



Filtro mediana

El filtro mediana nos permite procesar los datos leídos de un sensor de forma que podamos descartar errores de lectura, valores erróneos o cambios bruscos en los datos leídos de un sensor.

Los sensores no son perfectos, y menos algunos de los sensores “*low cost*” que se utilizan con Arduino en el entorno educativo. Por ejemplo un sensor de temperatura/humedad puede de vez en cuando dar valores dispares, y volver rápidamente a la normalidad, si no filtramos y descartamos esos valores fuera de lugar el resultado puede ser incorrecto y volvernos locos a la hora de poner en marcha un proyecto.

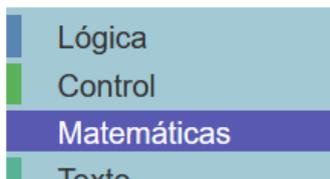
Una manera sencilla de mejorar y filtrar los datos es utilizando un algoritmo que calcula la mediana de los últimos valores leídos, el número de datos usados para el cálculo se llama “*ventana*” y con cada nuevo dato vamos refinando el valor obtenido final. Según el tamaño de ventana (número de muestras en el cálculo) el valor resultado de la mediana pero también el cálculo consumirá más tiempo del microcontrolador y más recursos de memoria.

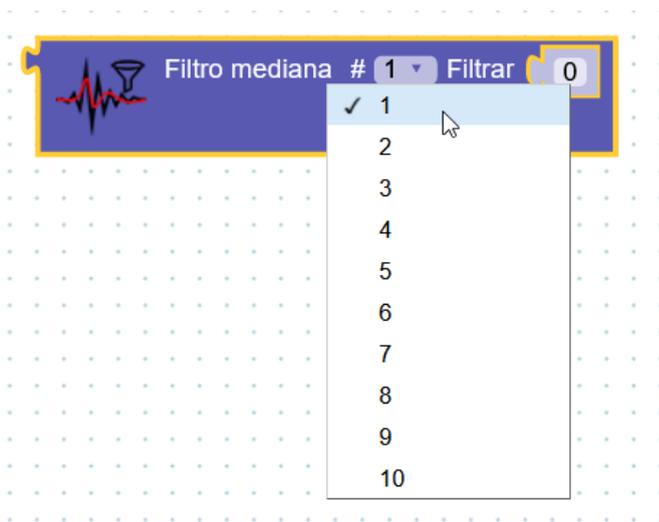
Las características de un filtro por cálculo de mediana dependerá del ancho de ventana y de la frecuencia de muestreo del sensor.

Basado en la librería e información de Luis Llamas:

<https://www.luisllamas.es/arduino-filtro-mediana-rapido/>

En ArduinoBlocks tenemos 10 filtros de mediana disponibles:





El uso del filtro es muy sencillo, simplemente le damos el nuevo valor a filtrar (se añadirá a los datos anteriores) y calculará la mediana internamente con el número de valores indicados (por defecto 5 valores)

Para usar el filtro por mediana tenemos dos bloques:

- 1) **Filtrar** : se debe indicar el nuevo valor leído para añadir al filtro y devuelve el valor calculado.



- 2) **Iniciar el filtro**: es opcional y se debe usar en el bloque inicializar. Permite modificar el ancho de ventana del filtro, es decir, el número de valores que se usarán internamente para el cálculo de la mediana.

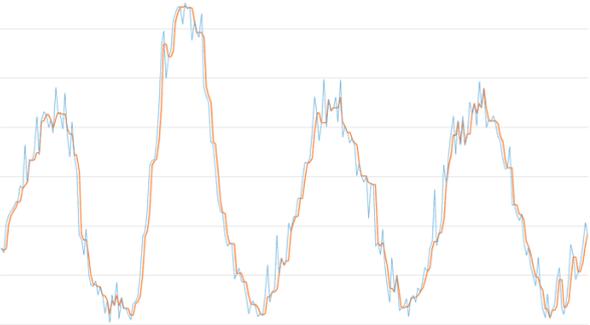
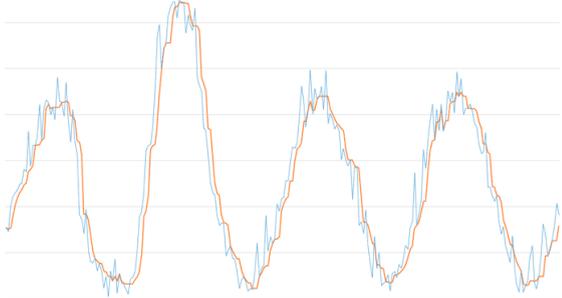
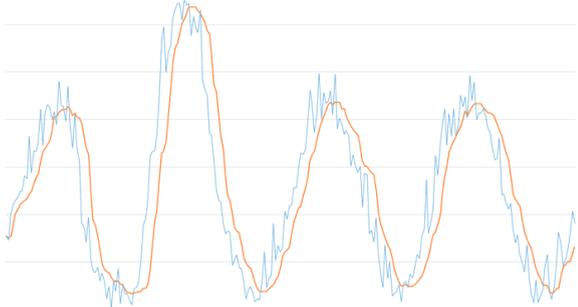


Por defecto, si no usamos este bloque el filtro usará 5 muestras para el cálculo.

Valores típicos recomendados (mejor valores impares para optimizar el cálculo interno):

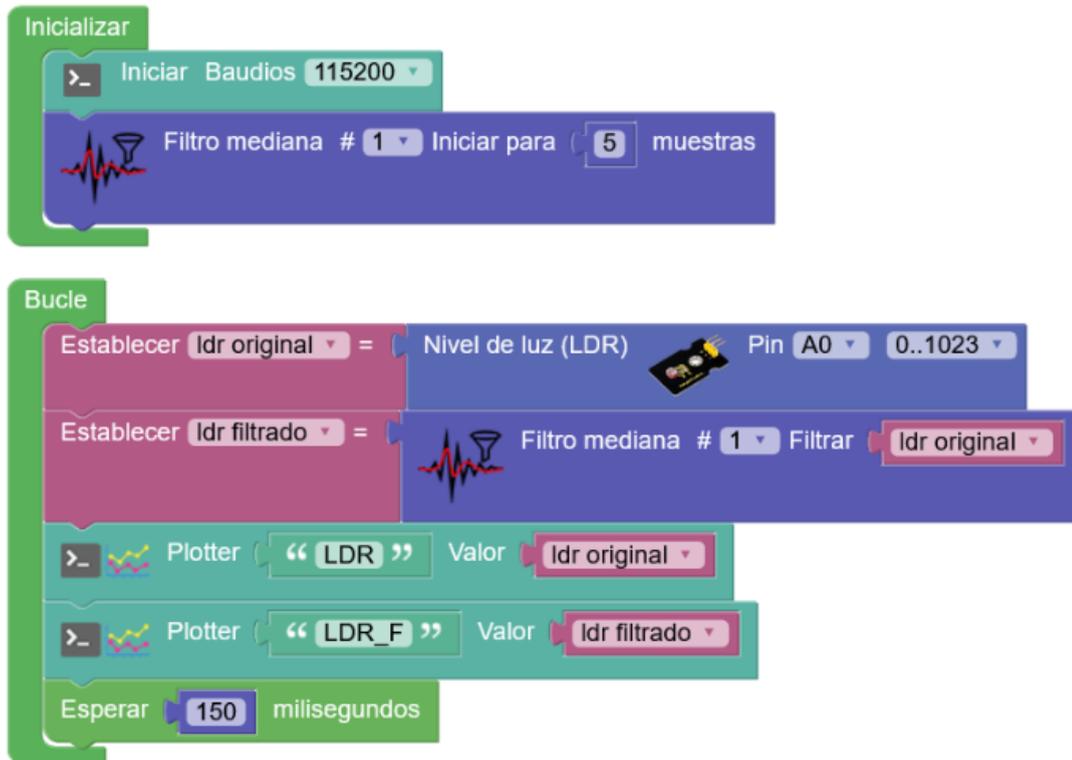
(*) *Imágenes e información obtenida del [blog de Luis Llamas](#)*

3 muestras	Funcionamiento muy rápido, pero con posibilidad de error en caso de varios picos de datos erróneos seguidos.
-------------------	--

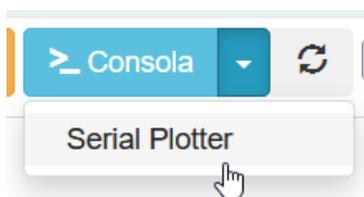
	
5 muestras	<p>Filtro con capacidad de “amortiguar” el ruido de forma bastante eficiente</p> 
11 muestras	<p>Filtro con respuesta muy suavizada, pero con retraso del resultado respecto a la original.</p> 

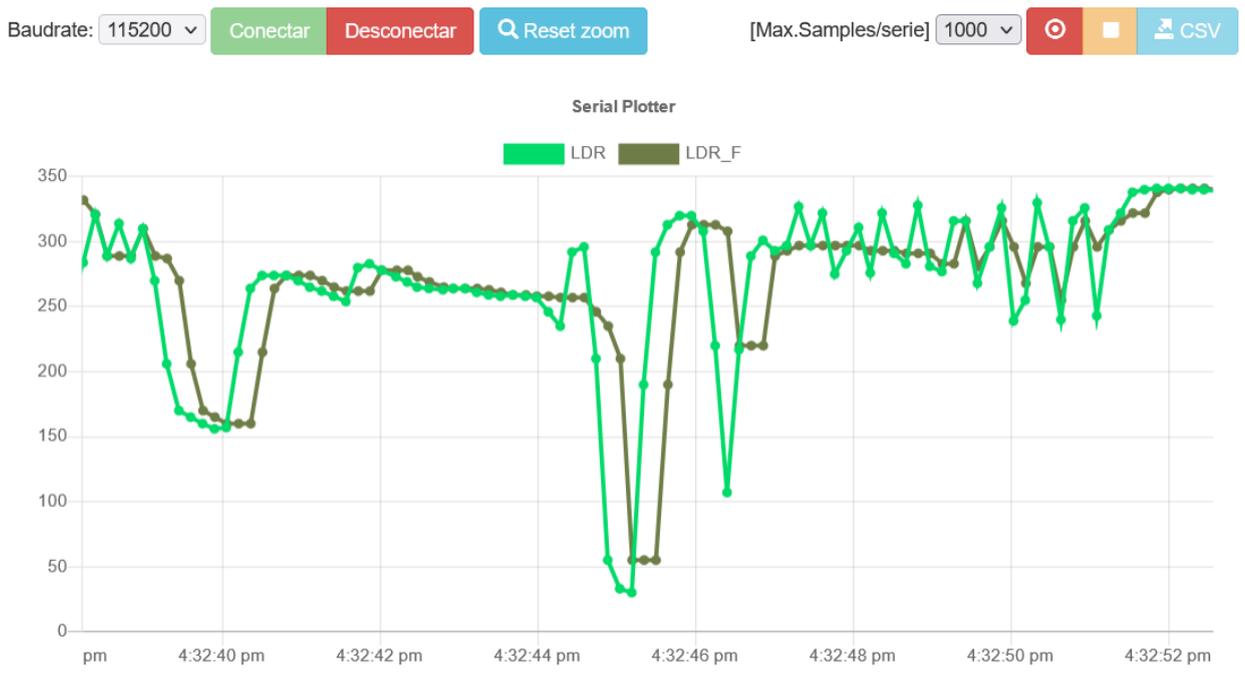
1) Ejemplo del filtro con sensor de luz LDR

Sensor LDR de luz procesado con el filtro mediana para evitar cambios bruscos, y mejorar la medición.



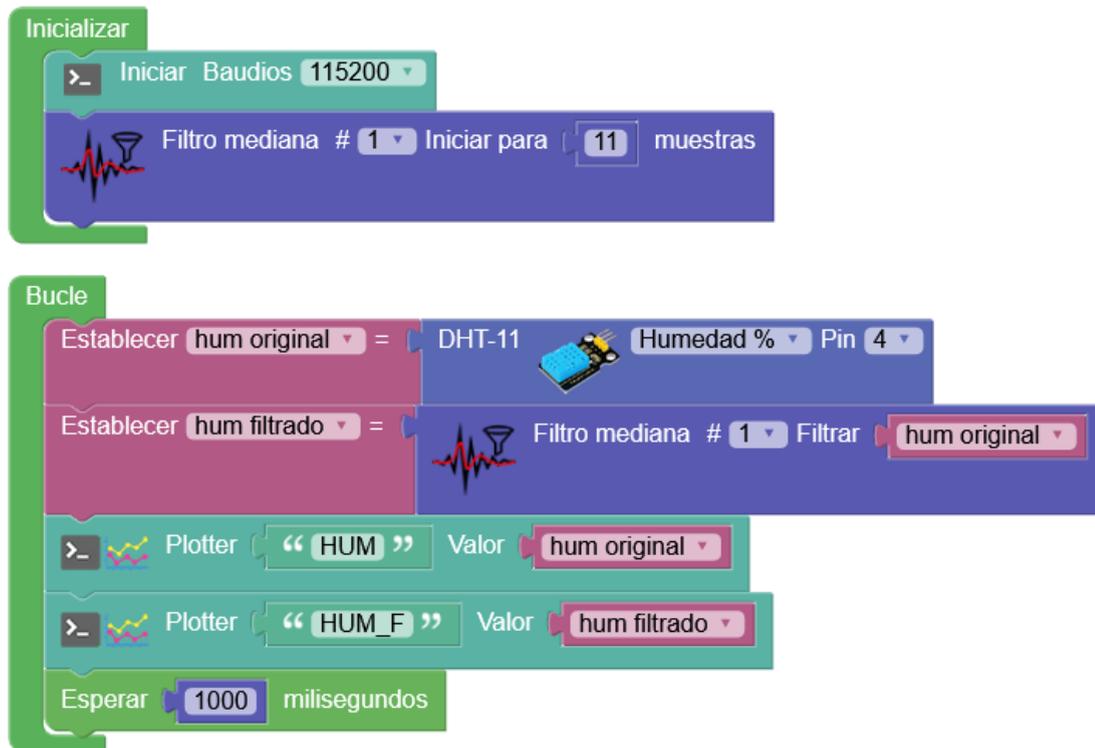
Mediante el serial plotter visualizamos la diferencia entre la señal leída del LDR original y la señal de la LDR procesada con el filtro mediana con un valor ventana de 5 muestras y una frecuencia de muestreo de 1/150 ms aprox.





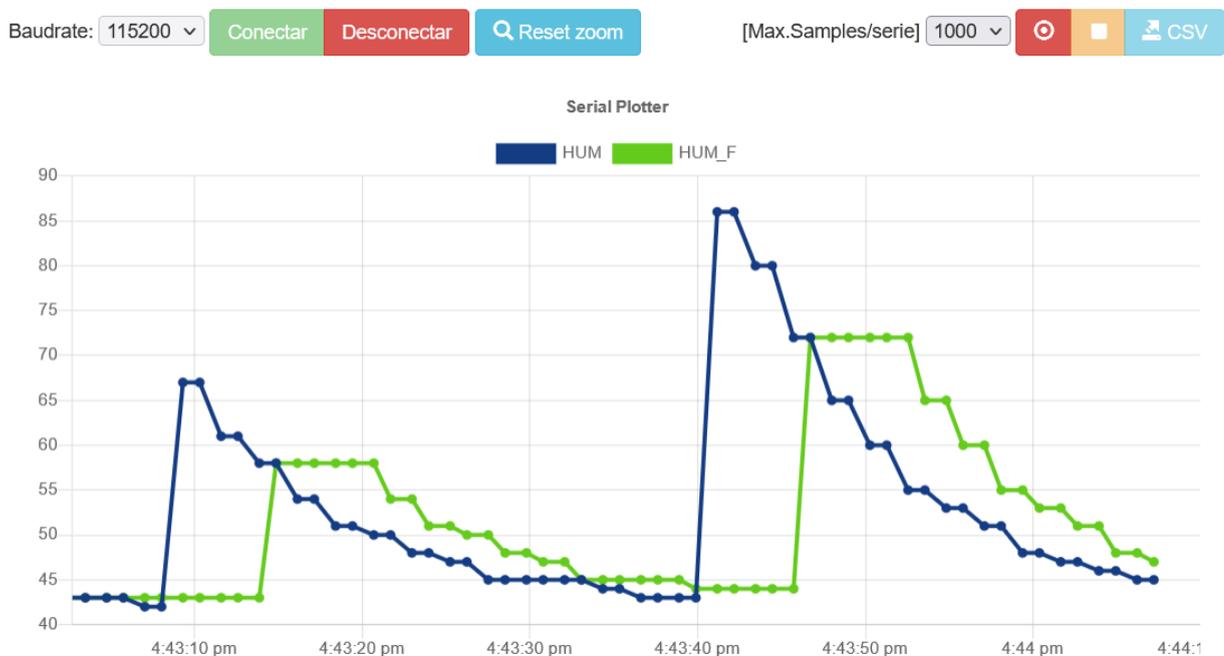
Se puede apreciar claramente con un valor de 5 como cuando la señal sin filtrar (verde intenso) varía rápidamente dando saltos, la señal filtrada (verde oscuro) va adaptándose sin dar saltos tan bruscos y evitando los dientes de sierra donde se produce muchas variaciones rápidas en la señal (normalmente ruido).

2) Ejemplo con sensor de temperatura y humedad



La lectura del sensor de temperatura en este ejemplo se realiza con una frecuencia mucho menor. El resultado visto en el plotter quedaría así:

ArduinoBlocks :: Serial plotter + Datalogger (BETA)



3) Ejemplo de uso general del filtro para varios sensores, y configurando cada filtro con un valor diferente de muestras para el cálculo:

```
Inicializar
  Filtro mediana # 1 Iniciar para 5 muestras
  Filtro mediana # 2 Iniciar para 7 muestras
  Filtro mediana # 3 Iniciar para 11 muestras
```

```
Bucle
  Establecer distancia = Filtro mediana # 1 Filtrar Distancia (cm) [Trigger] 2 [Echo] 3
  Establecer temperatura = Filtro mediana # 2 Filtrar DHT-11 Temperatura °C Pin 4
  Establecer sonido = Filtro mediana # 3 Filtrar Nivel de sonido Pin A0 %
```

Bibliografía:

<https://www.luisllamas.es/arduino-filtro-mediana-rapido/>

<https://www.luisllamas.es/libreria-arduino-median-filter/>

<http://www.arduinoblocks.com>