

Lowpass, 30-kHz Bessel filter offers high performance for audio applications

Troy Murphy, Analog Devices, San Jose, CA

Thanks to its property of applying an equal amount of delay to all frequencies below its cutoff frequency, the Bessel linear-phase filter sees service in audio applications in which it's necessary to remove out-of-band noise without degrading the phase relationships of a multifrequency in-band signal. In addition, the Bessel filter's fast step response and freedom from overshoot or ringing make it an excellent choice as a smoothing filter for an audio DAC's output or as an antialiasing filter for an audio ADC's input. Bessel filters are also useful for analyzing the outputs of Class D amplifiers and for eliminating switching noise in other applications to improve accuracy of distortion and oscilloscope-waveform measurements.

Although the Bessel filter provides flat magnitude and linear-phase—that

is, uniform group-delay—responses within its passband, it has worse selectivity than Butterworth or Chebyshev filters of the same order, or number of poles. Thus, to achieve a given level of stopband attenuation, you need to design a higher order Bessel filter, which, in turn, requires careful selection of amplifiers and components to achieve the lowest levels of noise and distortion.

Figure 1 shows a schematic for a high-performance, eighth-order, 30-kHz, lowpass Bessel filter. This design uses standard values for 1%-tolerance resistors and 5%-tolerance ceramic capacitors. As an alternative, you can use 10%-tolerance capacitors at the expense of increased group-delay variance within the passband. For best results, use temperature-stable capacitors.

In this application, the filter process-

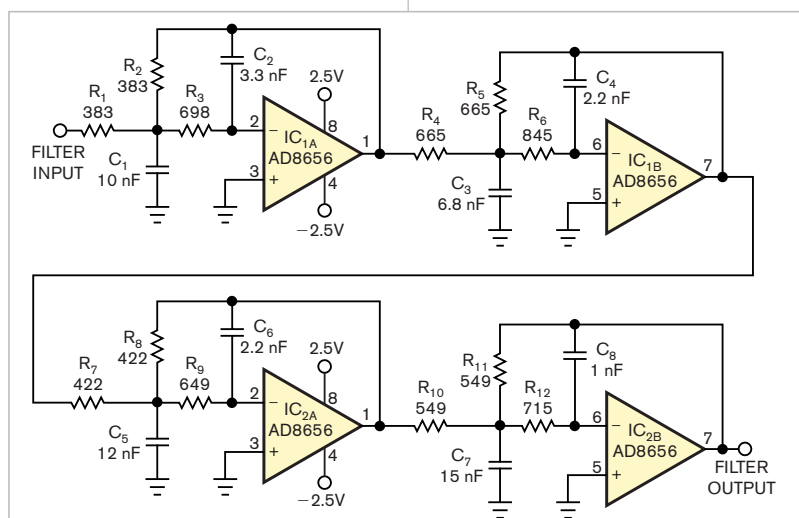


Figure 1 Two dual op amps and a handful of passive parts implement a high-performance, eighth-order, 30-kHz, lowpass Bessel filter.

DIs Inside

84 Use a PWM fan controller in an EMI-susceptible circuit

88 PC's parallel port and a PLD host multiple stepper motors and switches

► What are your design problems and solutions? Publish them here and receive \$150! Send your Design Ideas to edndesignideas@reedbusiness.com.

es audio signals that swing above and below ground, and its amplifiers draw power from positive and negative $\pm 2.5V$ supplies. Rail-to-rail output capability helps achieve maximum output-voltage swing at these low supply voltages. To achieve a high SNR in high-quality audio service, the amplifiers must exhibit unity-gain stability and low inherent noise. For example, Analog Devices' AD8656 low-noise, precision-CMOS dual op amp meets all of these requirements.

Connecting the amplifiers as inverting-gain stages maintains constant input-common-mode voltage and helps minimize distortion. Using less-than-1-k Ω resistors throughout the circuit reduces the resistors' thermal-noise contributions. Each AD8656 amplifier contributes less than 3 nV/ \sqrt{Hz} of noise across a 30-kHz bandwidth, and the total noise over a 30-kHz bandwidth measures less than 3.5 μV rms. For a 1V-rms input signal, the circuit yields an SNR of better than 109 dB, and, for a 1-kHz, 1V-rms input signal, the circuit yields a THD+N (total-harmonic-distortion-plus-noise) factor of better than 0.0006%.

Figure 2 shows the filter's measured magnitude response for a 1V-rms input signal. The filter's passband gain of 0 dB is flat within 1.2 dB for frequencies as

high as 20 kHz. With its -3 -dB point at 30 kHz, an eighth-order Bessel presents a theoretical attenuation of -110 dB at 300 kHz, decreasing at -160 dB/decade at higher frequencies. This characteristic provides sufficient attenuation of repetitive noise that switched-mode power supplies and other sources induce, which typically occurs at frequencies of 300 kHz and higher.

Figure 3 illustrates the filter's phase shift and its group delay, which remains relatively constant at roughly $17 \mu\text{sec}$, even for frequencies as high as 40 kHz. Note the linear scale on **Figure 3**'s frequency axis, which clearly illustrates the filter's linear-phase behavior within the passband. The following equation defines group delay as the negative partial derivative of phase shift with respect to frequency:

$$\text{Group delay} = -\delta\phi/\delta f.$$

At dc, resistor R_1 sets the filter's input resistance at 383Ω . If your application requires higher input impedance, you can insert a unity-gain buffer ahead of the filter at the expense of increased distortion and noise. For applications that require operation from $\pm 15\text{V}$ power supplies, replace the AD8656 with a higher voltage amplifier, such as Analog Devices' AD8672 low-distortion, low-noise ($3.8\text{-nV}/\sqrt{\text{Hz}}$), dual operational amplifier. **EDN**

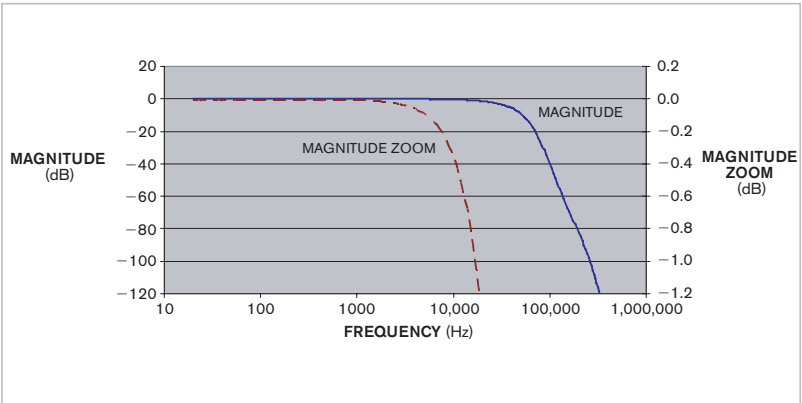


Figure 2 The measured amplitude-versus-frequency response of the circuit in Figure 1 shows a change of scale on the right vertical axis.

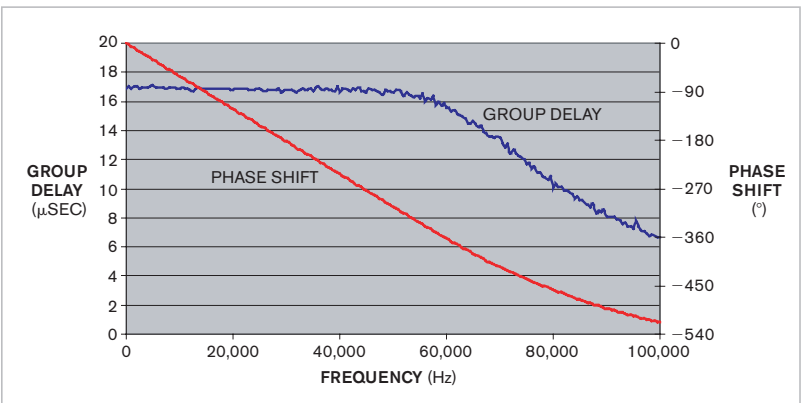



Figure 3 Measured within the passband of dc to 30 kHz, the Bessel filter's phase-shift and group-delay characteristics display excellent uniformity and linearity.

Use a PWM fan controller in an EMI-susceptible circuit

Dimitri Danyuk, Niles Audio Corp

 Microchip Technology (www.microchip.com) offers a family of cooling-fan speed controllers that operate in PWM mode for use with brushless dc fans (**Reference 1**). To control fan speed using the PWM waveform's duty cycle, you can use either an external NTC (negative-temperature-coefficient) thermistor or one of Microchip's PIC microcontrollers and its SMBus

serial-data bus. **Figure 1** illustrates a typical application that the data sheet describes for the TC664 and TC665 controllers (**Reference 2**). Using a frequency-control capacitor, C_F , with a value of $1 \mu\text{F}$, fan-controller IC_1 generates a PWM pulse train with a nominal frequency of 30 Hz and a temperature- or command-dependent duty cycle that varies from 30 to 100%.

Although using the controller in PWM mode reduces power dissipation in transistor Q_A , which drives the fan, the 100-mA, square-wave motor-drive current can cause unwanted interference in a nearby high-sensitivity audio circuit. The circuit in **Figure 2** solves the problem. An additional driver transistor, Q_1 , and an RC network comprising C_3 and R_3 form a simple PWM-to-linear converter. You can also use another PWM-to-linear-conversion circuit, such as an integrator based on an operational amplifier.

Figure 3 shows a graph of the dc voltage at Q_2 's collector versus IC_1 's PWM

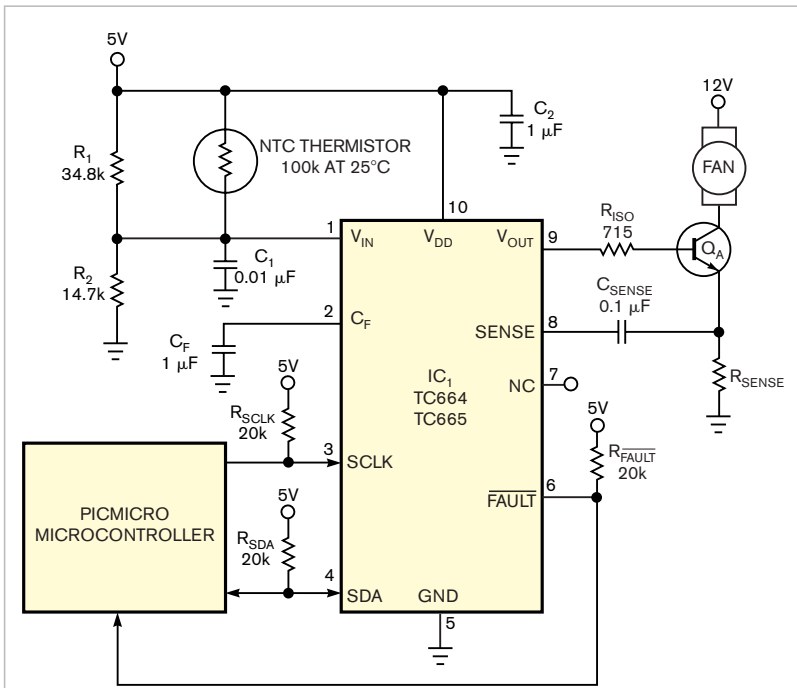


Figure 1 In a typical application, fan-controller IC₁ and transistor Q_A apply pulse-width-modulated current to vary a fan's speed as a function of temperature.

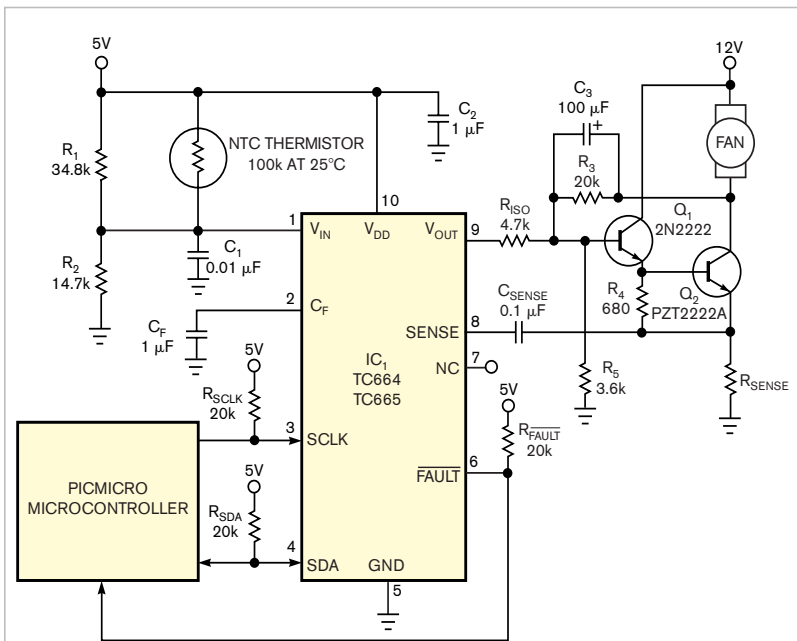


Figure 2 To minimize the effects of high-frequency noise on sensitive analog circuits, you can convert the high-current PWM waveform applied to the fan to a continuous analog voltage.

drive-output waveform's duty cycle. The voltage applied to the fan corresponds to the difference between Q₂'s collector voltage and the 12V supply voltage. Even though a steady voltage appears across the fan, current pulses that the fan motor's commutation produces still develop a voltage across current-sense resistor R_{SENSE} that connect to Q₂'s emitter, and all of IC₁'s protective and advisory features remain available.

The listed component values are valid for a 100-mA, 12V, brushless fan. Use a general-purpose NPN transistor such as the 2N2222 for driver-transistor Q₁ and an NPN transistor, such as Fairchild Semiconductor's PZT2222A, that can dissipate one-third of the fan's maximum power consumption for Q₂. Note that you can vary the PWM's nominal frequency over a range of 15 to 35 Hz by altering the value of C_F. **EDN**

REFERENCES

- 1 "Fan Speed Controller and Fan Fault Detector Family," Microchip Technology Inc, 2002, <http://ww1.microchip.com/downloads/en/DeviceDoc/21604c.pdf>.
- 2 SMBus PWM Fan Speed Controllers with Fan Fault Detection, Microchip Technology Inc, 2003, <http://ww1.microchip.com/downloads/en/DeviceDoc/21737a.pdf>.

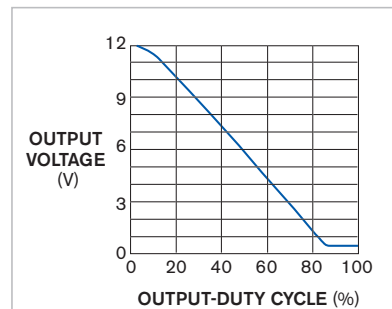


Figure 3 Output voltage at Q₂'s collector shows a linear relationship versus the controller's pulse-width-modulated output-duty cycle. (The pulse width increases as the temperature increases.) The fan's operating voltage corresponds to the difference between Q₂'s output voltage and the 12V supply rail.

PC's parallel port and a PLD host multiple stepper motors and switches

Eduardo Pérez-Lobato, Universidad de Antofagasta, Antofagasta, Chile

Robotic applications frequently include multiple stepper motors and switches. The stepper motors produce motion in several directions, and the switches identify home positions and detect proximity to obstacles. This Design Idea describes the development and implementation of a PLD (programmable-logic-device)-based interface that can connect a PC's parallel port to as many as eight switches and four stepper motors (Figure 1). This interface design serves many applications, and using IC₁, a 22V10 PLD, to minimize the circuit's component

count reduces complexity, weight, and overall dimensions. Drivers IC₃ through IC₆ for the stepper motors comprise three L293 quad half-H-bridge ICs (Figure 2).

Each rotation of the two-winding stepper motor in this design requires a sequence of four mechanical steps that you produce by applying a pair of 7V, 500-mA, 120-msec-long pulses to the motor's windings (Figure 3). To make a stepper motor rotate either CW (clockwise) or CCW (counterclockwise), you apply either of two pulse sequences (tables 1 and 2).

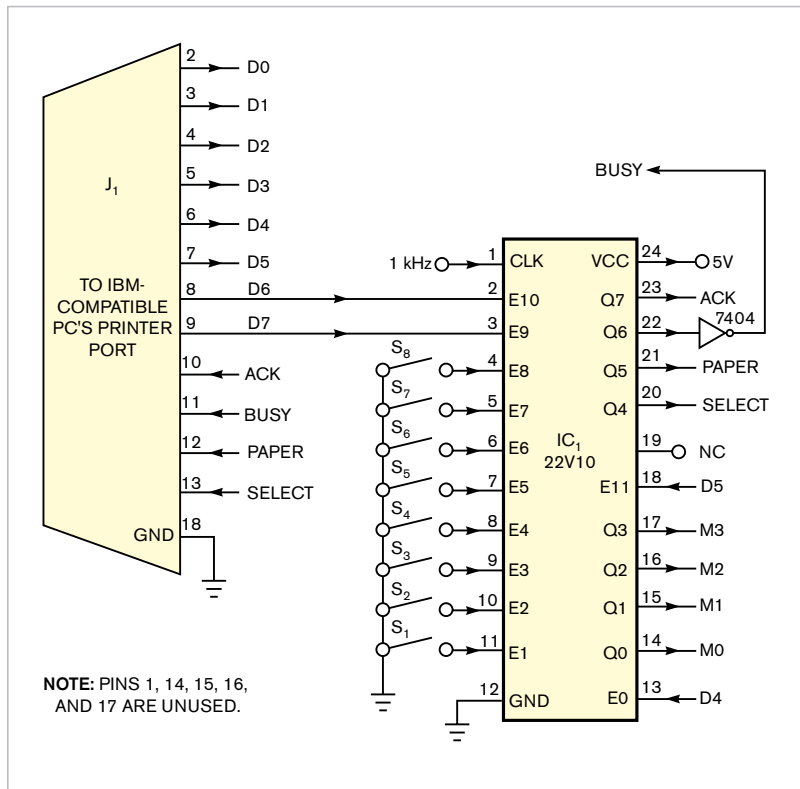


Figure 1 A programmable-logic device, IC₁, and a few additional components allow an IBM-compatible PC's parallel printer port to drive as many as four external stepper motors and to sense the states of as many as eight range-of-motion limit switches.

The following sections specify the functions of the input and output registers' bits that control the parallel-port interface and the PLD. The PLD output-register bits are 7, 6, 5, 4, 3, 2, and 1. Q7 signals the PC that one or more switches are active. Bit 0 means that a switch is active; bit 1 means that no switches are active. With Q6, Q5, and Q4, the BSS (buffered-status switch) tells the PC which of n switches is active: 000=S₁, 100=S₃, 001=S₂, 101=S₆, 010=S₃, 110=S₇, 011=S₄, and 111=S₈. For Q3, Q2, Q1, and Q0, the PLD's outputs enable one of the four motor-driver ICs to drive its associated stepper motor, with 1000=M₃, 0010=M₁, 0100=M₂, and 0001=M₀.

The PLD input register's bits are E11, E10, E9, and E0. For E11, the host PC controls the PLD, 0 disables the PLD, and 1 enables the PLD. For E10 and E9, the PLD reads these lines to determine which of the four motors in Figure 2 receives drive pulses: 00 for Motor 0, 10 for Motor 2, 01 for Motor 1, and 11 for Motor 3. For bit E0, the PLD reads this bit to determine what to do with the BSS settings: 0=hold, and 1=clear. For E8 through E1, the PLD reads the status of one switch and stores it in the BSS register:

- 00000001=S₁, 00010000=S₅,
- 00000010=S₂, 00100000=S₆,
- 00000100=S₃, 01000000=S₇,
- 00001000=S₄, 10000000=S₈.

The PLD ignores any unlisted bits.

For the parallel-port output register, address 888₁₀, D7, and D6, the PC tells

TABLE 1 CLOCKWISE-ROTATION SEQUENCE

Step	A	B	C	D
0	1	1	0	0
1	0	1	1	0
2	0	0	1	1
3	1	0	0	1

TABLE 2 COUNTERCLOCKWISE-ROTATION SEQUENCE

Step	A	B	C	D
0	1	0	0	1
1	0	0	1	1
2	0	1	1	0
3	1	1	0	0

the PLD which motor should run, with 00 for Motor 0, 10 for Motor 2, 01 for Motor 1, and 11 for Motor 3. For D5, the PC takes control of the PLD chip: 0 disables the PLD, and 1 enables the PLD. For D4, the PC commands the PLD to control the BSS register's contents, with 0 for hold and 1 for clear. For D3 through D0, the PC selects which pair of motor windings get energized: 1001=A and D, 1100=C and D, 0011=A and B, and 0110=C and B. Parallel-port input-register, address $888_{10} + 1$ indicates acknowledge, busy, paper, or select. The PC reads acknowledge to determine whether a switch is active: 0 means that any switch is active, and 1 means that no switch is

active. The PC reads the busy, paper, or select register to determine which of the switches is active:

- 000=S₁, 011=S₄,
- 110=S₇, 001=S₂,
- 100=S₅, 111=S₈,
- 010=S₃, 101=S₆.

You can download **Listing 1** for this Design Idea from www.edn.com/060216di3. Note that the PC's portion of the software is written in Pascal, and the PLD's internal software is written in an emulated version of Basic.**EDN**

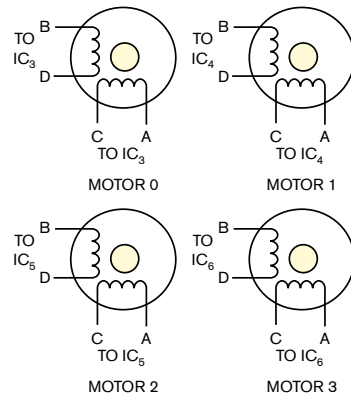


Figure 3 To control a stepper motor's direction of rotation, energize the windings as shown in tables 1 and 2.

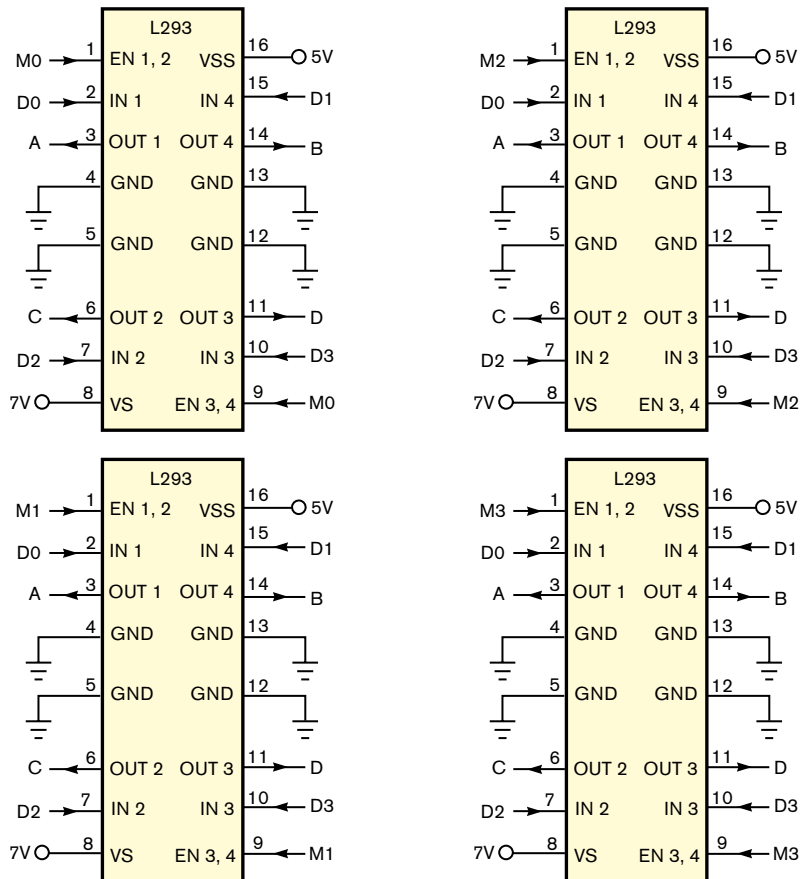


Figure 2 Each half-bridge-driver circuit, IC₃ through IC₆, controls a single two-winding stepper motor.