

designideas

READERS SOLVE DESIGN PROBLEMS

Fast 10-line-to-one-line data selector/multiplexer comprises only two ICs

Marián Štofka, Slovak University of Technology, Bratislava, Slovakia

When dealing with logic operations over BCD (binary-coded-decimal) numbers, you often need a 10-line-to-one-line data selector/multiplexer. In the past, you could use the famous 16-line-to-one-line 74150 multiplexer IC. Nowadays, however, when you look at the Web sites of the renowned semiconductor houses for the 150 and similar 16-to-one multiplexers, such as the 250, the 850, or the 851, you find that vendors have labeled them obsolete or no longer available. On the other hand, the eight-line-to-one-line multiplexers not only have survived but also are parts of advanced logic families, such as HC (high-speed CMOS) and AC (advanced CMOS).

The circuit in **Figure 1**, a 10-line-to-one-line data selector/multiplexer, comprises two eight-to-one multiplex-

ers, IC₁ and IC₂. The A, B, and C bits of the address input of IC₁ connect to corresponding address bits—A, B, C, and D—of the main address input. The eight data inputs, D0 to D7, of the circuit are identical to the equally denoted data inputs of IC₁.

Whenever the main address is a binary-coded eight or nine, when A, B, C, and D=eight, the data input, D4 of IC₂, is active. When A, B, C, and D=nine, D5 of IC₂ is active. This shift in addressing of IC₂'s data inputs is due to the IC's modified addressing: Address bit C connects to the MSB (most-significant bit) D of the main address input. The A and B are common to IC₁ and IC₂, respectively. To unite their outputs without using any additional logic, you must connect the noninverting output, Y,

DIs Inside

42 Implement a simple digital-serial NRZ data-recovery algorithm in an FPGA

44 LED strobe has independent delay and duration

48 Cancel sensor-wiring error with bias-current modulation

50 Simple FSK modulator enables data transmission over low-speed link

► To see all of EDN's Design Ideas, visit www.edn.com/designideas.

of IC₁ to data inputs D0 through D3 of IC₂. The eight lowest values, zero through seven, of the address always activate a signal of D0 through D3 in

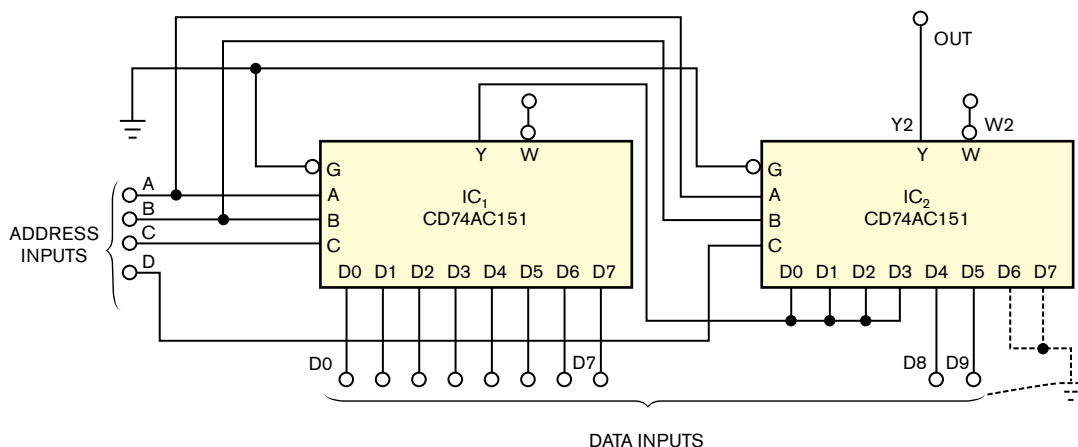


Figure 1 The maximum worst-case propagation delay of this 10-line-to-one-line data selector/multiplexer is 27 nsec, whereas the typical value is only 6.8 nsec. The circuit can also serve as a 12-to-one multiplexer.

IC₂. The output signal of IC₁ passes through one of these data inputs to the main output, Y2. If necessary, you can also use the W2 inverting output. Although the propagation delay from D0 through D7 to Y2 output is twice that from D8 and D9 to Y2, it is still

less than $2 \times 13.5 \text{ nsec} = 27 \text{ nsec}$ for the CD74AC151 with a 5V supply. The typical delay is only 6.8 nsec.

Note that you can also use the circuit as a 12-line-to-one-line data selector/multiplexer by using the remaining data inputs, D6 and D7 of IC₂, which

are idle in this circuit. In such a case, you attribute another notation of D10 to the D6 input of IC₂, and D11 holds for D7 one. Simultaneously, you must code the A, B, C, and D address in duodecimal code and, eventually, hexadecimal code, instead of BCD. **EDN**

Implement a simple digital-serial NRZ data-recovery algorithm in an FPGA

Jef Thoné and Bob Puers, ESAT-MICAS, Katholieke Universiteit Leuven, Leuven, Belgium

Serial-data links embed clocks in their data streams, and those clocks must be recovered at the receiver end. This Design Idea describes a data/clock-recovery algorithm for an NRZ (non-return-to-zero), 1.5-Mbps data stream in a Xilinx (www.xilinx.com) Spartan XC3S200 FPGA. The algorithm employs a modified data-recovery application note (**Reference 1**). The application note uses the DCM (digital-clock manager) on the Xilinx Spartan and Virtex models, but this application uses a simplified algorithm that compares the data edges, if any, with internally generated clock edges, dynamically changing the data-input-to-data-output delay. The simplified algorithm allows integration in smaller CPLDs or FPGAs that lack a DCM (**Figure 1**).

The algorithm uses a 3-bit, free-running counter to generate the output clock, an 8-bit shift register to sample the serial data, seven XOR ports for

edge detection, a 7-to-1 multiplexer with decoding for multiplexing the right-shift-register bit to the output,

and some buffering registers. The algorithm runs at eight times the serial-data-stream speed, without a known phase relationship between both. It clocks the data into the shift register, which implies that, after eight clock cycles, the shift register will contain a rising edge; a falling edge; or, when the input data remains the same, no edge. The multiplexer does not take into account cases in which the shift register contains no edges or more than one edge.

The edge location is checked in the shift register using the XOR-port array, which compares shift-register bit 0 with bit 1, bit 1 with bit 2, and so on. Depending on the output of the XOR array, showing where the edge occurs, a certain bit of the shift register multiplexes to the output. This action ensures that the output clock always toggles around the middle of the output-data bits.

When there are slight differences in clock speed and serial-input-data speed—for example, in the case of clock jitter or clock tolerances—the data-input phase continuously changes with regard to the output-clock phase as the algorithm tries to track the input-data phase. In this case, the multiplexer has an overflow, which happens when shift-register bit 7 multiplexes to the output, the next bit is

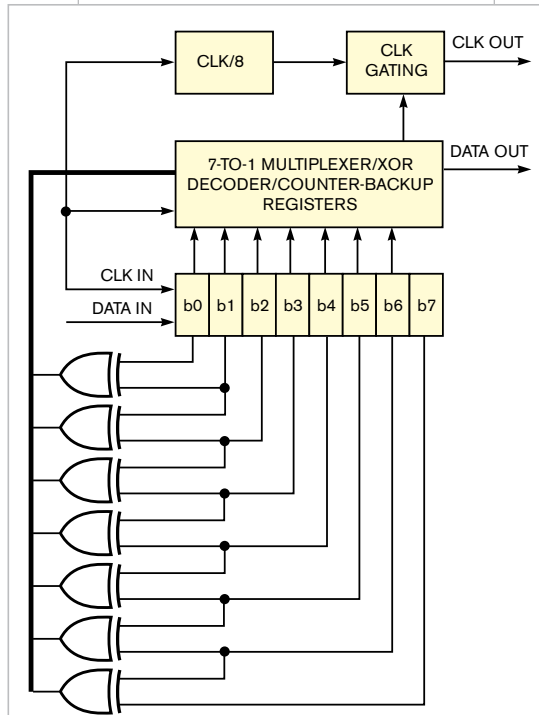


Figure 1 A clock-recovery circuit in an FPGA recovers data in a 1.5-Mbps data stream.

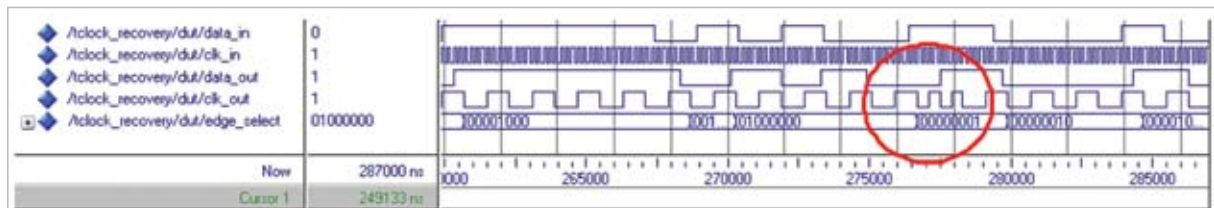


Figure 2 Doubling a clock output prevents a backward phase jump.

shift-register bit 1, or vice versa.

If bit 7 is output first—that is, the signal `edge_select` is 0100 0000—and the next selected bit is bit 1, with an `edge_select` of 0000 0001, a sudden phase jump in output data occurs. This phase jump is $-360^\circ \times 7/8$, or -315° . Because the next input-data bit already had shifted in completely in the shift register, you need to employ a double-output clock once, so that the register

doesn't miss a data bit (circled area in **Figure 2**).

When bit 1 is output, with an `edge_select` of 0000 0001, and the multiplexer jumps to bit 7, with an edge select of 0100 0000, a sudden phase jump in output data of $360^\circ \times 7/8$, or 315° , occurs. Because the shift-register data bit 7 is a delayed version of the last clocked bit, b1, the output clock must be stalled for one cycle. Otherwise, one bit too many

will clock at the output (circled area in **Figure 3**). You can solve the overflow-phase jumps by gating the output clock using combinatorial logic. **EDN**

REFERENCE

1 Sawyer, Nick, "Data Recovery," Xilinx Application Note XAPP224, Version 2.5, July 11, 2005, www.xilinx.com/support/documentation/application_notes/xapp224.pdf.

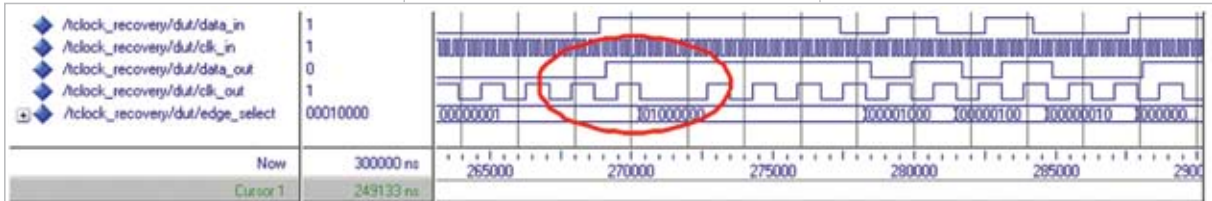


Figure 3 Stalling the output clock prevents a forward phase jump.

LED strobe has independent delay and duration

Michael C Page, Chelmsford, MA

The circuit in **Figure 1** is not complex, but it saved the day in an application involving visual inspection of the spray pattern of fuel injec-

tors for quality and consistency. In this application, xenon strobe lights did not work because they take up too much space, and the light they emit is too

intense. With a bank of six injectors with isolation panels, the reflection off a person's shirt or the wall behind him would interfere with the visual inspection. So the application instead used white HB LEDs (high-brightness light-emitting diodes) on "gooseneck"-type stands for adjustability in the chamber sections. Although the applica-

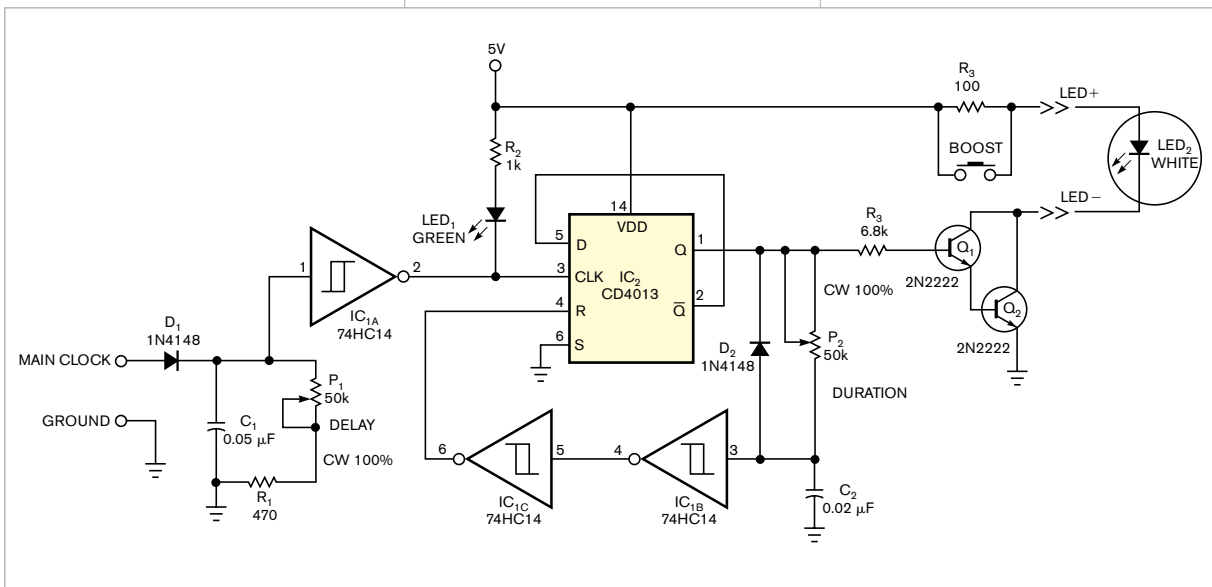


Figure 1 This circuit employs HB LEDs for a visual-inspection application.

tion could have used the trusty old 555 timer, the delay and duration duty-cycle controls interact, which is an awkward situation.

The circuit in **Figure 1** shows the main-clock input; the delay and duration potentiometers, P_1 and P_2 ; and the HB-LED output. The circuit also includes an on-board general-purpose LED for bench testing to indicate an input signal, although, when the circuit is operating at high speeds, this LED is useless. The main-clock input is a 5V pulse of approximately 30 μ sec coming from the fuel-pump index. Delay potentiometer P_1 adjusts the on-time delay of the LED from about 40 μ sec to 2 msec, and duration potentiometer P_2 adjusts the LED-on, or flash, time with a range of approximately 15 μ sec to 15 msec.

The circuit applies a 5V pulse, the main clock, to diode D_1 and capacitor C_1 to form a peak-hold circuit. C_1 then discharges at a rate that P_1 sets. Schmitt trigger IC_{1A} monitors C_1 's voltage, and, when it reaches the low threshold of IC_{1A} , it outputs a high level to IC_2 's clock input, setting the Q output high. With IC_2 's Q output high, the Darlington-transistor pair comprising Q_1 and Q_2 turns on, driving the output to the HB LED low at the output, lighting the LED. At this time, capacitor C_2 charges at a rate that P_2 sets. When this voltage reaches the upper threshold of IC_{1B} , IC_{1C} 's output switches to high, resetting flip-flop IC_2 's output back to low and turning off the HB LED. The circuit is now ready for another round. Diode D_2 ensures a complete discharge of capacitor C_2 for repeatability when you reset the Q

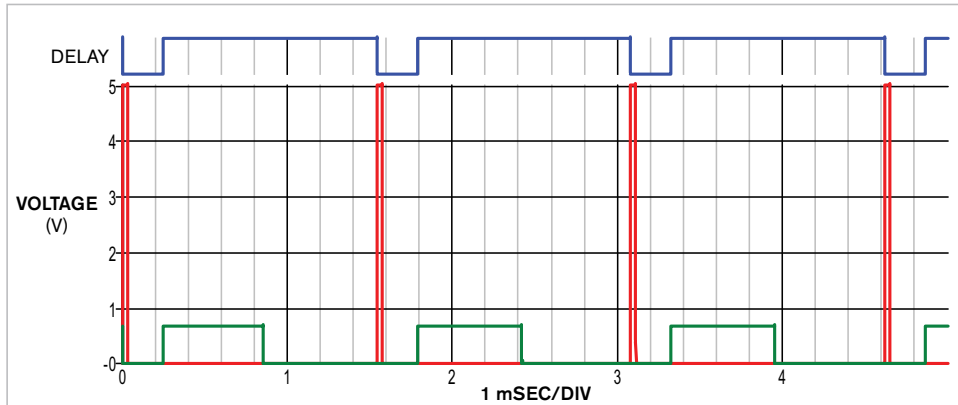


Figure 2 With a main-clock input of 650 Hz, the delay is approximately 250 μ sec, with P_1 at 10%, and the duration is approximately 600 μ sec, with P_2 at 75%. The top trace (blue) represents the strobe delay, the lower trace (green) represents Q_1 's base duration, and the 5V trace (red) represents the main clock.

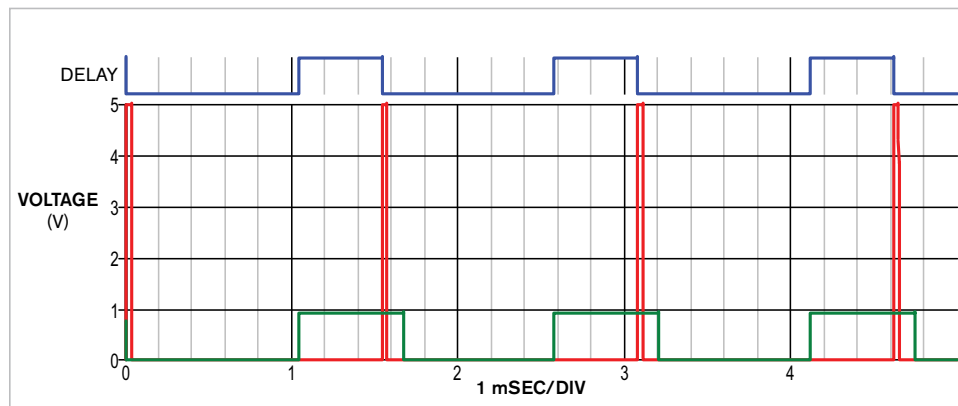


Figure 3 An adjustment change of delay occurs with the same duration as in **Figure 2**. The top trace (blue) represents the strobe delay, the lower trace (green) represents Q_1 's base duration, and the 5V trace (red) represents the main clock.

output of IC_2 to low. Because IC_2 requires an active-high signal, you can omit IC_{1B} and IC_{1C} , but you should use a Schmitt trigger following an RC circuit for repeatability, especially on slow capacitor-charge/discharge times.

Figure 2 shows the results of the circuit running with a main-clock input of 650 Hz and a delay of approximately 250 μ sec, with P_1 at 10%, and a duration of approximately 600 μ sec, with P_2 at 75%. **Figure 3** shows an adjusted change of delay with the same duration setting as in **Figure 2**. The new flash period overlaps the following fluid burst. You could, depending on the injector nozzle, see the end of one fuel burst of calibration fluid and the

start of another in the chamber during the same flash period without encountering an error. The circuit also has a boost switch for a momentary intensity increase; otherwise, R_3 normally limits the current to approximately 40 mA. When you press the boost switch, the Darlington pair, two 2N2222 transistors with current of approximately 400 mA, still limits the current, but long-term use of the switch will shorten the LEDs' life. You should tailor the values of C_1 , C_2 , P_1 , and P_2 to the application. Calculations will vary depending on the logic family you use, but, generally, $T=0.7 \times R \times C$, where T is the time in seconds, R is the resistance, and C is the capacitance. **EDN**

Cancel sensor-wiring error with bias-current modulation

W Stephen Woodward, Chapel Hill, NC

The approximately $-2\text{-mV}/^\circ\text{C}$ temperature coefficient of diode junctions is a popular means of temperature measurement, especially in cryogenic applications (Figure 1).

Diode temperature sensors are compact, stable, robust, sensitive, and inexpensive, and, unlike thermocouples, they require no reference junction. All of these benefits help explain the dura-

ble popularity of this—to use the polite term—“mature” technology.

A complicating factor and potential error source affecting these sensors arises from their need for bias-current excitation, however. The resulting contribution of ohmic IR (current/resistance)-voltage drop in the wiring and the connectors’ resistance to the sensor’s output voltage create spurious and temperature-sensitive voltage offsets. These offsets can introduce unacceptably large measurement error. This situation is especially likely when you use small and, therefore, high-resistance-gauge wire for sensor cabling, such as in cryogenic applications. In those cases, designers prefer exceptionally fine-gauge wire to minimize thermal conductivity and leakage.

The usual solution to the IR problem is to employ four-wire “Kelvin”-interconnection topologies, in which one pair of conductors carries the sensor’s bias current and a separate, independent pair differentially senses the sensor’s output voltage. This approach prevents corruption of the sensed voltage by IR drop in the bias pair. This traditional fix works well but complicates the wiring and doubles undesirable thermal leakage due to the extra wires, thus defeating much of the point of using fine-gauge cabling in the first place.

Figure 2 illustrates a circuit that implements a different approach. It cancels the wiring-resistance error and needs only two conductors in the sensor cable. It takes advantage of the fact that IR-voltage drop is directly proportional to current, but the sensor voltage is mostly constant. It works by alternating the magnitude of the excitation current, I_B , between two values, I_{B1} and I_{B2} , where $I_{B1} = 2I_{B2}$. The ac component of the resulting signal is thus approximately $I_B R_W$, where R_W is the total wiring resistance plus a minor contribution from nonzero sensor impedance.

The clock for both I_{B1}/I_{B2} excitation modulation and synchronous demodulation of the resulting response is the internal oscillator of the LTC1043, which you set to approximately 500

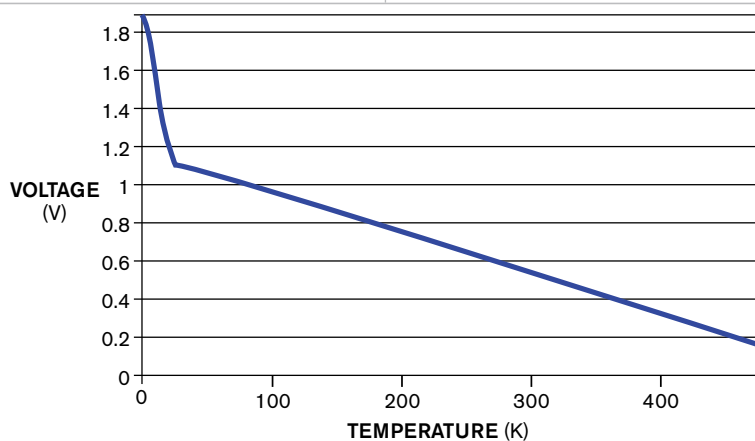


Figure 1 The typical $-2\text{-mV}/^\circ\text{C}$ -voltage-versus-temperature coefficient of diode sensors is large and nearly constant over a wide range of temperatures.

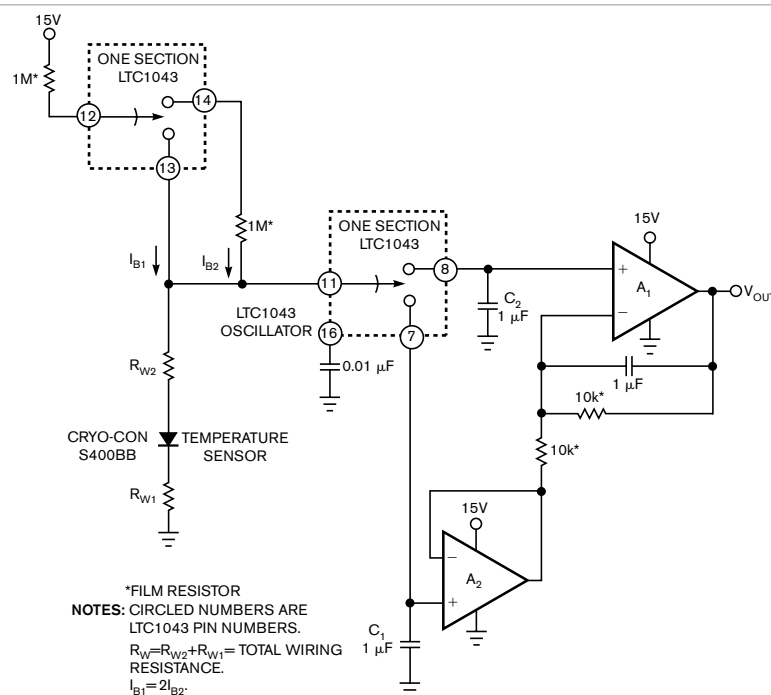


Figure 2 This circuit cancels the wiring-resistance error inherent in diode temperature sensors and requires only two conductors in the sensor cable.

Hz by connecting the external 0.01- μ F capacitor to Pin 16. The resulting toggling of the excitation ballast resistance between 1M and 1M+1M=2M creates the 2-to-1 current modulation and an ac-signal component proportional to wiring resistance: $I_B R_W$.

The other side of the LTC1043 synchronously rectifies the $I_B R_W$ ac component, storing the $I_{B1} R_W = V_{C1}$ phase on C_1 and the $I_{B2} R_W = V_{C2}$ phase on C_2 . Op amp A_2 buffers V_{C1} and inputs it to the resistor network and A_1 , which subtracts it from the average sensor sig-

nal, producing an output voltage independent of cabling-resistance offset. One downside of the technique is that, due to sensor-impedance effects on the order of 20 mV, the thermometric diode usually requires custom temperature calibration. **EDN**

Simple FSK modulator enables data transmission over low-speed link

Israel Schleicher, Prescott Valley, AZ

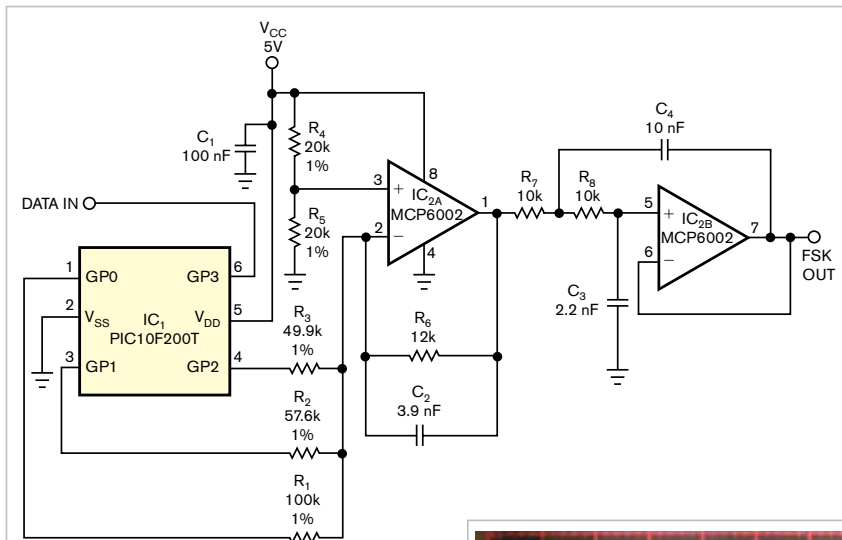


Figure 1 This microcontroller-based circuit generates Bell 202-compatible FSK modulation.

FSK (frequency-shift keying) is a type of signal modulation for transmitting digital data over an analog communication link. An FSK modulator comprises a digitally controlled sine-wave generator whose frequency shifts between two predetermined frequencies in response to the two logic levels of the digital data. The circuit in **Figure 1** generates a sine wave by continuously sampling a single sine cycle. The output of IC_{2A} is proportional to the currents through R_1 , R_2 , and R_3 . These resistors connect together at one end to the inverting input of IC_{2A} , which is biased at $V_{CC}/2$. The

age across the resistor is $-V_{CC}/2$. Select the values of R_1 , R_2 , and R_3 so that the current pulses have magnitudes proportional to samples of $\sin 30^\circ$, 60° , and 90° , respectively. Setting all the outputs of IC_1 to off produces the sample of $\sin(0^\circ)$, and no current flows through the resistors. Thus, starting with all outputs of IC_1 at off and consecutively and periodically setting GP0, GP1, and GP2 to high and then, in reverse order, setting GP1 and GP0 high again generates the positive half of a sine wave. Repeating the process but setting the outputs to low generates the negative half of the waveform.



Figure 2 The FSK modulator's output changes frequency based on a digital input.

outputs GP0, GP1, and GP2 of microcontroller IC_1 produce nonoverlapping pulse trains. When you set either output high or low, the others are off—that is, at high impedance. When you set an output high, the voltage across the resistor that connects to it is $V_{CC}/2$. When you set the output low, the volt-

This scheme produces a sampled sine waveform with 12 samples per cycle. In addition to the desired frequency component, f_0 , this waveform contains higher-frequency components at $(12k+1)f_0$ and $(12k-1)f_0$, $k=1,2,3$, and so forth. The lowpass filter comprising IC_{2B} , R_7 , R_8 , C_3 , and C_4 easily filters out these undesired components of smaller amplitude. **Listing 1**, which is available with the Web version of this Design Idea at www.edn.com/090611dia, is the assembly-program code that implements the Bell 202 FSK standard. When the control input

Data In is high, the output frequency is 1200 Hz; when the control is low, the output frequency is 2200 Hz. The transition from one frequency to the other occurs in a manner that retains phase continuity. **Figure 2** shows the FSK-modulator output (CH1) in response to a modulating signal (CH2). **EDN**