



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

**Grado en Ingeniería en Electrónica Industrial y
Automática**

**Diseño y construcción de una fuente de
corriente AC regulable para ensayo de
sondas de corriente**

Autor:

Martín Martín, Olga

Tutor:

**Domínguez, José Antonio
Tecnología electrónica**

Valladolid, Abril 2023.

Resumen

En el presente trabajo se realiza el diseño de un medidor de tensión y corriente alterna para ser utilizado en una fuente de corriente alterna de 50A, mostrando los resultados en un display, tanto de forma gráfica como numérica

Para ello se ha empleado el entorno de desarrollo integrado de Arduino (IDE) adaptando los drivers para funcionar con el microprocesador ESP-32. La visualización de la información se hará mediante una pantalla TFT que mostrará la forma de las ondas y el valor eficaz numérico.

Se ha comprobado todo el funcionamiento en un prototipo sobre una placa tipo proto-board.

Palabras clave: sonda Hall, corriente, tensión, ESP-32, pantalla TFT

Abstract

In the present work we are going to design an alternating voltage and alternating current measure instrument to be used in a 50A current source, while showing the results in a display, graphically and numerically.

To do so, Arduino's IDE is used, adapting the drivers to function with the microprocessor ESP-32. The information will be shown using a TFT screen that will display the shape of the wave and the numerical value of the effective value.

A proto-board has been used to test the device's functioning.

Key words: Hall sensor, current, voltage, ESP-32, TFT screen

Índice

Resumen	3
Abstract	5
Índice	7
Listado de figuras	9
Glosario	11
Nomenclatura utilizada.....	13
Capítulo 1: Introducción y objetivos	15
Capítulo 2: Descripción del equipo por bloques	17
Capítulo 3: Adquisición de medidas	19
3.1 Medición de la corriente.....	19
3.2 Datos de la sonda hall escogida.....	19
3.3 Elementos auxiliares a la sonda hall	20
3.4 Cálculos del filtro a la salida de la sonda hall	20
3.5 Simulación del filtro a la salida de la sonda hall	25
3.6 Circuito sumador a la salida del filtro de la sonda hall	26
3.7 Medición de la tensión	30
3.8 Análisis de medición directa de la tensión.....	30
3.9 Análisis de medición de la tensión a través de un amplificador operacional.	31
3.10 Elementos auxiliares a la medición de tensión	33
3.11 Elección del método de medida de la tensión.....	35
3.12 Simulación de la medición de tensión	36
Capítulo 4: Microcontrolador	37
4.1 Introducción.....	37
4.2 Elección del microprocesador.....	37
4.3 Descripción del funcionamiento del programa.....	38
4.4 Conexiones	41
Capítulo 5: Visualización de la información	43
5.1 Introducción.....	43
5.2 Análisis de un display 16x2	43

5.3 Análisis de la una pantalla TFT 480x320	44
5.4 Elección del método de visualización	45
5.5 Conexiones entre la pantalla y el microcontrolador	45
Capítulo 6: Alimentación de los componentes electrónicos	47
6.1 Introducción.....	47
6.2 Diseño de la fuente de alimentación	47
Capítulo 7: Prototipo construido y pruebas	49
Capítulo 8: Conclusiones.....	55
Bibliografía	57
Anexos	59
Código.....	59

Listado de figuras

Figura 1: Diagrama de bloques funcionales del equipo	17
Figura 2: Esquema eléctrico general	18
Figura 3: Esquema eléctrico de aplicación típica de ACS758 (sonda hall).....	19
Figura 4: Esquema eléctrico del filtro de dos etapas	21
Figura 5: Grafica de la respuesta en frecuencia del filtro de dos etapas.....	25
Figura 6: Esquema eléctrico del circuito sumador	26
Figura 7: Esquema eléctrico del divisor de tensión	31
Figura 8: Esquema eléctrico del medidor de tensión mediante un amplificador operacional	32
Figura 10: Esquema eléctrico del circuito sumador.....	34
Figura 9: Esquema eléctrico del divisor de tensión	34
Figura 11: Esquema eléctrico completo del medidor de tensión	36
Figura 12: Grafica de la medición de tensión	36
Figura 13: Arduino Nano Every.....	37
Figura 14: ESP32.....	38
Figura 15: Diagrama de bloques del proceso seguido por el programa	39
Figura 16: Pantalla LCD 16x2	43
Figura 17: Pines de conexión pantalla TFT 480x320.....	46
Figura 18: Esquema eléctrico de la fuente de alimentación.....	48
Figura 19: Fotografía primer diseño de la pantalla.....	50
Figura 20: Fotografía Primera grafica dibujada en la pantalla	51
Figura 21: Fotografía osciloscopio y pantalla	52
Figura 22: Fotografía osciloscopio	53
Figura 23: Fotografía pantalla.....	53

Glosario

E

Efecto Hall: fenómeno físico por el cual se separan las cargas en un conductor a través del cual circula una corriente eléctrica, efecto que genera un campo magnético.

ESP32: es un circuito integrado que incluye un microprocesador y módulos periféricos de entradas y salidas que permiten su conexión y comunicación.

I

I2C: protocolo de comunicación en serie con dos canales para realizar la comunicación, uno para la sincronización de los dispositivos que se comunican y otro para los datos.

L

LTSpice: Programa de simulación de circuitos electrónicos

P

PSIM: Programa de simulación de circuitos electrónicos

PWM: señales por modulación de ancho de pulso.

R

RMS: media cuadrática, empleada para calcular el valor eficaz de una señal dependiente del tiempo.

S

SPI: protocolo de comunicación, con 4 canales, uno de reloj (SCK) para sincronización, uno para la transmisión de información maestro-esclavo (MOSI),

otro para la transmisión de información esclavo-maestro (MISO) y un último canal empleado para seleccionar con que esclavo quiere comunicarse el maestro (CS).

U

UART: protocolo de comunicación en serie, con un único canal que transmite datos en sus bits individuales de forma secuencial.

Nomenclatura utilizada

V: Tensión

R: Resistencia

I_X: Corriente que atraviesa el componente X

P_X: Potencia que disipa la resistencia X

C: Condensador

f': Frecuencia de corte buscada para el filtro

ω_0 : Frecuencia natural inicial para el escalado en frecuencia del filtro

ω_0' : Frecuencia natural buscada para el filtro

α : Parámetro que relaciona la frecuencia natural buscada con la frecuencia natural inicial para el escalado en frecuencia del filtro

C': Valor teórico del condensador necesario para cumplir con la frecuencia de corte buscada en el filtro

β : Parámetro que relaciona el valor teórico del condensador con el valor real del condensador para el escalado en impedancia del filtro

R': Valor teórico de la resistencia necesaria para cumplir con la frecuencia de corte buscada en el filtro

f: Frecuencia de corte real del filtro

K: Ganancia de la etapa de un amplificador operacional

X_{RMS}: Valor eficaz de la señal dependiente en el tiempo x(t)

T: periodo de una señal

N: Numero de muestras que se toman de la señal en un periodo

t_m: Tiempo de muestreo

V_{ef}: Tensión eficaz

I_{ef}: Corriente eficaz

Capítulo 1: Introducción y objetivos

El objetivo final de este proyecto es desarrollar un aparato capaz de medir la tensión y corriente de una señal eléctrica. Las señales a medir por el aparato diseñado son la tensión de red de una frecuencia 50Hz y 230V eficaces, así como una corriente senoidal de 50A de valor eficaz. Ambas magnitudes se medirán en una fuente de corriente diseñada y usada para realizar pruebas de otros elementos de medida, concretamente unas sondas Rogowski. La fuente de corriente consiste en un conjunto de transformadores en serie de tal forma que la corriente sea elevada, pero la tensión muy reducida, de tal forma que las potencias tanto de entrada como de salida, sean casi nulas.

El proyecto se ha estructurado en distintos capítulos que tratan de resolver una de las funciones necesarias para completar el equipo de medida. En primer lugar, dividiremos el circuito en bloques funcionales, detallando la adquisición de las medidas. Después, se tratará el procesamiento de la información y su visualización para el usuario. Por último, nos centraremos en la fuente de alimentación encargada del suministro de los componentes electrónicos, con entrada de la tensión de red y salida en diversos voltajes bajos. En los anexos se incluirá el código del programa y los datasheets de los componentes electrónicos empleados.

Para el análisis de los circuitos se emplearán las herramientas de simulación Psim y LTSpice. Además, LTSpice se empleara para el análisis de pequeños circuitos analógicos y Psim para el análisis de mayores circuitos.

El análisis de funcionamiento y prestaciones de las sondas Rogowski sobre las que se harían las pruebas de medida con la fuente de corriente a diseñar han sido analizados por Velasco Caminero en su trabajo, Estudio de sondas de corriente Rogowski para su empleo en sistemas de monitorización de eficiencia energética [1].

Capítulo 2: Descripción del equipo por bloques

Para realizar el diseño del equipo hemos dividido las funciones que son necesarias para su funcionamiento en una serie de bloques. Esto nos permite dividir el problema y abordar cada parte de forma independiente.

Las etapas creadas son: una etapa de fuente de alimentación, la medición de corriente, la medición de tensión, el procesamiento de la información y la visualización de las señales.

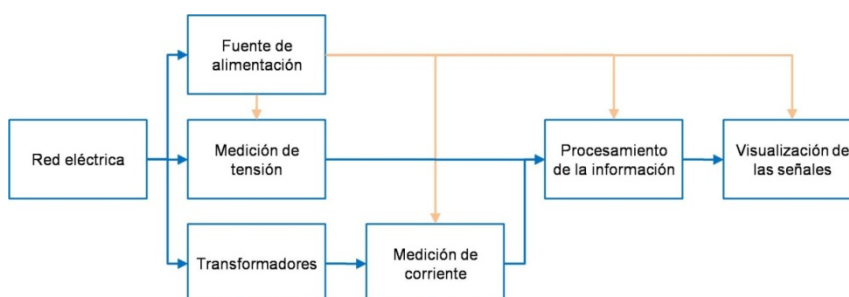


Figura 1: Diagrama de bloques funcionales del equipo

- La etapa de la fuente de alimentación hablaremos sobre cómo está será la encargada de transformar los 230V eficaces de la corriente alterna de cualquier enchufe de la red eléctrica en las tensiones continuas necesarias para alimentar el resto de componentes electrónicos del circuito.
- La etapa de la medición de la tensión será la parte del circuito encargada de la medición en tiempo real del valor instantáneo y eficaz de la tensión de red, cuyo valor nominal será normalmente 220V, 50Hz.
- La etapa de transformadores está constituida por tres transformadores en serie que, tomando la corriente eléctrica de un enchufe, la convertirán en una señal alterna de 50A de valor eficaz que se leerá en la etapa de medición de la corriente. Será necesario filtrar la señal tomada y adaptarla para que pueda llevarse a la etapa de procesamiento de la información.
- La etapa de procesamiento de la información estará compuesta de un microcontrolador que será capaz de leer las señales medidas para después procesarlas. Una vez hecho esto, enviará las señales adecuadas al elemento de visualización mostrando los datos necesarios.
- La etapa de visualización de las señales estará compuesta por una pantalla con la suficiente resolución como para poder mostrar la información de las medidas.

Esquema eléctrico general.

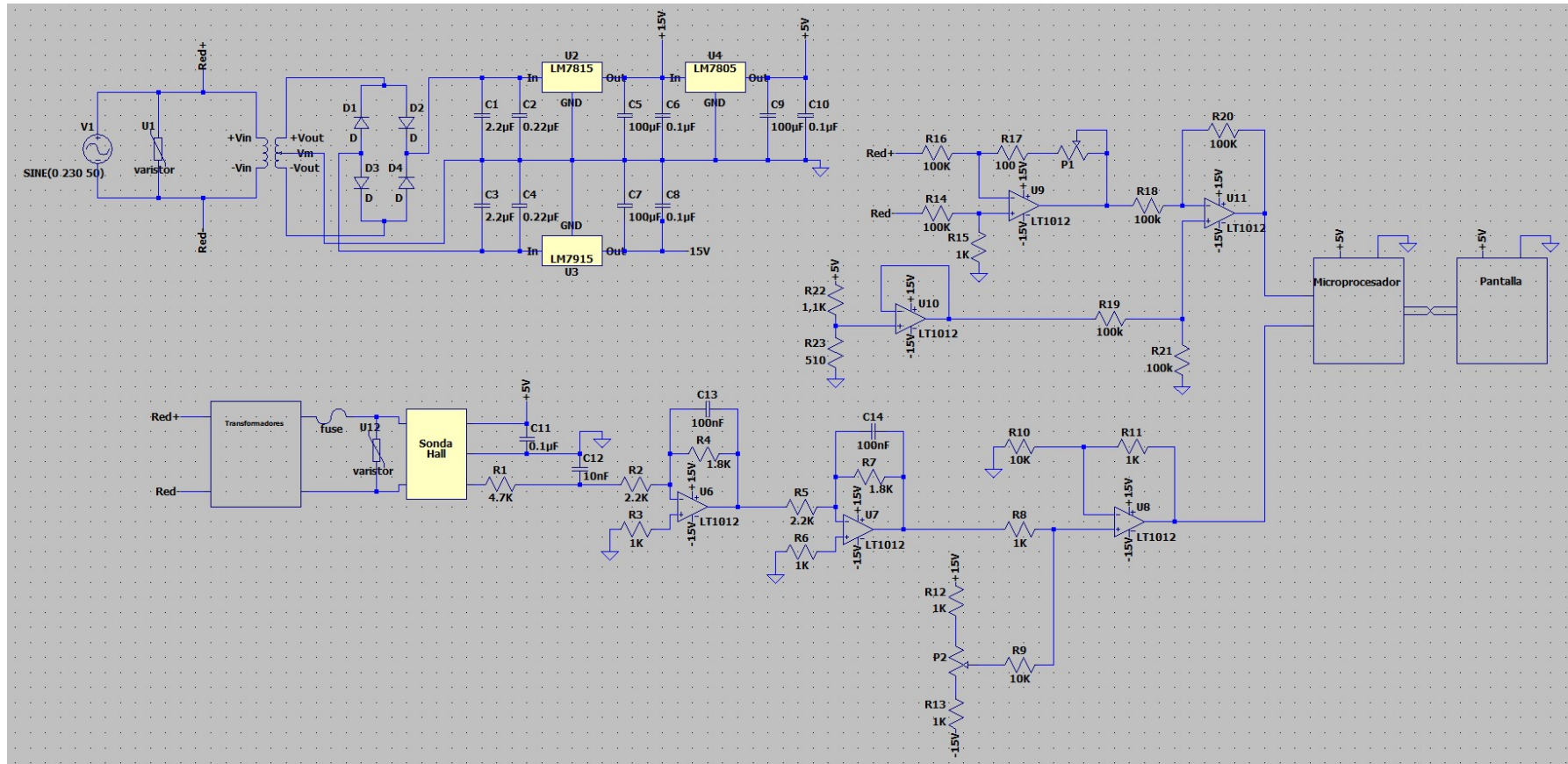


Figura 2: Esquema eléctrico general

Capítulo 3: Adquisición de medidas

3.1 Medición de la corriente

Se denomina efecto Hall al fenómeno físico de separación de las cargas en un conductor a través del cual circula una corriente eléctrica, efecto que genera un campo magnético.

Una sonda de efecto Hall es un instrumento de medida que se basa en la medición de los campos magnéticos generados por la circulación de corriente a través de un conductor. Esto nos permite calcular los amperios que atraviesan el material. [2]

Dado que la corriente alterna que vamos a medir es aproximadamente de 50A de valor eficaz, lo que supone un valor de pico de 70,7A, necesitamos utilizar un sensor que tenga un rango de medida de $\pm 100A$.

3.2 Datos de la sonda hall escogida

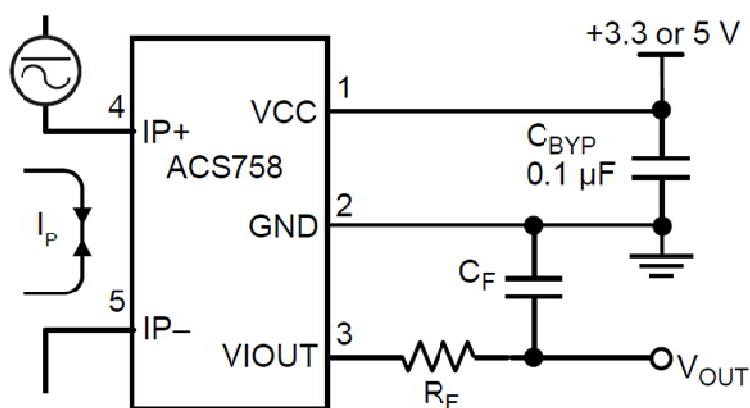


Figura 3: Esquema eléctrico de aplicación típica de ACS758 (sonda hall)

El sensor escogido es del fabricante Allegro, el ACS758LCB-100B-PFF-T que ya que es el único que ofrece con un rango de medida de $\pm 100A$. [3]

Para que esta sonda hall pueda funcionar necesita una tensión de alimentación entre 3V y 5V con una corriente de 10mA. En nuestro caso, elegiremos 5V como tensión de entrada (V_{cc}). Esta es la tensión recomendada por el fabricante, que ante una entrada nula la salida del sensor es de valor $V_{cc}/2$.

Este sensor tiene una sensibilidad de 20mV/A, lo que para un rango de medidas de 200A nos da una variación de 4V entre los valores extremos del rango, es

decir, entre la lectura de -100A y $+100\text{A}$. Lo que significa que la tensión de salida del sensor variara entre $\frac{V_{cc}}{2} + 2$ y $\frac{V_{cc}}{2} - 2$ voltios.

Como la tensión de entrada escogida es la recomendada en el datasheet del componente de 5V , tendríamos una salida que varía entre $4,5\text{V}$ y $0,5\text{V}$. Es un rango de salidas aceptable para trabajar con algunos microprocesadores como un Arduino, pero en el caso de otros microprocesadores podría causar problemas, como ocurre en el ESP32, que se analizará más adelante.

3.3 Elementos auxiliares a la sonda hall

El fabricante en su datasheet recomienda colocar un condensador entre las patillas de alimentación de $0,1\mu\text{F}$ (C_{11}), así como una resistencia llamada R_F , según la nomenclatura del fabricante, en la salida de mínimo $4,7\text{K}\Omega$ (R_1) y un condensador conectado entre la salida y la toma de tierra del componente de 10nF (C_{12}) como máximo. [3]

Por motivos de seguridad, añadiremos también un fusible en serie a la entrada de la sonda. Así, la podremos proteger en caso de que se produzca una sobretensión en la corriente a medir. Hay varias opciones en el mercado, pero se ha escogido el fusible $3\text{NC}2200\text{-}0\text{MK}$ de SIEMENS porque soporta la misma corriente máxima que es capaz de medir la sonda, hasta los 100A . [4]

Añadiremos también un varistor en paralelo a la entrada del circuito para proteger todos los componentes en caso de que se produzca una sobre tensión en la red eléctrica. Puesto que conocemos que el valor eficaz de la tensión que se va a medir es de 12V , sabemos que la tensión oscilara entre $\pm 16,97\text{V}$. Por lo que necesitamos un varistor cuya tensión nominal sea de 17V , el escogido en este caso sería $\text{VCACo}60317\text{X}330\text{MDP}$ de AVX. [5]

3.4 Cálculos del filtro a la salida de la sonda hall

A la salida de la sonda colocaremos un filtro, para eliminar ruidos y algunos armónicos en la señal medida. Además, aprovecharemos el filtro para adaptar la amplitud de la señal medida a la amplitud que puede leer el microprocesador. [6] Puesto que la señal medida tiene una amplitud máxima de 4V y la amplitud máxima que le puede llegar al microprocesador es de 3V necesitaremos que la ganancia de este filtro sea de $3/4$.

El filtro diseñado constará de dos etapas, cada una de ellas formada por un filtro activo de primer orden, por lo que la ganancia total que buscamos será el

producto de la ganancia de ambas etapas. Nuestra intención será utilizar dos etapas iguales para simplificar la construcción circuito, por lo que la ganancia que necesitamos obtener en cada etapa es $K_1 = K_2 = -\sqrt{3/4} \approx -0,866$.

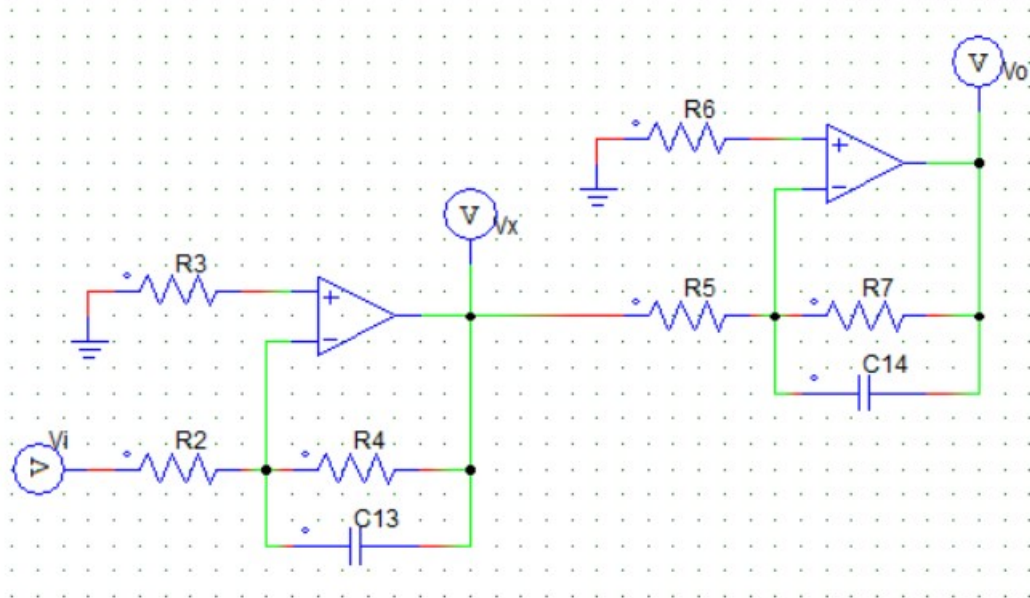


Figura 4: Esquema eléctrico del filtro de dos etapas

Etapas 1

$$\frac{V_i}{R_2} = \frac{-V_x}{R_4} + \frac{-V_x}{1/C_{13}S}$$

$$\frac{V_x}{V_i} = -\frac{R_4/R_2}{1 + R_4C_{13}S}$$

$$\omega_{0_1} = \frac{1}{R_4C_{13}} K_1 = \frac{-R_4}{R_2}$$

Para hacer los escalados suponemos como valores iniciales del prototipo $\omega_0 = 1$, $R_4 = 1\Omega$ y $C_{13} = 1F$

Escalado en frecuencia:

Definimos la frecuencia de corte para esta primera etapa en 900Hz.

$$\omega_{0_1} = 2\pi f = 2\pi \cdot 900 = \alpha\omega_0$$

$$\alpha = \frac{1800\pi}{1} = 1800\pi$$

$$C_0 = \frac{C_{13}}{\alpha} = \frac{1}{1800\pi} = 177\mu F$$

Escalado en impedancia:

Imponemos el condensador C_{13} de $100nF$

$$\beta = \frac{C_0}{C_{13}} = \frac{177 \cdot 10^{-6}}{100 \cdot 10^{-9}} = 1768,4$$

$$R_4 = \beta R_4 = 1768,4 \cdot 1 = 1768,4\Omega$$

Por lo que tomamos $C_{13} = 100nF$ y $R_4 = 1,8K\Omega$

La frecuencia real de corte del filtro es:

$$f_1 = \frac{\omega_1}{2\pi} = \frac{1/R_4 C_{13}}{2\pi} = \frac{1}{2\pi \cdot 100 \cdot 10^{-9} \cdot 1,8 \cdot 10^3} = 884Hz$$

La ganancia que buscamos encada etapa es de $-\sqrt{3/4} \approx -0,866$

$$K_1 = \frac{-R_4}{R_2} = \frac{-1,8 \cdot 10^3}{R_2} = -\sqrt{3/4} \Rightarrow R_2 = 2078,5\Omega$$

Por lo que el valor de R_2 escogido será $2,2K\Omega$.

El valor de R_3 se determina como el paralelo de las resistencias R_2 y R_4

$$R_3 = R_2 \parallel R_4 = 990\Omega \Rightarrow R_3 = 1K\Omega$$

Etapas 2

$$\frac{V_x}{R_5} = \frac{-V_0}{R_7} + \frac{-V_0}{1/C_{14}s}$$

$$\frac{V_0}{V_x} = -\frac{R_7/R_5}{1 + R_7 C_{14}s}$$

$$\omega_{02} = \frac{1}{R_7 C_{14}} K_2 = \frac{-R_7}{R_5}$$

Para hacer los escalados suponemos como valores iniciales del prototipo $\omega_0 = 1$, $R_7 = 1\Omega$ y $C_{14} = 1F$

Escalado en frecuencia:

La frecuencia de corte de esta etapa es la misma que la de la etapa anterior, es decir, 900Hz.

$$\omega_{0_2} = 2\pi f = 2\pi \cdot 900 = \alpha\omega_0$$

$$\alpha = \frac{1800\pi}{1} = 1800\pi$$

$$C_0 = \frac{C_{14}}{\alpha} = \frac{1}{1800\pi} = 177\mu F$$

Escalado en impedancia:

Imponemos un condensador C_{14} de 100nF

$$\beta = \frac{C_0}{C_{14}} = \frac{177 \cdot 10^{-6}}{100 \cdot 10^{-9}} = 1768,4$$

$$R_7 = \beta R_7 = 1768,4 \cdot 1 = 1768,4\Omega$$

Por lo que tomamos $C_{14} = 100nF$ y $R_7 = 1,8K\Omega$

La frecuencia real de corte del filtro es:

$$f_2 = \frac{\omega_2}{2\pi} = \frac{1/R_7 C_{14}}{2\pi} = \frac{1}{2\pi \cdot 100 \cdot 10^{-9} \cdot 1,8 \cdot 10^3} = 884Hz$$

La ganancia de la etapa necesitamos que sea la misma que la de la etapa anterior.

$$K_2 = \frac{-R_7}{R_5} = \frac{-1,8}{R_5} = -\sqrt{3/4} \Rightarrow R_5 = 2078,5\Omega$$

Por lo que el valor de R_5 escogido será $2,2K\Omega$.

El valor de R_6 se determina como el paralelo de las resistencias R_5 y R_7

$$R_6 = R_5 \parallel R_7 = 990\Omega \Rightarrow R_6 = 1K\Omega$$

La mejor opción para implementar este filtro sería utilizar un solo circuito integrado que contuviese al menos dos amplificadores operacionales. Uno de los que podría usarse es el LM358 de Texas Instruments. [7]

Para poder dimensionar los condensadores del filtro, necesitamos conocer la diferencia de tensiones máxima entre sus extremos. Para realizar los cálculos suponemos que la tensión de entrada al filtro es la máxima, es decir 4,5V.

El condensador C_{13} está sometido a la tensión V_x , mientras que el condensador C_{14} está sometido a la tensión V_0 . Ambos tienen una de las patillas conectada a la entrada negativa del amplificador operacional correspondiente y en este caso concreto la tensión de las entradas del amplificador operacional es cero. Para realizar estos cálculos suponemos que la tensión de entrada V_i es la máxima, es decir, 4,5V.

$$I_2 = \frac{V_i - 0}{R_2} = \frac{4,5}{2,2 \cdot 10^3} = 2 \text{ mA}$$

$$I_3 = I_2 = \frac{0 - V_x}{R_3}$$

$$V_x = -I_2 \cdot R_3 = -2 \cdot 10^{-3} \cdot 1,8 \cdot 10^3 = -3,68V$$

$$I_5 = \frac{V_x - 0}{R_5} = \frac{-3,68}{2,2 \cdot 10^3} = -1,67 \text{ mA}$$

$$I_7 = I_5 = \frac{0 - V_0}{R_7}$$

$$V_0 = -I_5 \cdot 5 = -(-1,67 \cdot 10^{-3}) \cdot 1,8 \cdot 10^3 = 3V$$

Será suficiente con que los condensadores elegidos para esta aplicación soporten al menos $\pm 5V$ entre sus extremos.

También necesitamos conocer la potencia que disipará cada resistencia, para lo cual primero debemos conocer el valor de la corriente que las atraviesa.

Podemos considerar que la corriente que pasa por las resistencias R_3 y R_6 es prácticamente o ya que están colocadas entre la toma de tierra y la entrada positiva del amplificador operacional que en este caso es cero. Por este motivo, será suficiente con que esas resistencias soporten 1/8W.

$$I_2 = \frac{V_i}{R_2} = \frac{4,5}{2,2 \cdot 10^3} = 2,05 \text{ mA}$$

$$P_{R_2} = R_2 \cdot I_2^2 = 2,2 \cdot 10^3 \cdot (2,05 \cdot 10^{-3})^2 = 9,25 \text{ mW}$$

$$I_4 = \frac{-V_x}{R_4} = \frac{3,378}{1,8 \cdot 10^3} = 1,877mA$$

$$P_{R_4} = R_4 \cdot I_4^2 = 1,8 \cdot 10^3 \cdot (1,877 \cdot 10^{-3})^2 = 6,34mW$$

$$I_5 = \frac{V_x}{R_5} = \frac{-3,378}{2,2 \cdot 10^3} = -1,535mA$$

$$P_{R_5} = R_5 \cdot I_5^2 = 2,2 \cdot 10^3 \cdot (-1,535 \cdot 10^{-3})^2 = 5,18mW$$

$$I_7 = \frac{-V_o}{R_7} = \frac{2,54}{1,8 \cdot 10^3} = -1,41mA$$

$$P_{R_7} = R_7 \cdot I_7^2 = 1,8 \cdot 10^3 \cdot (-1,41 \cdot 10^{-3})^2 = 3,58mW$$

Atendiendo a estos cálculos, podemos concluir que todas las resistencias del filtro pueden ser de 1/8W de potencia.

3.5 Simulación del filtro a la salida de la sonda hall

En la siguiente gráfica tenemos representada la respuesta en frecuencia del filtro de dos etapas anteriormente calculado. En verde se muestra la respuesta de una señal que no elimina ninguna frecuencia, por lo que se trata de la señal de entrada al filtro. En azul podemos ver la respuesta del filtro tras pasar solo por la primera etapa de filtrado. Por último, en rojo tenemos la respuesta a la salida del filtro, es decir, después de que la señal haya pasado por las dos etapas.

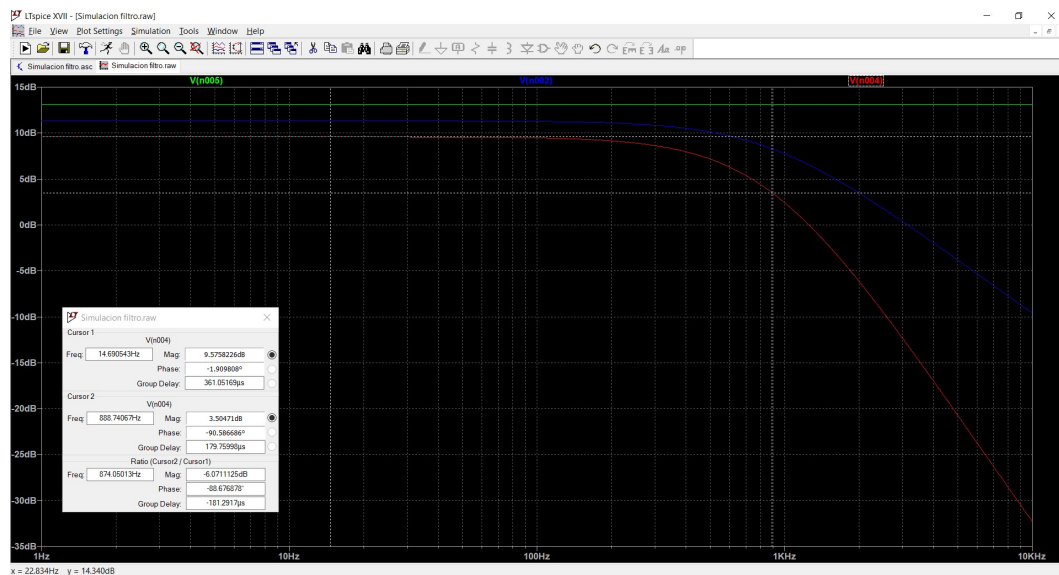


Figura 5: Grafica de la respuesta en frecuencia del filtro de dos etapas

El valor de la magnitud disminuye tras cada etapa del filtro, ya que la ganancia de este es menor que la unidad. Al tratarse de un filtro con dos etapas de primer orden en cadena, la frecuencia de corte se produce cuando la magnitud ha disminuido en 6dB, es decir, la magnitud ha disminuido 3dB por cada etapa.

La simulación indica una caída de aproximadamente 6dB a la frecuencia de 884Hz. Esta es un poco mayor al valor calculador teóricamente, ya que, al tratarse de dos filtros de primer orden en serie, se acorta la frecuencia efectiva.

3.6 Circuito sumador a la salida del filtro de la sonda hall

A la salida del filtro colocamos un circuito sumador para añadir a la señal filtrada una pequeña señal continua con el fin de poder ajustar el cero en la visualización. [8] Este ajuste es necesario porque la resolución de la pantalla es menor que la precisión de la sonda hall.

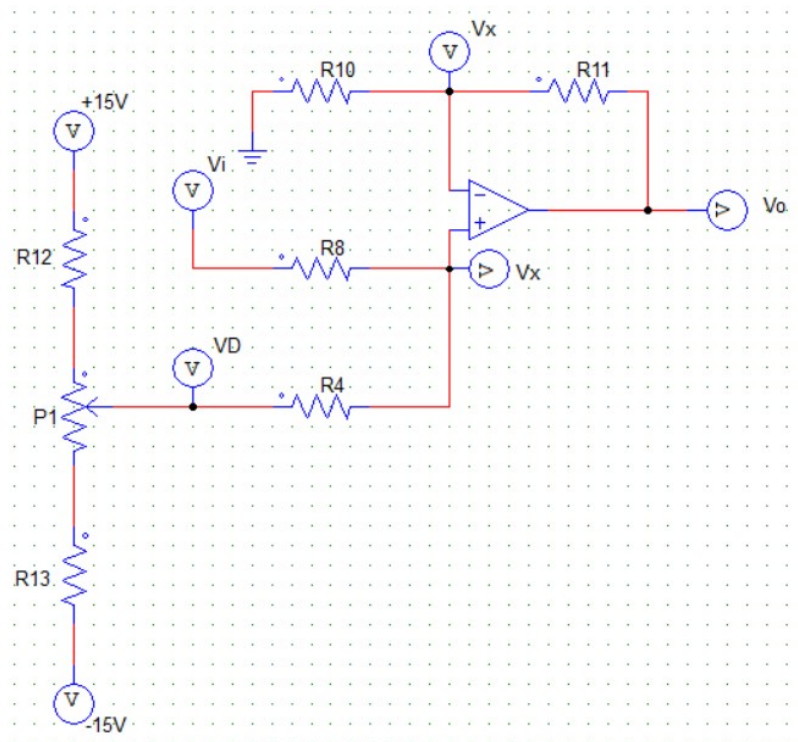


Figura 6: Esquema eléctrico del circuito sumador

$$\begin{cases} \frac{V_x - 0}{R_{10}} = \frac{V_o - V_x}{R_{12}} \\ \frac{V_i - V_x}{R_8} = \frac{V_x - V_D}{R_9} \end{cases}$$

$$V_o = \left(\frac{V_x}{R_{10}} + \frac{V_x}{R_{11}} \right) \cdot R_{11} = \frac{R_{11} \cdot V_x + R_{10} \cdot V_x}{R_{10} \cdot R_{11}} \cdot R_{11} = \frac{R_{10} + R_{11}}{R_{10}} \cdot V_x$$

$$\frac{-V_x}{R_8} - \frac{V_x}{R_9} = \frac{-V_D}{R_9} - \frac{V_i}{R_8}$$

$$\frac{R_9 \cdot V_x + R_8 \cdot V_x}{R_9 \cdot R_8} = \frac{V_D}{R_9} + \frac{V_i}{R_8}$$

$$V_x = \frac{R_9 \cdot R_8}{R_9 + R_8} \cdot \frac{R_9 \cdot V_i + R_8 \cdot V_D}{R_9 \cdot R_8} = \frac{R_9 \cdot V_i + R_8 \cdot V_D}{R_9 + R_8}$$

$$V_o = \frac{R_{10} + R_{11}}{R_{10}} \cdot \frac{R_9 \cdot V_i + R_8 \cdot V_D}{R_9 + R_8}$$

$$\text{Si } R_{10} \parallel R_{11} = R_9 \parallel R_8 \rightarrow \frac{R_{10} + R_{11}}{R_{10} \cdot R_{11}} = \frac{R_9 + R_8}{R_9 \cdot R_8} \rightarrow \frac{R_{10} + R_{11}}{(R_9 + R_8) R_{10}} = \frac{R_{11}}{R_9 \cdot R_8}$$

$$V_o = \frac{R_{11}}{R_9 \cdot R_8} (R_9 \cdot V_i + R_8 \cdot V_D) = \frac{R_{11}}{R_8} V_i + \frac{R_{11}}{R_9} V_D$$

La tensión V_i es la señal a medir, es decir, la salida del sensor tras pasar por el filtro, por lo que su ganancia debe de ser 1 ya que no queremos alterar la amplitud de la medida.

$$K_i = 1 = \frac{R_{11}}{R_8} \rightarrow R_{11} = R_8$$

La tensión V_D es una pequeña tensión continua variable, que súmanos a la señal medida para centrar la señal en la pantalla, por lo que nos interesa que la ganancia de esta señal sea pequeña, menor que la unidad, para ajustar lo mejor posible.

$$\frac{R_{11}}{R_9} = 0,1$$

$$R_{10} = R_9 = 10K\Omega$$

$$R_{11} = R_8 = 1K\Omega$$

Además debemos conocer los valores de las resistencias del divisor de tensión, necesario para obtener el valor de V_D .

$$I_{12} = I_{13} + I_D$$

$$\frac{15 - V_D}{R_{12} + x \cdot P_1} = \frac{V_D - (-15)}{R_{13} + (1 - x) \cdot P_1} + I_D$$

Si suponemos $I_D \approx 0$ y que $R_{12} = R_{13} = R$

$$\frac{15 - V_D}{R + x \cdot P_1} = \frac{V_D + 15}{R + (1 - x) \cdot P_1}$$

$$(15 - V_D) \cdot (P_1 - x \cdot P_1 + R) = (V_D + 15) \cdot (R + x \cdot P_1)$$

$$15P_1 - 15xP_1 + 15R - V_D P_1 + V_D x P_1 - V_D R = V_D R + V_D x P_1 + 15R + 15xP_1$$

$$15P_1 - 15xP_1 + 15R - 15R - 15xP_1 = V_D R + V_D x P_1 + V_D P_1 - V_D x P_1 + V_D R$$

$$15P_1 - 30xP_1 = V_D(2R + P_1)$$

$$V_D = \frac{(15 - 30x) \cdot P_1}{2R + P_1}$$

Si $x = 0 \rightarrow V_D = +5V$

$$5 = \frac{15P_1}{2R + P_1}$$

$$10R + 5P_1 = 15P_1$$

$$10R = 10P_1$$

$$R = P_1$$

Si $x = 1 \rightarrow V_D = -5V$

$$-5 = \frac{(15 - 30)P_1}{2R + P_1}$$

$$-10R - 5P_1 = -15P_1$$

$$-10R = -10P_1$$

$$R = P_1$$

Atendiendo a estos cálculos, podemos concluir que cualquier resistencia es válida para el divisor de tensión mientras ambas sean iguales entre sí e iguales al potenciómetro. Para que la corriente que atraviesa las resistencias sea pequeña, dichas resistencias, así como el potenciómetro, se escogerán con el valor de $1k\Omega$.

$$I = \frac{+15 - (-15)}{R_{12} + R_{13} + P_1} = \frac{30}{3 \cdot 10^3} = 0,01A$$

$$P_{P_1} = P_1 \cdot I^2 = 1000 \cdot 0,01^2 = 0,1W$$

De esta manera, ambas resistencias y el potenciómetro pueden ser todos de 1/8W de potencia.

Para dimensionar las resistencias del circuito sumador necesitamos conocer la potencia máxima que deben poder disipar y para ello debemos saber el valor de la corriente que las atraviesa. Antes de poder calcular dicho valor, debemos conocer el valor de la tensión entre sus extremos, que dependen de las variables anteriormente nombradas como V_o y V_x .

$$V_o = \frac{R_{11}}{R_8} V_i + \frac{R_{11}}{R_9} V_D = \frac{1 \cdot 10^3}{1 \cdot 10^3} V_i + \frac{1 \cdot 10^3}{10 \cdot 10^3} V_D = V_i + \frac{V_D}{10}$$

Sabemos que la tensión V_i varía entre 0 y 3V, ya que es la salida del filtro. La tensión V_D es la que proviene del divisor de tensión y varía entre 5V y -5V.

Si consideramos $V_i=0V$ y $V_D=-5V$ los valores que obtenemos de V_o y V_x son:

$$V_o = 0 + \frac{-5}{10} = -0,5V$$

$$V_x = \frac{R_{10}}{R_{10} + R_{11}} V_o = \frac{10 \cdot 10^3}{10 \cdot 10^3 + 1 \cdot 10^3} \cdot (-0,5) = -0,45V$$

Si consideramos $V_i=3V$ y $V_D=5V$ los valores que obtenemos de V_o y V_x son:

$$V_o = 3 + \frac{5}{10} = 3,5V$$

$$V_x = \frac{R_{10}}{R_{10} + R_{11}} V_o = \frac{10 \cdot 10^3}{10 \cdot 10^3 + 1 \cdot 10^3} \cdot (3,5) = 3,18V$$

Ahora que conocemos los valores máximos y mínimos de las tensiones entre los extremos de las resistencias, tenemos todos los datos necesarios para poder calcular las corrientes que las atraviesan.

$$I_{10} = \frac{V_x}{R_{10}} = \frac{3,18}{10 \cdot 10^3} = 0,318mA$$

$$P_{R_{10}} = R_{10} \cdot I_{10}^2 = 10 \cdot 10^3 \cdot (0,318 \cdot 10^{-3})^2 = 1mW$$

$$I_{11} = \frac{V_o - V_x}{R_{11}} = \frac{3,5 - 3,18}{1 \cdot 10^3} = 0,32mA$$

$$P_{R_{11}} = R_{11} \cdot I_{11}^2 = 1 \cdot 10^3 \cdot (0,32 \cdot 10^{-3})^2 = 0,1mW$$

$$I_9 = \frac{V_i - V_x}{R_9} = \frac{0 - (-0,45)}{10 \cdot 10^3} = -45\mu A$$

$$P_{R_9} = R_9 \cdot I_9^2 = 10 \cdot 10^3 \cdot (-45 \cdot 10^{-6})^2 = 20,25\mu W$$

$$I_8 = \frac{V_x - V_D}{R_8} = \frac{-0,45 - (-5)}{1 \cdot 10^3} = 4,55mA$$

$$P_{R_8} = R_8 \cdot I_8^2 = 1 \cdot 10^3 \cdot (4,55 \cdot 10^{-3})^2 = 20,7mW$$

Podemos concluir que sería suficiente con que todas las resistencias de este circuito sumador sean de 1/8W.

3.7 Medición de la tensión

Vamos a medir la tensión de entrada de los transformadores, es decir, la tensión de red a la que se conectará el primero de ellos. Esto es necesario porque la empresa que va a utilizar este medidor quiere combinar la medida de una alta corriente con una alta tensión para poder simular altas cargas de energía.

Podemos tomar esta medida de dos formas diferentes, las cuales analizaremos a continuación. En ambas soluciones estudiadas, la salida será una onda centrada en 0V, pero la parte negativa de esa señal no puede entrar al microprocesador. Por este motivo, a posteriori en ambos casos será necesario sumarle una tensión continua para que toda la señal resultante sea positiva a la entrada del microprocesador. Como la entrada al microprocesador no puede ser de más de 3V de amplitud necesitamos que, tras el proceso de medición, la señal, centrada en cero, tenga una amplitud de 1,5V y así, al sumarle una señal continua de 1,5V, obtendremos una señal que puede entrar al microprocesador.

3.8 Análisis de medición directa de la tensión

Podemos medir directamente la tensión de la red, pero, puesto que su valor eficaz es de 230V y la tensión de pico es de 325V, necesitaremos utilizar un divisor de tensión para que nuestro microprocesador pueda leerla.

El divisor de tensión estará formado por dos resistencias y un potenciómetro para poder hacer un ajuste final.

La resistencia RA será de $220\text{K}\Omega$, RB será de 510Ω y el potenciómetro será de 500Ω de resistencia máxima.

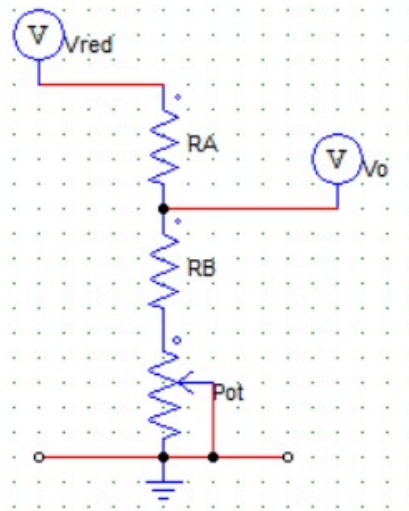


Figura 7: Esquema eléctrico del divisor de tensión

Las tensiones entre los extremos de las resistencias variarán con el valor del potenciómetro. Podemos considerar que, aproximadamente, deben caer 323V en RA, $0,75\text{V}$ en RB y $0,75$ en el potenciómetro.

Conociendo estos datos, podemos calcular la potencia que disipan. RA disipa como máximo $0,476\text{W}$ por lo que para el montaje real podemos escoger una resistencia de $1/2\text{W}$. Tanto el potenciómetro como RB disipan muy poca potencia porque en sus extremos cae una tensión muy baja, así que seleccionando unas resistencias que disipen $1/8\text{W}$ será suficiente.

3.9 Análisis de medición de la tensión a través de un amplificador operacional

Otro método que se puede utilizar para medir la tensión de red sería emplear un amplificador diferencial con realimentación negativa. [6]

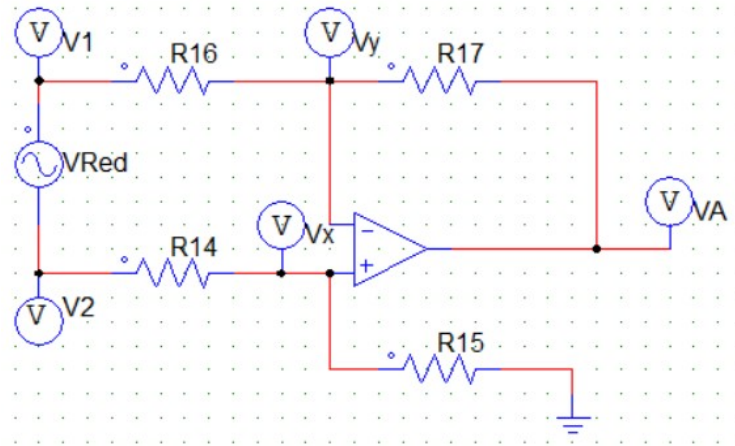


Figura 8: Esquema eléctrico del medidor de tensión mediante un amplificador operacional

$$\begin{cases} \frac{V_1 - V_x}{R_{16}} = \frac{V_x - V_A}{R_{17}} \\ \frac{V_2 - V_x}{R_{14}} = \frac{V_x - 0}{R_{15}} \end{cases}$$

$$\begin{aligned} \frac{V_2}{R_{14}} &= \frac{V_x}{R_{15}} + \frac{V_x}{R_{14}} = \frac{R_{15}V_x + R_{14}V_x}{R_{15}R_{14}} & \left| \right. & \frac{V_x}{R_{17}} - \frac{V_A}{R_{17}} = \frac{V_1}{R_{16}} - \frac{V_x}{R_{16}} \\ V_2R_{15} &= (R_{14} + R_{15})V_x & & \frac{V_A}{R_{17}} = \frac{R_{16}V_x + R_{17}V_x}{R_{16}R_{17}} - \frac{V_1}{R_{16}} \\ V_x &= \frac{R_{15}}{R_{14} + R_{15}} V_2 & & V_A = \frac{R_{16} + R_{17}}{R_{16}} V_x - \frac{R_{17}}{R_{16}} V_1 \\ & & & V_A = \frac{R_{16} + R_{17}}{R_{14} + R_{15}} \frac{R_{15}}{R_{16}} V_2 - \frac{R_{17}}{R_{16}} V_1 \end{aligned}$$

Si suponemos que $R_{15}=R_{17}$ y $R_{14}=R_{16}$

$$V_A = \frac{R_{15}}{R_{14}} (V_2 - V_1)$$

La tensión de salida V_A no puede ser superior a 1,5V. La tensión de entrada máxima a este circuito será la tensión máxima de pico de la tensión de red, es decir, 325V. Por lo que la ganancia para esta etapa será de $K=R_{15}/R_{14}$.

$$K = \frac{1,5}{325} = 0,004615$$

Los valores de las resistencias escogidas para R_{14} y R_{16} serán de $390K\Omega$. La resistencia R_{15} será de $1,8K\Omega$ y R_{17} de $1,5K\Omega$, con un potenciómetro P_2 de 330Ω para hacer un buen ajuste de la medida.

$$K_{real} = \frac{R_{15}}{R_{14}} = \frac{1800}{390000} = 0,004615$$

Para poder dimensionar correctamente las resistencias, necesitamos conocer la potencia que disipan y para averiguar esto, necesitamos conocer primero la corriente máxima que las atraviesa.

$$I_{16} = \frac{V_1 - V_x}{R_{16}} = \frac{V_1 - \frac{R_{15}}{R_{14}+R_{15}}V_2}{R_{14}} = \frac{V_1}{R_{14}} - \frac{R_{15}}{R_{14}(R_{14} + R_{15})}V_2 = 0,83mA$$

$$I_{15} = \frac{V_x - V_A}{R_{15}} = \frac{\frac{R_{15}}{R_{14}+R_{15}}V_2 - \frac{R_{15}}{R_{14}}(V_2 - V_1)}{R_{15}} = \frac{V_2}{R_{14} + R_{15}} - \frac{V_2 - V_1}{R_{14}} = 0,83mA$$

Puesto que las corrientes de entrada al amplificador operacional son despreciables podemos determinar que $I_{17} = I_{16}$ y $I_{14} = I_{15}$.

$$P_{R_{14}} = R_{14} \cdot I_{14}^2 = 390000 \cdot (0,83 \cdot 10^{-3})^2 = 270mW$$

$$P_{R_{15}} = R_{15} \cdot I_{15}^2 = 1800 \cdot (0,83 \cdot 10^{-3})^2 = 1,25mW$$

$$P_{R_{16}} = R_{16} \cdot I_{16}^2 = 390000 \cdot (0,83 \cdot 10^{-3})^2 = 270mW$$

$$P_{R_{17}} = R_{17} \cdot I_{17}^2 = 1500 \cdot (0,83 \cdot 10^{-3})^2 = 1mW$$

$$P_{P_2} = R_{P_2} \cdot I_{P_2}^2 = 330 \cdot (0,83 \cdot 10^{-3})^2 = 0,23mW$$

Las resistencias R_{14} y R_{16} tendrán que ser de $1/2W$ de potencia, pero el resto de las resistencias de este circuito, incluido el potenciómetro, podrán ser de $1/8W$.

3.10 Elementos auxiliares a la medición de tensión

Necesitaremos generar una señal continua de $1,5V$ para después sumarla con la señal medida y así obtener una señal entre $+3V$ y $0V$ que es el rango de entrada del microprocesador.

La señal continua la obtendremos mediante un divisor de tensión a partir de los 5V de continua que nos proporciona la fuente de alimentación.

$$V_B = \frac{R_{23}}{R_{22} + R_{23}} \cdot 5 = 1,5$$

$$R_{23} = 0,3(R_{22} + R_{23}) = 0,3 \cdot R_{22} + 0,3 \cdot R_{23}$$

$$R_{22} = \frac{1 - 0,3}{0,3} R_{23} = \frac{7}{3} R_{23}$$

Las resistencias pueden ser R_{22} de $1,1k\Omega$ y R_{23} de 510Ω .

$$V_B = \frac{510}{1100 + 510} \cdot 5 = 1,58V$$

La potencia que deben disipar estas resistencias será de $1/8W$ cada una de ellas.

$$P_{R_{22}} = \frac{V_B^2}{R_{22}} = \frac{1,58^2}{1100} = 2,27mW$$

$$P_{R_{23}} = \frac{(5 - V_B)^2}{R_{23}} = \frac{(5 - 1,58)^2}{510} = 22,9mW$$

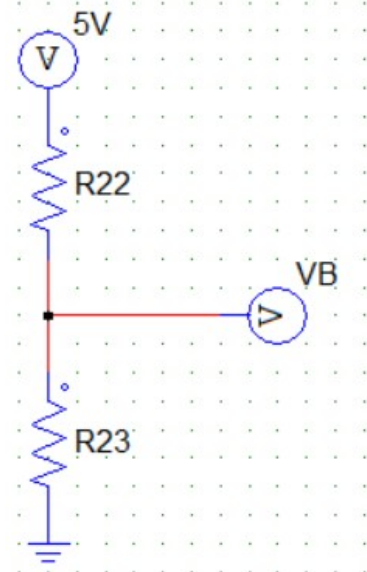


Figura 9: Esquema eléctrico del divisor de tensión

A continuación, utilizaremos un amplificador operacional para sumar las dos señales calculadas V_A y V_B y obtener una tensión de salida entre los 3V y los 0V que puede leer el microprocesador. [8]

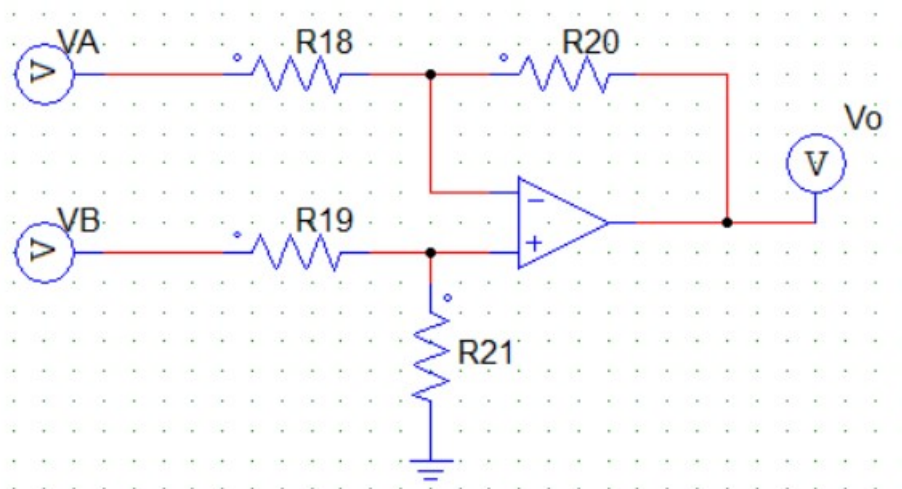


Figura 10: Esquema eléctrico del circuito sumador

Si suponemos que todas las resistencias son iguales a R_{18} tenemos que la tensión de salida es:

$$\begin{cases} \frac{V_A - V_x}{R_{18}} = \frac{V_x - V_o}{R_{18}} \\ \frac{V_B - V_x}{R_{18}} = \frac{V_x - 0}{R_{18}} \end{cases} \quad \begin{aligned} V_x &= \frac{V_B}{2} \\ V_o &= 2V_x - V_A \end{aligned}$$

$$V_o = V_B - V_A$$

La ganancia de este amplificador operacional será de 1, ya que no queremos modificar la amplitud de las señales que le llegan, solo sumarlas. Por lo que todas las resistencias R_{18} , R_{19} , R_{20} y R_{21} deben ser iguales. Los valores escogidos para estas resistencias serán de $100\text{k}\Omega$.

Para calcular las potencias que deben disipar estas resistencias, necesitamos conocer primero la corriente que las atraviesa.

$$I_{18} = I_{20} = \frac{V_A - V_x}{R_{18}} = \frac{-1,5 - 0,75}{100000} = -2,25 \cdot 10^{-2} \text{ mA}$$

$$I_{19} = I_{21} = \frac{V_B - V_x}{R_{18}} = \frac{1,5 - 0,75}{100000} = 7,5 \cdot 10^{-3} \text{ mA}$$

$$P_{R_{18}} = P_{R_{20}} = R_{18} \cdot I_{18}^2 = 5,06 \cdot 10^{-5} \text{ W}$$

$$P_{R_{19}} = P_{R_{21}} = R_{19} \cdot I_{19}^2 = 5,625 \cdot 10^{-6} \text{ W}$$

Por lo tanto, será suficiente con que las resistencias disipen $1/8\text{W}$ cada una de ellas.

3.11 Elección del método de medida de la tensión

Aunque ambas soluciones producen medidas igual de fiables, vamos a optar por la medición con el amplificador operacional ya que ofrece un aislamiento entre la red y nuestro circuito.

3.12 Simulación de la medición de tensión

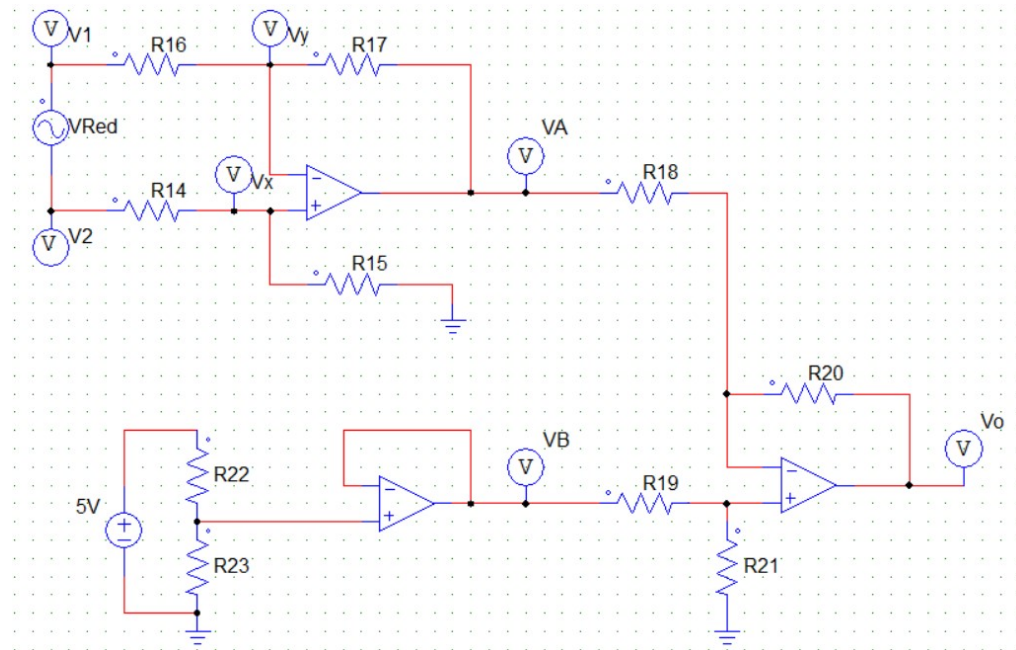


Figura 11: Esquema eléctrico completo del medidor de tensión

En una primera simulación, observamos que las impedancias de las resistencias son muy pequeñas, lo que provocaría distorsiones en las señales. Para evitar dichas distorsiones añadimos un buffer para aislar la tensión de entrada y la de salida.

Aunque a priori el ultimo amplificador operacional invertiría el signo de la señal original, no es así ya que la señal V_A ya está invertida respecto a la señal original medida ($V_1 - V_2$). Esto supondría que la señal que le llega al microcontrolador está desfasada todo un periodo respecto a la señal original, pero puesto que estamos trabajando con una frecuencia de unos 50Hz no es una diferencia significativa.

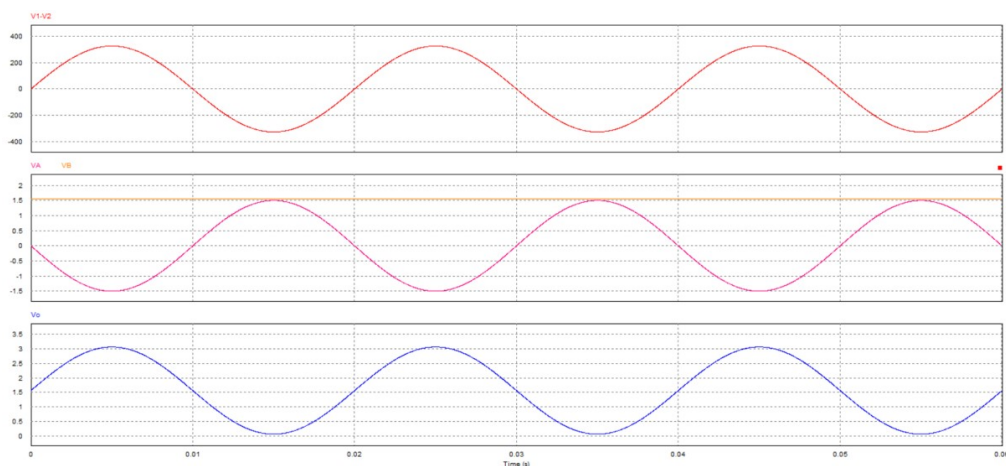


Figura 12: Grafica de la medición de tensión

Capítulo 4: Microcontrolador

4.1 Introducción

Para poder calcular el valor eficaz de las señales, emplearemos un chip programado previamente o microcontrolador, que nos servirá para leer la información proporcionada por los sensores, procesarla y enviar las señales adecuadas a los periféricos que se encargarán de mostrar las medidas y los resultados.

Hay muchas familias de microcontroladores en el mercado. Las dos opciones estudiadas a continuación han sido escogidas por su amplia distribución en el mercado, ya que son opciones económicas simples de programar.

4.2 Elección del microprocesador

La primera opción a analizar es el Arduino Nano Every. El microcontrolador utilizado en esta placa es el ATmega4809, con una velocidad de reloj de 20MHz. Tiene una tensión de operación típica de 5V y una corriente de 20mA. Cuenta con 8 entradas analógicas con una resolución de 8 bits cada una y 12 pins de entradas y salidas digitales, de las cuales 5 de ellas son capaces de dar una salida de tipo PWM. Además, puede soportar los protocolos de comunicación UART, SPI y I2C. [9]

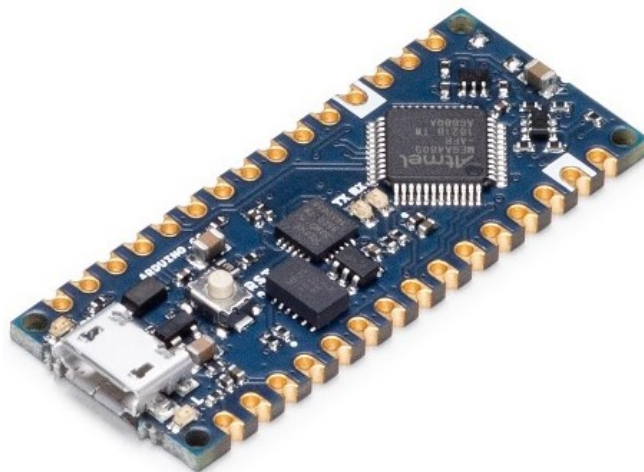


Figura 13: Arduino Nano Every

La segunda opción que analizaremos es el microcontrolador ESP32. El microprocesador utilizado en esta placa es el Tensilica Xtensa LX6 de doble núcleo, con una velocidad de reloj de hasta 240MHz. Tiene una tensión de operación típica de 5V y una corriente de 500mA, pero el voltaje máximo en sus entradas y salidas solo es de 3,3V, aunque cuenta con un pin de salida capaz de proporcionar 5V que podría usarse para alimentar el display. Cuenta con 18 entradas analógicas con una resolución de 12 bits cada una y 34 pines para entradas y salidas digitales. Además, soporta los protocolos de comunicación UART, SPI y I2C. [10]



Figura 14: ESP32

Pese a las desventajas iniciales del ESP-32, como la limitación de tensión en sus entradas o el hecho de solo poder ser alimentado por su puerto USB, sus ventajas hacen que sea la mejor opción para este proyecto, ya que es mucho más rápido en procesamiento y, como veremos más adelante, en nuestro aparato de medida necesitaremos trabajar con muchas muestras a la vez. Además, el código que se ha desarrollado es demasiado grande para la memoria disponible en otros microcontroladores.

4.3 Descripción del funcionamiento del programa

Comenzamos el programa definiendo las librerías, las constantes, los pines de entrada y todas las variables que se van a necesitar durante la ejecución del programa. A continuación, se inicializa la pantalla TFT, asegurándonos de poner

todos sus puntos led de color blanco para empezar la ejecución del programa con la pantalla limpia.

El primer paso será leer las entradas a las que están conectados ambos sensores. Puesto que queremos mostrar en la pantalla la forma de la onda, es mejor empezar la representación por un paso por cero de la señal, con una pendiente ascendente. Esta restricción complica un poco el proceso de lectura, por lo que el algoritmo solo aportará datos válidos una vez que haya encontrado y pintado el primer punto.

Mientras estamos buscando ese primer punto, entramos en un bucle en el que se hace una lectura cuyo valor se compara con las medidas anteriores para determinar si ha habido un cambio de signo en la señal y si las muestras tienen una pendiente ascendente. Almacenamos estos valores medidos en un vector, para poder utilizar esas medidas en futuras comparaciones en el bucle.

Puesto que el lenguaje Arduino no permite definir vectores sin un tamaño fijo, hay que añadir una condicional que aumente el valor del índice que indica las posiciones del vector donde se almacenan las lecturas temporales y que, en caso de llegar a la máxima capacidad del vector, haga de nuevo el índice 0. Tras acabar este bucle de forma exitosa, puesto que se ha encontrado ese primer paso por cero que se puede pintar en la pantalla, indicamos al programa que no tiene que volver a entrar en el bucle cambiando el valor de una variable booleana.

El siguiente paso sería pintar el punto correspondiente al valor leído en la pantalla. Pero antes de pintar, es más fácil borrar primero el punto situado en esa coordenada horizontal. Después, convertimos el valor leído que oscilará

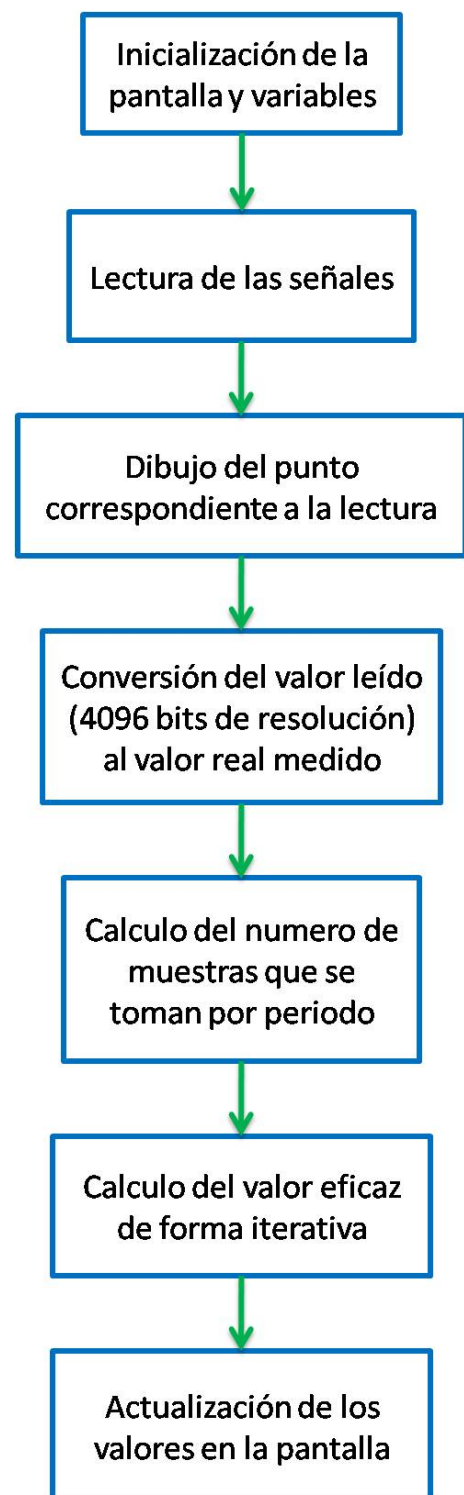


Figura 15: Diagrama de bloques del proceso seguido por el programa

entre 0 y 4095 al valor adecuado de la coordenada para representarlo en la pantalla en un rango entre 231 y 21. Borrando primero y pintando después nos permite emplear un único vector para almacenar los valores de las coordenadas de cada grafica que se pinta.

Con este valor se pinta en la pantalla un cuadrado rojo para la corriente y una estrella azul para la tensión. Guardamos ambos valores de las coordenadas en sus respectivos vectores para poder borrarlos cuando sea necesario. A continuación, incrementamos el valor de la coordenada horizontal para que, en la siguiente iteración, se borre y se dibuje el siguiente punto de la grafica.

A partir de este punto, en el programa utilizaremos los valores reales medidos, es decir, pasaremos los valores leídos, de entre 0 y 4095, al rango de valores reales medido, es decir, entre $\pm 71A$ para la corriente y $\pm 325V$ para la tensión.

El siguiente problema que debemos resolver es determinar el número de muestras que realizamos por periodo, ya que necesitaremos este valor para calcular el valor eficaz de las señales y no queremos limitar este proyecto solo para señales de 50Hz de la red. Partimos del hecho de que el primer punto detectado de la señal es un cero con una pendiente positiva. Por lo tanto, para que el algoritmo calcule cuántas muestras tiene un periodo solo necesitará encontrar el siguiente paso por cero con una pendiente positiva de la señal. Para poder determinar el signo de la pendiente de la señal, se almacenarán los valores reales en un vector. El número de muestras por periodo será el valor numérico anterior al detectar el fin del periodo.

La definición matemática para el cálculo del valor eficaz de una señal X cualquiera es:

$$X_{RMS} = \sqrt{\frac{1}{T} \int_0^T x^2(t) dt}$$

El periodo de una señal también se puede escribir como el número de muestras multiplicado por el tiempo de muestreo, que podemos considerar constante en nuestro código porque el tiempo de ejecución entre las mediciones es siempre el mismo.

$$T = N * t_m$$

Puesto que el microprocesador no es capaz de integrar, utilizaremos la transformada de Laplace para simplificar los cálculos.

$$X_{RMS}^2 = \frac{1}{T} \frac{1}{S} x^2$$

$$S X_{RMS}^2 = \frac{x^2}{T}$$

$$\frac{X_{RMS}^2 - X_{RMS-1}^2}{t_m} = \frac{x^2}{T}$$

$$X_{RMS}^2 = X_{RMS-1}^2 + \frac{t_m}{T} x^2$$

$$X_{RMS}^2 = X_{RMS-1}^2 + N \cdot x^2$$

Con esto podemos concluir que las ecuaciones para calcular los valores eficaces de forma iterativa para las señales de tensión y corriente son:

$$V_{ef}^2 = V_{ef-1}^2 + N \cdot v^2$$

$$I_{ef}^2 = I_{ef-1}^2 + N \cdot i^2$$

Para que funcione este algoritmo en la primera iteración el valor eficaz anterior debe de estar a cero. El valor eficaz válido se alcanzará cuando el valor de N sea el mismo que el número de muestras necesarias para abarcar el periodo de la señal. Una vez obtenido este dato, su raíz cuadrada será el valor eficaz buscado.

En caso de haber dibujado tantos puntos como caben en la pantalla, se reinician todas las variables necesarias para la siguiente iteración, es decir, se vuelve al inicio del programa. Con el fin de que los tiempos de ejecución sean los mismos en cada iteración del programa, si no hemos alcanzado el final de la pantalla, todas las variables se actualizan con su propio valor.

Si actualizásemos los valores eficaces mostrados en la pantalla tantas veces como se pintan las gráficas, el resultado obtenido sería que los valores cambian demasiado deprisa para el ojo humano como para leerlos claramente. Por lo tanto, solo actualizaremos los valores eficaces mostrados en la pantalla cada 50 gráficas pintadas.

4.4 Conexiones

Necesitamos que el microprocesador tenga dos entradas capaces de realizar lecturas de entradas analógicas.

Las únicas patillas que permiten realizar lecturas de señales analógicas con una resolución de 2^{16} bits, es decir, 4096 bits, son las entradas 34 y 35. Para nuestro proyecto utilizaremos la entrada 34 para la lectura de la tensión y la 35 para la lectura de la corriente.

Además de las entradas será necesario conectar las masas de las medidas con la masa del microprocesador.

Capítulo 5: Visualización de la información

5.1 Introducción

Una vez que las medidas han sido tomadas con los sensores y procesadas en el microcontrolador, debemos mostrarlas de alguna forma para que la persona que está manejando el aparato pueda leer los valores que se están midiendo en cada instante y los valores eficaces de las señales medidas.

Vamos a analizar varias opciones para encontrar la mejor solución a las necesidades de este proyecto.

5.2 Análisis de un display 16x2

Un display de 16x2 segmentos tiene 16 pines, de los cuales 8 son para controlar el dispositivo y los otros 8 son para la transmisión de caracteres e instrucciones que permitirán mostrar en él la información deseada.

Pines del modulo LCD:

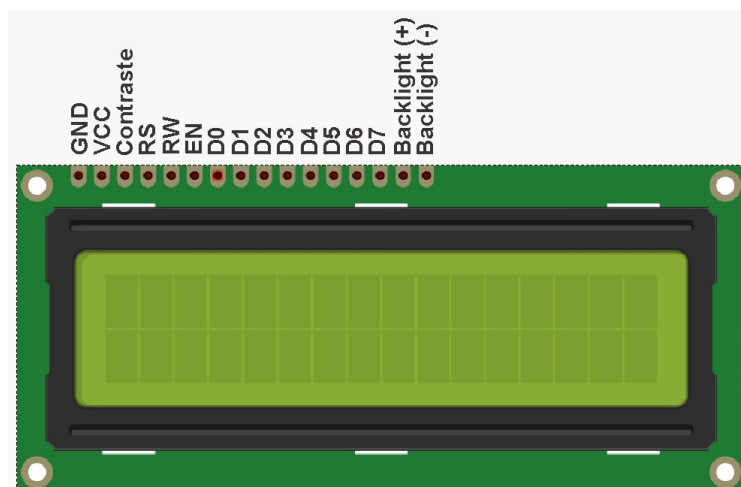


Figura 16: Pantalla LCD 16x2

- Pin 1 (GND): es la conexión a tierra del circuito que alimenta el display.
- Pin 2 (Vcc): es la conexión a 5V del circuito de alimentación del display.
- Pin 3 (Vo): es la conexión para ajustar el contraste del dispositivo mediante un potenciómetro.
- Pin 4 (RS): es la señal de selección de registro. Sirve para escoger el registro donde se almacena la información que se escribe en el display.

-Pin 5 (RW): es la señal que selecciona el modo de escritura o lectura. Para escribir se conecta a GND y para leer se conecta a 5V.

-Pin 6 (EN): es el habilitador de la señal de lectura/escritura.

-Pin 7-10 (D0-D3): son las líneas de datos, que comunican el microprocesador con el display, correspondientes a la parte baja de los datos de 8 bits.

-Pin 11-14 (D4-D7): son las líneas de datos, que comunican el microprocesador con el display, correspondientes a la parte alta de los datos de 8 bits.

-Terminal 15 (LED+): alimentación del LED BLK

-Terminal 16 (LED-): alimentación del LED BLK

Este tipo de display puede conectarse de cuatro formas diferentes: utilizando los 16 pines o utilizando solo 12 pines. En esta última forma de conexión, la información formada por 8 bits se envía al display dividida en dos partes: parte alta y parte baja. Para configurar la elección de una u otra se debe indicar al comienzo del programa. De la misma manera, podemos tener una conexión de 15 u 11 pines en caso de que queramos dejar que el ensamblador se encargue de decidir que registro utilizar, es decir, en estas dos formas no utilizaríamos el pin 4 del display.

También se puede conectar el display utilizando el bus de comunicación serie I2C. Este bus de comunicaciones utiliza dos líneas para transmitir la información, una para la línea de datos a transmitir y otra para la señal del reloj de sincronización. Los microcontroladores que son compatibles con este modo de comunicación tienen un par de pines específicos para implementar este protocolo. Además, el display y el microcontrolador han de tener una masa común.

5.3 Análisis de la una pantalla TFT 480x320

La pantalla escogida tiene 480x320 píxeles. Hay opciones en el mercado de mayor resolución y por lo tanto de mayor tamaño, pero con esta resolución sería suficiente para nuestra aplicación. Posteriormente, en el código se puede determinar la orientación de la misma, por lo que puede emplearse tanto en sentido vertical como en sentido horizontal.

Esta pantalla tiene 3 buses de comunicación SPI independientes entre sí. Uno sirve para controlar la lectura de una tarjeta SD, otro para controlar la lectura de la función táctil de la pantalla y el último para controlar el dibujo en la pantalla. Para nuestra aplicación solo utilizaremos este último.

El driver encargado de controlar este bus SPI es el ILI9488 [11], lo cual es un dato imprescindible a la hora de configurar la librería para que funcione con nuestro hardware [12].

5.4 Elección del método de visualización

El método elegido para la visualización es la pantalla TFT, ya que nos permite mostrar la forma de la gráfica y distinguir los posibles ruidos y armónicos de la misma empleando distintos colores. Esto nos permite diferenciar la representación de las dos señales que se van a medir, así como mostrar los valores numéricos de los valores eficaces; frente a la opción de utilizar un display 16x2 que solo nos permitiría mostrar los valores eficaces, sin representación gráfica.

5.5 Conexiones entre la pantalla y el microcontrolador

La pantalla emplea como sistema de comunicación el bus estándar SPI para transferir la información desde el microcontrolador (el maestro) a la pantalla TFT (el esclavo). El sistema SPI necesita de 4 canales para la sincronización y transmisión de la información.

Se utiliza un canal, llamado SCK, para marcar la sincronización entre ambos dispositivos mediante pulsos de reloj. Otro canal, llamado MOSI (Master Output Slave Input), está reservado para la información que el maestro envía al esclavo. Las respuestas que tenga que comunicar el esclavo al maestro se envían a través de otro canal llamado MISO (Master Input Slave Output). El cuarto canal CS (Chip Select), se utiliza para indicar con qué esclavo se quiere conectar el maestro en caso de que un maestro tenga conectado más de un esclavo.

La pantalla tiene 9 pines de entrada que utilizaremos en nuestro proyecto. 4 se utilizan para la comunicación SPI y los otros 5 para el funcionamiento del circuito electrónico.

La pantalla necesita alimentarse a 3.3V por lo que unimos el pin VCC a la salida del microcontrolador que los proporciona. También requiere de una conexión a tierra para cerrar el circuito, que uniremos a la tierra del microcontrolador para

que todo el montaje final tenga la misma conexión a tierra. La pantalla cuenta con un pin llamado LED que permite controlar la intensidad lumínica de los píxeles. La conectaremos a los 3.3V para máxima intensidad.

El driver de la pantalla ofrece la posibilidad de resetearla mediante un pin. Para evitar el reseteo habría que conectar este pin a la tensión positiva de 3.3V, pero como la librería empleada para el desarrollo del programa indica que este pin debe estar conectado a una de las posibles salidas del microcontrolador, lo hemos conectado al pin 4 del microcontrolador. El pin DC/RS se conecta a la entrada 2 del microcontrolador. [12]

Los pines encargados de la comunicación SPI hay que conectarlos a los pines del microcontrolador capaces de gestionar este tipo de comunicación. El pin MISO va conectado a la entrada 19 del microcontrolador, el pin SCK va conectado a la entrada 18 del microcontrolador, el pin MOSI va conectado al pin 23 del microcontrolador y el pin CS va conectado al pin 5 del microcontrolador.

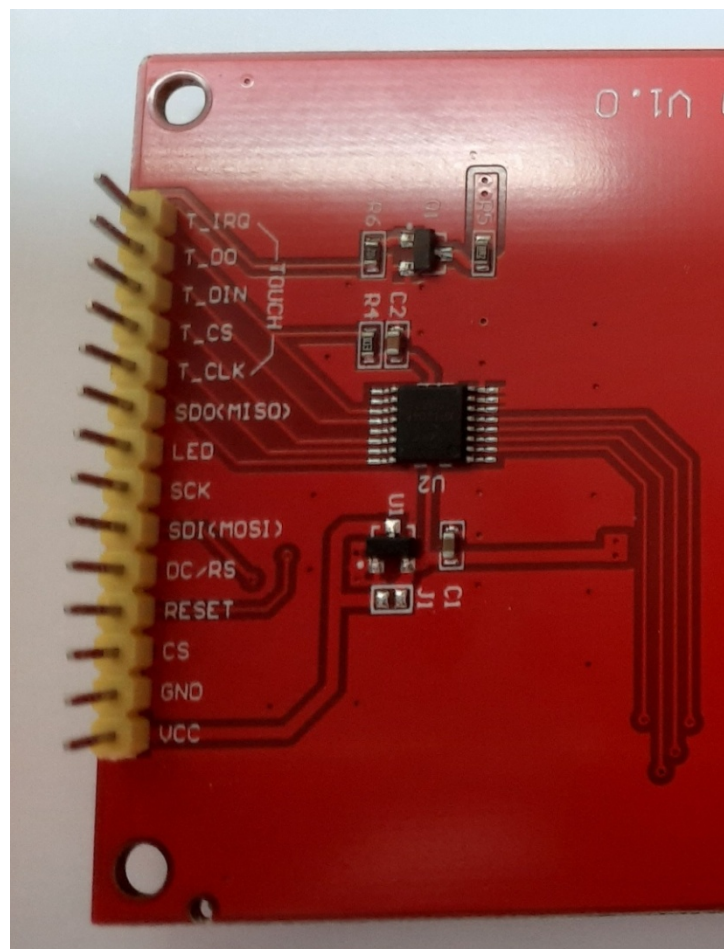


Figura 17: Pines de conexión pantalla TFT 480x320

Capítulo 6: Alimentación de los componentes electrónicos

6.1 Introducción

Una fuente de alimentación es un dispositivo electrónico que se encarga de transformar la corriente eléctrica de la red a la tensión continua requerida por otros dispositivos electrónicos. Necesitamos añadirla porque los componentes electrónicos empleados para medir, procesar y mostrar la información necesitan una pequeña tensión de corriente continua.

6.2 Diseño de la fuente de alimentación

El primer elemento en nuestra fuente de alimentación será un varistor para proteger todo el circuito. A continuación, colocaremos un transformador con toma intermedia, ya que necesitaremos tensiones continuas tanto positivas como negativas. La toma intermedia del secundario del transformador pasará a ser la toma de tierra para todos los componentes electrónicos conectados a tensión continua. El transformador escogido es el BV EI 482 1234 [13] para soportar los 230V de la red en el primario y 18V en cada una de las partes del secundario.

La salida del secundario se conectará a un puente rectificador de diodos. Para poder elegir el puente de diodos necesitamos conocer la tensión y corriente máxima de trabajo en el secundario del transformador, ya que es lo que deben soportar como mínimo los diodos del puente. Como la tensión eficaz en el secundario es de 18V, la tensión de pico serían 25,5V y el fabricante del transformador nos indica que la corriente en el secundario es de 667mA. Por lo tanto, los diodos que forman el puente deben de soportar tensiones y corrientes un poco mayores por seguridad.

A cada salida del puente rectificador de diodos, conectaremos dos condensadores electrolíticos en paralelo para suavizar el rizado y estabilizar la señal. Uno será de 2,2 μ F y el otro de 0,22 μ F. A estos conectaremos los primeros reguladores LM7815 [14] y LM7915 [15] para conseguir +15V y -15V de tensión continua respectivamente. Esto permitirá alimentar la tensión necesaria a los amplificadores operacionales. A la salida de cada uno de ellos colocamos otros dos condensadores electrolíticos en paralelo para hacer un último filtrado a la señal, de 100 μ F y 0,1 μ F.

Capítulo 7: Prototipo construido y pruebas

Con el fin de desarrollar el código adecuado para que el microprocesador realizase las funciones necesarias era necesario realizar pruebas en el laboratorio, en un entorno controlado y sin los riesgos de utilizar las grandes tensiones y corriente para las que se pretende diseñar este prototipo. En principio se ha desarrollado el código solo para la lectura de la señal de la corriente, ya que el algoritmo necesario para la medición de la tensión es el mismo, pero cambiando los valores numéricos dentro del propio programa.

Utilizamos un generador de ondas para simular la entrada al microprocesador, ajustándolo a una señal senoidal de 1,5V de continua, 3V de amplitud y una frecuencia de 50Hz. Esta señal correspondería a la medida realizada por la sonda hall tras pasar por el filtro y las adaptaciones necesarias para estar en el rango de de entradas al microprocesador.

El primer paso es entender la librería empleada para controlar la pantalla TFT [12], sus funciones para seleccionar el tamaño y la orientación del texto, así como las funciones para pintar líneas, rectángulos y puntos eligiendo el color de los mismos. A continuación diseñamos como será la disposición de la información en la pantalla. Para mostrar mejor la grafica colocamos la pantalla en sentido horizontal, dejando un pequeño espacio en la parte inferior de la misma con dos recuadros para poder mostrar los valores eficaces. El recuadro en el que se muestra el valor eficaz de la corriente es rojo y el de la tensión es azul, de esta forma podemos pintar los puntos de cada una de las señales en esos colores, de una forma más intuitiva.

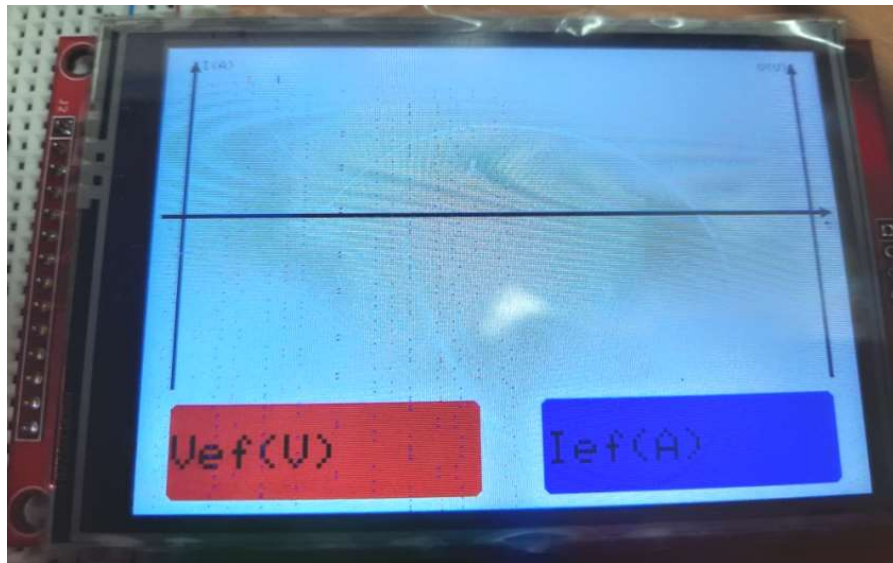


Figura 19: Fotografía primer diseño de la pantalla

Para empezar a hacer pruebas utilizamos un osciloscopio que leyese la señal producida por el generador de ondas, que también se está llevando como entrada al microprocesador, y la salida PWM que el microprocesador es capaz de realizar a partir de la señal que lee. De esta forma podemos conocer cuánto de la señal continua se pierde en el muestreo, a demás de asegurarnos que el microprocesador lee correctamente la señal de entrada.

Una vez que sabemos que la señal se lee correctamente la entrada el siguiente paso es pintar un punto por cada valor leído en la coordenada horizontal correspondiente. A demás de ser capaz de borrar el punto que se encontraba en esa coordenada horizontal en la anterior grafica que estaba pintada en la pantalla.

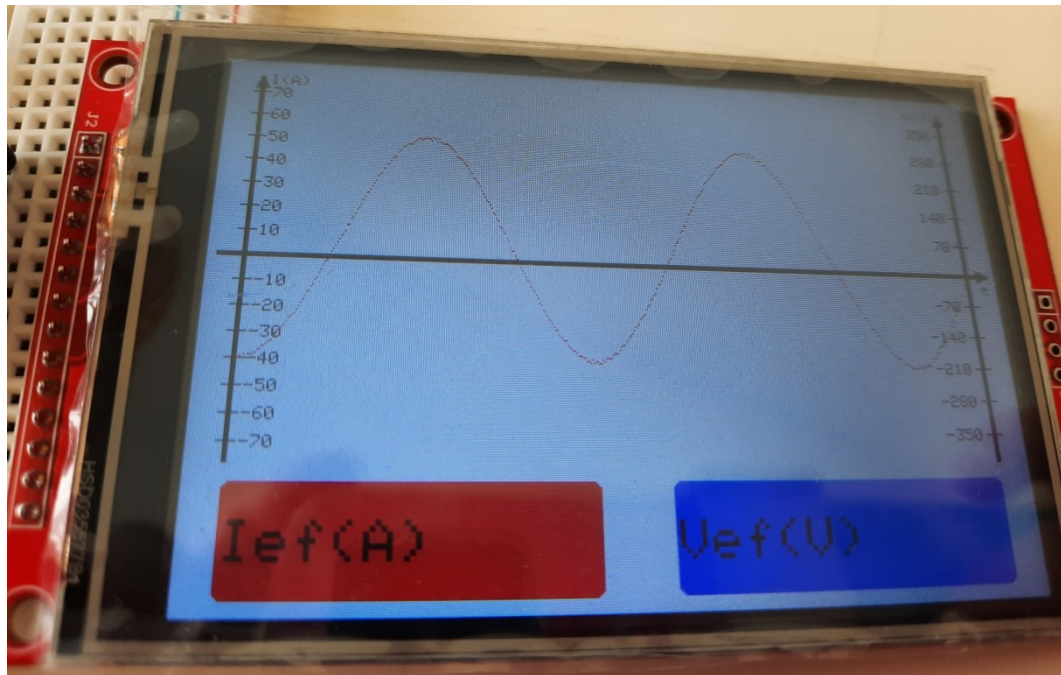


Figura 20: Fotografía Primera grafica dibujada en la pantalla

El siguiente paso era calcular y mostrar el valor eficaz de la señal en la pantalla. Para ello primero debíamos conseguir que la grafica mostrada empezase por un cero de la señal. De esta forma medir un periodo es más sencillo ya que sabemos siempre que el inicio del periodo corresponde con el inicio de la grafica. A partir de esa idea se desarrollo un algoritmo, explicado anteriormente en la descripción del funcionamiento del programa, capaz de calcular el valor eficaz de cualquier señal muestreada, sin necesidad de considerar que la frecuencia de la señal es una constante.

Hemos podido comprobar que el cálculo de los valores eficaces es correcto independientemente de la forma de la onda. Para ello hemos utilizado la función del generador de ondas para producir señales con otras formas, como la triangular o la cuadrada. Puesto que conocemos el valor de pico de la señal introducida y sus respectivas formulas matemáticas para calcular el valor eficaz a partir de este, y el valor calculado por el microprocesador coincide con el teórico. Podemos concluir que el valor eficaz que se muestra en la pantalla esta correctamente calculado.

La última prueba fue comprobar la fidelidad con la que capaz de mostrar señales con ruido o armónicos, producidos añadiendo en serie otro generador de ondas al que ya se estaba utilizando previamente.

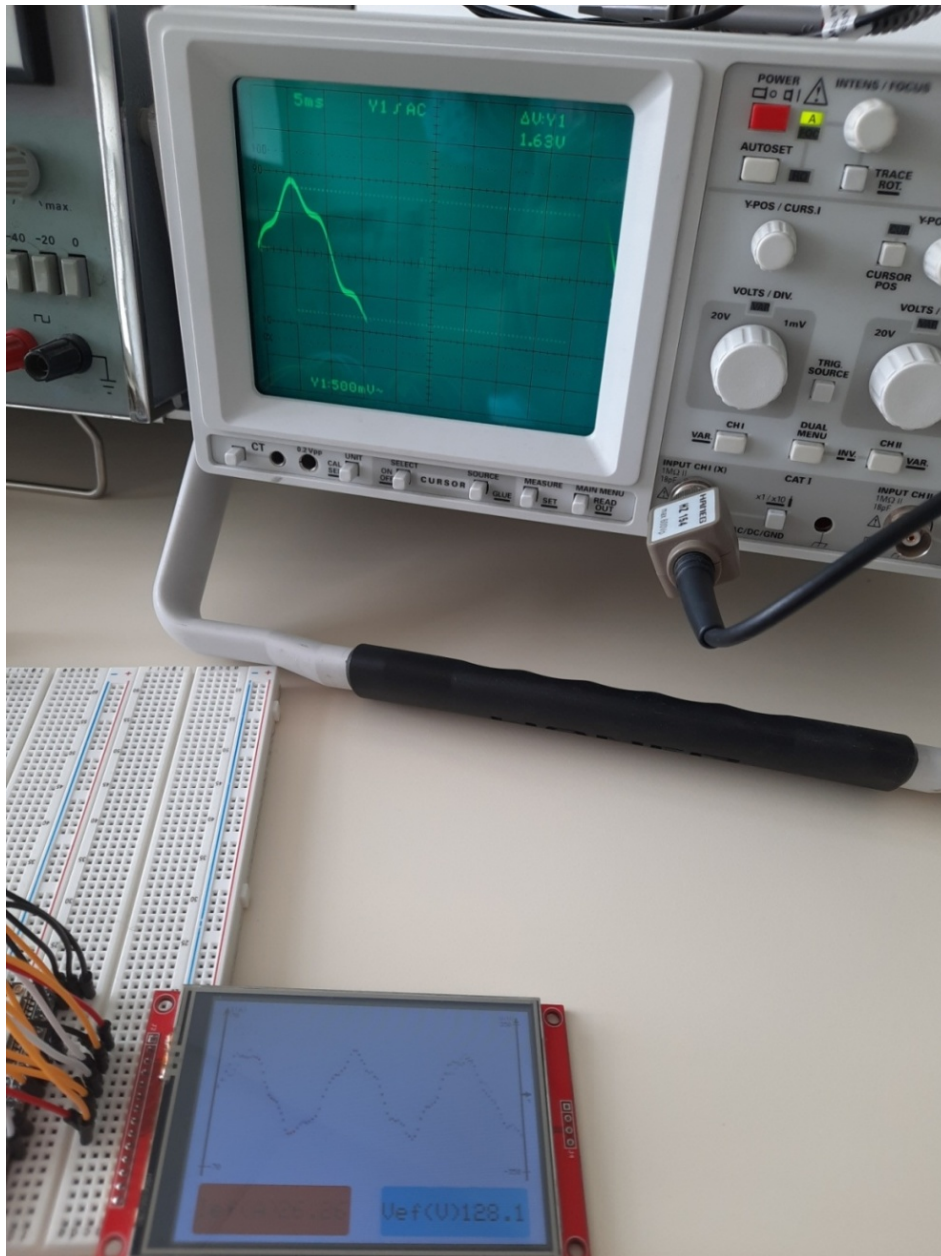


Figura 21: Fotografía osciloscopio y pantalla

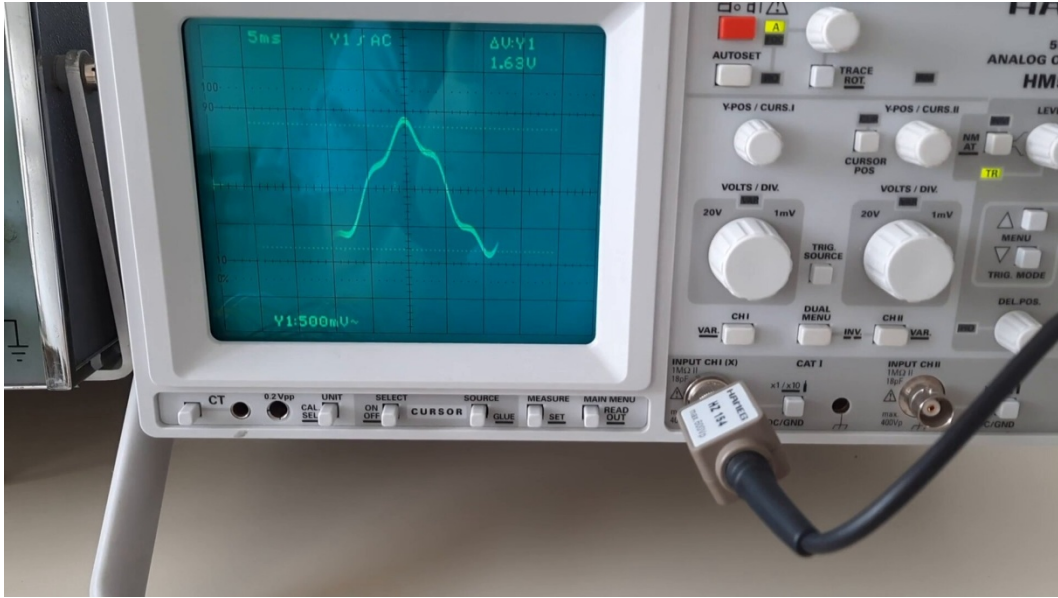


Figura 22: Fotografía osciloscopio

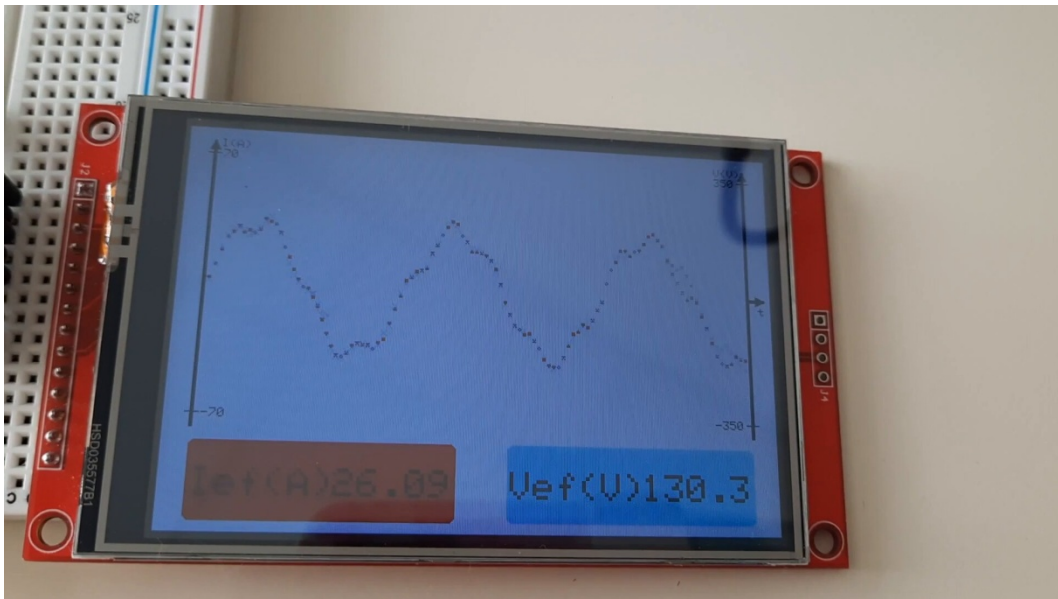


Figura 23: Fotografía pantalla

Capítulo 8: Conclusiones

En este trabajo se ha diseñado un equipo capaz de medir y mostrar las señales de corriente y tensión a la entrada de una fuente de corriente, mostrando tanto sus valores eficaces de forma numérica como sus formas de onda en forma gráfica.

Se ha mostrado en los diferentes capítulos, todo el proceso de diseño, cálculo y simulación del mismo. Para corroborar el proceso anterior, se ha construido un prototipo y se ha testeado su funcionamiento en laboratorio.

Como línea de trabajo futura, queda pendiente el diseño de una PCB definitiva que se adapte a la fuente de corriente mencionada para situar la de forma definitiva en el mismo. También se puede realizar un estudio para ver si es posible incluir más medidas de las señales como la distorsión armónica total de la señal y los principales armónicos.

Bibliografía

- [1] Velasco Caminero, M. (2021). *Estudio de sondas de corriente Rogowski para su empleo en sistemas de monitorización de eficiencia energética*. Universidad de Valladolid.
- [2] Tipler, P. A. (1993). *Física*. Reverté.
- [3] Allegro. *Thermally Enhanced, Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 100 $\mu\Omega$ Current Conductor ACS758LCB-100B-PFF-T Datasheet*.
- [4] SIEMENS. *SITOR cylindrical fuse link 3NC2200-0MK Datasheet*.
- [5] AVX. *Controlled Capacitance Multilayer Varistor VCACo60317X330MDP Datasheet*.
- [6] Robert L., B. & Louis, N. (2009). ***Teoría de Circuitos y Dispositivos Electrónicos 10ª Edición***. PEARSON.
- [7] LM358B. Wwww.ti.com. Recuperado el 20 de febrero de 2023, de <https://www.ti.com/product/LM358B>
- [8] Floyd, T. L. (2008). ***Dispositivos Electrónicos 8ª Edición***. PEARSON.
- [9] *Arduino Nano Every with headers*. Arduino Official Store. Recuperado el 20 de febrero de 2023, de <https://store.arduino.cc/arduino-nano-every-with-headers>

- [10]Espressif.com. Recuperado el 20 de febrero de 2023, de https://www.espressif.com/sites/default/files/documentation/esp32_data_sheet_en.pdf
- [11] aliexpress.com. Módulo de pantalla LCD TFT de serie SPI de 3,5 pulgadas, controlador de Panel táctil óptico IC ILI9341 para MCU, 320x240. Recuperado el 20 de febrero de 2023, de <https://es.aliexpress.com/item/1005001814475015.html?spm=a2g00.productlist.o.o.15297b9by7rnHh>
- [12] Bodmer. *TFT_eSPI: Arduino and Platform IO IDE compatible TFT library optimised for the Raspberry Pi Pico (RP2040), STM32, ESP8266 and ESP32 that supports different driver chips.* https://github.com/Bodmer/TFT_eSPI
- [13] Tme.eu. *Transformadores HAHN.* Recuperado el 20 de febrero de 2023, de <https://www.tme.eu/Document/412ff82a0d9ae839d6b24853d8f56565/Hahn-E.pdf>
- [14] LM7800. Www.ti.com. Recuperado el 20 de febrero de 2023, de <https://www.ti.com/product/LM7800>
- [15] LM79. Www.ti.com. Recuperado el 20 de febrero de 2023, de <https://www.ti.com/product/LM79>

Anexos

Código

```
#include "SPI.h" // Librerías para la comunicación SPI con
la pantalla TFT
#include "TFT_eSPI.h"
TFT_eSPI tft = TFT_eSPI();

// #define TFT_WHITE      0xFFFF //                // 255, 255,
255
#define TFT_CREMA        0xFFFE //                // 255, 255,
247
#define TFT_CORRIENTE    0xF800 //Rojo  0xF800 // 255,  0,
0
#define TFT_TENSION      0x067F //Azul  0x001F // 0,  0,
255

// Definiciones para hacer más claras las variables de
tipo bool
#define ON 1
#define OFF 0

int SensorPinI = 35; // Pin Analog Input Entrada de la
señal de la corriente
int SensorPinV = 34; // Pin Analog Input Entrada de la
señal de la tensión

int Contador, ContadorActualizacion;

double SensorValueI, SensorValueV; // Valores leídos por
los sensores
double lecturasCero[100]; // Almacenamos los valores de
las diez medidas anteriores para poder detectar bien la
pendiente ascendente de la señal
int contadorCero; // Para almacenar el índice adecuado
del vector anterior
bool INICIO, CERO; // Para habilitar o deshabilitar la
búsqueda de un periodo completo para empezar a pintar
en la señal

int ejeX; // Coordenadas para pintar las gráficas en la
pantalla
double ValorYCorriente, ValorYTension;
double AnteriorCicloI[150], AnteriorCicloV[150]; // Almacen
a los valores en un vector de la coordenada X que se
pintan en la pantalla para poder borrarlos después
```

```

double RealValueI, RealValueV; //Valores reales
instantaneos que se quieren medir
double RealValueIVector[150]; //Se utiliza para poder
determinar cuando se ha completado un ciclo de la señal
int N; //Numero de muestras que se toman por periodo
bool Nperiodo; //Esta variable indica si tenemos un
valor valido de N

double CorrienteEf2, CorrienteEfAnt2; //Variables para
calcular el valor eficaz de forma iterativa
double TensionEf2, TensionEfAnt2;
double CorrienteEf, TensionEf; //Variable que almacena el
valor eficaz calculado para cada periodo
double CorrienteEfP, TensionEfP; //Variables cuyo valor
se muestra por pantalla, la media de los valores
eficaces guardados en el vector anterior

void setup()
{
    pinMode(SensorPinI, INPUT);
    pinMode(SensorPinV, INPUT);

    //Inicializacion
    tft.init();
    tft.setRotation(3);
    Serial.begin(57600);
    tft.fillScreen(TFT_BLACK);

    FondoFijo(); // Dibujo de los elementos fijos de la
pantalla
    Contador = 0;
    ContadorActualizacion = 0;

    ejeX = 24;

    contadorCero = 0;
    for (int j = 0; j <= 100; j++)
    {
        lecturasCero[j] = 4096;
    }

    for (int i = 0; i <= 150; i++) //limpiar los vectores
de los puntos anteriores para limpiar la pantalla
    {
        AnteriorCicloI[i] = 0; //Para las coordenadas de la
grafica
        AnteriorCicloV[i] = 0;
    }
}

```

```

    RealValueIVector[i] = 0;//
}

CorrienteEf2 = 0;
CorrienteEfAnt2 = 0;
CorrienteEf= 0;
CorrienteEfP= 0;
TensionEf2 = 0;
TensionEfAnt2 = 0;
TensionEf= 0;
TensionEfP= 0;

INICIO = ON;
CERO = OFF;
Nperiodo= ON;
}

voidloop()
{
    Contador++; //Incrementamos la variable que se va a
    utilizar como indice en la mayoria de vectores

    //Lectura del sensor de corriente y de tension
    if (INICIO == OFF) //ON=1 OFF=0 //es decir si no
    estamos al principio de la pantalla buscando para
    empezar a pintar por un cero
    { //Lectura sin mas en cada vuelta del programa una
    vez hemos encontrado el cero inicial
        SensorValueI=analogRead(SensorPinI); //Resolucion
        2^12=4096
        SensorValueV=analogRead(SensorPinV);
    }
    else//Bucle para encontrar el cero inicial
    {
        contadorCero= 0;
        do
        {
            SensorValueI=analogRead(SensorPinI); //Lectura
            del los sensores
            SensorValueV=analogRead(SensorPinV);

            if (contadorCero>6)//Se hacen unas pocas lecturas
            antes de comprobar la pendiente y el paso por cero
            para que el ruido o las bajas frecuencias no den
            lecturas erroneas
            {

```

```

        //Si la pendiente es positiva, la medida
        anterior es menor que cero (2048) y la actual mayor que
        cero (2048), hemos encontrado un cero y cambiamos el
        estado de la variable que lo indica
        if (lecturasCero[contadorCero- 6]
<SensorValueI&&lecturasCero[contadorCero- 1] <= 2048 &&
        SensorValueI>= 2048)
            CERO = ON;
        }

        //Al macenamos el valor de la variable que hemos
        leído para poder ver la pendiente de la señal
        lecturasCero[contadorCero] =SensorValueI;

        if (contadorCero== 100)
            //reseteamos el contador al llega al maximo de
            la capacidad del vector, ya que no se puede definir un
            vector sin tamaño conocido
            contadorCero= 0;
        else
            //Si no hemos llegado al maximo de la capacidad
            del vector aumentamos contador para poder hacer y
            almacenar correctamente la siguiente lectura
            contadorCero++;
        } while (CERO == OFF);
        //Como hemos salido del bucle tras encontrar el
        primer cero indicamos con el cambio de estado de la
        variable que hemos completado la inicializacion de la
        lectura
        INICIO = OFF;
    }

    //Pintamos punto en la pantalla correspondiente al
    valor leído
    //Borrar el anterior punto que esta en esa coordenada
    Y para que la vieja grafica no entorpezca la
    visualizacion de la nueva
    BorrarGrafica(ejeX, AnteriorCicloI[Contador]);
    BorrarGrafica(ejeX, AnteriorCicloV[Contador]);

    //Expansiones y compresiones verticales de la señal
    para ajustarla al espacio disponible en la pantalla
    ValorYCorriente=map(SensorValueI, 0, 4095, 231, 21);
    ValorYTension=map(SensorValueV, 0, 4095, 231, 21);

    //Pintar nuevo punto.

```

```

    PintarCuadrado(ejeX,ValorYCorriente, TFT_CORRIENTE);
//En rojo, a base de cuadrados la corriente
    PintarCruz(ejeX,ValorYTension, TFT_TENSION); //En
azul, a base de cruces la tension

    ejeX=ejeX+ 4; //Incremento de la coordenada,
tiniendo en cuenta que los puntos son gordos y que
es mejor que la grafica a 50Hz quepan aprox 2 periodos
en la pantalla

    //Almacenar valores para poder borrarlos en la
siguiente iteracion
    AnteriorCicloI[Contador] =ValorYCorriente;
    AnteriorCicloV[Contador] =ValorYTension;

//Calculo del valor real medido
RealValueI=map(SensorValueI, 0, 4095,-71, 71);
RealValueV=map(SensorValueV, 0, 4095,-325, 325);

//Recalcular el valor de N para cuando varie la
frecuencia de la señal de entrada
//Partimos de la suposicion que como la pantalla
empieza a pintar y contar cuando la señal pasa por cero
con pendiente positiva, suponemos siempre que empezamos
por cero

    RealValueIVector[Contador] =RealValueI; //almacenar
los valores reales para poder determinar si la
pendiente de la señal es positiva o negativa en
proximos ciclos

    //Utilizando el valor real. Si la pendiente es
positiva, el valor anterior es menor que cero, el
actual mayor que cero,
    //nos hemos alejado de la primera medida(para evitar
que el ruido provoque una lectura incorrecta) y aun no
tenemos un valor valido para N
    if (RealValueIVector[Contador - 5]
<RealValueI&&RealValueIVector[Contador - 5] < 0 &&
RealValueI>= 0 && Contador > 10 &&Nperiodo== ON)
    {
        //El valor de N es el anterior al de contador,
porque cuando detectamos el cambio de signo ya seria el

```

```

primer valor del siguiente periodo, por lo que el
ultimo valor del primer periodo es Contador-1
    N = Contador - 1; //Nuevo valor de N
    Nperiodo= OFF; //Conocemos el valor de N
}
else//Para que el programa tarde lo mismo en cada
ciclo
{
    N = N;
    Nperiodo=Nperiodo;
}

//Calculo del valor eficaz de la señal de forma
iterativa
CorrienteEf2 = CorrienteEfAnt2 +
(RealValueI*RealValueI/ N);
TensionEf2 = TensionEfAnt2 + (RealValueV*RealValueV/
N);

CorrienteEfAnt2 = CorrienteEf2;
TensionEfAnt2 = TensionEf2;

if (Contador == N) //Se ha leído un periodo completo
de la señal, por lo que el valor eficaz es valido
{
    CorrienteEf=sqrt(CorrienteEf2);
    TensionEf=sqrt(TensionEf2);

    CorrienteEfAnt2 = 0;
    TensionEfAnt2 = 0;
}

//Al llegar al final de la pantalla nos volvemos a
colocar al principio de esta
if (Contador == 109) //145
{
    Contador = 0;
    ejeX= 24;
    ContadorActualizacion++;
    INICIO = ON; //Volvemos a buscar un inicio
    CERO = OFF; //No hemos encontrado aun un cero
    Nperiodo= ON; //No conocemos el valor de N

    CorrienteEfAnt2 = 0;//IMPRESINDIBLES

```



```

    TensionEfAnt2 = 0;
}
else//Para que el programa tarde lo mismo en cada
ciclo
{
    Contador = Contador;
    ejeX=ejeX;
    ContadorActualizacion=ContadorActualizacion;
    INICIO = INICIO;
    CERO = CERO;
    Nperiodo=Nperiodo;

    CorrienteEfAnt2 = CorrienteEfAnt2;
    TensionEfAnt2 = TensionEfAnt2;
}

//Actualizar valores eficaces por pantalla cada
muchas pantallas
if (ContadorActualizacion== 50)
{
    CorrienteEfP=CorrienteEf* 50 / 48.5;
    TensionEfP=TensionEf* 250 / 223.8;

    ActualizarPantalla (CorrienteEfP,TensionEfP); //
    ContadorActualizacion= 0;
}
}

voidPintarCuadrado(int Coordenada1,double
Coordenada2,uint32_t color)
{
    //fillRect(int32_t x, int32_t y, int32_t w, int32_t
h, uint32_t color);
    tft.fillRect(Coordenada1 - 1, Coordenada2 - 1, 3, 3,
color);
}

voidPintarCruz(int Coordenada1,double
Coordenada2,uint32_t color)
{
    //drawLine(int32_t xs, int32_t ys, int32_t xe,
int32_t ye, uint32_t color);
    tft.drawLine(Coordenada1 - 2, Coordenada2 - 2,
Coordenada1 + 2, Coordenada2 + 2, color);
    tft.drawLine(Coordenada1 - 2, Coordenada2 + 2,
Coordenada1 + 2, Coordenada2 - 2, color);
}

```

```

}

voidBorrarGrafica(int C1,double C2)
{
    int B = C2 - 2;
    //Para borrar el punto de la iteracion pasada en el
    caso de que el punto estuviese en el eje horizontal
    borramos el punto en negro, si no en blanco
    for (int Q = 0; Q <= 5; Q++)
    {
        if (B <= 127 && B >= 125) //entre 125 y 127
            tft.drawLine(C1 - 2, B, C1 + 2, B,TFT_CREMA);//
TFT_BLACK);
        else
            tft.drawLine(C1 - 2, B, C1 + 2, B, TFT_CREMA);
        B++;
    }
}

voidActualizarPantalla (double Corriente,doubleTension)
{
    //Repintamos los ejes verticales para evitar que los
    puntos blancos los borren demasiado
    EjeVerticalI();
    EjeVerticalD();
    //para borrar el hueco tras los numeros, por si un
    numero es demasiado grande para el recuadro
    //tft.fillRect(0, 252, 20, 80, TFT_CREMA);
    tft.fillRect(220, 252, 40, 60, TFT_CREMA);
    tft.fillRect(460, 252, 20, 60, TFT_CREMA);

    RecuadroI();
    RecuadroV();

    //Escribir valor eficaz por la pantalla
    tft.setTextColor(TFT_BLACK);
    tft.setTextSize(3);
    tft.setCursor(127, 272);
    tft.println(Corriente, 2);
    tft.setCursor(367, 272);
    tft.println(Tension, 1);
}

voidFondoFijo()
{
    //Base
    tft.fillRect(TFT_CREMA);

    RecuadroI();
}

```

```

    RecuadroV();

    EjeVerticalI();
    EjeVerticalD();
    EjeHorizontal();
}

voidRecuadroI()
{
    tft.fillRoundRect(20, 252, 200, 60, 5,
TFT_CORRIENTE);
    tft.setTextColor(TFT_BLACK);
    tft.setTextSize(3);
    tft.setCursor(22, 272);
    tft.println("Ief(A)");
}

voidRecuadroV()
{
    tft.fillRoundRect(260, 252, 200, 60, 5, TFT_TENSION);
    tft.setTextColor(TFT_BLACK);
    tft.setTextSize(3);
    tft.setCursor(262, 272);
    tft.println("Vef(V)");
}

voidEjeVerticalI()//eje para medir la corriente
{
    //Eje vertical izquierda
    tft.drawLine(20, 10, 20, 242, TFT_BLACK);
    tft.drawLine(19, 11, 19, 242, TFT_BLACK);
    tft.drawLine(21, 11, 21, 242, TFT_BLACK);
    tft.fillTriangle(20, 10, 16, 18, 24, 18, TFT_BLACK);
    tft.setTextColor(TFT_BLACK);
    tft.setCursor(25, 9);
    tft.setTextSize(1);
    tft.println("I(A)");

    //subdivisiones del eje
    tft.drawLine(15, 21, 25, 21, TFT_BLACK);
    tft.setCursor(27, 18);
    tft.println(70);

    tft.drawLine(15, 231, 25, 231, TFT_BLACK);
    tft.setCursor(27, 228);
    tft.println(-70);
}

voidEjeVerticalD()//eje para medir la tension

```

```

{
  //Eje vertical derecha
  tft.drawLine(460, 10, 460, 242, TFT_BLACK);
  tft.drawLine(459, 11, 459, 242, TFT_BLACK);
  tft.drawLine(461, 11, 461, 242, TFT_BLACK);
  tft.fillTriangle(460, 10, 464, 18, 456, 18,
TFT_BLACK);
  tft.setTextColor(TFT_BLACK);
  tft.setCursor(435, 9);
  tft.setTextSize(1);
  tft.println("V(V)");

  //subdivisiones del eje
  tft.drawLine(455, 21, 465, 21, TFT_BLACK);
  tft.setCursor(435, 18);
  tft.println(350);

  tft.drawLine(455, 231, 465, 231, TFT_BLACK);
  tft.setCursor(429, 228);
  tft.println(-350);
}

voidEjeHorizontal()
{
  //Eje horizontal
  tft.drawLine(460, 126, 475, 126, TFT_BLACK);
  tft.drawLine(460, 127, 474, 127, TFT_BLACK);
  tft.drawLine(460, 125, 474, 125, TFT_BLACK);

  tft.fillTriangle(475, 126, 467, 122, 467, 130,
TFT_BLACK);
  tft.setCursor(469, 132);
  tft.setTextSize(1);
  tft.println("t");
}

```