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 \( V_{IN} - V_K = V_N(1 - e^{-T/RC}) \), 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, \( CV_2 \), 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 \( R_1 \).
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, $V_{VK}$, 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

<table>
<thead>
<tr>
<th>Input Voltage</th>
<th>Output Voltage</th>
<th>Input Voltage</th>
<th>Output Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>3.324</td>
<td>8</td>
<td>5.782</td>
</tr>
<tr>
<td>3.5</td>
<td>3.667</td>
<td>8.5</td>
<td>5.861</td>
</tr>
<tr>
<td>4</td>
<td>3.954</td>
<td>9</td>
<td>5.886</td>
</tr>
<tr>
<td>4.5</td>
<td>4.227</td>
<td>9.5</td>
<td>5.945</td>
</tr>
<tr>
<td>5</td>
<td>4.506</td>
<td>10</td>
<td>6.098</td>
</tr>
<tr>
<td>5.5</td>
<td>4.705</td>
<td>10.5</td>
<td>6.187</td>
</tr>
<tr>
<td>6</td>
<td>4.956</td>
<td>11</td>
<td>6.204</td>
</tr>
<tr>
<td>6.5</td>
<td>5.151</td>
<td>11.5</td>
<td>6.312</td>
</tr>
<tr>
<td>7</td>
<td>5.315</td>
<td>12</td>
<td>6.371</td>
</tr>
<tr>
<td>7.5</td>
<td>5.444</td>
<td>12.5</td>
<td>6.378</td>
</tr>
<tr>
<td>8</td>
<td>5.615</td>
<td>13</td>
<td>6.476</td>
</tr>
</tbody>
</table>

**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 $\mu$sec, so the timing loop is 5 $\mu$sec long. You can run newer PIC16F84s with a 20-MHz clock, so, in principle, the timing loop can be 1 $\mu$sec 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.
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 does not 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 PIC raises SEND and polls VALID. In response, the PIC puts DATA on the line. The PIC then 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.


---

**LISTING 1—ZERO-POWER PHOTOGATE-ASSEMBLY PROGRAM**

```assembly
# hardware connections needed are as follows:
# PIC parallel port connections (see below for software aspect)
# DATO pin11, VALID pin 10, SEND pin 2, SIGNAL* is not used but reserve pin 3
# (see below for software aspect) and
# Vcc is output from parallel port pin 5 causing reset if low
# Vcc is output from parallel port pin 6 for power
# include "D:\pic\msp430\flash4.asm"

CONFIG (XT=OSC & _XT OFF & _PWRT OFF & _CP OFF)

EQUATES
BANK16 = equ 32H
DATO  = equ 0
SEND   = equ 2
SIGNAL  = equ 3

VARIABLES
block BANK16
BYTE0
BYTE2
BYTE3
BYTE4
COUNTS
end

org 0000H
goto MAIN
;
org 0012H ; should never get here

; MAISP
dsf STATUS, R0 ; select bank 1
; PORT A detects signal on bit 3, communicates with PC on 0-2
; B0 is serial data output (DATA)
; Bit 3 is valid data indicator (VALID)
; Bit 2 is input from PC saying it is ready (SEND)
; bcf TRISA, DAT0 ; PORT A Bit 0 output (DAT0)
; bcf TRISA, Valid ; PORT A Bit 0 output (DAT0)
; bcf TRISA, SEND ; PORT A Bit 2 input (SEND)
; bcf TRISA, SIGNAL ; PORT A Bit 3 input (signal)
; bcf STATUS, R0 ; select bank 0
; clrf BYTE2
; clrf BYTE3
; clrf BYTE4
; all counter bytes to 0
; bcf PORTA, VALID ; no valid data for PC

; bcf PORTA, SIGNAL
; goto OVERFLOW
; ; signal must NOT be on now

; WAITI bcf PORTA, SIGNAL
; goto END

; INCX incx BYTE2, F ; incr BYTE2 if BYTE3 over
; goto INC1

; INC1 incx BYTE2, F ; incr BYTE2 over, error exit

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

; END
```

---

*Note: The above code is a simplified representation of the actual assembly code provided in the document. The full code includes more detailed instructions and error handling routines.*
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 R1. For a power-line voltage of 220V ac, use a value of 470 kΩ for R1. 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, IC5, and the TLC2272 linear optocoupler, IC3, form a feedback amplifier in which the IIO current is proportional to the input voltage, VIN.

Resistor R2 adds a dc offset current to allow for both polarities in VIN. The match between the two photodiodes in the IL300, IC2, ensures that IPP is closely proportional to IIO. The output stage converts IPP to a voltage level isolated from the power line. Variable resistor VR2 trims the overall gain, and VR1 adjusts the output-voltage offset, which is nominally 2.5V. You can test this circuit using the model in Listing 1 for IC2. Typical values for K1 and K2 (optical transfer ratios) are approximately 0.007. The global optical transfer ratio is K3 = K2/K1. 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, $IC_4$, cannot be halves of the same chip; otherwise, you’d lose the isolation.

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.

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 2 shows two easy ways to create a differential-input instrumentation amplifier. In Figure 2a, IC3 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 milliampères 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 IC3 and IC4 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.

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. Q₁ plays the role of the switch. D₂ protects Q₂ against inductive surges. Resistor R₂ keeps Q₁ off as long as switch S₁ is open. R₁ limits the base current of Q₁ when S₁ is closed. S₁ can be a manual switch, a relay contact, an optocoupler, or a transistor. If you close S₁, Q₁ turns on, and the motor runs.

Q₂, D₁, and R₃ constitute the braking circuit. This circuit is similar to the output circuit of TTL gates. D₁ protects Q₂ from inductive surges. When S₁ closes, Q₁ turns on, and the voltage at Point A goes high (near VCC). The voltage at the base of Q₂ is higher than the voltage at the emitter, because of the voltage drop in D₁. If you open S₁ while the motor is running, Q₁ turns off. The voltage at Point A is near zero. The self-induced, back-EMF voltage from the motor sees a short circuit in Q₂, 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 Q₂ 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 Q₁ 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.