

How to Simplify the Interface between Microcontroller and Temperature Sensor

Temperature is an analog quantity, but digital systems often use temperature to implement measurement, control, and protection functions. If you apply the right techniques and components, the necessary conversion of analog temperature to digital information won't be difficult. This application note discusses thermal comparators, PWM-output temperature sensors, and remote diode (or thermal diode) temperature sensors.

Temperature is an analog quantity, but digital systems often use temperature to implement measurement, control, and protection functions. If you apply the right techniques and components, the necessary conversion of analog temperature to digital information won't be difficult.

Reading temperature with a microcontroller (μC) is simple in concept. The μC reads the output code of an analog-to-digital converter (ADC) driven by a thermistor-resistor voltage divider, analog-output temperature sensor, or other analog temperature sensor (**Figure 1**). The ADC built into some controllers can simplify this design. ADCs require a reference voltage, which can be generated by an external device. For example, the reference voltage for a thermistor sensor is usually the same as that applied to the top of the resistor-thermistor voltage divider. However, the following complications can arise in these systems:

- **The sensor's output-voltage range is significantly smaller than the ADC's input-voltage range.** A typical ADC for this purpose might have 8-bit resolution and a 2.5V reference voltage, which is normally equivalent to the input-voltage range. If the sensor's maximum output for the temperature range of interest is only 1.25V, the effective resolution drops to 7 bits. To achieve 8-bit resolution, either add gain via an external op amp or lower the ADC's reference voltage (which may reduce the accuracy of some ADCs).
- **The error budget is tight.** Combining the error from the thermistor-resistor combination or analog-sensor device with those contributed by the ADC, the amplifier offset voltage, the tolerance of gain-setting resistors, and the voltage reference error may be more error than your system can tolerate.
- **You want a linear temperature-to-code transfer function and you're using a thermistor.** The transfer function for thermistors is very nonlinear, but it may be sufficiently linear over the narrow temperature range required in many applications. You can compensate for the nonlinearity with a look-up table, but this approach requires resources that may not be available.
- **ADC inputs are limited.** If the number of temperatures you want to measure exceeds the number of ADC inputs available, you may need to add a multiplexer, which will increase the cost and development time.
- **The number of μC I/O pins is limited.** This won't be an issue for an internal ADC, but an external serial ADC will require two to four I/O pins as an interface to the μC .

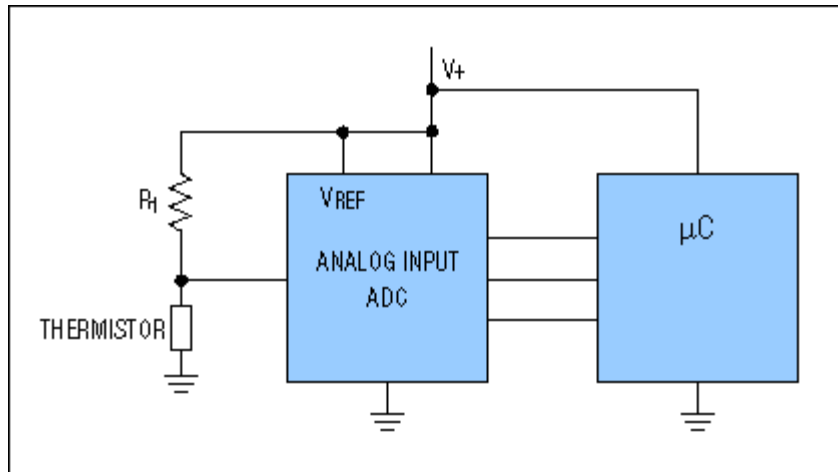


Figure 1. In this simple interface, the ADC's reference voltage is derived from the power-supply voltage. An analog temperature sensor can replace the thermistor-resistor voltage divider. In that case, the ADC (which can be internal to the μC) requires a reasonably accurate voltage reference.

The design problems are simplified if you use a temperature sensor with a digital interface. Similarly, temperature sensors with time- or frequency-based outputs can alleviate the measurement problem when ADC inputs and μC I/O pins are in short supply (**Figure 2**). The MAX6576 temperature sensor, for example, produces an output square wave whose period is proportional to absolute temperature. It comes in a 6-pin SOT23 package that requires very little board space. A single I/O pin interfaces this device to a μC ; after its internal counter measures the period, the μC calculates the temperature.

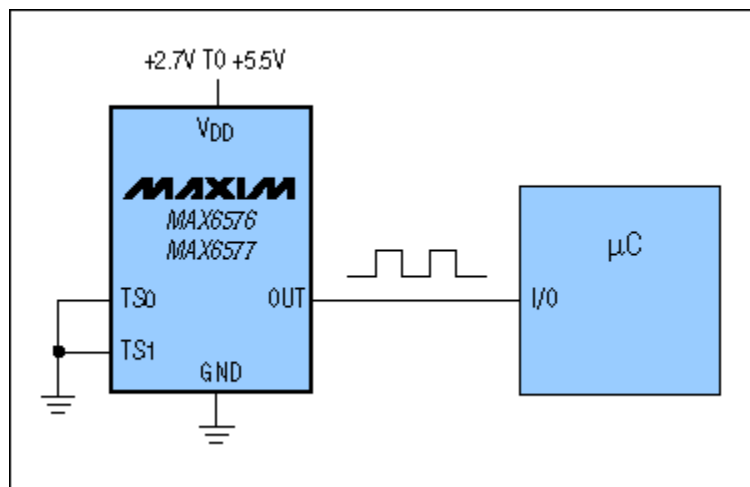


Figure 2. The MAX6576 produces a square wave with period proportional to absolute temperature; the MAX6577 produces an output frequency proportional to temperature. The resulting proportionality constant is set to one of four values by the TS0 and TS1 pins. No external components are necessary.

Applying either ground or the positive supply voltage to each of two logic inputs selects one of four period/temperature proportionality constants between $10\mu\text{s}/^\circ\text{K}$ and $640\mu\text{s}/^\circ\text{K}$.

A related temperature sensor (MAX6577) generates an output square wave whose frequency/temperature factor is programmable between $0.0675\text{Hz}/^\circ\text{K}$ and $4\text{Hz}/^\circ\text{K}$. Both devices simplify temperature acquisition by reducing the required PC board real estate, component count, and analog/digital I/O resources. They transmit temperature data to the μC through a single digital I/O pin, and the addition of a single optical isolator makes them ideal for applications that require electrical isolation between the sensor and the CPU. For measuring multiple temperatures at various locations, the choices become more complicated.

Thermistors or conventional analog sensors can be placed in appropriate locations and connected to the ADC inputs, provided the ADC has sufficient inputs available. As an alternative, the MAX6575 transmits temperature data directly to the μC ; as many as eight MAX6575s can be connected to a single μC I/O input. A single I/O trace connects the μC to these eight MAX6575s (**Figure 3**). To measure temperature, the μC briefly pulls the I/O line low, and after a short delay the first MAX6575 also pulls the I/O line low. This time delay is proportional to absolute temperature, with a proportionality constant programmed using two pins on the MAX6575.

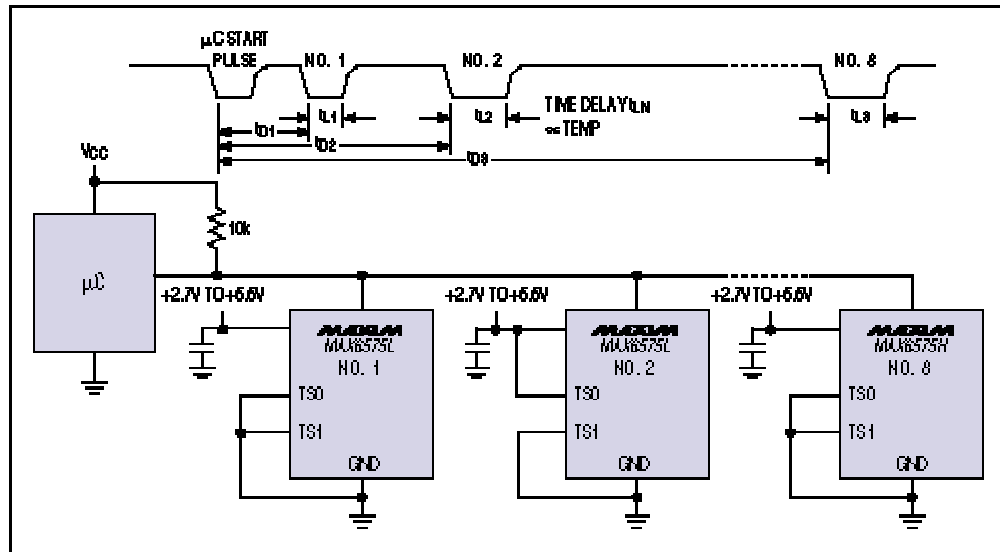


Figure 3. Using a delay scheme to encode temperature information, multiple MAX6575s transmit up to eight temperatures to the μC through a single digital I/O pin.

The first sensor holds the line low for a period proportional to temperature ($5\mu\text{s}/^\circ\text{K}$) and then releases it. After a second time delay, selected by setting the programming pins for a larger proportionality constant, the second MAX6575 pulls the I/O low and holds it for an interval defined by $5\mu\text{s}/^\circ\text{K}$. Four MAX6575s can be connected to the I/O line this way. Four more MAX6575s of the other, longer-delay version can be added to the same I/O line. The MAX6575L has delay multipliers ranging from $5\mu\text{s}/^\circ\text{K}$ to $80\mu\text{s}/^\circ\text{K}$, and the MAX6575H delay multipliers range from $160\mu\text{s}/^\circ\text{K}$ to $640\mu\text{s}/^\circ\text{K}$. Thus, as many as eight MAX6575s can be located in different places around the system, connected to the μC by a single I/O line.

For some systems, the information needed is not the exact temperature, but whether the temperature is above or below a specific value. This information can trigger a cooling fan, air conditioner, heater, or other environmental-control element. In system-protection applications, an "overtemperature bit" can trigger an orderly system shutdown to avoid losing data when the system power is cut off. This single bit of information can be obtained by measuring temperature as in the examples above, but that approach requires more software and hardware than the function demands.

Replacing the ADC in Figure 1 with a voltage comparator produces a simple 1-bit output that can drive a single I/O pin on the μC (**Figure 4**). Again, the thermistor shown can be replaced by an analog voltage-output temperature sensor. Most such devices have a relationship between temperature and output voltage that is unaffected by supply voltage. To preserve immunity from supply-voltage variations, connect the top of the comparator's resistor-divider to a voltage reference instead of the supply voltage.

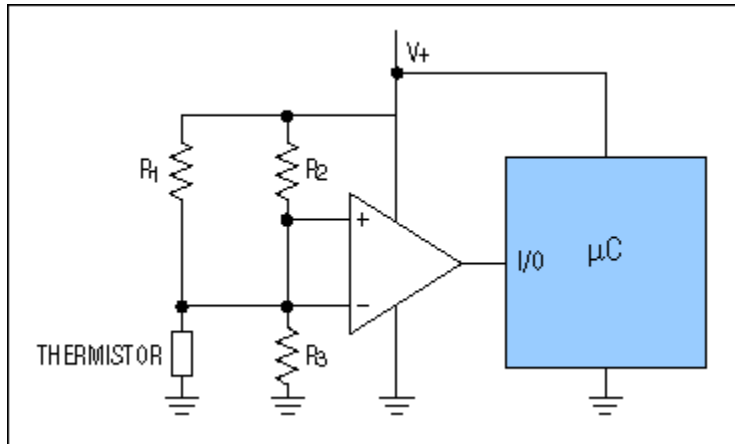


Figure 4. Combining a sensor with a comparator yields a 1-bit digital output that can warn the μC of temperature excursions beyond a predetermined threshold or trip point.

The system can be simplified by replacing the sensor-comparator combination with a thermal switch like the MAX6501. This monolithic device combines the functions of a sensor, comparator, voltage reference, and external resistors. When temperature exceeds the preset trip level, the open-drain output goes low. Some devices in this family have open-drain outputs that go low when temperature falls below the trip point (MAX6503), and others have push/pull outputs that go high when temperature goes either above or below the trip point (MAX6502, **Figure 5**, or MAX6504). In addition, the hysteresis can be set to 2°C or 10°C by connecting a package pin to $V+$ or ground. The available trip temperatures range from -45°C to $+115^{\circ}\text{C}$ in 10°C increments.

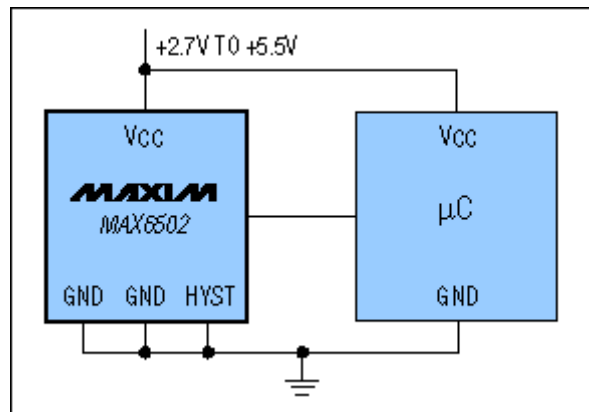


Figure 5. The MAX6502 produces a logic-high output when its temperature exceeds the preset threshold value.

As with the MAX6575, connecting several MAX6501s or MAX6503s to a single I/O trace enables the μC to be notified when temperature crosses the threshold at one or more locations. If the system must know which location has crossed the threshold, each switch output must be connected to a separate I/O pin. These sensors measure their own die temperatures, and because die temperature closely tracks lead temperature, each sensor should be placed so its leads assume the temperature of the component being monitored. In some cases, however, you must measure a temperature not tightly coupled to the sensor—such as that of a power ASIC, whose die can be much hotter than the surrounding board. An internal temperature sensor may enable the ASIC to shut itself down in response to a temperature fault, but that capability alone lacks accuracy, and it seldom warns the system of an impending thermal overload.

By adding an externally accessible p-n junction to the ASIC die, you can measure die temperature directly by forcing two or more different forward currents through the sensing junction and measuring the resulting voltages. The difference between the two voltages is proportional to the absolute die temperature:

$$V_2 - V_1 = \frac{kT}{q} \ln \frac{I_2}{I_1}$$

where I_1 and I_2 are the two current levels forced through the p-n junction, V_1 and V_2 are the resulting forward voltages across the junction, k is Boltzmann's constant, T is the absolute temperature of the junction in degrees Kelvin, and q is the electron charge.

This measurement, of course, requires precision circuitry for generating the accurate current ratios and measuring very small voltage differences while rejecting the noise produced by large transients on the power ASIC die. Fortunately, Maxim's remote-junction temperature sensors integrate these precision analog functions with a simple and versatile digital interface.

The MAX1618, for example, measures remote-junction temperatures with 8-bit (1°C) resolution and communicates the result to a μC over the SMBus (**Figure 6**). Originally designed for monitoring the CPU temperature in PCs, this device has other features that remove some of the controller's overhead. For example, the MAX1618 monitors a remote-junction temperature with a window comparator and interrupts the μC when temperature goes above or below the limit thresholds previously downloaded to its registers by the μC . Rather than poll the MAX1618 continually, the μC can set the temperature thresholds on start-up and then ignore the MAX1618 until a thermal problem requires its attention.

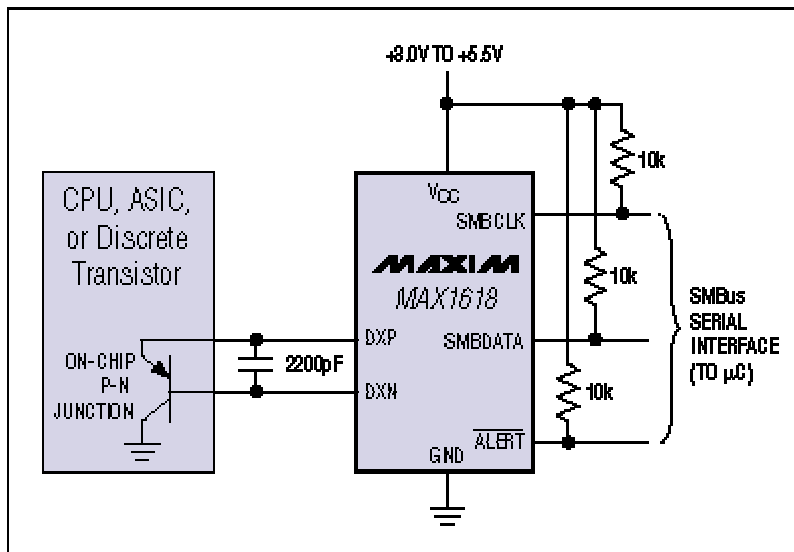


Figure 6. The MAX1618 measures the temperature of an external P-N junction (part of a discrete transistor, ASIC, or CPU) by forcing currents through the junction and measuring the resulting forward voltages.

Available in a 10-pin μMAX package, the MAX1618 can be placed close to the junction being measured. In turn, the resulting short trace lengths between the sense junction and MAX1618 help to avoid noise pickup.

MORE INFORMATION

MAX1618:	QuickView	-- Full (PDF) Data Sheet (248k)	-- Free Sample
MAX6502:	QuickView	-- Full (PDF) Data Sheet (216k)	-- Free Sample
MAX6576:	QuickView	-- Full (PDF) Data Sheet (152k)	-- Free Sample
MAX6577:	QuickView	-- Full (PDF) Data Sheet (152k)	-- Free Sample