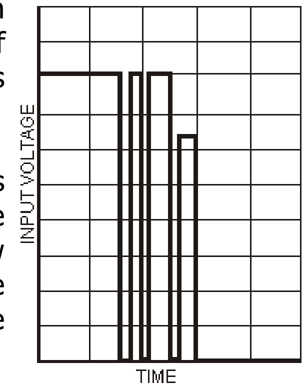
How to Debounce Occupancy Detector Output

When a mechanical device is used as input into a digital system, such as with a pushbutton, the change of state from not being pushed to being pushed is not a simple digital transition from off to on. When viewed over time with great magnification, the state can change back and forth several, if not dozens, of times before it stabilizes in its new state. This effect is known as “bounce”.



If that pushbutton were being used to trigger logic that was implemented with relays, then the bounce would be too fast for the relays to detect, and would not result in noisy conditions. The slow nature of relays acts as a built-in form of debouncing. If the same pushbutton were providing input to an electronic digital system, the bouncing can be very significant.

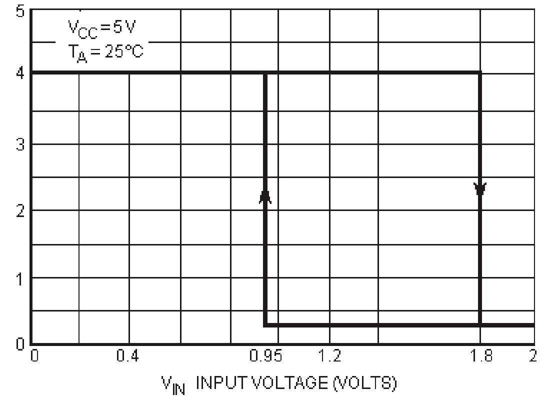
With an occupancy detector, bounce is magnified by track conditions. Slightly dirty sections of track can add to bounce noise. Momentary loss of detection can even be substantial enough to have an effect when driving a relay circuit. The Quad Occupancy Detector from Circuits4Tracks counters the effects of momentary loss of detection with the use of a capacitor on its output, but bouncing will still occur when the state changes from unoccupied to occupied and back again.

Removal of bounce is known as “debouncing”, and there are different ways to implement this. One of the design goals of the Quad Occupancy Detector was to keep costs as low as possible. The original design concept was to have it provide an input into a digital system that would process its inputs through software. Unlike hardware logic, software has the ability to not immediately react when an input changes. This means that debouncing can be performed through the software. One approach is to not “believe” that a state has changed just because an input has changed since it was last read. Instead, a certain number of reads returning the same new value would be needed before it is considered stable. If the new value was only momentary, then the old state would be continued to be seen as stable.

With the Quad Occupancy Detector initially being used with software systems, the added expense of hardware debouncing is unjustified. The open collector output of the detector provides an active low output for driving digital inputs or even relays. When driving digital hardware, bounce will be an issue. Not only is there bounce that can return the output all the way to the other state, but some bounce will result in output voltages at an “unknown” level, as far as digital hardware is concerned. What this means is that 5-volt logic circuits see +5 volts as a digital “1” and 0 volts as a digital “0”. Anything above +2.5 volts, in most every case, will also be seen as a digital 1, and anything below +0.8 volts, in most every case, will be seen as a digital 0. Between +0.8 and +2.5 volts, the state is random. One device might see +1.5 volts as a 1, while another might see it as a 0.

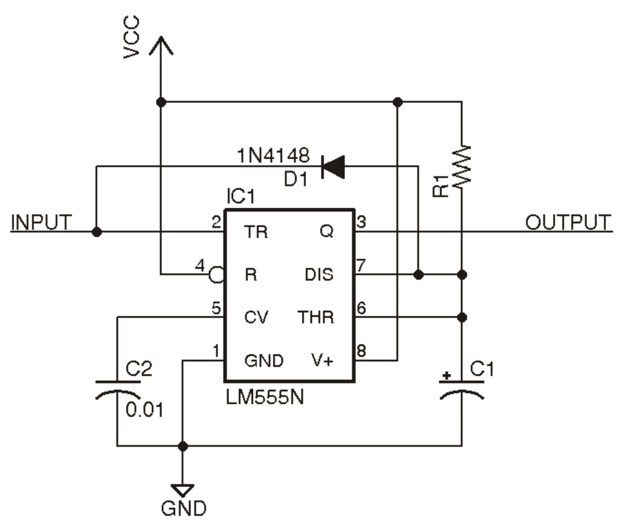
It should be noted that there are some digital inputs that may need input signals to be somewhat less than +0.8 volts in order to be considered a digital 0. The Quad Occupancy Detector’s outputs typically drop to 0.65 to 0.7 volts.

One way to counter debounce in the "unknown" region would be to design the digital hardware to have Schmitt trigger inputs. A Schmitt trigger input has what is called hysteresis, which means that the levels that define the difference between digital 0 and digital 1 are different when moving from a 0 to 1 than when moving from a 1 to 0. Refer to the diagram at the right showing an inverting gate with Schmitt trigger input: when the input is a digital 1, it will not be considered a digital 0 until it drops below +0.95 volts. Once the input is considered a digital 0, it must rise above +1.8 volts before it is considered a digital 1 again.



Another way to perform hardware debouncing would be with a slight delay similar to the software method. This would also have the effect of debouncing momentary loss of detection. Since there is no effect of a momentary false detection when the track is unoccupied, the delay is only necessary when the state becomes occupied. This means that from the moment the state becomes occupied, the output should be held for a minimum time delay, in addition to remaining in that state as long as the input remains occupied. Should the input return to an unoccupied state, the output will be held in its occupied state for the duration of the time delay before returning to unoccupied. If the input reverts back to occupied during the delay, the output continues to indicate occupancy without interruption.

This type of operation initially sounds like a monostable application for a 555 timer. The only problem with that is that when the input remains in the occupied state beyond the timer duration, the output will spike back to unoccupied very briefly and continue at the occupied state for another timer duration. This spike will occur every timer period.



The simple solution to the spike problem is to place a diode between the discharge pin of the 555 timer to the input signal. As long as the input remains low, the output will be active. Once the input is no longer low, the timing starts and the output will not go inactive until the timer period ends. Anytime the input returns low during the timing, the timing is halted and the output remains active. The next time the input goes high, the timing starts once again. The time duration is approximately equal to 1.1 times the product of R1 (in Ohms) and C1 (in Farads). For instance a 47 kΩ resistor and a 4.7 μF capacitor provides a delay of just under 0.25 seconds.

The only issue with this circuit is that it inverts the occupancy logic states. The output will be high when the track is occupied and low when unoccupied. This can be worked into the design of the circuit it drives or configured if it feeds a software system, or an inverting gate can be added to the output to restore an active low output.