Low-Power is Life Saver for Medical Electronics Devices

Steve Kennelly, Microchip Technology Inc.

Low-Power is Life Saver for Medical Electronics Devices

by Steve Kennelly, Microchip Technology Inc.

Designers of portable medical devices face unique challenges. Their chosen field is known for regulatory scrutiny, protracted design and life cycles, and a need for unparalleled robustness in the finished product. In addition, design objectives that are common to all electronics can have special significance when it comes to medical devices. For example, low power consumption is always an objective for designers of portable electronics. Less power means a smaller and lighter battery, which enhances portability. In a medical device, more portability can make a huge difference in a patient’s quality of life. It’s even possible that a patient’s life may directly depend on battery life. In this article, we will show how designers can exploit a microcontroller (MCU) to reduce the power requirements of medical devices.

Voltage and Battery Life

Static power consumption is an important figure-of-merit for an MCU in low-power applications. Some MCUs featuring advanced processing technology draw less than 50 nano-amps in sleep mode. To be useful in a variety of low-powered designs, it is important that an MCU operates with a wide range of power supplies. For instance, if Alkaline batteries are used, it is common to specify 1.8V operation because the end voltage on each cell is 0.9V and two cells are typically used in the application. Selecting an MCU that operates over a wide voltage range can extend the operating life of a portable device. However, the MCU's operating voltage range isn't the only deciding factor. The entire system's operating voltage range must be considered, including the peripherals on the MCU. If a single peripheral in a system consumes most of the power, reducing the MCU’s power will have little impact on the total system power consumption.

Peripheral Power Switching

A cardinal principle in the power management of portable embedded systems is to enable the MCU to control the power used by both internal and external peripherals. When you are designing your portable medical device, determine the required physical modes or states and partition the design to shutdown unwanted circuitry. Selecting the right MCU from among the many different vendors can help you eliminate external components and reduce costs. As previously mentioned, an MCU that operates over a wide range of voltages can

Mode	Time in Mode (mS)	Current (mA) By Part	Current (mA) Total	Charge Current * Time (Amp * Sec)
Sleeping	1989		0.001	1.989 e-6
CPU	Sleep	0.001		
Sensor	off	0.000		
EEPROM	off	0.000		
Sensor Warm-up	1		0.166	0.166 e-6
CPU	Sleep	0.001		
Sensor	on	0.165		
EEPROM	off	0.000		
Sensing	1		0.213	0.213 e-6
CPU	run	0.048		
Sensor	on	0.165		
EEPROM	off	0.000		
Scaling	1		0.048	0.048 e-6
CPU	run	0.048		
Sensor	off	0.000		
EEPROM	off	0.000		
Storing	8		2.048	16.384 e-6
CPU	run	0.048		
Sensor	off	0.000		
EEPROM	on	2.000		
Total Time (mS)	2000	Total Charge (Amp*Sec)		18.800 e-6

add versatility to your system design.

As an example of minimizing total system power consumption, consider an MCU-based, data-recording medical monitor device, which comprises a sensor, an EEPROM and a battery (see Figure 1). In practice, the sensor could be measuring temperature, oxygen saturation, blood pressure, glucose concentration, or any number of other quantities. This medical device is intended to monitor a patient over a period of several hours or more. In this example, the MCU takes a sensor reading every two seconds, scales the sensor data, stores the data in external EEPROM memory and awaits the next sensor reading. If power consumption were not a consideration, the EEPROM, the sensor and its bias circuit could all be powered up all the time. However, since this is a portable medical device, efficient use of the available power source is important. So, what could be done to save power in a system like this? The answer lies in having the MCU shut down these peripherals under program control when they are not required. As shown in Figure 1, designers can use the MCU's I/O pins and a few bytes of code to power the EEPROM and the sensor, when required. Because the I/O pins of the selected MCU can source up to 20 mA, additional components are not needed to switch the power.

MCU Power-Management Modes

A popular method of saving power in embedded applications is by putting the MCU to sleep periodically, at times when the system's demand for the MCU's resources is low. In our example, the system is taking measurements every two seconds. If the time required to actually perform the measurement and store the result is 11mS, then the MCU can sleep for 1989mS between measurements. The longer the MCU can be allowed to sleep, the lower the average power consumed by the application

will be. The system's MCU is awakened either through an interrupt or by the expiration of a watchdog timer. It is important to make sure that the watchdog time-out duration is appropriate for the application. Typically, this works as follows: if the application requires the MCU to process a data sample once every fixed time period, then the watchdog timer should wake up the MCU once in the required time period. Using this feature requires the selection of an MCU that supports the appropriate watchdog period.

Calculating Total Average Power Consumption

Using a technique called power budgeting, we will show how designers can estimate current consumption and battery life in an application. Again, consider Figure 1 and follow through the various modes of the data-recorder application: sleeping, sensor warm-up, sensing, scaling and storing. Analysis of the process loop allows us to determine the time spent in each mode during each cycle. Then, current-consumption numbers for each device are taken from the respective device's data sheet, as provided by the manufacturer. Multiplying the total current required in each mode by the duration of that mode yields the amount of charge consumed in that mode during each loop cycle. From Table 1, it follows that each loop cycle of our data-recorder application takes 2000 ms and requires a total charge of 18.8 e-6 Amp Seconds. Using the Table 1 data, we can derive an average current of 0.009 mA as below:

Average Current (mA) = Total Charge (Amp * Sec) /Total Time (Sec)
= 18.8 e-6/2000 e-3 = 0.009 mA
Peak Current = 2.048 mA

Table 1: Power budgeting calculations for the medical data-recorder application; using typical datasheet values for current consumption for the components used in Figure 1.

Conclusion

This article showed how, by using the latest MCUs, designers can implement power-management techniques and reduce power consumption in the design of electronic medical devices. Minimizing power consumption in medical devices cuts heat generation and enables the use of smaller batteries. This, in turn, increases the operating lifetime of the device, improves patient compliance and reduces physical size.

Steve Kennelly has been with Microchip Technology since 1999 and leads the Medical Products Group, which addresses the specific needs of the medical device industry. He has more than 20 years of experience in program-management, marketing, sales and applications-engineering positions and holds a Bachelor's degree in electrical engineering from Arizona State University.