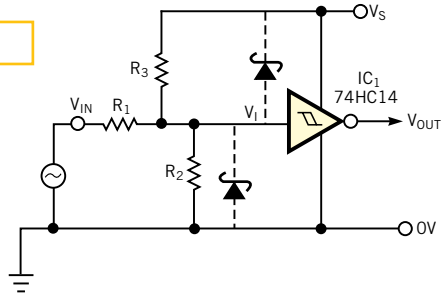
THE CIRCUIT IN **Figure 1** shows a familiar technique for converting a low-level analog signal to digital form. Resistors R_1 and R_2 set the quiescent dc level at the Schmitt inverter's input to a value roughly equal to the midpoint of the hysteresis band. Capacitor C_1 removes dc content from V_{IN}, such that the Schmitt trigger's input signal, V_I, centers itself on the midhysteresis level. Provided that V_{IN} is large enough to cross IC_1's threshold level, the output signal, V_{OUT}, provides a faithful digital representation of V_{IN}. Unfortunately, the circuit suffers from several drawbacks. The presence of C_1 makes it impossible for IC_1 to switch at specifically defined dc levels on V_{IN}. Furthermore, for low-frequency waveforms, C_1 must be extremely large to prevent unwanted signal attenuation. Also, if V_{IN} is of random period or is asymmetrical with time (for example, a pulse train with low duty cycle), the signal at V_I will not swing symmetrically about the quiescent dc level and may fail to cross one of IC_1's thresholds. You can solve all these problems by replacing C_1 with a resistor, as in **Figure 2**.

Figure 1



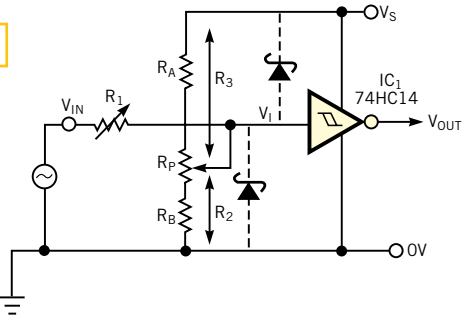
This Schmitt-trigger circuit is useful for converting an ac signal to digital form.

Figure 2



Eliminating the input capacitor avoids problems with asymmetrical input waveforms.

Figure 3



The potentiometer networks solve the problem of large spreads in component values.

In **Figure 2**, R_1 and the parallel combination of R_2 and R_3 act as an attenuator that allows IC_1 to switch at specific, user-defined dc levels that may be much greater than IC_1's switching thresholds. Furthermore, R_2 and R_3 introduce an offset that allows V_{IN}'s lower threshold to be negative if required. R_1 and R_2 relate to R_3 as follows:

$$R_1 = \frac{R_3(V_{TL}V_P - V_{TU}V_N)}{V_S(V_{TU} - V_{TL})},$$

$$R_2 = \frac{R_3(V_{TL}V_P - V_{TU}V_N)}{V_S(V_P - V_N + V_{TL} - V_{TU}) + V_{TU}V_N - V_{TL}V_P},$$

where V_S is the supply voltage; V_P and V_N are the required upper and lower V_{IN} thresholds, respectively; and V_{TU} and V_{TL} are the Schmitt trigger's upper and lower switching thresholds. By measuring V_{TU} and V_{TL} for a given Schmitt inverter and selecting a suitable value for R_3, you can calculate the corresponding values of R_1 and R_2. The circuit accommodates almost any values of V_P and V_N. The only restriction is that the hysteresis ($V_P - V_N$) is sufficiently larger than IC_1's hysteresis ($V_{TU} - V_{TL}$); otherwise, the equations can yield negative resistor values. If IC_1 is a CMOS device (for example, 74HC14, 74AC14, 4093B, or 40106B), you can use large resistances, thus ensuring high input impedance.

For cases in which it is inconvenient to measure the exact values of V_{TU} and V_{TL}, you can replace R_1 and R_2 with variable resistors to accommodate the worst-case spread in V_{TU} and V_{TL}. However, because R_2 and R_3 have a large influence on R_1, the spread of values you need for R_2 results in a broad variation in the R_2-R_3 parallel combination and results in an even broader spread of values for R_1. Replacing R_2 and R_3 with a potentiometer network, as in **Figure 3**, provides a solution to the "spread" problem. Because R_2 varies with R_3, the spread in the R_2-R_3 parallel combination, and hence in R_1, is narrower. This arrangement results in some fairly onerous equations relating the variables. However, you can simplify matters by observing that for a particular CMOS Schmitt inverter, each of its thresholds is a constant fraction of the supply voltage, V_S. Therefore, you can

define $V_{TU}=UV_S$ and $V_{TL}=LV_S$, where U and L are the respective fractions. This simplification results in the following equations:

$$R_1 = R_2 \frac{V_S(L-U)+V_P(1-L)+V_N(U-1)}{V_S(U-L)},$$

$$R_2 = R_X \frac{LV_P-UV_N}{V_S(L-U)+V_P-V_N}, \text{ and}$$

$$R_3 = R_X-R_2.$$

The design procedure is to select the desired values for V_S, V_P, and V_N and then to calculate R_1, R_2, and R_3 in terms of R_X for the worst-case spread in U and L. You can then scale the values of R_1, R_2, and R_3 accordingly. As an example, assume that you need to set V_P at 6V and V_N at −7.5V using a 74HC14 operating from a 5V supply. Although slight differences exist between manufacturers, the "typical" spread in thresholds for the 74HC14 on a 5V rail yields the following values: U=0.5 (minimum) to 0.7 (maximum), and L=0.2 (minimum) to 0.44 (maximum). These values are subject to the restrictions on hysteresis: (U−L)=0.09 (minimum) to 0.5 (maximum). You can intuitively see that R_1 is at a maximum when IC_1's hysteresis is small and the parallel combination of R_2 and R_3 is large. This scenario occurs when IC_1 has a narrow hysteresis band centered roughly on $V_S/2$. In this example, R_1 is a maximum of $7.25R_X$ when L=0.435 and U=0.525. Conversely, R_1 is at a minimum when IC_1's hysteresis is large and the parallel combination of R_2 and R_3 is small. This scenario occurs when L=0.2 and U=0.7, resulting in $R_1=1.067R_X$. The range of potentiometer R_P must allow you to set the quiescent value of V_I anywhere from the minimum midhysteresis band level (occurring when L and U are both minima), to the maximum midhysteresis level (occurring when L and U are both maxima). In this example, the values are $R_2=0.4125R_X$ and $R_3=0.5875R_X$ (when L=0.2 and U=0.5) and $R_2=0.6467R_X$ and $R_3=0.3533R_X$ (when L=0.44 and U=0.7). Assuming that you use resistors with ±1% tolerance and potentiometers with ±10% tolerance, you can accommodate the required spread in R_2 and R_3 with an adequate margin by making $R_A=1.1$ kΩ, $R_P=1$ kΩ, and $R_B=1.3$ kΩ. The corresponding spread in R_1 (including the tolerance in R_X itself) is 3.495 to 25.549 kΩ. You can obtain this range by using a parallel connection of a 50-kΩ potentiometer and a 51-kΩ resistor that is in series with a 3.3-kΩ resistor.

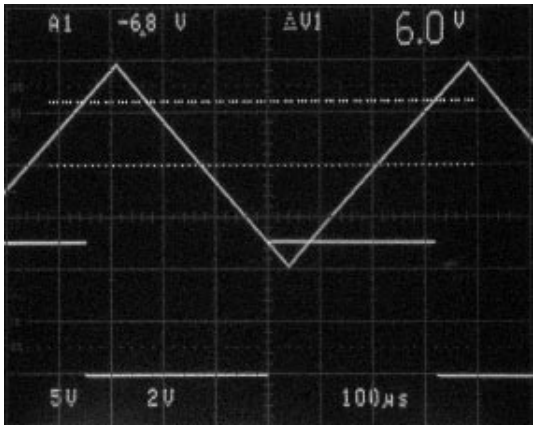


Figure 4 These waveforms indicate clean hysteretic switching with a triangle-wave input.

The oscilloscope screen in **Figure 4** illustrates the performance of the example circuit, in which V_{IN} is a ±10V triangle wave. By adjusting the two potentiometers in turn, we made the output waveform switch when $V_{IN}=6$V and −7.5V. Despite the interaction between the potentiometers, you can fairly easily (with a little patience) set the thresholds. Although the circuit is not intended for precision applications, it does extend the range of the garden-variety Schmitt inverter and allows you to implement positive and negative thresholds of several tens or even hundreds of volts. Moreover, the circuit allows V_N to be positive, provided that V_P is sufficiently greater than V_N to avoid negative resistance values. You can obtain operation of greater than 10-MHz frequency if you use suitable devices for IC_1. The 74AC14 or 74HC14 yield response times of just a few nanoseconds with a rail-to-rail output. For best high-frequency performance, use low resistor values, a shunt trimmer capacitor across R_1 to provide compensation, or both. Finally, use Schottky clamp diodes as in **Figure 3** to protect the inputs of IC_1 from overvoltage conditions.

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

Routine yields fast bit reversing for DSP algorithms

Mohammed Aziz, University of Leeds, UK

If you need efficient real-time performance in DSP applications, you need an efficient bit-reversing routine. For several FFT programs, data permutation can take 10 to 50% of the computation time, depending on the input-data dimensions and length. The idea behind bit reversing is to shuffle the data by flipping the address bits around the mid-

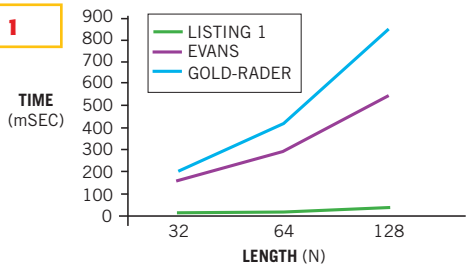
TABLE 1–BIT-REVERSING RUNTIME RESULTS

Length (N)	Listing 1 (msec)	Evans (msec)	Gold-Radar (msec)
32	10	159	200
64	18	294	418
128	34	549	847

dle of the address length so that if the data length is N=16, four bits from 0000 to 1111 represent the address. You achieve data shuffling by swapping the address bits around the middle so that $B_3B_2B_1B_0$ becomes $B_0B_1B_2B_3$, which represents the new data location. Note that this operation is not byte flipping unless the input-data length happens to be N=255 elements. Look-up-table techniques are inefficient, because the input data may be very long, and memory space is limited. For real-time DSP applications, you bit-reverse a data array by swapping each position in the data array with the position of its corresponding bit-reversed address by using DSP architectural features. You implement the method shown in **Listing 1** using the SHARC DSP chip.

The routine uses the data-memory segment and program-memory segments for input and output. This feature of the DSP-memory architecture allows reading from the input array "in-arry" and writing to the bit-reversed output array "out-arry" in parallel to double the speed. It then uses circular buffers and indirect addressing to go through the N elements, lending itself to simple and straightforward data moving. Existing techniques, such as Evans and Gold-Rader algorithms, do data checking and bit reversing before moving the data, thereby incurring overhead. **Table 1** and **Figure 1** give the runtime results for the different algorithms (**Listing 1**, Evans, and Gold-Rader) simulated on a 40-MHz SHARC DSP (ADSP-21060) from Analog Devices (www.analog.com). The runs represent three lengths of 32, 64, and 128 elements and involve 10,000 iterations each in simulating the 2-D case.

Figure 1



The routine in Listing 1 produces markedly faster results than other algorithms.

LISTING 1–BIT-REVERSE ALGORITHM

```
#define  BRmod      //bit-reverse of N/2, N-length of input data
#define  in-arry    //bit-reversed address of input data location
#define  out-arry   //address of output array
#define  N          //length of input data

BR-Algorithm:
bit set mode1 BR0;                    //this allows BR-mode
b0=in-arry;  l0=0;  m0=BRmod;         //this is circular buffers
b8=out-arry;  l8=N;  m8=1;
r0=dm(i0,m0);
lcntr = (N-1), do br until lce;       //this is a loop counter
br: r0=dm(i0,m0), pm(i8,m8)=r0;
pm(i8,m8)=r0;
bit clr mode1 BR0;                    //this clears the BR-mode
```

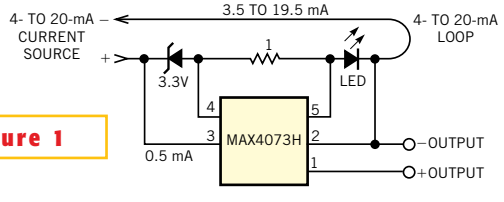
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

A 4- to 20-mA loop needs no external power source

Shyam Tiwari, Sensors Private Ltd, Gwalior, India

THE SIMPLE CIRCUIT in **Figure 1** uses a low-current-drain MAX-4073H amplifier to sense the current flowing through a 4- to 20-mA loop. The circuit senses the current through a 1Ω resistor with a fixed gain of 100 and uses no battery or dc power supply. The low current drain of the amplifier (0.5 mA) enables the circuit to tap its power from the 4- to 20-mA loop to power the amplifier chip. Note that the current flowing in the amplifier's power-supply Pin 3 (nominally 0.5 mA but may vary slightly) is not part of the sensing loop. It forms a negative offset in the measurement and is not a serious problem. To make this current nearly constant, a 3.3V zener diode and an LED in series with the sensing resistor form a voltage drop of 4 to 4.5V across pins 2 and 3 of the amplifier chip. The amplifier works well over 3 to 28V, so this 4 to 4.5V power-supply range presents no problems.

Figure 1



A current-sensing circuit derives its power from the 4- to 20-mA current loop.

The output of the amplifier is linear from 350 to 1950 mV for 4 to 20 mA through the loop. The measurement meter at the output must not draw more than 5 µA from the output for 1% full-scale measurement accuracy. The LED shows visual intensity variation for changing current in the loop. Its main purpose is to raise the voltage by approximately 1V across the sense resistor with respect to the power-supply return Pin 2 of the amplifier. This increased voltage gives better common-mode performance to the amplifier against common-mode noise in the sensing resistor and prevents the amplifier from saturating near the power-supply rails.

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

design ideas

Edited by Bill Travis

Calibrate scope jitter using a transmission-line loop

David Cuthbert, Micron Technology, Boise, ID

Digital-clock-period jitter is the variation in the period of a clock cycle compared with a nominal (average of many cycles) clock period. To accurately measure period jitter using an oscilloscope, you must subtract the oscilloscope jitter from the measured jitter. However, oscilloscopes rarely have a jitter specification, so you must determine the oscilloscope jitter. One method of measuring oscilloscope jitter is to use the oscilloscope to measure the jitter of a pulse generator with known jitter. The measured jitter, assuming the jitter has a Gaussian distribution, is $\sqrt{(\text{scope jitter})^2+(\text{generator jitter})^2}$. Rearranging the formula to solve for oscilloscope jitter, the scope jitter is $\sqrt{(\text{measured jitter})^2-(\text{generator jitter})^2}$. The ideal generator for measuring oscilloscope jitter would have zero jitter. **Figure 1** is a circuit for generating a calibration signal with near-zero timing jitter.

A transmission-line delay loop creates a delayed pulse at the oscilloscope. **Figure 2** shows the circuit in operation. You set the oscilloscope for internal triggering on the first pulse and then configure the scope to measure the time between the first pulse and the delayed pulse. You set the trigger hold-off such that the oscilloscope always triggers on the first pulse. The second pulse in each set of pulses is the first pulse delayed through the transmission-line delay loop. Two 50Ω coaxial transmission lines implement the circuit and connect to the oscilloscope using two BNC T adapters. The line from the 50Ω generator to the 50Ω oscilloscope can be of any length. The length of the delay-loop line determines the delay between the first pulse and the delayed pulse. You set the generator's pulse period to approximately five times the loop delay and the generator's pulse duration to approximately one-half the loop delay.

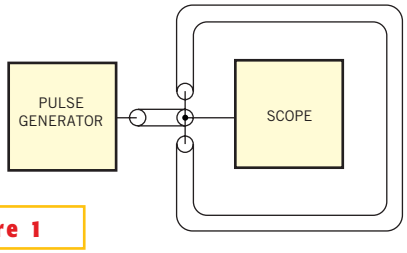


Figure 1

You can use a transmission-line delay loop to accurately measure an oscilloscope's jitter.

The waveforms in **Figure 2** represent a pulse period of 62 nsec and a pulse duration of 6 nsec. The delay loop consists of 2.4m of RG-58 coax and creates a delayed pulse 13 nsec after the first pulse. The 13-nsec delay is equivalent to calibrating the oscilloscope with a zero-jitter, 77-MHz signal. The impedance relationships are such that the delayed pulse has the same amplitude as the first pulse. With the 50Ω pulse generator set for 1V into 50Ω, the generator sends a 1V incident pulse toward the oscilloscope. As the incident pulse "sees," the node at the oscilloscope consists of the 50Ω scope in parallel with the two 50Ω ends of the delay loop. Therefore, the impedance at this node is 16.7Ω, and the reflection coefficient is −0.5. The impedance mismatch causes a pulse amplitude of 0.5V to appear at the oscilloscope and a reflected pulse of −0.5V to travel to the generator, where it dissipates. Two 0.5V pulses travel in opposite directions through the delay loop and meet 13 nsec later at the oscilloscope, forming a 0.5V pulse with a source impedance of 25Ω.

The 50Ω oscilloscope, in parallel with the 50Ω line to the generator, forms a 25Ω load that matches the 25Ω pulse source impedance. The delayed-pulse amplitude at the oscilloscope is 0.5V, and a 0.5V pulse travels to the pulse generator, where it dissipates. You can analyze the circuit's action by keeping track of where the energy goes. The 1V generator sends a 20-mW pulse down the 50Ω line. When this pulse encounters the oscilloscope and delay loop, the energy splits four ways. In this split, 5 mW reflects back to the generator where it dissipates, 5 mW dissipates in the oscilloscope, and 5 mW enters each end of the delay loop. When the two 5-mW pulses exit the delay loop, 5 mW dissipates in the oscilloscope and 5 mW travels to the generator, where it dissipates. You can calibrate several oscilloscopes and one pulse generator with the delay loop. Because the loop has near-zero jitter, the jitter that an os-

cilloscope measures is virtually all oscilloscope jitter. The loop allowed a generator with 19-psec rms jitter to calibrate an oscilloscope having 3.8-psec jitter.

The delay loop has some jitter, created by the conversion of amplitude noise to jitter. Without the loop, generator-amplitude noise causes the leading edge of each pulse to cross the oscilloscope's trip point either early or late. In this way, amplitude noise translates to jitter. The following formula gives jitter versus amplitude noise:

$$\text{JITTER} = \frac{\text{AMPLITUDE NOISE}}{\text{dV/dt}}.$$

An ideal loop would produce a delayed pulse that is identical to the first pulse. Amplitude noise on the delayed pulse would be identical to that on the first pulse; thus, jitter attributable to the noise would cancel out. Because of signal loss in a real loop, the delayed pulse is not identical to the first pulse. Therefore, the amplitude noise of the delayed pulse will be less that that on the first pulse, and a conversion of amplitude noise to jitter will occur. The following formula gives amplitude noise versus jitter in the loop:

$$\text{LOOP JITTER} = \frac{\text{AMPLITUDE NOISE}}{\text{dV/dt}} \times (\text{SIGNAL LOSS}).$$

The pulse generator used to calibrate the oscilloscopes exhibits 250 µV of rms noise and a dV/dt of 0.35V/nsec. The signal loss on the leading edge of the delayed pulse is 0.2V. The amplitude-noise-to-jitter conversion is thus:

$$\text{LOOP JITTER} = \frac{250\,\mu\text{V}}{0.35\text{V/nSEC}} \times (0.20) = 143\,\text{fSEC}.$$

The loop jitter of 145 fsec is so far below the jitter noise floor of the oscilloscopes under calibration that you can consider the delay loop as a zero-jitter source. Because digital-oscilloscope jitter is a function of the timebase setting and the amount of ADC dynamic range you use, you should calibrate the scope with a loop delay and amplitude that match the signal to the actual system or device under test.

Figure 2

The transmission-line delay loop creates a delayed signal with near-zero jitter for calibrating the oscilloscope.

REFERENCE

1. Adler, Joe, "Jitter in clock sources," Application Note, Vectron International.

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

Excel offers painless LCD initialization

Alberto Bitti, Eptar, Lugo, Italy

To display a font or a symbol on an LCD, you need to convert the desired character into numerical data. Creating the data for an entire font set requires specialized tools; even with these tools, the task can be daunting. Alternatively, you can build a font calculator using an Excel spreadsheet. This technique takes advantage of the tabular nature of a spreadsheet to automatically create the required initialization code. The example in **Figure 1** applies to displays arranged in blocks comprised of eight pixels by eight pixels. You can easily adapt it to any other arrangement, such as the seven-by-five-pixel format used in the eight customizable characters found in most alphanumeric displays. Cells inside the yellow area represent the symbol "pixels." You draw the desired font character or symbol using a bold character such as "#." The formula in **Figure 2** is all that

font calculator.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1											{					
2				#	#	#	#				60	,				
3			#	#			#	#			102	,				
4							#	#			6	,				
5						#	#				12	,				
6					#	#					24	,				
7				#	#						48	,				
8			#	#	#	#	#	#			126	,				
9											0					
10											}					
11																

Figure 1 You edit the graphical shape (yellow). A simple formula builds the initialization code (blue), which is ready to paste in your application.

you need to obtain the initialization data. For each of the eight pixels in a row, the formula tests whether the pixel is blank (LEN is zero); otherwise, it adds the pixel's "weight" (1, 2, 4, 8, 16, 32, 64, or 128) to the result.

You can prepare the spreadsheet in minutes. Type the formula in the first position (K2), then copy it to all eight rows in a symbol (the blue area from K3 to K9). Complete it with the separators for your language of choice (commas and parentheses for C). To build an entire character set, copy the whole block as many times as necessary. To edit the fonts, place the cursor over the cells and type "#" (or any other character), and use "Canc" to delete. After editing, select all of the columns K and L; then cut and paste the code into your application. Besides saving you money, this technique is convenient and flexible. You can adapt it in minutes to any language (useful when you switch between assembly dialects). Moreover, the method accommodates useful additions, such as inserting #define KEY_ICON to name a particular data set, to suit your application's requirements.

Figure 2

```
= IF( LEN(B2);128; 0 ) +
  IF( LEN(C2); 64; 0 ) +
  IF( LEN(D2); 32; 0 ) +
  IF( LEN(E2); 16; 0 ) +
  IF( LEN(F2);  8; 0 ) +
  IF( LEN(G2);  4; 0 ) +
  IF( LEN(H2);  2; 0 ) +
  IF( LEN(I2);  1; 0 )
```

The formula (which, in this example, shows cell K2) adds pixel "weights" to obtain initialization values.

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

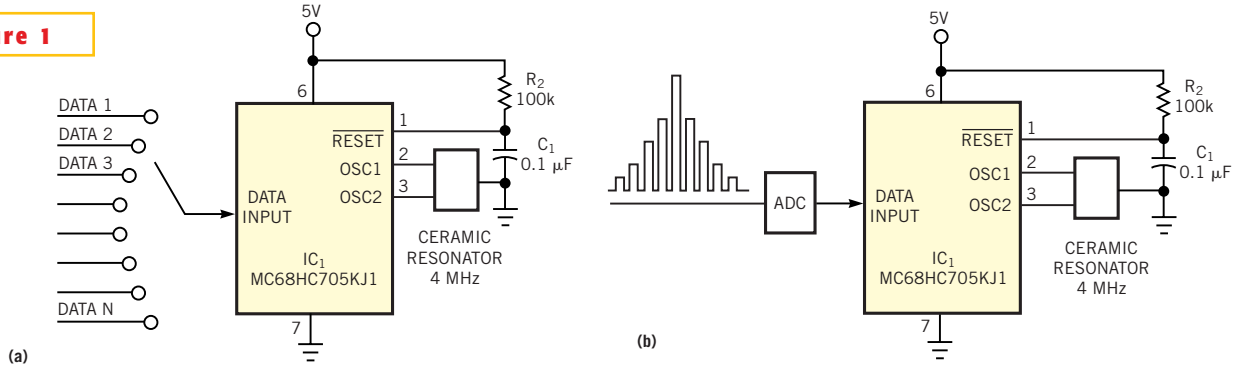
Microcontroller selects minimum/maximum value

Abel Raynus, Armatron International, Melrose, MA

Microcontroller-based systems for measurement, sensor-data processing, or control, sometimes require you to determine a maximum or minimum data value. For example, in an object-detection system, such as a radar or sonar system, the microcontroller receives echo signals from multiple targets and then must select the closest one; in other words, it must determine the minimum distance. Another example is an automatic-tuning system in which the microcontroller acquires data and must determine the maximum or minimum values of the data. A simple way to determine maximum and minimum value involves a microcontroller, IC_1, that receives a set of data from N sources (**Figure 1**). The data should be in 8-bit format. To simplify the process, consider that the data are 1-byte-long integers. In other words, the data cover the range 0 to 255. In most cases, this range produces satisfactory results. The same limitation applies to the number of data sources. If you need more data precision or more data sources, then you can use two or more bytes at the expense of added program complication.

Two approaches exist for the maximum/minimum values. In the first, the microcontroller memory collects the received data and processes the data to de-

Figure 1



In one approach to determining minimum and maximum values, the microcontroller stores data values in memory before processing them (a); in another approach, it processes data on the fly (b).

termine maximum or minimum values (**Figure 1a**). In the second approach, the microcontroller processes the data immediately after receiving it (**Figure 1b**). **Listing 1** shows the program for the first approach. Assume that a data array with N=8 exists in the microcontroller's memory. For demonstration purposes, **Listing 1** illustrates this part of the routine in lines six to eight and 17 to 21, in which the controller loads the data registers from a predetermined table. In a real situation, you would load the data registers from data sources before calling the program. The process of minimum-value detection is based on a method called "the bubble." The search algorithm starts testing the memory data array from its end. It clears the index register (X) after completion of the program. If you need to detect the maximum instead of the minimum value, you need change only one instruction in line 28.

In the second approach, the microcontroller waits for a data value entering the data register and, upon reception, starts processing the value (**Listing 2**). This approach needs no memory array. This variant of the program adds two features: First, it simultaneously selects minimum and maximum values of the incoming data and saves them in the DATAmin and DATAmax registers. Second, the routine considers a data value equal to 0 as no data. Thus, the data range is from 01_h to ff_h. For this project, you can use the inexpensive, one-time-programmable MC68HC705KJ1 from Motorola (www.motorola.com). Thus, the programs use Motorola's assembly language. However, the algorithms are so straightforward, they're amenable to any language and to practically any microcontroller. Go to the *EDN* Web site www.ednmag.com for an electronic version of the listings.

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

LISTING 1—MEMORY-BASED MINIMUM/MAXIMUM DETERMINATION

```
                                1  ********* MIN VALUE SELECTION VAR 1 **************
                                2  *nolist
0000                            3  $include "std-j1a.asm"
                                4  *list
                                5  ****************************************
0000                            6  tabl equ ROMend-$07 ;for demo only !
07C8                            7    org tabl        ;table of Data
07C8        03040502            8    fcb 3,4,5,2,8,14T,9,25T
            080E0919
                                9  ****************************************
00C0                           10     org RAM
00C0                           11  DATA       rmb 8
00C8                           12  DATAmin    rmb 1
0300                           13     org ROM
0300 [05] 3FC8                 14  start clr DATAmin
0302 [03] 5F                   15     clrx
                               16  ****************************************
0303 [05] D607C8               17  m0    lda tabl,x ;put the set of data from
0306 [05] E7C0                 18    sta DATA,x ;table to registers. This
0308 [03] 5C                   19    incx       ;part of program only for
0309 [02] A308                 20    cpx #8     ;the idea demonstration.
030B [03] 26F6                 21    bne m0
                               22  ****************************************
030D [02] AE07                 23       ldx #7     ;N-1 into X-register
030F [04] E6C0                 24    lda DATA,x ;data8 into Acc.
0311 [04] B7C8                 25    sta DATAmin;store data8 in DATAmin
0313 [03] 5A                   26  mn0  decx       ;go to the next data,x
0314 [04] E1C0                 27    cmp DATA,x ;if DATAmin < DATA,x
0316 [03] 2504                 28    blo mn1    ;  then go to mn1
0318 [04] E6C0                 29    lda DATA,x ;otherwise replace DATAmin
031A [04] B7C8                 30    sta DATAmin;  with new data value
031C [03] 5D                   31  mn1  tstx ;if there are more data to be tested,
031D [03] 26F4                 32    bne mn0    ;  then repeat from mn0
031F [02] 8E                   33    stop       ;end of the program
                               34  ****************************************
07FE                           35    org VECTORS+6
07FE        0300               36    fdb start
```

LISTING 2—ON-THE-FLY MINIMUM/MAXIMUM DETERMINATION

```
                                1  ********* MIN/MAX VALUE SELECTION VAR 2
**************
                                2  *nolist
 0000                           3  $include "std-j1a.asm"
                                4  *list
                                5  ****************************************
 00C0                           6     org RAM
 00C0                           7  DATA      rmb 1
 00C1                           8  DATAmin   rmb 1
 00C2                           9  DATAmax   rmb 1
 0300                          10     org ROM
 0300 [02] A6FF                11  init lda #$ff
 0302 [04] B7C1                12    sta DATAmin
 0304 [05] 3FC2                13    clr DATAmax
 0306 [05] 3FC0                14    clr DATA
 0308 [03] 5F                  15    clrx
                               16  ****************************************
 0309 [03] B6C0                17  start  lda DATA ;wait for DATA / 0
 030B [03] 27FC                18    beq start
 030D [03] B1C1                19    cmp DATAmin ; Data > DATAmin?
 030F [03] 2402                20    bhs m0
 0311 [04] B7C1                21    sta DATAmin
 0313 [03] B1C2                22  m0  cmp DATAmax ; Data < DATAmax?
 0315 [03] 23F2                23    bls start
 0317 [04] B7C2                24    sta DATAmax
 0319 [03] 20EE                25    bra start
                               26  ****************************************
 07FE                          27    org VECTORS+6
 07FE        0300              28    fdb init
```

Circuit forms industrial-grade digital potentiometer

Phill Leyva, Maxim Integrated Products, Sunnyvale, CA

Both AC and DC motors in modern industrial systems often receive their control from PLCs (programmable-logic controllers) in a control room safely away from the process. If an operator must manually set the motor speed while observing the process, the component of choice is usually an industrial-grade potentiometer. The wiper of this potentiometer produces a signal of 0 to 10V that feeds back to a motor controller in the control room. Such potentiometers are expensive and prone to wear, however. Because of wear, they can open the control loop, allowing the motor to ramp up uncontrollably. With a few components, you can implement a reliable, low-cost, surface-mount digital potentiometer for industrial applications (**Figure 1**). The result is a direct drop-in replacement for the wear-prone mechanical potentiometer. The digital potentiometer occupies the same space as a mechanical unit within an enclosure. It takes power from the 10V that is formerly supplied to the mechanical potentiometer from the motor controller. The solid-state unit provides a similar output of 0 to 10V and delivers as much as 15 mA to the controller.

The key to the circuit is the low-power, digital-potentiometer 100-kΩ IC_3. Configured as a voltage divider, this IC provides an output of 32 discrete voltage steps between its minimum and maximum settings (0 and 5V). A low-power linear regulator, IC_1, provides a 5V supply rail for IC_2, IC_3, and a resistor ladder internal to IC_3. PB_1 and PB_2 constitute a double-pushbutton industrial switch. Each high-to-low transition that PB_1 produces increments the digital potentiometer's "wiper" by one step. Depressing PB_2 while toggling PB_1 decrements the wiper by one step. IC_2 is a switch debouncer that provides (in addition to the debouncing) a 40-msec fixed delay between its outputs and the switch action. To provide 0 to 10V outputs as required by the motor controller, a single-supply, rail-to-rail op amp, IC_4, amplifies IC_3's output by a factor of two. The input common-mode range for this op amp—250 mV beyond either supply rail—allows it to generate 0 to 10V outputs like a mechanical potentiometer. The circuit's low quiescent current ranges from 86 µA for a 0V output to 186 µA for a 10V output. To even further lower this quiescent current, you can choose the 200-kΩ version of IC_3. Because the op amp is stable with any capacitive load, it easily drives long, shielded, multiconductor cable lines back to the control room.

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

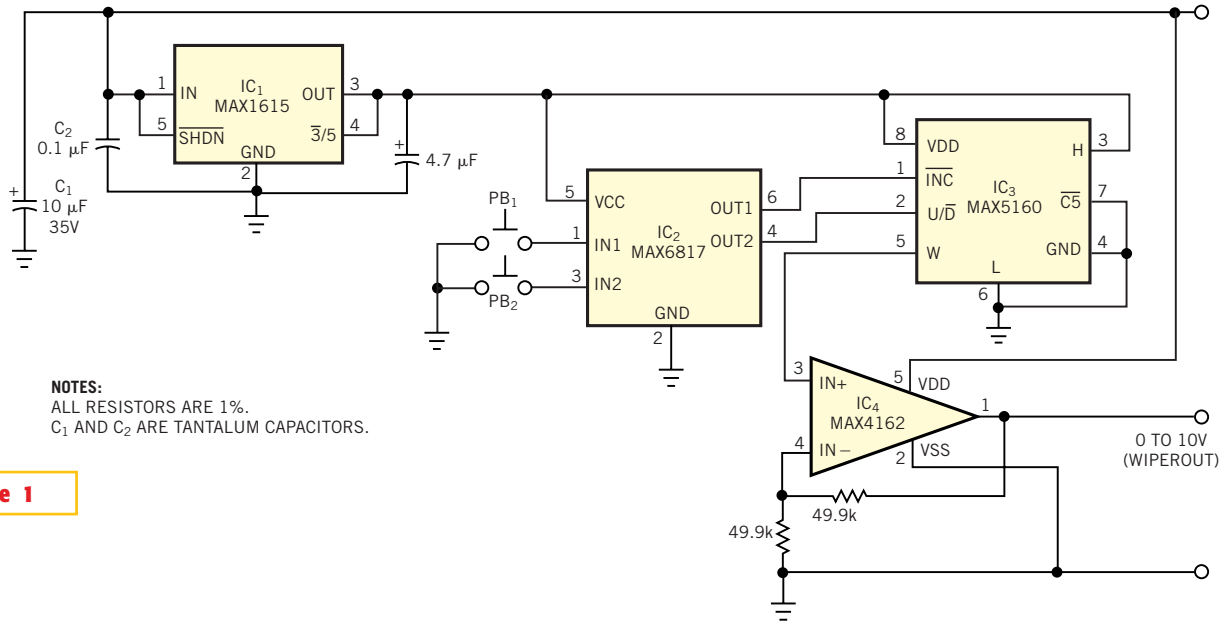


Figure 1

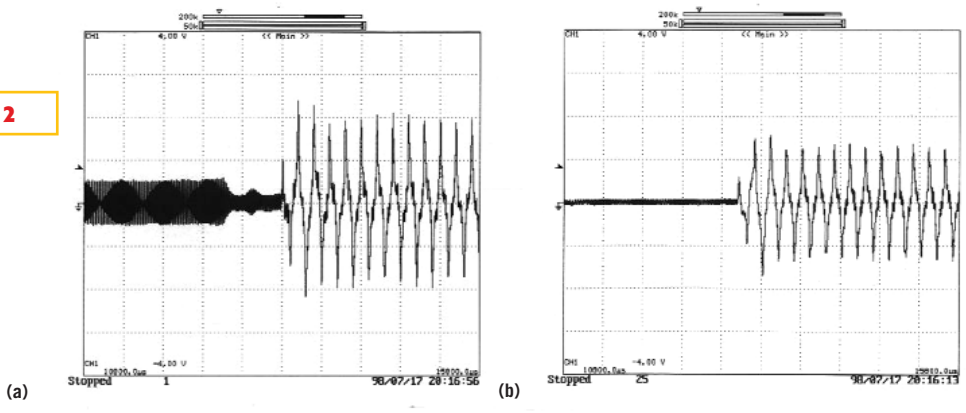
This solid-state potentiometer simulates a mechanical potentiometer and fits in the same space.

Passive filter cleans up power-line communications

Jose Sebastia and JJ Perez, Polytechnic University of Valencia, Spain

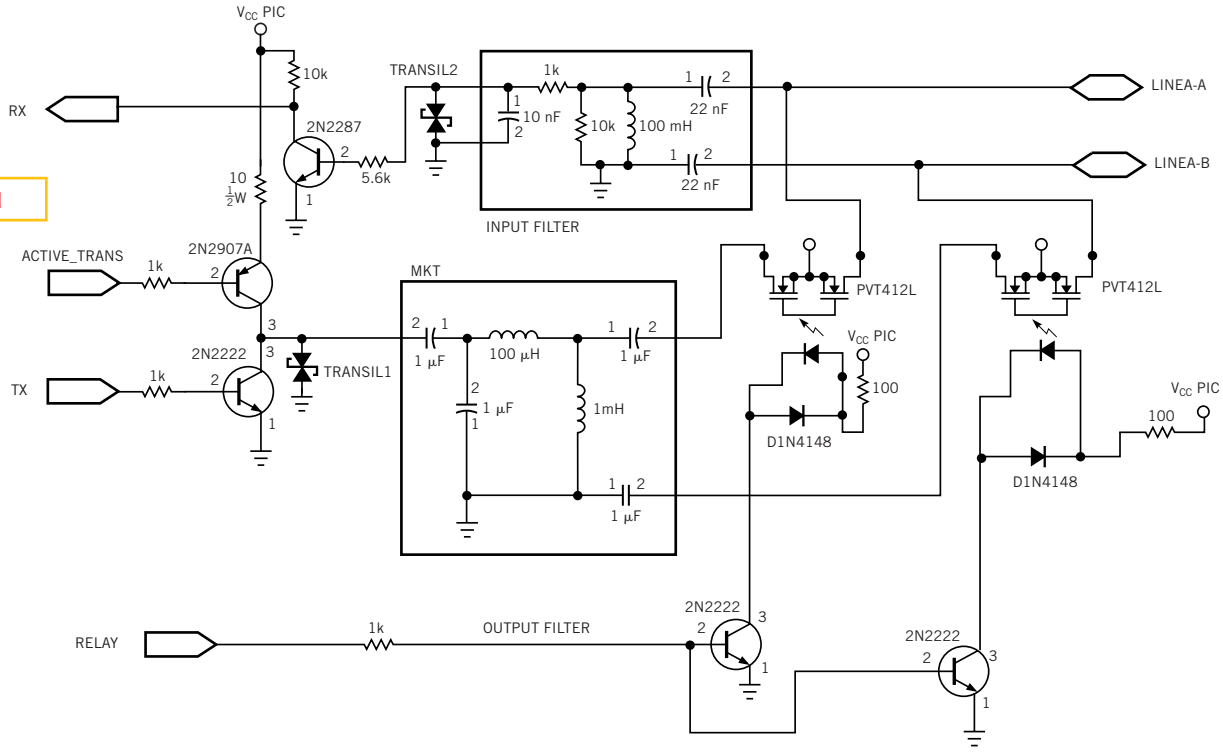
Many applications require the design of custom analog filters (**Reference 1**). The application for this design required simple and low-cost I/O filters for PLC (power-line communications), where low power consumption is a crucial factor. **Figure 1** shows the filters, which use passive components because of the requirement for low power consumption. The PLC system needs an in-

Figure 2



Without an input filter, a great deal of hash accompanies the input signal (a); the addition of the filter considerably cleans up the signal (b).

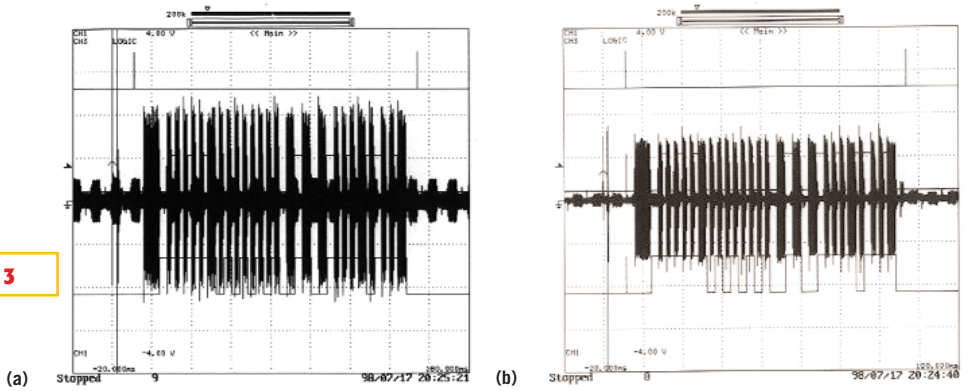
Figure 1



A power-line-communications system needs input and output filters to eliminate interference.

put and an output filter. They are 100-Hz to 20-kHz passband filters; the communication frequency is 5 kHz. The difference between the two filters lies in the input impedance. The input filter must present a 2.2-kΩ impedance, and the output filter must have a 30Ω impedance. The circuit also needs a solid-state relay, the PVT412 from International Rectifier (www.irf.com) to isolate the output filter. When the circuit is active, the relay connects the output filter to the line. A microcontroller controls the relay to implement the signal-transmission and -reception protocols. ACTIVE_TRANS, RELAY, and TX are the microcontroller pins that control transmission, and RX is the pin that controls reception. **Figure 2** shows waveforms before (**Figure 2a**) and after (**Figure 2b**) insertion of the input filter. **Figure 3** shows waveforms before (**Figure 3a**) and after (**Figure 3b**) insertion of the output filter.

Figure 3



Interference and noise are evident in the output signal (a); the addition of the output filter (b) markedly reduces the noise.

REFERENCE

1. Lacanette, Kerry, "A basic introduction to filters: active, passive, and switched-capacitor," Application Note AN-779, National Semiconductor, 1991.

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

design ideas

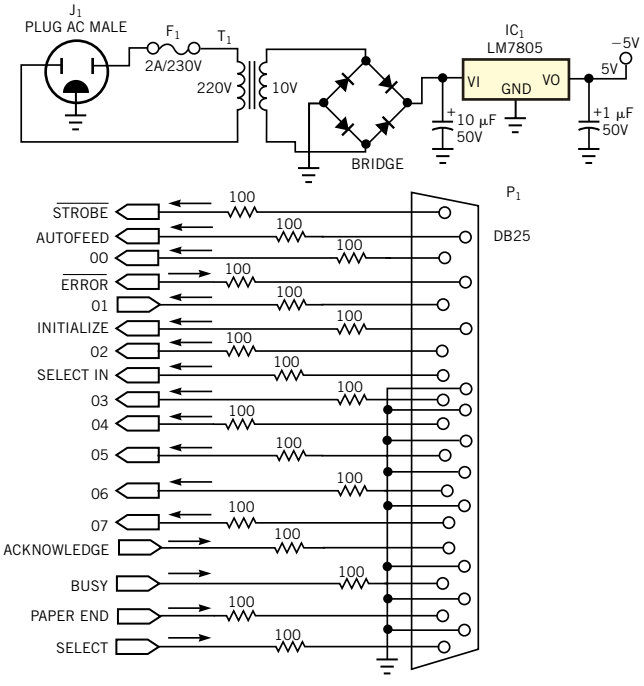
Edited by Bill Travis

Tricks increase utility of parallel port

Lorenzo Tazzari, Selex SNC, Alessandria, Italy

In this simple application of the 68HC68 microcontroller's serial-I/O utility, the goal is to configure a simple circuit, driven by any LPT parallel-printer port, which you can use as a remote I/O for a PC. You can independently program each I/O line as either an input or an output. The protocol in this application is an SPI (MISO/MOSI/SCK) type, using synchronous serial communications. **Figure 1** shows a circuit that effects the connection with the PC and power supply for all I/O signals. A bus carries signals of the SPI protocol, and the LPT port can drive all the $\overline{CE}$ (Chip Enable) signals. With this type of bus, you drive as many as five $\overline{CE}$ signals, and each $\overline{CE}$ line can address four 68HC68 chips. Each microcontroller can drive eight I/O lines, with each line independently programmable. Thus, the system can address as many as 160 I/O ports.

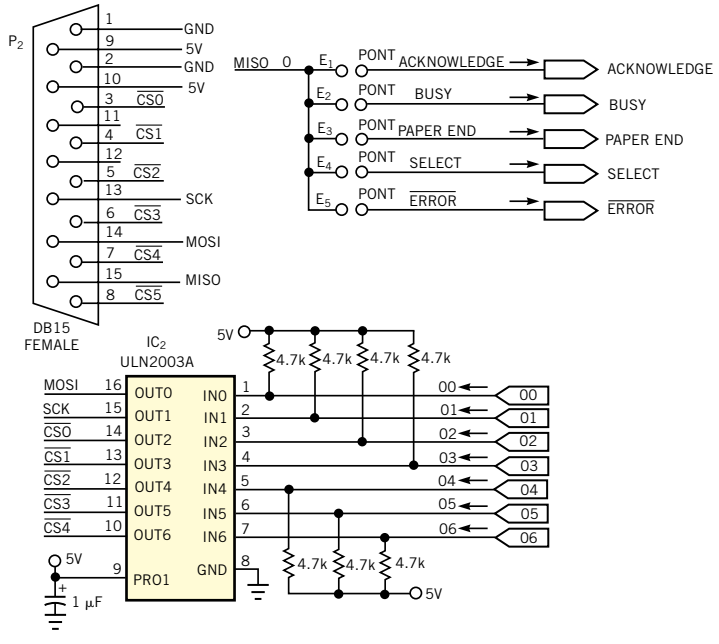
Figure 1



You address as many as 160 I/O signals with this simple connection of the LPT port.

Figure 2 shows a simple way to select the MISO signal. You can choose which pin of the LPT port to use in receiving the MISO signal. This circuit uses the ULN2003 as an amplifier to drive the critical $\overline{CE}$ signal. By joining the SIP and $\overline{CE}$ signals, you can configure a bus system in which all signals go to the bus connector. You need an external power supply, because the LPT port cannot supply sufficient current. You can connect as many as five I/O circuits. The circuit in **Figure 3** contains the 68HC68 chip with its $\overline{CE}$ and address selection. With jumpers E_{12} and E_{13}, you

Figure 2



This simple scheme facilitates selection of the MISO signal in the SPI protocol.

can select the chip's address. It is important to emphasize that the system has no check for an address conflict arising from choosing the same address, with the same $\overline{CE}$, in more than one chip. If the conflict arises, however, it does not cause hardware damage. Theoretically, using an LPT port at a baud rate of 1 Mbyte/sec, the technique can read or write 160 I/O signals in less than 16 msec.

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

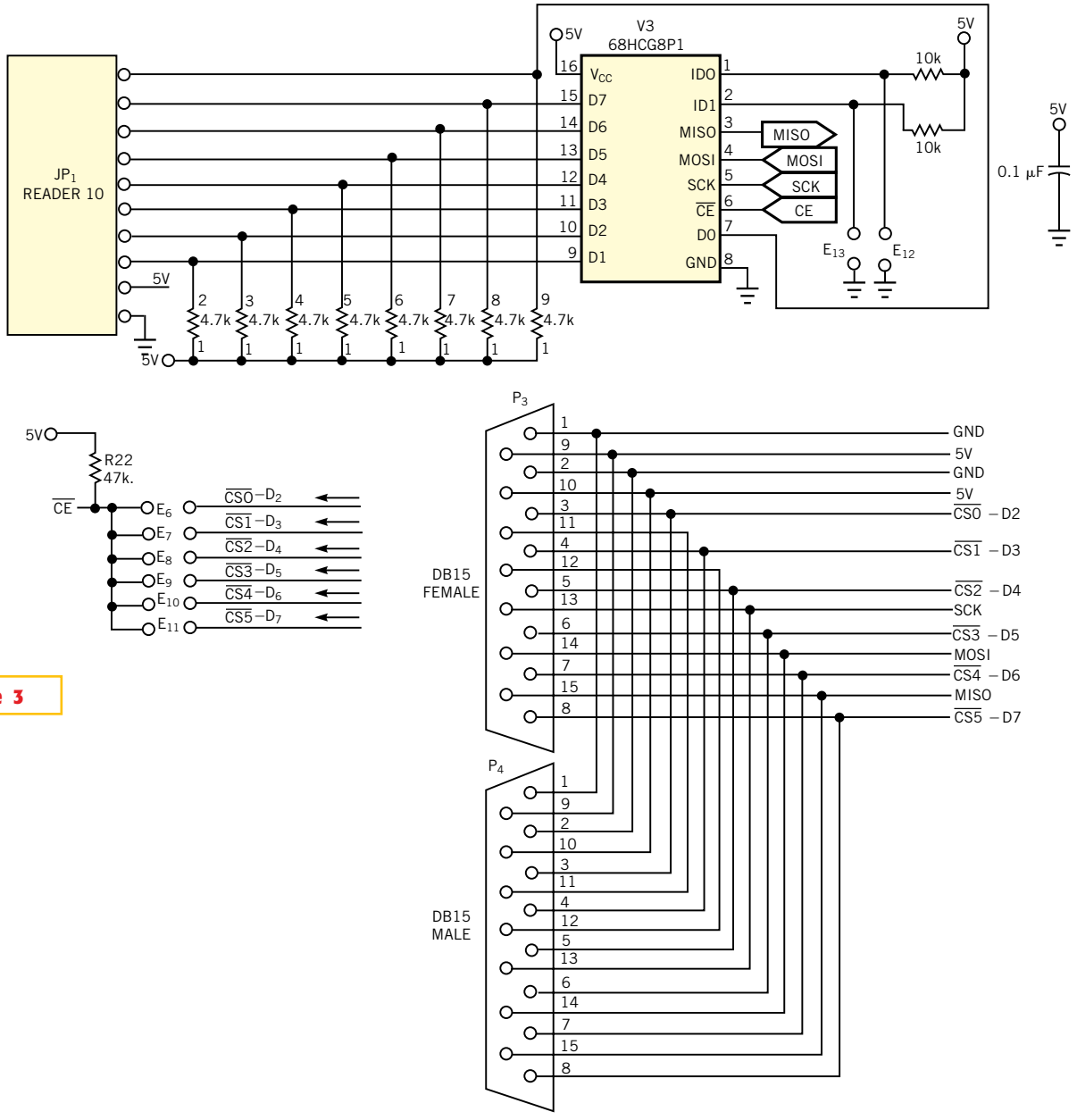


Figure 3

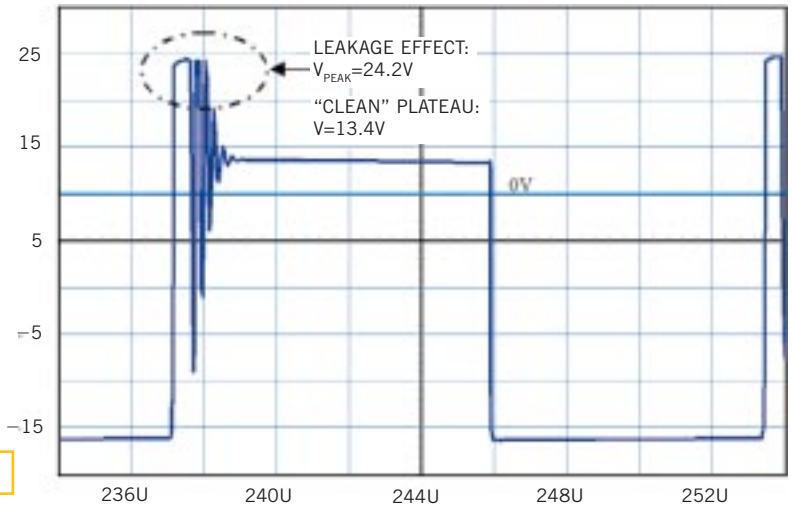
A 68HC68 microcontroller handles $\overline{CE}$ and address selection.

Transistors tame perfidious leakage inductance

Christophe Basso, On Semiconductor, Toulouse, France

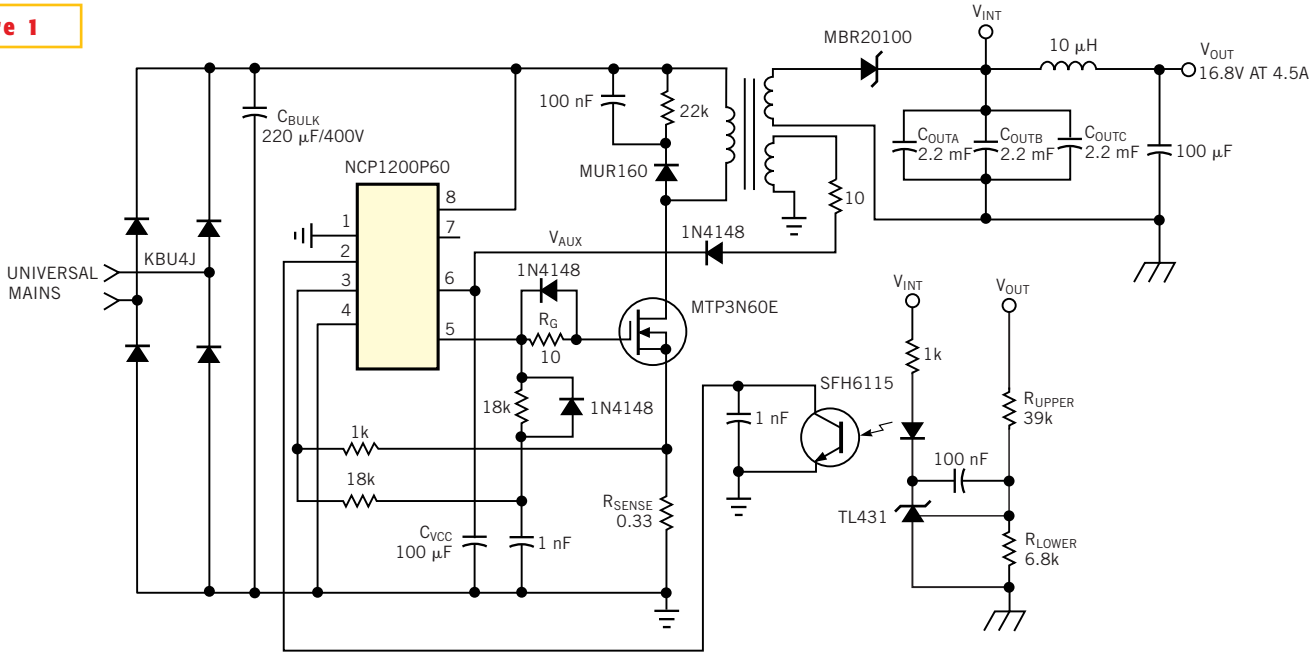
In flyback converters that use primary regulation, the loose coupling between the power secondary and the primary auxiliary windings often results in poor cross-regulation. This situation arises mainly from the leakage inductance but also comes from the level of the primary clamp voltage. **Figure 1** shows a typical application schematic using On Semiconductor's (www.onsemi.com) NCP1200 in an auxiliary-winding configuration. This IC uses a DSS (dynamic self-supply), but in some low-standby-power applications, it is desirable to permanently disconnect this feature through an auxiliary level. The DSS simply acts as a standard start-up current source until the auxiliary level takes over. In this application, the regulation takes place on the secondary side by means of the TL431, but the primary

Figure 2



The leakage inductance on the auxiliary-winding side causes high rectified voltages.

Figure 1



In this circuit, leakage inductance in the auxiliary winding can invalidate the controller's short-circuit-protection circuitry.

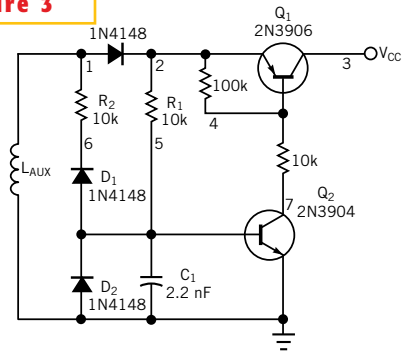
level assumes importance in short-circuit conditions. Each time the NCP1200's V_{CC} crosses 10V while dropping, the internal logic senses the eventual presence of a short circuit through the feedback pin. Should the circuit confirm a short circuit, the NCP1200 emits a safe-autorecovery, low-frequency burst. However, if poor coupling prevents the auxiliary winding from collapsing, in the presence of a secondary short circuit, V_{CC} never crosses the 10V threshold, and damage to the circuit may ensue.

Figure 2 details the effects of the leakage inductance when a short circuit occurs at the output. As you can see, the leakage spike pushes the auxiliary level well above its regular plateau voltage, which is the value you'd like to obtain. With the rectifying diode playing the role of an envelope detector, the result is a final level close to 24V, far from the 13.4V you would expect. As a result, a possibly destructive condition exists if the levels exceed the maximum ratings in the controller's data sheet. You need to clamp the auxiliary voltage using a dissipative element, such as a zener diode. **Figure 3** shows the circuitry you adopt to avoid the leakage-inductance problems. The component arrangement actually implements a self-contained sample-and-hold system. When the main power switch is on, capacitor C_1 discharges through R_2 and D_1, and D_2 avoids a large reverse bias of Q_2's base-emitter junction. When the main switch opens, the secondary voltage sharply rises, and Node 1 becomes positive. However, because C_1 discharges, Q_1 remains open, and V_{CC} does not increase.

After a short period (adjustable via R_1 or C_1), Q_2 closes and brings Q_1's base closer to ground. V_{CC} now increases and catches up to the level at Node 2, minus Q_1's $V_{CE(SAT)}$. If you correctly select the time delay, V_{CC} is devoid of any voltage spike, because you have sampled the plateau. **Figure 4** shows the final result. Performing some measurements on a 70W application board featuring low standby power yields the final tracking results in **Figure 5**. You can see that a 4.3A change in I_{OUT} results in a change of only 420 mV in V_{OUT}. You can use the circuit in a primary-regulation application in which you need a precise level without either heavily filtering the secondary winding (and thus lowering the available auxiliary energy in standby mode) or reducing the primary clamp voltage to a higher dissipative value. In the NCP1200 application, when a short circuit appears at the output, the auxiliary winding properly triggers the short-circuit protection.

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

Figure 3

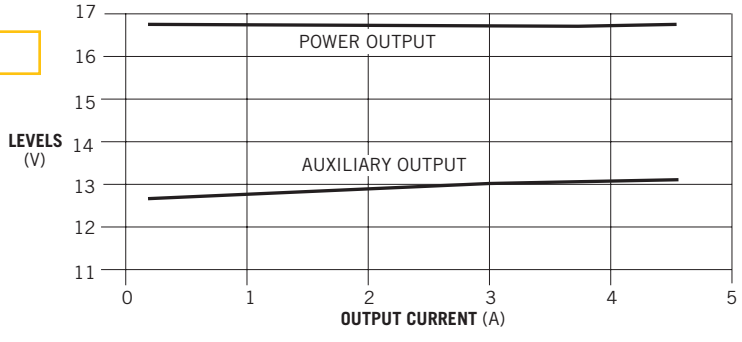


This component arrangement creates a discrete sample-and-hold system.

Figure 4

By delaying the sampling time, you obtain a clean auxiliary level that is devoid of any leakage-inductance effects.

Figure 5



Thanks to the circuit in Figure 3, the auxiliary winding better tracks the primary winding.

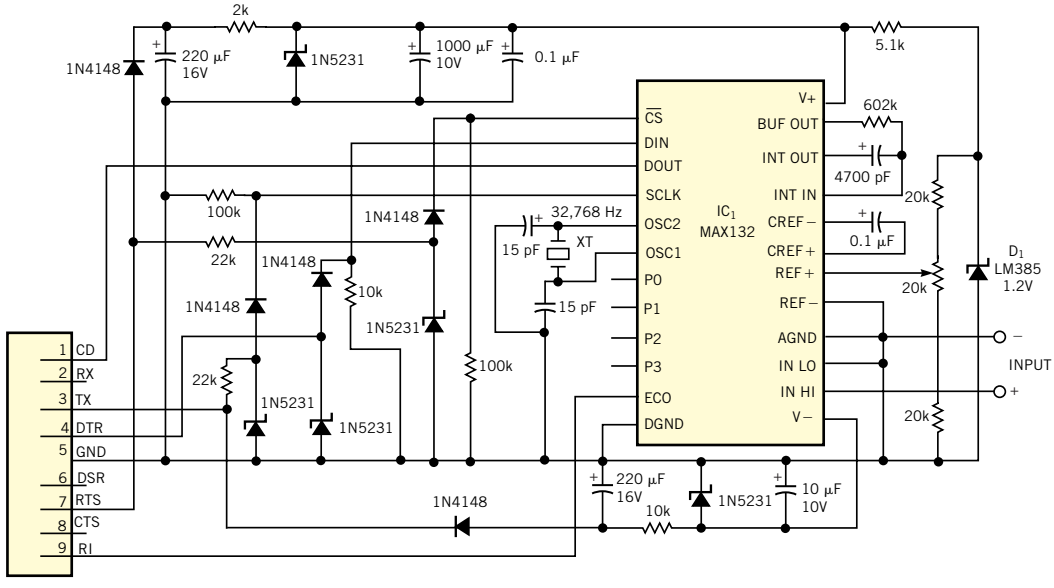
18-bit ADC uses PC's serial port

Yongping Xia, Teldata, Los Angeles, CA

A PC USUALLY REQUIRES a plug-in ADC card to process analog signals. However, with the circuitry in **Figure 1**, a PC can communicate with an 18-bit ADC through its serial port. The port provides both positive and negative power supplies as well as control signals. IC_1 is an 18-bit MAX132 ADC with a serial interface. It requires three input control signals, $\overline{CS}$, DIN, and SCLK, and emits serial data, DOUT, and EOC (end-of-conversion) signals. An RS-232 port has three output lines: Pin 3 (TX), Pin 4 (DTR), and Pin 7 (RTS). TX generates the clock signal for the MAX132 and provides the negative power supply. DTR transmits serial data. RTS provides the CS signal and the positive power supply. Both the positive and the negative supplies use large capacitors for energy storage. When TX generates a clock signal or DTR sends a CS logic-low signal, the capacitors provide power to the MAX-132. The MAX132 integrates everything except a reference that comes from a 1.2V LM385 voltage-reference diode, D_1. The input-voltage range of the MAX132 is −512 to +512 mV. **Listing 1** is a C program that displays the analog-to-digital-conversion result on-screen. You can download **Listing 1** from the Web version of this Design Idea at www.ednmag.com.

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

Figure 1



You can use a PC's serial port to communicate with an 18-bit A/D converter.

LISTING 1—SCREEN-DISPLAY ROUTINE FOR ANALOG-TO-DIGITAL-CONVERSION RESULTS

```
#include <stdio.h>
#include <dos.h>
#include <time.h>
#include <conio.h>
#include <bios.h>
#include <conio.h>

#define COM1    0
#define MCR     4                       /* control register    */
#define MSR     6                       /* status register     */

int i, j, base_add1=0x3f8, base_add2=0x2f8, out_data=0x03, in_data[4];

float data;

void send_clk(void)
{
  delay(1);
  outportb(base_add1, 0x00);
  delay(3);
}

void read_port(void)
{
int control[4], out_control;
data=0;
for (i=0; i<4; i++)
  in_data[i]=0;
control[0]=0x82;
control[1]=0x04;
control[2]=0x00;
control[3]=0x00;
out_data|=0x02;
outportb(base_add1+MCR, out_data);          /* CS high
      */
delay(10);
out_data&=0x01;
outportb(base_add1+MCR, out_data);          /* CS low
      */
delay(10);
out_control=control[0];
for (i=0; i<8; i++)
 {
   if (out_control>=0x80)
     out_data|=0x01;
   else
```

Continued on pg 86

LISTING 1—SCREEN-DISPLAY ROUTINE FOR ANALOG-TO-DIGITAL-CONVERSION (CONTINUED)

```
      out_data&=0x02;
    outportb(base_add1+MCR, out_data);
    send_clk();                                  /* clock out      */
    out_control&=0x7f;
    out_control=out_control*2;
  }
 out_data|=0x02;
 outportb(base_add1+MCR, out_data);                    /* CS high
        */
 delay(10);
 do{
  }while((inportb(base_add1+MSR)&0x40)==0);      /* waiting for EOC=high
        */
 for (j=1; j<4; j++)
 {
    out_control=control[j];
    in_data[j]=0;
    out_data&=0x01;
    outportb(base_add1+MCR, out_data);   /* CS low         */
    delay(10);
    for (i=0; i<8; i++)
      {
        if (out_control>=0x80)
          out_data|=0x01;
        else
          out_data&=0x02;
        outportb(base_add1+MCR, out_data);
        in_data[j]=in_data[j]*2+(inportb(base_add1+MSR)&0x80)/0x80;
        send_clk();                              /* clock out      */
        out_control&=0x7f;
        out_control=out_control*2;
     }
    out_data|=0x02;
    outportb(base_add1+MCR, out_data);           /* CE high        */
    delay(10);
 }
 if ((in_data[1]&0x08)==0)

data=(float)(in_data[1]&0x07)+(float)(in_data[2])*2048+(float)(in_data[3])*8;
  else                                           /* reading is negative      */
    {
      in_data[1]=in_data[1]&0x07;
      in_data[1]=(8-in_data[1])&0x07;
      in_data[2]=(256-in_data[2])&0xff;
      in_data[3]=(256-in_data[3])&0xff;
     data=-((float)(in_data[1])+(float)(in_data[2])*2048+(float)(in_data[3])*8);
    }
}

void dis_data(void)
{
  float show_data;
  show_data=0.000002*data;
  gotoxy(1,1);
  printf("%.5f  ", show_data);
  gotoxy(1,1);
}

void init(void)
{
   bioscom(0, 255, COM1);                        /* set up COM1
          */
   out_data=0x02;
   outportb(base_add1+MCR, out_data);              /* CS=high, DIN=low
          */
   delay(100);
}

 void main(void)
{
   clrscr();
   init();
   gotoxy(60,24);
   printf("Hit any key to quit");
   do{
          read_port();
          dis_data();
          delay(500);
     } while(!kbhit());
}
```

design ideas

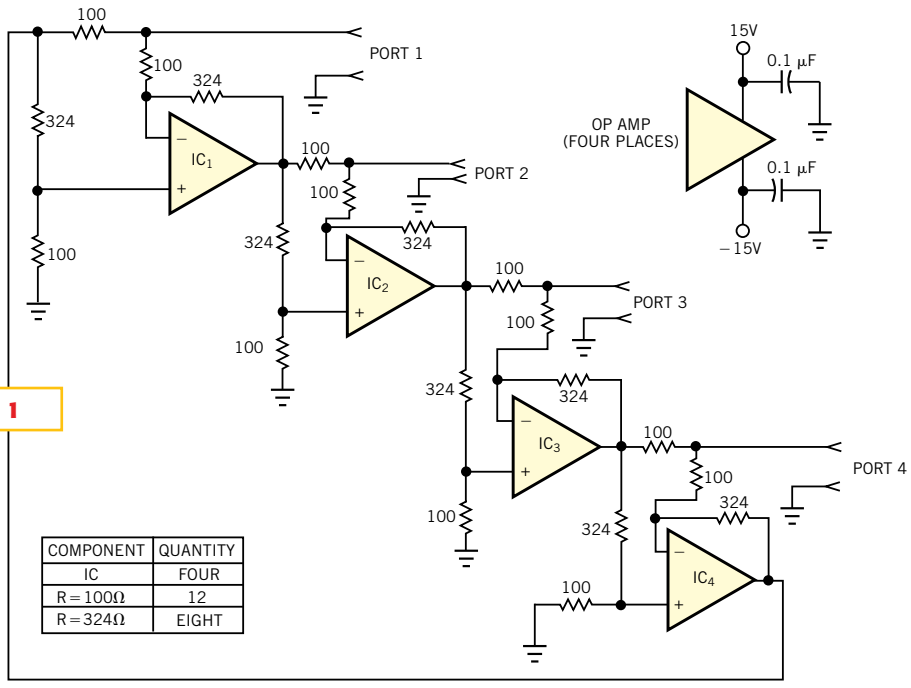
Edited by Bill Travis and Anne Watson Swager

Circuit forms low-frequency circulator

Richard Kurzrok, Queens Village, NY

THE ELECTRONIC circulator made its debut ten years ago (**Reference 1**). It functioned at VHF as a three-port unit using a Comlinear (now part of National Semiconductor, www.nsc.com) CLC 406 operational amplifier. The circuit in **Figure 1** extends the circulator's performance to four-port operation at low frequencies, using the readily available 941 (equivalent to the ubiquitous 741) and LM318 op amps. **Table 1** shows the measured data for the 741-equivalent op amp. **Table 2** shows the measured data for the LM318 op amp. The four-port circulators in **Figure 1** use 50Ω impedance levels. The circuit can readily accommodate other impedance levels, such as 75 and 600Ω. You can see that for typical circulator operation at frequencies below 50 kHz, you can use the 741-equivalent op amp. For typical operation at speeds as high as 1 MHz, you would use the LM318 op amp. The resistors in **Figure 1** are metal-film units with ±1% tolerances. The circulator breadboards use open (not shielded) construction, and the components are soldered to the vector board. The ICs use commercially available sockets soldered to the vector board.

Figure 1



COMPONENT	QUANTITY
IC	FOUR
R = 100Ω	12
R = 324Ω	EIGHT

An electronic circulator is useful for isolation and equalization.

TABLE 1–MEASURED DATA FOR 741-EQUIVALENT OP AMP

Frequency (kHz)	Forward loss (dB)	Reverse isolation (dB)
2	0.5	50.2
20	0.5	44.1
50	1	38
100	4.1	33

TABLE 2–MEASURED DATA FOR LM318 OP AMP

Frequency	Forward loss (dB)	Reverse isolation (dB)
10 Hz to 100 kHz	0	Greater than 56
100 kHz	0.1	Greater than 56
500 kHz	0.5	45
1 MHz	0.9	34
1.6 MHz	0	29.5
3.3 MHz	3	25.5

You can use the electronic four-port circulators in various applications with the fourth port terminated. You can configure baseband-amplitude and group-delay equalizers using the electronic circulator (**references 2** and **3**). You can also use the circuit as a low-frequency return-loss bridge or as an electronic isolator. Low-frequency op amps are available as quads with four independent op amps. You can configure a miniaturized, low-cost version of the circulator using surface-mount pc-board techniques.

REFERENCES

1. Wenzel, C, "Low Frequency Circulator/Isolator Uses No Ferrite or Magnet," *RF DESIGN*, July 1991.
2. Kurzrok, R, "Amplitude Equalizer is Circulator Coupled," *Microwaves*, Volume 10, September 1971, pg 50.
3. Kurzrok, R, "Circulator-Coupled Equalizers Applicable to High-Speed Data Links," *Applied Microwave & Wireless*, June 2001, pg 86.

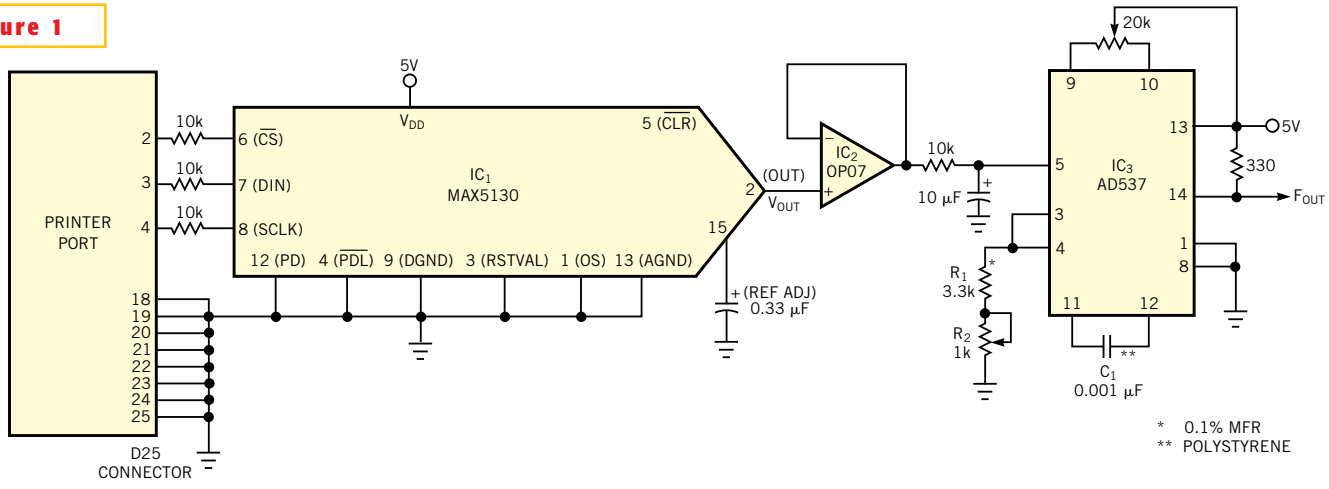
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Use printer port as programmable frequency generator

K Kanniappan, IGCAR, Tamil Nadu, India

A SIMPLE AND INEXPENSIVE circuit (**Figure 1**) and a simple C program (**Listing 1**) are all you need to turn your PC's printer port into a programmable frequency generator. Using a few low-cost and readily available components, the circuit occupies little space and is easily attachable to the printer port. The circuit has advantages over a 555-based astable multivibrator in that it eliminates the tedious task of adjusting a potentiometer while you watch a frequency counter or oscilloscope. With the circuit in **Figure 1**, you need only enter the desired frequency, and the PC does the rest. The circuit uses a MAX5130 low-power, programmable, 13-bit DAC, IC_1; an OP07 buffer; and an AD537 VFC (voltage-to-frequency converter). The PC controls the DAC using a three-wire serial interface. It also uses the data lines D0 to D2 of the data port ($0\times378_h$) of the printer interface to send the $\overline{CS}$ (chip-select), data, and CLK (clock) signals to the DAC. Depending on the data it receives from the PC, the DAC produces a voltage output of 0 to 4.0955V in 8192 steps with a step resolu-

Figure 1



Turn your PC's printer port into a programmable frequency generator with this simple circuit.

tion of 0.5 mV. Thus, a data word of $0\times000_h$ produces a DAC output of 0V, and a data word of $0\times1fff_h$ produces a DAC output of 4.0955V. Using the 2.5V internal reference, the DAC output and the data input follow the equation $V_{OUT}=2.5(DATA/8192)\times GAIN$.

After IC_2 buffers it, the DAC output drives the VFC and sets its frequency output according to the following equation: $f_{OUT}=V_{OUT}/(10(R_1+R_2)C_1)$ Hz. For the values of R_1, R_2, and C in **Figure 1** and the cited DAC-output range, the output of the VFC and hence the frequency of the programmable-frequency generator varies from 0 to 10 kHz in 8192 steps with a frequency resolution of 1.22 Hz. You choose and trim the values of R_1 and R_2 to produce a current range of 0 to 1 mA for a DAC output of 0 to 4.0955V. These values ensure good linearity (typically 0.01%) between V_{OUT} and f_{OUT}. Potentiometers P_1 and P_2 adjust f_{OUT} at the lower and higher ends of the frequency range, respectively. The C program in **Listing 1** obtains the desired frequency from the user and calculates the required output from the DAC to apply to the VFC. It then works out the ACTUALDATA to send to the DAC for mode control. The d2b routine converts the ACTUALDATA into 16-bit binary data. The program enables the DAC ($\overline{CS}$) low and then serially clocks the binary equivalent of ACTUALDATA, starting one bit at a time from the MSB to the LSB, to the data pin of the DAC. With the LSB set at the data pin, the low-to-high transition of the clock latches the ACTUALDATA completely into the DAC and sets f_{OUT} to the desired value. You can download **Listing 1** from the Web version of this article at *EDN*'s Web site, www.ednmag.com. You can easily change the frequency range by changing the value of C. For example, with R_1 and R_2 unchanged, you can extend the frequency to 100 kHz by changing C to 0.001 µF instead of 0.01 µF. You can also increase the frequency range by using a VFC with a higher frequency capability.

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

LISTING 1—PROGRAMMABLE FREQUENCY GENERATOR

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<bios.h>
#include<dos.h>

#define CLK1 0x04 /* Clock Pulse High*/
#define CLK0 0xfb /* Clock Pulse Low*/
#define CS1  0x01 /* Chip Select high to deactivate DAC*/
#define CS0 0xfe /* Chip Select low to activate DAC*/
#define DATA1 0x02 /* Data Pulse High*/
#define DATA0 0xfd /* Data Pulse low*/

int c[16],dport,ACTUALDATA,out,k;   /*Global Declarations*/
float VOUT;/*DAC OUTPUT*/

void d2b(unsigned int x, int*c)/*Routine for Decimal to Binary Conversion*/
{
int i;
for(i=0;i<=15;i++)
*(c++)=(x>>i) & 0x1;
}

float ftov()   /*Hertz to DAC output Conversion Routine*/
{
int HERTZ;
printf("\n \nEnter the frequency  within 0 to 10000Hz:");
scanf("%d",&HERTZ);
VOUT=0.00040955*HERTZ;
printf("\n VOUT=%f\n",VOUT);
return VOUT;
}

void CLOCK_DAC(void)/*Routine for clocking the DAC*/
{
out|=CLK1;
outportb(dport,out);/*Setting the clock high*/
delay(1);
out&=CLK0;
outportb(dport,out);/*Setting the clock low*/
delay(1);

}

void LOAD_DACDATA(int*c)/*Routine for loading actual data into the DAC*/
{
out|=CS1;
outportb(dport,out);/*Chip Select high to disable DAC*/
delay(1);
out&=CS0;
outportb(dport,out);/*Chip Select low to enable DAC*/
delay(1);
printf("\nDATA loaded into the DAC=");
for(k=15;k>=0;k--)
{
out|=c[k];
outportb(dport,out);
printf("%d",c[k]);
delay(1);
CLOCK_DAC();
}
out|=CS1;
outportb(dport,out);
delay(1);
}

main()
{
int v,inc;
float y;
unsigned  int x;
double fraction, integer, number;
clrscr();
printf("\tUse Your Printer Port as a Programmable Frequency Generator");
printf("\n\t\t\t by\n");
printf("\tK.Suresh,MSD,IGCAR,Kalpakkam,TamilNadu-603102,India");
dport= peek(0x40,8);/*Check up for availability of Printer Port*/
if (dport==0)
{
printf("\n\n LPT NOT AVIAILABLE! EXITING.....");
exit(1);
}
printf("\n\nAddress of the printer port found =0x%X",dport);
ftov();
y=(VOUT*8192)/(2.5*1.6384);
v=y/1;
number=y;
fraction = modf(number, &integer);
if (fraction<0.44)
inc=0;
else inc=1;
ACTUALDATA=16384+v+inc; /*Actual data including the Control Word for DAC*/
d2b(ACTUALDATA,c);
LOAD_DACDATA(c);
return 0;
}
```

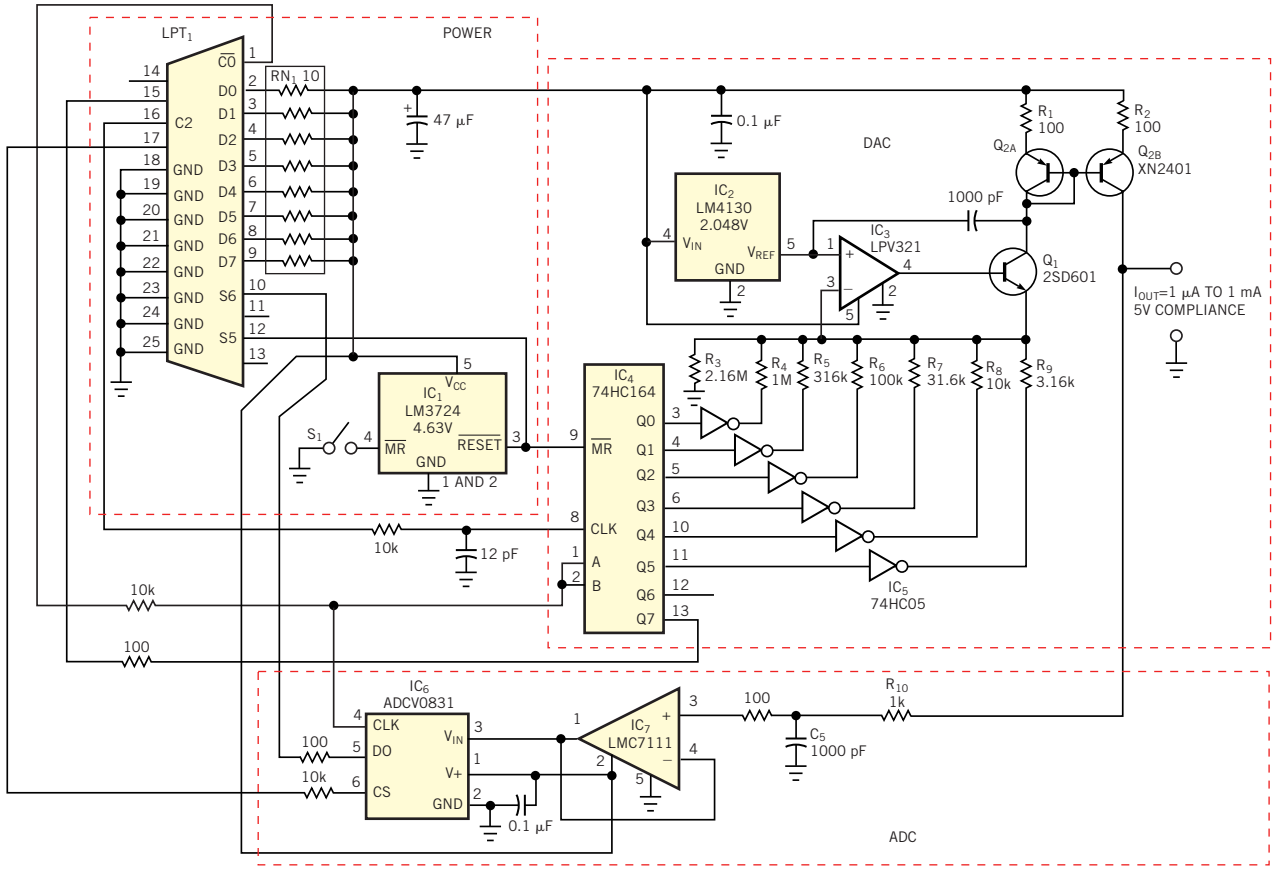
Trace voltage-current curves on your PC

Clayton Grantham, National Semiconductor, Tucson, AZ

Some years ago, one of the fundamental electronic instruments was the laboratory curve tracer. A CRT display would sweep out terminal behavior (current versus voltage) from which you could derive mathematical models. Classic presentations of diodes, transistors, and other devices enlightened designers about linear and nonlinear operation. From the displays, you could determine the bias points for optimum design performance. Today, however, you rarely find the classic curve tracers in the lab. Instead, you find design-simulation software, such as Spice, that's removed from hands-on, empirical analysis. Spice models now exist for almost all electronic components. Characterization analyzers still make the voltage-current measurements but not at the design-engineer level. Rather, departments are dedicated to characterizing processes and components and incorporating these characteristics into the simulated models. The low-cost circuit in **Figure 1** allows you to return to the hands-on approach by using your PC as a limited curve tracer.

The curve tracer sweeps out seven logarithmic-scaled currents from 1 µA to 1 mA while measuring the voltage, 0 to 5V (3.3V on some PCs), at each step. The circuit uses a programmable current source to force increasing discrete current values and samples the voltage at the I_{OUT} terminal at each step. A classic curve tracer continuously sweeps a voltage while measuring the sourced current. The program control resides in Excel (running in Office 2000) macros that perform I/O operations through the LPT_1 port of the PC. You can download the Excel program from *EDN*'s Web site, www.ednmag.com.

Figure 1



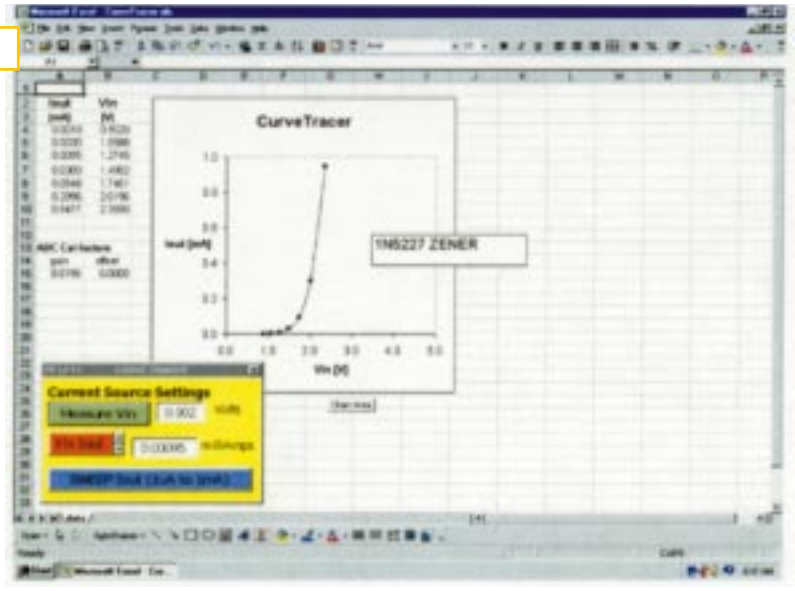
Remember the classic Tek curve tracers? You can easily configure something similar on your PC.

The program uses the free file "Input32.dll" to bit-wise control the parallel port's digital I/O. The author of the .dll file is Jonathan Titus, editorial director of *Test and Measurement World*. You load CurveTracer.xls with its macros, connect the circuit of **Figure 1** to the parallel port, and then run a macro called ControlPanel.

A user form pops up in the spreadsheet and connects the curve-tracer force and measurement actions with the electronics (**Figure 2**). The possible operations are a single voltage measurement, a single forced-current output, or a sweep of current steps lasting 2.8 msec each and a voltage measurement at each step. The voltage measurements go into cells B4 to B10 in the spreadsheet. The resulting graph shows an x-y scatter plot of the data in cells A4 to B10. With this use of macros within Excel, all the graphing, analysis, and data storage common to Excel are still available to use. You can test the terminal behavior of many electronic components with this simple curve tracer. Resistors yield a linear plot whose slope is the resistor value (R=V/I). Diodes exhibit a nonlinear plot ($I_D = I_S e^{qV_D/kT}$). You can also plot a diode's terminal behavior on a logarithmic current scale with a simple click within Excel's charting capabilities. Some other application examples are forward-biased transistor junctions, LEDs, and relay coils.

The components in **Figure 1** provide operation as low as 3V and low power consumption (low quiescent current). In the PWR block, resistor network RN_1 isolates and combines eight LPT_1 outputs at D0 to D7 to power the circuit. The supervisory circuit, IC_1, monitors the voltage from the LPT1 port. Use the LM3724 4.63V option for 5V PCs and the 3.08V option for 3.3V PCs. The reset output of IC_1 goes back to the parallel port at terminal S5 for software-error checking and clears IC_4 at start-up. IC_1 also has a manual reset that provides direct user control. If you press momentary switch SW_1, the output current resets to 1 µA. IC_2 through IC_5, Q_1, Q_2, and associated resistors R_1 through R_9 form a current-output D/A converter. Servoamplifier IC_3 sets Q_1's collector current. This current is a function of IC_2's reference voltage divided by the parallel combination of R_3 to R_9. For the lowest current, 1 µA, only R_3 connects (2.048V/2.16 MΩ). For the highest current (1 mA), all the resistors connect in parallel (2.048V/2.16 kΩ).

Figure 2



This curve-tracer user form, which floats in front of an Excel spreadsheet, controls the curve tracer's electronics.

You select resistor values for a cumulative half-decade change in I_{OUT}. I_{OUT} steps by the square root of 10 in value. With only these seven resistors, the circuit covers three decades of current range. The pnp pair, Q_{2A} and Q_{2B}, mirrors Q_1's collector current to the terminal I_{OUT}. Emitter-degeneration resistors R_1 and R_2 improve the mirror's output resistance. Shift register IC_4 and open-drain inverter IC_5 select which of the resistors to connect via program control. IC_4's input and clock connect the parallel port at C0 and C2 for serial shift-in operation. IC_5's on-resistance is lower than 40Ω. IC_6 and IC_7 perform voltage measurements (the ADC block). IC_7, a rail-to-rail op amp buffers the lowpass filter comprising R_{10} and C_5. The serial output of IC_6, D0, connects to the parallel port at S6. IC_6's clock input provides timing control. When IC_6's chip-select input goes low, a conversion starts. Pulling I_{OUT} above the PC's 5V level or below ground could result in circuit and PC damage. To be safe, operate this curve tracer with unpowered components.

The macros in the downloadable listing contain the basic interface features for changing the current-output values and measuring the voltage input. Within module 1, the declaration of Input32.dll must include its directory path. To minimize the effects of differing PC-clock and LPT_1-bus speeds in different PCs, the user form performs a 10-sec timing calibration at initialization. This calibration attempts to set the I_{OUT} steps during a sweep to 2.8 msec. The software uses this time-delay coefficient throughout the program. Also, software-calibration coefficients within the code minimize voltage-measurement gain and offset errors. The spreadsheet maintains these coefficients for ease of changing. Use a voltmeter and a resistor of known value to calibrate. The initial gain coefficient in the spreadsheet for a 5V PC is 5V divided by 2^8-1 (5/255=0.0196). The initial offset coefficient is zero. You can also calibrate the I_{OUT} current values in the spreadsheet. The user-form references these spreadsheet values. With external calibration, you can attain better than 1% error.

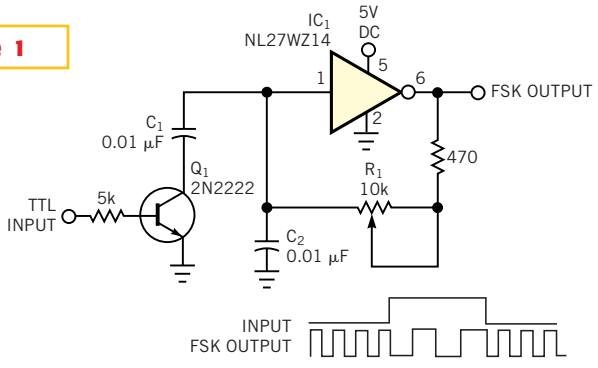
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Circuit makes simple FSK modulator

Shyam Tiwari, Sensors Technology Ltd, Gwalior, India

THE NEED FOR a compact telemetry system poses a challenge for designing a small, light, low-component-count system. Interfacing serial data from the microprocessor is also difficult because most low-cost RF transmitters do not accept dc levels at the input. Commercial FSK (frequency-shift-keying) modulators are bulky and need many passive components. The circuit in **Figure 1** uses a single NOT gate (inverter), an On Semiconductor NL27WZ14 in a surface-mount package, to generate continuous FSK data from TTL-level signals. The outputs from this circuit are compatible with available transmitters. When the TTL input has a low level, the circuit is a continuously running oscillator, producing approximately 2400 Hz (adjustable with R_1). When the input assumes a high level, the oscillator's frequency reduces by one- half with the introduction of a capacitor in the timing circuit via Q_1. The inverter IC can accommodate an operating frequency of approximately 80 kHz. You can easily operate the FSK modulator at higher frequencies, such as 4800 and 9600 Hz, by reducing the values of the timing capacitors C_1 and C_2.

Figure 1



An FSK modulator uses a single inverter with minimal added components.

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

design ideas

Edited by Bill Travis

Circuit forms efficient cosine calculator

Matt Kornblum, Bradenton, FL

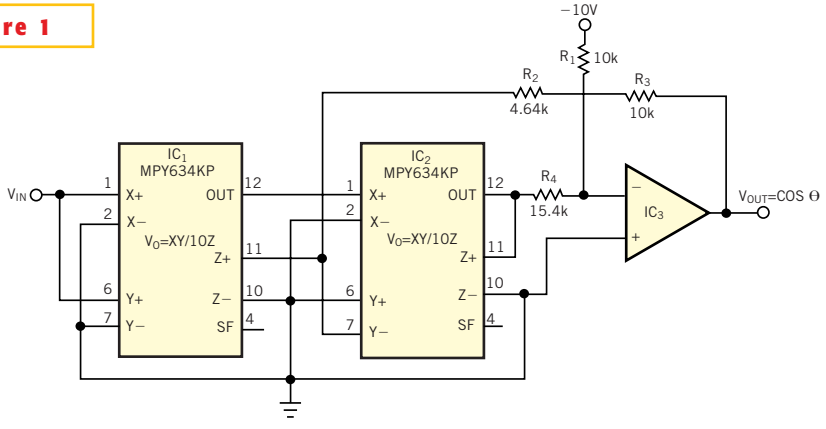
THE CIRCUIT in **Figure 1** converts a ±10V analog voltage representing an angle between θ_{MIN} and θ_{MAX} and emits a voltage equal to 10 cosθ. This circuit can have an accuracy of better than 1% over ±120° or better than 0.2% over ±90°. These figures represent an order-of-magnitude improvement over a Taylor-series estimate for the same range and for the same number of multiplications. The Taylor-series definition for a cosine (with θ in radians) is:

$$\cos\theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} \cdots + \frac{\theta^n}{n!}.$$

The series works well for high values of n or small angles. Generally, for n=4, significant errors start to accumulate for angles exceeding ±45°. When you use a Taylor-series expansion for better accuracies at larger angles, the number n becomes larger and demands more resources from the design. The Taylor series for n=4 has the form of $f(\theta)=a-b\theta^2+c\theta^4$, where a=1, b=0.5, and c=0.041667 (for angles in radians). By using a least-squares curve fit to optimize this function at n=4, you can find coefficients that allow you to obtain significantly better accuracies over the desired input range without raising the value of n to more than 4. The circuit in **Figure 1** embodies this least-squares approach.

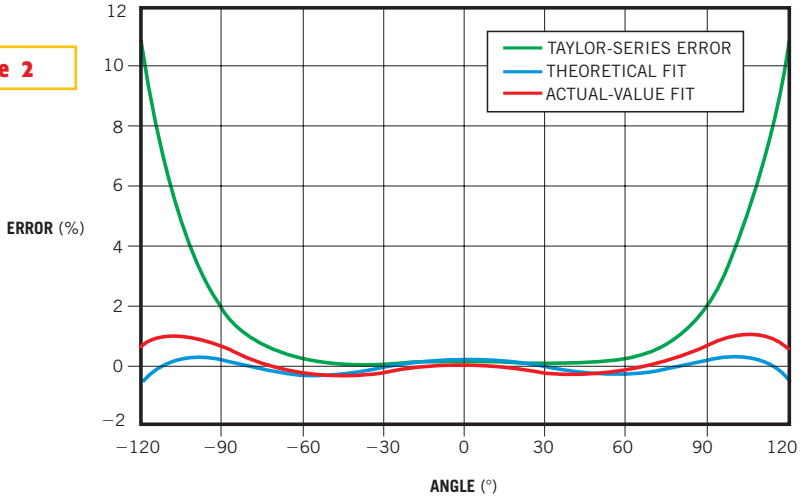
Choosing the resistor values for the circuit is relatively simple. Set R_1 and R_2 equal to each other (for 10V maximum input and a≈1), and determine values for R_2 and R_4 by applying the following equations:

Figure 1



By manipulating Taylor-series coefficients, you can obtain better accuracy when generating cosines.

Figure 2



For angles greater than ±90°, the revised coefficients in Figure 1 yield significant accuracy improvements in calculating cosines.

$$R_2 = \frac{R_3}{b\theta^2_{MAX}},$$

and

$$R_4 = \frac{R_3}{c\theta^4_{MAX}}.$$

IC_1 generates the square of V_{IN} and negates it. This output sums through R_2 into IC_3. IC_2 generates the fourth power of V_{IN} and sums it into IC_3 through R_4. A −10V reference across R_1 creates the "a"-coefficient constant current into IC_3. The output of IC_3 is the sum of the three terms. Because IC_1 is an inverting amplifier, the circuit configures the multipliers such that the output of IC_1 is positive and the output of IC_2 is negative. Choosing the proper 0.1% resistors can improve circuit accuracy to better than 1% for −120 to +120°. You should use a low-offset op amp for best results. **Figure 2** shows the Taylor-series error, the theoretical fit, and the actual fit. For a fit in a 90° range, the values change slightly, and the errors across the range become significantly smaller. The constant "a" becomes 0.9996, b=0.4962, and c=0.0371. Then, $R_1=R_3=10$ kΩ, $R_2=8.16$ kΩ, and $R_4=44.2$ kΩ.

You can use the same approach to efficiently calculate cosine and sine values in a DSP system more rapidly than using a look-up table.

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

Reference stabilizes exponential current

Tom Napier, North Wales, PA

In an antilog converter, the difference between the base voltages of two transistors sets the ratio of their collector currents:

$$I_1/I_2 = e^{V_{BE}q/kT}.$$

The use of matched transistors balances the first-order temperature coefficient but leaves a temperature-dependent gain term, q/kT. Classic antilog circuits use a thermistor in the drive circuitry to correct this temperature dependency. However, if the control input is a fraction of some reference voltage, as when you use a manual potentiometer or a DAC, you can achieve an exact temperature correction by adding a second reference transistor. **Figure 1** shows three of the five transistors in a CA3046 array. Q_1 is the exponential current source, and Q_2 is the conventional reference transistor. IC_2 forces Q_2's collector to ground so its collector current, 1 mA in this example, is simply the reference voltage divided by R_3. Typically, this current equals the maximum output required from Q_1; lower currents result from negatively driving the transistor's base.

The attenuator on the base of Q_1, R_1, and R_2 reduces the effects of IC_1's offset voltage. IC_3 drives the base of Q_3 via a second attenuator, R_4 and R_5, forcing its collector to ground. The reference current through Q_3 is a fraction of the main reference—one-tenth in this example. Despite the chip temperature, the base voltage of Q_3 is exactly the value you need to generate a 1-to-10 current ratio. Because IC_3's output supplies the reference voltage for the potentiometer, the ratio of the two attenuators defines the full-scale-current-adjustment range. If the ratio is 4 to 1, the output current has a four-decade tuning range that's independent of temperature. The circuit in **Figure 1** is dynamically stable, using either low-power or fast op amps.

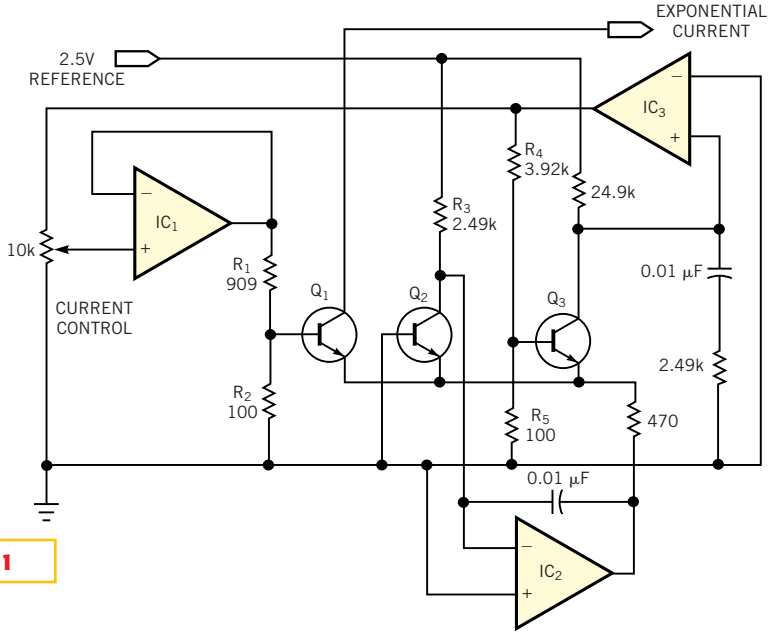


Figure 1 The use of a second reference eliminates temperature dependency in an antilog-ratio circuit.

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

Microcontroller becomes multifunctional

Abel Raynus, Armatron International, Melrose, MA

A microcontroller, by default, can execute only one program at a time. What do you do if, in a given project, you need to perform more than one operation at a time? Add more microcontrollers to the design? In certain cases it's unnecessary. Consider a real-life situation (**Figure 1**). The microcontroller constantly generates on its Pulse output pin a sequence of pulses with 25-msec duration and a repetition rate of 1 or 4 sec, depending on the state of the Rate input pin. LED illumination accompanies the pulse generation. Suppose that the microcontroller must simultaneously and independently perform some other functions using the rest of its six I/O pins. You can benefit from the fact that the pulse duration is much smaller than the repetition period. During this relatively long period, the microcontroller may not just wait for the generation of the next pulse, but, instead, it may perform some other operation. You organize the pulse-generating program as an interrupt-service routine and the rest of the program as a main program. To avoid any interference between these parts of the software, the interrupt-service routine execution time should be shorter than the smallest period of pulse repetition.

Listing 1 is the assembly routine for multifunctional operation. To make the interrupt program repeatable after the predetermined time interval, the best choice is to use the microcontroller's internal timer. This microcontroller has two timing options: timer-overflow interrupt and RTI (real-time interrupt). For a 2-MHz operating frequency, the timer overflow occurs every 0.51 msec.

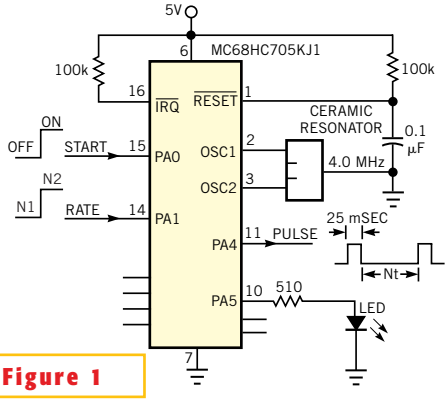


Figure 1

Between pulses, the microcontroller can perform other tasks using the software in Listing 1.

LISTING 1—ROUTINE FOR MULTIFUNCTIONAL OPERATION

```
                          1  ********** Multifunctional operation   **********
 0000                     2  $include "std-jla.asm" ;
 0000                     3  $PAGEWIDTH 160
 07F1                     4          org MOR;    resistor osc and input pulldown
 07F1       20            5          fcb %00100000;
                          6  ******  I/O PORT BITS ****************************
 07F2                     7  start    equ pA0
 07F2                     8  rate     equ pA1
 07F2                     9  pulse    equ pA4
 07F2                    10  LED      equ pA5
                         11  ********** CONSTANTS ****************************
 07F2                    12  N1       equ 16T  ;rate 65.5ms x16=1sec
 07F2                    13  N2       equ 64T  ;rate 65.5ms x64=4sec
                         14  ********** VARIABLES ****************************
 00C0                    15          org RAM
 00C0                    16  N        rmb 1 ; time counter
                         17  ******     INITIALIZATION  **********************
 0300                    18          org ROM
 0300 [02] A630          19  init     lda     #%00110000 ;set I/O prtA
 0302 [04] B704          20          sta     ddrA
 0304 [05] 3F00          21          clr     prtA ;all output devices off
 0306 [05] 3FC0          22          clr     N
 0308 [02] 9A            23          cli         ;Interrupt enable
                         24  ***********************************************
 0309 [05] 0100FD        25  MAIN:    brclr  start,prta,main ;wait for start
 030C [05] 1808          26          bset   RTIE,TSCR        ;start pulse
generating
 030E [05] 1A00          27          bset   LED,prtA         ;set LED on
 0310 [02] 9D            28  work     nop  ;this instruction for illustration only
                         29          ; perform some other operations
                         30          ;
 0311 [05] 0000FC        31          brset  start,prtA,work ;continue pulse
generating ?
 0314 [05] 1908          32          bclr   RTIE,TSCR        ;stop pulse generating
 0316 [05] 1B00          33          bclr   LED,prtA         ;set LED off
                         34          ;perform some other operations
                         35          ;
 0318 [03] 20EF          36          bra    MAIN
                         37  *           Real Time Interrupt Service routine
                         38
*********************************************************
 031A [05] 3CC0          39  RTI      inc     N                ;N + 1
 031C [03] B6C0          40        lda    N
 031E [05] 030010        41        brclr  rate,prtA,rN1     ;if rate=0,go to rN1
 0321 [03] B140          42          cmp    N2
 0323 [03] 2509          43          blo    res
 0325 [05] 3FC0          44  pls      clr    N
 0327 [05] 1800          45          bset   pulse,prtA        ;generate one pulse
 0329 [06] CD0337        46          jsr    dly25ms           ;  with 25 ms width
 032C [05] 1900          47          bclr   pulse,prtA        ;
 032E [05] 1408          48  res      bset   RTIFR,TSCR        ;RTIF reset
 0330 [09] 80            49          rti    ;return from Real Time Interrupt
 0331 [03] B110          50  rN1      cmp    N1
 0333 [03] 25F9          51          blo    res
 0335 [03] 20EE          52          bra    pls
                         53
*********************************************************
 0337 [02] A620          54  dly25ms lda #32T      ;25ms delay
 0339 [03] 5F            55  lp1      clrx
 033A [03] 5A            56  lp2      decx
 033B [03] 26FD          57          bne lp2
 033D [03] 4A            58          deca
 033E [03] 26F9          59          bne lp1
 0340 [06] 81            60          rts
                         61  ***********************************************
 07F8                    62          org    VECTORS
 07F8       031A         63          fdb    RTI
 07FE                    64          org    VECTORS+6
 07FE       0300         65          fdb    init
```

You can program the RTI period to be as long as 65.5 msec (with RT1-to-RT0=1-to-1). To simplify the counters, it is reasonable to choose the largest value: 65.5 msec. Then, to make the repetition periods equal to 1 and 4 sec, you create the counters modulo 16 and 62 accordingly in the RTI routine (**Listing 1**). You can see in **Listing 1** that the microcontroller waits for a high level on its Start pin to begin pulse generation and LED lighting. During the interval between pulses, it performs the other operations, continuously checking the state of its Start pin. After receiving a low level on the Start pin, the microcontroller stops pulse generating and switches off the LED. You can download the software for multifunctional operation from the Web version of this article at www.ednmag.com.

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

Circuit converts pulse width to voltage

James Mahoney, Linear Technology Corp, Milpitas, CA

THE CIRCUIT IN **Figure 1** converts pulse information to a clean dc voltage by the end of a single incoming pulse. In another technique, an RC filter can convert a PWM signal to an averaged dc voltage, but this method is slow in responding. Converting low-duty-cycle pulse information is slower yet. The circuit in **Figure 1** uses two low-input-bias-current LT1880 op amps, IC_2 and IC_3, and an LTC202 quad analog switch, IC_{1A}, IC_{1B}, IC_{1C}, and IC_{1D}, to configure the integrator and sample-and-hold stages that convert a single pulse to a dc voltage. The circuit's output is stable after a single pulse. This example shows the conversion of a low-duty-cycle positive pulse, whose width varies from 1 to 2 msec with a period of 25 msec, to a clean dc voltage. The input pulse starts, stops, and resets the integrator and controls the input to the sample-and-hold stage. After the reset operation, the positive pulse level-triggers the integrator, comprising R_1, C_1, and IC_2. The sample-and-hold stage, comprising IC_{1B}, C_2, and IC_3, is in the sample mode, sampling the output of the integrator, while the incoming pulse is high.

When the incoming pulse goes low, the circuit disconnects the input to the sample-and-hold stage, putting it into hold mode. The integrator then stays in the reset state until the next positive pulse arrives. During reset, analog switch IC_{1A} opens to disconnect the integrator's input, switch IC_{1C} closes to reset integration capacitor C_1, and switch IC_{1B} opens to disconnect the input to the sample-and-hold stage, placing the stage in hold mode. Analog switch IC_{1D} inverts the on/off states of switch IC_{1C}. The

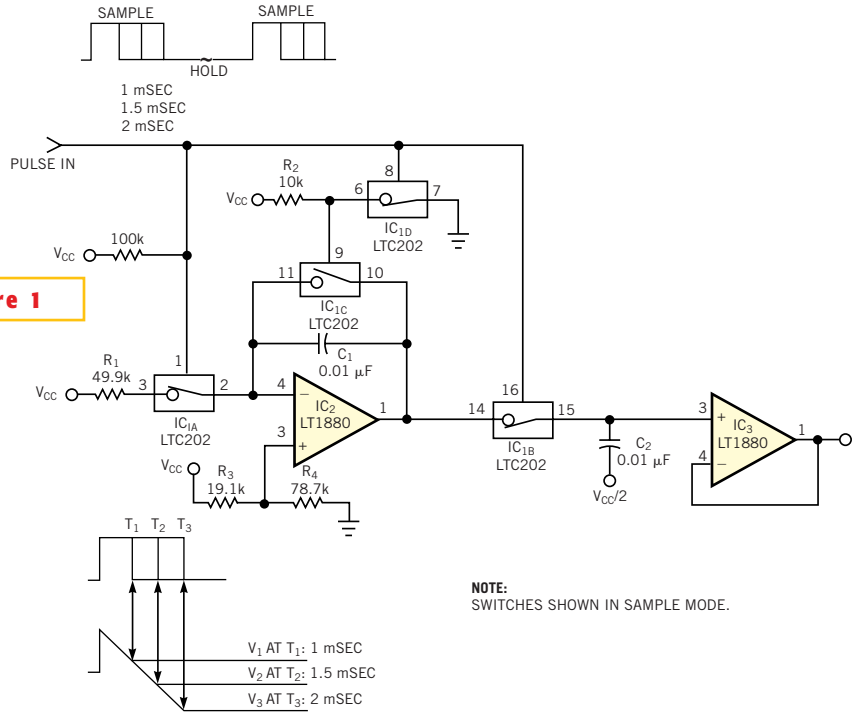


Figure 1

This circuit yields a clean dc voltage that indicates the width of an incoming pulse.

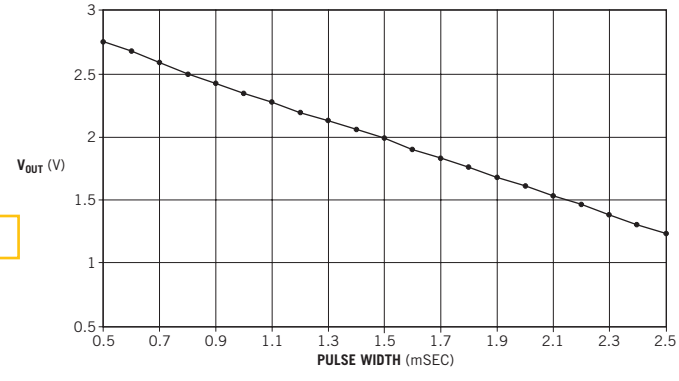


Figure 2

The circuit in Figure 1 linearly converts a pulse width to a dc voltage.

LT1880 op amp is a good choice for the integrator and sample-and-hold stages because of its maximum input-bias current of 900 pA at 25°C and maximum of 1500 pA maximum over the full −40 to +85°C ambient-temperature range. Another benefit of the LT1880 is its maximum input-offset-voltage drift of 1.2 µV/°C. Integrator capacitor C_1 and resistor R_1 set the conversion gain.

You should use polypropylene, polystyrene, or Teflon capacitors for C_1 and C_2 to minimize integrator drift and sample-and-hold droop rate. The voltage ratio that resistors R_3 and R_4 set establishes the dc level at the positive pulse's midrange value: 1.5 msec in this example. **Figure 2** shows input pulse width versus output voltage. You can easily modify the circuit in **Figure 1** to yield different conversion gains, output levels, and swings for different pulse widths. The circuit operates with pulse-width information and not duty-cycle values. The sample-and-hold stage is an analog-memory element that reveals the dc-voltage equivalent for this pulse width.

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

Short dc power-line pulses afford remote control

Tom Hornak, Portola Valley, CA

If you face the challenge of adding a second, independently controlled light source to an existing ceiling lamp controlled by a wall switch, you may find that stringing a second power line is impossible. First, you can replace the wall switch by the circuit in **Figure 1**. Pushing the on switch S_1 or S_2 for approximately 1 sec inserts the 12V zener diodes D_1 or D_2 in series with the hot wire of the power line. During the push, the polarity-dependent conduction of the zener diodes creates a small positive (negative for D_2) dc component across the line and only slightly reduces the line's 120V-ac component. A control circuit at the lamps' site reacts selectively to the polarity of this dc pulse and controls the power to the two lamps. The required power rating of the two zener diodes depends on the load current. The short duration and low duty cycle of the activation are helpful. The 1N2976 diodes in **Figure 1** are rated for continuous dissipation of 10W.

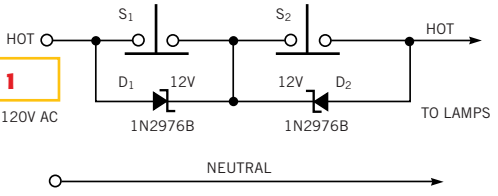


Figure 1 This circuit creates dc pulses for use with control circuitry located at the load.

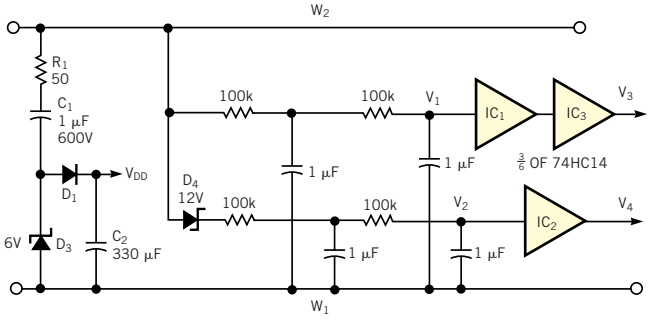


Figure 2 This control circuit uses dc pulses from the circuit in Figure 1 to drive triacs in the circuit in Figure 3.

Figure 2 shows the first part of the control circuit located at the lamps' site, including the two leads of the power line, W_1 and W_2. Current through capacitor C_1 and resistor R_1 creates a 60-Hz square wave across the 6V zener diode, D_3. Diode D_1 and filter capacitor C_2 generate a dc supply voltage of V_{DD}=5V for the control circuit's active elements. Two two-stage RC filters connected to W_2 create V_1 and V_2, with reference to W_1. The filters attenuate the 120V-ac voltage between W_1 and W_2 to a subvolt level in V_1 and V_2. An extra zener diode, D_4, creates a positive 5V dc bias in V_2. The filter outputs V_1 and V_2 drive inverting Schmitt triggers IC_1 and IC_2. Inserting zener diode D_1 by pushing S_1 in **Figure 1** changes V_1 from 0V to 5V and V_2 from 5V to 10V. Inserting D_2 in **Figure 1** by pushing S_2 changes V_1 from 0V to −5V and V_2 from 5V to 0V. Note that the input-protection diodes of IC_1 and IC_2 limit the voltage swings of V_1 and V_2. The output V_3 of IC_3 responds to pushing S_1 by a positive transition and has no response to pushing S_2. The output V_4 of IC_2 responds to pushing S_2 by a positive transition and has no response to pushing S_1.

Figure 3 shows the second part of the control circuit located at the lamps' site. Signals V_3 and V_4 in **Figure 2** drive the clock input of toggle flip-flops IC_1 and IC_2, respectively. For clarity, **Figure 3** doesn't show the connections of the flip-flops of $\overline{Q}$ to D and the Set terminal to V_{DD}. When you push switch S_1 in **Figure 1**, the positive transition in V_3 toggles flip-flop IC_1. Similarly, when you push S_2, the positive transition in V_4 toggles flip-flop IC_2. Thus, you can independently control the states of flip-

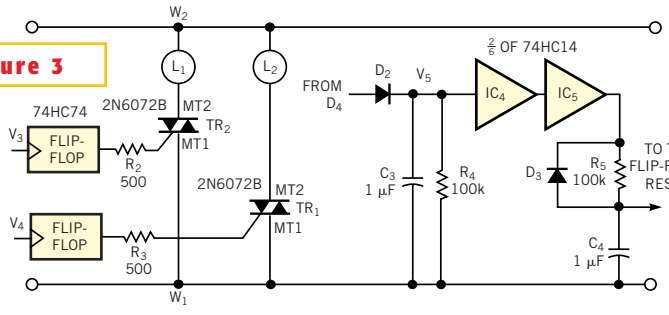
flops IC_1 and IC_2 by pushing S_1 and S_2, respectively. To drive the two lamps, the Q outputs of the flip-flops drive the gates of triacs TR_1 and TR_2 via coupling resistors R_2 and R_3. The MT2 terminal of each triac drives the lamps, L_1 and L_2, respectively. Pushing S_1 changes the state of lamp L_1; pushing S_2 changes the state of lamp L_2. Thus, you have independent control of both lamps on a single power line. In this application, you want to keep each lamp's terminals safely connected to the hot and neutral wires. Therefore, you make W_1 the hot wire and W_2 the neutral wire.

With the control circuit, the state of the flip-flops becomes uncertain if, after an interruption, the ac power returns. This situation is unacceptable because the lamps could turn on and stay on for an uncontrollable length of time. Therefore, you add a power-up reset circuit (**Figure 3**). To guarantee a safe reset also for short interruptions, the reset circuit must quickly pull down the flip-flops' Reset terminal, which is independent of V_{DD}'s slowly dropping level. Diode D_2 (driven by the 60-Hz square wave across zener diode D_4), capacitor C_3, and resistor R_4 act as an auxiliary rectifier supplying voltage V_5. When power experiences an interruption, V_5 drops to 0V much faster than V_{DD}. V_5 drives the cascade of inverting Schmitt triggers IC_4 and IC_5, which then quickly pull down the flip-flops' Reset terminals via diode D_3. When power returns, the Reset terminals pull up slowly via resistor R_5. The R_5C_4 time constant guarantees that the flip-flops' Reset does not release before V_{DD} reaches its full value.

Note that you can use this Design Idea in other applications. For example, if you omit the circuit in **Figure 3,** the transitions in V_3 and V_4 can drive an up/down counter that can perform an auxiliary control function for a device that the ac line powers. If you insert an additional conventional switch in series with the circuit in **Figure 1**, it can turn on and off the power to the device. If the application requires control signals near ground level, wire W_1 should be the neutral lead of the power line, and W_2 should be the hot lead. However, make sure that the powered device is not an inductive load because it can short out the controlling dc pulses.

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

Figure 3



Triacs independently control two loads based on signals from a pair of wall switches.

design ideas

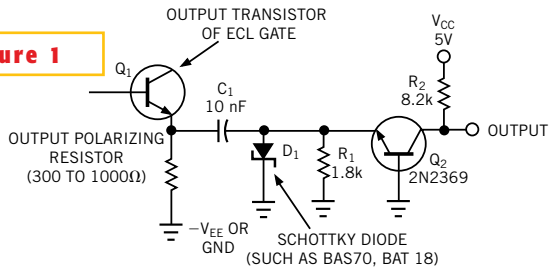
Edited by Bill Travis

Trigger a TTL circuit from ECL levels

Lukasz Sliwczynski, University of Mining and Metallurgy, Krakow, Poland

ECL circuits typically have relatively small logic spans of approximately 800 mV. Because of the small span, to drive TTL circuits from ECL levels normally entails the use of level converters, such as the MC10125, or comparators. Such circuits are relatively power-hungry and expensive. However, they are sometimes simply unnecessary. The circuit in **Figure 1** allows you to trigger some TTL circuitry by generating a fairly short negative-going pulse from the trailing edge of the ECL signal. The main requirement for the circuit to work is that the rate of ECL signal be in the tens of kilohertz. Such signals sometimes appear at the rear panels of some older types of measurement equipment. Such equipment can include sampling oscilloscopes or time-domain reflectometers, such as the 7S12 or 7S14 from Tektronix. In a measurement setup, the circuit in **Figure 1** exploits the sampling gate from a 7S12 plug-in unit.

Figure 1



You don't need an expensive level-converter IC to provide a TTL-level trigger from an ECL-level signal.

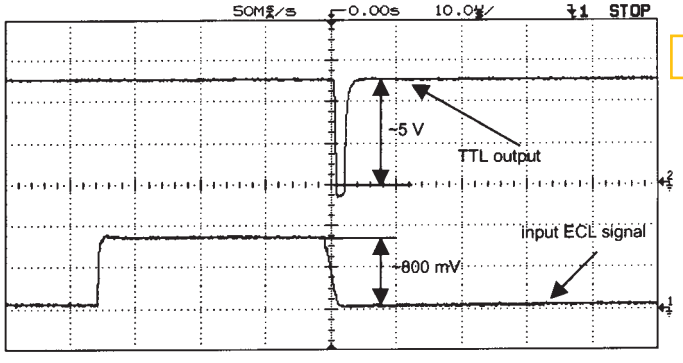
Figure 2 shows the waveforms associated with the circuit in **Figure 1**. The positive portion of the ECL signal charges capacitor C_1 through the Schottky diode, D_1. In this part of the operating cycle, transistor Q_2 is off, and the output voltage is approximately 5V. On the negative-going edge of the driving pulse, the charge from coupling capacitor C_1 causes the base-emitter junction of Q_2 to conduct, driving the transistor into saturation. The output voltage assumes a level slightly below 0V. The duration of the generated negative-going pulse depends on the speed with which C_2 discharges. The discharge takes place through the base-emitter junctions of Q_1 and Q_2 and resistor R_1. The duration is difficult to calculate, but for a rough estimate, you can use the following equation:

$$t_P = R_1 C_1 \ln \frac{\Delta V - V_{DS}}{V_{BE}} \approx 0.08\, R_1 C_1,$$

where $\Delta V \approx 0.8V$ is the ECL span, $V_{DS} \approx 0.15V$ is the voltage drop of the Schottky diode, and $V_{BE} \approx 0.6V$ is the voltage drop of the base-emitter junctions. In practice, the durations are shorter than predicted because the equation does not take account of the base-emitter resistances of Q_1 and Q_2. For the components in **Figure 1**, the duration is approximately 2 µsec. The crucial component in the circuit is D_1, which must be a Schottky type, because of the voltage swing of the ECL signal, which is nearly the same as the base-emitter voltage of the conducting silicon transistor. Proper operation of the circuit occurs because of the voltage difference between Schottky and silicon-junction levels, which is typically 0.1 to 0.3V. This difference allows for the strong saturation of Q_2 just after the trailing edge of the ECL signal.

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

Figure 2



The circuit in Figure 1 provides a 2-µsec, 5V negative-going trigger from an 800-mV ECL signal.

Microcontroller discerns addresses in RS-485 systems

Nigel Brooke and Ted Salazar, Maxim Integrated Products, Sunnyvale, CA

One of the many benefits of using the RS-485 data-interface system, unlike the RS-232 system, is its ability to implement multidrop networks. Such networks usually carry 9-bit data words, in which the ninth (parity) bit identifies each word as address or data. When using small microcontrollers without a hardware UART, such as IC_1 in **Figure 1**, designers must decide whether to add an external hardware UART or to configure a UART in software. External UARTs once represented a large increase in board area, complexity, and cost, and the available UARTs were usually overkill for simple microcontroller applications. On the other hand, sparing the program memory and processor resources you need for a software-based UART can sometimes be difficult. The program memory in IC_1, for example, has only 1k×14 bits of EEPROM. You have a third alternative—a small, low-cost external UART, IC_2. The use of this device liberates the program memory you otherwise need for a software-based UART.

An RS-485 bus can carry as many as 256 transceiver modules of the type in **Figure 1**. IC_3 is the RS-485 transceiver, and IC_4 is a "microcontroller supervisor" that holds the microcontroller in a reset state until a valid supply voltage is present. You can download the assembly-language program for the microcontroller from the Web version of this Design Idea at *EDN*'s Web site, www.ednmag.com. The application in **Figure 1** is a slave-test configuration, but you can modify the code to accommodate any specific RS-485 address-recognition application. The circuit works as follows: When the bus transmits an address, IC_2 in each slave module initiates a parity interrupt. IC_1 in each module then reads all the data in its internal FIFO, locates the address word, and compares that address with its own address stored in the eight DIP switches. A match causes the slave to clear the interrupt and transmit (to the master) an ASCII "A" (41_h), followed by its own address. If the slave module reads the FIFO's contents without finding a match, it clears the current address-word interrupt and waits for the next one.

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

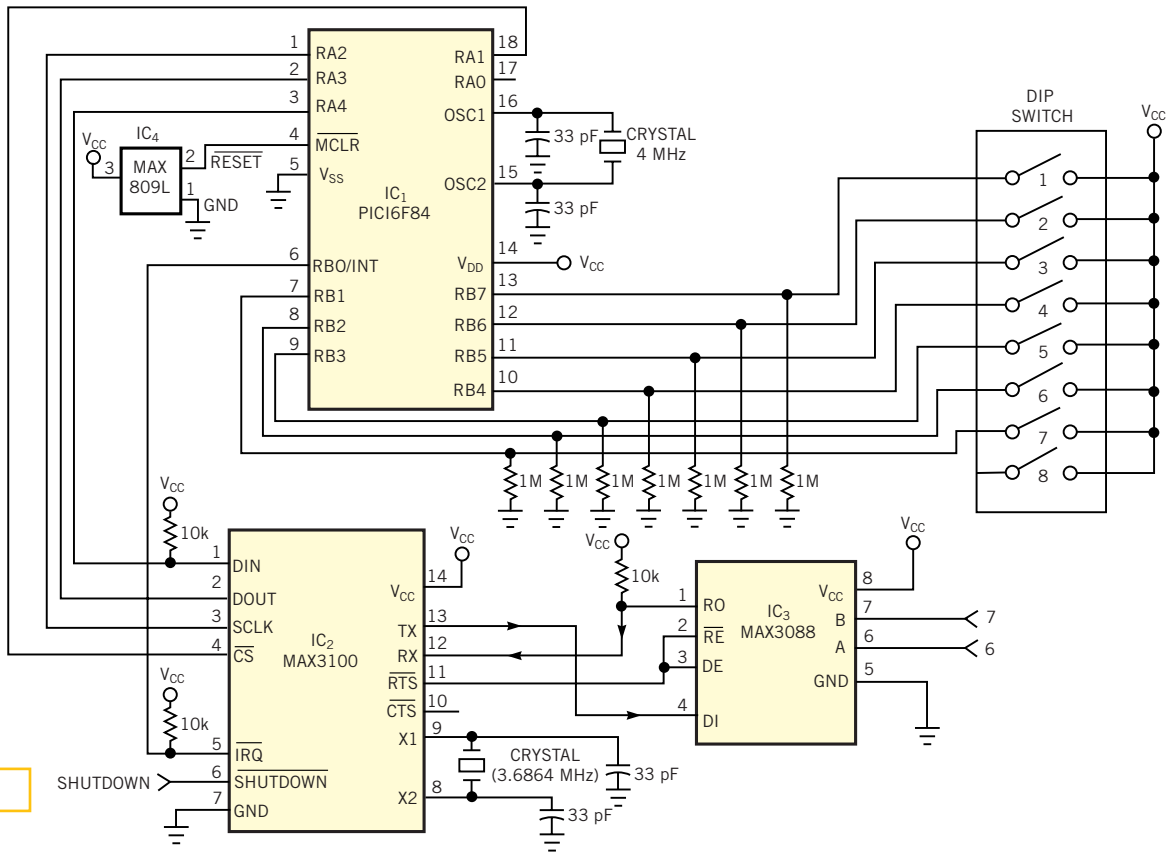


Figure 1

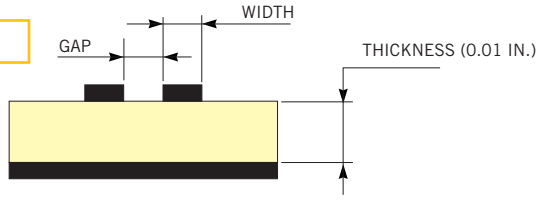
Adding a small UART, IC_2, and microcontroller, IC_1, to the RS-485 transceiver, IC_3, forms a slave data-transceiver module that responds to its own network address.

PC-board layout eases high-speed transmission

Gregory Adams, Moorestown Microwave Co, Moorestown, NJ

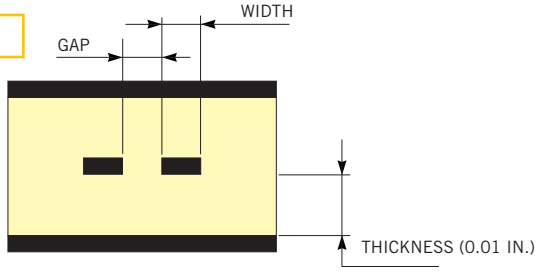
As digital techniques move to higher speeds, designers become aware of the need to treat pc-board traces as RF transmission lines. In these lines, you strive to hold the line impedance, Z_0, to a constant value—typically, 50Ω—and to terminate the line with the same impedance. Data families such as ECL, PECL, and LVDS send data over a pair of traces known as a balanced transmission line. One line switches high, while the other switches low. As with other high-speed logic families, you must hold the transmission-line impedance constant and properly terminate the line. If the spacing between the pair of traces is large, then you can design the traces as simply two 50Ω transmission lines. On the other hand, if the spacing between the traces is less than several times the board thickness, then the effect of one trace on the other changes the characteristic impedance of the line.

Figure 1



A microstrip transmission line has traces on one side of a pc board and a ground plane on the other side.

Figure 2



A stripline transmission line has two traces embedded in the pc board and ground planes above and below.

In RF parlance, when equal voltages drive the two lines, the resulting impedance of each individual line to ground is called Z_0 Even, or Z_{0e}. When equal and opposite voltages, as with differential signaling, drive the two lines, the impedance of one line to ground is called Z_0 Odd, or Z_{0o}. You need to concern yourself only with Z_{0o}, because it applies the to the impedance of a differential-data transmission line. The Z_{0o} of a differential pair is always lower than the Z_0 value of a single trace having the same width on the same board. To hold the impedance of a transmission line to some required value, you must make the traces narrower than would be the case with a single trace. Generally, this fact is good news for digital designers who need to make those transmission lines fit between the vias under a dense BGA chip.

If the traces are on the top of a board with a ground plane under them, then you can model them as coupled "microstrip" lines (**Figure 1**). On the other hand, if the traces are in a layer with ground planes above and below them, then you can model them as coupled "striplines" (**Figure 2**). In the stripline case, you assume that the pair of transmission lines is sandwiched between the two ground planes and that the board thicknesses to the top and ground planes are equal. **Tables 1** and **2** show the line width required to hold Z_{0o} constant at 50Ω for various values of the gap between the two traces. **Table 1** applies to the microstrip case with lines on top of the board; **Table 2** applies to the stripline case with lines sandwiched between equally spaced ground planes. Note that the trace widths are much smaller in the stripline case because of the second ground plane. Both tables assume a board thickness of 0.01 in. You can directly scale the line widths and gaps for other dielectric thicknesses. In every case, the dielectric material is FR-4 with a dielectric constant of 4.6. The tables use the old DOS version of HP's Appcad, a program HP distributes as freeware. The newer versions of this program do not handle coupled lines. To calculate the impedance, Z_0 Odd, of differential transmission lines of other dimensions, you can download a copy of Appcad from www.geocities.com/gregs downloadpage.

TABLE 1–DIMENSIONS FOR 50Ω MICROSTRIP

Gap (in.)	Width (in.)
0.005	0.011
0.01	0.0142
0.015	0.0158
0.02	0.0166
0.025	0.0171
∞	0.0185

TABLE 2–DIMENSIONS FOR 50Ω STRIPLINE

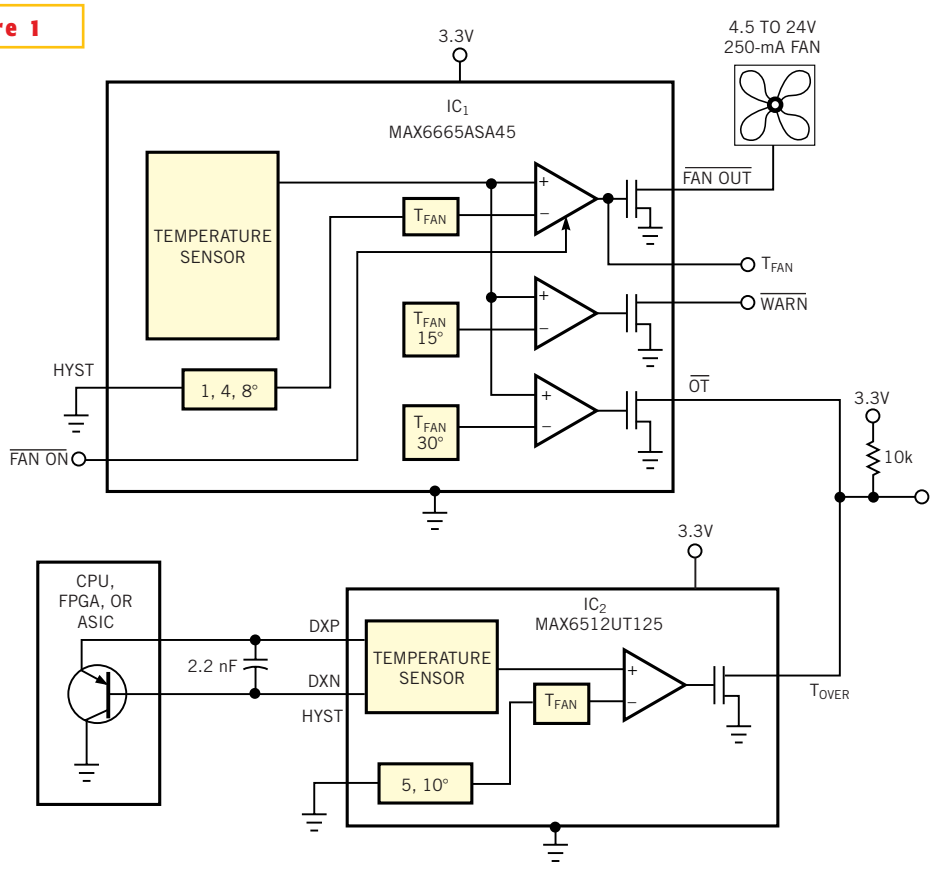
Gap (in.)	Width (in.)
0.005	0.0055
0.01	0.0075
0.015	0.0083
0.02	0.0086
0.025	0.0087
∞	0.0088

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

Circuit protects system from overheating

Kerry Lacanette, Maxim Integrated Products, Sunnyvale, CA

Figure 1



This circuit provides fan control and overtemperature protection for systems and high-power digital ICs.

The two-chip circuit in **Figure 1** provides fan control and overtemperature warning and shutdown signals to protect systems from excessive heat. The circuit monitors the temperature of the pc board and the die temperature of a CPU, an FPGA, or another IC with an on-chip temperature-sensing transistor. IC_1 is a temperature detector and fan driver for cooling fans with nominal operation of 250 mA. At low temperatures, the cooling fan is off, minimizing noise and fan wear. When the system temperature increases to more than 45°C, IC_1's factory-programmed temperature comparator causes the $\overline{\text{FAN OUT}}$ fan-drive pin to go active, pulling the fan's lower power-supply terminal to ground, thus providing low-side drive to the fan. The fan can accommodate supply voltages as high as 24V. After the fan activates, the system temperature normally either continues to rise at a slower rate or drops somewhat. If the temperature drops far enough, the fan turns off. To avoid causing the fan to continuously turn on and off, IC_1 provides hysteresis of 1, 4, or 8°C, which you can set by the HYST pin.

If a thermal problem, such as excessive power dissipation or blocked ventilation paths, exists, system temperature may continue to increase. IC_1 has two outputs that detect this condition. $\overline{\text{WARN}}$ becomes active when the temperature exceeds 60°C, and the $\overline{\text{OT}}$ output becomes active when the temperature exceeds 75°C. You can use $\overline{\text{OT}}$ as a system-shutdown signal. While IC_1 monitors board temperature, IC_2 monitors the die temperature of another chip—typically, a CPU, an FPGA, or an ASIC. The target IC must have a small-signal p-n junction, usually a substrate pnp, for temperature measurement. IC_2 forces current through sense junction, measures the resulting voltage, and calculates the temperature of the junction. IC_2 then compares this temperature with a preset threshold. When the junction temperature exceeds the threshold, 125°C in this case, IC_2's output pin goes active; you can use it to shut down the system.

The open-drain shutdown outputs of IC_1 and IC_2 connect to a common pullup resistor and to the power supply's shutdown terminal. If either the board temperature or the chip temperature exceeds the maximum safe rating, the system shuts down before damage can occur. IC_1 should be in a location that allows it to measure the temperature of interest. Depending on the system, this location could be near a "hot spot" or in the cooling fan's airflow path. The traces between IC_2 and the remote-sensing junction should be reasonably short and separated from high-speed data traces.

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

Network imitates thermocouples

Abel Raynus, Armatron International, Melrose, MA

THERMOCOUPLES FIND widespread use for temperature measurement in systems. During system design or testing, you must observe the system's response at different temperatures. However, it's inconvenient to heat a thermocouple every time you need to check a system's performance. You can use the simple trick of touching the thermocouple with a hot soldering iron, but this method provides only rough, approximate results. The simple network in **Figure 1** allows you to set a number of voltages equal to the thermocouples' outputs at given temperatures. A thermocouple's output is relatively in the tens of millivolts. The low level entails the use of a high-gain amplifier as a signal conditioner. These high-gain amplifiers are sensitive to noise. Susceptibility to noise is not a problem when the amplifier connects to a thermocouple, thanks to the thermocouple's output impedance of approximately 1Ω. But during system testing, substituting a high-impedance source for the thermocouple can result in noise pickup that can drive the amplifier into saturation. Hence, the output impedance of the thermocouple imitator must be low, and the output must connect to ground between tests.

TABLE 1—CORRECTION FOR COLD-JUNCTION TEMPERATURE

Temperature (°F)	Voltage at 32°F (mV)	Voltage at 100°F (mV)
550	11.71	10.19
855	18.82	17.3
900	19.89	18.37
1070	23.91	22.39

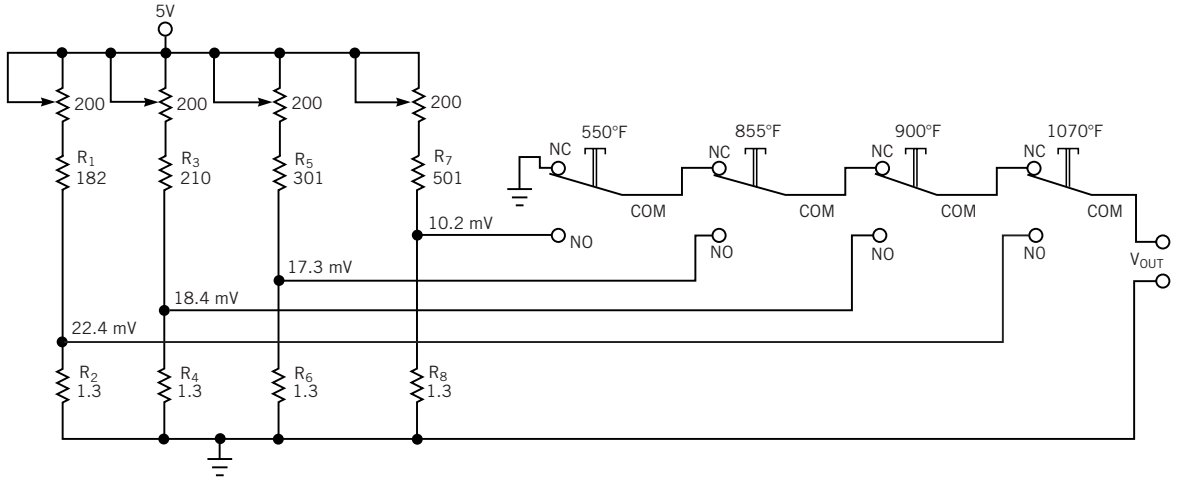
Figure 1 shows the thermocouple imitator for four temperatures. To obtain low output impedance, you set R_2, R_4, R_6, and R_8 to 1.3Ω. To satisfy the between-tests grounding requirement, the momentary SPDT key switches connect to the chain in a way that, when you press no switch, the output connects to ground. By pressing a switch, you obtain one of the predetermined voltages from dividers R_1/R_2, R_3/R_4, R_5/R_6, or R_7/R_8 at the output. Assume, for example, imitator-equivalent temperatures of 550, 855, 900, and 1070°F. You can find the voltages from a Chromel-Alumel thermocouple (**Reference 1**). But keep in mind that the voltages in the book apply only to a cold-junction temperature of 32°F. The working temperatures are always different, so you must recalculate the voltages. Assuming that the ambient temperature is approximately 100°F, you can find the thermocouples' output voltages by subtracting 1.52 mV from the 32°F value (**Table 1**). You can calculate the values of the divider resistors using the following equation: $R_U = R_L(V_{CC}/V_{OUT} - 1)$, where R_U is the upper divider resistor, R_L is the lower divider resistor, V_{CC} is the power-supply voltage, and V_{OUT} is the output voltage. To make the output-voltage adjustment easier, the upper divider resistor consists of a 200Ω potentiometer in series with a fixed resistor.

REFERENCES

1. *The Temperature Handbook*, Omega Engineering Inc, 2000.

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

Figure 1

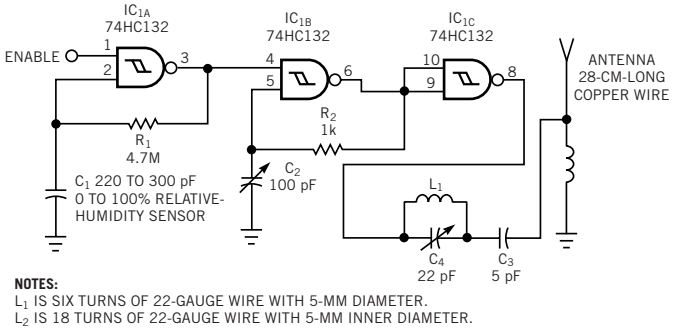


This network allows you to emulate thermocouple outputs at various temperatures.

Low-cost relative-humidity transmitter uses single logic IC

Shyam Tiwari, Sensors Technology Ltd, Gwalior, India

Figure 1



This percentage-relative-humidity transmitter uses 10- to 50-MHz, tunable RF, and 1- to 2-kHz on/off amplitude modulation.

The low-cost percentage-relative-humidity radio transmitter in **Figure 1** operates in a cold-storage warehouse for vegetable storage at temperatures of 1 to 5°C. It is generally difficult to collect such data from a low-temperature area with high humidity and low illumination. The transmitter design is simple: It uses a readily available, capacitor-type percentage-relative-humidity sensor for which the capacitor value increases with humidity. Generally, these sensors offer accuracies well within 5%. Humirel (www.humirel.com) relative-humidity sensors work well with this circuit; you can also use other types with low leakage resistance. The R_1C_1 product gives the time constant for the audible-modulating, 1- to 2-kHz signal oscillator, which you can gate to stop the communication. This oscillator starts the RF oscillator, which has a time constant, R_2C_2, equating to a 10- to 50-MHz RF band. The last inverter is a power driver for the tuned filter and antenna. The circuit requires a 3 to 5V battery. Two AAA cells can power it for approximately 15 days. If you need a high modulating frequency, then you can reduce R_1 to 1 MΩ, changing the modulating signal to the range of 10 to 20 kHz.

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

design ideas

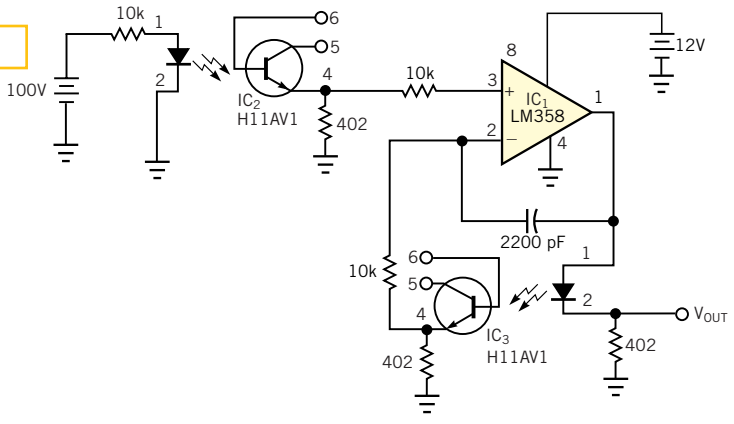
Edited by Bill Travis

Circuit compensates optocoupler temperature coefficient

J Michael Zias, Acme Electric Corp, Cuba, NY

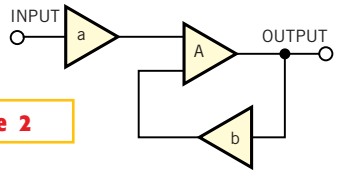
When using an optocoupler in a linear application, you should consider its gain drift with temperature. Traditional single- and dual-transistor-output devices have a notable gain drift with temperature. In recent years, some temperature-compensated optocouplers have appeared. However, another option is to use two optocouplers or a dual optocoupler with appropriate feedback to make the drift of one device cancel the drift of the other. The circuit in **Figure 1** accomplishes that task by using a differential amplifier with the drift treated as a common-mode signal. In operation, it is interesting to apply a dc signal to the input and use digital voltmeters to simultaneously monitor the output of each optocoupler and the differential amplifier. Apply a heat gun and observe the individual outputs change rapidly while the amplifier output moves much more slowly. This result occurs even with optocouplers from different manufacturers. With optocouplers of the same type, you can observe good drift cancellation. Parts from the same manufacturer and dual devices give outstanding results. You can use individual optocouplers instead of dual devices to meet safety-agency spacing requirements.

Figure 1



By using two optocouplers instead of one, you can cancel temperature-dependent gain drift.

Figure 2



Control-system feedback theory explains the operation of the circuit in Figure 1.

To examine the method in control-system terms, consider **Figure 2**, which shows one amplifier, a, in the forward path and another amplifier, b, in the feedback path. Also consider the following **equation**:

$$\text{GAIN} = a\frac{A}{1+Ab} = \frac{a}{b}\bullet\frac{1}{1+\frac{1}{Ab}},$$

where a/b is the ideal closed-loop gain and is multiplied by the loop-gain error term. Given that the error term is small (from the large gain A of the op amp), the gain of the system is seen as the ratio of the gains (current-transfer ratios) of the optocouplers. You can also easily find this same ratio by setting the voltages to the op-amp inputs equal. The input and output signals for this analysis are currents, which precision resistors translate to voltages. The optocouplers in this design are not particularly fast devices, so the phase delays could cause oscillation without a feedback capacitor. You choose its value empirically by applying a pulse at the input and observing the rise time and overshoot at the output.

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

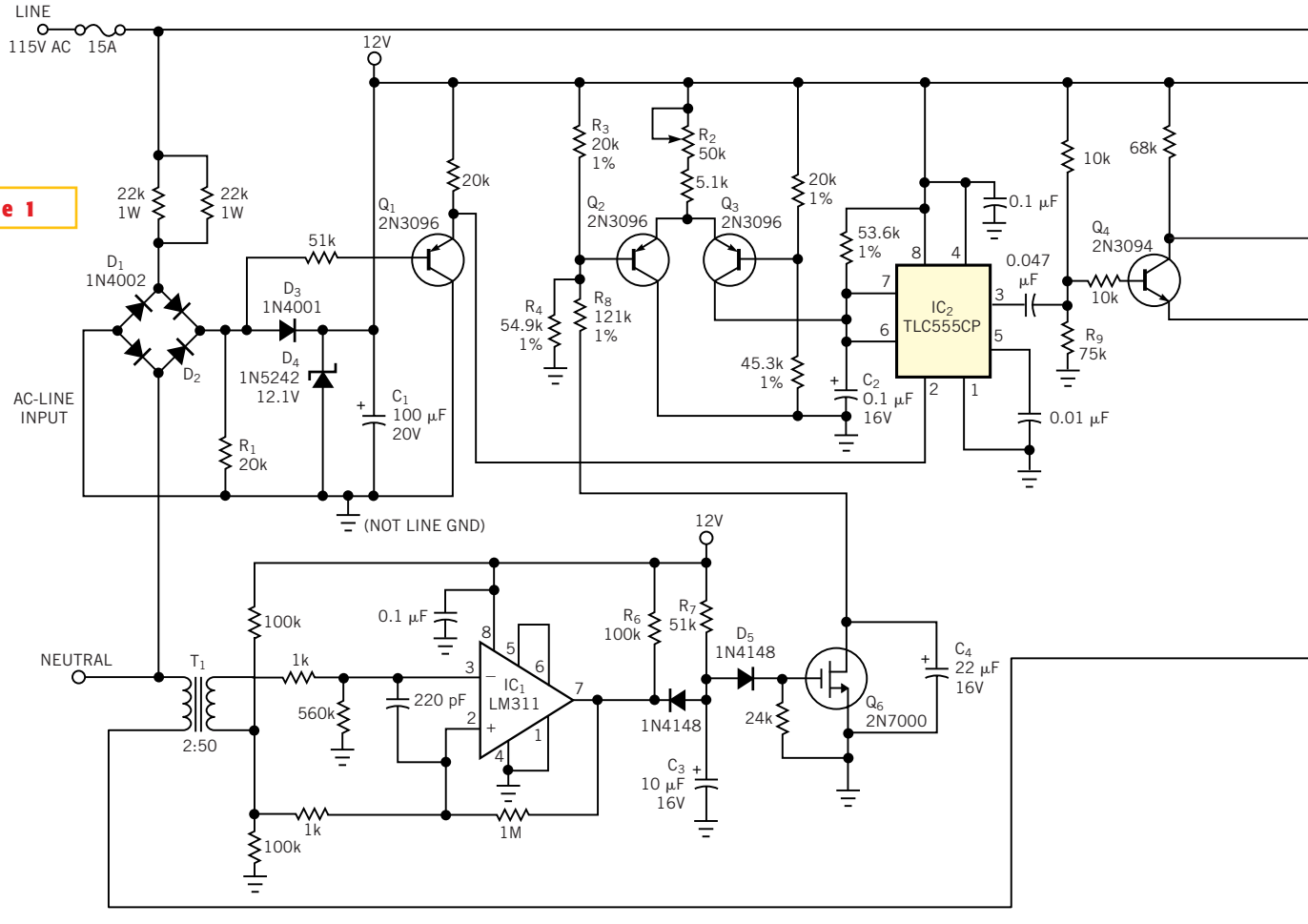
Soft-start controller is gentle on loads

Douglas Sudjian, Resonext Communications Inc, San Jose, CA

The control circuit in **Figure 1** senses a given load and automatically soft-starts the load by synchronously adjusting the power to that load. You can also manually adjust the power delivered to the load by controlling the phase angle of the line voltage across the load. The phase-angle adjustment for every ac half cycle covers 0 to 180°. When the isolation transformer, T_1, senses the load current in the ac ground return, IC_1's output changes state, driving the signal diode, D_5, into and out of conduction. R_6, R_7, and C_3 create a delay such that the voltage at Q_6's gate decays slowly to allow for the load's switch-closure noise or missed ac cycles. Once Q_6 turns off, the voltage at the base of Q_2 rises to a higher reference level, which voltage divider R_3 and R_4 sets. The bias current of transistor pair Q_2 and Q_3 slowly passes through Q_3 as the differential input voltage across Q_2 and Q_3 changes according to the time constant of R_8 and C_4. The additional current that Q_3 sources to C_2 increases the voltage rate of change at pins 6 and 7 of IC_2. IC_2, a TLC555CP, is a low-power timer configured as a monostable multivibrator.

In the monostable mode, the timer issues a positive pulse output every time a negative-going trigger pulse arrives at Pin 2 of IC_2. The output pulse width corre-

Figure 1



This soft-start circuit protects the load from large inrush currents.

sponds to the time it takes the voltage on capacitor C_2 to ramp from 0V to 2/3 V_{CC}. With a constant current essentially charging C_2, the charging is linear, and the output at IC_2's Pin 3 is proportional to the current set by R_2. The full-wave bridge, with D_3 and D_4 and filter capacitor C_1, forms a dc power supply for the timer/controller. The common cathode node for D_1 and D_2 pulls to ground via R_1 every time the line voltage approaches 0V. Q_1 turns on and supplies a negative-going trigger to Pin 2 of

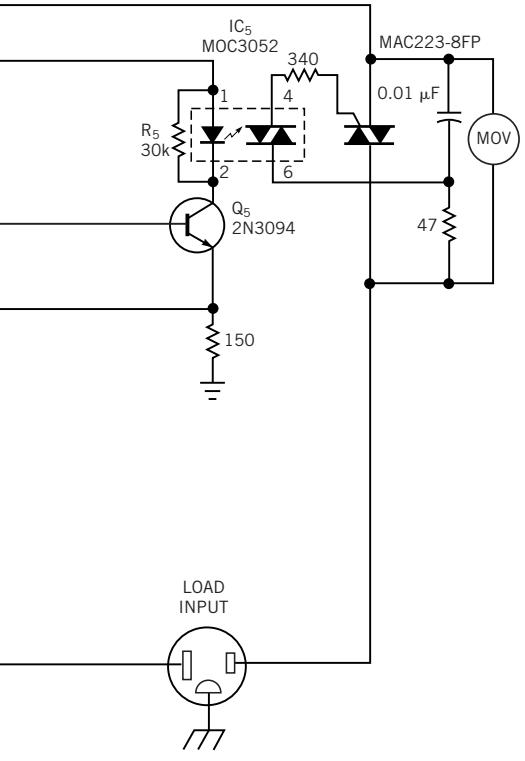
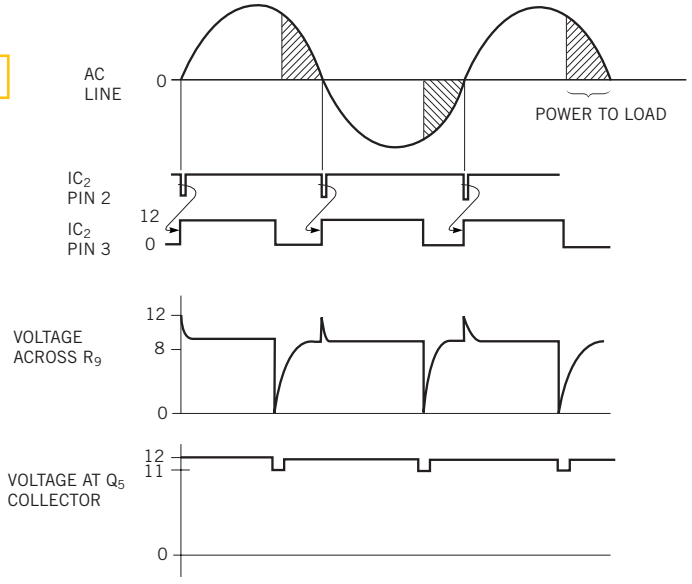


Figure 2



The circuit in Figure 1 provides soft-starting by adjusting the phase angle of the power applied to the load.

IC_2. This pulse uses its negative edge to provide a minimum pulse width of 200 µsec to the base of Q_4. The feedback pair Q_4 and Q_5 provides signal inversion and limits the current drawn from the 12V supply rail through IC_3. When sufficient LED current develops, the MOC3052 triac driver latches on and generates a gate current in the power triac, triggering it into the conducting state. Once the power triac latches on, the triac driver enters its off state, even if the LED current still exists.

The power triac's gate voltage falls below the optocoupler's threshold and cannot hold the optocoupler. The longer the phase delay from the zero-crossing trigger, the smaller the conduction angle and power delivered to the load. R_5 facilitates on-off switching of the triac-driver LED by providing a path for leakage currents. Potentiometer R_2 provides variable power to the load (to provide motor-speed control, for example). R_2 varies the dc-source current that charges C_2 every ac half-cycle. Note that the signal ground with respect to earth ground is floating; you must not tie these grounds together. The design in **Figure 1** has successfully controlled fans and high-amperage universal motors (100 mA to 11A). One example is a router for woodworking. By soft-starting these high-torque motors, the reaction torque (to the input current) that the user feels disappears. Moreover, other soft-start designs need two switches. The design in **Figure 1** needs only one on-off switch (located at the load). Thus, less danger exists for incurring an accidental starting condition. **Figure 2** shows some of the waveforms associated with the circuit in **Figure 1**. T_1 is a signal transformer that you can modify by wrapping two turns of 14-gauge wire around the bobbin to act as the primary winding.

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

Method offers fail-safe variable-reluctance sensors

Phil Levya, Maxim Integrated Products, Sunnyvale, CA

Variable-reluctance sensors are preferred for industrial and automotive environments, because they sustain mechanical vibration and operation to 300°C. In most applications, they sense a steel target that is part of a rotating assembly. Because the unprocessed signal amplitude is proportional to target speed, a sensor whose signal-processing circuitry is designed for high speed ceases to function at some lower rate of rotation. Hall-effect sensors are preferable for speeds of several pulses per second, but they require the attachment of a magnet to the rotating assembly. They're thus prone to failure when the magnet is broken or damaged. Neither variable-reluctance nor Hall-effect sensors offers fail-safe detection of the processed signal in the event of failure in the cable or sensor. The circuit in **Figure 1** is a fail-safe variable-reluctance sensor for low- to medium-speed operation.

The circuit comprises L_1; R_1; and a quad RS-422/RS-485 receiver, IC_1. It provides the complementary, independent output signals V_{OUT} and $\overline{V_{OUT}}$. **Table 1** lists the resulting fail-safe modes. The supply voltage can be 10V, 12V, or the control system's 24V-dc source. Coil L_1 consists of 2600 turns of #32 magnet wire wound on a 0.8-in. steel bar of 0.2-in. diameter, with 0.125 in. protruding from the sensor face. A magnet attached to the back of the steel bar supplies the necessary magnetic flux. The rotating target causes a change in reluctance and, hence, a change in the amount of magnetic flux conducted. This change produces a corresponding change in the current induced in L_1. R_1 converts the L_1 current to a time-varying voltage. This voltage goes to the inputs of IC_1, whose input-voltage range of ±25V, input threshold of ±0.2V, and typical input hysteresis of 45 mV enable the VR sensor to operate at low speeds.

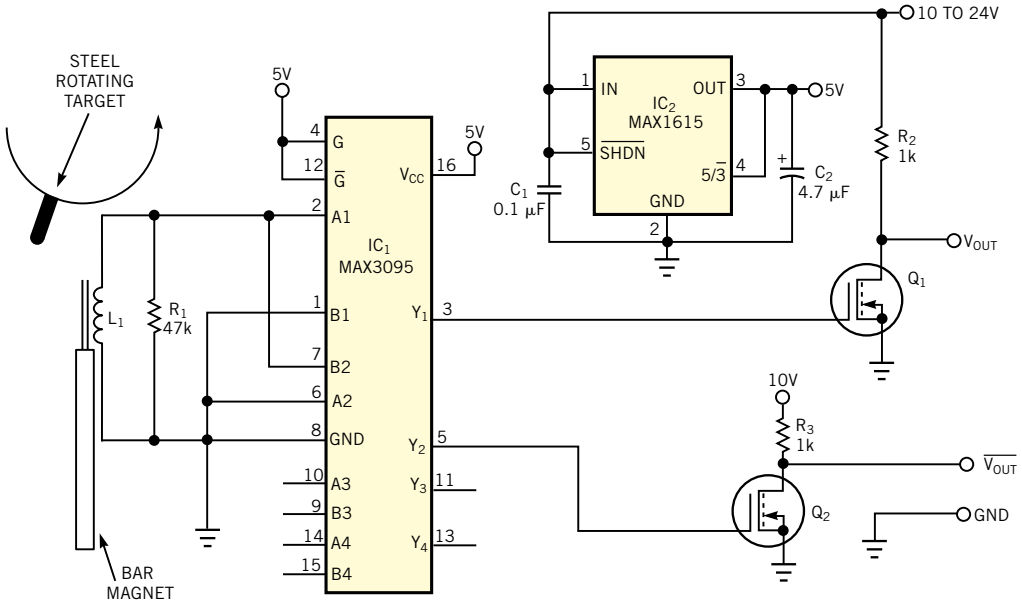
The separate, complementary outputs come from separate, ESD-protected in-

TABLE 1–FAIL-SAFE MODES (TWO CYCLES OF V_{OUT} OR $\overline{V_{OUT}}$)

(V_{OUT}, $\overline{V_{OUT}}$)	Mode
(1,0) then (0,1) or (0,1) then (1,0)	Normal mode, both pulses valid
(1,0) then (0,0) or (0,0) then (1,0)	Failure, valid $\overline{V_{OUT}}$ pulse, $\overline{V_{OUT}}$ failure, cable failure, or partial sensor failure*
(0,1) then (0,0) or (0,0) then (0,1)	Failure, valid $\overline{V_{OUT}}$ pulse, V_{OUT} failure, cable failure, or partial sensor failure*
Always (1,1)	Short-circuited cables or failure in IC_1
Always (0,0)	Severed cables, failure in IC_1 or failure in Q_1 and Q_2

*System remains functional in failure modes.

Figure 1



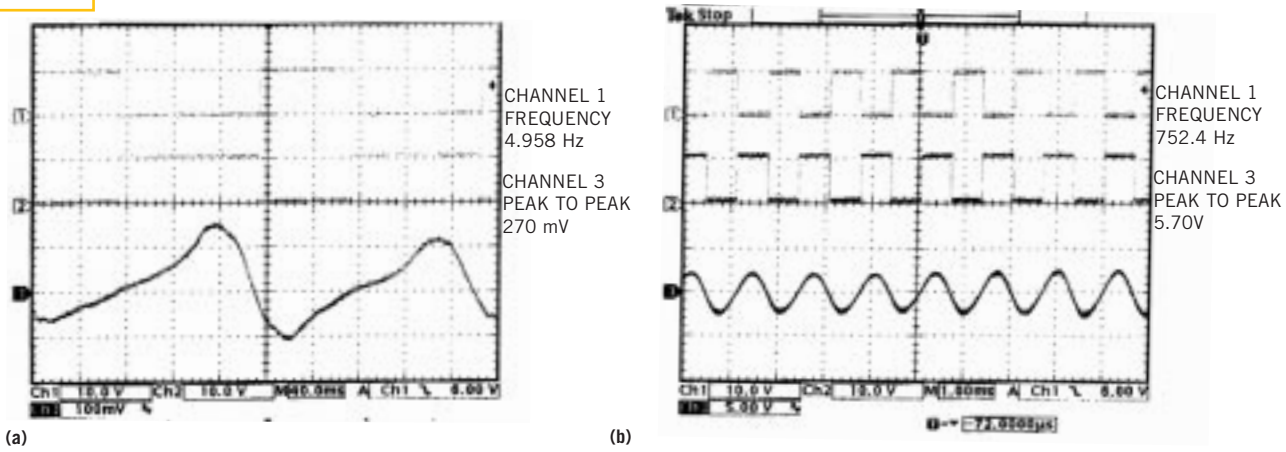
NOTE: Q_1, Q_2 = FAIRCHILD FDV303N.

This circuit provides a fail-safe, low- to medium-speed variable-reluctance sensor.

puts. IC_1's outputs Y_1 and Y_2 can source as much as 10 mA. They alternately switch the logic-level, n-channel MOSFETs Q_1 and Q_2, which in turn provide V_{OUT} and $\overline{V_{OUT}}$. A low-dropout regulator, IC_2, provides the 5V power source for IC_1. **Figure 2** illustrates low- (**Figure 2a**) and medium-speed (**Figure 2b**) operation for the sensor. For 5V-supply applications in which you can locate a microcontroller close to the sensor, you need only L_1, R_1, and IC_1 for a direct interface. For 3V applications, replace IC_1 with a MAX3096 IC.

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

Figure 2



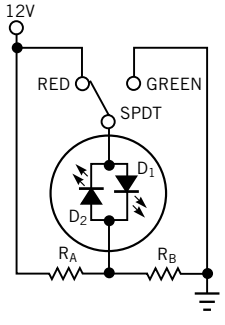
These waveforms represent operation at 4.9 Hz at 2.4 revolutions/sec (a) and 752.4 Hz at 376.2 revolutions/sec (b). Channel 1 is V_{OUT}, Channel 2 is $\overline{V_{OUT}}$, and Channel 3 is the voltage across R_1.

Circuit efficiently switches bipolar LED

Spehro Pefhany, Trexon Inc, Toronto, ON, Canada

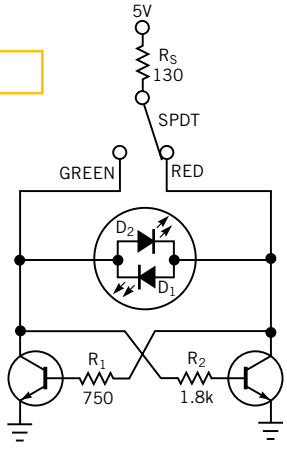
The circuit in **Figure 1** represents one method to switch a bipolar, two-color LED using an SPDT mechanical switch or relay. This circuit wastes power and does not work properly if the power-supply voltage is not substantially more than the sum of the LEDs' forward voltages. The circuit is, therefore, marginal, to the point of being unusable, with a 5V supply and a red or green LED, which typically has a total forward voltage of 4V. You can use a circuit resembling a flip-flop (**Figure 2**) that doesn't suffer the disadvantages of the circuit in **Figure 1**. It adds only one $V_{CE(SAT)}$ voltage to the V_F of each LED, so plenty of headroom exists with a 5V supply and a series resistor to control the LEDs' current. The circuit in **Figure 2** costs less than a dime for the parts, which include three resistors and two inexpensive, general-purpose npn transistors, such as the 2N4401 or the C8050. In this

Figure 1



This switching circuit wastes power and does not work with low supply voltages.

Figure 2



In this "flip-flop" switch, the only losses come from the $V_{CE(SAT)}$ and the base currents of the transistors.

example, D_1 is red (V_{F1}=1.6V), and D_2 is green (V_{F2}=2.4V). Based on D_2, the green LED, you can calculate that R_S=(5V−2.4V−0.1V)/0.02A=125Ω (use 130Ω for 19 mA).

As a result, using a single resistor, D_1 has a current of 25 mA. If it is desirable to have equal or arbitrarily different currents, you can insert an additional resistor in one leg of the switch to increase the effective R_S for that switch position. The base drive is a function of the V_F of the driven LED, so you can calculate the base resistors, using a forced beta of 20, as follows:

R_1=20(V_{F1}−0.7V)/I_{LED1}=720Ω (use 750Ω).

R_2=20(V_{F2}−0.7V)/I_{LED2}=1.8 kΩ.

The base drive reduces the actual LED current by 5%, which is visually negligible. As a bonus, the circuit does not introduce any switching glitches into the power supply. The circuit requires only two connections, rendering it ideal for front-panel use. Because the 130Ω resistor is in series with the power supply, any part of the circuit beyond R_S can short to ground without causing damage.

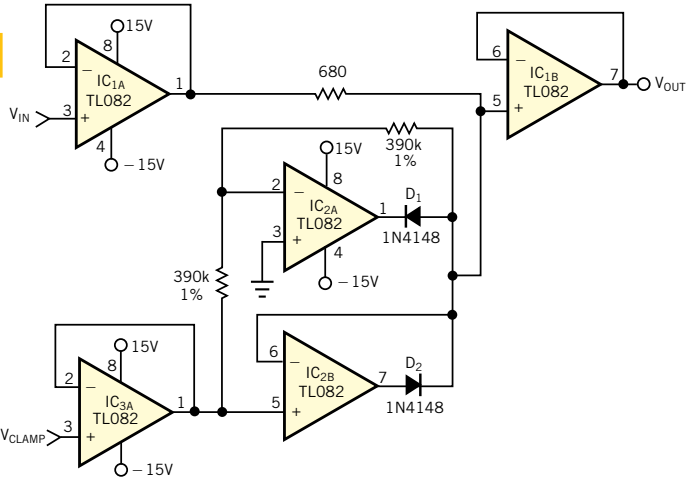
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Circuit forms adjustable bipolar clamp

Pautasso Luciano, Nichelino, Italy

THE EASY WAY to clamp a signal to a given value is to use two zener diodes, connected back-to-back. This method has several disadvantages. The accuracy of the clamping depends on the tolerance of the zener diodes, and the clamping is not adjustable, except by changing diodes. The circuit in **Figure 1** is a bipolar clamper with a range of ±1 to ±10V, with the clamping level a function of the input V_{CLAMP}. IC_{1A}, IC_{1B}, and IC_{3A} are unity-gain buffers. IC_{2A} is a positive clamper, and IC_{2B} is a negative clamper. **Figure 2** shows the transfer function, with V_{CLAMP} set at −5V. You can change V_{CLAMP} over the range of −1 to −10V and thereby change the clamping level. If V_{IN} is within $-V_{CLAMP}$ to $+V_{CLAMP}$, then V_{OUT}=V_{IN}. If V_{IN} exceeds V_{CLAMP}, then V_{OUT}=V_{CLAMP}. To explain how the circuit works, assume four cases, with four values of V_{IN}. Basically, the circuit works in two modes: the linear mode, in which diodes D_1 and D_2 are open switches, and the clamped mode, in which the diodes are closed switches. **Table 1** gives results for the four cases. In Case A, the input is 7V, V_{CLAMP} is −5V, D_1 conducts, and D_2 is an open switch. The feedback loop around IC_{2A} regulates the anode of D_1 to 5V and the output of IC_{2A} to 4.4V. In cases B and C, both diodes are open switches. In Case D, D_2 conducts, and D_1 is an open switch.

Figure 1



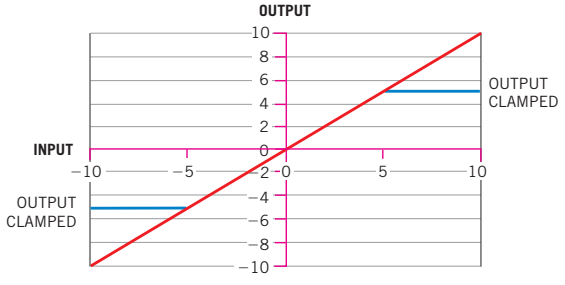
This circuit provides adjustable clamping over the range of ±1 to ±10V.

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

TABLE 1–RESULTS FOR CLAMPED AND LINEAR MODES

Case	V_{IN} (V)	V_{OUT} (V)	Mode
A	7	5	Clamped
B	3	3	Linear
C	−3	−3	Linear
D	−7	−5	Clamped

Figure 2



With V_{CLAMP} set at −5V, the output clamps firmly at ±5V.

Analog switch expands I²C interface

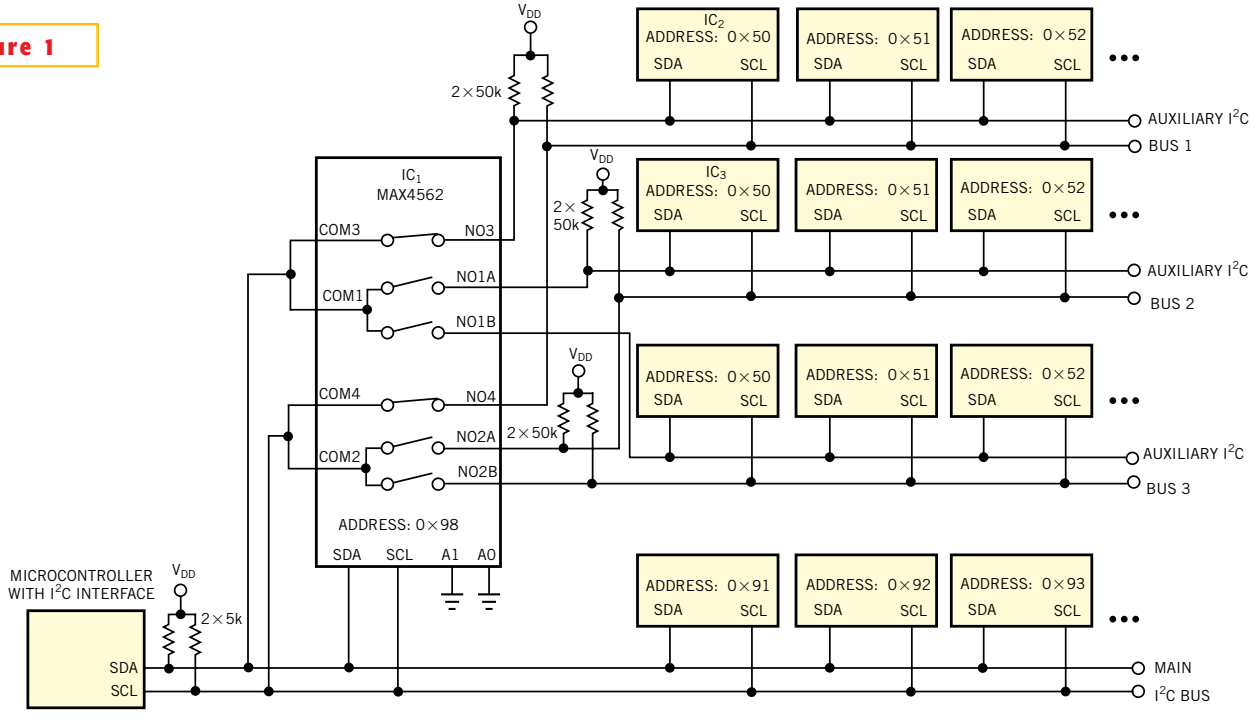
Luca Vassalli, Maxim Integrated Products, Sunnyvale, CA

Perhaps the most effective way to gain board space and increase component density is to minimize wiring on the board. A widely used architecture that allows such miniaturization is the I²C bus. Comprising only a bidirectional data line, SDA, and a clock line, SCL, this bus requires no chip selects or other additional connections. Microcontrollers from Philips, Microchip, and other manufacturers include dedicated I²C interfaces, but you can also implement the interface in software. To complete this task, you associate a 7-bit address with each master or slave transceiver and factory- or pin-program the device with two to four address options. An increasing number of slaves now include the I²C interface, but some of their 128 address locations are reserved for special functions, so not all locations are available to a designer. Yet, two or more devices could have the same address in some application. In **Figure 1**, analog switch IC_1, which is I²C-controlled, connects auxiliary branches that contain devices with the same address to the main I²C bus. IC_2 and IC_3, for example, have the same address but are located on different auxiliary buses.

The arrangement in **Figure 1** prevents the master from addressing multiple slaves at the same time. If that situation occurs, the data becomes corrupted during a master-read protocol, and all slaves may not receive data during a master-write protocol. The analog switch accepts bidirectional signals as required for the SDA line. The switch has low on-state resistance, adds almost no leakage on the lines, and provides four selectable slave addresses. You simultaneously control the switches by using the simple Send-Byte protocol (address plus 8-bit command). You can switch the three auxiliary buses on the fly. Power-up sets the switches to soft mode, an off state with 12-msec switching time. Then, a command byte of 0b11000000 sets the switches to hard mode (400-nsec switching time). Subsequent commands select the desired auxiliary bus. Command 0b1000011, for example, selects auxiliary bus 1. The main I²C bus includes necessary pullup resistors, and the auxiliary buses include weaker pullups that ensure a high state when you deselect the bus. The circuit in **Figure 1** allows you to add three times more devices on the bus. For a wider selection, you can replace the MAX4562 with a MAX4572, whose 14 switches allow you add as many as seven auxiliary buses.

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

Figure 1



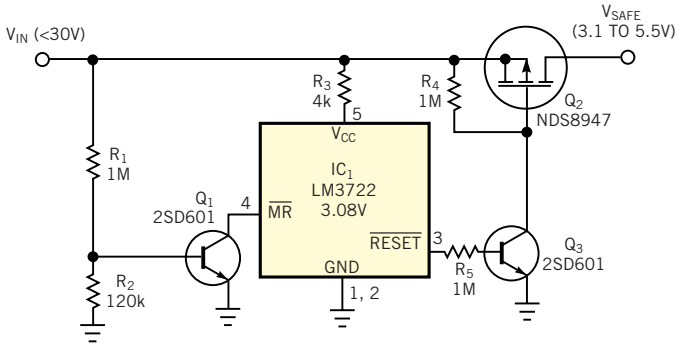
This I²C-controlled analog switch expands by three the number of devices connected to the bus.

Circuit safely applies power to ICs

Clayton Grantham, National Semiconductor, Tucson, AZ

Supervisory circuits normally monitor a microprocessor's supply voltage, asserting reset to the IC during power-up, power-down, and brownout. In this way, the circuit ensures that the supply voltage is stable before the microprocessor boots, thus preventing code-execution errors. Many analog and digital ICs also need a well-behaved start-up of their supply to avoid latch-up and logic-state errors. In addition to low-supply conditions, low-voltage CMOS circuits need overvoltage protection from any supply runaway. The additional components in **Figure 1** extend IC_1's supervisory functions to connect V_{IN} to V_{SAFE} only when V_{IN} is within set limits. This function protects circuitry at the V_{SAFE} terminal from power-up transients and overvoltage damage. As a supervisory circuit, IC_1 asserts a reset signal that is delayed by more than 100 msec whenever V_{IN} decreases below the precisely trimmed reset threshold. You can custom-select the reset threshold from 2.32 to 4.63V. You can also use a manual input, $\overline{MR}$, to assert the reset signal.

Figure 1



This LM3722 configuration connects only safe voltages to sensitive ICs.

This application uses IC_1's delayed reset signal to control switch Q_2. The delay ensures that V_{IN} is stable before application to V_{SAFE}. Q_3 inverts and isolates IC_1's reset signal to control the gate of Q_2. R_4 is Q_2's pullup resistor; R_5 limits Q_3's base current. Using Q_1 as an inexpensive 0.6V switch, resistor dividers R_1 and R_2 set the overvoltage threshold according to the equation $V_{OV} = V_{BE1}(R_1+R_2)/R_2$. An internal 22-kΩ resistor at IC_1's $\overline{MR}$ input provides Q_2's pullup. Typical V_{BE1} accuracy and temperature-coefficient errors are ±10% and −2 mV/°C, respectively.

TABLE 1–V_{SAFE} HYSTERESIS OVER TEMPERATURE

V_{SAFE}		0°C	25°C	50°C
On (V)	V_{IN} increasing	3.2	3.2	3.2
Off (V)	V_{IN} increasing	6.1	5.5	4.9
On (V)	V_{IN} decreasing	6	5.4	4.8
On (V)	V_{IN} decreasing	3.1	3.1	3.1

Adjustment of R_2 for an exact overvoltage value nullifies V_{BE1}'s accuracy error. **Table 1** shows typical setpoints over temperature. If you need further error reduction, you could exchange Q_2 for a comparator and voltage reference. For V_{IN} within the set limits, 3.1 to 5.5V, the circuit draws only 16 µA. A total of 5 µA flows into both the R_1 and R_4 nodes, and 6 µA flows into R_3's node. R_3 protects IC_1 by providing current limiting of less than 6 mA) for high voltages at V_{IN}. The typical IC_1 current of 6 µA through R_3 increases the undervoltage setpoint by 24 mV.

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

Simple circuit forms peak/clipping indicator

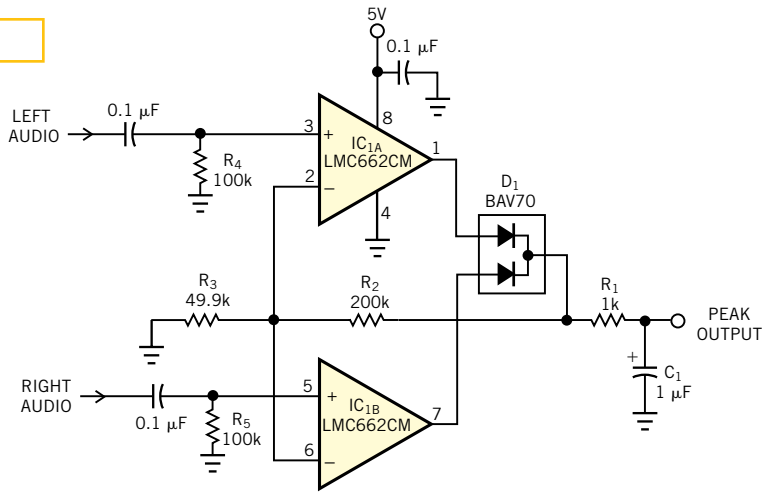
Steven Hageman, Agilent Technologies, Santa Rosa, CA

The simple peak detector in **Figure 1** is the result of a need for a single-5V-supply, level/clipping indicator for a multimedia-PC sound system. The design is unique in that it detects both stereo channels on a single peak-hold capacitor. All the adjustments in the circuit simultaneously apply to both left and right stereo inputs. The output is suitable for driving a bar-graph display or for analog-to-digital conversion and display with a microprocessor. The circuit operates as a dual positive-peak-detector circuit. The dual diode, D_1, serves to allow positive peaks to pass while disconnecting the op amp from the hold capacitor, C_1, on negative peaks. Also, because the diodes have an OR connection, the circuit detects only the larger peak from the left or right stereo input. The values shown in **Figure 1** are for standard 200-

mV-rms line-input levels, such as those you'd find on a PC's sound-card line input. Your personal preference or exact needs might require other performance parameters, and you can easily adjust these values.

The gain for both stereo channels is equal to $1+R_2/R_3$. The circuit as shown has a gain of 5. For a full-scale 200-mV-rms input, this gain produces an output of approximately 1.4V. This value is convenient for this application, which uses three green LEDs, two yellow LEDs, and one red LED to show the relative peak levels of the stereo channels. Nominal, full-scale line input of 200 mV rms lights two of the green LEDs. "Attack time" is the time it takes the peak detector to respond to 69% of an input-signal peak, or one time constant. The time constant R_1C_1 sets the attack time. In this circuit, the attack time is 1 msec. The decay time is the time it takes the peak to decay to 31% of its original value, or one time constant. This time equals $(R_2+R_3)C_1$ (assuming that R_1 is negligibly small compared with R_2+R_3). The decay time in this case is 250 msec, because that value produces a pleasing-looking bargraph display. Some applications may need different response rates; you can easily obtain them by following the design equations above.

Figure 1



This simple circuit provides peak detection and clipping indication for a PC's stereo channels.

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

design ideas

Edited by Bill Travis

Software makes full use of 8051's interrupt system

Deng Yong, Shanghai Jiaotong University, China

THE PROGRAM in **Listing 1** uses a pseudo-RETI instruction to provide a five-priority-level interrupt system for the 8051P microcontroller. The interrupt-priority order, from high to low, is INT0 IT0 INT1 IT1 INTP. Before the pseudo-RETI instruction arrives in the IT0 or IT1 interrupt-service routine, the address of the first instruction, which is after the pseudo-RETI instruction, goes back into the stack. The internal, nonaddressable flip-flop associated with IT0 or IT1 clears to acknowledge a higher interrupt after execution of the pseudo-RETI instruction, while the IT0 or IT1 interrupt-service routine executes continuously until the RETI instruction arrives. Hardware circuits can exchange the INT1 and INT2 interrupts, and software can set the IT1 and IT2 interrupts.

You can download **Listing 1** from the Web version of this article at www.ednmag.com.

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

LISTING 1—FIVE-PRIORITY-LEVEL INTERRUPT SYSTEM FOR 8051P

```
        ORG   0000H
        LJMP  START
        ORG   0003H
        LJMP  INT0
        ORG   000BH
        LJMP  IT0
        ORG   0013H
        LJMP  INT1
        ORG   001BH
        LJMP  IT1
        ORG   0023H
        LJMP  INTP
START:  MOV   SP, #80H
        MOV   IE, #9FH
        MOV   IP, #03H
        SJMP  $
INT0:                                ;INT0 interrupt service program
        .
        RETI
IT0:    PUSH  DPL                    ; IT0 interrupt service program
        PUSH  DPH
        MOV   DPTR, #True_IT0
        PUSH  DPL
        PUSH  DPH
        ANL   IP, #10000001B
        RETI                         ;pseudo-RETI
True_IT0:
        .
        POP   DPH
        POP   DPL
        ORL   IP, #00011100B
        RETI
INT1:                                ; INT1 interrupt service program
        .
        RETI
IT1:    PUSH  DPL                    ;IT1 interrupt service program
        PUSH  DPH
        MOV   DPTR, #True_IT1
        PUSH  DPL
        PUSH  DPH
        CLR   PS                     ;"pseudo-RETI"
        RETI
True_IT1:
        .
        POP   DPH
        POP   DPL
        SETB  PS
        RETI
INTP:   PUSH  DPL                    ;Serial port interrupt service program
        PUSH  DPH
        MOV   DPTR, #SPINT
        PUSH  DPL
        PUSH  DPH
        RETI
SPINT:
        .
        POP   DPH
        POP   DSL
        RETI
*****************************************************************
```

Improve FET-based gain control

Ron Mancini, Texas Instruments, rmancini@ti.com

One problem with standard FET gain-control circuits is increased noise when you use the FET as a part of a resistive attenuator in series with an op amp. This configuration attenuates the signal before amplification; hence, it requires much higher gain bandwidth and better noise performance from the op amp. When you substitute the FET for the gain-setting resistor in a noninverting op-amp circuit, distortion limits the circuit configuration to applications in which the input voltage is less than a few hundred millivolts. The FET imposes this limitation, because the channel-depletion layer is a function of V_{DG} and V_{GS}. The improved circuit in **Figure 1** uses the FET as part of the feedback loop. The voltage across the FET is limited in this application, and the noise performance is good. An added bonus is improved linearity performance. The transfer function for the improved circuit is as follows (**Reference 1**):

$$-\frac{V_{OUT}}{V_{IN}} = -G = \frac{R_2 + R_3 + \frac{R_2 R_3}{R_4}}{R_1}.$$

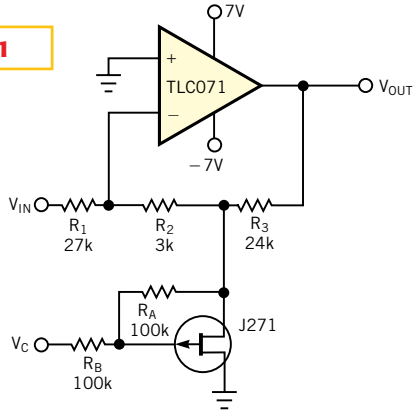
When $R_2+R_3=R_1$ and $R_4=R_{DS}$ (FET drain-source resistance), the transfer function reduces to $-G=1+R_2||R_3/R_{DS}$. The minimum drain-source resistance for the FET on hand, J271, is 76Ω at $V_{GS}=0V$. The actual V_{DS} at the inception of distortion varies with each FET, but keeping V_{DS} lower than 200 mV usually prevents distortion. In the design in **Figure 1**, the FET drain-source voltage is limited to approximately 100 mV to prevent distortion. The divider action between R_3 and R_{DS} creates V_{DS} from the output voltage, according to the following equation:

$$V_{DS} = V_{OUT}\frac{R_{DS}}{R_{DS}+R_3} =$$

$$0.1V = 5\frac{0.076}{0.076+R_3}.$$

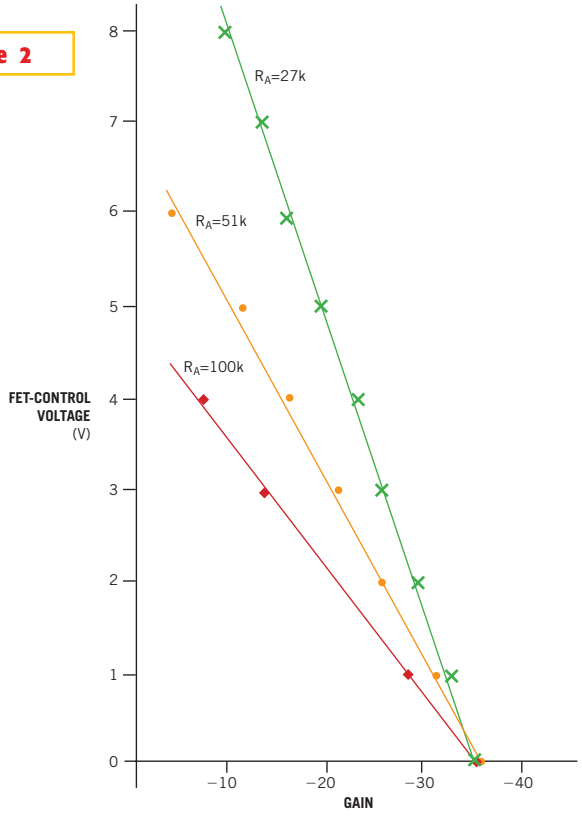
You can calculate R_3 as 24.5 kΩ and select 24 kΩ. The parallel value of R_2 and R_3 determine the maximum circuit gain. Selecting R_2 as 3 kΩ yields R_1 equal to 27 kΩ and a maximum gain of −37. The measured gain at $V_C=V_{GS}=0V$ is −36.1, which correlates well with the calculated value. R_A and R_B are feedback resistors that linearize the FET's V_{GS} versus R_{DS} transfer function. You can normally obtain adequate linearization with equal-value resistors, but you can also control the slope of the transfer function by setting the resistor ratio. The graph in **Figure 2** shows that R_A modifies the transfer function and linear control-voltage range (V_{GS}). The p-channel FET, J271, requires a positive control voltage, but you can use a negative control voltage with an equivalent n-channel FET, such as the J210. The circuit is versatile and provides low distortion, wide range, good linearity, and low cost. The TLC071 op amp has low input-bias currents and has provisions for input offset-voltage correction.

Figure 1



The drain-source resistance of the FET controls the gain of the op-amp stage.

Figure 2



The ratio R_A/R_B in Figure 1 controls the slope of the gain-control transfer function.

Reference

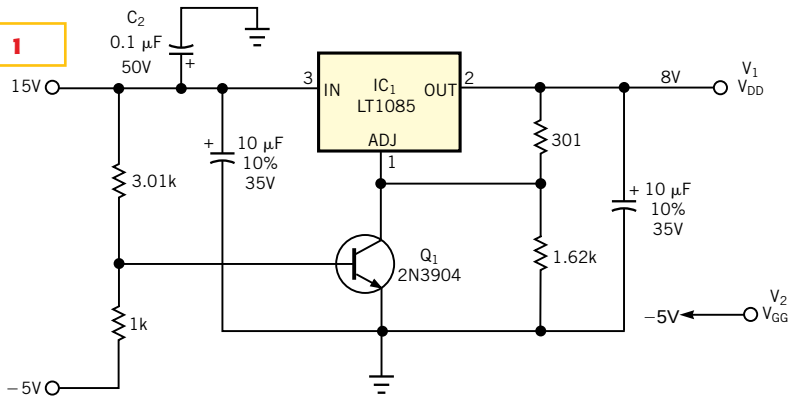
1. Mancini, Ron, "Op amps for everyone," Texas Instruments, September 2000, pg 3.

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

Circuit improves on bias for GaAs FETs

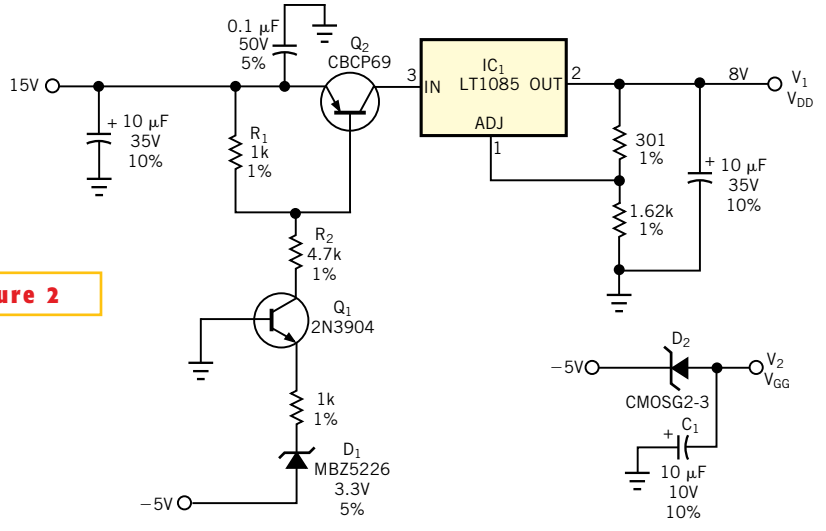
Tom Roberts, Anritsu Co, Morgan Hill, CA

Figure 1



In disabled mode, this circuit supplies a potentially damaging 1.25V to the drain of a GaAs FET or a MMIC.

Figure 2



This circuit provides safe power-up and power-down sequencing for sensitive GaAs FETs and MMICs.

It's important to properly sequence the bias applied to an RF/microwave GaAs FET or a MMIC (monolithic-microwave-IC) amplifier. These devices are extremely sensitive to drain and gate voltage levels as well as to the order in which these biases turn on and off. A GaAs-FET amplifier that uses two bias voltages—a negative supply, V_{GG}, on its gate and a positive supply, V_{DD}, on its drain—requires that V_{GG} be present before the application of V_{DD}. When powering down the amplifier, V_{DD} must go to 0V before V_{GG} changes from its negative value to 0V. **Figure 1** shows a commonly used disable circuit found in many voltage-regulator data sheets. The circuit uses a 2N3904 switching transistor to pull the ADJ pin to ground to disable the voltage regulator. The circuit does not set the output of the regulator to 0V but instead sets the output to the regulator's reference voltage, 1.25V. The condition in which a GaAs FET or MMIC has 0V on the gate and 1.25 on the drain can result in damage to the device. For example, M/A-Com's MAAM26100-PI MMIC power amplifier requires 8V for V_{DD} and −5V for V_{GG}. With 1.25V on V_{DD} and 0V on V_{GG}, the MMIC draws approximately three times its nominal drain current, sufficient to cause destructive failure. **Figure 2** shows an improved circuit for the adjustable regulator.

A medium-power pass transistor, Q_2, a Central Semiconductor CBCP69, connects to the input of the voltage regulator to disable the regulated output voltage. In disabled mode, the voltage at the regulator's output is 0V. In enabled mode, Q_1 saturates and activates a voltage divider comprising R_1 and R_2. Q_2 saturates, and the output swings from 0 to 8V. Because of the propagation delay of the transistor switching network, the 8V output switches from 0 to 8V after the −5V supply switches from 0 to −5V. D_1 sets the disable threshold of the −5V supply to approximately −4V to minimize the delay between the −5V supply switching from −5V to 0V and the regulator switching from 8V to 0V. To ensure that V_{GG} remains at −5V after disabling IC_1, you can exploit the high FET gate resistance and the low-leakage Schottky-diode characteristic. The combination of high gate resistance of the GaAs FET, the low-leakage Schottky diode, D_2, and the 10-µF capacitor, C_1, provides a high V_{GG} RC time constant when the −5V supply is off (in other words, at 0V). The RC time constant of the Schottky-diode leakage resistance, the FET gate resistance, and C_1 is long compared with the RC time constant at V_{DD}. As well as having low reverse leakage, D_2 has an inherently low (0.1V) forward drop.

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

Build your own bypass-capacitor tester

Carl Pugh, Pugh Magnetics, Newark, CA

Most circuits use bypass capacitors and can deliver substandard performance if the capacitors have poor pulse characteristics. Few if any articles cover how to test bypass capacitors for pulse characteristics. The circuit in **Figure 1** tests these characteristics. It charges the capacitor under test through 100 kΩ for approximately 1 msec and then discharges it through 10Ω for approximately 40 nsec. The cycle then repeats. The circuit uses a double-sided copper-clad pc board. All the components except the 10Ω resistor connect to one side of the board, so they can benefit from shielding by the cast-aluminum enclosure (**Figure 2**). All leads are as short as possible and as close as possible to the copper-clad board. The layout is such that you don't need the oscilloscope probe's ground lead; the ground on the probe contacts a ground post on the pc board. The ground posts, feedthroughs, connections to the capacitor under test, and oscilloscope probes use vector-board terminals.

The circuit comprises an astable multivibrator using two 2N3904 transistors, Q_1 and Q_2, and associated components. A voltage reducer/shaper uses a trimmer capacitor, C_1; a 100-pF capacitor, C_2; and a 200Ω resistor, R_1. An amplifier uses a 2N3906 transistor, Q_3, and associated components, and a power amplifier uses a PN2222A transistor, Q_4, and associated components. C_1, C_2, and R_1 produce a fast rise- and fall-time, 0.7V pulse when the multivibrator's output switches negative. Because the Q_3 transistor has no bias, it conducts only at the peak of the input pulse and produces a pulse with fast rise and fall times. The output from the Q_3 drives Q_4, causing that transistor to conduct for approximately 40 nsec. You can obtain interesting results when testing capacitors with long leads and then testing the same capacitors with short leads, corroborating the universal advice to keep leads short.

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

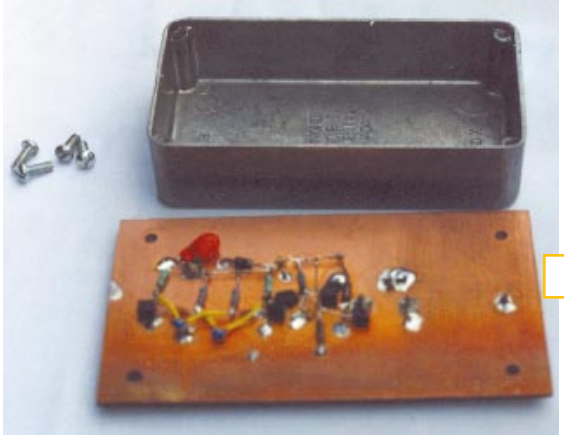
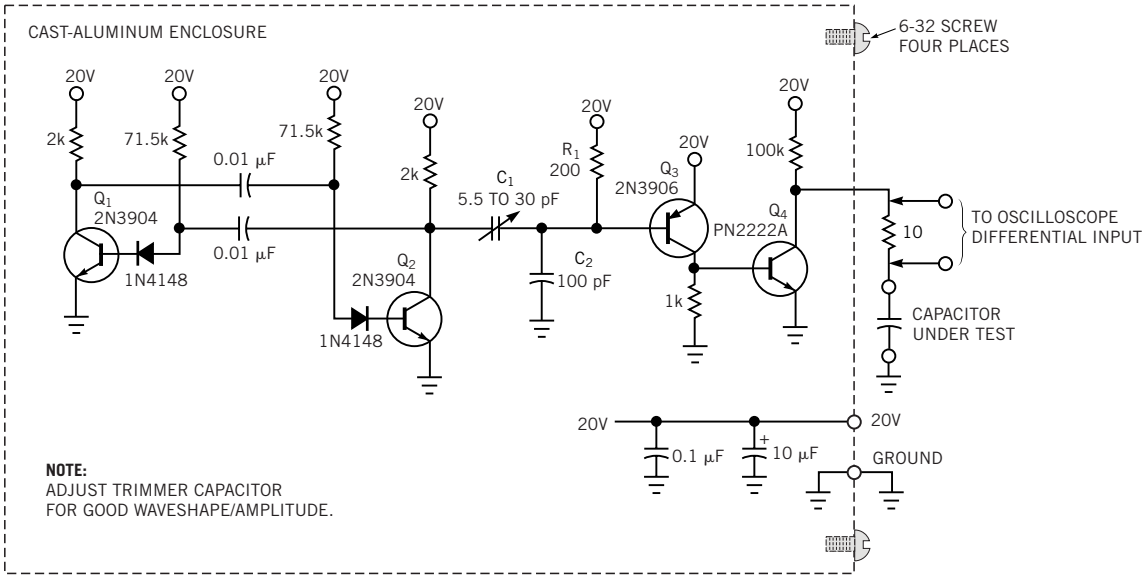


Figure 2

Components all connect to one side of the pc board, and a cast-aluminum cover provides shielding.

Figure 1



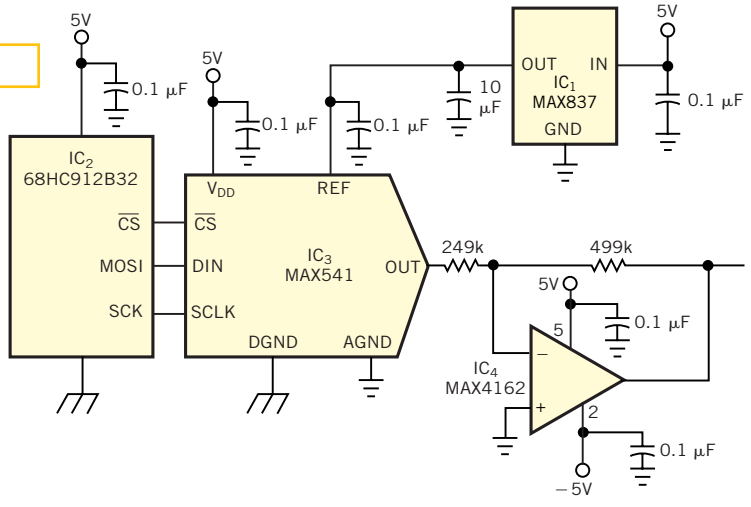
Test the pulse characteristics of bypass capacitors using this simple circuit.

DAC and op amp provide variable-control voltage

Chad Olson, Maxim Integrated Products, Sunnyvale, CA

Early DACs contained standard R-2R ladder networks, and produced a negative output voltage. These early DACs, such as the MAX7837/7847 and the MAX523, require both positive and negative supply rails to accommodate their negative output. With the transition to single-supply ICs, however, many modern DACs operate with a single supply rail and an inverted R-2R ladder network. The inverted R-2R network produces a positive output voltage. Despite the popularity of single-supply ICs, some applications still require a negative control voltage. **Figure 1** shows a circuit that satisfies this requirement. The circuit contains a modern, inverted R-2R ladder DAC and one op amp. In comparison with older DACs

Figure 1



This compact circuit allows microcontroller IC_2 to generate a variable negative voltage.

LISTING 1—TRIANGLE-WAVE GENERATOR

```
* Equates
*****************************************************************
SPOCR1          equ $00D8               ;SPI Control Register 1
SPOCR2          equ $00D9               ;SPI Control Register 2
SPOBR           equ $00DA               ;SPI Baud Rate Register
SPOSR           equ $00DB               ;SPI Status Register
SPODR           equ $00DD               ;SPI Data Register
PORTS           equ $00D6               ;Port S Data Register
DDRS            equ $00D7               ;Port S Data Direction
                                        ;Register

BIT4            equ %00010000
BIT6            equ %01000000
BIT7            equ %10000000

*****************************************************************
* External Variables
*****************************************************************
                XDEF Main

*****************************************************************
* Main Program
*****************************************************************

Main:           bsr Init
Function:       bsr RampDown
                bsr RampUp
                bra Function

*****************************************************************
* Initialization
*****************************************************************

Init:
                bset PORTS, #BIT7       ;Set SS line high to prevent
                                        ;glitch
                movb #$30, DDRS         ;MOSI, SCK, /SS = outputs
                movb #$00, SPOBR        ;SCK = 4MHz
                movb #BIT4, SPOCR1      ;CPOL, CPHA = 0
                                        ;SPI to Master
                ldaa SPOSR              ;clear SPIF Flag (1 of 2)
                ldaa SPODR              ;clear SPIF Flag (2 of 2)
                bset SPOCR1, #BIT6      ;Enable the SPI (SPE=1)

                rts

*****************************************************************
* RampUp
*       Outputs $0000 to $FFFF Through the SPI Interface
*****************************************************************
```

```
RampUp:
                lda #$0000
RampUpNext:     xgdx
                bclr PORTS, #BIT7       ;Clears /SS selecting MAX541
                staa SPODR              ;write most significant byte
                bsr WaitForSPI
                stab SPODR              ;write least significant byte
                bsr WaitForSPI
                bset PORTS, #BIT7       ;Sets SS latching data into
                                        ;the MAX541
                xgdx
                inx
                bne RampUpNext

                rts

*****************************************************************
* RampDown
*       Outputs $FFFF to $0000 Through the SPI Interface
*****************************************************************

RampDown:
                lda #$0000
RampDownNext:   dex
                xgdx
                bclr PORTS, #BIT7       ;Clears /SS selecting MAX541
                staa SPODR              ;write most significant byte
                bsr WaitForSPI
                stab SPODR              ;write least significant byte
                bsr WaitForSPI
                bset PORTS, #BIT7       ;Sets SS latching data into
                                        ;the MAX541
                xgdx
                cpx #$0000
                bne RampDownNext

                rts

*****************************************************************
* WaitForSPI
*       Tests the status of the SPIF bit in the SPOSR register.
*       This bit is set by the MCU when the SPI interface has
*       finished clocking out its data.     After SPIF is set
*       WaitForSPI returns.
*****************************************************************

WaitForSPI:     brclr SPOSR, #BIT7, WaitForSPI

                rts

*****************************************************************
```

containing standard R-2R ladders, this approach offers lower supply voltages, higher speed, and smaller packages. The DAC, IC_3, operating with a 2.5V reference voltage from IC_1 and driven by microcontroller IC_2, produces an output swing from 0 to 2.5V. Op amp IC_4 inverts and amplifies this output to produce a 0 to −5V output. For test purposes, the software routine in **Listing 1** commands the microcontroller to generate a 0 to −5V triangle-wave output. You can download the listing from the Web version of this article at www.ednmag.com.

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

Logic offers complementary-switch control

Yen-Hsu Chen, Analog Integrations Corp, Hsinchu, Taiwan

The complementary-switch controller in **Figure 1** uses a few inverter gates to provide drive signals for the complementary switches. Complementary-switch configurations find widespread use in synchronous-rectification circuits, charge pumps, full-bridge control circuits, and other circuits. The circuit in **Figure 1** provides not only a complementary drive signal but also a deadtime delay on both rising and falling edges. The high-speed inverter gates use IC_1, a 74HC04 CMOS circuit, and 1N5819 Schottky diodes D_1 and D_2. The 74HC04 inverter features symmetrical input thresholds, V_{IHMIN} and V_{ILMAX}, at 70 and 30% of the supply voltage, respectively. In **Figure 1**, IC_{1A} inverts the signal at Node A to produce $\overline{A}$. When $\overline{A}$ rises, C_1 rapidly charges through D_1. Output B drops immediately because of IC_{1B}'s inversion. However, Output C drops after a delay time that R_2 and C_2 determine because D_2 is reverse-biased. The following formula gives the delay time, t_1 (**Figure 2**):

$$t_1 = -R_2C_2 \ln\frac{0.7\,V_{DD}}{V_{DD}} = -R_2C_2 \ln 0.7.$$

When $\overline{A}$ falls, C_1 discharges through R_1. Output B rises after a delay time that R_1 and C_1 determine. C_2 discharges rapidly through D_2, and output C rises immediately. The following formula gives the delay time, t_2:

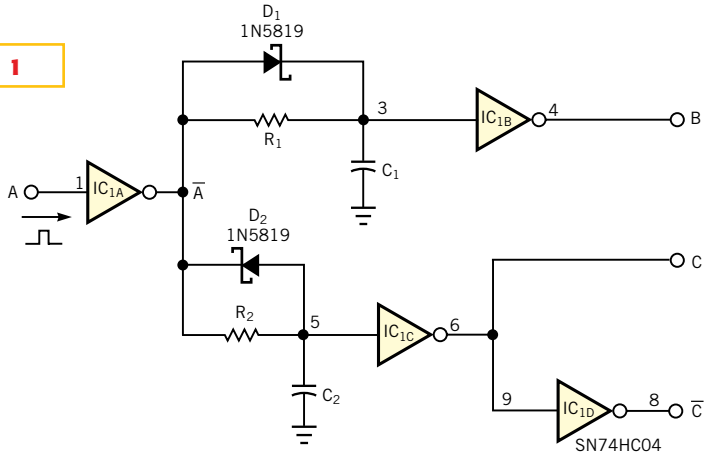
$$t_2 = -R_1C_1 \ln\frac{V_{DD} - 0.3\,V_{DD}}{V_{DD}} =$$

$$-R_1C_1 \ln 0.7.$$

By inverting C, IC_{1D} can provide $\overline{C}$ a signal with the same polarity as B. By selecting values for R_1, C_1, R_2, and C_2, you can program the delay times. The delay can be as short as 50 nsec and as long as several milliseconds. This range provides flexible, optimized control for target devices. R_1 and R_2 should be larger than 2 kΩ because of the limited current available from the inverter IC.

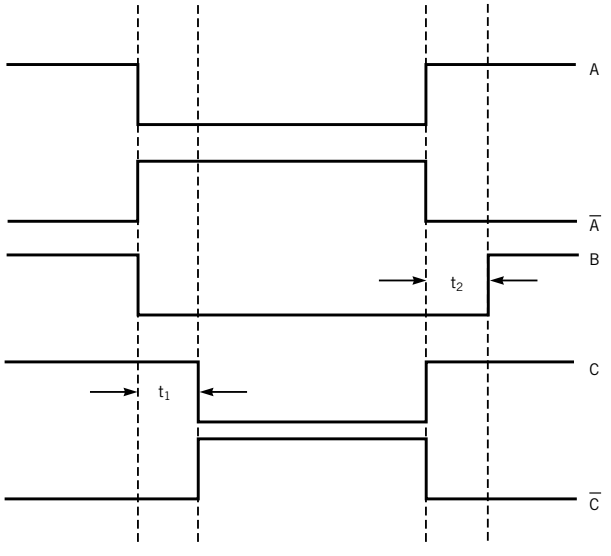
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Figure 1



This circuit provides drive for complementary switches with programmable deadtimes.

Figure 2



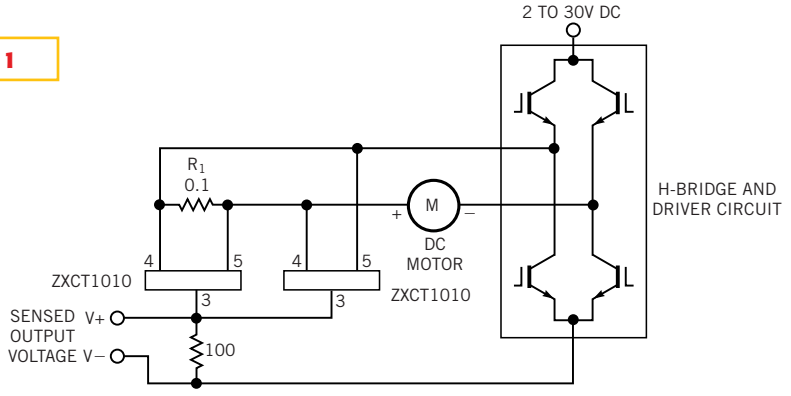
By manipulating the resistor and capacitor values in Figure 1, you can program t_1 and t_2.

Circuit measures currents in dc servo motor

Shyam Tiwari, Sensors Technology Private Ltd, Gwalior, India

THE SIMPLE CIRCUIT DESIGN in **Figure 1** lets you measure all components of a current flowing in a dc servo motor. The rectified output of the circuit uses ground as a reference, so you can measure the output by using a single-ended A/D converter. The current-sense resistor, R_1, has a value of 0.1Ω. The Zetex (www.zetex.com) ZXCT1010 IC converts the differential signal across R_1 to a single-ended signal. Two of these ICs form a signal rectifier. The single-ended signal makes measurement by an A/D converter cost-effective, small, and frugal in power consumption. The method also makes it possible to measure current from many sources at a time, such as in robots that use multiple servo motors. Measurement accuracy is approximately ±3%, which is adequate in most dynamic systems. Hence, an 8-bit A/D converter suffices to digitize the signal. If an average value of the current is of interest, then you can place an averaging capacitor between the V+ and V− terminals to remove the ac component. The unfiltered signal has 300-kHz response to ac current.

Figure 1



With this simple circuit, you can measure the currents in a dc servo motor.

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

design ideas

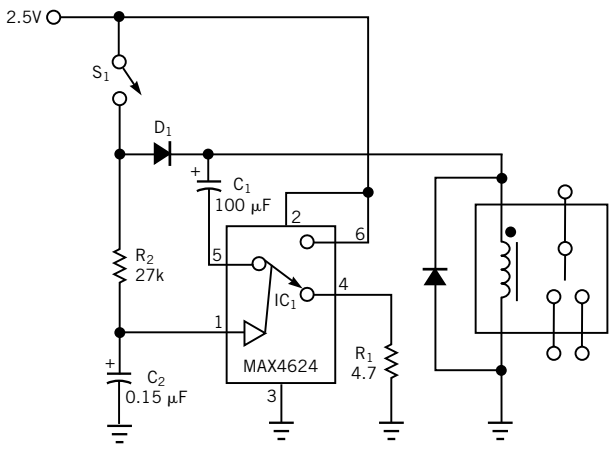
Edited by Bill Travis

Analog switch lowers relay power consumption

Steve Caldwell, Maxim Integrated Products, Chandler, AZ

Designers often use relays as electrically controlled switches. Unlike transistors, their switch contacts are electrically isolated from the control input. However, the power dissipation in a relay coil may render the device unattractive in battery-powered applications. You can lower this dissipation by adding an analog switch that allows the relay to operate at a lower voltage (**Figure 1**). The power that a relay consumes equals V^2/R_{COIL}. The circuit lowers this dissipation after actuation by applying less than the normal 5V operating voltage. Note that the voltage required to turn a relay on the pickup voltage is greater than the pickup voltage required to keep in on the dropout voltage. The relay in **Figure 1** has a 3.5V pickup voltage and a 1.5V dropout voltage. The circuit allows the relay to operate from an intermediate supply voltage of 2.5V. **Table 1** compares the relay's power dissipation with the fixed operating voltages applied and with the circuit in **Figure 1** in place.

Figure 1



By using an analog switch, you can reduce a relay's power consumption.

TABLE 1–RELAY POWER DISSIPATION

Voltage (V)	Current (mA)	Total power dissipation (mW)
5 (normal operating voltage)	90	450
3.5 (pickup voltage)	63	221
2.5 (circuit of Figure 1)	45	112

When you close S_1, current flows in the relay coil, and C_1 and C_2 begin to charge. The relay remains inactive because the supply voltage is lower than the pickup voltage. The RC time constants are such that C_1 charges almost completely before the voltage across C_2 reaches the logic threshold of the analog switch. When C_2 reaches that threshold, the analog switch connects C_1 in series with the 2.5V supply and the relay coil. This action turns the relay on by boosting the voltage across its coil to 5V, which is twice the supply voltage. As C_1 discharges through the coil, the coil voltage drops back to 2.5V minus the drop across D_1, but the relay remains on because its coil voltage is above the relay's 1.5V dropout voltage. Component values for this circuit depend on the relay characteristics and the supply voltage. The value of R_1, which protects the analog switch from the initial current surge through C_1, should be low enough to allow C_1 to charge rapidly but high enough to prevent the surge current from exceeding the peak current specified for the analog switch.

IC_1's peak current is 400 mA, and the peak surge current is $I_{PEAK}=(V_{IN}-V_{D1})/(R_1+R_{ON})$, where R_{ON} is the on-resistance of the analog switch (typically 1.2Ω). The value of C_1 depends on the relay characteristics and on the difference between V_{IN} and the relay's pickup voltage. Relays that need more turn-on energy need larger values of C_1. You select the values for R_2 and C_2 to allow C_1 to charge almost completely before C_2's voltage reaches the threshold of the analog switch. In this example, the time constant R_2C_2 is approximately seven times $(R_1+R_{ON})C_1$. Larger R_2C_2 values increase the delay between switch closure and relay activation.

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

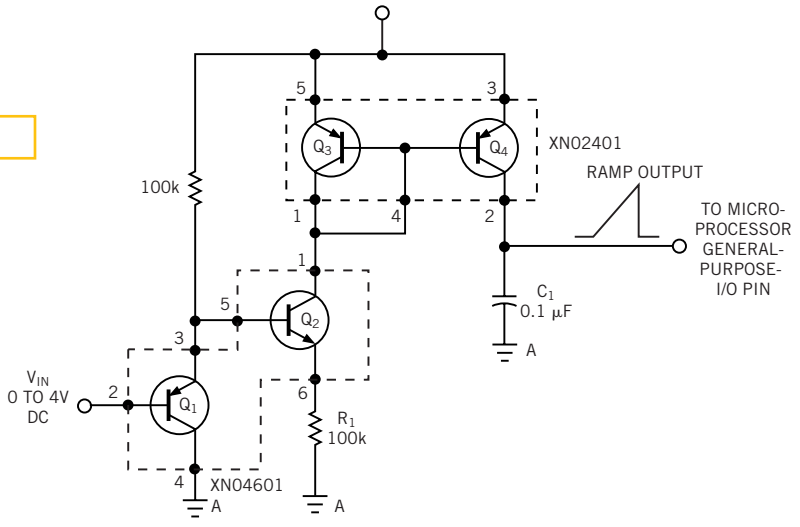
Analog-input circuit serves any microcontroller

Steven Hageman, Agilent Technologies, Santa Rosa, CA

THE SIMPLE ADC in **Figure 1** is perfect for getting analog signals into a purely digital microcontroller. Using just five surface-mount parts, you can assemble it for less than 50 cents (1000), which is approximately half the cost of a single-chip-ADC approach in the same volume. Moreover, this design takes only one pin from the microcontroller to operate. Although you can purchase many microcontrollers with built-in ADCs, in some circumstances, this solution is impractical. For example, you might have an all-digital microcontroller already designed in. In this design, a USB-compatible, digital-only microcontroller needed analog input at low cost for a consumer application. The basic analog portion of the circuit in **Figure 1** uses clever transistor arrays from Panasonic (www.panasonic.com). Q_1/Q_2 and Q_3/Q_4 are single-package, multiple-transistor arrays. The Q_1/Q_2 array forms a voltage-to-current converter. The voltage on Q_1's emitter is a diode drop higher than the voltage on Q_1's base. The V_{BE} drop in Q_2 returns the original input voltage to the top of R_1; R_1 then converts that voltage to a current.

The Q_3/Q_4 array forms a standard current-mirror circuit. The current flowing in Q_3's collector matches the current forced in Q_4's collector.Q_4's collector has high impedance, so Q_4 provides a suitable current source. The current from Q_4 charges C_1 at a rate that is proportional to the input voltage. The values in **Figure 1** allow for a range of conversion times of 3 msec for an input of 4V to 56 msec for an input of 0.1V. The design exploits the fact that most general-purpose microcontrollers have a bidirectional I/O-port structure. That is, you can program a port pin as either an input or an output. When you set a pin as an input, it has very high input impedance, so it can follow the ramp as C_1 charges up. When you program a pin as an output, you can set it low, and it discharges C_1 for the next conversion cycle. This action gives you the basic operation of a single-slope analog-to-digital-conversion cycle.

Figure 1



With two transistor arrays and three discrete components, you can configure an analog front end for a microcontroller.

The basic operations are as follows:

1. Set the ADC pin as a low output to discharge C_1.
2. Reset a suitable timer-counter in the microcontroller.
3. Set the ADC pin as an input.
4. Allow the timer to count until it reads as logic 1 in the microcontroller, or let the timer count to some suitably long value, which suggests that the input is essentially zero.
5. Stop the timer counter.
6. Convert to the timer count by some suitable scaling factor to an ADC reading.
7. Start over for the next conversion.

The conversion from the ramp time to a logic 1 on the microcontroller pin depends on the following factors:

- the logic-1 switching level of your microcontroller;
- the input voltage and, hence, the ramp rate of C_1;
- the value of C_1, which sets the ramp rate;
- the value of R_2, which sets the ramp rate; and
- the microcontroller's timer resolution.

You can boil down these variables to the following equation:

$$\frac{C_1 V_L}{dT} K = V_{IN},$$

where V_L is the voltage level of the microcontroller's zero-to-one conversion, K is the scaling factor that relates to the voltage-to-current conversion of the input stage and timer resolution, and dT is the time count of the conversion cycle. Because C_1V_L is also a constant for a given circuit, you can combine it with K to form a single conversion constant of K_1. Hence, you can reduce the equation to $K_1/dT = V_{IN}$.

In this case, the test code was written for Microchip Technology's (www.microchip.com) PIC16F84 microcontroller. This device has a measured V_L of 1.28V; the counter has a resolution of 1 µsec. It's probably best to empirically determine the factor K_1. Set up the counter resolution as desired, allow the microcontroller to make and display that conversion time or send it through a debugger, and, given that you have an exact V_{IN}, K_1 is then easy to determine. In this case, K_1 turned out to be 2V×5700 µsec=11,400.

The constant K_1 serves to convert the raw timer count to a voltage. To obtain high resolution, you normally use floating-point math. If you need to display the value, floating-point math might be ap-

propriate, but most applications entail reading a potentiometer or some other system level. In such applications, the output is a bar-chart display or some control value. Thus, you waste microcontroller resources by using floating-point math throughout the conversion process. With careful selection of circuit components, fixed-point math can usually provide, for example, an 8-bit representation (0 to 255) for an input range of 0 to 4V. If you scale the timer/counter by 64, instead of a count of 5700 µsec for an input of 2V, you obtain 89. Then, if you want this 89 to correspond to a half-scale value of 128, the value of K_1 becomes 11,392. A 16-bit unsigned word easily accommodates this value, and you need no floating-point math in the conversion. The accuracy of this ADC is approximately 5% with no adjustments. The resolution is a function of the timer resolution and how tight the code makes the conversion loop. The resolution can be many times the absolute accuracy. Moreover, the converter is monotonic.

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

Transistor tester fits into your pocket

Jean-Bernard Guiot, DCS AG, Allschwil, Switzerland

It can be helpful to rapidly and easily determine the polarity (npn or pnp) and function of a transistor. The pocket-transistor tester in **Figure 1** is ideal for quickly testing without regard to such parameters as gain and frequency response. You connect the transistor, or device under test, between the collectors, T, of an astable multivibrator. Thus, the V_{CE} voltage of the device under test is alternately positive and negative. Two LEDs connected in an antiparallel configuration to the device alternately light as long as the device is not conducting. The frequency of the multivibrator is a function of the values of C and R_B. If the device under test conducts in only one direction, then only one LED turns off. If the device conducts in both directions, then both LEDs turn off. You can leave the base of the device unconnected to check for excessive leakage current or short circuits between base and collector or base and emitter. Using the switch, S_1, you can connect the base to the collector to inject current into the base of the device under test. **Table 1** sums up the behavior of the tester.

TABLE 1—TESTING RESULTS

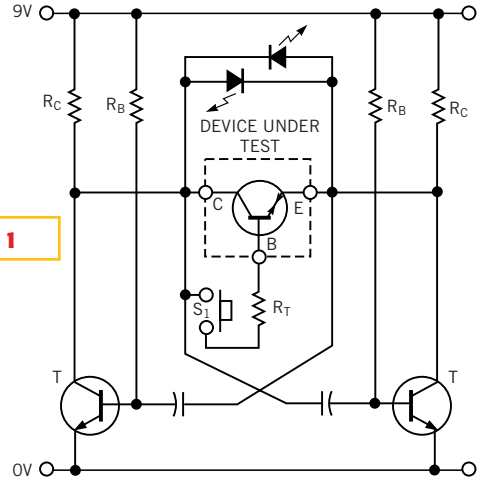
Test	D_1	D_2	S_1	Comments
1	On	Off	Off	Wrong connection? Invert C and B.
2	Off	On	Off	Wrong connection? Invert C and B.
3	Off	Off	Off	Device under test shorted (bad).
4	On	On	Off	Device under test is OK if test 5 or 6 is OK.
5	On	Off	On	Device under test is pnp.
6	Off	On	On	Device under test is npn.

You can also test diodes connected between C and E, FETs, small thyristors, and triacs. You can mount the entire circuit inside a small housing, such as one measuring 20×30×60 mm. You can effect the external connections to the device under test with wires terminated in alligator clips or by using a connector. It is practical and economical to use a five-pole DIN plug with the pinout shown in **Figure 2a**. This pinout allows you to easily connect any transistor, regardless of the arrangement of the CBE connections. **Figure 2b** shows the S_1 switch connections. S_1 is a DPDT switch with three positions:

- Position 1 is On, with no base current (S_1 open).
- Position 2 is Off (middle position).
- Position 3 is On, with base current (S_1 closed).

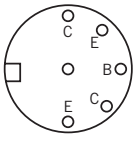
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Figure 1

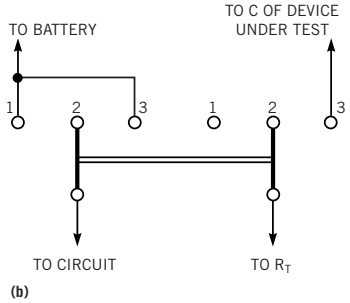


This simple tester allows you to test the polarity and function of a transistor.

Figure 2



(a)



(b)

A DIN connector (a) allows you to easily connect transistors; a DPDT switch provides various testing options (b).

Circuit combines power supply and audio amplifier

Susanne Nell, Breitenfurt, Austria

THE CIRCUIT IN **Figure 1** can help if you must transfer dc power and audio over a pair of copper wires. One application for such a circuit is a low-cost door-opening system with speech input. The circuit uses only one IC, the well-known LM317, a low-cost power-supply regulator. Using this chip, you can modulate the adjustment-pin input with the audio signal from an electret condenser microphone, connected between the output and the adjustment terminals of the IC. The LM317 regulates the output in such a way that the voltage on the microphone is always 1.25V dc. This application uses a WM34 electret microphone, which comes in a standard 10-mm capsule from Panasonic and is common in low-cost equipment. You can use nearly any electret capsule, because the well-regulated voltage on the microphone never exceeds 1.25V. Every electret capsule contains an integrated JFET-based impedance converter that translates speech into a current flowing from the source to the drain terminal. This current through the microphone modulates the voltage on the variable resistor, R_P. Because the output of the LM317 must follow the voltage on R_P, you obtain a low-impedance audio signal riding on the output dc voltage.

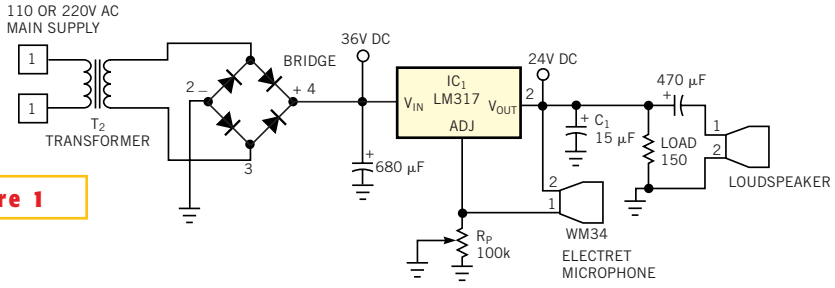


Figure 1

A novel circuit uses the adjustment pin of a regulator IC to provide audio amplification.

The microphone directly modulates the adjustment pin, so a smoothing capacitor, such as C_1, for noise and hum does not influence the level of the audio signal. C_1 shunts some of the audio signal to ground, but the LM317 compensates for the loss with internal gain. To avoid excessive losses in the LM317, use a capacitor with as low a value as possible. The circuit works well without a capacitor, but values as high as 47 µF do not present a problem. Using R_P, you can adjust the dc output voltage and the gain for the microphone signal. For proper operation, the LM317 needs to deliver a minimum current of 4 mA from its output terminal. If your design uses no loudspeaker, you can connect a load resistor to sink this 4 mA. Designs using low-impedance loudspeakers must also have load resistors. You must add the ac current in the audio signal to the minimum current requirement of 4 mA. For an 8Ω loudspeaker, you need a minimum resistive load of 470Ω to avoid distortion.

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

Supply derives 5 and 3.3V from USB port

Chad Olson, Maxim Integrated Products, Sunnyvale, CA

THE CIRCUIT IN **Figure 1** derives its power from a USB port and produces 5 and 3.3V supply rails for portable devices, such as digital cameras, MP3 players, and PDAs. The circuit allows the port to maintain communications while, for example, charging a lithium-ion battery. IC_2 boosts the battery voltage, V_{BATT}, to 5V, and IC_3 buck-regulates that 5V output down to 3.3V. IC_1, a lithium-ion battery charger, draws power from the USB port to charge the battery. Pulling its SELI terminal low sets the charging current to 100 mA for low-power USB ports, and pulling SELI high sets 500 mA for high-power ports. Similarly, pulling SELV high or low configures the chip for charging a 4.2 or 4.1V battery, respectively. To protect the battery, IC_1's final charging voltage has 0.5% accuracy. The $\overline{CHG}$ terminal allows the chip to illuminate an LED during charging.

IC_2 is a step-up dc/dc converter that boosts V_{BATT} to 5V and delivers currents as high as 450 mA. Its low-battery detection circuitry and true shutdown capability protect the lithium-ion battery. By disconnecting the battery from the output, "true shutdown" limits battery current to less than 2 µA. An external resistive divider between V_{BATT} and ground sets the low-battery trip point. Connecting the low-battery output, LBO, to shutdown, SHDN, causes IC_2 to disconnect its load in response to a low battery voltage. The internal source impedance of a lithium-ion battery makes IC_2 susceptible to oscillation when its low-battery-detection circuitry disconnects a low-voltage battery from its load. As the voltage drop across the battery's internal resistance disappears, the battery voltage increases and turns IC_2 back on. For example, a lithium-ion battery with 500-mΩ internal resistance, sourcing 500 mA, has a 250-mV drop across its internal resistance. When IC_2's circuitry disconnects the load, forcing the battery current to

zero, the battery voltage immediately increases by 250 mV.

The n-channel FET at LBO eliminates this oscillation by adding hysteresis to the low-battery-detection circuitry. The circuit in **Figure 1** has a low-battery trip voltage of 2.9V. When V_{BATT} drops below 2.9V, LBO opens and allows SHDN to switch high, turning on the FET. With the FET turned on, the parallel combination of 1.3 MΩ and 249 kΩ eliminates oscillation by setting the battery turn-on voltage to 3.3V. The turn-off and turn-on points are according to the following equations:

$$V_{BATT}(TURN-OFF) = V_{LBI} \times \frac{R_1 + R_2}{R_2},$$

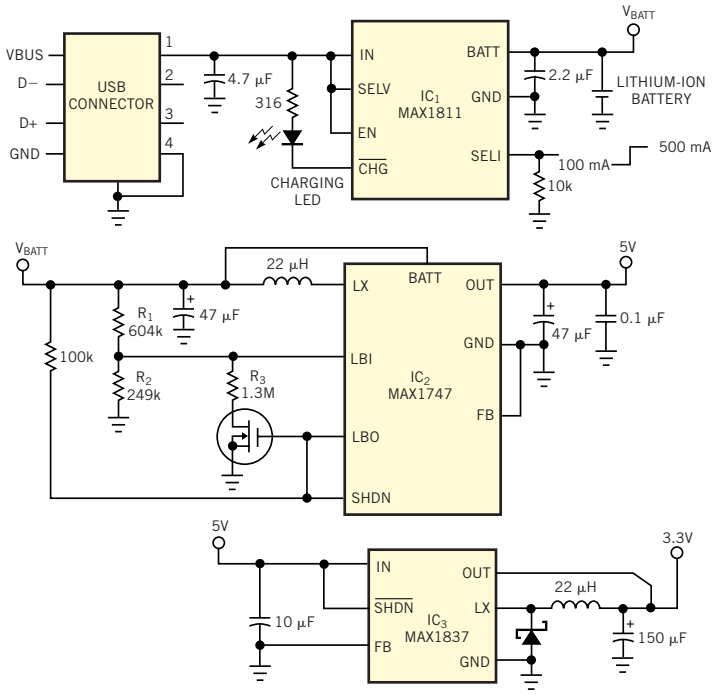
where V_{LBI}=0.85V, and

$$V_{BATT}(TURN-ON) = V_{LBI} \times \frac{R_1 + R'_2}{R'_2},$$

where

$$R'_2 = \frac{R_2 R_3}{R_2 + R_3}.$$

Figure 1



Drawing power from a USB port, this circuit generates 5 and 3.3V supply voltages for portable applications.

Finally, a step-down converter, IC_3, provides buck regulation to convert 5V to 3.3V and delivers currents as high as 250 mA with efficiency exceeding 90%.

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