How to Maximize the Amount of Time in Sleep Mode

Maximizing the duration of time in sleep mode is extremely important in low-power applications. In these applications, the microcontroller (MCU) spends the majority of the time in the low power or sleep mode. In addition to pin wake sources, MCUs with a wide range of peripherals that continue to operate while in sleep mode and can wake the device allows developers to fully optimize applications for low power. Because the microcontroller does not have to wake while the peripheral operates, the MCU can stay in sleep mode longer and save more power. Consider an application where an MCU needs to wake when receiving serial data through a Universal Asynchronous Receive/Transmit (UART) interface. A standard UART interface would require the device to wake before receiving data, receive the data on the bus, and then process the data. Low-power UART interfaces can stay in sleep mode while the data is being received and wake the MCU after data is available to be read from a buffer. This implementation allows the MCU to stay in sleep mode as long as possible.

Other examples of low-power peripherals that are extremely useful to low-power applications include:
- Sensor interface that can wake the MCU after a configurable number of counts
- Comparator that can wake the MCU on a rising or falling edge
- Real time clock (RTC) with a configurable wake alarm
- Low power timer with a configurable threshold

When selecting a low-power MCU, developers should select a device with ultra-low sleep mode supply currents and a wide range of flexible wake sources. Flexible wake sources allow MCUs to meet an array of applications’ specifications and allow the MCU to maximize the time in sleep mode. With a sleep mode current of less than 250 nA with the RTC enabled at 3.6 V, 9 peripheral wake sources and up to 14 pin wake sources, the new SiM3L1xx ARM® Cortex™-M3 32-bit microcontroller from Silicon Labs meets both criteria, and is an optimal choice for low-power applications.