ing 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.

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

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.□

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.

TABLE 1–INPUT VALUE IN SELECTED SERIES RETURNS A CLOSEST VALUE OF:

47.8	R_{24}	47
490	R_{24}	510
12.2	R_{96}	12.1
12.3	R_{96}	12.4

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** 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, "Resistance calculator yields precise values," in the Aug 3, 1995, issue. 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.□

EDN050203DI3580—LISTINGS 1, 2, and 3

"Calculator program finds closest standard-resistor values," *EDN*, Feb 3, 2005, pg 86.

Listing 1: Convert a resistor value into the nearest value of the R24 series.

```
« ?  r
« r MANT ?  m
« {1. 1.1 1.2 1.3 1.5 1.6 1.8 2. 2.2 2.4 2.7 3. 3.3 3.6 3.9
4.3 4.7 5.1 5.6 6.2 6.8 7.5 8.2 9.1 10. } 'L1' ST0
0
DO
 1 + DUP 'L1' SWAP GET
UNTIL m >
END
DUP 'L1' SWAP GET m / ?  b
« 1 – 'L1' SWAP GET m / INV ?  a
« a b
IF =
 THEN r b *
 ELSE r a /
END
-2 RND
» » » »
'L1' PURGE
»
***
```

Listing 2: Convert a resistor value into the nearest value of the R48 series

```
« '10^(1/48)' ?  r n 'n^IP(LN(x*vn/LN(n))' EVAL –3 RND »
***
```

Listing 3: Convert a resistor value into the nearest value of the R96 series.

```
« '10^(1/96)' ?  r n 'n^IP(LN(x*vn/LN(n))' EVAL –3 RND »
***
```