

Edited by Bill Travis and Anne Watson Swager

## Keypad sequence activates relay

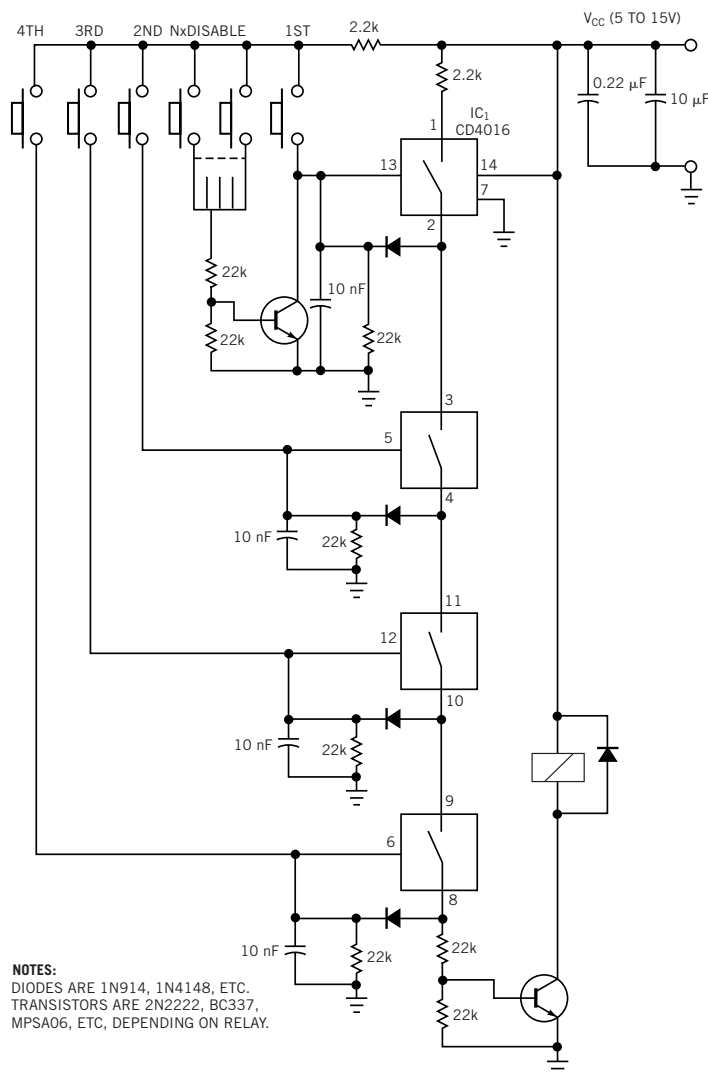
JB Guiot, DCS AG, Allschwil, Switzerland

**T**ODAY, IT SEEMS that  $\mu$ Ps assume even the easiest tasks, such as activating a relay (for an alarm or access control, for example) via a keyboard command. However, a simpler and less expensive solution is available (Figure 1). Moreover, the method does not entail programming and the concomitant, almost inevitable, debugging. A CD4016 CMOS analog switch, IC<sub>1</sub>, forms the heart of the system. Depressing the key labeled “1st” closes the topmost switch (between pins 1 and 2), which latches itself via the diode connected between pins 2 and 13. The same topology repeats three times, with the keys labeled “2nd,” “3rd,” and “4th.” Each switch provides power to the next one, so that depressing the key labeled “3rd” before the “1st” or “2nd” key does not activate the third switch. The Disable keys, connected in parallel, disable the entire chain by shorting the control input of the first switch to ground. The values of the power-supply voltage and the components are not critical; you can use almost anything you find in your drawer. Take care, however, that the output transistor has a rating that can han-

dle the relay. The entire circuit fits onto a 50×50-mm board, which you can easily mount on the back of the keypad.

**Is this the best Design Idea in this issue?** Vote at [www.ednmag.com/ednmag/vote.asp](http://www.ednmag.com/ednmag/vote.asp).

Figure 1



Keypad sequence activates relay .....	189
Spice models differential stripline.....	190
System monitors multiple temperatures, controls fan speed.....	192
Measure temperature in remote locations.....	194
PC's IRQ7 and INT1Ch measure currents, charges.....	196
Simple circuit provides high-side current sensing .....	200

Forget  $\mu$ Ps; this circuit uses a simple, sequential key press to activate a relay.

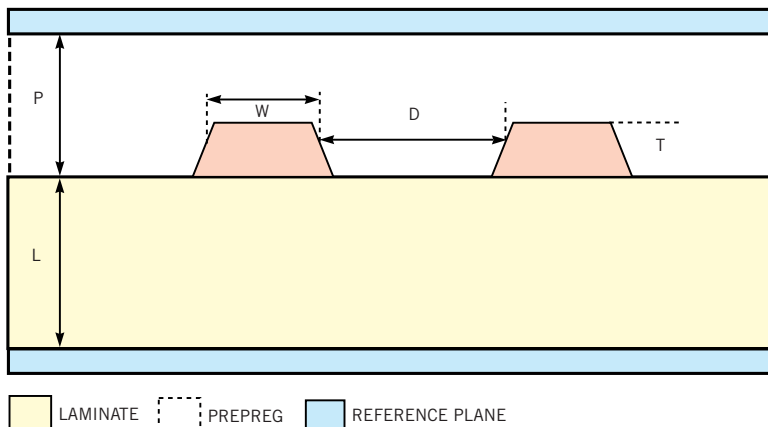
# Spice models differential stripline

Vittorio Ricchiuti, Siemens ICN, L'Aquila, Italy

**D**IFFERENTIAL SIGNALING IS a common technique for obtaining high noise immunity for critical signals in high-speed digital pc-board systems. The pc board's traces, carrying differential signals, are often edge-coupled offset striplines, with the traces sandwiched between two reference planes (Figure 1). Striplines are desirable because they offer good protection against EMI and ESD and because they allow you to tightly control tolerances in fabrication. Unfortunately, PSpice does not provide a geometry-characterized model of two edge-coupled striplines that carry differential signals. As a result, when it is necessary to simulate a differential stripline, you must use the PSpice Tline-coupled model with the line parameters (L, C, Lm, Cm) obtained by means of a 2-D field solver. Listing 1 shows the PSpice subcircuit that represents the model of two lossless, differentially routed striplines, as in Figure 1.

You can use the subcircuit for both transient and ac analysis. The model's input parameters are the geometrical dimension of the striplines in meters; the relative permittivity,  $\epsilon_r$ , of the surrounding medium; the trace length in meters; and the correction factor, kc (Reference 1). The model calculates the odd-mode impedance of each stripline (in other words, the impedance of a single stripline when the two striplines carry differential signals) with a maximum error of 5 to 6%. The model also calculates the propagation delay. The odd-mode impedance of each line is the parallel combination of two odd-mode impedances, each one calculated with respect

Figure 1



Spice can model edge-coupled offset striplines for impedance and propagation delay.

to each reference plane. The empirically determined correction factor, kc, takes into account the imperfection of the formulas in the model in determining the

odd-mode impedance of each line. The correction factor is a function of the pc-board material and its supplier. You can download Listing 1 from EDN's Web site, [www.ednmag.com](http://www.ednmag.com). Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2599.

## LISTING 1—PSPICE MODEL FOR DIFFERENTIAL STRIPLINE

```
.SUBCKT diff_stripline a1 a2 b1 b2 params: W=1u D=1u T=1u L=1u P=1 kc=1
+ er=4.6 LEN=1
.func Pi() {4*atan(1)}
.func A(x) {1+log(1+1/tanh(Pi()*D/(4*x+2*T)))/log(2)}
.func C(x) {2*log(2+T/(2*x))-T/(2*x+T)*log(T*(4*x+T)/(4*x**2))}
.func Z1(x) {60*Pi()*x/((sqrt(kc*er))*(W+(2*x+T)*C(x)*A(x)/(2*Pi())))}
T_T1 a1 0 b1 0
+Z0={2*Z1(L)*Z1(P-T)/(Z1(L)+Z1(P-T))}
+TD={(sqrt(er)/3.0e8)*LEN}
T_T2 a2 0 b2 0
+Z0={2*Z1(L)*Z1(P-T)/(Z1(L)+Z1(P-T))}
+TD={(sqrt(er)/3.0e8)*LEN}
.ENDS diff_stripline
```

## REFERENCE

1. Riscchiuti, Vittorio, "Propagation of High-Speed Digital Signals on Printed Circuit Boards," *Future Circuits International*, Issue 5, June 1999.

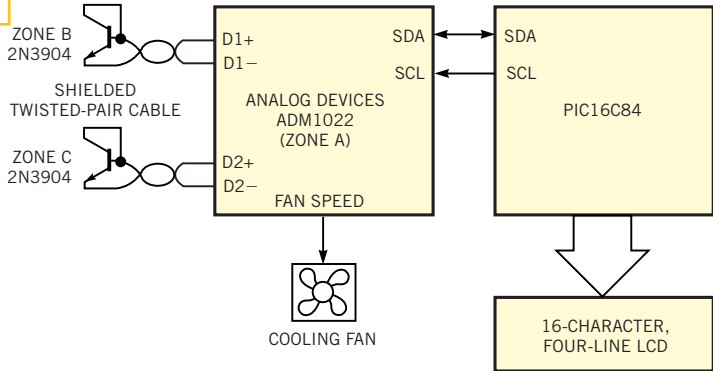
**Is this the best Design Idea in this issue? Vote at [www.ednmag.com/ednmag/vote.asp](http://www.ednmag.com/ednmag/vote.asp).**

# System monitors multiple temperatures, controls fan speed

David Hanrahan, Analog Devices Inc, Limerick, Ireland

**T**HE BLOCK DIAGRAM in **Figure 1** represents a complete remote-temperature-sensing and fan-control system. The system uses an Analog Devices temperature-monitor and fan-control ASIC and a PIC16C84  $\mu$ C from Microchip Technology. The ADM1022 allows you to measure the local temperature and two remote temperatures within a system. An on-chip, 8-bit DAC controls the speed of a cooling fan in response to the measured temperature. This circuit can form the basis of a central-heating/air-conditioning system with minimal component count and cost. The ADM1022 uses TDM (thermal-diode-modeling) techniques to accurately sense temperature. The use of readily available transistors, such as the 2N3904, eases temperature monitoring. The temperature-sensing elements remotely connect to the ADM1022, using a shielded

**Figure 1**



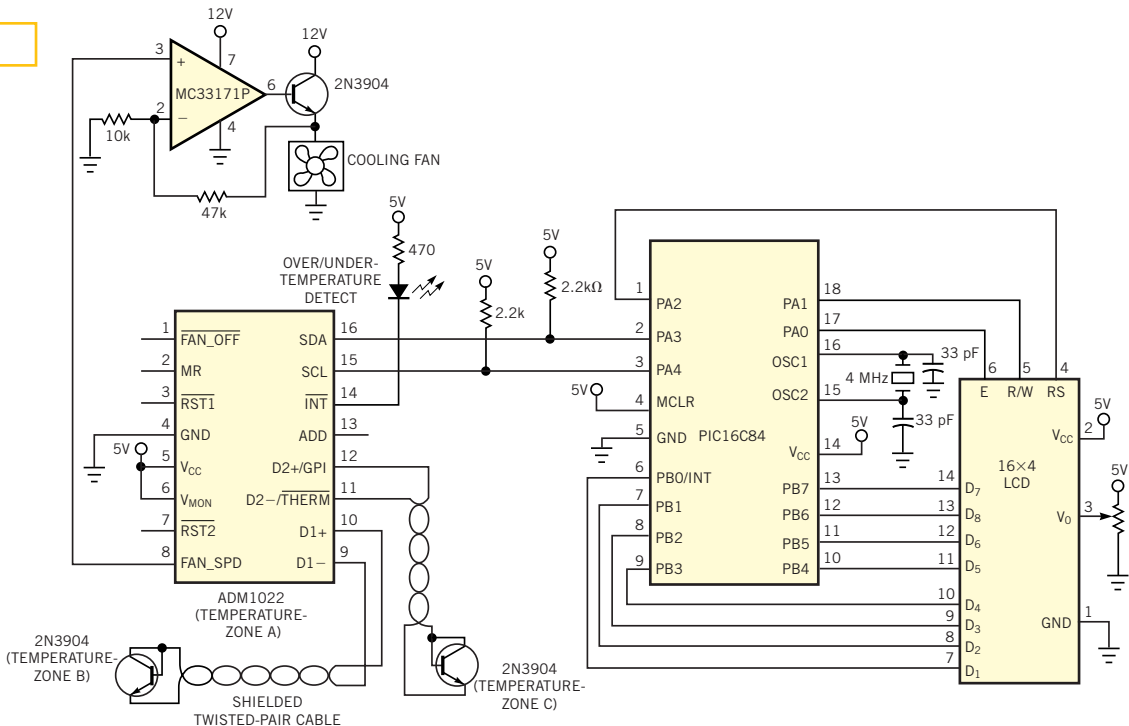
**An ASIC and a  $\mu$ C monitor three temperature zones and control a cooling fan.**

twisted-pair cable. Zone A represents ambient temperature; the internal bandgap temperature sensor in the ADM1022 measures this temperature. Writing to the

on-chip DAC controls the speed of the cooling fan.

**Figure 2** shows a complete schematic of the system. The heart of the unit is the

**Figure 2**



**A temperature-monitor/fan-control circuit uses minimal components and space.**

PIC16C84  $\mu\text{C}$ . The PC software-based communication system uses “bit-banging” of two of the pins. A 16-character-by-four-line LCD displays all measured values. The  $\mu\text{C}$  reads the temperature data from the ADM1022 via the PC bus. Zone A represents the ambient temperature. Zones B and C represent temperatures at a distance from the system and

use shielded twisted-pair cable. If any temperature goes outside the programmed limits, the over/undertemperature-detection LED lights up. In the schematic, the ADM1022 drives a 12V fan. You can substitute other fans or actuators by providing suitable drive circuitry. The three-state ADD pin can be high, low, or floating. Thus, as many as

three ADM1022s can connect to one  $\mu\text{C}$ , allowing you to easily expand the system to monitor nine temperature zones.

**Is this the best Design Idea in this issue?** Vote at [www.ednmag.com/ednmag/vote.asp](http://www.ednmag.com/ednmag/vote.asp).

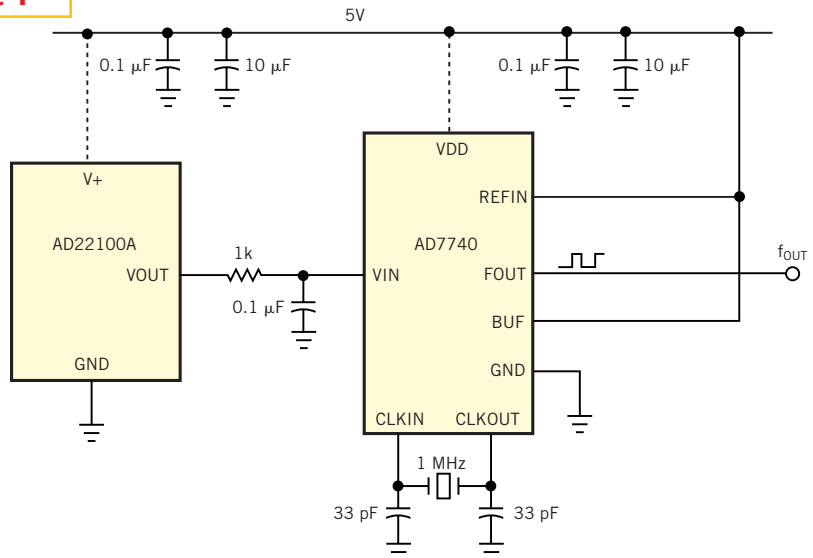
## Measure temperature in remote locations

Helen Stapleton, Analog Devices, Limerick, Ireland

**T**HE COMPACT AND low-power temperature-to-frequency converter in **Figure 1** is ideally suited for measuring temperature in remote locations. An AD22100A temperature sensor generates a voltage proportional to the ambient temperature. This voltage then drives an AD7740 VFC (voltage-to-frequency converter), which in turn generates a digital pulse train. The duty cycle of this pulse train is directly proportional to the ambient temperature. A 1-MHz crystal defines the full-scale frequency on  $f_{\text{OUT}}$ . In noisy environments, you can use a single optocoupler to feed  $f_{\text{OUT}}$  back to the host computer. The optocoupler provides more than 2 kV of isolation between the transducer and the host. The host counts the  $f_{\text{OUT}}$  pulses. The resolution of this system is a function of the number of  $f_{\text{OUT}}$  pulses counted for each temperature reading. A count interval of  $2^N/f_{\text{OUT}}$  (maximum) corresponds to N-bit resolution. Hence, a trade-off exists between resolution and conversion time. The synchronous nature of the AD7740 produces a more temperature-stable transfer function than asynchronous VFCs, which are prone to errors introduced by an external capacitor.

Both the temperature sensor and the VFC operate from the 5V supply. Connecting the REFIN of the AD7740 to the same supply eliminates the need for an external precision reference. Because the circuit is fully ratiometric, the outputs of both the temperature sensor and the VFC scale with the supply voltage, and any errors caused by supply variations cancel

**Figure 1**



**This remote-temperature measurement system is immune to power-supply variations.**

each other out. The AD22100A temperature sensor operates over  $-40$  to  $+85^\circ\text{C}$ . The corresponding output-voltage range is nominally 0.475 to 3.288V. The AD7740 converts this voltage range to a frequency range of 176 to 626 kHz. The transfer functions of these devices are as follows:

AD22100A:  

$$V_{\text{OUT}} = V_{\text{DD}}/5 \times [1.375 + (22.5 \text{ mV}/^\circ\text{C} \times T_A)] \text{ V.}$$

AD7740:  

$$f_{\text{OUT}} = \text{CLKIN} \times [0.1 + (0.8 \times V_{\text{IN}}/V_{\text{DD}})] \text{ Hz.}$$

For the circuit in **Figure 1**,  $f_{\text{OUT}} = (320 + 3.6T_A)$  kHz. The AD22100A is

available in a TO-92 package, and the AD7740 comes in an eight-lead SOT-23 package. The ICs require minimal external components. When  $V_{\text{IN}}$  is buffered (BUF is at logic 1), the power consumption in the two ICs is typically 8 mW. This figure does not include the power consumed by the crystal, which is a function of the effective series resistance and the associated capacitor values.

**Is this the best Design Idea in this issue?** Vote at [www.ednmag.com/ednmag/vote.asp](http://www.ednmag.com/ednmag/vote.asp).

# PC's IRQ7 and INT1Ch measure currents, charges

K Suresh, IGCAR, Tamil Nadu, India

**F**IGURE 1 SHOWS THAT YOU can use just two interrupts (IRQ7 and INT1Ch) of a PC and a few inexpensive components to make simultaneous measurements of two electrical quantities. Two examples are low currents and their associated charges, parameters important in ion implanters used in the semiconductor industry. The method shown here effects the simultaneous measurement of the interrelated parameters without the need for two dedicated, multiplexed analog-input channels of a data-acquisition system. Simple and simultaneous counting of the number of IRQ7 interrupts in two modes—one periodic and the other totalizing—gives the magnitude of the current and its associated charge. You can use this design as a low-current ammeter, a coulomb meter, or a current/coulomb meter with simple modifications to the software. The circuit in **Figure 1** provides simultaneous measurement of input currents of 0 to 100  $\mu$ A and the corresponding charges. Because the currents under measurement are low, the circuit uses an electrometer amplifier with low bias cur-

rent and noise current. The amplifier converts the input current to voltage as follows:

$$V_{OUT} = (I_{IN} \times R_F) V.$$

IC<sub>2</sub> inverts the amplifier's output and then undergoes digital integration, using a simple and inexpensive digital integrator, IC<sub>3</sub>, whose output is a 10-kHz/V pulse train. Each pulse represents a fixed charge, Q, as follows:

$$F_{OUT} = \frac{V_{OUT}}{10(R_1 + R_2)C} \text{ Hz.}$$

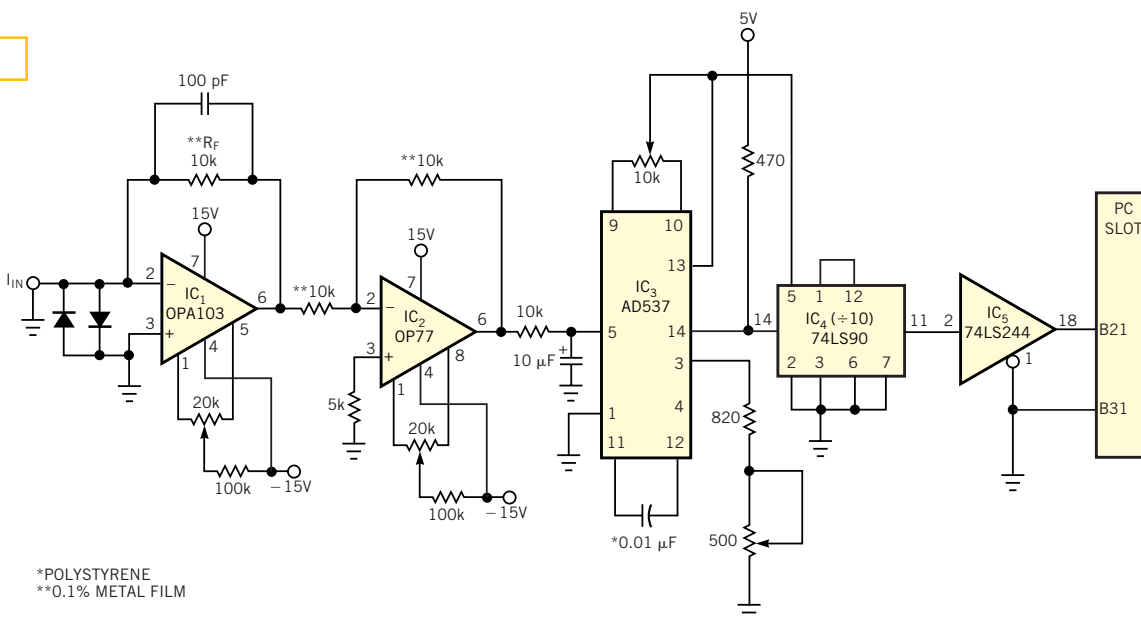
The output pulses of the digital integrator, after a division by 10 in IC<sub>4</sub> and buffering by IC<sub>5</sub>, interrupt the PC through IRQ7 to a maximum rate of 1000 pulses/sec. Access to IRQ7 is via the LPT printer port 10 ( $\overline{ACK}$ ) for testing the design. However, you can also directly plug pin B21 into the PC's slot, as in **Figure 1**. Each IRQ7 interrupt represents a fixed, 100-nC charge when counted in totalizing mode over a period, T. The following equation gives the total charge associated with the current in period T:

$$\int_0^T v(t) dt = \int_0^T i(t) R_F R_F dt =$$

$$k \int [dN(t)/dt] dt = kN,$$

where N is the total count during the integration period, T, and k is the charge/count-conversion factor. However, if you periodically count the IRQ7 interrupts—say, at 1-sec intervals—you obtain the magnitude of the input current. Though the circuit shown here is designed for positive input currents, you can process bipolar input currents by using an absolute-value circuit before the digital integrator. The Turbo C program in **Listing 1** controls the entire measurement. BIOS interrupt INT1Ch, which occurs 18.2 times/sec (normally used only for time-of-day data), generates the 1-sec timebase for the periodic counting of the IRQ7 interrupts. At the start, the variables ITIMER, QTIMER, ICOUNT, QCOUNT, ETIME, and TEMP are set to zero. The INT1Ch signal causes execution of the TIMEBASE() routine to update the ITIMER and QTIMER variables, and the IRQ7 interrupt causes execution

**Figure 1**



You can use a PC's interrupts to measure low currents and their associated charges.

of IROUTINE(), which updates the QCOUNT, ICOUNT, and TEMP variables. Subsequently, when the ITIMER reaches 18 (approximately 0.989 sec) and a correction factor (CF) of 18.2/18 adjusts the timebase to 1 sec, the DISPLAY() routine manipulates ICOUNT and TEMP with Q and the CF to display the current magnitude and associated charge, as follows:

$$\text{CURRENT} = (Q \times \text{ICOUNT} \times \text{CF}) / 1000 \mu\text{A}$$

$$\text{CHARGE} = (\text{TEMP} \times Q \times \text{CF}) / 1000 \mu\text{C}$$

The routine then resets the variables ITIMER, ICOUNT, and TEMP to zero in preparation for another measurement cycle. However, the system continues to update QTIMER and QCOUNT in totalizing mode with each INT1Ch and IRQ7, respectively. At any instant, if you wish to know the total charge associated with the input current, you can press any key to obtain the total charge QTOTAL and the elapsed time, as given here:

$$\text{QTOTAL} = (\text{QCOUNT} \times Q) / 1000 \mu\text{C}$$

$$\text{ETIME} = (\text{QTIMER} \times 54.95) / 1000 \text{SEC}$$

You can download **Listing 1** from EDN's Web site, [www.ednmag.com](http://www.ednmag.com). Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2613.

**Is this the best Design Idea in this issue?** Vote at [www.ednmag.com/ednmag/vote.asp](http://www.ednmag.com/ednmag/vote.asp).

### LISTING 1—TURBO C ROUTINE FOR CURRENT AND CHARGE MEASUREMENT

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

#define TIMERINTR 0X1C    /*Timer Interrupt*/
#define IRQ7 0X0F        /*IRQ7 interrupt*/

/*Global Variables*/
float Q =100.0;
int ITIMER=0,QTIMER,ICOUNT=0,QCOUNT=0;
float CF, CURRENT=0.0,ETIME=0.0,CHARGE=0.0;
long float QTOTAL=0.0
int TEMP=0;

static int EXISTINGIMR,OURIMR;

void interrupt (*oldfunc)(void); /* IRQ7 interrupt function pointer */
void interrupt IROUTINE();
void interrupt (*oldvect)(void); /*Timer Tick INT1Ch pointer */
void interrupt TIMEBASE();

void interrupt IROUTINE() /* ISR for IRQ7 */
{
    disable();
    TEMP++;
    QCOUNT ++;
    ICOUNT++;
    outp(0x20,0x20); /* End of Interrupt */
    enable();
}

void interrupt TIMEBASE() /* ISR for INT1Ch & Routine for 1Sec time
base
generation
*/
{
    disable();
    ITIMER++;
    QTIMER++;
    enable();
}

void INITIALISEIRQ7()
{
    disable();
    EXISTINGIMR=inportb(0x21);
    OURIMR=EXISTINGIMR & 0X7F; /*Enable IRQ7 to 8259 PIC*/
    outportb(0x21,OURIMR);
    oldfunc=getvect(IRQ7);
    setvect(IRQ7,IROUTINE);
    enable();
}

void INITIALSETIMER()
{
    disable();
    oldvect=getvect(TIMERINTR);
    setvect(TIMERINTR,TIMEBASE);
    enable();
}

void RESTOREALL()
{
    disable();
    outportb(0x21,EXISTINGIMR);
    setvect(TIMERINTR,oldvect);
    setvect(IRQ7, oldfunc);
    enable();
}

void DISPLAY()
{
    disable();
    CURRENT= (Q*ICOUNT*CF)/1000.0; /*Calculate Current from ICOUNT*/
    Gotoxy(15,15);
    printf("\nCURRENT=%6.2f uA",CURRENT); /*Display the current
measured*/
    CHARGE= (TEMP*Q*CF)/1000.0; /*Calculate Charge from
TEMP*/
    printf("\nCHARGE = %6.2f uC", CHARGE);/*Display the charge
measured*/
    enable();
}

void main(void) /* Main program starts here */
{
    int c,a,lpt;
    float CF;
    CF=18.2/18; /*Correction Factor for time base*/
    clrscr();

    lpt=peek(0x40,0x08);
    printf("\nPC'sIRQ7,INT1Ch provide simultaneous measurement of ");
    printf("\nlt Low Currents and Associated Charges");
    printf("\nltlt by");
    printf("\nlt K.SURESH, MSD,IGCAR,Kalpakkam,TamilNadu,India 603
102");
    outp(lpt+2,(inp(lpt+2)|0x10));/*Enable IRQ7 to the printer adapter*/
    outp(0x20,0x20);
    INITIALISEIRQ7();
    INITIALSETIMER();

    while(!kbhit())
    {
        if (ITIMER>=18)
        {
            DISPLAY();
            ITIMER=0;
            ICOUNT=0;
            TEMP=0;
        }
        /*Measurement continues*/
    }

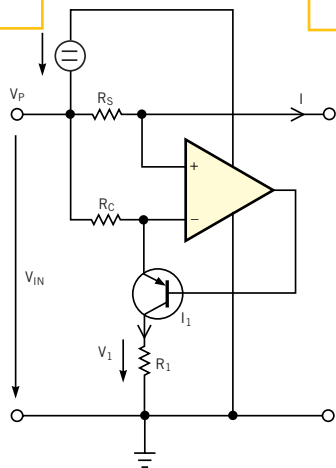
    /*Display the total associated charge due to the input current*/
    QTOTAL=(QCOUNT*Q)/1000.0;
    printf("\n Total Associated Charge during
measurement=%2.1fuC",QTOTAL);
    ETIME= ((QTIMER*54.945)/1000); /*Display the total elapsed time*/
    printf("\n Total Elapsed Time= %7.3f sec ", ETIME);
    RESTOREALL();
    return ;
} /* END OF MAIN */
```

# Simple circuit provides high-side current sensing

M Ossmann and B Kazay, Aachen University of Applied Sciences, Germany

IN POWER SUPPLIES or battery chargers, you often need information about the current flowing in the high-side rail. **Figure 1** shows a common circuit for obtaining this information. The circuit provides a ground-referenced voltage proportional to the current flowing through the high-side sense resistor,  $R_S$ . The circuit needs an additional high-side supply,  $V_P$ . If you use a low-voltage op amp, such as an OP90,  $V_P = 1.5V$  is adequate, so you can derive this supply from a series-transistor voltage drop in many applications. The op amp keeps the voltage drop in  $R_S$  and the conversion resistor,  $R_C$ , equal, so the current through  $R_1$  is  $(R_S/R_C)I$ , and the voltage across  $R_1$  is  $I(R_S/R_C)R_1$ . **Figure 2** shows a circuit that needs no auxiliary transistor. The output stage of the op amp, in a sense, replaces the transistor. The current through  $R_C$  is the same as that in **Figure 1**. Now, this current flows through the negative-supply pin of the op amp and serves as the current source for  $R_1$  as before. The circuit uses a low-power OP90 op amp, which draws a low supply current. The idea is that the supply current contributes a negligible offset to  $I_1$ . Measurements show that the positive-supply current,  $I_p$ , rises with current  $I_1$ . This situation leads to a higher current

**Figure 1**

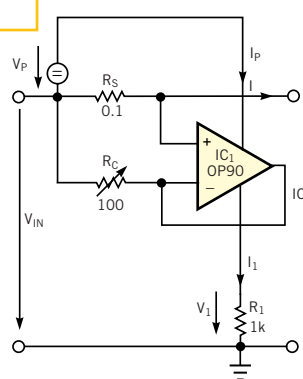


This method for measuring high-side supply current requires an auxiliary transistor.

$I_1$  because of the current  $I_p$ . **Figure 3** shows the result. The dots in **Figure 3** indicate the measured values of  $V_1$ , and the straight line shows the value if  $I_p$  were zero.

Fortunately, the supply,  $I_p$ , is nearly proportional to the op amp's output current. So the value of  $R_C$  in **Figure 2** is such to yield the right value at the maximum current of 2.5A. This simple, one-point adjustment corrects the gain error. **Fig-**

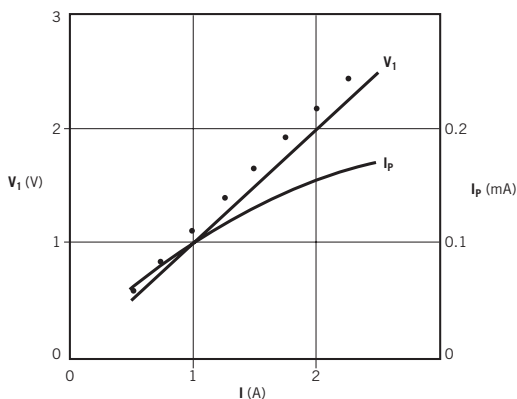
**Figure 2**



This circuit uses the negative-supply current of the op amp to measure the high-side supply current.

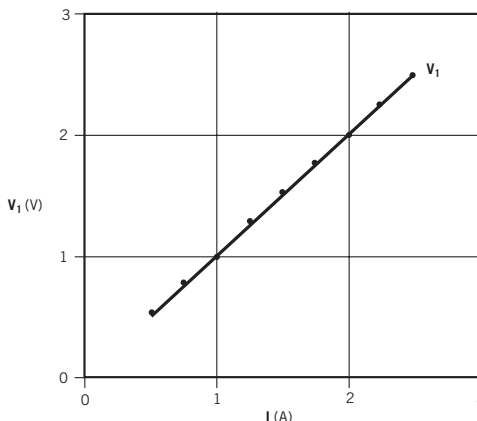
**ure 4** shows the result after this gain compensation. Now, the error between the measured  $V_1$  and the desired value is within 5%, which is tolerable for many applications. Measurements show that the accuracy holds for input voltages of 5 to 25V and an additional supply voltage of 2 to 15V.

Is this the best Design Idea in this issue? Vote at [www.ednmag.com/ednmag/vote.asp](http://www.ednmag.com/ednmag/vote.asp).



**Figure 3**

Because of the positive-supply current, the measured values (dots) do not agree well with the theoretical values (straight line).



**Figure 4**

Gain compensation by adjusting  $R_C$  improves the accuracy of the measurement.