the jitter/wander frequency. The relationship between phase deviation and jitter/wander amplitude is straightforward, and you can obtain it from:

$$\text{JITTER/WANDER}[\text{UI p-p}] = \frac{D_{PM}}{180°}, \quad (3)$$

where D_{PM} has units of radians in communications theory, but, for convenience, most signal generators specify units of degrees instead.

For FM, the phase θ(t) in **Equation 1** is proportional to the integral of the modulating signal:

$$\theta(t) = 2\pi D_{FM} \int \cos(2\pi f_m t)dt, \quad (4)$$

where D_{FM} is the frequency deviation (peak variation of the frequency) and f_m is the modulating frequency. The FM modulating frequency is the same as the jitter/wander frequency. **Equation 5** yields the jitter/wander amplitude:

$$\text{JITTER/WANDER}[\text{UI p-p}] = \frac{D_{FM}}{\pi f_m}, \quad (5)$$

which derives from **Equation 6**:

$$\theta(t) = 2\pi D_{FM} \int \cos(2\pi f_m t)dt = 2\pi D_{FM} \frac{1}{2\pi f_m} \cos\left(2\pi f_m t - \frac{\pi}{2}\right). \quad (6)$$

Therefore, the peak-to-peak deviation of θ(t) is:

$$\theta_{PP} = 2\frac{2\pi D_{FM}}{2\pi f_m} = \frac{2D_{FM}}{f_m}, \quad (7)$$

in which the factor of 2 originates from the peak-to-peak amplitude of a sine-wave function. To get jitter/wander in peak-to-peak unit intervals, divide **Equation 7** by the period of the sine-wave function, 2π. Thus, you get **Equation 6**, which is valid only when the modulating signal comprises a sine wave because the integral of a sine wave is also a sine-wave function shifted in phase. Fortunately, most jitter/wander-tolerance tests almost exclusively use sine-wave modulation.

Some signal generators specify modulation in phase and frequency span instead of phase and frequency deviation. For PM, the span is the peak-to-peak variation of the phase, and, for FM, the span is the peak-to-peak variation of the frequency. That is, the span equals twice the deviation for both PM and FM. In this case, the jitter/wander amplitude for PM is:

$$\text{JITTER/WANDER}[\text{UI p-p}] = \frac{\text{SPAN}_{PM}}{360°}, \quad (8)$$

and it is:

$$\text{JITTER/WANDER}[\text{UI p-p}] = \frac{\text{SPAN}_{FM}}{2\pi f_m}. \quad (9)$$

for FM.□

Temperature controller saves energy

Tito Smailagich, ENIC, Belgrade, Yugoslavia

Given the high cost of electrical power, replacing a conventional on/off temperature control with a proportional controller can often save energy and money. **Figure 1** shows a low-cost, high-efficiency, time-proportional temperature controller for a residential water heater. An Analog Devices ADT14, IC_1, serves multiple functions as a temperature sensor, quad-setpoint, programmable analog temperature monitor and controller. Resistors R_1, R_2, R_3, R_4, and R_5 adjust the desired temperature at setpoints SETP1, SETP2, SETP3, and SETP4, which IC_1 compares with the actual temperature from its internal sensor. The ADT14's active-low open-collector outputs drive Input Port A of IC_2, an 8-bit Motorola/Freescale 68HC908QT4 microcontroller that provides 4 kbytes of flash memory, 128 bytes of RAM, and an on-chip clock oscillator.

Available at *EDN*'s online version of this Design Idea at www.edn.com,

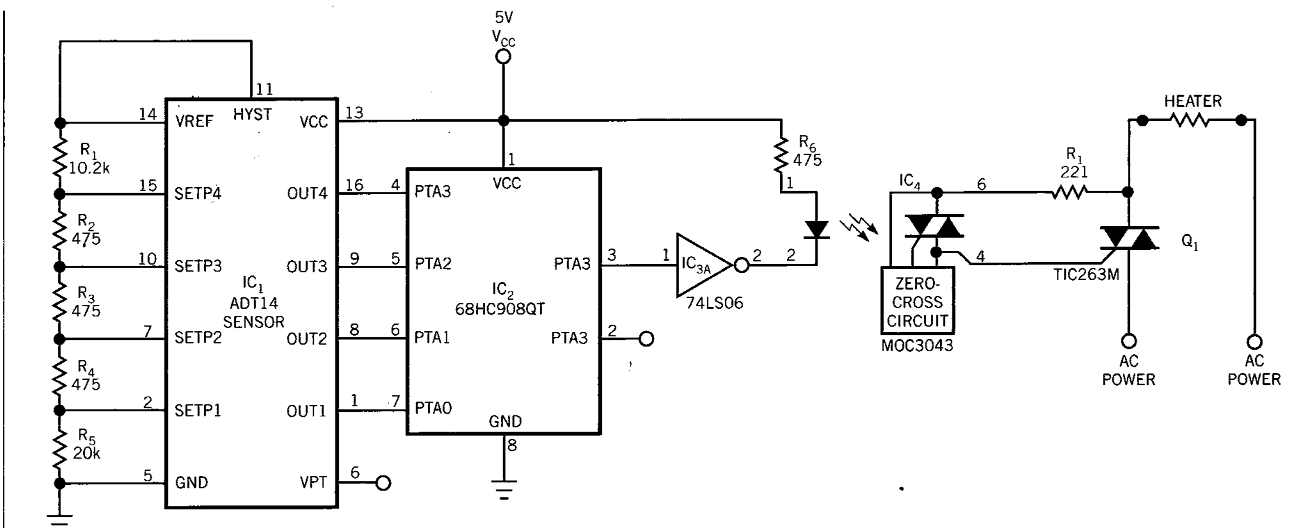

Figure 1 This proportional temperature controller features a minimal parts count.

Listing 1 contains commented assembly-language software. When you load it into the microcontroller's flash memory, the software provides the time-proportional control algorithm. When IC_1's OUT1, OUT2, OUT3, and OUT4 outputs are inactive, IC_2 switches its output PTA4 to a totally on-state, 100% duty cycle for maximum heating. **Listing 2** at the Web version of this Design Idea contains an assembled version of the software, and **Listing 3** presents the hex code for programming IC_2.

When IC_1's OUT1 output is active, IC_2 produces a 75%-duty-cycle output on PTA_4. In similar fashion, when IC_1's OUT2 output goes active, IC_2 produces a 50%-duty-cycle output on PTA4, and when IC_1's OUT3 output goes active, IC_2 produces a 25%-duty-cycle output on PTA_4. When IC_1's OUT4 output goes active, IC_1 disables the output on PTA4 to produce a totally-off state (0% duty cycle). **Table 1** summarizes the relationship of IC_2's inputs and output duty cycle.

To minimize component count, IC_2's internal oscillator generates a 12.8-MHz clock that divides to produce a sample pulse whose basic width is 0.1 sec for each 1% of output on-time. One cycle of output comprises 100 samples for a total duration of 10 sec. Thus, for a 25% duty cycle, IC_2's output PTA4 generates a 2.5-sec on interval followed by a 7.5-sec off interval. One section of an open-collector hex inverter, IC_{3A}, a 74LS06, drives optocoupler IC_4, an MOC3043, which features an internal zero-crossing circuit and pilot triac. Power triac Q_1, a TIC263M rated for 600V and 25A, controls application of power to the water heater's 2-kW resistive heating element. For best results, place IC_1 in close thermal contact with the water heater's inner tank.□

TABLE 1—IC_2'S LOGIC STATES VERSUS OUTPUT DURATIONS

PTA3	PTA2	PTA1	On (%)	Off (%)	On (sec)	Off (sec)
1	1	1	100	0	10	0
1	1	1	75	25	7.5	2.5
1	1	0	50	50	5	5
1	0	0	25	75	2.5	7.5
0	0	0	0	100	100	10

Calculator program finds closest standard-resistor values

Francesc Casanellas, Aiguafreda, Spain

ALTHOUGH IT MAY not appear obvious to newcomers to the electronics-design profession, components' values follow one of several progressions that divide a decadewide span into equally spaced increments on a logarithmic scale. For example, when you plot the values of 1, 2.2, and 4.7 on a logarithmic scale, they divide the range 1 to 10 into three roughly equal increments (1... 2... 5). To meet requirements for greater precision, resistor manufacturers offer parts in several additional series. The most precise series divide a decade into 24, 48, or 96 increments by computing $10^{n/m}$, where n=1... (m−1), and m=24, 48, or 96, and then rounding the values to two or three digits. The results are the R_{24}, R_{48}, and R_{96} series and respectively contain 24, 48, or 96 values per decade.

You can use a Hewlett-Packard HP-48 or HP-49 calculator and one of the following programs written in RPN (Reverse-Polish Notation) to compute the nearest standard value that's closest to a required value. You enter a required resistor value, and the program returns the closest higher or lower value in the selected series. **Table 1** lists a few examples.

Each program acts as an operator by processing the first line of the calculator's stack and returning the new value in the same line of the stack. The R_{48} and R_{96} series are mathematically exact, and their programs consist of only a single line of code. The **Listings** that you find at the Web version of this Design Idea at www.edn.com show the code. The values of the older R_{24} series are not as strictly rounded, and the program is thus somewhat more complex.

Note that the values of other components, such as capacitors, inductors, and zener diodes, also follow preferred-value series, making these programs universally applicable. You can view an earlier version of a standard-value calculator for IBM-compatible PCs at *EDN*'s online version of Design Ideas. David Kirkby of the Department of Medical Physics, University College London, UK, wrote the program in C. *EDN* first presented it in its Aug 3, 1995 issue ("Resistance calculator yields precise values)". You can read the instructions at www.edn.com/archives/1995/080395/16di5.htm. Note that certain portions of the software may require rewriting for better operation on today's PCs.□

TABLE 1– [illegible] SELECTED SERIES [illegible] VALUE OF:

47.8	R_{24}	47
[illegible]	R_{24}	510
12.2	R_{96}	12.1
12.3	R_{96}	12.4