

GRADO EN
INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA

TRABAJO FIN DE GRADO

ANÁLISIS DE CONSUMO MONOFÁSICO

Alumna: Lorenzo Ochoa, Bertha Joosely.

Director: Oleagordia Aguirre, Iñigo Javier.

Curso: 2017-2018

Fecha: 18,07,2018

RESUMEN

A medida que una sociedad es más desarrollada, consume más energía, pero no siempre lo hace de un modo eficiente. La eficiencia energética provoca un aumento de la calidad de vida. Con un uso responsable y eficiente, se podría disponer de mayores prestaciones de servicios y confort sin consumir más energía. Eso, además, nos haría menos vulnerables ante posibles crisis de suministros.

La energía final tal como se usa en los puntos de consumo, es la electricidad. La electricidad, producida en centrales es adaptada para que pueda transportarse a los lugares de consumo a través de redes de transmisión y distribución; durante la transmisión de esta energía hay muchos factores externos que pueden perjudicar la calidad de la energía suministrada.

El objetivo principal del proyecto es realizar un dispositivo capaz de medir el consumo de la energía eléctrica que realizan los electrodomésticos utilizados en el hogar y comprobar la calidad. Lo que permite monitorizar el consumo de cada dispositivo, permitiendo realizar las acciones correctivas necesarias para disminuir su consumo, ya sea regularlos o utilizarlos en horas a las que la energía cueste menos.

El medidor constará de tres partes:

1. La parte encargada de realizar las mediciones:
 - Un sensor que permite medir la intensidad que atraviesa un conductor sin necesidad de cortar o modificar el conductor.
 - La parte del transformador que permite medir la tensión alterna a la que se conecta el electrodoméstico a estudiar.
2. Emisor (Arduino Uno): Conectado de manera física al sensor y al transformador a través de un circuito acondicionador. Realiza la conversión de la señal a un valor digital y los envía al receptor a través de un cable USB.
3. Receptor (LabVIEW): se podría decir que es el cerebro del sistema, a este le llegan los datos del emisor y realiza los cálculos necesarios para dar, en una pantalla, los valores de consumo.



ÍNDICE DEL TEXTO

MEMORIA.....	1
1. Introducción	1
2. Contexto.....	1
2.1 Conceptos generales de la distribución de la energía eléctrica.....	1
2.2 Conceptos generales de la energía alterna.....	2
2.3 Análisis de una señal	3
2.4 Selección de una frecuencia de muestreo adecuada.....	6
2.5 Transformada de Fourier.....	10
3. Objetivos	11
4. Beneficios que aporta el trabajo	11
5. Descripción de requerimientos.....	12
6. Diseño.....	13
6.1 Planteamiento del sistema.....	14
6.2 Unidad de adquisición.....	15
6.2.1 Eligiendo la plataforma	15
6.2.2 Introducción a Arduino	16
6.2.3 Arquitectura de Arduino	16
6.2.4 ADC.....	17
6.3 Acondicionamiento de sensores	18
6.3.1 Sensor SCT-013-000	18
6.3.2 Transformador.....	25
6.4 Comunicaciones	31
6.4.1 Instalación toolkit-Arduino y LIFA-base	32
6.4.2 LabVIEW interface for Arduino (LIFA)	33
6.5 Programación en LabVIEW.....	34
6.5.1 Instrumentos Virtuales.....	34
6.5.2 Introducción al LabVIEW	35
6.5.3 Modo de trabajo en LabVIEW	35
6.5.4 Programación gráfica de LabVIEW	38
6.6 Implementación de la electrónica en PCB	47
METODOLOGÍA SEGUIDA EN EL DESARROLLO DEL TRABAJO	48
7. Descripción de tareas, fases, equipos o procedimientos.....	48
8. Diagrama de Gantt.....	49



9.	Resultados	51
9.1	Instrumento virtual final	52
9.1.1	Panel Frontal	52
9.1.2	Diagrama de Bloques	54
9.2	Mejora del dispositivo	55
ASPECTOS ECONÓMICOS		55
10.	Presupuesto ejecutado	55
11.	Análisis de rentabilidad	56
CONCLUSIONES		57
BIBLIOGRAFÍA.....		58
ANEXOS I		59
12.	Programación en arduino.....	59
1.1	Intensidad.....	59
1.2	Tensión	60
13.	Esquemas del diseño	61
14.	ATmega328/p Datasheet	63

Índice de tablas

Tabla 1.	Tiempo de conversión dl ADC.....	8
Tabla 2.	Características del Arduino UNO	15
Tabla 3.	Características conversor analógico.	17
Tabla 4.	Parámetros Técnicos del sensor SCT-013-000.....	20
Tabla 6.	Comparación entre instrumento real, LabVIEW sin calibrar y LabVIEW calibrado	25
Tabla 7.	Comparación entre los valores calculados por diferentes dispositivos.	31
Tabla 8.	Presupuesto.....	55



Índice de figuras

Fig. 1. Suministro eléctrico	1
Fig. 2. Onda sinusoidal de tensión y corriente	2
Fig. 3. Características de una señal de alterna	2
Fig. 4. Desfase entre tensión e intensidad según tipo de reactancia	3
Fig. 5. Etapas para la adquisición de datos.	4
Fig. 6. Características técnicas del ADC del Arduino UNO.	5
Fig. 7. Bloque esquemático del convertidor analógico digital del Arduino UNO	6
Fig. 8. Aparición de una señal fantasma por sub muestreo	7
Fig. 9. Preescalado y tiempo de conversión del ADC.	8
Fig. 10. Diagrama de tiempo y conversión auto trigger del ADC	8
Fig. 11. Programación de muestreo continuo en la librería de Interfaz LIFA	9
Fig. 12. Configuración del cálculo espectral en el panel de bloques de LabVIEW	10
Fig. 13. Sistema de adquisición del proyecto	13
Fig. 14. Arquitectura del proyecto.	14
Fig. 15. Placa Arduino UNO.	17
Fig. 16. Escalón mínimo ó LSB	18
Fig. 17. Sensor SCT-013-000	19
Fig. 18. Esquemático Circuito de acondicionamiento de la señal de intensidad.	22
Fig. 19. Representación de la curva de calibración ideal y la que corresponde a dos instrumentos de medida	23
Fig. 20. Curva de calibración de la intensidad	24
Fig. 21. Transformador	26
Fig. 22. Esquemático de circuito acondicionador de tensión.	28
Fig. 23. Circuito acondicionador de tensión	29
Fig. 24. Circuito para la calibración del sistema	30
Fig. 25. Curva de calibración del instrumento ideal y el instrumento diseñado calibrado	31
Fig. 26. Sistema de instrumentación virtual	35
Fig. 28. Paletas de Controles	37
Fig. 27. Panel frontal y de bloques de LabVIEW	36
Fig. 29. Paletas de funciones	37
Fig. 30. Paleta de herramientas	38
Fig. 31. Paletas de programación para Arduino	38
Fig. 32. Configuración del puerto de comunicación	39
Fig. 34. Bloque Continuous Acquisition ON	40
Fig. 33. Bloque Close	39
Fig. 35. Características de una onda sinusoidal	40
Fig. 36. Propiedades de la gráfica de la transformada rápida de Fourier	41
Fig. 37. Representación de la señal con un muestreo de 870 Hz	41
Fig. 38. Panel Frontal con la velocidad de adquisición a 870Hz	42
Fig. 39. Representación de la señal con una velocidad de adquisición de 10 Hz	42
Fig. 40. Representación de la señal con una velocidad de adquisición de 1001 Hz	43
Fig. 41. Bloque Continuous Acquisition Sample:	44
Fig. 42. Bloque WaveForm Peak Detection	44



fig. 43. Bloque para calcular la intensidad eficaz de del conductor.....	45
Fig. 44. Bloque para guardar los datos en un archivo .txt.....	45
fig. 45. Bloque para el análisis espectral.....	45
fig. 46. Diseño PCB mediante DesignSpark.....	47
Fig. 47. Prototipo de final.....	51
Fig. 48. Panel Frontal Final.....	52
Fig. 49. Diagrama de bloques final.....	55

MEMORIA

1. INTRODUCCIÓN

La monitorización del consumo eléctrico en el hogar puede derivar en ahorros económicos importantes. Permite conocer el consumo de reactiva y los excesos de la potencia contratada por la que se cobran implementos. Además permite ajustar el factor de potencia de la instalación y disminuir así el cobro de la potencia reactiva (potencia no útil).

Hay varios tipos de equipos en el mercado que se pueden utilizar para monitorizar la calidad de la energía eléctrica, o para realizar estudios del consumo de energía. Sin embargo, son generalmente muy costosos. En este trabajo fin de grado se ha diseñado un analizador de consumo de potencia eléctrica, utilizando Arduino y LabVIEW con el objetivo de reducir costes.

2. CONTEXTO

2.1 Conceptos generales de la distribución de la energía eléctrica

En general, una central generadora es una instalación capaz de convertir en energía eléctrica la energía mecánica procedente de la transformación de la energía potencial del agua almacenada en un embalse; de la energía térmica suministrada al agua mediante la combustión del carbón, gas natural, o fuel, o a través de la energía de fisión del uranio.

Esta energía eléctrica generada se envía a través de una red de líneas eléctricas hasta los lugares de consumo. A la salida de la central eléctrica, mediante un transformador se obtiene una alta tensión (110-380kV) para que las pérdidas en transporte sean mínimas y se distribuye al cliente industrial en forma de tensión trifásica. Después del suministro eléctrico de alta tensión se vuelve a transformar, mediante otros transformadores, a baja tensión (380V /220V) que llega a los hogares en forma de tensión monofásica.

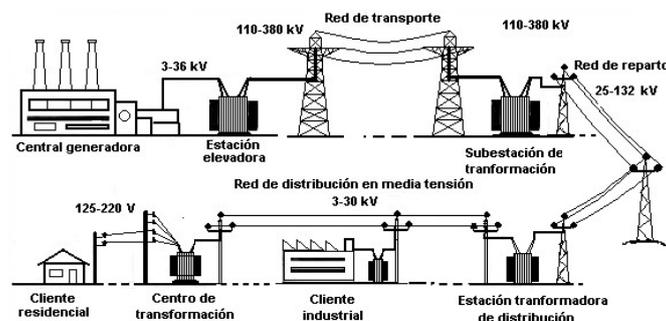


Fig. 1. Suministro eléctrico.

Sistema monofásico: Es un sistema de producción, distribución y consumo de energía eléctrica formado por una única corriente alterna o fase y por lo tanto todo el voltaje varía de la misma forma.

Sistema trifásico: Es un sistema de producción, distribución y consumo de energía eléctrica formado por una tres corrientes alternas monofásicas de igual frecuencia y amplitud que presentan una cierta diferencia de fase entre ellas, en torno a 120° , y están dadas en un orden determinado.

2.2 Conceptos generales de la energía alterna

Corriente alterna: Se caracteriza porque varía periódicamente en magnitud y sentido. La forma de oscilación de esta corriente más utilizada es la sinusoidal, con la que se consigue una transmisión más eficiente de la energía. El voltaje varía con la misma forma. Si representamos las dos magnitudes juntas podríamos tener algo como se muestra en la siguiente imagen:

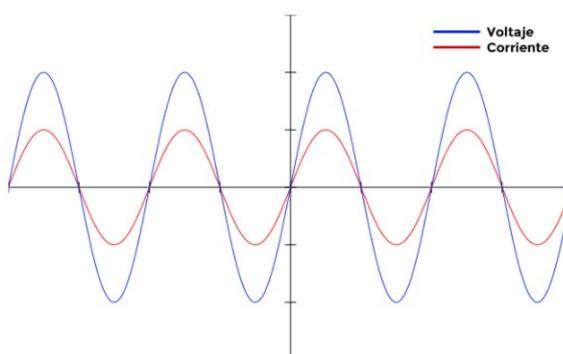


Fig. 2. Onda sinusoidal de tensión y corriente.

Los parámetros fundamentales de una señal sinusoidal son la frecuencia, el periodo, valor instantáneo, valor máximo (I_{max}), valor medio (I_{med}), valor eficaz (I_{efic}) y valor pico a pico (I_{pp}).

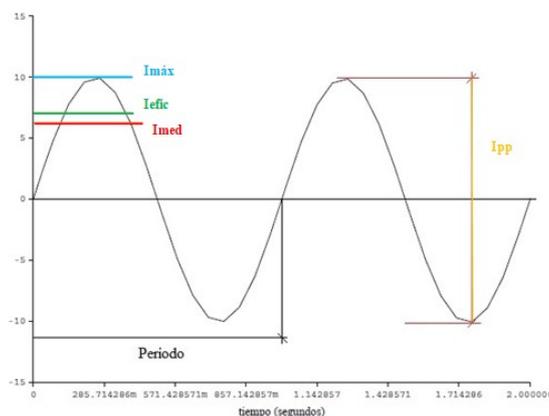


Fig. 3. Características de una señal de alterna.

Valor eficaz o RMS: Es el valor del voltaje o corriente que produce el mismo efecto de disipación de calor que su equivalente de voltaje o corriente en corriente continua.

Ecuación 1:

$$V_{RMS} = \frac{V_{m\acute{a}x}}{\sqrt{2}}$$

Dado que el objeto principal del proyecto es el análisis de la potencia consumida en un circuito y poder observar las mediciones resultantes en un monitor; a partir de la fórmula de potencia $P = V \cdot I$ y teniendo en cuenta que la señal de tensión es una función sinusoidal, dependiendo de la naturaleza del dispositivo se pueden dar los siguientes casos:

- **Dispositivo resistivo:** se asemeja a un caso ideal donde la corriente y el voltaje están en fase (sin desplazamiento en el tiempo). Son dispositivos cuya impedancia se debe exclusivamente a una componente resistiva; que usan toda la energía que se les suministra. Por ejemplo, las bombillas incandescentes, un calentador de agua o una plancha.
- **Dispositivo inductivo:** este caso ocurre cuando la impedancia del dispositivo se debe a un componente inductivo. Debido a esto la intensidad que circula por la carga se retrasa eléctricamente con respecto de la tensión, este desfase es de 90° . Las cargas inductivas pueden ser motores y/o transformadores.
- **Dispositivo capacitivo:** este caso ocurre cuando la impedancia del dispositivo se debe a una componente capacitiva. Debido a esto la corriente que circula por la carga se adelanta eléctricamente con respecto de la tensión, este desfase es de 90° .

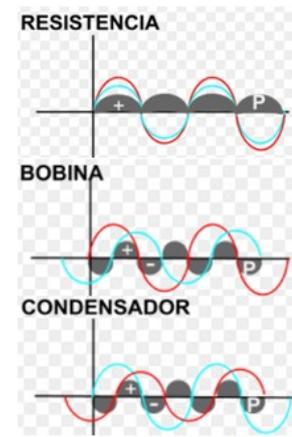


Fig. 4. Desfase entre tensión e intensidad según tipo de reactancia.

2.3 Análisis de una señal

La instrumentación es una rama científico-técnica cuyo objetivo es mejorar la capacidad de percepción de la realidad mediante el uso de cualquier tipo de técnicas y sistemas. La percepción de la realidad hace referencia a la extracción de información del entorno mediante la identificación y cuantificación de sus variables características.

Los sistemas de instrumentación electrónica entregan la información mediante señales eléctricas, listas para ser procesadas por otros sistemas electrónicos. La estructura del proyecto se presenta en la figura 5.



Fig. 5. Etapas para la adquisición de datos.

El término de sensor representa el concepto genérico de lectura de una información y entrega de una señal de salida en función de esa información. En este caso, el sensor recibe la información de la intensidad que se pretende medir y produce, como señal eléctrica, una corriente.

Dado que el sistema de análisis se basa en un microprocesador y los microprocesadores solo trabajan con señales digitales, es necesario que las señales producidas por los sensores (en este caso, analógicas) se adapten a las características de entrada del sistema de control utilizado. Para ello es necesario un circuito de acondicionamiento.

Una vez acondicionada la señal analógica, se podrá introducir en el sistema de control. Este sistema de control está formado por un circuito Arduino Uno que dispone, entre otras cosas, de un convertidor analógico –digital (ADC, en adelante) que permite la transformación de señales analógicas en señales digitales.

La resolución de un convertidor es su característica más destacada puesto que, como idea general, cuanta más resolución tenga un convertidor, mayor será su capacidad de distinguir valores y, por lo tanto, más se acercará al valor analógico real.

Otro parámetro relevante de los convertidores y que ha generado problemas a la hora de analizar la señal, es el tiempo de conversión, que se define como el tiempo que tarda el convertidor desde el momento en que recibe la orden de conversión hasta que tiene un valor digital válido. Es un parámetro importante porque define la separación temporal entre las muestras y, a la postre, la cantidad de información que se obtiene. Hay que tener en cuenta que mientras se hace la conversión de la señal analógica, esta puede variar; de ahí la importancia de realizar la conversión en el menor tiempo posible. En función de la rapidez con la que se lleva a cabo esta conversión se obtendrá una “frecuencia máxima de muestreo” en el ADC.

En líneas generales, una frecuencia de muestreo alta nos permitirá tomar muchas lecturas de la señal analógica con lo que la señal digital obtenida será más parecida a la señal analógica muestreada.

Teniendo en cuenta que en una señal de amplitud A , que tiene como frecuencia máxima $f_{m\acute{a}x}$, la máxima pendiente que puede representar la señal es:

Ecuación 2:

$$\Delta V = 2 \cdot \pi \cdot A \cdot f_{m\acute{a}x} \cdot Tc$$

Pues bien, en este caso, ΔV tiene que ser menor que 1 LSB para conseguir que la digitalización no tenga posibilidad de error.

En el caso que nos atañe, Arduino uno tiene un convertidor A/D de 10 bits de resolución que acepta señales entre 0 y 5 V y tiene un tiempo de conversión máximo de 260 μ s. A continuación se realizará la comprobación de la máxima frecuencia de la señal real a medir con la que puede trabajar sin empeorar los errores de conversión:

$$1 \text{ LSB} = \frac{5}{2^{10} - 1} = 4.888 \text{ mV}$$

$$A = \frac{5 \text{ V}}{2} = 2.5 \text{ V}$$

$$\Delta V = 2 \cdot \pi \cdot A \cdot f_{\text{máx}} \cdot T_c \rightarrow f_{\text{máx}} = \frac{\Delta V}{2 \cdot \pi \cdot A \cdot T_c} = \frac{4.888 \cdot 10^{-3}}{2 \cdot \pi \cdot 2.5 \cdot 260 \cdot 10^{-6}} = 1.196 \text{ Hz}$$

Lo que nos indica que el convertidor del Arduino no será capaz de convertir la frecuencia de la red, de 50 Hz, sin producir errores. El problema no radica en que el convertidor no funcione bien, está en que la señal varía durante la conversión con lo que no es posible una conversión correcta. Si se consigue que la señal no cambie durante el proceso de conversión, el asunto estaría resuelto. Esto se consigue con un dispositivo denominado sistema de muestro y retención (*sample & -hold*), que es capaz de mantener constante el valor de la tensión de entrada durante todo el tiempo que dura la conversión.

En la siguiente gráfica del Datasheet del Arduino se puede observar que contiene un dispositivo de Sample & hold.

ADC - Analog to Digital Converter

Features

- 10-bit Resolution
- 0.5 LSB Integral Non-Linearity
- ± 2 LSB Absolute Accuracy
- 13 - 260 μ s Conversion Time
- Up to 76.9kSPS (Up to 15kSPS at Maximum Resolution)
- Six Multiplexed Single Ended Input Channels

Fig. 6. Características técnicas del ADC del Arduino UNO.

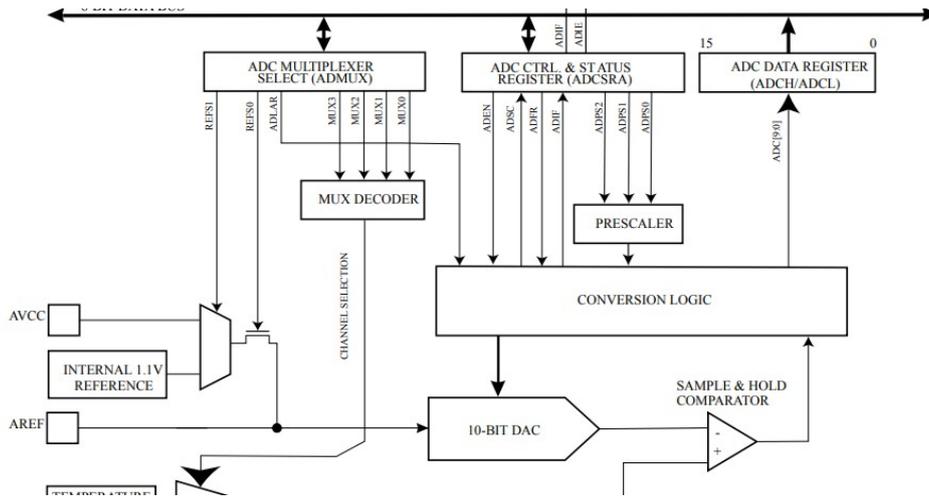


Fig. 7. Bloque esquemático del convertidor analógico digital del Arduino UNO

2.4 Selección de una frecuencia de muestreo adecuada

En una primera aproximación, trabajar con frecuencia de muestreo elevadas implica manejar mucha información lo que supone mayor carga de procesamiento por parte del sistema digital; sin embargo, esa cantidad de muestras permitirá obtener una imagen más exacta de la señal original.

A través del teorema del muestreo de Nyquist se determina la frecuencia de muestreo necesaria para recoger toda la información de la señal que se desea analizar. Según este teorema, una señal periódica y continua en el tiempo, puede ser reconstruida totalmente a partir de una señal muestreada siempre que se cumplan dos condiciones: que la señal tenga un ancho de banda limitado y que la frecuencia de muestreo sea superior al doble de la máxima frecuencia presente en la señal. El muestreo de la señal a una frecuencia mucho mayor de la mínima necesaria supone un incremento de coste sin obtener una mejora de los resultados.

La señal que va a ser digitalizada tiene una frecuencia de 50 Hz. La mínima frecuencia de muestreo necesaria para realizar el procesado sin pérdida de información será, por tanto:

Ecuación 3:

$$f_s = 2 \cdot f_{m\acute{a}x} = 2 \cdot 50 = 100\text{Hz}$$

Entonces, el tiempo de conversión máximo especificado para el convertidor que deba realizar la tarea será:

Ecuación 4:

$$T_c = \frac{1}{f}$$

Cuando un sistema de conversión A/D trabaja por encima de la frecuencia de muestreo establecida por el anterior teorema se dice que está funcionando en condiciones de sobre muestreo pero esto no produce más efectos negativos que el manejo de más información de la necesaria.

Por el contrario, si se trabaja por debajo del doble de la frecuencia de Nyquist, la señal no podrá reconstruirse de manera correcta, por lo que habrá pérdida de la información y conlleva otros problemas como el del aliasing, un fenómeno negativo que puede engañar al sistema digital provocando la lectura de una señal de frecuencia diferente a la muestreada. Esta señal reconstruida será de menor frecuencia que la original.

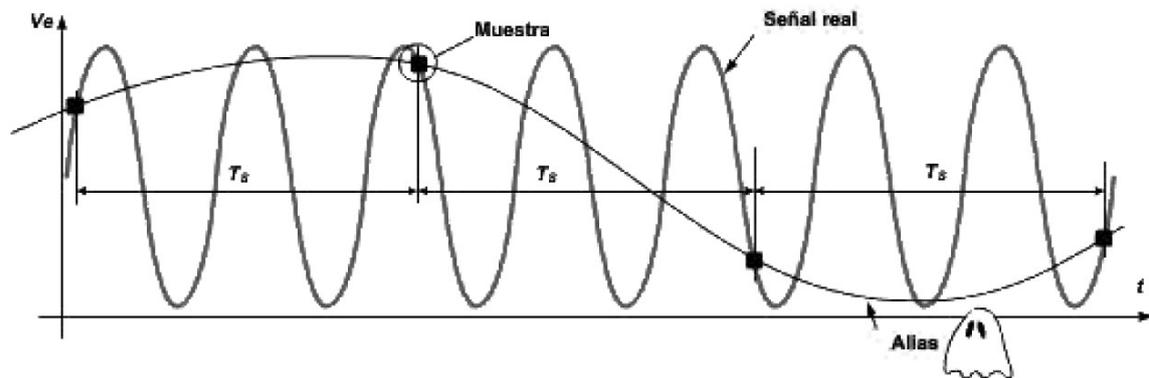


Fig. 8. Aparición de una señal fantasma por sub muestreo.

La forma de evitar este fenómeno de aliasing es utilizar siempre frecuencias de muestreo que cumplan con el teorema de Nyquist de manera que se evite la pérdida de información. Sin embargo, el problema se puede producir si se introduce una señal indeseada de más frecuencia de la prevista. En la práctica, la presencia del ruido puede ser una de las causas que produce tal problema.

A partir de los datos proporcionados por el Datasheet del micro controladores (en adelante, μC) montado en la placa Arduino Uno, el μC ATmega328P, se realiza un cálculo orientativo de la máxima frecuencia a la que es capaz de trabajar en modo de adquisición continua.

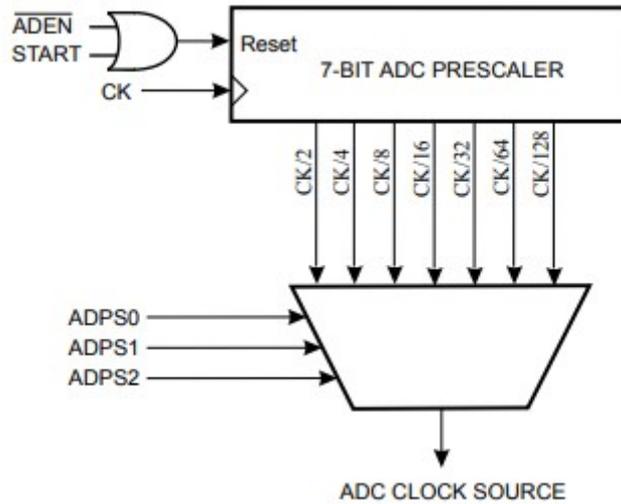


Fig. 9. Pre escalado y tiempo de conversión del ADC.

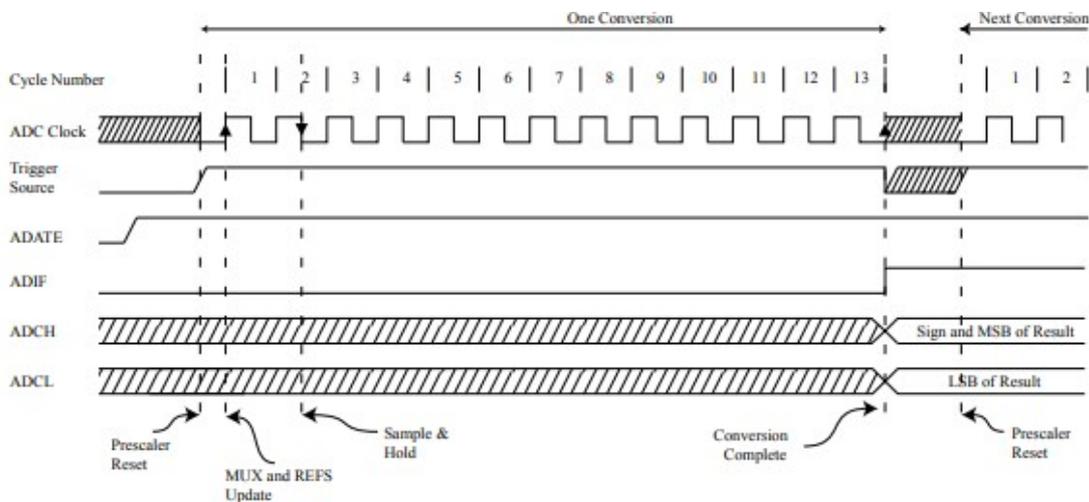


Fig. 10. Diagrama de tiempo y conversión auto trigger del ADC.

Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5

Tabla 1. Tiempo de conversión dl ADC.

Se puede observar en la gráfica 10 que en realizar una conversión el μC tarda 13 ciclos de ADC CLOCK SOURCE; teniendo en cuenta que el cristal del oscilador del Arduino es de 16Mz y este tiene predefinido una pre-escala de 128, se puede calcular ADC CLOCK SOURCE:

$$ADC\ CLOCK\ SOURCE = \frac{16000000}{128} = 125\ KHz \rightarrow F_c = \frac{125000}{13} = 9615\ Hz$$

Esta sería la máxima frecuencia de muestreo que se alcanzaría si el μC se dedicara únicamente a leer la entrada analógica.

Si la cantidad de instrucciones del μC aumenta, el tiempo de ejecución del programa aumenta. Y con ello, la frecuencia máxima de muestreo del convertidor puede verse reducida debido a que se deben realizar otras tareas que ralentizan la ejecución.

En el extracto de código que muestra en la Fig.11, se puede observar como la programación (LIFA_base), modifica el valor de la frecuencia de muestreo. Esta programación es necesaria cargarla en Arduino para mantener la comunicación con LabVIEW. En el momento que se elija una frecuencia de muestreo mayor que 1000 Hz se pierde dos dígitos de resolución, lo que supone una pérdida de información indeseada.

Teniendo en cuenta estos factores se ha llegado a la conclusión que el rango de frecuencia de muestreo debe de estar comprendida entre 100 y 1000 Hz.

```
void sampleContinuously()
{
  for(int i=0; i<iterations; i++)
  {
    retVal = analogRead(contAcqPin);
    if(contAcqSpeed>1000) //delay Microseconds is only accurate for values less that 16383
    {
      Serial.write( (retVal >> 2));
      delayMicroseconds(delayTime*1000000); //Delay to achieve desired sample rate
    }
    else
    {
      Serial.write( (retVal & 0xFF) );
      Serial.write( (retVal >> 8));
      delay(delayTime*1000);
    }
  }
}
```

Fig. 11. Programación de muestreo continuo en la librería de Interfaz LIFA.

2.5 Transformada de Fourier

La Transformada de Fourier es una operación matemática que Transforma una señal representada en el dominio del tiempo al dominio de la frecuencia pero sin alterar su contenido de información. La potencia del análisis de Fourier radica en que nos permite descomponer una señal compleja en un conjunto de componentes de frecuencia única; sin embargo, no nos indica el instante en que han ocurrido. Por ello, esta descomposición es útil para señales estacionarias: las componentes de las frecuencias que forman la señal compleja no cambian a lo largo del tiempo.

Como los cálculos para el análisis de la señal se realizan en un ordenador, se tendrá que trabajar con funciones discretas. Por este motivo, lo primero que se realiza es un muestreo de la señal para convertirla en finita, para ello se limita el número de puntos que se toman.

El cálculo directo de una Transformada Discreta de Fourier (DFT) para muestras grandes puede resultar demasiado lento y la capacidad de procesamiento disponible podría ser insuficiente, por lo que LabVIEW utiliza una transformada Rápida de Fourier; la cual consiste en un algoritmo para el cálculo eficiente de la DFT. Su importancia radica en el hecho que elimina una gran parte de los cálculos repetitivos a los que se ve sometida la DFT, por lo que se logra un cálculo más rápido a menor costo.

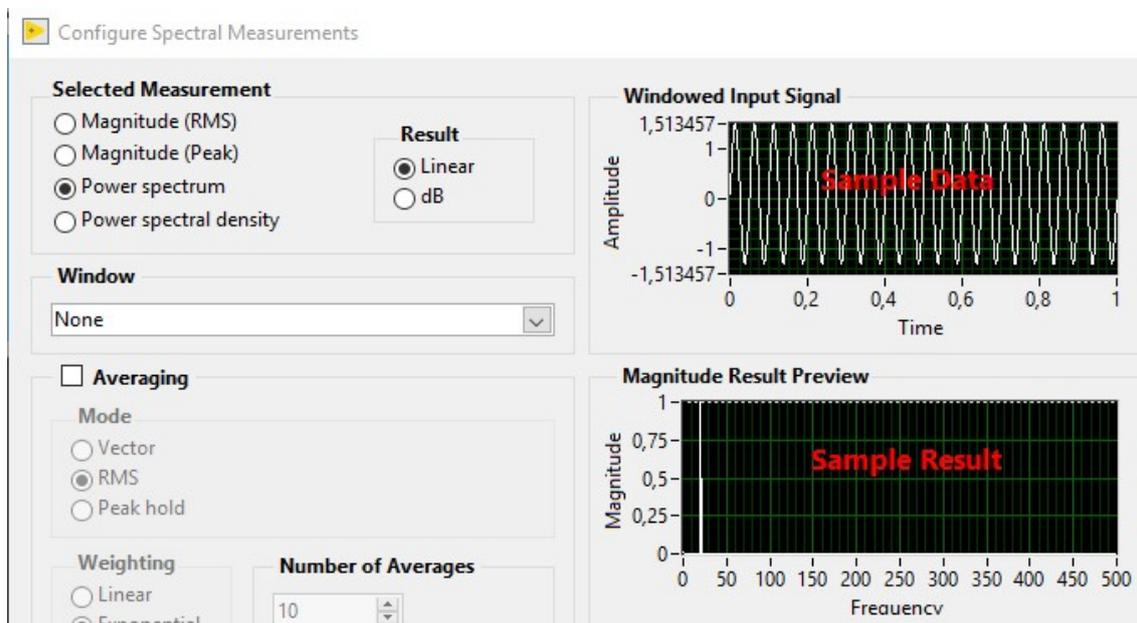


Fig. 12. Configuración del cálculo espectral en el panel de bloques de LabVIEW.



3. OBJETIVOS

El objetivo general del proyecto es el de diseñar y desarrollar un sistema de medición y monitoreo local de la potencia consumida por un dispositivo eléctrico, enfocado al uso doméstico.

Para alcanzar este objetivo principal, se han planteado los siguientes objetivos específicos:

- Diseñar un circuito acondicionador de medida de tensión e intensidad que se adapte a las características de las entradas analógicas de la placa de Arduino.
- Desarrollar una programación en LabVIEW que, interactuando con la placa de Arduino, realice el monitoreo, las mediciones, y el control de las señales objeto a análisis.
- Realizar la calibración del sistema para que se ajusten a las señales reales.

Una vez se hayan alcanzado estos objetivos mínimos, se tratará de lograr los siguientes objetivos adicionales como complemento:

- Cálculo del desfase entre la señal de tensión y de corriente que recibe el dispositivo analizado.
- Diseño de la placa PCB del circuito electrónico final.

Por último, con este proyecto se quieren adquirir las capacidades que permitan abordar futuros proyectos. Como por ejemplo, trabajar de manera constante y responsable, cumpliendo con los objetivos mínimos en el tiempo estipulado. Además de estas aptitudes, se aprenderá a realizar una búsqueda y manejo de la información de manera adecuada; familiarizándose así con un tipo programación gráfica como LabVIEW conforme avanza el proyecto.

4. BENEFICIOS QUE APORTA EL TRABAJO

Es un sistema que ayuda a saber que electrodomésticos consumen más, cuales son los hábitos del consumo del hogar analizado, ayudando a controlar el gasto y optimizando el uso de la energía eléctrica.

Por lo general se pueden destacar las siguientes ventajas del sistema:

- Ahorro de energía: Midiendo el consumo eléctrico se podrá detectar qué electrodomésticos consumen más y desconectarlos siempre que sea posible o utilizarlos en los tramos horarios en los que la luz sea más barata.
- Detectar problemas: Un medidor de consumo eléctrico nos va a permitir detectar si existe algún dispositivo que consume más de lo normal por un defecto del mismo.
- Comprobación de la red: comprueba la calidad de la energía suministrada por la empresa suministradora contratada.



5. DESCRIPCIÓN DE REQUERIMIENTOS

El Sistema de medición y monitorización a tiempo real enfocado al uso doméstico tiene unos requisitos previos establecidos.

El primero de estos requisitos es familiarizarse con conceptos básicos de la instrumentación electrónica como lo son el acondicionamiento de una señal, la resolución, calibración, variables muestreadas, conversión de señales variables analógicas a digitales, procesadores de señal, sistemas de adquisición de datos...

El segundo requisito es la minimización de costes. El sistema debe de ser lo más económico posible. Para eso se utiliza como unidad de adquisición, el μ C Arduino, que es una plataforma de bajo costo de electrónica de prototipos y LabVIEW como unidad de procesamiento. Para la comunicación entre los dispositivos esta la interfaz de LabVIEW para Arduino, "LIFA", que permite aprovechar la potencia del entorno de programación gráfica de LabVIEW para interactuar con Arduino.

El último requisito planteado, es que se diseñe un dispositivo de medición fiable con un error de medida bajo.

6. DISEÑO

La adquisición de datos es el proceso mediante el cual se recopilan datos de la magnitud que se desea medir a lo largo del tiempo. Para ello, habitualmente, se usan sistemas de procesamientos (como PCs, μC ,...). Un sistema de adquisición de datos dispone de sensores, hardware de medidas y software para procesar los datos obtenidos; estos sistemas aprovechan la potencia de procesamiento, la productividad, la visualización, y las habilidades de conectividad de las PCs estándares, proporcionando una solución de medidas más potente, flexible y rentable. En la siguiente imagen se puede observar el proceso básico para la adquisición de datos:

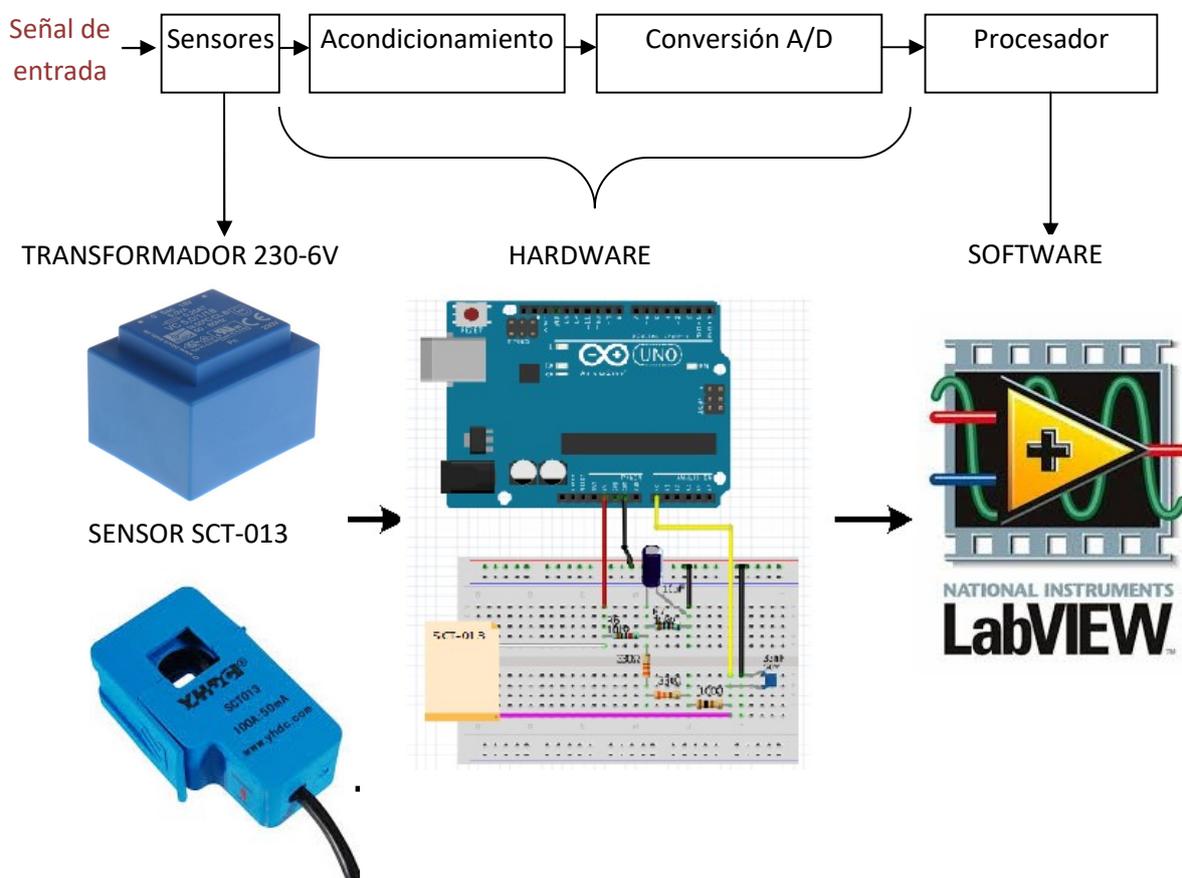


Fig. 13. Sistema de adquisición del proyecto.

6.1 Planteamiento del sistema

Basándonos en la arquitectura antes descrita, el proyecto se dividirá en tres apartados.

1. Unidad de adquisición: en este apartado se mencionarán las características propias del Arduino empleado con este fin. Este apartado es el primero en desarrollarse debido a que condiciona al resto de elementos del sistema.
2. Acondicionamiento de sensores: el objetivo de este apartado es mostrar los sensores utilizados y realizar los cálculos para desarrollar el circuito acondicionador necesario.
3. Unidad de proceso: Se describe en este apartado la programación del entorno gráfico utilizada en el proyecto. A través de este, se pretende conseguir una representación gráfica que muestre los resultados de manera clara.

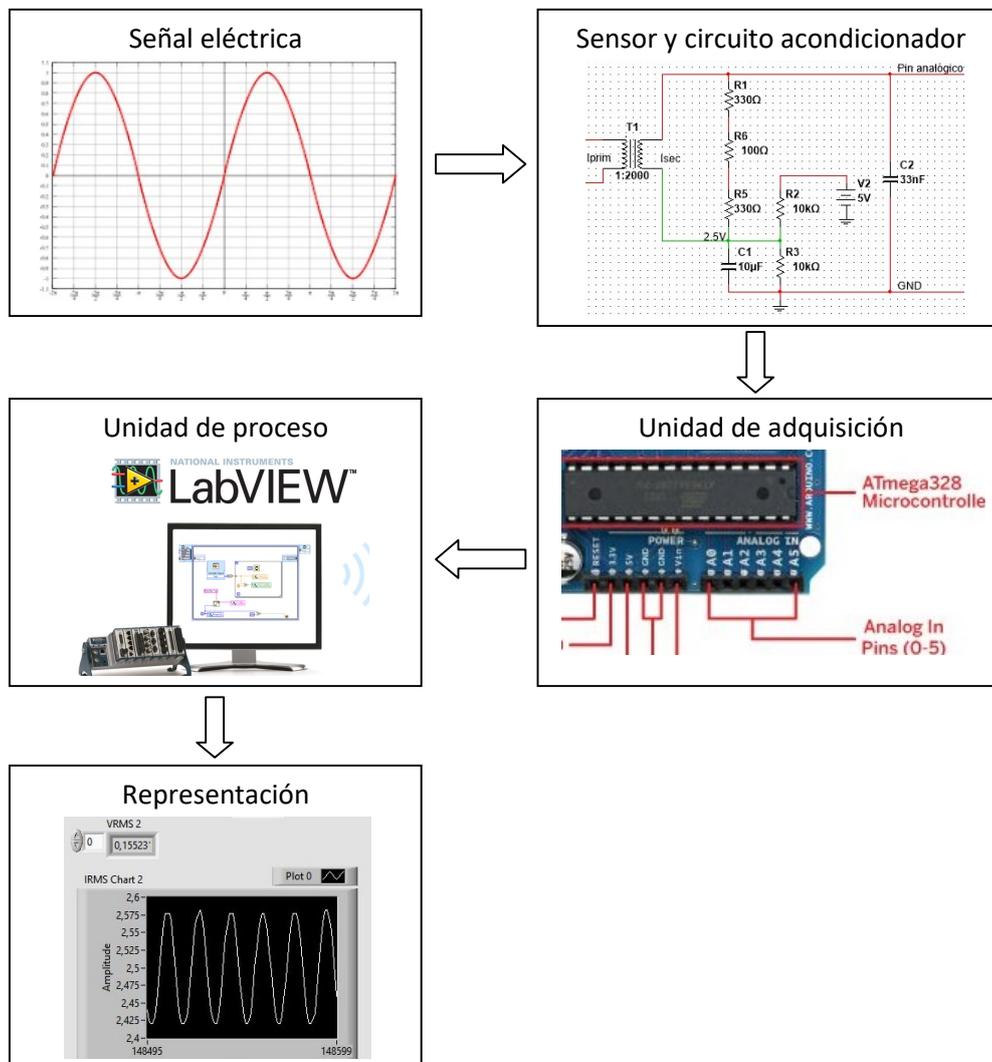


Fig. 14. Arquitectura del proyecto.



Además de los apartados ya descritos, que corresponden a la estructura básica, se verán otros puntos como pueden ser, un apartado destinado a las comunicaciones del sistema, una parte destinada a la programación y una vez diseñado el sistema su implementación sobre una placa de circuito impreso.

6.2 Unidad de adquisición

La unidad de adquisición es la encargada de recibir la señal y enviarla a la unidad de procesamiento (PC). El elemento más importante de la unidad de adquisición es el ADC, conversor analógico digital. El ADC será el elemento que nos permitirá transformar la señal analógica procedente del sensor a una medida digital que pueda ser procesada por el PC.

Debido a que LabVIEW es mucho más potente haciendo cálculos que lo que puede llegar a ser Arduino, será LabVIEW quien se encargue del análisis de datos quedando la conversión analógico/digital en manos de Arduino.

6.2.1 Elijiendo la plataforma

La unidad de adquisición es una parte crítica en el sistema analógico porque nos determina el acondicionamiento de la señal generada por los transductores. Según el ADC elegido y el μ C seleccionado, surgen unas limitaciones de diseño (niveles de resolución, niveles de voltaje de entrada, velocidad de procesamiento...).

La plataforma de desarrollo, Arduino, elegida para este proyecto, nos proporciona además del μ C otros elementos, por ejemplo, un método de comunicación entre el microprocesador y el ordenador basado en comunicación USB, esto permite el poder realizar una comunicación directa entre la placa Arduino Uno empleada y LabVIEW. Además, gracias a que National Instruments creó la interfaz LIFA (LabVIEW Interface For Arduino) se puede trabajar con la placa de Arduino desde LabVIEW.

El Arduino UNO utiliza el μ C ATmega328P-PU, cuyas características más importantes se representan en la tabla 2.

Microcontrolador	ATmega 328P-PU	Memoria EEPROM	1 KB
CPU	8 bits	Pines Digitales	14
Frecuencia Reloj	16 MHz	Pines PWM	6
Voltaje Operación	5V	Pines Analógicos	6
Memoria FLASH	32 KB	Resolución ADC	10 bits
Memoria SRAM	2 KB	Intensidad salida de Pines	40 A

Tabla 2. Características del Arduino UNO



6.2.2 Introducción a Arduino

Arduino es una plataforma de hardware y software libre basada en μC de la marca Atmel®. En los últimos años esta plataforma se ha ganado muchos seguidores debido a un lenguaje de programación de alto nivel basado en C, que permite la programación del μC de una manera muy sencilla. Permite a un usuario con unos conocimientos básicos en programación programar el μC integrado en la plataforma y así realizar una multitud de proyectos que van desde encender un led a introducirse en el mundo de la robótica, ó incluso realizar proyectos más avanzados basados en comunicaciones inalámbricas ó Ethernet.

Arduino nos proporciona todas las herramientas necesarias para programar el μC , desde los drivers que necesita el ordenador hasta el entorno de programación necesario para empezar a realizar proyectos. Además de todo esto, Arduino tiene dos grandes ventajas, la primera es que existe una comunidad muy numerosa que da soporte a esta plataforma, y segundo existen módulos de ampliación que permiten incorporar a la placa utilidades adicionales (comunicación WIFI, control de motores, etc.).

Para más detalle sobre esta plataforma se puede consultar la página web de la comunidad: www.arduino.cc.

6.2.3 Arquitectura de Arduino

Para conocer mejor las posibilidades que nos ofrece Arduino se debe conocer un poco más su arquitectura, esto significa entender las entradas y salidas a utilizar en el proyecto. Una de las ventajas que ofrece Arduino, es que todos los pines a utilizar están identificados, facilitando así su programación.

Primero de todo hay dos grandes grupos de pines, los pines digitales y los pines analógicos.

Los pines analógicos son claves en la realización del proyecto ya que son los que están conectados al convertor ADC. La placa Arduino Uno tiene 6 pines analógicos identificados desde el A0 hasta el A5. Cada uno de estos pines es capaz de digitalizar una señal de entrada entre los 0V y 5V, convirtiendo el voltaje de entrada en un valor numérico que va desde el cero hasta el 1023, este rango de valores corresponde a la resolución del ADC que es de 10bits.

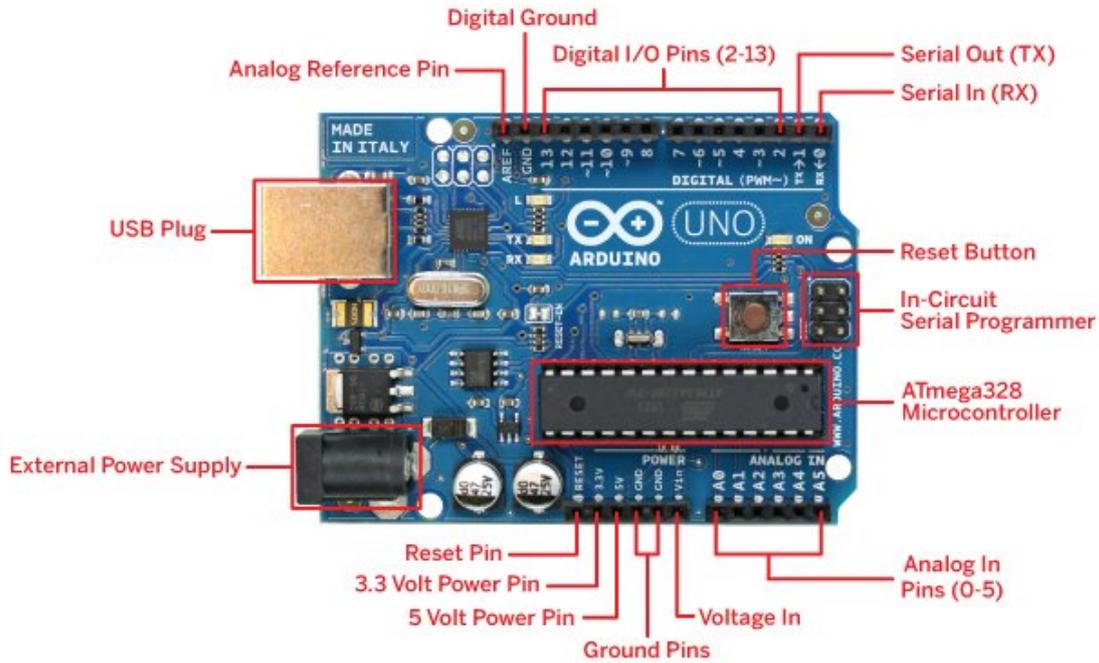


Fig. 15. Placa Arduino UNO.

6.2.4 ADC

Al principio de este capítulo se habló sobre la importancia del ADC. En este apartado se verán las características a tener en cuenta en el proyecto y que servirán de referencia en capítulos posteriores.

En la tabla 3 se pueden apreciar las características principales del convertor analógico digital. De entre todas las especificaciones destacan dos, la resolución y el rango de voltaje.

Parámetro	Valor
Resolución	10 bits
Rango de Voltaje	0 a VCC.
Tiempo de conversión	65 a 260 μ s
Canales	8
Incertidumbre Máxima	1 LSB

Tabla 3. Características convertor analógico.

El ADC incorporado en la placa Arduino Uno tiene una resolución de 10 bits, lo que determina la variación mínima o LSB que es capaz de distinguir el convertidor. Teniendo en cuenta que el convertidor esta alimentado a 5V, a continuación se realizará el cálculo de esta variación mínima:

Ecuación, 5:

$$LSB = \frac{5}{2^{10}} = \frac{5}{1024} = 0.0049 \text{ V}$$

Por lo tanto, el convertidor ADC no será capaz de reconocer variaciones de tensión menores que 0.0049V.

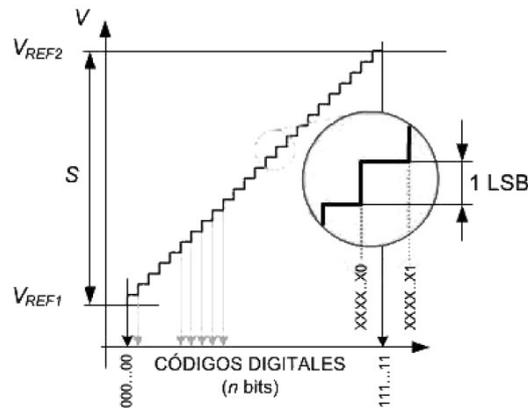


fig. 16. Escalón mínimo ó LSB.

6.3 Acondicionamiento de sensores

Una vez visto el capítulo de la unidad de adquisición, ya se pueden seleccionar y adaptar los transductores al sistema.

Definición de sensor: Un sensor, también llamado transductor, es un dispositivo capaz de transformar una magnitud física a una magnitud eléctrica, normalmente una señal analógica, aunque no necesariamente, ya que existen sensores que ofrecen una salida digital.

Cuando se habla de señal analógica, se hace referencia a una señal con dos componentes: amplitud y tiempo, básicamente se tiene, una determinada amplitud en un tiempo definido.

6.3.1 Sensor SCT-013-000

Para la medición de la corriente que circula por el circuito se va a usar como transductor de corriente un transformador de corriente. Un transformador de corriente es similar a un transformador de tensión y está basado en los mismos principios de funcionamiento (de hecho, formalmente es idéntico). Sin embargo persiguen objetivos diferentes y, en consecuencia, están diseñados y contruidos de forma distinta.

En el proyecto, el sensor de corriente elegido es el SCT-013-000, el cual genera una intensidad en el bobinado secundario proporcional a la intensidad que atraviesa su bobinado primario.

Su característica más relevante es el ser un sensor no invasivo, cuyo núcleo ferro magnético puede abrirse y arrollar un conductor como se puede ver en la figura 18.

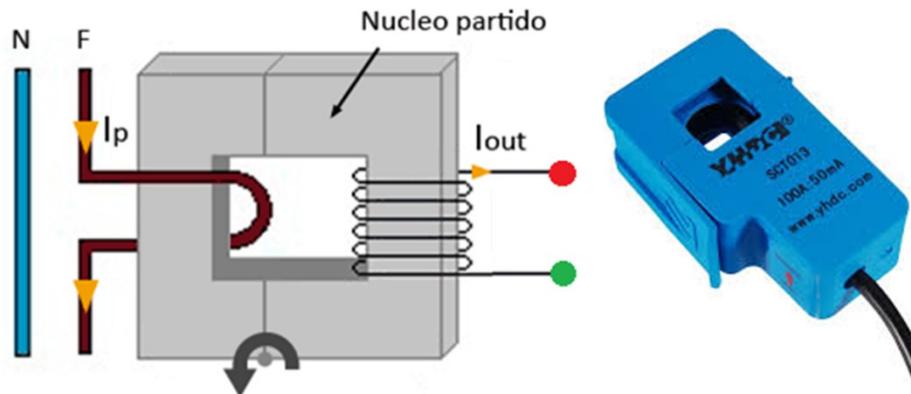


Fig. 17. Sensor SCT-013-000.

De esta forma, se tiene un transformador con la siguiente estructura:

- El cable por el que circula la intensidad a medir constituye un devanado primario.
- La “pinza” es el núcleo magnético.
- El devanado secundario está integrado como parte de la sonda.

Cuando la corriente alterna circula por el conductor que constituye el devanado primario se genera un flujo magnético en el núcleo ferro magnético, que a su vez genera una corriente eléctrica proporcional en el devanado secundario.

La relación de transformación de intensidad depende de la relación entre el número de espiras.

Ecuación, 6:

$$\frac{N_{\text{primario}}}{N_{\text{secundario}}} = \frac{I_{\text{secundario}}}{I_{\text{primario}}} = \frac{V_{\text{primario}}}{V_{\text{secundario}}}$$

El primario generalmente está formado por una única espira formada por el conductor a medir. Aunque es posible enrollar el conductor haciendo que pase por el interior de la “pinza” varias veces, modificando la relación de transformación. El número de espiras del secundario, integrado en la sonda, varía 1000-2000 según modelos del SCT-013.

El modelo SCT-013-000 tiene un bobinado secundario de unas 2000 espiras aproximadamente y en la tabla 1.3 se puede apreciar sus parámetros técnicos.

Modelo	SCT-013-000
Corriente de entrada	0-100 A
Corriente de salida	0-50 A

Tabla 4. Parámetros Técnicos del sensor SCT-013-000.

- Acondicionamiento

El objetivo de este apartado es adaptar las características de la señal entregada por el sensor de corriente a las especificaciones de las entradas analógicas que dispone Arduino.

El sensor utilizado genera una señal sinusoidal de corriente alterna cuyo rango variación es de $\pm 50\text{mA}$ y la entrada analógica del Arduino tiene un margen de funcionamiento de 0 a 5 V. Por lo tanto, resulta evidente realizar una etapa de acondicionamiento de la señal que transforme la corriente en tensión.

El acondicionamiento se realizará convirtiendo la corriente que circula por el devanado secundario en tensión a través de una resistencia. Con el fin de conseguir la máxima resolución posible, se harán unos cálculos de consumo con una potencia teórica que sería la más alta que el sistema podría medir.

Se ha elegido medir una potencia máxima de 1000W. Como se trata de un sistema monofásico, el voltaje suministrado por la red es de 220V. Sabiendo la fórmula de la potencia, se calcula la corriente eficaz que circula por devanado primario del sensor:

Corriente eficaz:

$$P = V \cdot I_{\text{RMS}} \rightarrow I_{\text{RMS}} = \frac{P}{V} = \frac{1000}{220} = 4.54\text{A}$$

Corriente máxima:

$$I_{\text{RMS}} = \frac{I_{\text{máx}}}{\sqrt{2}} \rightarrow I_{\text{máx}} = I_{\text{RMS}} \cdot \sqrt{2} = 6.42\text{ A}$$

Con la fórmula de relación de transformación vista antes, se puede calcular la corriente de pico y a su vez la corriente eficaz que circula por el devanado secundario:

Corriente máxima:

$$\frac{N_{\text{primario}}}{N_{\text{secundario}}} = \frac{I_{\text{secundario}}}{I_{\text{primario}}} \rightarrow \frac{1}{2000} = \frac{I_{\text{secundario}}}{6.42} \rightarrow I_{\text{secundario}} = 0.00321\text{ A}$$

Corriente eficaz:

$$I_{\text{RMS}} = \frac{I_{\text{máx}}}{\sqrt{2}} = \frac{0.00321}{\sqrt{2}} = 0.00227 \text{ A}$$

Para maximizar la resolución con el máximo valor de la entrada analógica es necesario tener en cuenta el voltaje máximo y mínimo que es capaz de medir el Arduino utilizado. La referencia interna "AREF" del Arduino es de 5V como máximo.

Sabiendo que en la salida del devanado secundario se obtendrá una onda sinusoidal que varía de positivo a negativo y cuyos valores máximos son ± 0.00321 A es necesario incluir una resistencia de carga cuyo objetivo es que se tenga un voltaje entre 2.5 V y -2.5 V.

$$R_{\text{carga}} = \frac{\frac{5}{2}}{I_{\text{máx}}} = \frac{\frac{5}{2}}{0.00321} = 778 \Omega$$

Debido a que no existe una resistencia de 778 ohmios se ha elegido unas resistencias que se aproximan al valor deseado, cuyo valor es de 760 Ω , por lo que es necesario realizar nuevamente los cálculos para actualizar los valores máximos capaz de medir el sistema.

Por otro lado, como se ha dicho anteriormente, las entradas analógicas del Arduino no permiten valores negativos siendo necesario añadir un offset que desplace la señal 2.5 V, generando una señal variable entre 0 y 5V en la entrada analógica. Para generar este offset se empleará un divisor de tensión alimentado con la tensión continua proporcionada por el Arduino.

Por último, se añaden dos condensadores uno a la entrada del Arduino para evitar que un pico de tensión muy alto entre al Arduino variando de forma no deseada las mediciones realizadas por este, y otro condensador con el divisor de tensión que hace que la impedancia sea baja en corriente alterna por lo cual la corriente alterna evita la resistencia con el objetivo de mantener una tensión de offset estable y puramente continua. En la siguiente gráfica se puede observar el esquema del diseño resultante:

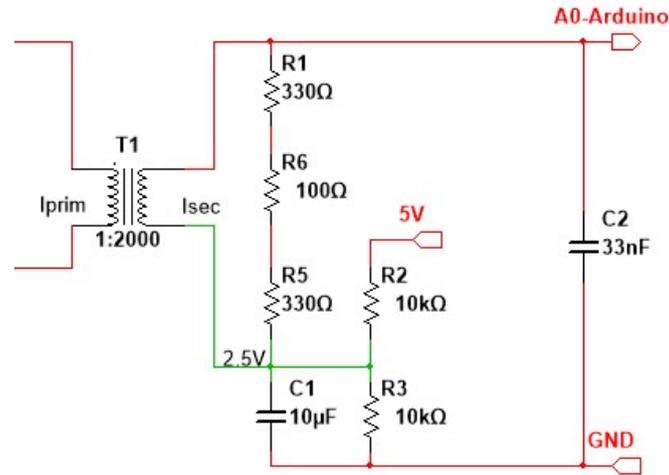


Fig. 18. Esquemático Circuito de acondicionamiento de la señal de intensidad.

Al utilizar un valor de resistencia diferente al valor teórico, se ha modificado el valor de la intensidad que el Arduino es capaz de medir, por lo que es necesario volver a realizar los cálculos.

Cuando la tensión en la entrada analógica del Arduino es la máxima la intensidad máxima en el devanado secundario es la siguiente:

$$V_{A0} = 5V$$

$$I_{\text{máxsec}} = \frac{5 - 2.5}{760} \Rightarrow I_{\text{Máxsecun}} = 3.289 \cdot 10^{-3}A$$

Generándose así una señal alterna cuyos valores oscilan entre $\pm 3.289 \cdot 10^{-3}A$.

La potencia que en un principio era de 1000w, también se verá afectada por la modificación de la resistencia, por lo que es necesario cambiar las especificaciones del diseño, siendo la siguiente potencia la que el diseño será capaz de medir:

$$\frac{N_{\text{primario}}}{N_{\text{secundario}}} = \frac{I_{\text{secundario}}}{I_{\text{primario}}} \rightarrow I_{\text{Máxprimario}} = \frac{I_{\text{Máxsecundario}} \cdot N_{\text{secundario}}}{I_{\text{primario}}}$$

$$I_{\text{Máxprimario}} = \frac{3.289 \cdot 10^{-3} \cdot 2000}{1} = 6.579A$$

$$I_{\text{Eficazprimario}} = \frac{6.579}{\sqrt{2}} = 4.65 A$$

$$P = V \cdot I_{\text{RMS}} = 220 \cdot 4.65 = 1023W$$

Como el valor de potencia supera al valor elegido para el proyecto, se puede dejar este valor como margen de seguridad. Aun así, es recomendable no superar la intensidad eficaz límite que el dispositivo es capaz de medir de 4.65 A, ya que podría llegar a dañar el equipo de medida.

- Calibración

Es necesario realizar la calibración del sistema, para poder establecer, con la mayor exactitud posible las mediciones del instrumento de medida y los valores reales de la magnitud que se mide con él.

El método de calibración utilizado es el de la calibración punto a punto; la cual consiste en actuar sobre el sistema de medida de forma que para un punto concreto la lectura sea lo más exacta posible.

La figura 20 es una representación de la curva de calibración ideal y la que corresponde a dos instrumentos de medida. Sobre estos instrumentos se actuará para que su valor de lectura sea lo más cercano a la curva ideal.

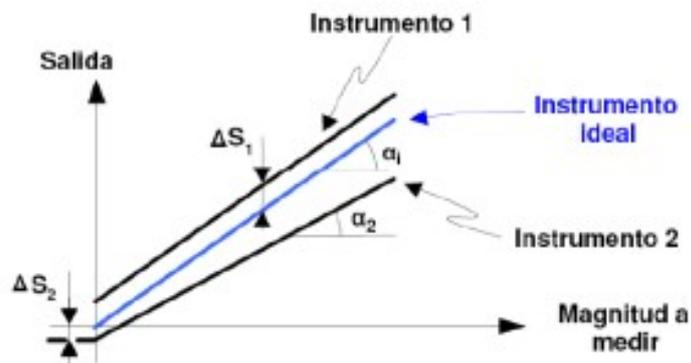


Fig. 19. Representación de la curva de calibración ideal y la que corresponde a dos instrumentos de medida.

Para establecer una curva de calibración lo primero que se ha realizado ha sido comparar la salida del ADC con un instrumento patrón de mejor resolución.

En la calibración del dispositivo se ha utilizado:

- Un resistor de 10Ω que es capaz de soportar como máximo una intensidad de 4 A.
- Por otro lado un autotransformador para variar la intensidad que circula por la resistencia generando una tensión sinusoidal variable de 0 a 220V. Debido a la limitación de intensidad del resistor, a tensión máxima generada por el autotransformador es de 40V.
- Amperímetro con alta precisión para comparar los datos medidos con los calculados realizados a través de LabVIEW.

Para obtener la curva de calibración existen muchos métodos y sistemas, el más utilizado ó al menos el más conocido, es el método de mínimos cuadrados. El método de mínimos cuadrados, es una aproximación matemática a una curva conocida. Para utilizar este método

simplemente basta con usar un programa de cálculo para calcular el parámetro de la curva de calibración. La herramienta de cálculo más sencilla para ello es usar una hoja de Excel.

Al realizar las comparaciones entre las mediciones del sistema y las del instrumento ideal, se pudo observar que el error en el sistema aumentaba de forma lineal, por lo que se ha decidido ajustar el dispositivo multiplicando por un factor de corrección de $k = 1.86$. de esta manera se ha reducido el error de la medición y con la que se obtiene una mayor exactitud cómo se puede observar en la figura 20.

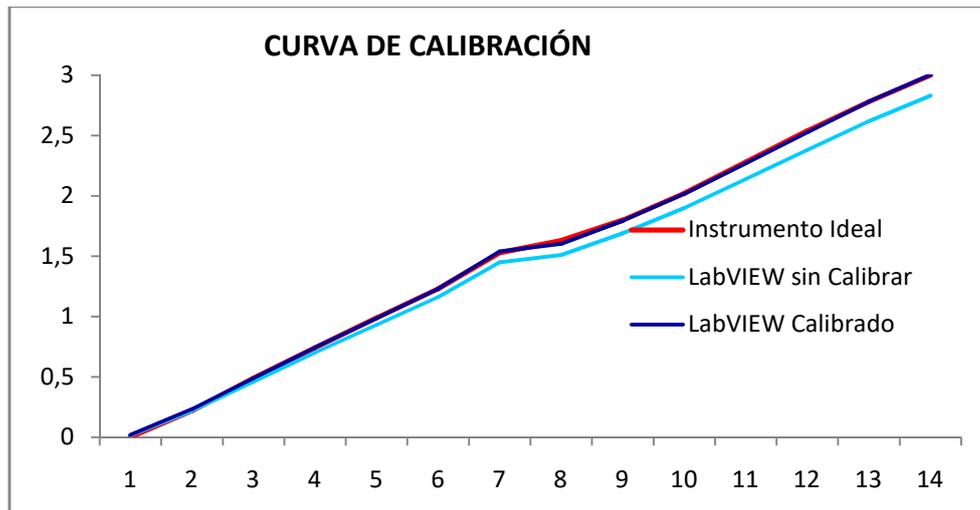


Fig. 20. Curva de calibración de la intensidad



Instrumento Ideal	LabVIEW sin Calibrar	LabVIEW Calibrado
0,001	0,017	0,018037
0,22	0,22	0,23342
0,49	0,46	0,48806
0,74	0,7	0,7427
0,99	0,93	0,98673
1,23	1,16	1,23076
1,53	1,45	1,53845
1,63	1,51	1,60211
1,8	1,69	1,79309
2,02	1,9	2,0159
2,28	2,14	2,27054
2,54	2,38	2,52518
2,78	2,62	2,77982
3	2,83	3,00263
3,27	3,08	3,26788
3,55	3,34	3,54374
3,74	3,53	3,74533
4,14	3,9	4,1379
4,41	4,16	4,41376
4,73	4,49	4,76389

Tabla 5. Comparación entre instrumento real, LabVIEW sin calibrar y LabVIEW calibrado.

6.3.2 Transformador

Un transformador es una máquina eléctrica de corriente alterna que emplea campos magnéticos para adaptar niveles de tensión y corriente entre dos circuitos. Se puede usar transformadores para adaptar la tensión de circuitos a un rango manejable para la electrónica y procesadores como Arduino.

Además de suministrar potencia eléctrica, los transformadores pueden, como es el caso del proyecto, ser empleados como elementos de instrumentación. En este proyecto, unos de los principales objetivos es medir la tensión de red eléctrica. Teniendo en cuenta que la red eléctrica opera a 230 VAC y 50 Hz, estos valores de tensión destruirían inmediatamente el Arduino.

Al usar un transformador no es necesario un opto-acoplador ya que el transformador supone un aislamiento galvánico en sí mismo. No obstante, existe un acoplamiento magnético, por lo que el secundario sigue siendo susceptible a grandes picos de corriente o sobrecargas en el circuito primario. Por este motivo, es necesario incorporar un condensador en el secundario.

En general, un transformador consta de dos devanados (es decir, bobinas de conductor) correspondientes con el circuito primario y secundario. En la mayoría de los casos los devanados se encuentran arrollados entorno a un núcleo ferro magnético.

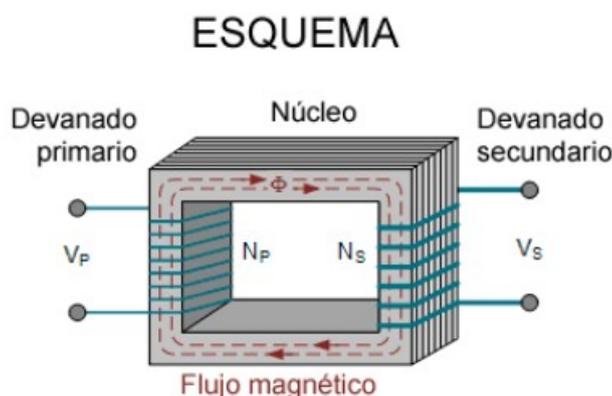


Fig. 21. Transformador

Cuando se hace circular corriente alterna por el devanado primario esta induce un flujo magnético en su interior. Este flujo magnético atraviesa el devanado secundario, generando a su vez una tensión inducida (o fuerza electromotriz).

El flujo generado por el devanado primario es proporcional a su número de espiras. A su vez, la tensión inducida en el devanado secundario también es proporcional a su número de espiras. Por tanto, la relación entre ambas tensiones es proporcional a la relación entre espiras de ambos devanados.

Suponiendo un transformador ideal que carezca de pérdidas, la potencia eléctrica se transmite íntegramente entre el devanado primario y secundario. Por tanto, es sencillo deducir que la relación de intensidades es inversa a la relación entre tensiones y, por tanto, a la relación entre las espiras de devanados.

Ecuación, 7:

$$\frac{V_p}{V_s} = \frac{I_s}{I_p} = \frac{N_p}{N_s}$$

Los transformadores reales tienen pérdidas y limitaciones que lo alejan de su comportamiento ideal. No obstante, en general, los transformadores son máquinas con elevados rendimientos, típicamente del 95-98%. Las principales pérdidas que alejan al transformador de su comportamiento ideal son pérdidas en el cobre por efecto Joule, pérdidas por corrientes parásitas inducidas en el núcleo (corrientes de Foucault), pérdidas por dispersión del campo magnético y pérdidas por histéresis en el material del núcleo.

Los transformadores preservan la frecuencia de la corriente alterna en ambos lados. Sin embargo, dado que las bobinas que componen el transformador son cargas reactivas, el transformador puede tener un desfase angular entre el circuito primario y secundario.



Por último, los transformadores generan distorsiones en la señal transmitida, debido a la falta de linealidad del material del núcleo especialmente en las zonas de saturación, lo que se traduce en la introducción de componentes armónicos en la salida.

- Acondicionamiento

Para realizar la medición de la tensión que alimenta el circuito se va a emplear un transformador de 230VAC/6VAC.

Como en el sensor de intensidad, la salida del transformador es una señal alterna, cuyo valores no está dentro del rango de las entradas analógicas (0 a +5V) del Arduino. De introducirse esta señal de manera directa al pin analógico del Arduino se dañaría el dispositivo de adquisición, por lo que debe haber un circuito de acondicionamiento.

Sabiendo que se realizará la medición de tensión de una red eléctrica de 230VAC a 50Hz:

$$V_{RMS} = \frac{V_{Max}}{\sqrt{2}} \rightarrow V_{RMS} \cdot \sqrt{2} = V_{pMax} = 230 \cdot \sqrt{2} = 325.27 \text{ V}$$

Y, teniendo en cuenta que la relación de transformación del transformador es de $Rt = \frac{230}{6} = 38.33$, la tensión de salida del devanado secundario quedará de la siguiente manera:

$$\frac{V_p}{V_s} = 38.33 \rightarrow V_{sMáx} = \frac{325.27}{38.33} = \pm 8.486 \text{ V}$$

Es necesario adaptar este valor de tensión a las características de las entradas del Arduino, para adquirir la señal de manera correcta, la cual tiene un rango de entrada límite de 0 a 5V. Por un lado, debido a que la amplitud de la señal pico a pico es de aproximadamente 17V, se utilizará un divisor de tensión para reducir su amplitud a 5V. Por otro lado, el sensor genera señales negativas de tensión siendo necesaria una tensión de offset, realizada a través de otro divisor de tensión, que ajuste la señal para que no tenga valores negativos.

En el circuito de acondicionamiento de la señal de intensidad se ha utilizado un divisor de tensión, el cual se puede aprovechar para esta parte del circuito acondicionador de tensión. En la siguiente gráfica se puede observar el diseño del circuito sin el valor de las resistencias utilizadas, ya que sus valores serán calculados a continuación.

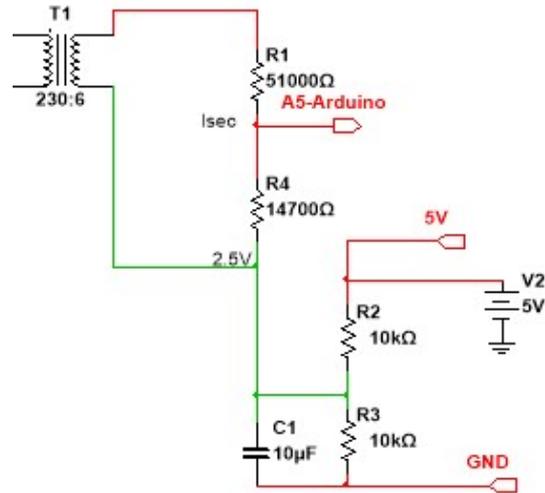


Fig. 22. Esquemático de circuito acondicionador de tensión.

Los cálculos necesarios para saber el valor de las resistencias R1 y R4 se determinan a partir del divisor de tensión y los valores límites que genera el transformador.

$$\frac{(8.486 + 2.5) - 2.5}{R1 + R2} \cdot R2 + 2.5 = 5 \rightarrow \frac{R2}{R1 + R2} = \frac{5 - 2.5}{8.486} = 0.294$$

$$\frac{(-8.486 + 2.5) - 2.5}{R1 + R2} \cdot R2 + 2.5 = 0 \rightarrow \frac{R2}{R1 + R2} = \frac{0 - 2.5}{-8.486} = 0.294$$

$$R1 \gg R4$$

Los valores elegidos de las resistencias, que cumplen con un margen de seguridad en caso de sobretensiones, han sido:

$$R1 = 51K\Omega$$

$$R4 = 14,7K\Omega$$

Con estos valores de resistencias la señal de entrada que será recibida por el Arduino es de:

$$\frac{(8.486 + 2.5) - 2.5}{51000 + 14700} \cdot 14700 + 2.5 = VA5 = 4.398V$$

$$\frac{(-8.486 + 2.5) - 2.5}{51000 + 14700} \cdot 14700 + 2.5 = VA5 = 0.60V$$

El circuito resultante es:

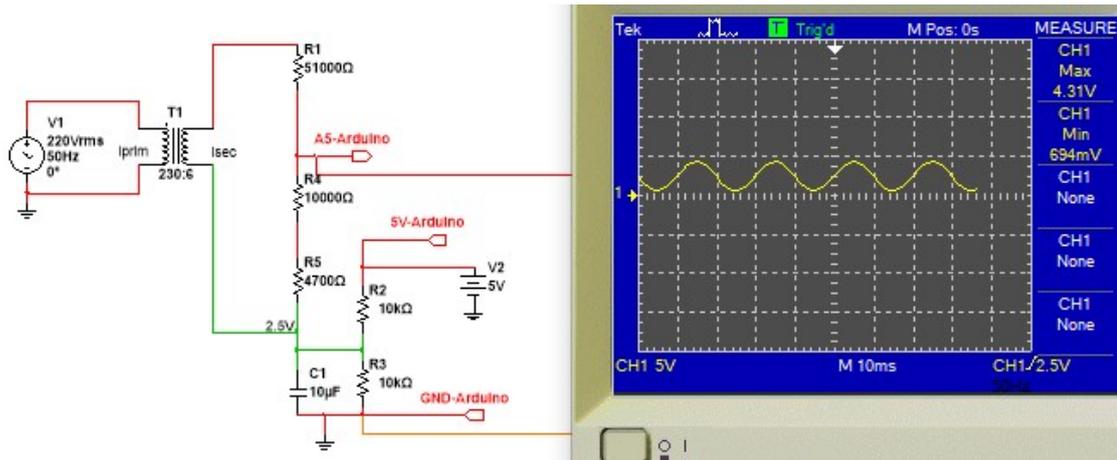


Fig. 23. Circuito acondicionador de tensión.

Una vez teniendo el circuito acondicionador, la fórmula necesaria para saber la tensión que recibe el transformador de la red es la siguiente:

$$\frac{(V_{S_{Máx}} + 2.5) - 2.5}{51000 + 14700} \cdot 14700 + 2.5 = VA5$$

$$V_{RMS} \cdot \sqrt{2} = V_{Max}$$

$$V_{S_{RMS}} = (VA5 - 2.5) \cdot \frac{51000 + 14700}{14700 \cdot \sqrt{2}}$$

$$\frac{V_{p_{RMS}}}{V_{S_{RMS}}} = 38.33 \rightarrow V_{p_{RMS}} = V_{S_{RMS}} \cdot 38.33$$

$$V_{p_{RMS}} = (VA5 - 2.5) \cdot \frac{51000 + 14700}{14700 \cdot \sqrt{2}} \cdot 38.33 = 121 \cdot (VA5 - 2.5)$$

- Calibración

El objetivo de este apartado es el de comparar los valores de tensión eficaz de la señal a medir obtenidos por el instrumento de medición diseñado con la medida correspondiente de un instrumento patrón. el objetivo es obtener un factor de escala que, al multiplicarlo por el valor medido a través del circuito diseñado se tenga un valor lo más exacto posible al valor medido por el instrumento patrón. El instrumento de referencia utilizado debido a su buena resolución es el osciloscopio.

En la siguiente figura se puede observar el sistema de calibración utilizado en la práctica a través del diseño en Multisim:

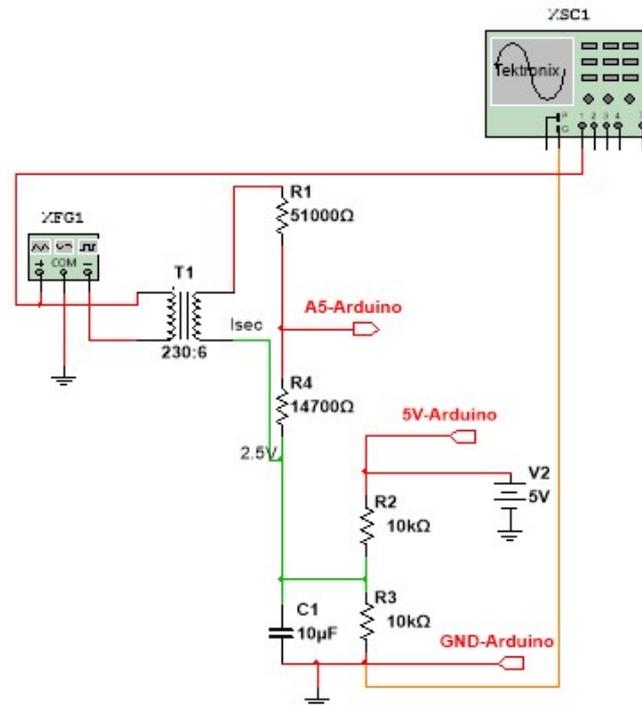


Fig. 24. Circuito para la calibración del sistema.

Este circuito consta, además del circuito del acondicionador de tensión a calibrar, de un generador de funciones capaz de generar como máximo una señal sinusoidal de 20 voltios de pico a pico y un osciloscopio como instrumento de referencia. Este instrumento es necesario ya que el generador de funciones tiene un error de medida alto.

Para calcular el factor de corrección teórico se ha realizado un programa en Arduino que realiza el siguiente cálculo para obtener la tensión eficaz:

$$V_{p_{RMS}} = (VA5 - 2.5) \cdot \frac{51000 + 14700}{14700 \cdot \sqrt{2}} \cdot 38.33 = 121 \cdot (VA5 - 2.5)$$

Al generar una señal sinusoidal de 50Hz y 10V de amplitud, cuya tensión eficaz es de 7.05V la tensión eficaz resultante en el programa de Arduino es de 9.1769V por lo cual, es necesario el siguiente factor de corrección:

$$V_{p_{RMS}Diseño} \cdot k = V_{p_{RMS}Real} \rightarrow k = \frac{V_{p_{RMS}Real}}{V_{p_{RMS}Diseño}} = \frac{7.05}{9.1769} = 0.768$$

Quedando la ecuación de la tensión eficaz de la siguiente manera:

Ecuación, 8:

$$V_{p_{RMS}} = 121 \cdot k \cdot (VA5 - 2.5) = 92.9 \cdot (VA5 - 2.5)$$

Por último, se ejecutara el programa de LabVIEW con la ecuación 8, antes descrita para comprobar si el factor de corrección teórico es el ideal o es necesario ajustarlo aún más para conseguir una curva de calibración que se asemeje a la del instrumento de referencia.

En la siguiente tabla se pueden observar los valores obtenidos de tensión eficaz de los diferentes elementos utilizados para cuatro señales sinusoidales, de 50 HZ, generadas con el generador de funciones:

Funtion Generator (Vpp)	Multisim (V)	LabVIEW Calibrado (V)	Osciloscopio (V)	Arduino Calibrado (V)
19.5	7.05	7.32	7.06	7.0571
17.6	6.32	6.7	6.30	6.1465
15.6	5.61	5.8	5.57	5.6912
11.4	4.23	4.07	4.08	4.01

Tabla 6. Comparación entre los valores calculados por diferentes dispositivos.

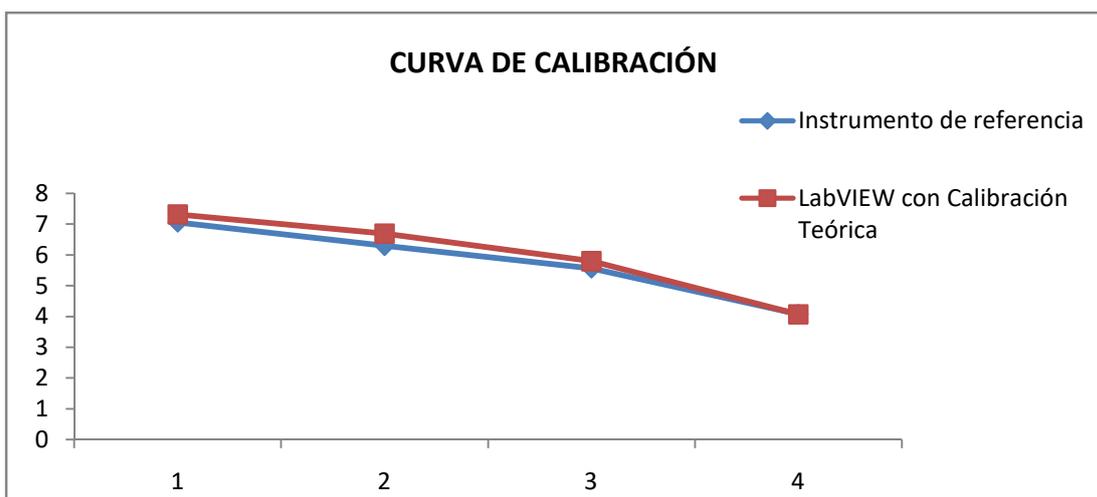


Fig. 25 Curva de calibración del instrumento ideal y el instrumento diseñado calibrado.

En la curva de calibración se puede observar que el factor de corrección teórico es el ideal ya que la tensión eficaz resultante se asemeja a la real.

6.4 Comunicaciones

El μ C Arduino es una plataforma de bajo costo de electrónica de prototipos. Con la interfaz de LabVIEW para Arduino LIFA se puede aprovechar la potencia del entorno de programación gráfica de LabVIEW para interactuar con Arduino. Esto se debe a que National Instruments ha creado componentes para LabVIEW (VI) y el firmware necesario para convertir la placa de Arduino en una tarjeta de adquisición de datos y poder manejarla por USB desde LabVIEW.



Para conectar la placa Arduino con el entorno de trabajo LabVIEW y comenzar a desarrollar cualquier proyecto, se necesitará la herramienta llamada LabVIEW Interface for Arduino (LIFA). Esta herramienta es una API basada en VIs que permiten controlar el microprocesador Arduino directamente desde el programa, sin necesidad de programar nada en el Arduino excepto un código estándar que ellos te proporcionan.

6.4.1 Instalación toolkit-Arduino y LIFA-base

A continuación describimos los pasos que se recomiendan para la puesta en marcha de la herramienta LIFA:

1) Instalar LabVIEW

Prerrequisito: es necesario tener LabVIEW 2009 o una versión superior para que los VIs de la herramienta LIFA funcionen correctamente.

Se puede descargar la versión de evaluación en el siguiente enlace, aunque para hacerlo primero hay que registrarse.

<http://www.ni.com/trylabview/>

2) Instalar los controladores NI-VISA

Para LabVIEW, Arduino aparece como un instrumento conectado a un puerto serie (RS-232), por lo que es necesario instalar la última versión de los drivers de NI-VISA.

<http://www.ni.com/download/ni-visa-5.1.1/2659/en/>

3) Instalar el IDE de Arduino y sus controladores

Se pueden descargar gratuitamente de la página oficial de Arduino siguiendo este enlace:

<http://arduino.cc/en/Main/Software>

Si es la primera vez que se conecta el Arduino UNO al ordenador mediante un cable USB, el sistema operativo avisará de la existencia de un dispositivo nuevo y se deberá instalar sus controladores para que sea reconocido en las futuras conexiones. Dichos controladores se encuentran en la subcarpeta drivers.

4) Instalar JKI VI Package Manager (VIPM)

VIPM es un administrador de contenidos que permite instalar nuevas librerías y herramientas para LabVIEW de forma rápida y sencilla. Se descarga a través de su página oficial:

<http://jki.net/vipm/download>

5) Instalar el Interfaz de LabVIEW para Arduino (LIFA)

Abrir el VIPM, buscar la herramienta LIFA en la lista de paquetes y pulsar el botón Install Package(s).

6) Cargar el firmware en el Arduino



Abrir el IDE de Arduino, descargado en el paso 3, haciendo doble click en el ejecutable arduino.exe.

Ir a File>>Open y abrimos el archivo LIFA_Base.ino que se suele encontrar en:

C:\Program Files\National Instruments\LabVIEW 20xx\vi.lib\LabVIEW Interface for Arduino\Firmware\LIFA_Base

Seleccionar la placa Arduino UNO en Tools>>Board.

Indicar el puerto COM asignado por el ordenador a Arduino UNO en Tools>>Serial Port, (el número del puerto puede verse en el Administrador de dispositivos del Panel de Control de Windows).

Para finalizar, pulsar el botón de Upload, para cargar el firmware al Arduino UNO. Una vez que termine de cargar es necesario cerrar el Arduino IDE. Es muy importante cerrarlo porque tanto LabVIEW como Arduino están usando el mismo puerto COM. Si no está cerrado, LabVIEW no podrá comunicarse y se bloqueará.

Después de estos seis pasos, la comunicación entre LabVIEW y la tarjeta Arduino estará correctamente configurada y se podrá empezar a desarrollar los programas rápidamente gracias a los bloques y ejemplos básicos incluidos en LIFA en LabVIEW.

Para ello, hay que tener presente que cada bloque que se emplea del paquete de Arduino instalado en LabVIEW activa una función grabada en la programación cargada en el Arduino (LIFA) y cada entrada del bloque en LabVIEW representa los parámetros de entrada para dicha función.

6.42 LabVIEW interface for Arduino (LIFA)

Es un conjunto de herramientas gratuitas que permiten al usuario controlar y adquirir datos a través del μ C Arduino y procesarlo en el entorno de programación gráfica de LabVIEW.

LIFA proporciona una simple y poderosa API para controlar las E/S Digitales, las salidas PWM, las entradas Analógicas, y las comunicaciones I2C y SPI, permitiendo al desarrollador centrarse en la aplicación en vez de la implementación a bajo nivel.

- Características
 - Comunicación mediante USB, Bluetooth o XBee.
 - Velocidades de hasta 200Hz con USB y 50Hz en conexiones inalámbricas.
 - Ejemplos para tareas básicas y sensores.
 - El firmware y la VIs de LabVIEW son de código abierto, permitiendo una completa personalización.

- Funcionamiento

La interfaz de LabVIEW para Arduino envía paquetes de datos de LabVIEW a Arduino. Arduino procesa los paquetes y envía paquetes de retorno. Los paquetes de retorno, se analizan por LabVIEW para proporcionar información útil para el usuario final. Cada paquete es de 15 bytes por defecto y contiene un encabezado, un byte de comando, los bytes de datos, y una suma de comprobación. El firmware LIFA en el Arduino procesa los paquetes, asegurando que los datos no se han dañado durante la transmisión. A continuación, una vez leído el paquete, comprueba el byte de comando y ejecuta las instrucciones con los bytes de datos proporcionados basándose en el byte de comando.

6.5 Programación en LabVIEW

6.5.1 Instrumentos Virtuales

La instrumentación virtual la constituye el conjunto de software y hardware que agregado a una PC, permite a los usuarios interactuar con la computadora como si se estuviera utilizando un instrumento electrónico hecho al gusto del cliente. Las mediciones y accionamientos se realizan sobre el campo real, pero los sistemas de control, mecanismos, registradores, y otros dispositivos serán adquiridos e interconectados en el campo virtual. La idea es sustituir y ampliar elementos "hardware" por otros "software".

La instrumentación virtual tiene una amplia gama de aplicaciones, una de sus principales aplicaciones es en la industria para sistemas de medición, monitoreo de procesos, y aplicaciones de control. También pueden ser utilizados en laboratorios de universidades en carreras de ingenierías orientadas a procesos de control industrial. También es muy utilizado en procesamiento digital de señales, procesamiento en tiempo real, diseño de filtros digitales, generación de señales, automatización de hogares y edificios inteligentes, entre otras.

La Figura 6 muestra el diagrama general de un sistema de instrumentación virtual basado en la adquisición de datos por PC. Puede decirse que un sistema de este tipo está formado básicamente por: sensor + placa de adquisición con acondicionamiento de señal + PC + Software.

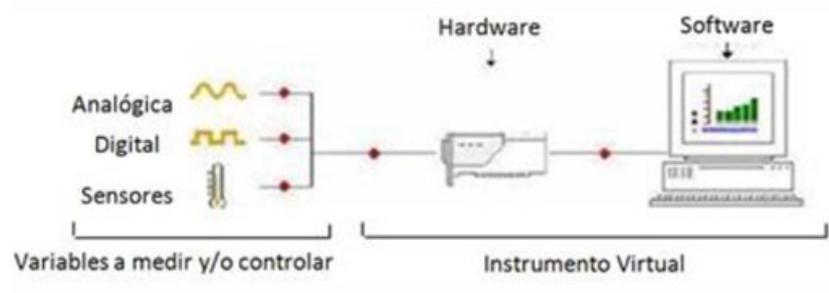


Fig. 26. Sistema de instrumentación virtual.

6.5.2 Introducción al LabVIEW

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje revolucionario de programación gráfica. Este programa fue creado por National Instruments para aplicaciones que involucren adquisición, control, análisis y presentación de datos.

Las ventajas que proporciona el empleo de LabVIEW se resumen en las siguientes:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

LabVIEW se diferencia de los programas de desarrollo de aplicaciones comerciales en un importante aspecto; los lenguajes tradicionales de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabVIEW emplea la programación gráfica ó lenguaje G para crear programas basados en diagramas de bloques.

Al apoyarse sobre símbolos gráficos en lugar de lenguaje escrito, resulta más intuitivo a la hora de realizar los programas, otorgando una gran facilidad de uso tanto a programadores profesionales como a personas con menores conocimientos en programación.

6.5.3 Modo de trabajo en LabVIEW

Al ser una herramienta gráfica de programación, los programas no se escriben, sino que se dibujan, facilitando su comprensión. Como tiene una gran cantidad de bloques pre-diseñados, hace más sencilla la creación del proyecto, con lo que el usuario invierte mucho menos tiempo en programar un dispositivo/bloque y puede dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final.

Los programas creados mediante LabVIEW se denominan Instrumentos Virtuales (VIs), porque su apariencia y funcionamiento imitan los de un instrumento real. Sin embargo son análogos a las funciones creadas con los lenguajes de programación convencionales. En la Fig.25 se puede apreciar que cada VI consta de dos partes diferenciadas:

- **Panel Frontal:** se trata de la interfaz gráfica del VI con el usuario. Se utiliza para interactuar con él cuando el programa se está ejecutando, de forma que pueda observar los datos actualizados en tiempo real. En esta interfaz se definen los controles (se usan como entradas) e indicadores (salidas). El VI recibe las instrucciones programadas del Diagrama de bloques.
- **Diagrama de Bloques:** constituye el código fuente del VI. Aquí es donde se realiza la implementación del programa del VI para controlar ó realizar cualquier procesado de las entradas y salidas que se crearon en el Panel Frontal. Los controles e indicadores que se colocan previamente, se materializan en el Diagrama de Bloques mediante los terminales. Uniendo todos estos distintos elementos entre sí, se construye el circuito del programa.

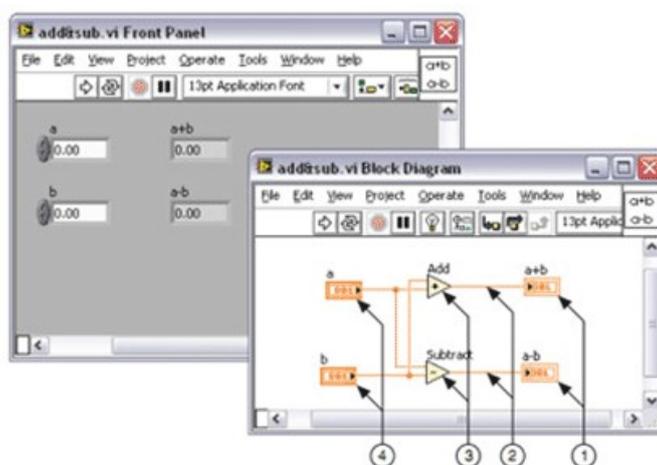


Fig. 27. Panel frontal y de bloques de LabVIEW.

Las Paletas nos proporcionan las herramientas para crear y modificar todos los elementos necesarios. Las utilizadas son las siguientes:

- *Paleta de Controles (Controls Palette), Fig.28:* se utiliza sólo en el Panel Frontal. Contiene todos los controles e indicadores que se emplearán para crear la interfaz del VI con el usuario.

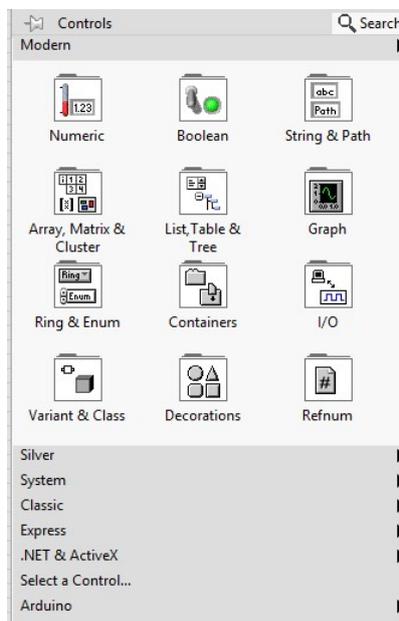


Fig. 28. Paletas de Controles.

- *Paleta de Funciones (Functions Palette), Fig.29:* se usa en el Diagrama de Bloques. Contiene todos los objetos que se emplean en la implementación del programa del VI, ya sean funciones aritméticas, de entrada/salida de señales, adquisición de datos, temporización de ejecución del programa, etc.

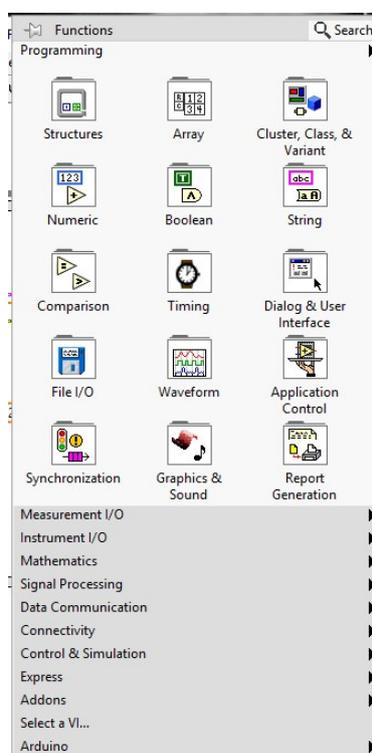


fig. 29. Paletas de funciones.

Paleta de Herramientas (Tools Palette): se emplea para operar y modificar objetos tanto en el panel frontal como en el diagrama de bloques.



fig. 30. Paleta de herramientas.

- Paleta específicas de Arduino: Las principales funciones utilizadas en el proyecto se encuentran en la categoría de Arduino y la de programación. En la gráfica se muestra los dos elementos que se debe añadir al programa para que el Arduino funcione en un modo de adquisición continua. Estos elementos están dentro de Funtions>Arduino>Low level> Analog sampling, y son Continuous Aquisition On y Continuous Acquisition Sample.

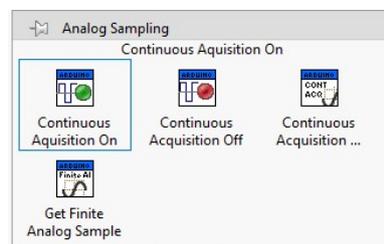


Fig. 31. Paletas de programación para Arduino.

6.5.4 Programación gráfica de LabVIEW

En este apartado se van a describir brevemente los diagramas de bloques de mayor relevancia empleados en este programa, de manera que se pueda comprender mejor el trabajo realizado.

Init: Para iniciar la comunicación con la placa Arduino es necesario colocar la subVI Init.vi. La configuración del bloque elegido es el predeterminado por LabVIEW debido a que el Arduino utilizado es el UNO y la conexión es a través del puerto USB.

Esta subVI cuenta con 6 variables de entrada y dos de salidas (ver figura 33), de las cuales es necesario una mención especial a los siguientes:

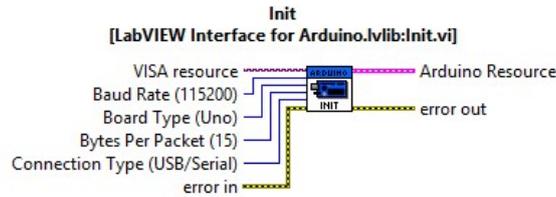


Fig. 33. Configuración del puerto de comunicación.

VISA Resource: se debe de elegir el puerto COM utilizado por el Arduino.

Baud Rate (velocidad de transmisión serial): La tasa de baudios también conocida como baudaje, es el número de unidades de señal por segundo. En el proyecto se ha elegido la predeterminada (115200) ya que es la máxima velocidad de transmisión permitida.

Arduino Resource: Es la salida que transmite la información obtenida en el Arduino a otras funciones.

Error out: La salida de errores.

Es necesario comprobar en el administrador de dispositivos del ordenador que el puerto COM utilizado para la comunicación coincide con la elegida en la programación.



Fig. 32. Configuración del puerto de comunicación.

Close: A su vez, también es necesario terminar la comunicación con Arduino, y esto se realiza con el bloque subVI Close.vi. La cual cuenta con dos entradas y una salida (figura 34), con las que definimos:

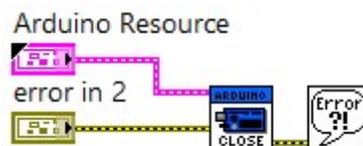


Fig. 33. Bloque Close.

Arduino Resource: Es la salida de información obtenida de Init.vi y que se utiliza en diversas funciones.

Error in: La salida de errores.

Error out: La salida de errores.

Una vez creados los subVI utilizados para iniciar y terminar la comunicación con LabVIEW, necesitamos otra serie de subVI para muestrear la señal analógica que proporcionan los sensores.

Continuous Aquisition On: programa al Arduino en modo de adquisición continua.

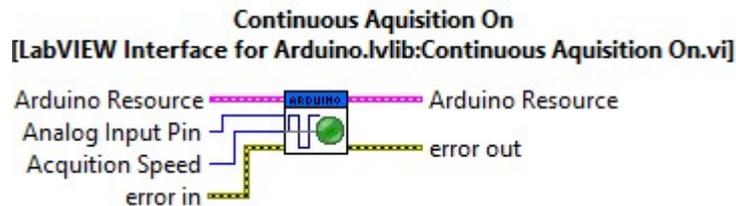


Fig. 34. Bloque Continuous Aquisition ON.

Analog Input Pin: el pin analógico utilizado es el 0.

Acquisition Speed (Velocidad de adquisición): es la frecuencia a la que LabVIEW convierte la forma de onda de la entrada analógica a datos digitales. se debe de elegir una frecuencia teniendo en cuenta que:

- LabVIEW limita la frecuencia máxima permitida, la cual no puede superar 5kHz.
- El Teorema de Muestreo de Nyquist antes visto que, explica la relación entre la velocidad de muestreo y la frecuencia de la señal medida. Afirma que la velocidad de muestreo debe ser mayor que el doble la señal medida. Como la frecuencia de la red es de 50Hz la frecuencia mínima necesaria para poder reproducir la onda es de 100Hz para que no se produzca el fenómeno de aliasing.
- El μC de Arduino UNO contiene internamente un conversor analógico a digital de 6 canales. El conversor tiene una resolución de 10 bits, devolviendo enteros entre 0 y 1023. La entrada analógica tiene un rango de 0 a 5 voltios por lo que la resolución es de $\frac{5}{2^{10}-1} = 4.88 \text{ mV}$.

A la hora de elegir la velocidad de muestreo se debe de tener en cuenta la frecuencia de la señal, 50 Hz. Con esta se puede calcular el periodo de muestreo, siendo:

$$T = \frac{1}{F} = 20ms$$

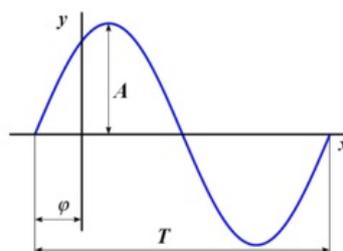


Fig. 35. Características de una onda sinusoidal.

- Comprobación de la velocidad de adquisición ideal:

La velocidad de muestreo ideal elegida es de 870 muestras por segundo. Con esta velocidad de muestreo la señal física que es digitalizada se puede reconstruir de forma adecuada.

Por un lado, realizando el análisis de la transformada rápida de Fourier en LabVIEW se puede comprobar que la señal objeto a análisis es una señal de 50Hz. Para que el programa realice este análisis es necesario modificar las propiedades del gráfico multiplicando el factor de escala por 870 como se puede ver en la figura 36.

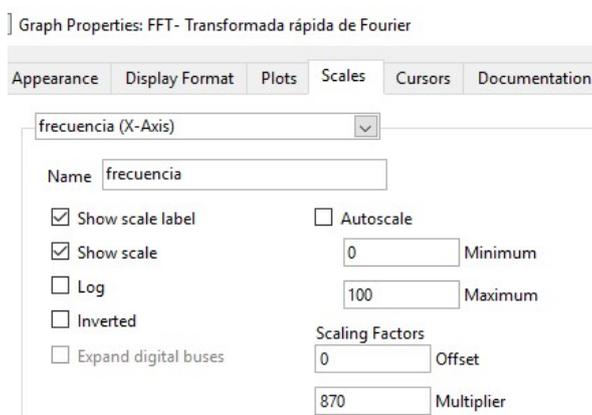


Fig. 36. Propiedades de la gráfica de la transformada rápida de Fourier.

En la siguiente gráfica se puede observar 50 ondas que representan la señal de la red. Para ello, se ha realizado una escala de la gráfica para que represente la señal que se produce en un segundo.

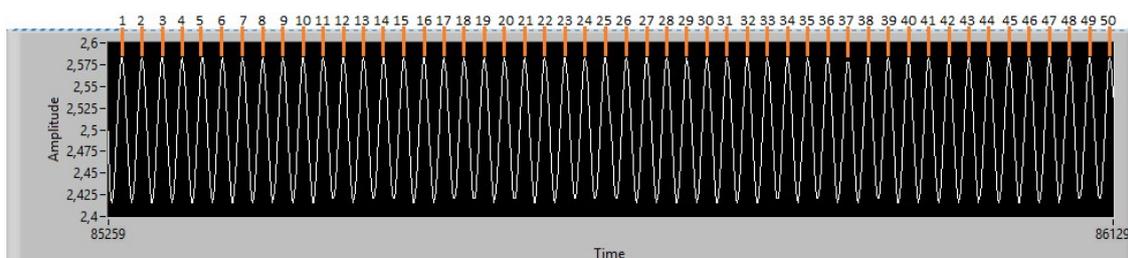


Fig. 37. Representación de la señal con un muestreo de 870 Hz.

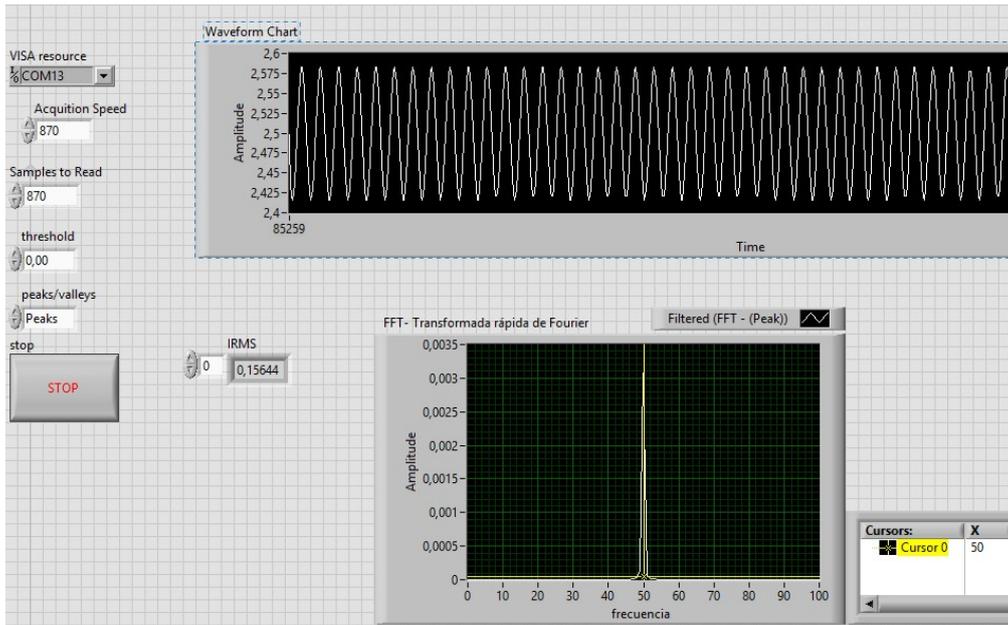


Fig. 38. Panel Frontal con la velocidad de adquisición a 870Hz.

Por otro lado, se han guardado 2000 datos para realizar la comprobación de la resolución de trabajo. El análisis de estos datos se ha realizado a través de una tabla de Excel, con la cual se ha podido comprobar que la resolución Teórica del ADC de 10 bits, ($1 \text{ LSB} = \frac{5}{2^{10}-1} = 4.888 \text{ mV}$) coincide con la real 0.0049 mV .

Frecuencia de muestreo menor que la ideal. Con una frecuencia de muestreo menor que 470 Hz se puede observar el fenómeno de aliasing, ya que aparentemente la onda representada es una onda real, pero la representación de FFT no muestra el valor de frecuencia de 50 Hz y el valor oscila. También se puede observar que el valor de la IRMS es erróneo.

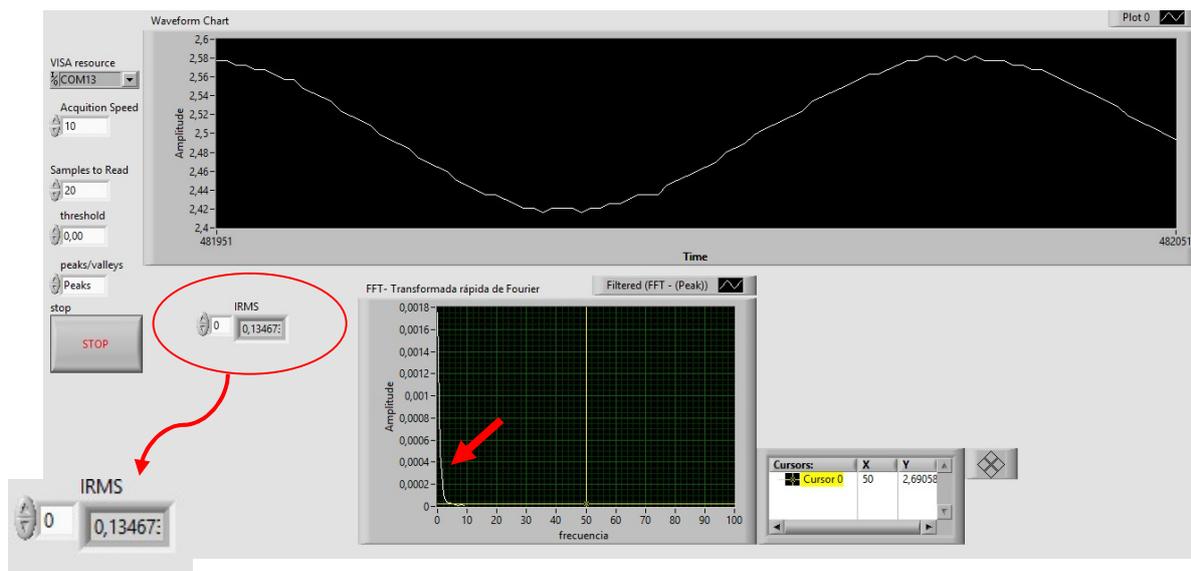


Fig. 39. Representación de la señal con una velocidad de adquisición de 10 Hz.

Frecuencia de muestreo mayor que la ideal. Con una frecuencia de muestreo mayor que 1000Hz el análisis de la onda es erróneo.

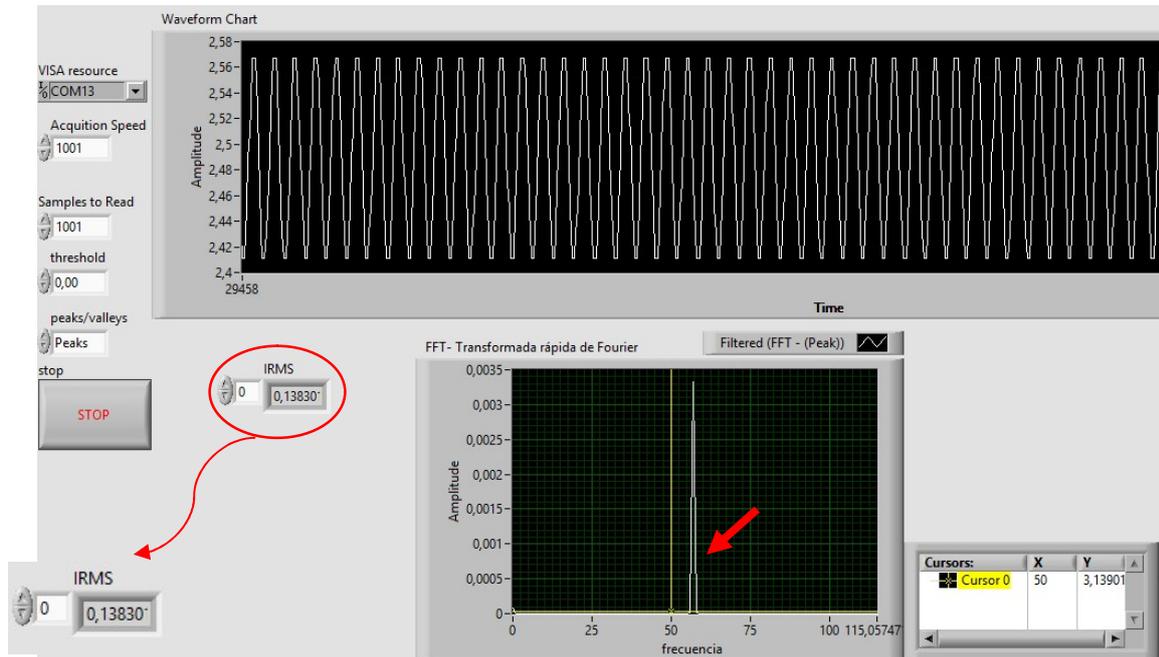


Fig. 40. Representación de la señal con una velocidad de adquisición de 1001 Hz

Como se ha explicado anteriormente a partir de 1001 Hz se pierden 2 bits de resolución. Realizando el análisis de datos a través de una hoja de cálculo de Excel se ha podido comprobar que la resolución a pasado de 10 bits a una de 8 bits, dando un mínimo cambio entre un valor y otro de $1 \text{ LSB} = 0.0196 \text{ mV}$. Esto hace que no se pueda representar y medir el valor máximo de la señal, obteniéndose así una lectura errónea de la intensidad eficaz.

A partir de este punto si se va aumentando la frecuencia de muestreo la señal empieza a perder resolución de forma contante hasta el límite de la frecuencia máxima permitida, 5kHz, donde la señal se pierde.

Teóricamente, el resultado de la gráfica y de los datos debería de asemejarse a la señal real, ya que al aumentar la velocidad de adquisición, aumenta las muestras por ciclo de la señal analizada, mejorando la calidad de la lectura. Sin embargo, se observa que el cálculo del IRMS es erróneo, ya que no se representa una señal con 50 ciclos por segundo como es en la realidad y por último mediante el análisis de Fourier se puede observar que no se analiza una señal de 50 Hz.

Aunque el teorema de Nyquist no menciona que surgen problemas cuanto mayor es la frecuencia de muestreo, es necesario comentar que las frecuencias de muestreo altas resultara un problema en el proyecto. Esto se debe a la programación de "LIFA_Base" cargada en el microprocesador a Arduino, donde la máxima frecuencia permitida para tener una mínima resolución de 0.0049mV es de 1000Hz.

Continuous Acquisition Sample: Da un número de muestras específicas a una velocidad específica de la señal recibida.

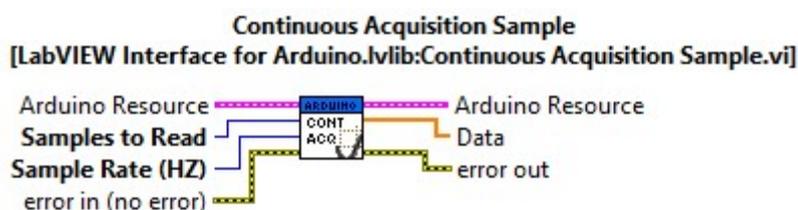


Fig. 41. Bloque Continuous Acquisition Sample.

Samples to Read: muestras para leer. Se ha elegido el mismo valor que el de la velocidad de muestreo para que cada segundo se actualicen los datos en LabVIEW.

Samples Rate: velocidad de muestreo.

Data: las muestras cogidas son digitalizadas y convertidas en un valor numérico de coma flotante de doble precisión y guardados formato Array.

WaveForm Peak Detection: Detecta si en la señal a la entrada hay picos o valles, dependiendo de lo que se desee.

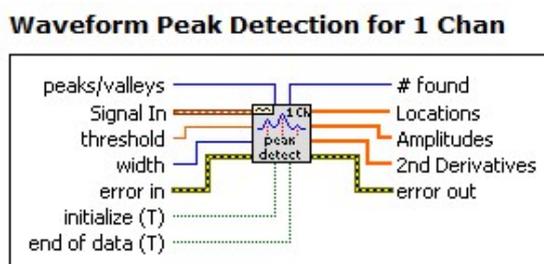


Fig. 42. Bloque WaveForm Peak Detection.

Peaks/valleys: Debido a que se analiza una señal sinusoidal es indiferente coger picos o valles, aun así se ha elegido detectar los picos.

Signal In: señal de entrada a la cual se desea analizar.

Threshold: banda limitadora. Al encender los aparatos electrónicos puede darse un valor de pico no deseado, por lo que con esta herramienta limita el valor máximo de la señal ignorando estos picos.

Amplitudes: muestra el valor más alto recibido.

Cálculo IRMS: Con las herramientas proporcionadas por LabVIEW se realiza la ecuación necesaria para realizar el cálculo de la corriente eficaz.

$$IRMS_{\text{primario}} = \frac{(V_{\text{pin-2.5}}) \cdot 2000}{760 \cdot \sqrt{2}} = \frac{(V_{\text{pin-2.5}}) \cdot 2000}{1074.80} = 1.862 \cdot (V_{\text{pin-2.5}})$$

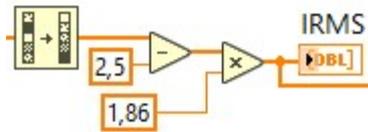


fig. 43. Bloque para calcular la intensidad eficaz de del conductor

Archivamiento de datos: Para el análisis posterior de los datos capturados a través del Arduino, los valores que LabVIEW traduce en tensión o intensidad son guardados en un documento .txt. Este bloque de programación se utiliza como comprobación de un apartado específico del proyecto, por lo que en el diagrama de bloques final no se incluye.

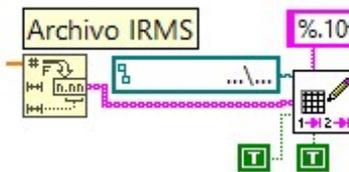


Fig. 44. Bloque para guardar los datos en un archivo .txt

Análisis Espectral: Mide el espectro y muestra los resultados en términos de potencia. Se utiliza esta herramienta para medir la frecuencia de la señal muestreada.

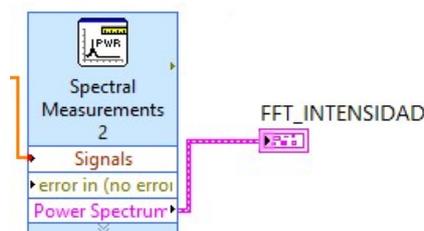


fig. 45. Bloque para el análisis espectral.

Representación conjunta de las señales: A través de los datos adquiridos de tensión e intensidad, se han construido sus respectivas ondas analógicas con el bloque de función *Build Waveform*. Estas señales se combinan con el bloque *Merge signal*, que devuelve en una sola señal de salida las señales combinadas. La señal resultante se conecta a un indicador gráfico y se podrá observar como LabVIEW representa las dos señales de tensión e intensidad en una misma gráfica. El bloque creado se puede ver en la siguiente figura:

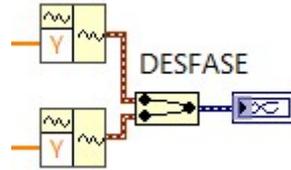


fig. 46. Combinación de dos señales.

Tone Measurement: se parte de los datos adquiridos en el bloque de adquisición continua, cuyos valores se encuentran en un array de datos, y se convierten en datos dinámicos, para así poder utilizar el bloque de tone measurement.

Este bloque mide la fase de una señal, siempre y cuando en la configuración del bloque se le especifique la frecuencia aproximada de la señal a medir.

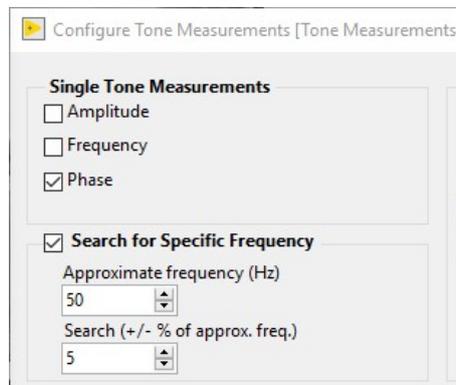


fig. 47. Configuración de Tone Measurements

Tendiendo el valor de fase de cada señal, solo basta con realizar la diferencia entre ambas para conseguir el desfase entre ellas.

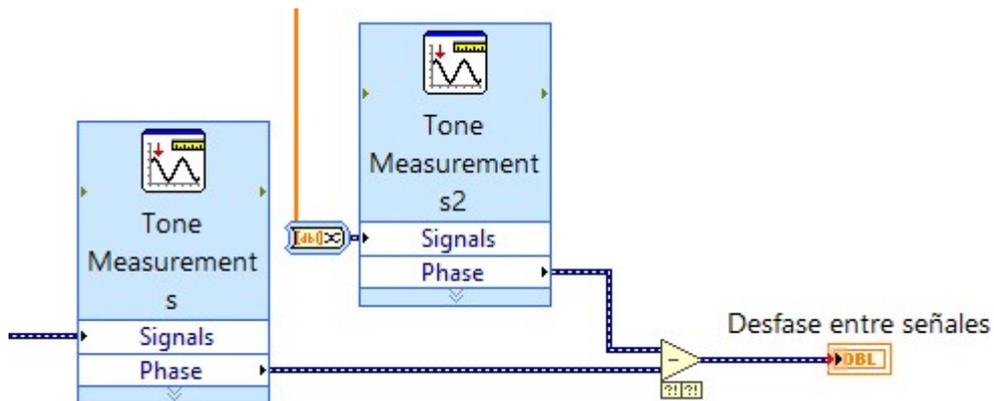


fig. 48. Medición del desfase entre señales.



METODOLOGÍA SEGUIDA EN EL DESARROLLO DEL TRABAJO

7. DESCRIPCIÓN DE TAREAS, FASES, EQUIPOS O PROCEDIMIENTOS

Para lograr el desarrollo del proyecto de manera satisfactoria ha sido necesario realizar las siguientes labores:

1. Realizar el estudio general del análisis de las señales analógicas. En concreto conceptos generales de la energía, la transformada de Fourier, modo de adquisición de las señales...
2. Elegir el sensor de intensidad más apropiado para el desarrollo del proyecto.
3. Elegir el sensor de tensión, cuyo único requisito preferible, es que la salida de este sea de una tensión baja.
4. Realizar el circuito acondicionador de los respectivos transductores.
5. Realizar la compra de los transductores y un instrumento de medida que sirva de comprobación de intensidad y tensión. Como los elementos se han comprado a través de una página web, ha sido necesario avanzar en otra etapa del proyecto hasta la llegada de los mismos.
6. Familiarizarse con la comunicación entre un Arduino y LabVIEW, descargándose las aplicaciones necesarias para conseguir el correcto funcionamiento entre ambas.
7. Familiarizarse con el entorno de LabVIEW con el objetivo de comprender el funcionamiento básico del programa. Aprendiendo así, a utilizar las estructuras más empleadas del sistema, utilizar las herramientas principales,..
8. Familiarizarse con la arquitectura de Arduino, específicamente con su convertidor ADC.
9. Implementación del diseño del circuito de intensidad:
 - comprobación de su funcionamiento a través de Arduino.
 - Creación de la interfaz en LabVIEW que analice los datos muestreados y calcule la intensidad eficaz del conductor
 - Realización de la calibración del sistema analizador de intensidad. Siendo necesario acudir con un especialista que proporcionó los dispositivos adecuados para utilizar la correcta calibración como un autotransformador, una resistencia especial y equipos de medición con una resolución elevada.
10. Realizar un estudio del análisis de señales analógicas mediante el muestreo. Para esto ha sido necesario:
 - Comprender el funcionamiento de la librería cargada en el Arduino LIFA, siendo esencial en el desarrollo del proyecto.
 - búsqueda de información y realizar pruebas con el dispositivos hasta de llegar a una conclusión clara.
11. Implementación del diseño del circuito de tensión:
 - comprobación de su funcionamiento a través de Arduino.



- Creación de la interfaz en LabVIEW que analice los datos muestreados y calcule la tensión eficaz del conductor.
 - Realización de la calibración del sistema analizador de intensidad. Siendo necesario acudir al laboratorio de la universidad que cuenta con los aparatos indispensables para la calibración del sistema como un oscilador y un generador de funciones.
12. Realización de la programación final del dispositivo a través de LabVIEW. La mejor forma de realizar la monitorización del sistema ha sido analizando las señales en serie.
 13. Medición del desfase entre las señales de tensión e intensidad.
 14. Realización de la placa PCB.
 15. Generar la documentación.



8. DIAGRAMA DE GANTT

TAREAS																																								
	OCTUBRE				NOVIEMBRE				DICIEMBRE				ENERO				FEBRERO				MARZO				ABRIL				MAYO				JUNIO				JULIO			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Tarea 1																																								
Tarea 2																																								
Tarea 3																																								
Tarea 4																																								
Tarea 5																																								
Tarea 6																																								
Tarea 7																																								
Tarea 8																																								
Tarea 9																																								
Tarea 10																																								
Tarea 11																																								
Tarea 12																																								
Tarea 13																																								
Tarea 14																																								
Tarea 15																																								

9. RESULTADOS

El objetivo general del proyecto se ha realizado, ya que el dispositivo desarrollado es capaz de monitorizar la potencia consumida por un dispositivo eléctrico.

Por otra parte se ha cumplido con los objetivos específicos del sistema. Como que éste realice una medición con un bajo error de medida, realizar la medición del desfase entre las señales analizadas, creación de un diseño de placa PCB.

En la siguiente figura, se puede observar el prototipo final, utilizado para, las mediciones, el calibrado del sistema y a lo largo de todo el proyecto para observar el funcionamiento de la representación gráfica realizada en LabVIEW hasta su programación final:

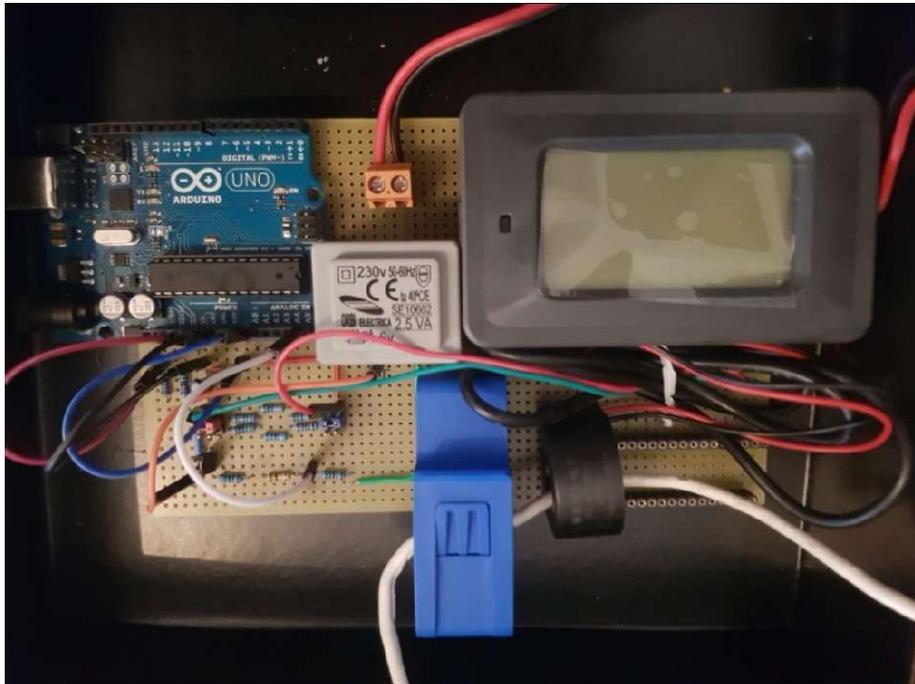


Fig. 50. Prototipo de final.



9.1 Instrumento virtual final

9.1.1 Panel Frontal

la pantalla de monitorización, permite al usuario visualizar:

- Los resultados de los cálculos de tensión, intensidad y potencia de la red que realiza el programa.
- Las gráficas de tensión e intensidad creadas a partir de los datos que reciben las entradas analógicas del Arduino y que provienen de circuitos acondicionadores que traducen las señales físicas deseadas captada por los sensores en señales eléctricas.
- El análisis espectral en función de la frecuencia de la señal muestreada. Un análisis realizado mediante la transformada rápida de Fourier.
- La dos señales muestreadas en una misma gráfica para poder observar el desfase entre ella. También se puede observar el valor numérico del desfase.

En la siguiente gráfica se puede observar la interfaz del usuario:

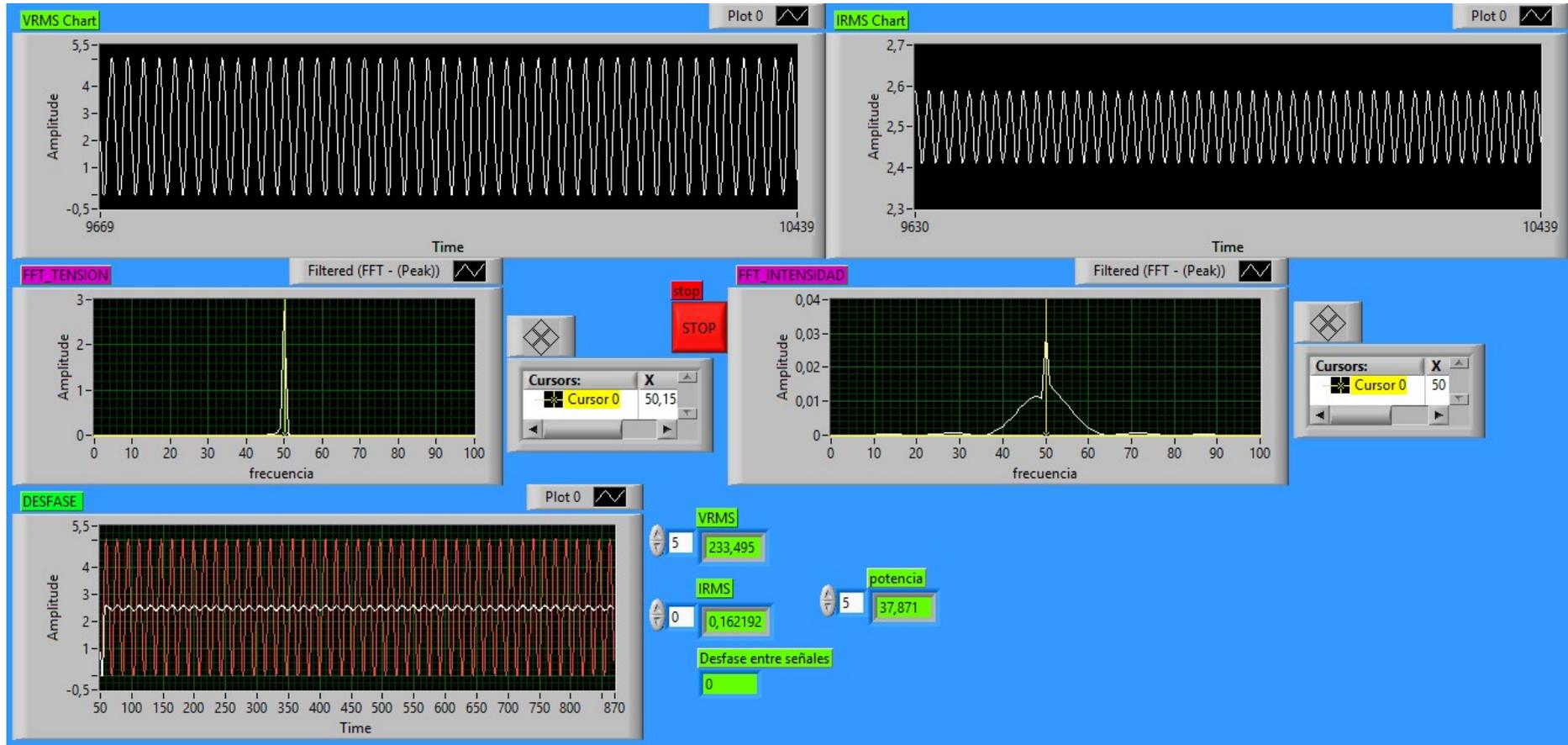


Fig. 51. Panel Frontal Final.



9.1.2 Diagrama de Bloques

La parte de la programación se ha desarrollado en tres bloques. El primer bloque realiza la adquisición de la señal de intensidad, en el segundo se realizan las mediciones de las señales de tensión e intensidad, y en el último bloque se mide el desfase entre las señales adquiridas.

En el primer bloque se realiza la adquisición de la intensidad y la obtención de unos resultados fiables. La estructura de la medida de la intensidad es reutilizada en la medida de la tensión, ya que realizan la misma función.

Para la implementación del primer bloque, se ha empezado realizando una estructura While, que realiza el análisis exclusivo de la intensidad que circula por el conductor, hasta que se detenga manualmente la ejecución del programa o hasta que se produzca un error.

Se han empleado bloques específicos de adquisición de datos de la librería de Arduino para el muestreo de la señal a una frecuencia adecuada. Resultando la elección de la frecuencia de muestreo una de las dificultades principales en la realización del proyecto, debido a las incoherencias entre la frecuencia de muestreo que permite el ADC y la que permite la interfaz de LabVIEW. Han sido necesarias numerosas pruebas y tiempo de investigación hasta concluir que el problema de las diferencias de muestreo se debía a la librería "LIFA". Para llegar a esta conclusión ha sido necesario generar un bloque que guarde los datos adquiridos en un archivo .txt, para su posterior análisis en una hoja de Excel.

Para poder observar la señal que recibe el Arduino en su entrada analógica, se ha generado un gráfico que agrega cada nuevo dato recibido a los adquiridos previamente. A través de esta gráfica se puede comprobar si los valores adquiridos concuerdan con los calculados teóricamente, permitiendo continuar con la programación a partir de unos datos fiables.

Seguidamente, se recopila la amplitud de la señal de entrada, generada por el bloque de Arduino. La información se almacena en formato Array y mediante un filtrado de datos se preparan los datos necesarios para los cálculos posteriores.

Por otra parte, se mide el espectro de la señal en función de la frecuencia y se comprueba si la señal adquirida coincide con la señal medida en la realidad.

En el segundo bloque de la programación se ha conseguido realizar las mediciones de las señales de tensión e intensidad de forma continua, gracias a una programación en serie a través de la estructura *Flat Sequence*, donde el programa sigue un orden estipulado. En un principio se intentó realizar la programación en paralelo, pero la adquisición de las señales se realizaba de manera inadecuada ya que las señales se adquirían de forma intercalada.

En el último bloque programado se ha de tener en cuenta que las pruebas se han realizado con una bombilla, y como se ha visto anteriormente, al ser una carga resistiva no produce un desfase entre la tensión y la intensidad. A través de la gráfica de desfase se pueden observar las dos señales en una misma gráfica, comprobándose que el desfase es constante entre ambas.



Teniendo esto en cuenta, se ha intentado por varios métodos encontrar el desfase deseado. En algunos métodos el desfase de cada señal era variable o realizaba una medida de desfase errónea. Finalmente se halló una función que media de forma correcta el desfase de las señales adquiridas.

A continuación se presenta el diseño final del sistema en el programa de LabVIEW:

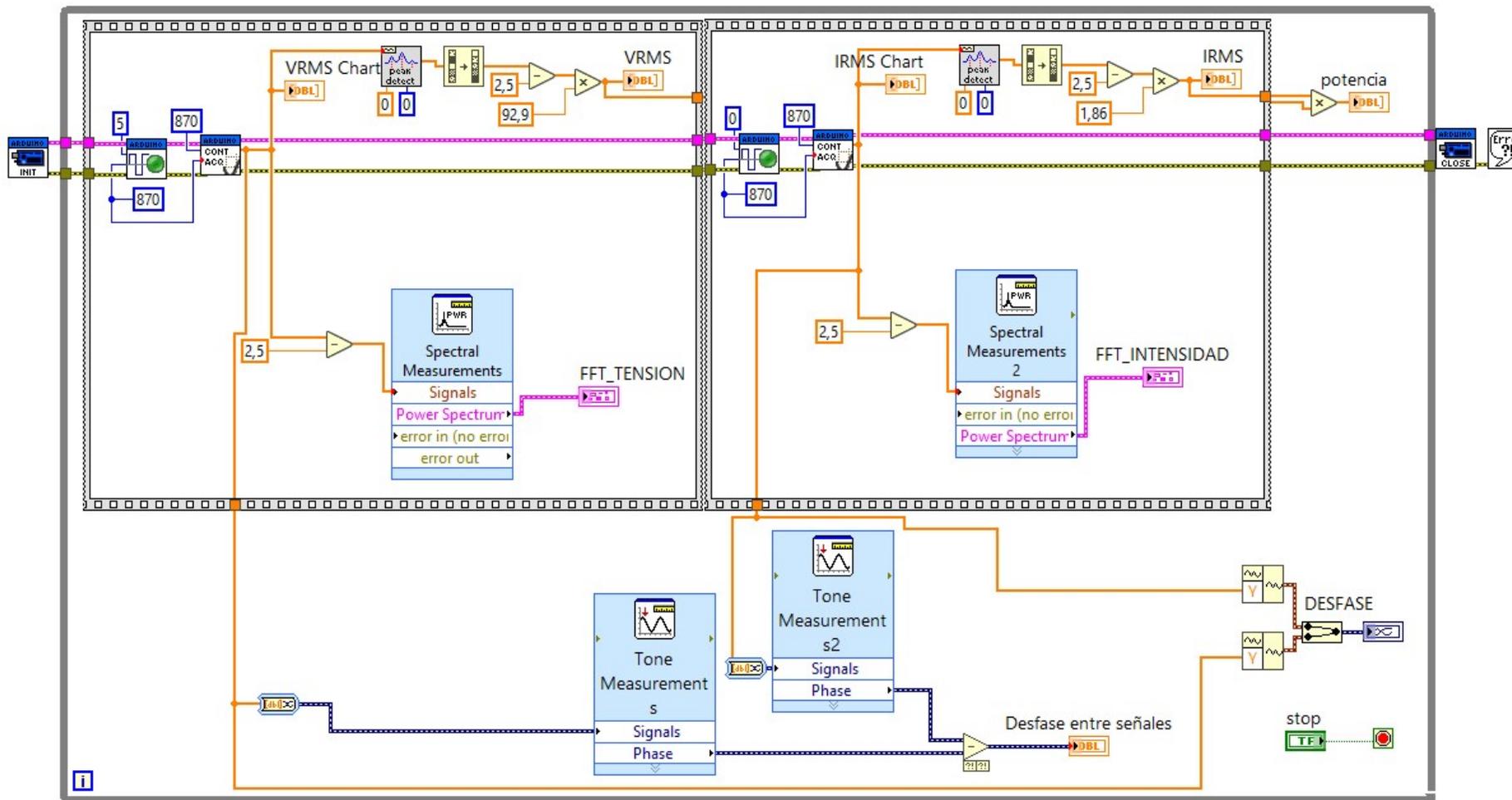


Fig. 52. Diagrama de bloques final.

9.2 Mejora del dispositivo

La programación en el μC se podría mejorar, ya que la programación actual que contiene el microprocesador es genérica y, como sucede en el proyecto creado, restrictiva. Lo ideal sería crear un programa específico para el Arduino que nos permita trabajar con una frecuencia mayor de 1000 Hz.

ASPECTOS ECONÓMICOS

10. PRESUPUESTO EJECUTADO

COMPONENTES	COSTE UNITARIO (€)	COSTE TOTAL (€)
Arduino UNO	25,71	25,71
Transformador de PCB, 6V AC, 2 salidas, Potencia 2.5VA	4,49	4,49
sensor SCT-013-000	3,6	3,6
3X Resistencia 10k Ω	0,045	0,135
2X Resistencia 330 Ω	0,277	0,554
Resistencia 4,7k Ω	0,13	0,13
Resistencia 100 Ω	0,048	0,048
Resistencia 51k Ω	0,134	0,134
Condensador electrolítico aluminio 10 μF	0,226	0,226
Condensador cerámico mono capa 33 nF	0,01	0,01
bloque terminal para PCB	0,421	0,421
COSTE TOTAL		35,458

Tabla 7. Presupuesto.

11. ANALISIS DE RENTABILIDAD

La rentabilidad del proyecto resulta relativa ya que, aunque la realización del mismo ha resultado económica, es necesaria la utilización de un ordenador para poder observar los resultados del análisis. En cambio con los dispositivos de venta al público de monitorización de la energía, son más costosos pero llevan implementada una pantalla que muestra los datos de la señal. Sin embargo al ser un dispositivo casero y para uso propio, es rentable la realización del dispositivo.

CONCLUSIONES

El sistema desarrollado cumple con las expectativas iniciales al ser un sistema de adquisición y monitorización económica que trabaja de forma continua entre los distintos dispositivos, es decir, entre la placa de adquisición de Arduino, los sensores con su sistema acondicionador y el procesador de LabVIEW.

Se han adaptado con éxito los transductores de tensión y corriente, como parte del sistema de adquisición de datos, en la placa de Arduino.

En la parte del procesador, se ha aprendido a programar en lenguaje de programación gráfica, a realizar la comunicación con otro dispositivo mediante comunicación por el puerto COM y se ha llegado a analizar una parte específica de la programación que comunica el Arduino con LabVIEW, la cual, es la utilizada en el proyecto. Este programa que se carga en el Arduino "LIFA" es un programa genérico y con gran variedad de funciones, que hace difícil el entendimiento de su programación, por lo que no es recomendable modificaciones en el mismo. Lo ideal sería realizar una programación específica en el Arduino, con los parámetros deseados para ganar en resolución, en rapidez y en espacio de memoria libre del Arduino. El ejemplo claro es que con el programa de LIFA no se puede sobrepasar la frecuencia de muestreo de 1000 Hz debido a que se pierden dos bits de resolución.

Gracias a los requisitos del diseño y a la gran cantidad de información acerca de la programación gráfica de LabVIEW, se ha podido realizar el dispositivo y aprender a manejar la herramienta. Aunque en un principio parecía una herramienta compleja por su gran cantidad de funciones e información, resultó ser un programa con una dificultad intermedia con un gran abanico de posibilidades para realizar una tarea concreta. Comprobados el potencial que tienen las herramientas de Arduino y LabVIEW para la realización de todo tipo de proyectos, ya que ambos unidos crean un entorno de desarrollo sencillo y eficiente, además de ser plataforma con una comunidad que ofrece soporte de muy buena calidad.

En cuanto a las conclusiones a nivel personal, el proyecto ha sido de satisfactorio, ya que he aprendido el entorno de un programa nuevo, de conceptos generales de la instrumentación electrónica de manera práctica, la realización del diseño de una placa pcb. Por último, la relación con el jefe de proyecto ha sido la adecuada, permitiendo un avance personal y recibiendo apoyo en puntos críticos del proyecto.

BIBLIOGRAFÍA

<https://programarfacil.com/blog/arduino-blog/sct-013-consumo-electrico-arduino/>

<http://diymakers.es/sensores-en-entradas-analogicas-de-arduino/>

<https://www.youtube.com/watch?v=swEfaxLzNkY>

<http://www.rinconeducativo.org/es/recursos-educativos/centrales-electricas>

<https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>

<https://www.luisllamas.es/medir-tensiones-de-220v-230v-con-arduino-y-transformador/>

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

http://stel.ub.edu/labfon/sites/default/files/EFE-X-JBernal_PGomez_JBobadilla-FFT_una_vision_practica_herramienta_para_el_analisis_espectral_de_la_voz.pdf

<https://es.scribd.com/doc/225082637/GUIA-DE-INSTALACION-TOOLKI-arduino-y-LIFA-base-docx>

<http://proyecto987.es/blog/wp-content/uploads/2016/04/Arduino-LabVIEW.pdf>

<https://uvadoc.uva.es/bitstream/10324/13170/1/TFG-P-251.pdf>

<https://designspark.zendesk.com/hc/en-us/categories/201145765-DesignSpark-PCB>

ANEXOS I

12. PROGRAMACIÓN EN ARDUINO

Se ha realizado la siguiente programación en el Arduino para realizar la calibración teórica y como comprobación de la resolución.

1.1 Intensidad

```
Float muestra [200], maxVal, maxReal;
Double corrienteRMS;
Void setup ()
{
    Serial.begin (115200); //Es necesaria una velocidad de transmisión alta para poder
reconstruir de manera correcta la señal
}
Void loop ()
{
    {
        Serial.print ("MAX: "); // voltaje en la entrada analógica
        Serial.println (maxReal, 4); // el valor del voltaje con cuatro dígitos.
        Serial.print ("corrienteRMS: "); // corriente eficaz en el conductor
        Serial.println (corrienteRMS, 4); // la corriente eficaz con cuatro dígitos
        Delay (2000); //retardo de dos segundos
    }
    maxVal = 0; // valor inicial de la muestra máxima = 0
    For (int i=0; i<200; i++) //recolección de 200 muestras
    {
        Muestra [i]= analogRead (A0); // valor de tensión codificado como un número
entero 0 a 1023.
        If (maxVal < muestra [i]) //compara cada número del array con el maxVal
        {
            maxVal= muestra [i]; //guarda el valor más alto de las muestras, solo
se puede reemplazar por una muestra de mayor valor
        }
    }
}
```

```

    maxReal=maxVal * (5.0 / 1023.0); // conversión del valor entero de la entrada
analógica a tensión
    Double voltajeSensor = (maxVal * (5.0 / 1023.0))-2.5; //Fórmula IRMS
    Double corriente =voltageSensor/760; //Fórmula IRMS
    corrienteRMS = corriente*2000*1.0649/sqrt (2);// 1.0649 parámetro de corrección
entre corriente teórica y real
    Return (corrienteRMS);
    Return maxVal;
}

```

1.2 Tensión

```

Float muestra [200], maxVal, maxReal;
Double voltajeRMS;
Void setup ()
{
    Serial.begin (115200);
}
Void loop ()
{
    {
        Serial.print ("MAX: "); // voltaje en la entrada analógica
        Serial.println (maxReal, 4); // el valor del voltaje con cuatro dígitos.
        Serial.print ("voltajeRMS: "); //Voltaje eficaz en el conductor
        Serial.println (voltajeRMS, 4); // voltaje eficaz con cuatro dígitos
        Delay (2000); //retardo de dos segundos
    }
    maxVal = 0; // valor inicial de la muestra máxima = 0
    For (int i=0; i<200; i++) //recolección de 200 muestras
    {
        Muestra [i]= analogRead (A5); // valor de tensión codificado como un número
entero 0 a 1023.
        If (maxVal < muestra [i]) //compara cada número del array con el maxVal
        {
            maxVal= muestra [i]; //guarda el valor más alto de las muestras, solo
se puede reemplazar por una muestra de mayor valor
        }
    }
}

```

```

maxReal=maxVal * (5.0 / 1023.0); // conversión del valor entero de la entrada
analógica a tensión

```

```

Double voltajeSensor = (maxVal * (5.0 / 1023.0))-2.5; //Fórmula VRMS

```

```

Double voltaje =voltajeSensor/14700; //Fórmula VRMS

```

```

voltajeRMS = voltaje*65700*38.33*0.769/sqrt (2); // 0.769 ajuste teórico

```

```

Return (voltajeRMS);

```

```

Return maxVal;

```

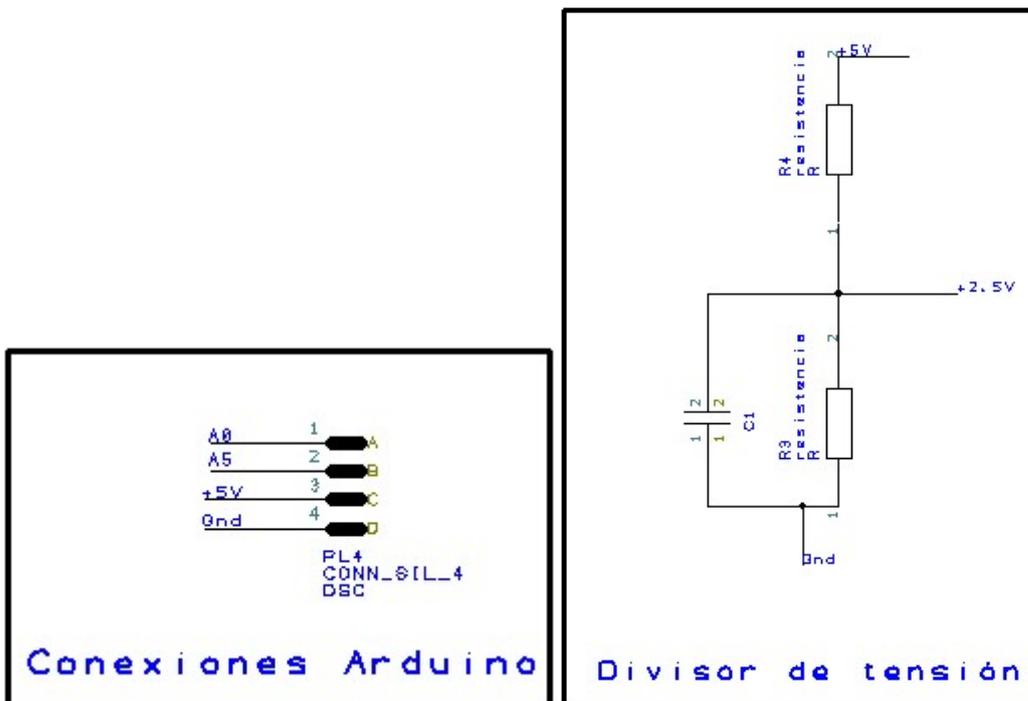
```

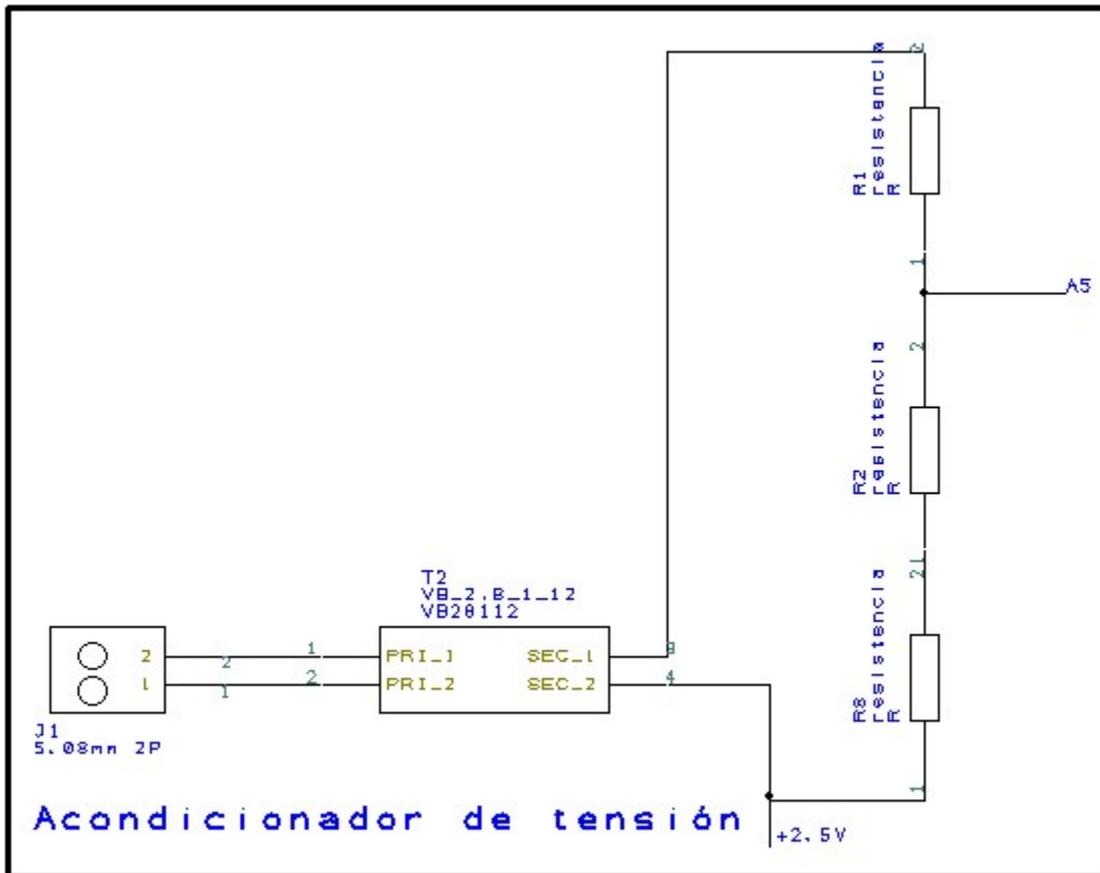
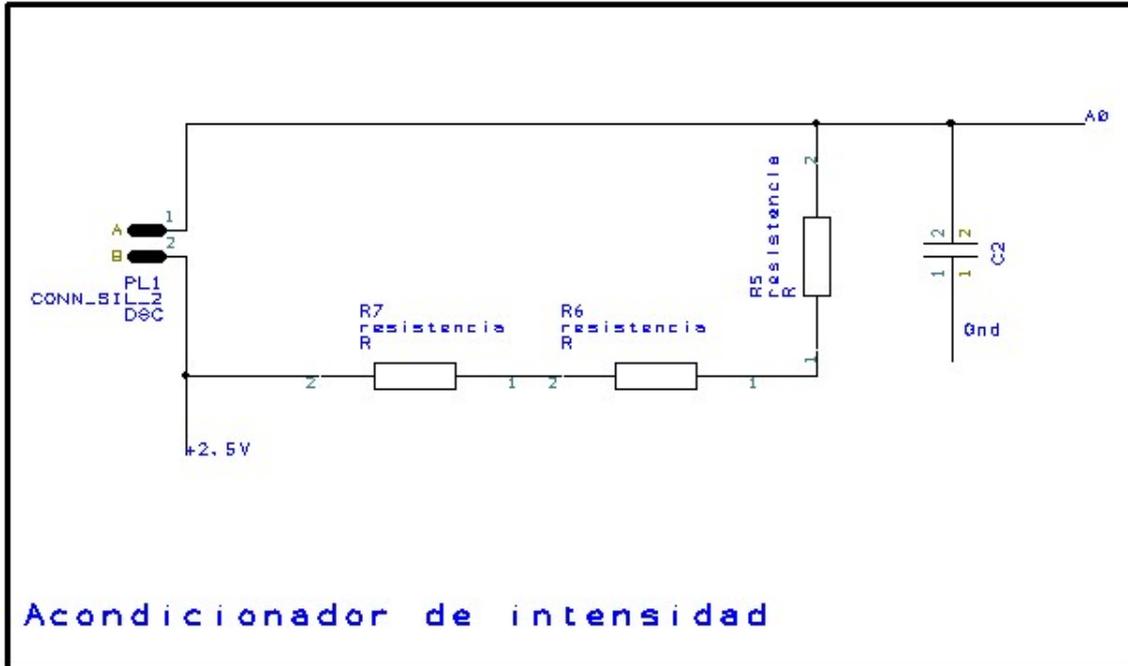
}

```

13. ESQUEMAS DEL DISEÑO

A continuación se presentan los esquemas del proyecto, generados con la herramienta designspark:





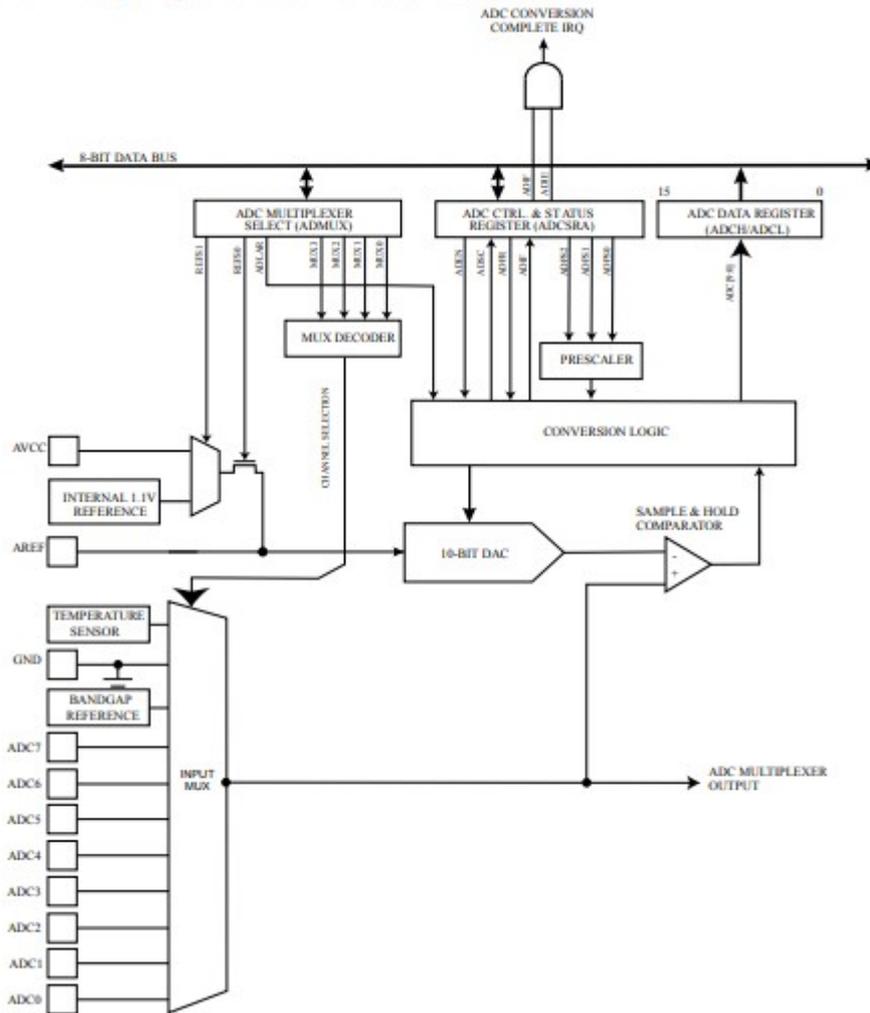
14. ATMEGA328/P DATASHEET

28. ADC - Analog to Digital Converter

28.1. Features

- 10-bit Resolution
- 0.5 LSB Integral Non-Linearity
- ± 2 LSB Absolute Accuracy
- 13 - 260 μ s Conversion Time
- Up to 76.9kSPS (Up to 15kSPS at Maximum Resolution)
- Six Multiplexed Single Ended Input Channels
- Two Additional Multiplexed Single Ended Input Channels (TQFP and VFQFN Package only)
- Temperature Sensor Input Channel
- Optional Left Adjustment for ADC Result Readout
- 0 - V_{CC} ADC Input Voltage Range
- Selectable 1.1V ADC Reference Voltage
- Free Running or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

Figure 28-1. Analog to Digital Converter Block Schematic Operation



Prescaling and Conversion Timing

Figure 28-3. ADC Prescaler

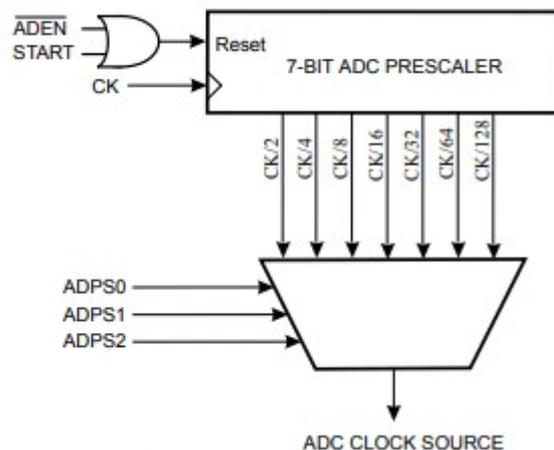


Figure 28-4. ADC Timing Diagram, First Conversion (Single Conversion Mode)

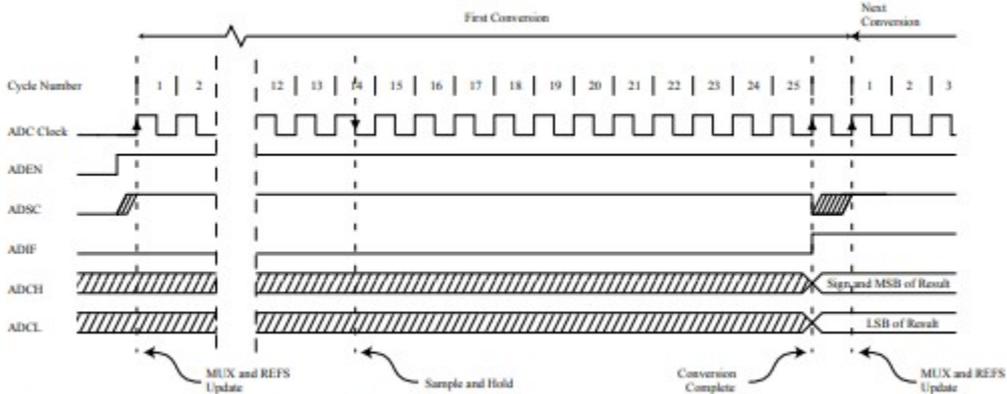


Figure 28-5. ADC Timing Diagram, Single Conversion

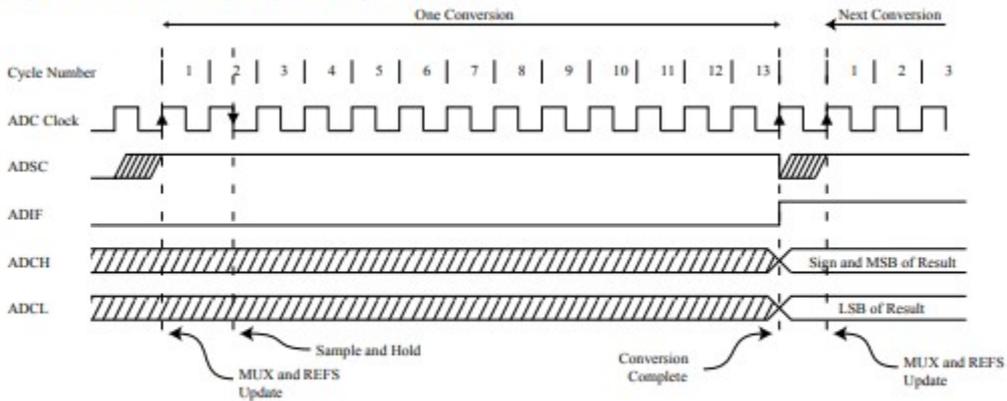


Figure 28-6. ADC Timing Diagram, Auto Triggered Conversion

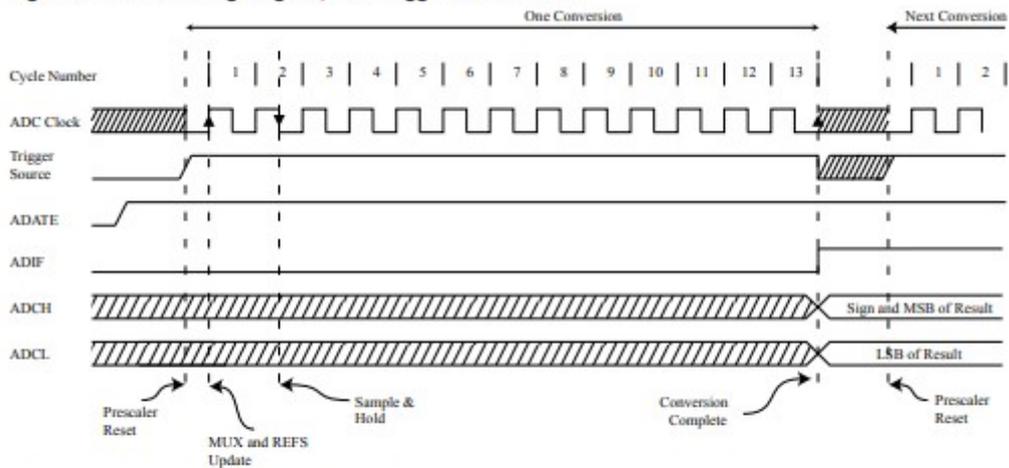


Figure 28-7. ADC Timing Diagram, Free Running Conversion

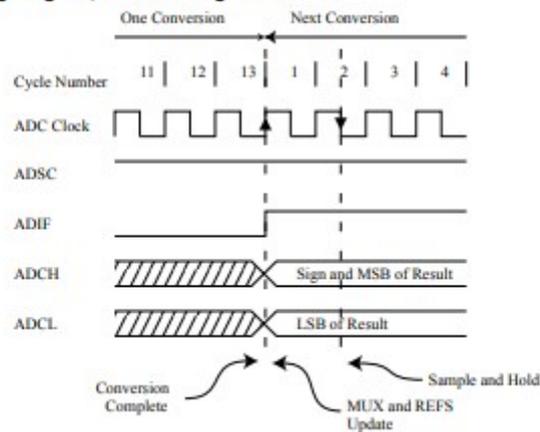


Table 28-1. ADC Conversion Time

Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5