design ideas

Edited by Bill Travis

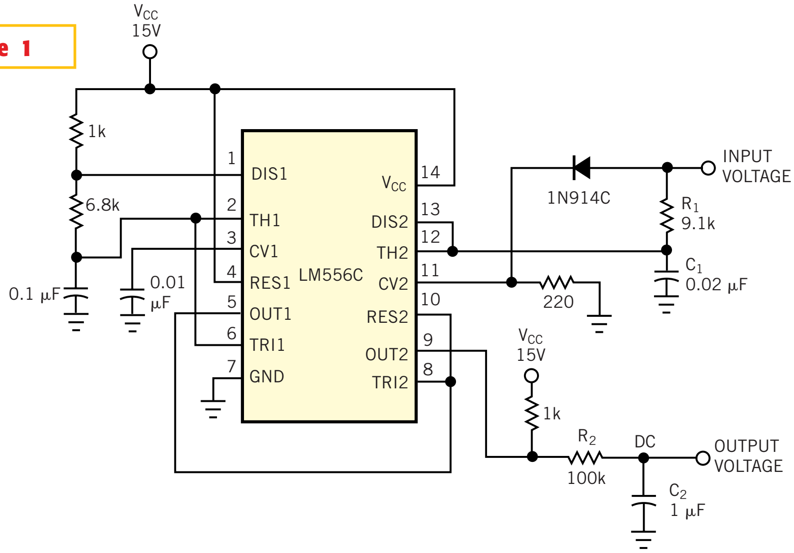
Log amp uses capacitor-charging law

Jayashree Raghuraman and Ajoy Raman, Aeronautical Development Establishment, Bangalore, India

THE NOVEL LOGARITHMIC amplifier in **Figure 1** relies on the exponential charging characteristics of a simple RC circuit. The expression for the time, T, required for a capacitor, C, to reach a voltage ($V_{IN}-V_K$) from 0V, when charged through a resistor, R, with an applied voltage of V_{IN}, is $V_{IN}-V_K=V_{IN}(1-e^{-T/RC})$, where V_K is a fixed voltage. The expression for T reduces to $T=RC\ln(V_{IN}/V_K)$, clearly showing an inherent logarithmic characteristic. The circuit in **Figure 1** demonstrates this characteristic, using a 556 timer. With the values shown, the first stage of the 556 timer is a standard astable circuit operating at a frequency of approximately 1 kHz. The output of this stage acts as the trigger for the second stage. The second stage operates as a modified monostable circuit. In this modified configuration, the RC combination, R_1 and C_1, charges from an external voltage, V_{IN}, instead of V_{CC}. The control-voltage pin, CV2, has the value V_{IN} minus one diode drop, V_K.

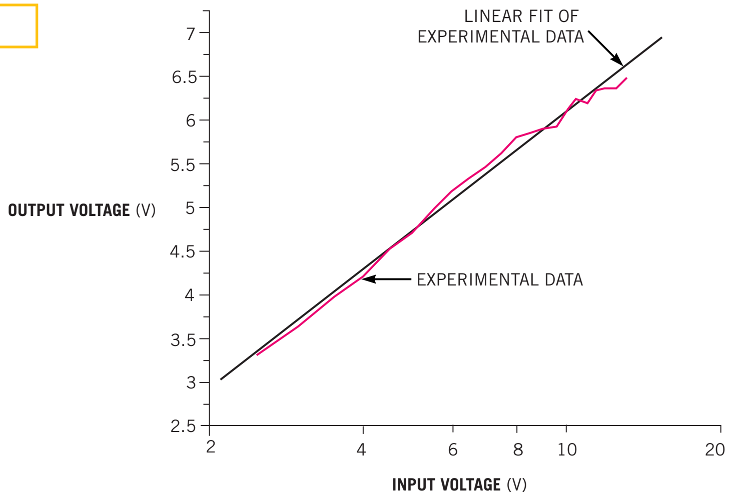
The monostable pulse width, T, then depends on the time required for capacitor C_1 to charge to $V_{IN}-V_K$ through R_1

Figure 1



A simple 556 timer depends on RC charging to form a logarithmic amplifier.

Figure 2



The circuit of Figure 1 produces a distinct logarithmic output.

with the applied voltage V_{IN}. The output of the second stage, filtered through R_2 and C_2, depends on the first stage's astable frequency; the supply voltage, V_{CC}; and the monostable pulse width, T. Because V_{CC} and the astable frequency are constant, V_{OUT} is proportional to T. **Table 1** tabulates the experimental results, and **Figure 2** shows graphical results. The circuit operation is limited to an input range of 2.5 to 13V to satisfy the internal biasing requirements of the second stage of the 556. The diode drop, V_K, is not strictly constant, because it varies with current. In spite of these limitations, **Table 1** and **Figure 2** clearly show a distinct logarithmic characteristic.

TABLE 1—OUTPUT VERSUS INPUT VOLTAGE

Input voltage	Output voltage	Input voltage	Output voltage
2.5	3.324	8	5.782
3	3.667	8.5	5.861
3.5	3.954	9	5.886
4	4.227	9.5	5.945
4.5	4.506	10	6.098
5	4.705	10.5	6.187
5.5	4.956	11	6.204
6	5.151	11.5	6.312
6.5	5.315	12	6.371
7	5.444	12.5	6.378
7.5	5.615	13	6.476

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

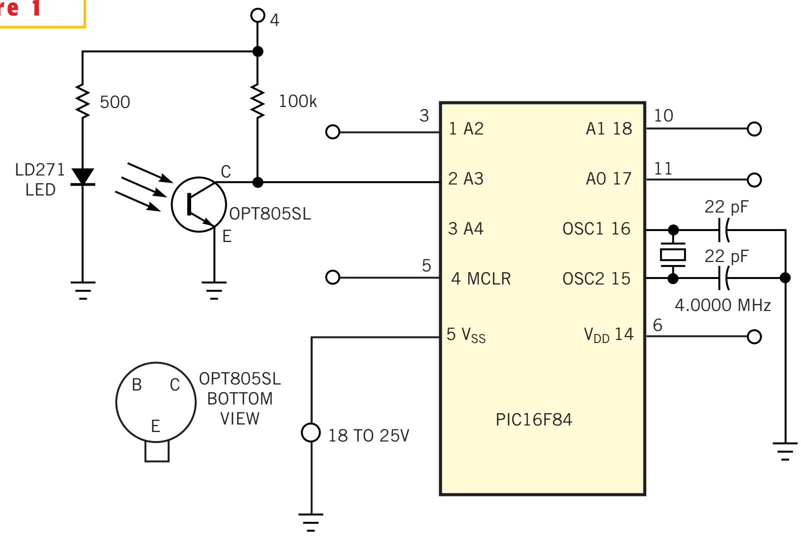
Extend the timing capabilities of a PC

Martin Connors, Athabasca University, Athabasca, AB, Canada

Even when you use the internal timing registers and under DOS, a PC cannot easily measure time intervals with better time resolution than a millisecond. Measuring long intervals with even this precision is a waste of many CPU cycles. A microcontroller is well-suited for this task; you can easily integrate a PIC with a PC to extend the timing precision into the microsecond range for periods from tens of microseconds to more than 24 hours. The flash-programmable PIC16F84 microcontroller from Microchip Technology (www.microchip.com) is an inexpensive and widely used device. The precision timer in **Figure 1** requires only the IC, two capacitors, and a crystal and accepts direct input of timing data to a PC via the parallel port. The PIC16F84 draws only 2 mA and can operate from an output pin in the parallel port without a battery. You can assemble the circuit on a small pc board with a male DB-25 connector glued or soldered to one end for connection to the parallel port, LPT1. In this example, the timing signal occurs when you block a photogate comprising a paired LED and a phototransistor.

Listing 1 represents the timing application implemented, which comprises two basic parts. The first part waits for a signal and starts a loop that checks the continuing presence of the signal and increments 32 timing bits while the signal is present. The second part transmits 32 bits of timing information to an external device, using one data-output line and two handshaking lines. With a 4-MHz crystal, most instructions take 1 µsec, so the timing loop is 5 µsec long. You can run newer PIC16F84s with a 20-MHz clock, so, in principle, the timing loop can be 1 µsec long. Port A of the PIC serves for the timing signal on bit 3 and for communication. A minor coding change allows you to use positive or negative log-

Figure 1



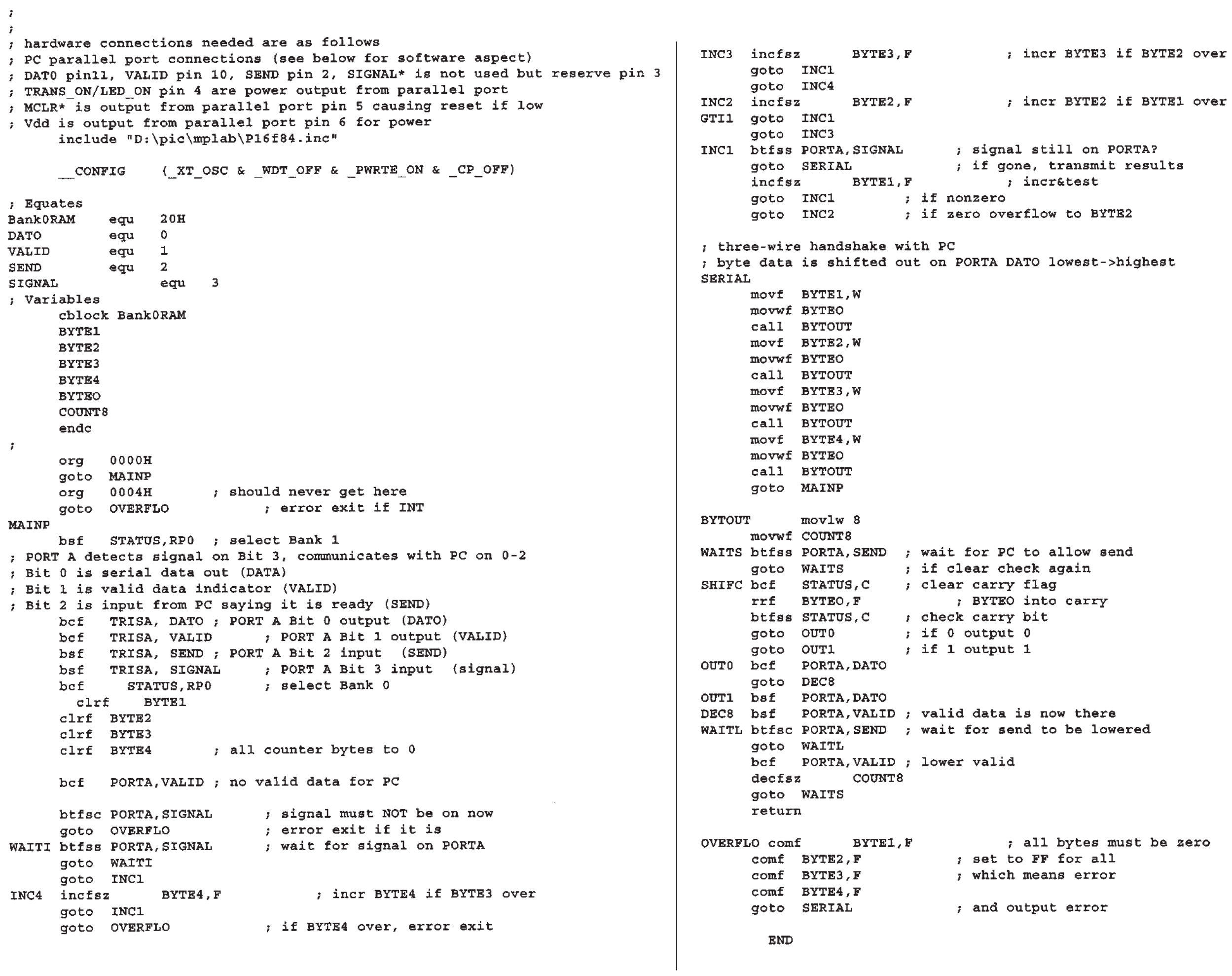
This zero-power photogate allows you to use a PC to generate precise timing intervals.

ic levels. If the timing signal is present at the start of the program, an error flag arises, with an output of 4 bytes of 0xFF. A similar error occurs if the signal is present long enough (roughly a day) to cause overflow of the counter. DATO (data output) occurs through bit 0. The routine uses two handshake lines: VALID on bit 1 from the PIC to signal the presence of valid data on the DATO line and SEND from the PC to bit 2, signifying that the PC is ready to receive data. This robust transmission method does not depend on timing characteristics in a critical way.

Listing 2 (pg 76) shows sample C code for Borland Turbo C for DOS with a simple timing conversion that doesn't take account of the overhead of byte overflow. After the PIC times an event, it waits for the PC to signal that it wants to download data. The transmission protocol for transmitting 1 bit of data is as follows: PC SEND is low, and the PIC polls it. PIC VALID is initially low; the PC raises SEND and polls VALID. In response, the PIC puts DATA on the line. The PIC than raises VALID and polls SEND; in response, the PC reads DATA. The PC then lowers SEND, and the PIC lowers VALID. This operation repeats for 32 bits, starting with the lowest bit of the lowest byte and proceeding to the highest bit of the highest (fourth) byte. Although this transmission method is inefficient, it is robust, and the polling timing is unimportant. The efficiency matters little, because the method involves little data transfer. By referring to the **listings**, you can "step through" the process to see how the transfer takes place. **Listing 2** includes a test routine that allows you to supply a signal from the PC to test the circuit's operation. You can download **listings 1** and **2** from the Web version of this article at *EDN*'s Web site, www.ednmag.com.

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

LISTING 1—ZERO-POWER PHOTOGATE-ASSEMBLY PROGRAM



```
;
;
; hardware connections needed are as follows
; PC parallel port connections (see below for software aspect)
; DAT0 pin11, VALID pin 10, SEND pin 2, SIGNAL* is not used but reserve pin 3
; TRANS_ON/LED_ON pin 4 are power output from parallel port
; MCLR* is output from parallel port pin 5 causing reset if low
; Vdd is output from parallel port pin 6 for power
      include "D:\pic\mplab\P16f84.inc"

      __CONFIG    (_XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF)

; Equates
Bank0RAM     equ    20H
DATO         equ    0
VALID        equ    1
SEND         equ    2
SIGNAL              equ    3
; Variables
      cblock Bank0RAM
      BYTE1
      BYTE2
      BYTE3
      BYTE4
      BYTEO
      COUNT8
      endc
;
      org    0000H
      goto   MAINP
      org    0004H        ; should never get here
      goto   OVERFLO             ; error exit if INT
MAINP
      bsf    STATUS,RP0   ; select Bank 1
; PORT A detects signal on Bit 3, communicates with PC on 0-2
; Bit 0 is serial data out (DATA)
; Bit 1 is valid data indicator (VALID)
; Bit 2 is input from PC saying it is ready (SEND)
      bcf    TRISA, DATO ; PORT A Bit 0 output (DATO)
      bcf    TRISA, VALID       ; PORT A Bit 1 output (VALID)
      bsf    TRISA, SEND ; PORT A Bit 2 input  (SEND)
      bsf    TRISA, SIGNAL      ; PORT A Bit 3 input  (signal)
      bcf      STATUS,RP0       ; select Bank 0
        clrf    BYTE1
      clrf   BYTE2
      clrf   BYTE3
      clrf   BYTE4        ; all counter bytes to 0

      bcf    PORTA,VALID ; no valid data for PC

      btfsc PORTA,SIGNAL       ; signal must NOT be on now
      goto   OVERFLO           ; error exit if it is
WAITI btfss PORTA,SIGNAL       ; wait for signal on PORTA
      goto   WAITI
      goto   INC1
INC4  incfsz       BYTE4,F            ; incr BYTE4 if BYTE3 over
      goto   INC1
      goto   OVERFLO           ; if BYTE4 over, error exit

INC3  incfsz      BYTE3,F           ; incr BYTE3 if BYTE2 over
      goto  INC1
      goto  INC4
INC2  incfsz      BYTE2,F           ; incr BYTE2 if BYTE1 over
GTI1  goto  INC1
      goto  INC3
INC1  btfss PORTA,SIGNAL      ; signal still on PORTA?
      goto  SERIAL            ; if gone, transmit results
      incfsz      BYTE1,F           ; incr&test
      goto  INC1        ; if nonzero
      goto  INC2        ; if zero overflow to BYTE2

; three-wire handshake with PC
; byte data is shifted out on PORTA DATO lowest->highest
SERIAL
      movf  BYTE1,W
      movwf BYTEO
      call  BYTOUT
      movf  BYTE2,W
      movwf BYTEO
      call  BYTOUT
      movf  BYTE3,W
      movwf BYTEO
      call  BYTOUT
      movf  BYTE4,W
      movwf BYTEO
      call  BYTOUT
      goto  MAINP

BYTOUT      movlw 8
      movwf COUNT8
WAITS btfss PORTA,SEND  ; wait for PC to allow send
      goto  WAITS       ; if clear check again
SHIFC bcf   STATUS,C    ; clear carry flag
      rrf   BYTEO,F           ; BYTEO into carry
      btfss STATUS,C    ; check carry bit
      goto  OUT0        ; if 0 output 0
      goto  OUT1        ; if 1 output 1
OUT0  bcf   PORTA,DATO
      goto  DEC8
OUT1  bsf   PORTA,DATO
DEC8  bsf   PORTA,VALID ; valid data is now there
WAITL btfsc PORTA,SEND  ; wait for send to be lowered
      goto  WAITL
      bcf   PORTA,VALID ; lower valid
      decfsz      COUNT8
      goto  WAITS
      return

OVERFLO comf      BYTE1,F           ; all bytes must be zero
      comf  BYTE2,F           ; set to FF for all
      comf  BYTE3,F           ; which means error
      comf  BYTE4,F
      goto  SERIAL            ; and output error

        END
```

LISTING 2—PHOTOGATE-SUPPORT PROGRAM

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <process.h>

/* SIGNAL* is not connected but reserve Pin 3 (D1) and A3 on PIC
   power to PIC through VDD_ON on pin 6 (D4)
   power to LED and transistor wiht TRANS_ON/LED_on pin 4 (D2)
   MCLR* on pin 5 (D3) normally high
   NORMAL OPERATION with Power and not reset (i.e. go) OR with POWGO (0x1C)
   The MSB of the printer Status port (S7*) or Pin 11 is used as input data
   Pin 10 (S6) is VALID signal from PIC that data is valid on Pin 11
   Pin 2 (D0) is SEND signal to PIC that PC is ready to receive
HAS BASIC CONVERSION ONLY 5 microsec per step
*/

#define DELAY     320           /* timing delay 320 for 166Pentium, 4Mhz */
#define TIMES     640000*16     /* length of time pulse if used */

#define SIGNAL    0x02
#define SEND      0x01
#define POWGO    0x1C

#define BYT 4         /*total number of bytes of data */
#define BIT 8        /*bits in 1 byte */

void main(void)
{
int dport_lpt1, sport_lpt1, bit, byt, columns, rows;
unsigned char shift_reg[BYT],temp;
char c;
long i,itemp;
float tempus;

/*Get LPT1 port addresses */

if(!(dport_lpt1 = peek(0x40,0x08)))
 { printf("\n\n\nLPT! not available... aborting\n\n\n"); exit(1); }
sport_lpt1 = dport_lpt1 + 1;  /* status port address */

/* Initialize the Printer DATA Port both for timing and SEND=0 */
outportb(dport_lpt1,0x00|POWGO);

/* Generate timing SIGNAL */
#undef USE_SIGNAL
#ifdef USE_SIGNAL
```

```
/* Generate timing SIGNAL */
#undef USE_SIGNAL
#ifdef USE_SIGNAL
outportb(dport_lpt1,SIGNAL|POWGO);
for(i=0;i<TIMES;i++) ; outportb(dport_lpt1,0x00|POWGO);
#endif USE_SIGNAL

for(byt=0; byt<BYT; byt++)  /* number of bytes */
  {
    for(bit=0; bit<BIT; bit++) /* bits / Switch */
      {
        outportb(dport_lpt1,SEND|POWGO); /* raise SEND */
       for(i=0;i<DELAY;i++);
       /* poll VALID i.e. bit 6 of STATUS register */
       while(!(inportb(sport_lpt1)&&0x40)) ;

      /* Read the Status Port bit S7* which is input data */

       temp = inportb(sport_lpt1) ;
       outportb(dport_lpt1, 0x00|POWGO); /* have data, lower SEND */
       temp ^= 0x80;      /*invert bit no. 7 since port inverts */
       temp &= 0x80;    /*mask all bits except bit 7 */

      /* Concatenate it in the variable */
       shift_reg[byt] >>= 1;       /* shift bit one right */
         shift_reg[byt]  &= 0x7f;    /* make MSB 0 */
       shift_reg[byt]   |= temp;    /* new bit in MSB */
       for(i=0;i<DELAY;i++);
      } /* for(bit */
  } /* for(byt */

for(byt=0; byt<BYT; byt++) printf("Byte %d = %02X\n", byt, shift_reg[byt]);

itemp=0;
itemp|=shift_reg[3]; itemp<<=8;
itemp|=shift_reg[2]; itemp<<=8;
itemp|=shift_reg[1]; itemp<<=8;
itemp|=shift_reg[0];
tempus=5.0e-6*itemp;
printf("physical time %f seconds\n",tempus);

printf("hit key to end\n"); while(!kbhit()) ; exit(0);
}
```

Optocoupler simplifies power-line monitoring

Alfredo del Rio and Ana Cao y Paz, University of Vigo, Spain

The use of a linear optocoupler and a capacitor-based power supply yields a simple, yet precise power-line-monitoring system. The circuit in **Figure 1** converts the 110V-ac power-line voltage to an ac output voltage centered at 2.5V, covering 0 to 5V. The circuit isolates the output signal from the power line. You can connect the output directly to an A/D converter. For other power-line voltages, simply change the value of R_1. For a power-line voltage of 220V ac, use a value of 470 kΩ for R_1. The input stage is a nonisolated block that uses the neutral line as a ground reference. This block receives power from a capacitor-based power supply that provides a stabilized 5V-dc voltage and a 3.3V dc reference. The TLC2272 op amp, IC_1, and the TLC2272 linear optocoupler, IC_3, form a feedback amplifier in which the I_{P1} current is proportional to the input voltage, V_{IN}.

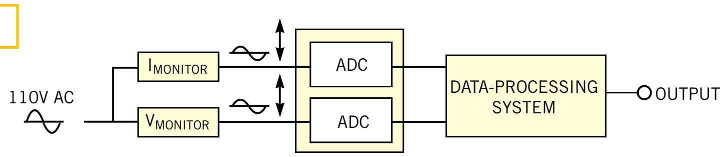
TABLE 1—OUTPUT OFFSET-VOLTAGE DRIFT

TIL300 (°C)	V_{OUT} (V)	TIL300 (°C)	V_{OUT} (V)
17.5	2.496	37.5	2.506
20	2.497	40	2.507
22.5	2.498	42.5	2.509
25	2.5	45	2.51
27.5	2.501	47.5	2.512
30	2.503	50	2.513
32.5	2.504	52.5	2.515
35	2.505		

Resistor R_2 adds a dc offset current to allow for both polarities in V_{IN}. The match between the two photodiodes in the IL300, IC_2, ensures that I_{P2} is closely proportional to I_{P1}. The output stage converts I_{P2} to a voltage level isolated from the power line. Variable resistor VR_2 trims the overall gain, and VR_1 adjusts the output-voltage offset, which is nominally 2.5V. You can test this circuit using simulation the model in **Listing 1** for IC_2. Typical values for K_1 and K_2 (optical transfer ratios) are approximately 0.007. The global optical transfer ratio is $K_3=K_2/K_1$. After performing the simulation, you can build and test a prototype. The power supply for the isolated block provides 5V dc and a 3.3V reference from an available voltage of 7 to 10V. You do not need the regulated 5V if that voltage is already available in your system.

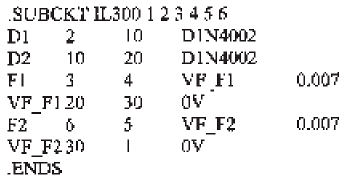
An important goal in this design is to obtain a stable dc voltage at the output. This property is crucial for dc measurements of V_{IN}. Even if you suppose the ac power line to be free of dc voltage, some types of loads drain dc currents, thereby introducing a small dc voltage because of voltage drops in the ac lines. Thermal drifts in the output voltage stem principally from drifts in K_3. In tests of the prototype, the K_3 temperature coefficient was 470 ppm/°C. **Table 1** shows V_{OUT} at different temperatures. The TLC2272 op amp has rail-to-rail output, yielding a wide output-voltage range, and low quiescent current, simplifying the capacitor-based power supply. Because the TLC2272 is a dual device, you can connect the unused half as a voltage follower. When you monitor a three-phase power line, you'd use one and one-half TLC2272s. Note that the op amps in the isolated block, IC_3, and the nonisolated block, IC_1, cannot be halves of the same chip; otherwise, you'd lose the isolation.

Figure 2



By adding two ADCs and a microcontroller, you can measure power-line voltage and current parameters.

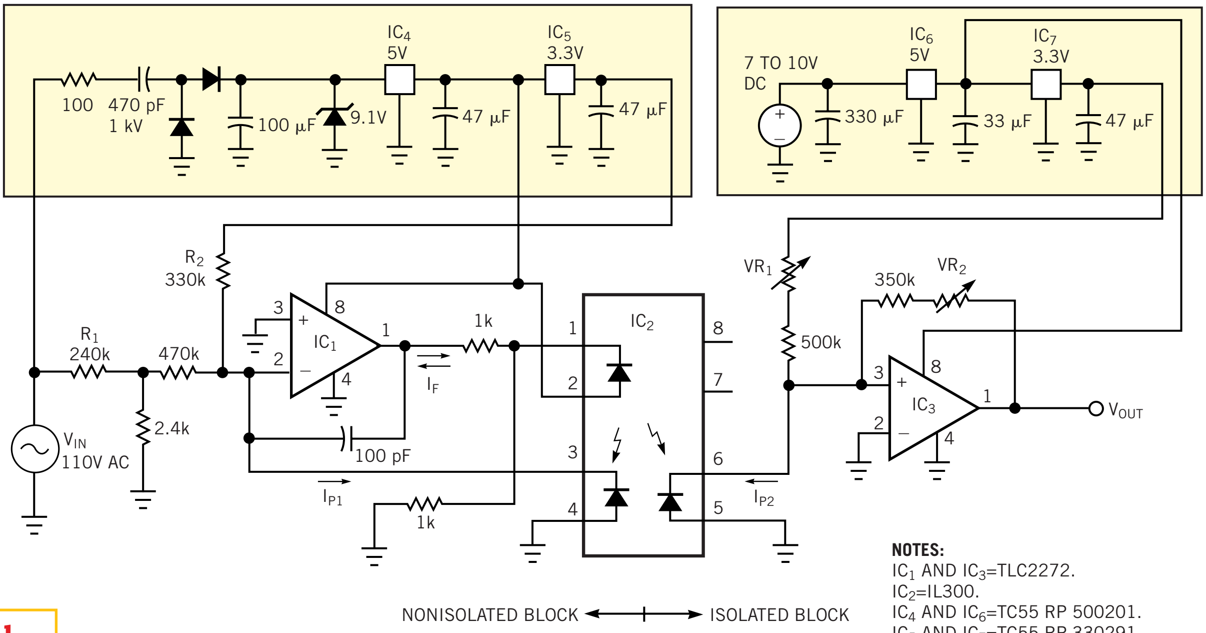
LISTING 1—SIMULATION MODEL



```
.SUBCKT IL300 1 2 3 4 5 6
D1      2       10      D1N4002
D2      10      20      D1N4002
F1      3       4       VF_F1           0.007
VF_F1 20        30      0V
F2      6       5       VF_F2           0.007
VF_F2 30        1       0V
.ENDS
```

The main specifications of the circuit are 5300V-ac-rms galvanic isolation, 0.08% linearity, 470-ppm/°C thermal shifts in V_{OUT}, 2° phase shift at 50 Hz, and dc to 1-kHz bandwidth at −3 dB. If you connect the output to a 10-bit A/D converter, one LSB is equivalent to 0.5V in the 110V power line. You can add a Hall-effect sensor to the circuit for current measurements. The LTS series from LEM (www.lemusa.com) is suitable for this purpose, because these devices operate from a single 5V supply and provide a 2.5V-centered output. **Figure 2** shows a system that integrates voltage and current measurements. The processor computes true-rms voltages and currents, apparent and active power, and power factor.

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



NOTES:
IC_1 AND IC_3=TLC2272.
IC_2=IL300.
IC_4 AND IC_6=TC55 RP 500201.
IC_5 AND IC_7=TC55 RP 330291.

Figure 1

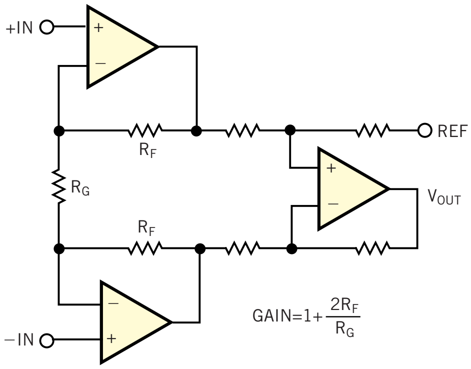
An isolated optocoupler circuit allows you to make dc measurements of the power-line voltage.

Improved amplifier drives differential-input ADCs

Stephan Goldstein, Analog Devices, Wilmington, MA

ADCs WITH differential inputs are becoming increasingly popular. This popularity isn't surprising, because differential inputs in the ADC offer several advantages: good common-mode noise rejection, a doubling of the available dynamic range without doubling the supply voltage, and cancellation of even-order harmonics that accrue with a single-ended input. But the differential input structure doesn't eliminate the frequent need for additional gain between the signal source and the ADC. A frequently used gain stage is the classic, three-op-amp instrumentation amplifier (**Figure 1**). This popular circuit offers excellent common-mode rejection and high input impedance. The circuit also has an output-reference (ground-sense) terminal, allowing you to reference the output voltage to a voltage other than ground. However, this circuit has a single-ended output (relative to the reference terminal), so it's a poor match for a differential-input ADC.

Figure 1

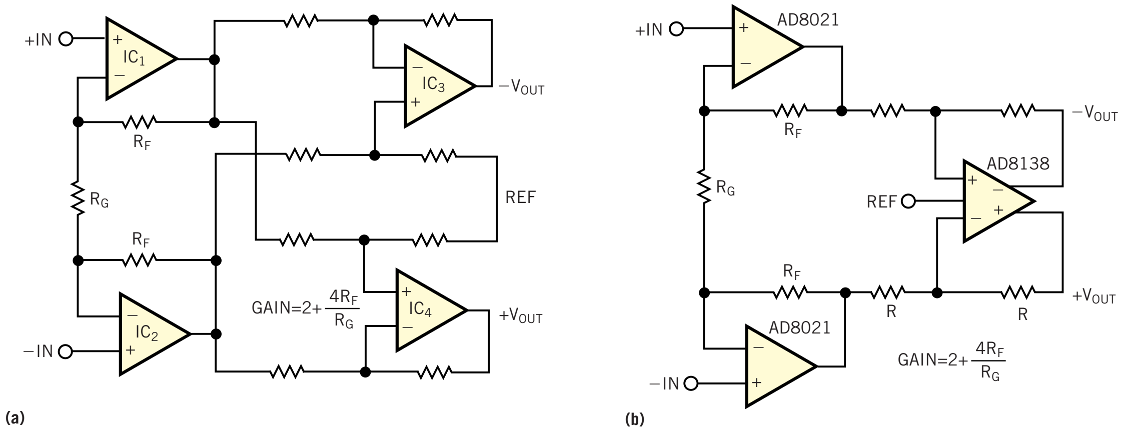


The classic three-op-amp instrumentation amplifier does not provide differential outputs.

Figure 2 shows two easy ways to create a differential-input instrumentation amplifier. In **Figure 2a**, IC$_4$ and its associated feedback resistors are connected in parallel with the original output amplifier but with inverted polarity relative to the original circuit. The two outputs together provide the desired function, but the circuit requires many matched resistors. Furthermore, the common-mode reference input could require several milliamperes of drive, depending on the resistor values and voltages involved. However, the circuit does the job, and you can build it by using a high-quality quad op amp and a handful of resistors. **Figure 2b** shows a more efficient and elegant approach, using only the four resistors required in the original output stage. In this circuit, a modern, fully differential op amp, such as the AD8138, replaces IC$_3$ and IC$_4$ in **Figure 2a**. The amplifier's two outputs swing symmetrically about its high-impedance, common-mode reference input. The differential outputs provide a clean, simple interface to a differential-input ADC.

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

Figure 2



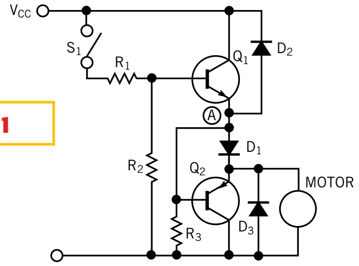
The four-op-amp instrumentation amplifier (a) provides differential outputs but requires many matched resistors. A differential-output op amp (b) reduces the IC count in Figure 2a to three.

Circuit forms dc-motor switch with brake

JB Guiot, DCS AG, Allschwil, Switzerland

Controlling a small dc motor without speed control sounds like a trivial task; a switch or a relay should suffice. However, several problems accompany this approach. For one, the switch, because of the inductive load and the low starting resistance of the motor, tends to wear out prematurely (with all the related sparks and EMI problems). Second, when you cut the power, the motor continues to rotate for a certain time, depending on its initial speed and inertia. The circuit in **Figure 1** can be useful for designs that don't need precise control of speed and stopping position but can benefit from enhanced deceleration. The circuit comprises two parts. Q_1 plays the role of the switch. D_2 protects Q_2 against inductive surges. Resistor R_2 keeps Q_1 off as long as switch S_1 is open. R_1 limits the base current of Q_1 when S_1 is closed. S_1 can be a manual switch, a relay contact, an optocoupler, or a transistor. If you close S_1, Q_1 turns on, and the motor runs.

Figure 1



This circuit provides both motor-drive and braking functions.

Q_2, D_1, and R_3 constitute the braking circuit. This circuit is similar to the output circuit of TTL gates. D_3 protects Q_2 from inductive surges. When S_1 closes, Q_1 turns on, and the voltage at Point A goes high (near V_{CC}). The voltage at the base of Q_2 is higher than the voltage at the emitter, because of the voltage drop in D_1. If you open S_1 while the motor is running, Q_1 turns off. The voltage at Point A is near zero. The self-induced, back-EMF voltage from the motor sees a short circuit in Q_2, whose emitter is more positive than its base and thus conducts. Short-circuiting the motor results in braking it. The higher the speed of the motor, the stronger the braking effect.

You should mount the circuit of Q_2 as near as possible to the motor to reduce the series resistance of the wiring. This parasitic resistance limits the braking current and, thus, the deceleration. The circuit of Q_1 can be remote. The dividing line between the two circuits is at Point A. This design mounts the circuit on the tool-changer motors of small machine tools, and it has worked perfectly for years. The values of the components are not critical. The transistors should preferably be Darlington pairs and, like the diodes, should be types commensurate with the power-supply voltage and the motor current. (Also, don't forget the high inductance of the motor.) The components in **Figure 1**, for example, are suitable for a 24V, 3.5A motor.

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