



Universidad Nacional de Río Cuarto

Facultad de Ingeniería

“Registrador de variables eléctricas para un sistema monofásico de baja tensión”

- **Estudiante: Sr. Silva, Martin Ezequiel**
- **Tutor por la UNRC: Dr. Donolo, Pablo Daniel**
- **Cotutor por la UNRC: Mg. Stoll, Rodolfo**
- **Tutor por la Empresa: An. Galfioni, Adrian**

Tabla de contenidos

Resumen	5
1. Objetivos	7
2. Descripción de la empresa.....	9
3. Introducción	11
3.1. Solución propuesta	14
4. Medición y adaptación de señales eléctricas.....	15
4.1. Medición de variables	15
4.1.1. Medición de corriente.....	15
4.1.2. Medición de tensión	16
4.2. Etapa de adaptación sensores - procesador	16
4.3. Simulación de la etapa de adaptación de señales	17
5. Procesamiento de las señales adquiridas.....	19
5.1. Arduino Nano.....	19
5.2. Código desarrollado para Arduino.....	19
5.3. Cálculos realizados en Arduino.....	21
5.3.1. Valores eficaces.....	21
5.3.2. Potencia eléctrica.....	21
6. Comunicación e interfaz de visualización.....	23
6.1. Protocolo de comunicación	23
6.2. Placa NodeMCU	23
6.3. Base de datos	23
6.4. Interfaz web de visualización	24
6.4.1. Cálculos realizados en Grafana.....	24
6.4.1.1. Energía	24
6.4.1.2. Emisiones equivalentes de dióxido de carbono	24
6.5. Esquema de comunicación	24
7. Resultados y validación experimental.....	27
7.1. Diagrama esquemático	27
7.2. Implementación del prototipo	27
7.3. Validación en laboratorio	28
7.4. Visualización de datos en Grafana	29

7.5.	Análisis del costo del prototipo final.....	33
8.	Conclusiones y trabajos futuros	35
8.1.	Conclusiones	35
8.2.	Trabajos futuros.....	35
9.	Bibliografía.....	37
10.	Anexos.....	39
	• Anexo 1: Datasheet SCT-013.....	39
	• Anexo 2: Amplificador operacional LM358.....	41
	• Anexo 3: Arduino nano	43
	• Anexo 4: Pinout NodeMCU	45
	• Anexo 5: Código fuente arduino	47
	• Anexo 6: Código fuente NodeMCU.....	53

AGRADECIMIENTOS

A mis tutores, Dr. Donolo, Pablo y Mg. Stoll, Rodolfo, por la buena predisposición e interés al colaborar en este proyecto, además de la paciencia y continua motivación para lograr cumplir con los objetivos propuestos.

A todo el personal de la cooperativa IT10 Informática y Telecomunicaciones, en primera instancia por ofrecerme la posibilidad de desarrollar mi práctica profesional y luego por acompañarme durante todo el proceso.

Agradezco también al personal del IPSEP, por poner a mi disposición las instalaciones e instrumentos de medición, las veces que fueron necesarios. Y por último, a los compañeros, que de una u otra manera aportaron para lograr finalizar mis estudios de grado.

DEDICATORIA

A mis padres, Gladys y Carlos, las personas más importantes de mi vida. Gracias por acompañarme en cada actividad que emprendo, por el apoyo incondicional, por guiarme y enseñarme a diario, con cada charla y consejo, siempre con importantes valores y con mucho amor.

A mi hermana Luciana, por compartir lo cotidiano de nuestra familia.

Dedico este trabajo también a mis familiares y amigos, aquellos que siempre están dispuestos a tener una conversación y pasar buenos momentos.

Resumen

En el mercado actual existen diferentes tipos de medidores inteligentes, los cuales permiten al usuario conocer en cualquier instante el consumo de electricidad del hogar. Esto permite a los consumidores comprender los factores que rigen su consumo eléctrico, derivando esto en la posibilidad de implementar medidas de ahorro energético y económico. En este trabajo se presentan los resultados obtenidos en el desarrollo de un prototipo de medidor inteligente para ser aplicado en un circuito eléctrico monofásico.

El prototipo desarrollado, adapta los valores de tensión y corriente de fase para ser medidos en una placa Arduino mediante sensores comerciales y circuitos electrónicos adecuados. A partir de los registros, se calculan valores eficaces, potencia, factor de potencia, energía consumida y emisiones equivalentes de CO₂. En el desarrollo también se incluye una placa NodeMCU que permite vincular el prototipo con una red WiFi, de esta manera los resultados obtenidos se envían a una base de datos aprovechando el nodo denominado “sensar” desarrollado por la empresa IT10. La consulta de los datos almacenados se realiza mediante una plataforma denominada Grafana, que posee una interfaz web.

El prototipo propuesto se validó experimentalmente en el laboratorio en diferentes condiciones de alimentación haciendo un contraste con instrumentos de medición comerciales. Los resultados permiten observar que para la medición de tensión, el error máximo relativo al multímetro utilizado es de 2,1%. Para la medición de corriente, se concluye que para valores de aproximadamente 5 A, el error relativo a la pinza amperométrica utilizada se encuentra por encima del 8%. Tal error disminuye para valores mayores de la magnitud medida, llegando a ser próximo a un 1% para una corriente de aproximadamente 40 A.

El análisis de los costos del prototipo desarrollado resulta competitivo con las propuestas comerciales existentes.

Palabras clave: Medición de tensión, medición de corriente, circuito monofásico, Arduino, NodeMCU, ESP8266.

1. Objetivos

El objetivo general de la presente práctica profesional, consiste en diseñar e implementar un prototipo de medidor inteligente, apto para un circuito monofásico que permita adquirir las señales de tensión, corriente y calcular las potencias, la energía y las emisiones equivalentes de CO₂ a la atmósfera. Se espera también que cumpla con los siguientes requisitos:

- Fácil instalación.
- Bajo costo.
- Pequeño tamaño.
- Sencillo de implementar para su producción.
- Interfaz de lectura de los datos desde internet.

A partir de lo expresado, se definen los siguientes objetivos particulares:

- Investigar sobre productos que ofrezcan una solución similar.
- Proponer circuitos para adaptar las señales de tensión y corriente.
- Simular los circuitos en un software para identificar el de mejor desempeño.
- Determinar los parámetros a calcular a partir de las señales medidas.
- Definir como conectar el dispositivo a una red y la interfaz de visualización de la información.
- Armar el prototipo.
- Validar los resultados experimentalmente en el laboratorio.
- Proponer mejoras a partir de los resultados obtenidos.

2. Descripción de la empresa

El trabajo fue desarrollado dentro de la cooperativa de Informática y Telecomunicaciones IT10 (figura 2.1). Integrada por jóvenes profesionales de diversas áreas (Ingeniería, Informática, Comunicación), brindan soluciones tecnológicas innovadoras de manera interdisciplinaria, flexible y eficaz. Ofrece servicios relacionados con:

- Robótica Educativa
- Aplicaciones Android
- Diseño e Impresión 3D
- Capacitación
- Centrales de sensado
- Sistemas Inteligentes



Figura 2.1: Logo de la empresa cooperativa

Contacto:

Galfioni, Adrián

- **Tel.:** 358 4118240
- **E-mail:** agalfioni@it10coop.com.ar

Cooperativa de Trabajo Informática y Telecomunicaciones 10 Ltda.

- **Dirección:** Belgrano 45, Oficina 23, Río Cuarto.
- **Tel.:** 358 4329528
- **E-mail:** contacto@it10coop.com.ar
- **Sitio Web:** www.it10coop.com.ar/

3. Introducción

El mercado energético actual se enfoca hacia el uso racional y eficiente de la energía, con el objetivo de reducir costos y niveles de emisión de gases de efecto invernadero, principalmente dióxido de carbono (CO₂) [1]. Para este fin, desde hace unos años los países más desarrollados han realizado grandes inversiones en la generación de energía basada en fuentes renovables, automatización de la distribución, y lectura remota de medidores de energía [2].

En este contexto, las redes de distribución deberán actualizarse para integrar la energía proveniente de fuentes no convencionales y reforzar la gestión de la demanda. Este cambio en las infraestructuras debe apoyarse con sistemas de medida que permitan conocer los perfiles de carga de los usuarios, entendidos no solo como consumidores sino también como productores de energía, dando origen en este escenario descentralizado a una nueva figura denominada prosumidor [1]. De estos cambios surge el concepto de redes eléctricas inteligentes o *smart grids*, que se puede sintetizar como la conjunción de la red eléctrica tradicional con tecnologías modernas de la información y comunicaciones más la integración de sistemas de generación distribuida [2].

Un medidor inteligente o *smart meter*, es un tipo de medidor o contador avanzado que calcula el consumo de una forma más detallada que los medidores convencionales. Estos aparatos también ofrecen la posibilidad de comunicar esta información a través de una red hasta un centro de procesamiento de datos de la empresa distribuidora de energía local, la cual puede utilizar los datos a efectos de facturación, seguimiento, facilitar a sus usuarios un mayor control sobre sus consumos, o incluso poder ofrecer servicios personalizados a los clientes. Así mismo, estos aparatos tienen la capacidad de configurar a medida el servicio, o interrumpir el suministro de manera remota, en caso por ejemplo de que no se haya realizado el pago por el servicio contratado e igualmente restituirlo una vez que la deuda haya sido cancelada [3].

Los medidores inteligentes son dispositivos de medida capaz de calcular no sólo la energía consumida, sino también otras variables como factor de potencia, potencia activa y reactiva, valores eficaces de tensión y corriente, esto sumado al beneficio de la capacidad de almacenaje de datos. Al tener acceso a información detallada, incluso de manera remota (online) y en tiempo real, los usuarios pueden analizar las características de su consumo para un determinado período y modificar sus conductas de acuerdo a sus preferencias. Esto permite por ejemplo, estimar el costo de la próxima factura, tomar medidas de ahorro energético o modificar la tendencia hacia un consumo más responsable, incluso es posible establecer alarmas que adviertan cuando ciertos umbrales de consumo se hayan superado [2] [4] [5].

En el caso de los países en vías de desarrollo y en particular de América Latina, el panorama de las redes eléctricas inteligentes está más rezagado. En lo que respecta a Argentina, la instalación a gran escala de medidores inteligentes es un tema pendiente. Sin embargo, se han realizado experiencias piloto en localidades como Armstrong en la provincia de Santa Fe. Esta experiencia muestra que hay una tendencia positiva a fomentar la inserción de proyectos afines a esta temática [2] [6].



Actualmente, el costo de la tecnología para desarrollar medidores inteligentes disminuye rápidamente mientras que las capacidades para almacenamiento de datos y ancho de banda están mejorando exponencialmente. Sensores diminutos, de bajo costo, de alta potencia y habilitados para redes de


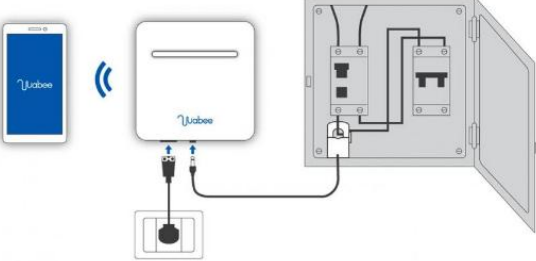


comunicación se están integrando en todo lo que nos rodea, lo que conduce a una interconectividad de activos cada vez mayor a medida que se expande internet de las cosas (IoT por sus siglas en inglés) [1].

Dentro del ámbito local, la empresa IT10 ha incursionado en proyectos relacionados a IoT, trabajando con diversos tipos de sensores para medir variables como humedad y temperatura, placas de desarrollo como Arduino, y complementando estos elementos con redes de comunicación. De esta manera, ofrecen a los clientes soluciones a medida, en donde la información de interés es recogida y posteriormente visualizada en un sitio web por ejemplo. Es en este marco es que surgió por parte de IT10 la necesidad de desarrollar un dispositivo capaz de medir las variables eléctricas asociadas a una vivienda o pequeña empresa con el fin de estimar el consumo de energía.

Por otra parte, en la Facultad de Ingeniería se desarrollan proyectos enfocados en el uso racional de la energía eléctrica, haciendo énfasis en la implementación de equipamiento de alto rendimiento y en los costos medioambientales asociados. Es por ello que se propuso que, además de obtener datos de tensión y corriente, el dispositivo calcule las potencias activa, reactiva y aparente, energía consumida y emisiones equivalentes de CO₂ en un sistema monofásico. Por lo tanto, esto fue el punto de partida para el desarrollo del presente proyecto.

Partiendo de esta necesidad se hizo una búsqueda de información relacionada a productos que ofrecen una solución similar para el problema identificado. La tabla 3.1 muestra datos comparativos, con dispositivos comerciales disponibles en Argentina en diciembre de 2018.

Tabla 3.1: Dispositivos comerciales		
Ref.	Imagen	Características
(1)		<p>Marca: HIKING</p> <p>Variables medidas: Corriente, Tensión</p> <p>Datos mostrados:</p> <ul style="list-style-type: none"> - Potencia (activa y reactiva) - Tensión - Corriente - Factor de Potencia - Frecuencia - Energía (KWh y KVArh) <p>Precio: \$2.490</p>
(2)		<p>Marca: PEACEFAIR</p> <p>Variables medidas: Corriente, Tensión</p> <p>Datos mostrados:</p> <ul style="list-style-type: none"> - Tensión - Corriente - Potencia activa - Energía (kWh) <p>Precio: \$1.699</p>

<p>(3)</p>		<p>Marca: IMAR VARIABLES MEDIDAS: Corriente, Tensión Datos mostrados:</p> <ul style="list-style-type: none"> - Tensión RMS - Corriente RMS - Potencia activa (valor instantáneo y gráfico histórico) - Factor de potencia <p>Precio: \$3.350</p>
<p>(4)</p>		<p>Marca: WABEE SMART ENERGY VARIABLES MEDIDAS: Corriente Datos mostrados:</p> <ul style="list-style-type: none"> - Potencia activa - Acumulativo de energía - Tensión (valor fijo) - Corriente - Costo estimado de la energía consumida. <p>Precio: \$5.200</p>
<p>(5)</p>		<p>Marca: SONOFF VARIABLES MEDIDAS: Corriente Datos mostrados:</p> <ul style="list-style-type: none"> - Potencia activa - Acumulativo de energía (kWh) <p>Precio: \$1.600</p>
<p>(6)</p>		<p>Marca: EFERGY VARIABLES MEDIDAS: Corriente Datos mostrados: (Actualización cada 10 seg.)</p> <ul style="list-style-type: none"> - Potencia activa - Energía (kWh) - Costo (Euros y otras divisas) - Unidades equivalentes a la huella de carbono (Kg de CO₂/ kWh). <p>Precio: \$7.230</p>

Analizando la información presentada, se pueden hacer las siguientes clasificaciones:

- **En función de las variables medidas**
 - Los dispositivos 1, 2 y 3 miden tensión y corriente.
 - Los dispositivos 4, 5 y 6 sólo miden corriente.

Los dispositivos que no miden tensión, asumen un valor fijo de este parámetro para los cálculos que realizan. Para algunos medidores, el mismo viene preestablecido de fábrica, y por ejemplo para el

dispositivo 6, el usuario mediante la misma pantalla de visualización, debe ingresar el valor medido o estimado.

- **En función de la presentación de los datos**
 - Los dispositivos 1 y 2 sólo muestran información mediante una pantalla LCD.
 - Los dispositivos 3, 4 y 5 utilizan una aplicación para teléfonos móviles.
 - El dispositivo 6, mediante una pantalla y también por una plataforma online.

- **En función de la máxima corriente admisible y forma de medición**
 - El dispositivo 1: 65 A (intercalado en el circuito principal, fase y neutro).
 - El dispositivo 2: 100 A (toroide) y 20 A (resistencia shunt).
 - El dispositivo 3: 100 A, utiliza una pinza.
 - El dispositivo 4: 60 A, pinza.
 - El dispositivo 5: 16 A, intercalado en el circuito a controlar.
 - El dispositivo 6: (50 mA – 95 A), pinza.

3.1. Solución propuesta

En función de los objetivos planteados inicialmente y la información presentada en la sección anterior, se expone en este informe la solución propuesta para el desarrollo de un prototipo de medidor de energía inteligente.

En la bibliografía se encuentran algunas propuestas de medidores inteligentes de energía que también utilizan Arduino, en combinación con tecnología GSM [5] y PLC (Power Line Comunicación) [7]. En base a los resultados obtenidos en [5] y [7] se propone avanzar en un registrador con el uso de una placa Arduino. Esta propuesta resulta conveniente para los objetivos planteados en el trabajo y los requerimientos acordados con la Cooperativa IT10, enfocados en que las aplicaciones tengan una excelente experiencia para el usuario con un diseño amigable e innovador. El resto del informe se organiza de la siguiente manera:

Inicialmente se muestran las etapas de medición de potencia de las señales de corriente y tensión con los dispositivos seleccionados en conjunto con la empresa. Se presentan los circuitos de adaptación de señal propuestos y los resultados de la simulación realizada en Proteus ®.

Posteriormente se presenta la etapa con el procesamiento de las señales, utilizando una placa Arduino Nano para registrar las señales de tensión y corriente. Con el mismo Arduino y a partir de las señales muestreadas se realiza el cálculo de: valores eficaces (RMS) de tensión y corriente, potencias (activa, reactiva y aparente) y factor de potencia. Los resultados de estas operaciones se envían al módulo NodeMCU, que permitirá conectar el prototipo a una red WiFi, con el fin de guardar los registros en una base de datos. El usuario podrá acceder a todas las variables en un entorno web, perteneciente a la plataforma Grafana.

Finalmente se presenta el prototipo implementado y validado. Se realiza una experiencia de comparación de las mediciones para las magnitudes de tensión y corriente con instrumentos de campo comerciales. También se muestra para la serie de mediciones realizadas, la información desde la interfaz web. El trabajo se finaliza con las conclusiones y algunas indicaciones para mejorar el prototipo en un futuro.

4. Medición y adaptación de señales eléctricas

En esta sección, se muestra en un primer punto la forma de medición de las variables y posteriormente la adaptación de las señales, entre las etapas de medición y procesado de las mismas. El esquema resultante es modelado y simulado en un software.

4.1. Medición de variables

En este apartado se presentan los sensores y los respectivos circuitos asociados para la medición de tensión y corriente. Esto constituye la primera instancia de desarrollo.

4.1.1. Medición de corriente

Para esta medición se utiliza un transformador de corriente de la serie SCT-013 perteneciente a la marca YHDC (Anexo 1). Estos elementos poseen un núcleo partido al igual que una pinza amperométrica. La corriente que circula por el cable a medir actúa como el devanado primario, por lo que la posibilidad de uso sin interrumpir o seccionar el circuito en cuestión, sumado a la aislación galvánica que conlleva, supone una ventaja frente a otras alternativas.

De las diversas opciones que ofrece la serie mencionada, el modelo seleccionado en conjunto con la empresa fue el SCT-013-100, presentado en la figura 4.1. Esta pinza tiene la ventaja frente a otras de la misma familia en que posee salida en corriente, lo que minimiza la posibilidad de ingreso de ruido en la señal medida. La pinza posee un rango de 0 a 100 A de corriente de entrada, y salida de 0 a 50 mA.

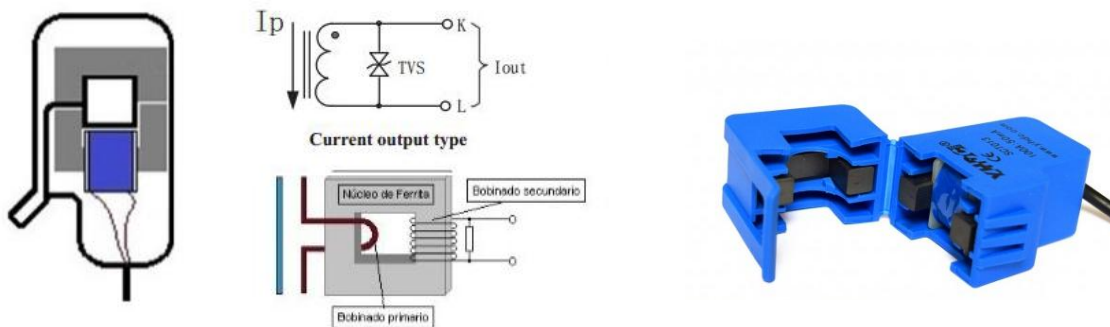


Figura 4.1: Sensor de corriente SCT-013-100

Para adaptar la señal de corriente y poder registrar con el microcontrolador, a la salida del sensor se añade una resistencia, con el objetivo de obtener un valor de tensión proporcional a la corriente del secundario. Este valor de tensión será el que ingrese a la etapa de adaptación de señales. El valor seleccionado es de 22 ohm, para obtener un 1V eficaz para la máxima corriente (50mA). El esquema propuesto se muestra en la figura 4.2.

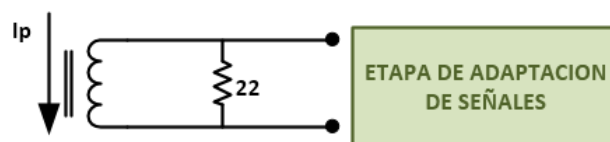


Figura 4.2: Medición de corriente

4.1.2. Medición de tensión

En la figura 4.3 se presenta el esquema de esta parte de la implementación. Se utiliza un transformador de 220/12 Vac (300 mA) para realizar la medición desde la red eléctrica de 50Hz. En su salida se dispone de un divisor resistivo conformado por una resistencia de 10 (k Ω) y otra de 820 (Ω), tomando a bornes de esta última la tensión para el respectivo circuito de adaptación de señales. Nuevamente, estos valores se seleccionan con la premisa de lograr aproximadamente 1 V eficaz para la máxima tensión de entrada, para la cual se considera una sobretensión de 15% de la nominal (Vn 220 V).

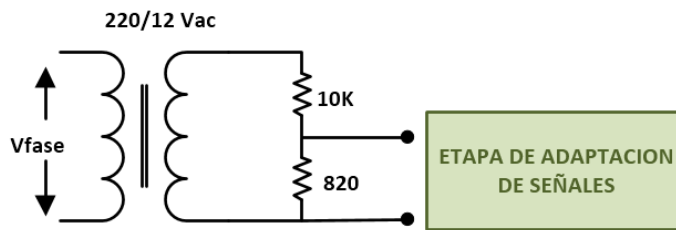


Figura 4.3: Medición de tensión

4.2. Etapa de adaptación sensores - procesador

Las señales provenientes de las etapas de medición presentan un semiciclo negativo, por lo cual es necesario acondicionar los valores a un rango manejable para el microcontrolador utilizado, esto es de 0 a +5V. Este dispositivo de adaptación de señales no está disponible comercialmente, por lo tanto, fue necesario desarrollarlo. En la figura 4.4 se muestra el circuito propuesto para adaptar las señales. Se utilizó la misma propuesta para las mediciones de tensión y corriente.

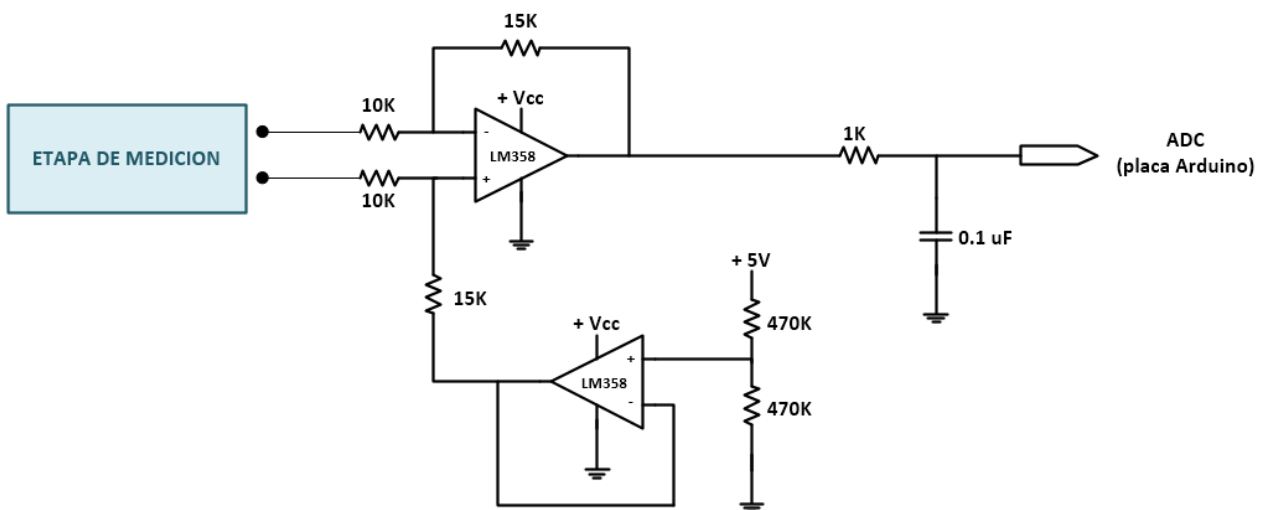


Figura 4.4: Adaptación de señales (Entre etapa de medición y procesamiento)

El esquema de la figura 4.4 presenta:

- Etapa de amplificación diferencial. Se utiliza un amplificador operacional LM358 (Anexo 2) en modo diferencial, cuya ganancia se configura a un valor de 1,5 utilizando resistencias de 10 (k Ω) y 15 (k Ω).

- Tensión de referencia (offset) de 2,5V. Para desplazar el valor medio de las señales se incluye un offset, el cual se obtiene mediante un divisor resistivo con dos componentes de 470 (k Ω), tomando 5V desde la placa Arduino (presentada en el capítulo 5). Para asegurar un valor más estable de tensión, el punto medio de las resistencias se lleva a la entrada de un amplificador operacional configurado como seguidor de tensión (este circuito se aplica una sola vez).
- A la salida del circuito de amplificación, previo al ingreso de la señal al microcontrolador, se agrega para evitar ruidos, una resistencia de 1 (k Ω) y un capacitor cerámico 104 (0,1 uF), configurando de esta manera un filtro.
- Se utilizarán dos pines de la placa Arduino pertenecientes al convertor analógico-digital (ADC) del microcontrolador. El pin A0 para la señal de corriente y el pin A4 para la señal de tensión.

4.3. Simulación de la etapa de adaptación de señales

El esquema propuesto para la adaptación de señales se simuló y luego se validó experimentalmente, de manera de garantizar su correcto desempeño.

En este apartado se presentan los resultados de la simulación de los circuitos propuestos. Para tal fin se utilizó el software *Proteus* ®. De esta manera, el circuito presentado en la figura 4.4 queda como se muestra en la figura 4.5 para su simulación.

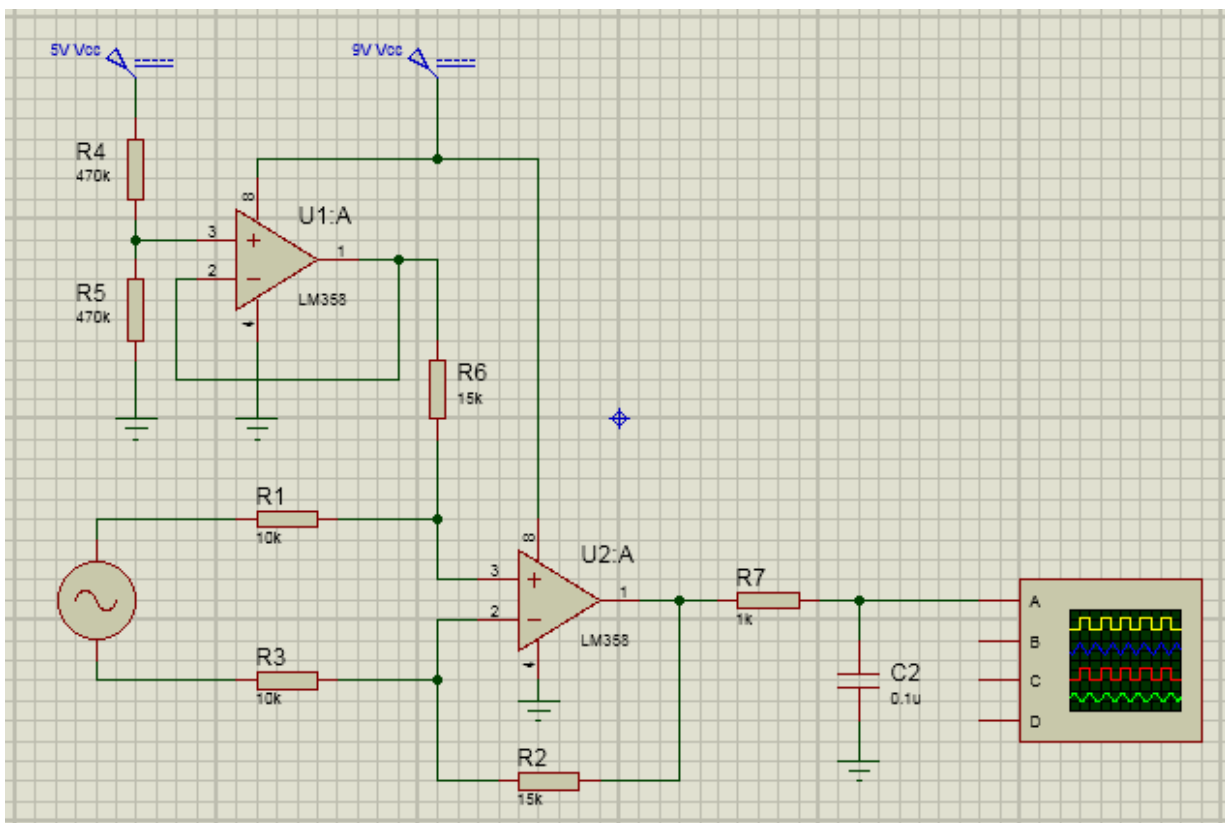


Figura 4.5: Circuito para simulación

En la figura 4.5 se observa que ambos LM358 se alimentan con una tensión de 9V. Esto es para evitar que la salida se sature por debajo de los 5V.

En las figuras 4.6 y 4.7 se muestran las señales resultantes para las señales de corriente y tensión respectivamente. En ambos casos se consideran los máximos valores de entrada, es decir 100 A para la corriente de fase y 253 V para la tensión.

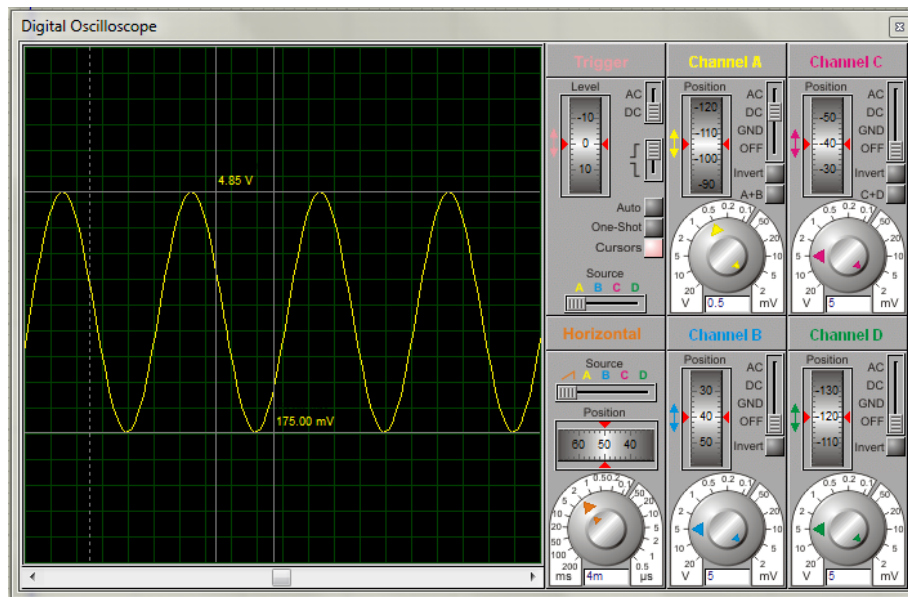


Figura 4.6: Salida de corriente

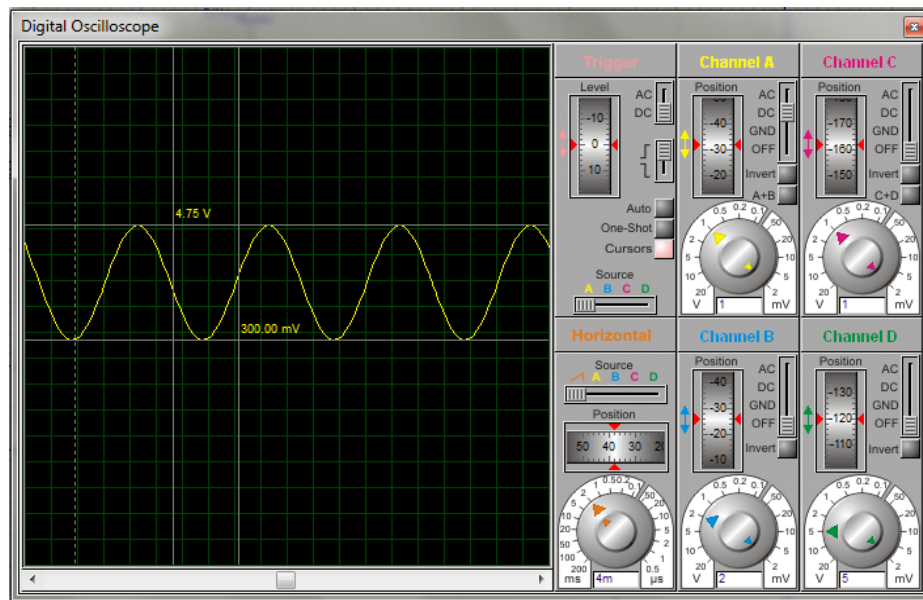


Figura 4.7: Salida de tensión

Se observa que las señales que ingresarán al microcontrolador, se encuentran dentro de los límites establecidos de 0 a 5 V, considerando la situación más desfavorable.

Para la medición de corriente, los valores picos superior e inferior son 4,85 V y 0,175 V respectivamente. Siendo para el caso de la tensión de 4,75 V y de 0,3 V.

5. Procesamiento de las señales adquiridas

En este capítulo se presenta la placa Arduino utilizada, el código desarrollado para la adquisición de datos y el cálculo de valores eficaces, potencia y factor de potencia.

5.1. Arduino Nano

Arduino es una plataforma electrónica de código abierto, la cual está basada en hardware y software libre [8]. Se optó por una placa Arduino Nano (anexo 3) debido a sus dimensiones más pequeñas frente a otras alternativas disponibles. La misma se muestra en la figura 5.1 y reúne las siguientes características principales:

- Microcontrolador: ATmega328
- Tensión de operación: 5V
- Tensión de alimentación (Recomendado): 7-12V
- Memoria Flash: 32KB
- Frecuencia de trabajo: 16MHz
- Conversor analógico / digital ADC: 10 bits

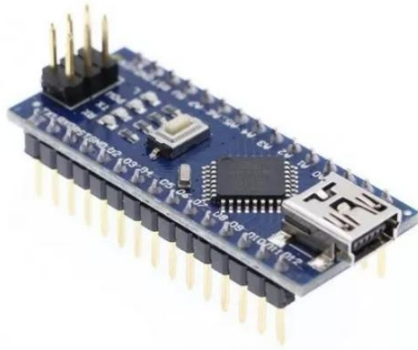


Figura 5.1: Arduino Nano

5.2. Código desarrollado para Arduino

Para la programación del microcontrolador se utiliza el lenguaje de programación Arduino y el software Arduino IDE (Entorno de desarrollo integrado).

El código utilizado en la programación del microcontrolador de la placa Arduino se encuentra en el Anexo 5. La figura 5.2 muestra un diagrama de flujo que posee las partes generales del mismo.

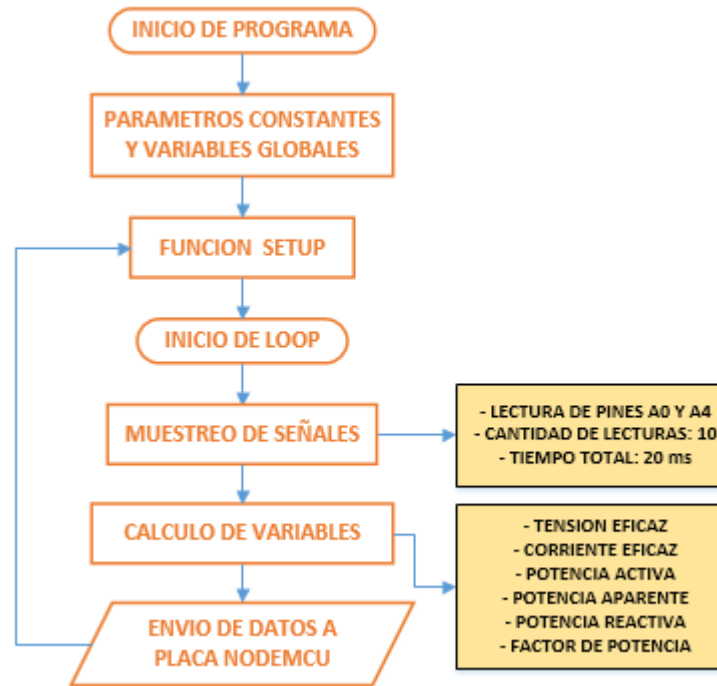


Figura 5.2: Diagrama de flujo de código de placa Arduino

En la figura 5.2 se destaca la etapa de muestreo dentro de la función loop. En el loop se leen los pines A0 y A4, pertenecientes al convertor analógico-digital del microcontrolador (ADC), y se realiza un muestreo de las señales. La cantidad de muestras por segundo (N), es lo que determina la frecuencia de muestreo. Atendiendo a las limitaciones de almacenamiento del microcontrolador, N se configuró a 10, lo cual da una frecuencia de muestreo de 500 Hz y de acuerdo con teorema del muestreo de Nyquist una frecuencia máxima de visualización de 250Hz.

El convertor analógico digital que dispone el procesador utilizado es de 10 bits, y en función de que la tensión de referencia es de 5V, esto da una resolución de 4,88 mV por bit.

A partir de la digitalización se obtienen dos arreglos o vectores, v_n e i_n , ambos compuestos por 10 elementos (N) y correspondientes a las señales de tensión y corriente respectivamente.

Los valores en bits obtenidos en el proceso de muestreo deben ser modificados en función de los parámetros que caracterizan las etapas de medición, adaptación y procesado de las señales, de manera tal que se haga una regresión a los valores reales de tensión y corriente del sistema medido.

Cada muestra en bits será corregida en función de: la tensión de referencia y de la cantidad de bits del ADC de Arduino, la tensión de offset y ganancia del amplificador operacional (utilizadas en la etapa de adaptación de señales), y además por una constante de regresión asociada a los dispositivos de medición utilizados (para la medición de corriente, se considera la relación de transformación del sensor y la resistencia de carga empleada y para la medición de tensión, la relación del transformador y el respectivo divisor resistivo).

En la figura 5.3 se presenta un ejemplo de los valores registrados en base decimal de tensión y corriente (cada componente de los vectores v_n e i_n). Estos vectores, son el punto de partida para realizar los cálculos presentados en la siguiente sección.

```

Iniciando muestreo
Tiempo de muestreo: 20 milisegundos

tensión: -303.22
tensión: -157.97
tensión: 24.72
tensión: 180.46
tensión: 289.02
tensión: 301.00
tensión: 188.69
tensión: 2.26
tensión: -155.72
tensión: -280.76

corriente: -8.28
corriente: -4.14
corriente: -1.47
corriente: 3.26
corriente: 4.74
corriente: 5.33
corriente: 1.49
corriente: -1.47
corriente: -2.36
corriente: -8.87
    
```

Figura 5.3: Valores registrados en base decimal de tensión y corriente

5.3. Cálculos realizados en Arduino

Se presentan los parámetros a calcular con el microcontrolador de la placa Arduino.

5.3.1. Valores eficaces

Se denomina valor eficaz o *rms* (del inglés, root mean square) al valor cuadrático medio de una magnitud eléctrica variable.

Se calculan los valores eficaces de tensión y corriente según las ecuaciones 5.1 y 5.2 respectivamente. Siendo el resultado de aplicar la raíz cuadrada al promedio de la sumatoria de cuadrados de los elementos de los arreglos v_n e i_n [9].

$$V_{rms} = \sqrt{\frac{1}{N} \sum_{n=1}^N v_n^2} \Rightarrow \sqrt{\frac{v_1^2 + v_2^2 + \dots + v_{10}^2}{10}} \quad (5.1)$$

$$I_{rms} = \sqrt{\frac{1}{N} \sum_{n=1}^N i_n^2} \Rightarrow \sqrt{\frac{i_1^2 + i_2^2 + \dots + i_{10}^2}{10}} \quad (5.2)$$

5.3.2. Potencia eléctrica

La potencia en un sistema eléctrico ideal se puede calcular a partir de las tensiones y corrientes medidas y considerando las relaciones trigonométricas entre ellas. La figura 5.4 muestra la relación entre las potencias activa, reactiva y aparente [9].

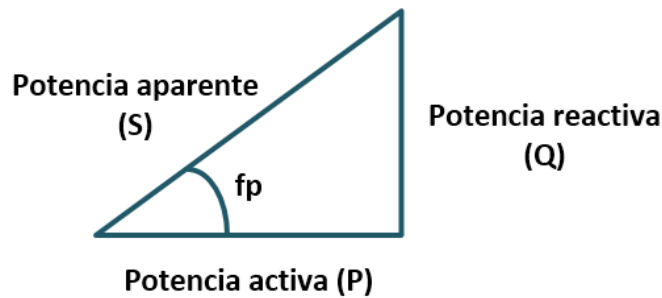


Figura 5.4: Triángulo de potencias

- **Potencia activa**

Este parámetro se obtiene según la ecuación 5.3, como el promedio de la sumatoria del producto punto a punto de los vectores v_n e i_n [9].

$$P = \frac{1}{N} * \sum_{n=0}^N v_n * i_n \quad (5.3)$$

- **Potencia aparente**

Se obtiene como la suma vectorial de las potencias activa y reactiva, tal como se muestra en la ecuación 5.4, partiendo del triángulo rectángulo presentado en la figura 5.4 [9].

$$S = \sqrt{P^2 + Q^2} \quad (5.4)$$

También la potencia aparente es el resultado del producto de los valores eficaces de tensión y corriente como se presenta en la ecuación 5.5, utilizando este cálculo en el proyecto.

$$S = V_{rms} * I_{rms} \quad (5.5)$$

- **Potencia reactiva**

Se obtiene según la ecuación 5.6, como la raíz cuadrada de la resta de los valores cuadráticos de S y P, resultado de despejar Q de la ecuación 5.4 [9].

$$Q = \sqrt{(S)^2 - (P)^2} \quad (5.6)$$

- **Factor de potencia**

El factor de potencia se define como, la relación entre potencia activa y aparente. Matemáticamente queda expresado como se muestra en la ecuación 5.7 [9].

$$fp = \frac{P}{S} \quad (5.7)$$

6. Comunicación e interfaz de visualización

En este capítulo se desarrolla la solución propuesta para conectar el prototipo con una red WiFi, el protocolo de comunicación empleado para almacenar la información en una base de datos y la interfaz web de visualización.

6.1. Protocolo de comunicación

Para la comunicación entre el prototipo de medidor de energía y la base de datos, se utilizará el protocolo *MQTT* (*Message Queue Telemetry Transport*). MQTT es un protocolo de red liviano y flexible que ofrece el equilibrio ideal para los desarrollos de IoT y que funciona a su vez sobre el protocolo TCP/IP [10].

Un sistema MQTT consiste en clientes que pueden ser publicadores de información o suscriptores y se comunican con un servidor, comúnmente llamado *broker*. La información que se transmite se organiza en temas o *topics*. Un topic se representa mediante una cadena de información y tiene una estructura jerárquica separada mediante barras '/' [10].

El broker es un software que implementa el protocolo y se encarga de gestionar los mensajes. Cuando un publicador tiene un nuevo dato para distribuir, envía el mensaje con un determinado topic al broker conectado. El broker luego redirige la información a cualquier cliente que se haya suscrito a ese topic, logrando de esta manera un desacoplo entre el receptor y el transmisor.

6.2. Placa NodeMCU

Debido a que la placa Arduino no posee conexión a internet, se propuso como solución agregar un dispositivo externo llamado NodeMCU que sí permite la vinculación con una red Wi-Fi. La placa NodeMCU se muestra en la figura 6.1.



Figura 6.1: Placa NodeMCU v1.0

La principal característica de NodeMCU es que incorpora el chip ESP8266. Además de poseer un microcontrolador, ESP8266 es un chip de bajo costo Wi-Fi compatible con el protocolo TCP/IP [11]. La disposición de los pines de la placa (*pinout*) se puede ver en el anexo 4.

6.3. Base de datos

En este proyecto se utilizará *InfluxDB* para almacenar todos los resultados obtenidos por el prototipo de medidor de energía. InfluxDB es un servidor de base de datos de código abierto. Se encuentra optimizado para el almacenamiento rápido y de alta disponibilidad de datos de series temporales en

campos como el monitoreo de operaciones, la métrica de aplicaciones, los datos de sensores de internet de las cosas y el análisis en tiempo real [12].

6.4. Interfaz web de visualización

Para visualizar la información, se utiliza la plataforma *Grafana*. Grafana es una herramienta de código abierto que permite la visualización de datos métricos mediante gráficos comúnmente llamados *dashboards*. Estos gráficos se pueden crear a partir de múltiples fuentes, incluidas bases de datos de series temporales como InfluxDB.

6.4.1. Cálculos realizados en Grafana

La plataforma utilizada, permite realizar operaciones matemáticas con la información que consulta a la base de datos. Este apartado describe como se calculan la energía y las emisiones equivalentes de CO₂ a partir del dato de potencia activa. Se optó por realizar estos cálculos en el entorno gráfico de Grafana debido a que permite modificar de una manera simple los valores de las constantes asociadas.

6.4.1.1. Energía

Se obtiene la energía consumida a partir del valor de potencia activa, multiplicado por una constante que tiene en cuenta una actualización del valor de P cada 1 minuto (60 valores por hora), tal como se observa en la ecuación 6.1.

$$Energía = P * \frac{1}{60} \quad (6.1)$$

6.4.1.2. Emisiones equivalentes de dióxido de carbono

Se calcula un valor representativo de la cantidad de emisiones de CO₂ que resulta de la producción de una unidad de energía eléctrica de la red Argentina. Por lo tanto, se obtiene al multiplicar el valor de energía consumido en un determinado período por un valor constante, el cual se extrae del último informe “Cálculo factor de emisión CO₂ 2015 (02/11/2016)” perteneciente a la Secretaría de Energía de la Nación **¡Error! No se encuentra el origen de la referencia..**

En la ecuación 6.2 se muestra el cálculo asociado, la unidad de medida es kilogramos de dióxido de carbono por kilowatt-hora (kg CO₂ /kWh) **¡Error! No se encuentra el origen de la referencia..**

$$Emisiones_{CO_2} = 0,535 * Energía \quad (6.2)$$

6.5. Esquema de comunicación

La figura 6.2 presenta el esquema de comunicación para el prototipo desarrollado entre la etapa de medición y adaptación de señales y el entorno de Grafana.

En la figura se ve que los resultados de los cálculos realizados en la placa Arduino, se envían mediante comunicación serial a la placa NodeMCU. En NodeMCU los datos recibidos son asignados cada uno a un topic en función del parámetro que representan (valores eficaces, potencias, factor de potencia) y luego se envían al broker. Es decir que en la arquitectura MQTT empleada, es el ESP8266 el cliente que se encarga de publicar los datos.

Un cliente suscriptor es el nexo entre el broker y la base de datos *InfluxD*. Finalmente en Grafana, se crean paneles de visualización para cada variable consultando la información a la base de datos.

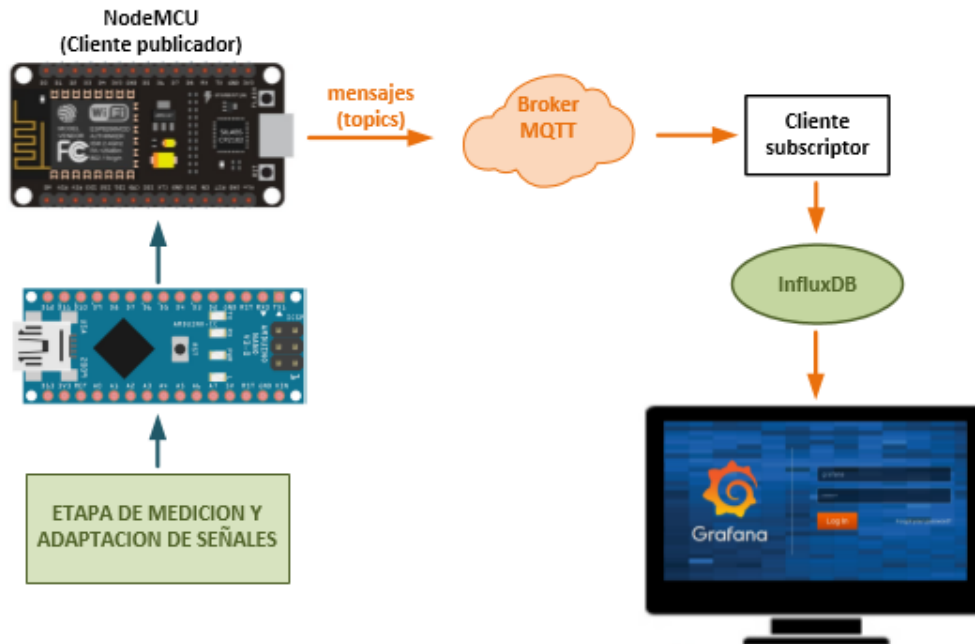


Figura 6.2: Esquema de comunicación entre la etapa de medición y adaptación de señales y la interfaz de visualización

7. Resultados y validación experimental

En este capítulo se presenta el diagrama esquemático final y el prototipo implementado. Luego se muestran los resultados de una experiencia en laboratorio donde se realizó un contraste para medición de tensión y corriente entre el prototipo e instrumentos comerciales. Se muestran también los resultados obtenidos en Grafana para las mediciones realizadas. En la parte final del capítulo se analizan los costos del prototipo, debido a que uno de los objetivos requería un bajo valor comercial de la propuesta.

7.1. Diagrama esquemático

En la figura 7.1 se muestra el diagrama esquemático del prototipo desarrollado. En el diagrama se destacan las placas Arduino Nano y NodeMCU, dos amplificadores operacionales configurados en modo diferencial y otro en modo seguidor de tensión.

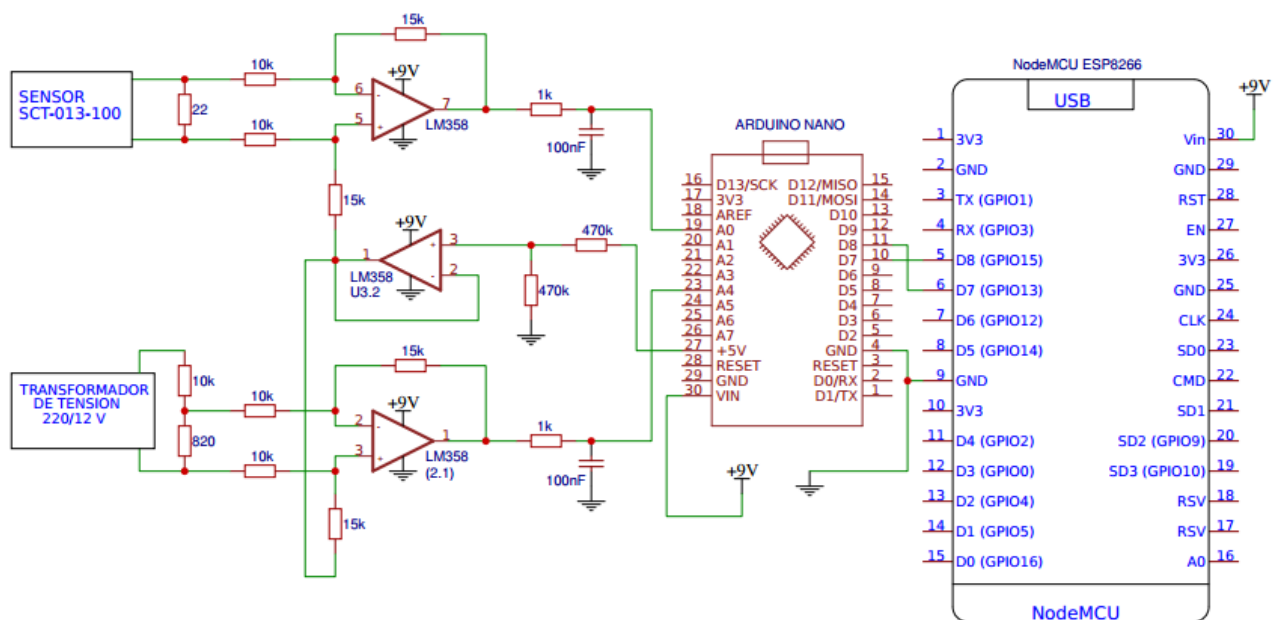


Figura 7.1: Diagrama esquemático del prototipo desarrollado

7.2. Implementación del prototipo

La figura 7.2 muestra el prototipo presentado en la figura 7.1 implementado sobre tres placas experimentales.

En la figura 7.2 se observan:

- 1- Transformador de tensión 220/12 V.
- 2- Fuente Step Up o Booster (Este tipo de fuente permite obtener a la salida una tensión superior a la de entrada; se encuentra configurada a 9 V, que es el valor seleccionado de alimentación para la placa Arduino, NodeMCU y los amplificadores operacionales).
- 3- Arduino Nano.
- 4- Fuente 5V – 2000 mA.
- 5- NodeMCU v1.0.
- 6- Sensor SCT 013-100.

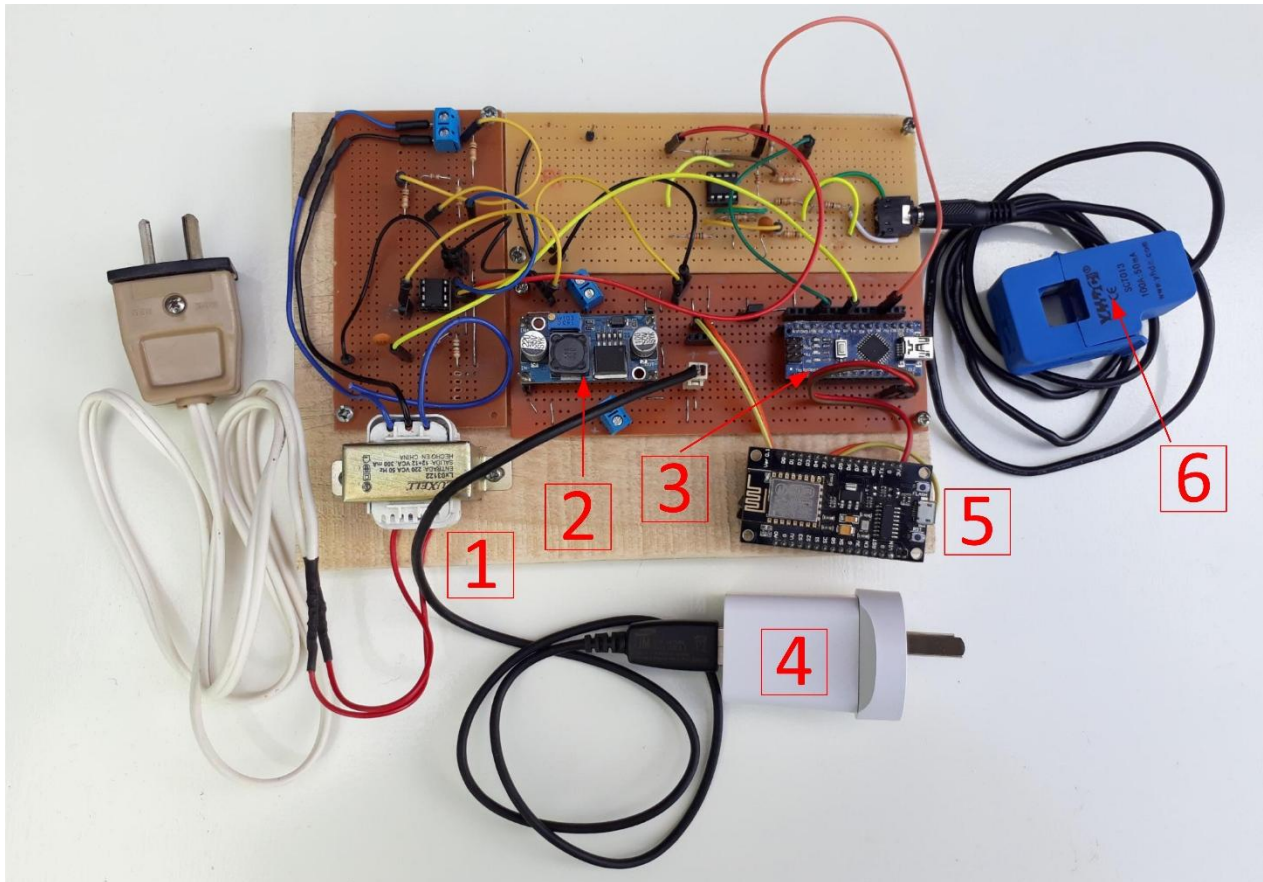


Figura 7.2: Implementación del medidor propuesto

Los demás componentes se corresponden a los circuitos de adaptación de señales (amplificadores operacionales, resistencias), y se agregan borneras de conexión, cables o *jumpers*, jack hembra 3,5 mm (para el sensor de corriente) y los zócalos correspondientes para montar los operacionales, la placa Arduino y la fuente booster sobre las placas experimentales.

7.3. Validación en laboratorio

Con el objetivo de contrastar las mediciones realizadas por el prototipo, se propuso realizar la validación de tensión y corriente con los siguientes instrumentos de campo (en las tablas 7.1 y 7.2 se muestran la resolución y exactitud de ambos instrumentos para los rangos en que fueron utilizados durante la experiencia [14] [15]):

- Multímetro FLUKE 87

Tabla 7.1: Perteneciente al multímetro 80 Series V Industrial True-RMS

Rango	Resolución	Exactitud
600 V (AC)	0,1 V	0,7%

- Pinza amperométrica PROVA 11

Tabla 7.2: Perteneciente a la pinza amperométrica PROVA 11

Rango	Resolución	Exactitud
30 A (AC)	10 mA	$\pm 2.0\% \pm 5 \text{dpts}$

En la experiencia se midieron las tensiones y corrientes de un banco de resistencias. El procedimiento se realizó de la siguiente manera:

- Se utilizaron 2 cargas resistivas de 22 (Ω), configurando un valor total de 44 y 11 (Ω) para las resistencias en serie y en paralelo respectivamente.
- Para la carga de 44 (Ω), se tomaron datos para 1, 2 y 3 vueltas del conductor de fase tanto en la pinza amperométrica como en el sensor de corriente. Para la carga de 11 (Ω), se tomaron datos para 1 y 2 vueltas solamente.

Los resultados obtenidos se muestran en la tabla 7.3.

Dispositivos comerciales		Valores medidos		Carga (Ω)	n° de vueltas	Errores relativos porcentuales	
Tensión	Corriente	Tensión	Corriente			Tensión	Corriente
224,4	5,16	224,31	5,6	44	1	0,04	8,53
225	10,34	226,59	10,18	44	2	0,71	1,55
224	15,51	219,14	14,8	44	3	2,17	4,58
214,6	19,4	215,18	19,94	11	1	0,27	2,78
215	38,02	217,93	38,13	11	2	1,36	0,29

Haciendo un análisis de la información mostrada se tiene lo siguiente:

- El prototipo presenta un error relativo máximo a los instrumentos utilizados del 2,17% para la medición de tensión.
- Para corrientes de valores cercanos a 5 A, el error se encuentra por encima del 8%.
- El error disminuye significativamente para valores próximos a 40 A, llegando incluso a obtener puntos inferiores a un 1%.

7.4. Visualización de datos en Grafana

Las figuras 7.3 a 7.10 muestran la representación de los datos obtenidos durante el proceso de validación en Grafana.

En la figura 7.3, se presentan los paneles o *dashboards* que muestran el último dato recibido para las tres variables seleccionadas, tensión, corriente y potencia activa.



Figura 7.3: Último valor actualizado – Tensión; Corriente; Potencia activa

En las figuras 7.4 a 7.10, se muestran los paneles con la información a modo de gráfico histórico, con las variables registradas en función del tiempo.

A modo de ejemplo, se toman los valores resaltados de la tabla 7.3 (correspondientes a la segunda fila de datos medidos), para hacer un seguimiento en los diferentes gráficos y verificar la correspondencia con las ecuaciones presentadas en las secciones 5.2 y 6.4.1.

El valor de tensión medido de 226,59 V se corresponde con una corriente de 10,18 A para el mismo instante de tiempo. Esto se verifica observando las figuras 7.4 y 7.5 que muestran los valores eficaces de tensión y corriente respectivamente.

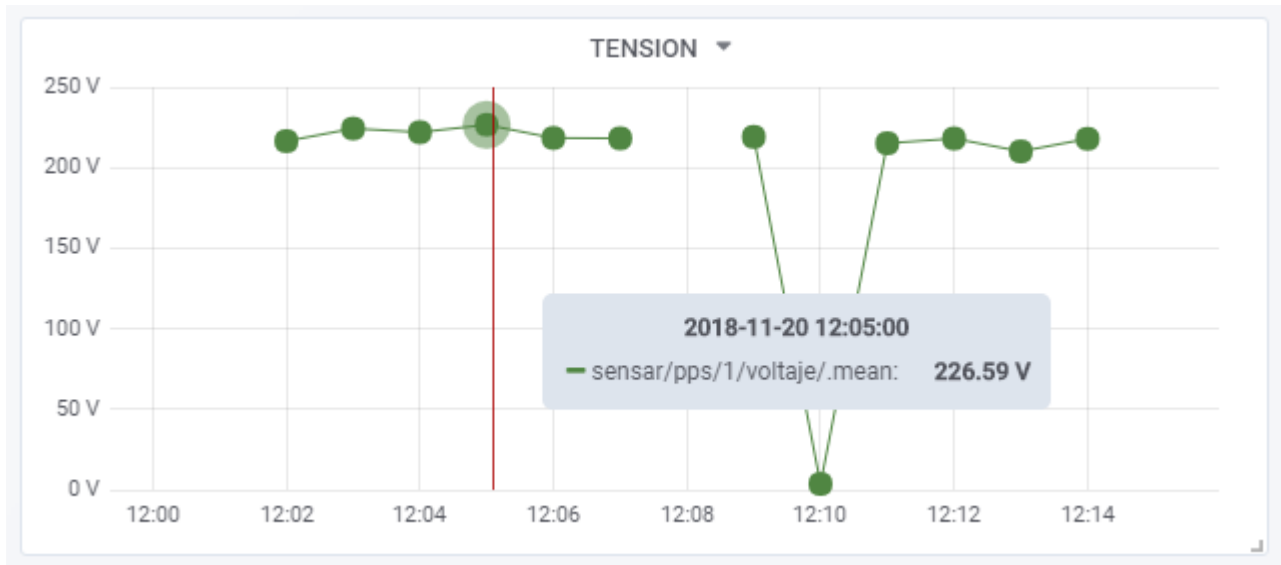


Figura 7.4: Gráfico Histórico – Tensión eficaz

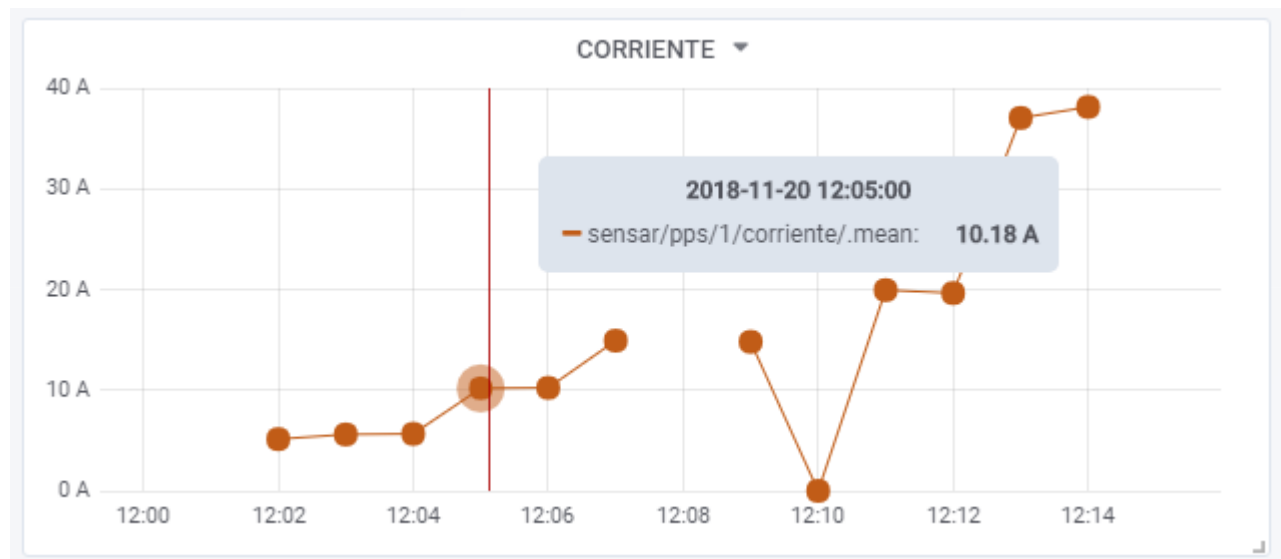


Figura 7.5: Gráfico Histórico – Corriente eficaz

En la figura 7.6 se muestran los valores de potencia activa para cada instante. Para el ejemplo, el valor es de 2,25 kW.

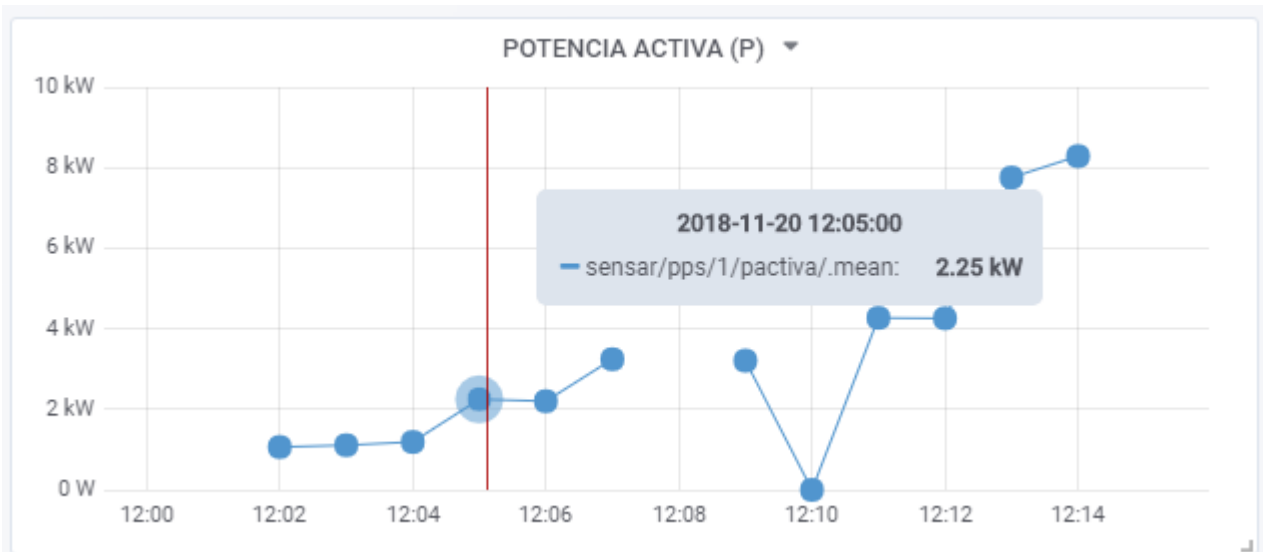


Figura 7.6: Gráfico Histórico – Potencia activa

La figura 7.7 presenta los valores correspondientes al factor de potencia. Es interesante destacar que se verifica la correcta relación entre el valor de potencia activa y el de potencia aparente mostrado en la figura 7.10 ($2,25 \text{ kW} / 2,31 \text{ kVA} = 0,97$).

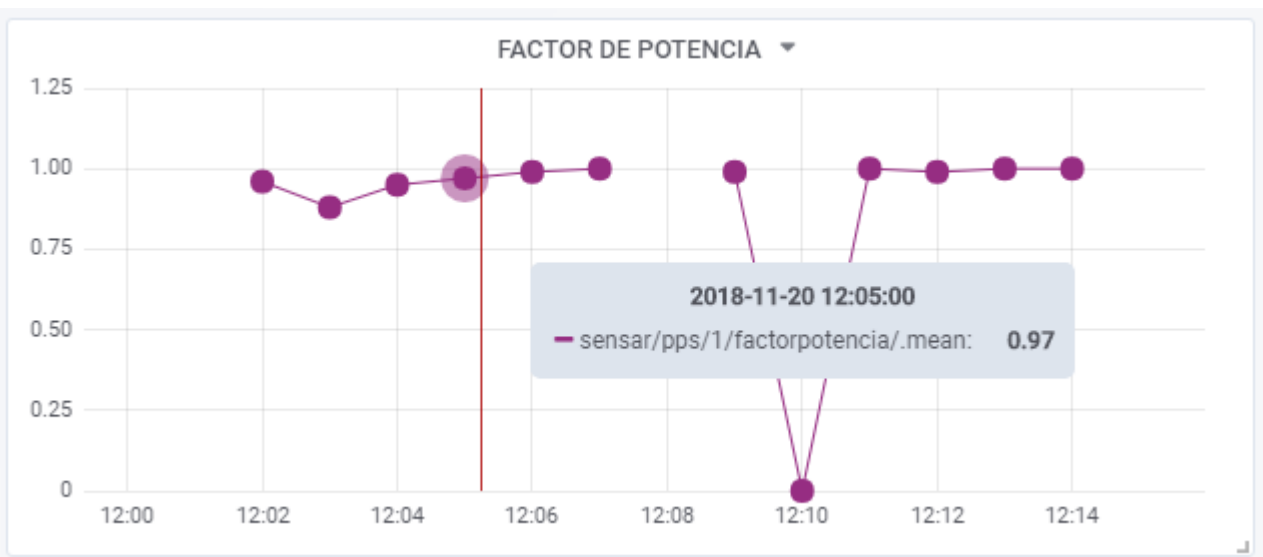


Figura 7.7: Gráfico Histórico – Valores de factor de potencia

En los paneles mostrados en las figuras 7.8 a 7.10, se incluye en la parte inferior, el valor del total correspondiente a la suma de los valores graficados.

La energía consumida para cada minuto se muestra en la figura 7.8. Para el ejemplo, un consumo de 2,25 kW durante 1 minuto, representa 0,0375 kWh. La cantidad total de energía consumida es de 0,6472 kWh.

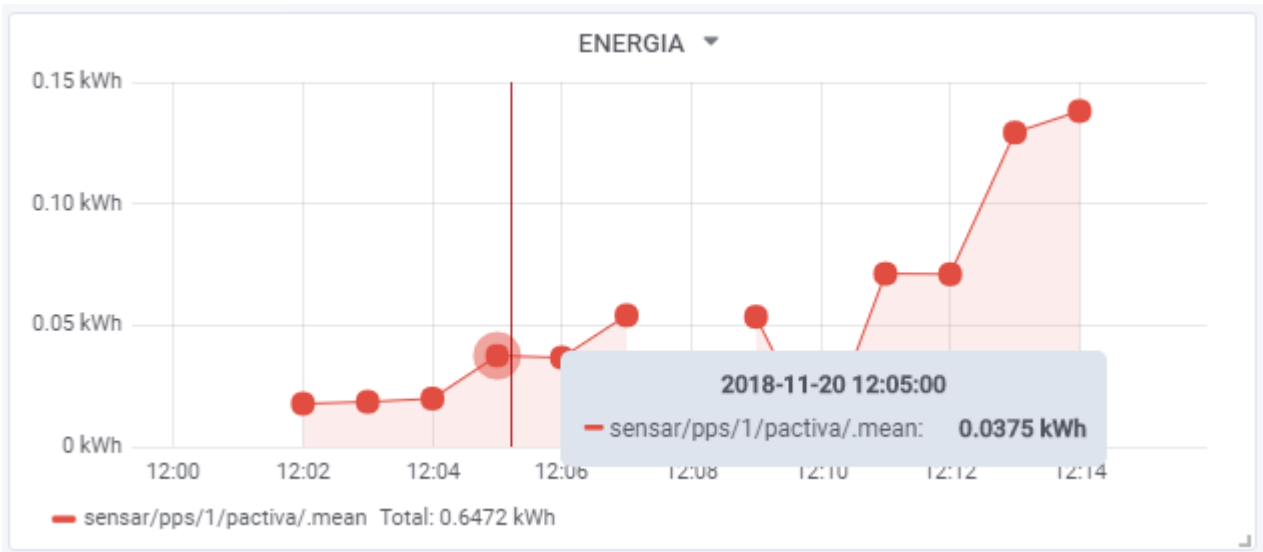


Figura 7.8: Gráfico Histórico – Energía consumida

La energía eléctrica consumida tiene asociadas emisiones de CO₂ de acuerdo con la matriz energética que la producen. De acuerdo con los datos de Argentina, el factor de relación es de 0,535 kg CO₂ /kWh.

La figura 7.9 muestra los resultados de emisiones durante el período de validación experimental en función de la energía consumida. Se observa que los valores para el ejemplo seguido y el total son de 0,02 y 0,3463 kg CO₂ /kWh respectivamente.

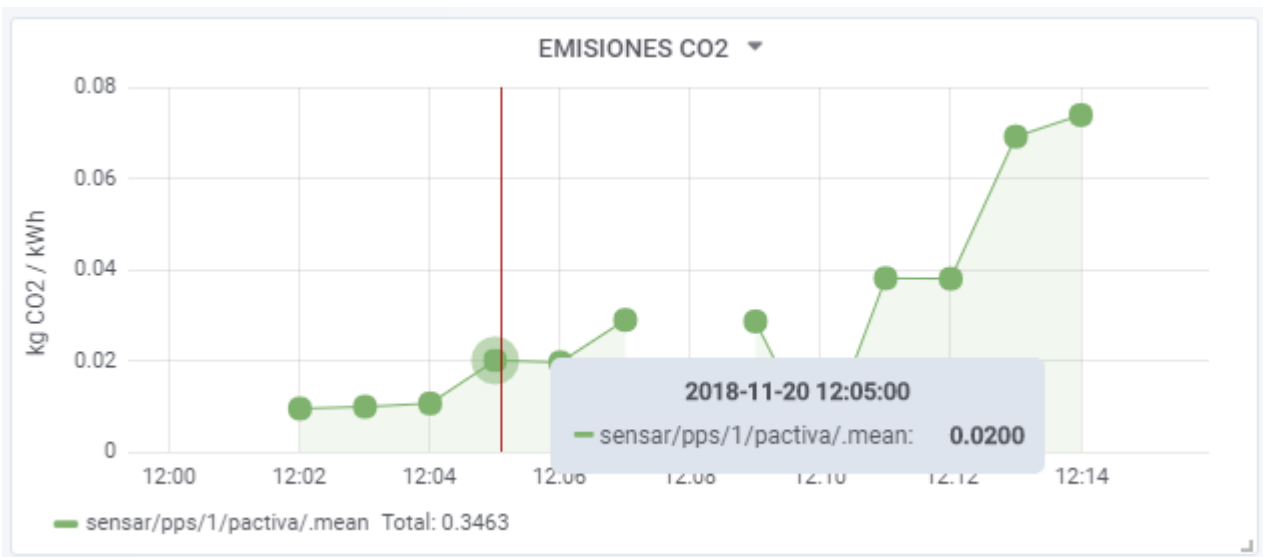


Figura 7.9: Valores estimados de emisiones de CO₂

Por último, la figura 7.10 presenta los datos relativos a la potencia aparente, verificándose para el ejemplo tratado que el valor obtenido se corresponde con el producto de los valores eficaces de tensión y corriente ($226,59 \text{ V} * 10,18 \text{ A} = 2,31 \text{ kVA}$). Siendo el valor total de 39,33 kVA.

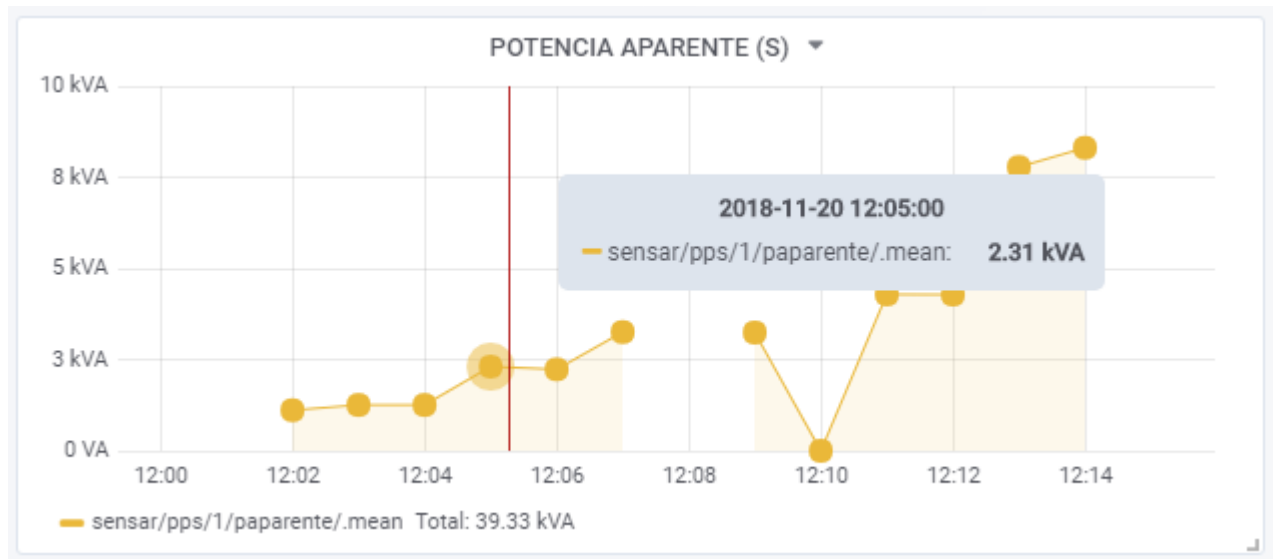


Figura 7.10: Gráfico Histórico – Valores de Potencia Aparente

7.5. Análisis del costo del prototipo final

En la tabla 7.4 se muestra el listado de componentes y los precios finales con impuestos para cada componente actualizadas al mes de Febrero de 2019.

Elemento	Precio
Fuente de alimentación (Salida 5V dc – 2000 mA)	\$ 250
Fuente Step Up	\$ 128
Arduino Nano	\$ 260
NodeMCU v1.0	\$ 280
Sensor SCT 013	\$ 550
Transformador 220/12 Vac	\$ 230
Resistencias (1/4 W): 22; 820; 1 k (2); 10 k (5); 15 k (4); 470 k (2)	\$ 20
Amplificador Operacional LM358 (2)	\$ 40
Zócalo 8 pines (2)	\$ 10
Capacitor 104 (2)	\$ 5
Jack hembra 3,5 mm	\$ 25
Borneras 2 pines	\$ 20
TOTAL (precio final con impuestos)	\$1818

El análisis de costos permite observar que aún para un primer prototipo existe potencial para su implementación comercial debido a que el costo por cantidad de cada una de las piezas y sin considerar impuestos es considerablemente menor al costo final publicado.

8. Conclusiones y trabajos futuros

En este capítulo se presentan las conclusiones asociadas al proyecto y se detallan algunas propuestas para mejorar el dispositivo.

8.1. Conclusiones

En este proyecto se desarrolló un sistema capaz de monitorizar los parámetros característicos de un sistema eléctrico monofásico, cumpliendo con los objetivos establecidos inicialmente.

Se realizó una propuesta para la medición y adaptación de las señales de corriente y tensión a valores requeridos por el microcontrolador utilizado en la adquisición de datos.

Se realizó una modelación y simulación mediante software de los circuitos propuestos y a partir de ella se implementó el más adecuado.

Se utilizó una placa Arduino Nano, la cual contiene al microcontrolador empleado en la adquisición de datos y donde se realizan además gran parte de los cálculos requeridos. Luego los resultados obtenidos se envían a una placa NodeMCU.

Para establecer una comunicación inalámbrica entre el medidor y la plataforma web de visualización, se implementó el protocolo de comunicación MQTT. De esta manera los datos recibidos en NodeMCU son retransmitidos a un servidor. Finalmente la información es almacenada y graficada.

Este diseño de prototipo de medidor se encuentra limitado en el sentido de poder medir pequeñas corrientes.

Se analizó el listado de componentes utilizados para el proyecto junto con los respectivos precios. Haciendo una comparación con los dispositivos comerciales presentados en la sección 3.1, el costo del prototipo desarrollado resulta competitivo con las propuestas comerciales. Puede requerir un análisis más profundo la comparación de costos luego de realizar algunas de las propuestas de mejoras presentadas en la siguiente sección de este capítulo, ya que se agregarían algunos componentes, y otros podrían ser reemplazados.

Desde un enfoque a nivel de investigación, resultó interesante el desarrollo de este prototipo, debido a que de una forma sencilla se pueden almacenar los datos de las variables medidas y se tiene un fácil acceso a ellos.

Es importante destacar que este trabajo presenta las bases para para otros proyectos a desarrollar en la Cooperativa y en la FI UNRC. En este sentido, los resultados de este trabajo serán enviados a evaluación para su presentación en la “XVIII Reunión de trabajo en Procesamiento de la Información y Control” a desarrollarse del 18 al 20 de septiembre de 2019 en Bahía Blanca.

8.2. Trabajos futuros

Dado el estado de avance del trabajo y los resultados obtenidos, se pudieron identificar algunos puntos para continuar el desarrollo propuesto. En este punto se detallan algunas posibles mejoras que se realizarían para pasar del prototipo a un producto terminado:

- Proteger los pines ADC de arduino mediante una configuración de diodos, tal como se muestra en la figura 8.1. De esta manera, si una tensión superase los 5 V más la caída de tensión en el diodo (0.7 para diodos de silicio, 0.2 para diodos de germanio), conduciría el diodo superior, derivando corriente hacia la fuente. Por otro lado, si ingresara una tensión de valor negativo, conduciría el diodo inferior cuando la señal sea menor a 0 V (menos la caída en el diodo).

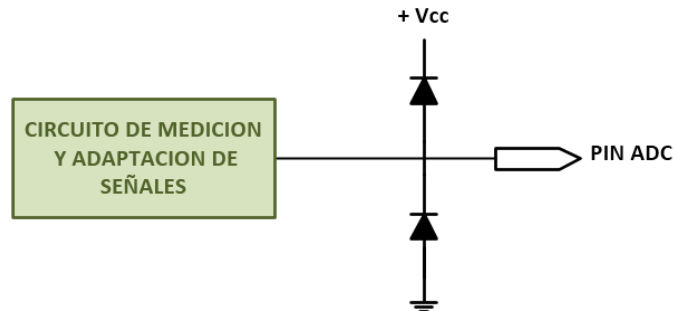


Figura 8.1: Protección de pines ADC

- En el presente del prototipo se necesitan dos tomacorrientes: uno para el transformador de medición de tensión y otro para la fuente de 5V. Luego, se tiene en serie la Fuente Step Up configurada a 9V.

Se plantea como mejora, reemplazar las dos fuentes de alimentación por una sola fuente switching de 9V, y derivar en paralelo la alimentación hacia el transformador de 220/12 V, tal como se observa en la figura 8.2.

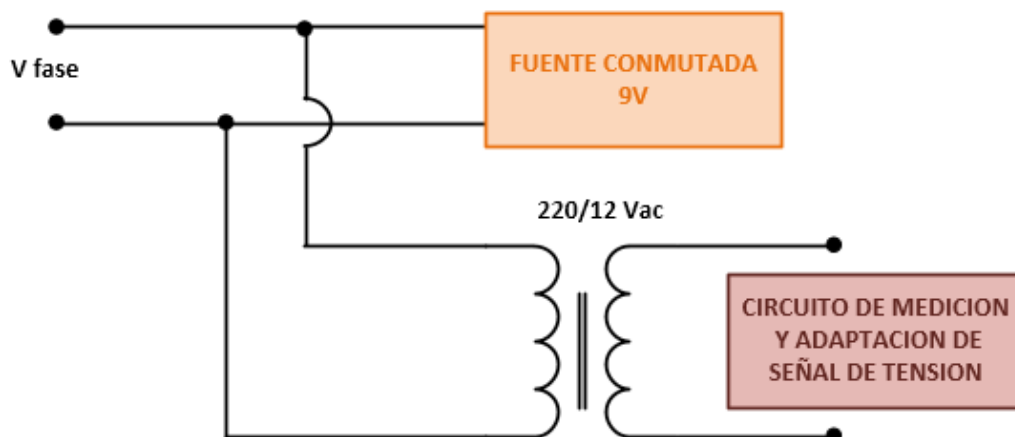


Figura 8.2: Alternativa para alimentación del medidor y transformador de tensión

- Resolviendo los puntos anteriores, se debería pasar de las placas experimentales a un circuito impreso o PCB, obteniendo un producto más compacto.
- Otra propuesta es realizar una validación utilizando cargas inductivas, para tener un detalle de las respuestas del medidor frente a los tipos de cargas más comunes que se pueden encontrar en cualquier instalación.

9. Bibliografía

- [1].Lagarrigue E. Descentralización y lo que esto significa. Ingeniería Eléctrica. Editores SRL. PP 12-14. Septiembre 2018. Página consultada 02/2019 https://www.editores-srl.com.ar/sites/default/files/ie335_schneider_descentralizacion.pdf
- [2].Donato G. Medidores inteligentes en Argentina: consideraciones para una implementación adecuada. Ingeniería Eléctrica. Editores SRL. Agosto 2017. Página consultada 02/2019 https://www.editores-srl.com.ar/sites/default/files/ie323_donato_medidores_inteligentes.pdf
- [3].Medidor inteligente. (2019, 7 de marzo). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 19:06, abril 2, 2019 desde https://es.wikipedia.org/w/index.php?title=Medidor_inteligente&oldid=114427787.
- [4].Barai, G.R., S. Krishnan, and B. Venkatesh. Smart metering and functionalities of smart meters in smart grid - a review. in 2015 IEEE Electrical Power and Energy Conference (EPEC). 2015.
- [5].Metering, A.S., S. Visalatchi, and K.K. Sandeep. Smart energy metering and power theft control using arduino & GSM. in 2017 2nd International Conference for Convergence in Technology (I2CT). 2017.
- [6].Reyes F. ¿Cómo funciona la primera red eléctrica inteligente de Argentina? INFOBAE 14 de julio de 2018. Página consultada 02/2019 <https://www.infobae.com/def/desarrollo/2018/07/14/como-funciona-la-primera-red-electrica-inteligente-de-argentina/>
- [7].Leme, M.O., et al., Digital Energy Management for Houses and Small Industries Based on a Low-cost Hardware. IEEE Latin America Transactions, 2016. 14(10): p. 4275-4278.
- [8].<https://www.arduino.cc/en/Guide/Introduction> - Página consultada 02/2019.
- [9].Reineri C. Armónicos en Sistemas Eléctricos: Fundamentos y Análisis.
- [10].Wikipedia contributors. (2019, February 25). MQTT. In Wikipedia, The Free Encyclopedia. Retrieved 14:14, March 1, 2019, from <https://en.wikipedia.org/w/index.php?title=MQTT&oldid=885079751>
- [11].ESP8266. (2019, 1 de enero). Wikipedia. Fecha de consulta: 15:36, marzo de 2019, desde <https://es.wikipedia.org/w/index.php?title=ESP8266&oldid=112986890>.
- [12].Colaboradores de Wikipedia. (2019, 13 de febrero). InfluxDB. En Wikipedia, The Free Encyclopedia. Obtenido 12:16, 2 de marzo de 2019, de <https://en.wikipedia.org/w/index.php?title=InfluxDB&oldid=883126203>
- [13].Ministerio de Energía y Minería. Secretaría de Coordinación de Planeamiento Energético. Dirección Nacional de Información Energética. Tecnología de la Información. Cálculo Factor de Emisión CO2 2015 (02/11/2016) Página consultada 02/2019 <https://datos.gob.ar/dataset/energia-calculo-factor-emision-co2-red-argentina-energia-electrica>
- [14].80 Series V Industrial True-RMS Multimeter with Temperature https://dam-assets.fluke.com/s3fs-public/2137623_6116_ENG_B_W.PDF
- [15].PROVA 11 AC/DC mA Clamp Meter. <https://aysan-market.com/wp-content/uploads/2016/10/prova11new.pdf>

10. Anexos

- Anexo 1: Datasheet SCT-013

Split core current transformer



Model: SCT-013 **Rated input current: 5A/100A**

Characteristics: Opening size: 13mm*13mm,

Non-linearity±3% (10%—120% of rated input current)

Im leading wire, standard Φ3.5 three core plug output.

Current output type and voltage output type (voltage output type built-in sampling resistor)

Purpose: Used for current measurement, monitor and protection for AC motor, lighting equipment, air compressor etc

Core material: ferrite

Mechanical strength: the number of switching is not less than 1000 times(test at 25°C)

Safety index: Dielectric strength(between shell and output)1000V AC/1min

Fire resistance property: In accordance with UL94-Vo

Work temperature: -25°C ~ +70°C



Outline size diagram: (in mm)

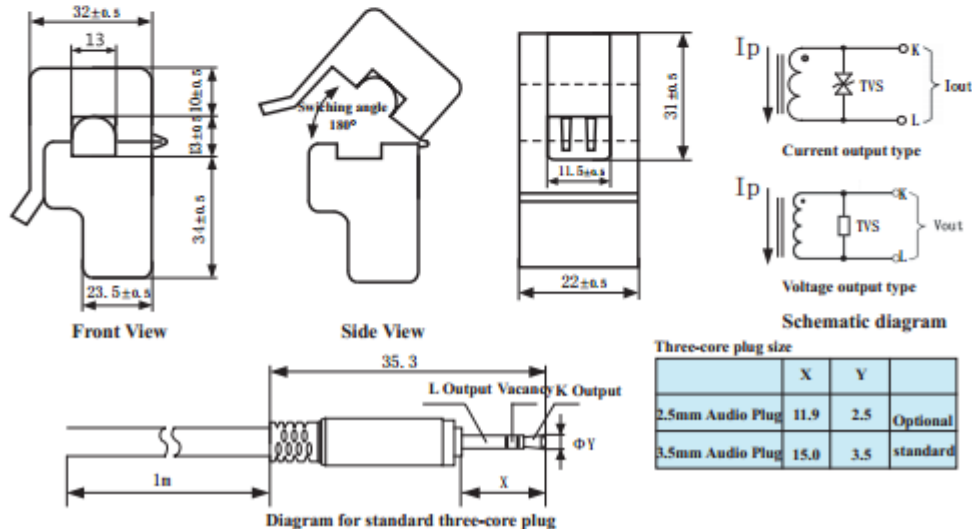


Table of technical parameter:

Model	SCT-013-000	SCT-013-005	SCT-013-010	SCT-013-015	SCT-013-020
Input current	0-100A	0-5A	0-10A	0-15A	0-20A
Output type	0-50mA	0-1V	0-1V	0-1V	0-1V
Model	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-000V
Input current	0-25A	0-30A	0-50A	0-60A	0-100A
Output type	0-1V	0-1V	0-1V	0-1V	0-1V

※ Output type: voltage output type built-in sampling resistor, current output type built-in protective diode.

Tel: 86-13933609279 Fax: 86-7929499-804 skype:macymeng1
 Web (China): www.yhdc.com Web (Latvia): www.yhdc.lv

- Anexo 2: Amplificador operacional LM358

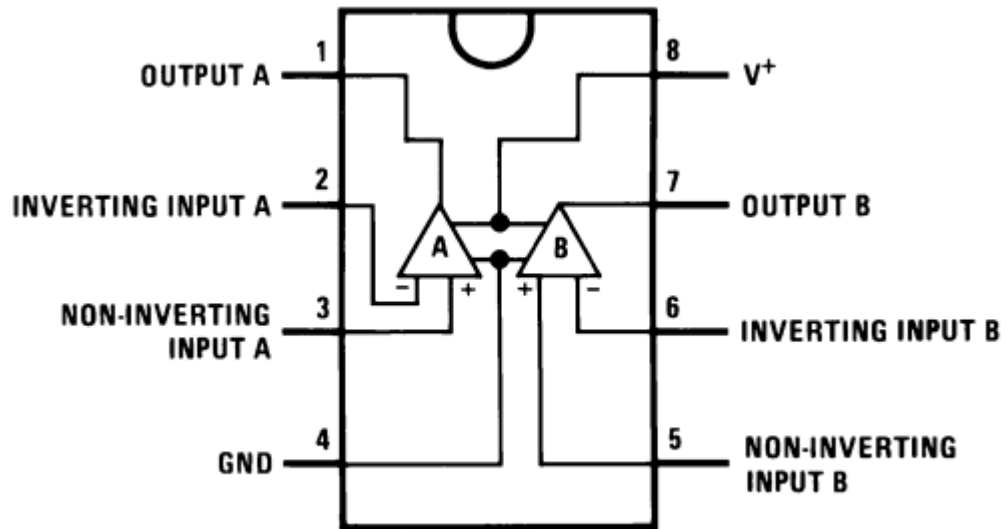
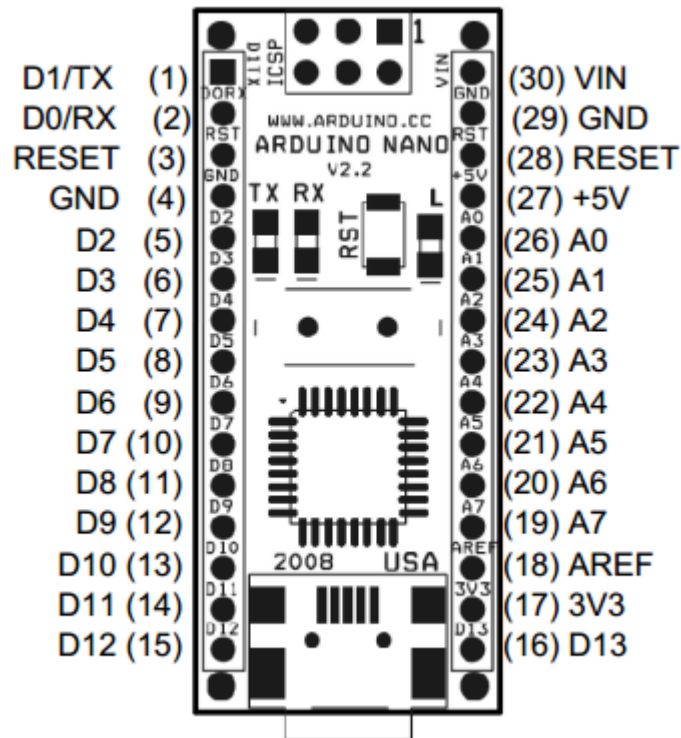


Figura 11.1: Configuración de pines y funciones

- Anexo 3: Arduino nano

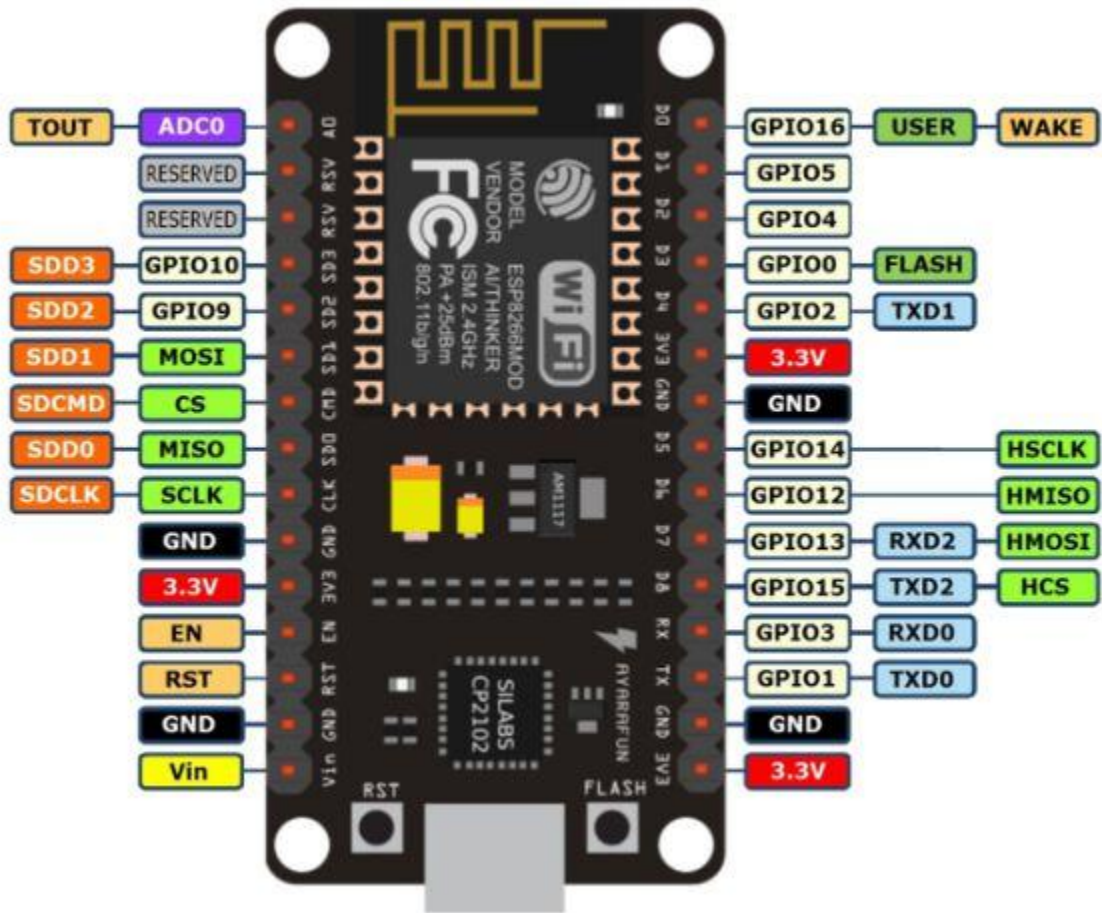


Alimentación de arduino nano

La placa tiene las siguientes entradas de alimentación:

- **Puerto mini USB:** llega al bus de 5V, procedente del puerto USB de un ordenador o de cualquier fuente de alimentación que posea tal salida.
- **Pin Vin:** no está protegida ante inversiones de polaridad. La conexión va directamente a la entrada del regulador. El polo negativo de voltaje se encuentra en los pines GND.
- **Pin 5V:** está conectado a la salida del regulador y al bus de 5V que se usa para alimentar cargas externas. En el caso de no aplicar tensión al puerto USB, este pin se puede utilizar para alimentar Arduino directamente (desde una fuente externa estabilizada de 5V), teniendo en cuenta que no hay ninguna forma de protección, ya que tanto el diodo como el fusible se encuentran por encima de este bus.

- Anexo 4: Pinout NodeMCU



• Anexo 5: Código fuente arduino

```

/*
////////////////////////////////////////////////////////////////////////////////
IT10 COOPERATIVA DE TRABAJO www.it10coop.com.ar

Modificaciones e implementación realizada por Ezequiel Silva a partir de plantilla
base provista por IT10

////////////////////////////////////////////////////////////////////////////////
*/

#include <SoftwareSerial.h>

SoftwareSerial Trans (10, 11); // RX, TX

#define OFFSET 2.5
#define CANT_MUESTRAS 10 //Cantidad de muestras por ciclo (20 milisegundos)
#define t 60000 //Tiempo de reseteo

// Variables Globales

float vector_i[CANT_MUESTRAS];
float vector_tension[CANT_MUESTRAS];
float I_rms;
float V_rms;
float P;
float S;

//-----

void(* resetFunc) () = 0;

void setup() {

    Serial.begin(9600);
    Serial.println("Iniciando Sensor");
    delay(500);

    Trans.begin(9600); // set the data rate for the SoftwareSerial port

    delay(500);

}

void loop() {

//-----//
//                                     MUESTRO DE SEÑALES                                     //
//-----//

    Serial.println("Iniciando muestreo ");

    delay(1000);

    unsigned long tiempo = millis();

```

```

for (short N = 0; N < CANT_MUESTRAS; N++) { // Ciclo de muestreo

    vector_i[N] = analogRead(A0);

    vector_tension[N] = analogRead(A4);

    delayMicroseconds(1680);

} // Fin ciclo de muestreo

Serial.print("Tiempo de muestreo: ");
Serial.print(millis()-tiempo);
Serial.println(" milisegundos ");

delay(4000);

// Llamado de funciones

V_rms = get_Vrms ();

I_rms = get_Irms ();

P = get_P ();

S = get_S ();

get_Q ();

float fp = get_fp (); //EL factor de potencia no se define como global, porque
                      su resultado no influye en otros parámetros.

Serial.println("----- ");

Trans.println('V'+String(V_rms)+'I'+String(I_rms)+'P'+String(P)+'S'+String(S)+
'f'+String(fp)); // Envío de datos a NodeMCU

Serial.println("Enviado");

delay(t);

resetFunc();

} // Fin de loop

//-----//
//                CÁLCULO DE VARIABLES                //
//-----//

// Tensión Eficaz (RMS)

float get_Vrms(){

```

```

float sum_V [10];

float s = 0;

for (short b = 0; b < CANT_MUESTRAS; b++) {

    Serial.print("Valor bit: ");
    Serial.print(vector_tension[b]);
    Serial.print(" ");

    vector_tension[b] = ((vector_tension[b] * 0.004883 - OFFSET) / 1.5)*230;
    //230 es la constante de regresión.

    Serial.print("tension: ");
    Serial.print(vector_tension[b]);
    Serial.print(" ");

    sum_V[b] = sq(vector_tension[b]);

    Serial.print(" Cuadrado: ");
    Serial.print(sum_V[b]);
    Serial.print(" ");

    Serial.println(" ");

}

for (short sv = 0; sv < CANT_MUESTRAS; sv++) {

    s = s + sum_V[sv];

}

Serial.print("Suma: ");
Serial.println(s);

float Vrms = sqrt(s/CANT_MUESTRAS);

Serial.print("Tension Eficaz: ");
Serial.print(Vrms);
Serial.println("V");

Serial.println("          -----          ");

return (Vrms);
}

//-----
// Corriente (RMS)

float get_Irms() {

float sum_I [10];

float r = 0;

```

```

for (short b = 0; b < CANT_MUESTRAS; b++) {

    Serial.print("Valor bit: ");
    Serial.print(vector_i[b]);
    Serial.print(" ");

    vector_i[b] = ((vector_i[b] * 0.004883 - OFFSET) / 1.5)*90.9; //

    Serial.print("corriente: ");
    Serial.print(vector_i[b]);
    Serial.print(" ");

    sum_I[b] = sq(vector_i[b]);

    Serial.print(" Cuadrado: ");
    Serial.print(sum_I[b]);
    Serial.print(" ");

    Serial.println(" ");

}

for (short si = 0; si < CANT_MUESTRAS; si++) {

    r = r + sum_I[si];

}

Serial.print("Suma: ");
Serial.println(r);

float Irms= sqrt(r / CANT_MUESTRAS);

if (Irms/90.9 <= 0.03) {

    Irms = 0;

}

Serial.print("Corriente RMS: ");
Serial.print(Irms);
Serial.println(" A");

Serial.println("          -----          ");

return(Irms);

}

//-----
// Potencia Activa

float get_P() {

```

```

float suma_p = 0;

float vector_p[CANT_MUESTRAS]; //Se construye con el producto punto a punto
de                               los valores de tensión y corriente medidos.

if (I_rms == 0 ){
    P = 0;
}
else {
    for (short vp = 0; vp < CANT_MUESTRAS; vp++) {

        Serial.print("v: ");
        Serial.print(vector_tension[vp]);
        Serial.print("    I: ");
        Serial.print(vector_i[vp]);

        vector_p[vp] = vector_i[vp] * vector_tension[vp];

        Serial.print("    P: ");
        Serial.print(vector_p[vp]);
        Serial.println();

    }

    for (short p = 0; p < CANT_MUESTRAS; p++) {

        suma_p = suma_p + vector_p[p];

    }

    P = suma_p/CANT_MUESTRAS;

    Serial.print("Pot Activa: ");
    Serial.print(P);
    Serial.println(" W");

    return(P);
}

//-----
// Potencia aparente

float get_S () {

    S = V_rms * I_rms;

    Serial.print("Pot Aparente: ");
    Serial.print(S);
    Serial.println(" VA");

    return (S);
}

```

```

//-----
// Potencia reactiva

float get_Q () {

    float Q = sqrt(S*S - P*P);

    Serial.print("Pot Reactiva: ");
    Serial.print(Q);
    Serial.println(" VAr");

}

//-----
// Factor de potencia

float get_fp () {

    float fp;

    if (I_rms == 0.00) {

        fp = 0;
    }

    else {

        fp = P / S;

    }

    Serial.print("Factor de Potencia: ");
    Serial.print(fp);
    Serial.println(" ");

    return(fp);

}

```

- Anexo 6: Código fuente NodeMCU

```

/*
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
IT10 COOPERATIVA DE TRABAJO www.it10coop.com.ar

Modificaciones e implementación realizada por Ezequiel Silva a partir de plantilla
base provista por IT10

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
*/

#include <ESP8266WiFi.h> //Instalar desde
http://arduino.esp8266.com/stable/package_esp8266com_index.json

#include <PubSubClient.h> // Librería MQTT

#include <SoftwareSerial.h>

SoftwareSerial mySerial(13, 15); // RX, TX (D7--->(11 arduino), D8--->(10
arduino))

const int lonbuffer = 12; // Longitud del buffer.
char buffer[lonbuffer]; // Buffer para almacenar el comando.

String tension;
String corriente;
String P;
String S;
String f;
String x;

bool mostrar = false;

//Características de la red

const char* ssid = "eze08"; //Nombre de Red
const char* password = "35008mes"; //Contraseña

const char* mqtt_server = "192.168.1.2"; //Dirección del Broker

//-----

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;

void setup_wifi() { // Conexión a la red WiFi.

    delay(10);
    Serial.println();
    Serial.print("Conectando a ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

```

```

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
randomSeed(micros());

Serial.println("");
Serial.println("Conexion exitosa");
Serial.println("Direccion IP: ");
Serial.println(WiFi.localIP());
Serial.println("Gateway: ");
Serial.println(WiFi.gatewayIP()); // Imprime datos de conexión.
Serial.println("DNS: ");
Serial.println(WiFi.dnsIP());

} //Fin de setup_wifi

void reconnect() { //Reconexión del cliente

    while (!client.connected()) {

        client.connect("IT10", "it10test", "IT10test");
        Serial.print("Conectando al Broker...");

        if (client.connect("node")) {

            Serial.println("Conexión Exitosa");

        } else {

            Serial.print("Error, Reconexión ");
            Serial.print(client.state());
            Serial.println(" Reintentando en 5 segundos");

            delay(5000); // Demora de 5 segundos para la reconexión.
        }
    }
} //Fin de reconexión

void setup() {

    Serial.begin(9600);
    mySerial.begin(9600);

    WiFi.mode(WIFI_STA);
    setup_wifi();
    client.setServer(mqtt_server, 1883);

}

void loop() {

    Serial.println("Esperando datos... ");

    if(mySerial.available() !=0) {

```

```

Serial.println("Cadena de datos recibida");

mySerial.readBytesUntil('V', buffer, lonbuffer);
tension = mySerial.parseFloat();

mySerial.readBytesUntil('I', buffer, lonbuffer);
corriente = mySerial.parseFloat();

mySerial.readBytesUntil('P', buffer, lonbuffer);
P = mySerial.parseFloat();

mySerial.readBytesUntil('S', buffer, lonbuffer);
S = mySerial.parseFloat();

mySerial.readBytesUntil('f', buffer, lonbuffer);
f = mySerial.parseFloat();

mySerial.readBytesUntil('x', buffer, lonbuffer);
x = mySerial.parseFloat();

    mostrar = true;
}

if (mostrar) {

    Serial.print("V: ");
    Serial.println(tension);
    Serial.print("I: ");
    Serial.println(corriente);
    Serial.print("P: ");
    Serial.println(P);
    Serial.print("S: ");
    Serial.println(S);
    Serial.print("fp: ");
    Serial.println(f);

    Serial.println("-----");

//-----
// Concatenación de datos y envío al servidor

    String id = "";
    id += tension;
    int long_v = id.length()+1;
    char V[long_v];
    id.toCharArray(V, long_v);
    client.publish("senzar/pps/1/voltaje/",V);

    String id1 = "";
    id1 += corriente;
    int long_i = id1.length()+1;
    char I[long_i];
    id1.toCharArray(I, long_i);
    client.publish("senzar/pps/1/corriente/",I);

    String id2 = "";
    id2 += P;

```

```

int long_P = id2.length()+1;
char activa[long_P];
id2.toCharArray(activa, long_P);
client.publish("senzar/pps/1/pactiva/",activa);

String id3 = "";
id3 += S;
int long_S = id3.length()+1;
char aparente[long_S];
id3.toCharArray(aparente, long_S);
client.publish("senzar/pps/1/paparente/",aparente);

String id4 = "";
id4 += f;
int long_fp = id4.length()+1;
char fp[long_P];
id4.toCharArray(fp, long_fp);
client.publish("senzar/pps/1/factorpotencia/",fp);

Serial.println("Enviado");

mostrar = false;

delay (1000);
}

if (!client.connected()) {
  reconnect();
}

client.loop ();      // Revisa si el cliente está conectado

} //Fin del Loop

```