design ideas

Edited by Bill Travis and Anne Watson Swager

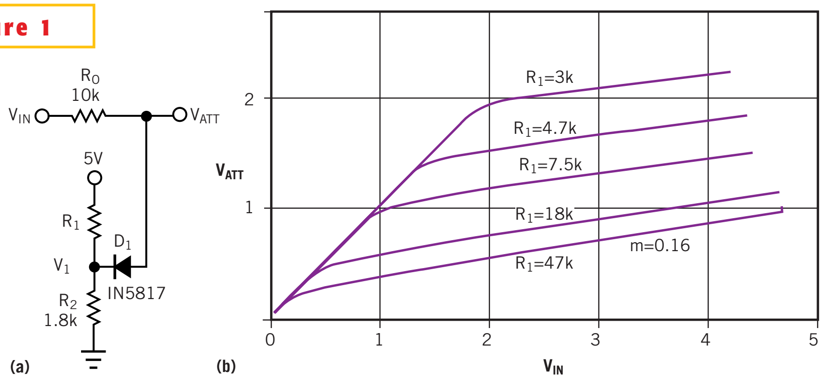
Simple circuits provide nonlinearity

Abel Raynus, Armatron International, Melrose, MA

In analog design, you might need to create an amplifier with nonlinear dynamic characteristics—for example, logarithmic, exponential, or square-law. Usually, such amplifiers are complicated. However, the project often does not require mathematical precision. For example, you might just need to increase the dynamic range of an amplifier, or to eliminate saturation for an extended input-voltage range. The Design Idea is based on the nonlinear voltage attenuator with the attenuation ratio $m=V_{ATT}/V_{IN}$, controlled by the input voltage (**Figure 1a**). When V_{IN} is small enough to hold off D_1, $m=1$. When the input voltage increases and attains a certain threshold voltage, V_{TH}, the diode conducts, and the attenuation ratio decreases. The new value of m depends on the values of R_0 and R_2. R_1 and R_2 determine the threshold level, V_{TH}. Hence, you can estimate the resistors R_1 and R_2 for a given R_0, V_{TH}, and m as: where V_D is the drop across diode D_1, and V_R is the dc voltage applied to R_1 and R_2.

$$R_2=R_0\frac{m}{1-m};$$

Figure 1

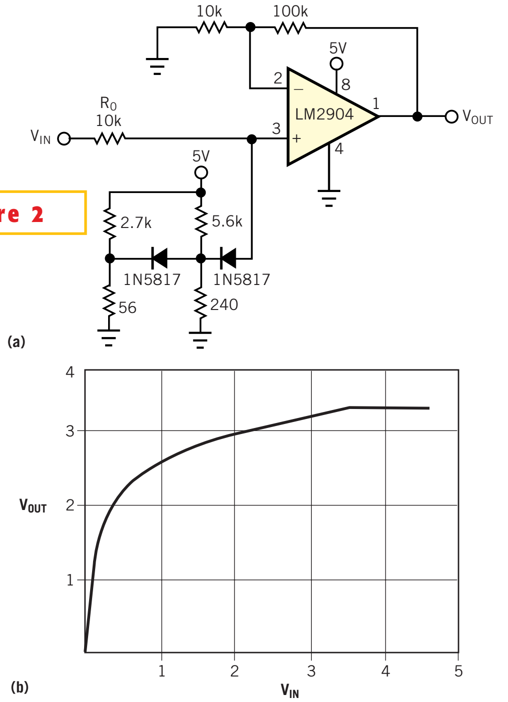


Manipulating the resistor values in an attenuator (a) changes the threshold for nonlinearity (b).

$$R_1=R_0\left(\frac{V_R}{V_{TH}-V_D}-1\right),$$

Note that you can create any characteristic by choosing the proper ratio, m, and the threshold voltage for each fragment of the resulting characteristic. Also, the linear approximation is good for calculation purposes, but the real ratio changes smoothly near the threshold voltage. Finally, use Schottky diodes to increase the voltage range of regulation. **Figure 1b** shows the dynamic response of the attenuator for the constant ratio $m=0.16$ but for different threshold voltages. The measured voltage is $V_1=V_{TH}-V_D$. To increase the dynamic range of an amplifier, you should put the nonlinear attenuator at the input of the amplifier. To

Figure 2

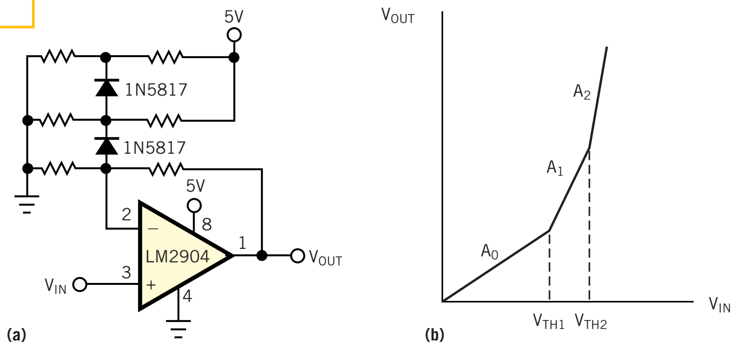


A multistage nonlinear attenuator (a) can increase the dynamic range of an amplifier (b).

widen the range, you can use a two-stage attenuator. **Figure 2** shows such an amplifier and its recorded characteristic. The applications of the nonlinear attenuator are not limited to increasing dynamic range. You can obtain a square-law response, for example, by putting the attenuator in a feedback circuit (**Figure 3**).

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

Figure 3



A nonlinear attenuator in the feedback loop (a) results in a square-law characteristic (b).

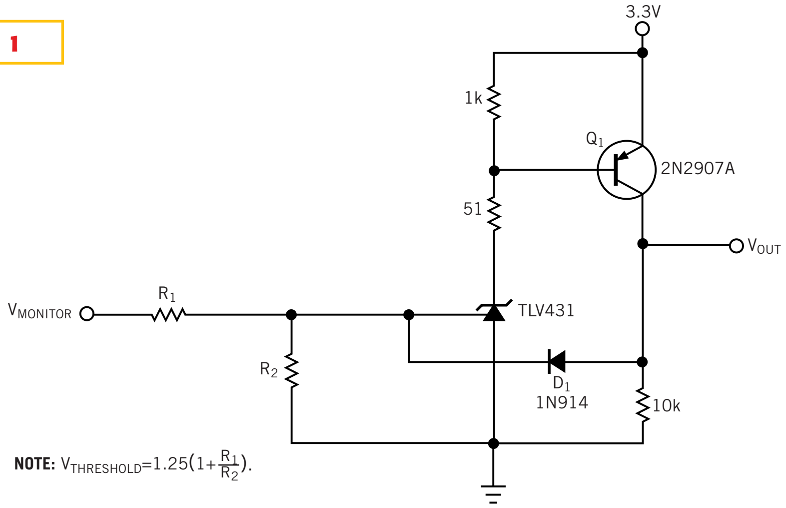
Regulator IC forms convenient overvoltage detector

Robert Bell, On Semiconductor, Phoenix, AZ

Figure 1 shows a simple, standalone overvoltage detector. The intent of the circuit is to monitor a voltage, $V_{MONITOR}$, and set the output, V_{OUT}, high when the monitored voltage exceeds a preset threshold. The minimum allowable threshold for this circuit is 1.25V. The operation of the circuit revolves around the TLV431 shunt regulator. This IC is based on the popular TL431 shunt regulator. The difference is that the TLV431's internal reference is 1.25V, as opposed to 2.5V for the TL431. When the voltage at the control input is less than 1.25V, the regulator's cathode current is essentially zero. If the control input exceeds 1.25V, the cathode conducts and turns Q_1 on to produce a high output at V_{OUT}. The trip threshold, determined by resistors R_1 and R_2, is $V_{THRESHOLD}=1.25(1+R_1/R_2)$. D_1, the diode between V_{OUT} and the control input, provides hysteresis and latches the overvoltage fault condition. If you don't need latching operation, you can add a resistor in series with the diode to lower the hysteresis value and prevent the circuit from latching.

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

Figure 1



NOTE: $V_{THRESHOLD}=1.25(1+\frac{R_1}{R_2})$.

A shunt regulator makes an inexpensive overvoltage detector.

Circuit provides effective LCD drive

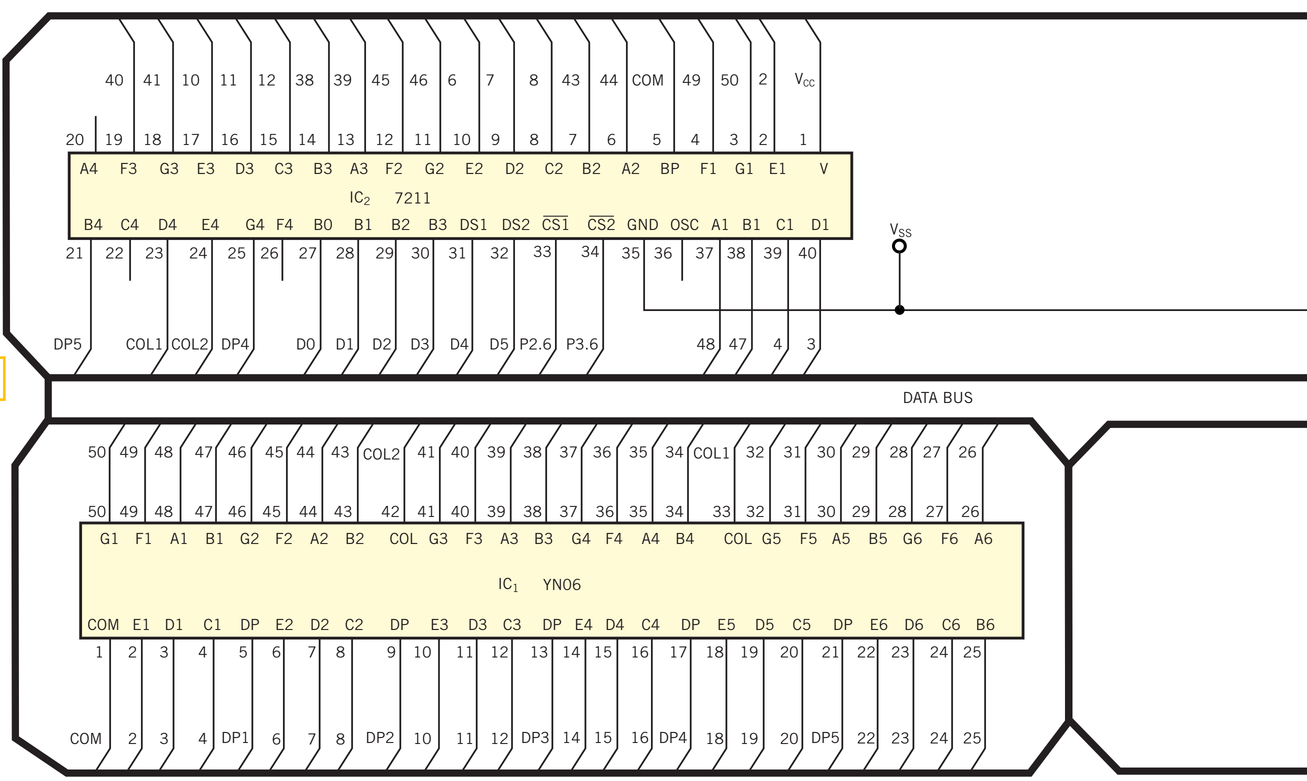
Luo Ben Cheng, Chinese Academy of Science, Beijing, China

LCDs FIND WIDE USE in portable instruments, thanks to their attractive display and low power consumption. The circuit in **Figure 1** is an effective driver for LCDs. The circuit comprises two main sections—the ICM7211 drivers (IC$_2$ and IC$_3$) and the YN06 display itself (IC$_1$). The Intersil (www.intersil.com) ICM7211 is a 4-bit LCD driver that needs no external components. It contains three basic sections: a reference signal-generator circuit, an input and display-channel section, and a digit-selection and drive circuit. It contains a complete pulse-generator unit and an oscillator-divider clock-drive circuit. When you disconnect the BP pin (Pin 5), the IC produces a 125-Hz pulse signal. YN06 is a six-bit character LCD, which uses 5 decimal bits and 2 column bits. To control the display, you need a 4-bit BCD driver.

In **Figure 1**, an AT89C51 µC controls the two ICM7211 drivers. The drivers in turn drive the 6-bit YN06. Pin 5 of IC$_2$ and IC$_3$ connect to the COM pin (Pin 1) of IC$_1$. The reference signal-generator circuit works in open-loop mode. This mode results when you disconnect the OSC pin (Pin 36) of IC$_2$ and connect the OSC pin (Pin 36) of IC$_3$ to ground. The result is a 125-Hz pulse train, which serves as the LCD's drive clock. The chip-enable signals $\overline{\text{CS1}}$ of IC$_2$ and IC$_3$ to connect to the µC's pins P2.5 and P2.6, and $\overline{\text{CS2}}$ connects to Pin P3.6, which serves as a read or write port. In addition, data-input ports B0 to B3 and digital-selection input ports DS1 and DS2 connect to the data bus through the D0 to D5 lines. To control the LCD, you need only provide 4-bit BCD codes through the µC. Unfortunately, in some cases, the display needs decimal bits. The normal method of providing these bits is to add another LCD decimal driver, such as a CD4056.

Note that the LCD in **Figure 1** needs only 6 bits, whereas the drivers can provide 8 bits. That fact means that two more seven-segment output ports go unused. You can take advantage of the unused ports of IC$_2$ and IC$_3$ to solve the decimal-bit problem. Connect DP1 (Pin 5 of IC$_1$) to Pin 25 of IC$_3$, DP2 (Pin 9 of IC$_1$) to Pin 23 of IC$_3$, DP3 (Pin 13 of IC$_1$) to Pin 21 of IC$_3$, and DP4 (Pin 17 of IC$_1$) to Pin 25 of IC$_2$. Also, connect COL1 (Pin 33 of IC$_1$) to Pin 23 of IC$_2$, and COL2 (Pin 42 of IC$_1$) to Pin 24 of IC$_2$. With the help of some µC software, you can control the LCD in a flexible fashion. **Listing 1** shows the AT89C51 assembly code for controlling the LCD. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2574.

Figure 1

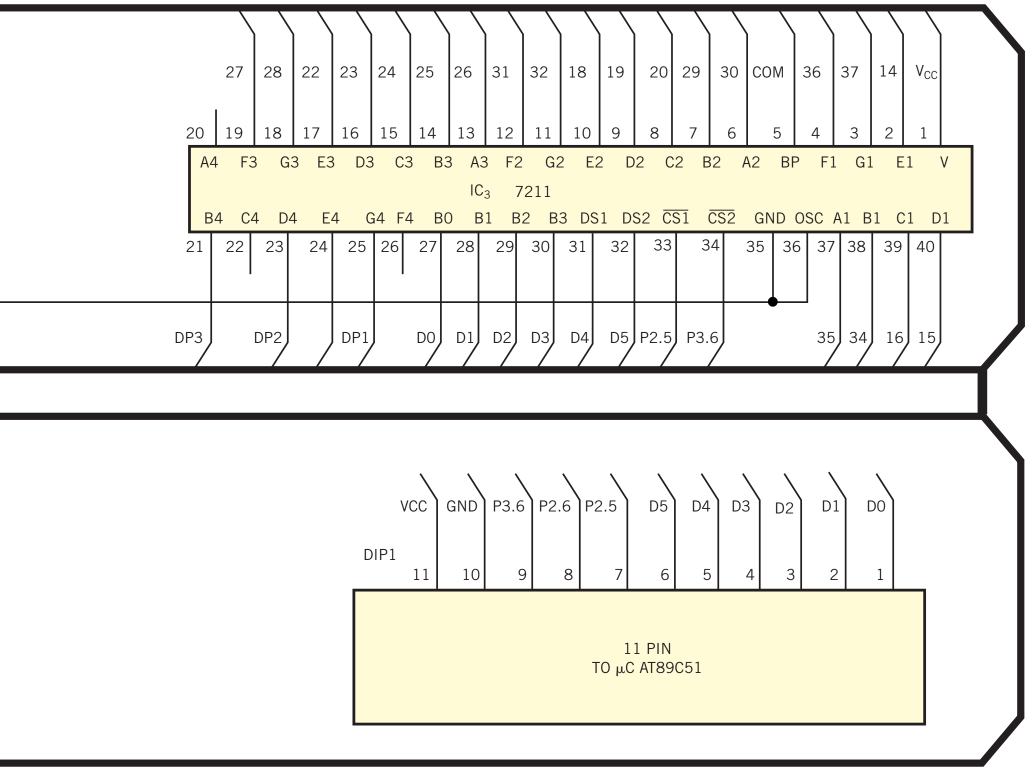


The unused driver outputs take care of the LCD's need for decimal-bit inputs.

LISTING 1—ASSEMBLY CODE FOR AT89C51 LCD DRIVE

```
ORG  0000H
          LJMP  MAIN
          ...
ORG  0030H
     FIG3          EQU  23H
     ADR00         EQU  40H
     SDATA         EQU  50H
     BITD          EQU  53H
  ADRU2    EQU  3000H
  ADRU3    EQU  5000H
MAIN:              CLR EA                ;CLOSE ALL THE INTERRUPTS
                   ...
                   LCALL DISPLAY              ;DISPLY DATUS
                   ...
HERE:              NOP
                   AJMP HERE                  ;WAIT FOR INTERRUPTS
; *********************************************************************
ORG  0120H
DISPLAY:           MOV R1, #SDATA             ;INPUT THE DATUS FOR DISPLAY
                   MOV A, BITD
     DIR:          MOV R6, #4                 ;FOR ONE 4_BIT LCD DRIVER
                   MOV R0, #ADR00             ;DATUS BUFFERS
                   MOV FLG3, A                ;STORE THE DECIMAL BIT
                   LCALL DOT                  ; PROCESS THE DECIMAL BIT
                   LCALL DXCH                 ;FORMAT THE DATA FOR DISPLAY
     SDIR:         MOV DPTR,#ADRU2            ;CHOICE THE LCD DRIVER U2
                   MOV A,@R0
                   MOVX @DPTR,A
                   INC R0
                   DJNZ R6,SDIR
                   MOV R6,#4
SDIRO:             MOV DPTR,#ADRU3            ;CHOICE THE LCD DRIVER U3
                   MOV A,@R0
                   MOVX @DPTR,A
                   INC R0
                   DJNZ R6,SDIRO
                   MOV R6,#4             ;RECOVER THE INFORMATION
                   MOV R0,#ADROO
                   RET
; *********************************************************************
DOT:               MOV R0,#ADROO              ;DECIMAL BIT PROCESSING
                   JB FLG3.0,D0
                   JB FLG3.1,D1
                   JB FLG3.2,D2
                   JB FLG3.3,D3
                   JB FLG3.4,D4
                   JB FLG3.5,D5
                   JB FLG3.6,COL
D0:           MOV @R0,#0FH
                   MOV R0,#ADROO+7
                   MOV @R0,#0FH
                   AJMP DCOL
D1:           MOV @R0,#0FH
                   MOV R0,#ADROO+7
                   MOV @R0,#01H
                   AJMP DCOL
D2:           MOV @R0,#0FH
                   MOV R0,#ADROO+7
                   MOV @R0,#0AH
                   AJMP DCOL
D3:           MOV @R0,#01H
                   MOV R0,#ADROO+7
                   MOV @R0,#0FH
                   AJMP DCOL
D4:           MOV @R0,#0DH
                   MOV R0,#ADROO+7
                   MOV @R0,#0FH
                   AJMP DCOL
D5:           MOV @R0,#0AH
                   MOV R0,#ADROO+7
                   MOV @R0,#0FH

                   AJMP DCOL
COL:               MOV @R0,#0FH
                   MOV R0,#ADROO+7
                   MOV @R0,#0DH
DCOL:              MOV R0,#ADROO+1       ;JUMP OVER THE DECIMAL BIT
                   RET
; *********************************************************************
DXCH:                   MOV A,@R1             ;DATA PROCESSING MODULE
                        MOV B,A
                        ANL A,#0FH
                        ORL A,#10H
                        MOV @R0,A        ;CHOICE THE SIXTH BYTLE FOR DISPLAY
                        INC R0
                        MOV A,B
                        SWAP A
                        ANL A,#0FH
                        ORL A,#20H       ;CHOICE THE FIFTH BYTLE
                        MOV @R0,A
                        INC R0
                        DEC R1
                        MOV A,@R1
                        MOV B,A
                        ANL A,#0FH
                        ORL A,#30H
                        MOV @R0,A        ;CHOICE THE FOURTH BYTLE
                        INC R0
                        MOV A,B
                        SWAP A
                        ANL A,#0FH
                        ORL A,#10H
                        MOV @R0,A         ;CHOICE THE THIRD BYTLE
                        INC R0
                        DEC R1
                        MOV A,@R1
                        MOV B, A
                        ANL A,#0FH
                        ORL A,#20H
                        MOV @R0,A         ;CHOICE THE SECOND BYTLE
                        INC R0
                        MOV A,B
                        SWAP A
                        ANL A,#0FH
                        ORL A,#30H
                        MOV @R0,A         ;CHOICE THE FIRST BYTLE
                        MOV R0,#ADROO
                        INC R1
                        INC R1
                        RET
```



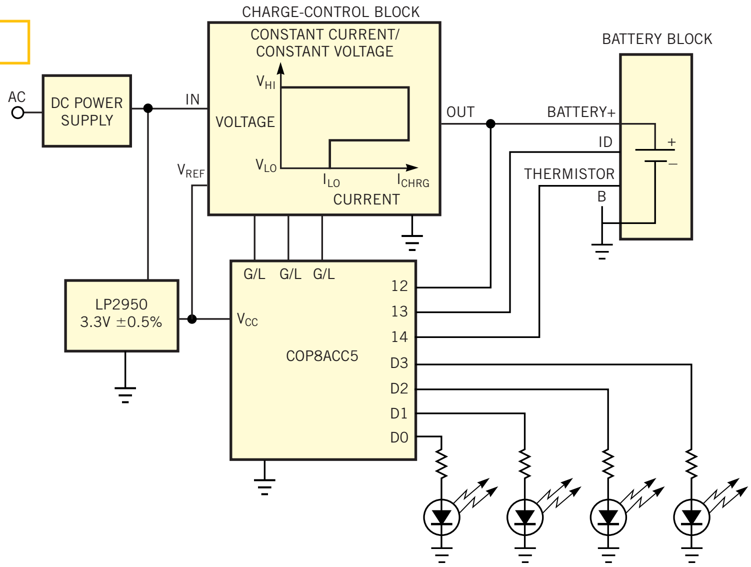
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

µC controls multichemistry battery charger

Kelly Flaherty, National Semiconductor, Santa Clara, CA

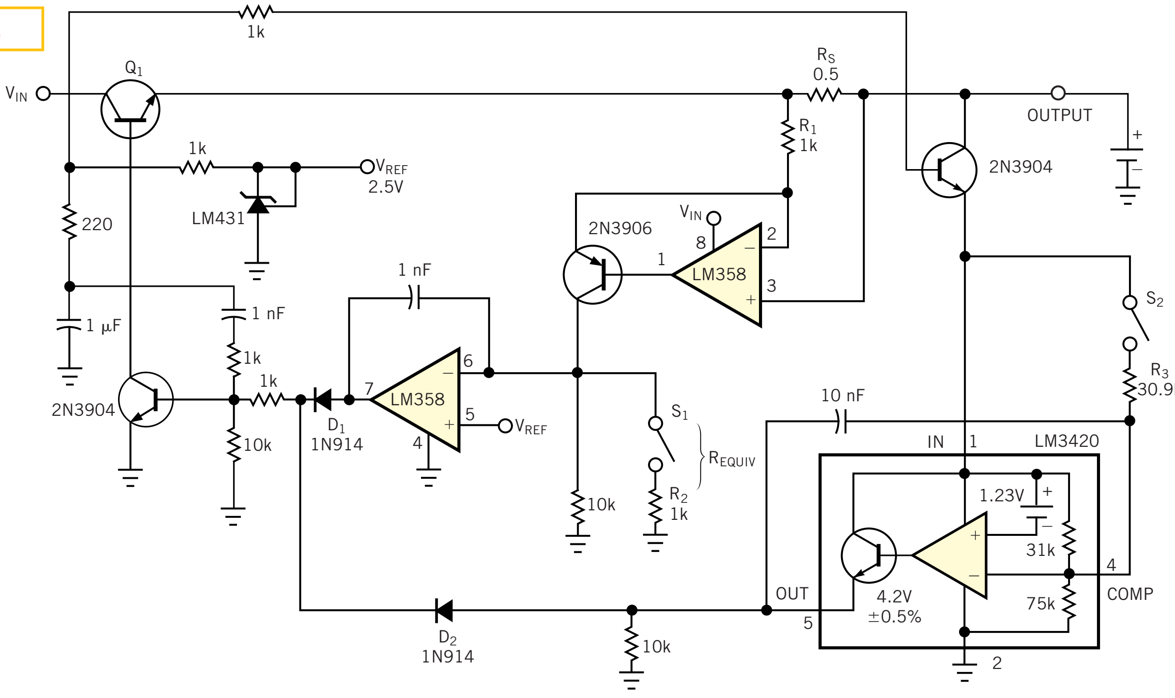
Figure 1 is a generalized block diagram of a multichemistry battery charger. A COP8ACC5 µC handles its key charging features. The µC is available in a 20-pin (15 I/O pins) SOIC or a 28-pin (23 I/O pins) SOIC/DIP. It contains 4 kbits of internal ROM. The controller's A/D inputs monitor the battery-voltage pin, ID pin, and thermistor pin. For more complex charging systems, you can add external EEPROM via the Microwire serial interface. Such an external EEPROM might be useful for storing battery-specific charge history, a battery-specific look-up table for more accurate charging, or both. The LP2950 doubles as a low-dropout-voltage regulator for the µC and as a ±0.5% reference for the charge-control block. The charge-control block is basically a constant-voltage, constant-current regulator, as the voltage-versus-current curve in **Figure 1** shows. The µC reads the battery pack's ID pin and adjusts the circuit

Figure 1



This stand-alone battery charger handles multiple battery chemistries.

Figure 2



A battery charge-control block operates in both constant-current and constant-voltage mode.

accordingly. If the battery is a lithium-ion type, the charge-control block adjusts the fast-charge rate according to the battery's capacity rating and adjusts the constant voltage to the critical maximum voltage for lithium-ion. If the battery is nickel-based, similar adjustments take place. However, the voltage adjustment is to a level greater than the maximum battery voltage that is still low enough to accommodate the power dissipation of the pass transistor.

Figure 2 shows a possible implementation of the charge-control block. It is an adjustable constant-current, constant-voltage regulator under control of a µC. The switches can be analog switches, such as the CD4066; discrete transistors, such as the 2N3904; or FETs, such as the 2N7002. The default setting (switches open) is 4.2V and 0.5A. When S_1 closes, current regulation increases by the change in equivalent resistance (R_{EQUIV}): $I_{CHRG}=(V_{REF}\times R_1)/(R_{EQUIV}\times R_{SENSE})$. S_1 switches in R_2, resulting in the doubling (to 1A) of the default regulated current. Closing S_2 similarly increases the level of the regulated voltage from the LM3420 lithium-ion charge controller. The LM3420 contains an error amplifier, a precision voltage reference, and a trimmed voltage divider that sets the regulated voltage to within ±0.5%. The IC is available in five fixed voltage levels that correspond to 4.2V per cell for one, two, three, and four cells. The Comp pin of the LM3420 switches an external resistor, R_3, in parallel with one of the internal divider resistors, and results in a regulated voltage of 7.2V. Q_1 provides a disconnect between the battery and the LM3420 upon removal of the input voltage. D_1 and D_2 act as an exclusive-OR gate for current regulation of voltage regulation. When V_{REG} is reached, D_2 overrides D_1.

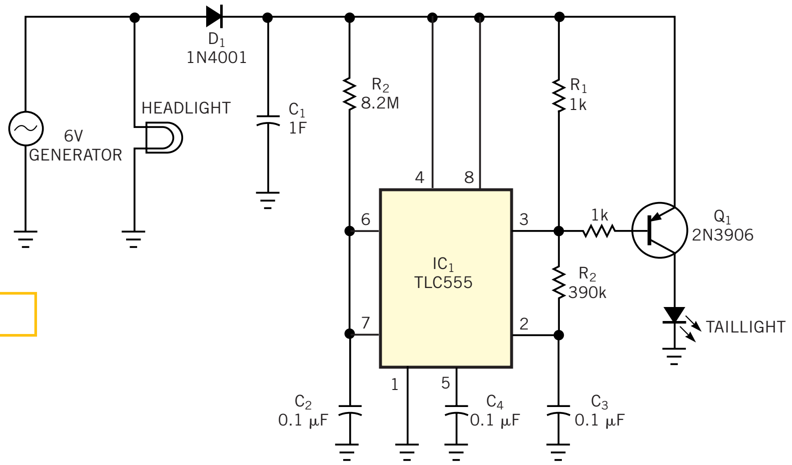
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Ultracapacitor powers bicycle light

Dennis Eichenberg, Parma Heights, OH

The circuit in **Figure 1** represents a simple and inexpensive way to provide additional bicycling safety. A conventional bicycle-generator lighting system works with the circuit to provide safety lighting for several minutes after the bicycle has stopped. Energy storage uses an ultracapacitor rather than a battery to avoid the typical battery problems of limited life, critical charging rates, and intolerance to cold. The advent of inexpensive, compact ultracapacitors has made this approach practical. A standard headlight connects to the generator in a normal fashion, so that the headlight is on whenever the bicycle is moving. An ultracapacitor receives its charge from the generator and connects to an astable multivibrator to pulse the taillight.

Figure 1



An ultracapacitor provides additional safety by flashing the taillight after the bike has stopped.

A typical bicycle generator is a 6V-ac device. The load regulation of the generator is poor, so connecting the headlight directly to the generator stabilizes the generator's output. Diode D_1 provides half-wave rectification of the generator's output to charge capacitor C_1 and to power the taillight circuit. D_1 also acts as a blocking diode to prevent C_1 from discharging back through the headlight and the generator. The CMOS 555 timer, IC_1, acts as an astable multivibrator tolerant of voltage variations as low as 2V. The circuit provides an off-time of 820 msec ($1.1\times R_1\times C_2$) and an on-time of 43 msec ($1.1\times R_2\times C_3$). IC_1 drives transistor Q_1 to pulse the grounded taillight. You must use an LED for the taillight, because it does not draw the extreme surge current of an incandescent lamp and thus provides several minutes of illumination. A 2V LED limited to 100 mA provides the greatest duration of light.

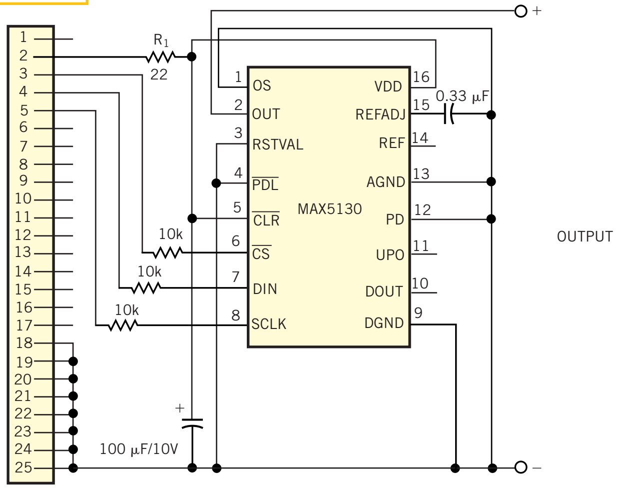
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Printer port controls reference generator

Yongping Xia, Teldata Inc, Los Angeles, CA

Figure 1 shows a programmable, precision reference-voltage generator. A PC's printer port controls the generator. The voltage range is 0 to 4.0955V in 0.5-mV increments. Because the computer's hard drive saves the reference setting, when you restart the computer, the output voltage is exactly the same as the previous setting. The Max5130 is a 13-bit serial voltage-output DAC with an internal reference. It uses a three-wire serial interface. Because the IC has everything necessary for a programmable reference, the circuitry is simple. The printer port's Pin 2 powers the circuit. Pins 3, 4, and 5 provide chip select (CS), data (DIN), and clock (SCLK), respectively, to the Max5130. **Listing 1** is a C program for the generator. "U" and "D" keys speed the voltage setting, given that the DAC has 8192 steps. Push "U" or "D" for 100-step changes, equivalent to 650-mV steps. The "u" and "d" keys fine-tune the output with 0.5 mV per step.

Figure 1



Use a PC's printer port and a 13-bit DAC to configure a precision reference generator.

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

LISTING 1—C PROGRAM FOR PRECISION REFERENCE GENERATOR

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <bios.h>
#include <math.h>

#define POWER_ON     0x01
#define CLK_HIGH     0x08
#define CLK_LOW      0xf7
#define CS_HIGH      0x02
#define CS_LOW       0xfd
#define DATA_HIGH    0x04
#define DATA_LOW     0xfb

typedef unsigned int WORD;
FILE  *data_file;

int i, out_port, in, out=0;
long data, step;
float ref_voltage;

void find_port(void)          /* find printer port address  */
{
  out_port=*(WORD far *)MK_FP(0x0040,8);
  out|=POWER_ON;
  outportb(out_port, out);                /* power on             */
  delay(100);
}

void show_ref(void)                       /* display voltage      */
{
  gotoxy(2,2);
  ref_voltage=(float)step/2000;
  printf("Voltage = %.4fV   ", ref_voltage);
}

void write_ref(void)          /* write data to c driver     */
{
  if ((data_file = fopen("c:\\ref_data", "wb")) == NULL)
    printf("File open failed.\n");
  fseek(data_file, 0, 0);
  fwrite(&step, 1, 2, data_file);
  fclose(data_file);
}
```

```
void read_ref(void)                  /* read data from c driver    */
{
  if ((data_file = fopen("c:\\ref_data", "rb")) == NULL)
    write_ref();
  fseek(data_file, 0, 0);
  fread(&step, 1, 2, data_file);
  fclose(data_file);
}

void send_clock(void)                /* send sclk to MAX5130 */
{

  out|=CLK_HIGH;
  outportb(out_port, out);              /* clock high  */
  delay(2);
  out&=CLK_LOW;
  outportb(out_port, out);              /* clock low   */
  delay(2);
}

void set_ref()                          /* send data to MAX5130   */
{
  out|=CS_HIGH;
  outportb(out_port, out);                 /* cs high  */
  delay(2);
  data=step+0x4000;
  out&=CS_LOW;
  outportb(out_port, out);                 /* cs low   */
  delay(2);
  for (i=0; i<16; i++)
   {
     if (data>=(pow(2, (15-i))))
      {
      data-=(pow(2, 15-i));
      out|=DATA_HIGH;
      }
     else
      out&=DATA_LOW;
     outportb(out_port, out);              /* send 1 bit data       */
     delay(2);
     send_clock();
   }
  out|=CS_HIGH;
  outportb(out_port, out);                 /* cs high       */
  delay(2);
  show_ref();
  write_ref();
}
```

(*continued on pg 134*)

LISTING 1—C PROGRAM FOR PRECISION REFERENCE GENERATOR (CONTINUED)

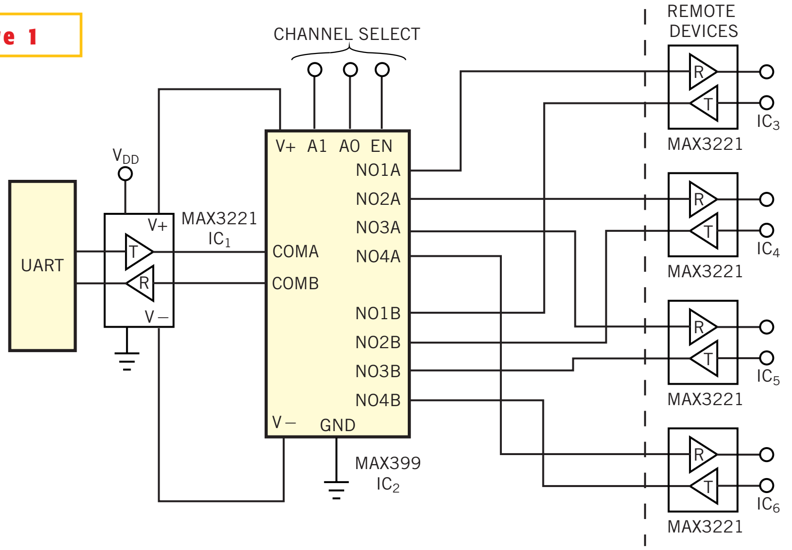
```
void main(void)
{
 clrscr();
 find_port();
 read_ref();
 show_ref();
 set_ref();
 do{
   do{
      } while(!kbhit());
   in=getch();
   if (in=='u')          /* if hit 'u', increase 0.5mV per step    */
    {
      if (step<8191)
      step++;
    }
   if (in=='U')          /* if hit 'U', increase 50mV per step     */
    {
      if (step<8191)
      step+=100;
      if (step>8191)
      step=8191;
    }
   if (in=='d')          /* if hit 'd', decrease 0.5mV per step    */
    {
      if (step>0)
      step--;
    }
   if (in=='D')          /* if hit 'U', decrease 50mV per step     */
    {
      if (step>0)
      step-=100;
      if (step<0)
      step=0;
    }
   set_ref();
 }while(in!='q' && in!='Q');  /* if hit 'q' or 'Q', quit    */
}
```

Multiplexer enables pseudomultidrop RS-232 transmission

Dan Christman, Maxim Integrated Products, Sunnyvale, CA

RS-232 communications with one µC and more than one remote system can be problematic, because most µCs contain only one UART, which provides an interface between synchronous and asynchronous ports. The multiplexer in **Figure 1,** IC_2, allows multiple channels (four, in this case) to share a single UART. The dual four-to-one multiplexer allows transceiver IC_1 to form a network with the four remote transceivers IC_3 to IC_6. **Table 1** defines the channel-selection codes. Selecting Channel 1, for instance, enables IC_1 to communicate with IC_3 without being loaded by IC_4 to IC_6. Pulldown resistors inside the remote transceivers force the outputs of unselected receivers to a known state.

Figure 1



One UART and one multiplexer enable one RS-232 transceiver to communicate with four others.

The circuit's supply-voltage range (3 to 5.5V) makes it compatible with 3 and 5V logic. IC_2 receives its power directly from the V+ and V− terminals of IC_1, whose ±5.5V outputs come from an internal charge pump. The multiplexer handles rail-to-rail signals, so obtaining its power from IC_1 ensures that RS-232 signals pass directly through, regardless of amplitude. Each transceiver's charge pump requires four small capacitors (not shown), whose values depend on the V_{DD} range but do not exceed 0.47 µF. Note that pulling too much current from the charge-pump terminals of IC_1, V+ and V−, can cause these rails to droop and may pull the IC's RS-232 transmission levels out of specification.

TABLE 1—CHANNEL SELECTION

A1	A0	EN	Selected channel
X	X	0	All channels disconnected
0	0	1	Channel 1 (IC3)
0	1	1	Channel 2 (IC4)
1	0	1	Channel 3 (IC5)
1	1	1	Channel 4 (IC6)

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

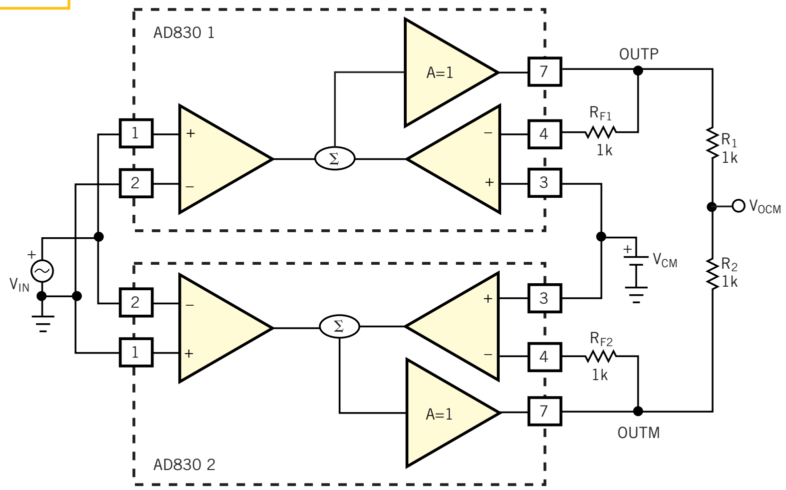
Differential amp drives high-speed ADC

Moshe Gerstenhaber and Chau Tran, Analog Devices, Wilmington, MA

THE SCHEMATIC IN **Figure 1** depicts a differential-input/differential-output or single-ended-input/differential-output amplifier with a gain of two. You can use the low-distortion circuit to drive high-speed ADCs. You can also use it to drive precision delta-sigma ADCs. The circuit contains two active-feedback amplifiers, with input connections such that one amplifier acts as a voltage follower and the other acts as an inverter. You take the output differentially from the amplifiers' outputs. The ADC's reference output can connect to VCM to set the output common-mode voltage of the amplifier stages, or you can set this voltage by external means. Resistors R_{F1} and R_{F2} reduce the distortion of the system. We added R_1 and R_2 for displaying the effects of the amplifiers' mismatch. **Figure 2** is a performance photo at 10 MHz and a gain of two. The top trace is the single-ended input signal; the two bottom traces are the output signals, out of phase with each other. **Figure 3** demonstrates the gain error and the low distortion of the system. The bottom trace, at 10 mV/div, shows the effects of the amplifiers' mismatch at the common-mode node.

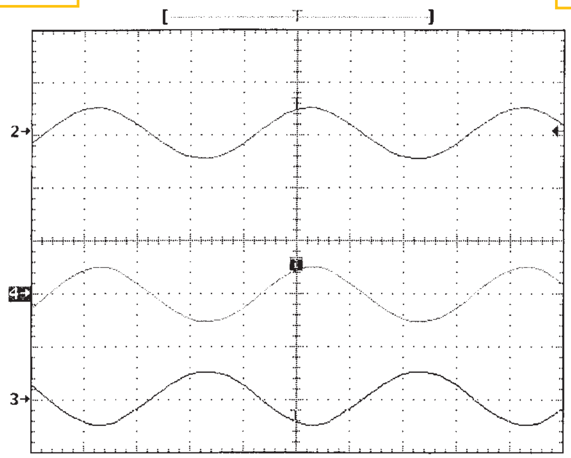
Is this the best Design Idea in this issue? Vote at www.ednmag.com/ednmag/vote.asp.

Figure 1



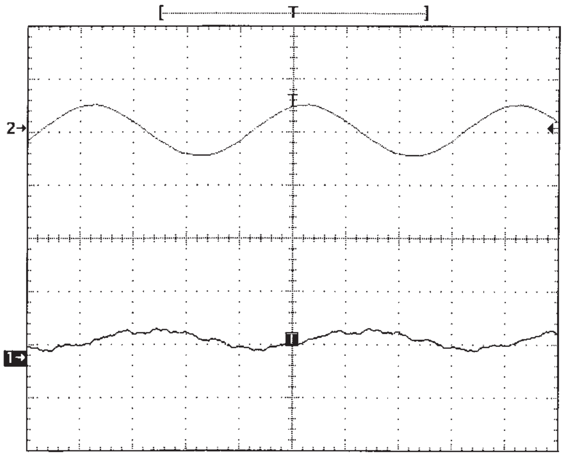
A single-ended-input/differential-output configuration drives high-speed ADCs with minimal common-mode error.

Figure 2



In the bottom traces, the differential outputs are 180° out of phase with each other.

Figure 3



The bottom trace shows the amplifiers' mismatch at 10 mV/div.