As Moore’s Law plunges us into the realm of multigigahertz processors and PCs with gigabytes of RAM, engineers face the task of removing the heat that these state-of-the-art components produce. Cooling such systems poses a dilemma. If you optimize the fan size and speed for nominal operating conditions, the system is susceptible to failure when conditions deteriorate. If, on the other hand, you select the fan to maintain acceptable operating temperatures under worst-case conditions, the fan may produce an annoying level of sound. Controlling fan speed is the obvious solution. If the system includes a system-management bus, you can add one of the many available sophisticated ICs for controlling fan speed. But if such a bus is unavailable, you need a stand-alone fan-speed controller (Figure 1).

Power comes from the 12V supply, and a dc/dc converter, IC1, steps down the input voltage to an intermediate voltage for powering the fan. The transfer function of this voltage is a function of resistors R1 and R2 and thermistor RT1. The thermistor is an NTC (negative-temperature-coefficient) type, so the output voltage increases with increasing temperature. The output voltage is approximately 5.5V at room temperature and increases to 12V at approximately 47°C (Figure 2). You can easily select the ratio of resistors R1, R2, and RT1 by using a spreadsheet. Note that thermistor manufacturers’ tables of resistance ratio versus temperature are easier to use than are the cumbersome equations for thermistor resistance.

Because the circuit in Figure 1 does not monitor fan speed or current, it includes R3, C1, and D1 to ensure that the fan starts turning during start-up. The time con-
stant of \( R_3 \) and \( C_1 \) serves that purpose by causing IC\(_3\)’s output to overshoot during the first few seconds of operation. After the fan starts, it easily sustains rotation at the lower operating voltages. An important criterion in selecting a dc/dc converter is the ability to operate at 100% duty cycle. IC\(_1\) satisfies that requirement and offers the convenience of an internal power MOSFET. IC\(_1\) supplies as much as 1A output current, which is enough to drive one to four standard fans. As an added benefit, its high efficiency helps to minimize the heat that the circuit removes.

Simple circuit forms multichannel temperature monitor

Susan Pratt, Analog Devices, Limerick, Ireland

You can use an ADT7461 single-channel temperature monitor, an ADG708 low-voltage, low-leakage CMOS 8-to-1 multiplexer, and three standard 2N3906 pnp transistors to measure the temperatures of three separate remote thermal zones (Figure 1). Multiplexers have an inherent-resistance on-resistance; the channel matching and flatness of this resistance normally results in a varying temperature offset. The ADT7461 temperature monitor in this system can automatically cancel resistances in series with the external temperature sensors. The resulting system is a multichannel temperature monitor. The resistance is automatically cancelled, so on-resistance flatness and channel-to-channel variations have no effect. Resistance associated with the pcb board tracks and connectors is also cancelled, thus allowing you to place the remote temperature sensors some distance from the ADT7461. The system requires no user calibration; therefore, you can connect the ADT7461 directly to the multiplexer.

The ADT7461 digital temperature monitor can measure the temperature of an external sensor with \( \pm 1 \)°C accuracy. The remote sensor can be a substrate-based or discrete transistor and normally connects to the D+ and D− pins on the ADT7461. In addition to the remote-sensor-measurement channel, the ADT7561 has an on-chip sensor. The diode-connected transistors with their emitters connected together connect to the D+ input of the ADT7461, and each of the base-collector junctions connects to a separate multiplexer input (S1 to S3). You effect the connection of the selected remote transistor to the D− input by addressing the multiplexer, which is digitally controlled by address bits, A\(_2\), A\(_1\), and A\(_0\). The ADT7461 then measures the temperature of whichever transistor connects through the multiplexer.

The ADT7461 measures the temperature of the selected sensor without interference from the other transistors. Figure 2 shows the results of measuring the temperature of three remote temperature sensors. The sensor at address 000 is at room temperature, the sensor at address 001 is at a low temperature, and the sensor at address 010 is at a high temperature. When you select no external sensor, the “open-circuit” flag in the ADT7461 register activates, and the Alert interrupt output asserts. You can expand the system to include as many external temperature sensors as you require. The limiting factor on the number of external sensors is the time available to measure all temperature sensors. If you require two-wire serial control of the multiplexer, you can use an ADG728 in place of the ADG708.
PWM controller drives LEDs from high-voltage lines
Christophe Basso, On Semiconductor, Toulouse, France

Powering LEDs from a wide dc range—say, 30 to 380V—without wasting a lot of power in the regulating block, is a difficult task when the LED current needs to be constant. Dedicated LED drivers are available, but they usually implement boost structures and are thus inadequate for high-voltage inputs. The NCP1200A, a high-voltage controller from On Semiconductor (www.onsemi.com), can serve as a constant-current generator if you add a simple coil in series with a power MOSFET. If you insert diodes between the coil and the MOSFET, the circuit becomes an economical light generator. Furthermore, there is no need for a transformer or any kind of generator if you add a simple coil to the minimum on-time (400 nsec) in high-line conditions. Because of the poor TRR (reverse-recovery time) of the power MOSFET, you can troubleshoot the common signals using the equation: \[ t_{ON} = \frac{L_i (\Delta I/V_{TOTAL})}{f_S} \] where \( f_S \) is the switching frequency. Extracting \( L_i \) yields \( L_i = \frac{1}{(1/f_S)(V_{TOTAL} + V_{IN})/(V_{TOTAL} + V_{IN})} \). If you select a ripple current of 20 mA peak-to-peak at 380V dc, then \( L_i = 16.66 \times 11.6 \times 50 = 9.6 \text{ mH} \). From this value, you can check the minimum on-time using the equation: \[ t_{ON} = \frac{9.6 \text{ mH} \times 0.02}{380} = 0.508 \text{ msec}, \] above the minimum limit. Figure 2 portrays typical signals captured on the prototype supplied with low line voltage.

Circuit forms satellite-dish command decoder
Mark Giebler, Oakdale, MN

By decoding the commands sent by a direct-broadcast satellite receiver that uses the DISEQC (digital-satellite-equipment-control) protocol, you can troubleshoot the commands or simply listen in. Eutelsat Corp (www.eutelsat.com) offers the DISEQC protocol. The technique uses only the coaxial cable between the receiver and the dish to send commands for actions such as changing the low-noise-block frequency range or switching between dishes for multisatellite reception. The DISEQC protocol specifies a bit time of 1.5 msec and bit values as shown in Figure 1, the timing diagram of bit modulation on the coaxial cable. The signal’s ac portion is a 22-Hz burst whose amplitude ranges from 300 to 600 mV. A voltage-doubler circuit detects the 22-Hz portion, producing a pulse stream in
which constant-voltage pulses having amplitudes of 0.6 to 1.2 V replace the 22-Hz bursts.

Decoding this bit stream into ASCII hex values is an ideal job for a low-cost 8-bit microcontroller. Using a microcontroller with onboard flash memory, such as the NEC Electronics μPD78F9418A (www.necelam.com), eliminates the need for external memory. The only external components are a few discrete devices for the signal detector and the coaxial-cable loop-through (Figure 2). You can add an RS-232 driver if you want to display the ASCII codes on a laptop computer via HyperTerminal. You can also use the μPD78F9418A’s onboard LCD controller to display the codes on a dedicated display.

One of the μPD78F9418A microcontroller’s 10-bit A/D converters performs pulse detection and acts as a simple timing device. Using a reference voltage of 5 V, the converter provides approximately 4.88 mV per step. An A/D-converter conversion value greater than 120 counts (585 mV) represents a valid pulse. Set the A/D converter’s conversion time to 28.8 μsec and wait to detect a pulse edge by reading the A/D converter until its value exceeds 120 and then perform a loop while doing analog-to-digital conversions. If the loop count reaches 24 with ADC values greater than 120, the bit is a zero. If the pulse has gone away, the bit is a one. Any extra delay from executing instructions in the loop has little effect, because the bit windows leave plenty of margin.

Use a microcontroller to design a boost converter

Ross Fosler, Microchip Technology, Chandler, AZ

Boost converters, like other switchers, have traditionally received their control signals from a dedicated circuit. However, a recent trend is to integrate simple switching-power-supply building blocks into generic devices, such as microcontrollers. An excellent example of this concept is a microcontroller that combines digital and analog circuitry and makes it easier to build simple power supplies. The programming capability of a microcontroller is an added benefit in power-supply designs, especially when you want to experiment with the supplies. Figure 1 illustrates a simple boost-converter design using a microcontroller; the basic boost topology in Figure 1 is a type of flyback circuit. The basic concept is easy to understand. When the MOSFET, Q, turns on, the current flowing through the inductor, L, begins to ramp up linearly (Figure 2), resulting in energy storage in the inductor. The MOSFET turns off before the inductor saturates. At this time, the inductor releases its energy to the storage capacitor, C, and the load.

You can design a simple boost converter with the following conditions: \( V_{\text{IN}} = 9 \text{V}, V_{\text{OUT}} = 18 \text{V}, R_{\text{LOAD}} = 72 \Omega, F = 1/ T = 62.5 \text{ kHz}, \eta = 70\%, \) and \( \Delta V_{\text{DROP}} = 50 \text{ mV}, \) where F is the switching frequency, \( \eta \) is the efficiency, and \( \Delta V_{\text{DROP}} \) is the output ripple voltage. You can calculate the on-time, current, ramp-down time, and the total period in terms of inductance:

\[
2L \frac{V_{\text{OUT}}^2}{R_{\text{L}}^2 V_{\text{IN}}^2} = t_{\text{ON}} = 0.1587L.
\]

\[
\frac{V_{\text{IN}}}{V_{\text{OUT}} - V_{\text{IN}}} t_{\text{ON}} = t_{\text{R}} = 0.1587L.
\]

\[
t_{\text{ON}} + t_{\text{R}} \geq T = 0.453L.
\]

Then, you calculate the peak current through the inductor and the inductance value:
Finally, you calculate the capacitance based on the ripple voltage:

\[ \frac{V_{IN}}{L} \cdot t_{ON} = I_{PEAK} = 1.428 \text{ A.} \]

\[ L = 35.28 \mu\text{H} = 33 \mu\text{H}. \]

Note that the design is slightly altered to use readily available components, by using a 33-\(\mu\text{H}\) inductor and a 220-\(\mu\text{F}\) capacitor. The difference in the inductor value is absorbed in the dead time, as is the power loss.

The control circuit can take many forms, especially if you choose a device such as the PIC16C782 microcontroller. This device integrates a built-in analog peripheral set, diverse analog visibility, and a mixed-signal PWM block. The control circuit in Figure 3 demonstrates how the analog and pulse-width modulation is contained within the PIC16C782, with the exception of the FET driver. This control circuit combines analog current control and firmware voltage control. The interesting part is the firmware, which is directly in the voltage-feedback path of the control loop. Through firmware, you can alter the dynamics of the control loop by changing the functions within the program. You may be able to design an adaptive power-control system by adjusting the phase and gain to meet the desired needs of a system.

Firmware placement within the control loop is not the only possibility; you could use a combination of firmware and hardware to monitor the system. Because the analog information is visible and the analog functions are controllable within the PIC16C782 device, you can monitor an active system for performance and function. In essence, the system can have self-diagnostic capabilities to check stability, load, input and output conditions, or anything else a system may require. You can also obtain Information about the system, through a serial port or some other means, by routing the data to a terminal or computer display. Even better, the firmware allows the design to change the functions without changing hardware. This approach eases experimentation; you simply changing firmware rather than spending hours in the lab adding or changing parts.

Figures 4 and 5 are oscilloscope photos from a working example of the boost converter implementing the basic topology in Figure 1 and the control block in Figure 3. The peak current in the inductor is 0.3 mV\(\times 0.2\Omega = 1.5\text{A} \) (Figure 4). The on-time is approximately 5.9 \(\mu\text{sec}\). The output voltage is 18V into a 72\(\Omega\) load (Figure 5). The efficiency is approximately 90%. These boost-converter design and control ideas are just a few of the many possible ones using a PIC16C782 device.

---

**Figure 2**

These curves show the switch and diode currents in the circuit of Figure 1.

**Figure 3**

A working example of the boost converter implementing the basic topology of Figure 1 shows the duty cycle (top) and the current-ramp-down waveform (bottom) for the circuit in Figure 1.

**Figure 4**

A microcontroller contains all the elements necessary for boost-converter control.

**Figure 5**

The example shows the duty cycle (top) and the output voltage (bottom) of the circuit in Figure 1.