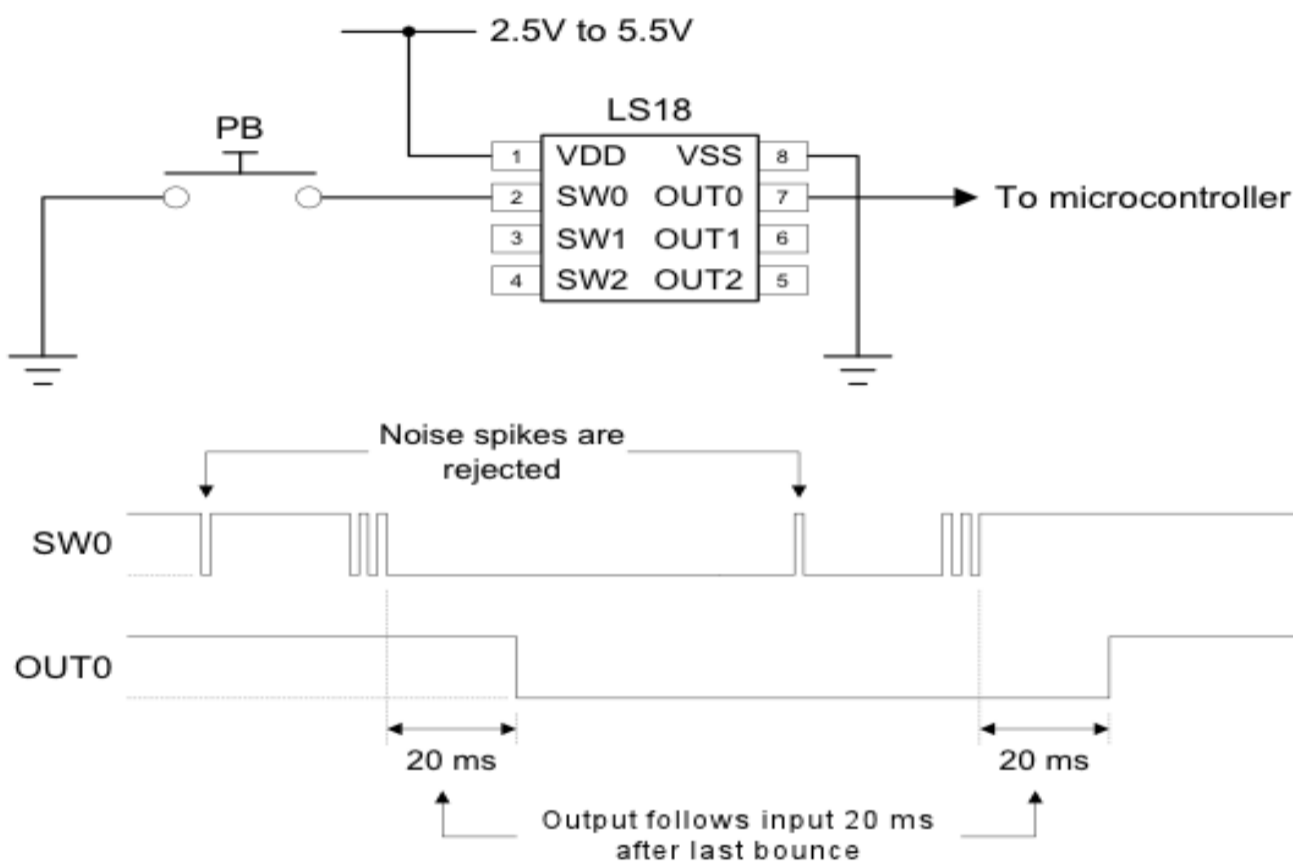


NEW PRODUCTS

Introducing a 21st century solution to the problem of switch bounce

🔍 [BOM](#), [Logiswitch](#), [resistor-capacitor](#), [SMT](#), [switch](#)



When a switch is actuated, the number of bounces can range between 1 to 100+ over the course of a fraction of a second to more than 6 milliseconds.

One of the most common ways of controlling an electronic system is to use mechanical switches. There are many different types of switches, including toggle switches, pushbutton switches, and limit switches, where the latter are operated by the motion of a machine part or the presence of an object. Limit switches, which may also be referred to as microswitches, were originally used to define the limit of travel for a part of a machine, which explains the origin of their name.

It's a sobering thought that—surrounded as we are by the most sophisticated systems conceived by humankind—we continue to predominantly control these systems using a technology that's well over a hundred years old. Pushbutton switches first appeared on the scene circa 1880, while the modern incarnations of toggle switches originally made their presence felt circa 1916. By some strange quirk of fate, 1916 was also the year that educational reformer, social activist, and best-selling American author Dorothy Canfield Fisher warned that "There is a great danger of coming to rely so entirely on the electric button and its slaves that the wheels of initiative will be broken, or at least



Recent Posts

[Miniature feedthru capacitors for compact broadband EMI filtering](#)

[Introducing a 21st century solution to the problem of switch bounce](#)

[How the latest enclosures are assisting the development of today's electronics](#)

become rusty from long disuse.” We can only wonder what Dorothy would have thought to see us now, surrounded by pushbutton switches—and their cousins—of every conceivable shape and size.

Switch bounce

One thing about switches is that they bounce. What this means is that when a switch is actuated, its output doesn’t simply transition from one voltage level to another. Instead, it may bounce back and forth between the two levels before eventually settling in its new state.

All switches (toggle, pushbutton, limit, etc.) bounce, apart from things like mercury tilt switches, which—ironically—most of us don’t use anyway. The number of bounces can range between 1 to 100+ times over the course of anywhere from a fraction of a second to over 6 milliseconds (ms); furthermore, the switch may bounce both when it’s activated and when it’s deactivated.

Although things can change from one actuation to the next, switches often present a sort of “bounce fingerprint.” Some switches will present a lot of narrow bounces close together and then stop bouncing; some may display a smaller number of wider (sometimes widely separated) bounces; and others may offer a combination of effects, such as multiple groups of narrow bounces.

Just to make things more interesting, the *absolute* number of times a switch bounces can vary from activation to activation. Meanwhile, the *average* number of times a switch bounces may be affected by environmental conditions like temperature and humidity; also, it may evolve over time as the switch ages.

Switch bounce wasn’t a problem prior to the introduction of high-speed electronic systems. Indeed, excluding scientists and engineers, the vast majority of people didn’t even appreciate that switches did bounce because any such bouncing occurs far too quickly for human senses to detect. The problem arises when a switch is used to provide a control signal to something like a microcontroller whose system clock is running at tens or hundreds of millions of cycles per second. In this case, unless something is done to mitigate the switch bounce, the microcontroller may perceive a single actuation of the switch as comprising multiple events. This may be mildly annoying in the case of the channel select button on a TV controller, but it can be life-threatening in the case of a piece of safety-critical medical equipment, for example.

Traditional debounce solutions

Switch bounce may be removed in hardware before being presented to the microcontroller, or it can be handled in software after being presented to the microcontroller.

For the purpose of these discussions, we will focus on the simplest—and arguably the most common—type of switch, which is referred to as a single pole, single throw (SPST) device.

Consider a typical hardware debounce solution based on the delay provided by a simple resistor-capacitor (RC) network. Depending on the system’s requirements, this may involve a capacitor, two resistors, a diode, and a buffer chip with a Schmitt trigger input. These components—whose values must be carefully selected to match the characteristics of the switch—occupy space on the printed circuit board (PCB), require additional routing resources, and increase the cost and bill of materials (BOM) for the system. Furthermore, this solution must be replicated for each switch, and things become more complex when multiple switches of different types are used because each switch may have its own unique characteristics.

[ETEL’s New TBS Linear Axis Provides Precision and Stability to Semiconductor Manufacturing Equipment](#)

[Designing through parts shortages](#)

An alternative is to feed the raw bouncing signal from the switch directly to the microcontroller's input and to remove the bounce using software. The idea of addressing switch bounce using software may appear to offer a more attractive solution because it doesn't involve adding any extra components to the board. However, problems may arise if the software developer lacks sufficient switch bounce expertise and relies on incorrect data.

For example, many sources state that a switch will have stopped bouncing after 1ms. These sources include respected authorities like *The Art of Electronics* (Second edition, Page 506) by Horowitz and Hill, which states: "When the switch is closed, the two contacts actually separate and reconnect, typically 10 to 100 times over a period of about 1ms." Countering this, embedded systems guru Jack Ganssle performed his own tests on a collection of switches, reporting an average bounce time of 1.6ms and a maximum bounce time of 6.2ms. This means that any developers who assume that everything will be stable 1ms after detecting the initial transition may be in for an unpleasant surprise.

Even if developers do understand switch bounce and employ sophisticated counter-based or shift register-based debounce solutions, for example, this adds to the size and complexity of the code. This may become burdensome in systems featuring multiple switches, especially when employing resource-limited microcontrollers.

A 21st century solution

LogiSwitch offers a variety of solutions to address switch bounce for SPST (single pole, single throw) and SPDT (single pole, double throw) switches. For the purposes of these discussions, we will focus on the 3-channel LS18 integrated circuit (IC), which can debounce up to three SPST switches. Accommodating a supply voltage of 2.5V to 5.5V, the LS18 is available in both lead through-hole (LTH) and surface-mount technology (SMT) packages (6-channel LS19 and 9-channel LS20 versions are also available).

Observe that no additional pull-up resistors are required on the output from the switch or the input to the microcontroller because these functions are provided internal to the LS18.

Using LogiSwitch's adaptive NoBounce™ technology, the LS18 rejects any noise spikes presented to its inputs. Each output from the LS18 tracks the state of its associated input 20 milliseconds after the final bounce on that input.

Addressing switch bounce using a LogiSwitch IC dramatically simplifies the software running in the microcontroller and removes the burden from software developers who may not understand the switch bounce problem. At the same time, having all of the electronics—including any pull-up-resistors—in a single device that automatically adapts to the characteristics of different switch bounce profiles makes this a solution a "no-brainer" for the hardware design team.

www.logiswitch.com