De-bouncing of contacts

Mechanical switches may have issue widely referred to as “bouncing”, when fact of single contact closure is recognized as several because of mechanical nature.

Such switches include:

* buttons (push-buttons, tact-buttons, switches)
* encoders (incremental encoders, flow-meters)
* reed switches
* even some digital Hall-sensors has this problem[[1]](#footnote-1)

If constantly checking the voltage on contact output for classic button input schematic with pull-up it may look like this:

|  |  |
| --- | --- |
| Image result for contact debouncing | Image result for contact debouncing |
| Pressing | Release |

Two major strategies to overcome this phenomenon are:

* RC-filter  
  Idea is to smooth the bouncing to a monotonic curve
* De-bouncing  
  Cutting-off the bouncing on both ends as sharp signal

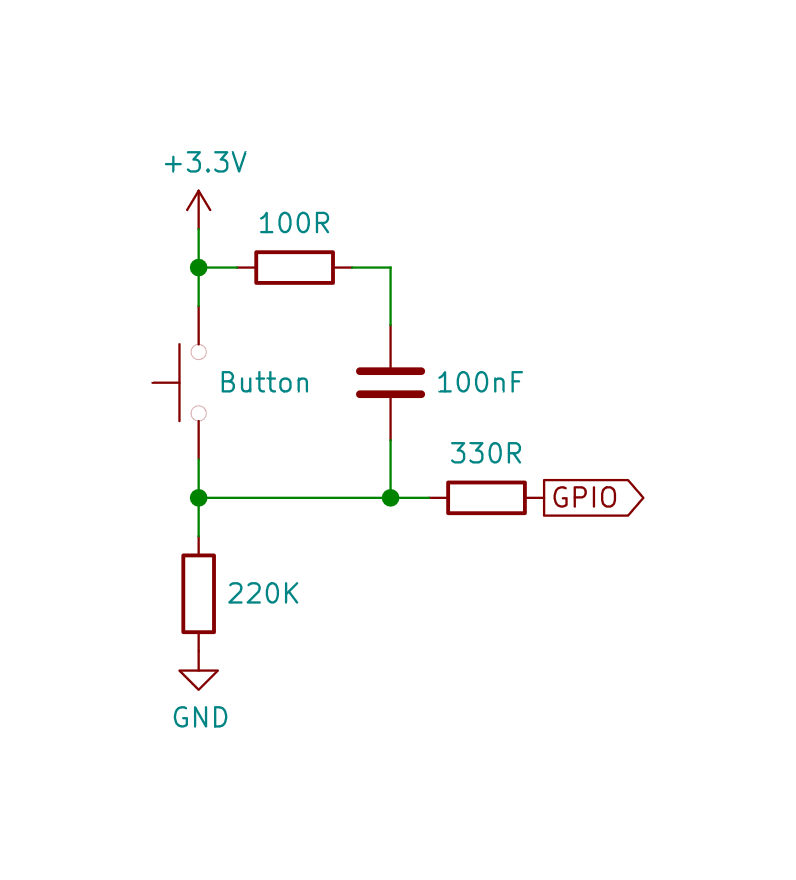
Important note: there are some good encoders which do not require debouncing.

# RC-Filter

In the simplest to smooth those jump couple resistors and a capacitor needs to be added to fix it.

|  |  |  |  |
| --- | --- | --- | --- |
| Pull-up with smoothing | | Pull-Down with smoothing | |
|  | |  | |
| Switch state | Output value | Switch state | Output value |
| Closed | LOW | Closed | HIGH |
| Open | HIGH | Open | LOW |

It works fine with buttons, which are not critical to be processed instantly. Pull-up and pull-down resistors can be optional if connected pin of microcontroller has internal pull-up or down. Another issue is that it is NOT recommended to have button connected directly to capacitor, since when it caused short-tern short-circuit while capacitor is charging or discharging directly from power pin. For example, STMicroelectronics use the following schematics at their STM32F407G-DISC1 development boards:



I case if some instant action required (like limit switch or encoder or any emergency input) you may want to make this process a bit “sharper”, and then it comes to debouncing.

# De-bouncing

## With inverting Schmidt trigger

E.g. : 74xx14, xxx40106B, К561ТЛ2

For that purpose the Schmidt trigger should be added to the output:

|  |  |  |  |
| --- | --- | --- | --- |
| Pull-Down de-bouncing with inverting Schmidt trigger | | Pull-up de-bouncing with inverting Schmidt trigger | |
|  | |  | |
| Closed | LOW | Closed | HIGH |
| Open | HIGH | Open | LOW |
| R1 = 10K; R2 = 10K; C1 = 10nF | |  |  |

Very similar approach was used a while ago by EPSON company for removing noise on reset button for S1C33 CHIP BOARD CIRCUIT. The pull-up schematics is reported in Texas Instruments Application report "Understanding Schmidt Triggers"[[2]](#footnote-2), however is considered to be improved[[3]](#footnote-3).

Presumably, the signal[[4]](#footnote-4) diode is used to equilibrate the charge/discharge times.

For 3.3V I applied this circuit without diode (suggested 1N4148) for both tact-buttons and rotary encoder with following values:

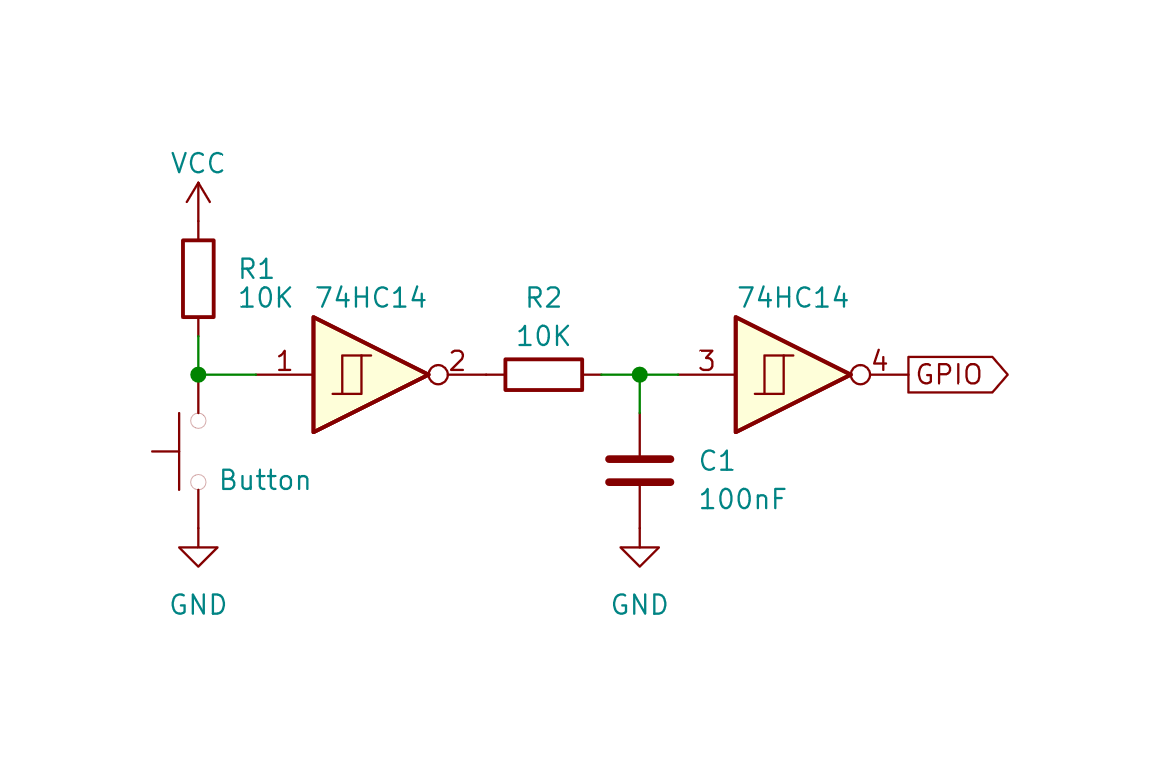
* R1 = 10 or 4.7 or 3.3 KΩ
* R2 = ~10% of R1
* C1 = 100 nF ceramic capacitor

However, it also should work with resistors of equal nominal values[[5]](#footnote-5). The pull-down variation is observed [[6]](#footnote-6) for rotary encoder demo[[7]](#footnote-7).

The 74HC14 IC can be substituted with CD40106, 74HC04, MAX812, КР156ТЛ2 or К561ТЛ2. Theoretically, debouncing is also possible with non-inverting Schmidt triggers alike 74xx7014 (and MAX811), but values will be flipped relative rto what is mentioned above.

## Alternative de-bouncing with inverting Schmidt trigger

This schematics is described in 74HC14 datasheet[[8]](#footnote-8) from Texas Instruments as "delayed buffer" (Chapter 9.2 Typical Application, Figure 5). With adding pull-ups it looks like:



Video demonstration for EC11 encoder was published[[9]](#footnote-9) with following values for components:

* 10 KΩ resistors and 100pF ceramic capacitor
* 9.1 KΩ resistors and 8.2 nF ceramic capacitor

Don't forget the 100nF decoupling capacitor for the 74xx14 IC itself.

## Specialized de-bouncing IC

Also there are several existing specialized solutions:

* MP14490 – hex debouncer
* ELM401 / ELM409 – specialized debouncer for 1 encoder with build-in tact-switch
* MAX6816, MAX6817, MAX6818 – hex debouncer
* LTC6994-1/LTC6994-2 : TimerBlox: Delay Block / Debouncer
* SN74AUP1G14 – single-channel Shmitt-trigger inverter

### MC14490

MC144900 has build-in pull-up resistors, and- doesn't invert the value of input. Requires extra capacitor to define the "de-bouncing time frame":

* 68 pF should be enough for tact-buttons and rotary encoders[[10]](#footnote-10)
* datasheet[[11]](#footnote-11) requires minimum of 100pF
* for 3.3 V (Raspberry Pi) there are suggestions for 0.47 µF [[12]](#footnote-12) or 0.01 µF [[13]](#footnote-13)

The only disadvantage of this IC is big size for DIP package, and high price for SOIC version.

References

<https://pubweb.eng.utah.edu/~cs5780/debouncing.pdf>

<https://www.onsemi.com/pub/Collateral/MC14490-D.PDF>

1. Such phenomena can be found also when using Hall sensor, if:

   signal wires from sensors are close to power lines or they are not screened

   they are too close to source of altering magnetic fields, like in some wheels of electrical bikes: <https://www.youtube.com/watch?v=HW3xqTXIlfo> [↑](#footnote-ref-1)
2. <https://e2e.ti.com/cfs-file/__key/telligent-evolution-components-attachments/00-151-01-00-00-84-01-94/schmitt-trigger.pdf> [↑](#footnote-ref-2)
3. <https://my.eng.utah.edu/~cs5780/debouncing.pdf> [↑](#footnote-ref-3)
4. It is suggested to use 1N4148, but equivalents could also be 1N914(\_,A\_B), 1N916(\_, A\_B), 1N4448(\_, WS, W), 1N4148W, LL4148, LS4148, КД522Б. [↑](#footnote-ref-4)
5. <https://circuitdigest.com/electronic-circuits/what-is-switch-bouncing-and-how-to-prevent-it-using-debounce-circuit> [↑](#footnote-ref-5)
6. <http://codius.ru/articles/255> [↑](#footnote-ref-6)
7. <https://www.youtube.com/watch?v=ojhhQqMy-9U> [↑](#footnote-ref-7)
8. https://www.ti.com/lit/ds/symlink/sn74hc14.pdf [↑](#footnote-ref-8)
9. https://www.youtube.com/watch?v=-Umq9BlGKuo [↑](#footnote-ref-9)
10. https://tsibrov.blogspot.com/2017/11/2-mc14490.html [↑](#footnote-ref-10)
11. https://www.onsemi.com/pdf/datasheet/mc14490-d.pdf [↑](#footnote-ref-11)
12. Gay W. (2015) Debouncing. In: Exploring the Raspberry Pi 2 with C++. Apress, Berkeley, CA.

    https://doi.org/10.1007/978-1-4842-1739-9\_10 [↑](#footnote-ref-12)
13. Gay W. (2017) MC14490 and Software Debouncing. In: Custom Raspberry Pi Interfaces. Apress, Berkeley, CA.

    https://doi.org/10.1007/978-1-4842-2406-9\_5 [↑](#footnote-ref-13)