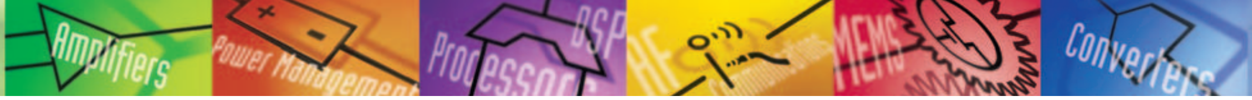
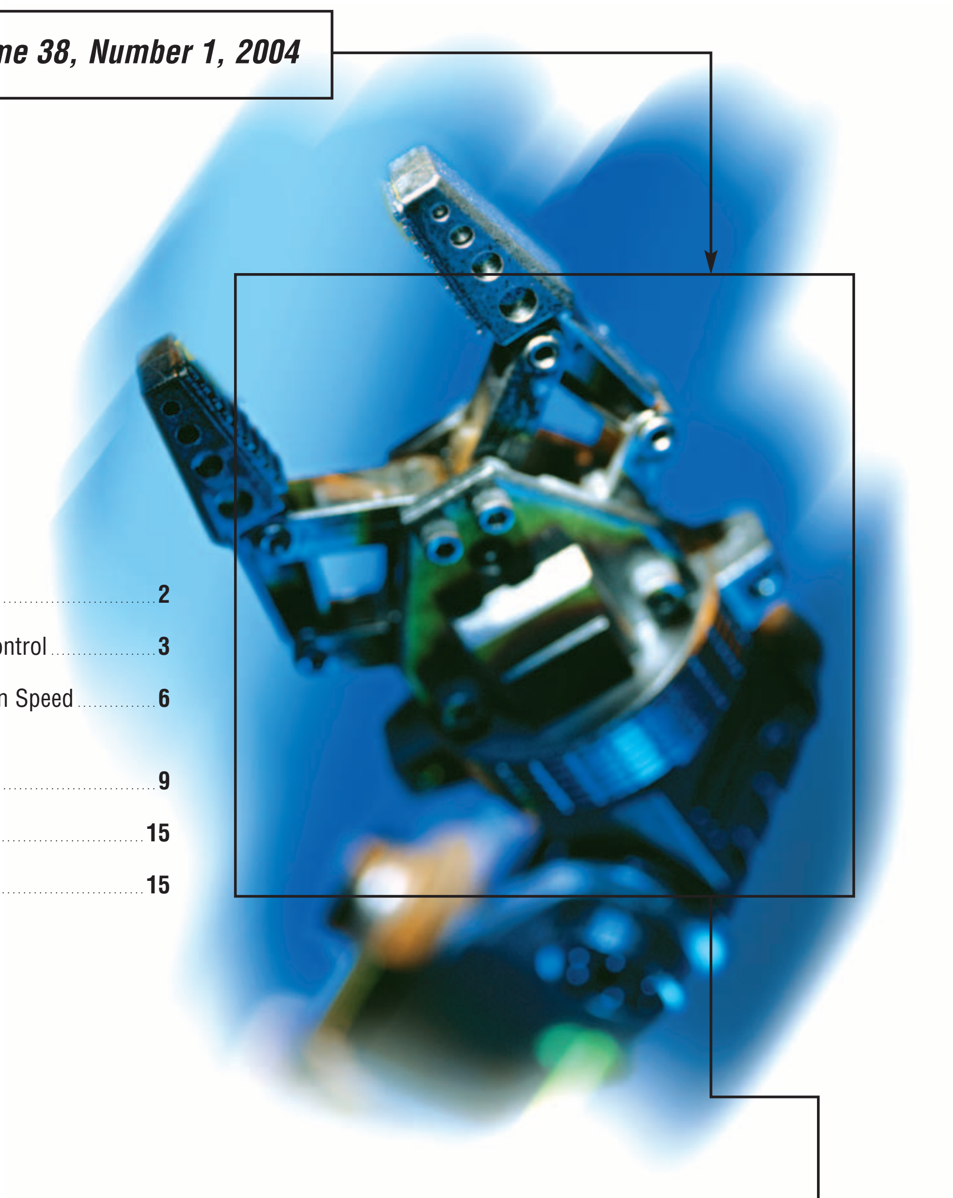


Analog Dialogue

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

Volume 38, Number 1, 2004



In This Issue

ANALOG DEVICES
www.analog.com/analogdialogue

Editors' Notes

MULTIPLIERS

Some 33 years ago, we noted the introduction of the AD530—the world's first complete analog multiplier-on-a-chip—with a rambling historical discourse appropriately dubbed, "Multiplier Memories and Meanderings." Eighteen years later, with many successful generations of multipliers (and a growing family of other translinear devices) in our portfolio, you could have read a column entitled, "Multiplier Meanderings—Revisited," as we feted the arrival of the 500-MHz AD834. Now—and for no overriding reason, other than an implied promise buried in the last issue, we continue to meander and furnish the newer generation of readers with thoughts about multipliers and related devices originating farther up the stream of time, as well as to jog the memories of those readers whose ripening with time may have paralleled our own.

An analog *multiplier* is in a special category of nonlinear electronic devices, since it can represent either an externally applied linearly adjustable (modulating) influence, or a means of introducing a parabolic (i.e., 2nd degree) function for analog computation—and can be connected in a feedback loop to perform division.

Although our earliest and ongoing primacy is in *analog* multiplier design, we should note that, as the world's *house of multipliers*, Analog Devices is no stranger to IC electronic multipliers of digital and hybrid ("mixed-signal" in today's lingo) provenance. One of our first DSP products was a fast *digital* multiplier in CMOS (1983); and our very first 10-bit CMOS D/A converter, the AD7520 (1974), was a *multiplying* DAC.

In the vacuum-tube years, well before silicon and the Gilbert approach, the design of analog multipliers with adequate linearity and bandwidth posed daunting challenges. Servos were accurate, but too slow for repetitive computation (and they had mechanical parts that could wear out). Linear modulator circuits were available, but the gain and biasing required to operate at the standard voltage levels and polarities needed for computing made them impractical for general-purpose applications. For a while, the elegant *quarter-square** relationship seemed to offer the method of choice, since it offered a direct, symmetrical, mathematical solution—if one could only provide two accurately matched, stable squaring circuits and could accompany them with instantly responding accurate sums, differences, and coefficients at zero drift! This concept, elegant as it was, merely lacked accuracy in practice. As you may surmise, analog multipliers of that era were large and clumsy, hot, expensive, imprecise, and usually of undependable fidelity.

The hottest applications of ICs that involve multiplication (and its logarithmic cousins) are currently in RF (up to 8 GHz), where one finds modulators, demodulators, log amps, mixers, power detectors, rms-to-dc converters, AGC, AFC, VGA, and gain- and phase measurement. The strictures of our budget permit little discussion, in this cramped space, of the concepts and applications of multiplication and translinear circuits. The wisest use of the remaining space may be to provide you with pointers to some of the material we've published in these pages that you can find either online in our Archives[1], elsewhere in a search of the *www.analog.com* website, or in your library:

"Accurate Gain/Phase Measurement at Radio Frequencies up to 2.5 GHz," by John Cowles and Barrie Gilbert. *Analog Dialogue* 35 (2001), pp. 5-8. Archives: find it in Volume 35, 2001.

"Accurate, Low-Cost, Easy-to-Use Multiplier," by Barrie Gilbert. *Analog Dialogue* 11-1 (1977). Archives: find it in (PDF) *The Best of Analog Dialogue*, 1967-1991.

"Complete Monolithic-Multifunction Chip," by Lew Counts, Charles Kitchin, and Steve Sherman. *Analog Dialogue* 19-1 (1985). Archives: find it in (PDF) *The Best of Analog Dialogue*, 1967-1991.

"Monolithic IC RMS-to-DC Converter," by Lew Counts, Barrie Gilbert, and Dave Kress. *Analog Dialogue* 11-2 (1977). Archives: find it in (PDF) *The Best of Analog Dialogue*, 1967-1991.

"Nonlinear Circuits Handbook," edited by D. H. Sheingold. Norwood, MA: Analog Devices, Inc., 1976 (out of print).

"Now—True RMS-to-DC Measurements, From Low Frequencies to 2.5 GHz." *Analog Dialogue* 34 (2000), p. 45. Archives: find it in Volume 34, 2000.

"X-Amp™, A New 45-dB, 500-MHz Variable-Gain Amplifier (VGA) Simplifies Adaptive Receiver Designs," by Eric J. Newman. *Analog Dialogue* 36, Part 1 (2002), pp. 3-5. Archives: find it in Volume 36, Part 1 (Jan-Jun), 2002.

Dan Sheingold [dan.sheingold@analog.com]

* $z = \frac{1}{4}\left[(x+y)^2 - (x-y)^2\right] = x \cdot y$

[1] http://www.analog.com/library/analogDialogue/archives.html

We would like to thank the more than 4000 readers who responded to our recent online survey. Responses came from 48 states in the US and 66 countries all over the world, with 96% of our readers identifying themselves as designers. We learned that, despite the popularity and timeliness of the Internet: 54% subscribe to the print edition, 43% to the online edition; 61% save their print copies, 21% give it to a colleague. Among online viewers, 57% print online articles, and 12% forward their eNewsletters to a colleague. Most readers told us that they prefer applications articles and tutorials over product articles.

When asked how to improve *Analog Dialogue* or make it more useful, most respondents said that it was fine as is, that they wanted more of the same, or that they wanted to subscribe. Some asked for better search capability. As a result, readers can now use Google™ to search the *Analog Dialogue* website from the Search page or from the navigation bar on most of the main pages. Readers asked for foreign language versions, and we are happy to announce that *Analog Dialogue* is now available in Chinese. Readers also asked for a CD-ROM version of *Analog Dialogue*, so we are considering issuing one as a 40th anniversary bonus, coming up in early 2006.

As always, we value the opinions of our faithful readers, so please send us feedback when you like something that you read, or when you would prefer to see something else instead (or in addition).

Scott Wayne [scott.wayne@analog.com]

Analog Dialogue

www.analog.com/analogdialogue **dialogue.editor@analog.com**

Analog Dialogue is the free technical magazine of Analog Devices, Inc., published continuously for 38 years—starting in 1967. It discusses products, applications, technology, and techniques for analog, digital, and mixed-signal processing. It is currently published in two editions—*online*, monthly at the above URL, and quarterly *in print*, as periodic retrospective collections of articles that have appeared online. In addition to technical articles, the online edition has timely announcements, linking to data sheets of newly released and pre-release products, and "Potpourri"—a universe of links to important and rapidly proliferating sources of relevant information and activity on the Analog Devices website and elsewhere. The *Analog Dialogue* site is, in effect, a "high-pass-filtered" point of entry to the **www.analog.com** site—the virtual world of *Analog Devices*. In addition to all its current information, the *Analog Dialogue* site has archives with all recent editions, starting from Volume 29, Number 2 (1995), plus three special anniversary issues, containing useful articles extracted from earlier editions, going all the way back to Volume 1, Number 1.

If you wish to subscribe to—or receive copies of—the print edition, please go to **www.analog.com/analogdialogue** and click on **<subscribe>**. Your comments are always welcome; please send messages to **dialogue.editor@analog.com** or to these individuals: *Dan Sheingold*, Editor **[dan.sheingold@analog.com]** or *Scott Wayne*, Managing Editor and Publisher **[scott.wayne@analog.com]**.

Mixed-Signal Control Circuits Use Microcontroller for Flexibility in Implementing PID Algorithms

By Eamon Neary [eamon.neary@analog.com]

INTRODUCTION

When a process is controlled (Figure 1), a characteristic of the process, such as a temperature (*regulated variable*), is compared with the desired value, or *setpoint*. The difference, or *error signal*, $e(t)$, is applied to a *controller*, which uses the error signal to produce a *control signal*, $u(t)$, that manipulates a physical input to the process (*manipulated variable*), causing a change in the regulated variable that will stably reduce the error.

A commonly used control operator is a *proportional-integral-derivative* (P-I-D, or PID) controller. It sums three terms derived from the error: a simple gain, or *proportional term*; a term proportional to the integral of the error, or *integral term*; and a term proportional to the rate of change of the error signal, or *derivative term*. In the closed loop, the proportional term seeks to reduce the error in proportion to its instantaneous value; the integral term—accumulating error—slowly drives the error towards zero (and its stored error tends to drive it beyond zero); and the derivative term uses the rate of change of error to anticipate its future value, speeding up the response to the proportional term and tending to improve loop stability by compensating for the integral term's lag.

The combination of these terms can provide very accurate and stable control. But the control terms must be individually adjusted or "tuned" for optimum behavior in a particular system. Because processes with many lags or substantially delayed response are hard to control, a simple PID controller is best used for processes that react readily to changes in the manipulated variable (which often controls the amount or rate-of-flow of energy added to the process). PID control is useful in systems where the load is continually varying and the controller is expected to respond automatically to frequent changes in setpoint—or deviations of the regulated variable (due to changes in ambient conditions and loading).

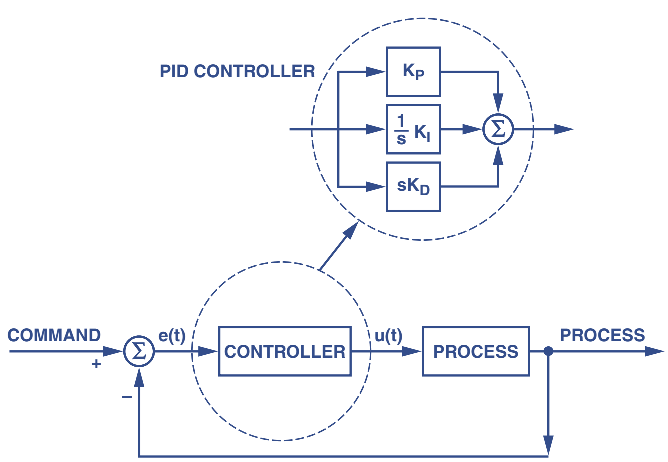


Figure 1. Control loop employing a PID control function.

The parameters of PID controllers for slow processes are usually obtained initially by working with system models scaled up in speed. There are many advanced control strategies, but the great majority of industrial control systems use PID controllers because they are standard, time-tested, well understood industrial components. Moreover, due to process uncertainties, a more-sophisticated control scheme is not necessarily more efficient than a well-tuned PID controller for a given process.

The PID terms were briefly explained above. Here is a more complete explanation of them.

Proportional Control

Proportional control applies a corrective term proportional to the error. The proportionality constant (K_P) is known as the *proportional gain* of the controller. As the gain is increased, the system responds faster to changes in setpoint, and the final (steady-state) error is smaller, but the system becomes less stable, because it is increasingly under-damped. Further increases in gain will result in overshoots, ringing, and ultimately, undamped oscillation.

Integral Control

Although proportional control can reduce error substantially, it cannot by itself reduce the error to zero. The error can, however, be reduced to zero by adding an *integral* term to the control function. An integrator in a closed loop must seek to hold its average input at zero (otherwise, its output would increase indefinitely, ending up in saturation or worse). The higher the integral gain constant, K_I, the sooner the error heads for zero (and beyond) in response to a change; so to set K_I too high is to invite oscillation and instability.

Derivative Control

Adding a *derivative* term—proportional to the time derivative, or rate-of-change, of the error signal—can improve the stability, reduce the overshoot that arises when proportional and/or integral terms are used at high gain, and improve response speed by anticipating changes in the error. Its gain, or the "damping constant," K_D, can usually be adjusted to achieve a critically damped response to changes in the setpoint or the regulated variable. Too little damping, and the overshoot from proportional control may remain; too much damping may cause an unnecessarily slow response. The designer should also note that differentiators amplify high frequency noise appearing in the error signal.

In summary, a proportional controller (P) will reduce the rise time and will reduce, but never eliminate, the steady state error. A proportional-integral (PI) controller will eliminate the steady state error, but it may make the transient response worse. A proportional-integral-derivative controller (PID) will increase the system stability, reduce the overshoot, and improve the transient response. Effects of increasing a given term in a closed-loop system are summarized in Table I.

Table I.

Gain Constant	Rise Time	Overshoot	Settling Time	Steady State Error
K_P	Decrease	Increase	Little Change	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	Little Change	Decrease	Decrease	Little Change

The sum of the three terms is

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}$$

The corresponding operational transfer function is:

$$K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s}$$

In the system of Figure 1, the difference between the setpoint value and the actual output is represented by the error signal $e(t)$. The error signal is applied to a PID controller, which computes the derivative and the integral of this error signal, applies the three coefficients, and performs the above summation to form the signal, $u(t)$.

Digital PID Control

The PID algorithm, now widely used in industrial process control, has been recognized and employed for nearly a century, originally in pneumatic controllers. Electronics—first used to model PID controls in control-system design with analog computers in the 1940s and '50s—became increasingly involved in actual process-control loops, first as analog controllers, and later as digital controllers. Software implementation of the PID algorithm with 8-bit microcontrollers is well documented.

In this article we show the basic components of a digital PID controller—and then show how process control can be implemented economically using a MicroConverter®, a data-acquisition system on a chip.

One might consider the use of a PID loop, for example, in an air conditioning or refrigeration system to accurately maintain temperature in a narrow range, using continuous monitoring and control (as opposed to thermostatic on-off control). Figure 2 shows a basic block diagram of a control system that regulates temperature by continuously adjusting fan speed, increasing or decreasing the airflow from a low-temperature source.

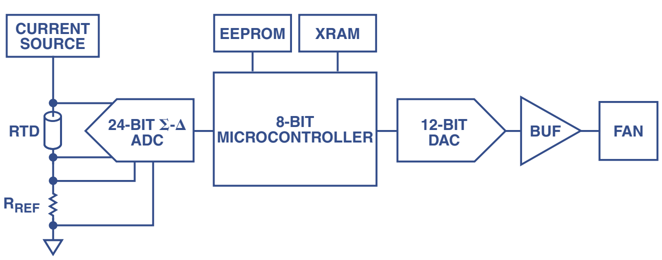


Figure 2. Example of a PID controller for a temperature-controlled ventilation system using discrete components.

The system is required to maintain the room temperature as close as possible to the user-selected (setpoint) value. To do this, the system must accurately measure the room temperature and adjust the fan speed to compensate.

In the system shown in Figure 2, a precision current source drives a current through a resistive temperature sensor—a thermistor or RTD—in series with a reference resistor, adjusted to represent the desired temperature. The analog-to-digital converter (ADC) digitizes the difference between the reference voltage and the thermistor voltage as a measure of the temperature error. An 8-bit microcontroller is used to process the ADC results, and to implement the PID controller. The microcontroller adjusts the fan speed, driving it via the digital-to-analog converter (DAC). External program memory and RAM are required to operate the 8-bit microcontroller and execute the program.

If proportional control (P) on its own were used, the rate at which the fans run would be directly related to the temperature difference from the setpoint. As mentioned earlier, this will leave a steady state error in place.

Adding an integral term (PI) results in the fan speed rising or falling with the ambient temperature. It adjusts the room temperature to compensate for errors due to the ambient rise in temperature through the day and then the temperature fall in the evening. The integral term thus removes the offset, but if the integral gain is too high, oscillation about the setpoint can be introduced. (Note that oscillation is inherent in temperature-control systems employing on-off thermostats.)

This oscillatory tendency can be greatly reduced by adding in a derivative term (PID). The derivative term responds to the rate of change of the error from the setpoint. It helps the system rapidly correct for sudden changes due to a door or window being opened momentarily.

To simplify this system, minimizing parts cost, assembly cost, and board area, an integrated system-on-a-chip (SOC) solution can be used, as shown in Figure 3.

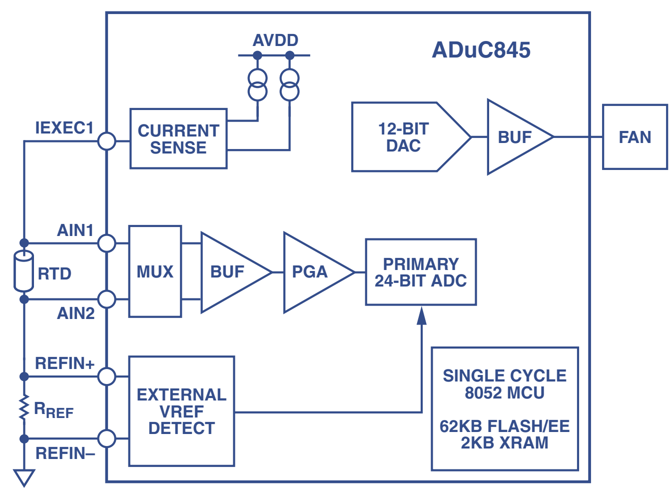


Figure 3. System-on-a-chip implementation.

The ADuC845 MicroConverter includes 62K bytes of flash/EE program memory, 4K bytes of flash data memory, and 2K bytes of RAM. The flash data memory can be used to store the coefficients for a 'tuned' PID loop, while the single-cycle core provides enough processing power to simultaneously implement the PID loop and perform general tasks.

Depending on which MicroConverter is selected, the resolution of the ADC ranges from 12 to 24 bits. In a system where the temperature needs to be maintained to 0.1°C accuracy, the ADuC845's high performance 24-bit sigma-delta ADC is ideal.

A second type of application where a PID control loop is useful is setpoint (servo) motor control. In this application the motor is required to move to, maintain, and follow an angular position defined by a user input (for example, the rotation of a potentiometer—Figure 4).

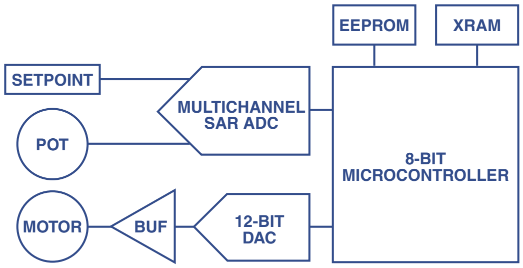


Figure 4. Example of a motor control system embodied with discrete components.

Again, this system can be implemented using many discrete components or, more simply, with an integrated solution. Figure 5 shows a demonstration system built using the MicroConverter. The circuitry on the board causes the pointer to follow the rotation of the setpoint input potentiometer.

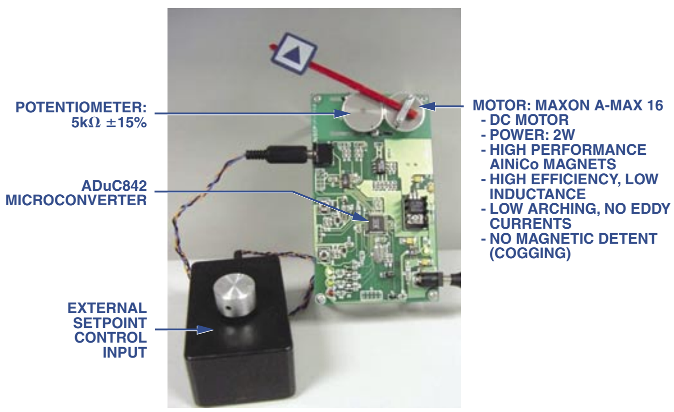


Figure 5. Sample motor control system using discrete components.

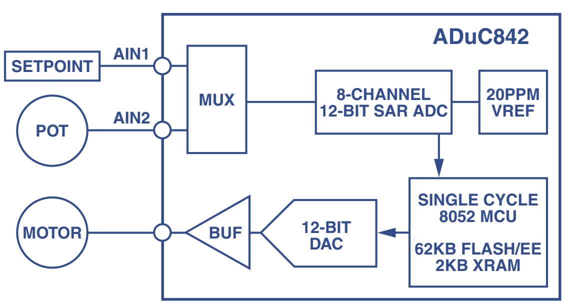


Figure 6. System-on-a-chip implementation of Figure 4.

With the blocks integrated in the compact form of the ADuC842, parts and assembly costs are lower; the computational electronics occupies considerably less space and is more reliable. Figure 6 shows the simplicity of the system hardware using the SOC approach.

Besides the ADuC842, the board includes a potentiometer buffer amplifier, an output power amplifier that drives the motor, a 5-V low-noise regulator for the low-power electronics, and a huskier 5-V regulator (with heatsink) for the motor. The board also includes status LEDs, a RESET button, a serial-data download button, and some passive elements.

Using PC software to simulate the rest of the system, Figure 7 shows responses for different levels of system tuning, and demonstrates the importance of the integral term.

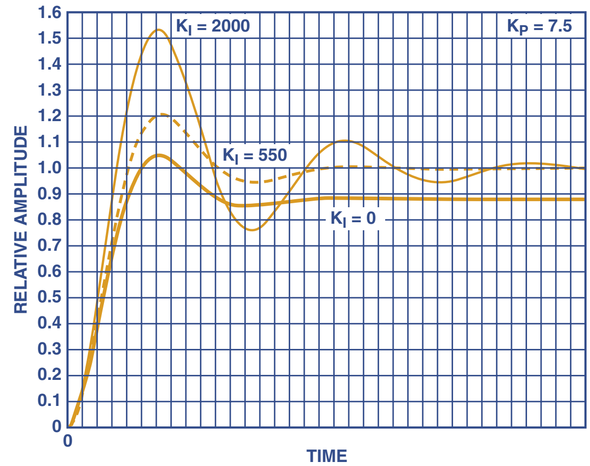


Figure 7. Proportional-integral (P-I) control for three settings of the integral term. Note the offset from 1.0 for $K_I = 0$, the lightly damped oscillatory tendency for $K_I = 2000$, with oscillations almost eliminated at $K_I = 550$.

The improvement in overall system step response when implemented with the full PID loop is clearly shown in Figure 8. The response is fast, accurate, and damped, with no offset, oscillation, or overshoot.

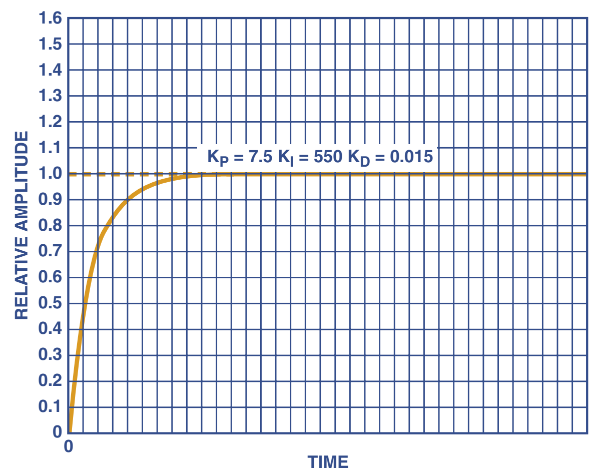


Figure 8. Proportional-integral-derivative (PID) control response.

This article can be found at http://www.analog.com/library/analogDialogue/archives/38-01/pid.html

Why and How to Control Fan Speed for Cooling Electronic Equipment

By Mary Burke [mary.burke@analog.com]

INTRODUCTION

Interest has been growing in integrated circuits for controlling the speed of cooling fans in personal computers and other electronic equipment. Compact electrical fans are cheap and have been used for cooling electronic equipment for more than half a century. However, in recent years, the technology of using these fans has evolved significantly. This article will describe how and why this evolution has taken place and will suggest some useful approaches for the designer.

Heat Generation and Removal

The trend in electronics, particularly consumer electronics, is towards smaller products with enhanced combinations of features. Consequently, lots of electronic components are being shoehorned into very small form factors. An obvious example is the notebook PC. Thin and "Lite," notebook PCs have shrunk significantly, yet their processing power has been maintained or increased. Other examples of this trend include projection systems and set-top boxes. What these systems all have in common, besides significantly smaller—and still decreasing—size, is that the amount of heat they must dissipate does not decrease; often it increases! In the notebook PC, much of the heat is generated by the processor; in the projector, most of the heat is generated by the light source. This heat needs to be removed quietly and efficiently.

The quietest way to remove heat is with passive components such as heat sinks and heat pipes. However, these have proved insufficient in many popular consumer electronics products—and they are also somewhat expensive. A good alternative is active cooling, introducing a fan into the system to generate airflow around the chassis and the heat-generating components, efficiently removing heat from the system. A fan is a source of noise, however. It is also an additional source of power consumption in the system—a very important consideration if power is to be supplied by a battery. The fan is also one more mechanical component in the system, not an ideal solution from a reliability standpoint.

Speed control—one way to answer some of these objections to the use of a fan—can have these advantages:

1. running a fan slower reduces the noise it emits,
2. running a fan slower can reduce the power it consumes,
3. running a fan slower increases its reliability and lifetime.

There are many different types of fans and ways of controlling them. We will discuss here various fan types and the advantages and disadvantages of control methods in use today. One way to classify fans is as:

1. 2-wire fans
2. 3-wire fans
3. 4-wire fans.

The methods of fan control to be discussed here include:

1. no fan control
2. on/off control
3. linear (continuous dc) control
4. low-frequency pulse-width modulation (PWM)
5. high-frequency fan control.

Fan Types

A *2-wire* fan has power and ground terminals. A *3-wire fan* has power, ground, and a *tachometric* ("tach") output, which provides a signal with frequency proportional to speed. A *4-wire fan* has power, ground, a tach output, and a PWM-drive input. PWM, in brief, uses the relative width of pulses in a train of on-off pulses to adjust the level of power applied to the motor.

A 2-wire fan is controlled by adjusting either the dc voltage or pulse width in low-frequency PWM. However, with only two wires, a tach signal is not readily available. This means that there is no indication as to how fast the fan is running—or indeed, if it is running at all. This form of speed control is *open-loop.*

A *3-wire* fan can be controlled using the same kind of drive as for 2-wire fans—variable dc or low-frequency PWM. The difference between 2-wire fans and 3-wire fans is the availability of *feedback* from the fan for closed-loop speed control. The tach signal indicates whether the fan is running and its rate of speed.

The tach signal, when driven by a dc voltage, has a square-wave output closely resembling the "ideal tach" in Figure 1. It is always valid, since power is continuously applied to the fan. With low-frequency PWM, however, the tach signal is valid only when power is applied to the fan—that is, during the *on* phase of the pulse. When the PWM drive is switched to the *off* phase, the fan's internal tach signal-generation circuitry is also off. Because the tach output is typically from an open drain, it will float high when the PWM drive is *off,* as shown in Figure 1. Thus, while the ideal tach is representative of the actual speed of the fan, the PWM drive in effect "chops" the tach signal output and may produce erroneous readings.

In order to be sure of a correct fan speed reading under PWM control, it is necessary to periodically switch the fan *on* long enough to get a complete tach cycle. This feature is implemented in a number of Analog Devices fan controllers, such as the ADM1031 and the ADT7460.

In addition to the power, ground, and tach signal, *4-wire fans* have a PWM input, which is used to control the speed of the fan. Instead of switching the power to the entire fan *on* and *off,* only the power to the drive coils is switched, making the tach information available continuously. Switching the coils on and off generates some *commutation noise.* Driving the coils at rates greater than 20 kHz moves the noise outside of the audible range, so typical PWM fan-drive signals use a rather high frequency (>20 kHz). Another advantage of 4-wire fans is that the fan speed can be controlled at speeds as low as 10% of the fan's full speed. Figure 2 shows the differences between 3-wire and 4-wire fan circuits.

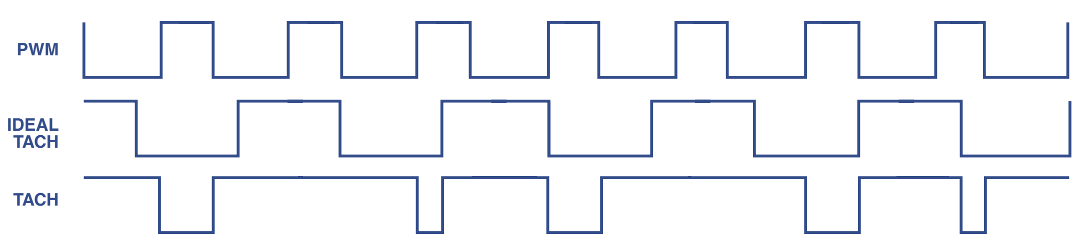


Figure 1. Tachometer-output waveforms in 3-wire fans—ideal, and under PWM control.

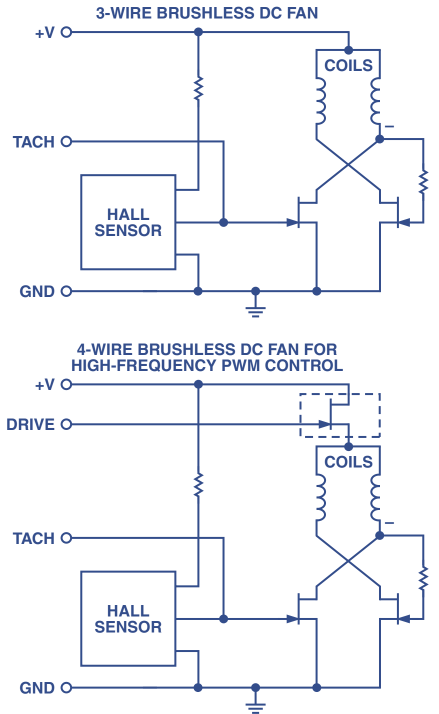


Figure 2. 3- and 4-wire fans.

Fan Control

No control: The simplest method of fan control is not to use any at all; just run a fan of appropriate capacity at full speed 100% of the time. The main advantages of this are guaranteed fail-safe cooling and a very simple external circuit. However, because the fan is always switched on, its lifetime is reduced and it uses a constant amount of power—even when cooling is not needed. Also, its incessant noise is likely to be annoying.

On/off control: The next simplest method of fan control is thermostatic, or *on/off control*. This method is also very easy to implement. The fan is switched on only when cooling is needed, and it is switched off for the remainder of the time. The user needs to set the conditions under which cooling is needed—typically when the temperature exceeds a preset threshold.

The Analog Devices ADM1032 is an ideal sensor for on/off fan control using a temperature setpoint. It has a comparator that produces a THERM output—one that is normally *high* but switches *low* when the temperature exceeds a programmable threshold. It automatically switches back to *high* when the temperature drops a preset amount below the THERM Limit. The advantage of this programmable *hysteresis* is that the fan does not continually switch on/off when the temperature is close to the threshold. Figure 3 is an example of a circuit using the ADM1032.

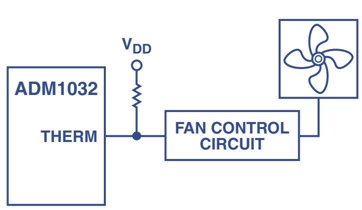


Figure 3. Example of an on/off control circuit.

The disadvantage of on/off control is that it is very limited. When a fan is switched *on*, it immediately spins up to its full speed in an audible and annoying manner. Because humans soon become somewhat accustomed to the sound of the fan, its switching *off* is also very noticeable. (It can be compared to the refrigerator in your kitchen. You didn't notice the noise it was making until it switched off.) So from an acoustic perspective, on/off control is far from optimal.

Linear control: At the next level of fan control, *linear control*, the voltage applied to the fan is variable. For lower speed (less cooling and quieter operation) the voltage is decreased, and for higher speed it is increased. The relationship has limitations. Consider, for example, a 12-V fan (rated maximum voltage). Such a fan may require at least 7 V to start spinning. When it does start spinning, it will probably spin at about half its full speed with 7 V applied. Because of the need to overcome inertia, the voltage required to start a fan is higher than the voltage required to keep it spinning. So as the voltage applied to the fan is reduced, it may spin at slower speeds until, say, 4 V, at which point it will stall. These values will differ, from manufacturer to manufacturer, from model to model, and even from fan to fan.

The Analog Devices ADM1028 linear fan-control IC has a programmable output and just about every feature that might be needed in fan control, including the ability to interface accurately to the temperature-sensing diode provided on chips, such as microprocessors, that account for most of the dissipation in a system. (The purpose of the diode is to provide a rapid indication of critical junction temperatures, avoiding all the thermal lags inherent in a system. It permits immediate initiation of cooling, based on a rise in chip temperature.) In order to keep the power used by the ADM1028 at a minimum, it operates on supply voltages from 3.0 V to 5.5 V, with +2.5-V full scale output.

5-V fans allow only a limited range of speed control, since their start-up voltage is close to their 5-V full speed level. But the ADM1028 can be used with 12-V fans by employing a simple step-up booster amplifier with a circuit such as that shown in Figure 4.

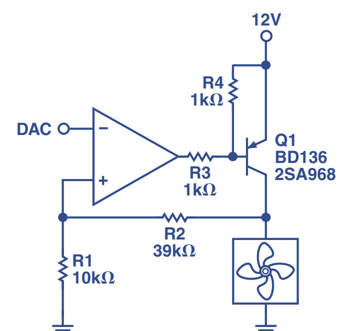


Figure 4. Boost circuit for driving a 12-V fan, using the output from the linear fan-control ADM1028's DAC.

The principal advantage of linear control is that it is quiet. However, as we have noted, the speed-control range is limited. For example, a 12-V fan with a control voltage range from 7 V to 12 V could be running at half speed at 7 V. The situation is even worse with a 5-V fan. Typically, 5-V fans will require that 3.5 V or 4 V be applied to get them started, but at that voltage they will be running at close to full speed, with a very limited range of speed control. But running at 12 V, using circuits such as that shown in Figure 4, is far from optimum from an efficiency perspective. That is because the boost transistor dissipates a relatively large amount of power (when the fan is operating at 8 V, the 4-V drop across the transistor is not very efficient). The external circuit required is also relatively expensive.

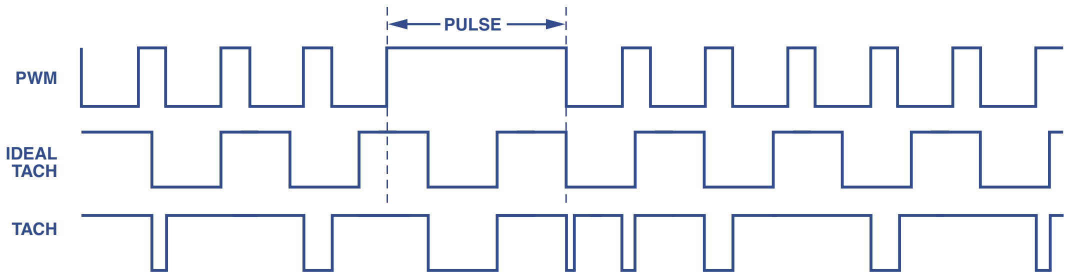


Figure 6. Pulse stretching to gather tach information.

PWM Control: The prevalent method currently used for controlling fan speed in PCs is low-frequency *PWM control.* In this approach, the voltage applied to the fan is always either zero or full-scale—avoiding the problems experienced in linear control at lower voltages. Figure 5 shows a typical drive circuit used with PWM output from the ADT7460 thermal voltage controller.

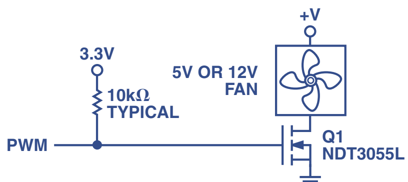


Figure 5. A low-frequency PWM fan-drive circuit.

The principal advantage of this drive method is that it is simple, inexpensive, and very efficient, since the fan is either fully *on* or fully *off.*

A disadvantage is that the tach information is chopped by the PWM drive signal, since power is not always applied to the fan. The tach information can be retrieved using a technique called *pulse stretching*—switching the fan on long enough to gather the tach information (with a possible increase of audible noise). Figure 6 shows a case of pulse stretching.

Another disadvantage of low-frequency PWM is commutation noise. With the fan coils continuously switched on and off, audible noise may be present. To deal with this noise, the newest Analog Devices fan controllers are designed to drive the fan at a frequency of 22.5 kHz, which is outside the audible range. The external control circuit is simpler with high-frequency PWM, but it can only be used with 4-wire fans. Although these fans are relatively new to the market, they are rapidly becoming more popular. Figure 7 depicts the circuit used for high-frequency PWM.

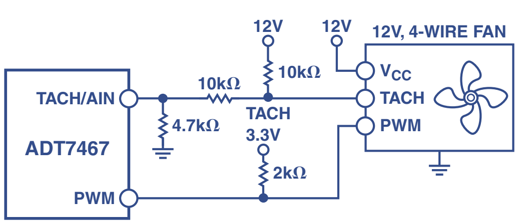


Figure 7. Circuit for driving a fan with high-frequency PWM.

The PWM signal drives the fan directly; the drive FET is integrated inside the fan. Reducing the external component count, this approach makes the external circuit much simpler. Since the PWM drive signal is applied directly to the coils of the fan, the fan's electronics are always powered on, and the tach signal is always available. This eliminates the need for pulse stretching—and the noise it can produce. The commutation noise is also eliminated, or reduced significantly, since the coils are being switched with a frequency outside the audible range.

SUMMARY

From the standpoints of acoustic noise, reliability, and power efficiency, the most preferable method of fan control is the use of high-frequency (>20 kHz) PWM drive.

Besides eliminating the need for noisy pulse stretching and the commutation noise associated with low-frequency PWM, it has a much wider control range than linear control. With high-frequency PWM, the fan can be run at speeds as low as 10% of full speed, while the same fan may only run at a minimum of 50% of full speed using linear control. It is more energy efficient, because the fan is always either fully on or fully off. (With the FET either off or in saturation, its dissipation is very low, eliminating the significant losses in the transistor in the linear case.) It is quieter than always-on or on/off control, since the fan can run at lower speeds—that can be varied gradually. Finally, running the fan slower also improves its lifetime, increasing system reliability.

Control Method	Advantages	Disadvantages
On/Off	Inexpensive	Worst acoustic performance—fan is always running.
Linear	Most quiet	Expensive circuit Inefficient—loss of power in the amplifier circuit
Low-Frequency PWM	Efficient Wide speed-control range when measuring speed	Fan commutation noise Pulse stretching required
High-Frequency PWM	Efficient Good acoustics, almost as good as linear. Inexpensive external circuit Wide speed-control range	Must use 4-wire fans

This article can be found at http://www.analog.com/library/analogDialogue/archives/38-02/fan_speed.html

Use of Video Technology To Improve Automotive Safety Becomes More Feasible with Blackfin™ Processors

By David Katz [david.katz@analog.com]
Tomasz Lukasiak [tomasz.lukasiak@analog.com]
Rick Gentile [richard.gentile@analog.com]

Dozens of processors control every performance aspect of today's automobiles, and not a single feature of the "vehicle experience" remains untouched by technology. Whether it's climate control, engine control, or entertainment, there has been constant evolution of capabilities in manufacturer offerings over the last decade. One of the forces behind this evolution, the rapidly increasing performance-to-cost ratio of signal processors, is about to have a profound impact on another critical automotive component—the safety subsystem.

While most currently available safety features utilize a wide array of sensors—principally involving microwaves, infrared light, lasers, accelerometers, or position detection—only recently have processors been introduced that can meet the real-time computation requirements that allow *video image processing* to contribute substantially to safety technology. The Analog Devices Blackfin *media-processor* family offers attractive solutions for this growing market, with its high processing speeds, versatile data-movement features, and video-specific interfaces. This article will discuss the roles that Blackfin processors can play in the emerging field of video-based automotive safety.

VIDEO IN AUTOMOTIVE SAFETY SYSTEMS

In many ways, car safety can be greatly enhanced by video-based systems that use high-performance media processors. Because short response times are critical to saving lives, however, image processing and video filtering must be done deterministically in real time. There is a natural tendency to use the highest video frame rates and resolution that a processor can handle for a given application, since this provides the best data for decision making. In addition, the processor needs to compare vehicle speeds and relative vehicle-object distances against desired conditions—again in real time. Furthermore, the processor must interact with many vehicle subsystems (such as the engine, braking, steering, and airbag controllers), process sensor information from all these systems, and provide appropriate audiovisual output to the driver. Finally, the processor should be able to interface to navigation and telecommunication systems to react to and log malfunctions, accidents, and other problems.

Figure 1 shows the basic video operational elements of an automotive safety system, indicating where image sensors might be placed throughout a vehicle, and how a lane departure system might be integrated into the chassis. There are a few things worth noting. First, multiple sensors can be shared by different automotive safety functions. For example, the rear-facing sensors can be used when the vehicle is backing up, as well as to track lanes as the vehicle moves forward. In addition, the lane-departure system might accept feeds from any of a number of camera sources, choosing the appropriate inputs for a given situation. In a basic system, a video stream feeds its data to the embedded processor. In more advanced systems, the processor receives other sensor information, such as position data from GPS receivers.

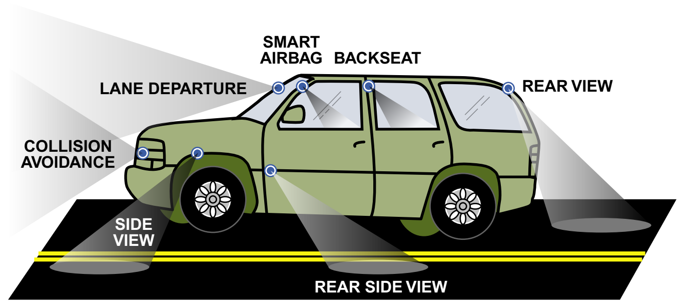


Figure 1. Basic camera-placement regions for automotive safety applications.

Smart Airbags

An emerging use of media processors in automotive safety is for "intelligent airbag systems," which base deployment decisions on who is sitting in the seat affected by the airbag. At present, weight-based systems are in widest use, but video sensing will become popular within five years. Either thermal or regular cameras may be used, at rates up to 200 frames per second, and more than one might be employed—to provide a stereo image of each occupant. The goal is to characterize the position and posture of the occupants—not just their size. In the event of a collision, the system must choose whether to restrict deployment entirely, deploy with a lower force, or deploy fully. In helping to determine body position, image-processing algorithms must be able to differentiate between a person's head and other body parts.

In this system, the media processor must acquire multiple image streams at high frame rates, process the images to profile the size and position of each occupant under all types of lighting conditions, and constantly monitor all the crash sensors, located throughout the car, in order to make the best deployment decision possible in a matter of milliseconds.

Collision Avoidance and Adaptive Cruise Control

Another high-profile safety application is *adaptive cruise control* (ACC), a subset of *collision avoidance systems*. ACC is a convenience feature that controls engine and braking systems to regulate the speed of the car and its distance from the vehicle ahead. The sensors employed involve a combination of microwave, radar, infrared, and video technology. A media processor might process between 17 and 30 frames per second in real time from a camera—focused on the roadway—mounted near the car's rear-view mirror. The image-processing algorithms may include frame-to-frame image comparisons, object recognition, and contrast equalization for varying lighting scenarios. Goals of the video sensor input are to provide information about lane boundaries and road curvature, and to categorize obstacles, including vehicles ahead of the car.

ACC systems are promoted as a convenience feature, while true collision avoidance systems actively aim to avoid accidents by coordinating the braking, steering, and engine controllers of the car. As such, they have been slower to evolve because of the complexity of the task, the critical reliability considerations, and legal and social consequences. It is estimated that deployment of these systems may be well on its way by 2010. In view of the typical 5-year automotive design cycle, such system designs are already underway.

Collision *warning* systems, like ACC, are a subset of the collision-avoidance category. These provide a warning of a possibly impending accident, but they don't actively avoid it. There are two main subcategories within this niche:

Blind spot monitors—Cameras are mounted strategically around the periphery of the vehicle to provide a visual display of the driver's blind spots—and to sound a warning if the processor senses the presence of another vehicle in a blind-spot zone. In reverse gear, these systems also serve as back-up warnings, cautioning the driver about obstructions in the rear of the car. A display could be integrated with the rear-view mirror, providing a full, unobstructed view of the car's surroundings. Moreover, the system might include a video of "blind spots" *within* the car cabin, allowing the driver to monitor a rear-facing infant, for example.

Lane-departure monitors—These systems can notify drivers if it is unsafe to change lanes or if they are straying out of a lane or off the road—thus aiding in detecting driver fatigue. Forward-facing cameras monitor the car's position relative to the roadway's centerline and side markers, up to 50 to 75 feet in front of the car. The system sounds an alarm if the car starts to leave the lane unintentionally.

LANE DEPARTURE—A SYSTEM EXAMPLE

In addition to the role that a media processor can play in video-based automotive safety applications, it is instructive to analyze typical components of just such an application. To that end, let's probe further into a lane-departure monitoring system that could employ the Blackfin media processor.

The overall system diagram of Figure 2 is fairly straightforward, considering the complexity of the signal processing functions being performed. Interestingly, in a video-based lane departure system, the bulk of the processing is image-based, and is carried out within a signal processor rather than by an analog signal chain. This represents a big savings on the system bill-of-materials. The output to the driver consists of a warning to correct the car's projected path before the vehicle leaves the lane unintentionally. It may be an audible "rumble-strip" sound, a programmed chime, or a voice message.

The video input system to the embedded processor must perform reliably in a harsh environment, including wide and drastic temperature shifts and changing road conditions. As the data stream enters the processor, it is transformed—in real time—into a form that can be processed to output a decision. At the simplest level, the lane departure system looks for the vehicle's position with respect to the lane markings in the road. To the processor, this means the incoming stream of road imagery must be transformed into a series of lines that delineate the road surface.

The processor can find lines within a field of data by looking for edges. These edges form the boundaries within which the driver should keep the vehicle while it is moving forward. The processor must track these line markers and determine whether to notify the driver of irregularities.

Keep in mind that several other automobile systems also influence the lane-departure system. For example, use of the braking system and the turn signals typically will block lane departure warnings during intentional lane changes and slow turns.

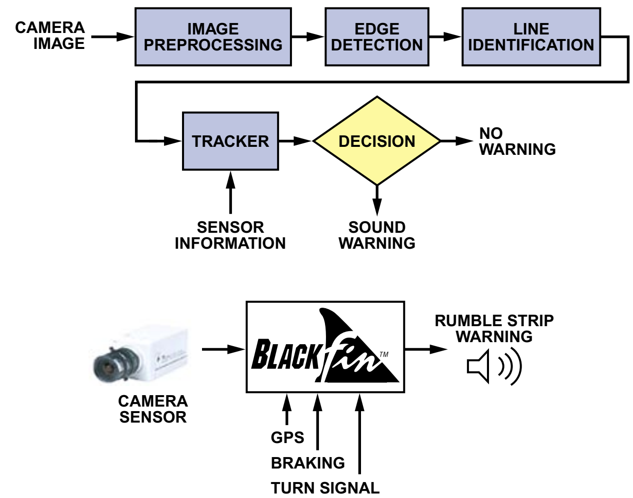


Figure 2. Basic steps in a lane-departure algorithm and how the processor might connect to the outside world.

Let's now drill deeper into the basic components of the lane-departure system example. Figure 3 follows the same basic operational flow as Figure 2 but with more insight into the algorithms being performed. The video stream coming into the system needs to be filtered and smoothed to reduce noise caused by temperature, motion, and electromagnetic interference. Without this step, it would be difficult to find clean lane markings.

The next processing step involves edge detection; if the system is set up properly, the edges found will represent the lane markings. These lines must then be matched to the direction and position of the vehicle. The *Hough transform* will be used for this step. Its

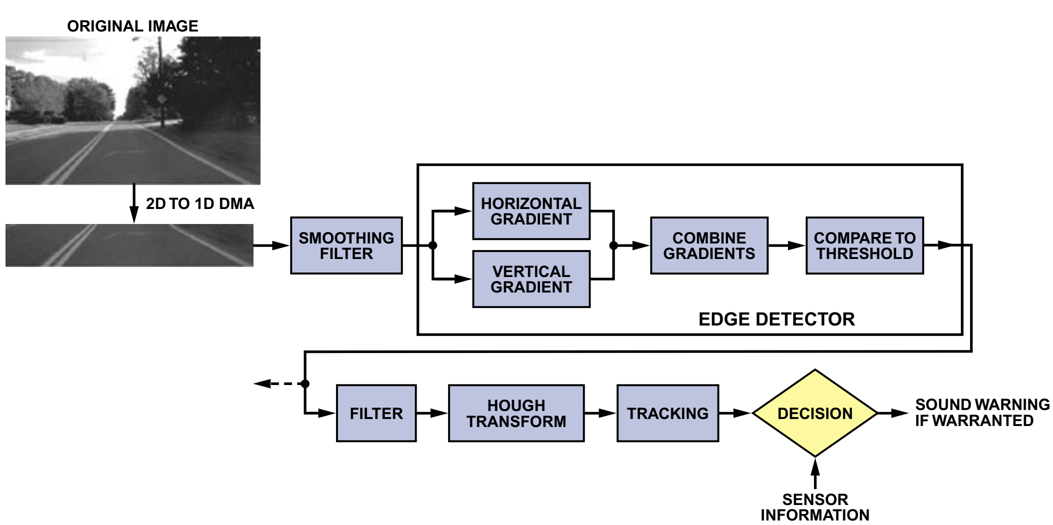


Figure 3. Algorithm flow, showing results of intermediate image-processing steps.

output will be tracked across frames of images, and a decision will be made based on all the compiled information. The final challenge is to send a warning in a timely manner without sounding false alarms.

Image Acquisition

An important feature of the Blackfin Processor is its *parallel peripheral interface* (PPI), which is designed to handle incoming and outgoing video streams. The PPI connects without external logic to a wide variety of video converters. In addition to ITU-R 656-compliant video encoders and decoders, the PPI can connect to CMOS camera chips and LCD displays, which find common use in the automotive industry. Because it can capture video in real time, the PPI is instrumental for the kinds of auto safety applications discussed in this article.

In devices supporting ITU-R 656, each boundary between blanking data and active video data is set using a 4-byte data sequence that is embedded within the data stream. The PPI automatically decodes this sequence, without processor intervention, to collect the incoming active video frames. With this embedded control scheme, the physical connection is simply eight data lines and a clock.

The PPI also connects to a wide range of image sensors and data converters that do not have an embedded control scheme. In these cases, the PPI provides up to three frame syncs to manage incoming or outgoing data. For a video stream, these frame syncs function as physical horizontal sync, vertical sync and field lines (HSYNC, VSYNC, and FIELD).

For automotive safety applications, image resolutions typically range from VGA (640 × 480 pixels/image) down to QVGA (320 × 240 pixels/image). Regardless of the actual image size, the format of the data transferred remains the same—but lower clock speeds can be used when less data is transferred. Moreover, in the most basic *lane-departure warning systems*, only gray-scale images are required. The data bandwidth is therefore halved (from 16 bits/pixel to 8 bits/pixel) because chroma information can be ignored.

Memory and Data Movement

Efficient memory usage is an important consideration for system designers because external memories are expensive, and their access times can have high latencies. While Blackfin processors have an on-chip SDRAM controller to support the cost-effective addition of larger, off-chip memories, it is still important to be judicious in transferring *only the video data needed* for the application. By intelligently decoding ITU-R 656 preamble codes, the PPI can aid this "data-filtering" operation. For example, in some applications, only the active video fields are required. In other words, horizontal and vertical blanking data can be ignored and not transferred into memory, resulting in up to a 25% reduction in the amount of data brought into the system. What's more, this lower data rate helps conserve bandwidth on the internal and external data buses.

Because video data rates are very demanding, frame buffers must be set up in external memory, as shown in Figure 4. In this scenario, while the processor operates on one buffer, a second buffer is being filled by the PPI via a DMA transfer. A simple semaphore can be set up to maintain synchronization between the frames. With Blackfin's flexible DMA controller, an interrupt can be generated at virtually any point in the memory fill process, but it is typically configured to occur at the end of each video line or frame.

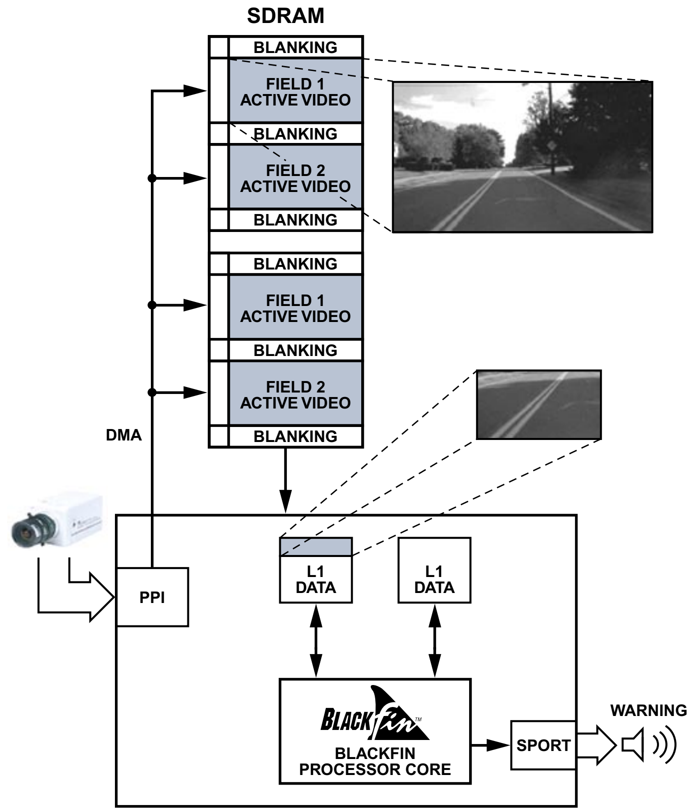


Figure 4. Use of external memory for a frame buffer.

Once a complete frame is in SDRAM, the data is normally transferred into internal L1 data memory so that the core can access it with single-cycle latency. To do this, the DMA controller can use two-dimensional transfers to bring in pixel blocks. Figure 5 shows an example of how a 16 × 16 "macroblock," a construct used in many compression algorithms, can be stored linearly in L1 memory via a 2D DMA engine.

To efficiently navigate through a source image, four parameters need to be controlled: X Count, Y Count, X Modify, and Y Modify. X and Y Counts describe the number of elements to read in/out in the "horizontal" and "vertical" directions, respectively. *Horizontal* and *vertical* are abstract terms in this application because the image data is actually stored linearly in external memory. X and Y Modify values achieve this abstraction by specifying an amount to "stride" through the data after the requisite X Count or Y Count has been transferred.

From a performance standpoint, up to four unique SDRAM internal banks can be active at any time. This means that in the video framework, no additional bank-activation latencies are observed when the 2D-to-1D DMA is pulling data from one bank while the PPI is feeding another.

Projection Correction

The camera used for the lane departure system can be located in the center-top location of the front windshield, facing forward, in the rear windshield, facing the road already traveled, or as a "bird's-eye" camera, which gives the broadest perspective of the upcoming roadway and can thus be used instead of multiple line-of-sight cameras. In this latter case, the view is warped because of the wide-angle lens, so the output image must be remapped into a linear view before parsing the picture content.

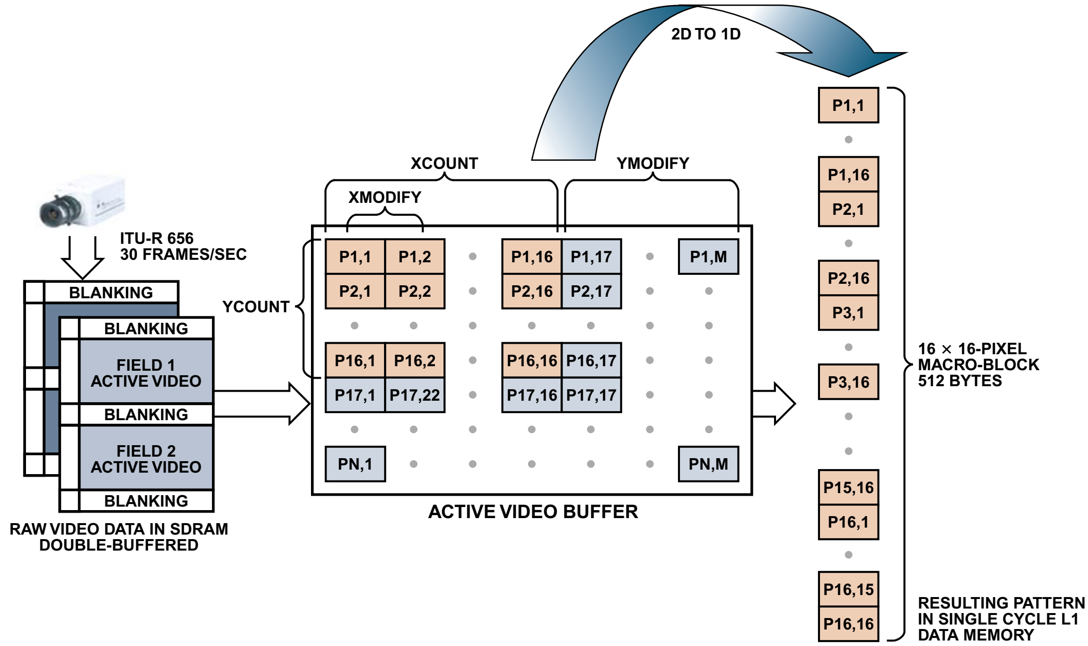


Figure 5. A 2D to 1D DMA transfer from SDRAM into L1 memory.

Image Filtering

Before doing any type of edge detection, it is important to filter the image to smooth out any noise picked up during image capture. This is essential because noise introduced into an edge detector can result in false edges output from the detector.

Obviously, an image filter needs to operate fast enough to keep up with the succession of input images. Thus, it is imperative that image filter kernels be optimized for execution in the fewest possible number of processor cycles. One effective means of filtering is accomplished with a basic two-dimensional convolution operation. Let's look at how this computation can be performed efficiently on a Blackfin Processor.

Convolution is one of the fundamental operations in image processing. In two-dimensional convolution, the calculation performed for a given pixel is a weighted sum of intensity values from pixels in the neighborhood of that pixel. Since the neighborhood of a mask is centered on a given pixel, the mask area usually has odd dimensions. The mask size is typically small relative to the image; a 3 × 3 mask is a common choice because it is computationally reasonable on a per-pixel basis but large enough to detect edges in an image.

The basic structure of the 3 × 3 kernel is shown in Figure 6. As an example, the output of the convolution process for a pixel at row 20, column 10 in an image would be:

$$Out(20,10) = A \times (19,9) + B \times (19,10) + C \times (19,11) + D \times (20,9) + E \times (20,10) + F \times (20,11) + G \times (21,9) + H \times (21,10) + I \times (21,11)$$

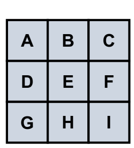


Figure 6. Basic structure of the 3 × 3 convolution kernel.

The high-level algorithm can be described with the following steps:

1. Place the center of the mask over an element of the input matrix.
2. Multiply each pixel in the mask neighborhood by the corresponding filter mask element.
3. Sum each of the multiplies into a single result.
4. Place each sum in a location corresponding to the center of the mask in the output matrix.

Figure 7 shows an input matrix, F, a 3 × 3 mask matrix, H, and an output matrix, G.

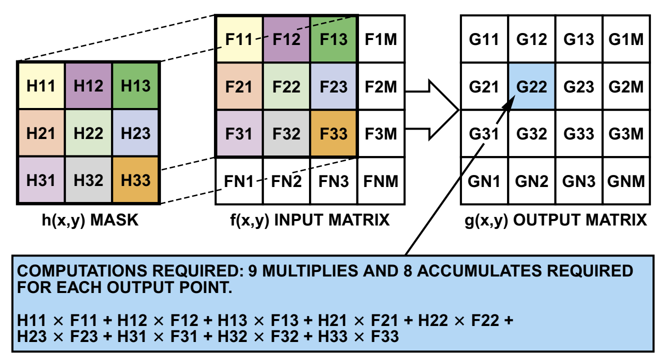


Figure 7. Input matrix, F; 3 × 3 mask matrix, H; and output matrix, G.

After each output point is computed, the mask is moved to the right. On the image edges, the algorithm wraps around to the first element in the next row. For example, when the mask is centered on element F2M, the H23 element of the mask matrix is multiplied by element F31 of the input matrix. As a result, the usable section of the output matrix is reduced by one element along each edge of the image.

By aligning the input data properly, both of Blackfin's *multiply-accumulate* (MAC) units can be used in a single processor cycle to process two output points at a time. During this same cycle, multiple data fetches occur in parallel with the MAC operation. This method allows efficient computation of 2 output points for each loop iteration, or 4.5 cycles per pixel instead of the 9 cycles per pixel of Figure 7.

Edge Detection

A wide variety of edge detection techniques are in common use. Before considering how an edge can be detected, the algorithm must first settle on a suitable definition for what an edge actually is, then find ways to enhance the edge features to improve the chances of detection. Because image sensors are non-ideal, two issues must be dealt with—*noise* and the effects of *quantization errors.*

Noise in the image will almost guarantee that pixels having equal gray scale levels in the original image will not have equal levels in the noisy image. Noise will be introduced based on many factors that can't be easily controlled, such as ambient temperature, vehicular motion, and outside weather conditions. Quantization errors in the image will result in edge boundaries extending across a number of pixels. These factors work together to complicate edge detection. Because of this, any image-processing algorithm selected must keep noise immunity as a prime goal.

One popular detection method uses a set of common derivative-based operators to help locate edges within the image. Each of the derivative operators is designed to find places where there are changes in intensity. In this scheme, the edges can be modeled by a smaller image that contains the properties of an ideal edge.

We'll discuss the Sobel Edge Detector because it is easy to understand and illustrates principles that extend into more complex schemes. The Sobel Detector uses two convolution kernels to compute gradients for both horizontal and vertical edges. The first is designed to detect changes in vertical contrast (S_x). The second detects changes in horizontal contrast (S_y).

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The output matrix holds an "edge likelihood" magnitude (based on horizontal and vertical convolutions) for each pixel in the image. This matrix is then thresholded in order to take advantage of the fact that large responses in magnitude correspond to edges within the image. Therefore, at the input of the Hough Transform stage, the image consists only of either "pure white" or "pure black" pixels, with no intermediate gradations.

If the true magnitude is not required for an application, this can save a costly square root operation. Other common techniques in building a threshold matrix include summing the gradients from each pixel or simply taking the largest of the two gradients.

Straight Line Detection—*Hough Transform*

The Hough transform is a widely used method for finding global patterns such as lines, circles, and ellipses in an image by localizing them in a parameterized space. It is especially useful in lane detection because *lines* can be easily detected as *points* in Hough transform space, based on the polar representation of Equation 1:

$$\rho = x\cos\theta + y\sin\theta \qquad (1)$$

The meaning of this equation can be visualized by extending a perpendicular from the given line to the origin, such that θ is the angle that the perpendicular makes with the abscissa and ρ is the length of the perpendicular. Thus, one pair of coordinates (ρ, θ) can fully describe the line. Lines L1 and L2 in Figure 8a demonstrate this concept. Figure 8b shows that L1 is defined by θ_1 and the length of the red perpendicular, while L2 is defined by θ_2 and the length of the blue perpendicular line.

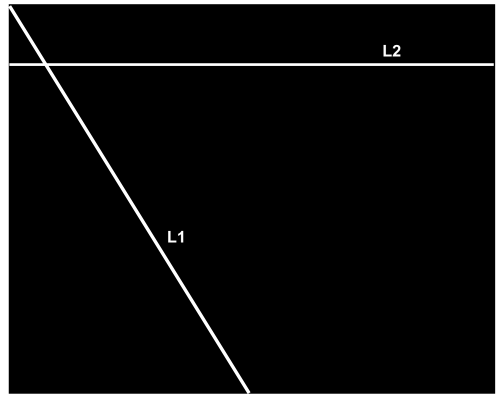


Figure 8a. The output of an edge detector is a binary image like this one, which can be visually inspected by a human observer to show lines. A Hough Transform allows localization of these two lines.

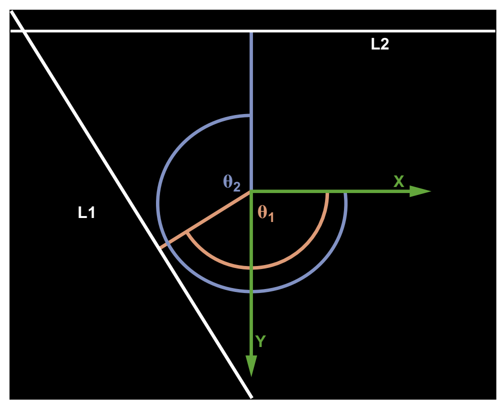


Figure 8b. The two white lines in the image above can be described by the lengths and angles of the red and blue perpendicular line segments extending from the origin.

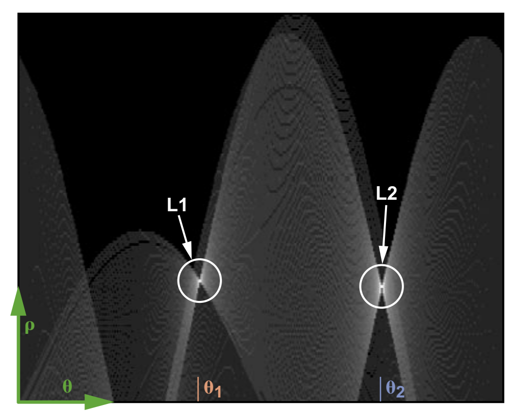


Figure 8c. The Hough transform of the image in Figure 8a. The range for θ is [0, 2π], and the range for ρ is one-half the diagonal of the input image in Figure 8a. The two bright regions correspond to local maxima, which can be used to reconstruct the two lines in Figure 8a.

Another way to look at the Hough Transform is to consider a way that the algorithm could be implemented intuitively:

1. Visit only white pixels in the binary image.
2. For each pixel and every θ value being considered, draw a line through the pixel at angle θ from the origin. Then calculate ρ, which is the length of the perpendicular between the origin and the line under consideration.
3. Record this (ρ, θ) pair in an accumulation table.
4. Repeat steps 1–3 for every white pixel in the image.
5. Search the accumulation table for the (ρ, θ) pairs encountered most often. These pairs describe the most probable "lines" in the input image, because in order to register a high accumulation value, there had to be many white pixels that existed along the line described by the (ρ, θ) pair.

The Hough transform is computationally intensive because a sinusoidal curve is calculated for each pixel in the input image. However, certain techniques can speed up the computation considerably.

First, some of the computation terms can be computed ahead of time, so that they can be referenced quickly through a lookup table. In Blackfin's fixed-point architecture it is very useful to store the lookup table only for the *cosine* function. Since the sine values are 90 degrees out of phase with the cosines, the same table can be used, with an offset. With the lookup tables in use, the computation of Equation (1) can be represented as two fixed-point multiplications and one addition.

Another factor that can improve performance is a set of assumptions about the nature and location of lane markings within the input image. By considering only those input points that could potentially be lane markings, a large number of unnecessary calculations can be avoided, since only a narrow range of θ values need be considered for each white pixel.

The output of a Hough Transform is a set of straight lines that could potentially be lane markings. Certain parameters of these lines can be calculated by simple geometric equations. Among the parameters useful for further analysis are the *offset* from the camera's center axis, the *widths* of the detected lines, and the *angles* with respect to the position of the camera. Since lane markings in many highway systems are standardized, a set of rules can eliminate some lines from the list of lane-marking candidates. The set of possible lane-marking variables can then be used to derive the position of the car.

Lane Tracking

Lane information can be determined from a variety of possible sources within an automobile. This information can be combined with measurements of vehicle-related parameters (e.g., velocity, acceleration, etc.) to assist in lane tracking. Based on the results of these measurements, the lane-departure system can make an intelligent decision as to whether an unintentional departure is in progress. In advanced systems, other factors could be modeled, such as the time of day, road conditions, and driver alertness.

The problem of estimating lane geometry is a challenge that often calls for using a *Kalman filter* to estimate the road curvature. Specifically, the Kalman filter can predict future road information—which can then be used in the next frame to reduce the computational load presented by the Hough transform.

As described earlier, the Hough transform is used to find lines in each image. But these lines also need to be tracked over a series of images. In general, a Kalman filter can be described as a *recursive filter that estimates the future state of an object*. In this case, the object is a line. The state of the line is based on its location and its motion path across several frames.

Along with the road state itself, the Kalman filter provides a variance for each state. The predicted state and the variance can be used in conjunction to narrow the search space of the Hough transform in future frames, which saves processing cycles.

Decision Making—*Current Car Position or Time to Lane-Crossing*

From our experience, we know that false positives are always undesirable. There is no quicker way to get a consumer to disable an optional safety feature than to have it indicate a problem that does not exist.

With a processing framework in place, system designers can add their own intellectual property (IP) to the decision phase of each of the processing threads. The simplest approach might be to take into account other vehicle attributes when making a decision. For example, a lane-change warning could be suppressed when a lane change is perceived to be intentional—as when a blinker is used or when the brake is applied. More complex systems may factor in GPS coordinate data, occupant driving profile, time of day, weather, and other parameters.

CONCLUSION

In the foregoing discussion, we have only described an example *framework* for how an image-based lane departure system might be structured. The point we have sought to establish is that when a flexible media processor is available for the design, there is plenty of room to consider feature additions and algorithm optimizations. ▶

This article can be found at http://www.analog.com/library/analogDialogue/archives/38-03/auto_safety.html

PRODUCT INTRODUCTIONS: VOLUME 38, NUMBER 1

Data sheets for all ADI products can be found by entering the model number in the Search Box at www.analog.com

January

ADC, Successive-Approximation, 12-bit, low power, 100-ksps, in SOT-23 package ... **AD7457**
Amplifier, Buffer, plus V_{COM}, single-supply, 5-channel for TFT LCD panels ... **ADD8706**
Amplifier, Operational, Quad, for TFT LCD panels ... **ADD8704**
Amplifier, Power, X-PA™, high efficiency for mobile handsets has integrated RF power control ... **ADL5552**
Amplifiers, Operational, JFET, precision low power single-supply ... **AD8625/26**
Amplifier, Transimpedance, 10.7-Gbps, with performance monitor ... **ADN2820**
Cable Line Driver operates on 3.3-V supply ... **AD8324**
DACs, Multiplying, 8-/10-/12-bit, wideband have parallel interface ... **AD5424/33/45**
DAC, Voltage-Output, 16-bit serial input, specifies 1-LSB max DNL and INL ... **AD5570**
Frequency Synthesizer, PLL, includes voltage-controlled oscillator ... **ADF4360-x**
Logarithmic Converter, low cost dual, provides 120-dB dynamic range ... **ADL5310**
Smart-Transducer Front-Ends, Complete, include ADC, DACs, and flash MCU ... **ADuC841/42/43**
Switches, CMOS, SPST, Quad, 0.5-ohm ... **ADG811/12/13**
Video Encoder, multiformat HDTV, has six 11-bit DACs ... **ADV7312**
Voltage References, precision, shunt-mode ... **ADR520/25/30/40/50**

February

ADCs, Successive-Approximation, 16-bit, 500-/1000-ksps include on-chip voltage reference ... **AD7666/67**
ADC, Time-Interleaved, 12-bit, 400-Msps, provides enhanced dynamic range ... **AD12400**
Amplifier, Variable-Gain, has ultralow noise preamp and programmable input impedance ... **AD8331**
Correlated Double Sampler ... **AD9823**
Delay-Locked Loop, triple, 6-channel, for LCD timing ... **AD8389**
Direct-Digital Synthesizers, CMOS, 400-Msps, 14-bit operate on 1.8-V supply ... **AD9951/52/53**
Laser-Diode Driver, 3-channel, includes on-chip oscillator .. **AD9662**
Op Amp, high speed, features ultralow noise and distortion ... **AD8099**
Op Amps, precision, combine low noise, low bias current, and low power ... **AD8672/74**
Voltage References, XFET®, ultralow noise, can source and sink current ... **ADR430/31/33/35/39**

March

ADC, Pipelined, 10-bit, 170-Msps ... **AD9411**
ADC, Successive-Approximation, 16-bit, 100-ksps, in 6-lead SOT-23 package ... **AD7680**
Amplifier, Variable-Gain, digitally controlled, operates to 750 MHz ... **AD8370**
Audio Codec, SoundMAX®, AC '97 ... **AD1981BL**
BTSC Encoder, dual digital, includes stereo audio DAC ... **AD71028**
Clock- and Data Recovery IC, 155-/622-Mbps, includes limiting amplifier ... **ADN2807**
Current Source, 12-bit, provides 300-mA full-scale output ... **ADN8810**
DAC, Multiplying, 8-bit, wideband, has serial interface ... **AD5425**
DACs, Multiplying, 8-/10-/12-bit, wideband, have serial interface ... **AD5426/32/43**
DACs, Multiplying, 16-/14-bit, include on-chip resistors for 4-quadrant applications ... **AD5546/56**
Decimating LCD DecDriver®, 10-bit, 6-channel ... **AD8383**
Decimating LCD Driver, 10-bit, 6-channel, includes level shifters ... **AD8384**
Digital Potentiometer, 64-position, OTP ... **AD5171**
Energy Metering IC has on-chip fault detection ... **ADE7760**
Energy Metering IC includes fault detection and missing-neutral detection ... **ADE7761**
Energy Metering IC, polyphase, provides per-phase information ... **ADE7758**
Synthesizer, Direct Digital (DDS), 10-bit, 400-Msps, operates on 1.8-V supply ... **AD9859**
Synthesizer, Frequency, Dual, fractional-N/integer-N ... **ADF4251**

AUTHORS

Mary Burke (page 6) is an applications engineer at Analog Devices in Limerick, Ireland, working in the Temperature and Systems Monitoring Group. She currently has responsibility for PC temperature sensors and fan controllers. Mary joined Analog Devices in 1999 after completing a BEng degree at the National University of Ireland, Cork. In her spare time, Mary enjoys soccer and badminton.

Rick Gentile (page 9), who joined ADI in 2000 as a senior DSP applications engineer, currently leads the Blackfin® DSP Applications Group. Before joining ADI, Rick was a member of the technical staff at MIT Lincoln Laboratory, where he designed several signal processors used in a wide range of radar sensors. He received a BS from the University of Massachusetts at Amherst and Master of Science degrees from Northeastern University—in both Electrical and Computer Engineering. He enjoys sailing, windsurfing, and spending time with his daughters.

David Katz (page 9) is a senior DSP applications engineer involved in specifying and supporting Blackfin media processors. He has published dozens of articles on embedded processors—in both domestic and international publications. Previously, he worked at Motorola, Inc., as a senior design engineer in cable-modem and automation groups. David holds BS and MEng degrees in Electrical Engineering from Cornell University.

Tom Lukasiak (page 9) has been a DSP applications engineer at Analog Devices since 2000. He is currently focusing on Blackfin processor products. Tom earned his ScB (2000) and ScM (2002) degrees in Electrical Engineering from Brown University.

Eamon Neary (page 3) is an applications engineer at ADI's facility in Limerick, Ireland. He graduated with a degree in Electronic Engineering from University College Dublin (UCD) in 2000. He is currently working in the Applications Group with the Limerick MicroConverter Development Team. In this role, Eamon provides technical support for MicroConverter products and applications and is working in development of the single-cycle 8052-based MicroConverter family. His interests include foreign travel, soccer, and swimming.

Worldwide Headquarters
One Technology Way
P.O. Box 9106
Norwood, MA
02062-9106 U.S.A.
Tel: 781.329.4700
(1.800.262.5643,
U.S.A. only)
Fax: 781.326.8703

Analog Devices, Inc. Europe
c/o Analog Devices SA
17–19, rue Georges Besse
Parc de Haute
Technologie d'Antony
F-92182
Antony Cedex, France
Tel: 33.1.46.74.45.00
Fax: 33.1.46.74.45.01

Analog Devices, Inc. Japan Headquarters
New Pier Takeshiba
South Tower Building
1-16-1 Kaigan,
Minato-ku, Tokyo
105-6891, Japan
Tel: 813.5402.8210
Fax: 813.5402.1063

Analog Devices, Inc. Southeast Asia Headquarters
RBS Tower, Rm 4501-3
Times Square
1 Matheson Street
Causeway Bay
Hong Kong, PRC
Tel: 852.2.506.9336
Fax: 852.2.506.4755

Printed in the U.S.A. M020003801-77-6/04

ANALOG DEVICES

www.analog.com/analogdialogue