DALLAS SEMICONDUCTOR MAXIM

Engineering Journal

Volume One

IN-DEPTH ARTICLES

DESIGN SHOWCASE

Microcontrollers improve power efficiency of 8051-based designs

Portable products continue to advance in features and capabilities. Customers are demanding more performance out of their products, which requires greater computing power. At the same time, they want products with less power consumption. At the heart of these competing demands is the microcontroller, which is typically one of the largest power consumers in portable instruments.

While many low-power processors exist, they are often limited in performance. The high-speed microcontroller family from Dallas Semiconductor is a good compromise of power and performance. It is based on the 8051 architecture—one of the most popular microcontrollers in the world. Designers prize its ease of use, rich I/O structure, and wide acceptance. Its prevalence has carried over into the portable arena, where it has found a home in many applications.

This article addresses approaches to minimizing power consumption using 8051 controllers, with emphasis on new architectural improvements that can extend the battery life of high-performance 8051-based designs. Integrating peripherals on-chip and selecting the proper clock source are discussed as ways to reduce power consumption. Software techniques for power conservation are presented, as well as a method of reducing power consumption in systems that use Stop mode.

Clock speed

The most important factor in determining power consumption in any microcontroller design is the system clock speed. The power consumption of complementary metal oxide semiconductor (CMOS) devices is directly proportional to clock speed. It follows, then, that it is beneficial from a power standpoint to run a processor at the slowest speed possible.

Figure 1 shows a typical power curve for a generic 8051 microcontroller, a relationship known to all portable system designers. In general, the current vs. frequency characteristic is linear, with a DC offset. This quiescent current is caused by static circuitry on-chip, such as comparators, operational amplifiers, etc. While this number is typically small (<1mA), it is a constant drain that needs to be considered.

Any power-conscious design will attempt to run as slowly as possible. Determination of the minimum system frequency, and hence minimum power consumption, is dependent on a number of factors, including desired performance and interrupt latency. Whatever criteria are used, however, the end goal is the same: match the operating frequency of the device as closely as possible to the requirements of the application.

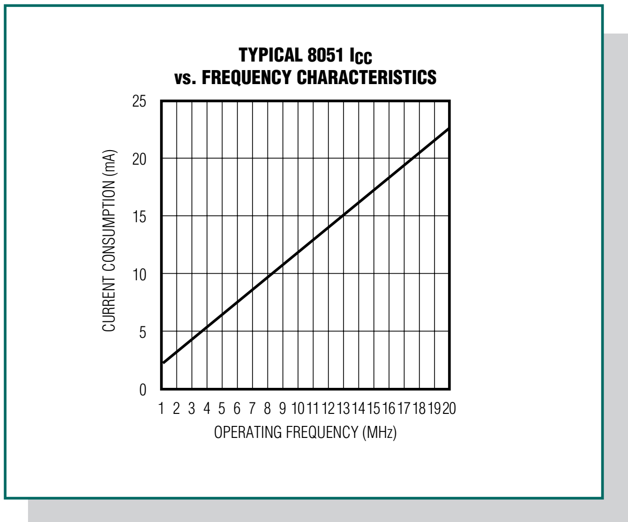


Figure 1. Typical power curve for generic 8051 microcontroller.

High-speed core

The most direct approach to decreasing power consumption of an 8051-based design is to improve the efficiency of the microcontroller. The original design of the 8051 was based on a 12-clock, 2-fetch-per-machine cycle architecture. The high-speed microcontroller family, however, uses a 4- or 1-clock per machine cycle core. It is more computationally efficient and requires fewer clock cycles to execute an instruction, resulting in faster execution times and increased maximum clock rates.

Although the advantages of a high-speed core are usually considered in terms of performance, they have important implications for power consumption as well. When the instruction execution of the processor is optimized, it takes less time to accomplish the same task. Many portable products operate in a burst mode, characterized by brief periods of activity, such as recording environmental data or scanning a bar code, followed by long period of inactivity. Reducing the time that the processor must be active achieves a corresponding reduction in energy consumption.

Another consequence of this improved efficiency is that comparable performance can be achieved while reducing clock speed. If a redesigned core uses 4 clocks per cycle rather than 12, this means that the same work can be accomplished at a reduced crystal speed. Because power consumption is directly proportional to crystal speed, power consumption can be reduced without sacrificing performance.

Figure 2 shows the power consumption of three microcontrollers performing the same task with the same level of performance. Two microcontrollers are standard 80C3x derivatives operating at 12 external clocks per machine cycle, while the second is a DS80C320 microcontroller operating at 4 clocks per machine cycle. Current consumption was measured for all devices and then compared, assuming a conservative performance improvement of 250% (2.5x) for the DS80C320. As is evident from the figure, the reduced clock per cycle core exhibits a significant current reduction at the same throughput, most notably at high performance levels.

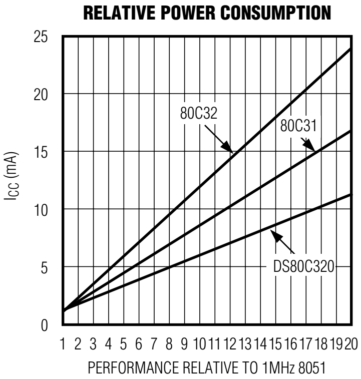


Figure 2. Reduced clock cycle core uses less current for same throughput.

Integration

Integrating peripherals on-chip is a method of power conservation. When driving a signal off-chip, the generating device must contend with the switching power required to drive the external loads and any DC losses. Switching power, PSW, is the power consumed when a digital signal changes. The switching power can be approximated as follows:

$$P_{SW} \alpha CV^{2}/T \qquad (1)$$

where C is the lumped capacitance of the receiving gate input buffer and the interconnection between the two gates, and T is the clock period of the signal. A typical input capacitance for a CMOS input is 10pF. Although it is difficult to calculate an exact value of the switching power for a system, it is obvious that each additional external load or pin that the microcontroller must drive consumes additional power.

Microcontroller-based systems typically use a number of peripherals. These range from external UARTs and power-on reset circuitry to watchdog timers. One of the strengths of the 8051 product family is the large number of peripheral functions that are available on-chip. In addition to simplifying a design by eliminating components, integrated peripherals also can reduce power consumption. One can assume that the core functionality of any peripheral consumes the same amount of power whether located internal or external to the processor. Locating a peripheral on-chip, however, will eliminate the switching power losses associated with driving an external bus.

Internal program memory

Another 8051 feature that is not commonly perceived as a peripheral is program memory. All 8051 derivatives incorporate various amounts of on-chip program memory. This is desired by many system designers as a method of reducing the component count and board area, but it also improves battery life in portable systems. As mentioned previously, this will reduce power consumption by eliminating the need to drive an external bus. There is an additional power savings when using on-chip memory. The 8051 architecture requires the use of a 74373-type latch to demultiplex the lower byte of address. **Figure 3** compares the use of internal vs. external program memory. The first uses a DS87C520 High-Speed Microcontroller with a 74AC573 latch and a 27C256 EPROM with an access time of 70ns. The second system uses the same microcontroller, but operating from internal memory. Both systems are operating at 11.0592MHz, executing a short, generic program. From the figure, it is apparent that as much as 49mA can be saved at high frequencies by eliminating the external EPROM and latch from the system.

Internal data memory

As previously mentioned, the use of on-chip memory instead of external RAM will save power. The enlarged scratchpad of the 80C32 derivatives (256 bytes) is sufficient for stack operations and some data storage in small programs, eliminating the need for external RAM.

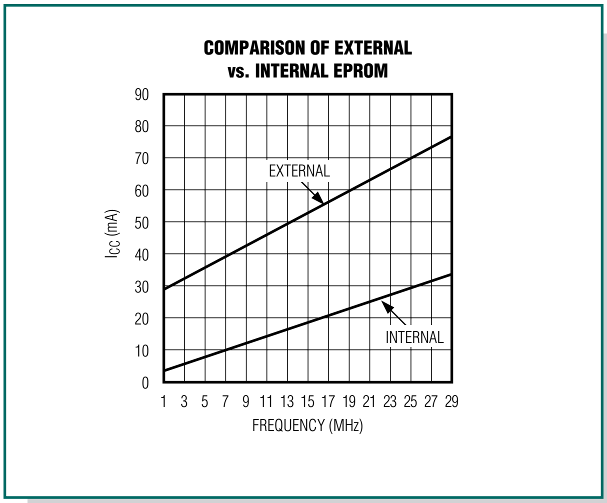


Figure 3. Using internal memory significantly reduces current consumption.

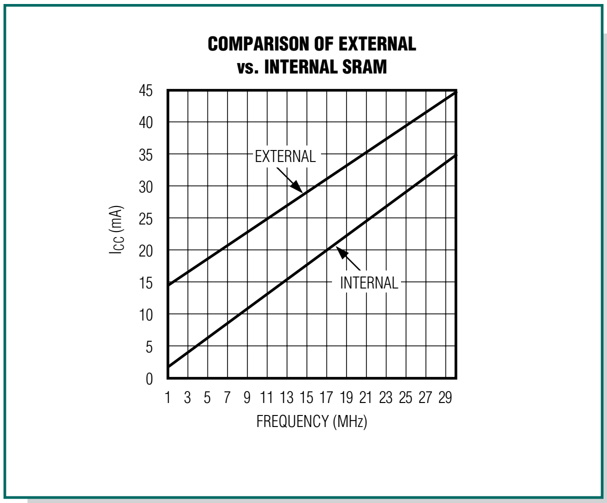


Figure 4. Eliminating internal SRAM and latch saves power.

For designs requiring more data memory or needing to implement an external stack, however, additional SRAM may be required. Although low-power SRAMs are available, their power consumption must also include that associated with a 74373 series latch as well as capacitive losses driving the external bus. This can be mitigated by using devices with expanded on-chip RAM. **Figure 4** shows the power consumption of two systems using SRAM mapped into the 8051 MOVX data space. The first uses a DS87C520 high-speed microcontroller with a 74AC573 latch and a DS2064 SRAM. The second system uses the same microcontroller but uses 1Kbyte of internal MOVX data memory. Both microcontrollers are operating at 11.0592MHz, executing a short, generic program that reads and writes to MOVX data memory. From the figure, it is apparent that as much as 9mA can be saved at high frequencies by eliminating the external SRAM and latch from the system.

Clock source

Another critical system component from a power standpoint is the clock source. Standard 8051 designs typically either excite an external quartz crystal with an internal oscillator amplifier or use an external crystal oscillator. If an external crystal oscillator is used, the waveform of the clock can affect power consumption. The input stage of the XTAL1 pin, used to drive external clock signals into an 8051, typically employs complementary drivers. As the input clock transitions between high and low, the drivers will momentarily both be on, causing a significant current rush. With a square wave, the transition between high and low states is almost instantaneous, and the time in which both devices are on is minimized. A waveform with a slower rise and fall time, such as a sine or triangle wave, will take longer to complete the transition and will spend more time with both drivers on. This will increase current and power consumption.

Figure 5 shows the relationship of current consumption to the waveform shape. The clock source was a programmable waveform generator, with the ability to output sine, triangle, or square waves. The current consumption shown is an average of four devices, both traditional and high-speed core. The comparison shows that current consumption is directly proportional to the rise (and fall) time of the clock waveform. The triangle wave has the lowest slope and the square wave the highest slope. The square wave averages 0.75mA less than the triangle wave. This implies that current consumption in external clock oscillator designs can be reduced by using oscillators with fast rise and fall times. This becomes even more important at lower frequencies, when the device spends more time in transition.

Some 8051 derivatives incorporate an on-chip internal ring oscillator. This is typically a chain of inverters that propagates a pulse around it. This provides an internal clock source of approximately 2MHz to 4MHz, capable of operating the device. Because it does not require the use of a crystal, it is a very-low-power clock source. Characterization of a DS87C520 high-speed microcontroller shows that operation from the ring oscillator can deliver performance comparable to a 7MHz 8051 at approximately 3.6mA. Although ring oscillators do not exhibit the stability of piezoelectric crystals, their low power and negligible power-on delay make them a significant factor in a power management scheme.

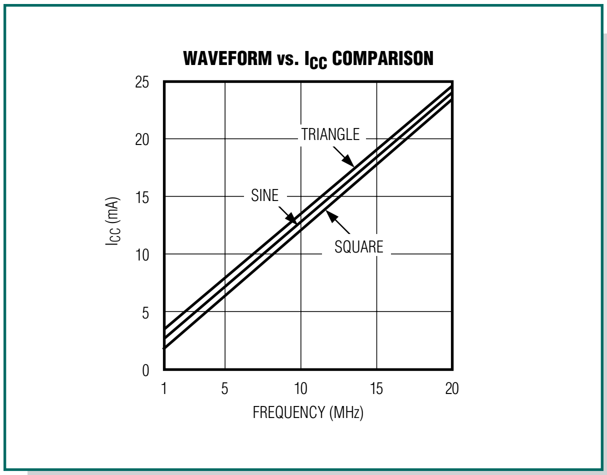


Figure 5. Oscillator waveforms with sharper edges reduce power consumption.

Clock management

As mentioned previously, the operating frequency of the microcontroller is the single largest factor affecting the power consumption of the device. Although the system clock frequency is primarily a hardware function, the 8051 has the ability to exercise limited control over it. These methods rely on slowing or halting the internal operating frequency of all or part of the device. Traditional 8051 architecture has used two clock control modes: Idle and Stop.

Improving stop mode

Stop mode is the lowest power state available to 8051 designers. In this mode, the internal crystal amplifier is stopped, halting operation of the device. Exiting from Stop mode is typically initiated by an external reset. Some variants also support exiting from Stop mode using external interrupts.

One of the disadvantages associated with Stop mode is the power consumed during the "dead time" while the crystal is resuming operation. A crystal oscillator relies on the motion of a quartz crystal for its operation. Physical limitations require a finite amount of time for the crystal oscillation to achieve sufficient amplitude for device operation. This warm-up period is encountered regardless of whether the clock source is an external crystal and internal crystal amplifier, or whether an external crystal oscillator is used. This time can be on the order of 3ms to 12ms, depending on the characteristics of the crystal and associated amplifier.

The effect of the warm-up period on power consumption is that while the device is not performing any useful work during this period, it is still consuming power. This can become significant if the device is entering and exiting Stop mode frequently, or is exiting Stop mode to perform short tasks. In fact, if the task is very short (<5ms), the crystal restart period can consume more power than the task itself. If a ring oscillator is used to perform a "quick start" from Stop mode, this delay can be avoided. This will greatly reduce the amount of power when out of Stop mode.

Figure 6 shows the operation of two systems exiting from Stop mode and performing a short task. One device incorporates an internal ring oscillator, and the other uses a traditional external crystal. The device without the ring oscillator must endure a crystal warm-up period. During this time the device continues to consume power, but no useful work is done. The second device is a DS87C520 high-speed microcontroller that incorporates an internal ring oscillator. This allows the device to resume operation immediately when exiting Stop mode. In this example, the routine to be executed is less than 4ms in duration at approximately 2MHz. As can be seen in the figure, energy consumption can be greatly reduced by using a ring oscillator to perform short tasks when exiting Stop mode.

In some applications, the stability of a crystal oscillator may be required shortly after exiting Stop mode. In this case, the ring oscillator can still be advantageous. Immediately upon exiting Stop mode, the device should restart the crystal oscillator. The device can then initialize any data or registers necessary while the crystal is still warming up. Most high-speed microcontrollers incorporate a status bit that indicates whether the crystal oscillator has stabilized or not. Once the initialization routine for the crystal oscillator code is complete, the software can poll the status bit to determine when the high-precision timing operation can commence.

Another method of improving Stop mode efficiency is to use an interrupt to exit instead of reset. This allows the processor to resume operation immediately with the instruction following the setting of the STOP bit, instead of having to restart from the reset vector. This eliminates the need to determine the cause of the reset and allows the processor to begin performing useful work in less time.

Idle mode

Idle mode is the second clock management mode used in original 8051 architecture. This mode halts operation of the CPU, but keeps the on-chip, general-purpose timers operational. In a power-sensitive application, these timers are used to periodically wake the device to perform a task or to poll if a task should be performed.

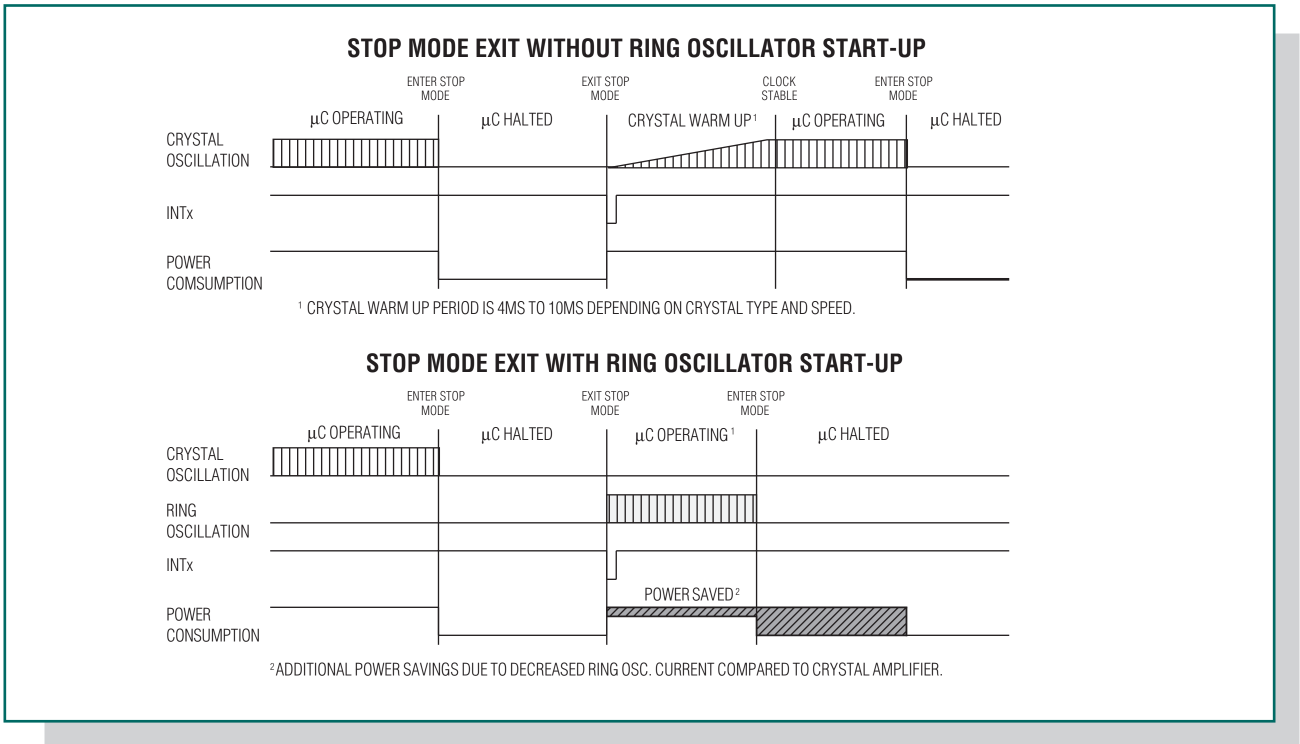


Figure 6. Comparison of stop mode exit with and without ring.

Because standard 8051 timers are limited to 16 bits, this allows a maximum timeout period of 31ms at a clock rate of 16MHz. If longer periods are needed, multiple timer overflows will be required. This will consume additional power because the device must resume full operation occasionally to increment a counter but not perform any useful work.

For longer periods, use an internal timer with a longer period. Some 8051 derivatives incorporate a watchdog timer, which can also be used to awaken the device. Watchdog timers can be programmed for long time-out periods, on the order of 226 clock cycles. This would allow a maximum timeout period of 4.2 seconds at 16MHz. As an example, assume an application wishes to awake from a low power state every 3 seconds to perform a task. If the internal timers are used to time the operation, the device would have to exit Idle mode 96 times without doing useful work. If a watchdog timer with a long timeout is used, the device would only exit Idle mode once, perform the task, and return to the low-power state.

One other option is to use a microcontroller with a real-time clock (RTC). The DS87C530 high-speed microcontroller incorporates an RTC capable of generating an alarm period as long as 24 hours. The internal interrupt generated by this alarm can cause the device to exit Idle or Stop mode. Using an RTC to exit Stop mode is the most efficient way to suspend device operation for long periods of time.

Power management modes

Although Idle mode reduces power consumption by halting program execution, the internal timers continue to operate at the external clock frequency. This consumes a considerable amount of power, considering the timers are basically operating in a "standby" capacity.

A better approach is to reduce the clock rate of the entire device. This can be done with an internal clock divisor, which divides the external clock frequency before it enters the CPU. Such a scheme has been implemented in the DS87C520 High-Speed Microcontroller. This device employs two clock divisor functions: Power Management Mode 1, which divides the input clock source by 64, and Power Management Mode 2, which divides the input clock source by 1024. These modes are enabled by setting the appropriate bits in a Special Function Register.

Figure 7 shows a comparison of the clock divisor and clock control modes on the DS87C520 High-Speed Microcontroller. The figure contrasts the current consumption in full speed (divide by 4), Power Management Mode 1 (divide by 64), Power Management Mode 2 (divide by 1024), Idle mode, and

Stop Mode. As expected, Stop mode draws the least current, because all internal clocking is halted. One interesting result of this comparison is that the two power management modes draw less current than Idle mode. This not only allows the device to conserve power, but permits it to function continuously at a low level of operation. In the traditional 8051 architecture, performing any type of CPU activity was "all or nothing." A device was forced to operate constantly at the highest performance level, even if high performance was only required for short periods. This unnecessarily increased power consumption. The use of power management modes (PMM) allows the device (and system) to match its power consumption to the level of performance needed.

Using interrupts with PMM

One possible consequence of using an internal clock divisor is that interrupt latencies may be greatly increased. In addition, slowing the internal timers would affect the ability of the 8051 serial ports to generate or synchronize with a standard baud rate. This could seriously interfere with the device's ability to respond to external stimuli. One solution is to incorporate a feature that automatically restores the device to full operation when an external interrupt or serial port activity is recognized. Such a mechanism has been implemented in the DS87C520. That device's Switchback feature allows the device to respond quickly to external interrupts. As soon as the interrupt is acknowledged, the device will automatically switch back to full speed (divide by 4) without software intervention.

The serial ports operate in a similar fashion. Upon receipt of a falling edge (start bit) on a serial port reception pin, the device will automatically switch back to full speed (divide by 4). Because this happens immediately at the start of the transmission, the device will be at full speed to correctly receive the rest of the transmission. With a traditional 8051 architecture, the only way to use the serial ports in a low-power configuration was with the Idle mode. The use of Power Management Modes provides a lower power alternative.

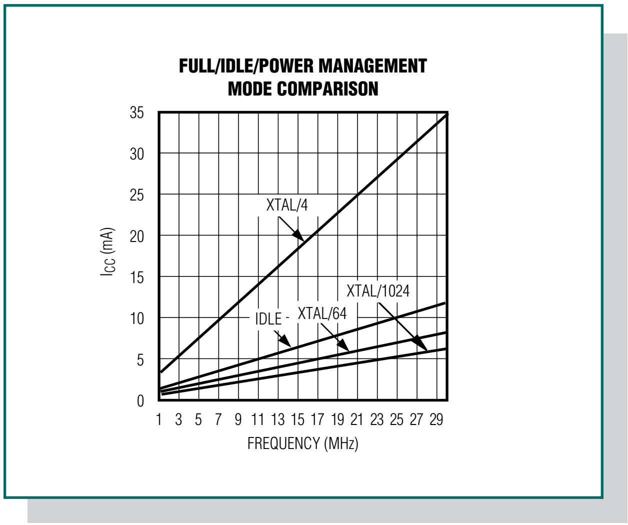


Figure 7. Full/Idle/Power Management Mode Comparison

Improving burst mode operation

A common mode of operation in power-conscious designs is to have the system awake from Stop mode, perform a burst of activity, and then return to Stop mode. One way to decrease power consumption in such a system is to increase the operating frequency. At first, this may seem counter-intuitive. For the time in which the device is operating, it will consume more power than a system operating at a lower frequency. The quiescent current consumed while the system is operating, however, is not a function of frequency. In the final system design, energy is typically evaluated to determine battery life. This distinction is important when evaluating a high-performance microcontroller because it combines the concept of time and processing power. If the product of power and time is smaller for a given system, then it will require less energy, regardless of the individual terms. In many instances, it can be shown that a high-speed microcontroller can actually reduce energy consumption by running fast for short periods, as opposed to running more slowly for a longer period of time.

This can be demonstrated by re-examining Figure 7. Assume that upon resuming from Stop mode, a DS87C520 must read an I/O port, perform a mathematical computation, and output the result to another port, requiring 500 machine cycles of CPU time. From the figure, the current consumption is 12.4mA at 10MHz, and 34.6mA at 30MHz. **Table 1** summarizes the results of the task at both speeds. As can be seen from the table, operation at 30MHz is the most energy efficient, with a more than 6% reduction in energy consumed.

Hurry up and wait

In many applications, the time out of Stop mode is not entirely speed-dependent. Frequently, a device will have to access a peripheral with a fixed response time, such as an A/D converter or thermostat. In such a case, the microcontroller will have a burst of activity, typically to initiate a process, followed by a period of little or no activity. In such a case, a combination of power conservation techniques can be effective.

Table 1. Energy consumed vs. processor speed for a 500 machine cycle task

Clock Frequency	Machine Cycle Period	Machine Cycles Required	Total Time	I_{CC}	Current Time Product
10MHz	400ns	500	200s	12.41mA	2.48As
30MHz	133ns	500	66.5s	34.66mA	2.30As

A practical example can illustrate the advantage of a high-speed microcontroller with PMM in such a system. Suppose that the DS87C520 is interfaced to a DS1620 digital thermometer and thermostat. This device is addressed serially using a standard 8051 serial port operating in mode 0. A host processor will occasionally wake the DS87C520 from Stop mode using an external interrupt and request that it read the temperature from the DS1620. After the data has been retrieved, the DS87C520 will store it in internal memory to be transmitted later. The DS1620 functions similarly to many A/D converters: a command is issued to start a conversion, then there is a delay while the conversion is completed, then the data is shifted out. In the case of the DS1620, conversion time is approximately 1 second. The device is polled to determine when the conversion is complete. The DS87C520 is well suited to such a task because it can perform the initialization and computation functions quickly. The device can then place itself in PMM while waiting for the conversion to complete. In a conventional 8051, Idle mode would be used to place the conventional 8051 in a low-power state once the conversion was started. The use of this mode allows an internal 16-bit timer to measure the conversion period. Operating at 16MHz, the conventional 8051 could require exiting Idle mode as many as 32 times before the conversion was complete.

One further improvement can be made in this example. Because the DS1620 is addressed as a synchronous device, high precision timing operations are not required. As a result, the microcontroller can operate from the ring oscillator while initiating and when reading the results of the conversion. This results in further power savings by eliminating the dead time needed to stabilize an external crystal.

Figure 8 illustrates the operation of two 8051 systems implementing the "hurry up and wait" schemes

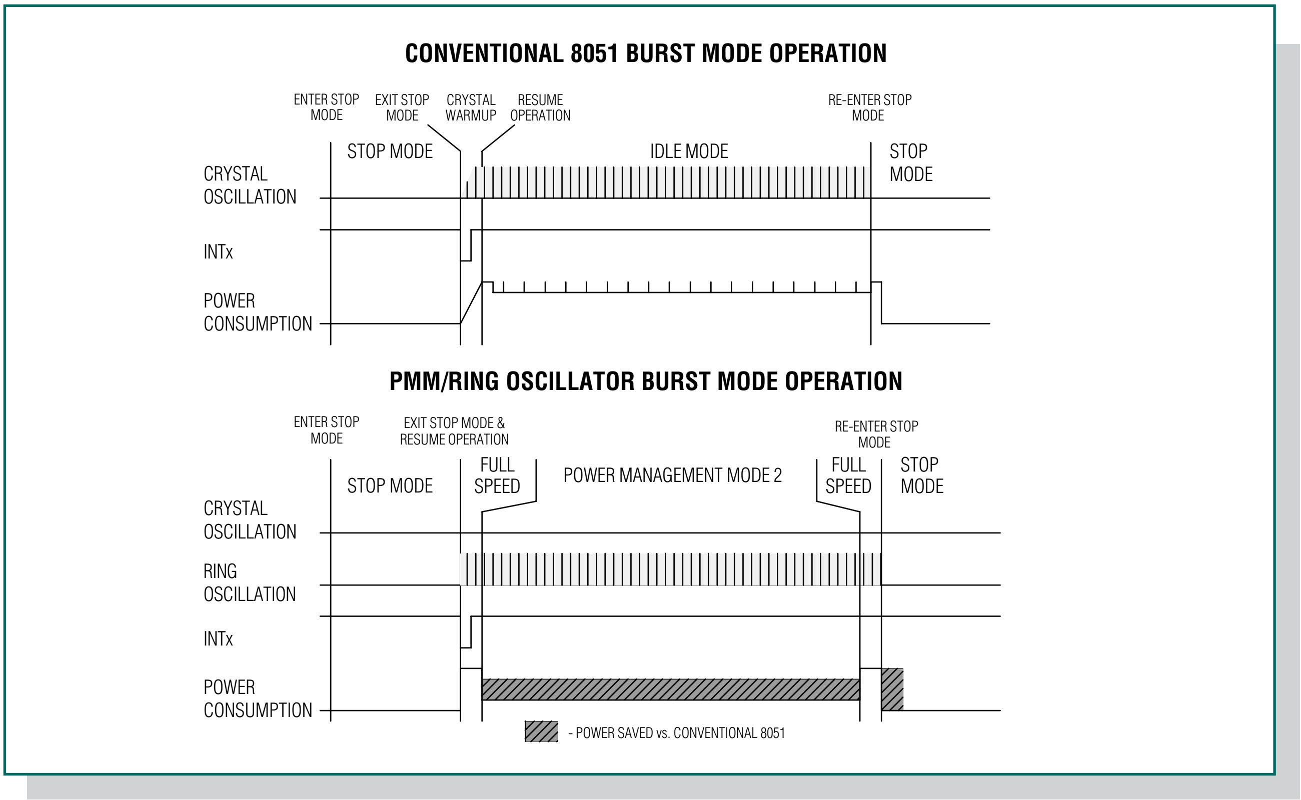


Figure 8. Implementing an 8051 "Hurry Up and Wait" scheme.

mentioned above. As can be seen from the figure, there is a significant power savings during program execution following the exit from Stop Mode. In addition to the power saved by using PMM2 instead of Idle mode, the elimination of the crystal warm-up period means that the routine can return to Stop mode more quickly. Running from the ring oscillator during the 1-second conversion delay slows the processor speed even more, allowing for greater power savings.

Summary

The 8051 microcontroller family remains one of the most popular processors in the world. Its ease of use and relatively high performance make it ideal for many applications, including portable and handheld products. The introduction of Dallas Semiconductor high-speed microcontrollers allows a way for existing 8051 designs to improve their power efficiency without a costly redesign.

The benefits of the high-speed microcontrollers that reduce power consumption can be summarized as follows:

- A high performance CPU allows the processor clock to be slowed, resulting in the same level of performance at less power. Alternatively, the performance of an existing system can be increased without increasing power consumption.
- The high-speed microcontroller incorporates features such as watchdog timers, additional UARTs, and precision reset circuits. External components consume more power.
- The introduction of two new low-power modes provides a low-power alternative to the Idle mode. In addition to reducing current consumption, power management modes such as those used in the DS87C520 allow the processor to perform tasks such as polling while in a low state. Conventional 8051 architectures require the processor to operate at the maximum clock rate, even if only minimal processing power is required.
- The benefits of a programmable clock rate and high-performance core can be combined with the Stop mode to greatly reduce power consumption. Examples have been presented that show how energy consumption can be reduced by matching the clock rate of the device to the desired performance level.

Behind the light show in optical transceivers

In recent years, news about communications networks technology always seems to involve some pronouncement on the urgent need for more bandwidth. The facts still bear repeating: an ever-growing number of people with telephones, faxes, modems, and computers are, through the exchange of terabytes of digitized information (videos, images, modeling procedures, as well as data and voice) demanding a larger share of the carrier spectrum. In response, high-tech communications companies who thrive on growth are competing to feed this appetite for bandwidth. Over the past decade, major resources have gone into developing fiber-optic networks in which light waves transport information at rates of gigabits per second through optical fibers finer than the human hair.

The stakes are very high. In a May 2000 news release, The Aberdeen Group (Boston, MA), an IT consulting firm, predicted that "The optical network market, excluding SONET elements, will grow to $17.7 billion by 2003. The suppliers that can deliver the technologies that solve the problems that carriers face will be the ones to succeed."

The use of the plural in "suppliers" and "technologies" highlights a key issue in this article.

The burgeoning communications network is highly complex. While a few large companies tend to dominate the global deployment of the optical network, behind the scenes there is a fusion of technologies developed by multiple companies, each with specialized technological expertise. Dallas Semiconductor falls into this latter category; we have designed a family of variable resistors especially for optical transceiver modules. A look at where Dallas' resistors fit into the grand scheme of communications networks reveals something about the way the communications industry develops solutions.

Identifying the big picture

Optical transceiver modules are designed and built by a variety of manufacturers. Applications for the modules include Synchronous Optical NETwork (SONET) and Synchronous Digital Hierarchy (SDH), Asynchronous Transfer Mode (ATM), Fiber Distributed Data Interface (FDDI), Fibre Channel, Fast Ethernet and Gigabit Ethernet. The names of these systems reflect the range of internationally defined transmission protocols and standards. On the other hand, the modules themselves were initially developed without definitive physical characteristics.

Recognizing the need for conformity if their products were to succeed, a group of manufacturers banded together and developed a multi-source agreement (MSA) for transceiver modules in 1998. The group consisted of AMP Incorporated, Hewlett-Packard Company, Lucent Technologies Microelectronics Group, Nortel (Northern Telecom), Siemens AG-Fiber Optics, and Sumitomo Electric Lightwave Corp. These parties agreed to cut the size of their modules in half (to 0.535 inch in width) and specified a set of module packages and pin-outs that would be interchangeable among the variety of RJ-45-style (including duplex LC, MT-RJ, and SC/DC) optical connectors used in high-speed fibre-channel applications.

Currently, a new consortium is drafting a new MSA for transceiver modules, reflecting a larger contingent of manufacturers and a new generation of modules. These multi-source manufacturers now include Agilent Technologies, Glaze Network Products, E2O Communications, Finisar, Fujikura Technology America, Hitachi Cable, Infineon Technologies, IBM, Lucent Technologies, Molex, OCP, Picolight, Stratos Lightwave, Sumitomo Electric Lightwave, and Tyco Electronics. The module specification is now called small form-factor pluggable (SFP) and covers expected transmission rates of up to 5.0Gb/s. The specifications reflect the industry's drive for high-density signal transmission in hot-pluggable modules of smaller size and higher speed.

To find where our resistors come into the picture, it helps to understand some basics about the transceiver module. The module converts incoming light waves to electrical signals and outgoing electrical signals back to light. Of fundamental significance, the optical transceiver is based on semiconductor laser technology. The module is a printed circuit board (PCB), and the optical source for the coveted bandwidth is a tiny semiconductor chip: a light-emitting or laser diode. At frequencies in the near-infrared spectrum, the laser's output can be modulated in tens of GHz, a capacious bandwidth.

The following briefly summarizes a signal path through the transceiver module. The receiving port connects to incoming light fibers. A photodetector diode converts the light to electrical signals, which are then amplified so that clock and data signals can be recovered, demultiplexed, and sent out through the electrical interface. The photodetector requires an automatically power-controlled bias circuit to provide a constant operating voltage (see **Figure 1**). Meanwhile on the transmitting side of the module, electrical clock and data-bit signals are synthesized and latched and sent to the laser driver. Finally, the laser driver sends the signal as electrical current to the laser diode, which converts electron energy to light.

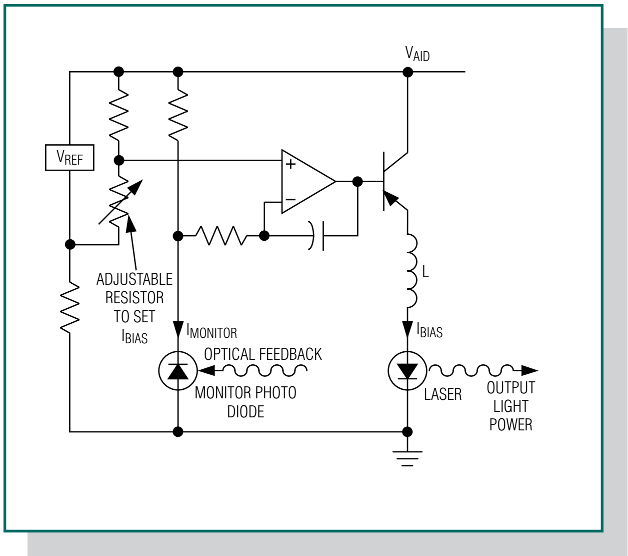


Figure 1. Average Power Control Loop—Typical average power control circuit using a monitor photodiode and adjustable resistor to set bias current.

In some designs that use laser diodes, a photodetector monitors the laser diode output and, in a feedback loop, reconverts the light back to electrical circuits that measure the laser's actual output power. This feedback stabilizes the laser output power. The optical feedback is a complicating drawback to this design. However, the latest laser technology, vertical cavity surface emitting lasers (VCSELs), often do not require a photodetector because of exceedingly low current.

The laser driver must do two things: it must maintain a consistent DC-bias current to set the laser operating point, and it must maintain a modulation current to carry the signal. As manufacturers strive to increase signal throughput in transceivers, the laser source must be carefully characterized for operating constants in order to control the light output.

The laser diode and VCSEL

The Fabry-Perot type of laser diode emits a coherent light beam from the narrow, beveled edge of the chip, with reflecting mirrors incorporated at the edges or stationed outside the chip. For the future of the communications industry, however, a more promising laser source is the VCSEL. As its name suggests, the VCSEL vertically emits the laser beam from a circular cavity 5 to 25 micrometers in diameter at the top (the bottom is a future possibility) of the chip. The mirrors are incorporated as an integrated array on both ends of the cavity—a design known as a "distributed Bragg reflector." In the future, parallel optical interconnects using multi-element VCSEL arrays could enable terabyte throughput.

Academic and corporate institutions are vigorously developing VCSEL designs for more widespread deployment. Compared to edge-emitters, the VCSEL requires less current and has a lower lasing threshold (1mA or 2mA versus 30mA). At this level, simple current control is often sufficient without the extra photodetector to monitor output. The VCSEL's emitting aperture is measurably larger, which means that the output beam's angle of divergence (a measure of dispersion) is significantly smaller. There are several manufacturing and processing advantages, as well. The die is much smaller, allowing more VCSELs to be packed on a wafer with more interconnects; all the VCSELs on an entire wafer can be tested at once. Lastly, the VCSEL is more robust in operation than a laser diode, with a longer life expectancy and lower failure rates.

Whether laser diode or VCSEL, the laser emitter in any optical transceiver is a semiconductor whose photoelectric effects depend on the interplay of current, voltage, and resistance. Some of the following factors affect safety and performance:

- Laser output is exceedingly sensitive to temperature.
- Laser power output tends to change over the life of the laser and this aging increases with temperature.
- As VCSELs operate at significantly lower power and temperature than diodes, the failure rate over time is proportionately lower.
- The laser emitter needs to be protected from random power transients as well as transients during power-on and power-off.
- Even though a laser's near-infrared light is invisible to humans, a beam entering the eye is still focused on the retina and can cause permanent damage. Because of potentially serious effects on personal safety and laser function, regulations require that laser power output be limited to a few hundred microwatts.

Controlling laser current is not only a safety issue in VCSELs and laser diodes, but it is also a factor in performance. As is consistent with semiconductor behavior, the VCSEL's maximum output power increases linearly with decreasing temperatures; conversely, the output wavelength increases with increasing temperatures. In short, controlling current in response to temperature is important in controlling performance.

From the big-picture viewpoint, we can delineate the accumulated facts as follows:

- Exploding demand for bandwidth leads to the development of optical networks.
- Optical networks use optical transceiver modules to physically convert optical and electrical signals.
- Manufacturers of transceiver modules are driven to decrease physical size and increase signal throughput to multiple gigabits per second.
- Transceiver modules use photodiodes to receive light signals and laser diodes or VCSELs to send light signals.
- As data rates continually increase, modules' photoactive components require ever more precise, reliable power control in order to prevent laser failure, prolong life expectancy, and/or operate within desired output parameters.

This finally brings us to Dallas Semiconductor's variable resistors. The way to control current through laser diodes and VCSELs, and thereby to control power output, is to control resistance. At one time, a human technician had a full-time job manually adjusting the "trim" potentiometer, trying to get a good "eye pattern." A better solution to this control and tuning problem is an electronically programmed device, which can respond to temperature changes.

A niche industry

While not in any strict sense one of the communications companies, Dallas Semiconductor brings expertise to the table in several pertinent technologies: digitally controlled variable resistors and potentiometers, EEPROM, temperature sensors, and extremely low-power CMOS methods. In response to the needs of gigabit optical technology, Dallas has produced a product family with a new range of features.

With the DS1845 Dual Potentiometer with EEPROM, Dallas designed the semiconductor industry's first potentiometer with integrated memory, specifically for service in pluggable gigabit transceiver modules. The DS1845 combines two linear-taper potentiometers with 256 bytes of EEPROM, which is required by the MSA standards. The higher-resolution, 256-position potentiometer can be used to control modulation current and the 100-position potentiometer can be used to control bias current. Users configure both outputs and store the wiper settings and required serial ID data in the on-chip, nonvolatile EEPROM memory for reference during operation.

In modules that aim to shrink into SFPs, more densely integrated components that combine memory and two separately configured potentiometers save space by replacing multiple parts. Furthermore, the DS1845's 2-wire interface meets the transceiver producers' requirement for in-circuit programmability and is compatible with existing 2-wire EEPROM.

To meet more a specialized need, Dallas developed the DS1846, which combines three linear taper potentiometers with nonvolatile memory and a CPU supervisor in reduced TSSOP packaging. This level of integration in such a small chip saves board space and cuts cost and procurement delays, expediting product development. As with the DS1845, the nonvolatile memory is used to configure and store application-specific calibration data. And, to control wiper settings for each potentiometer, there's also memory space available for user-specific data.

The DS1846's on-chip micromonitor tracks voltages. On detecting an out-of-tolerance voltage level, the micromonitor initiates and holds a system reset until safe operating conditions return. The micromonitor is programmable for various voltage levels and includes a manual reset.

The third potentiometer can be used to monitor another variable or to provide a coarse trim for one of the other resistors.

Intended for demanding laser applications, the DS1847 and DS1848 compensate for the laser's thermal characteristics over temperature ranges (see **Figure 2**). The DS1848 has an extra 128 bytes of general-purpose EEPROM; otherwise the two are alike. The chips store resistance characteristics relative to temperature in an onboard look-up table (LUT). An integrated temperature sensor constantly measures and reports temperatures during laser operation. The DS1847 or DS1848 compares the reading to the value stored in the LUT and adjusts resistance according to the designer's defined resistance characteristics. The value determined by the

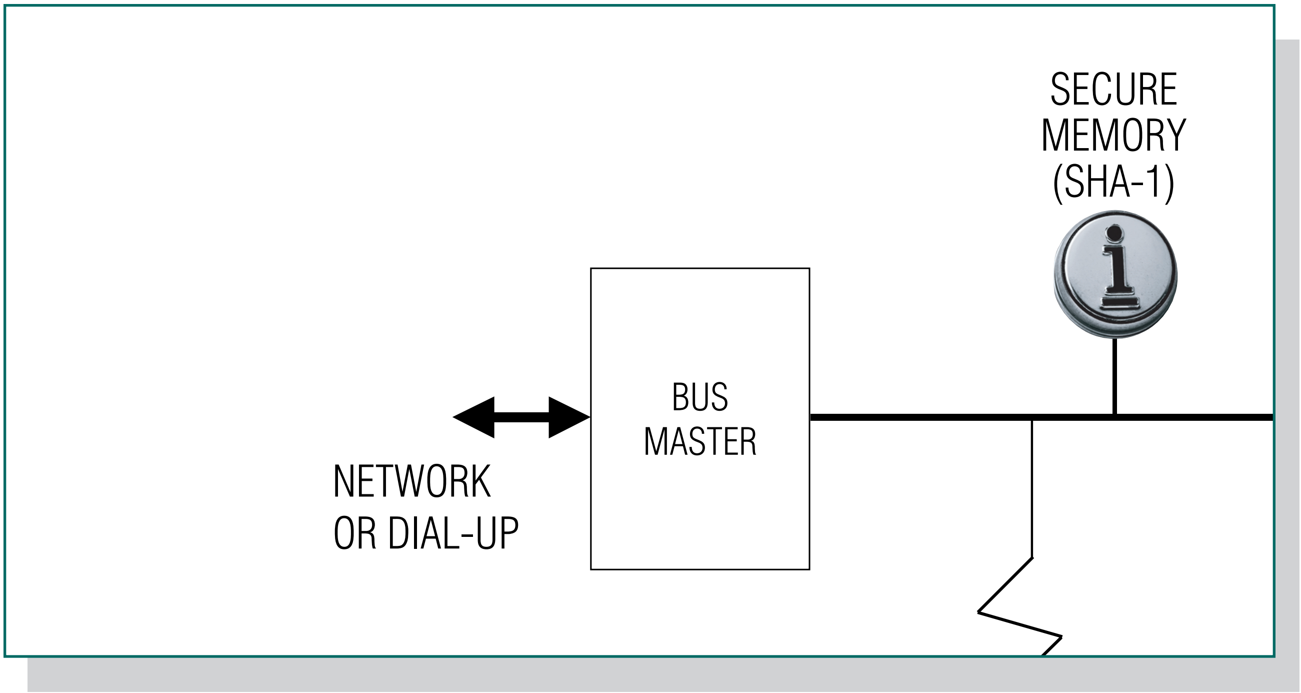


Figure 2. Optical Transceiver—Variable resistors, designed for optical transceivers like this one, automatically calibrate each diode more accurately than the older, mechanical-trim potentiometers.

temperature sensor is also stored in EEPROM (updated every 10msec), and is available to the user over the 2-wire bus. It should also be noted that the DS1847 and DS1848 operate automatically. As a temperature change is detected, control circuits automatically adjust resistance to achieve the compensating current value without any user intervention.

As a whole, all Dallas circuits employ a low-power CMOS technology that contributes to holding down a sensitive power budget. All circuits operate throughout the industrial temperature range and with both 3V and 5V power supplies.

Clearly, in a huge, complex, driven market like optical communications networks, many players operate at many levels. The success of the laser, the star in a new and exotic technology, can depend on a comparatively old and familiar player, the humble resistor. Of course, Dallas' caveat to the moral of this fable would be, "It's not your grandfather's resistor any more."

Securing electronic transactions using SHA-1

Increasingly, the world is turning to portable electronic devices with cryptographic capabilities to perform very secure network authentication, virtual private networking, vending system regulation, fare collection, and employee or citizen identification. The design of such a portable device (or token) requires careful study of the methods by which cryptographers authenticate and protect sensitive data and monetary information. Of course, an e-cash or identification token must be portable, durable, and secure, but it quickly becomes obvious that there are several additional cryptographic, electrical, and physical requirements that a secure token must satisfy.

The most critical function for a truly secure token is that of authentication. A token must be able to prove that it is authorized by the issuer (the service provider), and that it is authentic. A token can prove that it is authorized simply by virtue of the fact that it contains encrypted information that only the issuing authority could have created. Secure authentication, however, is a far more difficult issue. The device must be able to prove that it is not a fake or duplicate, and this requires some very special hardware functionality. A device will not be trusted if its design cannot be freely examined, so secrets in design or function are all but impossible to keep. If we assume that the inner workings of the device are published and widely known, then we must also assume that anyone with sufficient skill could build an emulation of the device and bypass some of the special controls that the device hardware imposes. Any token design based on security by obscurity is doomed to failure.

In cryptographic circles, truly secure authentication is handled by a method called challenge-and-response. This method involves a secret that is known only to valid tokens and hosts, and methods whereby tokens can prove that they know the secret, which proves that they are authentic. Of course, the secret must never be revealed in the process, and that is where a cryptographic concept called Zero Knowledge Proof comes into play. The token must support a mechanism where it can prove that it knows a secret without revealing any information about it. At first glance, this may seem impossible, but it is a common exercise in secure cryptographic systems. Here is how the scheme works: When a suspect token arrives, the host system creates a very big number, called a challenge, entirely at random, and sends it to the token. The token takes this challenge and, together with an internally stored secret, performs a complex mathematical operation on it. Then, it returns the result of the operation to the host (see **Figure 1**). The host, also knowing the same secret, performs the same special mathematical operation internally, and then compares the results. If the response from the token matches the one computed in the host, then the token has proven that it knows the secret, without revealing it (the essence of a Zero Knowledge Proof). Eavesdropping on this conversation is of no use to an attacker who does not know the secret. This is because the challenge is different each time; it is randomly generated. The next challenge can never be predicted. The secret remains safely hidden inside the token, and the host knows that the token is authentic (because only authentic tokens know the secret).

Of course, it is critical that the complex mathematical algorithm used is not reversible, since the attacker could run the operation in reverse and derive the secret. In fact, the choice of algorithm may be the single most important factor in judging the security of a challenge and response scheme. Electronic tokens have been created in the past that use home-brewed algorithms, stream-ciphers, rolling-codes, or cryptographic algorithms that have been curtailed or reduced for this purpose, but they have not been widely accepted. The problem is that without extensive peer-review, there is no assurance that these algorithms are secure against attack. And there is no guarantee that these algorithms have no “back doors” (intentional or accidental) that may be known only to the device manufacturer.

To make a truly secure cryptographic token, the algorithm selected must be one that is well-known, trusted, time-tested, and widely peer-reviewed in the global cryptographic community. An algorithm that takes input data and irreversibly creates a digest of that data is called a one-way hash function. One of the most studied and trusted one-way hash functions is SHA-1 (Secure Hash Algorithm). This government-approved (FIPS 180-1) algorithm is the basis of modern digital signatures and document protection schemes. It has a long history in the cryptographic community, is peer-reviewed, and is widely trusted.

However, SHA-1 is a complex algorithm that involves multiple 32-bit 5-way adds, complex logical functions, data shifting, and a great deal of repetition. Implemen-

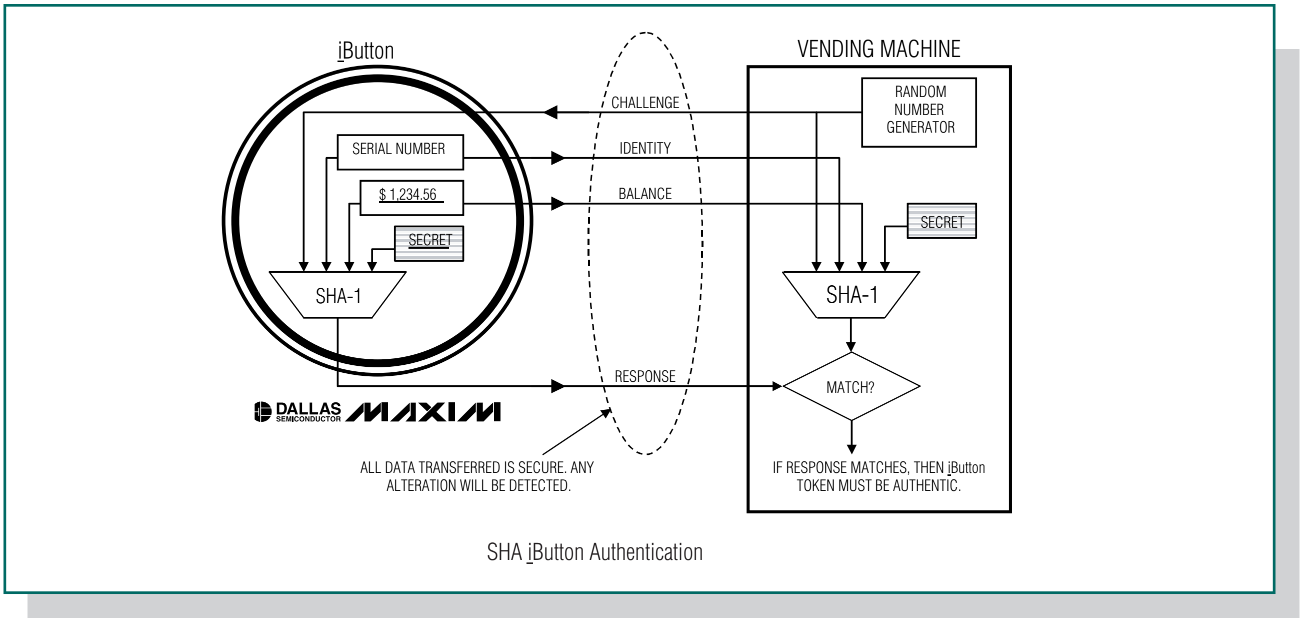

iButton
VENDING MACHINE
CHALLENGE
RANDOM NUMBER GENERATOR
SERIAL NUMBER
IDENTITY
$ 1,234.56
BALANCE
SECRET
SECRET
SHA-1
SHA-1
RESPONSE
MATCH?
DALLAS SEMICONDUCTOR MAXIM
ALL DATA TRANSFERRED IS SECURE. ANY ALTERATION WILL BE DETECTED.
IF RESPONSE MATCHES, THEN iButton TOKEN MUST BE AUTHENTIC.
SHA iButton Authentication

Figure 1. SHA iButton Authentication

tations of the SHA-1 algorithm in silicon have generally required large die areas and so made fairly expensive tokens. A new method has been devised to perform the algorithm in a serialized fashion, reducing the die area by a factor of 10 or more and making a reasonably priced SHA-1-based token possible.

Another critical feature of a truly secure token is a globally unique and unalterable identity. Including the unique token serial number as an additional input to the authentication algorithm binds the physical token and the contents together, making duplication of monetary value or credentials among tokens impossible.

When a secure token is used in monetary applications, the data it contains (presumably the account balance) is said to be dynamic, because the critical value is read, debited, and rewritten each time the device is used. This type of usage model introduces a type of attack known by cryptographers as a replay attack. That is to say, an attacker can record the value data from the token, and then deplete the token making legitimate purchases. He can then restore (or replay) the original data, and thereby restore the monetary value of the token to be spent repeatedly. This replay can be prevented only if the token provides mechanisms that make each instance of the data that it contains unique. The truly secure monetary token therefore provides a special counter. This counter is incremented each time the device is written to and cannot be wrapped-around, reset, decremented, or reloaded. Then, by including this counter value in the input to the authentication algorithm, the monetary data, the device identity, and the monetary instance are all bound together. Should the data be taken from the device and written back later, it will be found invalid because the counter will have changed. It's the same value, but it is not the same instance of the value.

An interesting feature of the challenge-and-response authentication process is that the data is also protected while en route from the token to the host. Since the data, the token identity, and the instance counter are all included in the SHA-1 algorithm input, any attempt to alter or inject data bits into the communications path between the token and the host will also render the transaction invalid. This means that any number of untrusted intermediaries can handle this challenge-and-response data exchange and the process remains secure. A distant server on the Internet can authenticate a user's token at a home computer through a myriad of routers, bridges, hubs, and eavesdroppers, and security is not compromised in the slightest bit.

Another important requirement for a secure authentication token is that it is able to protect the secrets that it holds. Plastic cards with embedded silicon chips are difficult to protect against physical attacks, and typical memory devices have no special provisions for secure storage areas where secrets might be kept. A secure token must provide a high level of physical security to protect the secrets it contains.

Cryptography also provides methods whereby the secret in each device may be derived from another master secret, which is not stored in the device, in combination with the unique device identity. This provides a unique secret for each device and prevents a system-wide break (called a class break) should the secret in any one device be compromised. Monetary systems, generally, also use one secret (stored inside the token) for token authentication, and another secret (not stored in the token) for validation of the monetary values stored in the token. This limits the scope of a physical attack on a token to just the ability to emulate that one, and only that one, token. This severely limits the profitability of a physical attack.

An often overlooked avenue of attack occurs at the service provider's facilities where an unscrupulous employee gains access to the critical secrets by which the scheme assures authenticity. To overcome this problem, cryptographers offer a method called Secret Sharing. The actual secret does not actually exist, but instead is the result of computations that combine two or more partial secrets. The service provider maintains these partial secrets on separate systems, at separate locations, and allows no one to have access to more than one of them. Without all of the partial secrets applied in the proper sequence, the actual secret cannot be computed. A truly effective, secure token must provide a means to combine the partial secrets entirely inside the token so that the final secret never actually exists where any human can ever observe it. As the token moves through the service provider's initialization process, it is injected with each subsequent partial secret, and the actual secret is being computed entirely inside each device. As a final step, the unique device identity is also injected into the process, so the resulting device secret is unique to it, and the master secret never actually exists where it could be compromised.

A secure electronic token that can be strongly authenticated also makes for a very secure key for door locks, access control systems, and equipment control lockouts. An electronic key using challenge and response cannot be duplicated or altered, and eavesdropping attacks are useless. In this application, the data is said to be static because, unlike an e-cash application, it does not change with each use. And the ability to have the token authenticate the person who presents it adds even more to the security of the system—even in systems that have no provisions for such PIN or password protection.

Peer-to-peer authentication applications allow networked appliances to authenticate one another. This protects them from being operated by outsiders. Tokens that perform the authentication quickly, and that protect the secrets inside the token, greatly reduce the physical security required around the appliance. The token can also be used to quickly generate unique session keys for the encryption of control commands or sensitive data.

The Dallas Semiconductor DS1963S iButton is a secure, electronic token that satisfies all of these requirements. Each token has a globally unique, factory-lasered 64-bit identity, 512 bytes of lithium-backed NV RAM data storage area, and eight protected 64-bit secrets. All of this is housed in a small, rugged stainless-steel container. Two-way communication with the device is performed on a single data conductor at up to 140kbps, and SHA-1 is performed internally in under 500 microseconds. The 64-bit secret size would require about 9,223,000,000,000,000,000 attempts to break by brute-force methods, making a brute-force attack unrealistic. The iButton can be worn like jewelry, attached to an ID card or badge, or carried like a key. It can be attached to a container or bin, to a product or shipping carton. It can also be embedded in a component or circuit board.

The Dallas Semiconductor DS1961S iButton is an EEPROM version of the DS1963S. It has 128 bytes of memory and provides a write-protection scheme where the host system must satisfy a SHA authentication before the device can be altered.

The DS1963S also has the ability to serve as a coprocessor in vending and toll systems where the processing power and speed at the host side is usually limited. Used as a co-processor, the DS1963S can safely store and protect the system secrets, perform the SHA-1 algorithm very quickly, and store the collected funds safely, as well as providing a globally unique serial number that identifies the fare box or vending machine. As a co-processor, the DS1963S iButton can also hold critical configuration and pricing data used by the host. The host electronics are reduced to simply moving data between iButtons. E-cash systems have been demonstrated that use all of the security features described herein and perform entire monetary transactions in under 50 milliseconds.

The DS1961S, which is also available in chip form, can also be simply embedded in networked devices (using only a single port pin for communication). This provides strong cryptographic authentication between networked devices, opening up new possibilities for interrogating and controlling them using the Internet. Networked devices can quickly and reliably authenticate one another and generate random number-based session keys for data encryption, all in a single exchange of messages. The system secrets are safely stored in protected EEPROM, and there is almost no overhead incurred to add strong cryptographic security to networked devices.

An EconOscillator to clock an 8051 microprocessor

Introduction

EconOscillators have an internal oscillator that provides a base frequency, and they use a built-in divider chain to lower a base frequency to the desired rate. Each part number can divide down four base frequency rates (60MHz, 66.67MHz, 80MHz, or 100MHz) for adjustments up to 2052 times slower than the base part frequency. EconOscillators can be used for any type of clocked logic, including microprocessor, FPGA, and CPLD circuits, depending on system requirements.

8051 microprocessors and RS-232 serial communication

When selecting a clock, two factors deserve careful consideration: clock rate and clock accuracy over the operational life span. In an 8051 microprocessor system, the use of RS-232 serial communications often determines the system clock rate. Consider, for example, asynchronous mode 1 serial communication using a 12MHz clock (the maximum clock rate for the original 8051). **Table 1** shows the Timer 1 auto-reload values required to establish standard baud rates.

The actual baud rates in Table 1 were calculated using the following formula (reproduced from Dallas Semiconductor's *High-Speed Microcontroller User's Guide*):

SMOD = Baud Rate Doubler

f_{OSC} = Oscillator Rate

TH1 = Timer 1 Auto-Reload Value

The table numbers are based on these Timer 1 conditions:

- Set to increment 12 clock cycles per timer (DS87C520 can increment Timer 1 each 4 or 12 clock cycles)
- Auto reload mode enabled
- Baud rate doubler (SMOD = 0) disabled

Most users of RS-232 serial communication agree that any baud rate error over 3% is likely to cause communication errors, which occur in spite of synchronization of the start and stop bits during data transfer. A 3% allowable error limits the maximum communication rate with a 12MHz crystal to 2400 baud—not bad for the early 1990s, but a little slow for today's standards. Luckily, there are crystals that cater to 8051 serial communications: either an 11.059MHz or 22.118MHz crystal. Microprocessors using these crystals see a considerable improvement in baud rate (**Table 2**), achieving data transfer rates up to 57.6kbps (115.2kbps with a DS87C520 using the baud rate doubler, SMOD = 1), which is respectable for most of today's microprocessor systems.

Using the DS1075 to clock an 8051 microprocessor

As mentioned above, the DS1075 comes in four base varieties, with the internal oscillators running at 100MHz, 80MHz, 66.667MHz, and 60MHz. Using the internal divider chain to slow them down enough for an 8051 application, in theory either of these parts could be used. However, if you plan to use the serial port of your 8051, you should select the base part that best fits your microprocessor's needs, depending on both the baud rate required and baud rate generation formula provided with your microprocessor.

In the case of the 8051 microprocessor in our example, oscillator frequencies of 11.059MHz and 22.118MHz were desirable, and an approximately 3% error rate was tolerable for baud rate generation. If you use the 66.667MHz base part, you would be able to divide the base frequency by six down to 11.111MHz. This has a small error from the ideal frequency of 11.059MHz (0.47%), and the error remains acceptably low even with a worst-case deviation of 1% from the programmed frequency. Thus, the DS1075-66 allows a maximum error of 1.47% from the desired frequency of 11.059MHz, which is adequate for communication at rates up to 28.8kbps.

Table 1. Baud rate and baud rate error using a 12MHz crystal for an original 8051 microprocessor

Timer 1 Auto-Reload Value	Actual Baud Rate (Desired Baud Rate)	Baud Rate Error
255	31250 (28800)	8.5%
254	15625 (14400)	8.5%
253	10417 (9600)	8.4%
249/250	4464/5208 (4800)	7%/8.5%
243	2404 (2400)	0.16%

Table 2. Baud rates generated using crystal frequencies selected for RS-232 serial communication

Timer 1 Auto-Reload Value	Baud Rate with f_{OSC} = 11.059MHz	Baud Rate with f_{OSC} = 22.118MHz
255	28,799.5	57598.9
254	14399.7	28799.5
253	9599.8	19199.6
250	4799.91	9599.83
244	2399.95	4799.91
232	1199.98	2399.95
208	599.98	1199.98
160	299.99	599.99
64	149.99	299.99

Note: Requiring the baud rate to be within 3% of the specified rate also places an accuracy requirement upon your clock. Even with an ideal clock rate selected for RS-232 communication, if the clock varies more than 3%, you may not be able to communicate consistently.

If you are using an 8051 with a higher allowable clock rate such as the DS87C520 (33MHz maximum clock rate), you could simply divide the clock rate by three to 22.222MHz. The maximum error is now 1.47%—still fine for communication at any of the 22.118MHz baud rates. The higher clock rate also provides a higher level of processor performance for your other application needs as well.

The big advantage in using the DS1075 for an 8051 design is flexibility. A design that started out using an original or equivalent 8051 microprocessor (12MHz maximum clock rate) can be simply upgraded by reprogramming the oscillator and replacing the microprocessor. Depending on the design, you may even be able to reprogram the DS1075 in-socket. If you are using the 40-pin DIP version of the 8051 microprocessor, Dallas Semiconductor and many other companies make several 100%-compatible replacement chips. The fastest of them all, Dallas Semiconductor's DS89C420, offers a 50x performance increase over the original 8051 design and has several resources available, including watchdog timers and power management that were not available on the original 8051. Other chips such as the DS87C520 can provide up to an 11x performance increase. Dallas Semiconductor also makes other versions of the 8051 that have PWM and A/D converters for control applications, but they are not available in the 40-pin DIP package.

Hardware setup

To use the DS1075, you will need to establish a way to program its EEPROM registers. The easiest way is to purchase the DS1075K Programming/Evaluation Kit. The kit's hardware and Windows® 95 software with samples provide an easy means to program the device and try it out in your application. Otherwise, the data sheet alone (available online at www.maxim-ic.com) has all of the information required to program the device without a kit.

Once the DS1075 is programmed, the schematic in **Figure 1** shows how to wire the DS1075 for proper operation with an 8051 microprocessor. Note that the output of the DS1075 goes into XTAL1, and XTAL2 is not connected. XTAL2 is normally the crystal oscillator output of the 8051. Connecting anything to this pin will just load the microprocessor down, which is not necessary when any auxiliary clocked device can be connected in parallel with the microprocessor on XTAL1. This is assuming that the joint loading of the auxiliary device and the 8051 does not exceed the output current specification of the DS1075.

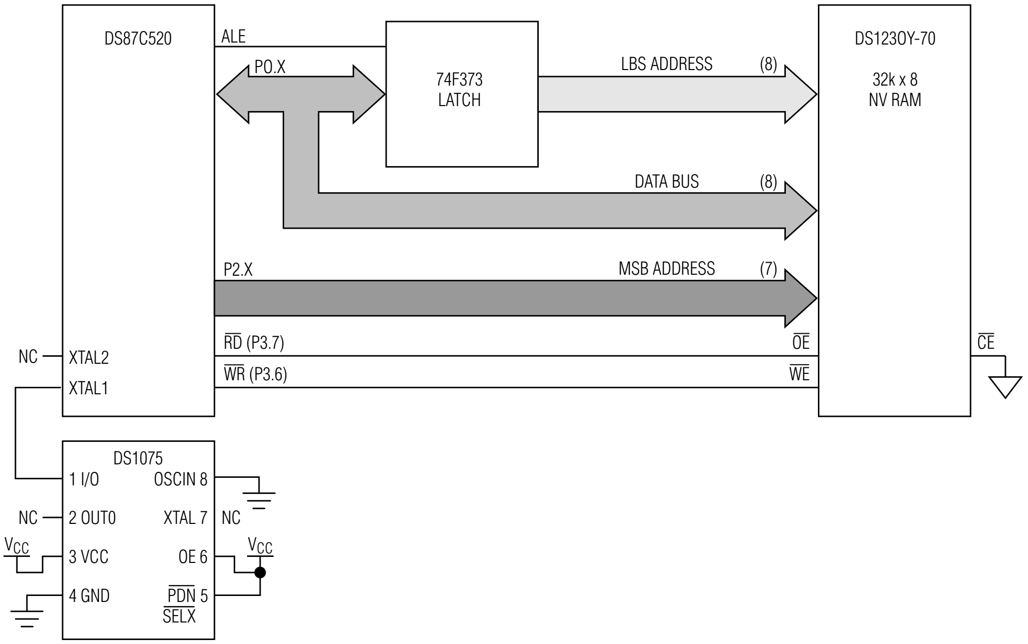


Figure 1. Hardware setup for using a DS075 oscillator to clock an 8051 microprocessor.

Windows is a registered trademark of Microsoft Corporation.

Dissecting the versatile Dallas 1-Wire network

The Dallas 1-Wire bus is a simple signaling scheme that performs two-way communications with peripheral devices over a single electrical connection. In any 1-Wire system, there is a single master and one or more slaves sharing a common data line. On this single data line are multiplexed address, control, and data information. Most of the devices operate entirely from power robbed from the data bus, although some opt to use local power if it is available. Charge is stored internally during periods when the data line is high, and the device operates using this charge during periods when the data line is low. An array of devices can be attached to a microprocessor on a single port pin. 1-Wire devices are available that store data (NV RAM, EPROM, and EEPROM), read and log temperatures and voltages, adjust resistance, count, control and sense, interface to other systems, and perform timekeeping and cryptographic functions.

The most basic feature that all 1-Wire bus devices share is that each device has a factory-lasered address (like a serial number) that will never be repeated in any other device. That is to say, every device is unique. This allows any single device to be individually selected from among many that may be connected to the same bus wire. Because one, two, or even dozens of 1-Wire devices can share a single wire for communications, a binary searching algorithm is used to find each device in turn. Once each device address is known, any device can be uniquely selected for communication using that address.

Electrically, the 1-Wire bus is a wired-OR configuration. A typical master consists of an open-drain pulldown and a resistor pull-up to 3 to 5 volts. Slaves have an open drain output and are only able to pull the bus down.

The 1-Wire data waveform is similar to pulse-width modulation. The bus master issues a reset (the longest low period) that synchronizes the entire bus. The master then initiates each bit time period, or time slot, and writes zero or one bits using wide or narrow pulse widths. To read data, the master initiates time slots using narrow pulses, and the slaves return a logical 0 bit by holding the line low and thereby extending the pulse, or a logical 1 bit by leaving the pulse unchanged.

Most 1-Wire devices support two data rates. The lower (standard) data rate is about 14kbps and the higher data rate is about 140kbps. An even higher data rate, 1Mbps, is in development. The protocol is self-clocking and tolerant of long inter-bit delays, making for easy operation in interrupted software environments.

The first part of any communication involves the selection of a slave device for subsequent communications. This can be done by selecting all slaves, selecting a specific slave (using the serial number of the device), or discovering the next slave on the bus using a binary search algorithm. Once a specific device has been selected, all other devices drop out and ignore subsequent communications until the next reset is issued.

Once a device has been isolated for bus communication, the master can issue device-specific commands to it, send data to it, or read data from it.

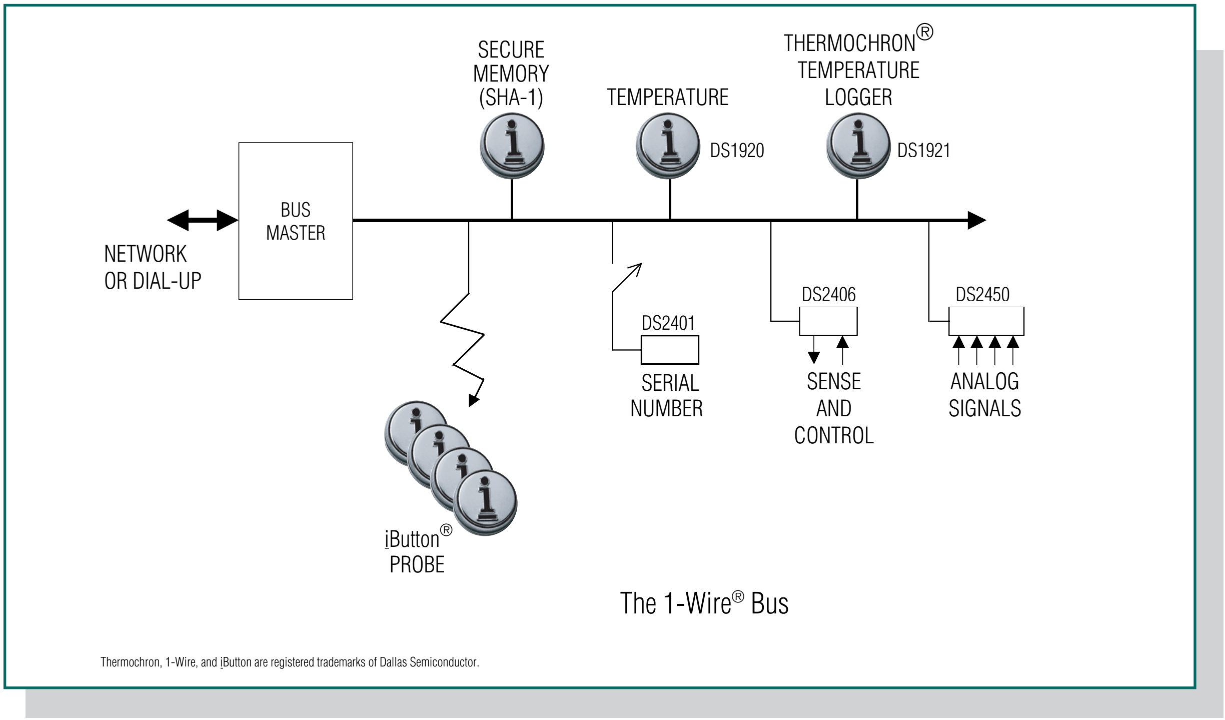
An integral part of the unique ID number in each slave is an 8-bit family code. This code is specific to the device type. Because each device type performs different functions, this code is used to select the protocol that will be used to control or interrogate it. Because each device type performs a different function and serves a different purpose, each has a unique protocol once it has been selected.

Because slave devices may have timed processes or monitor real-world (asynchronous) data sources, they sometimes need to be able to gain the attention of the master quickly. A poll of several dozen connected slaves, addressing each by unique ID number and then reading its internal registers, is somewhat slow and can be CPU-intensive. A special type of device search, called a conditional search, is also supported for this purpose. Slave devices will appear in this special search only if they are in a condition, or have had an event, that matches preset criteria. The master performs this conditional search at regular intervals, and any device that is found is a device in need of service.

Most 1-Wire devices are available in durable, stainless steel containers about the size of four stacked dimes called iButtons. Some iButtons contain tiny lithium cells that power internal real-time clocks or data loggers and maintain NV SRAM data or configuration information for well over 10 years. Some have EEPROM technology that requires no backup power. These stainless steel iButtons also have their unique serial number laser-etched into the lid so humans can identify them. Many 1-Wire devices are also available in standard SOIC, TSOC, or TO packages for PC board mounting. Chip-scale packaging (flip chip) forms are also available for some devices.

1-Wire peripherals include various serial- and parallel-port adaptors for PCs, probes, fob and holders, and a wide array of iButton attachments.

Extensive 1-Wire software examples and APIs are available that implement standards for file communications, structure, and error control at www.ibutton.com.



Thermochron, 1-Wire, and iButton are registered trademarks of Dallas Semiconductor.

Figure 1. Various 1-Wire devices store data (NV RAM, EPROM, and EEPROM), read and log temperature and voltage, adjust resistance, count, control and sense, interface to other systems, and perform timekeeping and cryptographic functions.