design ideas

Edited by Bill Travis and Anne Watson Swager

Scheme autodetects baud rate

Alexander Eisen, State University of New York, Buffalo, New York

THE POPULARITY AND easy access of RS-232 ports lend them to many communication projects. You can use a port "as is" or as a tiny parallel port when the exchange uses only control lines. Before the asynchronous serial-data transfer between two devices can take place, you must ensure that both devices are configured to the same data format and transmission rate. Usually, the OS utility or application program on the host computer performs this task, which the OS or a combination of switches selects in the peripheral.

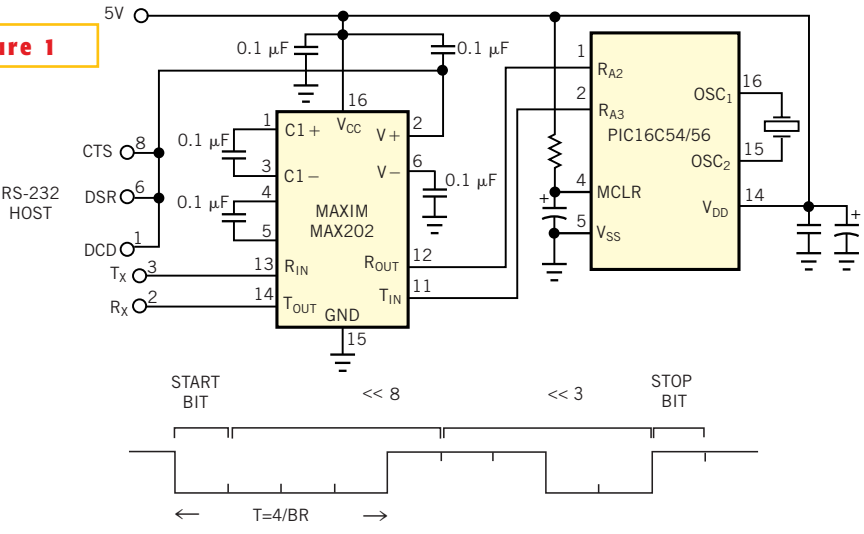
The design in **Figure 1** and the accompanying software realize the requirement of automatically equalizing the speed of the parties by adapting the bit rate of a µC's serial port to that of a host computer. The host sends a known ASCII code to a µC that estimates the bit rate of the transmission. The µC samples the receiving end of a transmission line. As soon as the line goes low, indicating a start bit, the µC clears the timer. After the line again goes high, the µC reads the timer's value and uses this value to calculate the bit rate of a transmission. The µC uses the calculated bit rate in a timing-delay loop for clocking out the data from the µC to the host computer during transmission and for sampling the incoming data during receiving mode. The time between those falling and rising edges on a transmission line depends, of course, on the transmitted ASCII symbol. Its good practice to use a symbol with 2^k-1 zeros on the low side. Then, when you include the start bit, there are 2^k trailing zeroes. One timing unit is easier to derive by substituting a division with k right shifts.

The application in **Figure 1** uses an 8-bit RISC µP and an ASCII code of 8 (38 hex) to establish a serial communication. Any symbol that ends with 8H also works. The timer increments its value every $4P/F_{OSC}$ seconds, where P is a prescaler factor and F_{OSC} is the µP's clock frequency. Thus, the timer's value after 4 bits have transmitted during the time between the falling and rising edge, is $T=4/BR$, is $N=F_{OSC}/(BR\times P)$, where BR is the bit rate.

Now you can derive the 1-bit transmission time, which is a reciprocal of a transmission speed, as $(N/4)(4P)/F_{OSC}=N\times P/F_{OSC}$.

Because the µP's timer is only 8 bits long, it is important to pick up the right prescaler factor. N must not exceed the timer's capacity of 255 but should be large enough to cover a certain range with good resolution. For the worst case, which is when the bit rate is at its minimum, P should be $P>F_{OSC}/(N_{MAX}\times BR_{MIN})$.

Figure 1



Using its internal timer, a µP calculates the bit rate of transmission from a host computer and adapts its serial port's bit rate to match.

After you determine all of these timing relationships, you need to design a software loop for a 1-bit delay. One machine cycle for the µP in **Figure 1** is $4/F_{OSC}$, so the total cycle count for the execution time of a loop should be NP/4. The loop for a bit_clk is 4(k+2) cycles long (**Listing 1**). Therefore, k=(NP/16)−2. If you choose P=16, k then conveniently becomes N−2. At F_{OSC}=8 MHz, these parameters reliably cover speeds of 2400 to 38,400 bps, or baud rate, which in the case of the serial port is the same thing. (DI #2546)

To Vote For This Design, Enter No. 328

LISTING 1—AUTOMATIC BIT-RATE-DETECTION ROUTINE

```
; Bit rate automatic detection routine
;*********************************************************
ser_out         =       ra.3            ;transmit pin
ser_in          =       ra.2            ;receive pin
;
        org     8
;
k                       ds      1       ;derived time constant
clk_cnt                 ds      1       ;counter for serial timing
bit_num                 ds      1       ;number of received/transmitted bits
xmt_b                   ds      1       ;byte to transmit
;
        device pic16c54,xt_osc,wdt_off,protect_off
        reset start
;
        org     0
;
start           mov     !ra,#4          ;ra.2 - input, ra.3 - output
                mov     option,#3       ;set prescaler to divide by 16
;
;******************** serial in ***************************************
;
st_by           snb     ser_in          ;skip next instruction if Rx goes low
                jmp     st_b            ;go back and wait till start bit
                clr     rtcc            ;reset timer, start to count ticks
                jb      ser_in,st_by    ;go back if it was only a glitch
;
;******************** start is good ***********************************
;
roll            jnb     ser_in,roll     ;wait till line goes high
                mov     k,rtcc          ;record timer's value
                dec     k               ;make k = N-2
                dec     k               ;decrement
;
;******************** success *****************************************
;
                mov     xmt_b,#'O'      ;output "OK" using calculated
                call    out             ;Bit Rate
                mov     xmt_b,#'K'
                call    out
                mov     xmt_b,#13       ;carriage return
                call    out
                mov     xmt_b,#10       ;line feed
                call    out
                goto    do_something    ;continue with your application
                nop
;
;****************** serial out ****************************************
;
out             mov     bit_num,#8      ;number bits to transmit
                clrb    ser_out         ;ready to send start bit
                call    bit_clk         ;send start bit
xmit            rr      xmt_b           ;move LSB to carry
                movb    ser_out,c       ;move carry to Tx
                call    bit_clk         ;send
                djnz    bit_num,xmit    ;decrement and go on if not done yet
                setb    ser_out         ;load stop bit
                call    bit_clk         ;and send
                ret                     ;done
                nop
;
bit_clk         mov     clk_cnt,k
loop            nop
                djnz    clk_cnt,loop
                ret
                nop
```

Circuit computes first derivative

Richard Panosh, Vista, Bolingbrook, IL

The circuit in **Figure 1** computes the derivative of an input signal as the integral of the input signal minus the signal itself. The response of the circuit is

$$V_{OUT} = -\frac{R_2}{R_1} V_{IN} \frac{R_{EQ} C_1 s}{1 + R_{EQ} C_1 s},$$

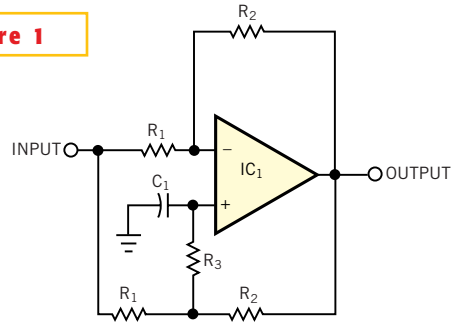
where R_{EQ} is the parallel equivalent resistance of R_1 and R_2 plus the resistance of R_3, or

$$R_{EQ} = \frac{R_1 R_2 + R_1 R_2 + R_2 R_3}{R_1 + R_2}.$$

This response is identical to that of the classic inverting differentiator, in which the response is $V_{OUT} \simeq -R_{EQ} C_1 s V_{IN}$ for input frequencies lower than $1/2\pi R_{EQ} C_1$. If R_3 is much greater than the parallel combination of R_1 and R_2, then $R_{EQ} \simeq R_3$. On the other hand, if R_3 is set at 0Ω, the bias currents of the op amp balance to minimize voltage offsets, and

$$R_{EQ} = \frac{R_1 R_2}{R_1 + R_2}.$$

Figure 1



This circuit produces less noise than the classic inverting differentiator.

A major problem with the classic inverting differentiator is high noise. By its nature, a differentiator must exhibit increasing gain with frequency, and this increasing gain amplifies the inherent amplifier noise. An equivalent input-noise voltage, V_N in the classic inverting differentiator, produces output noise of magnitude $RCsV_N$. In the case of the differentiator in **Figure 1**, the equivalent input noise produces an output magnitude of only $(1+R_2/R_1)V_N$. (DI #2522)

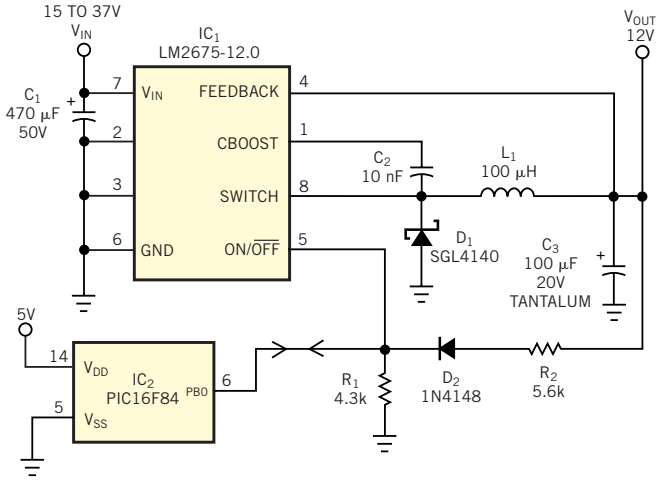
To Vote For This Design, Enter No. 329

Practice safe dc/dc converter

Eugene Kaplounovski, Nautilus International, Vancouver, Canada

Short-circuit protection is an obvious requirement for a power supply, especially when its load connects to a cable that's subject to damage. Many modern power-converter ICs include some means of protection, such as thermal shutdown, against the overload condition, but, in some cases, the built-in protection may be inadequate. **Figure 1** shows a step-down dc/dc converter for two videocameras installed on a remotely controlled vehicle. Because the vehicle operates in rather harsh environments, the cables can frequently short-circuit during the system's installation and normal operation. The cameras require 12V and consume approximately 250 mA each. The converter is based on National Semiconductor's LM2675 chip, which by itself includes good protection against overloads. However, in the case of a shorted output, the catch diode, D_1, must withstand the maximum current from the IC. This current, according to the data sheet, can be as high as 2.2A, calling for an oversized diode. Also, waiting for the thermal protection to kick in assumes that you are willing to allow the device's temperature to rise significantly, heating adjacent components. This situation is undesirable, given the long periods of nonsupervised operation, during which someone should notice the problem. Ideally, someone should immediately report the faults to an operator.

Figure 1



This dc/dc converter provides flexible overload protection and diagnostics.

The system in **Figure 1** uses a Microchip Technology PIC16F84 µC, which receives its power from a separate dc source. As usual, the I/O pins in the µC are at a premium, because of the multitude of other tasks the controller performs. However, you can obtain reliable short-circuit protection and on/off control of the power supply using just one I/O pin. When you first apply power, the µC starts up. In this condition, all of its I/O lines are high-impedance inputs, so the LM2675 cannot start; resistor R_1 ties its On/$\overline{\text{Off}}$ pin low. After the initialization routine, when it is time to turn on the camera, the µC makes its pin PB0 pin (or any pin that has a totem-pole driver) an output and sends a high level to that output. The dc/dc converter starts up. After a short delay, the µC again makes its PB0 pin an input, but the power supply keeps itself on because its output voltage connects to control input via R_2 and D_2. This condition prevails while the load is normal. When the output short-circuits, the bias voltage disappears, and the chip shuts itself down. The level at PB0 goes low, notifying the µC of this condition. (PB0 is especially useful in this situation, because it can generate an interrupt request.) The µC can then alert the user of the failure, try to restart the converter after a delay in a "hiccup" mode, or both.

The duration of the worst-case short-circuit condition with this scheme is a function of the length of the start-up pulse from the µC. This pulse should be long enough—usually, approximately 10 msec—to allow the normal load (with its own input capacitors and power converters) to start. The LM2675 with a catch diode in **Figure 1** withstands short circuits for several seconds without overheating or any other problem, so the short-circuit mode is perfectly safe. An added benefit is that the µC can at any time shut down the dc/dc converter, making it possible to save battery power and reduce heat dissipation. For shutdown, the µC again makes PB0 an output but sends a zero to the I/O pin. The powerful driver in the PIC16F84 easily overcomes the bias from R_2, D_2, and R_1 and shuts down the LM2675. The divider R_2, D_2, R_1 provides a voltage—5V in this case—close to the µC's V_{DD} when the output voltage is normal. Slight voltage variations cause no harm to the µC, thanks to the controller's input-diode protection and the fact that R_2 limits the current to the PB0 pin to a safe level. However, you should keep the values in the divider low enough to not create a significant voltage drop in R_1 by the bias current from the LM2675's On/$\overline{\text{Off}}$ pin (37 µA maximum). With the values shown, the largest bias current creates a drop that does not exceed 20% of the lowest possible threshold (0.8V) for the On/$\overline{\text{Off}}$ pin. (DI #2562)

To Vote For This Design, Enter No. 330

Single-sideband demodulator covers the HF band

Israel Schleicher, Bakersfield, CA

THE CIRCUIT IN **Figure 1** complements a previous Design Idea (**Reference 1**). The modulator used a phasing network to split a low-frequency (audio) signal into in-phase and quadrature (orthogonal) components. The phasing network has an advantage over other phasing circuits in that it delivers a phase error of only 0.15° and has low sensitivity to component tolerances (**Reference 2**). By reversing the direction of the network—in other words, feeding the output with two orthogonal signals and tapping the input—the network functions as a detector. Feeding the two signals one way may produce a signal at the input, but if you interchange the two signals, no signal goes through. Because the network in the modulator circuit has two floating differential inputs, for the demodulator you must feed them from two floating sources.

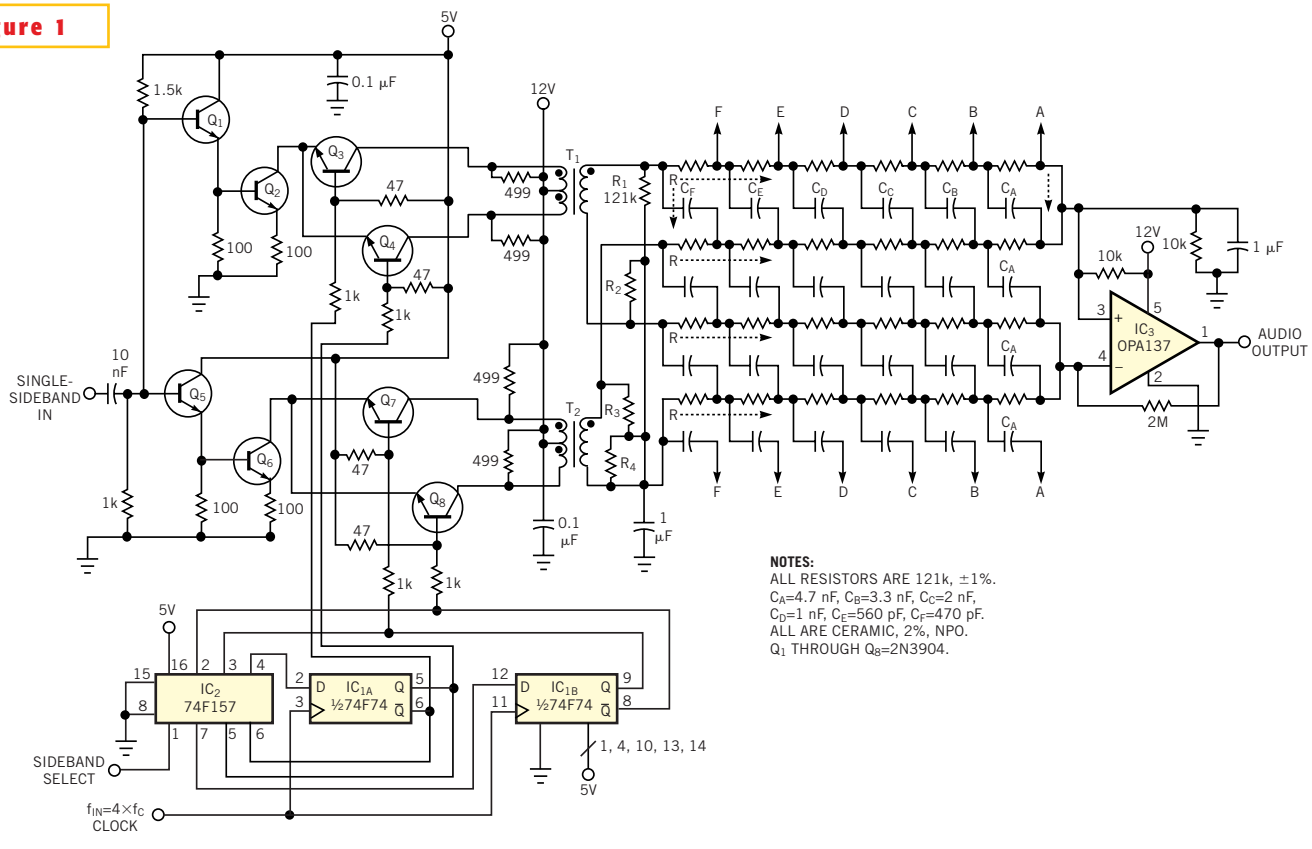
The simplest way to obtain the feed signals is to use transformers. T_1 and T_2 are 600Ω, 1-to-1 telephone-coupling transformers with center-tapped bifilar primary winding. It is important to minimize the capacitance between the primary and secondary. Q_1 through Q_4 and Q_5 through Q_8 function as balanced mixers. They provide a wide dynamic range to the circuit. The circuit forms part of a direct-conversion receiver. IC_1 provides two quadrature local-oscillator signals. IC_1 requires a drive signal with four times the carrier frequency. IC_2 allows upper or lower sideband selection. Measurements on the prototype circuit show 37 dB of unwanted-sideband rejection for a 1-kHz modulated carrier and 32 dB of rejection for a 3-kHz modulating signal. You must use a sharp-roll-off, 3-kHz lowpass filter with the circuit. (DI #2563)

REFERENCES

1. Schleicher, Israel, "SSB modulator covers the HF band," *EDN*, Sept 30, 1999, pg 122
2. Zavrell, Robert Jr, "New low-power single-sideband circuits," Philips Semiconductor, Application Note AN1981.

TO VOTE FOR THIS DESIGN, ENTER NO. 331

Figure 1



NOTES:
ALL RESISTORS ARE 121k, ±1%.
C_A=4.7 nF, C_B=3.3 nF, C_C=2 nF,
C_D=1 nF, C_E=560 pF, C_F=470 pF.
ALL ARE CERAMIC, 2%, NPO.
Q_1 THROUGH Q_8=2N3904.

This single-sideband demodulator provides sharp rejection of the unwanted sideband.

2.8-GHz prescaler keeps cost down

Neil Eaton, Emsys Engineering, Peterborough, ON, Canada

THE PRESCALER in **Figure 1** inexpensively extends the range of a frequency counter by dividing the input signal's frequency by a factor of 1000. The guaranteed input-frequency range of the input prescaler, IC_1, is 250 MHz to 2.8 GHz, although typical values are 100 MHz to 3.5 GHz. The prototype operates at frequencies well below 100 MHz, but its fastest generator goes only to 1.7 GHz, so you cannot confirm the upper range. The input-voltage range is 400 to 1000 mV p-p from 250 to 500 MHz and 100 to 1000 mV p-p for higher than 500 MHz. IC_1 serves as a divide-by-128 prescaler, whose output is a 1.6V p-p square wave. The RC network level-shifts the output of IC_1 to ensure that the top of the square wave is above the 2V input threshold of IC_{5A}. The output of IC_{5A} is a 5V, CMOS-compatible square wave with a frequency of $^1/_{128}$ of the input frequency. Most frequency counters can handle these frequencies, but the submultiple is inconvenient for an operator. A further division by a factor of 7.8125 (1000/128) produces a scaling factor of 1000.

Fortunately, the frequency counter averages its input over many cycles, so the output of the prescaler need not be exactly $^1/_{1000}$ of the input frequency for every input pulse. The 0.8125 figure is 13 divided by 16. The average frequency ratio is therefore 7.8125 if you divide 13 output pulses of 16 by eight and the remaining three by seven. For best results, the divide-by-seven pulses should be as evenly spaced as possible. The result is a repeating sequence 16 output pulses long with the following pattern:

Pulse number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Divide by	8	8	8	8	7	8	8	8	8	7	8	8	8	8	7	8

IC_2 divides the prescaler's output by seven or eight, depending on the state of the B input. IC_3 and IC_5 both connect to the output of IC_2, so they count output pulses, not prescaler pulses. IC_5 divides the output by five to generate the divide-by-seven periods. IC_4 divides the output by 16 to reset the cycle upon completion. Without IC_4, the cycle would continue to divide by seven at every fifth output pulse for a ratio of 7.8. The construction of the circuitry inside the dashed lines in **Figure 1** is critical. The MC12079 is available only as a surface-mount device. In the design, it and its associated passive components is mounted on a Surfboard (Capital Advanced Technologies Model 9081, available from Digikey). It is then fastened, component side up, to a bare copper-clad board to create a ground plane. All connections to ground from the high-

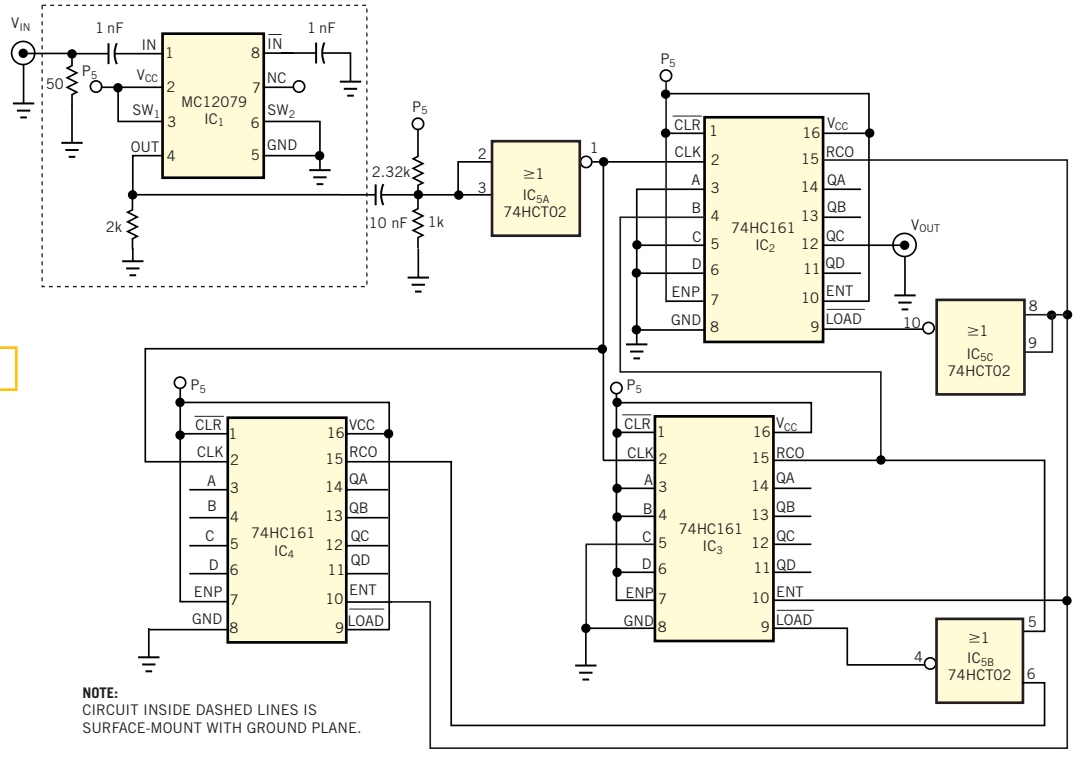


Figure 1

Extend your frequency-measurement capability with this inexpensive divide-by-1000 prescaler.

speed circuitry go directly to the ground plane using short lengths of copper braid. (Desoldering wick is ideal.) The input uses a BNC chassis-mount connector with its shell soldered directly to the ground plane, and the center pin connected to the Surfboard with the shortest possible wire. The rest of the circuit is noncritical. To avoid clutter, **Figure 1** shows no bypass capacitors, but you should place them near every IC. (DI #2564)

To Vote For This Design, Enter No. 332

Add harmony to your system

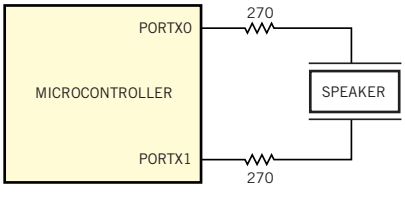
Eugene O'Bryan, Food and Drug Administration, Rockville, MD

Have you ever wished you could distinguish one device's operating state from another's by the sounds they make or that error states would sound harsh while normal operations would sound harmonious? By combining the NCO technique from a previous Design Idea (**Reference 1**) with digital mixing, you can obtain musical chords or intervals with a minimal amount of hardware and software. Any µC system can thus produce a variety of sounds. In **Figure 1**, a piezoelectric speaker, Radio Shack Model 273-091, and two 270Ω resistors transduce a pulse stream from a µC. Differential drive to this transducer increases the volume by doubling the effective voltage. The capacitance of the piezoelectric speaker reacts with the 270Ω resistors to integrate and smooth the pulse stream. The software (**Listing 1**), a tight loop, comprises a square-wave generator and an NCO's (numerically controlled oscillator's) summing part. After it sets up some registers, the sound-generating loop establishes an output level for each of two or more square waves.

The pseudocode example in **Listing 1** demonstrates a two-note generator in which the output levels of two square waves are established in registers r_vol1 and r_vol2. The frequency of each square wave is a function of the values set for variables first_note and second_note and by the cycle rate of the loop. Note that a half-cycle of a square wave concludes when the corresponding counter register (r_cnt1 or r_cnt2) reaches zero. The summing part of the loop uses the NCO technique to generate an output level for the digital mixture of the square waves generated in the first part of the loop. The mixing of two square waves with frequencies of 880 Hz, the C above middle C, and 988 Hz, the second D above middle C, in the equal-tempered scale produces the oscilloscope waveform in **Figure 2**. An AVR AT90S Series µC runs through the sound-generating loop in 18 clock cycles. Thus, to produce an 880-Hz square wave with this type of µC running at 8 MHz, the first_note or second_note value is set at 253, which is equal to 8 MHz/18/880/2.

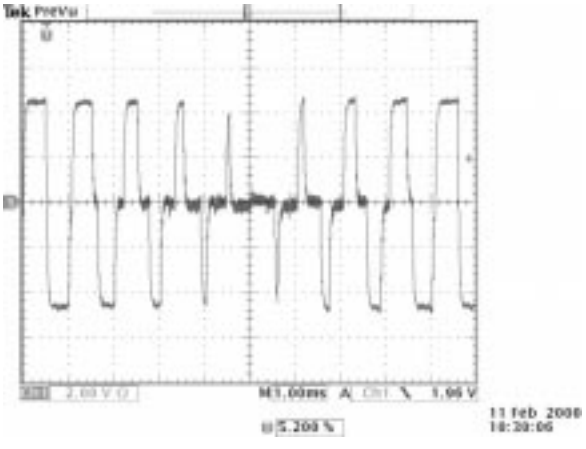
A note of caution is in order in calculating this value in an assembly program: Be careful of truncation issues; the musical intervals sound wrong if the frequencies are off. Also, when you use another type of µC, you must adjust the loop software to compensate for differences between the µC's instruction timing and the assumed instruction timing of the pseudocode. All instructions should complete in one clock cycle except for jump or branch executions, which take two clock cycles. You can create chords by expanding the software loop to include a third square-wave generator and modifying the mixer to add in the third note. A by-product of this sound-generation scheme is that it lets you control volume by changing the volume variable in **Listing 1**. The loudest possible volume occurs when the volume variable equals 80hex for a two-note generator. Setting the volume variable lower than 80hex reduces the voltage output from each square wave. With this scheme, you can produce 10 distinct volume levels, using hex values 80, 6A, 60, 58,

Figure 1



Produce harmonious sounds in your system, using this simple scheme.

Figure 2



Mixing square waves of 880 and 988 Hz produces this waveform.

50, 48, 40, 38, 30, and 28. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2561. (DI #2561)

REFERENCE

1. Ploss, Steve, "NCO technique helps μC produce clean analog signals," *EDN*, April 15, 1999, pg 129.

To Vote For This Design, Enter No. 333

LISTING 1—ASSEMBLY CODE FOR SOUND GENERATION

```
;****************************************
; Setup
;****************************************
  ldi  r_cnt1,first_note  ; load counter1 register with first_note value
  ldi  r_cnt2,second_note ; load counter1 register with second_note value
  ldi  r_vol,volume       ; load volume register with a value (0-80hex)
  ldi  r_lout,1           ; load output register with low value
  ldi  r_hout,2           ; load output register with high value

;****************************************
; Square Wave Generator
;****************************************
line1:
  dec  r_cnt1             ; decrement counter1 register
  breq line5              ; branch if result of decrement equals zero
  nop                     ; no operation (waste one instruction time)
  jmp  line7              ; jump to line7
line5:
  ldi  r_cnt1,first_note  ; load counter1 register with first_note value
  eor  r_vol1,r_vol       ; flip state of first_note intensity (0 or volume)

line7:
  dec  r_cnt2             ; decrement counter2 register
  breq line11             ; branch to line 11 if result of decrement equals zero
  nop                     ; no operation (waste one instruction time)
  jmp  line13             ; jump to line13
line11:
  ldi  r_cnt2,second_note ; load counter1 register with second_note value
  eor  r_vol2,r_vol       ; flip state of second_note intensity (0 or volume)

;****************************************
; NCO Summing
;****************************************
line13:
  add  r_acc,r_vol1       ; add first_note intensity to accumulator register
  brcs  line20            ; branch if result of add overflows
  add  r_acc,r_vol2       ; add second_note intensity to accumulator register
  brcs  line22            ; branch if result of add overflows
  nop                     ; no operation (waste one instruction time)
  out  portx,r_lout       ; output a low level to enunciator
  jmp  line1              ; jump to line1
line20:
  nop                     ; no operation (waste one instruction time)
  adc  r_acc,r_vol2       ; add second_note int. and overflow bit to accumulator
line22:
  out  portx,r_hout       ; output a high level to enunciator
  jmp  line1              ; jump to line1
```

Instrumentation amp provides unipolar and bipolar outputs

David Rathgeber, Alles Corp, Toronto, ON, Canada

Intelligent sensors and signal conditioners for plant-automation systems must produce outputs compatible with standard data-acquisition systems, such as PCs, programmable logic controllers, and remote terminals for supervisory and control systems. Standard analog inputs are 0 to 1V, ±10V, 0 to 1 mA, and 4 to 20 mA. The circuit in **Figure 1** makes all these signals available, using only three packages in addition to the DAC. The voltage-output section requires a thin-film resistor package, Beckman Series 668/698, which tracks with temperature, and two FET-input op amps. The current-output section requires the addition of a four-resistor package, Beckman Series 664/694, and two more op amps. A reference voltage of −1.000V yields voltage outputs of 0 to 1V and ±1V and either 0 to 1 mA with R_X=1 kΩ and JP_2 and JP_4 connected or 4 to 20 mA with R_X=83.33Ω and JP_1 and JP_3 connected. A reference of −10.00V and appropriate values of R_X yield corresponding voltage and current outputs. All R_1 and R_2 values in **Figure 1** are 100 kΩ±0.1%. You should select Q_1, and possibly use a heat sink, taking into consideration the supply voltage and output-current range. (DI #2560)

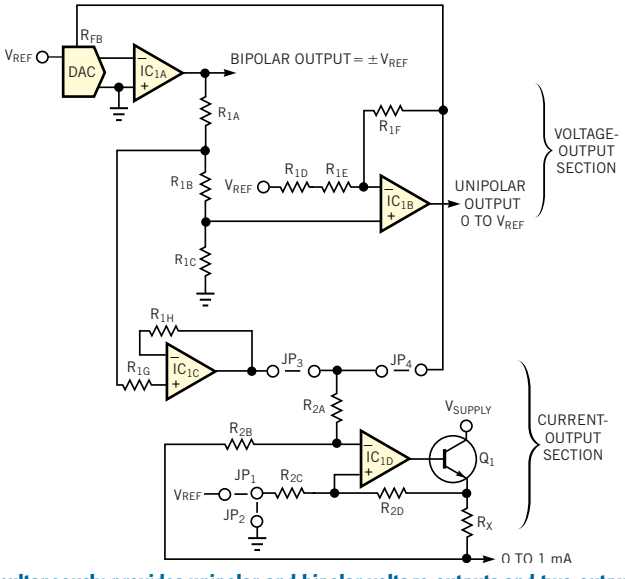


Figure 1

This circuit simultaneously provides unipolar and bipolar voltage outputs and two output-current ranges.

To Vote For This Design, Enter No. 334

design ideas

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02458

I hereby submit my Design Ideas entry. (Please print clearly)

Name _____

Title _____

Phone _____

E-mail _____ **Fax** _____

Company _____

_____

Address _____

_____

Country _____ **ZIP** _____

Design Idea Title _____

_____

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to *EDN*, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit text, software listings and all other computer-readable documentation on IBM PC disk or in plain ASCII by e-mail to b.travis@cahners.com.

Exclusive publishing rights remain with Cahners Business Information unless entry is returned to author, or editor gives written permission for publication elsewhere. The author must be willing to sign and return our publication agreement if the Design Idea is accepted for publication and must complete a W-9 tax form (W-8 for non-US residents) before payment can be processed.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by entering the appropriate number at www.ednmag.com/infoaccess.asp.