

# Switch Debouncing

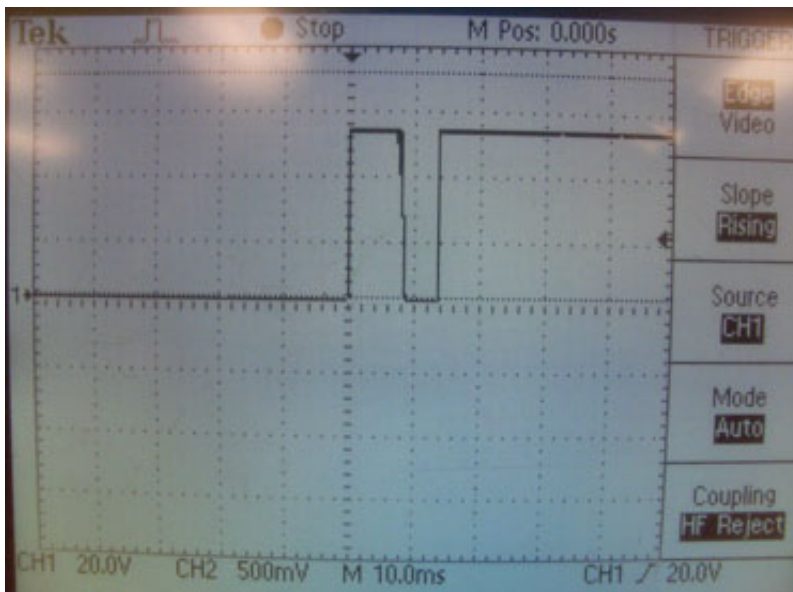
## Contents [hide]

### 1 Introduction

### 2 Example Circuit

## Introduction

When a mechanical switch is flipped/pressed, the metal contacts inside may not open or close cleanly. In the microseconds before the switch achieves a good solid connection, the switch's contacts may "bounce" against each other, turning the switch on and off in rapid succession. The result of closing a pushbutton switch is shown below on an oscilloscope:

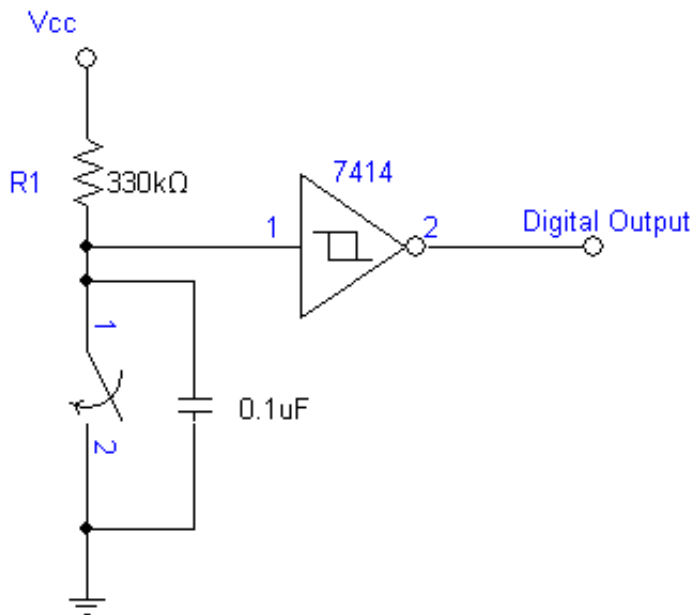


Even though the bounce only lasted milliseconds, a microprocessor is fast enough to tell the difference, and it will think that the switch was pressed twice when it was actually only pressed once.

We can remedy this problem by **debouncing** the switch. There are many ways to do this with both hardware and software, such as using a flip-flop to "latch" the signal, using a capacitor to "absorb" the bounces, or programming the software to ignore the bounces.

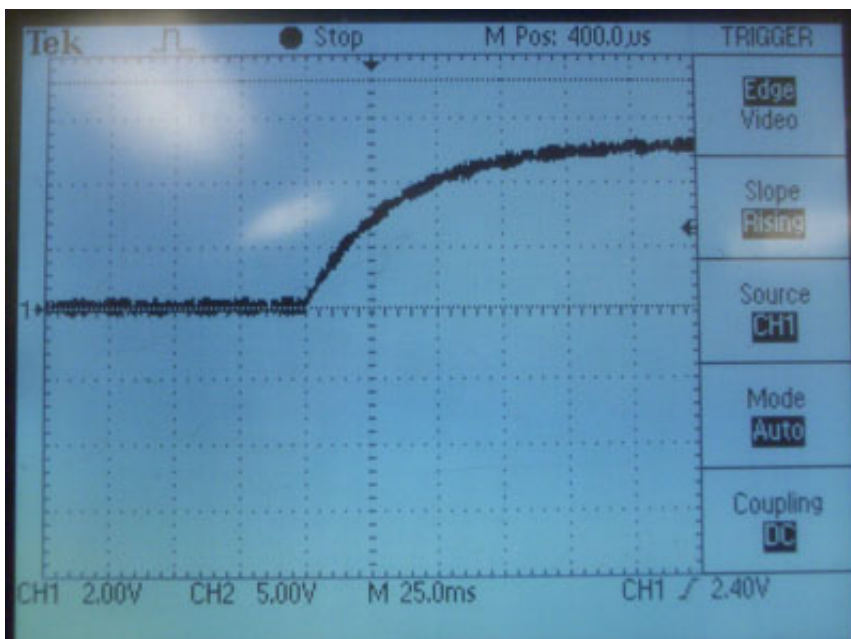
## Example Circuit

Below is a simple circuit for debouncing a switch for digital logic. When the switch is closed, the voltage is pulled down through the switch, and when the switch is open, the output voltage is pulled high through the resistor. The resistor also prevents a short circuit when the switch is closed. The capacitor "absorbs" the bounces when the switch changes states. Keep in mind that this circuit will cause a short delay in switching because the capacitor takes time to charge through the resistor. The delay can be adjusted by varying the values of the capacitor and resistor, but if the resistor is too small, it may not be sufficient to absorb the bounces.

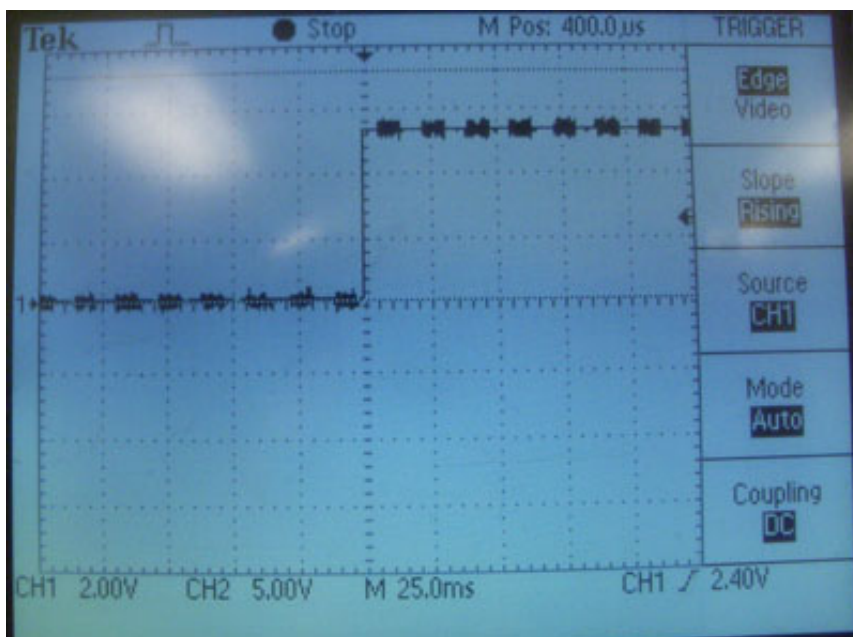


There is one problem: the capacitor exhibits a curve when charging, and parts of this curve will fall in the region where a logic chip is undecided whether to treat it as a high or low. To fix this, a **Schmitt trigger** is added to the output.

Without Schmitt trigger:



With Schmitt trigger:



Remember that the 7414 Schmitt trigger will invert the signal.