

designideas

READERS SOLVE DESIGN PROBLEMS

PIC microprocessor drives 20-LED dot- or bar-graph display

Noureddine Benabadji,
University of Sciences and Technology, Oran, Algeria

The circuit in **Figure 1** uses only five I/O lines to drive a dot- or bar-graph display of 20 LEDs. Although this version of the design uses a small and inexpensive one-time-programmable microprocessor, such as a Microchip (www.microchip.com) PIC12C508A, you can use other microprocessors with N I/O lines to drive as many as $N \times (N - 1)$ LEDs. For software development or modification, you can use a PIC12C508A-JW reprogrammable-EPROM version of the PIC12C508A, or you can substitute a less expensive PIC16F84A with flash memory.

To avoid application of excessive reverse voltage to the LEDs, the circuit's power supply, V_{DD} , must not exceed 3V dc. You can drive other types

of loads and provide electrically isolated interfaces by replacing the LEDs with appropriately rated optocouplers. For demonstration purposes, IC₁'s input line, GP3, connects to a pushbutton display-mode-selector switch and a pulldown resistor that simulates a digital-input-signal source with a voltage amplitude of 3V p-p.

Listing 1, available with the online version of this Design Idea at www.edn.com/060901di1, performs a variety of functions. To conserve battery power, the basic software drives one LED at a time in dot or bar mode with a minimum amount of current. Approximately 2 mA flashes a high-brightness LED. The software includes a delay routine that solves the problem of contact bounce. Tests show that a

DIs Inside

72 PC's serial port controls programmable sine-wave generator

74 I²C interface connects CompactFlash card to microcontroller

76 IC and DMM form direct-read-out temperature probe

► What are your design problems and solutions? Publish them here and receive \$150! Send your Design Ideas to edndesignideas@reedbusiness.com.

miniature pushbutton switch requires a delay of at least 1 msec for successful debouncing.

Consuming fewer than 256 words, the software avoids a C12C508A programming restriction that requires placement of subroutines only in page 0. Other features of the software include a two-level stack, the use of files common to both the PIC12C508A and

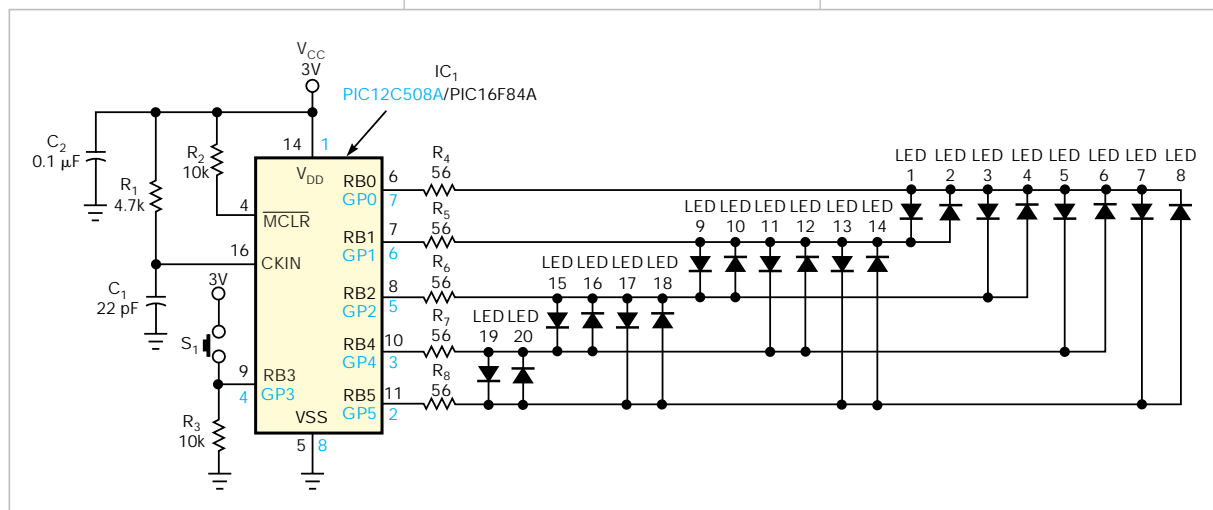


Figure 1 A dot- or bar-graph display uses either a one-time-programmable PIC12C508A or, for experimentation, a reprogrammable and reusable PIC16F84A. Use high-brightness diodes for LED 1 through LED 20.

the PIC16F84A in the $0C_H$ to $1F_H$ range, use of the RETLW 00_H instruction instead of Return, and avoidance of the ADDLW and SUBLW instruc-

tions. The software defaults to the dot-display mode. Pressing and holding S_1 before and during power application selects the bar-display mode. Note that

the configuration settings for the MPASM assembler vary according to which microprocessor you select for programming.^{EDN}

PC's serial port controls programmable sine-wave generator

Yongping Xia, Navcom Technology, Torrance, CA

This Design Idea describes a circuit that uses a PC's serial port to control a sine-wave generator that covers a frequency range of 2 Hz to 20 kHz in 1-Hz steps (Figure 1). The circuit's output voltage of approximately 2.2V p-p remains constant over the entire frequency range. The circuit's signal source, a Linear Technology (www.linear.com) LTC6904, IC₁, consists of a digitally programmable square-wave oscillator that, without using a clock crystal, covers a frequency of approximately 1 kHz to 68 MHz at 0.1% resolution and a few percentage points of accuracy. The LTC6904 features an I²C serial-communications interface that controls the output frequency according to: $OSC_{CLK} = 2a \times$

$2078 \text{ Hz} / [2 - (b/1024)]$. The variables "a" and "b" represent 4- and 10-bit digital codes, respectively, and the equation's frequency unit, OSC_{CLK} , is in hertz.

The OSC_{CLK} output from IC₁ at Pin 6 drives IC₂, a 14-stage 74HC4020 binary counter whose outputs at Q4 and Q10 serve as the clock and input signals for IC₃, a Maxim (www.maxim.com) eighth-order, switched-capacitor MAX291 lowpass filter.

The -3-dB corner of the filter's frequency response occurs at one-one-hundredth of IC₃'s clock frequency. Fixed at one-sixty-fourth of the clock-signal frequency, only the input square wave's fundamental frequency can pass the filter without undergoing consid-

erable attenuation. The filter's eighth-order response removes higher order harmonics, and the filter's output thus consists of a sine wave at the input's fundamental frequency. The filter's clock and input always occur in a 64-to-1 frequency ratio, and the sine wave's output amplitude thus remains constant over the entire frequency range.

To generate a 1-kHz sine wave, the circuit requires a filter clock at a frequency of 64 kHz, which sets the OSC_{CLK} frequency 64 times higher, or 1.024 MHz. To satisfy the equation, the LTC6904 requires programming constants of $a=09_H$ and $b=3d8_H$ to generate an OSC_{CLK} frequency of 1023.94 kHz and the nearest output frequency of 999.9 Hz.

Written for IBM-compatible PCs, the C-language program accompanying this Design Idea, which is available at www.edn.com/060901di2, accepts an

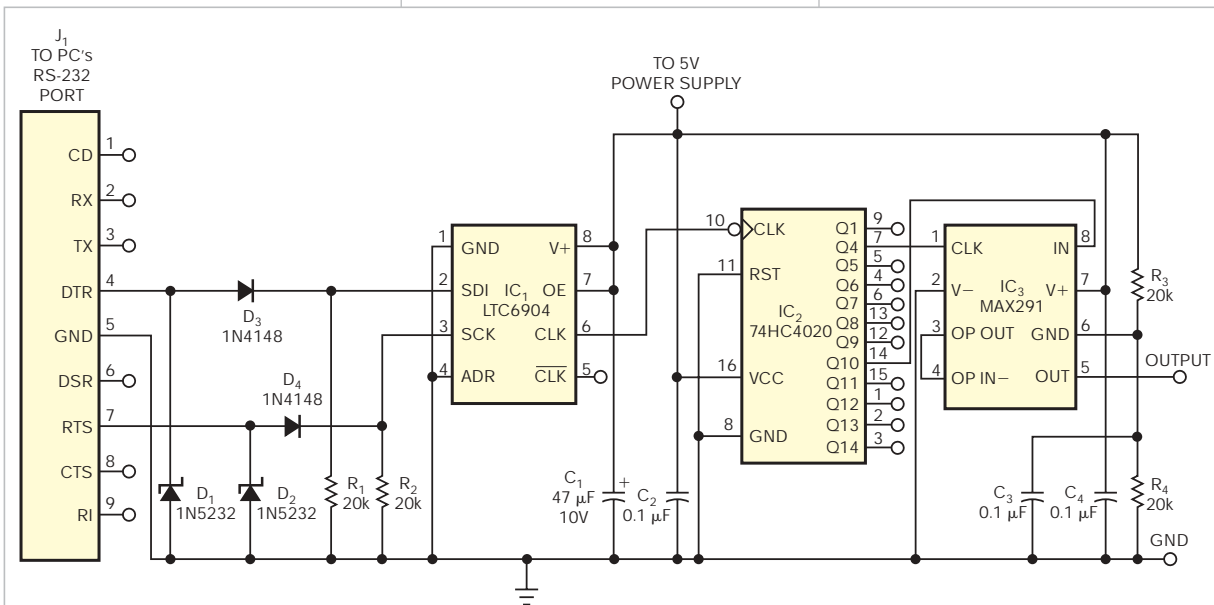


Figure 1 Three ICs and a few passive components generate sine waves under the control of a PC's serial port.

output-frequency request, calculates the nearest values of programming codes “a” and “b,” transmits the codes to IC₁, and shows the calculated fre-

quency on the PC’s display. Although a PC’s serial port delivers RS-232 signals, diodes D₁ through D₄ limit the voltages available at Pin 4, the data-ter-

minal-ready pin, and Pin 7, the ready-to-send pin, to levels compatible with the I²C bus’s SDI and SCK signals, respectively.EDN

I²C interface connects CompactFlash card to microcontroller

Fons Janssen, Maxim Integrated Products Inc, Sunnyvale, CA

Logging data from a large number of monitored channels usually requires a lot of memory for storing the measured data. Unfortunately, smaller microcontrollers offer only limited amounts of internal data RAM and EEPROM and may also lack spare address and data ports for adding external memory. Many low-end microcontrollers include an industry-standard I²C interface for attaching external ADCs, DACs, real-time clocks, and other peripherals.

The circuit in **Figure 1** connects a CompactFlash card to a microcon-

troller’s I²C interface through IC₁, a 16-bit I²C I/O extender. In memory-mapped mode, an 8-bit-wide data bus controls the CompactFlash card. Microcontroller IC₁’s Port 1 (I/O lines 0 through 7) connects to the CompactFlash card’s data bus and provides read and write access to the card’s data registers. Port 2 provides the card’s address and control registers and generates the read and write signals.

To write to a register, configure Port 1 as an output and write the data to the port. Next, write the register-control data three consecutive times to Port 2

while toggling the port’s WR_N pin from Logic 1 to Logic 0 to Logic 1 to generate the “write” signal. Address bits A2, A1, and A0 select the register that receives the written data. Applying Logic 0 to the CE pin while RD_N rests at Logic 1 enables the CompactFlash card. To read from a register, configure Port 1 as an input port and apply three writes to Port 2 while toggling the port’s RD_N pin from Logic 1 to Logic 0 to Logic 1 to generate the “read” signal.

After the three writes, the microcontroller reads Port 1 and makes the data available. Address bits A2, A1, and A0 address eight internal registers and allow read and write access (**Table 1**). Register 0x00 contains data for exchange between the host and the CompactFlash card. Registers 0x03, 0x04, 0x05, and 0x06 specify the track

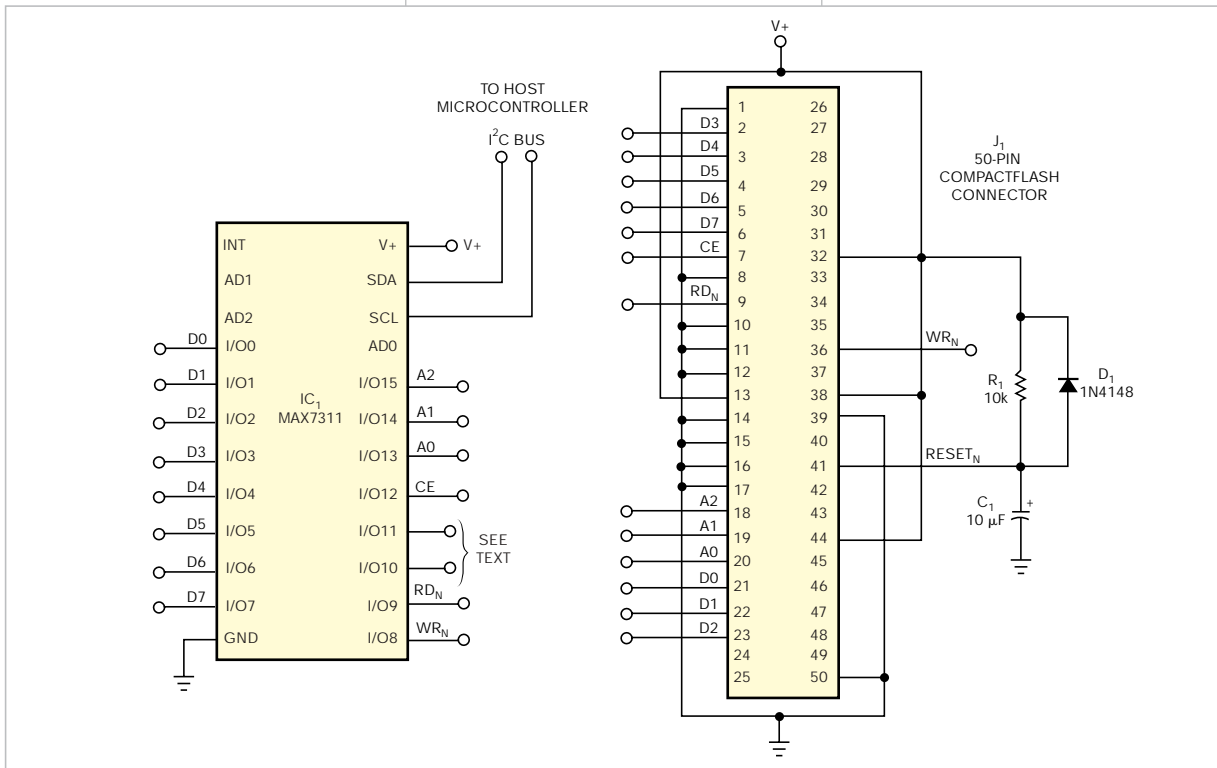


Figure 1 A 16-bit I/O extender, IC₁, and a CompactFlash connector add external data storage to a microcontroller’s I²C bus.

for reading or writing data. Each track contains 512 data bytes. The processor indicates reading and writing tracks and other functions by writing to 0x07, the command register, and registers 0x01 and 0x07 contain error conditions and status information.

Two unused pins, 10 and 11, on Port 2 are available to drive LEDs that display circuit activity and status. As an alternative, the pins can support a user-installed configuration jumper. In this configuration, IC₁'s interrupt output should connect to the host microcontroller's interrupt input so that installation or removal of the jumper can signal the microcontroller to recognize or ignore the CompactFlash card. Selecting a CompactFlash-card connector with hot-plugging contacts allows insertion or removal of a card without switching off power or disturbing an ongoing data-logging process.

With software modifications, a host microcontroller can switch between two CompactFlash cards. Adding a second MAX7311 supports an additional CompactFlash card and expands the circuit's storage capacity, and the hot-plug feature supports removal of a fully loaded card for data processing on another system. Microcontrollers that include hardware-based I²C interfaces can use two relatively simple I²C software functions to read and write a CompactFlash card through IC₁'s I/O ports.

The first function is: Write_MAX3711(slv,prt,dat). This procedure

TABLE 1 ADDRESSES AND REGISTERS

Address	Register	Address	Register
0x00	Data	0x04	Cylinder low
0x01	Error/features	0x05	Cylinder high
0x02	Sector count	0x06	Select card/head
0x03	Sector number	0x07	Status/command

starts the I²C bus and sends a data byte (dat) to a port (prt) on the MAX7311 using a slave address (slv). The other procedure, Read_MAX3711(slv,prt), starts the I²C bus and reads a data byte from a port on the MAX7311 at a slave address. These functions serve as foundations for two additional functions, which read and write to the CompactFlash card's registers. The first, Write_CF_REG(reg,dat), uses Write_MAX3711 to place the data on Port 1. Use the same procedure to place the register address (reg) and other control signals on Port 2. Executing this function three times while toggling WR_N generates the write signal. The Read_CF_REG(reg) procedure uses Write_MAX3711 to address the CompactFlash card's register and generates the read signal. Invoking Read_MAX7311 then reads the data from the register.

These functions, which in turn read and write the card's registers, create functions that access the CompactFlash-card sectors: Write_CF(cyl,head,sec). To perform a write operation, this procedure uses Write_CF_REG to designate the CompactFlash card's target cylinder, head, and sector

registers (0x03 to 0x06). Next, writing 0x30 to the command register configures the CompactFlash card to accept data. Executing Write_CF_REG 512 times writes data in the microcontroller's global array to the data register. The CompactFlash card automatically adds this data to the current track. To perform a read operation, the Read_CF(cyl,head,sec) procedure uses Write_CF_REG to designate the target cylinder, head, and sector. Next, writing 0x20 to the command register configures the CompactFlash card to deliver data to the host processor. Executing Read_CF_REG 512 times reads all 512 bytes through the data register from the current CompactFlash card's track and places the data in a global array.


If the microcontroller lacks sufficient internal memory to store 512 data bytes, the software can write each digitized data-point measurement directly to the CompactFlash card. For additional information on controlling CompactFlash cards, review the material in **Reference 1**.^{EDN}

REFERENCE

1 CF+ and CompactFlash specification, www.compactflash.org.

IC and DMM form direct-read-out temperature probe

Alfredo H Saab and Bich Pham,
Maxim Integrated Products Inc, Sunnyvale CA

 The simple temperature-measurement probe in **Figure 1** can serve as an indispensable tool for troubleshooting and debugging electronic circuits. To measure temperature at several points, you can equip IC₁, a Maxim (www.maxim-ic.com) MAX6610, with a probe, or you can perma-

nently integrate one or more devices into a pc board or attach them to components. Resistors R₁, R₂, and R₃ set the circuit's temperature-scaled voltage output to various values (**Table 1**). **Figure 2** shows the circuit's representative output versus temperature.

You can display the circuit's temper-

ature-proportional dc output voltage on any DVM (digital voltmeter) or handheld DMM (digital multimeter). The circuit draws only 200 μA from a nominal 3V power supply, such as a pair of AA alkaline cells. A CR2016 lithium-coin cell can operate the circuit continuously for several hundred hours or for several years if you equip the circuit with a normally open, momentary-contact pushbutton switch.

To produce the error curve in **Figure 3**, immerse the circuit and a platinum-resistance standard thermometer in a

temperature-controlled oil bath. The circuit's relative error with respect to the standard thermometer varies only 4°C over -40 to +125°C. The MAX6610's data sheet includes additional information on temperature-measurement error and output range.

To apply the circuit as a temperature probe, solder a 5-mm length of 1-mm-diameter, uninsulated copper wire directly to a small copper pad at IC₁'s GND pin. The wire should make thermal and electrical contact with the GND pin and thus provide a path of low thermal resistance from the sensor IC to the point of probing. Glue the wire to the pc board to add mechanical support. Heat loss affects the temperature measurement's accuracy, and, to minimize heat loss from the probe through the pc board, use long and thin copper traces to make electrical connections from IC₁ to its supporting components.

Applying the MAX6610 as a pc-board temperature sensor differs somewhat from using it as a temperature probe. For board-temperature sensing, IC₁ must reside in intimate thermal contact with the board. Connect large copper areas immediately to the IC's pins and use short, thick traces—or none at all—between the copper areas and the IC's pins. The copper areas guarantee accurate temperature readings by providing thermal contact with

the board and good heat transfer between the board and the sensor.^{EDN}

REFERENCE

1 "Precision, Low-Power, 6-pin SOT23 Temperature Sensors and

Voltage References," MAX6610/6611 data sheet, Maxim Integrated Products, November 2003, <http://pdfserv.maxim-ic.com/en/ds/MAX6610-MAX6611.pdf>.

TABLE 1 TEMPERATURE-SCALED VOLTAGE OUTPUT

	10 mV/°C	1 mV/°C	1 mV/°F
R ₁ (kΩ)	68.1	68.1	68.1
R ₂ (kΩ)	2.8	2.8	19.6
R ₃ (kΩ)	Open	2.21	3.32

Note: All resistors are of ±10% tolerance.

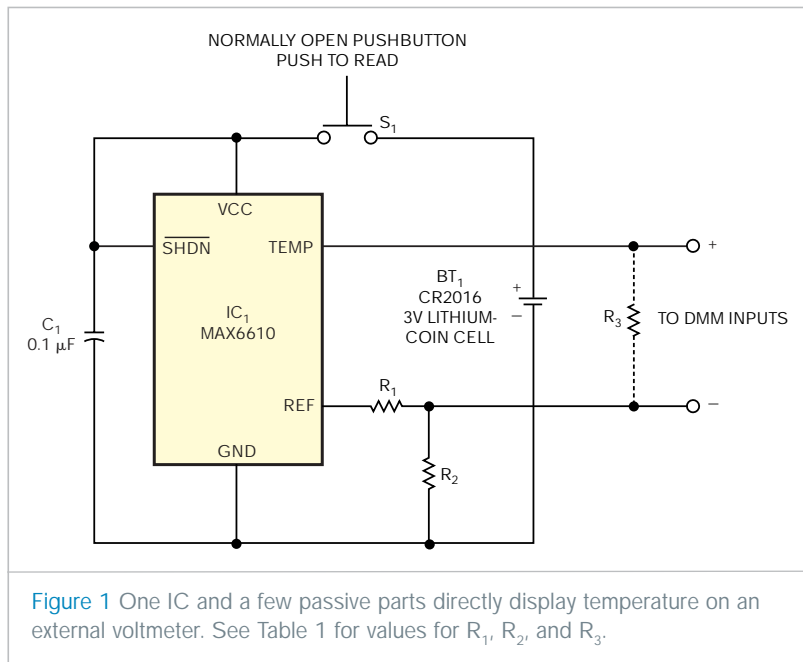


Figure 1 One IC and a few passive parts directly display temperature on an external voltmeter. See Table 1 for values for R₁, R₂, and R₃.

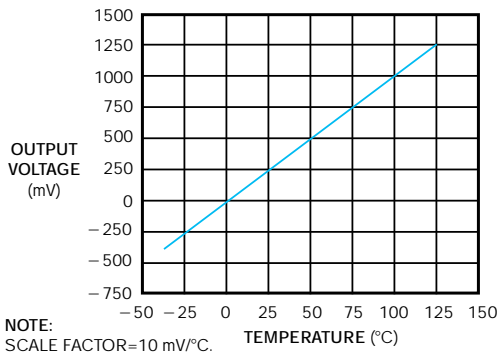


Figure 2 The circuit of Figure 1 exhibits a nearly linear output-voltage-versus-temperature characteristic.

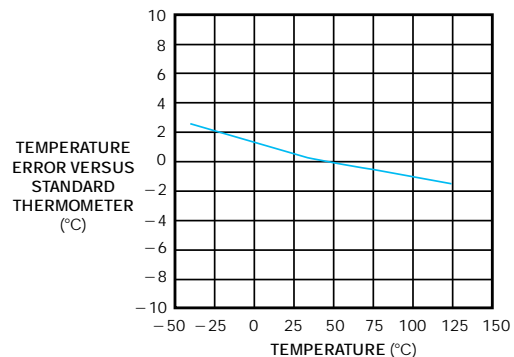


Figure 3 Immersed in a temperature-controlled oil bath and compared with a platinum-resistance standard thermometer, the circuit of Figure 1 exhibits ±2°C error over -40 to +125°C.