

**Universidad Tecnológica Nacional
Facultad Regional Paraná**

Etapa N°5

OSCILOSCOPIO DIGITAL

“Medidas Electrónicas II”

Docentes:

**Cappelletti, Carlos
Krenz, Monica
Filipuzzi, Fernando**

Grupo N° 2

Estudiantes:

**Domé, José
Medrano, Mauro
Velazquez, Fabricio**

2021

ÍNDICE

El Osciloscopio	4
Aplicaciones del Osciloscopio	4
Principales especificaciones de los osciloskopios digitales	5
Ancho de Banda	5
Velocidad de Muestreo	6
Profundidad de Memoria	7
Número de Canales	8
Velocidad de Captura de forma de onda	8
Funciones avanzadas	8
Identificación de las partes de un osciloscopio digital	9
Diagrama en bloque	13
Sistema vertical	13
Sistema horizontal	15
Funciones del sistema vertical y Horizontal - Circuito implementado	15
Interrupción del microcontrolador para los sistemas de escala vertical y horizontal	18
Sistema de disparo	19
Fuentes de disparo	22
Modos de disparo	22
Acoplamiento del disparo	22
Retención del disparo - Holdoff	23
Tipos de disparo	24
Posición del disparo	26
Nivel y pendiente de disparo	27
Sistema de adquisición	28
Interrupción como método de disparo	28
Fuentes de Interrupción en el ATMega328P	30
Prescaler	31
Modos de timers	31
Tipos de trimers	31
Prescaler y Registros de Comparación	32
Registros del TIMER	33
Configuración de timers	34
Muestreo de la señal	34
Valores automáticos - Frecuencia y Voltaje	36
Frecuencia	36
Voltaje	37
Sistema de visualización	38
Pantalla OLED	40
Visualización de gráficos en pantalla OLED	40

Pines de conexión módulo pantalla OLED con Arduino	41
Conexión pantalla OLED con Arduino I2C	41
Librerías pantalla OLED con Arduino	42
Trigger de la señal	43
Representación en pantalla	44
Señal representada en la pantalla OLED	45
Función de Hold	46
Implementación del osciloscopio	47
Esquema de conexión	47
Esquemático	47
Diseño 3D del PCB	48
Ensamble - Renderizado	49
Descripción de los componentes	50
Calibración - Actualización	51
Punta de Prueba	52
Ficha Técnica	53
Particularidades	53
Mejoras	53
Presupuesto de materiales	54
Comparativas comerciales	55
Conclusión	56
Bibliografía	56

El Osciloscopio

Un osciloscopio es un instrumento de visualización electrónico para la representación gráfica de señales eléctricas que pueden variar en el tiempo.

Existen dos tipos de controles que son utilizados como reguladores que ajustan la señal de entrada y permiten, consecuentemente, medir en la pantalla y de esta manera se puede ver la forma de la señal medida por el osciloscopio.

Para medir se lo puede comparar con el plano cartesiano. El primer control regula el eje X (horizontal) y representa fracciones de tiempo (segundos, milisegundos, microsegundos, etc., según la resolución del aparato). El segundo regula el eje Y (vertical) controlando la tensión de entrada (en Voltios, milivoltios, microvoltios, etc., dependiendo de la resolución del aparato).

Estas regulaciones determinan el valor de la escala cuadricular que divide la pantalla, permitiendo saber cuánto representa cada cuadrado de ésta manera conocer el valor de la señal a medir, tanto en tensión como el periodo. La imagen así obtenida se denomina oscilograma.

A diferencia de un osciloscopio analógico, un osciloscopio digital utiliza un conversor analógico digital (ADC, por sus siglas en inglés) para convertir el voltaje medido en información digital. Estos osciloscopios adquieren la forma de onda como una serie de muestras, las almacenan posteriormente reconstruye entonces la forma de onda para su presentación en pantalla.

Aplicaciones del Osciloscopio

Ya sea como instrumento de propósito general, como se encuentra descripto, o como instrumento de propósito específico, el osciloscopio encuentra una gran variedad de aplicaciones que van desde la medicina hasta el terreno de la industria, pasando por supuesto por una amplia gama de usos científicos que cubren desde la física hasta la biología.

A continuación se presenta una breve lista de usos típicos del osciloscopio.

- Instrumentación Electrónica: medición de amplitud, frecuencia, fase y distorsión de señales eléctricas. Trazador de curvas.
- Medicina: Electrocardiógrafo; electroencefalógrafo; medición de presión arterial y venosa; medición de ritmo respiratorio; electro- miógrafo (actividad eléctrica del tejido nervioso).
- Radiocomunicaciones: Analizador de espectros; medidores de modulación; medidores de frecuencia; pruebas de líneas de transmisión. (Caracterización de dispositivos).
- Navegación: Sistemas de radar; sistemas de sonar; señalizadores; sistemas de orientación; sistemas de simulación.
- Física: Duración de eventos cortos (pulsos de nanosegundos a milisegundos); caracterización de materiales; monitoreo de eventos nucleares; experimentos de espectroscopia.
- Industria: Sistemas de medición y prueba; monitoreo y pruebas en control de calidad.
- Servicios: Reparación de equipo electrónico; afinación electrónica automotriz.

Principales especificaciones de los osciloscopios digitales

Los usuarios que estén en busca de un osciloscopio deberán conocer ciertos criterios fundamentales para una correcta selección del instrumento, que mejor se adapte a nuestros requisitos de medición.

Los usuarios tendrán que elegir entre una amplia variedad de proveedores y de osciloscopios con innumerables características diferentes, por ejemplo osciloscopios con anchos de banda desde los 200KHz a 1GHz o incluso en alta gama hasta los 100GHz, o también osciloscopios con velocidades de muestreo entre 1Ms/s, 5Gs/s y superiores (40Gs/s, 80Gs/s, 250Gs/s), resoluciones de 8 bits o 12 bits, anchos de banda de entrada mínimos entre 2mV/div y 500 μ V/div, etc.

A continuación se analizan las características y funciones principales de los osciloscopios digitales:

Ancho de Banda

La entrada del osciloscopio consiste en dos componentes: una ruta de entrada analógica y un conversor analógico-digital (ADC). La ruta de entrada analógica atenúa, amplifica, filtra y/o acopla la señal para optimizarla como preparación para la digitalización por el ADC. El ADC muestrea la forma de onda condicionada y convierte la señal de entrada analógica en valores digitales que representan la forma de onda de entrada analógica.

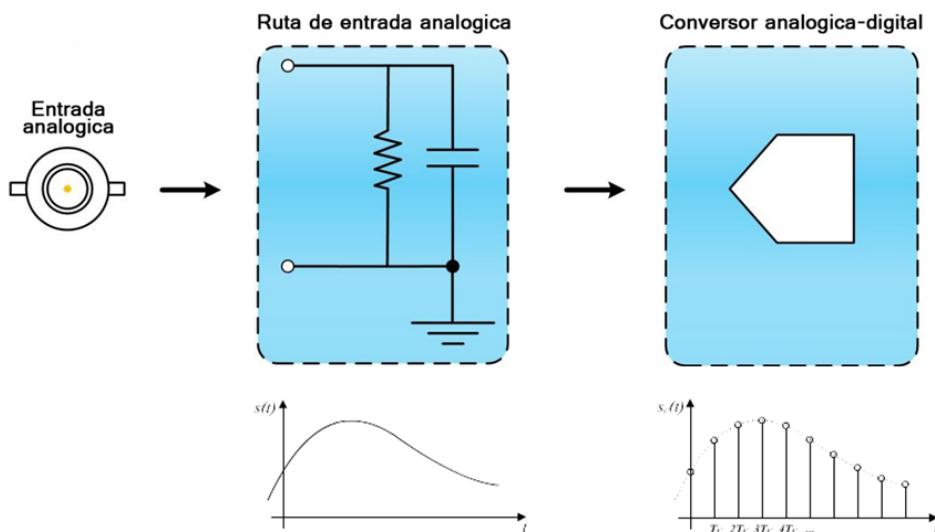


Figura 1 - Diagrama de entrada.

El ancho de banda describe la habilidad del panel de entrada analógico para llevar una señal del mundo exterior al ADC con pérdida de amplitud mínima, desde la punta de prueba a la entrada del ADC. En otras palabras, el ancho de banda describe la diversidad de frecuencias que un osciloscopio puede medir con precisión.

Se define como la frecuencia a la que una señal de entrada sinusoidal se atenúa al 70.7% de su amplitud original, que también se conoce como el punto de -3dB. Si la frecuencia de la onda sinusoidal aumenta mientras se mantiene constante la amplitud, la amplitud medida disminuirá, es decir, conforme aumenta la frecuencia de la señal, disminuye la capacidad del osciloscopio para presentar la señal con exactitud. Por tanto, un osciloscopio con un ancho de banda de 100MHz adquirirá una onda sinusoidal de 100MHz con una amplitud de 1Vpp a sólo 0,7Vpp, lo cual equivale a un error del 30%. Para reducir significativamente este error, debemos tener en cuenta que un tercio del ancho de banda corresponde a un error del 5%, una quinta parte corresponde a un 3%, etc. Por tanto debería utilizar un osciloscopio de 300MHz o incluso de 500MHz para medir una señal de 100MHz.

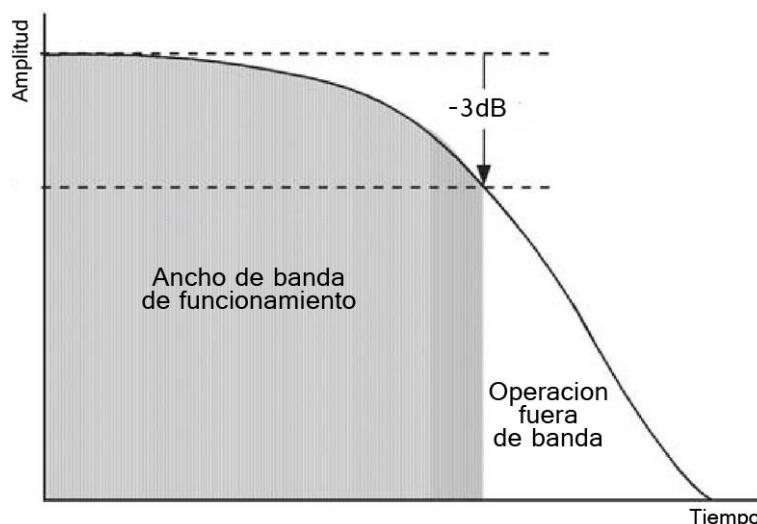


Figura 2 - Gráfica del ancho de banda de funcionamiento.

Velocidad de Muestreo

La velocidad de muestreo es la segunda característica más importante del osciloscopio digital. Especifica con qué frecuencia un osciloscopio digital tomará una muestra de la señal a medir, se especifica en muestras/segundo. Cuanto más rápida es la velocidad de muestreo, mayor será el detalle de la forma de onda presentada, y menor la probabilidad de que se pierda información o eventos de interés.

Para que no se produzca el efecto aliasing se recomienda que la velocidad de muestreo sea al menos 2,5 o 3 veces mayor que el ancho de banda de señal analógica (mejor si es 4 veces superior).

Según el teorema de muestreo de Nyquist/Shannon, la señal a digitalizar no puede incluir componentes de frecuencia superior a la mitad de la frecuencia de muestreo. Además, la señal se debe muestrear a intervalos de tiempo constante. Si no se cumplen estas condiciones se producirán efectos de solapamiento (aliasing). Por tanto, la velocidad de muestreo de un osciloscopio utilizado para medir señales “desconocidas” debería ser lo más alta posible, por ejemplo 1Gs/s, 2Gs/s, 4Gs/s o más en función de la señal a medir.

Para minimizar el muestreo de componentes de frecuencia por encima de la frecuencia de Nyquist y así prevenir el efecto aliasing, la mayoría de los fabricantes de osciloscopios digitales especifican por diseño el ancho de banda de sus equipos en $\frac{1}{4}$ o $\frac{1}{5}$ menos de la máxima frecuencia de muestreo.

Profundidad de Memoria

Se define como la longitud de registro máxima posible para una adquisición de muestras. La señal de entrada se digitaliza por medio del ADC y el flujo de datos resultante se carga en la memoria de alta velocidad del osciloscopio. Muchas veces creemos que la velocidad máxima de muestreo del osciloscopio es aplicable a todos los ajustes de deflexión horizontal, pero la capacidad de memoria del dispositivo es limitada. Por lo tanto, la velocidad de muestreo debe reducirse a medida que aumentan los períodos de muestreo, logrando así que el proceso de muestreo sea bastante lento durante largos intervalos. Cuanto mayor sea la capacidad de memoria de un osciloscopio, más tiempo podrá funcionar a su máxima velocidad de muestreo sin perder información importante de la señal.

Una memoria muy profunda permitirá que los usuarios capturen grandes corrientes de datos. La profundidad de memoria necesaria se puede calcular multiplicando el período de muestreo (o tiempo de muestreo) por la velocidad de muestreo (o frecuencia de muestreo). Los períodos largos de muestreo y las resoluciones temporales elevadas exigen una mayor capacidad de memoria. Se necesita una profundidad de memoria superior a 100 millones de puntos para trabajar con grandes cantidades de datos. La función zoom integrada permite que el usuario amplíe los detalles de una señal para una evaluación más precisa.

La figura muestra la relación entre la velocidad de muestreo, la longitud de registro y el tiempo de muestreo. Cuanto más profunda sea la memoria, más alta podrá mantenerse la velocidad de muestreo al captar intervalos largos de tiempo. Esto, a su vez, proporciona medidas más precisas y fiables, así como la seguridad de que no se omite ningún evento de señal importante.

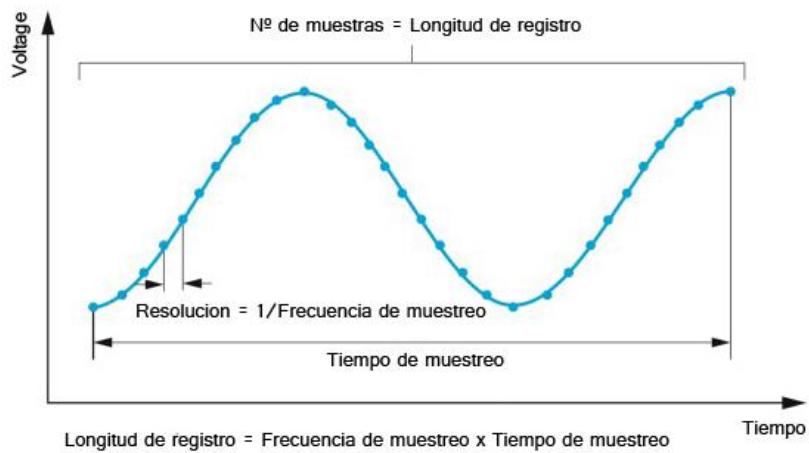


Figura 3 - Relación entre características.

Número de Canales

La selección del número de canales de un osciloscopio suele resultar bastante sencilla ya que los osciloscopios tienen 1, 2 o 4 canales.

Es necesario resaltar, sin embargo, que la velocidad de muestreo se divide entre dos en muchos osciloscopios de 2 canales, y entre cuatro (o dos a dos) en muchos dispositivos de 4 canales. Si se dispone de dos convertidores A/D separados se puede aplicar la máxima velocidad de muestreo. En un dispositivo de 4 canales se pueden emplear dos canales a la máxima velocidad de muestreo (por ejemplo, canales 1 y 3), mientras que para los canales 2 y 4 la velocidad de muestreo es la mitad de su valor máximo. También influye sobre la profundidad de memoria ya que solo se podrá disponer de la mitad de la memoria si se duplica el número de canales.

Velocidad de Captura de forma de onda

Otra característica destacable dentro de las prestaciones de un osciloscopio es la velocidad de captura de forma de onda, que define la frecuencia que utiliza el dispositivo para medir y visualizar los resultados. Cuanto mayor sea la frecuencia, mayor será también la probabilidad de capturar transitorios como las interferencias de señal o glitches.

Mientras que la velocidad de muestreo indica con qué frecuencia el osciloscopio toma muestras de la señal de entrada dentro de una forma de onda o ciclo, la velocidad de captura indica la rapidez con que el osciloscopio adquiere formas de ondas complejas.

Algunos osciloscopios digitales pueden adquirir millones de formas de ondas en cuestión de segundos, lo que aumenta de forma significativa la probabilidad de captura de eventos infrecuentes e intermitentes.

Funciones avanzadas

Algo muy importante del uso del osciloscopio son las funciones avanzadas. Estas se pudieron agregar debido al cambio de osciloscopios analógicos a digitales. A continuación se explican algunas funciones avanzadas de los osciloscopios:

- Decodificador de protocolos o buses seriales: Permite interpretar los paquetes enviados a través de alguna interfaz de comunicación con protocolos como USB, RS-232, I2C, SPI, CAN, LIN, Ethernet, Flexray entre otros. Esta codificación nos ayuda a validar que nuestros diseños funcionen de la manera correcta.
- Mediciones de potencia: Ideales para la investigación y desarrollo en electrónica de potencia, junto con una punta de corriente, permite calcular potencia activa, reactiva y aparente, así como realizar mediciones de distorsión armónica total (THD) y mediciones de rizo. Algunos más avanzados permiten incluso hacer cálculos de pérdidas por conmutación en los transistores y pérdidas magnéticas.
- Mediciones del dominio de la frecuencia: Además de la famosa transformada rápida de Fourier (FFT) que casi todos los osciloscopios tienen, hay algunos que cuentan con analizador de espectro incluido, estos se conocen como MDO (osciloscopios de dominio mixto), además hay algunas opciones para poder correlacionar el dominio del tiempo y de la

frecuencia en el instrumento permitiendo entender mejor los comportamientos de señales complejas.

- Analizador lógico: Antes existían sólo los analizadores lógicos, equipos con varios canales digitales para poder correlacionar múltiples señales digitales de dispositivos como microprocesadores, microcontroladores, etc. Hoy en día hay equipos que pueden tener hasta 64 canales digitales o más sustituyendo a los analizadores lógicos.

Identificación de las partes de un osciloscopio digital

Para tal fin se toma como ejemplo el Osciloscopio OWON SDS1052 siendo uno de los que se tomó como ejemplo en la lista comercial.

Se parte en dividir el equipo en sectores marcados como se indica en la siguiente figura.

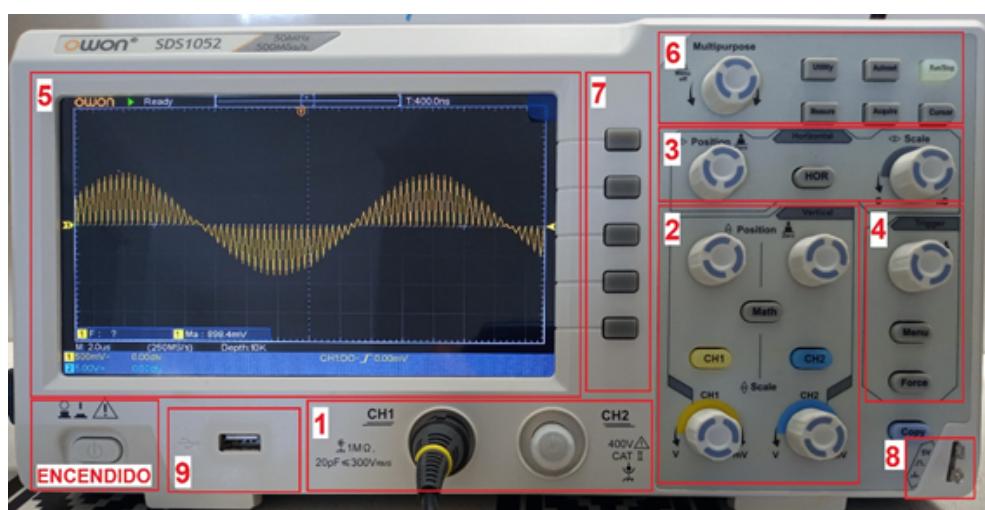


Figura 4 - Frente Osciloscopio OWON SDS1052.

Para empezar a utilizar nuestro osciloscopio, una vez que verificamos la correcta instalación a la red eléctrica (220 Vca / 50 Hz) presionamos el botón de encendido.

Continuando con el uso de nuestro osciloscopio. El mismo (en 1) dispone de dos entradas. Siendo estas, dos canales para la conexión de las puntas de prueba, comúnmente llamadas sonda de medición.



Figura 19 - Entradas / Canales.

Estas entradas están conformadas por conectores BNC Hembra y siempre se busca que el detalle técnico de las sondas supere en ancho de banda a la del osciloscopio.

Ejemplo para este caso:

Ancho de Banda: Osciloscopio
 50MHz

Punta de Prueba
 60MHz

En 2 tenemos los controles verticales que nos dan la posibilidad de cambiar la posición de la onda hacia arriba o hacia abajo en la pantalla, lo que nos resulta muy útil para comparar dos formas de ondas. También podemos cambiar la escala vertical, así la forma de onda se contraerá o ampliará en torno al nivel de tierra.

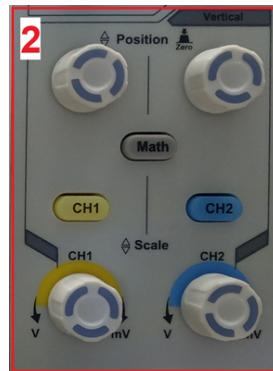


Figura 5 - Controles Vertical.

Los controles para posición horizontal (3) nos brinda la posibilidad de poder ajustar la forma de onda respecto a su posición horizontal. Para el ajuste de escala hacemos variación sobre el tiempo por división para la base de tiempo establecida. Esto nos da la posibilidad de poder visualizar un tramo acotado de la señal y poder distinguir múltiples periodos de ser necesario.

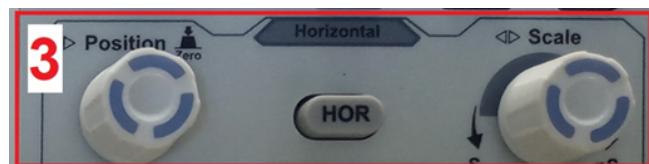


Figura 6 - Control Horizontal.

El control de disparo del osciloscopio (4), determina qué nivel de amplitud debe tener la señal de entrada para adquirir una forma de onda estable. Dentro de Menú encontramos las distintas configuraciones para el funcionamiento del Trigger que serán detalladas en posteriores informes.



Figura 7 - Control Trigger.

Osciloscopio Digital

Medidas Electrónicas II – Facultad Regional Paraná – Universidad Tecnológica Nacional



Pantalla, comúnmente también se lo denomina Display (8). Si bien este módulo está incorporado físicamente en el osciloscopio que usamos como ejemplo. En la actualidad existen diversos modelos que mediante una interfaz USB reproducen las señales en la pantalla de una PC.

Aquí es donde analizaremos las señales, tanto para detectar fallas como para el seguimiento y avance de nuevos desarrollos relaciones con formas de ondas.

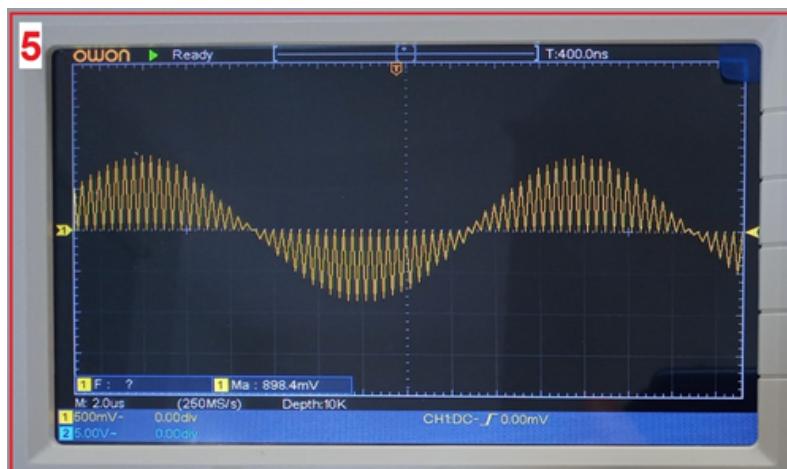


Figura 8 - Pantalla.

El equipo utilizado para ejemplo tiene un bloque multipropósito (6), en el mismo podemos acceder a distintas funciones de uso frecuente.

El Autoset, por ejemplo, ajusta automáticamente los controles del osciloscopio para reproducir las señales de entrada en una escala factible para el análisis.

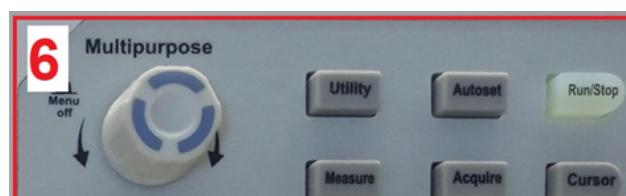


Figura 9 - Bloque multipropósito.

Cuando navegamos por alguna función del menú, con los botones que se encuentran al margen derecho de la pantalla (7) podemos elegir diferentes configuraciones para el análisis de nuestras señales.

Dichos pulsadores son de uso general. No solo para el bloque menú, sino que también nos permiten configurar aquellos parámetros que se desplieguen sobre la pantalla del lateral derecho.

Ejemplo de menú sobre lateral derecho:

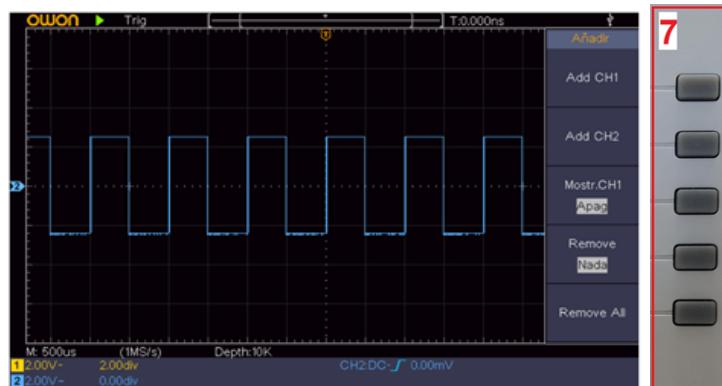


Figura 10 - Botones para selección de opciones.

Cuando se desea realizar un set o bien, una comparación con una señal conocida y reconfigurar los cursores (horizontal y vertical) con los valores que se hayan ajustado, el osciloscopio cuenta con un bloque que genera una señal cuadrada de 5 Vpp y de frecuencia 1KHz.

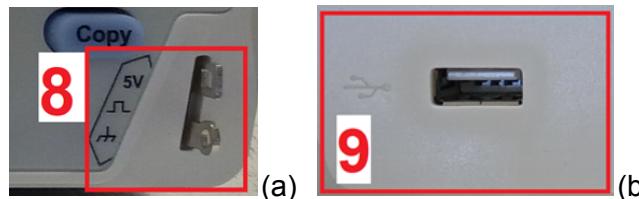


Figura 11 - (a) Señal interna para configuración / (b) Puerto USB.

Señal generada:

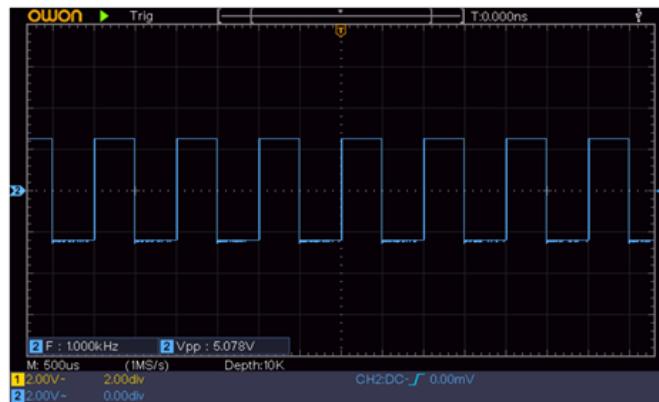


Figura 12 - Señal interna generada.

Mediante el botón (Copy – 8) y tras determinadas configuraciones (utility - 6) podemos capturar imágenes y almacenarlas en un pendrive conectado en la ranura del bloque 9.

Diagrama en bloque

Un osciloscopio digital tiene básicamente algunos de los siguientes bloques funcionales: bloque vertical, bloque horizontal, el bloque de disparo, bloque de adquisición y el por último el bloque de visualización.

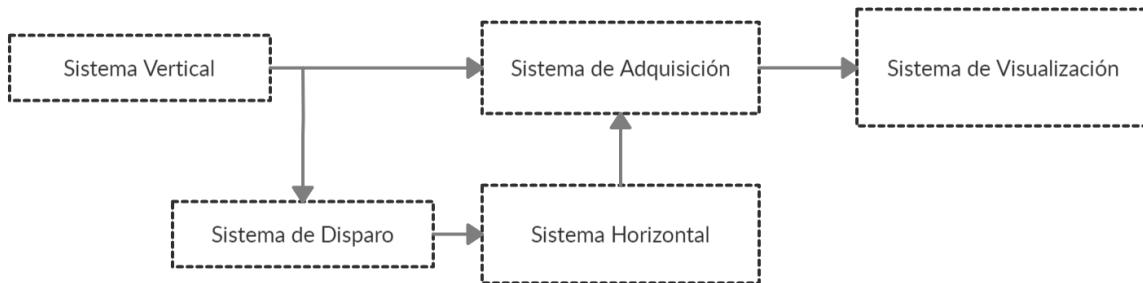


Figura 13 - Diagrama de bloques general.

Una gran parte del diseño de este diagrama se dedican a las funciones de los bloques vertical y horizontal ya que la mayoría de los ajustes necesarios se realizan en ellos para el acondicionamiento de la señal. El bloque vertical se encarga de la atenuación o amplificación de la señal por medio de selectores que regulan la tensión por división y varía así la atenuación o la amplificación de la señal de entrada. El bloque horizontal está relacionado con la base de tiempo del equipo, y la escala de segundos por división determina la cantidad de tiempo por división que mostrará horizontalmente la pantalla. El bloque de disparo tiene la función básica de estabilizar la señal, iniciar el osciloscopio para realizar una adquisición de datos la cual con un ADC convertirá la señal analógica en información digital, una memoria que vendrá controlada por unos contadores y un microcontrolador, el cual se encargará tanto de procesar la señal para ser mostrada. Finalmente, el bloque de visualización incluye la pantalla y sus controladores, así como el software necesario para implementar las numerosas funciones de visualización.

Sistema vertical

Esta parte del osciloscopio es el encargado de recibir la señal o señales permitiéndole al usuario escalar la forma de señal en sentido vertical y que usualmente es la señal que interesa medir en función del tiempo. Su esquema en bloques es el siguiente:

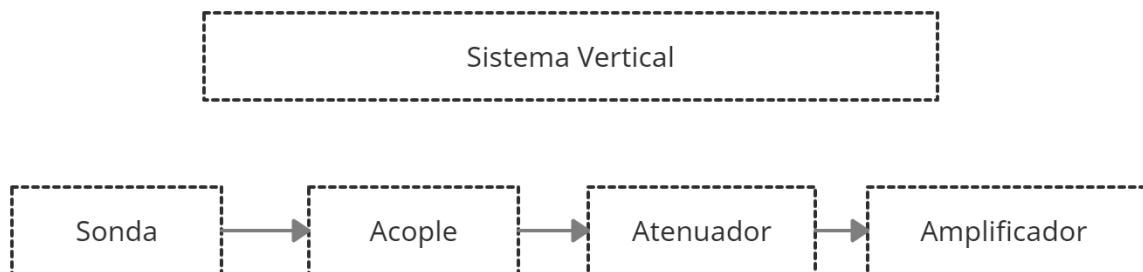


Figura 14 - Bloque del sistema vertical.

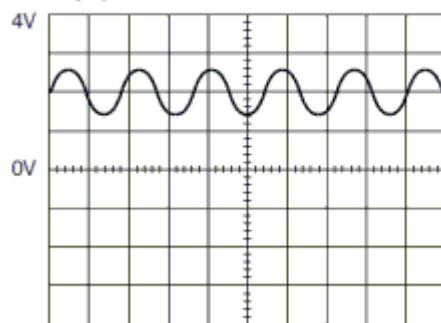
En la figura anterior se reconocen los siguientes elementos:

- **Punta o sonda:** es el dispositivo que provee la conexión del osciloscopio con el circuito a medir, es externo a aquél, y suele proveer funciones de atenuación.
- **Selector de acoplamiento:** permite que la señal ingrese al dispositivo de medición de diferentes maneras:

Acoplamiento DC: Este acoplamiento envía la señal de entrada directamente al amplificador - atenuador, permitiendo de esta manera medir tensiones tanto de corriente continua como de alterna.

Acoplamiento AC: Este acoplamiento es útil cuando la señal completa (con sus componentes DC y AC) es demasiado grande para la pantalla, con la configuración volts/div seleccionada. Al seleccionar este acoplamiento, internamente se intercala en serie con la entrada un capacitor, filtrando la componente DC de la señal, y permitiendo así observar la parte alterna de la misma, centrada en 0V.

Acoplamiento de CC de una onda sinusoidal de 1 V_{p-p} con un componente de 2 V CC



Acoplamiento de CA de la misma señal

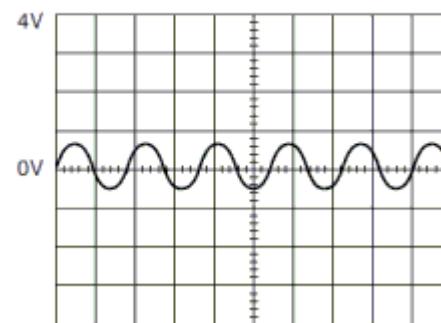


Figura 15 - Acoplamiento de entrada DC - AC.

Acoplamiento GND: La posición GND, envía la entrada del amplificador a tierra o masa, desconectando la señal que ingresa. Esto hace posible poder ubicar la traza, mediante el control de posición vertical en cualquier lugar de la pantalla para que sirva de referencia.

- **Atenuador-amplificador:** tienen por función adecuar los valores de señal a medir. Por razones de diseño, se ve un amplificador de ganancia variable en un número reducido de pasos, y complementarlo con un atenuador, con la cantidad necesaria de niveles de atenuación. El comando que acciona sobre este conjunto es el selector nombrado V/div, accesible en el panel de control del osciloscopio.

En cuanto al atenuador de entrada, la función que realiza es adecuar los valores de la señal a aplicar al amplificador, tiene usualmente la forma de una red RC compensada en frecuencia, por el hecho de que la impedancia de entrada del amplificador es un paralelo RC siendo la respuesta del atenuador independiente de la frecuencia de la señal entrante.

Sistema horizontal

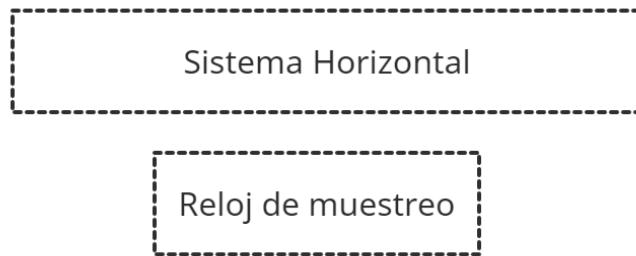


Figura 16 - Bloque del sistema Horizontal.

El bloque horizontal define la escala y la posición del eje de tiempo de la pantalla del osciloscopio, es decir, está más estrechamente relacionado con la adquisición de señal que el sistema vertical, y pone énfasis en parámetros de rendimiento como la velocidad de muestreo. El mismo está compuesto por un generador de barrido que se utiliza para controlar las velocidades de barrido del osciloscopio. Ese generador de barridos es lo que se conoce como base de tiempos.

Funciones del sistema vertical y Horizontal - Circuito implementado

A continuación se mencionan los comandos implementados en el IDE de Arduino para lograr el escalado en el sistema vertical y horizontal. Como se va a indicar, en la pantalla, en qué escala se está trabajando definimos una tabla que contiene las escalas, esta se almacena en la memoria flash del microcontrolador.

En el siguiente cuadro se indican las escalas tanto del sistema vertical como el horizontal.

```
const char vRangeName[10][5] PROGMEM = {" 50V", " 20V", " 10V", " 5V", " 2V", " 1V", "0.5V", "0.2V"}; // Escala vertical  
  
character (number of characters including \ 0 is required)  
const char * const vstring_table[] PROGMEM = {vRangeName[0],  
vRangeName[1], vRangeName[2], vRangeName[3], vRangeName[4],  
vRangeName[5], vRangeName[6], vRangeName[7]};  
  
const char hRangeName[10][6] PROGMEM = {"200ms", "100ms", "50ms", "20ms", "10ms", "5ms", "2ms", "1ms", "500us",  
"200us"}; // Escala horizontal  
  
const char * const hstring_table[] PROGMEM = {hRangeName[0],  
hRangeName[1], hRangeName[2], hRangeName[3], hRangeName[4],  
hRangeName[5], hRangeName[6], hRangeName[7], hRangeName[8],  
hRangeName[9]};  
  
const PROGMEM float hRangeValue[] = { 0.2, 0.1, 0.05, 0.02, 0.01,  
0.005, 0.002, 0.001, 0.5e-3, 0.2e-3}; // Rango del sistema  
horizontal en segundos
```

Dependiendo del valor que tome la variable vRange, lo cual se logra mediante una interrupción y pines configurados de tal manera para poder desplazarnos en un rango definido para tal variable, se puede ir barriendo diferentes casos de una función switch(). Esto permite que podamos escalar la señal muestreada en el alto de la pantalla según el valor que tome la variable rangeMaxDisp.

```
// Tomamos los datos del sistema vertical y horizontal  
almacenados en la memoria flash según el valor de la variable  
hRange y vRange - lsb50V y lsb5V variable de ajuste para  
  
(char*)pgm_read_word(&(hstring_table[hRange])); // H range  
(char*)pgm_read_word(&(vstring_table[vRange])); // V range  
  
switch (vRange) { // Seteo según vRange(0-1 pruebas)  
  
    case 2: { // 50V range  
        rangeMax = 50 / lsb50V; // Cantidad total de pixeles  
        rangeMaxDisp = 5000;  
        rangeMin = 0;  
        rangeMinDisp = 0;  
        break;  
    }  
    case 3: { // 20V range  
        rangeMax = 20 / lsb50V;  
        rangeMaxDisp = 2000;  
        rangeMin = 0;  
        rangeMinDisp = 0;  
        break;  
    }  
    case 4: { // 10V range  
        rangeMax = 10 / lsb50V;  
        rangeMaxDisp = 1000;  
        rangeMin = 0;  
        rangeMinDisp = 0;  
        break;  
    }  
    case 5: { // 5V range  
        rangeMax = 5 / lsb5V;  
        rangeMaxDisp = 500;  
        rangeMin = 0;  
        rangeMinDisp = 0;  
        break;  
    }  
    case 6: { // 2V range  
        rangeMax = 2 / lsb5V;  
        rangeMaxDisp = 200;  
        rangeMin = 0;  
        rangeMinDisp = 0;  
        break;  
    }  
    case 7: { // 1V range  
        rangeMax = 1 / lsb5V;
```

```
    rangeMaxDisp = 100;
    rangeMin = 0;
    rangeMinDisp = 0;
    break;
}
...
...
```

Para el caso de la variable hRange, lo que se afecta es el escalado horizontal. Que al igual que el sistema vertical, la variación de la fracción de señal que se visualiza es concentrada en la evaluación de una función switch(). Donde principalmente se trabaja con el preescalador del microcontrolador y los tiempos de cada operación.

```
switch (hRange) {
    case 0: { // 200ms range
        timeExec = 1600 + 60;
        ADCSRA = ADCSRA & 0xf8;
        ADCSRA = ADCSRA | 0x07;
        for (int i = 0; i < REC LENG; i++) {
            waveBuff[i] = analogRead(0);
            delayMicroseconds(7888);
            if (switchPushed == true) {
                switchPushed = false;
                break;
            }
        }
        break;
    }

    case 1: { // 100ms range
        timeExec = 800 + 60;
        ADCSRA = ADCSRA & 0xf8;
        ADCSRA = ADCSRA | 0x07;
        for (int i = 0; i < REC LENG; i++) {
            waveBuff[i] = analogRead(0);
            delayMicroseconds(3860);
            if (switchPushed == true) {
                switchPushed = false;
                break;
            }
        }
        break;
    }

    case 2: { // 50ms range
        timeExec = 400 + 60;
        ADCSRA = ADCSRA & 0xf8;
        ADCSRA = ADCSRA | 0x07;
        for (int i = 0; i < REC LENG; i++) {
            waveBuff[i] = analogRead(0);
            delayMicroseconds(1880);
        }
    }
}
```

```
        if (switchPushed == true) {
            break;
        }
    }
break;
}

case 3: {                                     // 20ms range
    timeExec = 160 + 60;
    ADCSRA = ADCSRA & 0xf8;
    ADCSRA = ADCSRA | 0x07;
    for (int i = 0; i < REC LENG; i++) {
        waveBuff[i] = analogRead(0);
        delayMicroseconds(686);
        if (switchPushed == true) {
            break;
        }
    }
break;
}

case 4: ...
```

Interrupción del microcontrolador para los sistemas de escala vertical y horizontal

Como se mencionó anteriormente, en los fragmentos de códigos, tanto el bloque vertical como el horizontal se evalúan con el valor que toma cada variable (vRange y hRange correspondientemente). Este valor parte de una interrupción en conjunto con pulsadores externos, donde se puede ir modificando las escalas de cada sistema.

```
void pin2IRQ() {                                // Pin2(int.0) interrupcion

// Pin 8,9,10,11 Los pulsadores se conectan con diodos al Pin2
// Esto hace que al presionar algun pulsador se inicie la funcion

    int x;

    x = PINB;                               // Leer el estado del puerto B
    if ((x & 0x07) != 0x07) {           // si los 3 bits inferiores no
        son todos Hi (cualquiera fue presionado)

        saveTimer = 5000;
        switchPushed = true;
    }
    if ((x & 0x01) == 0) {                // (Pin8)
        scopeP++;                         // Desplazamiento del cursor
        if (scopeP > 1) {
            scopeP = 0;
        }
    }
}
```

```
if (((x & 0x04) == 0) {                      // Up(Pin9)
    if (scopeP == 0) {                         // Si cursor es cero
        vRange++;
        if (vRange > 9) {
            vRange = 9;
        }
    }
    if (scopeP == 1) {                         // Si cursor es uno
        hRange++;
        if (hRange > 9) {
            hRange = 9;
        }
    }
}

if (((x & 0x02) == 0) {                      // DOWN(Pin10)
    if (scopeP == 0) {
        vRange--;
        if (vRange < 2) {
            vRange = 2;
        }
    }
    if (scopeP == 1) {                         // if scoped horizontal range
        hRange--;
        if (hRange < 0) {
            hRange = 0;
        }
    }
}
```

Sistema de disparo



Figura 17 - Bloque del sistema de disparo.

El sistema de disparo de un osciloscopio es sincronizar el barrido horizontal en el punto correcto de la señal, función esencial para una clara caracterización de la señal. Los controles de disparo permiten estabilizar las formas de onda repetitivas y capturar formas de onda de ocurrencia única.

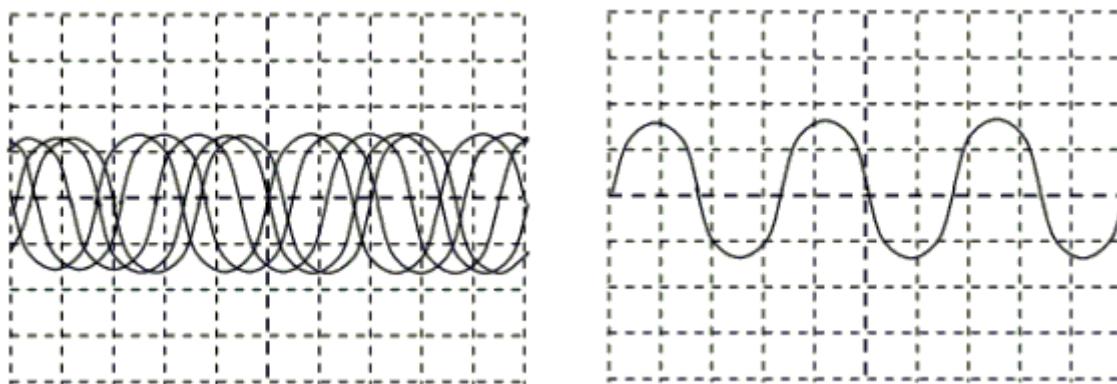


Figura 18 - misma señal, una sin sincronización y otra con sincronización.

El disparo hace que las formas de onda repetitivas parecen inmóviles en la pantalla del osciloscopio, mostrando repetidamente la misma sección de la señal de entrada. El disparo por flanco, disponible en los osciloscopios (tanto analógicos como digitales), es el tipo de disparo básico y más común.

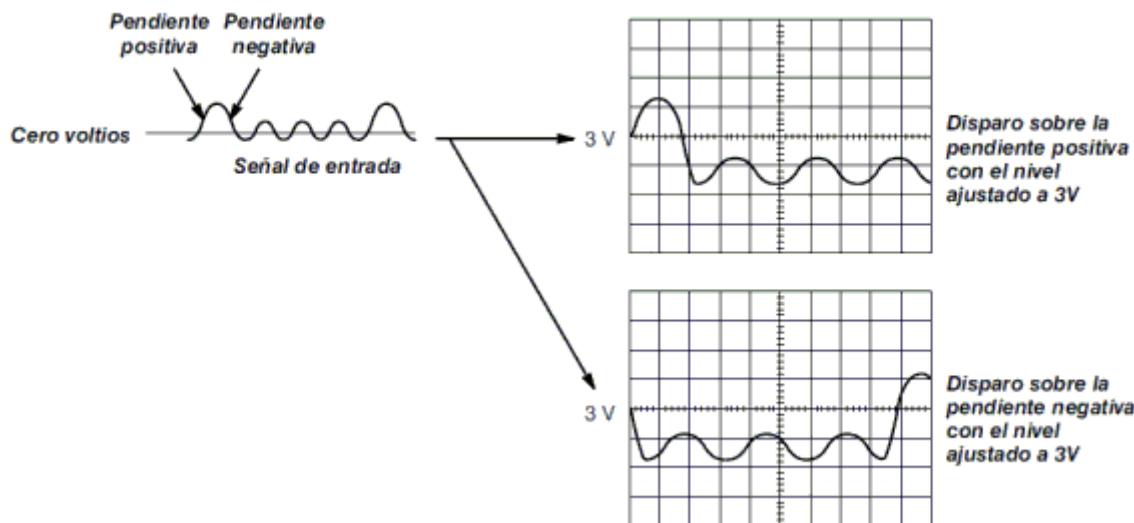


Figura 19 - Disparo por flanco (pendiente) positiva y negativa.

Las capacidades de disparos avanzados de algunos osciloscopios proporcionan un control altamente selectivo. Se puede disparar en pulsos definidos por su amplitud (tales como seudopulsos), cualificados por tiempo (ancho de pulso, espurio ("glitch"), velocidad de transición, tiempos de establecimiento y retención, y lapso de tiempo o "time-out"), y definidos por su estado lógico o patrón (disparo lógico).

En un osciloscopio digital comercial, el sistema de disparo permite sincronizar el barrido horizontal en el punto correcto de la señal, función esencial para una clara caracterización y visualización de la señal. Los controles de disparo permiten estabilizar las formas de onda y capturar formas de onda de ocurrencia única o bien eventos transitorios.

El disparo hace que las formas de onda parecen inmóviles en la pantalla del osciloscopio, mostrando repetidamente la misma sección de la señal de entrada. A continuación se representa una imagen donde cada barrido empieza en un punto diferente de la señal.

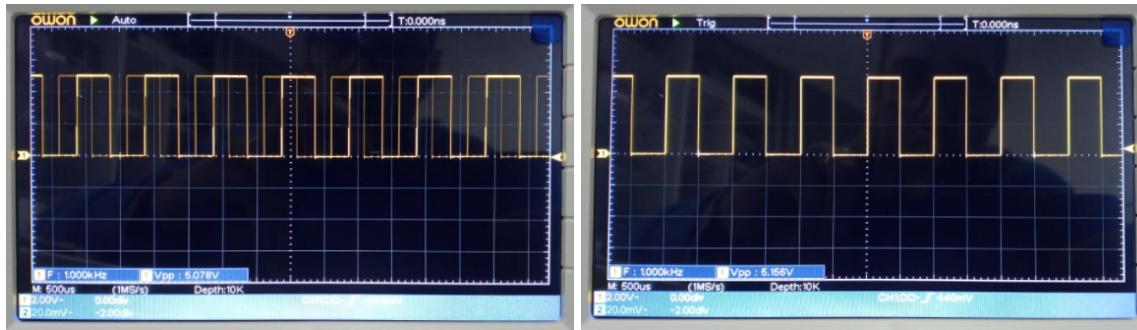


Figura 20 - misma señal, una sin sincronización y otra con sincronización.

El disparo por flanco, disponible en los osciloscopios analógicos y digitales, es el tipo de disparo de uso común. Además del umbral de disparo que ofrecen los osciloscopios analógicos y digitales, muchos osciloscopios digitales ofrecen un conjunto de configuraciones de disparos específicas que no ofrecen los instrumentos analógicos. Estos disparos responden a condiciones específicas de la señal de entrada y facilitan la detección, por ejemplo, de un pulso que sea más estrecho de lo que debería ser. Sería imposible detectar tal condición con un disparo de umbral de voltaje solamente.

Los controles avanzados de disparos nos permiten aislar eventos específicos de interés para optimizar la velocidad de muestreo del osciloscopio y la longitud de registro. Este tipo de disparo presente en algunos osciloscopios proporciona un control altamente selectivo. Se puede disparar en pulsos definidos por su amplitud, calificados por tiempo (ancho de pulso, "glitch", velocidad de transición, tiempos de establecimiento y retención, y lapso de tiempo o "time-out"), y definidos por su estado lógico (disparo lógico).

Los controles de disparo opcional en algunos osciloscopios están específicamente diseñados para analizar las señales de comunicaciones. En muchos casos, estos equipos disponen de controles intuitivos que permiten una rápida configuración de los parámetros de disparo, con una amplia flexibilidad en la configuración para maximizar el análisis de las señales. Cuando se utilizan más de cuatro canales para el disparo sobre señales, la herramienta ideal es un analizador lógico.

Misma relación con osciloscopio OWON SDS1052

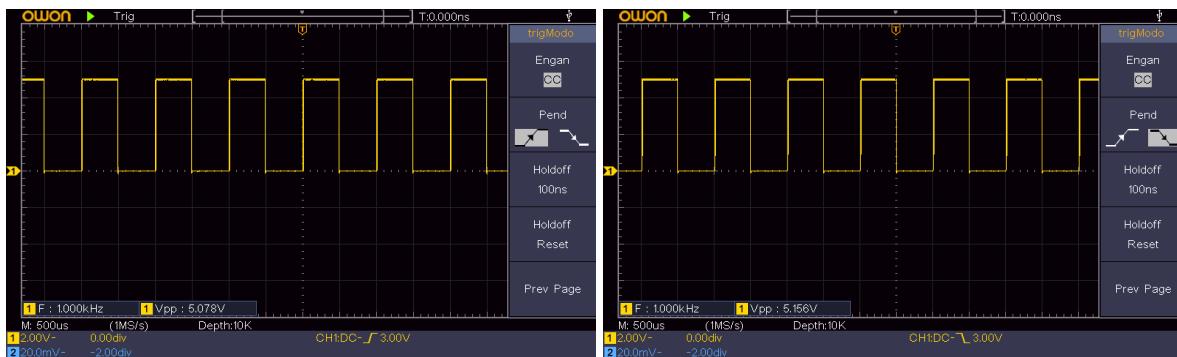


Figura 21 - Disparo en los flanco ascendente 3V y descendente 3V.

En la imagen de arriba podemos ver como el circuito de disparo actúa como un comparador. Donde el usuario mismo tiene la posibilidad de elegir la pendiente y el nivel de voltaje en

una entrada del comparador. Cuando la señal de disparo en la otra entrada del comparador iguala los parámetros seleccionados, el osciloscopio genera un disparo.

- El control de la pendiente determina si el punto de disparo está en el flanco de subida o en el de bajada. Un flanco de subida es una pendiente positiva, y un flanco de bajada es una pendiente negativa.
- El control del nivel determina en qué punto del flanco ocurre el punto de disparo.

Fuentes de disparo

El osciloscopio no necesita disparar en la señal que se está mostrando. En este caso varias fuentes pueden disparar la visualización, como ser:

- Cualquier canal de entrada
- Una fuente externa distinta de la señal aplicada a un canal de entrada
- La señal de alimentación
- Una señal definida internamente por el osciloscopio a partir de uno o más canales

Muchas veces, se puede dejar como fuente de disparo, el osciloscopio configurado para disparar con el canal mostrado. También, algunos osciloscopios proporcionan una salida de disparo para poder enviar la señal de disparo a otro instrumento.

En otras ocasiones, el osciloscopio puede utilizar una fuente de disparo alternativa, esté o no presente, por lo que se deberá cuidar, como ejemplo, de no disparar inadvertidamente en el canal 1 mientras se está mostrando el canal 2.

Modos de disparo

El modo de disparo determina si el osciloscopio representa o no una forma de onda basándose en una condición de señal. Los modos habituales de disparo incluyen el normal y el automático.

En el modo normal, el osciloscopio solamente genera un barrido si la señal de entrada alcanza el punto de disparo establecido, de lo contrario, no aparecerá la señal en la pantalla. Este modo puede desorientar al operador, ya que en un principio no se verá la señal si el control de nivel no está correctamente ajustado.

Para el caso del modo automático, este hace que el osciloscopio genere un barrido, incluso sin que haya disparo. Si no hay una señal presente, un temporizador en el osciloscopio disparará el barrido. Esto asegura que la traza no desaparezca de la pantalla si la señal no genera un disparo.

En la práctica, probablemente se utilizarán los dos modos: el modo normal, porque permite ver exactamente la señal de interés, incluso cuando el disparo ocurre a una velocidad lenta, y el modo automático, porque requiere menos ajustes.

Muchos osciloscopios incluyen también modos especiales para barridos individuales, disparo en señales de vídeo, o configuración automática del nivel de disparo.

Acoplamiento del disparo

Así como se puede seleccionar el acoplamiento de CA o CC para el sistema vertical, también se puede seleccionar el tipo de acoplamiento para la señal de disparo.

Además del acoplamiento de CA y CC, el osciloscopio también puede disponer de acoplamientos de disparo de rechazo a la alta frecuencia, rechazo a la baja frecuencia, y rechazo al ruido. Estas configuraciones especiales son útiles para eliminar el ruido en la señal de disparo y evitar así falsos disparos.

Retención del disparo - Holdoff

A veces se requiere una gran habilidad para conseguir que el osciloscopio dispare en la parte deseada de una señal. Muchos osciloscopios tienen características especiales para facilitar esta tarea, como es el caso de holdoff.

La retención del disparo es un período de tiempo ajustable después de un disparo válido durante el cual el osciloscopio no puede disparar. Esta característica es útil cuando se está disparando sobre formas de onda complejas, de manera que el osciloscopio solamente pueda disparar sobre un punto de disparo válido. En la siguiente figura se muestra cómo la utilización de la retención del disparo ayuda a crear una representación útil, sin desplazamientos.

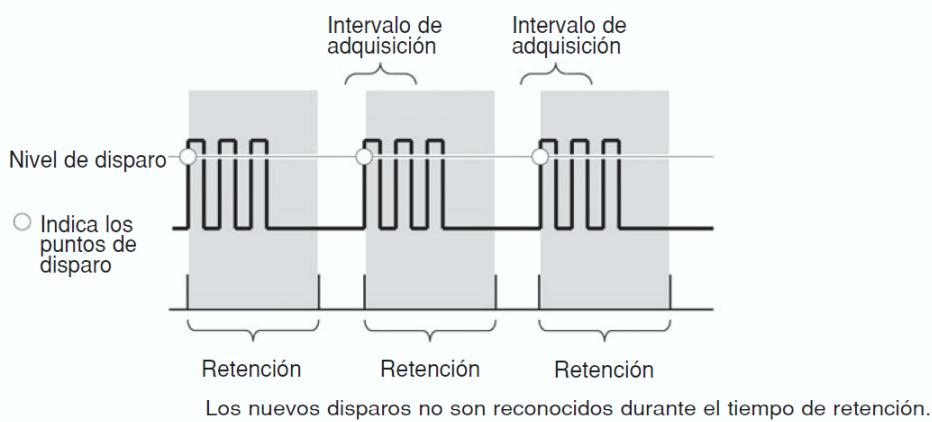


Figura 22 - Holdoff

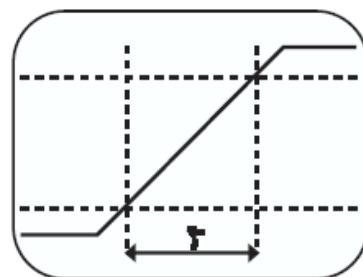
Referencia:

http://www.finaltest.com.mx/v/vspfiles/assets/datasheet/Tutorial_de_osciloskopios.pdf

Tipos de disparo

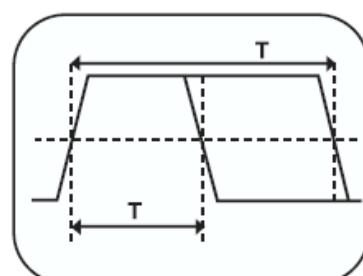
Disparo por velocidad de transición

Se da en el caso de señales de alta frecuencia con tiempos de transición más cortos de lo esperado. El disparo por velocidad de transición supera al disparo por flanco convencional, añadiendo el elemento tiempo y permitiendo el disparo selectivo sobre flancos lentos o rápidos.



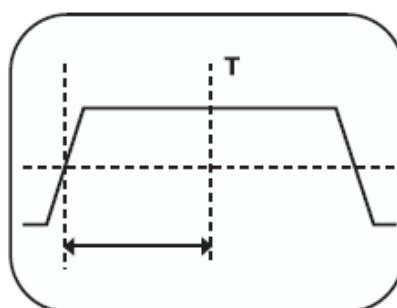
Disparo por ancho de pulso

Utilizando el disparo por ancho de pulso se puede monitorear indefinidamente una señal y efectuar el disparo en la primera ocurrencia de un pulso cuya duración (ancho de pulso) esté fuera de los límites permisibles.



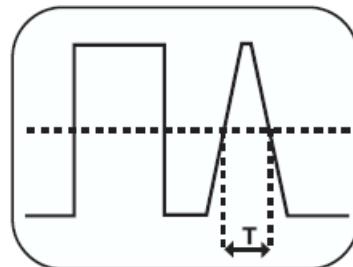
Disparo por lapso de tiempo (timeout)

El disparo por lapso de tiempo permite disparar en un evento sin necesidad de esperar a que termine el pulso del disparo, disparando en base a un lapso de tiempo especificado.



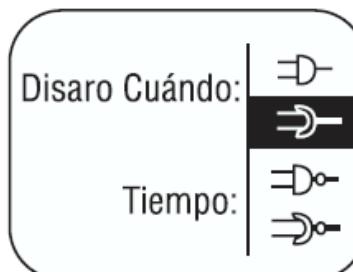
Disparo por armónicos ("glitch")

El disparo por armónicos permite disparar en pulsos digitales que son más estrechos o más anchos que un límite de tiempo definido por el operador del instrumento. Este control de disparo permite examinar las causas de los armónicos y sus efectos sobre otras señales.



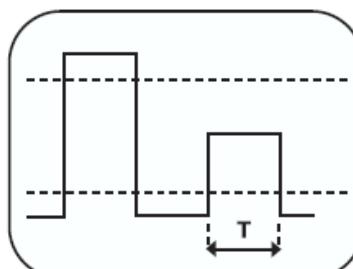
Disparo lógico

El disparo lógico permite disparar en cualquier combinación lógica de canales de entrada disponibles, siendo especialmente útil en la verificación de operaciones de lógica digital.



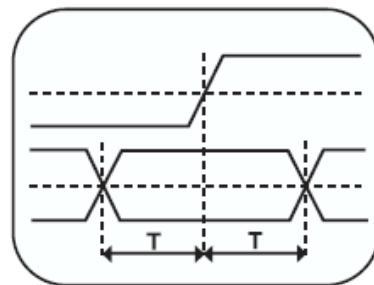
Disparo por seudopulso ("runt")

El disparo por seudopulso permite capturar y examinar pulsos que cruzan un umbral lógico, pero no ambos.



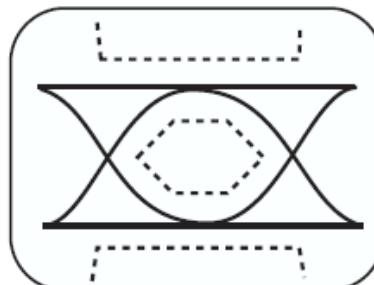
Disparo por tiempo de establecimiento y retención ("Setup" y "Hold")

El disparo por tiempo de establecimiento y retención permite capturar, bajo una forma determinada, un suceso aislado en los tiempos de establecimiento y retención. Este modo facilita la captura de detalles específicos de calidad de la señal y de temporización cuando una señal de datos síncronos no llega a cumplir las especificaciones de los tiempos de establecimiento y retención.



Disparo de comunicaciones

Disponible opcionalmente en ciertos modelos de osciloscopios, estos modos de disparo cubren la necesidad de adquirir una amplia variedad de señales de comunicaciones tales como Inversión Alternada de Marca (AMI), Inversión de Código y Marca (CMI) y No Retomo a Cero(NRZ).



Posición del disparo

El control de la posición del disparo horizontal solamente está disponible en los osciloscopios digitales. El control de la posición del disparo puede estar ubicado en la sección de controles horizontales del osciloscopio. Por lo que este, representa la posición horizontal del disparo en el registro de la forma de onda.

La variación de la posición del disparo horizontal permite capturar lo que ocurrió en una señal antes de un evento de disparo, lo que se conoce como visualización del predisparo. De esta forma, determina la cantidad de señal visible antes y después del punto de disparo.

Los osciloscopios digitales pueden proporcionar una visualización del predisparo porque procesan constantemente la señal de entrada, independientemente de que se haya recibido un disparo o no. Al haber un flujo estable de datos que es leído por el osciloscopio, el disparo le indica a este último que almacene los datos existentes en ese momento en la memoria. Por el contrario, los osciloscopios analógicos solamente presentan la señal, es

decir, la dibujan en el tubo de rayos catódicos (TRO), después de recibir el disparo. Por lo tanto, la visualización del predisparo no está disponible en los osciloscopios analógicos, con la excepción de una pequeña porción de predisparo proporcionada por una línea de retardo en el sistema vertical.

La visualización del predisparo es una valiosa ayuda para el diagnóstico. Si un problema ocurre de forma intermitente, se podrá disparar en el problema, almacenar los eventos que lo originaron y, posiblemente, localizar la causa.

Nivel y pendiente de disparo

Los controles de nivel y pendiente de disparo proporcionan la definición básica del punto de disparo y determinan cómo se presentará una forma de onda, según se ilustra en la siguiente figura.

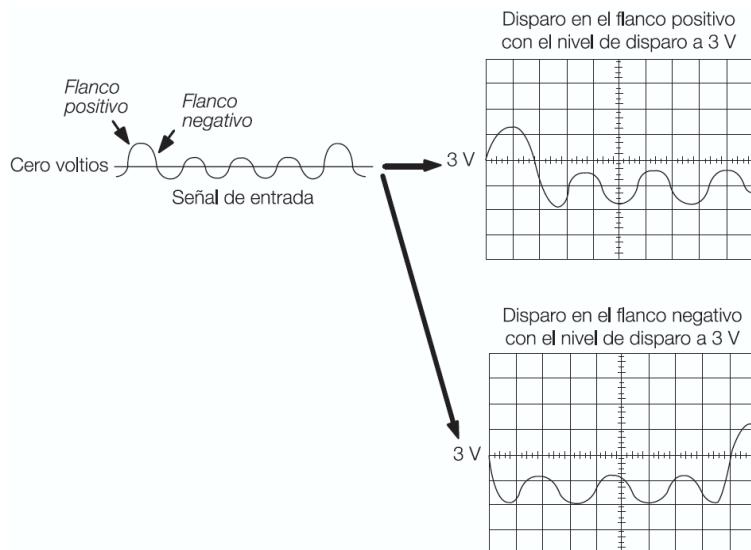


Figura 23 - Disparo en los flancos positivos y negativos.

Sistema de adquisición

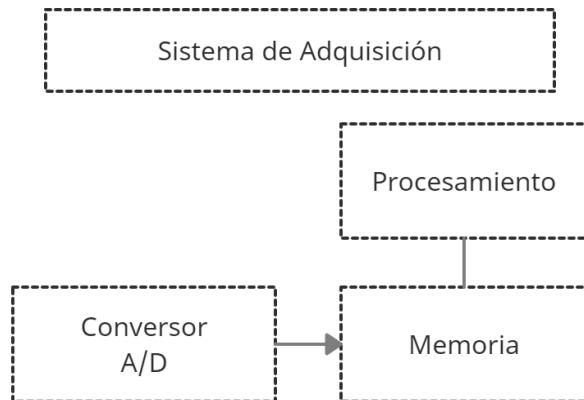


Figura 24 - Bloque del sistema de adquisición.

El conversor analógico-digital (A/D) del sistema de adquisición de datos muestrea la señal a intervalos de tiempo determinados y convierte la señal continua de voltaje en una serie de valores digitales llamados muestras. La magnitud de cada punto de muestra es igual a la amplitud de la señal de entrada en el momento en que la señal es muestreada.

En el sistema horizontal una señal de reloj determina cuando el conversor A/D toma una muestra. La velocidad de este reloj se denomina velocidad de muestreo y se mide en muestras por segundo, tal como se aprecia en la siguiente figura.

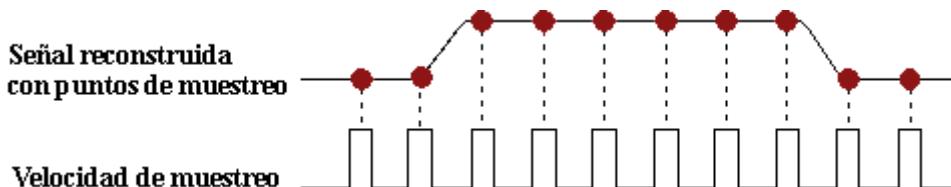


Figura 25 - Muestreo en tiempo real.

Los valores digitales muestreados se almacenan en una memoria como puntos de señal. El número de los puntos de señal utilizados para reconstruir la señal en pantalla se denomina registro. El sistema de disparo determina el comienzo y el final de los puntos de señal en el registro. El sistema de visualización recibe estos puntos del registro, una vez almacenados en la memoria, para presentar en pantalla la señal.

Dependiendo de las capacidades del osciloscopio se pueden tener procesos adicionales sobre los puntos muestreados.

Interrupción como método de disparo

Las interrupciones son recursos o mecanismos del microcontrolador para responder a eventos, permitiendo suspender temporalmente el programa principal, para ejecutar una subrutina de servicio de interrupción (ISR por sus siglas en inglés Interrupt Service Routines); una vez terminada dicha subrutina, se reanuda la ejecución del programa principal.

Las interrupciones se generan cuando dispositivos periféricos conectados a la tarjeta electrónica solicitan enviar información al microcontrolador, esto puede ser de manera asíncrona. También el proceso de interrupción se puede generar de manera periódica, es decir por medio de una señal digital (por ejemplo de un milisegundo de período) conectada a un pin específico del microcontrolador (INT0 o INT1) se puede atender tareas determinadas como adquisición de datos, monitoreo de sensores, cálculos numéricos, etc. La idea es que, en vez de estar preguntando continuamente en el programa si ha ocurrido una eventualidad, éste se dedique a hacer otras cosas, y al momento de ocurrir, se genere una notificación de interrupción para que ésta sea atendida. Por ejemplo, al ocurrir un cambio en un pin (entrada) de un puerto, al finalizar una conversión del ADC, al llegar un registro de temporización a un determinado valor, etc.

Cuando se genera una interrupción, la unidad de interrupciones indica al microcontrolador que un evento requiere solicitud de interrupción, el microcontrolador consulta los registros de la unidad de interrupciones para determinar qué tipo de interrupción se está generando. También guarda el estatus del contador de programa, con el número de interrupción se determina por medio de la tabla de vectores de interrupción (ver tabla 1) con las direcciones de la memoria flash donde se ubican las rutinas de servicio de interrupción ISR, de esta forma se puede acceder y ejecutar el correspondiente código de la ISR. Las direcciones donde residen las rutinas de servicio de interrupción ISR (tabla de vectores) se encuentran en las primeras localidades de memoria del espacio de programa, iniciando en la localidad 0x0000 con la interrupción con mayor prioridad: RESET.

En general, independientemente de la arquitectura del microcontrolador ocurren los siguientes eventos:

- a) Finaliza la instrucción que estaba ejecutando.
- b) Se desactiva el habilitador global de interrupciones, para que no pueda recibir una nueva interrupción mientras atiende a la actual.
- c) Incrementa el valor del registro apuntador de instrucción o program counter (PC), que es la dirección de retorno de la interrupción.
- d) Respalda el valor del PC en la pila (o Stack, normalmente se ubica como una porción predefinida de la memoria de datos).
- e) Asigna al PC el valor del vector de interrupciones, para dar paso a la rutina de interrupción.
- f) Al finalizar la interrupción , se recupera de la pila el valor del PC, que contiene la dirección de retorno de la rutina de interrupción, recordar que ésta es la línea siguiente de la cual partió.
- g) Retorna de la interrupción.

Se pueden distinguir dos tipos de Interrupciones:

- Interrupciones por Hardware: Son aquellas que son provocadas por dispositivos externos al microcontrolador, como sensores, pulsadores, interruptores, entre otros. Su característica es que no son programadas, esto es, pueden ocurrir en cualquier momento del programa.
- Interrupciones por software: son aquellas causadas por una acción ejecutada por el código principal.

Fuentes de Interrupción en el ATMega328P

Los Microcontroladores AVR 328P poseen 26 fuentes de interrupción. Cada una de estas fuentes de interrupción tiene asociado un vector de interrupción, que no es más que una dirección en el espacio de memoria del programa. La dirección más baja le corresponde al vector de interrupción por RESET. Ésta es la interrupción de mayor prioridad.

La tabla 1 presenta las interrupciones del microcontrolador Atmega 328P (8 bits, arquitectura AVR) que forma la plataforma electrónica del modelo Arduino UNO.

Interrupciones del microcontrolador Atmega328

Vector	Localidad	Fuente	Características	Identificador
1	0x000	RESET	Pin externo, reset al encender (<i>power-on</i>), apagón parcial (<i>brown-out</i>), reset generado por <i>watchdog system</i> .	
2	0x002	INT0	Solicitud de interrupción externa 0, pin 2.	INT0_vect
3	0x004	INT1	Solicitud de interrupción externa 1, pin 3.	INT1_vect
4	0x006	PCINT0	Solicitud de interrupción cambio de pin 0 (pins 8 al 13).	PCINT0_vect
5	0x008	PCINT1	Solicitud de interrupción cambio de pin 1 (pins A0 a A5).	PCINT1_vect
6	0x00A	PCINT2	Solicitud de interrupción cambio de pin 2 (pins 0 al 7).	PCINT2_vect
7	0x00C	WDT	Interrupción tiempo-fuera del Watchdog.	WDT_vect
8	0x00E	TIMER2 COMPA	Timer/Contador2 Match A.	TIMER2_COMPA_vect
9	0x010	TIMER2 COMPB	Timer/Contador2 compara Match B.	TIMER2_COMPB_vect
10	0x012	TIMER2 OVF	Sobreflujo del Timer/Contador2.	TIMER2_OVF_vect
11	0x014	TIMER1 CAPT	Captura de eventos Timer/Contador1.	TIMER1_CAPT_vect
12	0x016	TIMER1 COMPA	Compara Match A del Timer/Contador1.	TIMER1_COMPA_vect
13	0x018	TIMER1 COMPB	Compara Match B del Timer/Contador1.	TIMER1_COMPB_vect
14	0x01A	TIMER1 OVF	Sobreflujo del del Timer/Contador1.	TIMER1_OVF_vect
15	0x01C	TIMER0 COMPA	Compara Match A del Timer/Contador0.	TIMER0_COMPA_vect
16	0x01E	TIMER0 COMPB	Compara Match B del Timer/Contador0.	TIMER0_COMPB_vect
17	0x020	TIMER0 OVF	Sobreflujo del Timer/Contador0.	TIMER0_OVF_vect
18	0x022	SPI, STC	Transferencia serial completa SPI.	SPI_STC_vect
19	0x024	USART, RX	Recepción serial completa RX.	USART_RX_vect
20	0x026	USART, UDRE	Registro vacío de datos.	USART_UDRE_vect
21	0x028	USART, TX	Transmisión serial completa TX.	USART_TX_vect
22	0x02A	ADC	ConveISRón completa del ADC.	ADC_vect
23	0x02C	EE READY	Memoria EEPROM lista.	EE_READY_vect
24	0x02E	ANALOG COMP	Comparador analógico.	ANALOG_COMP_vect
25	0x030	TWI	Interface serial 2-cables (2-wire serial interface I2C).	TEL_vect
26	0x032	SPM READY	Listo para almacenar programa en memoria.	SPM_READY_vect

Prescaler

Antes de hablar de pre-escaladores necesitamos conocer los TIMERS del Arduino.

Es indudable que realizar una medición de tiempo precisa en un proyecto es de suma importancia por lo que se deberá entender cómo trabajar con los temporizadores.

El Timer con Arduino está fuertemente relacionado con lo que vimos en la explicación anterior ya que los timers serán ejecutados por medio de una interrupción para atender alguna tarea en específica, siendo la gran virtud de ésta técnica poder temporizar el tiempo en que deseamos que se ejecute dicha interrupción mientras simultáneamente estamos realizando otra tarea con la placa de desarrollo.

En resumen, el Timer del Arduino es un Temporizador que se programa en el Arduino que se dispara en un tiempo predeterminado. Y en cada momento que este temporizador se dispara, el Arduino ejecuta una función de Interrupción. Por ejemplo, se puede programar un timer que se dispare a cada segundo, o a cada 3 segundos y así sucesivamente, para que el Arduino cumpla una tarea específica.

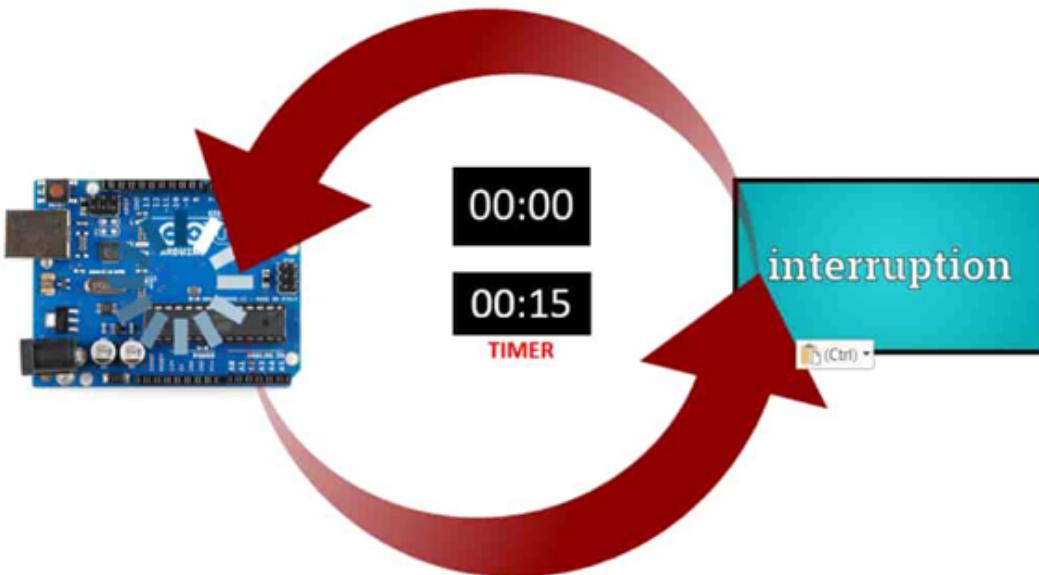


Figura 26 - Ciclo interrupción.

Modos de timers

Cuando hablamos de Timer en Arduino, podemos pensar en 2 modos de funcionamiento en los cuales se puede aplicar esta importante característica.

- Una señal PWM.
- CTC (Clear timer on compare match): Esta es la opción que vamos a utilizar y simplemente como su nombre lo indica, el sistema contabiliza un tiempo dentro de un contador y cuando este contador alcanza el valor del registro de los timers, se ejecuta la interrupción del Arduino.

Tipos de timers

Dado que el timer es un recurso del microcontrolador del Arduino podremos realizar el conteo de tiempo sin la necesidad de ningún hardware externo.

Cada placa de Arduino posee un cristal de cuarzo, para nuestro caso en el Arduino NANO tenemos un cristal de cuarzo de 16MHz. Ese cristal es el encargado de darle los tiempos de ejecución (CLOCK) del programa interno del Arduino.

El Timer se vale de ese cristal, donde teóricamente podríamos ejecutar una interrupción cada 1/16.000.000 segundos (que son 62,5 [ns]) o lo que se conoce como 1 “ciclo de clock”. Cada instrucción consume “x” ciclos de reloj, entonces tenemos que dividir este ciclo por la frecuencia de dicho reloj. La mayoría de las instrucciones consumen un ciclo; por eso se considera que la arquitectura AVR da (casi) 1MIPS por Mhz.

Sin embargo, para ejecutar las instrucciones propias del Arduino, muchas veces consumimos más de 1 ciclo de clock. Los Timers disponibles en Arduino poseen un registro el cual nos dice cuantos ciclos son necesarios para disparar el timer.

Estos son:

Timer 0: Es de 8bits. Usado para las funciones delay(), millis(), micros(). Es conveniente NO modificar este TIMER para evitar alterar estas funciones que son muy comunes en nuestros códigos.

Timer 1: Es de 16 bits. Se usa principalmente para el control de Servos, sin embargo, si no estás controlando servos, puedes usar este timer 1 arduino libremente.

Timer 2: Es de 8 bits. Usado en la función tone(). Si no usas este módulo, puedes emplear el timer 2 Arduino libremente para temporizar otra aplicación.

Prescaler y Registros de Comparación

Dado que el timer0 y el timer2 son temporizadores de 8 bits, significa que pueden almacenar un valor de contador máximo de 255. El Timer1 es un temporizador de 16 bits, lo que significa que puede almacenar un valor de contador máximo de 65535. Una vez que un contador alcanza su máximo valor, este volverá a cero. (Esto se llama desbordamiento).

Para controlar la velocidad del incremento del temporizador utilizamos el prescaler el cual establece la velocidad de temporización con la siguiente ecuación:

$$\text{velocidad del temporizador(Hz)} = \frac{\text{velocidad del reloj de Arduino (16MHz)}}{\text{prescaler}}$$

Por lo tanto:

- un preescalador de 1, incrementará el contador a 16MHz;
- un preescalador de 8 lo incrementará a 2MHz;
- un preescalador de 64 = 250kHz, y así sucesivamente.

El prescaler puede ser igual a 1, 8, 64, 256 y 1024.

Set de instrucciones ATMega328p

Table 15-6. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{IO} /1 (no prescaling)
0	1	0	clk _{IO} /8 (from prescaler)
0	1	1	clk _{IO} /64 (from prescaler)
1	0	0	clk _{IO} /256 (from prescaler)
1	0	1	clk _{IO} /1024 (from prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

La frecuencia de interrupción viene dado por la siguiente ecuación:

$$\text{frecuencia de interrupción(Hz)} = \frac{\text{velocidad de reloj Arduino (16MHz)}}{\text{prescaler. (registro de comparación + 1)}}$$

El +1 está ahí porque el registro de comparación está indexado a cero.

Al reorganizar la ecuación anterior, se puede resolver el valor de registro de comparación que le dará la frecuencia de interrupción deseada:

$$\text{registro de comparación} = \left\lceil \frac{16[\text{MHz}]}{\text{prescaler. frecuencia de interrupción(Hz)}} \right\rceil - 1$$

Entonces a modo de ejemplo si se desea una interrupción a cada segundo (**1[Hz]**) con un prescaler de 1024 se obtiene:

$$\text{registro de comparación} = \left\lceil \frac{16[\text{MHz}]}{1024.1} \right\rceil - 1 = 15624$$

Y como **256 < 15624 < 65536**, debe usar el **TIMER1** para esta interrupción.

Registros del TIMER

Internamente, el microcontrolador es manejado por diferentes **registros**. La gran ventaja de Arduino es que muchas veces no tenemos que entrar en detalle cómo configurar estos registros internos, ya que la propia interfaz de Arduino hace toda la configuración por nosotros. Sin embargo, en aplicaciones como el Timer, si no empleamos librerías, deberemos optar por modificar estos registros manualmente.

Configuración de timers

- En el modo CTC para poder activar las interrupciones cada que ocurra un desbordamiento del TIMER. Esto sucede cuando el registro contador TCNTn (registro contador. Por lo general lo inicializaremos en 0 al comenzar) coincide con el valor del registro de comparación de salida OCRnA/B (registro comparador de salida. Luego de definir nuestro pre-escalador, este registro lo usaremos con un valor específico para que las interrupciones ocurran cuando el valor del contador (TCNTn) coincida con este registro).
- La configuración de los timers de Arduino se hacen dentro de setup() del Sketch siempre desactivando las interrupciones antes de comenzar a configurar el/los timer/s y re-activarlas al final.
- La configuración del código de la interrupción se predefine con la función:

```
ISR(interrupcion) {  
    // Su código va aquí  
}
```

Finalmente, los registros que usaremos son

- **TCCRnA/B** (registros de control para poder configurar el timer en modo CTC y el pre-escalador).
 - **Timer0/2**
 1. **TCCRnA** debe estar todo en 0 excepto por el bit WGM01 que debe estar en 1 para configurar el modo CTC.
 2. **TCCRnB** debe estar todo en 0 excepto los bits que vayamos a usar en 1 para nuestro pre-escalador, si es que usaremos uno.
 - **Timer1**
 1. **TCCRnA** debe estar todo en 0 para nuestro uso.
 2. **TCCRnB** debe estar todo en 0 excepto por el bit WGMn2 que debe estar en 1 para configurar el modo CTC y los bits del pre-escalador que necesitemos si es que usaremos uno.
- TIMSKn** (registro de configuración de interrupción. En nuestro caso lo usaremos con todos los bits en 0 menos el bit OCIEnA que deberá estar en 1 para indicarle a nuestro timer que usaremos el registro OCRnA para comparar).

Muestreo de la señal

A diferencia de presentaciones anteriores, donde veíamos la posibilidad de utilizar los datos que se enviaban por el puerto serie. Fuimos notando que ese proceso se volvía un poco más complejo de que si optamos por un muestreo periódico y almacenamiento de los datos en un vector, para luego poder ser representados en la pantalla y que además, tenemos la posibilidad de poder ajustar el valor correspondiente a cada índice a una escala que prefiramos y así, poder tener controles de sistema vertical y horizontal (tema a tratar en la última presentación).

Aquí, se representa un fragmento de código donde llenamos un vector ($n = 200$) con las mediciones del puerto analógico A0

```
for (int i = 0; i < CantMuestras; i++)
{
    VecMuestras[i] = analogRead(0); // tiempo de lectura y
    escritura
    delayMicroseconds(688); // tiempo de ajuste
}
```

Para entender el funcionamiento de las muestras, como primera medida, en la pantalla se dibujó una cuadrícula para suponer que entrarían 4 ciclos de una señal periódica de 50[Hz]. Por lo que, cada columna tendrá un tiempo de 20[ms].

En este caso, la lectura del puerto analógico más el delay de 688[us], el vector tendría 25 muestras de 8 ciclos, dando un total de 200 muestras. A modo de ejemplo se ilustra la siguiente imagen.

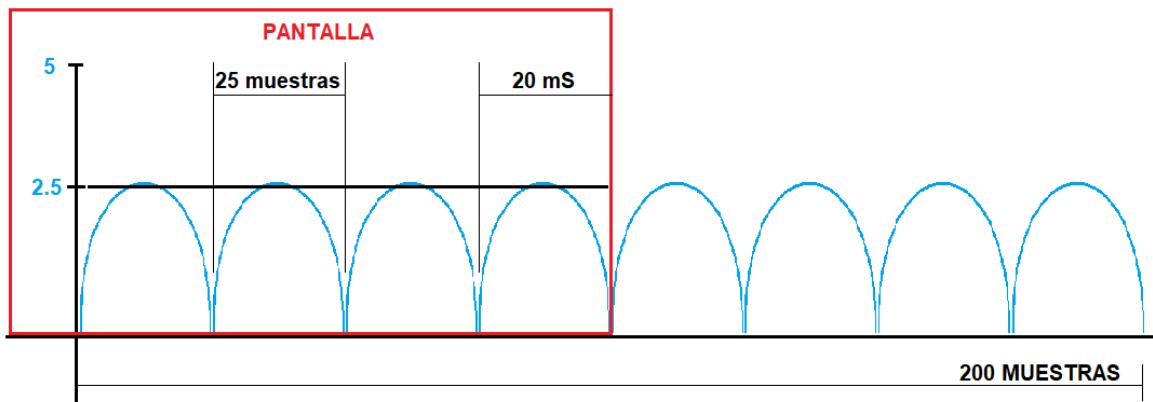


Figura 27 - Muestreo.

Dichas pruebas y las posteriores se realizaron con un generador de formas de ondas XR2206, el cual fue seteado para que entregara una señal senoidal de 50[Hz].

Valores automáticos - Frecuencia y Voltaje

En este apartado se realiza una descripción del ciclo del programa para poder determinar la frecuencia y el voltaje de forma automática.

Frecuencia

La función freq() opera en primera medida ubicando el valor medio de la señal, luego se realiza un recorrido sobre el vector que contiene las muestras ubicando la primer posición donde se da que el promedio de tres valores ($n-1$, n y $n+1$) es menor o igual al valor medio y la siguiente muestra de tres elementos es mayor que dicha muestra promedio. Luego mediante un contador se va almacenando la cantidad de muestras hasta encontrar otra igualdad como la anteriormente definida. Esta cantidad de muestras por el tiempo que demora cada muestra nos dará el periodo de la señal. Por lo que la frecuencia es el resultado del inverso del periodo.

Fragmento del código implementado para el cálculo de la frecuencia.

```
void freq() {
    int swingCenter;                                // Centro de la onda
    float p0 = 0;                                    // Primera pocision
    float p1 = 0;                                    // Longitud de un ciclo
    float p2 = 0;                                    // Ultima pocision
    float pFine = 0;
    float lastPosiEdge;

    float pPeriod;                                  // Pulso-Periodo

    int p1Count = 0;
    int p2Count = 0;

    boolean posiSercch = true;

    swingCenter = (3 * (dataMin + dataMax)) / 2;   // Calculo centro

    for (int i = 1; i < REC LENG - 2; i++) {
        if (posiSercch == true) { // Buscamos pocision de del
            periodo

            if ((sum3(i) <= swingCenter) && (sum3(i + 1) > swingCenter))
            { // la comparacion se realiza con tres elementos para descartar
            el ruido
                pFine = (float) (swingCenter - sum3(i)) / ((swingCenter -
                sum3(i)) + (sum3(i + 1) - swingCenter)); // Calculo del punto
                de inicio
                if (a0Detected == false) {
                    a0Detected = true;
                    p0 = i + pFine;
                }
                else {
                    p1 = i + pFine - p0;                      // longitud
                    p1Count++;
                }
            }
        }
    }
}
```

```
        lastPosiEdge = i + pFine;
        posiSerch = false;
    }
} else {
    if ((sum3(i) >= swingCenter) && (sum3(i + 1) <
swingCenter)) {
        pFine = (float)(sum3(i) - swingCenter) / ((sum3(i) -
swingCenter) + (swingCenter - sum3(i + 1)) );
        if (a0Detected == true) {
            p2 = p2 + (i + pFine - lastPosiEdge);
            p2Count++;
        }
        posiSerch = true;
    }
}
}

pPeriod = p1 / p1Count; // pulse period

waveFreq = 1.0 / ((pgm_read_float(hRangeValue + hRange) *
pPeriod) / 25.0); // frecuencia
}

int sum3(int k) { // Funcion que devuelve la suma de 3
elementos
    int m = waveBuff[k - 1] + waveBuff[k] + waveBuff[k + 1];
    return m;
}
```

Voltaje

El valor que toma la variable que representa el valor medio de voltaje de la señal muestreada es equivalente al promedio de las muestras que están contenidas en el vector de las muestras.

```
for (int i = 0; i < REC LENG; i++) { d = waveBuff[i];
    sum = sum + d;
...
}
dataAve = (sum + 10) / 20;
...
voltage = dataAve * lsb5V / 10.0; // Rango de
evaluacion
if (voltage < 10.0)
{
    // Si es menor que
10V
    dtostrf(voltage, 4, 2, chrBuff); // formato x.xx
}
else {
    dtostrf(voltage, 4, 1, chrBuff); // formato xx.x
}
oled.setCursor(98, 0); // Ubicacion del cursor
oled.print(chrBuff); // display average
```

Sistema de visualización



Figura 28 - Bloque del sistema de visualización.

El sistema de visualización de un osciloscopio incluye la pantalla de presentación y los controles, teclas, interruptores, e indicadores, utilizados para el control de la adquisición de la señal y de su presentación.

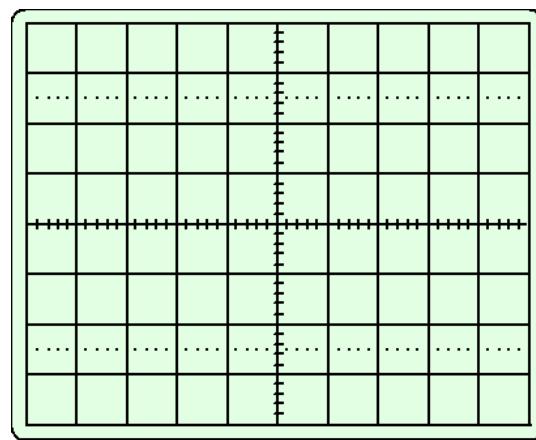


Figura 29 - Pantalla de osciloscopio.

Las muestras almacenadas en la memoria se visualizan en la pantalla usando distintas técnicas. Entre las más comunes tenemos:

Muestreo: Esta es la forma de muestreo más simple. El osciloscopio crea el punto de forma de onda (un punto en la pantalla) a partir de un punto muestreado en cada ciclo.

Detección de pico: El osciloscopio digital guarda el mínimo y el máximo de los puntos muestreados tomados durante dos intervalos de forma de onda, y usa estos puntos como los dos puntos de forma de onda correspondientes

Los controles del panel frontal de un osciloscopio, generalmente están divididos en las secciones horizontal, vertical y de disparo. El panel frontal incluye también los conectores de entrada. La pantalla se presenta usualmente en un patrón de divisiones. La rotulación en los controles del osciloscopio (p. ej., volt/div y sec/div) siempre se refiere a divisiones principales. Las acotaciones en las líneas verticales y horizontales de la retícula, se denominan divisiones menores. Muchos osciloscopios muestran en la pantalla los voltios que representa cada división vertical y los segundos que representa cada división horizontal.

Si consideramos un osciloscopio tradicional, donde el panel frontal incluye la pantalla de presentación y los controles, vamos a observar unas marcas en la pantalla, estas marcas crean la retícula. Donde, cada línea vertical y horizontal constituye una división principal. Esto se ve reflejado en la rotulación de los controles del osciloscopio (volt/div y tiempo/div) que relaciona las divisiones principales. Las pequeñas marcas en las líneas verticales y horizontales de la retícula, como se indica en la siguiente figura, se denominan divisiones menores o secundarias. Muchos osciloscopios muestran en la pantalla los voltios que representa cada división vertical y los segundos que representa cada división horizontal.

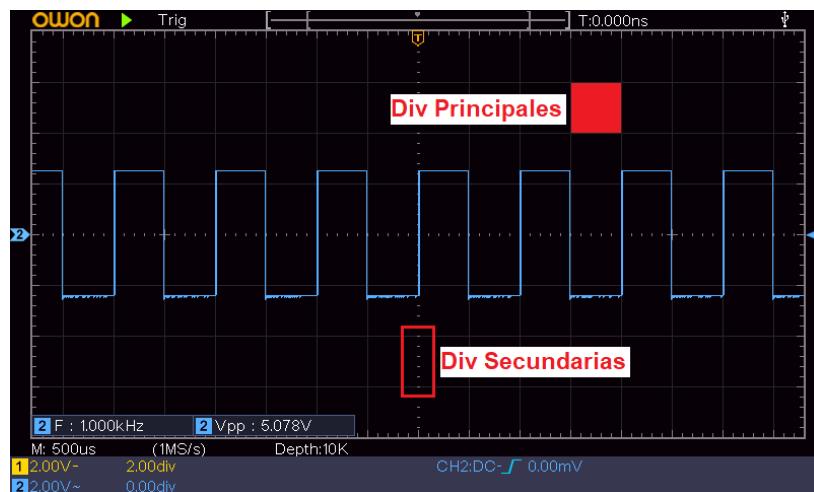


Figura 30 - Cuadrícula de un osciloscopio.

Los sistemas de visualización varían entre los osciloscopios analógicos y los osciloscopios digitales. Entre los controles habituales se incluyen:

Control de intensidad para ajustar el brillo de la forma de onda. Conforme se incrementa la velocidad de barrido de un osciloscopio analógico, se necesita aumentar el nivel de intensidad.

Control de enfoque para ajustar la nitidez de la forma de onda, y un control de rotación de la traza, para alinear la traza de la forma de onda con el eje horizontal de la pantalla. Los osciloscopios digitales, que utilizan pantallas basadas en barrido y LCD, pueden no disponer de estos controles porque el total de la pantalla está predeterminado como en la pantalla de una PC. Por el contrario, los osciloscopios analógicos utilizan una pantalla de haz electrónico directo o vectorial.

En muchos DSO y en los DPO, una paleta de color controla la selección de colores de la traza y los niveles de color de gradación de intensidad.

Otros controles de presentación pueden permitir el ajuste de la intensidad de iluminación de la retícula y la presentación o ausencia de cualquier información adicional en la pantalla, como puede ser las opciones del menú.

Referencia:

http://www.finaltest.com.mx/v/vspfiles/assets/datasheet/Tutorial_de_osciloskopios.pdf

Pantalla OLED

Las Pantallas OLED, por las siglas en inglés de Organic Light-Emitting Diode que traducido sería diodo orgánico de emisión de luz, son pantallas compuestas por láminas de materiales orgánicos como el carbón (de aquí el nombre de diodo orgánico). Estas láminas emiten luz cuando se les aplica electricidad entre ellos.

Una de las ventajas de las pantallas OLED con respecto a la pantalla LCD es que no requieren de una luz de fondo ni de filtros. Esto hace que las pantallas OLED sean más eficientes en términos de energía, más fáciles de fabricar y mucho más finas. A parte pueden ser flexibles y transparentes.

En nuestro caso, usamos una pantalla OLED de 0,96" con controlador SSD1306. Donde, para poder mostrar datos e información en ellas se necesita un driver o controlador. Normalmente viene todo ensamblado en un mismo módulo como el que implementamos.

Básicamente lo que hace el SSD1306 es comunicar con el microcontrolador para obtener los datos y enviarlos a la pantalla OLED para que dibuje esos datos. La comunicación entre el SSD1306 y el microcontrolador se realiza mediante I2C. El tamaño 0,96" tiene una distribución de 128x64 pixeles y es de dos colores, azul y amarillo.

En la siguiente tabla mostramos los rangos de alimentación para el SSD1306 y la pantalla OLED por separado:

Parámetro	Valor
Voltaje de operación SSD1306	1,65[V] a 3,3[V]
Voltaje de operación pantalla OLED	7[V] a 15[V]

Visualización de gráficos en pantalla OLED

Para mostrar los datos en la pantalla, el controlador SSD1306 tiene una memoria RAM gráfica que se llama GDDRAM (viene del inglés Graphic Display Data RAM) que ocupa 1[kB]. Esto equivale a 1.024 bytes o 8.192 bits que se distribuyen en la pantalla en una matriz de filas (páginas) y columnas (segmentos). En total hay 8 páginas (filas) y cada página tiene 128 segmentos (columnas) que, a su vez, cada segmento almacena 1 byte. Por lo tanto, tenemos una matriz de 128 columnas y 64 filas.

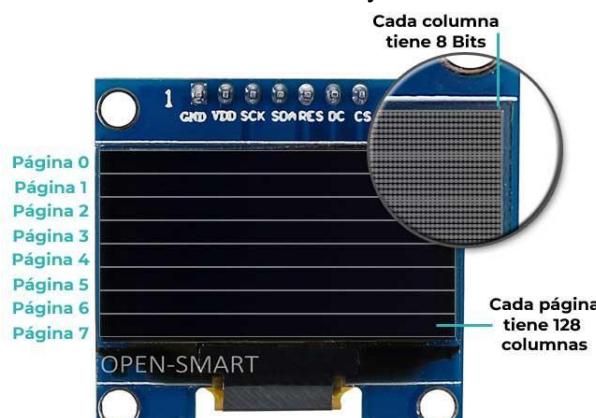


Figura 31 - Pantalla OLED.

Cada bit representa un píxel en la pantalla OLED y mediante la programación, se puede encender o apagar para que muestre cualquier información. De eso se encargan las librerías de Arduino como mencionaremos más adelante.

Pines de conexión módulo pantalla OLED con Arduino

La pantalla OLED que mostramos a continuación es una pantalla que utiliza la interfaz I2C. En este tipo de pantallas vas a encontrar 4 pines.

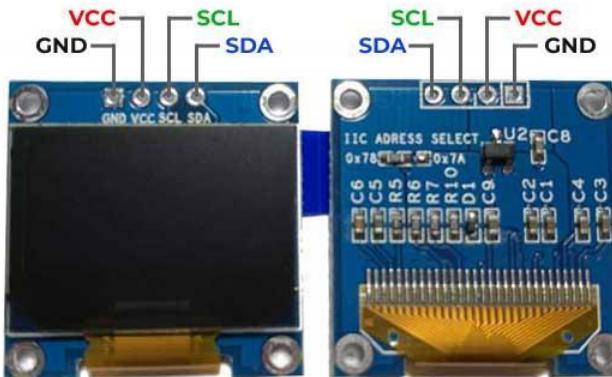


Figura 32 - Pines de conexión de la pantalla OLED.

GND: pin de tierra.

VCC: es el pin de alimentación. Se puede alimentar la pantalla entre 1,8[V] y 6[V].

SCL: es el pin de la señal de reloj de la interfaz I2C.

SDA: es el pin de la señal de datos de la interfaz I2C.

Conexión pantalla OLED con Arduino I2C

En el conexionado solo debemos conectar la alimentación VCC, el GND y luego los pines SCL y SDA del controlador a dos pines analógicos del arduino como mostramos en la siguiente imagen.

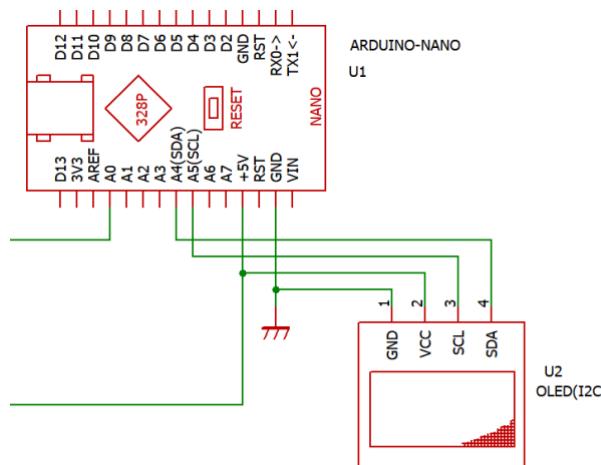
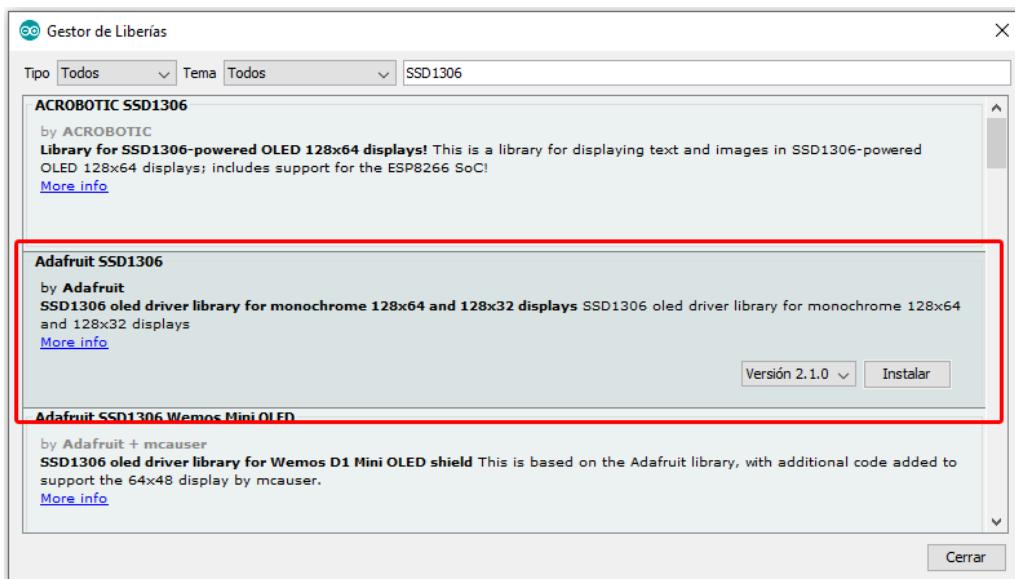


Figura 33 - Conexionado de la pantalla OLED.

Librerías pantalla OLED con Arduino

Para que el funcionamiento de este tipo de pantallas sea sencillo y correcto, lo primero que vamos a hacer es instalar las librerías: Adafruit_SSD1306 y Adfruit-GFX-Library desde el gestor de librerías del IDE de arduino.



Haciendo click en el botón de **Install all** instalará las dos librerías.

Estas dos librerías tienen propósitos diferentes. Cada una de ellas tiene básicamente:

- Librería Adafruit GFX: es la librería gráfica, es decir, la que se encarga de dibujar en la pantalla OLED. Proporciona una serie de gráficos primitivos como puntos, líneas, rectángulos, círculos, etc.

Para poder dibujar o pintar en la pantalla se debe de combinar con una librería de hardware específica para cada dispositivo. Para nuestro caso de pantalla OLED hay que combinarla con la librería Adafruit SSD1306.

- Librería Adafruit SSD1306: Esta librería es la específica para las pantallas OLED basadas en el controlador SSD1306. Funciona tanto para pantallas I2C como SPI.

Imagen de la pantalla utilizada

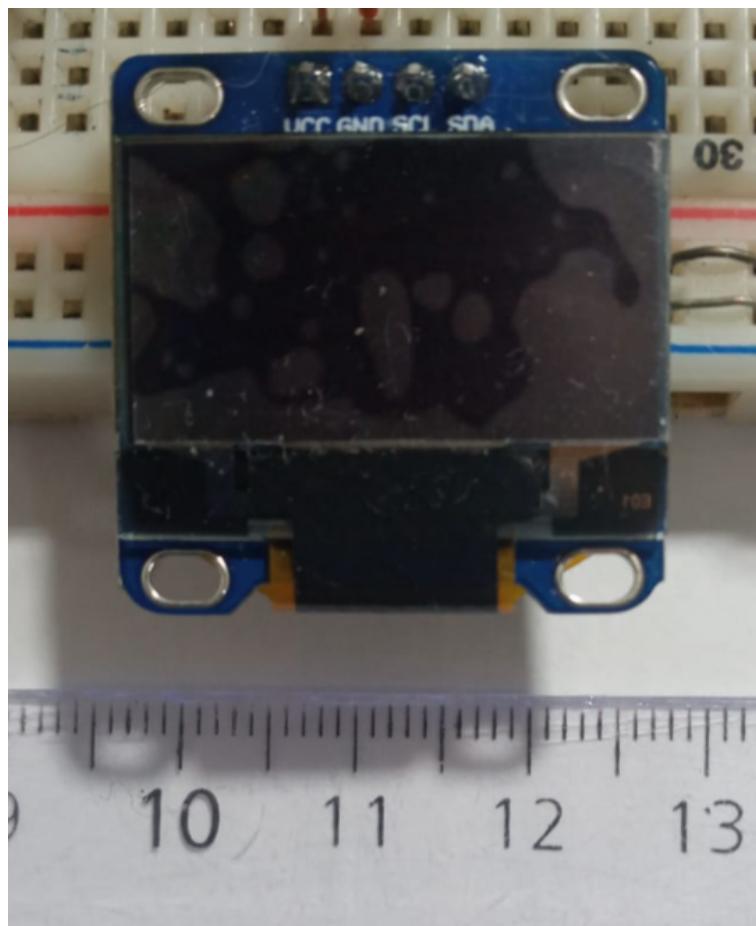


Figura 34 - Pantalla OLED de 0,956” utilizada para la implementación

Trigger de la señal

Otro punto mencionado, es el de poder visualizar la señal sin que tenga desplazamiento hacia los laterales. Es decir, que exista una sensación de que la forma de onda se encuentra estática.

Esto lo logramos comentando la siguiente muestra de código. Aquí, el valor de trigP sería el punto de trigger (punto a considerar para dar inicio a la visualización de la señal). Dentro del for evaluamos las muestras centrales.

```
for (trigP = ((CantMuestras / 2) - 51); trigP < ((CantMuestras / 2) + 50); trigP++)
{
    if ((VecMuestras[trigP - 1] < (valorMax + valorMin) / 2) &&
        (VecMuestras[trigP] >= (valorMax + valorMin) / 2))
    {
        break;
    }
}
```

Si se cumple la condición del IF, en valor de trigP será el punto de trigger. En la siguiente imagen vemos que la muestra correspondiente a dicho punto tendrá un valor aproximado a valorProm y la señal será graficada a partir de ese valor.

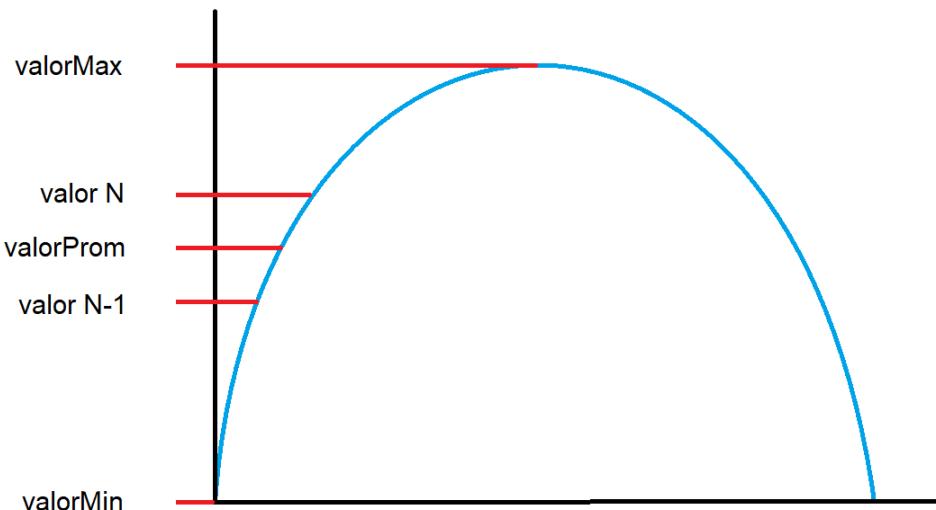


Figura 35 - Esquema para ejemplo de trigger.

Representación en pantalla

En este apartado se explica la función plotData que nos permitirá construir la gráfica en la pantalla OLED. Primero definimos dos variables largas de 32 bits (“long”) y1 e y2 para asegurarnos de no producir un overflow. Creamos un ciclo “for” que una vez se ingrese al mismo se repita 99 veces.

Luego con la función “map” la cual nos retorna el valor a mapear de la señal muestreada y que utiliza parámetros como; el vector VecMuestras que contiene los valores de la señal, un rango mínimo actual (rangoMin) así como uno final (63) y un rango máximo actual (rangoMax) así como uno final (17) (esto se logra debido a que “límites inferiores” de cualquiera de los rangos pueden ser mayores o menores que los “límites superiores”, por lo que ésta función se puede utilizar para invertir un rango de números). El valor de rangoMin se encuentra igualado a 0 y el valor de rangoMax a 1023.

```
void plotData()
{ long y1, y2;
  for (int x = 0; x <= 98; x++)
  {
    y1 = map(VecMuestras[x + trigP - 50], rangoMin, rangoMax, 63,
17);
    y1 = constrain(y1, 16, 63);
    y2 = map(VecMuestras[x + trigP - 49], rangoMin, rangoMax, 63,
17);
    y2 = constrain(y2, 16, 63);
    oled.drawLine(x + 27, y1, x + 28, y2, WHITE);
  }
}
```

La función “constrain” lo que nos permite es restringir los números para que se mantengan dentro de un rango establecido retornando el valor de la señal si es que se encuentra dentro del rango, el valor del límite inferior si es que dicho valor es menor y el valor del límite superior si es mayor.

Luego se llama a la función “drawLine()” la cual se encuentra en la librería “Adafruit_SSD1306.h” que nos dibuja una línea, indicando en la misma, las coordenadas de sus extremos y el color. Como se puede observar x+27 y x+28 lo que producen es un corrimiento de x, para lograr que exista un margen a la izquierda y que la gráfica no comience desde el borde de la pantalla, de esta manera se queda un espacio libre para ubicar las leyendas del eje correspondiente al voltaje.

Señal representada en la pantalla OLED

Datos de la señal:

Frecuencia: 50 Hz

Vpp: 2.1 V

Vmax: 2.9 V

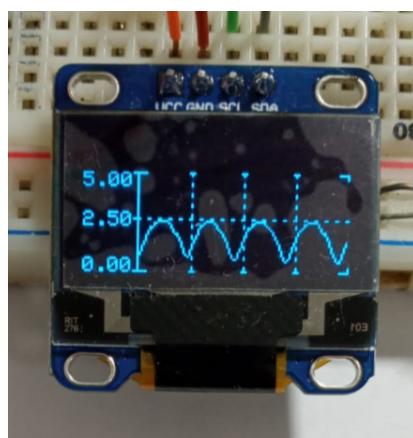


Figura 36 - Representación de la señal en la pantalla OLED.

Comparación contra Osciloscopio Digital OWON SDS1052

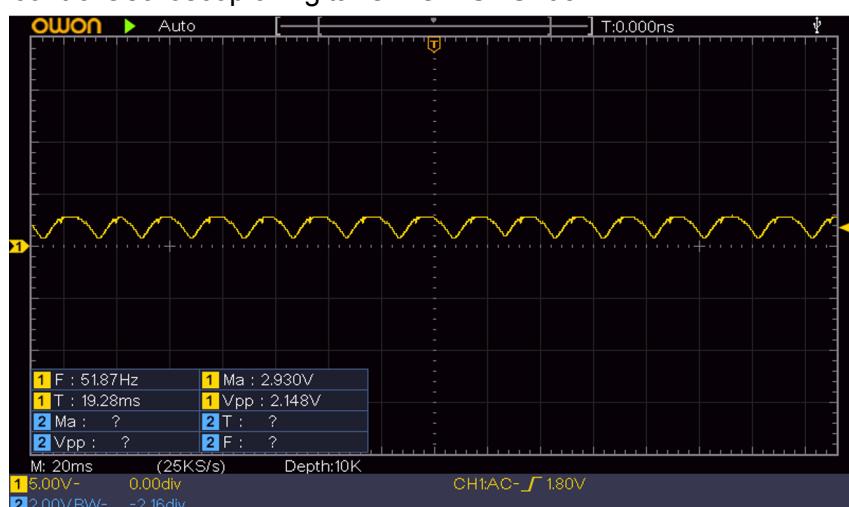


Figura 36 - Representación de la señal en la pantalla de un Osc. comercial.

Función de Hold

Esta función permite que podamos congelar los datos mostrados en la pantalla y así de esta manera poder analizarlos con mayor detalle. La misma opera por una interrupción en el pin 2 en conjunto con el pin 11 del microcontrolador. Donde mediante una variable booleana que determina si el muestreo se continúa haciendo o si se representa la última muestra tomada antes de que se detectó el cambio de la variable hold.

```
void pin2IRQ() {                                     // Interrupcion Pin2(int.0)
    if ((x & 0x08) == 0) {                         // Si el pulsador del pin 11
        se                                         presiona
        hold = ! hold;                            // Cambia el estado de la
                                                    variable hold
    }
    ...
oid loop() {
    ...
    while (hold == true) { //Mientras la variable hold se
                           mantenga true queda en un bucle
                           mostrando los valores de una muestra
                           dada
        dispHold();
        delay(10);
    }
}
```

Mostramos en pantalla la palabra Hold mediante la función oled.print()

```
void dispHold() {
    oled.setCursor(2, 9);
    oled.print(F("Hold"));                      // Imprimimos en pantalla Hold
    oled.display(); }
```

Imagen de la palabra Hold en la pantalla OLED

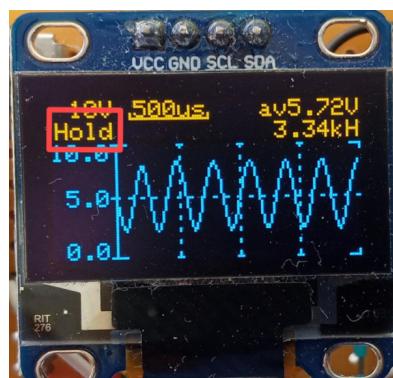


Figura 37 - Representación de la función Hold.

Implementación del osciloscopio

Hasta aquí se ha ido estudiando y describiendo cada una de las etapas del osciloscopio implementado, en relación a su funcionamiento y firmware implementado, el cual pasamos a describir sus partes y funcionalidades.

Esquema de conexión

En el siguiente esquema se representa el conexionado implementado que se utilizó para el desarrollo y funcionamiento del microprocesador Atmega 328P - Arduino nano, en donde, mediante una pantalla OLED se lo puede utilizar como un osciloscopio digital dinámico para uso en laboratorio de electrónica básica.

Esquemático

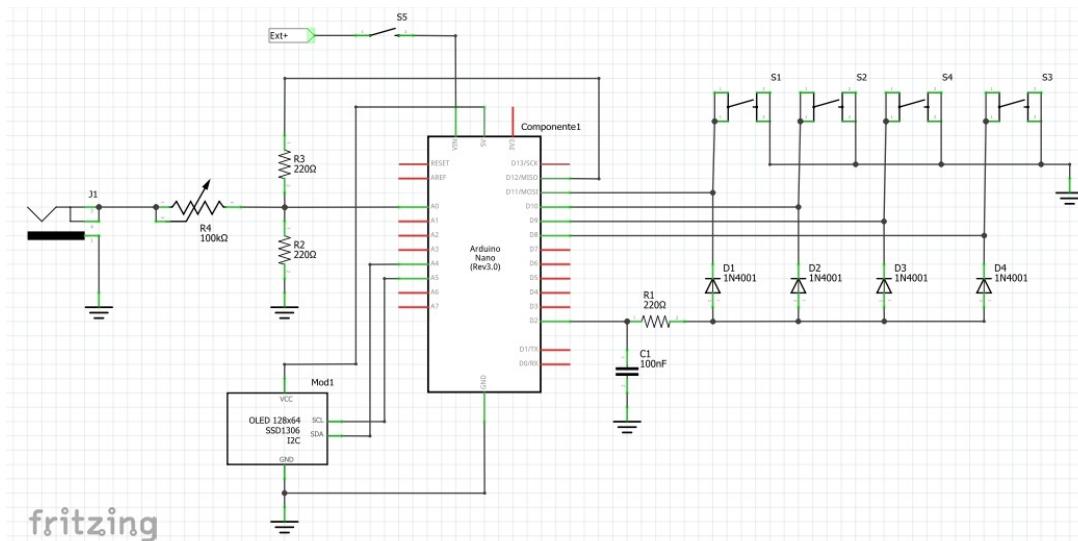


Figura 38 - Esquemático del circuito implementado.

Implementación en placa experimental

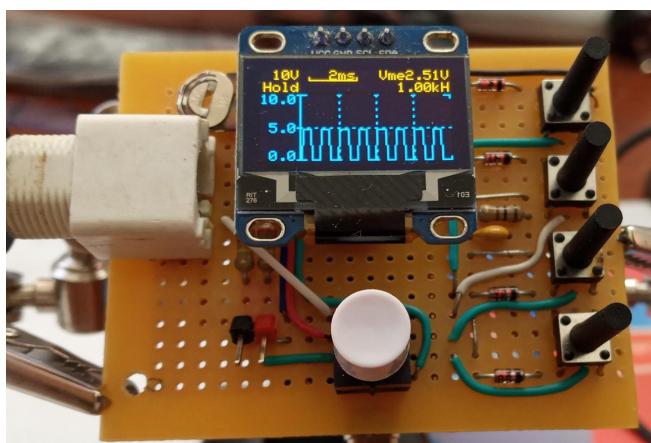


Figura 39 - Implementación en placa experimental.

Diseño 3D del PCB

El objetivo de implementar un osciloscopio, es la posibilidad de realizar un instrumento que se adapte a los usos básicos en un laboratorio de electrónica, como el poder detectar tensiones de alimentación, señales periódicas, señales provenientes de algún sensor, entre otras. Que a su vez, las dimensiones del mismo permitan catalogarlo como un instrumento de bolsillo.

Para ello, se recurrió al software SolidWork el cual nos permite realizar diseños en 3D, sumado a la posibilidad de poder diseñar un gabinete que se ajuste a las características mencionadas.

A continuación, se pueden observar las imágenes resultantes del diseño de la placa para el osciloscopio a desarrollar realizadas en SolidWork.

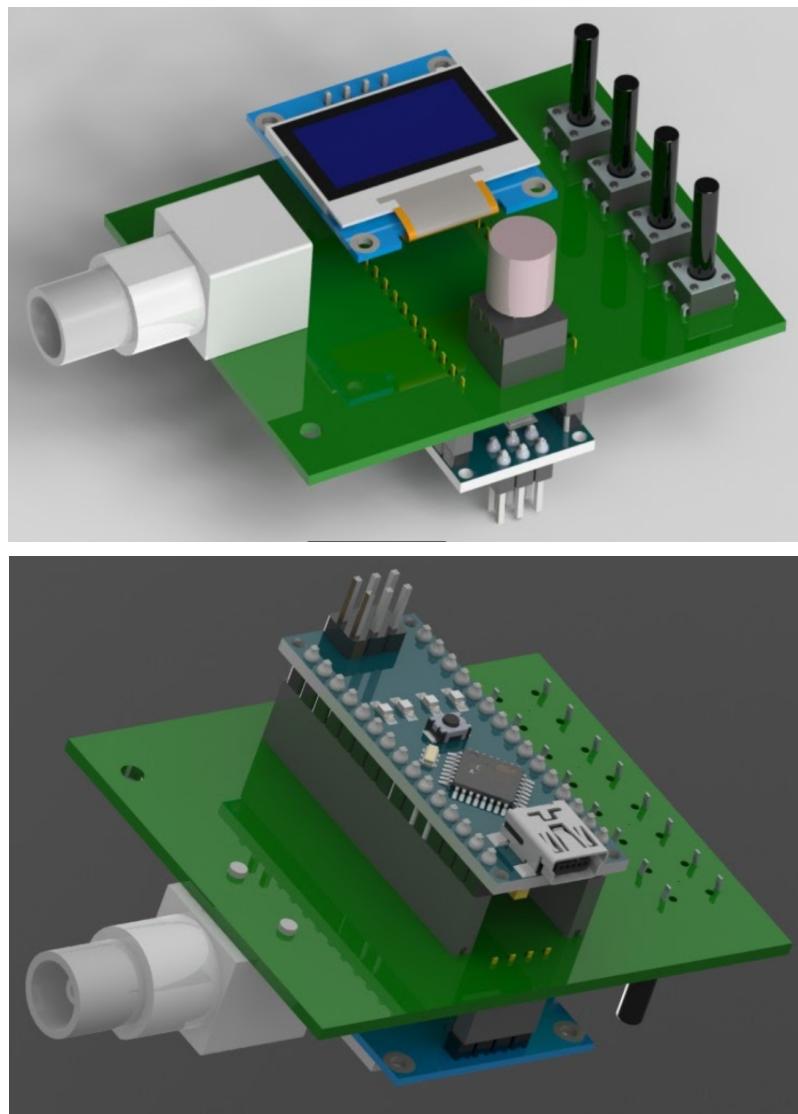


Figura 40 - Simulación en 3D de la placa implementada.

Ensamble - Renderizado

Como se mencionó anteriormente, también se realizó el diseño de un gabinete que satisfaga las necesidades para poder manipular el instrumento con una sola mano y así poder realizar los testeos necesarios o de interés.

La siguiente imagen corresponde al diseño y ensamble del osciloscopio de bolsillo.

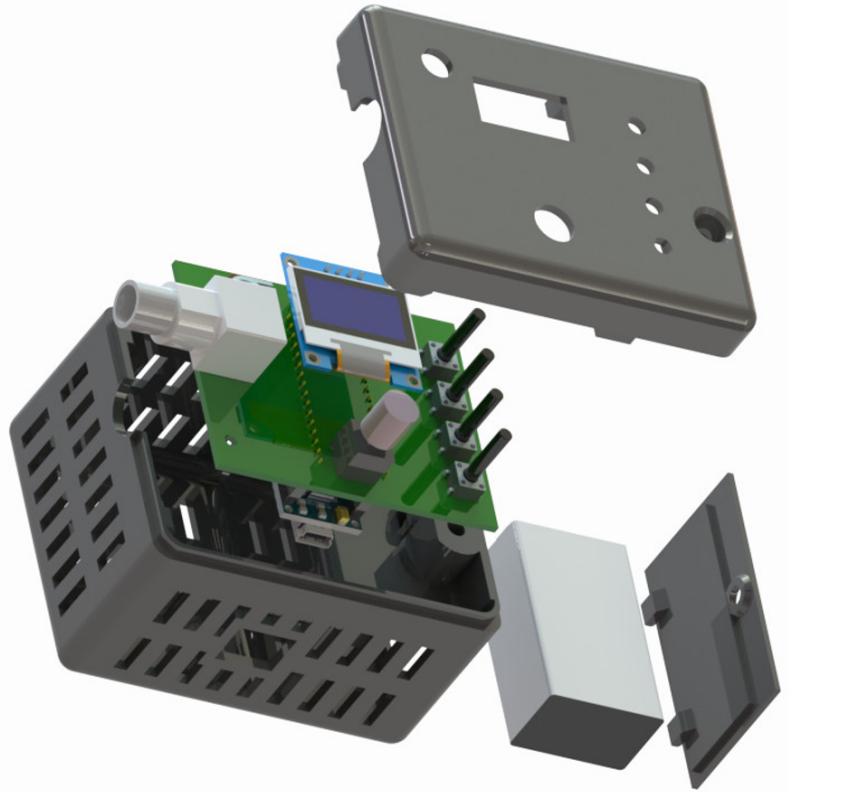


Figura 41 - (a1-a2)Simulación de gabinete y placa - (b)Implementación.

Descripción de los componentes

El osciloscopio de bolsillo desarrollado posee los siguientes componentes que hacen a la funcionalidad y ajuste de las señales visualizadas.

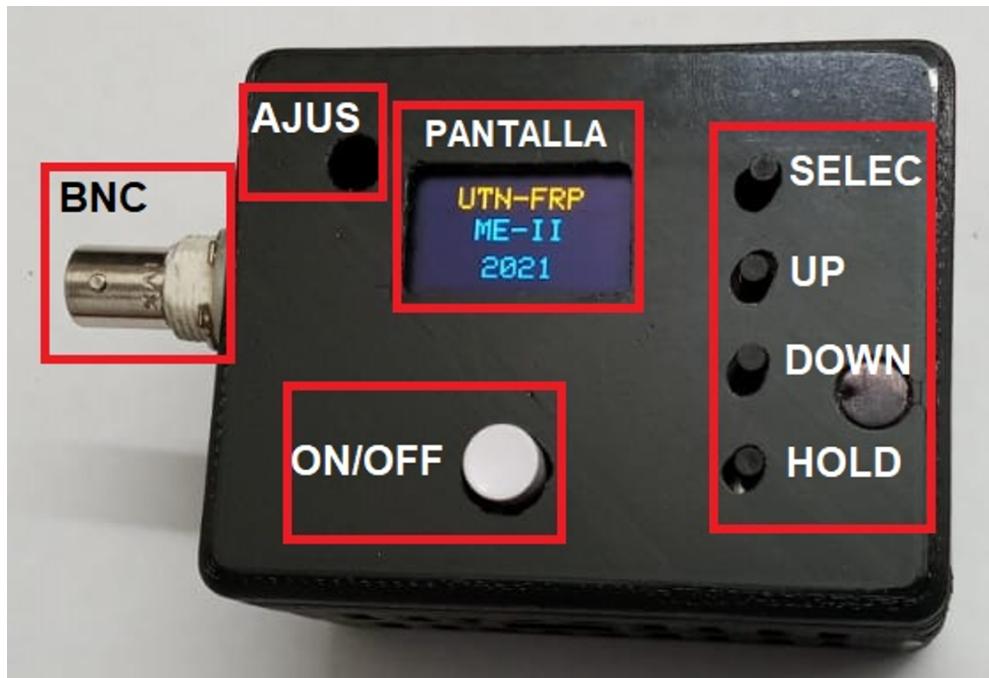


Figura 42 - Identificación de las partes y componentes.

OnOff: Este pulsador nos permite encender o apagar el instrumento.

Selec: Permite desplazar el cursor el cual nos determina la escala que podemos variar, estas son la escala de voltaje y de tiempo.

Up: Permite variar las escalas de forma ascendente correspondiente a la posición actual del cursor.

Down: Permite variar las escalas de forma descendente correspondiente a la posición actual del cursor.

Hold: Permite mantener estática una determinada muestra para su evaluación con mayor detalle.

BNC: Este conector permite el conexionado de puntas de prueba o sondas para realizar las mediciones.

Pantalla: Esta se encuentra dividida en dos sectores (colores). En amarillo se representan las escalas de tensión y de tiempo, también el valor medio de voltaje muestreado y la frecuencia, como así también la función Hold. En azul se representan, contenidas en una cuadrícula equivalente a las escalas de voltaje y de tiempo, las formas de ondas medidas.

Preset de Ajuste: Permite realizar un ajuste de los niveles de tensión en la entrada del circuito. Funciona como un divisor resistivo.

Calibración - Actualización

La posibilidad de utilizar un dispositivo, al cual lo podemos acceder mediante una conexión USB y con la utilización de un software dedicado, los parámetros y escalas son fácilmente configurados para así poder tener una mejor resolución y menor error posible en las mediciones. Sumado también a la posibilidad de agregar funciones y herramientas para el procesamiento de una señal dada.

Para la calibración se utilizó la señal de prueba que genera un osciloscopio digital, en este caso, el UWON SDS1052.

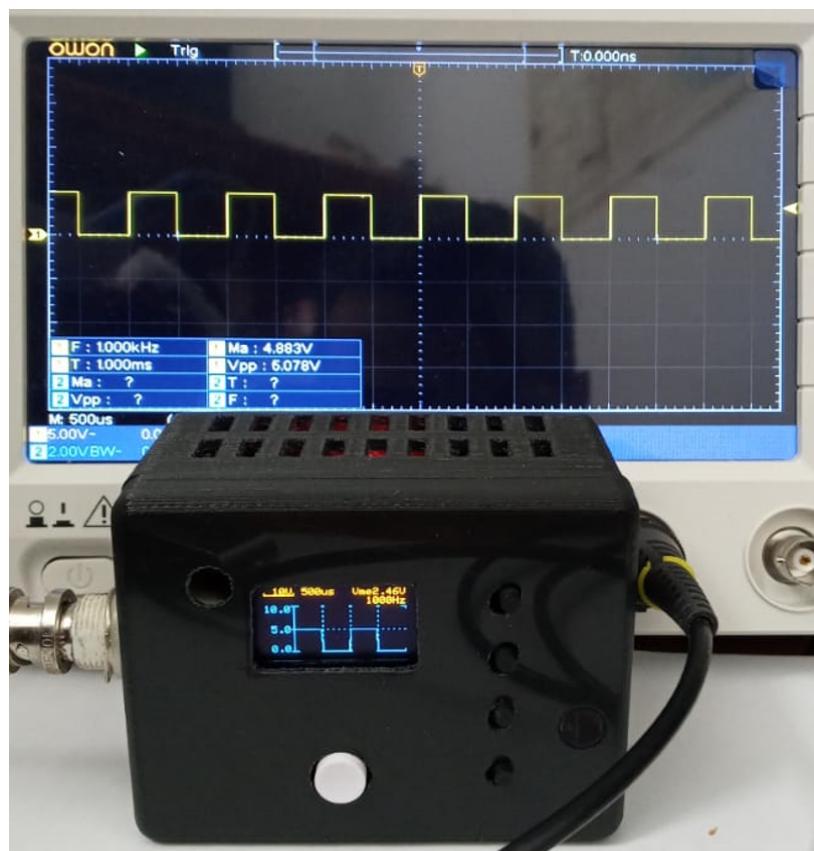
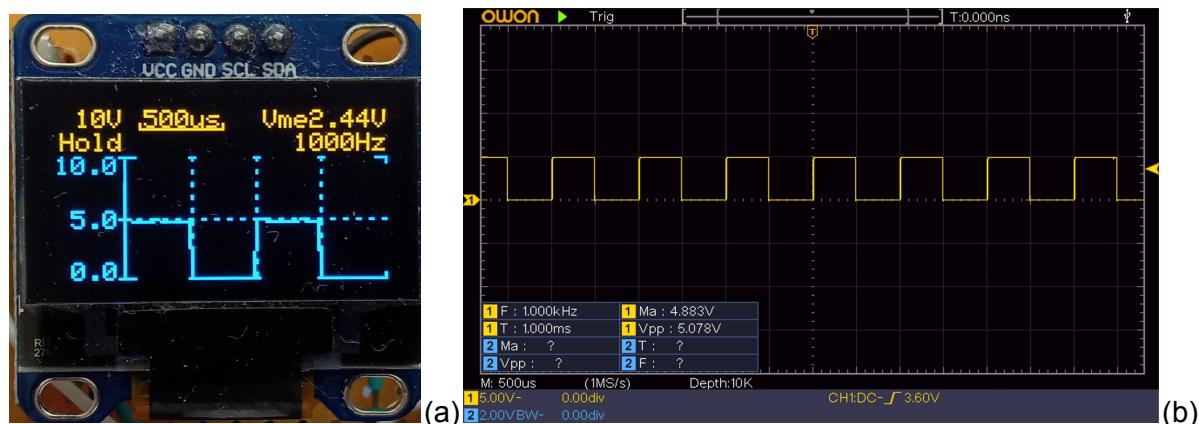


Figura 43 - Comparación de señales visualizadas-(a)Prototipo-(b)Osciloscopio comercial.

Punta de Prueba

Como el osciloscopio desarrollado tiene solo un canal, el conexionado con una punta de prueba se hizo a través de un conector BNC para poder adaptarle, de ser necesario, puntas comerciales estándar. En este caso, como el fin del instrumento desarrollado es para trabajo en campo y con un uso poco cuidadoso, se optó por armar una punta de prueba con un cable mallado RG179.

En la siguiente imagen se muestra la punta de prueba ensamblada.



Figura 44 - Punta de prueba implementada.

Ficha Técnica

La siguiente tabla representa las principales características del osciloscopio desarrollado.

Descripción	Valor Min - Max	Valor Típico
Batería	7 V - 15 V	9 V
Rango de Frecuencia	5 Hz - 20 KHz	1 KHz
Tensión de entrada	0 V (+) - 30 V (+)	5 V
Impedancia de entrada	220 KOhm - 1,2 MOhm	770 KOhm
Canales	1	
Pantalla	OLED 0,96"	
Factor de atenuación	1X	
Trigger	Central - Flanco ascendente	
Frecuencia de muestreo	125 Sa/s - 167 KSa/s	Rango de 10mseg - 2,5 KSa/s
Medición Automática	Voltaje medio de la señal y Frecuencia.	
Función Hold	SI	
Dimensiones	Ancho 80mm x Alto 65mm x Profundo 55mm	

Para el muestreo de señales con polarización negativa se debe colocar una fuente de alimentación continua (pila o batería) en serie con la punto de prueba ya que en la pantalla no se visualizan tensiones negativas.

Particularidades

- Las formas de ondas deben tener polaridad positiva
- Posibilidad de conexión a una computadora para actualización.
- Tensión de alimentación: 9V.
- Se pueden hacer medidas automáticamente (Vmedio y Frecuencia).
- Posibilidad de fijar una forma de onda para estudio (Hold).

Mejoras

Durante las pruebas y testeos surgieron observaciones para mejorar las prestaciones y funcionalidades del osciloscopio.

En futuras versiones se podría implementar la posibilidad de visualizar, además de la frecuencia de una señal, el porcentaje de Duty de la misma.

También, como las pruebas realizadas se hicieron sobre el sistema de encendido de un motor a combustión interna, que exista la posibilidad de visualizar las RPM del motor dependiendo la cantidad de cilindros y de tiempos que tenga el motor.

Otra mejora, que estaría dedicada exclusivamente al microcontrolador, sería que exista la posibilidad de una rutina que permita invernlar el instrumento para ahorro de energía.

Presupuesto de materiales

COMPONENTE	CANTIDAD	COSTO (\$ARS)
Pantalla OLED 0,96"	1	685
Placa Arduino NANO	1	875
Diodos 1N4148	4	20
Preset 1[MΩ]	1	95
Placa de pertinax	1	150
Resistencia 100[Ω]	1	0,50
Resistencia 120[kΩ]	1	0,50
Resistencia 12[kΩ]	1	0,50
Capacitor 100[nF]	1	2
Batería 9[V] (Eveready)	1	165
Interruptores tipo pulsador	5	125
Conector BNC hembra	1	88
Gabinete impreso	1	550
TOTAL		2756,5
cable mallado con punta atenuadora (x10) Hantek	1	1.235
TOTAL + Punta de Prueba		3991,5

OBSERVACIÓN: el costo de cada componente hace referencia a obtener cada uno por unidad (exceptuado resistencias, capacitores y pulsadores). Por cuestiones de tiempo no hemos podido consultar con proveedores para la obtención en cantidad de los mismos, lo cual suponemos, reduciría drásticamente el precio final. Éste precio final se basa en lo que costaría la implementación propia del dispositivo. También considerar que es solo precio de materiales, sin contemplar mano de obra y desarrollo del osciloscopio.

Osciloscopio Digital

Medidas Electrónicas II – Facultad Regional Paraná – Universidad Tecnológica Nacional



Comparativas comerciales

En la siguiente tabla se indican 3 modelos comerciales de características similares al instrumento desarrollado.

<p>Precio: \$10899 Número de canales: 1 Ancho de banda analógico: 0 - 200KHz Sensibilidad: 5mV / Div - 20V/Div Error de sensibilidad: <5% Resolución: 12 bits Impedancia de entrada: 1M ohmio Tensión de entrada máxima: 50 Vpk (500V pico a pico con punta atenuadora 10x) Acoplamiento: DC, AC, GND Frecuencia máxima de muestreo en tiempo real: 1Msps Base de tiempo: 10us/Div - 500s/Div Longitud de registro: 1024 Modos de disparo: Auto, Normal, Individual Tipos de disparo: Borde ascendente / descendente Posición de disparo: 1/2 del tamaño del búfer fijado Pantalla TFT LCD color de 2,4 pulgadas con resolución de 320 x 240 Idioma de la interface: Ingles Voltaje de alimentación: 9V DC (Atención: Máximo soportado 10V) Consumo: 120mA @ 9V Dimensión: 115mm x 75mm x 22mm Peso: 100 gramos</p>	<p>Precio: \$15030 Ancho de banda de frecuencia analógica: 0-2 MHz Velocidad máxima de muestreo en tiempo Real: 5 Msps Sensibilidad Vertical: 2mV/Div-20V/Div Total 12 grados (a modo de 1-2-5 progresivo) Rango de Tiempo Base Horizontal: 1μs/Div-50s/Div 21 grados (a modo de progresivo 1-2-5) Voltaje máximo de entrada: 50Vpk (100Vpp) Longitud máxima del registro: 512/1024 opcional Resistencia de entrada: 1 M ohm Precisión de ADC: 8 bits Método de acoplamiento: CC/CA Tipo de disparador: automático, Normal, sencillo Borde del disparador: borde que sube/baja Fuente del disparador: interno/externo Voltaje de disparo externo: 0-15 V Cursos de tiempo, pueden calcular y mostrar automáticamente la diferencia de tiempo Los cursos de voltaje pueden calcular y mostrar automáticamente la diferencia de voltaje Pantalla: Pantalla TFT Color Fuente de alimentación: Batería Li-ion de 3,7 V 1200 mAh (incluida)/interfaz USB Dimensión: 68*78*18mm/2,7*3,1 * 0.7in</p>	<p>Precio: \$6800 Máxima frecuencia de Muestreo en tiempo Real: 1 Msps Precisión: 8 Bits Tope de Profundidad de muestreo: 1024 bytes Ancho De Banda analógica: 0-200 KHz Sensibilidad Vertical: 10mV/Div-5 V/Div (1-2-5 de forma progresiva) Impedancia de entrada: 1MOhm Tensión máxima De Entrada: 50Vpp (con la punta 1: 1), 400Vpp (con la punta 10: 1) Modos de acoplamiento: DC/AC/GND Intervalo de Tiempo de Base Horizontal: 10μs/Div-50 s/Div (1-2-5 de forma progresiva) Tensión de alimentación: DC 9 V Tamaño del Circuito Impreso: 117 x 76mm Tamaño de la pantalla: 52 x 40mm Tamaño del paquete: 18 x 12.5 x 3 cm Peso del paquete: 120 gramos</p>

Precios: Octubre del 2021

Conclusión

Una vez llegado a este punto hemos podido entender en profundidad el osciloscopio como instrumento y conocer cuales son las partes que lo constituyen. En este informe final se hizo una reseña global de los sistemas que comprende el osciloscopio digital, se comentó en detalle aquellos bloques que son indispensables para un muestreo, proceso y representación de las formas de ondas. También se hicieron algunas referencias básicas del principio de funcionamiento de un microcontrolador, ya que éste es parte central en la toma de datos y procesamientos de los mismos.

El desarrollo de este informe resultó enriquecedor para entender en mayor profundidad la metodología de procesar señales y el gran espectro de usos que se le puede dar a un osciloscopio. Más aún, el estudio para la confección del mismo, nos permitió adquirir conocimientos más amplios de sus aplicaciones. En cuanto a la pantalla OLED, este tipo de tecnología nos permite tener una herramienta dinámica y flexible para lograr la mejor solución a lo que demanda un análisis de circuitos electrónicos, generando grandes ventajas sobre la portabilidad, comparándolas con los instrumentos de laboratorio.

También, el ejercicio de pensar en el diseño de nuestro instrumento, fue muy bueno porque nos generó un ambiente de discusión en torno a la mejor distribución de los controles y conexiones en comparación con los existentes en el mercado. Considerando además, que a su vez buscamos un instrumento de fácil uso y de rápida lectura de los datos representados en pantalla.

Bibliografía

El código del osciloscopio digital implementado está basado en el desarrollo de una propuesta de open-source de la comunidad Arduino, la cual se tomó como base y a la que se le realizaron los ajustes necesarios para adaptar el instrumento a nuestras necesidades.

Referencia web: <http://radiopench.blog96.fc2.com/blog-entry-893.html>