Precision analog bests digital in speed, noise, simplicity, and ease of implementation

Paul Antonucci, Alberti's Window, Watertown, MA

Every once in a while, I read that analog is on the way out, and everything should be digital. Recently, I was involved in a design project that illustrates that this belief doesn't apply in many situations.

The problem was to put two new optical breaks into the drive mechanism of a group of robotically controlled, Internet-accessible telescopes in an education application. The drive mechanisms have been showing signs of wear from excess slipping at end of travel: The sensors would signal end of travel, so there would be no slipping. The fork arms and other locations enclosed the internal wiring harnesses of the telescopes so that rewiring the telescopes would have been awkward.

So, the best idea was to encode the signals from the new sensors into the current wiring. There was one digital signal available; the challenge was to encode the two new sensors onto that signal.

Using a digital approach would have involved adding a small microcontroller to the base and encoding a serial digital signal to send up the tube, with appropriate synchronous pulses, data, and check sums, which then would undergo decoding at the CPU. This approach would have required some sort of reset provision because the telescope needed to operate independently for months, and those serial digital signals would have undesirable switching noise on them. The CPU would also have had to spend time grabbing, decoding, and synchronizing the signals, taking up more time. In addition, we would have had to have written some messy bit-banging code: not a huge challenge—but not a simple or elegant one, either.

Instead, this approach uses a variation on a simple adder circuit in which each sensor contributes a different amount. Taking the basic binary idea that one sensor adds ±1; the next, ±2; and the last, ±4, the approach uniquely represents each state.

The basic requirements are a voltage reference, some op amps, and a summing junction. This application uses IC_4, a Texas Instruments (www.ti.com) REF3040 voltage reference, which has an output tolerance of 0.2% yet costs only approximately $1 (**Figure 1**). This reference generates a voltage of 4.096V and produces enough current to run the op amps, which run rail to rail to within a few millivolts. Be careful, however: Some "rail-to-rail" op amps have insufficient current drive near the rails. This circuit uses 0.1%-precision resistors, which cost only about 20 cents. Remember that you can use two 10-kΩ resistors in series and two in parallel to create the 20- and 5-kΩ resistances that you see in the **figure**. The assembly and bill of materials are simpler and precision is better because the distributions around the ideal resistor value tend to cancel out. **Table 1** lists the predicted output voltages.

TABLE 1 PREDICTED OUTPUT VOLTAGES

Sensor 1	Sensor 2	Sensor 3	Output
0	0	0	0.256
0	0	1	0.768
0	1	0	1.28
0	1	1	1.792
1	0	0	2.304
1	0	1	2.816
1	1	0	3.328
1	1	1	3.84

Tests with a voltmeter show that all output voltages were within 1 mV of the predicted output values. The error budget of less than 1% shows that you could use this method to encode several more sensors. In the telescope, the CPU's ADC reads the outputs. Read it twice to ensure that you aren't catching it at a transi-

DIs Inside

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

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

tion. The advantages of the circuit include the fact that its dc signals ensure that there's no noise and that the updates are nearly instantaneous. Also, because op amps are simple, virtually indestructible, and insensitive to noise, no reset circuits are necessary. Best of all, the design requires no programming. EDN

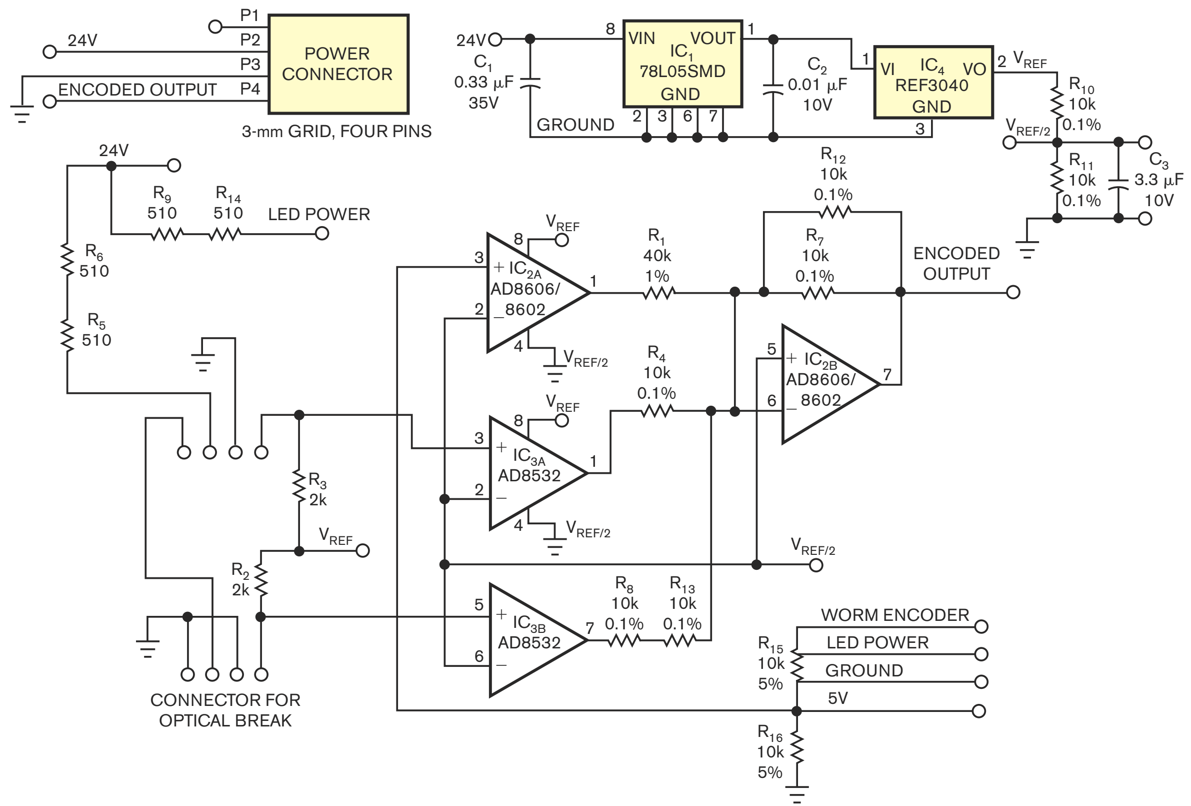

Figure 1 This application uses IC_4, a REF3040 voltage reference, which has an output tolerance of 0.2% yet costs only approximately $1.