



The debounce debacle

MECHANICAL SWITCHES PLAY an important role in many microcontroller and microprocessor applications. These types of switches are simple, in-

expensive, and easy to install. But, the perceived simplicity of mechanical switches can be deceiving, and you want to keep your users happy. For instance, an annoying and poor switch implementation is a TV button that advances the channel by two or three channels instead of the intended one-channel advance. A more critical implementation might be a switch sensor that determines the status of a door. An open door may trip an alarm or motivate an attendant to check this door. False alarms can be disruptive and expensive. Some switches reside in time-critical environments, such as in keyboards, in which a quick, accurate determination of the switch position is critical.

It is easy to blame these problems on the switches, instead of on the electronics that sense the switch position. An alternative approach is to examine the dynamic characteristics of your switches and assess their environmental influences. All switches demonstrate a switch-contact bouncing action as the switch opens or closes. The switch contacts actually bounce off each other several times before the contacts settle into their final position. (If the switch position is sensitive to touch, a person who has had too much coffee could cause bouncing by inadvertently touching the switch. Switch manufacturers call this inadvertent touching "playing" with the switch.) You also need to look at environmental interference, such as vibrations or EMI. Once you evaluate these system dynamics, you are ready to design the electronics in your circuit.

You can implement the switch-interface electronics using an ana-

log or digital option. The analog option uses an RC filter to control the bouncing signal. A digital gate with hysteresis takes the output of the RC filter and sends the switch-position signal into the microcontroller or the microprocessor. In this type of system, assume that the bounce time is slower than the RC time constant. You are probably safe if you design with a 500-msec switch-closure time. But don't hesitate to verify this timing with your

switch on the bench. Switches bounce less as they open than when they close. In this configuration, the microcontroller or microprocessor receives the switch position after the switch position changes.

An alternative to the analog option is a purely digital one. A digital option is almost free, because you no longer need your external components. The analog option has the additional cost of materials, installation, and board real estate. The digital option can immediately determine whether the switch has changed. The microcontroller or microprocessor can then digitally filter the switch bounce. Think of this debounce circuit as a lowpass, or decimation, filter. The brute-force-debounce digital option uses a programmed delay time in firmware to replace the analog RC time constant.

As an alternative, you can adjust this programmed delay time by using clever programming techniques. This type of algorithm increases the value of a counter when the switch output signal is high and decreases the value of the

same counter when the switch output is low. By taking multiple samples, the counter reaches its high or low range limits, which identifies the switch state. The programmer determines the number of and timing between samples. The timing of the samples should not correlate to the main frequency of microcontroller or microprocessor clocks.

Some say that paying attention to a switch's opening and closing is a lost art. If this scenario is true, evaluating all of your system's elements is a lost art as well. In the real world, electronic systems usually have a switch somewhere in the circuit. The opening or closing of a switch seldom demonstrates a clean tran-

SOME SAY THAT PAYING ATTENTION TO A SWITCH'S OPENING AND CLOSING IS A LOST ART.

sition. The connection consists of a series of breaks and makes before the final make or break. This bouncing phenomenon can occur several to hundreds of times before the mechanism settles. The best, most cost-effective option is the microcontroller/microprocessor approach. You can change your design as you experiment or on the fly at almost no cost. □

REFERENCES

1. Ganssle, Jack, "My favorite software debouncers," *Embedded Systems Programming*, June 16, 2004.
2. *Pickit 1 Flash Starter Kit, User's Guide*, Microchip Technology Inc, DS40051.
3. Mancini, Ron, "Examining switch-debounce circuits," *EDN*, Feb 21, 2002.

Bonnie Baker is the analog/mixed-signal-applications engineering manager for Microchip Technology's microperipherals division. You can reach her at Bonnie.Baker@microchip.com.