

Hola a todos, este artículo es para dar una idea(ayuda) de como poder leer un Switch mecánico sin la problemática de los errores o pulsos falsos que existe al momento de leerlos. A este problema se le conoce con el nombre de **"bounce"**.



Fig 1.- Ejemplos de Switchs más usados

Todos los Switch mecánicos poseen un "problema" en cierto momento de ser presionados y de igual forma al soltarlos, estos "momentos" son de corta duración(en milisegundos) se producen cuando las láminas tienen un leve contacto sin llegar a la presión adecuada, esto provoca cierta oscilación o vibración entre ellas lo que ocasiona los pulsos falsos y los conteos erróneos en nuestros sistemas(micros o lógica discreta). Esta vibración u oscilación se detiene cuando la presión entre las placas es la adecuada o cuando la separación es suficiente para no haber "contacto".

En la Figura 2 se muestra un ejemplo de como se verían esos pulsos erróneos, y debido a que nuestros sistemas son muy rápidos en comparación con los milisegundos de duración de cada uno de estos pulsos, es así como nuestro sistema logra detectarlos y activar/desactivar, contar, decrementar, etc., los registros o variables en nuestro programa.

Los ejemplos que veremos son solo algunos de los muchos métodos que existen para "tratar" las señales provenientes de los switches. Realmente se trata de dar una idea general y de ahí ustedes decidirán que método usan y como lo usan. Primero veremos las conexiones y al final algunas ideas para sus programas.

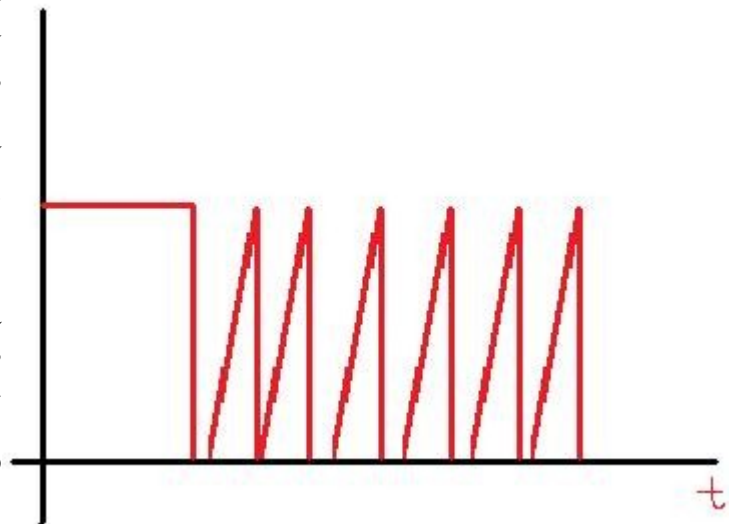


Fig 2.- Diagrama aproximado del efecto "bounce" en los micros

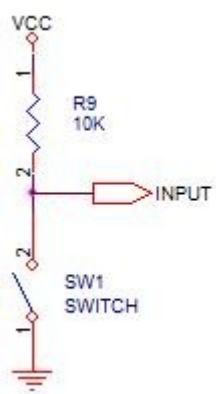


Fig 3.- Conexión que normalmente usaríamos

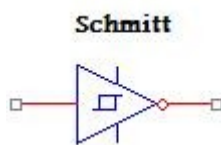


Fig 5.- Compuerta Schmitt inversora

El siguiente método se puede decir es el más común, se trata de una simple Red RC(snubber) a través de la cual opera el switch. En la Figura 4 se muestra el esquema de conexiones de los componentes de esta configuración. Observamos que solo son una resistencia y un capacitor, los valores son solo de referencia y si pueden ser usados en la práctica, dependerá si nuestro microprocesador incorpora pullups internas o no, el capacitor puede comprender valores de 10nF hasta los uF, considerando que mientras más grande sea el capacitor más lento será el regreso al nivel lógico "1". Un dato importante antes de proseguir es el nivel de voltaje en el cual nuestro sistema detecta el nivel lógico "1" y el "0", muchos sistemas basan este nivel en los 0.8V, por debajo de el se detecta el nivel "0" y por encima el "1", algunos micros incorporan entradas con buffer schmitt trigger el cual posee una "ventana" de detección que permite tener un mejor resultado ante cambios abruptos de voltaje en las entradas, un ejemplo de esto se detalla en las figuras 6 y 7 que demuestran los niveles a los cuales el sistema detectaría los cambios entre "1" y "0".

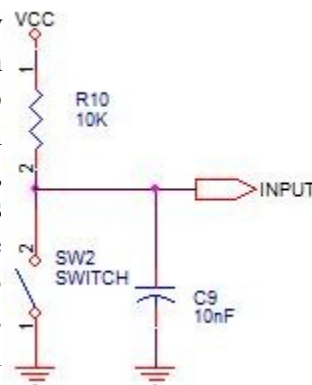


Fig 4.- Configuración RC simple.

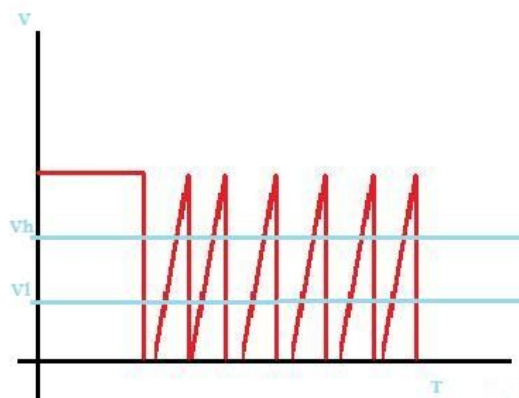


Fig 6.- Entrada sin RC

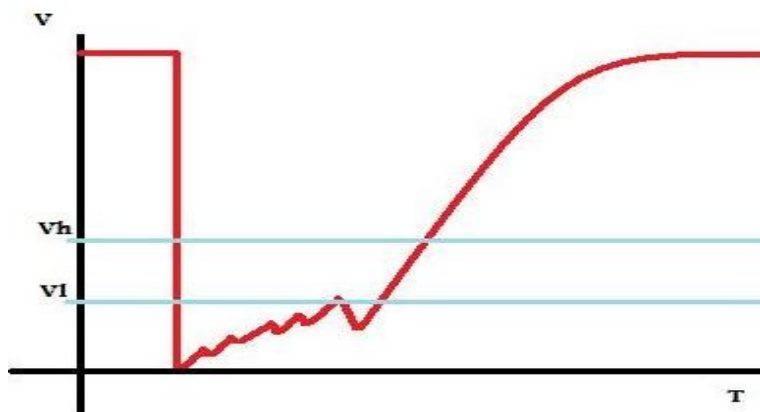


Fig 7.- Entrada con RC

Vl sería el nivel bajo de detección dentro del cual el valor sería "0", Vh es el nivel alto de detección para el "1", en el tramo comprendido entre estos dos niveles se mantiene el nivel anterior, es decir, que si se detecta primero el nivel bajo("0") la salida será igual mientras no se alcance el nivel alto, y viceversa. Para los sistemas donde se fija un solo nivel de determinación, se aconseja estar conscientes de esto para elegir los valores correctos de R y C para nuestro snubber(filtro).

Esta configuración es de las más simples pero tiene ciertas desventajas que hay que considerar. Como el capacitor está en paralelo al Switch, éste se encarga de recibir o drenar toda la corriente de descarga del capacitor, que aunque sea en tiempos muy pequeños a la larga puede terminar dañando el Switch antes de lo normal. Otro detalle es que al descargar casi inmediatamente el capacitor, puede accionar de nuevo un disparo no deseado. Para evitar esto muestro ahora la siguiente configuración, en la figura 8, que adiciona un resistor para anular estos desperfectos. A mi criterio es la mejor opción a usar.

Podemos observar que existe un resistor en serie al Switch, los dos anteriores permanecen igual. El R1 puede ser un PullUp interno de algún microcontrolador o circuito digital, un valor estable y confiable puede ser de 22K

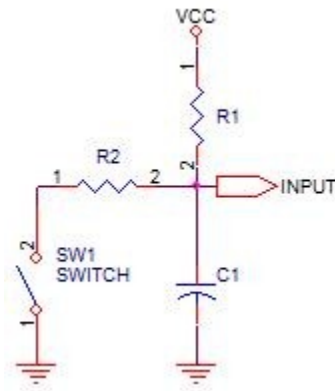


Fig 8.- Entrada con RC+R

Esta resistencia en serie permite una descarga más lenta y mejor controlada, de tal forma que ya no existe esa corriente momentánea dentro del Switch y de igual forma es más difícil producir un disparo erróneo por contacto leve. Ahora manejamos dos tiempos, el de descarga y el de carga. Recordemos que no vamos a hablar de fórmulas, pero lo ideal es conocer los límites de detección en las entradas del microcontrolador o del circuito digital para calcular de manera precisa los valores de cada componente. El rango de valores ideales para este circuito son $R1 = 10K-47K$, $R2 = 2K2-4K7$ y $C = 10nF-1\mu F$, una configuración típica sería de $R1 = 33k$, $R2 = 3K3$ y $C1 = 33nF$. El valor de $R2$ debe ser siempre más pequeño que $R1$ (unas 5 veces por lo menos) ya que de lo contrario el tiempo de descarga se vuelve muy lento, además que al ser una red divisora es posible que el voltaje no llegue a un nivel lo suficientemente bajo como para detectar el "0". La relación de valores se calcula como una simple Red divisora ($Vr2 = (VCC \cdot R1) / (R2 + R1)$) procurando que $Vr2$ sea menor a 0.8V.

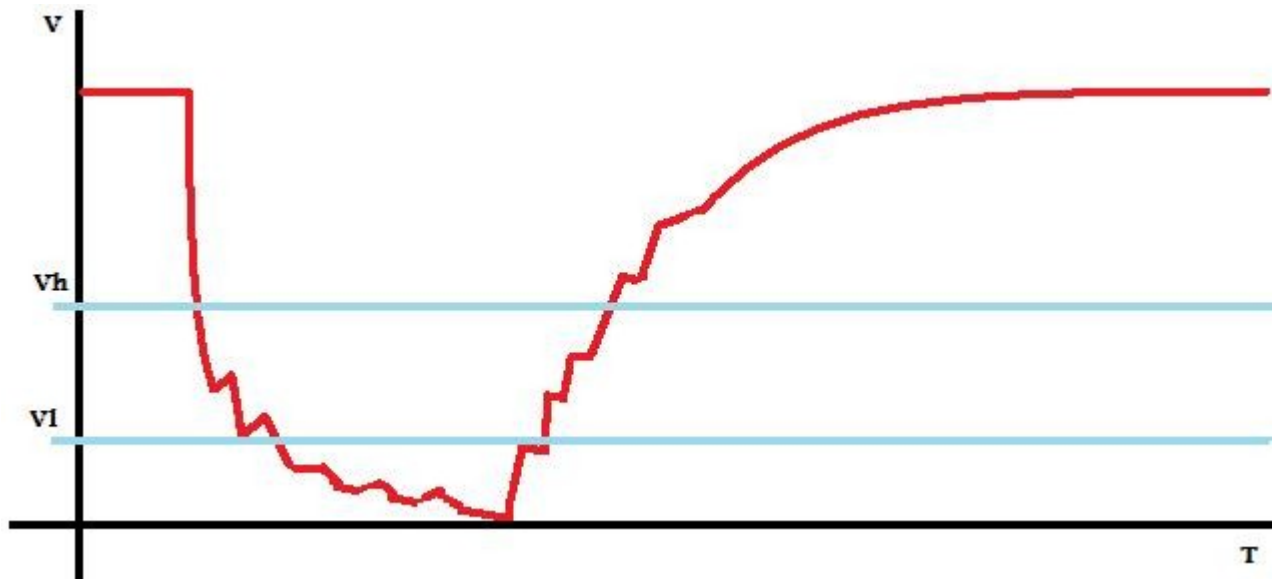


Fig 9.- Gráfica de tiempo del RC+R

En la gráfica de la figura 9 podemos apreciar que el tiempo que pasa desde que presionamos el Switch hasta que el voltaje llega al nivel de Vl es mayor que en el anterior donde prácticamente era cero. La figura 10 muestra en color crema como el sistema vería el pulso producido al presionar y soltar el Switch con la configuración RC+R.

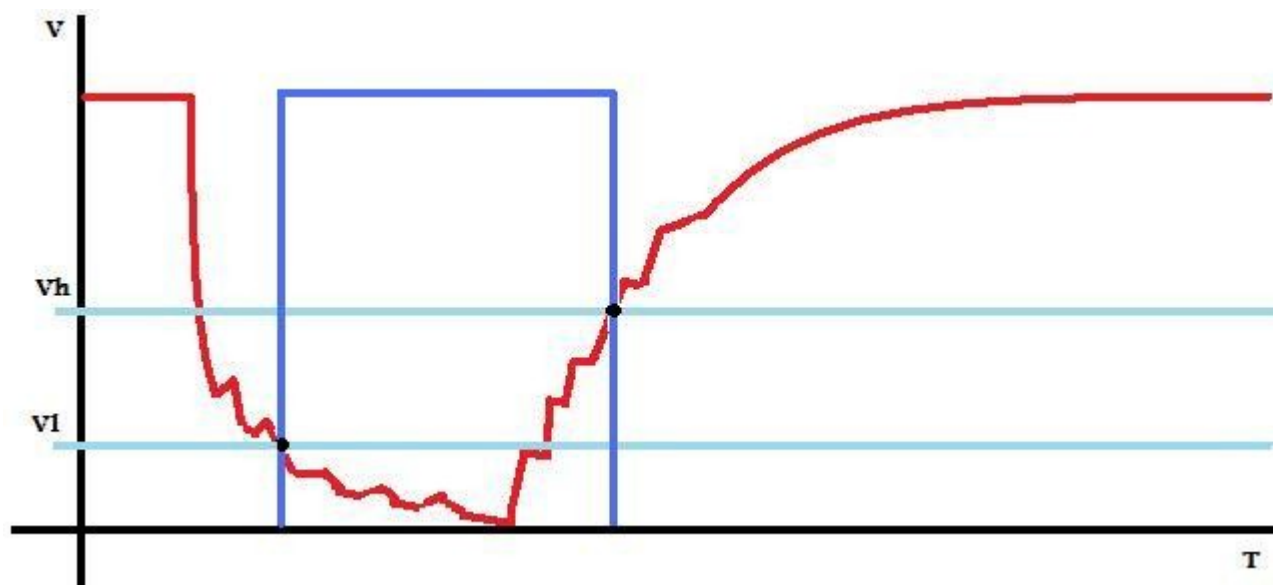


Fig 10.- Posible respuesta en el sistema con la configuración RC+R

Esto es lo básico en cuanto al “bounce” y su respectivo “de-bounce”, pero, y para que nos sirve conocer esto? R: De mucho, no es lo mismo un switch que está dispuesto para ser accionado por una persona, que uno accionado por una máquina. Los tiempos entre estos dos varían mucho. Por darles una idea, cuantas veces pueden presionar un Switch en un segundo?? R: 3,4,5,10 veces?? quien logre hacer 10 conteos en un segundo merece mucho respeto, si tomamos 10 como límite máximo de presiones que puede hacer una persona en un segundo nos da 100ms de tiempo o mejor dicho 10Hz. Este tiempo es importante que lo tengan en cuenta para más abajo en el texto. Ahora bien, imagínense una máquina por ejemplo una bicicleta estática o estacionaria(para hacer ejercicio) que incorpore un Switch que cuente el número de vueltas que da la rueda en un segundo, esto sirve para calcular velocidades, distancias y energía consumida, la pregunta es, cuantas veces se accionaría ese Switch en un segundo?? R: viendo el lado positivo podemos suponer que si la persona lo quiere tomar con “calma” podríamos hablar de unas 15 o 20 veces por segundo, unos 50ms, pero si se trata de todo un atleta fácil puede llegar a los 200 o más, esto supone un tiempo de 5ms(0.005s), esto es crítico a la hora de seleccionar nuestro método de “de-bounce” y los valores de los componentes, para el primer caso es bueno usar el RC+R, pero para la bicicleta en definitiva recomiendo más el RC, porque si usáramos el RC+R, debemos sumar el tiempo de descarga al tiempo total y en dado caso podría no sernos útil.