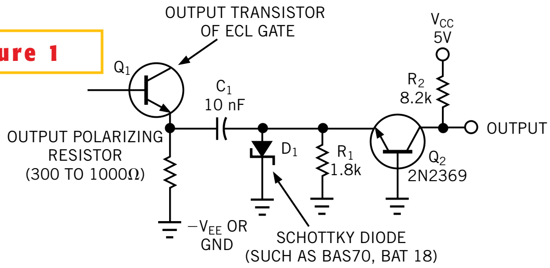
Edited by Bill Travis

Trigger a TTL circuit from ECL levels

Lukasz Sliwczynski, University of Mining and Metallurgy, Krakow, Poland

ECL circuits typically have relatively small logic spans of approximately 800 mV. Because of the small span, to drive TTL circuits from ECL levels normally entails the use of level converters, such as the MC10125, or comparators. Such circuits are relatively power-hungry and expensive. However, they are sometimes simply unnecessary. The circuit in **Figure 1** allows you to trigger some TTL circuitry by generating a fairly short negative-going pulse from the trailing edge of the ECL signal. The main requirement for the circuit to work is that the rate of ECL signal be in the tens of kilohertz. Such signals sometimes appear at the rear panels of some older types of measurement equipment. Such equipment can include sampling oscilloscopes or time-domain reflectometers, such as the 7S12 or 7S14 from Tektronix. In a measurement setup, the circuit in **Figure 1** exploits the sampling gate from a 7S12 plug-in unit.

Figure 1



You don't need an expensive level-converter IC to provide a TTL-level trigger from an ECL-level signal.

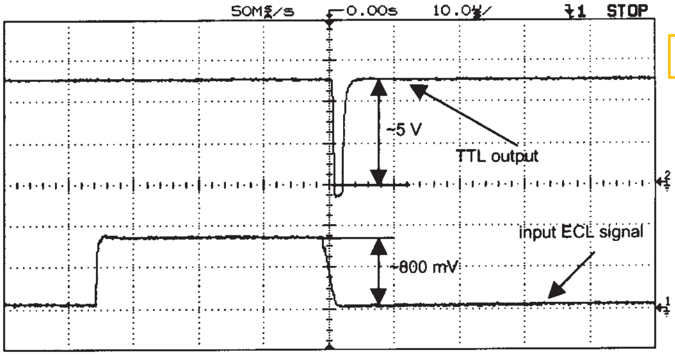
Figure 2 shows the waveforms associated with the circuit in **Figure 1**. The positive portion of the ECL signal charges capacitor C_1 through the Schottky diode, D_1. In this part of the operating cycle, transistor Q_2 is off, and the output voltage is approximately 5V. On the negative-going edge of the driving pulse, the charge from coupling capacitor C_1 causes the base-emitter junction of Q_2 to conduct, driving the transistor into saturation. The output voltage assumes a level slightly below 0V. The duration of the generated negative-going pulse depends on the speed with which C_2 discharges. The discharge takes place through the base-emitter junctions of Q_1 and Q_2 and resistor R_1. The duration is difficult to calculate, but for a rough estimate, you can use the following equation:

$$t_P = R_1C_1 \ln\frac{\Delta V - V_{DS}}{V_{BE}} \approx 0.08\, R_1C_1,$$

where $\Delta V \approx 0.8V$ is the ECL span, $V_{DS} \approx 0.15V$ is the voltage drop of the Schottky diode, and $V_{BE} \approx 0.6V$ is the voltage drop of the base-emitter junctions. In practice, the durations are shorter than predicted because the equation does not take account of the base-emitter resistances of Q_1 and Q_2. For the components in **Figure 1**, the duration is approximately 2 μsec. The crucial component in the circuit is D_1, which must be a Schottky type, because of the voltage swing of the ECL signal, which is nearly the same as the base-emitter voltage of the conducting silicon transistor. Proper operation of the circuit occurs because of the voltage difference between Schottky and silicon-junction levels, which is typically 0.1 to 0.3V. This difference allows for the strong saturation of Q_2 just after the trailing edge of the ECL signal.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Figure 2



The circuit in Figure 1 provides a 2-μsec, 5V negative-going trigger from an 800-mV ECL signal.

Microcontroller discerns addresses in RS-485 systems

Nigel Brooke and Ted Salazar, Maxim Integrated Products, Sunnyvale, CA

One of the many benefits of using the RS-485 data-interface system, unlike the RS-232 system, is its ability to implement multidrop networks. Such networks usually carry 9-bit data words, in which the ninth (parity) bit identifies each word as address or data. When using small microcontrollers without a hardware UART, such as IC$_1$ in **Figure 1**, designers must decide whether to add an external hardware UART or to configure a UART in software. External UARTs once represented a large increase in board area, complexity, and cost, and the available UARTs were usually overkill for simple microcontroller applications. On the other hand, sparing the program memory and processor resources you need for a software-based UART can sometimes be difficult. The program memory in IC$_1$, for example, has only 1k×14 bits of EEPROM. You have a third alternative—a small, low-cost external UART, IC$_2$. The use of this device liberates the program memory you otherwise need for a software-based UART.

An RS-485 bus can carry as many as 256 transceiver modules of the type in **Figure 1**. IC$_3$ is the RS-485 transceiver, and IC$_4$ is a "microcontroller supervisor" that holds the microcontroller in a reset state until a valid supply voltage is present. You can download the assembly-language program for the microcontroller from the Web version of this Design Idea at *EDN*'s Web site, www.ednmag.com. The application in **Figure 1** is a slave-test configuration, but you can modify the code to accommodate any specific RS-485 address-recognition application. The circuit works as follows: When the bus transmits an address, IC$_2$ in each slave module initiates a parity interrupt. IC$_1$ in each module then reads all the data in its internal FIFO, locates the address word, and compares that address with its own address stored in the eight DIP switches. A match causes the slave to clear the interrupt and transmit (to the master) an ASCII "A" (41_h), followed by its own address. If the slave module reads the FIFO's contents without finding a match, it clears the current address-word interrupt and waits for the next one.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

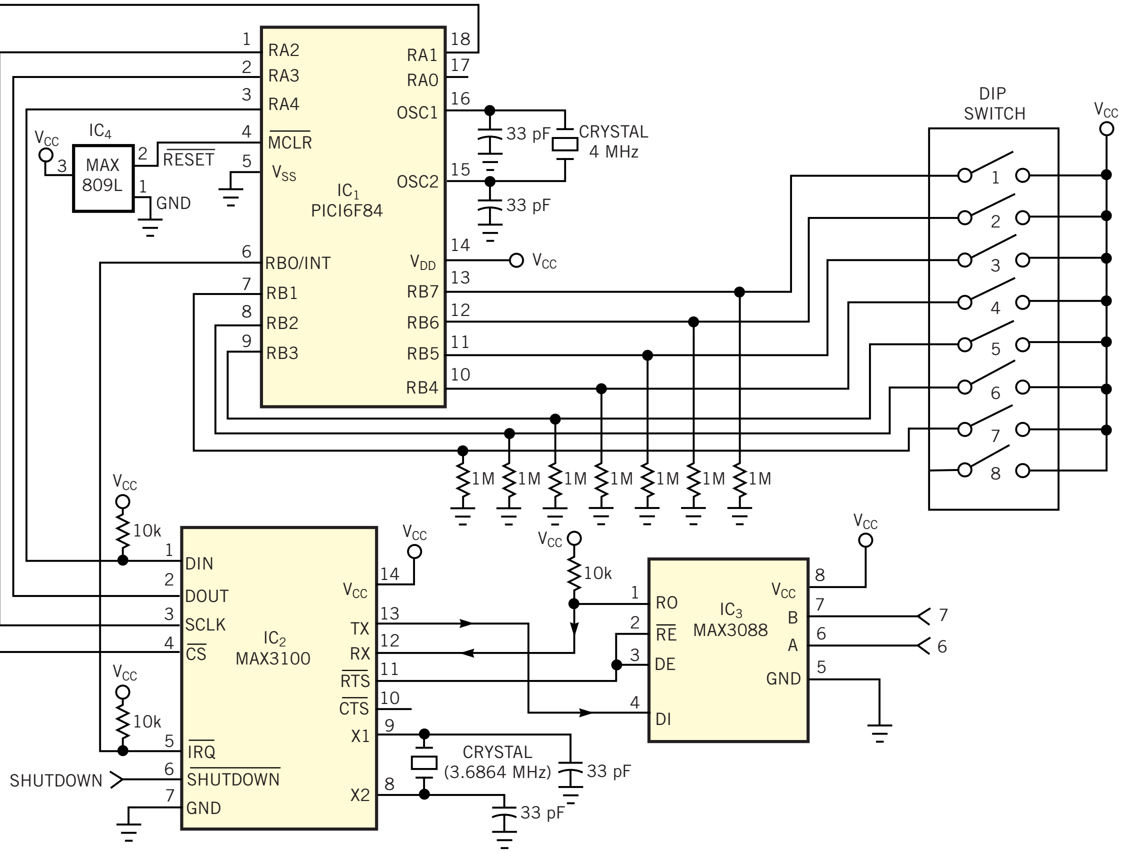


Figure 1

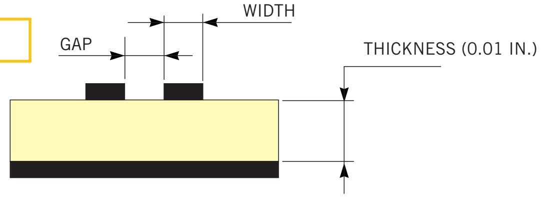
Adding a small UART, IC$_2$, and microcontroller, IC$_1$, to the RS-485 transceiver, IC$_3$, forms a slave data-transceiver module that responds to its own network address.

PC-board layout eases high-speed transmission

Gregory Adams, Moorestown Microwave Co, Moorestown, NJ

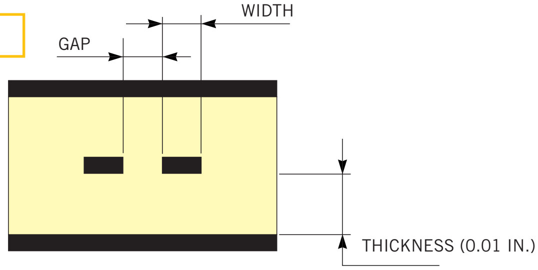
As digital techniques move to higher speeds, designers become aware of the need to treat pc-board traces as RF transmission lines. In these lines, you strive to hold the line impedance, Z_0, to a constant value—typically, 50Ω—and to terminate the line with the same impedance. Data families such as ECL, PECL, and LVDS send data over a pair of traces known as a balanced transmission line. One line switches high, while the other switches low. As with other high-speed logic families, you must hold the transmission-line impedance constant and properly terminate the line. If the spacing between the pair of traces is large, then you can design the traces as simply two 50Ω transmission lines. On the other hand, if the spacing between the traces is less than several times the board thickness, then the effect of one trace on the other changes the characteristic impedance of the line.

Figure 1



A microstrip transmission line has traces on one side of a pc board and a ground plane on the other side.

Figure 2



A stripline transmission line has two traces embedded in the pc board and ground planes above and below.

In RF parlance, when equal voltages drive the two lines, the resulting impedance of each individual line to ground is called Z_0 Even, or Z_{0e}. When equal and opposite voltages, as with differential signaling, drive the two lines, the impedance of one line to ground is called Z_0 Odd, or Z_{0o}. You need to concern yourself only with Z_{0o}, because it applies the to the impedance of a differential-data transmission line. The Z_{0o} of a differential pair is always lower than the Z_0 value of a single trace having the same width on the same board. To hold the impedance of a transmission line to some required value, you must make the traces narrower than would be the case with a single trace. Generally, this fact is good news for digital designers who need to make those transmission lines fit between the vias under a dense BGA chip.

If the traces are on the top of a board with a ground plane under them, then you can model them as coupled "microstrip" lines (**Figure 1**). On the other hand, if the traces are in a layer with ground planes above and below them, then you can model them as coupled "striplines" (**Figure 2**). In the stripline case, you assume that the pair of transmission lines is sandwiched between the two ground planes and that the board thicknesses to the top and ground planes are equal. **Tables 1** and **2** show the line width required to hold Z_{0o} constant at 50Ω for various values of the gap between the two traces. **Table 1** applies to the microstrip case with lines on top of the board; **Table 2** applies to the stripline case with lines sandwiched between equally spaced ground planes. Note that the trace widths are much smaller in the stripline case because of the second ground plane. Both tables assume a board thickness of 0.01 in. You can directly scale the line widths and gaps for other dielectric thicknesses. In every case, the dielectric material is FR-4 with a dielectric constant of 4.6. The tables use the old DOS version of HP's Appcad, a program HP distributes as freeware. The newer versions of this program do not handle coupled lines. To calculate the impedance, Z_0 Odd, of differential transmission lines of other dimensions, you can download a copy of Appcad from www.geocities.com/gregs downloadpage.

TABLE 1—DIMENSIONS FOR 50Ω MICROSTRIP

Gap (in.)	Width (in.)
0.005	0.011
0.01	0.0142
0.015	0.0158
0.02	0.0166
0.025	0.0171
∞	0.0185

TABLE 2—DIMENSIONS FOR 50Ω STRIPLINE

Gap (in.)	Width (in.)
0.005	0.0055
0.01	0.0075
0.015	0.0083
0.02	0.0086
0.025	0.0087
∞	0.0088

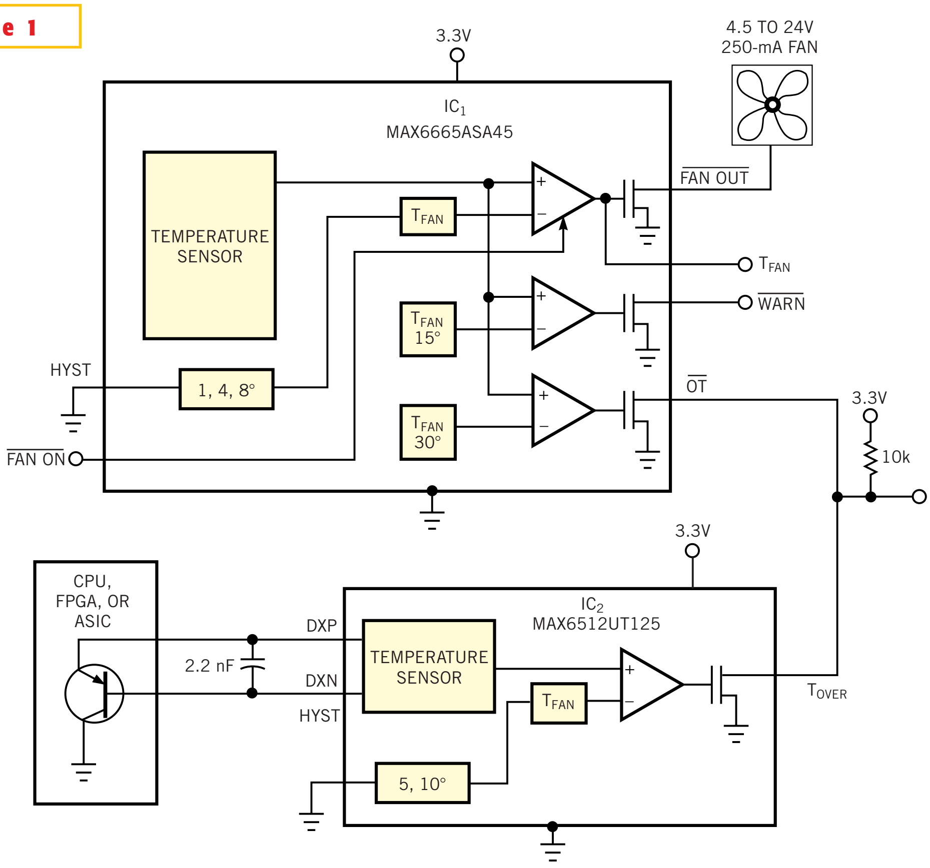
Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Circuit protects system from overheating

Kerry Lacanette, Maxim Integrated Products, Sunnyvale, CA

The two-chip circuit in **Figure 1** provides fan control and overtemperature warning and shutdown signals to protect systems from excessive heat. The circuit monitors the temperature of the pc board and the die temperature of a CPU, an FPGA, or another IC with an on-chip temperature-sensing transistor. IC$_1$ is a temperature detector and fan driver for cooling fans with nominal operation of 250 mA. At low temperatures, the cooling fan is off, minimizing noise and fan wear. When the system temperature increases to more than 45°C, IC$_1$'s factory-programmed temperature comparator causes the FAN OUT fan-drive pin to go active, pulling the fan's lower power-supply terminal to ground, thus providing low-side drive to the fan. The fan can accommodate supply voltages as high as 24V. After the fan activates, the system temperature normally either continues to rise at a slower rate or drops somewhat. If the temperature drops far enough, the fan turns off. To avoid causing the fan to continuously turn on and off, IC$_1$ provides hysteresis of 1, 4, or 8°C, which you can set by the HYST pin.

Figure 1



This circuit provides fan control and overtemperature protection for systems and high-power digital ICs.

If a thermal problem, such as excessive power dissipation or blocked ventilation paths, exists, system temperature may continue to increase. IC$_1$ has two outputs that detect this condition. $\overline{\text{WARN}}$ becomes active when the temperature exceeds 60°C, and the $\overline{\text{OT}}$ output becomes active when the temperature exceeds 75°C. You can use $\overline{\text{OT}}$ as a system-shutdown signal. While IC$_1$ monitors board temperature, IC$_2$ monitors the die temperature of another chip—typically, a CPU, an FPGA, or an ASIC. The target IC must have a small-signal p-n junction, usually a substrate pnp, for temperature measurement. IC$_2$ forces current through sense junction, measures the resulting voltage, and calculates the temperature of the junction. IC$_2$ then compares this temperature with a preset threshold. When the junction temperature exceeds the threshold, 125°C in this case, IC$_2$'s output pin goes active; you can use it to shut down the system.

The open-drain shutdown outputs of IC$_1$ and IC$_2$ connect to a common pullup resistor and to the power supply's shutdown terminal. If either the board temperature or the chip temperature exceeds the maximum safe rating, the system shuts down before damage can occur. IC$_1$ should be in a location that allows it to measure the temperature of interest. Depending on the system, this location could be near a "hot spot" or in the cooling fan's airflow path. The traces between IC$_2$ and the remote-sensing junction should be reasonably short and separated from high-speed data traces.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Network imitates thermocouples

Abel Raynus, Armatron International, Melrose, MA

Thermocouples find widespread use for temperature measurement in systems. During system design or testing, you must observe the system's response at different temperatures. However, it's inconvenient to heat a thermocouple every time you need to check a system's performance. You can use the simple trick of touching the thermocouple with a hot soldering iron, but this method provides only rough, approximate results. The simple network in **Figure 1** allows you to set a number of voltages equal to the thermocouples' outputs at given temperatures. A thermocouple's output is relatively in the tens of millivolts. The low level entails the use of a high-gain amplifier as a signal conditioner. These high-gain amplifiers are sensitive to noise. Susceptibility to noise is not a problem when the amplifier connects to a thermocouple, thanks to the thermocouple's output impedance of approximately 1Ω. But during system testing, substituting a high-impedance source for the thermocouple can result in noise pickup that can drive the amplifier into saturation. Hence, the output impedance of the thermocouple imitator must be low, and the output must connect to ground between tests.

TABLE 1—CORRECTION FOR COLD-JUNCTION TEMPERATURE

Temperature (°F)	Voltage at 32°F (mV)	Voltage at 100°F (mV)
550	11.71	10.19
855	18.82	17.3
900	19.89	18.37
1070	23.91	22.39

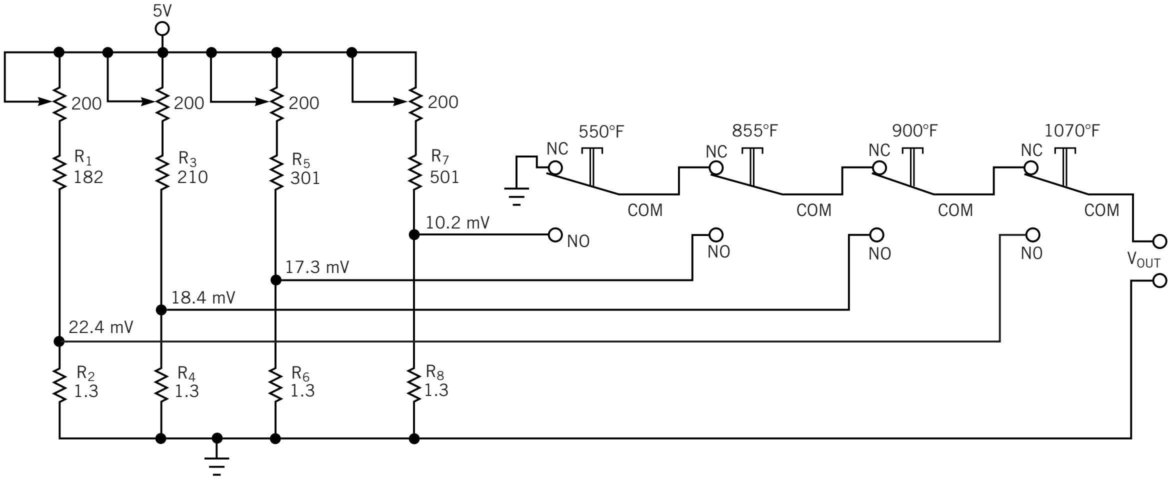
Figure 1 shows the thermocouple imitator for four temperatures. To obtain low output impedance, you set R_2, R_4, R_6, and R_8 to 1.3Ω. To satisfy the between-tests grounding requirement, the momentary SPDT key switches connect to the chain in a way that, when you press no switch, the output connects to ground. By pressing a switch, you obtain one of the predetermined voltages from dividers R_1/R_2, R_3/R_4, R_5/R_6, or R_7/R_8 at the output. Assume, for example, imitator-equivalent temperatures of 550, 855, 900, and 1070°F. You can find the voltages from a Chromel-Alumel thermocouple (**Reference 1**). But keep in mind that the voltages in the book apply only to a cold-junction temperature of 32°F. The working temperatures are always different, so you must recalculate the voltages. Assuming that the ambient temperature is approximately 100°F, you can find the thermocouples' output voltages by subtracting 1.52 mV from the 32°F value (**Table 1**). You can calculate the values of the divider resistors using the following equation: $R_U=R_L(V_{CC}/V_{OUT}-1)$, where R_U is the upper divider resistor, R_L is the lower divider resistor, V_{CC} is the power-supply voltage, and V_{OUT} is the output voltage. To make the output-voltage adjustment easier, the upper divider resistor consists of a 200Ω potentiometer in series with a fixed resistor.

References

1. *The Temperature Handbook*, Omega Engineering Inc, 2000.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.

Figure 1

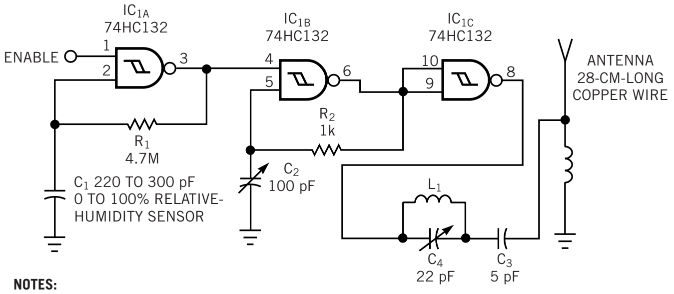


This network allows you to emulate thermocouple outputs at various temperatures.

Low-cost relative-humidity transmitter uses single logic IC

Shyam Tiwari, Sensors Technology Ltd, Gwalior, India

Figure 1



NOTES:
L_1 IS SIX TURNS OF 22-GAUGE WIRE WITH 5-MM DIAMETER.
L_2 IS 18 TURNS OF 22-GAUGE WIRE WITH 5-MM INNER DIAMETER.

This percentage-relative-humidity transmitter uses 10- to 50-MHz, tunable RF, and 1- to 2-kHz on/off amplitude modulation.

The low-cost percentage-relative-humidity radio transmitter in **Figure 1** operates in a cold-storage warehouse for vegetable storage at temperatures of 1 to 5°C. It is generally difficult to collect such data from a low-temperature area with high humidity and low illumination. The transmitter design is simple: It uses a readily available, capacitor-type percentage-relative-humidity sensor for which the capacitor value increases with humidity. Generally, these sensors offer accuracies well within 5%. Humirel (www.humirel.com) relative-humidity sensors work well with this circuit; you can also use other types with low leakage resistance. The R_1C_1 product gives the time constant for the audible-modulating, 1- to 2-kHz signal oscillator, which you can gate to stop the communication. This oscillator starts the RF oscillator, which has a time constant, R_2C_2, equating to a 10- to 50-MHz RF band. The last inverter is a power driver for the tuned filter and antenna. The circuit requires a 3 to 5V battery. Two AAA cells can power it for approximately 15 days. If you need a high modulating frequency, then you can reduce R_1 to 1 MΩ, changing the modulating signal to the range of 10 to 20 kHz.

Is this the best Design Idea in this issue? Vote at www.ednmag.com.