designideas

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

EDN Design ideas 2010

January 2010
Circuit uses two reference voltages to improve hysteresis accuracy
Resistor network sets gain for fixed-gain differential amplifier
LED flasher checks fiber-optic strands
Circuit routes audio signals between equipment
Microcontroller provides an alternative to DDS
Excel spreadsheet measures analog voltages
Schottky diodes improve comparator’s transient response
Algorithm keeps data safe
P-channel power-MOSFET driver uses unity-gain op amp

February 2010
Spice simulators provide behavioral sources to mere mortals
Multidecade BCD DAC uses resistors of only six values
Converter translates Bayer raw data to RGB format
Drive 12 LEDs with one I/O line
Circuit precludes common-mode conduction
Use eight timers with PIC16Fxxx microcontrollers
Tilt/fall detector has staggered thresholds
Electronically generate rotating magnetic fields
Voltage reference stabilizes current sink

March 2010
Circuit lets you test sample-and-hold amplifiers
Add hysteresis to a voltage comparator
Broken-coil detector is simple yet robust
Battery simulator has variable ESR response
Rectangular-waveform generator produces 25 and 75% duty cycles
Strategy processes video in RAM
Control stepper motors in both directions
Create LED-lighting patterns without a controlle

April 2010
Single IC forms precision triangular-wave generator
MSP430's port-interrupt-request logic helps debounce contacts
Efficient LED power supply has battery backup
Use a low-cost PWM ramp generator in switch-mode power supplies
Tables ease microcontroller programming
Three-transistor modulator-amplifier circuit works with swept-control frequencies
Monitor alarm and indicator display multiple deviation boundaries
Read multiple switches and a potentiometer setting with one microcontroller input pin

EDN Design ideas 2010

May 2010
Pulse generator with precision output-duty cycle operates at a repetition rate beyond 50 MHz
Bicolor LED driver uses two leads
Supervise and power-sequence an SOC
Circuit may save property and life
Photoresistor provides negative feedback to an op amp, producing a linear response
Three-phase digital-signal generator sweeps frequency
Water-leak detector uses 9V batteries

June 2010
Isolated FET pulse driver increases power rate and duty cycle
Detect missing pulses to avoid losing data
Circuit lets you measure zener voltages and test LEDs
Switched-capacitor voltage multiplier achieves 95% efficiency
Bootstrap circuit speeds solenoid actuation
Low-cost RF synthesizer uses generic ICs
Tricolor LEDs create a flashing array
Image-capture system uses USB and LabView
Notch filter autotunes for audio applications

July 2010
Pulse generator with precision output-duty cycle operates at a repetition rate beyond 50 MHz
Tables ease microcontroller programming
Bicolor LED driver uses two leads
Single IC forms precision triangular-wave generator
DC-voltage doubler reaches 96% power efficiency
Microcontroller serial port measures pulse width
Count objects as they pass by
Eight-digit counter works with common anode or common cathode
Modified DDS functions as baud-rate generator
Pulse generator corrects itself
Methods measure power electronics' efficiency
Circuit extends battery life
Reflective object sensor works in bright areas

August 2010
Get four colors from 2 bits
Simple battery-status indicator uses two LEDs
Hardware watchdog timer accepts range of frequencies
Implement trapezoidal velocity profiles in software
Control a dc motor with your PC
Look-up table eliminates the need for an IC
Microcontroller supervises 0- to 20-mA protection circuit
Operate circuits at voltages as high as 540V ac
LED indicates power source

EDN Design ideas 2010

September 2010
Current monitor compensates for errors
Buck regulator handles light loads
Sense multiple pushbuttons using only two wires
Tricolor LED emits light of any color or hue
Acquire images with a sensor and a microcontroller
Amplifiers deliver accurate complementary voltages
Circuit lets you isolate and measure current
Power-supply circuit operates from USB port
LED-flashlight circuit works at voltages as low as 0.5V

October 2010
Perform hexadecimal-to-BCD conversion in firmware
Ceramic speaker driver also drives motors
Light collar helps you find your pet in the dark
Accurately simulate an LED
Circuit synchronizes sensors and cameras
Circuit measures capacitance or inductance
Set LEDs' hue from red to green

November 2010
Circuit achieves constant current over wide range of terminal voltages
Limit inrush current in low- to medium-power applications
Electronically tinge white-light source
Transistor boosts regulator current
Detect live ac-mains lines
Microcontroller drives DSEC motors
CMOS circuit latches relays
Power USB devices from a vehicle
Microcontroller interfaces to 24V
Use LEDs as photodiodes

December 2010
LEDs light LAN-cable tester
Instrumentation amplifier is less sensitive to resistor mismatch
Circuit keeps wandering children and pets nearby
High-side current-shunt monitor offers reduced error
Make a quick-turnaround PCB for RF parts
PLL filter blocks undesired frequencies
Logic probe uses six transistors

EDN Design ideas 2010

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas
READERS SOLVE DESIGN PROBLEMS

Circuit uses two reference voltages to improve hysteresis accuracy

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

In advanced IC comparators, programmable hysteresis eliminates 0V-centered differential-input-voltage noise (**Reference 1**) and improves the comparator's response if its differential-input voltage is low or varies slowly over time. For example, the ADCMP609 comparator from Analog Devices (www.analog.com) lets you program its hysteresis from 0 to 160 mV with a single resistor that connects between the HYS (hysteresis) pin and ground. That voltage range may be too narrow for some applications, however. The circuit in **Figure 1** lets you widen the hysteresis by using two reference voltages applied to the noninverting input. The circuit uses IC_3, an Analog Devices ADR390B, to generate a high reference voltage, V_{REFH}, of 2.048V. Resistor divider R_1/R_2 produces a low reference voltage, V_{REFL}, of 0.2048V, or a difference of 1.8432V. Thus, the hysteresis equals the high reference voltage minus the low reference voltage. IC_2, an Analog Devices ADG772 dual-SPDT (single-pole/double-throw) switch, routes the voltages to the comparator's noninverting input (**Reference 2**).

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

Whenever the input voltage at the comparator's inverting input exceeds the high reference voltage, the output

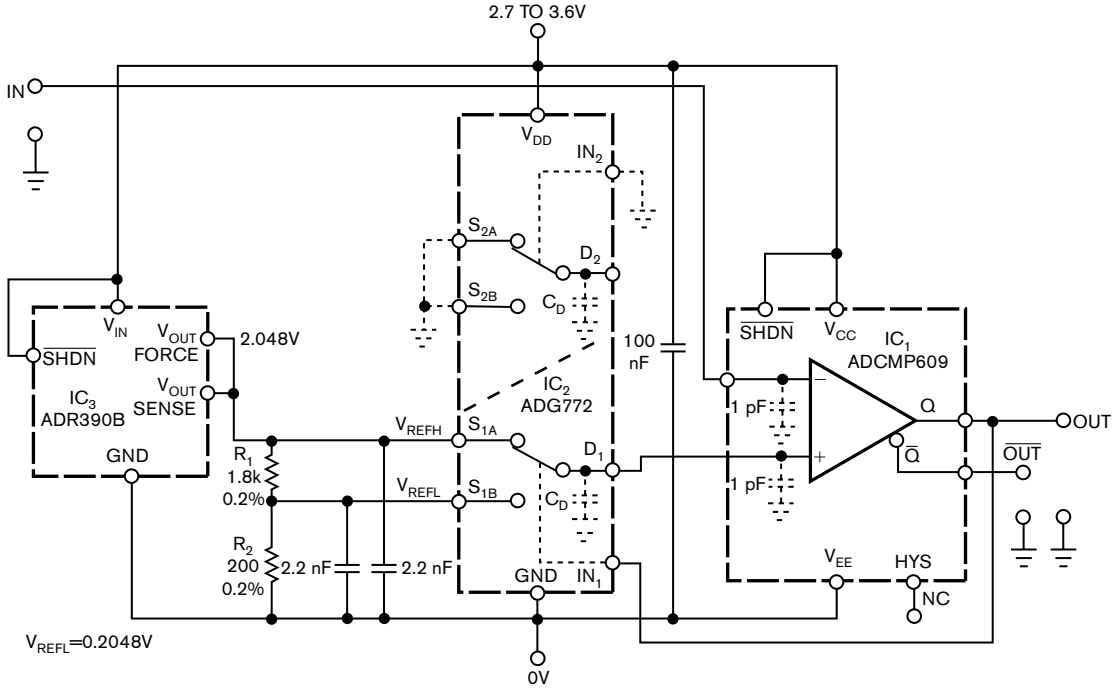


Figure 1 An analog switch changes the comparator's reference voltage, thus increasing hysteresis.

goes to a logic low. That action forces S_{1B} to connect the low reference voltage to the comparator's noninverting input. The circuit remains in that state until the input voltage drops below the low reference voltage. At that point, the switch connects the high reference voltage back to the comparator. For fast-ramping waveforms at the input, the hysteresis increases because of signal-propagation delays in both IC_1 and IC_2. The 35-nsec propagation delay in the ADCMP609 occurs at approximately 10 mV of input overdrive, and this overdrive roughly doubles in effect as an addition to the hysteresis voltage, increasing it by approximately 1%.

Because ADCMP60x comparators are rail-to-rail I/O devices, the low reference voltage could be 0V. In this case, however, the value of 0.2048V lets the comparator cooperate with other rail-to-rail I/O ICs from the same supply voltage. The outputs of these ICs can swing between 0V and the power-supply voltage with a margin of millivolts to tens of millivolts, depending on the load. An ADR390B provides the reference voltages (**Reference 3**). The 2.2-nF decoupling capacitors suppress variations of these voltages during voltage transitions at the Q output. These values are sufficient because the parasitic-charge transfer from the switch's control input, IN_1, to the respective channel's source electrode is typically 0.5 pC. The short-term variation of the decoupled reference voltages is less than 250 µV. EDN

REFERENCES

1 "ADCMP609 Rail-to-Rail, Fast, Low-Power 2.5 V to 5.5 V, Single-Supply TTL/CMOS Comparator," Analog Devices, 2007 to 2008, www.analog.com/static/imported-files/data_sheets/ADCMP609.pdf.

2 "ADG772 CMOS Low Power Dual 2:1 Mux/Demux USB 2.0 (480 Mbps)/USB 1.1 (12 Mbps)," Analog Devices, 2007 to 2008, www.analog.com/en/switchesmultiplexers/analog-switches/ADG772/products/product.html.

3 "ADR390/ADR301/ADR392/ADR395 Micropower, Low Noise Precision Voltage References with Shutdown," www.analog.com/static/imported-files/data_sheets/ADR390_391_392_395.pdf.

Resistor network sets gain for fixed-gain differential amplifier

Miles Thompson, Maxim Integrated Products, Sunnyvale, CA

When a fixed-gain amplifier doesn't offer the optimum gain for your application, you can adjust the gain to a lower value by adding an external resistor network. This attenuation circuit works like a voltage divider but with a key difference: Resistors inside the fixed-gain amplifier load down the external network (**Figure 1**). For the differential-input configuration, you can reduce this system to an equivalent half-circuit for analysis (**Figure 2**).

You determine the voltage gain for the differential configuration as follows:

$$\text{GAIN (dB)} = 20 \times \log\left(\frac{1}{1 + R_1/R_1 + 2R_1/R_2}\right) + \text{GAIN},$$

where gain is the amplifier's fixed-voltage gain in decibels. Maxim's (www.maxim-ic.com) MAX9705 differential-audio amplifier, for example, is available in fixed-gain versions of 6, 12, 15.5, and 20 dB, and its input resistance is typically 20 kΩ. For single-ended configurations, the gain is:

$$\text{GAIN (dB)} = 20 \times \log\left(R_2 \times \frac{1 - \frac{R_{1A}}{R_{1A} + (R_2 + R_{1B} \| R_{I2}) \| R_{I1}}}{R_2 + R_{1B} \| R_{I2}}\right) + \text{GAIN}.$$

These gain **equations** assume that

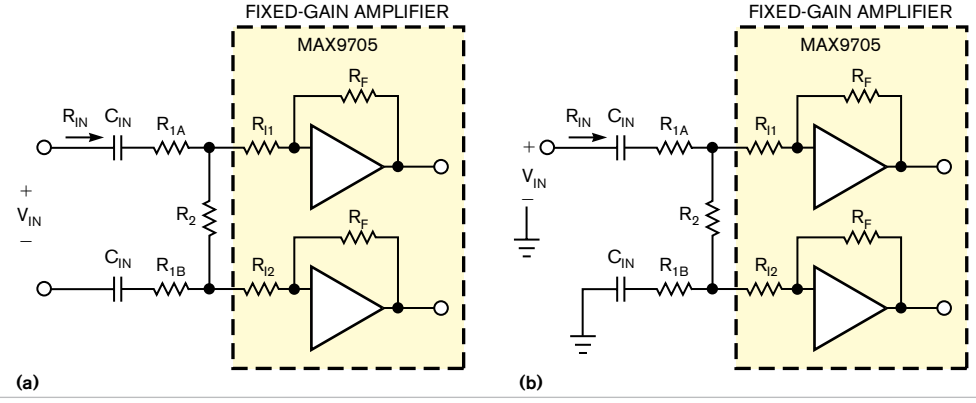


Figure 1 You can configure this fixed-gain audio amplifier for signals that are differential (a) or single-ended (b).

the frequency is much higher than the cutoff frequency of the highpass filter comprising C_{IN} and the equivalent input resistance of the circuit.

The fixed gain of the MAX9705 amplifier is typically within 5%, but the internal input resistors have an absolute tolerance of ±40%, which you must consider when calculating the resulting system gain. You must also account for the tolerance of the external resistors, using a worst-case approach for calculating the gain tolerance. Replacing the resistor values with their maximum deviations from normal—that is, the extreme values at the positive and negative tolerance limits—yields the worst-case scenario (**Table 1**).

For the tightest tolerance, choose a smaller value for R_I than for R_1 and remember to account for the source characteristics of the input voltage, which must be able to drive the equivalent load of the network. When determining the final gain for the system, remember that the output impedance of the input voltage forms a voltage divider with the input impedance of the attenuation circuit. You can determine this load, the input resistance, as follows: For the differential configuration, $R_{IN}=R_1+R_I\|(R_2/2)$. R_{IN} and C_{IN} form a highpass filter whose cutoff frequency, $F_{-3\,dB}$, is $1/(2\times\pi\times R_{IN}\times C_{IN})$. For the single-ended configuration, $R_{IN}=R_1+R_I\|(R_2+R_1\|R_I)$. Input impedance for the attenuation network is:

$$Z_{IN}=\frac{1}{2\pi\times F\times C_{IN}}+R_1+R_I\left\|\left(R_2+\left(R_I\left\|\left(R_1+\frac{1}{2\pi\times F\times C_{IN}}\right)\right)\right)\right.,$$

where Z_{IN} is the input impedance and F is the input-signal frequency.

Deriving the highpass cutoff frequency for the single-ended case is less straightforward. To find the cutoff frequency, you must know the resistor values. Once you know them, you can solve the following equation for the cutoff frequency: $Z_{IN}(f)=\sqrt{2}\times Z_{IN}(f=5000)$.

A spreadsheet is useful for determining the correct resistor values and system tolerances. Remember that resistors come in discrete values of resistance, and you must match rather than simply calculate them. A spreadsheet for determining custom gains and gain tolerances for single-ended and differential configurations of the MAX9705 is available at www.maxim-ic.com/an-4559-supplement. EDN

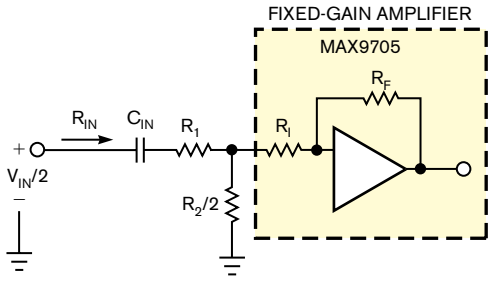


Figure 2 For the differential configuration of Figure 1a, an equivalent half-circuit simplifies analysis.

TABLE 1 WORST-CASE-TOLERANCE CALCULATION

Lowest gain	Single-ended	Differential	Highest gain	Single-ended	Differential
R_{1A}	Positive tolerance	Positive tolerance	R_{1A}	Negative tolerance	Negative tolerance
R_{1B}	Positive tolerance	Positive tolerance	R_{1B}	Negative tolerance	Negative tolerance
R_2	Negative tolerance	Negative tolerance	R_2	Positive tolerance	Positive tolerance
R_{I1}	Negative tolerance	Negative tolerance	R_{I1}	Positive tolerance	Positive tolerance
R_{I2}	Positive tolerance	Negative tolerance	R_{I2}	Negative tolerance	Positive tolerance

LED flasher checks fiber-optic strands

Edwin A Mack, Port Republic, NJ

The circuit in **Figure 1** allows you to verify fiber-optic strands, especially in cases in which a link goes through several patch panels. It uses two high-intensity LEDs that you can see at the far end. The flasher shuts itself off after about 50 minutes and has minuscule power drain when off. The circuit works on multimode fiber at distances greater than 1 km. It also works with single-mode fiber, but is more difficult to see on the receiving end. With short fiber, it is best to look at the far end at a slight angle due to the LED brightness.

IC_{1B}, a Schmitt-trigger oscillator running at approximately 5 Hz, drives IC_2, a 4020 binary divider. IC_{3A} is the control flip-flop. Pressing pushbutton switch S_1 sets the flip-flop, which starts the oscillator and enables the 4020 to start counting from its all-zero state. It also enables gates IC_{1A} and IC_{1D}, which control the PNP LED-driver transistors. Pressing pushbutton switch S_2 resets the control flip-flop; alternatively, the 4020 reaches the end of its count sequence, resetting the flip-flop. IC_{3B} divides the oscillator frequency by two to provide a 50% duty cycle to the LED drivers. The T1-¾ LEDs fit nicely into ST barrel connectors with some glue

to hold them in place. You can use a patch cable to match the fiber connectors in your network. Using a yellow LED works, considering the fiber attenuation. If you choose to use two red LEDs, however, then drive one of the LEDs at the oscillator frequency and drive the other at half the oscillator frequency. The component values and placement are not critical; the circuit in **Figure 1** resides in an old, surface-mount fiber-outlet box. EDN

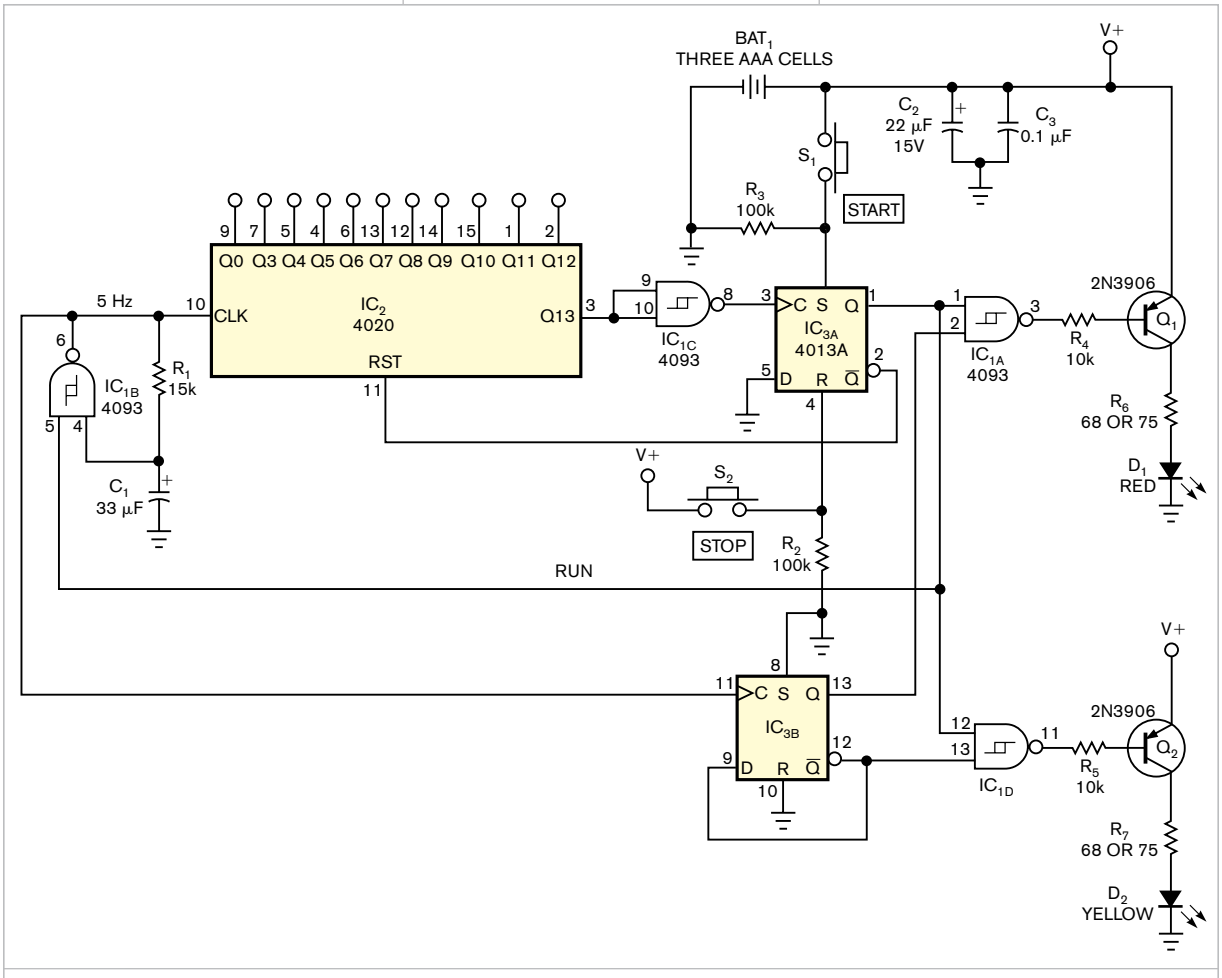


Figure 1 This circuit allows you to verify fiber-optic strands when a link goes through several patch panels.

Circuit routes audio signals between equipment

Roger Khourey, Los Angeles, CA

The audio-signal router in this Design Idea can automatically redirect cable connections between the audio equipment and an amplifier/receiver for dubbing or playback purposes without causing distortion or noise to the signal (**Figure 1**). It can be useful for older-model amplifier/receivers and any type of tape-deck-playback or -recorder equipment. The circuit uses four Maxim (www.maxim-ic.com) 4606 quad-SPST (single-pole/single-throw) CMOS analog switches to accommodate as many as eight playback and record signals for two pieces of audio equipment. Each MAX4606 contains two normally open and two normally closed logic switches. These switches have a maximum on-resistance of 5Ω and an on-resistance match of 0.5Ω between channels to minimize distortion. They can also handle rail-to-rail analog signals.

The circuit consumes little current and therefore requires no heat sinks on the power supplies. Because this circuit targets use with audio signals, it needs positive and negative supplies to pre-

(continued on pg 49)

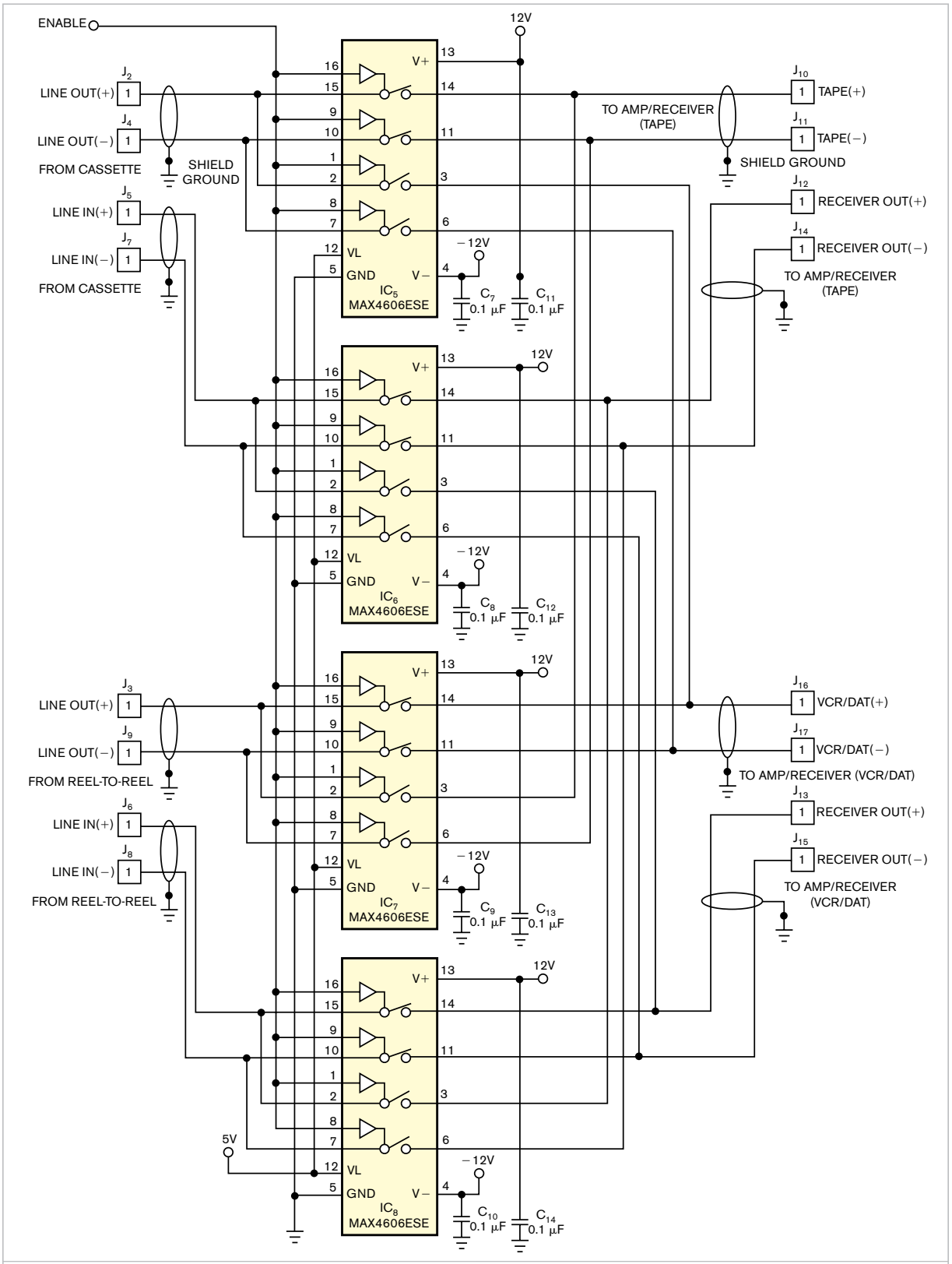


Figure 1 This audio-signal router automatically redirects cable connections between the audio equipment and an amplifier/receiver for dubbing or playback purposes without causing distortion or noise to the signal.

vent imbalance of the audio signal. It also needs a 5V supply for the logic input (VL), which lets you control the switching of the logic gates. **Figure 2** depicts the power-supply circuit, which requires a 10V-ac, 300-mA input. You can connect the enable input to a set/reset circuit or a flip-flop to toggle the internal switches from normally open to normally closed or from normally closed to normally open.

Assume that you have a cassette-tape deck and a reel-to-reel tape deck, in which the cassette deck connects to Tape 1 and the reel-to-reel deck connects to Tape 2 or another input in your amplifier/receiver. You want to record from the reel-to-reel to the cassette tape, but you can instead only record from the cassette deck to the reel-to-reel tape. When you apply a logic level, you will be able to switch your recording and playback direction to the target equipment with a single push of a button.

When wiring the circuit, you should shield and ground all wires from the equipment to the input switches and out to the equipment with reference to the audio cables' return path. This approach minimizes noisy signals due to a lack of proper grounding from one channel to another. Most of the circuit uses SMD-technology devices and can fit into a 2.5×3.5-in. enclosure and minimal installation space. EDN

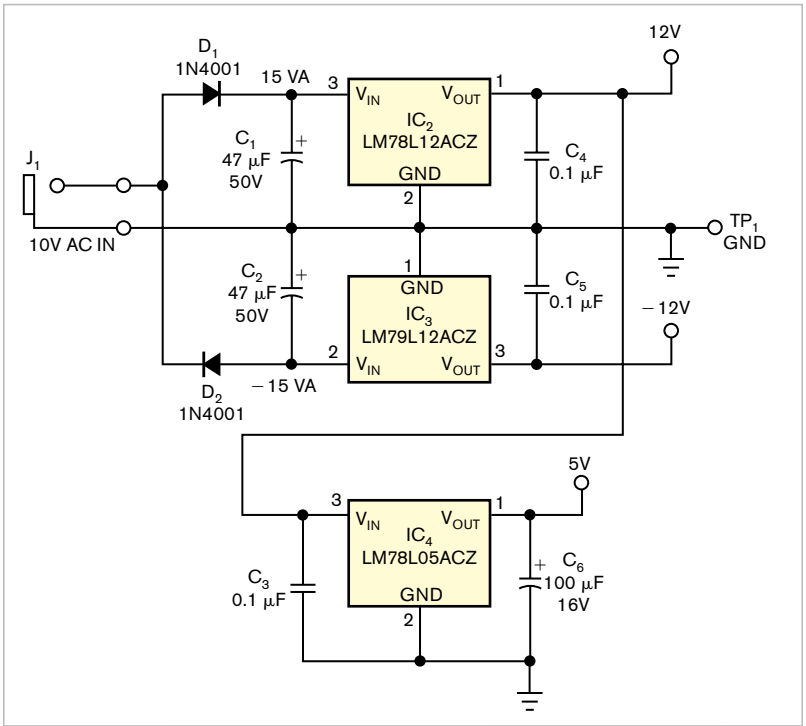


Figure 2 This ac/dc power-supply circuit powers the audio-signal router.

Microcontroller provides an alternative to DDS

Daniele Danieli, Eurocom-Pro, Venice, Italy

Audio and low-frequency circuit systems often require a signal source with a pure spectrum. DDS (direct-digital-synthesis) devices often perform the signal generation by using these specialized integrated circuits. A DDS device uses a DAC but often with no more than 16-bit resolution, limiting the SNR (signal-to-noise ratio). You can perform the same task with a microcontroller programmed as a DDS and use an external high-resolution DAC. To achieve 18 to 24 bits of resolution requires a large memory table containing the cosine function for any values of phase progression.

An alternative approach lets you use a standard microcontroller with a small memory and still implement an effective synthesizer. You can design a circuit to produce a sine wave using a scalable digital oscillator built with adder and multiplier block functions in a simple structure.

Figure 1 shows a microcontroller driving an audio DAC. To develop

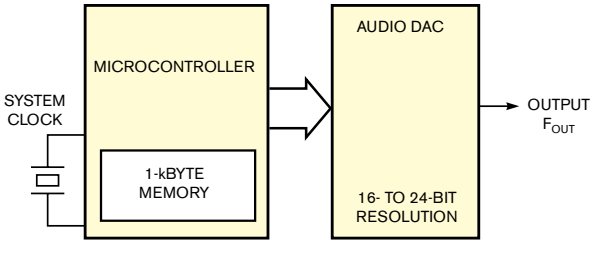


Figure 1 A microcontroller with a small memory can drive a DAC to generate audio-frequency signals.

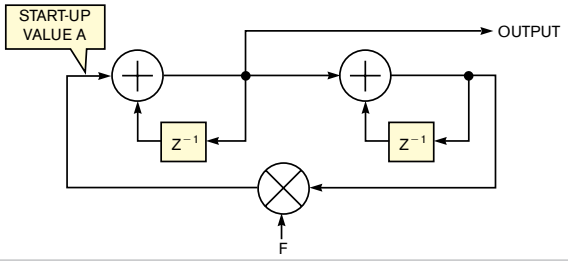


Figure 2 The functional software for a digital sine oscillator uses only two adders, one multiplier, and storage registers.

your code to generate a sine wave, the circuit in **Figure 2** comprises two integrators with an analog feedback loop equivalent to that of an ideal resonator. Parameter F defines the frequency and ranges from 0 to −0.2, and Parameter A sets the amplitude of the output signal with a single initial pulse at start-up. The following **equation** derives the frequency of generated signals: $F_{OUT}=(\sqrt{|F|})/(2\times\pi\times T)$, where T denotes the time for computing an entire sequence to obtain output data. The firmware for implementing this system is relatively straightforward. It requires just a few additions and one multiplication. Thus, you can use a slow microcontroller. Remember, though, that the precision of every operation must be adequate to warrant a complete signal reconstruction. Processing data with 8 or 16 bits isn't sufficient. You must write your firmware to emulate a greater number of bits, which requires accurate code implementation.

If you properly develop your code, then you should generate the DAC output codes that produce a sine wave (**Figure 3**). Remember that Parameter F is nonlinear with respect to the output frequency. If you need a directly proportional rate, you can square the value of F before applying it to the input. You'll find it useful when you need to make an easy frequency setting.

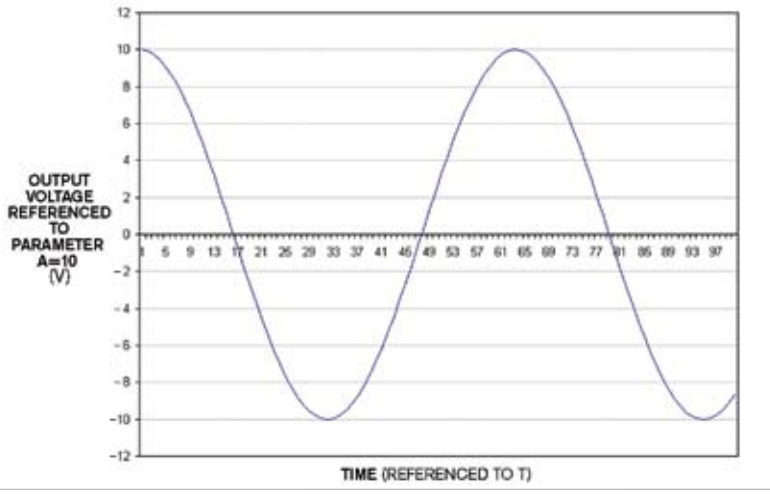


Figure 3 Use DDS to generate a sine wave.

You can use just about any microcontroller to implement the oscillator, together with a high-performance DAC. You can achieve an output SNR greater than 110 dB. Many audio DACs operating in monophonic mode have 20- to 24-bit resolution at a 192-kHz sampling rate. They also offer a dynamic range of 120 dB or more. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas
READERS SOLVE DESIGN PROBLEMS

Excel spreadsheet measures analog voltages

Yury Magda, Cherkassy, Ukraine

You can use the parallel port, which many computers still have, with some VBA (Visual Basic for Applications) code and Microsoft (www.microsoft.com) Excel 2007 to measure, record, and analyze analog signals. The hardware comprises a 12-bit Microchip (www.microchip.com) SAR (successive-approximation-register) ADC (**Figure 1**). The binary code that represents this signal passes to the parallel port through an SPI (serial-peripheral-interface)-compatible port, which includes three signal lines.

The analog input signal on Pin 2 of the MCP3201 ranges from 0 to 5V. The ADC's reference voltage is 5V, which provides an LSB (least-significant bit) of $5/2^{12}$, or 1.2207 mV. You can substitute the MCP3201 ADC with the Linear Technology (www.linear.com) LTC1286 chip, which is pin-compatible with MCP3201, but this replacement requires some changes in the source code. The circuit includes a 1- to 10-µF bypass capacitor that you should place as close as possible to the MCP3201's power pin. You can improve performance by placing a lowpass active filter between the signal source and the converter's input (Pin 2).

The design uses the PC's default parallel interface, LPT_1, which has three hardware registers: data, status, and control. Most PCs have only one parallel port with the base address 0x378 assigned to the data register. The status register has the address 0x379, and the unused control register has the 0x37A address. These settings are the defaults for most motherboards. This design uses addresses 0x378 and 0x379. The parallel port sometimes uses hardware addresses in the range of 0x278 to 0x27A. The hardware addresses may also vary when you apply a PCI (Peripheral Component Interconnect) expansion board with a built-in parallel interface. You should check the hardware addresses of the selected parallel interface and set appropriate values in the source code.

The program uses the inpout32.dll library, which is free and accessible for downloading at www.logix4u.net. The inpout32.dll includes two useful functions, `Inp32` and `Out32`. The `Inp32` function reads a data byte from the hardware register (`port`), and `Out32` writes the byte into the hardware port. Before using the inpout32.dll, you should copy the inpout32.dll into the \system32 directory so that the VBA application can find it. You can download **Listing 1**, the VBA code, from the online version of this Design Idea at www.edn.com/100121dia.

To write a VBA control program, you should create a new Excel workbook and open the Visual Basic editor where you enter the source code. After doing so, you must insert a new user form (**Figure 2**). You need to drop a button component from the toolbox onto the user form. When the application is running, clicking the button causes the value of the analog voltage to appear in the current cell of Column A of Spreadsheet 1.

You should declare functions `Out32` and `Inp32` from the inpout32.dll library with the `Declare` directive so that the application can use them. The `Out32` (`port, portVal`) holds the hardware address of the port to write data to. The value of the byte goes in the `portVal` parameter. The `Inp32` function

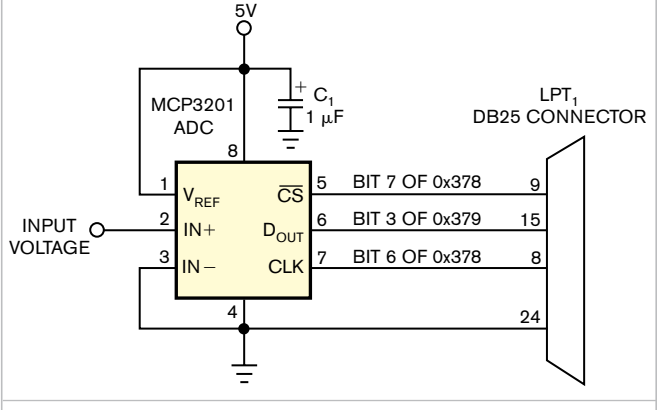


Figure 1 An MCP3201 ADC digitizes the input voltage.

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

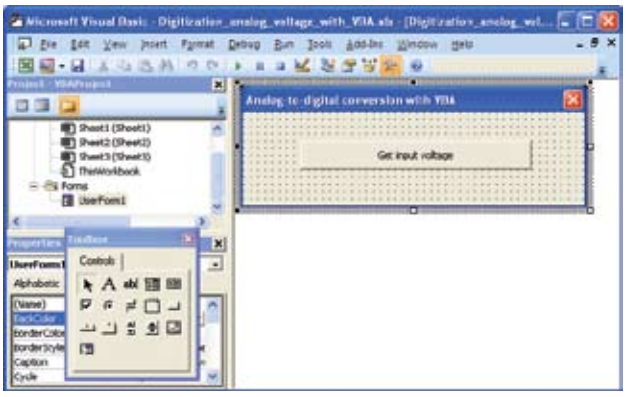


Figure 2 You can create a user form with a button to execute the conversion.

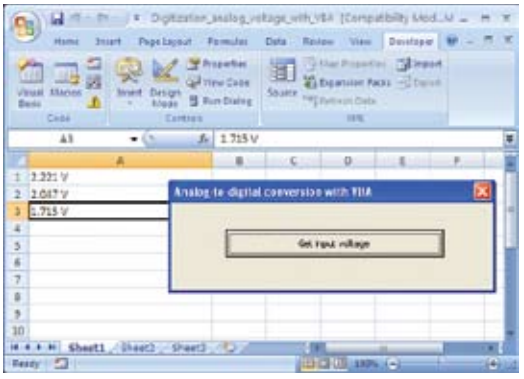


Figure 3 The VBA application collects data in an Excel spreadsheet.

takes the `port` parameter where the address of the port is held. When successful, this function returns the byte read from this port.

The program realizes the timing diagram of the conversion and outputs data when the `CommandButton1_Click` event handler is called. The statements

```
Call Out32(dataPort,
&H80)
Call Out32(dataPort,
&HC0)
Call Out32(dataPort,
&H40)
```

start the conversion. The `for()` loop passes data bits beginning from the MSB (most-significant bit) to the parallel port. The 12 bits of the result are saved in the `binData` variable, and the total variable holds the final result. **Figure 3** shows the main window of the running application. After you click the "get input voltage" button, the measurement result appears in the next cell in Column A of Spreadsheet 1. EDN

Schottky diodes improve comparator's transient response

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

In a previous Design Idea, a circuit switches precision dc reference voltages to the noninverting input of a high-speed IC comparator (**Reference 1**). The circuit uses a 2-to-1 multiplexer that functions in a BBM (break-before-make) fashion. Multiplexers have a parasitic capacitance, whose injection of charge, Q_{D1INJ}, into the D1 drain electrode of the multiplexer might cause error voltages at the comparator's reference input (**Figure 1**). The following **equation** defines the approximate peak error voltage:

$$\Delta V_{D1+} \simeq \frac{Q_{D1INJ+}}{C_{D1OFF} + C_{IN}},$$

where C_{D1OFF} is the capacitance of the D1 terminal of IC_2, and C_{IN}, approximately 1 pF, is the input capacitance of the comparator. IC_2 has a BBM interval of approximately 3 nsec, and any Analog Devices (www.analog.com) ADCMP608 and ADCMP609 comparators' signal-propagation delays are 10 times that value. Hence, they cannot change their state in 3 nsec. Analog Devices has 10-times-faster parts, the ADCMP601 and ADCMP602, in the same family. Unfortunately, these devices can sense these spikes (**Reference 2**). The voltage excursion over the high-level reference voltage, V_{REFH}, can result in an abrupt short-term elevation of the output voltage.

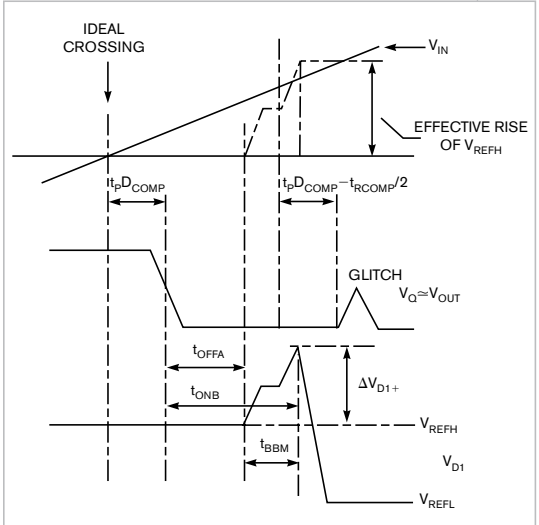


Figure 1 Charge injection can produce a glitch in a voltage comparator's output, and these glitches can cause logic hazards.

When the high reference voltage exceeds the voltage at the comparator's inverting input, its output goes high again, or generates a glitch. As the comparator crosses an ideal level, the comparator's output goes low with a delay. The positive error voltage starts to evolve with an additional delay of the

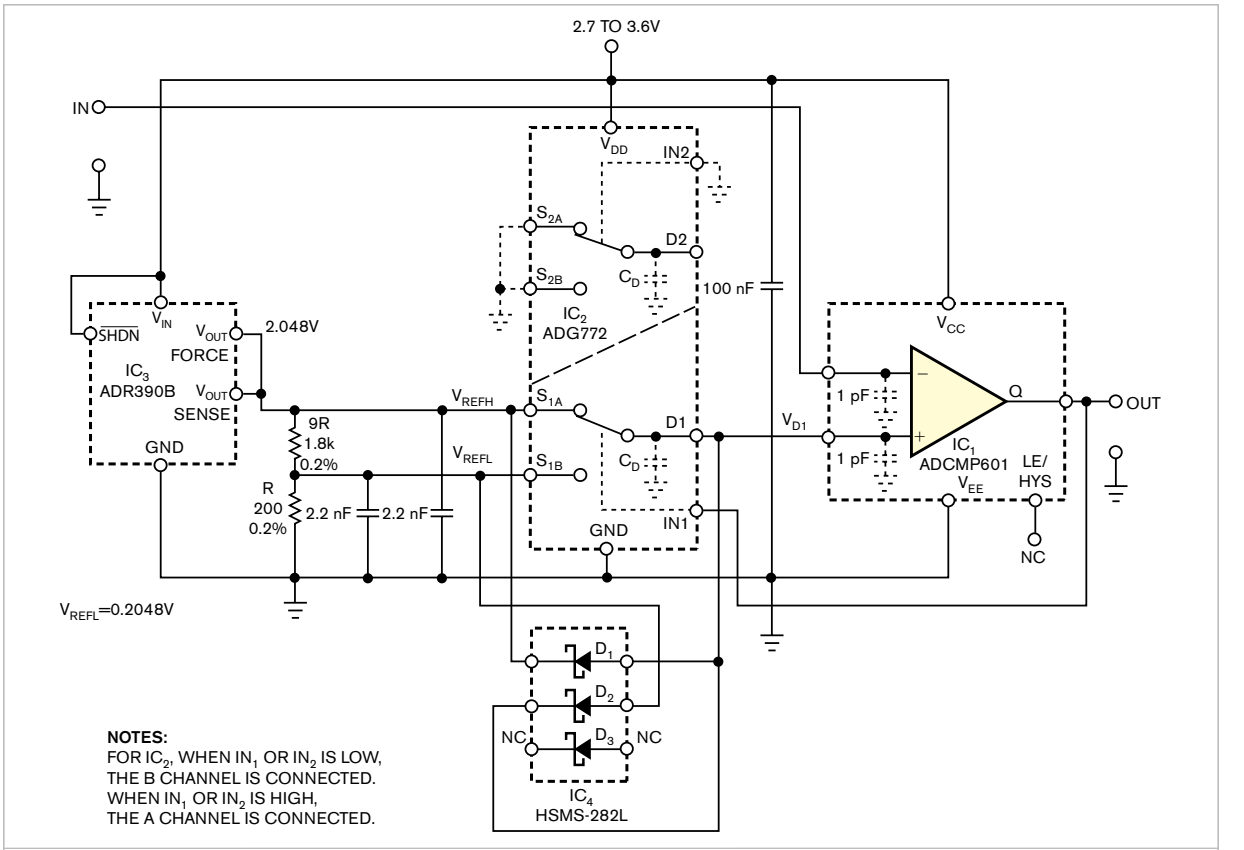


Figure 2 Schottky barrier diode D_1 damps positive voltage spikes that occur at the noninverting input of comparator IC_1 shortly after the high-to-low transition of the output voltage. Diode D_2 clamps the negative excursion of the reference voltage during a low-to-high transition of the output voltage.

turn-off time of Channel A of IC_2. The ADCMP601's data sheet shows a charge injection into the source electrode. In this case, however, charge injection to the drain electrode plays a role. As a rough estimate, you can use the data on charge injection in **Reference 2**. As the charge injection changes its sign, depending on the value of the common-mode voltage, the charge is about 0.8 pC at V_{D1}, which equals the high-level reference voltage, and it is approximately −0.3 pC when the reference voltage is low. This voltage ceases with the delay of the turn-on time of the channels in analog switch IC_2, an ADG772 (**Reference 3**). The inadvertent return of the comparator's output to high, although lasting no more than 5 nsec, might cause some logic circuits to respond unexpectedly. A similar situation might occur when the input ramp with a negative slope crosses the lower reference voltage.

To prevent these hazardous states, you can add two Schottky barrier diodes (**Figure 2**). Diode D_1 starts to conduct whenever a positive voltage excursion of approximately 200 mV above the higher reference voltage occurs at the D1 terminal, whereas D_2 conducts at excursions of 200 mV below the lower reference voltage. Further, the nonlinear junction capacitance of these diodes rises with forward voltage from 0.7 pF at a forward voltage of 0V to approximately 1.05 pF at a forward voltage of 100 mV. At input voltages close to the higher reference voltage, diode D_2 is reverse-biased by almost 2V, and its capacitance drops to approximately $2/3C_D(0)$, where C_D is the capacitance of the nonbiased diode. Diodes D_1 and D_2 have a combined capacitance of 1.5 pF. This amount can increase the input capacitance. Because the charge injection doesn't exceed 0.8 pC, the voltage excursions at the D1 terminal are less than 160 mV. Diodes D_1 and D_2 reside within diode trio IC_4, an Avago Technologies (www.avagotech.com) HSMS-282L (**Reference 4**), but you can also use a diode-pair IC. EDN

REFERENCES

1 Štofka, Marián, "Circuit uses two reference voltages to improve hysteresis accuracy," *EDN*, Jan 7, 2010, pg 43, www.edn.com/article/CA6713705.

2 "Rail-to-Rail, Very Fast, 2.5 V to 5.5 V, Single-Supply TTL/CMOS Comparators ADCMP600/ADCMP601/ADCMP602," Analog Devices, 2006, www.analog.com/static/imported-files/data_sheets/ADCMP600_601_602.pdf.

3 "ADG772 CMOS Low Power Dual 2:1 Mux/Demux USB 2.0 (480 Mbps)/USB 1.1 (12 Mbps)," Analog Devices, 2007–2008, www.analog.com/en/switchesmultiplexers/analog-switches/ADG772/products/product.html.

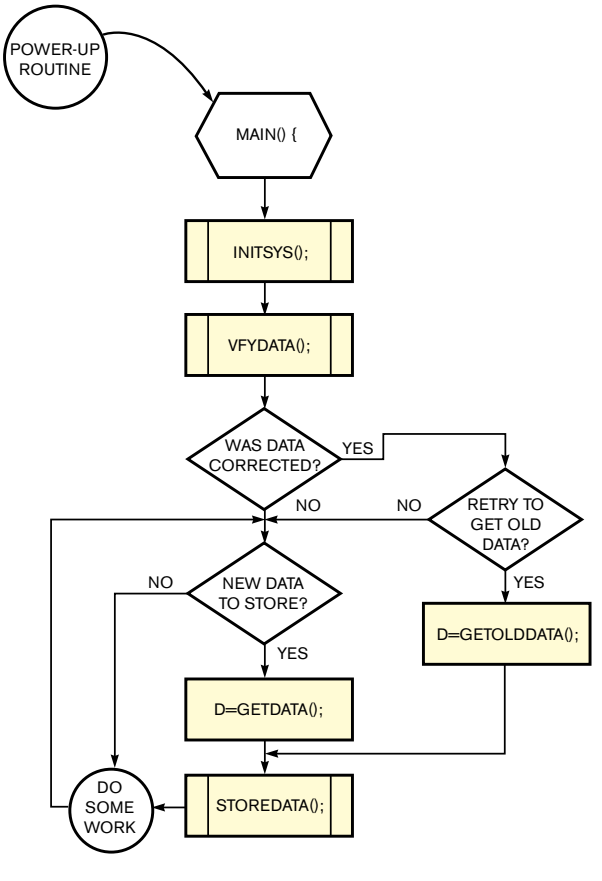
4 "HSMS-282x Surface Mount RF Schottky Barrier Diodes," Avago Technologies, May 28, 2009, www.avagotech.com/docs/AV02-1320EN.

Algorithm keeps data safe

Luis G Uribe C, Caracas, Venezuela

Many embedded systems must regularly update multibyte data to EEPROM, flash memory, or a database server. This Design Idea presents a robust algorithm for this process that prevents data losses and inconsistencies due to program interruptions or power failures. You can avoid data losses by maintaining two separate memory areas, duplicating critical variables in each. Memory can be battery-backed RAM, magnetic disks, flash memory, or local or remote storage subsystems. You can use a simple FSM (finite-state machine) that uses three or four states with appropriate Gray coding to track the status of each main variable and its mirrored value in the storage devices. The states run in a sequence that the software driver can use to write data to both the main and the backup variables. The driver sets and resets two variables, B0 and B1, as status bits. B0 is the main variable status bit, and B1 is a mirror status bit. Both bits are recorded in the same storage medium as the data.

This algorithm prevents race conditions and ambiguous situations. You can enter the power-up verification routine at any time, interrupting the main flow of the program without losing any data. **Figure 1** shows the flow of the algorithm, which begins with a power-up routine. **Table 1** identifies and defines each of the possible states for the variables. EDN



NOTE: TRYING TO RETRIEVE OLD DATA THAT WAS NOT STORED IS HIGHLY DEPENDENT ON THE APPLICATION.

Figure 1 This algorithm starts with a power-up routine and then verifies that the data is properly stored. (For other algorithms, go to www.edn.com/100121dib.)

TABLE 1 POSSIBLE STATES OF VARIABLES

Bit state		Description
B0	B1	
0	0	Both variables are in steady state.
		If the power-up routine finds this state, it may assume that everything is fine; both values are properly stored in the main site and in the backup. Your program may read and use the data as needed.
		When it's time to modify a value, the driver sets a one at B0 to indicate a zero/one.
0	1	Data begins to write to the main variable.
		If the power-up routine finds this state upon verification, you can assume that the main variable is corrupt and decide how to update it from the mirroring site.
		When the driver finishes updating the main variable, it sets a one at B1 and keeps B0 untouched at one to indicate two ones.
1	1	Data has written properly to the main variable, and the driver soon begins the next state to duplicate main data into its backup storage.
		This state is transient to avoid changing bit values between zero/one and one/zero at once. Using this sequence, the two data repositories may be in different local or remote physical subsystems.
		If the power-up routine finds this state, it may assume that the mirror variable is corrupt, and it will proceed (next state) to update the mirror site from the main location.
		At any convenient time from now on, the finite-state machine may leave this state and enter the next one, one/zero, which means:
1	0	Data has properly written to the main variable state, and the driver now begins to duplicate the main data into backup storage (similar meaning to previous state, one/one).
		If the power-up routine finds this state, the verification routine may assume that the mirror variable is corrupt and update the mirror site from the main variable.
		When the driver finishes updating the mirroring variable, it clears B1, and the finite-state machine reverts to the first state, two zeros, in the sequence.

P-channel power-MOSFET driver uses unity-gain op amp

Suded Emmanuel, Emmanuel's Controls, Auckland, NZ

P-channel MOSFETs can simplify designs when you use them as high-side switches on circuits with voltages exceeding 100V dc. When driving a MOSFET, you must rapidly charge and discharge the input capacitance between its gate and its source to reduce heat losses. The circuit in **Figure 1** can accomplish that task. Q_7, an International Rectifier (www.irf.com) IRF5305 power P-channel MOSFET, switches 50V to a load. A series of pulses from a pulse generator or PWM (pulse-with-modulation) source drives the load at frequencies as high as 60 kHz with a variable duty cycle. The circuit comprises Q_4, R_5, D_2, R_4, D_3, and R_3; provides a means of level-shifting; and ensures that the voltage drop between the gate-to-source voltage of Q_7 never exceeds 10V. When Q_4 is on, 10V develops across D_3. This voltage drop turns on Q_7 through op amp IC_{1A}, one-half of an MC33072 from On Semiconductor (www.onsemi.com). IC_{1A} has a 13V/µsec slew rate and can drive capacitances as high as 10 nF.

The combination of D_4, R_1, Q_1, Q_2, R_2, and C_1 provides "ground" for the op amp, which is at 38V—that is, 12V below the 50V rail voltage. The positive voltage is 50V, and ground is 38V. The anode of D_3 connects to the noninverting input of IC_{1A}, whose output drives Q_7's gate at 40V, which is 10V below the rail voltage of 50V. The circuit comprising R_6, Q_5, D_1, R_7, R_8, Q_6, R_9, R_{10}, and Q_3 rapidly switches D_3's anode to 50V, which turns off Q_7. Transistor Q_5 functions as an inverter that turns on Q_6, which subsequently drives Q_3 to rapidly switch D_3's anode to 50V and thus drives Q_7's gate. Schottky diodes D_1 and D_2 alternately enhance the switching speed of Q_5 and Q_4.

Unity-gain op amp IC_{1A}, with its high slew rate, fast settling, capacitive-driving capability, and feedback of the gate voltage, enhances Q_7's switching speed. Using this circuit, you can achieve a rise time and fall time of approximately 500 nsec at Q_7's output. EDN

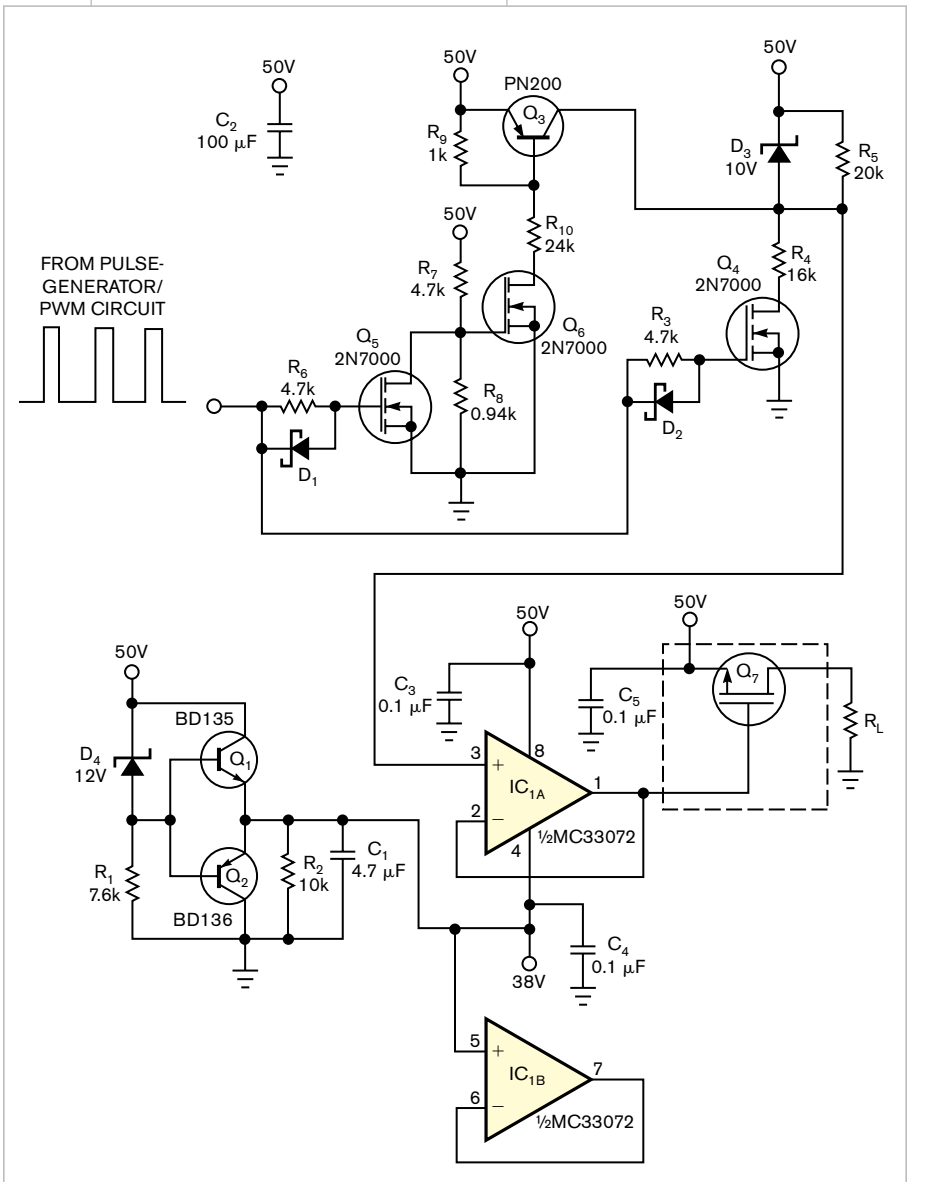


Figure 1 An op amp operating at 38 to 50V provides power to a load through power-MOSFET Q_7.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Spice simulators provide behavioral sources to mere mortals

William H Morong, Paoli, PA

Spice simulators include three types of tools for voltage and current sources: independent, dependent, and behavioral. Independent sources have two terminals and provide a specified number of volts or amperes you enter as a constant, just as a bench supply generates a set voltage or current. These simulators, like signal generators, also provide waveforms.

Figure 1a shows two independent sources. Voltage source V_1 provides 5V dc, and V_2 provides a 1V-peak sine wave. Dependent sources have at least four terminals. A first terminal pair is a voltage or current output. Another terminal pair is an input that attaches in shunt or in series with two circuit nodes. The output responds to the voltage across or current through the input nodes. Some dependent sources have multiple input-terminal pairs. The output responds to the inputs according to some linear mathematical rule. For example, a voltage-dependent voltage source set to the constant 100, such as E_1 in **Figure 1a**, is an ideal voltage amplifier with a gain of 100.

Behavioral, or arbitrary, sources are the least-used but most powerful of these sources. They have only an output-terminal pair, but they are more powerful than their simpler counterparts. They can implement a set of mathematical functions that roughly correspond to those available on scientific calculators. Their outputs can be independent or dependent. In the simplest case, a behavioral voltage source set to V=5 produces 5V, just as a power supply does.

A behavioral source can also respond to a designated voltage or current somewhere in the simulated circuit. In **Figure 1b**, B_1 responds to the voltage at the input node. Applying

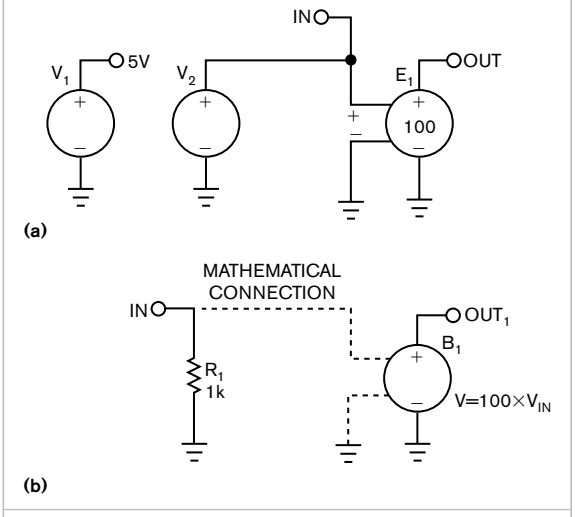


Figure 1 Independent voltage source V_1 provides 5V dc, and V_2 provides a 1V-peak sine wave (a). Behavioral source B_1 responds to the voltage at the input node (b).

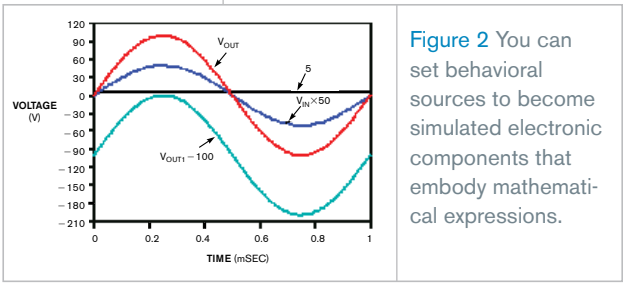


Figure 2 You can set behavioral sources to become simulated electronic components that embody mathematical expressions.

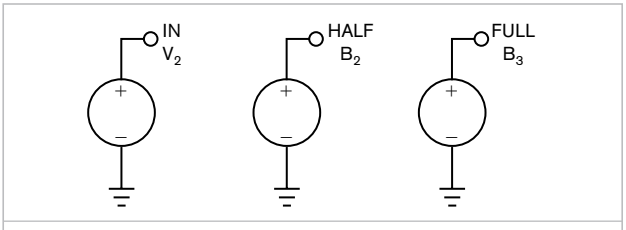


Figure 3 Behavioral source B_2 amplifies and half-wave-rectifies the voltage at the input node. Behavioral source B_3 also amplifies and full-wave-rectifies the same voltage.

the **equation** beside B_1 sets the behavioral source to $V=100\times V_{IN}$ and amplifies the voltage across R_1 by 100V.

In this case, the behavioral source behaves as the dependent source when the input node and ground drive it, but the behavioral source's input connection is purely mathematical. **Figure 2** shows several possible plots for output signals. You can set behavioral sources to become simulated electronic components that embody mathematical expressions. In the preceding example, if you set the behavioral source to $V=uramp(100\times V_{IN})$, the amplified output is ideally half-wave-rectified. If you substitute the abs function for uramp, you get full-wave rectification.

In **Figure 3**, behavioral source B_2 amplifies and half-wave-rectifies the voltage at the input node. Behavioral source B_3 also amplifies and full-wave-rectifies the same voltage. **Figure 4** shows the results. Engineers commonly generate ABMs (analog behavioral models) for Spice simulations to represent entire blocks of analog functions. Sometimes you need simple models when none are available, but you can use behavioral models to solve those problems. For example, you can set $I=uramp(V_{FB}-1.25)$ as a behavioral current source to draw 1 mA/mV at the FB (feedback) node that exceeds 1.25V (**Figure 5**). The behavioral source looks like an idealized three-terminal shunt regulator (**Figure 6**). If you need more refinement, you can set the feedback node to act as an RC filter or another function.

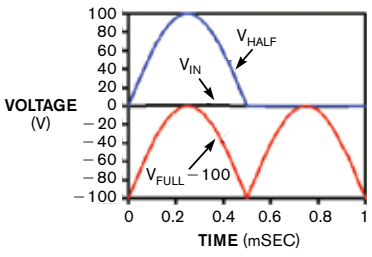


Figure 4 Behavioral source B_2's amplification and half-wave-rectification of the voltage at the input node and B_3's amplification and full-wave rectification of the same voltage yield these results.

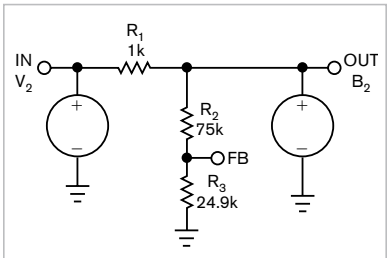


Figure 6 The behavioral source looks like an idealized three-terminal shunt regulator.

Because behavioral sources can embody mathematical functions, you can use them to test the mathematical viability of a design before simulating the circuit. Long mathematical expressions can be difficult to read, so you should break the idea into small blocks and set a behavioral source to implement each block. Just because your idea works in math, though, doesn't mean that its behavioral model is a practical circuit. In Spice, it is easy to make a model of a current source with 1 MV of near-perfect compliance, sometimes without intending to do so, but most practical engineers would rather defer the realization of such a circuit.

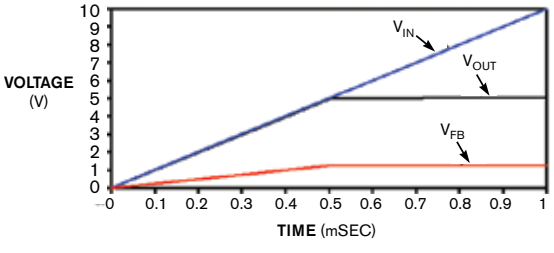


Figure 5 You can set $I=uramp(V_{FB}-1.25)$ as a behavioral current source to draw 1 mA/mV at the FB (feedback) node that exceeds 1.25V.

The caution to be realistic applies to Spice in general and to behavioral sources in particular. When you draw two equal resistors in your Spice schematic, they will be perfectly matched in simulation. The resistors in your stockroom are not that good. EDN

Multidecade BCD DAC uses resistors of only six values

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

A previous Design Idea uses a three-decade BCD (binary-coded-decimal) DAC to precisely set the output current of a current source (**Reference 1**). The circuit acts as a code-to-conductivity converter. The values of resistors of this DAC are staggered by powers of two within any of the decades, and the values of resistors at corresponding bits of the decades are staggered by powers of 10. Thus, the circuit needs 12 values of resistors, ranging from 125Ω to 100 kΩ.

In comparison, the circuit in this Design Idea enables you to construct a BCD DAC using only six resistor values, regardless of the number of decades. Moreover, these six resistor values vary within a relatively narrow 1:8 range. The voltage-output DAC operates ratiometrically. That is, if the temperature coefficients of the resistors are approximately the same—and you can assume they will be within this narrow range of values—then the variation of resistance with temperature has almost no detrimental effect on accuracy. This situation is not true, however, for code-to-conductivity DACs, in which the temperature coefficient of the resistors directly influences the temperature coefficient of the DAC.

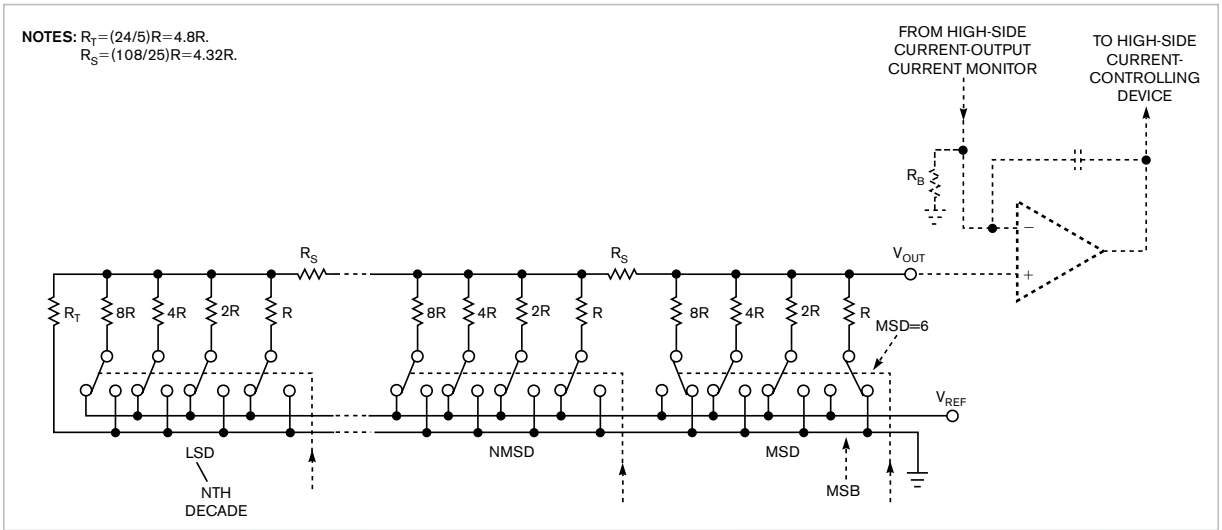


Figure 1 A voltage-output BCD DAC needs just six precision-resistor values. Resistors connect either to ground or to the reference voltage.

Figure 1 shows the voltage-output BCD DAC. The values of resistors are staggered by powers of two within each decade. The values of resistors at corresponding bits of the decades are of equal values. The switched ends of the resistors connect to either ground at logic zero or to the reference voltage at logic one. The voltage-output DAC thus has an advantage in that all the resistors' ends are at a defined potential. In a code-to-conductivity converter, on the other hand, one end of the resistor remains open at logic zero, and these open ends might act as capacitive sensors or even antennas, which could introduce additional errors. The common ends of four resistors in four bits of the MSD (most-significant decade) form the output. The common outputs of resistor quads in the less-significant decades successively connect to the main output through the series resistors, R_S, which all have the same value. Thus, R_S=(108/25)R=4.32R, where R is the value of the resistor at the MSB (most-significant bit) of any of the decades.

The common ends of bit resistors in the LSD (least-significant decade) connect to ground through terminating resistor R_T. This resistor represents the theoretically infinite number of decades having weight lower than the actual LSD, whereas these hypothetical decades are all set to zero. Thus, they contribute no voltage at the output. They do, however, influence the properties of the resistive network. R_T sets this influence and is equal to (24/5)R, or 4.8R. The full-scale output of the voltage-output BCD DAC is $3/5\times(1-10^{-N})V_{REF}$, where N is the number of decades—in this case, three.

To exploit a voltage-output BCD DAC in a single-supply programmable-current source, connect the output of the voltage-output BCD DAC to the noninverting input of an op amp, which accepts input voltages as low as 0V. The inverting input of the op amp connects to ground through resistor R_B, which has the value of $(3/5)\times(V_{REF}/10^{-2}A)$. EDN

REFERENCE

1 Diószegi, Gyula, and János Nagy, "Current-sense monitor and MOSFET boost output current," *EDN*, May 28, 2009, pg 44, www.edn.com/article/CA6659409.

Converter translates Bayer raw data to RGB format

Yu-Chieh Chen and Tai-Shan Liao,
National Applied Research Laboratories, Hsinchu, Taiwan

CMOS image sensors include the color filters of an RGB (red/green/blue) Bayer array, which lets the sensor detect colors. The image data, the output from the image sensor, is Bayer raw data (**Figure 1**). Unfortunately, most consumer-grade image-displaying devices require an RGB-image-data format with red, green, and blue in each pixel's data. Therefore, you often need a Bayer-raw-data-to-RGB converter between an image sensor and a displaying device. This Design Idea describes such a converter in Verilog HDL (hardware-description language). You can implement the code, available from the online version of this Design Idea at www.edn.com/100204dia, into a CPLD or an FPGA.

To make the design easy to understand, the RGB data is only 24 bits deep. A 320×240-pixel test-bench pattern verifies the design (**Figure 2**). The image data for red, green, and blue are 88h, 66h, and 22h, respectively. **Figure 3** shows the timing of the converter, and **Figure 4** shows the flow

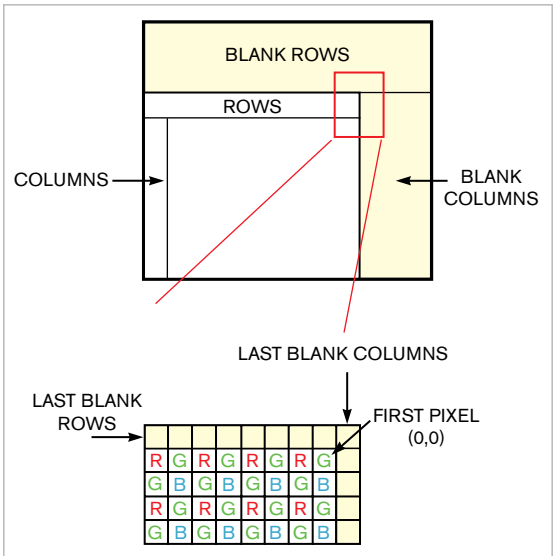


Figure 1 The Bayer raw data from an image sensor represents a 320×240-pixel array.

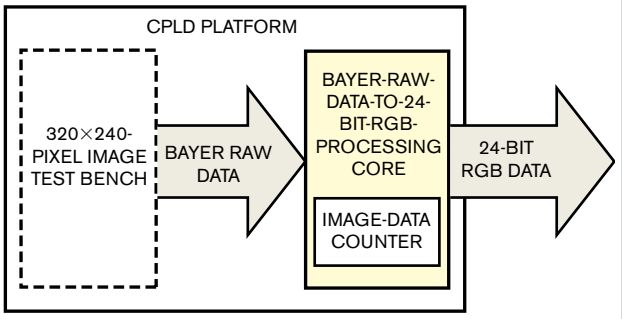


Figure 2 A CPLD processes Bayer raw data into 24-bit RGB data.

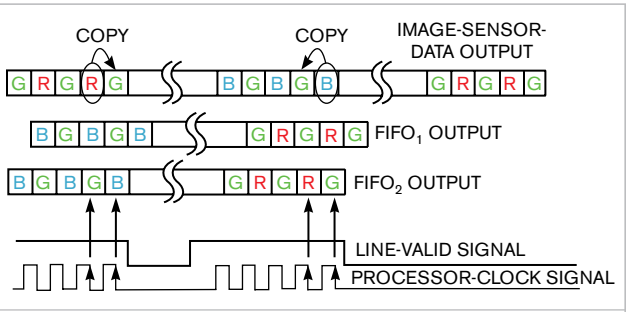


Figure 3 Rising edges of a clock latch Bayer raw data from FIFOs into RGB data.

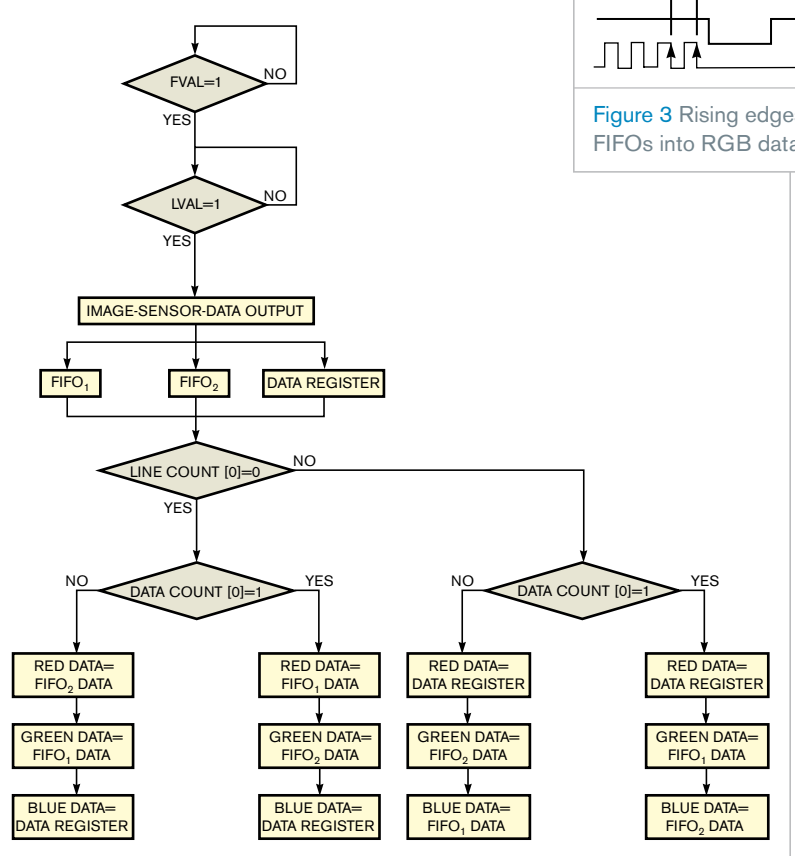


Figure 4 RGB data represents red, green, and blue for display.

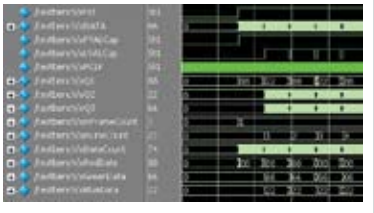


Figure 5 The values 88h, 66h, and 22h, respectively, represent red, green, and blue data.

chart. In **Figure 3**, the PCLK (processor-clock) signal's rising edge latches the 24-bit RGB data. The LVAL (line-valid) cap signal is the synchronized line-valid signal for reading the data. The process begins when the FVAL (frame-valid)-data signal goes high. When that action occurs, sensor data begins writing to $FIFO_1$ (first-in/first-out), $FIFO_2$, and the data register. After saving the data, $FIFO_1$ reads out the data. $FIFO_2$ and the data register read their data at the same time. Those readouts occur one clock cycle

after $FIFO_1$ starts reading. The next 24-bit RGB data remains the same in the data register, and it combines the data from $FIFO_1$ and $FIFO_2$, which read out at the same rising edge of the clock. The line-count signal shows whether the data is even or odd, which influences the combinational sequences of the data that reads out from $FIFO_1$, $FIFO_2$, and the data register.

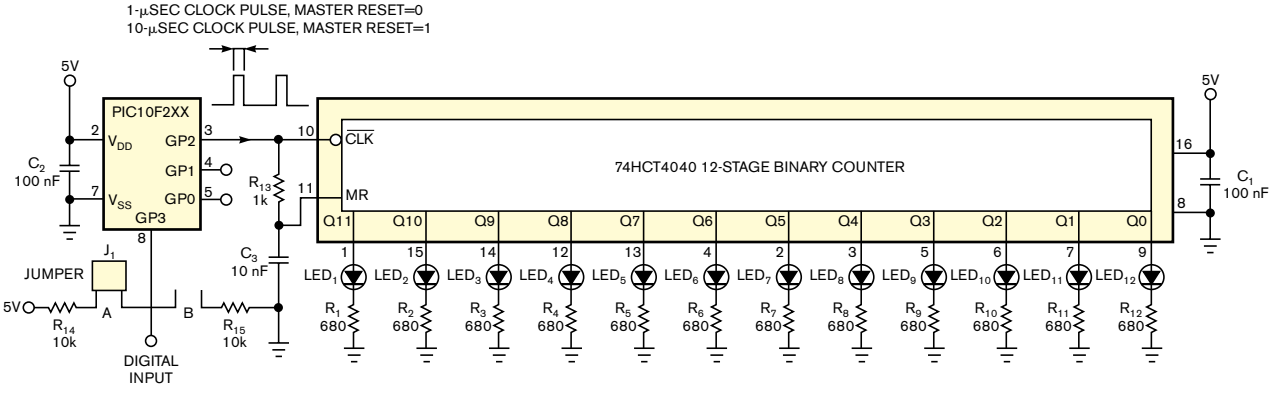
Figure 5 is the ModelSim simulation waveform of the converter. The 24-bit RGB output-data values 88h, 66h, and 22h are red, green, and blue data, respectively. The **figure** shows the 24-bit RGB data as having red, green, and blue values of 88h, 66h, and 22h, respectively, during every line-data period. The line-data period matches the default pixel value in the image data's test-bench pattern. EDN

Drive 12 LEDs with one I/O line

Charaf Laissoub, Maisons Alfort, France

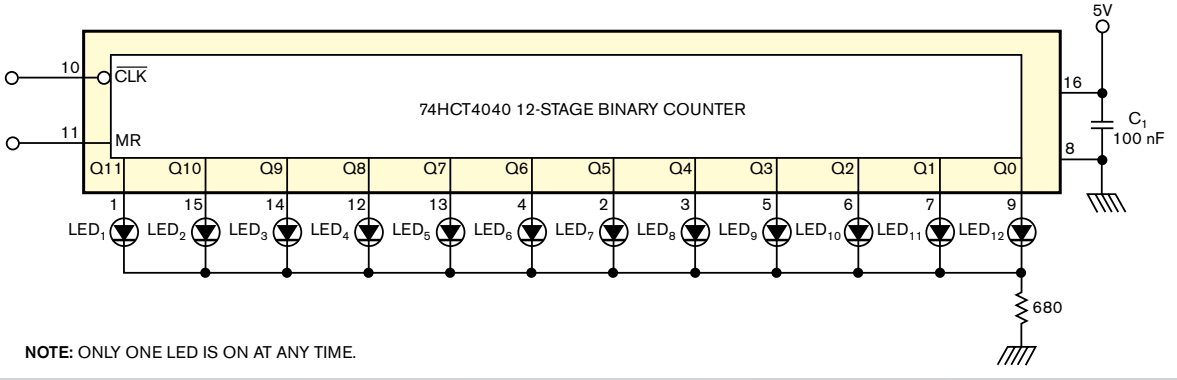
Several Design Ideas expand the I/O of a pin-limited microcontroller (**references 1** through **4**). The circuit in this Design Idea uses an alternative method (**Figure 1**). It limits external additional parts to one IC, and it drives as many as 12 LEDs in dot-bar or bar-graph mode. You can use the same technique in a dot-bar design (**Figure 2**). If you need a seven-segment LED display, you can use the circuit in **Figure 3**, which shows how to rearrange the circuit according to a classic multiplexed, four-digit common-cathode display. The prototype display uses Kingbright's (www.kingbright-led.com) SC52-11EWA high-efficiency LEDs, which emit 2000 to 5600 µcd at a forward current of 10 mA. The driver is a 12-stage NXP (www.nxp.com) 74HCT4040 binary counter or a 74HC4040 version for a lower power supply.

Listing 1, which you can download at the online version of this Design Idea at www.edn.com/100204dib, contains an assembly-language routine. It generates a precise quantity, Q, of high-frequency pulses, which deliver the number, N, that the outputs of the 74HCT4040 require. The relations are $Q=2^{N-1}$ in dot-bar mode and $Q=2^N-1$ in bar-graph mode. **List-**



NOTE: BEFORE POWER-ON, PUT JUMPER INTO A FOR BAR-GRAPH MODE OR PUT JUMPER INTO B FOR DOT-BAR MODE.

Figure 1 A 12-LED dot-bar/bar-graph display uses a 12-stage binary counter.



NOTE: ONLY ONE LED IS ON AT ANY TIME.

Figure 2 Using one current-limiting resistor can reduce parts count.

ing 2, which is also available at www.edn.com/100204dib, is a full example of using this routine with Microchip's (www.microchip.com) PIC10F202, a member of the PIC10F series, which is the company's most pin-limited microcontroller family.

Although the PIC's internal unique clock frequency is 4 MHz, you'll notice little flicker effect. You can reduce the flicker by using a midrange pin-limited PIC microcontroller, such as the PIC12F629, which has an internal clock frequency of 20 MHz. **Listing 3**, also available at www.edn.com/100204dib, uses a look-up table to convert the desired number into seven-segment code to replace the 12 LEDs with a four-digit display. EDN

REFERENCES

1 Anonymous, "Microcontroller provides low-cost analog-to-digital conversion, drives seven-segment displays," *EDN*, May 10, 2007, pg 80, www.edn.com/article/CA6437954.

2 Raynus, Abel, "Squeeze extra outputs from a pin-limited microcontroller," *EDN*, Aug 4, 2005, pg 96, www.edn.com/article/CA629311.

3 Jayapal, R, PhD, "Microcontroller's single I/O-port line drives a bar-graph display," *EDN*, July 6, 2006, p 90, www.edn.com/article/CA6347254.

4 Lekic, Nedjeljko, and Zoran Mijanovic, "Three microcontroller ports drive 12 LEDs," *EDN*, Dec 15, 2006, pg 67, www.edn.com/article/CA6399101.

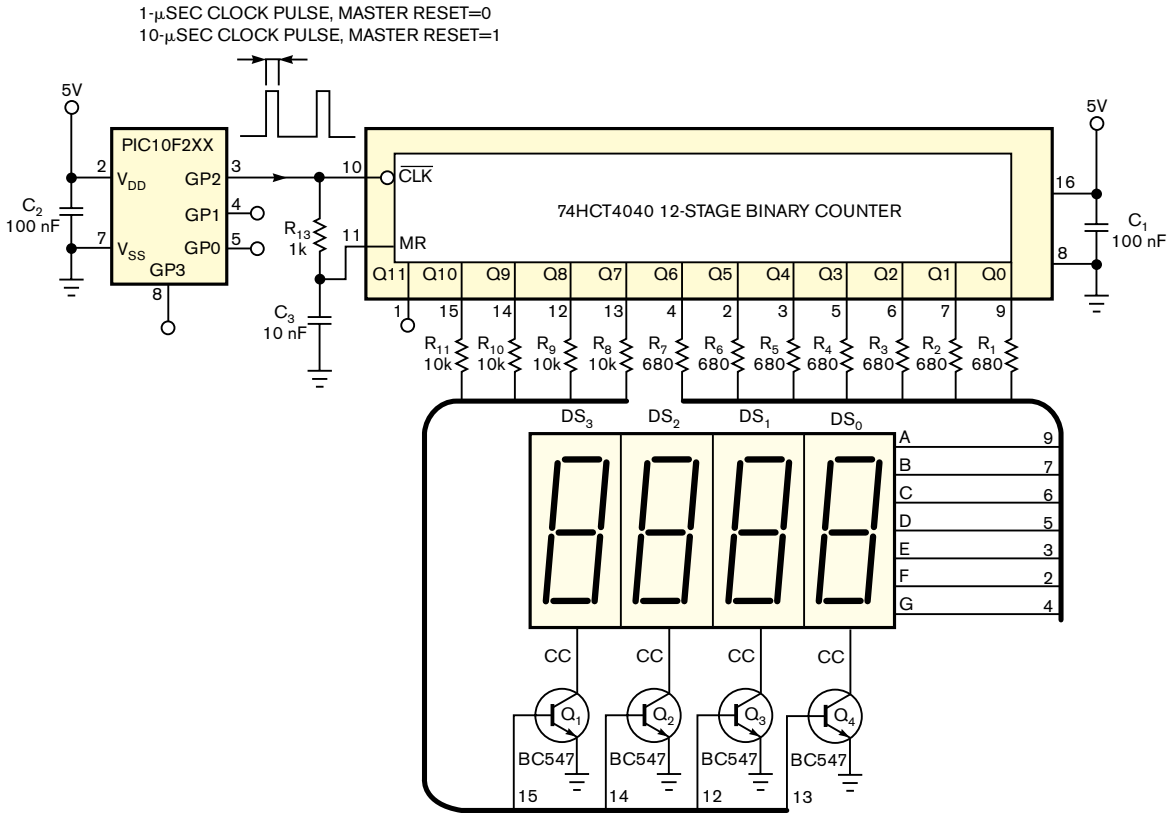


Figure 3 The same technique works with seven-segment displays.

Circuit precludes common-mode conduction

Ken Herrick, Oakland, CA

When driving an H-bridge or a similar circuit, you usually must ensure that two or more transistors are not on at the same time. Eliminating multiple transistors from turning on reduces power consumption and lowers EMI (electromagnetic interference). Crossover-delay circuits solve that problem. **Figure 1** shows a simple, two-phase design that lets you adjust the crossover delays equally by changing the value of one component with minimal phase delay.

Each Schmitt trigger inverter is driven during one half-cycle through

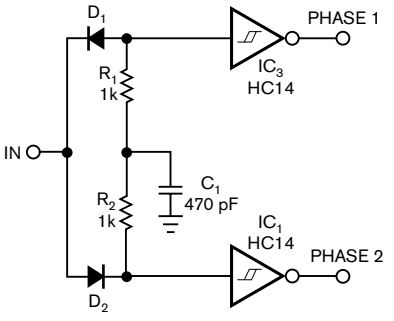


Figure 1 Each Schmitt trigger inverter is driven during one half-cycle through a diode. The RC delay occurs during the alternate half-cycle. Equal-value resistors R_1 and R_2 serve alternatively as delay elements and gate-coupling resistors.

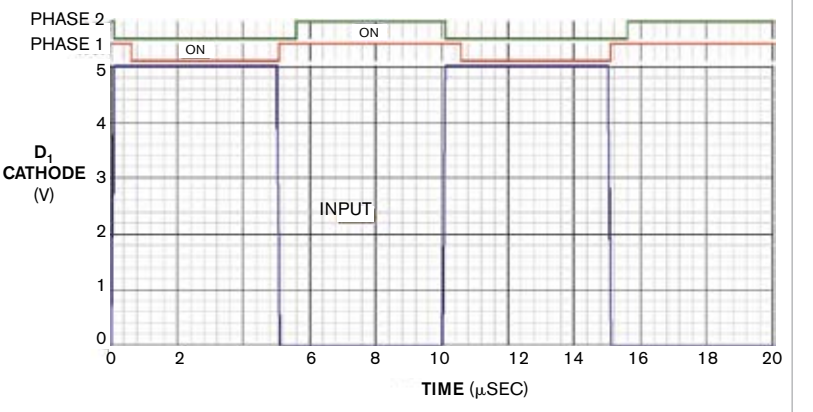


Figure 2 For the two out-of-phase half-cycles, leading edges are delayed equally with respect to the input transition, and trailing edges are coincident with the transition within about one gate delay.

a diode. The RC delay occurs during the alternate half-cycle. Equal-value resistors R_1 and R_2 serve alternatively as delay elements and gate-coupling resistors. The waveform in **Figure 2** shows the result. For the two out-of-phase half-cycles, leading edges are delayed equally with respect to the input transition, and trailing edges are coincident with the transition within about one gate delay. If you need equal polarity "on" half-cycles, insert an inverter in one of the two phase outputs. Alternatively, if biphase drivers, such as those for driving coupling transformers, will follow this circuit, merely interchange the outputs of one of those drivers to effect the inversion. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas
READERS SOLVE DESIGN PROBLEMS

Use eight timers with PIC16Fxxx microcontrollers

Luis G Uribe C, Caracas, Venezuela

The need for timing in embedded programming often exceeds the small number of available hardware timers in microcontrollers. For example, the Microchip (www.microchip.com) PIC16F84A has one timer, but you can create as many as eight timers with the Timers8.inc assembly code, which you can download from the online version of this Design Idea at www.edn.com/100218dia.

Often, you need a timer while waiting for some expected lapse to expire, which blocks your program until that time has elapsed. To accomplish that task, you can use a simple routine, such as `Wait_on 2, .30`, meaning that Timer 2 counts 300 msec. The time value is .30 ticks of 10 msec each. In this way, you can debounce input data (**Listing 1**).

Using Microchip's MPLab assembler, you can't guess whether a parameter is a constant value or a variable, so you need two macros, one for each kind of circumstance. In the following explanation, use K if time is a constant and use V for variables, such as Wait_onK or Wait_onV. The same library for the TBM (timebase module) on the MCHC9S08GP32 microcontroller from Freescale Semiconductor (www.freescale.com) uses only one macro, Wait_on, under the CodeWarrior assembler. The function deals with both constants and variables.

The "tmr" always stands for a constant from zero to seven. It designates the number of the timer you apply to a situation. You must always multiply time by 10, so .1 is 10 msec. You can therefore use it with times from 10 to 2560 msec. Precision is plus or minus one tick. If you need to trigger an event with a 20-msec timer, you may end with 10 msec. So use 30 msec to be safe. At the high end of the scale, you will have 2560±1 msec, which is acceptable.

Timers8.inc programs the TMR0 on the PIC16F84A. You can extend beyond eight timers or use 16-bit variables, but remember that the PIC16F84A has only 68 bytes of RAM.

On some occasions, you need to start a timer but don't need to block your program to time-out; in this case, use the `Setimer 7, .20` macro. You start Timer 7 to last 200 msec; doing so does not block your program. Whenever you need to know the status of your timers, test them using the TimeOut macro, `TimeOut 2, Two_done`, meaning that, if Timer 2 has expired, you will go to the "Two_done" label. Any of your main code labels will fit. Otherwise, your program will continue executing the next instruction in sequence.

Setimer comes in two versions. `SetimerK 2, .20` sets Timer 2 to count 200 msec, using Constant `time .20`, and `SetimerV 5, var` sets Timer 5 to count, for example, 300 msec, using variable time var, which you should have previously loaded with .30.

You may need to employ timers in ISRs (interrupt-service routines)—for example, to debounce the interrupt pin. This situation is awkward because the routine to serve the external INT pin runs with general interrupts disabled, as usually happens in the PIC16F84A, but timer routines require you to enable interrupts. This microcontroller architecture makes it difficult to enable interrupts in ISRs. You may, however, use either `ISRWait_onK` or `ISRWait_onV` to accomplish your purpose, as in `ISRWait_onK 7, 3`.

This approach works in a similar way

LISTING 1 CODE FOR DEBOUNCING DATA

```
Loop1:
  btfss   PORTA, 0      ; Wait unitl PortA, Bit0, be equal to 1
  goto    Loop1         ; No? go to repeat test...
  Wait_onK 0, 3         ; Now, PortA, Bit0 = 1 so, set up Timer 0 for 30
                        ; ..mSec to debounce it
  btfss   PORTA, 0      ; Time out! Re-test PortA, Bit0 input condition;
  goto    Loop1         ; ..if PortA, Bit0, = 0, discard it and go to Loop1
                        ; ..to begin again
Cont:  ...              ; ..if PortA, Bit0, = 1 you are done
```

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

to its twin, the Wait_on macro, except that you can use the approach in any ISR—a nice added value for such an inexpensive microcontroller. Use it with care, however. Interrupt latency increases because you block the program in an ISR for several milliseconds with global interrupt disabled. If you choose to debounce your interrupt signal using programmed delays, you will probably encounter the same problem. If you use a specific timer number in the main program, don't use it in the ISR.

To use the Timers8.inc library, you must include the library file and define some variables outside the timer's code. To find the exact place to include the library and define variables, refer to the sample code. Look for <<< TMR0 <<<, which overemphasizes portions of the code. In particular, inspect the lines "CBLOCK" and "INCLUDE <timers8.inc>".

Follow this plan in your program: Use the macro Init8Timers to activate the hardware and set up the eight software timers. This macro defines eight variables, from Timer 0 to Timer 7, each using one unsigned byte. Each timer ticks once every 10 msec, covering a range of 10 to 2560 msec. You need not worry about these variables, though, because the macros will handle them. A 1-byte variable, TimerFlags, has bits that represent the ready state of timers zero through seven. You need not deal with this internal variable.

To initialize a timer from zero to seven, use the Setimer macro, as in `SetimerK 2, .20` (set Timer 2 to count 200 msec using a constant time of .20) or `SetimerV 5, var` (set Timer 5 to count 300 msec using a variable time of .30, which you previously stored in var). Setimer macros are not self-blocking; they initialize the software timers and continue. This feature comes in handy when you plan to loop, asking for several events to time out and do not need one of them to block you.

THE SOFTWARE IN THIS CODE AVOIDS TOUCHING OR UPDATING VARIABLES IF THE STATUS BIT IS 1.

To test whether one timer has expired, use the TimeOut macro after Setimer: `TimeOut 2, Two_on`. If Timer 2 has expired, go to Two_on; otherwise, execute the next instruction in the sequence. Wait_on combines these macros in one: `Wait_onK 2, .30`. Set Timer 2 to count 300 msec using a constant time of .30 and block until time-out. Alternatively, using `Wait_onK 5, var`, set Timer 5 to count 300 msec using a value of .30, which you previously stored in var. Wait_on macros are self-blocking; they initialize the software timers and wait until time elapses. You can use ISRWait_on in ISRs: `ISRWait_onK 6, .35`. Set Timer 6 to count 350 msec using a constant time of .35 and then block.

Alternatively, you can use `ISRWait_onV 5, var`. Set Timer 5 to count 2000 msec using a value of .200, which you previously stored in var. ISRWait_on macros are self-blocking. You can use them in ISRs to initialize the software timers and wait until time elapses. You must include an interrupt handler; see the IntHandler in **Listing 2**, which you can download from the online version of this Design Idea at www.edn.com/100218dia. The library also includes TMR0ISR, Timer 0's ISR, and the UpdTmr (update-timer) internal macro.

Each timer has a status bit, which helps when your variables have 16 bits, 24 bits, or more. When the driver detects that one multibyte timer variable reaches zero, it signals this situation by setting the timer's status bit. That action spares you several instructions when you need to later decide whether the timer is zero. You can also use these bits as semaphores. You may start a timer with Setimer, and the hardware may interrupt you in the middle of the start-up to update your data structures, causing lots of problems. The software in this code, however, avoids touching or updating variables if the status bit is 1. Setimer begins raising the status bit and then loads the variables. If Timer 0 interrupts, it does not interfere with your data because it skips the updating process if the status bit is on. When Setimer is done, it clears the status bit, and Timer 0's ISR will begin to update whenever a tick arrives.

This code doesn't stop a timer before a time-out because the need never arises. If Setimer uses zero as a value for the time, it lasts for 256 10-msec ticks. If you need a 1-msec tick, you can load Timer 0 with −0.125 instead of −0.39 and use a prescaler of 8 (b'00000010' in OPTION_REG) instead of 256 (b 00000111), which are the values this code uses. The exact time is 125×8=1000 µsec (1 msec). This approach provides a range of 1 to 256 msec. EDN

Tilt/fall detector has staggered thresholds

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

Measurement-and-control applications may require action based on two distinct voltage levels. Crossing a threshold can produce a warning indication, whereas reaching a higher threshold may initiate emergency action, such as a system shutdown. In a fall-detector application, an apparent decrease in gravity below a lower threshold might be a controlled displacement, but a further decrease below a second threshold might indicate an uncontrolled fall.

The circuit in **Figure 1** uses a voltage divider to generate two reference voltages. Comparators and Schmitt-trigger-input NAND gates let you create two digital signals based on using reference voltages V_{REFA} and V_{REFB}. The sample circuit drives two LEDs, but

you can use the digital signals to drive transistors or relays, as well.

The voltage divider comprising R_S, R_A, and R_B sets the voltages for comparing the Z-axis output of an Analog Devices (www.analog.com) ADXL335 accelerometer (**Reference 1**). The higher reference voltage, V_{REFA}, corresponds to the lower-threshold tilt angle, where $\alpha_{TA}=45°$. The lower reference, with respect to the midvoltage supply minus V_{REFB}, corresponds to the upper-threshold tilt angle, where $\alpha_{TB}=60°$. If you choose a value of 100 kΩ for R_S, then you can calculate R_A+R_B:

$$\frac{R_S}{R_A+R_B}=\frac{V_S/2}{V_{GZ}\times\cos\alpha_{TA}}-1.$$

The Z-axis voltage, $V_{GZ}=300$ mV, occurs when the accelerometer's Z axis is oriented vertically. From the obtained value of R_A+R_B, you can calculate R_B:

$$R_B=\frac{\cos\alpha_{TB}}{\cos\alpha_{TA}}\times(R_A+R_B).$$

Based on the chosen values of the tilt angles, $R_B=(R_A+R_B)/\sqrt{2}$. You can then solve for R_A from the known values of R_A+R_B as well as R_B.

The AD8609 op amp's input-bias current causes errors, but these errors are negligible because the input-bias current at room temperature is just 1 pA. The AD8609's input offset voltage, which is typically 50 µV, also causes errors, which are negligible as well (**Reference 2**). The signals at the outputs of comparators IC_{2A}, IC_{2B} and IC_{2C}, IC_{2D} are ORed in NAND gates IC_{3A} and IC_{3B}, respectively. NAND gate IC_{3C} serves as an inverter, and the output of IC_{3D} is the logic output of a window comparator in which logic low appears only when the Z-axis output voltage is between V_{REFA} and V_{REFB}, referenced to supply midvoltage $V_S/2$.

Grouping the comparators into IC_{3A}, IC_{3B} and IC_{3C}, IC_{3D} pairs ensures independent detection on whether the Z axis is 0 or 180° in the vertical orientation. LED_1 and LED_2 illuminate successively upon slowly tilting the Z axis by 45 and 60° (**Reference 3**). Similar action occurs when you orient the Z axis steadily vertically while moving downward. LED_1's brightness is turned on at an apparent decrease of gravity to $g/\sqrt{2}$. LED_1 dims, and LED_2 simultaneously illuminates when the vertical acceleration is equal to or lower than g/2. The operation of the detector is ratiometric and is therefore virtually insensitive to supply-voltage variations. EDN

REFERENCES

1 "ADXL335 Small, Low Power, 3-Axis ±3 g Accelerometer," Analog Devices, 2009, www.analog.com/en/sensors/inertial-sensors/adxl335/products/product.html.

2 "AD8603/AD8607/AD8609 Precision Micropower, Low Noise CMOS, Rail-to-Rail Input/Output Operational Amplifiers," Analog Devices, 2003 to 2008, www.datasheetcatalog.com/datasheets_pdf/A/D/8/6/AD8609.shtml.

3 "HLMP-EGxx, HLMP-EHxx, HLMP-ELxx New T-1¾ (5mm) Extra High Brightness AlInGaP LED Lamps," Avago Technologies, AV02-1687EN, April 21, 2009, www.avagotechlighting.com/signageandsigns/signs/si_new_products.

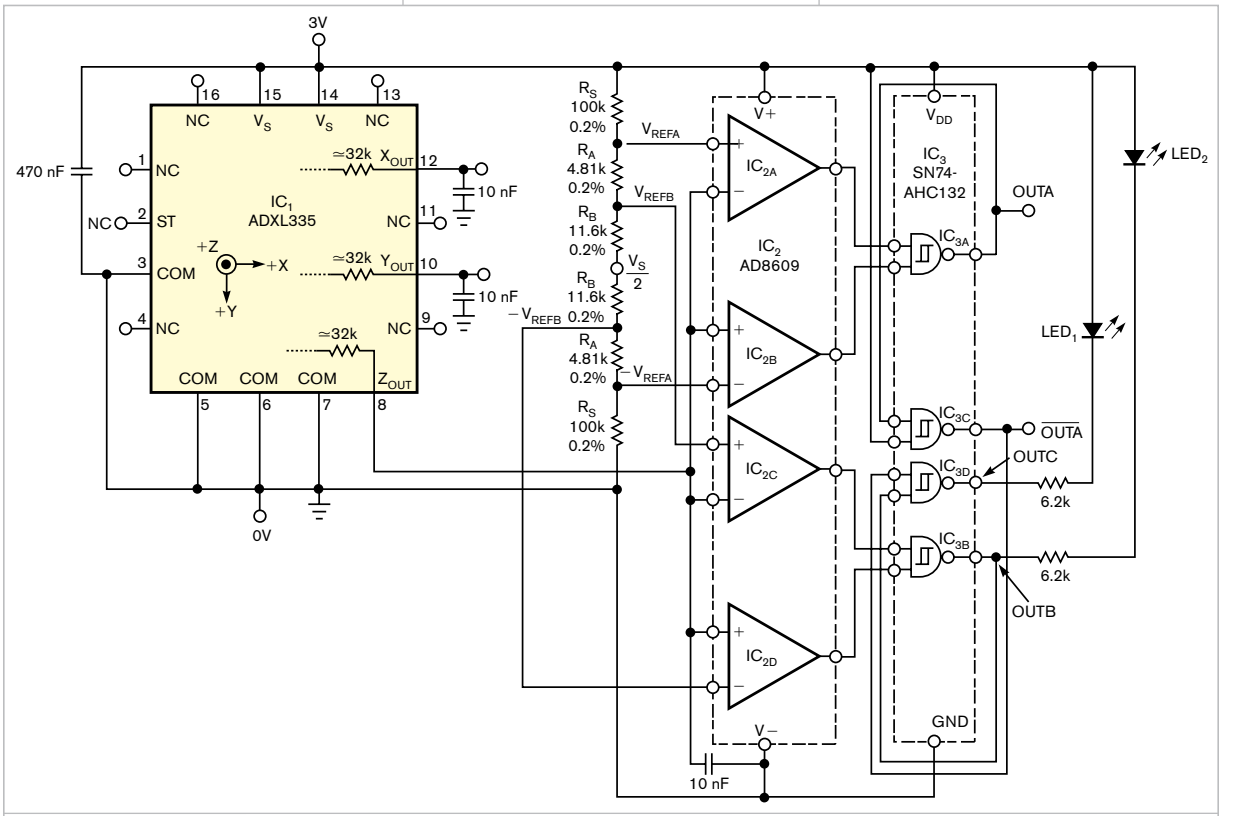


Figure 1 An accelerometer's Z-axis output, compared with two reference voltages, can generate two digital outputs.

Electronically generate rotating magnetic fields

F Ferrero, J Blanco, JC Campo, and M Valledor,
University of Oviedo, Gijón, Spain

Many applications, such as medical therapies, magnetic stirrers, and induction heating, call for a rotating magnetic field, which you can generate by attaching multiple permanent magnets to a dc motor. This technique involves problems, including noise and the need to maintain the moving parts. This Design Idea describes how you can instead use a microcontroller and a full-bridge driver to generate variable magnetic fields without mechanical elements. The approach requires no maintenance, does not wear out, and provides high-precision speed control. It does, however, require large cores to achieve powerful magnetic excitation.

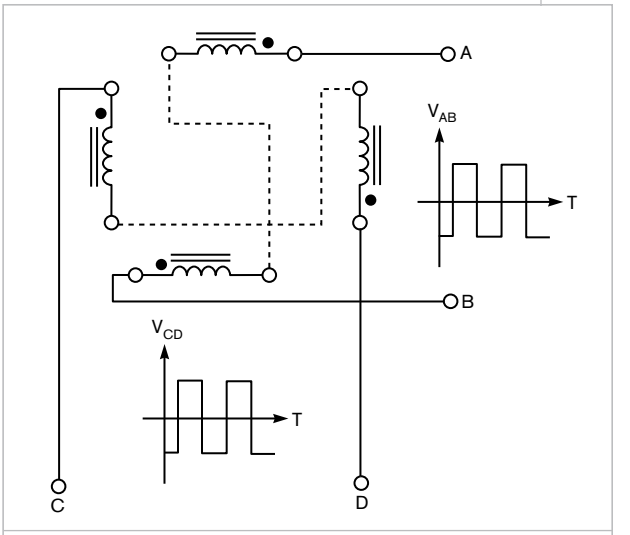


Figure 1 Two pairs of magnetic coils and their excitation waveforms show how to generate a rotating magnetic field.

You can excite a stationary magnetic coil with an ac current, which induces a north pole and a south pole that change at the frequency of the signal excitation. You can increase the number of poles by implementing a configuration with more magnetic coils. **Figure 1** shows a practical arrangement of the coils and the typical excitation waveforms. Note that the terminals of each pair of coils connect in series opposition to always obtain magnetic fields with different polarity.

Multiple ICs can drive inductive loads. This circuit uses an L6204 dual full-bridge driver from STMicroelectronics (www.st.com). Each bridge has four power-DMOS transistors with on-resistances of 1.2Ω. A PIC16F628 microcontroller from Microchip (www.microchip.com) controls the switches of the dual-bridge driver (**Figure 2**). Typical waveforms show how each circuit is excited (**Figure 3**).

To ensure the correct driving of high-side drivers, the circuit supplies a voltage higher than the supply voltage at IC_2's Pin 20. External capacitors C_1 and C_2 and diodes D_1 and D_2 use a charge-pump method to produce this voltage. You can independently control the four half-bridges by means of the IN1, IN2, IN3, IN4, ENABLE1, and ENABLE2 inputs.

The microcontroller timer's interrupt generates the IN1 to IN4 waveforms with high precision. Using a 10-MHz oscillator crystal and fixing the postscaler to eight, the microcon-

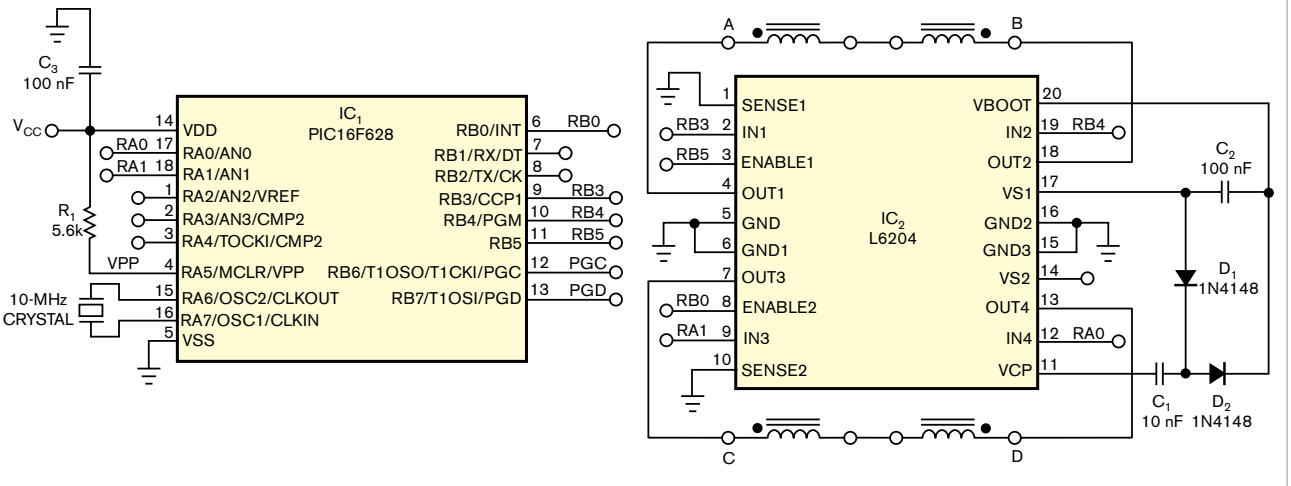


Figure 2 The circuit comprises a full-bridge driver and a microcontroller.

troller's counter increments every 3.2 µsec: 1/((10 MHz/four instructions)/eight). Taking into account that the interruptions generate when the counter overflows and the maximum count is as high as 65,535, or 16 bits, you can program the interruptions at 3.2 µsec and 210 msec: 3.2×65,535.

From this wide range of interruptions, the firmware lets the user select the precharge within a small subrange of frequencies divided into 10 levels,

THE FIRMWARE LETS THE USER SELECT THE PRECHARGE WITHIN A SMALL SUBRANGE OF FREQUENCIES.

meaning that you must vary the interruption from 49.89 to 60.45 µsec, a good range for this application. The new frequency of the interruption has a simple calculation that includes the level; the maximum frequency; and the separation between levels, which is a constant value that the operations include. You can download **listings 1** and **2**, which have complete C source code, from the Web version of this Design Idea at www.edn.com/100218dib.EDN

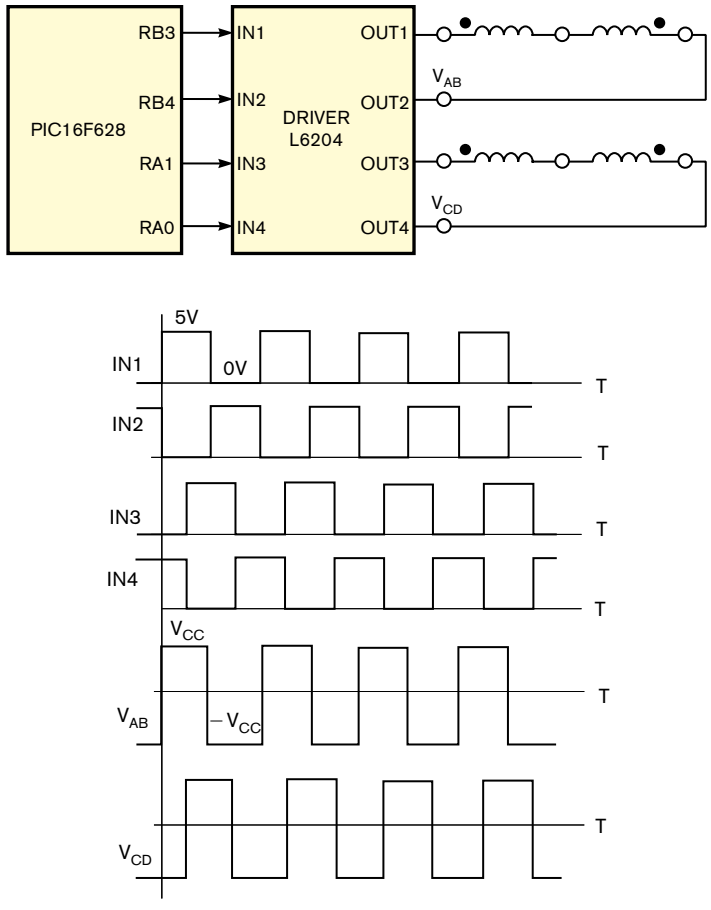


Figure 3 Waveforms show how each coil is excited.

Voltage reference stabilizes current sink

Suded Emmanuel, Emmanuel's Controls, Auckland, New Zealand

Analog circuits for long-term testing of passive components, such as 0.1%-tolerant resistors or high-intensity white LEDs, often require a constant current. Using two op amps and a voltage reference, you can develop a circuit that provides a constant-current sink with a variable setting of 0 mA to 0.99A. The circuit in **Figure 1** sinks a stable current through the load. The load current is insensitive to power-supply-voltage variations. IC_1 is a voltage reference that gives a stable 5V dc. It requires 500 µA of current from the power supply. IC_2 is a National Semiconductor (www.national.com) LM324 quad op-amp. Voltage follower IC_{2A} buffers the reference voltage from the rest of the circuit, which increases stability.

Resistor R_2 and potentiometer R_3 form a variable voltage divider that reduces the 5V reference voltage to a value between 0 and 3.26V. Unity-gain amplifier IC_{2D} drives the base of Q_1, a Darlington power transistor that has a current gain of 750, through R_4. R_4 and C_5 form a lowpass filter that prevents oscillation. You can drive Q_1 with a small base current. C_4 connects between the collector and the base of Q_1, adding further stability.

Operating as an emitter follower, Q_1 can drive an active or a passive load, such as a resistor or a high-brightness LED. Q_1's emitter connects to R_5, a 3.3Ω, 5W grounded power resistor. The voltage at IC_{2D}'s Pin 14 sets the voltage across R_5, which fixes Q_1's emitter current. Because of Q_1's high gain, the current in the load is effectively Q_1's emitter current.EDN

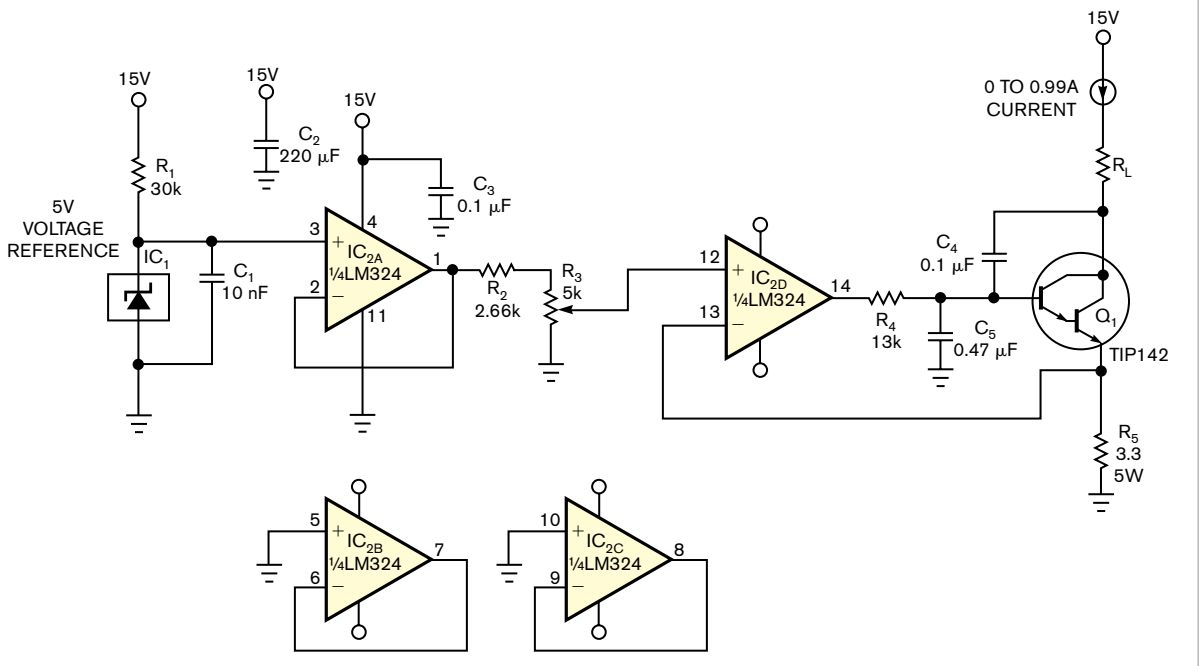


Figure 1 A voltage reference and two op amps provide a stable voltage to R_5, which provides a stable load current.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Circuit lets you test sample-and-hold amplifiers

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

Sample-and-hold amplifiers sample an analog voltage and hold it until an ADC can digitize it. A perfect sampling circuit holds a voltage until digitizing is complete. Thus, the amplifier's output is identical to its input. Real sample-and-hold amplifiers, however, can gain or lose voltage, producing an error. Offset voltages in amplifiers cause a static additive error. Further, there occurs a specific additive error, the so-called voltage pedestal, which originates within the transition from the sample state to the hold state because of a parasitic charge transfer to the hold capacitor.

A sample-and-hold amplifier uses an analog switch to connect a signal to a holding capacitor. When the switch closes, thus having low resistance, the capacitor charges to the sampled input voltage. During the hold time, when the switch has high resistance, the sampling capacitor holds the voltage until the ADC digitizes it. During the transition from low to high switch resistance, a parasitic charge injection, mainly from the gate of the switch to the hold capacitor, continues to charge the capacitor until the switch's control voltage reaches a steady logic level. The injected charge produces an error voltage at the capacitor. Additional errors may occur during the hold time. Leakage and bias currents in the amplifier combine with tens of picoamps of leakage current in the switch and capacitor to cause the capacitor to charge or discharge during hold time.

By applying a logic-control signal with a duty cycle of D and 1−D, you can measure a mean output voltage difference, $[\Delta V_{OUT}]=[V_{OUT}-V_{IN}]$, which the following **equations** show. $[\Delta V_{OUT}]=V_{STAT}+(1-D)V_{INJ}+½(1-D)^2V_{DROPPEAK}$, and $[\Delta V_{\overline{OUT}}]=V_{STAT}+DV_{INJ}+½D^2V_{DROPPEAK}$, where ΔV_{OUT} and $\Delta V_{\overline{OUT}}$ are the output-voltage differences for D and 1−D, respectively; V_{STAT} is the steady output-voltage difference for a selected value of the reference input voltage, D is the duty cycle, V_{INJ} is the voltage pedestal, and $V_{DROPPEAK}$ is the peak voltage drop. **Figure 1** shows how the voltages in the **equations** change over time. If you apply a complementary control waveform with a duty cycle of 25%, you can measure another dc component of the sample-and-hold amplifier's output voltage. Finally, when the sampling switch is continuously on, you can measure the V_{STAT} voltage, which is a real dc voltage. V_{OUT} and $V_{\overline{OUT}}$ contain a waveform superimposed onto a selected value of the reference voltage. Thus, you should measure the mean values of these voltages using a series resistor with a value of, say, 10 kΩ.

Multiplying the voltage pedestal, a simple rectangular waveform, by the duty cycle yields the av-

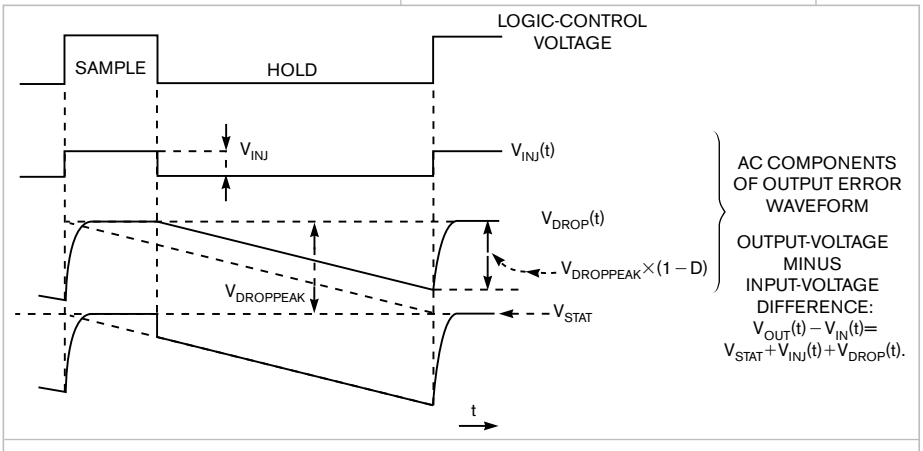


Figure 1 A sample-and-hold amplifier's holding capacitor experiences a voltage drop due to leakage and bias currents plus a voltage step, which results in a difference between the amplifier's output and input voltages.

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

erage value. In contrast, the voltage-drop waveform appears as a sawtooth. Its mean rises as one-half of the duty cycle squared. The peak-voltage-drop value denotes a hypothetical voltage drop at the end of a whole period, T, of the SAMPLE/$\overline{\text{HOLD}}$ logic-control waveform.

You can use the previous **equations** to find the values of the voltage pedestal and the peak voltage drop. A 75% duty cycle is a convenient value. The following **equations** are valid for this duty cycle: $V_{INJ}=6[\Delta V_{OUT}]-2/3[\Delta V_{\overline{OUT}}]-16/3V_{STAT}$, and $V_{DROPPEAK}=16[-[\Delta V_{OUT}]+1/3[\Delta V_{\overline{OUT}}]+2/3V_{STAT}]$. You must find the optimal repetition rate, f_{REP}, of the logic-control signal. As the optimal repetition rate increases, the difference in output voltage from the input is almost purely due to dc voltage offset plus the voltage pedestal: $(V_{\overline{OUT}}-V_{STAT})/(V_{OUT}-V_{STAT})\simeq 3$. The following **equation** finds the maximum value for the optimal repetition rate: $f_{REP}\leq(0.01/4)\times 1/(t_{ON}-t_{OFF})$, where t_{ON} and t_{OFF} are the on and off times, respectively. This **equation** ensures that the difference in values between the turn-on and turn-off times of the sample-and-hold amplifier's internal analog switch won't affect the accuracy of the precision 25 and 75% duty cycles by more than 1%.

If you evaluate the **equation** for a high-performance analog switch, such as the Analog Devices (www.analog.com) ADG1213, you get a repetition rate of 33 kHz or less. The difference due to voltage drop prevails at low-value repetition rates. In this case, the repetition rate can be the value of the frequency at which $V_{\overline{OUT}}-V_{STAT}\leq 1/10\times V_{INMAX}$, where V_{INMAX} is the maximum input-voltage range. The best way to determine the lower limit of the repetition rate is through experimentation.

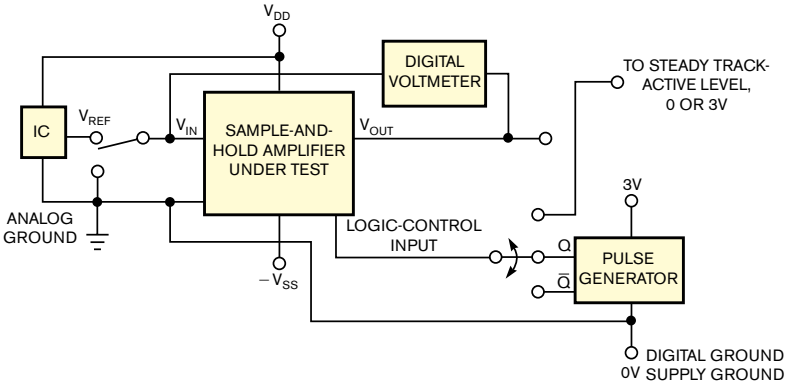


Figure 2 Use a digital voltmeter to measure the difference between the output and the input voltages of a sample-and-hold amplifier.

A tested sample-and-hold amplifier using the circuit in **Figure 2** uses a supply voltage of −1V, a drain-to-drain voltage of 5V, and a supply voltage of 3.3V for logic circuits in the pulse generator. Two sets of measurements at 25, 75, and 100% duty-cycle values for the AGD1213's internal switch control used input voltages of 0 and 2.5V. You will measure the output-voltage difference, approximately −0.0366 mV, and the pedestal voltage, approximately −0.0333 mV, at a repetition rate of 1.762 kHz. The value of the residual effective charge injection, Q_{INJ}, into the hold capacitor, $C_H=2$ nF, is $Q_{INJ}=C_H\times V_{INJ}$. The value is negative and doesn't exceed −75 fC. The following **equation** defines the difference of charge injection within the 2.5V range of input voltage: $\Delta Q_{INJ}=Q_{INJ}(2.5V)-Q_{INJ}(0V)$ and yields a value of −6.7 fC. The following **equation** determines the residual effective leakage current from the acquired values of peak voltage drop at a repetition rate of 160 Hz: $I_{LEAK}=C_H\times V_{DROPPEAK}\times f_{REP}$, where I_{LEAK} is the leakage current. A leakage current at the input voltage of 0V is approximately 17 pA, and a leakage current at the input voltage of 2.5V is approximately −17 pA. EDN

REFERENCE

1 "Low Capacitance, Low Charge Injection, ±15 V/+12 V iCMOSTM Quad SPST Switches," Analog Devices Inc, 2005, www.analog.com/en/switchesmultiplexers/analog-switches/adg1212/products/product.html.

Add hysteresis to a voltage comparator

Luca Bruno, ITIS Hensemberger Monza, Lissone, Italy

Positive feedback is a typical technique for distributing hysteresis around a comparator, provided that you have a resistive path between the comparator's output and the noninverting input. Positive feedback forms two threshold voltages that have (or assume) fixed values. In addition, they depend on the saturation values of the comparator's output stage. Plus, the load conditions affect their accuracy. The circuit in **Figure 1** provides an alternative for applications requiring a comparator with hysteresis that has precise thresholds that you can easily and independently set. The circuit includes two inverting and noninverting threshold comparators whose outputs directly drive a set/reset latch. You can use a latch with either active-low or active-high inputs.

You can generate the positive and negative threshold voltages using a

precision voltage reference that powers a resistor divider (not shown) or by driving the inputs with DACs if you need a digitally programmable comparator. The circuit's high input impedance facilitates this task. Because of its hold state, the latch nullifies the effects of frequent switching on the comparator's outputs due to noise on the input signal. The circuit thus acts as a Schmitt trigger even if there is no positive feedback. The latch introduces a propagation delay that's normally a few tens of nanoseconds and is negligible in low- to medium-speed applications. Because the latch has complementary outputs, the circuit provides a noninverting characteristic on the Q output and an inverting characteristic on the $\overline{Q}$ output (**Figure 1a** and **b**).

Some integrated latches have only the Q output. If you need an inverted output, you need only to exchange the comparator outputs with the latch inputs for both circuits; the upper comparator drives the reset input, and the lower comparator drives the set input. You can use open-collector or open-drain comparators to process bipolar or positive signals higher than the supply voltage of the latch. You can easily interface them without using clamping diodes. You must add only a pullup resistor that the logic supply powers.

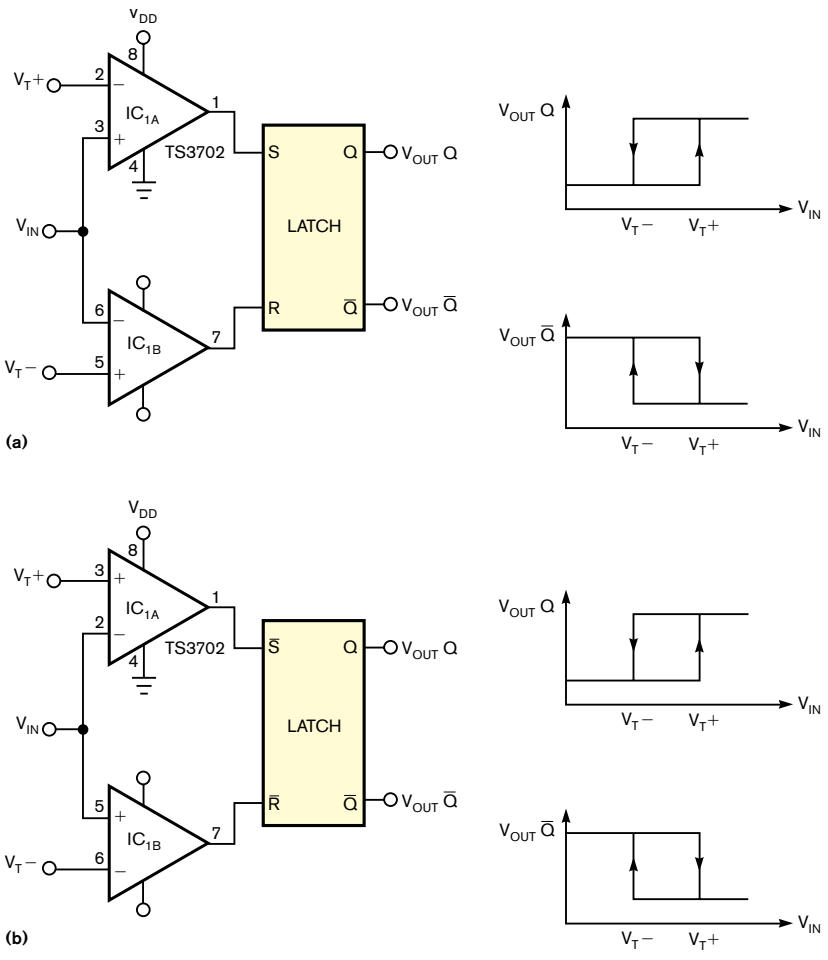


Figure 1 The set/reset latch provides hysteresis for the comparator. The circuit provides a noninverting characteristic on the Q output and an inverting characteristic on the $\overline{Q}$ output (a and b). Exchange the comparator outputs with the latch inputs for both circuits: The upper comparator drives the reset input, and the lower comparator drives the set input, and you get an inverting characteristic on the Q output and a noninverting characteristic on the $\overline{Q}$ output.

The circuit uses IC_1, an STMicroelectronics (www.st.com) dual micropower comparator with a push-pull output stage. In this case, the supply voltage must be the same as that for the latch. EDN

Broken-coil detector is simple yet robust

Juan Pablo Caram, Santiago, Chile

The circuit in this Design Idea was originally designed to detect damaged conveyor belts in the mining industry. Thin coils are embedded in the conveyor belt. If the belt suffers damage, it stretches at the affected location, causing one or more coils to break. The method for detecting the broken coil is to allow a "sensing" coil to magnetically couple with the passing coils in the belt, thus changing the total inductance of the magnetic pair. The sensing coil is part of an LC oscillator (**Figure 1**). When an intact coil passes the sensing coil, the frequency of the oscillator changes. If the conveyor belt moves at a fixed speed, the frequency of the oscillator modulates

at a fixed rate. When a broken coil passes the sensing coil, there is more time between modulations (and this is what you want to detect).

The oscillator doesn't generate a pure sine wave and is power-hungry but stable. It oscillates over a large range of LC pairs, even with a low quality factor and with almost any transistor. The amplitude is flat across a large bandwidth. The frequency is $1/(2\pi\sqrt{L_1C_1})$, where L_1 is the inductance of the sensing coil. R_2 represents its resistance. L_2 and switch S_1 represent either undamaged or broken coils in the conveyer belt. When S_1 is closed, the coils are undamaged, and when S_1 is open, the coils are broken. When the coupling between the sensing coil and a conveyor coil is perfect, it is equivalent to having the two connected in parallel, which would reduce the total inductance and increase the frequency of oscillation.

The problem now becomes how to detect different frequencies by implementing an FM demodulator or frequency-to-voltage converter. An easy way to accomplish this task is to pass the signal from the oscillator through an appropriately tuned lowpass filter. If the frequency range of the oscillator lies at the beginning of the roll-off, a higher frequency causes higher attenuation. At this point, an FM waveform has become an AM waveform, which you can easily demodulate using envelope detection.

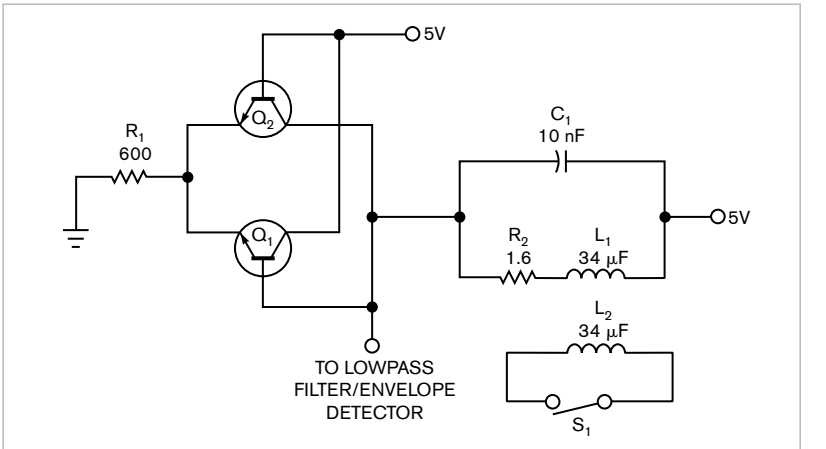


Figure 1 An oscillator generates a stable signal based on the values of L_1, L_2, and C_1.

Figure 2 shows a trivial RC lowpass filter comprising R_5 and C_2 and driving the envelope detector comprising D_1, R_6, and C_3. To find out whether the sensing coil couples to an external coil, you can check at the output of this circuit to see whether the voltage is above or below a certain threshold. You can experimentally determine the best value for the threshold. You can perform the comparison with an analog comparator or with a microcontroller after digitizing the signal. This last method would allow you to measure the time since the most recently detected undamaged coil passed. **Fig-**

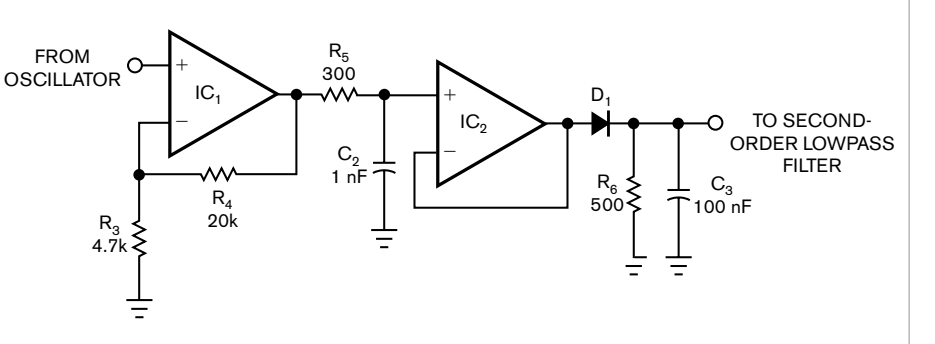


Figure 2 A lowpass filter changes the oscillator output into an envelope signal.

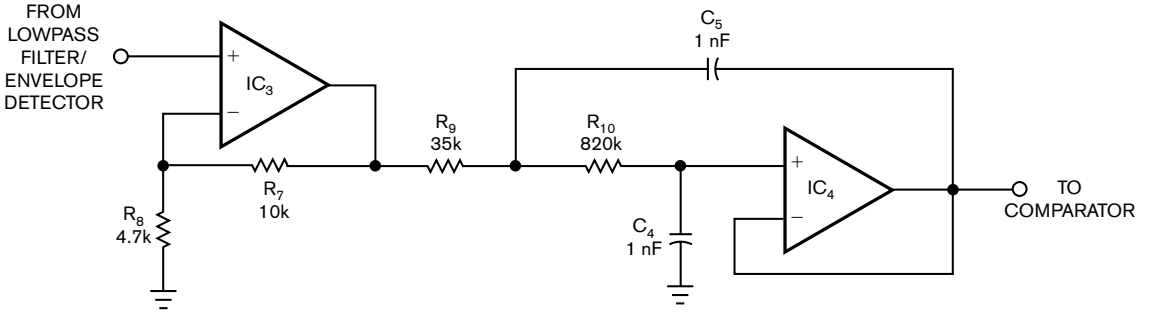


Figure 3 A second-order lowpass filter removes ripple from the envelope detector's output.

ures 3 and **4** show a simple and reliable analog method, using the envelope signal, to detect that a bad coil has passed. The output from the envelope detector contains ripple, so lowpass filtering allows a more precise discrimination in frequency. In this example, the filter in **Figure 3** makes the ripple insignificant without deteriorating the frequency response of the system.

The output of the filter then feeds into the comparator (**Figure 4**). R_{14} sets the threshold, which should be at the midpoint between the generated voltages with and without an external coil coupled to L_1. D_2, C_6, R_{12}, and comparator IC_6 behave as a countdown timer set to a time slightly longer than the period between passing coils. C_6 quickly charges to a maximum voltage when a good coil passes and then slowly discharges. If the time between (sensed or detected) consecutive undamaged coils is below a certain maximum, the voltage across C_6 should never go below the threshold that R_{15} sets, thus keeping the output of IC_6 low and lighting LED D_3. If a damaged coil passes, the oscillator's frequency should remain the same, allowing the voltage across C_6 to drop enough to trigger the comparator, turning off the "all-good" light.

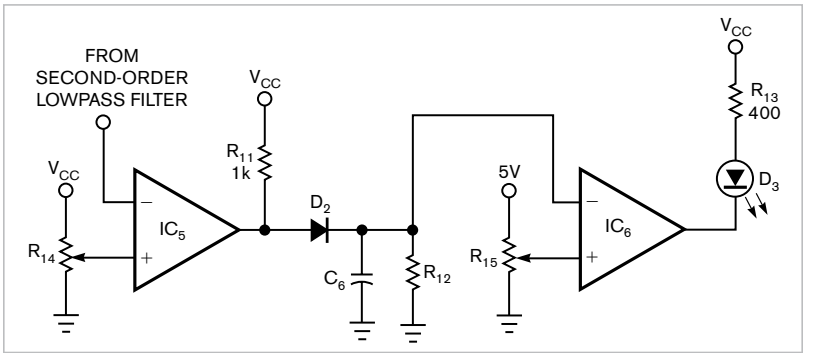


Figure 4 Comparator IC_6 monitors the voltage across C_6 and activates when the C_6 voltage exceeds a preset value from R_{15}.

In a real-world application, you should latch the output to ensure that the operator notices the alarm condition. The circuit could simply shut down the power to the conveyor belt, allowing immediate repairs and indicating where the fault occurred. The circuit uses just a few common components with large tolerances on their values. Its use of transistors, op amps, discrete linear and nonlinear components, oscillators, filters, demodulators, converters, and magnetically coupled circuits make it an excellent teaching resource. It even gives some insight into how modern proximity-card technologies, such as RFID (radio-frequency identification) work. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas
READERS SOLVE DESIGN PROBLEMS

Strategy processes video in RAM

Yu-Chieh Chen and Tai-Shan Liao,
National Applied Research Laboratories, Hsinchu, Taiwan

Many video devices, such as the Analog Devices (www.analog.com) ADV7179 DAC, have analog-baseband-TV interfaces for PAL (phase-alternating-line) and NTSC (National Television System Committee) video signals. Unfortunately, these kinds of DACs accept video in interlaced-image format only, but you may need progressive-scan video instead. Furthermore, many of the progressive-scan images vary in size, which makes it more difficult to convert a progressive-scan image to an interlaced image. Therefore, you need a universal and efficient image buffer, such as SDRAM or DRAM, as a strategy for separating the image field.

Figure 1 shows the timing for a typical progressive-image data format. The upper four signals include the progressive-image source, including a frame-synchronization signal, a line-synchronization signal, a signal, and a pixel clock with pixel-image data. The lower two signals are the frame-synchronization signal, which contains many line-synchronization signals when the frame-synchronization signal is high, and the line-synchronization signal.

The pixel clock writes the progressive-image data into FIFO (first-in/first-out) memory. A higher-rate data clock can then write the data into RAM when each line-synchronization signal is low. This procedure ensures that the progressive-image data will correctly write into SDRAM regardless of how the pixel clock changes because of the various progressive-image data sizes. When the RAM write-enable signal or RAM read-enable signal is high, the system writes data into or reads data from SDRAM.

Figure 2 shows the frame-synchronization signal of progressive-image data and the frame-synchronization signal of interlaced-image data. The write-new-data and read-old-data enable signal executes at every line-synchronization signal of the progressive-image data when at a low level and at every frame-synchronization signal when at a high level. You can execute the read-old-data enable

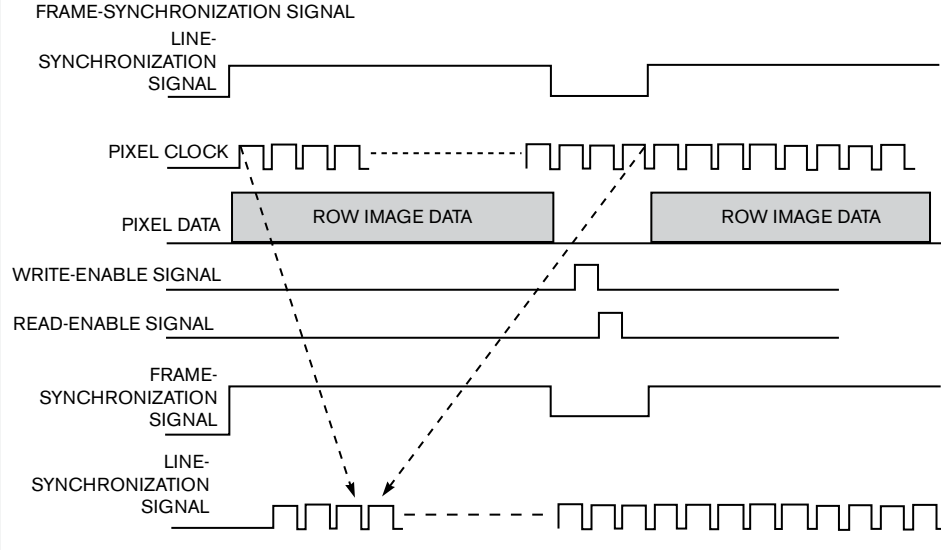


Figure 1 The pixel-clock signal puts image data into FIFO memory, which later synchronizes and goes into system RAM.

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

signal only when the frame-synchronization signal is low, however. This scenario occurs when there are no valid image data in this period. **Figure 3** shows the data flow of the SDRAM-accessing procedure. A frame may, for example, contain 15 rows, in which you define the row data to count from 00 to 0e. Image data for odd rows are one, three, five, seven, nine, 11, 13, and 15, and image data for even rows are two, four, six, eight, 10, 12, and 14.

By using this SDRAM-accessing strategy, you can generate the interlace data and synchronize it with the frame-synchronization signal of the original progressive data. Thus, you need not worry about image size. Moreover, it can easily tune the interlaced-image data timing, changing the number of blank rows, without changing the write-into- or read-from-SDRAM sequences. You need to decide only which line-synchronization signal in low-level periods reads the old image data from the SDRAM. EDN

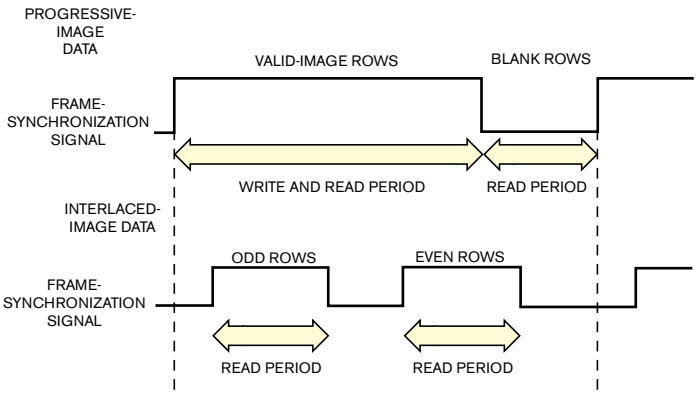


Figure 2 The frame-synchronization signal of progressive-image data alternates between reading odd and even rows of data in memory.

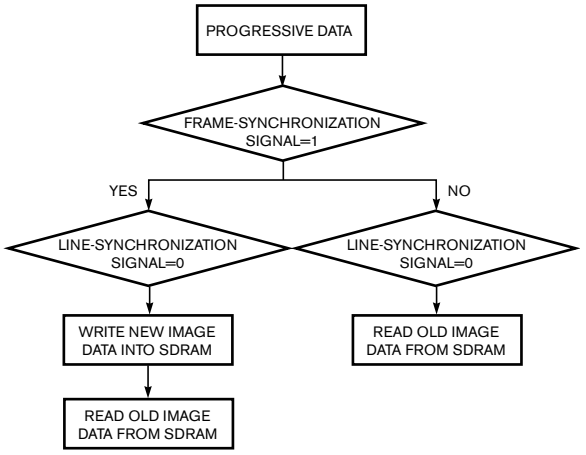


Figure 3 The state of the frame-synchronization signal determines whether the system performs a read or a write operation.

Rectangular-waveform generator produces 25 and 75% duty cycles

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

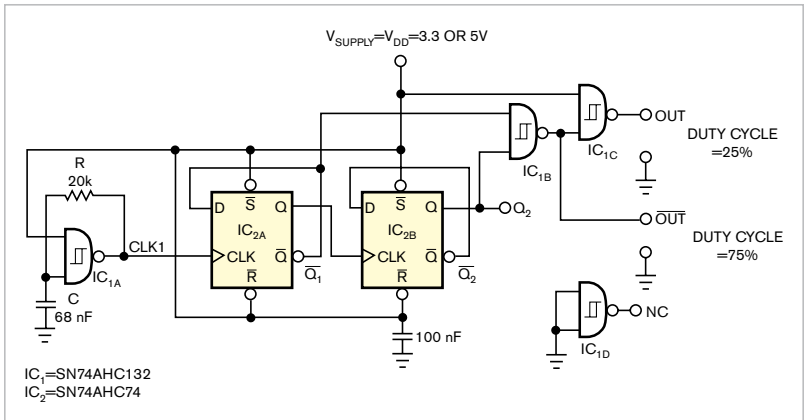


Figure 1 This circuit uses two flip-flops and three NAND gates to generate waveforms with 25 and 75% duty cycles.

Test applications may call for a rectangular waveform having a precision duty cycle higher or lower than 50%. The circuit in **Figure 1** is a free-running generator using just two ICs that produces rectangular-waveform duty cycles of both 25 and 75%. It holds the duty-cycle accuracy regardless of the duty-cycle accuracy from the signal source, an oscillator circuit comprising a Schmitt-trigger input NAND gate, IC_{1A}. Flip-flop IC_{2A} divides the oscillator's frequency by two at its $\overline{Q_1}$ and Q_1 outputs. Flip-flop IC_{2B} functions as a modulo-two divider clocked from the Q_1 output of IC_{2A}. Thus, IC_{2A} and IC_{2B} divide the oscillator's output by four.

NAND gates IC_{1B} and IC_{1C} generate the output waveform from the $\overline{Q_1}$ and Q_2 signals. **Figure 2** shows the output from NAND gate IC_{1B}. You can generate the 25% duty cycle by simply re-

placing the waveform that IC_{1B} outputs with the one that gate IC_{1C} outputs. If the active level is low instead of high, you can simply interchange the outputs of IC_{1B} and IC_{1C}.

The repetition frequency, $1/T_{REP}$, of the oscillator employing IC_{1A} is almost independent of the supply voltage within the range of 3 to 5V because both the positive and the negative thresholds of the input CMOS Schmitt trigger are roughly proportional to the supply voltage. Rough analysis gives a repetition frequency of approximately $2.7/\tau$, where $\tau=RC$, the time constant of the RC circuit around gate IC_{1A}. Further, the oscillator's waveform duty cycle is approximately 46.3%. EDN

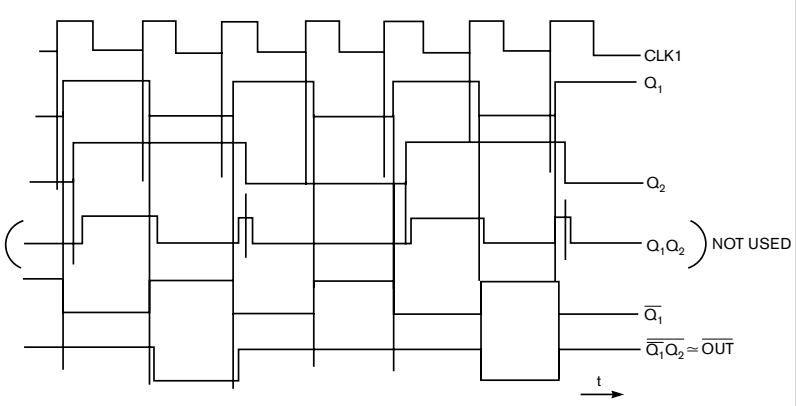


Figure 2 NANDing the $\overline{Q_1}$ and Q_2 logic signals gives a glitch-free output.

Battery simulator has variable ESR response

Barry Galvin, Grae LLC, Simi Valley, CA

You may lack experience and hardware when designing battery-operated products. The battery life of a product can depend more on the ESR (equivalent series resistance) than the terminal voltage. This situation is especially true when you use switching regulators to boost the battery voltage. The switching regulator creates a higher load as the battery voltage decreases. The ESR of a real battery is not constant. When you

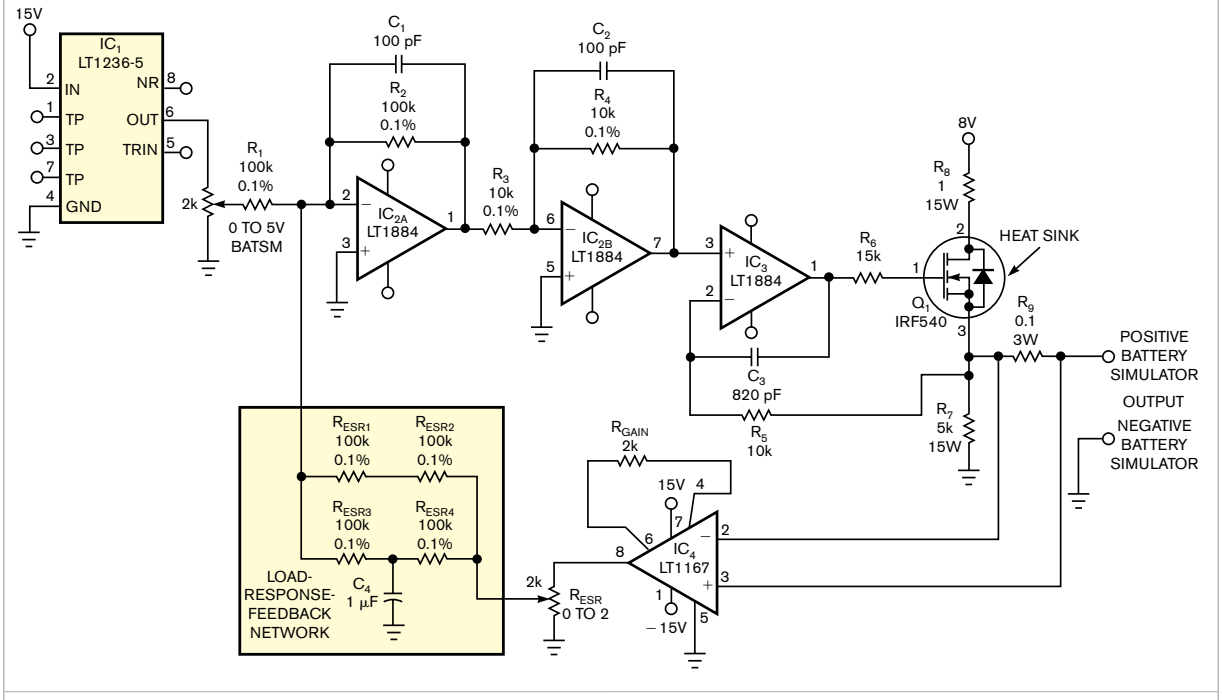


Figure 1 This simulator circuit represents the load response of many battery types.

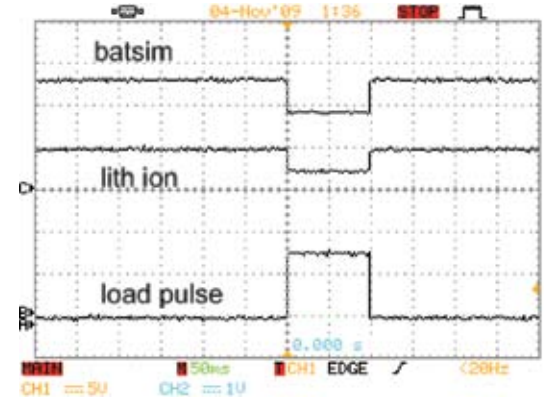


Figure 2 With no feedback capacitor, the simulator closely matches the response of a large lithium-ion battery.

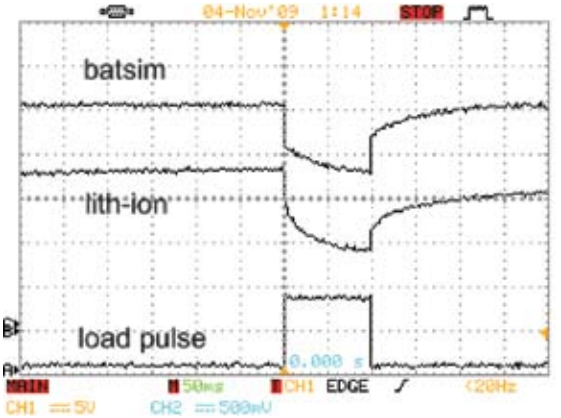


Figure 3 Adding capacitive feedback causes the simulator to act like a much smaller battery.

remove a battery load, it reacts and "heals" as its ions rediffuse. Portable electronics may include a low-power or a sleep mode. The device takes short high-power pulses from the battery.

The battery simulator in this Design Idea duplicates a battery's ESR-response curve. If you place different values in the feedback network, you can obtain various ESR curves. The circuit simulates most battery types, including lithium ion and alkaline. It supplies 0.5 to 4.2V at several amperes to the device under test, and it can simulate the ESR of a variety of battery types. You can change the delay to the final value of ESR by setting the ESR potentiometer. Some battery types exhibit this unique characteristic. It has a large influence on the delivery of pulsed current to a load.

In the circuit, IC_1 supplies a stable voltage, setting the unloaded output voltage (**Figure 1**). IC_2 provides the necessary inversions for the ESR function. IC_3 and Q_1 form a power-output stage that receives a voltage of 8V. Resistor R_8 limits the power. IC_4 senses the output current through R_9 and provides a gain of 20. This signal goes to the ESR timing circuit, providing both the ESR effect and the response timing.

You can simulate battery chemistries and sizes by varying the component values. If you omit C_4 and replace R_{ESR1} through R_{ESR4} with one 100-kΩ resistor, only the basic ESR function results. **Figure 1** omits power and bypass capacitors.

Applying a 1A load pulse without the capacitor in the feedback network causes the simulator response to closely follow the response of a 2000-mAhr lithium-ion 18650 battery (**Figure 2**). You can also add the capacitor to the feedback network to make the simulator better represent the response of a small, 200-mAhr lithium-ion battery (**Figure 3**). With proper adjustment of the circuit, you can produce many response curves. You can download National Instruments' (www.ni.com) LabView software and the voltage-ESR curves of selected battery types from Grae LLC (www.graellc.net). EDN

Create LED-lighting patterns without a controller

Jeff Tregre, www.BuildingUltimateModels.com, Dallas, TX

This Design Idea describes a simple LED-lighting-effects circuit comprising only five chips and costing only a few dollars. When you first observe the circuit in action, you will think that it uses a PIC (peripheral-interface-controller) chip requiring you to program hundreds of lines of code to generate the lighting effects. You can view the lighting effects in a video with the Web version of this Design Idea at www.edn.com/100318dia.

The circuit comprises seven functional blocks (**Figure 1**). IC_1 is an LM556, which has two 555 timers in one package. The first timer produces the main clock frequency of approximately 0.105 Hz. It toggles high to low approximately every 10 seconds. The polarity of the clock's signal changes the frequency of the VCO (voltage-controlled oscillator) that makes up the other half of IC_1 from low to high. Resistor R_2 and capacitor C_2 set the clock frequency. Changing either component changes the frequency.

The output from the first 555 timer feeds the control voltage input on the second 555 timer, letting it function as a VCO whose output frequency ranges from approximately 10 Hz when the first 555 timer output is high to approximately 33 Hz when the output is low. Components R_4 and C_3 set the VCO's frequency, and R_6 and C_4 control the smooth transi-

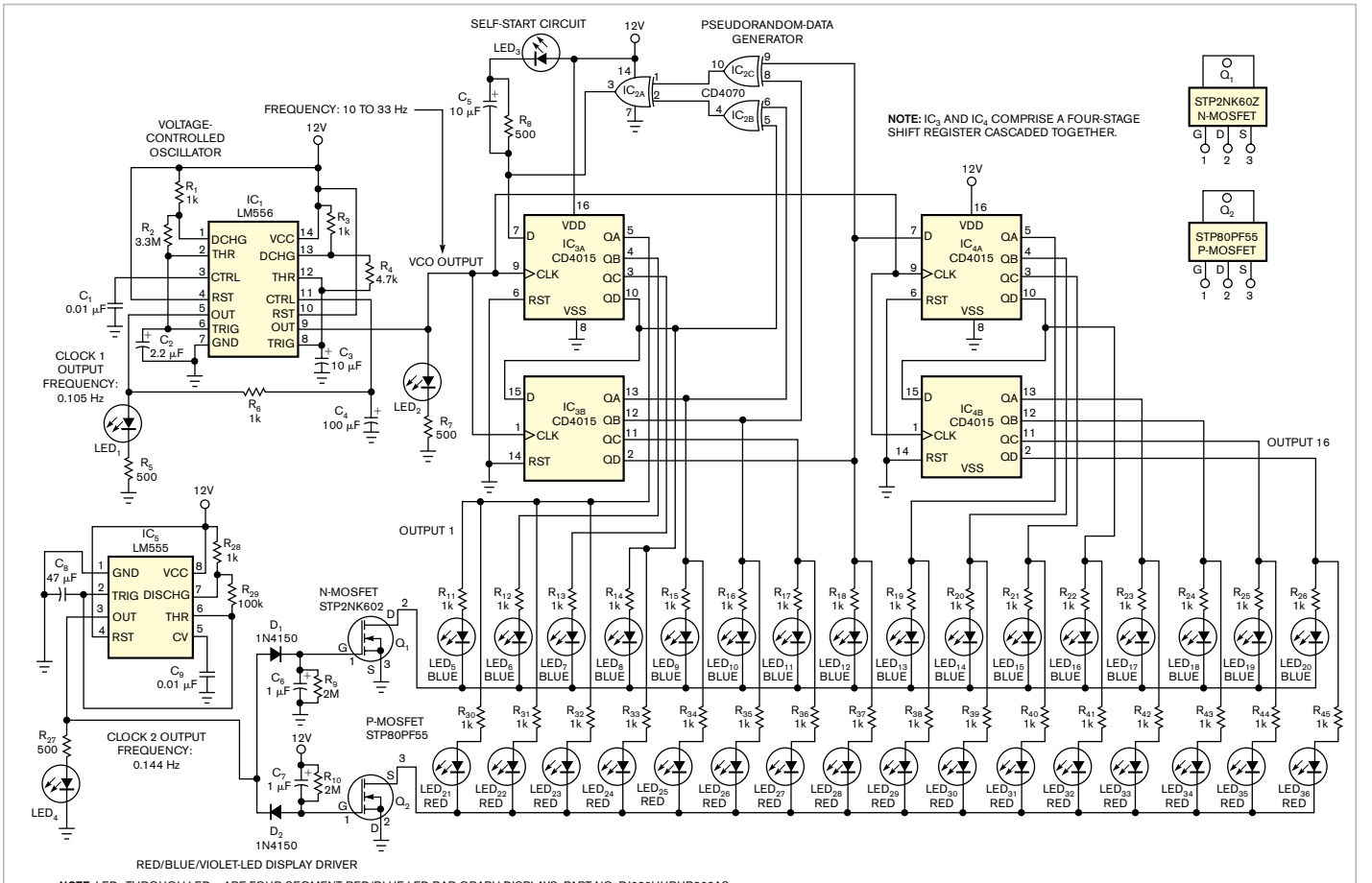


Figure 1 Two 555 timers create the clock pulses that drive blue and red LEDs.

tion of the VCO from 10 to 33 Hz.

LED_3, C_5, and R_8 act as a self-start circuit. Without it, you would need to add a pushbutton switch to toggle the data input of IC_{3A} from low to high during start-up. IC_2, a CD4070 quad exclusive-OR gate, acts as a pseudorandom-data generator. This circuit gives the illusion that bits of data span the bar graph.

IC_3 and IC_4 are CD4015 four-stage shift registers cascaded together. The data bits span the bar-graph displays in sequence from Output 1 to Output 16. IC_5, an LM555 timer, produces Clock 2's frequency of approximately 0.144 Hz. The inverse of this frequency toggles high to low approximately every 7 seconds, feeding the gates of N-MOSFET Q_1 and P-MOSFET Q_2, which act as the red/blue/violet LED-display driver. Clock 2 toggles high, enabling Q_1 and giving the blue LEDs a source to ground.

When Clock 2 toggles low, Q_2 turns on, giving the red LEDs a path to ground. C_6 and C_7, together with R_9 and R_{10}, respectively, act as a slow discharge circuit on the gates of the MOSFETs, keeping them on for approximately 2 seconds longer than Clock 2's pulse. The delay lets both the blue and the red LEDs be on at the same time for approximately 2 seconds and produces the color violet. This circuit uses N- and P-channel MOSFETs from ST-Microelectronics (www.st.com), but any general-purpose MOSFET should work. Just make sure that each one can handle at least 0.5A.

Figure 2 The bar graphs have both red and blue LEDs; turning on both yields violet.

The four-segment red/blue-LED bar-graph displays are unique. Each bar-graph display comprises one red and one blue LED in the same bar (**Figure 2**). Each LED has its own anode and cathode connections, thereby keeping this circuit simple without the need to add extra transistor drivers for each LED. You'd have to add them if their anodes, cathodes, or both were connected. This circuit requires four bar-graph displays. If you install any of the LED bar graphs backward, you will see the second color displayed, so that, if you were expecting red, you would get blue, and vice versa. EDN

Control stepper motors in both directions

Vladimir Rentyuk, Modul-98 Ltd, Zaporozhye, Ukraine

Stepper motors need bidirectional control in automatic machines or robotic applications. The circuit in **Figure 1** lets you control bipolar stepper motors and run them in both rotations. You can use the circuit in automatic devices and as an evaluation circuit for testing stepper motors. The circuit comprises clock oscillators IC_{3A} and IC_{3B}; a bidirectional, two-phase translator using an SN74HC74D dual flip-flop, IC_2, with a directional selector, IC_{3C} and IC_{3D}; and a push-pull L293DD channel-driver, IC_1. The circuit needs one power source, which depends on the stepper-motor specification. You can use a step-down voltage regulator to provide 5V dc. In many applications, an L7805A voltage regulator is suitable. Switch S_2 turns the motor on, and switch S_1 controls the motor's direction. Both signals can come from a sensor or a circuit with an open-collector output.

A circuit surrounding transistor Q_1 starts the motor. A forced starting is necessary because generators that employ two CMOS or TTL inverters are sometimes unstable after powering and can oscillate at a frequency of approximately 18 MHz. Thus, you need a delay after applying power to

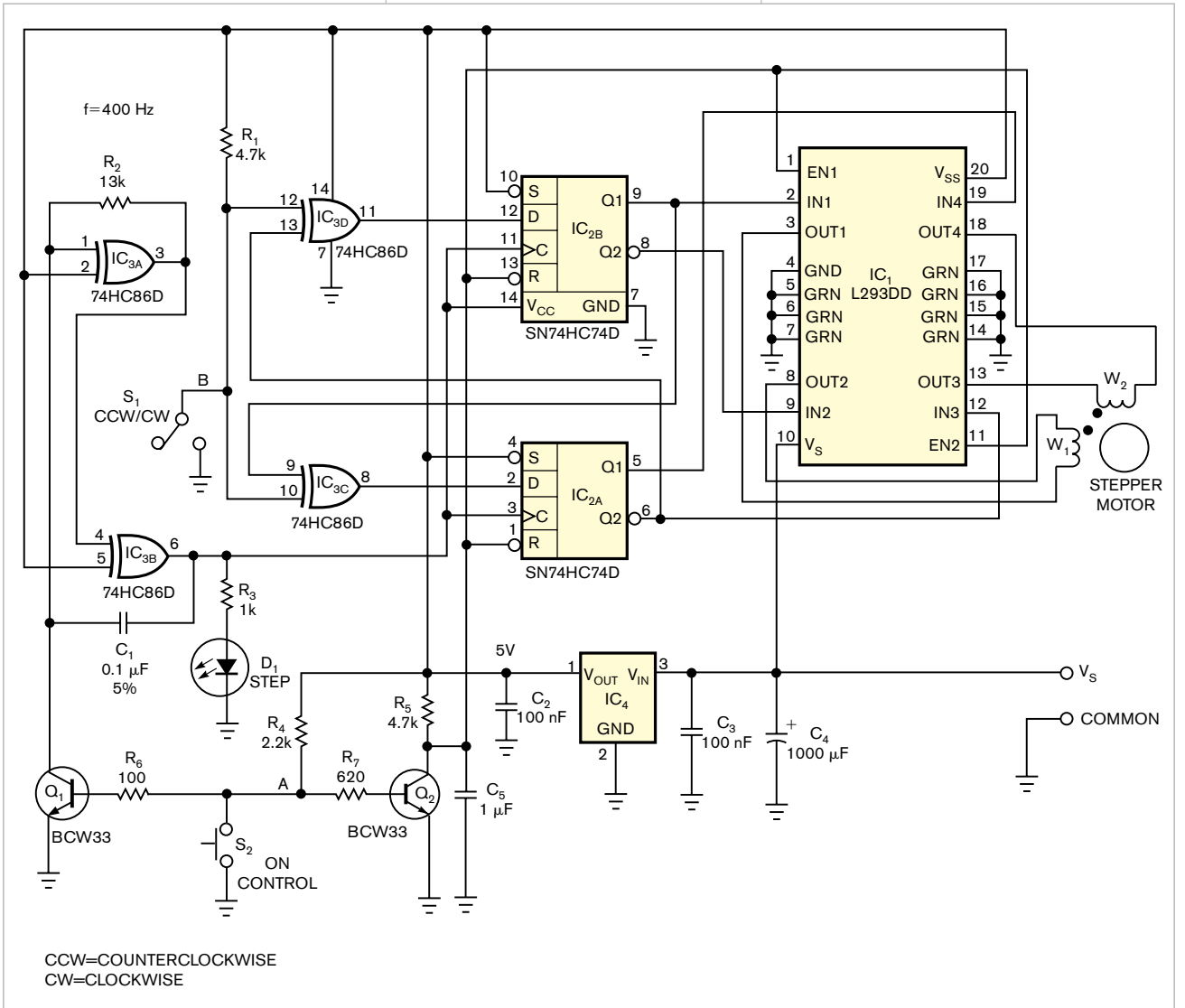
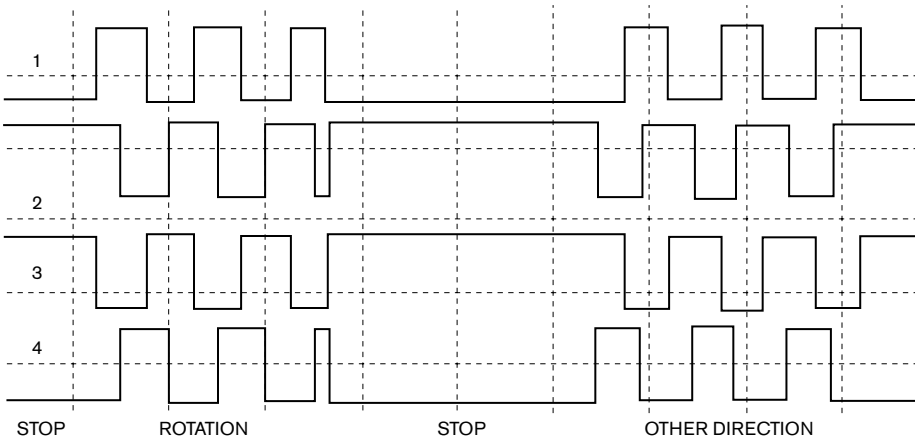


Figure 1 This circuit provides the necessary signals to drive a stepper motor. Switches turn the motor on and off and change its direction.

the circuit before sending the "on" command. The delay must be at least 100 µsec, but a delay of a few milliseconds is best. Capacitor C_5 eliminates the negative influence of bounce from S_2's contacts. The rotation of a rotor of the stepper motor begins when S_2 presents a low level to Point A. C_5 is unnecessary if a low-level signal from a circuit with an open collector comes to Point A—but not mechanical switches or buttons. Switch S_1 can be any suitable signal, such as that from a safety stop switch with a timer, trigger, or any open-collector output that connects to Point B. LED D_1 is a step indicator in "on" mode.



NOTES:
TRACE 1: OUTPUT Q1 OF IC_{2B} (IN1 OF IC_1).
TRACE 2: OUTPUT Q2 OF IC_{2A} (IN3 OF IC_1).
TRACE 3: OUTPUT Q2 OF IC_{2B} (IN2 OF IC_1).
TRACE 4: OUTPUT Q1 OF IC_{2A} (IN4 OF IC_1).

Figure 2 The motor stops when the outputs of IC_{2A}'s Q1 and IC_{2B}'s Q1 stay low. EN1 and EN2 are low, too.

The speed of rotation of a stepper motor depends on its specification from a step angle of the stepper motor and the frequency of the clock oscillator. **Figure 2** shows a timing diagram of the reversal mode of the circuit. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas
READERS SOLVE DESIGN PROBLEMS

Efficient LED power supply has battery backup

Zhihong Yu, Juno Lighting Group, Des Plaines, IL

LEDs find wide use in emergency lighting because of their high efficiency and control simplicity. The circuit in **Figure 1** provides a highly efficient and reliable design for emergency LED lighting at 3 to 6W. The circuit's input is 12V ac, which the full-wave bridge rectifies and one or two capacitors filter into dc. The battery (not shown) is a 12V lead-acid type. IC_1 compares the battery voltage to the supply voltage. When the rectified voltage drops below the battery voltage, the battery takes over to provide LED power.

The circuit has some small switching losses, which should be acceptable as long as IC_2, a 12V PB137 battery-charging circuit from STMicroelectronics (www.st.com), keeps the battery from draining.+ If this switch-over is unacceptable, add a 470-µF electrolytic capacitor to filter the input voltage to maintain a certain level above the battery voltage. Note that adding this capacitor lowers the power factor.

DIs Inside

To get 12V ac, you can use an electronic transformer. These transformers

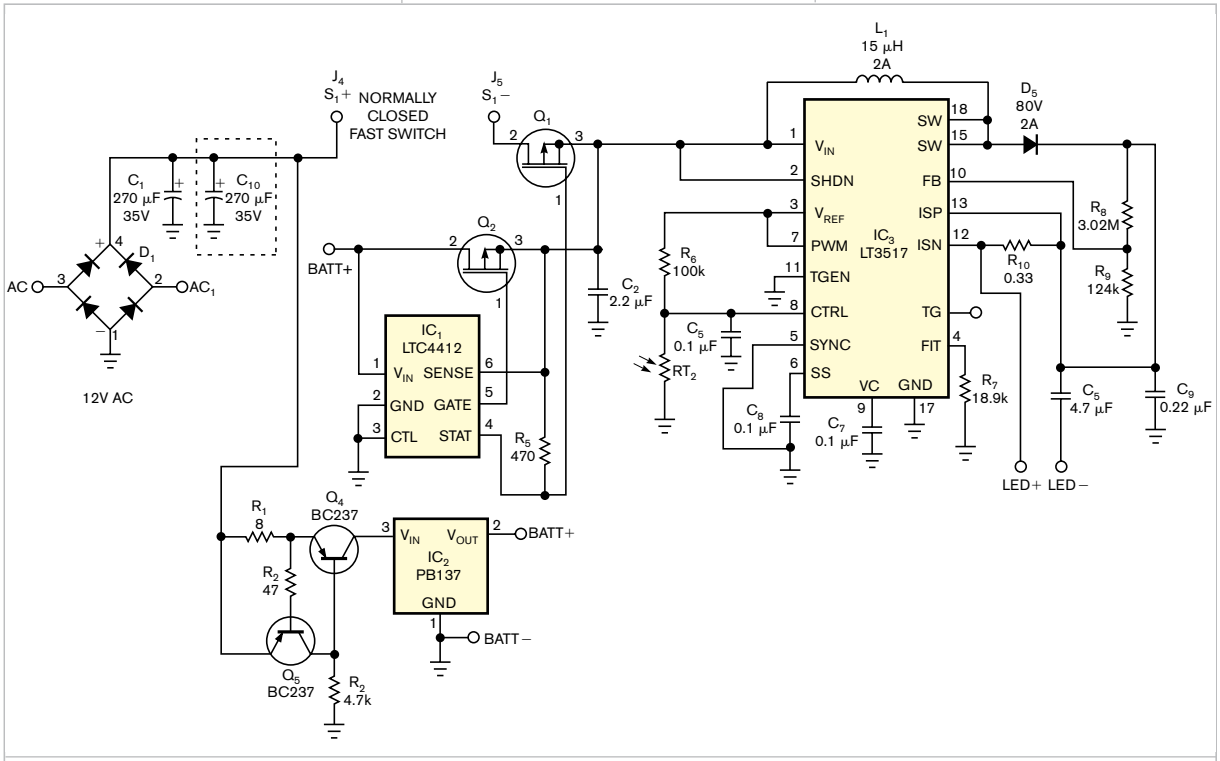


Figure 1 This circuit converts ac voltage to dc voltage, charges a battery, and drives LEDs from the ac source or the battery.

provide 12V at a higher frequency, so a 10-µF capacitor can hold the voltage high as well as provide a high power factor.

IC_1, a Linear Technology (www.linear.com) LTC4412, controls two external PFETs that create a near-ideal diode function for switching between ac and battery output. The PFETs' voltage drop is only about 20 mV compared with a normal 0.7V diode-voltage drop. Pin 5 is low when ac power is off, so you can use this pin to turn on a warning LED through another PFET. IC_2 has an internal current limit of 1.5A. Resistor R_1 limits IC_2's input; when the current reaches a certain level, Q_4 turns off the charging circuit. This IC does not require reverse-diode protection.

IC_3, an LT3517 LED driver from Linear Technology, acts as an inverting buck-boost converter because the input can range from 8 to 17V for rectified ac. R_{10} sets up the LEDs' current. Because the voltage drop from each of the three LEDs varies from 3 to 4V, the IC's output voltage can be higher or lower than its input voltage if all 300-mA LEDs connect in series.

By connecting a resistor divider, including a photocell, to the analog-dimming pin, Pin 8, you can achieve some dimming, which results in some power savings at higher ambient light. You can use IC_1's Pin 5 to turn on a transistor or an optoisolator to pull IC_3's control-pin voltage lower if you need to dim the LED when ac power is out. Resistor R_7 programs IC_1 to operate at 1 MHz. The circuit's efficiency is 82% when you power it directly from the ac power supply and about 70% from an electronic transformer.

With a few minor changes to the circuit, you can add LEDs. For example, you can use Linear Technology's LT3518, which is a pin-to-pin-compatible version of the LT3517 but with a higher switching-current limit. You may need to adjust the feedback-resistor pair R_8 and R_9 for higher output voltage. You may also need more input-filtering capacitance to hold up the voltage.

Tests show that the circuit can power as many as six LED in series. **Figure 2** shows the circuit in operation. EDN

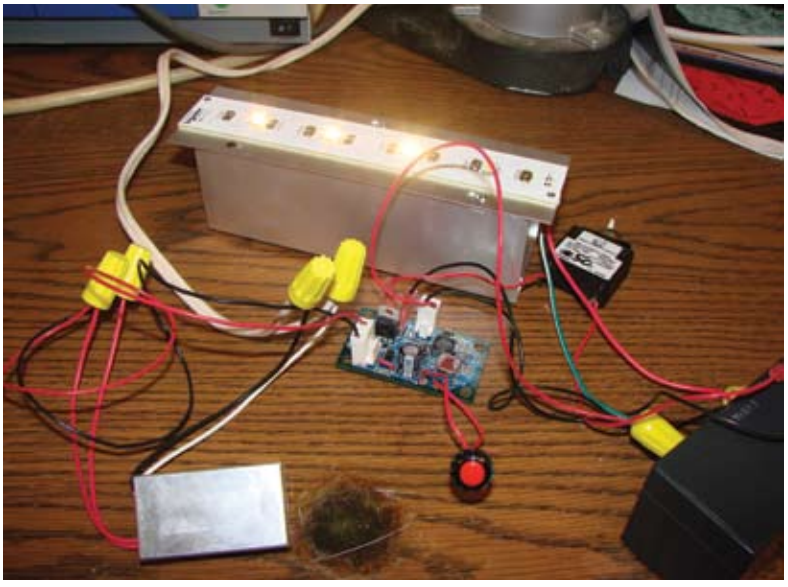


Figure 2 LEDs provide enough light for emergency lighting.

Single IC forms precision triangular-wave generator

Akshay Bhat, Maxim Integrated Products Inc, Sunnyvale, CA

The linearity of triangular waveforms makes the triangular-wave generator useful in sweep circuits and test equipment. For example, switched-mode power supplies and induction motor-control circuits often include a triangular-wave oscillator as part of their PWM (pulse-width-modulation) circuit.

The basic triangular-wave generator includes an integrator for generating the triangular-wave output and a comparator with external hysteresis, such as a Schmitt trigger, for setting the output amplitude (**Figure 1**). You can implement these components with a Maxim (www.maxim-ic.com) MAX9000 IC, which includes a high-speed operational amplifier, a 185-nsec comparator, and a precision 1.23V bandgap reference.

The integration of a constant current, which you obtain by applying constant voltage across a resistor, produces a linear ramp at the op amp's output. This output feeds a Schmitt trigger whose output feeds back to the integrator resistor. Abrupt state changes in the Schmitt trigger's output determine the peak voltages for the triangular-wave output. These changes in turn depend on the input threshold voltages you set for the Schmitt trigger.

Unfortunately for this circuit, the triangular-wave peaks must be symmetrical about the reference voltage you apply to the comparator's inverting input. To generate a triangular wave from 0.5 to 4.5V, for example, you must provide a reference voltage of (0.5V+4.5V)/2=2.5V.

It would be preferable to set this voltage range independently of the standard bandgap-reference voltage

available, 1.23V. You can achieve this flexibility by adding resistor R_3 to the hysteresis network in a single-IC version of the circuit (**Figure 2**). R_3 lets you set the triangular-wave peaks independently of the reference voltage.

To build the Schmitt-trigger comparator, you first select R_2. The comparator's input-bias current at C_{IN+} is less than 80 nA. To minimize the error this current causes, the current through R_2, $[(V_{REF}-V_{OUT})/R_2]$, should be at least 8 µA. R_2 requires two **equations**, corresponding to the two possible comparator-output states: $R_2=V_{REF}/I_{R2}$, and $R_2=(V_{DD}-V_{REF})/I_{R2}$.

Use the smaller of the two resulting resistor values. For example, if the supply voltage is 5V, the reference voltage is 1.23V, and the reference current is 8 µA, the two R_2 values are 471.25 and 153.75 kΩ, so this circuit uses the standard value of 154 kΩ.

Next, select R_1 and R_3. During a rising ramp, the comparator output is logic low (V_{SS}). Similarly, the comparator output is at logic high (V_{DD}) during a falling ramp. Thus, the comparator must change state according to the required peak and valley points of the triangular wave.

Two simultaneous **equations** result when you apply nodal analysis at the noninverting input of the comparator and solve for these two thresholds:

$$\frac{V_{IH}}{R_1}+\frac{V_{SS}}{R_2}=V_{REF}\left(\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}\right),$$

and

$$\frac{V_{IL}}{R_1}+\frac{V_{DD}}{R_2}=V_{REF}\left(\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}\right).$$

In this example, the voltage range of the triangular wave is 0.5 to 4.5V. You therefore substitute a value for V_{IH} of 4.5V, V_{IL} of 0.5V, V_{DD} of 5V, and V_{REF} of 1.23V into the above **equations** to obtain a value of 124 kΩ for R_1 and 66.5 kΩ for R_3.

You can now design the integrator. Considering the comparator's two possible output states, the magnitude of current flowing through R_4 is: $I_{R4}=(V_{DD}-V_{REF})/R_4$, or $I_{R4}=V_{REF}/R_4$. The op amp's maximum input-bias current is 2 nA. To minimize error, therefore, the current through R_4 must always be greater than 0.2 µA. This constraint implies that R_4's value is less than 6.12 MΩ.

The triangular-waveform frequency is:

$$f=1/\left(\frac{V_{OUT\,P-P}}{(V_{CC}-V_{REF})}(R_4C)+\frac{V_{OUT\,P-P}}{V_{REF}}(R_4C)\right).$$

For this example, the frequency is 25 kHz, the output voltage is 4V p-p, or 0.5 to 4.5V for a triangular wave, and the reference voltage is 1.23V. Solving for the resulting time constant, $R_{4C}=9.27$ µsec. Select a capacitance of 220 pF and a value of 42.2 kΩ for R_4.

The resulting output should match the desired frequency, provided that the op amp is not slew-limited. Because the feedback capacitor charges or discharges with a constant current, the output signal's maximum rate of

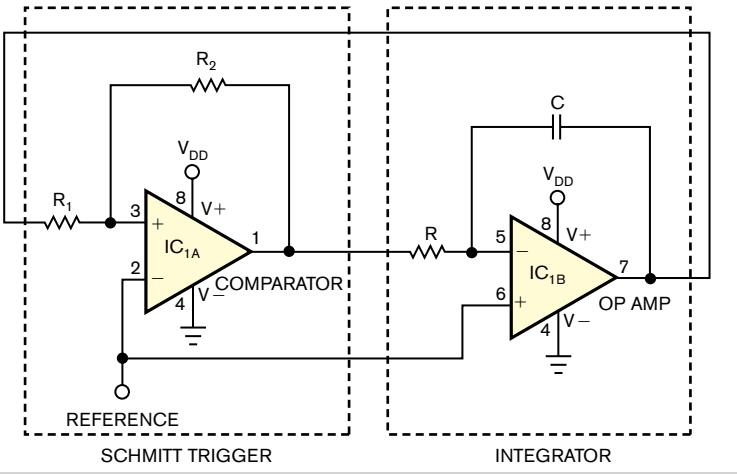


Figure 1 The basic triangular-wave generator includes an integrator for generating the triangular-wave output and a comparator with external hysteresis, such as a Schmitt trigger, for setting the output amplitude.

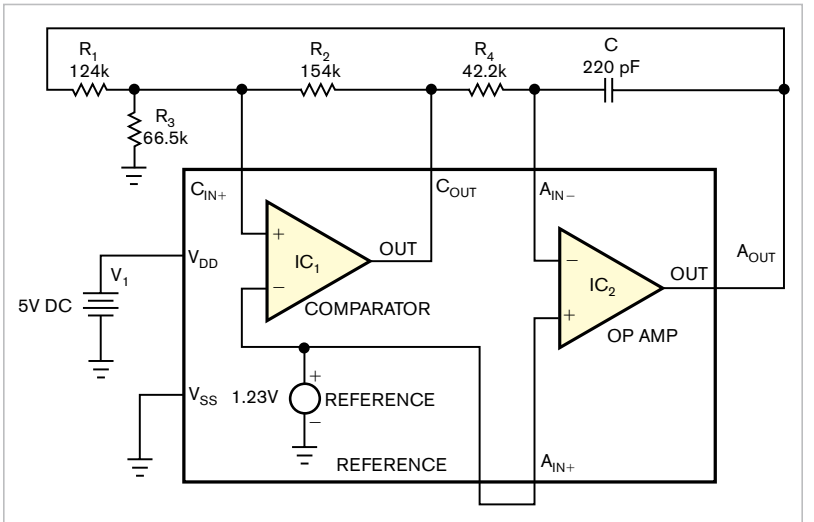


Figure 2 A triangular-wave generator employs an IC that includes an op amp, a comparator, and a bandgap reference.

change is:

$$\frac{dV_{OMAX}}{dt} = \frac{I_{R4MAX}}{C} =$$

$$\frac{V_{CC}-V_{REF}}{R_4C} = 0.406\frac{V}{\mu SEC}.$$

To provide a margin against process variations, the op amp's typical slew rate should be 40% higher than the maximum rate of change of the output signal—0.56V/µsec or greater in this case. The op amp's slew rate is 0.85V/µsec, which is therefore adequate for this 25-kHz waveform (**Figure 3**).EDN

REFERENCE

1 Terrell, David L, *Op Amps: Design, Application and Troubleshooting*, Butterworth-Heinemann, 1996.

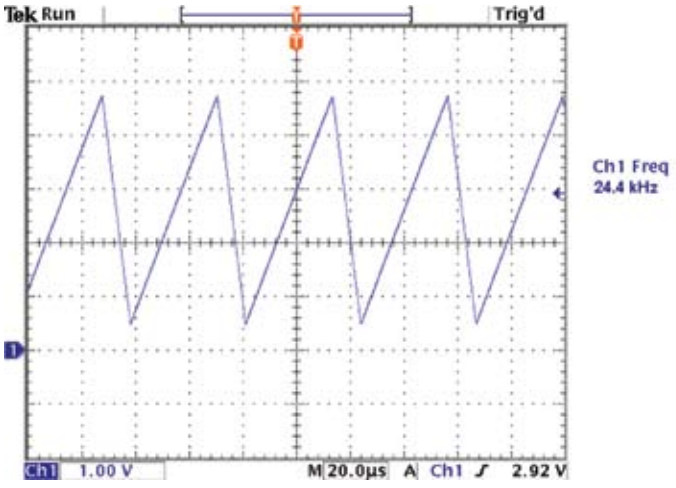


Figure 3 The op amp's slew rate is 0.85V/µsec, which is therefore adequate for this 25-kHz waveform.

Use a low-cost PWM ramp generator in switch-mode power supplies

Dwayne Reid, Edmonton, AB, Canada

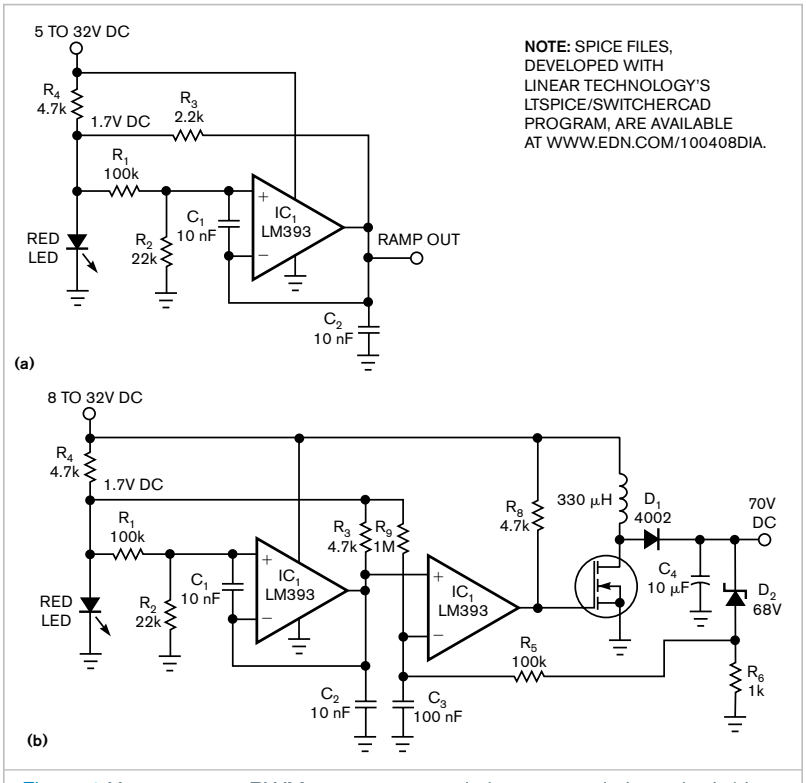


Figure 1 You can use a PWM ramp generator in low-cost switch-mode dc/dc power supplies (a). A 70V-dc upconverter employs the ramp generator (b).

The circuit in **Figure 1** shows a PWM (pulse-width-modulated) ramp generator that you can use in low-cost switch-mode dc/dc power supplies. Its supply voltage can range from 5 to 35V dc, and you can set the output-ramp amplitude of 0.3 to 1V. You can also set a minimum off time that lets you set a maximum 50% duty cycle for magnetic components that need duty-cycle limiting.

The ramp generator (**Figure 1a**) uses one-half of an LM393 dual comparator. The other half of the comparator is available to generate the PWM portion of the converter. The ramp amplitude and frequency depend on the reference. An ordinary red LED can act as a low-cost reference. Its forward voltage of approximately 1.7V is reasonably constant over indoor temperature ranges. The ratio of R_1 to R_2 sets the ramp amplitude relative to the reference, and R_1, R_2, and C_1 set the minimum off time. R_3 and C_2 establish a time constant, which sets the period. Note that the R_1, R_2, and C_1 network also affects the period. **Table 1** shows examples of various configurations.

Figure 1b, a 70V-dc upconverter, employs the ramp generator. You can easily configure it at any output ranging from the highest input voltage to whatever the FET can handle. This

example uses a 330-µH inductor, but you can easily change that value by choosing the appropriate PWM frequency.

Note that the output FET does not turn on quickly, and it doesn't need to, but it does turn off quickly. You can enhance the turn-off speed by adding a 2N4403 PNP transistor between the output of the comparator and the pullup resistor. Connect the base to the comparator, the emitter to the FET gate, and the collector to ground. Add a 100Ω resistor from the base to the emitter.

The circuit has slow load-transient response, which you can adjust by altering the time constant that R_5 and C_3 form. Note that R_9 and R_5 form a voltage divider that ensures the lowest error voltage at the PWM comparator is above the ramp's lowest point. The converter cannot operate without R_9.EDN

TABLE 1 EXAMPLES OF CONFIGURATIONS

Frequency (Hz)	R_3 (Ω)	C_2 (nF)	R_1 (Ω)	R_2 (Ω)	C_1 (F)	Approximate duty cycle (%)
500	120k	100	1M	220k	10n	100
700	100k	100	100k	22k	100n	100
62k	4.7k	10	100k	22k	10n	95
100k	4.7k	10	100k	22k	100p	100
200k	2.2k	3.3	100k	22k	3.3n	60
200k	1.8k	10	100k	22k	100p	95
400k	2.2k	1	100k	22k	10p	95
400k	5.6k	2.2	100k	22k	470p	50

MSP430's port-interrupt-request logic helps debounce contacts

Richard Neubert, Manchester, NH

Contact debouncing requires monitoring an input and waiting for it to stop toggling or at least establish that it's definitely switching from its initial state. You can use either analog or digital filtering plus hysteresis to accomplish contact debouncing, but this approach uses a lot of resources, including parts, board space, and CPU time, when multiple inputs need to be conditioned. Alternatively, you can either detect just the first state change or sample the input at least twice as often as the contacts can bounce. Sampling at mechanical vibration frequencies must be avoided. Both methods also require time delays to ensure that the contacts have finished bouncing.

These 1-bit approaches are attractive when you must condition multiple contact inputs. The first method requires conservative delay setting to avoid resuming the edge monitoring before the last bounce, and it's unsuitable for re-

LISTING 1 INTERRUPT-FLAG CHECKER

```
#define SwitchMask  0x01          // Port 1 bit(s) connected to switches.
#define Ndebounce   3             // (# of consecutive checks w/o an edge to wait for valid state)-1
                                  // (must be nonzero)
void chk_sw(void);
static unsigned char debounced_sw;      // The debounced switch state(s)
                                  //(output)

void main()
{
        P1DIR = ~SwitchMask;      // Set unused pins to output mode, unless
                                  //switched to a peripheral.
        debounced_sw = P1IES = P1IN;      // Init so we catch the first state
                                  //change.
        P1IFG = P1IE = 0;         // Keep Port 1 interrupts disabled.
        _EINT();                  // Set General Interrupt Enable (if used).

        while (1)
        {
                .                 // Main program loop.
                chk_sw();         // Do this here if loop execution period
                                  // is fixed by a timer,
                .                 // o/w use a timer to run chk_sw(). If
                                  // chk_sw() is made a timer  ISR, then
                .                 // debounced_sw will be volatile here.
        }
}

void chk_sw()                     // Port1 interrupt request
                                  //checker, called periodically.
{
        static int db_count = Ndebounce;  // Initialize to force P1IES
                                  //update on first pass of chk_sw().

        if (P1IFG & SwitchMask)
        {
                P1IFG &= ~SwitchMask;     // Clear the switch interrupt flag(s).
                db_count = Ndebounce;     // Must see P1IFG not set for
                                  //Ndebounce+1 passes.
        }

        if (db_count)
        {
                P1IES = P1IN;     // Cock for opposite edge (repeated in case we
                --db_count;       //  miss an edge while doing it).
        }
        else                      // Finished debounce
        {
                debounced_sw = P1IES;     // Debounced switch output

                // Can put code here that you want to run only when switch
                // inputs are stable.
                // If you care about the switch inputs only in this block, you can read the debounced
                // read the debounced state directly from P1IES, no need to save it
                // in debounced_sw.
        }
}
```

jecting noise. The second method adds to the actual bouncing time only the delay necessary for bridging the longest quasistatic input state during bounce—not the longest duration of bouncing. When you implement this function in software, however, it can add substantial overhead for monitoring the input for further transitions when the system detects a transition.

The Texas Instruments (www.ti.com) MSP430-series microcontrollers have I/O ports with configurable interrupt logic for each bit. You can select the rising or falling edge of the bit as the trigger. Even with the interrupt disabled, the microcontroller can read its interrupt-request flag to determine whether an active edge has occurred. You can use this technique in place of high-rate sampling in software. You must periodically check the interrupt-request flag at a rate high enough only to keep the switch response delay to an acceptable time. Calling the routine at the frequency of a mechanical vibration isn't an issue; the interrupt logic monitors the switch between calls.

IF YOUR APPLICATION HAS NO TASK TO RUN EXCEPT IN RESPONSE TO A SWITCH, YOU CAN MAKE A VERSION IN WHICH THE CPU SLEEPS WITH THE PORT INTERRUPTS ENABLED.

You can use multiple switch inputs, provided that the switches either never change state simultaneously—for example, with a keypad—or you don't mind delaying response to a switch until all simultaneously changing switch inputs have settled. You would lose the sequence of switch operations in this case; in the debounced output, they would all change at once.

Listing 1 periodically calls function chk_sw() to check the interrupt-request flags and update the output value, debounced_sw. The time interval between calls times Ndebounce should be short enough to satisfy the required response time after the last bounce and longer than the longest time between transitions when the contacts are bouncing. In a noisy environment, making the delay too long is counterproductive because noise transients during the delay extend the delay.

If your application has no task to run except in response to a switch, it's fairly simple to make a version in which the CPU sleeps with the port interrupts enabled (P1IE=SwitchMask) and the P1IES bits set to the last input state when there is no switch input activity. A port interrupt-service routine must respond to the first input change to set P1IE=0 and set a timer to periodically call chk_sw() until chk_sw() resolves the input state. When reacting to a brief noise impulse, the CPU would wake up once for the port interrupt and Ndebounce+1 times to run chk_sw(). EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas
READERS SOLVE DESIGN PROBLEMS

Read multiple switches and a potentiometer setting with one microcontroller input pin

Kevin Fodor, Palatine, IL

The circuit in this Design Idea provides a way to convey mixed analog and digital inputs into a microcontroller using one input pin. The output of the circuit connects to a microcontroller's ADC-input pin. The circuit comprises a single variable resistor and a number of SPST (single-pole/single-throw) switches (**Figure 1**). The pushbuttons allow the user to select modes, states, or options, and the analog input provides a method of conveying an adjustable parameter. The implementation requires you to analyze a parallel resistor circuit and a voltage divider. If you carefully select the resistor values, the circuit provides a discernible analog input as well as a number of discrete pushbutton-input states.

Selecting the resistor values is a multistep process, and a spreadsheet, which you can download at www.edn.com/100422dia, aids in performing the calculations. Say, for example, that you want 5-kΩ potentiometer R_{ADJ} to produce a 0 to 100% value into the microcontroller. Typically, you would map the sampled value of 0 to 255 into a 0 to 100 value to represent a percentage. However, by selecting the values of bias resistor R_{BIAS}, you arrive at a direct analog input centered on the 0 to 255 range of the ADC—for example, 78 to 178.

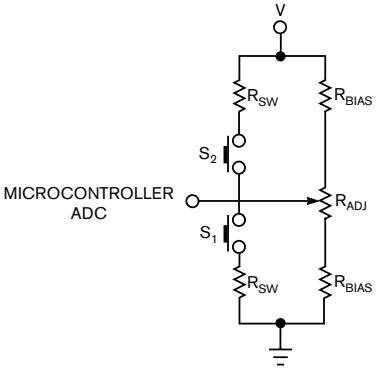


Figure 1 This circuit allows one microcontroller pin to read multiple switches and a potentiometer value.

To compute the appropriate high- and low-side bias-resistor values, the following **equations** solve this circuit as a simple voltage divider:

$$V_{LOW} = \frac{R_{BIAS}}{R_{ADJ} + 2 \times R_{BIAS}} \times V_{MAX};$$

$$V_{HIGH} = \frac{R_{BIAS} + R_{ADJ}}{R_{ADJ} + 2 \times R_{BIAS}} \times V_{MAX}.$$

Substituting and solving for R_{BIAS} and given that the maximum voltage reports a value of 255, the maximum low voltage reports a value of 78, the maximum high voltage value reported is 178, and R_{ADJ} has a value of 5 kΩ yield the following **equation**:

$$R_{BIAS} = \frac{V_{LOW} \times R_{ADJ}}{V_{MAX} - 2 \times V_{LOW}} =$$

$$\frac{R_{ADJ} \times (V_{HIGH} - V_{MAX})}{V_{MAX} - 2 \times V_{HIGH}} = 3875\Omega.$$

The computed value of R_{BIAS} is 3875Ω. Using a standard value of 3.3 kΩ, the potentiometer's input ranges from 73 to 182. This range yields a larger dynamic range than you need but allows for a guard range between the potentiometer's values and the pushbuttons' values. Because the position of R_{ADJ} affects the overall resistance the circuit sees when you press either switch, the microcontroller must interpret a range of values for each switch. To determine the switch resistance, R_{SW}, for either S_1 or S_2, you use a parallel-resistor network at both extremes of the potentiometer's position.

When you press S_1 and R_{ADJ} is at the maximum position, the effective resistance of the bottom leg of the divider is R_{SW} in parallel with the series combination of R_{ADJ} and R_{BIAS}. At the minimum position, the effective resistance is R_{SW} in parallel with R_{BIAS}:

$$R_{EFFMAX} = \frac{R_{SW} \times (R_{ADJ} + R_{BIAS})}{R_{SW} + R_{ADJ} + R_{BIAS}};$$

$$R_{EFFMIN} = \frac{R_{SW} \times R_{BIAS}}{R_{SW} + R_{BIAS}}.$$

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

You determine the value when you press S_1 by evaluating the voltage divider that R_{BIAS} and R_{RFFMAX} form:

$$V_{S1MAX} = \frac{R_{EFFMAX}}{R_{EFFMAX} + R_{BIAS}} \times V_{MAX}.$$

Observe that when R_{ADJ} is at its maximum value and you press S_1, it must produce a value less than the smallest value R_{ADJ} produces by itself to uniquely determine that you have pressed the switch. So the maximum effective resistance, R_{EFFMAX}, must produce a value less than the maximum low voltage, as the following **equation** shows:

$$R_{EFFMAX} < \frac{R_{BIAS}^2}{R_{BIAS} + R_{ADJ}}.$$

Substituting and solving this equation for the switch resistance yields:

$$R_{SW} < \frac{R_{BIAS}^3 + R_{BIAS}^2 \times R_{ADJ}}{R_{ADJ}^2 + 2 \times R_{ADJ} \times R_{BIAS}}.$$

Using the spreadsheet to compute the switch resistance yields 1558Ω, and you can choose a nominal 1.5-kΩ resistor. This selection causes S_1 to produce a range of 28 to 71 when you press it, depending on the potentiometer's position. Likewise, choosing the same value for S_2 produces a range of 184 to 227. These ranges are bands of values that you can use to determine which switch you pressed regardless of the potentiometer's position. Although selecting symmetrical resistor values is not necessary, it minimizes the number of calculations you need to perform and simplifies the design. Furthermore, selecting smaller series switch resistors opens the guard range between them and the potentiometer, which may be desirable if the resulting values are too close together. The microcontroller uses a small subroutine, **Listing 1**, which you can download at www.edn.com/100422dia, to determine both switch positions and the potentiometer's setting.

The limitation of this technique is that you cannot press more than one pushbutton at any time. In addition, the microcontroller can read the potentiometer's position only when you are not pressing any other pushbuttons. This example shows how to use two pushbuttons, but the number of pushbuttons can vary. Input ranges are available for as many as 10 pushbuttons and one potentiometer, all of which share the same input pin (**Figure 2**). Although the computed ranges do not overlap and are unique, it is doubtful that your ADC hardware can reliably distinguish these bands under all circumstances. Choosing smaller resistor values keeps these bands farther apart, creating a larger guard range.

Using this technique with four pushbuttons and one potentiometer is well within reason. Experimenting with the spreadsheet helps make quick work of determining just the right series-resistor values for each switch and its output range. EDN

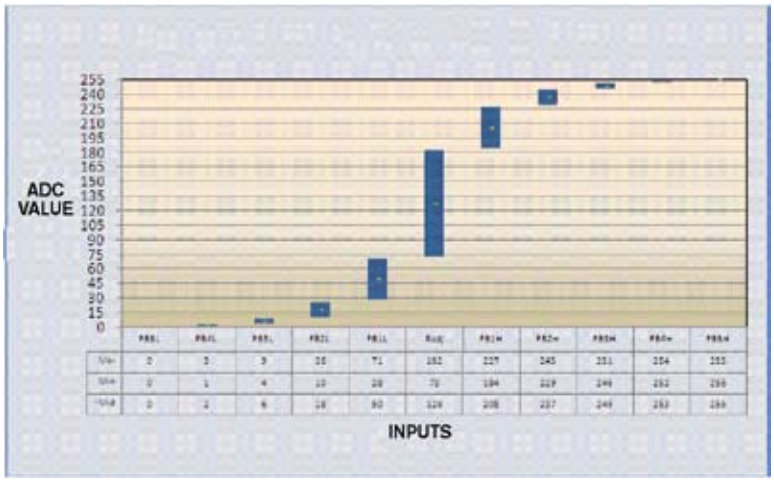


Figure 2 The circuit can have 10 pushbuttons and one potentiometer.

Three-transistor modulator-amplifier circuit works with swept-control frequencies

Horia-Nicolai L Teodorescu and Victor Cojocaru,
Gheorghe Asachi Technical University, Iasi, Romania

Many applications require a circuit to perform pulse modulation and voltage amplification to drive a load with a train of impulses. A typical application is driving a piezoelectric generator in a robot. Other applications include driving small motors or LEDs. Echolocation and ultrasound visualization use a sweeping-frequency, or chirp, signal. Nonlinear distortion is not important in these applications. When you drive a piezoelectric load, its natural resonance removes any frequency components other than the fundamental. This circuit combines a modulator and an amplifier into a single stage. The compactness of the circuit makes it appropriate for portable-system applications.

The load is in series with two switches (**Figure 1**). The input signal controls S_2, S_3 controls S_1, and the modulating signal controls S_3. This circuit's mod-

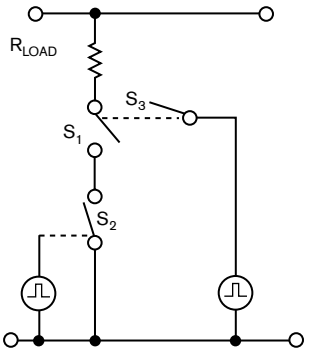


Figure 1 This simple modulator uses three switches.

ulation operation is similar to that of an AND gate. The switches must have internal resistance to dissipate the harmonics that the resonant load reflects. This circuit uses transistors Q_1 and Q_2 as switches, although they operate in the active region (**Figure 2**). Their operation resembles that of controlled resistors, and they perform voltage and current amplification. You drive Q_2 with a 42-kHz signal that matches the load's resonance. You modulate the Q_3 transistor with a periodic low-frequency impulse signal. These impulses open Q_3, which drives Q_1 and Q_2 toward saturation. When Q_3 opens, it drops the voltage across the base of Q_1, blocking

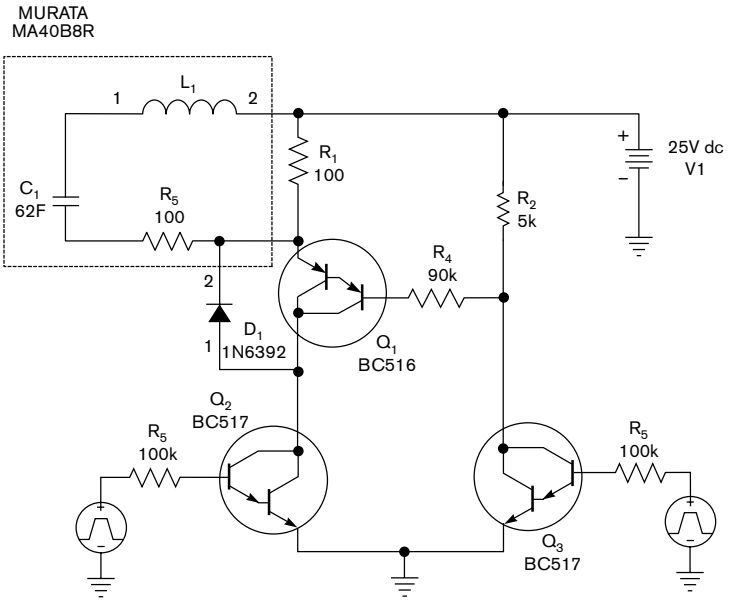


Figure 2 A three-transistor modulator with a resonant load works over a large input range.

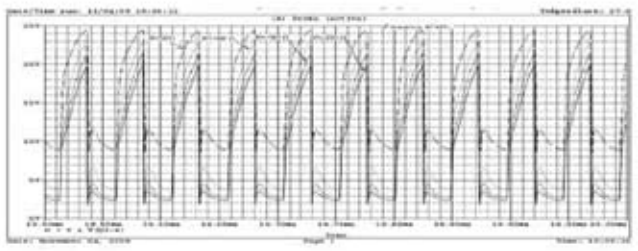


Figure 3 Changing the value of R_1 yields different response waveforms.

IF THE LOAD IMPEDANCE VARIES, THE CIRCUIT DOES NOT DEGRADE THE IMPULSE SHAPE.

the state of Q_2. Q_1 and Q_2 operate conjointly; Q_1 conducts only when Q_2 is conducting. You can view this scheme as a differential amplifier in which the signal in one branch controls the load of another branch.

Q_2 and Q_3 operate over large signals yet remain in the active region most of the time. The resistor values in the base and collector of Q_1 are critical. When the frequency of the signal is higher than the load's resonant frequency, D_1 protects Q_1 from the effects of L_1 and of harmonics on the LC circuit. The collector voltage has a spectrum rich in harmonics due to the nonlinear behavior of transistors. This characteristic is not a serious disadvantage because the resonant load removes the harmonics.

The value of R_1 is critical to the current and voltage amplification of the Q_1/Q_2 stage. The swing of voltage in the collector of Q_1 is sensitive to the value of R_1 (**Figure 3**). Q_1 operates in the active mode because its collector voltage increases slowly toward the maximal value. The significant glitch at small collector voltages shows that the blocking process partly occurs in the active regions of Q_2 and Q_3. If the load impedance varies, the circuit does not degrade the impulse shape. This situation is true even at twice the load's resonant frequency. The circuit functions with input voltages of 4.5 and 11V. This voltage range allows you to drive the circuit with a 5V microcontroller (**Reference 1**).EDN

REFERENCE

1 Teodorescu, Horia-Nicolai L, "Algorithm for Adaptive Distance Estimators for Echolocation in Air," International Solid-State Circuits Conference, 2009, www.adbiosonar.ugal.ro/ad/content/funding.

Tables ease microcontroller programming

Abel Raynus, Armatron International, Malden, MA

When creating microcontroller firmware, you often need to work with data arrays. Tables make easy work of data arrays, such as those for digital-code transformation, correction for sensor linearity, sophisticated calculations, and multiple output organization. **Table 1** shows how you can organize data in a table. Outputs A, B, and C have values based on the input value, V.

When using a lookup table, choose the proper microcontroller input and outputs. Assign values for input and outputs data in **Table 2**. These data can consist of constants in binary, hexadecimal, or decimal format or names. For names, you should assign a constant value to each one. For example:

```
data1  equ  $0a
data2  equ  $0b
data3  equ  $0c
data3  equ  $0d
```

Next, put the data from **Table 2** in either the beginning or the end of ROM, which makes the data easy to find. For definition of 1-byte data storage, use pseudo operators FCB or DB. For storage of data comprising 2 bytes, use FDB or DW, as in the following example:

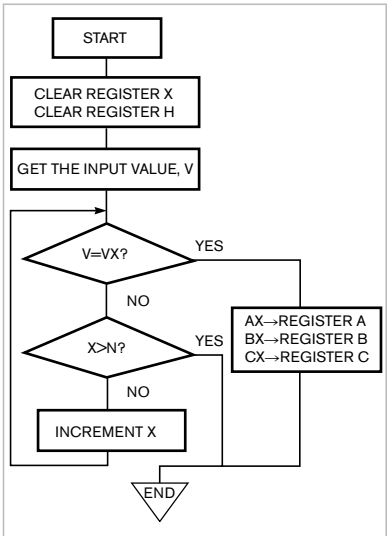


Figure 1 You can use a look-up table in microcontroller code.

```
ORG ROM
Vx  FCB  0T,2T,4T,6T
Ax  FCB  data1,data2,data3,data4
Bx  FCB  $aa,$bb,$cc,$dd
Cx  FDB  $1122,$3344,$5566,$7788
```

Note that commas separate the data. Don't place a comma after the last data, or it will be considered as $00.

When working with tables, you should always use indexed addressing mode. It provides access to data using variable addresses. Most microcontrollers have two index registers, X and H. Register X contains the low byte of the conditional address of the operand; H contains the high byte. The algorithm of working with tables is straightforward. After you detect the input value, you should then compare it with the table's input data. The X index determines this value, starting with X=0 and ending with X=N. In this example, N=4. When you find table data equal to the input value, you use the corresponding X as an index to load the output registers with their values. In the case of 2-byte numbers, you should load the output registers separately, first with a high byte and then with a low one. **Figure 1** illustrates this process.

The **listing** of assembler code is available from the online version of this Design Idea at www.edn.com/article/100422dib. In the **listing**, you can double-check the table content in memory at addresses $F800 through $F813. The listing uses Freescale (www.freescale.com) assembler because most of the appropriate applications employ inexpensive, 8-bit microcontrollers from Freescale's HC08 Nitron family. You can, however, use this approach with any type of microcontroller and assembly language. EDN

TABLE 1 OUTPUT VALUES VERSUS INPUT VALUES

Input V	Output A	Output B	Output C
V1	A1	B1	C1
V2	A2	B2	C2
....	....	....	....
VN	AN	BN	CN

TABLE 2 INPUT AND OUTPUT VALUES

Input V	Output A	Output B	Output C
V1=0T	data1	$aa	$1122
V2=2T	data2	$bb	$3344
V3=4T	data3	$cc	$5566
V4=6T	data4	$dd	$7788

Monitor alarm and indicator display multiple deviation boundaries

William Grill, Riverhead Systems, Lenexa, KS

A low-cost monitor can visually indicate a process problem, such as a failed cabinet fan or other high- or low-temperature characteristic. The microcontroller-based circuit in **Figure 1** provided a simple visual indication of both the direction and the magnitude of the temperature's deviation from a user-set mean in a solder pot. Using a Microchip (www.microchip.com) 12F675 controller, the coded sequences allow the user to both set the mean and scale the range of the monitored variation. The application uses the controller's internal clock and

two of the controller's four ADCs.

Asserting switch S_1 on Pin 4 copies the input voltage under test from Pin 7, which becomes the mean value. The code then evaluates the input-voltage deviation from the mean and applies scaled boundaries to a corresponding display format. The processor monitors both the input under test and a second analog level, on Pin 6, to scale the internal deviation/boundary tables. It then schedules as many as four sequences of one or both LEDs. The monitor also asserts an output on Pin 5 when the measured variation exceeds the third tabled boundary.

The circuit provides independent positive- and negative-deviation tables and multiplies the ranges by interpreting the voltage on Pin 6, resulting in the application of a multiple from one to eight on the boundary limits. You configure the converter reference to use the controller's V_{DD} voltage. Using only 8 bits of the controller's 10-bit ADC, the deviation can be as small as one step or $1/256 \times V_{DD}$, the drain-to-drain voltage. For a 5V reference, this voltage is approximately 9 mV.

Figure 2 shows the boundaries and their possible spans, which Pin 6 and corresponding display-format numbers set (**Table 1**). Using the provided minimum value of the deviation/boundary table, neglecting the error that results from the use of the 78L05 as a reference, and assuming the scaling derived from Pin 6 result in ×1, the first display-format step, in this application, which occurs when the measured input deviates more than the deviation/boundary-table value times the scale derived from Pin 6 times 1/256 times the drain-to-drain voltage equals 2×5/256×1, or 39 mV.

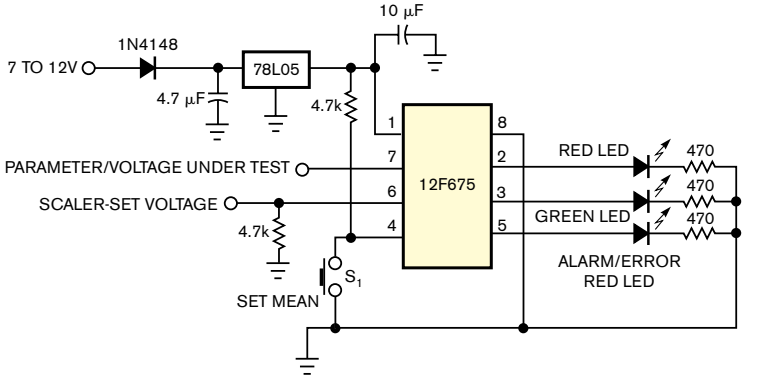


Figure 1 This microcontroller-based circuit provides a simple visual indication of both the direction and the magnitude of the temperature's deviation from a user-set mean in a solder pot.

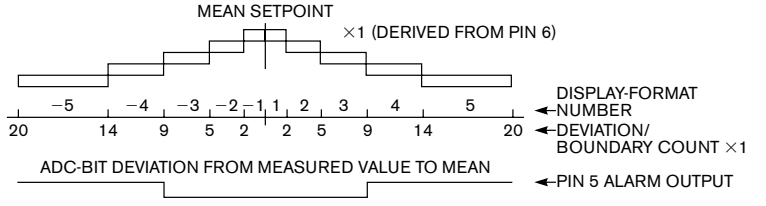


Figure 2 Pin 6 and the corresponding display-format numbers set the boundaries and their possible spans.

You can change the display-sequence formats for the five positive boundaries, beginning in a green-LED flash, and five negative boundaries, beginning in a red-LED flash, to suit simpler go or no/go applications or other needs. The circuit may also find a use in airflow or other physical-parameter monitors.

Using the controller's ADC, you can monitor any parameter that you can represent with a voltage. You can modify the code-based tables to accommodate a variety of other display sequences, parameter nonlinearities, or error distributions.

You can download **Listing 1**, code for the error monitor, from www.edn.com/100422dic. EDN

TABLE 1 DISPLAY-FORMAT NUMBERS AND TABLE-BASED SEQUENCE

Display-format number	Sequence
>5	Green, red/green, red/green, red/green
5	Green, green, green/red, green/red
4	Green, red, green/red
3	Green, green, green, red
2	Green, green, red
1	Green
−1	Red
−2	Red, red, green
−3	Red, red, red, green
−4	Red, green, red/green
−5	Red, red, red/green, red/green
≤5	Red, red/green, red/greeen, red/green

designideas

READERS SOLVE DESIGN PROBLEMS

Pulse generator with precision output-duty cycle operates at a repetition rate beyond 50 MHz

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

A previous Design Idea describes an astable multivibrator that gets hysteresis from the positive-feedback stage using a technique you can characterize as positive-capacitive feedback (**Reference 1**). It creates hysteresis by a charge transfer to its main timing-capacitor. The circuit uses two logic inverters to generate complementary logic outputs. In contrast, the circuit in **Figure 1** uses a single fast analog comparator that has complementary outputs, thus eliminating the need for an additional inverter.

Comparators such as Analog Devices' (www.analog.com) ADCMP603, IC_1, have symmetry that results in a very low time skew of the transitions at its Q and $\overline{Q}$ outputs that are fractions of a nanosecond. Thus, the charge transfer to the main capacitor, C, theoretically starts immediately at the start of the level transition at the Q output, from which C is charged through a resistor, R. No additional propagation delay occurs at any stage besides Q, resulting in a further increase in operating frequency.

The output frequency of the pulse generator in **Figure 1** is less sensitive to supply voltage variations than a generator with the ADCMP603, which uses the IC's internal hysteresis. The charging current of C and the charge-transfer-based hysteresis of the pulse generator rise almost linearly with the rising supply voltage, which results in an output signal's insensitivity.

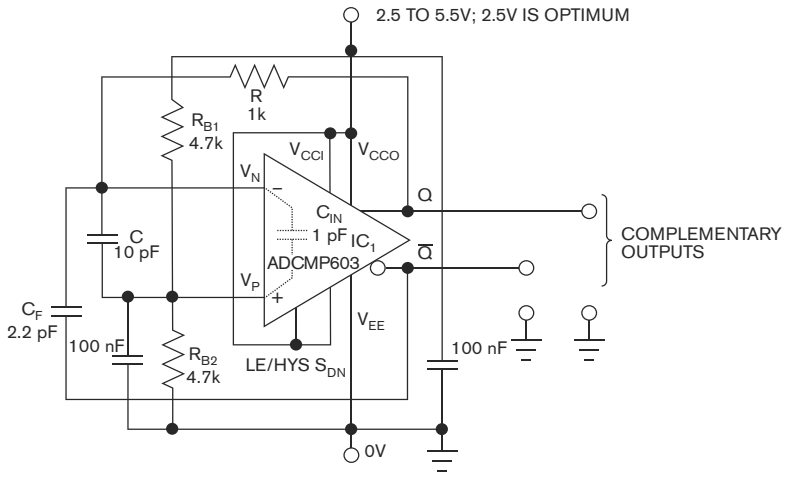


Figure 1 This pulse generator produces time-aligned complementary logic waveforms. The duty cycle holds at 50%.

Contrarily, the internal hysteresis, which an external resistor sets at the comparator's LE/HYS pin, varies nonlinearly with the supply voltage. If, for example, the external resistor's value is 225 kΩ, the hysteresis has the same value of about 36 mV for supply voltages of 2.5 and 5.5V.

At frequencies close to the upper frequency limit of operation, the charge injection through capacitor C_F is gradual rather than steplike because the rise and fall times of waveforms at the Q and $\overline{Q}$ outputs are still of finite value. **Figure 2** shows an idealized operation for the positive peak of voltage at capacitor C. Although the real voltages, V_Q and $V_{\overline{Q}}$, have somewhat rounder "corners," they resemble those in the **figure**.

If you assume that the voltage swing at capacitor C is considerably lower than that at Q and $\overline{Q}$, then a rectangular current pulse, $I_{CF}(t)$, charges capacitor C during a positive-voltage transition at output $\overline{Q}$. The current through resistor R, $I_{RF}(t)$, which also flows to capacitor C, changes its polarity at the midtransition at Q and $\overline{Q}$ outputs. The final current, which charges ca-

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

pacitor C, is a sum of $I_{RF}(t)$ and $I_{CF}(t)$. Although the waveshape of the voltage at capacitor C depends on the final charging current, it gains in peak solely due to $I_{CF}(t)$.

The following **equation** calculates the added peak:

$$\Delta V_C \approx \frac{I_{CF} \times t_R}{C_1} \approx \frac{C_F}{C_1} \Delta V_{OUT},$$

where ΔV_C is the added peak, $C_1 = C + C_{IN}$, and $C_1 \approx 1$ pF. The added peak is independent of $I_{RF}(t)$ due to the zero mean value of this current within the level transition at the Q and $\overline{Q}$ outputs.

For C=10 pF, C_1=11 pF, C_F=2.2 pF, and $\Delta V_{OUT} \approx 2.4$V, a voltage you derive from the **equation** $\Delta V_C \approx 0.48$V.

The following **equation** calculates the nearly constant part of $I_{RF}(t)$ current:

$$I_{RF} \approx \frac{1}{2} \times \frac{\Delta V_{OUT}}{R}.$$

I_{RF} determines the slope, S, of the $V_C(t)$ waveform in **Figure 2**, which excludes the time interval of level transitions at the comparator's Q and $\overline{Q}$ outputs. You calculate the slope with the following **equation**:

$$S = \frac{I_{RF}}{C} \approx \frac{1}{2RC} \times \Delta V_{OUT}.$$

The following **equation** determines the absolute value of peak voltage of $V_C(t)$, referred to supply midvoltage:

$$V_{CPEAK} = S\left(t_{PD} - \frac{t_R}{2}\right) + \Delta V_C.$$

Voltage $V_C(t)$ decreases from its peak value with a slope of −S. You calculate the time interval, T_{DESC}, when it reaches the reference level as:

$$T_{DESC} = \frac{V_{CPEAK}}{S} = t_{PD} - \frac{t_R}{2} + \frac{\Delta V_C}{S} = t_{PD} - \frac{t_R}{2} + 2C_F R.$$

By evaluating this **equation** for a t_{PD} of approximately 3.5 nsec and R with a value of 1 kΩ, the time interval is approximately 6.8 nsec.

The following **equation** calculates the total time, when $V_C(t)$ is higher than the reference voltage, $V_{CC}/2$:

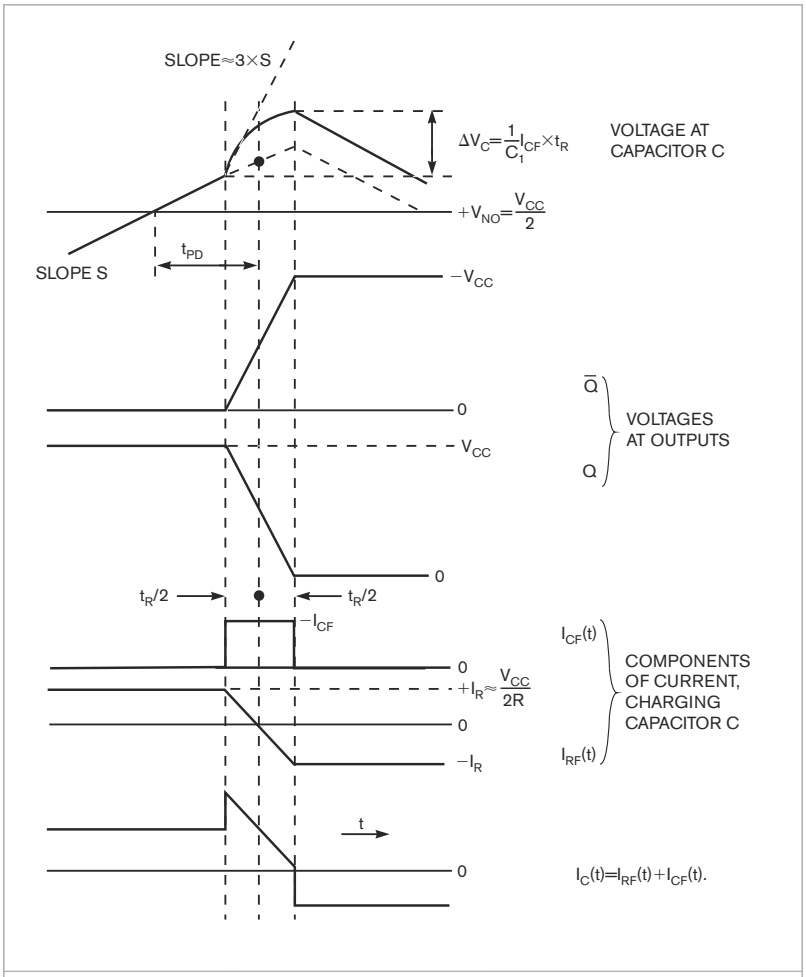


Figure 2 An idealized but still close-to-reality modeling of the operation of the generator gives you a formula for determining the output frequency.

$$T_H = t_{PD} + \frac{t_R}{2} + T_{DESC} = 2\left(t_{PD} + C_F R\right).$$

The symmetry of the ADCMP603's internal circuit architecture, T_H, is the right half-period of logic waveforms at the Q and $\overline{Q}$ outputs. In other words, the duty cycle of the output pulse is 50%. By evaluating the **equation** for T_H, you get 11.4 nsec. Thus,

$$f_Q = \frac{1}{2T_H} = 43.86 \text{ MHz}.$$

The circuit's output frequency is 56.75 MHz with a power-supply voltage of 2.052V. With a supply voltage of 3.51V, the frequency changes to 56.12 MHz. Thus, the relative sensitivity of the output frequency to the supply-voltage variation is approximately 8×10^{-3}/V. You can attribute an increase of experimental frequency as compared with a theoretical value to the fact that, during the estimated signal-propagation delay, t_{PD}, the comparator's input overdrive rises gradually to about 330 mV, which is more than triple the value at which you define the propagation delay. You can therefore assume a lower propagation delay and a higher frequency. EDN

REFERENCE

Larson, Robert, "Astable multivibrator gets hysteresis from positive-feedback stage," *EDN*, Oct. 22, 2009, pg 43, www.edn.com/article/CA6702271.

Bicolor LED driver uses two leads

Mario Marcoccia, United Circuits, Fort Lauderdale, FL

You can use the circuit in **Figure 1** to drive a bicolor LED with only two leads. This circuit detects the correct closed condition of the left and right side bags in a motorcycle companion. The bag has two locks that you must close for protection. When you push the two momentary SPDT (single-pole/double-throw) switches, they sense the correct closed bag. One bicolor red-and-green LED indicates the bag's status, with the red color showing the open-bag condition. To illuminate the LEDs, you must reverse the polarity of the applied voltage to the LED to change the color (**Table 1**).

Diodes D_1 and D_2 and resistor R_1 form a discrete OR gate. When either pushbutton switch connects to 12V, the voltage at Point A is positive with respect to Point B. Transistor Q_1 conducts, letting current illuminate the red LED. When neither switch connects to 12V, neither diode conducts. The base of Q_1 pulls low through R_1 and R_4, indicating that the bags are closed. Thus, the green LED illuminates as current passes through it and through R_1 and R_2.

TABLE 1 LOGIC CONDITIONS FOR THE CIRCUIT

	Switches		Bicolor LED	
Bag condition	**S_1**	**S_2**	**Red**	**Green**
Open	1	1	On	Off
Open	0	1	On	Off
Open	1	1	On	Off
Closed	0	0	Off	On

The diode, transistor, and resistor values are not critical, and you can adjust them according to your needs. You can also replace the bicolor LED with two discrete LEDs of different colors placed back to back. EDN

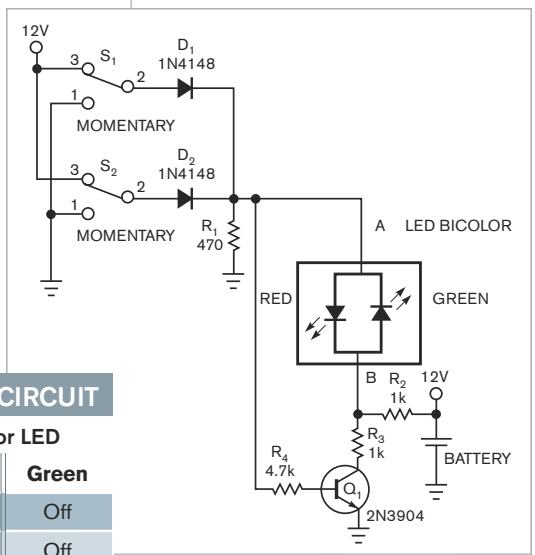


Figure 1 This circuit detects the correct closed condition of the left and right side bags in a motorcycle companion.

Supervise and power-sequence an SOC

Eric Schlaepfer, Maxim Integrated Products, Sunnyvale, CA

Microprocessors, microcontrollers, and SOCs (systems on chips) often need a reset pulse to initialize properly. Many of these devices also use separate I/O- and core-voltage supplies. When you use multiple supplies, you must turn them on in a specific sequence to prevent the circuits from ending up in an unknown state or burning out due to unexpected current paths. You should also monitor the voltages to ensure that the device does not come out of reset until both voltages settle to levels within the operating-voltage range.

A previous Design Idea presents a circuit performing the reset function (**Reference 1**). Unfortunately, this circuit suffers from a number of limitations. For example, it does not monitor the voltage on the 3.3V rail. The 3.3V rail acts as a reference, so the 1.8V rail suffers from poor monitoring accuracy. Further, the reset delay may not be present if you sequence the power rails in the reverse order, and the reset pulse has a glitch that could cause problems with the SOC. Finally, the reset-delay capacitor may reset incorrectly if you rapidly cycle power.

The circuit in **Figure 1** uses a reset IC to provide a glitch-free reset pulse with a well-defined pulse width. It accurately monitors both the 3.3 and the 1.8V rails. You adjust resistors R_1 and R_2 to set the monitoring threshold for dif-

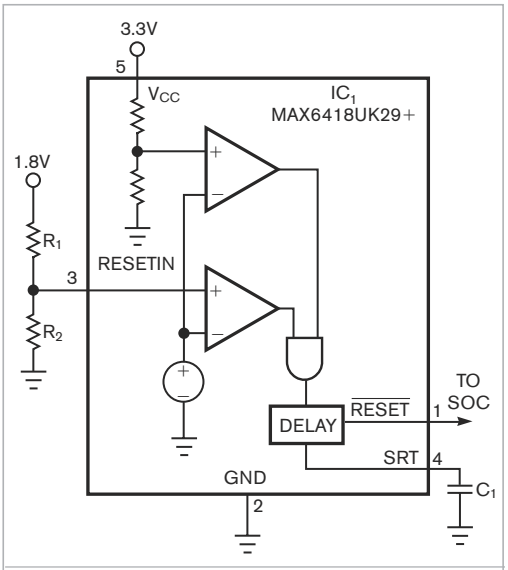


Figure 1 This circuit provides a microprocessor or an SOC with a clean reset pulse.

ferent core voltages using the **equation** $V_{TH}=1.263\times(1+R_1/R_2)$, where V_{TH} is the threshold voltage. You adjust C_1 for different pulse widths. You calculate C_1 using the following formula: $C_1=(t-275\times10^{-6})/(2.73\times10^6)$, where t is the desired delay in seconds and C_1 is in farads.

Two other previous Design Ideas present circuits for sequencing the two power-supply rails (**references 2** and **3**). One circuit requires many components to achieve a simple function, and the other circuit requires a microcontroller-firmware-development tool set. A simpler alternative implements a sequencing circuit using two ICs acting as voltage detectors (**Figure 2**). This circuit is useful when performing experiments to determine the proper sequencing order. You adjust R_1 and R_2 to set the sequence delay for each power rail. Each IC monitors the voltage on the RC circuit and asserts its output when the capacitor voltage crosses the threshold.

Once you determine a sequence order, you can implement the power sequencing with a single IC (**Figure 3**). This approach monitors the output voltage of the previously sequenced power supply before enabling the next power supply. It also monitors the 5V rail. For cost-sensitive applications, you can devise a passive circuit (**Figure 4**). It works, although the sequence delay is not well-controlled and the voltages are not monitored. In this case, the open-drain output from the first supply holds the node low until power is good. When the open-drain output releases the node, C_1 charges through R_1, asserting the active-high enable of the second supply. You choose R_1 and C_1 to determine the sequence delay. EDN

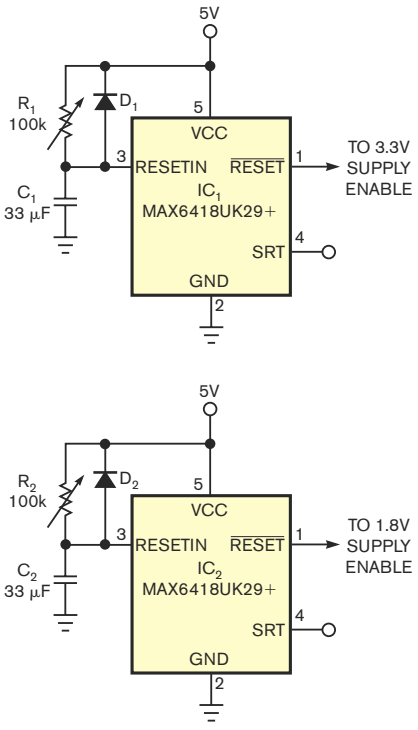


Figure 2 Sequence two power supplies using R_1 and C_1 to set the delay.

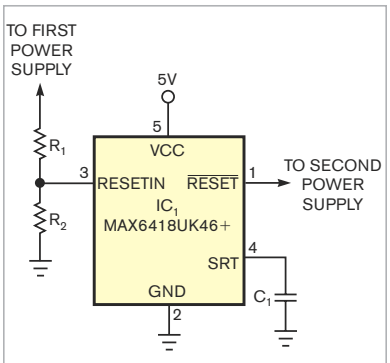


Figure 3 C_1 sets a delay to sequence two power supplies.

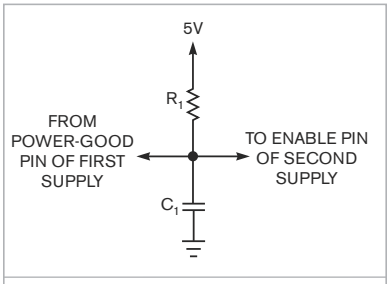


Figure 4 Use this circuit to try different sequencing delays and adjust the sequencing order.

REFERENCES

1 Ban Hok, Goh, "Reset an SOC only when power is ready," *EDN*, Dec 3, 2009, pg 40, www.edn.com/article/CA6709555.

2 Karmann, Dan, "Circuit provides simpler power-supply-sequence testing," *EDN*, Dec 3, 2009, pg 44, www.edn.com/article/CA6709556.

3 Ban Hok, Goh, "Circuit eases power-sequence testing," *EDN*, July 9, 2009, pg 50, www.edn.com/article/CA6668618.

Circuit may save property and life

Vladimir Oleynik, Moscow, Russia

The circuit in **Figure 1** sounds an alarm whenever the perimeter-protection line—a thin copper wire of appropriate length—breaks. You can get the thin wire from a used relay coil. Place the wire around a tent, a camping area, or another area. The wire is thin and invisible in the dark, especially when you place it in grass as high as 0.5m. Because the wire's resistance is small relative to the 200-kΩ resistor, the two form a voltage divider that keeps the MOS transistor off. When a human or an animal intruder breaks the protection wire, a transistor switches on and a buzzer sounds. Because the blinking LED controls the transistor, the buzzer sounds at the LED's switching frequency. The protection wire's resistance can be as great as 33 kΩ.

The circuit operates at a battery voltage as low as 4.8V. In standby mode, it consumes less than 50 µA. When the buzzer sounds, the circuit consumes about 30 mA using a 9.6V battery; this value decreases while the battery drains. The circuit uses a 9V-dc Varta Longlife Extra alkaline battery and operates in standby mode 24 hours a day, with a test alarm switching on for 10 seconds daily. Under those conditions, the daily battery voltage drain is 0.02V, so you can estimate battery life at about eight months while the battery voltage

declines from 9.6V when the battery is new to 4.8V when it is exhausted. If there were no alarm switching, the battery would operate for several years.

The circuit in **Figure 2** reduces current consumption to less than 0.5 µA and lets you use longer protection wire, thus enlarging the protected area. The protection wire can have resistance as great as 3.3 MΩ, and the circuit still works. Thus, you can calculate a possible perimeter-protection-wire length according to the appropriate wire diameter and copper resistance (**Reference 1**). Both circuits can stay in standby mode for years. EDN

REFERENCE

1. "Reference & Information, AWG Cable Description," American Wire Gauge, www.interfacebus.com/Copper_Wire_AWG_SIze.html.

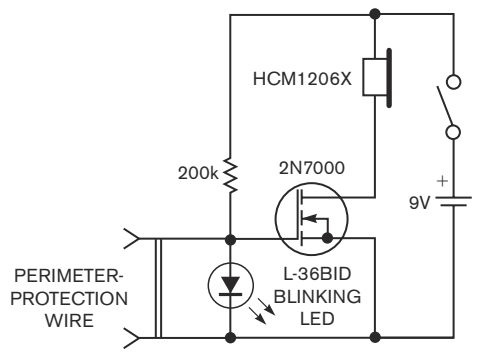


Figure 1 When the perimeter wire breaks, an LED lights and an alarm sounds.

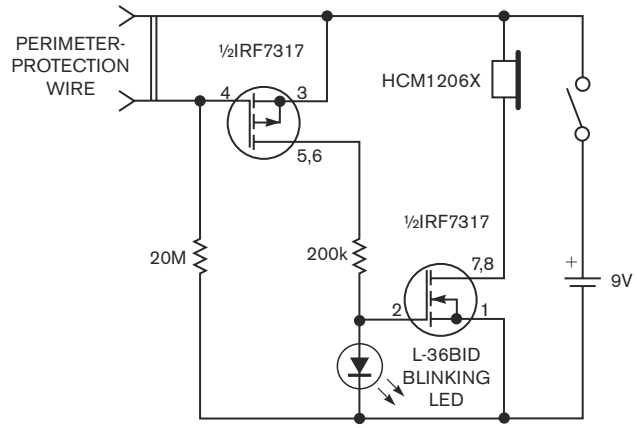


Figure 2 Adding a P-channel MOSFET cuts current consumption.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Photoresistor provides negative feedback to an op amp, producing a linear response

Julius Foit and Jan Novák,
Czech Technical University, Prague, Czech Republic

AGC (automatic-gain-control) amplifiers use the nonlinear characteristics of control devices. The magnitude of the real component in some of their differential parameters changes depending on variations in their dc operating points. A typical example is the VA characteristic of a silicon PN junction, which results in the differential conductance directly proportional to the passing dc current (**Reference 1**). In this form of control, the main problem is the control element's nonlinear transfer characteristic, which causes a relatively large degree of nonlinear signal distortion once the processed voltage amplitude exceeds millivolts (**Reference 2**).

A photoresistor, which has a VA characteristic that's linear in a large range of voltages, is up to the task. Common photoresistors remain perfectly linear for signal amplitudes of 100V or more. Therefore, the amplification-control device can be an optocoupler whose controlled element is a photoresistor.

Figure 1 A metal tube with an HB LED and a photoresistor forms the optocoupler (left).

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

The circuit in this Design Idea uses a radiation source whose spectral characteristic fits the spectral characteristic of the photoresistor, and its radiated power should, if possible, be a linear function of the drive signal. Such optocouplers are commercially available, but few have properties good enough for this purpose. Common photoresistors have spectral characteristics close to the spectral characteristics of the human eye, whose peak sensitivity has approximately a 500-nm wavelength. So a white or green LED (light-emitting diode) is a good alternative. To obtain the highest possible sensitivity, this circuit uses a white HB (high-brightness) LED.

Figure 1 shows the individual components of the optocoupler and the assembled device. The optocoupler comprises a cylindrical holder that accepts a standard 5-mm HB LED from one end and a photoresistor at the other end. An opaque nonconductive seal prevents external light from entering the device. The polished metallic inner wall of the holder results in minimum light loss between the LED and the photoresistor. Available off-the-shelf photoresistors include the LDR 05, the LDR 07, and a standard white, 5-mm HB LED type L-53MWC*E, with output-light flux of 2500 mcd at a 20-mA drive current (**Reference 3**).

Figure 2 shows the transfer function of the optocoupler using the LDR 07-type photoresistor. The output resistance of the device can vary from 100Ω to 10 MΩ with LED-drive currents from 34 mA to 0.1 μA, respectively. The photoresistor's linear VA characteristic, even for large-amplitude signals, lets you use it as the control element even in situations that require a relatively large signal voltage, such as when the photoresistor is part of the feedback loop of an operational amplifier. **Figure 2** also shows that you can obtain a variation of linear output resistance over at least five decades with a maximum LED-drive current within the limits of permitted output current of common monolithic operational amplifiers.

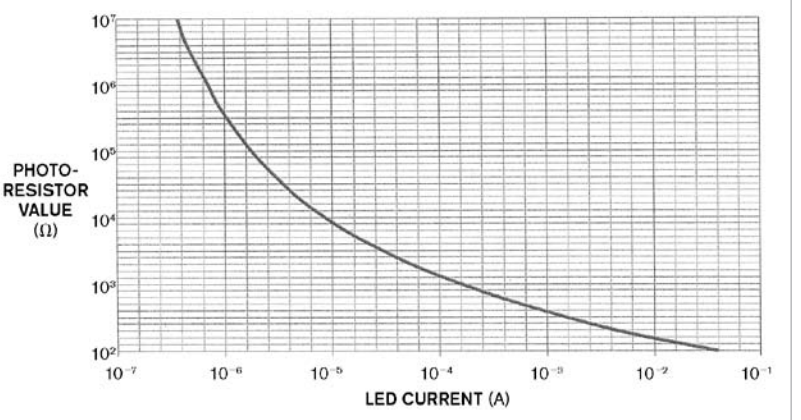


Figure 2 The optocoupler's logarithmic response in a feedback loop produces a linear amplifier response.

Such an amplifier can control the overall amplification of the system in the same range without additional current amplification. Due to the photoresistor's linearity, the resulting degree of processed signal nonlinear distortion is almost solely due to the nonlinearity of the operational amplifier. Within the normal operating range, the overall linearity of the system improves with increasing input-signal amplitude because the amount of negative feedback increases with increasing signal amplitude.

Figure 3 shows the amplifier system. The basic signal-processing device is inverting op amp A_1. Its inverting connection lets you set the absolute value of the overall amplification from input to output to a value smaller than unity, permitting correct processing of an input-signal amplitude even larger than the regulated output value. Optocoupler IC_1 is the core component of the system, whose output, the photoresistor, serves as a variable part of A_1's negative-feedback network. At no-signal conditions, the LED does not illuminate the photoresistor. Thus, its resistance rises to a high value, which can cause dc runaway and the loss of the quiescent operating point of A_1. Such a condition is not harmful in principle because the signal path is ac-coupled, preventing the dc error value from getting any further. When a nonzero signal suddenly appears at the input, however, A_1's open-loop amplification would amplify it, causing a rapid rise in LED current. This action would drop the optocoupler's output resistance almost stepwise to a value sufficient to restore the dc operating point of A_1. The ac coupling transfers this transient to the output, and it may cause problems in signal-processing circuits following the adaptive amplifier. To prevent this effect, you should limit the maximum value of the feedback resistance to a reasonable value, such as 47 MΩ,

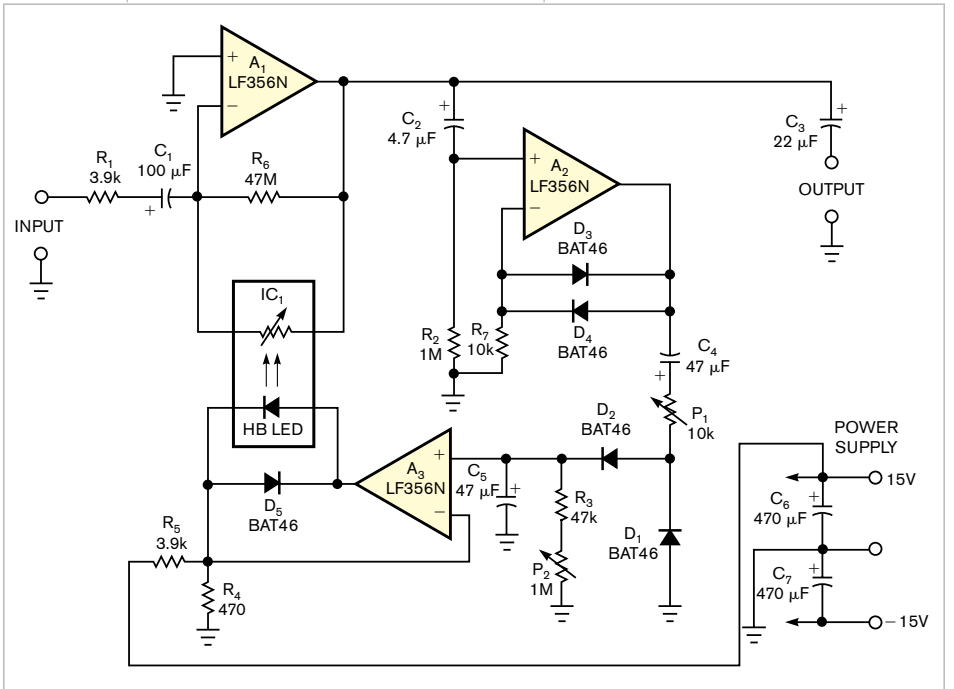


Figure 3 The adaptive-amplifier system has the optocoupler in a feedback loop.

the value of R_6. Because the op amps have JFET inputs, the value of R_6 can be rather high. The value of 47 MΩ is a reasonable compromise, limiting the maximum absolute value of voltage amplification in A_1 to approximately 82 dB. The limiting factors for selecting a value for R_6 are the noise and the open-loop amplification of A_1.

Buffer A_2 separates the nonlinear load through the rectifying diodes from the output signal, thus preventing the nonlinear load from the rectifying diodes from distorting the output signal. Diodes D_3 and D_4 compensate the threshold voltage, including its temperature coefficient, of rectifying diodes D_1 and D_2. If you do not need to set the regulated output-voltage amplitude to a value smaller than the threshold value that the bias current in R_4 sets, you can replace D_3 and D_4 with a short circuit and omit R_7. You can set a larger-than-unity voltage amplification in A_2 to obtain a regulated output amplitude lower than the threshold that the bias in R_4 sets. Just insert an additional resistance in series with the D_3/D_4 pair.

The rectifier uses Schottky diodes, which have a lower threshold voltage than conventional PN diodes. They also have a short recovery time, keeping the same rectification efficiency at high signal frequencies. The rectifier operates as a full-wave voltage doubler, providing peak-to-peak rectification even for signals with nonsymmetrical waveforms. The rectifier output feeds to A_3, a voltage-to-current converter, which drives the LED in the optocoupler. A rectification threshold-shifting bias-current source connects to current-sensing resistor R_4. In this case R_5 simulates a current source, setting the regulated output-voltage amplitude. If the 15V supply voltage isn't perfectly stable, obtain bias current from a separate stable source. An opposite-polarity diode connects across the optocoupler's input to protect the LED from reverse polarization at no-signal conditions.

This LED current-control circuit has an important advantage: It permits an almost-independent adjustment of the attack and release time. You can adjust the attack time through variable resistor P_1, using a higher value if necessary. You can also adjust the release time using P_2. The photoresistors used have a rather good response speed, and the introduced delay at a stepwise illumination variation is acceptable for most practical requirements.

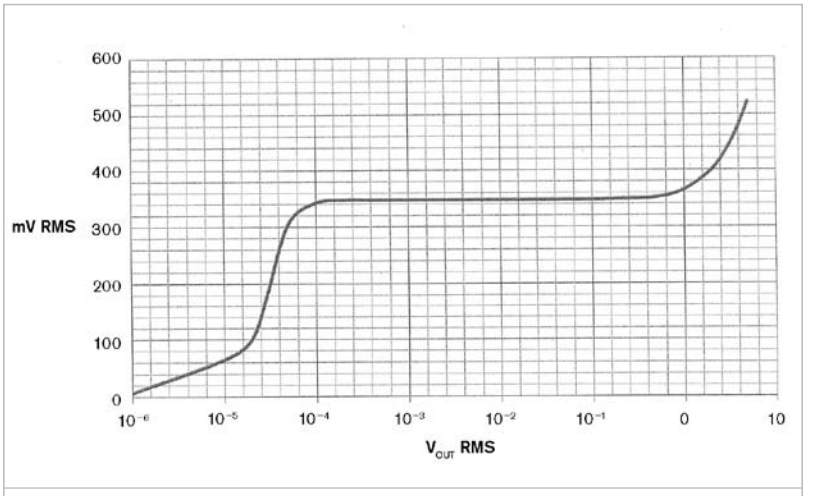


Figure 4 The amplifier system has a constant output from 0.1 mV to 1V-rms input.

Figure 4 shows the overall response of the adaptive amplifier system. The output signal remains constant at 350 mV rms ±1 dB for input-signal voltages of less than 70 µV rms to more than 1.2V rms—that is, over a more-than-85-dB range. The no-signal output noise is less than 6 mV rms, yielding an SNR (signal-to-noise ratio), or processed-signal dynamic range, better than 20 dB at the onset of regulation in the worst-case condition and improving proportionally with increasing input-signal level.

The key parameter this design follows is its linearity. Because of the photoresistor's linearity and the separation of the nonlinear rectifier load from the output, the gain control introduces negligible nonlinearity. Thus, A_1 alone, in principle, determines the overall linearity of the system.

Harmonic analysis of the output signal at 1 kHz yields higher harmonics with amplitudes lower than A_1's noise level for all input voltages to 200 µV rms and below −75 dB for input voltages to 1.5V rms. The nonlinear distortion becomes noticeable only at large input amplitudes exceeding the regulation range of the system, raising the second harmonic to −45 dB and the third harmonic to −40 dB at 2.5V-rms input.

Within the AGC's range limits, the overall transfer linearity improves with increasing input-signal amplitude due to the increasing degree of negative feedback to A_1 at increasing input-signal amplitudes. With a value of 10 kΩ for P_1 and 1 MΩ for P_2 and a stepwise input-signal variation between 100 µV and 50 mV rms, the attack and release times are approximately 0.2 and 2 seconds, respectively. The recovery time from a 1-kHz—more than 10V-rms input overdrive—to full no-signal sensitivity is less than 2 minutes. You can adjust all of these time intervals in a wide range by varying the values of C_4, C_5, P_1, and P_2, with P_1 setting the attack time and P_2 setting the release time. EDN

REFERENCES

1 Foit, Julius, "AGC amplifier features 60-dB dynamic range," *EDN*, Aug 4, 2005, pg 87, www.edn.com/article/CA629309.

2 Foit, Julius, "Logarithmic Processing Amplifier," *Proceedings of the Fifth WSEAS International Conference on Microelectronics, Nanoelectronics, Optoelectronics*, March 2006, pg 6.

3 *Opto-isolator Catalogue*, Tesla Blatná, www.tesla-blatna.cz.

Three-phase digital-signal generator sweeps frequency

Yi-Chu Liao and Shao-Wei Leu,
National Taiwan Ocean University,
Keelung, Taiwan

Many power ICs use frequency jitter, which spreads a control signal's spectrum, to control EMI (electromagnetic interference). If you need to add frequency jitter to power ICs that control three-phase signals, you can use an FPGA and the code in this Design Idea, which is available at www.edn.com/100527dia. The digital three-phase signal sweeps over a 20-kHz range of 100 to 120 kHz and back in 40 steps in 500 msec (**Figure 1**). The basic clock frequency can range from 600 to 720 kHz. You can develop the three-phase generator using Verilog (www.verilog.com) HDL code and Altera (www.altera.com) FPGAs.

The three-phase generator starts with a 50-MHz clock and ends with three output phases (**Figure 2**). The circuit's two main parts are the sequential frequency-scanning part and the three-phase model. The frequency-scanning part comprises the frequency model and Model 2 blocks. The sequential frequency-scanning part generates a frequency of 600 to 720 kHz. The three-phase model receives the variable clock-period frequency from the output of Model 2. One phase period comprises six clock periods from the three-phase input frequency. The relations of phases 1, 2, and 3 are 101, 100, 110, 010, 011, and 001, respectively, over six clock pulses. Together, they construct a three-phase waveform with a 120° phase difference.

Using the source code, you can implement the circuit using the Altera FPGA DE2 development tool, which has a basic frequency of 50 MHz, to control these circuits. The three-phase frequency sweeps in 40 1-kHz steps from 100 to 120 kHz and back in 0.5 seconds (**Figure 3**). The sequential frequency-scanning part generates an 80-Hz, 20-steps-up/20-steps-down sweep

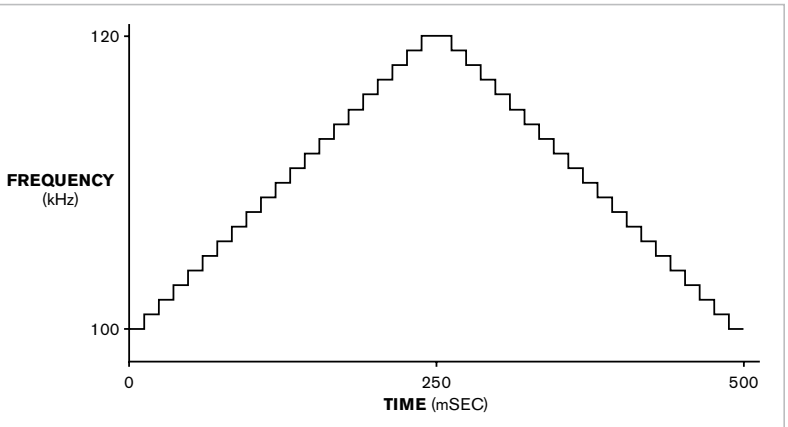


Figure 1 This digital three-phase signal sweeps over a 20-kHz range of 100 to 120 kHz and back in 40 steps in 500 msec.

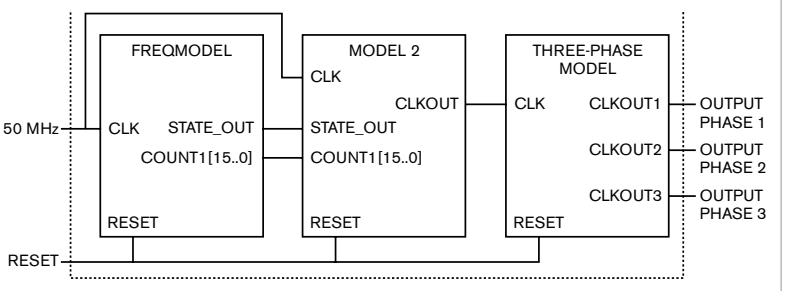


Figure 2 The three-phase generator starts with a 50-MHz clock and ends with three output phases.

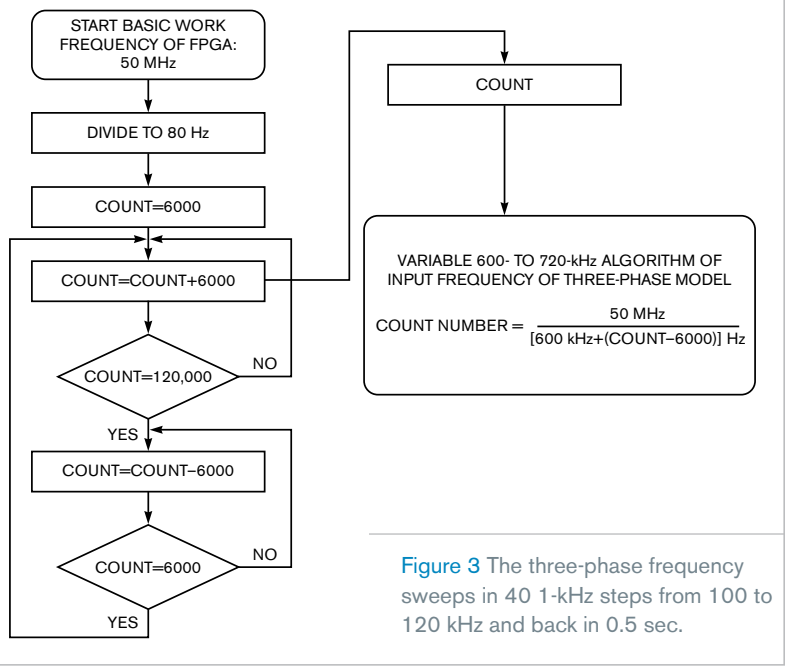


Figure 3 The three-phase frequency sweeps in 40 1-kHz steps from 100 to 120 kHz and back in 0.5 sec.

by the internal divider from the 50-MHz basic frequency. To obtain the 100- to 120-kHz frequency as the output frequency of the three-phase signal, you first generate clocks of 600 to 720 kHz as the input clock of the three-phase generator because one of the output-phase clock periods should be in six equal periods of 60°. The following **equation** shows how to get the frequency shift from the basic signal frequency:

$$100\,\text{kHz} \leftrightarrow 120\,\text{kHz} = \frac{600\,\text{kHz} \leftrightarrow 720\,\text{kHz}}{6}.$$

The first phase starts on the positive edge of the first period, and the second phase starts after two periods of the input clock at the positive edge. The second phase now lags by two input clock periods, or 120°. The third phase then starts after two more input-clock periods. The counter starts with a value of 6000 because the period of the three-phase model requires six clock times, and the sweeping frequency is 1 kHz. The **equation** in **Figure 3** shows how the count number derives from the 50-MHz clock signal.

The counter increases in value to 6000 in 0.0125 seconds, or 80 Hz, until it reaches 120,000. The count then decreases back to 6000. The variable "count" is the input to the Model 2 block, which generates clocks of 600 to 720 kHz—the input-clock period of the three-phase block. Finally, the three-phase frequency changes in sequence, counting up from 100 to 120 kHz and back down to 100 kHz in 1-kHz steps over 0.5 seconds. **Figure 4** shows the relationships among the three phases.

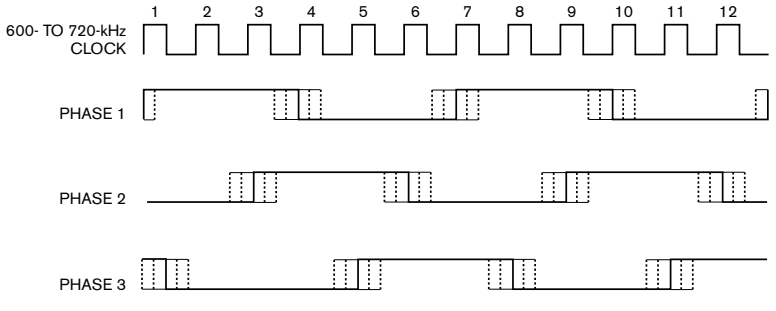


Figure 4 The three-phase frequency changes in sequence, counting up from 100 to 120 kHz and back down to 100 kHz in 1-kHz steps over 0.5 seconds.

Using this algorithm, you can develop and implement a lightweight, low-cost, three-phase signal with a 120° relative phase difference and simultaneous sweeping on one FPGA chip. You can use three lowpass filters to create sine-wave signals from the outputs. EDN

Water-leak detector uses 9V batteries

Yongping Xia, Navcom Technology, Torrance, CA

A previously published Design Idea describes a practical gadget that has the potential to save a lot with little investment (**Reference 1**). However, the circuit uses 120V line voltage and, as such, it is not that convenient for many applications. This Design Idea describes a portable water-leak detector that uses a common 9V battery for power (**Figure 1**). The circuit consumes less than 10 µA during detection mode, and a 9V alkaline battery has greater-than-500-mAhr capacitance. So one battery can last more than five years, which is equivalent to the battery's shelf life. When the battery voltage drops below 6.5V, the detector beeps to indicate that it is time to change the battery.

The design uses Maxim Integrated Circuits' (www.maxim-ic.com) MAX934, an ultra-low-power quad comparator with a built-in 1.2V reference. The chip uses about 6 µA. IC_{1A}, R_1, and R_2 provide water-leakage detection. R_1 is the water probe, which can be two bare copper wires wrapped in a sponge. R_1 has high impedance when the sponge is dry, so IC_{1A}'s output stays high. Once the circuit detects the water leak, R_1's value decreases to less than a few hundred kilohms, which forces IC_{1A}'s output low. Through D_1, it makes the output of IC_{1B} high.

THIS PORTABLE WATER-LEAK DETECTOR USES A COMMON 9V BATTERY.

IC_{1B}, R_3, and R_4 form a low-voltage detector. When the water probe is dry and the battery voltage becomes lower than 6.6V, the voltage on IC_{1B}'s negative input is less than 1.2V. Because the reference voltage is 1.2V, IC_{1B}'s output changes from low to high. So when the probe is dry and the battery voltage is higher than 6.6V, IC_{1B}'s output is low, which forces IC_{1C}'s output high, and IC_{1D}'s output stays low.

Either a wet probe or a low-voltage battery can force IC_{1B}'s output high, freeing a narrow-duty-cycle oscillator comprising IC_{1C}, C_2, R_5, R_8, and D_3. The oscillation period is approximately 7 seconds, and IC_{1C}'s output is low for about 0.3 seconds. That low output allows a 2.4-kHz oscillator comprising IC_{1D}, C_3, and R_9 to operate. When the circuit detects a water leak or the battery's power is low, the buzzer sounds for a fraction of a second every 7 seconds. In this way, the warning sound can last for a long time before the battery gets too low.

Resistors R_6 and R_7 increase IC_{1C}'s hysteresis, which lets you use a smaller value for C_2. R_{10} and R_{11} increase IC_{1D}'s hysteresis to improve the sound frequency's stability. All capacitors are ceramic, ensuring low leakage current. **EDN**

REFERENCE

Tregre, Jeff, "Doorbell transformer acts as simple water-leak detector," *EDN*, Dec 15, 2009, pg 48, www.edn.com/article/CA6711862.

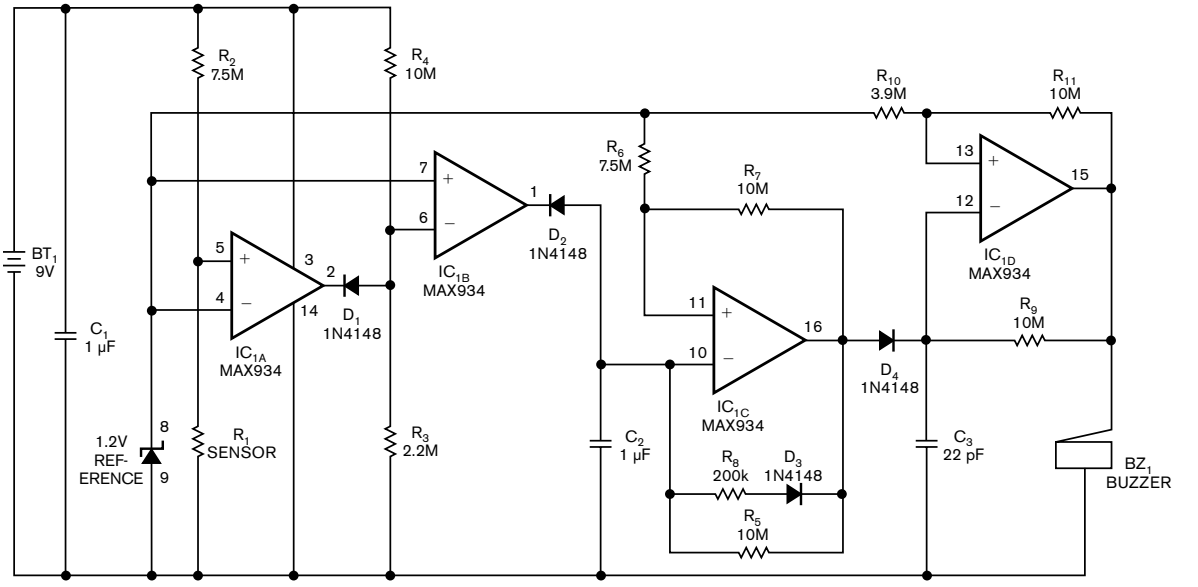


Figure 1 IC_{1A} and IC_{1B} determine the conditions for sounding the buzzer, and IC_{1C} oscillates to provide the trigger for IC_{1D}.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Isolated FET pulse driver increases power rate and duty cycle

Jaime Castelló, José M Espí, and Rafael García-Gil
University of Valencia, Spain

In power converters, pulse-drive circuits transmit the pulses a controller generates to the power transistor. Driver circuits must both transmit the controller's switching on/off signals with galvanic isolation and provide energy to turn the switch on and off and to maintain the required on or off state. The required energy increases with the power transistor's input capacitance, which also increases with the power that the transistor module manages. Thus, when the circuit requires high power, designers typically parallel the power transistors, increasing the input capacitance. When you need to operate IGBT (insulated-gate-bipolar-transistor) modules in parallel, it is best to share the gate drive because using different driver circuits introduces additional variation in turn-on and -off times and creates a possible imbalance between each power module.

The basis of the circuit in **Figure 1** is an earlier Design Idea (**Reference 1**). The operation of the circuits is basically the same as the one in the previous Design Idea, but this one can drive MOS-

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

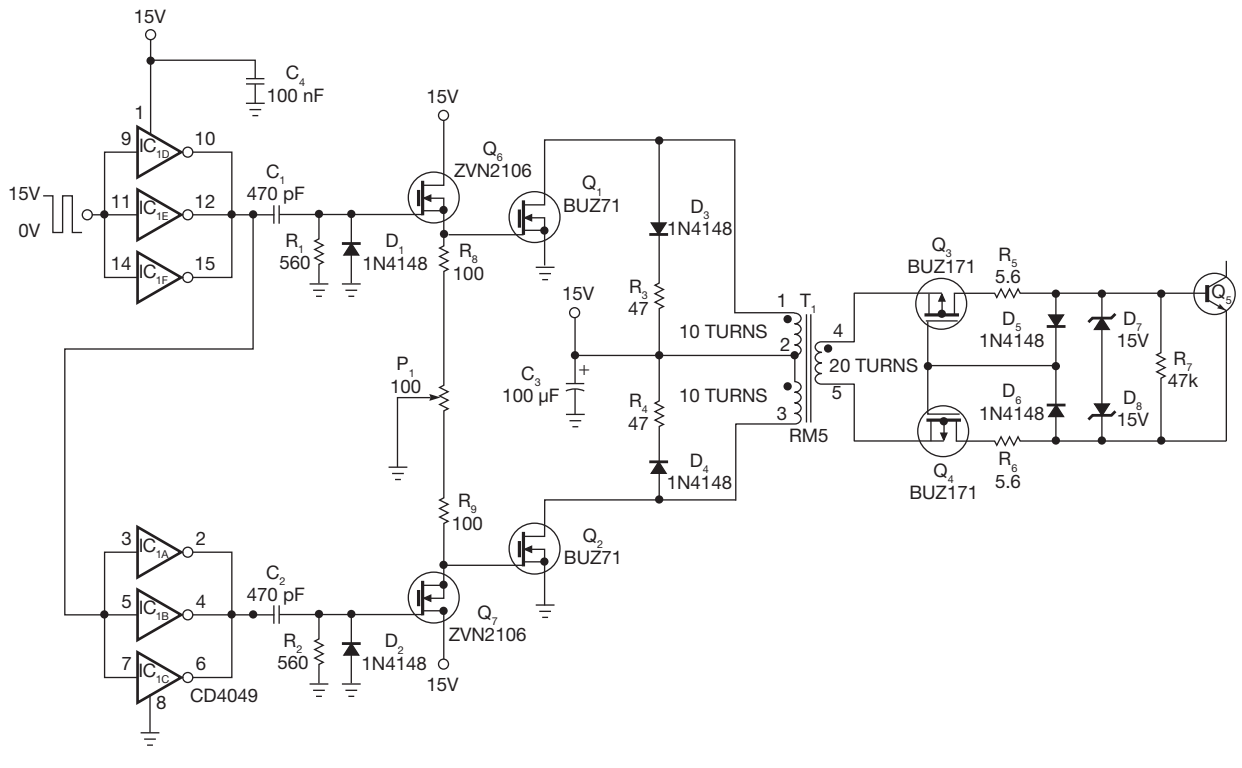


Figure 1 This isolated pulse driver can transmit all duty cycles, even with high-power MOS-FET/IGBT modules that have large input-gate capacitance.

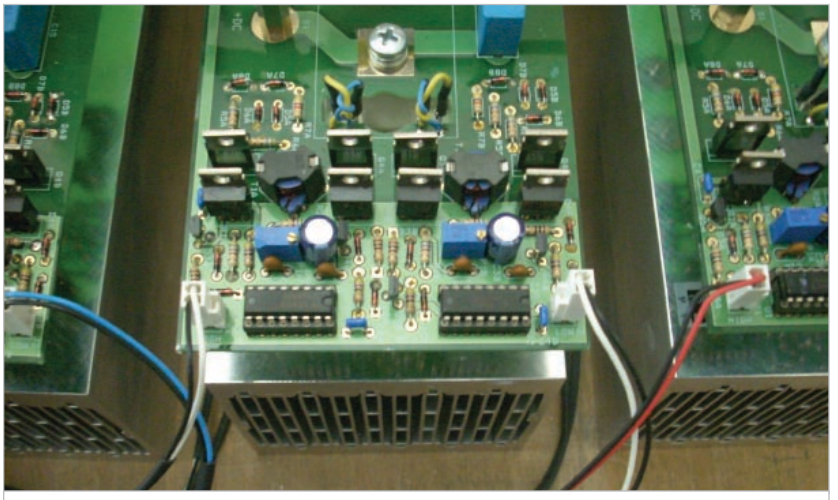


Figure 2 An isolated FET pulse driver in a 10-kW, three-phase inverter for grid injection requires few components and has galvanic isolation.

FETs or IGBTs with input capacitances higher than 5 nF. This circuit provides full galvanic isolation and requires no floating power supplies; it can transmit duty cycles that approach 100%.

This circuit adds transistors Q_6 and Q_7 to the circuit in the previous Design Idea. Transistors Q_1, Q_2, Q_3, and Q_4 are now higher power because they can manage a larger current depending on the transistor they need to drive. Transistors Q_1 and Q_2 are BUZ71 units, Q_3 and Q_4 are BUZ171 devices, and Q_6 and Q_7 are ZNV2106s. The differentiator circuits, C_1/R_1 and C_2/R_2, generate 1-µsec-long pulses, and you do not apply them directly to the gates of the Q_1 and Q_2 transistors, as in **Reference 1**, but to transistors Q_6 and Q_7. Although the input capacitances of Q_1 and Q_2 are nearly 700 pF, the input capacitances of Q_6 and Q_7 are approximately 75 pF, ensuring that the narrow pulses will transmit properly.

During the rising edge of the drive-control signal, Q_7 turns on, and its current starts charging the input capacitance of Q_2 through the on-resistance of Q_7. Because Q_7's on-resistance is only a few ohms and no additional drain resistance exists, the charging process of Q_2's input capacitance becomes fast, although its input capacitance is high.

As the gate voltage of Q_2 increases, the gate-to-source voltage of Q_7 decreases, and the transistor turns off. As a result, the narrow pulses that the differentiator circuit generates transmit to transistors Q_7 and Q_2 through coupling transformer T_1 to transistor Q_3, which charges Q_5's gate-to-source input capacitance. The same process occurs with Q_6, Q_1, and Q_4 during the falling edge of the drive-control signal to discharge Q_5's gate-to-source input capacitance.

With potentiometer P_1, you can control the discharge time of Q_1 and Q_2 and thus adjust the offset of the drive signal you apply to the power transistor. Because Q_6/Q_1 and Q_7/Q_2 transmit narrow pulses and have fast rising and falling edges, you can obtain a great duty-cycle variation even for high switching frequencies. You can control the duty cycle from 2 to 98% with a 20-kHz switching frequency. The circuit's compact design lets you mount it close to the power module, which minimizes parasitic elements.

Figure 2 shows the driver prototype for a 10-kW/20-kHz three-phase power inverter for grid injection. The circuit uses SKM75GB128 power transistors from Semikron (www.semikron.com). The transistors have a measured input capacitance higher than 15 nF. In this situation, the total current consumption of the FET pulse driver is lower than 30 mA. EDN

REFERENCE

1 Espí, José M, Rafael García-Gil, and Jaime Castelló, "Isolated FET pulse driver reduces size and power consumption," *EDN*, March 30, 2006, pg 98.

Detect missing pulses to avoid losing data

Chien-Hung Chen and Tai-Shan Liao,
National Applied Research Laboratories, Hsinchu, Taiwan

In an SWIR (short-wave-infrared) indium-gallium-arsenide-based imaging system, the system must detect every trigger pulse; otherwise, lost data will result. This Design Idea solves the problem of a data-conversion system that does not enable an acquisition in the first of 256 trigger pulses. **Figure 1** shows the SWIR sensor, IC_1. The AD-Trig signal, Pin 24, produces one pulse for each of 256 pixels. **Figure 2** shows the AD-Trig pulses, Channel 1 (Trace 1) and Channel 2 (Trace 2), as the two pulse trains. Channel 3 (Trace 3) is the electro-exposure time signal. The signal is active-high when you expose the image sensor to light. The signal goes low before the AD-Trig signals begin.

THIS CIRCUIT SOLVES THE PROBLEM OF A DATA-CONVERSION SYSTEM THAT DOES NOT ENABLE AN ACQUISITION IN THE FIRST OF 256 PULSES.

Oscilloscope Channel 4 (Trace 4) from the microcontroller's I/O pin goes high during the second pulse, but it should go high during the first pulse. Thus, the microcontroller missed the first pulse and the first pixel of the sensor image. A hardware and firmware redesign solves the problem. The AD-Trig signals connect to a microcontroller's PA0 pin, which you can use as an external interrupt input. **Figure 3** shows the program flow, and **Listing 1**, which is available at www.edn.com/100610dia, shows the source code.

In **Figure 3**, the left side is the TD-poll routine in the **listing**. It uses an I/O pin to detect the electro-exposure signal. When the electro-exposure time signal finishes, the firmware enters the external interrupt subprogram when you activate the AD-Trig signal. The interrupt subprogram reads all of 256 pixels data. The pixel data is no longer lost because the time for firmware processing of the TD poll differs from that of the external interrupt subroutines, and the external interrupt has a higher priority than does the TD poll. EDN

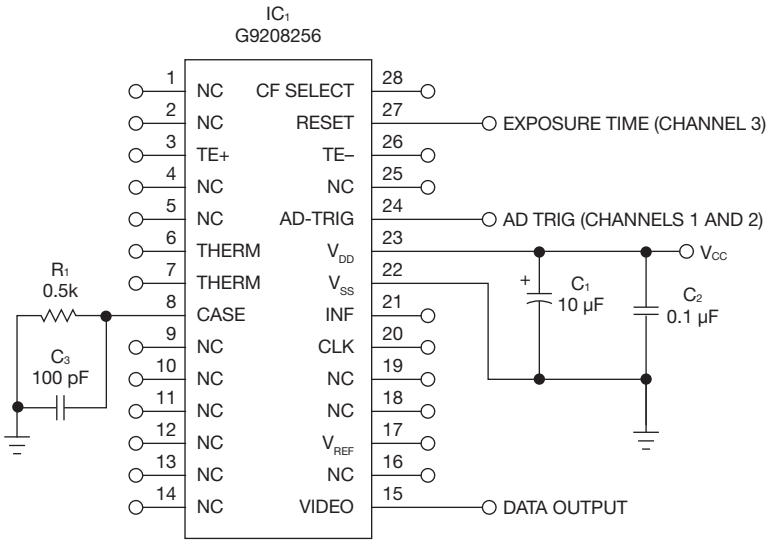


Figure 1 An SWIR sensor's AD-Trig signals produce one pulse per pixel.

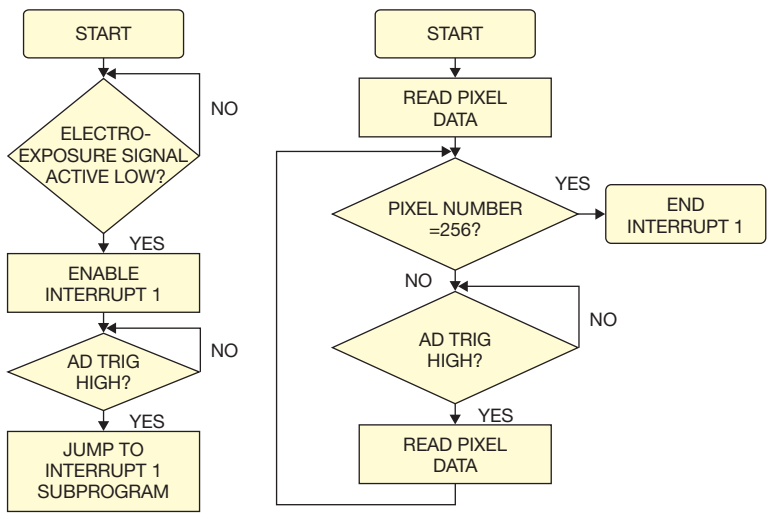


Figure 3 Using an interrupt, the firmware code detects the first pulse, AD-Trig high.

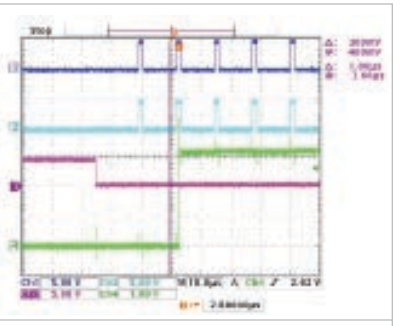


Figure 2 A problem occurs because the microcontroller missed the first pulse on Channel 4.

Circuit lets you measure zener voltages and test LEDs

Vladimir Oleynik, Moscow, Russia

To measure a zener diode's breakdown voltage, you need a dc voltage source whose voltage exceeds that of the zener voltage. In **Figure 1**, resistor R_{SER} provides voltage drop between V_{IN} and V_{ZEN}. In any case, V_{IN} should exceed V_{ZEN}. Resistor R_{SER} must provide current, I_{ZEN}, that can keep the zener diode in reverse breakdown. That is, the current must be more than I_{ZEN}–I_{ZENMIN} and less than I_{ZEN}–I_{ZENMAX}. You also need to consider the current that flows through the load. Otherwise, V_{ZEN} will be unregulated and less than the nominal breakdown voltage. Also, the power that the zener diode dissipates should not exceed the manufacturer's specifications. Except for the value of I_{ZEN}–I_{ZENMIN}, all necessary data appears in zener-diode data sheets.

The circuit in **Figure 2** uses one or two AA/AAA cells, which ensures testing irrespective of the value of the tested zener voltage to approximately 20 to 25V. The heart of the circuit is a Zetex (www.diodes.com) LED driver, ZXLD381. It operates mainly from 1.5 or 1.2V cells, and it has a maximum input voltage of 10V. The LED driver generates constant-power pulses that charge output of the 10-µF capacitor, which requires low leakage current. The capacitor's voltage provides constant current through R_1 and tthe zener diode that connects in series. When you connect output probes to a digital multimeter's V and COM sockets, you can directly measure the zener's voltage when the S_2 switch is in the position **Figure 2** shows.

When S_2 is in the upper position, the meter measures voltage across R_1, a 1-kΩ resistor; the meter displays a negative-voltage-drop value. R_1's value ensures direct reading of the meter; the voltage drop across R_1 corresponds to the zener's current, so there is no need to switch over DVM (digital-voltmeter) ranges. R_1's voltage drop limits the zener diode's voltage value that the circuit measures. If R_1's value is 1Ω, then the voltage drop across it is insignificant and, in millivolts, is equal to the zener diode's current in milliamps.

If you need to measure zener-diode voltage higher than 20 to 25V, you can add an LED driver (**Figure 3**). When the S_2 switch is in the upper position, both LED drivers connect in series, and you can measure zener voltage to approximately 40V at 0.7 mA. With S_2 in the lower position, both LED drivers connect in parallel, and the tested zener voltage is approximately 20 to 25V at several milliamps of current. Both parallel and in-series connections provide zener-diode-voltage measurement at two current values. In some cases, I_{ZEN} values may not fit the **equation** I_{ZEN}–I_{ZENMIN}<I_{ZEN}<I_{ZEN}–I_{ZENMAX}, and the zener diode may go out of regulation.

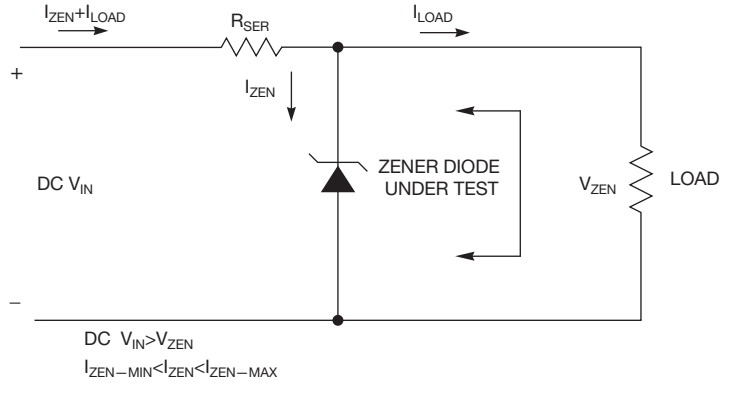


Figure 1 Current through a zener diode creates a fixed voltage across the device.

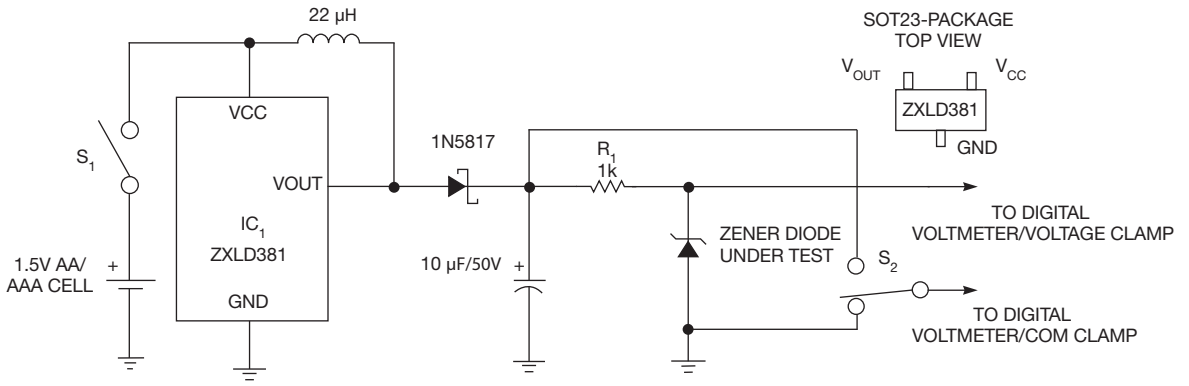


Figure 2 An LED driver provides current for the LED under test.

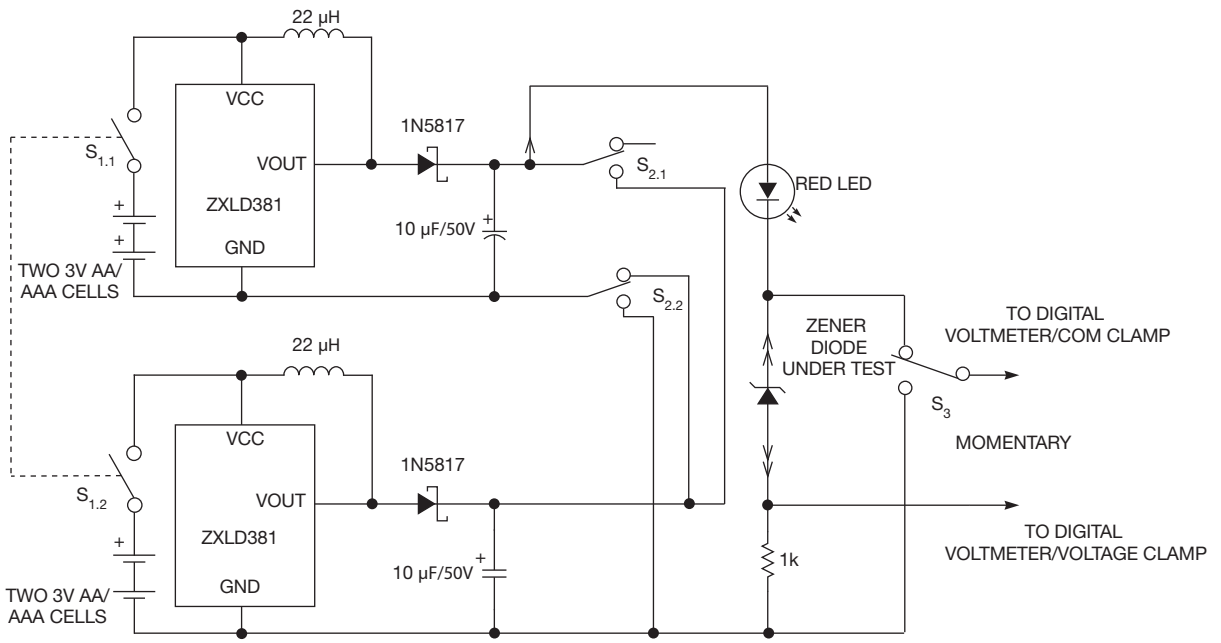


Figure 3 Two LED drivers operating in series or parallel let you boost zener voltage or current.

The red LED provides visual indication of the current flowing through the resistor-zener-diode circuit. The higher the current, the brighter the LED will light. You can omit this LED if you don't need such an indication. The voltage drop from the red LED lowers voltages by about 1.8 to 2V. If you connect the zener diode's conducting current in the forward direction, its voltage drop is equal to that of an ordinary silicon diode, or approximately 0.6 to 0.8V. When you apply forward voltage, Schottky and small-signal germanium diodes exhibit 0.2 to 0.25V and 0.35 to 0.45V drops, respectively.

Figure 4 shows an assembled zener-diode-tester circuit. The zener diode under test is On Semiconductor's (www.onsemi.com) BZX55C15RL, which has a working voltage of 13.8 to 15.6V. The voltage in **Figure 4** was measured at a zener-diode current of 2.8 mA.

You can use the circuits in **figures 2** and **3** to test LEDs, regardless of their color. Place the forward-biased LED you would like to test and set S_2 to the lower position. The voltage drop across the 1-kΩ resistor corresponds to the current in milliamps that flows through the LED under test. The circuit can light up even HB (high-brightness) LEDs because, due to their high efficacy, they start lighting at current values as low as a few milliamps. EDN

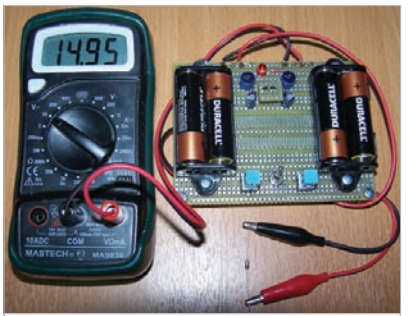


Figure 4 Measure the voltage across a zener diode with a digital multimeter.

Switched-capacitor voltage multiplier achieves 95% efficiency

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

A capacitor that you charge through a resistor operates at 50% efficiency; hence, many engineers avoid using switched-capacitor dc/dc converters. That efficiency figure holds true only for capacitors with no initial voltage, however. If you decide to switch a precharged capacitor, you can transfer energy to an output with a power-efficiency approaching 100%.

To attain a four-thirds multiple of an input supply voltage, you can charge three capacitors to one-third of each supply voltage and then add that one-third of the input voltage to the input voltage to yield an output voltage that is four-thirds of the input voltage. You series-connect three capacitors of equal value and then charge this series string to a voltage equal to the input voltage. Because the values are equal, each of the capacitors charges to one-third the input voltage. The circuit then connects these three capacitors in parallel on top of the input voltage and switches this increased voltage to the output (**Figure 1**). The circuit repeats these two phases of operation at clock frequency F.

C_{IN} and C_{OUT} are filtering capacitors at the input and output, respectively. R_p is a protective resistor, which limits the inrush current to the capacitors at power-on. As the output voltage rises, it closes IC_5 and shorts out this resistor. Schott-

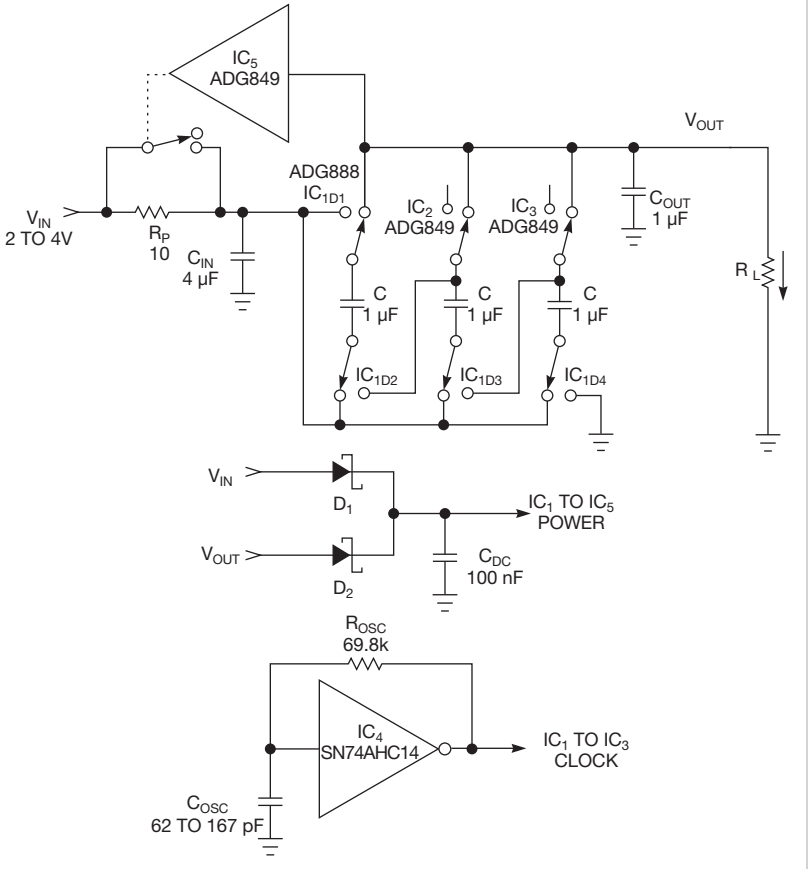


Figure 1 This charge-pump circuit raises the input voltage to a 1.33 multiple. You can use it as a 2.5-to-3.3V converter. The circuit works in environments with high magnetic fields that might interfere with an inductor-based converter.

ky diodes D_1 and D_2 allow you to power the ICs using the input voltage until the output rises, at which time the higher output voltage powers the ICs. C_{DC} is a storage and decoupling capacitor for this power bus. The higher power-supply voltage is necessary for proper operation, and it lowers the on-resistance of the analog switches. The 0.4Ω on-resistance of IC_1 results in low circuit losses and high-efficiency operation. IC_1, IC_2, and IC_3 exhibit a break-before-make operation, which is essential in this case.

For a 50%-duty-cycle clock, you can calculate the theoretical power-efficiency of the converter, according to the following **equation**:

$$\eta \simeq 1-\left(\frac{1}{2}+\frac{\frac{C_{OUT}}{3C}}{\left(1+\frac{C_{OUT}}{3C}\right)^2}\right)\times\frac{1}{12R_LCF}.$$

If the value of C_{OUT} is equal to the value of C, the power loss due to charging of the three capacitors is about two-thirds of the power loss during the discharging phase. The power consumption of the control circuit reduces the efficiency of this calculated value. For CMOS circuits, the power consumption rises linearly with the operating frequency. By choosing the operating frequency, you can optimize the efficiency of the circuit. The optimum frequency is inversely proportional to the load resistance, R_L. Fortunately, the efficiency maximum is flat, so you can achieve efficiencies higher than 90% over a wide range of values for R_L. You can attain 94% efficiency driving a 120Ω load over clock frequencies of 100 to 400 kHz. If you set a 229-kHz operating frequency, an input of 2.2V yields a 2.87V output at an efficiency of 95.9%. The optimum clock frequency shifts to lower values at lighter loads. You can drive a 269Ω load at 100 kHz and achieve an output of 2.88V. EDN

Bootstrap circuit speeds solenoid actuation

By Ralf Kelz, Seefeld, Germany

The circuit in this Design Idea bootstraps a large capacitor in series with the solenoid to provide a large actuation voltage (**Figure 1**). This higher voltage provides substantially more current to operate the solenoid (**Figure 2**, which is available at www.edn.com/100610dib), speeding the operation of the solenoid. You can also choose operating voltages or solenoid specifications that result in lower continuous current through the solenoid, reducing dc power consumption and resulting in a cooler-running solenoid with better reliability.

When there is a 0V input to the circuit, both transistors are off. Resistor R_1 slowly charges the left side of capacitor C_1 to the 24V power-supply voltage. D_2 clamps the right side of capacitor C_1 to 0.6V. When the input signal goes high, both the Q_1 and the Q_2 transistors turn on. This action quickly drives the left side of C_1 to ground. Because voltage cannot change instantaneously across a capacitor, the right side of C_1 goes down to –23.4V. D_2 steers the solenoid current into the capacitor until it discharges, at which time the solenoid current conducts through D_2 to ground. D_1 prevents a voltage-overshoot spike when the circuit turns off, and current suddenly stops flowing in D_1. It clamps the bottom leg of the solenoid to 24.6V until the current decays in the solenoid.

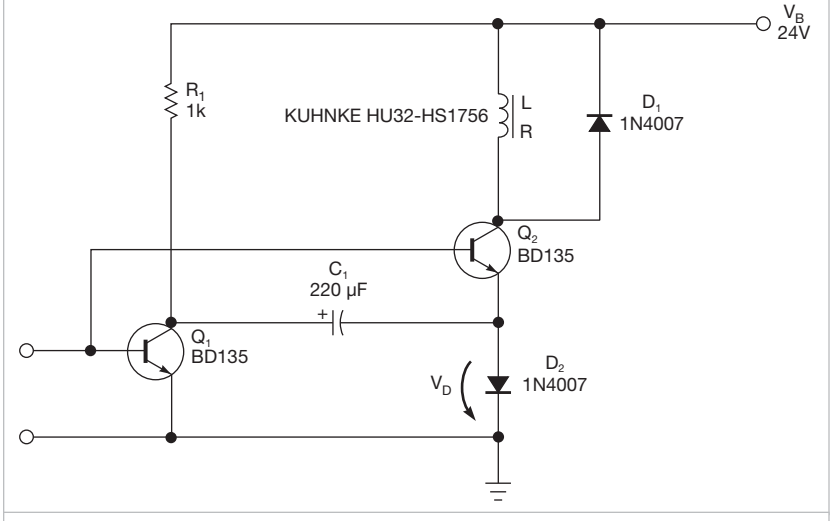


Figure 1 This circuit bootstraps the power supply voltage across the solenoid to temporarily double the actuation voltage.

THE TIME CONSTANT DEPENDS ON THE SOLENOID'S INDUCTANCE AND THE CAPACITOR'S VALUE.

The time constant of the circuit depends on the inductance of the solenoid and the value you choose for the capacitor, which you can calculate with the following **equations**:

$$I(t)=\frac{(2V_{IN}-V_D)\,e^{\left(-\frac{t}{\tau}\right)}\sinh(\omega t)}{\omega L};$$

$$\omega=\sqrt{\left(\frac{R}{2L}\right)^2-\frac{1}{LC}}\ ;\ \text{and}\ \tau=\frac{2L}{R}.$$

In these **equations**, e is the mathematical constant, ω is the radian angular frequency, and t is time in seconds. In addition, L is inductance and R is resistance. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Notch filter autotunes for audio applications

John R Ambrose, Mixed Signal Integration, San Jose, CA

Tracking notch filters find use in harmonic-distortion analyzers; they also can remove heterodyne noise from ham-radio systems. A conventional tracking switched-capacitor notch filter relies on a bandpass filter, a voltage-to-frequency converter, and a notch filter to track the incoming signal and remove undesired tones. The bandpass filter in these circuits sometimes adjusts to the wrong frequency, meaning that the undesired tone would have no attenuation.

The circuit in this Design Idea uses IC_1, a 74HC4046 PLL (phase-locked-loop) IC, which operates as fast as 1 MHz, to improve the noise immunity of the system (**Figure 1**). IC_2, an RDD104 IC from LSI Computer Systems Inc (www.lsicsi.com), provides a 1000-to-1 divider in an eight-pin package. IC_3, Mixed Signal Integration's (www.mix-sig.com) MSHN5 1000-to-1 clock-to-corner switched-capacitor highpass/notch filter, comes in an eight-pin package.

You feed IC_1's VCO (voltage-controlled oscillator) output into the clock input of IC_2. IC_2 can perform 10-, 100-, 1000-, and 10,000-to-1 divisions using the DIV1 and DIV2 pins. You tie the output of RDD104 to the COIN of IC_1. By using IC_1's EX/OR phase comparator, you can improve noise immunity. You apply the input signal to both IC_1 and the input of IC_3, whose clock you derive from the CLKOUT pin of IC_2.

The MSHN5, IC_3, contains both selectable highpass filters and selectable

DIs Inside

▸To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

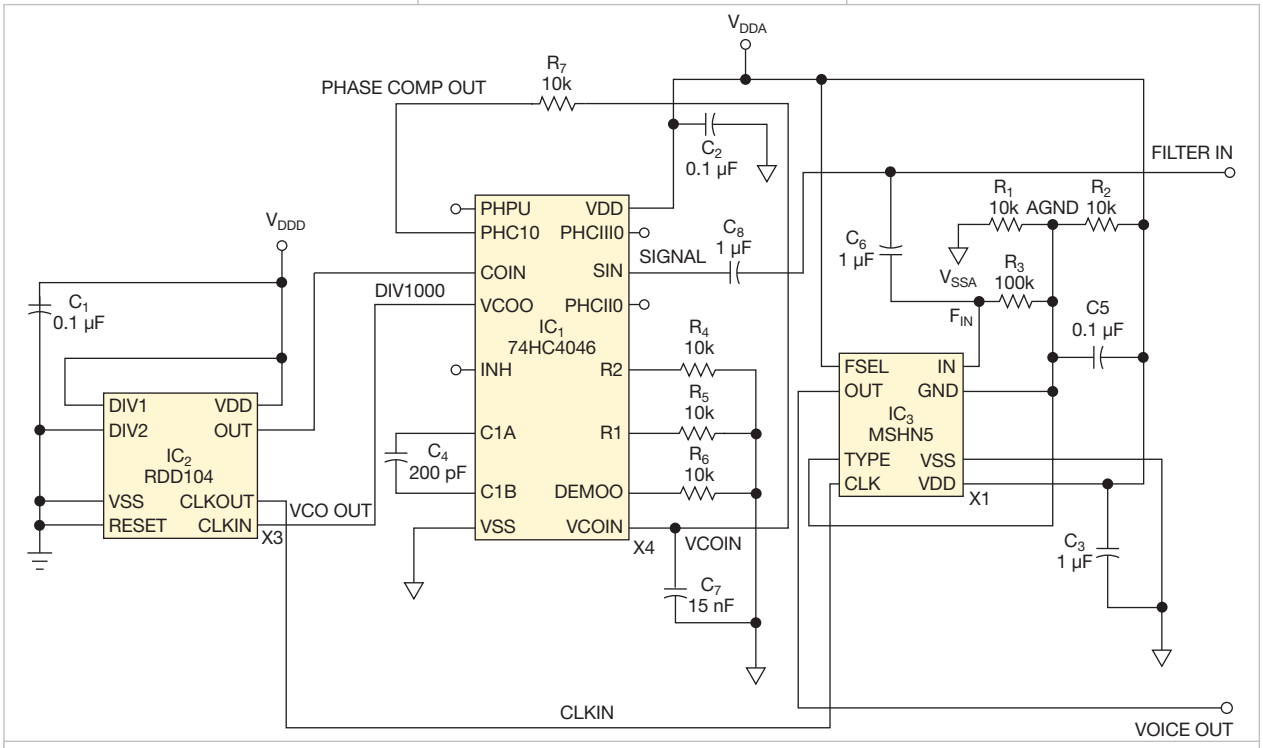


Figure 1 This audio notch-filter circuit uses a PLL to improve noise immunity.

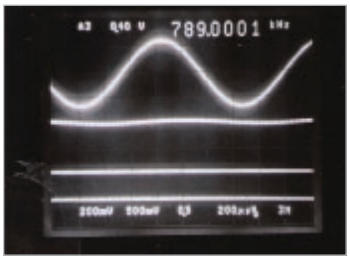


Figure 2 You feed the circuit an input tone (top trace) and get the output signal from the MSHN5 (second trace). The third and fourth traces represent the clock signal from the 74HC4046 PLL.

notch filters. When you tie the FSEL pin high, it selects notch; tying TYPE to AGND selects the narrow notch filter. This step ensures the removal of only one tone from the input signal with little information loss. IC_3's 1000-to-1 clock-to-corner ratio reduces the chance that aliasing signals will affect the output. For voice applications, for example, no signals of 500 kHz or higher would be available to alias into the passband. A sample setup uses an input frequency of 789.13 Hz at a clock frequency of 789.13 kHz, 1000 times the input signal (**Figure 2**). The PLL tracks the input, moving the notch filter to 1.24 kHz. EDN

Image-capture system uses USB and LabView

Chien-Hung Chen and Po-Jui Chen,
National Applied Research Laboratories, Hsinchu, Taiwan

Capturing and processing graphical images requires manipulating data into a form that you can use. This Design Idea describes an imaging system using a USB (Universal Serial Bus) image-capturing system that uses OmniVision's (www.ovt.com) 640×480-pixel, 8-bit-color OV7660 image sensor. The CY7C68013A-128AXC from Cypress Semiconductor (www.cypress.com) provides a USB interface between a PC and the image sensor (**Figure 1**). The control software is written in LabView from National Instruments (www.ni.com/labview).

To avoid losing data from the image sensor, the system employs a data buffer in the image-processing algorithm. The buffer uses system memory for data storage. The queue ensures that the system will not lose data regardless of how much time it takes to process each row in an image. This technique is useful in measurement systems in which the speed of data acquisition and data processing may differ.

Figure 2 shows the programming flow chart. After the system starts, you must set the driver to NI-VISA (Virtual Instrument Software Architecture), a software layer that provides a common programming interface across many types of measurement instruments and software drivers. Once you set the driver, you can initiate the USB device. LabView provides a driver wizard that helps you to build drivers. The LabView code for this graphic-system design can easily implant USB data transmission and its applications. You can download the LabView code from the online version of this Design Idea at www.edn.com/100624dia.

After initializing the USB device, the software allocates system memory in a FIFO (first-in/first out) configuration to become the data buffer. A memory endpoint sets the input buffer's size to 4 kbytes. The software then reads the image in rows and stores data from the sensor in the buffer memory. After reading the data from the buffer, the system image uses two threads to process the data.

Figure 3, available with the online version of this Design Idea at www.edn.com/100624dia, shows the LabView programming diagram for USB data transmission. The program includes for-loop procedures for storing the image in the buffer memory, reading and processing image data, and performing state checking.

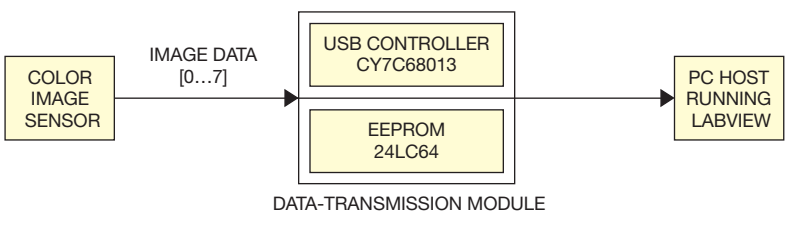


Figure 1 A USB bus controller lets a PC control an image sensor.

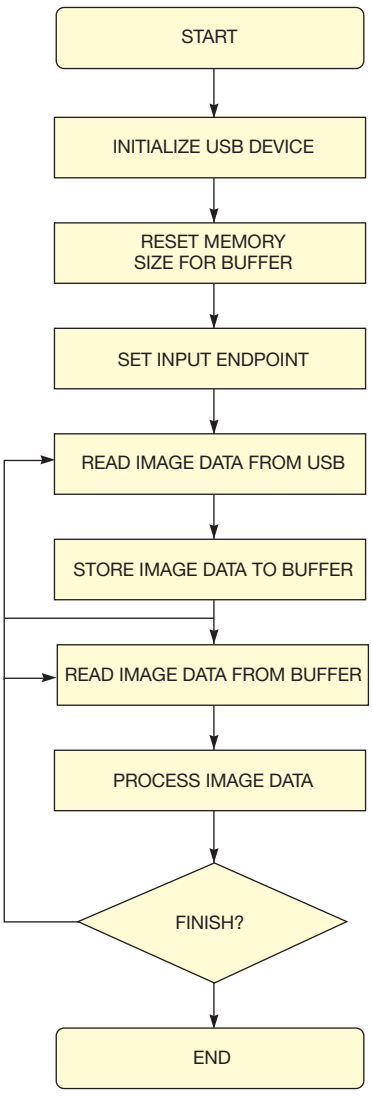


Figure 2 This algorithm lets a host PC capture data from the image sensor.

The main processing algorithm obtains and displays red, green, and blue data of each pixel. **Figure 4**, available with the online version of this Design Idea at www.edn.com/100624dia, shows the test result. The element in the buffer shows that the system processed 614,400 pixels. The actual amount will vary based on the PC's performance. A powerful PC can smoothly run this program, whereas a weak PC will cause the data to accumulate in the buffer. EDN

Low-cost RF synthesizer uses generic ICs

James B Wood, Inovonics Inc, Felton, CA

You can design a hardware-based frequency synthesizer with one inexpensive IC and a few passive components. Such synthesizer chips are not always available, however, because they are typically single-sourced and are not in stock with parts distributors. The need for a working circuit in a short time and using common parts prompted the creation of the circuit in this Design Idea. The synthesizer covers the US commercial AM (amplitude-modulation) broadcast band. It tunes in 10-kHz steps from 500 to 1800 kHz, but you can scale the frequencies for other applications.

The PLL (phase-locked loop) time base is a 100-kHz, tuning-fork-cut crystal of the same size as those in wrist watches. Using a more common crystal requires some extra parts to scale the frequency. Note that if you attempt to use one of these tiny crystals with a CMOS-gate oscillator circuit, however, the circuit will either fail to start or exhibit visible jitter. A discrete-transistor Franklin oscillator, such as the one comprising Q_3 and Q_4 works better (**Figure 1**). This circuit also works well in the VCO (voltage-controlled-oscillator) portion of the synthesizer.

You use IC_{4A}, one-half of a 74HC390 dual decade-divider IC, to divide the 100-kHz reference into the 10-kHz frequency that the PLL uses. This 10-kHz square wave feeds one input of the phase comparator, IC_3, and drives a voltage-tripler circuit comprising D_{12} through D_{15}. This tripler creates approximately 12V and obviates the need for a second higher-voltage power rail. You need the 12V to bias the VCO's varactor diode to the top of its tuning range.

The VCO, comprising Q_1 and Q_2, runs at twice the desired output frequency. Varactor diode D_1 and inductor L_1 provide a tunable tank circuit. Any varactor for AM-radio tuning should

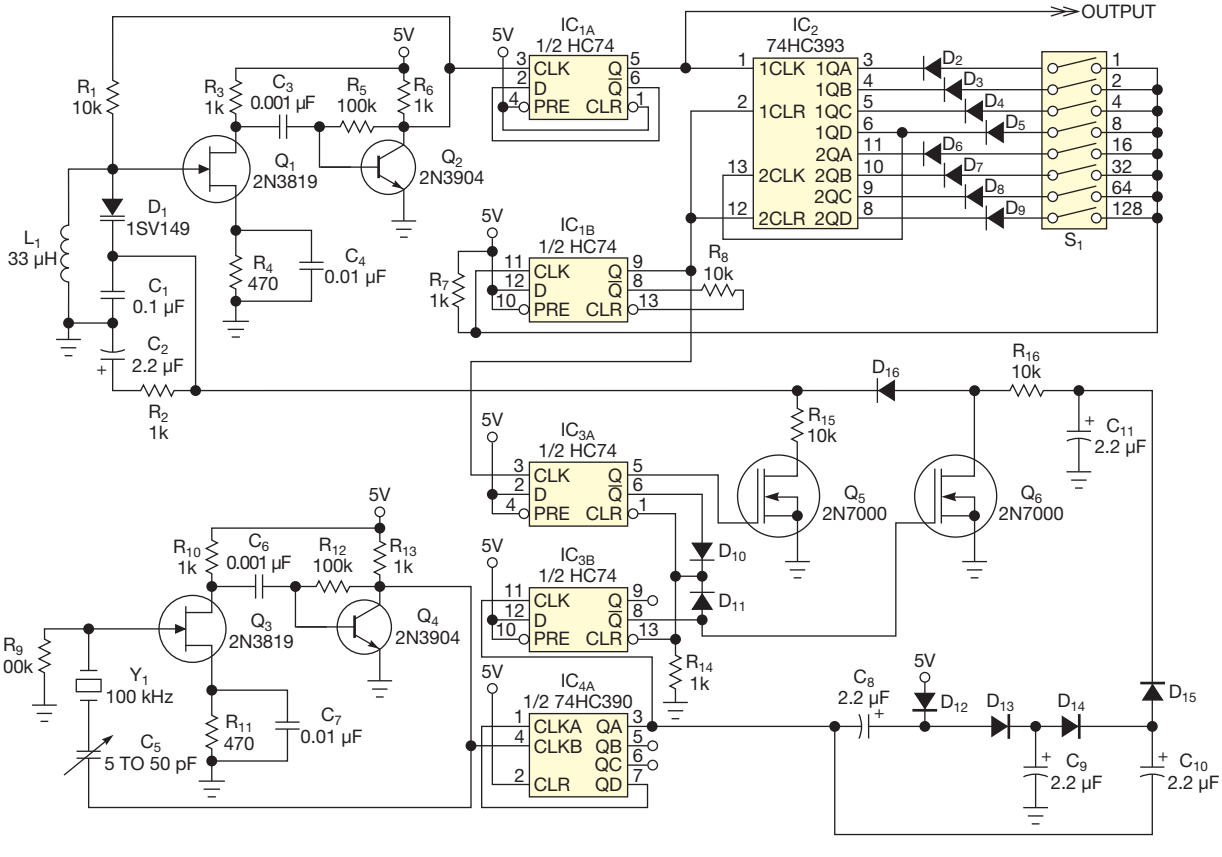


Figure 1 This AM-radio-band frequency synthesizer uses common off-the-shelf parts.

work. The capacitance of these diodes varies from 500 pF with no dc bias to 25 pF with a 12V reverse bias. IC_{1A} divides the LC oscillator by two to yield a symmetrical output waveform.

IC_2 further divides the VCO to the PLL's frequency. IC_2, an eight-stage binary counter, resets itself to zero when it reaches the desired count. IC_{1B}, a pulse-stretching one-shot, ensures that all sections of IC_2 reset at the target count. You program the divider with DIP switch S_1. Diodes D_2 through D_9 supply the necessary AND-logic function.

To set the synthesizer frequency, you first calculate the required divisor. For a 1140-kHz output, you must divide the VCO by 114 to equal the PLL's frequency of 10 kHz. You can close the DIP switches in S_1—in this case, switches 64, 32, 16, and 2—so that the numbers add up to the divisor: 114.

The PLL comparator is a three-state phase and frequency detector (**Reference 1**, in the online version of this Design Idea at www.edn.com/100624dib). When the divided VCO frequency is greater than 10 kHz, the $\overline{Q}$ output of IC_{3B} goes high and the Q output of IC_{3A} pulses at a 10-kHz rate. This action turns on Q_6, back-biasing D_{16} to create a high-impedance state with respect to the 12V supply. Loop-filter capacitor C_2 then discharges through R_{15} and Q_5. When the divided VCO is lower than the loop frequency, the Q output of IC_{3A} goes low, turning off Q_5 and creating a high-impedance state with respect to ground. Q_6 now pulses on and off, allowing C_2 to charge through D_{15} and R_{16}. At PLL lock, Q_5 is off and Q_6 is on, except for a narrow "keep-alive" pulse at the loop frequency.EDN

Tricolor LEDs create a flashing array

Jeff Tregre, www.BuildingUltimateModels.com, Dallas, TX

You can build a matrix of RGB (red/green/blue) LEDs using a simple and inexpensive circuit comprising the control logic and driver circuit in **Figure 1** and some LEDs (**Figure 2**). The center RGB LED is the first to come on, after which each sequential LED in the 8×8-LED matrix follows. This process gives the appearance that the display is alive and moving outward. This sequence repeats, producing a rainbow effect of colors.

You can adjust the frequency of each clock by changing the values of R_{17}, R_{19}, and R_{23}. Use different frequencies for each clock, which will display eight colors from the 65 tricolored LEDs, because using the same frequencies for all the clocks causes your display to appear white. The cost of building this circuit

should be $25 to $30. You can purchase 100 5-mm RGB LEDs from eBay for a total of about $18. Be sure to use common-cathode LEDs.

This simple circuit comprises three clocks and three counters, one for each of the three LED colors. Setting each clock frequency to a different rate causes each color of each LED to appear to be random. All resistors are 0.25W, except for R_3, R_8, and R_{13}, which are 0.5W; R_4, R_9, and R_{14}, which are 1W; and R_5, R_{10}, and R_{15}, which are 1.5W resistors. These high-wattage resistors and the 12 NPN transistors are necessary because all LEDs in this matrix, except the center one, connect in parallel. Start by bending all of the ground leads flat and connecting

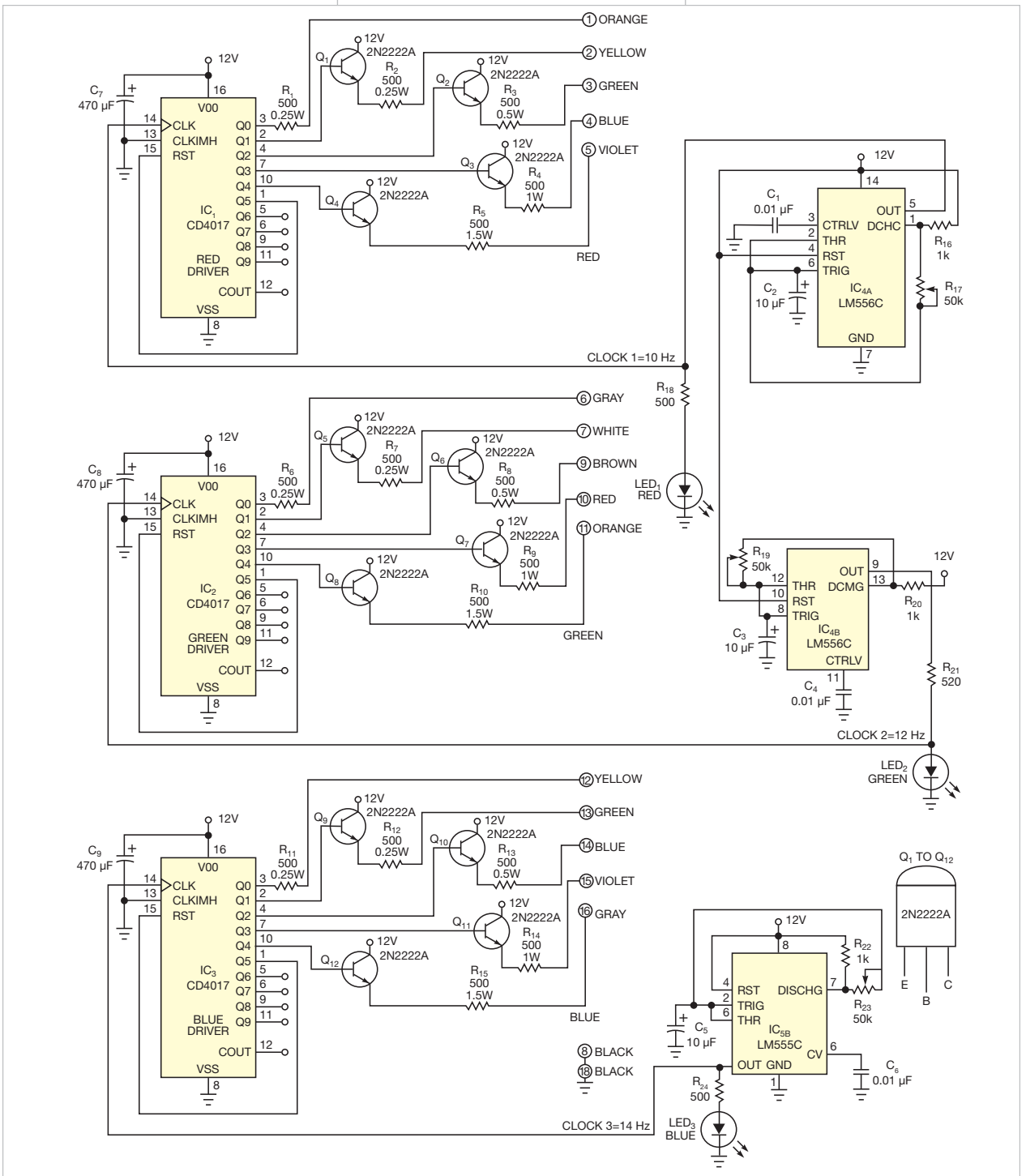


Figure 1 Three 555 timers generate clock signals, and CD4017 counters provide the drive signals for the transistors.

them together. When wiring the LEDs, begin in the center and work outward. You can then mount the LED board onto the top of the PCB (printed-circuit board). See the online version of this Design Idea at www.edn.com/100624dic for photos, a parts list, and a video of this circuit in action.

To add the finishing touches to your project, use a small picture frame and install waxed paper onto the inside of the glass. Mount the LED board ¼ to 1 in. away. The magnifying lens of the LEDs will produce a beautiful effect when they shine through the waxed paper. EDN

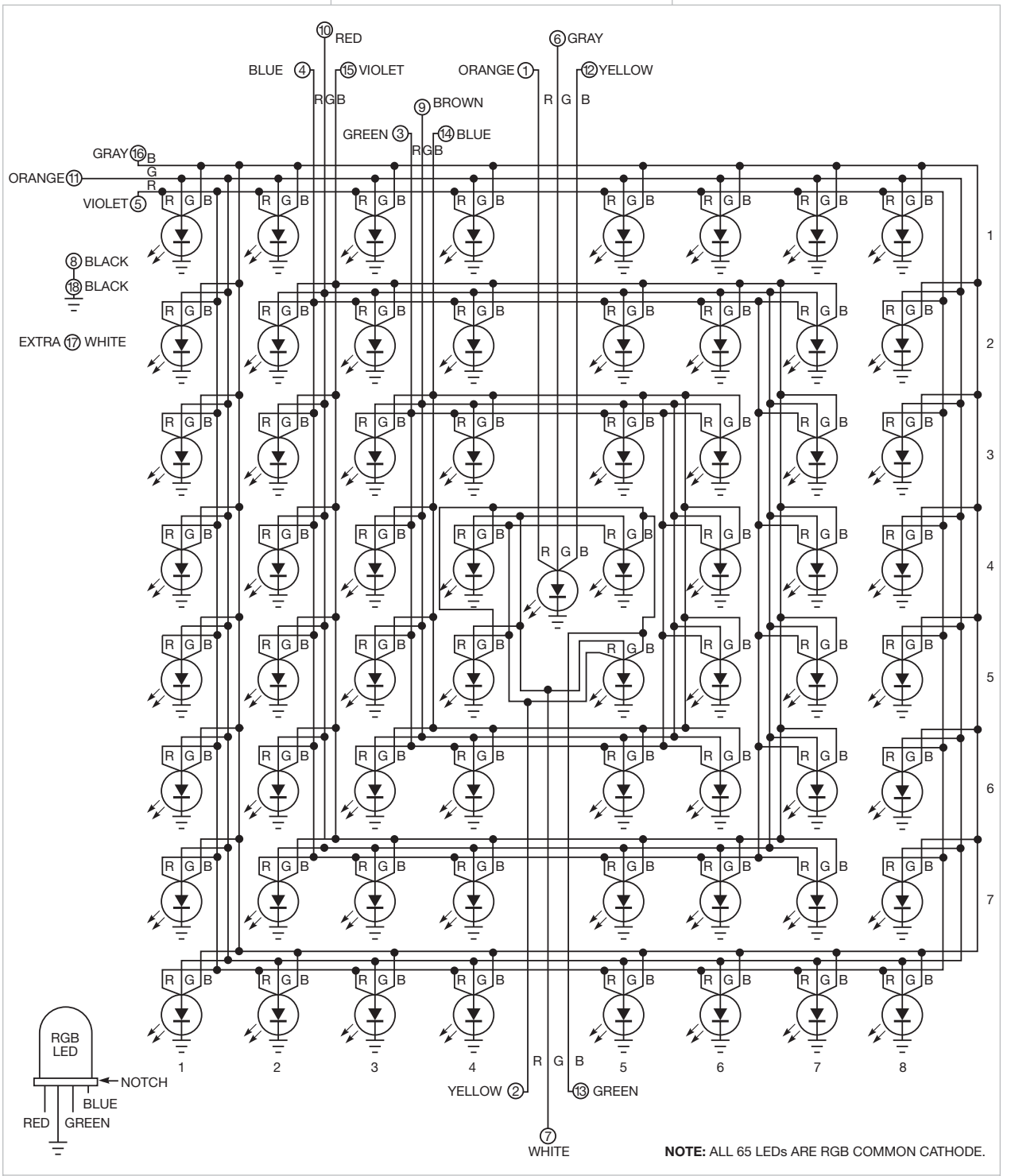


Figure 2 The LED in the center lights first, and the light then moves outward until the circuit products an 8×8-LED display.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas
READERS SOLVE DESIGN PROBLEMS

Pulse generator with precision output-duty cycle operates at a repetition rate beyond 50 MHz

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

A previous Design Idea describes an astable multivibrator that gets hysteresis from the positive-feedback stage using a technique you can characterize as positive-capacitive feedback (**Reference 1**). It creates hysteresis by a charge transfer to its main timing-capacitor. The circuit uses two logic inverters to generate complementary logic outputs. In contrast, the circuit in **Figure 1** uses a single fast analog comparator that has complementary outputs, thus eliminating the need for an additional inverter.

Comparators such as Analog Devices' (www.analog.com) ADCMP603, IC_1, have symmetry that results in a very low time skew of the transitions at its Q and $\overline{Q}$ outputs that are fractions of a nanosecond. Thus, the charge transfer to the main capacitor, C, theoretically starts immediately at the start of the level transition at the Q output, from which C is charged through a resistor, R. No additional propagation delay occurs at any stage besides Q, resulting in a further increase in operating frequency.

The output frequency of the pulse generator in **Figure 1** is less sensitive to supply voltage variations than a generator with the ADCMP603, which uses the IC's internal hysteresis. The charging current of C and the charge-transfer-based hysteresis of the pulse generator rise almost linearly with the rising supply voltage, which results in an output signal's insensitivity.

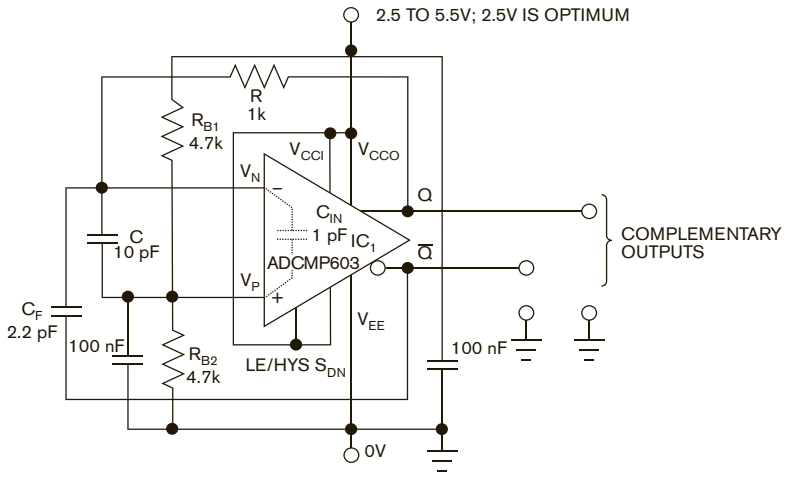


Figure 1 This pulse generator produces time-aligned complementary logic waveforms. The duty cycle holds at 50%.

Contrarily, the internal hysteresis, which an external resistor sets at the comparator's LE/HYS pin, varies nonlinearly with the supply voltage. If, for example, the external resistor's value is 225 kΩ, the hysteresis has the same value of about 36 mV for supply voltages of 2.5 and 5.5V.

At frequencies close to the upper frequency limit of operation, the charge injection through capacitor C_F is gradual rather than steplike because the rise and fall times of waveforms at the Q and $\overline{Q}$ outputs are still of finite value. **Figure 2** shows an idealized operation for the positive peak of voltage at capacitor C. Although the real voltages, V_Q and $V_{\overline{Q}}$, have somewhat rounder "corners," they resemble those in the **figure**.

If you assume that the voltage swing at capacitor C is considerably lower than that at Q and $\overline{Q}$, then a rectangular current pulse, $I_{CF}(t)$, charges capacitor C during a positive-voltage transition at output $\overline{Q}$. The current through resistor R, $I_{RF}(t)$, which also flows to capacitor C, changes its polarity at the midtransition at Q and $\overline{Q}$ outputs. The final current, which charges capacitor

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

C, is a sum of $I_{RF}(t)$ and $I_{CF}(t)$. Although the waveshape of the voltage at capacitor C depends on the final charging current, it gains in peak solely due to $I_{CF}(t)$.

The following **equation** calculates the added peak:

$$\Delta V_C \approx \frac{I_{CF} \times t_R}{C_1} \approx \frac{C_F}{C_1} \Delta V_{OUT},$$

where ΔV_C is the added peak, $C_1 = C + C_{IN}$, and $C_1 \approx 1$ pF. The added peak is independent of $I_{RF}(t)$ due to the zero mean value of this current within the level transition at the Q and $\overline{Q}$ outputs.

For C=10 pF, C_1=11 pF, C_F=2.2 pF, and $\Delta V_{OUT} \approx 2.4$V, a voltage you derive from the **equation** $\Delta V_C \approx 0.48$V.

The following **equation** calculates the nearly constant part of $I_{RF}(t)$ current:

$$I_{RF} \approx \frac{1}{2} \times \frac{\Delta V_{OUT}}{R}.$$

I_{RF} determines the slope, S, of the $V_C(t)$ waveform in **Figure 2**, which excludes the time interval of level transitions at the comparator's Q and $\overline{Q}$ outputs. You calculate the slope with the following **equation**:

$$S = \frac{I_{RF}}{C} \approx \frac{1}{2RC} \times \Delta V_{OUT}.$$

The following **equation** determines the absolute value of peak voltage of $V_C(t)$, referred to supply midvoltage:

$$V_{CPEAK} = S\left(t_{PD} - \frac{t_R}{2}\right) + \Delta V_C.$$

Voltage $V_C(t)$ decreases from its peak value with a slope of −S. You calculate the time interval, T_{DESC}, when it reaches the reference level as:

$$T_{DESC} = \frac{V_{CPEAK}}{S} = t_{PD} - \frac{t_R}{2} + \frac{\Delta V_C}{S} = t_{PD} - \frac{t_R}{2} + 2C_F R.$$

By evaluating this **equation** for a t_{PD} of approximately 3.5 nsec and R with a value of 1 kΩ, the time interval is approximately 6.8 nsec.

The following **equation** calculates the total time, when $V_C(t)$ is higher than the reference voltage, $V_{CC}/2$:

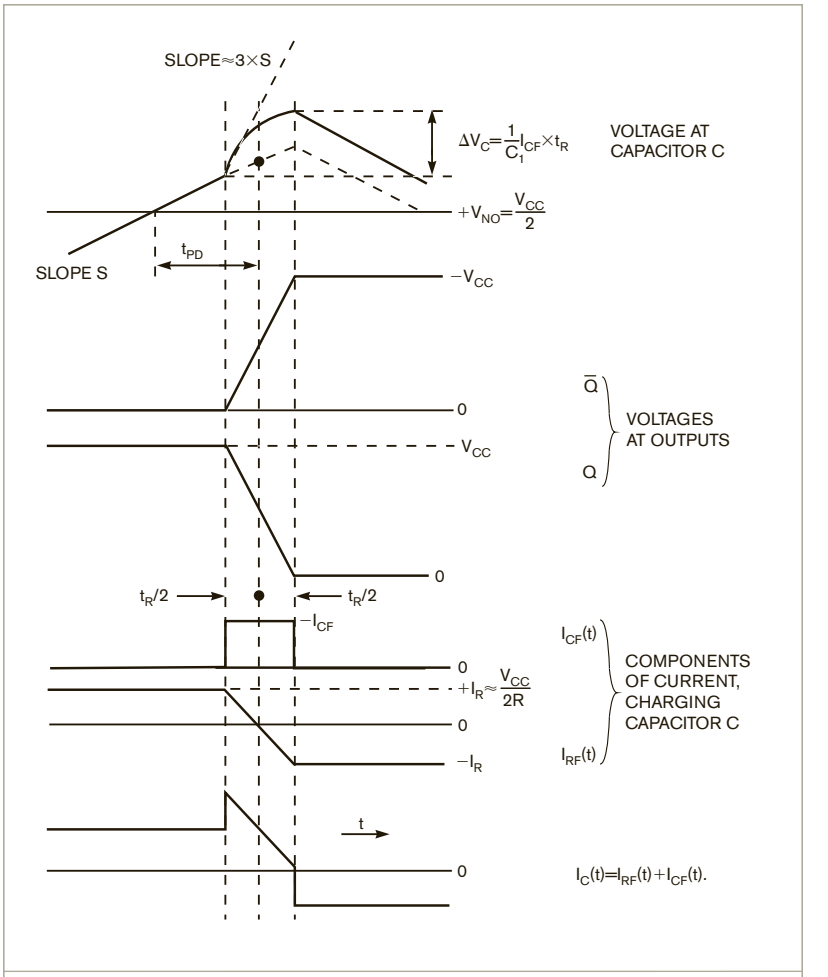


Figure 2 An idealized but still close-to-reality modeling of the operation of the generator gives you a formula for determining the output frequency.

$$T_H = t_{PD} + \frac{t_R}{2} + T_{DESC} = 2(t_{PD} + C_F R).$$

The symmetry of the ADCMP603's internal circuit architecture, T_H, is the right half-period of logic waveforms at the Q and $\overline{Q}$ outputs. In other words, the duty cycle of the output pulse is 50%. By evaluating the **equation** for T_H, you get 11.4 nsec. Thus,

$$f_Q = \frac{1}{2T_H} = 43.86 \text{ MHz}.$$

The circuit's output frequency is 56.75 MHz with a power-supply voltage of 2.052V. With a supply voltage of 3.51V, the frequency changes to 56.12 MHz. Thus, the relative sensitivity of the output frequency to the supply-voltage variation is approximately 8×10^{-3}/V. You can attribute an increase of experimental frequency as compared with a theoretical value to the fact that, during the estimated signal-propagation delay, t_{PD}, the comparator's input overdrive rises gradually to about 330 mV, which is more than triple the value at which you define the propagation delay. You can therefore assume a lower propagation delay and a higher frequency. EDN

REFERENCE

Larson, Robert, "Astable multivibrator gets hysteresis from positive-feedback stage," *EDN*, Oct. 22, 2009, pg 43, www.edn.com/article/CA6702271.

Tables ease microcontroller programming

Abel Raynus, Armatron International, Malden, MA

When creating microcontroller firmware, you often need to work with data arrays. Tables make easy work of data arrays, such as those for digital-code transformation, correction for sensor linearity, sophisticated calculations, and multiple output organization. **Table 1** shows how you can organize data in a table. Outputs A, B, and C have values based on the input value, V.

When using a lookup table, choose the proper microcontroller input and outputs. Assign values for input and outputs data in **Table 2**. These data can consist of constants in binary, hexadecimal, or decimal format or names. For names, you should assign a constant value to each one. For example:

```
data1  equ  $0a
data2  equ  $0b
data3  equ  $0c
data3  equ  $0d
```

Next, put the data from **Table 2** in either the beginning or the end of ROM, which makes the data easy to find. For definition of 1-byte data storage, use pseudo operators FCB or DB. For storage of data comprising 2 bytes, use FDB or DW, as in the following example:

```
ORG ROM
Vx  FCB  0T,2T,4T,6T
Ax  FCB  data1,data2,data3,data4
Bx  FCB  $aa,$bb,$cc,$dd
Cx  FDB  $1122,$3344,$5566,$7788
```

Note that commas separate the data. Don't place a comma after the last data, or it will be considered as $00.

When working with tables, you should always use indexed addressing mode. It provides access to data using variable addresses. Most microcontrollers have two index registers, X and H. Register X contains the low byte of the conditional address of the operand; H contains the high byte. The algorithm of working with tables is straightforward. After you detect the input value, you should then compare it with the table's input data. The X index determines this value, starting with X=0 and ending with X=N. In this example, N=4. When you find table data equal to the input value, you use the corresponding X as an index to load the output registers with their values. In the case of 2-byte numbers, you should load the output registers separately, first with a high byte and then with a low one. **Figure 1** illustrates this process.

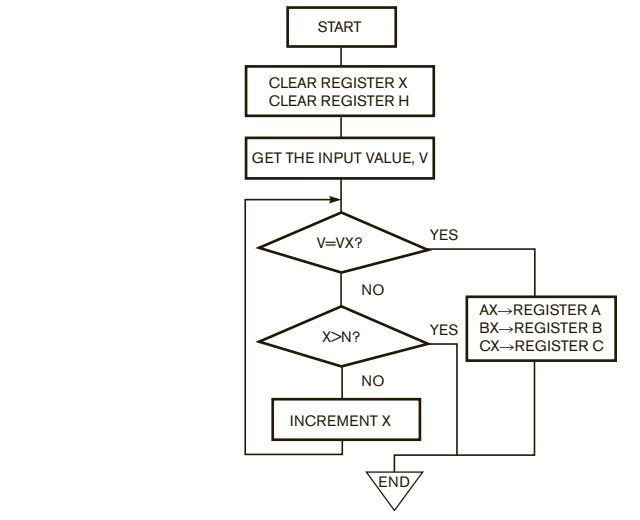


Figure 1 You can use a look-up table in microcontroller code.

TABLES MAKE EASY WORK OF DATA ARRAYS, SUCH AS THOSE FOR DIGITAL-CODE TRANSFORMATION AND SOPHISTICATED CALCULATIONS.

The **listing** of assembler code is available from the online version of this Design Idea at www.edn.com/article/100422dib. In the **listing**, you can double-check the table content in memory at addresses $F800 through $F813. The listing uses Freescale (www.freescale.com) assembler because most of the appropriate applications employ inexpensive, 8-bit microcontrollers from Freescale's HC08 Nitron family. You can, however, use this approach with any type of microcontroller and assembly language. EDN

TABLE 1 OUTPUT VALUES VERSUS INPUT VALUES

Input V	Output A	Output B	Output C
V1	A1	B1	C1
V2	A2	B2	C2
....	....	....	....
VN	AN	BN	CN

TABLE 2 INPUT AND OUTPUT VALUES

Input V	Output A	Output B	Output C
V1=0T	data1	$aa	$1122
V2=2T	data2	$bb	$3344
V3=4T	data3	$cc	$5566
V4=6T	data4	$dd	$7788

Bicolor LED driver uses two leads

Mario Marcoccia, United Circuits, Fort Lauderdale, FL

You can use the circuit in **Figure 1** to drive a bicolor LED with only two leads. This circuit detects the correct closed condition of the left and right side bags in a motorcycle companion. The bag has two locks that you must close for protection. When you push the two momentary SPDT (single-pole/double-throw) switches, they sense the correct closed bag. One bicolor red-and-green LED indicates the bag's status, with the red color showing the open-bag condition. To illuminate the LEDs, you must reverse the polarity of the applied voltage to the LED to change the color (**Table 1**).

Diodes D_1 and D_2 and resistor R_1 form a discrete OR gate. When either pushbutton switch connects to 12V, the voltage at Point A is positive with respect to Point B. Transistor Q_1 conducts, letting current illuminate the red LED. When neither switch connects to 12V, neither diode conducts. The base of Q_1 pulls low through R_1 and R_4, indicating that the bags are closed. Thus, the green LED illuminates as current passes through it and through R_1 and R_2. The diode, transistor, and resistor values are not critical, and you can adjust them according to your needs. You can also replace the bicolor LED with two discrete LEDs of different colors placed back to back. EDN

TABLE 1 LOGIC CONDITIONS FOR THE CIRCUIT

	Switches		Bicolor LED	
Bag condition	**S_1**	**S_2**	**Red**	**Green**
Open	1	1	On	Off
Open	0	1	On	Off
Open	1	1	On	Off
Closed	0	0	Off	On

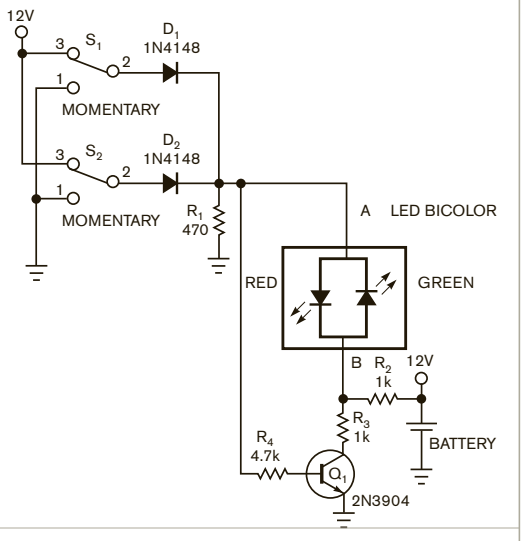


Figure 1 This circuit detects the correct closed condition of the left and right side bags in a motorcycle companion.

Single IC forms precision triangular-wave generator

Akshay Bhat, Maxim Integrated Products Inc, Sunnyvale, CA

The linearity of triangular waveforms makes the triangular-wave generator useful in sweep circuits and test equipment. For example, switched-mode power supplies and induction motor-control circuits often include a triangular-wave oscillator as part of their PWM (pulse-width-modulation) circuit.

The basic triangular-wave generator includes an integrator for generating the triangular-wave output and a comparator with external hysteresis, such as a Schmitt trigger, for setting the output amplitude (**Figure 1**). You can implement these components with a Maxim (www.maxim-ic.com) MAX9000 IC, which includes a high-speed operational amplifier, a 185-nsec comparator, and a precision 1.23V bandgap reference.

The integration of a constant current, which you obtain by applying constant voltage across a resistor, produces a linear ramp at the op amp's output. This output feeds a Schmitt trigger whose output feeds back to the integrator resistor. Abrupt state changes in the Schmitt trigger's output determine the peak voltages for the triangular-wave output. These changes in turn depend on the input threshold voltages you set for the Schmitt trigger.

Unfortunately for this circuit, the triangular-wave peaks must be symmetrical about the reference voltage you apply to the comparator's inverting input. To generate a triangular wave from 0.5 to 4.5V, for example, you must provide a reference voltage of (0.5V+4.5V)/2=2.5V.

It would be preferable to set this voltage range independently of the standard bandgap-reference voltage available, 1.23V. You can achieve this flexibility by adding resistor R_3 to the hysteresis network in a single-IC version of the circuit (**Figure 2**). R_3 lets you set the triangular-wave peaks independently of the reference voltage.

To build the Schmitt-trigger comparator, you first select R_2. The comparator's input-bias current at C_{IN+} is less than 80 nA. To minimize the error this current causes, the current through R_2, $[(V_{REF}-V_{OUT})/R_2]$, should be at least 8 µA. R_2 requires two **equations**, corresponding to the two possible comparator-output states: $R_2=V_{REF}/I_{R2}$, and $R_2=(V_{DD}-V_{REF})/I_{R2}$.

Use the smaller of the two resulting resistor values. For example, if the supply voltage is 5V, the reference voltage is 1.23V, and the reference current is 8 µA, the two R_2 values are 471.25 and 153.75 kΩ, so this circuit uses the standard value of 154 kΩ.

Next, select R_1 and R_3. During a rising ramp, the comparator output is logic low (V_{SS}). Similarly, the comparator output is at logic high (V_{DD}) during a falling ramp. Thus, the comparator must change state according to the required peak and valley points of the triangular wave.

Two simultaneous **equations** result when you apply nodal analysis at the noninverting input of the comparator and solve for these two thresholds:

$$\frac{V_{IH}}{R_1}+\frac{V_{SS}}{R_2}=V_{REF}\left[\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}\right],$$

and

$$\frac{V_{IL}}{R_1}+\frac{V_{DD}}{R_2}=V_{REF}\left[\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}\right].$$

In this example, the voltage range of the triangular wave is 0.5 to 4.5V. You therefore substitute a value for V_{IH} of 4.5V, V_{IL} of 0.5V, V_{DD} of 5V, and V_{REF} of 1.23V into the above **equations** to obtain a value of 124 kΩ for R_1 and 66.5 kΩ for R_3.

You can now design the integrator. Considering the comparator's two possible output states, the magnitude of current flowing through R_4 is: $I_{R4}=(V_{DD}-V_{REF})/R_4$, or $I_{R4}=V_{REF}/R_4$. The op amp's maximum input-bias current is 2 nA. To minimize error, therefore, the current through R_4 must always be greater than 0.2 µA. This constraint implies that R_4's value is less than 6.12 MΩ.

The triangular-waveform frequency is:

$$f=1/\left[\frac{V_{OUT\,P-P}}{(V_{CC}-V_{REF})}(R_4C)+\frac{V_{OUT\,P-P}}{V_{REF}}(R_4C)\right].$$

For this example, the frequency is 25 kHz, the output voltage is 4V p-p, or 0.5 to 4.5V for a triangular wave, and the reference voltage is 1.23V. Solving for the resulting time constant, R_{4C}=9.27 µsec. Select a capacitance of 220 pF and a value of 42.2 kΩ for R_4.

The resulting output should match the desired frequency, provided that the op amp is not slew-limited. Because the feedback capacitor charges or discharges with a constant current, the output signal's maximum rate of change is:

$$\frac{dV_{OMAX}}{dt}=\frac{I_{R4MAX}}{C}=\frac{V_{CC}-V_{REF}}{R_4C}=0.406\frac{V}{SEC}.$$

To provide a margin against process variations, the op amp's typical slew rate should be 40% higher than the maximum rate of change of the output signal—0.56V/µsec or greater in this case. The op amp's slew rate is 0.85V/µsec, which is therefore adequate for this 25-kHz waveform (**Figure 3**).EDN

REFERENCE

Terrell, David L, *Op Amps: Design, Application and Troubleshooting*, Butterworth-Heinemann, 1996.

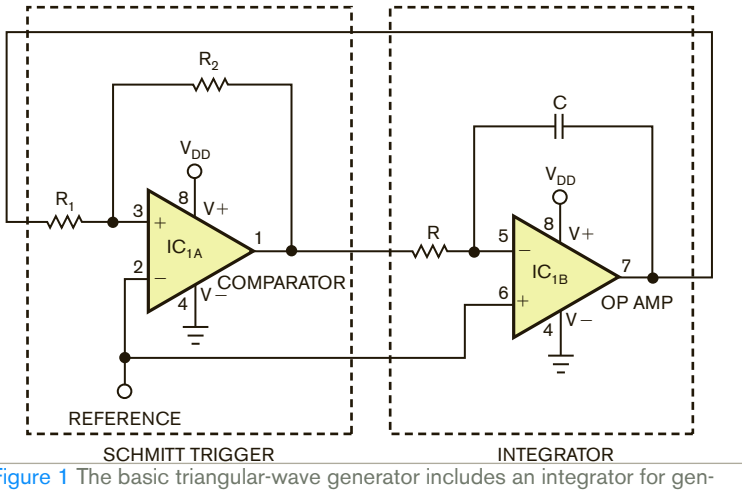


Figure 1 The basic triangular-wave generator includes an integrator for generating the triangular-wave output and a comparator with external hysteresis, such as a Schmitt trigger, for setting the output amplitude.

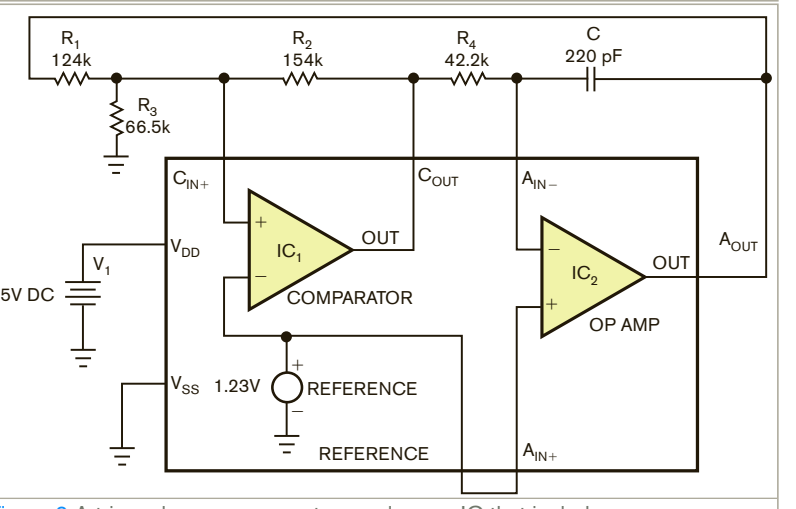


Figure 2 A triangular-wave generator employs an IC that includes an op amp, a comparator, and a bandgap reference.

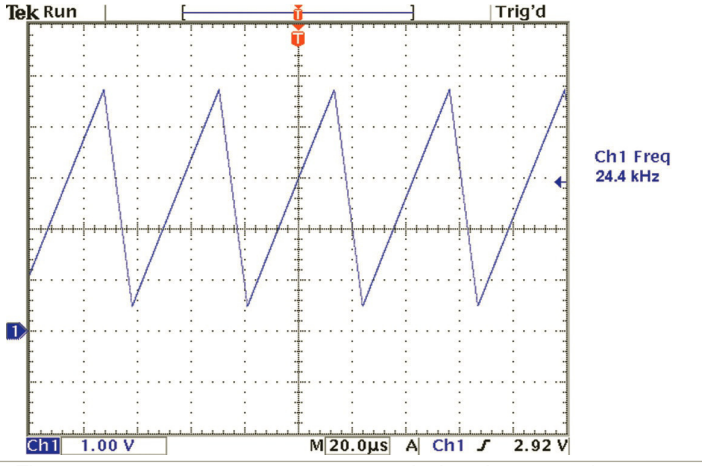


Figure 3 The op amp's slew rate is 0.85V/µsec, which is therefore adequate for this 25-kHz waveform.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Eight-digit counter works with common anode or common cathode

Charaf Laissoub, Valeo Interior Controls, Créteil, France

A classical design for directly driving eight seven-segment LED displays requires as many as 15 I/O lines. Previous Design Ideas have described many approaches for using a maximum number of LEDs with a minimum number of I/O lines (**references 1** through **5**). The following idea reuses one of these approaches to drive a maximum number of seven-segment LED displays, and it may be useful in designing a low-component-count, low-power, and low-cost LED-display module for a 24-bit frequency meter, for example.

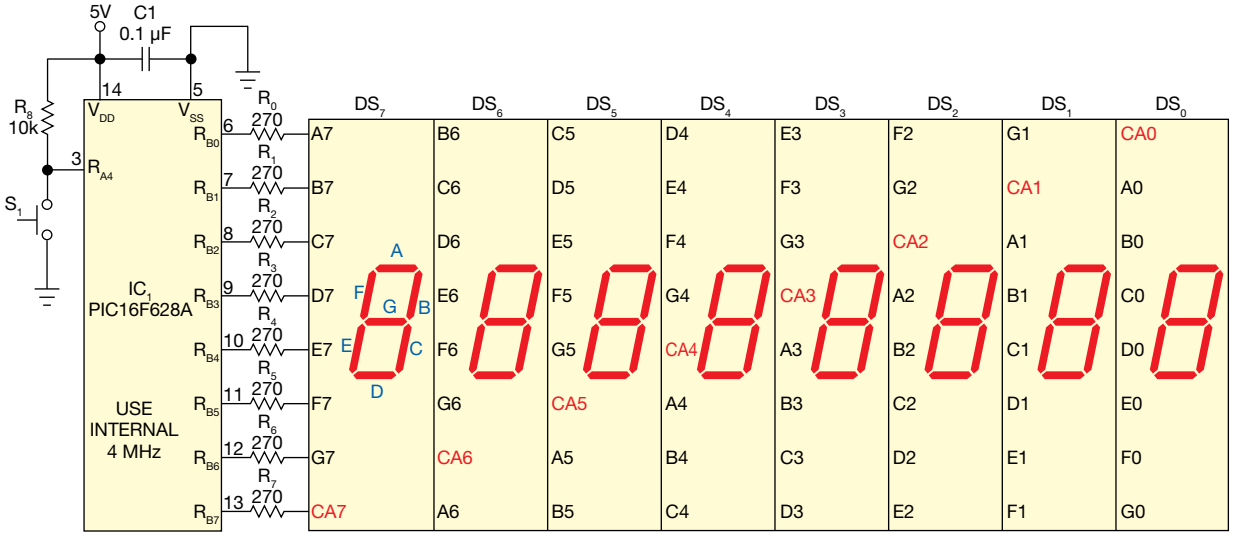
You can use the circuit in **Figure 1** to replace classical designs for digital counters that use TTL (transistor-transistor logic) or CMOS ICs. The single microcontroller is less expensive and readily available. By using conditional assembly in your programming, you can choose between common-anode and common-cathode configurations.

The algorithm uses double multiplexing, driving one digit at a time, segment by segment. This technique suits battery-powered designs because the circuit consumes a constant current of less than 2 mA when using superbright, seven-segment LED displays, such as KingBright's (www.kingbright.com) SC52-11EWA, and 270Ω resistors R_0 to R_7. Assembling the eight digits, DS_7, DS_6, DS_5, DS_4, DS_3, DS_2, DS_1, and DS_0, on a PCB (printed-circuit board) involves linking their corresponding pins A7, B6, C5, D4, E3, F2, G1, and CA0 to the I/O line, R_{B0}. **Figure 1** shows the connections.

This circuit uses the Microchip (www.microchip.com) PIC16F628A for test purposes. You can download assembly code at www.edn.com/100715dia and use it with any PICmicro midrange family, providing that a full 8-bit port is available. EDN

DIs Inside

▸To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.



NOTE: CHOOSE EIGHT COMMON-ANODE OR EIGHT COMMON-CATHODE, SEVEN-SEGMENT LED DISPLAYS AND UNCOMMENT ONE OF THESE TWO LINES IN THE INCLUDE FILE: "8dgtcacc.inc" #define use CAdisplay IF USING COMMON-ANODE, SEVEN-SEGMENT, DISPLAY OR #define use CCdisplay IF USING COMMON-CATHODE,SEVEN-SEGMENT DISPLAY.

Figure 1 This circuit can replace classical designs for digital counters that use TTL or CMOS ICs.

REFERENCES

1 Anonymous, "Microcontroller provides low-cost analog-to-digital conversion, drives seven-segment displays," *EDN*, May 10, 2007, pg 80, http://bit.ly/b3KBre.

2 Raynus, Abel, "Squeeze extra outputs from a pin-limited microcontroller," *EDN*, Aug 4, 2005, pg 96, http://bit.ly/aKKGEf.

3 Jayapal, R, "Microcontroller's single I/O-port line drives a bar-graph display," *EDN*, July 6, 2006, pg 90, http://bit.ly/barbMI.

4 Lekic, Nedjeljko, and Zoran Mijanovic, "Three microcontroller ports drive 12 LEDs," *EDN*, Dec 15, 2006, pg 67, http://bit.ly/deQTib.

5 Gadre, Dhananjay V, and Anurag Chugh, "Microcontroller drives logarithmic/linear dot/bar 20-LED display, *EDN*, Jan 18, 2007, pg 83, http://bit.ly/azQ9dw.

Count objects as they pass by

Vladimir Rentyuk, Zaporozhye, Ukraine

Counting objects is easy when the objects have regular motion, but it becomes more difficult when the objects vibrate or you have to manually move them. To deal with this problem, you need a reliable, error-free system of detecting an object. In this case, simple circuits with optical interruption switches, IR barriers, or other sensors don't work because objects may cross a sensor more than once.

The circuit in **Figure 1** solves this problem for objects such as cards, boxes, and even people. You can detect other objects if you use the proper sensors. The circuit produces two control pulses. One, OUT_1, occurs if any object runs through this system. It produces another pulse, OUT_4, only if an identified object passes through the system. These pulses let you count both the number of objects and the identified objects that pass through the system. The system doesn't produce counting errors if objects repeatedly cross each sensor, even when the objects return to the system, provided the object does not move from Sensor 1's zone. The system requires no difficult mechanical unit for stabilizing the speed of the moving objects. The counter selects objects that are slightly longer than the distance between its sensors. **Figure 2** shows how the circuit tracks an object between the sensors.

THE CIRCUIT PRODUCES TWO CONTROL PULSES.

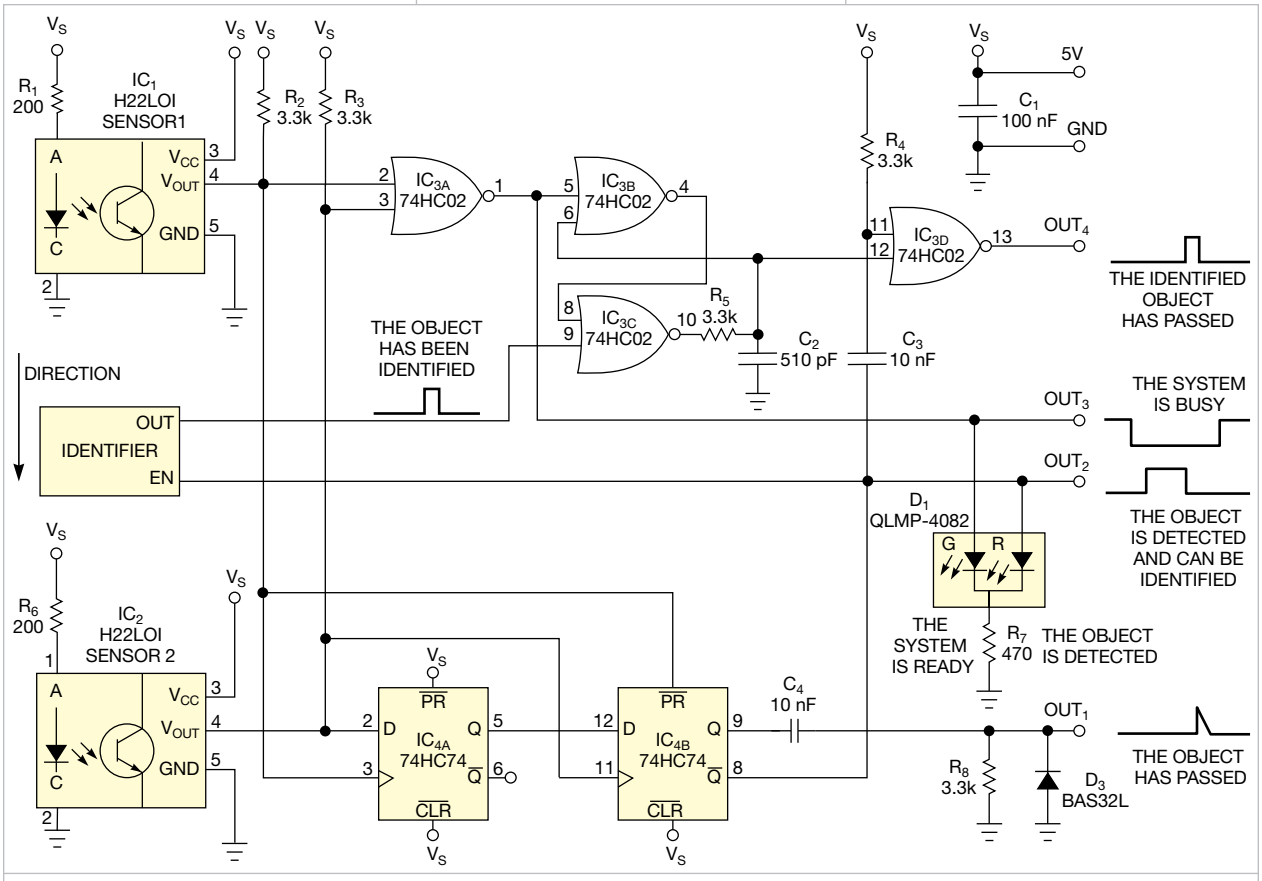


Figure 1 This circuit counts objects such as cards, boxes, and even people.

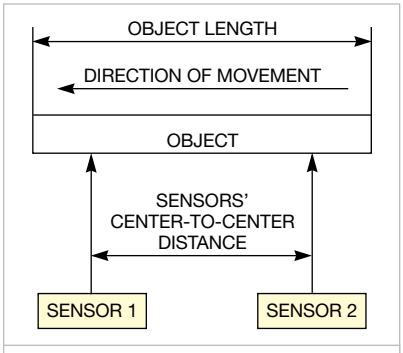


Figure 2 The circuit tracks movement using an object's length and direction.

The object counter comprises IC_1 and IC_2, H22LOI sensors from Fairchild Optoelectronics Group (www.fairchildsemi.com). The sensors are optical-interruption switches with open-collector outputs. A control circuit uses IC_4, a 74HC74 dual D-type flip-flop, and IC_3, a quad, two-input NOR gate. Sensors IC_1 and IC_2 produce logic-low output levels when objects are not between the IR diode and the IR receiver. LED D_1 shines green if the system is ready to take an object and red if the system has detected the object.

If the system is busy, LED D_1 does not illuminate. Resistor R_5 and capacitor C_2 protect RS trigger IC_{3B}/IC_{3C} from the chance of failure due to undesired signals. Some sensors require that you add buffers between them and the control circuit.

Table 1 describes the circuit's signals. A downloadable sheet describing the sequence of events is available at www.edn.com/100715dib.EDN

TABLE 1 SIGNAL DESCRIPTION

Trace or zone	Indication
Trace 1	OUT_1: common counting of passing objects
Trace 2	OUT_2: detection of an object
Trace 3	OUT_3: ready to take an object
Trace 4	OUT_4: counting an identified object
Zone A	Object identified
Zone B	Object not identified

Modified DDS functions as baud-rate generator

Vardan Antonyan, Glendale, CA

You can often use an available oscillator to generate a baud-rate clock for a UART. You must divide the oscillator frequency to attain the proper baud rate, but dividing can produce baud-rate errors. **Table 1** shows the percentage of error when you generate a baud rate using an 8-MHz crystal oscillator and a conventional binary divider. The system in this Design Idea obtains a clock 16 times faster than the baud rate.

Errors in baud-rate setting increase when the oscillator frequency doesn't match. In this case, you can add an oscillator operating at 18.432 MHz, for example, to minimize the error rate. Alternatively, you can use DDS (direct digital synthesis) to reduce errors at higher baud rates using the same oscillator (**Table 2**).

Reference 1 describes basic DDS operation. This design uses a simpler version of DDS with only a square-wave output (**Figure 1**). You can extract the square-wave output from the MSB of the phase accumulator. You can also add the divide-by-two stage to make the resulting signal with a 50% duty cycle. Calculate the baud-rate clock frequency using baud-rate clock=(reference clock×tuning word/2^N)/2, where N is the number of bits for the phase accumulator. A Verilog implementation of the DDS baud-rate generator using a 20-bit phase accumulator and 16-bit tuning word is available at www.edn.com/100715dic.EDN

TABLE 1 BAUD RATE WITH REGULAR DIVIDER

Baud rate	Divisor	Error (%)
50	10,000	0
300	1666	0.04
600	833	0.04
2400	208	0.16
4800	104	0.16
9600	52	0.16
19,200	26	0.16
38,400	13	0.16
57,600	8	7.84
115,200	4	7.84
230,400	2	7.84

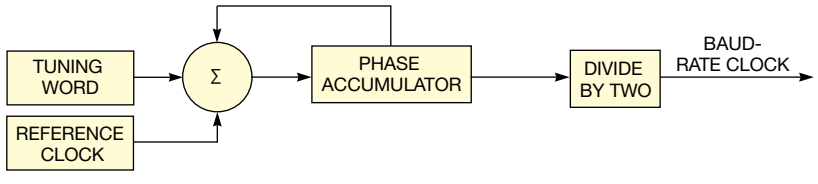


Figure 1 You can generate a baud-rate clock from an available oscillator.

REFERENCE

1 "A Technical Tutorial on Direct Digital Synthesis," Analog Devices, 1999, www.analog.com/static/imported-files/tutorials/450968421DDS_Tutorial_rev12-2-99.pdf.

TABLE 2 BAUD RATE WITH 20 DDS BITS

Output frequency	Phase word	Error (%)
50	13	–0.825
300	78	–0.825
600	157	–0.182
2400	629	–0.023
4800	1258	–0.023
9600	2516	–0.023
19,200	5033	–0.003
38,400	10,066	–0.003
57,600	15,099	–0.003
115,200	30,198	–0.003
230,400	60,397	–0.002

DC-voltage doubler reaches 96% power efficiency

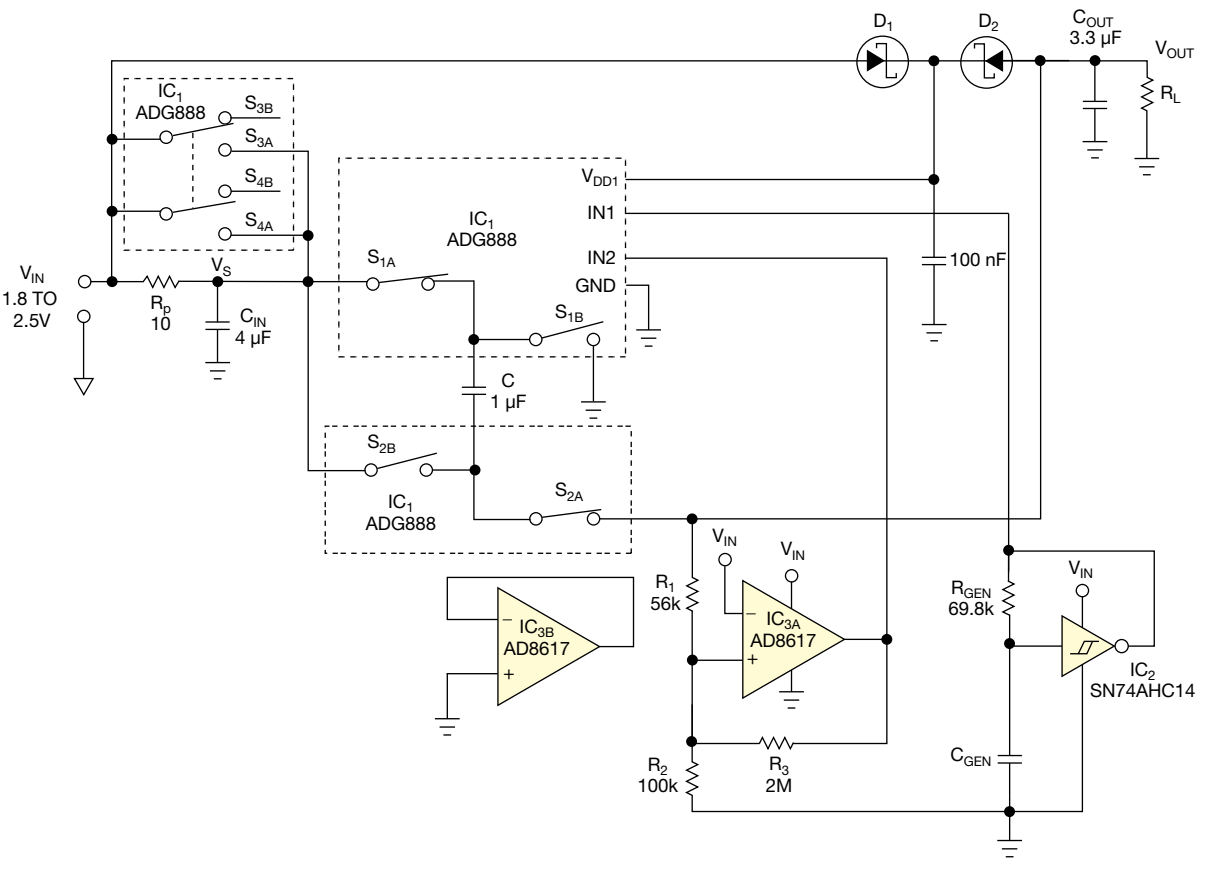
Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

The voltage-doubler circuit in **Figure 1** can convert 2.5V dc to 5V dc or 1.8V to 3.3V. Most voltage doublers use an inductor, but this circuit doesn't need one. The circuit uses a capacitor, C, by charging it through serially connected switches. The charge switches let capacitor C charge, and the discharge switches are open. In the subsequent discharging phase, the charge switches are off, and the discharge switches close. The two discharge switches now connect capacitor C between the source of the input voltage, V_S, and the output capacitor, C_{OUT}. This connection scheme lets the applied voltages combine. Thus, the voltage at the output terminal has a value close to $2V_S$.

The two phases of operation repeat periodically at frequency f, which clock generator IC_2 determines. The duty cycle is about 50%, but the value isn't all that critical. One half of the Analog Devices (www.analog.com) high-performance ADG888 analog multiswitch provides the switching. The IC's two halves have independent control, so the other half occasionally shorts R_P, the 10Ω inrush-current-limiting resistor, which protects the charge switches from an initial overcurrent. That current occurs after power-on, before the output voltage reaches the predetermined percentage of the output's full voltage.

A micropower op amp, IC_{3A}, runs as a comparator with hysteresis. It compares input voltage to output voltage. Its output starts low and then goes high, which turns on paralleled switches S_3 and S_4. The comparator's action is ratiometric because the reference input voltage at the inverting input is the input-supply voltage, V_{IN}. This connection is possible because of the AD8617's rail-to-rail input/output operation. The circuit also provides overload protection for an excessive load, which connects to the circuit's output before power-on.

During soft start, the output voltage can't reach the threshold level for loads below a certain value. Consequently, the circuit remains in soft-start mode. The minimum value of R_L, which activates the protective subcircuit, is $R_L \le m^2 \times (\alpha/(1-\alpha)) \times R_P$, where the multiplication factor $m=(V_{OUT}/V_{IN})$ and α is



NOTE: C_{GEN} HAS AN OPTIMAL VALUE OF 62 pF WHEN R_L IS APPROXIMATELY 180Ω.

Figure 1 You can use this step-up dc/dc converter in applications in which power efficiency is a critical issue.

a fraction of V_{OUT}, at which the soft start turns off. For m=2, α=0.8, and R_P=10Ω, R_L is 160Ω. Thus, loads of 160Ω or less will overload the circuit if you connect them to the circuit's output before power-on. IC_2 and IC_3 get their power from the input supply. IC_1, however, switches voltages of as much as $2V_{IN}$, and its V_{DD1} supply-voltage pin must remain at the same level. An analog OR switch comprising Schottky barrier diodes D_1 and D_2 provides that voltage. The higher of the input or output voltages appears at the V_{DD1} pin of IC_1. The high levels of output voltages for both IC_2 and IC_3 suffice for control of IC_1 because the ADG888's data sheet allows a $0.36V_{DD1}$ value for the high value at the control inputs. The circuit has been tested at an input voltage of 2.386V, R_L of 178.46Ω, a frequency of 200 kHz, a supply voltage of 2.377V, an input supply current of 51.285 mA, and an output voltage of 4.588V. Evaluating these data gives a multiplication factor of 1.929 and power efficiency of 96.39%.

This power efficiency remains more than 96% for frequencies of 150 to 350 kHz. The 9-mV drop at the switch-shorted R_P at the given input current indicates that the on-resistance of the paralleled switches has a value of approximately 0.175Ω.EDN

Microcontroller's serial port measures pulse width

Vishwas Vaidya, Tata Motors Ltd, Pune, India

Many industrial and instrumentation systems need to measure the duration of pulse inputs, such as frequency from rotational-speed sensors, gating and strobe pulses from external systems, and PWM (pulse-width-modulated) inputs. Designers generally use on-chip timers and edge-driven interrupts for this purpose. If one of these components is unavailable, however, you can employ an unused on-chip serial-synchronous receiver to make those measurements.

You can set the baud rate of the serial-port receiver for the necessary timing accuracy. The receiver interrupts the microcontroller after every 8 bits. You can embed the pulse-width acquisition routine, which resides in your application program, to read the byte that the ISR (interrupt-service routine) receives. It counts and accumulates the number of ones and zeros the bytes receive to measure the duration of an incoming pulse (**Figure 1**).

The algorithm measures the duration between two consecutive rising edges.

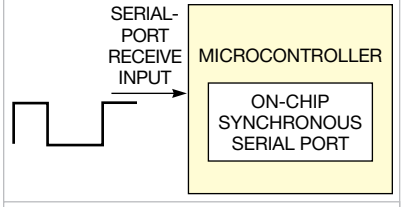


Figure 1 A microcontroller's serial port can receive pulses from various sources.

The microcontroller detects a rising edge or a falling edge when a received byte is neither 0xff nor 0x00. If a byte is less than 0x80H (100 000b), then the byte marks a rising edge. If the byte is equal to or greater than this value, then the byte marks a falling edge.

The leading rising edge resets the bit counter to the number of trailing ones in the rising-edge byte by shifting the bits to the right. The bit counter increments by eight at the arrival of every byte, including the one that marks the falling edge. When the counter receives the trailing rising edge, marked by the next rising-edge byte, which is greater than 0x80h but less than 00H, it again counts the number of leading zeros in this byte and adds them to the accumulated-bit counter. The accumulated-bit count at this point directly relates to the time period of the pulse train by a factor equaling the baud rate.

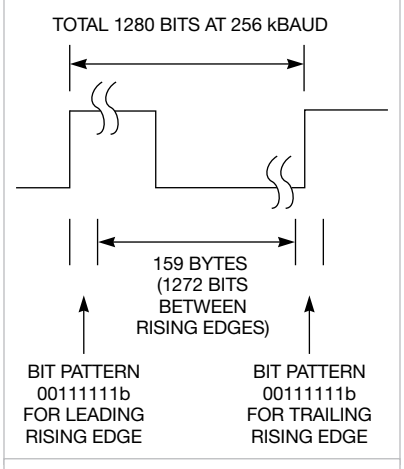


Figure 2 The algorithm uses a 001111 bit pattern to detect edges.

Figure 2 depicts a 200-Hz pulse train, which has a 5-msec period between two consecutive rising edges. The baud rate is 256 kbaud. During a measurement cycle, assume that the leading rising edge is marked as 0011 1111b. The microcontroller counts the number of trailing ones by shifting them right and initializing the bit counter as six. This count corresponds to approximately 23.43 µsec.

Next, every byte before the rising-edge byte increments the bit counter by eight. Simple calculation shows that the sum is 159 bytes, or 1272 bits. At this point, the total bit count is 1278, including six one bits received in the first rising-edge byte.

The pulse train now encounters its trailing rising-edge byte as 0011 1111b. When this encounter occurs, you need to shift the zeros left to count two bits. The total bit count between the rising edge now is 1280. At a 256-kbaud rate, this figure corresponds exactly to 5 msec, or 200 Hz.

Figure 3, a flow chart, is available with the online version of this Design Idea at www.edn.com/100715did. It explains how you can use this concept to measure frequencies in hundreds of hertz.

You can tailor this bit-counting concept to your application's requirements. For measuring only a low period of a pulse, you need to detect a falling edge and count the bits until you encounter a rising edge. You can use this concept to read an incoming PWM signal by reading high periods of a known incoming pulse frequency.EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Methods measure power electronics' efficiency

Liping Zheng, Calnetix, Yorba Linda, CA

Validating the sytem efficiency of a power-electronics circuit is essential in evaluating the overall system performance, design optimization, and sizing of cooling systems. **Figure 1** shows the conventional method of performing efficiency measurement. The power-electronics system operates at the rated output-power level, and, by measuring the input power and output power, you can calculate the system's efficiency using the **equation** $\eta=(P_{OUT}/P_{IN})\times100\%$, where P_{OUT} is output power and P_{IN} is input power. In other words, the measured input power is equal to the output power plus the power loss of the system.

However, measuring the efficiency of a high-power system that delivers power to loads such as motors, generators, or industrial-computer equipment requires a source that delivers the rated power. The infrastructue therefore should comprise a suitably rated source and an equivalent load that can support the rating of the power-electronics system you are evaluating. These requirements can drive up the facility's infrastructure cost; for one-time design-validation measurements, this cost is difficult to justify.

This Design Idea describes alternative methods of measuring the efficiency of a high-power power-electronics system that simplifies the test-infrastructure requirement by eliminating the test load and using a source that must support only the loss of the power-electronics system. **Figure 2** shows the proposed method, which eliminates the test load by shorting the output/load terminals. The system's control algorithm maintains the required input- and output-current amplitude and frequency by developing circulating reactive power. IGBTs (insulated-gate bipolar transistors) and magnetic components dominate the system's losses, which are functions of the amplitude and frequency of the input and output currents. The loss is also less sensitive to the power-factor and PWM (pulse-width-modulation) index.

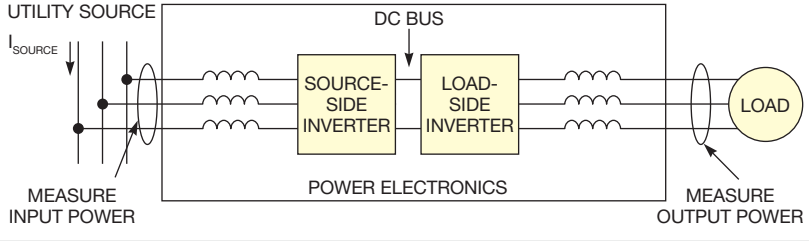


Figure 1 In a conventional method of performing efficiency measurement, the power-electronics system operates at the rated output-power level.

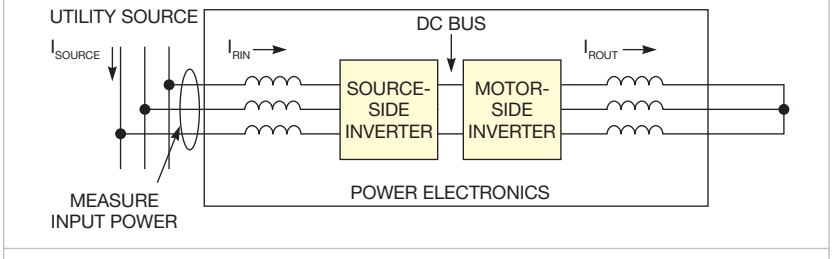


Figure 2 This method eliminates the test load by shorting the output-load terminals.

DIs Inside

▸To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

To know the required input and output current, you must estimate the system's power factor, the motor's back EMF (electromotive force), and the systems's source voltage. This example uses a field-oriented control for both source- and load-side inverters, resulting in the following **equations**:

$$I_{ROUT} = I_{ROUT\_RE} + jI_{ROUT\_IM} = \frac{P_{OUT}}{\sqrt{3}V_{BEMF}};$$

$$I_{RIN} = I_{RIN\_RE} + jI_{RIN\_IM} = \frac{P_{RIN}}{\sqrt{3}V_{GRID}} = \frac{P_{OUT}/\eta_E}{\sqrt{3}V_{GRID}},$$

where I_{ROUT} is the required output current, which comprises real current, I_{ROUT_RE}, and reactive current, I_{ROUT_IM}; I_{RIN} is the required input current, which comprises the real current, I_{IN_RE}, and the reactive current, I_{IN_IM}; P_{RIN} is the

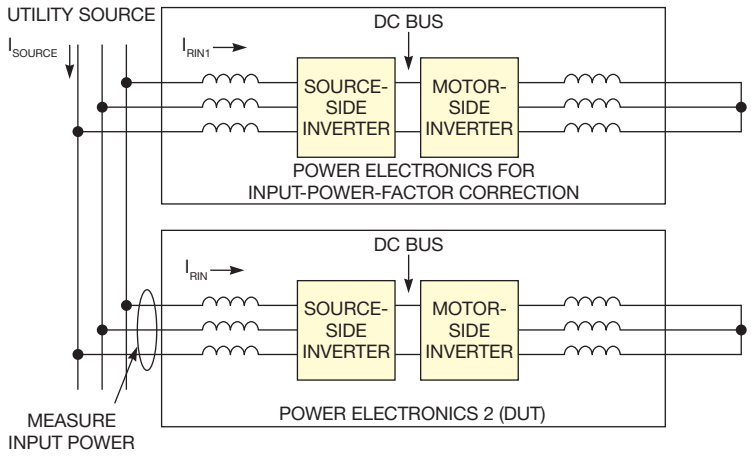


Figure 3 This method uses two identical power-electronics systems. The second system offsets the input reactive current that the test system creates.

required input power; P_{OUT} is the output power at the test condition; V_{BEMF} is the motor's back EMF; V_{GRID} is the grid voltage; and η_E is the estimated efficiency of the circuit.

By maintaining the input current to be I_{RIN} and the output current to be I_{ROUT}, the measured input real power will be close to the power loss, P_{LOSS}, at the actual output-power level, P_{OUT}. Therefore, you can calculate the efficiency as follows: $\eta=(P_{OUT})/(P_{OUT}+P_{LOSS})\times100\%$.

If the measured efficiency, which you calculate using this **equation**, does not quite match the estimated efficiency, η_E, update the second **equation** using the measured efficiency, η, and repeat the measurement until they are close. Calnetix (www.calnetix.com) has used this method to evaluate the efficiency of a 125-kW power-electronics system, compared the results with the conventional measurements, and found them to be closely matching.

Most high-power power-electronics systems have high efficiency, which means that the real current is much less than the reactive current. To reduce the required current from the grid, you can use the method in **Figure 3**, which uses another identical system to offset the input reactive current that the test system creates. By providing a path for circulating reactive power, the utility sources the lost power only, not the total power. In **Figure 3**, the input current of the second power-electronics circuit is $I_{RIN}=I_{RIN\_RE}+jI_{RIN\_IM}$. By setting the first circuit to have an input current of $I_{RIN1}\approx I_{RIN\_RE}-jI_{RIN\_IM}$, the power from the source is only $I_{SOURCE}=I_{RIN1}+I_{RIN}\approx I_{RIN\_RE}+I_{RIN\_RE}+j(I_{RIN\_IM}-I_{RIN\_IM})=2I_{RIN\_RE}$. The circuit uses the input current from the source only to overcome the power losses of the two circuits, thereby eliminating the need for a high-power infrastructure. EDN

Circuit extends battery life

José M Espí, Rafael García-Gil, and Jaime Castelló,
University of Valencia, Valencia, Spain

Two previous Design Ideas describe simple ways to automatically disconnect a battery from its load after a preset on period, which extends battery life (**references 1** and **2**). These circuits have little loss in standby operation, but they do draw some current. The circuit in this Design Idea presents a simpler way to perform the same function with fewer components and with no power consumption during standby operation (**Figure 1**). Moreover, the network comprising R_2, D_2, and C_2 activates and deactivates the circuit. An additional control signal, control on/off, becomes slower than the battery's on/off cycle.

Switching S_1 to Position 1, the on position, the 24V battery quickly charges capacitor C_1 through diode D_1. That voltage drives transistor Q_1 into saturation. Q_1's saturation magnetizes and activates relay coil L_1, connecting the battery to the main power and control board. Meanwhile, capacitor C_2 charges more slowly through 100-kΩ resistor R_2, thus generating the control on/off signal with some delay relative to the relay coil's closing. That scenario occurs after

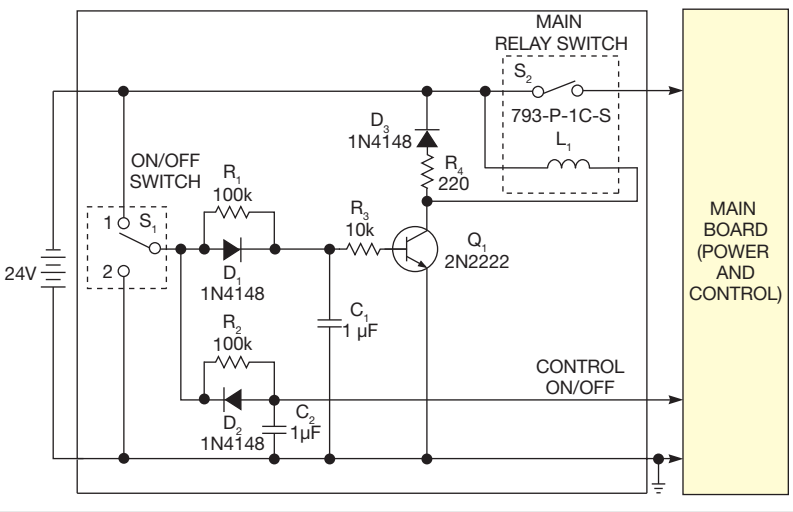


Figure 1 This on/off management circuit uses a relay to remove battery power.

the proper power supply to the power stage and control circuits.

Switching S_1 to Position 2, the off position, causes capacitor C_1 to slowly discharge through resistor R_1 when diode D_1 is off. That action delays Q_1's turn-off. Before Q_1 turns off, C_2 quickly discharges through D_2, indicating that the control should shut down the power. The relay switches off with minimum current. Once Q_1 is off, the relay coil demagnetizes through R_4 and D_3. The relay switches off, disconnecting the main power and control board from the battery. During this off state, current flows neither in the on/off circuit of the management board nor to the main board. EDN

REFERENCES

1 Gimenez, Miguel, "Scheme provides automatic power-off for batteries," *EDN*, May 13, 2004, pg 92, http://bit.ly/aUdD3s.

2 Xia, Yongping, "Battery automatic power-off has simpler design," *EDN*, March 31, 2005, pg 80, http://bit.ly/bLJNgb.

Pulse generator corrects itself

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

Using a shift register with parallel output is a common way to design a pulse generator with N inputs and pulsed outputs having a width of T/N. To keep the output pulses consecutive, you can use feedback from the last output to the first input. At power-on, such a circuit can have a random combination of logic zeros and ones, forming an undesired data content of the shift register. To avoid circulating undesired states and to enter a proper sequence, you need a special feedback.

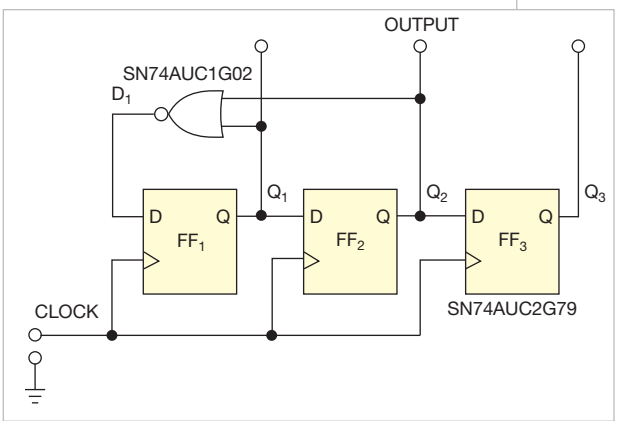


Figure 1 This one-of-three pulse generator uses three ICs, yet it avoids false logic states.

The circuit in **Figure 1** is a three-stage shift register that uses D-type flip-flops. It has three outputs, Q_1, Q_2, and Q_3, each of which produces a periodic pulse having a width of $T_{REP}/3$. $T_{REP}=3T_{CLK}$ is the period at which the sequence repeats at any of the three outputs. A two-input NOR gate creates the feedback. The gate's D_1 output connects to the D input of flip-flop FF_1, and its inputs connect to Q_1 and Q_2. A logic-one bit at D_1 means that, at the nearest low-to-high transition of the clock, this signal will place a logic one at output Q_1.

You can interpret this feedback in words by writing a logic zero into FF_1 at the nearest low-to-high transition of the clock signal, if at least one of the Q_1 or Q_2 outputs has a logic-one state. You write a logic one into FF_1 if both the Q_1 and the Q_2 outputs are at logic zero. This feedback adds a self-correcting feature, which is illustrated by the assumption that the initial state of the circuit is intentionally undesired.

Using this result, the following sequences illustrate state correction, in which the logical states in the bit triads correspond left to right to Q_1, Q_2, and Q_3:

111→011→001→100→010→001

000→100→010→001

From this example, you can see that erroneous state 111 self-corrects within two periods of the clock. For the undesired 000 state, the proper cycling enters at the nearest low-to-high transition of the clock signal.

You can determine the upper limit of the clock frequency from an assumption of the gate output, which changes after a low-to-high transition of the clock. This condition must be ready with a setup time, T_{SETUP}, the next time the clock transitions from low to high (**Figure 2**). Thus, $T_{CLKMIN}=T_{PQHL}+T_{PGLH}+T_{SETUP}$, where T_{PQHL} and T_{PGLH} are signal-propagation delays of the flip-flop and the gate, respectively, at the respective output-level transition. By using the worst-case values of propagation delays from the devices' data sheets, you get a minimum clock period of 4.4 nsec for a supply voltage of 1.8V and a minimum clock period of 3.5 nsec for a supply voltage of 2.5V. As the 3.5-nsec value gives a clock frequency higher than the guaranteed toggle frequency for the flip-flop, you should accept the maximum clock frequency at 275 MHz for a supply voltage of 2.5V. For a supply voltage of 1.8V, the maximum clock frequency should be 227 MHz. The maximum repetition rate of signals at Q_1, Q_2, and Q_3 outputs is the maximum clock frequency divided by three, or 75.6 MHz. EDN

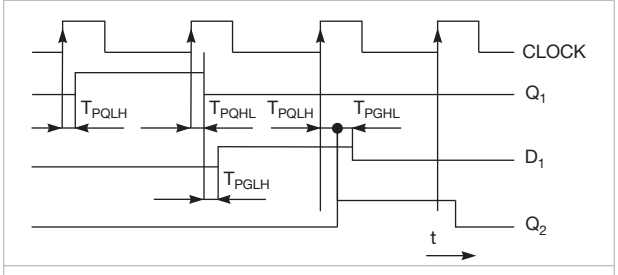


Figure 2 The circuit generates glitch-free logic waveforms that are successive in time and have accurate duty cycles of 33.3%.

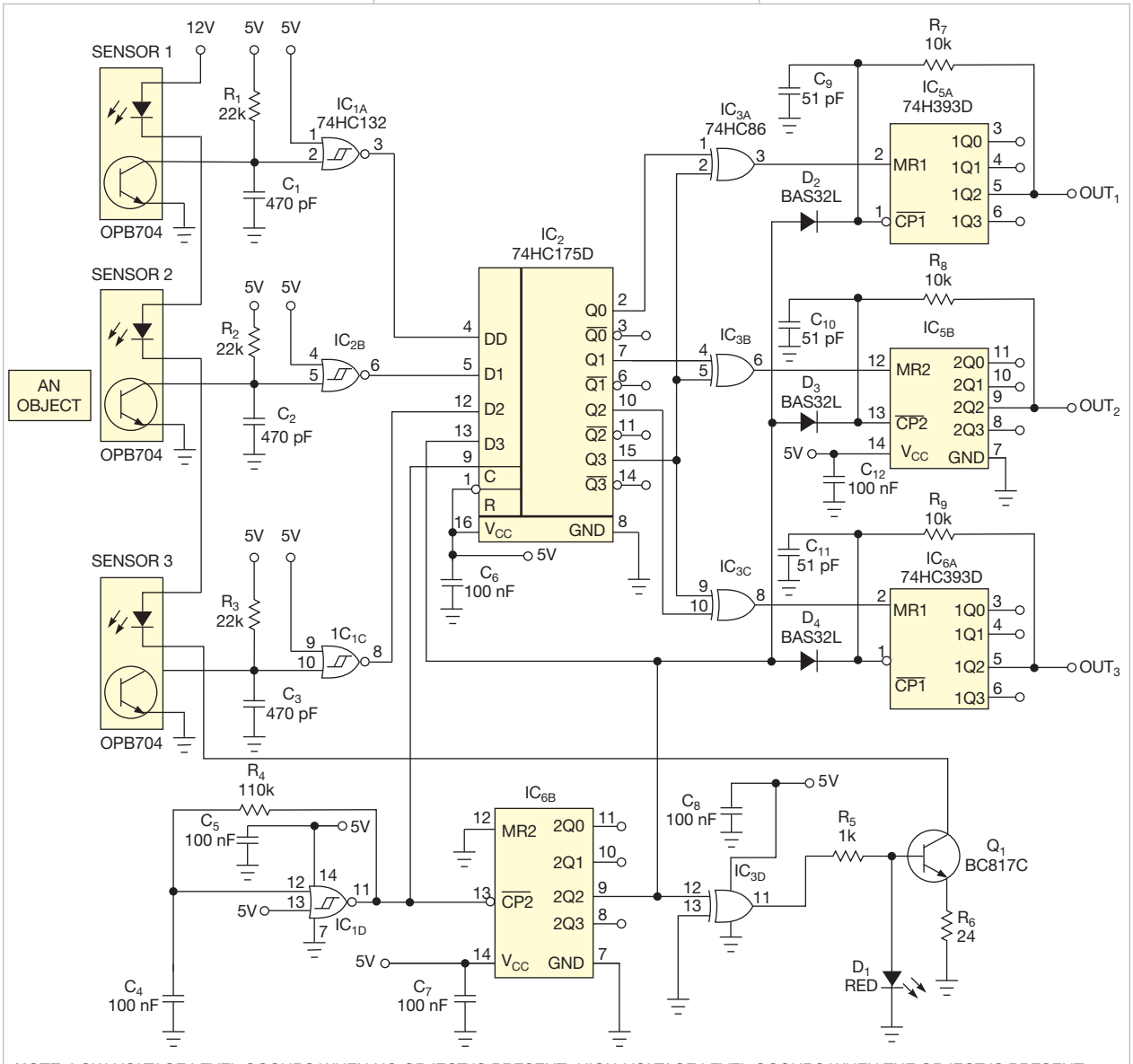
Reflective object sensor works in bright areas

Vladimir Rentyuk, Modul-98 Ltd, Zaporozhye, Ukraine

When using a reflective object sensor, counting and identifying objects is sometimes difficult in the presence of electrical noise or bright ambient light. The circuit in **Figure 1** shows an inexpensive solution to this problem using three independent and simultaneously working reflective object sensors. The circuit is suitable for many types of objects, but it targets use with objects such as cards.

The circuit uses three OPB704 reflective optical sensors with Schmitt-trigger NAND comparators IC_{1A}, IC_{1B}, and IC_{1C} on each output. IC_{1D} functions as a clock generator, and counter IC_{6B} functions as a divide-by-eight counter that divides the clock frequency. That signal drives IC_{4D}, which acts as a buffer to drive transistor Q_1.

To understand how the circuit works, consider Sensor 2. IC_{1B}'s output will be low if the sensor's phototransistor doesn't detect IR rays reflected from an object. Both of IC_{1B}'s inputs are high; therefore, the D1 input of IC_2 is low. In any case, if the sensor's phototransistor detects IR rays reflected from an object, the D1 input of IC_2 is high. The level corresponding to the current situation transfers through IC_2's Q_1 output (Pin 7) by a write signal on the C input (Pin 9). The write signal is a leading edge of puls-



NOTE: LOW-VOLTAGE LEVEL OCCURS WHEN NO OBJECT IS PRESENT; HIGH-VOLTAGE LEVEL OCCURS WHEN THE OBJECT IS PRESENT.

Figure 1 Infrared sensors and logic circuits detect the presence of an object.

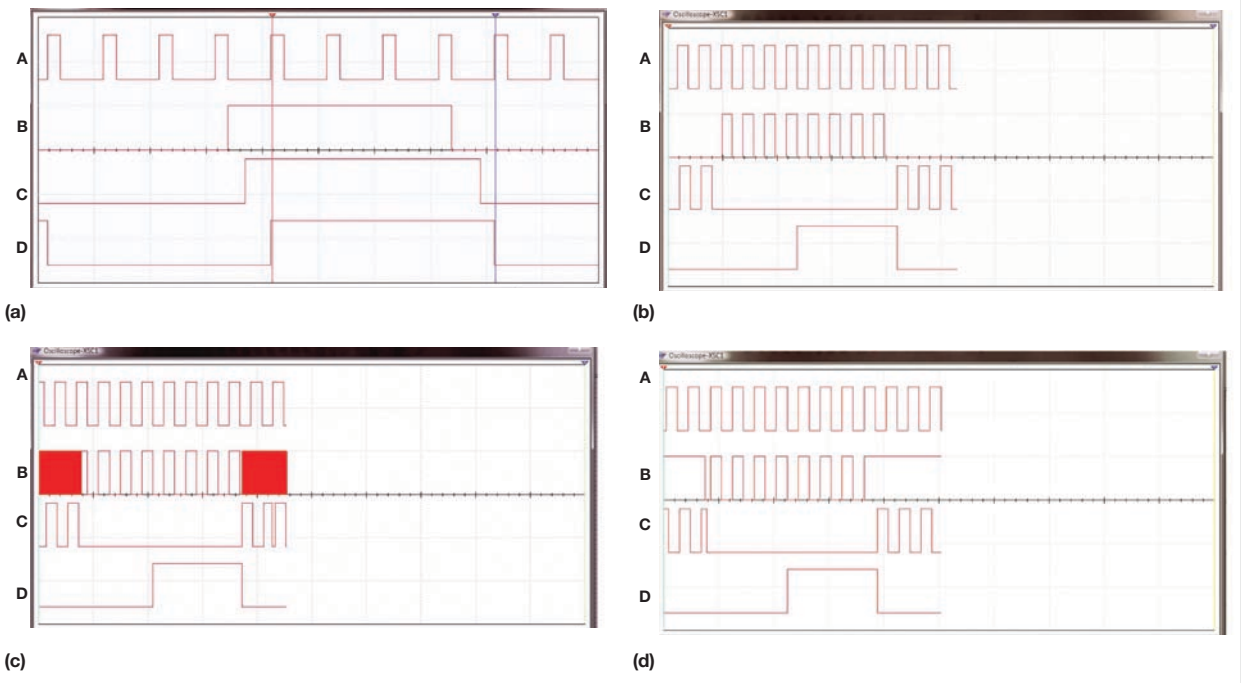


Figure 2 Traces show the process of normalization (a) and three cases of using the presented device (b, c, and d).

es from the clock generator. The signal from divider IC_{6B} becomes the D3 input of IC_2. A level of the divided clock signal transfers to IC_2's Q3 output (Pin 15) upon receiving a write signal from the C input (Pin 9). The signals on the Q1 and Q3 outputs have equal duration except when the sensor's phototransistor detects IR rays reflected from an object. **Figure 2a** shows the process of this normalization. Exclusive-OR gate IC_{3B} compares the Q1 and Q3 outputs from IC_2. If they have the same logic level and duration, then IC_{3B}'s Pin 6 is low, and IC_{3B} generates pulse signals. If signals from outputs Q1 and Q3 on IC_2 are unequal, you must reset counter IC_{5B}'s reset signal, and its output 2Q2 at OUT_2 is low. The Q2 outputs of counters IC_{5A}, IC_{5B}, and IC_{6A} are low whenever the input signals of comparator circuits IC_{3A}, IC_{3B}, and IC_{3C} are unequal. This situation occurs if Sensor 2 doesn't detect an object or receive any external signals—for example, IR noise from fluorescent lamps or interfering ambient light, alternating light, or flashes.

The outputs of IC_{3A}, IC_{3B}, and IC_{3C} are equal only when all phototransistors detect a signal from their respective IR emitting diodes—that is, when a card is presented in front of Sensor 2 (**Figure 2b**). You must choose a clock frequency with regard to a delay time of the system. A leading edge triggers IC_2, a 74HC175, and a falling edge triggers IC_{6B}, a 74HC393. Because of the counters, this system automatically adjusts itself after any changes of frequency in its clock generator. Thus, if counter IC_{5B} does not have a reset signal during a period equal to four periods of a reference signal, its output (Pin 9) is high, and the counter latches through R_8. The logic-high level appears on OUT_2 until you remove the card. In this case, the detected inequality signal from the sensor with the reference signal and the counter, IC_{5B}, causes a reset signal. **Figures 2b**, **2c**, and **2d** show three cases of using the presented device.

Figure 2b shows a case of normal operation. You can see the results of comparing a reference signal (Trace C) and a signal of IC_{1B}'s output (Trace D). The signal of IC_{1B}'s output (Trace B) is low when no card appears. When the card enters the zone of vision of a sensor (Trace B), it is a sequence of normalized pulses. The output of the device at Pin 9 of IC_{5B} (Trace D) changes its level from low to high after four cycles of both signals, but it will immediately change to low if you remove the card.

Figure 2c shows operation of the device under strong IR noise. The signal of IC_{1B}'s output (Trace B) contains some high-frequency signals if a card isn't present and is a sequence of normalized pulses when a card is present. The output of the device at Pin 9 of IC_{5B} (Trace D) indicates the presence of a card by changing its level from low to high after four cycles of these signals. It immediately changes to low if you remove the card from the zone.

Figure 2d shows operation of the device under ambient direct lighting. You can see the results of comparing signals. In this case, the signal at IC_{1B}'s output (Trace B) is constant high when a card isn't present. When the card enters a sensor's zone of vision (Trace B), the signal is a sequence of normalized pulses. The output of the device at Pin 9 of IC_{5B} (Trace D) indicates this condition by changing its level from low to high after four cycles of both signals. It immediately changes from high to low when you remove the card from the zone.

Capacitors C_1, C_2, and C_3 are optional. They protect input circuits from electromagnetic noise when, for example, long wires connect the sensors and the device. Capacitors C_9, C_{10}, and C_{11} provide performance reliability by protecting the counters from short pulses. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Simple battery-status indicator uses two LEDs

Abhijeet Deshpande, People's Education Society Institute of Technology, Bangalore, India

Properly maintained rechargeable batteries can provide good service and long life. Maintenance involves regular monitoring of battery voltage. The circuit in **Figure 1** works in most rechargeable batteries. It comprises a reference LED, LED_{REF}, which operates at a constant current of 1 mA and provides reference light of constant intensity regardless of battery voltage. It accomplishes this task by connecting resistor R_1 in series with the diode. Therefore, even if the battery voltage changes from a charged state to a discharged state, the change in current is only 10%. Thus, the intensity of LED_{REF} remains constant for a battery state from a fully charged state to a fully discharged state.

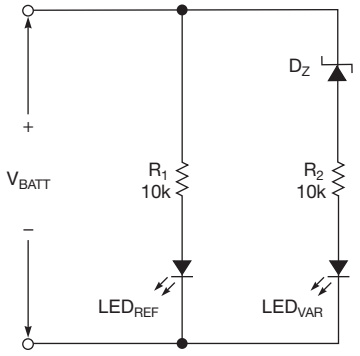


Figure 1 This circuit works in most rechargeable batteries. It comprises a reference LED, LED_{REF}, which operates at a constant current of 1 mA and provides reference light of constant intensity regardless of battery voltage.

The light output of the variable LED changes with respect to changes in battery voltage. The side-by-side-mounted LEDs let you easily compare light intensities and, thus, battery status. Using diffused LEDs as crystal-clear LEDs can damage your eyes. Instead, mount the LEDs with sufficient optical isolation so that the light from one LED does not affect the intensity of the other LEDs.

The variable LED operates from 10 mA to less than 1 mA as the battery voltage changes from fully charged to fully discharged. Zener diode D_Z in series with resistor R_2 causes the current to change with battery voltage. The sum of the zener voltage and the drop across the LED should be slightly less than the lowest battery voltage. This voltage appears across R_2. As the battery voltage varies, it produces a large variation of current in R_2. If the voltage is approximately 1V, then 10 mA will flow through LED_{VAR}, which is much brighter than LED_{REF}. If the voltage is less than 0.1V, then the light intensity of LED_{VAR} will be less than LED_{REF}, indicating that the battery has discharged.

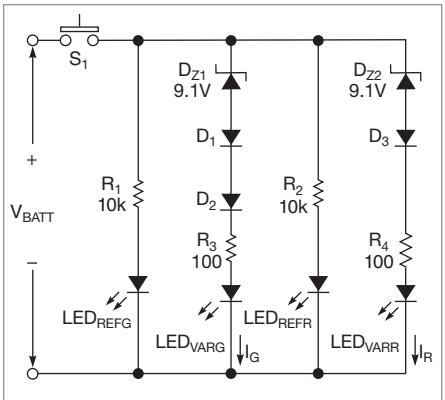


Figure 2 This circuit can withstand 13V because it has a 10-mA margin. If the LEDs are bright, quickly release pushbutton switch S_1.

Immediately after the battery has charged, the battery voltage is more than 13V. The circuit can withstand this voltage because it has a 10-mA margin. If the LEDs are bright, quickly release pushbutton switch S_1 to avoid damage to the LEDs (**Figure 2**).

The **figure** uses a 12V lead-acid battery indicator as an example, but you

TABLE 1 LED INTENSITY

Light output of LED_{VARG}	Light output of LED_{VARR}	Battery status (%)
Much brighter than LED_{REFG}	Much brighter than LED_{REFR}	70 to 100
Equally as bright as LED_{REFG}	Much brighter than LED_{REFR}	60
Off	Brighter than LED_{REFR}	50 to 30
Off	Equally as bright as LED_{REFR}	20
Off	Off	0 to 10

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

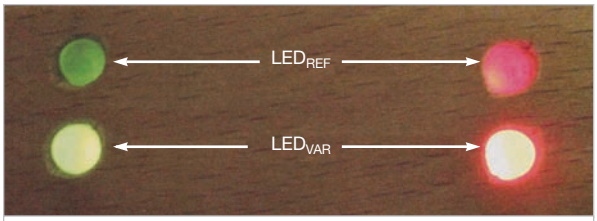


Figure 3 Both variable-intensity LEDs are brighter than the reference LEDs, indicating that the battery is 100% charged.

can extend the design to accommodate other types of chargeable batteries. You can also use it for voltage monitoring. It uses two green LEDs to indicate whether the battery has charged above 60%. A set of red LEDs indicates whether the battery charge drops below 20%. LED_{REFG} and LED_{REFR} feed through 10-kΩ resistors R_1 and R_2. For the variable-intensity LEDs, a zener diode works in series with 100Ω resistors R_3 and R_4. Diodes D_1, D_2, and D_3 provide the required clamping voltages. **Table 1** shows how LED intensity indicates battery charge.

The following **equation** calculates the variable intensity for the green LED: $V_{BATT}=I_G\times100+V_{D1}+V_{D2}+V_{LEDG}+V_{DZ1}$. For a green-LED current of 1 mA, $V_{BATT}=10^{-3}\times100+0.6+0.6+1.85+9.1=12.25V$. The selected LEDs have a drop of 1.85V at 1 mA.

If the LED has different characteristics, then you must recalculate the resistor values. At this voltage, the LEDs have the same intensity, and the battery is 60% charged. See **Reference 1** for lead-acid-battery voltages.

The following **equation** calculates the variable intensity for the red LED: $V_{BATT}=I_R\times100+V_{D3}+V_{LEDR}+V_{ZD2}$. For a green-LED current of 1 mA, $V_{BATT}=10^{-3}\times100+0.6+1.85+9.1=11.65V$.

YOU CAN USE THE DESIGN TO ACCOMMODATE OTHER TYPES OF CHARGEABLE BATTERIES. YOU CAN ALSO USE IT FOR VOLTAGE MONITORING.

At this voltage, both red LEDs have equal intensities, and the battery is 20% charged. LED_{VARG} is off. **Figure 3** shows that both variable-intensity LEDs are brighter than the reference LEDs, indicating that the battery is 100% charged. EDN

REFERENCE

1 "Lead Acid Battery Charging," *Solar Navigator*, 2005, www.solarnavigator.net/battery_charging.htm.

Hardware watchdog timer accepts range of frequencies

Robert Most, Ferris State University, Big Rapids, MI

Microcontrollers usually require a watchdog timer to bring high currents or high voltages into a safe condition. Many microcontrollers have built-in watchdog timers for this purpose. You may, however, prefer that an external circuit be the judge. An external watchdog circuit checks the integrity of the microcontroller's code, bringing the outputs to a safe state when it judges that the microcontroller's firmware has gone awry. Conditions such as errant code or failed hardware can cause these issues.

You have many options for designing an interface between your external timer and the microcontroller. For example, you can use a port pin, which yields a flexible and adaptable way to couple with almost any microcontroller.

In applications that require energizing an output or series of outputs, you need an enable signal. You can use this

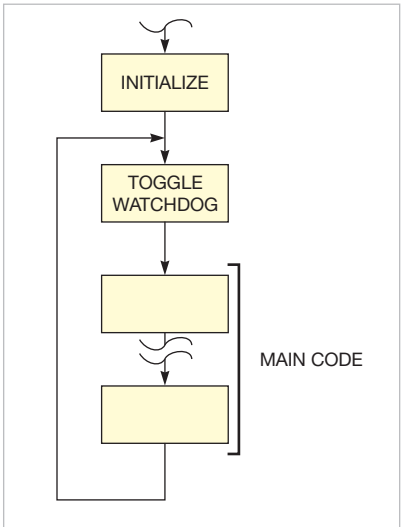


Figure 1 A microcontroller must constantly generate a pulse to keep a watchdog timer from timing out.

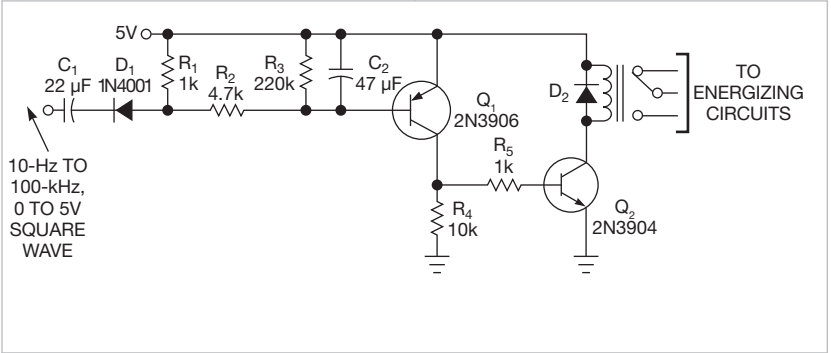


Figure 2 Capacitor C_2 remains charged enough to keep the circuit energized as long as a pulse stream of at least 10 Hz is present. If the pulse drops, the circuit removes power through the relay.

enable signal to energize a relay providing power to an output device. Turning the enable signal into a watchdog signal is therefore a wash in terms of port-pin usage. The implementation of this watchdog circuit requires the output port pin to toggle, rather than stay at a constant state.

Most microcontroller code has a main loop that is always either performing a task or calling larger functions and interacting with interrupt-service routines. If an errant task, unforeseen bug, or unintentional vector executes, the main loop either gracefully reinitializes or becomes stuck indefinitely. Either situation breaks the execution of the main loop. When that situation occurs, the timer needs to remove power from your circuits.

You can implement the watchdog timer by toggling a port output whenever the main loop runs, provided that the main loop executes 10 to 100,000 times/sec. **Figure 1** demonstrates this concept. You can implement this watchdog philosophy in several ways. You must allow for variability in the rate of the toggle signal because extraneous interrupts and other nondeterministic events may cause variability in the main loop's execution loop time. If

THE IMPLEMENTATION OF THIS WATCHDOG CIRCUIT REQUIRES THE OUTPUT PORT PIN TO TOGGLE, RATHER THAN STAY AT A CONSTANT STATE.

the watchdog circuit is not sufficiently forgiving, it may lead to false triggers, defeating its purpose. The recovery time of the watchdog timer is the maximum time between toggling events in duration. This scenario can happen when the system is in a recovery, or "limp-home," mode. This circuit can accommodate the recovery duration, but, if it deems necessary, the recovery operation can disable the watchdog timer's output. You can download **Listing 1**, a document containing sample code, from the Web version of this Design Idea at www.edn/100812dia.

The circuit in **Figure 2** uses two bipolar transistors. The second transistor is simply a relay driver. The circuit works by removing the dc component from the incoming toggling square wave, rectifying and creating an average dc value. This wave feeds transistor Q_1, biased such that a prolonged absence of pulses turns it off, thus turning off Q_2 and the relay. Changing the value of C_2 also changes how quickly the watchdog timer reacts to an absence of pulses. The circuit accepts toggling frequencies of 10 Hz to 100 kHz.

Changing the value of capacitor C_2 changes the range of compliance frequencies that the main loop generates. The circuit also functions with a 3.3V processor, but either the relay must have its own 5V supply or you must use a lower-voltage relay. EDN

Get four colors from 2 bits

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

Three-color LEDs contain red, green, and blue LEDs in one package. Using two digital control signals, you can drive these LEDs to produce four colors. The circuit in **Figure 1** uses an Analog Devices (www.analog.com) ADG854 dual analog 1-to-2 demultiplexer that lets you select the current through each LED.

The circuit uses a distinct current, I or 2I, to drive each LED. The demultiplexers determine the routes of the currents through transistors Q_1, Q_2, and Q_3 in transistor array IC_2 to the LEDs. These transistors act as both current sources and summing elements.

The following **equation** yields the value of the current: $I=(V_{REF}-V_{BE})/R_E$, where V_{BE} is the base-emitter voltage of bipolar transistors Q_1, Q_2, and Q_3. The base-emitter-voltage value varies slightly depending on the total collector current, but you can neglect this variation. Refer to the data sheet of your transistor array for this information.

One unit of current constantly flows through the green LED. Demultiplexer D_1 routes another unit of current to either the red LED or the blue LED, and D_2 routes the third unit of current to either the green LED (2I total) or the red LED.

Table 1 shows the states and colors that this circuit produces. The sum of currents flowing through all LEDs is 3I at one time for all four combinations of control variables. Thus, the generated light is approximately of the same intensity regardless of color.

The decreasing value of the base-emitter voltage with temperature, which is approximately –1.42 mV/°C, causes an increase in current through the LEDs by approximately 0.33%/°C. It has a ben-

THE DEMULTIPLEXERS DETERMINE THE ROUTES OF THE CURRENTS THROUGH TRANSISTORS Q_1, Q_2, AND Q_3 IN TRANSISTOR ARRAY IC_2 TO THE LEDS.

TABLE 1 DISTRIBUTION OF CURRENT AND COLORS

IN_1	IN_2	I_R	I_G	I_B	Color
0	0	I	I	I	White
0	1	Off	2I	I	Aqua
1	0	2I	I	Off	Red-orange
1	1	I	2I	Off	Yellow

eficial effect because it compensates for the decreasing radiance of the LEDs as temperature increases.

Drops in radiance are approximately –0.27%/°C for the blue LED and about –0.35%/°C for the green LED. The radiance of these two LEDs, which are both indium-gallium-nitride types, thus remains almost constant over ambient temperature. The red LED is an aluminium-indium-gallium-phosphorus type, having a radiance drop of approximately –0.77%/°C, and the current source roughly halves this drop.

The R_0 resistors force the logic inputs to logic zero at manual control by connecting or not connecting the IN_1 and IN_2 control leads to V_{DD}, the power-supply voltage. The maximum current flowing through the LEDs, about 26 mA, is far below the nominal current of 350 mA that Avago Technologies (www.avagotech.com) rates for the ASMT-MT00 power RGB (red/green/blue) LED that this circuit uses.

The radiance is sufficient, yet the junction temperature of the LEDs is low. Junction-to-pin thermal resistance for the green LED is 20°C/W. IC_1 dissipates approximately 0.1W. Therefore, you can estimate the junction temperature to be higher than the ambient temperature by less than 2°C (**Reference 1**). Consequently, you increase the LED's expected lifetime well beyond thousands of hours. EDN

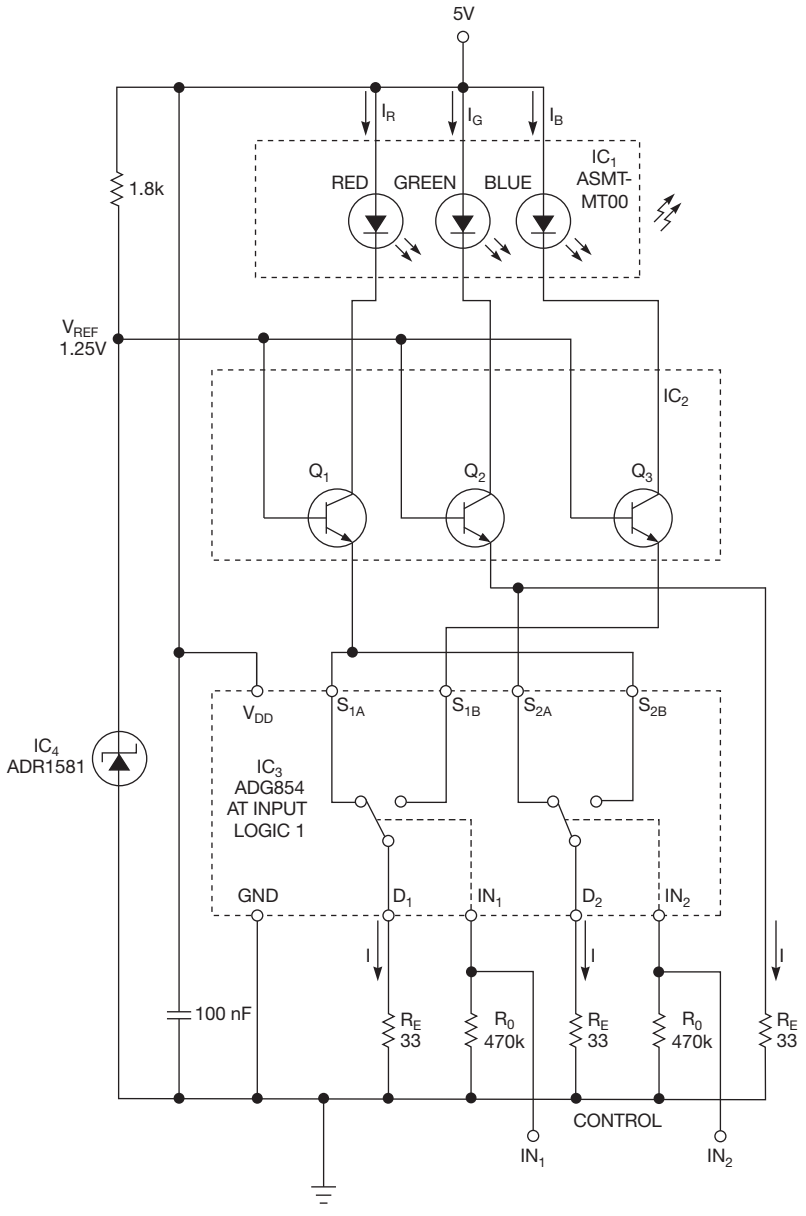


Figure 1 A three-color LED IC emits mixtures of two or three spectrally "pure" colors. The human eye perceives the mixtures as special colors.

REFERENCE

1 Oon, Siang Ling, "The Latest LED Technology Improvement in Thermal Characteristics and Reliability: Avago's Moonstone 3-in-1 RGB High Power LED," White Paper AV02-1752EN, Avago Technologies, Jan 20, 2009, www.avagotech.com/docs/AV02-1752EN.

Implement trapezoidal velocity profiles in software

Adil Ansari, General Electric, Energy Division, Atlanta, GA

Trapezoidal velocity profiles provide smooth motion for starting and stopping motor-control systems. **Figure 1** shows a velocity-profile section to be implemented in software that you can use to provide digital or analog control to a motor. In the **figure**, $\dot{\theta}$ represents the desired motor velocity (trapezoidal velocity profile) and $\dot{\theta}_{MAX}$ represents the maximum motor velocity.

You predetermine the velocity profile based on the load acceleration and deceleration requirements. A trapezoidal velocity profile has three regions: accel-

eration (Interval 1), constant velocity (Interval 2), and deceleration (Interval 3). **Equation 1** shows the calculation of the desired position, θ_D, or the distance the motor needs to travel, as the area under the curve of **Figure 1**, which is an integral of the velocity.

$$\theta_D = \underbrace{\int_{T1}^{T2} \frac{\dot{\theta}_{MAX} \times t}{(T2-T1)}\,dt}_{\text{INTERVAL 1}} + \underbrace{\int_{T2}^{T3} \dot{\theta}_{MAX}\,dt}_{\text{INTERVAL 2}} + \underbrace{\int_{T3}^{T4} \frac{\dot{\theta}_{MAX} \times t}{(T4-T3)}\,dt}_{\text{INTERVAL 3}}, \quad (1)$$

TRAPEZOIDAL VELOCITY PROFILES PROVIDE SMOOTH MOTION FOR STARTING AND STOPPING MOTOR-CONTROL SYSTEMS.

In **Equation 1**, each integral term represents the areas under the curve for regions 1, 2, and 3, respectively. To implement the velocity profile in a position-control loop, the software calculates the position input at every sampling period, T, by a point-by-point numerical integration of $\dot{\theta}$ (**Listing 1**). You can download the code for a Matlab simulation of this algorithm from the online version of this Design Idea at www.edn.com/100812dib.

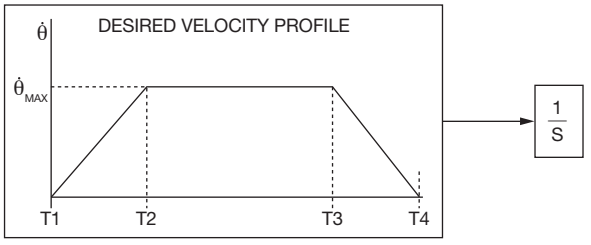


Figure 1 You can implement this velocity-profile section in software that you can use to provide digital or analog motor control.

Equations 2, **3**, and **4** give the value of $\dot{\theta}$ for intervals 1, 2, and 3, respectively:

$$\text{INTERVAL 1: } \dot{\theta} = \frac{\dot{\theta}_{MAX} \times t}{(T2-T1)} t \quad \text{FOR } t = T1 \text{ TO } T2. \quad (2)$$

$$\text{INTERVAL 2: } \dot{\theta} = \dot{\theta}_{MAX} \quad \text{FOR } t = T2 \text{ TO } T3. \quad (3)$$

$$\text{INTERVAL 3: } \dot{\theta} = \frac{\dot{\theta}_{MAX}}{(T4-T3)} t \quad \text{FOR } t = T3 \text{ TO } T4. \quad (4)$$

Next, calculate the position input to the control loop θ_i every sampling period for each of the intervals by numerically integrating $\dot{\theta}$ in **equations 2**, **3**, and **4**:

$$\text{INTERVAL 1: } \theta_i = \frac{\dot{\theta}_{MAX}}{2(T2-T1)} \times t^2 \,|\, (t = iT,\ i = 0..(T2-T1)). \quad (5)$$

$$\text{INTERVAL 2: } \theta_i = \dot{\theta}_{MAX} \times (t-T2) \,|\, (t = iT,\ i = 0..(T3-T2)). \quad (6)$$

For Interval 3, you calculate the area under the curve (shaded region in **Figure 2**) as t goes from T3 to T4 by subtracting the area of the triangle for the interval [t .. T4] from the area of the entire triangle for the interval [T3 .. T4]:

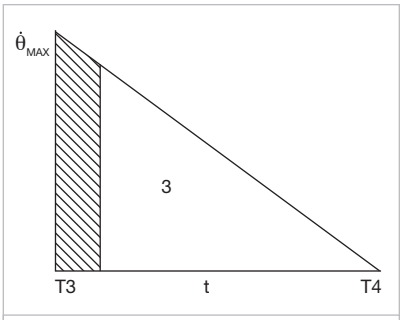


Figure 2 You use subtraction to calculate the area under the curve (shaded region) as t goes from T3 to T4.

$$\text{INTERVAL 3: } \theta_i = \left[\frac{(T4-T3)\,\dot{\theta}_{MAX}}{2} - \frac{\theta_{MAX}}{2(T4-T3)} \times (T4-T3-t)^2 \,|\, (t = iT,\ i = 0..t..(T4-T3)) \right]. \quad (7)$$

To simplify the implementation, you can define constants A, B, C, and D (D equals the area under the curve of Interval 3) such that

$$A = \frac{\dot{\theta}_{MAX}}{2(T2-T1)};\ B = \dot{\theta}_{MAX};\ C = \frac{\dot{\theta}_{MAX}}{2(T4-T3)};\ D = \frac{T4-T3}{2} \times \dot{\theta}_{MAX}. \quad (8)$$

$$\theta_i = A \times t^2\big|_{(t = iT,\ i = 0..(T2-T1))} + B \times t\big|_{(t = iT,\ i = 0..(T3-T2)} + (D - C \times (T4 - T3 - t)^2)\big|_{(t = iT,\ i = 0..(T4-T3))}. \quad (9)$$

You can then express θ_i as **Equation 9** (above) shows. EDN

LISTING 1 PSEUDO CODE

```
// Interval I
// T is sampling interval

    Tint1 = T2 – T1  // Interval I
          For  t = 0 .. Tint1
                        {
                           θi =A.t²
                            t = t + T
                        }

// Interval II

    Tint2 = T3 – T2  // Interval II
          For  t = 0 .. Tint2
                       {
                            θi = θi +B.t
                             t = t + T
                       }

// Interval III

    Tint3 = T4 – T3  // Interval III
          For  t = 0 .. Tint3
          {
            θi = θi + D – C.(Tint3 – t)²
             t = t + T
          }
```

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Control a dc motor with your PC

Firas M Ali Al-Raie,
Polytechnic Higher Institute of Yefren, Yefren, Libya

The circuit in this Design Idea controls the speed of a 5V permanent-magnet dc motor through the PC's parallel port (**Figure 1**). You use the C++ computer program, available at www.edn.com/100826dia, to run the motor at three speeds. The circuit uses PWM (pulse-width modulation) to change the average value of the voltage to the dc motor. You connect the motor to the PC's parallel port with an interface circuit. The design comprises IC_1, a 74LS244 buffer; IC_2, a ULN2003 driver; relay switches S_1, S_2, and S_3; IC_3, a 555 astable multivibrator circuit; and Q_1, a 2N2222 driving transistor. The 555 timer operates as a variable-pulse-width generator. You change the pulse width by using relays to insert or split resistors in the 555 circuit.

The computer program controls these resistors. When S_1 is on and both S_2 and S_3 are off, the timer output is set to logic one, thereby driving the motor with its maximum speed. When S_1 and S_2 are on, the 555 timer generates a pulse signal with a 50% duty cycle. In this case, the charging resistor, R_{A1}, is equal to the discharging resistor, R_B. In the third case, S_1 and S_3 are on, and the charging resistor is R_{A2},where $R_{A2}=0.1\times R_B$, reducing the on time of the pulse signal and, consequently, the speed of the motor to the lower limit. **Table 1** summarizes the on/off-operation conditions of the relays and the corresponding dc-motor speeds.

DIs Inside

▸To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

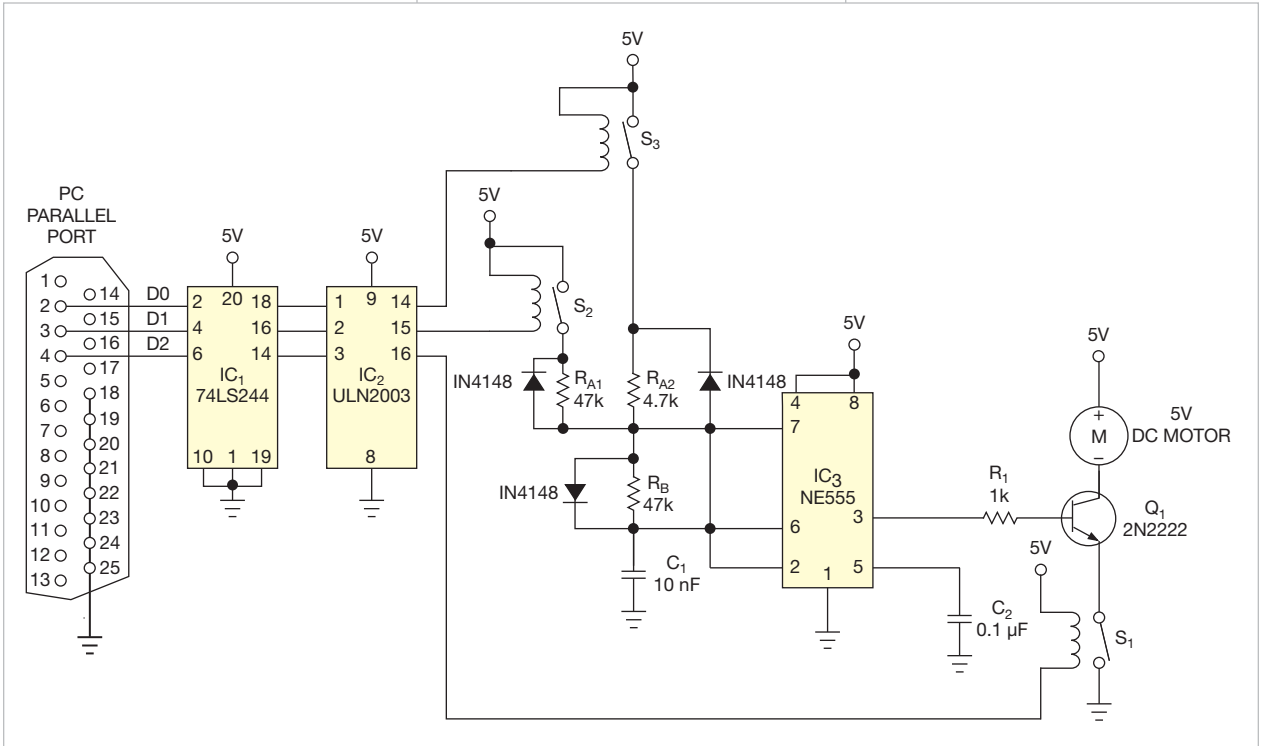


Figure 1 This circuit controls the speed of a 5V permanent-magnet motor through the PC's parallel port.

The code prompts you to select a certain speed, stores your selection as an integer variable choice, generates the proper digital sequence, and stores it at another integer variable. You place the value of the integer variable data at a PC's parallel port using the outportb function. The program uses the kbhit function to stop the motor when you hit any key on the keyboard. EDN

TABLE 1 SWITCH STATES AND GENERATED PC SEQUENCES

S_3	S_2	S_1	Equivalent digital sequence	Motor speed
Off	Off	Off	000	Stop
Off	Off	On	001	Maximum
Off	On	On	011	Medium
On	Off	On	101	Minimum

Look-up table eliminates the need for an IC

Abel Raynus, Armatron International, Malden, MA

When you need to connect a microcontroller to a seven-segment LED display, you can use a BCD (binary-coded-decimal)-to-seven-segment decoder. **Figure 1** shows a typical circuit that uses a CD4511 to translate a 4-bit code into BCD.

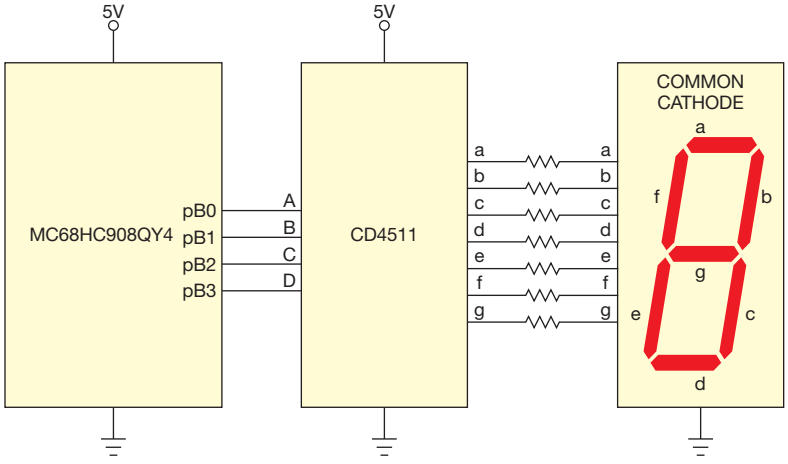


Figure 1 A CD4511 converts BCD codes to seven-segment-displays.

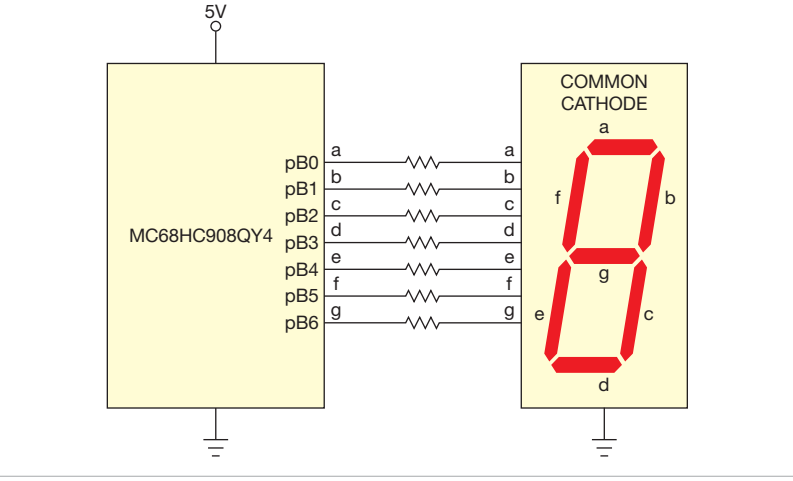


Figure 2 The look-up table lets the microcontroller produce a seven-segment output.

Unfortunately, a limited size and budget may force you to omit components whenever possible. This requirement is especially critical with consumer products. Simple firmware allows you to overcome this limitation by directly connecting the display to a microcontroller (**Figure 2**).

A recent project used the 8-bit Freescale (www.freescale.com) MC68HC908QY4 microcontroller. When you write code for a microcontroller, you often represent data in decimal, hexadecimal, or BCD formats. A numeric display needs a special seven-segment code, the value of which depends on the common point of the display LEDs—common cathode or common anode—and on microcontroller outputs you choose for display. **Tables 1** and **2** show how to obtain seven-segment code values for common-cathode and common-anode displays, respectively.

A NUMERIC DISPLAY NEEDS A SPECIAL SEVEN-SEGMENT CODE, THE VALUE OF WHICH DEPENDS ON THE COMMON POINT OF THE DISPLAY LEDs.

No mathematical connection exists between seven-segment code and any of these formats. Thus, you must use a table that a previous Design Idea

TABLE 1 CODE FOR COMMON-CATHODE DISPLAY

Decimal number	pB7 —	pB6 (g)	pB5 (f)	pB4 (e)	pB3 (d)	pB2 (c)	pB1 (b)	pB0 (a)	Seven-segment code
0	0	0	1	1	1	1	1	1	$3f
1	0	0	0	0	0	1	1	0	$06
2	0	1	0	1	1	0	1	1	$5b
3	0	1	0	0	1	1	1	1	$4f
4	0	1	1	0	0	1	1	0	$66
5	0	1	1	0	1	1	0	1	$6d
6	0	1	1	1	1	1	0	0	$7c
7	0	0	0	0	0	1	1	1	$07
8	0	1	1	1	1	1	1	1	$7f
9	0	1	1	0	1	1	1	1	$6f

TABLE 2 CODE FOR COMMON-ANODE DISPLAY

Decimal number	pB7 —	pB6 (g)	pB5 (f)	pB4 (e)	pB3 (d)	pB2 (c)	pB1 (b)	pB0 (a)	Seven-segment code
0	0	1	0	0	0	0	0	0	$40
1	0	1	1	1	1	0	0	1	$79
2	0	0	1	0	0	1	0	0	$24
3	0	0	1	1	0	0	0	0	$30
4	0	0	0	1	1	0	0	1	$19
5	0	0	0	1	0	0	1	0	$12
6	0	0	0	0	0	0	1	1	$03
7	0	1	1	1	1	0	0	0	$78
8	0	0	0	0	0	0	0	0	$00
9	0	0	0	1	0	0	0	0	$10

describes (**Reference 1**). The project described here used one seven-segment LED display to show the digits 0 through 9. A table makes the firmware simple and short.

Using the assembly-language code that you can download from the online version of this article at www.edn.com/100826dib, you need just 7 bytes to execute the program, plus 10 bytes of memory for the code table. A decimal number for conversion goes into register DECreg and acts as an index, X, applying to the code table. The result appears at the microntroller's Port B output. EDN

A DECIMAL NUMBER FOR CONVERSION GOES INTO A REGISTER AND ACTS AS AN INDEX APPLYING TO THE CODE TABLE.

REFERENCE

1 Raynus, Abel, "Tables ease microcontroller programming," *EDN*, April 22, 2010, pg 76, www.edn.com/article/457500-Tables_ease_microcontroller_programming.php.

Operate circuits at voltages as high as 540V ac

Vipin Bothra and John Lo Giudice, STMicroelectronics, Schaumburg, IL

Energy meters, HVAC (heating/ventilation/air-conditioning) systems, and high-power equipment that runs on three-phase ac inputs pose a challenge to power-supply designers because nominal input voltage can be as high as 540V ac. The challenge increases if the power supply must operate from 100 to 540V ac. Design choices are numerous, and final system costs can vary dramatically with those choices. An abundance of parts is available for power supplies with input power as high as 240V ac. High-voltage input-power supplies are, however, still a niche area for most semiconductor companies.

The power-supply circuit in **Figure 1** uses an input-chopper circuit that allows the clamping of input voltage to the flyback power stage so that the power is less than 400V. That voltage lets you use a standard design technique for the flyback stage. The input chopper provides many advantages over prevailing high-voltage-input power supplies.

Unlike a standard flyback converter, this circuit eliminates the need for a high-voltage MOSFET for the switch, thereby letting you use lower-cost, commonly available MOSFETs. Moreover, overall switching losses in the power supply decrease dramatically with the reduction in bus voltage. The circuit can use smaller and lower-cost transformers because of a reduction in creepage-clearance requirements.

This circuit's reduced bus voltage eliminates the stacked-FET-flyback topology and the need for two or more high-voltage capacitors. It also improves the overall efficiency of the system by eliminating the high losses of a stacked FET, replacing them with small losses from a bypass switch.

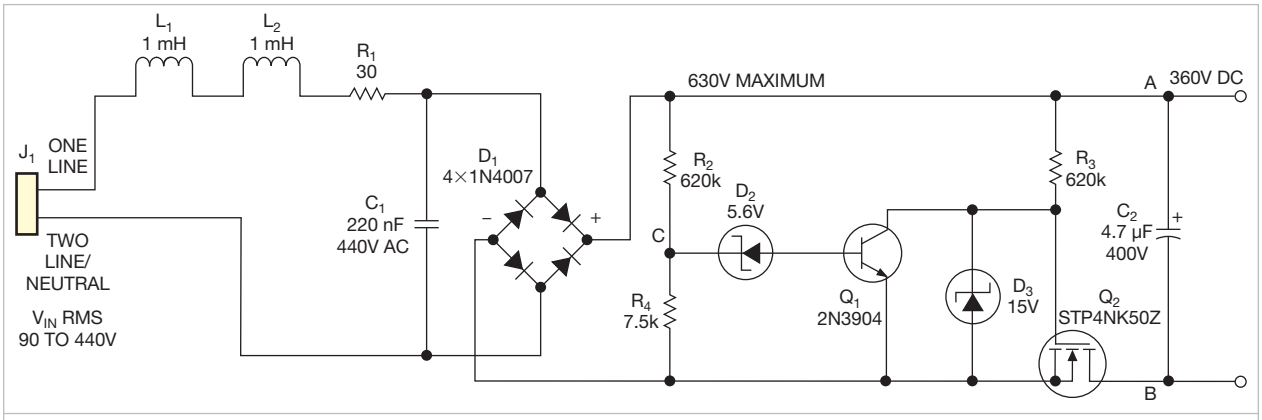


Figure 1 A chopper circuit reduces power-bus voltage to less than 400V dc per pin.

The 500V, 2.7Ω STP4NK50Z N-channel MOSFET switches at the line frequency. It turns on at a predetermined voltage, and it turns off at any higher voltage. It limits the voltage on C_2 to approximately 360V dc. When the voltage at the divider of R_2 and R_4 reaches approximately 6.3V, or 360V at the top of the divider, Q_1 turns on and steals current from the gate of Q_2, and the MOSFET turns off. The divider sets the level at which Q_2 switches. All resistors are 0.25W except for R_1, which can be 2W to survive surge. The circuit underwent testing with 12W of output power at 90 to 440V-ac input. The maximum input current to the power supply depends on the thermal performance of Q_2.EDN

THE 500V MOSFET TURNS ON AT A PREDETERMINED VOLTAGE; IT TURNS OFF AT ANY HIGHER VOLTAGE.

Microcontroller supervises 0- to 20-mA protection circuit

Anatoly Andrusevich, Maxim Integrated Products, Moscow, Russia

The 0- to 20-mA current loop is a reliable means of data communication in industrial applications. These circuits use a precision shunt in the receiver to convert the current signal into a voltage signal. Accidentally connecting the precision shunts to the current-loop power supply can cause damage, after which you must replace the shunt and recalibrate the system. To avoid that expense, you can use a microcontroller-controlled protection circuit (**Figure 1**).

With conventional techniques, you protect the shunt with a fast fuse or by turning off the loop with an automatic switch, which then turns back on after a specified period. The circuit in **Figure 1** provides protection that is much faster than a fuse. IC_1, the slowest device in the circuit, switches off in less than 500 µsec. It offers a higher-precision switching threshold than a fuse, and, of course, there's no fuse to replace. Rather than making you cycle power to restore the loop, the microcontroller provides control of the protection circuit. The microcontroller also logs the event, thereby providing a record that the system invoked the protection circuit.

The protection circuit has virtually

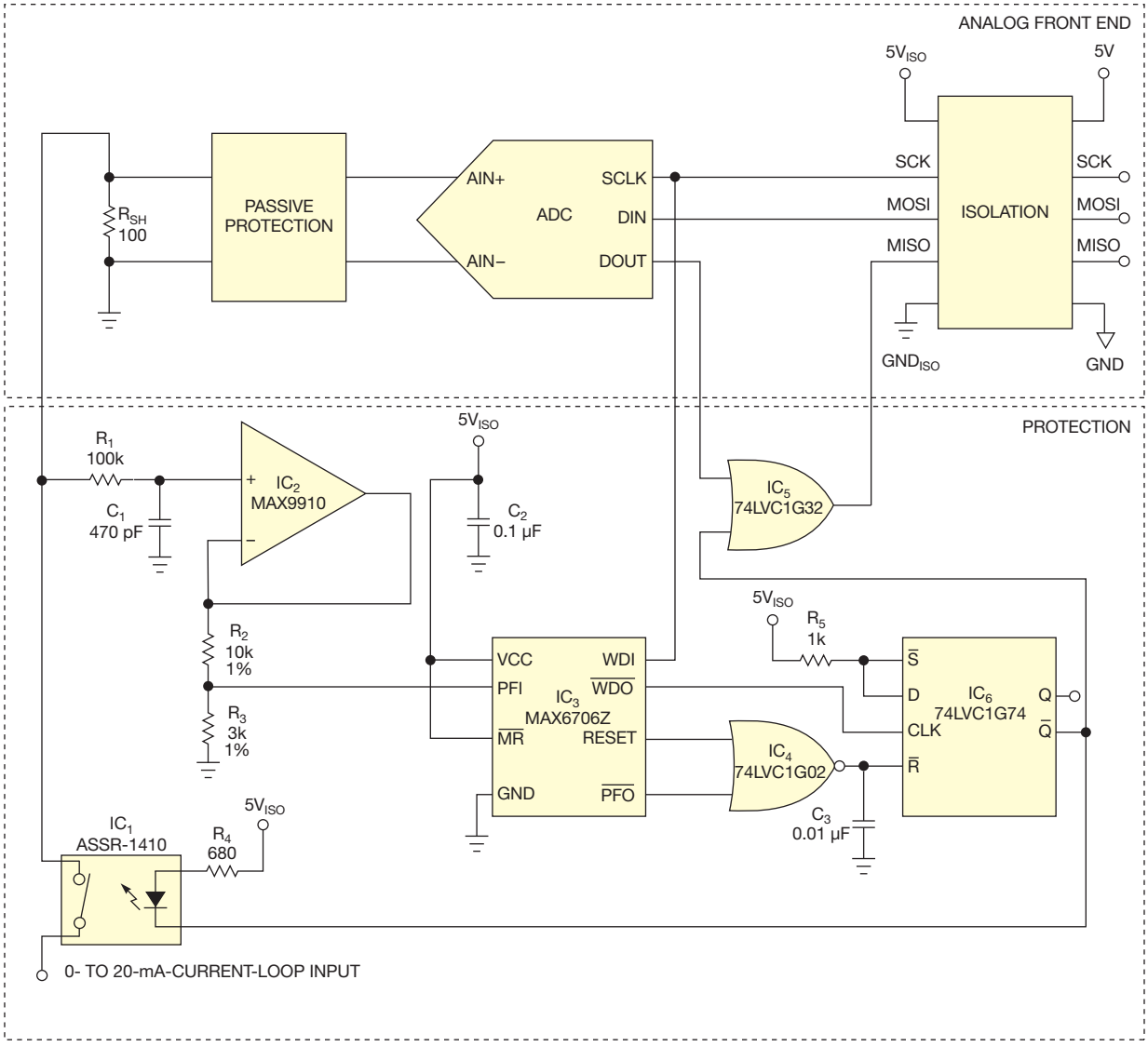


Figure 1 This circuit protects the isolated analog front end of a current loop. For simplicity, the drawing omits power and ground connections for IC_2, IC_4, IC_5, and IC_6.

no effect on the analog front end. The IC_2 buffer ensures an input current of less than 30 pA. The on-resistance of IC_1 is less than 2Ω. The circuit needs no additional isolated data channels or microcontroller-I/O ports, and it prevents damage during system installation or repair. It also turns off the loop after power-up and when no power is available.

You implement the protection algorithm with a power-fail comparator and a watchdog circuit, available as separate outputs on IC_3, together with IC_6, a D-type flip-flop.

At power-up, the flip-flop is in the reset state, and the current loop is open, due to a high-level reset signal from IC_3 driving IC_4, a NOR gate. After the first low-to-high transition on the SCK (clock-signal) line, a rising edge from IC_3's $\overline{WDO}$ (watchdog output) sets the flip-flop and pulls current through the solid-state relay, IC_1, thus connecting the input to the loop.

THE CIRCUIT NEEDS NO ADDITIONAL ISOLATED DATA CHANNELS, AND IT PREVENTS DAMAGE DURING SYSTEM INSTALLATION.

In the event of a loop-current overload greater than 27 mA, a high level from the $\overline{PFO}$ (power-fail-output) comparator on IC_3 resets the flip-flop and switches off IC_1. Thanks to the IC_5 gate, the microcontroller inputs ones at the MISO (master input/slave output), meaning overcurrent.

To again switch on the loop, the microcontroller must stop the SCK line for at least 2.4 sec. The next low-to-high transition on SCK then reconnects the current loop. EDN

LED indicates power source

Brian Conley, Circuitsville Engineering LLC, Beaverton, OR

LED circuits with current-limiting resistors find extensive use as power indicators and for debugging circuits (**Reference 1**). In some cases, however, your design may require a different approach. Bipolar transistors have a little-discussed behavior: reverse active region. For low voltages and small currents, an NPN transistor can operate in reverse with a significantly lower gain, which can be undesirable. Some linear regulators also operate in this way.

THE LED ILLUMINATES WHEN THE BOARD RECEIVES VOLTAGE FROM THE WALL WART, BUT NOT FROM THE USB PORT.

The circuit in **Figure 1** gets its input-voltage power primarily from a wall-wart dc-power supply that can provide 7 to 12V. It may also get 5V from a USB (Universal Serial Bus) port. This design requires a circuit that indicates whether the board is receiving voltage from the wall wart or from the USB port.

The circuit uses Q_1, a 2N7002 FET, and zener diode D_1 to solve the problem. The FET is in series with LED_1 and current-limiting resistor R_1. Diode D_1 is a Vishay (www.vishay.com) AZ23C4V3-V, which has a typical reverse voltage of 4.3V within a range of 4 to 4.6V. When Q_1's gate-to-source voltage exceeds its threshold-voltage range of 1 to 2.5V, the LED turns on. The voltage coming from the USB port is insufficient to turn on LED_1 because of the voltage drop across D_1. Thus, the LED illuminates when the board receives voltage from the wall wart, but not from the USB port.

Under testing, the LED illuminates when the input voltage is at least 7.1V. When it is below that voltage, the LED is off, indicating that the USB port is powering the circuit.

Resistor R_3 comprises two 1-kΩ resistors in parallel. This setup is necessary because the input voltage is 12V and the zener diode's minimum voltage is 4V. A voltage of 8V appears across R_3, producing 0.128W—too much power for one resistor in a 0805 package. EDN

REFERENCE

1 Conley, Brian, "Go on green," *EDN*, June 24, 2010, http://www.edn.com/article/509479-Go_on_green.php.

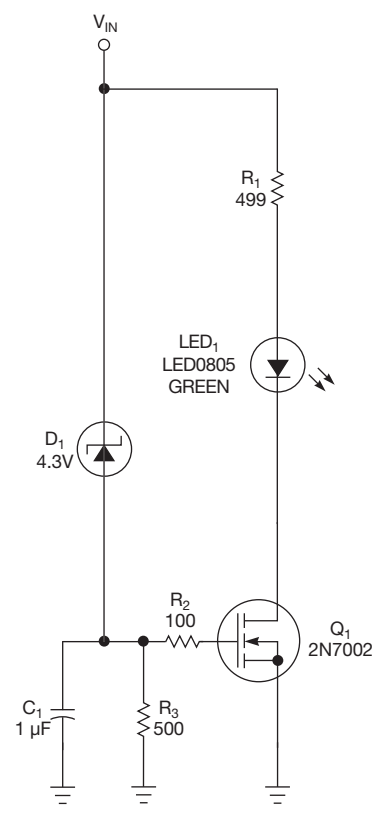


Figure 1 This circuit gets its power from a wall-wart dc-power supply or from a USB port. The LED illuminates when the board receives its voltage from the wall wart. When it is off, the USB port is powering the circuit.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Current monitor compensates for errors

Chau Tran and Paul Mullins, Analog Devices, Wilmington, MA

You sometimes need to measure load currents as large as 5A in the presence of a common-mode voltage as high as 500V. To do so, you can use Analog Devices' (www.analog.com) AD8212 high-voltage current-shunt monitor to measure the voltage across a shunt resistor. You can use this circuit in high-current solenoid or motor-control applications. **Figure 1** shows the circuit, which uses an external resistor and a PNP transistor to convert the AD8212's output current into a ground-referenced output voltage proportional to the IC's differential input voltage. The PNP transistor handles most of the supply voltage, extending the common-mode-voltage range to several hundred volts.

An external resistor, R_{BIAS}, safely limits the circuit voltage to a small fraction of the supply voltage. The internal bias circuit and 5V regulator provide an output voltage that's stable over the operating temperature range, yet it minimizes the required number of external components. Base-current compensation lets you use a low-cost PNP pass transistor, recycling its base current, I_B, and mirroring it back into the signal path to maintain system precision. The common-emitter breakdown voltage of this PNP transistor becomes the operating common-mode range of the circuit.

The internal regulator sets the voltage on COM to 5V below the power-supply voltage, so the supply voltage for the measurement circuit is also 5V. Choose a value for the bias resistor, R_{BIAS}, to allow enough current to flow to turn on and continue the operation of the regulator.

DIs Inside

▸What are your design problems and solutions? Publish them here and receive $150! Send your Design Ideas to edndesignideas@cancom.com.

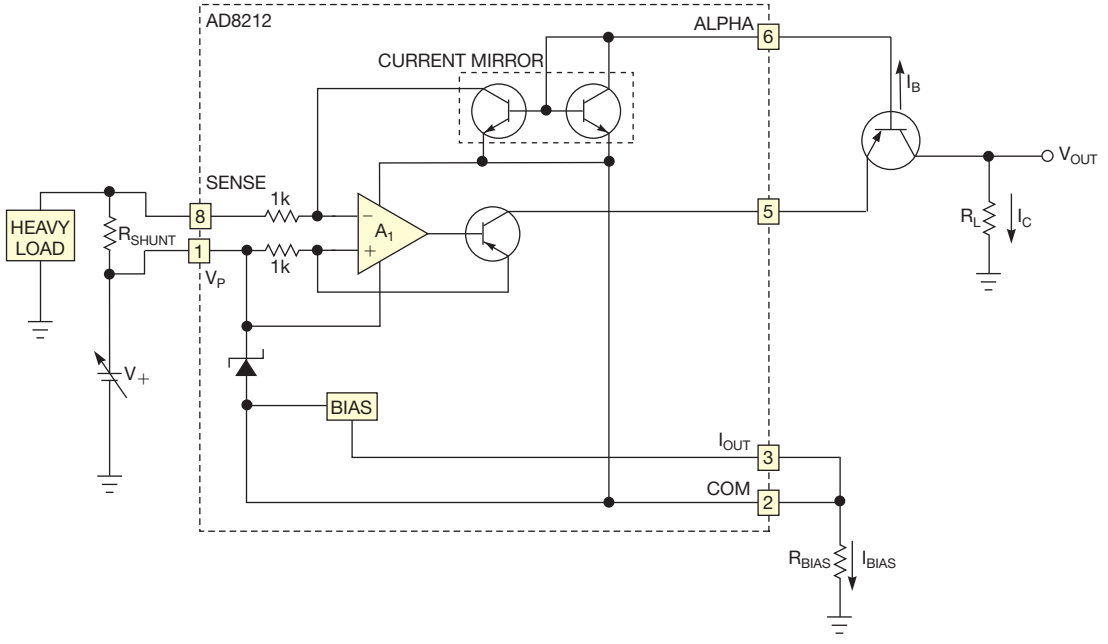


Figure 1 An external PNP transistor lets you operate the circuit at high voltages.

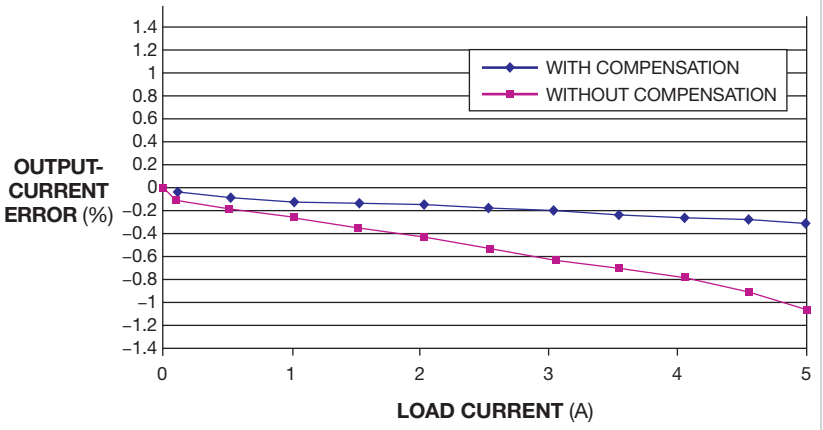


Figure 2 Internal base-current compensation reduces error.

For high-voltage operation, set I_{BIAS} at 200 µA to 1 mA. The low end ensures the turn-on of the bias circuit; the high end is limited, depending on the device you use.

With a 500V battery and an R_{BIAS} value of 1000 kΩ, for example, $I_{BIAS}=(V_+–5V)/R_{BIAS}=495V/1000\ k\Omega=495\ \mu A$.

The circuit creates a voltage on the output current approximately equal to the voltage on COM plus two times the V_{BE} (base-to-emitter voltage), or $V_+–5V+2V_{BE}$. The external PNP transistor withstands two times the base-to-emitter voltage of more than 495V, and all the internal transistors withstand voltages of less than 5V, well below their breakdown capability.

Current loss through the base of the PNP transistor reduces the output current of the AD8212 to form the collector current, I_C. This reduction leads to an error in the output voltage. You can use a FET in place of the PNP transistor, eliminating the base-current error but increasing the cost. This circuit uses base-current compensation, allowing use of a low-cost PNP transistor and maintaining circuit accuracy. In this case, current-mirror transistors, the AD8212's internal resistors, and amplifier A_1 combine to recycle the base current.

Figure 2 shows a plot of output-current error versus load current with and without the base-current-compensation circuit. Using the compensation circuit reduces the total error from 1 to 0.4%. You should choose the gain of the load resistor, R_L, to match the input voltage range of an ADC. With a 500-mV maximum differential-input voltage, the maximum output current would be 500 µA. With a load resistance of 10 kΩ, the ADC would see a maximum output voltage of 5V.EDN

Buck regulator handles light loads

Justin Larson and Frank Kolanko, On Semiconductor, East Greenwich, RI

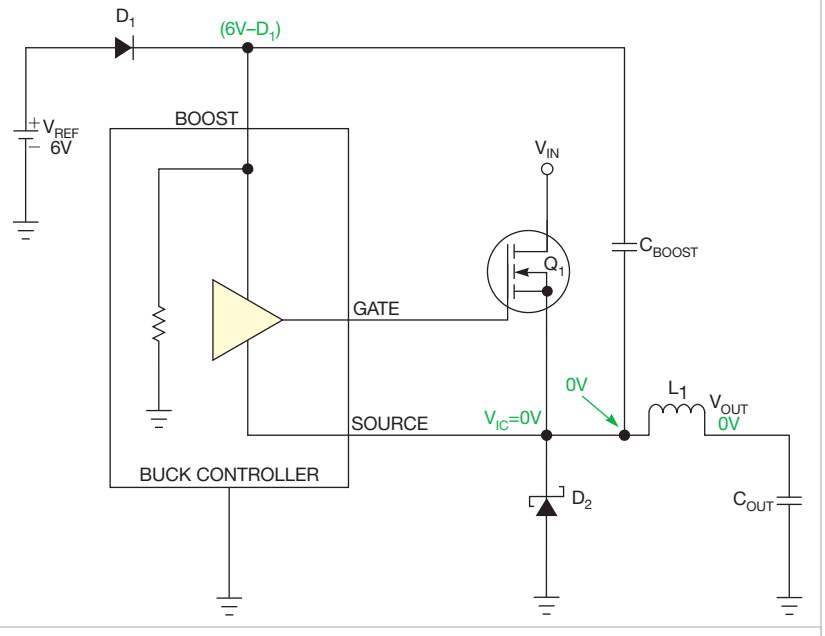


Figure 1 Many buck-controller ICs use a floating-gate driver.

Buck regulators operating in CCM (continuous-conduction mode) have straightforward operation, allowing for easy calculation of output voltage and system design. However, lightly loaded buck regulators operate in DCM (discontinuous-conduction mode), and their operation is more complicated. The duty cycle changes from a ratio of the output voltage and the input voltage. A regulator that reduces a 12V input to 6V has a 50% duty cycle. When the regulator is too lightly loaded to keep some current continuously flowing in the inductor, it enters DCM. The duty cycle changes to a complex function of inductor value, input voltage, switching frequency, and output current, greatly slowing the control-loop response.

Many buck-controller ICs use a floating-gate driver (**Figure 1**). You use a separate supply reference voltage, V_{REF}, for high efficiency. During start-up, it powers the NFET gate driver to a diode drop below the reference voltage. Sufficient voltage is available to drive the

gate of the FET because the initial conditions dictate 0V on the output and on the source of FET Q_1.

During CCM, current always flows through the inductor. Q_1 or D_2 supplies this current during the flyback event that Q_1's turn-off causes (**Figure 2**). The flyback event creates a voltage at the source of Q_1, and the drop across D_2 limits this voltage, making it a negative voltage with respect to ground. Sufficient voltage is available to drive Q_1 because the C_{BOOST} capacitor boosts the gate voltage. This boost provides a high voltage to the boost pin and the resultant negative voltage on the Q_1 source.

The system enters DCM when the load drops to the point at which the average current demand is less than one-half the current ripple. Diode D_2 prevents reverse current in the inductor. Depending on the chip you use, the output may overshoot due to the slower response time of the control loop. The regulator may also miss pulses and generally operate unpredictably. After Q_1 turns off, C_{BOOST} starts to bleed down through the boost pin and D_1 (**Figure 3**). The extended off time of Q_1 in DCM starts to discharge the C_{BOOST} capacitor. At approximately 3V across C_{BOOST}, Q_1 does not turn on until the output capacitor, C_{OUT}, discharges adequately to provide a lower voltage on the source of Q_1 than that of the boost pin through D_1. This behavior is unacceptable in a voltage regulator.

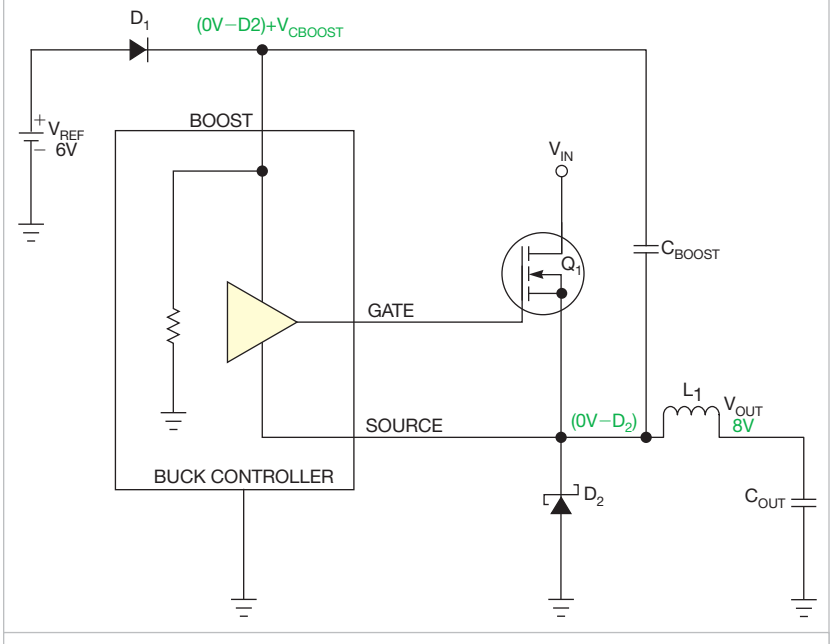


Figure 2 During CCM, current always flows through the inductor. Q_1 or D_2 supplies this current during the flyback event that Q_1's turn-off causes.

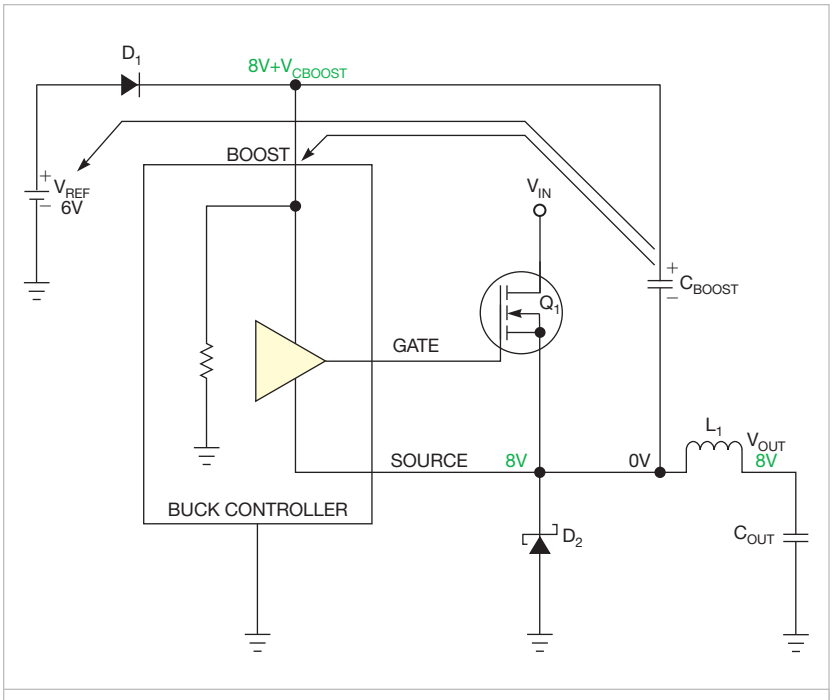


Figure 3 The C_{BOOST} capacitor discharges when the regulator goes into DCM.

YOU DON'T KNOW THE TEMPERATURE COEFFICIENT OF THE CURRENT INTO THE BOOST PIN, SO YOU SHOULD ALSO CHECK OPERATION AT LOW TEMPEATURE.

High temperatures create a situation with higher leakage currents. You don't know the temperature coefficient of the current into the boost pin, so you should also check operation at low temperature. Evaluate the system to determine the lowest capacitor value, using this result in your worst-case evaluation simulations. You can thus ensure that the design will operate in DCM by increasing the value of C_{BOOST}. You could also increase the reference voltage to which D_1 connects. You may want to consider replacing D_1 with a low-leakage Schottky diode. If none of these approaches results in reliable operation, you can switch to an IC that uses a gate driver referenced to ground or modify your design to use a synchronous-buck architecture. EDN

Sense multiple pushbuttons using only two wires

Bernhard Linke, Maxim Integrated Products Inc, Dallas, TX

Keyboards and numeric keypads often provide the user interface for electronic equipment, but many applications require only a few pushbuttons. For those applications, you can monitor multiple pushbuttons over a single pair of wires (**Figure 1**).

The multichannel 1-Wire addressable switch, IC_1, provides PIO (input/output ports) P0 through P7, which in this application serve as inputs. The 1-MΩ R_{PD} resistors connect these ports to ground to ensure a defined logic-zero state. Diode/capacitor combination D_1/C_1 forms a local power supply that steals energy from the 1-Wire communication line. Pressing a pushbutton connects the corresponding port to the local supply voltage, which is equivalent to logic one. This change of state sets the port's activity latch (**Reference 1**).

As a 1-Wire slave device, IC_1 doesn't initiate communications. Instead, the master device—typically, a microcontroller—polls the 1-Wire line. To minimize overhead, IC_1 supports conditional search, a 1-Wire network function. Before using that function, however, you must configure IC_1 according to the needs of the application. That configuration includes channel selection, which defines the qualifying input ports; channel-polarity selection, which specifies the polarity of the qualifying ports; choosing a pin or an activity latch for the port; and specifying whether the device will respond to activity at a single port, an OR, or at all ports, an AND.

Consider, for example, that IC_1 shall respond to a conditional search if it detects activity at any of the eight ports. This search requires a channel-selection mask of 11111111b for address 008Bh. The numeral one indicates that IC_1 has selected a port. This search also requires a channel-polarity selection of 11111111b for address 008Ch, where the numeral one indicates a high level, and a control/status register setting of 00000001b for address 008Dh, which selects the port's activity latch as a source and specifies OR as the conditional search term—that is, activity at any port.

After power-up, the configuration data must be loaded into IC_1 using the write-conditional-search-register command. Next, the channel-access-write command, with FFh as a PIO output-data byte, defines the ports as inputs. Subsequently, the issue of a reset-activity-latches command completes the configuration. IC_1 is now ready to handle pushbutton activity.

After you configure IC_1, the application software enters an endless loop, in which a conditional-search command follows a 1-Wire reset. With no pushbutton activity, IC_1 does not respond, as

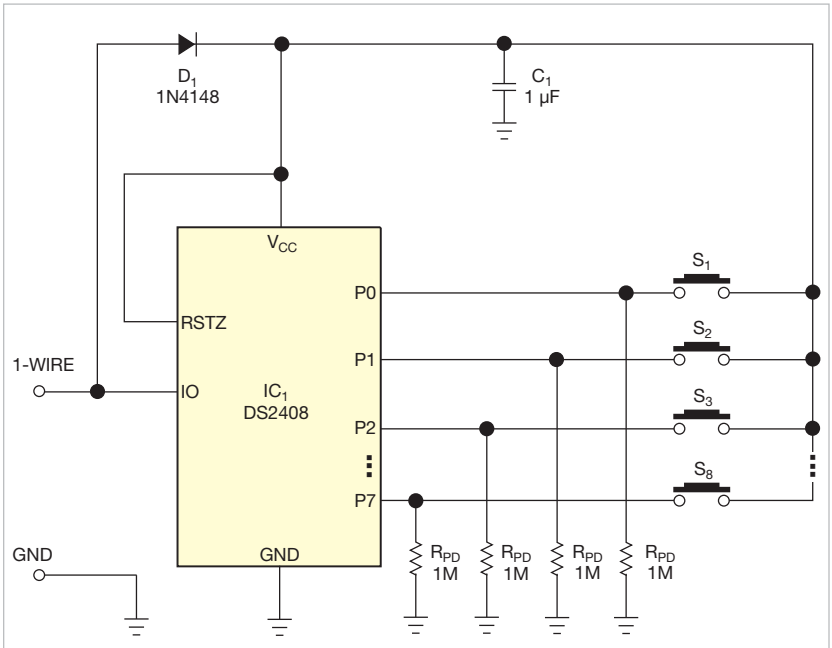


Figure 1 This circuit connects to a microcontroller and can monitor eight pushbuttons using only two wires.

a logic one indicates, for the 2 bits immediately after the conditional-search-command code. In that case, the microcontroller cancels the conditional search and starts over.

If IC_1 responds to the conditional search, the first 2 bits will be one and zero, representing the least-significant bit of the device's family code, 29h, in its true and inverted forms. In that case, the microcontroller should complete the conditional-search flow, which comprises a 192-bit sequence. Next, the microcontroller reads from IC_1 by issuing a read-PIO-registers command using 008Ah, the address of the PIO-activity-latch-state register. The microcontroller then issues a 1-Wire reset, a resume command, and a resume-and-reset-activity-latches command. It then returns to the endless loop, polling for the next pushbutton event.

If IC_1 responds and no other 1-Wire slave is connected, the microcontroller could cancel the conditional search after reading the first 2 bits, issue a 1-Wire reset, issue a skip-ROM command, and then read the PIO-activity-latch-state register. Next, it must issue a 1-Wire reset, a skip-ROM command, and a reset-activity-latches command before returning to the endless loop.

The code read from the PIO-activity-latch-state register tells which button was pressed. If you press S_1, the data is 00000001b; if you press S_2, it is

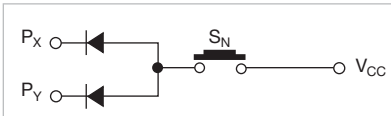


Figure 2 This circuit can monitor 28 additional pushbuttons if you use diodes to connect them to two ports.

00000010b; and so forth. At least one of the 8 bits will be one. If you press several buttons after the last reset-activity-latches command, several bits are one. The application software must then decide whether such a condition is valid. In the simplest case, one-of-eight code, the software considers all codes that have several bits at one as invalid.

You can expand this concept to more than eight pushbuttons. Instead of associating one pushbutton with one port, you can associate additional pushbuttons with two simultaneously activated ports, representing two-of-eight code (**Figure 2**). If another pushbutton activates P_X or P_Y, the diodes prevent that activity from propagating to other ports. Again, the application software must check the code it reads from the PIO-activity-latch-state register to decide whether it is valid. The theoretical limit of this concept is 255 pushbuttons, which require combinations of two, three, four, five, six, seven, or eight diodes per additional pushbutton. When the cost of diodes for each additional pushbutton begins to exceed the benefits, you will find that adding another DS2408 is more cost-effective. EDN

REFERENCE

1 "DS2408 1-Wire 8-Channel Addressable Switch," Maxim Integrated Products Inc, www.maxim-ic.com/ds2408.

Tricolor LED emits light of any color or hue

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

The human eye can see any color as a mixture of blue, red, and green. The circuit in **Figure 1** produces all three colors through an Avago (www.avagotech.com) ASMT-YTB0 tricolor LED. You can produce a wide range of colors by varying the current in the blue, red, and green LEDs.

The collector outputs of bipolar differential stages form the current sources. A classic symmetrical differential stage with two equal bipolar transistors is a backbone of almost all bipolar analog ICs. In this case, however, the differential stage is asymmetrical, with a 2-to-1 collector-current distribution instead of the common 1-to-1 ratio at 0V base-voltage difference. The circuit produces the 2-to-1 current ratio by paralleling a third equal transistor, Q_3, to Q_1. The common collector of the paralleled transistor pair connects to the common emitter of the Q_4/Q_5 differential stage. Thus, the base differential voltages equal 0V at both the stages, and collector currents I_R, I_G, and I_B are almost equal.

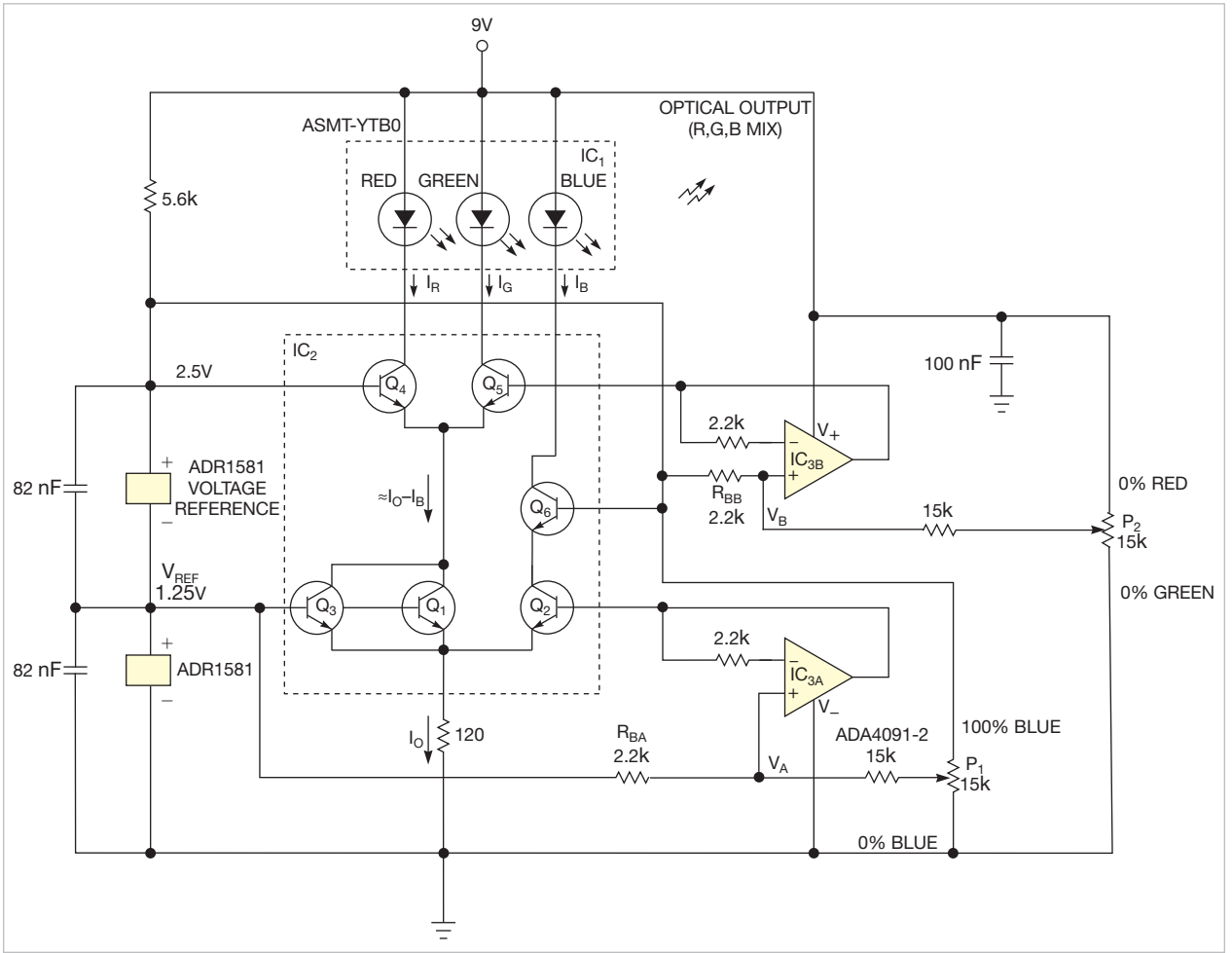


Figure 1 Potentiometers P_1 and P_2 let you control the color of emitted light.

The differential stages let you vary I_R, I_G, and I_B over a range of 0 to I_O, where $I_R+I_G+I_B \approx I_O=4.43$ mA. This value is approximate because $I_R+I_G+I_B$ is lower by a relative value of $3/\beta$, where β is a current gain of the bipolar transistors. The relative error is less than 1%. Transistor Q_6 equalizes Q_2's collector voltage with those of the Q_1 and Q_3 collectors. This approach preserves the matching of the base-emitter voltages of Q_1, Q_2, and Q_3. The base currents of bipolar transistors in this case can reach to as much as 100 µA. For this reason, you route the color and hue control voltages, V_A and V_B, which you derive from resistive potentiometers P_1 and P_2, to the bases of Q_2 and Q_5 through voltage-follower-connected op amps IC_{3A} and IC_{3B}, two halves of an Analog Devices' (www.analog.com) ADA4091-2. The ADA4091-2 has low power consumption and input offset voltage of less than 500 µV with a typical value of 80 µV.

The ADA4091-2 has a maximum input bias current of 65 nA, which causes a negligible voltage drop on resistors R_{BA} and R_{BB}. This voltage drop is less than 130 µV. You can achieve even more accuracy by inserting resistors of the same value as R_{BA} between the respective inverting inputs and outputs of both the A and the B followers. This step brings reduction of input-bias-current-caused errors to one-sixth worst case—down to 1/600.

Potentiometer P_1 controls the blue LED's intensity. At the upper-end position, when the LED is 100% blue, transistors Q_2 and Q_3 are off, which turns off Q_4 and Q_5. Thus I_O flows solely through Q_2 and Q_6. The red and green LEDs are therefore off. When P_1's wiper is at 0V, output current flows exclusively through paralleled Q_1 and Q_3 and distributes itself to Q_4 and Q_5, depending on the position of the wiper of potentiometer P_2. With P_2's wiper at its upper end, the circuit emits 100% green light. At 0V, the emitted light is fully red. An intermediate position of the wiper yields a mixture of red and green. By moving P_1's wiper from the ground position, the circuit produces a mixture of red, green, and blue.

Transistors Q_1, Q_2, and Q_3 should tightly match. You need a difference in base-emitter voltages of less than 1.5 mV. The same requirement holds true for the Q_4/Q_5 pair. Matching requirements are less stringent for Q_6. You should use a bipolar NPN matched-transistor pair for Q_1 through Q_6, or at least Q_1 through Q_5, whereas Q_6 is a single transistor. Eventually, you can use three matched-transistor pairs. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Amplifiers deliver accurate complementary voltages

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

The circuit in **Figure 1** generates two analog voltages, which you can vary in a complementary manner. When the straight output voltage rises, the complementary output voltage decreases, and vice versa. The sum of both output voltages is a constant: $V_{OUT}+V_{OUTC}=V_{REF}$, where V_{OUT} is the straight output voltage, V_{OUTC} is the complementary output, and V_{REF} is a reference voltage you derive from bandgap cell IC_1. You choose the ratio of the resistor divider that connects to the output of IC_1 so that the reference voltage is approximately 400 mV. Potentiometer R_P sets the desired analog voltage, which connects to the noninverting input of voltage follower IC_{2A}. The output of IC_{2A} provides the straight output voltage, which connects to the inverting input of unity-gain inverter IC_{2B}. The noninverting input of IC_{2B} has a gain of two and connects to the middle of the high-precision resistive divider comprising R_1 and R_2, which halves the reference voltage. The following **equation** calculates the output voltage of IC_{2B} with respect to ground: $V_{OUTC}=-V_{OUT}+2\times(V_{REF}/2)=V_{REF}-V_{OUT}$. Thus, the straight output voltage plus the complementary output voltage give the desired constant value equal to the reference voltage.

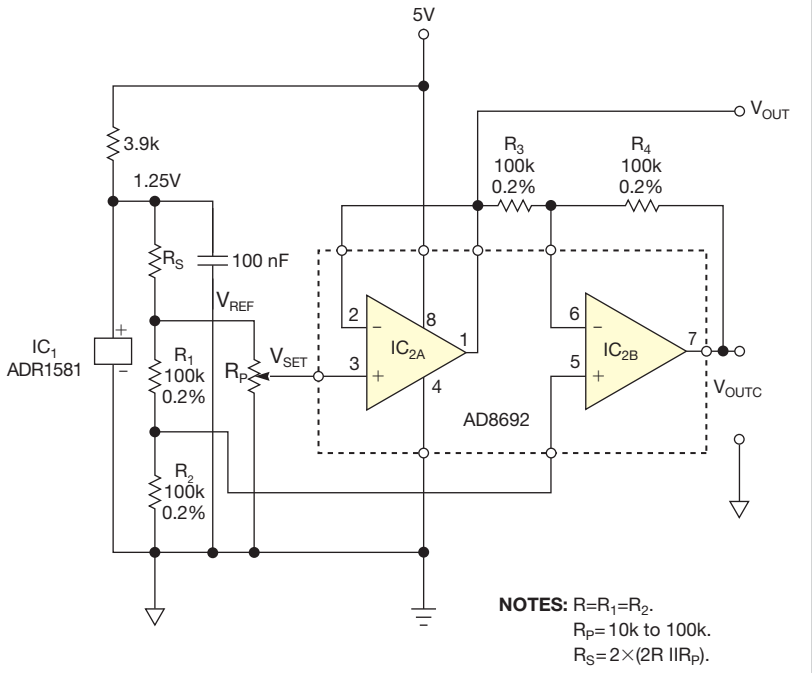


Figure 1 The circuit outputs two analog voltages whose sum always equals the reference voltage.

You should use either a quad resistor or two pairs of matched resistors for precision resistors R_1 through R_4. Resistors R_3 and R_4 form the negative feedback in IC_{2B}, and the other pair of resistors halves the reference voltage. You can omit these four resistors if you use an instrumentation amplifier instead of IC_{2B}. In this case, you must use an RRIO (rail-to-rail-input/output) type of instrumentation amplifier. The output of a contemporary RRIO instrumentation ampli-

TABLE 1 COMPLEMENTARY VOLTAGES FOR THREE INPUT SETTINGS

V_{SET}	V_{OUT} (mV)	V_{OUTC} (mV)
V_{REF}	411.45	0.15
0	0.45	410.45
$V_{REF}/2$	205.8	205.1

DIs Inside

▸ What are your design problems and solutions? Publish them here and receive $150! Send your Design Ideas to edndesignideas@cancom.com.

fier approaches the low side by a margin of approximately 60 mV, and it would severely degrade the circuit's accuracy. The output of the Analog Devices (www.analog.com) AD8692 op amp, however, typically approaches the lower rail by 0.75 mV at a 10-µA load current. The guaranteed value of the margin is 1 mV at this current.

The circuit has undergone testing for three values of test voltages: the reference voltage, which represents a full-scale; half the reference voltage; and 0V. **Table 1** lists the measured voltages at both outputs. Any of the output voltages can approach the lower supply rail with an error of less than 0.25% at 400 mV full-scale. EDN

Circuit lets you isolate and measure current

Anton Mayer, Murr, Germany

You often need to measure current during circuit design and debugging. You can perform that task by breaking a path, inserting a shunt resistor, measuring its voltage, and converting the voltage to current. Unfortunately, that approach is sometimes impractical with an oscilloscope because one side of an oscilloscope probe connects to ground. Thus, you need to isolate the oscilloscope from the circuit under test.

The circuit in **Figure 1** produces a voltage proportional to current and isolates the oscilloscope from the measurement point. The circuit uses IC_1, an HCPL7800 isolation amplifier, which adds input-to-output isolation of as much

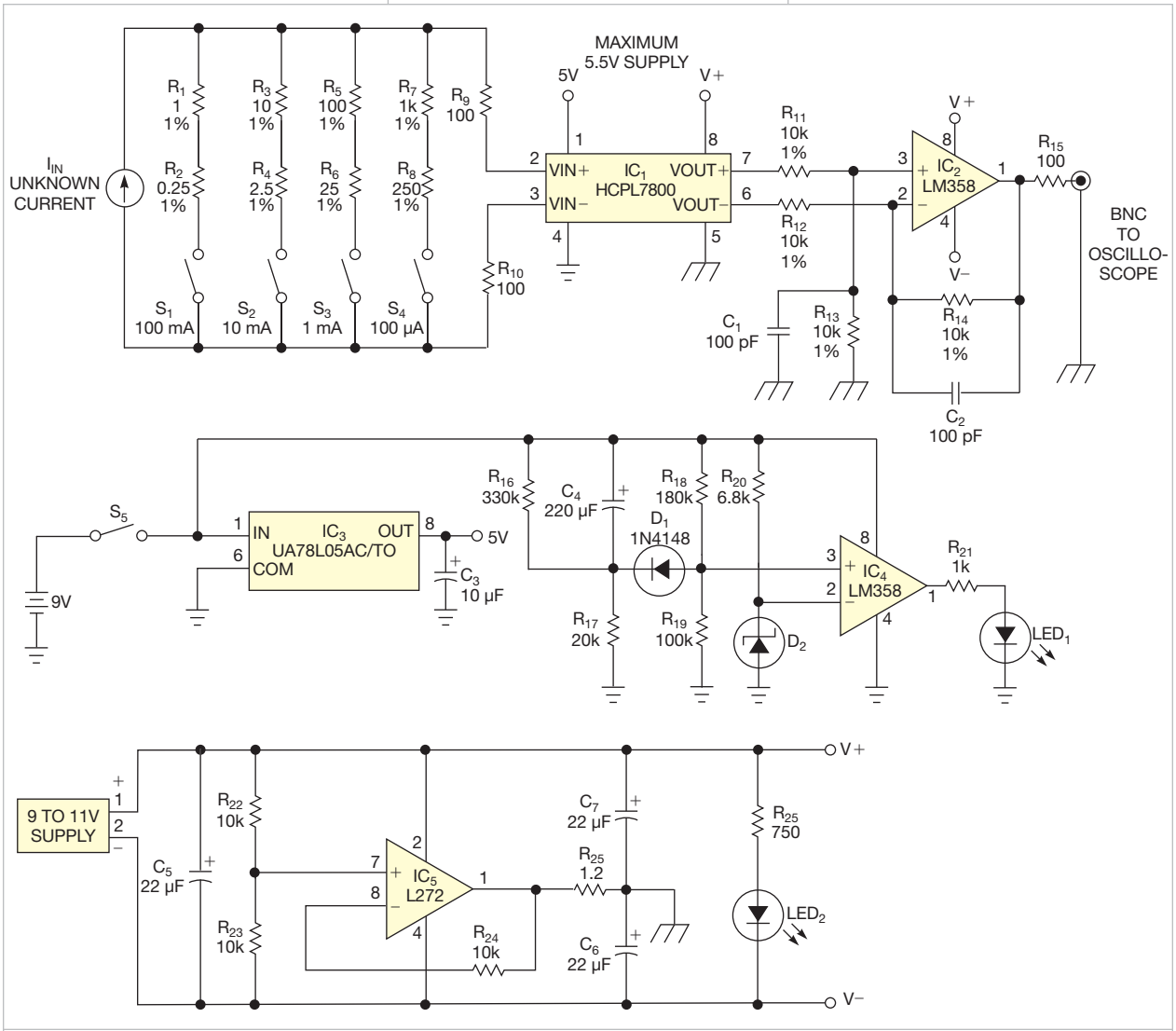


Figure 1 This circuit produces a voltage proportional to current and isolates the oscilloscope from the measurement point.

TABLE 1 GAIN FOR EACH INPUT RANGE

Switch	Gain	Maximum input current
S_1	1V/100 mA	160 mA
S_2	1V/10 mA	16 mA
S_3	1V/1 mA	1.6 mA
S_4	1V/100 µA	160 µA

as 890V. It also amplifies its input voltage by eight. **Table 1** shows the overall gain for each input range. The circuit's bandwidth is typically 100 kHz.

A set of switches lets you select a range of current to measure by inserting resistors into the circuit. Use resistors with 1% or less tolerance to minimize errors. For example, when you close S_4, you select the 100-µA range. The unknown current passes through serial resistors R_1 and R_2, which have values of 1 and 0.25Ω, respectively. Thus, the voltage at IC_1's inputs is I_{IN}×1.25 kΩ; if the input current is 100 µA, the voltage at IC_1 is 125 mV. The circuit has a gain of eight, yielding 125 mV times 8, or 1V. The LM358 acts as a unity-gain differential amplifier. For best linearity, the input voltage at IC_1 should not exceed ±200 mV.

A 9 TO 11V WALL-WART POWER SUPPLY POWERS THE OUTPUT SIDE OF IC_1 WITH IC_2, AN LM358 AMP.

The HCPL7800 has a 3% tolerance. When you are using resistors with 1% tolerance, the 3% tolerance dominates the overall uncertainty of the circuit. The circuit uses two independent voltage supplies. A 9V battery supplies the input part of IC_1. A stabilized 9 to 11V wall-wart power supply powers the output side of IC_1 with IC_2, an LM358 successive amplifier.

When battery switch S_5 closes and the voltage of the battery is sufficient for the circuit, LED_1 illuminates for approximately 3 seconds. The duration of this illumination minimizes drain on the battery. LED_2 is on when the 9 to 11V power supply is operating. IC_5, an L272, provides an additional ground potential halfway between the supply voltage. With this split supply, you can measure both positive and negative currents. EDN

Acquire images with a sensor and a microcontroller

Ioan Ciascai, Technical University of Cluj-Napoca, Cluj-Napoca, Romania, and Liliana Ciascai, Babes-Boylai University, Cluj-Napoca, Romania

The TAOS (Texas Advanced Optoelectronic Solutions, www.taosinc.com) TSL1412S image sensor, IC_2, can acquire a linear image of 1536×1 pixels, or 400 dpi (**Figure 1**). It uses a single voltage supply, and you can control it with just a few digital signals. Thus, you can design an image-acquisition system that uses the sensor and an AVR (www.atmel.com) ATmega328 microcontroller, IC_1.

Figure 1 shows how you can connect the sensor to the microcontroller. You program the microcontroller to generate the control signals for the sensor. The design uses a 16-MHz clock frequency. The microcontroller's 8-bit Timer 2 generates the command signals. In Mode 2, the timer generates hard clock signals CLK1 and CLK2 and soft strobe signals SI1, HOLD1, and HOLD2. The TSL1412S uses serial connections. The SO2 signal connects to the ICP input of TSL1412S when you activate flag ICF1.

Timer 2 generates a handler interrupt, which ensures the correct phase of the clock signal, generates the strobe signal, and acquires and saves the TSL1412S's output analog data. You can see a model for the interrupt subroutine in the online version of this Design Idea at www.edn.com/100923dia. The code sets the microcontroller's stack, register, ADC, Timer 2, and interrupt functions. To save image data, you must set the T bit in SREG to 1 and set pointer X=0×0200. You can do these settings in the last clock of time integration (R25, R24=0×0001).

By modifying the data from the register, you can set the sensor's integration time at 2.5 to 50 msec, or 100 msec with the prescaler of T2. Knowing that the sensor acquired the data in the previous cycle, you can perform a data-acquisition design using the microcontroller's internal ADC. The integration time must be greater than 50 msec.

THE SYSTEM QUICKLY PROCESSES THE SENSOR'S ANALOG OUTPUT SIGNAL.

The conversion time for a 16-MHz-frequency clock is approximately 16 µsec, which corresponds to an integration time of approximately 25 msec. Because the conversion frequency is 1 MHz—higher than that of IC_1's recommended frequency of 200 kHz—you reduce the ADC's precision from 10 bits to 8 bits. The microcontroller saves a byte for each pixel, which lets you save the data to the microcontroller's internal memory for one frame. The rest of the microcontroller's 2-kbyte memory performs stack and data-acquisition tasks.

The system quickly processes the sensor's analog output signal through the analog comparator of the microcontroller's internal schematic. You can make a comparison with a fixed voltage using an internal voltage reference of 1.25V and a resistive divider or a variable voltage you can obtain from a DAC or a PWM (pulse-width-modulated) signal the microcontroller's timer generates. EDN

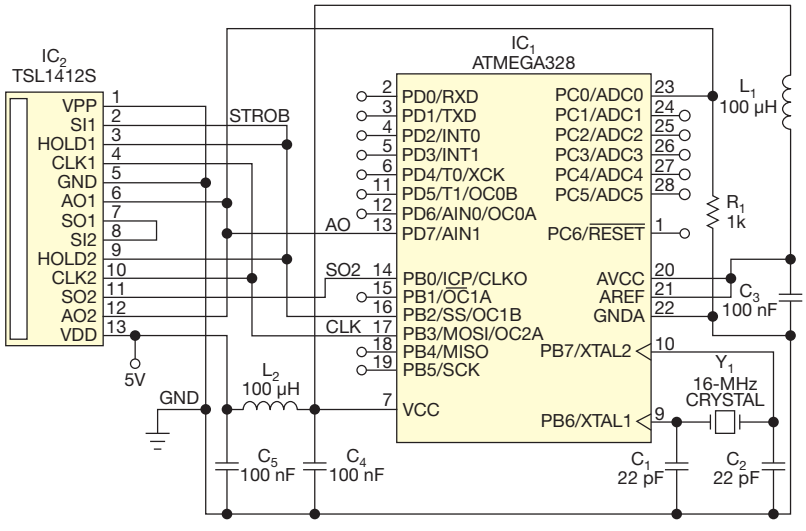


Figure 1 A microcontroller can produce control signals for an image sensor.

Power-supply circuit operates from USB port

Stefano Palazzolo, Senago, Italy

Every PC has a USB (Universal Serial Bus) port that can supply 5V±5% at 500 mA for peripherals. Powered USB hubs also provide this power. You can use a USB port to power an external circuit, which is useful when you have no other dc source available.

A USB port has V_{BUS}, the power pin; a return pin, GND (ground); and two signal pins. If you need just a simple 5V supply, you can tap the power pins from a USB connector, but you should place a 10-µF filter capacitor between the ground and power-supply pins.

You can, however, use an adjustable voltage regulator to get voltages of 1.25 to 3.75V, a range that many circuits use. The circuit in **Figure 1** covers that range. You use R_3 to change that range, as the following **equation**

shows: V_{OUT}=1.25V×(1+R_3/R_2). The 1.25V in the **equation** occurs because the LM1117-ADJ linear regulator generates 1.25V between the V_{OUT} and the ADJ (adjust) pins. Resistor R_2, therefore, has a constant current that passes through resistor R_3; the I_{ADJ} (adjusted current) is generally small enough to ignore. Selecting 100Ω for R_2 sets its current to 12.5 mA. If you use a 200Ω potentiometer for R_3, you get a voltage range of 1.25V when R_3 is 0Ω, causing a short, to 3.75V when R_3 is 200Ω.

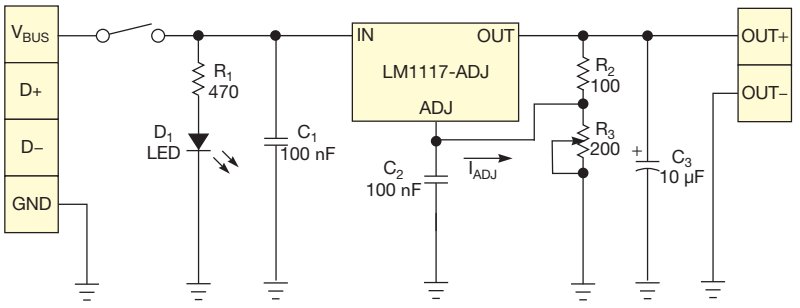


Figure 1 Resistors R_2 and R_3 set the adjustable voltage regulator's output at 1.25 to 3.75V.

To prevent circuit damage if the output becomes shorted or when you don't know the load, you can add a current-limiting circuit that keeps the maximum current at 500 mA. A polyswitch fuse or pair of transistors can easily implement this current-limiter site at the power-supply input line.

The filter capacitor shouldn't exceed 10 µF. That level keeps the inrush current under control in the absence of a current-limiting circuit. Generally, capacitors of 1 to 10 µF work best. EDN

LED-flashlight circuit works at voltages as low as 0.5V

GY Xu, XuMicro, Houston, TX

Most commercial LED flashlights use three AAA or AA batteries in series that produce 4.5V. The batteries then drive four white LEDs that connect in parallel. These LEDs can work at voltages as low as 2.7V and, in some cases, 2.4V. At those voltages, the LEDs become dim, and you must frequently change the batteries. Thus, the lowest working voltage in this case is approximately 0.8 to 0.9V per battery.

When a 1.5V alkaline battery discharges to 0.9V, it still has more than 10% of its original energy left. If you replace or discard the battery, you waste that energy. You can, however, use this small amount of battery energy with the circuit in **Figure 1**. The Linear Technology (www.linear.com) LT1932 LED driver is a step-up voltage-booster chip with constant-current capability for LED lighting. It works with input voltages of 1 to 10V, and it can drive several serial LEDs.

The trick is to choose the supply voltage. Because LT1932 can work at voltages as low as 1V, using a two-cell, 3V supply results in the lowest working voltage: 0.5V per cell. Choosing a three-cell, 4.5V supply results in a lower voltage of 0.33V per cell. A 4.5V supply can power as many as eight LEDs. Tests show that this circuit works from 4.5V to 0.94V, which is lower than the data-sheet-specified 1V. The LED driver uses a 4.7-µH inductor.

Setting the value of resistor R_1 regulates the constant current through the LEDs. Setting a higher resistance results in lower brightness. In this case, the current is 18 mA. The LT1932 is available only in a surface-mount, tiny, six-pin SOT package, with fine pitch as small as 0.037 in. So, using it requires a PCB (printed-circuit board).

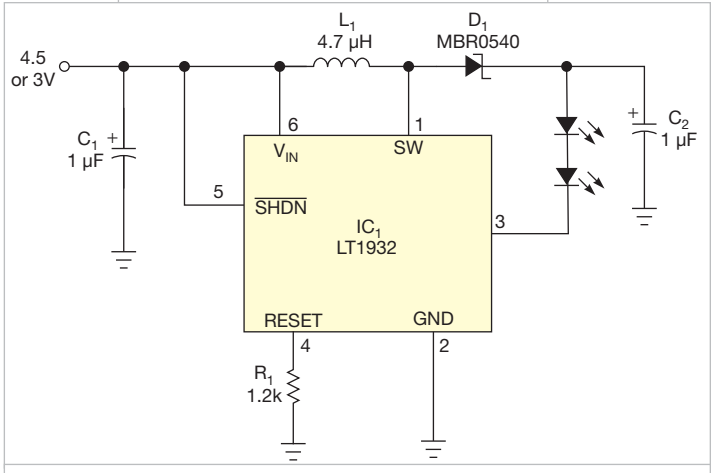


Figure 1 This circuit lets you use the leftover energy from a "dead" battery.

USE "DEAD" BATTERIES FROM COMMERCIAL LED FLASHLIGHTS TO FEED THE LT1932.

Once you build a flashlight with this circuit, you don't throw away the "dead" batteries from commercial LED flashlights. Instead, use them to feed the LT1932 circuit. Depending on whether you use an AA or an AAA cell, it will give you a few more hours of light. Keep in mind that some battery manufacturers define 0.5V per cell as an alkaline battery's cutoff voltage and recommend that you remove the battery from the load to avoid the possibility of battery leakage and gassing effects. Don't let these effects happen to your flashlight. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Ceramic speaker driver also drives motors

John Guy, National Semiconductor Corp, Santa Clara, CA

National Semiconductor's (www.national.com) LM48556 ceramic speaker driver drives speaker elements with peak-to-peak voltages as high as 14V. However, the device also works in other applications. This Design Idea shows how to use the speaker driver to drive an ERM (eccentric-rotating-mass) vibration motor. In this application, the speaker driver delivers as much as 5V into a typical ERM motor, decreasing its start-up time by approximately 50%.

The LM48556 has balanced differential inputs with an internal bias voltage.

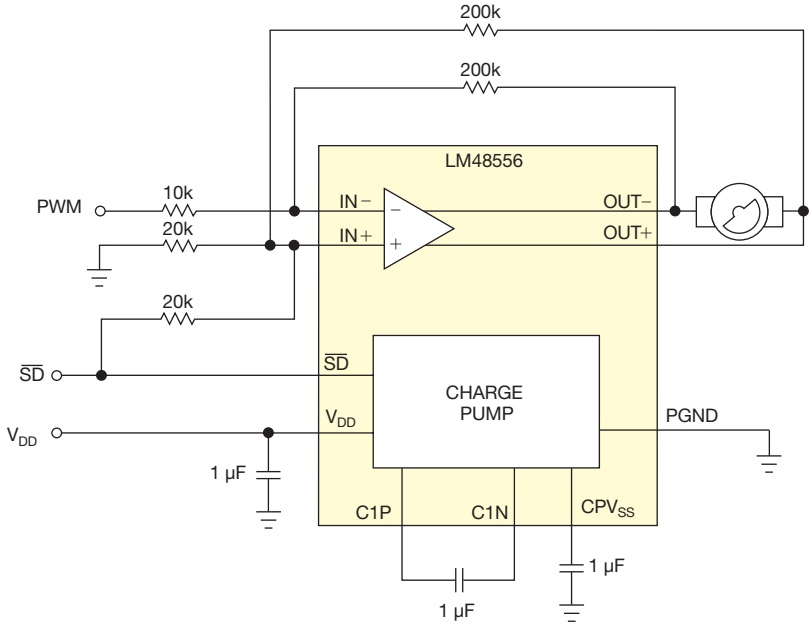


Figure 1 The LM48556 can operate dc-coupled to drive a motor; it needs ac-coupling capacitors to drive a speaker.

TABLE 1 BILL OF MATERIALS

Quantity	Description	Dimensions (L×W×H, mm)	Part
One	Driver	2×1.5×0.65	LM48556
Three	1-µF capacitor	1×0.5×0.5	6V X5R
Seven	Miscellaneous resistors and capacitors	0.6×0.3×0.3	Noncritical

DIs Inside

▸To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

▸What are your design problems and solutions? Publish them here and receive $150! Send your Design Ideas to edndesignideas@cancom.com.

Although it functions with ac-coupling capacitors in series with the input resistors and across the feedback resistors, it also functions as a dc-coupled device without the capacitors. **Figure 1** shows the external components you need to drive an EMR motor with the IC.

Note that the shutdown $\overline{SD}$ signal's high voltage must be the same as the PWM (pulse-width-modulation) high level to get a differential zero output at 50% duty cycle. Testing found this circuit operational with logic voltages of 1.2 to 3.4V, but the speaker driver has a guaranteed logic high of 1.2V or higher. If the PWM source has a logic voltage of less than 1.2V, you must drive the $\overline{SD}$ pin with a separate higher logic voltage.

When you use this device with 1-µF capacitors, it can deliver 5V into 30Ω at voltages as low as 3.2V. Replacing these capacitors with a 4.7-µF "flying" capacitor and a 10-µF reservoir capacitor allows the LM48556 to drive 5V into 30Ω at voltages as low as 3V. **Table 1** lists the parts you need. EDN

Light collar helps you find your pet in the dark

Vladimir Oleynik, Moscow, Russia

If you have a pet, you know how difficult is to find him or her in the dark, especially when your pet has dark fur. The circuit in **Figure 1** lets you build a collar with LEDs that helps you find your pet.

The heart of the circuit is the ZXLD381FHTA LED driver from Zetex (www.diodes.com). Two 1.2 or 1.5V cells power the LED driver, providing the current pulses to illuminate six to eight 20-mA LEDs of any color series. The ZXLD381 dc/dc boost converter's primary application is in LED drivers. A 10-µH inductor helps maintain output voltage higher than the input voltage. Zetex's ZXLD381 data sheet contains hints on inductor selection (**Reference 1**).

The circuit contains a light sensor and the LED driver. The LED driver's operation starts when the ambient-light level drops below a certain threshold. When the light level is high, the low collector-to-emitter resistance of the L-7113P3C phototransistor from Kingbright (www.kingbright.com) keeps transistors Q_1 and Q_2 off, and the LEDs don't light. When it gets dark and the voltage at Point A goes high enough, Q_1 and Q_2 turn on and saturate, providing power for the LED driver.

To power the light collar, you can use any cell or battery with voltage of 1.2 to 10V. Zetex's ZXLD381FHTA operates at 0.9 to 10V at its common-cathode-voltage pin, V_{CC}, but the light sensor's voltage drop slightly narrows it. The LEDs' efficiency decreases when the power supply is at its low limit. If you don't need the light sensor, then omit all resistors, the phototransistor, and both transistors. Meanwhile, the light sensor lengthens battery life, especially if you expect to use the collar frequently. If you use the collar, you can discharge the battery to a voltage lower than 1.2V before discarding it. When the battery voltage drops to 1.2V, connect its battery's positive terminal through switch S_1 directly to the LED driver's V_{CC} pin, which bypasses Q_1's collector-to-emitter junction with jumper J_1. In that case, the light sensor is off, and the minimum power-supply range decreases to 0.9V.

The ZXLD381FHTA comes in a space-saving SOT-23 package. The other circuit components are available in either surface-mount or through-hole versions. The surface-mount versions for Q_1 and Q_2 are the BC857C and the BC847C, respectively, and the through-hole versions are the BC557C and the BC547C, respectively. The circuit's size does not exceed 0.5 in.^{2} (12.7×25.4 mm) when you use SMD components and 1 in.^{2} (25.4×25.4 mm) when you use through-hole-type components.

TO IMPROVE DETECTION OF YOUR PET, YOU SHOULD PLACE ALL LEDs EVENLY ALONG THE COLLAR'S PERIMETER.

The phototransistor in a through-hole package is easier to see under your pet's fur. Also, it's more convenient to use LEDs with wider viewing angles and larger diameters.

Powering the collar with two fresh AA 1.5V cells in a well-illuminated environment in standby mode consumes approximately 25 µA. In a dark environment, the average consumption is about 7 mA. Four Kingbright L-53SRD-H LEDs illuminate the collar.

To improve detection of your pet, you should place all LEDs evenly along the collar's perimeter. You may also need to prevent the pet's fur from overshadowing the phototransistor and LEDs. Your pet may need some time to get used to its new lighting collar. The collar bends in different directions during use, so make it flexible or arrange it as several rigid PCBs (printed-circuit boards) with a wired connection. EDN

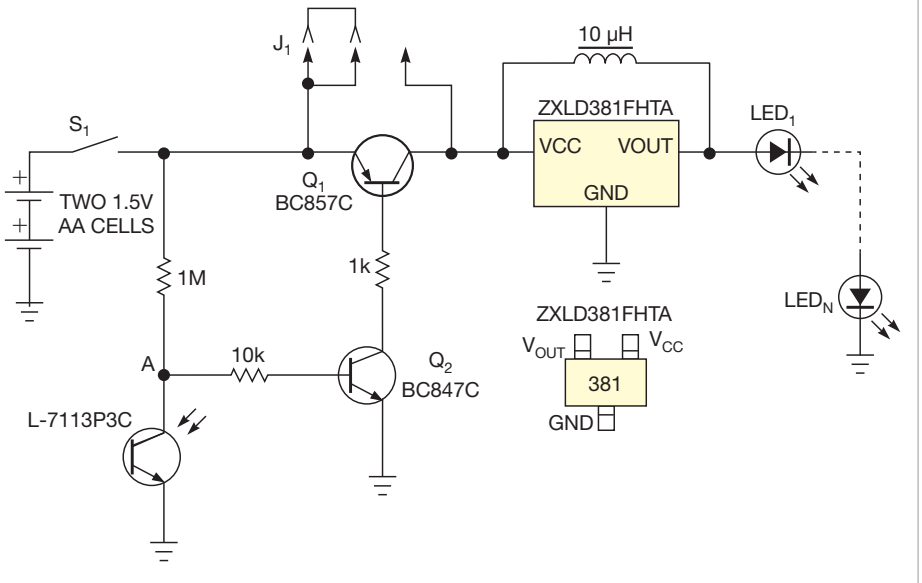


Figure 1 This circuit lets you build your pet a lighted collar.

REFERENCE

1 "ZXLD381, Single or multi cell LED driver solution," Diodes Inc, May 2010, www.diodes.com/datasheets/ZXLD381.pdf.

Accurately simulate an LED

Jon Roman and Donald Schelle,
National Semiconductor Corp, Santa Clara, CA

Solid-state-lighting applications are quickly moving into the mainstream. Although they are more efficient, the LEDs that produce the low-cost light often require a complicated driver circuit.

Testing the driver circuit using LEDs, although easy, yields only typical results because the tests don't factor in worst-case LED parameters and often generate undesirable light and heat during driver debugging. Although using a constant resistance might seem to be an appropriate approach, a resistor approximates an LED load at only one point on the current/voltage curve. An electronic load may prove to be a more useful approach. The control loops of the driver circuit and the electronic load, however, often result in system instability and oscillations.

Figure 1 illustrates a typical LED-driver circuit using a low-cost simulated-LED circuit. The simulated LED accurately mimics a real LED at a user-programmable threshold voltage. A simple Darlington current sink, Q_1, provides a wide range of LED threshold voltages. The size of the heat sink attached to Q_1 and the power capability of Q_1 are the only limits on the amount of power the simulated LED can dissipate.

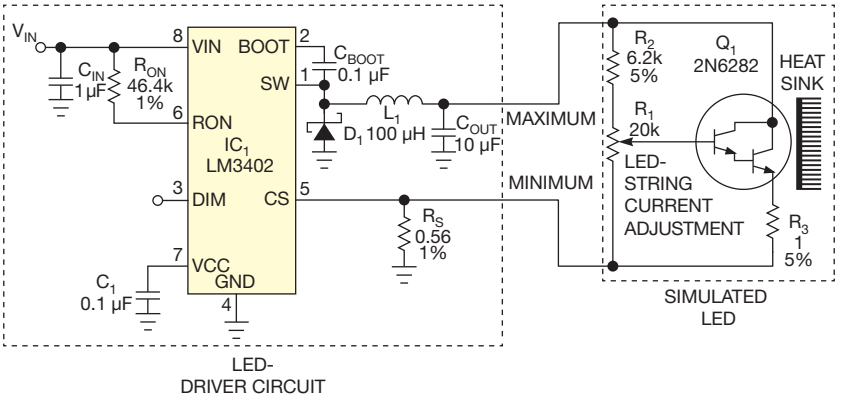


Figure 1 Use this circuit for quick testing of an LED-driver circuit over minimum, typical, and maximum LED parameters.

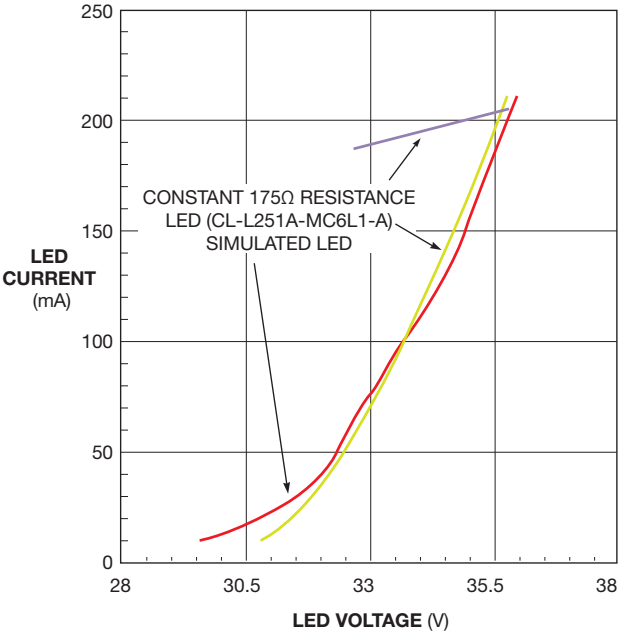


Figure 2 The simulated LED approximates the turn-on characteristics of a real LED. A constant-resistance load approximates a real LED load at only one point on the curve.

THE SOFT TURN-ON OF THE SIMULATED LED ACCURATELY MIMICS THAT OF A REAL LED.

You can easily tune the circuit for any LED voltage. Place a constant voltage across the simulated LED. Tune the circuit by adjusting resistor R_1 until the circuit draws the desired current. You can adjust the shape of the voltage knee by making small changes to resistor R_3, although this step is not usually necessary.

Figure 2 compares the simulated LED's current and voltage characteristics to those of a real LED and a constant resistance. The soft turn-on of the simulated LED accurately mimics that of a real LED. Furthermore, the simulated LED quickly retunes to test minimum and maximum LED characteristics, thus giving you confidence that the circuit will work over all load conditions. EDN

Perform hexadecimal-to-BCD conversion in firmware

Abel Raynus, Armatron International, Malden, MA

Microcontroller firmware usually deals with hexadecimal code. You sometimes need to display the content of registers, which requires a hexadecimal-to-BCD (binary-coded-decimal)-code conversion. The programming technique for this conversion is an add-three algorithm. Fortunately, you can adapt this technique for use with an 8-bit microcontroller assembler. You can, however, easily adapt this approach to any assemblers because the approach uses a set of standard instructions.

The number of BCD registers this program requires depends on the number of bits of the hexadecimal digits necessary to contain the maximum value of equivalent BCD code. The most frequent situation for 8-bit microcontrollers is an input hexadecimal code that comprises one byte with a value of $00 to $ff. Its decimal equivalent, accordingly, has a value of 00 to 255. Hence, it needs three digits to be displayed. Each BCD digit needs 4 bits for presentation. Thus, you should organize three 4-bit groups, or BCD tetrads, in firmware.

EACH BCD DIGIT NEEDS 4 BITS FOR PRESENTATION, SO ORGANIZE THREE 4-BIT GROUPS, OR BCD TETRADS, IN FIRMWARE.

The units and tens tetrads form one 8-bit BCD register, and the hundreds tetrad forms half of the second register. Note that all registers for hexadecimal and BCD digits should reside consecutively in memory for ease in performing the multibyte-shift operation. For this reason, they are called digit for initial hexadecimal code, digit+1 for BCD units and tens, and digit+2 for BCD hundreds (**Figure 1**).

Figure 2 shows the flow chart of the conversion process. You can find the assembly code for the procedure in the online version of this Design Idea at www.edn.com/101007dia. The conversion starts by checking each BCD tetrad to detect whether its value is more than or equal to five. If it happens in any of BCD tetrads, then it adds a three to that tetrad. Next, all the registers—digit, digit+1, and digit+2—shift left starting with the LSB (least-significant-bit) digit. These operations continue until the registers shift N times, where N is the number of bits in the initial hexadecimal code. In this case, N=8. The result of conversion can be read from the registers digit+1 and digit+2. EDN

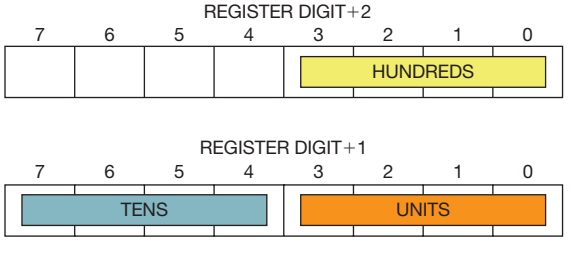


Figure 1 Representing two hex digits (one byte) in BCD format requires three 4-bit tetrads.

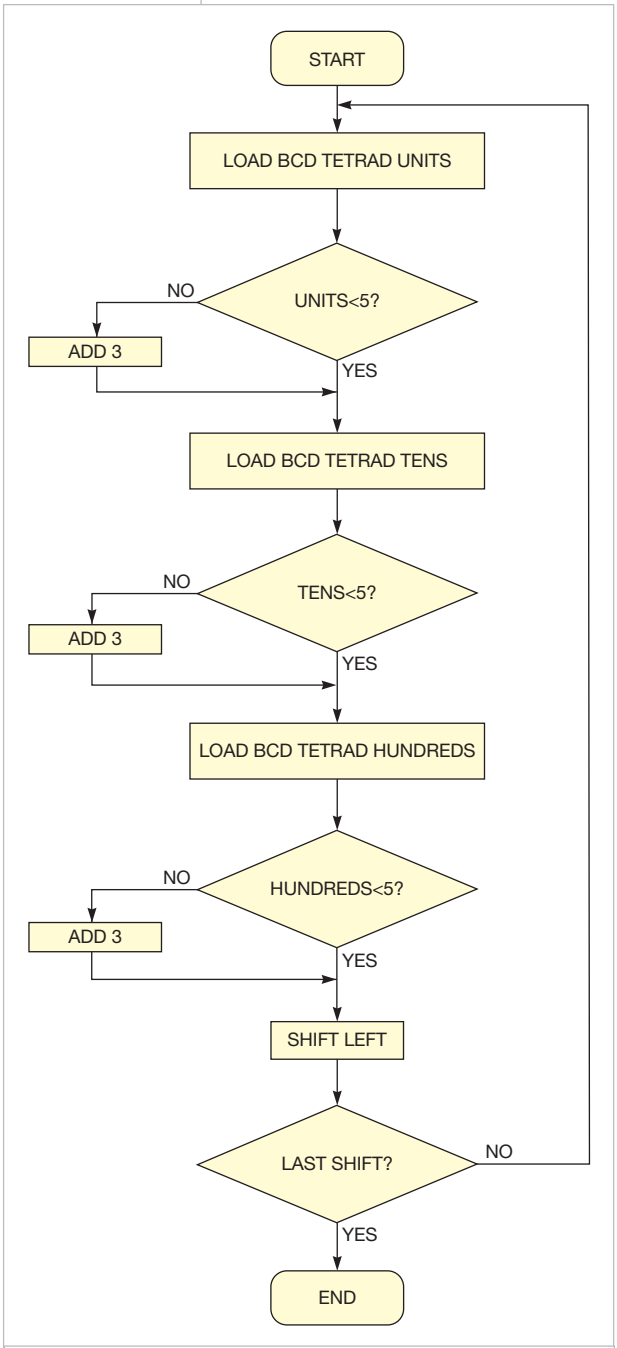


Figure 2 The conversion algorithm checks the units, tens, and hundreds for values less than five and adds three.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Set LEDs' hue from red to green

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

The circuit in **Figure 1**, which lets you create light of 32° of hues, uses red and green LEDs. A constant current divides into two components. One component flows through a red LED, and another one flows through a green LED. You can vary the current from 0 to 100% through the red LED, and thus you simultaneously vary the current through the green LED as a slave-type complement to 100%. When this scenario happens, your eye perceives the resulting light mixture as any hue between red and green. Roughly speaking, the transition from red to green passes through orange, amber, and yellow. You can set any of the 32 hues between red and green, passing through orange, amber, and yellow.

IC_3, an Analog Devices (www.analog.com) AD5228 resistive DAC, has one-in-32 resolution, and it thus sets the resolution of this circuit. In this application, the resistive DAC functions as a digital potentiometer. You can manually set its wiper position through short-term grounding of its $\overline{PU}$ pullup and $\overline{PD}$ pulldown control pins. The resistive DAC has no memory, so you have to make this setting after each power-on.

Holding the $\overline{PU}$ and $\overline{PD}$ pins to a logic low, the wiper position increments or decrements with an increased speed of one step per 0.25 sec, so the output light's color varies stepwise for a low pullup (**Figure 2**). You can also preset the hue of the LED, which appears at power-on. For a high Preset, the color is 100% red when you apply power. At a low Preset, a midposition is preset at the resistive DAC and thus

DIs Inside

▶What are your design problems and solutions? Publish them here and receive $150! Send your Design Ideas to edndesignideas@cancom.com.

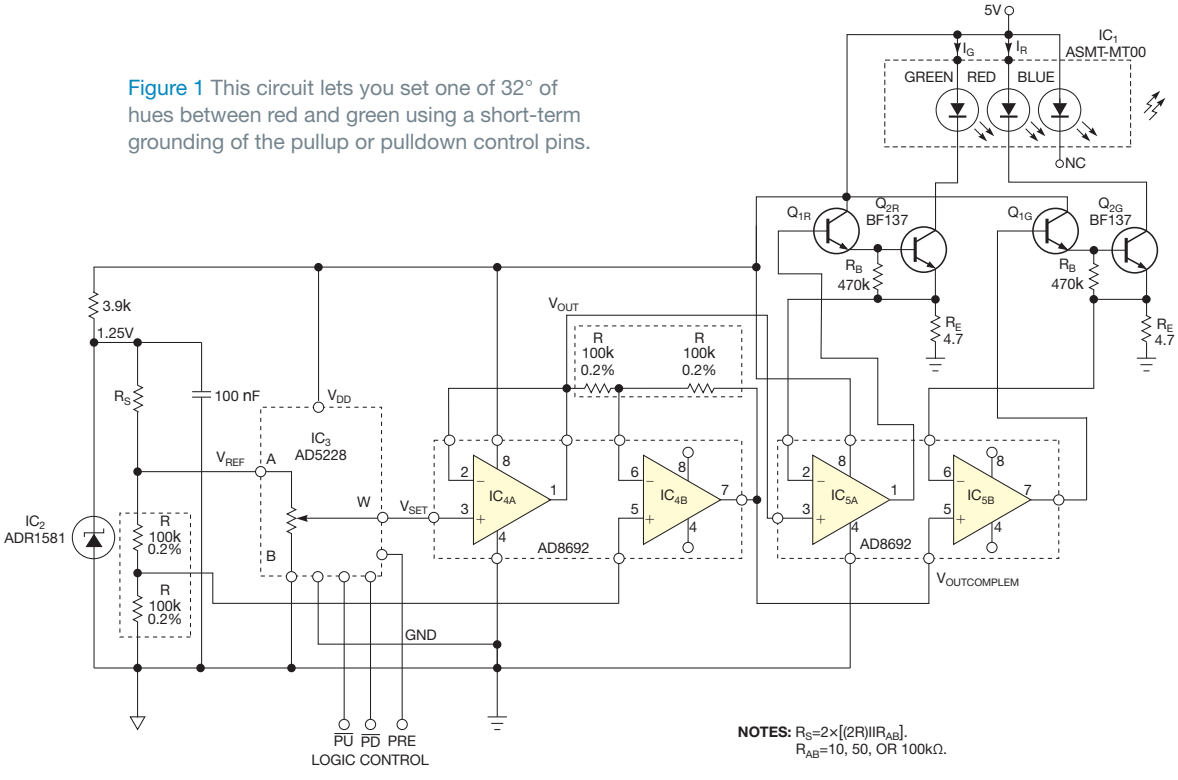


Figure 1 This circuit lets you set one of 32° of hues between red and green using a short-term grounding of the pullup or pulldown control pins.

the color at power-on is 50% red and 50% green; you perceive it as yellow.

The circuit uses two LEDs in IC_1, a high-performance, tricolor ASMT-MT00 LED from Avago Technologies (www.avago.com). The blue LED remains unused. You can, however, connect any of the remaining five red/green, red/blue, blue/red, green/blue, or blue/green combinations instead of the green/red combination this circuit uses.

Although the sum of currents flowing through the red and green LEDs is approximately one-fourth of the nominal per-LED current, the radiance is high, and you should not look directly at the lid of IC_1 when it is on from a distance of less than approximately 1 foot.

IC_2, IC_3, and IC_4 comprise a low-side source of two complementary analog voltages (**Reference 1**). The resistive DAC replaces the classic potentiometer in the earlier Design Idea. These complementary analog voltages are the input voltages for the two power stages comprising transistor Q_1 and midrange-power transistor Q_2.

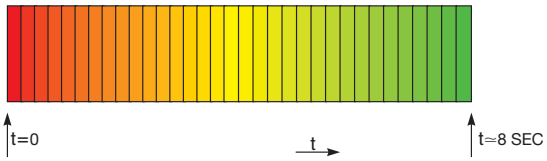


Figure 2 The light output changes quasicontinually from red to green within approximately 8 seconds, using long-term grounding of the pullup pin or a continuous grounding of the pin at power-on.

The power stage—voltage-to-current converters you make by cascading two bipolar transistors and an op amp—drives each of the two LEDs. The circuit senses output current at resistor R_E. The R_B resistors eliminate the leakage currents of both bipolar transistors in the cascaded series. These power stages would be functional even with one bipolar transistor instead of two. The cascaded bipolar transistors provide precision in the voltage-to-current converter. With a single power transistor, the relative error would be approximately $1/\beta$, whereas using the cascaded series, the error is approximately $1/(\beta_1\beta_2)$, where β_1 and β_2, the current gains of the bipolar transistors, are approximately 300 and 100, respectively. The error results from the current flowing through resistor R_E, which is the sum of the output current and the base current of transistor Q_1.

You can use this circuit in industries ranging from entertainment to toys; it may eventually find use in experimental psychology and in modern fine arts, which involves the use of optoelectronics.

Holding $\overline{PD}$ low and feeding a 50% duty cycle, 0.05-Hz-frequency logic waveform to the $\overline{PU}$ pin produces a slow, periodic, quasicontinuous "waving" of the color from red to green and back. EDN

REFERENCE

1 Stofka, Marian, "Amplifiers deliver accurate complementary voltages," *EDN*, Sept 23, 2010, pg 44, http://bit.ly/cljzKQ.

Circuit synchronizes sensors and cameras

By Shih-Jie Chou, Rui-Cian Weng, and Tai-Shan Liao,
National Applied Research Laboratories, Hsinchu, Taiwan

Measurement systems often use cameras and other sensors that require synchronization. This Design Idea describes an aerial-photography system that uses a camera comprising CCD (charge-coupled-device) image sensors, an inertial-measurement unit, and a GPS (global-positioning-system) unit. The resulting circuits provide trigger signals to synchronize the measurements at the optimal rate. The GPS provides information on spatial location, and the inertial-measurement unit provides information on spatial azimuth. The unit combines a gyroscope, a magnetometer, and an accelerometer to produce angular and acceleration measurements of a three-axis vector.

Figure 1 shows a system for taking aerial photographs. It comprises four Atmel (www.atmel.com) area-scan CCD image modules, one linear image-sensor module, two Dalsa (www.dalsa.com) PCI (Peripheral Component Interconnect) frame-grabber cards, the measurement unit, a clock-adjustment circuit, and a microcontroller. A Tektronix (www.tektronix.com) digital oscilloscope views the trigger signals during development.

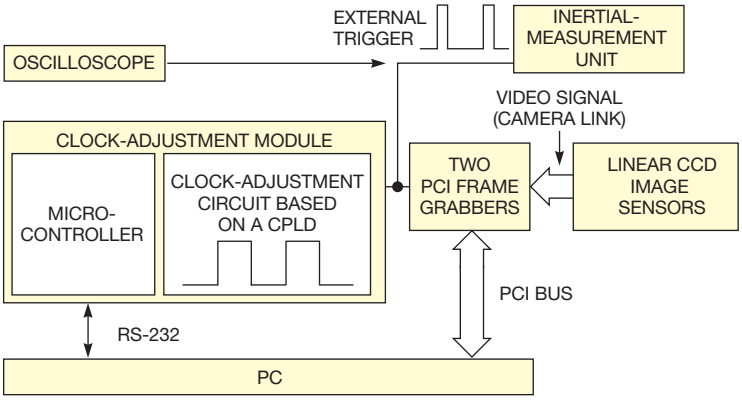


Figure 1 This clock-adjustment circuit generates a series of pulses that trigger the frame grabbers and the inertial-measurement unit.

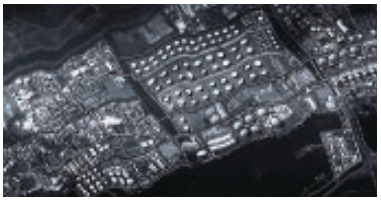


Figure 2 This aerial photograph from a height of 7000 feet was taken during a flight over Mailiao Township, Yunlin County, Taiwan.

The trigger signals that synchronize the sensors are the keys to this measurement system. The clock-adjustment circuit sends an external trigger pulse to the frame-grabber cards, which generate trigger signals for the system. Video modules comprising image sensors receive trigger signals from the frame grabbers. Each frame grabber captures an image and stores it in onboard memory before capturing the next image.

The external trigger pulses also control the sensors, GPS, and inertial-measurement unit. **Figure 2** shows a photo taken at 7000 feet in Mailiao Township, Yunlin County, Taiwan, using the external trigger circuit to drive and combine with the linear sensor and the measurement unit.

The circuit must change the external trigger clock's frequency to obtain the best frame rate. The CCD sensors that go into the linear-image-sensor module have 12,288 pixels, and each pixel measures 5×5 microns, producing images of approximately 500 lines/frame. The CCD image sensors have a maximum output rate of 320M pixels/sec. They use a Camera Link interface to send image data to the frame grabbers, which transfer the images to a PC over the PCI bus.

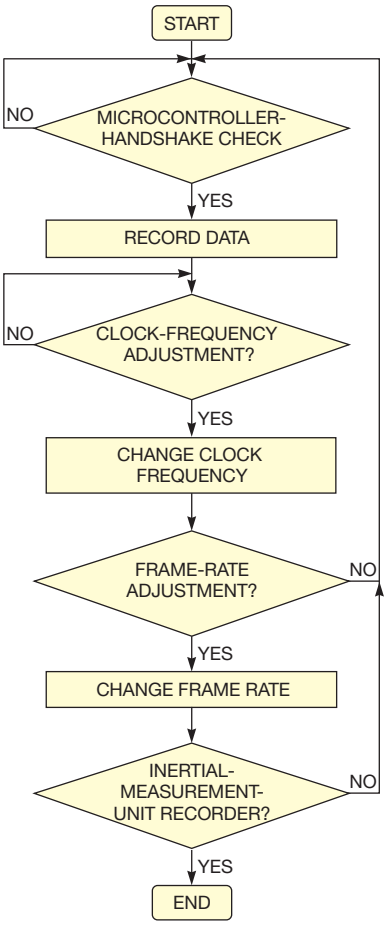


Figure 3 Microcontroller software adjusts the trigger-pulse frequency that drives the measurements.

The clock-adjustment circuit generates the external trigger clock pulses. The circuit employs on an Altera (www.altera.com) CPLD (complex programmable-logic device) using Altera's development software to simulate the trigger signals and design the circuit. The clock-adjustment circuit provides as many as 15 trigger-signal frequencies to the system.

The system's Atmel microcontroller contains 256 bytes of RAM and 8 kbytes of programmable flash memory for program storage. The microcontroller communicates to a PC over an RS-232 interface so that it can also receive commands and report its current state. This handshake process includes the decoding and encoding parameters for generating the trigger signal. The microcontroller also sends commands to the digital-timing-adjustment circuit; these commands change the pulse frequency of the external trigger.

You can adjust the frame rate of the CCD image module using 15 trigger frequencies. The external trigger signal also triggers the measurement unit to record and store spatial parameters. **Figure 3** shows the algorithm for finding the optimal trigger frequency. The frame rate and the trigger frequency are linearly proportional.

The inertial-measurement unit is a key sensor in the system, and there must be a direct correlation between it and the frame grabbers. If the external trigger frequency is 1 kHz, then each of the two frame grabbers captures 1000 frames/sec and the unit samples at 1k samples/sec. Through experimental results using aerial photography, the system successfully synchronizes all of the sensors. EDN

Circuit measures capacitance or inductance

Jim McLucas, Broomfield, CO

Engineers usually have access to signal and function generators, as well as frequency counters and oscilloscopes, but they may not have access to capacitance or inductance meters. Using the test setup in **Figure 1**, you can measure capacitance or inductance using a function generator, a multimeter, a frequency counter, and an oscilloscope.

Use the setup to measure the magnitude of two signals. You can then calculate the capacitance or inductance without measuring phase angles. You can express the ratio of input voltage to output voltage as:

$$\left|\frac{V_{IN}}{V_{OUT}}\right| = \frac{\sqrt{R^2 + X_C^2}}{X_C}, \quad (1)$$

which you can put into the standard

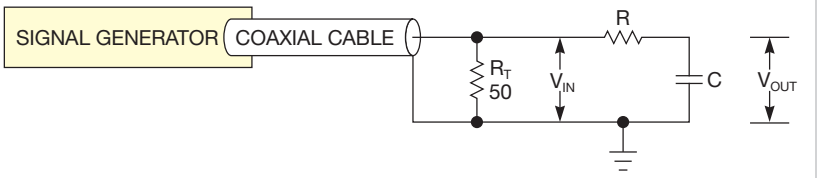


Figure 1 You can measure capacitance or inductance with this test setup.

form:

$$X_C^2+\frac{R^2}{1-\left|\frac{V_{IN}}{V_{OUT}}\right|^2}=0. \quad (2)$$

After solving the **equation** for X_C, the result is:

$$X_C=\frac{R}{\sqrt{\left|\frac{V_{IN}}{V_{OUT}}\right|^2-1}}. \quad (3)$$

Using the relationship

$$C=\frac{1}{2\pi f X_C}, \quad (4)$$

the basic **equation** for capacitance is:

$$C=\frac{\sqrt{\left|\frac{V_{IN}}{V_{OUT}}\right|^2-1}}{2\pi fR}. \quad (5)$$

Using the convenient ratio $|V_{IN}/V_{OUT}|=2$, then

$$C=\frac{\sqrt{3}}{2\pi fR}. \quad (6)$$

To measure the value of a capacitor, measure the input voltage and then adjust the frequency of the signal generator to make the output voltage one-half of the input voltage. You need not use a 2-to-1 ratio for V_{IN}/V_{OUT}. You can just measure the input voltage and the output voltage and use one of the basic equations to calculate the value of the capacitance or inductance, but a ratio close to 2-to-1 is a good choice.

For best results, you can use a frequency counter to measure the frequency and a digital multimeter to measure the resistance. Most modern oscilloscopes can accurately measure the signals without loading the circuit, except for the capacitance of the probe. Capacitance is usually marked on the probe. Use the previous **equation** to calculate the value of the capacitor. Subtract the value of the probe capacitance from the result, and you have an accurate value for the measured capacitance.

USE A FREQUENCY COUNTER TO MEASURE THE FREQUENCY AND A DIGITAL MULTIMETER TO MEASURE THE RESISTANCE.

Usually, you know the approximate value of the capacitance you want to measure, so you can pick a starting value for the resistance, R, and the frequency, f, by using the following **equations**:

$$R=\frac{\sqrt{3}}{2\pi fC}, \quad (7)$$

$$f=\frac{\sqrt{3}}{2\pi RC}, \quad (8)$$

$$X_C=\frac{R}{\sqrt{3}}. \quad (9)$$

You can use a similar procedure to measure inductance. In this case,

$$X_L=\frac{R}{\sqrt{\left|\frac{V_{IN}}{V_{OUT}}\right|^2-1}}, \quad (10)$$

and the basic **equation** for inductance is expressed as:

$$L=\frac{R}{2\pi f\sqrt{\left|\frac{V_{IN}}{V_{OUT}}\right|^2-1}}. \quad (11)$$

Set $V_{IN}/V_{OUT}=2$, then

$$L=\frac{R}{2\pi f\sqrt{3}}, \quad (12)$$

$$R=2\pi fL\sqrt{3}, \quad (13)$$

$$f=\frac{R}{2\pi L\sqrt{3}}, \quad (14)$$

and

$$X_L=\frac{R}{\sqrt{3}}. \quad (15)$$

For an example of measuring capacitance, assume C is approximately equal to 1000 pF and let f equal 1 MHz. Calculate as:

$$R=\frac{\sqrt{3}}{2\pi fC}=\frac{\sqrt{3}}{2\pi(10^6)(10^{-9})}=275.66\ \Omega. \quad (16)$$

Use a 301Ω resistor or any convenient value of approximately 250 to 500Ω in the setup of **Figure 1**. Adjust the frequency while measuring the input voltage and the output voltage to get a ratio of 2-to-1. If the frequency you obtain is 912 kHz, the measured resistance of R is 304Ω, and the probe capacitance is 10 pF, then the capacitance is:

$$C=\frac{\sqrt{3}}{2\pi(912)(10^3)(304)}=994.3\ \text{pF}, \quad (17)$$

minus 10 pF for the probe capacitance. The result is 984.3 pF.

The values for R and f are not critical; you should choose them to minimize parasitic effects. Resistance values of 300 to 3000Ω and frequencies of 100 kHz to 1 MHz should work. EDN

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Circuit achieves constant current over wide range of terminal voltages

Donald Boughton, Jr, International Rectifier, Orlando, FL

Analog-circuit design often requires a constant-current sink. An example would be for a TRIAC (triode-for-alternating current) dimmer holding current in fluorescent or solid-state lighting. Other examples include a precise current sink at the end of a long line, such as a cable or an ADSL (asymmetric digital-subscriber-line) modem, which produces a "signature" current value that alerts the device at the source end, such as an exchange office or a cable center, that the remote equipment is attached. The trick is to make a circuit that gives a constant current over a variety of terminal voltages.

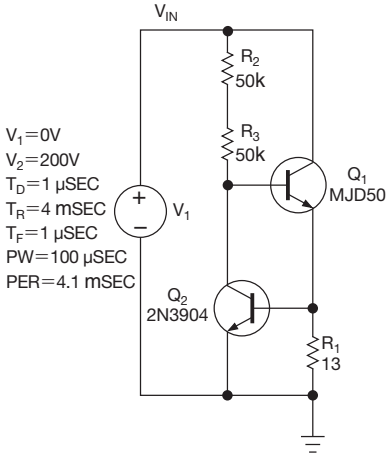


Figure 1 Resistor R_1 sets a constant current through Q_1.

A common circuit for achieving this task uses a sense resistor, a transistor, and a power device. **Figure 1** shows the circuit using a power transistor, Q_1. The circuit provides an approximate constant current at high voltages, but it doesn't enter regulation until it reaches nearly 60V due to the base current the transistor requires. **Figure 2** shows the circuit using a MOSFET, Q_2, for the power device. With a MOSFET, you can use smaller biasing resistors, and the circuit comes into regulation at a much lower terminal voltage.

Unfortunately, the current-sense resistor, R_1, in **figures 1** and **2** doesn't sense the bias current. As the terminal voltage increases, the terminal current also increases because of the increased bias current. A simple way to improve the regulation of both circuits is to add re-

DIs Inside

▸To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

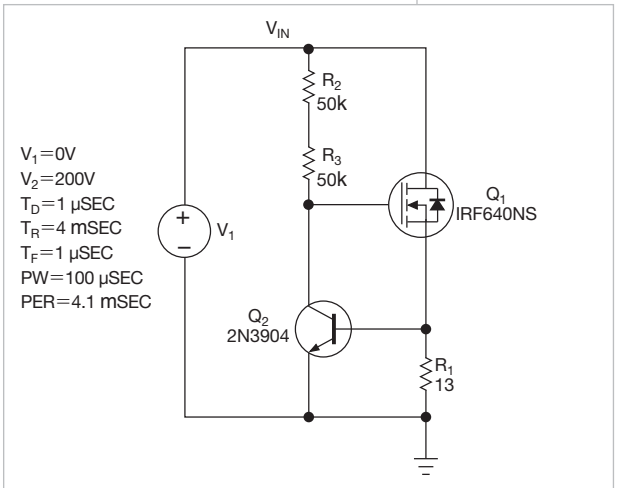


Figure 2 This circuit substitutes a MOSFET for Q_1 in Figure 1 and uses smaller resistors.

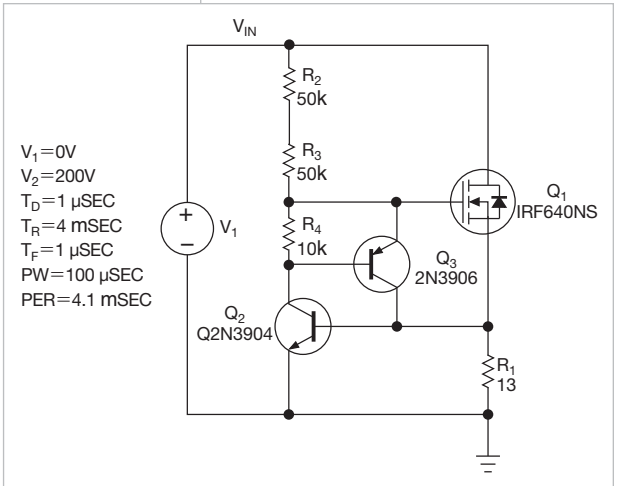


Figure 3 The addition of Q_3 and R_4 improves current regulation.

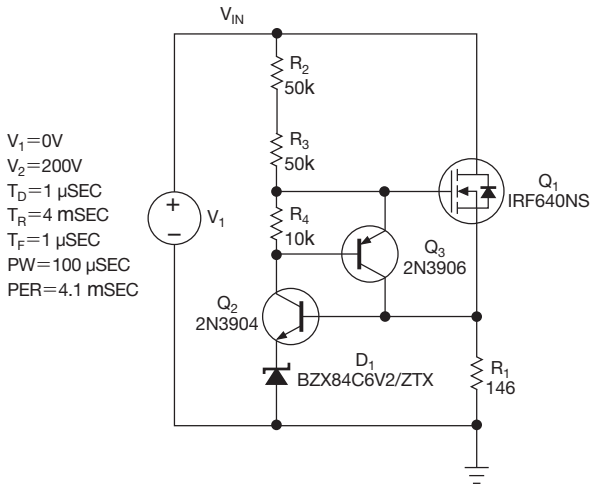


Figure 4 Adding a zener diode improves current regulation over temperature.

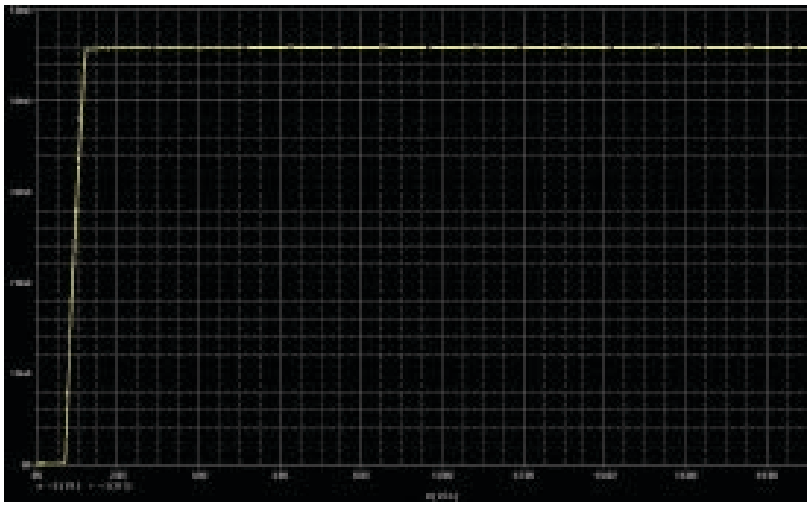


Figure 5 A constant current in Q_1 has a steep rise relative to V_{IN}.

sistor R_4 and PNP transistor Q_3 (**Figure 3**). R_4 and Q_3 form a constant-current source to the collector of Q_2. The circuit diverts any excess bias current through the collector of Q_3 to sense resistor R_1. Thus, as the terminal voltage increases, the bias current remains relatively constant, and the current regulation appears much flatter.

The negative temperature coefficient of the base-to-emitter junction of transistor Q_2 causes another problem with this kind of circuit. The temperature coefficient is approximately –1.6 mV/°C, which causes the current value to vary widely with temperature.

A 100-mV CHANGE WITH TEMPERATURE DOES NOT SERIOUSLY AFFECT THE REGULATED CURRENT.

One way to approach this problem is to add a 6.2V zener diode, D_1, in series with the emitter of Q_2, which increases the sense voltage (**Figure 4**). A 6.2V diode has a positive temperature coefficient, which counteracts the negative temperature coefficient of the transistor. Furthermore, the total sense voltage is much larger, so 100 mV or so of voltage change with temperature does not seriously affect the regulated current. **Figure 5** shows a PSpice simulation of the circuit that uses a MOSFET for Q_1. EDN

Limit inrush current in low- to medium-power applications

JB Castro-Miguens, Cesinel, Madrid, Spain
C Castro-Miguens, University of Vigo, Vigo, Spain

When switched-mode power supplies, including those for notebook computers, turn on, the bulk capacitor of the uncontrolled rectifier is completely discharged. This can result in a large charging current for a high instantaneous line voltage because the discharged capacitor temporarily short-circuits the power supply's diode bridge.

With a large bulk capacitor, the current spike can trigger the mains breaker or even destroy rectifier diodes. Capacitor and line ESRs (equivalent series resistances) and inductances help to reduce the initial spike. Even so, current peak can reach tens of amperes. The rectifier-diode selection must account for this nonrepetitive spike. An initial spike also affects the lifetime of the bulk capacitor. The circuit in **Figure 1** lets you avoid the large initial spike.

At turn-on, if the instantaneous rectified ac-line voltage is greater than about 14V, MOSFET Q_1 is on, ensuring that IGBT (insulated-gate bipolar transistor) Q_2 is off. In this situation, no current flows through charging the bulk capacitor.

Whenever the rectified ac-line voltage is lower than the voltage across the bulk capacitor plus approximately 14V ($V_1=V_{IN}–V_{OUT}\leq 14V$), Q_1 is off, and Q_2

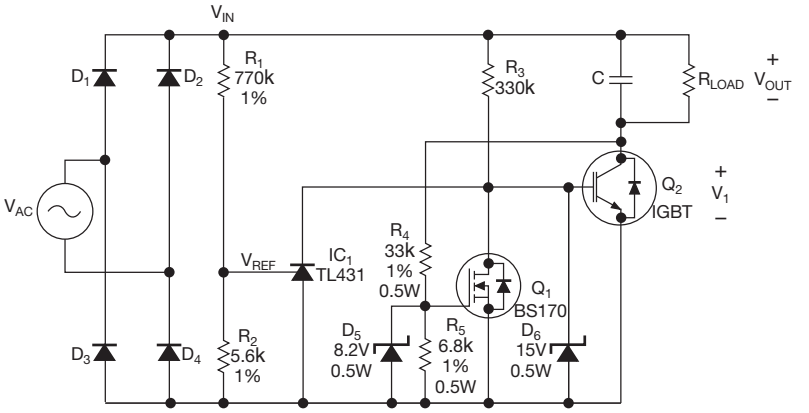


Figure 1 You can use this circuit to limit inrush current and clamp the output voltage to values lower than 380V.

turns on through R_3, connecting the capacitor and R_{LOAD} to the rectifier. Q_2 remains on thereafter, making Q_1 useless.

In the steady state, whenever the rectified input ac voltage matches the voltage across the bulk capacitor, Q_1 is off and Q_2 is on, making charging of the capacitor possible.

The current-limiting circuit lets you implement straightforward overvoltage protection. When the rectified output voltage is higher than 380V, the reference-to-anode voltage of IC_1 is higher than its internal reference of 2.495V, making the anode-to-cathode voltage drop to approximately 2V. In this situation, the cathode sinks the current across R_3, turning off Q_2.

When the rectified line voltage is lower than 380V, the cathode current of the TL431 is approximately 0A. Thus, Q_2 turns on through R_3, connecting bulk capacitor C and R_{LOAD} to the full-wave rectifier if $V_1=V_{IN}-V_{OUT}\leq 14V$.

All the components have a small power dissipation. The GP10NC60KD transistor, with an input of 230V rms and a load as high as 500W, is suitable for Q_2. EDN

Electronically tinge white-light source

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

White-light LEDs are becoming commonplace in everyday life. A classic white LED is an InGaN (indium-gallium-nitride) LED, emitting spectrally "pure" blue light. Photoluminescence spreads the spectrum of these LEDs and converts it into a light resembling daylight. This conversion takes place in a layer of yellow ZnSe (zinc selenide), which covers the InGaN chip. Due to production tolerances in the thickness of the ZnSe layer, these white LEDs are available in yellowish warm-white, neutral, and slightly bluish cool-white grades. To derive another hue from a white-light source, you can use the LED driver circuit in **Figure 1**.

The light source, IC_1, is an Avago Technologies (www.avagotech.com) ASMT-MT00 RGB (red/green/blue) LED. The driver contains two resistive DACs, IC_3 and IC_5, which function as potentiometers (**Reference 1**). The DAC's preset pins, PRE1 and PRE2, are grounded. Both resistive DACs are therefore set to midscale after power-on.

The true midscale for the IC_3 DAC at its wiper pin is one-third scale for IC_3, set by the resistor $R_{SH}=R_{A1B1}/2$, where R_{A1B1} is the resistance between the ends of the potentiometer. Voltage follower IC_{8A} ensures that the voltage at potentiometer IC_3 remains a constant reference voltage regardless of the position of wiper P_1. Voltage follower IC_{8A} is necessary because the resistance between the A1 and B1 terminals of IC_3 varies from 10 kΩ for wiper P_1, when grounded, to 3.33 kΩ for the A1 position of P_1. In this way, the drivers for the red, green, and blue LEDs generate the same value of current of $V_{REF}/3R_F$, where V_{REF} is the reference voltage. Thus, you get white light at power-on.

If you require, for example, a pale-pink hue, you ground the $\overline{PD1}$ pin for a short period. Wiper P_1 thus moves down by one step, decreasing the content of green light in the resulting light while increasing the contents from the red and blue LEDs. The sum of the I_R, I_G, and I_B currents remains constant, regardless of the positions of wipers P_1 and P_2. The luminous intensity of the output light holds constant. Any further short-term grounding of the $\overline{PD1}$ pin leads to a deeper violet hue of the output light. To get turquoise or bluish-green-tinted white light, you simply ground the $\overline{PD2}$ pin for short periods. The relative content of the red component then decreases below one-third of full-scale. If you ground the $\overline{PU1}$, $\overline{PD1}$, $\overline{PU2}$, or $\overline{PD2}$ pins for short periods, you can arbitrarily set hues of the light. The color resolution comes from adding or removing current in approximately 3% steps while removing or adding an equal number of approximately 3% steps of the remaining basic color components. A 100% step equals the total light intensity, regardless of color. This intensity is constant because the sum of currents flowing through the red, green, and blue LEDs is constant and has a value of V_{REF}/R_F. The resistive DACs have wiper-position margins.

The zero-scale relative margin is typically 1% of full-scale. The upper-position relative margin, or margin of the upper value of resistance between the B and the wiper terminals, is δ_V=2.4% of full-scale. Resistor R_{SH} artificially increases the upper margin of the V_{OUTG} voltage. The following **equation** yields the maximum settable voltage for V_{OUTG}:

$$V_{OUTGMAX}\simeq\frac{1}{1+3\delta_V}\times V_{REF}$$
$$\simeq(1-3\delta_V)\times V_{REF}.$$

By evaluating the **equation**, you determine you can set 92.8% green and subdivide the remaining 7.2% between the red and blue components by grounding $\overline{PU1}$ for a long time. If you also ground the $\overline{PU2}$ pin for more than 4 seconds, you get a yellowish- or warm-green color. In contrast, if you ground the $\overline{PD2}$ pin for more than 4 seconds, you get aqua or a cool-green color. Thus, changing even a moderate 7.2% of basic components of the light results in highly discernible hues.

Paralleling the R_{SH} between the B1 and P_1 terminals of resistive DAC IC_3 causes these terminals to exhibit nonlinear behavior. The step change of voltage at wiper P_1 decreases to two-thirds at the midscale of IC_3 and gradually rises when moving the wiper from the midscale toward zero. At zero, this step change recovers fully to its original relative value of 1/32. When moving P_1 from midscale toward full-scale, the step change rises and triples to a value of 3/32 at full-scale. This nonlinear behavior has, however, no detrimental effects. In contrast, close to the midscale, it makes the resolution 1.5 times that of the resistive DAC alone. EDN

REFERENCE

1 Štofka, Marián, "Set LEDs' hue from red to green," *EDN*, Oct 21, 2010, pg 59, http://bit.ly/a96DXg.

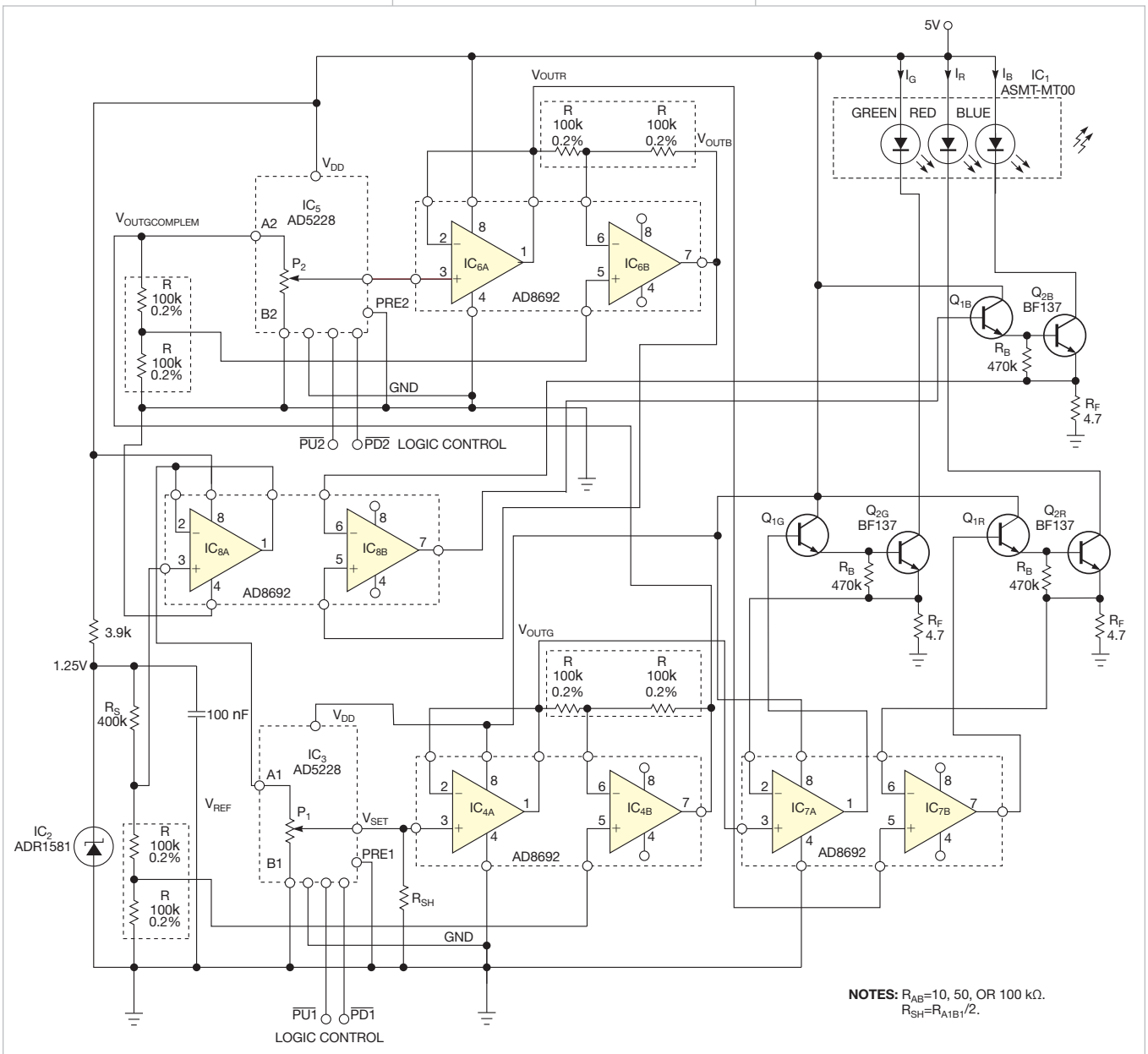


Figure 1 At power-up, the AD5228s are set automatically to their midscales, and the circuit produces white light. Short-term grounding of four control pins lets you tinge the light while holding intensity constant.

Transistor boosts regulator current

Yngve Linder, Örsundsbro, Sweden

Some circuits require a constant-current source that doesn't necessarily connect to a power-supply rail or to ground. The circuit in **Figure 1** shows a simple method for achieving that configuration.

The LM317 voltage regulator develops 1.25V between the OUT and the ADJ pins. Placing a resistor between those pins produces a constant current. Thus, the circuit's output current is $1.25V/R_{ADJ}$. The transistor lets the circuit source more current than the regulator alone can provide once the current through R_1 creates enough voltage to turn the transistor on. Otherwise, the regulator supplies the load current. EDN

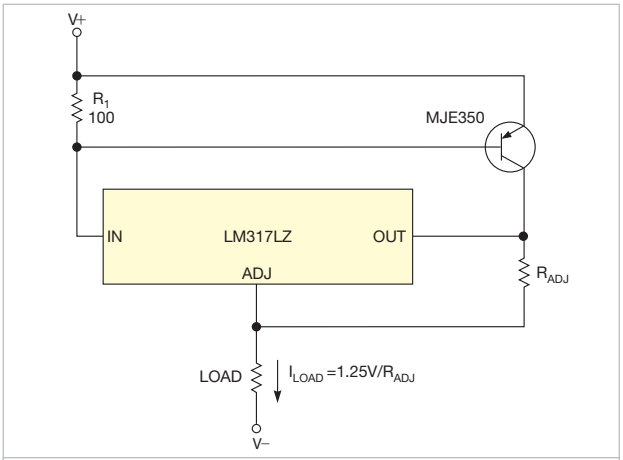


Figure 1 This simple method achieves a constant-current source that doesn't necessarily connect to a power-supply rail or to ground.

Detect live ac-mains lines

Raju R Baddi, Raman Research Institute, Bangalore, India

You can use a simple battery-powered circuit to detect whether an ac-mains wire is live without making any electrical contact with it. The circuit uses a CD4011 NAND gate's high input impedance to sense a magnetic field from a 50- or 60-Hz ac-mains line. You simply bring the detector coil near the socket to see whether it has a proper ac connection. If it does, then the LED will illuminate (**Figure 1**).

The detector in this case is a coil of copper wire. When you place it near a live wire carrying ac current, the coil develops a voltage across the CD4011 at pins 1 and 2. This voltage produces square waves at the output of the gate, driving the LED active. In the absence of any hot ac wire near the detector plate, the 1N4148 diode connected to the first gate's inputs keeps the gate biased. This bias ensures that, under normal conditions, the final output from the gates is low, keeping the LED off.

Placing the detector plate close to a live wire sets up an oscillating voltage at the gate's input at pins 1 and 2. That voltage produces square waves corresponding to the ac-mains frequency. The remaining three gates of CD4011 connect in parallel, which increases the current through the LED enough to light it.

A rechargeable, 3.6V nickel-cadmium battery powers the circuit. You can assemble the detector into a convenient, pocket-sized glue-stick tube (**Figure 2**). The circuit consumes nearly no power when the indicator LED is off. Thus, you can also power the circuit using lithium cells, such as the popular CR2032. EDN

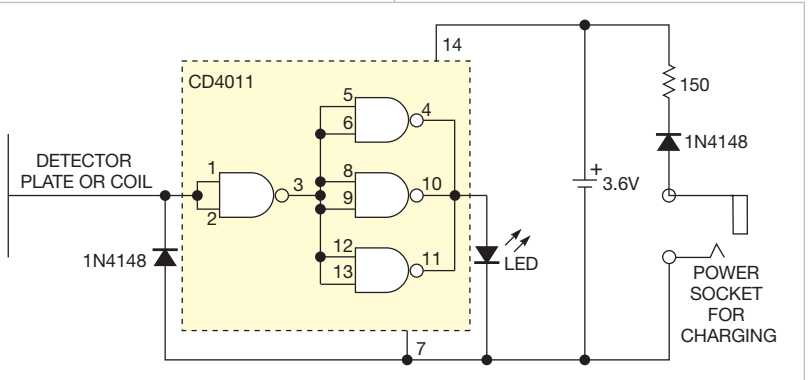


Figure 1 A battery-powered live-wire detector lights an LED in the presence of a field.

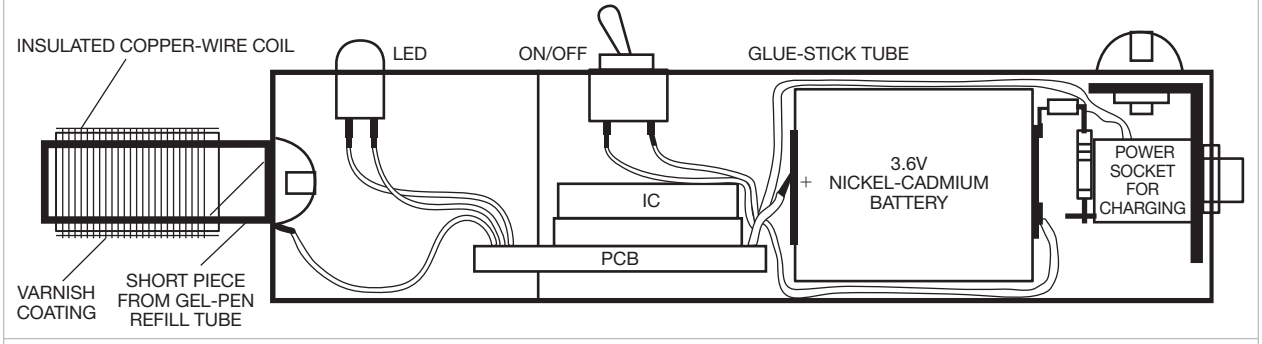


Figure 2 The circuit can fit into an empty tube of glue. Be sure to varnish the coil for best performance.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

CMOS circuit latches relays

JC Maillet, Gabriola Island, BC, Canada

Control applications often require that you set a relay latch in position until you need it to change state. Latching relays accomplish that task. When you send them a pulse, they either remain in the current state or change states, depending on the polarity of the pulse and the current state of the relay. The circuit in **Figure 1** switches the state of a DPDT (double-pole/double-throw) latching relay based on a pulse. It comprises a momentary-switch-to-step-voltage-signal generator, a differential-pulse converter, a relay driver, and a relay coil.

A momentary switch produces a step-voltage signal that drives the circuit. The circuit uses a simple pulldown switching action (push-on/release-off), such as the one comprising R_S, C_S, and S_2, or a flip-flop latching action (push-on/push-off), such as the one comprising IC_{1A}, IC_{1B}, R_1, R_2, C_1, and S_1. In the simple pulldown case, you can also add a debounce circuit. The pushbutton switches let you test the circuit before connecting it to another input source.

The differential-pulse converter comprises IC_{1C}, IC_{1D}, IC_{1E}, and IC_{1F}. The last two stages of the CD4069 hex inverter are self-biased in linear mode around $V_{DD}/2$, where V_{DD} is the drain-to-drain voltage and corresponds to Pin 14 of IC_1. The circuit takes a rise or a fall at IC_{1C} and converts it to opposing pulses of equal length at the outputs of IC_{1E} and IC_{1F}. The order of the pulses is synchronous with the edge direction at the input of IC_{1C}.

The output-driver stage buffers the voltage outputs of IC_{1E} and IC_{1F} to drive the relay coil. The op amps provide differential current dumping through the load without incurring substantial waste in idle mode. An LED-indicator circuit comprising R_4 and D_1 shows the orientation of the relay switch.

Assuming that the circuit powers up with C_1 uncharged, IC_{1A} and IC_{1B} always start off in a state in which R_1 sees V_{DD} on both sides. All inverter stages operate in digital mode except for IC_{1E} and IC_{1F}.

DIs Inside

▸To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

When you apply power to the circuit, these two stages self-bias around $V_{DD}/2$ and operate in linear mode. The op amps, wired as followers, also bias their outputs near $V_{DD}/2$. That action leaves the relay coil with a negligible offset/error voltage thanks to the matching of the devices in the hex-inverter IC and the op amps' high open-loop gain.

Stages IC_{1E} and IC_{1F} ac-couple through capacitors C_5 and C_3, respectively. The capacitors produce an impulse at the in-

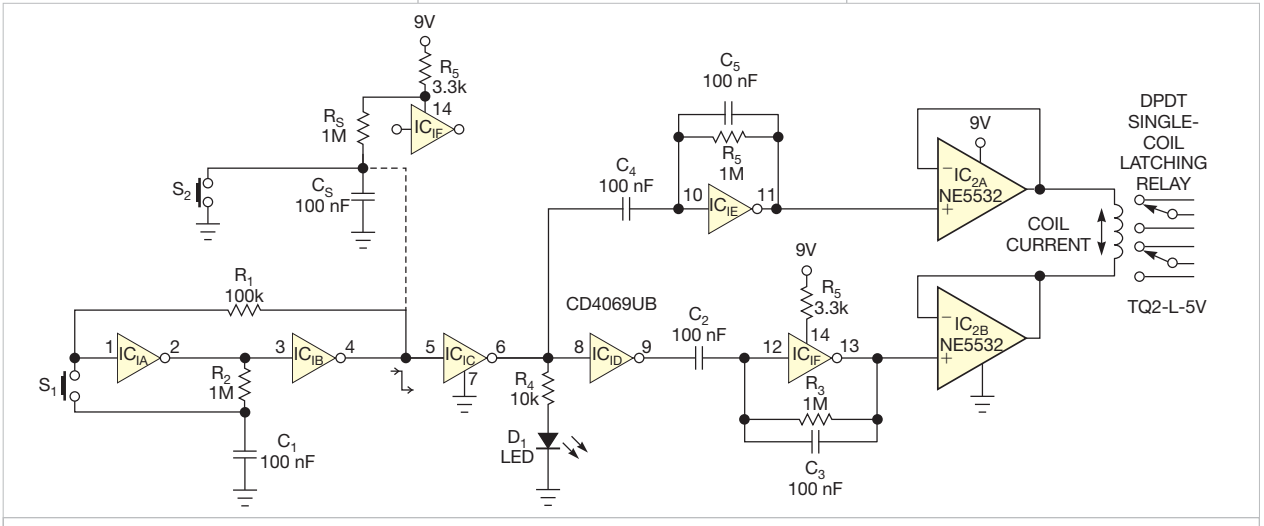


Figure 1 A change in state at the input of inverter IC_{1C} can drive a latching relay to change state.

puts of IC_{1E} and IC_{1F} from the step outputs of IC_{1C} and IC_{1D}. Lossy integrators IC_{1E} and IC_{1F} lengthen the inverted pulses at the outputs. Following these events, the outputs of IC_{1E} and IC_{1F} gradually return to their equilibrium state, which helps prevent a bucking field from forming in the relay coil and toggling back. **Figure 2** shows the shape and timing of the pulses.

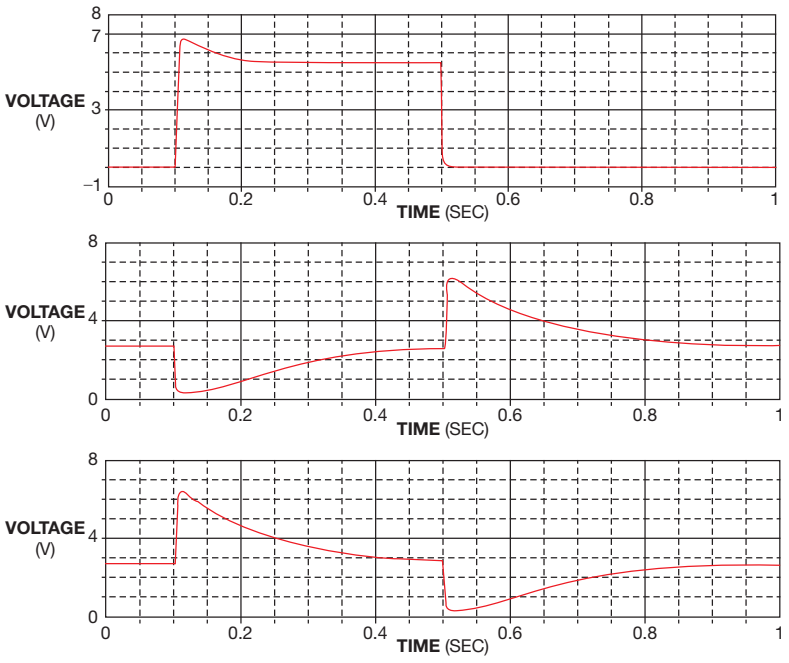


Figure 2 Integrating impulses with capacitors across inverters produces the signals necessary to set the state of the relay.

R_3/C_3 and R_4/C_4 establish time constants, which roughly set the total length of these pulse tails at 500 msec. This time is more than enough to satisfy the relay coil's hold requirements, which, for the Panasonic TQ2-L-5V, are 3 msec or less. If the relay coil at first finds itself in an asynchronous position, it can reorient itself after a single push of the switch if necessary.

Dropping IC_1's V_{DD} through series-supply resistor R_5 limits the idling currents in linear-biased stages IC_{1E} and IC_{1F}. This resistor represents a compromise between the power these two stages dissipate and the available voltage swing to toggle the relay through the op amps.

The circuit operates from a 9V source, and, with the component values in **Figure 1**, IC_1's V_{DD} lies at approximately 5.5V. The overall current draw between toggles is approximately 8 mA. EDN

Power USB devices from a vehicle

Fons Janssen, Maxim Integrated Products Inc, Bilthoven, the Netherlands

Automotive accessories such as PNDs (portable navigation devices) usually receive their power or charge using a simple adapter that a user plugs into a cigarette lighter. Sometimes, however, you may want to power or charge two devices at once. The circuit in **Figure 1** can handle that task.

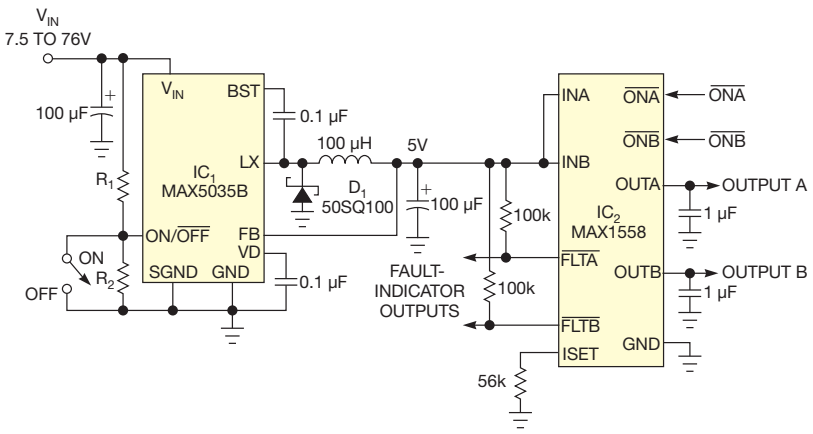


Figure 1 An automotive USB power supply generates two regulated, supply-voltage outputs from an unregulated input.

IC_1 generates 5V from any 7.5 to 76V input—a wide enough range to include the complete range of car-battery voltage plus the 40V spike that can occur during a load dump. The IC is simple to use because it has an internal power switch and requires no compensation circuit.

IC_2 distributes to two outputs the 5V that IC_1 generates. It not only distributes power but also protects against overload conditions. Most portable equipment receives its power or charge through a USB (Universal Serial Bus) interface, whose current limit is 500 mA. Because IC_2 targets use in USB applications, it latches off any port that tries to deliver more than 500 mA but does not affect the other port. Automatic-restart capability ensures that the port recovers automatically after the removal of the overload condition.

Figure 2 shows the protection feature in action. Output B has a constant load of 300 mA, and Output A switches between a 100-mA load and a 600-mA overload. IC_2 switches off Output A after an overload but allows a 20-msec delay to avoid responding to brief tran-

sients (**Figure 2a**). The circuit removes the overload 80 msec later; after another 20-msec delay, the automatic-restart feature brings Output A back online. Output B is unaware of the problem in Output A (**Figure 2b**). The fault-indicator output, however, goes low to indicate a problem in Channel A.

This circuit is small because it requires few external components. You can build it into a cigarette-lighter plug or place it in a small space behind the dashboard. EDN

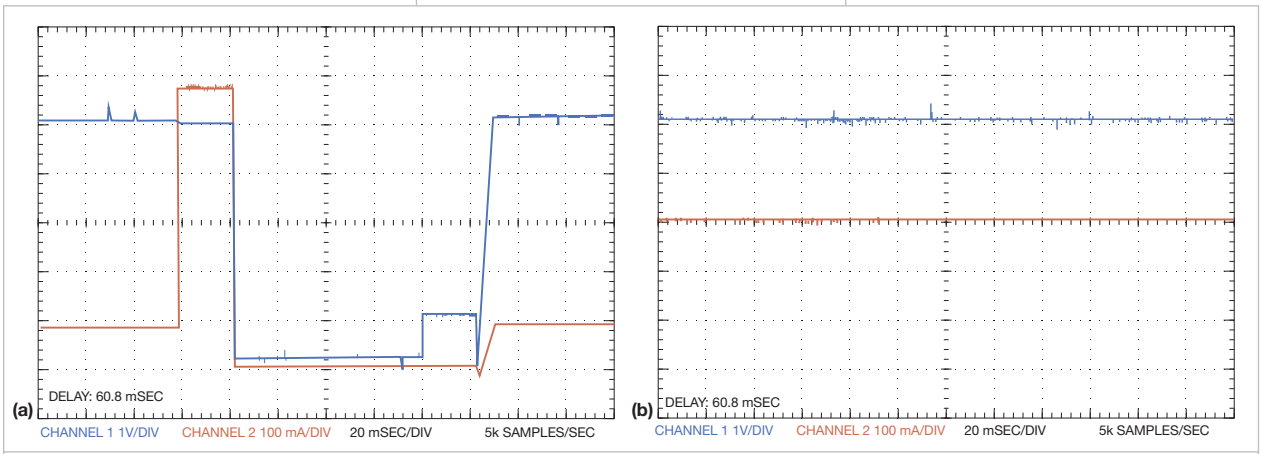


Figure 2 These current/voltage waveforms from Figure 1 show that an overload on Output A (a) has no effect on Output B (b).

Microcontroller interfaces to 24V

Adolfo Mondragon, Electrolux Products, Juarez, Mexico

Industrial-control applications often use PLCs (programmable-logic controllers) working at logic levels of 24V. That voltage creates a challenge to the safe use of a microcontroller. Such a design requires a physical barrier between the microcontroller and the 24V signals to avoid damage in case of errors or short circuits.

A simple and inexpensive way to switch 24V with a microcontroller is to use the ULN2003 or ULN2803 transistor drivers, which have seven and eight outputs, respectively. These ICs can energize light bulbs or solenoid valves at 500 mA. Damper diodes in these ICs eliminate the need for numerous passive components, especially in designs using coils.

Because of their digital inputs, passive components require numerous parts, complicating assembly, increasing cost, and increasing the need for troubleshooting and maintenance. Few simple ICs can handle more than 24V.

With this voltage in mind, you can use an interface IC, such as the MC1489 or the SN75189 inverter, as an RS-232 line receiver. These ICs can receive digital signals as large as ±30V. As a bonus, they have some hysteresis-level transition, making them able to discriminate some of the electrical noise in signals (**Figure 1**).

You can connect these devices directly to a microcontroller. If you mount them in DIP sockets, you can easily replace

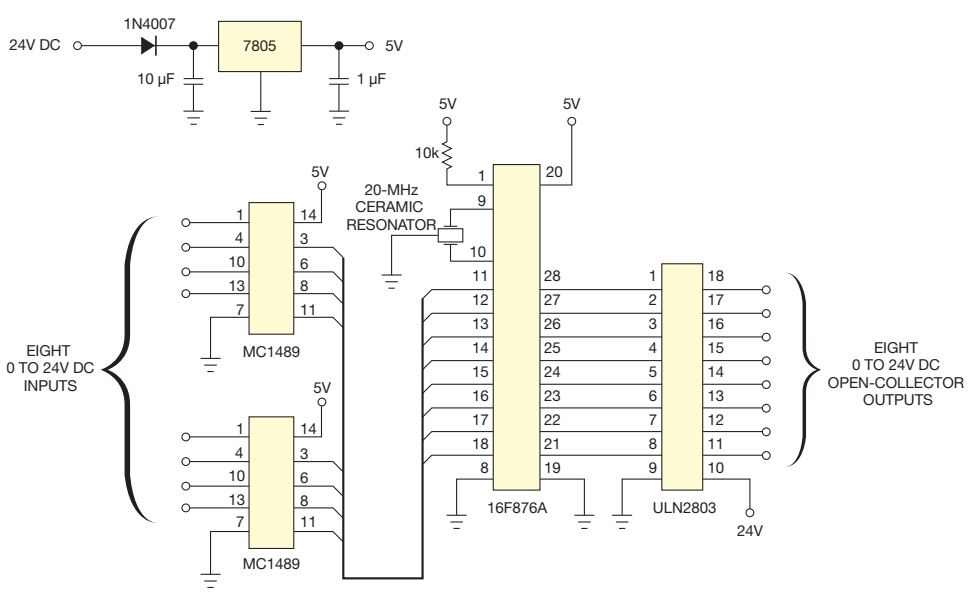


Figure 1 RS-232-interface ICs, such as the MC1489, can adapt 24V inputs to a microcontroller.

them in case of damage. The circuit uses a 78L05 linear regulator to decrease the power-supply voltage to 5V. The 1N4007 between the 24V dc and the 5V LM7805 regulator protects the circuit from possible power-supply wire reversal. You must take that fact into account in programming your microcontroller because the signals complement the input. EDN

Use LEDs as photodiodes

Raju R Baddi, Raman Research Institute, Bangalore, India

The simple circuit in **Figure 1**, which can be powered with a 3.6V nickel-cadmium rechargeable battery, lets you use an LED to detect light. The circuit consumes practically no quiescent power. Two LEDs act as photodiodes to detect and respond to ambient light. When ambient light is present, the upper LED, a small, red, transparent device covered with a black pipe, has a higher effective resistance than the lower, large, green LED. The voltage drop across the input of the NAND gate is less than its threshold voltage for logic 1, making the output of the NAND gate low. When the ambient light goes off, the voltage drop across the reverse-biased green LED increases, forcing the NAND gate's output high.

This type of light detector is highly power-efficient and is ideal for battery applications. You can use the NAND gate's logic output to drive an LED driver or a relay driver, or you can connect it to a microcontroller.

Place the circuit so that sufficient light falls on the green sensor LED. Doing so avoids any voltage buildup near the junction that could be close to the NAND gate's threshold voltage. The NAND gate's power consumption rises sharply at the threshold voltage. When the gate's input voltage is within the defined limits for the logic state, its power consumption is extremely small. EDN

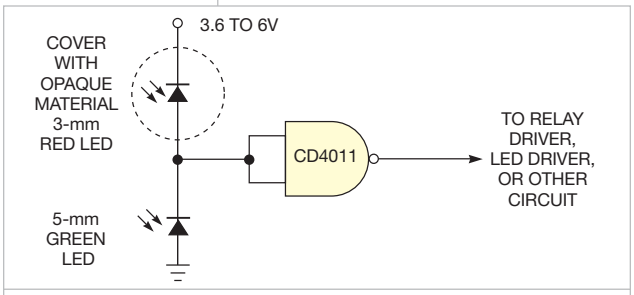


Figure 1 An LED's resistance changes with ambient light, which changes a voltage that drives a logic gate.

Microcontroller drives DSEC motors

Charaf Laissoub, Valeo Interior Controls, Créteil, France

DSEC (digital-satellite-equipment-control) motors find wide use in TV-satellite reception; thus, they are readily available. Eutelsat (www.eutelsat.com) defined the DSEC control protocol, which has been in existence since 1998. DSEC motors offer a resolution angle as high as 0.1°. Thus, you can use them as low-cost alternatives to stepper motors.

The circuit in **Figure 1** is a simple design to drive protocol Version 1.2 of DSEC motors using a PIC10F200 microcontroller from Microchip (www.microchip.com). Version 4.2 fully describes the bus-functional method of data-bit signaling (**Reference 1**; see www.edn.com/101118dia). You derive the 22-kHz-frequency tone from the internal 4-MHz±1% clock. Position-er commands from the protocol specification suit one-way communication. This application requires no receive responses or data messages from the remote motor unit. The long-term recommendation for the dc supply is 12V±1V, and the maximum current is 400 mA.

The circuit uses just one pushbutton switch, S_1, and two LEDs to control the state of the motor. At power-on, LED_1 remains continuously on. One 1-second-long push on S_1 blinks LED_1 for 0.25 second and drives the motor one step. A 1- to 2-second push on S_1 changes the state of direction. LED_1 turns off, and LED_2 turns on. Pressing S_1 for more than 3 seconds causes one LED to blink during the time necessary to drive the motor back to 0°.

This design uses a motor with a resolution angle of 0.3° and maximum angles of 75° east and 75° west. So this circuit has only 250 pulses for each direction. Once it reaches that value, the active LED continues to blink.

You can download fully commented, three-file assembler source code from the online version of this Design Idea at www.edn.com/101118dia. You can adapt this code to any other baseline family of Microchip microcontrollers that use 12-bit instruction code. EDN

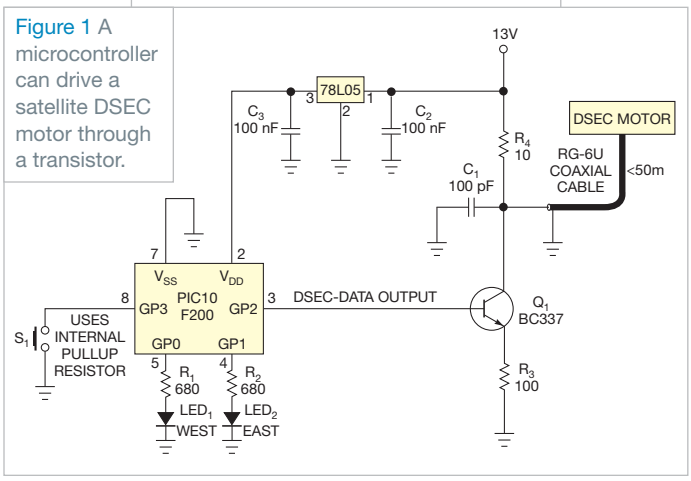


Figure 1 A microcontroller can drive a satellite DSEC motor through a transistor.

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

LEDs light LAN-cable tester

Noureddine Benabadji, University of Sciences and Technology, Oran, Algeria

Cable and connector faults are relatively common in LANs (local-area networks). You can use the circuit in **Figure 1** to test straight- or cross-link 10BaseT, 100BaseT, or gigabit UTP (unshielded-twisted-pair) and STP (shielded-twisted-pair) cables. The circuit performs a continuity test for each linked pair. An LED representing each linked pair flashes when the corresponding pair connects properly between the cable's RJ-45 connectors. It can also determine whether the cable is a straight-link or a cross-link type.

IC_1, a small, six-pin Microchip (www.microchip.com) PIC10F200 microcontroller, performs the test. On power-up, the four I/O pins act as outputs and are driven high for approximately 0.5 second and then are driven low in the following sequence: GP0, GP1, GP2, and GP3. Once this task finishes, the microcontroller has a dead time for about 4 seconds, after which it loops to the beginning to repeat the procedure.

Adding a low-power BS170 N-channel MOSFET allows Pin 6 (GP3), normally an input-only pin, to act as an output, according to a previous Design Idea (**Reference 1**). The power-supply voltage of 5V is not critical, and you can choose any supply from 2 to 5.5V for the 8-bit PIC microcontroller. **Figure 2** shows cable wiring for both types of ca-

DIs Inside

▸See all of *EDN*'s Design Ideas at www.edn.com/designideas.

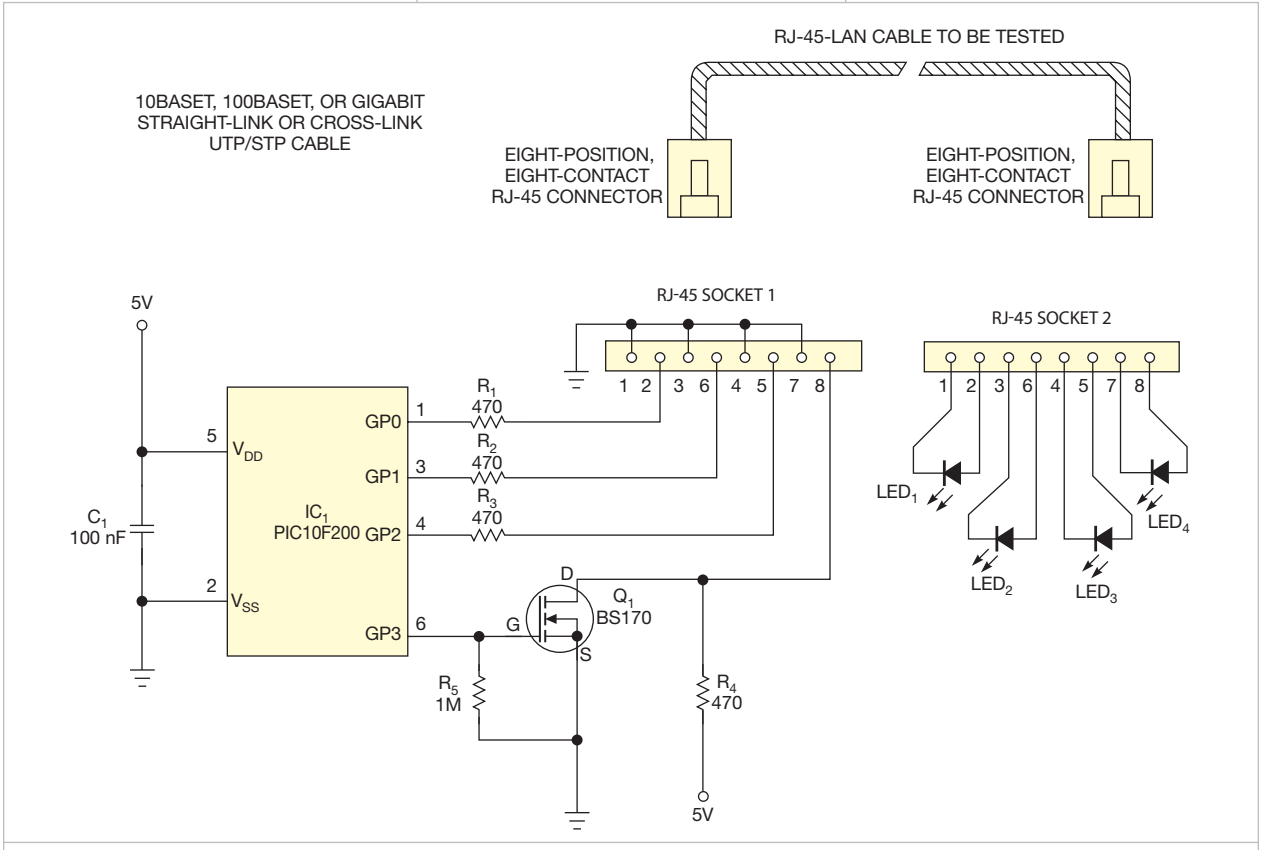


Figure 1 You can use this circuit to test straight- or cross-link 10BaseT, 100BaseT, or gigabit UTP and STP cables.

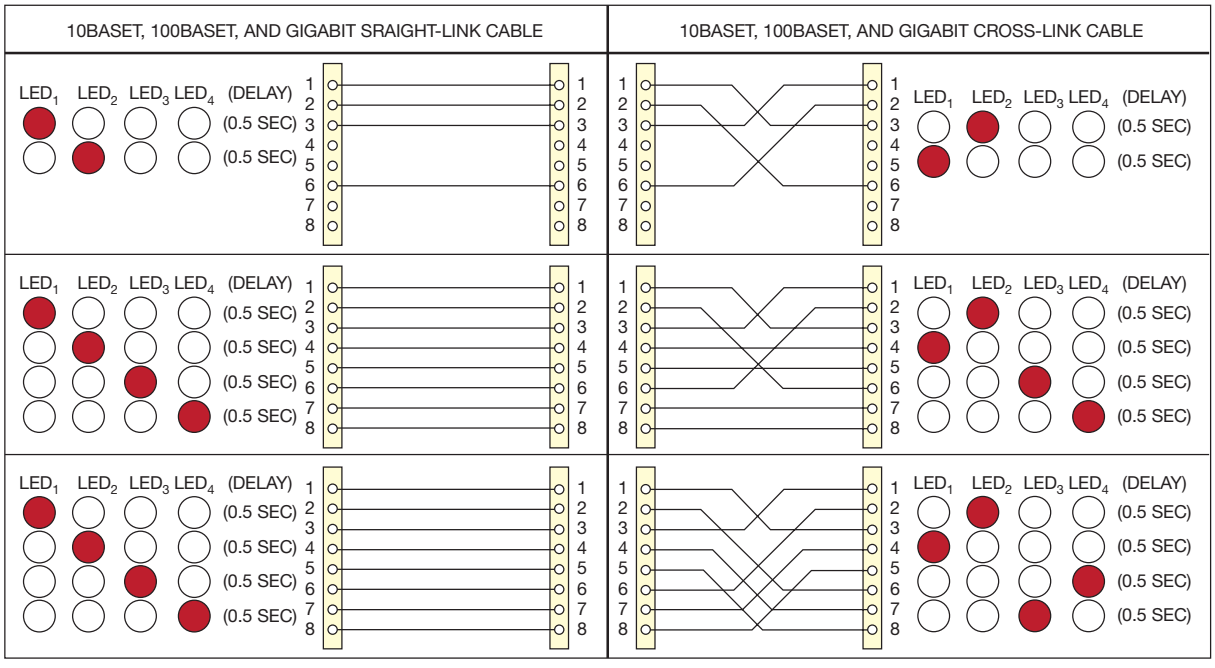


Figure 2 Cable wiring for both straight-link and cross-link cable fits with RJ-45 wiring.

bles and how they fit with RJ-45 connectors. For each type, the microcontroller initiates lighting. You can mount the RJ-45 socket in a separate small box to test the inside wall's long cables. The source code is annex at the end of this section.

It uses fewer than 256 words of memory. Because it does not specifically target the PIC10F200, it is easy to understand and adapt to other microcontrollers. EDN

REFERENCE

1 Muñoz, Antonio, "Use the MCLR pin as an output with PIC microcontrollers," *EDN*, Jan 10, 2008, pg 65, www.edn.com/article/470769-Use_the_MCLR_pin_as_an_output_with_PIC_microcontrollers.php.

Instrumentation amplifier is less sensitive to resistor mismatch

Reza Moghimi, Analog Devices, San Jose, CA

This Design Idea offers a new instrumentation-amplifier topology that improves ac CMR (common-mode rejection). Because it uses discrete amplifiers, you can customize it for the lowest power, price, noise, and supply voltage, depending on your application's requirements. Previously, instrumentation amps using discrete components had poor CMR. System designers still build their own using discrete designs because standard integrated instrumentation amps don't meet their design requirements, have undesired package options, or are too expensive. You can build discrete instrumentation amps with two or three op amps and a few resistors (**Figure 1**). Most monolithic instrumentation amps, however, use the three-op-amp configuration. This approach provides the best ac and dc CMR. A big challenge when building discrete instrumentation amps is to achieve CMR on par with that of monolithic instrumentation amps.

The poor CMR of the three-op-amp topology (**Figure 1b**) is due to resistor mismatching. Any common-mode signal at V_{IN1} and V_{IN2} appears at the outputs of A_1 and A_2. The difference amplifier, comprising R_1, R_2, R_3, R_4, and A_3, should reject this common-mode voltage. To achieve high CMR, this difference amplifier requires matched resistors and the A_3 op amp, which specifies high CMR. With 0.1%-resistor matching, the best possible CMR at dc is 54 dB. This CMR further degrades over frequency, depending on the op amp you select.

The poor CMR in two-op-amp instrumentation amps is due to the unequal phase shift that occurs at the two inputs of A_2. Signal V_{IN1} must travel through amplifier A_1 before amplifier A_2 subtracts it from V_{IN2}. Thus, the voltage at the output of A_1 becomes delayed with respect to V_{IN1}. Amplifier A_1 introduces the delay that causes its output to lag behind the directly applied input voltage at V_{IN2}. This phase difference results in an instantaneous difference in A_1's output and V_{IN2}, even if the amplitudes of both voltages are equal. This difference causes a frequency-dependent common-mode error voltage at V_{OUT}. Further, this ac common-mode error increases with common-mode frequency because the phase shift through A_1 increases with frequency due to the single-pole roll-off frequency response.

Instrumentation-amplifier-design guidelines give the error relationship,

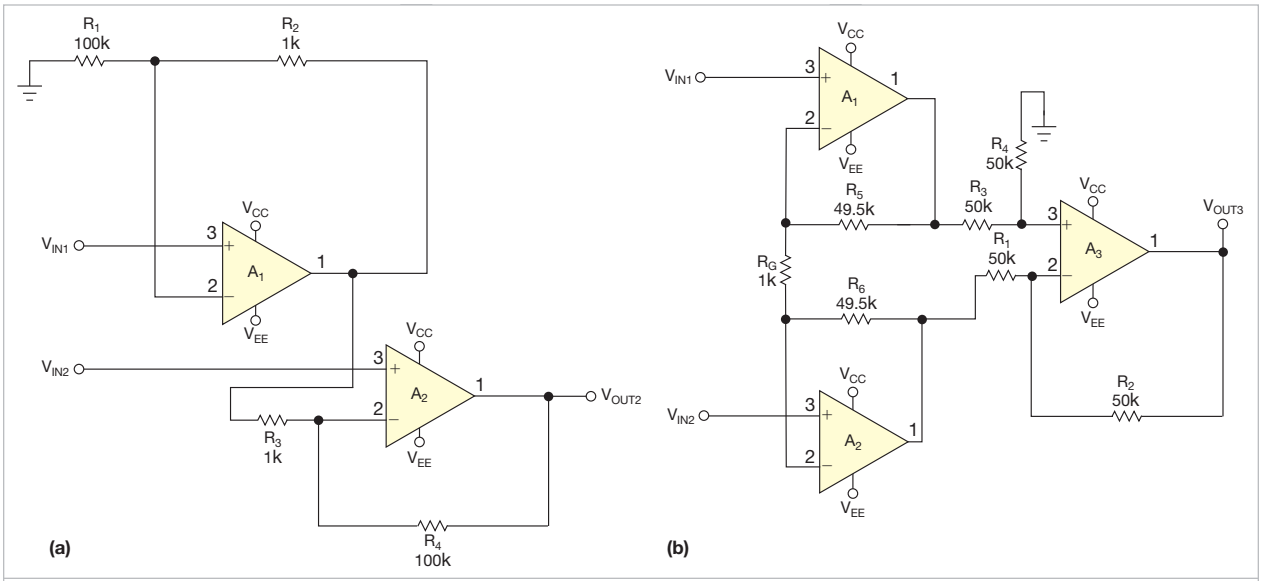


Figure 1 You can use conventional two-op-amp (a) and three-op-amp (b) configurations to build an instrumentation amplifier.

according to the following **equation**: (100 Hz/100 kHz)×100%=0.1%, when you design an instrument amplifier for a closed-loop bandwidth of 100 kHz by using micropower op amps with a unity-gain bandwidth of 500 kHz configured for a gain of five (**Reference 1**). A common-mode error of 0.1% is equivalent to a 60-dB CMRR (common-mode-rejection ratio). Even if you trim this circuit to achieve 100-dB CMRR at dc, this performance would hold only for frequencies less than 1 Hz. At 100 Hz, the CMRR could not be better than 60 dB.

You can overcome the shortcomings of the two-op-amp circuit if you compensate for the phase delay of A_1 using active-feedback compensation. You can reduce phase error to virtually zero at low frequencies with a dual matched op amp (**Reference 2**). This approach adds an equal but opposite phase shift in the amplifier's feedback loop. Because the circuit depends on amplifier matching, you must use a dual- or quad-op-amp part.

The active feedback requires an extra op amp and two external resistors to achieve phase-error cancellation (**Figure 2**). In the circuit, amplifier A_1 provides forward gain of the composite amplifier. K_1 determines the closed-loop gain ($A_V=1+K_1$), where A_V is voltage gain and K is a constant. Amplifier A_2 provides feedback to amplifier A_1. K_2 determines the amount of phase-error compensation and has no effect on the forward gain of the composite amplifier. Optimum error cancellation occurs when K_1 is equal to K_2. The error terms are functions of complex frequency response, according to the following **equations**:

$$A_{ERROR}=\left(\frac{\omega_{CLOSED\_LOOP}}{\beta\times\omega_{VGB}}\right)^2;$$

$$\phi_{ERROR}=\left(\frac{\omega_{CLOSED\_LOOP}}{\beta\times\omega_{VGB}}\right)^3.$$

Figure 3 shows the new configuration, which applies an active feedback network to a two-op-amp configuration.

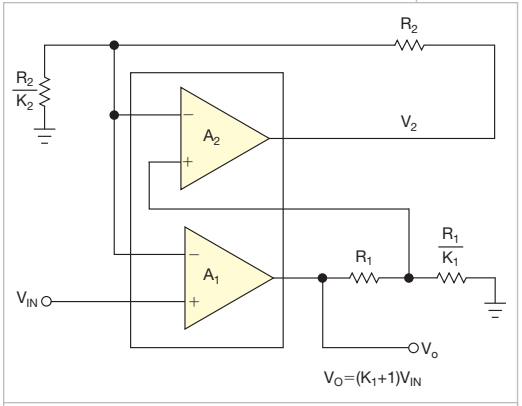


Figure 2 Using A_2 in an active-feedback network greatly improves A_1's effective bandwidth. The op amps must match, so you should use a dual-amplifier package for this circuit.

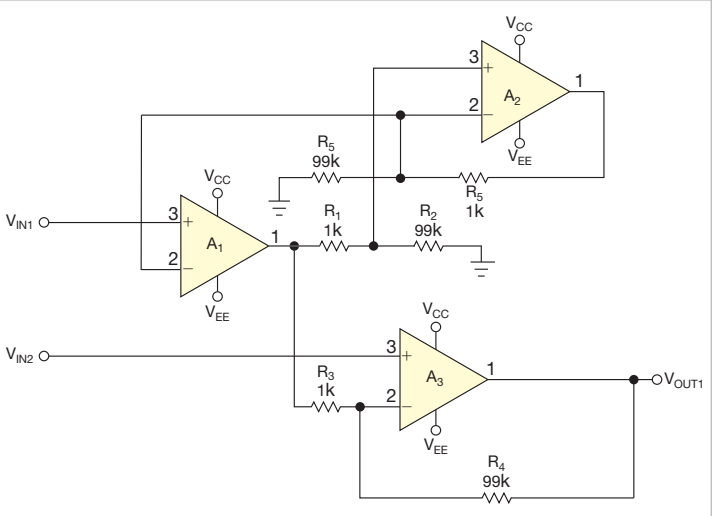


Figure 3 The three-op-amp instrumentation amplifier employs a two-amplifier topology; a third amplifier provides active-feedback phase compensation.

You correct the phase delay through A_1 using the active-feedback circuitry comprising A_2 and four resistors. The classic two- and three-amplifier instrumentation amplifiers and this proposed three-amplifier topology using active feedback have similar dc CMRRs. However, the circuits' ac CMRs differ. This difference in performance arises when you apply and sweep a large ac common-mode voltage at the input. These test specs are for a circuit using Analog Devices' (www.analog.com) AD8603 op amps. The supply voltage is ±2.5V, the common-mode voltage is 0.001 mV, and the gain is 100. You take error measurements with a 500-Hz input signal.

You then change the operating conditions to a common-mode voltage of 2.001V and a gain of 100. The circuit in **Figure 3** provides great improvement over the circuit in **Figure 1a**. The circuit has less sensitivity to resistor mismatch than the circuit in **Figure 1b** with perfectly matched resistors. (For more data, go to www.edn.com/101202dib.)EDN

REFERENCES

1 Kitchin, Charles, and Lew Counts, *A Designer's Guide to Instrumentation Amplifiers*, Chapter 2, Analog Devices, 2000, pg 2-1, http://bit.ly/b0rXnT.

2 Wong, James, "AN-107 Active Feedback Improves Amplifier Phase Accuracy," Analog Devices, September 1987, http://bit.ly/a8Ol38.

Circuit keeps wandering children and pets nearby

Tom Au-Yeung and Craig Sakamoto, Maxim Integrated Products, Sunnyvale, CA

The receiver circuit in **Figure 1** sounds an audio alarm when the transmitter (**Figure 2**) moves beyond a designated perimeter. The transmitter, a voltage-controlled oscillator, operates at approximately 915 MHz in the unlicensed ISM (industrial/scientific/medical) band. It has a tuning voltage of $1.5V=3\times R_2/(R_1+R_2)$, which lets you easily adjust the frequency by varying the values of resistors R_1 and R_2.

The receiver comprises low-noise amplifier IC_1, power detector IC_2, comparator IC_3, and a buzzer. When the transmitter is within range—for example, when a child or a pet is carrying it—the receiver detects the RF signal and provides a voltage greater than 400 mV at the inverting terminal of the comparator. Resistors R_9 and R_{10} preset the reference voltage at the comparator's noninverting terminal. The reference voltage is $3\times R_{10}/(R_9+R_{10})$, and the comparator's output remains low.

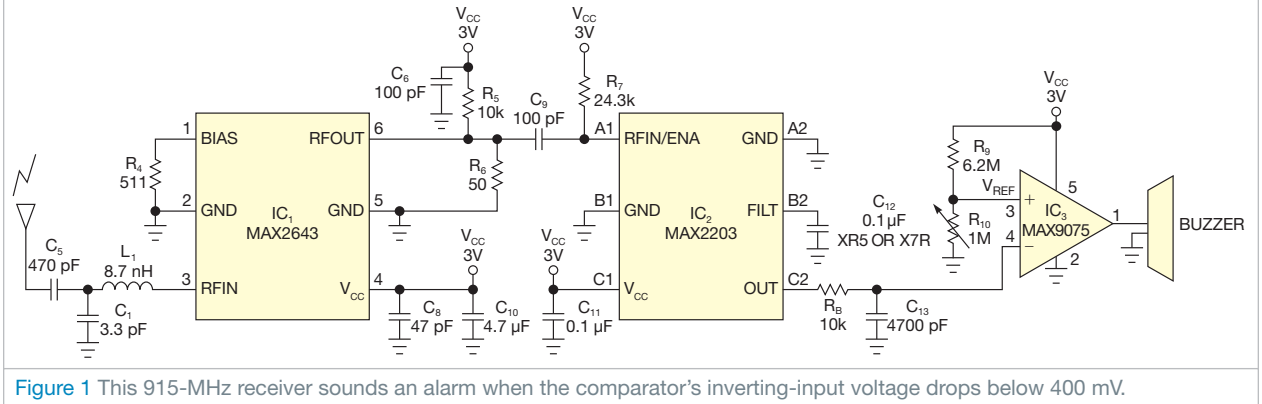


Figure 1 This 915-MHz receiver sounds an alarm when the comparator's inverting-input voltage drops below 400 mV.

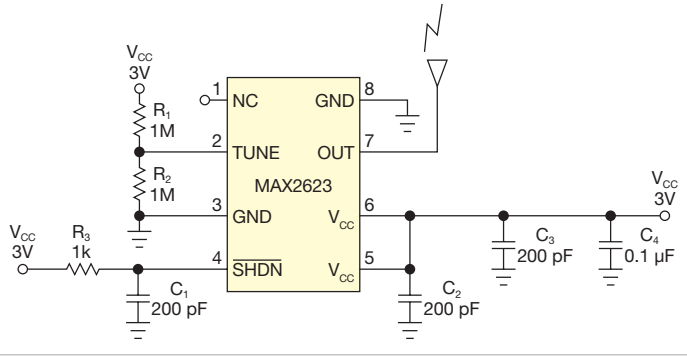


Figure 2 The transmitter comprises a voltage-controlled oscillator, which R_1 and R_2 tune to approximately 915 MHz.

When the transmitter moves outside the predetermined boundary, the detected RF produces less than 400 mV at the comparator. The comparator then generates an output of approximately 3V, which turns on the buzzer and sounds an alert that the transmitter has moved beyond the restricted perimeter. To increase the detection range, you can place additional low-noise amplifiers or VGAs (variable-gain amplifiers) in front of the power detector. You can also increase or decrease the desired perimeter by adjusting R_{10} to change the comparator's reference voltage.EDN

LEDs light LAN-cable tester source code

```
; Listing1.ASM: RJ45 LAN Cable Fault Tester
; BENABADJI Noureddine - ORAN - Sep. 04th, 2010
;
        List P = 10F200
#include <p10F200.inc>
;       List P = 10F202
;#include <p10F202.inc>
;       List P = 10F204
;#include <p10F204.inc>
;       List P = 10F206
;#include <p10F206.inc>

        __CONFIG _MCLRE_OFF & _CP_OFF & _WDT_OFF & _IntRC_OSC

#define  PAIR12  GPIO, 0        ; I/O line output to test pair 1-2
#define  PAIR36  GPIO, 1        ; I/O line output to test pair 3-6
#define  PAIR45  GPIO, 2        ; I/O line output to test pair 4-5
#define  PAIR78  GPIO, 3        ; I/O line output to test pair 7-8

;----------------------------- define variables ------------------------------

        CBLOCK0x10              ; Beginning of RAM (PIC10F200/204), 16 bytes
;       CBLOCK0x08              ; Beginning of RAM (PIC10F202/206), 24 bytes

        cnt1, cnt2, cnt3 ; counter1, counter2, counter3

        ENDC

;////////////////////////////////////////////////////////////////////////////
;       MAIN PROGRAM
;////////////////////////////////////////////////////////////////////////////
        ORG  0
        GOTO  Setup
;////////////////////////////////////////////////////////////////////////////

;-------------------------------------------------------------------------
Delay0s5
        CLRF    cnt2
        CLRF    cnt1
        GOTO    $+1             ; 2us
        GOTO    $+1             ; 2us
        NOP                             ; 1us
        DECFSZ cnt1, F  ; 1us
        GOTO    $-4             ; 2us => 8*256 = 2048 us, approx. 2 ms internal delay loop
        CLRWDT
        DECFSZ cnt2, F  ; external loop = 2 * 256 = 512 ms, approx. 0.5 s
        GOTO    $-7
        RETLW  0
;-------------------------------------------------------------------------
Delay4s
        MOVLW           .8
        MOVWF           cnt3
        CALL    Delay0s5
        DECFSZ cnt3, F  ; 8 * 0.5s = 4s
        GOTO    $-2
        RETLW  0

;////////////////////////////////////////////////////////////////////////////
Setup:
        MOVWF           OSCCAL          ; trim the internal oscillator

        MOVLW           b'00001000'
        TRIS    GPIO            ; Config. GP0, GP1 & GP2 as outputs
        CLRF    GPIO            ; all LEDs off

;;;;;for 204/206 only
;       MOVLW  b'11110111'    ; bit3:CMPON=0 => comparator OFF, ...
;       MOVWF  CMCON0                   ;...to minimize power consumption while in Sleep mode

        MOVLW  b'10001000'    ; no wake-up, <6>=0 pull-up, <5>=0(use GP2, not TOCKI)
        OPTION
test
;test pair 1-2
        BSF             PAIR12          ; activate pair 1-2 for ...
        CALL    Delay0s5        ; ... about 0.5 s
        BCF             PAIR12

;test pair 3-6
        BSF             PAIR36          ; activate pair 3-6 for ...
        CALL    Delay0s5        ; ... about 0.5 s
        BCF             PAIR36

;test pair 4-5
        BSF             PAIR45          ; activate pair 4-5 for ...
        CALL    Delay0s5        ; ... about 0.5 s
        BCF             PAIR45

;test pair 7-8
        MOVLW  b'11001000'    ; <6>=1 no pull-up => FET off => activate pair 7-8
        OPTION
        CALL    Delay0s5        ; about 0.5 s
        MOVLW  b'10001000'    ; <6>=0 pull-up => FET on => deactivate pair 7-8
        OPTION

;4s deadtime, then repeat the procedure
        CALL    Delay4s
        GOTO    test

        END
```

EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

High-side current-shunt monitor offers reduced error

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

The circuit in **Figure 1** is an alternative to a high-side current monitor in a recent Design Idea (**Reference 1**). That monitor uses the Analog Devices (www.analog.com) AD8212 and an external high-voltage bipolar PNP transistor. The AD8212 can compensate for errors, which can reduce from the approximately –1% error of an uncompensated circuit to about –0.4%.

Circuit errors occur mainly because of the finite current gains of the two bipolar PNP transistors in the circuit: an external transistor and an internal low-voltage PNP transistor in the AD8212. The internal PNP transistor's base-emitter junction forms a negative-feedback loop for the op amp within the AD8212. Both PNP transistors form a cascade of two common-base-operated transistors. In the ideal case, the emitter current of the internal PNP, which is proportional to a sensed current, should equal the collector current of the external PNP. This collector current mediates the information about the sensed current. In practice, however, the collector current of the external PNP transistor equals the emitter current of the internal PNP minus the sum of the base currents of both PNP transistors.

The base current is also a source of error in this circuit. The circuit reduces the undesired base current of the Darlington PNP by a factor of one divided by β_{PNP} compared with the circuit in the earlier Design Idea. In that Design Idea, β_{PNP} is the current gain of one PNP transistor. The circuit in **Figure 1** reduces error by using a PNP-to-Darlington connection in place of an external PNP transistor. The difference between the emitter and the collector currents in the Darlington connection is so low that you can omit compensation circuitry and the internal PNP transistor, which are associated with the compensation circuit. You could thus integrate the two 1-kΩ resistors and the zener diode into a monolithic IC that is simpler than the AD8212.

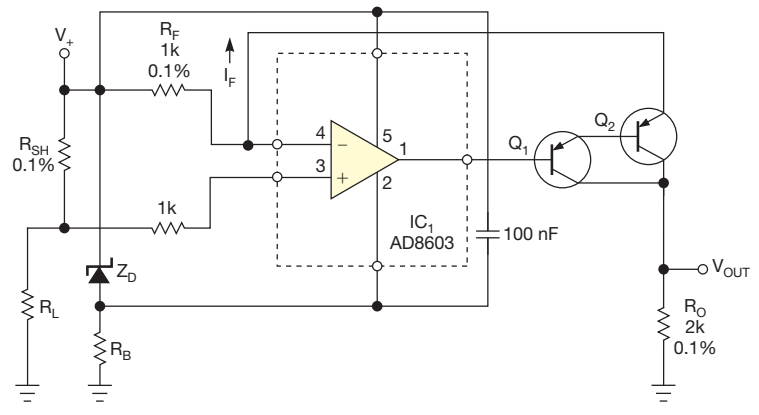


Figure 1 The circuit senses the current flowing through load resistor R_L at the high side and transfers it directly to the low side by means of feedback current I_F, which has 500 µA at full-scale.

The circuit in **Figure 1** uses an Analog Devices AD8603 op amp, which has a 40-µV input offset voltage. When its input voltage is close to the upper supply rail, the offset voltage is less than 200 µV. The worst-case input offset-voltage value would cause an additive error of 0.04% of the full-scale because the full-scale is 500 mV. IC_1's subpicoampere input bias current rises at elevated temperatures to about 320 pA at 125°C, but that increase is still not significant enough to affect circuit accuracy. The same holds true also for the leakage current of the Darlington connection because the leakage currents flowing through the emitter and the collector of Q_2 have almost the same value. Leakage current I_{CE0} becomes a part of the feedback current that flows through resistor R_F.

When I_{CE0} rises, the op amp's output voltage goes slightly more positive. Feedback current I_F, flowing through resistor R_F, still remains constant. The only condition is that the minimum feedback current must be larger than the maximum leakage current. The selected PNP transistors allow V_+ to be as high as 30V. Q_1, an MMBT3906 type, exhibits a low drop in the value of current gain at low emitter currents. It

DIs Inside

▸To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

drops to just 75% of its maximum value of 130 at an emitter current of –100 µA. Q_2 is an MMBT4403 type.

For applications requiring higher values of V_+, select PNP transistors having sufficient collector-emitter voltage ratings and increase the value of R_B as $R_B=(V_+–5V)/5\times10^{-4}A$. Z_D, a ZPY5V6, has a zener voltage of about 4.7V at about 500 µA. A test reveals that the relative difference between the emitter and the collector currents of the Darlington pair doesn't exceed 0.06% at full-scale. At 0.01 times full-scale, the relative error rises to 1.77%, indicating that the overall current-gain value of the Darlington decreases to about 56. The reduction of error lets you reduce the full-scale voltage at shunt resistor R_{SH} to 250 mV, reduce the power dissipated in R_{SH} by 50%, and maintain the error at 0.15%. EDN

REFERENCE

1 Tran, Chau, and Paul Mullins, "Current monitor compensates for errors," *EDN*, Sept 9, 2010, pg 47, http://bit.ly/aFnEBW.

Make a quick-turnaround PCB for RF parts

Steve Hageman, AnalogHome.com, Windsor, CA

Using low-cost PCBs (printed-circuit boards), you can easily design a board in a few hours with nearly any CAD package, even the free ones. You can have your prototype board on your desk in just two days. The design rules in many software packages are good, and most suppliers can fabricate a PCB with line width and spacing down to 0.006 in.

That precision is fine for low-frequency circuits, but RF circuits usually need 50Ω traces for proper circuit operation. Parts get smaller, but the laws of physics don't change. Thus, a microstrip trace on a 0.062-in.-thick standard prototype board that was calculated to be 0.11 in. wide 30 years ago is still 0.11 in. wide today. Many surface-mount parts are far smaller than their predecessors, however, so it would seem that low-cost, two-layer prototype boards for RF prototyping are unsuitable for today's small SMT (surface-mount-technology) parts.

You can use a CPWG (coplanar-waveguide-over-ground) structure to build 50Ω RF traces on PCBs. A CPWG structure lets you make the required trace width smaller than that of a microstrip structure.

Bringing a grounded copper ground plane on the top of the board closer to a microstrip trace adds capacitance to the microstrip structure. To compensate and to keep the entire structure at 50Ω, you must make the center trace more inductive by reducing its width—to a point.

How can you design the CPWG structure for a low cost and a fast PCB process? You can find many online CPWG calculators, but they often fail when the ground-plane gap gets less than approximately 30 to 50% of the trace width because the height of the copper traces on the board becomes a significant factor. It adds more capacitance than the calculators assume. Hence, the lines these calculators design have too much capacitance, which reduces their impedance to less than 50Ω. The equations date back many years to IC design.

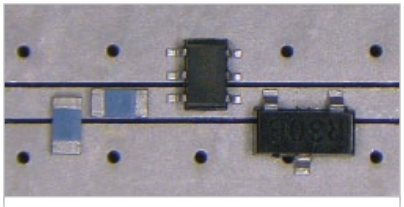


Figure 1 A nominally small SMT part fits well onto the 0.032-in.-wide CPWG 50Ω line structure. The 0603 resistors and capacitors and a small gallium-arsenide FET-amplifier SC-70 IC also fit well.

The equations in many calculators fall apart because today PCBs differ physically from ICs. The best way to properly design a CPWG on a PCB with a narrow gap-to-center-trace ratio is to use a full 3-D electromagnetic simulator. This Design Idea provides the values for a few common structures.

In keeping with the minimum trace-to-trace spacing of 6 mils, I simulated, built, and tested a CPWG structure. For a common 0.062-in.-thick FR-4 PCB material, a trace width of 0.032 in. with a gap of 0.006 in. is as close to 50Ω as you can get. It provides better than 40-dB return loss on the trace at 6 GHz.

This approach is better than using a 0.11-in.-wide trace and is compatible with SMT-sized parts. A 0603-sized SMT part and a common SMA (surface-mount-assembly) edge-launch connector fit the line perfectly. **Figure 1** compares several common RF-type parts with the fabricated PCB. For parts with larger pad dimensions than the 0.032-in. trace width, just increase the spacing to the top ground plane to compensate. For instance, increase the spacing to the top plane of a 0805 SMT pad to approximately 0.008 in. and increase the top-plane spacing for a 1206 SMT-component pad to 0.012 in. to keep the pad from being too capacitive.

In keeping with common design rules, I pulled back the copper planes on the tested PCBs 0.01 in. from the routed board edge. This pull-back and the edge-launch connector both add a slight amount of inductance to the transition, however. The big center pin of the edge-launch connector on top of the trace adds extra capacitance, providing built-in capacitive compensation. Cutting the pin to about half its original length yields about equal capacitance to balance the transition inductance.

The CPWG structure needs a solid ground plane under the trace; leaving cutouts in the bottom ground plane under the topside trace adds a significant inductance to the structure, which degrades high-frequency performance. You also need to "stitch" the top ground plane to the bottom ground plane with vias. Place the stitching vias less than one-eighth of a wavelength of the highest frequency that your circuit will use. Note that 0.1-in. spacing works well at frequencies greater than 10 GHz.

Spacing of the stitching vias to the center trace follows the same spacing rules. You can easily get enough vias in and around the trace to make it work.

If you don't have enough vias, you will see a slight but rapid 0.5- to 1-dB drop in the S_{21} transmission characteristics instead of a linear loss slope with frequency. You can instantly see this effect by using a VNA (vector network analyzer). Measuring the test board shows approximately 0.25 dB/in. of loss at 3 GHz and 1 dB/in. of loss at 10 GHz, including two edge-launch connectors.

To interface to an SMT part or an IC with narrower pads than 0.032 in., narrow down the center conductor as needed as close to the part as possible. If the discontinuity is physically small, it will have little effect until very high frequencies. EDN

PLL filter blocks undesired frequencies

Stephen Kamichik, Ile Bizard, PQ, Canada

You often need to block signals of specific frequencies; of these frequencies, 50- or 60-Hz line frequency is the most common. You can use the PLL notch filter in **Figure 1** to block unwanted frequencies. IC_1, an LM567C, is a tone decoder. Components C_1, R_{1A}, and R_{1B} determine the frequency, F, that IC_1 detects: $F=1/[C_1(R_{1A}+R_{1B})]$. When you feed frequency F to Pin 3 of IC_1, the output, Pin 8, goes low because the output transistor in IC_1 is saturated.

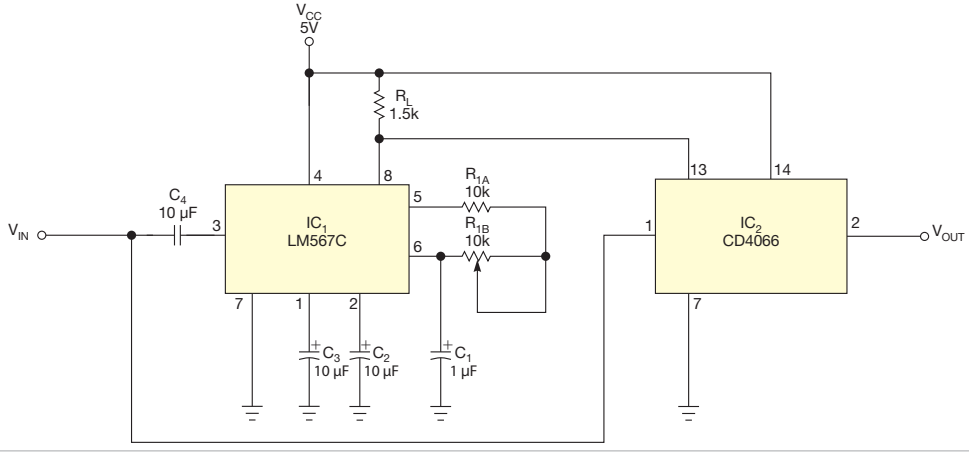


Figure 1 A tone decoder and a switch block frequencies that external components determine.

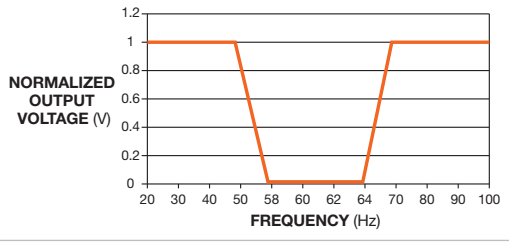


Figure 2 The components in Figure 1 block frequencies of approximately 60 Hz.

When you feed frequency F to Pin 3 of IC_1, the output, Pin 8, goes low because the output transistor in IC_1 is saturated.

The LM567 decoder comprises an in-phase and quadrature detector, which a VCO (voltage-controlled oscillator) drives. The VCO determines the decoder's center frequency. The bandwidth of the decoder is $1070\sqrt{V/(C_2F)}$, where V is the rms (root-mean-square) input voltage and C_2 is capacitance in microfarads. The bandwidth is a percentage of the frequency.

The tone decoder's output runs to the control pin, Pin 13, of IC_2, a CD4066 quad bilateral switch. The input voltage connects to the CD4066's input pin, Pin 1. That signal controls the switch. The CD4066 switch is closed, or on, when the control pin is high at logic one and open, or off, when the control pin is low at logic zero. When IC_1 detects the unwanted frequency—in this case, 60 Hz—IC_1's Pin 8 and, thus, IC_2's Pin 13 go low. That action opens the switch, which blocks the signal with the unwanted frequency (**Figure 2**). EDN

Logic probe uses six transistors

Raju R Baddi, Raman Research Institute, Bangalore, India

The circuit in **Figure 1** lets you build a logic probe using three NPN transistors and three PNP transistors. Two transistors act as switches that drive the LEDs; logic one is a green LED, and logic zero is red. Q_1 and Q_2 test the probe-tip condition for logic one, and Q_3 and Q_4 test it for logic zero. Q_1 acts as a zener diode in the emitter circuit of Q_2. The voltage divider comprising R_{12} and R_{14} determines the diode's value. That value creates a lower limit for the breakdown of the base-emitter junction of Q_2 through V_L. These values ensure that the threshold value for logic one at the probe tip is approximately 3.2V. Q_1's breakdown voltage in the emitter circuit of Q_2 is approximately 2.6V. The **equation** for setting this threshold is $V_{HIGH}=1.2+(VR_{14}/$

R_{12}+R_{14}),where V is the supply voltage. Because V_{HIGH} is a function of the supply voltage, the probe is suitable for CMOS transistors, as well. When the voltage at the probe tip goes above this voltage, the base-emitter junctions of both Q_1 and Q_2 are forward-biased, and they have a common collector-emitter current that flows through R_4, producing enough voltage to forward-bias Q_5 and turning on the green LED. Ideally, R_1 and R_2 maintain the voltage at the probe tip at approximately 2.5V, which is less than 3.2V.

Transistors Q_3 and Q_4 form a comparator of their base voltages. The divider combination comprising R_8 and R_9 maintains the base of Q_4 at a specific voltage, which is approximately 1.9V. Because the probe's suspended voltage is greater, Q_3 conducts, and no current flows through R_6. Thus, Q_6 and the red LED are both off. If the voltage at the probe tip goes below 1.9V, however, Q_4 has a higher voltage at its base than Q_3, and the common-emitter current through R_7 diverts to R_6 through Q_4. This action produces sufficient voltage drop across R_6 to turn on Q_6 and, hence, the red LED. The following equation sets the low-voltage threshold: $V_{LOW}=[VR_9/(R_8+R_9)]$.

The current through the probe tip is –50 to +80 µA for a logic voltage of 0 to 5V. An appendix in the online version of this Design Idea, at www.edn.com/101215dia, details the derivation of these equations and the probe-tip current. **Figure 2** shows the construction method for building a compact probe. EDN

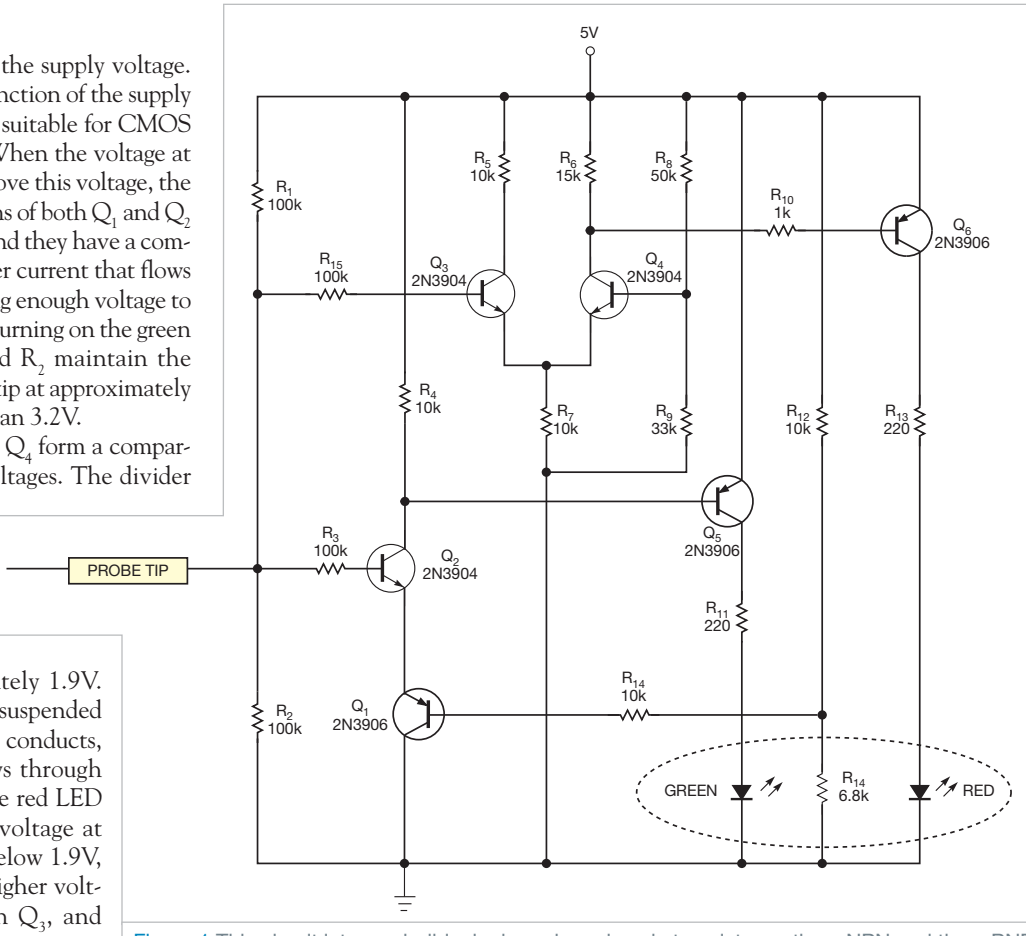


Figure 1 This circuit lets you build a logic probe using six transistors—three NPN and three PNP.

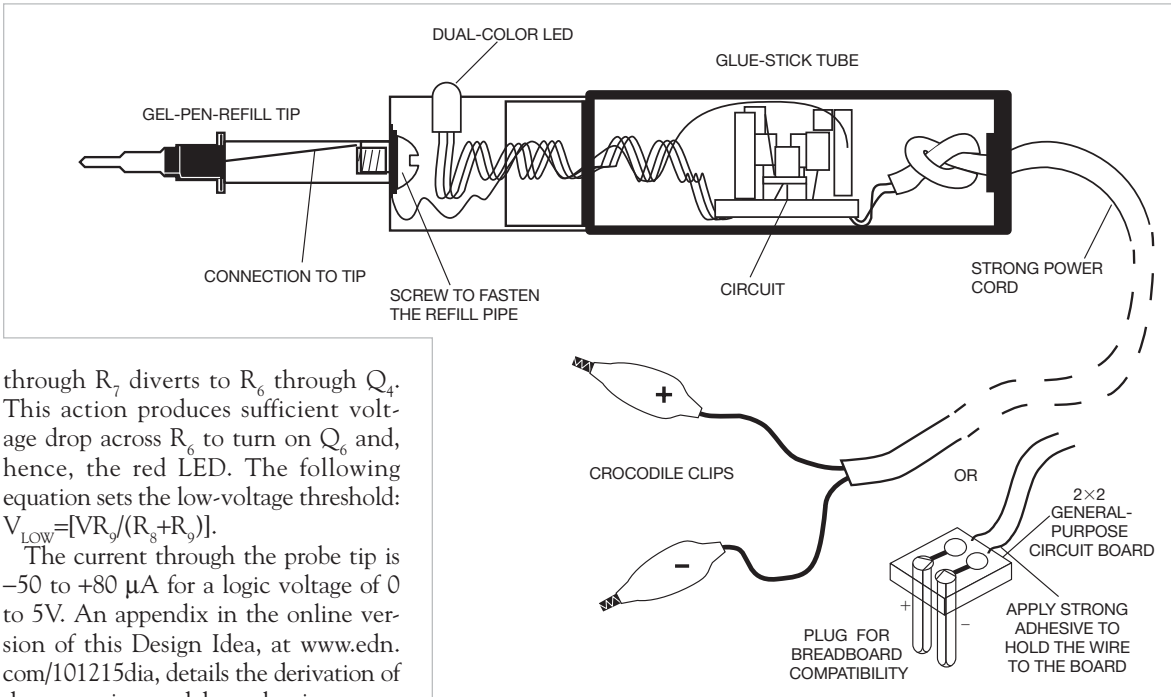


Figure 2 The construction method for building a compact probe requires a dual-color LED, a screw, the circuit, and crocodile clips.