## Smart Autonomous 32-bit Microcontroller Peripherals
## Push the Boundaries of Ultra-Low-Power Embedded System Design

### Introduction

Embedded system developers have a constant need to reduce the power consumption in their systems. This low-power design trend has accelerated over the past decade as applications and customers require even lower energy footprints. The benefits of the movement to save power include longer battery life, increased functionality for a given power consumption, and new applications such as implantable medical electronic devices and energy scavenging wireless nodes.

As a result, microcontroller (MCU) suppliers have taken multiple paths in an effort to save every last microampere of current. Table 1 shows many of the techniques developed over the past two decades to reduce the power consumption of an MCU.

**Table 1. Common MCU Power Management Techniques**

| Technique | Description | Result |
|---|---|---|
| Technology scaling | Use finer process node | Reduced active-mode CV$f$ current<br>Higher leakage current |
| Clock-gating and suspension | Apply clocking signals only to the circuits needed at any one time | No power wasted clocking unused circuits |
| Static voltage scaling | Reduce supply voltage based on measuring transistor properties | Reduced CV$f$ and leakage current |
| Dynamic voltage and frequency scaling | Dynamically modify MCU core supply voltage and frequency based on processing needs | Reduced CV$f$ and leakage current |
| RAM and digital state retention | Retain the state of RAM and the MCU core | Very fast transition between power modes |
| Autonomous peripheral operation with direct memory access (DMA) | Place the MCU core in a suspended or sleep state while peripherals are performing functions | Allows a peripheral to perform a single task while the MCU core is suspended |
| Interrupt-based processing | Rely on interrupts to drive MCU instructions | Eliminate power wasted from unnecessary polling |
| Maximize code efficiency | Reduction in required memory<br>Fewer branches in code<br>Efficient code caching | Reduction in flash read power<br>Minimize time spent executing instructions |

Using the techniques in Table 1, MCU designers have driven down on-chip power consumption to levels that support 10 to 20 years of life on a single battery or even battery-free operation by scavenging energy from the environment. Despite these advances, it is now up to microcontroller suppliers to provide embedded system designers even more flexibility in managing power.

One method of implementing additional power savings is to free the CPU core from simple tasks, such as peripheral-to-peripheral information transfer. The resulting reduction in current consumption comes from implementing such tasks through optimized hardware and eliminating the need to execute MCU instructions from flash memory.

Another method is to architect the MCU to comprehend system-level power constraints. This is accomplished by making sure that the MCU has high-efficiency peripheral interfaces with common components, such as LCD displays, wireless transceivers and sensor transducers.

### Data Transfer Manager

MCUs with autonomous peripherals have become more common in the embedded market, but there are limits to this autonomy. Frequently, a peripheral can be configured to perform a single task, such as a comparator monitoring an external pin that wakes up the rest of the system. More advanced peripherals, such as analog-to-digital converters (ADCs) have numerous automation settings, including scanning multiple channels, flexible sample rates and even basic signal processing techniques. However, the block is limited to performing a single set of samples and then waking the MCU core for further processing. In all of these cases, there is a single peripheral performing a simple task while all or most of the MCU core is held in a low-power state. At best, the peripheral can store or fetch data directly into memory through a direct memory access (DMA) channel.

Taking the idea of autonomous peripherals a step further, what if we could chain peripheral tasks together to perform more complex functions, all without the intervention of the MCU core?

In wireless systems, for example, raw data is processed through multiple operations before being ultimately delivered to the radio for transmission. The microcontroller must encrypt the raw data, add error correction, encode the packet, and pass the packet to a serial interface in one or more bursts depending on the FIFO of the transceiver. This process is shown in Figure 1.

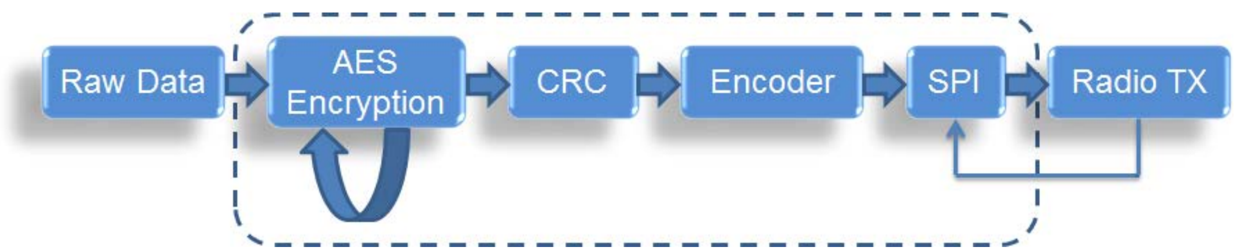



**Figure 1. Example Radio Packet Construction**

Traditional MCU solutions require frequency CPU intervention to configure the transfer of data between each processing step. This means that power is wasted since the CPU is merely repeatedly configuring peripherals and DMA channels based on interrupts.

In contrast, using a dedicated and programmable data transfer manager (DTM), the embedded designer can successfully chain together a complex set of tasks that execute autonomously without relying on the MCU core. In these instances, the core is kept in its lowest power state until all the tasks have completed.

The Si3ML1xx series of ultra-low-power 32-bit microcontroller devices from Silicon Labs has a dedicated DTM hardware peripheral as described above. The DTM module provides two primary functions:

1. Request-driven peripheral-to-peripheral DMA
2. DMA sequencing to handle packet transport, including loop state

The DTM module accomplishes these tasks by collecting DMA request signals from various peripherals and generating a series of master DMA requests based on a state-driven configuration. This master request drives a set of DMA channels to perform functions such as assembling and transferring communication packets to external radio peripherals. This capability saves power by allowing the microcontroller to remain in low-power modes during complex transfer operations.

## Smart Interfaces

Sensor Interface (Advanced Capture Counter)
Another situation where power is consumed outside of an MCU is when a sensor transducer must interface to the MCU. Typically, the microcontroller core provides a signal to stimulate the transducer(s) and then processes the result, either through detection of pulses, quantization or some other means. Once the results are available, they are placed into RAM through a DMA channel.

The Si3ML16x 32-bit microcontroller provides a unique sensor interface manager called the Advanced Capture Counter (ACC) to offload the MCU core of this task, as show in Figure 2.

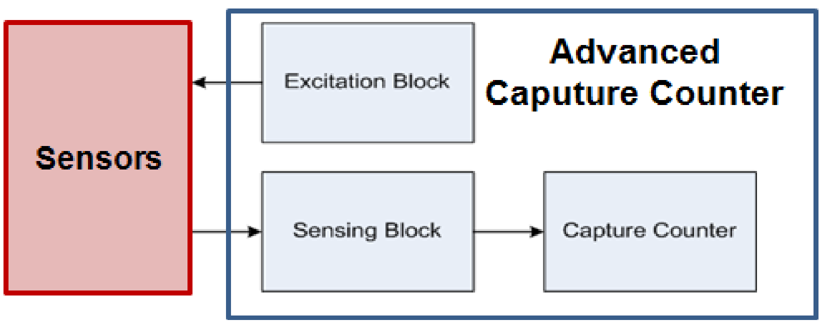
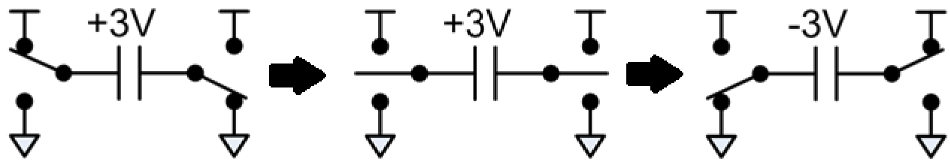



**Figure 2. Advanced Capture Counter**

The ACC module can be configured to periodically excite a sensor for a defined duration and then process the results based on sample rate, threshold detection and counting modes. All of this is completed without MCU core intervention, allowing the system to remain in a low-power state. The ACC peripheral can service capacitive, inductive, Wheatstone bridge/resistive divider, reed switch/pulse and switch closure sensors. Typical power consumption of the ACC system is 500 nA at a 2 ms scan rate.

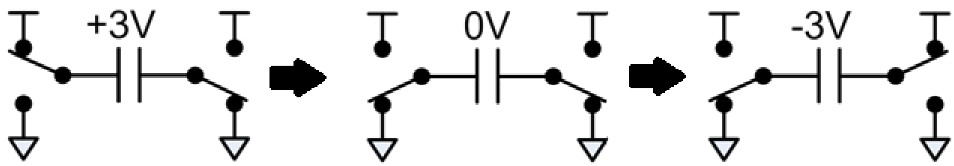
Smart LCD Switching
Segmented LCDs are a common component in low-power systems. The display itself can be abstracted to a matrix of capacitors that are being switched between different voltages using AC excitation. Traditional LCD switching techniques employ a ”break-before-make” scheme to prevent shoot-through current, as shown in Figure 3.




$$Q_{BAT} = 6V \cdot C$$

**Figure 3. Traditional "Break-before-Make" LCD Switching**

For a 3 V supply, the total charge sourced from the battery in the final step is 6 V x C, where C is the value of the capacitance of the segment. An improved method that reduces the charge drawn from the supply is shown in Figure 4.




$$Q_{BAT} = 3V \cdot C$$

**Figure 4. Improved Switching Method to Reduce LCD Power Consumption**

The improved switching technique reduces the charge drawn from the supply voltage by 50 percent. In practice, parasitic losses limit reduce the power savings to 30 to 40 percent, a significant amount that comes without any loss in contrast or performance. The Si3ML1xx 32-bit microcontroller employs the switching techniques described above, providing superior battery life for applications requiring segmented LCDs.

Integrated DC-DC Converter

An MCU is rarely the only component in an embedded system that consumes significant power. Many ICs can be powered from a 1.8 V supply, and it is wasteful to power these components from anything other than their minimum required voltage. For example, many wireless systems are powered from a battery with voltages as high as 3.8 V. If the 2 V drop between the battery voltage and the system voltage is implemented by a linear low drop-out (LDO) regulator, more than 60 percent of the battery's energy and potential life is wasted as heat. A dc-dc converter, on the other hand, can easily convert 3.8 V to 1.8 V with efficiencies greater than 85 percent, resulting in much longer system life. The SiM3L1xx ultra-low-power 32-bit microcontroller family includes an integrated dc-dc converter capable of powering off-chip components through VDC and VDRV, as shown in Figure 5.

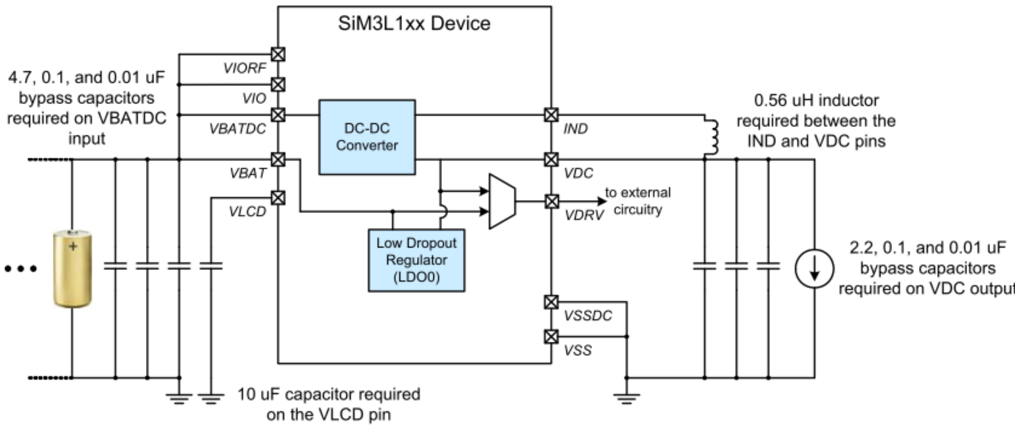



**Figure 5. Example dc-dc Converter Driving External Circuitry**

By powering external components and ICs with a high-efficiency integrated dc-dc converter, battery life is extended while minimizing component costs.

## Summary

The market demand for low-power embedded systems will continue to accelerate, requiring MCU and embedded system designers to advance beyond the traditional power-saving techniques that are standard in low-power MCUs today. This innovation will center around flexible MCU architectures that enable optimized and autonomous operation of smart peripherals and interfaces. Only by offloading the MCU core of mundane tasks and enabling the optimized performance of common off-chip components will battery life continue to be extended. Ultra-low-power 32-bit microcontroller products such as the Si3ML1xx family provide the mixed-signal architecture, integrated smart peripherals and development tools needed to empower embedded system designers to continue to drive advances in energy efficiency.

# # #