Volume 43, Number 3, 2009

A forum for the exchange of circuits, systems, and software for real-world signal processing

Analog Dialogue

In This Issue

www.analog.com/analogdialogue

ANALOG DEVICES

Editors' Notes

IN THIS ISSUE

Detecting Human Falls with a 3-Axis Digital Accelerometer

Elderly individuals can suffer accidental falls due to weakness or dizziness. The initial injury can be further aggravated if treatment is not obtained within a short time. MEMS-based fall detectors can sense changes in body position by tracking acceleration, determine when an individual is falling, and issue an alert for assistance. This article describes the use of an ADXL345 three-axis digital accelerometer as a fall detector. Page 3.

Measuring Noise of Low-Fixed-Gain Differential Amplifiers

Measuring noise of low-gain differential amplifiers presents a challenge, as their integrated feedback and gain resistors preclude high-gain configurations, and a differential-to-single-ended conversion is needed to match the spectrum analyzer. A second amplifier stage can provide gain and the differential-to-SE conversion, neatly solving both of these problems. Page 10.

The Basics of Video Decoders in Supervision and Inspection

Video cameras—which range from inexpensive, low-definition black-and-white closed-circuit television systems to state-of-the-art high-definition digital-video systems—are found in diverse applications including product inspection, traffic monitoring, and real-time face recognition. Dropping unneeded data and passing only the essential parts of the picture can simplify the video system—saving memory and computational cycles. Page 11.

How to Apply Low-Dropout Regulators Successfully

A low-dropout regulator (LDO) is capable of maintaining a specified output voltage over a wide range of load current and input voltage, down to a very small difference between input and output voltages. This difference, known as the dropout voltage or headroom requirement, can be as low as 80 mV at 2 A. Page 14.

Phase Response in Active Filters, Part 2: Low-Pass and High-Pass Responses

The transfer function of an active filter can be viewed as the cascaded response of the filter transfer function and an amplifier transfer function. This article examines the phase shift of the filter transfer function itself. While filters are designed primarily for their amplitude response, the phase response can be important in applications such as time delay simulation, cascaded filter stages, and especially process-control loops. Page 18.

Difference Amplifier Forms Heart of Precision Current Source

Precision current sources are used to provide excitation for RTDs in process-control systems; to measure unknown elements in digital multimeters; and to drive 4-mA to 20-mA current loops, which are widely used to transmit information over long distances. This article shows how a difference amplifier can be used to implement a precision current source. Page 22.

Dan Sheingold [dan.sheingold@analog.com]

Scott Wayne [scott.wayne@analog.com]

PRODUCT INTRODUCTIONS: VOLUME 43, NUMBER 3

Data sheets for all ADI products can be found by entering the part number in the search box at www.analog.com.

July

Accelerometer, digital, 3-axis, ±2-*g* /±4-*g* /±8-*g* /±16-*g*	**ADXL345**
Amplifier, instrumentation, rail-to-rail output	**AD8227**
Driver, flash LED, dual, high power	**ADP1655**
Driver, line, quad LVDS	**ADN4665**
Drivers, MOSFET, dual, 4-A, high-speed	**ADP3624/ADP3634**
Isolators, digital, 4-channel, 500-mW dc-to-dc	**ADuM640x**
Supervisor, voltage, 4-channel, OV/UV	**ADM2914**
Supervisor, voltage, 4-channel, ±supplies	**ADM6339**

August

Buffer, clock fanout, 6 LVDS/12 CMOS outputs	**ADCLK846**
Converter, current-to-digital, 128-channel, 24-bit	**ADAS1128**
Gain Block, RF/IF, 20-dB gain, 50-MHz to 4-GHz	**ADL5602**
Isolator, digital, dual, enhanced ESD protection	**ADuM3211**
Microcontrollers, precision analog, 24-bit ADCs	**ADuC7060/ADuC7061**
Mixer, balanced, 1200-MHz to 2500-MHz	**ADL5355**
Receiver, line, quad LVDS	**ADN4666**

September

Accelerometers, 3-axis, low-power, low-profile	**ADXL32x**
ADC, sigma-delta, 4-channel, 24-bit, 4.8-kHz	**AD7193**
ADCs, pipelined, dual, 10-/12-/14-bit, 80 MSPS	**AD9204/AD9231/AD9251**
Amplifier, difference, dual, unity-gain	**AD8277**
Amplifier, difference, gain of ½ or 2	**AD8278**
Amplifier, operational, dual, micropower	**ADA4051-2**
Amplifier, operational, dual, rail-to-rail	**ADA4691-2**
Amplifier, operational, dual, rail-to-rail	**AD8622**
Amplifier, operational, JFET-input	**ADA4627-1**
Amplifier, operational, quad, JFET-input	**ADA4062-4**
Amplifier, operational, quad, rail-to-rail	**ADA4091-4**
Amplifier, operational, zero crossover distortion	**ADA4505-1**
Buffer, clock fanout, 2-input, 8-output	**ADCLK948**
Buffer, clock fanout, 2-input, 10-output	**ADCLK950**
Converter, dc-to-dc, step-down, 3-A	**ADP2118**
Converter, dc-to-dc, step-down, dual, 2-A/4-A	**ADP2114**
DAC, voltage-output, quad, 16-bit	**AD5066**
DACs, voltage-output, single/quad, 12-bit	**AD5501/AD5504**
Delay Line, video, triple skew-compensating	**AD8120**
Demodulator, quadrature, 400-MHz to 6-GHz	**ADL5380**
Digitizer, video/graphics, 12-bit, 170-MHz	**ADV7840**
Driver, laser-diode, dual-loop, 50-Mbps to 3.3-Gbps	**ADN2872**
Drivers, current/voltage, programmable	**AD5750/AD5750-1**
Drivers, MOSFET, dual 4-A	**ADP362x/ADP363x**
Front Ends, analog, 8-channel	**AD9276/AD9277**
Generator, clock, 2-output, PCI Express®	**AD9573**
Generator, clock, 7-output, Fibre Channel/Ethernet	**AD9572**
Generator/Synchronizer, network clock, 4-input	**AD9547**
Isolator, digital, bidirectional, USB	**ADuM4160**
Mixer, balanced, 500-MHz to 1700-MHz	**ADL5357**
Modulator, sigma-delta, dual, 640 MSPS	**AD9267**
Multiplexer, CMOS, 8-channel, 4.5-Ω	**ADG1608**
Multiplexer, CMOS, differential, 4-channel, 4.5-Ω	**ADG1609**
Multiplexer/Demultiplexer, 6.5-Gbps, dual, 2:1	**AD8155**
Receiver, HDMI, quad, Deep Color	**ADV7614**
Switch, CMOS, quad SPDT, 4.5-Ω	**ADG1634**
Switch, CMOS, triple SPDT, 4.5-Ω	**ADG1633**
Transmitter, HDMI/DVI, CEC	**ADV7523**

Analog Dialogue

Analog Dialogue, www.analog.com/analogdialogue, the technical magazine of Analog Devices, discusses products, applications, technology, and techniques for analog, digital, and mixed-signal processing. Published continuously for 43 years—starting in 1967—it is currently available in two versions. Monthly editions offer technical articles; timely information including recent application notes, new-product briefs, pre-release products, webinars and tutorials, and published articles; and potpourri, a universe of links to important and relevant information on the Analog Devices website, www.analog.com. Printable quarterly issues feature collections of monthly articles. For history buffs, the *Analog Dialogue* archive includes all regular editions, starting with Volume 1, Number 1 (1967), and three special anniversary issues. If you wish to subscribe, please go to www.analog.com/analogdialogue/subscribe.html. Your comments are always welcome; please send messages to dialogue.editor@analog.com or to *Dan Sheingold*, Editor [dan.sheingold@analog.com] or *Scott Wayne*, Publisher and Managing Editor [scott.wayne@analog.com].

ISSN 0161-3626 ©Analog Devices, Inc. 2011

Detecting Human Falls with a 3-Axis Digital Accelerometer

By Ning Jia

Foreword

For a human, experiencing a fall unobserved can be doubly dangerous. The obvious possibility of initial injury may be further aggravated by the possible consequences if treatment is not obtained within a short time. For example, many elderly individuals can suffer accidental falls due to weakness or dizziness—or, in general, their diminished self-care and self-protective ability. Since they tend to be fragile, these accidents may possibly have serious consequences if aid is not given in time. Statistics show that the majority of serious consequences are not the direct result of falling, but rather are due to a delay in assistance and treatment. Post-fall consequences can be greatly reduced if relief personnel can be alerted in time.

Besides senior citizens, there are many other conditions and activities for which an immediate alert to a possible fall, especially from substantial height, would be quite helpful—for example mountaineers, construction workers, window washers, painters, and roofers.

In light of this need to warn of falls, the development of devices for detection and prediction of all types of falls has become a hot topic. In recent years, technological advances in *microelectromechanical-system* (MEMS) acceleration sensors have made it possible to design fall detectors based on a 3-axis *integrated MEMS* (*i*MEMS®) accelerometer. The technique is based on the principle of detecting changes in motion and body position of an individual, wearing a sensor, by tracking acceleration changes in three orthogonal directions. The data is continuously analyzed algorithmically to determine whether the individual's body is falling or not. If an individual falls, the device can employ GPS and a wireless transmitter to determine the location and issue an alert in order to get assistance. The core element of fall detection is an effective, reliable detection principle and algorithm to judge the existence of an emergency fall situation.

This article, based on research into the principles of fall detection for an individual body, proposes a new solution for detection of fall situations utilizing the ADXL345,[1] a 3-axis accelerometer from Analog Devices.

The ADXL345 *i*MEMS Accelerometer

*i*MEMS semiconductor technology combines micromechanical structures and electrical circuits on a single silicon chip. Using this technology, *i*MEMS accelerometers sense acceleration on one, two, or even three axes, and provide analog or digital outputs. Depending on the application, the accelerometer may offer different ranges of detection, from several *g* to tens of *g*. Digital versions may even have multiple *interrupt* modes. These features offer the user convenient and flexible solutions.

The recently introduced ADXL345 is an *i*MEMS 3-axis accelerometer with digital output. It features a selectable ±2-*g*, ±4-*g*, ±8-*g*, or ±16-*g* measurement range; resolution of up to 13 bits; fixed 4-m*g*/LSB sensitivity; a tiny 3-mm × 5-mm × 1-mm package; ultralow power consumption (25 µA to 130 µA); standard I^2C® and SPI serial digital interfacing; and 32-level FIFO storage. A variety of built-in features, including motion-status detection and flexible interrupts, greatly simplify implementation of the algorithm for fall detection. As you will see, this combination of features makes the ADXL345 an ideal accelerometer for fall-detector applications.

The fall-detection solution proposed here takes full advantage of these internal functions, minimizing the complexity of the algorithm—with little requirement to access the actual acceleration values or perform any other computations.

Interrupt System

Figure 1 shows the system block diagram and pin definitions of the ADXL345.

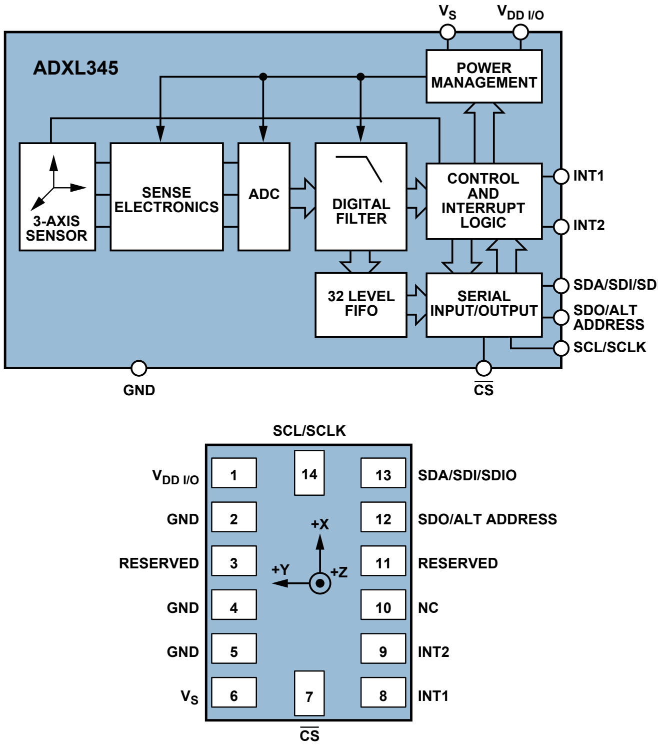


Figure 1. ADXL345 system block diagram and pin designations.

The ADXL345 features two programmable interrupt pins—INT1 and INT2—with a total of eight interrupt functions available. Each interrupt can be enabled or disabled independently, with the option to map to either the INT1 or INT2 pin. All functions can be used simultaneously—the only limiting feature is that some functions may need to share interrupt pins. The eight functions are: DATA_READY, SINGLE_TAP, DOUBLE_TAP, ACTIVITY, INACTIVITY, FREE_FALL, WATERMARK, and OVERRUN. Interrupts are enabled by setting the appropriate bit in the **INT_ENABLE** register and are mapped to either the INT1 or INT2 pins, based on the contents of the **INT_MAP** register. The interrupt functions are defined as follows:

1. DATA_READY is set when new data is available—and cleared when no new data is available.
2. SINGLE_TAP is set when a single acceleration event that is greater than the value in the THRESH_TAP register occurs for a shorter time than specified in the DUR register.
3. DOUBLE_TAP is set when two acceleration events that are greater than the value in the THRESH_TAP register occur and are shorter than the time specified in the DUR register, with the second tap starting after the time specified by the LATENT register and within the time specified in the WINDOW register.

Figure 2 illustrates the valid SINGLE_TAP and DOUBLE_TAP interrupts.

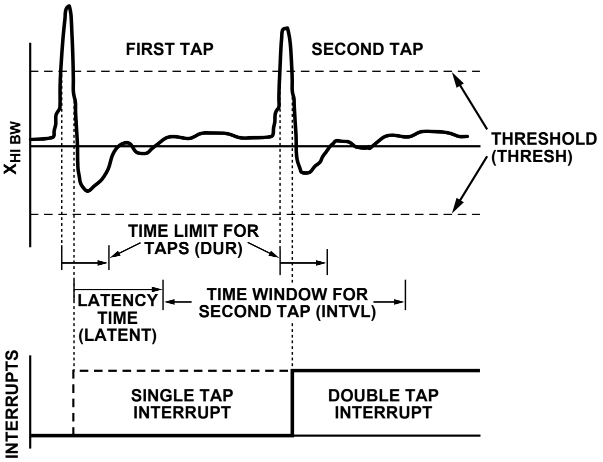


Figure 2. SINGLE_TAP and DOUBLE_TAP interrupts.

4. ACTIVITY is set when acceleration greater than the value stored in the THRESH_ACT register is experienced.
5. INACTIVITY is set when acceleration of less than the value stored in the THRESH_INACT register is experienced for longer than the time specified in the TIME_INACT register. The maximum value for TIME_INACT is 255 s.

   Note: With ACTIVITY and INACTIVITY interrupts, the user can enable or disable each axis individually. For example, the ACTIVITY interrupt for the X-axis can be enabled while disabling the interrupts for the Y-axis and Z-axis.

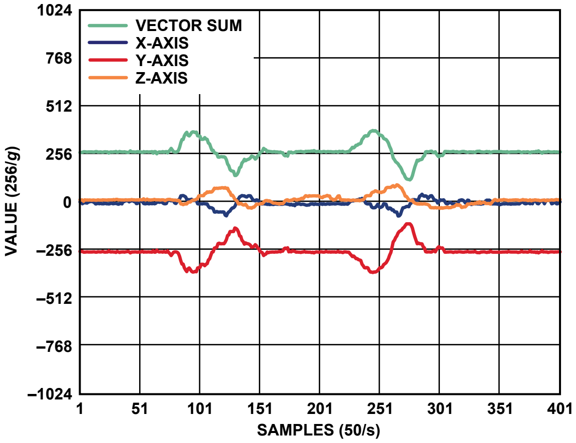
   Furthermore, the user can select between *dc*-coupled or *ac*-coupled operation mode for the ACTIVITY and INACTIVITY interrupts. In dc-coupled operation, the current acceleration is compared with THRESH_ACT and THRESH_INACT directly to determine whether ACTIVITY or INACTIVITY is detected. In ac-coupled operation for activity detection, the acceleration value at the start of activity detection is taken as a reference value. New samples of acceleration are then compared to this reference value; if the magnitude of the difference exceeds THRESH_ACT, the device will trigger an ACTIVITY interrupt. In ac-coupled operation for inactivity detection, a reference value is used for comparison and is updated whenever the device exceeds the inactivity threshold. Once the reference value is selected, the device compares the magnitude of the difference between the reference value and the current acceleration with THRESH_INACT. If the difference is below THRESH_INACT for a total of TIME_INACT, the device is considered inactive and the INACTIVITY interrupt is triggered.
6. FREE_FALL is set when acceleration of less than the value stored in the THRESH_FF register is experienced for longer than the time specified in the TIME_FF register. FREE_FALL interrupt is mainly used in detection of free-falling motion. As a result, the FREE_FALL interrupt differs from the INACTIVITY interrupt in that all axes always participate, the timer period is much shorter (1.28 s maximum), and it is always dc-coupled.
7. WATERMARK is set when the number of samples in the FIFO has filled up to the value stored in the SAMPLES register. It is cleared automatically when the FIFO is read and its content emptied below the value stored in the SAMPLES register.

   Note: the FIFO register in the ADXL345 has four operation modes: Bypass, FIFO, Stream, and Trigger; and can store up to 32 samples (X-, Y-, and Z-axis). The FIFO function is an important and very useful feature; *however, the proposed solution does not use the FIFO function*, so it will not be further discussed.
8. OVERRUN is set when new data has replaced unread data. The precise operation of OVERRUN depends on the operation mode of FIFO. In bypass mode, OVERRUN is set when new data replaces unread data in the DATAX, DATAY, and DATAZ registers. In all other modes, OVERRUN is set when the FIFO is filled with 32 samples. OVERRUN is cleared by reading the FIFO contents and is automatically cleared when the data is read.

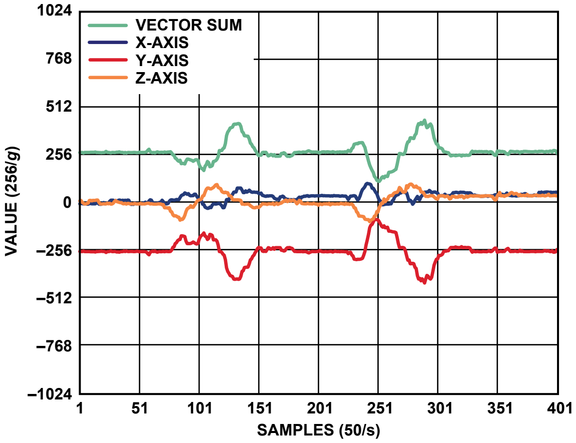
Acceleration-Change Characteristics While Falling

The main research on the principles of fall detection focuses on the changes in acceleration that occur when a human is falling.

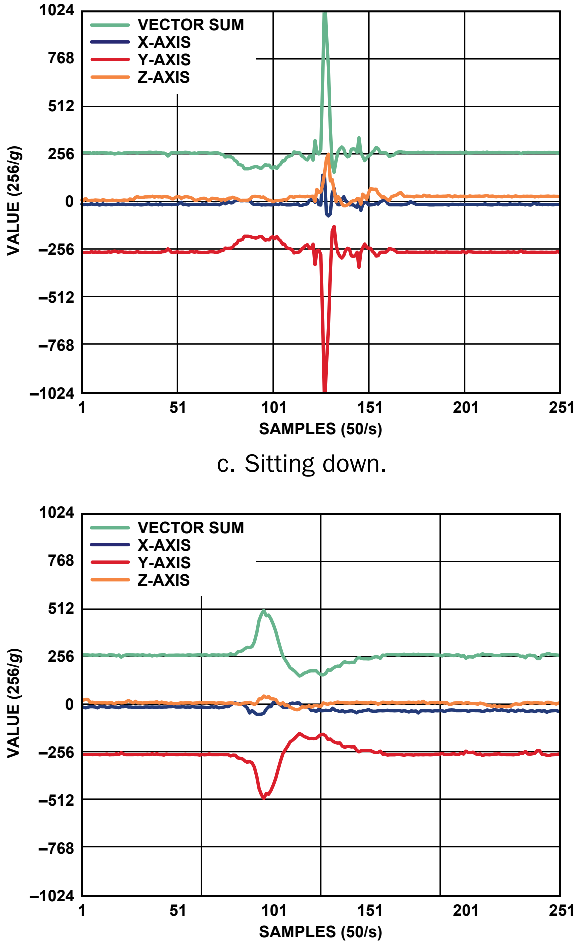
Figure 3 illustrates changes in acceleration that occur when (a) walking downstairs, (b) walking upstairs, (c) sitting down, and (d) standing up from a chair. The fall detector is mounted to a belt on the individual's body. The red trace is the Y-axis (vertical) acceleration; it is –1 *g* at equilibrium. The black and yellow traces are the respective X-axis (forward) and Z-axis (sideways) accelerations. They are both 0 *g* at equilibrium. The green trace is the vector sum magnitude, 1 *g* at equilibrium.



a. Walking downstairs.



b. Walking upstairs.



c. Sitting down.

d. Standing up.

Figure 3. Accelerometer responses to different types of motion.

Because the movement of elderly people is comparatively slow, the acceleration change will not be very conspicuous during the walking motions. The most pronounced acceleration is a 3-*g* spike in Y (and the vector sum) at the instant of sitting down.

The accelerations during falling are completely different. Figure 4 shows the acceleration changes during an accidental fall. By comparing Figure 4 with Figure 3, we can see four critical differences characteristic of a falling event that can serve as the criteria for fall detection. They are marked in the red boxes and explained in detail as follows:

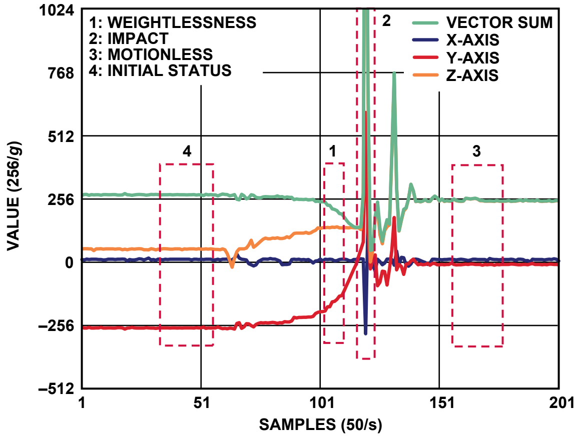


Figure 4. Acceleration change curves during the process of falling.

1. *Start of the fall*: The phenomenon of weightlessness will always occur at the start of a fall. It will become more significant during free fall, and the vector sum of acceleration will tend toward 0 *g*; the duration of that condition will depend on the height of freefall. Even though weightlessness during an ordinary fall is not as significant as that during a freefall, the vector sum of acceleration will still be substantially less than 1 *g* (while it is generally greater than 1 *g* under normal conditions). Therefore, this is the first basis for determining the fall status that could be detected by the ADXL345's FREE_FALL interrupt.
2. *Impact*: After experiencing weightlessness, the human body will impact the ground or other objects; the acceleration curve shows this as a large shock. This shock is detected by the ACTIVITY interrupt of ADXL345. Therefore, the second basis for determining a fall is the ACTIVITY interrupt right after the FREE_FALL interrupt.
3. *Aftermath*: Generally speaking, the human body, after falling and making impact, can not rise immediately; rather it remains in a motionless position for a short period (or longer as a possible sign of unconsciousness). On the acceleration curve, this presents as an interval of flat line, and is detected by the INACTIVITY interrupt of ADXL345. Therefore, the third basis for determining a fall situation is the INACTIVITY interrupt after the ACTIVITY interrupt.
4. *Comparing before and after*: After a fall, the individual's body will be in a different orientation than before, so the static acceleration in three axes will be different from the initial status before the fall (Figure 4). Suppose that the fall detector is belt-wired on the individual's body, to provide the entire history of acceleration, including the initial status. We can read the acceleration data in all three axes after the INACTIVITY interrupt and compare those sampling data with the initial status. In Figure 4, it is evident that the body fell on its side, since the static acceleration has changed from –1 *g* on the Y axis to +1 *g* on the Z-axis. So the fourth basis for determining a fall is if the difference between sampling data and initial status exceeds a certain threshold, for example, 0.7 *g*.

The combination of these qualifications forms the entire fall-detection algorithm, which, when exercised, can cause the system to raise an appropriate alert that a fall has occurred. Of course, the time interval between interrupts has to be within a reasonable range. Normally, the time interval between FREE_FALL interrupt (weightlessness) and ACTIVITY interrupt (impact) is not very long unless one is falling from the top of a very high building! Similarly, the time interval between ACTIVITY interrupt (impact) and INACTIVITY interrupt (essentially motionless) should not be very long. A practical example will be given in the next section with a set of reasonable values. The related interrupt detection threshold and time parameters can be flexibly set as needed.

If a fall results in serious consequences, such as unconsciousness, the human body will remain motionless for an even longer period of time, a status that can still be detected by the INACTIVITY interrupt, so a second critical alert could be sent out if the inactive state was detected to continue for a defined long period of time after a fall.

Typical Circuit Connection

The circuit connection between the ADXL345 and a microcontroller is very simple. For this article, the test platform uses the ADXL345 and an ADuC7026 analog microcontroller—which features 12-bit analog I/O and an ARM7TDMI® MCU. Figure 5 shows the typical connection between ADXL345 and ADuC7026.[2] With the $\overline{CS}$ pin of ADXL345 tied high, the ADXL345 works in I^{2}C mode. The SDA and SCL, the data and clock of the I^{2}C bus, are connected to the corresponding pins of ADuC7026. A GPIO of ADuC7026 is connected to the ADXL345's ALT pin to select the I^{2}C address of the ADXL345, and the INT1 pin of ADXL345 is connected to an IRQ input of the ADuC7026 to generate the interrupt signal.

Other MCU or processor types could be used to access the ADXL345, with similar circuit connections to Figure 5, but the ADuC7026 also provides a data-acquisition facility including multichannel analog-to-digital and digital-to-analog conversion. The ADXL345 data sheet describes SPI-mode applications to achieve higher data rates.

Using the ADXL345 to Simplify Fall Detection

Table 1, Figure 5, and the Appendix (Page 8) define the realization of the algorithm for the solution mentioned above. The function of each register is included in the table, and the values used in the present algorithm are as indicated. Please refer to the ADXL345 data sheet for the detailed definition of each register bit.

Some of the registers in Table 1 will have two values. This indicates that the algorithm switches between these values for different aspects of detection. Figure 6 is an algorithm flow chart.

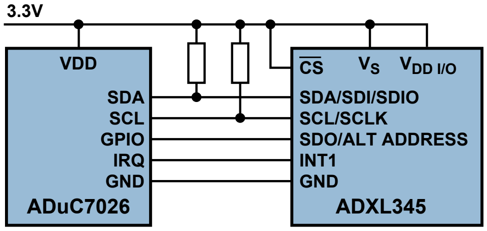


Figure 5. Typical circuit connection between the ADXL345 and microcontroller.

Table 1. ADXL345 Registers Function Descriptions

Hex Address	Register Name	Type	Reset Value	Description	Settings in Algorithm	Function of the Settings in Algorithm
0	DEVID	Read-only	0xE5	Device ID	Read-only	
1-1C	Reserved			Reserved, do not access	Reserved	
1D	THRESH_TAP	Read/write	0x00	Tap threshold	Not used	
1E	OFSX	Read/write	0x00	X-axis offset	0x06	X-axis, offset compensation get from initialization calibration
1F	OFSY	Read/write	0x00	Y-axis offset	0xF9	Y-axis offset compensation, get from initialization calibration
20	OFSZ	Read/write	0x00	Z-axis offset	0xFC	Z-axis offset compensation, get from initialization calibration
21	DUR	Read/write	0x00	Tap duration	Not used	
22	LATENT	Read/write	0x00	Tap latency	Not used	
23	WINDOW	Read/write	0x00	Tap window	Not used	
24	THRESH_ACT	Read/write	0x00	Activity threshold	0x20/0x08	Set activity threshold as 2 *g*/0.5 *g*
25	THRESH_INACT	Read/write	0x00	Inactivity threshold	0x03	Set inactivity threshold as 0.1875 *g*
26	TIME_INACT	Read/write	0x00	Inactivity time	0x02/0x0A	Set inactivity time as 2 s or 10 s
27	ACT_INACT_CTL	Read/write	0x00	Axis enable control for activity/inactivity	0x7F/0xFF	Enable activity and inactivity of X-, Y-, Z-axis, wherein inactivity is ac-coupled mode, activity is dc-coupled/ac-coupled mode
28	THRESH_FF	Read/write	0x00	Free-fall threshold	0x0C	Set free-fall threshold as 0.75 *g*
29	TIME_FF	Read/write	0x00	Free-fall time	0x06	Set free-fall time as 30 ms
2A	TAP_AXES	Read/write	0x00	Axis control for tap/double tap	Not used	
2B	ACT_TAP_STATUS	Read-only	0x00	Source of activity/tap	Read-only	
2C	BW_RATE	Read/write	0x0A	Data rate and power mode control	0x0A	Set sample rate as 100 Hz
2D	POWER_CTL	Read/write	0x00	Power save features control	0x00	Set as normal working mode
2E	INT_ENABLE	Read/write	0x00	Interrupt enable control	0x1C	Enable activity, inactivity, free-fall interrupts
2F	INT_MAP	Read/write	0x00	Interrupt mapping control	0x00	Map all interrupts to Int1 pin
30	INT_SOURCE	Read-only	0x00	Source of interrupts	Read-only	
31	DATA_FORMAT	Read/write	0x00	Data format control	0x0B	Set as ±16 *g* measurement range, 13-bit right alignment, high level interrupt trigger, I^{2}C interface
32	DATAX0	Read-only	0x00	X-axis data	Read-only	
33	DATAX1	Read-only	0x00		Read-only	
34	DATAY0	Read-only	0x00	Y-axis data	Read-only	
35	DATAY1	Read-only	0x00		Read-only	
36	DATAZ0	Read-only	0x00	Z-axis data	Read-only	
37	DATAZ1	Read-only	0x00		Read-only	
38	FIFO_CTL	Read/write	0x00	FIFO control	Not used	
39	FIFO_STATUS	Read/write	0x00	FIFO status	Not used	

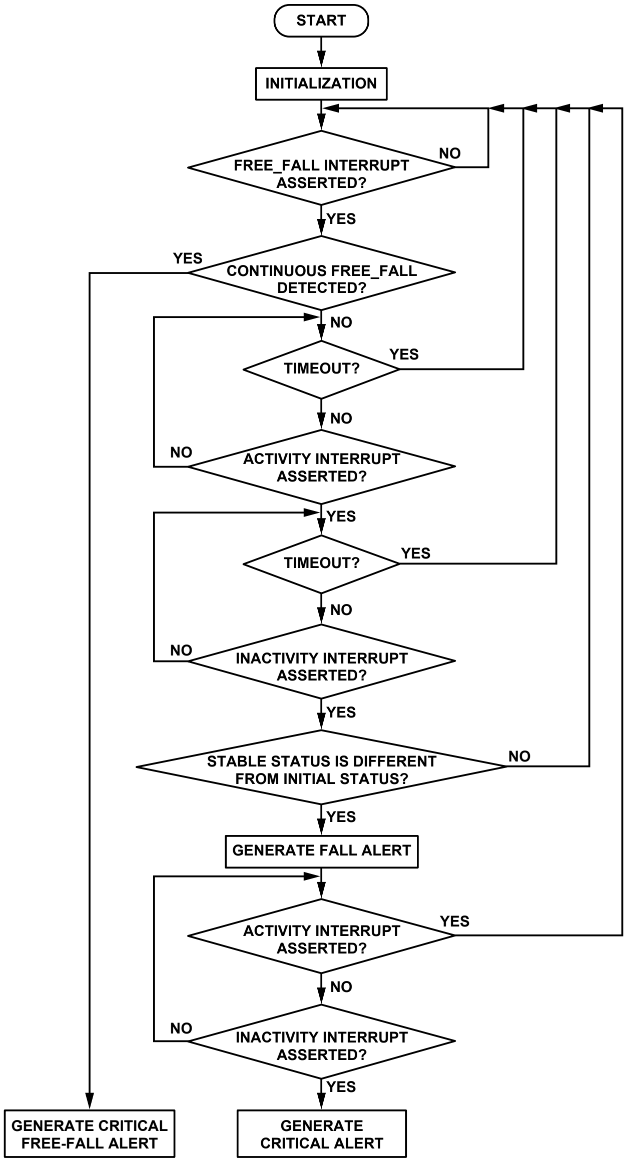

START
INITIALIZATION
FREE_FALL INTERRUPT ASSERTED?
NO
YES
CONTINUOUS FREE_FALL DETECTED?
YES
NO
TIMEOUT?
YES
NO
ACTIVITY INTERRUPT ASSERTED?
NO
YES
TIMEOUT?
YES
NO
INACTIVITY INTERRUPT ASSERTED?
NO
YES
STABLE STATUS IS DIFFERENT FROM INITIAL STATUS?
NO
YES
GENERATE FALL ALERT
ACTIVITY INTERRUPT ASSERTED?
YES
NO
INACTIVITY INTERRUPT ASSERTED?
NO
YES
GENERATE CRITICAL FREE-FALL ALERT
GENERATE CRITICAL ALERT

Figure 6. Algorithm flow chart.

Each interrupt threshold, and the related time parameter in the algorithm, is as described below.

1. After initialization, the system waits for the FREE_FALL interrupt (weightlessness). Here THRESH_FF is set to 0.75 *g* and TIME_FF is set to 30 ms.
2. After FREE_FALL interrupt is asserted, the system begins waiting for the ACTIVITY interrupt (impact). THRESH_ACT is set to 2 *g* and the ACTIVITY interrupt is in dc-coupled mode.
3. Time interval between FREE_FALL interrupt (weightlessness) and ACTIVITY interrupt (impact) is set to 200 ms. If the time between these two interrupts is greater than 200 ms, the status is not valid. The 200-ms counter is realized through the MCU timer.
4. After the ACTIVITY interrupt is asserted, the system begins waiting for the INACTIVITY interrupt (motionless after impact). THRESH_INACT is set to 0.1875 *g* and TIME_INACT is set to 2 s. INACTIVITY interrupt works in ac-coupled mode.
5. The INACTIVITY interrupt (motionless after impact) should be asserted within 3.5 s after the ACTIVITY interrupt (impact). Otherwise, the result is invalid. The 3.5-s counter is realized through the MCU timer.
6. If the acceleration difference between stable status and initial status exceeds the 0.7-*g* threshold, a valid fall is detected, and the system will raise a fall alert.
7. After detecting a fall, the ACTIVITY interrupt and INACTIVITY interrupt have to be continuously monitored to determine if there is a long period of motionlessness after the fall. The THRESH_ACT is set to 0.5 *g* and the ACTIVITY interrupt is running in the ac-coupled mode. THRESH_INACT is set to 0.1875 *g*, TIME_INACT is set to 10 s, and the INACTIVITY interrupt is working in ac-coupled mode. In other words, if the subject's body remains motionless for 10 s the INACTIVITY interrupt will be asserted and the system raises a critical alert. Once the subject's body moves, the ACTIVITY interrupt will be generated to complete the entire sequence.
8. The algorithm can also detect if the individual's body freefalls from a high place. Here, we consider that the two FREE_FALL interrupts are continuous if the interval between them is shorter than 100 ms. A critical freefall alert will be raised if the FREE_FALL interrupt (weightlessness) is continuously asserted for 300 ms

$$S = \frac{1}{2}gt^2 = \frac{1}{2}\times 10\times 0.3^2 = 0.45\text{ m}.$$

This algorithm is developed in C language to be executed on the ADuC7026 microcontroller (See Appendix, Page 8). A test case is also presented with the proposed solution to verify the algorithm. Each position, including falling forward, falling backward, falling to the left, and falling to the right is tested 10 times. Table 2 presents the test results. Check marks (✓) indicate each condition that is satisfied.

Table 2. Test Results

Falling Position	Prolonged Motionless Period after Falling	1	2	3	4	5	6	7	8	9	10
Falling Forward	No	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Yes	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*
Falling Backward	No	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Yes	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*
Falling to the Left	No	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Yes	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*
Falling to the Right	No	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Yes	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*	✓*

Note: The ✓ symbol indicates a detected fall; the * symbol indicates a detected prolonged motionless period after falling.

This experiment shows that falling status can be effectively detected with the proposed solution, based on the ADXL345. This is only a simple experiment. More comprehensive, effective, and long-term experimentation will be required to verify the reliability of the proposed solution.

Conclusion

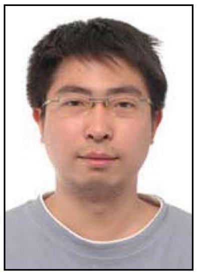
The ADXL345 is a powerful and full-featured accelerometer. We have described a proposed new solution for the fall-detection problem that takes advantage of the various built-in motion-status detection features and flexible interrupts. Tests have shown that it combines low algorithm complexity and high detection accuracy.

Appendix—An Example of the Code

This section presents an example of C code for the proposed solution, based on the ADXL345 and ADuC7026 platform. There are four .h files and one .c file in the project, compiled by Keil UV3. The .c file code is listed below.

Author

Ning Jia [ning.jia@analog.com], a field applications engineer, has been a member of the Analog Devices China Applications Support Team for two years. He is responsible for the technical support of a broad range of analog products across China. He graduated in 2007 from Beijing University of Posts and Telecommunications with a master’s degree in signal and information processing.



References

[1] Information on all ADI components can be found at www.analog.com.

[2] www.analog.com/en/analog-microcontrollers/ADuC7026/products/product.html.

```
// Include header files
#include “FallDetection.h”

void IRQ_Handler() __irq            // IRQ interrupt
{
   unsigned char i;

   if((IRQSTA & GP_TIMER_BIT)==GP_TIMER_BIT)//TIMER1 Interrupt, interval 20ms
      {
      T1CLRI = 0;                   // Clear Timer1 interrupt
      if(DetectionStatus==0xF2)     // Strike after weightlessness is detected, waiting for stable
      {
            TimerWaitForStable++;
            if(TimerWaitForStable>=STABLE_WINDOW)// Time out, restart
            {
                        IRQCLR  = GP_TIMER_BIT;// Disable ADuC7026’s Timer1 interrupt
                        DetectionStatus=0xF0;
                        putchar(DetectionStatus);
                        ADXL345Registers[XL345_THRESH_ACT]=STRIKE_THRESHOLD;
                        ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                        ADXL345Registers[XL345_TIME_INACT]=STABLE_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE  | XL345_INACT_Y_ENABLE
                                                                                    | XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                    | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
                                                                                    | XL345_ACT_X_ENABLE | XL345_ACT_DC;
      xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
            }
      }
      else if(DetectionStatus==0xF1)// Weightlessness is detected, waiting for strike
      {
            TimerWaitForStrike++;
            if(TimerWaitForStrike>=STRIKE_WINDOW)// Time out, restart
            {
                        IRQCLR  = GP_TIMER_BIT;// Disable ADuC7026’s Timer1 interrupt
                        DetectionStatus=0xF0;
                        putchar(DetectionStatus);
                        ADXL345Registers[XL345_THRESH_ACT]=STRIKE_THRESHOLD;
                        ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                        ADXL345Registers[XL345_TIME_INACT]=STABLE_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE  | XL345_INACT_Y_ENABLE
                                                                                    | XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                    | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
                                                                                    | XL345_ACT_X_ENABLE | XL345_ACT_DC;
xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
            }
      }
   }
   if((IRQSTA&SPM4_IO_BIT)==SPM4_IO_BIT)// External interrupt form ADXL345 INT0
   {
      IRQCLR = SPM4_IO_BIT;         // Disable ADuC7026’s external interrupt
      xl345Read(1, XL345_INT_SOURCE, &ADXL345Registers[XL345_INT_SOURCE]);
      if((ADXL345Registers[XL345_INT_SOURCE]&XL345_ACTIVITY)==XL345_ACTIVITY)// Activity interrupt asserted
      {
            if(DetectionStatus==0xF1)// Waiting for strike, and now strike is detected
            {
                        DetectionStatus=0xF2;// Go to Status “F2”
                        putchar(DetectionStatus);
                        ADXL345Registers[XL345_THRESH_ACT]=STABLE_THRESHOLD;
                        ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                        ADXL345Registers[XL345_TIME_INACT]=STABLE_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE  | XL345_INACT_Y_ENABLE
                                                                                    | XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                    | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
                                                                                    | XL345_ACT_X_ENABLE | XL345_ACT_AC;
      xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
                        IRQEN|=GP_TIMER_BIT;// Enable ADuC7026’s Timer1 interrupt
                        TimerWaitForStable=0;
            }
            else if(DetectionStatus==0xF4)// Waiting for long time motionless, but a movement is detected
            {
                        DetectionStatus=0xF0;// Go to Status “F0”, restart
                        putchar(DetectionStatus);
                        ADXL345Registers[XL345_THRESH_ACT]=STRIKE_THRESHOLD;
                        ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                        ADXL345Registers[XL345_TIME_INACT]=STABLE_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE  | XL345_INACT_Y_ENABLE
                                                                                    | XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                    | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
                                                                                    | XL345_ACT_X_ENABLE | XL345_ACT_DC;
   xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
            }
      }
      else if((ADXL345Registers[XL345_INT_SOURCE]&XL345_INACTIVITY)==XL345_INACTIVITY) // Inactivity interrupt asserted
      {
            if(DetectionStatus==0xF2)           // Waiting for stable, and now stable is detected
            {
                        DetectionStatus=0xF3;// Go to Status “F3”
                        IRQCLR  = GP_TIMER_BIT;
                        putchar(DetectionStatus);

                        xl345Read(6, XL345_DATAX0, &ADXL345Registers[XL345_DATAX0]);

                        DeltaVectorSum=0;
                        for(i=0;i<3; i++)
                        {
                                    Acceleration[i]=ADXL345Registers[XL345_DATAX1+i*2]&0x1F;
                                    Acceleration[i]=(Acceleration[i]<<8)|ADXL345Registers[XL345_DATAX0+i*2];
                                    if(Acceleration[i]<0x1000)
                                    {
                                                Acceleration[i]=Acceleration[i]+0x1000;
                                    }
                                    else //if(Acceleration[i]>=4096)
                                    {
                                                Acceleration[i]=Acceleration[i]-0x1000;
                                    }

                                    if(Acceleration[i]>InitialStatus[i])
                                    {
                                                DeltaAcceleration[i]=Acceleration[i]-InitialStatus[i];
                                    }
                                    else
                                    {
                                                DeltaAcceleration[i]=InitialStatus[i]-Acceleration[i];
                                    }

                                    DeltaVectorSum=DeltaVectorSum+DeltaAcceleration[i]*DeltaAcceleration[i];
                        }

                        if(DeltaVectorSum>DELTA_VECTOR_SUM_THRESHOLD)   // The stable status is different from the initial status
                        {
                                    DetectionStatus=0xF4;      // Valid  fall detection
                                    putchar(DetectionStatus);

                                    ADXL345Registers[XL345_THRESH_ACT]=STABLE_THRESHOLD;
                                    ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                                    ADXL345Registers[XL345_TIME_INACT]=NOMOVEMENT_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE | XL345_INACT_Y_ENABLE
| XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                                              | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
| XL345_ACT_X_ENABLE | XL345_ACT_AC;
                                    xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
                        }
                        else        // Delta vector sum is not exceed the threshold
                        {
                                    DetectionStatus=0xF0;      // Go to Status “F0”, restart
                                    putchar(DetectionStatus);
                                    ADXL345Registers[XL345_THRESH_ACT]=STRIKE_THRESHOLD;
                                    ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                                    ADXL345Registers[XL345_TIME_INACT]=STABLE_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE | XL345_INACT_Y_ENABLE
| XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                                              | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
| XL345_ACT_X_ENABLE | XL345_ACT_DC;
                                    xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
                        }
            }
            else if(DetectionStatus==0xF4) // Wait for long time motionless and now it is detected
            {
                        DetectionStatus=0xF5;      // Valid critical fall detection
                        putchar(DetectionStatus);

                        ADXL345Registers[XL345_THRESH_ACT]=STRIKE_THRESHOLD;
                        ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                        ADXL345Registers[XL345_TIME_INACT]=STABLE_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE | XL345_INACT_Y_ENABLE
| XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                                   | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
| XL345_ACT_X_ENABLE | XL345_ACT_DC;
                        xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
                        DetectionStatus=0xF0;               // Go to Status “F0”, restart
                        putchar(DetectionStatus);
            }
      }
      else if((ADXL345Registers[XL345_INT_SOURCE]&XL345_FREEFALL)==XL345_FREEFALL) // Free Fall interrupt asserted
      {
            if(DetectionStatus==0xF0)           // Waiting for weightless, and now it is detected
            {
                        DetectionStatus=0xF1;   // Go to Status “F1”
                        putchar(DetectionStatus);
                        ADXL345Registers[XL345_THRESH_ACT]=STRIKE_THRESHOLD;
                        ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                        ADXL345Registers[XL345_TIME_INACT]=STABLE_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE | XL345_INACT_Y_ENABLE
| XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                                   | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
| XL345_ACT_X_ENABLE | XL345_ACT_DC;
                        xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
                        IRQEN|=GP_TIMER_BIT;    // Enable ADuC7026’s Timer1 interrupt
                        TimerWaitForStrike=0;
                        TimerFreeFall=0;
            }
            else if(DetectionStatus==0xF1) // Waiting for strike after weightless, and now a new free fall is detected
            {
                        if(TimerWaitForStrike<FREE_FALL_INTERVAL)       // if the Free Fall interrupt is continuously assert within the time of
“FREE_FALL_INTERVAL”,
                        {           // then it is consider as a continuous free fall
                                    TimerFreeFall=TimerFreeFall+TimerWaitForStrike;
                        }
                        else        / Not a continuous free fall
                        {
                                    TimerFreeFall=0;
                        }
                        TimerWaitForStrike=0;
                        if(TimerFreeFall>=FREE_FALL_OVERTIME)                        // if the continuous time of free fall is longer than
“FREE_FALL_OVERTIME”
                        {           // consider that a free fall from high place is detected
                                    DetectionStatus=0xFF;
                                    putchar(DetectionStatus);
                                    ADXL345Registers[XL345_THRESH_ACT]=STRIKE_THRESHOLD;
                                    ADXL345Registers[XL345_THRESH_INACT]=NOMOVEMENT_THRESHOLD;
                                    ADXL345Registers[XL345_TIME_INACT]=STABLE_TIME;
            ADXL345Registers[XL345_ACT_INACT_CTL]=XL345_INACT_Z_ENABLE | XL345_INACT_Y_ENABLE
| XL345_INACT_X_ENABLE | XL345_INACT_AC
                                                                                                              | XL345_ACT_Z_ENABLE | XL345_ACT_Y_ENABLE
| XL345_ACT_X_ENABLE | XL345_ACT_DC;
                                    xl345Write(4, XL345_THRESH_ACT, &ADXL345Registers[XL345_THRESH_ACT]);
                                    DetectionStatus=0xF0;
                                    putchar(DetectionStatus);
                        }
            }
            else
            {
                        TimerFreeFall=0;
            }
      }
      IRQEN  |=SPM4_IO_BIT;         // Enable ADuC7026’s external interrupt
   }
}

void  main(void)
{
    ADuC7026_Initiate();                        // ADuC7026 initialization
   ADXL345_Initiate();                          // ADXL345 initialization
   DetectionStatus=0xF0;                        // Clear detection status, Start
   InitialStatus[0]=0x1000;         // X axis=0g, unsigned short int, 13 bit resolution, 0x1000 = 4096 = 0g, +/-0xFF = +/-256 = +/-1g
   InitialStatus[1]=0x0F00;         // Y axis=-1g
   InitialStatus[2]=0x1000;         // Z axis=0g
   IRQEN =SPM4_IO_BIT;                          // Enable ADuC7026’s external interrupt, to receive the interrupt from ADXL345 INT0
   while(1)// Endless loop, wait for interrupts
   {
      ;
   }
}
```

Measuring Noise of Low-Fixed-Gain Differential Amplifiers

By Angel Caballero

Noise, composed of small, random voltages, can be difficult to measure. Lab instruments add their own noise, further complicating the measurement. Special techniques are often used when measuring noise. For example, amplifiers are typically configured with high closed-loop gains, multiplying their input noise to make it easier to measure. Low-fixed-gain differential amplifiers present a greater challenge, however, as their integrated feedback and gain resistors preclude the use of a high-gain configuration. Additionally, differential-to-single-ended conversion is needed to interface with available spectrum analyzers. A second amplifier stage can provide gain and the differential-to-single-ended conversion, neatly solving both of these problems.

Figure 1 shows an ADA4950-1[1] selectable-gain (1, 2, or 3) differential amplifier followed by an AD8099[2] low-noise, low-distortion op amp. The AD8099, configured for a gain of 10, converts the differential output into a single-ended signal. Its 1-nV/$\sqrt{\text{Hz}}$ input-referred voltage noise is negligible compared to that of the ADA4950-1. The output of the ADA4950-1 is multiplied by 10, making its noise proportionately larger as well. With a 0.5-pF compensation capacitor and gain of 10, the AD8099 has enough bandwidth to measure the noise of the ADA4950-1 up to 10 MHz before the system's frequency response starts to roll off.

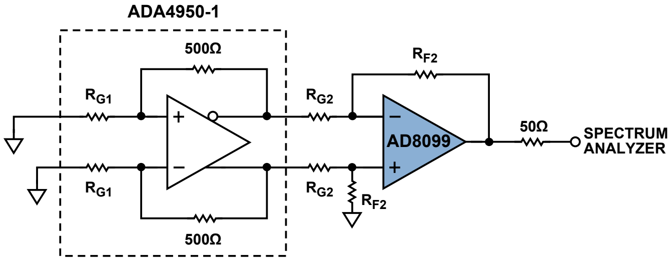


Figure 1. The AD8099 low-noise, low-distortion op-amp is used to measure the noise of the ADA4950-1 selectable-gain differential amplifier.

The output voltage of the AD8099 is simply:

$$V_{OUT} = V_{OUT,\,DIFF} \times \frac{R_{F2}}{R_{G2}} \qquad (1)$$

The noise contribution of the AD8099, measured with inputs grounded, is treated as the noise floor of the measurement system. The total output noise including the ADA4950-1 was then measured, with the contribution from the AD8099 subtracted using root-sum-square math, as shown in Equation 2, where V_{n1} is the output noise of the ADA4950-1 and V_{n2} is the output noise of the AD8099.

Total output noise:

$$V_{total}^{\;2} = \left(V_{n1} \times \frac{R_{F2}}{R_{G2}}\right)^2 + V_{n2}^{\;2} \qquad (2)$$

A few other techniques were implemented to accurately measure the system noise:

- When measuring the noise of the AD8099, its inputs were grounded with SMA connectors that had their center conductor shorted to the ground pins of the connector. Additionally, the SMA connectors were soldered together, creating a shared electrical connection to ground directly at the connectors, instead of through the board.
- An analog-controlled power supply was used for the AD8099 and the ADA4950-1. Compared to digitally controlled power supplies, analog-controlled power supplies are better at rejecting 60-Hz noise and harmonics that couple in from the power line.
- All nearby instruments were turned off unless they were being used for the measurement. This minimized oscillations generated by the instruments to control their digital circuitry. These oscillations can couple through the air and into the amplifiers. For the same reason, 4-ft cables were used to connect the circuit boards to the spectrum analyzer, which was picking up the refresh frequency of the display and affecting the output of the AD8099.
- Low-value resistors (R_F = 250 Ω; R_G = 25 Ω) were used to configure the AD8099's gain in order to keep their noise contribution small. Lower values caused the AD8099 to oscillate. When the ADA4950-1 was connected to the AD8099 with a short cable, an oscillation was observed at 250 MHz. When a 1-ft cable was used, the oscillations went away.

The AD8099 itself contributed only a small amount of noise:

$$v_{OUT}^{\;2} = \left[\left(1+\frac{R_{F2}}{R_{G2}}\right)v_n\right]^2 + R_{F2}^{\;2}\left(n_{i+}^{\;2}+n_{i-}^{\;2}\right) + 2\left(n_{RG2} \times \frac{R_{F2}}{R_{G2}}\right)^2 + 2n_{RF2}^{\;2} + n_{50}^{\;2} \qquad (3)$$

where v_n is the input voltage noise; and n_{i+} and n_{i-} are the input current noise of the AD8099.

Measuring the ADA4950-1's current noise is impossible because a large feedback resistor is needed to amplify the noise, but the value of the internal feedback resistor cannot be changed.

The Stanford Research Systems SR785 was used to measure noise up to 100 kHz, while the Agilent E4440 PSA spectrum analyzer was used for noise beyond 100 kHz.

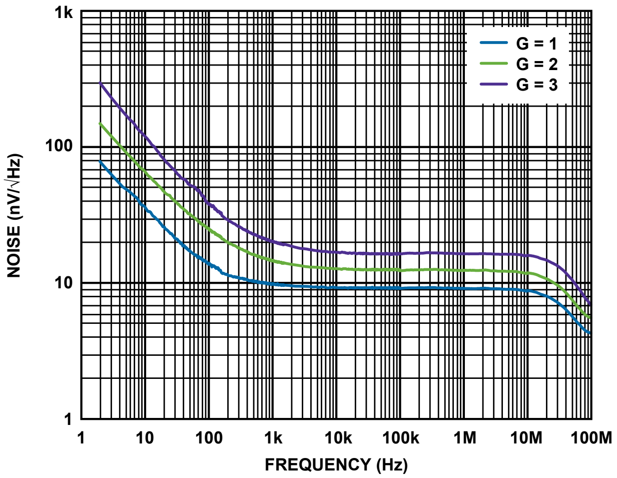


Figure 2. Test results.

References

[1] www.analog.com/en/amplifiers-and-comparators/differential-amplifiers/ada4950-1/products/product.html.

[2] www.analog.com/en/amplifiers-and-comparators/operational-amplifiers-op-amps/ad8099/products/product.html.

The Basics of Video Decoders in Supervision and Inspection

By Witold Kaczurba

Video inspection[1] systems are used in many commercial and industrial processes. Cameras—which range from those in inexpensive, low-definition black-and-white closed-circuit television (CCTV) systems to those in state-of-the-art high-definition digital-video systems—are used in diverse applications ranging from product inspection to traffic monitoring to real-time face recognition.

Video inherently carries a lot of data, which can complicate signal-processing and data-storage tasks. Video inspection can often be simplified by cropping useless information and passing only the essential parts of the picture, which saves both memory and computational cycles. Figure 1 shows the elements of a typical system.

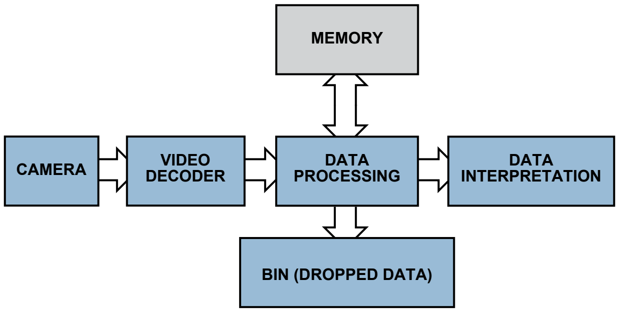


Figure 1. Simplified video-inspection data flow.

This article shows a few examples of how extracting useful data can minimize processing, memory size, and DSP usage—and illustrates how special features of Analog Devices video decoders[2] can simplify video algorithms and speed up development of video inspection systems.

Example 1. Counting and Inspecting Objects

Imagine a wide conveyor belt containing many rapidly moving products. The large number of products makes manual counting difficult. In addition to automating the counting task, a camera can be used to monitor product quality. This can be accomplished by modifying the simple count algorithm to focus on particular details and artifacts.

Storing all of the video data requires a huge amount of memory, and processing a large amount of data will cost a lot in terms of hardware and power. Instead of collecting whole pictures in memory, the system can find interesting details in the pile of data and drop as much useless data as possible when inspecting the products on the conveyor belt.

In most cases, gray-scale pictures can carry enough information. So chrominance information can be dropped by converting RGB signals to Y (luminance only). The resulting monochromatic picture can then be examined for content by using *edge detection* to find products on the belt and compare their shapes with a template to determine whether the product is misshapen.

Edge detection algorithms—which require only a few lines of active video and a small amount of memory—find discontinuities in the brightness of adjacent pixels by calculating the first and second derivatives of active pictures, as described in *Digital Image Processing* by Bernd Jähne.[3] Edge detection can be implemented in practice by extracting information using matrix calculations, such as the Sobel[4] matrix operator. In an FPGA (field-programmable gate-array) implementation, doing this on a pixel basis gives satisfying results. A simple FPGA implementation is shown in *A Proposed FPGA Based Architecture for Sobel Edge Detection Operator* by Tanvir A. Abbasi and Mohm. Usaid Abbasi.[5] Noise can be removed by adding a Gaussian 2D filter, as described in *Hardware Acceleration of Edge Detection Algorithm on FPGAs* by Mathukumar Venkatesan and Daggu Venkateshwar Rao,[6] which describes a successful implementation of a detector similar to the Canny edge detector.[7]

Several other optimization algorithms can enhance the picture quality, but all occupy significant space on the FPGA design. However, some integrated-circuit (IC) video decoders are already equipped with useful preprocessing algorithms or filters; so choosing one of these would save space in the FPGA. For example, the ADV7802 video decoder[8] includes both *luma transient improvement* (LTI) and *chroma transient improvement* (CTI) blocks. These blocks, which enhance the resulting picture by improving the steepness of luma and chroma transitions, use adaptive peaking and nonlinear methods—without increasing noise or introducing artifacts—and can be very useful in the process of edge detection. In addition, luma-shaping and other built-in input filters can remove high-frequency noise from the source—focusing on the signal and ignoring incidental noise.

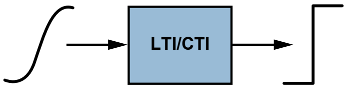


Figure 2. LTI/CTI operation diagram.

Edge detection provides information on an object's edge transitions instead of a full picture of the object. This reduction, from 3 × 8 bits per pixel (bpp) to 1 bpp, saves a lot of memory:

- •640 pixels × 480 pixels = 307,200 bits at 1 bpp
- •800 pixels × 600 pixels = 480,000 bits at 1 bpp
- •1024 pixels × 768 pixels = 786,432 bits at 1 bpp
- •1280 pixels × 720 pixels = 921,600 bits at 1 bpp

By converting RGB to Y, storing just a few lines of active video in memory, and using FPGA algorithms, we can detect objects and see their shapes. Once their locations on the moving belt are known, we can estimate their movement and collect color or other information from the next frames with the assurance that a minimum amount of memory is being used. The process involves

1. Edge detection
2. Storing information
3. Predicting the next position x_{n+1}
4. Extracting information in areas where product is supposed to be

Example 2. Detecting Motion and Quality

A robot is looking for items at a particular distance and within a limited range. Ultrasound can be used in some applications; but if the surface absorbs ultrasound or the items are behind glass, video can be used. The camera is set to focus on nearby objects. Items within a narrow range will have sharp edges, but background items—which are outside that range—have fuzzy edges (Figure 3).

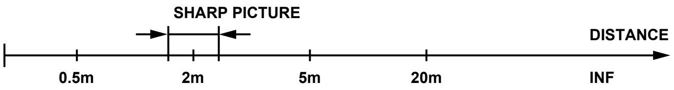


Figure 3. Focus—narrow depth of field.

Edge detection can be used to distinguish the items within the target range, as these are the only ones with sharp edges. Items in the background will be fuzzy enough to fail an edge detection test. Processing yields a binary bitmap where a 1 means that an edge was detected and a 0 means that no edge was detected. The position (x, y) of each detected edge pixel can be used to approximate the middle of an isolated object using Equation 1:

$$x_{average}=\frac{\sum_{n}^{N}x_n}{N}\ ;\qquad y_{average}=\frac{\sum_{n}^{N}y_n}{N} \tag{1}$$

Where x_n is the x-position of edge pixel, n; y_n is the y-position of edge pixel, n; and N is the number of edge pixels detected.

Once the position of the object and its edges are known, we can try to trace it. The key is to extract exactly one object from the picture, transforming its edges to an outline that can be used to determine if the item is moving toward the camera by checking the average distance of pixels from the middle of the object to see if the size of object is changing, as shown in Equation 2 (below).

N is the number of edge pixels in FRAME; M is the number of edge pixels in FRAME–1.

Focusing on the horizontal axis leads to Equation 3 (below).

The value of this equation will be positive when the object is moving toward the camera (pixels are spreading from the middle of object). A negative value means that the object is moving away from the camera, as shown in Figure 4.

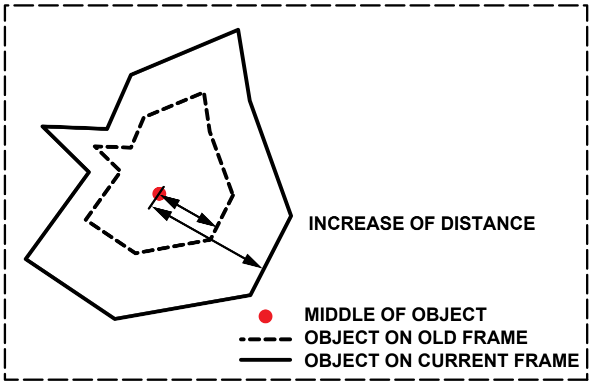


Figure 4. Frame change of moving object.

Note that the object has to be within the camera's range of focus. By modifying the algorithm, we can actively change the focus[9] to scan a wider area. Once the objects are detected, they can be segmented, processed, and tracked.

Tracking objects becomes more difficult as video complexity increases, especially with textured objects and objects that lose sharpness because they move quickly. Some tracking algorithms are shown in *Good Features to Track* by Jianbo Shi.[10] As objects lose sharpness, edge detection fails. Tracking can still be done by using complex correlation techniques such as block matching—used to estimate motion—or other methods detailed in *Video Processing and Communications* by Yao Wang, Jörn Ostermann, and Ya-Qin Zhang.[11]

Thanks to continuous data flow from the camera, an object can be tracked to determine its acceleration and other parameters. However, a high-quality video sequence must be used in order to obtain good video analysis results. When detecting edges by analyzing adjacent pixels, the resolution will be better if progressive-scan video is used instead of low-quality interlaced PAL or NTSC signals. The ADV7401 and ADV7403 video decoders[12] accept a variety of video standards, including progressive modes. Capable of digitizing video signals up to 140 MHz, they can handle SD, ED, and HD component signals, CVBS, and graphics. In addition, they support nonstandard video modes, allowing the use of less-popular standards, such as STANAG. The flexible pixel output bus allows data processing in 4:2:2, 4:4:4 YCrCb, or 4:4:4 RGB formats. Nonstandard video formats can be oversampled or undersampled to get a given horizontal width, as described in AN-0978 Application Note, *Component Processor Nonstandard Video Formats*.[13]

The built-in *color-space converter* (CSC), shown in Figure 5, transforms the color space to suit user requirements (Equation 4 below, where A1… A4, B1… B4, C1… C4 are adjustable CSC parameters). YPrPb or RGB input signals can be converted to other formats using configurable matrix conversion. For instance, converting RGB to YCrCb allows chroma information (Cb, Cr) to be dropped, simplifying edge detection with a monochrome picture.

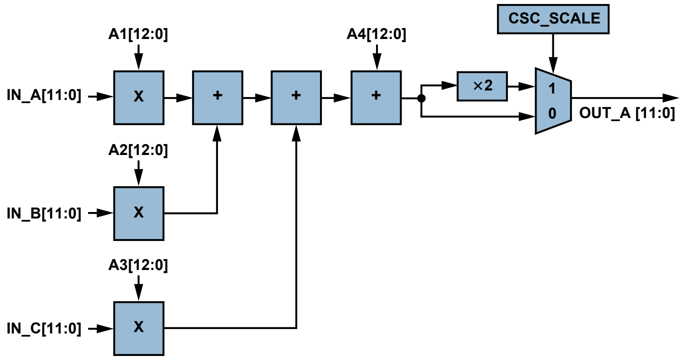


Figure 5. Single CSC channel (ADV7403).

$$\frac{1}{N}\sum_{n}^{N}\left(\sqrt{\left(x_n^{FRAME}-x_{average}^{FRAME}\right)^2+\left(y_n^{FRAME}-y_{average}^{FRAME}\right)^2}\right)- \\ +\frac{1}{M}\sum_{m}^{M}\left(\sqrt{\left(x_m^{FRAME-1}-x_{average}^{FRAME-1}\right)^2+\left(y_m^{FRAME-1}-y_{average}^{FRAME-1}\right)^2}\right) \tag{2}$$

$$\frac{1}{N}\sum_{n}^{N}\left(x_i^{FRAME}-x_{average}^{FRAME}\right)-\frac{1}{M}\sum_{n}^{M}\left(x_i^{FRAME-1}-x_{average}^{FRAME-1}\right) \tag{3}$$

$$\begin{bmatrix} Channel\_A\_out \\ Channel\_B\_out \\ Channel\_C\_out \end{bmatrix}=\begin{bmatrix} A1 & A2 & A3 \\ B3 & B1 & B2 \\ C2 & C3 & C1 \end{bmatrix}\begin{bmatrix} Channel\_A\_in \\ Channel\_B\_in \\ Channel\_C\_in \end{bmatrix}+\begin{bmatrix} A4 \\ B4 \\ C4 \end{bmatrix} \tag{4}$$

The CSC is very useful. With an RGB or YCrCb input, color information can be simply transformed using a color-space matrix. Figure 6 shows a YUV color space that is similar to YCrCb.

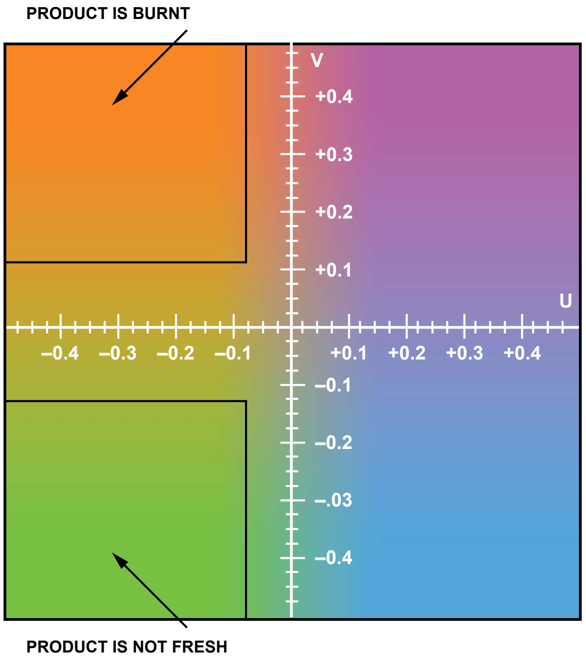


Figure 6. YUV color space in product-quality evaluation can be used to detect (for example) when a product is burned or moldy. Y (luma) is constant.

As Figure 6 shows, the color (or YPrPb value), can help to detect the quality of the product, for example, whether it is burned or moldy. Color-space conversion is necessary in video processing and for interfacing to ICs that use other standards. The ADV7401/ADV7403 include an input multiplexer that enables easy switching of video sources, a useful feature when switching from a stopped conveyor belt to a working one.

Example 3. Adjusting White and Color Balance for Video Inspection
Significant effort is required to develop a video system that extracts objects from a picture, as simple changes in light angle or intensity can affect the inspection results. Video engineers can use the ADV7401/ADV7403 gain and offset adjustments to adjust the brightness and contrast by adding two small reference stripes (one dark, one bright) to the conveyor belt. The offset and gain of the ADV7401/ADV7403 are adjusted to get comparable values, thus allowing the system to compensate for changes in light, color, angle, and intensity.

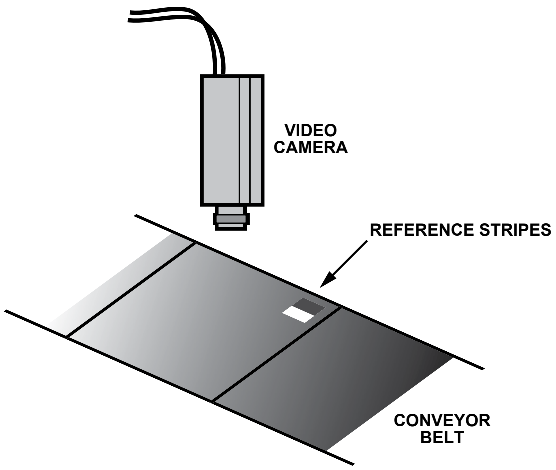


Figure 7. Small reference stripes are added to the visible area.

The algorithm for adjusting proper white balance[14] can be very simple. First, get reference RGB (or YCrCb) values for the stripes. Then, to compensate for light, simply change the offset and gain to get the same values as the reference. This algorithm can be used:

1. Get RGB (or YCrCb) values of the dark stripe.
2. Adjust offset to match desired RGB (or YCrCb) value of dark stripe.
3. Get RGB (or YCrCb) values of the light stripe
4. Adjust gain to match desired RGB (or YCrCb) value of light stripe.
5. To improve accuracy, repeat steps 2 and 4.

This procedure is especially useful during system development, as it provides the correct offset (brightness) and gain (contrast)—even when the light is too strong or too weak, as shown in Figure 8. The offset and gain registers are available via the I^2C bus, allowing quick adaptation.

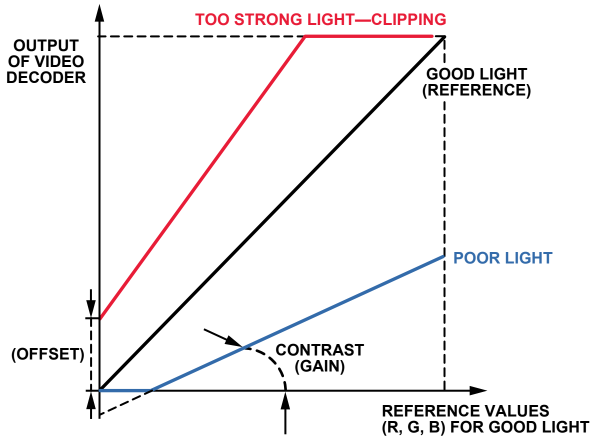


Figure 8. The offset and gain are adjusted to compensate for changes in ambient lighting.

Colors can also be used for the reference stripes. This compensation is similar to white balance, which is widely used, but while white balance matches a human's perception, the color correction is to compensate for changes due to different lighting. Although the algorithm is similar, an additional offset causes dark colors to look unnatural. The ADV7401/ADV7403 color-space conversion, flexible output pixel port, and offset and gain adjustment registers allow engineers to quickly develop algorithms using data that is already prepared for processing. As discussed earlier, it is important to reduce the amount of data required for video processing and to avoid advanced algorithms if they're not needed for simple video. An evaluation board for the ADV7401/ADV7403 with an easily accessible pixel port is available to speed up the start of new design. It's a matter of simply plugging a video-capture board into the pixel port of the evaluation board and capturing the video-data (Figure 9).

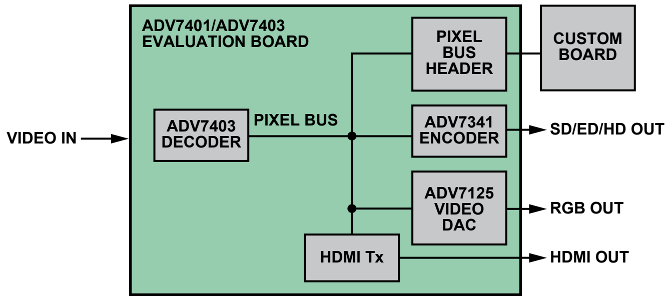


Figure 9. Pixel bus on ADV7401/ADV7403 evaluation board.

continued on page 17

How to Apply Low-Dropout Regulators Successfully

By Ken Marasco

A low-dropout regulator (LDO) is capable of maintaining its specified output voltage over a wide range of load current and input voltage, down to a very small difference between input and output voltages. This difference, known as the dropout voltage or headroom requirement, can be as low as 80 mV at 2 A. The adjustable-output low-dropout regulator[1] first came to public attention in 1977. Nowadays, portable devices often require up to 20 low-dropout linear regulators. Many of the LDOs in today's portable devices are integrated into multifunction power-management ICs[2] (PMICs)—highly integrated systems with 20 or more power domains for audio, battery charging, housekeeping, lighting, communications, and other functions.

As portable systems rapidly evolve, however, the integrated PMIC cannot keep up with peripheral power requirements. Dedicated LDOs must be added in the later stages of system development to power such optional items as camera modules, Bluetooth, Wi-Fi, and other bolt-on modules. LDOs have also been used as band-aids for noise reduction, to solve voltage-regulation problems caused by electromagnetic interference (EMI) and printed-circuit board (PCB) routing, and to improve system efficiency by switching off unneeded functions.

This article reviews the basic LDO topology, explains key specifications, and shows the application of low-dropout voltage regulators in systems. Examples will be given using design characteristics of Analog Devices LDO families.[3]

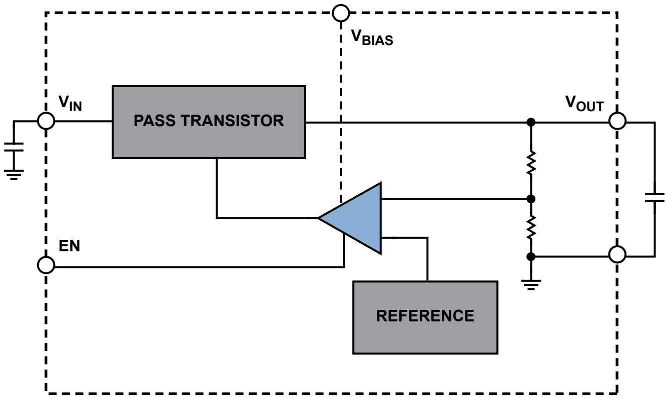


Figure 1. An LDO regulates the output voltage with a low dropout voltage (the difference between V_{OUT} and the lowest specified value of V_{IN} at the rated load current).

Basic LDO Architecture.[4] An LDO consists of a voltage reference, an error amplifier, a feedback voltage divider, and a pass transistor, as shown in Figure 1. Output current is delivered via the pass device. Its gate voltage is controlled by the error amplifier—which compares the reference voltage with the feedback voltage, amplifying the difference so as to reduce the error voltage. If the feedback voltage is lower than the reference voltage, the gate of the pass transistor is pulled lower, allowing more current to pass and increasing the output voltage. If the feedback voltage is higher than the reference voltage, the gate of the pass transistor is pulled higher, restricting the current flow and decreasing the output voltage.

The dynamics of this closed-loop system are based on two main poles—the internal pole formed by the error amplifier/pass transistor and the external pole formed by the amplifier's output impedance and the output capacitor's equivalent series resistance (ESR). The output capacitance and its ESR affect the loop stability and the response to transient changes in load current. An ESR of 1 Ω or less is recommended to ensure stability. Also, LDOs require input and output capacitors to filter noise and control load transients. Larger values improve the transient response of the LDO but increase the startup time. Analog Devices LDOs are designed to be stable over the specified operating conditions when the specified capacitors are used.

LDO Efficiency. Increased efficiency is a constant demand from the design engineer. This translates into a reduction of the quiescent current (I_Q) and forward voltage drop.

$$LDO\ efficiency = \left(\frac{V_{OUT}\, I_{OUT}}{V_{IN}\left(I_{OUT}+I_Q\right)}\right) \times 100\% \tag{1}$$

With I_Q in the denominator, it is evident that the higher I_Q, the lower the efficiency. Today's LDOs have reasonably low I_Q, and for simplicity, I_Q can be neglected in efficiency calculations if I_Q is very small compared to the load current, I_{OUT}. Then LDO efficiency is simply (V_{OUT}/V_{IN}) × 100%. Because the LDO has no way to store significant amounts of unused energy, power not delivered to the load is dissipated as heat within the LDO.

$$Power\ dissipated\ (P_D) = \left(V_{IN} - V_{OUT}\right) \times I_{IN} \tag{2}$$

Providing a stable power supply voltage independent of load and line variations, changes in ambient temperature, and the passage of time, LDOs are most efficient with small differences between supply voltage and load voltage. For example, as a lithium-ion battery drops from 4.2 V (fully charged) to 3.0 V (discharged), a 2.8-V LDO connected to the battery will maintain a constant 2.8 V at the load (dropout voltage less than 200 mV), but its efficiency would increase from 67% with the fully charged battery to 93% with the discharged battery.

To improve efficiency, LDOs can be connected to an intermediate voltage rail generated by a high-efficiency switching regulator. With a 3.3-V switching regulator, for example, the LDO efficiency would be constant at 85%, and the overall system efficiency would be 81%, assuming 95% efficiency for the switching regulator.

Circuit Features Enhance LDO Performance. An enable input permits external control of LDO turn-on and turn-off, allowing supplies to be sequenced in proper order in multirail systems. Soft-start limits inrush current and controls output-voltage rise time during power-up. A sleep state minimizes power drain, especially useful in battery-based systems, while allowing fast turn-on. Thermal shutdown turns the LDO off if its temperature exceeds the specified value. Overcurrent protection limits the LDO's output current and power dissipation. Undervoltage lockout disables the output when the supply voltage is below the specified minimum value. Figure 2 shows a simplified typical power system for portable designs.

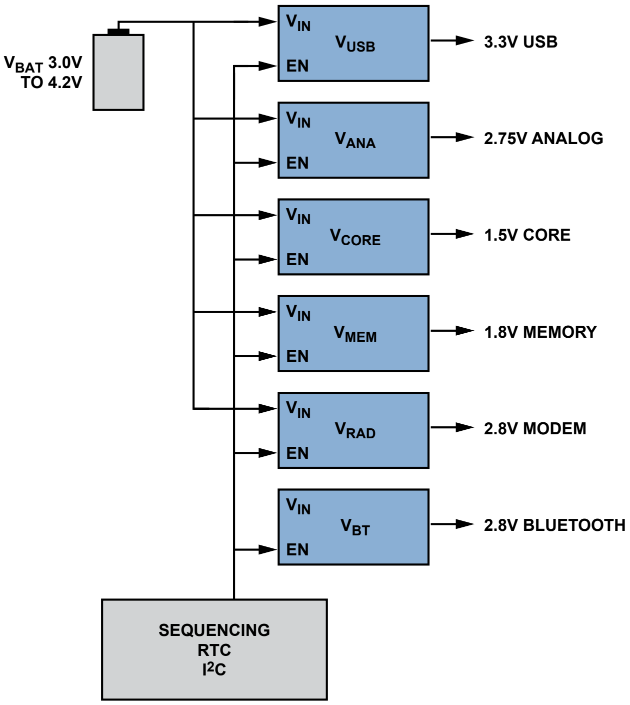


Figure 2. Typical power domains in a portable system.

Understanding Linear Regulator Requirements

LDOs for Digital Loads. Digital linear regulators, such as the ADP170 and ADP1706, are designed to support the main digital requirements of a system, usually microprocessor cores and system input/output (I/O) circuitry. LDOs for DSPs and microcontrollers have to work with good efficiency and handle high and rapidly varying currents. Newer application requirements put tremendous strain on the digital LDO because processor cores often change clock speed to save energy. The clock-speed variation, in response to software-induced loading, translates into a demanding need for LDO load-regulation capability.

The important characteristics for digital loads are line- and load regulation and transient under- and overshoot. When powering low-voltage microprocessor cores, accurate output-control is very important at all times; inadequate regulation can allow the core to latch up. The above parameters are not always featured in data sheets, and graphs of transient response may show optimistic rise and fall speed in response to transient signals. Line and load regulation is stated in two ways: percent deviation of output voltage with load change, actual V/I values, or both, at a specific load current.

To save energy, digital LDOs are designed with low I_Q to increase battery life, and portable systems have long periods of low-power operation when the software is idling. During periods of inactivity, systems will go to sleep—requiring the LDO to shut down and consume less than 1 µA. While the LDO is in sleep mode, all of the circuits—including the band-gap reference—are switched off. When the system returns to active mode, fast turn-on times are required—during which the digital supply voltage must not overshoot excessively. Excessive overshoot can result in system latch up, sometimes requiring the removal of the battery or activation of the master reset button to correct the problem and restart the system.

LDOs for Analog and RF Loads. Low noise and high power supply rejection (PSR), as found in the ADP121 and ADP130, are important for LDOs used in the analog environment because analog devices are more sensitive to noise than digital devices. Analog LDO requisites are mainly driven by the wireless interface requirements: "Do no harm to the receiver or transmitter, and create no pop or hum in the audio system." The wireless connection is highly susceptible to noise, and the receiver's effectiveness can be reduced if the noise interferes with the signal. When considering an analog linear regulator, it is important for the device to suppress noise from upstream sources and downstream loads, while not adding further noise itself.

Analog regulator noise is measured in volts rms and PSR, its ability to suppress upstream noise. Adding an external filter or a bypass capacitor can reduce noise, but it adds cost and size. Noise reduction and supply-noise rejection can also be achieved by care and ingenuity in the LDO's internal design. When selecting LDOs, it is important to review the product details in relation to the overall performance needed for each system.

Key LDO Specifications and Definitions

Note: Specifications on the front page of manufacturers' data sheets are brief summaries, often presented in a way that emphasizes the device's attractive features. The key parameters often emphasize typical performance characteristics, which can be more fully understood only when consulting the complete specifications and other data within the body of the document. Also, because there is little standardization among manufacturers in the way specifications are presented, power designers need to understand the definition and the methodology used to obtain key parameters listed in the electrical specifications table. The system designer should pay close attention to key parameters, such as ambient and junction temperature range, X-Y scales of graphic information, loads, rise- and fall times of transient signals, and bandwidth. We list here a discussion of important parameters relating to the characterization and application of Analog Devices LDOs.

Input Voltage Range: An LDO's input voltage range determines the lowest usable input supply voltage. The specifications may show a wide input-voltage range, but the lowest input voltage must be greater than the dropout voltage plus the desired output voltage. For example, a 150-mV dropout means that the input voltage must be above 2.95 V for a regulated 2.8-V output. If the input voltage drops below 2.95 V, the output voltage will drop below 2.8 V.

Ground (Quiescent) Current: The quiescent current, I_Q, is the difference between the input current, I_{IN}, and the load current, I_{OUT}, measured at the specified load current. For fixed-voltage regulators, I_Q is the same as the ground current, I_G. For adjustable-voltage regulators, such as the ADP171, the quiescent current is the ground current minus the current from the external resistance-divider network.

Shutdown Current: The input current consumed when the device has been disabled. Usually below 1.0 µA for portable LDOs, this specification is important for battery charge life during long standby times when the portable device is turned off.

Output Voltage Accuracy: Analog Devices LDOs are designed for high output-voltage accuracy; they are factory-trimmed to within ±1% at 25°C. Output-voltage accuracy is specified over the operating-temperature, input-voltage, and load-current ranges; error is specified as ±x% worst case.

Line Regulation: Line regulation is the change in output voltage for a change in the input voltage. To avoid inaccuracy due to changes in chip temperature, the measurement is made under conditions of low power dissipation or by using pulse techniques.

Dynamic Load Regulation: Most LDOs can easily hold the output voltage nearly constant as long as the load current changes slowly. When the load current changes quickly, however, the output voltage will change briefly. How much the output voltage changes when subjected to a change in load current defines load transient performance.

Dropout Voltage: Dropout refers to the smallest difference between input and output voltages required to maintain regulation. That is, an LDO can hold the output load voltage constant as the input is decreased until the input reaches the output voltage plus the dropout voltage, at which point the output "drops out" of regulation. The dropout voltage should be as low as possible to minimize power dissipation and maximize efficiency. Typically, dropout is considered to be reached when the output voltage has dropped to 100 mV below its nominal value. The load current and junction temperature can affect the dropout voltage. The maximum dropout voltage should be specified over the full operating temperature range and load current.

Start-Up Time: Start-up time is defined as the time between the rising edge of the enable signal and V_{OUT} reaching 90% of its nominal value. This test is usually performed with V_{IN} applied and the enable pin toggled from off to on. Note: in some cases where the enable is connected to V_{IN}, the start-up time can substantially increase because the band gap reference takes time to stabilize. Start-up time of a regulator is an important consideration for applications where the regulator is frequently turned off and on to save power in portable systems.

Current-Limit Threshold: Current-limit threshold is defined as the load current at which the output voltage drops to 90% of the specified typical value. For example, the current limit for a 3.0-V output voltage is defined as the current that causes the output voltage to drop to 90% of 3.0 V, or 2.7 V.

Operating Temperature Range: Operating temperature range can be specified by ambient and junction temperature. Because an LDO dissipates heat, the IC will always operate above the ambient. How much above the ambient temperature depends on the operating conditions and the PCB thermal design. A maximum junction temperature (T_J) is specified because operation above the maximum junction temperature for extended periods may affect device reliability—statistically specified as mean time to failure (MTTF).

Thermal Shutdown (TSD): Most LDOs have silicon thermostats for protecting the IC from thermal runaway. They are used to power down the LDO if the junction temperature exceeds the specified thermal shutdown threshold. Hysteresis is required to allow the LDO to cool down before restarting. TSD is important because it protects more than the LDO alone; excessive heat affects more than just the regulator. Heat conducted from the LDO to the PCB (or to the LDO from hotter elements on the board) can damage the PCB material and solder connections over time, and also damage nearby components, reducing the life of the portable device. Also, thermal shutdown affects system reliability. Thermal design to control the board temperature (heat sinking, cooling, etc.) is thus an important system consideration.

Enable Input: LDO enables, offered in positive and negative logic, turn the device on and off. Active-high logic enables the device when the enable voltage exceeds the logic high threshold. Active-low logic enables the device when the enable voltage is below the logic low threshold. The enable input permits external control of LDO turn-on and turn-off, an important feature in the sequencing of supplies in multirail systems. Some LDOs have substantially shorter start-up times, because their band gap reference is on while the LDO is disabled, allowing the LDO to turn on faster.

Undervoltage Lockout: Undervoltage lockout (UVLO) will ensure voltage is supplied to the load only when the system input voltage is above the specified threshold. UVLO is important because it only allows the device to power on when the input voltage is at or above what the device requires for stable operation.

Output Noise: The LDO's internal band gap voltage reference is the source of noise, usually specified in microvolts rms over a specific bandwidth. For example, the ADP121 has an output noise of 40 μV rms from 10 kHz to 100 kHz at a V_{OUT} of 1.2 V. When comparing data sheet specifications, the specified bandwidth and operating conditions are important considerations.

Power-Supply Rejection: PSR, expressed in decibels, is a measure of how well the LDO rejects ripple from the input power supply over a wide frequency range (1 kHz to 100 kHz). In an LDO, PSR can be characterized in two frequency bands. Band 1 is from dc to the control loop's unity-gain frequency; PSR is set by the open-loop gain of the regulator. Band 2 is above the unity-gain frequency; PSR is unaffected by the feedback loop. Here PSR is set by the output and any leakage paths from the input to the output pins. Choosing a suitably high-value output capacitance will typically improve PSR in this latter band. In Band 1, Analog Devices proprietary circuit design reduces changes in PSR due to input voltage and load variations. For optimum supply rejection, the PCB layout must be considered to reduce the leakage from input to output, and grounding should be robust.

Minimum Input- and Output Capacitance: The minimum input- and output capacitance should be greater than specified over the full range of operating conditions, especially operating voltage and temperature. The full range of operating conditions in the application must be considered during device selection to ensure that the minimum capacitance specification is met. X7R- and X5R-type capacitors are recommended; Y5V and Z5U capacitors are not recommended for use with any LDO.

Reverse-Current Protection Feature: A typical LDO with a PMOS pass device has an intrinsic body diode between V_{IN} and V_{OUT}. When V_{IN} is greater than V_{OUT}, this diode is reverse-biased. If V_{OUT} is greater than V_{IN}, the intrinsic diode becomes forward-biased and conducts current from V_{OUT} to V_{IN}, potentially causing destructive power dissipation. Some LDOs, such as the ADP1740/ADP1741, have additional circuitry to protect against reverse current flow from V_{OUT} to V_{IN}. The reverse current protection circuitry detects when V_{OUT} is greater than V_{IN} and reverses the direction of the intrinsic diode connection, reverse-biasing the diode.

Soft Start: Programmable soft start is useful for reducing inrush current upon startup and for providing voltage sequencing. For applications that require a controlled inrush current at startup, LDOs such as the ADP1740/ADP1741 provide a programmable soft-start (SS) function. To implement soft start, a small ceramic capacitor is connected from SS to GND.

Conclusion

LDOs perform a vital function. Though simple in concept, there are many factors to consider in applying them. This article has reviewed basic LDO topology, and explained key specifications and application of low-dropout voltage regulators in systems. Data sheets contain much helpful information. Further information (selection guides, data sheets, application notes)—as well as ways to get human help—is available on the power management[5] website. Also available is ADIsimPower™[6], the fastest, most accurate dc-to-dc power management design tool.

References

(Information on all ADI components can be found at www.analog.com)

[1] Dobkin, R. "Break Loose from Fixed IC Regulators." *Electronic Design*. April 12, 1977.

[2] www.analog.com/en/power-management/multifunction-power-ics/products/index.html.

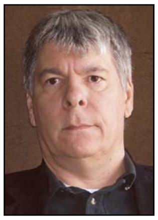
[3] www.analog.com/en/power-management/linear-regulators/products/index.html.

[4] Patoux, J. "Low-Dropout Regulators (Ask The Applications Engineer—37)." *Analog Dialogue* 41-2 (2007). pp. 8-10. www.analog.com/library/analogDialogue/archives/41-05/ldo.html.

[5] www.analog.com/en/power-management/products/index.html.

[6] http://designtools.analog.com/dtPowerWeb/dtPowerMain.aspx.

Author



Ken Marasco [ken.marasco@analog.com] is a system applications manager. Responsible for the technical support of Portable Power Products, he has been a member of the Portable Applications Team for three years. He graduated from NYIT with a degree in applied physics and has 35 years of system and component design experience.

continued from page 13

The video encoder, video DAC, and AD9889B HDMI transmitter[15] are connected to the same pixel bus, allowing viewing of the current picture on the second output. Analog Devices video decoders include the blocks required for processing video, giving robust performance and stable pictures.

Conclusion

Video cameras provide many benefits in industrial applications. This is particularly important when moving items must be sorted, tracked, or recorded. Video technology and real-time processing with highly integrated video decoders can be used to efficiently analyze items or sort mixed products on a moving conveyor belt.

References

[1] http://www.analog.com/en/industrial-solutions/machine-vision/applications/index.html.

[2] http://www.analog.com/en/audiovideo-products/video-decoders/products/index.html.

[3] http://www.amazon.com/Digital-Image-Processing-CD-ROM-Bernd/dp/3540677542.

[4] http://en.wikipedia.org/wiki/Sobel_operator.

[5] http://www.oldcitypublishing.com/FullText/JAPEDfulltext/JAPED2.4fulltext/Abbasi.pdf.

[6] http://www.uweb.ucsb.edu/~shahnam/HAoEDAoF.pdf.

[7] http://en.wikipedia.org/wiki/Canny_edge_detector.

[8] http://www.analog.com/en/analog-to-digital-converters/video-decoders/adv7802/products/product.html.

[9] http://en.wikipedia.org/wiki/Autofocus.

[10] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=323794.

[11] http://www.amazon.com/Video-Processing-Communications-Prentice-Hall-Signal/dp/0130175471.

[12] http://www.analog.com/en/analog-to-digital-converters/video-decoders/adv7403/products/product.html.

[13] http://www.analog.com/static/imported-files/application_notes/AN-0978.pdf.

[14] http://en.wikipedia.org/wiki/White_balance.

[15] http://www.analog.com/en/audiovideo-products/analoghdmidvi-interfaces/ad9889b/products/product.html.

Author

Witold (Vito) Kaczurba [witold.kaczurba@analog.com] is an applications engineer in our Advanced TV Group in Limerick, Ireland, where he supports decoders and HDMI products. He joined ADI in 2007, following graduation from Technical University of Wroclaw, Poland, with a master's degree in electronic engineering. As a student, he gained experience while working for small electronic and IT companies, then joined Analog Devices in Ireland as a co-op student—and subsequently as an applications engineer. In his spare time, he enjoys sightseeing Ireland, tasting Guinness beer with his friends, and reading.

Phase Response in Active Filters

Part 2, the Low-Pass and High-Pass Responses

By Hank Zumbahlen

A previous article[1] examined the relationship of the filter phase to the topology of its implementation. This article will examine the phase shift of the filter transfer function itself. While filters are designed primarily for their amplitude response, the phase response can be important in applications such as time-delay simulation, cascaded filter stages, and especially process-control loops.

This article will concentrate on the low-pass and high-pass responses. Future articles in this series will examine the band-pass and notch (band-reject) responses, the all-pass response, and the impulse and step responses of the filter.

To review, the transfer function of an active filter can be viewed as the cascaded response of the filter transfer function and an amplifier transfer function (Figure 1).

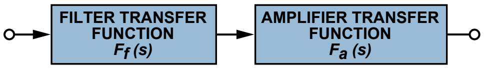

FILTER TRANSFER FUNCTION $F_f(s)$
AMPLIFIER TRANSFER FUNCTION $F_a(s)$

Figure 1. Filter as cascade of two transfer functions.

The Low-Pass Transfer Equation

First, we will reexamine the phase response of the transfer equations.

For the single-pole low-pass case, the transfer function has a phase shift given by:

$$\phi(\omega) = -\tan^{-1}\left(\frac{\omega}{\omega_0}\right) \tag{1}$$

where ω represents a radian frequency ($\omega = 2\pi f$ radians per second; 1 Hz = 2π radians per second) and ω_0 denotes the radian *center frequency* of the filter. The center frequency can also be referred to as the *cutoff frequency*. In terms of phase, the center frequency will be the frequency at which the phase shift is at 50% of its range. Since the radian frequency is used in a ratio, the frequency ratio, f/f_0, can be easily substituted for ω/ω_0.

Figure 2 (left axis) evaluates Equation 1 from two decades below the center frequency to two decades above the center frequency. Since a single-pole low-pass filter has a 90° range of phase shift—from 0° to 90°—the center frequency has a phase shift of –45°. At $\omega = \omega_0$ the normalized center frequency is 1.

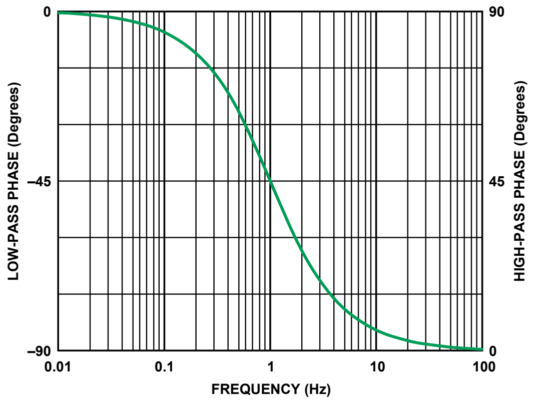


Figure 2. Phase response of a single-pole low-pass filter (left axis) and high-pass filter (right axis) with a center frequency of 1.

Similarly, the phase response of a single-pole high-pass filter is given by:

$$\phi(\omega) = \frac{\pi}{2} - \tan^{-1}\left(\frac{\omega}{\omega_0}\right) \tag{2}$$

Figure 2 (right axis) evaluates Equation 2 from two decades below to two decades above the center frequency. The center frequency (=1) has a phase shift of +45°.

If the low-pass *pass band* is defined as frequencies below the cutoff frequency and the high-pass pass band as frequencies above the center frequency, note that the lowest phase shifts (0° to 45°) are in the pass band. Conversely, the highest phase shifts (45° to 90°) occur in the *stop* bands (frequencies above low-pass cutoff and below high-pass cutoff).

In the low-pass case, the output of the filter lags the input (negative phase shift); in the high-pass case the output leads the input (positive phase shift). Figure 3 shows waveforms: an input sinewave signal (center trace), the output of a 1-kHz-cutoff single-pole high-pass filter (top trace), and the output of a 1-kHz-cutoff single-pole low-pass filter (bottom trace). The signal frequency is also 1 kHz—the cutoff frequency of both filters. The 45° lead and lag of the waveforms are clearly evident.

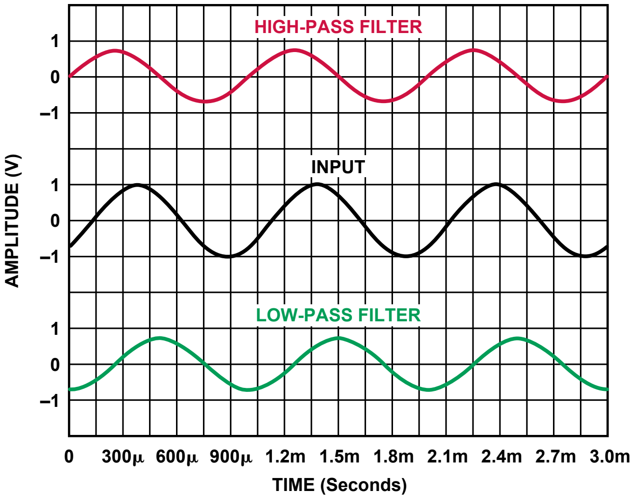


Figure 3. Input (center trace) and outputs of a single-pole high-pass filter (top trace) and low-pass filter (bottom trace).

For the second-order low-pass case, the transfer function's phase shift can be approximated by:

$$\phi(\omega) = -\tan^{-1}\left[\frac{1}{\alpha}\left[2\frac{\omega}{\omega_0} + \sqrt{4-\alpha^2}\right]\right] - \tan^{-1}\left[\frac{1}{\alpha}\left[2\frac{\omega}{\omega_0} - \sqrt{4-\alpha^2}\right]\right] \tag{3}$$

Figure 4 (left axis) evaluates this equation (using $\alpha = \sqrt{2} = 1.414$) from two decades below the center frequency to two decades above the center frequency. Here the center frequency is 1, with a phase shift of –90°.

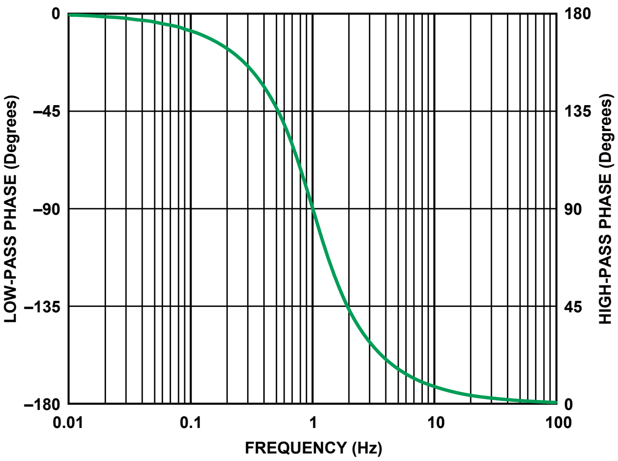


Figure 4. Phase response of a 2-pole low-pass filter (left axis) and high-pass filter (right axis) with a center frequency of 1.

In Equation 3, α, the *damping ratio* of the filter, is the inverse of Q (that is, Q = 1/α). It determines the peaking in the amplitude (and transient) response and the sharpness of the phase transition. An α of 1.414 characterizes a 2-pole Butterworth (maximally flat) response.

The phase response of a 2-pole high-pass filter can be approximated by:

$$\phi(\omega) = \pi - \tan^{-1}\left[\frac{1}{\alpha}\left[2\frac{\omega}{\omega_0} + \sqrt{4-\alpha^2}\right]\right] - \tan^{-1}\left[\frac{1}{\alpha}\left[2\frac{\omega}{\omega_0} - \sqrt{4-\alpha^2}\right]\right] \quad (4)$$

In Figure 4 (right axis), this equation is evaluated with α = 1.414 from two decades below the center frequency to two decades above the center frequency. At the center frequency (=1), the phase shift is 90°.

Figure 2 and Figure 4 use single curves because the high-pass and the low-pass phase responses are similar, just shifted by 90° and 180° (π/2 and π radians). This is equivalent to a change of the sign of the phase, causing the outputs of the low-pass filter to lag and the high-pass filter to lead.

In practice, a high-pass filter is really a wideband band-pass filter because the amplifier's response introduces at least a single low-pass pole.

Figure 5 shows the phase- and gain response of a 2-pole low-pass filter, plotted as a function of *Q*. The transfer function shows that phase change can spread over a fairly wide range of frequencies, and the range of the change varies inversely with the circuit's *Q*. While this article is primarily about phase response, the relationship between rate of change of phase and rate of change of amplitude is worth considering.

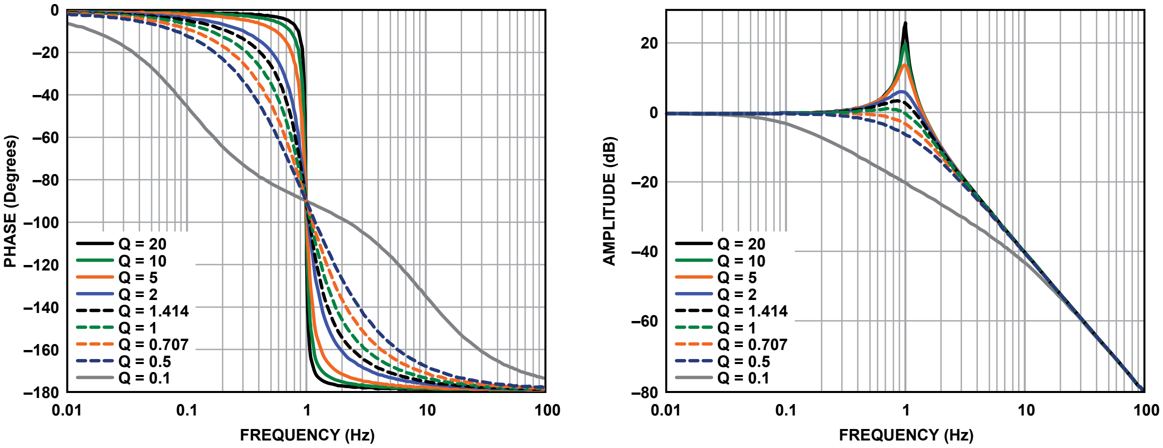


Figure 5. Phase and amplitude response of a 2-pole low-pass filter section as a function of Q.

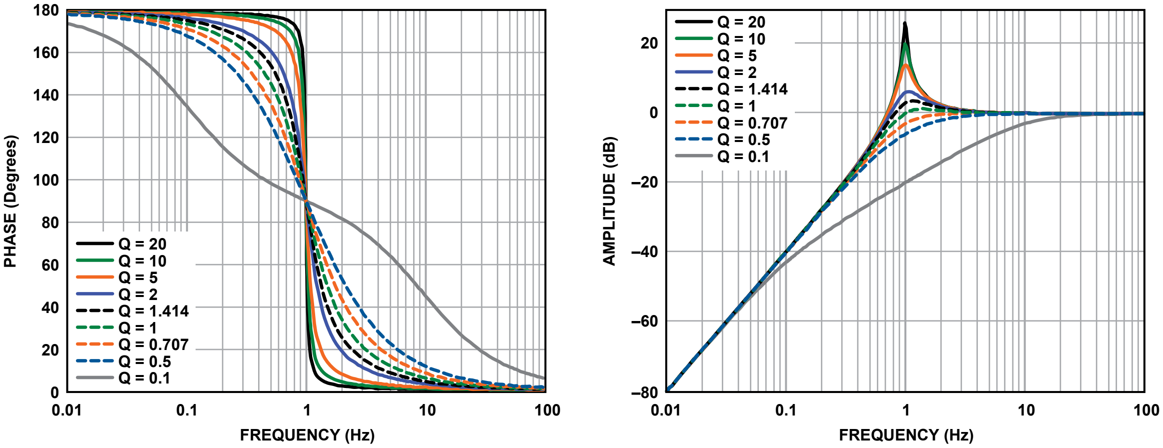


Figure 6. Phase and amplitude response of a 2-pole high-pass filter section as a function of Q.

Note that each 2-pole section provides a maximum 180° of phase shift; and, at the extremities, a phase shift of –180°, though lagging by 360°, is an angle with the same properties as a phase shift of 180°. For this reason, a multistage filter will often be graphed in a restricted range, say 180° to –180°, to improve the accuracy of reading the graph (see Figures 9 and 11). In such cases, it must be realized that the angle graphed is actually the true angle plus or minus $m \times 360°$. While in such cases there will appear to be a discontinuity at the top and bottom of the graph (as the plot transitions ±180°), the actual phase angle is changing smoothly and monotonically.

Figure 6 shows the gain- and phase response of a 2-pole high-pass filter with varying Q. The transfer function shows that the 180° of phase change can take place over a large frequency range, and the range of the change is inversely proportional to the Q of the circuit. Also note that the shapes of the curves are very similar. In particular, the phase responses have the same shape, just over a different range.

The Amplifier Transfer Function

The open-loop transfer function of the amplifier is basically that of a single-pole filter. If it is an inverting amplifier, it is in effect inserting 180° of additional phase shift. The closed-loop phase shift of the amplifier is generally ignored, but it can affect the overall transfer of the composite filter if its bandwidth is insufficient. The AD822 was chosen for the simulations of the filters in this article. It affects the composite filter transfer functions, but only at the higher frequencies, because its gain and phase shift are maintained up to considerably higher frequencies than the corner frequency of the filter itself. The open-loop transfer function of the AD822, from the data sheet, is shown in Figure 7.

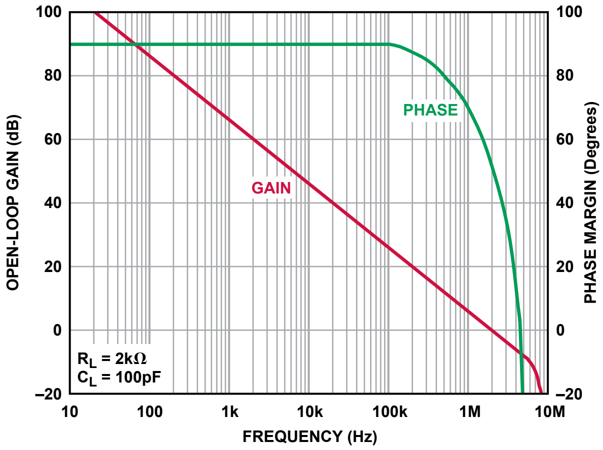


Figure 7. AD822 Bodé plot gain and phase.

Example 1: A 1-kHz, 5-Pole, 0.5-dB Chebyshev Low-Pass Filter

As an example, we will examine a 1-kHz, 5-pole, 0.5-dB Chebyshev low-pass filter. A few reasons for this specific choice:

1) Unlike the Butterworth case, the center frequencies of the individual sections are all different. This allows a graph that spreads out the traces a bit more, so the graph is a little more interesting.
2) The Q's are generally a bit higher.
3) An odd number of poles emphasizes the difference between single- and two-pole sections.

The filter sections were designed using the Filter Design Wizard, available on the Analog Devices website.

The f_0's and Q's of the sections follow:

f_{01} = 615.8 Hz	f_{02} = 960.8 Hz	f_{03}= 342 Hz
$Q1$ = 1.178	$Q2$ = 4.545	

Figure 8 shows the schematic of the complete filter. The filter topology chosen—*multiple feedback* (MFB)—was again arbitrary, as was the choice to make the single-pole section an active integrator rather than a simple buffered passive RC circuit.

Figure 9 shows phase shifts at each stage of the complete filter. The graph shows the phase shift of the first section alone (Section 1—blue), the first two sections (Sections 1 and 2—red) and the complete filter (Sections 1, 2, and 3—green). These include the basic phase shifts of the filter sections, the 180° contributed by each inverting amplifier, and the effects of amplifier frequency response on overall phase shift.

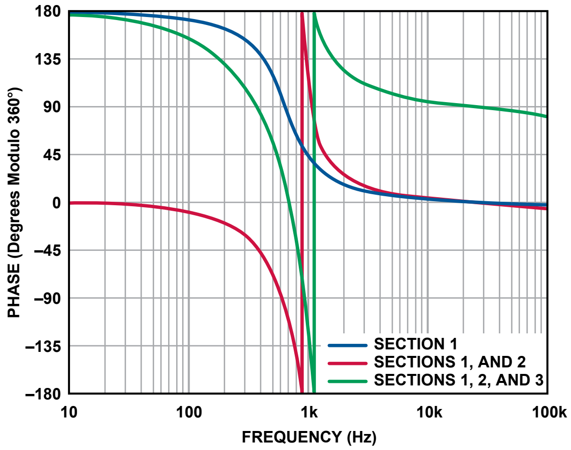


Figure 9. Phase response of the 1-kHz, 5-pole, 0.5 dB Chebyshev low-pass filter in Figure 8.

A few details of interest: First the phase response, being a net lag, accumulates negatively. The first 2-pole section starts with –180° (=180° *modulo* 360°) due to amplifier phase inversion at low frequencies, increasing to –360° (=0° *modulo* 360°) at high frequencies. The second section adds another phase inversion starting at –540° (=180° *modulo* 360°), and the phase increases to –720° (=0° modulo 360°) at high frequencies. The third section starts at –900° (=180° *modulo* 360°) at low frequencies and increases to –990° (=90° *modulo* 360°) at high frequencies. Also note that at the frequencies above 10 kHz the phase is rolling off slightly due to the amplifier's frequency response. That roll-off is seen to be cumulative, increasing for each section.

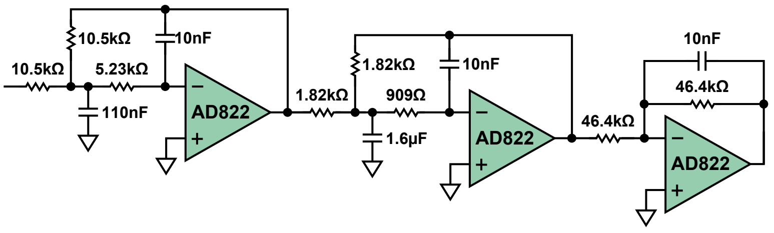


Figure 8. A 1-kHz, 5-pole, 0.5 dB Chebyshev low-pass filter.

Example 2: A 1-kHz, 5-Pole, 0.5-dB Chebyshev High-Pass Filter

The second example (see Figure 10) considers the phase response of a 1-kHz, 5-pole, 0.5-dB Chebyshev high-pass filter. In this case, the filter was designed (again using the Filter Design Wizard) with Sallen-Key *voltage-controlled voltage source* (VCVS) sections rather than multiple feedback (MFB). Though an arbitrary choice, VCVS requires only two capacitors per 2-pole section, rather than MFB's three capacitors per section, and the first two sections are noninverting.

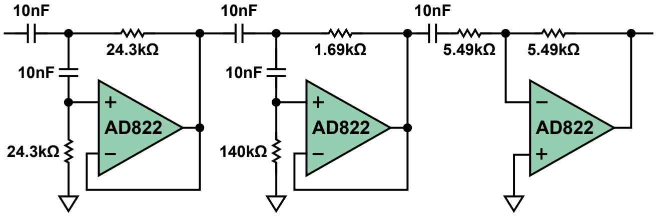


Figure 10. A 1-kHz, 5-pole, 0.5 dB Chebyshev high-pass filter.

Figure 11 shows the phase response at each section of the filter. The first section's phase shift starts at 180° at low frequencies, dropping to 0° at high frequencies. The second section, adding 180° at low frequencies, starts at 360° (=0° *modulo* 360°) and drops to 0° at high frequencies. The third section, adding a phase inversion, starts at –180° + 90° = 90° at low frequencies, dropping to –540° (= –180° *modulo* 360°). Note again the additional roll-off at high frequencies owing to amplifier frequency response.

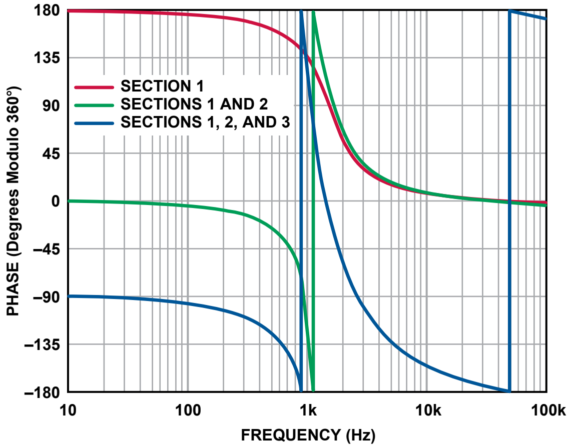


Figure 11. Phase response of the 1 kHz, 5-pole, 0.5 dB Chebyshev low-pass filter in Figure 8.

Conclusion

This article considers the phase shift of low-pass and high-pass filters. The previous article in this series examined the phase shift in relation to filter topology. In future articles, we will look at band-pass, notch, and all-pass filters—in the final installment, we will tie it all together and examine how the phase shift affects the transient response of the filter, looking at the group delay, impulse response, and step response.

Appendix

Generic operational equations for single- and two-pole low-pass and high-pass filters are given by equations A1 through A4.

The transfer function of a single-pole low-pass filter:

$$\frac{V_o}{V_{IN}} = \frac{\omega_0}{s+\omega_0} \qquad \text{(A1)}$$

where $s = j\omega$ and $\omega_0 = 2\pi f_0$.

The transfer function of a two-pole active low-pass filter:

$$\frac{V_o}{V_{IN}} = \frac{H_0 {\omega_0}^2}{s^2 + \frac{\omega_0}{Q}s + {\omega_0}^2} \qquad \text{(A2)}$$

where H_O is the section gain.

The transfer function of a single-pole high-pass filter:

$$\frac{V_o}{V_{IN}} = \frac{s}{s+\omega_0} \qquad \text{(A3)}$$

The transfer function of a two-pole active high-pass filter:

$$\frac{V_o}{V_{IN}} = \frac{H_o s^2}{s^2 + \frac{\omega_0}{Q}s + {\omega_0}^2} \qquad \text{(A4)}$$

The values of f_0 and Q for a 1-kHz, 0.5-dB Chebyshev low-pass filter:

Section	f_0	**Q**
Section 1	690.5	1.1779
Section 2	1017.8	4.5451
Section 3	362.3	———

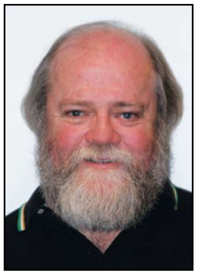
For a more detailed discussion, see References 6, 7, and 8.

References

1. Zumbahlen, H. "Phase Relations in Active Filters." *Analog Dialogue*. Volume 41, No. 4. October 2007.
2. Daryanani, G. *Principles of Active Network Synthesis and Design*. J. Wiley & Sons (1976). ISBN: 0-471-19545-6.
3. Graeme, J., G. Tobey, and L. Huelsman. *Operational Amplifiers—Design and Applications*. McGraw-Hill (1971). ISBN: 07-064917-0.
4. Van Valkenburg, M.E. *Analog Filter Design*. Holt, Rinehart & Winston (1982). ISBN: 0-03-059246-1
5. Williams, A.B. *Electronic Filter Design Handbook*. McGraw-Hill (1981).
6. Zumbahlen, H. "Analog Filters," Chapter 5 in Jung, W. *Op Amp Applications Handbook*. Newnes-Elsevier (2006). (Original chapter from ADI Seminar Notes is available online, www.analog.com/library/analogdialogue/archives/39-05/op_amp_applications_handbook.html.)
7. Zumbahlen, H. *Basic Linear Design*. Ch. 8. Analog Devices (2006). www.analog.com/library/analogdialogue/archives/43-09/linear_circuit_design_handbook.html.
8. Zumbahlen, H. *Linear Circuit Design Handbook*. Newnes-Elsevier (2008). ISBN: 978-0-7506-8703-4.

Author

Louis "Hank" Zumbahlen [hank.zumbahlen@analog.com] has worked at ADI since 1989, originally as a field applications engineer based in California. For the last several years he has been involved with training and seminar development as a senior staff applications engineer. Previously, he held a similar position at Signetics (Philips)—and positions as a design engineer at several companies, primarily in the test and measurement areas. Hank has a BSEE from the University of Illinois. He is the author of *Linear Circuit Design Handbook* (Newnes-Elsevier 2008).



Difference Amplifier Forms Heart of Precision Current Source

By Neil Zhao, Reem Malik, and Wenshuai Liao

Precision current sources provide a constant current in many applications, including industrial process control, instrumentation, medical equipment, and consumer products. For example, current sources are used to provide excitation for resistance-temperature detectors (RTDs) in process-control systems; to measure unknown resistors, capacitors, and diodes in digital multimeters; and to drive 4-mA to 20-mA current loops, which are widely used to transmit information over long distances.

Precision current sources have traditionally been built using op amps, resistors, and other discrete components—with limitations due to size, accuracy, and temperature drift. Now, high-precision, low-power, low-cost integrated difference amplifiers,[1] such as the AD8276, can be used to achieve smaller, higher performance current sources, as shown in Figure 1. The feedback buffer uses amplifiers with low offset and low bias current, such as the AD8538, AD8603, AD8605, AD8628, AD8655, AD8661, AD8663, OP177, or OP1177, depending on the required current range.

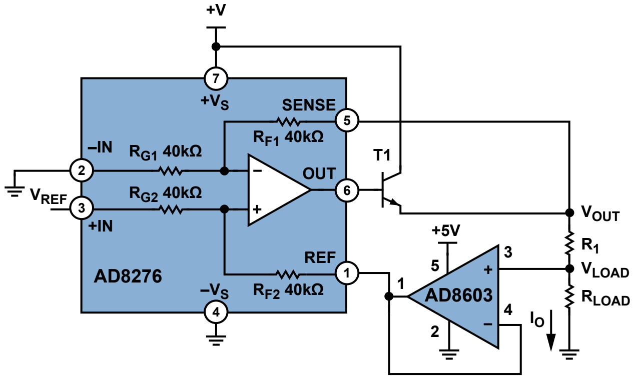


Figure 1. Difference amp and op amp form precision current source.

The output current can be calculated as follows:

$$I_O = V_{REF}\frac{\frac{R_{F2}}{R_{G2}}+\frac{R_{F1}}{R_{G1}}\times\frac{R_{F2}}{R_{G2}}}{R_1\left(1+\frac{R_{F2}}{R_{G2}}\right)+R_{LOAD}\left(\frac{R_{F2}}{R_{G2}}-\frac{R_{F1}}{R_{G1}}\right)} \quad (1)$$

If $R_{g1} = R_{g2} = R_{f1} = R_{f2}$, the equation can be reduced to:

$$I_O = \frac{V_{REF}}{R_1} \quad (2)$$

The maximum output current is limited by the op amp input range, difference amp output range, and difference amp SENSE pin voltage range. The following three conditions must be met:

$$V_{LOAD} = I_O \times R_{LOAD} \quad (3)$$

within op amp input range

$$V_{OUT} = I_O \times (R_{LOAD} + R_1) \quad (4)$$

within SENSE pin voltage range = 2×($-V_S$) – 0.2 V to 2×($+V_S$) – 3 V

$$I_O \times (R_{LOAD} + R_1) + V_{BE} \quad (5)$$

within AD8276 output voltage range = $-V_S$ + 0.2 V to $+V_S$ – 0.2 V

The SENSE pin can tolerate voltages almost twice as large as the supplies, so the second limitation will be very loose. The wide 2.5-V to 36-V supply range makes the AD8276 ideal for many applications. The maximum gain error of A- and B-grades is 0.05% and 0.02%, respectively, allowing current sources with up to 0.02% accuracy to be achieved.

Configuration Variations

For cost-sensitive applications that can tolerate a little more error, the circuit can be simplified by removing the feedback buffer, as shown in Figure 2.

With

$$R_{G1} = R_{G2} = R_{F1} = R_{F2} = R_F \quad (6)$$

the output current is

$$I_O = V_{REF}\frac{2R_F + R_1}{R_1(2R_F + R_{LOAD})} \quad (7)$$

where:

$$2R_F >> R_{LOAD} \text{ and } R_1 \quad (8)$$

and

$$I_O \approx \frac{V_{REF}}{R_1} \quad (9)$$

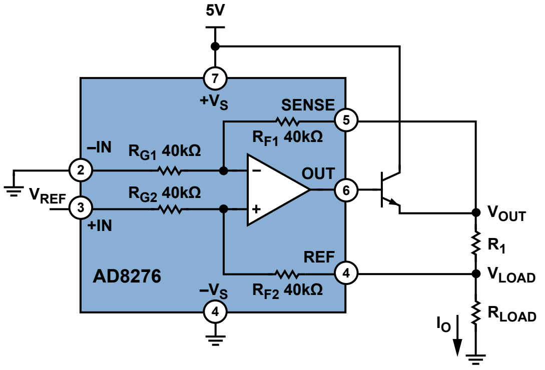


Figure 2. Simplified circuit eliminates feedback amplifier.

If the required output current is less than 15 mA—the output capability of AD8276—then the boost transistor can be eliminated, as shown in Figure 3. If both low current and reduced accuracy are acceptable, the simpler, lower-cost configuration of Figure 4 can be employed.

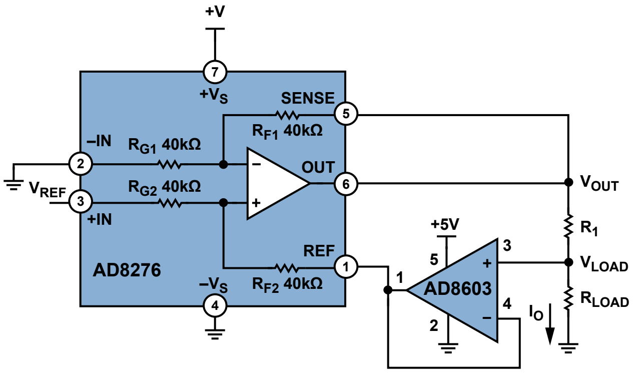


Figure 3. Simplified circuit for low-current applications.

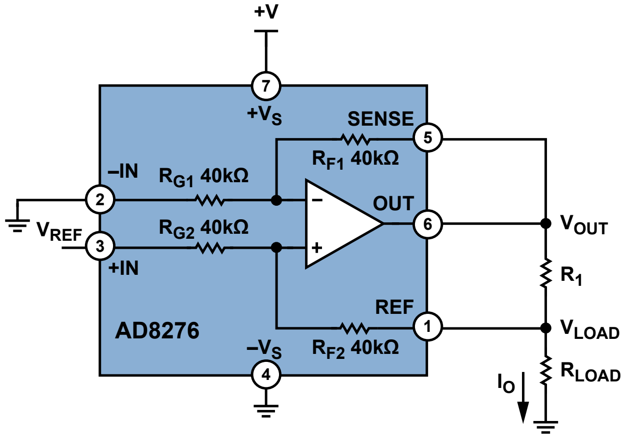


Figure 4. Simplified circuit for low-cost, low-current applications.

Figure 5 shows a topology that can be used for high-current, high-accuracy applications without the limitation of op amp input range.

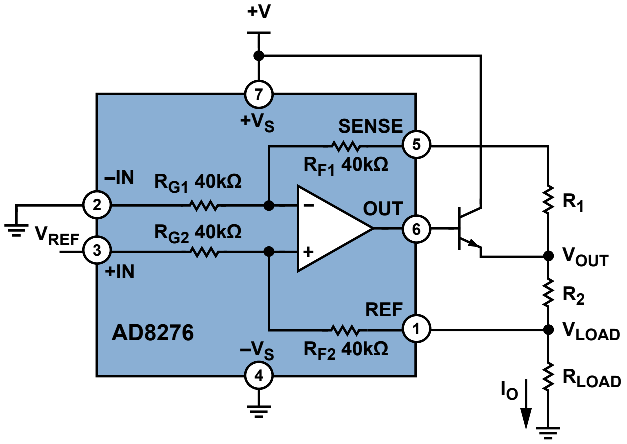


Figure 5. Difference amp and matched resistors form precision current source.

The output current can be calculated as:

$$I_O = V_{REF} \frac{R_{F2} R_{F1}\left(\frac{R_{G1}}{R_{F1}} + 1\right) + R_{F2} R_1 + R_{G1} R_2}{R_1 R_{G1} R_{G2}\left(1 + \frac{R_{F2}}{R_{G2}}\right) + R_{LOAD}\left(R_{G1} R_{F2} - R_{G2} R_{F1} + R_{G1} R_2 - R_{G2} R_1\right)} \quad (10)$$

If ideally matched, $R_{G1} = R_{G2} = R_{F1} = R_{F2} = 40\ k\Omega$ and $R_1 = R_2$, the output current is:

$$I_O = V_{REF}\left(\frac{1}{40\,k\Omega} + \frac{1}{R_2}\right) \quad (11)$$

External resistors R_1 and R_2 should have ultrahigh precision and matching, or the output current will vary with the load, an error that cannot be corrected with software.

Peripheral Components

The input voltage, V_{REF}, can be a DAC output, voltage reference, or transducer output. If a programmable current source is needed, precision 14- or 16-bit DACs, such as the AD5640, AD5660, AD5643R, and AD5663R, are recommended. For voltage references, the precision ADR42x and ADR44x are recommended for higher performance; the ADR36x is recommended for low power; the AD158x and ADR504x are recommended for low cost; and the ADR82x integrated op amp and voltage reference is recommended for small size.

The reference can connect to either the inverting or the noninverting input of the AD8276. If using the noninverting input, the common-mode voltage will be

$$\frac{V_{REF} + V_{LOAD}}{2} \quad (12)$$

and the output current will be

$$I_O = V_{REF} \frac{R_F + R_2}{R_F R_2} \quad (13)$$

If using the inverting input, the common-mode voltage will be

$$\frac{V_{LOAD}}{2} \quad (14)$$

and the output current will be

$$I_O = -V_{REF} \frac{R_F + R_2}{R_F R_2} \quad (15)$$

When using the inverting input, a buffer amplifier is required; the noninverting input is thus recommended for simplicity.

Transistor Selection

When selecting the boost transistor, make sure that V_C is higher than the power supply voltage and I_C is higher than the desired output current. Low-cost devices such as 2N3904, 2N4401, and 2N3391 are recommended. For lower current, the transistor is not needed.

Experimental Bench Results and Analysis

The input voltage vs. output current measured using the circuit of Figure 1 is shown in Figure 6. The AD8276 and AD8603 are powered by +5 V. The tolerance of R_1 is 0.1%. The transistor is a 2N3904. The reference was swept from 0.05 V to 1.20 V with 0.01-V steps. The input range is limited by the power supply and the AD8603 input range.

The maximum error is 0.87%, and the average is 0.10%. The current-sense error is limited by the external resistors. Higher accuracy resistors will produce higher accuracy current sources.

Conclusion

The AD8276 difference amplifier—with its low offset voltage, low offset voltage drift, low gain error, low gain drift, and integrated resistors—can be used to implement accurate, stable current sources. Its wide power supply range (2.5 V to 36 V) allows it to accommodate a wide range of loads. Its space-saving 8-lead MSOP package and its low power dissipation make it ideal for battery-powered and portable systems. Implementing a precision current source with a difference amplifier can reduce PCB area, simplify layout, decrease system cost, and improve reliability.

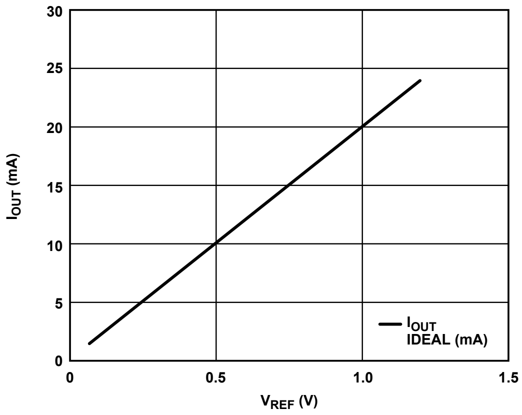


Figure 6. Test result using difference amplifier and feedback amplifier (see Figure 1).

References

(Information on all ADI components can be found at www.analog.com)

[1] www.analog.com/en/amplifiers-and-comparators/current-sense-amplifiers/products/index.html.

Authors

Neil Zhao [neil.zhao@analog.com] is a field applications engineer in ADI's China Applications Support Team, where he has been working for one and a half years. He is responsible for technical support for core analog products across China. Neil graduated in January 2008 from Beihang University with a master's degree in communication and information systems. He has published articles in *Analog Dialogue*, *EDN*, *Well Logging Technology*, and *Electronic Measurement Technology*.

Reem Malik [reem.malik@analog.com] is an applications engineer in ADI's Integrated Amplifier Products (IAP) Group in Wilmington, MA. She supports customers in the instrumentation, industrial, and medical areas, and is responsible for products such as difference amplifiers and variable gain amplifiers. Reem holds BSEE and MSEE degrees, both from Worcester Polytechnic Institute, earned in 2003 and 2008, respectively. She joined Analog Devices in June 2008.

Wenshuai Liao [wenshuai.liao@analog.com] is a marketing engineer in ADI's Integrated Amplifier Products (IAP) Group in Beijing, China. After earning a master's degree in optical engineering from Tsinghua University, Wenshuai spent three years as a TD-SCDMA Node B RF engineer at Datang Telecommunications Group. He joined ADI in 2002.

Difference Amplifiers

Model	Common-Mode Range (V)	Bandwidth (MHz typ)	CMRR (dB)	Gain Range	Supply Voltage (V_{MIN})	Supply Voltage (V_{MAX})	Supply Current (mA)	V_{OS} TC (µV/°C)	Gain TC (ppm/°C)	Temperature Range	Package	Price (1000s)
AD8270	$-V_S$ to $+V_S$	10	98	1.5	+5	+36	2.5	1.5	1	–40 to +125	LFCSP	$1.91
AD8271	$-V_S - 0.4$ to $+V_S + 0.4$	15	80	1.5	+2.5	+36	2.6	2	2	–40 to +85	MSOP	$1.25
AD8273	±40	20	86	1.5	+5	+36	2.5	3	2	–40 to +125	SOIC	$1.67
AD8274	±3	10	86	1.5	+5	+36	2.6	3	0.5	–40 to +85	MSOP, SOIC	$1.05
AD8275	–13 to +24	15	96	0.2	+3.3	+15	2.3	2.5	0.3	–40 to +85	MSOP	$1.60
AD8276	$2(-V_S) + 0.2$ to $2(+V_S) - 3$	0.55	86	1	+2.5	+36	0.22	2	1	–40 to +125	MSOP, SOIC	$1.00

Analog Devices, Inc.
Worldwide Headquarters
Analog Devices, Inc.
Three Technology Way
P.O. Box 9106
Norwood, MA 02062-9106
U.S.A.
Tel: 781.329.4700
(800.262.5643,
U.S.A. only)
Fax: 781.461.3113

Analog Devices, Inc.
Europe Headquarters
Analog Devices, Inc.
Wilhelm-Wagenfeld-Str. 6
80807 Munich
Germany
Tel: 49.89.76903.0
Fax: 49.89.76903.157

Analog Devices, Inc.
Japan Headquarters
Analog Devices, KK
New Pier Takeshiba
South Tower Building
1-16-1 Kaigan, Minato-ku,
Tokyo, 105-6891
Japan
Tel: 813.5402.8200
Fax: 813.5402.1064

Analog Devices, Inc.
Southeast Asia
Headquarters
Analog Devices
22/F One Corporate Avenue
222 Hu Bin Road
Shanghai, 200021
China
Tel: 86.21.2320.8000
Fax: 86.21.2320.8222

www.analog.com/analogdialogue

ANALOG
DEVICES